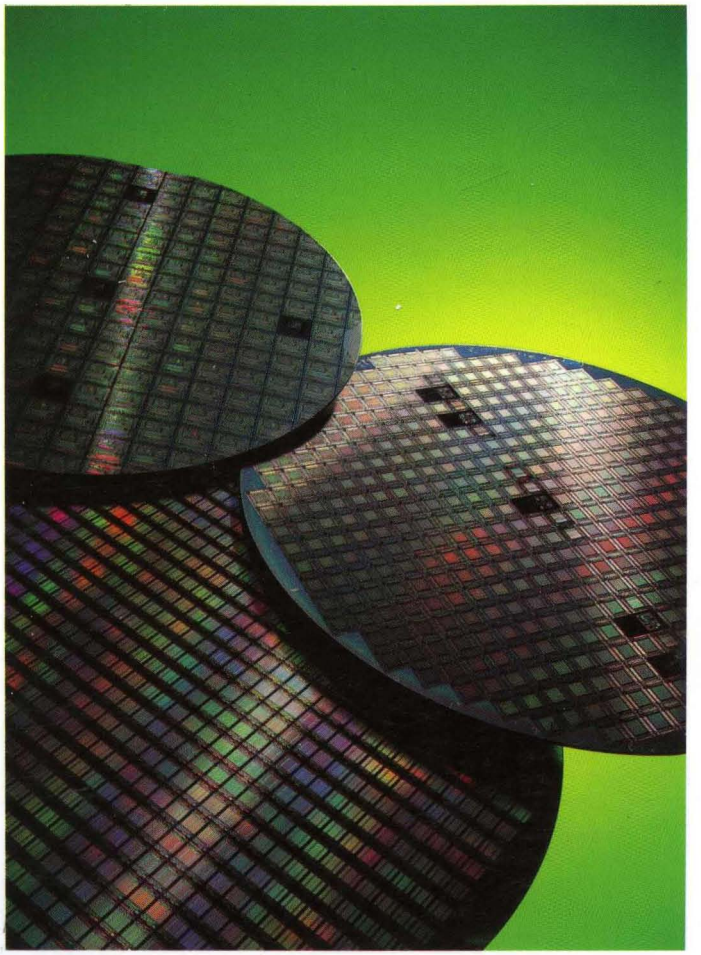
MHS

8 BIT MICROCONTROLLERS

1990

8 BIT MICROCONTROLLERS



1990

MHS

GENERAL INFORMATION

MHS BACKGROUND

MATRA MHS was formed in 1979 as a joint venture company beetween MATRA of France and HARRIS Corporation of the United States. Its charter was to develop a leading CMOS design and manufacturing operation in Europe.

In the early 80's, MHS reached its objectives and became a pioneer with several novative products, especially CMOS static memories.

Then, several other agreements contributed to MHS development.

In 1981, MHS signed an agreement with Intel Corp. covering the manufacture of NMOS circuits in Nantes, France and the establishment of a joint design facility for telecom chips and video controllers (82716).

Resultingly, MHS manufactured Intel's 8086, 8088, 8051 and 8052, as well as Harris' 80C86/88.

MHS was also entitled to desing CMOS versions of the 8051 MCU family. The 80C51 and its derivates have become one of MHS major successes, while fabrication of 16 bit MPU and NMOS devices was stopped, to concentrate on CMOS MCU.

In 1985, a joint venture was created between MHS and SGS Microelettronica to develop a fully automated assembly and test line for integrated circuits.

In a deal with Cypress, MHS received licensing rights to manufacture Cypress fast 16 K, 64 K and 256 K CMOS SRAMs and utilize Cypress fast 1.2 and 0.8 micron processes for MHS designs as well.

With France's national Telecom research labs (CNET), MHS developed an advanced sub-micron process, named Super-CMOS, to combine speed and low power consumption *(see page 4)*. This process is in production at MHS since 1988, while most new devices are designed to run on it.

Recently, MHS and NEC also signed a second-source agreement covering mutual manufacture and design rights of NEC's 78312A 16 bit microcontroller family.

MHS INDUSTRIAL CAPABILITY

MHS plant in Nantes (western France) includes a 2,000 m^2, class 10 wafer fab which is capable to produce 100,000 125 mm wafers per year.

Around 15 million integrated circuits are shipped every year by MHS.

The Nantes operation also has its own assembly and test lines, as well as R & D and Quality departments.

This factory has been fully qualified by most major military and space agencies according to their highest standards ; its quality has also been praised by some of the world's most demanding I.C. users.

MHS SALES NETWORK

MHS has its own worldwide sales and distribution network, with direct subsidiaries in Paris, Munich, London, Milan, Stockolm, Santa-Clara, and Hong-Kong.

These locations also have a technical center to ensure local support for MHS' expertise inintegration and ASIC design.

MHS PRODUCT OFFERING

MHS offering includes four main product lines, all in CMOS ; most circuits are available in commercial, indusrtrial, and military temperature ranges.

MHS is also a leading European manufacturer of Hi-Rel devices for military, aeronautics and space applications : its factory has been certified AQAP-1 and a variety of products have been listed by the corresponding agencies.

MICROCONTROLLERS

The 8051 family in CMOS, with a complete palette of options :

- ROM capacity from 4 K to 32 Kbytes : 80C51, 80C52, 83C154, 83C154D.
- Low-voltage (2.7 V), fuse-protected "secret ROM", high-speed (20 MHz) versions.
- Quick ROM service : three weeks for ROM code customization.
- and a specific single-chip keyboard controller : 80C752.

STATIC RAMs

- **fast** 16 K & 64 K devices : HM65728/767/768 (down to 15 ns) & HM65764/787-790 and a fast 8 K x 9 : HM65779, and 256 K devices 65756/65797/65798
- **very low-power** 16 K and 64 K memories (6 transistors per cell) : HM65162/262 and HM65641 (8K x 8, 55 ns, 1 µA)
- **"ultimate"** 64 K SRAM, such as HM65664/65687/65688 : 35 ns, 1 µA
- application-specific memories.
- ECL SRAM 256 x 4 and 1 K x 4 in BICMOS process technology. Access time : 3 ns.

DATACOM PRODUCTS

- a family of combo devices (HC3054/57) compatible with a market standard
- specific chips for modem applications, the 29C42 error correction circuit for V42/LAP M Modems, the HC55421 X21 interface, etc.

MHS is introducing a range of circuits dedicated to ISDN applications : rate adaptators (29C93), HDLC and ECMA102 multiplexed controllers, video codecs...

These products can be used in Terminal Equipments or at the Network Termination end.

They are targeted for the upcoming generation of equipment requiring powerful, flexible and low-cost components.

CMOS ASIC

- Five **gate-array** families, with gate-counts from 250 up to 55,000 gates, ultra-fast CMOS arrays, proprietary and standard software tools running on VAX, SUN, or turnkey systems such as DAISY, MENTOR, VALID, HP.
- Two families of **Composite arrays** mixing optimized blocks (RAM, ROM, & others) with regular arrays of gates.
- Specific **smart software** for system analysis and logical synthetisis.
- Digital and/or analog **custom designs** capabilities using standard software from Silicon Compilers Systems Corp. : GDT and genesil.
- Field technical centers in most of its subsidiaries.

MHS DIFFERENTIATION

After having successfully proved its ability to provide the electronic market with quality CMOS standard products as well as ASICs, MHS has decided to offer additional and newer solutions for system integration.

We named it : "THE TOOLBOX".

It combines our best strengthes :

- A unique sub-micron process : the Super-CMOS.
- A proven experience in making microcontrollers, SRAM, ASIC and Telecom chips which results in our mastering several of the most frequent functionalities needed in modern electronic systems.
- A design methodology based the availability of the above functions in a building block form, and a set of advanced design tools - including silicon compilation - which allows to mix them on a single I.C. as required.
- The total flexibility of these tools, which offers the possibility to develop most specific product in gate-array, composite array, or optimized silicon, according to our customer needs, expressed in terms of time-to-market, prototype cost, and production price.
- A team of very capable system and device architects, fully dedicated to analyze and discuss our customers' needs, in order to choose with them the best suited architecture and mean of integration to completely satisfy their specific requirements.

The above concept has been working so well that we have developped priviledged relations with our customers. As a result, we are able to move with them through the frontier between standard and user specific I.C. We are diluting this frontier ; our customer's experience has fertilized our tool box, and MHS's constantly richer tool box allows us to propose more and smarter solutions every day.

With our customers, we became :

A CONSULTANT IN INTEGRATION
and
AN EXPERT IN INTEGRATION

ALTERNATIVES FOR SYSTEM INTEGRATION

Each of the below criteria come from one or several different departments of our customers : marketing, design, manufacturing, and purchasing or finance. Program managers will evaluate the appropriate trade-offs, depending on the context of each program, including technical, market, and other specific factors.

This will help in choosing with MHS the best suitable solution and planning, taking into account possibilities offered by MHS to shift or evolve from one solution to another one at a further program stage.

SAMPLE OF CRITERIA	STANDARD CIRCUITS & PLD, GATE-ARRAYS	FULL CUSTOM CIRCUITS	COMPOSITE ARRAYS	OPTIMIZED* PARTITION
Breadboarding	++	–	–	+
Fast redesign	++	–	++	+
Level of integration	–	++	+	+
Complex systems	–	+	+/–	++
Flexibility	++	–	+	++
Technical risks	+	–	+	+
Test & emulation	++	–	+	++
Development cost	++	–	+	+
First production cost	–	–/+	+	+
Mature production cost	–	++	+	+
Second source	++	–	+	+
Confidentiality	–/+	++	+	+

(+ = satisfactory ; – = poor)

* **Note :** the optimized partition is a multi-chip solution based on standard circuits and composite arrays, for which the balance of criteria is often more favorable than solutions using custom circuits.

MHS SCMOS PROCESS

1. MHS COMMITMENT TO CMOS

For development and fabrication of the most advanced integrated circuits in both the ASIC and standard product fields MHS has made and stick to the choice of CMOS technology. A lot of developments and progress have been made from the initial 4 µm CMOS process, back in 1980, to the most recent processes bearing on materials, device physics or lithography which allow now the new technology to provide speed and high integration density on top of the traditionnal virtues of CMOS : low power consumption, wide voltage and temperature operating range or high noise immunity.

These continuous efforts made by the company in a very focused way around CMOS led MHS to the introduction in 1988 of the Super CMOS (SCMOS) technology on which most of the new MHS integrated circuits are now built or developed.

One single process however cannot fulfill all product requirements, and for this purpose complementary developments have been performed to derivate from a generic technology process varieties more suited to particular needs as presented below.

2. THE SCMOS PROCESS

Co-developed with the France's National Telecom Research Laboratory (CNET) the SCMOS goal has been to offer a generic process as described here before with submicronic minimum features to provide a very high speed potential as well as maximum integration capability. Such advantages however should not be gained against reliability characteristics which are of prime importance in highly integrated system especially for avionics or space application. For all these reasons special options have been taken in building up the process as can be viewed on the cross section of the double metal version of SCMOS. Let's review the key points of that construction :

THE SUBSTRATE

The latch-up phenomenon has always been a major concern of CMOS technology, that triggers parasitic thyristor which can generate very high current flows resulting in circuit non functionnality of destruction. By using special electrical structure on circuits I/O's and adopting careful layout rules inside the chips, accuity of the problem has been greatly reduced on MHS products. However, shrinking down the dimensions requires new solutions. Building the devices in a shallow high resistive layer epitaxially grown on very low resitivity substrate has proven its efficiency in killing the latch-up phenomenon. By using such P+ epitaxial wafers in SCMOS, voltage drops, induced by current injection in the substrate, are greatly reduced, while chance of triggering the parasitic SCR is close to zero, resulting in potential latch-up free circuits.

THE DEVICES

To guarantee high operation frequency of products, it is necessary to move toward submicronic transistor size in new processes. SCMOS emphasizes this trend as 0.8 micron drawn devices reach electrical channel length as low as 0.65 micron for the N channel transistor thus conferring high speed potential to the circuits.

Short channels - however - have several drawbacks that needed to be considered in SCMOS to guarantee reliable operation and keep high performance :

- To prevent punchthrough effects, such as voltage limitations and subthreshold currents degrading the circuit standby power consumption, in-depth study and optimization of transistor ion implant have been carried out.
- When going to short channel length, very high electrical fields are applied to electrons in the devices and the chance for those carriers to get enough energy to be injected in gate oxide becomes significant.
  This "hot electron" effect creates voltage and transconductance shifts that degrade device and circuit reliability. To counter this threat on SCMOS, both P and N channel transistors are built with LDD structures. This "Lightly Doped Drain" structure reduces the electrical fields in the devices drain vicinity thus lowering the probability of hot electron emission.

These two examples illustrate, among other actions, the particular care that was taken in designing the SCMOS devices to get the best performances without giving up any in reliability.

THE GATE MATERIAL

Used to build the transistor gates and being the first interconnection layer, this level has to have as low resistivity as possible. The SCMOS technology achieves that goal by replacing the polysilicon material by a bilayer of polysilicon which keeps the transistor threshold voltage characteristics, and Titanium Silicide which provides the low resistance. A tenfold improvement has been obtained through this solution, lowering typical value of this layer from 30 ohm/square down to 3 ohm/square. A great benefit results for all product performances and especially for memory, for which long word lines use to be realized with the gate level.

THE DOUBLE METAL SYSTEM

More and more circuits now require an enhanced routing capability, either to achieve higher integration density (memories), or to allow automation in placement and routing tasks (ASIC). If multilayer interconnection is a necessity, several limitations however have to be overcome to implement it efficiently in very dense technology. Among the concerns of double metal systems

let us mention the silicon metal interface, the metal step coverage, the risk of hillock formation generating shorts between metal layers, contact filling, intermetal dielectric planarization, via-contact stacking constrain, and electromigration.

Here again, with SCMOS, an innovative solution has been chosen to address most of these problems : it stands in the utilization of Tungsten as the first metal material, followed by a planarized intermetal dielectric before deposition of the 2nd metal layer using aluminium material.

Among other properties, Tungsten can be deposited by chemical vapor deposition technique which leads to better topology coverage, and offers the possibility of contact filling by an autoplanarization mechanism. Advantage is taken of this characteristic in allowing the stacking of vias and contact to interconnect the 2 metal layers. Such a possibility autorizes tighter metal pitches thus saving area in repetitive structures and interconnections.

Other advantages come with tungsten utilization, including better reliability resulting from the very good electromigration endurance of this material.

Through these specific developments, adding to the basic works in new techniques around lithography (direct stepping on wafers) or material deposition and etching as well as manufacturing engineering works, MHS as been able to master this advanced CMOS process.

With its characteristics, of which an abstract is given in *table 1*, the SCMOS appears as a key technology for the next 5 years. Thanks to innovative options that were made, it also bears the basics for future enhancements : it is already prepared for an analog version of

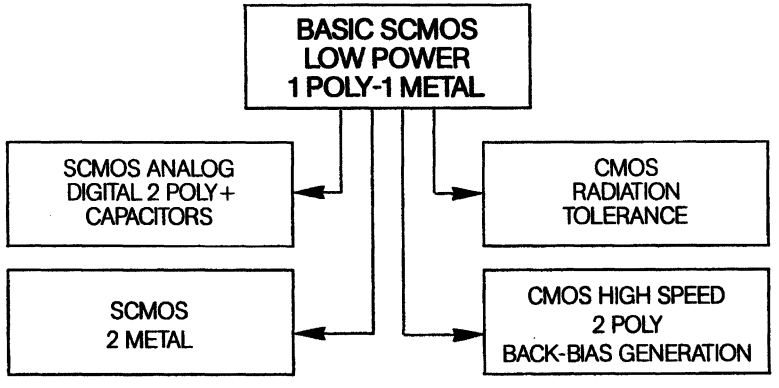
Gate oxide thickness	200 A
Electrical channel length • NMOS transistor • PMOS transistor	 0.65 µm 0.9 µm
Minimum drawn features	0.8 µm
Metal 1 pitch	2.9 µm
Metal 2 pitch	3.4 µm
Propagation delay	120 ps/gate
Integration density	25000 transistor/mm^2 (RAM)

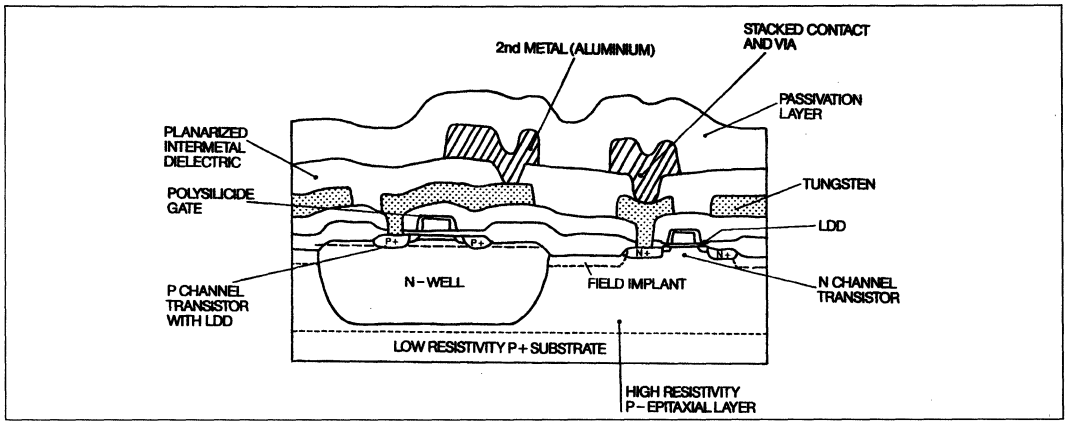
Table 1.

PRODUCT REQUIREMENTS

PRODUCTS	NEEDS	PROCESS
MICROCONTROLLER OR COMPLEX LOGIC	Digital high density	CMOS
DATACOM	Mixed analog digital	CMOS analog
MEMORIES	High density High speed	CMOS High speed
ASIC	Flexibility routeability	CMOS multilayer

PROCESS VARIANCES





MHS LITTERATURE

the same process and for even smaller lithographies (0.7 and 0.5 µm).

In order to provide our customers with a more exhaustive and regularly updated information, MHS has now split its data book into several handbooks.

Following volumes are available :

MICROS - MEMORIES - DATACOM - ASIC - GRAPHIC - HI-REL.

Each volume includes all data pertinent to its topic : data-sheets, application and technical notes, software/progamming manuals, as well as a cross-reference guide to common industry equivalents, if any.

Last chapters deal with general information : quality and reliability, dice/wafer form products, dice geometry index, package selection & dimensions guides, and sales network.

DATA SHEET CLASSIFICATION

CLASSIFICATION	PRODUCT STAGE	DISCLAIMERS
Preview	Formative or design	This document contains the design specifications for product under development. Specifications may be changed in any manner without notice.
Advance Information	Sampling or pre-production	This is advanced information, and specifications are subject to change without notice.
Preliminary	First production	Additional data may be published at a later date. MHS reserves the right to make changes at any time without notice, to improve design and supply the best possible product.

PRODUCT INDEX

HANDBOOK	TOPIC
8 BIT MICROCONTROLLERS	80C51/C31 ; 80C51-L/C31-L ; 80C51F ; 80C51S/C31S ; 80C52/32 ; 83C154 ; 83C154D ; 80C732/752.
16 BIT MICROCONTROLLER	78312/310 78312A/310A.
MEMORIES	HM 65687/688/664/767/768/770/772/728/787/788/789/790/791/764/779/797/798/799/795/796/756/162/262/641/161/6116L/6207/65231.
DATACOM	29C93/94/95, 29C42/43, 29C80/82/84. HC 55421/5570A/3052/3053/3054/3057.
ASIC	MA, MB, MAF, MC/MCR, MBM, MCM, CMOS Foundry, Macrocell 6402/I2C, Macrocells in Development, MA & MB Summary, Daisy, Mentor, Valid, Hewlett Packard, Gateaid II Vax, Superdesigner, Gasp.

MHS : A WORLD LEADER IN 80C51 FAMILY OF MICROCONTROLLERS

In 1985, MHS became a pioneer in CMOS microcontroller by introducing the CMOS version of the popular 8051.

Beyond the well-known advantages of CMOS such as low-power consumption, MHS design offered a fully static core, allowing chip operation down to zero MHz clock without data loss.

MHS kept on leading the way by continuoulsy introducing innovative versions :

1986 : - a low-voltage version (80C51-L) operating with V_{CC} down to 2.7 V,

1987 : - a "secret ROM" version (80C51F) with which simply blowing a fuse allows to protect ROM content from being read or dumped by any mean,

- the first CMOS version of 8052 : 8 K ROM and three timers.

1988 : - 16 K bytes of ROM and additional features with the 83C154,

- the "Quick ROM" service : a precious advantage providing our customers with customized ROM parts in less than 3 weeks (80C51 and 80C52),

- the long awaited 20 MHz version : 80C51S,

- a complete keyboard controller, integrated in one single I.C. : 80C752.

All above devices have been successful around the world in a wide variety of applications : over 8 million parts have been shipped.

MHS has more versions, more packages, more temperature ranges and screening levels than any other vendor on this family of microcontroller, for which MHS is by far the first european source.

1989 : - for software-greedy applications : at last a 32 K ROM device : 83C154D,

- piggy-back circuits 80C51 PX

- and a very clever solution for integrated systems...

If your already know most of all this, we will still surprize you. If you did not so far, trying MHS might help you to catch-up with those who know !

8 BIT MICROS

1

PRODUCT INDEX

MHS

PRODUCT INDEX

PART NUMBER	DESCRIPTION	PAGE
80C51/31	4 KBYTES ROM, 12 MHz VERSION	p 6-3
80C51/31-1	4 KBYTES ROM, 16 MHz VERSION	p 6-3
80C51/31S	4 KBYTES ROM, 20 MHz VERSION	p 6-23
80C51/31-L	4 KBYTES ROM, LOW POWER VERSION	p 6-41
80C51F	4 KBYTES ROM, ROM PROTECTED VERSION	p 6-59
80C52/32	8 KBYTES ROM, 12 MHz VERSION	p 6-77
80C52/32-1	8 KBYTES ROM, 16 MHz VERSION	p 6-77
80C52/32S	8 KBYTES ROM, 20 MHz VERSION	p 6-77
80C52/32-L	8 KBYTES ROM, LOW POWER VERSION	p 6-77
80C52F	8 KBYTES ROM, ROM PROTECTED VERSION	p 6-77
80C154/83C154	16 KBYTES ROM, 12 MHz VERSION	p 6-101
80C154/83C154-1	16 KBYTES ROM, 16 MHz VERSION	p 6-101
80C154/83C154-L	16 KBYTES ROM, LOW POWER VERSION	p 6-101
83C154F	16 KBYTES ROM, ROM PROTECTED VERSION	p 6-101
83C154D	83C154 WITH 32 KBYTES OF ROM	p 6-101
80C752/732	4 KBYTES ROM, KEYBOARD CONTROLLER	p 6-139

2

CROSS REFERENCE

CROSS REFERENCE

8 BIT MICROCONTROLLERS with 4K BYTES OF ON-CHIP ROM

Reference	Supplier	Package			Freq. MHz	Temp. Range				16 Bit Timer	RAM bytes	I/O	ICC max	Others
		DIL	PLCC	FLAT			I	A	M					
80C31/80C51	MHS	40	44	44	0-16	X	X	X	X	2	128	32	26	QUICK ROM
80C31-L/80C51-L	MHS	40	44	44	0-6	X	X	X	X	2	128	32	10	QUICK ROM/2.7 to 6 V
80C51F	MHS	40	44	...	0-16	X	X	X	X	2	128	32	26	QUICK AND SECRET ROM
80C31S/80C51S	MHS	40	44	44	0-20	X	X			2	128	32	32	QUICK ROM/20 MHz
80C752/732	MHS	40	44	...	0-12	X	X			3	256	32	26	QUICK ROM/KEYB. CONT.
80C31BH/80C51BH	INTEL	40	44	...	16	X	X			2	128	32	26	80C51BHP PROTEC. ROM
80C451/83C451	INTEL	...	68	...	16	X				2	128	56	28	MORE I/O
80C31BH/80C51BH	AMD	40	44	...	16	X	X			2	128	32	26	NO
MSM80C31/80C51	OKI	40	44	44	16	X	X	X		2	128	32	26	NO
80C31/80C51	PHILIPS	40	44	44	16	X	X	X		2	128	32	26	NO
80C451/83C451	PHILIPS	64	68	...	16	X	X			2	128	56	29	MORE I/O
80C550/83C550	PHILIPS	40	44	...	16	X	X			2	128	32	???	CAD + 2 PWM
80C851/83C851	PHILIPS	40	44	44	12	X	X			2	128	32	24	256 bytes EEPROM
SAB80C31/80C51	SIEMENS	40	44	...	12	X	X			2	128	32	???	

8 BIT MICROCONTROLLERS with 8K BYTES OF ON-CHIP ROM

Reference	Supplier	Package			Freq. MHz	Temp. Range				16 Bit Timer	RAM bytes	I/O	ICC max	Others
		DIL	PLCC	FLAT			I	A	M					
80C32/80C52	MHS	40	44	44	0-16	X	X	X	X	3	256	32	27	QUICK ROM
80C32-L/80C52-L	MHS	40	44	44	0-6	X	X	X	X	3	256	32	12	QUICK ROM/2.7 to 6 V
80C52F	MHS	40	44	...	0-16	X	X	X	X	3	256	32	27	QUICK AND SECRET ROM
80C32S/80C52S	MHS	40	44	44	0-20	X	X			3	256	32	32	QUICK ROM/20 MHz
80C32T2/80C52T2	AMD	40	44	...	16	X				2	256	32	???	NO
80C321/80C521	AMD	40	44	...	16	X				2	256	32	32	WATCHDOG + DDP
80C325/80C525	AMD	68	68	...	16	X				2	256	48	???	MORE I/O, SLAVE INT.
80C52/80C32	PHILIPS	40	44	...	16	X	X			3	256	32	???	I2C
80C552/83C552	PHILIPS	...	68	80	12	X	X	X		2	256	48	34	8 CAD, 2 PWM, I2C
80C652/83C652	PHILIPS	40	44	44	12	X	X	X		2	256	32	24	I2C
80C562/83C562	PHILIPS	...	68	80	16	X	X	X		2	256	48	45	CAD, 2 PWM
80C31FA/80C51FA	INTEL	40	44	...	16	X	X			3	256	32	39	NO
SAB80C32/80C52	SIEMENS	40	44	...	12	X				3	256	32	???	NO
SAB80C515/80C535	SIEMENS	...	68	...	12	X	X			3	256	48	45	CAD, MORE I/O
SAB80C517/80C537	SIEMENS	...	84	...	12	X				3	256	52	???	CAD, MORE I/O, 2 USART

8 BIT MICROCONTROLLERS with 16K BYTES OF ON-CHIP ROM

Reference	Supplier	Package			Freq.	Temp. Range				16 Bit	RAM	I/O	ICC	Others
		DIL	PLCC	FLAT	MHz		I	A	M	Timer	bytes		max	
80C154/83C154	MHS	40	44	44	0-16	X	X	X	X	3	256	32	36	WATCHDOG
80C154-L/83C154-L	MHS	40	44	44	0-6	X	X	X	X	3	256	32	16	WATCHDOG/2.7 to 6 V
83C154F	MHS	40	44	...	0-16	X	X	X	X	3	256	32	36	WATCHDOG/SECRET ROM
80C154/83C154	OKI	40	44	44	16	X	X	X		3	256	32	26	WATCHDOG
80C31FB/80C51FB	INTEL	40	44	...	16	X				3	256	32	???	NO
80C654/83C654	PHILIPS	40	44	44	16	X	X	X		2	256	32	38	I2C
80C541	AMD	40	44	...	12	X				2	256	32	???	NO

2

8 BIT MICROCONTROLLERS with 32K BYTES OF ON-CHIP ROM

Reference	Supplier	Package			Freq.	Temp. Range				16 Bit	RAM	I/O	ICC	Others
		DIL	PLCC	FLAT	MHz		I	A	M	Timer	bytes		max	
83C154D	MHS	40	44	44	0-16	X				3	256	32	36	83C154 with 32 KROM

DIL : Dual In Line ; PLCC : Plastic Lead Chip Carrier ; FLAT : Flat pack.
DDP : Dual Data Pointer.

MICROCONTROLLERS

M D 80C51 /B

- **TEMPERATURE RANGE**
  - BLANK : Commercial (only one character before device type)
  - I : Industrial – 40°C + 85°C
  - A : Automotive – 40°C + 125°C
  - M : Military – 55°C + 125°C
  - D : Dice probed at 25°C only
  - Q : Commercial with Burn-in (48 H/125°C)
  - L : Industrial with Burn-in (48 H/125°C)
- **PACKAGE TYPE**
  - C : Side brazed
  - D : Cerdip
  - P : Plastic
  - R : Leadless chip carrier
  - X : Dice form
  - S : PLCC
  - F : Flat Pack
  - J : J leaded LCC
- **MISCELLANEOUS**
  /B mil STD 883C
  - I 16 MHz
  - S 20 MHz
  - L 2.7 V to 6 V
- **ROM CODE : XXX**
- **SECRET ROM : Fxxx**
- **DEVICE TYPE**

EXAMPLES

P-80C31	: Standart plastic 8 bit microcontroller ROMless. Commercial temperature range.
ID-80C32	: Industrial temperature range (– 40°C + 85°C) Cerdip package
XX-80C31	: Dice probed at 25°C - packaged in chip-tray
D-80C32-1	: 80C32 in Cerdip - 16 MHz
MD-80C51/B	: 80C51 after mil std. 883 - cond. B Military temperature range (– 55°C + 125°C) Burn-in 168 H.

ARCHITECTURAL OVERVIEW OF THE MHS C51 FAMILY MICROCONTROLLERS

3

ARCHITECTURAL OVERVIEW OF THE MHS C51 FAMILY MICROCONTROLLERS

MEMBERS OF THE FAMILY

The MHS C51 family of microcontrollers consists of the devices listed in Table 1. The basic architectural structure of these devices is shown in figure 1.

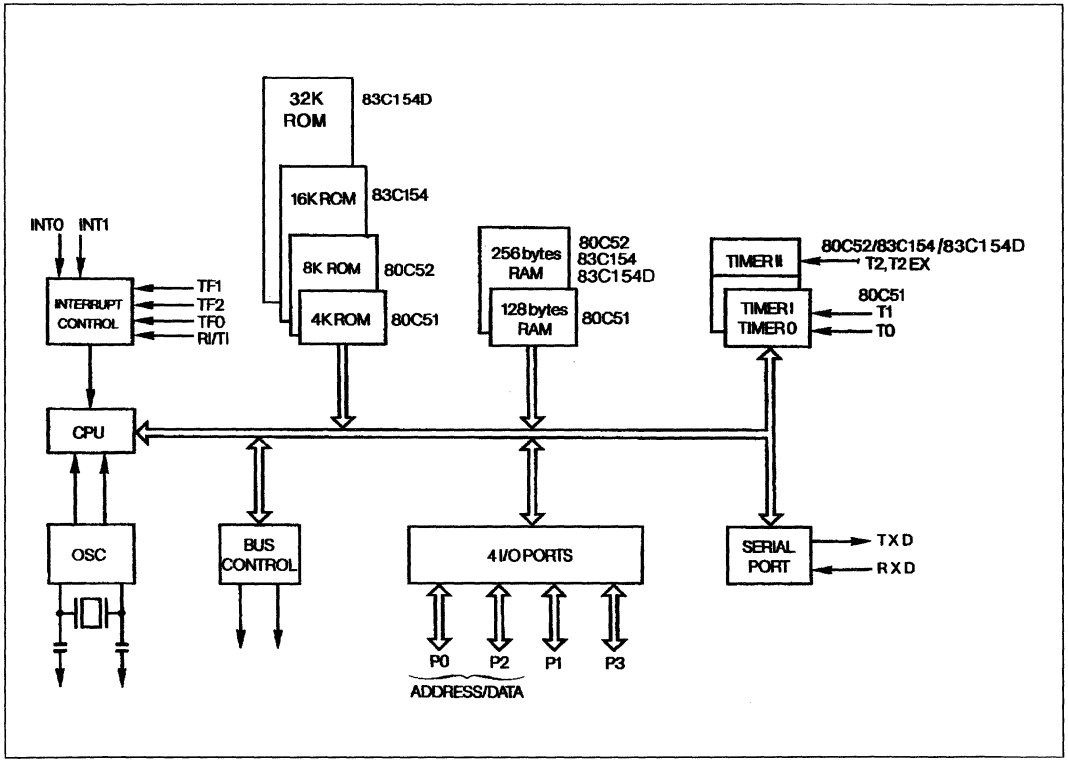


Figure 1 : Block Diagram of the 80C51/80C52/83C154/83C154D.

DEVICE NAME	ROMLESS VERSION	ROM BYTES	RAM BYTES	16-BIT TIMERS	TECHNO
80C51	80C31	4K	128	2	CMOS
80C52	80C32	8K	256	3	CMOS
83C154/C154D	80C154	16K/32K	256	3	CMOS

Table 1 : MHS C51 Family of Microcontrollers.

- 8-bit CPU optimized for control applications
- Extensive boolean processing (single-bit logic) capabilities
- 32 bidirectionnal and individually addressables I/O lines
- On chip clock oscillator.
- Full duplex UART

	80C51	80C52	83C154/C154D
• On Chip Program ROM	4 K bytes	8 K bytes	16 K/32 K bytes
• On Chip Data RAM	128 bytes	256 bytes	256 bytes
• External Data RAM	64 K bytes	64 K bytes	64 K bytes
• External Program Code	64 K bytes	64 K bytes	64 K bytes
• Timer/Counter	2	3	3
• Source Interrupt	5	6	6
• Priority Level	2	2	2

- The 80C51, 80C52, 83C154 and 83C154D differs from 80C31, 80C32 and 80C154 in having the on-chip program ROM. Instead, the 80C31, 80C32 and 80C154 fetches all instructions from external memory.

80C51

The 80C51 is the CMOS version of the 8051. Functionally, it is fully compatible with the 8051, but being CMOS it draws less current than its HMOS counterpart. To further exploit the power savings available in CMOS circuitry, two reduced power modes are added ;

- Software-invoked Idle Mode, during which the CPU is turned off while the RAM and other onchip peripherals continue operating. In this mode, current draw is reduced to about 15 % of the current drawn when the device is fully active.
- Software-invoked Power Down Mode, during which all on-chip activities are suspensed. The on-chip RAM continues to hold its data. In this mode the device typically draws less than 10 µA.

Although the 80C51 is functionally compatible with its HMOS counterpart, specific differences between the two types of devices must be considered in the design of an application circuit if one wishes to ensure complete interchangeability between the HMOS and CMOS devices.

The ROMless version of the 80C51 is the 80C31.

80C52

The 80C52 is an enhanced 80C51. It is fabricated with CMOS technology, and is backwards compatible with the 80C51. Its enhancements over the 80C51 are as follows :

- 256 bytes of on-chip RAM
- Three timer/counters
- 6-source interrupt structure
- 8 K bytes of on-chip Program ROM

The ROMless version of the 80C52 is the 80C32.

A separate product, the 80C52-BASIC, is an 80C52 with a full BASIC interpreter on-chip ROM.

83C154/C154D

The 83C154 is an enhanced 80C52. It is fabricated with CMOS technology, and is backwards compatible with the 80C52. Its enhancements over the 80C51 are as follows :

- 256 bytes of on-chip data RAM
- Three timer/counters (included watchdog and 32 bits timer/counters)
- 6 source interrupt structure
- Serial reception error detection
- New modes of power reduction consumption
- Programmable impedance port
- 16 K bytes of on-chip ROM for 83C154 and 32 K bytes for 83C154D
- Asynchronous Counter/Serial port mode during power-down

The ROMless version of the 83C154/C154D is the 80C154.

MEMORY ORGANIZATION IN MHS C51 DEVICES

LOGICAL SEPARATION OF PROGRAM AND DATA MEMORY

All MHS C51 devices have separate address spaces for program and Data Memory, as shown in figure 2. The logical separation of Program and Data Memory allows the Data Memory to be accessed by 8-bit addresses, which can be more quickly stored and manipulated by an 8-bit CPU. Nevertheless, 16-bit Data Memory addresses can also be generated through the DPTR register.

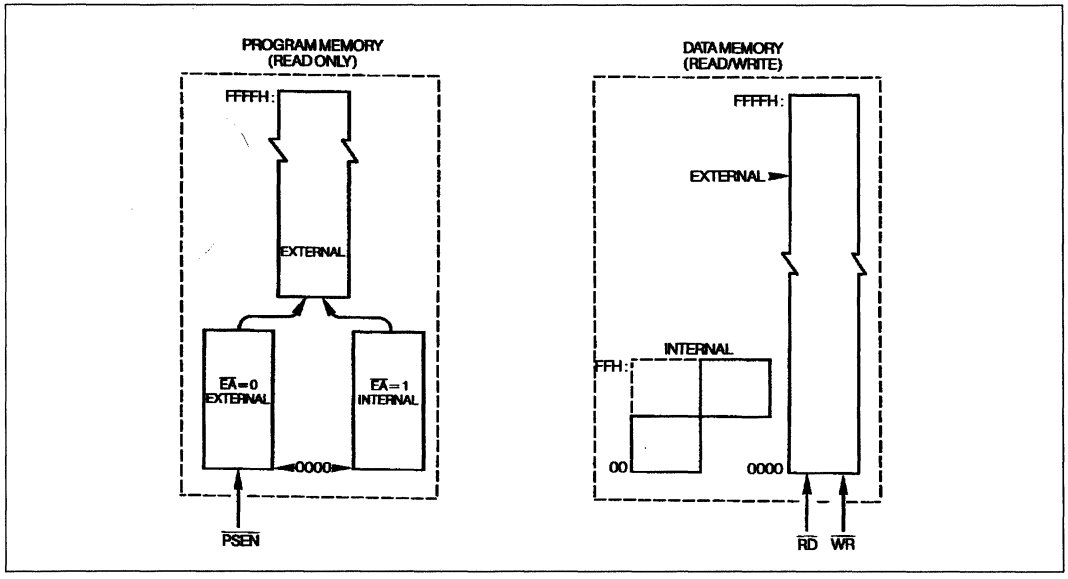


Figure 2 : MHS C51 Memory Structure.

Program Memory can only be read, not written to. There can be up to 64 K bytes of program Memory. In the 80C51 the lowest 4 K bytes of Program Memory are on-chip. The 80C52 provides 8 K bytes of on-chip Program Memory storage. The 83C154 provides 16 K bytes of on-chip Program Memory storage, and the 83C154D 32 K bytes. In the ROMless versions (80C31, 80C32, 80C154) all Program Memory is external. The read strobe for external Program Memory is the signal PSEN (Program Store Enable).

Data Memory occupies a separate address space from Program Memory. Up to 64 K bytes of external RAM can be addressed in the external Data Memory space. The CPU generates read and write signals, RD and WR, as needed during external Data Memory accesses. External Program Memory and external Data Memory may be combined if desired by applying the RD and PSEN signals to the inputs of an AND gate and using the output of the gate as the read strobe to the external Program/Data memory.

PROGRAM MEMORY

Figure 3 shows a map of the lower part of the Program Memory. After reset, the CPU begins execution from location 0000H.

As shown in Figure 3, each interrupt is assigned a fixed location in Program Memory. The interrupt causes the CPU to jump to that location, where it commences execution of the service routine. External Interrupt 0, for example, is assigned to location 0003H. If External Interrupt 0 is going to be used, its service routine must begin at location 0003H. If the interrupt is not going to be used, its service location is available as general purpose Program Memory.

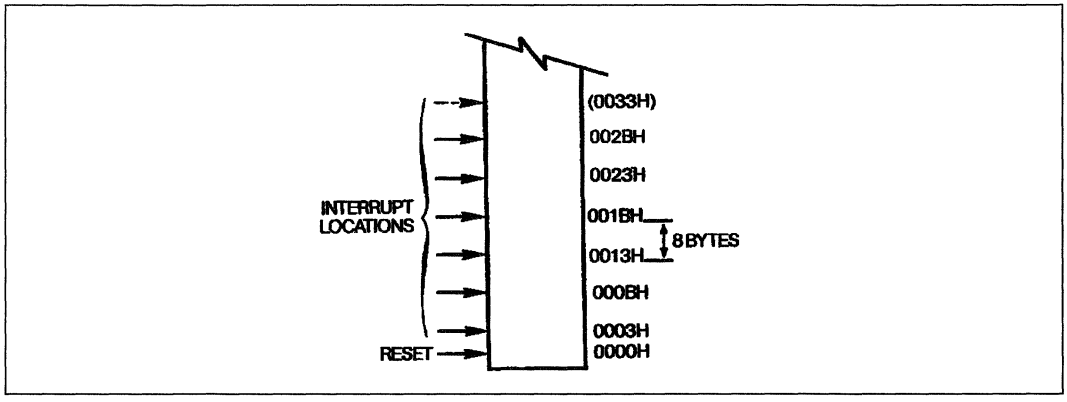


Figure 3 : MHS C51 Program Memory.

The interrupt service locations are spaced at 8-byte intervals : 0003H for External Interrupt 0, 000BH for Timer 0, 0013H for External Interrupt 1, 001BH for Timer 1, etc. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8-byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

The lowest 4 K (or 8 K in the 80C52 or 16 K in the 83C154 or 32 K in the 83C154D) bytes of Program Memory can be either in the on-chip ROM or in an external ROM. This selection is made by strapping the $\overline{EA}$ (External Access) pin to either V_{CC} or V_{SS}. In the 80C51 and its derivatives, if the $\overline{EA}$ pin is strapped to V_{CC}, then program fetches to addresses 0000H through 0FFFH are directed to the internal ROM. Program fetches to addresses 1000H through FFFFH are directed to external ROM.

In the 80C52, $\overline{EA} = V_{CC}$ selects addresses 0000H through 1FFFH to be internal, and addresses 2000H through FFFFH to be external.

In the 83C154, $\overline{EA} = V_{CC}$ selects addresses 0000H through 3FFFH to be internal, and addresses 4000H to FFFFH to be external.

In the 83C154D, $\overline{EA}$ = VCC selects addresses 0000H through 7FFFH to be internal and addresses 8000H to FFFFH to be external.

If the $\overline{EA}$ pin is strapped to V_{SS}, then all program fetches are directed to external ROM. The ROMless parts must have this pin externally strapped to V_{SS} to enable them to execute from external Program Memory.

The read strobe to external ROM, $\overline{PSEN}$, is used for all external program fetches. $\overline{PSEN}$ is not activated for internal program fetches.

The harware configuration for external program execution is shown in figure 4. Note that 16 I/O lines (Ports 0 and 2) are dedicated to bus functions during external Program Memory fetches. Port 0 (P0 in Figure 4) serves as a multiplixed address/data bus. It emits the low byte of the Program Counter (PCL) as an address, and then goes into a float state awaiting the arrival of the code byte from the Program Memory. During the time that the low byte of the Program Counter is valid on P0, the signal ALE (Address Latch Enable) clocks this byte into an address latch. Meanwhile, Port 2 (P2 in Figure 4) emits the high byte of Program Counter (PCH). Then PSEN strobes the EPROM and the code byte is read into the microcontroller.

Program Memory addresses are always 16 bits wide, even though the actual amount of Program Memory used may be less than 64 K bytes. External program execution sacrifices two of the 8-bit ports, P0 and P2, to the function of addressing the Program Memory.

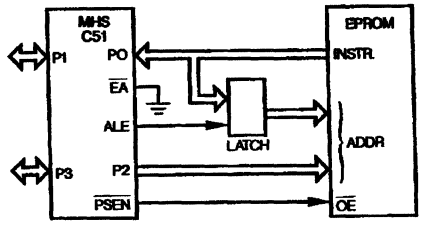


Figure 4 : Executing from External Program Memory.

DATA MEMORY

The right half of Figure 2 shows the internal and external Data Memory spaces available to the MHS C51 user. Figure 5 shows a hardware configuration for accessing up to 2 K bytes of external RAM. The CPU in this case is executing from internal ROM. Port 0 serves as multiplexed address/data bus to the RAM, and 3 lines of Port 2 are being used to page the RAM. The CPU generates RD and WR signals as needed during external RAM accesses.

There can be up to 64 K bytes of external Data Memory. External Data Memory addresses can be either 1 or 2 bytes wide. One-byte address is often used in conjunction with one or more other I/O lines to page the RAM, as shown in Figure 5. Two-byte addresses can also be used, in which case the address byte is emitted at Port 2.

Internal Data Memory is mapped in figure 6. The memory space is shown divided into three blocks, which are generally referred to as the lower 128, the Upper 128, and SFR space.

Internal Data Memory addresses are always one byte wide, which implies an address space of only 256 bytes. However, the addressing modes for internal RAM can in fact accomodate 384 bytes, using a simple trick. Direct addresses higher than 7FH access one memory space, and indirect addresses higher than 7FH access a different memory space. Thus figure 6 shows the Upper 128 and SFR space occupying the same block of addresses, 80H through FFH, although they are physically separate entities.

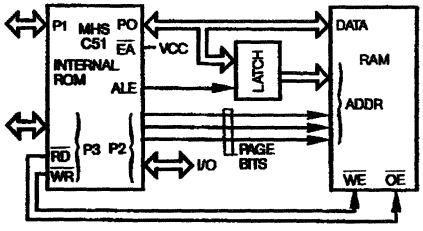


Figure 5 : Accessing External Data Memory. If the Program Memory is external, the other bits of P2 are available as I/O.

The Lower 128 bytes of RAM are present in all MHS C51 devices as mapped in Figure 7. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word (PSW) select which register bank is in use. This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing.

The next 16 bytes above the register banks form a block of bit-addressable memory space. The MHS-C51 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00H through 7FH.

All of the bytes in the Lower 128 can be accessed by either direct or indirect addressing. The Upper 128 (Figure 8) can only be accessed by indirect addressing. The Upper 128 bytes of RAM are not implemented in the 80C51 but are in the 80C52, 83C154 and 83C154D.

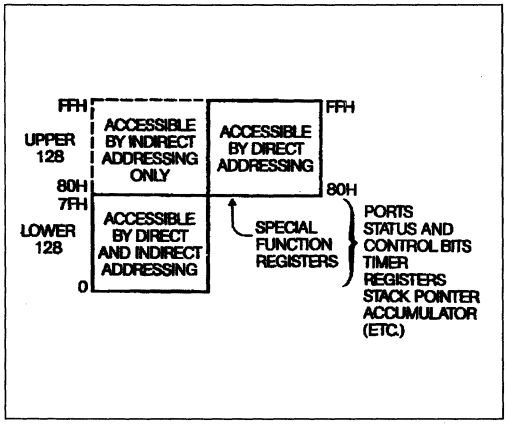


Figure 6 : Internal Data Memory.

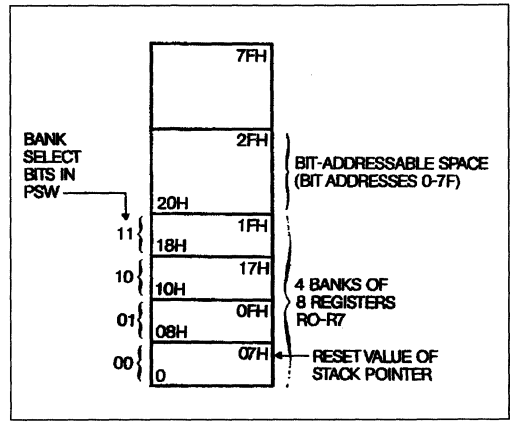


Figure 7 : The Lower 128 Bytes of Internal RAM.

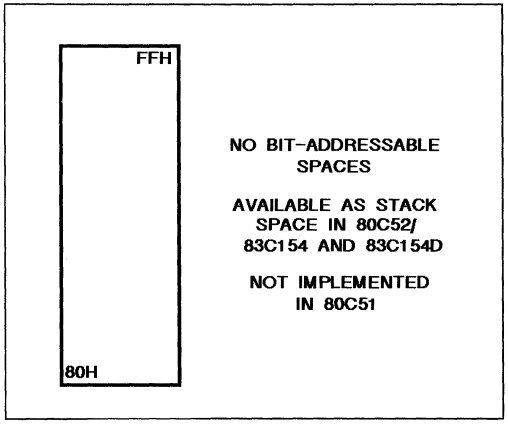


Figure 8 : The Upper 128 Bytes of Internal RAM.

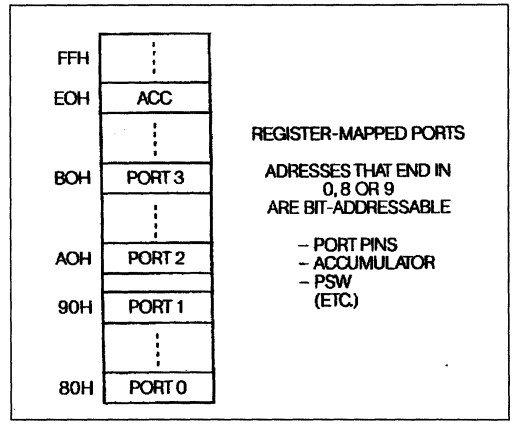


Figure 9 : SFR Space.

Figure 9 gives a brief look at the Special Function Register (SFR) space. SFRs include the Port latches, timers, peripheral controls, etc. These registers can only be accessed by direct addressing. In general, all MHS C51 microcontrollers have the same SFRs as the 80C51, and at the same addresses in SFR space. However, enhancements to the 80C51 have additional SFRs that are not present in the 80C51, nor perhaps in other proliferation of the family.

Sixteen addresses in SFR space are both byte-and bit-addressable. The bit-addressable SFRs are those whose ad-

THE MHS C51 INSTRUCTION SET

dress ends in 0, 8 or 9. The bit addresses in this area are 80H through FFH.

All members of the MHS C51 family execute the same instruction set. (except code A5H, skip opcode in MHS C51/C52). The MHS C51 instruction set is optimized for 8-bit control applications. It provides a variety of fast addressing modes for accessing the internal RAM to facilitate byte operations on small data structures. The instruction set provides extensive support for one-bit variables as a separate data type, allowing direct bit manipulation in control and logic systems that require Boolean processing.

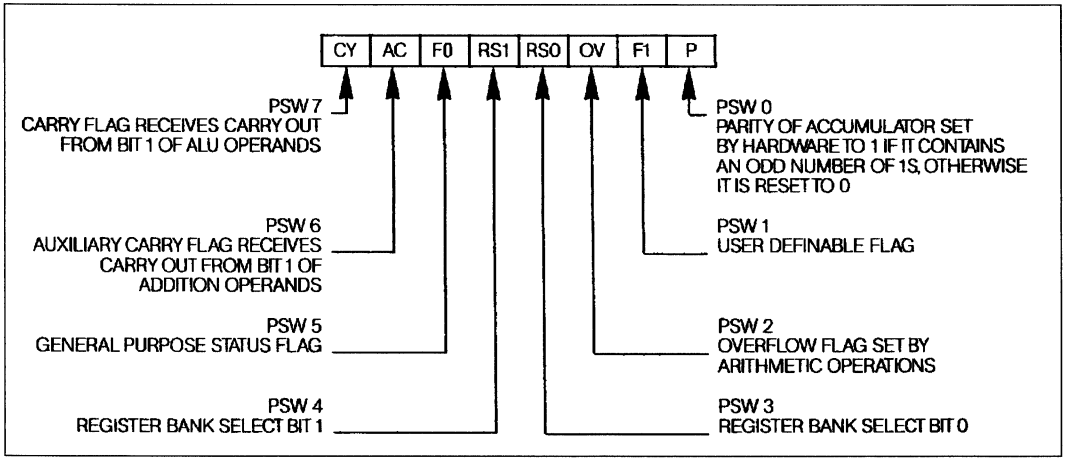


Figure 10 : PSW (Program Status Word) Register in MHS C51 Devices.

3

An overview of the MHS C51 instruction set is presented below, with a brief description of how certain instructions might be used.

PROGRAM STATUS WORD

The Program Status Word (PSW) contains several status bits that reflect the current state of the CPU. The PSW, shown in Figure 10, resides in SFR space. It contains the Carry bit, the Auxiliary Carry (for BCD operations), the two register bank select bits, the Overflow flag, a parity bit, and two user-definable status flags.

The Carry bit, other than serving the functions of a Carry bit in arithmetic operations, also serves as the "Accumulator" for a number of Boolean operations.

The bits RS0 and RS1 are used to select one of the four register banks shown in Figure 7. A number of instructions refer to these RAM locations as R0 through R7. The selection of which of the four banks is being referred to is made on the basis of the bits RS0 and RS1 at execution time.

The parity bit reflects the number of 1 s in the Accumulator : P = 1 if the Accumulator contains an odd number of 1 s, and P = 0 if the Accumulator contains an even number of 1 s. Thus the number of 1 s in the Accumulator plus P is always even.

Two bits in the PSW are uncommitted and may be used as general purpose status flags.

ADDRESSING MODES

The addressing modes in the MHS C51 instruction set are as follows :

Direct addressing

In direct addressing the operand is specified by an 8-bit address field in the instruction. Only 128 Lowest bytes of internal Data RAM and SFRs can be directly addressed.

Indirect addressing

In indirect addressing the instruction specifies a register which contains the address of the operand. Both internal and external RAM can be indirectly addressed.

The address register for 8-bit addresses can be R0 or R1 of the selected register bank, or the Stack Pointer.
The address register for 16-bit addresses can only be the 16-bit "data pointer" register, DPTR.

Register instructions

The register banks, containing registers R0 through R7, can be accessed by certain instructions which carry a 3-bit register specification within the opcode of the instruction. Instructions that access the registers this way are code efficient, since this mode eliminates an address byte. When the instruction is executed, one of the eight registers in

the selected bank is accessed. One of four banks is selected at execution time by the two bank select bits in the PSW.

Register-specific instructions

Some instructions are specific to a certain register. For example, some instructions always operate on the Accumulator, or Data Pointer, etc., so no address byte is needed to point to it. The opcode itself does that. Instructions that refer to the Accumulator as A assemble as accumulator-specific opcodes.

Immediate constants

The value of a constant can follow the opcode in Program Memory. For example,

MOV A, # 100

loads the Accumulator with the decimal number 100. The same number could be specified in hex digits as 64H.

Indexed addressing

Only Program Memory can be accessed with indexed addressing, and it can only be read. This addressing mode is intended for reading look-up tables in Program Memory. A 16-bit base register (either DPTR or the Program Counter) points to the base of the table, and the Accumulator is set up with the table entry number. The address of the table entry in Program Memory is formed by adding the Accumulator data to the base pointer.

Another type of indexed addressing is used in the "case jump" instruction. In this case the destination address of a jump instruction is computed as the sum of the base pointer and the Accumulator data.

ARITHMETIC INSTRUCTIONS

The menu of arithmetic instructions is listed in Table 2. The table indicates the addressing modes that can be used with each instruction to access the <byte> operand. For example, the ADD A, <byte> instruction can be written as :

ADD A, 7FH (direct addressing)
ADD A, @ RO (indirect addressing)
ADD A, R7 (register addressing)

MNEMONIC	OPERATION	ADDRESSING MODES				EXECUTION TIME (µs)
		Dir	Ind	Reg	Imm	
ADD A, <byte>	A = A + <byte>	X	X	X	X	1
ADDC A, <byte>	A = A + <byte> + C	X	X	X	X	1
SUBB A, <byte>	A = A – <byte> – C	X	X	X	X	1
INC A	A = A + 1	Accumulator only				1
INC <byte>	<byte> = <byte> + 1	X	X	X		1
INC DPTR	DPTR = DPTR + 1	Data Pointer only				2
DEC A	A = A – 1	Accumulator only				1
DEC <byte>	<byte> = <byte> – 1	X	X	X		1
MUL AB	B:A = B x A	ACC and B only				4
DIV AB	A = Int [A/B] B = Mod [A/B]	ACC and B only				4
DA A	Decimal Adjust	Accumulator only				1

Table 2 : A list of the MHS C51 Arithmetic Instructions.

ADD A, # 127 (immediate constant)

The execution times listed in Table 2 assume a 12 MHz clock frequency. All of the arithmetic instructions execute in 1 µs except the INC DPTR instruction, which takes 2 µs, and the Multiply and Divide instructions, which take 4 µs.

Note that any byte in the internal Data Memory space can be incremented or decremented without going through the Accumulator.

One of the INC instructions operates on the 16-bit Data Pointer. The Data Pointer is used to generate 16-bit addresses for external memory, so being able to increment it in one 16-bit operation is a useful feature.

The MUL AB instruction multiplies the Accumulator by the data in the B register and puts the 16-bit product into the concatenated B and Accumulator registers.

The DIV AB instruction divides the Accumulator by the data in the B register and leaves the 8-bit quotient in the Accumulator, and the 8-bit remainder in the B register.

Oddly enough, DIV AB finds less use in arithmetic "divide" routines than in radix conversions and programmable shift operations. An example of the use of DIV AB in a radix conversion will be given later. In shift operations, dividing a number by 2^n shifts its n bits to the right. Using DIV AB to perform the division completes the shift in 4 μs leaves the B register holding the bits that were shifted out.

The DA A instruction is for BCD arithmetic operations. In BCD arithmetic ADD and ADDC instructions should always be followed by a DA A operation, to ensure that the result is also in BDC. Note that DAA will not convert a binary number to BCD. The DA A operation produces a meaningful result only as the second step in the addition of two BCD bytes.

LOGICAL INSTRUCTIONS

Table 3 shows the list of MHS C51 logical instructions. The instructions that perform Boolean operations (AND, OR, Exclusive OR, NOT) on bytes perform the operation on a bit-by-bit basis. That is, if the Accumulator contains 00110101B and <byte> contains 01010011B, then

ANL A, <byte>

will leave the Accumulator holding 00010001B.
The addressing modes that can be used to access the <byte> operand are listed in Table 3. Thus, the ANL A, <byte> instruction may take any of the forms.

MNEMONIC	OPERATION	ADDRESSING MODES				EXECUTION TIME (μs)
		Dir	Ind	Reg	Imm	
ANL A, <byte>	A = A AND <byte>	X	X	X	X	1
ANL <byte>,A	<byte> = <byte> AND A	X				1
ANL <byte>, # data	<byte> = <byte> AND # data	X				2
ORL A, <byte>	A = A OR <byte>	X	X	X	X	1
ORL <byte>, A	<byte> = <byte> OR A	X				1
ORL <byte>, # data	<byte> = <byte> OR # data	X				2
XRL A, <byte>	A = A XOR <byte>	X	X	X	X	1
XRL <byte>,A	<byte> = <byte> XOR A	X				1
XRL <byte>,# data	<byte> = <byte> XOR # data	X				2
CLR A	A = 00H	Accumulator only				1
CPL A	A = NOT A	Accumulator only				1
RL A	Rotate ACC Left 1 bit	Accumulator only				1
RLC A	Rotate Left through Carry	Accumulator only				1
RR A	Rotate ACC Right 1 bit	Accumulator only				1
RRC A	Rotate Right through Carry	Accumulator only				1
SWAP A	Swap Nibbles in A	Accumulator only				1

Table 3 : A list of the MHS C51 Logical Instructions.

```
ANL  A, 7FH     (direct addressing)
ANL  A, @ R1    (indirect addressing)
ANL  A, R6      (register addressing)
ANL  A, # 53H   (immediate constant)
```

All of the logical instructions that are Accumulator specific in 1 µs (using a 12 MHz clock). The others take 2 µs.

Note that Boolean operations can be performed on any byte in the internal Data Memory space without going through the Accumulator. The XRL <byte>, # data instruction, for example, offers a quick and easy way to invert port bits, as in

XRL P1, #OFFH

If the operation is in response to an interrupt, not using the Accumulator saves the time and effort to stack it in the service routine.

The Rotate instructions (RLA, RLCA, etc.) shift the Accumulator 1 bit to the left or right. For a left rotation, the MSB rolls into the LSB position. For a right rotation, the LSB rolls into the MSB position.

The SWAP A instruction interchanges the high and low nibbles within the Accumulator. this is a useful operation in BCD manipulations. For example, if the Accumulator contains a binary number which is known to be less than 100, it can be quickly converted to BCD by the following code :


```
MOV  B, #10
DIV  AB
SWAP A
ADD  A,B
```

Dividing the number by 10 leaves the tens digit in the low nibble of the Accumulator, and the ones digit in the B register. The SWAP and ADD instructions move the tens digit to the high nibble of the Accumulator, and the ones digit to the low nibble.

DATA TRANSFERS

Internal RAM

Table 4 shows the menu of instructions that are available for moving data around within the internal memory spaces, and the addressing modes that can be used with each one. With a 12 MHz clock, all of these instructions execute in either 1 or 2 µs.

The MOV <dest>, <src> instruction allows data to be transfered between any two internal RAM or SFR locations without going through the Accumulator. Remember the Upper 128 bytes of data RAM can be accessed only by indirect, and SFR space only by direct addressing.

Note that in all MHS C51 devices, the stack resides in on-chip RAM, and grows upwards. The PUSH instruction first increments the Stack Pointer (SP), then copies the byte into the stack. PUSH and POP use only direct addressing to identify the byte being saved or restored, but the stack itself is accessed by indirect addressing using the SP register. This means the stack can go into the Upper 128, if they are implemented, but not into SFR space.

The Upper 128 are not implemented in the 80C51, nor in their ROMless. With these devices, if the SP points to the

MNEMONIC	OPERATION	ADDRESSING MODES				EXECUTION TIME (µs)
		Dir	Ind	Reg	Imm	
MOV A, <src>	A = <src>	X	X	X	X	1
MOV <dest>, A	<dest> = A	X	X	X		1
MOV <dest>, <scr>	<dest> = <scr>	X	X	X	X	2
MOV DPTR,# data 16	DPTR = 16-bit immediate constant				X	2
PUSH <scr>	INC SP : MOV"@SP", <scr>	X				2
POP <dest>	MOV <dest>, "@SP" : DEC SP	X				2
XCH A,<byte>	ACC and <byte> Exchange Data	X	X	X		1
XCHD A,@Ri	ACC and @ Ri exchange low nibbles		X			1

Table 4 : A list of the MHS C51 Data Transfer Instructions that Access Internal Data Memory Space.

Upper 128 PUSHed bytes are lost, and POPped bytes are indeterminate.

The Data Transfer instructions include a 16-bit MOV that can be used to initialize the Data Pointer (DPTR) for look-up tables in Program Memory, or for 16-bit external Data Memory accesses.

The XCH A, <byte> instruction causes the Accumulator and addressed byte to exchange data.
The XCHD A, @ Ri instruction is similar, but only the low nibbles are involved in the exchange.

The see how XCH and XCHD can be used to facilitate data manipulations, consider first the problem of shifting an 8-digit BCD number two digits to the right. Figure 11 shows how this can be done using direct MOVs, and for com-

	2A	2B	2C	2D	2E	ACC
MOV A,2EH	00	12	34	56	78	78
MOV 2EH, 2DH	00	12	34	56	56	78
MOV 2DH, 2CH	00	12	34	34	56	78
MOV 2CH, 2BH	00	12	12	34	56	78
MOV 2BH, # 0	00	00	12	34	56	78

(a) Using direct MOVs : 14 bytes, 9 µs

	2A	2B	2C	2D	2E	ACC
CLR A	00	12	34	56	78	00
XCH A,2BH	00	00	34	56	78	12
XCH A,2CH	00	00	12	56	78	34
XCH A,2DH	00	00	12	34	78	56
XCH A,2EH	00	00	12	34	56	78

(b) Using XCHs : 9 bytes, 5 µs

Figure 11 : Shifting a BCD Number Two Digits to the Right.

		2A	2B	2C	2D	2E	ACC
	MOV R1,# 2EH	00	12	34	56	78	XX
	MOV R0,# 2DH	00	12	34	56	78	XX
	loop for R1 = 2EH :						
LOOP :	MOV A, @R1	00	12	34	56	78	78
	XCHD A, @R0	00	12	34	58	78	76
	SWAP A	00	12	34	58	78	67
	MOV @R1, A	00	12	34	58	67	67
	DEC R1	00	12	34	58	67	67
	DEC R0	00	12	34	58	67	67
	CJNE R1, #2AH, LOOP						
	loop for R1 = 2DH :	00	12	38	45	67	45
	loop for R1 = 2CH :	00	18	23	45	67	23
	loop for R1 = 2BH :	08	01	23	45	67	01
	CLR A	08	01	23	45	67	00
	XCH A,2AH	00	01	23	45	67	08

Figure 12 : Shifting a BCD Number One Digit to the Right.

parison how it can be done using XCH instructions. To aid in understanding how the code works, the contents of the registers that are holding the BCD number and the content of the Accumulator are shown alongside each instruction to indicate their status after the instruction has been executed.

After the routine has been executed, the Accumulator contains the two digits that were shifted out on the right. Doing the routine with direct MOVs uses 14 code bytes and 9 µs of execution time (assuming a 12 MHz clock). The same operation with XCHs uses less code and executes almost twice as fast.

To right-shift by an odd number of digits, a one-digit shift must be executed. Figure 12 shows a sample of code that will right-shift a BCD number one digit, using the XCHD instruction. Again, the contents of the registers holding the number and of the Accumulator are shown alongside each instruction.

First, pointers R1 and R0 are set up to point to the two bytes containing the last four BCD digits. Then a loop is executed which leaves the last byte, location 2EH, holding the last two digits of the shifted number. The pointers are decremented, and the loop is repeated for location 2DH. The CJNE instruction (Compare and Jump if Not Equal) is a loop control that will be described later.

The loop is executed from LOOP to CJNE for R1 = 2EH, 2DH, 2CH and 2BH. At that point the digit that was originally shifted out on the right has propagated to location 2AH. Since that location should be left with 0s, the lost digit is moved to the Accumulator.

External RAM

ADDRESS WIDTH	MNEMONIC	OPERATION	EXECUTION TIME (µs)
8 bits	MOVX A, @ Ri	Read external RAM @ Ri	2
8 bits	MOVX @ Ri,A	Write external RAM @ Ri	2
16 bits	MOVX A, @ DPTR	Read external RAM @ DPTR	2
16 bits	MOVX @ DPTR,A	Write external RAM @ DPTR	2

Table 5 : A list of the MHS C51 Data Transfer Instructions that Access External Data Memory Space.

Table 5 shows a list of the Data Transfer instructions that access external Data Memory. Only indirect addressing can be used. The choice is whether to use a one-byte address, @Ri, where Ri can be either R0 or R1 of the selected register bank, or a two-byte address, @DPTR. The disadvantage to using 16-bit addresses if only a few K bytes of external RAM are involved is that 16-bit addresses use all 8 bits of Port 2 as address bus. On the other hand, 8-bit addresses allow one to address a few K bytes of RAM, as shown in Figure 5, without having to sacrifice all of Port 2.

All of these instructions execute in 2 µs, with a 12 MHz clock.

Note that in all external Data RAM accesses, the Accumulator is always either the destination or source of the data.

The read and write strobes to external RAM are activated only during the execution of a MOVX instruction. Normally these signals are inactive, and in fact if they're not going to be used at all, their pins are available as extra I/O lines. More about that later.

MNEMONIC	OPERATION	EXECUTION TIME (µs)
MOVC A, @A + DPTR	Read Pgm Memory at (A + DPTR)	2
MOVC A, @A + PC	Read Pgm Memory at (A + PC)	2

Table 6 : The MHS C51 Lookup Table Read Instructions.

Lookup Tables

Table 6 shows the two instructions that are available for reading lookup tables in Program Memory. Since these instructions access only Program Memory, the lookup tables can be read, not updated. The mnemonic is MOVC for "move constant".

If the table access is to external Program Memory, then the read strobe is $\overline{PSEN}$.

The first MOVC instruction in Table 6 can accomodate a table of up to 256 entries, numbered 0 through 255. The number of the desired entry is loaded into the Accumulator, and the Data Pointer is set up to point to beginning of the table. Then

MOVC A, @A + DPTR

copies the desired table entry into the Accumulator.

The other MOVC instruction works the same way, except the Program Counter (PC) is used as the table base, and the table is accesses through a subroutine. First the number of the desired entry is loaded into the Accumulator, and the subroutine is called :

```
MOV     A, ENTRY_NUMBER
CALL    TABLE
```

The subroutine "TABLE" would look like this :

```
TABLE : MOVC A, @A + PC
        RET
```

The table itself immediately follows the RET (return) instruction in Program Memory. This type of table can have up to 255 entries, numbered 1 through 255. Number 0 can not be used, because at the time the MOVC instruction is executed, the PC contains the address of the RET instruction. An entry numbered 0 would be the RET opcode itself.

BOOLEAN INSTRUCTIONS

MHS C51 devices contain a complete Boolean (single-bit) processor. The internal RAM contains 128 addressable bits, and the SFR space can support up to 128 other addressable bits. All of the port lines are bit-addressable, and each one can be treated as a separate single-bit port. The instructions that access these bits are not just conditional branches, but a complete menu of move, set, clear, complement, OR and AND instructions. These kinds of bit operations are not easily obtained in other architectures with any amount of byte-oriented software.

The instruction set for the Boolean processor is shown in Table 7. All bit accesses are by direct addressing. Bit addresses 00H through 7FH are in the Lower 128, and bit addresses 80H through FFH are in SFR space.

MNEMONIC	OPERATION	EXECUTION TIME (µs)
ANL C,bit	C = C AND bit	2
ANL C,/bit	C = C AND (NOT bit)	2
ORL C,bit	C = C OR bit	2
ORL C,/bit	C = C OR (NOT bit)	2
MOV C,bit	C = bit	1
MOV bit,C	bit = C	2
CLR C	C = 0	1
CLR bit	bit = 0	1
SETB C	C = 1	1
SETB bit	bit = 1	1
CPL C	C = NOT C	1
CPL bit	bit = NOT bit	1
JC rel	Jump if C = 1	2
JNC rel	Jump if C = 0	2
JB bit,rel	Jump if bit = 1	2
JNB bit,rel	Jump if bit = 0	2
JBC bit,rel	Jump if bit = 1 ; CLR bit	2

Table 7 : A list of the MHS C51 Boolean Instructions.

Note how easily an internal flag can be moved to a port pin :

```
MOV     C, FLAG
MOV     P1.0, C
```

In this example, FLAG is the name of any addressable bit in the lower 128 or SFR space. An I/O line (the LSB of Port 1, in the case) is set or cleared depending on whether the flag bit is 1 or 0.

The Carry bit in the PSW is used as the single-bit Accumulator of the Boolean processor. Bit instructions that refer to the Carry bit as C assemble as Carry-specific instructions (CLR C, etc). The Carry bit also has a direct address, since it resides in the PSW register, which is bit-addressable.

Note that the Boolean instruction set includes ANL and ORL operations, but not the XRL (Exclusive OR) operation. An XRL operation is simple to implement in software. Suppose, for example, it is required to form the Exclusive OR of two bits :

C bit1 XRL bit2

The software to do that could be as follows :

```
MOV     C, bit1
JNB     bit2, OVER
CPL     C
```

OVER : (continue)

First, bit 1 is moved to the Carry. If bit 2 = 0, then C now contains the correct result. That is, bit 1 XRL bit2 = bit1 if bit2 = 0. On the other hand, if bit2 = 1 C now contains the complement of the correct result. It need only be inverted (CPL C) to complete the operation.

This code uses the JNB instruction, one of a series of bit-test instructions which execute a jump if the addressed bit

is set (JC, JB, JBC) or if the addressed bit is not set (JNC, JNB). In the above case, bit2 is being tested, and if bit2 = 0 the CPL C instruction is jumped over.

JBC executes the jump if the addresed bit is set, and also clears the bit. Thus a flag can be tested and cleared in one operation.

All the PSW bits are directly addressable, so the Parity bit, or the general purpose flags, for example, are also available to the bit-test instructions.

Relative offset

The destination address for these jumps is specified to the assembler by a label or by an actual address in Program

MNEMONIC	OPERATION	EXECUTION TIME (µs)
JMP addr	Jump to addr	2
JMP @A + DPTR	Jump to A + DPTR	2
CALL addr	Call subroutine at addr	2
RET	Return from subroutine	2
RETI	Return from interrupt	2
NOP	No operation	1

Table 8 : Unconditional Jumps in MHS C51.

Memory. However, the destination address assembles to a relative offset byte. This is a signed (two's complement) offset byte which is added to the PC in two's complement arithmetic if the jump is executed.

The range of the jump is therefore - 128 to + 127 Program Memory bytes relative to the first byte following the instruction.

JUMP INSTRUCTIONS

Table 8 shows the list of unconditional jumps.

The table lists a single "JMP addr" instruction, but in fact there are three - SJMP, LJMP, AJMP - which differ in the format of the destination address. JMP is a generic mnemonic which can be used if the programmer does not care which way the jump is encoded.

The SJMP instruction encodes the destination address as relative offset, as described above. The instruction is 2 bytes long, consisting of the opcode and the relative offset byte. The jump distance is limited to range of - 128 to

In all cases the programmer specifies the destination address to the assembler in the same way : as a label or as a 16-bit constant. the assembler will put the destination address into the correct format for the given instruction. If the format required by the instruction will not support the distance to the specified destination address, a "Destination out of range" message is written, into the list file.

The JMP @ A + DPTR instruction supports case jumps. The destination address is computed at execution time as the sum of the 16-bit DPTR register and the Accumulator. Typically, DPTR is set up with the address of a jump table, and the Accumulator is given an index to the table. In a 5-way branch, for example, an integer 0 through 4 is loaded into the Accumulator.

The code to be executed might be as follows :

```
MOV     DPTR, # JUMP_TABLE
MOV     A, INDEX_NUMBER
RL      A
JMP     @ A + DPTR
```

The RLA instruction converts the index number (0 through 4) to an even number on the range 0 through 8, because each entry in the jump table is 2 bytes long :

```
JUMP_TABLE :
        AJMP CASE_0
        AJMP CASE_1
        AJMP CASE_2
        AJMP CASE_3
        AJMP CASE_4
```

Table 8 shows a single "CALLaddr" instruction, but there are two of them - LCALL and ACALL - which differ in the format in which the subroutine address is given to the CPU. CALL is a generic mnemonic which can be used if the programmer does not care which way the address is encoded.

The LCALL instruction uses the 16-bit address format, and the subroutine can be anywhere in the 64K Program Memory space. The ACALL instruction uses the 11-bit format, and the subroutine must be in the same 2K block as the instructon following the ACALL.

In any case the programmer specifies the subroutine address to the assembler in the same way : as a label or as a 16-bit constant. The assembler will put the address into the correct format for the given instructions.

Subroutines should end a RET instruction, which returns execution following the CALL.

RETI is used to return from an interrupt service routine. The only difference between RET and RETI is that RETI tells the interrupt control system that the interrupt in progress is done. If there is no interrupt in progress at the time RETI is executed, then the RETI is functionnally identical to RET.

Table 9 shows the list of conditional jumps available to the MHS C51 user. All of these jumps specify the destination address by the relative offset method, and so are limited to a jump distance of - 128 to + 127 bytes from the instruction following the conditional jump instruction. Important to note, however, the user specifies to the assembler the actual destination address the same way as the other jumps : as a label or a 16-bit constant.

MNEMONIC	OPERATION	ADDRESSING MODES				EXECUTION TIME (µs)
		DIR	IND	REG	IMM	
JZ rel	Jump if A = 0	Accumulator only				2
JNZ rel	Jump if A ≠ 0	Accumulator only				2
DJNZ <byte>,rel	Decrement and jump if not zero	X		X		2
CJNE A,<byte>,rel	Jump if A = <byte>	X			X	2
CJNE <byte>,#data,rel	Jump if <byte> = #data		X	X		2

Table 9 : Conditional Jumps in MHS C51 Devices.

There is no Zero bit in the PSW. The JZ and JNZ instructions test the Accumulator data for that condition.

The DJNZ instruction (Decrement and Jump if Not Zero) is for loop control. To execute a loop N times, load a counter byte with N and terminate the loop with DJNZ to the beginning of the loop, as shown below for N = 10 :

```
        MOV     COUNTER, # 10
LOOP :  (begin loop)
          *
          *
          *
        (end loop)
          DJNZ    COUNTER, LOOP
        (continue)
```

The CJNE instruction (Compare and Jump if Not Equal) can also be used for loop control as in Figure 12. Two bytes are specified in the operand field of the instruction. The jump is executed only if the two bytes are not equal. In the example of Figure 12, the two bytes were the data in R1 and the constant 2AH. The initial data in R1 was 2EH. Every time the loop was executed, R1 was decremented, and the looping was to continue until the R1 data reached 2AH.

Another application of this instruction is in "greater than, less than" comparisons. The two bytes in the operand field are taken as unsigned integers. If the first is less than the second, then the Carry bit is set (1). If the first is greater than or equal to the second, then the Carry bit is cleared.

CPU TIMING

All MHS C51 microcontrollers have an on-chip oscillator which can be used if desired as the clock source for the CPU. To use the on-chip oscillator, connect a crystal or ceramic resonator between the XTAL1 and XTAL2 pins of the microcontroller, and capacitors to ground as shown in Figure 13.

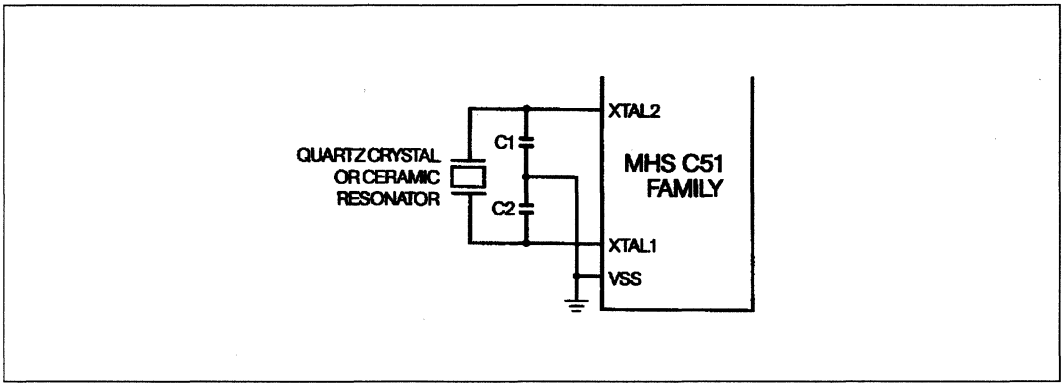


Figure 13 : Using the On-Chip Oscillator.

3

Examples of how to drive the clock with an external oscillator are shown in Figure 14. In the MHS C51 devices the signal at the XTAL1 pin drives the internal clock generator. If only one pin is going to be driven with the external oscillator signal, make sure it is the right pin.

The internal clock generator defines the sequence of states that make up the MHS C51 machine cycle.

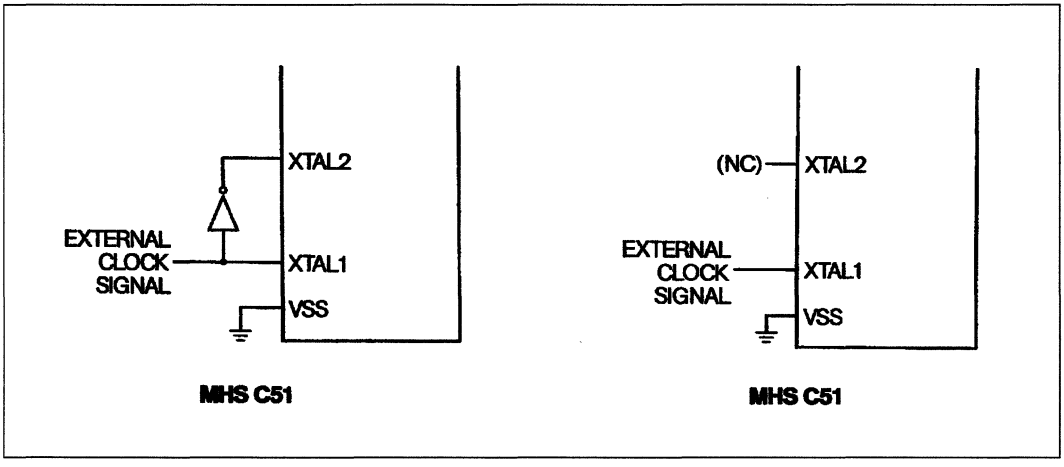


Figure 14 : Using an External Clock.

MACHINE CYCLES

A machine cycle consists of a sequence of 6 states, numbered S1 through S6. Each state time lasts for two oscillator periods. Thus a machine cycle takes 12 oscillator periods or 1 µs if the oscillator frequency is 12 MHz.

Each state is divided into a Phase 1 half and a Phase 2 half. Figure 15 shows the fetch/execute sequences in states and phases for various kinds of instructions. Normally two program fetches are generated during each machine cycle, even if the instruction being executed doesn't require it. If the instruction being executed doesn't need more code bytes, the CPU simply ignores the extra fetch, and the Program Counter is not incremented.

Execution of a one-cycle instruction (Figure 15A and B) begins during State 1 of the machine cycle, when the opcode is latched into the Instruction Register. A second fetch occurs during S4 of the same machine cycle. Execution is completed at the end of State 6 of this machine cycle.

The MOVX instructions take two machine cycles to execute. No program fetch is generated during the second cycle of a MOVX instruction. This is the only time program fetches are skipped. The fetch/execute sequence for MOVX instructions is shown in Figure 15 (D).

The fetch/execute sequences are the same whether the Program Memory is internal or external to the chip. Execution times do not depend on whether the Program Memory is internal or external.

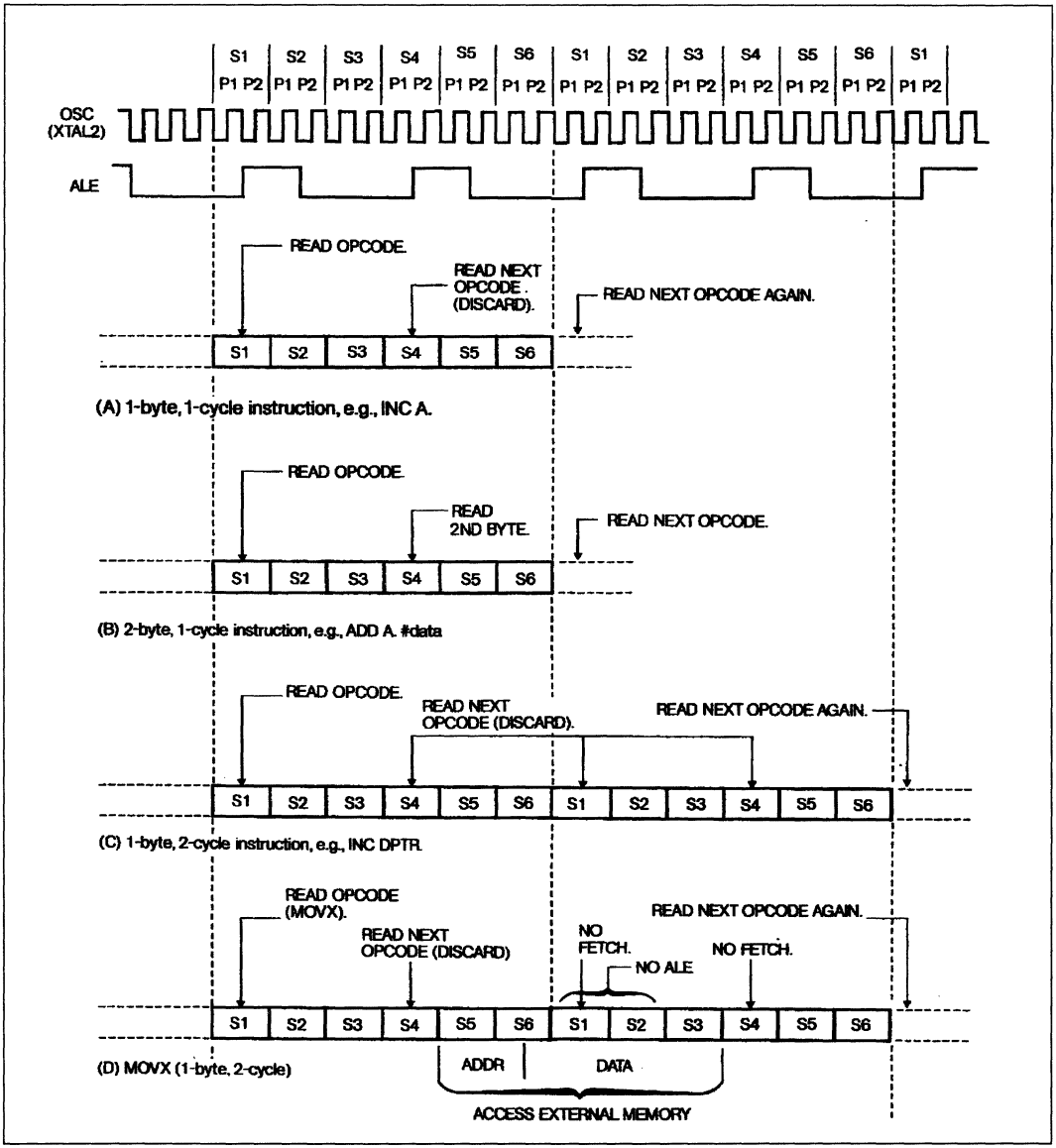


Figure 15 : State Sequences in MHS C51.

Figure 16 shows the signals and timing involved in program fetches when the Program Memory is external. If Program Memory is external, then, the Program Memory read strobe PSEN is normally activated twice per machine cycle, as shown in Figure 16 (A).

If an access to external Data Memory occurs, as shown in Figure 16 (B), two $\overline{\text{PSEN}}$s are skipped, because the address and data bus are being used for the Data Memory access.

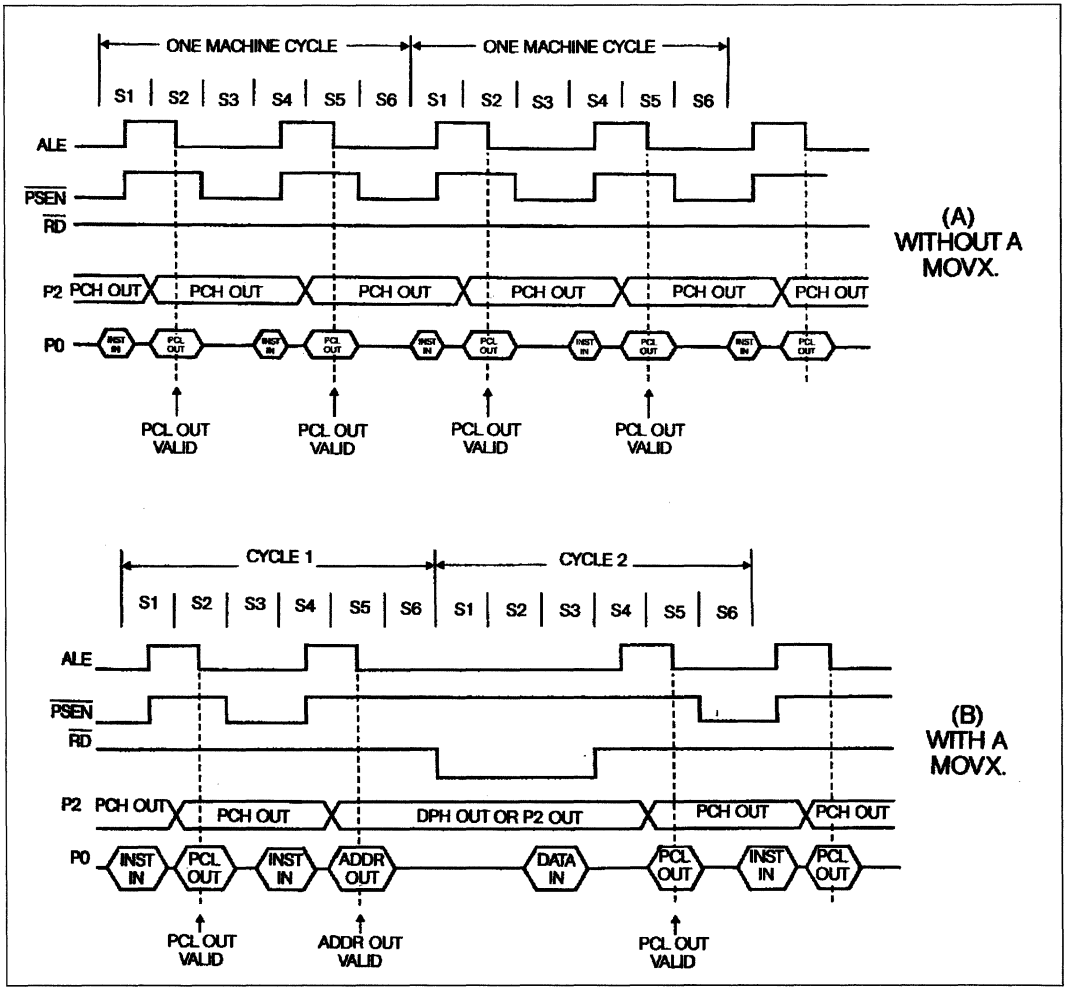


Figure 16 : Bus Cycles in MHS C51 Devices Executing from External Program Memory.

Note that a Data Memory bus cycle takes twice as much time as a Program Memory bus cycle. Figure 16 shows the relative timing of the addresses being emitted at ports 0 and 2, and of ALE and PSEN. ALE is used to latch the low address byte from P0 into the address latch.

When the CPU is executing from internal Program Memory, PSEN is not activated, and program addresses are not emitted. However, ALE continues to be activated twice per machine cycle and so is available as a clock output signal. Note, however, that one ALE is skipped during the execution of the MOVX instruction.

INTERRUPT STRUCTURE

The 80C51 and his ROMless version provide 5 interrupt sources : 2 external interrupts, 2 timer interrupts, and the serial port interrupt. the 80C52, 83C154 and 83C154D and their ROMless version provide these 5 plus a sixth interrupt that is associated with the third timer/counter which is present in those devices.

What follows is an overview of the interrupt structure for these devices. More detailed information for specific members of the MHS C51 family is provided in the chapters of this handbook that describe the specific devices.

Interrupt Enables

Each of the interrupt source can be individually enabled or disabled by setting or clearing a bit in the SFR named IE (Interrupt Enable). This register also contains a global disable bit, which can be cleared to disable all interrupts at once. Figure 17 shows the IE register for the 80C52 and 83C154 or the 83C154D.

(MSB)							(LSB)
EA	X	ET2	ES	ET1	EX1	ET0	EX0

Symbol	Position	Function
EA	IE.7	disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1,each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
–	IE.6	reserved
ET2	IE.5	enables or disables the Timer 2 overflow or capture interrupt. If ET2 = 0, the Timer 2 interrupt is disabled.
ES	IE.4	enables or disables the Serial Port interrupt. If ES = 0, the Serial Port interrupt is disabled.
ET1	IE.3	enables or disables the Timer 1 Overflow interrupt. If ET1 = 0, the Timer 1 interrupt is disabled.
EX1	IE.2	enables or disables External Interrupt 1. If EX1 = 0, External Interrupt 1 is disabled.
ET0	IE.1	enables or disables the Timer 0 Overflow interrupt. If ET0 = 0, the Timer 0 interrupt is disabled.
EX0	IE.0	enables or disables External Interrupt 0. If EX0 = 0, External Interrupt 0 is disabled.

Figure 17 : IE (Interrupt Enable) Register in the 80C52, 83C154 and 83C154D.

(MSB)							(LSB)
PCT	X	PT2	PS	PT1	PX1	PT0	PX0

Symbol	Position	Function
PCT	IP.7	83C154/C154D only. Priority interrupt circuit control bit. The priority register contents are valid and priority assigned interrupts can be processed when this bit is "0". When the bit is "1", the priority interrupt circuit is stopped, and interrupts can only be controlled by the interrupt enable register (IE).
–	IP.6	reserved
PT2	IP.5	defines the Timer 2 interrupt priority level. PT2 = 1 programs it to the higher priority level.
PS	IP.4	defines the Serial Port interrupt priority level. PS = 1 programs it to the higher priority level.
PT1	IP.3	defines the Timer 1 interrupt priority level. PT1 = 1 programs it to the higher priority level.
PX1	IP.2	defines the External Interrupt 1 priority level. PX1 = 1 programs it to the higher priority level.
PT0	IP.1	defines the Timer 0 interrupt priority level. PT0 = 1 programs it to the higher priority level.
PX0	IP.0	defines the External Interrupt 0 priority level. PX0 = 1 programs it to the higher priority level.

Figure 18 : IP (Interrupt Priority) Register in the 80C52, 83C154 and 83C154D.

Interrupt priorities

Each interrupt source can also be individually programmed to one of two priority level by setting or clearing a bit in the SFR named IP (Interrupt Priority). Figure 18 shows the IP register in the 80C52, 83C154 and 83C154D.

A low-priority interrupt can be interrupted by a high-priority interrupt, but not by another low-priority interrupt.

A high-priority interrupt can't be interrupted by any other interrupt source.

If two interrupt requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If interrupt requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence.

Figure 19 shows, for the 80C52, 83C154 and 83C154D, how the IE and IP registers and the polling sequence work to determine which interrupt will be serviced.

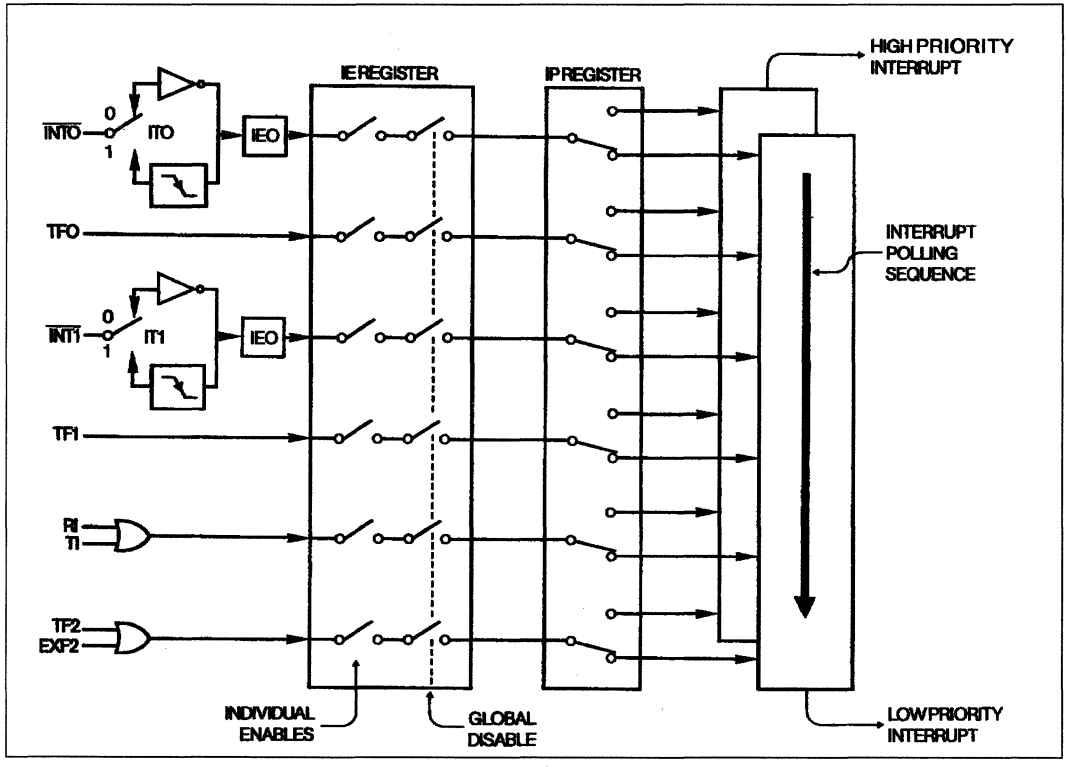


Figure 19 : 80C52, 83C154 and 83C154D Interrupt Control System.

In operation, all the interrupt flags are latched into the interrupts control system during State 5 of every machine cycle. The samples are polled during the following machine cycle. If the flag for an enabled interrupt is found to be set (1), the interrupt system generates an LCALL to the appropriate location in Program Memory, unless some other condition blocks the interrupt. Several conditions can block an interrupt, among them that an interrupt of equal or higher priority level is already in progress.

The hardware-generated LCALL causes the contents of the Program Counter to be pushed onto the stack, and reloads the PC with the beginning address of the service routine. As previously noted (Figure 3), the service routine for each interrupt begins at a fixed location.

Only the Program Counter is automatically pushed onto that stack, not the PSW or any other register. Having only the PC be automatically saved allows the programmer to decide how much time to spend saving which other registers. This enhances the interrupt response time, albeit at the expense of increasing the programmer's burden of responsability. As a result, many interrupt functions that are typical in control applications-toggling a port pin, for example,

or reloading a timer, or unloading a serial buffer can often be completed in less time than it takes other architectures to commence them.

Simulating a third priority level in software

Some applications require more than the two priority levels that are provided by on-chip hardware in MHS C51 devices. In these cases, relatively simple software can be written to produce the same effect as a third priority level. First, interrupts that are to have higher priority than 1 are assigned to priority 1 in the IP (Interrupt Priority) register. The service routines for priority 1 interrupts that are supposed to be interruptible by "priority 2" interrupts are written to include the following code :

```
        PUSH    IE
        MOV     IE, # MASK
        CALL    LABEL
              **********
        (execute service routine)
                ********
        POP     IE
        RET
LABEL : RETI
```

As soon as any priority 1 interrupt is acknowledged, the IE (Interrupt Enable) register is redefined so as to disable all but "priority 2" interrupts. Then, a CALL to LABEL executes the RETI instruction, which clears the priority 1 interrupt-in-progress flip-flop. At this point any priority 1 interrupt that is enabled can be serviced, but only "priority 2" interrupts are enabled.

POPping IE restores the original enable byte. Then a normal RET (rather than another RETI) is used to terminate the service routine. The additional software adds 10 µs (at 12 MHz) to priority 1 interrupts.

HARDWARE DESCRIPTION OF THE 80C51, 80C52 and 83C154/C154D

4

MHS

HARDWARE DESCRIPTION OF THE 80C51, 80C52, 83C154/C154D

INTRODUCTION

This chapter presents a comprehensive description of the on-chip hardware features of the MHS C51 microcontrollers. Included in this description are

- The port drivers and how they function both as ports and, for Ports 0 and 2, in bus operations
- The Timer/Counters
- The serial Interface
- The Interrupt System
- Reset
- The Reduced Power Modes

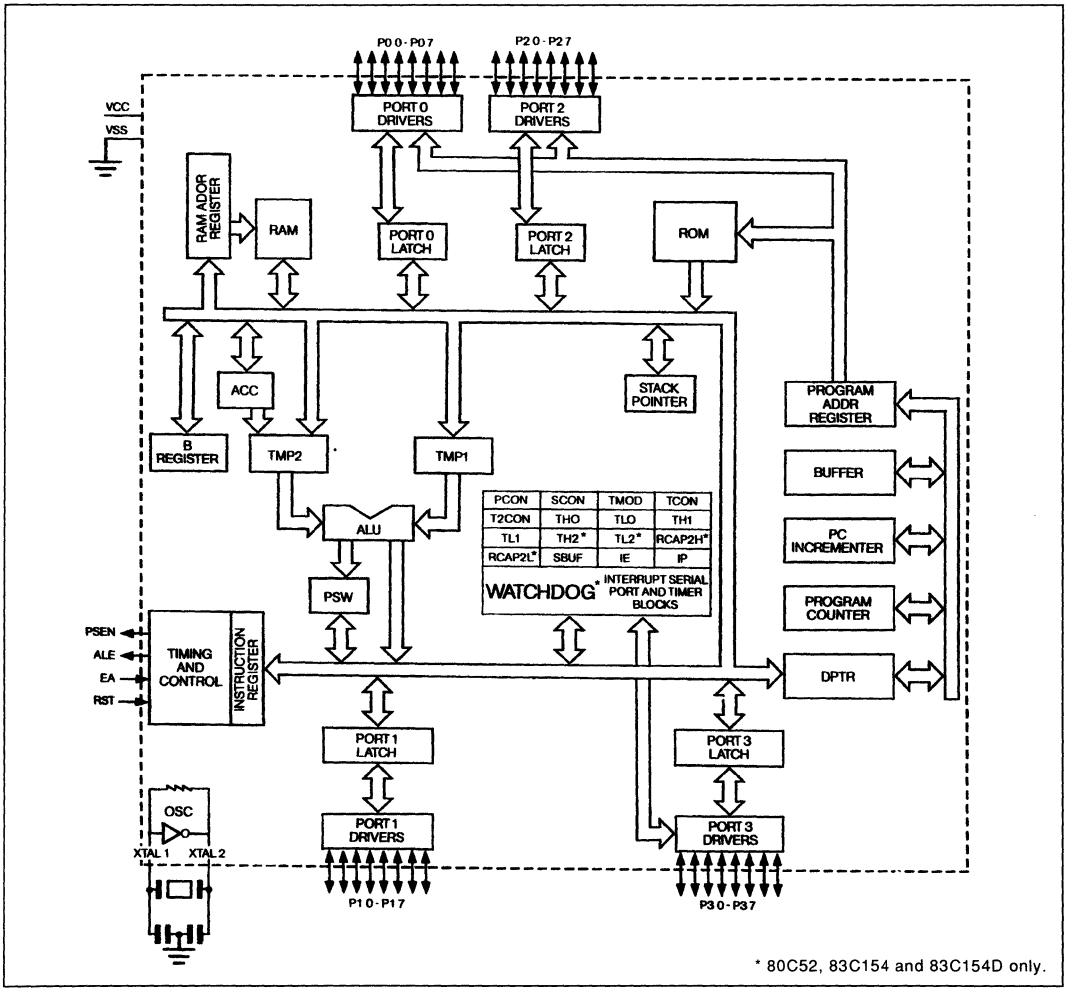


Figure 1 : MHS C51 Architectural Block Diagram.

DEVICE NAME	ROMLESS VERSION	ROM BYTES	RAM BYTES	16-BIT TIMERS	PROCESS TYPE
80C51	80C31	4K	128	2	CMOS
80C52	80C32	8K	256	3	CMOS
83C154	80C154	16K	256	3*	CMOS
83C154D	80C154	32 K	256	3*	CMOS

* included watch dog and Timer 32 bits.

Table 1 : The MHS C51 Family of Microcontrollers.

The devices under consideration are listed in Table 1. As it becomes unwieldy to be constantly referring to each of these devices by their individual names, we will adopt a convention of refering to them generically as 80C51s, 80C52s and 83C154s, unless a specific member of the group is being refered to, in which case it will be specifically named. The 80C51s include the 80C51 and 80C31. The 80C52s are the 80C52 and 80C32. The 83C154s are the 83C154, the 80C154 and the 83C154D.

Figure 1 shows a functional block diagram of the 80C51s, 80C52s and 83C154s.

Special Function Registers

A map of the on-chip memory area called SFR (Special Function Register) space is shown in Figure 2. SFRs marked by parentheses are resident in the 80C52s and 83C154s but not in the 80C51s. IOCON marked by a star is only resident in the 83C154s.

Note that not all of the addresses are occupied. Unoccupied addresses are not implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have no effect.

	8 Bytes								
F8	*IOCON								FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8	(T2CON)		(RCAP2L)	(RCAP2H)	(TL2)	(TH2)			CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87

* 83C154s only.

Figure 2 : SFR Map. (...) Indicates Resident in 80C52s and 83C154s, not in 80C51s.

User software should not write 1 s to these unimplemented locations, since they may be used in future MHS C51 products to invoke new features. In that case the reset or inactive values of the new bits will always be 0, and their active values will be 1.

The functions of the SFRs are outlined below.

Accumulator

ACC is the Accumulator register. The mnemonics for Accumalator-Specific instructions, however, refer to the Accumulator simply as A.

B Register

The B register is used during multiply and divide operations. For other instructions it can be treated as another scratch pad register.

Program Status Word

The PSW register contains program status information as detailed in Figure 3.

(MSB)							(LSB)
CY	AC	F0	RS1	RS0	OV	F1	P

Symbol	Position	Function
CY	PSW.7	Carry flag
AC	PSW.6	Auxiliary Carry flag. (For BCD operations).
F0	PSW.5	Flag 0 (Available to the user for general purposes).
RS1	PSW.4	Register bank select
RS0	PSW.3	Control bits 1 & 0. Set/cleared by software to determine working register bank (see Note).
OV	PSW.2	Overflow flag.
F1	PSW.1	User definable flag.
P	PSW.0	Parity flag. Set/Cleared by hardware each instruction cycle to indicate an odd/even number of "one" bits in the Accumulator, i.e., even parity.

Note :

The contents of (RS1, RS0) enable the working register banks as follows :

(0.0)–Bank 0	(00H–07H)
(0.1)–Bank 1	(08H–0FH)
(1.0)–Bank 2	(10H–17H)
(1.1)–Bank 3	(18H–1FH)

Figure 3 : Program Status Word Register.

Stack Pointer

The Stack Pointer Register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the Stack Pointer is initialized to 07H after a reset. This causes the stack to begin at location 08H.

Data Pointer

The Data Pointer (DPTR) consists of a high byte (DPH) and low byte (DPL). Its intented function is to hold a 16-bit address. It may be manipulated as a 16-bit register or as two independent 8-bit registers.

Ports 0 to 3

P0, P1, P2 and P3 are the SFR latches of Ports 0, 1, 2 and 3, respectively.

Serial Data Buffer

The Serial Data Buffer is actually two separate registers, a transmit buffer and a receive buffer register. When data is moved to SBUF, it goes to the transmit buffer where it is held for serial transmission. (Moving a byte to SBUF is what initiates the transmission). When data is moved from SBUF, it comes from the receive buffer.

Timer Registers

Register pairs (TH0, TL0), (TH1, TL1), and (TH2, TL2) are the 16-bit counting registers for Timer/Counters 0, 1, and 2, respectively.

Capture Registers

The register pair (RCAP2H, RCAP2L) are the Capture registers for the Timer 2 "Capture Mode". In this mode, in response to a transition at the 8052s' and 83C154s' T2EX pin, TH2 and TL2 are copied into RCAP2H and RCAP2L. Timer 2 also has a 16-bit auto-reload mode, and RCAP2H and RCAP2L hold the reload value for this mode. More details about Timer 2's features are in a later section.

Control Registers

Special Function registers IP, IE, TMOD, TCON, T2CON, SCON, IOCON and PCON contain control and status bits for the interrupt system, the Timer/Counters, and the serial port. they are described in later sections.

PORT STRUCTURES AND OPERATION

All four ports in the MHS C51 family are bidirectional. Each consists of a latch (Special Function Registers P0 through P3), an output driver and an input buffer.

The output drivers of Ports 0 and 2, and the input buffers of Port 0, are used in accesses to external memory. In this application, Port 0 outputs the low byte of the external memory address, time-multiplexed with the byte being written or read. Port 2 outputs the high byte of the external memory address when the address is 16-bits wide. Otherwise the Port 2 pins continue to emit the P2 SFR content.

All the Port 3 pins, and (in the 80C52, 83C154 and 83C154D) two Port 1 pins are multifunctional. They are not only port pins, but also serve the functions of various special features as listed below.

Port Pin	Alternate Function
*P1.0	T2 (Timer/Counter 2 external input)
*P1.1	T2EX (Timer/Counter 2 capture/Reload trigger)
P3.0	RDX (serial input port)
P3.1	TDX (serial output port)
P3.2	INT0 (external interrupt)
P3.3	INT1 (external interrupt)
P3.4	T0 (Timer/Counter 0 external input)
P3.5	T1 (Timer/Counter 1 external input)
P3.6	WR (external Data Memory write strobe)
P3.7	RD (external Data Memory read strobe)

*P1.0 and P1.1 serve the alternate functions only on the 80C52s and 83C154s.

The alternate functions can only be activated if the corresponding bit latch in the port SFR contains a 1. Otherwise the port pin is stuck at 0.

I/O Configurations

Figure 4 shows a functional diagram of a typical bit latch and I/O buffer in each of the four ports. The bit latch (one bit in the port's SFR) is represented as a type D flip-flop, which will clock in a value from the internal bus in response to a "write to latch" signal from the CPU. The Q output of the flip-flop is placed on the internal bus in response to a "read latch" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a "read pin" signal from the CPU. Some instructions that read a port activate the "read latch" signal, and others activate the "read pin" signal. More about that later.

As shown in Figure 4, the output drivers of Ports 0 and 2 are switchable to an internal ADDR and ADDR/DATA bus by an internal CONTROL signal for use in external memory accesses. During external memory accesses, the P2 SFR remains unchanged, but the P0 SFR gets 1 s written to it.

Also shown in Figure 4, is that if a P3 bit latch contains a 1, then the output level is controlled by the signal labeled "alternate output function". The actual P3.X pin level is always available to the pin's alternate input function, if any.

Ports 1, 2, and 3 have internal pullups. Port 0 has open drain outputs. Each I/O line can be independently used as an input or an output. (Ports 0 and 2 may not be used as general purpose I/O when being used as the ADDR/DATA BUS). To be used as an input, the port bit latch must contain a 1, which turns off the output driver FET. Then, for Ports 1, 2, and 3, the pin is pulled high by the internal pullup, but can be pulled low by an external source.

Port 0 differs in not having internal pullups. The pullup FET in the P0 output driver (see Figure 4) is used only when the Port is emitting 1 s during external memory accesses. Otherwise the pullup FET is off. Consequently P0 lines that are being uses as output port lines are open drain. Writting a 1 to be bit latch leaves both output FETs off, so the pin floats. In that condition it can be used a high-impedance input.

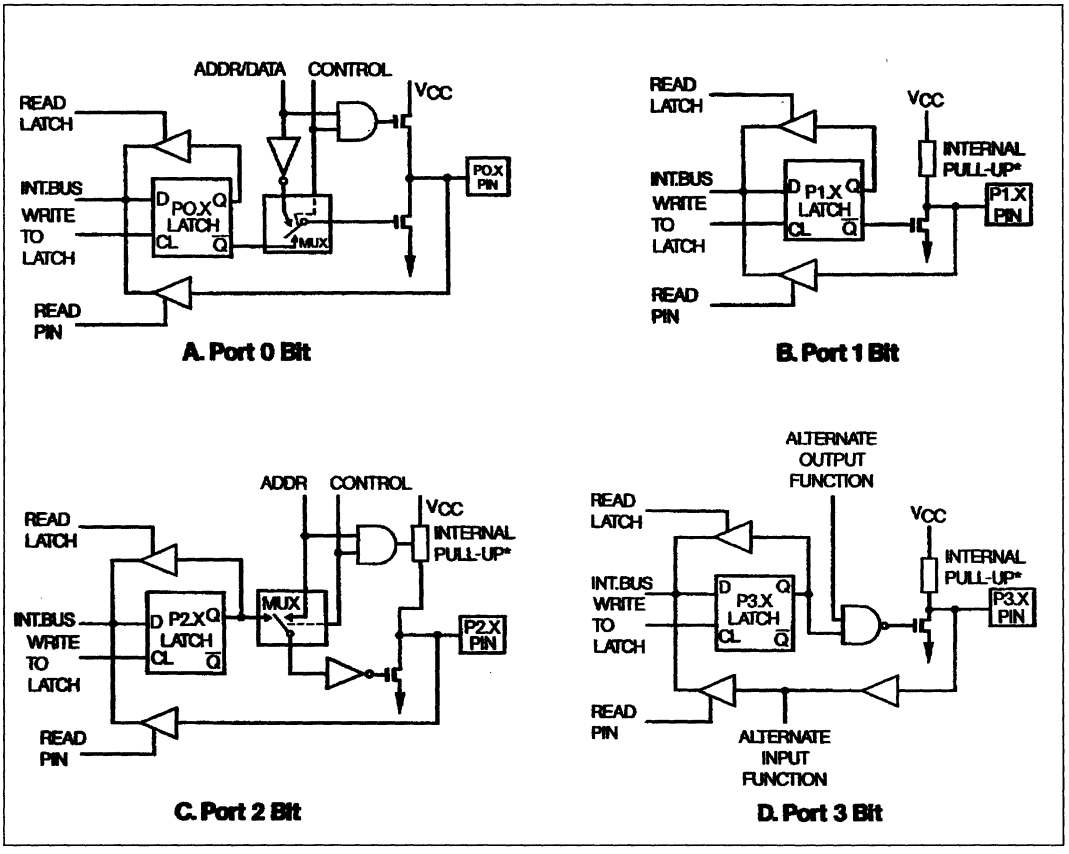


Figure 4 : MHS C51 FAMILY Port Bit Latches and I/O Buffers.
* See Figure 5 for details of the internal pullup.

Because Ports 1, 2 and 3 have fixed internal pullups they are sometimes called "quasi-bidirectional" ports. When configured as inputs they pull high and will source current (IIL, in the data sheets) when externally pulled low. Port 0, on the other hand, is considered "true" bidirectional, because when configurated as an input it floats.

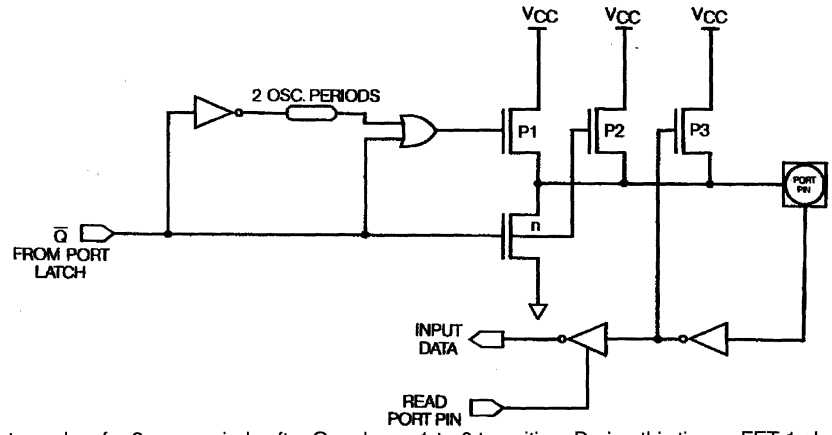
All the port latches in the MHS C51 FAMILY have 1 s written to them by the reset function. If a 0 is subsequently written to a port latch, it can be reconfigured as an input by writting a 1 to it.

Writing to a Port

In the execution of an instruction that changes the value in a port latch, the new value arrives at the latch during S6P2 of the final cycle of the instruction. However, port latches are in fact sampled by their output buffers only during Phase 1 of any clock period. (During Phase 2 the output buffer holds the value it saw during the previous Phase 1.) Consequently, the new value in the port latch won't actually appear at the output pin until the next Phase 1, which will be at S1P1 of the next machine cycle.

If the change requires a 0-to-1 transition in Port 1, 2, or 3, an additional pullup is turned on during S1P1 and S1P2 of the cycle in which the transition occurs. This is done to increase the transition speed. The extra pullup can source about 100 times the current that the normal pullup can. It should be noted that the internal pullups are field-effect transistors, not linear resistors. The pullup arrangements are shown in Figure 5.

In the MHS C51 family, the pullup consists of three pFETs. It should be noted that an n-channel FET (nFET) is turned on when a logical 1 is applied to its gate, and is turned off when a logical 0 is applied to its gate. A p-channel FET (pFET) is the opposite : it is on when its gate sees a 0, and off when its gate sees a 1.



pFET 1 is turned on for 2 osc. periods after Q makes a 1-to-0 transition. During this time, pFET 1 also turns on pFET 3 through the inverter to form a latch which holds the 1.pFET 2 is also on.

Figure 5 : Port 1 and Internal Pullup Configurations. Port 2 is Similar Except That It Holds The Strong Pullup On While Emitting 1 s That Are Address Bits. (See Text, "Accessing External Memory".)

4

pFET 1 in Figure 5 is the transistor that is turned on for 2 oscillator periods after a 0-to-1 transition in the port latch. While it's on, it turns on pFET3 (a weak pullup), through the inverter. This inverter and pFET from a latch which hold the 1.

Note that if the pin is emitting a 1, a negative glitch on the pin from some external source can turn off pFET3, causing the pin to go into a float state. pFET2 is a very weak pullup which is on whenever the nFET is off, in traditional CMOS style. It's only about 1/10 the strength of pFET3. Its function is to restore a 1 to the pin in the event the pin *had* a 1 and lost it to a glitch.

Port Loading and Interfacing

The output buffer of Ports 1, 2, and 3 can each drive 3LS TTL inputs. The pins can be driven by open-collector and open-drain outputs, but note that 0-to-1 transitions will not be fast. In the CMOS device, an input 0 turns off pullup pFET3, leaving only the very weak pullup pFET2 to drive the transition.

Port 0 output buffers can each drive 8 LS TTL inputs. They do, however, require external pullups to drive NMOS inputs, except when being used as the ADDRESS/DATA bus.

83C154 and 83C154D I/O Configurations

The structure and behaviour of the 83C154s' ports P1, P2 and P3 are indentical to those of the 80C52. Only the control block for the different pullups and pulldowns has been changed. The pullup resistance value can be programmed by means of the IOCON register.

There are three possible values :

- three states (P1, P2, P3 and N are OFF),
- high impedance (100 kΩ, P2 = ON),
- low impedance (10 kΩ, P3 = ON).

Figure 6 is a functional diagram of the PORT.

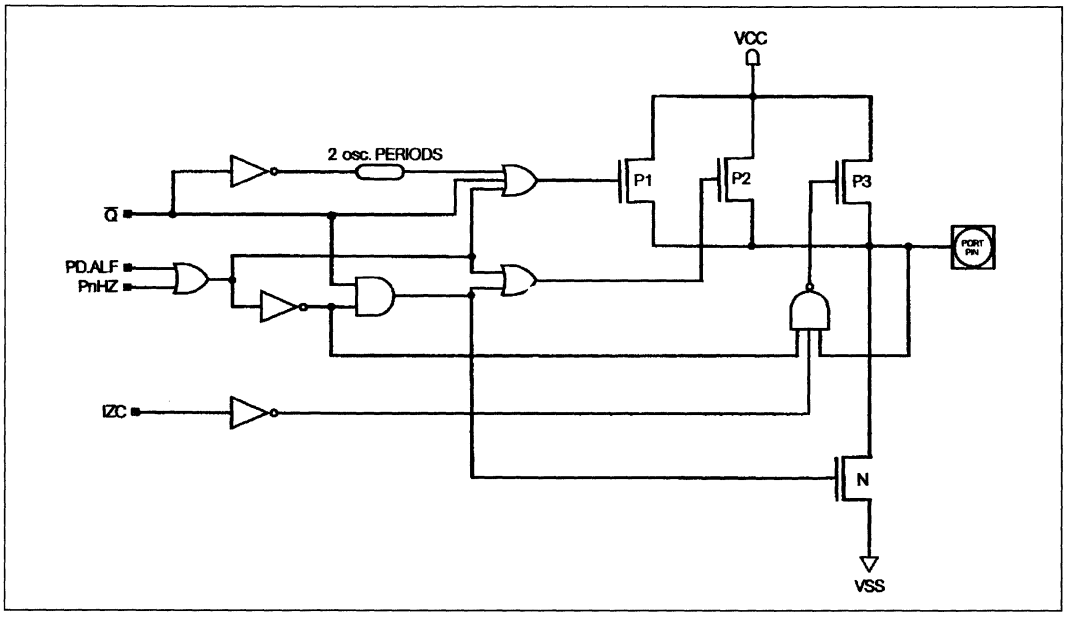


Figure 6 : PORTS 1 and 3 internal pullup configurations. PORT 2 is similar except that it holds the strong pullup on while emitting 1 s that are address bits.

Read-Modify-Write Feature

Somes instructions that read a port read the latch and others read the pin. Which ones do what ? The instructions that read the latch rather than the pin are the ones that read value, possibly change it, and then rewrite it to the latch. These are called "read-modify-write" instructions. The instructions listed below are read-modify-write instructions. When the destination operand is a port, or a port bit, these instructions read the latch rather than the pin :

ANL	(logical AND, e.g., ANL P1, A)
ORL	(logical OR, e.g., ORL P2, A)
XRL	(logical EX-OR, e.g., XRL P3, A)
JBC	(jump if bit = 1 and clear bit, e.g., JBC P1.1, LABEL)
CPL	(complement bit, e.g., CPL P3.0)
INC	(increment, e.g., INC P2)
DEC	(decrement, e.g., DEC P2)
DJNZ	(Decrement, and jump if not Zero, e.g., DJNZ P3, LABEL)
MOV PX.Y,C	(move carry bit to bit Y of Port X)
CLR PX.Y	(clear bit Y of Port X)
SET PX.Y	(set bit Y of Port X)

It is obvious that the last three instructions in this list are read-modify-write instructions, but they are. They read the port byte, all 8 bits, modify the addressed bit, then write the new byte back to the latch.

The reason that read-modify-write instructions are directed to the latch rather than the pin is to avoid a possible misinterpretation of the voltage level at the pin. For example, a port bit might be used to drive the base of a transistor. When a 1 is written to the bit, the transistor is turned on. If the CPU then reads the same port bit at the pin rather than the latch, it will read the base voltage of the transistor and interpret it as a 0. Reading the latch rather than the pin return the correct value of 1.

The first four bits of register IOCON (OF8H) must be used for programming the output pullup values. Figure 7 shows how IOCON must be programmed in order to obtain the required value.

(MSB)							(LSB)
WDT	T32	SERR	IZC	P3HZ	P2HZ	P1HZ	ALF

Symbol	Position	Function
ALF	IOCON.0	- Set to 1 and in Power Down mode PORTS 1, 2, and 3 are floating.
P1HZ	IOCON.1	- If P1HZ = 0 and IZC = 0, PORT P1 is at low impedance. - If P1HZ = 0 and IZC = 0, PORT P1 is at high impedance. - If P1HZ = 1, PORT P1 is floating.
P2HZ	IOCON.2	- If P2HZ = 0 and IZC = 0, PORT P2 is at low impedance. - If P2HZ = 0 and IZC = 0, PORT P2 is at high impedance. - If P2HZ = 1, PORT P2 is floating.
P3HZ	IOCON.3	- If P3HZ = 0 and IZC = 0, PORT P3 is at low impedance. - If P3HZ = 0 and IZC = 0, PORT P3 is at high impedance. - If P3HZ = 1, PORT P3 is floating.
IZC	IOCON.4	- In conjunction with PnHZ selects the output pullup value.

Figure 7 : IOCON register content.

Loading and Interfacing

- When PnHZ = 0 and IZC = 0, ports P1, P2 and P3 are identical to the 80C52's ports and each of them can load 3 LS TTL gates. Each Input/Output can be loaded by a collector or an open drain. However, it is important to note that transistors going from 0 to 1 are slower. The circuit in figure 8 shows an I/O of a port loaded by a 100 pf capacitor and a transistor T (open drain).

