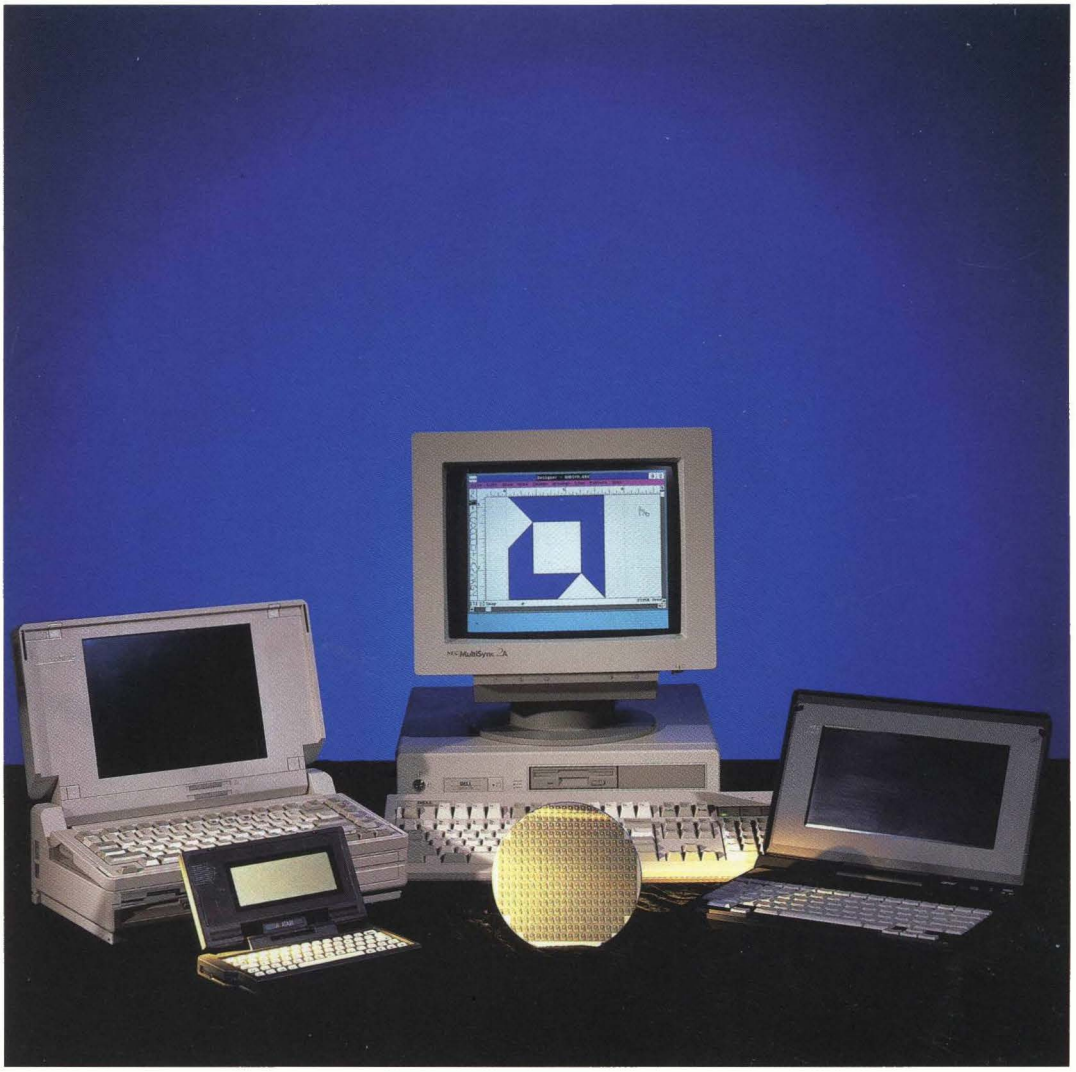
AMD

Am286™ZX/LX Integrated Processor

1991

Am286™ZX/LX Integrated Processor Technical Manual

Advanced
Micro
Devices



Am286™ZX/LX
Integrated Processor
Technical Manual

ADVANCED MICRO DEVICES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

DESIGN GOALS

The Am286ZX/LX integrated processor integrates all of the logic functions on the original AT motherboard, along with enhancements in operational speed, memory management, DRAM requirements, and configurability. AMD's goal was the integration of all "non-memory-based" functional blocks (not including the DRAMs, BIOS EPROMs, or the ROM-based 8042 keyboard controller) onto a single CMOS device.

As AMD was already producing the logic functions contained in the AT design as individual silicon devices, development of the Am286ZX/LX integrated processor was the next logical step. Also, AMD had the manufacturing and packaging technology to place all of these devices, along with flexible control logic, into a single silicon device.

Target markets for the Am286ZX/LX integrated processor shaped its feature set, taking it beyond just integrating a standard AT design. The prime system targets for the Am286ZX/LX integrated processor are 80286-based hand-held, notebook, and laptop computer systems. Another system target, due to the nature of the functional units that were combined into a single device, is the "embedded computer" system, where integration, ease of implementation, and power consumption are critical system issues.

The system requirements for these target markets prompted AMD to extend the definition of the Am286ZX/LX integrated processor to include high-speed operation with standard DRAM and system bus speeds, direct interconnection with other devices on the motherboard, and the power saving features of the Am286LX processor version.

AMD's design goals for this project were ease of implementation, low total system cost, high reliability, and strict compatibility. These design goals mapped directly into the following device specific goals.

- Direct interface of memory-based devices within the system (DRAM, EPROM, and 8042 controller).
- A triple bus design for minimizing the need for external buffers and transceivers.
- An AT-compatible bus controller with both synchronous and asynchronous operation for ease of system implementation at all CPU/system bus speed combinations.
- Fail-safe internal data routing, guaranteeing that data bus contention will not happen.
- DRAM size mixing without limitations.
- Flexible clocking options with fail-safe clock speed switching.
- Power savings by controlling internal clocks and minimizing DRAM power consumption during refreshes.
- Device configurability via software for interfacing to various speeds of DRAM, ROM, and system peripherals.

These design goals molded the Am286ZX/LX integrated processor into a system solution that breaks new ground in the integration of the PC-AT system function, surpassing other solutions forced to stop at the processor boundary. This device's adaptability to a wide variety of system environments and requirements make it an ideal solution for the implementation of 80286 microprocessor-based, AT-compatible systems.

DISTINCTIVE CHARACTERISTICS

- Integrates entire IBM PC-AT motherboard logic plus enhancements.
  - —80C286 Microprocessor core
  - —Enhanced Bus Controller
  - —Enhanced Clock Generator
  - —DMA Controllers
  - —Interrupt Controllers
  - —Counter/Timer
  - —Real Time Clock and CMOS Static RAM
- Direct connection to DRAM, AMD 80C287™ math coprocessor, EPROMs, keyboard controller, and AT-bus slots for a complete AT-compatible system.
- Full hardware support for EMS 4.0 memory management with 128 EMS registers.
- 100% compatible with the IBM PC-AT standard.
- Low power operation with power saving features for battery powered systems (Am286LX processor only).
- Industry standard I/O port 92H fast reset and GATE A20 features for high-speed switching between real and protected mode.
- Page mode/interleave DRAM Controller with direct interface to 256-kbit, 1-Mbit, and 4-Mbit DRAMs supports up to 16 Mb of physical memory.
- Complete line of support products available including Demo Boards, ICE, BIOS, and EMS Drivers.
- 28mm x 28mm, 216-Pin EIAJ Plastic Quad Flat Pack (PQFP), with socket available.

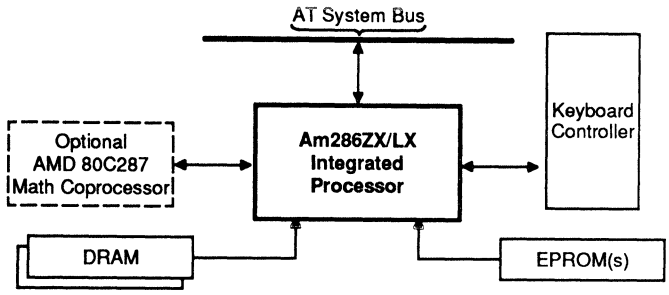
GENERAL DESCRIPTION

The Am286ZX and Am286LX integrated processors are AMD proprietary PC-AT motherboard-on-a-chip devices for personal computers. They integrate all of the logic functions from the original IBM PC-AT motherboard, plus enhancements, onto one chip. Included are the 80C286 microprocessor, all of the AT standard peripherals, and memory management, to provide a high-performance, low-cost, low-power system solution for personal computers. The high integration of the Am286ZX/LX processor allows designers to reduce size, power consumption, and cost of a PC-AT compatible system while increasing functionality and adding features.

The Am286ZX/LX integrated processor is ideal for design of hand-held, notebook, laptop, embedded, and other industry standard AT personal computers where performance, size, power consumption, and cost are critical factors. The Am286LX processor version provides additional power-saving features including CPU shutdown mode, system shutdown mode, staggered DRAM refresh, and slow-refresh DRAM support.

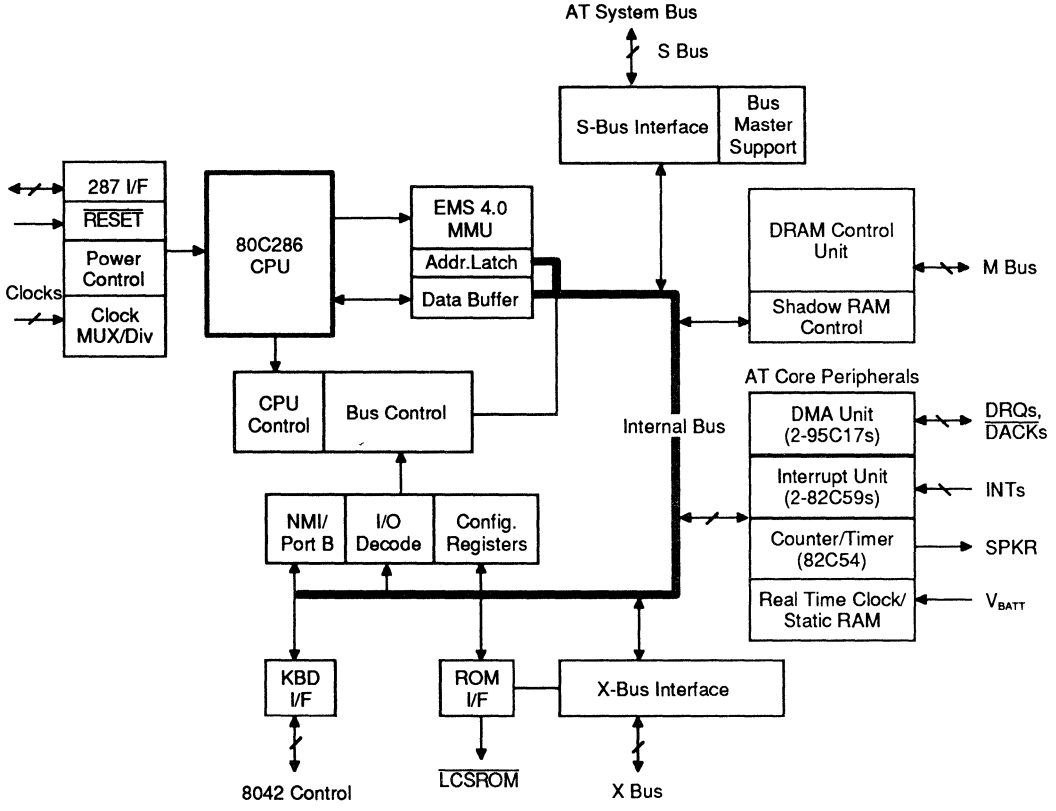
Figure 1-1 shows a system block diagram for an AT motherboard illustrating the high level of integration achieved by the Am286ZX/LX integrated processor.

Figure 1-1 PC-AT System Block Diagram Using the Am286ZX/LX Integrated Processor



14753C–001

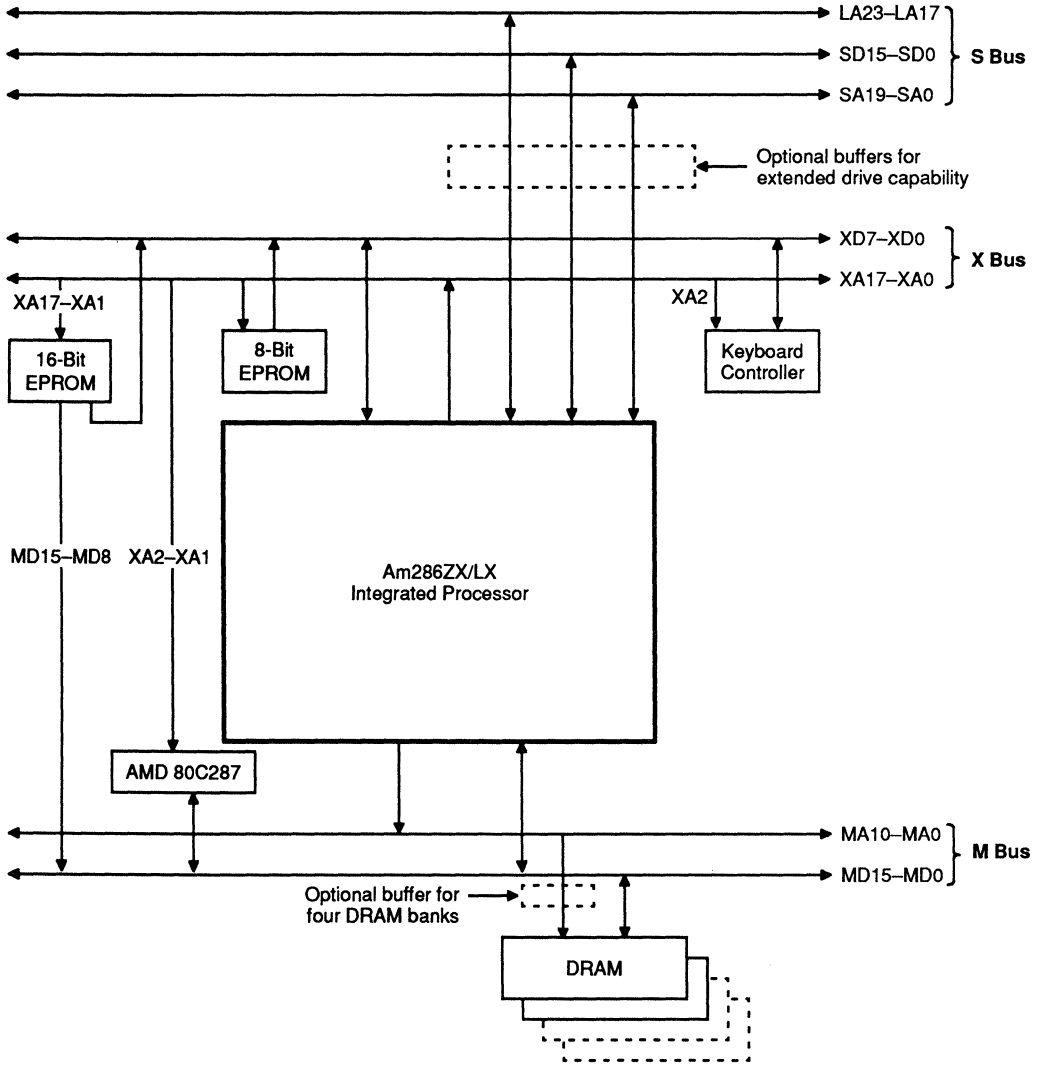
Figure 1-2 Internal Block Diagram



14753C–003

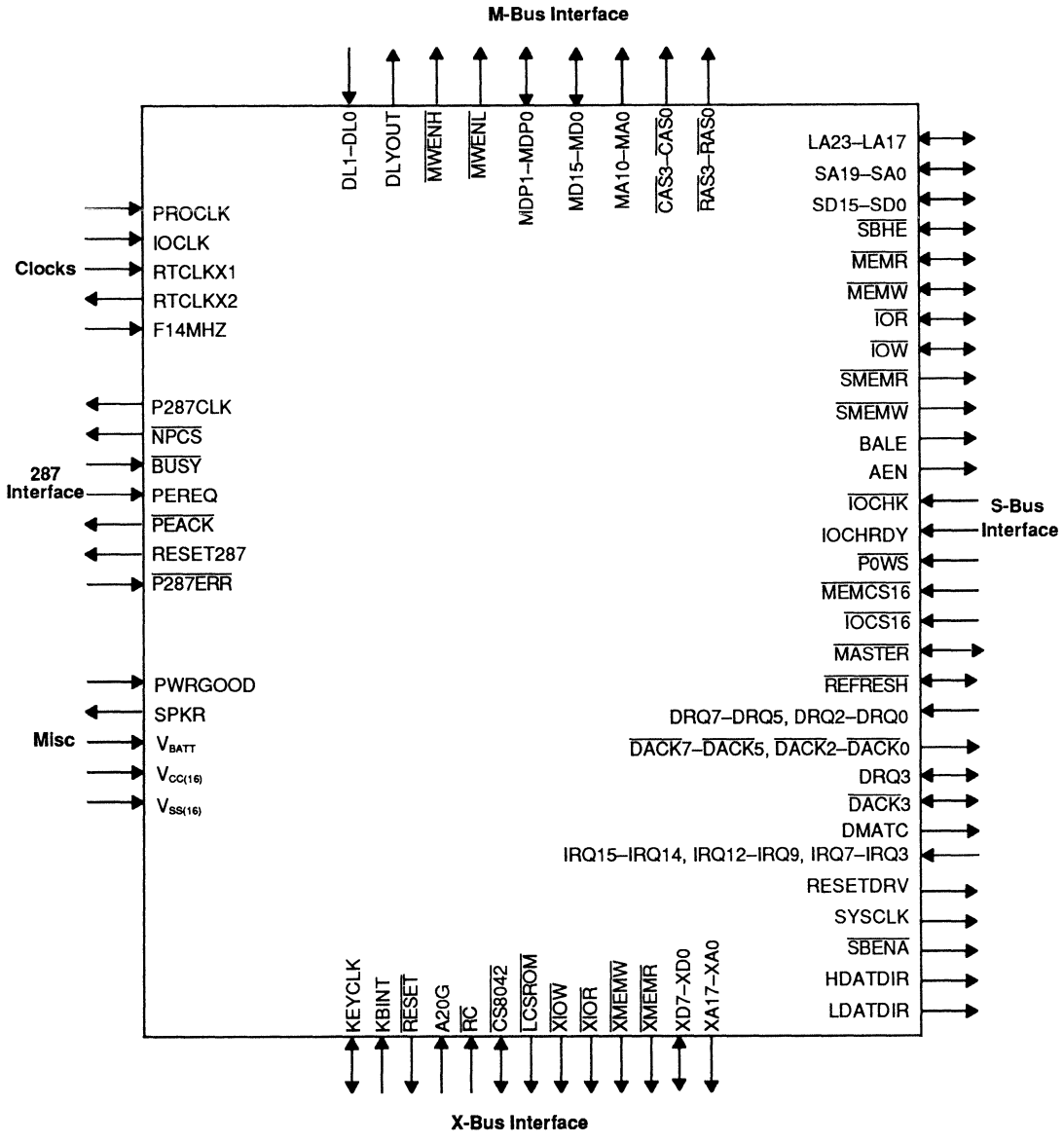
Figure 1-3 shows the Am286ZX/LX integrated processor bus interface. The AT system bus (S Bus), memory bus (M Bus), and I/O Bus (X Bus) directly interface to the rest of the system components. Optional buffers are shown for applications that require extended S-bus drive capability and/or four DRAM banks.

Figure 1-3 Am286ZX/LX Integrated Processor Bus Interface



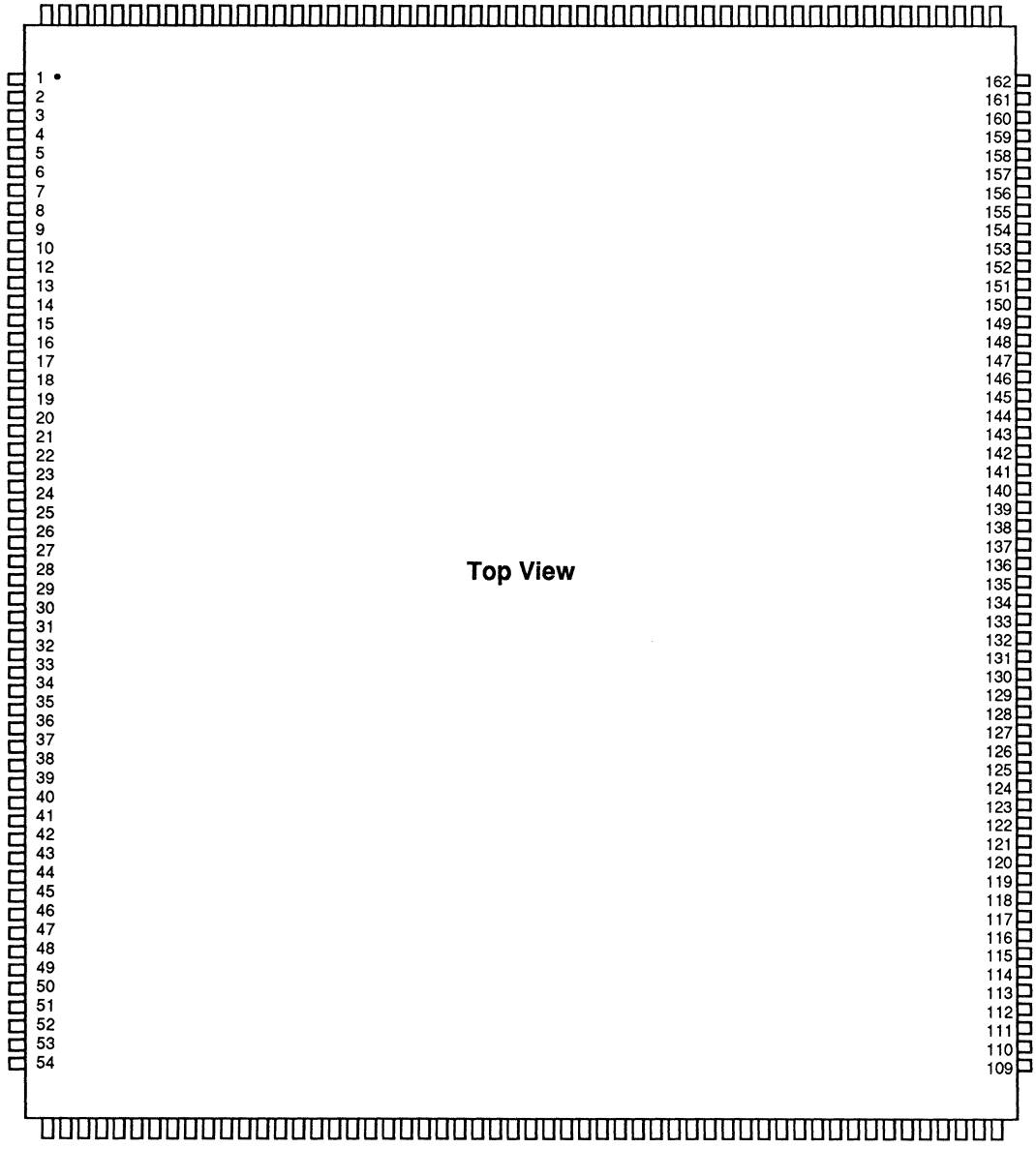
14753C–002

Figure 1-4 Am286ZX/LX Integrated Processor Logic Symbol



14753C–004

Figure 1-5 Am286ZX/LX Integrated Processor Connection Diagram



Note: Pin 1 is marked for orientation.

14753C–005

Figure 1-6 Pin Designation Table (by pin number)

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	LDATDIR	56	V_{CC}	111	SA18	166	MDP0
2	$\overline{SBENA}$	57	V_{SS}	112	LA18	167	MDP1
3	$\overline{MASTER}$	58	XA12	113	SA19	168	SD0
4	DRQ7	59	XA11	114	LA19	169	SD1
5	$\overline{DACK7}$	60	XA10	115	LA20	170	SD2
6	DRQ6	61	XA9	116	LA21	171	SD3
7	$\overline{DACK6}$	62	XA8	117	LA22	172	SD4
8	DRQ5	63	XA7	118	LA23	173	SD5
9	$\overline{DACK5}$	64	XA6	119	$\overline{RAS0}$	174	SD6
10	DRQ0	65	XA5	120	$\overline{RAS1}$	175	SD7
11	V_{CC}	66	XA4	121	V_{CC}	176	V_{CC}
12	V_{SS}	67	XA3	122	V_{SS}	177	V_{SS}
13	$\overline{DACK0}$	68	XA2	123	$\overline{RAS2}$	178	SD8
14	$\overline{MEMR}$	69	XA1	124	$\overline{RAS3}$	179	SD9
15	$\overline{MEMW}$	70	XA0	125	$\overline{CAS0}$	180	SD10
16	IRQ14	71	V_{CC}	126	$\overline{CAS1}$	181	SD11
17	IRQ15	72	V_{SS}	127	V_{CC}	182	SD12
18	IRQ12	73	MA0	128	V_{SS}	183	SD13
19	IRQ11	74	MA1	129	$\overline{CAS2}$	184	SD14
20	IRQ10	75	MA2	130	$\overline{CAS3}$	185	SD15
21	$\overline{IOCS16}$	76	MA3	131	$\overline{MWENL}$	186	V_{BATT}
22	$\overline{MEMCS16}$	77	MA4	132	$\overline{MWENH}$	187	RTCLKX1
23	$\overline{SBHE}$	78	V_{CC}	133	DLYOUT	188	RTCLKX2
24	BALE	79	V_{SS}	134	DL0	189	V_{CC}
25	DMATC	80	MA5	135	V_{CC}	190	V_{SS}
26	$\overline{DACK2}$	81	V_{CC}	136	V_{SS}	191	$\overline{LCSROM}$
27	V_{CC}	82	V_{SS}	137	DL1	192	$\overline{XMEMW}$
28	V_{SS}	83	MA6	138	XD0	193	$\overline{XMEMR}$
29	IRQ3	84	MA7	139	XD1	194	$\overline{CS8042}$
30	IRQ4	85	MA8	140	XD2	195	$\overline{XIOR}$
31	IRQ5	86	MA9	141	XD3	196	$\overline{XIOW}$
32	IRQ6	87	MA10	142	XD4	197	$\overline{RC}$
33	IRQ7	88	V_{CC}	143	XD5	198	A20G
34	$\overline{REFRESH}$	89	V_{SS}	144	XD6	199	$\overline{RESET}$
35	DRQ1	90	SA0	145	XD7	200	KEYCLK
36	SYSCLK	91	SA1	146	V_{CC}	201	KBINT
37	$\overline{DACK1}$	92	SA2	147	V_{SS}	202	PWRGOOD
38	DRQ3	93	SA3	148	MD0	203	PROCLK
39	$\overline{DACK3}$	94	SA4	149	MD1	204	IOCLK
40	$\overline{IOW}$	95	SA5	150	MD2	205	F14MHZ
41	$\overline{IOR}$	96	SA6	151	MD3	206	V_{CC}
42	$\overline{SMEMW}$	97	SA7	152	MD4	207	V_{SS}
43	$\overline{SMEMR}$	98	SA8	153	MD5	208	SPKR
44	IOCHRDY	99	SA9	154	MD6	209	P287CLK
45	AEN	100	SA10	155	MD7	210	$\overline{NPCS}$
46	$\overline{P0WS}$	101	SA11	156	MD8	211	$\overline{BUSY}$
47	DRQ2	102	SA12	157	MD9	212	PEREQ
48	IRQ9	103	SA13	158	MD10	213	$\overline{PEACK}$
49	RESETDRV	104	SA14	159	MD11	214	RESET287
50	$\overline{IOCHK}$	105	SA15	160	V_{CC}	215	$\overline{P287ERR}$
51	XA17	106	SA16	161	V_{SS}	216	HDATDIR
52	XA16	107	SA17	162	MD12		
53	XA15	108	V_{CC}	163	MD13		
54	XA14	109	V_{SS}	164	MD14		
55	XA13	110	LA17	165	MD15		

PIN DESCRIPTIONS

Clocks

PROCLK

The Processor Clock input supplies a clock to the clock divider and multiplexer logic.

IOCLK

The I/O Clock (Input) supplies a clock to the clock divider and multiplexer logic.

F14MHZ

14.318-MHz Frequency inputs to a divider that generates the clock for the internal 82C54 counter/timer.

RTCLKX1

The RTCLKX1 is the input to the RTC X1 crystal oscillator which supplies the internal real-time clock's 32.768-kHz frequency.

RTCLKX2

RTCLKX2 is the output of the RTC X2 crystal oscillator which supplies the internal real-time clock's 32.768-kHz frequency.

AT System Bus (S Bus) Interface

SYSCLK

System Clock output is the S-bus synchronous clock. It is always half the frequency of the AT state machine clock.

LA23–LA17

Extended Address input/output lines are used to address up to 16 Mb of memory on the S Bus. These signals are not latched internally with BALE and are valid earlier in the cycle than the SA lines. These pins can be driven by the internal CPU or DMA controller or by an external bus master.

SA19–SA0

S-Bus Address input/output lines are used to address memory and I/O devices on the S Bus and are valid for the entire bus cycle. These pins can be driven by the internal CPU or DMA controller or by an external bus master.

$\overline{\text{SBHE}}$

S-Bus Byte High Enable input/output signal indicates a transfer of data on the upper byte of the data bus, SD15–SD8, to a 16-bit S-bus peripheral. This signal can be driven by the internal CPU or DMA controller or by an external bus master.

SD15–SD0

S-Bus Data input/output lines are used to transfer data on the S Bus. All 8-bit devices use SD7–SD0. All 16-bit devices use SD15–SD0. Transfers may be initiated by the internal CPU or DMA controller or by an external bus master.

BALE

Buffered Address Latch Enable output is an active High output used to latch valid addresses and memory decodes during CPU transfer cycles. BALE is forced High during DMA and external bus master cycles.

$\overline{IOW}$

I/O Write is an active Low signal for I/O write cycles on the S Bus. This signal can be driven by the internal CPU or DMA controller or by an external bus master.

$\overline{IOR}$

I/O Read is an active Low control signal for I/O read cycles on the S Bus. This signal can be driven by the internal CPU or DMA controller or by an external bus master.

$\overline{MEMW}$

Memory Write is an active Low input/output control signal for all memory write cycles on the S Bus. This signal can be driven by the internal CPU or DMA controller or by an external bus master.

$\overline{MEMR}$

Memory Read is an active Low input/output control signal for all memory read cycles on the S Bus. This signal can be driven by the internal CPU or DMA controller or by an external bus master.

$\overline{SMEMW}$

S-Memory Write is an active Low output control signal for write cycles to S-bus memory with addresses less than 1 Mb.

$\overline{SMEMR}$

S-Memory Read is an active Low output control signal for read cycles to S-bus memory with addresses less than 1 Mb.

$\overline{MEMCS16}$

Memory 16-Bit Chip Select is an active Low input signal indicating that the present S-bus transfer cycle is a 16-bit memory cycle. This pin should be driven by open collector or three-state buffers. There is an internal pull-up on the pin which can be disabled.

$\overline{IOCS16}$

I/O 16-Bit Chip Select is an active Low input signal indicating that the present S-bus transfer cycle is a 16-bit I/O cycle. This pin is also sampled at reset time to determine ROM size. This pin should be driven by open collector or three-state buffers. There is an internal pull-up on the pin which can be disabled.

$\overline{P0WS}$

Zero Wait State is an active Low input signal indicating to the bus controller that it can complete the present bus cycle without inserting any additional wait states. This pin should be driven by open collector or three-state buffers. There is an internal pull-up on the pin which can be disabled.

$\overline{IOCHK}$

I/O Channel Check is an active Low input from the S Bus which can cause a NMI to be generated to the internal CPU, indicating a I/O error condition on the S Bus. This pin should be driven by open collector or three-state buffers. There is an internal pull-up on the pin which can be disabled.

IOCHRDY

I/O Channel Ready is an active High input signal from the S Bus. When Low it indicates a "not ready" condition and inserts wait states in AT I/O or memory cycles. When High it allows the current S-bus cycle to complete. This pin should be driven by open collector or three-state buffers. There is an internal pull-up on the pin which can be disabled.

AEN

Address Enable is an active High output signal used to indicate to S-bus I/O device address decoders that a DMA cycle is in progress. When active, the internal DMA controller has control of the S-bus address, data, and control pins.

$\overline{\text{MASTER}}$

$\overline{\text{MASTER}}$ is usually an active Low input signal asserted by an external device on the S Bus to allow S-bus peripherals to access system resources as a bus master. In Bus Master Mode, this pin is an output asserted when a mastering cycle is in progress. This pin should be driven by open collector or three-state buffers. There is an internal pull-up on the pin which can be disabled.

DRQ7–DRQ5, DRQ2–DRQ0

DMA Request signals 7–5 and 2–0 are active High asynchronous DMA channel request inputs used by peripheral devices to gain access to a DMA service. These pins can also be used by bus masters to gain S-bus control. DRQ2–DRQ0 perform 8-bit DMA transfers and DRQ7–DRQ5 perform 16-bit DMA transfers. There are weak pull-downs on these pins.

DRQ3

DMA Request 3 is usually defined the same as the other DRQ inputs. In Bus Master Mode, this pin is an output asserted to request bus access on a host system's bus. There is a weak pull-down on this pin.

$\overline{\text{DACK}}$7–$\overline{\text{DACK}}$5, $\overline{\text{DACK}}$2–$\overline{\text{DACK}}$0

DMA Acknowledge signals 7–5 and 2–0 are active Low output pins which acknowledge their corresponding DMA requests.

$\overline{\text{DACK}}$3

DMA Acknowledge 3 is usually defined the same as the other $\overline{\text{DACK}}$ outputs. In Bus Master Mode, this pin is an input which senses the status of a bus request to the host system.

DMATC

DMA Terminal Count is an active High output signal indicating that terminal count for a DMA channel has been reached.

$\overline{\text{REFRESH}}$

$\overline{\text{REFRESH}}$ is an active Low input/output signal to indicate a memory refresh cycle. This signal can be driven by an external bus master. There is an internal pull-up on the pin which can be disabled.

IRQ15–IRQ14, IRQ12–IRQ9, IRQ7–IRQ3

Interrupt Request input pins signal the internal 82C59 interrupt controllers that an I/O device needs attention. There are weak pull-ups on these pins.

RESETDRV

Reset Drive is an active High output signal used to reset or initialize system logic at power up time. It is synchronous to SYSCLK.

$\overline{\text{SBENA}}$

S-bus Buffer Enable is an active Low output enable for the S-bus address and data expansion buffers.

HDATDIR

High Data Direction is an active High direction control pin for the High byte of the S-bus data buffer. A logic High on this pin indicates a drive direction out from the chip.

LDATDIR

Low Data Direction is an active High direction control pin for the Low byte of the S-bus data buffer. A logic High on this pin indicates a drive direction out from the chip.

DRAM INTERFACE

$\overline{\text{RAS3}}$–$\overline{\text{RAS0}}$

Row Address Strobes are active Low outputs used by the DRAMs as $\overline{\text{RAS}}$ signals to clock in row addresses for each of the four possible DRAM banks.

$\overline{\text{CAS3}}$–$\overline{\text{CAS0}}$

Column Address Strobes are active Low outputs used by the DRAMs as $\overline{\text{CAS}}$ signals to clock in column addresses for each of the four possible DRAM banks.

MA10–MA0

Memory Address lines are multiplexed outputs and convey the following information: row addresses during $\overline{\text{RAS}}$, column addresses during $\overline{\text{CAS}}$, and refresh addresses during refresh cycles. 256-kbit DRAMs use MA8–MA0; 1-Mbit DRAMs use MA9–MA0, and 4-Mbit DRAMs use MA10–MA0.

