DESCRIPTION

The Signetics 8X300 Microcontroller is a monolithic, high-speed microprocessor implemented with bipolar Schottky technology. As the central processing unit, CPU, it allows 16-bit instructions to be fetched, decoded and executed in 250ns. A 250ns instruction cycle requires maximum memory access of 65ns, and maximum I/O device access of 35ns.

Microcontroller instructions operate on 8-bit, parallel data. Logic is distributed along the data path within the Microcontroller. Input data can be rotated and masked before being subject to an arithmetic or logical operation; and output data can be shifted and merged with the input data, before being output to external logic. This allows 1- to 8-bit I/O and data memory fields to be accessed and processed in a single instruction cycle.

PROGRAM STORAGE INTERFACE

Program Storage is typically connected to the A0-A12 (A12 is least significant bit) and I0-I15 signal lines. An address output on A0-A12 identifies one 16-bit instruction word in program storage. The instruction word is subsequently input on I0-I15 and defines the Microcontroller operations which are to follow.

The Signetics 82S115 PROM, or any TTL compatible memory, may be used for program storage.

I/O DEVICES INTERFACE

An 8-bit I/O bus, called the Interface Vector (IV) data bus, is used by the Microcontroller to communicate with 2 fields of I/O devices. The complementary $\overline{LB}$ and $\overline{RB}$ signals identify which field of the I/O devices is selected.

Both I/O data and I/O address information can be output on the IV bus. The SC and WC signals are typically used to distinguish between I/O data and I/O address information as follows:

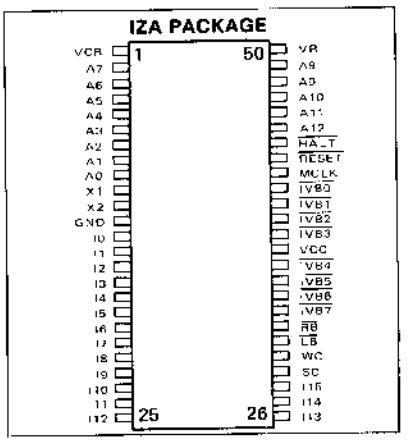
SC	WC	
1	0	I/O address is being output on IV bus
0	1	I/O data is being output on IV bus
0	0	I/O data is expected on the IV, bus, as input to the Microcontroller
1	1	Not generated by the Microcontroller

The Signetics 82SXXX series RAM, and the 8T32/33 may be attached to the IV bus.

FEATURES

- **185ns instruction decode and execute delay (with Signetics 8T32/33 I/O port)**
- **Eight 8-bit working registers**
- **Single instruction access to 1-bit, 2-bit, 3-bit or 8-bit field on I/O bus**
- **Separate instruction address, instruction, and I/O data buses**
- **On-chip oscillator**
- **Bipolar Schottky technology**
- **TTL inputs and outputs**
- **Tri-state output on I/O data bus**
- **+5 volt operation from 0° to 70°C**

PIN CONFIGURATION



PIN DESIGNATION

PIN	SYMBOL	NAME AND FUNCTION	TYPE
2-9, 45-49	A0-A12:	Instruction address lines. A high level equals "1." These outputs directly address up to 8192 words of program storage. A12 is least significant bit.	Active high
13-28	I0-I15:	Instruction lines. A high level equals "1." Receives instructions from Program Storage. I_{15} is least significant bit.	Active high
33-36, 38-41	$\overline{IVB0}$-$\overline{IVB7}$	Interface Vector (IV) Bus. A low level equals "1." Bidirectional tri-state lines to communicate with I/O devices. $\overline{IVB7}$ is least significant bit.	Three-state Active low
42	$\overline{MCLK}$:	Master Clock. Output to clock I/O devices, and/or provide synchronization for external logic	
30	WC:	Write Command. High level output indicates data is being output on the IV Bus.	Active high
29	SC:	Select Command. High level output indicates that an address is being output on the IV Bus.	Active high
31	$\overline{LB}$:	Left Bank. Low level output to enable one of two sets of I/O devices ($\overline{LB}$ is the complement of RB).	Active low
32	$\overline{RB}$:	Right Bank. Low level output to enable one of two sets of I/O devices ($\overline{RB}$ is the complement of $\overline{LB}$).	Active low
44	$\overline{HALT}$:	Low level is input to stop the Microcontroller.	Active low
43	$\overline{RESET}$:	Low level is input to initialize the Microcontroller.	Active low
10-11	X1, X2:	Inputs for an external frequency determining crystal. May also be interfaced to logic or test equipment.	
50	VR	Reference voltage to pass transistor	
1	VCR	Regulated output voltage from pass transistor.	
37	V_{CC}:	5V power connection.	
12	GND:	Ground.	

MICROCONTROLLER ARCHITECTURE

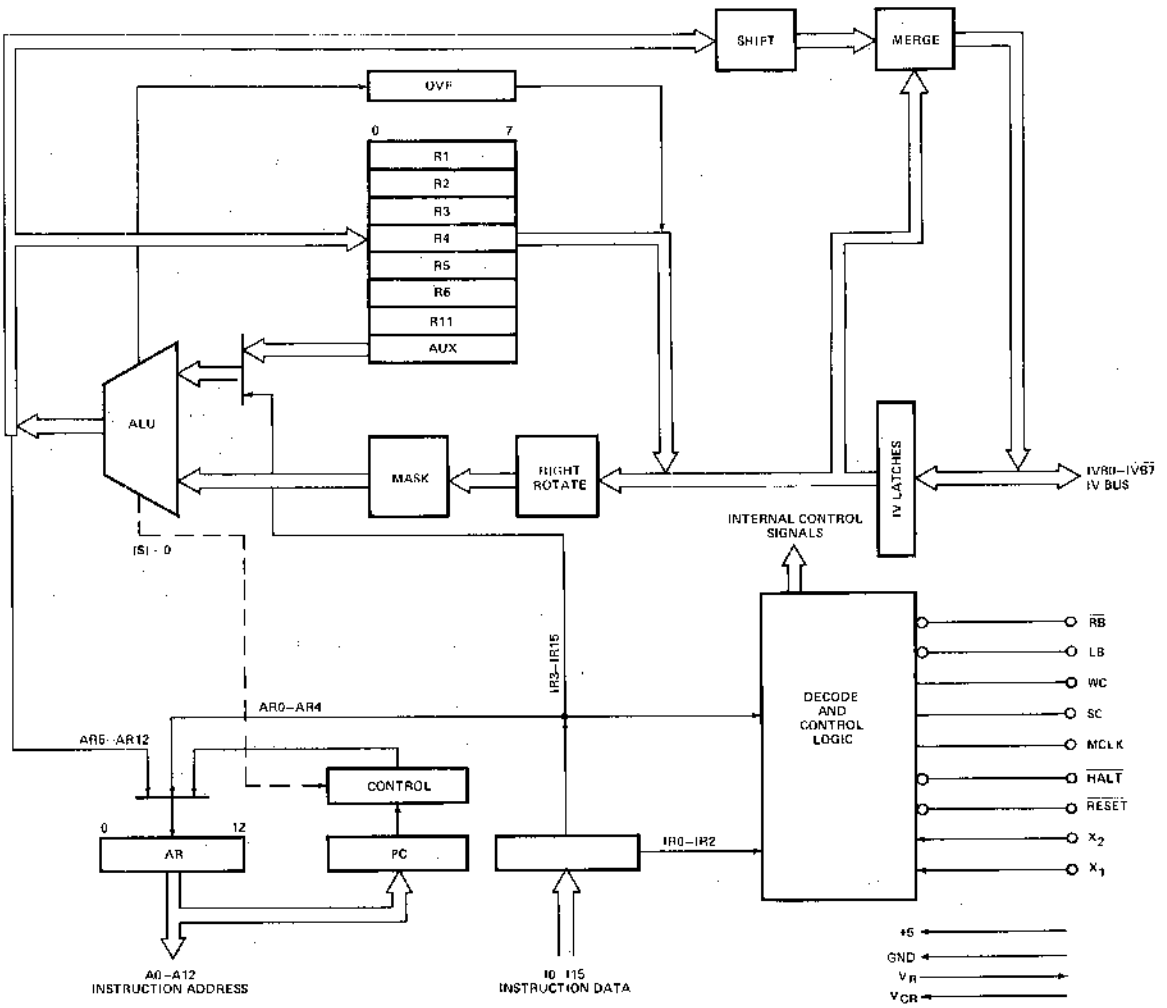


Figure 1

R1 — General working register
R2 — General working register
R3 — General working register
R4 — General working register
R5 — General working register
R6 — General working register
R11 — General working register
AUX — General working register. Contains second term for arithmetic or logical operations.