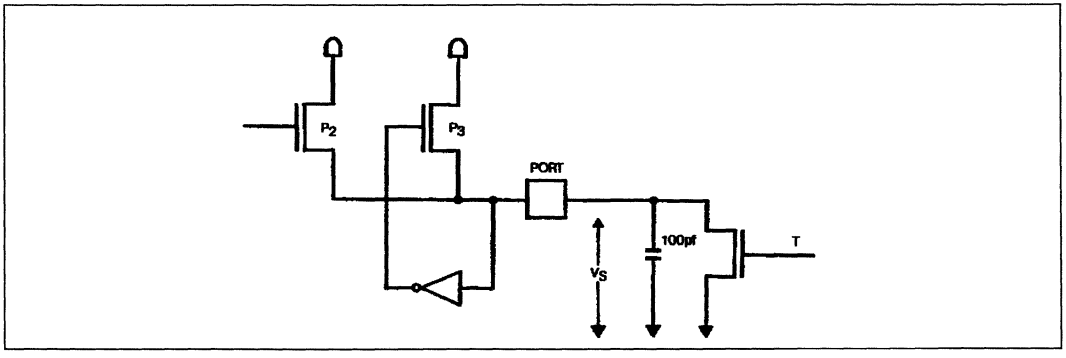


Figure 8 : A port loaded by an open drain transistor and a 100 pf capacitor.

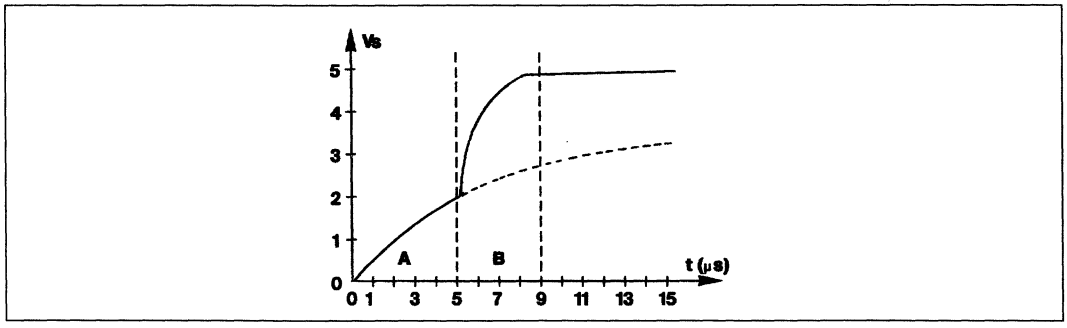


Figure 9 : Shows the behaviour of V_S when transistor T is blocked.
Influence of transistor P3 in area B.

In area A, Vs is less than ≈ 2 V. Transistor P3 is blocked and only transistor P2 is active during the transition (therefore, the time constant is 10 times greater than normal). In area B, Vs becomes greater than 2 V and transistors P2 and P3 are active in order to terminate the transition.

- When PnHZ = 0 and IZC = 1, only transistor P2 loads the ports Input/Output and one LS TTL load can be accepted by the port. Each I/O can be loaded by a collector or an open drain. However, as stated above, it is important to note that transistors going from 0 to 1 are slower. During the transition, only transistor P2 conducts (whatever the value of Vs) and the time taken for going from 0 to 2.4 V is ten times greater than normal.

ACCESSING EXTERNAL MEMORY

Accesses to external memory are of two types : accesses to external Program Memory and accesses to external Data Memory. Accesses to external Program Memory use signal PSEN (program store enable) as the read strobe. Accesses to external Data Memory use RD or WR (alternate functions of P3.7 and P3.6) to strobe the memory.

Fetches from external Program Memory always use a 16-bit address. Accesses to external Data Memory can use either a 16-bit address (MOVX @ DPTR) or an 8-bit address (MOVX @ RI).

Whenever a 16-bit address is used, the high byte of the address comes out on Port2, where it is held for the duration of the read or write cycle. Note that the Port2 drivers use the strong pullups during the entire time that they are emitting address bits that are 1 s. This is during the execution of a MOVX @ DPTR instruction. During this time the Port2 latch (the Special Function Register) does not have to contain 1 s, and the contents of the Port2 SFR are not modified. If the external memory cycle is not immediately followed by another external memory cycle, the undisturbed contents of the Port2 SFR will reappear in the next cycle.

If an 8-bit address is being used (MOVX @ RI), the contents of the Port2 SFR remain at the Port2 pins throughout the external memory cycle. This will facilitate paging.

In any case, the low byte of the address is time-multiplexed with data byte on Port0. The ADDR/DATA signal drives both FETs in the Port0 output buffers. Thus, in this application the Port0 pins are not open-drain outputs, and do to not require external pullups. Signal ALE (Address Latch Enable) should be used to capture the address byte into an external latch. The address byte is valid at the negative transition of ALE. Then, in a write cycle, the data byte to be written appears on Port0 just before WR is activated, and remains there until after WR is desactivated. In a read cycle, the incoming byte is accepted at Port0 just before the read strobe is desactivated.

During any access to external memory, the CPU writes 0FFH to the Port0 latch (the Special Function Register), thus obliterating whatever information the Port 0 SFR may have been holding.

External Program Memory is accessed under two conditions :

1) Whenever signal $\overline{EA}$ is active ; or
2) Whenever the program counter (PC), contains a number that is larger than 0FFFH (1FFFH for the 80C52, 3FFFH for the 83C154 or 7FFFh for the the 83C154D).

This requires that the ROMless versions have $\overline{EA}$ wired low to enable the lower 4k (8k for the 80C32, 16K for the 80C154 or 32 K for the 83C154D) program bytes to be fetched from external memory.

When the CPU is executing out of external Program Memory, all 8 bits of Port 2 are dedicated to an output function and may not be used for general purpose I/O. During external program fetches they output the high byte of the PC. During this time the Port 2 drivers use the strong pullups to emit PC bits that are 1 s.

TIMERS/COUNTERS

The 80C51 has two 16-bit Timer/Counter registers : Timer 0 and Timer 1. The 80C52, 83C154 and 83C154D have these two plus one more : Timer 2. All three can be configured to operate either as timers or event counters.

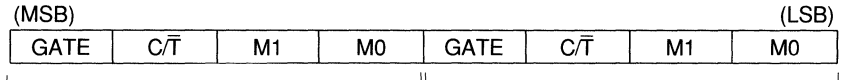
In the "Timer" function, the register is incremented every machine cycle. Thus, one can think of it as counting machine cycles. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the "Counter" function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T0, T1 or (in the 80C52/C154/C154D) T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since it takes 2 machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be help for at least one full machine cycle.

In addition to the "Timer" or "Counter" selection, Timer 0 and Timer 1 have four operating modes from which to select. Timer 2, in the 80C52/C154/C154D, has three modes of operation : "Capture", "Auto-Reload" and "baud rate generator".

Timer 0 and Timer 1

These Timer/Counters are present in the 80C51, the 80C52, the 83C154 and 83C154D. The "Timer" or "Counter" function is selected by control bits C/T in the Special Function Register TMOD (figure 10). These two Timer/Counters have four operating modes, which are selected by bit-pairs (M1, M0) in TMOD and three more for the 83C154/C154D which are selected by bit-pairs (WDT, T32) in IOCON. Modes 0, 1, and 2 are the same for both Timer/Counters. Mode 3 is different and mode 4, and 6 are reserved for 83C154/C154D only. The seven modes, 80C51, 80C52 and 83C154/C154D operating modes are described in the following text.



Timer 1 **Timer 0**

GATE Gating control when set. Timer/Counter "x" is enabled only while "INTx" pin is high and "TRx" control pin is set. When cleared Timer "x" is enabled whenever "TRx" control bit is set.

C/T Timer or Counter Selector cleared for Timer operation (input from internal system clock). Set for Counter operation (input from "Tx" input pin).

M1	M0	Operating Mode
0	0	13 bit Timer/Counter. Timer "TLx serves as a 5 bit prescaler
0	1	16-bit Timer/Counter "THx" and "TLx" are cascaded ; there is no prescaler
1	0	8-bit auto-reload Timer/Counter "THx" holds a value which is to be reloaded into "TLx" each time it overflows.
1	1	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only controlled by Timer 1 control bits.
1	1	(Timer 1) Timer/Counter 1 stopped.

Figure 10 : TMOD : Timer/Counter Mode Control Register.

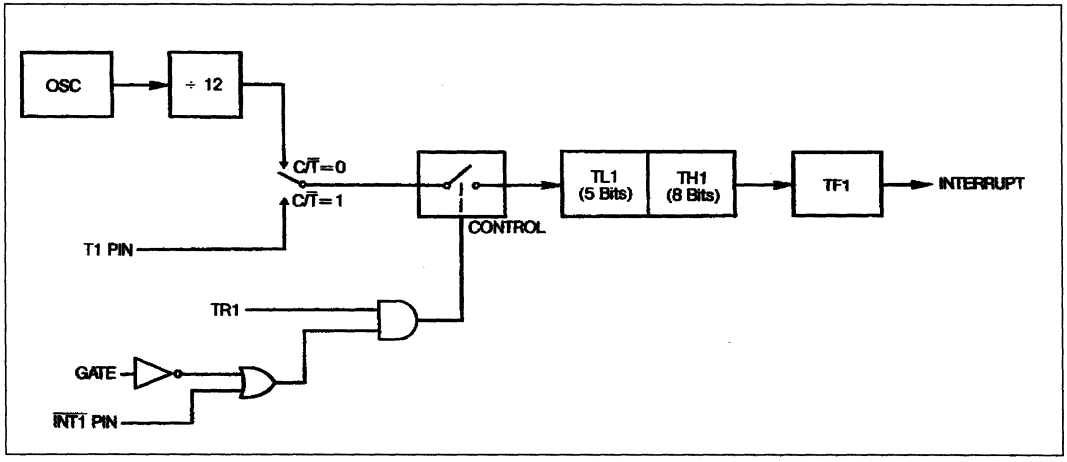


Figure 11 : Timer/Counter 1 Mode 0 : 13 Bit Counter.

(MSB)							(LSB)
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Symbol	Position	Function
TF1	TCON.7	Timer 1 overflow Flag. Set by hardware on Timer/ Counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR1	TCON.6	Timer 1 Run control bit. Set/cleared by software to turn Time/Counter on/off.
TF0	TCON.5	Timer 0 overflow Flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR0	TCON.4	Timer 0 Run control bit. Set/cleared by software to turn Timer/Counter on/off.
IE1	TCON.3	Interrupt 1 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT1	TCON.2	Interrupt 1 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
IE0	TCON.1	Interrupt 0 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT0	TCON.0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.

Figure 12 : TCON : Timer/Counter Control Register.

Mode 0

Putting either Timer into Mode 0 makes it look like an 8-bit Counter with a divide-by-32 prescaler. Figure 11 shows the Mode 0 operation as it applies to Timer 1.

In this mode, the Timer register is configured as a 13-Bit register. As the count rolls over from all 1 s to all 0s, it sets the Timer interrupt flag TF1. The counted input is enabled to the Timer when TR1 = 1 and either GATE = 0 or $\overline{INT1}$ = 1. (Setting GATE = 1 allows the Timer to be controlled by external input $\overline{INT1}$, to facilitate pulse width measurements). TR1 is a control bit in the Special Function Register TCON (figure 12). GATE is in TMOD.

The 13-bit register consists of all 8 bits of TH1 and the lower 5 bits of TL1. The upper 3 bits of TL1 are indeterminate and should be ignored. Setting the run flag (TR1) does not clear the registers.

Mode 0 operation is the same for Timer 0 as for Timer 1. Substitute TR0, TF0 and $\overline{INT0}$ for the corresponding Timer 1 signals in figure 11. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

Mode 1

Mode 1 is the same as Mode 0, except that the Timer register is being run with all 16 bits.

Mode 2

Mode 2 configures the Timer register as an 8-bit Counter (TL1) with automatic reload, as shown in figure 13. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is preset by software.

The reload leaves TH1 unchanged.

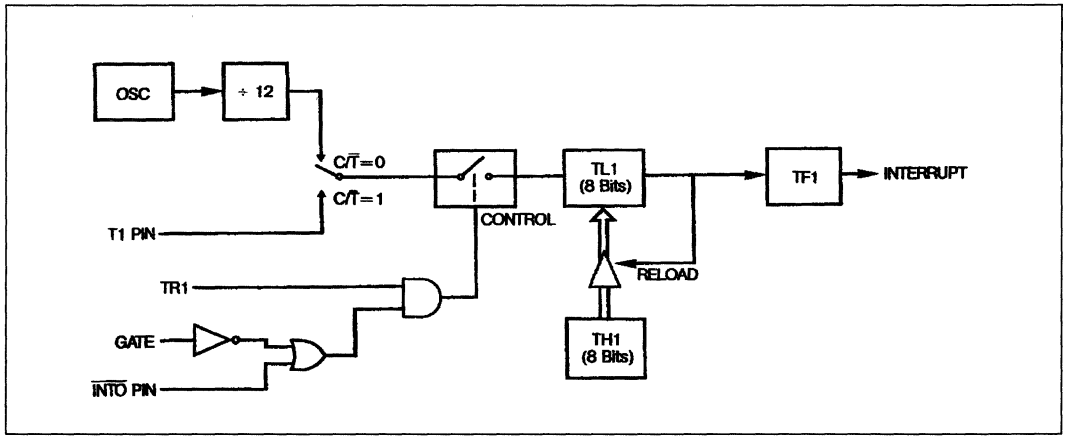


Figure 13 : Timer/Counter 1 Mode 2 : 8-Bit Auto-Reload.

Mode 3

Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0.

Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in Figure 14. TL0 uses the Timer 0 control bits : C/T, GATE, TR0, INT0, and TF0, TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the "Timer 1" interrupt.

Mode 3 is provided for applications requiring an extra 8-bit timer or counter. With Timer 0 in Mode 3, an 80C51 can look like it has three Timer/Counters, and an 80C52, like it has four. When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3, or can still be used by the serial port as a baud rate generator, or in fact, in any application not requiring an interrupt.

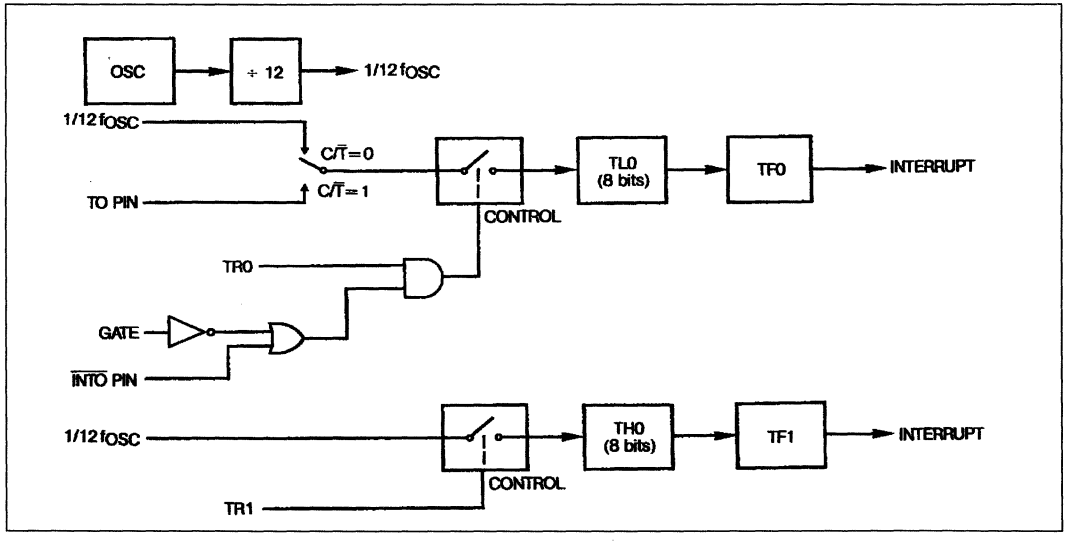


Figure 14 : Timer/Counter 0 Mode 3 : Two 8-Bit Counters.

83C154 and 83C154D

- The 83C154 has two supplementary modes. They are accessed by bits WDT and T32 of register IOCON. Figure 15 shows how IOCON must be programmed in order to have access to these functions.

(MSB) (LSB)

WDT	T32	SERR	IZC	P3HZ	P2HZ	P1HZ	ALF

Symbol	Position	Function
T32	IOCON.6	- If T32 = 1 and if C/$\overline{T0}$ = 0, T1 and T0 are programmed as a 32 bit TIMER. - If T32 = 1 and if C/$\overline{T0}$ = 1, T1 and T0 are programmed as a 32 bit COUNTER.
WDT	IOCON.7	- If WDT = 1 and according to the mode selected by TMOD, an 8-bit or 32-bit WATCHDOG is configured from TIMERS 0 and 1.

Figure 15 : Timer/Counter/Watch-dog Mode Control Register.

32-Bit Mode

- T32 = 1 enables access to this mode. As show in figure 16, this 32-bit mode consists in cascading TIMER 0 for the LSBs and TIMER 1 for the MSBs.

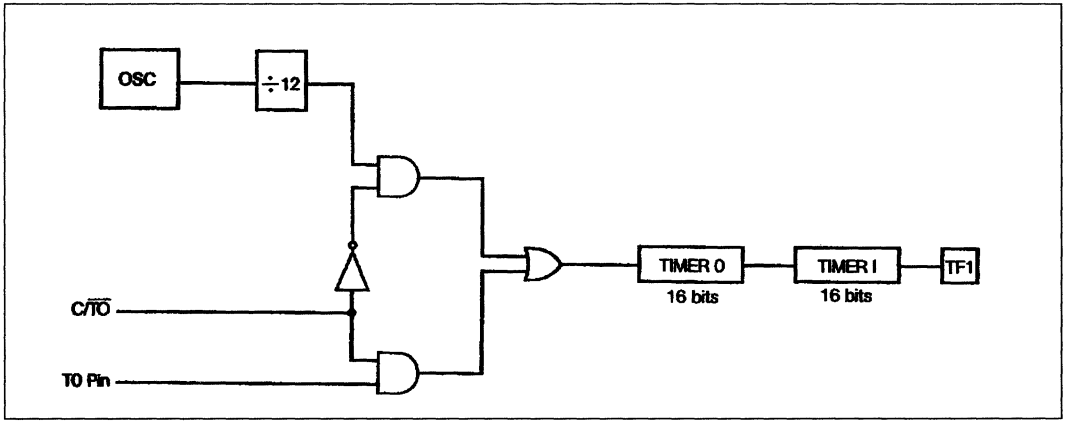


Figure 16 : 32-bit Timer/Counter.

T32 = 1 starts the timer/counter and T32 = 0 stops it.

It should be noted that as soon as T32 = 0, TIMERs 0 and 1 assume the configuration specified by register TMOD. Moreover, if TR0 = 1 or if TR1 = 1, the content of the TIMERs evolves. Consequently, in 32-bit mode, if the TIMER/COUNTER must be stopped (T32 = 0), TR0 and TR1 must be set to 0.

32-Bit Timer

- Figure 17 illustrates the 32-Bit TIMER mode.

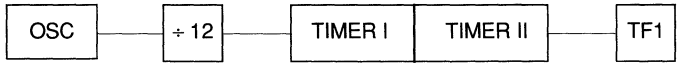


Figure 17 : 32-Bit Timer Configuration.

- In this mode, T32 = 1 and C/$\overline{T0}$ = 0, the 32-bit timer is incremented on each S3P1 state of each machine cycle. An overflow of TIMER 0 (TF0 has not been set to 1) increments TIMER 1 and the overflow of the 32-bit TIMER is signalled by setting TF1 (S5P1) to 1.
- The following formula should be used to calculate the required frequency :

$$f = \frac{OSC}{12 \times (65536 - (T0, T1))}$$

32-Bit Counter

- Figure 18 illustrates the 32-BIT COUNTER mode.

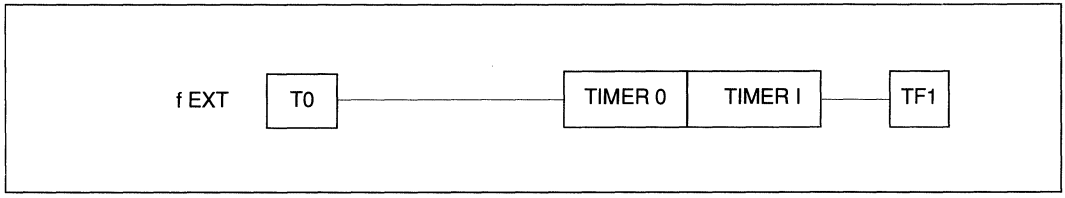


Figure 18 : 32-Bit Counter Configuration.

- In this mode, T32 = 0 and C/$\overline{T0}$ = 1. Before it can make an increment, the 83C154 must detect two transitions on its T0 input. As shown in figure 19, input T0 is sampled on each S5P2 state of every machine cycle or, in other words, every OSC ÷ 12.

4

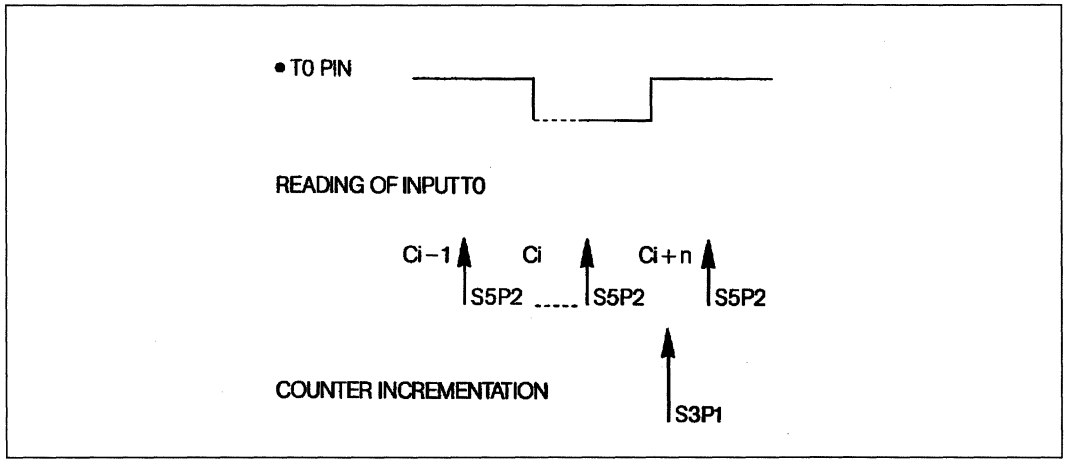


Figure 19 : Counter Incrementation Condition.

- The counter will only evolve if a level 1 is detected during state S5P2 of cycle Ci and if a level 0 is detected during state S5P2 of cycle Ci + n.
- Consequently, the minimal period of signal fEXT admissible by the counter must be greater than or equal to two machine cycles. The following formula should be used to calculate the operating frequency.

$$f = \frac{fEXT}{65536 - (T0, T1))}$$

$$f\,EXT \leq \frac{OSC}{24}$$

Watch-Dog Mode

- WDT = 1 enables access to this mode. As shown in figure 20, all the modes of TIMERSs 0 and 1, of which the overflows act on TF1 (TF1 = 1), activate the WATCH-DOG Mode.

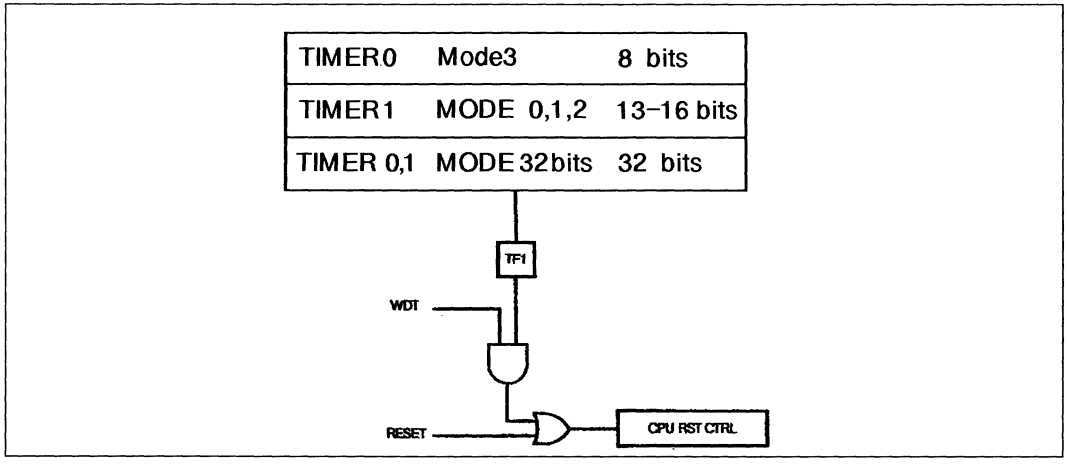


Figure 20 : The Different WATCH-DOG Configurations.

- If C/$\overline{T}$ = 0, the WATCH-DOG is a TIMER that is incremented every machine cycle. If C/$\overline{T}$ = 1, the WATCH-DOG is a counter that is incremented by an external signal of which the frequency cannot exceed OSC ÷ 24.
- The overflow of the TIMER/COUNTER is signalled by raising flag TF1 to 1. The reset of the 83C154/83C154(D) is executed during the next machine cycle and lasts for the next 5 machine cycles. The results of this reset are identical to those of a hardware reset. The internal RAM is not affected and the special register assume the values shown in Table 2.

REGISTER	CONTENT
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	00H
DPTR	0000H
P0-P3	0FFH
IP	00H
IE	0X000000B
TMOD	00H
TCON	00H
T2CON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
TH2	00H
TL2	00H
RCAP2H	00H
RCAP2L	00H
SCON	00H
SBUF	Indeterminate
IOCON	00H

Table 2 : Content of the SFRs after a reset triggered by the WATCHDOG.

- As there are no precautions for protecting bit WDT from spurious writing in the IOCON register, special care must be taken when writing the program. In particular, the user should use the IOCON register bit handling instructions :

- SETB and CLR x

in preference to the byte handling instructions :

- MOV IOCON, # XXH, ORL IOCON, # XXH,
- ANL IOCON, # XXH, ...

External Counting in Power-down Mode (PD = PCON.1 = 1)

- In the power-down mode, the oscillator is turned off and the 83C154s' activity is frozen. However, if an external clock is connected to one of the two inputs, T1/T0, TIMER/COUNTERS 0 and 1 can continue to operate. In this case, counting becomes asynchronous and the maximum, admissible frequency of the signal is OSC : 24.
- The overflow of either counter TF0 or TF1 causes an interrupt to be serviced or forces a reset if the counter is in the WATCH-DOG MODE (T32 = ICON. 7 = 1).

Timer 2

Timer 2 is a 16-bit Timer/Counter which is present only in the 80C52, 83C154 and 83C154D. Like Timers 0 and 1, it can operate either as a timer or as an event counter. This is selected by bit C/T2 in the Special Function Register T2CON (figure 21). It has three operating modes : "capture", "auto-load" and "baud rate generator", which are selected by bits in T2CON as shown in Table 3.

(MSB)							(LSB)
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

Symbol	Position	Function
TF2	T2CON.7	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 OR TCLK = 1.
EXF2	T2CON.6	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2 EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the timer 2 interrupt routine. EXF2 must be cleared by software.
RCLK	T2CON.5	Receive clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in modes 1 and 3. TCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	T2CON.4	Transmit clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	T2CON.3	Timer 2 external enable flag. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	T2CON.2	Start/stop control for Timer 2. A logic 1 starts the Timer.
C/$\overline{T2}$	T2CON.1	Timer or counter select. (Timer 2) 0 = internal timer (OSC/12) 1 = external event counter (falling edge triggered).
CP/$\overline{RL2}$	T2CON.0	Capture/reload flag. When set, captures will occur on negative transitions at T2EX if EXEN2 = 1. When cleared, auto-reloads will occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

Figure 21 : T2CON : Timer/Counter 2 Control Register.

RCLK + TCLK	CP/$\overline{RL2}$	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(off)

Table 3 : Timer 2 Operating Modes.

In the capture Mode there are two options which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then Timer 2 is a 16-bit timer or counter which upon overflowing sets bit TF2, the Timer 2 overflow bit, which can be used to generate an interrupt. If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into registers, RCAP2L and RCAP2H, respectively. (RCAP2L and RCAP2H are new Special Function Registers in the 80C52, 83C154 and 83C154D). In addition, the transition at T2EX causes bit EXF2 in T2CON to be set, and EXF2, like TF2, can generate an interrupt.

The Capture Mode is illustrated in figure 22.

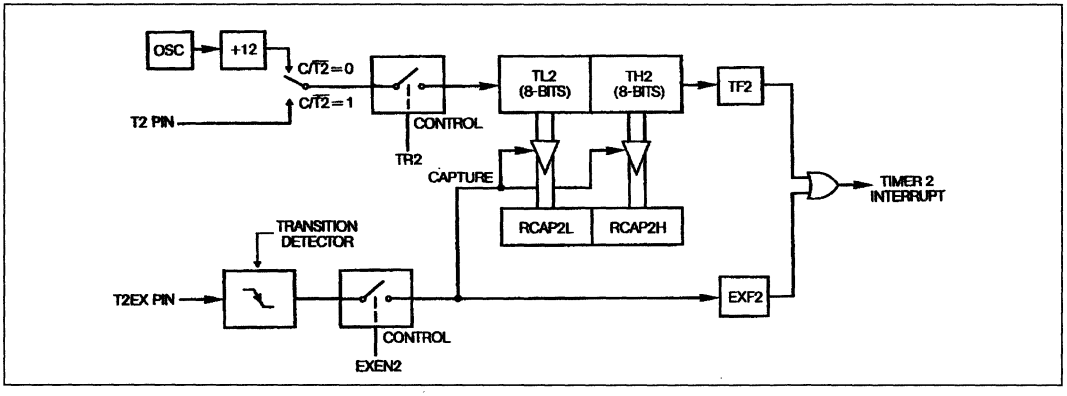


Figure 22 : Timer 2 in Capture Mode.

In the auto-reload mode there are again two options, which are selected by EXEN2 in T2CON. If EXEN2 = 0, then Timer 2 rolls over it not only sets TF2 but also causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. If EXEN2 = 1, then Timer 2 still does the above, but with the added feature preset by software. If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX will also trigger the 16-bit reload and set EXF2.

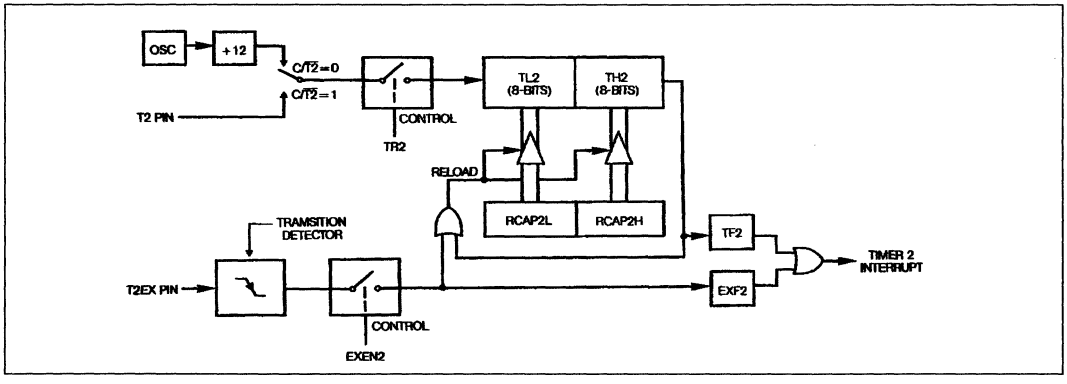


Figure 23 : Timer 2 in Auto-Reload Mode.

The auto-reload mode is illustrated in figure 23.

The baud rate generator mode is selected by RCLK = 1 and/or TCLK = 1. It will be described in conjunction with the serial port.

SERIAL INTERFACE

The serial ports is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can start reception of a second byte before a previously received byte has been read from the receive register. (However, if the first byte still hasn't been read by the time reception of the second byte is completed, one of the bytes will be lost). The serial port receive and transmit registers are both accessed at Special Function Register SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

The serial port can operate in modes :

Mode 0 : Serial data enters and exits through RDX, TDX outputs the shift clock. 8 bits are transmitted/received : 8 data bits (LSD first). The baud rate is fixed at 1/12 the oscillator frequency.

Mode 1 : 10 bits are transmitted (through TXD) or received (through RXD) : a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable.

Mode 2 : 11 bits are transmitted (through TXD) or received (through RXD) : a start bit (0), 8 data bits (LSB first), a programmable 9 th data bit and a stop bit (1). On Transmit, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8. On receive, the 9th data bit goes into RB8 in Special Function Register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency.

Mode 3 : 11 bits are transmitted (through TXD) or received (through RXD) : a start bit (0), 8 data bits (LSB first), a programmable 9th data bit and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except the baud rate. The baud rate in Mode 3 is variable.

In all four modes, transmission, is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

4

Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received. The 9th one goes into RB8. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON. A way to use this feature in multiprocessor systems is as follows.

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received bytes that will be coming. The slaves that weren't being addressed leave their SM2s set and go on about their business, ignoring the coming data bytes.
SM2 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the stop bit. In a Mode 1 reception, if SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

Serial Port Control Register

The serial port control and status register is the Special Function Register SCON, shown in figure 24. This register contains not only the mode selection bits, but also the 9th data bit for transmit and receive (TB8 and RB8), and the serial port interrupt bits (TI and RI).

Baud rates

The baud rate in Mode 0 is fixed :

$$\text{Mode 0 Baud Rate} = \frac{\text{Oscillator Frequency}}{12}$$

The baud rate in Mode 2 depends on the value of bit SMOD in Special Function Register PCON. If SMOD = 0 (which is the value on reset), the baud rate 1/64 the oscillator frequency. If Smod = 1, the baud rate is 1/32 the oscillator frequency.

(MSB)							(LSB)
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Where SM0, SM1 specify the serial port mode, as follows :

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Shift register	fosc /12
0	1	1	8-bit UART	variable
1	0	2	9-bit UART	fosc /64 or fosc /32
1	1	3	9-bit UART	variable

- SM2 enables the multiprocessor communication feature in Modes 2 and 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In Mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0.
- REN enables serial reception. Set by software to enable reception. Clear by software to disable reception.
- TB8 is the 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.
- RB8 in Modes 2 and 3, is the 9th data bit that was received. In Mode 1, if SM2 = 0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.
- TI is transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.
- RI is receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.

Figure 24 : SCON : Serial Port Control Register.

$$\text{Mode 2 Baud Rate} = \frac{2^{SMOD}}{64} \times (\text{Oscillator Frequency})$$

In the 80C51, the baud rates in Modes 1 and 3 is determined by the Timer 1 overflow rate. In the 80C52, 83C154 and 83C154D, these baud rates can be determined by Timer 1, or by Timer 2, or by both (one for transmit and the other for receive).

Using Timer 1 to Generate Baud Rates

When Timer 1 is used as the baud rate generator, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD as follows :

Modes 1, 3

$$\text{Baud Rate} = \frac{2^{SMOD}}{32} \times (\text{Timer 1 Overflow Rate})$$

The Timer 1 interrupt should be disabled in this application. The Timer itself can be configured for either "timer" or "counter" operation, and in any of its 3 running modes. In the most typical applications, it is configured for "timer" operation, in the auto-reload mode (high nibble of TMOD = 0010B). In that case, the baud rate is given by the formula.

Modes 1, 3

$$\text{Baud Rate} = \frac{2^{SMOD}}{32} \times \frac{\text{Oscillator Frequency}}{12 \times [256 - (TH1)]}$$

One can achieve very low baud rates with Timer 1 by leaving the Timer 1 interrupt enabled, and configuring the Timer to run a 16-bit timer (high nibble of TMOD = 0001B), and using the Timer 1 interrupt to do a 16-bit software reload.

Figure 25 lists various commonly used baud rates and how they can be obtained from Timer 1.

BAUD RATE	f_{osc}	SMOD	TIMER 1		
			C/T	MODE	RELOAD VALUE
Mode 0 Max : 1 M	12 MHZ	X	X	X	X
Mode 2 Max : 375 K	12 MHZ	1	X	X	X
Mode 1,3 : 62.5 K	12 MHZ	1	0	2	FFH
19.2 K	11.059 MHZ	1	0	2	FDH
9.6 K	11.059 MHZ	0	0	2	FDH
4.8 K	11.059 MHZ	0	0	2	FAH
2.4 K	11.059 MHZ	0	0	2	F4H
1.2 K	11.059 MHZ	0	0	2	E8H
137.5 K	11.986 MHZ	0	0	2	1DH
110 K	6 MHZ	0	0	2	72 H
110 K	12 MHZ	0	0	1	FEEBH

Figure 25 : Timer 1 Generated Commonly Used Baud Rates.

Using Timer 2 to Generate Baud Rates

In the 80C52 and 83C154/83C154D, Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (figure 12). Note then the baud rates for transmit and receive can be simultaneously different. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 26.

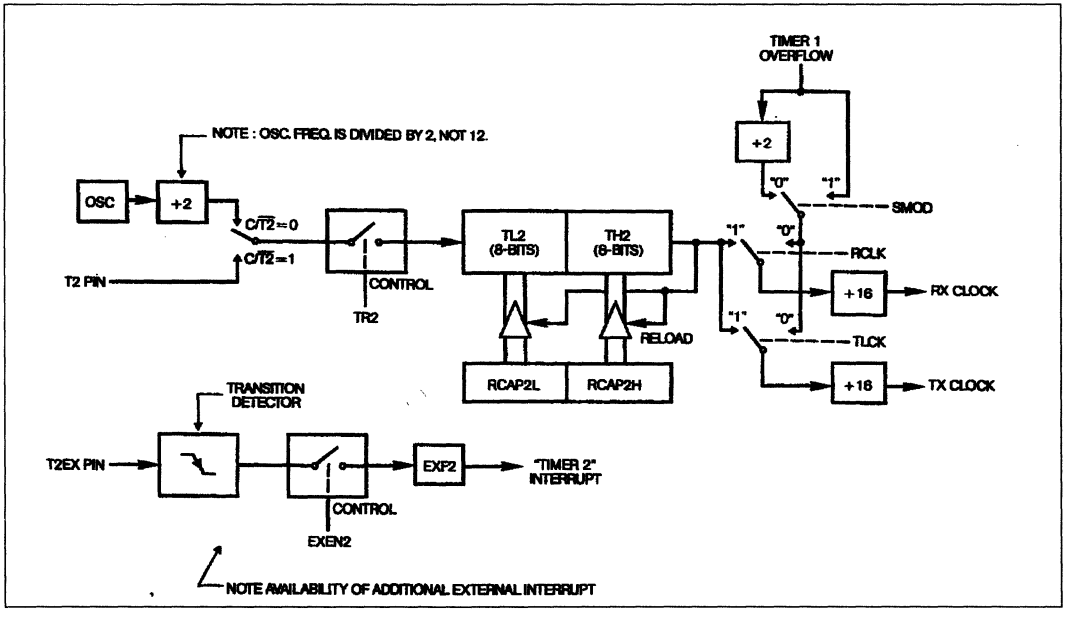


Figure 26 : Timer 2 in Baud Rate Generator Mode.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

Now, the baud rates in Modes 1 and 3 are determinated by Timer 2's overflow rate as follows :

$$\text{Modes 1, 3 Baud Rate} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either "timer" or "counter" operation. In the most typical applications, it is configured for "timer" operation ($C/\overline{T2} = 0$). "Timer" operation is a little different for Timer 2 when it's being used as a baud rate generator. Normally, as a timer it would increment every machine cycle (thus at 1/12 the oscillator frequency). In that case the baud rate is given by the formula

$$\text{Modes 1,3 Baud Rate} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H, RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 25. This Figure is valid only if RCLK + TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2, and will not generate an interrupt. Therefore, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt, if desired.

It should be noted that when Timer 2 is running (TR2 = 1) in "timer" function in the baud rate generator mode, one should not try to read or write TH2 or TL2. Under these conditions the Timer is being incremented every state time, and the results of a read or write may not be accurate. The RCAP registers may be read, but shouldn't be written to, because a write might overlap a reload and cause write and/or reload errors. Turn the Timer off (clear TR2) before accessing the timer 2 or RCAP registers, in this case.

More about Mode 0

Serial data enters and exits through RXD.TXD outputs the shift clock. 8 bits are transmitted/received : 8 data bits (LSB first). The baud rate is fixed at 1/12 the oscillator frequency.

Figure 27 shows a simplified functional diagram of the serial port in Mode 0, and associated timing.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal at S6P2 also loads a 1 into the 9th position of the transmit shift register and tells the TX Control block to commence a transmission. The internal timing is such that one full machine cycle will elapse between "write to SBUF", and activation of SEND.

SEND enables the output of the shift register to the alternate output function of P3.0, and also enables SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK is low during S3, S4, and S5 of every machine cycle, and high during S6, S1 and S2. At S6P2 of every machine cycle in which SEND is active, the contents of the transmit shift register are shifted to the right one position.

As data bits shift out the right, zeroes come in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position, is just to the left of the MSB, and all positions to the left of contain zeroes. This condition flags the TX Control block to do one last shift and then deactivate SEND and set TI. Both of these actions occur at S1P1 of the 10th machine cycle after "write to SBUF".

Reception is initiated by the condition REN = 1 and R1 = 0. At S6P2 of the next machine cycle, the RX Control unit writes the bits 11111110 to the receive shift register, and in the next clock phase activates RECEIVE.

RECEIVE enables SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK makes transitions at S3P1. and S6P1. of every machine cycle. At S6P2 of every machine cycle In which RECEIVE is active, the contents of the receive shift register are shifted to the left one position. The value that comes in from the right is the value that was sampled at the P3.0 pin at S5P2 of the same machine cycle.

As data bits come in from the right, 1s shift out to the left. When the 0 that was initially loaded into the rightmost position arrives at the leftmost position in the shift register, it flags the RX Control block to do one last shift and load SBUF. At S1P1 of the 10th machine cycle after the write to SCON that cleared RI, RECEIVE is cleared and RI is set.

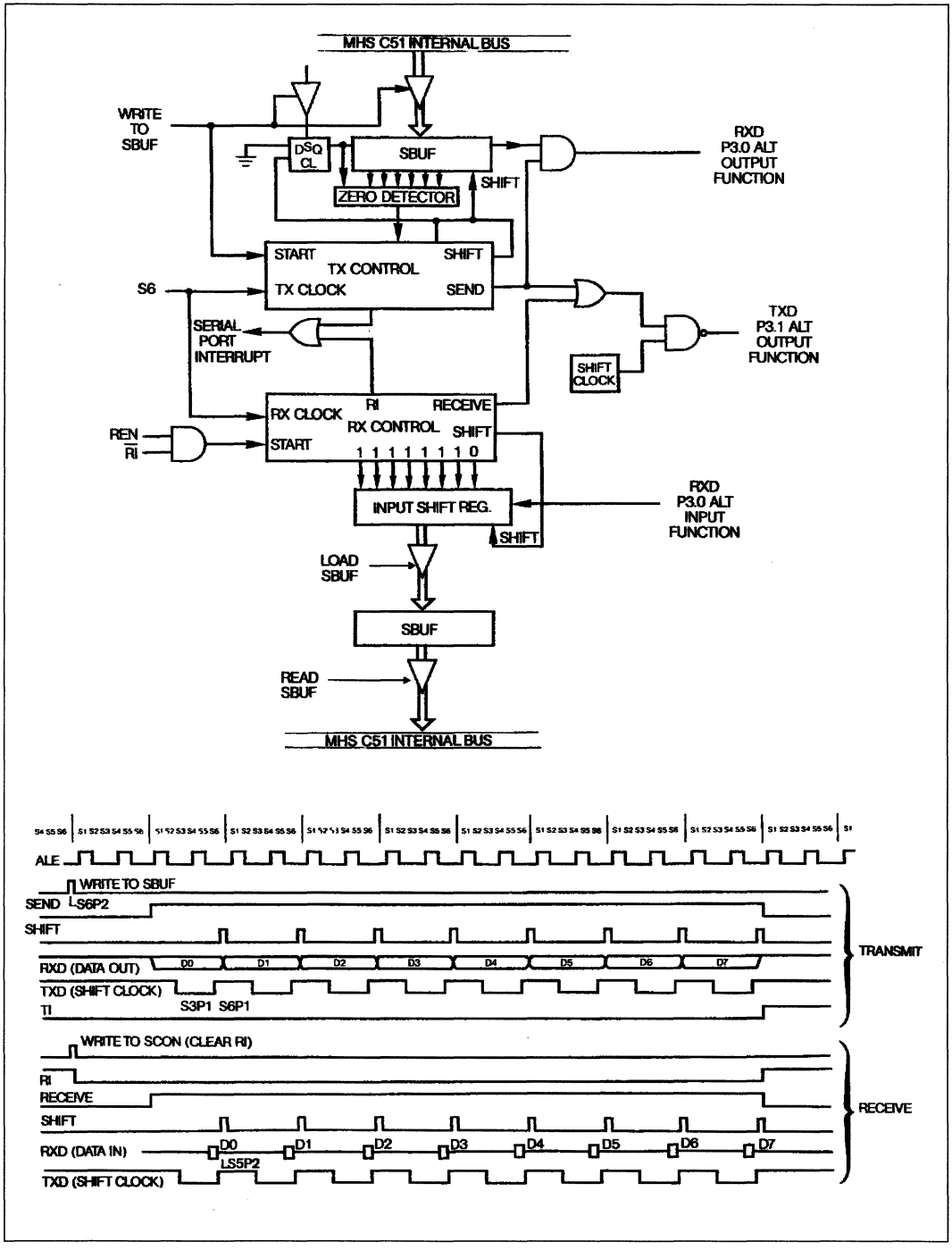


Figure 27 : Serial Port Mode 0.

More about Mode 1

Ten bits are transmitted (through TDX), or received (through RXD) : a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in SCON. In the 80C51 the baud rate is determinated by the Timer 1 overflow rate. In the 80C52, 83C154 and 83C154D it is determinated either by the Timer 1 overflow rate, or the Timer 2 overflow rate, or both (one for transmit and the other for receive).

Figure 28 shows a simplified functional diagram of the serial port in Mode 1, and associated timings for transmit receive.

Transmission is initiated by any Instruction that uses SBUF as a destination register. The "write to SBUF" signal also loads a 1 into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission actually commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. (Thus, the bit times are synchronized to the divide-by-16 counter, not to the "write to SBUF" signal). The transmission begins with activation of SEND, which puts the start bit at TXD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TXD. The first shift pulse occurs one bit time after that.

As data bits shift out to the right, zeroes clocked in from the left. When the MSB of data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position is just to the left of the MSB, and all positions to the left that contain zeroes. This condition flags TX Control unit to do one last shift and then desactivate SEND and set TI. This occurs at the 10th divide-by-16 rollover after "write to SBUF".

Reception is initiated by a detected 1-to-0 transition at RXD. For this purpose RXD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written into the input shift register. Resetting the divide-by-16 counter aligns its rollovers with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16ths. At the 7th, 8th, and 9th counter states of each bit time, the bit detector samples the value of RXD. The value accepted is the value that was seen in at least 2 of 3 samples. This is done for noise rejection. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. This is to provide rejection of false start bits. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come in from the right, 1s shift out of the left. When the start bit arrives at the leftmost position in the shift register, (which in mode 1 is a 9-bit register), it flags the RX Control block to do one last shift, load SBUF and RB8, and to set RI, the signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated.

1) RI = 0, and
2) Either SM2 = 0, or the received stop bit = 1

In either of these two conditions is not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bits go into SBUF, and RI is activated. At this time, whether the above conditions are met or not, the unit goes back to looking for a 1-to-0 transition in RXD.

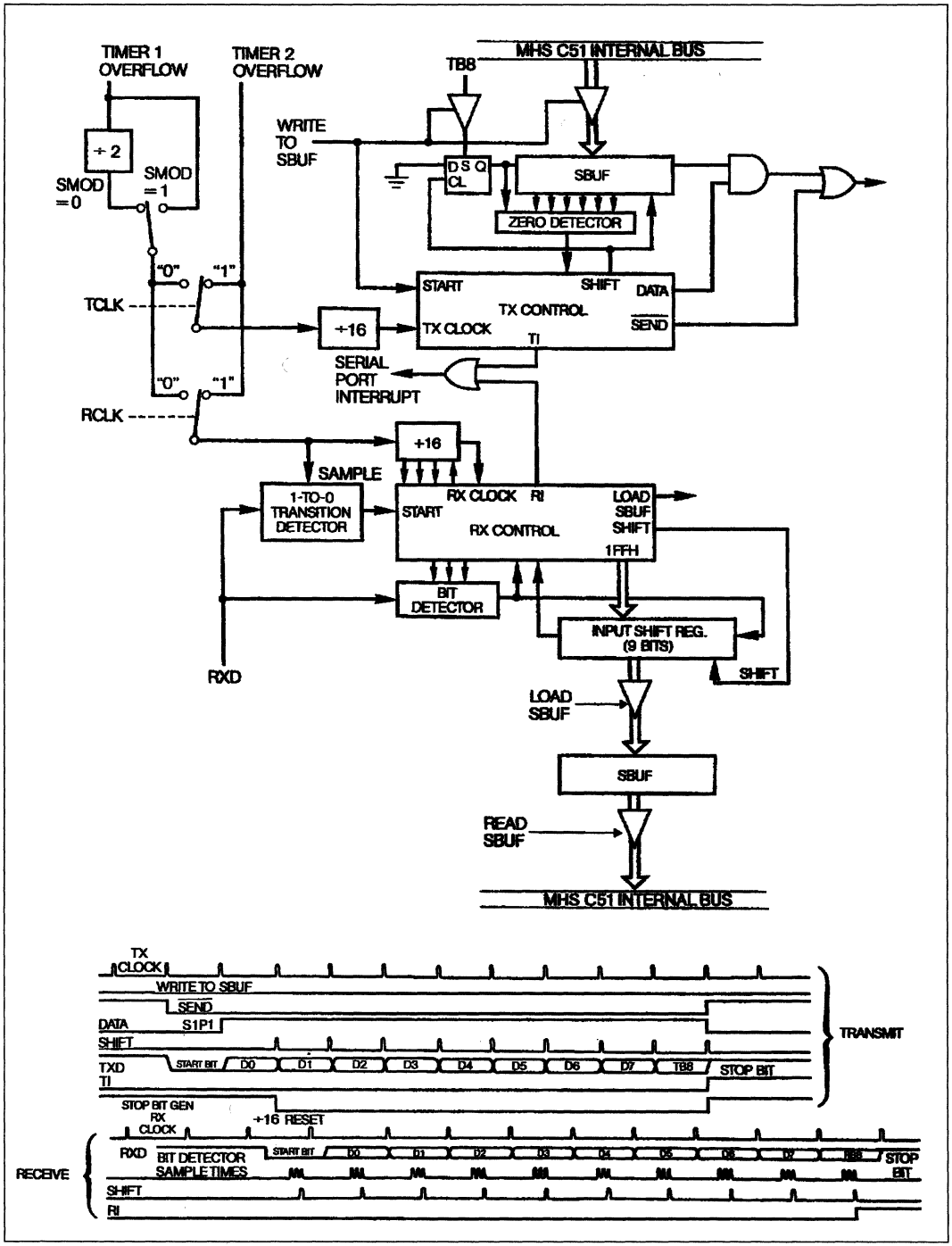


Figure 28 : Serial Port Mode 1. TCLK, RCLK and Timer 2 are Present in the 80C52, 83C154 and 83C154D only.

More about Modes 2 and 3

Eleven bits are transmitted (through TXD), or received (through RXD) : a start bit (0), 8 data bits (LSB first), a programmable 9th data bit and a stop bit (1). On transmit the 9th data bit (TB8) can be assigned the value of 0 or 1. On receive, the 9th data bit goes into RB8 in SCON. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency in Mode 2.

Mode 3 may have a variable baud rate generated from either Timer 1 or 2 depending on the state of TCLK and RCLK.

Figures 29 and 30 show a functional diagram of the serial port in Modes 2 and 3. The receive portion is exactly the same as in Mode 1. The transmit portion differs from Mode 1 only in the 9th bit of the transmit shift register.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal also loads TB8 into the 9th bit position of the transmit shift register and flags the TX control unit that a transmission is requested. Transmission commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. (Thus, the bit times are synchronized to the divide-by-16 counter, not to the "write to SBUF" signal).

The transmission begins with activation of SEND, which puts the start bit at TXD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TXD. The first shift pulse occurs one bit time after that. The first shift clocks a 1 (the stop bit) into the 9th bit position of the shift register. Thereafter, only zeroes are clocked in. Thus as data bits shift out to the right, zeroes are clocked in from the left. When TB8 is at the output position of the shift register, then the stop bit is just to the left of TB8, and all positions to the left of that contain zeroes. This condition flags the TX Control unit to do one last shift and then desactivate SEND and set TI. This occurs at the 11th divide-by-16 rollover after "write to SBUF".

Reception is initiated by a detected 1-to-0 transition at RXD. For this purpose RXD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written to the input shift register.

At the 7th, 8th and 9th counter states of each bit time, the bit detector samples the value of RXD. The value accepted is the value that was been in at least 2 of the 3 samples. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register (which in Modes 2 and 3 is a 9-bit register), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated :

1) RI = 0, and
2) Either SM2 = 0 or the received 9th data bit = 1

If either of these conditions is not met, the received frame is irretrievably lost, and RI is not set. If both conditions are met, the received 9th data bit goes into RB8, and the first 8 data bits go into SBUF. One bit time later, whether the above conditions were met or not, the unit goes back to looking for 1-to-0 transition at the RXD input.

Note that the value of the received stop bit is irrelevant to SBUF, RB8 or RI.

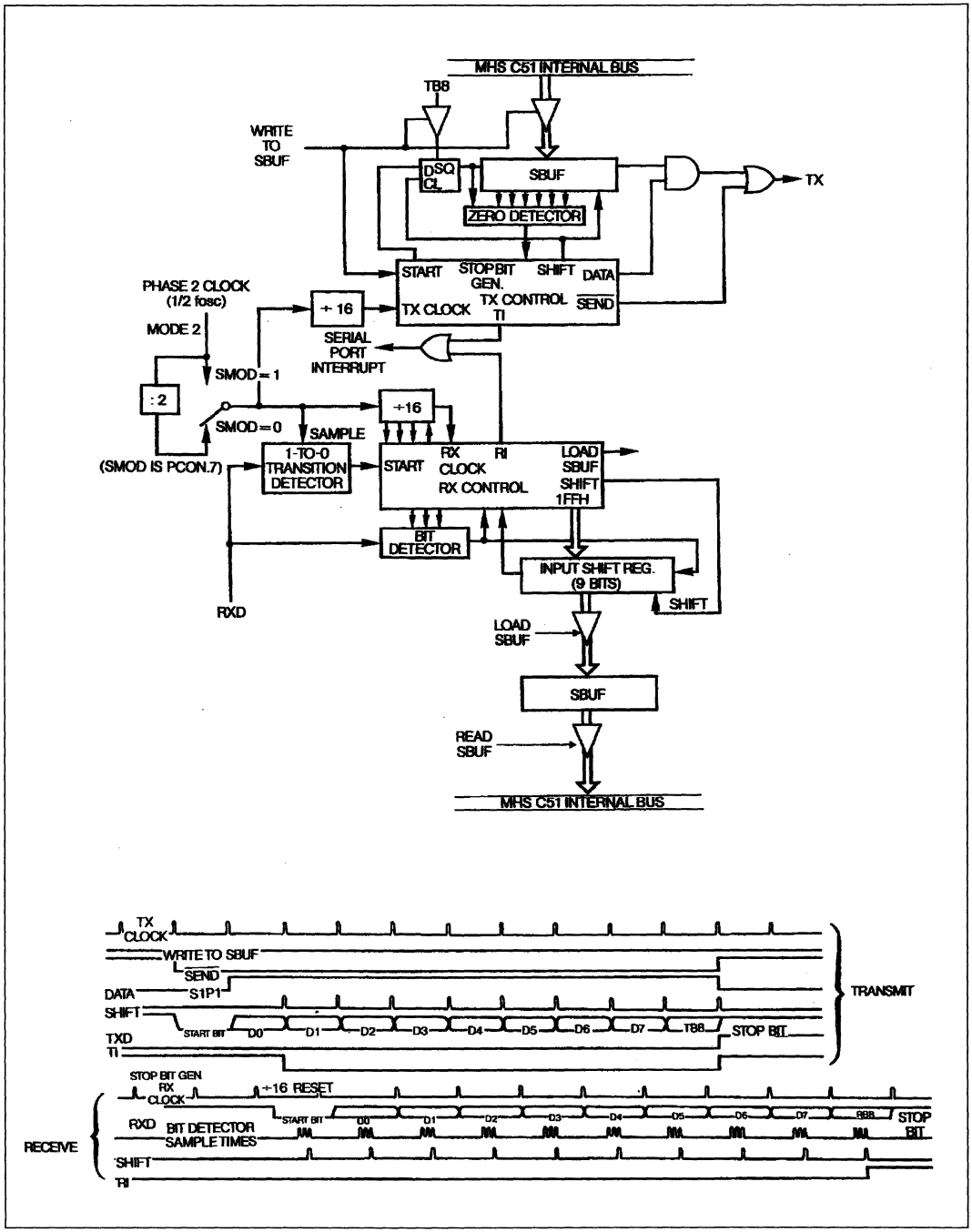


Figure 29 : Serial Port Mode 2.

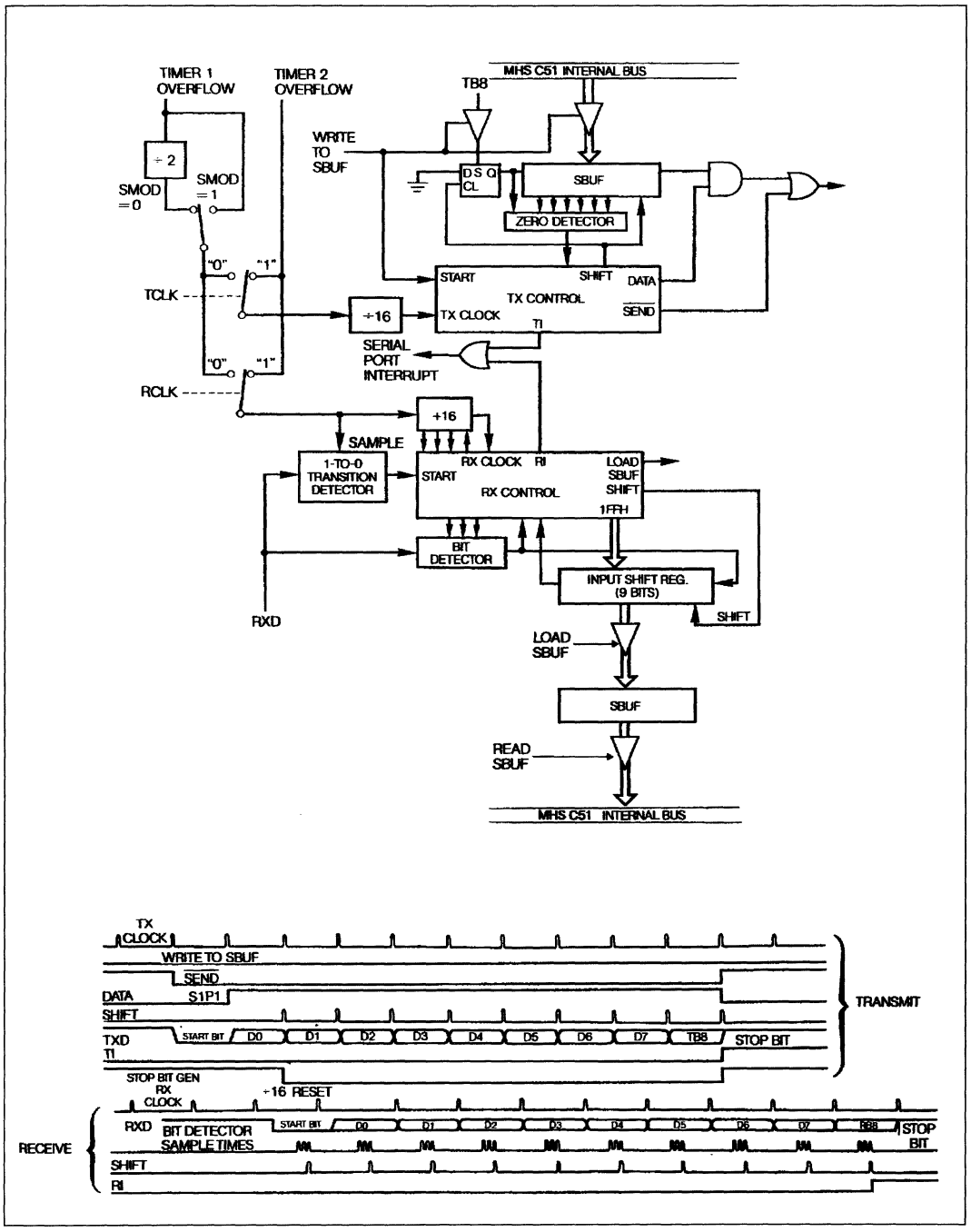


Figure 30 : Serial Port Mode 3. TCLK, RCLK, and Timer 2 are Present in the 80C52/80C32 and 83C154/83C154D.

Error Detection in Reception (83C154 and 83C154D only)

- A supplementary IOCON register bit, SERR = IOCON.5, enables detection of a RECEPTION error. Two types of error are possible : OVERRUN error and FRAME error.

Frame Error

- SERR = 1 indicates that a data format error has been detected. All the bits of a same character are sampled on the 7th, 8th and 9th RECEPTION clock cycles. A majority vote determines the logical state of the bit received. A character terminates with one or more stop bits (level 1). In figure 31, the stop bit is missing (level) and bit SERR is set to 1 at the same time as bit RI.

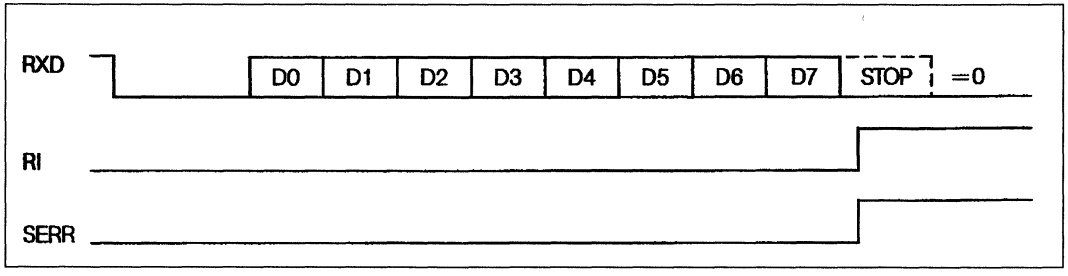


Figure 31 : SERR = 1 signals an error in the format of the received bit.

- SERR is cleared by the software.

4

Overrun Error

- SERR = 1, indicates that the previously received character has not been read and has been replaced by the next character. In figure 32, a first character has been received and flag RI is set at 1. A second character is received before the first character has been read (RI is still at 1). The first character is lost and the SERR flag is raised to 1 to signal this error.

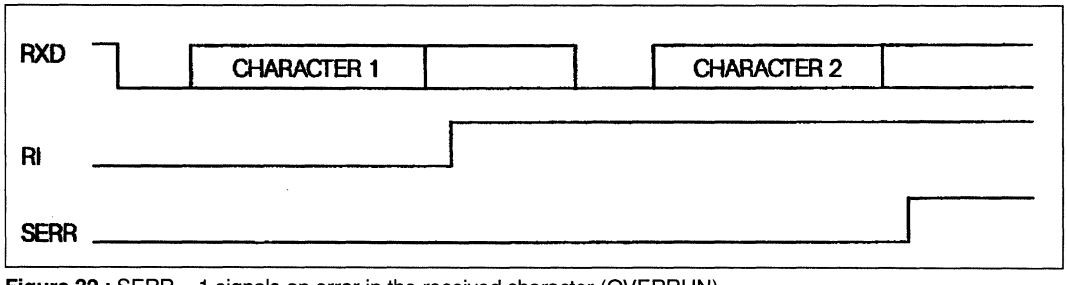


Figure 32 : SERR = 1 signals an error in the received character (OVERRUN).

- SERR is cleared by the software.

Serial Link in Power-down and Idle Mode

- In POWER-DOWN (PD = 1) or IDLE (IDL = 1) mode the serial link can continue to transmit and receive in Modes 1 and 3. The transmission/reception clock is generated by counter 1 ($C/\overline{T1}$ = 1, GATE = 0) and the external clock must not exceed OSC ÷ 24. An interrupt generated by the serial link (RI = 1 or TI = 1) enables exit from these two modes.
- All of the bits that generate interrupts can be set or cleared by software, with the same result as though it had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be cancelled in software.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE (figure 34). IE contains also a global disable bit, EA, which disables all interrupts at once.

Note in figure 34 that bit position IE.6 is unimplemented. In the 80C51 bit position IE.5 is also unimplemented. User software should not write 1s in these bit positions, since they may be used in future MHS C51 products.

INTERRUPTS

The 80C51 provides 5 interrupt sources. The 80C52, 83C154 and 83C154D provides 6. They are shown in figure 33.

The External Interrupts $\overline{INT0}$ and $\overline{INT1}$ can each be either level-activated or transition-activated, depending on bits IT0 and IT1 in Register TCON. The flags that actually generate these interrupts are bits IE0 and IE1 in TCON. When an external interrupt is generated, the flag that generated it is cleared by the hardware when the service routine is vectored to only if the interrupt was transition-activated. If the interrupt was level-activated, then the external requesting source is what controls the request flag, rather than the on-chip hardware.

The Timer 0 and Timer 1 Interrupts are generated by TF0 and TF1, which are set by a rollover in their respective Timer/Counter registers (except see Timer 0 in Mode 3). When a timer interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored to.

The Serial Port Interrupt is generated by the logical OR of RI and TI. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine will normally have to determine whether it was RI or TI that generated the interrupt, and the bit will have to be cleared in software.

In the 80C52, 83C154 and 83C154D, the Timer 2 Interrupt is generated by the logical OR of TF2 and EXF2. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and the bit will have to be cleared in software.

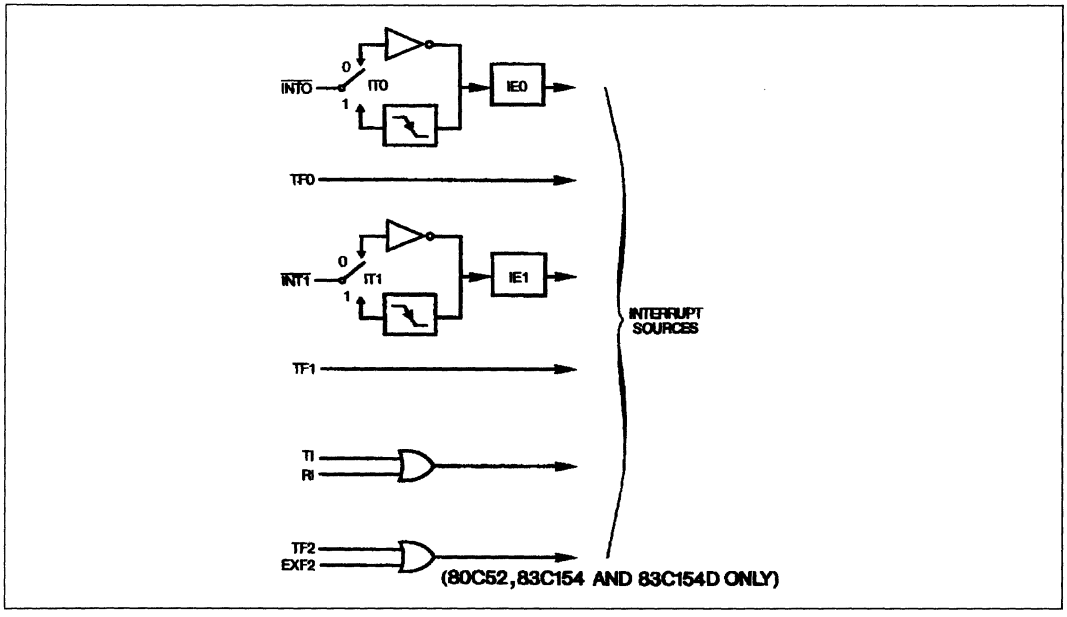


Figure 33 : MHS C51 Interrupt Sources.

(MSB)							(LSB)
EA	X	ET2	ES	ET1	EX1	ET0	EX0

Symbol	Position	Function
$\overline{EA}$	IE.7	disables all interrupts. If $\overline{EA}$ = 0, no interrupt will be acknowledge. If $\overline{EA}$ = 1 each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
X	IE.6	reserved
$\overline{ET2}$	IE.5	enables or disables the Timer 2 Overflow or capture interrupt. If ET2 = 0, the Timer 2 interrupt is disabled.
ES	IE.4	enables or disables the Serial Port interrupt. If ES = 0, the Serial Port interrupt is disabled.
ET1	IE.3	enables or disables the Timer 1 Overflow interrupt. If ET1 = 0, the Timer 1 interrupt is disabled.
EX1	IE.2	enables or disables External Interrupt 1. If EX1 = 0, External Interrupt is disabled.
ET0	IE.1	enables or disables the Timer 0 Overflow interrupt. If ET0 = 0, the Timer 0 interrupt is disabled.
EX0	IE.0	enables or disables External Interrupt 0. If EX0 = 0, External Interrupt 0 is disabled.

Figure 34 : IE : Interrupt Enable Register.

4

Priority Level Structure

- Register IP (B8H) makes it possible for all interrupts to have 2 levels of priority. Figure 35 shows the content of this register.

(MSB)							(LSB)
PCT	X	PT2	PS	PT1	PX1	PT0	PX0

Symbol	Position	Function
PCT	IP.7	PCT = 1, only one level (83C154 and 83C154D only)
	IP.6	reserved
PT2	IP.5	defines the Timer 2 interrupt priority level. PT2 = 1 programs it to the higher priority level.
PS	IP.4	defines the Serial Port interrupt priority level. PS = 1 programs it to the higher priority level.
PT1	IP.3	defines the Timer 1 interrupt priority level. PT1 = 1 programs it to the higher priority level.
PX1	IP.2	defines the external interrupt 1 priority level. PX1 = 1 programs it to the higher priority level.
PT0	IP.1	defines the Timer 0 interrupt priority level. PT0 = 1 programs it to the higher priority level.
PX0	IP.0	defines the External interrupt 0 priority level. PX0 = 1 programs it to the higher priority level.

Figure 35 : IP : Interrupt Priority Register.

IP = 0, selection of first priority level. Therefore, all interrupts have this priority level. Figure 36 shows the order in which the interrupts are accepted within a same level.

	SOURCE	PRIORITY WITHIN LEVEL
1	IE0	(Highest)
2	TF0	
3	IE1	
4	TF1	
5	RI + TI	
6	TF2 + EX2	(Lowest)

Figure 36 : Order of priority within a priority level.

- Interrupt requests are read during states S2 to S5 of each machine cycle. During state S6, a polling is executed to determine which interrupt will be served. The order in which the requests are read conforms to the figure above, IE0 ----> TF2 + EX2. The first request read is executed and instruction RETI terminates the interrupt sub-routine. The flag corresponding to the interrupt is cleared (by software or hardware). If two interrupt requests occur at the same time, the first read is executed.
- To enable an interrupt with a lower read priority to be serviced in priority, it is possible to program the interrupt bit of register IP to 1. Thus, if 2 interrupt requests occur simultaneously, the first to be serviced will not be the first request read but that with the highest priority. Therefore, if PT1 = 1 a simultaneous request from IE0 and IE1 will result in IE1 being serviced first.
- Bit PCT IP.7, only present in the 83C154 and 83C154D, enables inhibiting of the 2nd priority level. Therefore, all interrupts will have the same level and operation is identical to that of IP = 0.

How Interrupts are Handled

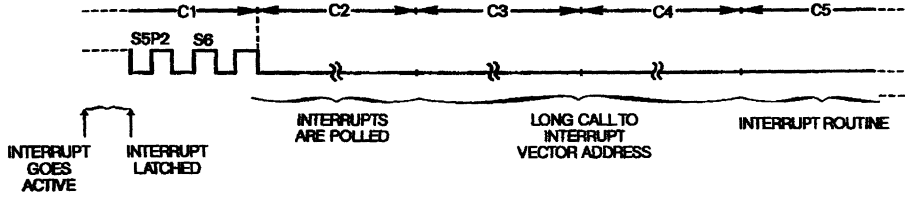
- The interrupt flags are sampled at S5P2 of every machine cycle. The samples are polled during the following machine cycle. If one of the flags was set in a condition at S5P2 of the preceding cycle, the polling cycle will find it and interrupt system will generate a LCALL to the appropriate service routine, provided this hardware-generated LCALL is not blocked by any of the following conditions :

1. An interrupt of equal or higher priority is already in progress.
2. The current (polling) cycle is not the final cycle in the execution of the instruction in progress.
3. The instruction in progress is not RET or any write to the IE or IP registers.

- Any of these three conditions will block the generation of the LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress will be completed before vectoring to any service routine. Condition 3 ensures that if the instruction in progress is RETI or any access to IE or IP, then at least one more instruction will be executed before any interrupt is vectored to.
- The polling cycle is repeated with each machine cycle, and the values that were present at S5P2 of the previous machine cycle. Note then that if an interrupt flag is active but not being responded to for one of the above conditions, if the flag is not still active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new.

The polling cycle/LCALL sequence is illustrated in figure 37.

- Note that if an interrupt of higher level goes active priority to S5P2 of the machine cycle labelled C3 in figure 37 then in accordance with the above rules it will be vectored to during C5 and C6, without an instruction of the lower priority routine having been executed.



This is the fastest possible response when C2 is the final cycle of an instruction other than RETI or an access to IE or IP.

Figure 37 : Interrupt Response Timing Diagram.

Thus the processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate servicing routine. In some cases it also clears the flag that generated the interrupt, and in other cases it doesn't. It never clears the Serial Port or Timer 2 flags. This has be done in the user's software. It clears an external interrupt flag (IE0 or IE1) only if it was transition-activated. The hardware-generated LCALL pushes the contents of the Program Counter onto the stack (but it does not save the PSW) and reloads the PC with an address that depends on the source of the interrupt being vectored to, as shown below.

Source	Vector Address	CLEARED BY HARDWARE
IE0	0003H	Only on trigger edge (IT0 = 0)
TF0	000BH	At the end of the interrupt routine
IE1	0013H	Only on trigger edge (IT1 = 0)
TF1	001BH	At the end of the interrupt routine
RI + TI	0023H	} Cleared by software
TF2 + EXF2	002BH	

Execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that this interrupt routine is no longer in progress, then pops the top two bytes from the stack and reloads the Program Counter. Execution of the Interrupted program continues from where it left off.

Note that a simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking an interrupt was still in progress.

External Interrupts

The external sources can be programmed to be level-activated or transition-activated by setting or clearing bit IT1 or IT0 in Register TCON. If ITx = 0, external interrupt x is triggered by a detected low at the INTx pin. If ITx = 1, external interrupt x is edge-triggered. In this mode if successive samples of the INTx pin show a high in one cycle and a low in the next cycle, interrupt requests flag IEx in TCON is set. Flag bit IEx then requests the interrupt.

Since the external interrupt pins are sampled once each machine cycle, an input high or low should hold for at least 12 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin high for at least one cycle, and then hold it low for at least one cycle to ensure that the transition is seen so that interrupt request flag IEx will be automatically cleared by the CPU when the service routine is called.

If the external interrupt is level-activated, the external source has to hold the request active until the requested interrupt is actually generated. Then it has to desactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

Response Time

The INT0 and INT1 levels are inverted and latched into IE0 and IE1 at S5P2 of every machine cycle. The values are not actually polled by the circuitry until the next machine cycle. If a request is active and conditions are right for it to be acknowledged, a hardware subroutine call to the requested service routine will be the next instruction to be executed. The call itself takes two cycles. Thus, a minimum of three complete machine cycles elapse between activation of an external interrupt request and the beginning of execution of the first instruction of the service routine. Figure 37 shows interrupt response timings.

A longer response time would result if the request is blocked by one of the 3 previously listed conditions. If an interrupt of equal or higher priority level is already in progress, the additional wait time obviously depends on the nature of the other interrupt's service routine. If the instruction in progress is not in its final cycle, the additional wait time cannot be more than 3 cycles, since the longest instructions (MUL and DIV) are only 4 cycles long, and if the instruction in progress is RETI or an access to IE or IP, the additional wait time cannot be more than 5 cycles (a maximum of one more cycle to complete the instruction in progress, plus 4 cycles to complete the next instruction if the instruction is MUL or DIV).

Thus, in a single-interrupt system, the response time is always more than 3 cycles and less than 9 cycles.

SINGLE-STEP OPERATION

The MHS C51 interrupt structure allows single-step execution with very little software overhead. As previously noted, an interrupt request will not be responded to while an interrupt of equal priority level is still in progress, nor will it be responded to after RETI until at least one other instruction has been executed. Thus, once an interrupt routine has been entered, it cannot be re-entered until at least one instruction of the interrupted program is executed. One way to use this feature for single-stop operation is to program one of the external interrupts (INT0) to be level-activated. The service routine for the interrupt will terminate with the following code :

JNB P3.2,S : Wait Here Till $\overline{INT0}$ Goes High
JB P3.2,S : Now Wait Here Till it Goes Low
RETI : Go Back and Execute One Instruction

Now if the $\overline{INT0}$ pin, which is also the P3.2 pin, is held normally low, the CPU will go right into the External Interrupt 0 routine and stay there until $\overline{INT0}$ is pulsed (from low to high to low). Then it will execute RETI, go back to the task program, execute one instruction, and immediately reenter the External Interrupt 0 routine to await the next pulsing of P3.2. One step of the task program is executed each time P3.2 is pulsed.

RESET

The reset input is the RST pin, which is input to a Schmitt Trigger.

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), *while the oscillator is running*. The CPU responds by generating an internal reset, with the timing shown in figure 38.

The external reset signal is asynchronous to the internal clock. The RST pin is sampled during State 5 Phase 2 of every machine cycle. The port pins will maintain their current activities for 19 oscillator periods after a logic 1 has been sampled at the RST pin ; that is, for 19 to 31 oscillator periods after the external reset signal has been applied to the RST pin.

While the RST pin is high, ALE and PSEN are weakly pulled high. After RST is pulled low, it will take 1 to 2 machine cycles for ALE and PSEN to start clocking. For this reason, other devices can not be synchronized to the internal timings of the 8051.

Driving the ALE and $\overline{PSEN}$ pins to 0 while reset is active could cause the device to go into an indeterminate state.

The internal reset algorithm writes 0s to all the SFRs except the port latches, the Stack Pointer, and SBUF. The port latches are initialized to FFH, the Stack Pointer to 07H, and SBUF is indeterminate. Table 2 lists the SFRs and their reset values.

The internal RAM is not affected by reset. On power up the RAM content is indeterminate.

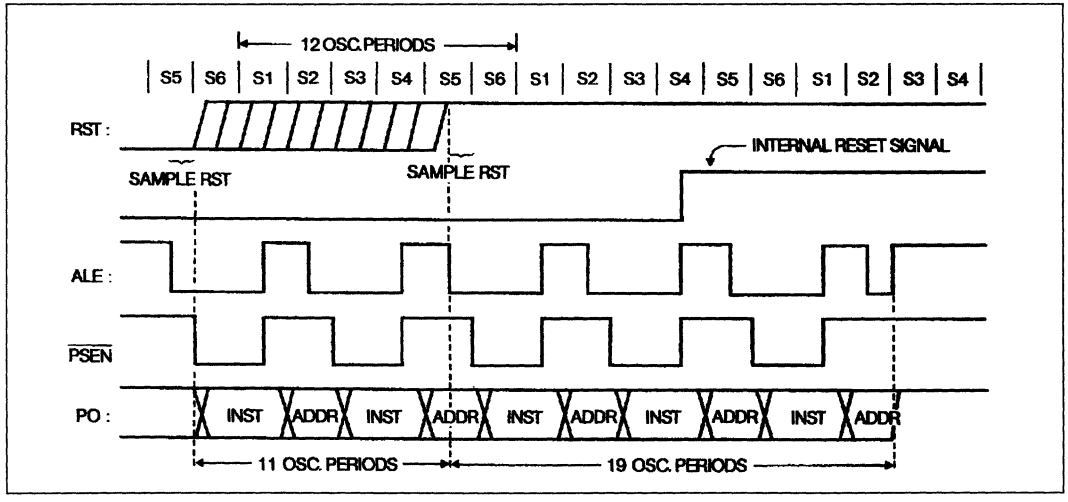


Figure 38 : Reset Timing.

REGISTER	CONTENT
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	07H
DPTR	0000H
P0-P3	0FFH
IP	00H
IE	0X000000B
TMOD	00H
TCON	00H
T2CON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
TH2	00H
TL2	00H
RCAP2H	00H
RCAP2L	00H
SCON	00H
SBUF	Indeterminate
IOCON	00H

Table 4 : Reset Values of the SFRs.

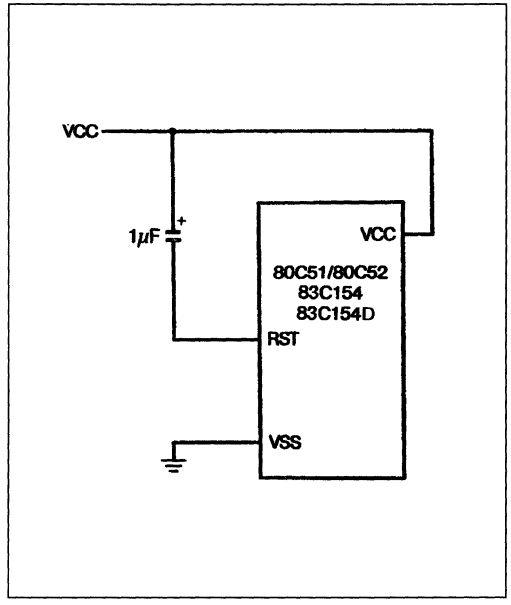


Figure 39 : Power on Reset Circuit.

4

POWER-ON RESET

- An automatic reset on powering-up can be obtained by connecting a 0.1µF capacitance between input RST and the Vcc. The power supply's rise time must not exceed 1 ms and the oscillator start-up time must not exceed 10 ms.
- Note that the port I/Os will be in a determinate state (0FFH) as soon as the input RST is active (high).
- With this circuit (figure 39), if Vcc is reduced rapidly, input RST will momentarily go below 0. However, input RST is protected internally.

POWER-SAVING MODES OF OPERATION

- For applications in which power consumption is a critical parameter, the MHS C51 family offers two power-saving modes : IDLE and POWER-DOWN. Figure 40 illustrates the principle used for implementing these two modes.

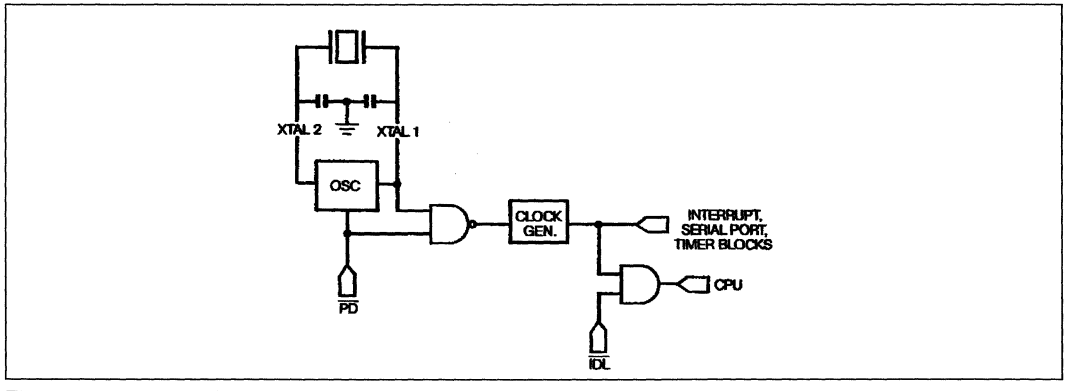


Figure 40 : Principle of the IDLE and POWER-DOWN Modes.

- In IDLE mode, the oscillator continues to operate and the interrupts the serial port and timers 0 and 1 remain under the internal oscillator's control. Only the CPU is no longer driven by the clock.
- In POWER-DOWN mode, the oscillator is turned off and none of the functions operate.

These two modes are called by two bits, IDL = PCON.0 and PD = PCON.1, which are contained in the special PCON register (address 87H). This register is not bit-addressable. Figure 41 shows the detail of the PCON register's content.

(MSB)							(LSB)
SMOD	HPD	RPD	X	GF1	GF0	PD	IDL

Symbol	Position	Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1, 2 or 3.
HPD	PCON.6 (83C154 and 83C154D only)	Hard Power Down bit. Setting this bit allows CPU to enter in Power Down state on an external event (1 to 0 transition) on bit T1 (P3.5) the CPU quit the Hard Power Down mode when bit $\overline{T1}$ (P3.5) go high or when reset is activated.
RPD	PCON.5 (83C154 and 83C154D only)	Recover from idle or Power Down bit. When 0 RPD has no effect. When 1, RPD permits to exit from idle or Power Down with any non enabled interrupt source (except timer 2). In this case the program start at the next address. When interrupt is enabled the appropriate interrupt routine is serviced.
X	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activates idle mode operation

Figure 41 : PCON Power Control Register.

Idle Mode

- Entry into this mode is effective when an instruction sets bit IDL = PCON.0 of register PCON (87H) to 1. In this mode only the CPU is no longer driven by the clock. However, its state before execution of the IDLE activation instruction is fully stored : the stack pointer, Program Counter, Program Status Word, Accumulator and all the other registers conserve their data during the IDLE mode. The ports maintain their data and the ALE and PSEN are at level 1.
- Exit from IDLE mode is controlled by register IE for the 80C51/C52 and register IE and bit RPD of register PCON for the 83C154 and the 83C154D.

Exit from Idle Mode on the 80C51 and 80C52

- There are two possibilities for exiting this mode : by interrupt or by clearing the circuit (reset).
- When an interrupt is activated, bit IDL is set to zero and the interrupt is serviced. Return to the main program is effective as soon as instruction RETI has been executed. The next instruction to be executed is that immediately following the IDLE activation instruction.
- The flag bits GFO and GF1 can be used to give an indication if an interrupt occured during normal operation or during an IDLE. For example, an instruction that activates IDLE can also set one or both flag bits. When IDLE is terminated by an interrupt, the interrupt service routine can examine the flag bits.
- The other way of quitting IDLE mode is by resetting via the RST input. The oscillator operates freely, the reset execution time is only 24 clock periods. The reset redefines all the SFRs but not the internal RAM.

Exit from Idle Mode on the 83C154 and 83C154D

- Exit from IDLE mode on the 80C154 series is controlled by register IE and bit RPD (PCON.5) of register PCON. For RPD = 0, exit conditions are identical to those of the 80C51/C52. For RPD = 1, whether or not the interrupts are enabled, an interrupt request terminates IDLE mode execution. If IE = 0 and RPD = 1, the program counter with the instruction following the IDLE activation instruction. If IE = 1 and RPD = 1, the program executes the interrupt. If no interrupt request is made while the mode is activated, only a reset via the RST input can terminate this mode.

ENTRY CONDITIONS		EXIT CONDITIONS		
IDLE	IDL	RPD	INTERRUPTS	RST
SOFTWARE	1	0	If authorized	yes
	1	1	Authorized or not	yes

Figure 42 : IDLE Mode Operation.

Power-down Mode

- In this mode, the oscillator is turned off and all the functions that were driven by the oscillator are frozen. However, the internal RAM, the special SFR registers and the ports maintain their data throughout operation in POWER-DOWN mode and during this time, signals ALE and PSEN are configured in the low state.

80C51/C52

- Entry into this mode is effective when an instruction writes a 1 in bit PD = PCON.1 of register PCON (87H).
- The only way to exit from this mode is to activate a reset via the RST input. This reset reconfigures the special SFR registers and the ports but not the internal RAM.

83C154/83C154D

- Unlike the 80C51 or the 80C52, TIMERs 0 and 1 and the UART can operate if an external clock is connected to one of the inputs T0 or T1.

Control of this mode can be done by :

- software by bits RPD, PD and register IE,

or by

- hardware by bit HPD.

Hardware Control

HPD = 1, enables this mode to be controlled by means of an external signal connected to T1. The trailing edge of this signal activates the POWER-DOWN mode as soon as the current instruction has been terminated. The leading edge of this same signal or a reset enable exit from this mode. Interrupt requests, even if enabled, do not permit exit from the mode.

Software Control

- Entry into the mode is effective when an instruction writes a 1 in bit PD = PCON.1 of register PCON (87H). Exit from the mode is controlled by bit RDP of register PCON and register IE.
- If RPD = 0 and if the interrupts are enabled or RPD = 1 and the interrupts are not enabled, an interrupt request terminates the mode.
- If this mode is terminated by an enabled interrupt, the next instruction to be executed an LCALL to the relevant interrupt routine. If the mode is terminated by an interrupt that is not enabled and RPD = 1, the next instruction to be executed is that immediatly following the power-down activation instruction. The exit-time from the mode depends on the oscillator's start-up time and the frequency. Exit from the mode does not modify the data of the internal RAM, the special SFR registers and the ports.
- If no interrupt request is made, or if RPD = IE = 0 (interrupts not enabled), the POWER-DOWN mode can only be terminated by a reset. This operation reconfigures the special SFR registers and the ports, but not the internal RAM.

Software and Hardware Control

- This mode can be controlled by mixing software and hardware commands.
  Entry to the mode can made either by setting bit PD to 1 or by setting bit HPD to 1 and presenting a trailing edge on T1.
- Exit from this mode is effective if the software and hardware end-of-mode conditions are met : a leading edge on T1 and an interrupt request. If these conditions are not satisfied, only a reset can terminate the mode.

Figure 43 summarizes the different types of operation of this mode.

ENTRY CONDITIONS				OUTPUT CONDITIONS			
POWER-DOWN	HPD	PD	T1	T1	RPD	INTERRUPTS	RST
SOFTWARE	0	1	X	X	0	If authorized	Yes
	0	1	X	X	1	Authorized or not	Yes
HARDWARE	1	0		X	X		Yes
SOFTWARE and HARDWARE	1	1			0	If authorized	Yes
	1	1			1	Authorized or not	Yes

X = without action.

Figure 43 : Software and Hardware Operation.

Voltage Reduction in Power-down Mode

- In the POWER-DOWN mode of operation, Vcc can be reduced as low as 2 V. Care must be taken, however, to ensure that Vcc is not reduced before the POWER-DOWN mode is invoked, and that Vcc is restored to its normal operating level, before the POWER-DOWN mode is terminated. The reset that terminates POWER-DOWN also frees the oscillator. The reset should not be activated before Vcc is restored to its normal operating level, and must be held active long enough to allow the oscillator to restart and stabilize (normally less than 10 msec).
- The utilization of the interrupts, the TIMERs and UART in POWER-DOWN mode is only guaranteed within the limit of the Vcc specifications.

Table 5 shows the state of the signals during POWER-DOWN and IDLE mode.

MODE	PROGRAM MEMORY	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Port Data	Port Data	Port Data	Port Data
Idle	External	1	1	Floating	Port Data	Address	Port Data
Power Down	Internal	0	0	Port Data	Port Data	Port Data	Port Data
Power Down	External	0	0	Floating	Port Data	Port Data	Port Data

Table 5 : Status of the External Pins during Idle and Power Down Modes.

OSCILLATOR CHARACTERISTICS

The oscillator is integrated in the microcontroller and consists of an inverting amplifier of which the input is XTAL1 and the output is XTAL2 (figure 44). A quartz crystal or a ceramic resonator (parallel resonance) must be used.

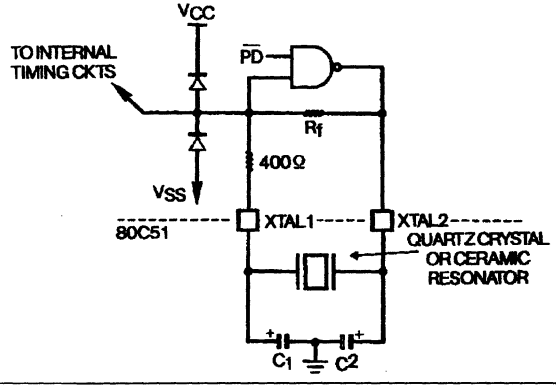


Figure 44 : Oscillator Utilization Configuration.

The MHS C51 family is able to turn off its oscillator under software control (by writing a 1 to the PD bit in PCON). In the MHS C51 family the internal clocking circuitry is driven by the signal at XTAL1.

The feedback resistor R in figure 45 consists of parallele n- and p-channel FETs controlled by the PD bit, such that R is opened when PD = 1. The diodes D1 and D2 which act as clamps to Vcc and Vss, are parasitic to the R FETs.

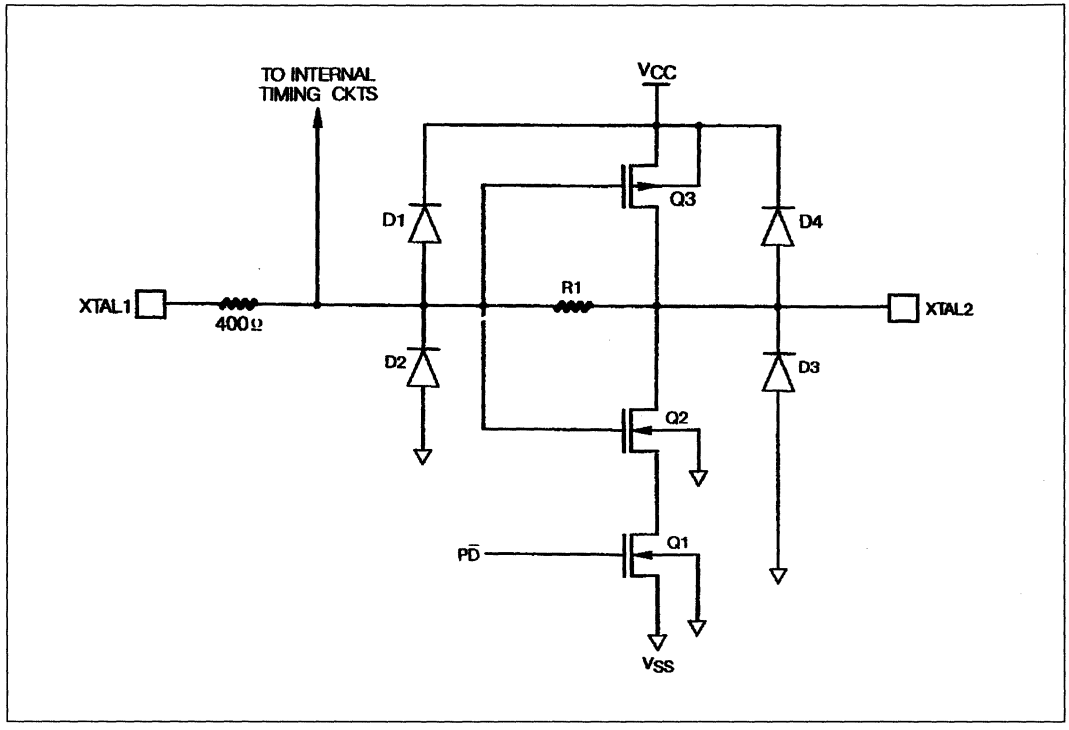
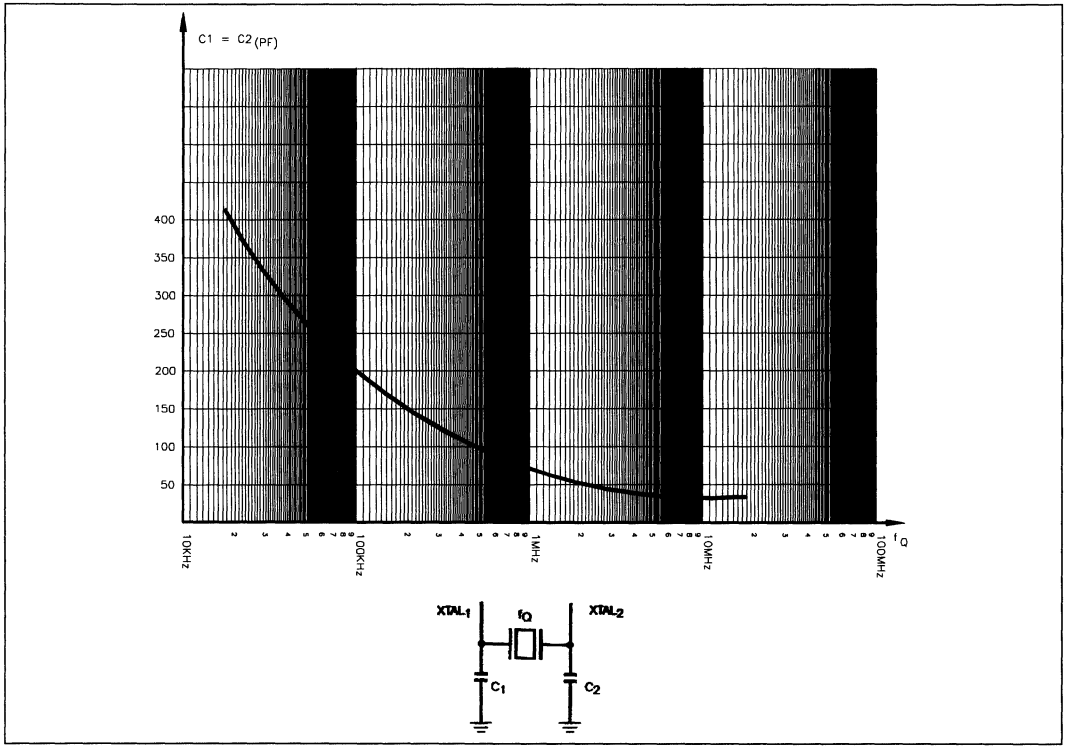


Figure 45 : Oscillator Circuit Diagram.

MHS C51 microcontrollers have a wide operating range as, depending on the version, they operate from 0 to 16 MHz. Consequently, the value of capacitors C1 and C2 is determined by the nomograph below.



Nomograph giving the value of C1 and C2 according to the frequency.

If an external circuit is to be driven by the MHS C51 microcontroller's clock, it must be connected to input XTAL1, in which case XTAL2 is floating (Figure 46).

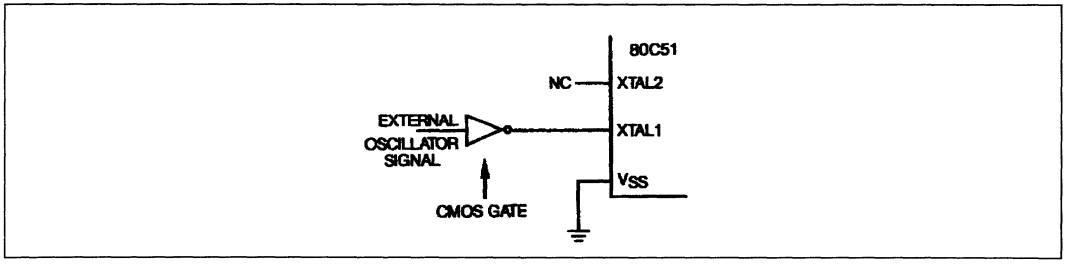


Figure 46 : Driving the MHS C51 Parts with an External Clock Source.

INTERNAL TIMING

Figures 47 through 50 show when the various strobe and port signals are clocked internally. The figures do not show rise and fall times of the signals, nor do they show propagation delays between the XTAL2 signal and the events at other pins.

Rise and fall times are dependent on the external loading that each pin must drive. They are often taken to be something in the neighbourhood of 10 nsec, measured between 0.8 V and 2.0 V.

Propagation delays are different for different pins. For a given pin they vary with pin loading, temperature, VCC and manufacturing lot. If the XTAL2 waveform is taken as the timing reference, propagation delays may vary from 25 to 125 nsec.

The CA Timings section of the data sheets do not reference any timing to the XTAL2 waveform. Rather, they relate the critical edges of control and input signals to each other. The timings published in the data sheet include the effects of propagation delays under the specified test conditions.

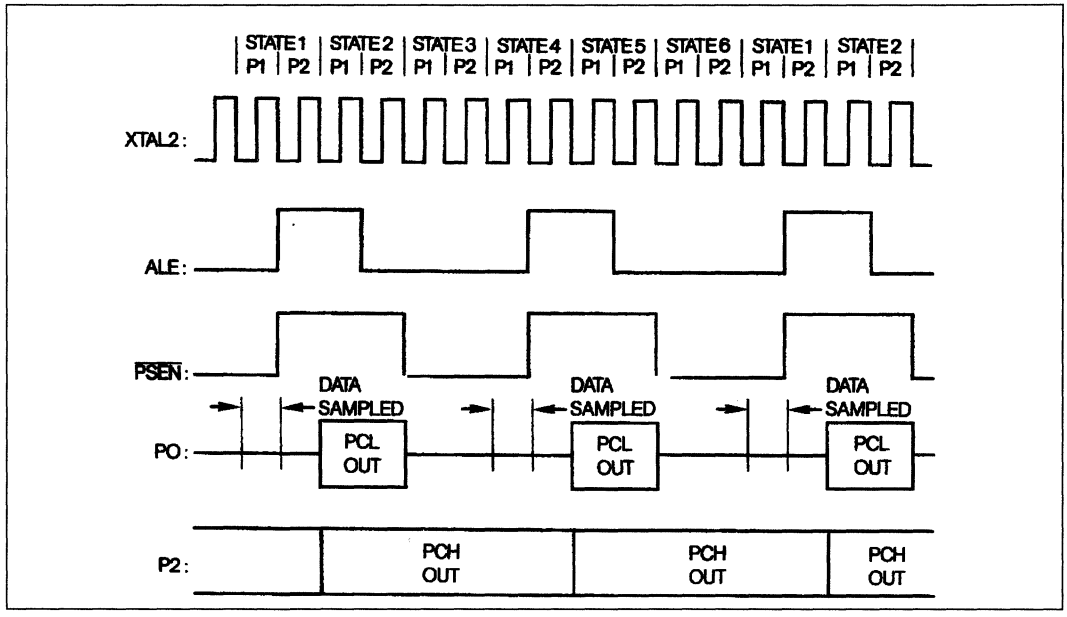


Figure 47 : External Program Memory Fetches.

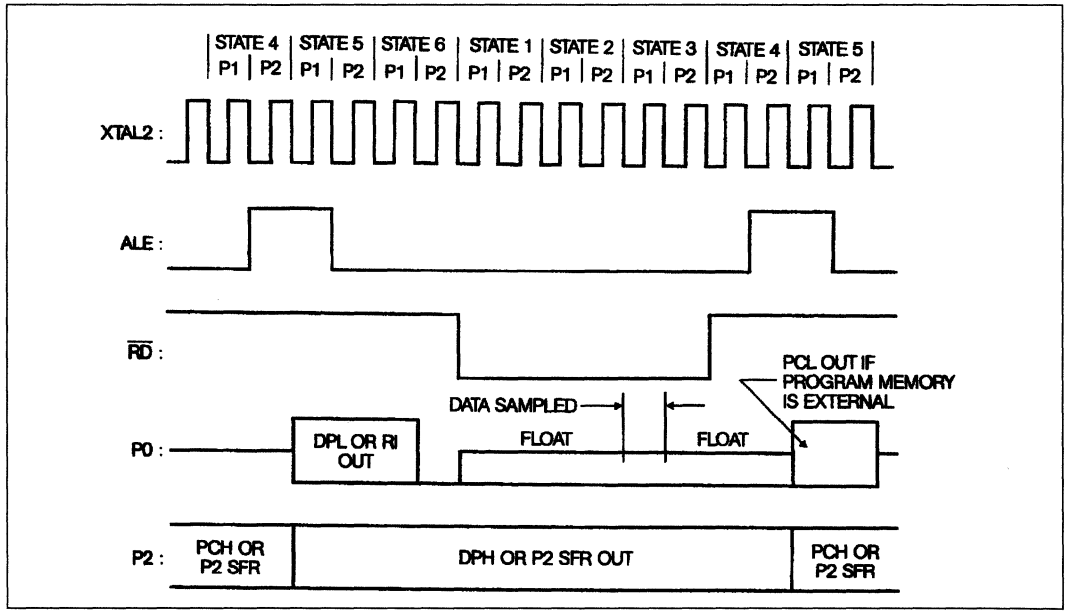


Figure 48 : External Data Memory Read Cycle.

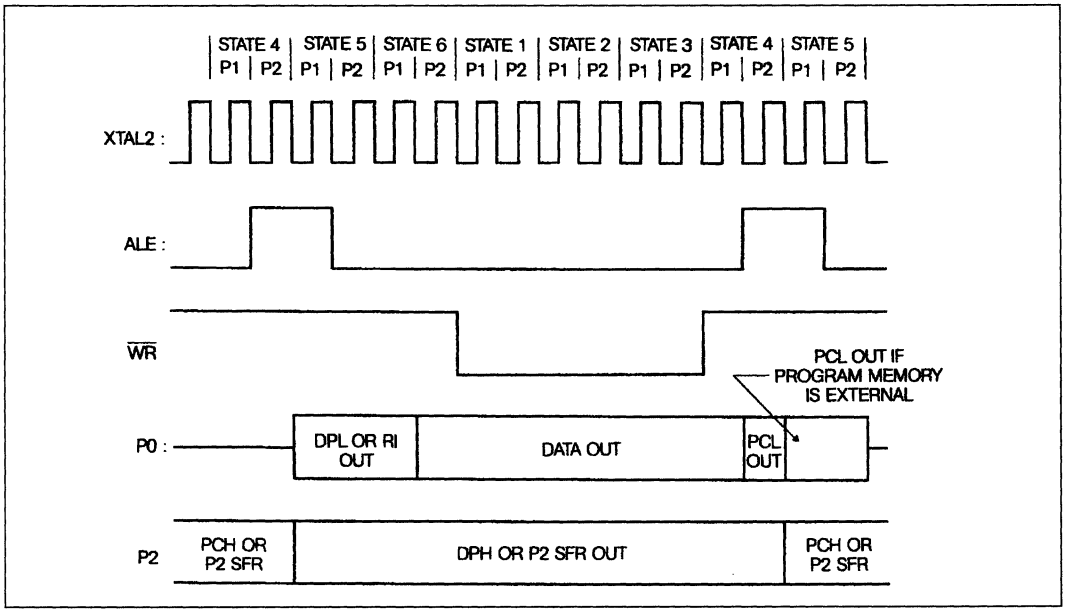


Figure 49 : External Data Memory Write Cycle.

4

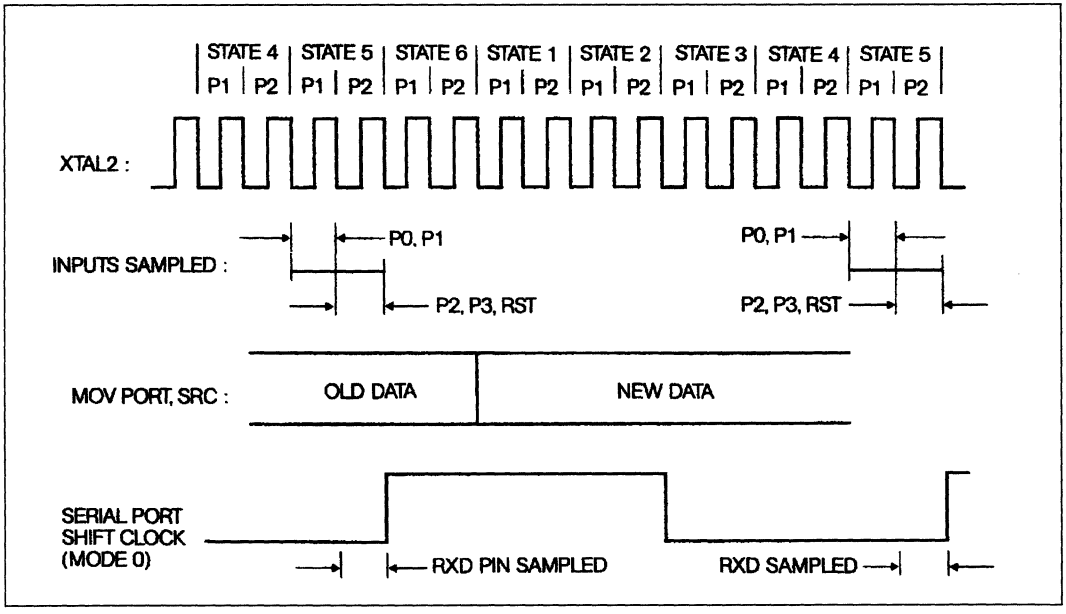


Figure 50 : Port Operation.

MHS C51 PIN DESCRIPTIONS

VCC : Supply voltage.

VSS : Circuit ground potential.

Port 0 : Port 0 is an 8-bit open drain bidirectional I/O port. As an open drain output port it can sink 8 LS TTL loads. Port 0 pins that have 1s written to them float, and in that state will function as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external memory. In this application it uses strong internal pullups when emitting 1's. Port 0 also emits code bytes during program verification. In that application, external pullups are required.

Port 1 : Port 1 is an 8-bit bidirectional I/O port with internal pullups. The port 1 output buffers can sink/source 3 LSTTL loads. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups.

In the 80C52, 83C154 and 83C154D, pins P1.0 and P1.1 also serve the alternate functions of T2 and T2EX. T2 is the Timer 2 external input. T2EX is the input through which a Timer 2 "capture" is triggered.

Port 2 : Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source 3 LSTTL loads. Port 2 emits the high-order address byte during accesses to external memory that use 16-bit addresses. In this application it uses the strong internal pullups when emitting 1s. Port 2 also receives the high-order address and control bits during program verification in the MHS C51 Family.

Port 3 : Port 3 is an 8-bit bidirectional I/O port with internal pullups. It also serves the functions of various special features of the MHS C51 Family, as listed below :

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

The Port 3 output buffers can source/sink 3 LSTTL loads.

RST : Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE : Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. ALE is emitted at a constant rate of 1/6 of the oscillator frequency, for external timing or clocking purposes, even when there are no accesses to external memory. (However, one ALE pulse is skipped during each access to external **Data** memory).

$\overline{\text{PSEN}}$: Program Store Enable is the read strobe to external Program Memory. When the device is executing out of external Program Memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle (except that two $\overline{\text{PSEN}}$ activations are skipped during accesses to external **Data** Memory). $\overline{\text{PSEN}}$ is not activated when the device is executing out of Internal Program Memory.

$\overline{\text{EA}}$: When $\overline{\text{EA}}$ is held high the CPU executes out of Internal Program Memory (unless the Program Counter exceeds OFFFH in the 80C51, or 1FFFH in the 80C52, or 3FFFH in the 83C154 or 7FFFH in the 83C154D). Holding $\overline{\text{EA}}$ low forces the CPU to execute out of external memory regardless of the Program Counter value. In the 80C31, 80C32 and 80C154, $\overline{\text{EA}}$ must be extremely wired low.

XTAL1 : Input to the inverting oscillator amplifier.

XTAL2 : Output from the inverting oscillator amplifier.

MHS C51 PROGRAMMER'S GUIDE AND INSTRUCTION SET

5

MHS C51 PROGRAMMER'S GUIDE AND INSTRUCTION SET

The information presented in this chapter is collected from the previous MHS C51 chapter of this book. The material has been selected and rearranged to form a quick and convenient reference for the programmers of the MHS C51.

The following list should make it easier to find a subject in this chapter.

MEMORY ORGANIZATION

PROGRAM MEMORY

The MHS C51 Microcontroller Family has separate address spaces for program Memory and Data Memory. The program memory can be up to 64 K bytes long. The lower 4 K for the 80C51 (8 K for the 80C52, 16 K for the 83 C154 and 32 K for the 83C154D) may reside on chip.
Figure 1 to 4 show a map of 80C51, 80C52, 83C154 and 83C154D program memory.

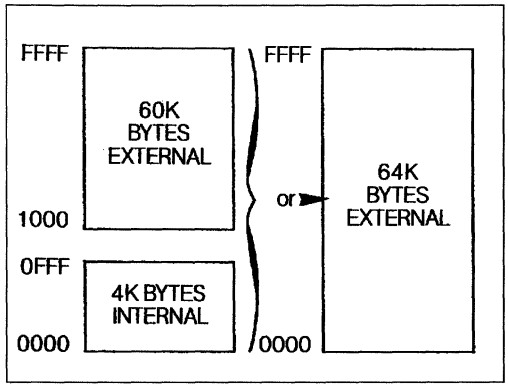


Figure 1 : The 80C51 Program Memory.

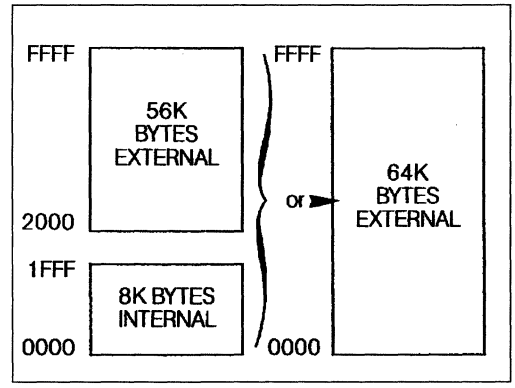


Figure 2 : The 80C52 Program Memory.

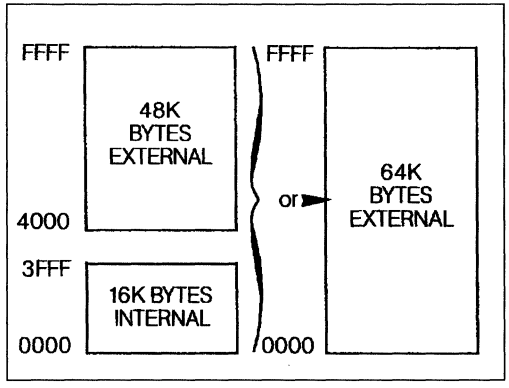


Figure 3 : The 83C154 Program Memory.

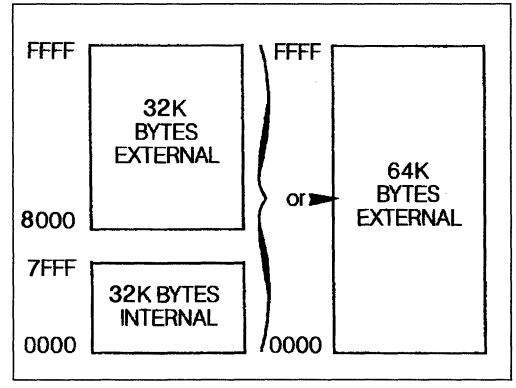


Figure 4 : The 83C154D Program Memory.

DATA MEMORY

The MHS C51 Microcontroller Family can address up to 64 K bytes of Data Memory to the chip. The " MOVX " instruction is used to access the external data memory (refer to the MHS C51 instruction set, in this chapter, for detailed description of instructions).

The 80C51 has 128 bytes of on-chip-RAM (256 bytes in the 80C52 and 83C154/83C154D) plus a number of Special Function Registers (SFR). The lower 128 bytes of RAM can be accessed either by direct addressing (MOV data addr). or by indirect addressing (MOV @Ri). Figure 5 and 6 show the 80C51, 80C52 and 83C154/83C154D Data Memory organization.

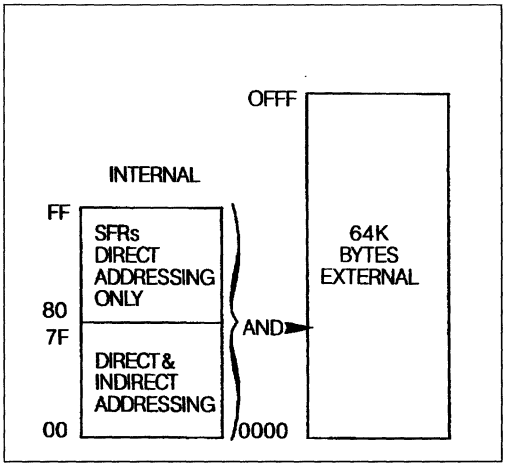


Figure 5 : The 80C51 Data Memory Organisation.

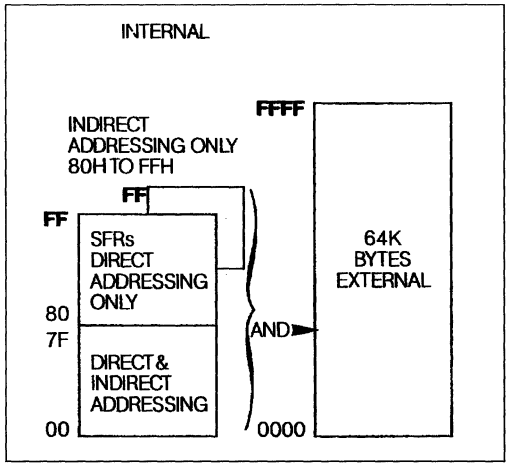


Figure 6 : The 80C52 and 83C154/83C154D Data Memory Organisation.

INDIRECT ADDRESS AREA :

Note that in Figure 6 - the SFRs and the indirect address RAM have the same addresses (80H-OFFH). Nevertheless, they are two separate areas and are accessed in two different ways.
For example the instruction

MOV 80H, #0AAH

writes 0AAH to Port 0 which is one of the SFRs and the instruction

MOV R0, # 80H
MOV @ R0, # 0BBH

writes 0BBH in location 80H of the data RAM. Thus, after execution of both of the above instructions Port 0 will contain 0AAH and location 80 of the RAM will contain 0BBH.

DIRECT AND INDIRECT ADDRESS AREA :

The 128 bytes of RAM which can be accessed by both direct and indirect addressing can be divided into 3 segments as listed below and shown in figure 7.

1. Register Banks 0.3 : Locations 0 through 1FH (32 bytes). ASM-51 and the device after reset default to register bank 0. To use the other register banks the user must select them in the software. Each register bank contains 8 one-byte registers, 0 through 7.
Reset initializes the Stack Pointer to location 07H and it is incremented once to start from location 08H which is the first register (R0) of the second register bank. Thus, in order to use more than one register bank, the SP should be initialized to a different location of the RAM where it is not used for data storage (ie, higher part of the RAM).

2. Bit Addressable Area : 16 bytes have been assigned for this segment, 20H-2FH. Each one of the 128 bits of this segment can be directly addressed (0-7FH).
The bits can be referred to in two ways both of which are acceptable by the ASM-51. One way is to refer to their addresses, ie, 0 to 7FH. The other way is with reference to bytes 20H to 2FH. Thus, bits 0-7 can also be referred to as bits 20.0-20.7, and bits 8-FH are the same as 21.0-21.7 and so on.
Each of the 16 bytes in this segment can also be addresses as a byte.

3. Scratch Pad Area : Bytes 30H through 7FH are available to user as data RAM. However, if the stack pointer has been initialized to this area, enough number of bytes should be left aside to prevent SP data destruction.

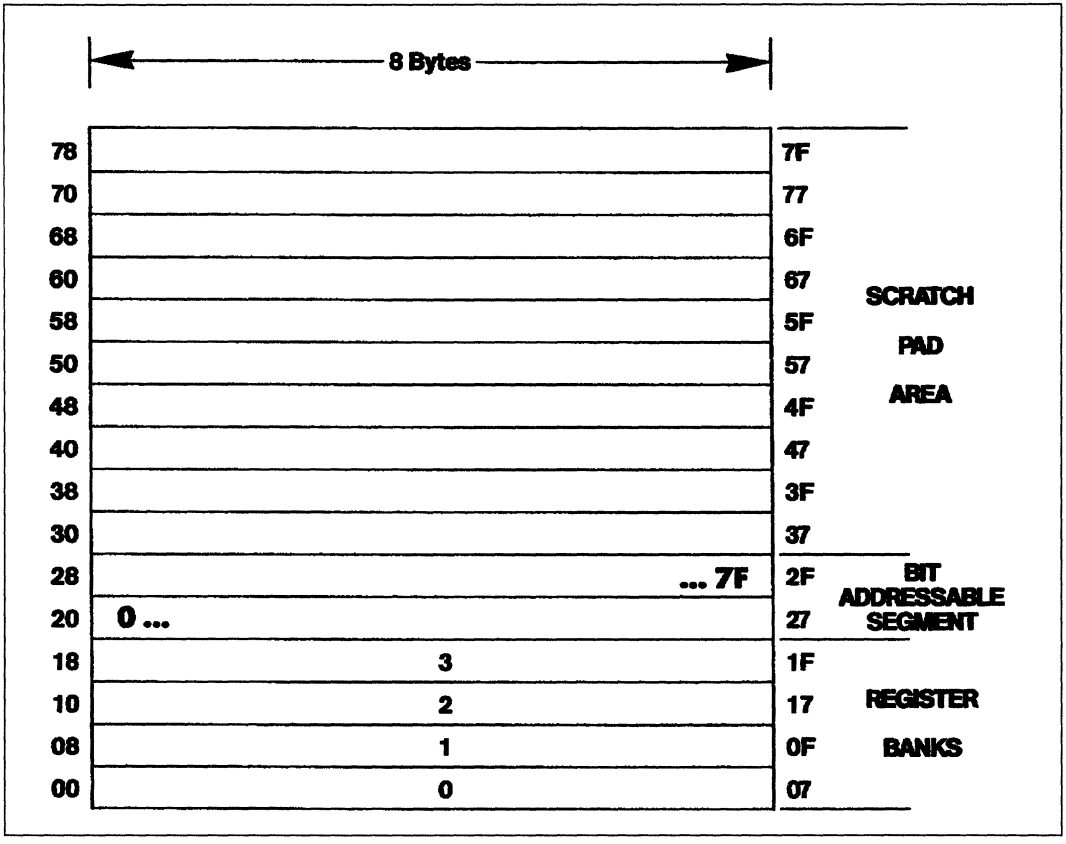


Figure 7 shows the different segments of the on-chip RAM.
128 Bytes of RAM Direct and Indirect Addressable.

5

SPECIAL FUNCTION REGISTERS :

Table 1 contains a list of all the SFRs and their addresses.
Comparing table 1 and figure 7 shows that all of the SFRs that are byte and bit addressable are located on the first column of the diagram in figure 7.