MD15–MD0

Memory Data bus transfers data to/from the DRAMs and the math coprocessor. The High byte of the bus, MD15–MD8, is also used for transfers involving 16-bit ROM and 16-bit X-bus I/O devices.

MDP1–MDP0

Memory Data Parity bits 1 and 0 transfer parity bit information to/from the optional parity DRAM. Parity is generated and verified internally. MDP0 is the parity for the Low byte and MDP1 is the parity for the High byte.

$\overline{\text{MWENH}}$

Memory Write Enable High is an active Low output for the High byte DRAM write enable.

$\overline{\text{MWENL}}$

Memory Write Enable Low is an active Low output for the Low byte DRAM write enable.

DLYOUT

Delay Line Out is an active High output to the delay line for generating DRAM control signals.

DL1–DL0

Delay Line inputs 1 and 0 are active High inputs from two taps of a delay line to generate DRAM control signals.

X Bus Interface

XA17–XA0

X-bus Address output lines provide addressing for the BIOS ROM, keyboard controller, math coprocessor, and other X-bus peripherals.

XD7–XD0

X-bus Data lines are used to transfer the low 8 bits of data to/from X-bus devices such as the keyboard controller, BIOS ROM, and other X-bus peripherals.

$\overline{\text{XIOW}}$

X-bus I/O Write is an active Low output control signal directing an I/O port to accept data from the XD Bus.

$\overline{\text{XIOR}}$

X-bus I/O Read is an active Low output control signal directing an I/O port to place data on the XD Bus.

$\overline{\text{XMEMW}}$

X-bus Memory Write is an active Low output control signal for memory write cycles on the X Bus.

$\overline{\text{XMEMR}}$

X-bus Memory Read is an active Low output control signal for memory read cycles on the X Bus, such as BIOS ROM reads.

Coprocessor Interface

P287CLK

287 Clock provides the output clock to the numeric coprocessor (AMD 80C287 math coprocessor). The clock is generated by the clock divider and multiplexer logic, and is derived from PROCLK or IOCLK.

$\overline{\text{NPCS}}$

Numeric Processor Chip Select is an active Low output which indicates a data transfer involving the math coprocessor.

$\overline{\text{BUSY}}$

Busy is an active Low input from the math coprocessor indicating it is executing an instruction. There is a weak internal pull-up resistor on this pin.

PEREQ

Processor Extension Request is an active High input pin which indicates that the math coprocessor is ready to transfer data to/from the CPU. There is a weak pull-down on this pin.

$\overline{\text{PEACK}}$

Processor Extension Acknowledge is an active Low output which indicates that the requested transfer due to an active PEREQ has started.

RESET287

287 RESET is an active High output signal which resets the math coprocessor.

$\overline{\text{P287ERR}}$

287 Error is an active Low input which indicates a math coprocessor error condition and can trigger the coprocessor interrupt to the interrupt controller. There is a weak pull-up on this pin.

Keyboard Controller Interface

$\overline{\text{CS8042}}$

8042 Chip Select is an active Low signal used for selecting the external keyboard controller. If the XT keyboard interface is enabled, the pin definition changes to XTKBDATA, a bi-directional, serial data transfer line with an internal pull-up.

$\overline{\text{RC}}$

Reset CPU is an active Low input signal which will activate the internal CPU's reset when active. This is generated from the 8042 keyboard controller.

A20G

Address 20 is the Gate input for CPU address line 20. A logic High on this input will enable A20 pass-through from the CPU. A logic Low will force the internal A20 inactive. This signal is generated by the 8042 keyboard controller.

$\overline{\text{RESET}}$

$\overline{\text{RESET}}$ output is an active Low, CPU clock synchronized reset signal derived from PWRGOOD. It is usually used to reset the keyboard controller. If the XT keyboard interface is enabled, the pin definition changes to XTKBRST, an I/O controlled reset output for a XT keyboard.

KEYCLK

Keyboard Clock provides the output clock to the keyboard controller. The clock is generated by the clock divider and multiplexer logic, and can be derived from several clock sources. If the XT keyboard interface is enabled, the pin definition changes to XTKBCLK, an I/O controlled clock line with an internal pull-up.

KBINT

Keyboard Interrupt is a direct connection to the IRQ1 input on the master 82C59A interrupt controller in AT keyboard mode (default).

Miscellaneous Signals

$\overline{\text{LCSROM}}$

ROM Chip Select is an active Low output which provides the chip select for the BIOS ROM/EPROM.

PWRGOOD

Power Good is an active High input signal which indicates a stable system power condition. This is normally driven by the power supply. A logic Low on this input resets the device. There is a Schmitt trigger input on this pin.

SPKR

Speaker output signal is the output of the internal tone generation logic, which includes one channel of the internal 82C54 timer. This is a standard TTL level output.

V_{BATT}

Negative Battery Voltage input to power the internal Real Time Clock and CMOS memory.

V_{CC}

Power.

GND

Ground.

CHAPTER 2

80C286 AND STANDARD PERIPHERALS

80C286 MICROPROCESSOR CORE

The Am286ZX/LX integrated processor uses AMD's CMOS 80C286 microprocessor as the core of the internal system. The 80C286 core maintains the complete functional features of a standard 80C286. This includes the full 80286 instruction set, registers, status information, internal memory organization, 16 Mb of physical and 1 Gb of virtual address space, all addressing modes, address and data segmentation, pipelining schemes, real and protected mode environments with four levels of memory protection, interrupt priorities, and halt/shutdown cycles. See the 80C286 data sheet (order #11625) for more information.

In addition, the core is fully static. This means that the Am286ZX/LX integrated processor can operate anywhere from its maximum speed down to 0 MHz. Power consumption is significantly reduced by lowering the clock speed. For maximum power saving, the clock may be shut off completely. The Am286ZX/LX integrated processor retains its state while the clock is stopped and then continues to operate from its original state when the clock is resumed.

DMA CONTROLLERS

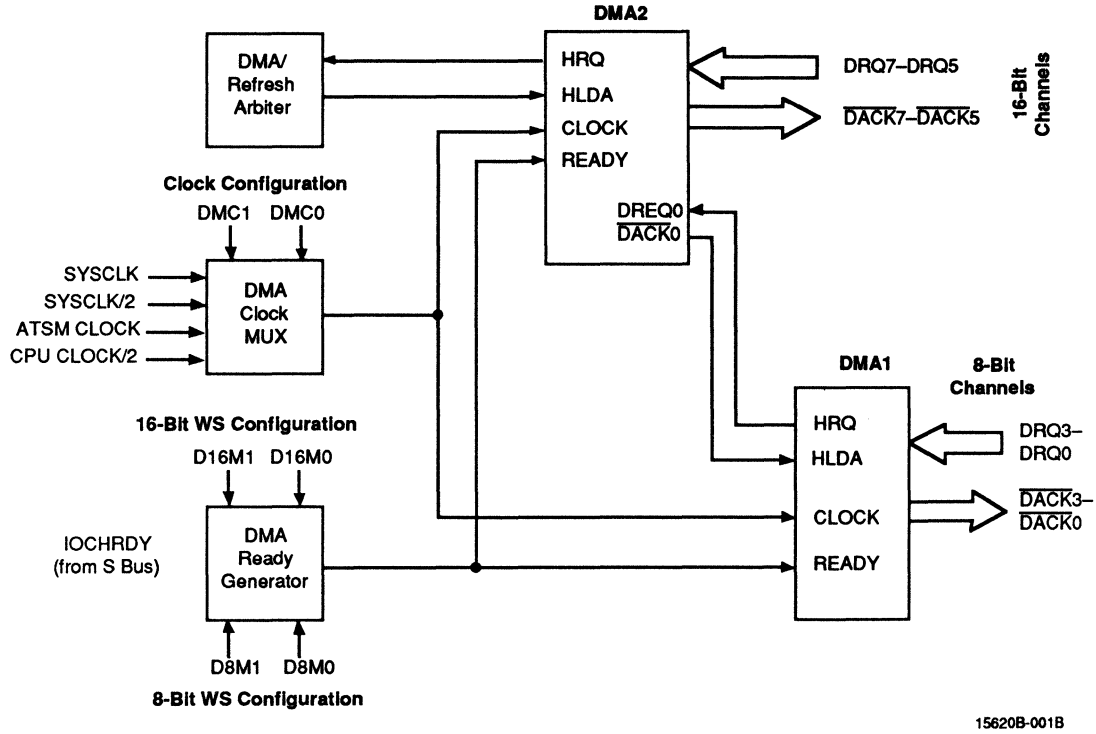
Two identical, 9517/8237-compatible DMA controllers and a page register are integrated into the Am286ZX/LX integrated processor. Figure 2-1 shows a block diagram illustrating DMA controller interconnection, clocking, and control. The two DMA controllers maintain the complete functional features of a standard 9517/8237 device. DMA1 occupies addresses 000H–00FH and DMA2 occupies addresses 0C0H–0DFH. Each controller is a four channel DMA device which generates the memory addresses and control signals necessary to directly transfer information between a peripheral device and memory, with little CPU intervention. The two DMA controllers are internally cascaded to provide four DMA channels for 8-bit transfers (DMA1) and three DMA channels for 16-bit transfers (DMA2). DMA2 Channel 0 is internally cascaded.

Programmable registers in the Am286ZX/LX integrated processor allow independent control for both 8-bit and 16-bit transfers by inserting wait states. Four different clock sources can also be selected to provide the clocks for the DMA controllers.

Each of the seven available DMA channels has a pair of 16-bit counters and a pair of reload registers for each counter. This allows up to 64-kb block transfers with DMA1 and up to 128-kb block transfers with DMA2. Several modes of operation are possible using programmable features in the registers. See the 9517 Technical Manual for detailed information on the programming and operation of the DMA controllers.

The page registers occupy addresses 080H–08FH and are used to generate the High order address during DMA cycles. Only eight of these registers are used, but all 16 are included to maintain PC-AT compatibility. Each DMA channel has an associated page register, with the exception of Channel 0 of DMA2. One of the page registers, located at address 08FH, is used to provide the High order address for refresh cycles. The Am286ZX/LX integrated processor includes weak pull-downs on the DREQ input pins. (See Figure 2-1.)

Figure 2-1 DMA Controller Block Diagram



Clock Control

The DMA clock inputs on DMA1 and DMA2 are driven by a clock multiplexer and synchronizer. Switching between different clocks is controlled by the synchronizer so that no glitches are presented to the DMA controllers. There are four possible sources for the DMA clocks, with only one, the default SYSCLK/2, being AT compatible. The other selections are provided for embedded and bus master system environments. The common clock to both DMA1 and DMA2 is selectable via two configuration bits, DMC1 and DMC0, in the Peripheral Selection Register (PSR, index 03H).

DMA CLOCK SELECTION

PSR.1 DMC1	PSR.0 DMC0	Selection
0	0	SYSCLK/2
0	1	SYSCLK
1	0	AT State Machine Clock
1	1	CPU Clock/2

The SYSCLK/2 selection is the AT-compatible setting where the DMA controllers are clocked at half of the system bus speed.

The other clock selections run the DMA controllers at a rate either the same as SYSCLK, the same as the AT state machine (twice SYSCLK), or at CPU state clock frequency (CPU Clock/2).

DMA cycle time specifications and DRAM controller DMA-mode access times may limit the clocking method used by the DMA controllers in some system configurations.

Cycle Control

The duration of DMA cycles can be controlled by inserting wait states during DMA transfer cycles. Wait states are pre-programmed by a configuration register entry, but DMA cycles may also be extended by a DMA device by deasserting IOCHRDY during the cycle.

Programmed DMA wait states are individually selectable for both 8- and 16-bit cycles. 8-bit wait states are selectable via bits D8M1 and D8M0 in the Peripheral Selection Register (PSR, index 03H). 16-bit wait states are selectable via bits D16M1 and D16M0, also in PSR.

DMA WAIT STATE SELECTION

PSR.5 D8M1	PSR.4 D8M0	8-Bit Wait States
0	0	1
0	1	2
1	0	3
1	1	4

PSR.7 D16M1	PSR.6 D16M0	16-Bit Wait States
0	0	1
0	1	2
1	0	3
1	1	4

The AT-compatible setting of one wait state is the default setting. The others may be used when faster DMA clocking options are being utilized.

X-Bus Redirection

The four 8-bit DMA channels (0–3) are capable of being redirected such that the DMA device can be located on the X Bus. The requests and acknowledges will still use the DRQ/$\overline{\text{DACK}}$ lines, but data transfers will occur between the DMA device on the XD Bus and either the MD Bus, or SD Bus, depending on the memory location being accessed.

There are four configuration bits in the X-bus DMA Channel Enable Register (XBD, index 21H), each of which map one of the 8-bit DMA channels onto the X Bus. The bits are XBD0, XBD1, XBD2, and XBD3, and they enable X-bus DMA for DMA channels 0 through 3, respectively. If these bits are programmed to a 1, the corresponding DMA channel will be mapped to the X Bus. The default values for these configuration bits are 0, which map all four of the 8-bit DMA channels to the S Bus.

X-BUS DMA SELECTION

Name	Position	Channel
XBD3	XBD.3	DMA1 Channel 3
XBD2	XBD.2	DMA1 Channel 2
XBD1	XBD.1	DMA1 Channel 1
XBD0	XBD.0	DMA1 Channel 0

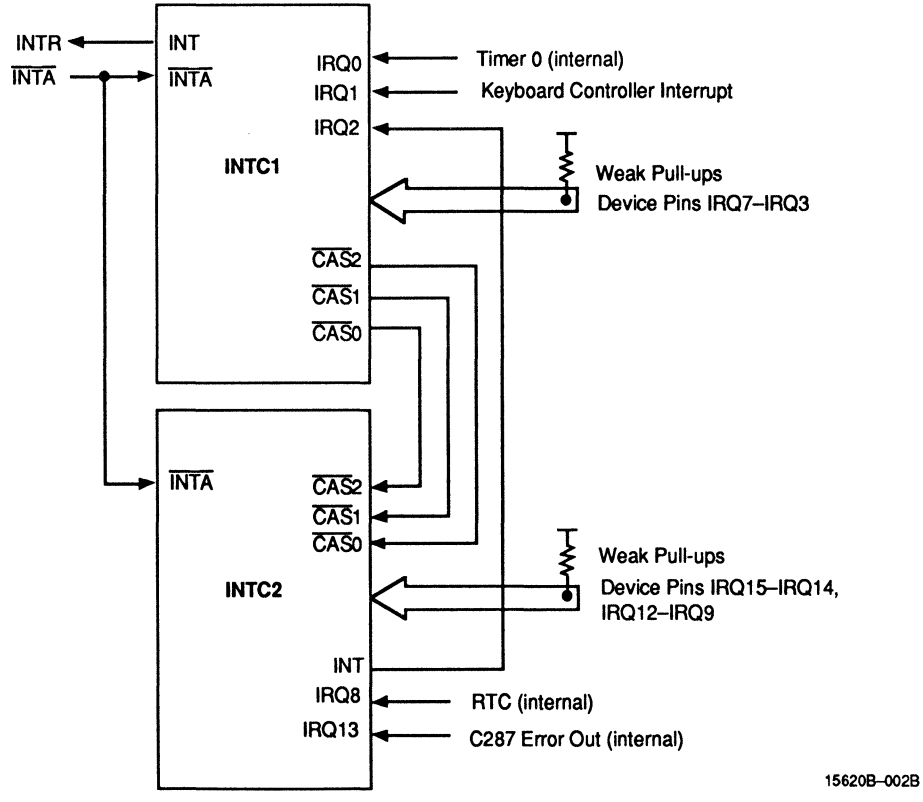
INTERRUPT CONTROLLERS

Two identical, 8259-compatible interrupt controllers, INTC1 and INTC2, are integrated into the Am286ZX/LX integrated processor, as shown in Figure 2-2. They accept requests from peripherals, resolve priority on pending interrupts and interrupts in service, issue an interrupt request to the processor, and provide interrupt vectors for interrupt service routines.

The two devices are internally connected and must be programmed to operate in cascade mode for operation of all 15 interrupt channels. INTC1 occupies addresses 020H–021H, and is configured for master operation in cascade mode. INTC2 occupies addresses 0A0H–0A1H and is configured for slave operation. The interrupt request output signal from INTC2 (INT) is internally connected to the interrupt request input Channel 2 (IRQ2) of INTC1. This configuration is compatible to the IBM PC-AT.

The output of Timer 0 in the counter/timer section is connected to Channel 0 (IRQ0) of INTC1. The interrupt request from the Real Time Clock is connected to Channel 0 (IRQ8) of INTC2. The keyboard interrupt is connected to Channel 1 (IRQ1) of INTC1. Also, the coprocessor exception interrupt is connected to Channel 5 (IRQ13) of INTC2. The other interrupts are also available to external peripherals, as in the AT architecture. The Am286ZX/LX integrated processor includes weak pull-ups on the interrupt input pins.

Figure 2-2 Interrupt Controller Block Diagram



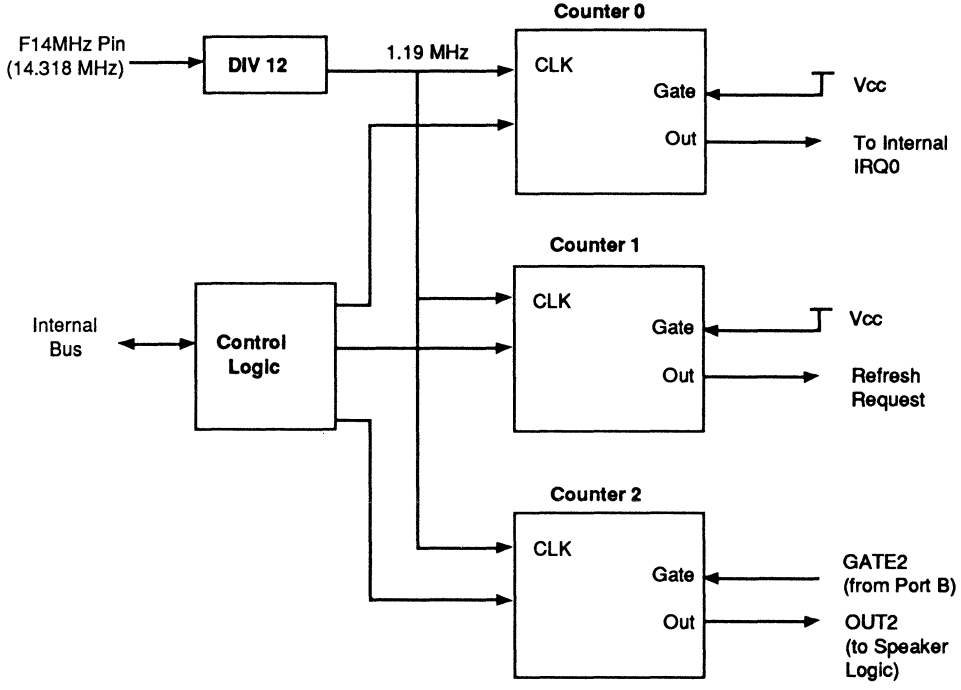
COUNTER/TIMER

A three-channel, general purpose, 8254-compatible, 16-bit counter/timer is integrated into the Am286ZX/LX integrated processor, as shown in Figure 2-3. It provides critical timing parameters for the PC system under software control. It can be programmed to count in binary or in BCD. Each counter operates independently of the other two and can be programmed for operation as a timer or a counter. All three are controlled from a common set of control logic, which provides controls to load, read, configure, and control each counter. The counter occupies I/O addresses 040H–043H. There are six modes of operation.

Mode 0	Interrupt on terminal count
Mode 1	Hardware re-triggerable one-shot
Mode 2	Rate Generator
Mode 3	Square Wave Generator
Mode 4	Software triggered strobe
Mode 5	Hardware re-triggered strobe

All three counters are driven from a common clock which is internally generated as a divide by 12 of the F14MHz pin. The output of Counter 0 is connected to IRQ0 interrupt input of INTC1, and may be used as a software "timer tick" for time-keeping and task-switching activities. The output of Counter 1 is internally connected as a refresh request. The output of Counter 2 is internally connected to speaker logic.

Figure 2-3 Counter/Timer Block Diagram



15620B–003B

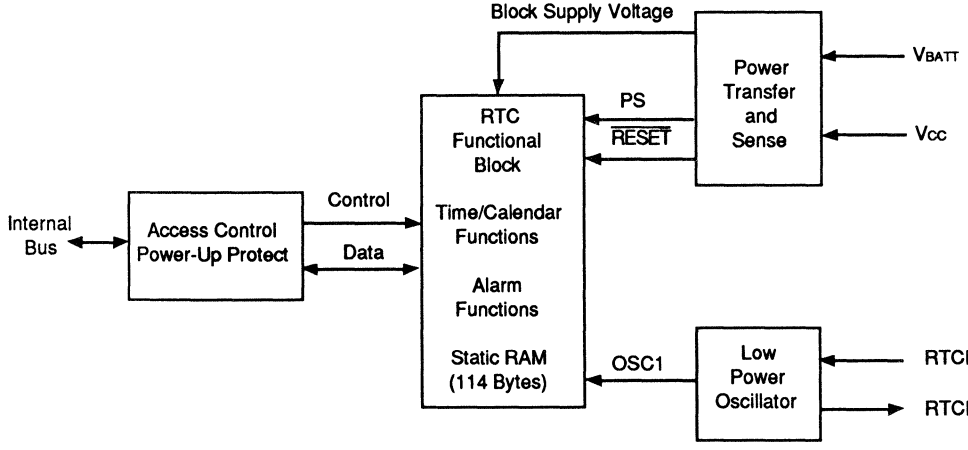
REAL TIME CLOCK and CMOS STATIC RAM

A Real Time Clock (RTC) function is implemented in the Am286ZX/LX integrated processor, as shown in Figure 2-4. The RTC's address and data registers are located at I/O addresses 070H and 071H. It combines a complete time-of-day clock with alarm, a 100-year calendar, a programmable periodic interrupt, and 114 bytes of CMOS static RAM. The static RAM is battery backed to save its contents in the absence of main system power. Also, since it is battery backed, clock counting continues to maintain the date and time when power is shut off.

The Am286ZX/LX integrated processor includes a low-power CMOS crystal oscillator circuit to handle the 32.768-kHz clock signal to the RTC. This feature saves board space by integrating the oscillator on-chip, and it extends the life of the system's battery. An indexed addressing scheme is implemented to write to the 128 locations of the RTC. The index register is written with the address of the memory location, which acts as an address pointer. Data is then transferred to/from the location. The 128 addressable locations in the RTC contain ten locations for time, calendar and alarm data, four general purpose registers, and 114 static RAM locations.

The alarm bytes can be programmed to generate an interrupt at a specific time, or they can be programmed to generate a periodic interrupt. The static RAM, from index addresses 0EH to 7FH, is not affected by the RTC. This area can be used to store configuration and calibration information.

Figure 2-4 RTC Block Diagram



15620B–004B

RTC Address Map

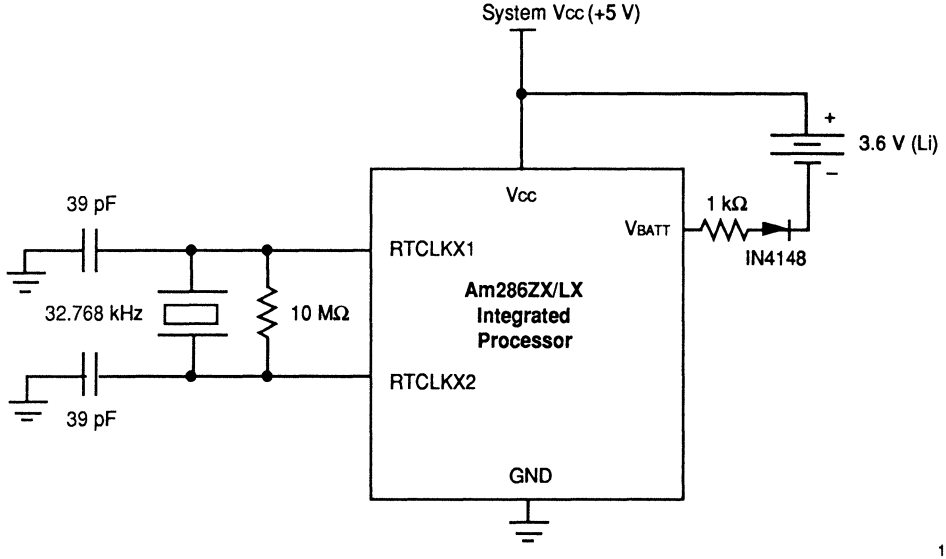
Index	Contents
0	Seconds
1	Seconds Alarm
2	Minutes
3	Minutes Alarm
4	Hours
5	Hours Alarm
6	Day of Week
7	Date of Month
8	Month
9	Year
10	Register A
11	Register B
12	Register C
13	Register D
14–127	User Static RAM

External Connections

The recommended external connections for the battery circuit and the oscillator circuit are shown in Figure 2-5.

The RTC oscillator circuit includes the parallel resonant 32.768-kHz crystal, along with three other passive components: a resistor and two capacitors. PC board trace length should be minimized, and unnecessary pin-to-pin capacitance between RTCLKX1 and RTCLKX2 should be avoided.

Figure 2-5 RTC External Connection Example



15620B-005B

The battery back-up circuit for the RTC is quite different than other implementations. With the Am286ZX/LX integrated processor, the positive side of the battery connects to the system V_{CC}, and the negative side connects through switching diodes to the V_{BATT} pin. At least one switching diode is required. For use of other battery voltages, the following constraint for the voltage at the V_{BATT} pin can be used to calculate the number of diodes required.

$$V_{BATT(MIN)} < V_{BATTERY} - (V_D)N < V_{CC(MIN)}$$

Where $V_{BATT(MIN)}$ = Data Sheet V_{BATT} Spec
$V_{BATTERY}$ = Battery Voltage
V_D = Diode Voltage Drop
N = Number of Diodes
$V_{CC(MIN)}$ = Minimum System Supply

Power loss due to a bad or disconnected battery will be reflected in the VRT (Valid RAM & Time) bit in Register D.

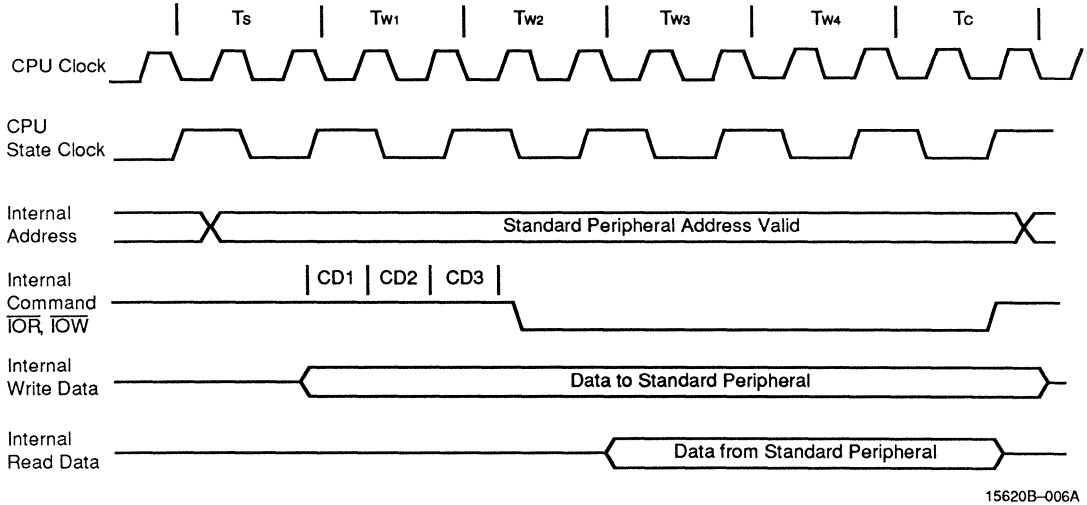
STANDARD PERIPHERAL I/O CYCLE CONTROL

Description of Cycles

Data transfers between the 80C286 CPU and the standard peripherals (DMA, Interrupt, Counter/Timer, and RTC) are internal cycles and normally will not be echoed outside of the device (the exception is FASTBUS mode). It is important to note that these data transfers are also synchronous to the CPU, using the high-speed state machine. See Figure 2-6 for a diagram of a typical internal transfer cycle to one of the standard peripherals.

The internal transfer cycle looks like a standard demultiplexed bus cycle which, in Figure 2-6, is programmed for three command delays and four wait states. The internal address bus goes valid in the CPU TS state and is decoded for a given peripheral. The internal I/O

Figure 2-6 Internal Transfer Cycle (3 Command Delays, 4 Wait States Shown)



15620B–006A

Read or I/O Write command strobe goes active after three command delays, which are counted in terms of CPU input clock cycles. The command strobe goes inactive after the specified four wait states (labeled TW1 through TW4), plus the final command state (TC). The data is transferred to the internal peripheral during the command active time, as shown.

Programmable Options

The wait states and command delays invoked during an internal transfer cycle to one of the standard peripherals are programmable via configuration register bits. Each standard peripheral group has a set of associated wait states and command delays which can be specified.

The default cycle options for all the peripherals are four wait states and zero command delays.

The recommended setting for correct device operation across all CPU speeds is four wait states and three command delays.

The command delays and wait states for the standard peripherals are configured in two wait state registers, IPWS1 (index 06H) and IPWS2 (index 07H), and in one command delay register, IPCD (index 08H).

DMA1, DMA2, AND PAGE REGISTER

Name	Bits	Setting
IDWS3–IDWS0	IPWS1.3–IPWS1.0	4-bit wait states, 0–15
IDCD1–IDCD0	IPCD.1–IPCD.0	2-bit command delays, 0–3

INTC1 AND INTC2

Name	Bits	Setting
IIWS3–IIWS0	IPWS1.7–IPWS1.4	4-bit wait states, 0–15
IICD1–IICD0	IPCD.3–IPCD.2	2-bit command delays, 0–3

TMR (COUNTER/TIMER)

Name	Bits	Setting
ICWS3–ICWS0	IPWS2.7–IPWS2.4	4-bit wait states, 0–15
ICCD1–ICCD0	IPCD.5–IPCD.4	2-bit command delays, 0–3

RTC/MISC (INCLUDES PORT B, PORT 92, NMI CONTROL)

Name	Bits	Setting
IMWS3–IMWS0	IPWS2.3–IPWS2.0	4-bit wait states, 0–15
IMCD1–IMCD0	IPCD.7–IPCD.6	2-bit command delays, 0–3

The 4-bit configuration fields for wait state settings are a straight binary encoding, with 0000 indicating zero wait states and 1111 indicating 15 wait states. Similarly, the 2-bit configuration fields for command delay settings are a straight binary encoding, with 00 indicating zero command delays and 11 indicating three command delays.

CHAPTER 3

SYSTEM CONTROL LOGIC

CONFIGURATION REGISTERS

The configuration registers of the Am286ZX/LX integrated processor provide a uniform method of accessing the device's control and configuration parameters. These parameters are mapped into configuration bits and bit fields, and they are contained in logical groupings of 8-bit registers. The combination of the software programmable system configuration and the extended static RAM in the RTC for the storage of that configuration data, allows maximum system flexibility without a proliferation of switches and jumpers on the system board.

Most of the configuration registers have full read/write capability. Any configuration bit that can be written can also be read. Full read/write capability allows for read-modify-write operations for setting and clearing individual bits in the register, without effecting the other bits. Only a few of the configuration bits are read only. They are provided for the purpose of sensing status, instead of specifying configuration. Configuration bits are read/write if they are not specified as read only.

All of the configuration registers are set to a default value by the PWRGOOD pin being inactive (a system reset condition). The default values have been specified to allow the device to correctly execute code out of the X-bus ROM/EPROM upon exiting the reset state.

Access Method

Access to the configuration registers is through a two port I/O register interface circuit. There is a write-only index register at I/O address 022H, and there is a read/write data register at I/O address 023H. The contents of the index register select which configuration register will be accessed when a data register I/O cycle occurs. There are no special recovery time requirements for the index and data registers; a data register read/write can immediately follow an index register write.

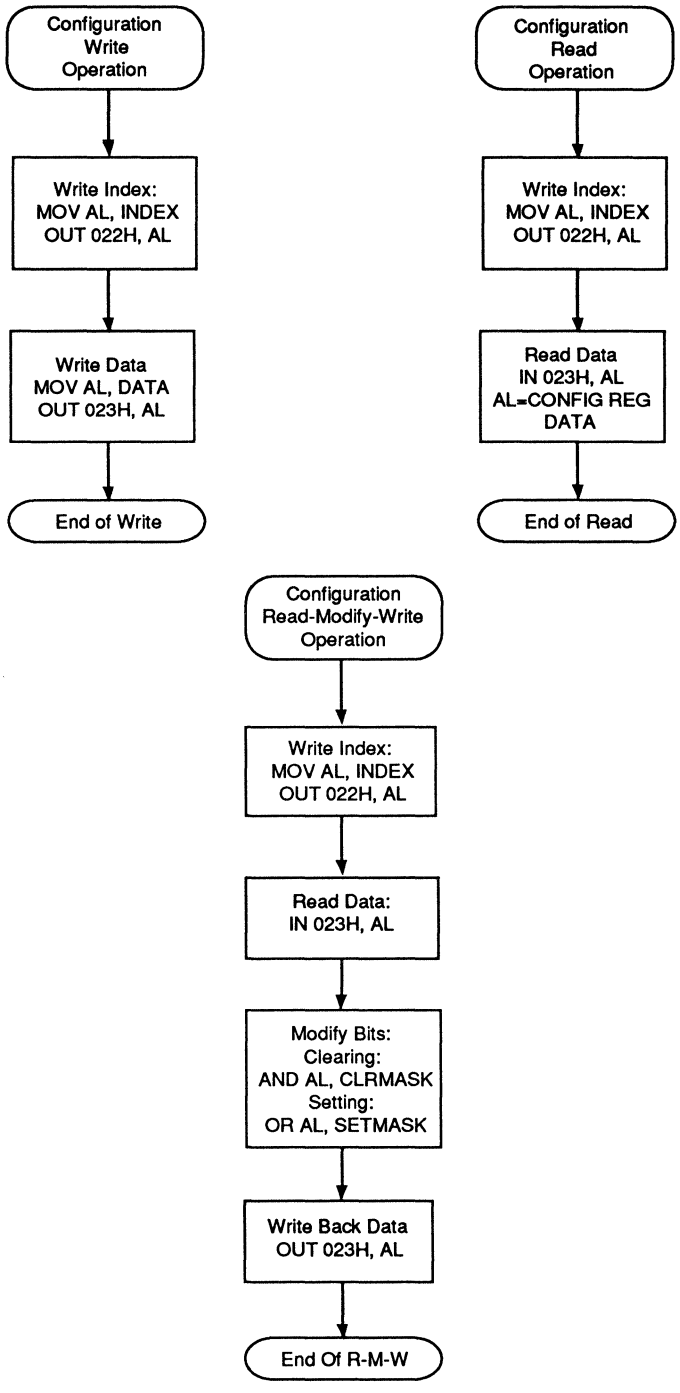
A flow chart with assembly code for reading, writing, and modifying configuration register contents is shown in Figure 3-1. The basic access routine consists of writing out the desired register's index into I/O location 022H, and then either writing or reading the actual configuration register's contents through the data register at I/O location 023H.

Note that the contents of the index register is static relative to multiple data register reads and writes. This allows the read-modify-write operation shown in Figure 3-1. The index is written, and then the data is read, modified, and written back into the register, without rewriting the index.