OVF — The least-significant bit of this register is used to reflect overflow status resulting from the most recent ADD operation (see Instruction Set Summary).

Instruction Register (IR)
— Holds the 16-bit instruction word currently being executed.

Program Counter (PC)
— Normally contains the address of the current instruction and is incremented to obtain the next instruction address.

Address Register (AR)
— A 13-bit register containing the address of the current instruction.

Table 1 INTERNAL REGISTERS

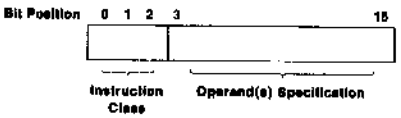
INSTRUCTION CYCLE

Each Microcontroller operation is executed in 1 instruction cycle, which may be as short as 250ns. The Microcontroller generates MCLK to synchronize external logic to the instruction cycle. Instruction cycles are subdivided into quarter cycles. MCLK is an output during the last quarter cycle.

During the third quarter cycle of an instruction, an address is output on A0-A12, identifying the location in program storage of the next instruction word. This instruction word defines the next instruction, which must be input on I0-I15 during the first quarter cycle of the next instruction cycle (see Table 2).

Instruction Set Summary

The 16-bit instruction word input on I0-I15 is decoded by the instruction decode logic to implement events that are to occur during the remainder of the instruction cycle. Generally the 16-bit instruction word is decoded as follows:



A detailed usage of the 13 "operand(s) specification" bits is given in following sections.

Three operation code bits allow for 8 instruction classes. The 8 instruction classes are summarized in Table 3. Each entry is referred to as an "instruction class" because the unique architecture of the Interpreter allows a number of powerful variations to be specified by the 13 operand(s) specification bits. A complete description of instruction formats and some instruction examples are provided in the Microprocessor Applications manual.

Data Processing

The Microcontroller architecture includes eight 8-bit working registers, an arithmetic logic unit (ALU), an overflow register, and the 8-bit IV Bus. Internal 8-bit data paths connect the registers and IV Bus to the ALU inputs, and the ALU output to the registers and IV Bus. Data processing logic is distributed along these internal 8-bit data paths. Rotate and mask logic precedes the ALU on the data entry path. Shift and merge logic follows the ALU on the data output path. All 4 sets of logic can operate on 8 data bits in a single instruction cycle (See Figure 1).

When less than 8 bits of data are specified for output to the IV bus by the ALU, the data field (shifted if necessary) is inserted into the prior contents of the IV bus latches. The IV bus latches contain data input at the start of an instruction. This data in the IV bus latches will be specified in the instruction as a) IV bus source data or b) data from an automatic read when the IV bus is specified as a destination. Therefore, IV bus bit positions outside an inserted bit field are unmodified.

Data Addressing

Sources and destinations of data are specified using a 5-bit octal number. The source and/or destination of data to be operated upon is specified in a single instruction word.

Referring to Figure 1, the Auxiliary register (address 00) is the implied source of the second argument for ADD, AND or XOR operations.

IVL and IVR are write-only registers used only as a destination. They have addresses and are treated as registers, but in reality they do not exist. When IVL is specified as a destination or the D field = $20\text{-}27_8$, then LB = 'low', RB = 'high' are generated; when IVR is specified as a destination or the D field = $30\text{-}37_8$, then RB = low, LB = 'high' are generated.

When IVL or IVR is specified as the destination in an instruction, SC is also activated and data is placed on the IV bus. If IVL or IVR is specified as a source of data, the source data is all zeroes.

TYPICAL SYSTEM CONFIGURATION

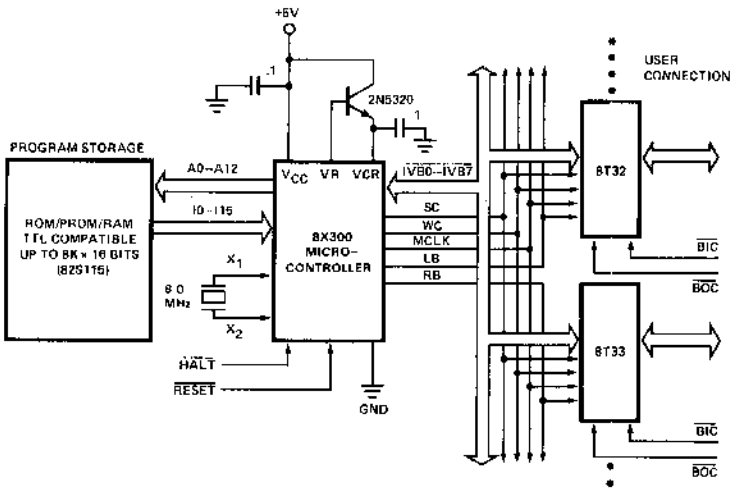


Figure 2

INSTRUCTION AND IV BUS DATA INPUT	DATA PROCESSING	ADDRESS AND IV BUS CHANGING	ADDRESS AND IV BUS DATA VALID MCLK-HIGH
← ¼ cycle →	← ¼ cycle →	← ¼ cycle →	← ¼ cycle →

Table 2 INSTRUCTION CYCLE

INSTRUCTION SEQUENCE CONTROL

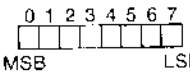
The Address Register and Program Counter are used to generate addresses for accessing an instruction. The Address Register is used to form the instruction address, and in all but 3 instructions (XEC, NZT, and JMP) the address is copied into the Program Counter. The instruction address is formed in 1 of 3 ways:

1. For all instructions but the JMP, XEC, and a satisfied NZT, the Program Counter is incremented by 1 and placed in the Address Register.
2. For the JMP instruction, the full 13-bit address field from the JMP instruction is placed into the Address Register and copied into the Program Counter.
3. For the XEC and NZT instructions, the high order 5- or 8-bits of the Program Counter are combined with 8- or 5-lower-order bits of ALU output (XEC or NZT) and placed in the Address Register. For the NZT instruction, it is also copied into the Program Counter.

INSTRUCTION SET

The 8X300 Microcontroller has a repertoire of 8 instruction classes which allow the user to test input status lines, set or reset output control lines, and perform high speed input/output data transfers. All instructions are 16 bits in length and each is fetched, decoded and executed in 250ns.

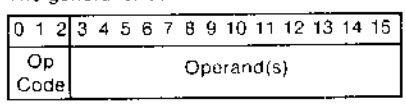
Data is respresented as an 8-bit byte; bit positions are numbered from left to right, with the least significant bit in position 7.



Within the 8X300, all operations are performed on 8-bit bytes. Arithmetic operations use 8-bit, unsigned 2's complement arithmetic.

INSTRUCTION FORMATS

The general 8X300 instruction format is:



All instructions are specified by a 3-bit Operation (Op Code) field. The operand may consist of the following fields: Source (S) field, Destination (D) field, Rotate/Length (R/L) field, Immediate (I) field, or (Program Storage) Address (A) field

The instructions are divided into 5 format types, based on the Op Code and the Operand(s), as shown in Figure 3.