SYMBOL	NAME	ADDRESS
*ACC	Accumulator	0E0H
*B	B Register	0F0H
*PSW	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer 2 Bytes	
DPL	Low Byte	82H
DPH	High Byte	83H
*P0	Port 0	80H
*P1	Port 1	90H
*P2	Port 2	0A0H
*P3	Port 3	0B0H
*IP	Interrupt Priority Control	0B8H
*IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
*TCON	Timer/Counter Control	88H
*+T2CON	Timer/Counter 2 Control	0C8H
TH0	Timer/Counter 0 High Byte	8CH
TL0	Timer/Counter 0 Low Byte	8AH
TH1	Timer/Counter 1 High Byte	8DH
TL1	Timer/Counter 1 Low Byte	8BH
+TH2	Timer/Conuter 1 High Byte	0CDH
+TL2	Timer/Counter 2 Low Byte	0CCH
+RCAP2H	T/C 2 Capture Reg. High Byte	0CBH
+RCAP2L	T/C 2 Capture Reg. Low Byte	0CAH
*SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H
*IOCON (1)	IO Control	F8H

+ 80C52 and 83C154/83C154D only * bit addressable
(1) 83C154/83C154D only

Table 1.

SFR MEMORY MAP

	8 Bytes								
F8	IOCON								FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8	T2CON		RCAP2L	RCAP2H	TL2	TH2			CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87

↑ bit addressable

Figure 8.

WHAT DO THE SFRs CONTAIN JUST AFTER POWER-ON RESET ?

Table 2 lists the contents of each SFR after a power-on reset or a hardware reset.

REGISTER	VALUE IN BINARY
*ACC	0000 0000
*B	0000 0000
*PSW	0000 0000
SP	0000 0111
DPTR	0000 0000
P0	1111 1111
*P1	1111 1111
*P2	1111 1111
*P3	1111 1111
*IP	XXX0 0000 80C51 XX00 0000 80C52 0X00 0000 83C154/C154D
*IE	0XX0 0000 80C51 0X00 0000 83C154/C154D and 80C52
TMOD	0000 0000

REGISTER	VALUE IN BINARY
*TCON	0000 0000
+*T2CON	0000 0000
TH0	0000 0000
TL0	0000 0000
TH1	0000 0000
TL1	0000 0000
+ TH2	0000 0000
+ TL2	0000 0000
+ RCAP2L	0000 0000
+ RCAP2H	0000 0000
*SCON	0000 0000
SBUF	indeterminate
PCON	0XXX 0000 80C51 and 80C52 000X 0000 83C154 and 83C154D
-*IOCON	0000 0000

* : bit addressable.
+ : 80C52, 83C154 and 83C154D only.
- : 83C154 and 83C154D only.
X : Undefined.

Table 2 : Contents of the SRFs after reset.

These SFRs that have their bits assigned for various functions are listed in this section. A brief description of each bit is provided for quick reference. For more detailed information refer to the Architecture chapter of this book.

PSW : PROGRAM STATUS WORD. BIT ADDRESSABLE.

CY	AC	F0	RS1	RS0	OV	F1	P

CY	PSW.7	Carry Flag.
AC	PSW.6	Auxiliary Carry Flag.
F0	PSW.5	Flag 0 available to the user for general purpose.
RS1	PSW.4	Register Bank selector bit 1 (SEE NOTE).
RS0	PSW.3	Register Bank selector bit 0 (SEE NOTE).
OV	PSW.2	Overflow Flag.
F1	PSW.1	Flag F1 available to the user for general purpose.
P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of " 1 " bits in the accumulator.

Note :

The value presented by RS0 and RS1 selects the corresponding register bank.

RS1	RS0	REGISTER BANK	ADDRESS
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

* User software should not write 1s to reserved bits. These bits may be used in future MHS C51 products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

PCON : POWER CONTROL REGISTER. NOT BIT ADDRESSABLE.

SMOD	HPD	RPD	–	GF1	GF0	PD	IDL

SMOD	PCON.7	Double baud rate bit. If SMOD = 1, the baud rate is doubled when the serial part is used in mode 1, 2 and 3.
HPD	PCON.6	Hard Power Down. (83C154/83C154D only). The falling/rising edge of a signal connected on pin P3.5 Starts/Stops the Power-Down mode. A reset can also stop this mode.
RPD	PCON.5	Recover Power Down bit. (83C154/83C154D only). It's used to cancel a Power-Down/IDLE mode. If it's set, an interrupt (enable or disable) can cancel this mode. A reset can also stop this mode (see Note 1).
-	PCON.4	Not implemented, reserved for futur used*
GF1	PCON.3	General purpose bit.
GF0	PCON.2	General purpose bit.
PD	PCON.1	Power Down bit. If set, the oscillator is stopped. A reset or an interrupt (83C154 and 83C154D only) can cancel this mode (Note 1).
IDL	PCON.0	IDLE bit. If set the activity CPU is stopped. A reset or an interrupt can cancel this mode (See Note 1).

* User software should not write 1s to reserved bits. These bits may be used in future MHS C51 products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

Note 1 (83C154/83C154D only) :

- if RPD = 0 and if an interrupt cancels the mode Power-Down/IDLE, the next instruction to execute is a LCALL at the interrupt routine.
- RPD = 1 - if interrupt request is enable the next instruction to execute is a LCALL at the interrupt routine.
  - if interrupt request is disable, the program continue with the instruction immediately after the Power-Down/Idle instruction.

INTERRUPTS :

In order to use any of the interrupts in the MHS C51, the following three steps must be taken.
1. Set the EA (enable all) bit in the IE register to 1.
2. Set the corresponding individual interrupt enable bit in the IE register to 1.
3. Begin the Interrupt service routine at the corresponding Vector Address of that interrupt. See Table below.

INTERRUPT SOURCE	VECTOR ADDRESS
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI & TI	0023H
TF2 & EXF2	002BH

In addition, for external interrupts, pins INT0 and INT1 (P3.2 and P3.3) must be set to 1, and depending on whether the interrupt is to be level or transition activated, bits IT0 or IT1 in the TCON register may need to be set to 1.

ITX = 0 level activated
ITX = 1 transition activated

IE : INTERRUPT ENABLE REGISTER BIT ADDRESSABLE.

If the bit is 0, the corresponding interrupt is disabled. If the bit is 1, the corresponding interrupt is enabled.

EA	–	ET2	ES	ET1	EX1	ET0	EX0

EA	IE.7	Disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, interrupt source is individually enable or disabled by setting or clearing its enable bit.
-	IE.6	Not implemented, reserved for future use*.
ET2	IE.5	Enable or disable the Timer 2 overflow or capture interrupt (80C52, 83C154 and 83C154D only).
ES	IE.4	Enable or disable the Serial port interrupt.
ET1	IE.3	Enable or disable the Timer 1 overflow interrupt.
EX1	IE.2	Enable or disable External interrupt 1.
ET0	IE.1	Enable or disable the Timer 0 overflow interrupt.
EX0	IE.0	Enable or disable External Interrupt 0.

* User software should not write 1s to reserved bits. These bits may be used in future MHS C51 products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

ASSIGNING HIGHER PRIORITY TO ONE MORE INTERRUPTS

In order to assign higher priority to an interrupt the corresponding bit in the IP register must be set to 1. Remember that while an interrupt service is in progress, it cannot be interrupted by a lower or same level interrupt.

PRIORITY WITHIN LEVEL :

Priority within level is only to resolve simultaneous requests of the same priority level. From high to low, interrupt sources are listed below :

IE0
TF0
IE1
TF1
RI or TI
TF2 or EXF2

IP : INTERRUPT PRIORITY REGISTER. BIT ADDRESSABLE.

If the bit is 0, the corresponding interrupt has a lower priority and if the bit is the corresponding interrupt has a higher priority.

PCT	–	PT2	PS	PT1	PX1	PT0	PX0

PCT	IP.7	Defines the same priority level for all the source interrupt (83C154 and 83C154D only).
-	IP.6	Not implemented, reserved for future use*.
PT2	IP.5	Defines the Timer 2 interrupt priority level (80C52, 83C154 and 83C154D only).
PS	IP.4	Defines the Serial Port interrupt priority level.
PT1	IP.3	Defines the Timer 1 Interrupt priority level.
PX1	IP.2	Defines External Interrupt priority level.
PT0	IP.1	Defines the Timer 0 interrupt priority level.
PX0	IP.0	Defines the External Interrupt 0 priority level.

* User software should not write 1s to reserved bits. These bits may be used in future MHS C51 products to invoke new features. In that case, the reset or inactive value of the now bit will be 0, and its active value will be 1.

5

IOCON. (83C154 and 83C154D only). Input/Output Control Register.

WDT	T32	SERR	IZC	P3HZ	P2HZ	P1HZ	ALF

WDT	IOCON.7	Watch Dog Timer bit. Set when Timer 1 is overflow (TF = 1). The CPU is reset and the program is executed from address 0.
T32	IOCON.6	Timer 32 bits. The Timer 1 and Timer 0 are connected together to form a 32 bits Timer/Counter. If C/TO = 0, it's a Timer. If C/TO = 1, it's a counter.
SERR	IOCON.5	Serial Port Reception Error flag. Set when an overrun on frame error is received.
IZC	IOCON.4	Set/Cleared by software to select 100/10 K pull up resistance for Port 1, 2 and 3.
P3HZ	IOCON.3	When Set, Port 3 becomes a tri-state input. When cleared, the pull-up resistance value is selected by IZC.
P2HZ	IOCON.2	When Set, Port 2 becomes a tri-state input. When cleared, the pull-up resistance value is selected by IZC.
P1HZ	IOCON.1	When Set, Port 1 becomes a tri-state input. When cleared, the pull-up resistance value is selected by IZC.
ALF	IOCON.0	All Port tri-state. When Set and CPU in Power-Down mode, port 1, 2 and 3 are tri-state.

TCON : TIMER/COUNTER CONTROL REGISTER. BIT ADDRESSABLE.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1	TCON.7	Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
TR1	TCON.6	Timer 1 run control bit. Set/cleared by software to turn Timer/Counter ON/OFF.
TF0	TCON.5	Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.
TR0	TCON.4	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.
IE1	TCON.3	External Interrupt 1 edge flag. Set by hardware when External interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT1	TCON.2	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/flow level triggered External Interrupt.
IE0	TCON.1	External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.
IT0	TCON.0	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

TMOD : TIMER/COUNTER MODE CONTROL REGISTER. NOT BIT ADDRESSABLE.

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
TIMER 1				TIMER 0			

GATE	When TRx (in TCON) is set and GATE = 1, TIMER/COUNTERx will run only while INTx pin is high (hardware control). When GATE = 0, TIMER/COUNTERx will run only while TRx = 1 (software control).
C/$\overline{T}$	Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).
M1	Mode selector bit (NOTE 1).
M0	Mode selector bit (NOTE 1).

NOTE 1 :

M1	M0		OPERATING MODE
0	0	0	13-bit Timer
0	1	1	16-bit Timer/Counter
1	0	2	8-bit Auto-Reload Timer/Counter
1	1	3	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits, TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.
1	1	3	(Timer 1) Timer/Counter 1 stopped.

TIMER SET-UP

Tables 3 through 6 give some values for TMOD which can be used to set up Timer 0 in different modes.

It is assumed that only one timer is being used at a time. It is desired to run Timers 0 and 1 simultaneously, in any mode, the value that in TMOD for Timer 0 must be ORed with the value shown for Timer 1 (Tables 5 and 6).

For example, if it is desired to run Timer 0 in mode 1 GATE (external control) and Timer 1 in mode 2 COUNTER, then the value must be loaded into TMOD is 69H (09H from Table 3 ORed with 60H from Table 6).

Moreover, it is assumed that the user, at this point, is not ready to turn the timers on and will do that a different point in the program by setting bit TRx (in TCON) to 1.

TIMER/COUNTER 0

MODE	TIMER 0 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	00H	08H
1	16-bit Timer	01H	09H
2	8-bit Auto-Reload	02H	0AH
3	Two 8-bit Timers	03H	0BH

As a Timer : Table 3

MODE	TIMER 0 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	04H	0CH
1	16-bit Timer	05H	0DH
2	8-bit Auto-Reload	06H	0EH
3	one 8-bit counter	07H	0FH

As a Counter : Table 4

Notes : 1. The Timer is turned ON/OFF by setting/clearing bit TR0 in the software.
2. The Timer is turned ON/OFF by the 1 to 0 transition on $\overline{INT0}$ (P3.2) when TR0 = 1 (hardware control).

TIMER/COUNTER 1

MODE	TIMER 1 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	00H	80H
1	16-bit Timer	10H	90H
2	8-bit Auto-Reload	20H	A0H
3	does not run	30H	B0H

As a Timer : Table 5

MODE	COUNTER 1 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	40H	C0H
1	16-bit Timer	50H	D0H
2	8-bit Auto-Reload	60H	E0H
3	not available	–	–

As a Counter : Table 6

Notes : 1. The Timer is turned ON/OFF by setting/clearing bit TR1 in the software.
2. The Timer is turned ON/OFF by the 1 to 0 transition on $\overline{INT1}$ (P3.2) when TR1 = 1 (hardware control).

T2CON : TIMER/COUNTER 2 CONTROL REGISTER. BIT ADDRESSABLE (80C52, 83C154 and 83C154D only)

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2

TF2 T2CON.7 Timer 2 overflow flag set by hardware and cleared by software. TF2 cannot be set when either RCLK = 1 or CLK = 1

EXF2 T2CON.6 Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX, and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software.

RCLK T2CON.5 Receive clock flag. When set, causes the Serial Port to use Timer 2 overflow pulses for its receive clock in modes 1 & 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.

TCLK T2CON.4 Transmit clock flag. When set, causes the Serial Port use Timer 2 overflow pulses for its transmit clock in modes 1 & 3, TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.

EXEN2 T2CON.3 Timer 2 external enable flag. When set, allows a capture or reload to occur as a result of negative transition on T2EX if Timer 2 is not being used to clock the Serial Port. EXEN2 = 0 causes Timer 2 to ignore events as T2EX.

TR2 T2CON.2 Software START/STOP control for Timer 2. A logic 1 starts the Timer.

C/$\overline{T2}$ T2CON.1 Timer or Counter select.

CP/$\overline{RL2}$ T2CON.0 Capture/Reload flag. When set, captures will occur on negative transitions at T2EX if EXEN2 = 1. When cleared, Auto-Reloads will occur either with Timer2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the Timer is forced to Auto-Reload on Timer 2 overflow.

TIMER/COUNTER 2 SET-UP

Except for the baud rate generator mode, the values given for T2CON do not include the setting of the TR2 bit. Therefore, bit TR2 must be set, separately, to turn the Timer on.

5

MODE	T2CON	
	INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
16-bit Auto-Reload	00H	08H
16-bit Capture	01H	09H
BAUD rate generator receive & transmit same baud rate	34H	36H
receive only	24H	26H
transmit only	14H	16H

As a Timer : Table 7

MODE	TMOD	
	INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
16-bit Auto-Reload	02H	0AH
16-bit Capture	03H	0BH

As a Counter : Table 8

Notes : 1. Capture/Reload occurs only Timer/Counter overflow.
2. Capture/Reload occurs on Timer/Counter overflow and a 1 to 0 transition on T2EX (P1.1) pin except when Timer 2 is used in the baud rate generating mode.

SCON / SERIAL PORT CONTROL REGISTER. BIT ADDRESSABLE.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SCON.7	Serial Port mode specifier (NOTE 1).
SM1	SCON.6	Serial Port mode specifier (NOTE 1).
SM2	SCON.5	Enables the multiprocessor communication feature in mode 2 & 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0 (See table 9).
REN	SCON.4	Set/Cleared by software to Enable/Disable reception.
TB8	SCON.3	The 9th bit that will be transmitted in modes 2 & 3. Set/Cleared by software.
RB8	SCON.2	In modes 2 & 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI	SCON.1	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes. Must be cleared by software.
RI	SCON.0	Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or half way through the stop bit time in the other modes (except see SM2). Must be cleared by software.

NOTE 1 :

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	SHIFT REGISTER	Fosc./12
0	1	1	8-Bit UART	Variable
1	0	2	9-Bit UART	Fosc./64 OR Fosc./32
1	1	3	9-Bit UART	Variable

SERIAL PORT SET-UP : Table 9

MODE	SCON	SM2 VARIATION
0	10H	Single Processor Environment (SM2 = 0)
1	50H	
2	90H	
3	D0H	
0	NA	Multiprocessor Environment (SM2 = 1)
1	70H	
2	B0H	
3	F0H	

GENERATING BAUD RATES

Serial Port in Mode 0 :

Mode 0 has a fixed baud rate which is 1/12 of oscillator frequency. To run serial port in this mode none of the Timer/Counters need to be set up. Only the SCON register needs to be defined.

$$\text{Baud Rate} = \frac{\text{Osc Freq}}{12}$$

Serial Port in Mode 1 :

Mode 1 has a variable baud rate. The baud rate can be generated by either Timer 1 or Timer 2 (80C52 and 83C154/83C154D only).

USING TIMER/COUNTER 1 TO GENERATE BAUD RATES :

For this purpose, Timer 1 is used in mode 2 (Auto-Reload). Refer to Timer Setup section of this chapter.

$$\text{Baud Rate} = \frac{\text{K x Oscillator Freq.}}{32 \times 12 \times [256 - (\text{TH1})]}$$

if SMOD = 0, then K = 1.
If SMOD = 1, then K = 2. (SMOD is the PCON register).
Most of the time the user knows the baud rate and needs to know the reload value for TH1. Therefore, the equation to calculate TH1 can be written as :

$$\text{TH1} = 256 - \frac{\text{K x Osc Freq.}}{384 \times \text{baud rate}}$$

TH1 must be integer value. Rounding off TH1 to the nearest integer may not produce the desired baud rate. In this case, the user may have to choose another crystal frequency.
Since the PCON register is not bit addressable, one way to set the bit is logical ORing the PCON register (ie, ORL PCON, #80H). The address of PCON is 87H.

USING TIMER/COUNTER 2 TO GENERATE BAUD RATES :

For this purpose, Timer 2 must be used in the baud rate generating mode. Refer to Timer 2 Setup Table in this chapter. If Timer 2 is being clocked through pin T2 (P1.0) the baud rate is :

$$\text{Baud Rate} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

And if it being clocked internally the baud rate is :

$$\text{Baud Rate} = \frac{\text{Osc Freq.}}{32 \times [65536 - (\text{RCAP2H, RCAP2L})]}$$

To obtain the reload value for RCAP2H and RCAP2L the above equation can be written as :

$$\text{RCAP2H, RCAP2L} = 65536 - \frac{\text{Osc Freq.}}{32 \times \text{Baud Rate}}$$

5

SERIAL PORT IN MODE 2 :

The baud rate is fixed in this mode and 1/32 or 1/64 of the oscillator frequency depending on the value of the SMOD bit in the PCON register.

In this mode none of the Timers are used and the clock comes from the internal phase 2 clock.

SMOD = 1, Baud Rate = 1/32 Osc Freq.

SMOD = 0, Baud Rate = 1/64 Osc Freq.

To set the SMOD bit : ORL PCON, #80H. The address of PCON is 87H.

SERIAL PORT IN MODE 3 :

The baud rate in mode 3 is variable and sets up exactly the same as in mode 1.

MHS C51 INSTRUCTION SET

Interrupt Response time : Refer to Hardware Description Chapter.

Instructions that Affect Flag Settings (1)

INSTRUC.	FLAG			INSTRUC.	FLAG		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLRC	O		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C, bit	X		
MUL	0	X		ANL C,/bit	X		
DIV	0	X		ORL C, bit		X	
DA	X			ORL C, bit		X	
RRC	X			MOV C, bit		X	
RLC	X			CJNE	X		
SETB C	1						

(1) Note that operations on SFR byte address 208 or bit addresses 209–215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

Note on Instruction set and adrressing modes :

Rn – Register R7-R0 of the currently selected Register Bank

direct – 8-bit internal data location's address. This could be an Internal Data RAM location (0-127) or a SFR (i.e., I/O port, control register, status register, etc. (128-255)).

@ Ri – 8-bit internal data RAM location (0-255) addresses indirectly through register R1 or R0.

data – 8-bit constant included in instruction.

data 16 – 16-bit constant included in instruction.

addr 16 – 16-bit destination address. Used by LCALL & LJMP. Abranch can be anywhere within the 64K-byte Program memory address space.

addr 11 – 11-bit destination address. Used by ACALL & AJMP. The branch will be within the same 2K-byte page of program memory as the first byte of the following instruction.

rel – Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditionnal jumps. Range is –128 to + 127 bytes relative to first byte of the following instruction.

bit – Direct Addressed bit in Internal Data RAM or special Function Register.

* – New operation not provided by 8048AH/8049AH

MNEMONIC	DESCRIPTION	BYTE	OSCIL. PERIOD
ARITHMETIC OPERATIONS			
ADD A, Rn	Add register to Accumulator	1	12
ADD A, direct	Add direct byte to Accumulator	2	12
ADD A, @Ri	Add indirect RAM to Accumulator	1	12
ADD A, #data	Add immediate data to Accumulator	2	12
ADDCA, Rn	Add register to Accumulator with Carry	1	12
ADDCA, direct	Add direct byte to Accumulator with Carry	2	12
ADDCA, @Ri	Add indirect RAM to Accumulator with Carry	1	12
ADDCA, #data	Add immediate data to Acc with Carry	2	12
SUBB A, Rn	Subtract Register from Acc with borrow	1	12
SUBB A, direct	Subtract direct byte from Acc with borrow	2	12
SUBB A, @Ri	Subtract indirect RAM from ACC with borrow	1	12
SUBB A, #data	Subtract immediate data from Acc with borrow	2	12
INC A	Increment Accumulator	1	12
INC Rn	Increment register	1	12
INC direct	Increment direct byte	2	12
INC @Ri	Increment direct RAM	1	12
DEC A	Decrement Accumulator	1	12
DEC Rn	Decrement Register	1	12
DEC direct	Decrement direct byte	2	12
DEC @Ri	Decrement indirect RAM	1	12

Table 10 : 80C51 Instruction Set Summary

MNEMONIC	DESCRIPTION	BYTE	OSCIL. PERIOD
ARITHMETIC OPERATIONS (continued)			
INC DPTR	Increment Data Pointer	1	24
MUL AB	Multiply A & B	1	48
DIV AB	Divide A by B	1	48
DA A	Decimal Adjust Accumulator	1	12
LOGICAL OPERATIONS			
ANL A, Rn	AND Register to Accumulator	1	12
ANL A, direct	AND direct byte to Accumulator	2	12
ANL A, @Ri	AND indirect RAM to Accumulator	1	12
ANL A, #data	AND immediate data to Accumulator	2	12
ANL direct, A	AND Accumulator to direct byte	2	12
ANL direct, #data	AND immediate data to direct byte	3	24
ORL A, Rn	OR register to Accumulator	1	12
ORL A, direct	OR direct byte to Accumulator	2	12
ORL A, @Ri	OR indirect RAM to Accumulator	1	12
ORL A, #data	OR immediate data to Accumulator	2	12
ORL direct, A	OR Accumulator to direct byte	2	12
ORL direct, #data	OR immediate data to direct byte	3	24
XRL A, Rn	Exclusive-OR register to Accumulator	1	12
XRL A, direct	Exclusive-OR direct byte to accumulator	2	12
XRL A, @Ri	Exclusive-OR indirect RAM to Accumulator	1	12
XRL A, #data	Exclusive-OR immediate data to Accumulator	2	12
XRL direct, A	Exclusive-OR Accumulator to direct byte	2	12
XRL direct, #data	Exclusive-OR immediate data to direct byte	3	24
CLR A	Clear Accumulator	1	12
CPL A	Complement Accumulator	1	12

MNEMONIC	DESCRIPTION	BYTE	OSCIL. PERIOD
LOGICAL OPERATIONS (continued)			
RL A	Rotate Accumulator Left	1	12
RLC A	Rotate Accumulator Left through the Carry	1	12
RR A	Rotate Accumulator Right	1	12
RRC A	Rotate Accumulator Right through the Carry	1	12
SWAP A	Swap nibbles within the Accumulator	1	12
DATA TRANSFER			
MOV A, Rn	Move Register to Accumulator	1	12
MOV A, direct	Move direct byte to Accumulator	2	12
MOV A, @Ri	Move indirect RAM to Accumulator	1	12
MOV A, #data	Move immediate data to Accumulator	2	12
MOV Rn, A	Move Accumulator to register	1	12
MOV Rn, direct	Move direct byte to register	2	24
MOV Rn, #data	Move immediate data to register	2	12
MOV direct, A	Move Accumulator to direct byte	2	12
MOV direct, Rn	Move register to direct byte	2	24
MOV direct, direct	Move direct byte to direct	3	24
MOV direct, @Ri	Move indirect RAM to direct byte	2	24
MOV direct, #data	Move immediate data to direct byte	3	24
MOV @Ri, A	Move Accumulator to indirect RAM	1	12

MNEMONIC	DESCRIPTION	BYTE	OSCIL. PERIOD
DATA TRANSFER (continued)			
MOV @Ri, direct	Move direct byte to indirect RAM	2	24
MOV @Ri, #data	Move immediate data to indirect RAM	2	12
MOV DPTR, #data16	Load Data Pointer with a 16-bit constant	3	24
MOVCA, @A+DPTR	Move Code byte relative to DPTR to Acc	1	24
MOVCA, @A+PC	Move Code byte relative to PC to Acc	1	24
MOVXA, @Ri	Move External RAM (8-bit addr) to Acc	1	24
MOVX A, @DPTR	Move External RAM (16-bit addr) to Acc	1	24
MOVX@Ri, A	Move Acc to External RAM (8-bit addr)	1	24
MOVX @DPTR,A	Move Acc to External RAM (16-bit addr)	1	24
PUSH direct	Push direct byte only stack	2	24
POP direct	Pop direct byte from stack	2	24
XCH A, Rn	Exchange register with Accumulator	1	12
XCH A, direct	Exchange direct byte with Accumulator	2	12
XCH A, @Ri	Exchange indirect RAM with Accumulator	1	12
XCHD A, @Ri	Exchange loworder Digit indirect RAM with Acc	1	12

MNEMONIC	DESCRIPTION	BYTE	OSCIL. PERIOD
BOOLEAN VARIABLE MANIPULATION			
CLR C	Clear Carry	1	12
CLR bit	Clear direct bit	2	12
SETB C	Set Carry	1	12
SETB bit	Set direct bit	2	12
CPL C	Complement Carry	1	12
CPL bit	Complement direct bit	2	12
ANL C, bit	AND direct bit to Carry	2	24
ANL C, /bit	AND complement of direct bit to Carry	2	24
ORL C, bit	OR direct bit to Carry	2	24
ORL C, /bit	OR complement of direct bit to Carry	2	24
MOV C, bit	Move direct bit to Carry	2	12
MOV bit, C	Move Carry to direct bit	2	24
JC rel	Jump if Carry is set	2	24
JNC rel	Jump if Carry not set	2	24
JB bit, rel	Jump if direct Bit is set	3	24
JNB bit, rel	Jump if direct Bit is Not set	3	24
JBC bit, rel	Jump if direct Bit is set & clear bit	3	24
PROGRAM BRANCHING			
ACALL addr11	Absolute Subroutine Call	2	24
LCALL addr16	Long Subroutine Call	3	24
RET	Return from Subroutine	1	24
RETI	Return from interrupt	1	24
AJMPaddr11	Absolute Jump	2	24
LJMPaddr16	Long Jump	3	24
SJMP rel	Short Jump (relative addr)	2	24

MNEMONIC	DESCRIPTION	BYTE	OSCIL. PERIOD
PROGRAM BRANCHING (continued)			
JMP @A+DPTR	Jump indirect relative to the DPTR	1	24
JZ rel	Jump if Accumulator is zero	2	24
JNZ rel	Jump if Accumulator is not Zero	2	24
CNJE A, direct, rel	Compare direct byte to Acc and Jump if Not Equal	3	24
CJNE A, #data, rel	Compare immediate to Acc and Jump if Not Equal	3	24

MNEMONIC	DESCRIPTION	BYTE	OSCIL. PERIOD
PROGRAM BRANCHING (continued)			
CJNE Rn, #data, rel	Compare immediate to register and Jump if Not Equal	3	24
CJNE @Ri, #data, rel	Compare immediate to indirect and Jump if Not Equal	3	24
DJNZ Rn, rel	Decrement register and Jump if Not Zero	2	24
DJNZ direct, rel	Decrement direct byte and Jump if Not Zero	3	24
NOP	No Operation	1	12

5

INSTRUCTION DEFINITIONS

ACALL addr 11

Function : Absolute Call

Description : ACALL unconditionally calls a subroutine located at the indicated address. The instruction increments the PC twice to obtain the address of the following instruction, then pushes the 16-bit result onto the stack (low-order byte first) and increments the Stack Pointer twice. The destination address is obtained by successively concatenating the five high-order bits of the incremented PC, opcode bits 7-5, and the second byte of the instruction. The subroutine called must therefore start within the same 2 K block of the program memory as the first byte of the instruction following ACALL. No flags are affected.

Example : Initially SP equals 07H. The labs " SUBRTN " is at program memory location 0345 H. After executing the instruction,

ACALL SUBRTN

at location 0123H, SP will contain 09H, internal RAM locations 08H and 09H will contain 25H and 01H, respectively, and the PC will contain 0345H.

Bytes : 2

Cycles : 2

Encoding :

a10 a9 a8 1	0 0 0 1

a7 a6 a5 a4	a3 a2 a1 a0

Operation :
ACALL
$(PC) \leftarrow (PC) + 2$
$(SP) \leftarrow (SP) + 1$
$[(SP)] \leftarrow (PC_{7-0})$
$(SP) \leftarrow (SP) + 1$
$[(SP)] \leftarrow (PC_{15-8})$
$(PC_{10-0}) \leftarrow$ page address

5

ADD a, <src-byte>

Function : Add

Description : ADD adds the byte variable indicated to the Accumulator, leaving the result in the Accumulator. The carry and auxiliary-carry flags are set, respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occured.

OV is set there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not bit 6 ; otherwise OV is cleared. When adding signed integers, OV indicates a negative number produced as the sum of two positive operands, or a positive sum from two negative operands.

Four source operand addressing modes are allowed : register, direct, register-indirect, or immediate.

Example : The Accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B). The instruction,

ADD A, R0

will leave 6DH (01101101B) in the Accumulator with the AC flag cleared and both the carry flag and OV set to 1.

ADD A, Rn

Byte : 1

Cycle : 1

Encoding :

0 0 1 0	1 r r r

Operation :

ADD
(A) ← (A) + (Rn)

ADD A, direct

Bytes : 2

Cycle : 1

Encoding :

0 0 1 0	0 1 0 1	direct address

Operation :

ADD
(A) ← (A) + (direct)

ADD A, @RI

Byte : 1

Cycle : 1

Encoding :

0 0 1 0	1 1 1 i

Operation :

ADD (A) ← (A) + (RI)
ADD A, # data

ADD A, # data

Bytes : 2

Cycle : 1

Encoding :

0 0 1 0	0 1 0 0	Immediate data

Operation :

ADD
(A) ← (A) + # data

ADDC A, <src-byte>

Function : Add with Carry

Description : ADDC simultaneously adds the byte variable indicated, the carry flag and the Accumulator contents, leaving the result in the Accumulator. The carry and auxiliary-carry or bit flags are set, respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occured.

OV is set if there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not out of bit 6 ; otherwise OV is cleared. When adding signed intergers, OV indicates a negative number produced as the sum of two positive operands or a positive sum from two negative operands. Four source operand addressing mode are allowed ; register, direct, register-indirect, or immediate.

Example : The Accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B) with the carry flag set. The instruction,

ADDC A, R0

will leave 6EH (01101110B) in the Accumulator with AC cleared and both the Carry flag and OV set to 1.

ADDC A, RN

Byte : 1

Cycle : 1

Encoding :

0	0	1	1	1	r	r	r

Operation : ADDC

$(A) \leftarrow (A) + (C) + (R_n)$

ADDC A, direct

Bytes : 2

Cycle : 1

Encoding :

0	0	1	1	0	1	0	1	direct address

Operation : ADDC

$(A) \leftarrow (A) + (C) + (direct)$

ADDC A, @ RI

Byte : 1

Cycle : 1

Encoding :

0	0	1	1	0	1	1	i

Operation : ADDC

$(A) \leftarrow (A) + (C) + ((R_i))$

ADDC A, #data

Bytes : 2

Cycle : 1

Encoding :

0	0	1	1	0	1	0	0	immediate data

Operation : ADDC

$(A) \leftarrow (A) + (C) + \#\ data$

AJMP addr11

Function : Absolute Jump

Description : AJMP transfers program execution to the indicated address, which is formed at run-time by concatenating the high-order five bits of the PC (*after* incrementing the PC twice), opcode bits 7-5, and the second byte of the instruction. The destination must therefore be within the same 2 K block of program memory as the first byte of the instruction following AJMP.

Example : The label " JMPADR " is at program memory location 0123H. The instruction,

AJMP JMPADR

is a location 0345H and will load the PC with 0123H.

ADD A, direct

Bytes : 2

Cycles : 2

Encoding :

a10	a9	a8	0	0	0	0	1

a7	a6	a5	a4	a3	a2	a1	a0

Operation :
AJMP
(PC) ← (PC) + 2
($PC_{10\text{-}0}$) ← page address

ANL <dest-byte>, <src-byte>

Function : Logical-AND for byte variables

Description : ANL performs the bitwise logical-AND operation between the variables indicated and stores the results in the destination variable. No flags are affected.
The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing ; when the destination is a direct address, the source can be the Accumulator or immediate data. *Note* : When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example : If the Accumulator holds 0C3H (11000011B) and register 0 holds 55H (01010101B) then the instruction,
ANL A, R0
will leave 41H (01000001B) in the Accumulator.
When the destination is a directly addressed byte, this instruction will clear combinations of bits in any RAM location or hardware register. The mask byte determining the pattern of bits to be cleared would either be a constant contained in the instruction or a value computed in the Accumulator at run-time. The instruction,
ANL P1, #01110011B
will clear bits 7, 3, and 2 of output port 1.

ANL A, RN

Bytes : 1

Cycles : 1

Encoding :

0	1	0	1	1	r	r	r

Operation :
ANL
(A) ← (A) ∧ (Rn)

ANL A, direct

Bytes : 2

Cycles : 1

Encoding :

0	1	0	1	0	1	0	1

direct address

Operation :
ANL
(A) ← (A) ∧ (direct)

ANL A, @ RI

Byte : 1

Cycle : 1

Encoding :

0 1 0 1	0 1 1 i

Opération : ANL
(A) ← (A) ∧ ((Ri))

ANL A, #DATA

Bytes : 2

Cycle : 1

Encoding :

0 1 0 1	0 1 0 0	immediate data

Operation : ANL
(A) ← (A) ∧ # data

ANL direct, A

Bytes : 2

Cycle : 1

Encoding :

0 1 0 1	0 0 1 0	direct address

Operation : ANL
(direct) ← (direct) ∧ (A)

ANL direct, # data

Bytes : 3

Cycles : 2

Encoding :

0 1 0 1	0 0 1 1	direct address	immediate data

Operation : ANL
(direct) ← (direct) ∧ # data

ANL C, <src-bit>

Function : Logical-AND for bit variables

Description : If the Boolean value of the source bit is logical 0 then clear the carry flag ; otherwise leave the carry flag in its current state. A slash (" / ") preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, *but the source bit itself is not affected.* No other flags are affected.

Only direct addressing is allowed for the source operand.

Example : Set the carry flag if, P1.0 = 1, ACC.7 = 1, and OV = 0 :

```
MOV C, P1.0          ; LOAD CARRY WITH INPUT PIN STATE
ANL C, ACC.7         ; AND CARRY WITH ACCUM. BIT 7
ANL C,/OV            ; AND WITH INVERSE OF OVERFLOW FLAG
```

ANL C, bit

Bytes : 2

Cycles : 2

Encoding :

1 0 0 0	0 0 1 0	bit address

Operation : ANL
(C) ← (C) ∧ (bit)

ANL C, /bit

Bytes : 2

Cycles : 2

Encoding :

1 0 1 1	0 0 0 0	bit address

Operation : ANL
(C) ← (C) ∧ $\overline{(bit)}$

5

CJNE <dest-byte>, <src-byte>, rel

Function : Compare and Jump if Not Equal

Description : CJNE compares the magnitudes of the first two operands, and branches if their values are not equal. The branch destination is computed by adding the signed relative-displacement in the last instruction byte to the PC, after incrementing the PC to the start of the next instruction. The carry flag is set if the unsigned integer value of <dest-byte> is less than the unsigned integer value of <src-byte> ; otherwise, the carry is cleared. Neither operand is affected.

The first two operands allow four addressing mode combinations : the Accumulator may be compared with any directly addressed byte or immediate data, and any indirect RAM location or working register can be compared with an immediate constant.

Example : The Accumulator contains 34H, register 7 contains 56H. The first instruction in the sequence,

```
                CJNE   R7, #60H, NOT_EQ
;               ...    ...               ; R7 = 60H
NOT_EQ :        JC     REQ_LOW           ; IF R7 < 60H
;               ...    ...               ; R7 > 60H
```

sets the carry flag and branches to the instruction at label NOT-EQ. By testing the carry flag, this instruction determines whether R7 is greater or less than 60H.

If the data being presented to Port 1 is also 34H, then the instruction,

WAIT : CJNE A, P1, WAIT

clears the carry flag and continues with the next instruction in sequence, since the Accumulator does equal the data read from P1. (If some other value was being input on P1, the program will loop at this point until the P1 data changes to 34H).

CJNE A, direct, rel

Bytes : 3

Cycles : 2

Encoding :

1 0 1 1 | 0 1 0 1	direct address	rel. address

Operation :

(PC) ← (PC) + 3
IF (A) <> (direct)
THEN
    (PC) ← (PC) + *relative offset*1
IF (A) < (direct)
THEN
    (C) ← 1
ELSE
    (C) ← 0

CJNE A, # data, rel

Bytes : 3

Cycles : 2

Encoding :

1 0 1 1 | 0 1 0 0	immediate data	rel. address

Operation :

(PC) ← (PC) + 3
IF (A) <> *(data)*
THEN
    (PC) ← (PC) + *relative offset*
IF (A) < data
THEN
    (C) ← 1
ELSE
    (C) ← 0

CJNE Rn, # data, rel

Bytes : 3

Cycles : 2

Encoding :

1 0 1 1 | 1 r r r	immediate data	rel. address

Operation :

(PC) ← (PC) + 3
IF (Rn) <> *data*
THEN
    (PC) ← (PC) + *relative offset*
IF (Rn) < *data*
THEN
    (C) ← 1
ELSE
    (C) ← 0

CJNE @Ri, # data, rel

Bytes : 3

Cycles : 2

Encoding :

1 0 1 1 | 0 1 1 i	immediate data	rel. address

Operation :

(PC) ← (PC) + 3
IF (Ri) <> *data*
THEN
    (PC) ← (PC) + *relative offset*
IF ((Ri)) < *data*
THEN
    (C) ← 1
ELSE
    (C) ← 0

CLR A

Function : Clear Accumulator

Description : The Accumulator is cleared (all bits set on zero). No flags are affected.

Example : The Accumulator contains 5CH (01011100B). The instruction,
CLRA
Will leave the Accumulator set to 00H (00000000B).

Bytes : 1
Cycles : 1

Encoding :

1 1 1 1	0 1 0 0

Operation : CLR
(A) ← 0

CLR bit

Function : Clear bit

Description : The indicated bit is cleared (reset to zero). No other flags are affected. CLR can operate on the carry flag or any directly addressable bit.

Example : Port 1 has previously been written with 5DH (01011101B). The instruction,
CLR P1.2
will leave the port set to 59H (01011001B).

CLR C

Bytes : 1
Cycles : 1

Encoding :

1 1 0 0	0 0 1 1

Operation : CLR
(C) ← 0

CLR bit

Bytes : 2
Cycles : 1

Encoding :

1 1 0 0	0 0 1 0	bit address

Operation : CLR
(bit) ← 0

CPLA

Function : Complement Accumulator

Descritpion : Each bit of the Accumulator is logically complemented (one's complement). Bits which previously contained a one are changed to a zero and vice-versa. No flags are affected.

Example : The accumulator contains 5CH (01011100B). The instruction,
CPLA
will leave the Accumulator set to 0A3H (10100011B).

Bytes : 1
Cycles : 1

Encoding :

1 1 1 1	0 1 0 0

Operation : CPL
$(A) \leftarrow \overline{(A)}$

CPL bit

Function : Complement bit

Description : The bit variable specified is complemented. A bit which had been a one is changed to zero and vice-versa. No other flags are affected. CLR can operate on the carry or any directly addressable bit.
Note : When this instruction is used to modify an output pin, the value used as the original data will be read from the output data latch, not the input pin.

Example : Port 1 has previously been written with 5BH (01011101B). The instruction sequence.

```
CPL P1.1
CPL P1.2
```

will leave the port set to 5BH (01011011B).

CPLC

Bytes : 1

Cycles : 1

Encoding :

1	0	1	1	0	0	1	1

Operation :
CPL
$(C) \leftarrow \overline{(C)}$

CPL bit

Bytes : 2

Cycles : 1

Encoding :

1	0	1	1	0	0	1	0	bit address

Operation :
CPL
$(bit) \leftarrow \overline{(bit)}$

DA A

Function : Decimal-adjust Accumulator for Addition

Description : DA A adjusts the eight-bit value in the Accumulator resulting from the earlier addition of two variables (each in packed-BCD format), producing two four-bit digits, Any ADD or ADDC instruction may have been used to perform the addition.

If Accumulator bits 3-0 are greater than nine (xxxx1010-xxxx1111), or if the AC flag is one, six is added to the Accumulator producing the proper BCD digit in the low-order nibble. This internal addition would set the carry flag if a carry-out of the low-order four-bit field propagated through all high-order bits, but it would not clear the carry flag otherwise.

If the carry flag is now set, or if the four high-order bits now exceed nine (1010xxxx - 111xxxx), these high-order bits are incremented by six, producing the proper BCD digit in the high-order nibble. Again, this would set the carry flag if there was a carry-out of the high-order bits, but wouldn't clear the carry. The carry flag thus indicates if the sum of the original two BCD variables is greater than 100, allowing multiple precision decimal. OV is not affected.

All of this occurs during the one instruction cycle. Essentially, this instruction performs the decimal conversion by adding 00H, 06H, 60H, or 66H to the Accumulator, depending on initial Accumulator and PSW conditions.

Note : DA A *cannot* simply convert a hexadecimal number in the Accumulator to BCD notation, nor does DA A apply to decimal substraction.

Example : The Accumulator holds the value 56H (01010110B) representing the packed BCD digits of the decimal number 56. Register 3 contains the value 67H (01100111B) representing the packed BCD digits of the decimal number 67. The carry flag is set. The instruction sequence.

```
ADDC  A, R3
DA    A
```

will first perform a standard twos-complement binary addition, resulting in the value 0BEH (10111110), in the Accumulator. The carry and auxillary carry flags will be cleared.

The decimal Adjust instruction will then after the Accumulator to the value 24H (00100100B) indicating the packed BCD digits of the decimal number 24, the low-order two digits of the decimal sum of 56,67, and the carry-in. The carry flag will set by the Decimal Adjust instruction, indicating that a decimal overflow occured. The true sum 56,67, and 1 is 124.
BCD variables can be incremented or decremented by adding 01H or 99H. If the Accumulator initially holds 30H (representing the digits of 30 decimal), then the instruction sequence,

```
ADD   A, #99H
DA    A
```

will leave the carry set and 29H in the Accumulator, since 30 + 99 = 129. The low-order byte of the sum can be interpreted to mean 30 - 1 = 29.

Bytes : 1

Cycles : 1

Encoding :

1	1	0	1	0	1	0	0

Operation :

DA
- contents of Accumulator are BCD
IF [[(A_{3-0}) > 9] V [(AC) = 1]]
THEN (A_{3-0}) ← (A_{3-0}) + 6
AND
IF [[(A_{7-4}) > 9] V [(C) = 1]]
THEN (A_{7-4}) ← (A_{7-4}) + 6

DEC byte

Function : Decrement

Description : The variable indicated is decremented by 1. An original value of 00H will underflow to 0FFH. No flags are affected. Four operand addressing modes are allowed : accumulator, register, direct, or register-indirect.

Note : When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example : Register 0 contains 7FH (01111111B). Internal RAM locations 7 EH and 7FH contain 00H and 40H, respectively. The instruction sequence.

DEC @ R0

DEC R0

DEC @ R0

will leave register 0 set to 7EH internal RAM locations 7EH and 7FH to 0FFH and 3FH.

DEC A

Bytes : 1

Cycles : 1

Encoding :

0 0 0 1	0 1 0 0

Operation : DEC
(A) ← (A) - 1

DEC Rn

Bytes : 1

Cycles : 1

Encoding :

0 0 0 1	1 r r r

Operation : DEC
(Rn) ← (Rn) - 1

DEC direct

Bytes : 2

Cycles : 1

Encoding :

0 0 0 1	0 1 0 1	direct address

Operation : DEC
(direct) ← (direct) - 1

DEC @ RI

Bytes : 1

Cycles : 1

Encoding :

0 0 0 1	0 1 1 i

Operation : DEC
((Ri)) ← ((Ri)) - 1

DIV AB

Function : Divide

Description : DIV AB divides the unsigned eight-bit integer in the Accumulator by the unsigned eight-bit integer in register B. The Accumulator receives the integer part of the quotient ; register B receives the integer remainder. The carry and OV flags will be cleared.
Exception : If B had originally contained 00H ; the values returned in the Accumulator and B-register will be undefined and the overflow flag will be set. The carry flag is cleared in any case.

Example : The Accumulator contains 251 (0FBH or 11111011B) and B contains 18 (12H or 00010010B). The instruction,

DIV AB

will leave 13 in the Accumulator (0DH or 00001101B) and the value 17 (11H or 00010001B) in B, since 251 = (13 x 18) + 17. Carry and OV will both be cleared.

Bytes : 1

Cycles : 4

Encoding :

1	0	0	0	0	1	0	0

Operation : DIV

$(A)_{15-8} \leftarrow (A)/(B)$

$(B)_{7-0}$

DJNZ <byte>, <rel-addr>

Function : Decrement and Jump if Not Zero

Description : DJNZ decrements the location indicated by 1, and branches to the address indicated by the second operand if the resulting value is not zero. An original value of 00H will underflow to 0FFH. No flags are affected. The branch destination would be computed by adding the signed relative-displacement value in the last instruction byte to the PC, after incrementing the PC to the first byte of the following instruction.

The location decremented may be a register or directly addressed byte.

Note : When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example : Internal RAM locations 40H, 50H, and 60H contain the values 01H, 70H, and 15H, respectively. the instruction sequence,

DJNZ 40H, LABEL_1

DJNZ 50H, LABEL_2

DJNZ 60H, LABEL_3

will cause a jump to the instruction at label LABEL2 with the values 00H, 6FH, and 15H in the three RAM locations. The first jump was *not* taken because the result was zero.

This instruction provides a simple way of executing a program loop a given number of times, or for adding a moderate time delay (from 2 to 512 machine cycles) with a single instruction. The instruction sequence,

```
        MOV   R2, #8
TOGGLE: CPL   P1.7
        DJNZ  R2, TOGGLE
```

will toggle P1.7 eight times, causing four output pulses to appear at bit 7 of output Port 1. Each pulse will last three machine cycles ; two for DJNZ and one to after the pin.

DJNZ Rn, rel

Bytes : 2

Cycles : 2

Encoding :

1 1 0 1 1 r r r	rel. address

Operation :

```
DJNZ
(PC) ← (PC) + 2
(Rn) ← (Rn) - 1
IF (RN) > 0 or (Rn) < 0
        THEN
                (PC) ← (PC) + rel
```

DJNZ direct, rel

Bytes : 3

Cycles : 2

Encoding :

1 1 0 1 0 1 0 1	direct address	rel. address

Operation :

```
DJNZ
(PC) ← (PC) + 2
(direct) ← (direct) - 1
IF (direct) > 0 or (direct) < 0
        THEN
                (PC) ← (PC) + rel
```

INC <byte>

Function : Increment

Description : INC increments the indicated variable by 1. An original value of 0FFH will overflow to 00H. No flags are affected. There addressing modes are allowed : register, direct, or register-indirect.

Note : When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example : Register 0 contains 7EH (011111110B). Internal locations 7EH and 7FH contain 0FFH and 40H, respectively. The instruction sequence,

```
INC @R0
INC R0
INC @R0
```

will leave register 0 set to 7FH and internal RAM locations 7EH and 7FH holding (respectively) 00H and 41H.

INC A

Bytes : 1

Cycles : 1

Encoding :

0 0 0 0 0 1 0 0

Operation :

```
INC
(A) ← (A) + 1
```

INC Rn

Bytes : 1

Cycles : 1

Encoding :

0 0 0 0 1 r r r

Operation :

```
INC
(Rn) ← (Rn) + 1
```

INC direct

Bytes : 2

Cycles : 1

Encoding :

0 0 0 0	0 1 0 1	direct address

Operation : INC
(direct) ← (direct) + 1

INC @ RI

Bytes : 1

Cycles : 1

Encoding :

0 0 0 0	0 1 1 i

Operation : INC
((Ri)) ← ((Ri)) + 1

INC DPTR

Function : Increment Data Pointer

Description : Increment the 16-bit data pointer by 1.A 16-bit increment (modulo 2^{16}) is performed ; an overflow of the low-order byte of the data pointer (DPL) from 0FFH to 00H will increment the high-order byte (DPH). No flags are affected.

This is the only 16-bit register which can be incremented.

Example : Registers DPH and DPL contain 12H and 0FEH, respectively. The instruction sequence,

INC DPTR

INC DPTR

INC DPTR

will change DPH and DPL to 13H and 01H.

Bytes : 1

Cycles : 2

Encoding :

1 0 1 0	0 0 1 1

Operation : INC
(DPTR) ← (DPTR) + 1

JB bit,rel

Function : Jump if Bit set

Descritpion : If the indicated bit is a one, jump to the address indicated ; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. *The bit tested is not modified.* No flags are affected.

Note : When this instruction is used to test an output pin, the value used as the original data will be read from the output data latch, *not* the input pin.

Example : The data present at input port 1 is 11001010B. The Accumulator holds 56 (01010110B). The instruction sequence.

JB P1.2, LABEL 1
JB ACC.2, LABEL 2

will cause program execution to branch to the instruction at label LABEL 2.

Bytes : 3

Cycles : 2

Encoding :

0 0 1 0 0 0 0 0	bit address	rel. address

Operation :

```
JB
(PC) ← (PC) + 3
IF (bit) = 1
        THEN
                (PC) ← (PC) + rel
```

JBC bit, rel

Function : Jump if Bit is set and Clear bit

Description : If the indicated bit is a one, branch to the address indicated ; otherwise proceed with the next instruction. *The bit will not be cleared if it is already a zero.* The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. No flags are affected.

Note : When this instruction is used to test an output pin, the value used as the original data will be read from the output data latch, *not* the input pin.

Example : The Accumulator holds 56H (01010110B). The instruction sequence,

```
JBC ACC.3, LABEL 1
JBC ACC.2, LABEL 2
```

will cause program execution to continue at the instruction identified by the label LABEL2, with the Accumulator modified to 52H (01010010B).

Bytes : 3

Cycles : 2

Encoding :

0 0 0 1 0 0 0 0	bit address	rel. address

Operation :

```
JBC
(PC) ← (PC) + 3
IF (bit) = 1
        THEN
                (bit) ← 0
                (PC) ← (PC) + rel
```

JC rel

Function : Jump if Carry is set

Description : If the carry flag is set, branch to the address indicated ; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. No flags are affected.

Example : The carry flag is cleared. The instruction sequence,

```
JC      LABEL 1
CPL     C
JC      LABEL 2
```

will set the carry and cause program execution to continue at the instruction identified by the label LABEL2.

Bytes : 2

Cycles : 2

Encoding :

0 1 0 0 0 0 0 0	rel. address

Operation :

```
JC
(PC) ← (PC) + 2
IF (C) = 1
        THEN
                (PC) ← (PC) + rel
```

JMP @A + DPTR

Function : Jump indirect

Description : Add the eight-bit unsigned contents of the Accumulator with the sixteen-bit data pointer, and load the resulting sum to the program counter. This will be the address for subsequent instruction fetches. Sixteen-bit addition is performed (modulo 2^{16}) : a carry-out from the low-order eight bits propagates through the higher-order bits. Neither the Accumulator nor the Data Pointer is altered. No flags are affected.

Example : An even number from 0 to 6 is in the Accumulator. The following sequence of instructions will branch to one of four AJMP instructions in a jump table starting at JMP-TBL :

```
                MOV   DPTR, #JMP_TBL
                JMP   @ A + DPTR
JMP_TBL :       AJMP  LABEL0
                AJMP  LABEL1
                AJMP  LABEL2
                AJMP  LABEL3
```

If the Accumulator equals 04H when starting this sequence, execution will jump to label LABEL2. Remembers that AJMP is a two-byte instruction, so the jump instructions start at every other address.

Bytes : 1

Cycles : 2

Encoding :

0 1 1 1	0 0 1 1

Operation :
JMP
(PC) ← (A) + (DPTR)

JNB bit, rel

Function : Jump if Bit not set

Description : If the indicated bit is a zero, branch to the indicated address ; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. *The bit tested is not modified.* No flags are affected.

Example : The data present at input port 1 is 11001010B. The Accumulator holds 56H (01010110B). The instruction sequence,

```
JNB P1.3, LABEL1
JNB ACC3, LABEL2
```

will cause program execution to continue at the instruction at label LABEL2.

Bytes : 3

Cycles : 2

Encoding :

0 0 1 1	0 0 0 0	bit address	rel. address

Operation :
JNB
(PC) ← (PC) + 3
IF (bit) = 0
THEN (PC) ← (PC) + rel

JNC rel

Function : Jump if Carry not set

Description : If the carry flag is a zero, branch to the address indicated ; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice to point to the next instruction. The carry flag is not modified.

Example : The carry flag is set. The instruction sequence,

```
JNC LABEL1
CPLC
JNC LABEL2
```

will clear the carry and cause program execution to continue at the instruction identified by the label LABEL2.

Bytes : 2

Cycles : 2

Encoding :

0 1 0 1 | 0 0 0 0	rel. address

Operation :

```
JNC
(PC) ← (PC) + 2
IF (C) = 0
        THEN (PC) ← (PC) + rel
```

JNZ rel

Function Jump if Accumulator Not Zero

Description : If any bit of the Accumulator is a one, branch to the indicated address ; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The Accumulator is not modified. No flags are affected.

Example : The Accumulator originally holds 00H. The instruction sequence,

```
JNZ LABEL1
INC A
JNZ LABEL2
```

will set the Accumulator to 01H and continue at label LABEL2.

Bytes : 2

Cycles : 2

Encoding :

0 1 1 1 | 0 0 0 0	rel. address

Operation :

```
JNZ
(PC) ← (PC) + 2
IF (A) ≠ 0
        THEN (PC) ← (PC) + rel
```

5

JZ rel

Function : Jump if Accumulator Zero

Description : If all bits of the Accumulator are zero, branch to the address indicated ; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The Accumulator is not modified. No flags are affected.

Example : The Accumulator originally contains 01H. The instruction sequence.

```
JZ      LABEL1
DEC     A
JZ      LABEL2
```

will change the Accumulator to 00H and cause program execution at the instruction identified by the label LABEL2.

Bytes : 2

Cycles : 2

Encoding :

0 1 1 0	0 0 0 0	rel. address

Operation :

JZ
(PC) ← (PC) + 2
IF (A) = 0
THEN (PC) ← (PC) + rel

LCALL addr16

Function : Long call

Description : LCALL calls a subroutine located at the indicated address. The instruction adds three to the program counter to generate the address of the next instruction and then pushes the 16-bit result onto the stack (low byte first), incrementing the Stack Pointer by two. The high-order and low-order bytes of the PC are then loaded, respectively, with the second and third bytes of the LCALL instruction. Program execution continues with the instruction at this address. The subroutine may therefore begin anywhere in the full 64K-byte program memory address space. No flags are affected.

5

Example : Initially the Stack Pointer equals 07H. The label " SUBRTN " is assigned to program memory location 1234H. After executing the instruction,

LCALL SUBRTN

at location 0123H, the Stack Pointer will contain 09H, internal RAM locations 08H and 09H will contain 26H and 01H, and the PC will contain 1235H.

Bytes : 3

Cycles : 2

Encoding :

0 0 0 1	0 0 1 0	addr15-addr8	addr7-addr0

Operation :

LCALL
(PC) ← (PC) + 3
(SP) ← (SP) + 1
((SP)) ← (PC_{7-0})
(SP) ← (SP) + 1
((SP)) ← (SP_{15-8})
(PC) ← $addr_{15-0}$

LJMP addr16

Function : Long Jump

Description : LJMP causes an unconditional branch to the indicated address, by loading the high-order and low-order bytes of the PC (respectively) with the second and third instruction bytes. The destination may therefore be anywhere in the full 64K program memory address space. No flags are affected.

Example : The label " JMPADR " is assigned to the instruction at program memory location 1234H. The instruction,

LJMP JMPADR

at location 0123H will load the program counter with 1234H.

Bytes : 3

Cycles : 2

Encoding :

0 0 0 0 0 0 1 0	addr15-addr8	addr7-addr0

Operation : LJMP
(PC) ← $addr_{15-0}$

MOV <dest-byte>, <src-byte>

Function : Move byte variable

Description : The byte variable indicated the second operand is copied into the location specified by the first operand. The source byte is not affected. No other register or flag is affected.

This is by far the most flexible operation. Fifteen combinaisons of source and destination addressing modes are allowed.

Example : Internal RAM location 30H holds 40H. The value of RAM location 40H is 10H.
The data present at input port 1 is 11001010B (0CAH).

```
MOV   R0, #30H      ; R0 <= 30h
MOV   A, @ R0       ; A <= 40H
MOV   R1, A         ; R1 <= 40h
MOV   R, @ R1       ; B <= 10h
MOV   @ R1, P1      ; RAM (40H) <= OCAH
MOV   P2, P1        ; P2 # 0CAH
```

leaves the value 30H in register 0,40H in both the Accumulator and register 1,10H in register B, and 0CAH (11001010B) both in RAM location 40H and output on port 2.

MOV A, Rn

Bytes : 1

Cycles : 1

Encoding :

1 1 1 0 1 r r r

Operation : MOV
(A) ← (Rn)

***MOV A,direct**

Bytes : 2

Cycles : 1

Encoding :

1 1 1 0 0 1 0 1	direct address

Operation : MOV
(A) ← (direct)

*MOV A, ACC is not valid instruction.

MOV A,@ RI

Bytes : 1

Cycles : 1

Encoding :

1	1	1	0	0	1	1	i

Operation : MOV
(A) ← (Ri)

MOV A, # data

Bytes : 2

Cycles : 1

Encoding :

0	1	1	1	0	1	0	0	immediate data

Operation : MOV
(A) ← # data

MOV Rn, A

Bytes : 1

Cycles : 1

Encoding :

1	1	1	1	1	r	r	r

Operation : MOV
(Rn) ← (A)

MOV Rn, direct

Bytes : 2

Cycles : 2

Encoding :

1	0	1	0	1	r	r	r	direct addr.

Operation : MOV
(Rn) ← (direct)

MOV Rn, # data

Bytes : 2

Cycles : 1

Encoding :

0	1	1	1	1	r	r	r	immediate data

Operation : MOV
(Rn) ← # data

MOV direct, A

Bytes : 2

Cycles : 1

Encoding :

1	1	1	1	0	1	0	1	direct address

Operation : MOV
(direct) ← (A)

MOV direct, Rn

Bytes : 2

Cycles : 2

Encoding :

1	0	0	0	1	r	r	r	direct address

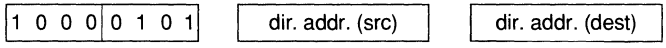
Operation : MOV
(direct) ← (Rn)

MOV direct, direct

Bytes : 3

Cycles : 2

Encoding :



Operation : MOV
(direct) ← (direct)

MOV direct, @ Ri

Bytes : 2

Cycles : 2

Encoding :

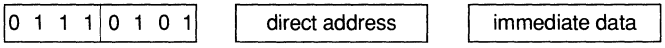
Operation : MOV
(direct) ← (Ri)

MOV direct, # data

Bytes : 3

Cycles : 2

Encoding :



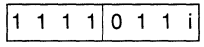
Operation : MOV
(direct) ← # data

MOV @ Ri, A

Bytes : 1

Cycles : 1

Encoding :



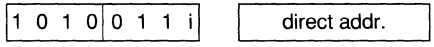
Operation : MOV
((Ri)) ← (A)

MOV @ Ri, direct

Bytes : 2

Cycles : 2

Encoding :



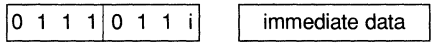
Operation : MOV
((Ri)) ← (direct)

MOV @ Ri, data

Bytes : 2

Cycles : 1

Encoding :



Operation : MOV
((Ri)) ← # data

5

MOV <dest-bit>, <src-bit>

Function : More bit data

Description : The Boolean variable indicated by the second operand is copied into the location specified by the first operand. One of the operands must be the carry flag ; the other may be any directly addressable bit. No other register or flag is affected.

Example : The carry flag is originally set. The data present at input Port 3 is 11000101B. The data previously written to output Port 1 is 35H (00110101B).

```
MOV     P1.3, C
MOV     C, P3.3
MOV     P1.2, C
```

will leave the carry cleared and change Port 1 to 39H (00111001B).

MOV C, bit

Bytes : 2

Cycles : 1

Encoding :

1 0 1 0 0 0 1 0	bit address

Operation :
MOV
(C) ← (bit)

MOV bit, C

Bytes : 2

Cycles : 2

Encoding :

1 0 0 1 0 0 1 0	bit address

Operation :
MOV
(bit) ← (C)

5

MOV DPTR, # data 16

Function : Load Data Pointer with a 16-bit constant

Description : The Data Pointer is loaded with the 16-bit constant indicated. the 16-bit constant is loaded into the second and third bytes of the instruction. The second byte (DPH) is the high-order byte, while the third byte (DPL) holds the low-order byte. No flags are affected.

This is the only instruction which moves 16-bits of data at once.

Example : The instruction,

MOV DPTR, 1234H

will load the value 1234H into the Data Pointer : DPH will hold 12H and DPL will hold 34H.

Bytes : 3

Cycles : 2

Encoding :

1 0 0 1 0 0 0 0	immed. data 15-8	immed. data 7-0

Operation :
MOV
(DPTR) ← # $data_{15-0}$
DPH DPL ← # $data_{15-8}$ # $data_{7-0}$

MOVC A, @ A + <base-reg>

Function : Move Code byte

Description : The MOVC instructions load the Accumulator with a code byte, or constant from program memory. The address of the byte fetched is the sum of the original unsigned eight-bit. Accumulator contents and the contents of a sixteen-bit base register, which may be either the Data Pointer or the PC. In the latter case, PC is incremented to the address of the following instruction before being added with the Accumulator ; otherwise the base register is not altered. Sixteen-bit addition is performed so a carry-out from the low-order eight bits may propagate through higher-order bits. No flags are affected.

Example : A value between 0 and 3 is in the Accumulator. The following instructions will translate the value in the Accumulator to one of four values defined by the DB (define byte) directive.

```
REL PC :    INC     A
            MOVC    A, @ A + PC
            RET
            DB      66H
            CB      77H
            CB      88H
            DB      99H
```

If the subroutine is called with the Accumulator equal to 01H, it will return with 77H in the Accumulator. The INC A before the MOVC instruction is needed to " get around " the RET instruction above the table. If several bytes of code separated the MOVC from the table, the corresponding number would be added to the Accumulator instead.

MOVC A, @ A + DPTR

Bytes : 1

Cycles : 2

Encoding :

1	0	0	1	0	0	1	1

Operation : MOVC
(A) ← ((A) + (DPTR))

MOVC A, @ A + PC

Bytes : 1

Cycles : 2

Encoding :

1	0	0	0	0	0	1	1

Operation : MOVC
(PC) ← (PC) + 1
(A) ← ((A) + (PC))

MOVX <dest-byte>, <src-byte>

Function : Move External

Description : The MOVX instructions transfer data between the Accumulator and a byte of external data memory, hence the " X " appended to MOV. There are two types of instructions, differing in whether they provide an eight-bit or sixteen-bit indirect address to the external data RAM.
In the first type, the contents of R0 or R1 in the current register bank provide an eight-bit address multiplexed with data on P0. Eight bits are sufficient for external I/O expansion decoding or for a relatively small RAM array. For somewhat larger arrays, any output port pins can be used to output higher-order address bits. These pins would be controlled by an output instruction preceding the MOVX.
In the second type of MOVX instruction, the Data Pointer generates a sixteen-bit address. P2 outputs the high-order eight address bits (the contents of DPH) while P0 multiplexes the low-order eight bits (DPL) with data. The P2 Special Function Register retains its previous contents while the P2 output buffers are emitting the contents of DPH. This form is faster and more efficient when accessing very large data arrays (up to 64K bytes), since no additional instructions are needed to set up the output ports.
It is possible in some situation to mix the two MOVX types. A large RAM array with its high-order address lines driven by P2 can be addressed via the Data Pointer, or with code to output high-order address bits to P2 followed by a MOVX instruction using R0 or R1.

Example : An external 256 byte RAM using multiplexed address/data lines is connected to the 80C51 Port 0. Port 3 provides control lines for the external RAM. Ports 0 and 2 are used for normal I/O. Registers 0 and 1 contain 12H and 34H. Location 34H of the external RAM holds the value 56H. The instruction sequence

```
MOVX    A, @ R1
MOVX    @ R0, A
```

copies the value 56H into both the Accumulator and external RAM location 12H.

MOVX A, @ Ri

Bytes : 1

Cycles : 2

Encoding :

1	1	1	0	0	0	1	i

Operation :
MOVX
(A) ← ((Ri))

MOVX @ Ri, A

Bytes : 1

Cycles : 2

Encoding :

1	1	1	1	0	0	1	i

Operation :
MOVX
((Ri)) ← (A)

MOVX A, @ DPTR

Bytes : 1

Cycles : 2

Encoding :

1	1	1	0	0	0	0	0

Operation :
MOVX
(A) ← ((DPTR))

MOVX @ DPTR, A

Bytes : 1

Cycles : 2

Encoding :

1	1	1	1	0	0	0	0

Operation :
MOVX
(DPTR) ← (A)

NOP

Function : No Operation

Description : Execution continue at the following instruction. Other than the PC, no registers or flags are effected.

Example : It is desired to produce a low-going output pulse on bit 7 of Port 2 lasting exactly 5 cycles. A simple SETB/CLR sequence would generate a one-cycle pulse, so four additional cycles must be inserted. This may be done (assuming no interrupts are enable) with the instruction sequence.

```
CLR     P2.7
NOP
NOP
NOP
NOP
SETP    P2.7
```

Bytes : 1

Cycles : 1

Encoding :

0 0 0 0	0 0 0 0

Operation : NOP

$(PC) \leftarrow (PC) + 1$

MUL AB

Function : Multiply

Description : MUL AB multiplies the unsigned eight-bit integers in the Accumulator and register B. The low-order byte of the sixteen-bit product is left in the Accumulator, and the high-order byte in B. If the product is greater than 255 (OFFH) the overflow flag is set ; otherwise it is cleared. The carry flag is always cleared.

Example : Originally the Accumulator holds the value 80 (50H). Register B holds the value 160 (OAOH). The instruction,

MUL AB

will give the product 12,800 (3200H), so B is changed to 32H (00110010B) and the Accumulator is cleared. The overflow flag is set, carry is cleared.

Bytes : 1

Cycles : 4

Encoding :

1 0 1 0	0 1 0 0

Operation : MUL

$(A)_{7-0} \leftarrow (A) \times (B)$

$(B)_{15-8}$

ORL <dest-byte> <src-byte>

Function : Logical-OR for byte variables

Description : ORL performs the bitwise logical-OR operation between the indicated variables, storing the results in the destination byte, No flags are affected.
The two operands allow six addressing mode combinaisons. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing ; when the destination is a direct address, the source can be the Accumulator or immediate data.
Note : When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.

Example : If the Accumulator holds 0C3H (11000011B) and R0 holds 55H (01010101B) then the instruction,
ORL A, R0
will leave the Accumulator holding the value 0D7H (11010111B).
When the destination is a directly addressed byte, the instruction can set combinations of bits in any RAM location or hardware register. The pattern of bits to be set is determined by a mask byte, which may be either a constant data value in the instruction or a variable computed in the Accumulator at run-time. The instruction.,
ORL P1, # 00110010b
will set bits 5, 4, and 1 of output Port 1.

ORL A, Rn

Bytes : 1

Cycles : 1

Encoding :

0 1 0 0	1 r r r

Operation ORL
(A) ← (A) V (Rn)

ORL A, direct

Bytes : 2

Cycles : 1

Encoding :

0 1 0 0	0 1 0 1	direct address

Operation : ORL
(A) ← (A) V (direct)

ORL A, @ Ri

Bytes : 1

Cycles : 1

Encoding :

0 1 0 0	0 1 1 i

Operation : ORL
(A) ← (A) V ((Ri))

ORL A, # data

Bytes : 2

Cycles : 1

Encoding :

0 1 0 0	0 1 0 0	immediate data

Operation ORL
(A) ← (A) V # data

ORL direct, A

Bytes : 2

Cycles : 1

Encoding :

0 1 0 0	0 0 1 0	direct address

Operation : ORL
(direct) ← (direct) V (A)

ORL direct, # data

Bytes : 3

Cycles : 2

Encoding :

0 1 0 0	0 0 1 1	direct address	immediate data

Operation : ORL
(direct) ← (direct) V # data

ORL C, <src-bit>

Function : Logical-OR for bit variable

Description : Set the carry flag if the Boolean value is a logical 1 ; leave the carry in its current state otherwise. A slash (" / ") preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, but the source bit it self is not affected. No other flags are affected.

Example : Set the carry flag if and only if P1.0 = 1, ACC. 7 = 1, or OV = 0 :

```
MOV   C, P1.0        ; LOAD CARRY WITH INPUT PIN P10
ORL   C, ACC.7       ; OR CARRY WITH THE ACC. BIT7
ORL   C,/OV          ; OR CARRY WITH THE INVERSE OF OV
```

ORL C, bit

Bytes : 2

Cycles : 2

Encoding :

0 1 1 1	0 0 1 0	bit address

Operation : ORL
(C) ← (C) V (bit)

ORL C, /bit

Bytes : 2

Cycles : 2

Encoding :

1 0 1 0	0 0 0 0	bit address

Operation : ORL
(C) ← (C) V $\overline{(bit)}$

POP direct

Function : Pop from stack.

Description : The contents of internal RAM location addressed by the Stack Pointer is read, and the Stack Pointer is decremented by one. The value read is then transferred to the directly addressed byte indicated. No flags are affected.

Example : The Stack Pointer originally contains the value 32H, and internal RAM locations 30H through 32H contain the values 20H, 23H, and 01H, respectively. The instruction sequence,

```
POP DPH
POP DPL
```

will leave the Stack Pointer equal to the value 30H and the Data Pointer set to 0123H. At this point the instruction,

```
POP SP
```

will leave the Stack Pointer set to 20H. Note that in this special case the Stack Pointer was decremented to 2FH before being loaded with the value popped (20H)

Bytes : 2

Cycles : 2

Encoding :

1 1 0 1	0 0 0 0	direct address

Operation : POP
(direct) ← ((SP))
(SP) ← (SP) – 1

PUSH direct

Function : push onto stack.

Description : The Stack Pointer is incremented by one. The contents fo the indicated variable is then copied into the internal RAM location addressed by the Stack Pointer. Otherwise no flags are affected.

Example : On entering interrupt routine the Stack Pointer contains 09H. The Data Pointer holds the value 0123H. The instruction sequence,

```
PUSH DPL
PUSH DPH
```

will leave the Stack Pointer set to 0BH and store 23H and 01H in internal RAM location 0AH and 0BH, respectively.

Bytes : 2

Cycles : 2

Encoding :

1 1 0 0	0 0 0 0	direct address

Operation :

```
PUSH
(SP) ← (SP) + 1
((SP)) ← (direct)
```

RET

Function : Return from subroutine

Description : RET pops the high-and low-order bytes of the PC successively from the stack, decrementing the Stack Pointer by two. Program execution continues at the resulting address, generally the instruction immediately following en ACALL or LCALL. No flags are affected.

Example : The Stack Pointer originally contains the value 0BH. Internal RAM locations 0AH and 0BH contain the values 23H, and 01H, respectively. The instruction,

```
RET
```

will leave the Stack Pointer equal to the value 09H. Program execution will continue at location 0123H.

Bytes : 1

Cycles : 2

Encoding :

0 0 1 0	0 0 1 0

Operation :

```
RT
(PC15 - 8) ← ((SP))
(SP) ←.(SP) – 1
(PC7-0) ← ((SP))
(SP) ← (SP) – 1
```

RETI

Function : Return from interrupt

Description : RETI pops the high-and low-order bytes of the PC successively from the stack, and restores the interrupt logic to accept additional interrupts at the same priority level as the one just processed. The Stack Pointer is left decremented by two. No other registers are affected ;the PSW is *not* automatically restored to its pre-interrupt status. Program execution continues at the resulting address, which is generally the instruction immediately after the point at which the interrupt request was detected. If a lower-or-same-level interrupt had been pending when the RETI instruction is executed, that one instruction will be executed before the pending interrupt is processed.

Example : The Stack Pointer originally contains the value 0BH. An interrupt was detected during the instruction ending at location 0122H. Internal RAM locations 0AH and 0BH contain the values 23H and 01H, respectively. The instruction,

```
RETI
```

will leave the Stack Pointer equal to 09H and return program execution to location 0123H.

Bytes : 1

Cycles : 2

Encoding :

0	0	1	1	0	0	1	0

Operation :
RETI
$(PC_{15-8}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$
$(PC_{7-0}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$

RL A

Function : Rotate Accumulator Left

Description : The eight bits in the Accumulator are rotated one bit to the left. Bit 7 rotated into the bit 0 position. No flags are affected.

Example : The Accumulator holds the value 0C5H (11000101B). The instruction,

RL A

leaves the Accumulator holding the value 8BH (100001011B) with the carry unaffected.

Bytes : 1

Cycles : 1

Encoding :

0	0	1	0	0	0	1	1

Operation :
RL
$(An + 1) \leftarrow (An)$ n = 0 - 6
$(A0) \leftarrow (A7)$

RLC A

Function : Rotate Accumulator Left through the Carry flag

Description : The eight bits in the Accumulator and the carry flag are together rotated one bit to the left. Bit 7 moves into the carry flag ; the original state of the carry flag moves into the bit 0 position. No other flags are affected.

Example : The Accumulator holds the value 0C5H (11000101B), and the carry is zero. The instruction,

RCL A

leaves the Accumulator holding the value 8BH (10001010B) with the carry set.

Bytes : 1

Cycles : 1

Encoding :

0	0	1	1	0	0	1	1

Operation :
RLC
$(An + 1) \leftarrow (An)$ n = 0 - 6
$(A0) \leftarrow (C)$
$(C) \leftarrow (A7)$

RR A

Function : Rotate Accumulator Right

Description : The eight bits in the Accumulator are rotated one bit to the right. Bit 0 is rotated into the bit 7 position. No flags are affected.

Example : The Accumulator holds the value 0C5H (11000101B). The instruction,

RR A

leaves the Accumulator holding the value 0E2H (11100010B) with the carry unaffected.

Bytes : 1

Cycles : 1

Encoding :

0 0 0 0	0 0 1 1

Operation :

RR
(An) ← (An + 1) n = 0 - 6
(A7) ← (A0)

RRC A

Function : Rotate Accumulator Right through Carry flag

Description : The eight bits in the Accumulator and the carry flag are together rotated one bit to the right. Bit 0 moves into the carry flag ; the original value of the carry flag moves into the bit 7 position. No other flags are affected.

Example : The Accumulator holds the value 0C5H (11000101B), and the carry is zero. The instruction,

RRC A

leaves the Accumulator holding the value 62 (01100010B) with the carry set.

Bytes : 1

Cycles : 1

Encoding :

0 0 0 1	0 0 1 1

Operation :

RRC
(An) ← (A_n + 1)n = 0 - 6
(A7) ← (C)
(C) ← (A0)

SETB <bit>

Function : Set bit

Description : SETB sets the indicated bit to one. SETB can operate on the carry flag or any direct addressable bit. No other flags are affected.

Example : The carry flag is cleared. Output Port 1 has been written with the value 34H (00110100B). The instructions,

```
SETB    C
SETB    P1.0
```

will leave the carry flag set to 1 and change the data output on Port 1 to 35H (00110101B).

SETB C

Bytes : 1

Cycles : 1

Encoding :

1 1 0 1	0 0 1 1

Operation :

```
SETB
(C) ← 1
```

SETB bit

Bytes : 2

Cycles : 1

Encoding :

1 1 0 1	0 0 1 0	bit address

Operation :

```
SETB
(bit) ← 1
```

SJMP rel

Function : Short Jump

Description : Program control branches unconditionally to the address indicated. The branch destination is computed by adding the signed displacement in the second instruction byte to the PC, after incrementing the PC twice. Therefore, the range of destinations allowed is from 128 bytes preceding this instruction to 127 bytes following it.

Example : The label "RELADR" is assigned to an instruction at program memory location 0123H. The instruction,

```
SJMP RELADR
```

will assemble into location 0100H. After the instruction is executed, the PC will contain the value 0123H.

(*Note* : Under the above conditions the instruction following SJMP will be at 102H. therefore, the displacement byte of the instruction will be the relative offset (0123H - 0102H) = 21H. Put another way, an SJMP with a displacement of 0FEH would be an one-instruction infinite loop).

Bytes : 2

Cycles : 2

Encoding :

1 0 0 0	0 0 0 0	rel. address

Operation :

```
SJMP
(PC) ← (PC) + 2
(PC) ← (PC) + rel
```

5

SUBB A, <src-byte>

Function : Subtract with borrow

Description : SUBB subtracts the indicated variable and the carry flag together from the Accumulator, leaving the result in the Accumulator. SUBB sets the carry (borrow) flag if a borrow is needed for bit7, and clears C otherwise. (If C was set before executing a SUBB instruction, this indicates that a borrow was needed for the previous step in a multiple precision substraction so the carry is subtracted from the Accumulator along with the source operand). AC is set if a borrow is needed for bit 3, and cleared otherwise. OV is set if a borrow is needed into bit 6, but not into bit 7, or into bit 7, but not bit 6.
When subtracting signed integers OV indicates a negative number produced when a negative value is subtracted from a positive value, or a positive result when a positive number is subtracted from a negative number.
The source operand allows four addressing modes : register, direct, register-indirect, or immediate.

Example : The Accumulator holds 0C9H (11001001B), register 2 holds 54H (01010100B), and the carry flag is set. the instruction,

SUBB A, R2

will leave the value 74H (01110100B) in the accumulator, with the carry flag and AC cleared but OV set.

Notice that 0C9H minus 54H is 75H. The difference between this and the above result is due to the carry (borrow) flag being set before the operation. If the state of the carry is not known before starting a single or multiple-precision substraction, it should not be explicity cleared by a CLRC instruction.

SUBB A, Rn

Bytes : 1

Cycles : 1

Encoding :

1	0	0	1	1	r	r	r

Operation : SUBB
(A) ← (A) – (C) – (Rn)

SUBB A, direct

Bytes : 2

Cycles : 1

Encoding :

1	0	0	1	0	1	0	1	direct address

Operation : SUBB
(A) ← (A) – (C) – (direct)

SUBB A, @ Ri

Bytes : 1

Cycles : 1

Encoding :

1	0	0	1	0	1	1	i

Operation : SUBB
(A) ← (A) – (C) – (Ri)

SUBB A, # data

Bytes : 2

Cycles : 1

Encoding :

1	0	0	1	0	1	0	0	immediate data

Operation : SUBB
(A) ← (A) – (C) – # data

SWAP A

Function : Swap nibbles within the Accumulator

Description : SWAP A interchanges the low-and high-order nibbles (four-bit fields) of the Accumulator (bits 3 - 0 and bits 7 - 4). The operation can also be thought of a four-bit rotate instruction. No flag are affected.

Example : The Accumulator holds the value 0C5H (11000101B). The instruction,

SWAP A

leave the Accumulator holding the value 5CH (01011100B).

Bytes : 1

Cycles : 1

Encoding :

1	1	0	0	0	1	0	0

Operation :

SWAP
$(A_{3-0}) \rightleftarrows (A_{7-4})$

XCH A, <byte>

Function : Exchange Accumulator with byte variable

Description : XCH loads the Accumulator with the contents of the indicated variable, at the same time writing the original Accumulator contents to the indicated variable. The source/destination operand can use register, direct, or register-indirect addressing.

Example : R0 contains the addres 20H. The Accumulator holds the value 3FH (00111111B). Internal RAM location 20H holds the value 75H (01110101B). The instruction,

XCH A, @R0

will leave RAM location 20H holding the values 3FH (00111111B) and 75H (01110101B) in the Accumulator.

XCH A, Rn

Bytes : 1

Cycles : 1

Encoding :

1	1	0	0	1	r	r	r

Operation :

XCH
$(A) \rightleftarrows (Rn)$

XCH A, direct

Bytes : 2

Cycles : 1

Encoding :

1	1	0	0	0	1	0	1

direct address

Operation :

XCH
$(A) \rightleftarrows$ (direct)

XCH A, @Ri

Bytes : 1

Cycles : 1

Encoding :

1	1	0	0	0	1	1	i

Operation :

XCH
$(A) \rightleftarrows ((Ri))$

XCHD A, @ Ri

Function : Exchange Digit

Description : XCHD exchanges the low-order nibble of the Accumulator (bits 3 - 0), generally representing a hexadecimal or BCD digit, with that of the internal RAM location indirectly addressed by the specified register. The high-order nibbles (7 - 4) of each register are not affected. No flags are affected.

Example : R0 contains the address 20H. The Accumulator holds the value 36H (00110110B). Internal RAM location 20H holds the value 75H (01110101B). The instruction,

XCHD A, @ R0

will leave RAM location 20H holding the value 76H (01110110B) and 35H (00110101B) in the Accumulator.

Bytes : 1

Cycles : 1

Encoding :

1	1	0	1	0	1	1	i

Operation : XCHD

$(A_{3-0}) \rightleftarrows ((Ri_{3-0}))$

XRL <dest-byte>, <src-byte>

Function : Logical Exclusive-OR for byte variable

Description : XRL performs the bitwise logical Exclusive- OR operation between the indicated variables, storing the results in the destination. No flags are affected.
The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing ; when the destination is a direct address, the source can be the accumulator or immediate data.

(*Note* : When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins).

Example : If the Accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B) then the instruction,

XRL A, R0

will leave the Accumulator holding the value 69H (01101001B).
When the destination is a directly addressed byte, this instruction can complement combinations of bits in any RAM location or hardware register. The pattern of bits to be complemented is then determined by a mask byte, either a constant contained in the instruction or a variable computed in the Accumulator at run-time. The instruction,

XRL P1, # 00110001B

will complement bits 5, 4, and 0 of output Port 1.

XRL A, Rn

Bytes : 1

Cycles : 1

Encoding :

0	1	1	0	1	r	r	r

Operation : XRL

$(A) \leftarrow (A) \forall (Rn)$

XRL A, direct

Bytes : 2

Cycles : 1

Encoding :

0	1	1	0	0	1	0	1

direct address

Operation : XRL

$(A) \leftarrow (A) \forall (direct)$

XRL A, @ Ri

Bytes : 1

Cycles : 1

Encoding :

0	1	1	0	0	1	1	i

Operation : XRL
(A) ← (A) ∀ ((Ri))

XRL A, #data

Bytes : 2

Cycles : 1

Encoding :

0	1	1	0	0	1	0	0	immediate data

Operation : XRL
(A) ← (A) ∀ # data

XRL direct, A

Bytes : 2

Cycles : 1

Encoding :

0	1	1	0	0	0	1	0	direct address

Operation : XRL
(direct) ← (direct) ∀ (A)

XRL direct, # data

Bytes : 3

Cycles : 2

Encoding :

0	1	1	0	0	0	1	1	direct address	immediate data

Operation : XRL
(direct) ← (direct) ∀ # data

DATA SHEETS

6

DATA SHEET

80C51/80C31

CMOS SINGLE-CHIP 8 BIT MICROCONTROLLER

- 80C51-CMOS SINGLE-CHIP MICROCONTROLLER with factory mask-programmable ROM.
- 80C31-ROM LESS VERSION OF THE 80C51
- 80C51/C31 : 0 TO 12 MHz
  80C51/C31-1 : 0 TO 16 MHz
  80C51/C31s : 0 TO 20 MHz
- OTHER DEVICES WITH A SPECIFIC DATA SHEET :
  80C51/C31-L : V_{CC} = 2.7 V TO 6 V (0 TO 6 MHz)
  80C51F : THE INTERNAL ROM CODE CANNOT BE READ OR DUMPED AFTER ACTIVATION OF A SPECIAL PROTECTION
  80C51/C31s : 0 TO 20 MHz

FEATURES

- POWER CONTROL MODES
- 128 x 8 BIT RAM
- 32 PROGRAMMABLE I/O LINES
- TWO 16-BIT TIMER/COUNTERS
- 64 K PROGRAM MEMORY SPACE
- FULLY STATIC DESIGN
- HIGH PERFORMANCE SAJI VI CMOS PROCESS
- BOOLEAN PROCESSOR
- 5 INTERRUPT SOURCES
- PROGRAMMABLE SERIAL PORT
- 64 K DATA MEMORY SPACE
- TEMPERATURE RANGE : Commercial, Industrial, Automotive and Military

DESCRIPTION

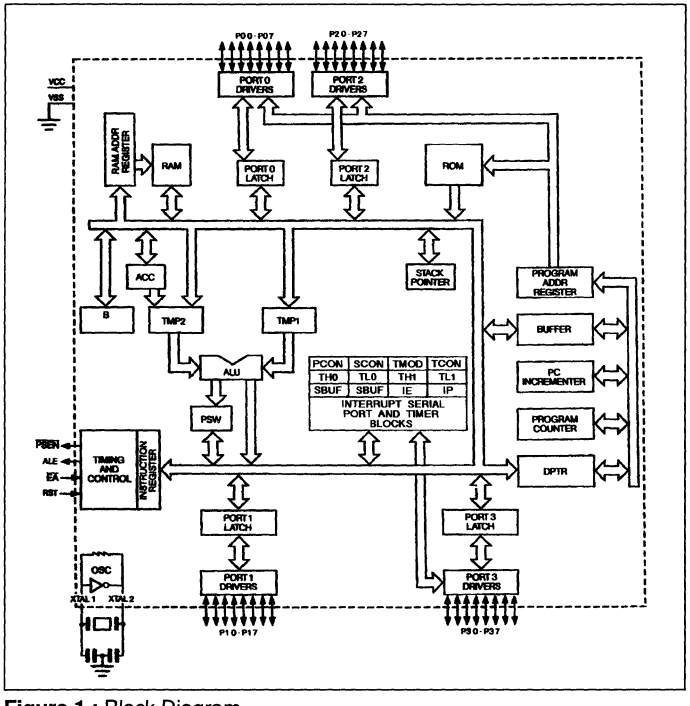


Figure 1 : Block Diagram.

MHS's 80C51 and 80C31 are high performance CMOS versions of the 8051/8031 NMOS single chip 8 bit µC and is manufactured using a self-aligned silicon gate CMOS process (SAJI VI).

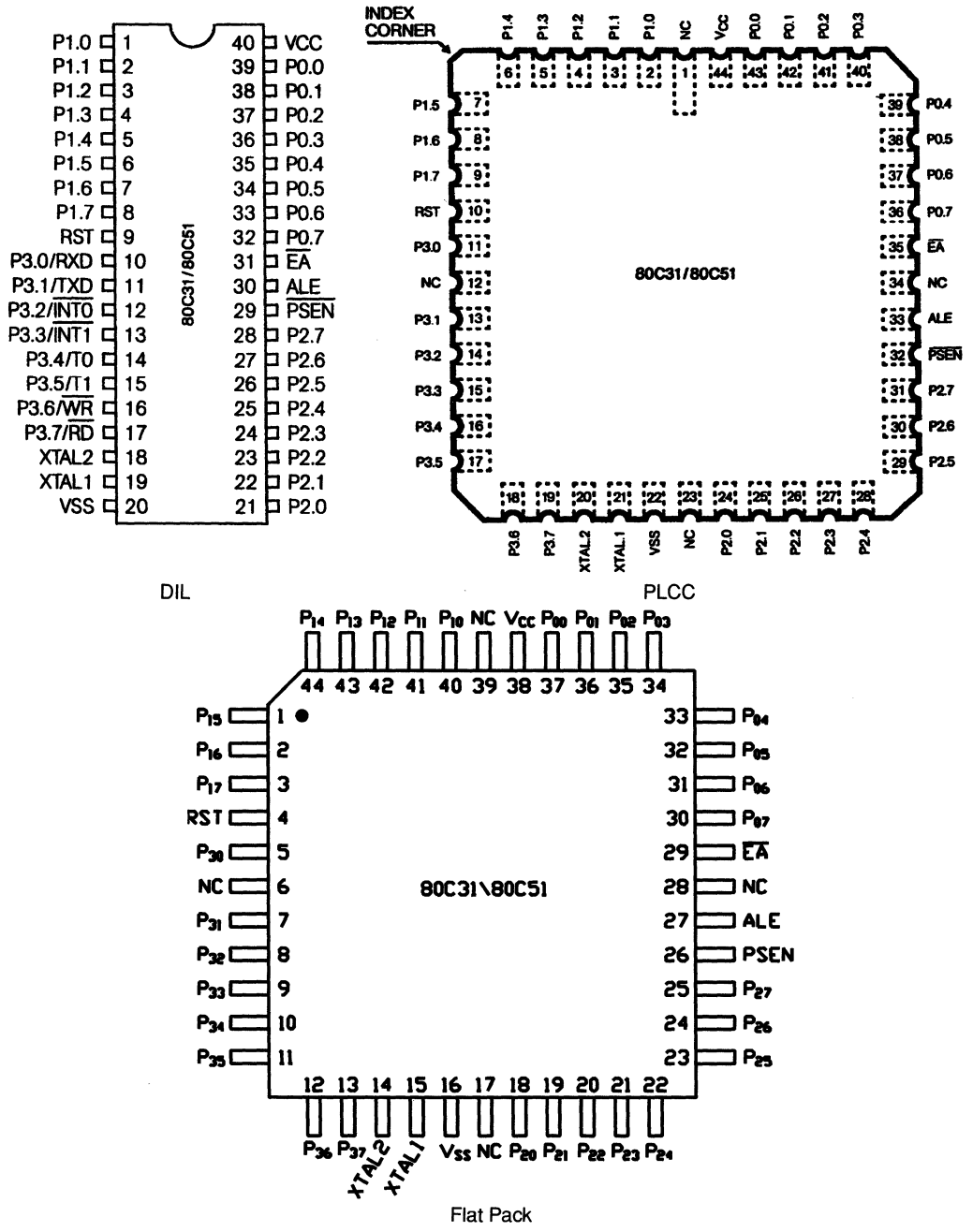
The fully static desing of the MHS 80C51/80C31 allows to reduce system power consumption by bringing the clock frequency down to any value, even DC, without loss of data.

The 80C51 retains all the features of the 8051 : 4 K bytes of ROM ; 128 bytes of RAM ; 32 I/O lines ; two 16 bit timers ; a 5-source 2-level interrupt structure ; a full duplex serial port ; and on-chip oscillator and clock circuits.

In addition, the 80C51 has two software-selectable modes of reduced activity for further reduction in power consumption. In the Idle Mode the CPU is frozen while the RAM, the timers, the serial port and the interrupt system continue to function. In the Power Down Mode the RAM is saved and all other functions are inoperative.

The 80C31 is identical to the 80C51 except that it has no on-chip ROM.

6



Diagrams are for pin reference only. Package sizes are not to scale.

Figure 2 : Configurations.

IDLE AND POWER DOWN OPERATION

Figure 3 shows the internal Idle and Power Down clock configuration. As illustrated, Power Down operation stops the oscillator. Idle mode operation allows the interrupt, serial port, and timer blocks to continue to function while the clock to the CPU is gated off.
These special modes are activated by software via the Special Function Registers, its hardware address is 87H. PCON is not bit addressable.

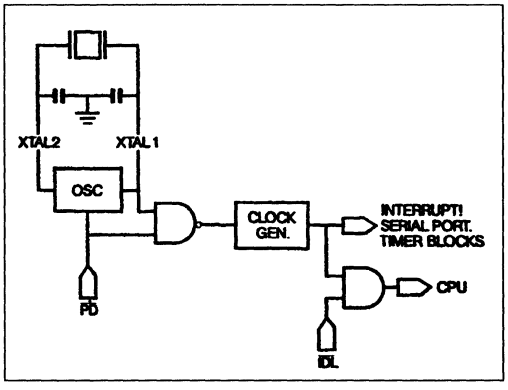


Figure 3 : Idle and Power Down Hardware.

PCON : Power Control Register

(MSB) (LSB)

SMOD	–	–	–	GF1	GF0	PD	IDL

Symbol	Position	Name and Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1, 2 or 3.
–	PCON.6	(Reserved)
–	PCON.5	(Reserved)
–	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activates idle mode operation.

If 1's are written to PD and IDL at the same time. PD takes precedence. The reset value of PCON is (0XXX0000).

MODE	PROGRAM MEMORY	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Port Data	Port Data	Port Data	Port Data
Idle	External	1	1	Floating	Port Data	Address	Port Data
Power Down	Internal	0	0	Port Data	Port Data	Port Data	Port Data
Power Down	External	0	0	Floating	Port Data	Port Data	Port Data

Table 1 : Status of the external pins during Idle and Power Down modes.

6

IDLE MODE

The instruction that sets PCON.0 is the last instruction executed before the mode is activated. Once in the idle mode the CPU status is preserved in its entirety : the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM, and all other registers maintain their data during Idle. Table 1 describes the status of the external pins during Idle mode.

There are two ways to terminate the Idle mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating Idle mode. The interrupt is serviced, and following RETI, the next instruction to be executed will be the one following the instruction that wrote a 1 to PCON.0.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during the Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an enabled interrupt, the service routine can examine the status of the flag bits.

The second way of terminating the Idle mode is with a hardware reset. Since the oscillator is still running, the hardware reset needs to be active for only 2 machine cycles (24 oscillator periods) to complete the reset operation.

POWER DOWN MODE

The instruction that sets PCON.1 is the last executed prior to entering power down. Once is power down, the oscillator is stopped. The contents of the onchip RAM and the Special Function Register is saved during power down mode. A hardware reset is the only way of exiting the power down mode. The hardware reset initiate the Special Function Register (see Table 1).

In the Power Down mode, Vcc may be lowered to minimize circuit power consumption. Care must be taken to ensure the voltage is not reduced until the power

down mode is entered, and that the voltage is restored before the hardware reset is applied which frees the oscillator. Reset should not be released until the oscillator has restarted and stabilized.

Table 1 describes the status of the external pins while in the power down mode. It should be noted that if the power down mode is activated while in external program memory, the port data that is held in the Special Function Register P2 is restored to Port 2. If the data is a 1, the port pin is held high during the power down mode by the strong pullup, T1, shown in *Figure 4.*

STOP CLOCK MODE

Due to static desing, the MHS 80C31/C51 clock speed can be reduced until 0 MHZ without any data loss in memory or registers. This mode allows step by step utilization, and permits to reduce system power consumption by bringing the clock frequency down to any value. At 0 MHz, the power consumption is the same as in the Power Down Mode.

80C51 I/O PORTS

The I/O port drive of the 80C51 is similar to the 8051. The I/O buffers for Ports 1, 2 and 3 are implemented as shown in figure 4.

When the port latch contains a 0, all pFETS in *figure 4* are off while the nFET is turned on. When the port latch makes a 0-to-1 transition, the nFET turns off. The strong pullup pFET, T1, turns on for two oscillator periods, pulling the output high very rapidly. As the output line is drawn high, pFET T3 turns on through the inverter to supply the I_{OH} source current. This inverter and T3 form a latch which holds the 1 and supported by T2.

When Port 2 is used as an address port, for access to external program of data memory, any address bit that contains a 1 will have his strong pullup turned on for the entire duration of the external memory access.

When an I/O pin on Ports 1, 2, or 3 is used as an input, the user should be aware that the external circuit must sink current during the logical 1-to-0 transition. The maximum sink current is specified as ITL under the D.C. Specifications. When the input goes below approximately 2 V, T3 turns off to save ICC current. Note, when returning to a logical 1, T2 is the only internal pullup that is on. This will result in a slow rise time if the user's circuit does not force the input line high.

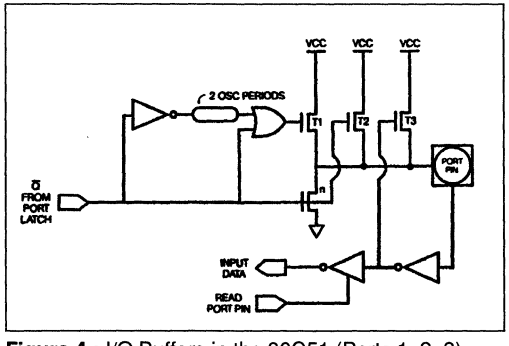


Figure 4 : I/O Buffers in the 80C51 (Ports 1, 2, 3).

80C31/80C51 PINS DESCRIPTION

V_{SS}

Circuit ground potential

VCC

Supply voltage during normal, Idle, and Power Down operation.

Port 0

Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impendance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1's. Port 0 also outputs the code bytes during program verification in the 80C51. External pullups are required during program verification. Port 0 can sink eight LS TTL inputs.

Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. Port 1 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address bytes during program verification. In the 80C51, Port 1 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. Port 2 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups. Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that uses 16-bit addresses (MOVX @ DPTR). In this application, it uses strong internal pullups when emitting 1's. During accesses to external Data Memory that uses 8-bit addresses (MOVX @ Ri), Port 2 emits the contents of the P2 Special Function Register.

It also receives the high-order address bits and control signals during program verification in the 80C51. Port 2 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. Port 3 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the pullups. It also serves the functions of various special features of the MHS-51 Family, as listed below.

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	WR (external Data Memory write strobe)
P3.7	RD (external Data Memory read strobe)

Port 3 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

RST

A high level on this for two machine cycles while the oscillator is running resets the device. An internal pulldown resistor permits Power-On reset using only a capacitor connected to V_{CC}.

ALE

Address Latch Enable output for latching the low byte of the address during accesses to external memory. ALE is activated as though for this purpose at a constant rate of 1/6 the oscillator frequency except during an external data memory access at which time one ALE pulse is skipped. ALE can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

$\overline{PSEN}$

Program Store Enable output is the read strobe to external Program Memory. PSEN is activated twice each machine cycle during fetches from external Program Memory. (However, when executing out of external Program Memory, two activations of PSEN are skipped during each access to external Data Memory). PSEN is not activated during fetches from internal Program Memory. PSEN can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

$\overline{EA}$

When EA is held high, the CPU executes out of internal Program Memory (unless the Program Counter exceeds 0FFFH). When EA is held low, the CPU executes only out of external Program Memory. EA must not be floated.

XTAL1

Input to the inverting amplifier that forms the oscillator. Receives the external oscillator signal when an external oscillator is used.

XTAL2

Output of the inverting amplifier that forms the oscillator, and input to the internal clock generator. This pin should be floated when an external oscillator is used.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output respectively, of an inverting amplifier which is configured for use as an on-chip oscillator, as shown in *figure 5*. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected as shown in *figure 6*. There is no requirement on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

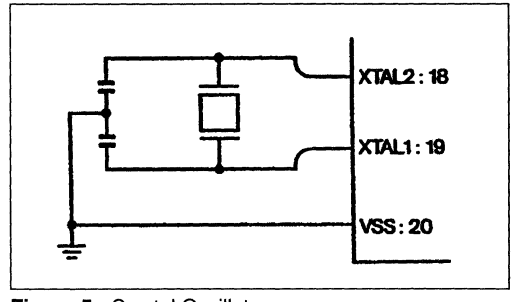


Figure 5 : Crystal Oscillator.

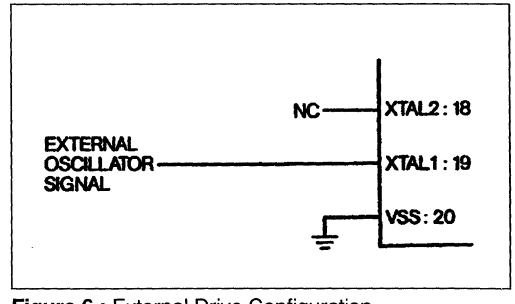


Figure 6 : External Drive Configuration.

6

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias :
C = Commerical..... 0°C to 70°C
I = Industrial..... – 40°C to 85°C
Storage Temperature..... – 65°C to + 150°C
Voltage on V_{CC} to V_{SS}..... – 0.5 V to + 7 V
Voltage on Any Pin to V_{SS}..... – 0.5 V to V_{CC} + 0.5 V
Power Dissipation..... 1 W**
** This value is based on the maximum allowable die temperature and the thermal resistance of the package.

*** NOTICE :**

Stresses at or above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

DC CHARACTERISTICS

T_A = – 40°C to 85°C ; VCC = 5 V ± 20 % ; VSS = 0 V ; F = 0 to 12 MHz
T_A = – 40°C to 85°C ; VCC = 5 V ± 10 % ; VSS = 0 V ; F = 0 to 16 MHz

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITIONS
VIL	Input Low Voltage	– 0.5	0.2 VCC – 0.1	V	
VIH	Input High Voltage (Except XTAL and RST)	0.2 VCC + 0.9	VCC + 0.5	V	
VIH1	Input High Voltage (RST and XTAL1)	0.7 VCC	VCC + 0.5	V	
VOL	Output Low Voltage (Ports1, 2, 3)		0.45	V	IOL = 1.6 mA (note3)
VOL1	Output Low Voltage Port 0, ALE, $\overline{PSEN}$		0.45	V	IOL = 3.2 mA (note 3)
VOH	Output High Voltage Ports 1, 2, 3	0.9 VCC		V	IOH = – 10 µA
		0.75 VCC		V	IOH = – 25 µA
		2.4		V	IOH = – 60 µA VCC = 5 V ± 10 %
VOH1	Output High Voltage (Port 0, ALE, PSEN)	0.9 VCC		V	IOH = – 80 µA
		0.75 VCC		V	IOH = – 300 µA
		2.4		V	IOH = – 800 µA VCC = 5 V ± 10 %
IIL	Logical 0 Input Current Ports 1, 2, 3		C – 50 I – 60	µA	Vin = 0.45 V
ILI	Input Leakage Current (Port 0, EA)		± 10	µA	0.45 < Vin < VCC
ITL	Logical 1 to 0 Transition Current (Ports 1, 2, 3)		– 650	µA	Vin = 2.0 V
IPD	Power Supply Current (Power Down Mode)		50	µA	VCC = 2.0 V to 6 V (note 2)
RRST	RST Pulldown Resistor	50	150	kΩ	
CIO	Capacitance of I/O Buffer		10	pF	f_C = 1 MHz, T_A = 25°C
ICC	Power supply current Active mode 12 MHz Idle mode 12 MHz		 20 5	 mA mA	(notes 1, 2)

Note 1 : ICC max is given by :
Active Mode : ICCMAX = 1.47 x FREQ + 2.35
Idle Mode : ICCMAX = 0.33 x FREQ + 1.05
where FREQ is the external oscillator frequency in MHz. ICCMAX is given in mA. See *Figure 1.*
See figures *1* through *5* for ICC test conditions.

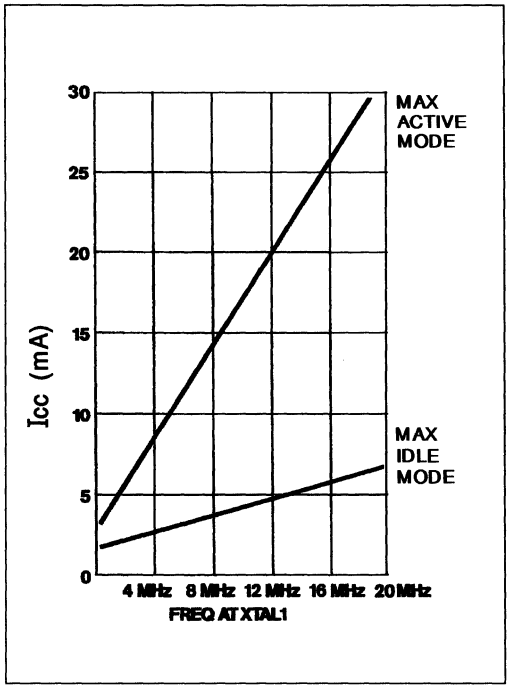


Figure 1 : ICC vs. Frequency. Valid only within frequency specifications of the device under test.

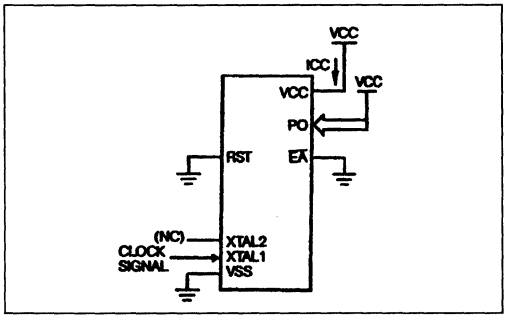


Figure 2 : ICC Test Condition, Idle Mode. All other pins are disconnected.

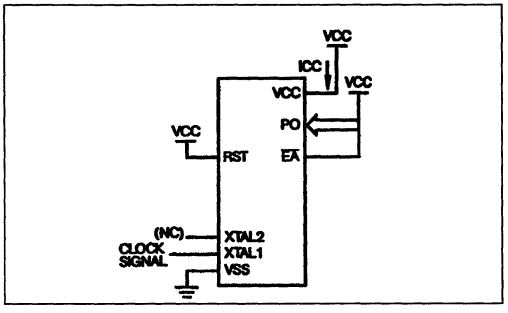


Figure 3 : ICC Test Condition, Active Mode. All other pins are disconnected.

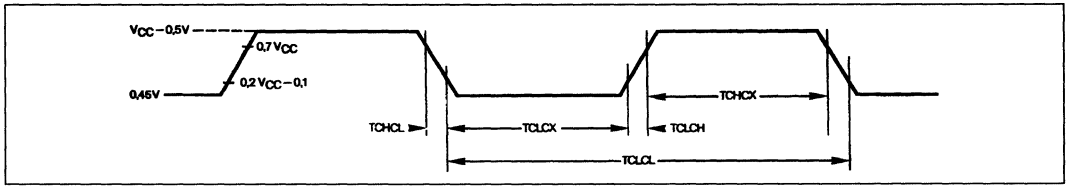


Figure 4 : Clock Signal Waveform for ICC Tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns.

Note 2 : ICC is measured with all output pins disconnected ; XTAL1 driven with TCLCH, TCHCL = 5ns, VIL = VSS + .5 V, VIH = VCC – .5V ; XTAL2 N.C ; EA = RST = Port 0 = VCC .ICC would be slightly higher if a crystal oscillator used. Idle ICC is measured with all output pins disconnected ; XTAL1 driven with TCLCH = TCHCL = 5ns, VIL = VSS + .5V,
VIH = VCC – .5V ; XTAL2 N.C ; Port 0 = VCC ; EA = RST = VSS.
Power Down ICC is measured with all output pins disconnected ; EA = PORT0 = VCC ; XTAL2 N.C. ; RST = VSS.

Note 3 : Capacitance loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OLS} of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1 to 0 transitions during bus operations. In the worst case (capacitive loading 100 pF), the noise pulse on the ALE line may exceed 0.45 V with maxi VOL peak 0.6 V. A Schmitt Trigger use is not necessary.

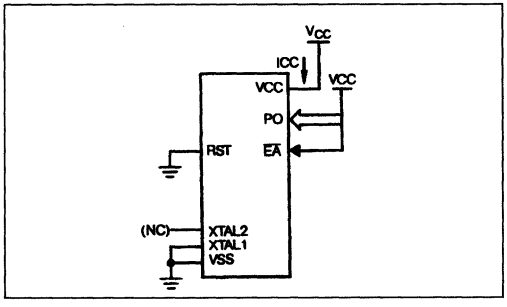


Figure 5 : ICC Test Condition, Power Down Mode. All other pins are disconnected.

EXTERNAL CLOCK DRIVE CHARACTERISTICS (XTAL 1)

SYMBOL	PARAMETER	VARIABLE CLOCK FREQ = 0 to 16 MHz		UNIT
		MIN	MAX	
1/TCLCL	Oscillator Frequency	62.5		ns
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

A.C. PARAMETERS :
TA = – 40°C + 85°C ; VSS = 0 V ; VCC = 5 V ± 20 % ; F = 0 to 12 MHz
TA = – 40°C + 85°C ; VSS = 0 V ; VCC = 5 V ± 10 % ; F = 0 to 16 MHz
(Load Capacitance for Port 0, ALE, and $\overline{PSEN}$ = 100 pf ; Load Capacitance for All Other Outputs = 80 pf).

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

SYMBOL	PARAMETER	MIN	MAX	UNIT
TLHLL	ALE Pulse Width	2TCLCL-40		ns
TAVLL	Address Valid to ALE	TCLCL-55		ns
TLLAX	Address Hold After ALE	TCLCL-35		ns
TLLIV	ALE to Valid Instr in		4TCLCL-100	ns
TLLPL	ALE to $\overline{PSEN}$	TCLCL-40		ns
TPLPH	$\overline{PSEN}$ Pulse Width	3TCLCL-45		ns
TPLIV	$\overline{PSEN}$ to Valid Instr in		3TCLCL-105	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		TCLCL-25	ns
TPXAV	PSEN to Address Valid	TCLCL-8		ns
TAVIV	Address to Valid Instr in		5TCLCL-105	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		10	ns

6

EXTERNAL DATA MEMORY CHARACTERISTICS

SYMBOL	PARAMETER	MIN	MAX	UNIT
TRLRH	$\overline{RD}$ Pulse Width	6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	6TCLCL-100		ns
TLLAX	Data Address Hold After ALE	TCLCL-50		ns
TRLDV	$\overline{RD}$ to Valid Data in		5TCLCL-165	ns
TRHDX	Data Hold After $\overline{RD}$	0		ns
TRHDZ	Data Float After $\overline{RD}$		2TCLCL-70	ns
TLLDV	ALE to Valid Data in		8TCLCL-150	ns
TAVDV	Address to Valid Data in		9TCLCL-165	ns
TLLWL	ALE to $\overline{WR}$ or $\overline{RD}$	3TCLCL-50	3TCLCL+50	ns
TAVWL	Address to $\overline{WR}$ or $\overline{RD}$	4TCLCL-130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	TCLCL-60		ns
TQVWH	Data Setup to $\overline{WR}$ High	7TCLCL-150		ns
TWHQX	Data Hold After $\overline{WR}$	TCLCL-50		ns
TRLAZ	RD Low to Address Float		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	TCLCL-40	TCLCL+40	ns

ABSOLUTE MAXIMUM RATINGS*

Ambiant Temperature Under Bias :

M = Military..... – 55°C to + 125°C

A = Automotive..... – 40°C to + 125°C

Storage Temperature. ..... – 65°C to + 150°C

Voltage on Any Pin to V_{SS}. ..... – 0.5 V to V_{CC} + 0.5 V

Voltage on V_{CC} to V_{SS} .....–- 0.5 V to 6.5 V

Power Dissipation..... 200 mW

*** NOTICE :**

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these orany other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC CHARACTERISTICS

TA = – 55°C to + 125°C ; VSS = 0 V ; VCC = 5 V ± 10 %

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITIONS
VIL	Input Low Voltage	– 0.5	0.2 VCC – 0.1	V	
VIH	Input High Voltage (Except XTAL1, RST)	0.2 VCC + 0.9	VCC + 0.5	V	
VIH1	Input High Voltage (XTAL1, RST)	0.7 VCC	VCC + 0.5	V	
VOL	Output Low Voltage (Ports 1, 2, 3)		0.45	V	IOL = 1.6 mA (note 3)
VOL1	Output Low Voltage (Port 0, ALE, $\overline{PSEN}$)		0.45	V	IOL = 3.2 mA (note 3)
VOH	Output High Voltage (Ports 1, 2, 3)	2.4		V	IOH = – 60 µA VCC = 5 V ± 10 %
		0.75 VCC		V	IOH = – 25 µA
		0.9 VCC		V	IOH = – 10 µA
VOH1	Output High Voltage (Port 0 in External Bus Mode, ALE, PEN)	2.4		V	IOH = – 800 µA VCC = 5 V ± 10 %
		0.75 VCC		V	IOH = – 300 µA
		0.9 VCC		V	IOH = 80 µA
IIL	Logical 0 Input Current Ports 1, 2, 3		– 75	µA	Vin = 0.45 V
ITL	Logical 1 to 0 Transition Current (Ports 1, 2, 3)		– 750	µA	Vin = 2 V
ILI	Input Leakage Current (Port 0, $\overline{EA}$)		± 10	µA	0.45 < Vin < VCC
RRST	Reset Pulldown Resistor		50	150	kΩ
CIO	Pin Capacitance		10	pF	Test Freq = 1 MHz, T_A = 25°C
IPD	Power Down Current		75	µA	VCC = 2 V to 5.5 V
ICC	Power supply current Active mode 12 MHz Idle mode 12 MHz		 21 7	 mA mA	 VCC = 5.5 V VCC = 5.5 V

6

AC PARAMETERS :

TA = – 55°C + 125°C ; VSS = 0 V ; VCC = 5 V ± 10 %
(Load Capacitance for Port 0, ALE, and PSEN = 100 pf ; Load Capacitance for All Other Outputs = 80 pf).

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

FREQ = 12 MHz (MAX)

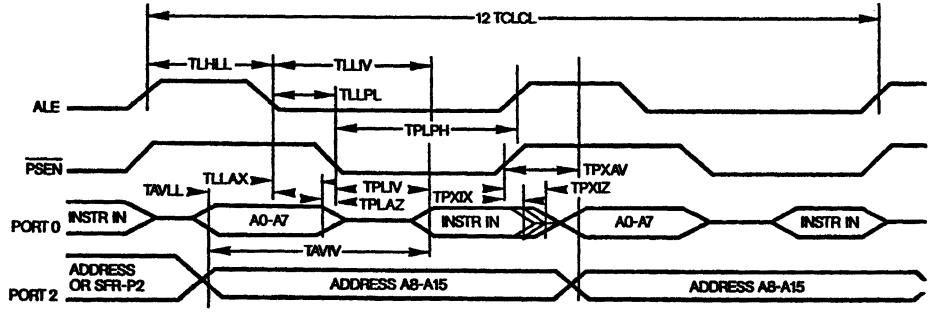
SYMBOL	PARAMETER	MIN	MAX	UNIT
TLHLL	ALE Pulse Width	2TCLCL-55		ns
TAVLL	Address Valid to ALE	TCLCL-70		ns
TLLAX	Address Hold After ALE	TCLCL-50		ns
TLLIV	ALE to Valid Instr in		4TCLCL-115	ns
TLLPL	ALE to $\overline{PSEN}$	TCLCL-55		ns
TPLPH	$\overline{PSEN}$ Pulse Width	3TCLCL-60		ns
TPLIV	$\overline{PSEN}$ to Valid Instr in		3TCLCL-120	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		TCLCL-40	ns
TPXAV	$\overline{PSEN}$ to Address Valid	TCLCL-8		ns
TAVIV	Address to Valid Instr in		5TCLCL-120	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		25	ns

EXTERNAL DATA MEMORY CHARACTERISTICS

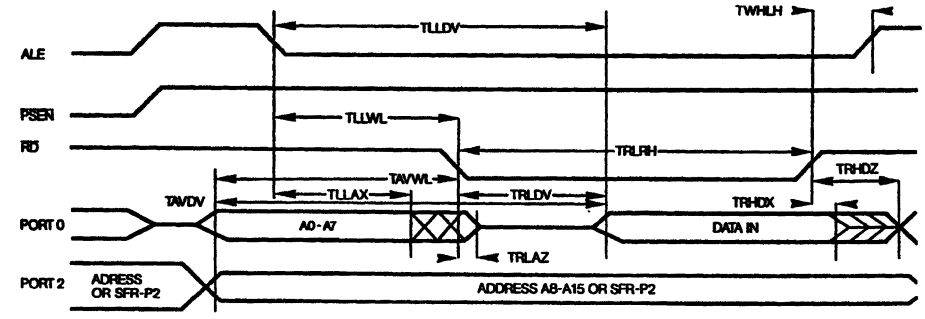
SYMBOL	PARAMETER	MIN	MAX	UNIT
TRLRH	$\overline{RD}$ Pulse Width	6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	6TCLCL-100		ns
TLLAX	Data Address Hold After ALE	TCLCL-50		ns
TRLDV	$\overline{RD}$ to Valid Data in		5TCLCL-185	ns
TRHDX	Data Hold After $\overline{RD}$	0		ns
TRHDZ	Data Float After $\overline{RD}$		2TCLCL-85	ns
TLLDV	ALE to Valid in		8TCLCL-170	ns
TAVDV	Address to Valid Data in		9TCLCL-185	ns
TLLWL	ALE to $\overline{WR}$ or $\overline{RD}$	3TCLCL-65	3TCLCL+65	ns
TAVWL	Address to $\overline{WR}$ or $\overline{RD}$	4TCLCL-145		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	TCLCL-75		ns
TQVWH	Data Setup to $\overline{WR}$ High	7TCLCL-150		ns
TWHQX	Data Hold After $\overline{WR}$	TCLCL-65		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	TCLCL-65	TCLCL+65	ns

AC TIMING DIAGRAMS

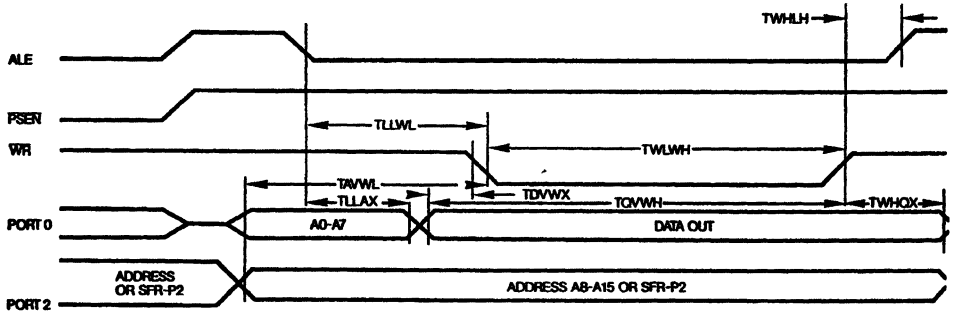
EXTERNAL PROGRAM MEMORY READ CYCLE



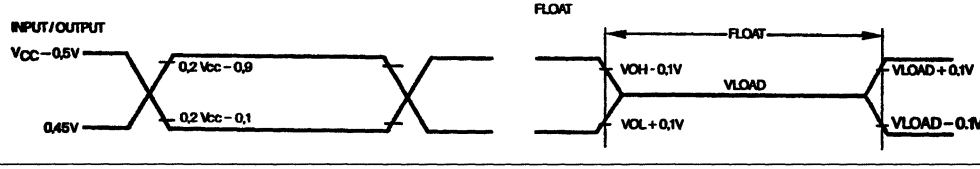
EXTERNAL DATA MEMORY READ CYCLE



EXTERNAL DATA MEMORY WRITE CYCLE



AC TESTING INPUT/OUTPUT, FLOAT WAVEFORMS



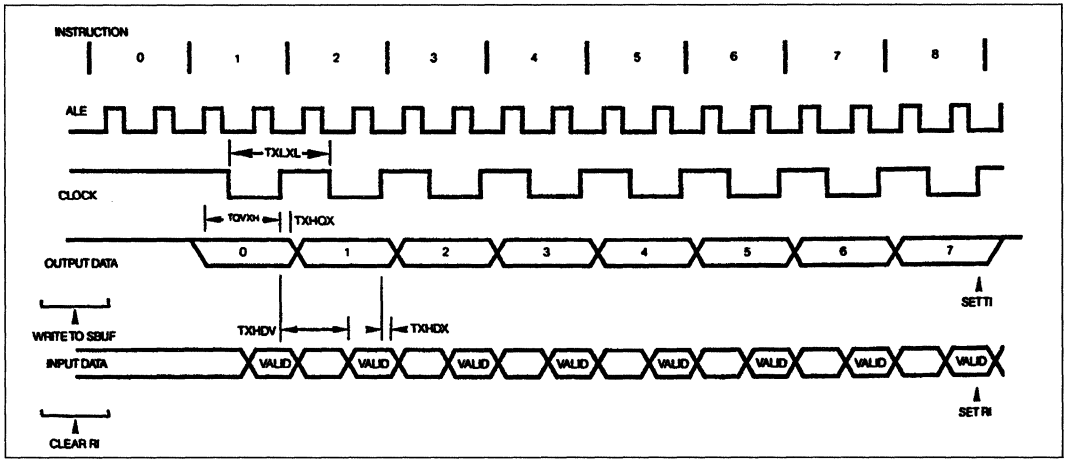
AC inputs during testing are driven at V_{CC} – 0.5 for a logic "1" and 0.45 V for a logic "0". Timing measurements are made at VIH min for a logic "1" and VIL max for a logic "0". For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change from the loaded VOH/VOL level occurs. Iol/IoH ≥ ± 20 mA.