In order to keep the device configuration secure during normal system operation, the suggested value at which to leave the index register is 0. The register at index 0 is a read-only identification register, and therefore an erroneous write to I/O port 023H will not affect the device configuration.

Because the index register is write-only, interrupt routines which must access a configuration register cannot properly restore the index. Therefore, interrupts should be

Figure 3-1 Configuration Register Access Flow



15620B-007B

disabled for configuration register accesses that could be interrupted by another routine that also accesses configuration registers.

The bus transfer cycles for index and data register accesses are CPU synchronous cycles. Data register reads and writes are internal cycles and normally will not be echoed outside of the device (the exception is FASTBUS mode, as described in the Clock Generator section). The wait states and command delays invoked during a transfer cycle to a configuration register vary according to which register is being accessed.

Configuration Register Accesses

Register	Cycle Parameters
Index (address 022H)	4 wait states, 2 command delays
CCR1 (index 01H)	1 wait state, 0 command delay
PSFR (index 23H)	9 wait states, 0 command delay
All Other Registers	0 wait state, 0 command delay

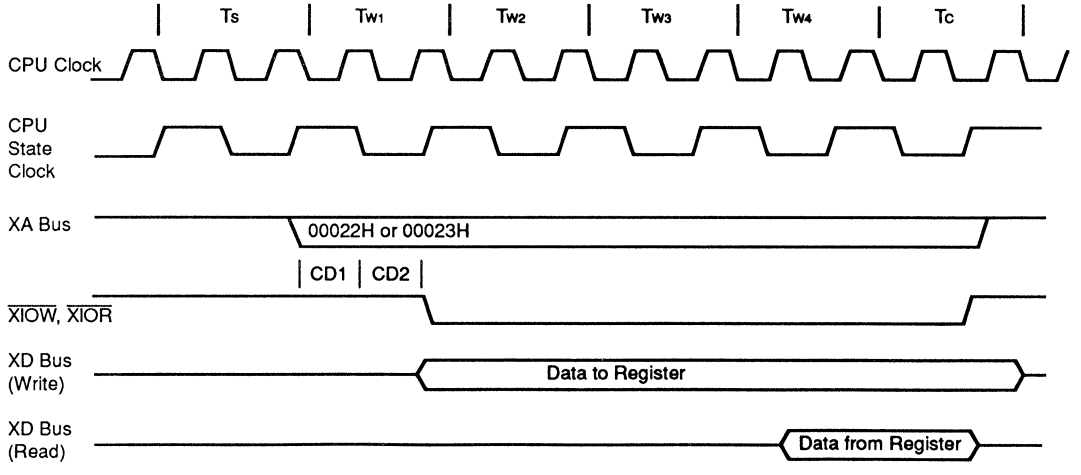
Accesses to CCR1 default to the one wait state, zero command delay specified above, but the cycle may be extended by the clock synchronization logic, in order to complete the clock source change before the end of the cycle.

Off-Chip Expansion

The indexed addressing scheme used in the Am286ZX/LX integrated processor allows for up to 255 configuration registers. However, less than 64 are used. To provide the capability for a system with extended configuration registers, configuration register reads and writes with an index greater than or equal to 64 decimal places will be directed out onto the X Bus. This allows for off-chip system functions to be configured identically to the internal system functions, via the index/data access method.

External configuration register accesses on the X Bus are four wait states and two command delays, CPU synchronous cycles. All index register writes come out on the X Bus, to allow an external index value to be maintained. See Figure 3-2 for an example of the external configuration cycle.

Figure 3-2 External Configuration Register Access Cycle (4 Wait States, 2 Command Delays)



15620B–008B

ENHANCED CLOCK GENERATOR

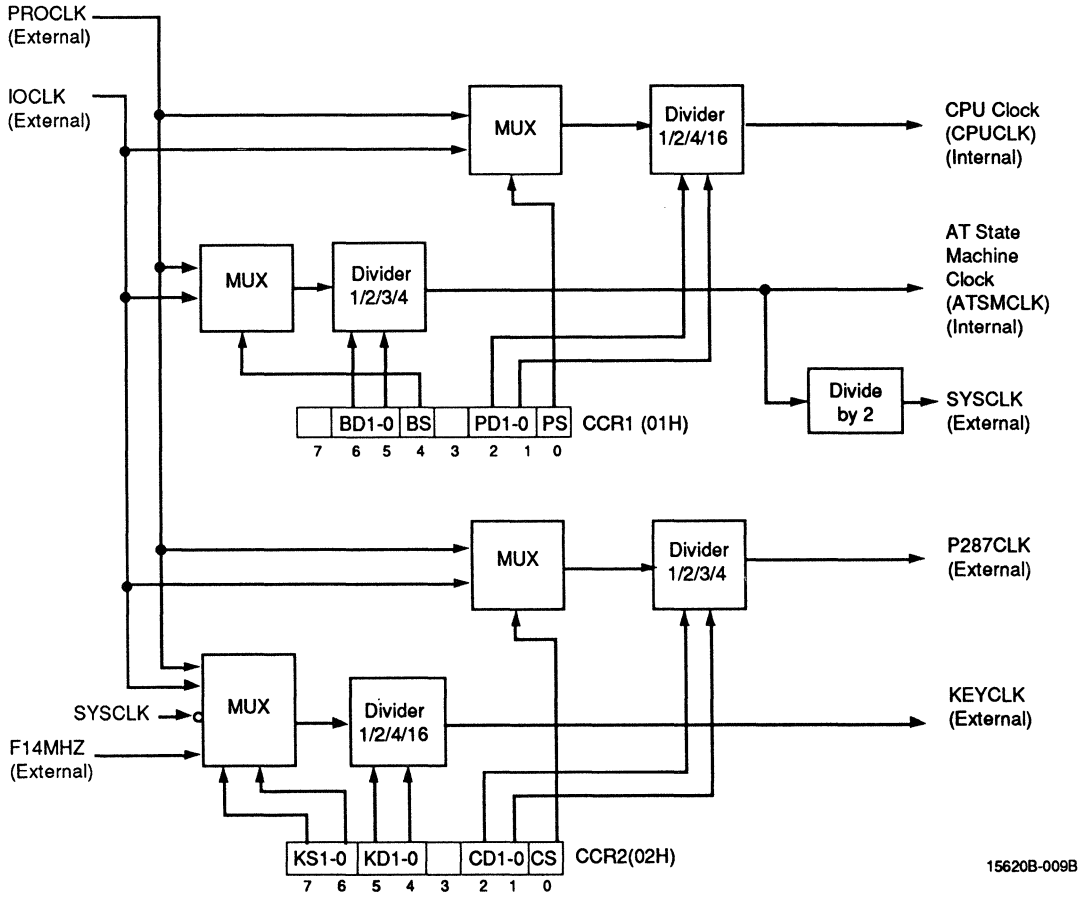
The clock generator provides all of the major clocks in an Am286ZX/LX integrated processor-based system. There are two main clock inputs, the PROCLK and IOCLK pins. They are actually generic clock inputs, the "PRO" (processor) and "IO" (I/O) prefixes are just for common usage. There is an additional input clock pin, F14MHZ, which is used for internal peripheral timing and as an alternate source for the keyboard clock output (KEYCLK).

The outputs of this functional unit provide the clocks for 80C286 CPU (CPUCLK), the AT State Machine (ATSMCLK), the AMD 80C287 math coprocessor (P287CLK), and the AT keyboard controller clock (KEYCLK).

The clock generator performs three main functions for each output clock: source multiplexing, clock division, and synchronization. The basic structure of the generator is shown in Figure 3-3.

The clock sources enter a multiplexer, with the selection for the source coming from a configuration register. The multiplexer output is fed to a programmable divider, with the

Figure 3-3 Clock Generator Block Diagram



divisor selection also coming from a configuration register. There is synchronization logic throughout the clock generator to oversee the clock switching process, in order to guarantee that no glitches will be present on any of the output clocks.

Figure 3-4 shows a generic clock switch operation with the switch occurring between two asynchronous clock sources. Note that the output clock is disabled at a safe time relative to the original clock source and that it is enabled at a safe time relative to the new clock source, guaranteeing no glitches on the clock output.

The CPUCLK and KEYCLK divider selections include a high-divide ratio (divide by 16). This selection is provided for the internal CMOS CPU and for a possible external CMOS keyboard controller; in the case where a low-power consumption, slow clock polling situation is required.

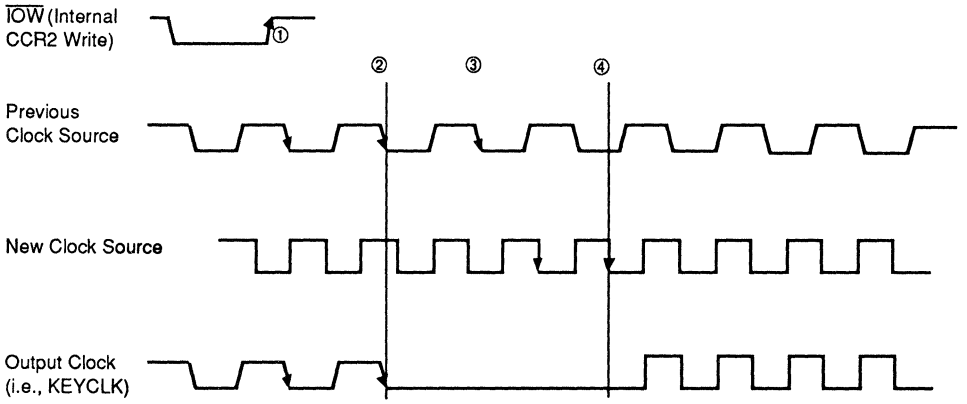
Driving both the clock input pins, PROCLK and IOCLK, is not a necessity. There are several possible single input clocking configurations, depending on the CPU and S-bus clock rates required. In the case of a single clock being supplied, the other clock pin should be hardwired to either supply rail, V_{CC} or GND. A series resistor to the supply rail is not required.

CPU Clock

The main destination of the CPU Clock (CPUCLK) is the internal 80C286 processor. However, this clock also goes to the internal CPU state machine, the bus controller's high-speed state machine (HSSM), and to the DRAM controller's page mode state machine.

The input source for CPUCLK can either be PROCLK or IOCLK. The selection is based on the configuration bit PS in Clock Configuration Register 1 (CCR1, index 01H). The default value selects PROCLK as the source.

Figure 3-4 Generic Clock Switch Operation



Sequence:

① I/O Write command to CCR2 initiates clock switch.
② After 2 falling edges of previous clock, output clock is safely inhibited (no glitching).
③ After 3rd falling edge of previous clock, internal MUXes switch to new source.
④ After 2 falling edges of new clock, output clock is safely re-enabled at new frequency (no glitching).

15620B-010B

CPU Clock Source Selection

CCR1.0 PS	CPUCLK Source
0	PROCLK Input Pin
1	IOCLK Input Pin

The programmable divider for CPUCLK can be set to divide by 1, 2, 4, or 16. The divisor is set by the configuration bits PD1 and PD0, also in CCR1. The default value selects a divisor of 4. Therefore, the CPU's clock at power-up or hard reset is a divide by four of PROCLK.

CPU Clock Divisor Selection

CCR1.2 PD1	CCR1.1 PD0	CPUCLK Divisor
0	0	Divide by 4
0	1	Divide by 1
1	0	Divide by 2
1	1	Divide by 16

AT State Machine Clock

The main destination of the AT State Machine Clock (ATSMCLK) is the AT State Machine in the bus controller. However, this clock also provides the clocking for the refresh generator, and it is an input to the DMA clock multiplexer.

The SYSCLK pin on the S Bus is always a divide by 2 of ATSMCLK. If ATSMCLK is the same as CPUCLK, then the phase of SYSCLK is the same as the CPU state clock.

The input source for ATSMCLK can either be PROCLK or IOCLK. The selection is based on the configuration bit BS in Clock Configuration Register 1 (CCR1, index 01H). The default value selects PROCLK as the source.

ATSM Clock Source Selection

CCR1.4 BS	ATSMCLK Source
0	PROCLK Input Pin
1	IOCLK Input Pin

The programmable divider for ATSMCLK can be set to divide by 1, 2, 3, or 4. The divisor is set by the configuration bits BD1 and BD0, also in CCR1. The default value selects a divisor of 4. The divide ratio of three can be useful in some single clock configurations to arrive at an S-bus speed between 6 and 8 MHz.

ATSM Clock Divisor Selection

CCR1.6 BD1	CCR1.5 BD0	ATSMCLK Divisor
0	0	Divide by 4
0	1	Divide by 1
1	0	Divide by 2
1	1	Divide by 3

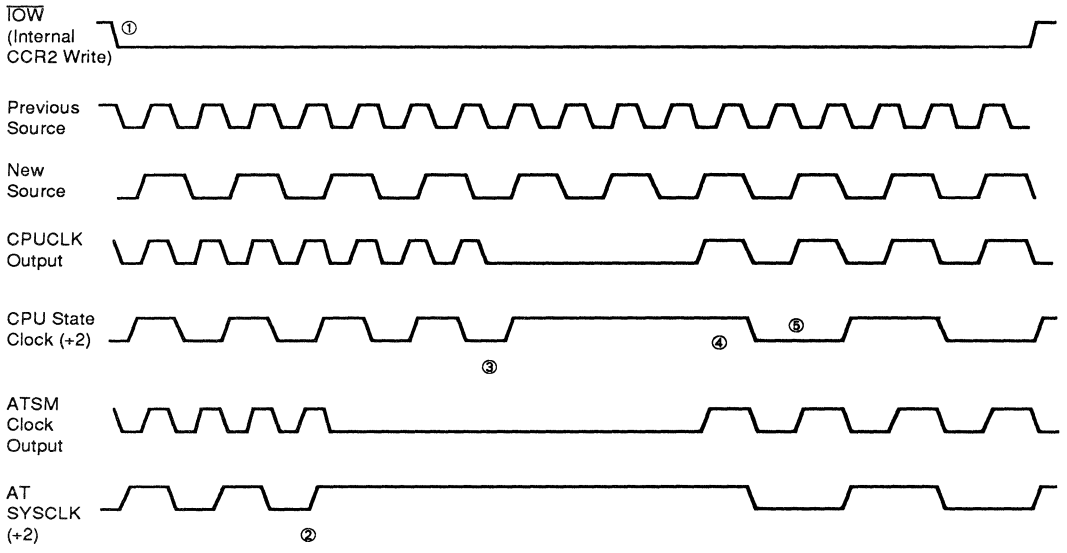
CPU and AT State Machine Clock Interaction

In order to maintain the proper phase relationship between the CPU clock and the ATSM clock, especially when they are synchronous to each other, there is a special dual shutdown clock switching sequence. As shown in Figure 3-5, a clock source or divider change for either the CPU or ATSM clock will invoke a special clock switch sequence which will shut down both of the clocks and then bring them up with the new setting, with the proper phase relationship between the two clocks, if they are synchronous.

Keyboard Clock

The destination for the keyboard clock is the KEYCLK output pin. This pin only carries the clock generator output when the keyboard interface is in AT keyboard mode. In XT keyboard mode, the definition of the KEYCLK pin changes, and the clock generator output is internally disconnected.

Figure 3-5 Dual Shutdown CLock Switch Example



Sequence:

① I/O Write to CCR1 indicates change from CPUCLK and ATSM clock sourced by PROCLK to new setting of both sourced by PROCLK/2. Note that the internal cycle doesnot complete ($\overline{IOW}\uparrow$) until end of switch.

② AT State Machine clock stopped with SYSCLK High.

③ After Safe ATSM clock stop, CPU clock stop initialized, ending with CPUCLK inactive with CPU state clock High.

④ Both CPUCLK and ATSM clock restart with new source, simultaneously because CCR1 new clock selection indicates synchronous clocks.

⑤ Note that CPU state clock and AT SYSCLK maintained proper phase relationship.

15620B-011B

The input source for KEYCLK can be PROCLK, IOCLK, inverted SYSCLK, or F14MHZ. The inverted SYSCLK selection allows for generation of a two-phase clock, consisting of SYSCLK and KEYCLK, for possible direct connection to a CMOS keyboard controller. The F14MHZ selection provides a fixed frequency, clock configuration independent source for the keyboard clock. The selection is based on the configuration bits KS1 and KS0 in Clock Configuration Register 2 (CCR2, index 02H). The default value selects F14MHZ as the source.

Keyboard Clock Source Selection

CCR2.7 KS1	CCR2.6 KS0	KEYCLK Source
0	0	PROCLK Input Pin
0	1	IOCLK Input Pin
1	0	Inverted SYSCLK
1	1	F14MHZ Input Pin

The programmable divider for KEYCLK can be set to divide by 1, 2, 4, or 16. The divisor is set by the configuration bits KD1 and KD0, also in CCR2. The default value selects a divisor of four. Therefore, the CPU's clock at power-up or hard reset is a divide by four of F14MHZ.

Keyboard Clock Divisor Selection

CCR2.5 KD1	CCR2.4 KD0	KEYCLK Divisor
0	0	Divide by 4
0	1	Divide by 1
1	0	Divide by 2
1	1	Divide by 16

Coprocessor Clock

The destination for the coprocessor clock is the P287CLK output pin. This pin always carries the clock generator output. There are no other internal connections for this clock.

It is not a requirement for the P287CLK output to be used for the actual math coprocessor's clock. An independent clock source may be used for the math coprocessor. The P287CLK output may then be used as a general purpose programmable clock output, or it may not be used at all.

The input source for P287CLK can either be PROCLK or IOCLK. The selection is based on the configuration bit CS in Clock Configuration Register 2 (CCR2, index 02H). The default value selects PROCLK as the source.

Coprocessor Clock Source Selection

CCR2.0 CS	P287CLK Source
0	PROCLK Input Pin
1	IOCLK Input Pin

The programmable divider for P287CLK can be set to divide by 1, 2, 3, or 4. The divisor is set by the configuration bits command delay 1 and command delay 0, also in CCR2. The default value selects a divisor of four. Therefore, the coprocessor's clock at power-up or hard reset is a divide by four of PROCLK.

Coprocessor Clock Divisor Selection

CCR2.2 CD1	CCR2.1 CD0	P287CLK Divisor
0	0	Divide by 4
0	1	Divide by 1
1	0	Divide by 2
1	1	Divide by 3

The divide by three setting for the divisor will result in an output waveform that is a logic High for one input clock and a logic Low for two input clocks (a 33% duty cycle, with some variation according to propagation delays in the divider and drive buffer).

Clocking Mode Indication

The values in Clock Configuration Register 1 (CCR1, index 01H) select the sources and divisors for both the CPU and AT State Machine (ATSM) clocks. The selection indicated for these two clocks defines the clocking mode. The operating characteristics of the ATSM are determined by the current clocking mode. Specifically, the way the ATSM initiates and terminates cycles relative to the CPU and its clock is different with each clocking mode.

The four clocking modes are called SYNCSAME, SYNCDIFF, ASYNC, and FASTBUS. The effects of the different modes are described in the Enhanced Bus Controller section. The clock selections which indicate the clocking mode are shown in the table below.

Clocking Mode Selection

Mode	CPU Clock	ATSM Clock
SYNCSAME	PROCLK/1	PROCLK/1
	IOCLK/1	IOCLK/1
	PROCLK/2	PROCLK/2
	IOCLK/2	IOCLK/2
	PROCLK/4	PROCLK/4
	IOCLK/4	IOCLK/4
FASTBUS	SYNCSAME with FBE=1	
SYNCDIFF	PROCLK/1	PROCLK/2
	IOCLK/1	IOCLK/2
	PROCLK/2	PROCLK/4
	IOCLK/2	IOCLK/4
ASYNC	All other combinations	

ENHANCED BUS CONTROLLER

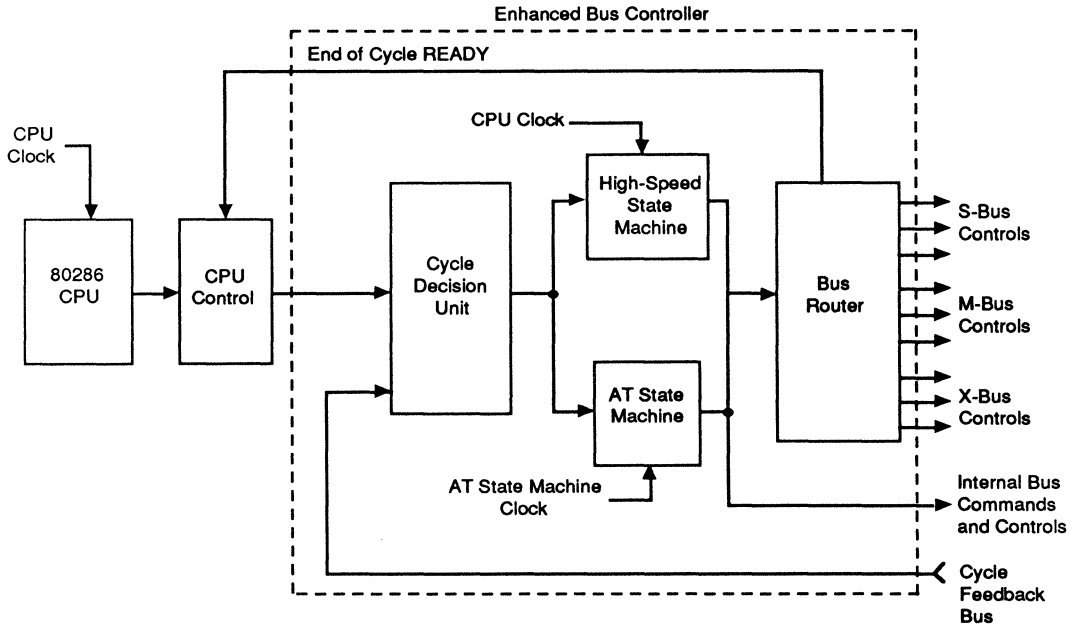
Features

The enhanced bus controller provides the control logic necessary for all data transfers initiated by any of the possible bus masters: the 80C286 CPU, either of the internal DMA controllers, the refresh generator, or an external bus master. It generates the command strobes and buffer control signals with the proper timing characteristics for the target device. The controller supervises cycles to both internal and external devices, determining their bus width and routing data when bus size mismatches are detected. Figure 3-6 shows a block diagram of the bus control logic.

The bus controller features fully programmable wait state and command delay generation capabilities. Wait states extend the command pulse width by delaying the completion of the cycle by a number of cycle clocks. The cycle clock frequency is half that of the state machine input clock. Command delays postpone the active (Low going) edge of the command pulse, allowing more address setup time to the command active edge. Command delays delay the command by a number of state machine clocks. See Figure 3-7A for a basic cycle diagram.

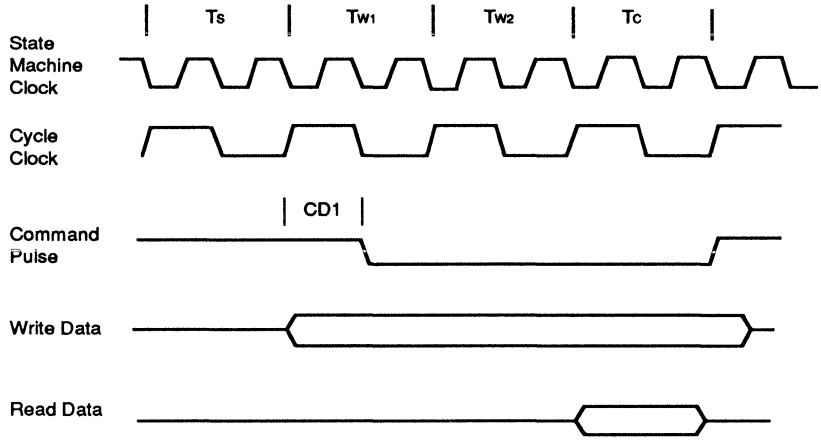
The bus controller in an AT-compatible system must solve a very basic problem: transparently connecting a 16-bit processor to 8-bit peripheral devices. The bus controller does this by providing conversion and data routing functions when bus size incompatibilities exist. A conversion cycle is initiated when a 16-bit read or write is attempted with an 8-bit device as the target of the transfer. See Figure 3-7B for a conversion cycle diagram. The bus controller will split the cycle into two 8-bit cycles,

Figure 3-6 Bus Controller Block Diagram

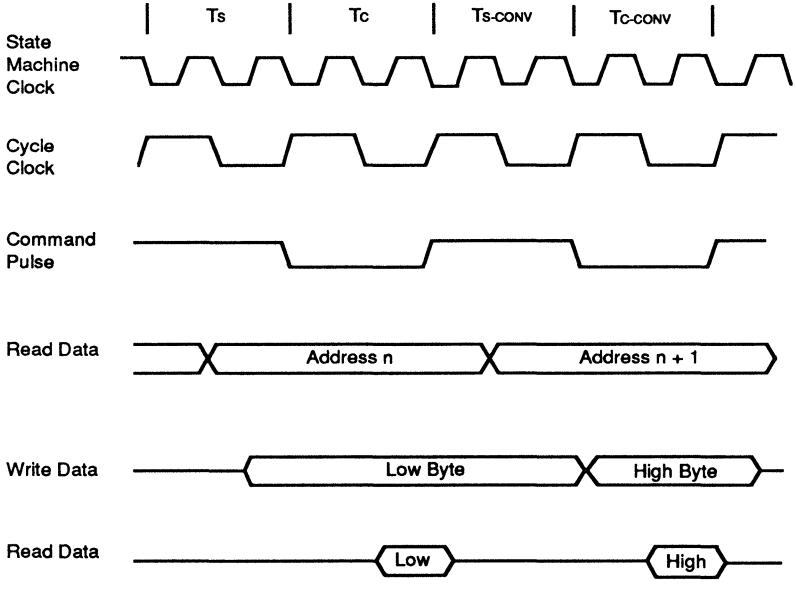


15620B-012B

Figure 3-7 Command Timing and Conversion Cycle



a. Basic Command Timing (2 Wait States, 1 Command Delay Cycle)



b. Basic Conversion Cycle (16 Bit to 8 Bit)

15620B-013B

toggling the address and routing the High byte data during the second access. Routing is performed when an 8-bit odd (High byte) transfer is attempted with an 8-bit target device. The bus controller routes the CPU's High byte to or from the Low byte of the internal data bus, depending upon whether the cycle is a read or a write.

BUS CONTROLLER ARCHITECTURE

The bus controller has a dual-state machine architecture to optimize the speed of data transfer with the target device. In high-speed ISA bus systems, there needs to be a high-performance access path to the main system memory, while at the same time the system must provide standard ISA bus cycles with a compatible bus speed, typically in the 6- to 8-MHz range. The controller solves this speed differential by providing two separate bus state machines that generate cycle timing, along with a decision unit to determine which state machine to activate on a per cycle basis. The two state machines are the High-Speed State Machine (HSSM), which handles the fast, CPU synchronous cycles, and the ATSM, which handles the ISA type cycles, possibly operating asynchronous to the CPU.

CYCLE DECISION UNIT

The decision unit for the two state machines operates from a cycle feedback bus, which provides cycle type information for the current transfer. The cycle type information includes wait states, command delays, bus size, and an indicator for HSSM or ATSM operation. Individual peripherals on the internal bus provide information on the feedback bus in response to seeing their respective address and transfer type on the internal bus. Also, the bus router can provide information on the external X-bus peripherals. If no feedback is received by the decision unit, it defaults to activating the ATSM and instructing the bus router to direct the cycle onto the S Bus (the AT system bus).

HIGH-SPEED STATE MACHINE

The High Speed State Machine (HSSM) services all the CPU cycles needing high-speed, CPU synchronous data transfers. The HSSM directly tracks the CPU state machine, feeding back a fast synchronous ready to the CPU, allowing zero wait state performance at the maximum rated device speed. The main transfer handled by the HSSM is CPU to local DRAM. The HSSM, along with the integrated DRAM controller, forms a high-performance data and control path for CPU to local memory transfers (instruction fetches and data reads/writes). The HSSM also services the X-bus ROM/EPROMs, the numeric coprocessor, and all internal peripherals (DMAs, interrupts, etc.).

AT STATE MACHINE

The ATSM provides cycle control for cycles that require ISA timing and function compatibility. The ATSM may or may not be running synchronous to the CPU, depending on the clocking mode defined in Clock Configuration Register 1 (CCR1, index 01H). The ATSM detects the clocking mode and adjusts its CPU synchronization method according to the relationship between the CPU clock and the ATSM's clock. The main transfer handled by the ATSM is S-bus cycles. The timing and functionality of ISA cycles can be maintained at the S-bus interface regardless of the CPU speed. The other cycles serviced by the ATSM are X-bus keyboard controller cycles, X-bus programmable I/O range cycles, and internal address manager register accesses.

ROUTING CONTROL

The bus router generates the enable and direction controls for the internal S-bus, M-bus, and X-bus address and data buffers, along with the external S-bus buffer controls. The bus router generates the proper buffer enables for all cycles generated by the bus masters (CPU, DMA, refresh, and external bus masters), routing data through the device to effect the indicated transfer.

Bus Control Configuration

S-BUS CONNECTION

The reset, or default, condition of the S-bus interface is a disabled, or "disconnected" state. The bus controller will generate the internal control signals for an S-bus transfer, but the routing controller will not turn on the buffer controls. Therefore, the transfer cycle will not appear externally. Any data read will have no meaning. Connection of the S Bus is controlled by the BCON bit in the Bus Control Register (BCR, index 09H). If this bit is programmed to a 1, the S Bus is connected and the buffer controls will be enabled in order to allow S-bus targeted cycles to appear on the S-bus interface.

S-Bus Connection Control

BCR.0 BCON	Function
0	S Bus Disconnected
1	S Bus Connected

The connection control is required to support the bus master mode of the Am286ZX/LX integrated processor. The device cannot power up with the S-bus interface connected because, in a bus master mode application, the device should not be driving interface pins that the host processor is driving, including the strong pull-ups on some of the inputs.

The intended programming of the BCON bit in a typical ISA system environment is to program the BCON bit to a 1 as soon as possible after a hardware/power-up reset.

FULL I/O DECODE

For full AT compatibility, all I/O decodes for internal and external peripherals controlled by the Am286ZX/LX integrated processor are done with 10-bit address decoding. That is, only address bits 9–0 are used to generate chip selects for the peripherals. I/O address bits 10 through 15 are ignored.

For applications where AT style I/O decode compatibility is not needed, and where an expanded I/O decode space is desired, there is a configuration option to allow full 16-bit I/O address decoding. With this full I/O decode option enabled, peripheral chip selects generated by the device also depend upon address bits 15–10 being in a Low, or 0, state. This opens up a large I/O space from 0100H through FFFFH for external S-bus peripherals, without 10-bit "aliasing" holes appearing in the I/O space.

This full I/O decode option is controlled by the configuration bit FIO in the Bus Control Register (BCR, index 09H). If this bit is set to a 1, full 16-bit I/O decode is enabled. If this bit is set to a 0, normal AT style 10-bit decoding is used. The default value for this bit selects 10-bit decoding.

I/O Decoding Selection

BCR.3 FIO	Decode Method
0	10-bit AT Decoding
1	16-bit Full I/O Decoding

BUS RESET CONTROL

The S-bus reset pin, RESETDRV, will be driven active during the hardware reset of the device. A logic Low on the PWRGOOD input pin will activate RESETDRV. However, because of the capability of the bus controller to connect and disconnect the S-bus interface, a separate, parallel path is provided to generate an active RESETDRV.

The configuration bit SRST in the Bus Control Register (BCR, index 09H) provides the ability to activate and deactivate the RESETDRV pin. Programming the SRST bit to a 1 will force the RESETDRV pin active (High). Setting the SRST bit to a 0 will force the pin inactive (Low), as long as PWRGOOD is active. The default value for this bit indicates that RESETDRVshould not be driven active except for a hardware reset via PWRGOOD.

S-Bus RESETDRV Control

BCR.2 SRST	Function
0	Normal RESETDRV
1	Force RESETDRV Active

PROGRAMMABLE X-BUS RECOVERY TIME

Some devices on the X Bus could require a long recovery time (which is the time between consecutive cycles on the X Bus). One typical requirement for extended recovery time is for devices which have a long read command to data three-state time. In this case, an X-bus write immediately after an X-bus read could cause data bus contention between the peripheral releasing the bus, and the Am286ZX/LX integrated processor starting to drive the data bus for the write. Another typical requirement for extended recovery time is for devices that can not process another data transfer for a specified time after the end of the previous cycle.

The bus controller can be configured to either provide, or not provide, extra recovery time for both X-bus ROM/EPROM and X-bus I/O devices. The extra recovery time is implemented as a delay of one CPU clock cycle after the read or write inactive command signal edge, but before a "ready" is generated for the CPU, which allows the next cycle to start.

X-bus ROM/EPROM recovery time is controlled by the configuration bit RCROM in the Memory Wait States Register (MWS, index 12H). The RCROM bit is an active High disable bit, which means that when RCROM is programmed to a 1, the extra ROM/EPROM recovery time is disabled. When the bit is programmed to a 0, which is the default condition, the extra recovery time is enabled.

ROM Recovery Time Control

MWS.6 RCROM	ROM Recovery Time
0	Enabled
1	Disabled

X-bus I/O recovery time is controlled by the configuration bit RCXIO in the Memory Wait States Register (MWS, index 12H). The RCXIO bit is an active High disable bit, which means that when RCXIO is programmed to a 1, the extra X-bus I/O device recovery time is disabled. When the bit is programmed to a 0, which is the default condition, the extra recovery time is enabled.

X-Bus I/O Recovery Time Control

MWS.7 RCXIO	X-Bus I/O Recovery Time
0	Enabled
1	Disabled

SPECIFYING THE CLOCKING MODE

As stated previously in the Enhanced Clock Controller section, the clocking mode is defined by the values set in Clock Configuration Register 1 (CCR1, index 01H). The relationship between the CPU clock and the ATSM clock selections in CCR1 determines how the ATSM initiates and terminates cycles.

High Speed State Machine Cycles

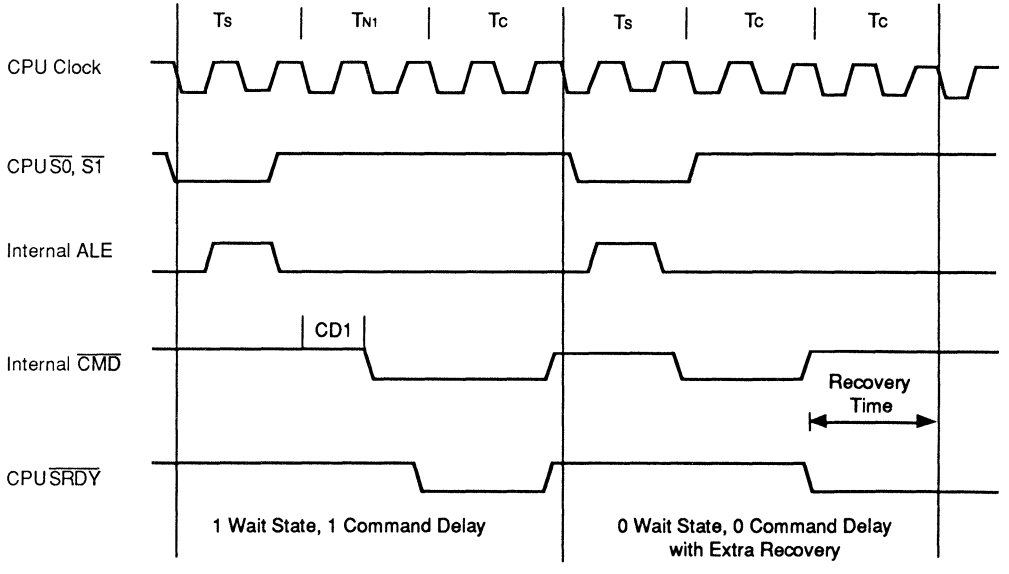
HSSM transfer cycles are very straightforward because they are always synchronous to the 80C286 CPU. Figure 3-8 shows the simple relationship between the CPU states and the command and control signals generated by the HSSM. The figure also shows the effect of extra recovery time being enabled in the second cycle. The primary outputs of the HSSM are the command pulse (CMD) and the synchronous ready (SRDY), which feeds back to the CPU control logic.