OPERATION	FORMAT	RESULT	NOTES
MOVE	I,II	Content of data field specified by {S, R/L} replaces data in field specified by {D, R/L.}	If S and D both are registers, then R/L specifies a right rotate of the register specified by S.
ADD	I,II	Sum of AUX and data specified by {S, R/L} replaces data in field specified by {D, R/L}.	
AND	I,II	Logical AND of AUX and data specified by {S, R/L} replaces data in field specified by {D, R/L.}	
XOR	I,II	Logical exclusive OR of AUX and data specified by {S, R/L} replaces data in field specified by {D, R/L.}	
XMIT	III,IV	The literal value I replaces the data in the field specified by {S,L.}	If S is an I/O address then I is limited to range 00-37. Otherwise I is limited to range 000-377.
NZT	III,IV	If the data in the field specified by {S, L} equals zero, perform the next instruction in sequence. If the data specified by {S,L} is not equal to zero, execute the instruction at address determined by using the literal I as an offset to the Program Counter.	If S is an I/O address then I is limited to range 00-37. Otherwise I is limited to range 000-377.
XEC	III,IV	Perform the instruction at address determined by applying the sum of the literal I and the data specified by {S, L} as an offset to the Program Counter. If that instruction does not transfer control, the program sequence will continue from the XEC instruction location.	The offset operation is performed by reducing the value of PC to the nearest multiple of 32 (if I = 00-37) or 256 (if I = 000-377) and adding the offset.
JMP	V	The address value A replaces contents of the Program Counter.	A limited to the range $0\text{-}177778$.

Table 3 8X300 INSTRUCTION SUMMARY

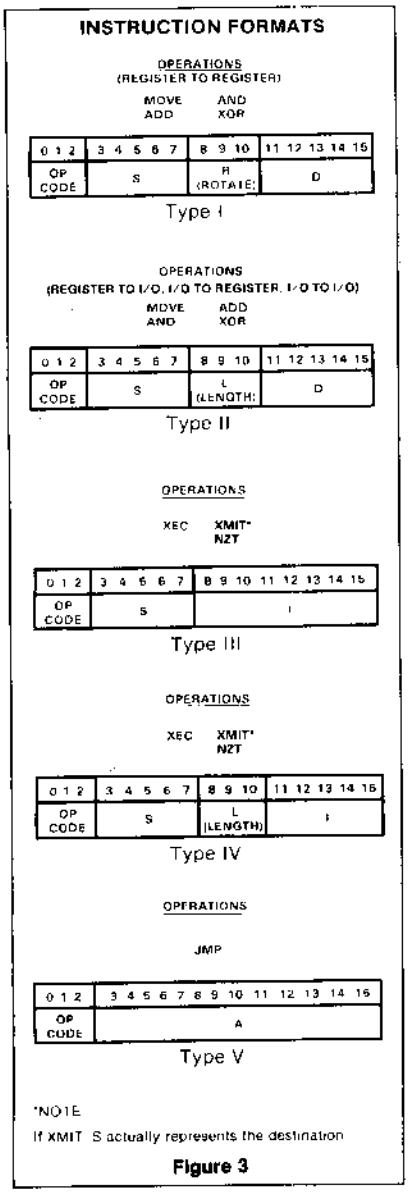


Figure 3

INSTRUCTION FIELDS

Op Code Field (3-Bit Field)

The Op Code field is used to specify 1 or 8 8X300 instructions as shown in Table 4.

OP CODE OCTAL VALUE	INSTRUCTION	
0	MOVE	S,R/L,D
1	ADD	S,R/L,D
2	AND	S,R/L,D
3	XOR	S,R/L,D
4	XEC	I,R/L,S or I,S
5	NZT	I,R/L,S or I,S
6	XMIT	I,R/L,D or I,D
7	JMP	A

Table 4 OP CODE FIELD OCTAL ASSIGNMENTS

S,D Fields (5-Bit Fields)

The S and D fields specify the source and destination of data for the operation defined by the Op Code field. The Auxiliary Register is an implied second source for the instructions ADD, AND and XOR, each of which require two source fields. That is, instructions of the form,

ADD X, Y

imply a third operand, say Z, located in the Auxiliary Register so that the operation which takes place is actually X + Z, with the result stored in Y.

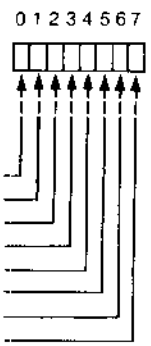
The S and/or D fields may specify a register, or a 1 to 8-bit I/O field. S and D field value assignments in octal are shown in Table 5.

0_8-17_8 is used to specify 1 of 7 working registers (R1-R6, R11), the Auxiliary Register, the Overflow Register, or IVL and IVR write-only registers.

OCTAL VALUE		OCTAL VALUE	
00	AUX-Auxiliary Register	10	OVF-Overflow register-Used only as a source
01	R1	11	R11
02	R2	12	Unassigned
03	R3	13	Unassigned
04	R4	14	Unassigned
05	R5	15	Unassigned
06	R6	16	Unassigned
07	IVL Register-Left Bank I/O address register. Used only as a destination.	17	IVR Register-Right Bank I/O address register. Used only as a destination.

a. Register Specification

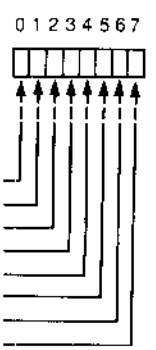
20_8-27_8 is used to specify the least significant bit of a variable length field within the I/O Port previously selected by the IVL register. The length of the field is determined by R/L.



OCTAL VALUE	
20	Field within previously selected Port; position of LSB = 0
21	Field within previously selected Port; position of LSB = 1
22	Field within previously selected Port; position of LSB = 2
23	Field within previously selected Port; position of LSB = 3
24	Field within previously selected Port; position of LSB = 4
25	Field within previously selected Port; position of LSB = 5
26	Field within previously selected Port; position of LSB = 6
27	Field within previously selected Port; position of LSB = 7

b. Left Bank I/O Field Specification

30_8-37_8 is used to specify the least significant bit of a variable length field within the I/O Port previously selected by the IVR Register. The length of the field is determined by R/L.



OCTAL VALUE	
30	Field within previously selected Port; position of LSB = 0
31	Field within previously selected Port; position of LSB = 1
32	Field within previously selected Port; position of LSB = 2
33	Field within previously selected Port; position of LSB = 3
34	Field within previously selected Port; position of LSB = 4
35	Field within previously selected Port; position of LSB = 5
36	Field within previously selected Port; position of LSB = 6
37	Field within previously selected Port; position of LSB = 7

c. Right Bank I/O Field Specification

Table 5 S AND D FIELD OCTAL ASSIGNMENTS

R/L Field (3-Bit Field)

The R/L field performs one of two functions, specifying either a field length (L) or a right rotation (R). The function it specifies for a given instruction depends upon the contents of the S and D fields:

A. When both S and D specify registers, the R/L field is used to specify a right rotation of the data specified by the S field. (Rotation occurs on the bus and not in the source register.) The register source data is right rotated within one instruction cycle time independent of the number of bit positions specified in the R/L field.

B. When either or both the S and D fields specify a variable length I/O data field, the R/L field is used to specify the length of that data field.

C. R/L field assignments are shown in Table 6.

R/L FIELD OCTAL VALUE	SPECIFICATION
0	Field Length = 8 Bits
1	Field Length = 1 Bit
2	Field Length = 2 Bits
3	Field Length = 3 Bits
4	Field Length = 4 Bits
5	Field Length = 5 Bits
6	Field Length = 6 Bits
7	Field Length = 7 Bits

Table 6 R/L FIELD OCTAL ASSIGNMENTS

I Field (5/8-Bit Field)

The I field is used to load a literal value (contained in the instruction) into a register, or a variable I/O data field, or to modify the low order bits of the Program Counter.