6

SERIAL PORT TIMING - SHIFT REGISTER MODE

SYMBOL	PARAMETER	MIN	MAX	UNIT
TXLXL	Serial Port Clock Time	12TCLCL		µs
TQVXH	Output Data Setup to Clock Rising Edge	10TCLCL-133		ns
TXHQX	Output Data Hold after Clock Rising Edge	2TCLCL-117		ns
TXHDX	Input Data Hold after Clock Rising Edge	0		ns
TXHDV	Clock Rising Edge to Input Data Valid		10TCLCL-133	ns

SHIFT REGISTER TIMING WAVEFORMS



6

EXPLANATION OF THE AC SYMBOL

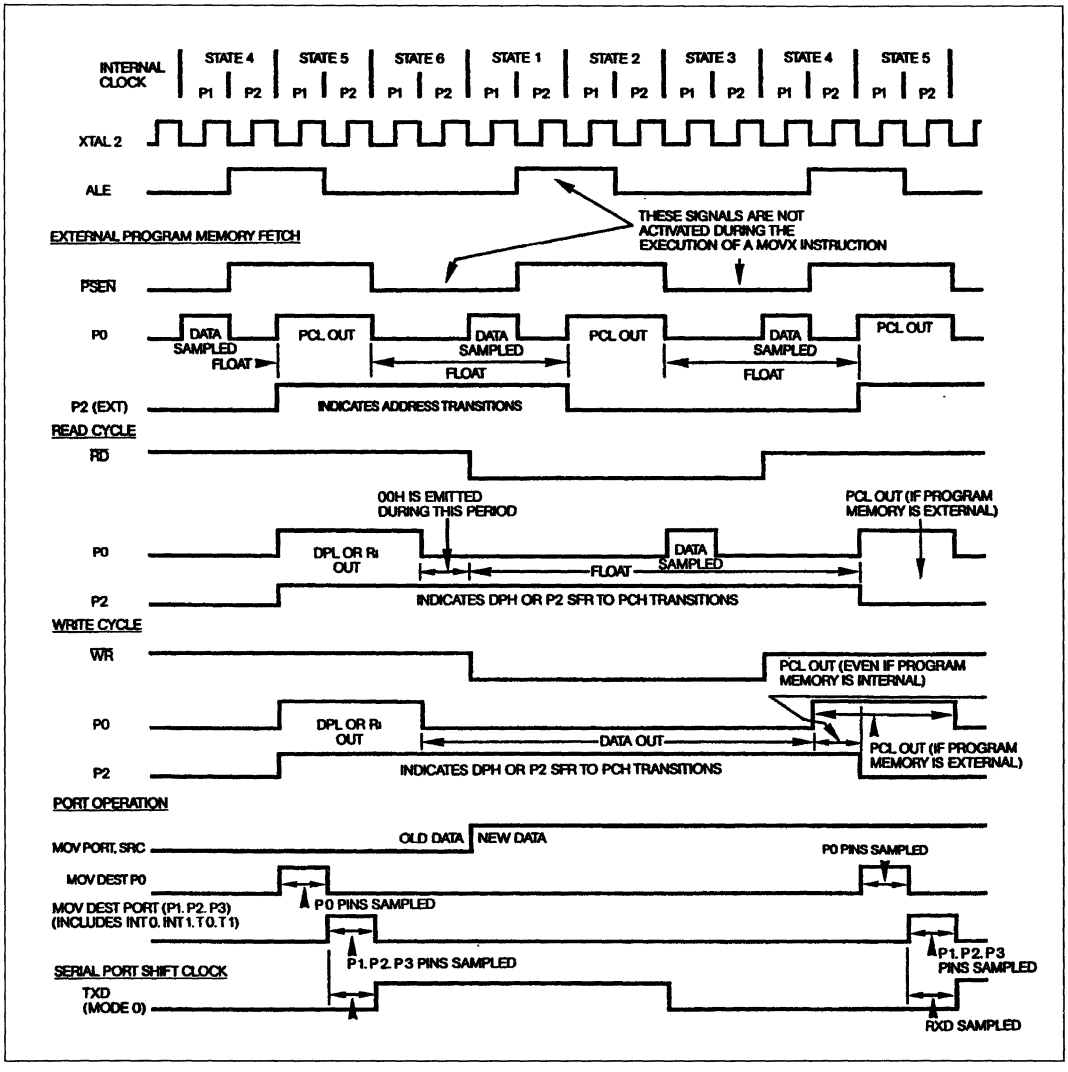
Each timing symbol has 5 characters. The first character is always a "T" (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

Example :

TAVLL = Time for Address Valid to ALE low.
TLLPL = Time for ALE low to PSEN low.

A : Address.
C : Clock
D : Input data.
H : Logic level HIGH.
I : Instruction (program memory contents).
L : Logic level LOW, or ALE.
P : PSEN.
Q : Output data.
R : READ signal.
T : Time
V : Valid.
W : WRITE signal.
X : No longer a valid logic level.
Z : Float.

CLOCK WAVEFORMS



This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins, however, ranges from 25 to 125 ns. This propagation delay is dependant on variables such as temperature and pin loading. Propagation also varies from output to output and component. Typically though (T_A = 25°C fully loaded) RD and WR propagation delays are approximately 50 ns. The other signals are typically 85 ns. Propagation delays are incorporated in the AC specifications.

ARITHMETIC OPERATIONS				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with Carry	1	1
ADDC	A,direct	Add direct byte to A with Carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with Carry flag	1	1
ADDC	A,#data	Add immediate data to A with Carry flag	2	1
SUBB	A,Rn	Subtract register from A with Borrow	1	1
SUBB	A,direct	Subtract direct byte from A with Borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A with Borrow	1	1
SUBB	A,#data	Subtract immed. data from A with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Incriment indirect RAM	1	1
INC	DPTR	Incriment Data Pointer	1	2
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1
LOGICAL OPERATIONS				
MNEMONIC		**DESTINATION**	**BYTE**	**CYC**
ANL	A,Rn	AND register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate A Left through the Carry flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate A Right through Carry flag	1	1
SWAP	A	Swap nibbles within the Accumulator	1	1

Table 1 : MHS C51 Instruction Set Description.

DATA TRANSFER				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
MOV	A,Rn	Move register to Accumulator	1	1
MOV	A,direct	Move direct byte to Accumulator	2	1
MOV	A,@Ri	Move indirect RAM to Accumulator	1	1
MOV	A,#data	Move immediate data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move Accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct byte	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data 16	Load Data Pointer with a 16-bit constant	3	2
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A	1	2
MOVC	A,@A+PC	Move Code byte relative to PC to A	1	2
MOVX	A,@Ri	Move External RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move External RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to External RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to External RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with Accumulator	1	1
XCH	A,direct	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order nibble ind RAM with A	1	1
BOOLEAN VARIABLE MANIPULATION				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
CLR	C	Clear Carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry flag	1	1
SETB	bit	Set direct Bit	2	1
CPL	C	Complement Carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry flag	2	2
ANL	C, /bit	AND complement of direct bit to Carry	2	2
ORL	C, bit	OR direct bit to Carry flag	2	2
ORL	C, /bit	OR complement of direct bit to Carry	2	2
MOV	C, bit	Move direct bit to Carry flag	2	1
MOV	bit,C	Move Carry flag to direct bit	2	2
PROGRAM AND MACHINE CONTROL				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
ACALL	addr 11	Absolute Subroutine Call	2	2
LCALL	addr 16	Long Subroutine Call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr 11	Absolute Jump	2	2
LJMP	addr 16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is Zero	2	2
JNZ	rel	Jump if Accumulator is Not Zero	2	2
JC	rel	Jump if Carry flag is set	2	2
JNC	rel	Jump if No Carry flag	2	2

Table 1. (Cont.)

PROGRAM AND MACHINE CONTROL (cont.)				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
JB	bit,rel	Jump if direct Bit set	3	2
JNB	bit,rel	Jump if direct Bit Not set	3	2
JBC	bit,rel	Jump if direct Bit is set & Clear bit	3	2
CJNE	A,direct,rel	Compare direct to A & Jump if Not Equal	3	2
CJNE	A,#data,rel	Comp. immed. to A & Jump if Not Equal	3	2
CJNE	Rn,#data,rel	Comp. immed. to reg & Jump if Not Equal	3	2
CJNE	@Ri,#data.rel	Comp. immed. to ind. & Jump if Not Equal	3	2
DJNZ	Rn,rel	Decrement register & Jump if Not Zero	2	2
DJNZ	direct.rel	Decrement direct & Jump if Not Zero	3	2
NOP		No operation	1	1

Table 1. (Cont.)

Notes on data addressing modes :

Rn – Working register R0-R7
direct – 128 internal RAM locations, any I/O port, control or status register
@Ri – Indirect internal RAM location addressed by register R0 or R1
#data – 8-bit constant included in instruction
#data 16 – 16-bit constant included as bytes 2 & 3 of instruction
bit – 128 software flags, any I/O pin, control or status bit

Notes on program addressing modes :

addr 16 – Destination address for LCALL & LJMP may be anywhere within the 64-k program memory address space
Addr 11 – Destination address for ACALL & AJMP will be within the same 2-k page of program memory as the first byte of the following instruction
rel – SJMP and all conditional jumps include an 8-bit offset byte. Range is + 127 – 128 bytes relative to the first byte of the following instruction.

6

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr,code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr,code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A,data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6
2F	1	ADD	A,R7
30	3	JNB	bit addr,code addr
31	2	ACALL	code addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A,#data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr,A
43	3	ORL	data addr,#data
44	2	ORL	A,#data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr,A
53	3	ANL	data addr,#data
54	2	ANL	A,#data
55	2	ANL	A,data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	code addr
61	2	AJMP	code addr
62	2	XRL	data addr A
63	3	XRL	data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr

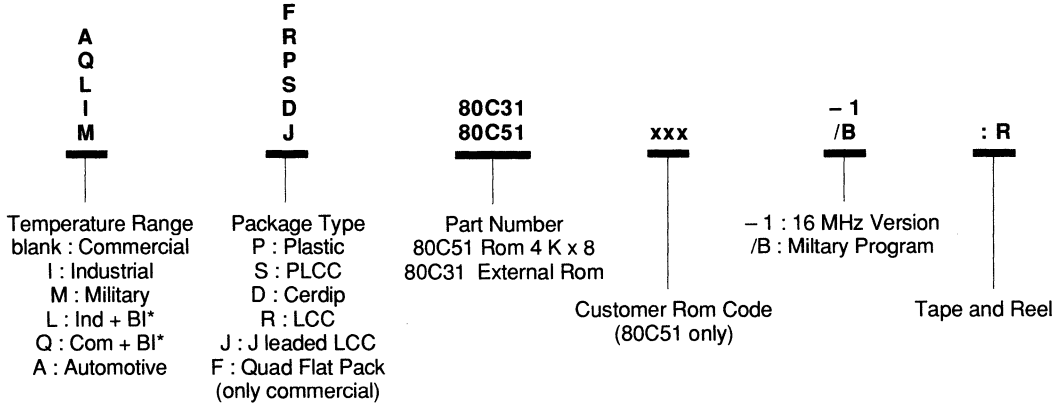
Table 2 : Instruction Opcodes in Hexadecimal Order.

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C,bit addr
73	1	JMP	@A + DPTR
74	2	MOV	A,#data
75	3	MOV	data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C,bit addr
83	1	MOVC	A,@A + PC
84	1	DIV	AB
85	3	MOV	data addr,data addr
86	2	MOV	data addr,@R0
87	2	MOV	data addr,@R1
88	2	MOV	data addr,R0
89	2	MOV	data addr,R1
8A	2	MOV	data addr,R2
8B	2	MOV	data addr,R3
8C	2	MOV	data addr,R4
8D	2	MOV	data addr,R5
8E	2	MOV	data addr,R6
8F	2	MOV	data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	code addr
92	2	MOV	bit addr,C
93	1	MOVC	A,@A + DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,bit addr
A1	2	AJMP	code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,bit addr
B1	2	ACALL	code addr
B2	2	CPL	Bit addr
B3	1	CPL	C
B4	3	CJNE	A,#data,code addr
B5	3	CJNE	A,data addr,code addr
B6	3	CJNE	@R0,#data,code addr
B7	3	CJNE	@R1,#data,code addr
B8	3	CJNE	R0,#data,code addr
B9	3	CJNE	R1,#data,code addr
BA	3	CJNE	R2,#data,code addr
BB	3	CJNE	R3,#data,code addr
BC	3	CJNE	R4,#data,code addr
BD	3	CJNE	R5,#data,code addr
BE	3	CJNE	R6,#data,code addr
BF	3	CJNE	R7,#data,code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3

Table 2. (Cont.)

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7, A

Table 2. (Cont.)



* BI : Burn-In

6

DATA SHEET

80C51S/80C31S

CMOS SINGLE-CHIP 8 BIT MICROCONTROLLER

- **80C51S-CMOS SINGLE-CHIP 8 BIT MICRO-CONTROLLER with factory mask-programmable ROM**
- **80C31S-ROM LESS VERSION OF THE 80C51**
- **80C51S/80C31S : 0 To 20 MHz**

FEATURES

- **POWER CONTROL MODES**
- **128 X 8 BIT RAM**
- **32 PROGRAMMABLE I/O LINES**
- **TWO 16-BIT TIMER/COUNTERS**
- **64 K PROGRAM MEMORY SPACE**
- **FULLY STATIC DESIGN**
- **HIGH PERFORMANCE SAJI VI CMOS PROCESS**
- **BOOLEAN PROCESSOR**
- **5 INTERRUPT SOURCES**
- **PROGRAMMABLE SERIAL PORT**
- **64 K DATA MEMORY SPACE**
- **TEMPERATURE RANGE : Commercial**

DESCRIPTION

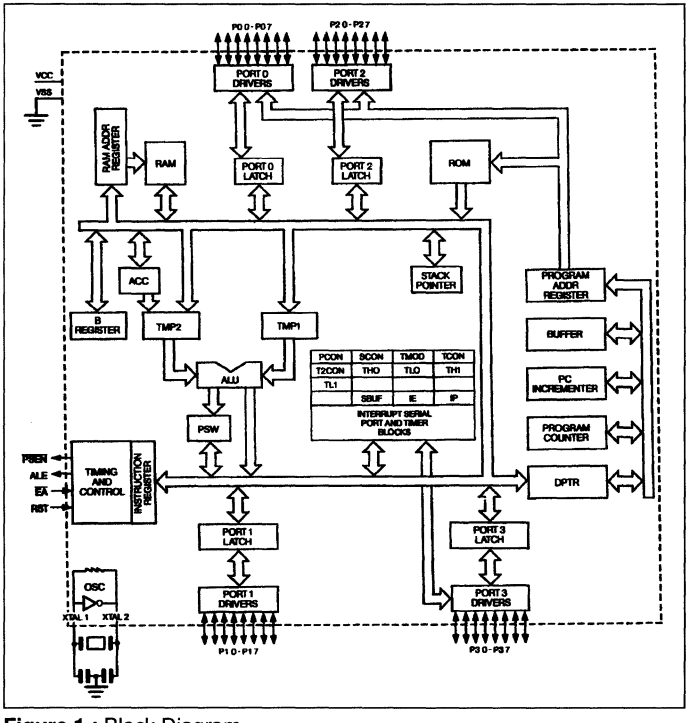


Figure 1 : Block Diagram.

MHS's 80C51S and 80C31S are high performance CMOS versions of the 8051/8031 NMOS single chip 8 bit µC and is manufactured using a self-aligned silicon gate CMOS process (SAJI VI).

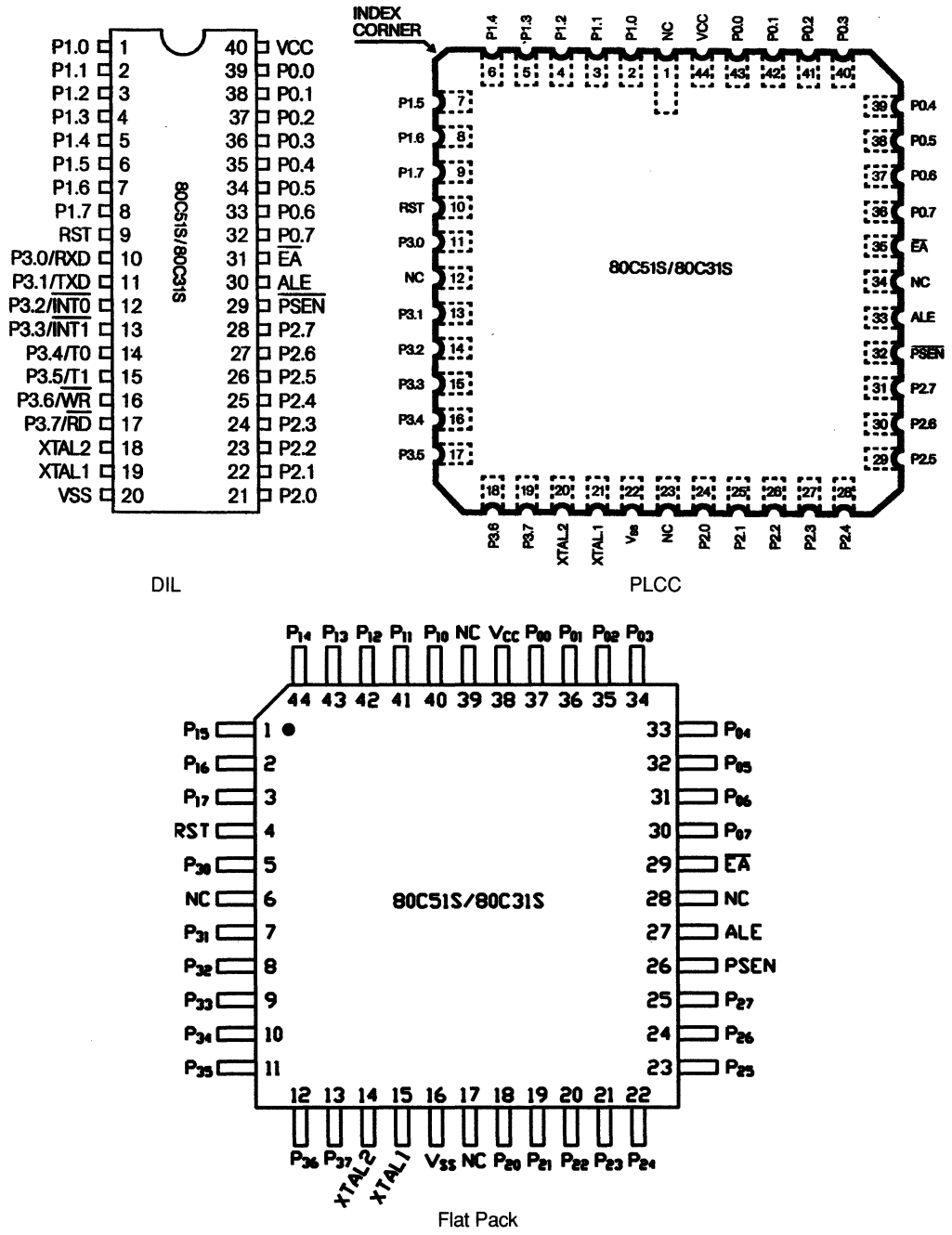
The fully static design of the MHS 80C51S/80C31S allows to reduce system power consumption by bringing the clock frequency down to any value, even DC, without loss of data.

The 80C51S retains all the features of the 8051 : 4 K bytes of ROM ; 128 bytes of RAM ; 32 I/O lines ; two 16 bit timers ; a 5-source, 2-level interrupt structure ; a full duplex serial port ; and on-chip oscillator and clock circuits.

In addition, the 80C51S has two software-selectable modes of reduced activity for further reduction in power consumption. In the Idle Mode the CPU is frozen while the RAM, the timers, the serial port, and the interrupt system continue to function. In the Power Down Mode the RAM is saved and all other functions are inoperative.

The 80C31S is identical to the 80C51S except that it has no on-chip ROM.

6



Diagrams are for reference only. Package sizes are not to scale.

Figure 2 : Configurations.

IDLE AND POWER DOWN OPERATION

Figure 3 shows the internal Idle and Power Down clock configuration. As illustrated, Power Down operation stops the oscillator. Idle mode operation allows the interrupt, serial port, and timer blocks to continue to function while the clock to the CPU is gated off.
These special modes are activated by software via the Special Function Register, PCON. Its hardware address is 87 H. PCON is not bit addressable.

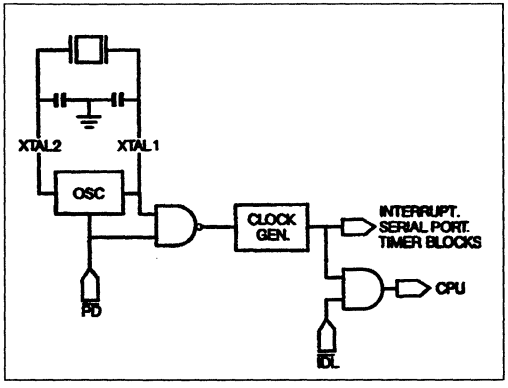


Figure 3 : Idle and Power Down Hardware.

PCON : Power Control Register

(MSB) (LSB)

SMOD	–	–	–	GF1	GF0	PD	IDL

Symbol	Position	Name and Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1, 2 or 3.
–	PCON.6	(Reserved)
–	PCON.5	(Reserved)
–	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activates idle mode operation.

If 1's are written to PD and IDL at the same time. PD takes precedence. The reset value of PCON is (0XXX0000).

MODE	PROGRAM MEMORY	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Port Data	Port Data	Port Data	Port Data
Idle	External	1	1	Floating	Port Data	Address	Port Data
Power Down	Internal	0	0	Port Data	Port Data	Port Data	Port Data
Power Down	External	0	0	Floating	Port Data	Port Data	Port Data

Table 1 : Status of the external pins during Idle and Power Down modes.

IDLE MODE

The instruction that sets PCON.0 is the last instruction executed before the Idle mode is activated. Once in the Idle mode the CPU status is preserved in its entirety : the Stock Pointer, Program Counter, Program Status Word, Accumulator, RAM, and all other registers maintain their data during Idle. *Table 1* describes the status of the external pins during Idle mode.

There are two ways to terminate the Idle mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating Idle mode. The interrupt is serviced, and following RETI, the next instruction to be executed will be the one following the instruction that wrote a 1 to PCON.0.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during the Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an enabled interrupt, the service routine can examine the status of the flag bits.

The second way of terminating the Idle mode is with a hardware reset. Since the oscillator is still running, the hardware reset needs to be active for only 2 machine cycles (24 oscillator periods) to complete the reset operation.

POWER DOWN MODE

The instruction that sets PCON.1 is the last executed prior to entering power down. Once in power down, the oscillator is stopped. The contents of the onchip RAM and the Special Function Register is saved during power down mode. A hardware reset is the only way of exiting the power down mode. The hardware reset initiate the Special Function Register (see *Table 1*).
In the Power Down mode, V_{CC} may be lowered to minimize circuit power consumption. Care must be taken to ensure the voltage is not reduced until the power down mode is entered, and that the voltage is restored before the hardware reset is applied which frees the oscillator. Reset should not be released until the oscillator has restarted and stabilized.

Table 1 describes the status of the external pins while in the power down mode. It should be noted that if the power down mode is activated while in external program memory, the port data that is held in the Special Function Register P2 is restored to Port 2. If the data is a 1, the port pin is held high during the power down mode by the strong pullup, T1, shown in *Figure 4.*

Due to static design, the MHS 80C31S/C51S clock speed can be reduced until 0 MHz without any data loss in memory or registers. This mode allows step by step utilization, and permits to reduce system power consumption by bringing the clock frequency down to any value. At 0 MHz, the power consumption is the same as in the Power Down Mode.

80C51 I/O PORTS

The I/O port drive of the 80C51S is similar to 8051. The I/O buffers for Ports 1, 2, and 3 are implemented as shown in *figure 4.*
When the port latch contains a 0, all pFETS in *figure 4* are off while the nFET is turned on. When the port latch makes a 0-to-1 transition, the nFET turns off. The strong pullup pFET, T1, turns on for two oscillator periods, pulling the output high very rapidly. As the output line is drawn high, pFET T3 turns on through the inverter to supply the I_{OH} source current. This inverter and T3 form a latch which holds the 1 and is supported by T2.

When Port 2 is used as an address port, for access to external program of data memory, any address bit that contains a1 will have his strong pullup turned on for the entire duration of the external memory access.

When an I/O pin on Ports 1, 2, or 3 is used as an input, the user should be aware that the external circuit must sink current during the logical 1-to-0 transition. The maximum sink current is specified as ITL under the D.C. Specifications. When the input goes below approximately 2 V, T3 turns off to save IC current. Note, when returning to a logical 1, T2 is the only internal pullup that is on. This will result in a slow rise time if the user's circuit does not force the input line high.

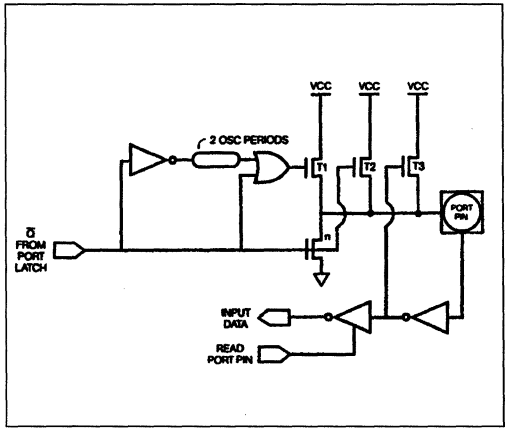


Figure 4 : I/O Buffers in the 80C51S (Ports 1, 2, 3).

80C31S/80C51S PIN DESCRIPTIONS

V_{SS}

Circuit ground potential

V_{CC}

Supply voltage during normal, Idle, and Power Down operation.

Port 0

Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1's. Port 0 also outputs the code bytes during program verification in the 80C51S. External pullups are required during program verification. Port 0 can sink eight LS TTL inputs.

Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. Port 1 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address bytes during program verification. In the 80C51S, Port 1 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. Port 2 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups. Port 2 emits the high order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pullups when emitting 1's. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

It also receives the high-order address bits and control signals during program verification in the 80C51S. Port 2 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. Port 3 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the pullups. It also serves the func-

tions of various special features of the MHS-51 Family, as listed below.

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 Timer 1 external input)
P3.6	$\overline{WR}$ (external Data Memory write strobe)
P3.7	$\overline{RD}$ (external Data Memory read strobe)

Port 3 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

RST

A high level on this for two machine cycles while the oscillator is running resets the device. An internal pulldown resistor permits Power-On reset using only a capacitor connected to VCC.

ALE

Address Latch Enable output for latching the low byte of the address during accesses to external memory. ALE is activated as though for this purpose at a constant rate of 1/6 the oscillator frequency except during an external data memory access at which time one ALE pulse is skipped. ALE can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

$\overline{PSEN}$

Program Store Enable output is the read strobe to external Program Memory. PSEN is activated twice each machine cycle during fetches from external Program Memory (However, when execution out of external Program Memory, two activations of PSEN are skipped during each access to external Data Memory). PSEN is not activated during fetches from internal Program Memory. PSEN can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

$\overline{EA}$

When EA is held high, the CPU executes out of internal Program Memory (unless the Program Counter exceeds 0FFFH). When EA is held low, the CPU executes only out of external Program Memory. EA must not be floated.

XTAL1

Input to the inverting amplifier that forms the oscillator. Receives the external oscillator signal when an external oscillator is used.

XTAL2

Output of the inverting amplifier that forms the oscillator, and input to the internal clock generator. This pin should be floated when an external oscillator is used.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output respectively, of an inverting amplifier which is configured for use as an on-chip oscillator, as shown in *figure 5.* Either a quartz crystal or ceramic resonator may be used.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected as shown in *figure 6.* There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

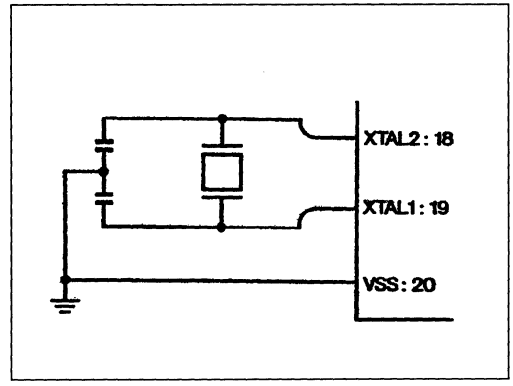


Figure 5 : Crystal Oscillator.

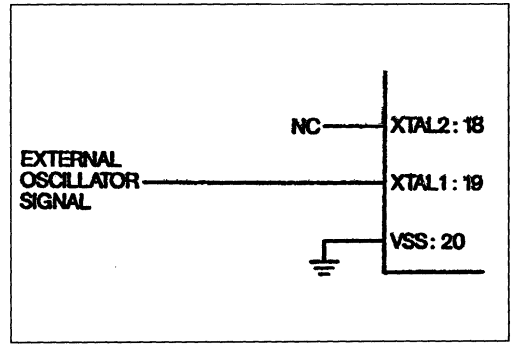


Figure 6 : External Drive Configuration.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias :
- commercial ..... 0°C to 70°C

Storage Temperature ..... – 65°C to + 150°C
Voltage on V_{CC} to V_{SS} ..... – 0,5 V to + 7 V
Voltage on Any Pin to V_{SS} ..... – 0.5 V to V_{CC} + 0.5 V
Power Dissipation ..... 1 W**

** This value is based on the maximum allwable die temperature and the thermal resistance of the package.

*** NOTICE :**

Stresses at or above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

DC CHARACTERISTICS (see Note 2)
T_A = 0°C to 70°C ; VCC = 5 V ± 10 % ; VSS = 0 V ; F = 0 to 20 MHz

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITIONS
VIL	Input Low Voltage	– 0.5	0.2 VCC – 0.1	V	
VIH	Input High Voltage (Except XTAL and RST)	0.2 VCC + 0.9	VCC + 0.5	V	
VIH1	Input High Voltage (RST and XTAL1)	0.7 VCC	VCC + 0.5	V	
VOL	Output Low Voltage (Ports 1, 2, 3)		0.45	V	IOL = 1.6 mA (note3)
VOL1	Output Low Voltage Port 0, ALE, $\overline{PSEN}$		0.45	V	IOL = 3.2 mA (note 3)
VOH	Output High Voltage Ports 1, 2, 3	0.9 VCC		V	IOH = –10 µA
		0.75 VCC		V	IOH = – 25 µA
		2.4		V	IOH = – 60 µA VCC = 5 V ± 10 %
VOH1	Output High Voltage (Port 0, ALE, $\overline{PSEN}$)		0.9 VCC	V	IOH = – 80 µA
			0.75 VCC	V	IOH = – 350 µA
			2.4	V	IOH = – 800 µA VCC = 5 V ± 10 %
IIL	Logical 0 Input Current Ports 1, 2, 3		– 50	µA	Vin = 0.45 V
ILI	Logical Leakage Current (Port 0, EA)		± 10	µA	0.45 < Vin < VCC
ITL	Logical 1 to 0 Transition Current (Ports 1, 2, 3)		– 650	µA	Vin = 2.0 V
IPD	Power Supply Current (Power Down Mode)		50	µA	VCC = 2.0 V to 5.5 V (note 2)
RRST	RST Pulldown Resistor	50	150	kΩ	
CIO	Capacitance of I/O Buffer		10	pF	f_C = 1 MHz, T_A = 25°C
ICC	Power supply current Active mode 20 MHz Idle mode 20 MHz	 32 8	 mA mA		(notes 1, 2)

Note 1 : ICC max is given by :
Active Mode : ICCMAX = 1.47 x FREQ + 2.35
Idle Mode : ICCMAX = 0.33 x FREQ + 1.05
where FREQ is the external oscillator frequency in MHz. ICCMAX is given in mA. See *Figure 1.*
See *figure 1* through *5* for ICC test conditions.

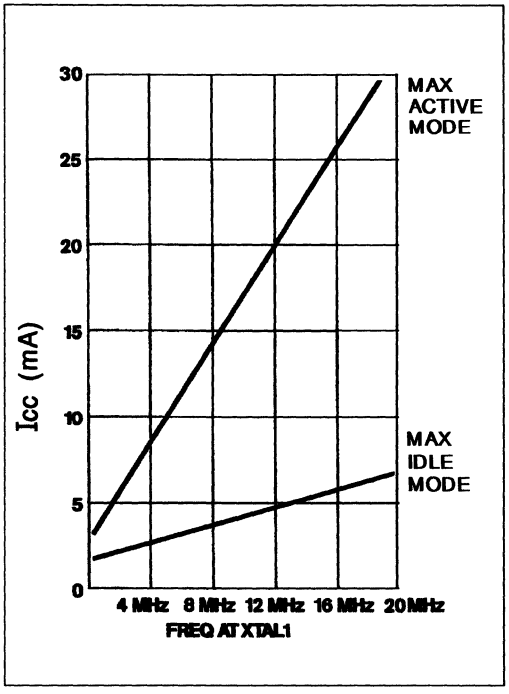


Figure 1 : ICC vs. Frequency. Valid only within frequency specifications of the device under test.

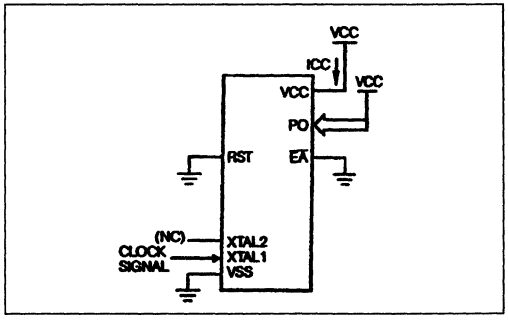


Figure 2 : ICC Test Condition, Idle Mode. All other pins are disconnected.

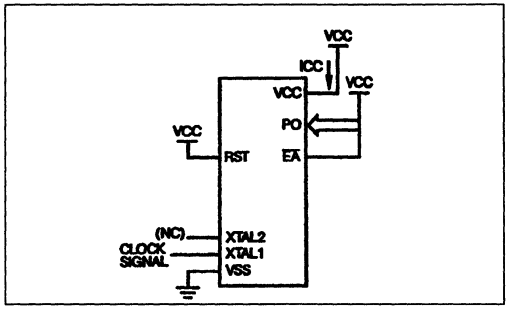


Figure 3 : ICC Test Condition, Active Mode. All other pins are disconnected.

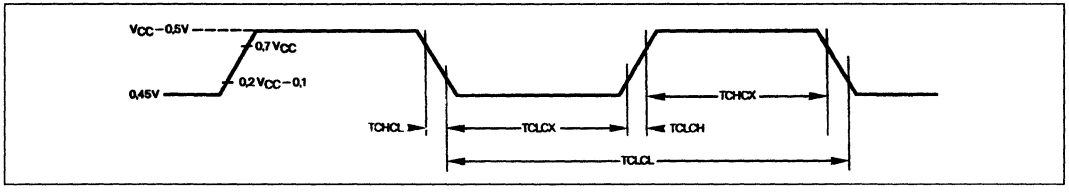


Figure 4 : Clock Signal Waveform for ICC Tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns.

Note 2 : ICC is measured with all output pins disconnected ; XTAL1 driven with TCLCH, TCHCL = 5 ns, VIL = VSS + .5 V, VIH = VCC – .5 V ; XTAL2 N.C. ; EA = RST = Port 0 = VCC ICC would be slightly higher if a crystal oscillator used. Idle ICC is measured with all output pins disconnected ; XTAL1 driven with TCLCH, TCHCL = 5 ns, VIL = VSS + .5 V, VIH = VCC –.5 V ; XTAL2 N.C. ; Port 0 = VCC ; EA = RST = VSS.
Power Down ICC is measured with all output pins disconnected ; EA = PORT 0 = VCC ; XTAL2 N.C. ; RST = VSS.

Note 3 : Capacitance loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the VOLS of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1 to 0 transitions during bus operations. In the worst cases (capacitive loading 100 pF), the noise pulse on the ALE line may exceed 0,45 V with maxi VOL peak 0.6 V. A Schmitt Trigger use is not necessary.

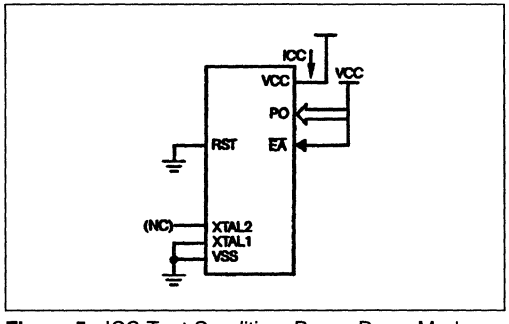


Figure 5 : ICC Test Condition, Power Down Mode. All other pins are disconnected.

6

EXTERNAL CLOCK DRIVE CHARACTERISTICS (XTAL 1)

SYMBOL	PARAMETER	VARIABLE CLOCK FREQ = 0 to 20 MHz		UNIT
		MIN	MAX	
1/TCLCL	Oscillator Frequency	50		ns
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

A.C. PARAMETERS :

TA = 0°C + 70°C ; VSS = 0 V ; VCC = 5 V ± 10 % (commercial)
(Load Capacitance for Port 0, ALE, and PSEN = 100 pf ; Load Capacitance for All Other Outputs = 80 pf).

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

SYMBOL	PARAMETER	MIN	MAX	UNIT
TLHLL	ALE Pulse Width	2TCLCL-40		ns
TAVLL	Address Valid to ALE	TCLCL-35		ns
TLLAX	Address Hold After ALE	TCLCL-30		ns
TLLIV	ALE to Valid Instr in		4TCLCL-100	ns
TLLPL	ALE to $\overline{PSEN}$	TCLCL-40		ns
TPLPH	$\overline{PSEN}$ Pulse Width	3TCLCL-45		ns
TPLIV	$\overline{PSEN}$ to Valid Instr in		3TCLCL-50	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		TCLCL-20	ns
TPXAV	$\overline{PSEN}$ to Address Valid	TCLCL-8		ns
TAVIV	Address to Valid Instr in		5TCLCL-105	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		10	ns

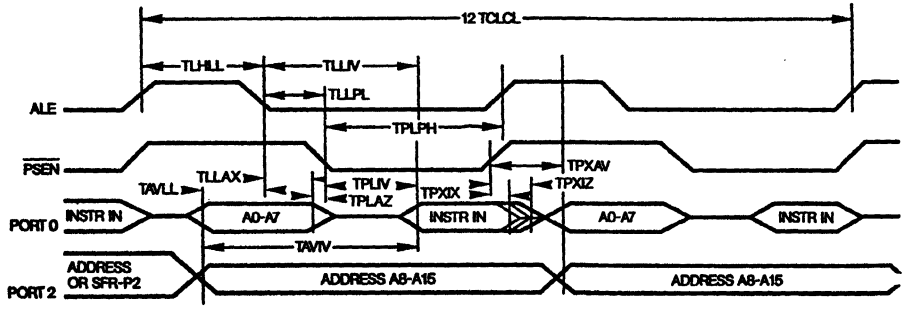
6

EXTERAL DATA MEMORY CHARACTERISTICS

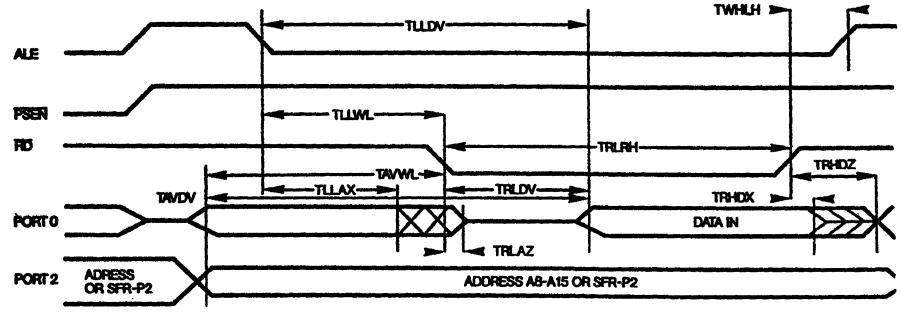
SYMBOL	PARAMETER	MIN	MAX	UNIT
TRLRH	$\overline{RD}$ Pulse Width	6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	6TCLCL-100		ns
TLLAX	Data Address Hold After ALE	TCLCL-30		ns
TRLDV	$\overline{RD}$ to Valid Data in		5TCLCL-100	ns
TRHDX	Data Hold After $\overline{RD}$	0		ns
TRHDZ	Data Float After $\overline{RD}$		2TCLCL-50	ns
TLLDV	ALE to Valid Data in		8TCLCL-150	ns
TAVDV	Address to Valid Data in		9TCLCL-100	ns
TLLWL	ALE to $\overline{WR}$ or $\overline{RD}$	3TCLCL-50	3TCLCL+50	ns
TAVWL	Address to $\overline{WR}$ or $\overline{RD}$	4TCLCL-130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	TCLCL-60		ns
TQVWH	Data Setup to $\overline{WR}$ High	7TCLCL-150		ns
TWHQX	Data Hold After $\overline{WR}$	TCLCL-20		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	TCLCL-40	TCLCL+40	ns

AC TIMING DIAGRAMS

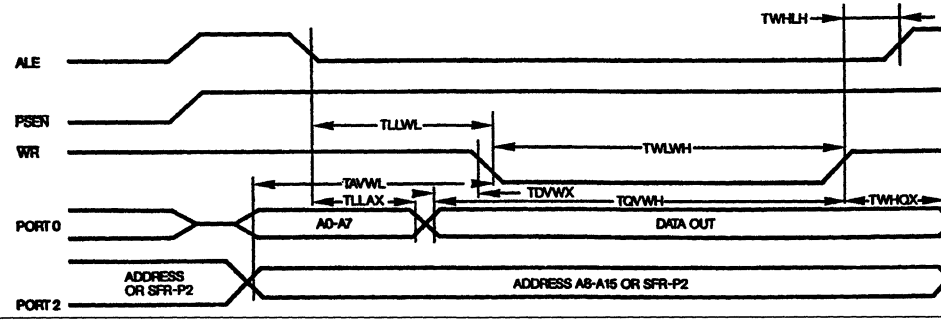
EXTERNAL PROGRAM MEMORY CYCLE



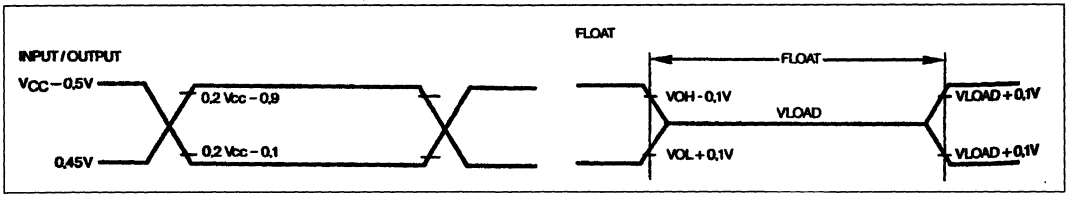
EXTERNAL DATA MEMORY READ CYCLE



EXTERNAL DATA MEMORY WRITE CYCLE



AC TESTING INPUT/OUTPUT, FLOAT WAVEFORMS

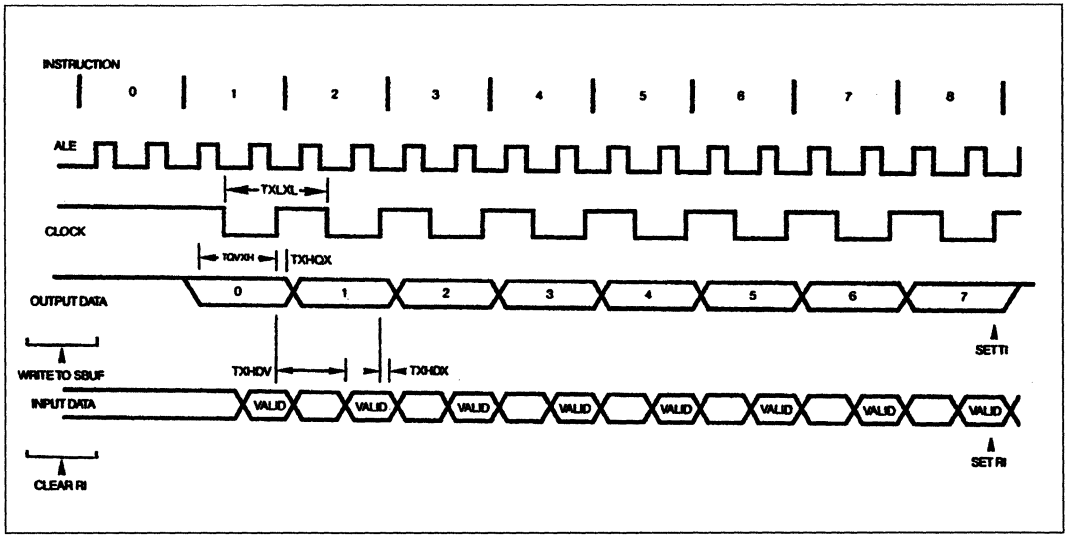


AC inputs during testing are driven at V_{CC} - 0.5 for a logic "1" and 0.45 V for a logic "0". Timing measurements are made at VIH min for a logic "1" and VIL max for a logic "0". For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change from the loaded VOH/VOL level occurs. IOL/IOH ≥ ± 20 mA.

SERIAL PORT TIMING - SHIFT REGISTER MODE

SYMBOL	PARAMETER	MIN	MAX	UNIT
TXLXL	Serial Port Clock Cycle Time	12TCLCL		µs
TQVXH	Output Data Setup to Clock Rising Edge	10TCLCL-133		ns
TXHQX	Output Data Hold After Clock Rising Edge	2TCLCL-117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		ns
TXHDV	Clock Rising Edge to Input Data Valid		10TLCL-133	ns

SHIFT REGISTER TIMING WAVEFORMS



6

EXPLANATION OF THE AC SYMBOLS

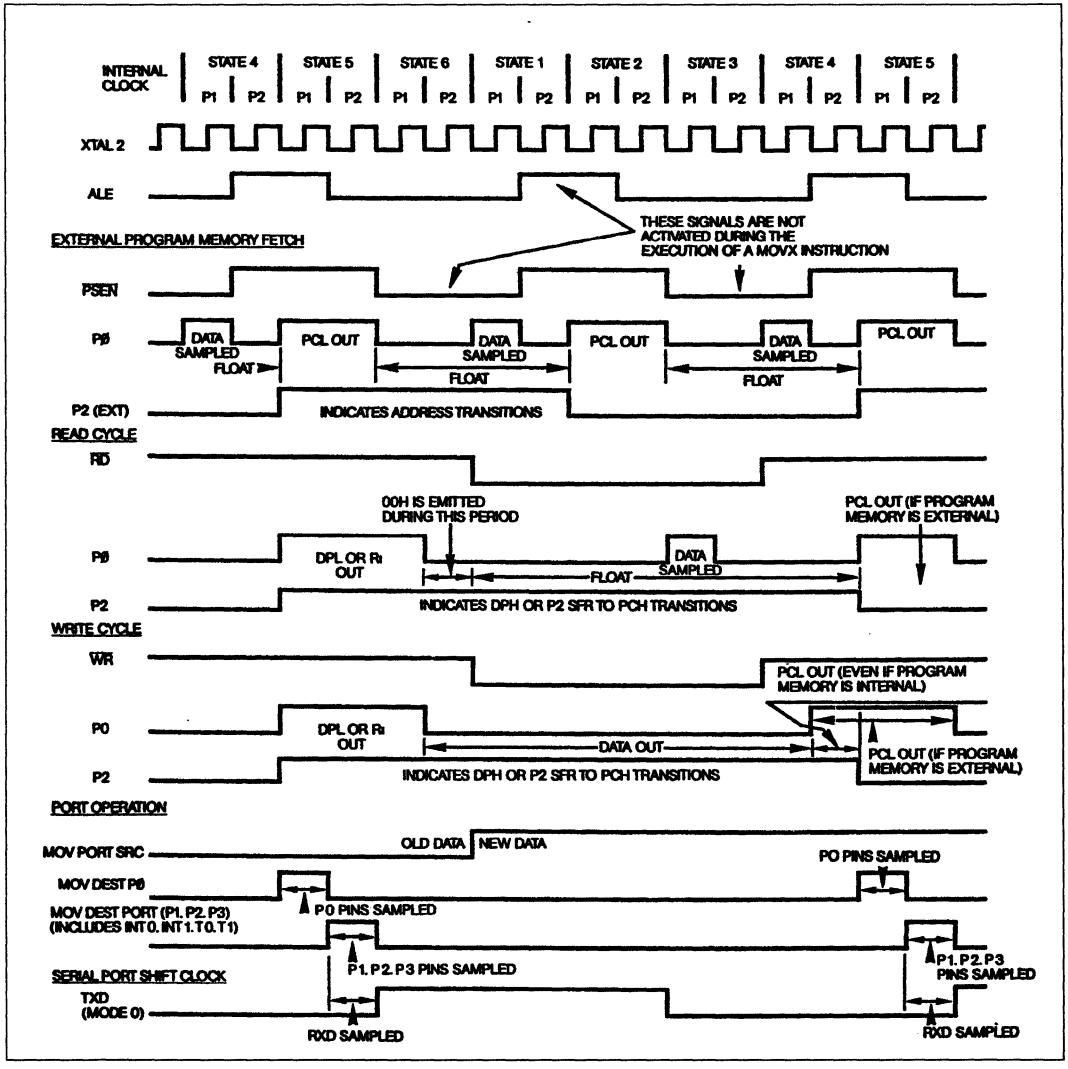
Each timing symbol has 5 characters. The first character is always a "T" (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

Example :

TAVLL = Time for Address Valid to ALE low.
TLLPL = Time for ALE low to PSEN low.

A : Address.	Q : Output data.
C : Clock.	R : READ signal.
D : Input data.	T : Time.
H : Logic level HIGH.	V : Valid.
I : Instruction (program memory contents).	W : WRITE signal.
L : Logic level LOW, or ALE.	X : No longer a valid logic level.
P : PSEN.	Z : Float.

CLOCK WAVEFORMS



This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins, however, ranges from 25 to 125 ns. This propagation delay is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component. Typically though (T_A = 25°C fully loaded) RD and WR propagation delays are approximately 50 ns. The other signals are typically 85 ns. Propagation delays are incorparated in the AC specifications.

ARITHMETIC OPERATIONS				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with Carry	1	1
ADDC	A,direct	Add direct byte to A with Carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with Carry flag	1	1
ADDC	A,#data	Add immediate data to A with Carry flag	2	1
SUBB	A,Rn	Subtract register from A with Borrow	1	1
SUBB	A,direct	Subtract direct byte from A with Borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A with Borrow	1	1
SUBB	A,#data	Subtract immed. data from A with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1
LOGICAL OPERATIONS				
MNEMONIC		**DESTINATION**	**BYTE**	**CYC**
ANL	A,Rn	AND register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate A Left through the Carry flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate A Right through Carry flag	1	1
SWAP	A	Swap nibbles within the Accumulator	1	1

Table 1 : MHS - 51 Instruction Set Description.

DATA TRANSFER				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
MOV	A,Rn	Move register to Accumulator	1	1
MOV	A,direct	Move direct byte to Accumulator	2	1
MOV	A,@Ri	Move indirect RAM to Accumulator	1	1
MOV	A,#data	Move immediate data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move Accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data 16	Load Data Pointer with a 16-bit constant	3	2
MOVC	A,@A + DPTR	Move Code byte relative to DPTR to A	1	2
MOVC	A,@A + PC	Move Code byte relative to PC to A	1	2
MOVX	A,@Ri	Move External RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move External RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to External RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to External RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with Accumulator	1	1
XCH	A,direct	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order nibble ind RAM with A	1	1
BOOLEAN VARIABLE MANIPULATION				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
CLR	C	Clear Carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry flag	1	1
SETB	bit	Set direct Bit	2	1
CPL	C	Complement Carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry flag	2	2
ANL	C, /bit	AND complement of direct bit to Carry	2	2
ORL	C, bit	OR direct bit to Carry flag	2	2
ORL	C, /bit	OR complement of direct bit to Carry	2	2
MOV	C, bit	Move direct bit to Carry flag	2	1
MOV	bit,C	Move Carry flag to direct bit	2	2
PROGRAM AND MACHINE CONTROL				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
ACALL	addr11	Absolute Subroutine Call	2	2
LCALL	addr16	Long Subroutine Call	3	2
RET		Return from subroutine	1	
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute Jump	2	2
LJMP	addr16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A + DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is Zero	2	2
JNZ	rel	Jump if Accumulator is Not Zero	2	2
JC	rel	Jump if Carry flag is set	2	2
JNC	rel	Jump if No Carry flag	2	2

Table 1. (Cont.)

PROGRAM AND MACHINE CONTROL (cont.)				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
JB	bit,rel	Jump if direct Bit set	3	2
JNB	bit,rel	Jump if direct Bit Not set	3	2
JBC	bit,rel	Jump if direct Bit is set & Clear bit	3	2
CJNE	A,direct, rel	Compare direct to A & Jump if Not Equal	3	2
CJNE	A,#data, rel	Comp. immed. to A & Jump if Not Equal	3	2
CJNE	Rn,#data, rel	Comp. immed. to reg & Jump if Not Equal	3	2
CJNE	@Ri,#data. rel	Comp. immed. to ind. & Jump if Not Equal	3	2
DJNZ	Rn,rel	Decrement register & Jump if Not Zero	2	2
DJNZ	direct. rel	Decrement direct & Jump if Not Zero	3	2
NOP		No operation	1	1

Table 1. (Cont.)

Notes on data addressing modes :

Rn – Working register R0-R7
direct – 128 internal RAM locations, any I/O port, control or status register
@Ri – Indirect internal RAM location addressed by register R0 or R1
#data – 8-bit constant included in instruction
#data 16 – 16-bit constant included as bytes 2 & 3 of instruction
bit – 128 software flags, any I/O pin, control or status bit

Notes on program addressing modes :

addr 16 – Destination address for LCALL & LJMP may be anywhere within the 64-k program memory address space
Addr 11 – Destination address for ACALL & AJMP will be within the same 2-k page of program memory as the first byte of the following instruction
rel – SJMP and all conditional jumps include an 8-bit offset byte. Range is + 127 – 128 bytes relative to the first byte of the following instruction.

6

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr,code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr,code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A,data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6
2F	1	ADD	A,R7
30	3	JNB	bit addr,code addr
31	2	ACALL	code addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A,#data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr,A
43	3	ORL	data addr,#data
44	2	ORL	A,#data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr,A
53	3	ANL	data addr,#data
54	2	ANL	A,#data
55	2	ANL	A,data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	code addr
61	2	AJMP	code addr
62	2	XRL	data addr A
63	3	XRL	data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr

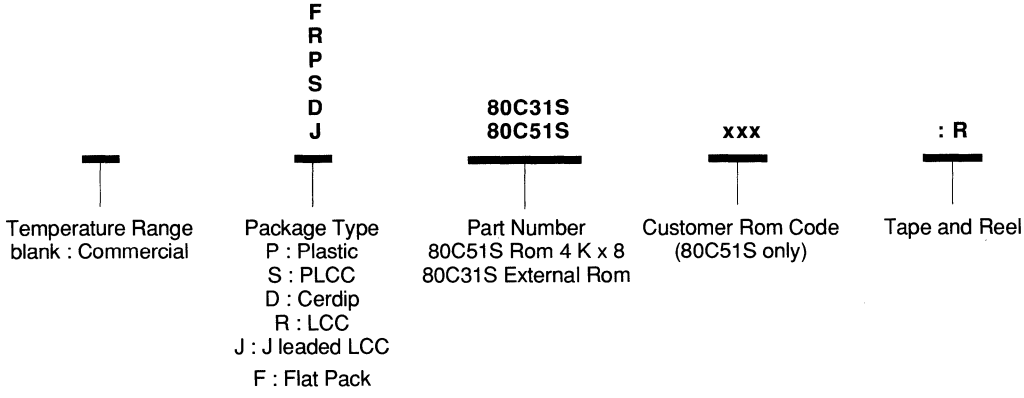
Table 2 : Instruction Opcodes in Hexadecimal Order.

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C,bit addr
73	1	JMP	@A + DPTR
74	2	MOV	A,#data
75	3	MOV	data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C,bit addr
83	1	MOVC	A,@A + PC
84	1	DIV	AB
85	3	MOV	data addr,data addr
86	2	MOV	data addr,@R0
87	2	MOV	data addr,@R1
88	2	MOV	data addr,R0
89	2	MOV	data addr,R1
8A	2	MOV	data addr,R2
8B	2	MOV	data addr,R3
8C	2	MOV	data addr,R4
8D	2	MOV	data addr,R5
8E	2	MOV	data addr,R6
8F	2	MOV	data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	code addr
92	2	MOV	bit addr,C
93	1	MOVC	A,@A + DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,bit addr
A1	2	AJMP	code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,bit addr
B1	2	ACALL	code addr
B2	2	CPL	Bit addr
B3	1	CPL	C
B4	3	CJNE	A, #data, code addr
B5	3	CJNE	A, data addr, code addr
B6	3	CJNE	@R0,#data, code addr
B7	3	CJNE	@R1,#data, code addr
B8	3	CJNE	R0,#data, code addr
B9	3	CJNE	R1,#data, code addr
BA	3	CJNE	R2,#data, code addr
BB	3	CJNE	R3,#data, code addr
BC	3	CJNE	R4,#data, code addr
BD	3	CJNE	R5,#data, code addr
BE	3	CJNE	R6,#data, code addr
BF	3	CJNE	R7,#data, code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3

Table 2. (Cont.)

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

Table 2. (Cont.)



DATA SHEET

80C51-L/80C31-L

CMOS SINGLE-CHIP 8 BIT MICROCONTROLLER-LOW POWER

- **80C51-L-CMOS SINGLE-SHIP 8-BIT MICROCONTROLLER with factory mask-programmable ROM**
- **80C31-L-CMOS SINGLE-CHIP 8-BIT CONTROL-ORIENTED CPU with RAM and I/O**
- **80C51-L/C31-L : 0 TO 6 MHz, VCC = 2.7 V TO 6 V**

FEATURES

- **POWER CONTROL MODES**
- **128 x 8 BIT RAM**
- **32 PROGRAMMABLE I/O LINES**
- **TWO 16-BIT TIMER/COUNTERS**
- **64 K PROGRAM MEMORY SPACE**
- **FULLY STATIC DESIGN**
- **HIGH PERFORMANCE SAJI VI CMOS PROCESS**
- **BOOLEAN PROCESSOR**
- **5 INTERRUPT SOURCES**
- **PROGRAMMABLE SERIAL PORT**
- **64 K DATA MEMORY SPACE**
- **TEMPERATURE RANGE : Commercial, Industrial**

DESCRIPTION

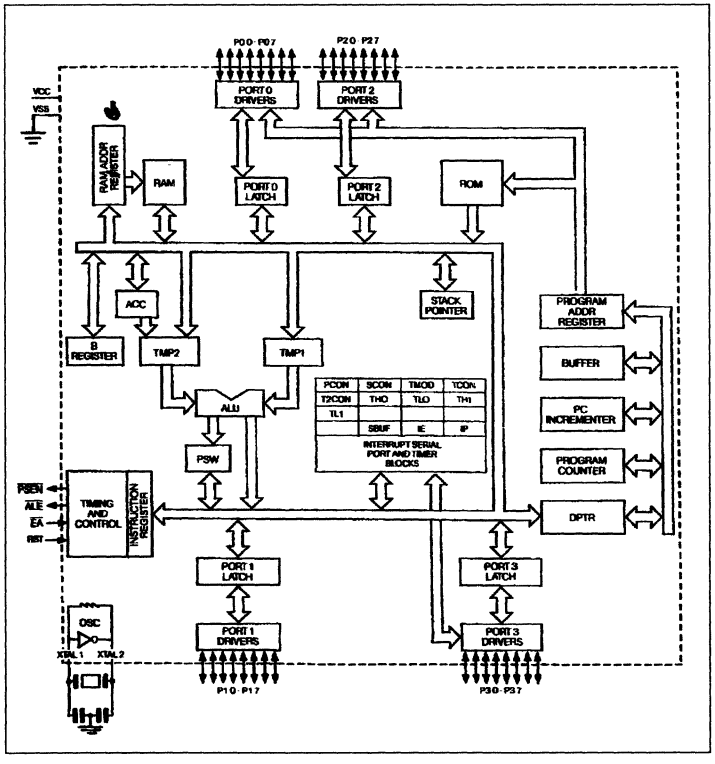


Figure 1 : Block Diagram.

MHS's 80C51 and 80C31 are high performance CMOS versions of the 8051/8031 NMOS single chip 8 bit µC and is manufactured using a self-aligned silicon gate CMOS process (SAJI VI).

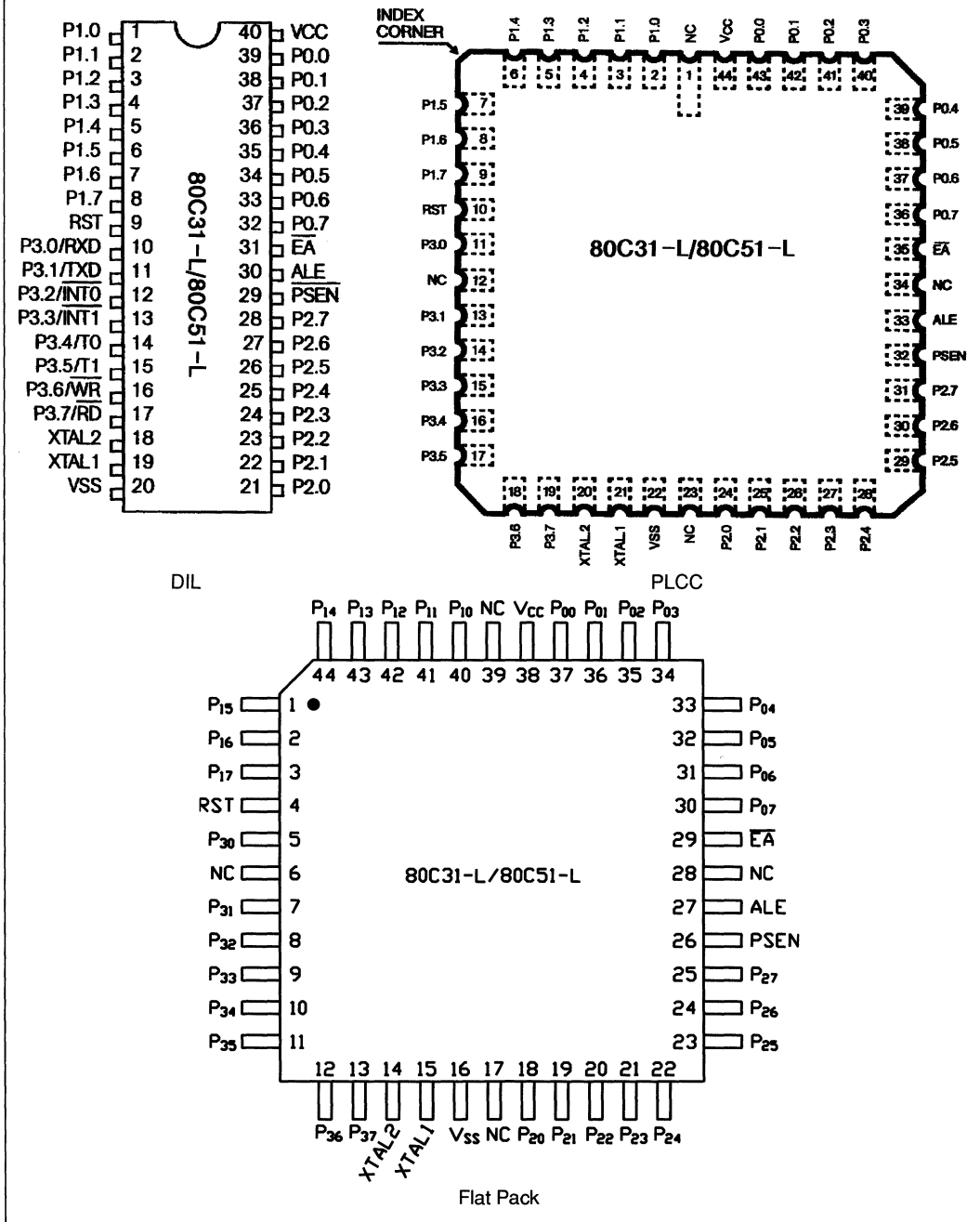
The fully static design of the MHS 80C51/80C31 allows to reduce system power consumption by bringing the clock frequency down to any value, even DC, without loss of data.

The 80C51 retains all the features of the 8051 : 4 K bytes of ROM ; 128 bytes of RAM ; 32 I/O lines ; two 16 bit timers ; a 5-source 2-level interrupt structure ; a full duplex serial port ; and on-chip oscillator and clock circuits.

In addition, the 80C51 has two software-selectable modes of reduced activity for further reduction in power consumption. In the idle Mode the CPU is frozen while the RAM, the timers, the serial port, and the interrupt system continue to function. In the Power Down Mode the RAM is saved and all other functions are inoperative.

The 80C31 is identical to the 80C51 except that it has no on-chip ROM.

6



Diagrams are for pin reference only. Package sizes are not to scale.

Figure 2 : Configurations.

IDLE AND POWER DOWN OPERATION

Figure 3 shows the internal Idle and Power Down clock configuration. As illustrated, Power Down operation stops the oscillator. Idle mode operation allows the interrupt, serial port, and timer blocks to continue to function while the clock to the CPU is gated off.
These special modes are activated by software via the Special Function Register, its hardware address is 87H. PCON is not bit addressable.

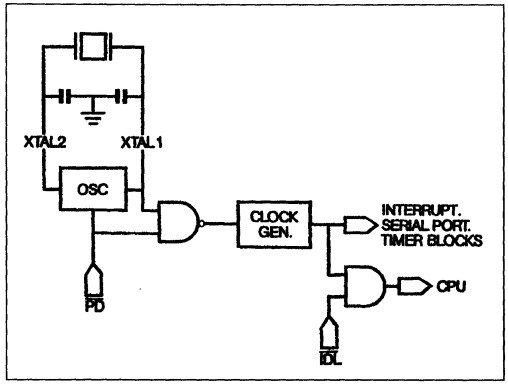


Figure 3 : Idle and Power Down Hardware.

PCON : Power Control Register

(MSB) (LSB)

SMOD	–	–	–	GF1	GF0	PD	IDL

Symbol	Position	Name and Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1, 2 or 3.
–	PCON.6	(Reserved)
–	PCON.5	(Reserved)
–	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activates idle mode operation.

If 1's are written to PD and IDL at the same time. PD takes precedence. The reset value of PCON is (0XXX0000).

MODE	PROGRAM MEMORY	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Port Data	Port Data	Port Data	Port Data
Idle	External	1	1	Floating	Port Data	Address	Port Data
Power Down	Internal	0	0	Port Data	Port Data	Port Data	Port Data
Power Down	External	0	0	Floating	Port Data	Port Data	Port Data

Table 1 : Status of the external pins during Idle and Power Down modes.

6

IDLE MODE

The instruction that sets PCON.0 is the last instruction executed before the Idle mode is activated. Once in the Idle mode the CPU status is preserved in its entirety : the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM, and all other registers maintain their data during Idle. *Table 1* describes the status of the external pins during Idle mode.

There are two ways to terminate the Idle mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating Idle mode. The interrupt is serviced, and following RETI, the next instruction to be executed will be the one following the instruction that wrote a 1 to PCON.0.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during the Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an enabled interrupt, the service routine can examine the status of the flag bits.

The second way of terminating the Idle mode is with a hardware reset. Since the oscillator is still running, the hardware reset needs to be active for only 2 machine cycles (24 oscillator periods) to complete the reset operation.

POWER DOWN MODE

The instruction that sets PCON.1 is the last executed prior to entering power down. Once in power down, the oscillator is stopped. The contents of the on chip RAM and the Special Function Register is saved during power down mode. A hardware reset is the only way of exiting the power down mode. The hardware reset initiates the Special Function Register *(see Table 1)*.

In the Power Down mode, V_{CC} may be lowered to minimize circuit power consumption. Care must be taken to ensure the voltage is not reduced until the power

down mode is entered, and that the voltage is restored before the hardware reset is applied which frees the oscillator. Reset should not be released until the oscillator has restarted and stabilized.

Table 1 describes the status of the external pins while in the power down mode. It should be noted that if the power down mode is activated while in external program memory, the port data that is held in the Special Function Register P2 is restored to Port 2. If the data is a 1, the port pin is held high during the power down mode by the strong pullup, T1, shown in *figure 4.*

STOP CLOCK MODE

Due to static design, the MHS 80C31/C51 clock speed can be reduced until 0 MHz without any data loss in memory or registers. This mode allows step by step utilization, and permits to reduce system power consumption by bringing the clock frequency down to any value. At 0 MHz, the power consumption is the same as in the Power Down Mode.

80C51 I/O PORTS

The I/O port drive of the 80C51 is similar to the 8051. The I/O buffers for Ports 1, 2 and 3 are implemented as shown in *figure 4.*

Where the port latch contains a 0, all pFETS in *figure 4* are off while the nFET is turned on. When the port latch makes a 0-to-1 transition, the nFET turns off. The strong pullup pFET, T1, turns on for two oscillator periods, pulling the output high very rapidly. As the output line is drawn high, pFET T3 turns on through the inverter to supply the I_{OH} source current. This inverter and T3 form a latch which holds the 1 and is supported by T2. When Port 2 is used as an address port, for access to external program of data memory, any address bit that contains a 1 will have his strong pullup turned on for the entire duration of the external memory access.

When an I/O pin on Ports 1, 2, or 3 is used as an input, the user should be aware that the external circuit must sink current during the logical 1-to-0 transition. The maximum sink current is specified as ITL under the D.C. Specifications. When the input goes below approximately 2 V, T3 turns off to save ICC current. Note, when returning to a logical 1, T2 is the only internal pullup that is on. This will result in a slow rise time if the user's circuit does not force the input line high.

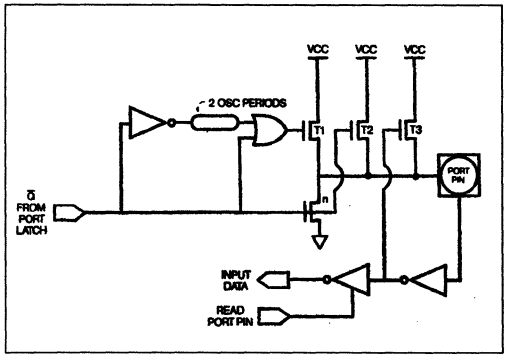


Figure 4 : I/O Buffers in the 80C51 (Ports 1, 2, 3).

80C31/80C51 PIN DESCRIPTIONS

V_{SS}

Circuit ground potential

V_{CC}

Supply voltage during normal, Idle, and Power Down operation.

Port 0

Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1's. Port 0 also outputs the code bytes during program verification in the 80C51. External pullups are required during program verification. Port 0 can sink eight LSTTL inputs.

Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. Port 1 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address bytes during program verification. In the 80C51, Port 1 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. Port 2 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups. Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @ DPTR). In this application, it uses strong internal pullups when emitting 1's. During accesses to external Data Memory that uses 8-bit addresses (MOVX @ Ri), Port 2 emits the contents of the P2 Special Function Register.

It also receives the high-order address bits and control signals during program verification in the 80C51. Port 2 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. Port 3 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the pullups. It also serves the functions of various special features of the MHS-51 Family, as listed below.

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{WR}$ (external Data Memory write strobe)
P3.7	$\overline{RD}$ (external Data Memory read strobe)

Port 3 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

RST

A high level on this for two machine cycles while the oscillator is running resets the device. An internal pulldown resistor permits Power-On reset using only a capacitor connected to V_{CC}.

ALE

Address Latch Enable output for latching the low byte of the address during accesses to external memory. ALE is activated as though for this purpose at a constant rate of 1/6 the oscillator frequency except during an external data memory access at which time one ALE pulse is skipped. ALE can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

$\overline{PSEN}$

Program Store Enable output is the read strobe to external Program Memory. PSEN is activated twice each machine cycle during fetches from external Program Memory. (However, when executing out of external Program Memory, two activations of PSEN are skipped during each access to external Data Memory). PSEN is not activated during fetches from internal Program Memory. PSEN can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

$\overline{EA}$

When EA is held high, the CPU executes out of internal Program Memory (unless the Program Counter exceeds 0FFFH). When EA is held low, the CPU executes only out of external Program Memory. EA must not be floated.

XTAL1

Input to the inverting amplifier that forms the oscillator. Receives the external oscillator signal when an external oscillator is used.

XTAL2

Output of the inverting amplifier that forms the oscillator, and input to the internal clock generator. This pin should be floated when an external oscillator is used.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output respectively, of an inverting amplifier which is configured for use as an on-chip oscillator, as shown in figure 5. Either a quartz crystal or ceramic resonator may be used.
To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected as shown in *figure 6*. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

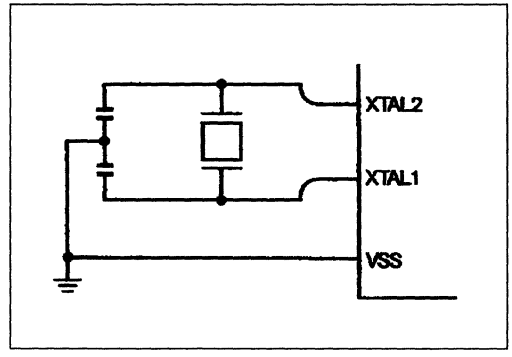


Figure 5 : Crystal Oscillator.

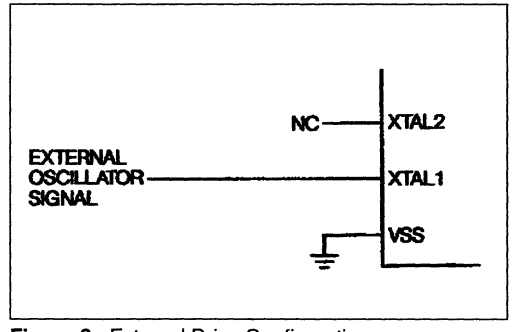


Figure 6 : External Drive Configuration.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias :
C = Commercial ..... 0°C to 70°C
I = Industrial ..... – 40°C to 85°C
Storage Temperature ..... – 65°C to + 150°C
Voltage on VCC to VSS ..... – 0.5 V to + 7 V
Voltage on Any Pin to VSS .... – 0.5 V to VCC + 0.5 V
Power Dissipation ..... 1 W**
** This value is based on the maximum allowable die temperature and the thermal resistance of the package.

*** NOTICE :**

Stresses at or above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

DC CHARACTERISTICS

T_A = – 40°C to 85°C ; V_{CC} = 2.7 V to 6 V ; VSS = 0 V ; F = 0 to 6 MHz

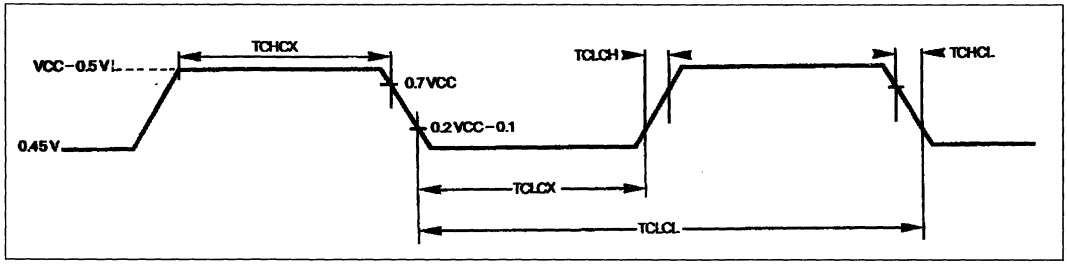
SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITIONS
VIL	Input Low Voltage	– 0.5	0.2 V_{CC} – 0.1	V	
VIH	Input High Voltage (Except XTALs and RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V	
VIH1	Input High Voltage to RST for Reset	0.7 V_{CC}	V_{CC} + 0.5	V	
VIH2	Input High Voltage to XTAL1	0.7 V_{CC}	V_{CC} + 0.5	V	
VPD	Power Down Voltage to V_{CC} in PD Mode	2.0	6.0	V	
VOL	Output Low Voltage (Ports 1, 2, 3)		0.45	V	IOL = 1.6 mA (note 1)
VOL1	Output Low Voltage Port 0, ALE, $\overline{PSEN}$		0.45	V	IOL = 3.2 mA (note 1)
VOH	Output High Voltage Ports 1, 2, 3	0.9 V_{CC}		V	IOH = –10 µA
		2.4		V	IOH = – 60 µA V_{CC} = 5 V ± 10 %
VOH1	Output High Voltage (Port 0 in External Bus Mode), ALE, $\overline{PSEN}$	0.9 V_{CC}		V	IOH = – 80µA
		2.4		V	IOH = – 800 µA V_{CC} = 5 V ± 10 %
IIL	Logical 0 Input Current Ports 1, 2, 3		C – 50	µA	Vin = 0.45 V
			I – 60		
ILI	Input Leakage Current		± 10	µA	0.45 < Vin < V_{CC}
ITL	Logical 1 to 0 Transition Current (Ports 1, 2, 3)		– 500	µA	Vin = 2.0 V
ICCPD	Power Supply Current (Power Down Mode)	10	50	µA	V_{CC} = 2.0 V to 5.5 V (note 2)
RRST	RST Pulldown Resistor	50	150	kΩ	
CIO	Capacitance of I/O Buffer		10	pF	f_C = 1 MHz, T_A = 25°C

Note 1 :

Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading 100 pF), the noise pulse on the ALE line may exceed 0.45 V with maxi V_{OL} peak 0.6 V. A Schmitt Trigger use is not necessary.

EXTERNAL CLOCK DRIVE CHARACTERISTICS (XTAL 1)

SYMBOL	PARAMETER	VARIABLE CLOCK FREQ = 0 to 6 MHz		UNIT
		MIN	MAX	
TCLCL	Oscillator Period	166		ns
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns



AC CHARACTERISTICS

(TA = – 40°C to 85°C, VCC = 2.7 V to 6 V, VSS = 0 V)
(Load Capacitance for Port 0, ALE, and PSEN = 100 pf ; Load Capacitance for All Other Outputs = 80 pf).

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

SYMBOL	PARAMETER	MIN	MAX	UNIT
TLHLL	ALE Pulse Width	2TCLCL-40		ns
TAVLL	Address Valid to ALE	TCLCL-55		ns
TLLAX	Address Hold After ALE	TCLCL-35		ns
TLLIV	ALE to Valid Instr In		4TCLCL-170	ns
TLLPL	ALE to $\overline{\text{PSEN}}$	TCLCL-25		ns
TPLPH	$\overline{\text{PSEN}}$ Pulse Width	3TCLCL-35		ns
TPLIV	$\overline{\text{PSEN}}$ to Valid Instr In		3TCLCL-220	ns
TPXIX	Input Instr Hold After $\overline{\text{PSEN}}$	0		ns
TPXIZ	Input Instr Float After $\overline{\text{PSEN}}$		TCLCL-20	ns
TPXAV	PSEN to Address Valid	TCLCL-8		ns
TAVIV	Address to Valid Instr In		5TCLCL-220	ns
TPLAZ	$\overline{\text{PSEN}}$ Low to Address Float		0	ns

See next page for External Data Memory Characteristics.

EXTERNAL DATA MEMORY CHARACTERISTICS

SYMBOL	PARAMETER	MIN	MAX	UNIT
TRLRH	$\overline{RD}$ Pulse Width	6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	6TCLCL-100		ns
TLLAX	Data Address Hold After ALE	TCLCL-50		ns
TRLDV	$\overline{RD}$ to Valid Data In		5TCLCL-165	ns
TRHDX	Data Hold After $\overline{RD}$	0		ns
TRHDZ	Data Float After $\overline{RD}$		2TCLCL-70	ns
TLLDV	ALE to Valid Data In		8TCLCL-150	ns
TAVDV	Address to Valid Data In		9TCLCL-165	ns
TLLWL	ALE to $\overline{WR}$ or $\overline{RD}$	3TCLCL-50	3TCLCL+50	ns
TAVWL	Address to $\overline{WR}$ or $\overline{RD}$	4TCLCL-130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	TCLCL-60		ns
TQVWH	Data Setup to $\overline{WR}$ High	7TCLCL-150		ns
TWHQX	Data Hold After $\overline{WR}$	TCLCL-50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	TCLCL-40	TCLCL+40	ns

MAXIMUM I_{CC} (mA)

	OPERATING (NOTE 3)			IDLE (NOTE 4)		
FREQ. VCC	2.7 V	5 V	6 V	2.7 V	5 V	6 V
1 MHz	0.8 mA	1.5 mA	1.8 mA	400 µA	800 µA	1 mA
6 MHz	4 mA	8 mA	10 mA	1.2 mA	3.5 mA	3.8 mA

Note 2 : Power Down I_{CC} is measured with all output pins disconnected ; EA = Port 0 = V_{CC} ; XTAL2 N.C. ; RST = V_{SS}

Note 3 : I_{CC} is measured with all output pins disconnected ; XTAL1 driven with TCLCH, TCHCL = 5 ns, $V_{IL} = V_{SS} + 0.5$ V ; $V_{IH} = V_{CC} - 0.5$ V ; XTAL2 N.C. ; EA = RST = Port 0 = V_{CC}. I_{CC} would be slightly higher if a crystal oscillator is used.

Note 4 : Idle I_{CC} is measured with all output pins disconnected ; XTAL1 driven TCLCH, TCHCL = 5 ns, $V_{IL} = V_{SS} + 0.5$ V ; $V_{IH} = V_{CC} - 0.5$ V ; XTAL2 N.C. ; Port 0 = V_{CC} ; EA = RST = V_{SS}.

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a ' T ' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list all the characters and what they stand for.

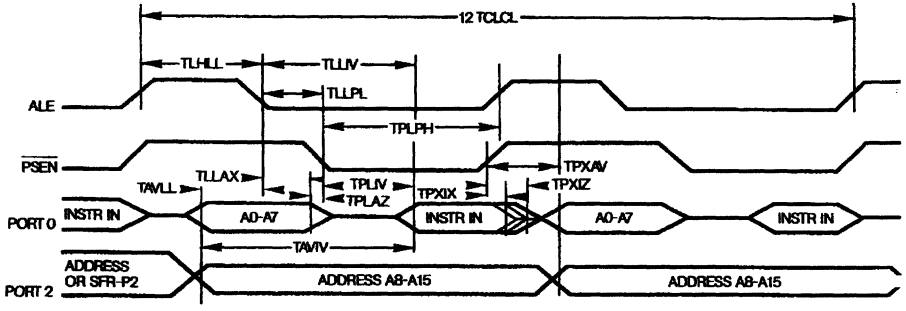
EXAMPLE :

TAVLL = Time for Address Valid to ALE low.
TLLPL = Time for ALE low to PSEN low.

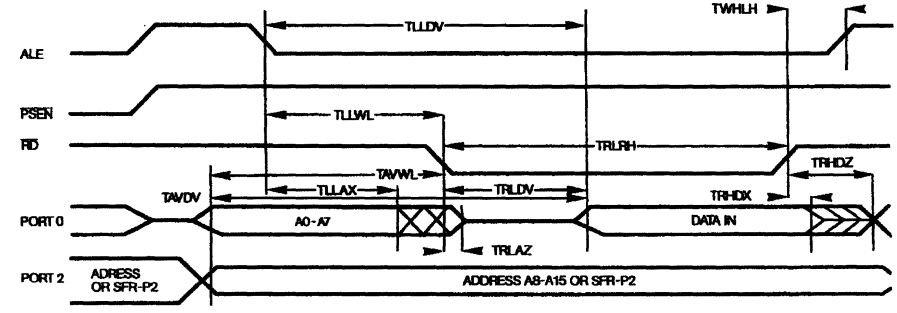
A : Address.
C : Clock.
D : Input data.
H : Logic level HIGH.
I : Instruction (program memory contents).
L : Logic level LOW, or ALE.
P : PSEN.
Q : Output data.
R : READ signal.
T : Time.
V : Valid.
W : WRITE signal
X : No longer a valid logic level.
Z : Float.

AC TIMING DIAGRAMS

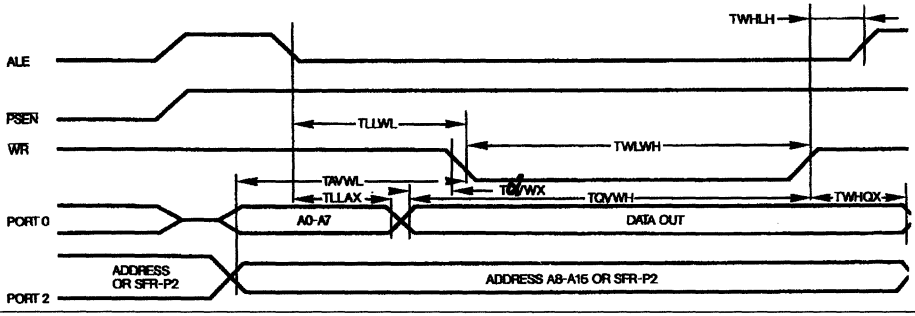
EXTERNAL PROGRAM MEMORY READ CYCLE



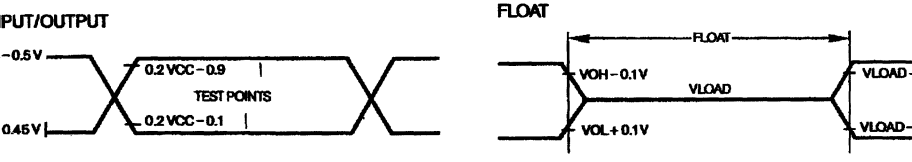
EXTERNAL DATA MEMORY READ CYCLE



EXTERNAL DATA MEMORY WRITE CYCLE



AC TESTING INPUT/OUTPUT, FLOAT WAVEFORMS



AC inputs during testing are driven at V_{CC} - 0.5 for a logic " 1 " and 0.45 V for a logic " 0 ". Timing measurements are made at VIH min for a logic " 1 " and VIL max for a logic " 0 ". For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change from the loaded VOH/VOL level occurs. IOL/IOH $\geq \pm$ 20 mA.

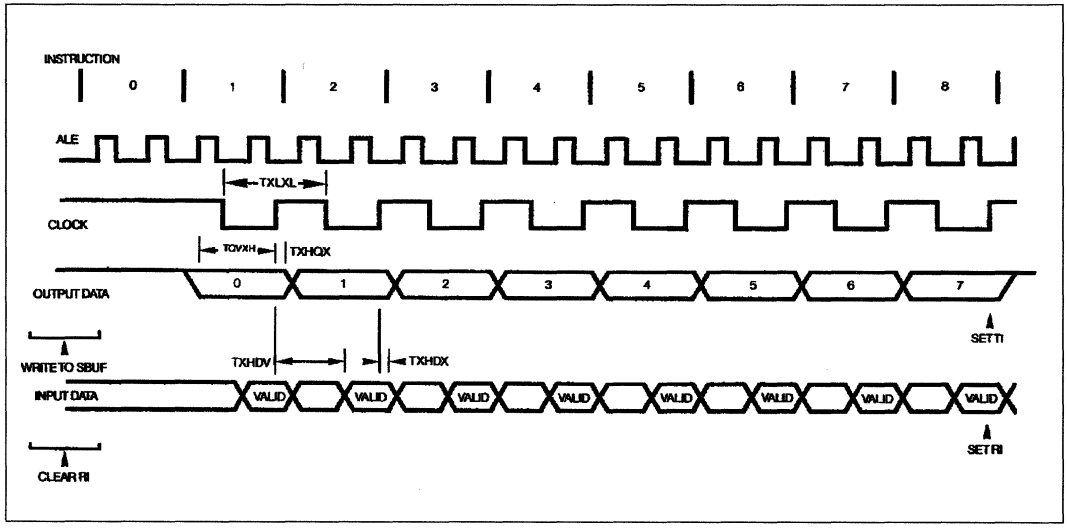
SERIAL PORT TIMING - SHIFT REGISTER MODE

A.C. CHARACTERISTICS :

TA = – 40°C to + 85°C ; VSS = 0 V ; VCC = 2.7 V to 6 V ; Load Capacitance = 80 pF)

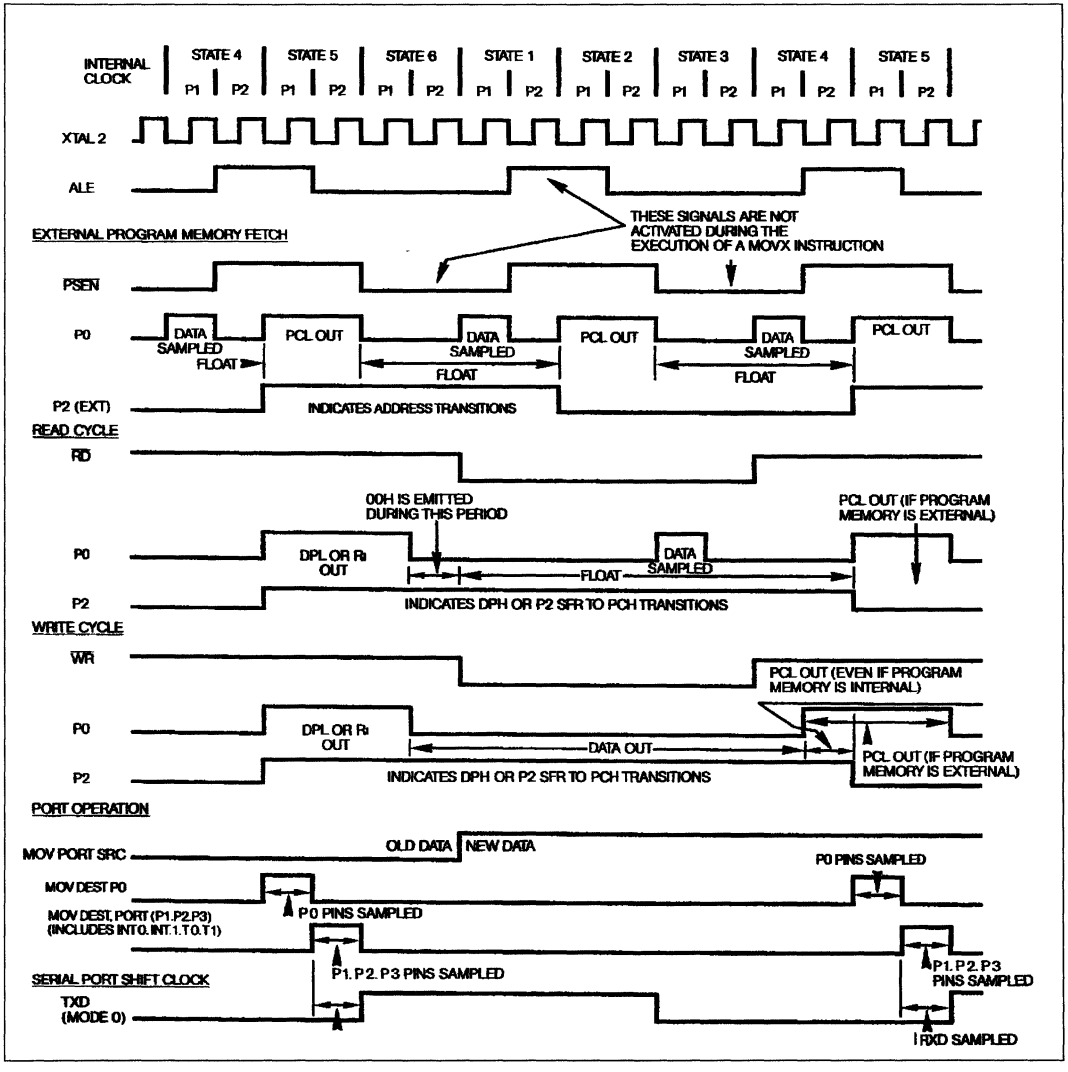
SYMBOL	PARAMETER	MIN	MAX	UNIT
TXLXL	Serial Port Clock Cycle Time	12TCLCL		ns
TQVXH	Output Data Setup to Clock Rising Edge	10TCLCL-133		ns
TXHQX	Output Data Hold After Clock Rising Edge	2TCLCL-117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		ns
TXHDV	Clock Rising Edge to Input Data Valid		10TLCL-133	ns

SHIFT REGISTER TIMING WAVEFORMS



6

CLOCK WAVEFORMS



This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins, however, ranges from 25 to 125 ns. This propagation delay is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component. Typically though (T_A = 25°C fully loaded) RD and WR propagation delays are approximately 50 ns. The other signals are typically 85 ns. Propagation delays are incorporated in the AC specifications.

ARITHMETIC OPERATIONS				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with Carry	1	1
ADDC	A,direct	Add direct byte to A with Carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with Carry flag	1	1
ADDC	A,#data	Add immediate data to A with Carry flag	2	1
SUBB	A,Rn	Subtract register from A with Borrow	1	1
SUBB	A,direct	Subtract direct byte from A with Borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A with Borrow	1	1
SUBB	A,#data	Subtract immed. data from A with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1
LOGICAL OPERATIONS				
MNEMONIC		**DESTINATION**	**BYTE**	**CYC**
ANL	A,Rn	AND register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,@data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate A Left through the Carry flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate A Right through Carry flag	1	1
SWAP	A	Swap nibbles within the Accumulator	1	1

Table 1 : MHS 51 Instruction Set Description.

DATA TRANSFER				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
MOV	A,Rn	Move register to Accumulator	1	1
MOV	A,direct	Move direct byte to Accumulator	2	1
MOV	A,@Ri	Move indirect RAM to Accumulator	1	1
MOV	A,#data	Move immediate data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move Accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data 16	Load Data Pointer with a 16-bit constant	3	2
MOVC	A,@A + DPTR	Move Code byte relative to DPTR to A	1	2
MOVC	A,@A + PC	Move Code byte relative to PC to A	1	2
MOVX	A,@Ri	Move External RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move External RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to External RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to External RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with Accumulator	1	1
XCH	A,direct	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order nibble in RAM with A	1	1
BOOLEAN VARIABLE MANIPULATION				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
CLR	C	Clear Carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry flag	1	1
SETB	bit	Set direct Bit	2	1
CPL	C	Complement Carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry flag	2	2
ANL	C,/bit	AND complement of direct bit to Carry	2	2
ORL	C,bit	OR direct bit to Carry flag	2	2
ORL	C,/bit	OR complement of direct bit to Carry	2	2
MOV	C,bit	Move direct bit to Carry flag	2	1
MOV	bit,C	Move Carry flag to direct bit	2	2
PROGRAM AND MACHINE CONTROL				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
ACALL	addr11	Absolute Subroutine Call	2	2
LCALL	addr16	Long Subroutine Call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute Jump	2	2
LJMP	addr16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A + DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is Zero	2	2
JNZ	rel	Jump if Accumulator is Not Zero	2	2
JC	rel	Jump if Carry flag is set	2	2
JNC	rel	Jump if No Carry flag	2	2

Table 1. (Cont.)

PROGRAM AND MACHINE CONTROL (cont.)				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
JB	bit,rel	Jump if direct Bit set	3	2
JNB	bit,rel	Jump if direct Bit Not set	3	2
JBC	bit,rel	Jump if direct Bit is set & Clear bit	3	2
CJNE	A,direct,rel	Compare direct to A & Jump if Not Equal	3	2
CJNE	A,#data,rel	Comp. immed. to A & Jump if Not Equal	3	2
CJNE	Rn,#data,rel	Comp. immed. to reg & Jump if Not Equal	3	2
CJNE	@Ri,#data. rel	Comp. immed. to ind. & Jump if Not Equal	3	2
DJNZ	Rn,rel	Decrement register & Jump if Not Zero	2	2
DJNZ	direct. rel	Decrement direct & Jump if Not Zero	3	2
NOP		No operation	1	1

Table 1. (Cont.)

Notes on data addressing modes :

Rn – Working register R0-R7
direct – 128 internal RAM locations, any I/O port, control or status register
@Ri – Indirect internal RAM location addressed by register R0 or R1
#data – 8-bit constant included in instruction
#data 16 – 16-bit constant included as bytes 2 & 3 of instruction
bit – 128 software flags, any I/O pin, control or status bit

Notes on program addressing modes :

addr 16 – Destination address for LCALL & LJMP may be anywhere within the 64-k program memory address space
Addr 11 – Destination address for ACALL & AJMP will be within the same 2-k page of program memory as the first byte of the following instruction
rel – SJMP and all conditional jumps include an 8-bit offset byte. Range is + 127 – 128 bytes relative to the first byte of the following instruction.

6

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr,code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr,code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A,data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6
2F	1	ADD	A,R7
30	3	JNB	bit addr,code addr
31	2	ACALL	code addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A,#data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr,A
43	3	ORL	data addr,#data
44	2	ORL	A,#data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr,A
53	3	ANL	data addr,#data
54	2	ANL	A,#data
55	2	ANL	A,data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	code addr
61	2	AJMP	code addr
62	2	XRL	data addr A
63	3	XRL	data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr

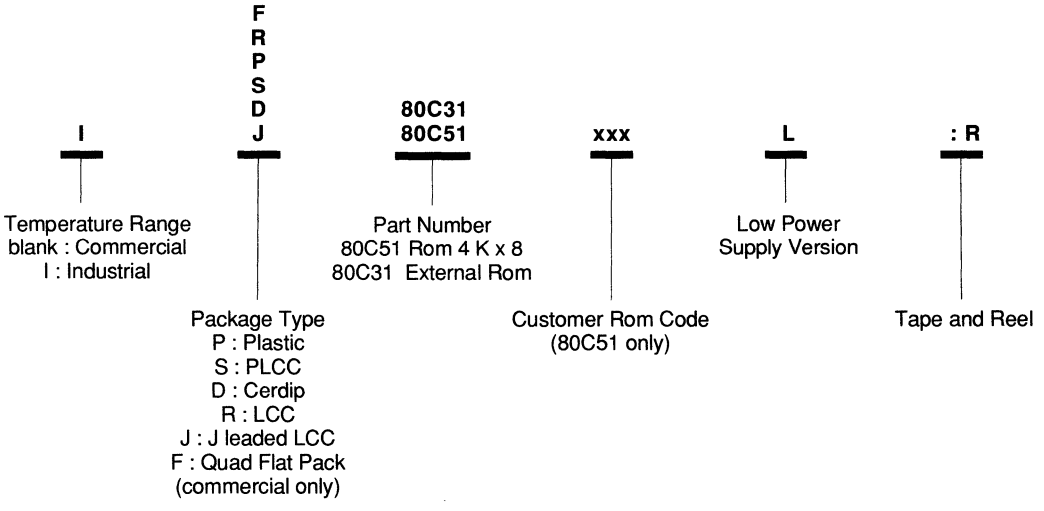
Table 2 : Instruction Opcodes in Hexadecimal Order.

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C,bit addr
73	1	JMP	@A + DPTR
74	2	MOV	A,#data
75	3	MOV	data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C.bit addr
83	1	MOVC	A,@A + PC
84	1	DIV	AB
85	3	MOV	data addr,data addr
86	2	MOV	data addr,@R0
87	2	MOV	data addr,@R1
88	2	MOV	data addr,R0
89	2	MOV	data addr,R1
8A	2	MOV	data addr,R2
8B	2	MOV	data addr,R3
8C	2	MOV	data addr,R4
8D	2	MOV	data addr,R5
8E	2	MOV	data addr,R6
8F	2	MOV	data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	code addr
92	2	MOV	bit addr,C
93	1	MOVC	A,@A + DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,bit addr
A1	2	AJMP	code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,bit addr
B1	2	ACALL	code addr
B2	2	CPL	Bit addr
B3	1	CPL	C
B4	3	CJNE	A,#data,code addr
B5	3	CJNE	A,data addr,code addr
B6	3	CJNE	@R0,#data,code addr
B7	3	CJNE	@R1,#data,code addr
B8	3	CJNE	R0,#data,code addr
B9	3	CJNE	R1,#data,code addr
BA	3	CJNE	R2,#data,code addr
BB	3	CJNE	R3,#data,code addr
BC	3	CJNE	R4,#data,code addr
BD	3	CJNE	R5,#data,code addr
BE	3	CJNE	R6,#data,code addr
BF	3	CJNE	R7,#data,code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3

Table 2. (Cont.)

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

Table 2. (Cont.)



DATA SHEET

80C51F

CMOS SINGLE-CHIP 8 BIT MICROCONTROLLER WITH ROM PROTECTION

- **80C51F - CMOS SINGLE-CHIP 8-BIT MICROCONTROLLER with factory mask-programmable ROM**
- **80C51F : 0 TO 12 MHz**
- **80C51F-1 : 0 TO 16 MHz**

FEATURES

- **SECRET ROM**
- **POWER CONTROL MODES**
- **128 x 8 BIT RAM**
- **32 PROGRAMMABLE I/O LINES**
- **TWO 16-BIT TIMER/COUNTERS**
- **64 K PROGRAM MEMORY SPACE**
- **FULLY STATIC DESIGN**
- **HIGH PERFORMANCE SAJI VI CMOS PROCESS**
- **BOOLEAN PROCESSOR**
- **5 INTERRUPT SOURCES**
- **PROGRAMMABLE SERIAL PORT**
- **64 K DATA MEMORY SPACE**
- **TEMPERATURE RANGE : Commercial, Industrial**

DESCRIPTION

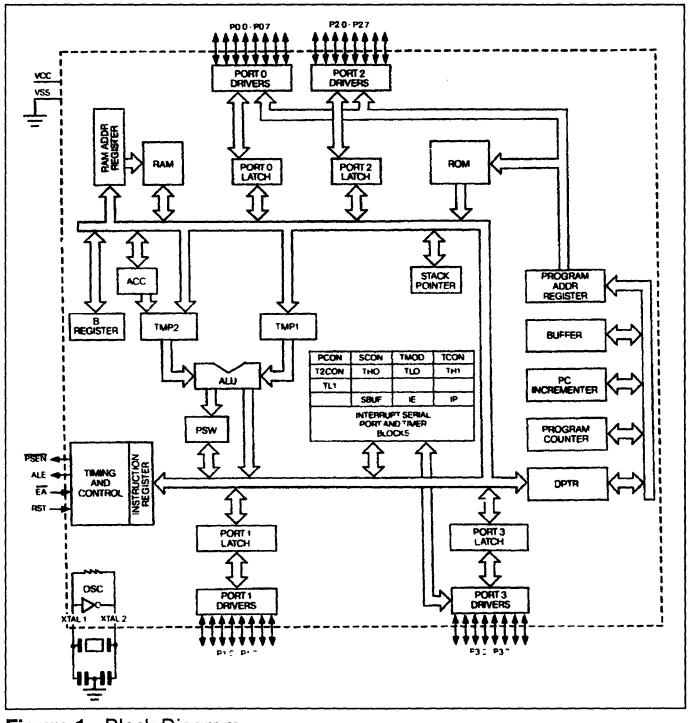


Figure 1 : Block Diagram.

MHS provides a new member in the 80C51 Family named " 80C51F " which permits full protection of the internal ROM contents.

With a non protected 80C51, it is very easy to read out the contents of the internal 4 K bytes of ROM.

Three methods exist, two of them are special test modes and the last one is by means of MOVC instructions.

- **Test mode "VER" :** Using this special test mode, the internal ROM contents are output on port P0 ; the address being applied on ports P2 (AD15..AD8) and P1 (AD7..AD0).
- **Test mode "TMB" :** With this second test mode, the contents of the 80C51 internal bus is presented on port P1 during the PH2 clock phases.
- **Using MOVC instructions :** If EA = 0, and following a reset, the 80C51 fetches its instructions from external program memory. It is then possible to write a small program whose purpose is to dump the internal ROM contents by means of MOVC A, @A + DPTR and MOVC A, @A + PC instructions.

6

80C51F WITH PROGRAM PROTECTION FEATURES

This new version adds ROM protection features in some strategic points of the 80C51F in order to eliminate the possibility of reading the ROM contents (once the protection has been programmed) by one if the three forementioned methods (VER and TMB test modes, or MOVC instructions).

Nevertheless the customer must note the following :

- Once the protection has been programmed, the 80C51F program always starts at address 0 in the internal ROM.
- The application program must be self contained in the internal 4 K of ROM, otherwise it would be possible to trap the program counter address in the external PROM/EPROM (beyond 4 K) and then to dump the internal ROM contents by means of a patch using MOVC instructions.

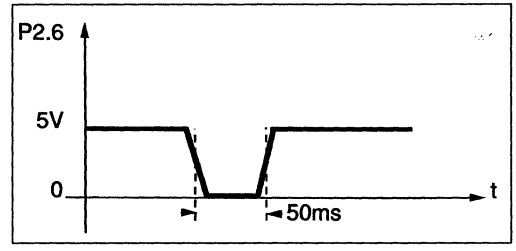
Thus, if an extra EPROM is necessary, it is advised to ensure that it will contain only constants or tables.

TEST OF THE ON-CHIP PROGRAM MEMORY

- **Before protection is activated** : The 80C51F can be tested as any normal 80C51 (using test equipment or any other methods).
- **After protection is activated :** It is then no longer possible to dump the internal ROM contents.

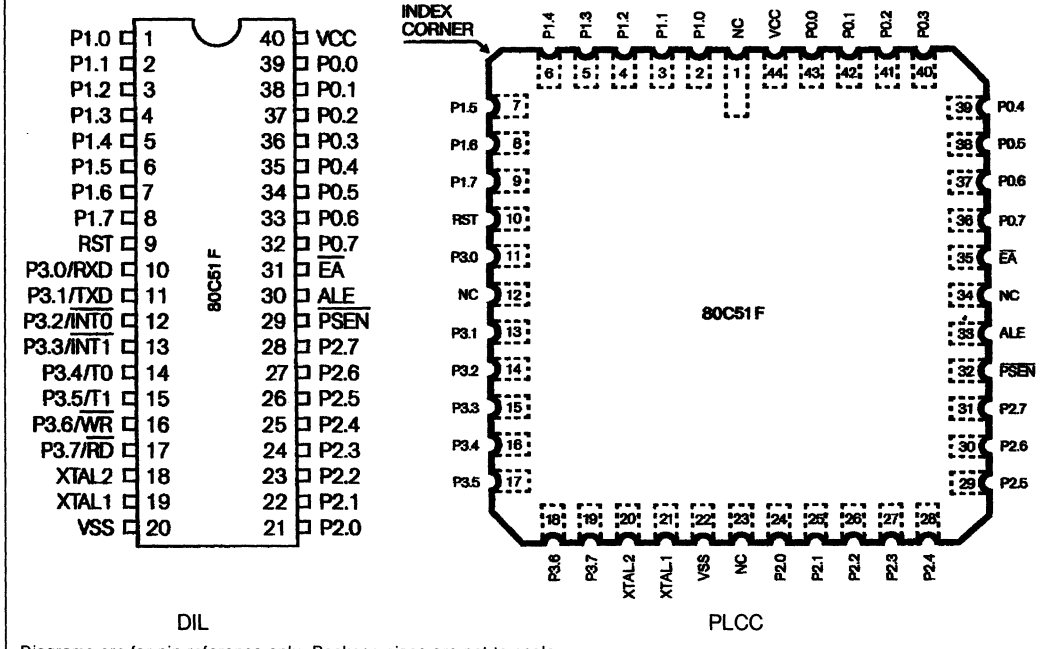
HOW TO PROGRAM THE PROTECTION MECHANISM

- To burn correctly the fuse a specific configuration of inputs must be settled as below :
  - RST = ALE = 1
  - P2.7 = 1
- Futhermore PSEN signal must be tied at + 9 V ± 5 % level voltage and a pulse must be applied on P2.6 input Port. The timing on P2.6 is shown below :



Time Rise and Fall Rise ≤ 100 µs.

- The electrical schematic shows a typical application to deliver P2.6 signal.



Diagrams are for pin reference only. Package sizes are not to scale.

Figure 2 : Configurations.

IDLE POWER DOWN OPERATION

Figure 3 shows the internal Idle and Power Down clock configuration. As illustrated, Power Down operation stops the oscillator. Idle mode operation allows the interrupt, serial port, and timer blocks to continue to function while the clock to the CPU is gated off.
These special modes are activated by software via the Special Function Register, Its Hardware address is 87H. PCON is not bit addressable.

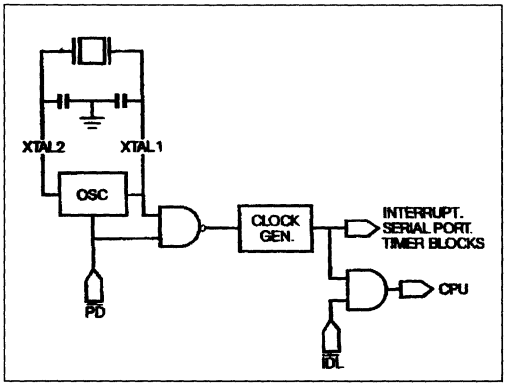


Figure 3 : Idle and Power Down Hardware.

PCON : Power Control Register

(MSB) (LSB)

SMOD	–	–	–	GF1	GF0	PD	IDL

Symbol	Position	Name and Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1, 2 or 3.
–	PCON.6	(Reserved)
–	PCON.5	(Reserved)
–	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activates idle mode operation.

If 1's are written t o PD and IDL at the same time. PD takes precedence. The reset value of PCON is (0XXX0000).

MODE	PROGRAM MEMORY	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Port Data	Port Data	Port Data	Port Data
Idle	External	1	1	Floating	Port Data	Address	Port Data
Power Down	Internal	0	0	Port Data	Port Data	Port Data	Port Data
Power Down	External	0	0	Floating	Port Data	Port Data	Port Data

Table 1 : Status of the external pins during Idle and Power Down modes.

IDLE MODE

The instruction that sets PCON.0 is the last instruction executed before the Idle mode is activated. Once in the Idle mode the CPU status is preserved in its entirety : the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM, and all other registers maintain their data during Idle. *Table 1* describes the status of the external pins during Idle mode.

There are two ways to terminate the Idle mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating Idle mode. The interrupt is serviced, and following RETI, the next instruction to be executed will be the one following the instruction that wrote a 1 to PCON.0.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during the Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an enabled interrupt, the service routine can examine the status of the flag bits.

The second way of terminating the Idle mode is with a hardware reset. Since the oscillator is still running, the hardware reset needs to be active for only 2 machine cycles (24 oscillator periods) to complete the reset operation.

POWER DOWN MODE

The instruction that sets PCON.1 is the last executed prior to entering power down. Once in power down, the oscillator is stopped. The contents of the onchip RAM and the Special Function Register is saved during power down mode. A hardware reset is the only way of exiting the power down mode. The hardware reset initiates the Special Function Register *(see table 1).*

In the Power Down mode, V_{CC} may be lowered to minimize circuit power consumption. Care must be taken to ensure the voltage is not reduced until the power down mode is entered, and that the voltage is restored before the hardware reset is applied which frees the oscillator. Reset should not be released until the oscillator has restarted and stabilized.

Table 1 describes the status of the external pins while in the power down mode. It should be noted that if the power down mode is activated while in external program memory, the port data that is held in the Special Function Register P2 is restored to Port 2. If the data is a 1, the port pin is held high during the power down mode by the strong pullup, T1, shown in *figure 4.*

STOP CLOCK MODE

Due to static design, the MHS 80C51F clock speed can be reduced until 0 MHz without any data loss in memory or registers. This mode allows step by step utilization, and permits to reduce system power consumption by bringing the clock frequency down to any value. At 0 MHz, the power consumption is the same as in the Power Down Mode.

80C51F I/O PORTS

The I/O port drive of the 80C51F is similar to the 8051. The I/O buffers for Ports 1, 2 and 3 are implemented as shown in *figure 4.*

When the port latch contains a 0, all pFETS in *figure 4* are off while the nFET is turned on. When the port latch makes a 0-to-1 transition, the nFET turns off. The strong pullup pFET, T1, turns on for two oscillator periods, pulling the output high very rapidly. As the output line is drawn high, PFET T3 turns on through the inverter to supply the I_{OH} source current. This inverter and T3 form a latch which holds the 1 and is supported by T2. When Port 2 is used as an address port, for access to external program of data memory, any address bit that contains a 1 will have his strong pullup turned on for the entire duration of the external memory access.

When an I/O pin on Ports 1, 2 or 3 is used as an input, the user should be aware that the external circuit must sink current during the logical 1-to-0 transition. The maximum sink current is specified as ITL under the D.C. Specifications. When the input goes below approximately 2 V, T3 turns off to save ICC current. Note, when returning to a logical 1, T2 is the only internal pullup that is on. This will result in a slow rise time if the user's circuit does not force the input line high.

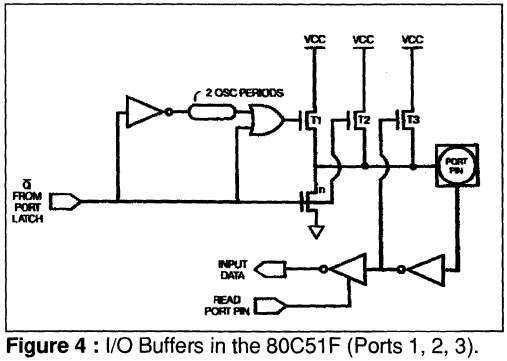


Figure 4 : I/O Buffers in the 80C51F (Ports 1, 2, 3).

80C51F PIN DESCRIPTIONS

Vss

Circuit ground potential

Vcc

Supply voltage during normal, Idle, and Power Down operation.

Port 0

Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1's. Port 0 also outputs the code bytes during program verification in the 80C51F. External pullups are required during program verification. Port 0 can sink eight LSTTL inputs.

Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. Port 1 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address bytes during program verification. In the 80C51F, Port 1 can sink/source three LSTTL inputs. It can drive CMOS inputs without external pullups.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. Port 2 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups. Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @ DPTR). In this application, it uses strong internal pullups when emitting 1's. During accesses to external Data Memory that use 8-bit addresses (MOVX @ Ri), Port 2 emits the contents of the P2 Special Function Register.

It also receives the high-order address bits and control signals during program verification in the 80C51. Port 2 can sink/source three LSTTL inputs. It can drive CMOS inputs without external pullups.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pul-

lups. Port 3 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the pullups. It also serves the functions of various special features of the MHS-51 Family, as listed below.

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	WR (external Data Memory write strobe)
P3.7	RD (external Data Memory read strobe)

Port 3 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

RST

A high level on this for two machine cycles while the oscillator is running resets the device. An internal pulldown resistor permits Power-On reset using only a capacitor connected to V_{CC}.

ALE

Address Latch Enable output for latching the low byte of the address during accesses to external memory. ALE is activated as though for this purpose at a constant rate of 1/6 the oscillator frequency except during an external data memory access at which time one ALE pulse is skipped. ALE can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

PSEN

Program Store Enable output is the read strobe to external Program Memory. PSEN is activated twice each machine cycle during fetches from external Program Memory. (However, when executing out of external Program Memory, two activations of PSEN are skipped during each access to external Data Memory). PSEN is not activated during fetches from internal Program Memory. PSEN can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

EA

When EA is held high, the CPU executes out of internal Program Memory (unless the Program Counter exceeds 0FFFH). When EA is held low, the CPU executes only out of external Program Memory. EA must not be floated.

XTAL1

Input to the inverting amplifier that forms the oscillator. Receives the external oscillator signal when an external oscillator is used.

XTAL2

Output of the inverting amplifier that forms the oscillator, and input to the internal clock generator. This pin should be floated when an external oscillator is used.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output respectively, of an inverting amplifier which is configured for use as an on-chip oscillator, as shown in figure 5. Either a quartz crystal or ceramic resonator may be used.

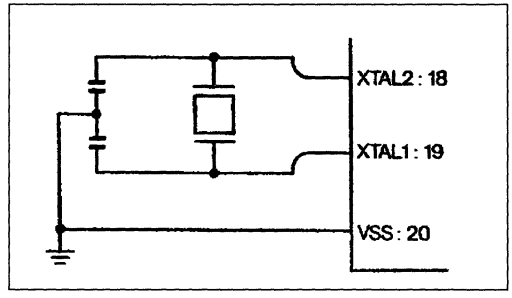


Figure 5 : Crystal Oscillator.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected as shown in *figure 6*. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

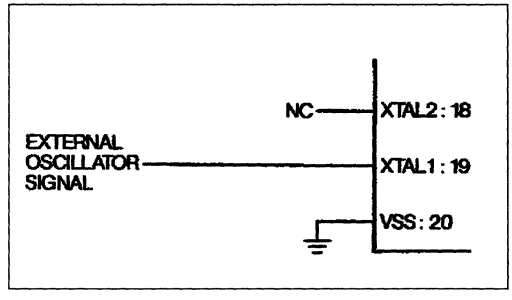


Figure 6 : External Drive Configuration.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias :
C = commercial ..... 0°C to 70°C
I = industrial ..... – 40°C to + 85°C
Storage Temperature ..... – 65°C to + 150°C
Voltage on V_{CC} to V_{SS} ..... – 0.5 V to + 7 V
Voltage on Any Pin to V_{SS} ..... – 0.5 V to VCC + 0.5 V
Power Dissipation ..... 1 W**
** This value is based on the maximum allowable die temperature and the thermal resistance of the package.

* NOTICE :

Stresses at or above those listed under " Absolute Maximum Ratings " may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

DC CHARACTERISTICS (see Note 2)
T_A = – 40°C to 85°C ; VCC = 5 V ± 20 % ; VSS = 0 V ; F = 0 to 12 MHz
T_A = – 40°C to 85°C ; VCC = 5 V ± 10 % ; VSS = 0 V ; F = 0 to 16 MHz

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITIONS
VIL	Input Low Voltage	– 0.5	0.2 VCC – 0.1	V	
VIH	Input High Voltage (Except XTAL and RST)	0.2 VCC + 0.9	VCC + 0.5	V	
VIH1	Input High Voltage (RST and XTAL1)	0.7 VCC	VCC + 0.5	V	
VOL	Output Low Voltage (Ports 1, 2, 3)		0.45	V	IOL = 1.6 mA (note 3)
VOL1	Output Low Voltage Port 0, ALE, $\overline{PSEN}$		0.45	V	IOL = 3.2 mA (note 3)
VOH	Output High Voltage Ports 1, 2, 3	0.9 VCC		V	IOH = – 10 µA
		0.75 VCC		V	IOH = – 25 µA
		2.4		V	IOH = – 60 µA VCC = 5 V ± 10 %
VOH1	Output High Voltage (Port 0, ALE, PSEN)	0.9 VCC		V	IOH = – 80 µA
		0.75 VCC		V	IOH = – 300 µA
		2.4		V	IOH = – 800 µA VCC = 5 V ± 10 %
IIL	Logical 0 Input Current Ports 1, 2, 3		C – 50 I – 60	µA	Vin = 0.45 V
ILI	Logical Leakage Current (Port 0, EA)		± 10	µA	0.45 < Vin < VCC
ITL	Logical 1 to 0 Transition Current (Ports 1, 2, 3)		– 650	µA	Vin = 2.0 V
IPD	Power Supply Current (Power Down Mode)		50	µA	VCC = 2.0 V to 6 V (note 2)
RRST	RST Pulldown Resistor	50	150	kΩ	
CIO	Capacitance of I/O Buffer		10	pF	f_C = 1 MHz, T_A = 25°C
ICC	Power supply current Active mode 12 MHz Idle mode 12 MHz		 20 5	 mA mA	(notes 1, 2)

Note 1 : ICC max is given by :
Active Mode : ICC MAX = 1.47 x FREQ + 2.35
Idle Mode : ICC MAX = 0.33 x FREQ + 1.05
where FREQ is the external oscillator frequency in MHz. ICC MAX is given in mA. See *figure 1*.
See *figures 1* through *5* for ICC test conditions.

6

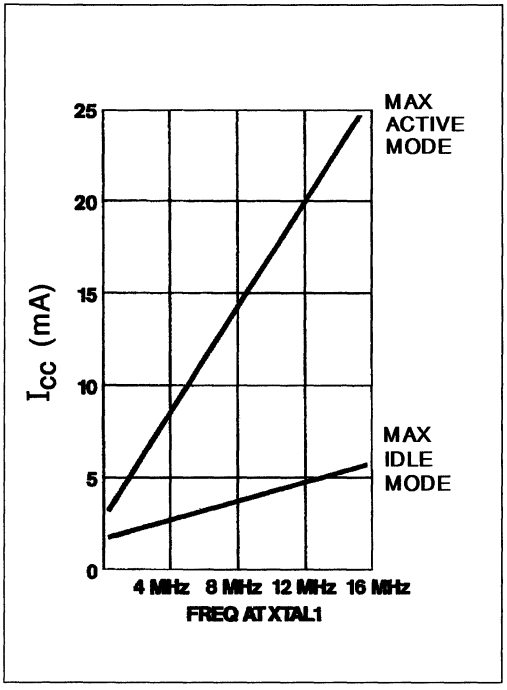


Figure 1 : ICC vs. Frequency. Valid only within frequency specifications of the device under test.

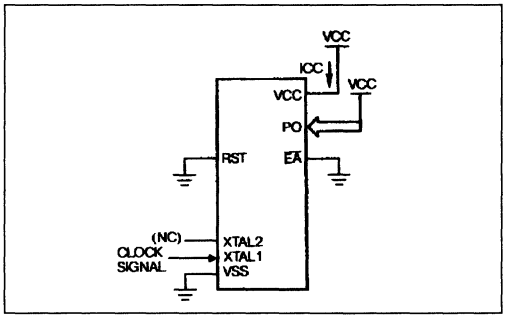


Figure 2 : ICC Test Condition, Idle Mode. All other pins are disconnected.

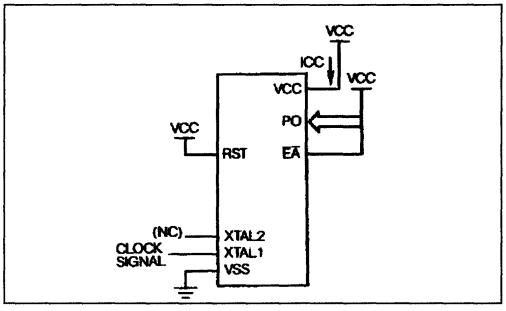


Figure 3 : ICC Test Condition, Active Mode. All other pins are disconnected.

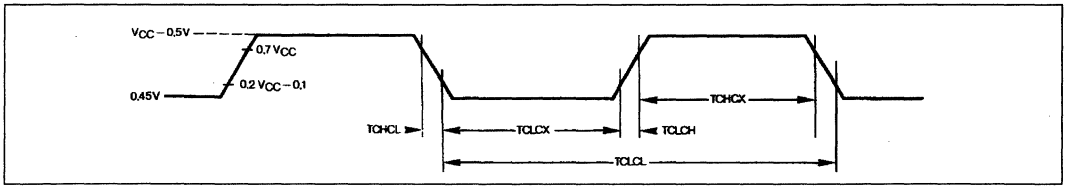


Figure 4 : Clock Signal Waveform for ICC Tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns.