AT State Machine Cycles

CYCLE CONTROL

The ATSM generates ISA-compatible command and control signals whose timing varies according to the type of transfer being processed and the bus size of the target peripheral for that transfer. For each combination of transfer type (memory or I/O)

Figure 3-8 High Speed State Machine Transfer Cycles



15620B-014B

and bus size (8 or 16 bit), there is a programmable default cycle. The key parameters for the default cycle which can be programmed are the number of wait states and the number of command delays. However, the timing of the default cycle can be modified by feedback signals on the S Bus ($\overline{\text{P0WS}}$ and IOCHRDY).

The default number of wait states for each of the four cycle combinations are specified in the Wait State Register (WSR, index 05H). The 16-bit memory cycle wait states are specified in configuration bits M16W1 and M16W0. The 16-bit I/O cycle wait states are specified in bits I16W1 and I16W0. The 8-bit memory cycle wait states are specified in bits M8W1 and M8W0. The 8-bit I/O cycle wait states are specified in bits I8W1 and I8W0.

The default value for the configuration bits is all ones, which indicates three wait states for 16-bit cycles and five wait states for 8-bit cycles.

16-Bit Memory and I/O Wait State Control

WSR.7/3 M16W1 I16W1	WSR.6/2 M16W0 I16W0	Wait States
0	0	0
0	1	1
1	0	2
1	1	3

8-Bit Memory and I/O Wait State Control

WSR.5/1 M8W1 I8W1	WSR.4/0 M8W0 I8W0	Wait States
0	0	2
0	1	3
1	0	4
1	1	5

The default number of command delays for each of the four cycle combinations are specified in the Command Delay Register (CDR, index 04H). The 16-bit memory cycle command delays are specified in configuration bits M16C1 and M16C0. The 16-bit I/O cycle command delays are specified in bits I16C1 and I16C0. The 8-bit memory cycle command delays are specified in bits M8C1 and M8C0. The 8-bit I/O cycle command delays are specified in bits I8C1 and I8C0.

The default value for the configuration bits is all ones, which indicates three command delays for both 8- and 16-bit cycles.

16-Bit Memory and I/O Command Delay Control

CDR.7/3 M16C1 I16C1	CDR.6/2 M16C0 I16C0	Command Delays
0	0	0
0	1	1
1	0	2
1	1	3

8-Bit Memory and I/O Command Delay Control

CDR.5/1 M8C1 I8C1	CDR.4/0 M8C0 I8C0	Command Delays
0	0	0
0	1	1
1	0	2
1	1	3

ATSM cycles generated according to the values indicated in the configuration registers (WSR and CDR) can be modified by feedback pins on the S Bus. The cycles can either be terminated earlier than the programmed number of wait states, or the cycles can be extended indefinitely, adding to the number of wait states. The $\overline{\text{P0WS}}$ pin can be used to shorten, or cause early termination of, default cycles. The IOCHRDY pin can be used to lengthen, or add wait states to, default cycles. See the AT System Bus chapter for more information on the operation of these two pins.

SYNCSAME CYCLES

The SYNCSAME clocking mode is defined as both the CPU clock and ATSM clock having the same clock source pin and divisor, but with the FBE configuration bit in CCR1 (index 01H) set to 0. The effect of this clocking mode upon ATSM operation is that the state machine is running synchronous to the CPU, but only S-bus transfer cycles appear on the S Bus. See FASTBUS mode for the alternate case where FBE is set to 1. Figure 3-9 shows the synchronous relationship between the CPU states and the command and control signals generated by the ATSM. It also shows a 16- to 8-bit conversion cycle being executed.

In SYNCSAME mode, the ATSM status cycle (AT-TS) does not start until the end of the CPU status cycle, when the decision logic determines that the current cycle is an ATSM cycle. In the AT-TS cycle, the BALE signal is generated, and then the programmed command pulse is generated according to the specified number of command delays and wait states.

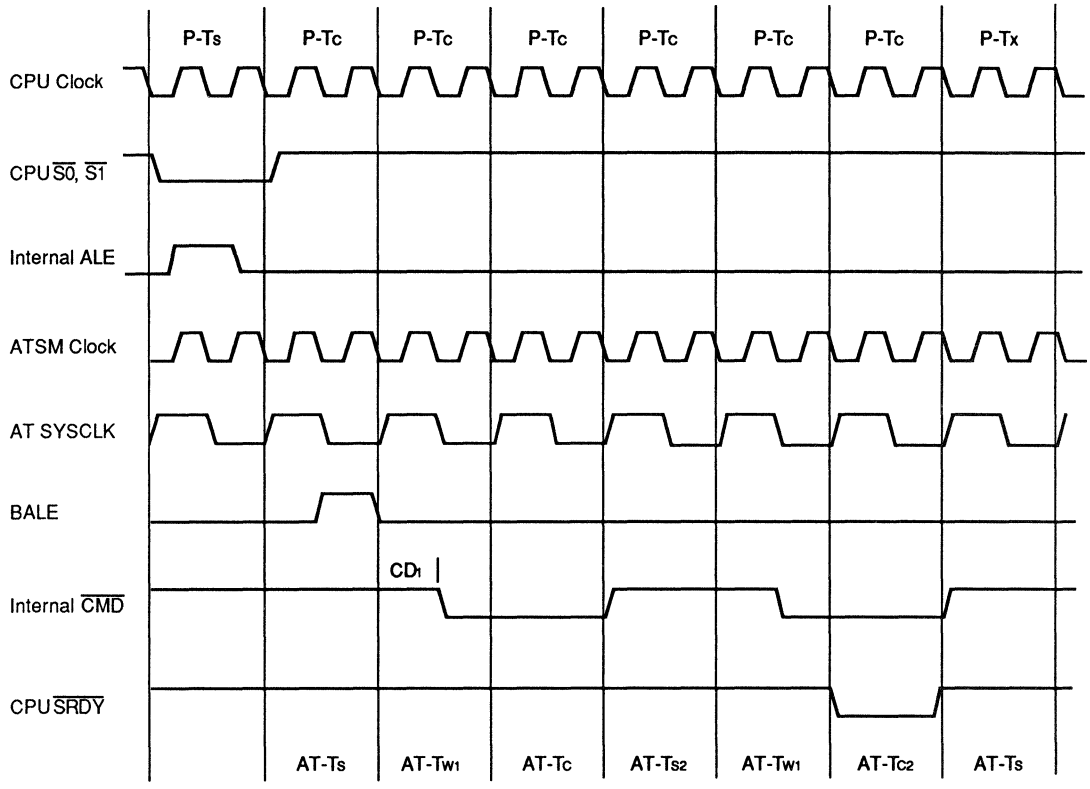
A SYNCSAME cycle ends with a synchronous ready (SRDY) being fed back to the CPU control logic, with no added synchronization states needed.

SYNCSAME mode has the benefit of clocking the CPU at a rate equal to that of the AT system bus (S Bus). This provides code execution rates equal to the original AT system design, where the CPU and system bus used the same clock.

SYNCDIFF CYCLES

The SYNCDIFF clocking mode is defined as CPU clock and ATSM clock having the same clock source pin, but with the divisors selected such that the CPU clock frequency is twice that of the ATSM clock. Even though the clocks to the CPU and ATSM are "different," the state machine takes advantage of the relationship between the two clocks in order to optimize the cycle synchronizations. There is special phase alignment logic which will start the ATSM status cycle as soon as possible, aligning the cycle to SYSCLK automatically. Figure 3-10 shows the synchronous but different relationship between CPU states and ATSM states, and control signals.

Figure 3-9 SYNCSAME Mode AT State Machine Cycle



AT State Machine SYNCSAME (one wait state, one command conversion cycle)

P-Ts = Processor State Cycle
P-Tc = Processor Command Cycle
AT-Ts = ATSM Status Cycle with BALE
AT-Tw = ATSM Wait State
AT-Tc = ATSM Final Command Cycle

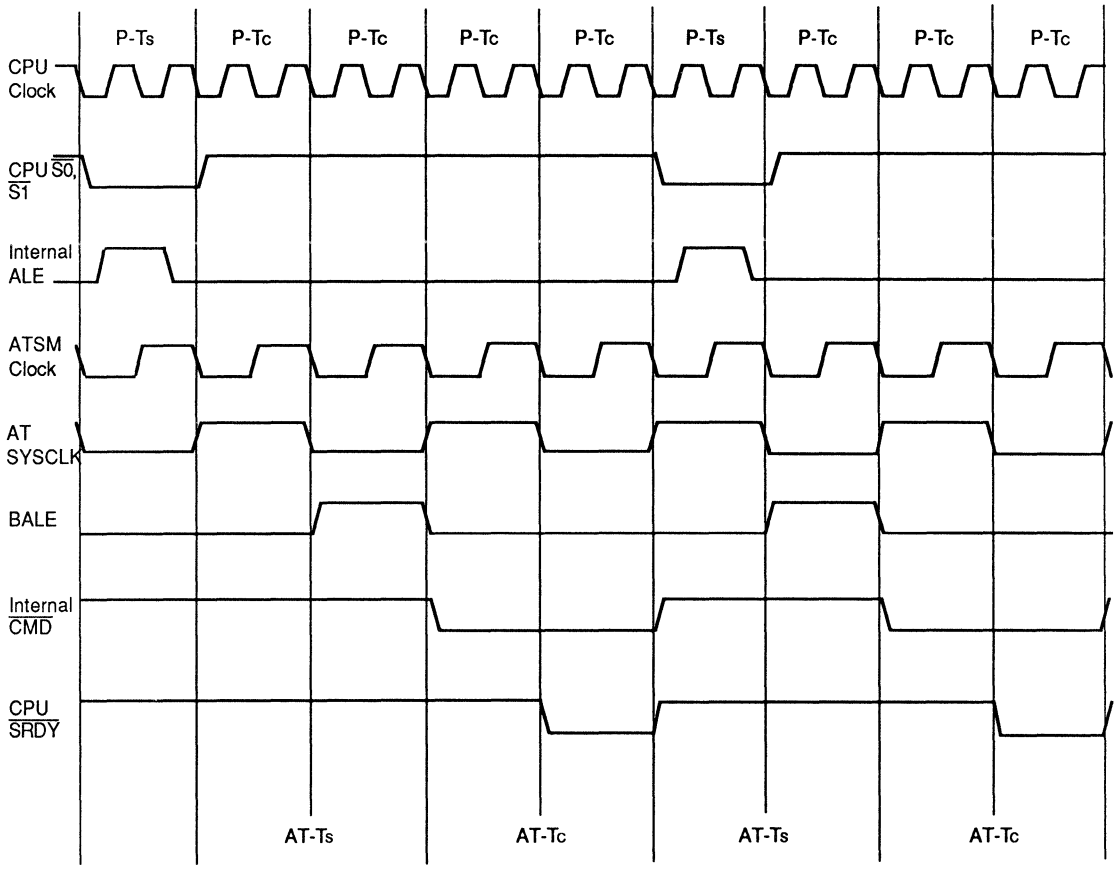
15620B-015B

In SYNCDIFF mode, the ATSM status cycle (AT-TS) will start at either the end of the CPU status cycle or one ATSM clock later, depending on the phase of SYSCLK. See Figure 3-10 for examples of both types of cycle starts. Note that in the second transfer BALE is immediately issued, starting the cycle earlier than it could be in a pure asynchronous mode.

The end of the cycle also takes advantage of the synchronous relationship of the two clocks, terminating the cycle without any synchronization overhead between the ATSM and the CPU.

The benefit of SYNCDIFF mode is that the CPU can run at a different, faster clock rate than the ATSM without incurring the extra synchronization delays normally associated with disparate clocking.

Figure 3-10 SYNCDIFF Mode AT State Machine Cycles



AT State Machine SYNCDIFF (0 wait states, 0 command transfer cycle)

P-Ts = Processor State Cycle
P-Tc = Processor Command Cycle
AT-Ts = ATSM Status Cycle with BALE
AT-Tc = ATSM Final Command Cycle

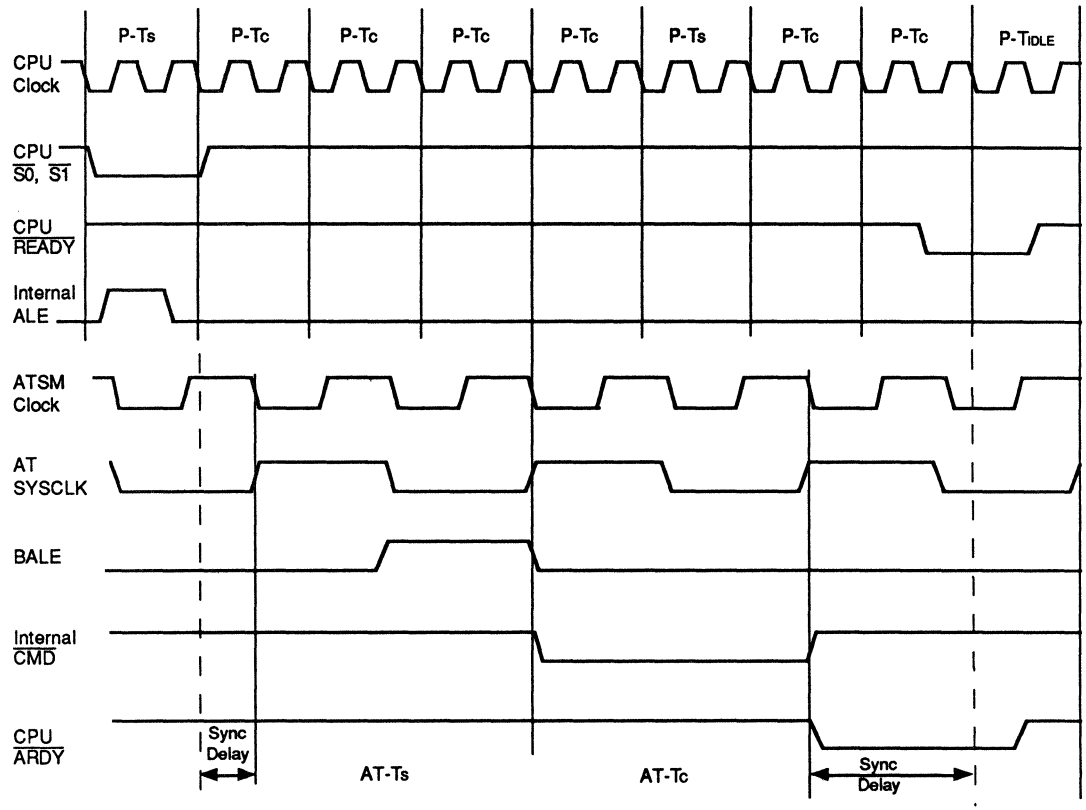
15620B-016B

ASYNC CYCLES

The ASYNC clocking mode is defined as the CPU clock and the ATSM clock either being sourced differently, or having the same source and having divisors that do not fit the SYNCSAME, SYNCDIFF, or FASTBUS mode definitions. ASYNC mode allows totally asynchronous CPU and ATSM clocks, and automatically handles the processor to state machine synchronization issues. This mode is especially useful when a fixed-speed AT system bus (S Bus) is desired for compatibility, but a higher processor speed, that is not twice the S-bus speed, is desired. The only drawback in this mode is that time must be spent synchronizing the ATSM to the CPU and vice versa. See Figure 3-11 for an example ASYNC mode transfer cycle.

In ASYNC mode, the ATSM status cycle (AT-TS) will start after the end of the CPU status cycle, and only then after a rising edge on SYSCLK is seen. After the proper phase of SYSCLK is established, BALE can be sent out and the command can be activated.

Figure 3-11 ASYNC Mode AT State Machine Cycle



P–Ts = Processor State Cycle
P–Tc = Processor Command Cycle
AT–Ts = ATSM Status Cycle with BALE
AT–Tc = ATSM Final Command Cycle

15620B-017B

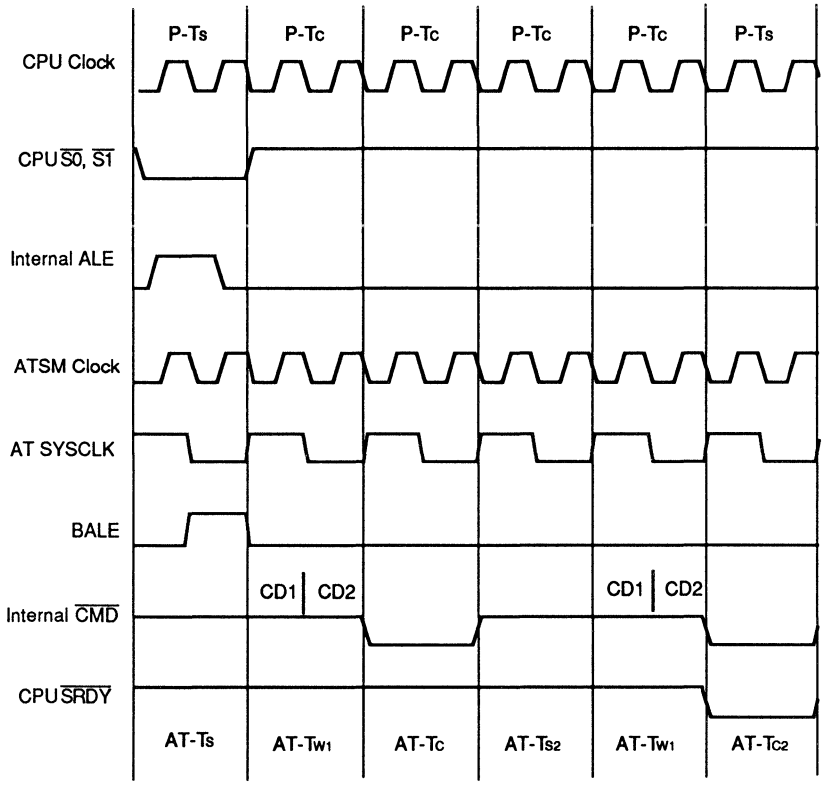
The ATSM uses the asynchronous ready line (ARDY) going back to the CPU control logic for cycle termination. The ARDY line is activated after the command strobe goes inactive, and is driven inactive after the end of the CPU transfer cycle.

FASTBUS CYCLES

The FASTBUS clocking mode is defined as both the CPU clock and ATSM clock having the same clock source pin and divisor, and with the FBE configuration bit in CCR1 (index 01H) set to 1 (enabled). The effect of this clocking mode upon ATSM operation is the same as SYNCSAME mode, except that the AT status cycle is coincident with the CPU status cycle, and that all cycles are echoed onto the S Bus. Figure 3-12 shows the synchronous relationship between the CPU states and the command and control signals generated by the ATSM. It also shows the coincident CPU and ATSM status cycles.

If the FBE bit in CCR1 is set to 1 without the clock sources and divisors the same for the CPU and ATSM clocks, FASTBUS mode is not enabled, and the FBE bit has no effect. Because all cycles are echoed out to the S Bus in FASTBUS mode, AT bus "snooper" or debugger cards will be able to monitor all internal bus transactions.

Figure 3-12 FASTBUS Mode AT State Machine Cycle



P-Ts = Processor State Cycle
P-Tc = Processor Command Cycle
AT-Ts = ATSM Status Cycle with BALE
AT-Tw = ATSM Wait State
AT-Tc = ATSM Final Command Cycle

15620B-018B

POWER SAVING FEATURES (Am286LX ONLY)

The Am286LX processor version of the device supports several power saving features to allow a degree of control for both system and device level power consumption. Clock shutdown modes and refresh generation options are included in the Am286LX processor to provide the capability to reduce power consumption in battery powered systems. These features are not supported in the Am286ZX processor version of the device.

Clock Control

The Am286LX processor is a low-power, static CMOS design. Therefore, power consumption is directly proportional to the speed of the clock supplied to the active circuitry. If the clocks to significant amounts of logic on the device are stopped, the power consumption of the device drops significantly. This is exactly what is implemented on the Am286LX processor.

There are two different "stop clock" modes that can be entered under software control: CPU Stop Clock mode and System Standby mode. No external hardware assistance is required to support these power conservation modes. The input clocks, PROCLK and IOCLK, can continue to run. The internal control logic supporting these two modes disables the routing of the clocks to different logic blocks under control of configuration register bits. Figure 3-13 shows a state transition diagram for the power saving modes.

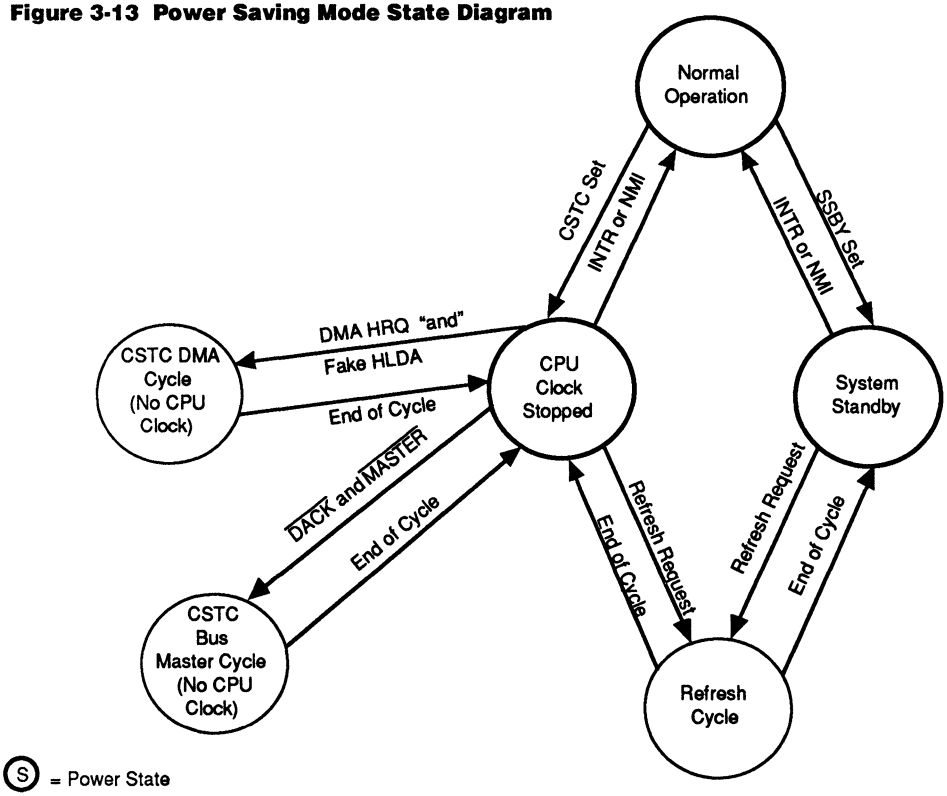
CPU STOP CLOCK MODE

Entering CPU Stop Clock mode stops the clock to the 80C286 microprocessor core. All other clocks continue to run. Therefore, refresh cycles, DMA cycles, and external bus master cycles can continue while the CPU clock is stopped.

Any interrupt from the internal interrupt controllers or from the NMI generation logic will restart the CPU clock, with no loss of register data or state information in the CPU. An interrupt from the internal interrupt controllers will restart the clock regardless of the CPU's interrupt enable flag.

CPU Stop Clock mode is entered by a 0 to 1 transition in the CSTC bit in the Power Save Function Register (PSFR, index 23H). The bit is not cleared upon exiting CPU Stop Clock mode; it must be explicitly written to 0 after exiting. Writing a 1 to the CSTC bit when it already contains a 1 will not initiate the power saving mode. The default value for the CSTC bit is 0 (normal operation).

Figure 3-13 Power Saving Mode State Diagram



15620B-019B

CPU Stop Clock Control

CSTC PSFR.0	Function
0	Normal operation
0 → 1	Stop CPU clock
1	Normal, after mode exit

SYSTEM STANDBY MODE

Entering System Standby mode stops the clocks going to the 80C286 microprocessor core, the DMA controllers, and the CPU and Bus Control logic. All clocks not having functional impact on refresh cycles are stopped. Refresh cycles can continue while these other clocks are stopped.

Any interrupt from the internal interrupt controllers or from the NMI generation logic will restart the clocks, with no loss of register data or state information in the device. An interrupt from the internal interrupt controllers will restart the clock regardless of the CPU's interrupt enable flag.

System Standby mode is entered by a 0 to 1 transition in the SSBY bit in the Power Save Function Register (PSFR, index 23H). The bit is not cleared upon exiting System Standby mode; it must be explicitly written to 0 after exiting. Writing a 1 to the SSBY bit when it already contains a 1 will not initiate the power saving mode. The default value for the SSBY bit is 0 (normal operation).

System Standby Mode Control

SSBY PSFR.1	Function
0	Normal operation
0 → 1	Enter System Standby
1	Normal, after mode exit

If both the SSBY and the CSTC bits are set at the same time, System Standby mode is in effect, as it is a super set of the CPU Stop Clock mode.

Refresh Cycle Options

There are two power saving features implemented in the DRAM controller of the Am286LX processor: slow refresh and staggered refresh. These two features are described in the Memory Subsystem section, with cycle diagrams in the M-Bus Interface chapter.

BUS MASTER MODE

Functional Description

The Am286ZX/LX integrated processor is normally the main CPU of an AT architecture motherboard. However, the device's bus master mode allows it to act as a bus master peripheral located on the expansion bus of a host system. Bus master mode provides the capability for the Am286ZX/LX integrated processor to be the CPU on a high-performance, bus-master add-in card. The device directly supports ISA bus mastering and, with some external logic, bus mastering on MCA and EISA systems.

A bus mastering peripheral based on the Am286ZX/LX integrated processor can provide two important capabilities for a host system: high-performance CPU independent data transfers, and computing/processing capability for the peripheral, also independent of the

host CPU. Both these capabilities have the net effect of increasing system level throughput without having to speed up the host CPU.

Bus master mode should not be confused with the ability of the device to support external bus masters when in a main system CPU application. Supporting external bus masters is a standard AT architecture feature which is fully implemented in the Am286ZX/LX integrated processor, without any special configuration or connection needed.

A bus mastering application for the device would have the S Bus logically connected (either directly or through buffers) to the AT expansion bus of a host system. Physically, the Am286ZX/LX integrated processor could be on a expansion card plugged into an AT compatible motherboard. It is important to note that the Am286ZX/LX integrated processor can process information, access local memory, access X-bus peripherals, and execute DMA cycles locally without activating the S-bus interface. S-bus data transfers to local devices on the bus master board can even be done in buffered master mode (described below).

ENABLING BUS MASTER MODE

Bus master mode is enabled by setting the configuration bit BM in the Bus Master Control Register (BMCR, index 0AH). With bus master mode enabled, the bus acquisition and cycle redirection logic is enabled, and the definitions of pins associated with bus mastering change. The default value of 0 for the BM bit selects normal mode operation. The BM bit should be set as soon as possible after reset in a bus mastering application.

Bus Master Mode Selection

BM BMCR.0	Selection
0	Normal mode
1	Bus master mode enabled

Selecting bus master mode changes the operating mode of several S-bus pins to allow direct connection of the Am286ZX/LX integrated processor to the system expansion bus of another system.

When bus master mode is enabled, the DRQ3 (DMA Request 3) pin changes to an output which is used to request access to the host bus. It is driven by the bus acquisition logic.

The $\overline{\text{DACK3}}$ (DMA Acknowledge 3) pin, which is normally an output, becomes an input to the bus acquisition logic, which is monitored for a reply to the DRQ3 sent out in order to request the bus.

Also, the $\overline{\text{MASTER}}$ pin becomes an output which is used to turn off the host's bus drive capability after a $\overline{\text{DACK3}}$ is granted, so that the Am286ZX/LX integrated processor can start driving the bus. Also, the strong pull-up on the $\overline{\text{MASTER}}$ pin is disabled in bus master mode, regardless of the state of the BCON bit (BCR, index 09H, bit 0).

CYCLE REDIRECTION

Data transfer between the master device and the host system's expansion bus is accomplished by cycle redirection. The target of the data transfer two or from the host system may have the same address as a local resource (a local memory location or I/O peripheral). Therefore, some means of redirecting a cycle which would normally be directed to a local peripheral or memory device must be provided.

Cycle redirection is implemented in the Am286ZX/LX integrated processor by means of a page register and two enable bits (memory and I/O redirect enables). In general, the page

register specifies either a memory or I/O address range for redirecting cycles out onto the S Bus. The enable bits turn on the redirection feature and specify which type of cycle to redirect. Note that no cycle redirection will ever occur without the bus master mode enabled.

The Bus Master Page Register (BMPR, index 0BH) contains the 8-bit page specifier which will specify a memory or I/O address range for cycle redirection. During memory cycle redirection, bits 7–0 are compared to CPU address lines 23–16, respectively, to determine a "redirect hit" cycle. Therefore, for host memory accesses the register defines a 64-kb page. During I/O cycle redirection, bits 7–0 are compared to CPU address lines 9–2, respectively, to determine a "redirect hit" cycle. Therefore, for host I/O accesses the register defines a 4-byte page. The page register contents are ignored unless one of the cycle redirect enables (HIE or HME) is set. The BMPR configuration register defaults to a value of 0 for all bits.

The HIE (Host I/O Enable) and HME (Host Memory Enable) configuration bits in the Bus Master Control Register (BMCR, index 0AH) control the cycle redirection process for their respective cycle types. The value of these two bits has no effect if bus master mode is not enabled. Also, the HIE and HME bit functions are mutually exclusive; they should never both be set at the same time. The default value for HIE and HME configuration bits is 0 (redirection disabled).

Host I/O Redirection Enable

HIE BMCR.4	Function
0	Normal Operation
1	Redirect I/O to S Bus

Host Memory Redirection Enable

HME BMCR.3	Function
0	Normal Operation
1	Redirect Memory to S Bus

BUS ACQUISITION METHODS

If bus master mode is enabled, and the page register and one of the redirection enable bits are set, then transfer cycles to the host system can take place as soon as the host bus is acquired. The process of bus acquisition proceeds as follows.

1. The bus master Am286ZX/LX integrated processor asserts DRQ3, requesting a cascade mode DMA channel.
2. The device waits for a $\overline{\text{DACK3}}$ from the host DMA controller to go active (Low). This signifies that the bus master can take over the bus.
3. The device drives $\overline{\text{MASTER}}$ active at the beginning of a transfer cycle, and it drives the pin inactive again at the end of the cycle.

The device can handle bus acquisition relative to redirection cycles in two ways: auto acquire and forced acquire. The trade off between these two methods is local cycle redirection overhead versus host bus utilization.

The auto acquire access mode will acquire and release the host bus at the beginning and end of each redirected cycle, with the exception of back to back redirected cycles. Using this access mode will keep the bus master off the host bus as much as possible, but it will

also decrease the data transfer rate that the Am286ZX/LX integrated processor-based bus master can accomplish due to higher overhead per redirect cycle.

The forced acquire access mode utilizes two bits in the Bus Master Control Register (BMCR, index 0AH) to acquire and hold the host bus while several redirect cycles take place, releasing the bus after the cycles are complete. The bus acquire and release process is under software control via these configuration bits.

The BA (Bus Acquire) bit in BMCR provides the bus master control program with direct control over the bus acquisition process. Setting the BA bit will initiate a bus acquire. The host bus will be held as long as the BA bit is set. When the bit is reset, the host bus will be released. The default value of the BA configuration bit is 0.

Bus Acquisition Control

BA BMCR.1	Function
0	Release Host S Bus
1	Acquire and Hold S Bus

The AS (Acquire Status) bit in BMCR provides a read only status bit which indicates the acquire status of the host bus. It is not a requirement that the control program poll the AS bit before the first redirect cycle (but after the BA bit is set), because a redirect cycle will always wait for bus acquisition before proceeding with the cycle.

Acquire Status

AS BMCR.2	Indicated Status
0	Host CPU owns its S Bus
1	Host bus acquired

BUS RELEASE

With the auto acquisition method being used, the bus is released back to the host at the end of each redirect cycle, except for back-to-back redirect cycles. With forced bus acquisition, the bus release is triggered by a 1 to 0 transition of the BA bit. The bus release process proceeds as follows:

1. $\overline{\text{MASTER}}$ and $\overline{\text{DRQ3}}$ are deactivated, releasing the host S Bus.
2. The host will then deactivate $\overline{\text{DACK3}}$ and regain bus ownership.

Configuration and Connection

BUFFERING MODES

The Am286ZX/LX integrated processor can be configured to operate with two different types of connections between the device's S Bus and the host's expansion bus. The placement of buffers in this connection path and the corresponding internal configuration comprises the "buffered bus master mode." The direct wired connection of the device to another host bus and the corresponding internal configuration comprises the "unbuffered bus master mode."

The combination of the BM (Bus Master Mode) bit in the Bus Master Control Register (BMCR, index 0AH) and the BCON (Bus Connect) bit in the Bus Control Register (BCR, index 09H) defines the buffering mode of the device. The interaction of these two

configuration bits also effects the characteristics of some of the S-bus output buffers on the device. See Figure 3-14 for details on the effect of the buffering mode on the S-bus interface.

Buffering Mode Specification

BM BMCR.0	BCON BCR.0	Mode
0	0	S-bus disconnect (default)
0	1	Normal host mode
1	0	Unbuffered master mode
1	1	Buffered master mode

UNBUFFERED BUS MASTER MODE

Unbuffered bus master mode allows direct connection of the device's S Bus to the host's AT system bus, without intervening buffers. This simplicity of interface reduces chip count and cost, and also saves board area for an intelligent bus mastering peripheral. S-bus loading analysis should be done for the target system environment of the bus mastering peripheral.

While not executing data transfers on the host system bus, the Am286ZX/LX integrated processor-based peripheral can be executing code performing data transfers on the M Bus and X Bus. The M Bus can be used to access the local DRAM array. The X Bus can be used to access ROM/EPROM devices, X-bus SRAM, and 8- and 16-bit I/O peripherals. Also, 8-bit X-bus DMA peripherals are supported in unbuffered bus master mode, with DMA channels 0, 1, and 2 available.

Figure 3-15 shows a block diagram for an unbuffered bus master mode application with the Am286ZX/LX device. The latch on the LA address lines provides for DMA style addressing during the bus master access to the host system, as opposed to the normal pipelined timing of the LA lines. $\overline{\text{MEMCS16}}$ and $\overline{\text{IOCS16}}$ should be held active, at a logic Low, during bus master accesses because data byte routing is performed by the host system based on the target device size.

Figure 3-14 S-Bus Buffering Modes

Mode	BM (BMCR.0)	BCON (BCR.0)	$\overline{\text{MASTER}}$	DRQ3	$\overline{\text{DACK3}}$	S-Bus Pull-ups*	S-Bus Connection/ Operation
Reset/ Power-up	0	0	Input	Input	N.C.	Not Active	Disconnected. No Cycles Allowed
Normal Host	0	1	Input	Input	Output	Active	Normal Operation
Unbuffered BMM	1	0	Output	Output	Input	Not Active	Bus only active while host bus acquired
Buffered BMM	1	1	Output	Output	Input	Active	Bus active during acquired host and during local S-bus cycles

* Pins affected are LA23–LA17, SA19–SA0, SD15–SD0, $\overline{\text{MEMW}}$, $\overline{\text{MEMR}}$, $\overline{\text{IOW}}$, $\overline{\text{IOR}}$, $\overline{\text{IOCHK}}$, IOCHRDY, $\overline{\text{P0WS}}$, $\overline{\text{MEMCS16}}$.