The length of the I field is based on the S field in XEC, NZT, and XMIT instructions, as follows:

A. When S specifies a register, the literal I is an 8-bit field (Type III format).

B. When S specifies variable I/O data field, the literal I is a 5-bit field (Type IV format).

A Field (13-Bit Field)

The A field is a 13-bit Program Storage address field. This allows the 8X300 to directly address 8192 instructions.

REGISTER OPERATIONS

When a register is specified as the source and a variable I/O data field is specified as the destination, the low order bits of the results of the instructions MOVE, ADD, XOR are merged with the original destination data.

When an I/O data field of 1 to 8 bits is specified as the source, and a register as the destination, the 8-bit result of the operations MOVE, ADD, AND, XOR is stored in the register. The operations ADD, AND, XOR actually use the I/O data field (1 to 8 bits) with leading zeros to obtain 8-bit source data for use with the 8-bit AUX data during the operation.

IVL and IVR are write-only pseudo registers, and therefore can be specified as destination fields only. Operations involving IVL and IVR as sources are not possible. For example, it is not possible to increment IVR or IVL in a single instruction, and the contents of IVL or IVR cannot be transferred to a working register, or I/O Port.

The OVF (Overflow) Register can only be used as a source field; it is set or reset *only* by the ADD instruction.

ADDRESSING DATA ON THE INTERFACE VECTOR

I/O data fields are implemented via general purpose 8-bit I/O registers called Interface Vector (IV) Bytes. The IV registers serve to select IV bytes. In order for an instruction to access (read or write) an I/O data field, the address must be output to the IVL or IVR registers.

Thus, two instructions are required to operate on an Interface Vector byte.

```
XMIT    ADDRESS, IVL
MOVE    LB, RB
```

Each of the two IV registers (IVL and IVR) may be set to select an IV byte, therefore two I/O ports may be active at one time—one on the Right Bank (IVR) and one on the Left Bank (IVL). Data may be input and output in one instruction following the selection of IV bytes:

```
XMIT    ADDRESS1, IVL
XMIT    ADDRESS2, IVR
ADD     LB, RB
```

Once the IV byte is selected (addressed) it will remain selected until another address is output to the same IV register. Since an IV register (IVL, IVR) can be used only as a destination field of an instruction, any instruction sending data to IVL or IVR can be used to select an IV byte.

From the user's standpoint, however, all IV byte outputs can be read by an external device regardless of whether they are selected or not.

The address range of IVL and IVR is $0\text{-}255_{10}$.

INSTRUCTION DESCRIPTIONS

The following instruction descriptions employ MCCAP (the 8X300 Cross Assembly Program) programming notation. This notation varies somewhat from the instruction descriptions provided in Tables 3 through 5. Thus, for example, explicit L field definition, as shown in Table 3 and Table 4 is not required by MCCAP instructions; MCCAP can create appropriate variable field addresses from information contained in Data Declaration statements which may be provided by the programmed at the beginning of his program.

The 8X300 instruction set is described below with examples shown in Figures 4 through 11.

MOVE S,D or MOVE S(R),D

Format: Type I, Type II

Operation: (S)→(D)

Description

Move data. The contents of S are transferred to D; the contents of S are unaffected. If both S and D are registers, R/L specifies a right rotate of the source data before the move. Otherwise, R/L specifies the length of the source and/or destination I/O data field. If the MOVE is between Left Bank and Right Bank I/O field, an 8-bit field must always be moved.

Example

Store the least significant 3 bits of register 5 (R5) in bits 4, 5 and 6 of the I/O Port previously addressed by the IVL register. See Figure 4.

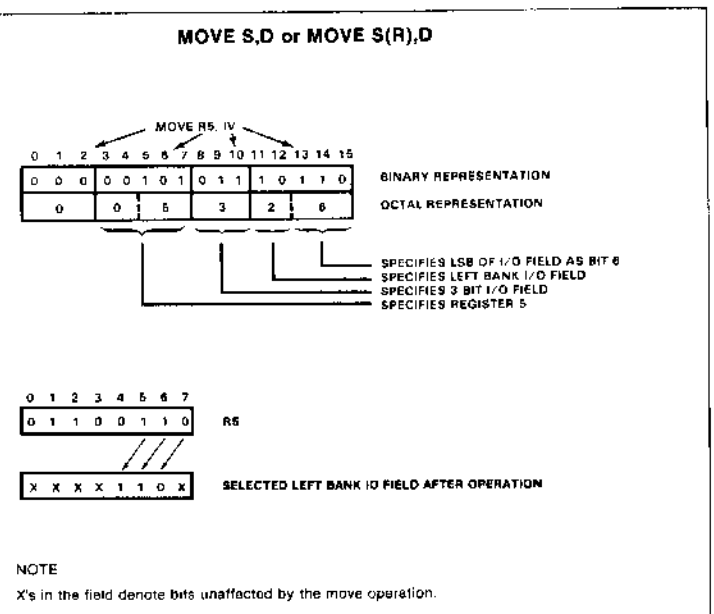


Figure 4

ADD S,D or ADD S(R),D

Format: Type I, Type II

Operation: (S) + (AUX)→D
Carry→OVF

Description

Unsigned 2's complement 8-bit addition. The contents of S are added to the contents of the Auxiliary Register. The result is stored in D; OVF is set to the value of the carry. If both S and D are registers, R/L specifies a right rotate of the source (S) data before the operation. Otherwise, R/L specifies the length of the source and/or destination I/O data fields. S and AUX are unaffected unless specified as the destination.

Example

Add the contents of R1 rotated 4 places to AUX and store the result in R3. See Figure 5.

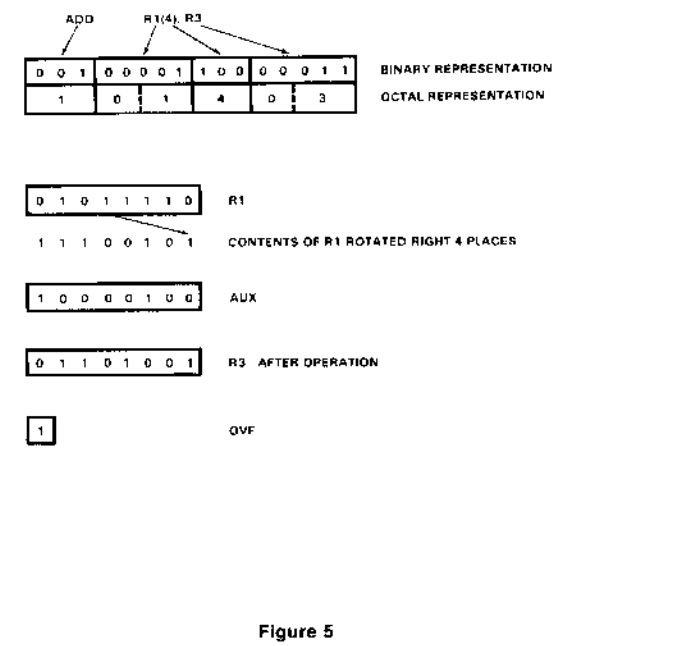


Figure 5

AND S,D or AND S(R),D

Format: Type I, Type II

Operation: (S) ∧ (AUX)→D

Description

Logical AND. The AND of the source field and the Auxiliary Register is stored into the destination. If both S and D are registers, R/L specifies a right rotate of the source (S) data before the AND operation. Otherwise R/L specifies the length of the source and/or destination I/O data fields. S and AUX are unaffected unless specified as a destination.

Example

Store the AND of the selected right bank I/O field and AUX in R4. The right bank data field is called WSBCD and is 4 bits long and located in bits 2, 3, 4 and 5. See Figure 6.