Note 2 : ICC is measured with all output pins disconnected ; XTAL1 driven with TCLCH, TCHCL = 5 ns, VIL = VSS + 5 V, VIH = VCC – .5 V ; XTAL2 N.C. ; EA = RST = Port 0 = VCC. ICC would be slightly higher if a crystal oscillator used. Idle ICC is measured with all output pins disconnected ; XTAL1 driven with TCLCH, TCHCL = 5 ns, VIL = VSS + 5 V,
VIH = VCC – 5 V ; XTAL2 N.C. ; Port 0 = VCC ; EA = RST = VSS.
Power Down ICC is measured with all output pins disconnected ; EA = PORT 0 = VCC ; XTAL2 N.C. ; RST = VSS.

Note 3 : Capacitance loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the VOLS of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1 to 0 transitions during bus operations. In the worst cases (capacitive loading 100 pF), the noise pulse on the ALE line may exceed 0.45 V may exceed 0.45 V with maxi VOL peak 0.6 V A Schmitt Trigger use is not necessary.

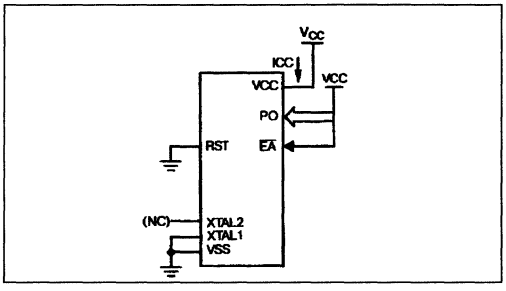


Figure 5 : ICC Test Condition, Power Down Mode. All other pins are disconnected.

EXTERNAL CLOCK DRIVE CHARACTERISTICS (XTAL 1)

SYMBOL	PARAMETER	VARIABLE CLOCK FREQ = 0 to 16 MHz		UNIT
		MIN	MAX	
1/TCLCL	Oscillator Frequency	62.5		ns
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

A.C. PARAMETERS :

TA = – 40°C + 85°C ; VSS = 0 V ; VCC = 5 V ± 20 % F = 0 to 12 MHz
TA = – 40°C + 85°C ; VSS = 0 V ; VCC = 5 V ± 10 % F = 0 to 16 MHz
(Load Capacitance for Port 0, ALE, and $\overline{PSEN}$ = 100 pf ; Load Capacitance for All Other Outputs = 80 pf).

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

SYMBOL	PARAMETER	MIN	MAX	UNIT
TLHLL	ALE Pulse Width	2TCLCL-40		ns
TAVLL	Address Valid to ALE	TCLCL-55		ns
TLLAX	Address Hold After ALE	TCLCL-35		ns
TLLIV	ALE to Valid Instr in		4TCLCL-100	ns
TLLPL	ALE to $\overline{PSEN}$	TCLCL-40		ns
TPLPH	$\overline{PSEN}$ Pulse Width	3TCLCL-45		ns
TPLIV	$\overline{PSEN}$ to Valid Instr in		3TCLCL-105	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		TCLCL-25	ns
TPXAV	$\overline{PSEN}$ to Address Valid	TCLCL-8		ns
TAVIV	Address to Valid Instr in		5TCLCL-105	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		10	ns

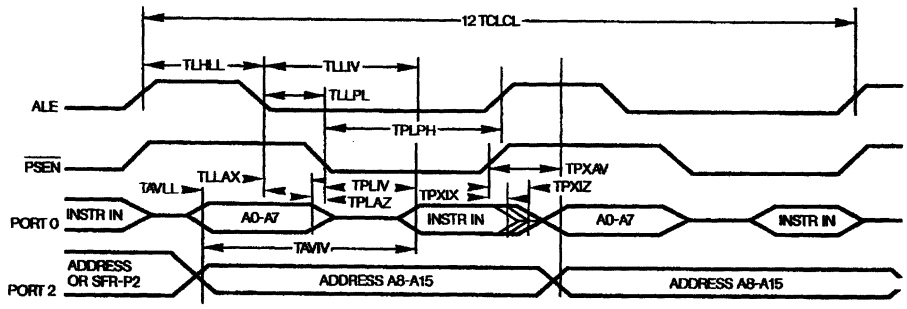
6

EXTERNAL DATA MEMORY CHARACTERISTICS

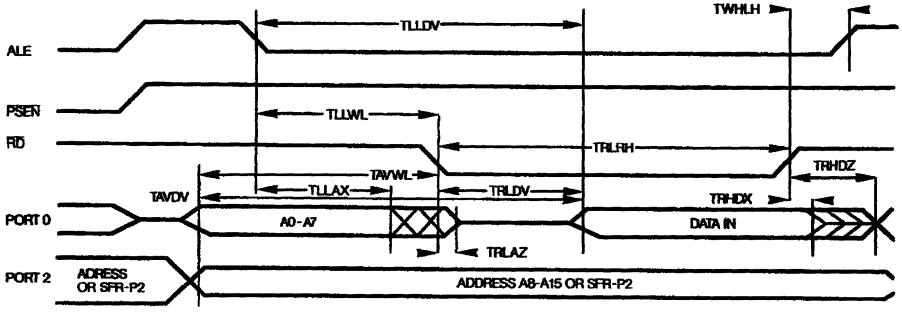
SYMBOL	PARAMETER	MIN	MAX	UNIT
TRLRH	$\overline{RD}$ Pulse Width	6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	6TCLCL-100		ns
TLLAX	Data Address Hold After ALE	TCLCL-50		ns
TRLDV	$\overline{RD}$ to Valid Data in		5TCLCL-165	ns
TRHDX	Data Hold After $\overline{RD}$	0		ns
TRHDZ	Data Float After $\overline{RD}$		2TCLCL-70	ns
TLLDV	ALE to Valid Data in		8TCLCL-150	ns
TAVDV	Address to Valid Data in		9TCLCL-165	ns
TLLWL	ALE to $\overline{WR}$ or $\overline{RD}$	3TCLCL-50	3TCLCL+50	ns
TAVWL	Address to $\overline{WR}$ or $\overline{RD}$	4TCLCL-130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	TCLCL-60		ns
TQVWH	Data Setup to $\overline{WR}$ High	7TCLCL-150		ns
TWHQX	Data Hold After $\overline{WR}$	TCLCL-50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	TCLCL-40	TCLCL+40	ns

AC TIMING DIAGRAMS

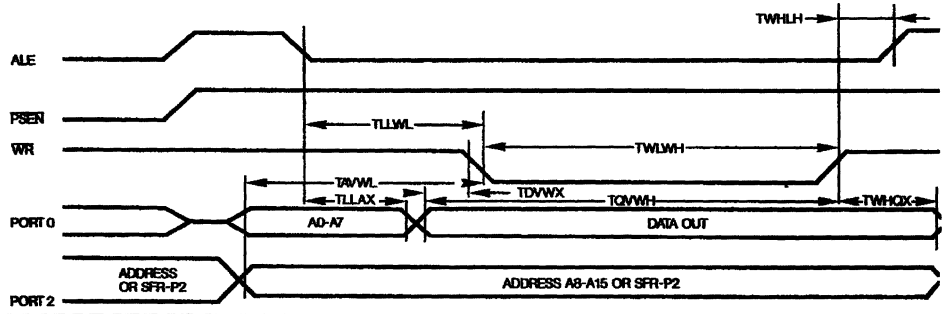
EXTERNAL PROGRAM MEMORY CYCLE



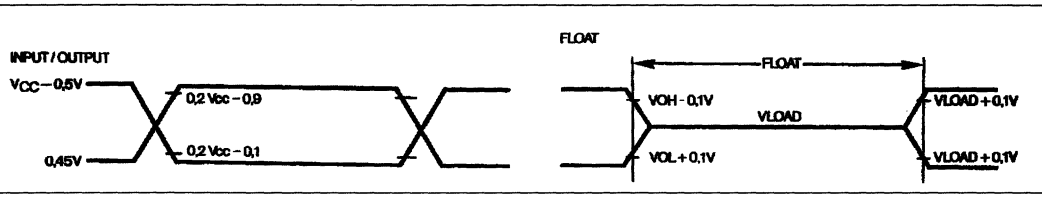
EXTERNAL DATA MEMORY READ CYCLE



EXTERNAL DATA MEMORY WRITE CYCLE



AC TESTING INPUT/OUTPUT, FLOAT WAVEFORMS

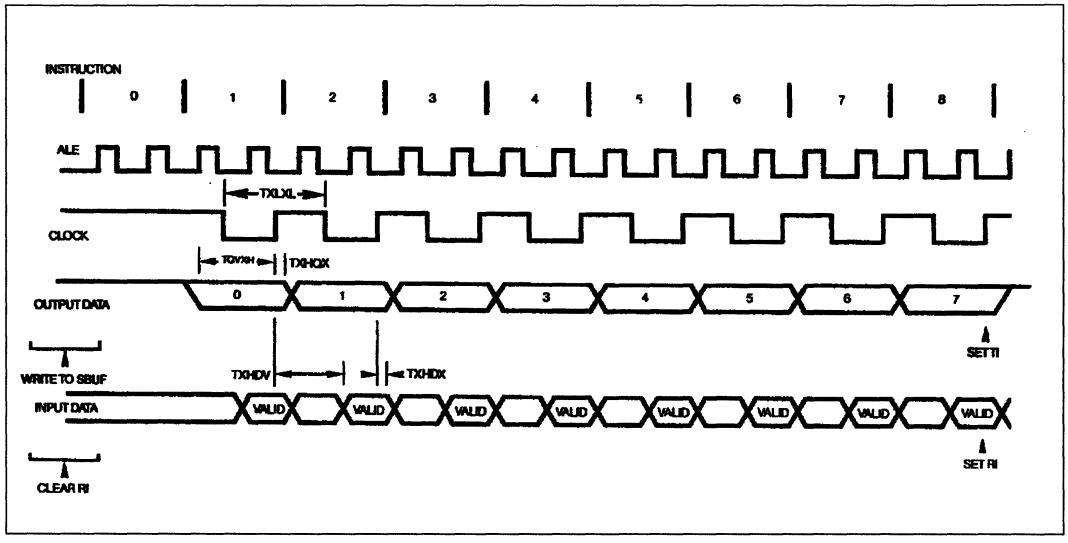


AC inputs during testing are driven at V_{CC} – 0.5 for a logic "1" and 0.45 V for a logic "0". Timing measurements are made at VIH min for a logic "1" and VIL max for a logic "0". For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change from the loaded VOH/VOL level occurs. IOL/IOH ≥ ± 20 mA.

SERIAL PORT TIMING - SHIFT REGISTER MODE

SYMBOL	PARAMETER	MIN	MAX	UNIT
TXLXL	Serial Port Clock Cycle Time	12TCLCL		ns
TQVXH	Output Data Setup to Clock Rising Edge	10TCLCL-133		ns
TXHQX	Output Data Hold After Clock Rising Edge	2TCLCL-117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		ns
TXHDV	Clock Rising Edge to Input Data Valid		10TCLCL-133	ns

SHIFT REGISTER TIMING WAVEFORMS



6

EXPLANATION OF THE AC SYMBOLS

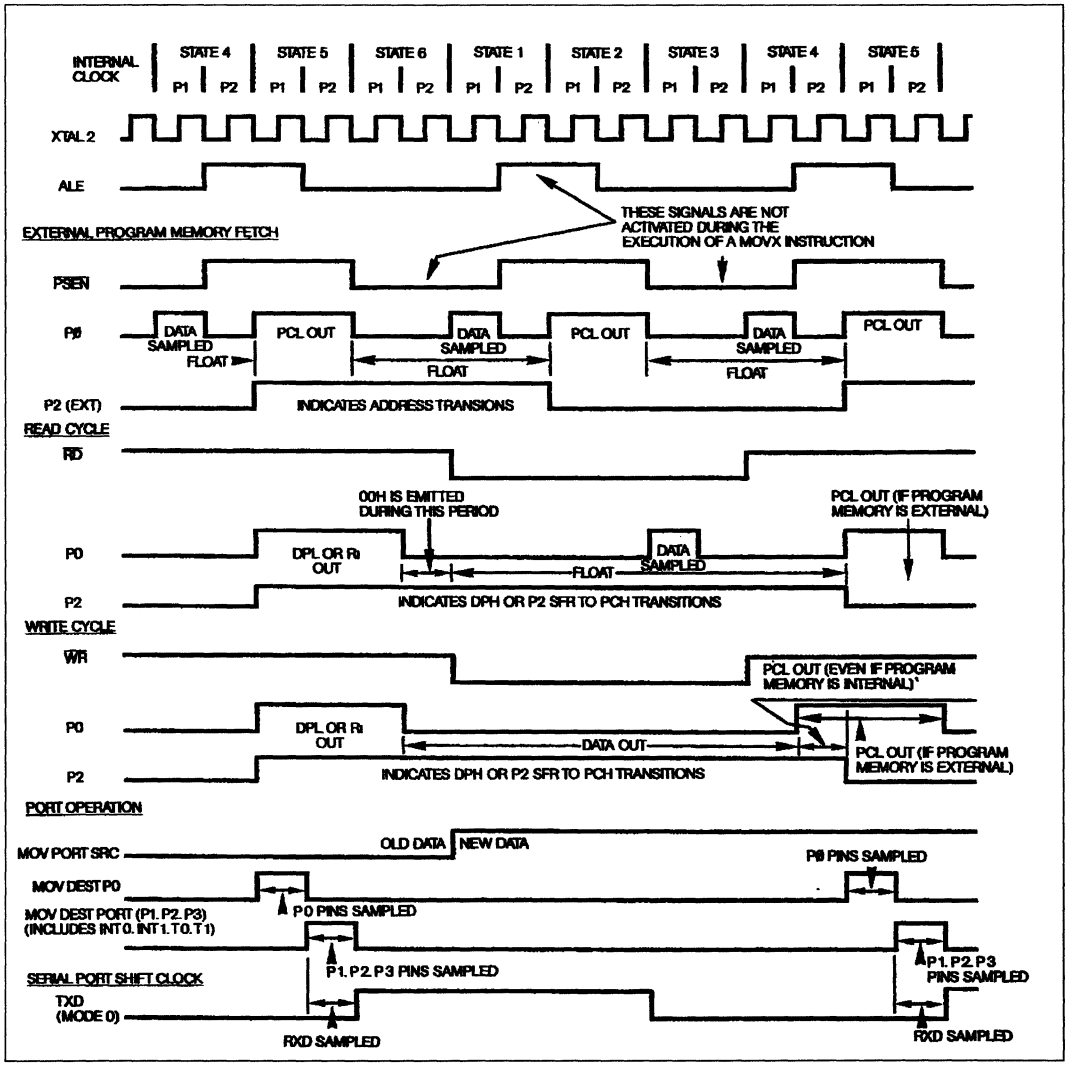
Each timing symbol has 5 characters. The first character is always a " T " (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

Example :

TAVLL = Time for Address Valid to ALE low.
TLLPL = Time for ALE low to PSEN low.

A : Address.
C : Clock.
D : Input data.
H : Logic level HIGH.
I : Instruction (program memory contents).
L : Logic level LOW, or ALE
P : PSEN.
Q : Output data.
R : READ signal.
T : Time.
V : Valid.
W : WRITE signal.
X : No longer a valid logic level.
Z : Float.

CLOCK WAVEFORMS



This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins, however, ranges from 25 to 125 ns. This propagation delay is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component. Typically though (T_A = 25°C fully loaded) RD and WR propagation delays are approximately 50 ns. The other signals are typically 85 ns. Propagation delays are incorporated in the AC specifications.

ARITHMETIC OPERATIONS				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with Carry	1	1
ADDC	A,direct	Add direct byte to A with Carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with Carry flag	1	1
ADDC	A,#data	Add immediate data to A with Carry flag	2	1
SUBB	A,Rn	Subtract register from A with Borrow	1	1
SUBB	A,direct	Subtract direct byte from A with Borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A with Borrow	1	1
SUBB	A,#data	Subtract immed. data from A with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1
LOGICAL OPERATIONS				
MNEMONIC		**DESTINATION**	**BYTE**	**CYC**
ANL	A,Rn	AND register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate A Left through the Carry flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate A Right through Carry flag	1	1
SWAP	A	Swap nibbles within the Accumulator	1	1

Table 1 : MHS - 51 Instruction Set Description.

DATA TRANSFER				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
MOV	A,Rn	Move register to Accumulator	1	1
MOV	A,direct	Move direct byte to Accumulator	2	1
MOV	A,@Ri	Move indirect RAM to Accumulator	1	1
MOV	A,#data	Move immediate data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move Accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data 16	Load Data Pointer with a 16-bit constant	3	2
MOVC	A,@A + DPTR	Move Code byte relative to DPTR to A	1	2
MOVC	A,@A + PC	Move Code byte relative to PC to A	1	2
MOVX	A,@Ri	Move External RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move External RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to External RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to External RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte form stack	2	2
XCH	A,Rn	Exchange register with Accumulator	1	1
XCH	A,direct	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order nibble ind RAM with A	1	1
BOOLEAN VARIABLE MANIPULATION				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
CLR	C	Clear Carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry flag	1	1
SETB	bit	Set direct Bit	2	1
CPL	C	Complement Carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry flag	2	2
ANL	C, /bit	AND complement of direct bit to Carry	2	2
ORL	C, bit	OR direct bit to Carry flag	2	2
ORL	C, /bit	OR complement of direct bit to Carry	2	2
MOV	C, bit	Move direct bit to Carry flag	2	1
MOV	bit,C	Move Carry flag to direct bit	2	2
PROGRAM AND MACHINE CONTROL				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
ACALL	addr11	Absolute Subroutine Call	2	2
LCALL	addr16	Long Subroutine Call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute Jump	2	2
LJMP	addr16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A + DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is Zero	2	2
JNZ	rel	Jump if Accumulator is Not Zero	2	2
JC	rel	Jump if Carry flag is set	2	2
JNC	rel	Jump if No Carry flag	2	2

Table 1. (Cont.)

PROGRAM AND MACHINE CONTROL (cont.)				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
JB	bit,rel	Jump if direct Bit set	3	2
JNB	bit,rel	Jump if direct Bit Not set	3	2
JBC	bit,rel	Jump if direct Bit is set & Clear bit	3	2
CJNE	A,direct, rel	Compare direct to A & Jump if Not Equal	3	2
CJNE	A,#data, rel	Comp. immed. to A & Jump if Not Equal	3	2
CJNE	Rn,#data, rel	Comp. immed. to reg & Jump if Not Equal	3	2
CJNE	@Ri,#data. rel	Comp. immed. to ind. & Jump if Not Equal	3	2
DJNZ	Rn,rel	Decrement register & Jump if Not Zero	2	2
DJNZ	direct,rel	Decrement direct & Jump if Not Zero	3	2
NOP		No operation	1	1

Table 1. (Cont.)

Notes on data addressing modes :

Rn – Working register R0-R7
direct – 128 internal RAM locations, any I/O port, control or status register
@Ri – Indirect internal RAM location addressed by register R0 or R1
#data – 8-bit constant included in instruction
#data 16 – 16-bit constant included as bytes 2 & 3 of instruction
bit – 128 software flags, any I/O pin, control or status bit

Notes on program addressing modes :

addr 16 – Destination address for LCALL & LJMP may be anywhere within the 64-k program memory address space
Addr 11 – Destination address for ACALL & AJMP will be within the same 2-k page of program memory as the first byte of the following instruction
rel – SJMP and all conditional jumps include an 8-bit offset byte. Range is + 127 – 128 bytes relative to the first byte of the following instruction.

6

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr,code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr,code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A,data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6
2F	1	ADD	A,R7
30	3	JNB	bit addr,code addr
31	2	ACALL	code addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A,#data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr,A
43	3	ORL	data addr,#data
44	2	ORL	A,#data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr,A
53	3	ANL	data addr,#data
54	2	ANL	A,#data
55	2	ANL	A,data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	code addr
61	2	AJMP	code addr
62	2	XRL	data addr A
63	3	XRL	data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr

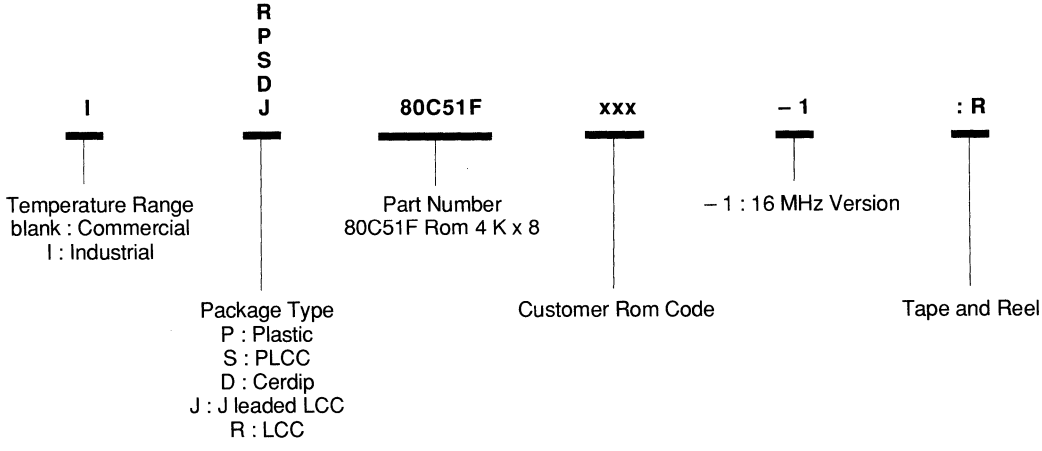
Table 2 : Instruction Opcodes in Hexadecimal Order.

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C,bit addr
73	1	JMP	@A + DPTR
74	2	MOV	A,#data
75	3	MOV	data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C,bit addr
83	1	MOVC	A, @A + PC
84	1	DIV	AB
85	3	MOV	data addr,data addr
86	2	MOV	data addr,@R0
87	2	MOV	data addr,@R1
88	2	MOV	data addr,R0
89	2	MOV	data addr,R1
8A	2	MOV	data addr,R2
8B	2	MOV	data addr,R3
8C	2	MOV	data addr,R4
8D	2	MOV	data addr,R5
8E	2	MOV	data addr,R6
8F	2	MOV	data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	code addr
92	2	MOV	bit addr,C
93	1	MOVC	A,@A + DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,bit addr
A1	2	AJMP	code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,bit addr
B1	2	ACALL	code addr
B2	2	CPL	Bit addr
B3	1	CPL	C
B4	3	CJNE	A, #data, code addr
B5	3	CJNE	A, data addr, code addr
B6	3	CJNE	@R0,#data, code addr
B7	3	CJNE	@R1,#data, code addr
B8	3	CJNE	R0,#data, code addr
B9	3	CJNE	R1,#data, code addr
BA	3	CJNE	R2,#data, code addr
BB	3	CJNE	R3,#data, code addr
BC	3	CJNE	R4,#data, code addr
BD	3	CJNE	R5,#data, code addr
BE	3	CJNE	R6,#data, code addr
BF	3	CJNE	R7,#data, code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3

Table 2. (Cont.)

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr, code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR, A
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

Table 2. (Cont.)



6

DATA SHEET

80C52/80C32

CMOS SINGLE-CHIP 8 BIT MICROCONTROLLER

- **80C52 - CMOS SINGLE -CHIP 8 BIT MICRO-CONTROLLER with factory mask-programmable ROM**
- **80C32 - CMOS SINGLE - CHIP 8-BIT CONTROL ORIENTED CPU with RAM and I/O**

80C52/C32 : 0 to 12 MHz
80C52-1/C32-1 : 0 to 16 MHz
80C52S/C32S : 0 to 20 MHz
80C52-L/C32-L : V_{CC} = 2.7 V to 5.5 V (0 to 6 MHz)
80C52F : SECRET ROM

FEATURES

- **POWER CONTROL MODES**
- **256 x 8 BIT RAM**
- **32 PROGRAMMABLE I/O LINES**
- **THREE 16-BIT TIMER/COUNTER**
- **64 K PROGRAM MEMORY SPACE**
- **FULLY STATIC DESIGN**
- **HIGH PERFORMANCE SAJI VI CMOS PROCESS**
- **BOOLEAN PROCESSOR**
- **6 INTERRUPT SOURCES**
- **PROGRAMMABLE SERIAL PORT**
- **64 K DATA MEMORY SPACE**
- **TEMPERATURE RANGE : Commercial, Industrial, Automotive and Military**

DESCRIPTION

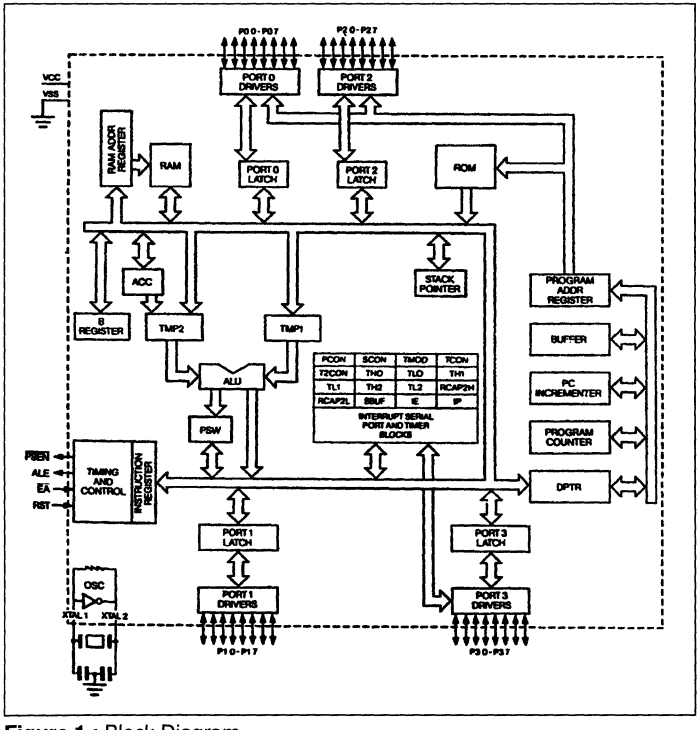


Figure 1 : Block Diagram.

MHS's 80C52 and 80C32 are high performance CMOS versions of the 8052/8032 NMOS single chip 8 bit µC and is manufactured using a self-aligned silicon gate CMOS process (SAJI VI).

The fully static design of the MHS 80C52/80C32 allows to reduce system power consumption by bringing the clock frequency down to any value, even DC, without loss of data.

The 80C52 retains all the features of the 8052 : 8 K bytes of ROM ; 256 bytes of RAM ; 32 I/O lines ; three 16 bit timers ; a 6-source, 2-level interrupt structure ; a full duplex serial port ; and on-chip oscillator and clock circuits.

In addition, the 80C52 has two software-selectable modes of reduced activity for further reduction in power consumption. In the Idle Mode the CPU is frozen while the RAM, the timers, the serial port, and the interrupt system continue to function. In the Power Down Mode the RAM is saved and all other functions are inoperative.

The 80C32 is identical to the 80C52 except that it has no on-chip ROM.

MHS provides a new member in the 80C52 Family named "80C52F" which permits full protection of the internal ROM contents.

With a non protected 80C52, it is very easy to read out the contents of the internal 8 K bytes of ROM.

Three methods exist, two of them are special test modes and the last one is by means of MOVC instructions.

- **Test mode "VER" :** Using this special test mode, the internal ROM contents are output on port P0 ; the address being applied on ports P2 (AD15...AD8) and P1 (AD7...AD0).
- **Test mode "TMB" :** With this second test mode, the contents of the 80C52 internal bus is presented on port P1 during the PH2 clock phases.
- **Using MOVC instructions :** If EA = 0, and following a reset, the 80C52 fetches its instructions from external program memory. It is then possible to write a small program whose purpose is to dump the internal ROM contents by means of MOVC A, @A + DPTR and MOVC A, @A + PC instructions.

80C52F WITH PROGRAM PROTECTION FEATURES

This new version adds ROM protection features in some strategic points of the 80C52F in order to eliminate the possibility of reading the ROM contents (once the protection has been programmed) by one if the three forementioned methods (VER and TMB test modes, or MOVC instructions).

Nevertheless the customer must note the following :

– Once the protection has been programmed, the 80C52F program always starts at address 0 in the internal ROM.

– The application program must be self contained in the internal 8 K of ROM, otherwise it would be possible to trap the program counter address in the external PROM/EPROM (beyond 8 K) and then to dump the internal ROM contents by means of a patch using MOVC instructions.

Thus, if an extra EPROM is necessary, it is advised to ensure that it will contain only constants or tables.

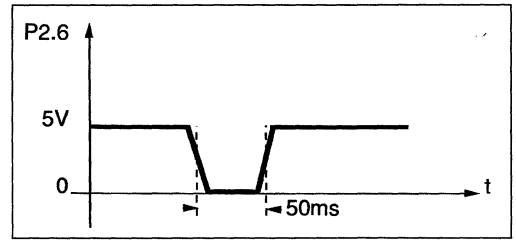
TEST OF THE ON-CHIP PROGRAM MEMORY

- **Before protection is activated :** The 80C52F can be tested as any normal 80C52 (using test equipment or any other methods).
- **After protection is activated :** It is then no longer possible to dump the internal ROM contents.

HOW TO PROGRAM THE PROTECTION MECHANISM

- To burn correctly the fuse a specific configuration of inputs must be settled as below :
  – RST = ALE = 1
  – P2.7 = 1

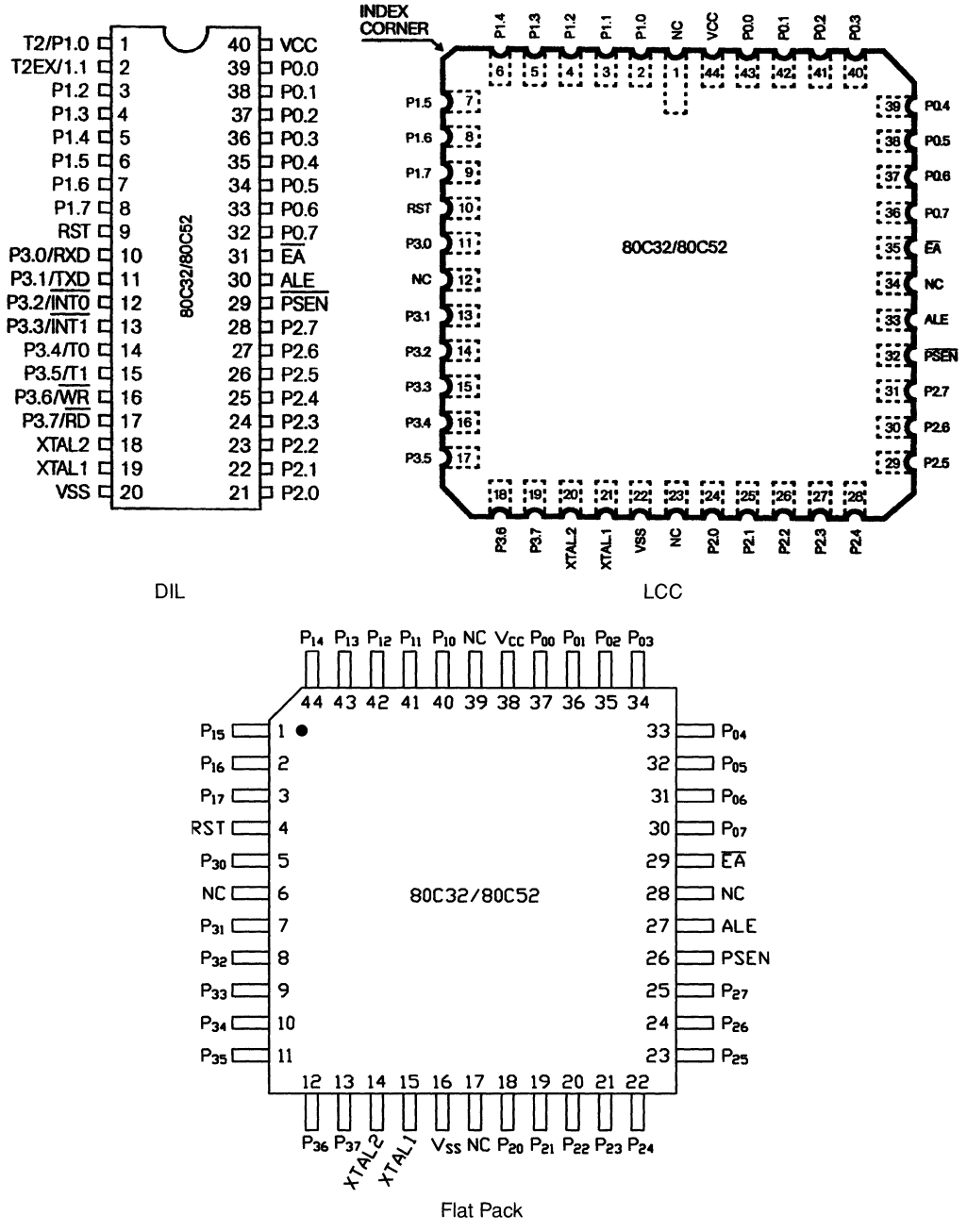
Furthermore PSEN signal must be tied at + 9 V ± 5 % level voltage and a pulse must be applied on P2.6 input Port. The timing on P2.6 is shown below :



Time Rise and Fall Rise ≤ 100 µs.

- The electrical schematic shows a typical application to deliver P2.6 signal.

6



Diagrams are for reference only. Package sizes are not to scale.

Figure 4 : Configurations.

IDLE AND POWER DOWN OPERATION

Figure 5 shows the internal Idle and Power Down clock configuration. As illustrated, Power Down operation stops the oscillator. Idle mode operation allows the interrupt, serial port, and timer blocks to continue to function while the clock to the CPU is gated off.
These special modes are activated by software via the Special Function Register, PCON. Its hardware address is 87H. PCON is not bit addressable.

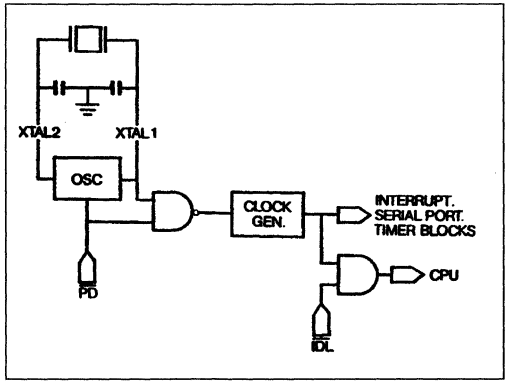


Figure 5 : Idle and Power Down Hardware.

PCON : Power Control Register

(MSB) (LSB)

SMOD	–	–	–	GF1	GF0	PD	IDL

Symbol	Position	Name and Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1, 2 or 3.
–	PCON.6	(Reserved)
–	PCON.5	(Reserved)
–	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activates idle mode operation.

If 1's are written to PD and IDL at the same time. PD takes precedence. The reset value of PCON is (0XXX0000).

IDLE MODE

The instruction that sets PCON.0 is the last instruction executed before the Idle mode is activated. Once in the Idle mode the CPU status is preserved in its entirety : the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM, and all other register maintain their data during Idle. *Table 2* describes the status of the external pins during Idle mode.

There are two ways to terminate the Idle mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating Idle mode. The interrupt is serviced, and following RETI, the next instruction to be executed will be the one following the instruction that wrote 1 to PCON.0.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during the Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an enabled interrupt, the service routine can examine the status of the flag bits.

The second way of terminating the Idle is with a hardware reset. Since the oscillator is still running, the hardware reset needs to be active for only 2 machine cycles (24 oscillator periods) to complete the reset operation.

Power Down Mode

The instruction that sets PCON.1 is the last executed prior to entering power down. Once in power down, the oscillator is stopped. The contents of the onchip RAM and the Special Function Register is saved during power down mode. A hardware reset is the only way of exiting the power down mode. the hardware reset initiates the Special Function Register (see *Table 2*). In the Power Down mode, V_{CC} may be lowered to minimize circuit power consumption. Care must be taken to ensure the voltage is not reduced until the power down mode is entered, and that the voltage is restored before the hardware reset is applied which frees the oscillator. Reset should not be released until the oscillator has re-

MODE	PROGRAM MEMORY	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Port Data	Port Data	Port Data	Port Data
Idle	External	1	1	Floating	Port Data	Address	Port Data
Power Down	Internal	0	0	Port Data	Port Data	Port Data	Port Data
Power Down	External	0	0	Floating	Port Data	Port Data	Port Data

Table 2 : Status of the external pins during Idle and Power Down modes.

started and stabilized.

Table 2 describes the status of the external pins while in the power down mode. It should be noted that if the power down mode is activated while in external program memory, the port data that is held in the Special Function Register P2 is restored to Port 2. If the data is a 1, the port pin is held high during the power down mode by the strong pullup, T1, shown in *Figure 6.*

STOP CLOCK MODE

Due to static design, the MHS 80C32/C52 clock speed can be reduced until 0 MHz without any data loss in memory or registers. This mode allows step by step utilization, and permits to reduce system power consumption by bringing the clock frequency down to any value. At 0 MHz, the power consumption is the same as in the Power Down Mode.

80C52 I/O PORTS

The I/O port drive of the 80C52 is similar to the 8052. The I/O buffers for Ports 1, 2 and 3 are implemented as shown in *figure 6.*
When the port latch contains a 0, all pFETS in figure 6 are off while the nFET is turned on. When the port latch makes a 0-to-1 transition, the nFET turns off. The strong pFET, T1, turns on for two oscillator periods, pulling the output high very rapidly. As the output line is drawn high, pFET T3 turns on through the inverter to supply the I_{OH} source current. This inverter and T form a latch which holds the 1 and is supported by T2.

When Port 2 is used as an address port, for access to external program of data memory, any address bit that contains a 1 will have his strong pullup turned on for the entire duration of the external memory access.

When an I/O pin on Ports 1, 2 or 3 is used as an input, the user should be aware that the external circuit must sink current during the logical 1-to-0 transition. The maximum sink current is specified as ITL under the D.C.

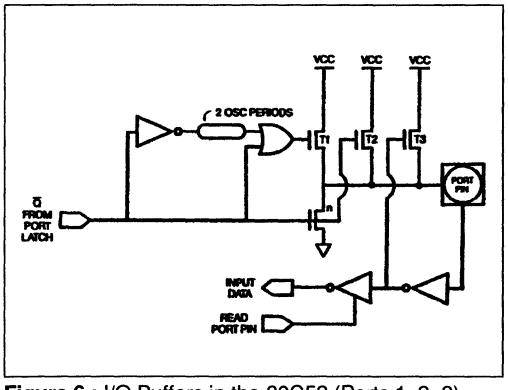


Figure 6 : I/O Buffers in the 80C52 (Ports 1, 2, 3).

Specifications. When the input goes below approximately 2 V, T3 turns off to save ICC current. Note, when returning to a logical 1, T2 is the only internal pullup that is on. This will result in a slow rise time if the user's circuit does not force the input line high.

PIN DESCRIPTIONS

V_{CC}

Supply voltage during normal, Idle, and Power Down operation.

Port 0

Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1's. Port 0 also outputs the code bytes during program verification in the 80C52. External pullups are required during program verification. Port 0 can sink eight LS TTL inputs.

Port 1

Port is an 8-bit bi-directional I/O port with internal pullups. Port 1 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address byte during program verification. In the 80C52, Port 1 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. Port 2 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups. Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pullups when emitting 1's. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

It also receives the high-order address bits and control signals during program verification in the 80C52. Port 2 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. Port 3 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the pullups. It also serves the function of various special features of the MHS 51 Family, as listed below.

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	WR (external Data Memory write strobe)
P3.7	RD (external Data Memory read strobe)

Port 3 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

RST

A high level on this for two machine cycles while the oscillator is running resets the device. An internal pulldown resistor permits Power-On reset using only a capacitor connected to V_{CC}.

ALE

Address Latch Enable output for latching the low byte of the address during accesses to external memory. ALE is activated as though for this purpose at a constant rate of 1/6 the oscillator frequency except during an external data memory access at which time on ALE pulse is skipped. ALE can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

$\overline{PSEN}$

Program Store Enable output is the read strobe to external Program Memory. PSEN is activated twice each machine cycle during fetches from external Program Memory. (However, when executing out of external Program Memory, two activations of PSEN are skipped during each access to external Data Memory). PSEN is not activated during fetches from internal Program Memory. PSEN can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

$\overline{EA}$

When EA is held high, the CPU executed out of internal Program Memory (unless the Program Counter exceeds 1FFFH). When EA is held low, the CPU executes only out of external Program Memory. EA must not be floated.

XTAL1

Input to the inverting amplifier that forms the oscillator. Receives the external oscillator signal when an external oscillator is used.

XTAL2

Output of the inverting amplifier that forms the oscillator, and input to the internal clock generator. This pin should be floated when an external oscillator is used.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output respectively, of an inverting amplifier which is configured for use as an on-chip oscillator, as shown in figure 7. Either a quartz crystal or ceramic resonator may be used.
To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected as shown in *figure 8*. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

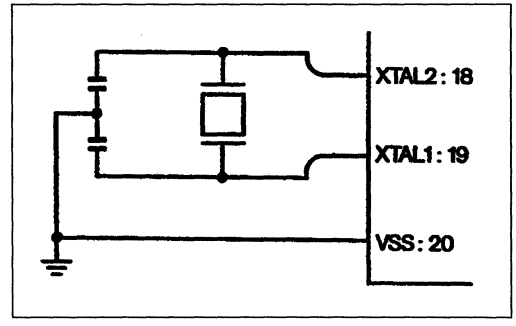


Figure 7 : Crystal Oscillator.

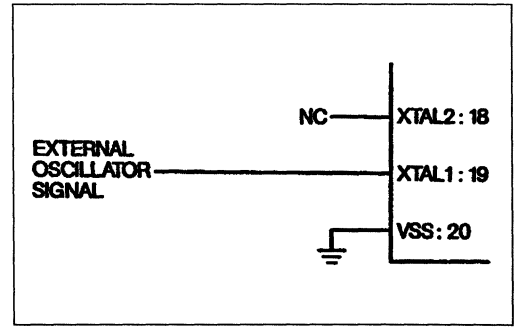


Figure 8 : External Drive Configuration.

TIMER/EVENT COUNTER 2

Timer 2 is a 16-bit timer/counter like Timers 0 and 1, it can operate either as a timer or as an event counter. This is selected by bit C/$\overline{T2}$ in the Special Function Register T2CON (Figure 1). It has three operating modes : "capture", "autoload" and "baud rate generator", which are selected by bits in T2CON as shown in

RCLK + TCLK	CP/$\overline{RL2}$	TR2	MODE
0	0	1	16-bit auto-reload
0	1	1	16-bit capture
1	X	1	baud rate generator
X	X	0	(off)

Table 1 : Timer 2 Operating Modes.

Table 1.

In the capture mode there are two options which are selected by bit EXEN2 in T2CON; If EXEN2 = 0, then Timer 2 is a 16-bit timer or counter which upon overflowing sets bit TF2, the Timer 2 overflow bit, which can be used to generate an interrupt. If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into registers RCAP2L and RCAP2H, respectively, (RCAP2L and RCAP2H are new Special Function Register in the 80C52). In addition, the transition at T2EX causes bit EXF2 in T2CON to be set, and EXF2, like TF2, can generate an interrupt.

The capture mode is illustrated in *Figure 2.*

In the auto-reload mode there are again two options, which are selected by bit EXEN2 in T2CON.If EXEN2 = 0, then when Timer 2 rolls over it does not only set TF2 but also causes the Timer 2 register to be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX will also trigger the 16-bit reload and set EXF2.

The auto-reload mode is illustrated in *Figure 3.*

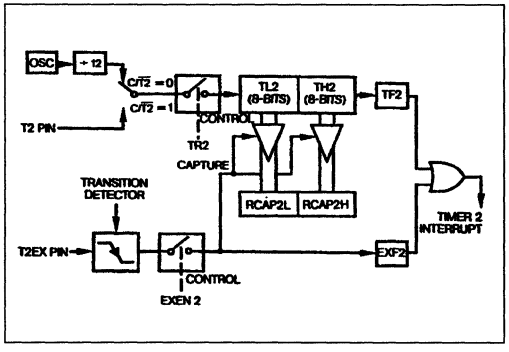


Figure 2 : Timer 2 in Capture Mode.

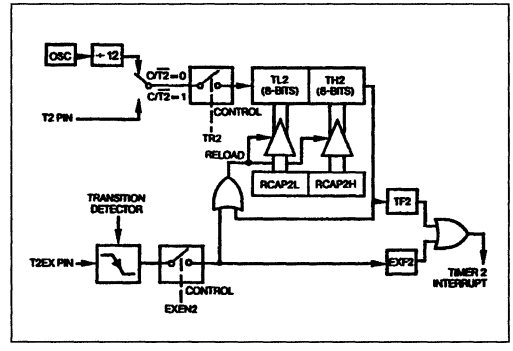


Figure 3 : Timer in Auto-Reload Mode.

6

(MSB)							(LSB)
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

The baud rate generator mode is selected by :
RCLK = 1 and/or TCLK = 1.

Symbol	Position	Name and Significance
TF2	T2CON.7	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 OR TCLK = 1.
EXF2	T2CON.6	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software.
RCLK	T2CON.5	Receive clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	T2CON.4	Transmit clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	T2CON.3	Timer 2 external enable flag. When set, allows capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	T2CON.2	Start/stop control for Timer 2. A logic 1 starts the timer.
C/$\overline{T2}$	T2CON.1	Timer or counter select. (Timer 2) 0 = Internal timer (OSC/12) 1 = External event counter (falling edge triggered).
CP/$\overline{RL2}$	T2CON.0	Capture/Reload flag. When set, captures will occur on negative transitions at T2EX if EXEN 2 = 1. When cleared, auto reloads will occur either with Timer 2

T2CON : Timer/Counter 2 Control Register.

6

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias :
C = commercial ..... 0°C to 70°C
I = industrial ..... – 40°C to +85°C
Storage Temperature ..... – 65°C to + 150°C
Voltage on V_{CC} to V_{SS} ..... – 0.5 V to + 7 V
Voltage on Any Pin to V_{SS} ..... – 0.5 V to V_{CC} + 0.5 V
Power Dissipation ..... 1 W**
** This value is based on the maximum allowable die temperature and the thermal resistance of the package.

* NOTICE

Stresses at or above those listed under " Absolute Maximum Ratings " may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

DC CHARACTERISTICS

T_A = – 40°C to 85°C ; VSS = 0 V ; VCC = 5 V ± 10 % ; F = 0 to 16 MHz

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITIONS
VIL	Input Low Voltage	– 0.5	0.2 VCC – 0.1	V	
VIH	Input High Voltage (Except XTAL and RST)	0.2 VCC + 0.9	VCC + 0.5	V	
VIH1	Input High Voltage (RST and XTAL1)	0.7 VCC	VCC + 0.5	V	
VOL	Output Low Voltage (Port 1, 2, 3)		0.45	V	IOL = 1.6 mA (note 3)
VOL1	Output Low Voltage Port 0, ALE, $\overline{PSEN}$		0.45	V	IOL = 3.2 mA (note 3)
VOH	Output High Voltage Ports 1, 2, 3	0.9 VCC		V	IOH = – 10 µA
		0.75 VCC		V	IOH = – 25 µA
		2.4		V	IOH = – 60 µA VCC = 5 V ± 10 %
VOH1	Output High Voltage (Port 0 in External Bus Mode, ALE, PSEN)	0.9 VCC		V	IOH = – 80 µA
		0.75 VCC		V	IOH = – 300 µA
		2.4		V	IOH = – 800 µA VCC = 5 V ± 10 %
IIL	Logical 0 Input Current Ports 1, 2, 3		C – 50 I – 60	µA	Vin = 0.45 V
ILI	Input Leakage Current (Port 0, $\overline{EA}$)		± 10	µA	0.45 < Vin < VCC
ITL	Logical 1 to 0 Transition Current (Ports 1, 2, 3)		– 650	µA	Vin = 2.0 V
IPD	Power Supply Current (Power Down Mode)		50	µA	VCC = 2.0 V to 6 V (note 2)
RRST	RST Pulldown Resistor	50	150	kΩ	
CIO	Capacitance of I/O Buffer		10	pF	f_C = 1 MHz, T_A = 25°C
ICC	Power Supply Current				(notes 1, 2)
	Active Mode 12 MHz		22	mA	
	16 MHz		27	mA	
	20 MHz		32	mA	
	Idle Mode 12 MHz		7	mA	
	16 MHz		9	mA	
	20 MHz		11	mA	

Note 1 : See figures 9 through 12 for ICC test conditions.

6

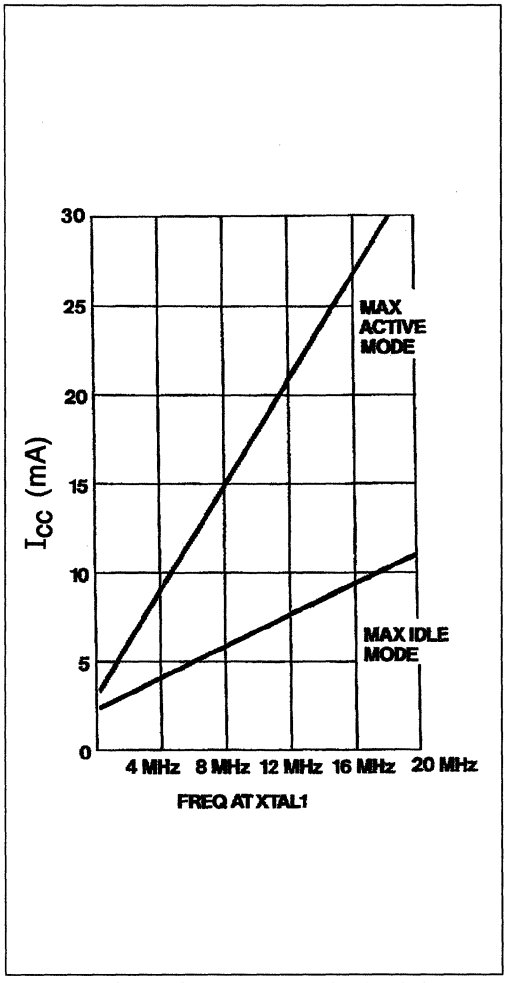


Figure 9 : ICC vs. Frequency. Valid only within frequency specifications of the device under test.

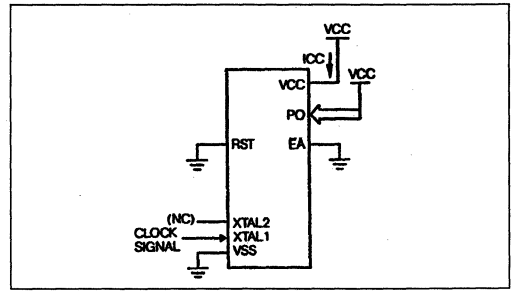


Figure 10 : ICC Test Condition, Idle Mode. All other pins are disconnected.

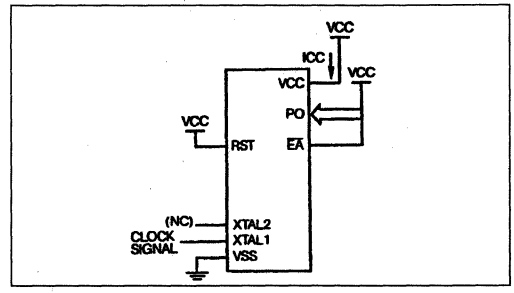


Figure 11 : ICC Test Condition, Active Mode. All other pins are disconnected.

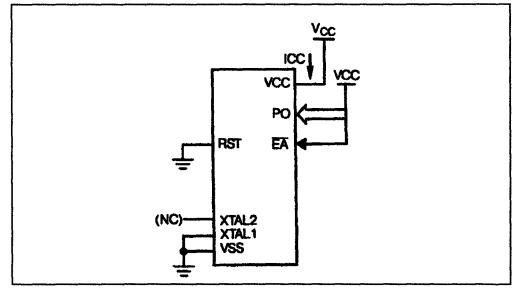


Figure 12 : ICC Test Condition, Power Down Mode. All other pins are disconnected.

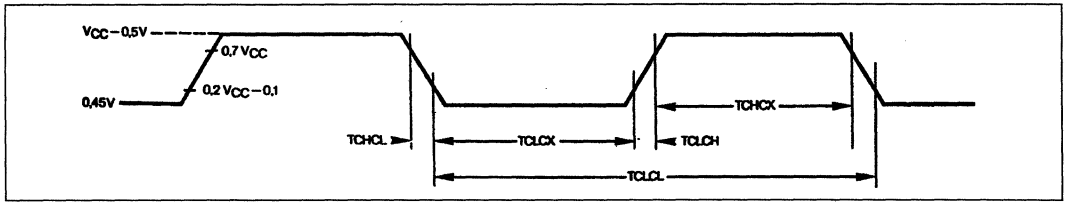


Figure 13 : Clock Signal Waveform for ICC Tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns.

Note 2 : ICC is measured with all output pins disconnected ; XTAL1 driven with TCLCH, TCHCL = 5 ns, VIL = VSS + .5 V, VIH = VCC – .5 V ; XTAL2 N.C. ; EA = RST = Port 0 = VCC. ICC would be slightly higher if a crystal oscillator used.

Idle ICC is measured with all output pins disconnected ; XTAL1 driven with TCLCH, TCHCL = 5 ns, VIL = VSS + .5 V, VIH = VCC – .5 V ; XTAL2 N.C ; Port 0 = VCC ; EA = RST = VSS.

Power Down ICC is measured with all output pins disconnected ; EA = PORT 0 = VCC ; XTAL2 N.C. ; RST = VSS.

Note 3 : Capacitance loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the VOLS of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1 to 0 transitions during bus operations. In the worst cases (capacitive loading 100 pF), the noise pulse on the ALE line may exceed 0.45 V may exceed 0,45 V with maxi VOL peak 0.6 V. A Schmitt Trigger use is not necessary.

EXTERNAL CLOCK DRIVE CHARACTERISTICS (XTAL 1)

SYMBOL	PARAMETER	VARIABLE CLOCK FREQ = 0 to 16 MHz		UNIT
		MIN	MAX	
1/TCLCL	Oscillator Frequency	50		ns
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

A.C. CHARACTERISTICS

TA = – 40°C to 85°C ; VSS = 0 V ; VCC = 5 V ± 10 %

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

SYMBOL	PARAMETER	MIN	MAX	UNIT
TLHLL	ALE Pulse Width	2TCLCL-40		ns
TAVLL	Address Valid to ALE	TCLCL-55		ns
TLLAX	Address Hold After ALE	TCLCL-35		ns
TLLIV	ALE to Valid Instr in		4TCLCL-100	ns
TLLPL	ALE to $\overline{PSEN}$	TCLCL-40		ns
TPLPH	$\overline{PSEN}$ Pulse Width	3TCLCL-45		ns
TPLIV	$\overline{PSEN}$ to Valid Instr in		3TCLCL-105	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		TCLCL-25	ns
TPXAV	$\overline{PSEN}$ to Address Valid	TCLCL-8		ns
TAVIV	Address to Valid Instr in		5TCLCL-105	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		10	ns

EXTERNAL DATA MEMORY CHARACTERISTICS

SYMBOL	PARAMETER	MIN	MAX	UNIT
TRLRH	$\overline{RD}$ Pulse Width	6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	6TCLCL-100		ns
TLLAX	Data Address Hold After ALE	TCLCL-50		ns
TRLDV	$\overline{RD}$ to Valid Data in		5TCLCL-165	ns
TRHDX	Data Hold After $\overline{RD}$	0		ns
TRHDZ	Data Float After $\overline{RD}$		2TCLCL-70	ns
TLLDV	ALE to Valid Data in		8TCLCL-150	ns
TAVDV	Address to Valid Data in		9TCLCL-165	ns
TLLWL	ALE to $\overline{WR}$ or $\overline{RD}$	3TCLCL-50	3TCLCL+50	ns
TAVWL	Address to $\overline{WR}$ or $\overline{RD}$	4TCLCL-130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	TCLCL-60		ns
TQVWH	Data Setup to $\overline{WR}$ High	7TCLCL-150		ns
TWHQX	Data Hold After $\overline{WR}$	TCLCL-50		ns
TRLAZ	RD Low to Address Float		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	TCLCL-40	TCLCL+40	ns

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias :
A = Automotive ..... – 40°C to +125°C
M = Military ..... – 55°C to + 125°C
Storage Temperature ..... – 65°C to + 150°C
Voltage on Any Pin to V_{SS} ..... – 0.5 V to V_{CC} + 0.5 V
Voltage on V_{CC} to V_{SS} ..... – 0.5 V to 6.5 V
Power Dissipation ..... 1 W

*** NOTICE :**

Stresses above those listed under " Absolute Maximum Ratings " may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC CHARACTERISTICS

TA = – 55°C to + 125°C ; VSS = 0 V ; VCC = 5 V ± 10 % ; F = 0 to 12 MHz

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITIONS
VIL	Input Low Voltage	– 0.5	0.2 VCC – 0.1	V	
VIH	Input High Voltage (Except XTAL1, RST)	0.2 VCC + 0.9	VCC + 0.5	V	
VIH1	Input High Voltage (XTAL1, RST)	0.7 VCC	VCC + 0.5	V	
VOL	Output Low Voltage (Ports 1, 2, 3)		0.45	V	IOL = 1.6 mA (note 2)
VOL1	Output Low Voltage (Port 0, ALE, $\overline{PSEN}$)		0.45	V	IOL = 3.2 mA (note 2)
VOH	Output High Voltage (Ports 1, 2, 3)	2.4		V	IOH = – 60 µA VCC = 5 V ± 10 %
		0.75 VCC		V	IOH = – 25 µA
		0.9 VCC		V	IOH = – 10 µA
VOH1	Output High Voltage (Port 0 in External Bus Mode, ALE, PSEN)	2.4		V	IOH = – 800 µA VCC = 5 V ± 10 %
		0.75 VCC		V	IOH = – 300 µA
		0.9 VCC		V	IOH = – 80 µA
IIL	Logical 0 Input Current Ports 1, 2, 3		– 75	µA	Vin = 0.45 V
ITL	Logical 1 to 0 Transition Current		– 750	µA	Vin = 2 V
ILI	Input Leakage Current (Port 0, $\overline{EA}$)		± 10	µA	0.45 < Vin < VCC
RRST	Reset Pulldown Resistor	50	150	kΩ	
CIO	Pin Capacitance		10	pF	Test Freq = 1 MHz, T_A = 25°C
IPD	Power Down Current		75	µA	VCC = 2 to 5.5 V (note 1)
ICC	Power supply current Active mode 12 MHz Idle mode 12 MHz		 25 10	 mA mA	 VCC = 5.5 V VCC = 5.5 V

Note 1 : ICC is measured with all output pins disconnected ; XTAL1 driven with TCLCH, TCHCL = 5 ns, VIL = VSS + .5 V, VIH = VCC – .5 V ; XTAL2 N.C. ; EA = RST = Port 0 = VCC. ICC would be slightly higher if a crystal oscillator used.

Idle ICC is measured with all output pins disconnected ; XTAL1 driven with TCLCH, TCHCL = 5 ns, VIL = VSS + .5 V, VIH = VCC – .5 V ; XTAL2 N.C ; Port 0 = VCC ; EA = RST = VSS.

Power Down ICC is measured with all output pins disconnected ; EA = PORT 0 = VCC ; XTAL2 N.C. ; RST = VSS.

Note 2 : Capacitance loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the VOLS of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1 to 0 transitions during bus operations. In the worst cases (capacitive loading 100 pF), the noise pulse on the ALE line may exceed 0.45 V may exceed 0,45 V with maxi VOL peak 0.6 V. A Schmitt Trigger use is not necessary.

AC PARAMETERS :

TA = – 55°C to + 125°C ; VSS = 0 V ; VCC = 5 V ± 10 %
(Load Capacitance for Port 0, ALE, and PSEN = 100 pf ; Load Capacitance for All Other Outputs = 80 pf).

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

FREQ = 12 MHz (MAX)

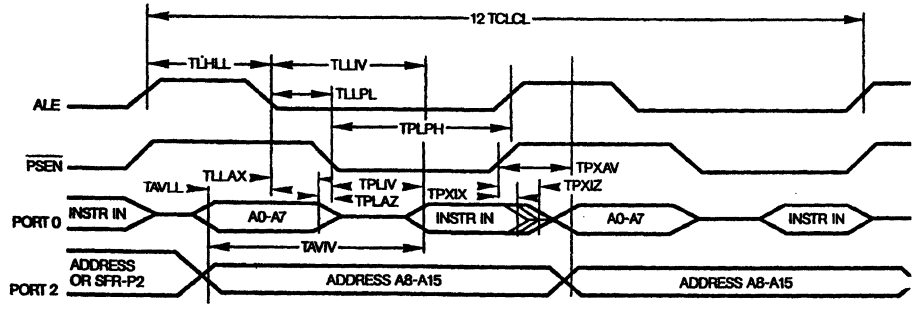
SYMBOL	PARAMETER	MIN	MAX	UNIT
TLHLL	ALE Pulse Width	2TCLCL-55		ns
TAVLL	Address Valid to ALE	TCLCL-70		ns
TLLAX	Address Hold After ALE	TCLCL-35		ns
TLLIV	ALE to Valid Instr in		4TCLCL-115	ns
TLLPL	ALE to $\overline{PSEN}$	TCLCL-55		ns
TPLPH	$\overline{PSEN}$ Pulse Width	3TCLCL-60		ns
TPLIV	$\overline{PSEN}$ to Valid Instr in		3TCLCL-120	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		TCLCL-40	ns
TPXAV	$\overline{PSEN}$ to Address Valid	TCLCL-8		ns
TAVIV	Address to Valid Instr in		5TCLCL-120	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		25	ns

EXTERNAL DATA MEMORY CHARACTERISTICS

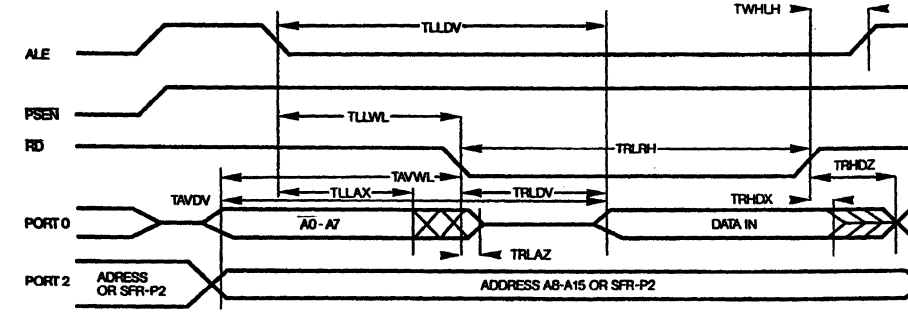
SYMBOL	PARAMETER	MIN	MAX	UNIT
TRLRH	$\overline{RD}$ Pulse Width	6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	6TCLCL-100		ns
TLLAX	Data Address Hold After ALE	TCLCL-50		ns
TRLDV	$\overline{RD}$ to Valid Data in		5TCLCL-185	ns
TRHDX	Data Hold After $\overline{RD}$	0		ns
TRHDZ	Data Float After $\overline{RD}$		2TCLCL-85	ns
TLLDV	ALE to Valid in		8TCLCL-170	ns
TAVDV	Address to Valid Data in		9TCLCL-185	ns
TLLWL	ALE to $\overline{WR}$ or $\overline{RD}$	3TCLCL-65	3TCLCL+65	ns
TAVWL	Address to $\overline{WR}$ or $\overline{RD}$	4TCLCL-145		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	TCLCL-75		ns
TQVWH	Data Setup to $\overline{WR}$ High	7TCLCL-150		ns
TWHQX	Data Hold After $\overline{WR}$	TCLCL-65		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	TCLCL-65	TCLCL+65	ns

AC TIMING DIAGRAMS

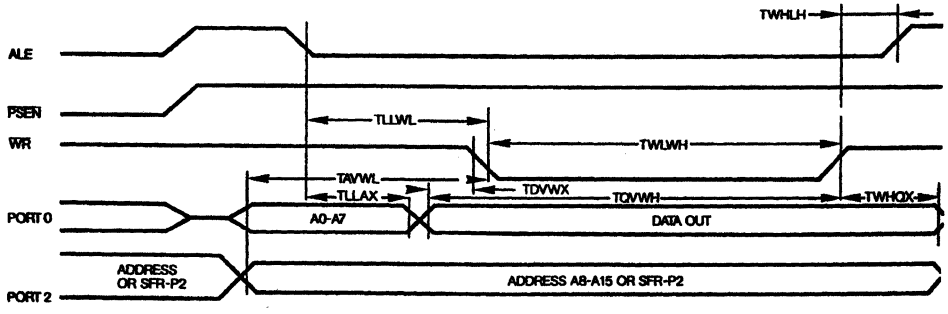
EXTERNAL PROGRAM MEMORY READ CYCLE



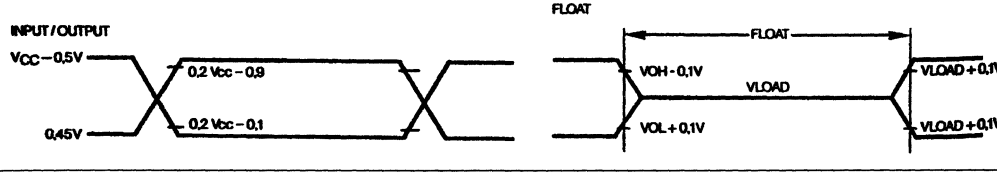
EXTERNAL DATA MEMORY READ CYCLE



EXTERNAL DATA MEMORY WRITE CYCLE



AC TESTING INPUT/OUTPUT, FLOAT WAVEFORMS



AC inputs during testing are driven at V_{CC} - 0.5 for a logic " 1 " and 0.45 V for a logic " 0 ". Timing measurements are made at VIH min for a logic " 1 " and VIL max for a logic " 0 ". For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change

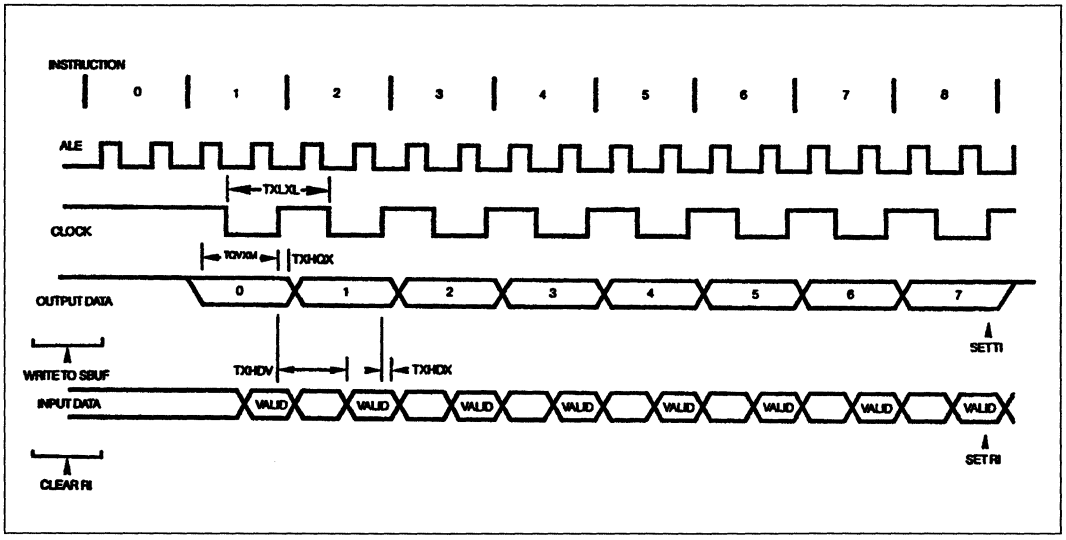
SERIAL PORT TIMING - SHIFT REGISTER MODE

A.C. CHARACTERISTICS

TA = – 40°C to 85°C ; VSS = 0 V ; VCC = 5 V ± 10 %

SYMBOL	PARAMETER	MIN	MAX	UNIT
TXLXL	Serial Port Clock Time	12TCLCL		µs
TQVXH	Output Data Setup to Clock Rising Edge	10TCLCL-133		ns
TXHQX	Output Data Hold After Clock Rising Edge	2TCLCL-117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		ns
TXHDV	Clock Rising Edge to Input Data Valid		10TLCL-133	ns

SHIFT REGISTER TIMING WAVEFORMS



6

EXPLANATION OF THE AC SYMBOLS

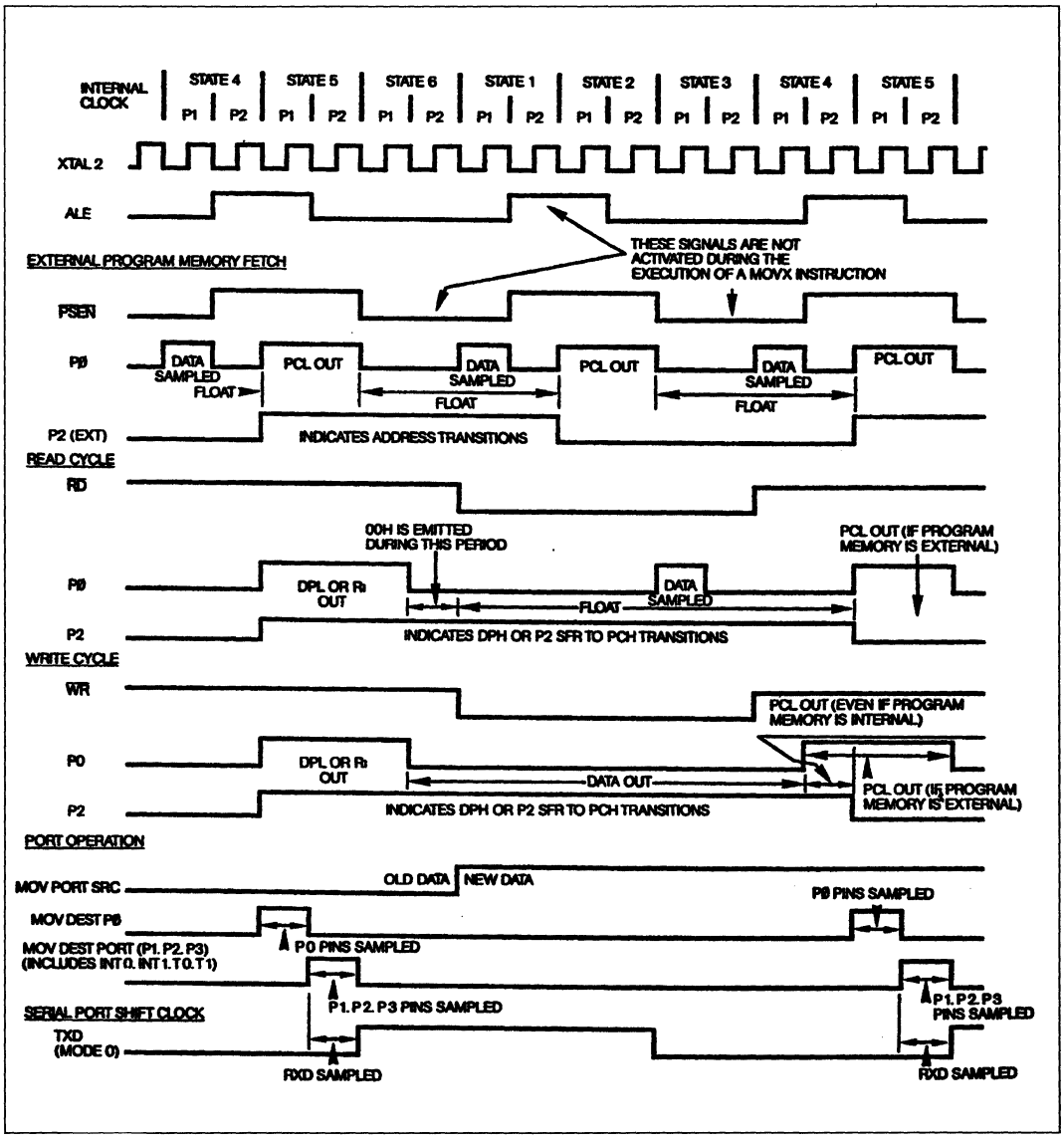
Each timing symbol has 5 characters. The first character is always a " T " (stands for time) The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

Example :

TAVLL = Time for Address Valid to ALE low.
TLLPL = Time for ALE low to PSEN low.

A : Address.
C : Clock.
D : Input data.
H : Logic level HIGH.
I : Instruction (program memory contents).
L : Logic level LOW, or ALE.
P : PSEN.
Q : Output data.
R : READ signal.
T : Time.
V : Valid.
W : WRITE signal.
X : No longer a valid logic level.
Z : Float.

CLOCK WAVEFORMS



This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins, however, ranges from 25 to 125 ns. This propagation delay is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component. Typically though (T_A = 25°C fully loaded) RD and WR propagation delays are approximately 50 ns. The other signals are typically 85 ns. Propagation delays are incorporated in the AC specifications.

ARITHMETIC OPERATIONS				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
ADD	A, Rn	Add register to Accumulator	1	1
ADD	A, direct	Add direct byte to Accumulator	2	1
ADD	A, @Ri	Add indirect RAM to Accumulator	1	1
ADD	A, #data	Add immediate data to Accumulator	2	1
ADDC	A, Rn	Add register to Accumulator with Carry	1	1
ADDC	A, direct	Add direct byte to A with Carry flag	2	1
ADDC	A, @Ri	Add indirect RAM to A with Carry flag	1	1
ADDC	A, #data	Add immediate data to A with Carry flag	2	1
SUBB	A, Rn	Subtract register from A with Borrow	1	1
SUBB	A, direct	Subtract direct byte from A with Borrow	2	1
SUBB	A, @Ri	Subtract indirect RAM from A with Borrow	1	1
SUBB	A, #data	Subtract immed. data from A with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Incriment indirect RAM	1	1
INC	DPTR	Incriment Data Pointer	1	2
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1
LOGICAL OPERATIONS				
MNEMONIC		**DESTINATION**	**BYTE**	**CYC**
ANL	A, Rn	AND register to Accumulator	1	1
ANL	A, direct	AND direct byte to Accumulator	2	1
ANL	A, @Ri	AND indirect RAM to Accumulator	1	1
ANL	A, #data	AND immediate data to Accumulator	2	1
ANL	direct, A	AND Accumulator to direct byte	2	1
ANL	direct, #data	AND immediate data to direct byte	3	2
ORL	A, Rn	OR register to Accumulator	1	1
ORL	A, direct	OR direct byte to Accumulator	2	1
ORL	A, @Ri	OR indirect RAM to Accumulator	1	1
ORL	A, #data	OR immediate data to Accumulator	2	1
ORL	direct, A	OR Accumulator to direct byte	2	1
ORL	direct, #data	OR immediate data to direct byte	3	2
XRL	A, Rn	Exclusive-OR register to Accumulator	1	1
XRL	A, direct	Exclusive-OR direct byte to Accumulator	2	1
XRL	A, @Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A, #data	Exclusive-OR immediate data to A	2	1
XRL	direct, A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct, #data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate A Left through the Carry flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate A Right through Carry flag	1	1
SWAP	A	Swap nibbles within the Accumulator	1	1

Table 1 : MHS - 51 Instruction Set Description.

DATA TRANSFER				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
MOV	A, Rn	Move register to Accumulator	1	1
MOV	A, direct	Move direct byte to Accumulator	2	1
MOV	A, @Ri	Move indirect RAM to Accumulator	1	1
MOV	A, #data	Move immediate data to Accumulator	2	1
MOV	Rn, A	Move Accumulator to register	1	1
MOV	Rn, direct	Move direct byte to register	2	2
MOV	Rn, #data	Move immediate data to register	2	1
MOV	direct, A	Move Accumulator to direct byte	2	1
MOV	direct, Rn	Move register to direct byte	2	2
MOV	direct, direct	Move direct byte to direct	3	2
MOV	direct, @Ri	Move indirect RAM to direct byte	2	2
MOV	direct, #data	Move immediate data to direct byte	3	2
MOV	@Ri, A	Move Accumulator to indirect RAM	1	1
MOV	@Ri, direct	Move direct byte to indirect RAM	2	2
MOV	@Ri, #data	Move immediate data to indirect RAM	2	1
MOV	DPTR, #data 16	Load Data Pointer with a 16-bit constant	3	2
MOVC	A, @A + DPTR	Move Code byte relative to DPTR to A	1	2
MOVC	A, @A + PC	Move Code byte relative to PC to A	1	2
MOVX	A, @Ri	Move External RAM (8-bit addr) to A	1	2
MOVX	A, @DPTR	Move External RAM (16-bit addr) to A	1	2
MOVX	@Ri, A	Move A to External RAM (8-bit addr)	1	2
MOVX	@DPTR, A	Move A to External RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A, Rn	Exchange register with Accumulator	1	1
XCH	A, direct	Exchange direct byte with Accumulator	2	1
XCH	A, @Ri	Exchange indirect RAM with A	1	1
XCHD	A, @Ri	Exchange low-order nibble ind RAM with A	1	1
BOOLEAN VARIABLE MANIPULATION				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
CLR	C	Clear Carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry flag	1	1
SETB	bit	Set direct Bit	2	1
CPL	C	Complement Carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry flag	2	2
ANL	C,/bit	AND complement of direct bit to Carry	2	2
ORL	C,bit	OR direct bit to Carry flag	2	2
ORL	C,/bit	OR complement of direct bit to Carry	2	2
MOV	C,bit	Move direct bit to Carry flag	2	1
MOV	bit, C	Move Carry flag to direct bit	2	2
PROGRAM AND MACHINE CONTROL				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
ACALL	addr 11	Absolute Subroutine Call	2	2
LCALL	addr 16	Long Subroutine Call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr 11	Absolute Jump	2	2
LJMP	addr 16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A + DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is Zero	2	2
JNZ	rel	Jump if Accumulator is Not Zero	2	2
JC	rel	Jump if Carry flag is set	2	2
JNC	rel	Jump if No Carry flag	2	2

Table 1. (Cont.)

PROGRAM AND MACHINE CONTROL (cont.)				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
JB	bit, rel	Jump if direct Bit set	3	2
JNB	bit, rel	Jump if direct Bit Not set	3	2
JBC	bit, rel	Jump if direct Bit is set & Clear bit	3	2
CJNE	A, direct, rel	Compare direct to A & Jump if Not Equal	3	2
CJNE	A, #data, rel	Comp. immed. to A & Jump if Not Equal	3	2
CJNE	Rn, #data, rel	Comp. immed. to reg & Jump if Not Equal	3	2
CJNE	@Ri, #data. rel	Comp. immed. to ind. & Jump if Not Equal	3	2
DJNZ	Rn, rel	Decrement register & Jump if Not Zero	2	2
DJNZ	direct. rel	Decrement direct & Jump if Not Zero	3	2
NOP		No operation	1	1

Table 1. (Cont.)

Notes on data addressing modes :

Rn – Working register R0-R7
direct – 128 internal RAM locations, any I/O port, control or status register
@Ri – Indirect internal RAM location addressed by register R0 or R1
#data – 8-bit constant included in instruction
#data 16 – 16-bit constant included as bytes 2 & 3 of instruction
bit – 128 software flags, any I/O pin, control or status bit

Notes on program addressing modes :

addr 16 – Destination address for LCALL & LJMP may be anywhere within the 64-k program memory address space
Addr 11 – Destination address for ACALL & AJMP will be within the same 2-k page of program memory as the first byte of the following instruction
rel – SJMP and all conditional jumps include an 8-bit offset byte. Range is + 127 – 128 bytes relative to the first byte of the following instruction.

6

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr, code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr, code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A, data
25	2	ADD	A, data addr
26	1	ADD	A, @R0
27	1	ADD	A, @R1
28	1	ADD	A, R0
29	1	ADD	A, R1
2A	1	ADD	A, R2
2B	1	ADD	A, R3
2C	1	ADD	A, R4
2D	1	ADD	A, R5
2E	1	ADD	A, R6
2F	1	ADD	A, R7
30	3	JNB	bit addr, code addr
31	2	ACALL	code addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A, #data
35	2	ADDC	A, data addr
36	1	ADDC	A, @R0
37	1	ADDC	A, @R1
38	1	ADDC	A, R0
39	1	ADDC	A, R1
3A	1	ADDC	A, R2
3B	1	ADDC	A, R3
3C	1	ADDC	A, R4
3D	1	ADDC	A, R5
3E	1	ADDC	A, R6
3F	1	ADDC	A, R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr, A
43	3	ORL	data addr, #data
44	2	ORL	A, #data
45	2	ORL	A, data addr
46	1	ORL	A, @R0
47	1	ORL	A, @R1
48	1	ORL	A, R0
49	1	ORL	A, R1
4A	1	ORL	A, R2
4B	1	ORL	A, R3
4C	1	ORL	A, R4
4D	1	ORL	A, R5
4E	1	ORL	A, R6
4F	1	ORL	A, R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr, A
53	3	ANL	data addr, #data
54	2	ANL	A, #data
55	2	ANL	A, data addr
56	1	ANL	A, @R0
57	1	ANL	A, @R1
58	1	ANL	A, R0
59	1	ANL	A, R1
5A	1	ANL	A, R2
5B	1	ANL	A, R3
5C	1	ANL	A, R4
5D	1	ANL	A, R5
5E	1	ANL	A, R6
5F	1	ANL	A, R7
60	2	JZ	code addr
61	2	AJMP	code addr
62	2	XRL	data addr A
63	3	XRL	data addr, #data
64	2	XRL	A, #data
65	2	XRL	A, data addr

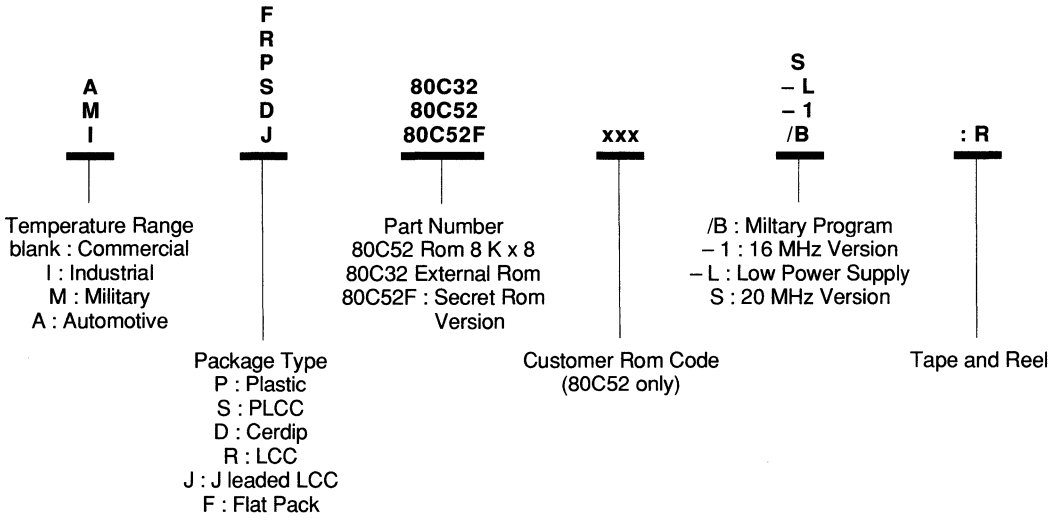
Table 2 : Instruction Opcodes in Hexadecimal Order.

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
66	1	XRL	A, @R0
67	1	XRL	A, @R1
68	1	XRL	A, R0
69	1	XRL	A, R1
6A	1	XRL	A, R2
6B	1	XRL	A, R3
6C	1	XRL	A, R4
6D	1	XRL	A, R5
6E	1	XRL	A, R6
6F	1	XRL	A, R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C, bit addr
73	1	JMP	@A + DPTR
74	2	MOV	A, #data
75	3	MOV	data addr, #data
76	2	MOV	@R0, #data
77	2	MOV	@R1, #data
78	2	MOV	R0, #data
79	2	MOV	R1, #data
7A	2	MOV	R2, #data
7B	2	MOV	R3, #data
7C	2	MOV	R4, #data
7D	2	MOV	R5, #data
7E	2	MOV	R6, #data
7F	2	MOV	R7, #data
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C, bit addr
83	1	MOVC	A, @A + PC
84	1	DIV	AB
85	3	MOV	data addr, data addr
86	2	MOV	data addr, @R0
87	2	MOV	data addr, @R1
88	2	MOV	data addr, R0
89	2	MOV	data addr, R1
8A	2	MOV	data addr, R2
8B	2	MOV	data addr, R3
8C	2	MOV	data addr, R4
8D	2	MOV	data addr, R5
8E	2	MOV	data addr, R6
8F	2	MOV	data addr, R7
90	3	MOV	DPTR, #data
91	2	ACALL	code addr
92	2	MOV	bit addr, C
93	1	MOVC	A, @A + DPTR
94	2	SUBB	A, #data
95	2	SUBB	A, data addr
96	1	SUBB	A, @R0
97	1	SUBB	A, @R1
98	1	SUBB	A,R0
99	1	SUBB	A, R1
9A	1	SUBB	A, R2
9B	1	SUBB	A, R3
9C	1	SUBB	A, R4
9D	1	SUBB	A, R5
9E	1	SUBB	A, R6
9F	1	SUBB	A, R7
A0	2	ORL	C, bit addr
A1	2	AJMP	code addr
A2	2	MOV	C, bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		reserved	
A6	2	MOV	@R0, data addr
A7	2	MOV	@R1, data addr
A8	2	MOV	R0, data addr
A9	2	MOV	R1, data addr
AA	2	MOV	R2, data addr
AB	2	MOV	R3, data addr
AC	2	MOV	R4, data addr
AD	2	MOV	R5, data addr
AE	2	MOV	R6, data addr
AF	2	MOV	R7, data addr
B0	2	ANL	C, bit addr
B1	2	ACALL	code addr
B2	2	CPL	Bit addr
B3	1	CPL	C
B4	3	CJNE	A, #data, code addr
B5	3	CJNE	A, data addr, code addr
B6	3	CJNE	@R0, #data, code addr
B7	3	CJNE	@R1, #data, code addr
B8	3	CJNE	R0, #data, code addr
B9	3	CJNE	R1, #data, code addr
BA	3	CJNE	R2, #data, code addr
BB	3	CJNE	R3, #data, code addr
BC	3	CJNE	R4, #data, code addr
BD	3	CJNE	R5, #data, code addr
BE	3	CJNE	R6, #data, code addr
BF	3	CJNE	R7, #data, code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A, data addr
C6	1	XCH	A, @R0
C7	1	XCH	A, @R1
C8	1	XCH	A, R0
C9	1	XCH	A, R1
CA	1	XCH	A, R2
CB	1	XCH	A, R3

Table 2. (Cont.)

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
CC	1	XCH	A, R4
CD	1	XCH	A, R5
CE	1	XCH	A, R6
CF	1	XCH	A, R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr, code addr
D6	1	XCHD	A, @R0
D7	1	XCHD	A, @R1
D8	2	DJNZ	R0, code addr
D9	2	DJNZ	R1, code addr
DA	2	DJNZ	R2, code addr
DB	2	DJNZ	R3, code addr
DC	2	DJNZ	R4, code addr
DD	2	DJNZ	R5, code addr
DE	2	DJNZ	R6, code addr
DF	2	DJNZ	R7, code addr
E0	1	MOVX	A, @DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A, @R0
E3	1	MOVX	A, @R1
E4	1	CLR	A
E5	2	MOV	A, data addr
E6	1	MOV	A, @R0
E7	1	MOV	A, @R1
E8	1	MOV	A, R0
E9	1	MOV	A, R1
EA	1	MOV	A, R2
EB	1	MOV	A, R3
EC	1	MOV	A, R4
ED	1	MOV	A, R5
EE	1	MOV	A, R6
EF	1	MOV	A, R7
F0	1	MOVX	@DPTR, A
F1	2	ACALL	code addr
F2	1	MOVX	@R0, A
F3	1	MOVX	@R1, A
F4	1	CPL	A
F5	2	MOV	data addr, A
F6	1	MOV	@R0, A
F7	1	MOV	@R1, A
F8	1	MOV	R0, A
F9	1	MOV	R1, A
FA	1	MOV	R2, A
FB	1	MOV	R3, A
FC	1	MOV	R4, A
FD	1	MOV	R5, A
FE	1	MOV	R6, A
FF	1	MOV	R7, A

Table 2. (Cont.)



DATA SHEET

83C154/83C154D

CMOS SINGLE-CHIP 8 BIT MICROCONTROLLER

- 83C154 - CMOS SINGLE CHIP 8 BIT MICRO-CONTROLLER with factory mask programmable ROM
- 83C154D - 83C154 with DOUBLE ROM (under development)
- 80C154 : ROMLESS version of 83C154
- 80C154/83C154 : 0 to 12 MHz
- 80C154-1/83C154-1 : 0 to 16 MHz
- 80C154-L/83C154-L : V_{CC} = 2.7 V to 5.5 V (0 to 6 MHz)
- 83C154F : SECRET ROM VERSION

FEATURES

- 16 K x 8 BIT INTERNAL ROM (32 K x 8 for 83C154D)
- 256 x BIT RAM
- 32 PROGRAMMABLE I/O LINES (PROGRAM-MABLE IMPEDANCE)
- THREE 16-BIT TIMER/COUNTERS (INCLUDING WATCH DOG AND 32 BIT TIMER)
- 64 K PROGRAM MEMORY SPACE
- FULLY STATIC DESIGN
- POWER CONTROL MODES
- INTERRUPT PRIORITY CONTROL
- 0 TO 16 MHz
- BOOLEAN PROCESSOR
- 6 INTERRUPT SOURCES
- PROGRAMMABLE SERIAL PORT
- 64 K DATA MEMORY SPACE
- TEMPERATURE RANGE : Commercial, Industrial, Automotive and Military

DESCRIPTION

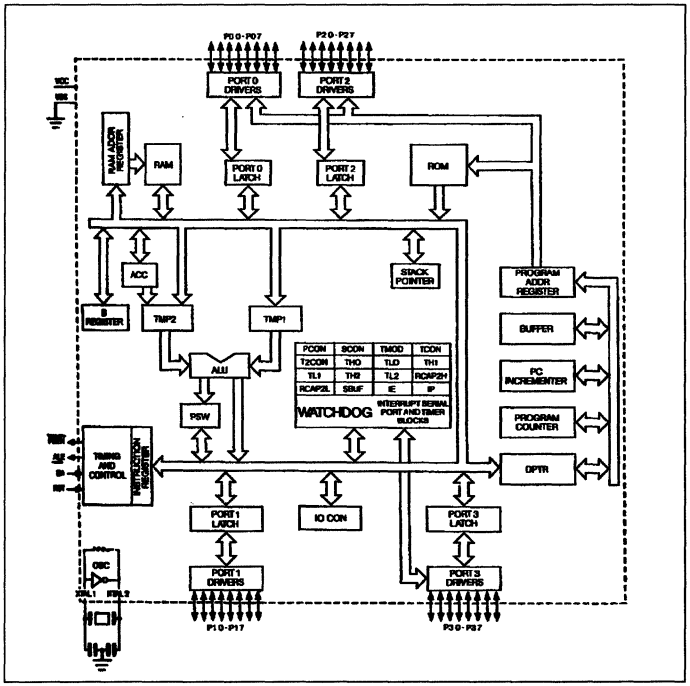


Figure 1 : Block Diagram.

The 83C154/83C154D retains all the features of the MHS 80C52 with extended ROM capacity (16 K bytes or 32 K bytes), 256 bytes of RAM, 32 I/O lines, a 6-source 2-level interrupts, a full duplex serial port, an on-chip oscillator and clock circuits, three 16 bit timers with extra features : 32 bit timer and watch dog functions. Timer 0 and 1 can be configured by program to implement a 32 bit timer. The watch dog function can be activated either with timer 0, or timer 1 or both together (32 bit timer). In addition, the 83C154/83C154D has two software selectable modes of reduced activity for further reduction of power consumption. In the Idle Mode, the CPU is frozen while the RAM is saved, and the timers, the serial port, and the interrupt system continue to function. In the Power Down Mode, the RAM is saved and the timers, serial port and interrupts continue to function when driven by external clocks. In addition as for the MHS 80C51/C52, the stop clock mode is also available.

6

MHS provides a new member in the 83C154/154D Family named "83C154F/C154DF" which permits full protection of the internal ROM contents.

With a non protected 83C154/C154D, it is very easy to read out the contents of the internal 16 K/32 K bytes of ROM.

Three methods exist, two of them are special test modes and the last one is by means of MOVC instructions.

- **Test mode "VER" :** Using this special test mode, the internal ROM contents are output on port P0 ; the address being applied on ports P2 (AD15...AD8) and P1 (AD7...AD0).
- **Test mode "TMB" :** With this second test mode, the contents of the 83C154/C154D internal bus is presented on port P1 during the PH2 clock phases.
- **Using MOVC instructions :** If EA = 0, and following a reset, the 83C154/C154D fetches its instructions from external program memory. It is then possible to write a small program whose purpose is to dump the internal ROM contents by means of MOVC A, @A + DPTR and MOVC A, @A + PC instructions.

83C154F/C154DF WITH PROGRAM PROTECTION FEATURES

This new version adds ROM protection features in some strategic points of the 83C154F/C154DF in order to eliminate the possibility of reading the ROM contents (once the protection has been programmed) by one if the three forementioned methods (VER and TMB test modes, or MOVC instructions).

Nevertheless the customer must note the following :

- Once the protection has been programmed, the 83C154F/C154DF program always starts at address 0 in the internal ROM.
- The application program must be self contained in the internal 16 K/32 K of ROM, otherwise it would be possible to trap the program counter address in the external PROM/EPROM and then to dump the internal ROM contents by means of a patch using MOVC instructions.

Thus, if an extra EPROM is necessary, it is advised to ensure that it will contain only constants or tables.

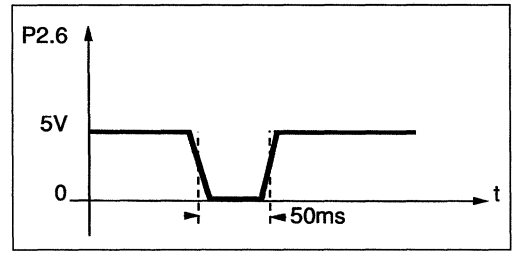
TEST OF THE ONE CHIP PROGRAM MEMORY

- **Before protection is activated :** The 83C154F/C154DF can be tested as any normal 83C154/C154D (using test equipment or any other methods).
- **After protection is activated :** It is then no longer possible to dump the internal ROM contents.

HOW TO PROGRAM THE PROTECTION MECHANISM

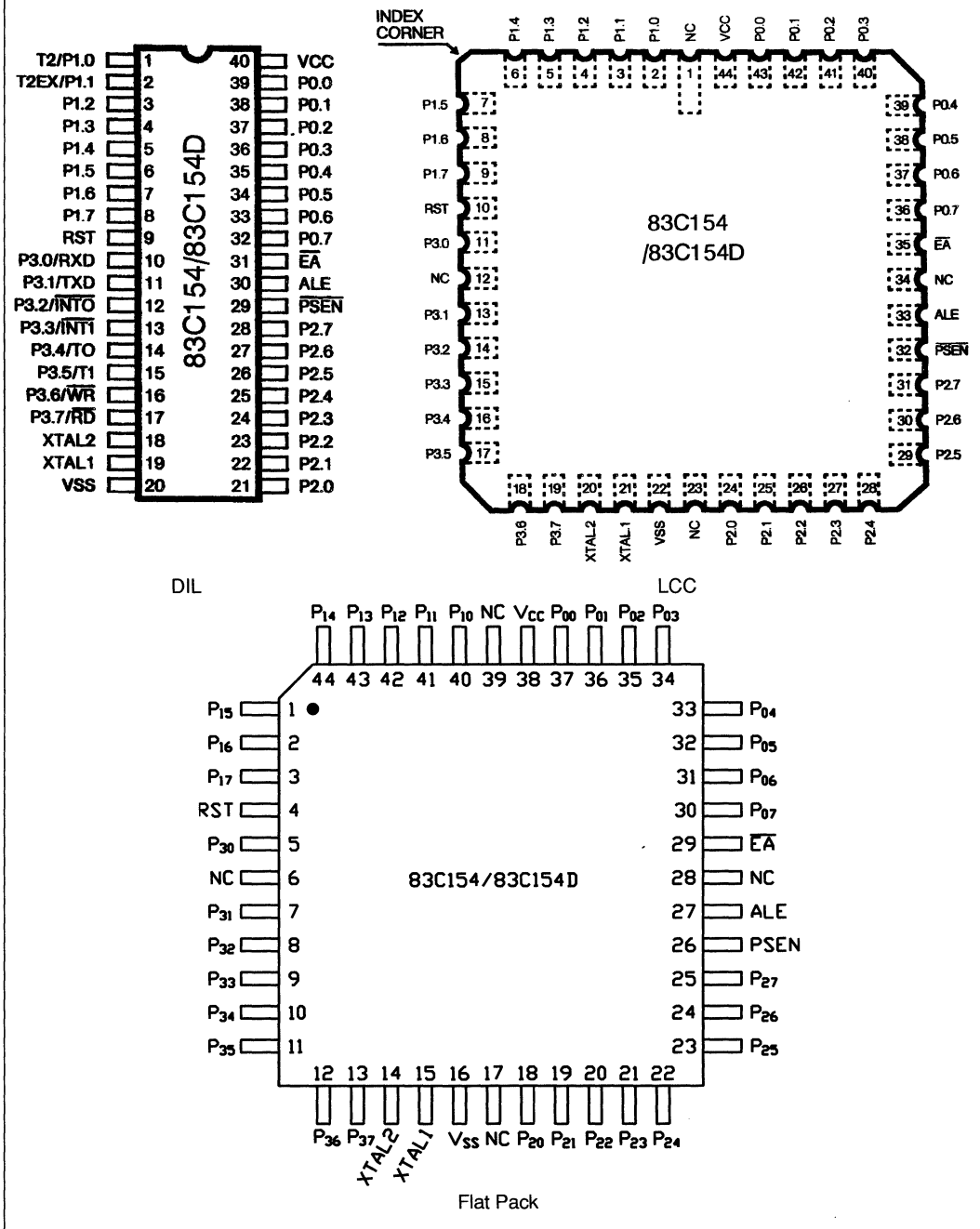
- To burn correctly the fuse a specific configuration of inputs must be settled as below :
  - RST = ALE = 1
  - P2.7 = 1

Furthermore PSEN signal must be tied at + 9 V ± 5 % level voltage and a pulse must be applied on P2.6 input Port. The timing on P2.6 is shown below :



Time Rise and Fall Rise ≤ 100 µs.

- The electrical schematic shows a typical application to deliver P2.6 signal.



Diagrams are for pin reference only. Package sizes are not to scale.

Figure 2 : Configurations.

IDLE AND POWER DOWN OPERATION

Figure 3 shows the internal Idle and Power Down clock configuration. As illustrated, Power Down operation stops the oscillator. The interrupt, serial port, and timer blocks continue to function only with external clock (INT0, INT1, T0, T1).

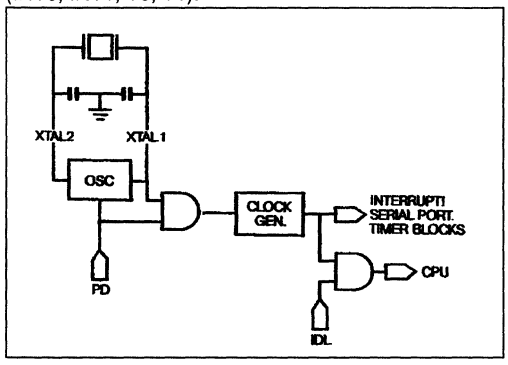


Figure 3 : Idle and Power Down Hardware.

Idle Mode operation allows the interrupt, serial port, and timer blocks to continue to function with internal or external clocks, while the clock to CPU is gated off. The special modes are activated by software via the Special Function Register, PCON. Its hardware address is 87H. PCON is not bit addressable.

PCON : Power Control Register

(MSB)							(LSB)
SMOD	HPD	RPD	-	GF1	GF0	PD	IDL

Symbol	Position	Name and Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1, 2 or 3.
HPD	PCON.	Hard power Down bit. Setting this bit allows CPU to enter in Power Down state on an external event (1 to 0 transition) on bit T1 (p. 3-5) the CPU quit the Hard Power Down mode when bit T1 (p. 3-5) go high or when reset is activated
RPD	PCON.5	Recover from Idle or Power Down bit. When 0 RPD has no effect. When 1, RPD permits to exit from idle or Power Down with any non enabled interrupt source (except timex 2). In this case the program start at the next address. When interrupt is enabled, the appropriate interrupt routine is serviced.
	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activates idle mode operation.

If 1's are written to PD and IDL at the same time. PD takes, precedence. The reset value of PCON is (000X0000).

IDLE MODE

The instruction that sets PCON.0 is the last instruction executed before the Idle mode is activated. Once in the Idle mode the CPU status is preserved in its entirety : the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM and all other registers maintain their data during idle. In the idle mode, the internal clock signal is gated off to the CPU, but interrupt, timer and serial port functions are maintained. Table 1 describes the status of the external pins during Idle mode.

There are three ways to terminate the Idle mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating Idle mode. The interrupt is serviced, and following RETI, the next instruction to be executed will be the one following the instruction that wrote 1 to PCON.0.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during the Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an enabled interrupt, the service routine can examine the status of the flag bits.

The second way of terminating the Idle mode is with a hardware reset. Since the oscillator is still running, the hardware reset needs to be active for only 2 machine cycles (24 oscillator periods) to complete the reset operation.

The third way to terminate the Idle mode is the activation of any disabled interrupt when recover is programmed (RPD = 1). This will cause PCON.0 to be cleared. No interrupt is serviced. The next instruction is executed. If interrupt are disabled and RPD = 0, only a

POWER DOWN MODE

The instruction that sets PCON.1 is the last executed prior to entering power down. Once in power down, the oscillator is stopped. The contents of the onchip RAM and the Special Function Register is saved during power down mode. The three ways to terminate the Power Down mode are the same than the Idle mode. But since the onchip oscillator is stopped, the external interrupts, timers and serial port must be sourced by external clocks only, via INT0, INT1, T0, T1.

In the Power Down mode, V_{CC} may be lowered to minimize circuit power consumption. Care must be taken

to ensure the voltage is not reduced until the power down mode is entered, and that the voltage is restored before the hardware reset is applied which frees the oscillator. Reset should not be released until the oscillator has restarted and stabilized.

When using voltage reduction : interrupt, timers and serial port functions are guaranteed in the V_{CC} specification limits.

Table 1 describes the status of the external pins while in the power down mode. It should be noted that if the power down mode is activated while in external program memory, the port data that is held in the Special Function Register P2 is restored to Port 2. If the port switches from 0 to 1, the port pin is held high during the power down mode by the strong pullup, T1, shown in figure 4.

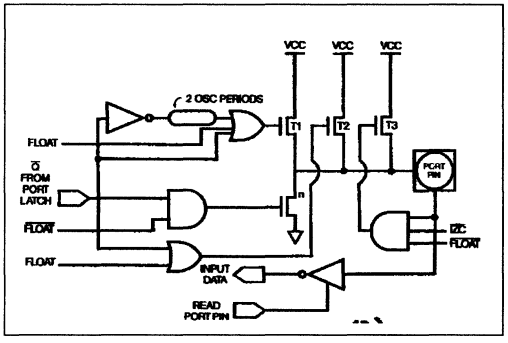


Figure 4 : I/O Buffers in the 83C154/83C154D (Ports 1, 2, 3).

STOP CLOCK MODE

Due to static design, the MHS 83C154/83C154D clock speed can be reduced until 0 MHz without any data loss in memory or registers. This mode allows step by step utilization, and permits to reduce system power consumption by bringing the clock frequency down to any value. At 0 MHz, the power consumption is the same as in the Power Down Mode.

83C154/83C154D I/O PORTS

The I/O drives for P1, P2, P3 of the 83C154/83C154D are impedance programmable. The I/O buffers for Ports 1, 2 and 3 are implemented as shown in *figure 4.*

When the port latch contains 0, all pFETS in figure 4 are off while the nFET is turned on. When the port latch makes a 0-to-1 transition, the nFET turns off. The strong pullup pFET, T1, turns on for two oscillator periods, pulling the output high very rapidly. As the output line is drawn high, pFET T3 turns on through the inverter to supply the I_{OH} source current. This inverter and T3 form a latch which holds the 1 and is supported by T2. When Port 2 is used as an address port, for access to external program of data memory, any address bit that contains a 1 will have his strong pullup turned on for the entire duration of the external memory access.

When an I/O pin on Ports 1, 2, or 3 is used as an input, the user should be aware that the external circuit must sink current during the logical 1-to-0 transition. The maximum sink current is specified as ITL under the D.C. Specifications. When the input goes below approximately 2 V, T3 turns off to save ICC current. Note, when returning to a logical 1, T2 is the only internal pullup that is on. This will result in a slow rise time if the user's circuit does not force the input line high.

The input impedance of Port 1, 2, 3 are programmable through the register IOCON. The ALF bit (IOCON0) set all of the Port 1, 2, 3 floating when a Power Down mode occurs. The P1HZ, P2HZ, P3HZ bits (IOCON1, IOCON2, IOCON3) set respectively the Ports P1, P2, P3 in floating state. The IZC (IOCON4) allows to choose input impedance of all ports (P1, P2, P3). When IZC = 0, T2 and T3 pullup of I/O ports are active ; the internal input impedance is approximately 10 K. When IZC = 1 only T2 pull-up is active. The T3 pull-up is turned off by IZC. The internal impedance is approximately 100 K.

PIN DESCRIPTIONS

Vss

Circuit ground potential.

Vcc

Supply voltage during normal, Idle, and Power Down operation.

PORT 0

Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

MODE	PROGRAM MEMORY	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Port Data	Port Data	Port Data	Port Data
Idle	External	1	1	Floating	Port Data	Address	Port Data
Power Down	Internal	0	0	Port Data	Port Data	Port Data	Port Data
Power Down	External	0	0	Floating	Port Data	Port Data	Port Data

Table 1 : Status of the external pins during Idle and Power Down Modes.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1's. Port 0 also outputs the code bytes during program verification in the 83C154/-83C154D. External pullups are required during program verification. Port 0 can sink eight LS TTL inputs.

PORT 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. Port 1 pins that have 1's written to them are pulled high by the internal pullups, an in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups.
Port 1 also receives the low-order address byte during program verification. In the 83C154, Port 1 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

PORT 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. Port 2 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (ILL, on the data sheet) because of the internal pullups. Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @ DPTR). In this application, it uses strong internal pullups when emitting 1's. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.
It also receives the high-order address bits and control signals during program verification in the 83C154. Port 2 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

PORT 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. Port 3 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (ILL, on the data sheet) because of the pullups. It also serves the functions of various special features of the MHS-51 Family, as listed below.

Port Pin	Alternate Function
P3.0	$\overline{RXD}$ (serial input port)
P3.1	$\overline{TXD}$ (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	$\overline{T1}$ (Timer 1 external input)
P3.6	$\overline{WR}$ (external Data Memory write strobe)
P3.7	RD (external Data Memory read strobe)

Port 3 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

RST

A high level on this for two machine cycles while the oscillator is running resets the device. An internal pulldown resistor permits Power-On reset using only a capacitor connected to V_{CC}.

ALE

Address Latch Enable output for latching the low byte of the address during accesses to external memory. ALE is activated as though for this purpose at a constant rate of 1/6 the oscillator frequency except during an external data memory access at which time one ALE pulse is skipped. ALE can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

$\overline{PSEN}$

Program Store Enable output is the read strobe to external Program Memory. PSEN is activated twice each machine cycle during fetches from external Program Memory. (However, when executing out of external Program Memory, two activations of PSEN are skipped during each access to external Data Memory). PSEN is not activated during fetches from internal Program Memory. PSEN can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

$\overline{EA}$

When EA is held high, the CPU executes out of internal Program Memory (unless the Program Counter exceeds 3 FFFH). When EA is held low, the CPU executes only out of external Program Memory. EA must not be floated.

XTAL1

Input to the inverting amplifier that forms the oscillator. Receives the external oscillator signal when an external oscillator is used.

XTAL2

Output of the inverting amplifier that forms the oscillator. This pin should be floated when an external oscillator is used.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output respectively, of an inverting amplifier which is configured for use as an on-chip oscillator, as shown in *figure 5.* Either a quartz crystal or ceramic resonator may be used.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected as shown in *figure 6.* There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-

by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

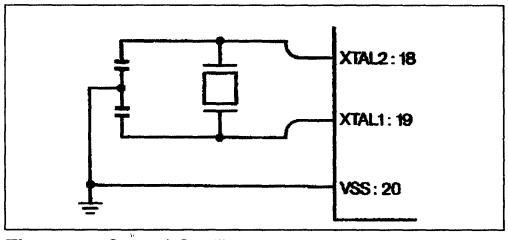


Figure 5 : Crystal Oscillator.

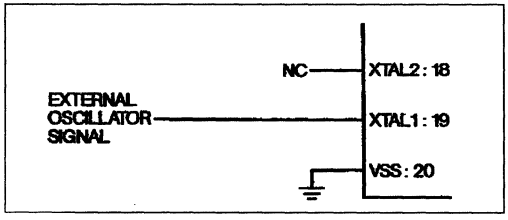


Figure 6 : External Drive Configuration.

PORT 1 SECONDARY FUNCTIONS

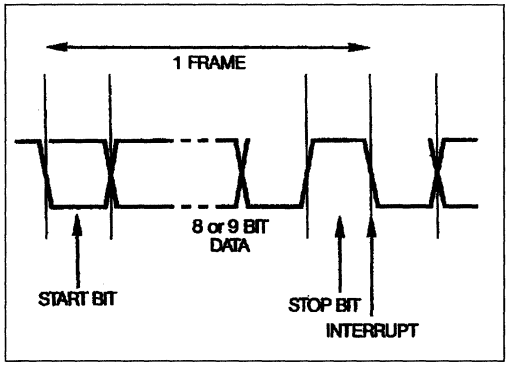
This is a quasi-bidirectional I/O port, internally pulled up when used as input ports. Two of the ports have been allocated a second function which are :
P1.0 [T2] : External clock input for timer/counter 2.
P1.1 [T2EX] : A trigger input for timer/counter 2, to be reloaded or captured causing the timer/counter 2 interrupt.

INTERRUPT MODES

The MHS 80C154/83C154/83C154D is capable of handling two external interrupts, three interrupts from the timers, and one interrupt from the serial port, through its incorporated six source, two-level interrupt structure.

SERIAL PORT TIMING

The interrupt is executed after the Stop Bit.



TIMER FUNCTIONS

In fact, timer 0 & 1 can be connected by a software instruction to implement a 32-bit timer function. Timer 0 (mode 3) or timer 1 (mode 0, 1, 2) or a 32-bit timer consisting of timer 0 + timer 1 can be employed in the watchdog mode, in which case a CPU reset is generated upon a TF1 flag.
The internal pull-up resistances at ports 1 ~ 3 can be set to a ten times increased value simply by software.

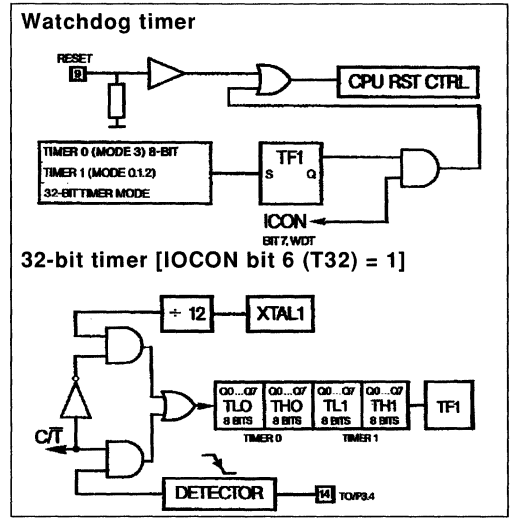


Figure 7.

TIMER/EVENT COUNTER 2

Timer 2 is a 16-bit timer/counter like Timers 0 and 1, it can operate either as a timer or as an event counter. This is selected by bit C/T2 in the Special Function Register T2CON *(Figure 7)*. It has three operating modes : "capture", "autoload", and "baud rate generator", which are selected by bits in T2CON as shown in

RCLK + TCLK	CP/RL2	TR2	MODE
0	0	1	16 bit auto-reload
0	1	1	16-bit capture
1	X	1	baud rate generator
X	X	0	(off)

Table 2 : Timer 2 Operating Modes.

Table 2.

In the capture mode there are two options which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then Timer 2 is a 16-bit timer or counter which upon overflowing sets bit TF2, the Timer 2 overflow bit, which can be used to generate an interrupt. If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX causes the current value in the Timer 2 registers, TL2 ans TH2, to be

captured into registers RCAP2L and RCAP2H, respectively. (RCAP2L and RCAP2H are new Special Function Registers in the 80C52). In addition, the transition at T2EX causes bit EXF2 in T2CON to be set, and EXF2, like TF2, can generate an interrupt.
The capture mode is illustrated in *Figure 8.*
In the auto-reload mode there are again two options, which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then when Timer 2 rolls over it not only sets TF2 but also causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX will also trigger the 16-bit reload and set EXF2.
The auto-reload mode is illustrated in *Figure 9.*

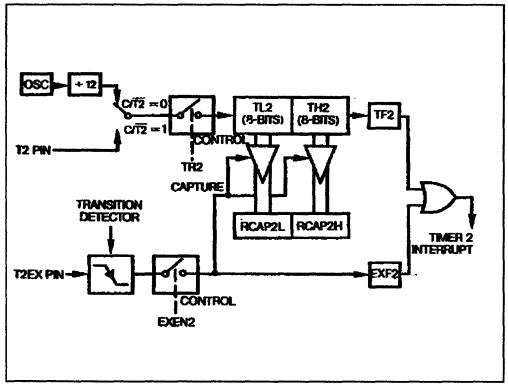


Figure 8 : Timer 2 in Capture Mode.

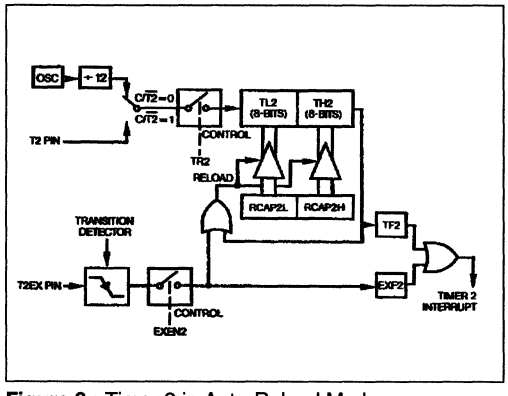


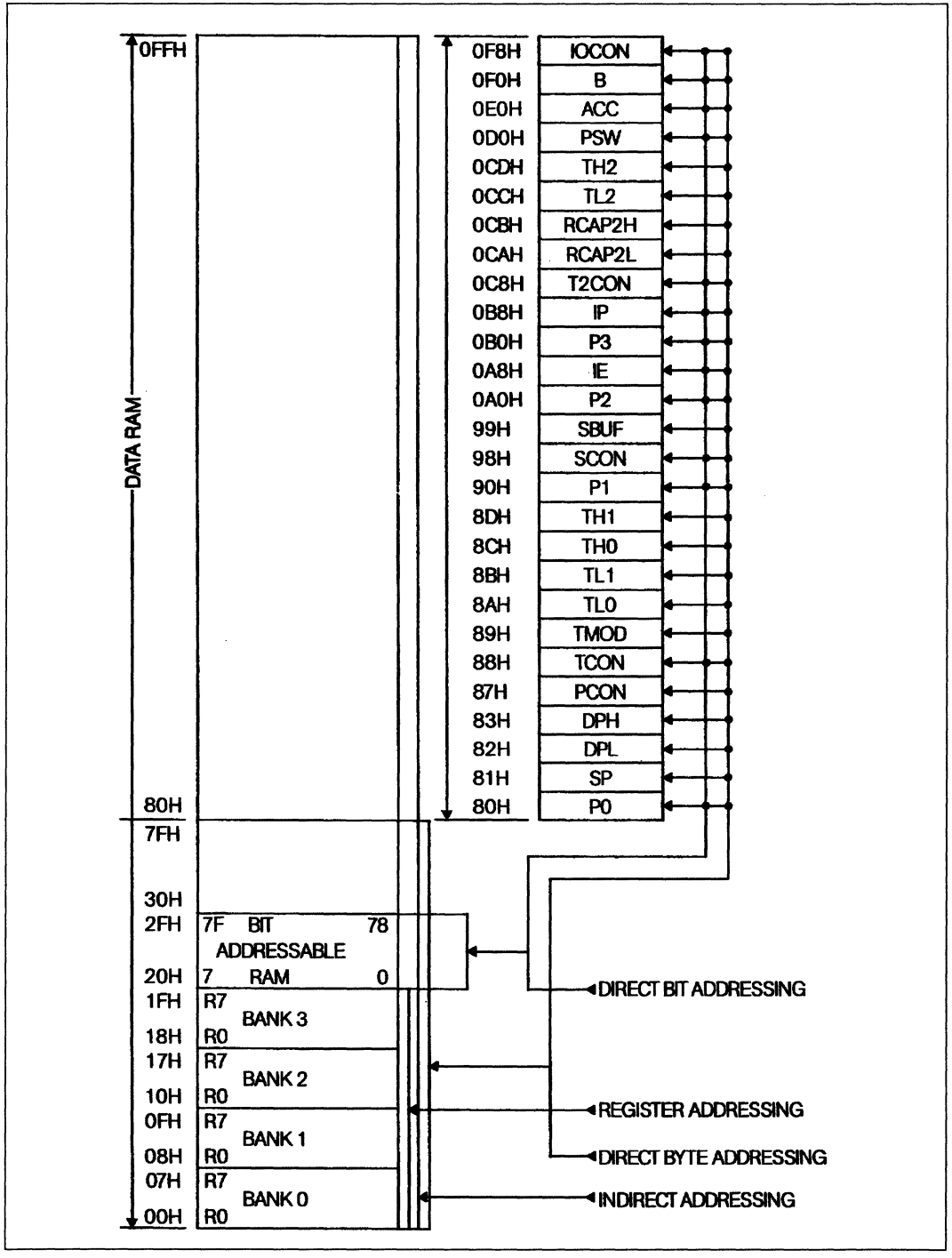
Figure 9 : Timer 2 in Auto-Reload Mode.

(MSB)							(LSB)
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2

Symbol	Position	Name and Significance
TF2	T2CON.7	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1
EXF2	T2CON.6	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software.
RCLK	T2CON.5	Receive clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	T2CON.4	Transmit clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	T2CON.3	Timer 2 external enable flag. When set, allows capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events as T2EX.
TR2	T2CON.2	Start/stop control for Timer 2. A logic 1 starts the timer.
C/$\overline{T2}$	T2CON.1	Timer or counter select. (Timer 2) 0 = Internal timer (OSC/12) 1 = External event counter (falling edge triggered).
CP/$\overline{RL2}$	T2CON.0	Capture/Reload flag. When set, captures will occur on negative transitions at T2EX if EXEN 2 = 1. When cleared, auto reloads will occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

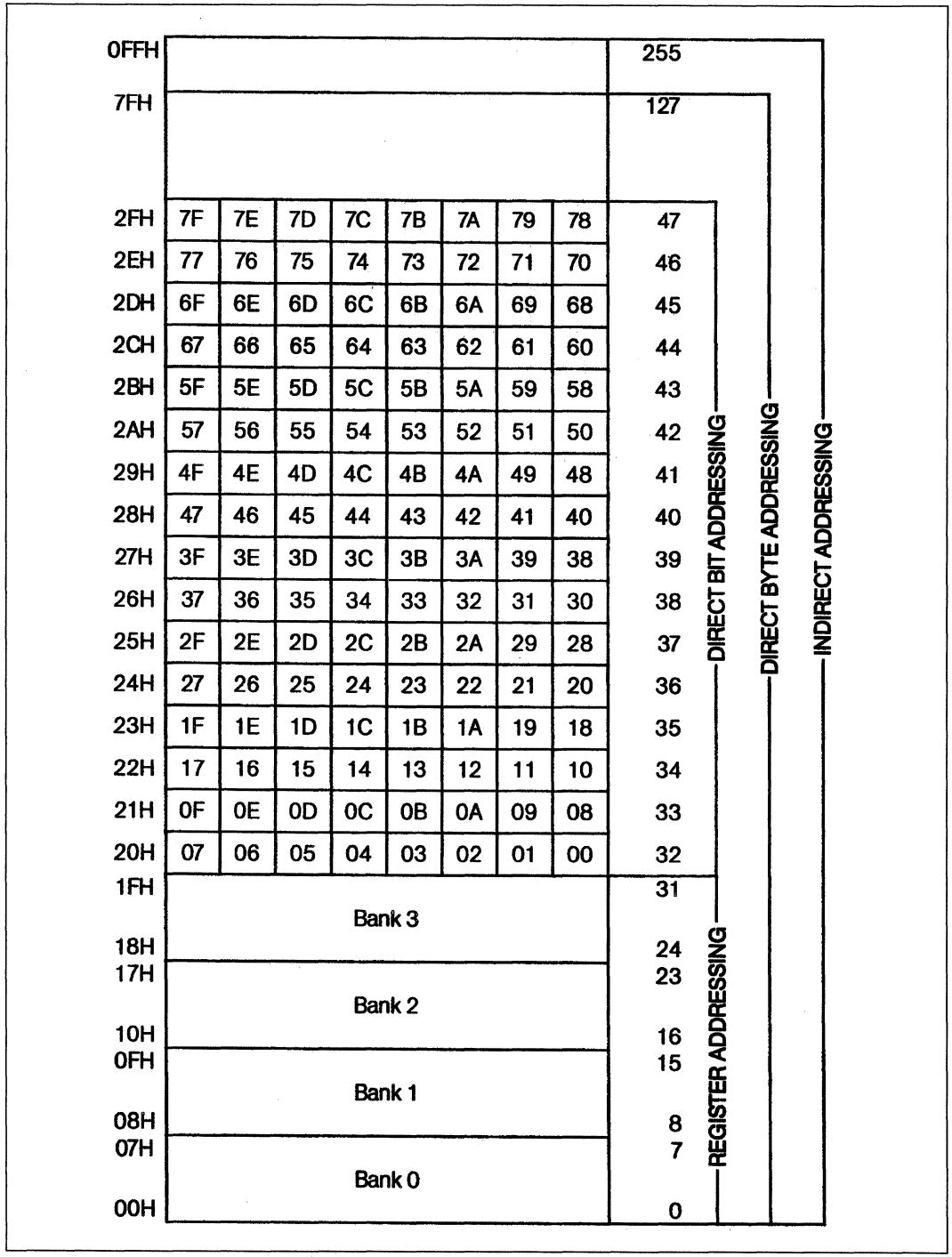
Figure 7 : T2CON :
Timer/Counter 2 Control Register.