15620B-020B

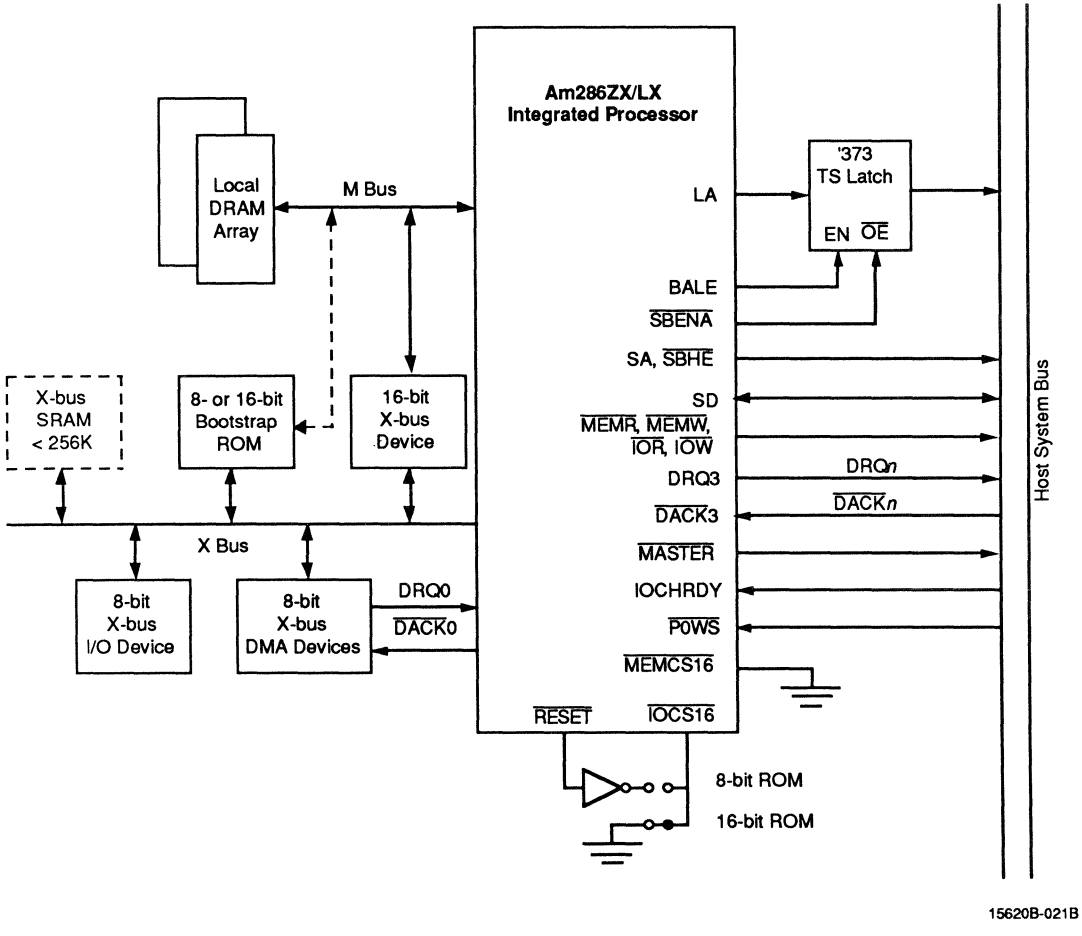
BUFFERED BUS MASTER MODE

The buffered bus master configuration utilizes buffers between the device's S Bus and the host's AT system bus. This configuration can provide greater host bus drive capability, and it will allow for a private, local S Bus with ISA timing and function compatibility.

While not executing data transfers on the host system bus, the Am286ZX/LX integrated processor-based peripheral can be executing code and performing data transfers on the M Bus, X Bus, and the private local S Bus. As with the unbuffered configuration, the M Bus can be used to access the local DRAM array. The X Bus can be used to access ROM/ EPROM devices, X-bus SRAM, and 8- and 16-bit I/O peripherals. Also, 8-bit X-bus DMA peripherals are supported, with DMA channels 0, 1, and 2 available.

Buffered bus master mode, however, provides a private local S Bus independent of the host system bus. This local bus can support on board memory, I/O, and DMA peripherals of both 8- and 16-bit sizes.

Figure 3-15 Unbuffered Bus Master Mode



15620B-021B

Figure 3-16 shows a block diagram for a buffered bus master mode application with the Am286ZX/LX device. As with the unbuffered application, a latch must be provided for the host LA lines. The host enable signal, HSTENA, is only active when both the MASTER and the SBENA control signals are active. This will provide proper bus drive timing on an ISA host bus, relative to the activation of the MASTER pin. The diagram also shows the generation of the local S-bus command signals with a disabling function during actual bus master transfer cycles. The local S-bus peripherals will see the address and data lines transitioning, but they will not see a local transfer cycle because all the read/write command lines are inactive. A multiplexer between local and host signals must be provided for the four S-bus cycle feedback lines IOCS16, MEMCS16, IOCHRDY, and P0WS.

Bus Master Mode Transfer Cycles

AUTO ACQUIRE CYCLES

Figure 3-17 is a cycle diagram showing several auto acquire cycles. The first one is a single access with a full bus acquire and release handshake for the cycle. The second two cycles, a memory read and a memory write, demonstrate that back-to-back cycles do not release the bus between cycles.

In the diagram, circle 1 shows the initiation of a bus acquire due to a "redirect hit" off the page register contents and the HME bit. Circle 2 shows the driving of MASTER as soon as the DACK3 comes back from the host in response to the DRQ3 being asserted. At circle 3, the actual transfer cycle starts with the output buffers being enabled, as indicated by the SBENA pin going active. At circle 4, the cycle is finished so the bus is released by driving the MASTER and DRQ3 pins inactive.

FORCED ACQUIRE CYCLES

Figure 3-18 is a cycle diagram showing several forced acquire cycles. It demonstrates acquisition of the bus with the Bus Acquire (BA) bit, multiple transfer cycles with other intervening cycles while the bus is acquired from the host, and eventual release of the bus by clearing the BA bit.

In the diagram, circle 1 shows the initiation of a bus acquire due to the BA configuration bit in BMCR being set by an internal I/O write to the register. Circle 2 shows the initiation of a transfer cycle, with MASTER going active due to a "redirect hit" off the page register contents and the HME bit. At circle 3, the actual transfer cycle starts with the output buffers being enabled, as indicated by the SBENA pin going active. At circle 4, the bus is released as a result of another internal I/O write to BMCR which clears the BA bit.

MEMORY SUBSYSTEM

DRAM Interface

CONTROLLER FEATURES

The Am286ZX/LX integrated processor DRAM support consists of a multimode DRAM controller which addresses a 4 bank DRAM array containing up to 16 Mb of memory. The controller typically can directly drive at least two unbuffered banks (36 DRAM devices), with bank expansion to four banks by adding a buffer for the memory address lines (MA9–MA0). The need for address buffering depends on the capacitive loading on these lines. The DRAM controller supports 256-kb, 1-Mb, and 4-Mb type DRAM's in non-paged, paged, and paged interleave modes. Any DRAM type can go in any bank; the DRAM controller does not limit the system in bank DRAM configurations.

Figure 3-16 Buffered Bus Master Mode Block Diagram

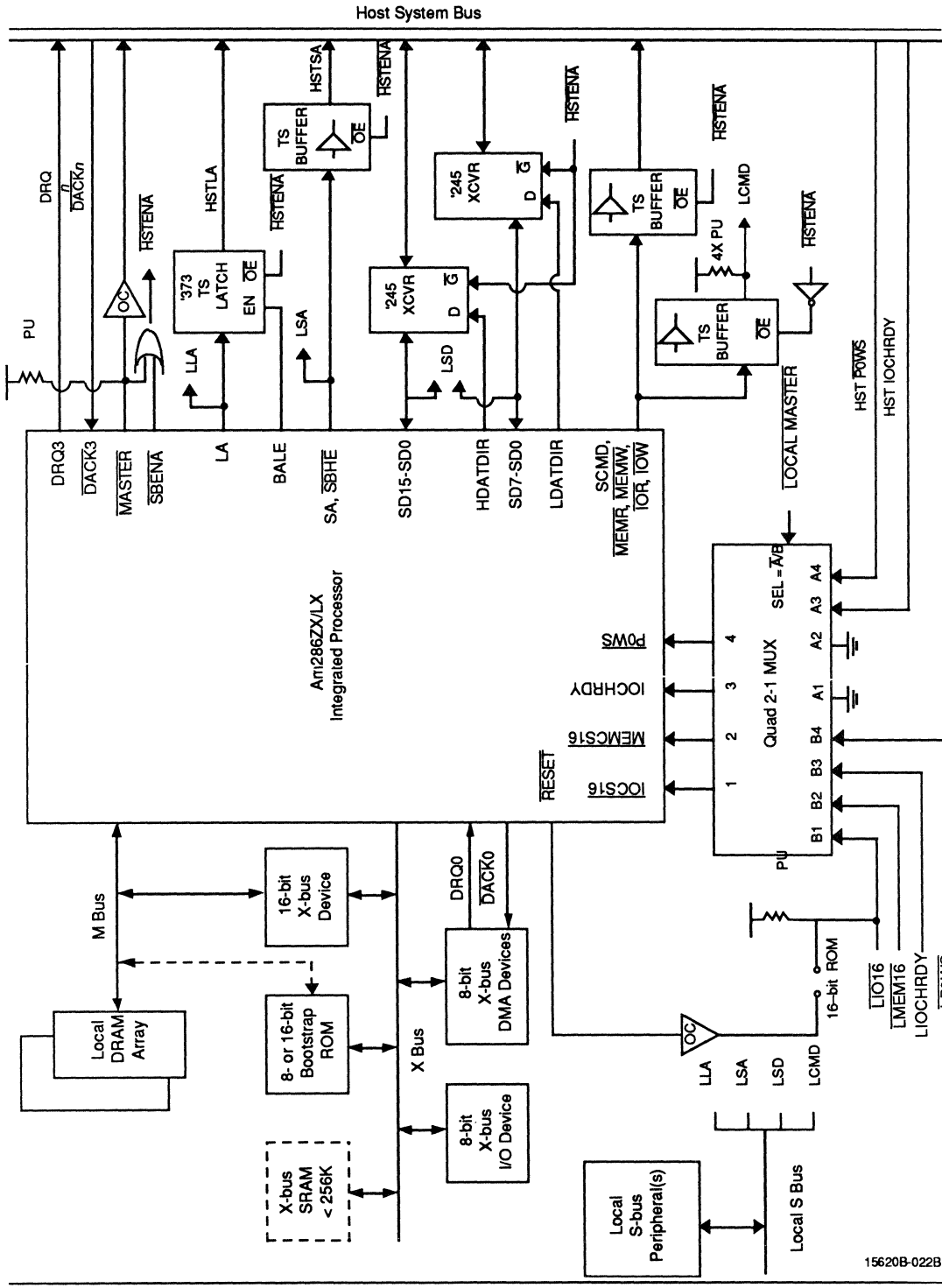


Figure 3-17 Auto Acquire Bus Master Transfer Cycles

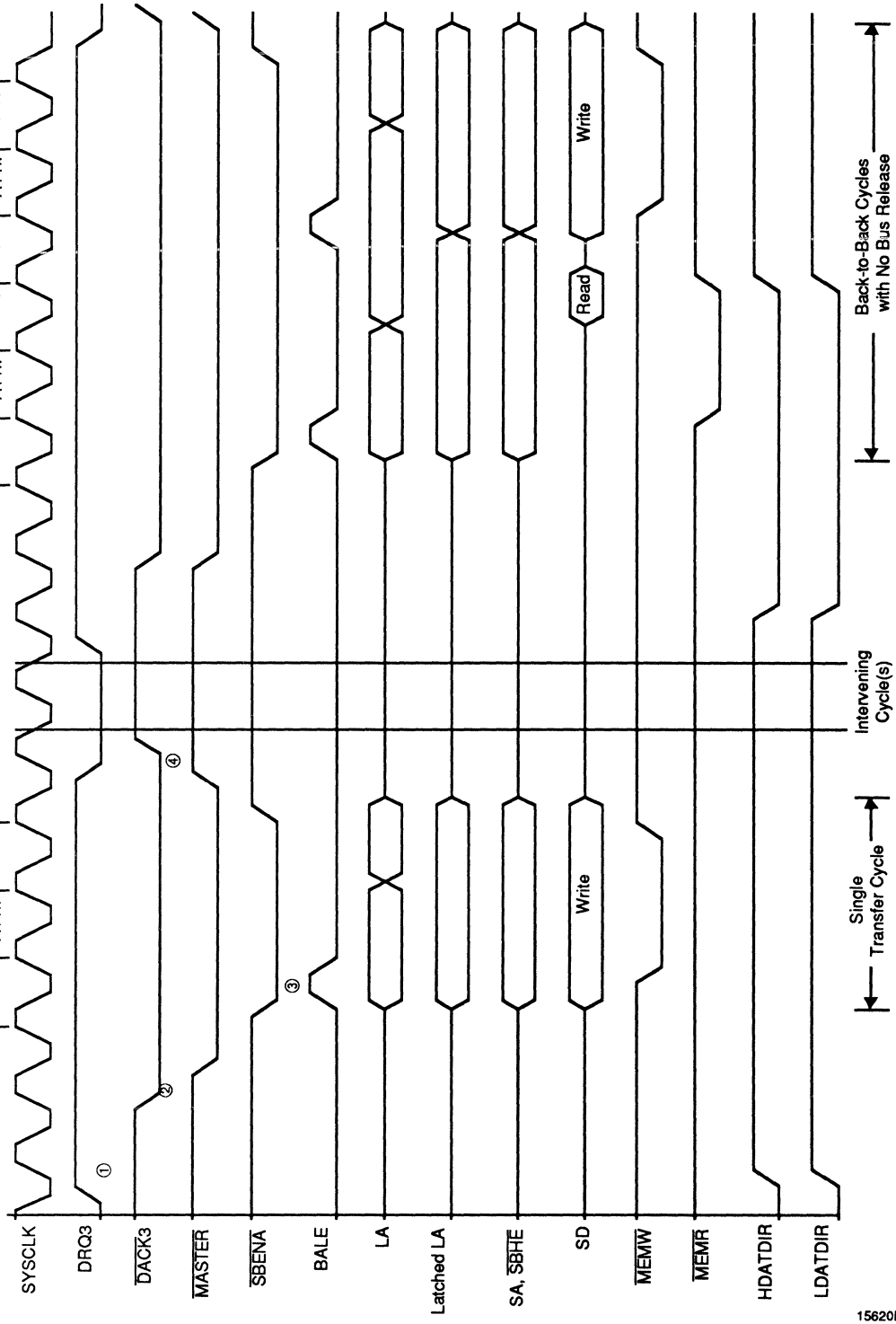
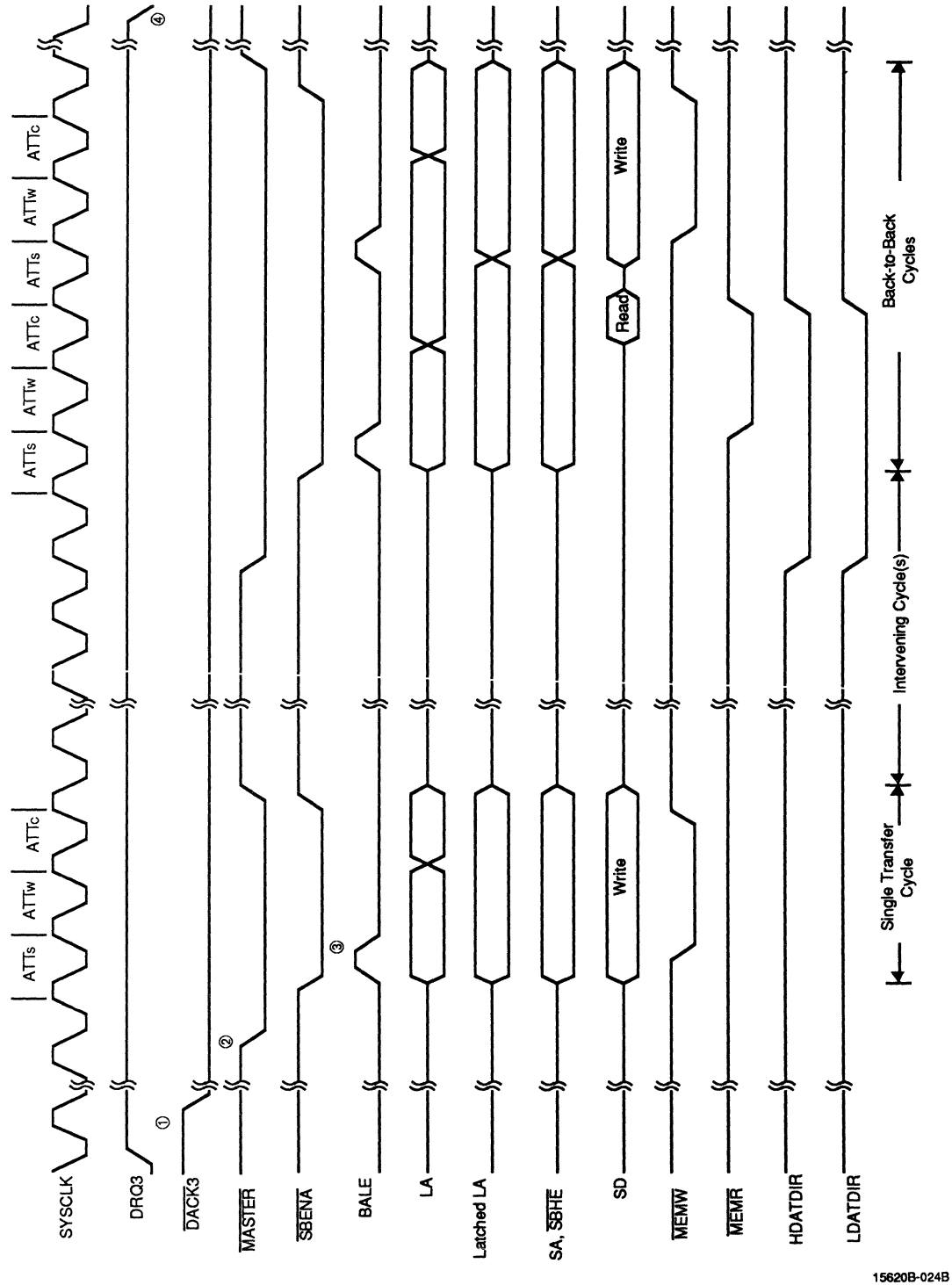


Figure 3-18 Forced Acquire Bus Master Transfer Cycles



The DRAM controller can be programmed for 0, 1, or 2 wait states and provides parity generation, checking, and masking. The controller optimizes performance by selecting the appropriate interleaving mode for the programmed DRAM configuration. For the controller to implement interleaving, adjacent banks must be loaded with identical size DRAMS. Pairs of identical banks allow two-way interleaving, and a configuration of four homogeneous banks allows four-way interleaving. The automatic detection of interleave options not only optimizes performance, but also prevents inadvertent system lockup by ignoring the interleave selection when the programmed DRAM configuration precludes interleaving. If interleaving is programmed but not possible, the controller does not attempt interleaving.

The DRAM controller is tightly coupled with the system's address manager for efficient memory accesses in memory mapping schemes such as LIM 4.0. When accessing a mapped address the address manager warns the DRAM controller, which adjusts the memory access cycle timing to assure proper memory address setup. Figure 3-19 illustrates where the DRAM controller resides in the system architecture.

MODES

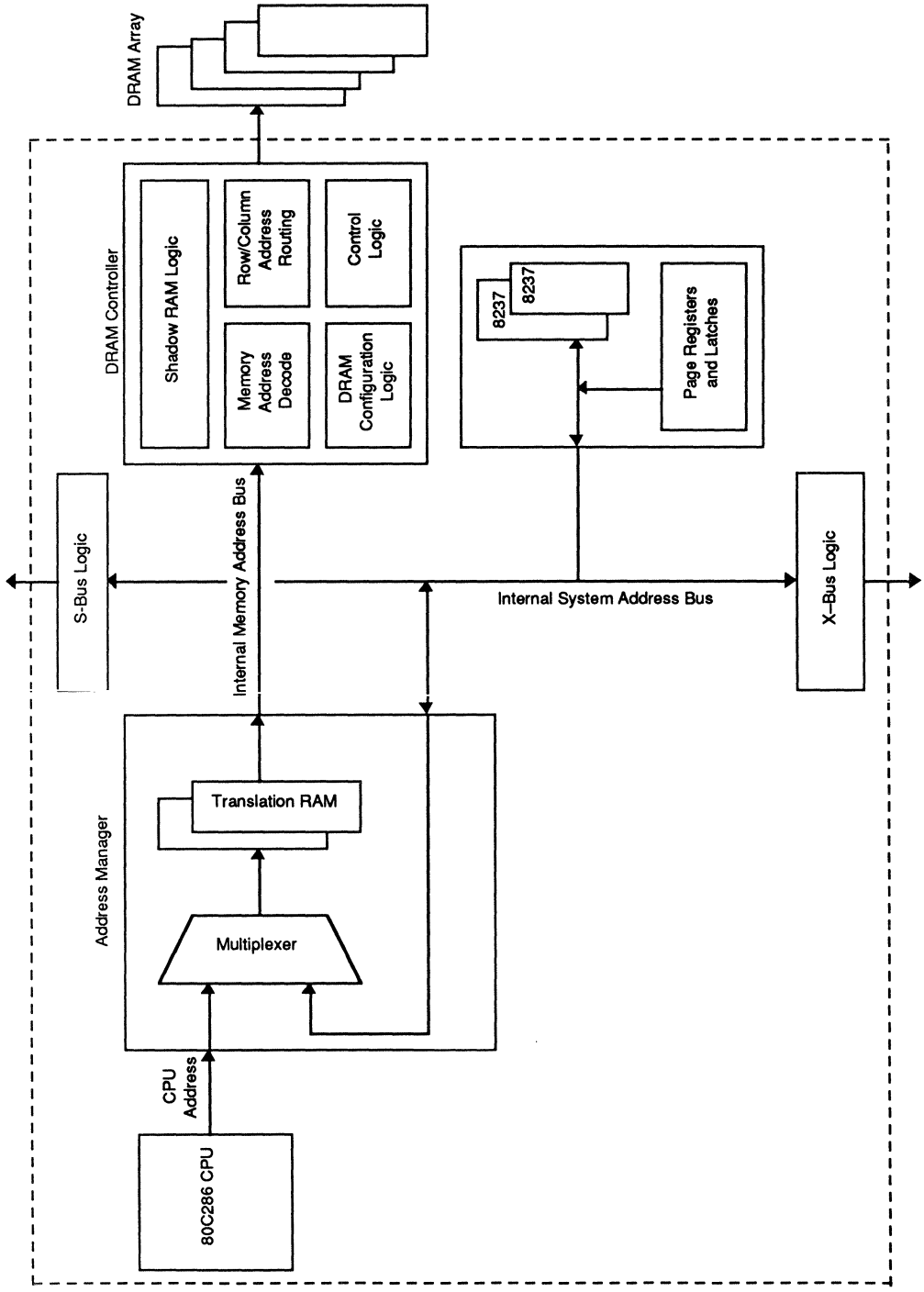
The DRAM Configuration Register (DCR, index 16H) interleave-enable and page-enable bits (INE and PGE) control the access mode of the DRAM controller. The default mode of operation is non-paged, non-interleaved DRAM accesses (INE and PGE disabled). To enable page mode, the PGE bit is set. To enable page-interleaved mode, PGE and INE are set.

Non-Page Mode: This mode accesses the DRAM array in the standard manner. For a DRAM read access, the controller asserts the row address (on MA) followed by the Row Address Strobe ($\overline{RAS}$) for the appropriate bank; the address is then changed to the column address followed by the appropriate Column Address Strobe ($\overline{CAS}$). During write accesses, the write enable signals are asserted ($\overline{MWENH}$ and $\overline{MWENL}$). This type of DRAM access can be the most efficient in its use of CPU cycles. However, it generally requires very fast DRAM's for good performance, or requires the insertion of wait states for a less expensive, lower performance level.

Page Mode: This mode employs the page access technique available in most DRAM's to provide near zero wait state performance. In this type of access, the row address strobe stays active while only column addresses appear on the Memory Address bus (MA), and only column address strobes need to be asserted. This reduced access protocol dramatically shortens the access cycle. The DRAM controller inserts wait states only when the current access requires a change in the row address of the DRAM. This technique allows the use of slower DRAM at higher CPU clock frequencies, providing for a cost effective high-performance system.

Page-Interleave Mode: This mode adds interleaving to the paging technique, accessing a given bank on every other word (two-way interleave) or on every fourth word (four-way interleave). The net effect of this strategy is to increase the size of the DRAM row address page, thus allowing the processor to fetch from the local memory longer without incurring wait states.

Figure 3-19 Am286ZX/LX Integrated Processor Address Architecture



DRAM Controller Mode Selection

DCR.2 PGE	DCR.1 INE	Access Mode
0	0	Non-paged (default)
0	1	Illegal configuration
1	0	Paged, non-interleaved
1	1	Paged-interleaved

The condition where INE is set without PGE being set will result in unpredictable memory operation.

When changing the DCU in and out of page mode, the page comparator must be cleared after setting or clearing PGE. If the page comparator is not cleared, the DCU will not execute subsequent cycles properly. When switching to page mode, the first cycle may be interpreted as a page hit rather than a page start. When switching from page to non-page mode, the $\overline{RAS}$ lines may be left active, preventing non-page cycles from completing.

The simplest way to reset the page comparator is to toggle the slow refresh control bit of the DCR (SRE, bit 6 of DCR) after changing the PGE bit. The following code fragments are examples of how to change the DCU mode. These code fragments must be executed from memory other than DCU-controlled DRAM. If they are executed from DCU-controlled DRAM, the internal processor of the Am286ZX/LX integrated processor may attempt to pre-fetch code or data between writing the page mode enable bit and toggling the slow refresh bit. Any code or data accesses between these operations is not guaranteed to be correct. It is recommended that these code fragments be implemented in EPROM.

1. Changing from Non-Page mode to Page mode ROM routine:

```
PUSH    AX              ;SAVE AX
PUSH    BX              ;SAVE BX

MOV     AL, 016H
OUT     022H, AL        ;SET CONFIGURATION POINTER TO DCR
IN      AL,023H         ;READ IN DCR  INTO AH
OR      AL,006H         ;SET BITS 1,2 FOR PAGE MODE AND
                        ;   INTERLEAVED.  LOGICALLY OR AH
                        ;   WITH 04 FOR PAGE MODE AND
                        ;   NON-INTERLEAVED MODE.
OUT     023H,AL         ;WRITE DCR  WITH NEW  VALUE

;NOW TOGGLE SLOW REFRESH BIT
MOV     BL,AL           ;SAVE DESIRED VALUE FOR DCR IN BL
AND     AL,0BFH         ;FORCE SLOW REFRESH BIT LOW FOR
                        ;         TOGGLE
OUT     023H,AL         ;WRITE TO DCR

OR      AL,040H         ;FORCE SLOW REFRESH BIT HIGH
OUT     023H,AL         ;WRITE TO DCR

AND     AL,0BFH         ;FORCE SLOW REFRESH BIT LOW
OUT     023H,AL         ;WRITE TO DCR

MOV     AL,BL           ;WRITE DESIRED  FINAL VALUE OF DCR
```

```
OUT     023H,AL         ;WRITE TO DCR WITH DESIRED DCR VALUE

POP     BX                      ;RESTORE BX
POP     AX                      ;RESTORE AX
```

2. Changing from Page mode to Non-Page mode ROM routine:

```
PUSH  AX                ;SAVE AX
PUSH  BX                ;SAVE BX

MOV    AL, 016H
OUT    022H, AL         ;SET CONFIGURATION POINTER TO DCR
IN     AL,023H          ;READ IN DCR  INTO AH
AND    AL,0F9H          ;CLEAR BITS 1,2 FOR PAGE MODE AND
                        ;    INTERLEAVED.
OUT    023H,AL          ;WRITE DCR  WITH NEW  VALUE

;NOW TOGGLE SLOW REFRESH BIT
MOV    BL,AL            ;SAVE DESIRED VALUE FOR DCR IN BL
AND    AL,0BFH          ;FORCE SLOW REFRESH BIT LOW FOR
                        ;    TOGGLE
OUT    023H,AL          ;WRITE TO DCR

OR     AL,040H          ;FORCE SLOW REFRESH BIT HIGH
OUT    023H,AL          ;WRITE TO DCR

AND    AL,0BFH          ;FORCE SLOW REFRESH BIT LOW
OUT    023H,AL          ;WRITE TO DCR

MOV    AL,BL            ;WRITE DESIRED  FINAL VALUE OF DCR
OUT    023H,AL          ;WRITE TO DCR WITH DESIRED DCR VALUE

POP    BX               ;RESTORE BX
POP    AX               ;RESTORE AX
```

REFRESH

The DRAM controller provides several refresh options. The default operation employs the standard $\overline{RAS}$ only refresh at the standard rate prescribed by the 8254 timer channel 0. In this mode all DRAM banks are refreshed at once (all $\overline{RAS}$ signals become active at the same time).

The Am286LX processor DRAM controller includes slow refresh support for new DRAMs which can operate longer without refresh. In this mode, system refresh requests are generated at one-eighth of the rate prescribed by the 8254 timer channel 0 output. Refresh is a major system current consumer, thus fewer refresh cycles will extend battery life in portable systems. With refresh consuming less of the system bus bandwidth, system performance will also be enhanced.

The Am286LX processor slow refresh feature is controlled by the SRE bit in the DRAM Configuration Register (DCR, index 16H). The Slow Refresh Enable configuration bit defaults to disabled.

Slow Refresh

DCR.6 SRE	Function
0	Slow refresh disabled
1	Slow refresh enabled

Staggered refresh mode is provided to reduce peak current demand in systems using the Am286LX processor. During a staggered refresh cycle, the $\overline{RAS}$ signals are not asserted in unison. Instead, the $\overline{RAS}$ signals are staggered. First banks 0 and 2 are refreshed, followed by banks 1 and 3. In choosing this option, the system designer trades a small amount of $\overline{RAS}$ pulse width margin for a savings in peak current demand. This is particularly useful in laptop designs. Staggered refresh is enabled by setting the STR bit in the DRAM Configuration Register (DCR, index 16H). The staggered refresh configuration bit defaults to disabled.

Staggered Refresh

DCR.7 STR	Function
0	Staggered refresh disabled
1	Staggered refresh enabled

$\overline{RAS}$ Time Out

With page mode enabled, it is possible while in a tight code loop to spend an extended period of time with $\overline{RAS}$ active. The probability of this situation increases as interleaving increases the effective DRAM page size. In this case it is possible for the system to violate the DRAM maximum $\overline{RAS}$ pulse width specification. The DRAM controller provides a $\overline{RAS}$ Time Out circuit to prevent this DRAM specification violation. $\overline{RAS}$ time out is controlled with the RTE bit, DRAM Configuration Register bit 4 (DCR, index 16H). The RTE configuration bit defaults to disabled. It is recommended that this bit be set when page mode is enabled. DRAM's typically have a $\overline{RAS}$ active maximum specification of 10 µs (10,000 ns). When RTE is enabled, $\overline{RAS}$ will be automatically deasserted after ten times the period of the 82C54 input clock, which is derived from the F14MHZ input clock divided by 12.

$\overline{RAS}$ Time Out Enable Selection

DCR.4 RTE	$\overline{RAS}$ Time Out Enable
0	Disabled
1	Enabled

PARITY CONTROL

The DRAM controller provides a parity enable feature which is an addition to the standard AT-compatible parity controls provided in PORT B (I/O address 61). Parity checking can be enabled or disabled with the PE bit, DRAM Configuration Register bit 5 (DCR, index 16H). The PE configuration bit defaults to disabled. To enable parity checking, PE must be set. Parity errors may be cleared by clearing the PE bit once an error has been registered.

Parity Enable Selection

DCR.5 PE	Parity Enable
0	Disabled
1	Enabled

DRAM TYPE SELECTION

The type of DRAM installed in each bank is set in the RAM Bank Configuration register (RABR, index 15H). RABR consists of a two bit field for each of the four banks. The configuration bits for DRAM Type default of 00 for each bank signifies that no DRAM is installed. If all DRAM type fields in RABR are programmed for no DRAM installed, the DRAM controller will not respond to any CPU memory accesses, allowing the transaction to take place on the external S Bus. The DRAM type can be changed by writing the following bit patterns to each of the fields.

Example DRAM Type Selection

RABR.1 D0T1	RABR.0 D0T0	DRAM Type
0	0	Not installed
0	1	256 kb
1	0	1 Mb
1	1	4 Mb

If the DRAM type selections for the four banks total to more than 16 Mb, only the first 16 Mb is used, starting from bank 0 and progressing towards bank 3.

LOCAL MEMORY WAIT STATES

There are many ways to approach cost/performance ratio issues in system design. Fast systems have few if any wait states in memory access cycles and typically require expensive memory arrays. Slower systems might trade performance for lower system cost by opting to use slower memories, requiring wait states be added to CPU memory access cycles. The DRAM controller wait states can be programmed to the minimum values of 0, 1, or 2 wait states. The minimum wait state setting significantly affects the timing of the local memory cycles. This flexibility in wait state selection allows the Am286ZX/LX integrated processor to be used in several different price/performance configurations.

The zero wait state, non-paged DRAM access cycle has very aggressive timing, to achieve zero wait state performance. $\overline{RAS}$ is asserted on the falling edge of the CPU clock during phase two of the CPU status cycle. The Memory Address (MA) is multiplexed from valid row address to column address on the rising edge of CPU clock during the same phase, with $\overline{CAS}$ asserting on the falling edge of the CPU clock during phase one of the command cycle. At higher CPU speeds, this timing will be too aggressive for the less expensive grades of DRAM.

If slower DRAM is in use, one or more wait states can be programmed, moving the Row/Column address multiplexing to the falling edge of the CPU clock during phase one of the first wait cycle. $\overline{CAS}$ is asserted on the falling edge of the CPU clock during phase two of

the first wait cycle. Also, $\overline{RAS}$ is deasserted during phase 1 of the final CPU command cycle to allow for more $\overline{RAS}$ precharge time.

Page mode accesses have been optimized for best performance with low cost DRAM. There are essentially three types of cycles in page mode; $\overline{RAS}$ inactive, page hit, and page miss. DMA cycles are always treated as non-page mode accesses, which inactivates $\overline{RAS}$ after completion of the DMA transfer.

In page mode, most accesses are page hits which are zero wait state cycles. Page hits continue until an access is made outside the DRAM page, or until $\overline{RAS}$ timeout occurs. DRAM page size is dependent on the DRAM size. The effective page size is extended in interleave mode. When interleave is selected with two matching banks installed, the page size is doubled. When four matching banks are installed, the page size is quadrupled.

The DRAM controller incurs wait states when $\overline{RAS}$ is inactive or page misses occur. The $\overline{RAS}$ inactive state occurs on power up, after DMA cycles, after refresh cycles, or in the event that a $\overline{RAS}$ time out has occurred. $\overline{RAS}$ inactive inflicts a two wait state penalty on CPU performance. When the CPU accesses locations outside the current DRAM page, a page miss occurs which inflicts a three wait state penalty on CPU performance. In both of these cases only one CPU cycle is penalized with wait states, per bank accessed, before zero wait state page hit cycles resume.

In most cases, page mode accesses will be used with no wait states. If wait states are programmed to one or two in page mode, the wait states are simply inserted before the final command cycle, with no control signal transitions during the wait states. Control signal transitions remain in the CPU status and final command cycles as in the zero wait state access, effectively stretching the active DRAM command pulse widths.