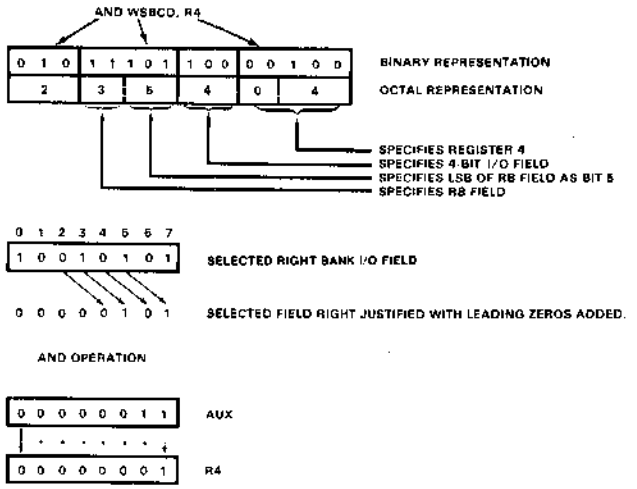


Figure 6

XOR S,D or XOR S(R),D

Format: Type I, Type II

Operation: (S) ⊕ (AUX)→D

Description

Exclusive-OR. The Exclusive-OR of the source field and the Auxiliary Register is stored in the destination. If both S and D are registers, R/L specifies a right rotate of the source (S) data before the XOR operation. Otherwise R/L specifies the length of the source and/or destination I/O data fields. S and AUX are unaffected unless specified as a destination.

Example

Replace the selected I/O data field with the XOR of the field and AUX. The I/O data field is called STATUS and is 5 bits in length and located in bits 3, 4, 5, 6 and 7 of left bank. See Figure 7.

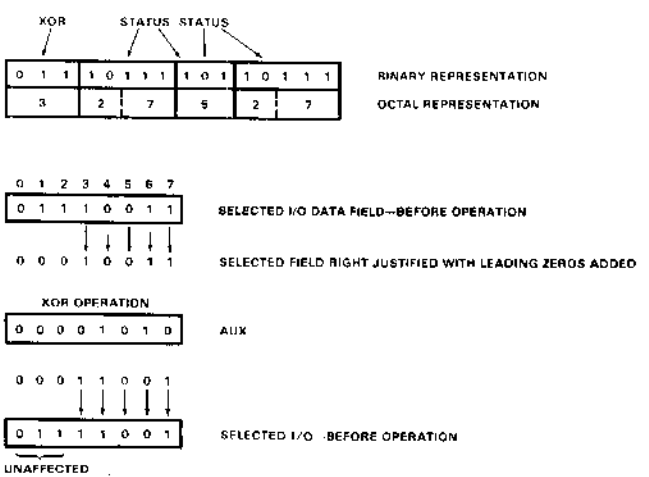


Figure 7

XEC I(S)

Format: Type III, Type IV

Operation

Execute instruction at the address specified by the Address Register with lower 5 or 8 bits replaced by (S) + I.

Description

Execute the instruction at the address determined by replacing the low order bits of the Address Register (AR) with the low order bits of the sum of the literal I and the contents of the source field. If S is a register, the low order 8 bits of AR are replaced; if S is an I/O data field, the low order 5 bits of AR are replaced, resulting in an execute range of 256 and 32 respectively. The Program Counter is not affected unless the instruction executed is a JMP or NZT (whose branch is taken).

Example

Execute one of n JMPs in a table of JMP instructions determined by the value of the selected I/O data field on the left bank. The table follows immediately after the XEC instruction and the I/O field is called INTERPT and is a 3-bit field located in bits 4, 5 and 6. See Figure 8.

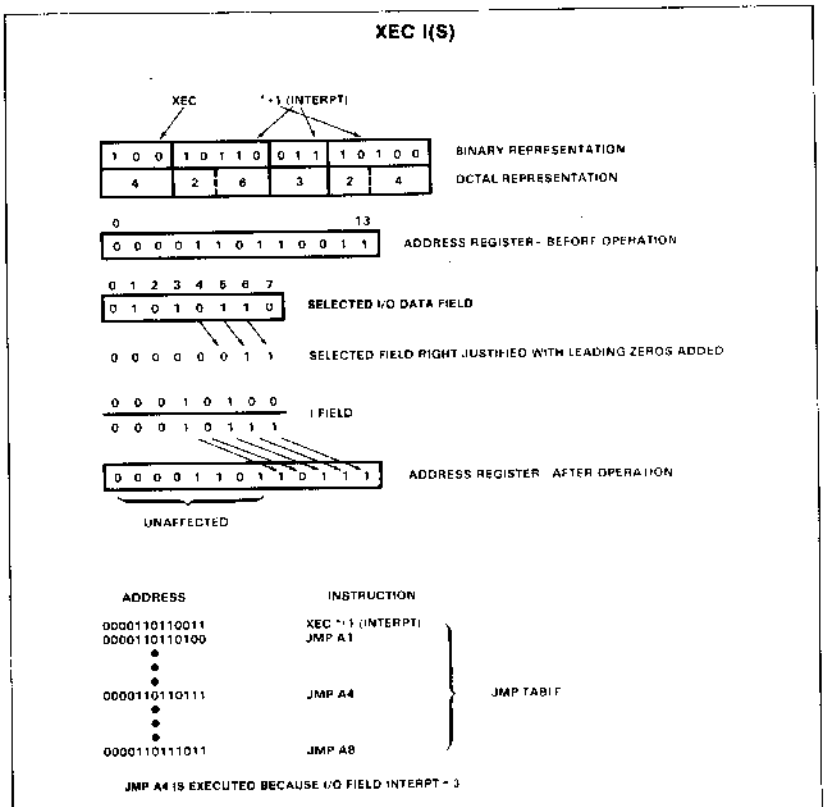


Figure 8

XMIT I,D

Format: Type III, Type IV

Operation: I→(D)

Description

Transmit literal. The literal field I is stored in D. If D is a register, an 8-bit field is transferred; if D is an I/O data field, up to a 5-bit field is transferred.

Example

Store the bit pattern 110 in the selected I/O data field on the right bank. The field name is VALUE and is located in bits 3, 4 and 5. See Figure 9.

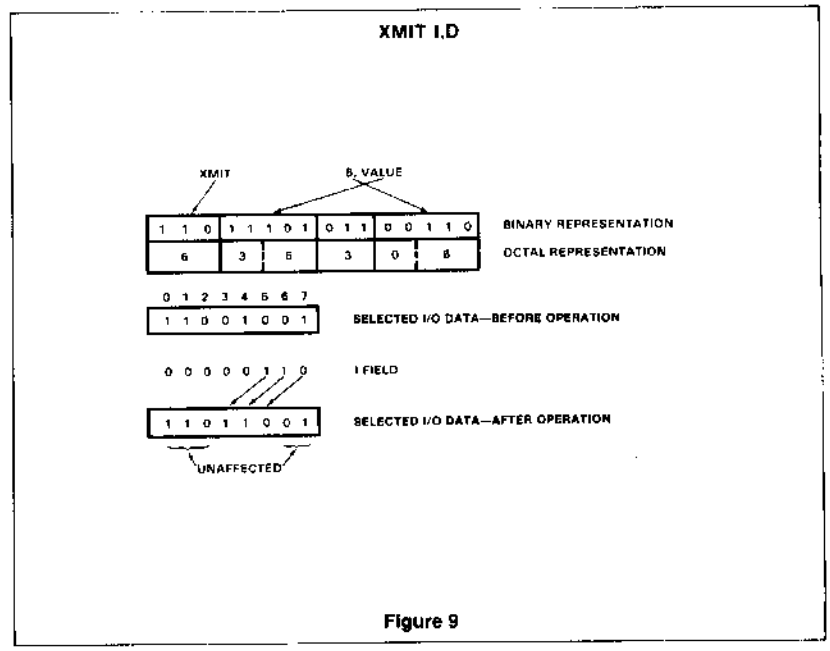


Figure 9

NZT S,I

Format: Type III, Type IV

Operation:

Non-Zero Transfer. If (S) ≠ 0, PC offset by I → PC; otherwise PC + 1 → PC.