DATA MEMORY AND SPECIAL FUNCTION REGISTER LAYOUT DIAGRAM



6

DETAILED DIAGRAM OF DATA MEMORY (RAM)



6

DETAILED DIAGRAM OF SPECIAL FUNCTION REGISTERS

Direct Byte Address	Bit Address								Special Function Register Symbol
	(MSB)							(LSB)	
	WDT	T32	SERR	IZC	P3HZ	P2HZ	P1HZ	ALF	
0F8H	FF	FE	FD	FC	FB	FA	F9		IOCON
0F0H	F7	F6	F5	F4	F3	F2	F1	F0	B
0E0H	E7	E6	E5	E4	E3	E2	E1	E0	ACC
	CY	AC	F0	RS1	RS0	0V	F1	P	
0D0H	D7	D6	D5	D4	D3	D2	D1	D0	PSW
0CDH	Not Bit Addressable								TH2
0CCH	Not Bit Addressable								TL2
0CBH	Not Bit Addressable								RCAP2H
0CAH	Not Bit Addressable								RCAP2L
	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$	
0C8H	CF	CE	CD	CC	CB	CA	C9	C8	T2CON
	PCT		PT2	PS	PT1	PX1	PT0	PX0	
0B8H	BF	-	BD	BC	BB	BA	B9	B8	IP
0B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3
	EA		ET2	ES	ET1	EX1	ET0	EX0	
0A8H	AF	-	AD	AC	AB	AA	A9	A8	IE
0A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2
99H	Not Bit Addressable								SBUF
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
98H	9F	9E	9D	9C	9B	9A	99	98	SCON
90H	97	96	95	94	93	92	91	90	P1
8DH	Not Bit Addressable								TH1
8CH	Not Bit Addressable								TH0
8BH	Not Bit Addressable								TL1
8AH	Not Bit Addressable								TL0
89H	Not Bit Addressable								TMOD
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
88H	8F	8E	8D	8C	8B	8A	89	88	TCON
87H	Not Bit Addressable								PCON
83H	Not Bit Addressable								DPH
82H	Not Bit Addressable								DPL
81H	Not Bit Addressable								SP
80H	87	86	85	84	83	82	81	80	P0

SPECIAL FUNCTION REGISTERS

TIME MODE REGISTER (TMOD)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
TMOD	89H	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

BIT LOCATION	FLAG	FUNCTION		
TMOD.0	M0	M1	M0	Timer/counter 0 mode setting.
		0	0	8-bit timer/counter with 5-bit prescalar.
		0	1	16-bit timer/counter.
		1	0	8-bit timer/counter with 8-bit auto reloading.
TMOD.1	M1	1	1	Timer/counter 0 separated into TL0 (8-bit) timer/counter and TH0 (8-bit) timer/counter. TF0 is set by TL0 carry, and TF1 is set by TH0 carry.
TMOD.2	C/$\overline{T}$	Timer/counter 0 count clock designation control bit. XTAL1.2 divided by 12 clocks is the input applied to timer/counter 0 when C/$\overline{T}$ = "0". The external clock applied to the T0 pin is the input applied to timer/counter 0 when C/$\overline{T}$ = "1".		
TMOD.3	GATE	When this bit is "0", the TR0 bit of TCON (timer control register) is used to control the start and stop of timer/counter 0 counting. If this bit is "1", timer/counter 0 starts counting when both the TR0 bit of TCON and $\overline{INT0}$ pin input signal are "1", and stops counting when either is changed to "0".		
TMOD.4	M0	M1	M0	Timer/counter 1 mode setting.
		0	0	8-bit timer/counter with 5-bit prescalar.
		0	1	16-bit timer/counter.
TMOD.5	M1	1	0	8-bit timer/counter with 8-bit auto reloading.
		1	1	Timer/counter 1 operation stopped.
TMOD.6	C/$\overline{T}$	Timer/counter 1 count clock designation control bit. XTAL.2 divided by 12 clocks is the input applied to timer/counter 1 when C/$\overline{T}$ = "0". The external clock applied to the T1 pin is the input applied to timer/counter 1 when C/$\overline{T}$ = "1".		
TMOD.7	GATE	When this bit is "0", the TR1 bit of TCON is used to control the start and stop of timer/counter 1 countig. If this bit is "1", timer/counter 1 starts counting when both the TR1 bit of TCON and $\overline{INT1}$ pin input signal are "1", and stops counting when either is changed to "0".		

6

POWER CONTROL REGISTER (PCON)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
PCON	87H	SMOD	HPD	RPD	–	GF1	GF0	PD	IDL

BIT LOCATION	FLAG	FUNCTION
PCON.0	IDL	IDLE mode set when this bit is set to "1". CPU operations are stopped when IDLE mode is set, but XTAL1-2, timer/counters 0, 1 and 2, the interrupt circuits, and serial port remain active. IDLE mode is cancelled when the CPU is reset or when an interrupt is generated.
PCON.1	PD	PD mode set when this bit is set to "1". CPU operations and XTAL1-2 are stopped when PD mode is set. PD mode is cancelled when the CPU is reset or when an interrupt is generated.
PCON.2	GF0	General purpose bit. Testing this flag when IDLE mode is cancelled by an interrupt shows whether the interrupt is a normal interrupt or an IDLE mode release interrupt.
PCON.3	GF1	General purpose bit. Testing this flag when PD mode is cancelled by an interrupt shows whether the interrupt is a normal interrupt or a PD mode release interrupt.
PCON.4	–	Reserved bit. The output data is "1" if the bit is read.
PCON.5	RPD	Bit used to specify cancellation of CPU power down mode (IDLE or PD) by interrupt signal. Power down mode cannot be cancelled by interrupt signal if interrupt is not enabled by IE (interrupt enable register) when this bit is "0". If the interrupt flag is set to "1" by an interrupt request signal when this bit is "1" (even if interrupt is disabled), the program is executed from the next address of the power down mode setting instruction. The flag is reset to "0" by software.
PCON.6	HPD	The hard power down setting mode is enabled when this bit is set to "1". If the level of the power failure detect signal applied to the HPD1 pin (pin 3.5) is changed from "1" to "0" when this bit is "1", XTAL1-2 oscillation is stopped and the system is put into hard power down mode. HPD mode is cancelled when the CPU is reset, or HPD1 pin go high.
PCON.7	SMOD	When the timer/counter 1 carry signal is used as a clock in mode 1, 2 or 3 of the serial port, this bit has the following functions. The serial port operation clock is reduced by 1/2 when the bit is "0" for delayed processing. And when the bit is "1", the serial port operation clock is normal for faster processing.

ENABLE	RECOVER	
0	0	PWD not cancelled
1	0	Execute interrupt routine
0	1	Execute next address
1	1	Execute interrupt routine

TIMER CONTROL REGISTER (TCON)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
TCON	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

BIT LOCATION	FLAG	FUNCTION
TCON.0	IT0	External interrupt 0 signal used in level detect mode when this bit is "0", and in trigger detect mode when "1".
TCON.1	IE0	Interrupt request flag for external interrupt 0. Bit is reset automatically when interrupt is serviced. Bit can be set an reset by software when IT0 = "1".
TCON.2	IT1	External interrupt 1 signal used in level detect mode when this bit is "0", and in trigger detect mode when "1".
TCON.3	IE1	Interrupt request flag for external interrupt 1. Bit is reset automatically when interrupt is serviced. Bit can be set and reset by software when IT1 = "1".
TCON.4	TR0	Counting start and stop control bit for timer/counter 0. Timer/counter 0 starts counting when this bit is "1", and stops counting when "0".
TCON.5	TF0	Interrupt request flag for timer interrupt 0. Bit is reset automatically when interrupt is serviced. Bit is set to "1" when carry signal is generated from timer/counter 0.
TCON.6	TR1	Counting start and stop control bit for timer/counter 1. Timer/counter 1 starts counting when this bit is "1", and stops counting when "0".
TCON.7	TF1	Interrupt request flag for timer interrupt 1. Bit is reset automatically when interrupt is serviced. Bit is set to "1" when carry signal is generated from timer/counter 1.

SERIAL PORT CONTROL REGISTER (SCON)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
SCON	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI

BIT LOCATION	FLAG	FUNCTION
SCON.0	RI	"End of serial port reception" interrupt request flag. This flag must be reset by software during interrupt service routine. This flag is set after the eighth bit of data has been received when in mode 0, or by the STOP bit when in any other mode. In mode 2 or 3, however RI is not set if the RB8 data is "0" with SM2 = "1". RI is set in mode 1 if STOP is received when SM2 = "1".
SCON.1	TI	"End of serial port transmission" interrupt request flag. This flag must be reset by software during interrupt service routine. This flag is set after the eighth bit of data has been sent when in mode 0, or after the last bit of data has been when in any other mode.
SCON.2	RB8	The ninth bit of data received in mode 2 or 3 is passed to RB8. The STOP bit is applied to RB8 if SM2 = "0" when in mode 1. RB8 can not be used in mode 0.
SCON.3	TB8	The TB8 data is sent as the ninth data bit when in mode 2 or 3. Any desired data can be set in TB8 by software.
SCON.4	REN	Reception enable control bit. No reception when REN = "0". Reception enabled when REN = "1".
SCON.5	SM2	If the ninth bit of received data is "0" with SM2 = "1" in mode 2 or 3, the "end of reception" signal is not set in the RI flag. Nor is the "end of reception" signal set in the RI flag if the STOP bit is not "1" when SM2 = "1" in mode 1.
SCON.6	SM1	SM0 = 0, SM1 = 0, MODE = 0: 8-bit shift register I/O. SM0 = 0, SM1 = 1, MODE = 1: 8-bit UART variable baud rate.
SCON.7	SM0	SM0 = 1, SM1 = 0, MODE = 2: 9-bit UART 1/32 XTAL1, 1/64 XTAL1 baud rate. SM0 = 1, SM1 = 1, MODE = 3: 9-bit UART variable baud rate.

6

INTERRUPT ENABLE REGISTER (IE)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
IE	0A8H	EA	–	ET2	ES	ET1	EX1	ET0	EX0

BIT LOCATION	FLAG	FUNCTION
IE.0	EXQ	Interrupt control bit for external interrupt 0. Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".
IE.1	ET0	Interrupt control bit for timer interrupt 0. Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".
IE.2	EX1	Interrupt control bit for external interrupt 1. Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".
IE.3	ET1	Interrupt control bit for timer interrupt 1. Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".
IE.4	ES	Interrupt control for serial port. Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".
IE.5	ET2	Interrupt control bit for timer interrupt 2. Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".
IE.6	–	Reserved bit. The output data is "1" if the bit is read.
IE.7	EA	Overall interrupt control bit. All interrupts are disabled when bit is "0". All interrupts are controlled by IE.0 through IE.5 when bit is "1".

INTERRUPT PRIORITY REGISTER (IP)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
IP	0B8H	PCT	–	PT2	PS	PT1	PX1	PT0	PX0

BIT LOCATION	FLAG	FUNCTION
IP.0	PX0	Interrupt priority bit for external interrupt 0. Priority is assigned when bit is "1".
IP.1	PT0	Interrupt priority bit for timer interrupt 0. Priority is assigned when bit is "1".
IP.2	PX1	Interrupt priority bit for external interrupt 1. Priority is assigned when bit is "1".
IP.3	PT1	Interrupt priority bit for timer interrupt 1. Priority is assigned when bit is "1".
IP.4	PS	Interrupt priority bit for serial port. Priority is assigned when bit is "1".
IP.5	PT2	Interrupt priority bit for timer interrupt 2. Priority is assigned when bit is "1".
IP.6	–	Reserved bit. The output data is "1" if the bit is read.
IP.7	PCT	Priority interrupt circuit control bit. The priority register contents are valid and priority assigned interrupts can be processed when this bit is "0". When the bit is "1", the priority interrupt circuit is stopped, and interrupts can only be controlled by the interrupt enable register (IE).

PROGRAM STATUS WORD REGISTER (PSW)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
PSW	0D0H	CY	AC	F0	RS1	RS0	OV	F1	P

BIT LOCATION	FLAG	FUNCTION
PSW.0	P	Accumulator (ACC) parity indicator. "1" when the "1" bit number in the accumulator is an odd number, and "0" when an even number.
PSW.1	F1	User flag which may be set to "0" or "1" as desired by the user.
PSW.2	OV	Overflow flag which is set if the carry C_6 from bit 6 of the ALU or CY is "1" as a result of an arithmetic operation. The flag is also set to "1" if the resultant product of executing a multiplication instruction (MULAB) is greater than 0FFH, but is reset to "0" if the product is less than or equal to 0FFH.
PSW.3	RS0	RAM register bank switch.
PSW.4	RS1	RS1 = 0, RS0 = 0, BANK = 0, RAM ADDRESS = 00H - 07H RS1 = 0, RS0 = 1, BANK = 1, RAM ADDRESS = 08H - 0FH RS1 = 1, RS0 = 0, BANK = 2, RAM ADDRESS = 10H - 17H RS1 = 1, RS0 = 1, BANK = 3, RAM ADDRESS = 18H - 1FH
PSW.5	F0	User flag which may be set to "0" or "1" as desired by the user.
PSW.6	AC	Auxiliary carry flag. This flag is set to "1" if a carry C_3 is generated from bit 3 of the ALU as a result of executing an arithmetic operation instruction. In all other cases, the flag is reset to "0".
PSW.7	CY	Main carry flag. This flag is set to "1" if a carry C_7 is generated from bit 7 of the ALU as result of executing an arithmetic operation instruction. If a carry C_7 is not generated, the flag is reset to "0".

I/O CONTROL REGISTER (IOCON)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
IOCON	0F8H	WDT	T32	SERR	IZC	P3HZ	P2HZ	P1HZ	ALF

BIT LOCATION	FLAG	FUNCTION
IOCON.0	ALF	If CPU power down mode (PD, HPD) is activated with this bit set to "1", the outputs from ports 0, 1, 2 and 3 are switched to floating status. When this bit is "0", ports 0, 1, 2 and 3 are in output mode.
IOCON.1	P1HZ	Port 1 becomes a floating state input port when this bit is "1".
IOCON.2	P2HZ	Port 2 becomes a floating state input port when this bit is "1".
IOCON.3	P3HZ	Port 3 becomes a floating state input port when this bit is "1".
IOCON.4	IZC	The 10 kohm pull-up resistance for ports 1, 2 and 3 is switched off when this bit is "1", leaving only the 100 kohm pull-up resistance.
IOCON.5	SERR	Serial port reception error flag. This flag is set to "1" if an overrun or framing error is generated when data is received at a serial port. The flag is reset by software.
IOCON.6	T32	Timer/counters 0 and 1 are connected serially to from a 32-bit timer/counter when this bit is set to "1". TF1 of TCON is set if a carry is generated in the 32-bit timer/counter.
IOCON.7	WDT	Watchdog timer mode is set when this bit is set to "1". And if TF1 is set to "1" after watchdog timer mode has been set, the CPU is reset and the program is executed from address 0.

TIMER 2 CONTROL REGISTER (T2CON)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
T2CON	0C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

BIT LOCATION	FLAG	FUNCTION
T2CON.0	CP/$\overline{RL2}$	Capture mode is set when TCLK + RCLK = "0" and CP/$\overline{RL2}$ = "1". 16-bit auto reload mode is set when TCLK + RCLK = "0" and CP/$\overline{RL2}$ = "0". CP/$\overline{RL2}$ is ignored when TCLK + RCLK = "1".
T2CON.1	C/$\overline{T2}$	Timer/counter 2 count clock designation control bit. The internal clocks (XTAL1-2 ÷ 12, XTAL1-2 ÷ 2) are used when this bit is "0", and the external clock applied to the T2 is passed to timer/counter 2 when the bit is "1".
T2CON.2	TR2	Timer/counter 2 counting start and stop control bit. Timer/counter 2 commences counting when this bit is "1" and stops counting when "0".
T2CON.3	EXEN2	T2EX timer/counter 2 external control signal control bit. Input of the T2EX signal is disabled when this bit is "0", and enabled when "1".
T2CON.4	TCLK	Serial port transmit circuit drive clock control bit. Timer/counter 2 is switched to baud rate generator mode when this bit is "1", and the timer/counter 2 carry signal becomes the serial port transmit clock. Note, however, that the serial ports can only use the timer/counter 2 carry signal in serial port modes 1 and 3.
T2CON.5	RCLK	Serial port receive circuit drive clock control bit. Timer/counter 2 is switched to baud rate generator mode when this bit is "1", and the timer/counter 2 carry signal becomes the serial port receive clock. Note, however, that the serial ports can only use the timer/counter 2 carry signal in serial port modes 1 and 3.
T2CON.6	EXF2	Timer/counter 2 external flag. This bit is set to "1" when the T2EX timer/counter 2 external control signal level is changed from "1" to "0" while EXEN2 = "1". This flag serves as the timer interrupt 2 request signal. If an interrupt is generated, EXF2 must be reset to "0" by software.
T2CON.7	TF2	Timer/counter 2 carry flag. This bit is set to "1" by a carry signal when timer/counter 2 is in 16-bit auto reload mode or in capture mode. This flag serves as the timer interrupt 2 request signal. if an interrupt is generated, TF2 must be reset to "0" by software.

LIST OF INSTRUCTIONS

LIST OF INSTRUCTION SYMBOLS

A	: Accumulator
AB	: Register pair
AC	: Auxiliary carry flag
B	: Arithmetic operation register
C	: Carry flag
DPTR	: Data pointer
PC	: Program counter
Rr	: Register indicator (r = 0 ~ 7)
SP	: Stack pointer
AND	: Logical product
OR	: Logical sum
XOR	: Exclusive OR
+	: Addition
–	: Substraction
x	: Multiplication
/	: Division
(x)	: Denotes the contents of x
((x))	: Denotes the contents of address determined by the contents of x
#	: Denotes the immediate data
@	: Denotes the indirect address
=	: Equality
≠	: Non equality
←	: Substitution
→	: Substitution
–	: Negation
<	: Smaller than
>	: Larger than
bit address	: RAM and the special function register bit specifier address (b_0 ~ b_7)
code address	: Absolute address (A_0 ~ A_1)
data	: immediate data (I_0 ~ I_7)
relative offset	: Relative jump address offset value (R_0 ~ R_7)
direct address	: RAM and the special function register byte specifier address (a_0 ~ a_7)

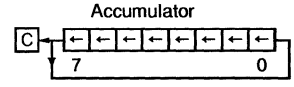
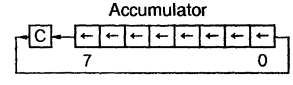
INSTRUCTION TABLE

L / H	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111
0 0000	NOP	AJMP address 11 (Page 0)	LJMP address 16	RR A	INC A	INC direct	INC @ RO	INC @ R1
1 0001	JBC bit, rel	ACALL address 11 (Page 0)	LCALL address 16	RRC A	DEC A	DEC direct	DEC @ RO	DEC @ R1
2 0010	JB bit, rel	AJMP address 11 (Page 1)	RET	RL A	ADD A, # data	ADD A, direct	ADD A, @ R0	ADD A, @ R1
3 0011	JNB bit, rel	ACALL adress 11 (Page 1)	RETI	RLC A	ADDC A, # data	ADDC A, direct	ADDC A, @ R0	ADDC A, @ R1
4 0100	JC bit, rel	AJMP address 11 (Page 2)	ORL direct, A	ORL direct, # data	ORL A, # data	ORL A, direct	ORL A, @ R0	ORL A, @ R1
5 0101	JNC rel	ACALL address 11 (Page 2)	ANL direct, A	ANL direct # data	ANL A, # data	ANL A, direct	ANL A, @ R0	ANL A, @ R1
6 0110	JZ rel	AJMP address 11 (Page 3)	XRL direct, A	XRL direct # data	XRL A, # data	XRL A, direct	XRL A, @ R0	XRL A, @ R1
7 0111	JNZ rel	ACALL address 11 (Page 3)	ORL C, bit	JMP @ A+ DPTR	MOV A, # data	MOV direct, # data	MOV @ R0, # data	MOV @ R1, # data
8 1000	SJMP rel	AJMP address 11 (Page 4)	ANL C, bit	MOVC A, @ A+ PC	DIV AB	MOV direct 1, direct 2	MOV direct, @ R0	MOV direct, @ R1
9 1001	MOV DPTR # data 16	ACALL address 11 (Page 4)	MOV bit, C	MOVC A, @ A+ DPTR	SUBB A, # data	SUBB A, direct	SUBB A, @ R0	SUBB A, @ R1
A 1010	ORAL C, bit	AJMP address 11 (Page 5)	MOV C, bit	INC DPTR	MUL AB		MOV @ R0, direct	MOV @ R1, direct
B 1011	ANL C bit,	ACALL address 11 (Page 5)	CPL bit	CPL C	CJNE A, # data, rel	CJNE A, direct, rel	CJNE @ R0, # data, rel	CJNE @ R1, # data, rel
C 1100	PUSH direct	AJMP address 11 (Page 6)	CLR bit	CLR C	SWAP A	XCH A, direct	XCH A, @ R0	XCH A, @ R1
D 1101	POP direct	ACALL address 11 (Page 6)	SETB bit	SETB C	DA A	DJNZ direct, rel	XCHD A, @ R0	XCH A, @ R1
E 1110	MOVX A, @ DPTR	AJMP address 11 (Page 7)	MOVX A, @ R0	MOVX A, @ R1	CLR A	MOV A, direct	MOV A, @ R0	MOV A, @ R1
F 1111	MOVX @ DPTR, A	ACALL address 11 (Page 7)	MOVX @ R0, A	MOVX @ R1, A	CPL A	MOV direct, A	MOV @ R0, A	MOV @ R1, A

2 BYTE 2 CYCLE	MNEMONIC	3 BYTE 4 CYCLE

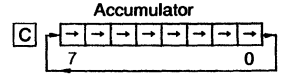
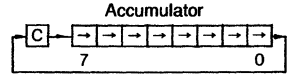
L / H	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
0 0000	INC R0	INC R1	INC R2	INC R3	INC R4	INC R5	INC R6	INC R7
1 0001	DEC R0	DEC R1	DEC R2	DEC R3	DEC R4	DEC R5	DEC R6	DEC R7
2 0010	ADD A, R0	ADD A, R1	ADD A, R2	ADD A, R3	ADD A, R4	ADD A, R5	ADD A, R6	ADD A, R7
3 0011	ADDC A, R0	ADDC A, R1	ADDC A, R2	ADDC A, R3	ADDC A, R4	ADDC A, R5	ADDC A, R6	ADDC A, R7
4 0100	ORL A, R0	ORL A, R1	ORL A, R2	ORL A, R3	ORL A, R4	ORL A, R5	ORL A, R6	ORL A, R7
5 0101	ANL A, R0	ANL A, R1	ANL A, R2	ANL A, R3	ANL A, R4	ANL A, R5	ANL A, R6	ANL A, R7
6 0110	XRL A, R0	XRL A, R1	XRL A, R2	XRL A, R3	XRL A, R4	XRL A, R5	XRL A, R6	XRL A, R7
7 0111	MOV R0, # data	MOV R1, # data	MOV R2, # data	MOV R3, # data	MOV R4, # data	MOV R5, # data	MOV R6, # data	MOV R7, # data
8 1000	MOV direct, R0	MOV direct, R1	MOV direct, R2	MOV direct, R3	MOV direct, R4	MOV direct, R5	MOV direct, R6	MOV direct, R7
9 1001	SUBB A, R0	SUBB A, R1	SUBB A, R2	SUBB A, R3	SUBB A, R4	SUBB A, R5	SUBB A, R6	SUBB A, R7
A 1010	MOV R0, direct	MOV R1, direct	MOV R2, direct	MOV R3, direct	MOV R4, direct	MOV R5, direct	MOV R6, direct	MOV R7, direct
B 1011	CJNE R0, # data rel	CJNE R1, # data rel	CJNE R2, # data rel	CJNE R3, # data rel	CJNE R4, # data rel	CJNE R5, # data rel	CJNE R6, # data rel	CJNE R7, # data rel
C 1100	XCH A, R0	XCH A, R1	XCH A, R2	XCH A, R3	XCH A, R4	XCH A, R5	XCH A, R6	XCH A, R7
D 1101	DJNZ R0, rel	DJNZ R1, rel	DJNZ R2, rel	DJNZ R3, rel	DJNZ R4, rel	DJNZ R5, rel	DJNZ R6, rel	DJNZ R7, rel
E 1110	MOV A, R0	MOV A, R1	MOV A, R2	MOV A, R3	MOV A, R4	MOV A, R5	MOV A, R6	MOV A, R7
F 1111	MOV R0, A	MOV R1, A	MOV R2, A	MOV R3, A	MOV R4, A	MOV R5, A	MOV R6, A	MOV R7, A

INSTRUCTION SET DETAILS

MNEMONIC		INSTRUCTION CODE D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	BYTES	CYCLES	DESCRIPTION
ARITHMETIC OPERATION INSTRUCTIONS												
ADD	A,Rr	0	0	1	0	1	r_2	r_1	r_0	1	1	(AC), (0V), (C), (A) ← (A) + (Rr)
ADD	A, direct	0	0	1	0	0	1	0	1	2	1	(AC), (0V), (C), (A) ← (A) + (direct address)
ADD	A, @Rr	0	0	1	0	0	1	1	r_0	1	1	(AC), (0V), (C), (A) ← (A) + ((Rr))
ADD	A, #data	0 I_7	0 I_6	1 I_5	0 I_4	0 I_3	1 I_2	0 I_1	0 I_0	2	1	(AC), (0V), (C), (A) ← (A) + #data
ADDC	A, Rr	0	0	1	1	1	r_2	r_1	r_0	1	1	(AC), (0V), (C), (A) ← (A) + (C) + (Rr)
ADDC	A, direct	0 a_7	0 a_6	1 a_5	1 a_4	0 a_3	1 a_2	0 a_1	1 a_0	2	1	(AC), (0V), (C), (A) ← (A) + (C) + (direct address)
ADDC	A, @Rr	0	0	1	1	0	1	1	r_0	1	1	(AC), (0V), (C), (A) ← (A) + (C) + ((Rr))
ADDC	A, #data	0 I_7	0 I_6	1 I_5	1 I_4	0 I_3	1 I_2	0 I_1	0 I_0	2	1	(AC), (0V), (C), (A) ← (A) + (C) + #data
SUBB	A, Rr	1	0	0	1	1	r_2	r_1	r_0	1	1	(AC), (0V), (C), (A) ← (A) – (C) – (Rr)
SUBB	A, direct	1	0	0	1	0	1	0	1	2	1	(AC), (0V), (C), (A) ← (A) – (C) – (direct address)
SUBB	A, @Rr	1	0	0	1	0	1	1	r_0	1	1	(AC), (0V), (C), (A) ← (A) – (C) – ((Rr))
SUBB	A, #data	1 I_7	0 I_6	0 I_5	1 I_4	0 I_3	1 I_2	0 I_1	0 I_0	2	1	(AC), (0V), (C), (A) ← (A) – (C) – #data)
MUL	AB	1	0	1	0	0	1	0	0	1	4	(AC) ← (A) x (B)
DIV	AB	1	0	0	0	0	1	0	0	1	4	(A) quotient (B) remainder ← (A)/B
DA	A	1	1	0	1	0	1	0	0	1	1	When the contents of accumulator bits 0 thru 3 are greater than 3, or when auxiliary carry (AC) is 1, 6 added to bits 0 thru 3. Bits 4 thru 7 are then examined and when bits 4 thru 7 following compensation of lower bits 0 thru 3 is greater than 9, or when carry (C) is 1, 6 added to bits 4 thru 7. As a result, the carry flag can be set, but cannot be cleared.
ACCUMULATOR OPERATION INSTRUCTIONS												
CLR	A	1	1	1	0	0	1	0	0	1	1	(A) ← 0
CPL	A	1	1	1	1	0	1	0	0	1	1	(A) ← $\overline{(A)}$
RL	A	0	0	1	0	0	0	1	1	1	1	
RLC	A	0	0	1	1	0	0	1	1	1	1	

6

INSTRUCTION SET DETAILS (CONT.)

MNEMONIC		INSTRUCTION CODE D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	BYTES	CYCLES	DESCRIPTION
RR	A	0	0	0	0	0	0	1	1	1	1	
RRC	A	0	0	0	1	0	0	1	1	1	1	
SWAP	A	1	1	0	0	0	1	0	0	1	1	$(A_{3-0}) \leftarrow (A_{7-4})$
INCREMENT/DECREMENT												
INC	A	0	0	0	0	0	1	0	0	1	1	(A) ← (A) + 1
INC	Rr	0	0	0	0	1	r_2	r_1	r_0	1	1	(Rr) ← (Rr) + 1
INC	direct	0	0	0	0	0	1	0	1	2	1	(direct address) ← (direct address) + 1
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
INC	@Rr	0	0	0	0	0	1	1	r_0	1	1	((Rr)) ← ((Rr)) + 1
INC	DPTR	1	0	1	0	0	0	1	1	1	2	(DPTR) ← (DPTR) + 1
DEC	A	0	0	0	1	0	1	0	0	1	1	(A) ← (A) – 1
DEC	Rr	0	0	0	1	1	r_2	r_1	r_0	1	1	(Rr) ← (Rr) – 1
DEC	direct	0	0	0	1	0	1	0	1	2	1	(direct address) ← (direct address) – 1
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
DEC	@Rr	0	0	0	1	0	1	1	r_0	1	1	((Rr)) ← ((Rr)) – 1
LOGICAL OPERATION INSTRUCTIONS												
ANL	A, Rr	0	1	0	1	1	r_2	r_1	r_0	1	1	(A) ← (A) AND (Rr)
ANL	A, direct	0	1	0	1	0	1	0	1	2	1	(A) ← (A) AND (direct address)
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
ANL	A, @Rr	0	1	0	1	0	1	1	r_0	1	1	(A) ← (A) AND (Rr)
ANL	A,#data	0	1	0	1	0	1	0	0	2	1	(A) ← (A) AND #data
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0			
ANL	direct, A	0	1	0	1	0	0	1	0	2	1	(direct address) ← (direct address) AND (A)
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
ANL	direct, #data	0	1	0	1	0	0	1	1	3	2	(direct address) ← (direct address) AND #data
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0			
ORL	A, Rr	0	1	0	0	1	r_2	r_1	r_0	1	1	(A) ← (A) OR (Rr)
ORL	A, direct	0	1	0	0	0	1	0	1	2	1	(A) ← (A) OR (direct address)
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
ORL	A, @Rr	0	1	0	0	0	1	1	r_0	1	1	(A) ← (A) OR ((Rr))
ORL	A, #data	0	1	0	0	0	1	0	0	2	1	(A) ← (A) OR #data
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0			

6

INSTRUCTION SET DETAILS (CONT.)

MNEMONIC		INSTRUCTION CODE D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	BYTES	CYCLES	DESCRIPTION
ORL	direct, A	0	1	0	0	0	0	1	0	2	1	(direct address) ← (direct address) OR (A)
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
ORL	direct, #data	0	1	0	0	0	0	1	1	3	2	(direct address) ← (direct address OR #data
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0			
XRL	A, Rr	0	1	1	0	1	r_2	r_1	r_0	1	1	(A) ← (A) XOR (Rr)
XRL	A, direct	0	1	1	0	0	1	0	1	2	1	(A) ← (A) XOR (direct address)
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
XRL	A, @Rr	0	1	1	0	0	1	1	r_0	1	1	(A) ← (A) XOR ((Rr))
XRL	A, #data	0	1	1	0	0	1	0	0	2	1	(A) ← XOR #data
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0			
XRL	direct, A	0	1	1	0	0	0	1	0	2	1	(direct address) ← (direct address) XOR (A)
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
XRL	direct, #data	0	1	1	0	0	0	1	1	3	2	(direct address) ← (direct address) XOR #data
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0			
IMMEDIATE DATA SETTING INSTRUCTIONS												
MOV	A, #data	0	1	1	1	0	1	0	0	2	1	(A) ← #data
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0			
MOV	Rr, #data	0	1	1	1	1	r_2	r_1	r_0	2	1	(Rr) ← #data
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0			
MOV	direct, ← data	0	1	1	1	0	1	0	1	3	2	(direct address) ← #data
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0			
MOV	@Rr, #data	0	1	1	1	0	1	1	r_0	2	1	(Rr) ← #data
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOV	DPTR, #data 16	1	0	0	1	0	0	0	0	3	2	(DPTR) ← #data 16
		I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8			
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0			
CARRY FLAG OPERATION INSTRUCTIONS												
CLR	C	1	1	0	0	0	0	1	1	1	1	(C) ← 0
SETB	C	1	1	0	1	0	0	1	1	1	1	(C) ← 1
CPL	C	1	0	1	1	0	0	1	1	1	1	(C) ← $(\overline{C})$
ANL	C, bit	1	0	0	0	0	0	1	0	2	2	(C) ← (C) AND (bit address)
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0			
ANL	C,/bit	1	0	1	1	0	0	0	0	2	2	(C) ← (C) AND $(\overline{\text{bit address}})$
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0			
ORL	C, bit	0	1	1	1	0	0	1	0	2	2	(C) ← (C) OR (bit address)
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0			
ORL	C,/bit	1	0	1	0	0	0	0	0	2	2	(C) ← (C) OR $(\overline{\text{bit address}})$
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0			
MOV	C, bit	1	0	1	0	0	0	1	0	2	1	(C) ← (bit address)
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0			
MOV	bit, C	1	0	0	1	0	0	1	0	2	2	(bit address) ← (C)
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0			

INSTRUCTION SET DETAILS (CONT.)

MNEMONIC	INSTRUCTION CODE D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	BYTES	CYCLES	DESCRIPTION
BIT OPERATION INSTRUCTIONS											
SETB bit	1	1	0	1	0	0	1	0	2	1	(bit address) ← 1
	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0			
CLR bit	1	1	0	0	0	0	1	0	2	1	(bit address) ← 0
	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0			
CPL bit	1	0	1	1	0	0	1	0	2	1	(bit address) ← $\overline{\text{(bit address)}}$
	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0			
DATA TRANSFER INSTRUCTIONS											
MOV A, Rr	1	1	1	0	1	r_2	r_1	r_0	1	1	(A) ← (Rr)
MOV A, direct	1	1	1	0	0	1	0	1	2	1	(A) ← (direct address)
	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOV A, @Rr	1	1	1	0	0	1	1	r_0	1	1	(A) ← ((Rr))
MOV Rr, A	1	1	1	1	1	r_2	r_1	r_0	1	1	(Rr) ← (A)
MOV Rr, direct	1	0	1	0	1	r_2	r_1	r_0	2	2	(Rr) ← (direct address)
	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOV direct, A	1	1	1	1	0	1	0	1	2	1	(direct address) ← (A)
	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOV direct, Rr	1	0	0	0	1	r_2	r_1	r_0	2	2	(direct address) ← (Rr)
	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOV direct 1, direct 2	1	0	0	0	0	1	0	1	3	2	(direct address 1) ← (direct address 2)
	a_7^2	a_6^2	a_5^2	a_4^2	a_3^2	a_2^2	a_1^2	a_0^2			
	a_7^1	a_6^1	a_5^1	a_4^1	a_3^1	a_2^1	a_1^1	a_0^1			
MOV direct, @Rr	1	0	0	0	0	1	1	r_0	2	2	(direct address) ← ((Rr))
	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
MOV @Rr, A	1	1	1	1	0	1	1	r_0	1	1	((Rr)) ← (A)
MOV @Rr, direct	1	0	1	0	0	1	1	r_0	2	2	((Rr)) ← (direct address))
	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
CONSTANT CODE INSTRUCTIONS											
MOVC A, @A + DPTR	1	0	0	1	0	0	1	1	1	2	(A) ← ((A) + (DPTR))
MOVC A, @A + PC	1	0	0	0	0	0	1	1	1	2	(PC) ← (PC) + 1 (A) ← ((A) + (PC))
DATA EXCHANGE INSTRUCTIONS											
XCH A, Rr	1	1	0	0	1	r_2	r_1	r_0	1	1	(A) (Rr)
XCH A, direct	1	1	0	0	0	1	0	1	2	1	(A) (direct address)
	a_7	a_8	a_5	a_4	a_3	a_2	a_1	a_0			
XCH A, @Rr	1	1	0	0	0	1	1	r_0	1	1	(A) ((Rr))
XCHD A, @Rr	1	1	0	1	0	1	1	r_0	1	1	$(A_{0 \sim 3})$ $((Rr_{0 \sim 3}))$

INSTRUCTION SET DETAILS (CONT.)

MNEMONIC	INSTRUCTION CODE D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	BYTES	CYCLES	DESCRIPTION
SUBROUTINE INSTRUCTIONS				
PUSH direct	1 1 0 0 0 0 0 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2	(SP) ← (SP) + 1 ((SP)) ← (direct address)
POP direct	1 1 0 1 0 0 0 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2	(direct address) ← ((SP)) (SP) ← (SP) − 1
ACALL addr 11	A_{10} A_9 A_8 1 0 0 0 1 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0	2	2	(PC) ← (PC) + 2 (SP) ← (SP) + 1 ((SP)) ← ($PC_{0\sim7}$) (SP) ← (SP) + 1 ((SP)) ← ($PC_{8\sim15}$) ($PC_{0\sim10}$) ← $A_{0\sim10}$
LCALL addr 16	0 0 0 1 0 0 1 0 A_{15} A_{14} A_{13} A_{12} A_{11} A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0	3	2	(PC) ← (PC) + 3 (SP) ← (SP) + 1 ((SP)) ← ($PC_{0\sim7}$) (SP) ← (SP) + 1 ((SP)) ← ($PC_{8\sim15}$) ($PC_{0\sim15}$) ← $A_{0\sim15}$
RET	0 0 1 0 0 0 1 0	1	2	($PC_{8\sim15}$) ← ((SP)) (SP) ← (SP) − 1 ($PC_{0\sim7}$) ← ((SP)) (SP) ← (SP) − 1
RETI	0 0 1 1 0 0 1 0	1	2	($PC_{8\sim15}$) ← ((SP)) (SP) ← (SP) − 1 ($PC_{0\sim7}$) ← ((SP)) (SP) ← (SP) − 1
JUMP INSTRUCTIONS				
AJMP addr 11	A_{10} A_9 A_8 0 0 0 0 1 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0	2	2	(PC) ← (PC) + 2 ($PC_{0\sim10}$) ← $A_{0\sim10}$
LJMP addr 16	0 0 0 0 0 0 1 0 A_{15} A_{14} A_{13} A_{12} A_{11} A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0	3	2	($PC_{0\sim15}$) ← $A_{0\sim15}$
SJMP rel	1 0 0 0 0 0 0 0 R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0	2	2	(PC) ← (PC) + 2 (PC) ← (PC) + relative offset
JMP @A + DPTR	0 1 1 1 0 0 1 1	1	2	(PC) ← (A) + (DPTR)
BRANCH INSTRUCTIONS				

CJNE A, direct, rel	1 0 1 1 0 1 0 1
	a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0
	R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0
		3	2	(PC) ← (PC) + 3
				IF (A) ≠ (direct address)
				THEN
				(PC) ← (PC) + relative offset
				IF (A) < (direct address)
				THEN
				(C) ← 1
				ELSE
				(C) ← 0

CJNE A, #data, rel	1 0 1 1 0 1 0 0
	I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0
	R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0
		3	2	(PC) ← (PC) + 3
				IF (A) ≠ #data
				THEN
				(PC) ← (PC) + relative offset
				IF (A) < #data
				THEN
				(C) ← 1
				ELSE
				(C) ← 0

INSTRUCTION SET DETAILS (CONT.)

MNEMONIC		INSTRUCTION CODE D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	BYTES	CYCLES	DESCRIPTION
CJNE	Rr, #data, rel	1	0	1	1	1	r_2	r_1	r_0	3	2	(PC) ← (PC) + 3 IF ((Rr)) ≠ #data THEN (PC) ← (PC) + relative offset IF ((Rr)) < #data THEN (C) ← 1 ELSE (C) ← 0
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0			
		R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0			
CJNE	@Rr, #data, rel	1	0	1	1	0	1	1	r_0	3	2	(PC) ← (PC) + 3 IF ((Rr)) ≠ #data THEN (PC) ← (PC) + relative offset IF ((Rr)) < #data THEN (C) ← 1 ELSE (C) ← 0
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0			
		R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0			
DJNZ	Rr, rel	1	1	0	1	1	r_2	r_1	r_0	2	2	(PC) ← (PC) + 2 (Rr) ← (Rr) – 1 IF (Rr) > 0 or (Rr) < 0 THEN (PC) ← (PC) + relative offset
		R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0			
DJNZ	direct, rel	1	1	0	1	0	1	0	1	3	2	(PC) ← (PC) + 3 (direct address) ← (direct address) – 1 IF (direct address) ≠ 0 THEN (PC) ← (PC) + relative offset
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			
		R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0			
JZ	rel	0	1	1	0	0	0	0	0	2	2	(PC) ← (PC) + 2 IF (A) ≠ 0 THEN (PC) ← (PC) + relative offset
		R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0			
JNZ	rel	0	1	1	1	0	0	0	0	2	2	(PC) ← (PC) + 2 IF (A) ≠ 0 THEN (PC) ← (PC) + relative offset
		R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0			
JC	rel	0	1	0	0	0	0	0	0	2	2	(PC) ← (PC) + 2 IF (C) = 1 THEN (PC) ← (PC) + relative offset
		R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0			
JNC	rel	0	1	0	1	0	0	0	0	2	2	(PC) ← (PC) + 2 IF (C) = 0 THEN (PC) ← (PC) + relative offset
		R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0			
JB	bit, rel	0	0	1	0	0	0	0	0	3	2	(PC) ← (PC) + 3 IF (bit address) = 1 THEN (PC) ← (PC) + relative offset
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0			
		R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0			
JNB	bit, rel	0	0	1	1	0	0	0	0	3	2	(PC) ← (PC) + 3 IF (bit address) = 0 THEN (PC) ← (PC) + relative offset
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0			
		R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0			
JBC	bit, rel	0	0	0	1	0	0	0	0	3	2	(PC) ← (PC) + 3 IF (bit address) = 1 THEN (bit address) ← 0 (PC) ← (PC) + relative offset
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0			
		R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0			

INSTRUCTION SET DETAILS (CONT.)

MNEMONIC	INSTRUCTION CODE								BYTES	CYCLES	DESCRIPTION
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0			
EXTERNAL MEMORY INSTRUCTIONS											
MOVX A, @Rr	1	1	1	0	0	0	1	r_0	1	2	(A) ← ((Rr)) EXTERNAL RAM
MOVX A, @DPTR	1	1	1	0	0	0	0	0	1	2	(A) ← ((DPTR)) EXTERNAL RAM
MOVX @Rr, A	1	1	1	1	0	0	1	r_0	1	2	(Rr) ← (A) EXTERNAL RAM
MOVX @DPTR, A	1	1	1	1	0	0	0	0	1	2	((DPTR)) ← (A) EXTERNAL RAM
OTHER INSTRUCTIONS											
NOP	0	0	0	0	0	0	0	0	1	1	(PC) ← (PC) + 1

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias :
C = commercial ..... 0°C to 70°C
I = industrial ..... – 40°C to + 85°C
Storage Temperature ..... – 65°C to + 150°C
Voltage on V_{CC} to V_{SS} ..... – 0.5 V to + 7 V
Voltage on Any Pin to V_{SS} ..... – 0.5 V to V_{CC} + 0.5 V
Power Dissipation ..... 200 mW

**NOTICE : Stresses at or above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.*

DC CHARACTERISTICS

(T_A = – 40°C to 85°C ; VCC = 5 V ± 10 % ; VSS = 0 V ; F = 0 to 16 MHz)

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITIONS
VIL	Input Low Voltage	– 0.5	0.2 VCC – 0.1	V	
VIH	Input High Voltage (Except XTAL and RST)	0.2 VCC + 0.9	VCC + 0.5	V	
VIH1	Input High Voltage (RST and XTAL1)	0.7 VCC	VCC + 0.5	V	
VOL	Output Low Voltage (Ports 1, 2, 3)		0.45	V	IOL = 1.6 mA (note 3)
VOL1	Output Low Voltage Port 0, ALE, $\overline{PSEN}$		0.45	V	IOL = 3.2 mA (note 3)
VOH	Output High Voltage Ports 1, 2, 3	0.9 VCC		V	IOH = – 10 µA
		0.75 VCC		V	IOH = 25 µA
		2.4		V	IOH = – 60 µA VCC = 5 V ± 10 %
VOH2	Output High Voltage Port 1, 2, 3 IZC = 1	0.75 VCC		V	IOH = – 2.5 µA
VOH1	Output High Voltage (Port 0 (Port 0, ALE, $\overline{PSEN}$)	0.9 VCC		V	IOH = – 80 µA
		0.75 VCC		V	IOH = – 300 µA
		2.4		V	IOH = – 800 µA VCC = 5 V ± 10 %
IIL	Logical 0 Input Current Ports 1, 2, 3		C – 50 I – 60	µA	Vin = 0.45 V
ILI	Input Leakage Current (Port 0, EA)		± 10	µA	0.45 < Vin < VCC
ITL	Logical 1 to 0 Transition Current (Ports 1, 2, 3)		– 650	µA	Vin = 2.0 V
IPD	Power Supply Current (Power Down Mode)		50	µA	VCC = 2.0 V to 5.5 V (note 2)
RRST	RST Pulldown Resistor	50	150	kΩ	
CIO	Capacitance of I/O Buffer		10	pF	f_C = 1 MHz, T_A = 25°C
ICC	Power supply current Active mode 16 MHz Idle mode 16 MHz		32 9	mA mA	(notes 1, 2)

Note 1 :

ICC max is given by :
Active mode : ICCMAX = 2 x FREQ + 4
Idle Mode : ICCMAX = 0.5 x FREQ + 2

where FREQ is the external oscillator frequency in MHz. ICCMAX is given in mA. See *figure 1.*
See *figures 2 through 5* for ICC test conditions.

DC CHARACTERISTICS (AUTOMOTIVE)

(T_A = – 40°C to + 125°C ; VCC = 5 V ± 10 % ; VSS = 0 V)

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITIONS
VIL	Input Low Voltage	– 0.5	0.2 VCC – 0.1	V	
VIH	Input High Voltage (Except XTAL and RST)	0.2 VCC + 0.9	VCC + 0.5	V	
VIH1	Input High Voltage (RST and XTAL1)	0.7 VCC	VCC + 0.5	V	
VOL	Output Low Voltage (Ports 1, 2, 3)		0.45	V	IOL = 1.6 mA (note 3)
VOL1	Output Low Voltage Port 0, ALE, $\overline{PSEN}$		0.45	V	IOL = 3.2 mA (note 3)
VOH	Output High Voltage Ports 1, 2, 3	0.9 VCC		V	IOH = – 10 µA
		0.75 VCC		V	IOH = 25 µA
		2.4		V	IOH = – 60 µA VCC = 5 V ± 10 %
VOH1	Output High Voltage (Port 0 (Port 0, ALE, $\overline{PSEN}$)	0.9 VCC		V	IOH = – 80 µA
		0.75 VCC		V	IOH = – 300 µA
		2.4		V	IOH = – 800 µA VCC = 5 V ± 10 %
VOH2	Output High Voltage Port 1, 2, 3 IZC = 1	0.75 VCC		V	IOH = – 2.5 µA
IIL	Logical 0 Input Current Ports 1, 2, 3		– 75	µA	Vin = 0.45 V
ILI	Input Leakage Current (Port 0, EA)		± 10	µA	0.45 < Vin < VCC
ITL	Logical 1 to 0 Transition Current (Ports 1, 2, 3)		– 750	µA	Vin = 2.0 V
IPD	Power Supply Current (Power Down Mode)		75	µA	VCC = 2.0 V to 5.5 V (note 2)
RRST	RST Pulldown Resistor	50	150	kΩ	
CIO	Capacitance of I/O Buffer		10	pF	f_C = 1 MHz, T_A = 25°C
ICC	Power supply current Active mode 12 MHz Idle mode 12 MHz		28 8	mA mA	(notes 1, 2)

Note 2 :

ICC is measured with all output pins disconnected ; XTAL1 driven with TCLCH, TCHCL = 5 ns, VIL = VSS + .5 V, VIH = VCC – .5 V ; XTAL2 N.C. ; EA = RST = Port 0 = VCC. ICC would be slightly higher if a crystal oscillator is used. Idle ICC is measured with all output pins disconnected ; XTAL1 driven with TCLCH, TCHCL = 5 ns, VIL = VSS + .5 V, VIH = VCC – .5 V ; XTAL2 N.C. ; Port 0 = VCC ; EA = RST = VSS.
Power Down ICC is measured with all output pins disconnected ; EA = PORT 0 = VCC ; XTAL2 N.C. ; RST = VSS.

Note 3 :

Capacitance loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the VOLS of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1 to 0 transitions during bus operations. In the worst cases (capacitive loading 100 pF), the noise pulse on the ALE line may exceed 0.45 V with maxi VOL peak 0.6 V.A Schmitt Trigger use is not necessary.

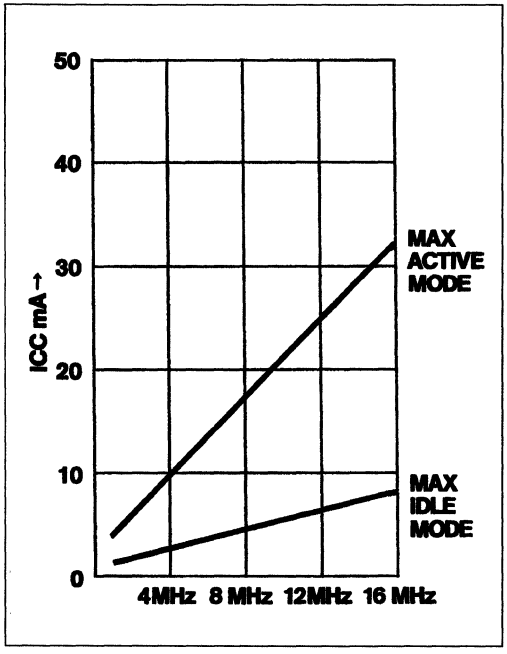


Figure 1 : ICC vs. Frequency. Valid only within frequency specifications of the device under test.

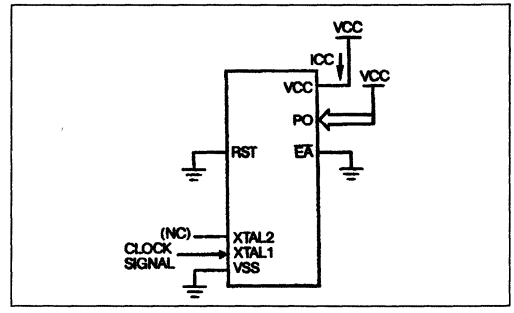


Figure 2 : ICC Test Condition, Idle Mode. All other pins are disconnected.

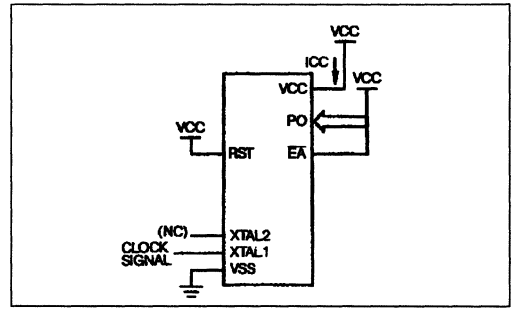


Figure 3 : ICC Test Condition, Active Mode. All other pins are disconnected.

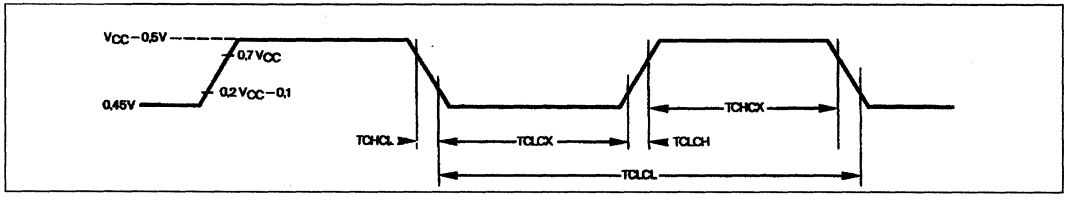


Figure 4 : Clock Signal Waveform for ICC Tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns.

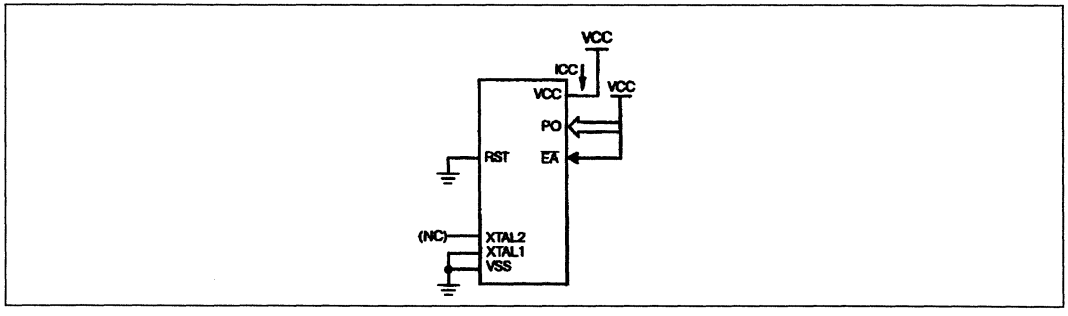


Figure 5 : ICC Test Condition, Power Down Mode. All other pins are disconnected.

EXTERNAL CLOCK DRIVE CHARACTERISTICS (XTAL1)

SYMBOL	PARAMETER	VARIABLE CLOCK FREQ = 0 to 16 MHz		UNIT
		MIN	MAX	
1/TCLCL	Oscillator Frequency	62.5		ns
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

*83C154-1/80C154-1 versions only.

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

A.C. PARAMETERS :

TA = 0°C + 70°C ; V_{SS} = 0 V ; V_{CC} = 5 V ± 10 % (commercial).

TA = – 40°C + 85°C ; V_{SS} = 0 V ; V_{CC} = 5 V ± 10 % (industrial).

(Load Capacitance for port 0, ALE, and $\overline{PSEN}$ = 100 pf ; Load Capacitance for All Other Outputs = 80 pf).

SYMBOL	PARAMETER	MIN	MAX	UNIT
TLHLL	ALE Pulse Width	2TCLCL-40		ns
TAVLL	Address Valid to ALE	TCLCL-40		ns
TLLAX	Address Hold After ALE	TCLCL-35		ns
TLLIV	ALE to Valid Instr in		4TCLCL-100	ns
TLLPL	ALE to $\overline{PSEN}$	TCLCL-40		ns
TPLPH	$\overline{PSEN}$ Pulse Width	3TCLCL-45		ns
TPLIV	$\overline{PSEN}$ to Valid Instr in		3TCLCL-105	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		TCLCL-15	ns
TPXAV	$\overline{PSEN}$ to Address Valid	TCLCL-8		ns
TAVIV	Address to Valid Instr in		5TCLCL-105	ns
TPAZ	$\overline{PSEN}$ Low to Address Float		10	ns
TRLRH	$\overline{RD}$ Pulse Width	6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	6TCLCL-100		ns
TLLAX	Data Address Hold After ALE	TCLCL-35		ns
TRLDV	$\overline{RD}$ to Valid Data in		5TCLCL-165	ns
TRHDX	Data Hold After $\overline{RD}$	0		ns
TRHDZ	Data Float After $\overline{RD}$		2TCLCL-70	ns
TLLDV	ALE to Valid Data in		8TCLCL-150	ns
TAVDV	Address to Valid Data in		9TCLCL-165	ns
TLLWL	ALE to $\overline{WR}$ or $\overline{RD}$	3TCLCL-50	3TCLCL+50	ns
TAVWL	Address to $\overline{WR}$ or $\overline{RD}$	4TCLCL-130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	TCLCL-60		ns
TQVWH	Data Setup to $\overline{WR}$ High	7TCLCL-150		ns
TWHQX	Data Hold After $\overline{WR}$	TCLCL-50		ns
TRLAZ	RD Low to Address Float		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	TCLCL-40	TCLCL+40	ns

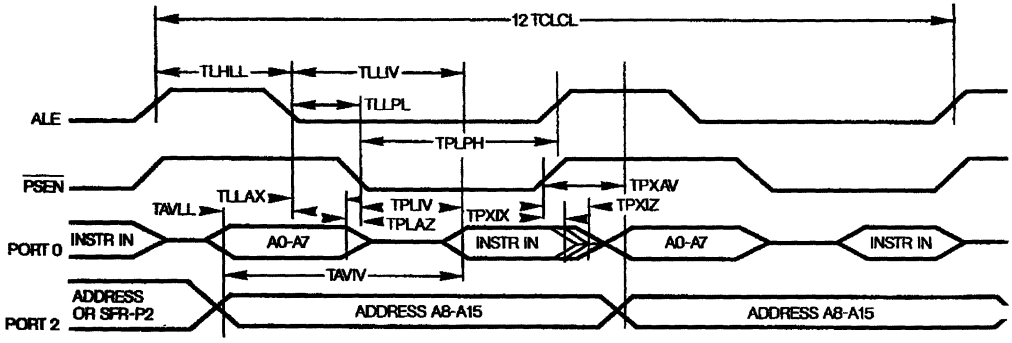
AC PARAMETERS :

TA = – 40°C to + 125°C ; V_{SS} = 0 V ; V_{CC} = 5 V ± 10 % (Automotive)

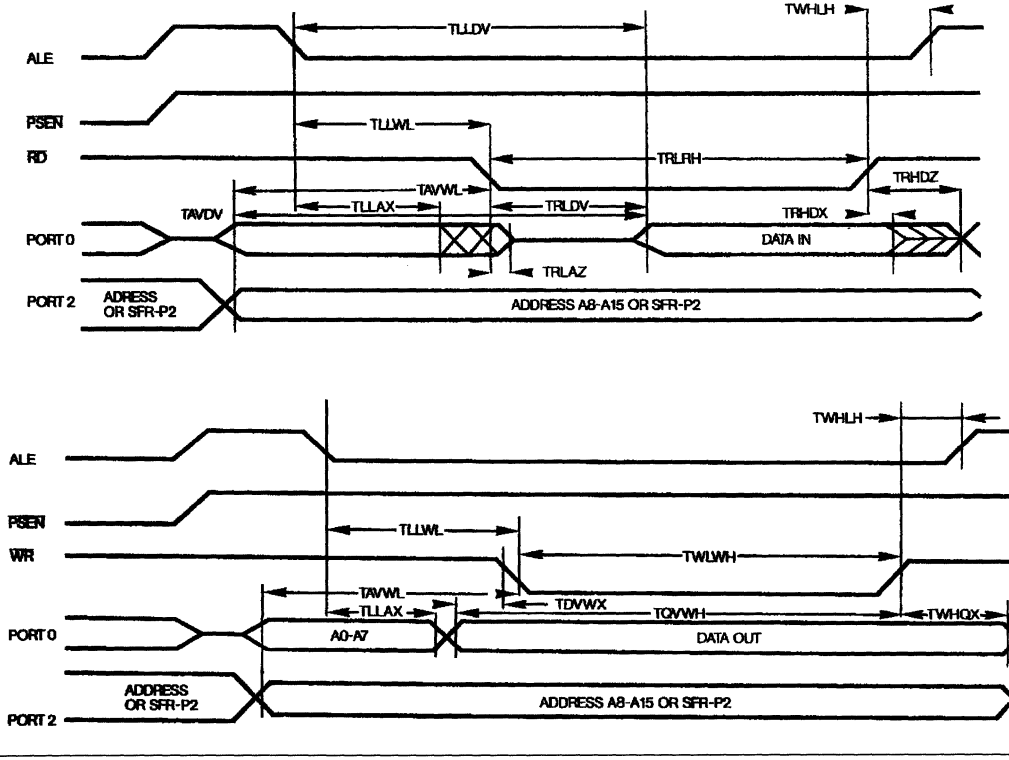
SYMBOL	PARAMETER	MIN	MAX	UNIT
TLHLL	ALE Pulse Width	2TCLCL-55		ns
TAVLL	Address Valid to ALE	TCLCL-70		ns
TLLAX	Address Hold After ALE	TCLCL-35		ns
TLLIV	ALE to Valid Instr in		4TCLCL-115	ns
TLLPL	ALE to $\overline{PSEN}$	TCLCL-55		ns
TPLPH	$\overline{PSEN}$ Pulse Width	3TCLCL-60		ns
TPLIV	$\overline{PSEN}$ to Valid Instr in		3TCLCL-120	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		TCLCL-40	ns
TPXAV	$\overline{PSEN}$ to Address Valid	TCLCL-8		ns
TAVIV	Address to Valid Instr in		5TCLCL-120	ns
TPAZ	$\overline{PSEN}$ Low to Address Float		25	ns
TRLRH	$\overline{RD}$ Pulse Width	6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	6TCLCL-100		ns
TLLAX	Data Address Hold After ALE	TCLCL-50		ns
TRLDV	$\overline{RD}$ to Valid Data in		5TCLCL-185	ns
TRHDX	Data Hold After $\overline{RD}$	0		ns
TRHDZ	Data Float After $\overline{RD}$		2TCLCL-85	ns
TLLDV	ALE to Valid Data in		8TCLCL-170	ns
TAVDV	Address to Valid Data in		9TCLCL-185	ns
TLLWL	ALE to $\overline{WR}$ or $\overline{RD}$	3TCLCL-65	3TCLCL+65	ns
TAVWL	Address to $\overline{WR}$ or $\overline{RD}$	4TCLCL-145		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	TCLCL-75		ns
TQVWH	Data Setup to $\overline{WR}$ High	7TCLCL-150		ns
TWHQX	Data Hold After $\overline{WR}$	TCLCL-65		ns
TRLAZ	RD Low to Address Float		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	TCLCL-65	TCLCL+65	ns

AC TIMING DIAGRAMS

EXTERNAL PROGRAM MEMORY READ CYCLE

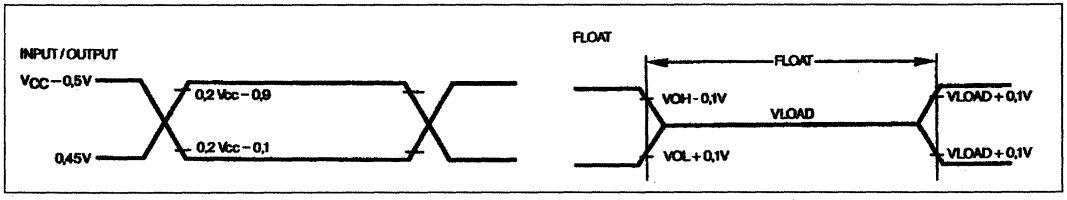


EXTERNAL DATA MEMORY READ CYCLE



6

AC TESTING INPUT/OUTPUT, FLOAT WAVEFORMS

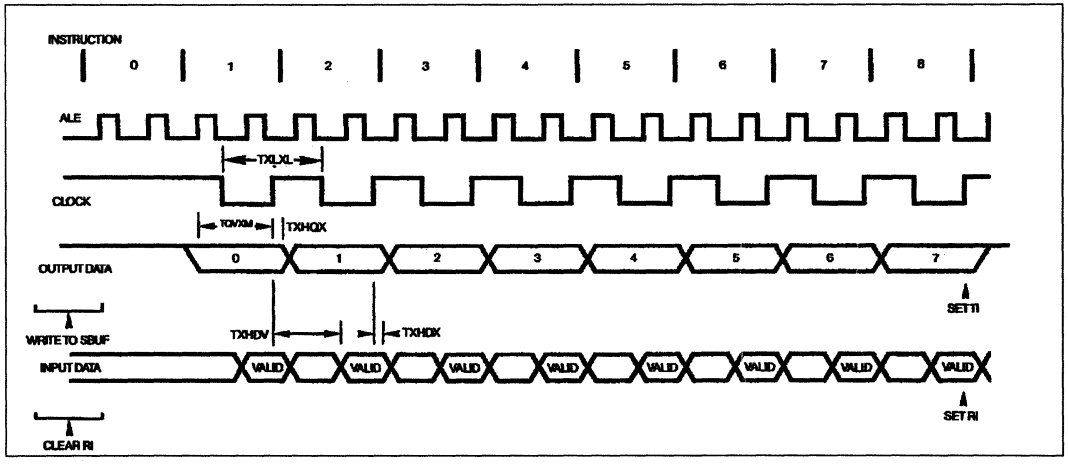


AC inputs during testing are driven at V_{CC} - 0.5 for a logic "1" and 0.45 V for a logic "0". Timing measurements are made at VIH min for a logic "1" and VIL max for a logic "0". For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change from the loaded VOH/VOL level occurs. Iol/IoH ≥ ± 20 mA.

SERIAL PORT TIMING - SHIFT REGISTER MODE

SYMBOL	PARAMETER	MIN	MAX	UNIT
TXLXL	Serial Port Clock Cycle Time	12TCLCL		µs
TQVXH	Output Data Setup to Clock Rising Edge	10TCLCL-133		ns
TXHQX	Output Data Hold after Clock Rising Edge	2TCLCL-117		ns
TXHDX	Input Data Hold after Clock Rising Edge	0		ns
TXHDV	Clock Rising Edge to Input Data Valid		10TLCL-133	ns

SHIFT REGISTER TIMING WAVEFORMS



EXPLANATION OF THE AC SYMBOLS

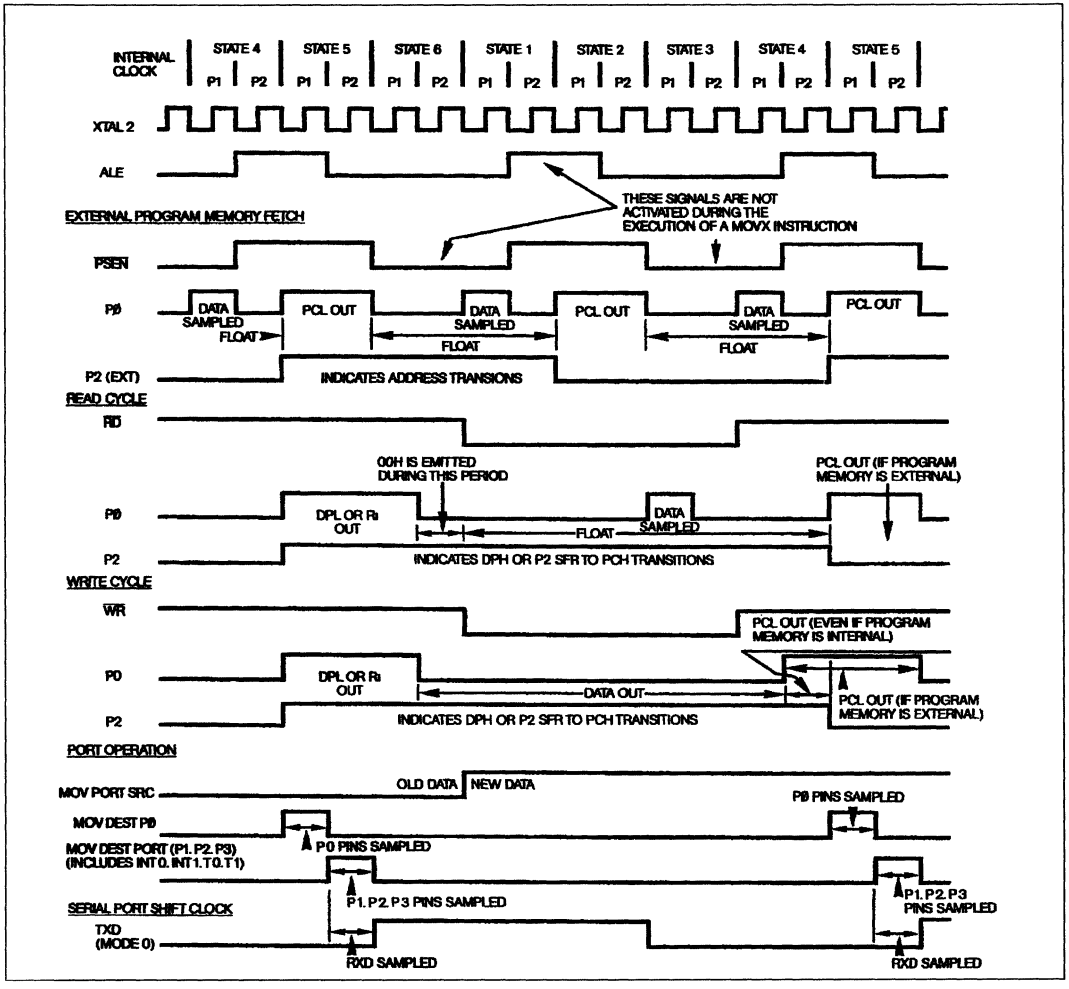
Each timing symbol has 5 characters. The first character is always a "T" (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

Example :

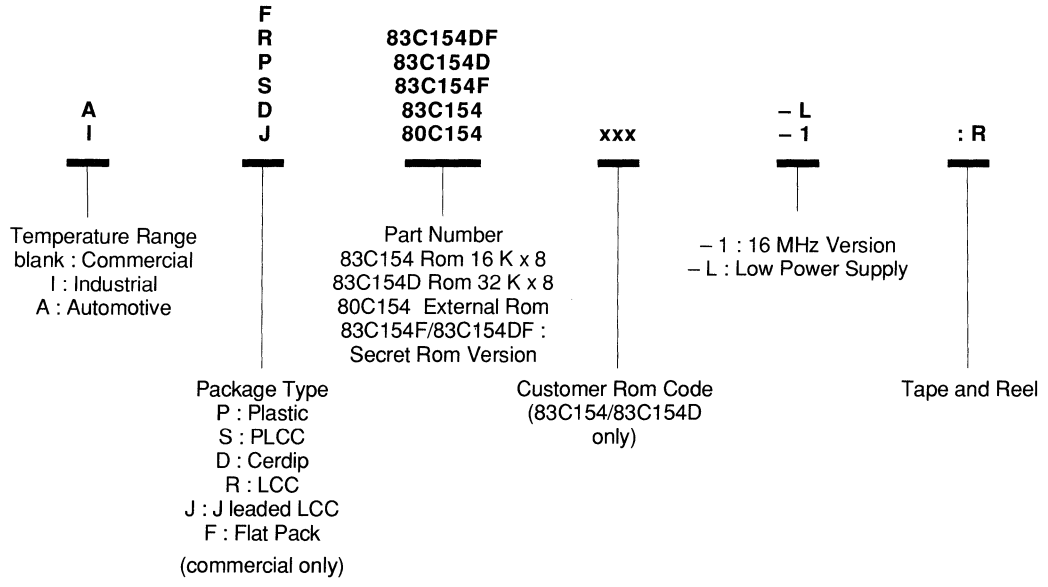
TAVLL = Time for Address Valid to ALE low.
TLLPL = Time for ALE low to PSEN low.

A : Address.
C : Clock.
D : Input data.
H : Logic level HIGH.
I : Instruction (program memory contents).
L : Logic level LOW, or ALE.
P : PSEN.
Q : Output data.
R : READ signal.
T : Time.
V : Valid.
W : WRITE signal.
X : No longer a valid logic level.
Z : Float.

CLOCK WAVEFORMS



This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins, however, ranges from 25 to 125 ns. This propagation delay is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component. Typically though (T_A = 25°C, fully loaded) RD and WR propagation delays are approximately 50 ns. The other signals are typically 85 ns. Propagation delays are incorporated in the AC specifications.


A
I
Temperature Range
blank : Commercial
I : Industrial
A : Automotive
F
R
P
S
D
J
Package Type
P : Plastic
S : PLCC
D : Cerdip
R : LCC
J : J leaded LCC
F : Flat Pack
(commercial only)
83C154DF
83C154D
83C154F
83C154
80C154
Part Number
83C154 Rom 16 K x 8
83C154D Rom 32 K x 8
80C154 External Rom
83C154F/83C154DF :
Secret Rom Version
xxx
Customer Rom Code
(83C154/83C154D
only)
– L
– 1
– 1 : 16 MHz Version
– L : Low Power Supply
: R
Tape and Reel

6

DATA SHEET

80C752/80C732

CMOS SINGLE-CHIP 8 BIT KEYBOARD CONTROLLER

FEATURES

- **4 K BYTES "QUICK-ROM" (80C752 only)**
- **256 BYTES RAM**
- **7 HIGH CURRENT I/O FULLY CUSTOMIZABLE (80C752 only)**
- **THREE 16-BIT TIMERS/COUNTERS**
- **INTERFACE FOR MECHANICAL AND RESISTIVE KEYBOARDS (82C752-M)**
- **INTERFACE FOR CAPACITIVE AND SWITCH CAP KEYBOARDS (80C752-C)**
- **COMPATIBLE WITH 80C52**
- **N-KEY ROLLOVER COMPATIBLE**
- **32 I/O LINES**
- **PROGRAMMABLE SERIAL PORT**
- **6 INTERRUPT SOURCES**
- **KEY PRESSED DETECTION**
- **FULLY STATIC DESIGN**
- **SAJI VI CMOS PROCESS**
- **32 K DATA MEMORY SPACE**
- **64 K PROGRAM MEMORY SPACE**
- **POWER CONTROL MODES**
- **IBMPC SOFTWARE ROUTINE (PCS52, PCK52)**

DESCRIPTION

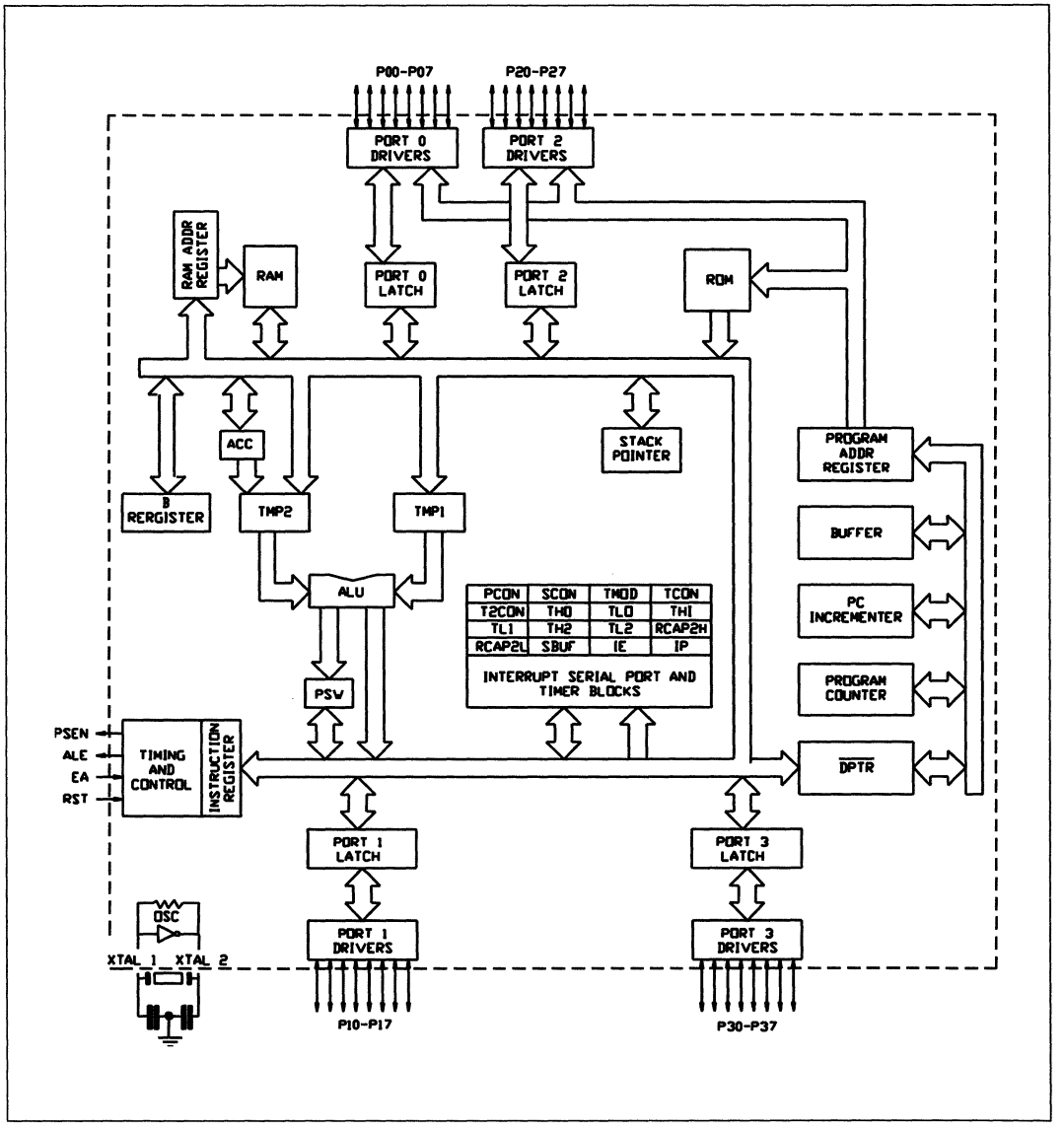
MHS's 80C752/80C732 is a high performance 8-bit single-chip microcontroller designed for keyboard applications. It is derived from the 80C52/80C32 and bear all its internal features, (except the ROM size, the I/O structure and addressing : refer to MHS's 80C52/80C32 data-sheet).

The 80C752/80C732 allows the user to build powerful, cost effective, and flexible keyboard controllers for mechanical or capacitive keyboards using only this "single chip solution". This is achieved by :

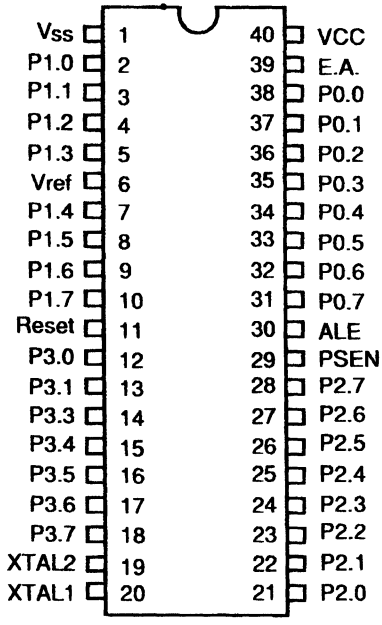
- the on-chip analog interface for capacitive or mechanical matrix,
- the keypressed detection circuit,
- the 7 high current and fully customizable I/O of port 3 which allow the user to configure the circuit to fit a wide range of keyboards arrays and to directly interface with accesories like mouse, card reader, bar code reader, LCD display, etc.

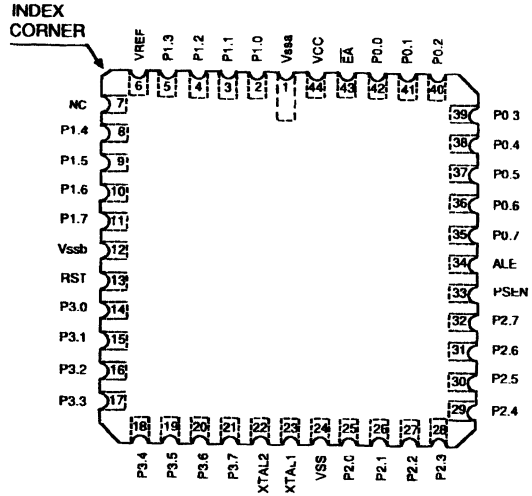
The MHS 80C752/80C732 are manufactured using the SAJI VI CMOS PROCESS and supplied in DIL 40 pins (80C752 only) or PLCC44 pins packages.

FUNCTIONAL BLOCK DIAGRAM

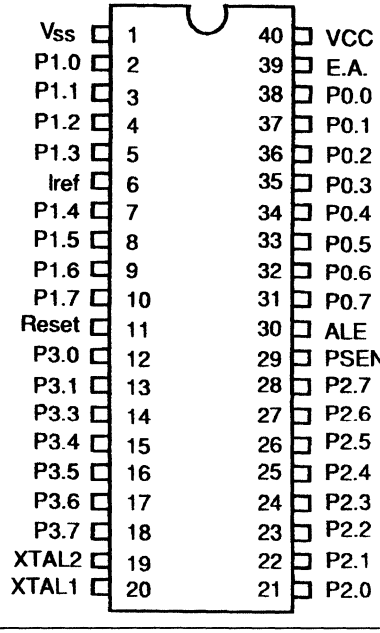


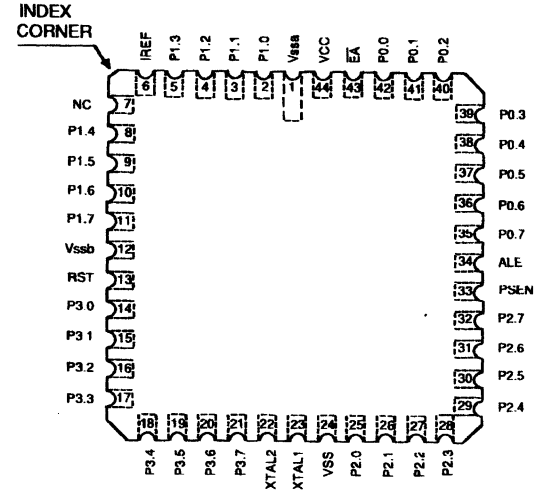
PIN CONFIGURATIONS 80C752-M/80C732-M





PIN CONFIGURATIONS 80C752-C/80C732-C





6

PIN DEFINITIONS AND FUNCTIONS

SYMBOL	INPUT (I) OUTPUT (O)	FUNCTION
P3.0-P3.1 P3.2* P3.3-P3.7 P3.6	I/O O I/O I/O	Port3 is an 7 bit quasi-bidirectional port plus a 1-bit output port. For the masked version (80C752), the input level (TTL/CMOS) and the output structure (totempole/open-drain) of all the I/O can be individualy selected by mask during the processing. Port3 also contains the interrupt, timer, serial port, key detection and memory strobe pins : - P3.0 : RxD (serial input port) - P3.1 : TxD (serial output port) - P3.2 : INT0 (key detection), output only - P3.3 : INT1 (external interrupt) - P3.4 : T0 (timer 0 external input) - P3.5 : T1 (timer 1 external input) - P3.6 : WR (external data memory write strobe) - P3.7 : RD (external data memory read strobe)
Xtal1	I	Input to the inverting amplifier that forms the oscillator. Receives the external oscillator signal when an external oscillator is used.
Xtal2	O	Output of the inverting amplifier that forms the oscillator, and input to the internal clock generator. This pin should be floated when an external oscillator is used.
Vss		Ground (Ov).
P2.0 P2.7	I/O	Port2 is an 8 bit port. For the 80C732. Port2 emits the high-order address during code fetches and external memory accesses. For the 80C752, Port2 is an 8 bit port that can be used as a scanning output port. Port2 is also used for the high order address and the control signals during program verification.
$\overline{PSEN}$	O	The Program Store Enable output is a control signal that enables the external program memory fetch operations. It is activated every six oscillator periods except during to the bus during external data memory accesses. Remains high during internal program execution.
ALE	O	Provides address latch enable output used for latching the adress into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access at which time one ALE pulse is omitted.
$\overline{EA}$	I	When EA is held high, the CPU executes out of internal program memory (unless the program counter exceeds 4096). When EA is held low, the CPU executes only out of external program memory. EA must not be left floating. EA is internally connected to Vcc in the 40 pins DIL package.
P0.0-P0.7	I/O	Port0 is an 8 bit port. In the 80C732, Port0 is only the multiplexed low-order adress and data bus during accesses to external program or data memory. For the 80C752, Port0 is either an 8 bit output port, which can be used for scanning output, or the multiplexed address and data bus, depending on the state of GF0 flag bit (see 80C752' structure). It is also used during code program verification.
Vcc		Power supply (+ 5 V power supply).
Vss, Vssa, Vssb		Reference Ground.
RST	I	A high level on this pin for two machine cycles while the oscillator is running resets the device. An internal pull-down resistor permits Power-On reset using only a capacitor connected to Vcc.

PIN DEFINITIONS AND FUNCTIONS (continued)

SYMBOL	INPUT (I) OUTPUT (O)	FUNCTION
P1.0-P1.3 P1.4-P1.7	I/O I	Port1 is an 8 bit port. All bits can be programmed as analog inputs or quasibidirectional I/O, depending on the organization of the matrix. This programmability is possible only for the masked version (80C752). Default configuration is 8 bit analog input (80C732). P1-0 and P1.0, as quasibidirectional I/O can also serve Timer2 as follow : - P1.0 : T2 (external input to timer2) - P1.1 : T2EX (timer2 external trigger input)
Vref	I	Reference voltage for the mechanical interface or reference current for the capacitive interface : Vref set-up the value of the threshold of the resistive switch of the key for P1.0 to P1.7 of the 80C752-M or 80C732-M.
Iref	I	Reference current for the capacitive interface : Iref sets up the value of the threshold for the capacitive. Version : 80C752-C or 80C732-C.

DEVICE DESCRIPTION

INTRODUCTOIN

The 80C752 is a microcontroller designed to be used mainly in keyboards applications. The two different versions are :

- 80C752-M for mechanical keyboards
  - . direct contact
  - . resistive contact
- 80C752-C for capacitive keyboards
  - . switch capacitive
  - . capacitive

They can be provided in :

ROMless version : 80C732 with standard Input/Output port structure (see table 2 page 8),

MASKED ROM versions : 80C752 with customizable Input levels and Output port structure (see table 1 page 7).

80C752 OVERVIEW

This keyboard controller is derived from the MHS80C32 microcontroller and has the same instruction set than the 80C51 processor family. It has been designed for single-chip keyboard or control applications which use a keyboard. The structure of the I/O ports, the interrupt system and the I/O port addressing have been modified for those types of applications.

INPUT/OUTPUT PORTS

ORGANIZATION

The Input/Output ports have been modified and specialized to do different functions related to the keyboard control :

Port0, and Port2 pins are the scanning outputs, Port 1'pins are the return lines of the matrix, Port 3' pins are the high current I/O.

Standard configuration :

The 80C732 do not offers any options on the I/O configuration.

The standard configuration is shown in the table 2 page 8.

Custom configuration :

The 80C752 is also the masked version of the 80C732 which means that the ROM code is implemented on the 80C752 using a special mask. On this mask, MHS offers several options which allow the customer to personalize some I/O of Port1 and Port3 to eliminate the " glue logic " around the 80C752.

The first four I/O of Port1 (P1.0 to P1.3) can be individually selected as a quasi bi-directional port or as an input line from the matrix.

Each of the 7 high current I/O of Port3 can be entirely configured ; this means that the input level can be selected between CMOS or TTL and that the structure of the output stage can be selected between open-drain or quasi bi-directional.

The table 1 shows the characteristics of the I/O port of the 80C752.

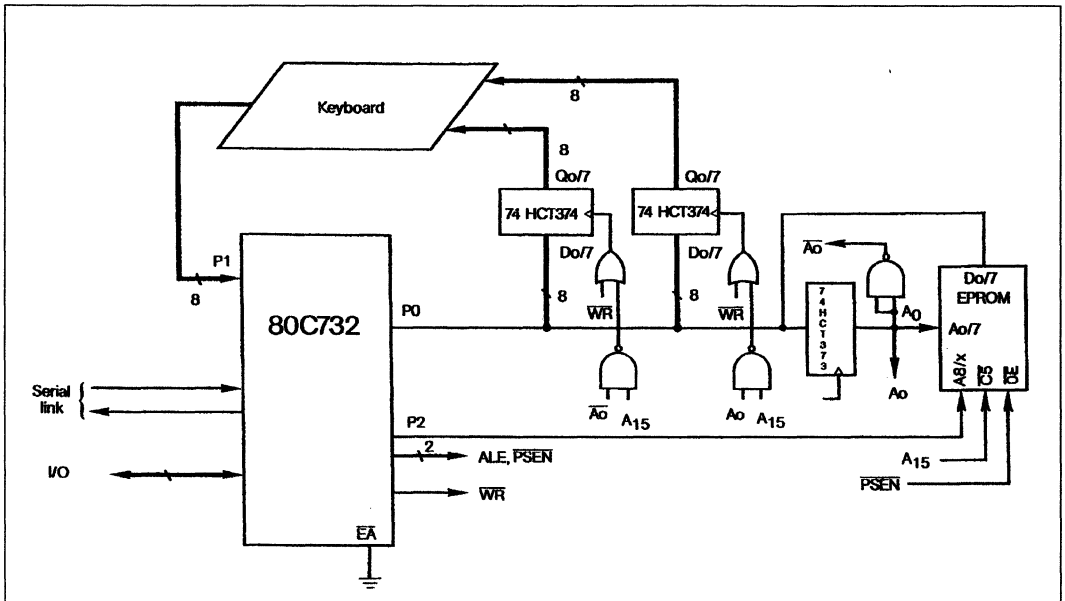
STRUCTURE

The structure of each port of the 80C752/80C732 is quite different of the 80C52/80C32's. Let us consider first the I/O structure of the 80C752 (80C752-C and 80C752-M).

The 80C752 can operate in two modes : the Keyboard controller Mode and the General Purpose Microcontroller Mode. The selection between these two modes in done with GFO, general purpose flag 0 in PCON (*), power control register.

1) The Keyboard Controller Mode : in this mode, GFO is set
   - in ROMLESS version PORT0 is a standard 80C52 Address/Data PORT, and PORT 2 is and Address/SFR Standard PORT.
     When using external ROM, some dedicated glue is necessary to recreat I/O PORT 0 and 2 (see figure 1). To have a complete software compatibility between ROMESS version and ROM version configuration, these 2 new external PORT 0 and 2, must be mapped at Address : FFFFh and FFFEh respectively.
   - in ROM version PORT 0 and PORT 2 are dedicated for the keyboard scanning output lines.
   - When the Controller executes a MOVX instruction :
     - no control signal si sent to PORT3 (this allows the use of P3.6 and P3.7 as an I/O PORT)
     - no Address Data are sent on PORT 0 and PORT 2.

2) The General Purpose Microcontroller Mode :
   In this Mode GFO is reset, and all functions are working as a standard 80C52.
   * Note : For more information, see MHS 80C51 user's manual.



Note : The two scanning ports are mapped in the external memory spaced of address OFFFEh and OFFFFh so that the software will be the same when switching to the ROMed controller.

Figure 1 : Keyboard Controller Mode in ROMLESS Version.

I/O	FUNCTION	STRUCTURE	OPTION
P0.0-P0.7	ADbus or scanning Out	# GF0 = 0 bidirectional ADbus # GF0 = 1 scanning output : output Port	Soft Soft
P1.0-P1.3 P1.4-P1.7	I/O port or scanning In scanning In	Quasi bi-directional I/O with TTL input level Input from resistive or capacitive matrix Input from resistive or capacitive matrix	Mask Mask
P2.0-P2.7	Scanning Out	Scanning output : output port only	
P3.0/RxD	High current I/O	CMOS or TTL input level Open-drain or quasi bi-directional I/O port - IOL = 10 mA @ 0.45 Volt	Mask Mask
P3.1/TxD	High current I/O	CMOS or TTL input level Open-drain or quasi bi-directional I/O port - IOL = 12 mA @ 0.45 Volt	Mask Mask
P3.2/INT0	Key-pressed output signal	Output activated (at 0) when a key is pressed in the scanned line.	
P3.3/INT1	High current I/O	CMOS or TTL input level Open-drain or quasi bi-directional I/O port - IOL = 10 mA @ 0.45 Volt	Mask Mask
P3.4/T0	High current I/O	CMOS or TTL input level Open-drain or quasi bi-directional I/O port - IOL = 3.2 mA @ 0.45 Volt can be used as an 17th scanning Out.	Mask Mask
P3.5/T1	High current I/O	CMOS or TTL input level Open-drain or quasi bi-directional I/O port - IOL = 10 mA @ 0.45 Volt	Mask Mask
P3.6/WR	High current I/O	CMOS or TTL input level Open-drain or quasi bi-directional I/O port - IOL = 10 mA @ 0.45 Volt No WR pulse in single-chip mode	Mask Mask
P3.7/RD	High current I/O	CMOS or TTL input level Open-drain or quasi bi-directional I/O port - IOL = 10 mA @ 0.45 Volt No RD pulse in single-chip mode	Mask Mask

Table 1 : 80C752 (-M or -C version).

80C732's STRUCTURE

The 80C732 has no on-chip ROM ; as the 80C32, it uses Port0 and Port2 to fetch opcodes and to access external data memory (RAM or I/O port mapped in the external memory space). But, as the chip has no masked-ROM, no options are possible on Port1 and Port2.

Port1 operation is the same as 80C752's. Port3' I/O lines operate as Port3 of 80C32.

The table 2 shows the characteristics of the I/O port of the 80C732.

I/O	FUNCTION	STRUCTURE	OPTION
P0.0-P0.7	ADbus	Bidirectional ADbus	
P1.0-P1.3 P1.4-P1.7	scanning In	Input from resistive or capacitive matrix	
P2.0-P2.7	Add bus	A8-A15 adress bus	
P3.0/RxD	High current I/O	TTL input level quasi bi-directional I/O port - IOL = 10 mA @ 0.45 Volt	
P3.1/TxD	High current I/O	TTL input level quasi bi-directional I/O port - IOL = 12 mA @ 0.45 Volt	
P3.2/INT0	Key-pressed output signal	Output activated (at 0) when a key is pressed in the scanned line.	
P3.3/INT1	High current I/O	TTL input level quasi bi-directional I/O port - IOL = 10 mA @ 0.45 Volt	
P3.4/T0	High current I/O	TTL input level quasi bi-directional I/O port - IOL = 3.2 mA @ 0.45 Volt	
P3.5/T1	High current I/O	TTL input level quasi bi-directional I/O port - IOL = 10 mA @ 0.45 Volt	
P3.6/WR	High current I/O	TTL input level quasi bi-directional I/O port - IOL = 10 mA @ 0.45 Volt WR pulse during each WRITE instruction	
P3.7/RD	High current I/O	TTL input level quasi bi-directional I/O port - IOL = 10 mA @ 0.45 Volt RD pulse during each READ instruction	

Table 2 : 80C752 (-M or -C version).

MECHANICAL KEYBOARDS

80C752-M/80C732-M

The operation of the mechanical keyboards is based on the use of contacting switches. These switches can be elastomer-dome, mechanical, membrane and snap-dome ; they rely on basically the same hardware and software techniques.

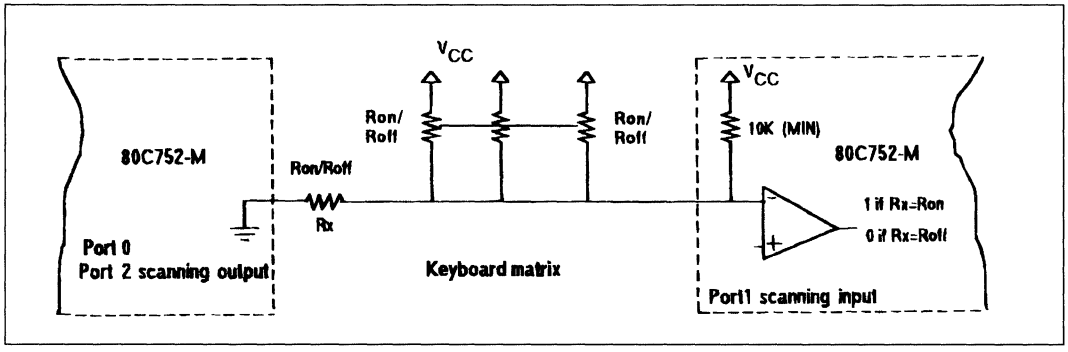
The 80C752-M and the 80C732-M are designed to directly interface these types of mechanical keyboards, whatever the types of the contacting switches.

THE ON-CHIP ANALOG INTERFACE

PRINCIPLE

The measurement is based on a voltage comparison between the selected threshold voltage and the voltage coming from the matrix :

The output voltage from the matrix may depend on the state of the non-scanned keys. The user must take this into account when selecting the threshold voltage.



SWITCH ROLLOVER TECHNIQUES

Depending on the keyboard's technology and its application, designers can use several rollover or validation schemes (defined as the number of keys that the keyboard circuit can process as closed in the correct sequence at the same time). The most common types of roll-over in use today are N-key lockout, two-key rollover, three-key rollover, and N-key rollover.

The 80C752-M and the 80C732-M allow all these different schemes. The number of roll-over is determined by the value of the ON-resistor and the presence of a blocking diode.

- without blocking diode.

If the ON-resistor is less than 2 kΩ, 16 keys can be pressed at the same time on a same column (16 Key Rollover). To avoid risk of phantom key only one column must be activated at the same time.

- with blocking diode.

Adding a blocking diode at each switch location eliminates phantom key closures and provides current protection with low ON-resistor switches. This technique also enables the use of the N-key rollover scheme, whatever the characteristics of the switch. The blocking diode is mandatory for keyboards with mechanical switches.

SCANNING TECHNIQUE

The scanning consists of resetting one of the drive lignes high and reading the state of the voltage comparators. Only one output at a given time can be active (low level) ; this means that between two different active states.

THRESHOLD SETTING

To accomodate different types of contacting switches, the user can adjust the threshold voltage of the 80C752-M/80C732-M by adjusting the input voltage on Vref pin.

KEY PRESSED DETECTION

The 80C752-M and the 80C732-M provide hardware detection of a pressed key. This information (state of the P3.2 pin) can be used as a flag (state read by software) or as an interrupt source, if INT0 is enabled. The state of P3.2 is updated after the start of every new scanning ; this is the reason why INT0 must be edge triggerred (bit IT0 in TCON must be set).

6

DEVELOPMENT

The 80C752-M has been designed so that any program developed for the 80C732-M, with the two output scanning ports mapped at external memory addresses OFFFEh and OFFFFh, can be exactly the same for the 80C752-M. This is the reason why Port0 and Port2 of the 80C752 are mapped in the SFR space and in the external memory space.

EMULATION

For the software and/or hardware debugging, any 80C52 emulator can be used to emulate the 80C752-M, but the user has to add some external "glue logic" around the emulator probe to build the mechanical interface of the 80C752-M and the drivers integrated in Port3 pins. Hereafter is the schematic of the mechanical interface which must be added between the 8 outputs from the matrix and the 8 pins of Port1 of the emulator :

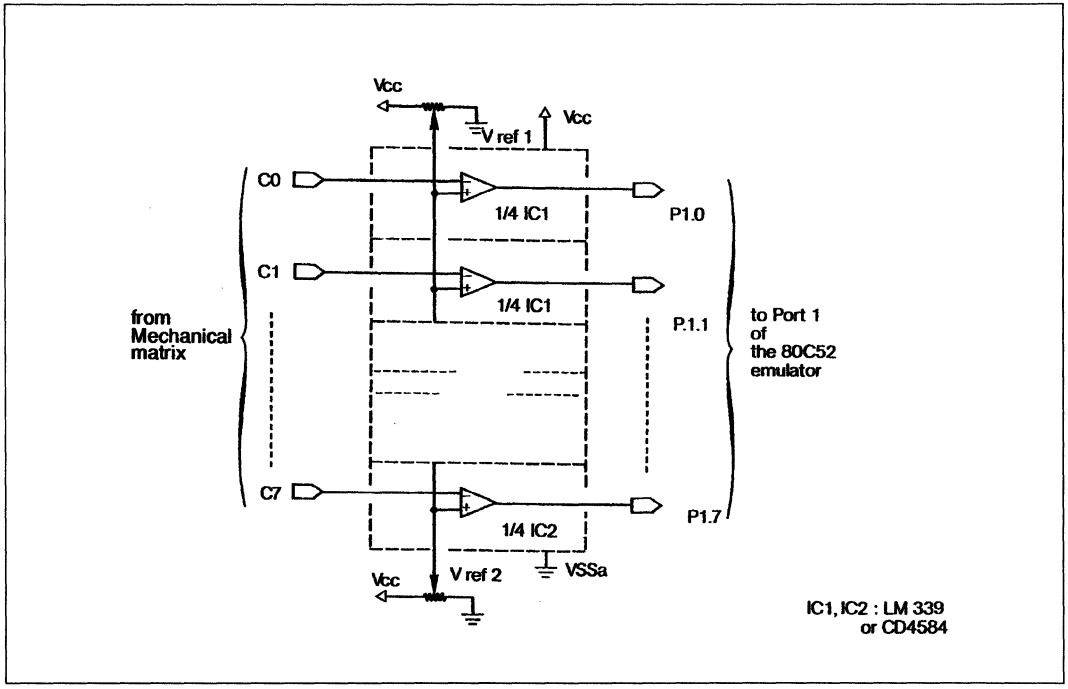


Figure 2 : Mechanical Interface Schematic.

6

80C732-M EMULATION

For 80C732-M emulation, the user has to provide one or two external ports, to be mapped in the external memory space for the scanning output port. It is recommended to map these ports at the addressed OFFFEh and OFFFFh because when going from ROMless to ROMed version, the software will remain the same. This is the complete emulation schematic :

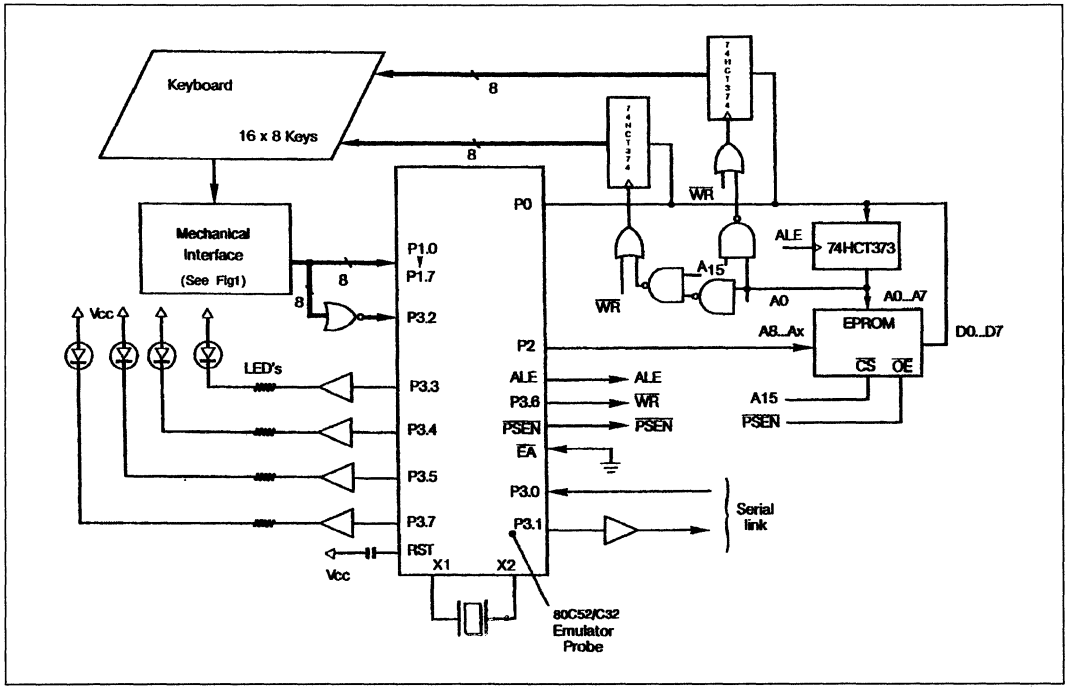


Figure 3 : 80C732-M Emulation Schematic.

6

80C752-M EMULATION

For 80C752-M emulation, it is not useful to build the external output ports (and it is not recommended) because the SFR and bit addressing of the scanning ports (P0 and P2) is preferable when using the external memory addressing and also because this allows the use of P3.6 as an I/O.

The below figure shows the complete emulation schematic :

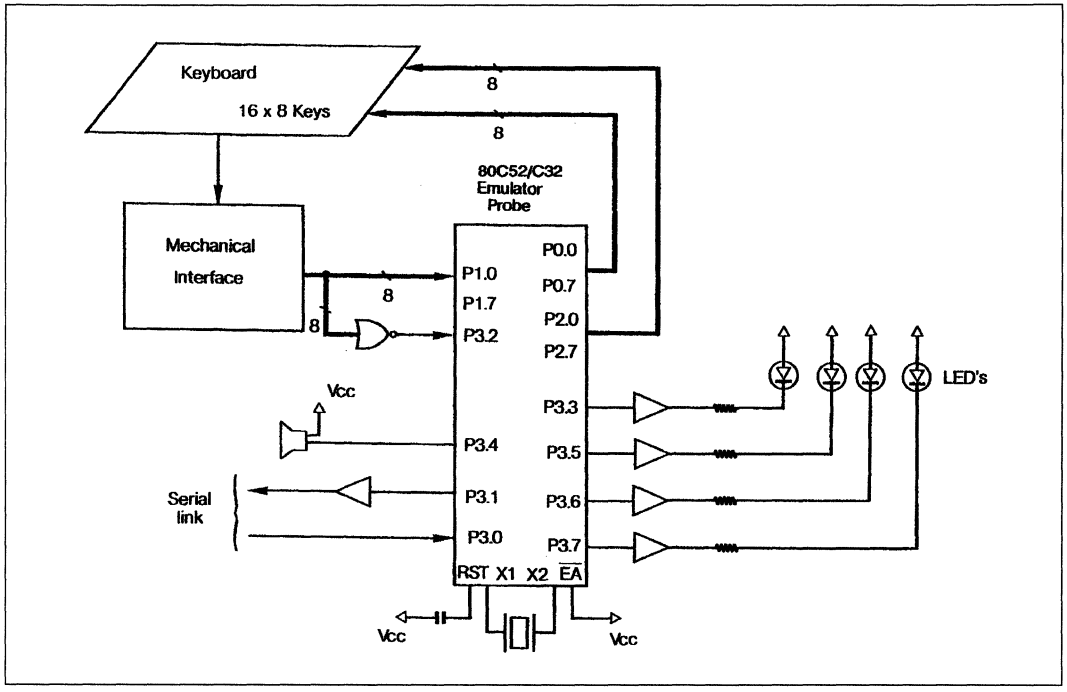


Figure 4 : 80C752-M Emulation Schematic.

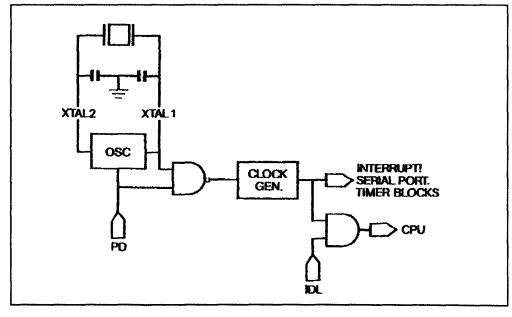
6

IDLE AND POWER-DOWN OPERATION

As shown in the below figure, the idle and power-down modes are the same than with 80C51 (see MHS 80C51 user's manual). As illustrated, Power-down operation stops the oscillator and idle mode operation allows the interrupt, serial port, and timer blocks to continue to function while the clock to the CPU is gated off.

During power-down, the analog interface is powered-off to minimize circuit power consumption. The only way to escape from power-down mode is to reset the CPU.

During idle, if any key is pressed, P3.2 will fall down and, if INT0 is enabled, the CPU will escape from idle mode.



CAPACITIVE KEYBOARDS

80C752-C/80C732-C

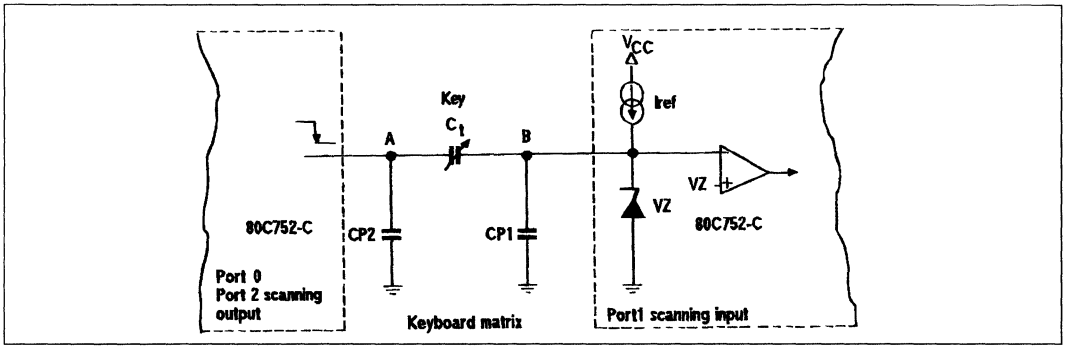
There are 2 types of capacitive keys : the full capacitive type and the switch-capacitor type.

The 80C752-C and the 80C732-C are designed to directly interface these two types of capacitive keyboards, independently of the state of each capacitive key.

THE ON-CHIP ANALOG INTERFACE

PRINCIPLE

The 80C752-C and the 80C732-C allow the N-Key rollever technique ; this means that they can detect a key independently of the state of the other keys and of the technology of the matrix (flexprint...).
The measurement is based on a constant current charge of the capacitor of the key :



At the falling edge of the scanning output, the voltage at node B goes down below the regulation voltage of the " zener ".

The voltage decreasing virtually disconnect the regulator and all the current from the current generator goes to the capacitors.

So, the capacitors are charged with a constant current. At the end of the process, the voltage value is the same as before the falling edge ; this means that Cp1 has no influence on the time to charge Ct the capacitor of the key :

$$t = Va \times Ct/Iref \Rightarrow Ct = (t \times Iref)/Va$$

The voltage driven technique on the node A eliminates the influence of Cp2.

Cp1 and Cp2 have no influence on Ct measurement ; this allows the 80C752-C and the 80C732-C to be used in design with N-key rollover technique. They can fit with a wide range of capacitive matrices because the on-chip hardware allows the selection of the capacitor threshold value.

SCANNING TECHNIQUE

The scanning must be done with a rolling zero on the scanning outputs (the other outputs remain high). Only one output at a given time can be active (low level) ; this means that between two different active states all scanning outputs must be desactivated.

THRESHOLD SETTING

The formula which determines the operation of the capacitive interface is :

Ct = (t x Iref)/Va

- Ct, is the capacitor threshold value,
- Va, which is the value of the voltage variation on the scanning output, Vcc - 0.5 V,
- t, determines the moment when the output of the comparator is strobed (0 if the capacitor is recharged, 1 if not recharged).

Iref, which is the value of the constant current driving Ct. It can be adjusted by an external resistor. This allows the 80C752-C and the 80C732-C to operate with different types of capacitive matrixes. Iref can be fixed with only one resistor ; the formula to select Iref is Iref = (Vcc – 1.5 V)/R, where R is the value of the external resistor.

KEY PRESSED DETECTION

The 80C752-C and the 80C732-C provide an internal hardware detection of pressed key. This signal is output on P3.2 and internally connected to INT0. This information (state of P3.2) can be used as a flag (state read by software) or as an interrupt source, if INT0 is enabled. This is the reason why INT0 must be edge triggered (bit IT0 in TCON must be set).

DEVELOPMENT

The 80C752-C has been designed so that any program developed for the ROMless, with the two output scanning ports mapped at external memory address 0FFFEH and 0FFFFh, can be exactly the same for the 80C752-C. This is the reason why Port0 and Port2 of the 80C752 are mapped in the SFR space and in the external memory space.

EMULATION

For software and/or hardware debugging, any 80C52 emulator can be used to emulate the 80C752-C, but the user has to add some external " glue logic " around the emulator probe to build the capacitive interface of the 80C752-C and the drivers integrated in port3 pins. Hereafter is the schematic diagram of the capacitive interface which mut be added between the 8 inputs from the matrix and the 8 pins of Port1 of the emulator :

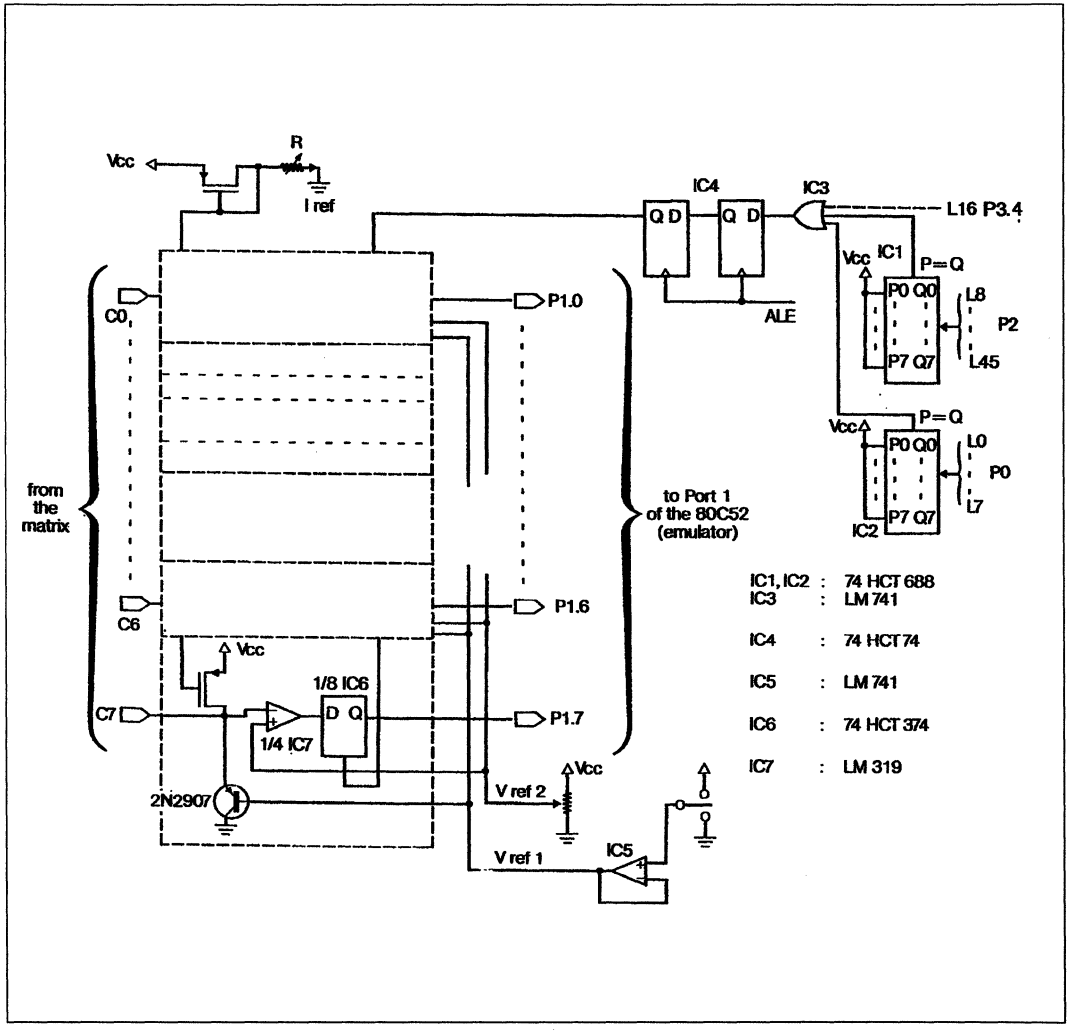


Figure 5 : Capacitive Interface Schematic.

80C732-C EMULATION

For 80C732-C emulation, the user has to provide one or two external ports, to be mapped in the external memory space for the scanning output port. It is recommended to map these ports at the addresses 0FFFEh and 0FFFFh because when going from ROMless to ROMed version, the software will remain the same. Here is the complete emulation schematic :

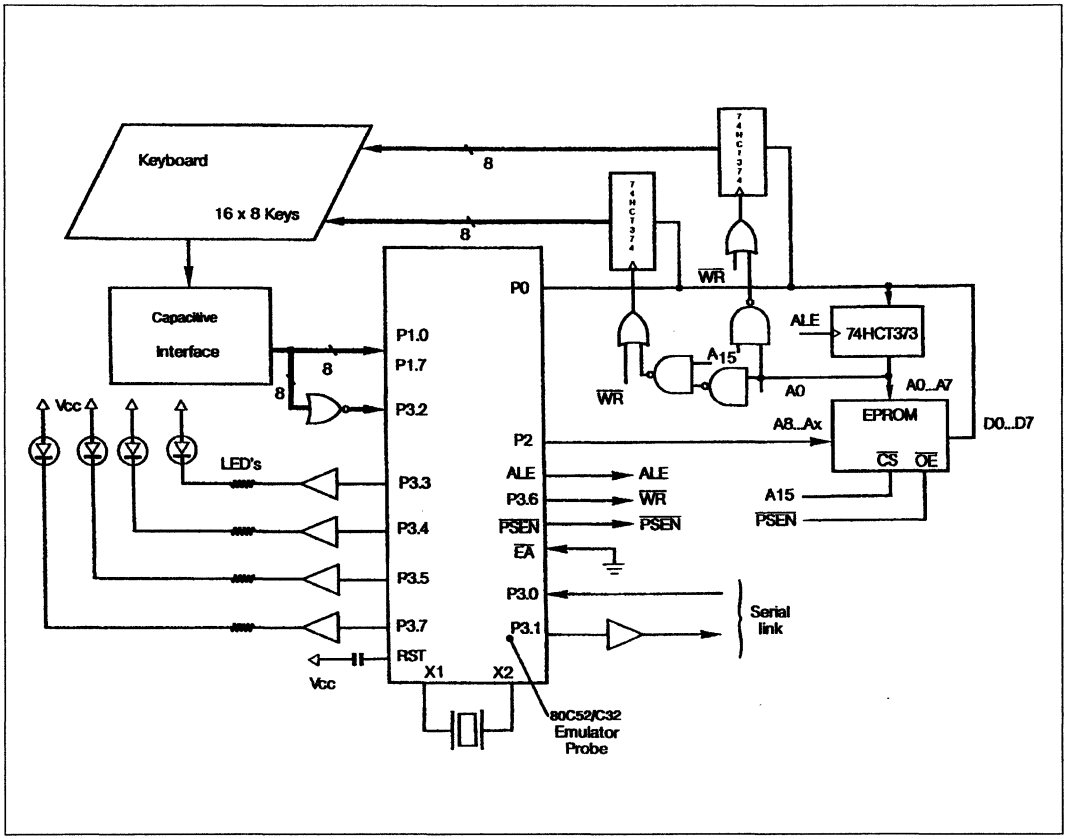


Figure 6 : 80C732-C Emulation Schematic.

80C752-C EMULATION

For 80C752-C emulation, there is no need to build the external output ports. However it is not recommended since the SFR and bit addressing of the scanning ports (P0 and P2) is better than the external memory addressing and also because this allows the use of P3.6 (WR) as an I/O. The below figure shows the complete emulation schematic :

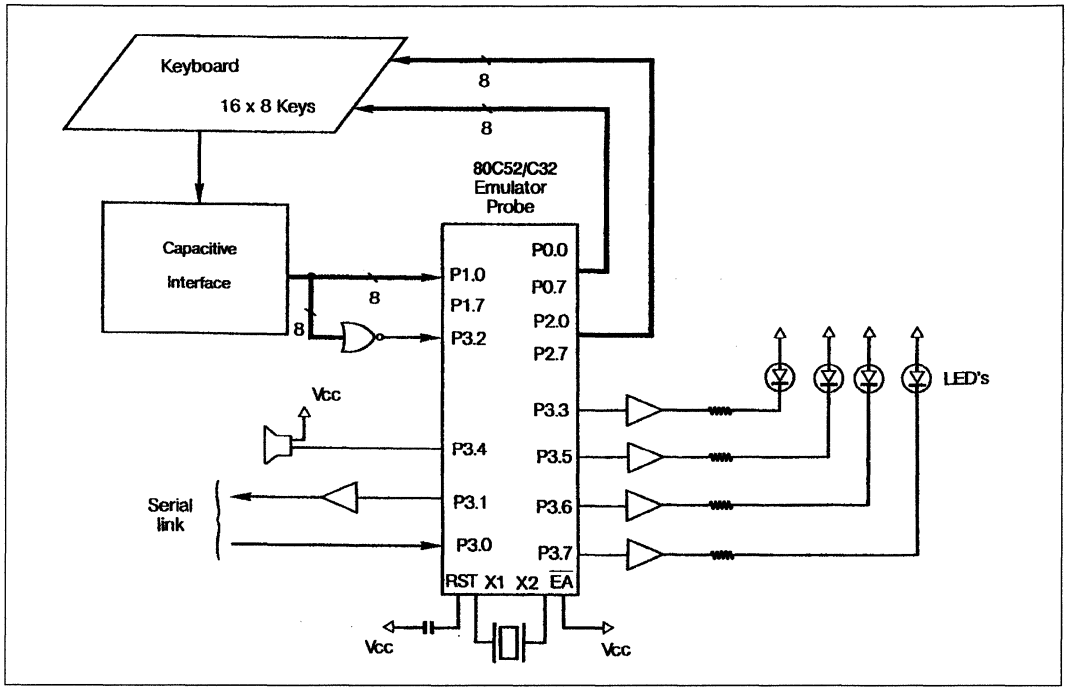


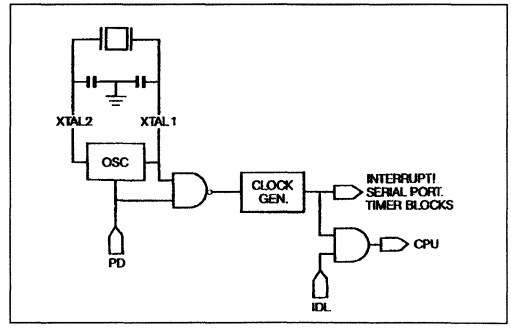
Figure 7 : 80C752-C Emulation Schematic.

6

IDLE AND POWER-DOWN OPERATION

As shown in the figure below, the idle and power-down modes are the same as for 80C51 (see MHS 80C51 user's manual). As illustrated, Power-down operation stops the oscillator and Idle mode operation allows the interrupt, serial port, and timer blocks to continue to function while the clock to the CPU is gated off.

During power-down the capacitive interface is powered-off to minimize circuit power consumption. P3.2 remains in the same state until a new scanning value is sent ; if any key is pressed during Idle or Power-down, it will not change the state of P3.2 and the circuit will stay in the same mode.



SOFTWARE CONSIDERATIONS

The 80C752 has been built to simplify the design of keyboards and thus allowing the user to concentrate on the real problems such as keys organisation, type of matrix, design of the keys and the box of the keyboard...