Local memory wait state settings are programmed in the Memory Wait State register (MWS, 12H). The RAMWS1–RAMWS0 field (bits 5-4) controls the minimum number of wait states for a local memory access. The configuration bits for RAM Wait States default to zero wait states.

Local Memory Wait States

MWS.1 RAMWS1	MWS.0 RAMWS0	Minimum wait states
0	0	0 wait states
0	1	1 wait state
1	0	2 wait states
1	1	Reserved

DRAM SPEED TARGETS/CONSIDERATIONS

Due to the Am286ZX/LX integrated processor's wide variety of possible configurations, the following table is provided to assist in DRAM selection. The speed selections are based on a composite DRAM specification compiled from five popular DRAM manufacturers. These DRAM speed selections are suggestions. The system designer should make his own analysis to verify a specific design.

DRAM Speed Selection.

DRAM Configuration	Clock Speed 12 MHz	16 MHz
Non-paged 0 wait states	80 ns	60 ns
Non-paged 1 wait state	100 ns	80 ns
Non-paged 2 wait states	120 ns	100 ns
Paged 0 wait states*	100 ns**	100 ns

* 0 wait states on page hit
**100 ns for 8 MHz S-bus speed, 120 ns for 6 MHz S-bus speed.

DELAY TAP SELECTION

Most local memory accesses are synchronous, or have timing derived from the CPU clock. DMA, refresh, and external master access to local memory cannot rely on the CPU clock, and thus derive their timing from command strobes and the delay input signals. The delay inputs DL1–DL0 are derived from routing the output DLYOUT though a delay circuit. External master cycles are the most demanding with respect to DMA cycle length, and therefore are the primary consideration in the delay line selection (refer to Figure 3-20). On DMA cycles, DL0 is used to time the change from row to column address on the MA bus at the beginning of the cycle. It is also used for timing the multiplexing of the column address back to row address at the end for the cycle. DL1 is used to time the assertion and deassertion of the $\overline{CAS}$ signals. During staggered refresh DL0 is used to time the delay between the assertion of $\overline{RAS0}$, $\overline{RAS2}$ and $\overline{RAS1}$, $\overline{RAS3}$.

Shadow RAM

The DRAM controller can enhance system performance when executing ROM-based code with ROM shadowing. In system configurations containing 1Mb or more of RAM, the ROMs in the reserved area between 0A0000H and 0FFFFFH can be copied to RAM. This RAM area can then be write protected. Figure 3-21 illustrates this function. The result is a very fast, unalterable ROM image. Shadow RAM can also be used as a protected high memory scratch pad storage. In this application, system software can use a block of RAM to save data or system status and protect it from corruption outside of the standard system address space.

Shadow RAM can be enabled for individual 16-kb blocks, allowing for flexible shadowed ROM configurations. Each 16-kb block can also be individually write protected. The enabling of Shadow RAM blocks is controlled by the Shadow RAM Enable Registers (SER1–SER3, indexes 0CH, 0DH, and 0EH).

Shadow RAM Enable Registers

Register	Index
SER1	0CH
SER2	0DH
SER3	0EH

The configuration bits in the Shadow RAM enable registers default to disabled for all blocks. Each bit enables a specific address range. See Appendix B for full configuration bit information for the Shadow RAM Enable registers.

Figure 3-20 S-Bus Master/DMA Access to Local Memory

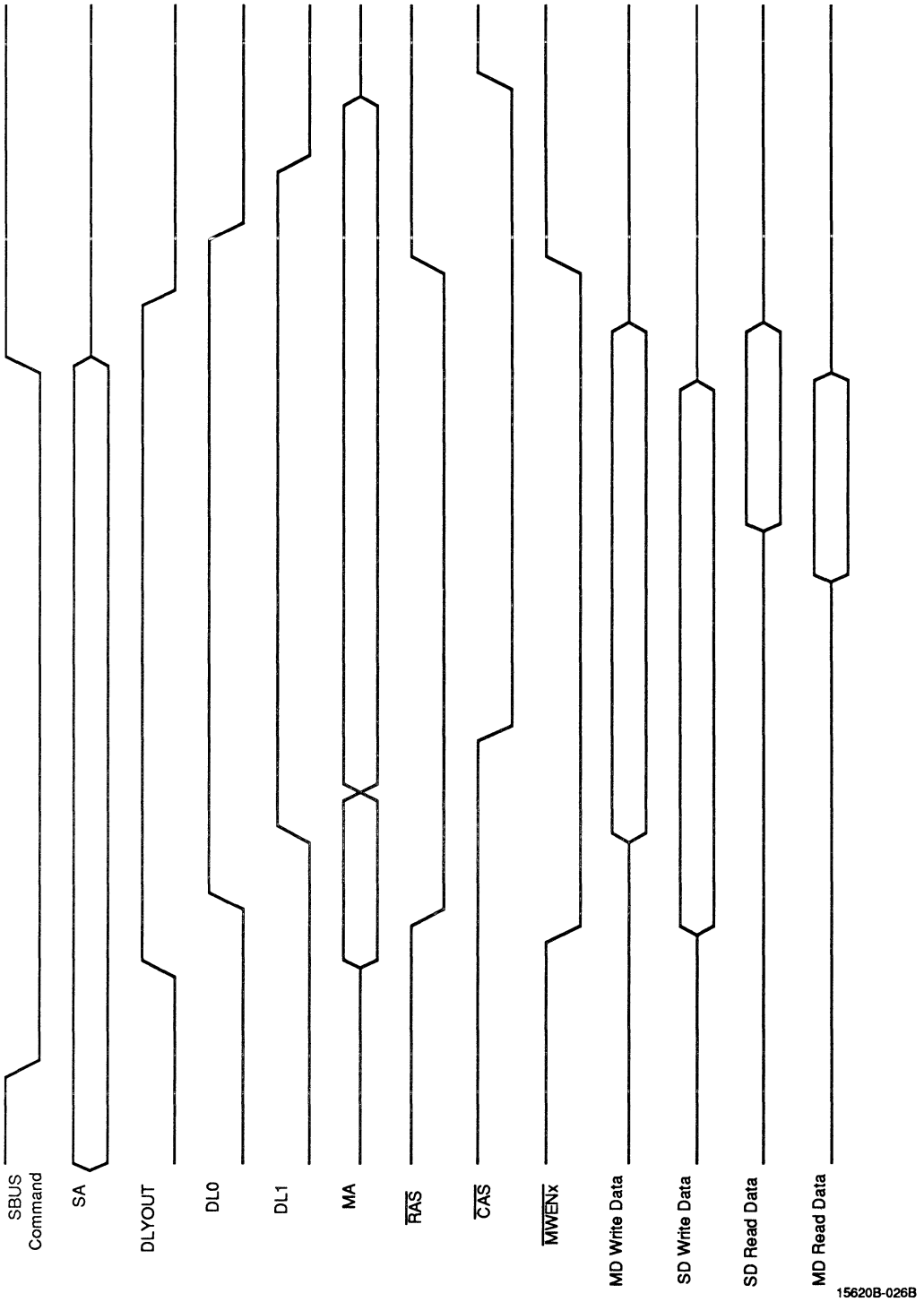
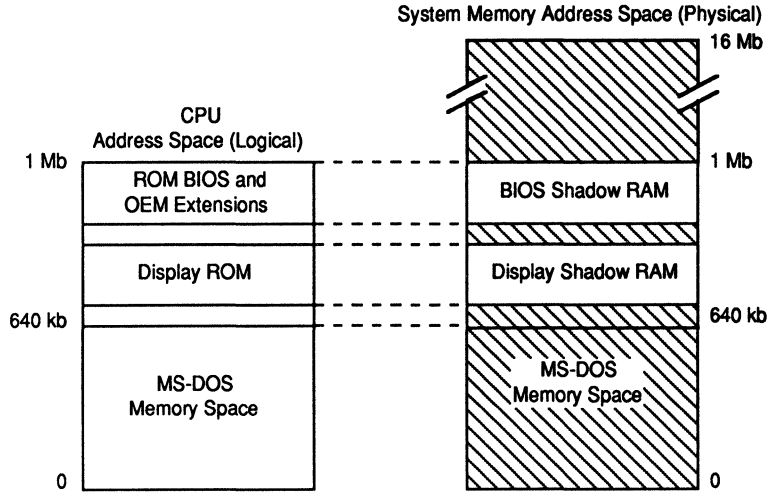


Figure 3-21 Shadowing the ROM Into RAM



Example Shadow RAM Block Enable

SER1.7 SE7	Block Function
1	Enable BC000H to BFFFFH
0	Disabled

The protection of Shadow RAM blocks is enabled or disabled by the Shadow RAM Protection Registers (SPR1, SPR2, and SPR3; indexes 0FH, 10H, and 11H).

Shadow RAM Protection Registers

Register	Index
SPR1	0FH
SPR2	10H
SPR3	11H

The configuration bits in the Shadow Protection Registers default to disabled for all blocks. Each bit enables a specific address range. See Appendix B for full configuration bit information for the Shadow RAM Protection registers.

Example Shadow RAM Block Protection

SPR1.7 SP7	Block Function
1	Enabled BC000H to BFFFFH
0	Disabled

Shadow RAM can be initialized and used by first copying the target ROM image to a standard memory buffer. If the ROM space resides on the Am286ZX/LX integrated processor X Bus, it should then be disabled through the use of the ROM Bank configuration registers (ROBR1, index 13H, and ROBR2, index 14H). The next step is to enable the Shadow RAM with the SER1–SER3 registers. Once the appropriate block of Shadow RAM is enabled, the ROM image can be copied from its memory buffer back to its original

address in Shadow RAM. If the read only property of the ROM image is to be preserved, then the ROM space should be protected with the SPR1–SPR3 registers. Figures 3-22 and 3-23 detail SER and SPR register to address range assignments.

384-kb Relocation

In systems containing only 1 Mb of RAM, when ROM shadowing is not desired, the 384 kb of RAM from 0A0000h and 0FFFFFh can be relocated to above the 1-Mb boundary (refer to Figure 3-24).

DCR.0 REL	Function
0	Disabled
1	Relocated to 100000H

Figure 3-22 Shadow Enable Register to Address Range

SER0	A0000-A3FFF	SER1	A4000-A7FFF	SER2	A8000-ABFFF	SER3	AC000-AFFFF
SER4	B0000-B3FFF	SER5	B4000-B7FFF	SER6	B8000-BBFFF	SER7	BC000-BFFFF
SER8	C0000-C3FFF	SER9	C4000-C7FFF	SER10	C8000-CBFFF	SER11	CC000-CFFFF
SER12	D0000-D3FFF	SER13	D4000-D7FFF	SER14	D8000-DBFFF	SER15	DC000-DFFFF
SER16	E0000-E3FFF	SER17	E4000-E7FFF	SER18	E8000-EBFFF	SER19	EC000-EFFFF
SER20	F0000-F3FFF	SER21	F4000-F7FFF	SER22	F8000-FBFFF	SER23	FC000-FFFFF

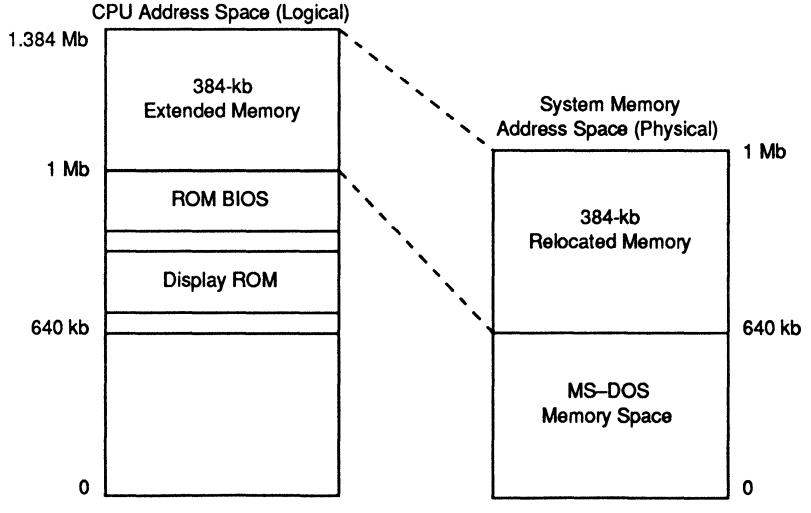
15620B-028B

Figure 3-23 Shadow Protection Register to Address Range

SPR0	A0000-A3FFF	SPR1	A4000-A7FFF	SPR2	A8000-ABFFF	SPR3	AC000-AFFFF
SPR4	B0000-B3FFF	SPR5	B4000-B7FFF	SPR6	B8000-BBFFF	SPR7	BC000-BFFFF
SPR8	C0000-C3FFF	SPR9	C4000-C7FFF	SPR10	C8000-CBFFF	SPR11	CC000-CFFFF
SPR12	D0000-D3FFF	SPR13	D4000-D7FFF	SPR14	D8000-DBFFF	SPR15	DC000-DFFFF
SPR16	E0000-E3FFF	SPR17	E4000-E7FFF	SPR18	E8000-EBFFF	SPR19	EC000-EFFFF
SPR20	F0000-F3FFF	SPR21	F4000-F7FFF	SPR22	F8000-FBFFF	SPR23	FC000-FFFFF

15620B-029B

Figure 3-24 384 kb Relocation



15620B-030B

EMS 4.0 Memory Management

The Am286ZX/LX integrated processor Address Manager resides at the nexus of all internal device address buses. It serves as routing agent for all address traffic. The address manager contains a mapping RAM with which CPU, DMA, and external master addresses may be translated, or mapped, to a reserved physical address within the Am286ZX/LX integrated processor local memory space. The purpose of this mapping process is to enable the system to map all or any 16-kb portion of the 80286 real mode 1-Mb address space to anywhere in the reserved local memory space. Figure 3-25 illustrates the system address architecture.

The most obvious use of this mapping capability is for the implementation of the LIM 4.0 expanded memory specification. This popular method of memory management in the IBM PC environment allows real mode programs to utilize data sets larger than the 640-kb IBM PC address space. The LIM 4.0 specification provides a software specification for the allocation and management of memory segments which reside outside the standard address space. The method typically employs mapping a reserved memory space into the system address space in 16-kb pages. The Am286ZX/LX integrated processor family provides the hardware for robust implementation of such mapping schemes.

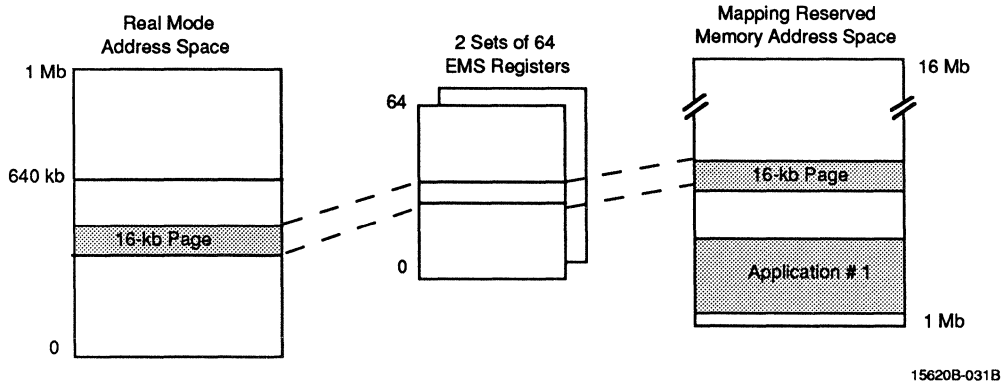
REGISTER STRUCTURE

To establish a logical address space, the Map Space Control registers is used (MSC, index 25h). With this register, a logical space of programmable size is removed from the top of Am286ZX/LX integrated processor local memory physical address space. This provides a pool of reserved memory for the implementation of mapping schemes such as LIM 4.0, which cannot be corrupted by accesses to physical memory. Local memory can be allocated to the logical map space in 512-kb sections with the Map Space Size field in the MSC register. The default space reserved is 0 kb, (MSS4–MSS0 = 0).

Map Space Size Control

MSC.4:0/MSS4:0	Map Space Reserved
0	0 kb
1	512 kb
2	1024 kb
3	1536 kb
....	
31	15782 kb

Figure 3-25 EMS Address Translation



There is one special case provided for in map space reservation. The 384-kb RAM space which exists between 640 kb and 1Mb can be allocated to map logical space by programming the SEMS bit in the MSC configuration register. In this case, the available map space is 384 kb minus the 16-kb blocks reserved for Shadow RAM. The SEMS bit defaults to disabled. It should be noted that the use of SEMS precludes the use of the 384-kb relocation feature.

Map Space Size Control

SEMS MSC.6	Special EMS Map Space
0	Disabled
1	Enabled

Control of the mapping is accomplished via the two sets of 64 address translation registers (ATR's). The two sets provide main and alternate mapping configurations for quick switching between mapping schemes.

There is a main and alternate translation register for each 16-kb memory page within the physical address space 0H to 100000H. Therefore, there are 128 translation registers, the ATRM63–ATRM0 main set, and the ATRA63–ATRA0 alternate set. These registers allow the system software to establish page frames into which logical pages will be mapped. This is shown in Figure 3-27. The lowest order translation register, ATRM0 or ATRA0, corresponds to the lowest 16-kb page in memory, addresses 000000h to 003FFFh. Accordingly, the highest translation register, ATRM63 or ATRA63, corresponds to the highest 16-kb memory page in the 1Mb space, address 0FC000h to 0FFFFFh. A complete table of ATR address range assignments is given in Figure 3-26.

The Address Translation Registers are configured as a register file. Access to the mapping registers is accomplished through a four register window which resides in the Am286ZX/LX integrated processor I/O space. Access to the ATR's is controlled through two mechanisms; window I/O address and register file index. Which ATR's appear in the window is set with the Physical Window field (PW5–PW0) in the Address Translation Control Register (ATCR). The ATCR appears just above the ATR window in the system I/O space. The ATR window and the location of the ATCR is programmable to avoid system resource address conflicts. The second nibble of the I/O address for these registers is programmable via the ARL field in the ATU I/O Register Location register (ARL, index 18H).

Figure 3-26 Address Translation Register to Address Range

ATR0	00000-03FFF	ATR1	04000-07FFF	ATR2	08000-0BFFF	ATR3	0C000-0FFFF
ATR4	10000-13FFF	ATR5	14000-17FFF	ATR6	18000-1BFFF	ATR7	1C000-1FFFF
ATR8	20000-23FFF	ATR9	24000-27FFF	ATR10	28000-2BFFF	ATR11	2C000-2FFFF
ATR12	30000-33FFF	ATR13	34000-37FFF	ATR14	38000-3BFFF	ATR15	3C000-3FFFF
ATR16	40000-43FFF	ATR17	44000-47FFF	ATR18	48000-4BFFF	ATR19	4C000-4FFFF
ATR20	50000-53FFF	ATR21	54000-57FFF	ATR22	58000-5BFFF	ATR23	5C000-5FFFF
ATR24	60000-63FFF	ATR25	64000-67FFF	ATR26	68000-6BFFF	ATR27	6C000-6FFFF
ATR28	70000-73FFF	ATR29	74000-77FFF	ATR30	78000-7BFFF	ATR31	7C000-7FFFF
ATR32	80000-83FFF	ATR33	84000-87FFF	ATR34	88000-8BFFF	ATR35	8C000-8FFFF
ATR36	90000-93FFF	ATR37	94000-97FFF	ATR38	98000-9BFFF	ATR39	9C000-9FFFF
ATR40	A0000-A3FFF	ATR41	A4000-A7FFF	ATR42	A8000-ABFFF	ATR43	AC000-AFFFF
ATR44	B0000-B3FFF	ATR45	B4000-B7FFF	ATR46	B8000-BBFFF	ATR47	BC000-BFFFF
ATR48	C0000-C3FFF	ATR49	C4000-C7FFF	ATR50	C8000-CBFFF	ATR51	CC000-CFFFF
ATR52	D0000-D3FFF	ATR53	D4000-D7FFF	ATR54	D8000-DBFFF	ATR55	DC000-DFFFF
ATR56	E0000-E3FFF	ATR57	E4000-E7FFF	ATR58	E8000-EBFFF	ATR59	EC000-EFFFF
ATR60	F0000-F3FFF	ATR61	F4000-F7FFF	ATR62	F8000-FBFFF	ATR63	FC000-FFFFF

15620B-032B

Address Translation Register and ATCR I/O Addresses

ARL.3:0	ATR(n)	ATR(n+1)	ATR(n+2)	ATR(n+3)	ATCR
0	0208H	4208H	8208H	C208H	0209H
1	0218H	4218H	8218H	C218H	0219H
2	0228H	4228H	8228H	C228H	0229H
3	0238H	4238H	8238H	C238H	0239H
4	0248H	4248H	8248H	C248H	0249H
5	0258H	4258H	8258H	C258H	0259H
6	0268H	4268H	8268H	C268H	0269H
7	0278H	4278H	8278H	C278H	0279H
8	0288H	4288H	8288H	C288H	0289H
9	0298H	4298H	8298H	C298H	0299H
A	02A8H	42A8H	82A8H	C2A8H	02A9H
B	02B8H	42B8H	82B8H	C2B8H	02B9H
C	02C8H	42C8H	82C8H	C2C8H	02C9H
D	02D8H	42D8H	82D8H	C2D8H	02D9H
E	02E8H	42E8H	82E8H	C2E8H	02E9H
F	02F8H	42F8H	82F8H	C2F8H	02F9H

Note: n = ATCR index (bits 5–0)

ATR Physical Window Index

ATCR.5:0 PW5:0	Function
0 to 63	Window base for ATR

It should be noted that the ATR registers are accessed only as 16-bit I/O devices, and the ATCR can be accessed only as an 8-bit I/O device. This resolves the apparent overlap of ATR0 and ATCR. The 16-bit registers respond only to 16-bit accesses.

The translation register window access is enabled with the AIE bit (ARL, bit 4). If the Address translation function is not to be used, AIE can be left cleared, and the Address Translation Registers and Address Translation Control Register will not appear in the system I/O space. The AIE bit is disabled by default. Therefore, the AIE bit must be set for any access to the translation register file.

ATR I/O Enable

ARL.4 AIE	Function
0	ATR access disabled
1	ATR access enabled

Access to the ATR registers is accomplished by establishing the ATR window/ATRC I/O address, enabling I/O via AIE, and writing an index into the register array to the ATCR. Once this index is written, four ATR registers can be written and read without changing the index. When these four registers are programmed, a new index can be written to the ATCR and four more registers initialized or updated. If an index is written to the ATCR such that an ATR register in the access window would be beyond the end of the register, file access will wrap to the bottom ATR registers.

When the main set is programmed, the ASE bit in the ATCR can be set to enable access to the alternate set of translation registers. With ASE set, the access to the main set of ATR registers is disabled as is any main set address translation. ASE is disabled by default.

Alternate Set Enable

ATCR.6 ASE	Function
0	Main set enabled
1	Alternate set enabled

The alternate set programming procedure is the same as with the main set. When this is complete the ASE bit may be cleared, if the activation of the main set is desired, or left with the alternate set enabled.

To enable translation the Global Translation Enable bit (GTE) in the ATCR is set. All access to the first 1 Mb of system address space will now be checked for translation, and translated if the appropriate translation register has been programmed for translation.

Global Translation Enable

ATCR.7 GTE	Function
0	Address mapping disabled
1	Address mapping enabled

Each Address Translation Register (ATR) consists of eleven bits. A ten bit Physical Page Data field determines where the 16-kb address block will be mapped in the logical address space reserved by programming the Map Space Control register (MSC, index 25h).

Physical Page Translation Data

ATRM.9:0 or ATRA.9:0 PPD9:0	Function
0–3FFH	Selects 16-kb page in 16-Mb address space

These ten bits replace the upper ten bits of the non-translated address.

Translation Example

CPU real mode address	00002H
Map for ATRM0=02H	010
Translated address	080002H

Also in each Address Translation Register there is a Page Enable bit (PGE). Setting this bit enables translation for the 16-kb page concerned. This provides a simple way to disable portions of a mapping scheme. Since the PGE bit defaults to an unknown state, all the translation registers need to be programmed during ATR initialization. Once an ATR is initialized with the PGE bit set, translation will occur for that ATR's address range even if it is not currently in the ATR register access window.

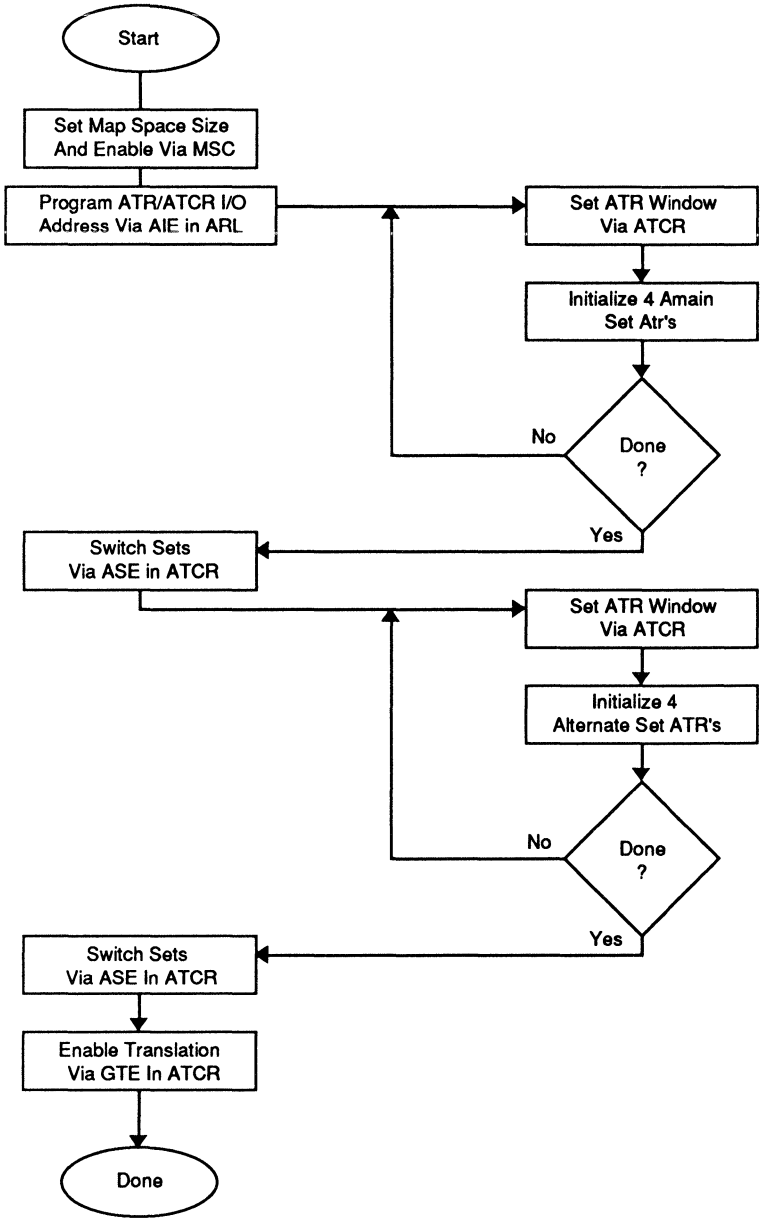
Page Translation Enable

ATR.10 PGE	Function
0	Page translation disabled
1	Page translation enabled

PROGRAMMING

In summary, the MSC is programmed to reserve a logical address space from the local memory array, which can be mapped into any of the 64 possible page frames. The ARL is programmed to enable translation register and control register access at a particular I/O address range. The ATCR is programmed to the appropriate physical window into the translation register file. Since the ASE (Alternate Set Enable) bit was not programmed, the translation register window points to the main set. The mapping values for each 16-kb page frame chosen in the 1-Mb real mode address space can now be set in each ATR. Four consecutive translation registers may be set without updating the ATCR to point to a new four register window. A flow chart detailing this process can be found in Figure 3-27.

Figure 3-27 Address Translation Initialization Flow



15620B-032B

CHAPTER 4

AT SUPPORT LOGIC

NMI AND PORT B LOGIC

Port B Register

Port B is an AT standard miscellaneous feature control register which is located at I/O address 061H. The lower four bits of the 8-bit register are read/write control bits which enable or disable NMI check condition sources and sound generation features. The top, or most significant, four bits are read-only bits which return status and diagnostic information.

Port B Register

Bit	Name	Function
7	PCHK	Memory parity check
6	ICHK	I/O channel check
5	T2OUT	Timer 2 output status
4	RFD	Refresh detect
3	EIO	I/O channel check enable
2	EPR	Memory parity enable
1	SPK	Speaker data
0	T2G	Timer 2 gate control

The PCHK bit in the Port B register is a read-only status bit which indicates whether or not a parity error has occurred within the system RAM (RAM residing on the M Bus). This bit is normally polled in the NMI interrupt service routine to determine if the memory subsystem was the cause of the NMI. This bit defaults to 0.

Parity Check Status Bit

PCHK PORTB.7	Indication
0	No memory parity errors detected
1	Parity error detected

The ICHK bit in the Port B register is a read-only status bit which indicates whether or not the I/O channel check ($\overline{IOCHK}$) pin on the S Bus has been activated by an S-bus peripheral. This bit is normally polled in the NMI interrupt service routine to determine if the $\overline{IOCHK}$ was the cause of the NMI. This bit defaults to 0.

I/O Channel Check Status Bit

ICHK PORTB.6	Indication
0	No I/O channel check detected
1	I/O channel check has occurred

The T2OUT bit in the Port B register is a read-only status bit which can be used by diagnostic routines to verify proper operation of channel 2 of the 82C54 timer, which

sources speaker tone generation. The T2OUT bit reflects the current value of the internal 82C54's Timer 2 Output.

The RFD bit in the Port B register is a read-only status bit which indicates whether or not system refresh requests are being generated by channel 1 of the internal 82C54 timer. The RFD bit is a divide by two of the internal refresh request, so the bit toggles every refresh cycle. Therefore, the value of the bit contains no information, but the fact that the RFD bit is transitioning back and forth indicates that refresh requests are being generated. This bit is normally polled by the BIOS power-up self-test code to verify that system refresh has been started correctly.

The EIO bit in the Port B register is a read/write control bit which enables and disables I/O channel checks, which generate NMIs to the 80C286 CPU. The bit has an active Low sense: a 0 enables I/O channel checks, and a 1 disables them. When an I/O channel check occurs, the EIO bit must be set and then cleared in order to clear out the latched error condition which is visible in the ICHK bit of the same Port B register. The default value of this bit is 0 (I/O checks enabled).

Enable I/O Channel Check

EIO PORTB.3	Function
0	I/O channel check enabled
1	I/O channel check disabled, cleared

The EPR bit in the Port B register is a read/write control bit which enables and disables memory parity checks, which generate NMIs to the 80C286 CPU. The bit has an active Low sense: a 0 enables parity checks, and a 1 disables them. When a parity check occurs, the EPR bit must be set and then cleared in order to clear out the latched error condition which is visible in PCHK bit of the same Port B register. The default value of this bit is 0 (parity checks enabled).

Enable Parity Check

EPR PORTB.2	Function
0	Parity check enabled
1	Parity check disabled, status cleared

The SPK bit in the Port B register is a read/write control bit which allows direct programmed I/O control over the speaker data line. The default value is 0, which will force the speaker data line inactive.

Speaker Data Line Control

SPK PORTB.1	Function
0	Speaker data forced inactive
1	Speaker data = Timer 2 output

The T2G bit in the Port B register is a read/write control bit which allows direct programmed I/O control over the internal 82C54's channel 2 GATE input. The value programmed into this bit is the value presented at the channel 2 gate input. The GATE input of a 82C54 enables counting when High and disables counting when Low. This bit can be used to modulate the tone generated by timer channel 2. The default value for T2G is 0, which disables counting for channel 2 of the internal 82C54.

Timer 2 Gate Control

T2G PORTB.0	Function
0	Disable timer 2 counting
1	Enable timer 2 counting

NMI Enable Register

Both the memory parity check and I/O channel check are possible sources for the generation of a NMI to the internal 80C286 CPU. Both of these sources can be individually disabled. However, there is also a master NMI enable function provided which can inhibit any NMI's from reaching the CPU, regardless of the state of the individual source enables. This master NMI control is located in a single bit register at I/O address 070H. The NMI enable bit in this register is a write-only bit, and it has an active Low sense. The register is also located at the same I/O address as the RTC index register. Because the RTC index register is only 7 bits wide, and is also write only, there is no conflict between the two registers. The default value for the NMI enable bit is 1, which inhibits NMI generation.

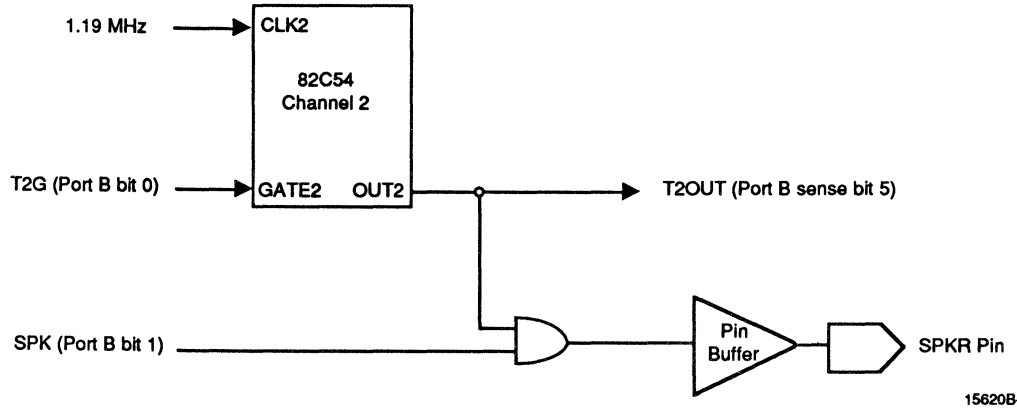
NMI Enable Register (070H)

DNMI NMI.7	Function
0	Enable NMI generation
1	Master disable for NMI

SPEAKER INTERFACE

The AT standard tone generation interface for the system speaker is implemented in the Am286ZX/LX integrated processor, as shown in Figure 4-1. There are two data paths to the SPKR pin of the device. The first path is sourced by the output of channel 2 of the internal 82C54 counter/timer. The counter/timer can be programmed in various ways to generate a waveform at the output, OUT2. Also, the gate input of timer channel 2 is controlled by the T2G bit in Port B. The timer gate can be used to inhibit tone generation by the timer channel.

Figure 4-1 Speaker/Tone Generation Block Diagram



The second path is sourced directly by the SPK bit in Port B. This bit can be manipulated by the CPU to generate a wide range of digital waveforms at the SPKR pin. Note that the SPK bit is logically ANDed with the timer output.

DEVICE AND CPU RESET

Device Hardware Reset

The Am286ZX/LX integrated processor requires an external hardware reset in order to correctly initialize internal logic after system power-up. AT power supplies have a Power Good output signal which is used as an active Low asynchronous reset input (PWRGOOD) for the device. The input buffer for the PWRGOOD pin is a Schmitt trigger, for tolerance of slow rise and fall times on the signal. PWRGOOD is internally synchronized to the 80C286 CPU clock to provide the internal hardware reset.

A buffered version of the internal hardware reset is available on the $\overline{\text{RESET}}$ output of the device. This output is normally used to reset the AT keyboard controller, but it has no keyboard specific function. $\overline{\text{RESET}}$ can be used as a general purpose, CPU synchronous hardware reset signal.

CPU-Only Reset Control

Besides the device hardware reset, the internal 80C286 processor has several other possible reset sources. These other sources generate only CPU resets. The rest of the device state remains unchanged during the CPU-only reset.