Description

If the data specified by the S field is non-zero, replace the low order bits of the Program Counter with I. Otherwise, processing continues with the next instruction in sequence. If S is a register, the low order 8 bits of the PC are replaced; if S is an I/O data field, the low order 5 bits of the PC are replaced, resulting in an NZT range of 256 and 32 respectively.

Example

Jump to Program Address ALPHA if the selected right bank I/O field is non-zero. The field name is OVERFLO and it is a 1-bit field located in bit 3. See Figure 10.

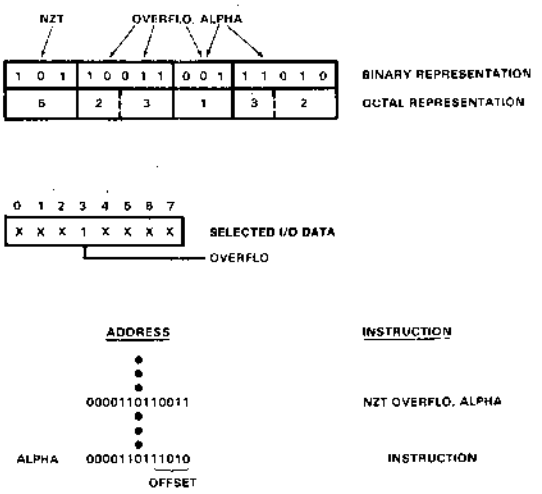


Figure 10

JMP A

Format: Type V

Operation: A → PC

Description

The literal value A is placed in the Program Counter and Address Register, and processing continues at location A. A has a range of $0\text{-}17777_8$ (0-8191).

Example

Jump to location ALPHA (0000101110001). See Figure 11.

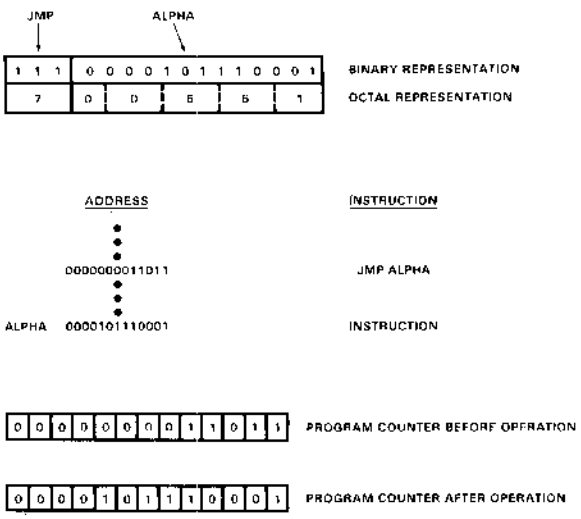


Figure 11

SYSTEM DESIGN USING THE 8X300 MICROCONTROLLER

Designing hardware around the 8X300 Interpreter reduces to selecting a program storage device (ROM, PROM, etc.), selecting I/O devices (IV byte, multiplexers, RAM, etc.), selecting clock mode (system driven or crystal controlled) and interfacing the Microcontroller to these components.

A specific example of a control system using the 8X300 Microcontroller is shown in Figure 12. Only 8 components—four 8T32 I/O Ports, one 82S208 RAM, two 82S215 ROMs, and an 8X300 are required to build this system which contains 512 words of program storage, 32 TTL I/O connection points, 256 bytes of working storage, and operates at a 250ns instruction cycle time.

Halt, Reset Signals

$\overline{\text{HALT}}$:
A low level at the $\overline{\text{HALT}}$ input stops internal operation of the Microcontroller at the start of the next instruction after $\overline{\text{HALT}}$ is applied (quarter cycle after MCLK). Since $\overline{\text{HALT}}$ is sampled at the start of each instruction cycle it is possible to prevent a cycle by applying $\overline{\text{HALT}}$ early in that cycle. $\overline{\text{HALT}}$ does not inhibit MCLK or affect any internal registers. Normal operation begins with the next complete cycle after the $\overline{\text{HALT}}$ input goes high.

$\overline{\text{RESET}}$:
A low level at the $\overline{\text{RESET}}$ input sets the program counter and address register to zero. While $\overline{\text{RESET}}$ is low MCLK is inhibited. If $\overline{\text{RESET}}$ is applied during the last 2 quarter cycles, the MCLK during that cycle may be shortened. $\overline{\text{RESET}}$ should be applied for 1 full instruction cycle time to assure proper operation. When $\overline{\text{RESET}}$ input goes high an MCLK occurs prior to the resumption of normal processing. $\overline{\text{RESET}}$ does not affect the other internal registers.

SYSTEM TIMING

In systems with fast instruction cycle times, most Microcontroller delays are strictly determined by internal gate propagation delays.