As a result, only one software has to be written by the user. The same software can be used for :

- ROMless devices,
- ROMed devices,
- mechanical or resistive matrixes,
- capacitive matrixes.

Moreover MHS can also provide a masked version for IBM-PC (*) compatible keyboards, the PCK52, and the assembly source program of the PCK52, named PCS52.

1. MECHANICAL OR RESISTIVE MATRIXES. The scanning is based on the static level of each input : at any moment, the state of P3.2 and the data read in SFR Port1 indicate the real state of the scanned-keys.

2. CAPACITIVE MATRIXES. The scanning is based on the dynaic level of each input of the matrix : the capacitor measurement until (integrated in Port1) is retriggered at each new scan generation (i.e. after the falling edge of a scan line) and for some micro-second only.

So, the state of P3.2 and the data read in SFR Port1 do not indicate at every moment the state of the scanned-keys.

Design considerations :

The data in Port1, which should be normally strobed 3 µs after a new scan generation, are not strobed on this design.

The consequences are :

- P3.2 must not be used for software tests and for interrupt generation.
- The 3 µs delay must be done by software,
- During the active time of the scanning, all interrupts must be masked to ensure that the 3 µs software delay will be always the same.
- The data in SFR Port1 must be read immediatly after the 3 µs delay.

3. EXAMPLE : Hereafter is an example of a scanning routine which can be used for capacitive or resistive matrixes and for ROMed or ROMless devices.

(*) IBM is a trademark of International Business Machine Corporation.

```
SCANNING_ROUTINE :                          ; ASSUME (R7, R6) CONTAINS THE
                                            ; SCANNING VALUE
            SETB    C
            MOV     A, R6
            RLC     A                       ; ROLLING 0 THROUGH (R7, R6)
            MOV     R6, A
            MOV     PSW.5, C                ; STORE CARRY FLAG IN GF1
            CJNE    A, #0FFH, SCAN_LOW
            MOV     C, PSW.5
            MOV     A, R7
            RLC     A
            MOV     R7, A
            CJNE    A, #0FFH, SCAN_HIGH
            MOV     R6, #0FEH
SCAN_LOW :
            MOV     A, R6
            MOV     DPTR, #0FFFFH
            CLR     IE.7                    ; DISABLE ALL IT
            MOVX    @DPTR, A                ; SCAN : FALLING EDGE ON PO.X
            NOP
            NOP
            MOV     A, P1                   ; 3 µS DELAY BETWEEN SCAN AND SENSE
            JMP     TEST_DETECT
SCAN_HIGH :
            MOV     A, R7
            MOV     DPTR, #0FFFEH
            CLR     IE.7                    ; DISABLE ALL IT
            MOVX    @DPTR, A                ; SCAN : FALLING EDGE ON P2.X
            NOP
            NOP
            MOV     A, P1                   ; 3 µS DELAY BETWEEN SCAN AND SENSE
TEST_DETECT :
            SETB    IE.7                    ; RE-ENABLES ALL IT
            CPL     A
            JZ      NO_KEY
KEY-DETECTED :                              ; TEST FOR PUSHED KEYS
NO_KEY :                                    ; TEST FOR RELEASED KEYS
            RET
```

PCK52

The PCK52 has been specially designed for IBM-PC keyboards. It is supplied in 2 versions :

- PCK52-M for mechanical or resistive matrix,
- PCK52-C for capacitive matrix.

It can be used in PC-XT or PC-AT applications, depending of the state of switch 1 (SW1).
The Key-mapping is shown in table 3. The number given in this table designates the keybutton position, conforming to the IBM nomenclature for keyboards.

	P1.0	P1.1	P1.2	P1.3	P1.4	P1.5	P1.6	P1.7
P0.0	0	0	0	0	0	0	0	0
P0.1	53	60	52	58	61	62	56	54
P0.2	47	46	45	44	51	50	49	48
P0.3	33	32	31	30	37	36	35	34
P0.4	19	18	17	16	23	22	21	20
P0.5	4	3	2	1	8	7	6	5
P0.6	0	0	0	SW1	0	0	0	0
P0.7	114	113	112	110	118	117	116	115
P2.0	0	0	0	0	0	0	0	0
P2.1	91	96	101	103	0	42	93	98
P2.2	41	40	39	38	83	64	57	0
P2.3	108	106	105	104	84	79	89	99
P2.4	90	95	100	102	86	85	92	97
P2.5	27	26	25	24	81	76	43	28
P2.6	12	11	10	9	80	75	15	13
P2.7	122	121	120	119	126	125	124	123

SW1 : ON : PC-XT, OFF : PC-AT3.

Table 3 : Key Mapping of PCK52.

PCK52 MASK OPTIONS :

Hereunder are described the Mask options of the actual version of PCK52 for Mechanical Keyboard : 80C752M.290.

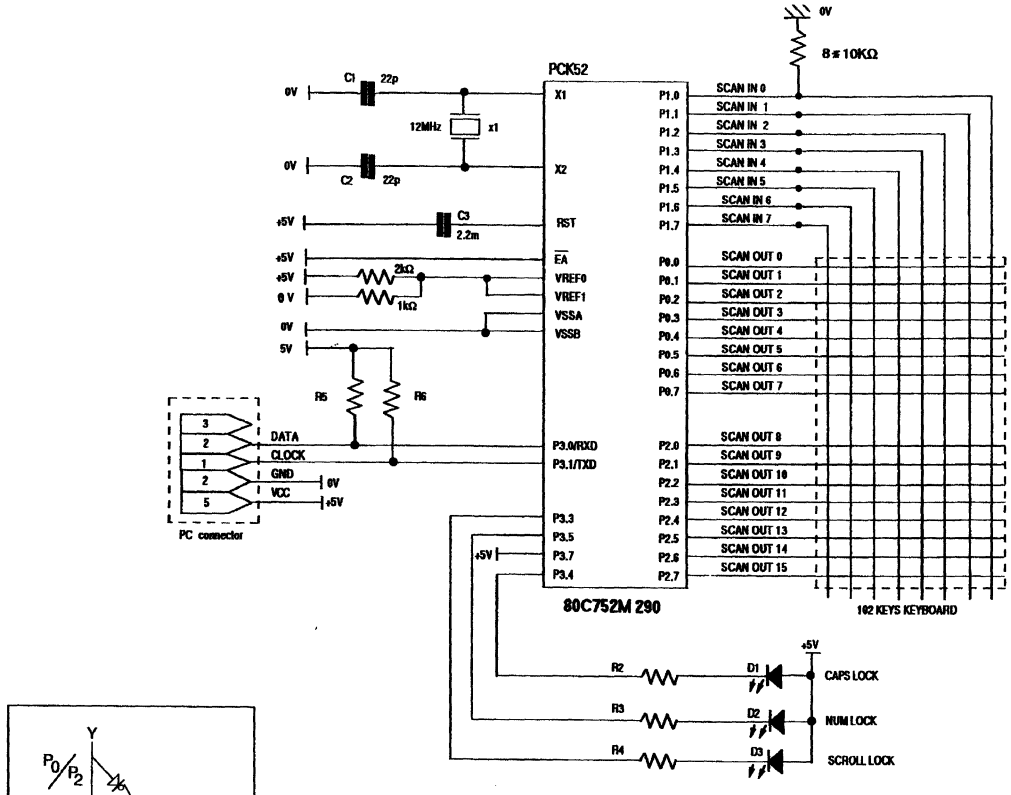
80C752-M 290 MASK OPTIONS

KEYBOARD MATRIX								
MECHANICAL				or	CAPACITIVE			
PORT 1 SELECTIONS	P1.0	P1.1	P1.2	P1.3	P1.4	P1.5	P1.6	P1.7
I/O port with TTL input	YES	YES	YES	YES	YES	YES	YES	YES

	INPUT SELECTION		OUTPUT SELECTION		
PORT 3 OPTIONS	**TTL (51 type)**	**CMOS**	**C51 TYPE (*)**	**OPEN DRAIN**	
P3.0/RXD	YES			YES	10 mA (max)
P3.1/TXD	YES			YES	12 mA (max)
P3.3/INT1	YES			YES	10 mA (max)
P3.4/T0		YES	YES		3.2 mA (max)
P3.5/T1		YES	YES		10 mA (max)
P3.6/WR		YES	YES		10 mA (max)
P3.7/RD		YES	YES		10 mA (max)

(*) : Quasi bi-directionnal high current @ 0.45 V.

The Hardware environment must be following :



Keyboard using scanning 1 method.

PCS52

The PSC52 is the assembly source of the PCK52 program.

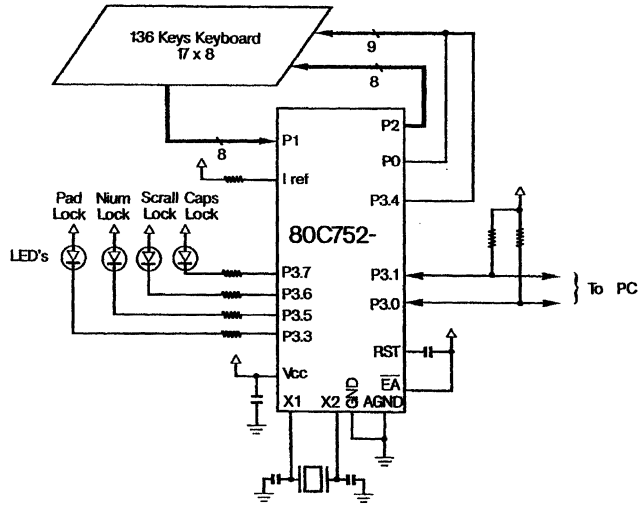
This software, fully documented, allows to change the main characteristics of a keyboard by modifing some parameters in the assembly is the source program :

- the key-mapping
- the 3 internal timings :
  - the debounce time,
  - the scanning rate,
  - the auto-repeat rate.
- the default type of the keys :
  - make/break,
  - typematic.
- the type of the scanning :
  - positive or negative.

This source file, in 8051 assembly code is provided in 1 floppy (5 or 3 inches) for MS-DOS compatible PC.

APPLICATION EXAMPLES

1. Keyboard controller for Personal Computer :



In this configuration, the 80C752-C can control keyboards with up to 136 keys.
The following options must be selected in order to drive the capacitive matrix and to comply with the DC specifications of the serial link with the PC :
Input levels :
P3.0 and P3.1 : TTL (0.8 V ; 2.4 V), selected by mask option.

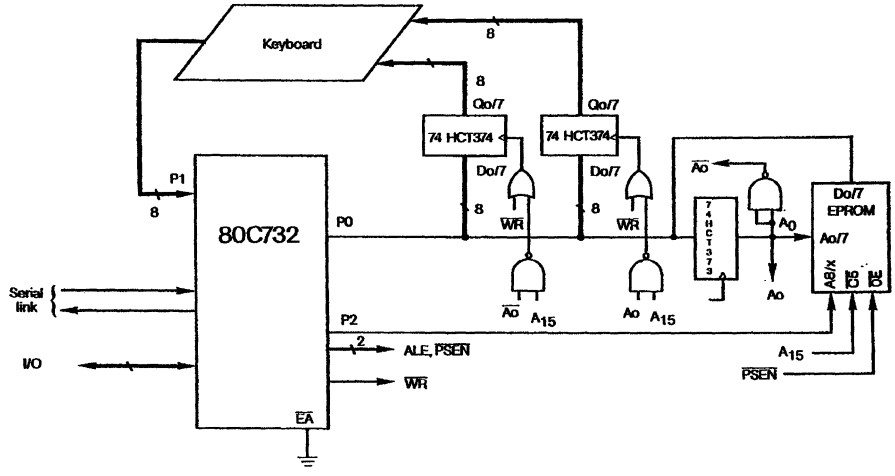
2. Keyboard controller with external EPROM :

Output structures :

P3.0, P3.1 : open drain, selected by mask option,

P3.4 : quasi bi-directional port, selected by mask option,

P3.3, P3.5, P3.7 : open drain, selected by mask option.



Note : The two scanning ports are mapped in the external memory space of address OFFFEh and OFFFFh so that the software will be the same when switching to the ROMed controller.

6

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias :
Commercial ..... – 0°C to 70°C
Storage Temperature ..... – 65°C to + 150°C
Voltage on V_{CC} to V_{SS} ..... – 0.5 to V_{CC} + 7 V
Voltage on any pin to V_{SS} ..... – 0.5 V to V_{CC} + 0.5 V
Power Dissipation ..... 200 mW

* NOTICE : Stresses at or above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

DC CHARACTERISTICS

TA = 0°C to + 70°C ; V_{SS} = 0 V ; V_{CC} = 5 V ± 10 %

SYMBOL	PARAMETER	MIN.	MAX.	UNIT	TEST CONDITIONS
VIL	Input Low Voltage	– 0.5	0.2 V_{CC} – 0.1	V	
VIH	Input High Voltage (Except XTAL1 and RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V	
VIH1	Input High Voltage (RST and XTAL1)	0.7 V_{CC}	V_{CC} + 0.5	V	
VOL	Output Low Voltage (Ports 1, 2, 3)		0.45	V	IOL = 1.6 mA (note 2)
VOL1	Output Low Voltage Port 0, ALE, $\overline{PSEN}$, P3.4, Port 2		0.45	V	IOL = 3.2 mA (note 2)
VOL2	Output Low Voltage P3.3, P3.5, P3.6, P3.7		0.45	V	IOL = 10 mA (note 2)
VOL3	Output Low Voltage P3.0, P3.1		0.45	V	IOL = 12 mA (note 2)
VOH	Output High Voltage Ports 1, 2, 3	0.9 V_{CC}		V	IOH = – 10 µA
		0.75 V_{CC}		V	IOH = – 25 µA
		2.4		V	IOH = – 60 µA V_{CC} = 5 V ± 10 %
VOH1	Output High Voltage (Port 0 in External Bus Mode, ALE, $\overline{PSEN}$)	0.9 V_{CC}		V	IOH = – 40 µA
		0.75 V_{CC}		V	IOH = 150 µA
		2.4		V	IOH = – 400 µA V_{CC} = 5 V ± 10 %
IIL	Logical 0 Input Current Ports 1, 2, 3		– 50	µA	Vin = 0.45 V
ILI	Input Leakage Current (Port 0, $\overline{EA}$)		± 10	µA	0.45 < Vin < V_{CC}
ITL	Logical 1 to 0 Transition Current (Ports 1, 2, 3)		– 650	µA	Vin = 2.0 V
IPD	Power Supply Current (Power Down Mode)		100	µA	V_{CC} = 2.0 V to 5.5 V (note 1)
RRST	RST Pulldown Resistor	50	150	kΩ	
CIO	Capacitance of I/O Buffer		10	pF	fc = 1 MHz TA = 25°C
VREF	Reference Voltage	1.0	V_{CC} – 1.0	V	80C752-M/80C732-M
RVREF	Source Impedance of VREF	1.0		MΩ	80C752-M/80C732-M
IRef	Current reference for current minors	10	500	µA	80C752-C/80C732-C

6

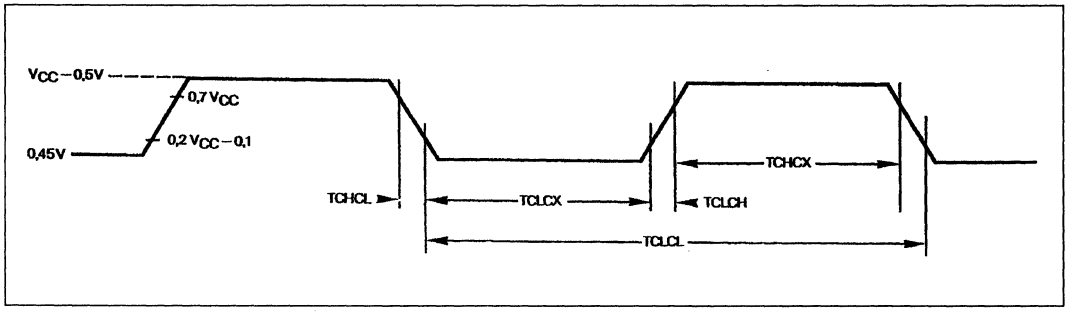


Figure : Clock Signal Waveform for ICC Tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns.

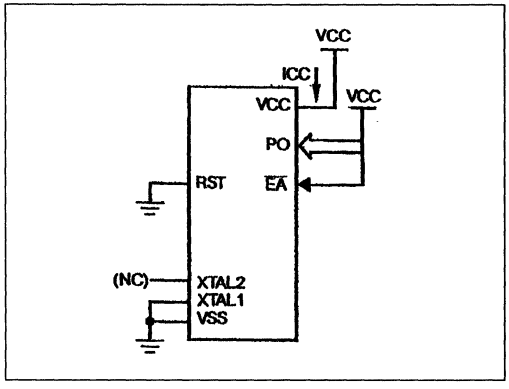


Figure : ICC Test Condition, Power Down Mode.
All other pins are Disconnected.

Note : 1. ICC is measured with all output pins disconnected ; XTAL1 driven with TCLCH, TCHCL = 5 ns, VIL = VS + 5 V, VIH = VCC − .5 V ; XTAL2 N.C. ; EA = RST = Port 0 = VCC. ICC would be slightly higher if a crystal oscillator used.

Idle ICC is measured with all output pins disconnected ; XTAL1 driven with TCLCH, TCHCL = 5 ns, VIL = VSS + .5 V, VIH = VCC − .5 V ; XTAL2 N.C. ; Port 0 = VCC ; EA = RST = VSS.

Power down ICC is measured with all output pins disconnected ; EA = PORT 0 = VCC ; XTAL2 N.C. ; RST = VSS.

Note : 2. Capacitance loading on Ports 0 and 2 may cause spurious noise, pulses to be superimposed on the VOLS of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1 to 0 transistions during bus operations. In the worst cases (capacitive loading 100 pF), the noise pulse on the ALE line may exceed 0.45 V with maxi VOL peak 0.6 V. A Schmitt Trigger use is not necessary.

6

EXTERNAL CLOCK DRIVE CHARACTERISTICS (XTAL1)

SYMBOL	PARAMETER	VARIABLE CLOCK FREQ = 0 to 12 MHz		UNIT
		MIN.	MAX.	
1/TCLCL	Oscillator Frequency	83		ns
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

AC PARAMETERS

TA + 0°C to 70°C ; VSS = 0 V ; VCC = 5 V ± 10 % (commercial)
(load capacitance for Port 0, ALE ; and $\overline{PSEN}$ = 100 pf ; load capacitance for all other outputs = 80 pf).

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

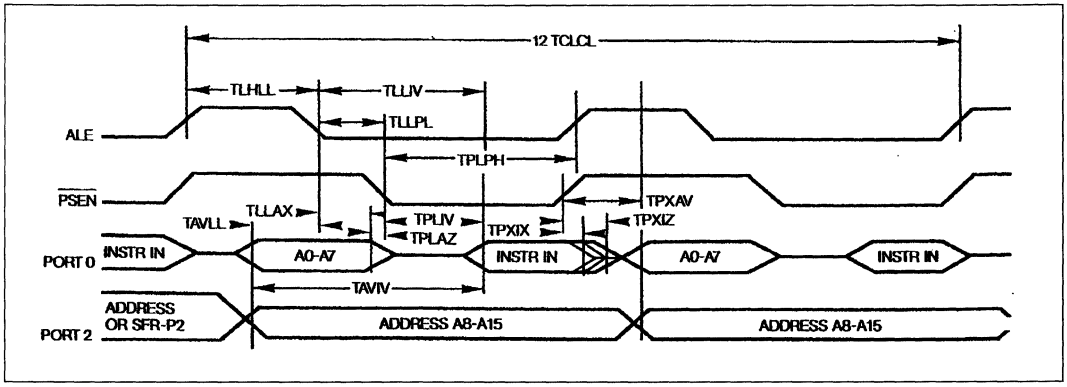
SYMBOL	PARAMETER	MIN.	MAX.	UNIT
TLHLL	ALE Pulse Width	2TCLCL-40		ns
TAVLL	Address Valid to ALE	TCLCL-55		ns
TLLAX	Address Hold After ALE	TCLCL-35		ns
TLLIV	ALE to Valid Instr in		4TCLCL-110	ns
TLLPL	ALE to $\overline{PSEN}$	TCLCL-40		ns
TPLPH	$\overline{PSEN}$ Pulse Width	3TCLCL-45		ns
TPLIV	$\overline{PSEN}$ to Valid Instr in		3TCLCL-105	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		TCLCL-25	ns
TPXAV	$\overline{PSEN}$ to Address Valid	TCLCL-8		ns
TAVIV	Address to Valid Instr in		5TCLCL-105	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		10	ns

EXTERNAL DATA MEMORY CHARACTERISTICS

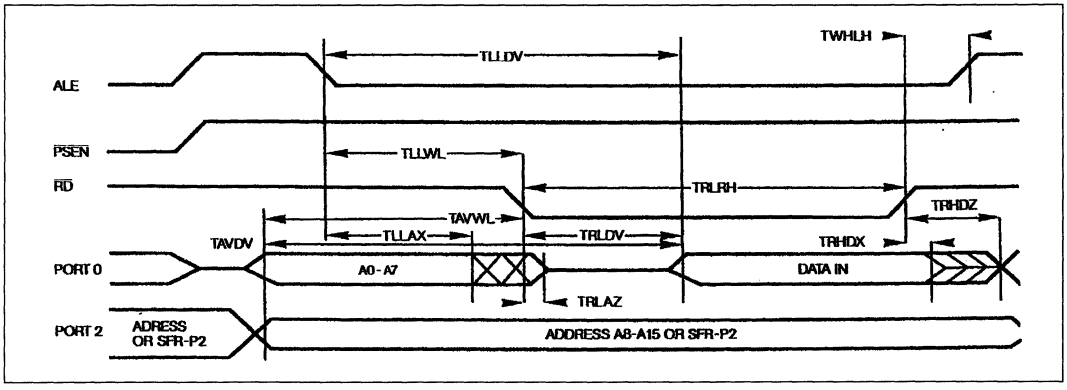
SYMBOL	PARAMETER	MIN.	MAX.	UNIT
TRLRH	$\overline{RD}$ Pulse Width	6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	6TCLCL-100		ns
TLLAX	Data Address Hold After ALE	TCLCL-35		ns
TRLDV	$\overline{RD}$ to Valid Data in		5TCLCL-165	ns
TRHDX	Data Hold After $\overline{RD}$	0		ns
TRHDZ	Data Float After $\overline{RD}$		2TCLCL-70	ns
TLLDV	ALE to Valid Data in		8TCLCL-150	ns
TAVDV	Address to Valid Data in		9TCLCL-165	ns
TLLWL	ALE to $\overline{WR}$ or $\overline{RD}$	3TCLCL-50	3TCLCL+50	ns
TAVWL	Address to $\overline{WR}$ or $\overline{RD}$	4TCLCL-130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	TCLCL-60		ns
TQVWH	Data Setup to $\overline{WR}$ High	7TCLCL-150		ns
TWHQX	Data Hold After $\overline{WR}$	TCLCL-50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	TCLCL-40	TCLCL+40	ns

AC TIMING DIAGRAMS

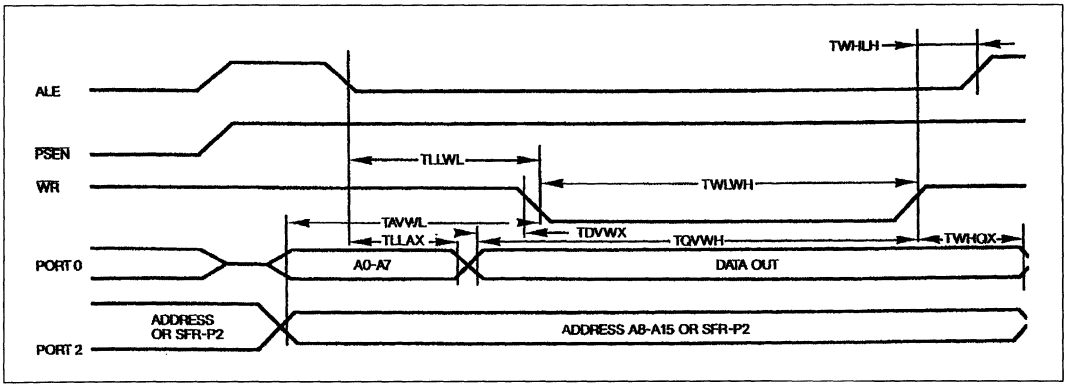
EXTERNAL PROGRAM MEMORY READ CYCLE



EXTERNAL DATA MEMORY READ CYCLE

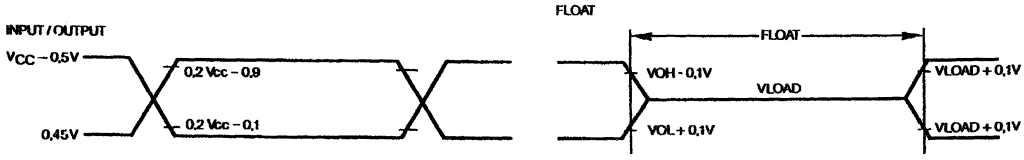


EXTERNAL DATA MEMORY WRITE CYCLE



6

AC TESTING INPUT/OUTPUT, FLOAT WAVEFORMS

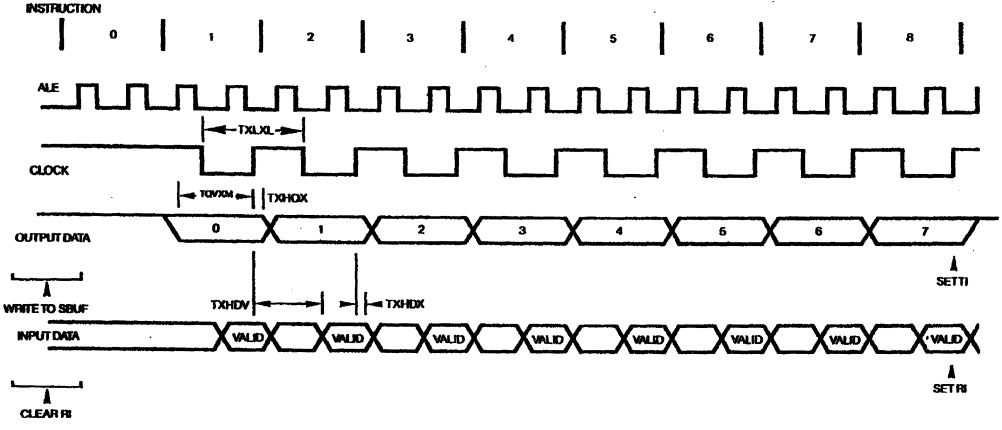


AC inputs during testing are driven at V_{CC} – 0.5 for a logic " 1 " and 0.45 V for a logic " 0 ". Timing measurements are made at VIH min for a logic " 1 " and VIL max for a logic "0".For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change from the loaded VOH/VOL level occurs. Iol/IoH ≥± 20 Ma.

SERIAL PORT TIMING - SHIFT REGISTER MODE

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
TXLXL	Serial Port Clock Cycle Time	12TCLCL		µs
TQVXH	Output Data Setup to Clock Rising Edge	10TCLCL-133		ns
TXHQX	Output Data Hold after Clock Rising Edge	2TCLCL-117		ns
TXHDX	Input Data Hold after Clock Rising Edge	0		ns
TXHDV	Clock Rising Edge to Input Data Valid		10TLCL-133	ns

SHIFT REGISTER TIMING WAVEFORMS



EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a " T " (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

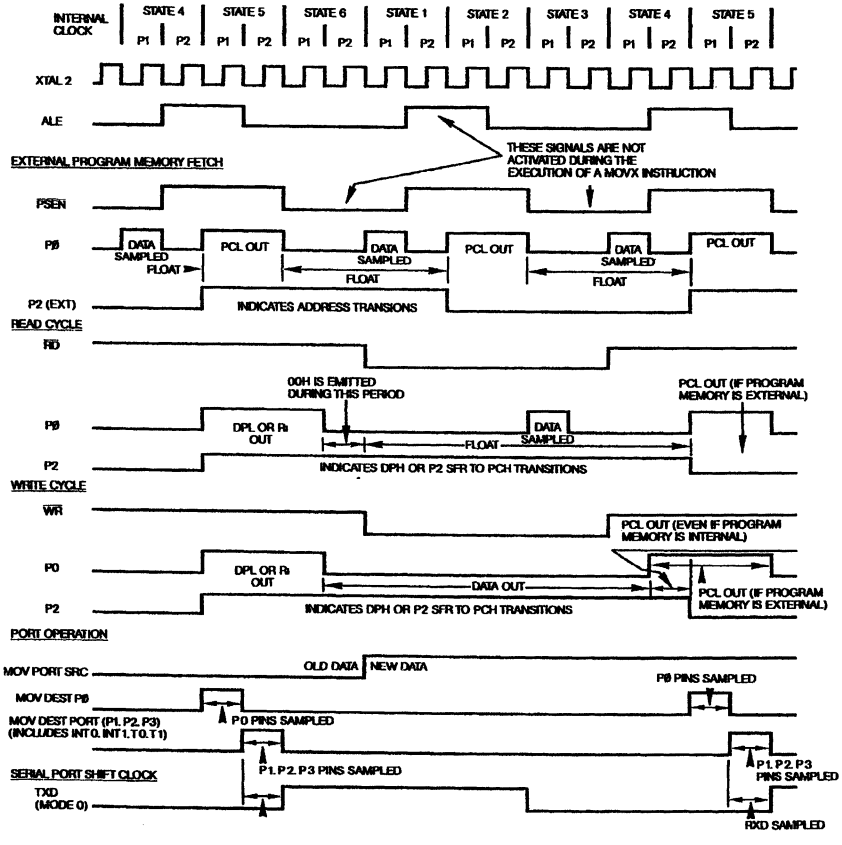
EXAMPLE :

TAVLL = Time for Address Valid to ALE low.

TLLPL = Time for ALE low to $\overline{\text{PSEN}}$ low.

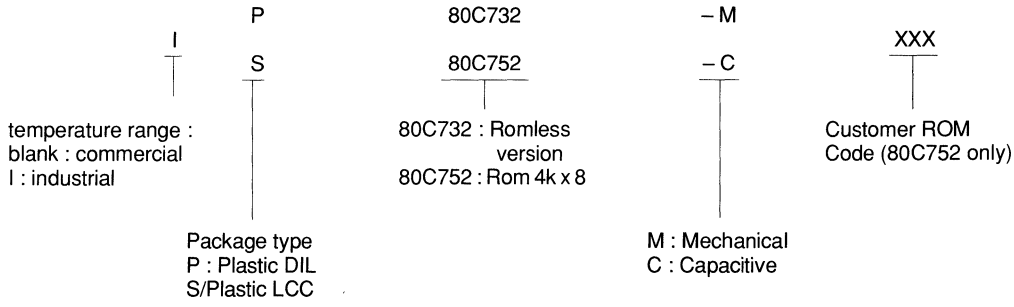
A : Address	Q : Output data
C : Clock	R : READ Signal
D : Input Data	T : Time
H : Logic Level HIGH	V : Valid
I : Instruction (program memory contents)	W : WRITE Signal
L : Logic Level LOW, or ALE	X : No Longer a Valid Logic Level
P : PSEN	Z : Float

CLOCK WAVEFORMS



This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins however ranges from 25 to 125 ns. This propagation delay is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component to component. Typically though (TA = 25°C, fully loaded) RD and WR propagation delays are approximately 50 ns. The other signal are typically 85 ns. Propagation delays are incorporated in the AC specifications.

ORDERING INFORMATION



PCK52

* Commercial version in PDIL(P) or PLCC(S) package of PCK52 (Personal Computer Keyboard Controller containing Software PCS52).

PCS52 : (assembly source code only)

* Personal Computer Software dedicated for : IBM PC keyboard.

6

APPLICATION NOTES

7

MHS

APPLICATION NOTE

AN1022

DESIGN OF A PROCESSOR CARD USING THE 8051/31 AH OR THE 80C51/C31

NOTA: IN THE TEXT HEREAFTER, 8051 REFERS TO THE 8051AH (HMOS) OR 80C51 (CMOS).

I - 8051/31 OVERVIEW

The 8051 is a stand-alone high performance single-chip microcontroller. It provides hardware features, architectural enhancements and new instructions that make it a powerful and cost-effective controller for applications that require up to 64 K bytes of program memory and/or 64 K bytes of data memory.

The 8031 is a control-oriented CPU without on-chip program memory (ROM). It can address up to 64 K bytes of program memory in addition to 64 K bytes of external data memory.

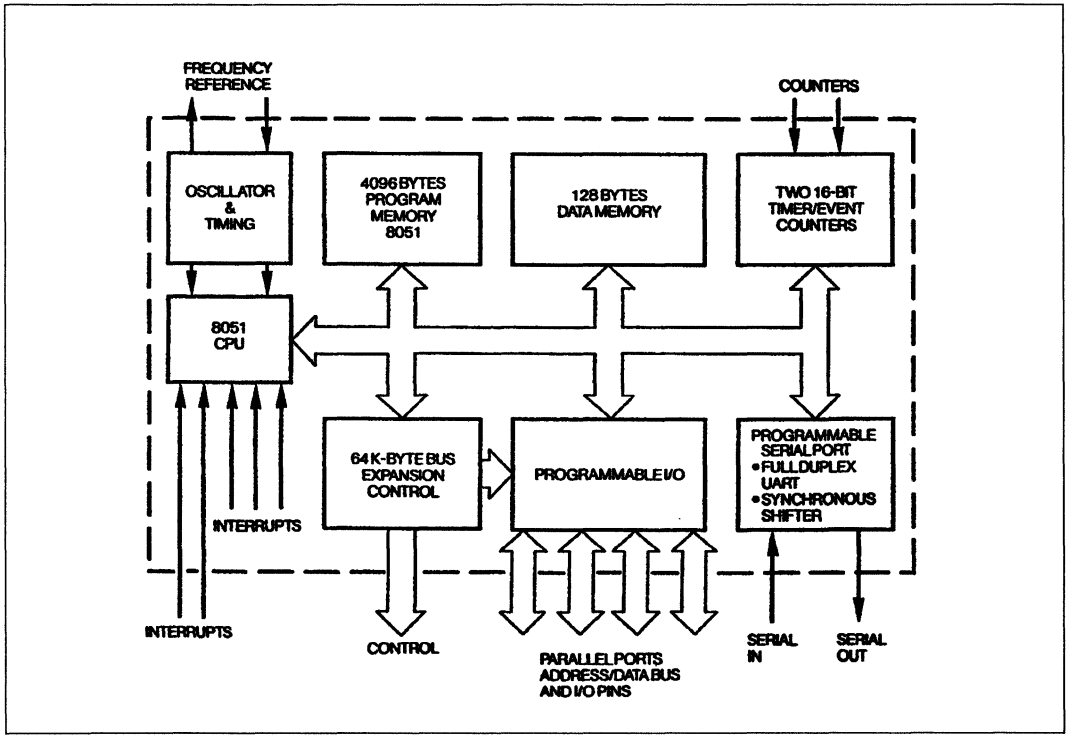
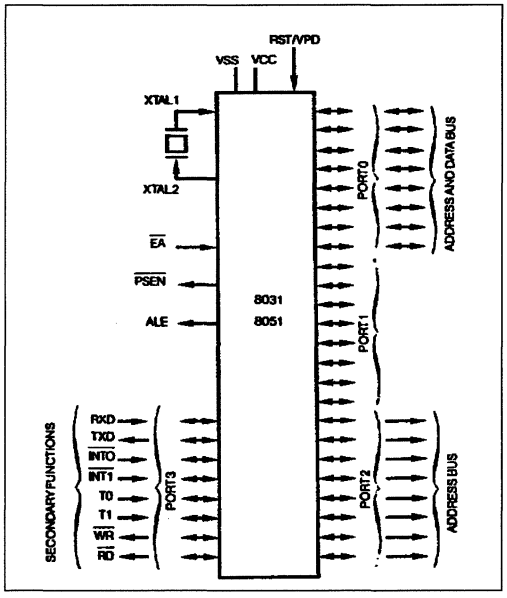
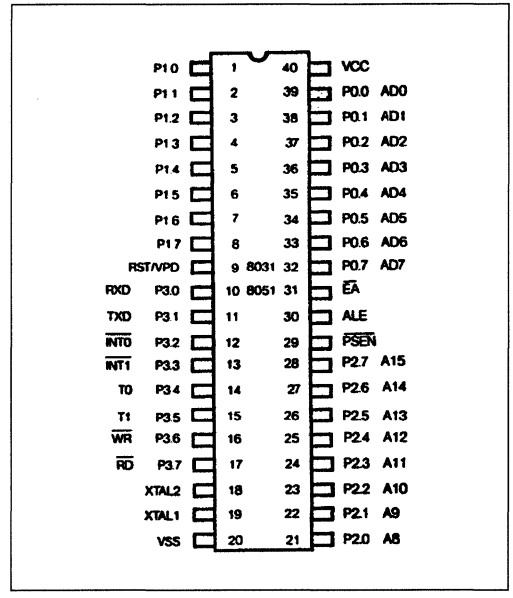


Figure 1 : Block Diagram.

7





Internally the 8051/31 comprises :

- 4 K bytes of program memory (8051 only)
- 128 bytes of data memory
- Four 8-bit ports giving up to 32 input/output lines
- Two 16-bit timers/event counters
- Full-duplex serial communications port
- An enhanced 8048 architecture
- Boolean processor within the CPU
- Externally an 8051/31 system may be expanded to comprise :
  - Up to 64 K bytes of program memory
  - Up to 64 K bytes of data memory
- Input/output expansion using memory-mapped peripherals

Most instructions are of one or two bytes and are executed in one or two cycles. With a 12 MHz crystal the cycle time is 1 µs. Only multiplication and division require 4 cycles to execute (4 µs at 12 MHz).

Because the architecture is based on the 8048 processors, programs written for the 8048 can be transferred to the 8051 with some modification. These progams will run at 2 1/2-10 times the speed of equivalent programs on the 8048, due to the 8051 processor's higher throughput.

7

8051 FAMILY PIN DESCRIPTION

VSS

Circuit ground potential.

VCC

+ 5 V power supply during operation and program verification.

PORT 0

Port 0 is an 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus when using external memory. It is used for data output during program verification. Port 0 can sink/source eight LS TTL loads.

PORT 1

Port 1 is an 8-bit quasi-bidirectional I/O port. It is used for the low-order address byte during program verification. Port 1 can sink/source three LS TTL loads.

PORT 2

Port 2 is an 8-bit quasi-bidirectional I/O port. It also emits the high-order address byte when accessing external memory. It is used for the high-order address and the control signals during program verification. Port 2 can sink/source three LS TTL loads.

PORT 3

Port 3 is an 8-bit quasi bidirectional I/O port. It also contains the interrupt, timer, serial port and RD and WR pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. Port 3 can sink/source three LS TTL loads. The secondary functions are assigned to the pins of Port 3, as follows :

- RXD/data (P3.0). Serial port's receiver data input

(asynchronous) or data input/output (synchronous).

- TXD/clock (P3.1). Serial port's transmitter data output (asynchronous) or clock output (synchronous).
- INT0 (P3.2). Interrupt 0 input or gate control input for counter 0.
- INT1 (P3.3). Interrupt 1 input or gate control input for counter 1.
- T0 (P3.4). Input to counter 0.
- T1 (P3.5). Input to counter 1.
- WR (P3.6). The write control signals latches the data byte from Port 0 into the External Data Memory.
- RD (P3.7). The read control signal enables External Data Memory to Port 0.

RST/VPD

A high level on this pin resets the 8051. A small internal pulldown resistor permits power-on reset using only a capacitor connected to VCC. If VPD is held within its spec while VCC drops below spec, VPD will provide standby power to the RAM. When VPD is low, the RAM's current is drawn from VCC.

ALE

Provides Address Latch Enable output used for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.

$\overline{PSEN}$

The Program Store Enable output is a control signal that enables the external Program Memory to the bus during external fetch operations. It is activated every six oscillator periods, except during external data memory accesses. Remains high during internal program execution.

$\overline{EA}$

When held at a TTL high level, the 8051 executes instructions from the internal ROM when the PC is less than 4096. When held at a TTL low level, the 8051 fetches all instructions from external Program Memory.

XTAL1

Input to the oscillator's high gain amplifier. Required when a crystal is used. Connect to VSS when external source is used on XTAL2.

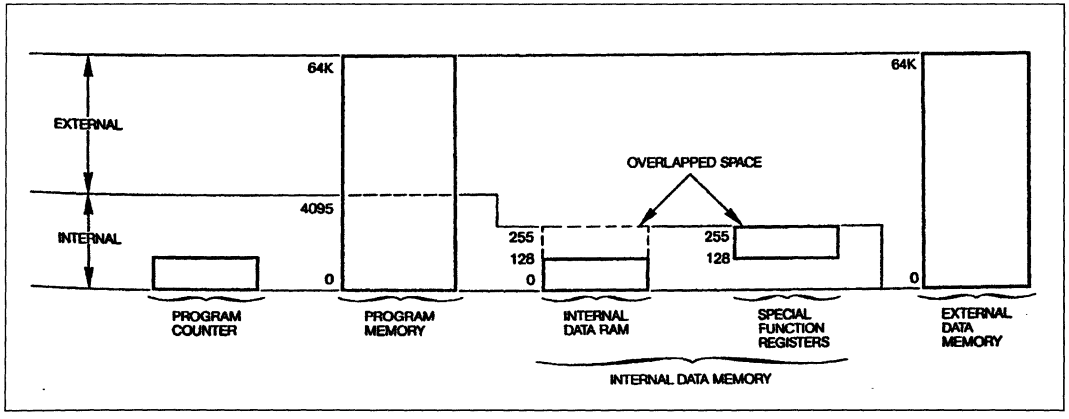
XTAL2

Output from the oscillator's amplifier. Input to the internal timing circuitry. A crystal or external source can be used.

1.1. 8051 CPU ARCHITECTURE

The CPU operates in four memory spaces. These are :
- 64 K byte program memory
- 64 K byte external data memory
- 384 byte internal data memory
- 16 bit program counter

The 384 byte internal data memory is divided into 256 bytes of RAM and 128 bytes for the special function registers (SFR). The top 128 bytes of RAM and the SER are overlapped. Of the 384 bytes theoretically available, only 128 bytes of RAM are provided together with 20 bytes in the SFR.
The SFR contains all the 8051 registers except the program counter, allowing operations to be carried out on all registers. Within the internal RAM and SFR there are also bit variables. The 16 bytes from 20H to 2FH contain 128 directly addressable bits. There are another 128 bits allocated to the SFR although not all of them are assigned. Contained within the 8051 CPU is a boolean processing unit.
This unit uses the carry as an accumulator in conjunction with bit instructions allowing fast and simple bit-manipulation without the need for masking.



a) RAM Bit Addresses.

RAM BYTE (MSB)								(LSB)
7FH								
≈								≈
2FH	7F	7E	7D	7C	7B	7A	79	78
2EH	77	76	75	74	73	72	71	70
2DH	6F	6E	6D	6C	6B	6A	69	68
2CH	67	66	65	64	63	62	61	60
2BH	5F	5E	5D	5C	5B	5A	59	58
2AH	57	56	55	54	53	52	51	50
29H	4F	4E	4D	4C	4B	4A	49	48
28H	47	46	45	44	43	42	41	40
27H	3F	3E	3D	3C	3B	3A	39	38
26H	37	36	35	34	33	32	31	30
25H	2F	2E	2D	2C	2B	2A	29	28
24H	27	26	25	24	23	22	21	20
23H	1F	1E	1D	1C	1B	1A	19	18
22H	17	16	15	14	13	12	11	10
21H	0F	0E	0D	0C	0B	0A	09	08
20H	07	06	05	04	03	02	01	00
1FH 18H	Bank 3							
17H 10H	Bank 2							
0FH 08H	Bank 1							
07H 00H	Bank 0							

b) Hardware Register Bit Addresses.

Direct Byte Address	Bit Addresses (MSB)							(LSB)	Hardware Register Symbol
0FFH									
0F0H	F7	F6	F5	F4	F3	F2	F1	F0	B
0E0H	E7	E6	E5	E4	E3	E2	E1	E0	ACC
0D0H	D7	D6	D5	D4	D3	D2	D1	D0	PSW
0B8H	-	-	-	BC	BB	BA	B9	B8	IP
0B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3
0A8H	AF	-	-	AC	AB	AA	A9	A8	IE
0A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2
98H	9F	9E	9D	9C	9B	9A	99	98	SCON
90H	97	96	95	94	93	92	91	90	P1
88H	8F	8E	8D	8C	8B	8A	89	88	TCON
80H	87	86	85	84	83	82	81	80	P0

As in the 8048 there are also eight general purpose registers R0-R7 asssigned to RAM addresses. In the 8051 there are four banks of eight registers that are available, the bank in use being selected by two bits in the processor status word (PSW). This is useful for task changing, such as interrupt processing. The stack has also changed from that of the 8048. There is an 8-bit stack pointer that directly addresses internal RAM allowing all of the internal memory to be used (up to 128 bytes).

In the 8051 the lower 4 K of the 64 K program memory address space is filled by internal ROM. By tying the EA pin high, the processor can be forced to fetch from the internal ROM for program memory addresses between 0 and 4 K. If the EA pin is tied low, then all program memory fetches are from external memory. The execution speed is the same regardless of whether fetches are from internal or external memory.

1.2 8051 INSTRUCTION SET

The following table summarises the 8051 instruction set.

ARITHMETIC OPERATIONS				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
ADD	A.Rn	Add register to Accumulator	1	1
ADD	A.direct	Add direct byte to Accumulator	2	1
ADD	A.@Ri	Add indirect RAM to Accumulator	1	1
ADD	A.#data	Add immediate data to Accumulator	2	1
ADDC	A.Rn	Add register to Accumulator with Carry	1	1
ADDC	A.direct	Add direct byte to A with Carry flag	2	1
ADDC	A.@Ri	Add indirect RAM to A with Carry flag	1	1
ADDC	A.#data	Add immediate data to A with Carry flag	2	1
SUBB	A.Rn	Subtract register from A with Borrow	1	1
SUBB	A.direct	Subtract direct byte from A with Borrow	2	1
SUBB	A.@Ri	Subtract indirect RAM from A with Borrow	1	1
SUBB	A.#data	Subtract immed. data from A with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
MUI	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1
LOGICAL OPERATIONS				
MNEMONIC		**DESTINATION**	**BYTE**	**CYC**
ANL	A.Rn	AND register to Accumulator	1	1
ANL	A.direct	AND direct byte to Accumulator	2	1
ANL	A.@Ri	AND indirect RAM to Accumulator	1	1
ANL	A.#data	AND immediate data to Accumulator	2	1
ANL	direct.A	AND Accumulator to direct byte	2	1
ANL	direct.#data	AND immediate data to direct byte	3	2
ORL	A.Rn	OR register to Accumulator	1	1
ORL	A.direct	OR direct byte to Accumulator	2	1
ORL	A.@Ri	OR indirect RAM to Accumulator	1	1
ORL	A.#data	OR immediate data to Accumulator	2	1
ORL	direct.A	OR Accumulator to direct byte	2	1
ORL	direct.#data	OR immediate data to direct byte	3	2
XRL	A.Rn	Exclusive-OR register to Accumulator	1	1
XRL	A.direct	Exclusive-OR direct byte to Accumulator	2	1
XRL	A.@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A.#data	Exclusive-OR immediate data to A	2	1
XRL	direct.A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct.#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator left	1	1
RLC	A	Rotate A left through the Carry flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate A Right through Carry flag	1	1
SWAP	A	Swap nibbles within the Accumulator	1	1
DATA TRANSFER				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
MOV	A.Rn	Move register to Accumulator	1	1
MOV	A.direct	Move direct byte to Accumulator	2	1

MOV	A.@Ri	Move indirect RAM to Accumulator	1	1
MOV	A.#data	Move immediate data to Accumulator	2	1
MOV	Rn.A	Move Accumulator to register	1	1
MOV	Rn.direct	Move direct byte to register	2	2
MOV	Rn.#data	Move immediate data to register	2	1
MOV	direct.A	Move Accumulator to direct byte	2	1
MOV	direct.Rn	Move register to direct byte	2	2
MOV	direct.direct	Move direct byte to direct	3	2
MOV	direct.@Ri	Move indirect RAM to direct byte	2	2
MOV	direct.#data	Move immediate data to direct byte	3	2
MOV	@Ri.A	Move Accumulator to indirect RAM	1	1
MOV	@Ri.direct	Move direct byte to indirect RAM	2	2
MOV	@Ri.#data	Move immediate data to indirect RAM	2	1
MOV	DPTR.#data16	Load Data Pointer with a 16-bit constant	3	2
MOVC	A.@A + DPTR	Move Code byte relative to DPTR to A	1	2
MOVC	A.@A + PC	Move Code byte relative to PC to A	1	2
MOVX	A.@Ri	Move External RAM (8-bit addr) to A	1	2
MOVX	A.@DPTR	Move External RAM (16-bit addr) to A	1	2
MOVX	@Ri.A	Move A to External RAM (8-bit addr)	1	2
MOVX	@DPTR.A	Move A to External RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A.Rn	Exchange register with Accumulator	1	1
XCH	A.direct	Exchange direct byte with Accumulator	2	1
XCH	A.@Ri	Exchange indirect RAM with A	1	1
XCHD	A.@Ri	Exchange low-order Digit ind. RAM w/A	1	1
BOOLEAN VARIABLE MANIPULATION				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
CLR	C	Clear Carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry flag	1	1
SETB	bit	Set direct Bit	2	1
CPL	C	Complement Carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C.bit	AND direct bit to Carry flag	2	2
ANL	C.bit	AND complement of direct bit to Carry	2	2
ORL	C.bit	OR direct bit to Carry flag	2	2
ORL	C.bit	OR complement of direct bit to Carry	2	2
MOV	C.bit	Move direct bit to Carry flag	2	1
MOV	bit.C	Move Carry flag to direct bit	2	2
PROGRAM AND MACHINE CONTROL				
MNEMONIC		**DESCRIPTION**	**BYTE**	**CYC**
ACALL	addr11	Absolute Subroutine Call	2	2
LCALL	addr16	Long Subroutine Call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute Jump	2	2
LJMP	addr16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A + DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is Zero	2	2
JNZ	rel	Jump if Accumulator is Not Zero	2	2
JC	rel	Jump if Carry flag is set	2	2
JNC	rel	Jump if No Carry flag	2	2
JB	bit.rel	Jump if direct Bit set	3	2
JNB	bit.rel	Jump if direct Bit Not set	3	2
JBC	bit.rel	Jump if direct Bit is set & Clear bit	3	2
CJNE	A.direct.rel	Compare direct to A & Jump if Not Equal	3	2
CJNE	A.#data.rel	Comp. immed. to A & Jump if Not Equal	3	2

CJNE	Rn.#data.rel	Comp. immed. to reg & Jump if Not Equal	3	2
CJNE	@Ri.#data.rel	Comp. immed. to ind. & Jump if Not Equal	3	2
DJNZ	Rn.rel	Decrement register & Jump if Not Zero	2	2
DJNZ	direct.rel	Decrement direct & Jump if Not Zero	3	2
NOP		No operation	1	1

NOTES ON DATA ADDRESSING MODES :

Rn	Working register R0-R7
direct	128 internal RAM locations, any I O port, control or status register
@Ri	Indirect internal RAM location addressed by register R0 or R1
#data	8-bit constant included in instruction
#data16	16-bit constant included as bytes 2 & 3 of instruction
bit	128 software flags, any I O pin, control or status bit

NOTES ON PROGRAM ADDRESSING MODES :

addr16	Destination address for LCALL & LJMP may be anywhere within the 64-Kilobyte program memory address space.
addr11	Destination address for ACALL & AJMP will be within the same 2-Kilobyte page of program memory as the first byte of the following instruction
rel	SJMP and all conditional jumps include an 8-bit offset byte. Range is + 127 – 128 bytes relative to first byte of the following instruction.

1.3. INPUT/OUTPUT PORTS

The 8051 contains four 8-bit ports known as P0-P3. Any line on any port may be individually configured as an input or an output. Ports 0, 2 and 3 can also carry out other functions, depending upon how the 8051 has been programmed.

Port 0 :

Bidirectional port with open-drain pins capable of driving 2 TTL loads. Multiplexed low-order address and data bus for 8051 expansion.

Port 1 :

Quasi-bidirectional port capable of driving a single TTL load.

Port 2 :

Quasi-bidirectional port capable of driving a single TTL load. High-order address bus for 8051 expansion.

Port 3 :

Quasi-bidirectional port capable of driving a single TTL load. Contains interrupt and timer inputs, serial input/output and external memory control signals.

The differences between the types of port are at the chip level. At functional and programming levels there are no differences (except for Port 0 drive capability). After power-up all ports are configured as inputs, they may be reconfigured by writing a zero (0) to the pin.

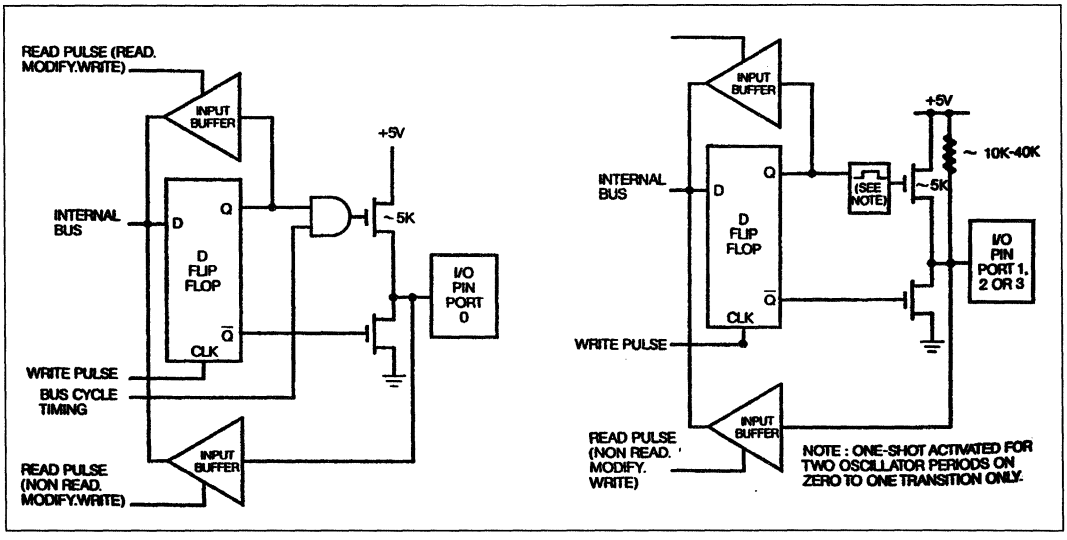


Figure 7 : Port Pin Structure.

When writing a new value to port output pins, a one (1) must be written to all those pins that are to remain as inputs. All port pins may also be addressed individually by bit instructions. The secondary functions of Port 3 are generated if the pin is configured as an input.

In the 8051 instruction set there are three classes of instructions.

Read : Read current value from source
Write : Write value to destination
Read-modify-write : Read from source, perform operation, write to destination.

Each class of instruction operates differently on the I/O parts.

The flip-flop contains the value that was written to the port. If it was one (1) the port pulls up the output line. This gives an output of one (1) for an output or allows reading of the port pin for an input. Read-modify-write instructions use the value in the flip-flop as the source. This means that pins configured as inputs will not be changed (unless done so by the instructions) when the write back is performed. Consequently, read-modify-write instructions are used to configure ports. After power-on all bits are inputs (at the one (1) level). Performing an AND of one (1) with bits to remain as inputs and zero (0) with bits to become outputs will change this configuration.

Example : Configure bits P1.0-P1.2 as outputs leaving P1.3-P1.7 as inputs.

```
ANL P1, # 11111000B
```

If later on, we wish P1.1 to be an input and P1.7 to be an output, then the following is performed.

```
ORL P1, # 00000010B ; set P1.1 to an input
ANL P1, # 01111111B ; set P1.7 to an output
```

Reading and writing of ports is accomplished by "MOVE" instructions.

```
MOV A, P1 ; read port 1
MOV P0, # 0FFH ; write to port 0
```

1.4. TIMER/COUNTERS

The 8051 contains two 16-bit progammable timers/event counters. They can be used to measure time intervals, external pulse widths or generate periodic interrupt request. Each timer may be configured into one of four modes. Modes 0-2 are identical for each timer, only Mode 3 is different.

A-Mode 0 :

An 8-bit counter/timer with a 5-bit prescaler. Reading the high order half (TH0 or TH1) accesses bits 12-5 of the counter. The lower half (TL0 or TL1) contains the prescaler in bits 4-0. Bits 5-7 are not used and should be set to zero.

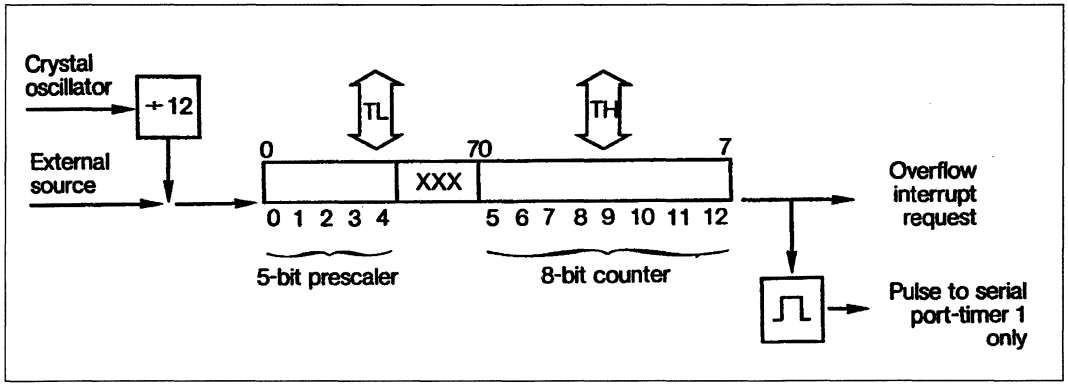


Figure 8 : Timer Mode 0.

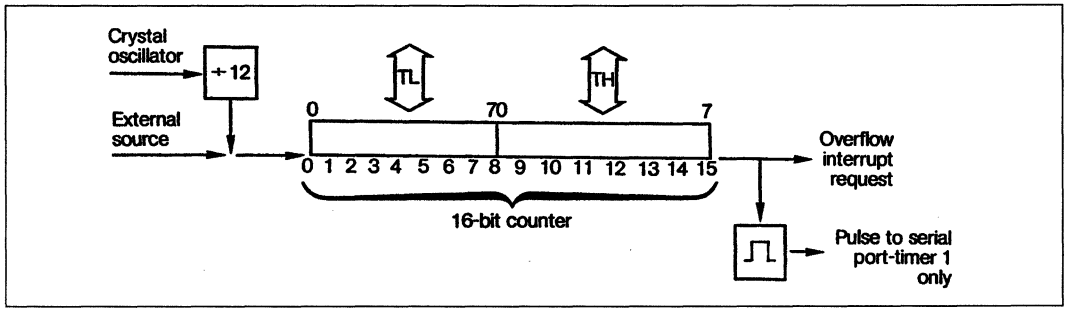


Figure 9 : Timer Mode 1.

C - Mode 2 :

An 8-bit auto-reload timer. The lower half is incremented until it overflows. The auto-reload value in the top half is then loaded into the lower half and counting resumes.

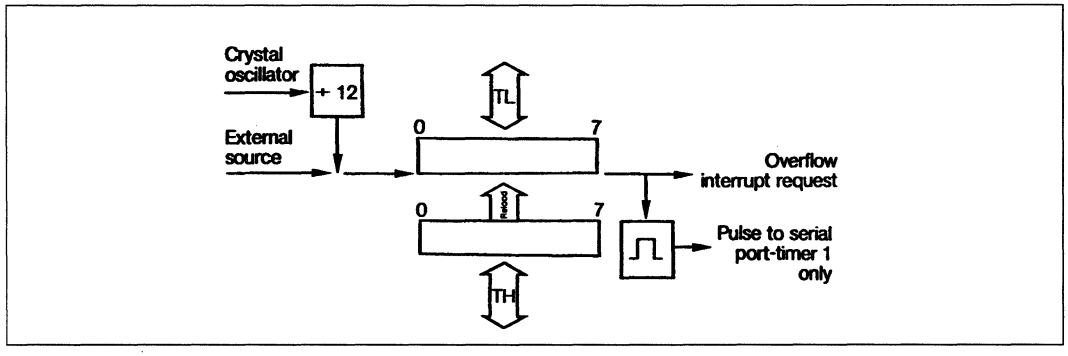


Figure 10 : Timer Mode 2.

D-Mode 3 :

Timer 0 is configured as two independent 8-bit timers. The upper half, TH0, is an 8-bit timer using the clock as a source. It is controlled by the bits of Timer 1. The 'Run' and 'Overflow' bits fo Timer 1 are, therefore, not available to Timer 1. The lower half of Timer 0 , TL0, is configured as an 8-bit counter/timer in the usual manner.

Timer 1 has no overflow or run bits when Timer 0 is in mode 3. To overcome this, mode 3 of Timer 1 is used to halt counting, changing to another mode will start it again. Although Timer 1 can be used in Modes 0-2, note that it cannot generate an interrupt.

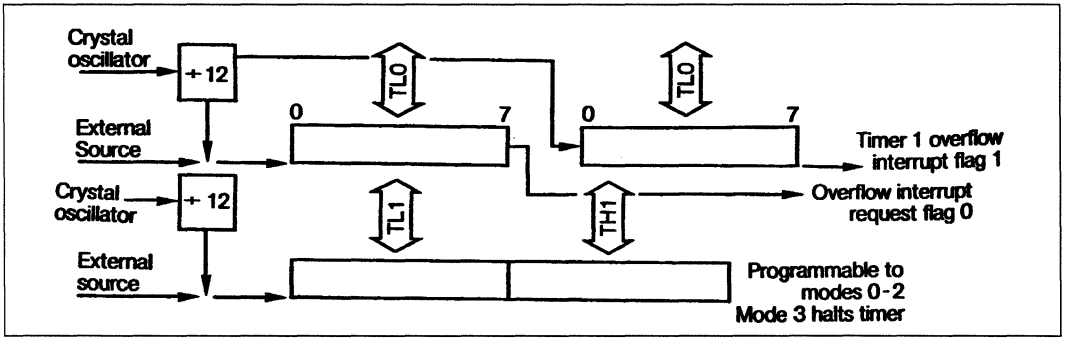


Figure 11 : Timer Mode 3.

E-Timer control

Associated with the timers are two control registers called TMOD and TCON-timer mode and timer control. The bit explanations are given in the tables.

(MSB)							(LSB)
GATE	C/T	M1	M0	GATE	C/T	M1	M0
TIMER 1				TIMER 0			

GATE

Gating control. When set, Timer/counter "x" is enabled only while "INTx" pin is high and "TRx" control bit is set. When cleared, timer/counter is enabled whenever "TRx" control bit is set.

C/T

Timer or Counter Selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from "Tx" input pin).

M1	M0	Operating Mode
0	0	8048 Timer. "TLx" serves as five-bit prescaler.
0	1	16-bit timer/counter. "THx" and "TLx" are cascaded ; there is no prescaler.
1	0	8-bit auto-reload timer/counter. "THx" holds a value which is to be reloaded into "TLx" each time it overflows.
1	1	(Timer 0) TL0 is an eight-bit timer/counter controlled by the standard Timer 0 control bits. TH0 is an eight-bit timer only controlled by Timer 1 control bits.
1	1	(Timer 1) Timer/counter 1 stopped.

(MSB)							(LSB)
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Symbol	Position	Name and Significance
TF1	TCON.7	Timer 1 overflow Flag. Set by hardware on timer counter overflow. Cleared when interrupt processed.
TR1	TCON.6	Timer 1 Run control bit. Set/ cleared by software to turn timer/ counter on/off.
TF0	TCON.5	Timer 0 overflow Flag. Set by hardware on timer/counter overflow. Cleared when interrupt processed.
TR0	TCON.4	Timer 0 Run control bit. Set/ cleared by software to turn timer/ counter on/off.
IE1	TCON.3	Interrupt 1 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT1	TCON.2	Interrupt 1 Type control bit. Set/ cleared by software to specify falling edge/low level triggered external interrupts.
IE0	TCON.1	Interrupt 0 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT0	TCON.0	Interrupt 0 Type control bit. Set/ cleared by software to specify falling edge/low level triggered external interrupts.

1.5. SERIAL PORT

The 8051 contains a serial port that can link with UART devices or expand input/output. The serial port is programmable into one of four modes.

Mode 0 : Synchronous input/output using TTL or CMOS shiff registers.

Mode 1 : UART interface with a 10-bit frame and variable transmission rate.

Mode 2 : UART interface with an 11-bit frame and fixed transmission rate.

Mode 3 : UART interface with an 11-bit frame and variable transmission rate.

The port is full duplex, meaning that it can transmit and receive at the same time. The serial port buffer register (SBUF) holds the received data and the data to be transmitted, a write loading the transmit register and a read reading from the receive register.

The control register, SCON, contains the information to configure the port.

(MSB)							(LSB)
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Symbol	Position	Name and Significance
SM0	SCON.7	Serial port Mode control bit 0. Set/cleared by software (see note).
SM1	SCON.6	Serial port Mode control bit 1. Set/cleared by software (see note).
SM2	SCON.5	Serial port Mode control bit 2. Set by software to disable reception of frames for which bit 8 is zero.
REN	SCON.4	Receiver Enable control bit. Set/ cleared by software to enable/ disable serial data reception.
TB8	SCON.3	Transmit Bit 8. Set/cleared by hardware to determine state of ninth data bit transmitted in 9-bit UART mode.
RB8	SCON.2	Receive Bit 8. Set/cleared by hardware to indicate state of ninth data bit received.
TI	SCON.1	Transmit interrupt flag. Set by hardware when byte transmitted. Cleared by software after servicing.
RI	SCON.0	Received Interrupt flag. Set by hardware when byte received. Cleared by software after servicing.

Note - the state of (SM0.SM1) selects :
(0, 0) - Shift register I/O expansion.
(0, 1) - 8 bit UART, variable data rate.
(1, 0) - 9 bit UART, fixed data rate.
(1, 1) - 9 bit UART, variable data rate.

A-Mode 0 : synchronous input/output

Two lines are used, P3.0 and P3.1. The first is the serial data, either input or output, and the second is a bit clock. Every time a byte is written to SBUF it is shifted out on P3.0. Every time the receive flag (R1) is cleared, 8-bits are read in from P3.0 to SBUF. Once completed, R1 is set again. Data can be clocked into a shift register on the rising edge of the clock (P3.1).
With the addition of more hardware and port lines a fully interlocked full or half-duplex system can be built. For parallel intercommunications it is probably more expedient to use one of the ports. The advantage of this serial system is that it is faster than a UART, transfer is at 1M bit/second with a 12 MHz clock, and may be cheaper than a parallel system since only two lines are needed (P3.0 and P3.1).

B-Modes 1-3

These are the UART modes of the Serial Port. Each bit is divided into 16 "ticks", the bit being sampled during the seventh, eighth and ninth ticks. Format is the usual UART/RS 232C format of one start bit, eight or nine data bits and one or two stops bits. Typical frame formats are shown below.

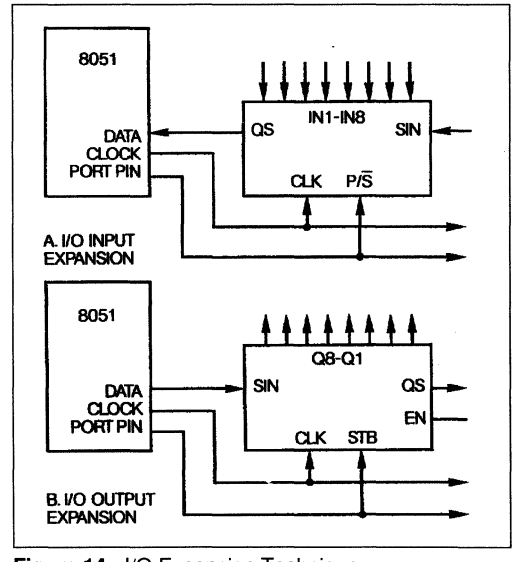


Figure 14 : I/O Expansion Technique.

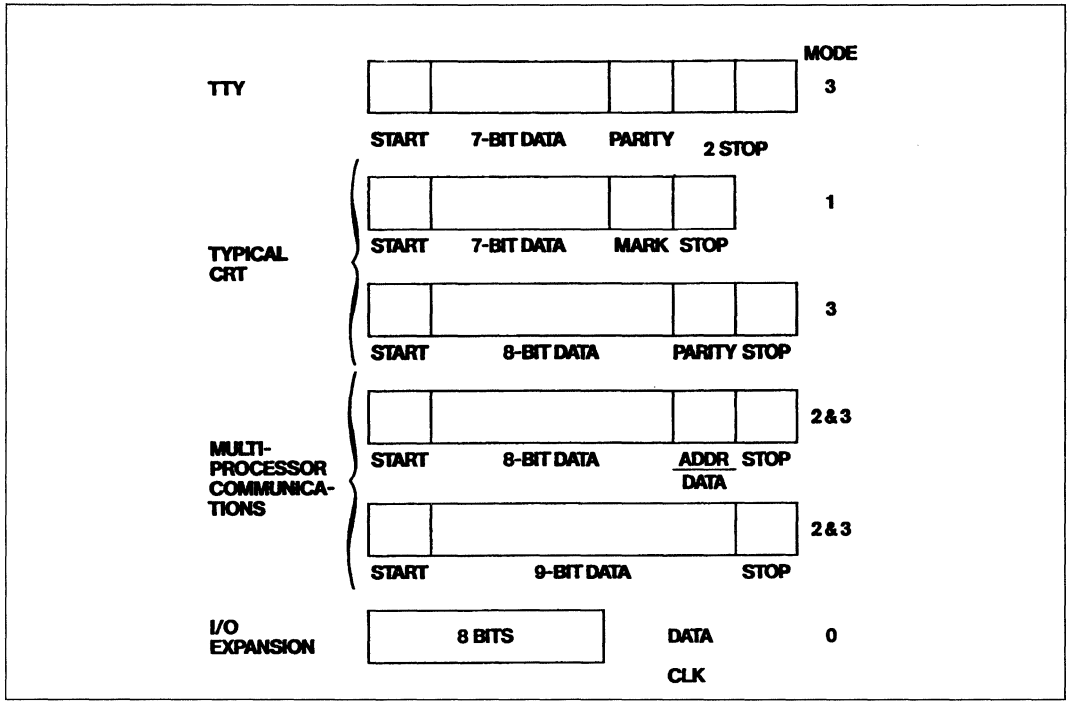


Figure 15 : Typical Frame Formats.

Mode 1 consists of start and stop bits around 8 data bits. This is a format used by many VDUs. The code is 7-bit ASCII and parity, or is an 8-bit code (normally ASCII with bit 7 at zero plus 128 special codes). For teletypes two stop bits are usually required to reset the mechanical apparatus of the next character. Mode 3 can used with the eighth bit set to a one (1) so that two stop bits are always generated. The same mode can be used of a CRT device having special codes as well as character codes plus parity.

1. Slaves - Configure serial port to interrupt CPU if the received ninth data bit is a one (1).

2. Master - Transmit frame containing address in first 8 data bits and set ninth data bit (i.e. ninth data bit designates address frame).

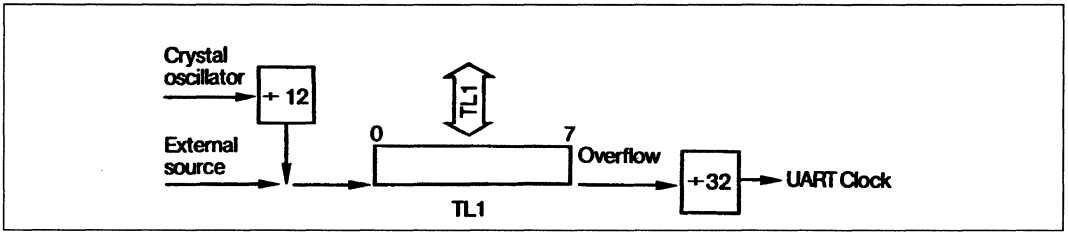
3. Slaves - Serial port interrupts CPU when address frame is received. Interrupt service program compares received address to its address. The slave which has been addressed reconfigures its serial port to interrupt the CPU on all subsequent transmissions.

4. Master - Transmit control frames and data frames (these will be accepted only by the previously addressed slave).

For multiprocessor communication between 8051 systems modes 2 and 3 are used. There are special features of the 8051 which make it advantageous to use these modes. Mode 2 is exactly the same as mode 3 except that its transmission rate is fixed.
As in mode 0 the transmit interrut (TI) bits is set when the byte has been transmitted. This indicates that the buffer is free for another character. On reception of a byte the receive interrupt (RI) bit is set. If the interrupt enable bits are set then an interrupt will occur whenever another character can be transmitted or when a character has been received.

C-Transmission rate

The serial port is clocked by the overflow of timer 1. The input to timer 1 is either the crystal frequency divided by 12, or an external clock via the T1 input of Port 3. In modes 0 and 2 the input frequency is feied at the oscillator frequency divided by 12 or 64 respectively. For mode 0 this gives 1 M bit per second with a 12 MHz crystal, and 187,500 bits per second for mode 2 (again with a 12 MHz crystal). Modes 1 and 3 have the transmission rate determined by timer 1. The timer is configured in autoreload to generate a fixed frequency. Either one-twelfth of the oscillator frequency or the T1 input is divided by "256-Minus-The-Value-in-TH1" (auto-reload value) which is then divided by 32 and used to clock the UART shift register. At 12 MHz this allows transmission rates from 122 to 31,250 bits per second.



For a UART operating at 300 bps using an 8051 clock at 12 MHz, the auto reload value is given by :

$$\text{Auto-reload} = \frac{\text{Clock Rate}}{12 \times 32 \times \text{Bit Rate}}$$

If an external clock is used, via T1

$$\text{Auto-reload} = \frac{\text{Clock Rate}}{32 \times \text{Bit Rate}}$$

So with an 8051 clock of 12 MHz and a transmission rate of 300 bps.

$$\text{Auto-reload} = \frac{12 \times 10^6}{12 \times 32 \times 300}$$

Auto-reload value = 104.16

The nearest integer is 104, so the actual value for the timer is 256 - 104 = 152 (98H in hexadecimal). Loading TH1 with a value of 98H gives the correct speed.

For a simple polling operation of the UART (no interrupts) the following routines can be used.

```
INIT  : MOV TMOD,# 0010XXXXB ; timer 1 = mode 2
        MOV TH1,# 98H        ; auto-reload
                               value
        MOV SCON,# 01010000B ; set up UART
        SETB TR1             ; start timer
```

Bit T1 is set in SCON to enable the routine to load the first character. Afterwards it waits until T1 is set before sending any new characters.

```
READ  : JNB RI, READ         ; wait for
                               character
        MOV A, SBUF          ; load character
        CLR RI               ; reset flag
        CLR ACC7             ; clear parity
        RET                  ; end routine
WRITE: MOV A, CHAR           ; load character
        MOV C, P             ; load parity into
                               carry
        MOV ACC7 C           ; transfer to
                               accumulator
WAIT  : JNB TI, WAIT         ; wait for buffer
                               to be free
        MOV SBUF, A          ; load character
        RETI                 ; end routine
```

II - EXTERNAL MEMORY

2.1. Accessing external memory

All memory, either program or data, is addressed using 16 address lines. Normally the first 4 K of program memory is contained in the CPU. The rest is accessed by using the secondary functions of the I/O ports. Port 2 emits the upper address byte (A8-A15) and Port 0 emits the lower address byte and data byte (AD0-AD7). The two signals ALE and PSEN are used together with two pins from Port 3 (RD and WR). The PSEN line selects external Program Memory, RD selects external Data Memory and WR latches data into external Data Memory. If PSEN and RD are OR-ed together, then the processor has a continuous 64 K program and data space, as found on many computers (cf. 8085). The Address Latch Enable (ALE) is used to latch the address into external latches or synchronous memories.
The 8051 can address 64 K bytes of external program memory (as the 8031 does) when the EA pin is tied to a low level. This disables the internal 4 K of program memory Ports 0 and 2 are automatically configured as outputs. Data access is via MOVX instructions using the DPTR or R0 or R1.

2.2 Timing diagrams

The basic cycle of the 8051 is six oscillator periods. When accessing Program Memory the instruction fetches overlap with the start of the next cycle. This means that the 8051 always fetches an even number of bytes, even if the second is ignored. External Data Memory accesses last for two cycles allowing slower peripherals or data memory than program memory. The address is valid on the falling edge of ALE. This is used to latch the address into the memory or for demultiplexing circuitry. For a read, of either data or program memory, the PSEN or RD signal enables the bus drivers. Data or an instruction is clocked into the processor on the rising edge of PSEN or RD. The WR signal indicates that the data on the bus is valid. During a write cycle, it can also be used to enable external buffers or latches in the system.

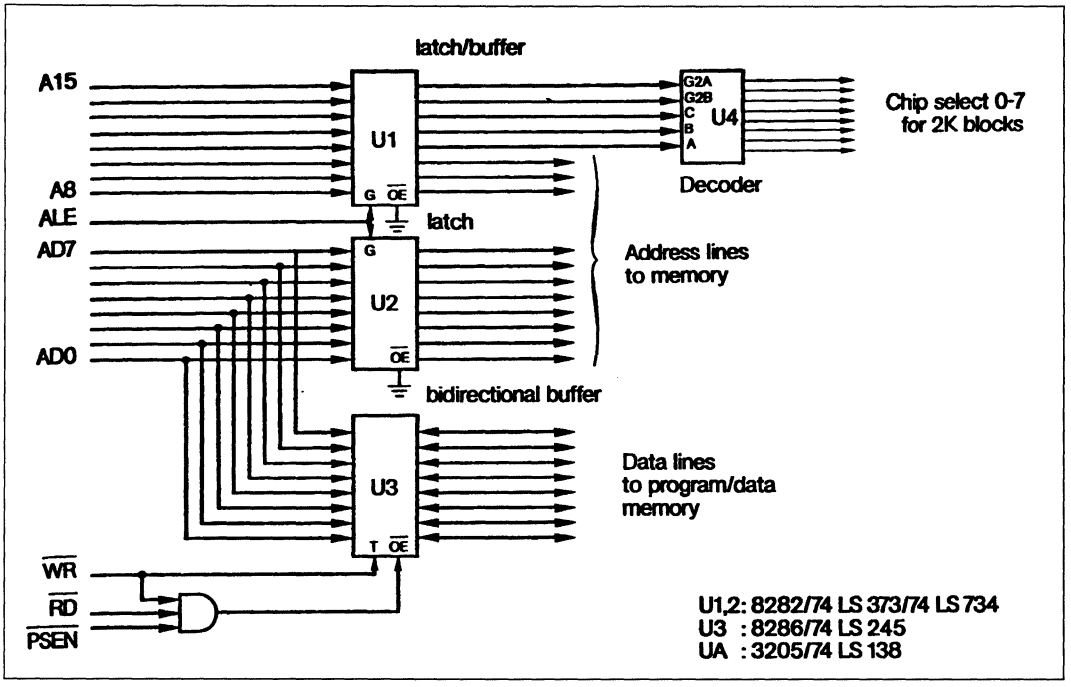


Figure 18 : Address/Data Latching.

2.3. External program memory

To add external memory we must examine the timing diagrams and choose a component with a suitable access time. The access time is the time from the validation of the chip select (CS) signal to the latching of the data into the processor, less and delays and set-up times. Most memory components require a latched demultiplexed bus with some decoding. The diagram below divides the memory space between 0 and 16 K into 2 K blocks.

For the 8051, addresses are stable by the falling edge of ALE. This is used to clock the address into the buffers. The data is clocked into the processor on the rising edge of PSEN. The time between the two is the maximum time available to access program memory.

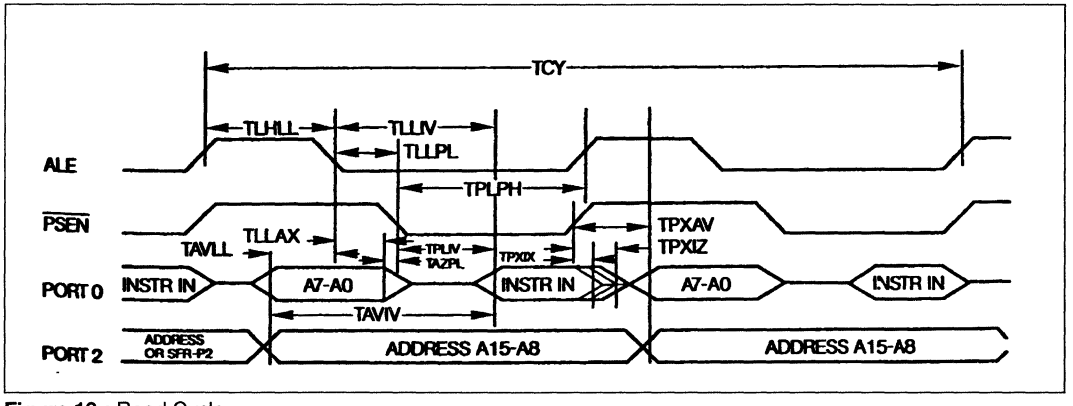


Figure 19 : Read Cycle.

Time to chip select being valid after ALE = delay through buffer and decoder. Time for data to be valid at processor = access time and delay through buffer.

We have to calculate the time allowed by the processor for a memory access, including any set-up times, according to the waveforms above. This is as follows :

Address hold time after ALE (TLLAX)
+ Address float to PSEN (TAZPL)
+ PSEN pulse width (TPLPH)
- Set-up time for 8051

This gives up :

48 + 0 + 215 + 0 = 263 nS (minimum) at 12 MHz

We now have to calculate the worst case delays across all the system components (buffers, decoders, etc.). For a typical system using all MOS components and a system with all TTL LS components, we have :

Buffer delay (max) from ALEMOS : 45 nS LSTTL : 30 nS from ALE
Decoder delay (max) MOS : 40 nS LS TTL : 39 nS
Data buffer delay (max) MOS : 30 nS LS TTL : 12 nS
Total MOS : 115 nS LS TTL : 81 nS

Times can be reduced by using inverting buffers and/or Schottky TTL logic. These times are subtracted from the time calculated above to give us the maximum access time on the memory.

MOS components : 263 - 115 = 148 nS
TTL components : 263 - 81 = 182 nS

It is possible to lengthen the access times if the address latches are replaced by transparent latches, i.e., the outputs follow the inputs until latched. The address from the 8051 is valid 53 nS before the falling edge of ALE so it is possible to use this time to perform selection, effectively gaining all of the 53 nS. This technique does not work for flip-flop latches (i8282, 74 LS 374).

This effectively increases the access time to :

MOS components : 148 + 53 = 201 nS
TTL components : 182 + 53 = 235 nS

If a demultiplexed system is not required and large loads are not being driven, then it may be possible to do without latching and buffers altogether. There are certain memories that incorporate on chip address latches and are, therefore, ideally suited for multiplexed buses. These are known as synchronous memories, while those without internal latches are synchronous memories.

Also associated with memories is the Output Enable (OE). This signal enables the external drivers to pass information onto the bus. For demultiplexed signals the PSEN line can be used directly for program memory output enable.

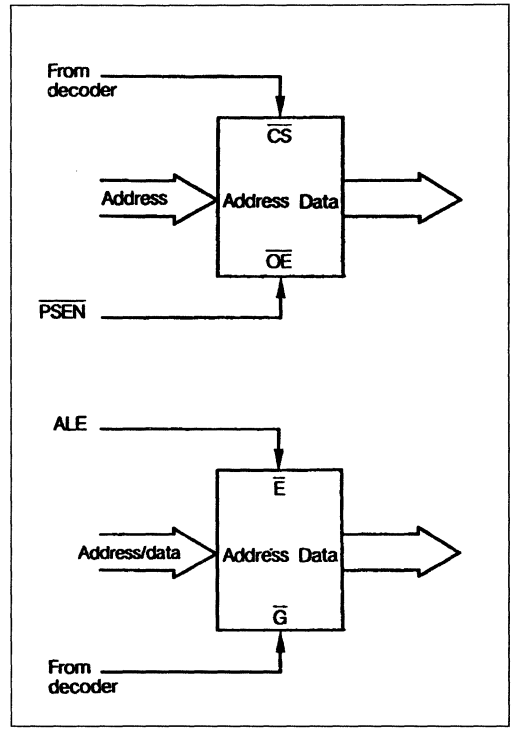


Figure 20 : ROM Selection.

In synchronous systems ALE is used to load the address register and OE (or G in this case) is selected by the decoder. This gains a little extra time as address decoding can start immediately after ALE, with all memories reading a particular byte. The decoder then selects the correct memory and enables its output drivers. Power consumption doesn't increase too much, as it is the output drivers that consume the most current. Note also, that a latched decoder (or equivalent system) mut also be used if A0-A7 are used for address decoding (A8-A15 are available throughout the cycle).

2.4. External data memory

External data memory will usually be RAM since the ability to read and write data will be required. Generally in a system of this sort most accesses will be to the program memory, interspersed with RAM accesses. Because RAM accesses are not as common as ROM accesses, the time allowed for them can be longer without much system degradation. In the 8051 RAM accesses are twice as long as ROM accesses allowing the use of slower and cheaper RAMs. As the timing diagrams show, the read (RD) and write (WR) signals are active for much longer, in fact a minimum of 400 nS.

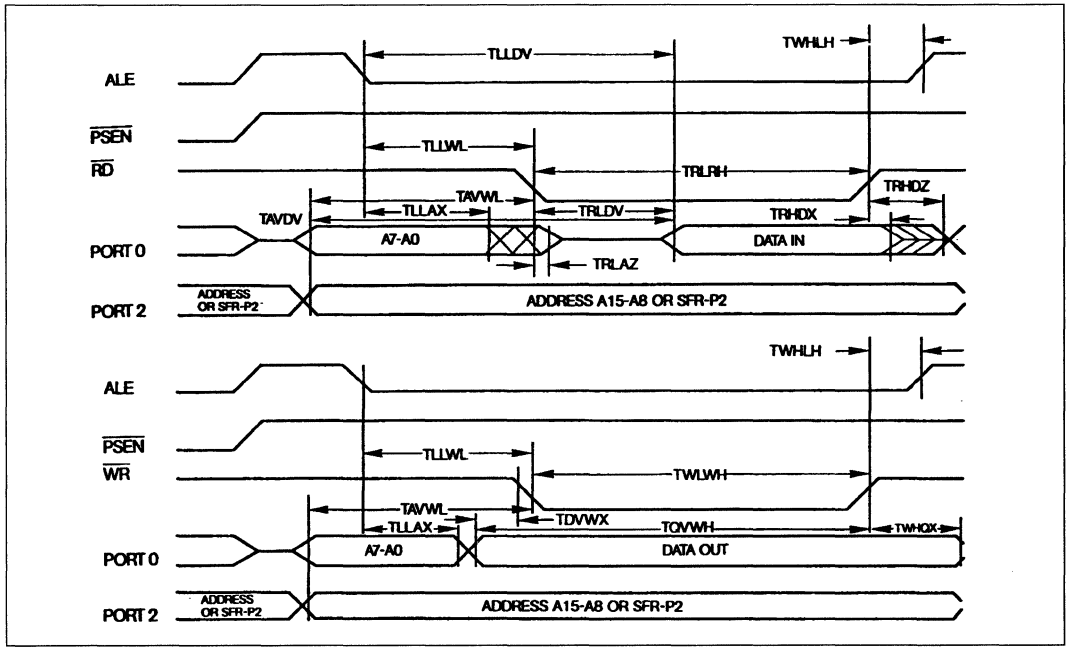


Figure 21 : RAM Read and Write Timing.

If using an asynchronous system with latches this allows an access time of at least 485 nS (for MOS components) which is ample for most memories. The circuit configurations are very similar. The RAM and ROM can be placed in parallel and the output enable lines (OE) selected by PSEN or RD. Synchronous RAMs are more difficult to deal with. We cannot use the same techniques as for PROMs. Loading the address into all memories and selecting a particular output with the RD signal and decoding logic is fine, but we will also have to do this with the WR signal whenever we update data. It becomes apparent that it is much easier to decode the address first and use the WR and RD signals for writing and output enable.

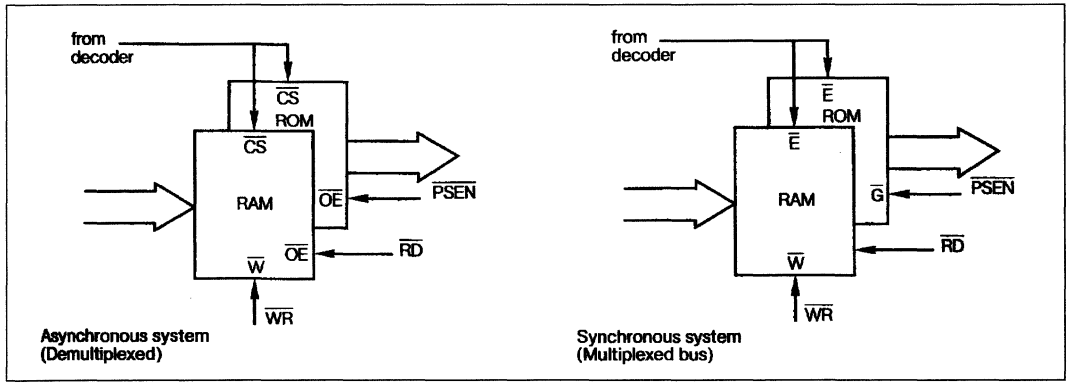
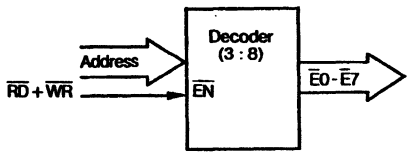


Figure 22 : RAM/ROM Selection.

To generate the control signals for the synchronous scheme, the following circuit can be used.



This scheme is possible because the addresses are stable before RD + WR and remain valid throughout the cycle (A8 – A15). It may be possible to use ALE in place of RD + WR, but this depends upon the memory characteristics.

Figure 23 : Generation of Selection Signals.

2.5. Single memory space configuration

Often it is useful to be able to treat program information as data. This is especially true of development systems where you wish to enter instructions as data, and then execute them. To accomplish this on the 8051 it is necessary to combine the program and data memory spaces. Both 64 K byte areas can be mapped to a single memory space that is selected by RD, WR or PSEN. The only modification necessary is to "OR" the PSEN and RD signals to form a new read signal for memory. Note that the RAM access time must be the same as that for the ROM - in the region of 200 nS.

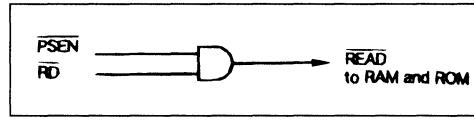


Figure 24 : Generation of Read.

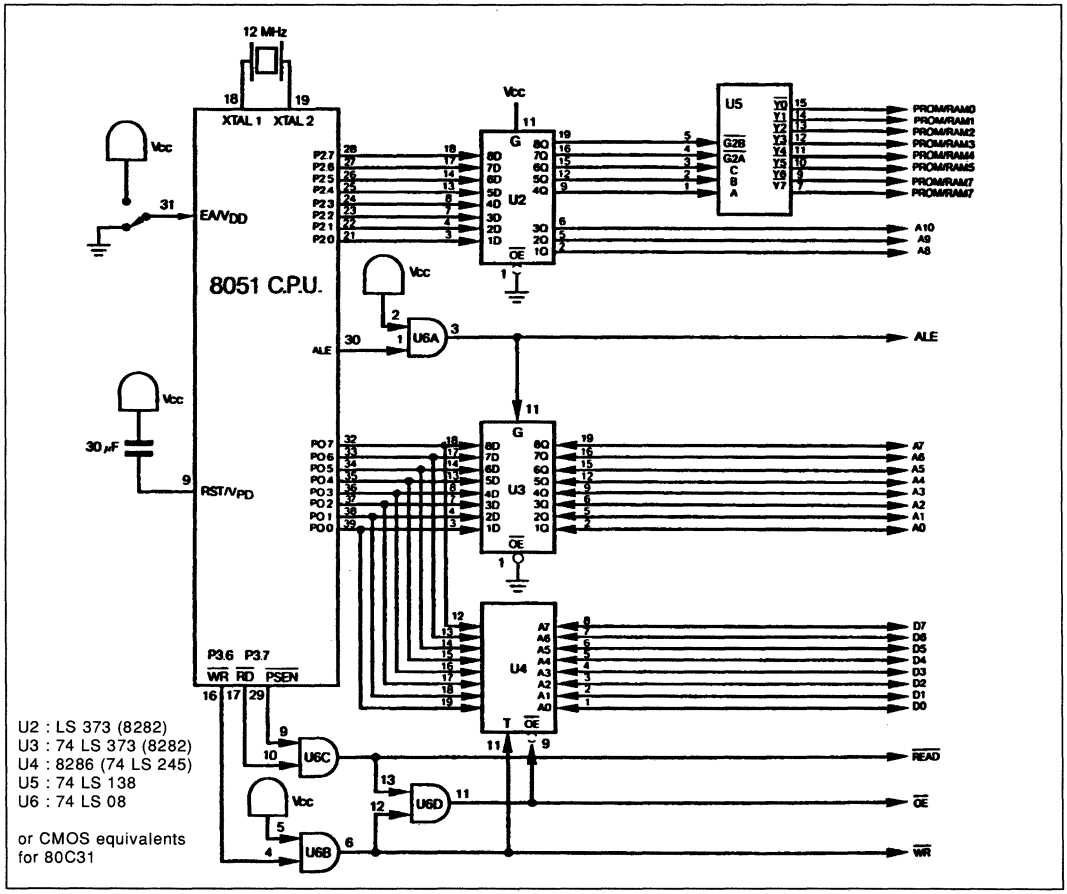


Figure 26 : Example 8051 System.

A circuit diagram of a CPU card using this system is given below. All signals are buffered to provide drive to external systems via a connector as well as all the memory components. Only the first 16 K from 0 - 16 K has been decoded into 2 K blocks that are selectable between ROM and RAM (apart from the first block). This is accomplished by a switch on the write ($\overline{WR}$) line. The ROM and RAM chips used are the HM 6616 and HM 6116 which are directly pin-compatible. Pin 21 is either the program or write ($\overline{WR}$) line.

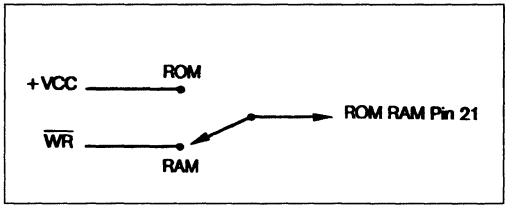


Figure 25 : Selection of RAM/ROM.

III - INTERFACING PERIPHERAL DEVICES

This section contains details of how various devices can be connected to the 8051, both via input/output ports and by memory-mapping techniques. The applications presented were used to build a small, completely self-contained 8051 system that could be used to form the basis of an industrial controller.

3.1. Hexadecimal Keyboard and LED display

A small keyboard consisting of 20 keys in a 4 x 5 matrix was interfaced to the 8051 together with a 7 digit 7-segment LED display. Because the system uses external memory it means that only Port 1 was free for input/output. The final scheme consists of using Port 1 together with a memory-mapped latch to drive both the keyboard and the LED display.

A-LED display

To keep hardware costs to a minimum the LED display was multiplexed. Each of the digits is lit up in turn to give the appearance of a continuous display. To select a digit the segment code must be set up and then the digit turned on. The same code is presented to all digits but only one of them is ever turned on, this process is repeated for each of the digits in turn so that the whole display is refreshed about 40-50 times each second. The LED segments are connected in parallel as follows :

- all segments of the same LED display are connected to the same point. This is normally done within the package, in this case common cathode devices were used, but common anode devices also exist and can be used in much the same manner.
- the same segments of each LED digit are connected in parallel. All the anodes are connected together.

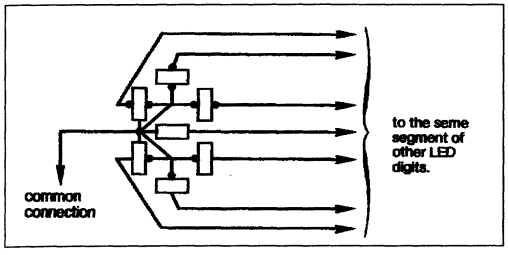


Figure 27 : LED Digit Connections.

This gives rise to two sets of drivers, since MOS or TTL cannot drive LEDs directly. These are digit drivers and segment drivers, which are connected to the common connections of the cathodes and anodes. These are made from discrete transistors because large currents are required. An active high TTL signal switches the transistors on, using PNP transistors will allow an active low TTL signal to turn them on.

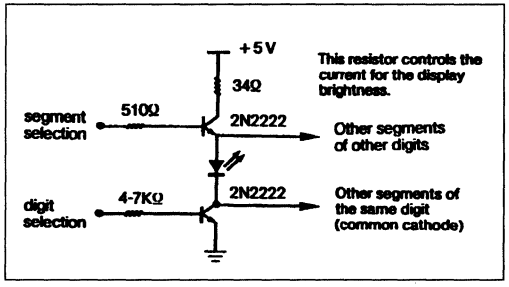


Figure 28 : LED Display Drivers.

Digit selection is accomplished by using a 3 : 8 decoder connected to Port 1 bits P1.5-P1.7. This leaves 5 bits left for the keyboard. Of the eight signals available only seven are actually used. The decoder signals are active low, whereas the drivers use active high signals ; in this prototype inverters were added but the same effect could be obtained by using a PNP transistors as the digit driver.

Segment selection requires an external latch since only 5 bits of Port 1 remain and we require 7-bits (8 if the decimal point is included). The latch used was a 74 LS 374 8-bit latch which is loaded on the rising edge of the clock input. Only very simple decoding was used. Writing to any byte for which A15 is a one (1), i.e., address 8000H upwards, will write to the latch. More sophisticated decoding can be used if the application demands it.

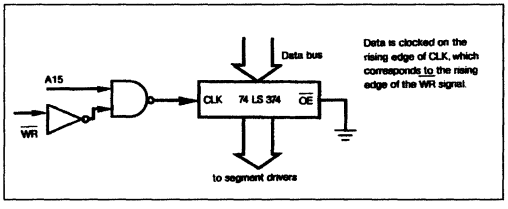


Figure 29 : Latch Configuration.

B-Hexadecimal keyboard

This is a small keyboard organised as a 4 x 5 matrix. It too is connected using the 3 : 8 decoder of the LED display and the rest of Port 1. This allows an expansion of up to 8 x 5 (40 keys) with very little hardware modification.

Pins P1.5-P1.7 are programmed as outputs, controlling the decoder, while P1.0-P1.4 are inputs from the keyboard matrix. Software test to see if any of P1.0-P1.4 are zero (0) and then checks if the combination is valid. If so the value is translated to the key value and presented to the calling program.

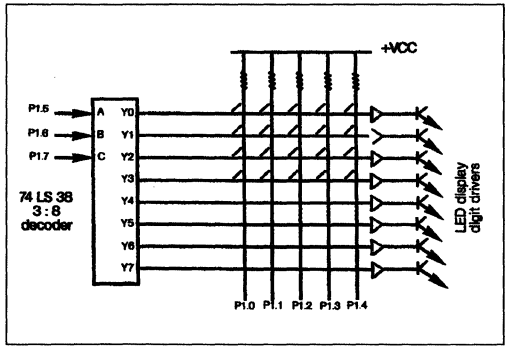


Figure 30 : Keyboard Configuration.

The diagram above shows the actual keyboard and the expansion possibilities.

C-Software

The keyboard scanning and LED display refresh are controlled by the same program. This is called periodically by interruption from one of the 8051 internal timers. To achieve a refresh rate of 40-50 times a second the timer interrupts the main program every 3 ms, since the program only updates a single digit at a time. This 3 ms time delay is also used to provide debouncing of the keyboard which is read by the same routine. The values to be displayed are kept in a table in RAM, changing the value in the table automatically causes the digit of the display to be changed.

The interrupt program used is given in the appendix.

3.2. Parallel input/output

By adding more sophisticated decoding it is possible to have as many latches as required. Bidirectional ports can be constructed by using two latches in the same manner as before.

Both latches are 8282 or 74 LS 374 or similar. The external device buses may also be connected together if they are three-state and bidirectional. Obviously there is no interlock or handshaking between devices so its area of applications is somewhat limited. Adding control via interrupt or status lines can be carried out but this soon becomes expensive and much easier to use LSI peripheral controller such as the 8255.

3.3. Serial input/output

The 8051 provides an UART giving a single full-duplex channel. By itself the UART control cannot provide an RS 232C compatible interface since the voltage levels are incorrect. It is necessary to add line drivers and receivers, suitable circuits being the 75188 and 75189. If more serial channels are desired it is normally necessary to add a UART such as the HD 6402 plus associated decoding and status logic. However if only one channel is in operation at any one time it is possible to multiplex the 8051 serial channel between several actual channels. A two channel system, using a single extra port pin was constructed using this method.

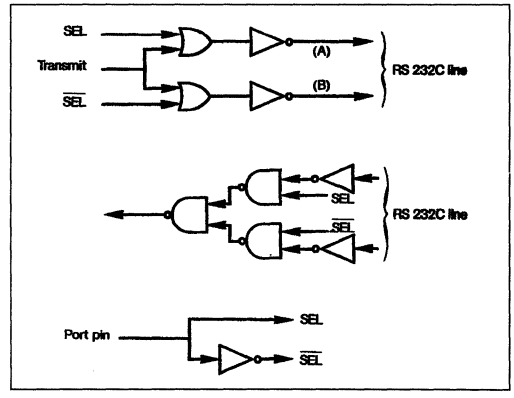


Figure 32 : Multiplexing Circuit for UART.

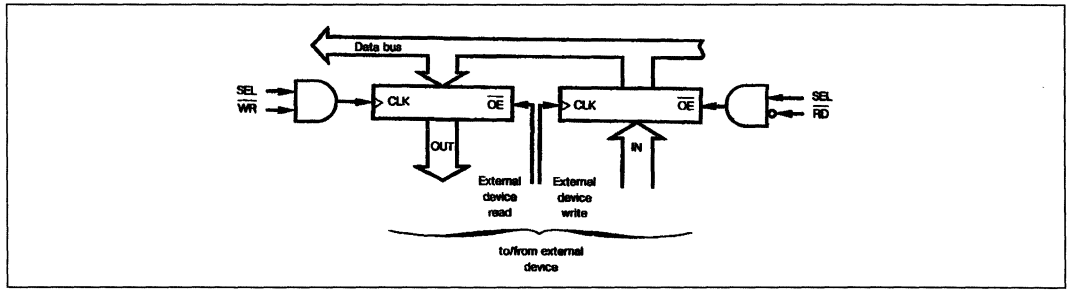


Figure 31 : Parallel Input/Output.

By using more port pins and decoding logic it is possible to expand this system. The only problem to be aware of is when changing channels. If a character is being transmitted it is essential that the routine waits until transmission is finished [indicated by TI being one (1)] before changing the channel. Otherwise some of the character will appear on one line and the rest on another, which will more than likely cause framing errors.

But apart from this point the system works well in practice. By keeping a record of the channel and its speed it is possible to reload the timer every channel change and have multiple channels all operating at different speeds.

IV - CONCLUSION

This application note has attempted to describe and show what can be done using the 8051 and a modest amount of hardware. All of the applications describe have been implemented and debugged. The actual CPU card was built on a single eurocard with another eurocard being used for the keyboard and LED display. At the moment the CPU card is running a Tiny BASIC system that controls an RS 232C terminal and hardcopy device, such as a line printer or cassette recorder. This gives some idea of the applications available. Other applications include :

- consumer products
- medical instruments
- portable instruments
- aero-space applications
- telecommunications
- automotive products
- test equipment
- etc.

In all the 8051 can be used where a small high-performance processor and associated peripherals are necessary up to large applications requiring up to 64 K of RAM and 64 K of ROM.

APPENDIX I

Listing of the interrupt program used to control the keyboard and display.

```
MCS-51 MACRO ASSEMBLER KEYBOARD/DISPLAY REFRESH PROGRAM
ISIS-II MCS-51 MACRO ASSEMBLER V2.0
OBJECT MODULE PLACED IN : FO : APPLIC.OBJ
ASSEMBLER INVOKED BY : : F1 : ASM51 APPLIC.A51 XREF
LOC OBJ LINE   SOURCE
        1   $TITLE (KEYBOARD/DISPLAY REFRESH PROGRAM)
        2   ;
        3   ; _____________________________________________
        4   ;
        5   ; KEYBOARD/DISPLAY SCAN INTERRUPT ROUTINE
        6   ;
        7   ; THIS ROUTINE UPDATES THE SEVEN SEGMENT DISPLAY
        8   ; AND CHECKS THE KEYPAD TO SEE IF A KEY IS BEING
        9   ; PRESSED. IF SO, THE FLAG "DBNCE" IS SET. THIS
       10   ; CAUSES THE ROUTINE TO CHECK THE REST OF THE
       11   ; KEYBOARD TO DETECT MULTIPLE CLOSURES, IT ALSO
       12   ; SERVES TO DEBOUNCE THE CURRENT KEY. WHEN A KEY
       13   ; HAS BEEN SEEN THE FLAG "CHARFND" IS SET TO
       14   ; INDICATE THAT A VALID KEY CODE IS IN THE BYTE
       15   ; "LAST KY". NO MORE KEYS WILL BE READ UNTIL THE
       16   ; KEY IS RELEASED - INDICATED BY CLEARING
       17   ; "CHARFND".
       18   ;
       19   ;       P1.5-P1.7 = NUMBER OF COLUMN OUTPUT
       20   ;       P1.0-P1.4 = NUMBER OF ROW INPUT
       21   ;
       22   ; _____________________________________________
       23   ;
```

7

```
LOC  OBJ       LINE  SOURCE
                 24  ; DEFINITION OF CONSTANTS AND WORKING VARIABLES
                 25  ;
----             26                    DSEG    AT 30 H
                 27  ;
0030             28  LAST_KY:  DS      1               ; LAST KEY READ FROM KEYBOARD IN MATRIX
                 29                                    ; NOTATION.
0031             30  ROW_NO:   DS      1               ; NUMBER OF LINE STROBED ON KEYBOARD.
0032             31  DIGIT:    DS      1               ; NUMBER OF DISPLAY DIGIT CURRENTLY LIT.
0033             32  DATADDR:  DS      1               ; DIGITS OF DISPLAY
  FFFF           33  DISPLAY   EQU     0FFFFH          ; ADDRESS FOR 7-SEGMENT DISPLAY LATCH
  00F4           34  INTVALH   EQU     0F4H            ; TIME BETWEEN INTERRUPTS
  0000           35  INTVALL   EQU     00H
                 36  ;
----             37                    BSEG    AT0
                 38  ;
0000             39  CHARFND:  DBIT    1               ; CHARACTER VALID FLAG
0001             40  DBNCE:    DBIT    1               ; DELAY FLAG
                 41  ;
                 42  ; END OF VARIABLE DEFINITIONS
                 43  ;
----             44            CSEG    AT 0
                 45  ;
0000             46            ORG     RESET
                 47  ;
0000 0195        48            AJMP    START           ; RESET VECTOR
                 49  ;
000B             50            ORG     TIMER0
                 51  ;
000B 0130        52            AJMP    KEYSCAN         ; JUMP TO INTERRUPT ROUTINE
                 53  ;
0030             54            ORG     30H             ; START AFTER OTHER INTERRUPT VECTORS
                 55  ;
0030 C0E0        56  KEYSCAN:  PUSH    ACC             ; SAVE REGISTERS
0032 C083        57            PUSH    DPH
0034 C082        58            PUSH    DPL
                 59  ;
                 60  ; READ THE VALUE FROM THE PORT. IF NO KEYS HAVE BEEN DEPRESSED BITS P1.0-P1.4
                 61  ; WILL BE ONE (1). THIS IS TESTED BY INVERTING AND LOOKING FOR ZERO. IF A KEY HAS
                 62  ; BEEN PRESSED THEN THE DIGIT NUMBER IS ADDED BACK TO CREATE THE FULL CODE.
                 63  ;
0036 E590        64            MOV     P,P1            ; READ PORT-NEW DATA IN P1.4-P1.0
0038 44E0        65            ORL     A,   0E0H       ; MASK-IF NO KEY, ACC IS FFH
003A F4          66            CPL     A               ; INPUT IS THE INVERSE
003B 6015        67            JZ      FINISH          ; TEST FOR A KEY
003D 2532        68            ADD     A,DIGIT         ; FORM FULL ADDRESS-ADD DIGIT NUMBER
003F 200109      69            JB      DBNCE,CHECK     ; TEST DEBOUNCE FLAG
                 70  ;
                 71  ; FIRST DEPRESSION-SAVE THE KEY CODE AND THE DIGIT NUMBER. SET THE DBNCE
                 72  ; FLAG TO INDICATE THIS HAS BEEN DONE.
                 73  ;
0042 F530        74            MOV     LAST_KY,A       ; FIRST TIME-STORE ACC
0404 D201        75            SETB    DBNCE           ; SET DBNCE FLAG
0046 853231      76            MOV     ROW_NO,DIGIT    ; SAVE DIGIT NUMBER
0049 8010        77            SJMP    NO_KEY          ; KEY NOT VALID YET
                 78  ;
                 79  ; IF THE DBNCE FLAG IS SET THEN THIS CODE CHECKS ALL OTHER KEYS IN TURN. WHEN IT
                 80  ; SEES A KEY PRESS IT TESTS FOR THE SAME KEY AS BEFORE, THIS DELAY ALSO PROVIDES
                 81  ; TIME TO ELIMINATE CONTACT BOUNCE. IF IT IS THE SAME KEY THEN A VALID KEY HAS
                 82  ; BEEN SEEN, IF NOT THE DBNCE FLAG IS RESET AND THE PROGRAM RESTARTS.
                 83  ;
004B B53009      84  CHECK:   CJNE    A,LAST_KY,DIFF  ; SAME KEY AFTER DEBOUNCE ?
004E D200        85            SETB    CHARFND         ; YES-CHARACTER VALID
```

```
LOC  OBJ      LINE  SOURCE
0050 8009       86  ;               SJMP    NO_KEY                  ; CONTINUE
                87  ;
                88  ; HERE THERE ARE NO KEYS PRESSED AT ALL. THE CURRENT DIGIT NUMBER IS
                89  ; COMPARED WITH THAT OF THE LAST VALID KEY. IF THEY ARE THE SAME THEN THE KEY
                90  ; HAS BEEN RELEASED AD THE CHARFND FLAG CAN BE CLEARED, OTHERWISE
                91  ; CONTROL PASSES DIRECTLY TO THE REFRESH ROUTINE
                92  ;
0052 E532       93  FINISH:        MOV     A,DIGIT                 ; NO KEYS PRESSED-LOAD PRESENT LINE
0054 B53104     94                  CJNE    A,ROW_NO,NO_KEY         ; SAME AS LAST TIME ?
0057 C201       95  DIFF:          CLR     DBNCE                   ; YES-RESET FLAGS FOR NEXT KEY
0059 C200       96                  CLR     CHARFND
                97  ;
                98  ; REFRESH DISPLAY ROUTINE. LIGHTS EACH DIGIT OF THE DISPLAY IN TURN
                99  ;
               100  ; THE PRESENT DIGIT IS BLANKED BY CLEARING THE SEGMENTS. AFTERWARDS THE NEXT
               101  ; DIGIT IS SELECTED BY INCREMENTING THE DIGIT NUMBER OF P1.5-P1.7, DONE BY
               102  ; ADDING 20H TO THE PRESENT PORT VALUE.
               103  ;
005B 90FFFF    104  NO_KEY:        MOV     DPTR,   DISPLAY         ; ADDRESS OF OUTPUT LATCH FOR DISPLAY
005E E4        105                 CLR     A                       ; BLANK CHARACTER
005F F0        106                 MOVX    @DPTR,A                 ; OUTPUT TO DISPLAY
0060 E532      107                 MOV     A,DIGIT                 ; CURRENTLY LIT DIGIT
0062 2420      108                 ADD     A,  20H                 ; NEXT DIGIT
0064 F532      109                 MOV     DIGIT,A                 ; STORE FOR NEXT TIME
0066 441F      110                 ORL     A,  1FH                 ; SET P1
0068 F590      111                 MOV     P1,A                    ; OUTPUT ACC TO SELECT NEXT DIGIT
               112  ;
               113  ; LOAD THE LATCH WITH THE CORRECT CODE FROM THE TABLE. THIS IS DONE BY
               114  ; SHIFTING THE DIGIT NUMBER AND USING IT TO INDEX THE RAM TABLE CONTAINING
               115  ; THE 7-SEGMENT CHARACTER CODES. REGISTER R0 IS SED TO INDEX THE TABLE
               116  ; AFTER HAVING BEEN SAVED FIRST. IT IS RESTORED AFTERWARDS
               117  ;
006A E532      118                 MOV     A,DIGIT                 ; GET DIGIT NUMBER AGAIN
006C C4        119                 SWAP    A                       ; DIGIT = P1.3-P1.1
006D 03        120                 RR      A                       ; DIGIT = P1.2-P1.0
006E 2433      121                 ADD     A,  DATADDR             ; ADD RAM TABLE ADDRESS
0070 C8        122                 XCH     A,R0
0071 C0E0      123                 PUSH    ACC                     ; SAVE R0
0073 E6        124                 MOV     A,@R0                   ; SEVEN SEGMENT CODE OF DIGIT
0074 F0.       125                 MOVX    @DPTR,A                 ; OUTPUT TO DISPLAY
0075 D0E0      126                 POP     ACC                     ; RESTORE OLD VALUE OF R0
0077 C8        127                 XCH     A,R0
               128  ;
               129  ; RELOAD THE TIMER FOR THE NEXT INTERRUPT IN 3 MS. RESTORE ALL THE REGISTERS
               130  ; AND RETURN FROM THE INTERRUPT.
               131  ;
0078 118E      132                 ACALL   RELOAD                  ; LOAD TIMER AGAIN
007A D082      133                 POP     DPL                     ; RESTORE VARIABLES FROM
007C D083      134                 POP     DPH                     ; STACK AND EXIT INTERRUPT
007E D0E0      135                 POP     ACC
0080 32        136                 RETI
               137  ;
               138  ;
               139  ; END OF INTERRUPT SERVICE ROUTINE
               140  ;
               141  ;
0081 C201      142  LOAD:          CLR     DBNCE                   ; INITIALISE VARIABLES
0083 C200      143                 CLR     CHARFND
0085 753200    144                 MOV     DIGIT,  0
0088 5389F0    145                 ANL     TMOD,   0F0H            ; SET UP KEYBOARD TIMER
008B 438901    146                 ORL     TMOD,   01H
008E 758A00    147  RELOAD:        MOV     TL0,    INTVALL         ; LOAD DELAY
0091 758CF4    148                 MOV     TH0,    INTVALH
```

```
LOC  OBJ    LINE  SOURCE
0094 22      149          RET              ;JUMP TO MONITOR
             150  ;
             151  ;END OF INITIALISATION
             152  ;
             153  ;
0095 1181    154  START:  ACALL  LOAD      ;START OF PROG-SET UP TIMER
             155  ;               ..
             156  ;               ..
             157  ;          REST OF PROGRAM
             158  ;          CONTINUES FROM HERE
             159  ;               ..
             160  ;
             161  ;
             162  ;
             163  ;
             164  ;
             165  END
```

MCS-51 MACRO ASSEMBLER KEY BOARD/DISPLAY REFRESH PROGRAM

XREF SYMBOL TABLE LISTING

NAME	TYPE	VALUE		ATTRIBUTES AND REFERENCES
ACC	D ADDR	00E0H	A	56 123 126 135
CHARFND	B ADDR	0020H.0	A	39 85 96 143
CHECK	C ADDR	004BH	A	69 84
DATADDR	D ADDR	0033H	A	32 121
DBNCE	B ADDR	0020H.1	A	40 69 75 95 142
DIFF	C ADDR	0057H	A	84 95
DIGIT	D ADDR	0032H	A	31 68 76 93 107 109 118 144
DISPLAY	NUMB	FFFFH	A	33 104
DPH	D ADDR	0083H	A	57 134
DPL	D ADDR	0082H	A	58 133
FINISH	C ADDR	0052H	A	67 93
INTVALH	NUMB	00F4H	A	34 148
INTVALL	NUMB	0000H	A	35 147
KEYSCAN	C ADDR	0030H	A	52 56
LAST-KY	D ADDR	0030H	A	28 74 84
LOAD	C ADDR	0081H	A	142 154
NO-KEY	C ADDR	005BH	A	77 86 94 104
P1	D ADDR	0090H	A	64 111
RELAOD	C ADDR	008EH	A	132 147
RESET	C ADDR	0000H	A	46
ROW-NO	D ADDR	0031H	A	30 76 94
START	C ADDR	0095H	A	48 154
TH0	D ADDR	008CH	A	148
TIMER0	C ADDR	000BH	A	50
TL0	D ADDR	008AH	A	147
TMOD	D ADDR	0089H	A	145 146

REGISTER BANK (S) USED : 0, TARGET MACHINE(S) : 8051

ASSEMBLY COMPLETE, NO ERRORS FOUND

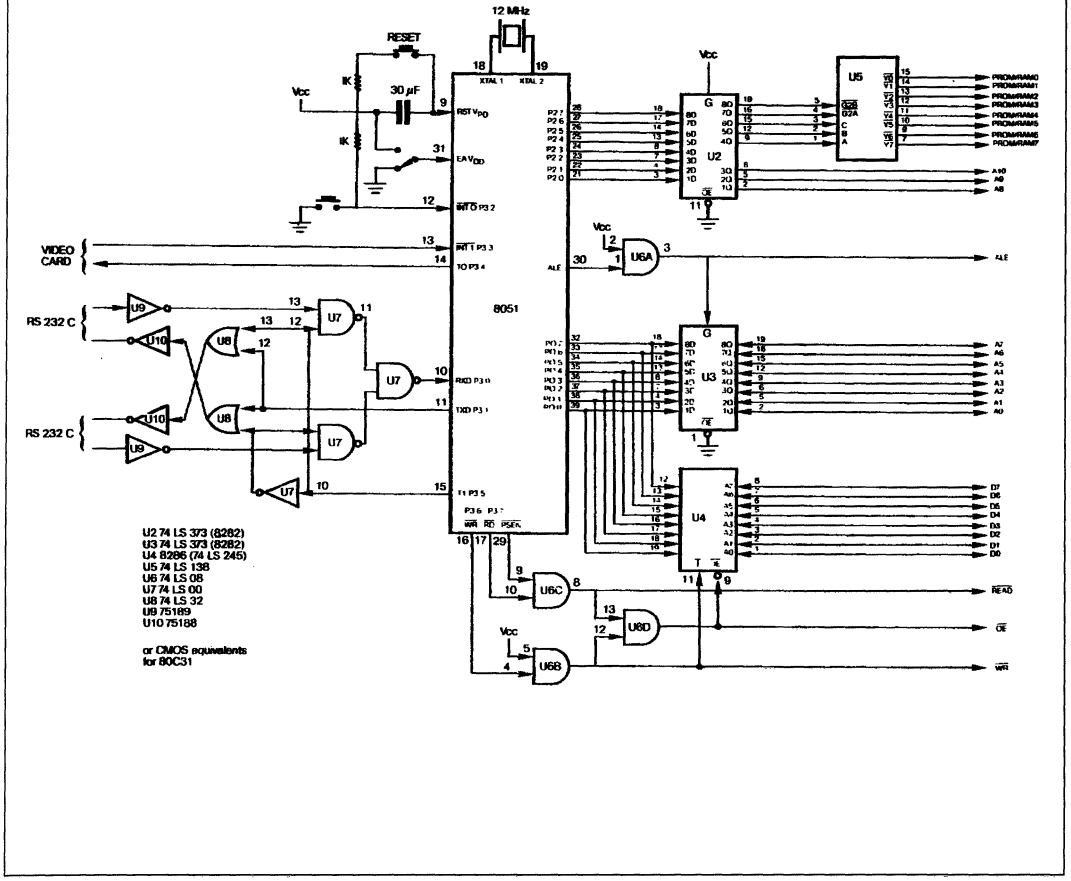
APPENDIX 2

Circuit diagram

The circuit diagram for the CPU card is given overleaf. All connections, except to RAM/ROM are shown. The decoding has been carried out in 2 K blocks allowing the use of HM6516, HM65161 or HM6116 RAMs and the HM6616 PROM. All of these components are directly pin compatible allowing easy system changes or upgrades. Although this 8051 card uses an 8031, it is possible to plug in a preprogrammed 8051. Changing the EA switch setting will cause the internal ROM to be enabled.

Because the program memory time access is about 200 ns the only EPROMs that can be used are the 2732A or 2764. The 2716 EPROMs are not guaranteed to this speed. In practice the prototype system described here works well with 2716 EPROMs from four different manufacturers. It must be stressed that this is under laboratory conditions, working well within the manufacturers specifications. If speed is not a problem, slowing the processor clock will increase the available access time, guaranteeing operation under worst case conditions.

8051 CPU CARD



APPLICATION NOTE

AN1040

DIFFERENCES BETWEEN THE 8051AH AND THE 80C51

This article discusses the few differences which exist between the HMOS 8051AH and the CMOS 80C51 microcontrollers. From the CMOS family's point of view, these differences consist of the addition to the software of two instructions for power saving and, hardwarewise, the inversion of the input from an external clock on inputs XTAL1 and XTAL2, and the increased value of the resitance of input RST.

SOFTWARE DIFFERENCES BETWEEN THE TWO FAMILIES

The 80C51 microcontroller has two additional power saving modes : POWER-DOWN and IDLE.
These additional modes are accessed by the PCON (87H) register. To activate these modes, it is simply necessary to put a "1" in bit PD = PCON. 1 or bit IDL = PCON.0

POWER DOWN

This mode is entered by setting PD = PCON.1 to "1". As shown in figure 1, the oscillator is stopped (PD = 0) and the 80C51's activity is suspended.
While this mode is active, the status of the special registers and the internal RAM is fixed. Only a reset via the RST input enables exit from this status. The time taken by the 80C51 to exit from this mode is equivalent to the duration of the RESET and depends on the oscillator's start-up time, the operating frequency and the quartz used. If tosc is the oscillator's start-up time, the duration of RESET should be equal to tosc + 2 machine cycles.
When the 80C51 returns to the normal mode, only the content of the internal RAM remains unchanged, all the special registers are reconfigured as a result of the RESET.