KEYBOARD CONTROLLER RESET (RC)

The $\overline{\text{RC}}$ pin, which is typically connected to an output of the 8042 keyboard controller, is a level sensitive CPU-only reset input. $\overline{\text{RC}}$ is an active Low input which will force the CPU reset active as long as it is active. $\overline{\text{RC}}$ is an asynchronous input which is internally synchronized to the CPU clock.

CPU SHUTDOWN DETECT

If the CPU reaches a state where it cannot continue to execute because of faults and error conditions, it will issue a status code indicating shutdown, and the CPU will halt operation with no means of continuing except for a reset. If this shutdown status is detected, a 16-clock minimum pulse width reset is automatically sent to the 80C286 CPU.

Some system software and application software packages actually use this feature to force a CPU reset faster than a typical AT keyboard controller can generate one via the $\overline{\text{RC}}$ input.

FAST RESET SUPPORT

The Am286ZX/LX integrated processor provides a means for quickly generating a minimum pulse width reset to the CPU under program control. An industry standard control port at I/O address 092H contains the Fast Reset bit, FRST. The FRST bit is an edge sensitive control bit, with a 0 to 1 transition in the bit triggering a CPU reset. A static 0 or 1 bit state will not trigger a reset. The bit must be reset back to 0 after a fast reset before another fast reset can be triggered. The default value for the bit is 0.

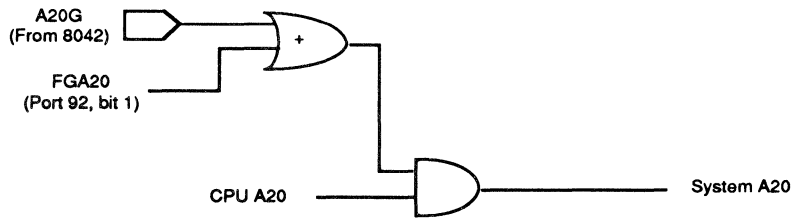
Port 92h Fast Reset

FRST P92H.0	Function
0	Normal operation
0→1	Reset CPU
1	Normal, after fast reset

A20 ADDRESS CONTROL

With the 80C286 processor, full real mode address compatibility requires that address "rollover" at the 1-Mb address boundary be handled the same as the early 8088-based PCs did. This requires system address line 20 to have the capability of being forced to a 0 during real mode execution. The Am286ZX/LX integrated processor provides two different methods for controlling the A20 system address line: AT standard A20 control and Fast Gate A20 support. See Figure 4-2 for a logic diagram which illustrates the Gate A20 control function implemented in the Am286ZX/LX integrated processor.

Figure 4-2 A20 Gating Logic



AT Standard A20 Control

The A20G input pin on the device is normally connected to an output of the AT keyboard controller. A logic High on this input will force the "pass through" of the CPU's A20 onto the internal system address bus. A logic Low on this input will force the system address bus A20 line Low, as long as the Fast Gate A20 bit is not being utilized.

Fast Gate A20 Support

The Am286ZX/LX processor provides a high-performance method for controlling the system A20 line, independent of the relatively slow AT keyboard controller. An industry standard control port at I/O address 092H contains the Fast Gate A20 bit, FGA20.

A logic High written to this bit will force the "pass through" of the CPU's A20 onto the internal system address bus. A logic Low written to this bit will force the system address bus A20 line Low, as long as the AT keyboard controller A20G input is not being utilized. The default value for the bit is 0, allowing the keyboard controller to control A20 gating via the A20G input.

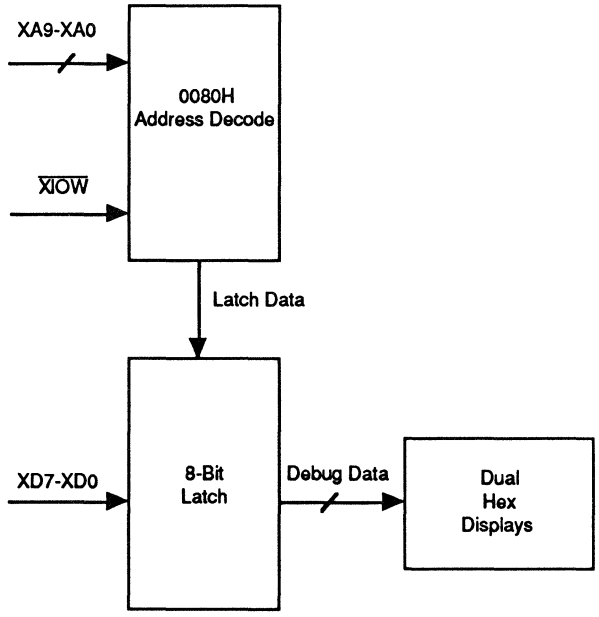
Port 92h Fast Gate A20

FGA20 P92H.1	Function
0	Force system A20 Low
1	Pass through CPU A20

MANUFACTURING DEBUG PORT

The I/O location at 0080H, which corresponds to the DMA page register location, is often used as a debug port location to indicate progress though test code in the system BIOS. The Am286ZX/LX integrated processor supports the use of an address 0080H debug port by echoing internal writes to the DMA page register at address 0080H out onto the X Bus. Therefore, a simple address decoder and data latch circuit on the X Bus can monitor progress through test code which does I/O outputs to address 0080H. An example debug port circuit block diagram with X-bus hookup is shown in Figure 4-3.

Figure 4-3 Manufacturing Debug Port Example



15620B-035B

CHAPTER 5

EXTERNAL BUS INTERFACES

AT SYSTEM BUS (S BUS) CONNECTION

The AT system bus provides a standard interface for expanding a system's capabilities with expansion cards designed for ISA bus signals and cycles. The Am286ZX/LX integrated processor provides a full, non-multiplexed AT system bus, called the S Bus, with all the operative signals and timing required for ISA bus compatibility.

Configuration

There are two main configurations used for S-bus connection with the device: direct connection and buffered. Direct connection is the simplest, most cost effective configuration, but it may not have the drive capability needed for the intended system target. Buffered connection requires an increase in component count and cost, but it insures greater drive capability for the bus.

BUS LOADING ANALYSIS

Because the required bus drive capability is the prime factor which drives the decision between the direct and buffered configurations, a detailed analysis of the target system environment should be done. The S-bus drive capability of the Am286ZX/LX device, as specified in the data sheet, should be compared against the drive current and load capacitance requirements of the design.

The bus drive requirements for a system depend on several aspects of the design. If ISA standard slots are to be provided in the system, a definition of the loading characteristics of an "average" or "worst case" expansion card must be determined. If a standard system peripheral such as a VGA video controller is included on the S Bus of the motherboard, its loading must be determined. An on-board peripheral will have a different load characteristic than the same peripheral on an add-in board, and steps can be taken in the bus interface design for the peripheral in order to minimize S-bus loading.

BUS CONNECTION

The Bus Connection (BCON) bit in the Bus Control Register (bit 0 of BCR, index 09H) enables the S-bus interface. No data transfers will occur on the S Bus if this bit is not set. See the full description of the BCON bit in the Enhanced Bus Controller section of Chapter 3.

STRONG PULL-UP CONTROL

The Am286ZX/LX integrated processor provides strong pull-ups on the following pins: $\overline{\text{MEMCS16}}$, $\overline{\text{IOCS16}}$, $\overline{\text{IOCHK}}$, IOCHRDY, $\overline{\text{MASTER}}$, and $\overline{\text{REFRESH}}$. If the effective resistance values offered by the pull-up devices on the silicon are not applicable for the specific application, then the pull-ups can be disabled with a configuration bit.

The KSPUL (Kill Strong Pull-up) configuration bit in the Bus Control Register (BCR, index 09H) disconnects the strong pull-ups on the device, allowing external pull-ups to be

implemented in a design. A logic 1 written to this bit will disable the strong pull-ups. The default for the KSPUL bit is 0, which enables the strong pull-ups.

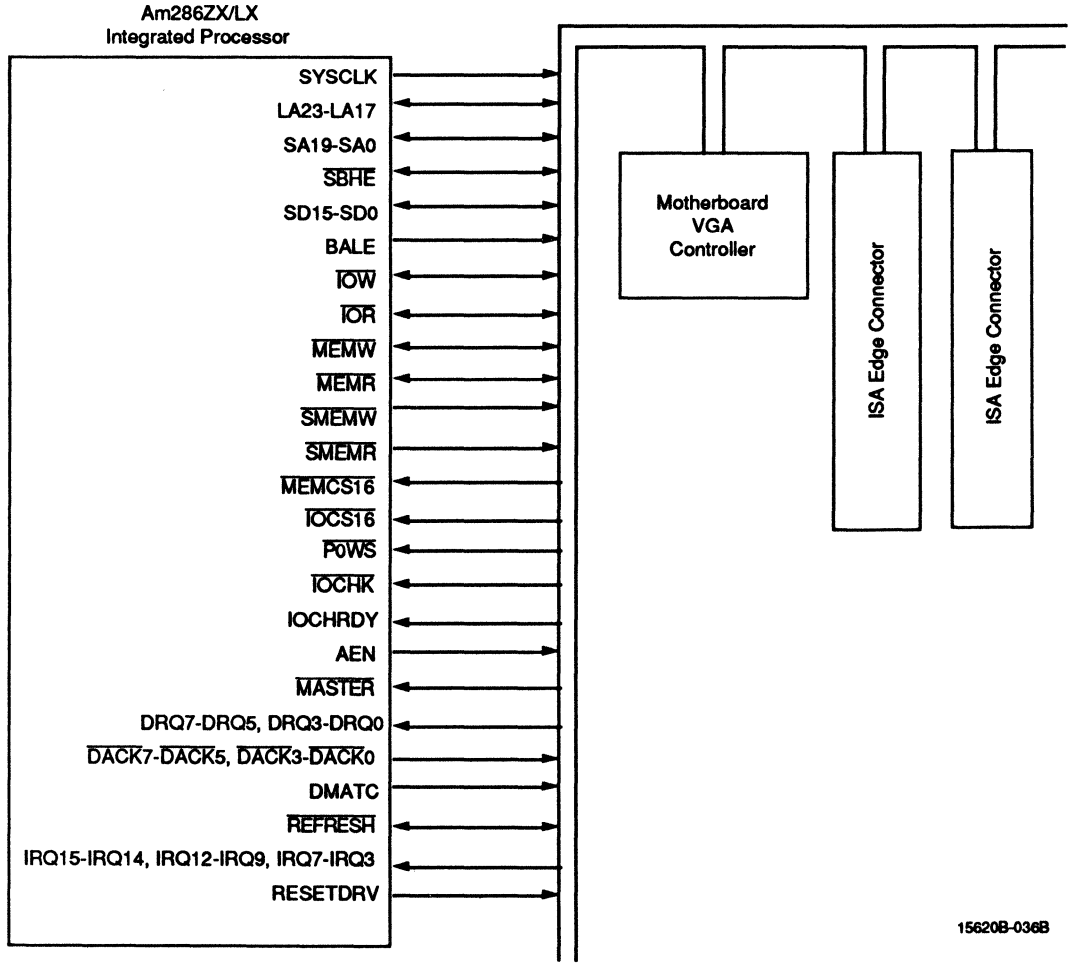
Strong Pull-up Configuration

KSPUL BCR.7	Function
0	Strong pull-ups active
1	Strong pull-ups disabled

Direct Connection

The direct connection configuration is a straightforward point-to-point connection of the S-bus interface pins to the motherboard peripherals, and to any ISA edge connectors for expansion capability. An example of a direct connection from the S Bus to the rest of a system is shown in Figure 5-1.

Figure 5-1 S-Bus Direct Connection Example



Buffered Connection

The buffered connection configuration places buffers between the S-bus interface pins of the device and the AT system bus. This is done for the output pins that need a drive capability in excess of that specified for the pin. Lightly loaded lines such as the $\overline{DACK}$ lines do not need buffering. Figure 5-2 shows an example of a buffered connection configuration.

The table below shows the proper buffer control signals for the S-bus interface pins that could require buffering.

Buffered S-Bus Drive Control

Signal	Enable	Direction
LA23–LA17	Always	$\overline{MASTER}$
SA19–SA0	$\overline{SBENA}$	($\overline{MASTER}$ + $\overline{REFRESH}$)
$\overline{SBHE}$	$\overline{SBENA}$	($\overline{MASTER}$ + $\overline{REFRESH}$)
$\overline{MEMR}$	$\overline{SBENA}$	($\overline{MASTER}$ + $\overline{REFRESH}$)
$\overline{MEMW}$	$\overline{SBENA}$	($\overline{MASTER}$ + $\overline{REFRESH}$)
$\overline{IOR}$	$\overline{SBENA}$	($\overline{MASTER}$ + $\overline{REFRESH}$)
$\overline{IOW}$	$\overline{SBENA}$	($\overline{MASTER}$ + $\overline{REFRESH}$)
SD15–SD8	$\overline{SBENA}$	HDATDIR
SD7–SD0	$\overline{SBENA}$	LDATDIR
SYSCLK	Always	N/A
BALE	Always	N/A
ResetDrv	Always	N/A
$\overline{SMEMR}$	Always	N/A
$\overline{SMEMW}$	Always	N/A
AEN	Always	N/A
DMATC	Always	N/A

S-BUS CYCLES

CPU driven transfer cycles on the S Bus are controlled by the ATSM, as described in the Enhanced Bus Controller section of Chapter 3. The other possible masters for S-bus cycles are the internal DMA controllers, the refresh generator, and the external bus masters.

CPU Cycles

Figure 5-3 shows an S-bus I/O Write cycle to a 16-bit peripheral. The first cycle shows the standard one command delay, and one wait state cycle. Note the assertion of $\overline{IOCS16}$ by the peripheral to indicate 16-bit bus size. The second cycle shows the extension of the cycle by two wait states, by driving the IOCHRDY input inactive.

Figure 5-4 shows a 16-bit I/O Write cycle to an 8-bit peripheral ($\overline{IOCS16}$ is not asserted). This necessitates a conversion cycle, which first writes the Low byte, then routes the High byte to SD7–SD0, toggles the SA0 line, and generates another I/O Write pulse.

Figure 5-2 S-Bus Buffered Connection Example

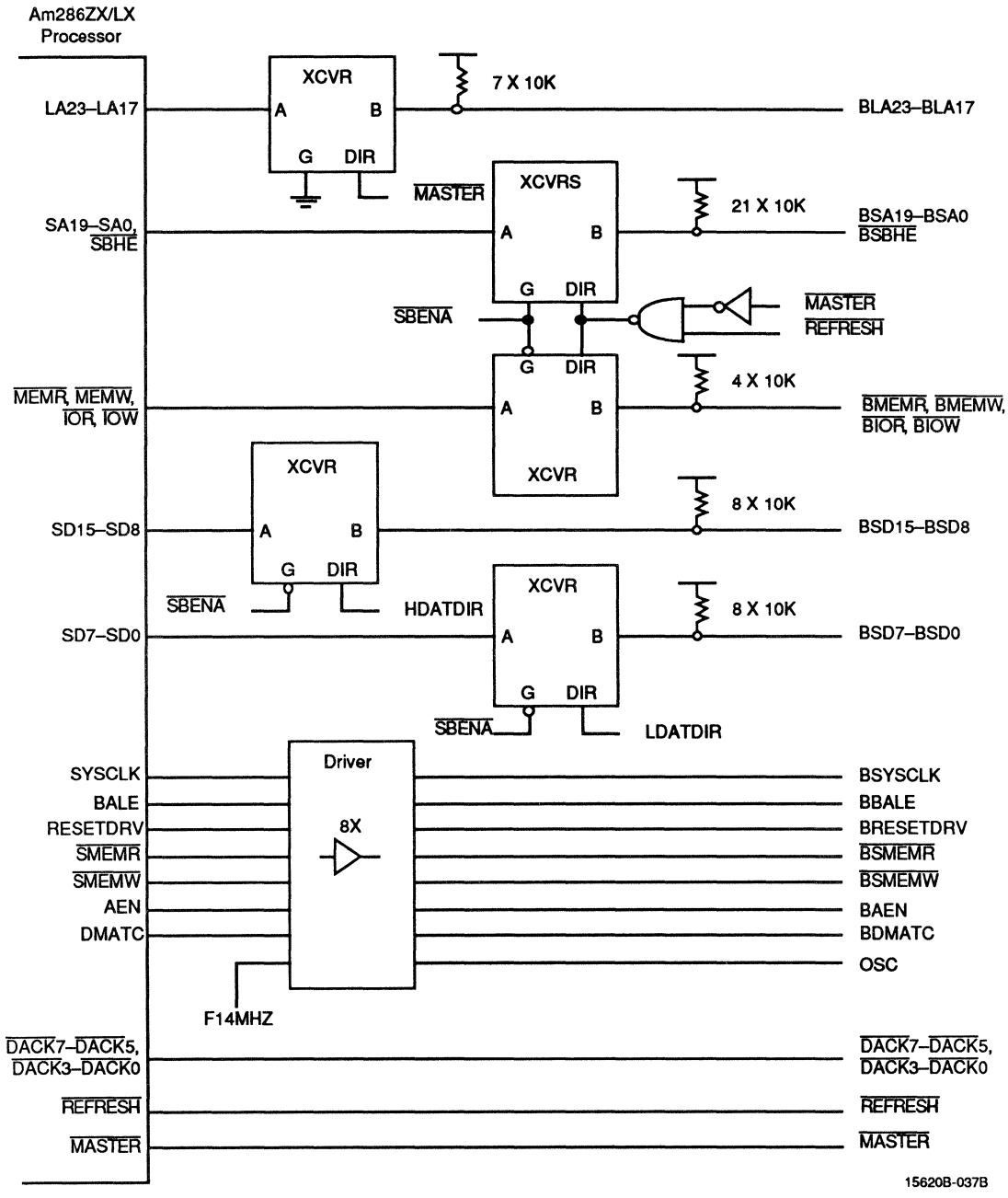
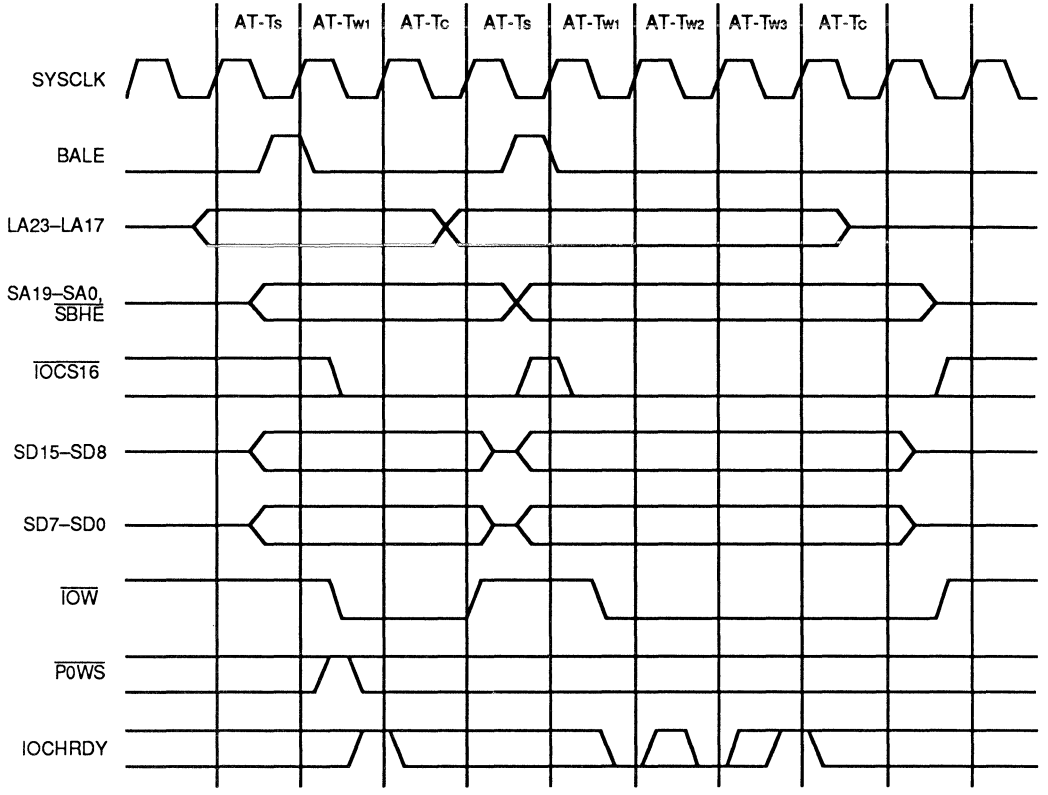


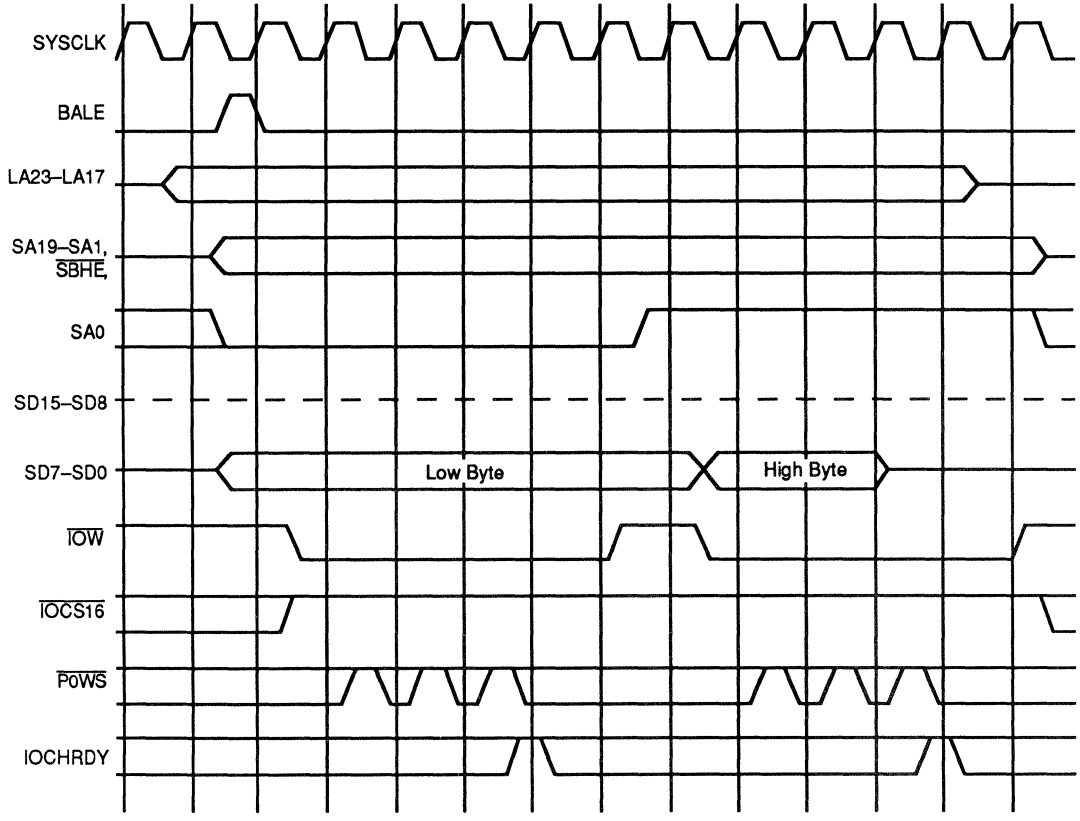
Figure 5-3 S-Bus I/O Write, 16-Bit Peripheral



Note: AT system bus I/O write to 16-bit device: one command delay, one wait state followed by cycle extended two wait states to one command delay, three wait states by IOCHRDY unasserted.

15620B-038B

Figure 5-4 S-Bus I/O Write, 8-Bit Peripheral



Note: AT system bus I/O write 16- to 8-bit conversion. 15620B-039B

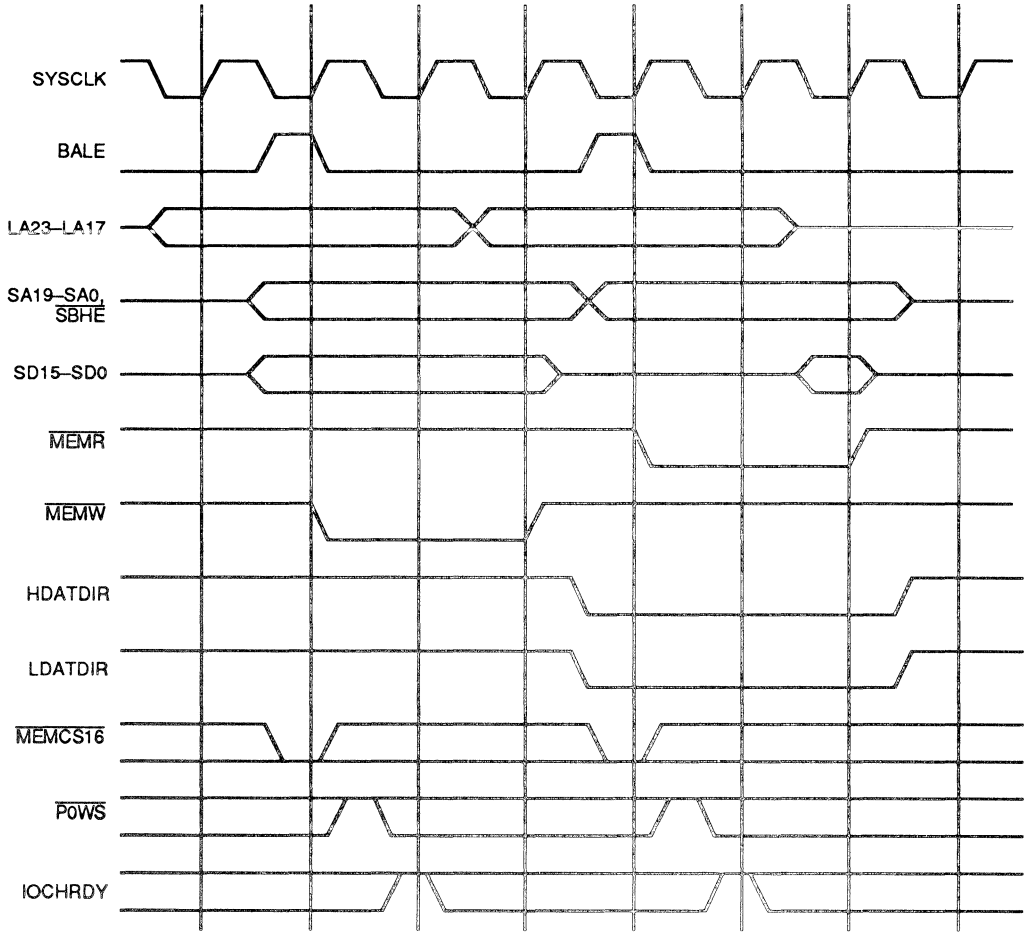
Figure 5-5 shows S-bus Memory cycles to a 16-bit memory device. The first cycle demonstrates a memory write with the standard 0 command delay, one wait state cycle. The second cycle demonstrates a memory read cycle. Note the sampling of the MEMCS16 input at the trailing edge of BALE to indicate 16-bit bus size.

Figure 5-6 shows an S-bus Memory Write cycle which is terminated early with the POWS input. The input is sampled active (Low) at the middle of the first AT command cycle (AT-TC) to force a zero wait state cycle.

DMA Cycle

Figure 5-7 shows an S-bus DMA transfer cycle. The DMA Clock is normally half the frequency of SYSCLK. The active AEN pin prevents I/O devices from improperly activating off the combination of SA and IOR or IOW. The BALE forced active opens the latches on memory cards so they can decode the LA lines for the DMA transfer. Note that two S-bus command lines are active simultaneously for a "fly by" DMA transfer. DMA cycle wait states can be programmed as directed in the DMA Controller section of Chapter 2.

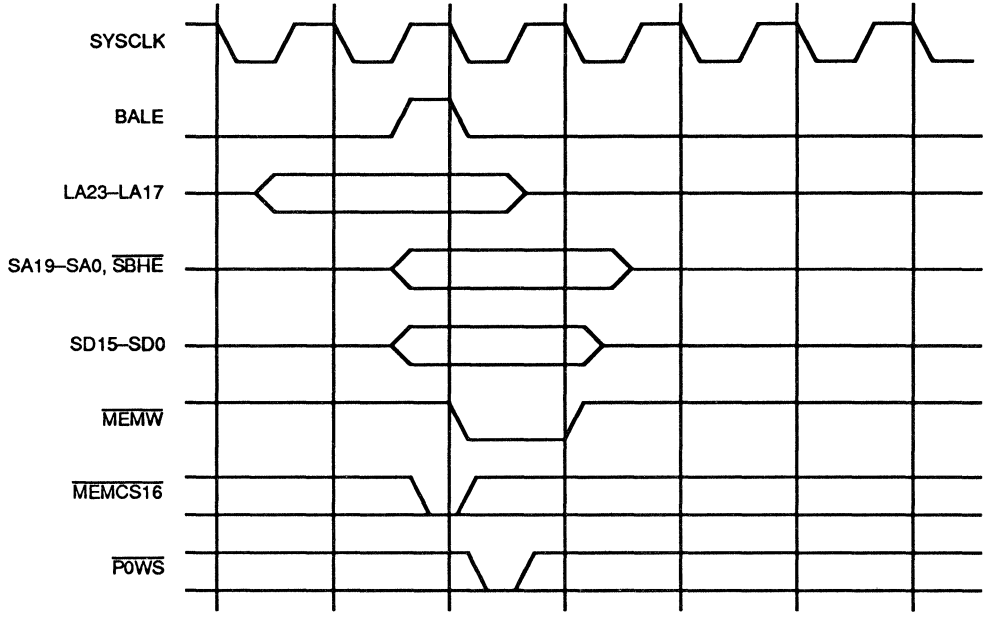
Figure 5-5 S-Bus Memory Cycles, 16 Bit



Note: AT System Bus Memory Write and Read Zero Command Delay, One Wait State 16-bit Device

15620B-040B

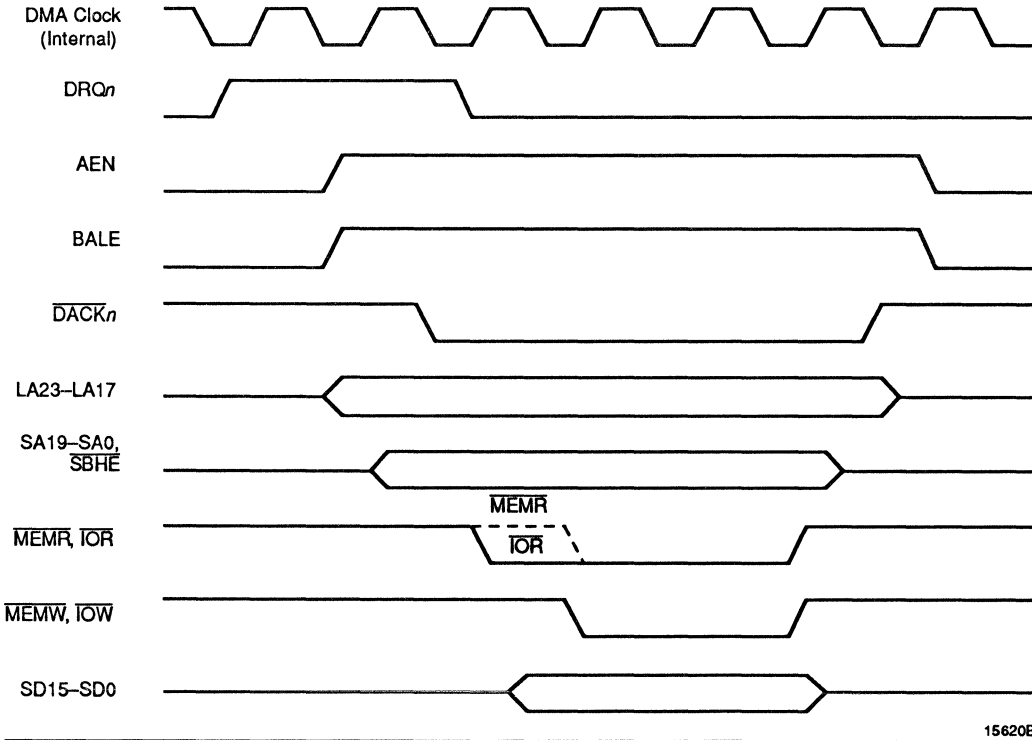
Figure 5-6 S-Bus Memory Write, Zero Wait State



Note: S-bus Memory Write Cycle; Zero Command Delay terminated early with active POWS.

15620B-041B

Figure 5-7 S-Bus DMA Cycle



15620B-042B

Refresh Cycle

Figure 5-8 shows an S-bus refresh cycle. The $\overline{\text{REFRESH}}$ pin going active indicates the start of a refresh cycle. The LA lines are driven from a value in the DMA page register, which is normally set to 00H. The SA lines are driven by the value in the refresh row counter. After a one clock delay allowing setup time for the address, the $\overline{\text{MEMR}}$ and $\overline{\text{SMEMR}}$ pins pulse active to trigger the $\overline{\text{RAS}}$ cycle for DRAMs on S-bus memory cards.

Note that if slow refresh is enabled, the frequency of refresh cycles on the S Bus will also be divided by 8.

Bus Master Cycle

Figure 5-9 shows a typical S-bus bus master access cycle, in this case a memory write. The cycle timing is under the control of the external bus master, not the Am286ZX/LX integrated processor. However, the device will correctly respond to read and write cycles to resources under its control.

The transfer cycle begins with the bus master asserting a DRQ line with the DMA channel programmed to cascade mode. When the $\overline{\text{DACK}}$ response comes back, the bus master asserts the $\overline{\text{MASTER}}$ line and then waits for one SYSCLK. Next, the master drives the address lines, and activates one of the command lines a SYSCLK cycle later. The data transfer occurs, with the address and data driven during the $\overline{\text{MEMW}}$ command pulse, and then the bus master releases the bus, deactivating the DRQ and $\overline{\text{MASTER}}$ lines.