EXAMPLE OF CONTROL SYSTEM

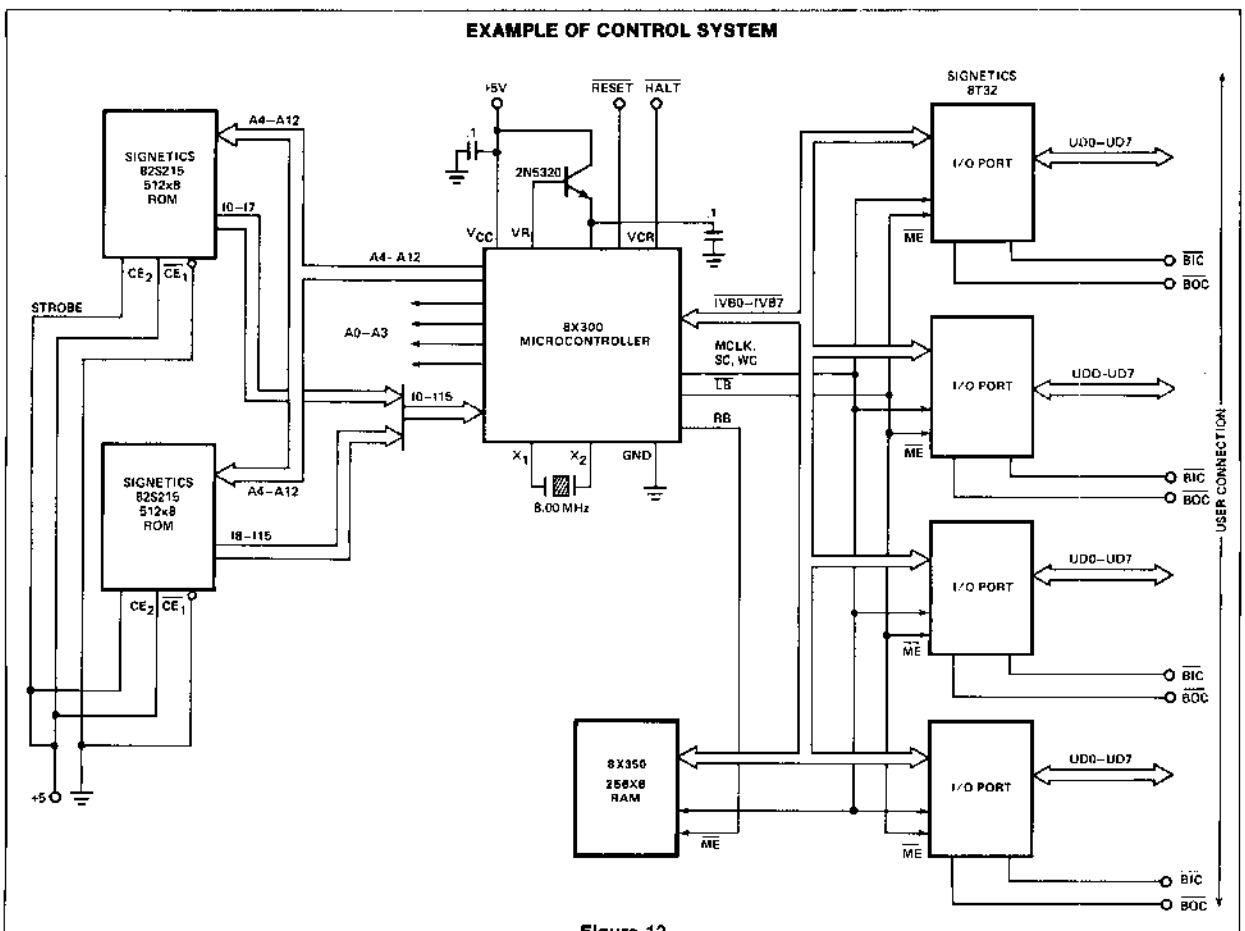


Figure 12

Since some events are constrained to occur in certain quarter cycles, as system cycle times become slower, the delays will appear to increase due to gating with internal clocks. In the table of AC Electrical Characteristics, 2 columns are used: 1 to denote times which occur due to internal clock intervention and 1 to denote minimum delays for fast cycle times.

When using Signetics 8T32 I/O Ports, selection of instruction cycle time involves calculating the maximum program storage access time. Assuming the instruction is available when MCLK falls, the I/O control lines are stable 35ns later. Signetics 8T32's require another 35ns to disable a previously selected port and enable the desired port (assumes a change in bank signals). A 10ns margin has been added to the 8T32 enable for this evaluation to reflect the fact that most systems will have more capacitive loading than the 50pF test condition in the 8T32 specification and to allow for line delays.

The system instruction cycle time for normal systems such as shown in Figure 12 is determined by Microcontroller propagation delays, program storage access time, and port output enable times. Instruction cycle time is normally constrained by one or more of the following conditions:

1. Instruction to LB/RB (input phase) and I/O Port output enable:
   $T_{OE} \leq$ ½ cycle - 55ns (Figure 13).
2. Program storage access time and instruction to LB/RB (input phase) and I/O Port output enable and IV data (input phase) to address ≤ instruction cycle time (Figure 14).
3. Program storage access time and instruction to address ≤ instruction cycle time (Figure 15).

The first constraint can be used to determine the minimum cycle time. Using the inequality 35ns + 35ns ≤ ½ cycle − 55ns implies ½ cycle ≥ 125ns or an instruction time of 250ns.

Program storage access time for a 250ns instruction cycle can be calculated from the second constraint. Noting that the specification for IV data (input phase) to address is 115ns: Program storage access time + 35ns + 35ns + 115ns ≤ 250ns implies program storage access time ≤ 65ns.

The third constraint can be used to verify the maximum program storage access time. Noting that the specification for instruction to address is 185ns: Program storage access time + 185ns ≤ 250ns confirms that program storage access time 65ns is satisfactory.

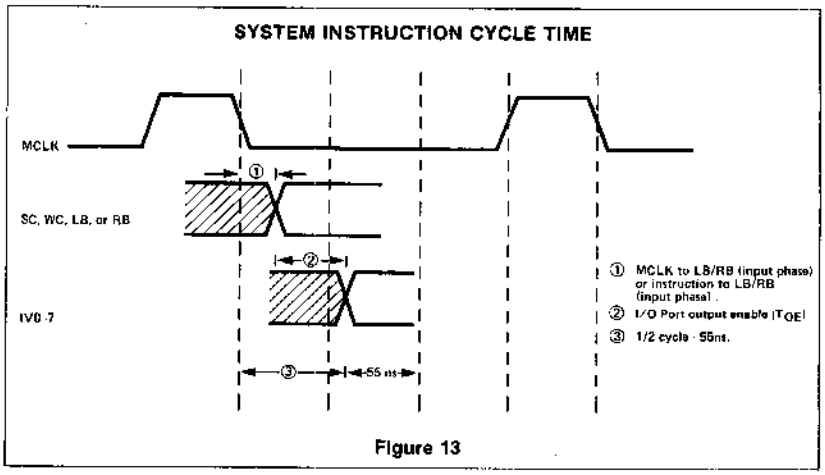


Figure 13

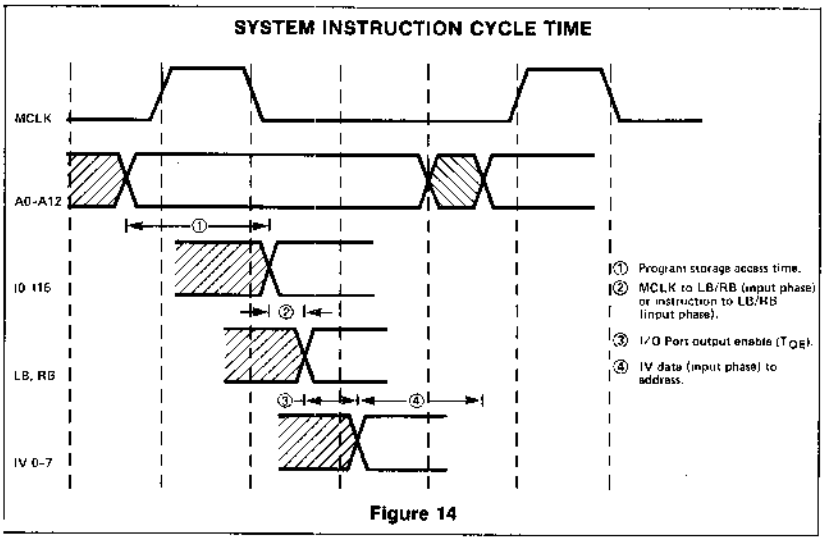


Figure 14

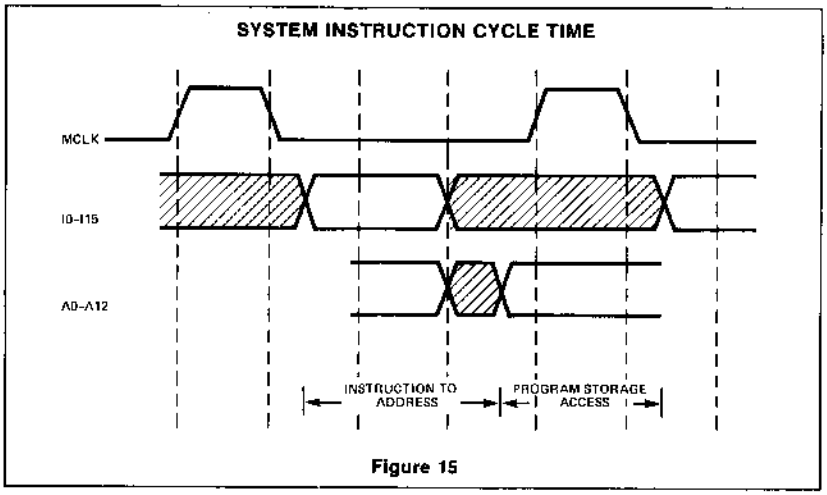


Figure 15

System Clock

The Microcontroller has an integrated oscillator which generates all necessary clock signals. The oscillator is designed to connect directly to a series resonant quartz crystal via pins X1 and X2. The crystal resonant frequency, f, is related to the desired cycle time, T, by the relationship f = 2/T. For a 250ns system, f = 8.00MHz.