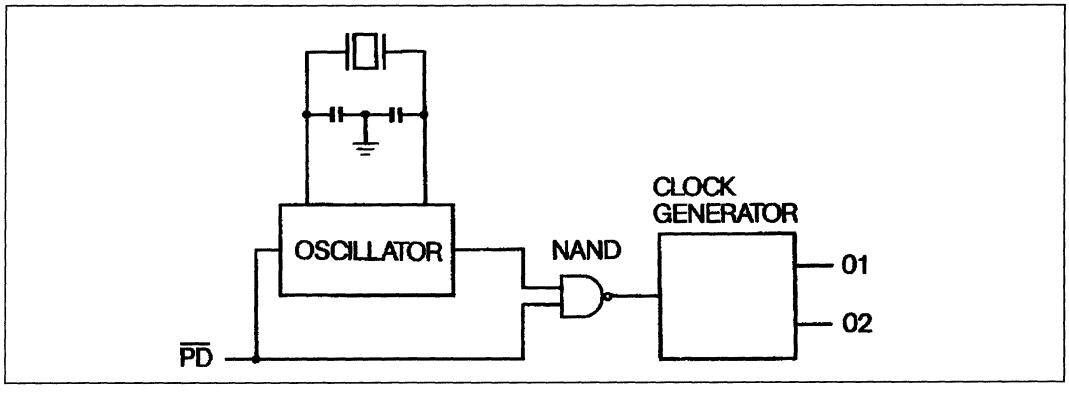


Figure 1.

7

Table 1 shows the content of the special registers after execution of a reset.

REGISTERS	CONTENT
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	07H
DPTR	0000H
P0-P3	0FFH
IP	XXX00000B
IE	0XX00000B
TMOD	00H
TCON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
SCON	00H
SBUF	00H
PCON	0XXX0000B

IDLE MODE

This mode is activated by setting bit IDL = PCON.0 to "1". In this mode (as shown in figure 2), the CPU is no longer driven by the clock and its arithmetical, logical, etc. operations are suspended. Only the TIMERs, the UART and the INTERRUPTS remain under the clock's control.

Power dissipation which was 15mA at VCC = 5V and at 12MHz, drops to 3mA. Once this mode has been entered, the status of the special registers and the internal RAM is frozen.

There are two ways of exiting from this mode, either by a reset or by means of any of the interrupts.

A reset, maintained during 2 machine cycles has the same effects as a normal reset : the internal RAM's content is conserved and the special registers assume the values given in table 1.

The interrupt sub-routine, which enables exit from this mode, will be executed and instruction RETI will result in the continuation in sequence of the instruction placed immediately after the IDLE mode instruction.

DISSIPATION

Table 2 shows the different dissipation according to the operating mode and the technology used.

OPERATING MODE AT 12 MHZ VCC = 5 V	80131/51AH	80C31/ 80C51
Normal	125 mA	15 mA
Power-down		50 µA
Idle		3 mA

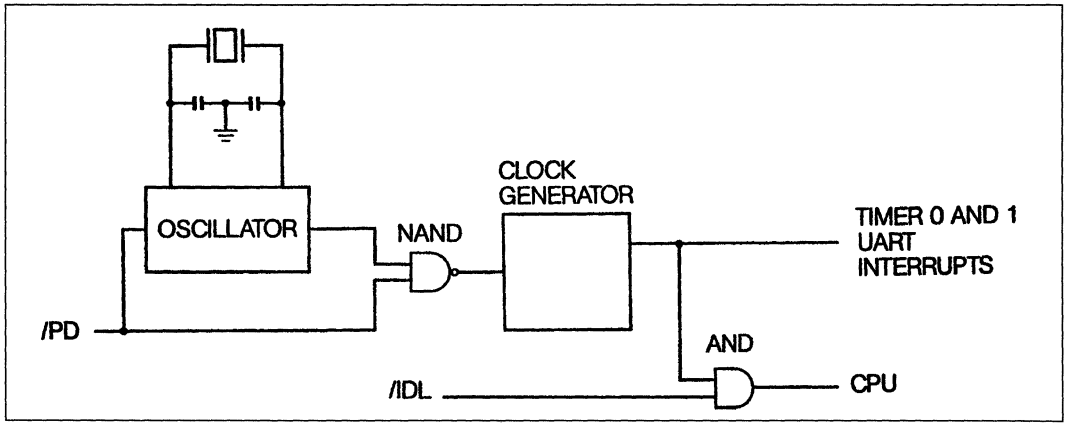


Figure 2.

Table 3 shows the status of the different Inputs/Outputs during the two power saving modes.

MODE	PROGRAM MEMORY	ALE	$\overline{\text{PSEN}}$	PORT0	PORT1	PORT2	PORT3
IDLE	Internal	1	1	Port Data	Port Data	Port Data	Port Data
IDLE	External	1	1	Floating	Port Data	Address	Port Data
Power-down	Internal	0	0	Port Data	Port Data	Port Data	Port Data
Power-down	External	0	0	Floating	Port Data	Port Data	Port Data

HARDWARE DIFFERENCES BETWEEN THE TWO FAMILIES

The impedance of input RST and the input on XTAL1, when an external clock is used, are the differences between the two families.

EXTERNAL CLOCK

A fundamental difference is the case of the driving of the microcontroller by an external clock.
Where as driving is on input 18 (XTAL2) for the 8051AH, it is on input 19 (XTAL1) on the 80C51 *(see figure 3)*.

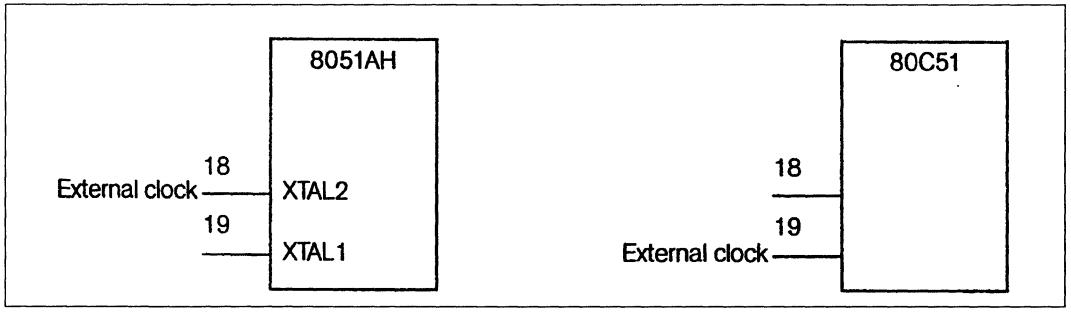


Figure 3.

RST INPUT

The value of the RST input resistor is changed from 8.2 k to 125k in the CMOS version. As a result, the value of the reset capacitance can be lower while conserving the same time constant.

APPLICATION NOTE

AN1041

STATIC WORKING OF MHS C51 MICROCONTROLLERS USED FOR THE SYNCHRONIZATION OF SLOW PERIPHERALS

The MHS C51 family of microcontrollers are built in CMOS technology and are 100 % static. Their feature is that they can operate at very low frequency and can even stop the clock whilst operating without altering execution of the program.

LIMIT OF INTERFACING WITH A SLOW PERIPHERAL

In certain applications, for example that given in *figure 1*, it often happens that the peripheral's access time is longer than the microcontroller's read/write cycle. Therefore, it is necessary that the peripheral can subject the microcontroller's activity to its own speed to ensure correct operation of the cycle.

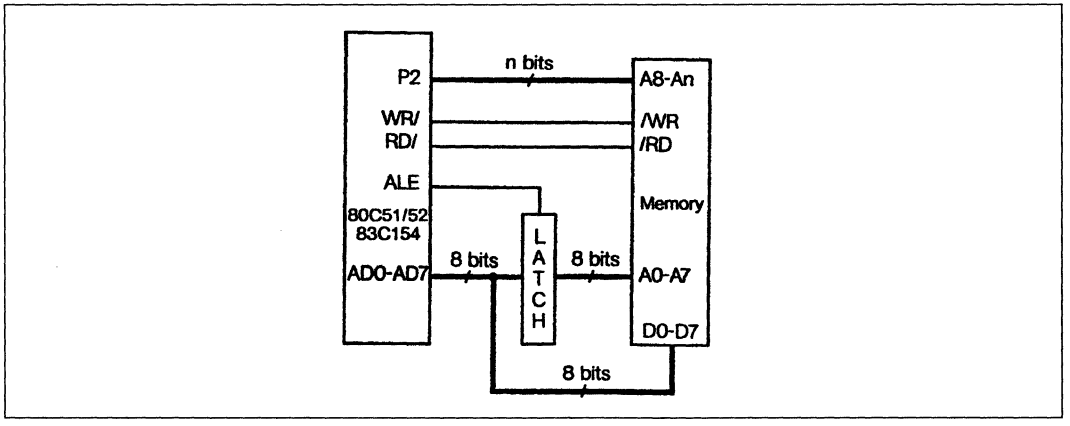


Figure 1 : Interfacing with a peripheral.

The timing diagram in *figure 2* shows the different parameters that must be taken into account in order to solve the problems of peripheral access time.
Thus, one of the parameters, tplatch, (latch address propagation time) introduces a delay which varies according to the latch technology used (cf. TAB1).

LATCH	74LS373	74HCT373	74S373
tpd(ns)	30	49	18

TAB 1. tpd = tplatch according to technology used.

7

READ CYCLE

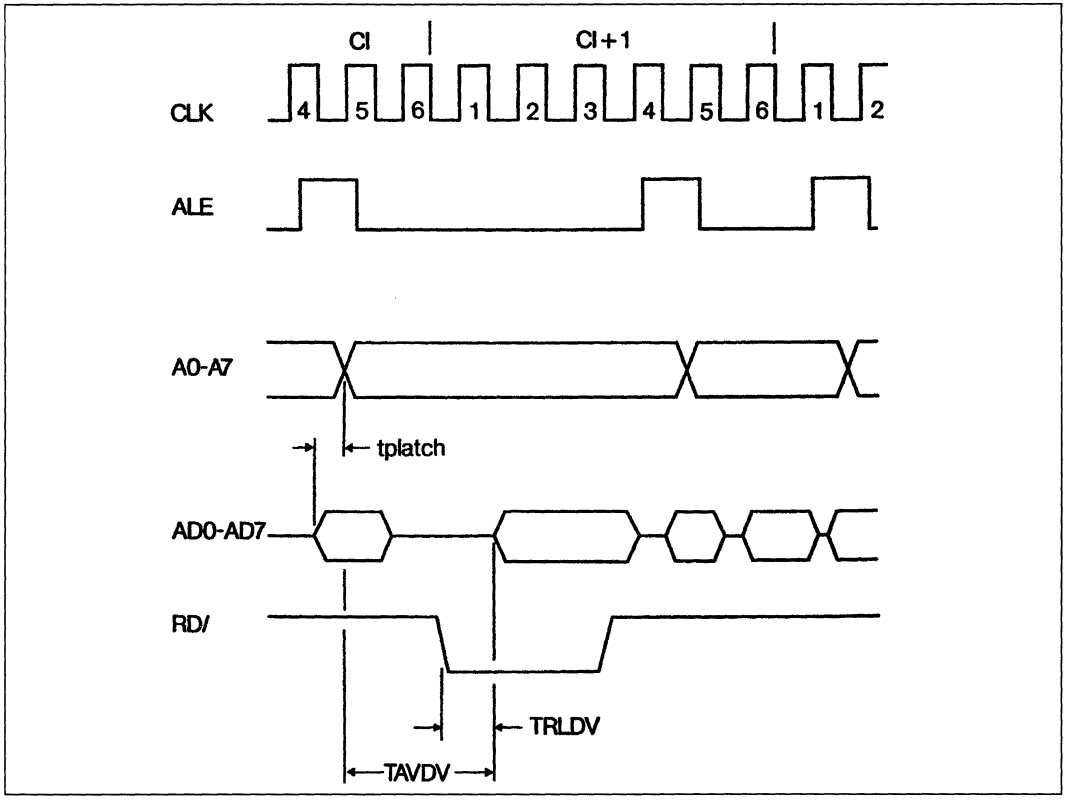


Figure 2 : Peripheral read access.

The peripheral's maximum response time is given by other parameters relating to the microcontroller : TRDLV and TAVDV. Therefore, if the microcontroller's operating frequency is 12 MHz and the latch is a 74HCT193, the peripheral must respond in at least :

- trep < TADV = (9 x TCLCL – 165) – tplatch = 585 – 49 = 563 ns
  in relation to the appearance of the address

and

- trep < TRLDV = (5 x TCLCL – 165) = 251 ns
  in relation to the appearance of the command

If the peripheral's response time is greater that 251 ns or 536 ns, it is necessary to prolong the cycle until the peripheral has completed the command.
For example, if the peripheral's response time in relation to the command is 600 ns, the command cycle must be prolonged by : 600 - 251 = 349 ns or 349/83.3 4 clock periods at 12 MHz.

SOLUTION FOR INTERFACING WITH A SLOW PERIPHERAL

Figure 3 is a schematic of a typical application in which the peripheral controls the microcontroller's by means of the Ready signal.

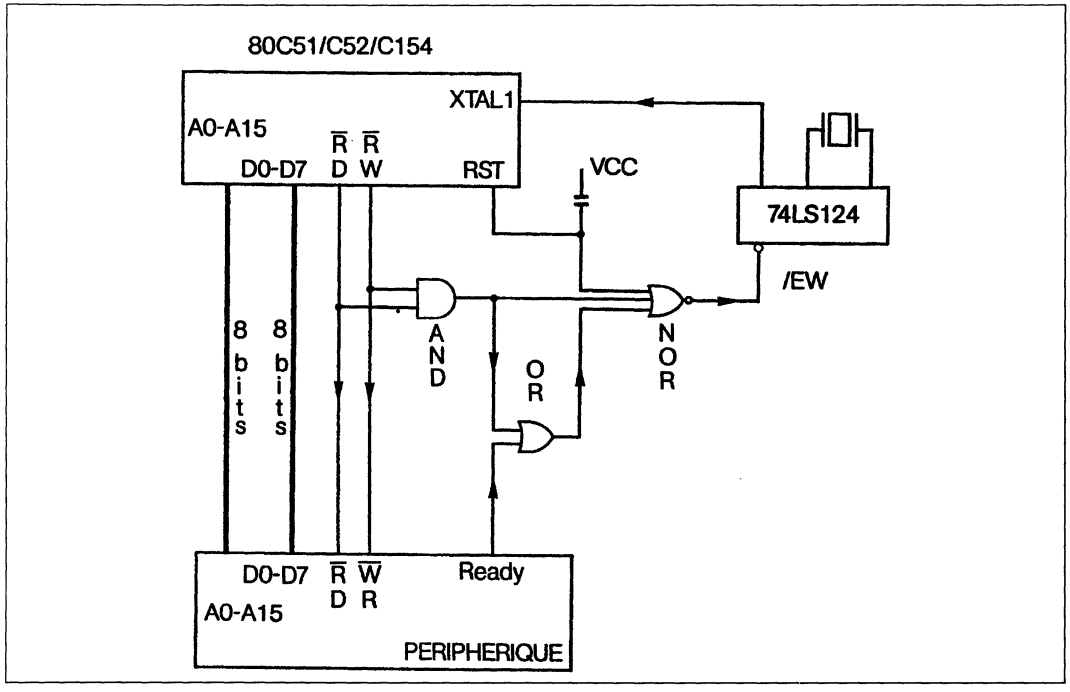


Figure 3 : Interfacing with a slow peripheral.

Figure 4 gives the timing diagrams for operating with a peripheral that is not adapted to the microcontroller's read cycle. The data arrives too late on the data bus D0-D7. Therefore, the access time, TRDLV and TAVDV, must be prolonged by a value equivalent to twait.
The schematic in *figure 3* enables prolongation of the read/write cycle for as long as the peripheral requires in order to execute the command generated by the microcontroller.

The microcontroller is equipped with an external oscillator, built with a 74LS124, driven by a quartz and connected to XTAL1. The 74LS124 oscillator's operation can be blocked by means of its /EW input. A "1" applied to this input forces the 74LS124 to terminate the current clock cycle and to stop on a low level. A "0" level on input/EW reinitiates the clock which effectively restarts on a level "1". Replacement of this circuit by a 74S124 is not possible, once enabled this circuit requires several cycles before it becomes stable.

Therefore, the particularity of this circuit is to provide a stop and restart specific to the microcontroller.

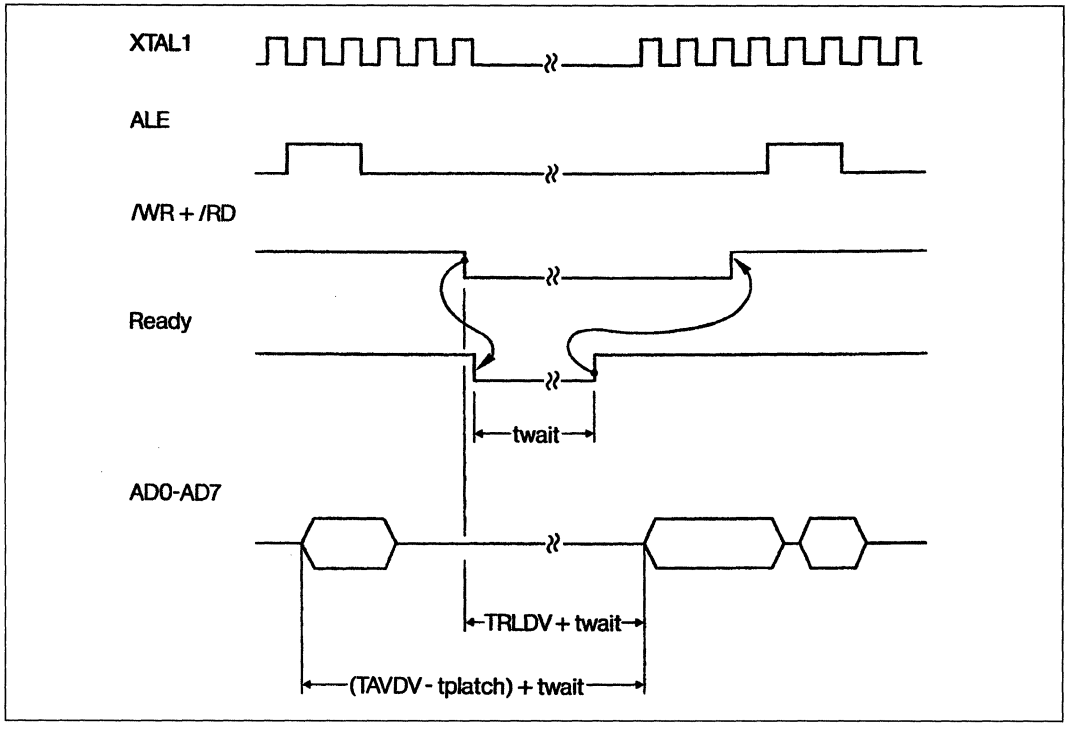


Figure 4 : Timing diagrams for Figure 3.

At the beginning of the read/write cycle, the peripheral's Ready output is at "1" and input/EW is at "0". When a command is generated, /WR or /RD = 0, input /EW goes to "1" and the oscillator stops effectively on completion of its current clock cycle, setting its Ready output to "0". At the same time, the command synchronizes the Ready signal's transition to level "0". After prolonging the read cycle by twait, the Ready signal's transition to "1" indicates to the microcontroller that the peripheral has terminated execution of the command and that it can complete the current cycle.
Another solution, if no 74LS124 is available, is to use the schematic shown in *figure 5.*

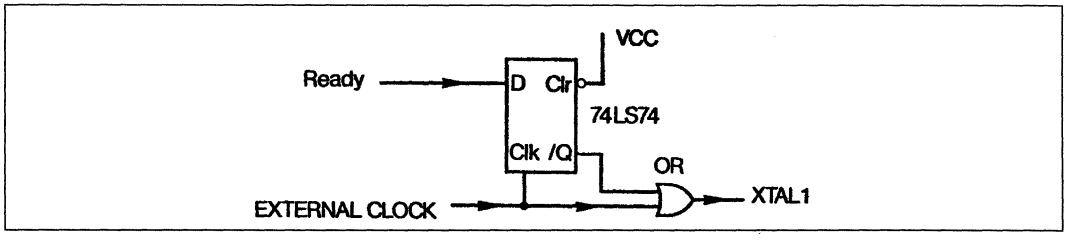


Figure 5.

This circuit performs the same functions as the 74LS124 circuit, that is to effectively stop and restart the microcontroller. The timing diagrams in *figure 6* show the electrical operation of this circuit.

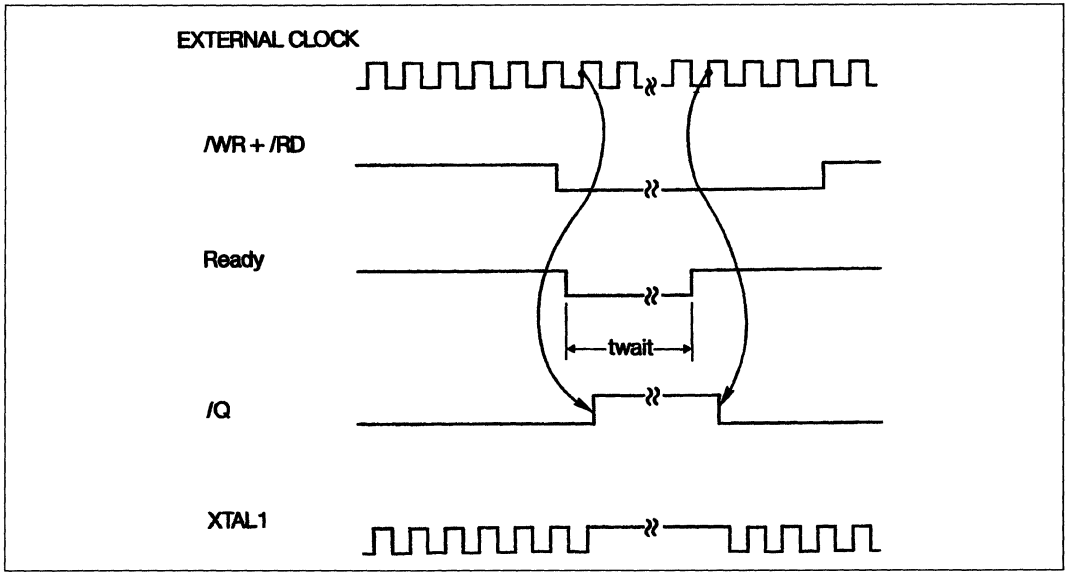


Figure 6.

The peripheral controls the Ready signal and several circuit can be used. *Figures 7 and 8* show two of the many possible solutions.

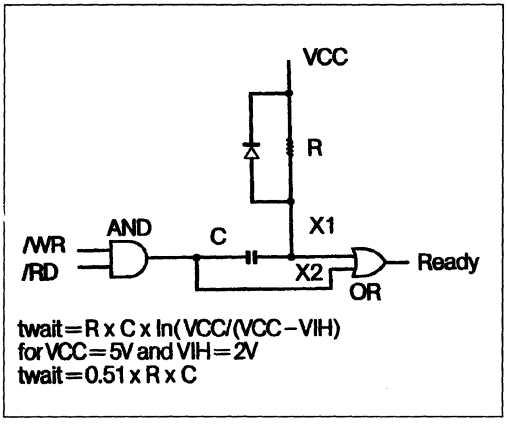


Figure 7 : Solution 1.

Before the command is generated (/WR and /RD are at "1") the two inputs X1 and X2 are at 1 and the Ready output is at "1". A write or read access changes the state of inputs X1 and X2 (X2 = 0, X1 = 0) and the Ready output goes to "0". Input X1 evolves at the speed of the time constant RC. When the voltage reaches the threshold voltage VIH (2 V) at the end of a time equivalent to twait, X1 = 1 and X2 = 0 and the Ready output returns to "1". The microcontroller terminates the current command by raising the write/read signal.

MONOSTABLE Timer

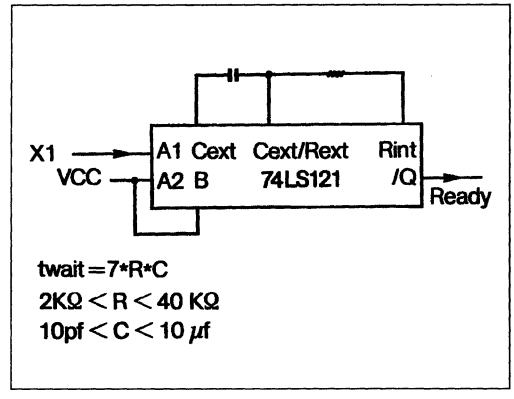


Figure 8 : Solution 2.

CONCLUSION

The staticity of the MHS C51 microcontrollers makes it possible to provide simple interfacing for any peripheral, fast or slow.

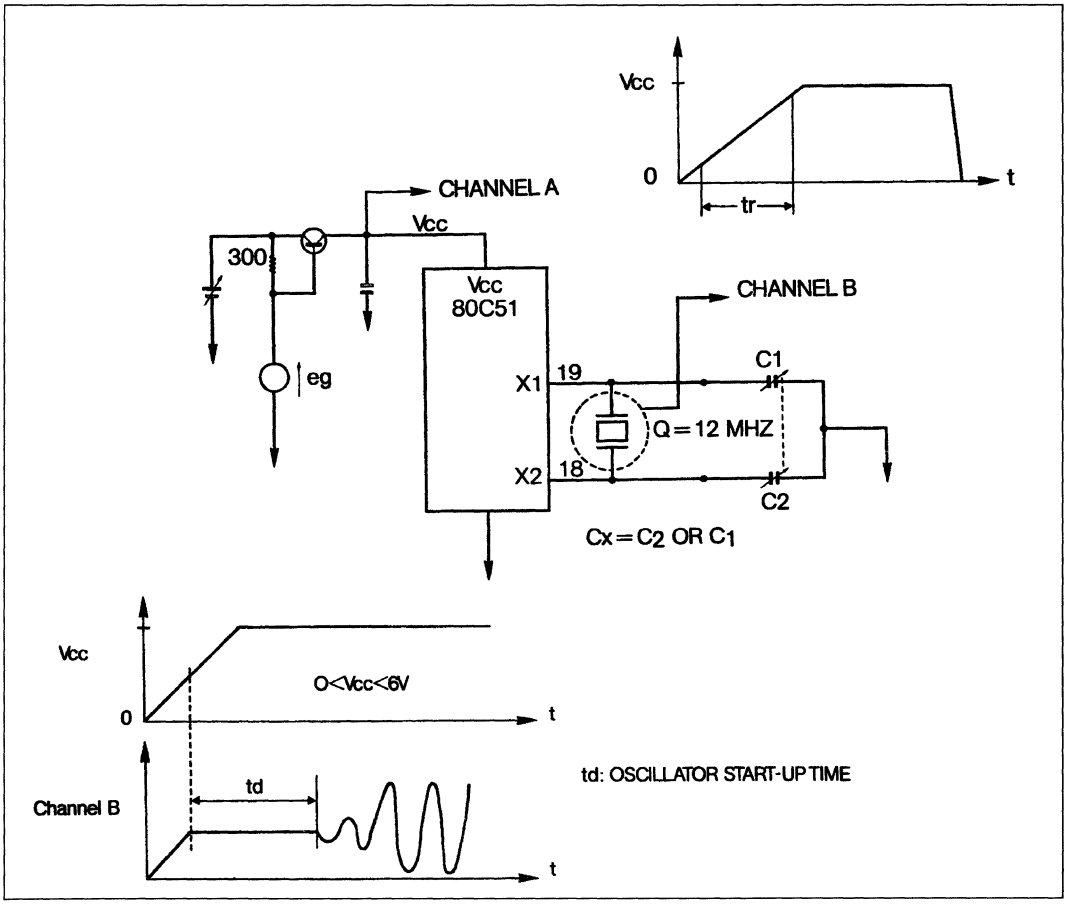
APPLICATION NOTE

AN1042

80C51 OSCILLATORS

OSCILLATOR START-UP

- The 80C51's oscillator start-up time depends on two main parameters :
  - the characteristics of the power supply,
  - the values of the capacitances placed at the terminals of the quartz.
- The continuation of this article discusses the oscillator start-up time as a function of these two parameters.



1. Test Circuit.

7

2. Power supply characteristics

- At the given capacitance Cx, the oscillator's start-up time depends on the value of the power supply and its rise time.
- Value of Vcc
  - For a given rise time, the start-up time td decreases when the supply voltage Vcc increases.

V_{CC} (V)	3	4	5	6
td (ms)	2	1.5	1.2	1

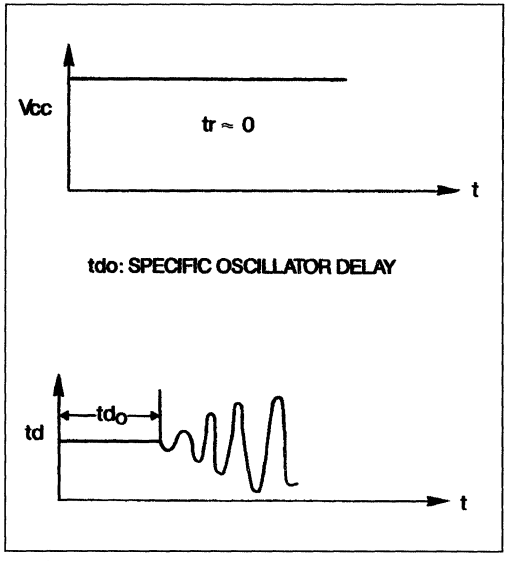
Cx = 54 pf ; tr = 1 µs

- Vcc rise time
  - For a given Vcc value, the start-up time td increases when the rise time tr increases.

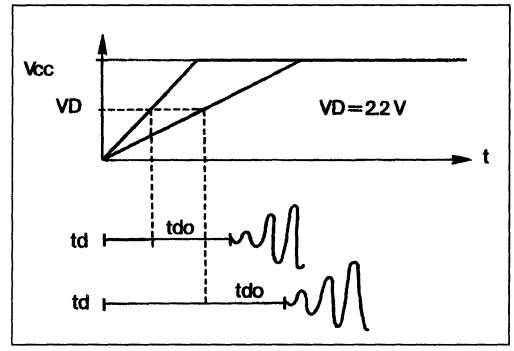
tr (ms)	.001	.01	.1	1	10	100
td (ms)	1.2	1.5	1.6	2.4	7	64

Cx = 54 pf ; VCC = 5 V

- Comment : on powering-up, the oscillator has its own internal delay. It can be measured by approaching the rise time of the power supply close to zero.



- Conclusion : The power supply's rise time causes an additional delay to the oscillator's rise time.



VD : is the minimal supply voltage required by the 80C51 in order to start its oscillator.

3. Capacitance Cx at the terminals of the quartz

- For a given power supply characteristic, the oscillator's start-up time td increases when capacitance Cx increases.

Cx (pf)	27	54	81	108
INTEL (ms)	.4	.8	1.6	3.2
MHS (ms)	.6	1.2	2.4	3.2
OKI (ms)	.2	.4	1.2	2.4
PHILIPPS (ms)	.4	1.2	2.4	4

V_{CC} = 5 V ; tr = 1 µs

Cx (pf)	27	54	81	108
INTEL (ms)	.75	1.5	3	6
MHS (ms)	1	2	4	8
OKI (ms)	.5	1	2	3
PHILIPPS (ms)	.6	1.8	3.6	7.2

V_{CC} = 3 V ; tr = 1 ms

4. Synthesis

The table below shows the parameters to be manipulated in order to increase or decrease the oscillator's start-up time.

	tr	Vcc	Cx
td ↗	↗	↘	↗
td ↘	↘	↗	↘

7

APPLICATION NOTE

AN1043

DIFFERENCES BETWEEN THE 83C154 AND THE 80C52

The 83C154 is an 8-bit microcontroller belonging to the MHS C51 family of microcontrollers. Its instruction set and the number of functions implemented are fully compatible with this family. The innovations concern the size of RAM available, 16 Kbytes instead of 8 and the new functions listed in table 1.

- TIMER 1 and TIMER 0 :
32-bit TIMER/COUNTER,
WATCH DOG,
Asynchronous counting in POWER-DOWN mode.

- PORTS, 1, 2, 3 :
Choice of output resistance value.

- UART :
Receive error detection.

- POWER-DOWN MODE :
Software and hardware control.

- IDLE MODE :
New possibility for exiting from this mode.

- INTERRUPTS :
New mode.

IOCON

Figure 1 shows the correspondence between the 83C154's new functions and the different bits of the IOCON register (0F8H) which can only be addressed by bit.

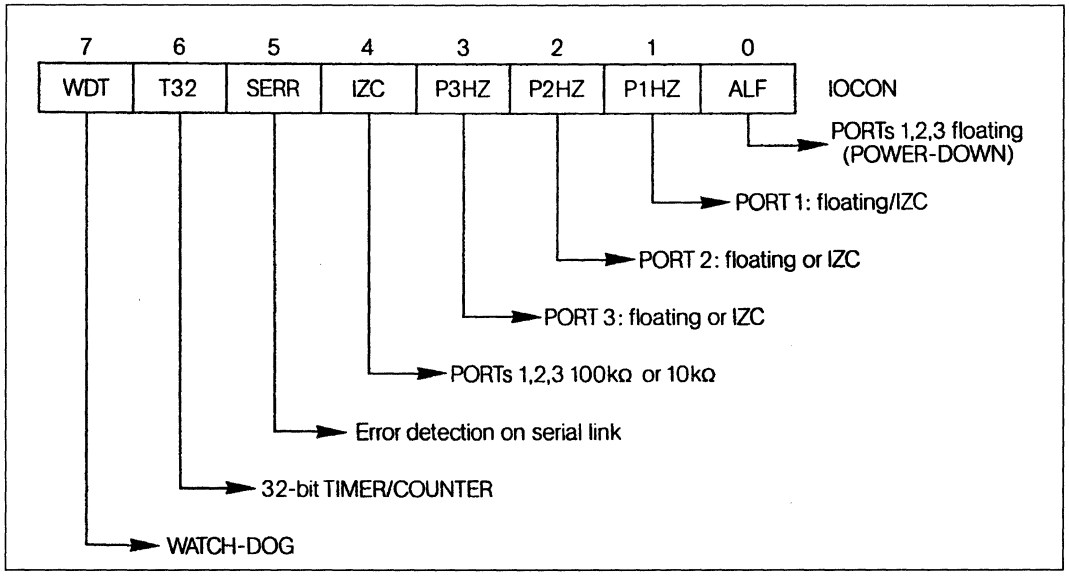


Figure 1.

7

BITS RPD AND HPD

Two supplementary bits of register PCON, RPD and HPD, are used to provide the additional management functions for the POWER-DOWN and IDLE modes.

Figure 2 shows the correspondence between the PCON register bits and the new power saving modes. PCON is not bit-addressable.

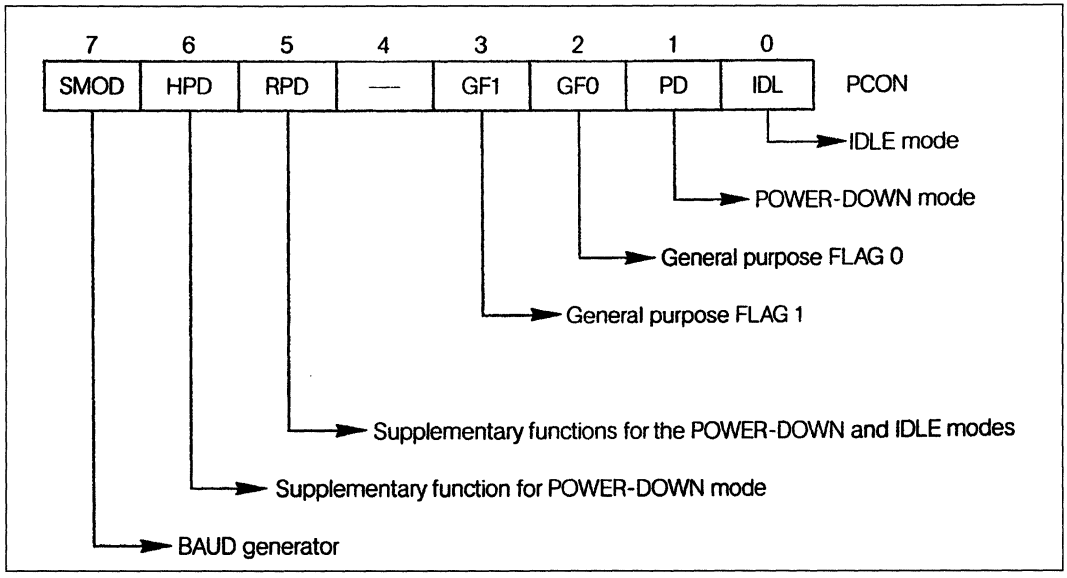


Figure 2.

TIMER/COUNTERs 0 AND 1

The 83C154 has three 16-bit TIMER/COUNTERS, TIMER 0, TIMER 1 and TIMER 2. The architecture and instruction set of these three TIMERS are compatible with the MHS C51 family. In addition to the 4 existing modes, 2 other modes have been added for TIMER 1 and TIMER 0 :

- a WATCH-DOG mode,
- A 32-bit TIMER/COUNTER mode.

These new modes are explained in detail below.

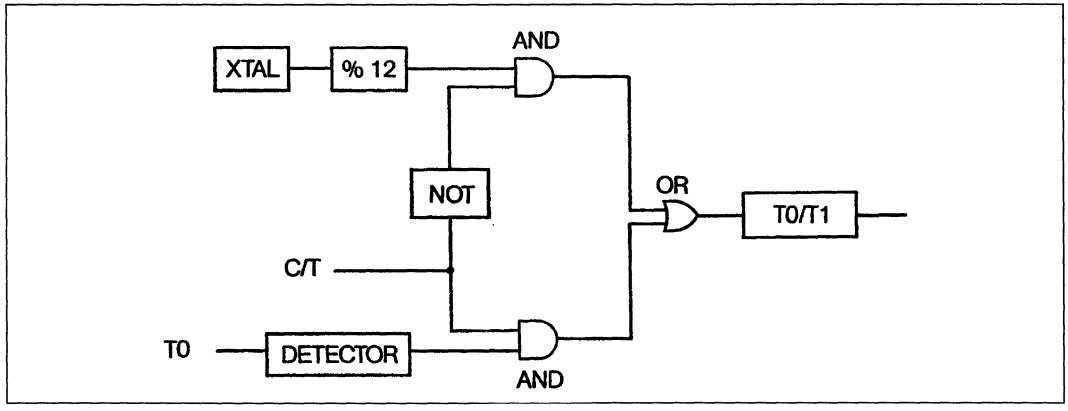
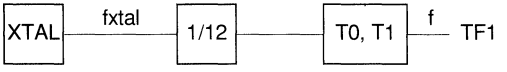


Figure 3.

32-BIT MODE :

This mode is activated by setting bit T32 of register IOCON (ICON.6 = 0FEH) to " 1 ". This action causes TIMER 0 AND TIMER 1 to be configured as a 32-bit TIMER/COUNTER and this, whatever the value of the configuration register TMOD. TIMER 0 constitutes the LSBs and TIMER 1 the MSBs. Two sources provide control of this 32-bit TIMER/COUNTER, either the 83C154's clock (TIMER mode) or an external clock connected to input T0 (COUNTER mode). This selection is made by programming bit C/T0 of register TMOD (089H). If (C/T0) = 0, the 83C154 is in TIMER mode and if (C/T) = 1, it is COUNTER mode.
Counting is started when bit (T32) = 1 and is stopped by complementing T32. If the TIMER/COUNTER is to be stopped, when restarted and stopped again, care must be taken to ensure that bits TR0 and TR1 are programmed with the value 0. The contrary would result in the restarting of one of the two TIMER/COUNTERs which would modify the content of the 32-bit TIMER/COUNTER Bit TF1 enables detection of a TIMER/COUNTER overflow. The following formulae are to be used for calculating the required frequency :

32-bit TIMER MODE



$$f = \frac{fxtal}{12 \times (65\,536 - (T0, T1))}$$

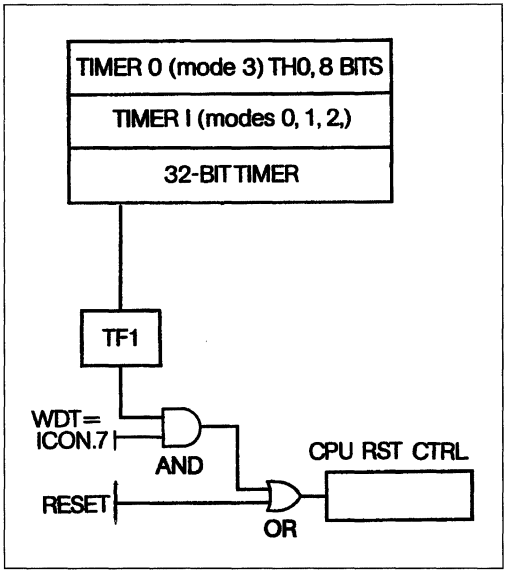
32-bit COUNTER MODE

$$f = \frac{fext}{65\,536 - (T0, T1)}$$

$$fext < \frac{fxtal}{24}$$

In order to be able to increment its counter, the 83C154 must detect a complete signal at its input, that is to say a succession of two transitions. On each machine cycle the 83C154 samples its T0 input (fxtal/12). Therefore, to increment its counter, it must read its T0 input at least twice, in other words a minimum time of 24 clock periods. Thus, the maximum frequency of signal fxtal is less than or equal to fxtal/24.

WATCH-DOG MODE



This mode is activated by setting bit WDT of register ICON (ICON.7 = 0FFH) to " 1 ".
Several configurations are possible, but always based on TIMERs 0 and 1 :

- TIMER 0 : program in mode 3, TH0 is seen as an 8-bit TIMER and is controlled by TR1.
- TIMER 1 : can be programmed in mode 0, 1 and 2.
- TIMER 32 : special 83C154 mode which combines the bits of TIMER 0 and TIMER 1 to form a single 32-bit TIMER.

Whatever the chosen configuration, the WATCH-DOG can be controlled either by an internal source (C/T = 0) or by an external source (C/T = 1). The TIMER is started by setting bit TR0 or TR1 or TR32 of register TCON or ICON to " 1 ". A timer overflow is detected by flag TF1 (TF1 = 1) of register TCON (TCON.7 = 08FH). When an overflow occurs (TF1) = 1, the 83C154 is reset immediately. This action has the same effects as a hardware reset. As there are no precautions for protecting bit WDT, special care must be taken during program writing to avoid accidental manipulation of this bit. In particular, the user should use the IOCON register bit manipulation instructions :
- SETB X and CLR X
in preference to the byte manipulation instructions :
- MOV IOCON, #XXH, ORL IOCON, #XXH, ANL IOCON, #OXXH,.....

7

EXTERNAL COUNTING IN POWER-DOWN MODE

In POWER-DOWN mode, the oscillator is stopped and the 83C154's activity is frozen. However, if an external clock is connected to one of the inputs T1 or T0, implementation of the functions of TIMER 0 and TIMER 1 can continue. In this case, counting is asynchronous and the maximum, admissible signal frequency on input T1 or T0 only depends on the counter's intrinsic constants. Overflow of one of the counters, TF0 = 1 or TF1 = 1 will either trigger the interrupt or will force a reset if the counter is programmed in the WATCH-DOG mode (COUNTER 1 only). In both cases, the overflow of one of the two counters results in exit from the POWER-DOWN mode.

POWER SAVING MODE IDLE MODE

This mode is 100 % compatible with that of the 80C52 and has an additional function. This mode is software-controlled. Entry and execution are implemented by setting the IDL bit to " 1 ". Exit from this mode is controlled by bit RPD of register PCON and the interrupt register IE :
- RPD = 0

If no interrupt is enabled, the only possibility of exiting from this mode is a reset of the 83C154.
If the interrupts are enabled, exit from the mode can be made either by interrupt or by reset.
- RPD = 1
- Whether enabled or not, an interrupt request causes the 83C154 to exit from the POWER-DOWN mode.
- If no interrupt is present, only a reset will cause the 83C154 to exit from this mode.

Table 2 summarizes the different types of operation of this mode.

INPUT CONDITIONS		OUTPUT CONDITIONS		
IDLE	IDL	RPD	INTERRUPTS	RST
SOFTWARE	1	0	if authorized	YES
	1	1	Authorized or not	YES

Table 2.

POWER-DOWN MODE

This mode is controlled by :
- software by bits PD, RPD and the IE register,
- hardware by bit HPD.

On entry into this mode, the clock is stopped and the 83C154's activity is suspended. However, the UART ant TIMER (0/1) functions continue to work if :
- an external clock is connected to one of the inputs T0 or T1
- register TMOD is programmed correctly (C/T = 1).

HARDWARE CONTROL

Hardware control (HPD = 1)

This mode is entirely software-controlled by an external signal connected to input T1. The trailing edge of this signal activates the POWER-DOWN mode (after the current instruction has been executed). The leading edge of this same signal (T1) or a reset enables the 83C154 to quit this mode. Interrupt requests, even if enabled, do not enable exit from this mode.

SOFTWARE CONTROL

- Entry to mode (PD = 1)

The POWER-DOWN mode is entered when bit PD of register PCON is at " 1 ".
- Exit from mode

Exit from this mode is controlled by bit RPD.
(RPD = 0)
If the interrupts are not enabled, the only means of exit from this mode is to apply a reset to input RST.
(RPD = 1)
Whether the interrupts are enabled or not, an interrupt request or a reset causes the 83C154 to quit this mode.
If the interrupt requests are enabled by the IE register, execution of the program continues with the servicing of the interrupt sub-routine.
If the interrupt requests are not enabled, the instruction following the POWER-DOWN mode instruction is executed.
Table 3 summarizes the different types of operation in this mode.

INPUT CONDITIONS				OUTPUT CONDITIONS			
POWER-DOWN	HPD	PD	T1	T1	RPD	INTERRUPTS	RST
SOFTWARE	0	1	X	X	0	if authorized	Yes
	0	1	X	X	1	Authorized or not	Yes
HARDWARE	1	0			X	X	Yes
HARDWARE and SOFTWARE	1	0			0	If authorized	Yes
	1	1			1	Authorized or not	Yes

X = without action

Table 3.

COMMENT

In the case of mixed-hardware software working, the POWER-DOWN mode can be entered by means of HPD = 1 or PD = 1.
When RPD = 1, exit from the mode occurs when T1 returns to 1 and when an interrupt request is generated. Otherwise the only way of quitting the mode is to apply a reset to input RST.

It is possible to operate the POWER-DOWN and IDLE modes in parallel. Exit is only possible when exit T1 goes high and if, with RPD 1 = 1, an interrupt request has been generated. Otherwise the only possibility of exiting is to generate a reset on input RST.

SERIAL LINK

The 83C154 has all four of the 80C52's operating modes, with an addition :

- FRAME and OVERRUN error detection,
- Operation (mode 1, 3) in POWER-DOWN and IDLE mode.

FRAME ERROR

This function enables detection of a transmission error in the format of a received character. Arrival of a character is detected by the trailing edge of the character start bit. All received bits are sampled on the 7th, 8th and 9th bits of the receive clock (16 or 32 times the reception speed). A majority vote is taken on these 3 bits to determine if the received bit is a "1" or a "0". If a "0" is read in place of a stop bit (which is always at "1", there is an error in the transmission format and bit SERR (SERR) = 1 of register IOCON (IOCON = 0F8H) is set at "1". The timing diagram below represents a character with its stop bit missing. A format error is signalled by bit SERR.
Figure 3 gives the timing diagrams for a serial gate presenting this error.

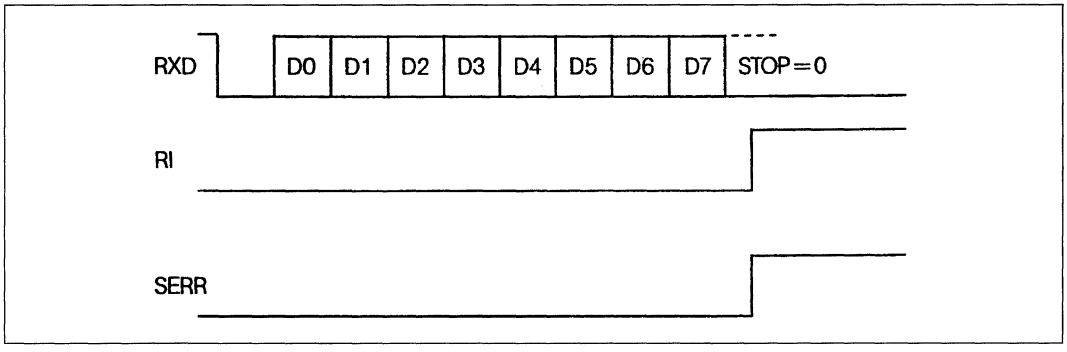


Figure 3.

As in the case of the RI flag, the SERR flag is reset to zero by the software.

OVERRUN ERROR

This function detects when a received character has not been read and has been replaced by another character. Reception of a character is signalled to the 83C154 by raising the RI flag to "1". This flag stays at "1" until the user resets it to "0" (CLR RI). If the next character is sent before the previous character has been read, an error is detected and bit SERR of register IOCON (IOCON = 0F8H) is set to "1".
Figure 4 shows the timing diagrams of the serial link for this error.

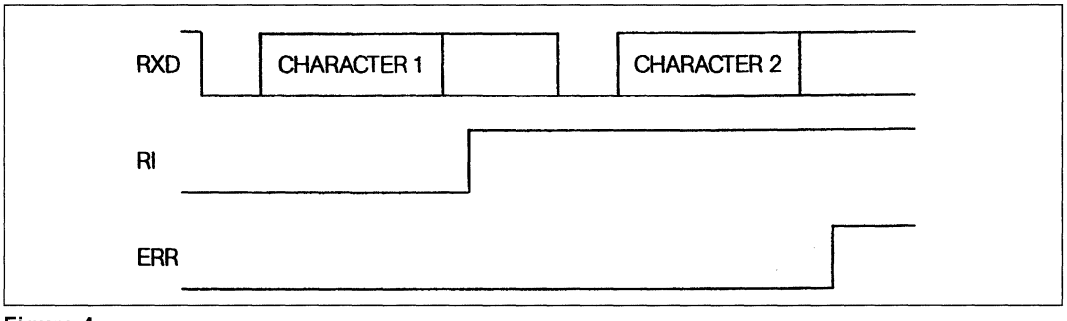


Figure 4.

As in the case of the RI flag, the SERR flag is reset to zero by the software.

POWER-DOWN AN IDLE MODE

The serial link is able to run in power down and idle mode. As the CPU clock is frozen, only the UART mode 1 and 3 are operationnal.

The transmission clock has to be generated with Timer 1 and use the external clock ((C/T) = 1, ((Gate = 0)). Max frequency will be : Fext (Fxtal/24).
Fext ≤ OSC/24

I/O PORT

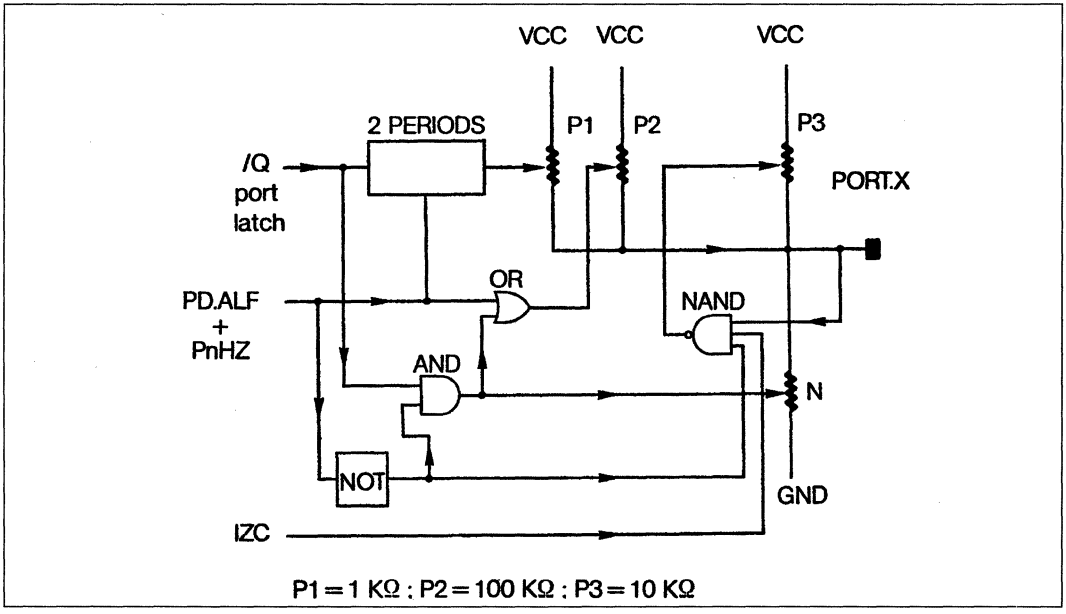


Figure 6.

The I/O drives for P1, P2, P3 of the 83C154 are impedance programmable.
The I/O buffers for ports 1, 2 and 3 implemented as shown in *Fig. 6.*

The impedance can be programmed through the register IOCON (IOCON = 0F8H).
Table 4 is a detail of register IOCON showing PORT impedance selection.

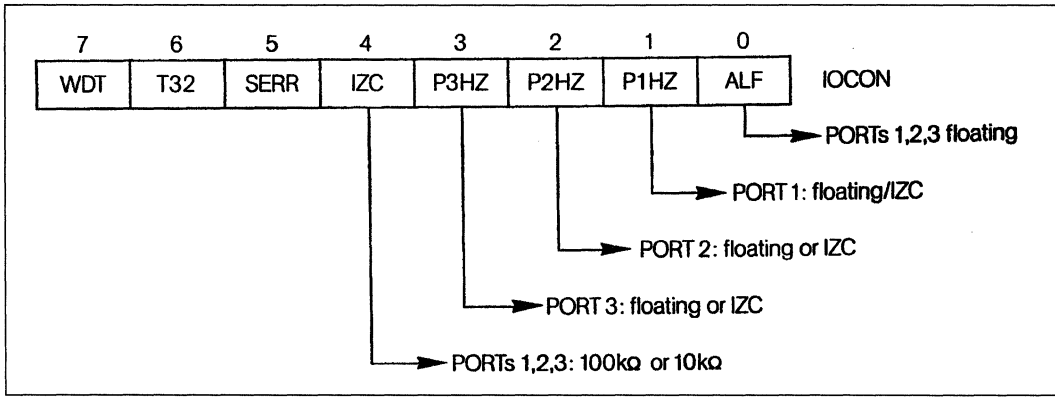


Table 4.

There is a choice of three possible resistance values : 10 k, 100 k, and floating.
ALF = 1, all the PORTS (1, 2 and 3) are floating in POWER-DOWN mode.
P1HZ, P2HZ or P3HZ = 0 the output resistance depends on the choice of IZC.
P1HZ, P2HZ or P3HZ = 1 the PORT is floating.
IZC = 0, the output resistance is 10 kΩ.
IZC = 1, the output resistance is 100 kΩ.
Table 5 below is a summary of the possibilities offered by register IOCON.

ALF	IZC	PnHZ	Pn
0	0	0	10 kΩ
0	0	1	F
0	1	0	100 kΩ
0	1	1	F
1	X	X	*F

F = FLOATING ; X = 1 or 0 ; $1 \leq n \leq 3$
* in POWER-DOWN mode

The 83C154's IP register (0B8H) has a new function as the possibility of making the interrupt level identical for all types of 83C154 interrupts.

Programming of bit PCT (PCT = 1, PCT = IP.7) gives all interrupts the same level.
Table 6 is a detail of the IP register.

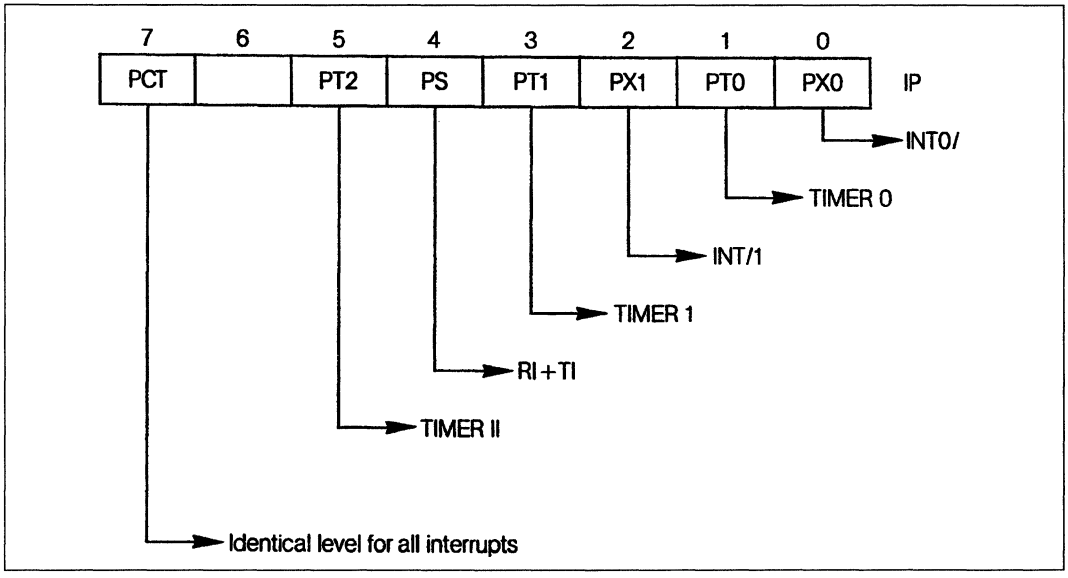


Table 4.
All the bits of this register can be addressed directly.

7

APPLICATION NOTE

AN1044

DIFFERENCES BETWEEN OKI AND MHS 83C154s

MHS's 83C154 is a development of INTEL's 80C51/52 family of microcontrollers. All the basic mechanisms (interrupts, I/O, etc.) of MHS's 83C154 are 100 % compatible with those of the INTEL 80C51/C52 family.

There are several incompatibilities between the basic mechanisms of the OKI 83C154 and the MHS 83C154. In practice, these differences are invisible to the user. They are listed hereafter :

- Division by zero.
- Conditional jump.
- Long jump on a call from a sub-routine.
- Serial port.
- Port writing.
- TIMER interrupt request.

These differences are discussed in the following paragraphs.

DIVISION BY ZERO

Division by zero is performed by putting the numerator in register B and the denominator in register A. The result in the division is stored in register A and the remainder in register B. The difference between MHS and OKI is in the result and is shown below :

A div B → result = A remainder = B

MHS B = FF OKI B = 00

CONDITIONAL JUMP

Conditional jumps JC, JNC, JZ and JNZ are single byte instructions that execute in 2 machine cycles. Before branching to the new address :

- MHS increments the PC twice.
- OKI increments the PC once.

LONG JUMP

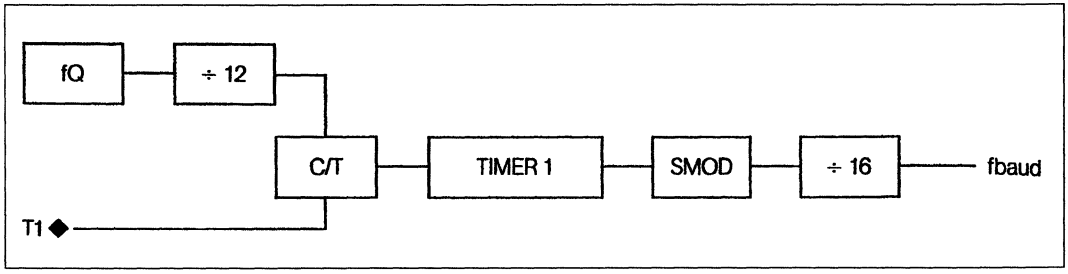
The long jumps, LCALL and LJMP are three-byte instructions that execute in 2 machine cycles. Before branching to the new address :

- MHS increments the PC 3 times,
- OKI increments the PC twice.

7

SERIAL PORT

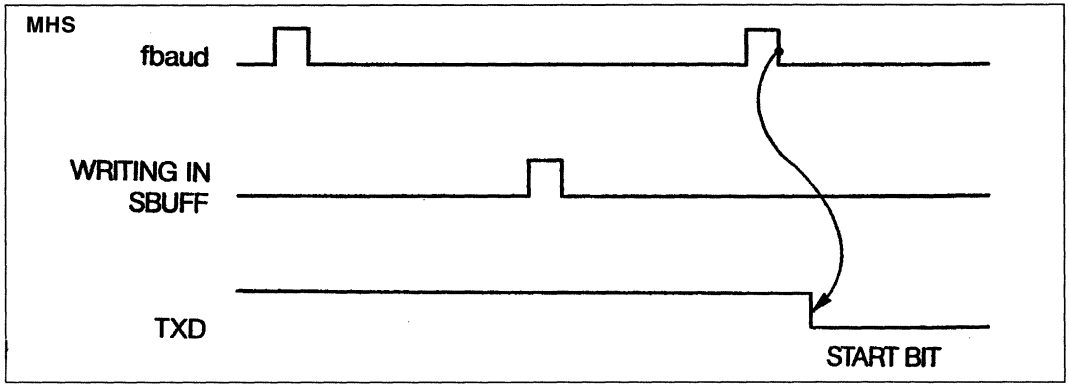
Transmission clock start-up in modes 1, 2 and 3.

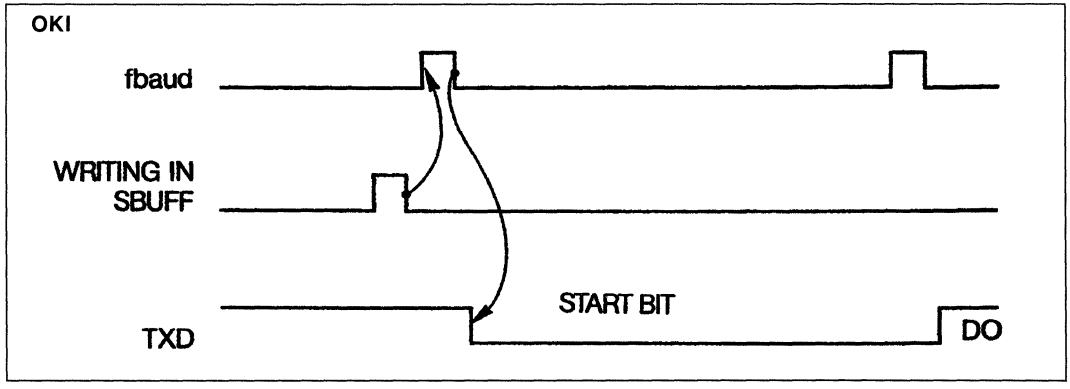


The divider by 16 which, ultimately, generates the clock, is controlled differently according to the manufacturer :

- MHS : the divider starts on completion of RESET,
- OKI : the divider starts after the following instructions :
  - MOV TCON, #XX
  - MOV SCON, #XX
  - MOV SBUF, #XX

The following timing diagrams illustrate the differences :



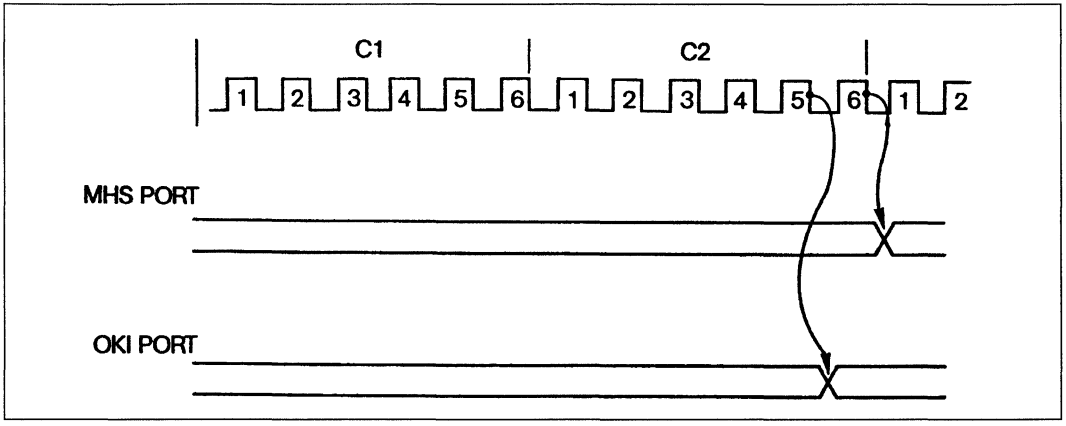


7

INPUT/OUTPUT PORTS

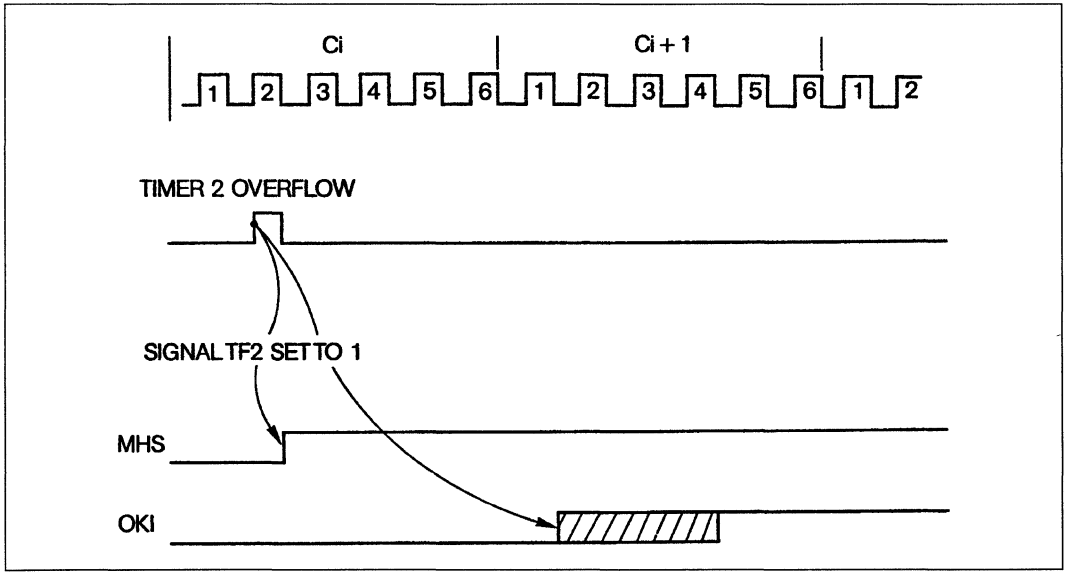
The port write instructions execute in 2 cycles. The data arrives at the gate outputs on the second instruction cycle. The rapidity with which the data arrives at the gates varies with the manufacturer :

MHS : the data arrives in phase with the 1st clock cycle of the instruction cycle following the write cycle.
OKI : the data arrive in phase with the 6th clock cycle of the 2nd write cycle.

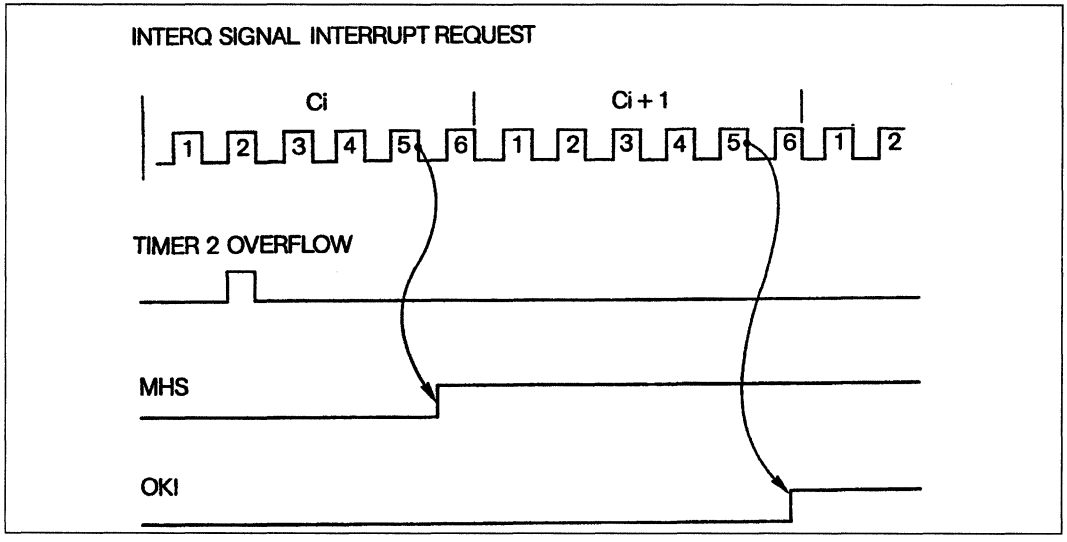


INTERRUPTS

When a TIMER (0, 1 or 2) times out, the interrupt request is generated in the same instruction cycle in the case of the MHS microcontroller and in the next cycle in the case of the OKI.



7



CONCLUSION

In pratice, all these differences are transparent from the User's point of view. Only the differences in the division by zero and the interrupt requests form the TIMERS are likely to prevent full compatibility between the two circuits.

7

PACKAGING

8

PACKAGE SELECTION GUIDE

	CERAMIC PACKAGE					PLASTIC PACKAGE		
PART NUMBER	SIDE BRAZE	CERDIP	LCC	JLCC	PGA	DIL	PLCC	QUAD FLAT
80C31/51	–	5H	L09	J44	–	X22	K04	F03
80C32/52	–	C4	EA	J44	–	X29	K03	F04
80C154/83C154	–	C4	EA	J44	–	X29	K16	F04
80C732/752	–	C4	EA	J44	–	X29	K16	–

* Contact your nearest sales office for other requirements.

STANDARD NOTES FOR PLASTIC D.I.L.

1 - Controlling dimensions : inches
In case of conflict or interpretation between the english and metric tabulation, the inch dimensions are controlling.

2 - Dimensioning and tolerancing per ansi y 14.5M-1982.

3 - Dimensions A.A1. and L are measured with the package seated in jedec seatind plane gauge GS-3.

4 - D and E1 dimensions do not include mold flash or protusions. Mold flash or protusions shall not exceed. 010 inch (0.25 mm).

5 - E and eA measured at the leads contrained to be perpendicular to the base plane.

6 - eB is measured at the lead tips with the leads uncontrained.

7 - Corner leads may be configured as shown in *figure 2.*

STANDARD NOTES FOR CERAMIC D.I.L.

1 - Controlling dimensions : inches.
In case of conflict or interpretation between the english and metric tabulation, the inch dimensions are controlling.

2 - Dimensioning and tolerancing per ansi y 14.5M-1982.

3 - Dimensions A.A1. and L are measured with the package seated in jedec seatind plane gauge GS-3.

4 - E and eA measured at the leads contrained to be perpendicular to the base plane.

5 - eB is measured at the lead tips with the leads uncontrained.

6 - Corner leads may be configured as shown in *figure 2.*

7 - Leads within 0.127 radius of true position at gauge plane with MMC (maximum material condition) and unit installed.

STANDARD NOTES FOR PLCC

1 - Controlling dimensions : inches.
In case of conflict or interpretation between the english and metric tabulation, the inch dimensions are controlling.

2 - Dimensioning and tolerancing per ansi y 14.5 M-1982.

3 - D and E1 dimensions do not include mold flash or protusions. Mold flash or protusions shall not exceed 0.25 mm (.010 inch).

STANDARD NOTES FOR PLASTIC QFP

1 - Controlling dimensions : mm.

2 - Dimensioning and tolerancing per ansi y 14.5 M-1982.

3 - D1 and E1 dimensions do not include mold flash or protusions. Mold flash or protusions shall not exceed 0.25 mm (.010 inch).

4 - Datum plane -H- located at top of mold parting line and coincident with top of lead, where lead exit plastic body.

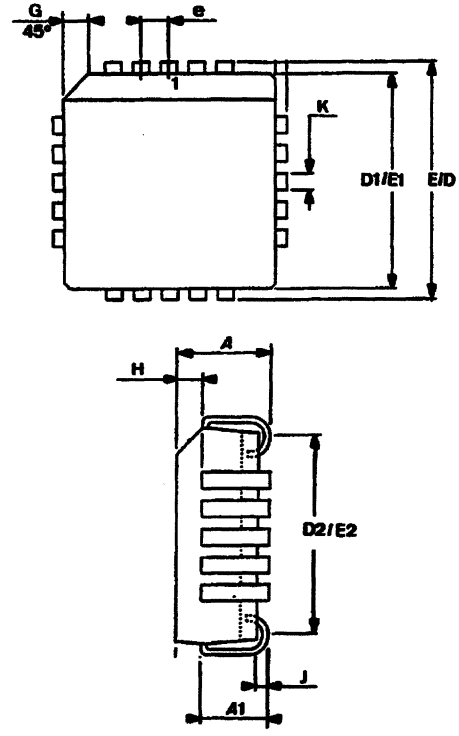
5 - Datums -A- and -D- to be determined where center of leads exit plastic body.

6 - When NBR of lead per side is even. Datums -A- and -D- are determined by adding the half pitch basic dim to the centerline of the adjacent lead.
When NBR of lead per side is odd. Datums -A- and -D- are determined by the centerline of the lead.

PLCC PACKAGE

CODES : K03 - K04 - K16 44 PINS PLCC **REV : C**

	MM dimens.		IN dimens.	
	min	max	min	max
A	4.20	4.57	.165	.180
A1	2.29	3.04	.090	.120
D	17.40	17.65	.685	.695
D1	16.51	16.66	.650	.656
D2	14.99	16.00	.590	.630
E	17.40	17.65	.685	.695
E1	16.51	16.66	.650	.656
E2	14.99	16.00	.590	.630
e	1.27 B.S.C		.050 B.S.C	
G	1.07	1.22	.042	.048
H	1.07	1.42	.042	.056
J	0.51	–	.020	–
K	0.33	0.53	.013	.021
PKG STD : 00				

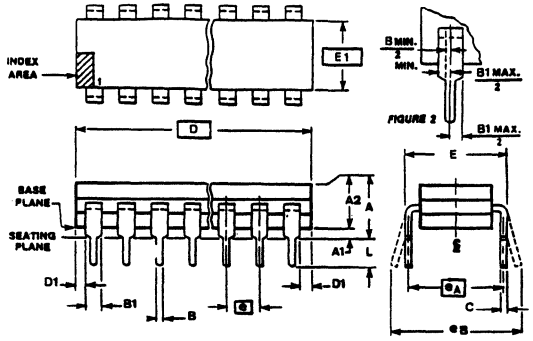


8

CERDIP DUAL IN LINE

CODES : 5H - C4 **CERDIP 40 PINS.600** **REV : D**

	MM dimens.		IN dimens.	
	min	max	min	max
A	4.07	5.58	.160	.220
A1	0.51	1.78	.020	.070
A2	–	5.08	–	.200
B	0.38	0.51	.014	.020
B1	1.15	1.39	.045	.055
C	0.20	0.38	.006	.015
D	52.00	53.87	2.047	2.070
D1	0.25	–	.010	–
E	15.41	15.62	.607	.615
E1	12.95	14.73	.510	.580
e	2.54 B.S.C		.100 B.S.C	
eA	15.24 B.S.C		.600 B.S.C	
eB	–	17.78	–	.700
L	3.17	5.06	.125	.200
PKG STD : 01				

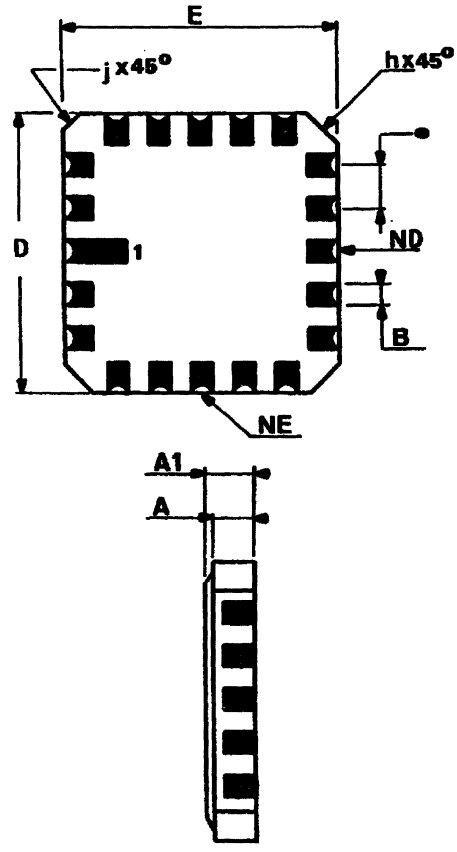


8

LEADLESS CHIP CARRIER

CODES : L09 - EA **44 LDS .050 CENTER LEADLESS SQUARE CHIP CARRIER** **REV : C**

	MM dimens.		IN dimens.	
	min	max	min	max
A	1.47	1.83	.058	.072
A1	1.73	2.26	.068	.089
B	0.635TYP		.025TYP	
D	16.33	16.82	.643	.662
E	16.33	16.82	.643	.662
e	1.27 B.S.C.		.050 B.S.C.	
h	1.016		.040	
j	0.51		.020	
ND	11		11	
NE	11		11	
PKG STD : 01				

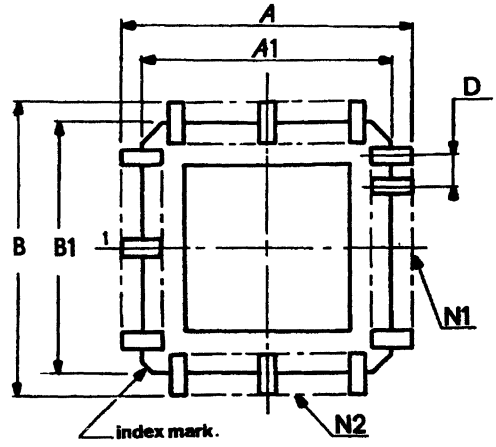


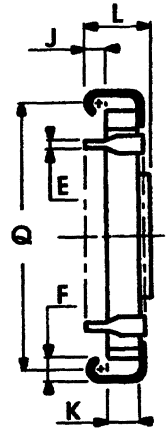
8

J LEADED CHIP CARRIER

CODE : J44 **44 LEADS JLCC. LEADED J TYPE** **REV : C**

	MM dimens.		IN dimens.	
	min	max	min	max
A-B	17.40	17.65	.685	.695
A1-B1	16.26	16.76	.640	.660
D	1.27 B.S.C		.050 B.S.C	
E	0.51 TYP		.020 TYP	
F	2.03 REF		.080 REF	
J	0.89	1.14	.035	.045
K	1.50	1.80	.059	0.71
L	2.64	3.25	.104	.128
Q	14.99	16.00	.590	.630
N1	11		11	
N2	11		11	
PKG STD : 01				



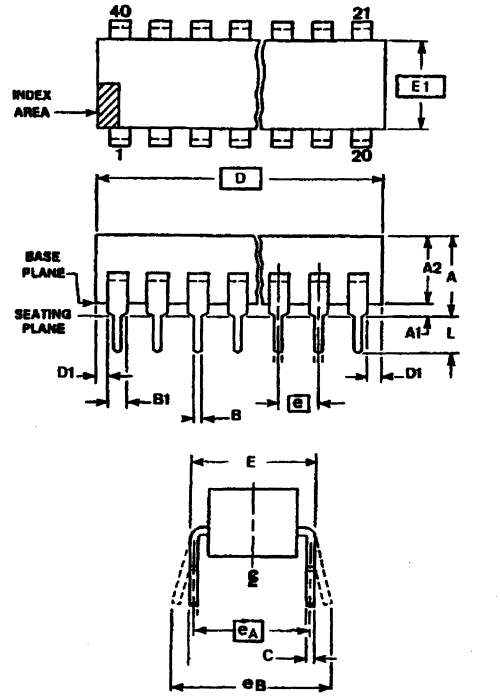


8

PLASTIC DUAL IN LINE

CODES : X22 - X29 **40 PINS PLASTIC .600** **REV : D**

	MM dimens.		IN dimens.	
	min	max	min	max
A	–	5.08	–	.200
A1	0.38	–	.015	–
A2	3.18	4.95	.125	.195
B	0.36	0.56	.014	.022
B1	0.76	1.78	.030	.070
C	0.20	0.38	.008	.015
D	50.29	53.21	1.980	2.095
E	15.24	15.87	.600	.625
E1	12.32	14.73	.485	.580
e	2.54 B.S.C.		.100 B.S.C.	
eA	15.24 B.S.C.		.600 B.S.C.	
eB	–	17.78	–	.700
L	2.93	3.81	.115	.150
D1	0.13	–	.005	–
PKG STD : 02				

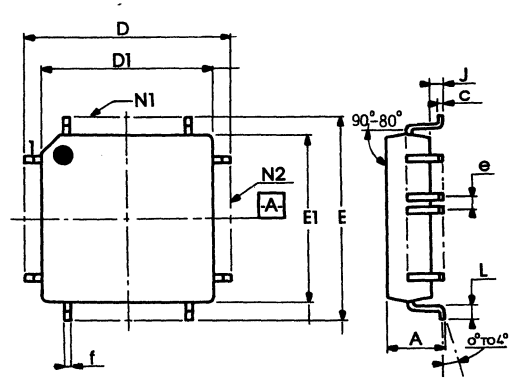


8

PLASTIC QUAD FLAT PACK

CODE : F03 - F04 **40 PINS PLASTIC QUAD FLAT PACK. SQUARE GULLWING** **REV : C**

	MM dimens.		IN dimens.	
	min	max	min	max
A	2.08	2.43	.081	.095
c	0.13	0.20	.005	.008
D1-E1	9.63	10.13	.379	.399
D-E	13.90	14.50	.547	.559
e	.80 B.S.C.		.0315 B.S.C.	
f	0.36	0.46	.014	.018
J	0.15	0.30	.006	.012
L	0.51	1.11	.020	.044
N1-N2	11		11	



8

QUALITY

9

QUALITY

1 - INTRODUCTION

1.1 - STATEMENT OF SCOPE

This section establishes the detail requirements for MATRA MHS' circuits screened and tested under the Quality Assurance Program.

Included in this section are the Quality standards and screening methods for commercial parts which must perform reliable in the field.

1.2 - APPLICABLE DOCUMENTS

The following documents form a part of this section to the extent referenced herein and provide the foundation of Matra MHS Quality Program :

MIL-M-38510G	"General Specification of Microcircuits"
MIL-STD-883C	"Test Methods and Procedures for Microelectronics"
ESA/SCC9000	"European Space Agency Specification for Microelectronics"

The MHS Quality Assurance Manual, which is available upon request, describes the total function and policies of the organization to assure product reliability and quality. All customers are encouraged to visit the MATRA MHS facilities and survey the deployment of the Quality function.

MATRA MHS maintains a Quality Assurance Program (QAP) using the above defined documents as a guide. This program assures compliance with the requirements and quality standard of control drawings and the requirements of this specification.

The special military and space programs will also be found useful by those MATRA MHS customers who must generate their own procurement specifications *(see hi-rel databook).*

Use of the enclosed MATRA MHS standard test tables, test parameters and burn-in circuits will aid in reducing specification negociation time.

1.3 - QUALITY AND RELIABILITY AT MATRA MHS

Our Quality Division strives to assure that the quality and reliability of products shipped to customers are high quality level and consistent with customer's requirements.

To achieve these requirements, MATRA MHS has started in early 1988 a Quality Improvement Program. The objective of this program is to call for continuous Progress and committment of every employee to total Quality.

The reliability approach at MATRA MHS is based on designing in reliability rather than testing for reliability only. The latter is applied to check and confirm that sound design with quality and reliability ground rules are observed and correctly executed in a new product design.

Reliability engineering becomes involved as early as concept review of a new product and continues to remain involved through design and layout reviews. At these critical development points of a new design, basic reliability layout guidelines are invoked to insure an all around reliable design. This concept is reflected by the MATRA MHS reliability procedures which encompass mandatory first run product evaluation. This is done at not only the circuit level, but also at the process and package level. Reliability engineering approval is required before new product designs are released to manufacturing.

Both maximum rated and accelerated stress conditions are performed. Acceleration is important to determine how and at what stress level a new design would fail. From this information, necessary design changes can be implemented to insure a wider and safer margin between the maximum rated stress condition and the device's stress limitation.

PPM PROGRAM

- For standard and volume products, MHS proposes to his customers a PPM program. Cooperation agreement could be established with customer willing to engage such an improvement program.
- PPM programs are already existing and we expect an optimum of 2 or 3 customers agreements by product. It is obvious that the upgrade of the quality level achieved is effective for all customers.

1.4 - AGENCIES QUALIFICATIONS

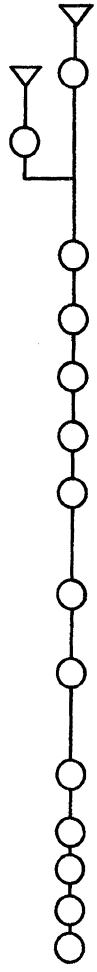
As part of specific qualification on military or space programs, MATRA MHS received several agreements from french agencies.

In 1986, MHS received the RAQ 1 certification (RAQ 1/ AQAP 1 Regulation quality assurance level 1) from the SIAR (service de surveillance industrielle de l'armement).

The RAQ1 means that MATRA MHS quality assurance procedures satisfy design, manufacturing and delivery of MOS products and on customer support on custom design.

The major MHS technologies and products are agreed by french telecommunication agency (CNET). These products are in qualified list (LNZ).

2 - QUALITY CONTROL

FLOW	PROCESS	TYPICAL ITEM	FREQUENCY	REQUIREMENTS
	Silicon wafers Incoming Inspection	• Resistivity • Bow–particles • Flatness • Taper • Oxygen content • Dimensions • Appearance	Every manufactured lot	25 Wafers/LOT
	Masks Incoming Inspection	• Defects • Dimensions • Registration • Conformity	Every mask	
	Oxidize	Thickness	Every run	3 Wafers/run 5 points/wafer
	Implant	Resistivity Thersal wave	Every run	2 Wafers/run 5 points/wafer
	Diffuse	Resistivity Thickness	Every run	3 Wafers/lot 5 points/wafer
	Silicon nitride	Thickness Critical dimensions (*)	Every run	3 Wafers/run 5 points/wafer
	Gate oxide	Thickness Defect rates VFBDVFB	Every run	3 Wafers/lot 5 points/wafer
	Polysilicon	• Resistivity • Thickness • Critical dimensions (*) • Sem Inspection	Every run Periodical	3 Wafers/lot 5 points/wafer
	Metallization	• Resistivity • Thickness • Critical dimensions (*) • Sem Inspection	Every run Periodical	1 Wafer/run
	Passivation	• Doping • CVD Thickness • Sem Inspection	Every run Periodical	3 points/wafer
	Test site	Electrical charact.	Every Wafer	5 PCM/wafer
	Blacklap	Thickness	Every lot	1 wafer/lot
	QC visual gate	Visual	Sampling	5 wafer/lot
	Wafer sort	Electrical charact.	100 % chips	

* All Critical Dimensions (After etching) : Every lot - 3 wafers/lot
- 5 points/wafer.

Table 1 : Quality flow chart of wafer processing.

2.1 - PROCESS CONTROLS

As shown by table 1 each integrated circuit is constructed by manufacturing processes which are under the surveillance of MATRA MHS Quality Control Department. The processes are monitored and controlled by use of statistical techniques and computerization in accordance with published specifications and procedures. MATRA MHS prepares and maintains suitable documentation (such as quality control manuals, inspection instructions, control charts, etc.) covering all phases of incoming part and material inspection and in-process specification. The customer may verify, with the permission of and in the company of MATRA MHS's designated representative, that suitable documentation exists and is being applied. Information designated as proprietary by MATRA MHS is made available to the customer or its representative only with the written permission of MATRA MHS.

Process control is recognized as being vital to the concept of "built-in" quality. The process control program includes a scanning electron microscope (SEM) monitor program for evaluating the metal integrity over oxide step and oxide step contour. The SEM analysis is defined in a Quality & Reliability Assurance document.

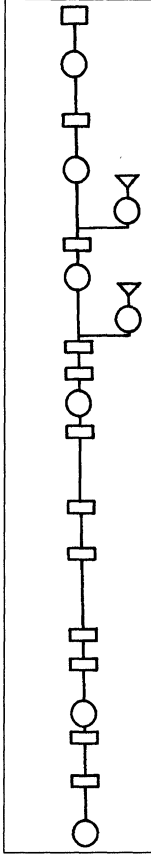
FLOW	PROCESS MATERIALS	INSPECTION	METHOD	FREQUENCY
	Scribing	Visual	2010 Cond B	Monitor
	2 Nd Optical	Visual	2010 Cond B	100 %
	Scribing	Visual	2010 Cond B	100 %
	QC Inspection	Visual	2010 Cond B	Every lot
	Lead Frame, base (incoming inspection)			
	Die Bonding			
	QC Inspection	Appearance		Every lot
	Wire (incoming inspection)			
	Wire Bonding	Bond sirength	2011	Every lot
	Praseal inspection	Visual		100 %
	QC Inspection	Visual		Every lot
	Prestabilization sealing			
	Temperature Cycling		1010 Condition C	100 %
	Centrifuge		2001 Condition E	100 %
	Lead cut			
	Plating			
	Plating inspection	Appearance thickness		Every lot
	Marking	Permanency	2015	Every lot
	Fine leak		1014 Condition A or B	100 %
	Gross leak		1014 Condition C	100 %
	QA monitoring		All QC inspection	

Table 2 : QC Ceramic flow charts of assembly process (1).

2.1 - CONTROL OF PROCUREMENT SOURCES

MATRA MHS is responsible for assuring that all supplies and services conform to this specification, the detail specification and MHS's procurement requirements.

A - MATRA MHS/supplier convention

Prior to use in production, MATRA MHS verifies the capability of the supplier QA, manufacturing engineering and services to deliver material conform to specification and kept under control. Formal agreement is established between the two partners.

B - Receiving inspection

Purchased supplies are subjected to inspection after receipt as necessary to ensure conformance to contract requirements. In selecting sampling plans, consideration is given to the controls exercised by the procurement source and evidence of substained quality conformance.

C - MATRA MHS initiates corrective action with the procurement source depending upon the nature and frequency of receipt of nonconforming supplies.

COMMERCIAL TEMP. RANGE 0°C to 70°C			INDUSTRIAL TEMP. RANGE – 40°C to + 85°C			AUTOMOTIVE – 40°C to 110°C	MILITARY TEMP. RANGE – 55°C to + 125°C		
Family	STD	STD + B.I.	**Family**	STD	STD + B.I.	**Family**	**Family**	STD	STD + B.I.
Memory suffix	– 5	– 5 +	**Memory suffix**	– 9	– 9 +		**Memory**	– 2	– 8
Micro's prefix		Q	**Micro's prefix**	I	L	**Micro's prefix A**	**Micro's**		M---/B
Gate array suffix	– 5	– 5 +	**Gate array suffix**	– 9	– 9 +		**Gate array**	– 2	– 8

Table 1 : PROCESS FLOWS INFORMATION

	COMMERCIAL		INDUSTRIAL		AUTOMOTIVE	MILITARY	
	STD	STD + B.I.	STD	STD + B.I.		STD	STD + B.I.
QA Wafer visual inspection	Monitor	Monitor	Monitor	Monitor	Monitor	Monitor	Monitor
Electrical test and probe 25°C	100%	100%	100%	100%	100%	100%	100%
Assembly (see table 2 and 3)	100%	100%	100%	100%	100%	100%	100%
Pre Burn–in test		100%		100%	100%		100%
Burn–in		100% (1)		100% (1)	100% (1)		100% (1)
Post Burn–in test		100%		100%	100%		100%
P.D.A (percentage defective allowable)		5% (2)		5% (2)	5% (2)		5% (2)
Final electrical test (per MHS specification)	100% high temp. Low temp. optional	idem	idem	idem	idem	idem	idem
Marking (lot number + branding week code, per MATRA MHS specification)	100%	100%	100%	100%	100%		100%
Lead Straigthen	100%	100%	100%	100%	100%	100%	100%
MATRA MHS Quality final acceptance electrical	(3)	(3)	(3)	(3)	(3)	(3)	(3)
mechanical (visual)	100%	100%	100%	100%	100%	100%	100%

Notes : 1) Burn-in is performed as 24 h, 125°C (or equivalent) minimum.
2) If a lot fails the 5 % PDA, but is < 10 %, the lot may be submitted to burn-in on time only to the same time and temperature condition.
3) MHS quality final acceptance is performed following quality dispositions to assure 200 ppm. Average Outgoing Quality.

3 - RELIABILITY RESULTS

The objective failure rate at 55°C, 60 % UCL is in any case lower than 100 fits. Details about data base are available upon request in Reliability Reports, written for each product or product family.

4 - MILITARY HI-REL PRODUCTS

A broad choice of quality grades is available.

CB : CECC program according to level B of CECC 90000.

MB : military program according to class B of MIL-STD 883C.

SB : space program according to level B of SCC 9000 (LAT 1, 2, 3).

SC : space program according to level C or SCC9000 (LAT 1, 2, 3).

DB : dice military program with qualification flow.

(refer to MATRA MHS high reliability Data-book)

5 - DICE/WAFER FORM

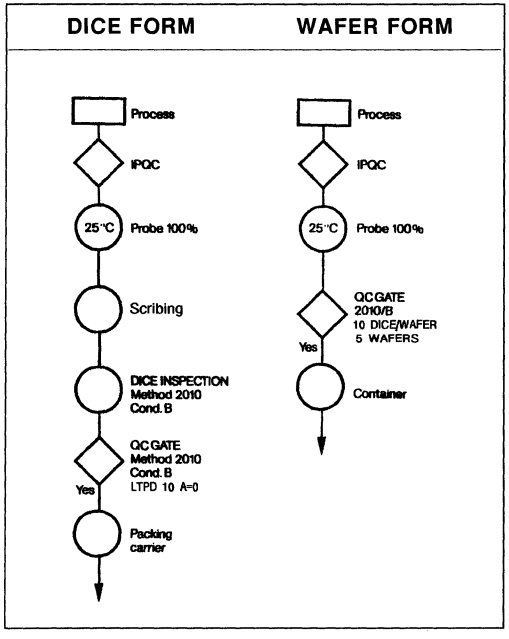
MATRA MHS Memory, Gate Arrays and Microcontroller products are available in chip form and wafer form to the hybrid microcircuit designer. Table 1 gives the different flows used for Military and Standard levels with the production operations and QC gates.

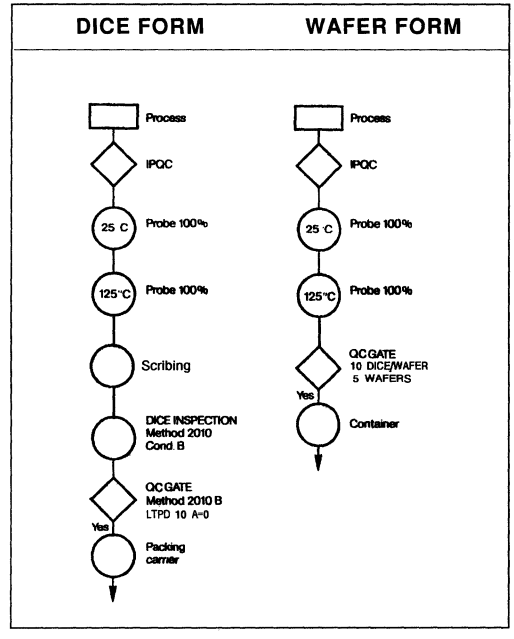
So as to respond to specific requirements of the hybrid industry, MATRA MHS has several additionnal options to table 1 as electrical qualification lots for military dice, available upon request at extra cost.

PROCESS FLOWS INFORMATION (Table 1)

STANDARD FLOW		
FAMILY	**DICE FORM**	**WAFER FORM**
MEMORY	HM0–65162-6	HMW–65162-6
MICRO'S	XX–80C31	XW–80C31
GATE ARRAY	MA0–250A69-6	MAW–250A69-6

MILITARY FLOW		
FAMILY	**DICE FORM**	**WAFER FORM**
MEMORY	HM0–65162-2	HMW–65162-2
MICRO'S	–80C31	XW–80C31
GATE ARRAY	MA0–250A69-2	MAW–250A69-2





9

MHS LOCATIONS

10

M.H.S.
LOCATIONS

MHS ELECTRONIC CENTER

La Chantrerie/Route de Gachet
CP 3008
44087 Nantes Cedex 03/France
Tel. : (33) 40303030 - Twx : 711930 - Fax : (33) 40300216

SALES OFFICES

FRANCE

"Les Quadrants"
3, avenue du Centre
BP 309
78054 St-Quentin-Yvelines Cedex
Tel. : (33) 1-30607000
Twx : 697317
Fax : (33) 1-30640693

GERMANY

Erfurterstrasse 29
D-8057 Eching
Tel. : (49) 89-31900550
Twx : 524126
Fax : (49) 89-31900555

ITALY

Via Vigliani 13
I-20148 Milano
Tel. : (39) 2-4984586
(39) 2-462602
Twx : 334595
Fax : (39) 2-4818660

SCANDINAVIA

MD Semiconductor AB
Dragon Plan 1, Rissne
Box 2042 - 17202 Sundbyberg
Sweden
Tel. : (46) 87.33.00.90
Fax : (46) 87.33.05.58

UNITED KINGDOM

Easthampstead Road
Bracknell
Berkshire RG12 1LX
Tel. : (0344) 485757
Intl : (44) 344-485757
Twx : 849392
Fax : (44) 344-427371

FAR EAST

MATRA DESIGN SEMICONDUCTOR
ASIA
Room 1601, 16/FL, Shui on Centre
6-8 Harbour Road
Wanchai
Hong Kong
Tel. : (852)-5-8650861
Fax : (852)-5-8651273
Tlx : 85351 HX

USA

MATRA DESIGN SEMICONDUCTOR
2895 Northwestern Parkway
Santa Clara, CA 95051
Tel. : (1) 408/986-9000
Twx : 299656
Fax : (1) 408/748-1038

EUROPEAN DISTRIBUTORS

AUSTRIA

TRANSISTOR
VERTRIEBSGESELLSCHAFT
mbH CO-KG
Auhofstrasse 41A
A-1130 Wien
Tel. : (43) 222-829401
Twx : 133738
Fax : (43) 222-826440

BELGIUM

MICROTRON NV S.A.
Generaal Dewittelaan 7
2800 Mechelen
Tel. : (32) 15-212223
Twx : 22606
Fax : (32) 15-210069

DENMARK

DITZ SCHWEITZER A.S.
Vallensbakvej 41
Post Box 5
DK-2600 Glostrup
Tel. : (45) 2-453044
Twx : 33257
Fax : (45) 2-459206

FINLAND

YLEISELEKTRONIKKA OY
P.O. Box 73
SF-02201 Espoo
Tel. : (358) 0-4521255
Twx : 123212
Fax : (358) 0-428932

FRANCE

A2M
6, av. Charles-de-Gaulle
78150 Le Chesnay
Tel. : (33) 1-39549113
Twx : 698376
Fax : (33) 1-39543061

ALMEX
48, rue de l'Aubépine
92160 Antony
Tel. : (33) 1-46662112
Twx : 250067
Fax : (33) 1-46666028

ELTEK (Wafer dice only)
BP 1077
Z.A. La Tuilerie
78204 Mantes-la-Jolie Cedex
Tel. : (33) 1-34771616
Twx : 699717

EUROPEAN DISTRIBUTORS (continued)

FRANCE (continued)

FEUTRIER
5, rue Jean-Zay
42271 St-Priest-en-Jarez
Tel. : (33) 77934040
Twx : 300021
Fax : (33) 77932631

RTF
81, rue Pierre Semard
92320 Chatillon
Tel. : (33) 1-49652700
Twx : 632247
Fax : (33) 1-49652738

ERICSSON COMPOSANTS
1, Parc Club Ariane
Rue Hélène Boucher
78284 Guyancourt
Tel. : (33) 1-30640900
Twx : 697347
Fax : (33) 1-30641146

GERMANY

ALFRED NEYE ENATECHNIK GMBH
Schillerstrasse 14
D-2085 Quickborn
Tel. : (49) 4106-6121
Twx : 213590
Fax : (49) 4106-612268

JERMYN GMBH
Im Dachsstück 9
D-6250 Limburg
Tel. : (49) 6431-5080
Twx : 4152570
Fax : (49) 6431-508289

RUTRONIK-RSC-Halbleiter und
Elektronische Bauelemente GMBH
Industriestrasse 2
D-7536 Ispringen
Tel. : (49) 7231-8010
Twx : 783650
Fax : (49) 7231-82282

SPOERLE ELECTRONICS GMBH
Max-Plank-Strasse 1-3
Postfach 102140
D-6072 Dreleich 1
Tel. : (49) 6103-3040
Twx : 417972
Fax : (49) 6103-304201

ITALY

CAMEL ELETTRONICA
Via Tiziano 18
I-20145 Milano
Tel. : (39) 2-4981481
Twx : 325237
Fax : (39) 2-4818637

KONTRON SPA
Via Fantoli 16/15
I-20138 Milano
Tel. : (39) 2-50721
Twx : 312288
Fax : (39) 2-5060918

LASI ELETTRONICA SPA
Viale Fulvio Testi 126
I-20092 Cinisello Balsamo
Tel. : (39) 2-2440012
Twx : 352040
Fax : (39) 2-2487717

NETHERLANDS

NIJKERK
Drentestraat 7
1083 HK AMSTERDAM
Postbus 7920
1008 AC AMSTERDAM
Tel. : (31) 20-5495-969
Twx : 11625 NESCO
Fax : (31) 20-423-948

NORWAY

TAHONIC A/S
P.O. Box 140/KALBAKKEN
N-0902 OSLO 9
Tel. : (47) 2-161610
Twx : 77397
Fax : (47) 2-257317

SPAIN

SELCO SA
Paseo de la Habana 190
28036 Madrid
Tel. : (34) 1-4054213
Twx : 45458
Fax : (34) 1-2592284

SWITZERLAND

ELECTRONITEL
Chemin du Grand Clos 1
B.P. 93
CH-1752 VILLARS-SUR-GLANE
Tel. : (37) 41.00.60
Fax : (37) 41.00.70

UNITED KINGDOM AND IRELAND

MACRO MARKETING LTD
Burnham Lane - SLOUGH
Berkshire SL1 6LN.
Tel. : (06286) 4422
Intl. : (44) 6286 4422
Twx : 847945
Fax : (06286) 66873

MTL MICROTECHNOLOGY LIMITED
(Die Distributor)
Test House - Mill Lane - ALTON
Hampshire. GU34 2QG.
Tel. : (0420) 88022
Intl. : 44 420 88022
Tlx : 858456
Fax : (0420) 87259

POLAR ELECTRONICS LIMITED
Cherrycourt Way
Leighton Buzzard
Bedfordshire. LU7 8YY.
Tel. : (0525) 377093
Intl. : 44 525 377093
Tlx : 825238
Fax : (0525) 378367

R R ELECTRONICS LIMITED
St. Martin's Way Industrial Estate
Cambridge Road - BEDFORD
Bedfordshire. MK42 0LF
Tel. : (0234) 47188
Intl. : 44 234 47188
Tlx : 826251
Fax : (0234) 210674

THAME COMPONENTS LIMITED
Thame Park Road - THAME
Oxfordshire OX9 3XD.
Tel. : (084426) 1188
Intl. : 44 84426 1188
Twx : 837917
Fax : (084426) 1681

SWEDEN

TH:S ELEKTRONIK
P.O. Box 3027
16303 Spanga - Sweden
Tel. : (46) 8.36 29 70
Fax : (46) 8.761.30.65

FAR EAST REPRESENTATIVES AND DISTRIBUTORS

INDIA

SPARTEX SYSTEMS & SERVICES
Pvt Ltd.,
N° 68, Michael Palyam,
Near Deccan Studio,
C.V. RAMAN NAGAR POST,
BANGALORE - 560 093,
INDIA
Tel. : 564211, 568772
Tlx : 0845 - 2190 MLHR IN

JAPAN

CHRONIX INCORPORATED
Maruyama Bld 4F
7-9-7, Nishishinjuku
Shinjuku-ku, Tokyo, 160
Tel. : (81) 3-371-5711
Twx : 26244
Fax : (81) 3-371-5738

KOREA

BUK SUNG INDUSTRIAL CO LTD
3/FL, Samyuna Bida, 159-22
Docksan-4 Dong, Guro-Ku
Seoul
Tel. : (82) 2.854.1362
Twx : BUSUCOK26925
Fax : (82) 2.8621273

10

FAR EAST REPRESENTATIVES AND DISTRIBUTORS (continued)

SINGAPORE

WESTECH ELECTRONICS PTE LTD.
3, Lorong Bakar Balu # 05-02
Brightway Building
Singapore 1334
Tel. : (65)-7436355
Twx : 55070
Fax : (65)-7461396

SCAN TECHNOLOGY(s) PTE LTD
So Kallana Bahru # 04-01/03
Kallana Basin Ind
Singapore 1233
Tel. : (65)-2942112
Twx : RS74983 STECH
Fax : (65) 2961685

AUSTRALIA

CONSULAUST INTERNATIONAL PTY, LTD.
1, Norfolk Road - Surrey Hills,
PO Box 357, Camberwell
Victoria, 3124
Tel. : (61) 3-8362566
Twx : CONAUS AA37455
Fax : 61-3-8301764

TAIWAN

UNION TECHNOLOGY CORPORATION
3/FL, 585 Ming Sheng East Road
Taipei, Taiwan
Tel. : (886) 2-505.8616
Twx : 202 61 MECTAL
Fax : (886) 2-5056609

WORLD PEACE INDUSTRIAL CO LTD
5/fL, 309 Sung Chiang Road
Taipei, Taiwan
Tel. : (886) 2-5056621/5056345
Fax : (886) 2-5058760

HONG KONG

PROTECH COMPONENTS LTD
Flat 3, 10/F, Wing Shing Ind. Bldg
26 Ng Fong Street
San Po Kong, Kowloon
Tel. : (852) 3-3522181
Fax : (852) 3-3523759

WILLAS COMPANY LTD
8/F, Wing Tai Centre
12 Hing Yip Street
Kwun Tong, Kowloon
Tel. : (852) 3-414281, 3-890343
Twx : 39315 WILAS HX
Fax : (852) 3-431229

US REPRESENTATIVES

ANCHOR ENGR.

11 WALKUP DR,
WESTBORO
MASSACHUSSETTS 01581-1018
Tel. : 508-898-2724
Fax : 508-870-0573

ARBOTEK

10404 W. JOPPA RD.
TOWSON
MARYLAND 21204
Tel. : 301-825-0775
Fax : 301-337-2781

2201 ANGUS RD., SUITE 14
CHARLOTTESVILLE
VIRGINIA 22901
Tel. : 804-971-5736

C.C. ELECTRO

5335 N. TACOMA AVE., SUITE #1
INDIANAPOLIS
INDIANA 48220
Tel. : 317-255-1508
Fax : 317-266-6875

5635 FORDHAM CIRCLE #203
CANTON
MICHIGAN 48187
Tel. : 313-981-9298

9735 RAVENNA RD.
TWINSBURG
OHIO 44087
Tel. : 216-425-8338
Fax : 216-425-2147

CAHILL, SCHMITZ & HOWE

4905 Lakeside Drive
N.E. SUITE 100
CEDAR RAPIDS
IOWA 52402
Tel. : 319-377-8219
Fax : 319-377-0958

E.M.A.

6695 PEACHTREE IND. BLVD.,
SUITE 101
ATLANTA
GEORGIA 30360
Tel. : 404-448-1215
Fax : 404-446-9363

210 W. STONE AVE.
GREENVILLE
S. CAROLINA 29609
Tel. : 803-233-4637
Fax : 803-242-3089

7501 South Memorial Parkway, #202
HUNTSVILLE
ALABAMA 35802
Tel. : 205-880-8050
Fax : 205-880-8054

8512 SIX FORKS RD., SUITE 601 A
RALEIGH
N. CAROLINA 27615
Tel. : 919-847-8800
Fax : 919-848-1787

ELECTEC

3211 SCOTT BLVD., SUITE 101
SANTA CLARA
CALIFORNIA 95054
Tel. : 408-496-0706
Fax : 408-727-9817

8465 ROYAL OAKS DRIVE
GRANITE BAY
CALIFORNIA 95661
Tel. : 916-797-0414
Fax : 916-456-6001

GEORGE RUSSEL & ASSOC.

8030 CEDAR AVE. SOUTH,
SUITE 114
MINNEAPOLIS
MINNESOTA 55420
Tel. : 612-854-1168

LANDA & ASSOC.

1518 COTNER AVE.
LOS ANGELES
CALIFORNIA 90025
Tel. : 213-879-0770
Fax. : 213-478-0190

1616 E. 4TH ST.
SANTA ANA
CALIFORNIA 92701
Tel. : 714-543-7805
Tax : 714-543-1380

662 NARDO AVE.
SOLANA BEACH
CALIFORNIA 92075

US REPRESENTATIVES (continued)

M.E.C.
700 W. HILLSBORO BLVD., BLDG. 4, #204
DEERFIELD BEACH
FLORIDA 33441
Tel. : 305-426-8944
Fax : 305-426-8799

830 N. ATLANTIC BLVD., SUITE B401
COCOA BEACH
FLORIDA 32931
Tel. : 407-799-0520
Fax : 407-799-0923

511 CARRIAGE ROAD
INDIAN HARBOUR BEACH
FLORIDA 32937
Tel. : 407-332-7158
Fax : 407-830-5436

1001 45TH AVE., NE
ST. PETERSBURG
FLORIDA 33703
Tel. : 813-522-3433
Fax : 813-522-3933

N.E. COMPONENTS
155 GRANDVIEW LANE
MAHWAH
NEW JERSEY 07430
Tel. : 201-825-0233
Fax : 201-934-1310

22 LAWRENCE AVE,
SMITHTOWN
NEW YORK 11787
Tel. : 516-724-3485

PHOENIX SALES
257 MAIN ST.
TORRINGTON
CONNECTICUT 06790
Tel. : 203-496-7709
Fax : 203-496-0912

SW MARKETING ASSOC.
10940 ALDER CIRCLE
DALLAS
TEXAS 75238
Tel. : 214-341-8631
Fax : 214-340-5870

13006 KELLIES FARMLANE
AUSTIN
TEXAS 78727
Tel. : 512-255-8010

400 FM 1960 WEST, SUITE 100-15
HOUSTON
TEXAS 77090
Tel. : 713-537-8166
Fax : 713-537-9738

6713 E. 54TH ST.
TULSA
OKLAHOMA 74145
Tel. : 918-663-7536

SYNERGISTIC SALES
501 MITCHELL ROAD
GLENDALE HEIGHTS
ILLINOIS 60139
Tel. : 312-858-8686
Fax : 312-790-9799

TECH SALES ASSOC.
EXEC. MEWS, RJ-52, 2300
COMPUTER AVE.
WILLOW GROVE
PENNSYLVANIA 19090
Tel. : 215-784-0170
Fax : 215-784-9201

THORSON CO. n.w.
12340 N.E. 8TH ST., SUITE 201
BELLEVUE
WASHINGTON 98005
Tel. : 206-455-9180
Fax : 206-455-9185

8700 S.W. 105TH AVE.
BEAVERTON
OREGON 97005
Tel. : 503-644-5900
fax : 503-644-5919

TRUE NORTH TECH. LTD.
100 WESTMORE DR., SUITE 12E
REXDALE, ONTARIO
CANADA M0V 5C3
Tel. : 416-744-2233
Fax : 416-744-3376

1883 LONGMAN CRESCENT
GLOUCESTER, ONTARIO
CANADA K1C 5G7
Tel. : 613 824-8957
Fax : 614-745-0315

WEST. INC.
1740 PLATTE SR. #200
DENVER
COLORADO 80202
Tel. : 303-477-1134

460 EAST 100 SOUTH
CENTERVILLE
UTAH 84014-5087
Tel. : 801-292-8787
Fax : 801-298-0788

US DISTRIBUTORS

ADDED VALUE ELECTRONIC DISTRIBUTION, INC.

1512 PARKWAY LOOP, UNIT G
TUSTIN, CA 92680
Tel. : (714) 259-8258
Fax : (714) 259-0828

31194 LA BAYA DRIVE, SUITE 100
WESTLAKE VILLAGE, CA 91362
Tel : (805) 643-2101
(818) 889-2861
Fax : (818) 889-2472

7741 E. GRAY ROAD, SUITE 9
SCOTTSDALE, AZ 85260
Tel. : (602) 951-9788
Fax : (602) 951-4182

A.V.E.D. - ROCKY MOUNTAIN, INC.
1836 PARKWAY BLVD.
WEST VALLEY CITY, UT 84119
Tel. : (801) 975-9500
Fax : (801) 977-0245

A.V.E.D. - ROCKY MOUNTAIN, INC.
4090 YOUNGFIELD ST.
WHEAT RIDGE, CO 80033
Tel. : (303) 422-1701
Fax : (303) 422-2529

A.V.E.D. - SOUTHWEST INC.
4470 SPRING VALLEY ROAD
DALLAS, TX 75244
Tel. : (214) 404-1144
Fax : (214) 233-2614

ALL AMERICAN SEMICONDUCTOR CORP.

2360 QUME DRIVE, SUITE C
SAN JOSE, CA 95131
Tel. : (408) 943-1200
Fax : (408) 943-1393

16251 N.W. 54TH. AVENUE
MIAMI, FL 33014
Tel. : (305) 621-8282
(800) 228-7459
Fax : (305) 620-7831

369 VAN NESS WAY, SUITE 701
TORRANCE, CA 90501
Tel. : (213) 320-0240
(800) 669-8300
Fax : (213) 320-7207

US DISTRIBUTORS (continued)

ALL AMERICAN SEMICONDUCTOR CORP. (continued)

5009 HIATUS ROAD
SUNRISE, FL

107 AUDUBON ROAD, SUITE 104
WAFEFIELD, MA 01880
Tel. : (617) 246-2300

1819 FIRMAN DRIVE, #127
RICHARDSON, TX 75081
Tel. : (214) 231-5300
Fax : (214) 437-0353

14636 ROTHGEB DRIVE
ROCHVILLE, MD 20850
Tel. : (301) 251-1205
Fax : (301) 251-8574

711-2 KOEHLER AVENUE
RONKONKOMA, NY 11779
Tel. : (516) 981-3935
(800) 874-2830
Fax : (516) 931-3947

11409 VALLEY VIEW ROAD
EDEN PRAIRIE, MN 55344
Tel. : (800) 342-7364
Fax : (612) 944-9803

BELL INDUSTRIES

1031 PUTNAM DRIVE, SUITE A
HUNTSVILLE, AL 35816
Tel. : (205) 837-1074
Fax : (205) 830-5598

306 E. ALONDRA BLVD.
GARDENA, CA 90247
Tel. : (213) 515-1800
Fax : (213) 777-3111 X306

11812 SAN VINCENTE BLVD., #300
LOS ANGELES, CA 90049
Tel. : (213) 826-6778
Fax : (213) 258-6932

1705 W. 4TH ST.
TEMPE, AZ 85281
Tel. : (602) 966-7800
Fax : (602) 967-6584

11095 KNOTT AVE., SUITE E
CYPRESS, CA 90630
Tel. : (714) 891-4570

4311 ANTHONY COURT, #100
ROCKLIN, CA 96677
Tel. : (916) 652-0414
Fax : (916) 652-0403

7450 RONSON ROAD
SAN DIEGO, CA 92111
Tel. : (619) 268-1277
Fax : (619) 268-3733

638 SO, MILITARY TRAIL
DEERFIELD BEACH, FL 33442
Tel. : (305) 421-1997
Fax : (305) 421-5705

3020 A BUSINESS PARK DRIVE
NORCROSS, GA 30071
Tel. : (404) 662-0923
Fax : (404) 449-6901

130 KILLARNEY
URBANA, IL 61801
Tel. : (217) 328-1077
Fax : (217) 328-1148

5230 WEST 79TH ST.
INDIANAPOLIS, IN 46268
Tel. : (317) 875-8200
Fax : (317) 875-8219

1161 NO. FAIROAKS AVE.
SUNNYVALE, CA 94089
Tel. : (408) 734-8570
Fax : (408) 734-8875

12421 W. 49TH AVENUE
WHEATH RIDGE, CO 80033
Tel. : (303) 424-1985
fax : (303) 424-0932

10810 72ND ST. NORTH, SUITE 201
LARGO, FL 33541
Tel. : (813) 541-4434
Fax : (813) 546-6418

515 BUSSE AVENUE, UNIT D-I
ELK GROVE VILLAGE, IL 60007
Tel. : (312) 640-1910
Fax : (312) 640-0474

3433 E. WASHINGTON BLVD.
FT. WAYNE, IN 46803
Tel. : (21ç) 423-3422
Fax : (219) 424-2433

1221 PARK PLACE, N.E.
CEDAR RAPIDS, IA 52402
Tel. : (319) 395-0730
Fax : (319) 395-9761

100 BURTT ROAD? #106
ANDOVER, MA 01810
Tel. : (508) 474-8880
Fax : (508) 474-8902

814 PHOENIX DRIVE
ANN ARBOR, MI 48404
Tel. : (313) 971-9093
Fax : (313) 971-9178

444 WINDSOR PARK DRIVE
DAYTON, OH 45459
Tel. : (513) 435-8660
Fax : (513) 435-6765

6024 SOUTHWEST JEAN ROAD
LAKE OSWEGO, OR 97034
Tel. : (503) 241-4115
Fax : (503) 635-6500

6912 SOUTH 185 WEST, SUITE B
MIDVALE, UT 84044
Tel. : (801) 255-9611
Fax : (801) 255-2477

W. 227N, 913 WESTMOUND DRIVE
WAUKESHA, WI 53186
Tel. : (414) 547-8879
Fax : (414) 547-6547

6979 WASHINGTON AVE. SO.,
SUITE 200
EDINA, MN 55435
Tel. : (612) 941-1493
Fax : (612) 941-2964

11728 LINN, N.E.
ALBUQUERQUE, NM 87123
Tel. : (505) 292-2700
Fax : (505) 275-2819

118 WESTPARK ROAD
DAYTON, OH 45459
Tel. : (513) 434-8231
Fax : (513) 434-8103

1701 GREENVILLE, #306
RICHARDSON, TX 75081
Tel. : (214) 690-0466
Fax : (214) 690-0822

8553 154TH AVE. N.E.
REDMOND, WA 98052
Tel. : (206) 885-9963
Fax : (206) 867-5159

30101 AGOURA COURT, SUITE 118
AGOURA HILLS, CA 91301
Tel. : (818) 706-2608
Fax : (818) 891-7695

10

MHS

US DISTRIBUTORS (continued)

CAM RPC ELECTRONICS

2975 BRIGHTON HENRIETTA
TOWN LINE RD.
ROCHESTER, NY 14623
Tel. : (716) 427-9999
Fax : (716) 427-7559

7973-B WASHINGTON WOOD DRIVE
DAYTON, OH 45459
Tel. : (513) 433-5551
Fax : (513) 461-4329

749 MINER ROAD
CLEVELAND, OH 44143
Tel. : (216) 461-4700
Fax : (216) 461-4329

620 ALPHA DRIVE
PITTSBURGH, PA 15238
Tel. : (412) 963-6202
Fax : (412) 963-6210

FALCON ELECTRONICS INC.

5 HIGGINS DRIVE
MILFORD, CT 06460
Tel. : (203) 878-5272
Fax : (203) 877-2010

CATON RESEARCH CENTER,
SUITE Q
1520 CATON CENTER DRIVE
BALTIMORE, MD 21227
Tel. : (301) 247-5800
Fax : (301) 247-5893

1383 VETERAN'S MEMORIAL HWY.
HAUPPAGE, NY 11788
Tel. : (516) 724)-0980
(800) 528-0016
Fax : (516) 724-0993

HAMMOND ELECTRONICS, INC.

4411-B EVANGEL CIRCLE N.W.
HUNTSVILLE, AL 35816
Tel. : (205) 830-4764
Fax : (205) 830-4287

6600 N.W. 21ST AVE., 8AY D
FT. LAUDERDALE, FL 33309
Tel. : (305) 973-7103
(FT LAUDERDALE)
Fax : (305) 973-7601

2923 PACIFIC AVE.
GREENBORO, NC 27420
Tel. : (919) 275-6391

1230 WEST CENTRAL BLVD.
ORLANDO, FL 32802
Tel. : (407) 841-1010 (ORLANDO)
Fax : (407) 648-8584

5680 OAKBROOK PKWY., SUITE 160
NORCROSS, GA 30093
Tel. : (404) 449-1996
(800) 241-5437
Fax : (404) 424-9834

NU HORIZONS ELECTRONICS CORP.

151 ANDOVER ST.
DANVERS, MA 01923
Tel. : (617) 777-8800
Fax : (617) 777-8806

6000 NEW HORIZONS BLVD.
AMITYVILLE, NY 11701
Tel. : (516) 226-6000
Fax : (516) 226-6262

39 U.S. ROUTE 46
PINE BROCK, NJ 07058
Tel. : (201) 882-8300
Fax : (201) 882-8398

2002C GREENTREE EXECUTIVE
CAMPUS
MARLTON, NJ 08053
Tel. : (609) 596-1833
Fax : (609) 596-0612

100 BLUFF DRIVE
EACH ROCHESTER, NY 1445
Tel. : (716) 248-5980
(203) 265-0162
Fax : (716) 248-9132

SEMAD/DGW ELECTRONICS CORP.

85 SPY COURT
MARKHAM, ONTARIO
CANADA L3R 424
Tel. : (416) 475-3922
Fax : (416) 475-4158

1827 WOODWARD DRIVE,
SUITE 303
OTTAWA, ONTARIO
CANADA K2C OR3
Tel. : (613) 727-8325
Fax : (613) 727-9489

8563 GOVERNMENT ST.
BURNABY, B.C.
CANADA V3N 4S9
Tel. : (604 420-9889
Fax : (604) 420-0124

243 PLACE FRONTENAC
POINTE CLAIRE, PQ
CANADA H9R 4Z7
Tel. : (514) 694-0860
Fax : (514) 694-0965

6120 THIRD ST., SE UNIT #G
CALGARY, ALBERTA
CANADA T2H 1K4
Tel. : (403) 252-5664
Fax : (403) 255-0966

10

MATRA MHS