Type:	Fundamental mode, series resonant
Impedance at Fundamental:	35 ohms maximum
Impedance at harmonics and spurs:	50 ohms minimum

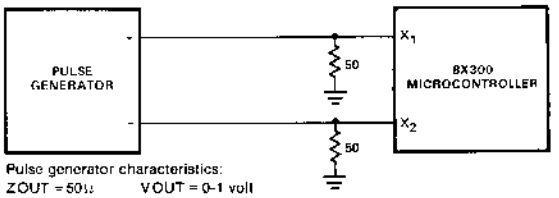
Table 7 CRYSTAL CHARACTERISTICS

In lower speed applications where the cycle time need not be precisely controlled, a capacitor may be connected between X1 and X2 to drive the oscillator. Approximate capacitor values are given in Table 8. If cycle time is to be varied, X1 and X2 should be driven from complementary outputs of a pulse generator. Figure 16 shows a typical configuration. For systems where the Interpreter is to be driven from a master clock the X1 and X2 lines may be interfaced to TTL logic as shown in Figure 17.

Cx,pF	CYCLE TIME
100	300ns
200	500ns
500	1.1μs
1000	2.0μs

Table 8 CLOCK CAPACITOR VALVES

CLOCKING WITH A PULSE GENERATOR

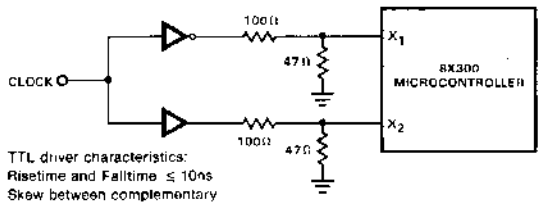


Pulse generator characteristics:
ZOUT = 50Ω VOUT = 0-1 volt
Risetime ≤ 10ns ≤ Skew ≤ 10ns
Complementary outputs

NOTE: All resistors values are typical and in ohms.

Figure 16

CLOCKING WITH TTL



TTL driver characteristics:
Risetime and Falltime ≤ 10ns
Skew between complementary outputs ≤ 10ns

Figure 17

AC ELECTRICAL CHARACTERISTICS V_{CC} = 5V ± 5% and 0°C ≤ T_A < 70°C

DELAY DESCRIPTION	PROPAGATION DELAY TIME	CYCLE TIME LIMIT
X1 falling edge to MCLK (driven from external pulse generator)	75ns	
MCLK to SC/WC falling edge (input phase)	25ns	
MCLK to SC/WC rising edge (output phase)		½ cycle + 25ns
MCLK to LB/RB (input phase)	35ns	
Instruction to LB/RB output (input phase)	35ns	
MCLK to LB/RB (output phase)		¼ cycle + 35ns
MCLK to IV data (output phase)	185ns	½ cycle + 60ns
IV data (input phase) to IV data (output phase)	115ns	
Instruction to Address	185ns	½ cycle + 40ns
MCLK to Address	185ns	½ cycle + 40ns
IV data (input phase) to Address	115ns	
MCLK to IV data (input phase)		½ cycle - 55ns
MCLK to Halt falling edge to prevent current cycle		¼ cycle - 40ns
Reset rising edge to first MCLK		0 to 1 cycle

NOTE
1. Reference to MCLK is to the falling edge when loaded with 300pF
2. Loading on Address lines is 150pF

TYPICAL INSTRUCTION CYCLE TIMING

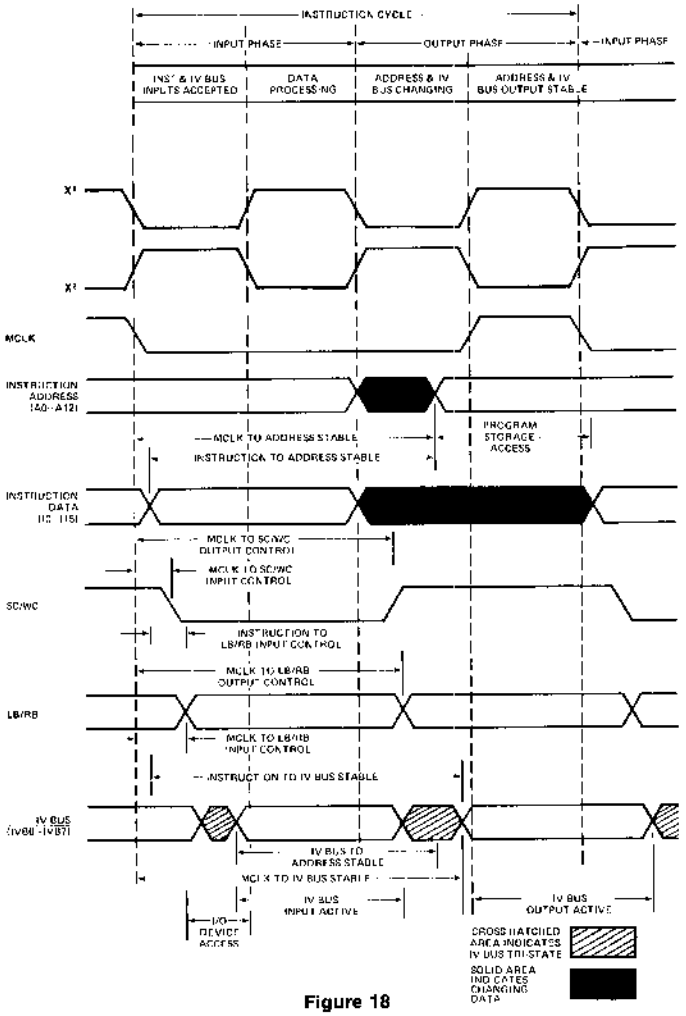


Figure 18

DC ELECTRICAL CHARACTERISTICS

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IH}	High level input voltage					
	X1,X2		.6			V
	All others		2			V
V_{IL}	Low level input voltage					
	X1,X2				.4	V
	All others				.8	V
V_{IC}	Input clamp voltage (Note 1)	V_{CC} = 4.75V, I_I = -10mA			-1.5	V
I_{IH}	High level input current					
	X1,X2	V_{CC} = 5.25V, V_{IH} = .6V		2700		μA
	All others	V_{CC} = 5.25V, V_{IH} = 4.5V		<1	50	μA
I_{IL}	Low level input current					
	X1,X2	V_{CC} = 5.25V, V_{IL} = .4V		-2500		μA
	$\overline{IVBO}$-7			-140	-200	μA
	IO-I15	V_{CC} = 5.25V, V_{IL} = .4V		-880	-1600	μA
	$\overline{HALT}$, $\overline{RESET}$	V_{CC} = 5.25V, V_{IL} = .4V		-230	-400	μA
V_{OL}	Low level output voltage	V_{CC} = 5.25V, V_{IL} = .4V				
	A0-A12	V_{CC} = 4.75V, I_{OL} = 4.25mA		.35	.55	V
	All others	V_{CC} = 4.75V, I_{OL} 16mA		.35	.55	V
V_{OH}	High level output voltage	V_{CC} = 4.75V, I_{OH} = 3mA	2.4			V
I_{OS}	Short circuit output current (Note 2)	V_{CC} = 5.25V	-30		-140	mA
V_{CC}	Supply voltage		4.75	5	5.25	V
I_{CC}	Supply current	V_{CC} = 5.25V			1.60	mA
I_{REG}	Regulator control	V_{CC} = 5.0V	-14		-21	mA
I_{CR}	Regulator current (Note 3)				290	mA
V_{CR}	Regulator voltage (Note 3)		2.2		3.2	V

NOTES

1. Crystal inputs X1 and X2 do not have clamp diodes.
2. Only one output may be grounded at a time.
3. (Limits apply for V_{CC} = 5V ± 5% and 0°C < T_A < 70°C unless specified otherwise.)

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Supply voltage	7	V
	Logic input voltage	5.5	V
	Crystal input voltage	2	V