Figure 5-8 S-Bus Refresh Cycle

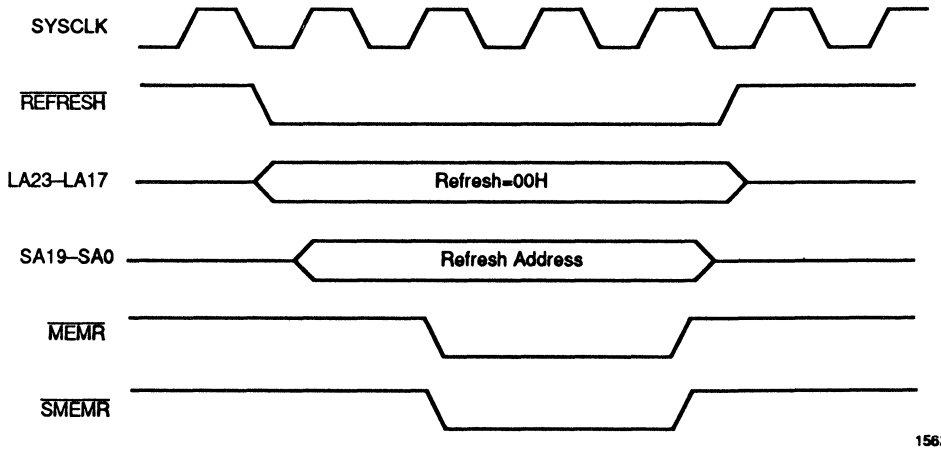
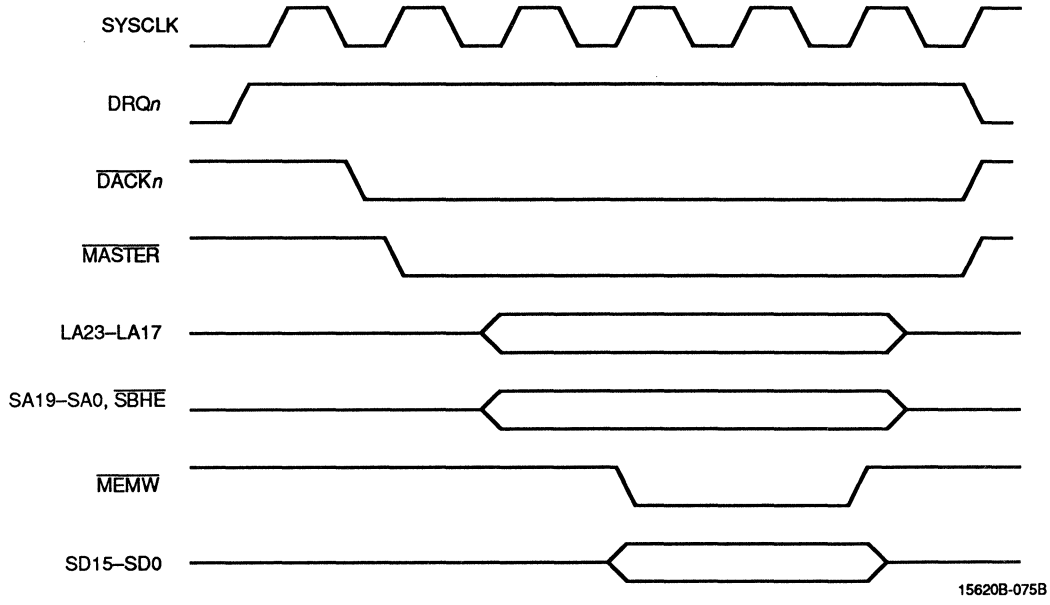


Figure 5-9 S-Bus External Bus Master Cycle (Memory Write)



CHAPTER 6

PERIPHERAL EXPANSION BUS (X BUS)

KEYBOARD CONTROLLER INTERFACE

Mode Selection

The Am286ZX/LX integrated processor provides a direct interface for an AT keyboard controller (8042/8742) or for a XT keyboard. The interface is chosen by an XTE configuration bit in the Peripheral Selection Register (PSR, index 03H). The default for the XTE bit is 0, which enables the AT controller interface and disables the XT keyboard interface. Note that selecting the XT keyboard mode changes the definition of two device pins.

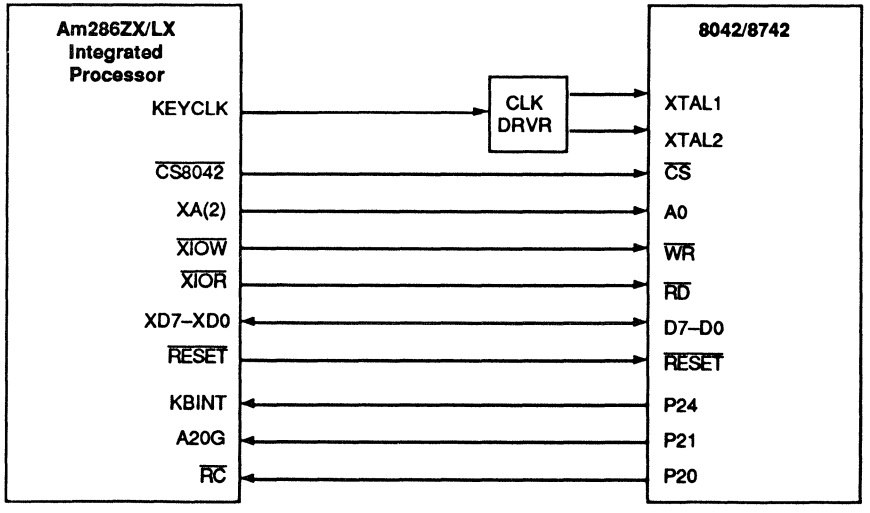
Keyboard Mode Selection

XTE PSR.2	Selection
0	AT 8042 interface
1	XT direct interface

AT Mode Connection

With the AT keyboard interface enabled, a 8042/8742 style keyboard controller may be directly connected to the Am286ZX/LX device as an X-bus peripheral. Figure 6-1 shows the connection diagram for an AT keyboard interface.

Figure 6-1 AT Keyboard Controller Interface



15620B-044B

Address, data, and read/write controls are provided by the corresponding X-bus signals. The other controller specific signals required for direct connection are also provided: the chip select ($\overline{CS8942}$), reset ($\overline{RESET}$), interrupt (KBINT), A20 gate signal (A20G), and CPU reset ($\overline{RC}$).

The keyboard clock output of the Am286ZX/LX integrated processor (KEYCLK) can be used to provide the clock signal for the controller. The clock driver circuit is dependent on the actual controller device used. A clock drive method which meets the manufacturer's specifications should be used.

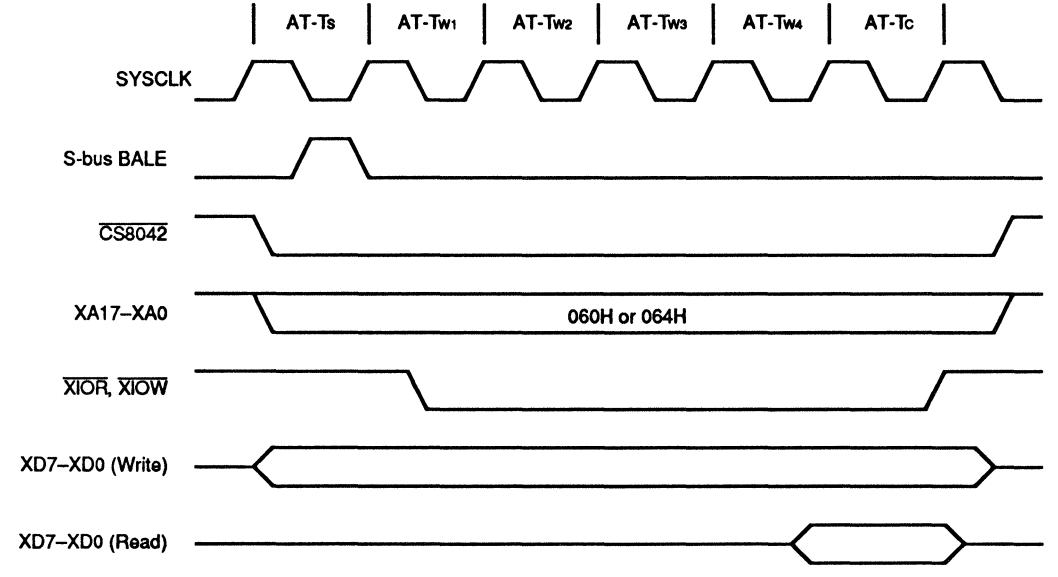
If the controller is a CMOS device which consumes significantly less power at slower clock rates, the KEYCLK output can be programmed with a divide by 16, in low-power applications. Use of the KEYCLK output for the keyboard controller's clock is not absolutely necessary. The keyboard controller can be clocked independently, and the KEYCLK output can be used as a general purpose programmable clock output.

AT Mode Bus Cycles

Data transfer cycles to and from the AT keyboard controller (I/O ports 60H and 64H) are routed to the X Bus, and are always controlled by the ATSM. Therefore, the access cycle will generally be timed the same as an S-bus 8-bit I/O cycle. Figure 6-2 shows a typical data transfer cycle to or from the AT keyboard controller. The actual command delays and wait states used are dependent upon the 8-bit I/O cycle setting in the Command Delay Register (CDR, index 04H) and the Wait State Register (WSR, index 05H).

An extra CPU clock cycle of data recovery time is always appended to the end of a AT keyboard transfer cycle, to allow for the relatively long command to data three-state time specified for most 8042-style controllers.

Figure 6-2 AT Keyboard Transfer Cycle



15620B-045B

XT Mode Direct Keyboard Interface

With the XT keyboard interface enabled, a XT style keyboard may be directly connected to the Am286ZX/LX device. Input and assembly of the serial scan code from the keyboard is done within the Am286ZX/LX device. Figure 6-3 shows the connection diagram for a XT keyboard interface.

When the XTE configuration bit is set to enable the XT interface, the definitions of the $\overline{CS8042}$ and KEYCLK pins change. $\overline{CS8042}$ becomes XTKBDATA, which is an open collector I/O line with an internal pull-up. XTKBDATA carries the keyboard scan code data in a serial format. It can also be driven Low by the processor to signal the keyboard. KEYCLK becomes XTKBCLK, which is also an open collector I/O line with an internal pull-up. XTKBCLK is used to carry clock edges to register the serial data on the XTKBDATA line. It can also be driven Low by the processor to signal the keyboard.

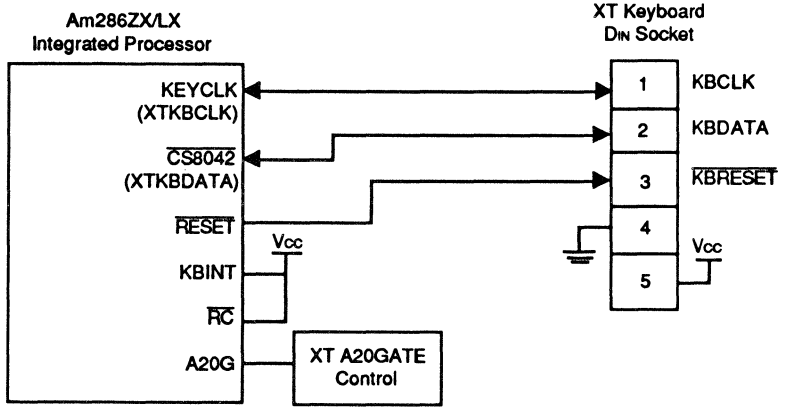
The keyboard interrupt is driven internally by the XT interface logic, so the external KBINT pin just needs to be tied High (to V_{CC}). The XT keyboard does not provide the CPU only reset signal, $\overline{RC}$, so it should be tied inactive (High). The XT keyboard does not provide the A20 gate signal, A20G, so it should also be tied inactive (Low). Note that CPU only reset and A20 gating can still be done in systems with a XT keyboard interface, by using the Port 92H Fast Reset and Fast A20 Gate functions.

With the XT keyboard interface enabled, the I/O port at address 060H becomes a read only 8-bit register, which provides the 8-bit scan code assembled from the incoming keyboard serial data stream. The I/O port at address 064H becomes a read/write control port with only two bits defined.

Bit 7 in the control port at address 064H is the XTCLR bit. XTCLR, when set to a 1, clears out the previous scan code received, and enables the shift register logic to receive another scan code. When active, this bit also clears the active interrupt line and the drive of a 0 onto the XTKBDATA line, which inhibits the keyboard from sending more scan codes. This bit should be pulsed High then Low after a keyboard interrupt under program control.

Bit 6 in the control port is the XTCLK bit. XTCLK, when cleared to a 0, will force the XTKBCLK line Low. When XTCLK is set to a 1, it allows the internal pull-up to pull the line High, which then allows the keyboard to drive the line.

Figure 6-3 XT Keyboard Interface



Both the XTCLR and XTCLK bits in the control port default to 0 after a hardware reset via the PWRGOOD pin.

XT Mode Cycles

In XT keyboard mode, data transfers with the data and control ports at addresses 060H and 064H are internal register accesses and will not be seen on the X Bus.

Figure 6-4 shows the serial transmission of a keyboard scan code to the Am286ZX/LX integrated processor. The XTKBDATA line is sampled with each XTKBCLK pulse. The serial data format begins with an active High start bit, then the 8 bits of data follow with the least significant bit coming first.

MATH COPROCESSOR INTERFACE

Interface Control Functions

COPROCESSOR RESET

An 8-bit I/O write by the CPU to address 0F1H will trigger a reset pulse to the coprocessor on the RESET287 pin. The active High pulse will be 16 CPU cycle clocks wide (32 CPU input clocks).

ERROR DETECTION

If the coprocessor error pin, $\overline{\text{P287ERR}}$, is active (Low) at the same time as the coprocessor busy pins, $\overline{\text{BUSY}}$, is active (Low), an interrupt will be generated to Channel 13 (IRQ13) of the slave 82C59 interrupt controller. At the same time, the internal $\overline{\text{BUSY}}$ line to the 80C286 will be latched active, preventing more coprocessor cycles until the error condition is cleared.

An 8-bit I/O write by the CPU to address 0F0H will clear the latched internal $\overline{\text{BUSY}}$ signal. This cycle is a zero command delay, zero wait state, HSSM controlled internal cycle.

Connection

Figure 6-5 shows a connection diagram for the coprocessor. For coprocessor cycles, the bus controller issues the address on the XA address bus, and it transfers the data on the MD data bus. Pull-ups should not be placed on the MD Bus. There are "keepers" on the MD pin buffers which maintain the state of the bus after it is three-stated by any active drive buffer. This bus keeper feature is required on the MD Bus in order to support some numeric

Figure 6-4 XT Scan Code Reception

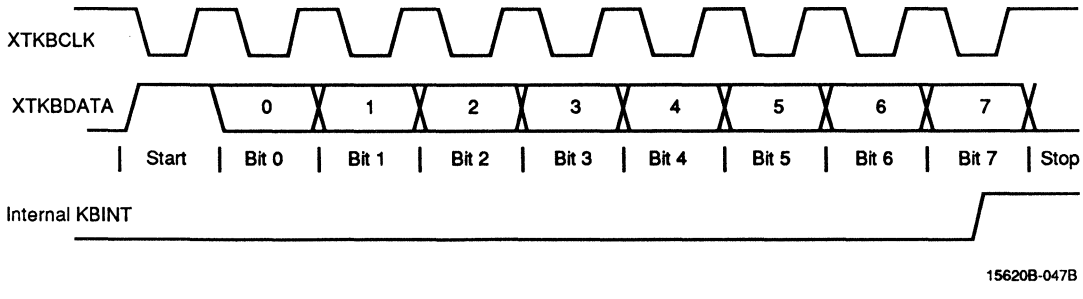
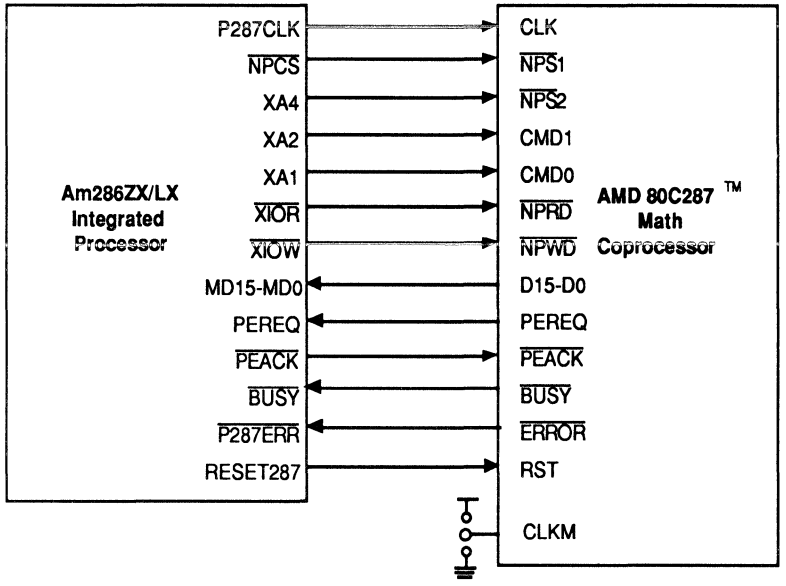


Figure 6-5 Numeric Coprocessor Interface



15620B-048B

coprocessor detection methods. Pull-ups on this bus will override the keepers and always force the bus to float to a high level.

All interface signals required by the coprocessor are supplied as direct connect pins on the Am286ZX/LX integrated processor. The P287CLK pin is a programmable clock output which can be sourced by either the PROCLK or IOCLK inputs, with divisors of 1, 2, 3, and 4. See the Enhanced Clock Generator section in Chapter 3 for more detail on the P287CLK output. Use of the P287CLK output for the coprocessor's clock is not absolutely necessary. The coprocessor can be clocked independently, and the P287CLK output can be used for a general purpose programmable clock output.

Bus Cycles

Figure 6-6 shows a bus transfer cycle between the internal 80C286 CPU and the external numeric coprocessor. It is a CPU synchronous, HSSM controlled, two wait states, and two command delay transfer cycles with recovery time (from the CPU's perspective, a three wait state cycle). The transfer cycle provides cycle timing compatibility with a wide range of processor and coprocessor speed combinations.

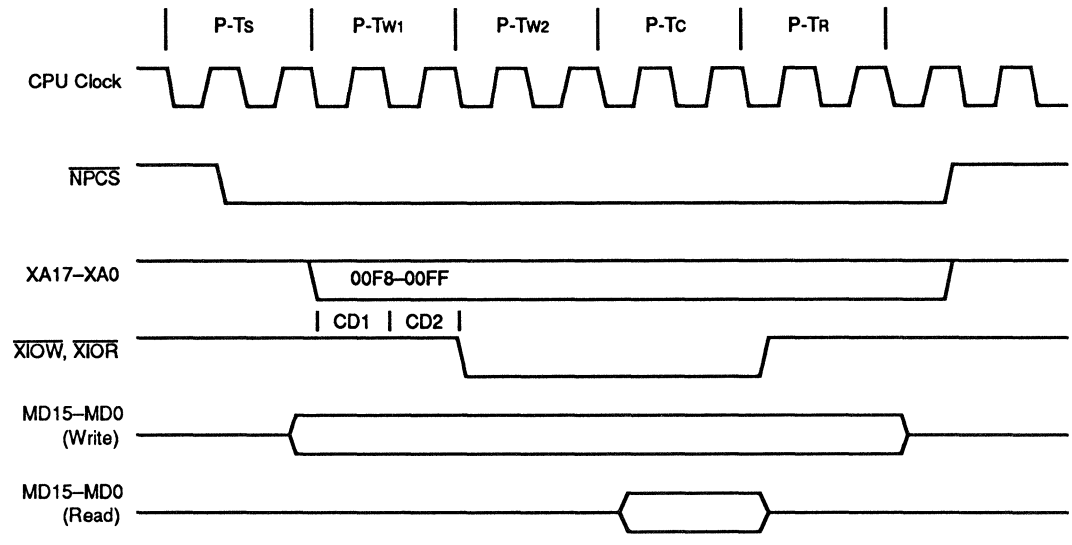
An extra CPU clock cycle of data recovery time is always appended to the end of a coprocessor transfer cycle, to allow for the relatively long command to data three state time specified for most math coprocessors for 80286-based systems.

ROM/EPROM INTERFACE

Functional Description

The ROM/EPROM interface's main function is to provide access to non-volatile memory for the purposes of having code to execute from device reset (bootstrap code), and of

Figure 6-6 Math Coprocessor Bus Cycle



P-Ts CPU Status Cycle
P-Tw CPU Wait State
P-Tc CPU Final Command Cycle
P-Tr CPU Recovery Cycle

15620B-049B

providing a set of standard hardware interface routines for system software (BIOS code). The Am286ZX/LX integrated processor implements this function by providing support for direct connection of static memory devices to the processor's X Bus. The static memory devices can be ROM, EPROM, or SRAM, with a bus width of either 8 or 16 bits, and with a total memory size of up to 256 kb.

ROM/EPROM bus width must be determined at reset time in order for the processor to be able to correctly fetch the first and following bytes of code immediately after reset. The Am286ZX/LX integrated processor supports ROM interface size detection at reset by sampling the $\overline{\text{IOCS16}}$ pin during reset. If the line is externally pulled Low, it indicates a 16-bit bus size. If the line is not pulled Low and the internal pull-up keeps the line High, the bus size defaults to 8 bits. See the ROM/EPROM connection diagram in Figure 6-7.

The fixed upper bound of the memory block supported by the ROM interface is at the end of the 1-Mb real mode address limit (address 0FFFFFH). The lower limit of memory block space supported by the ROM interface is 256 kb below the 1 Mb limit (address 0C0000H). However, the 80C286 processor, when it comes out of reset, starts fetching code from the very top of its physical address space, 16 Mb–16 bytes (FFFFF0H), even though it is in real mode. Therefore, CPU accesses to addresses FF0000H through FFFFFFH are mapped down to the ROM locations at 0F0000H through 0FFFFFH.

The Am286ZX/LX integrated processor provides programmable address range enable bits for the ROM interface. The full 256-kb ROM area can be individually enabled or disabled in 16-kb blocks. The enabled blocks do not have to be contiguous. These enable bits are contained in two configuration registers, ROM Bank Configuration Registers 1 and 2 (ROBR1 and ROBR2, at indexes 13H and 14H, respectively). Each ROM enable bit in

these registers enables an individual 16-kb block for ROM access at a specific address. The two top blocks are enabled at reset, for 32 kb of enabled ROM. The rest of the enables are cleared at reset.

ROM Bank Enable Bits

Name	Location	Address Range Enabled
RE15	ROBR2.7	0FC000–0FFFFF
RE14	ROBR2.6	0F8000–0FBFFF
RE13	ROBR2.5	0F4000–0F7FFF
RE12	ROBR2.4	0F0000–0F3FFF
RE11	ROBR2.3	0EC000–0EFFFF
RE10	ROBR2.2	0E8000–0EBFFF
RE9	ROBR2.1	0E4000–0E7FFF
RE8	ROBR2.0	0E0000–0E3FFF
RE7	ROBR1.7	0DC000–0DFFFF
RE6	ROBR1.6	0D8000–0DBFFF
RE5	ROBR1.5	0D4000–0D7FFF
RE4	ROBR1.4	0D0000–0D3FFF
RE3	ROBR1.3	0CC000–0CFFFF
RE2	ROBR1.2	0C8000–0CBFFF
RE1	ROBR1.1	0C4000–0C7FFF
RE0	ROBR1.0	0C0000–0C3FFF

The "ROM" interface also includes a write control line, $\overline{\text{XMEMW}}$, which will activate during writes to the ROM interface address space. This allows for such memory alternatives as Flash EPROM support with write-in-place capability, or for secondary SRAM on the X Bus.

Connection (8 bit and 16 bit)

Figure 6-7 shows the ROM/EPROM interface connection diagrams for both 8- and 16-bit configurations. Both the chip enable and the output enable should be connected on the ROM, as shown. Addressing for the ROM is provided on the XA lines. Note that only XA17–XD0 are used in the 16-bit configuration. The data path for the ROM data is via the XD bus for the 8-bit configuration. For a 16-bit ROM interface, XD7–XD0 transfer the Low byte and MD15–MD8 transfer the High byte.

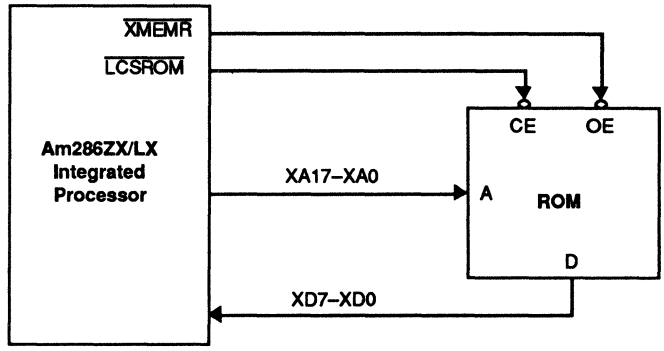
The 8-bit configuration is the default. The 16-bit configuration is selected by connecting an open-collector driver between the $\overline{\text{RESET}}$ output and the $\overline{\text{IOCS16}}$ input. If $\overline{\text{IOCS16}}$ is sampled Low during a reset, 16-bit ROM/EPROM accesses are enabled.

Bus Cycles

Figure 6-8 shows a bus transfer cycle with a ROM/EPROM. In this example, it is a CPU synchronous, HSSM controlled, one wait state, one command delay transfer cycle. It is always a HSSM controlled, one command delay cycle. However, the number of wait states for the cycle is controlled by a 4-bit field in the Memory Wait States register (MWS, index 12H). The bits are ROMWS3–ROMWS0, and they specify a straight binary encoded 0 through 15 wait states for ROM access cycles. ROMWS3–ROMWS0 are located in bits 3 through 0 of the MWS register, respectively.

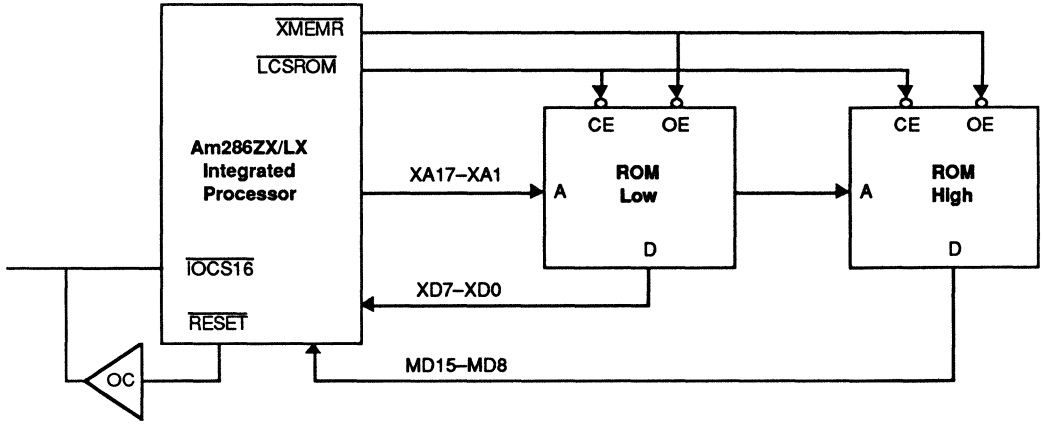
The default number of wait states is 1 (ROMWS = 0001). ROM access time requirements must be checked at reset time, as well as during full speed operation. The analysis should be done with the reset conditions of a one wait state ROM access with the CPU Clock equal to a fourth of PROCLK.

Figure 6-7 ROM/EPROM Interface Connection



Note: This is the default configuration.

a. 8-Bit EPROM Configuration



Note: Open collector buffer between $\overline{\text{RESET}}$ and $\overline{\text{IOCS16}}$.

b. 16-Bit EPROM Configuration

15620B-050B

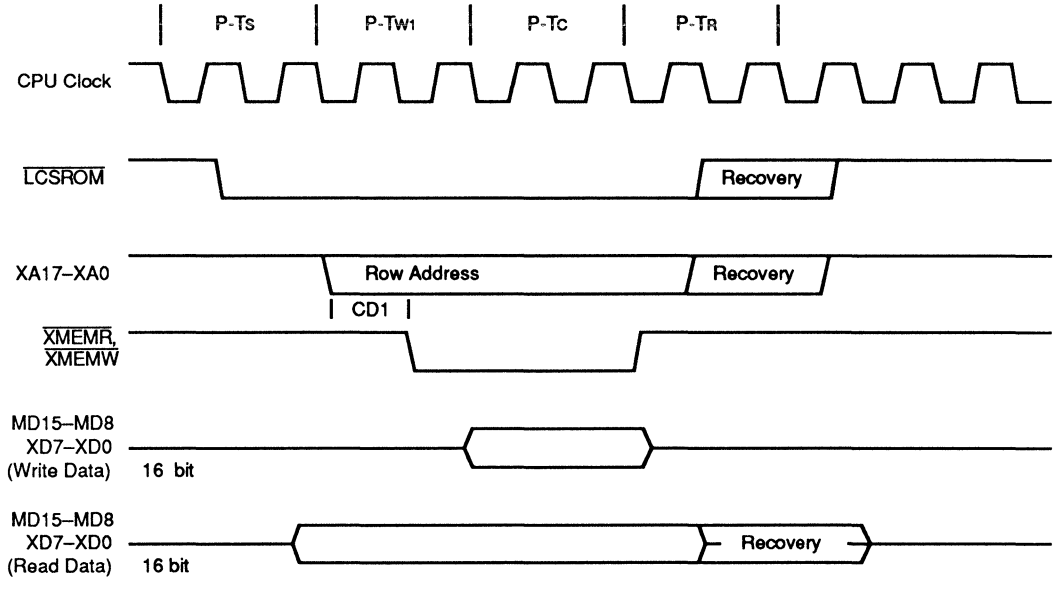
An extra CPU clock cycle of data recovery time can optionally be appended to the end of a ROM interface transfer cycle, to allow for the relatively long command to data three-state time specified for most ROMs and EPROMs. See the Enhanced Bus Controller section in Chapter 3 for the configuration options relative to recovery time.

OTHER X-BUS PERIPHERALS

Functional Description

The standard devices that are connected to the X Bus are the keyboard controller, ROM/EPROM, and numeric coprocessor. The bus controller's routing logic is hardwired to route accesses to these devices onto the X Bus. However, there is also a programmable method to direct any other I/O peripheral's access cycles onto the X Bus. This allows the system designer to use the X Bus as a true motherboard expansion bus. Another advantage is that

Figure 6-8 ROM/EPROM Bus Cycles (1CD, 1WS)



P-Ts CPU Status Cycle
P-Tw CPU Wait State
P-Tc CPU Final Command Cycle
P-Tr Optional Recovery Time

15620B-051B

putting a motherboard peripheral on the X Bus instead of the S Bus decreases the loading on the S Bus, providing more drive capability for expansion slots.

Enhanced X-bus capabilities are provided for maximum flexibility for X-bus peripherals: 16-bit peripherals can be mapped to the X Bus, as well as any of the 8-bit DMA channels.

Placing an I/O peripheral on the X Bus requires some method of instructing the bus controller to direct accesses to that peripheral to the X Bus, instead of the default S Bus. That capability is provided by a set of configuration registers, which can be programmed with an I/O address range that specifies the routing of that range onto the X Bus. There is also a configuration register to specify bus size for each address range, and there is one to specify which 8-bit DMA channels to route to the X Bus.

There are eight X-bus I/O Range Enable Registers, XBE1–XBE8 (indices 19H–20H, respectively). Each register can be used to route an 8-byte I/O address range onto the X Bus. The bit functions for the eight redirection registers are identical.

Bit 7 of each register is the Range Enable Bit, XBEN. If the bit is a 1, then the register is activated and all I/O accesses greater than 100H are compared against the contents of the rest of the register for possible X-bus redirection of the cycle. I/O addresses less than 100H cannot be redirected onto the X Bus. If the XBEN bit is a 0, then the register will not do any redirection, and the contents of the rest of the register will be ignored. The default value for the XBEN bits is 0, which means that redirection is disabled.

X-Bus Redirection Enables 1-8 (q=1 through 8)

XBENq XBEq.7	Function
0	Register disabled, bits 6–0 ignore
1	Register enabled, bits 6–0 used for range comparisons

If the XBEN bit is set for the register, then bits 6–0 (called XBq.9–XBq.3, respectively) are compared to address bits 9–3 during each I/O access, to test for redirection. If the I/O access is within the 8-byte range specified by XBq.9–XBq.3, then it is redirected to the X Bus.

There are two methods for specifying that an X-bus peripheral defined in one of the XBE registers is a 16-bit device. One method is static, or pre-configured, and the other method is dynamic, or "on the fly." The dynamic method is for the X-bus peripheral to activate the S-bus $\overline{\text{IOCS16}}$ line during access cycle in the same manner as the S-bus devices. The static method is implemented by programming the size of the peripheral for a range enable to be 16 bits. This static method utilizes the X-bus I/O Size Register (XBSZ, index 22H), with a bit per X-bus Enable Register to specify the size of the peripheral for that register. Bits 7 through 0 of the XBSZ register are the XBSZ8–XBSZ1 configuration bits. Each bit is attached to its like numbered XBE register. If the XBSZ bit is a 1, then the device for that range is a 16-bit peripheral. If the XBSZ bit is a 0, then the device for that range is either an 8-bit peripheral or a hardware signaled 16-bit device (via $\overline{\text{IOCS16}}$).

16-bit peripherals on the X Bus must have true word wide registers, with the CPU always doing word transactions to the registers. This requirement is because there is no X-bus equivalent of a $\overline{\text{BHE}}$ signal. Therefore, all transactions to that peripheral must be word transactions.

8-bit DMA peripherals are also supported on the X Bus. Any of the four 8-bit DMA channels (0–3) can be mapped to the X Bus, with none to all four redirected being possible. The classic example for the use of this feature would be the connection of a floppy disk controller to the X Bus. In this case, not only would an XBE register be programmed to direct the proper I/O accesses to the controller, but also DMA Channel 2 would be enabled for X Bus redirection. Note that the standard DRQ2 and $\overline{\text{DACK2}}$ signals would be taken from the S Bus, for use by the floppy controller on the X Bus.

The X-bus DMA Channel Enable Register (XBD, index 21H) contains the four enable bits (XBD3–XBD0) which control X Bus redirection for the four DMA channels. If the XBD bit is a 1, then that channel is redirected onto the X Bus. If the XBD bit is a 0, which is the default state, then DMA transfers with that channel will occur on the S Bus.

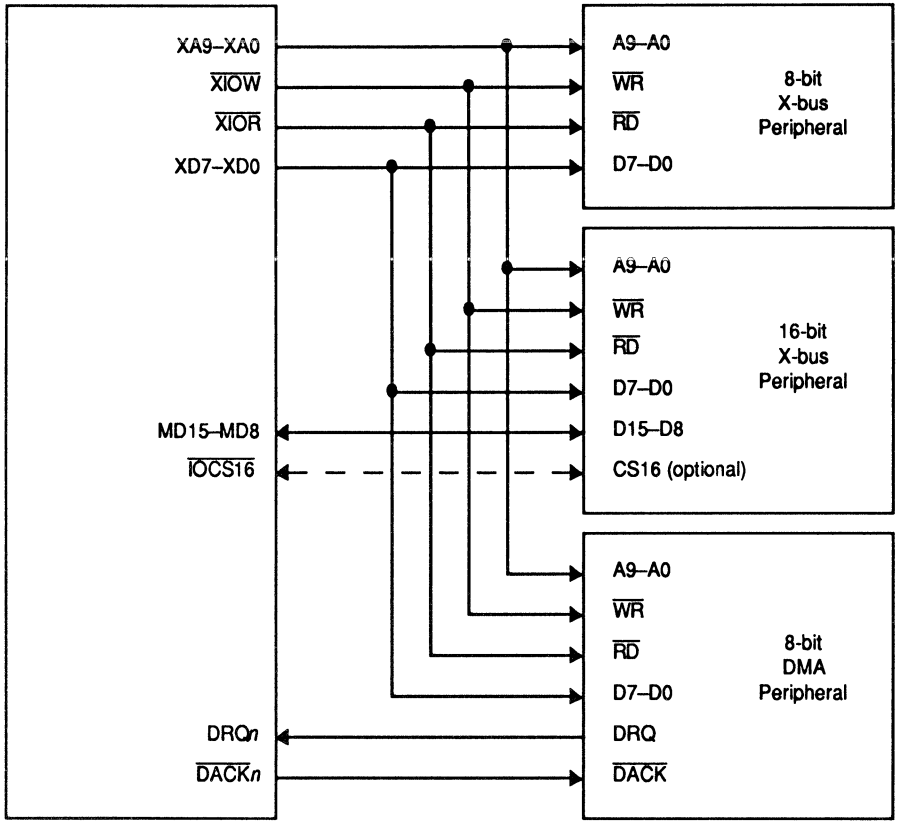
X-Bus DMA Channel Enables

Name	Location	Function
XBD3	XBD.3	X-bus DMA for Channel 3
XBD2	XBD.2	X-bus DMA for Channel 2
XBD1	XBD.1	X-bus DMA for Channel 1
XBD0	XBD.0	X-bus DMA for Channel 0

Connection

Figure 6-9 shows the connection of X-bus peripherals to the Am286ZX/LX integrated processor. 8-bit, 16-bit, and 8-bit DMA peripherals are shown. For the 16-bit device, the

Figure 6-9 X-Bus Peripheral Connections



15620B-052B

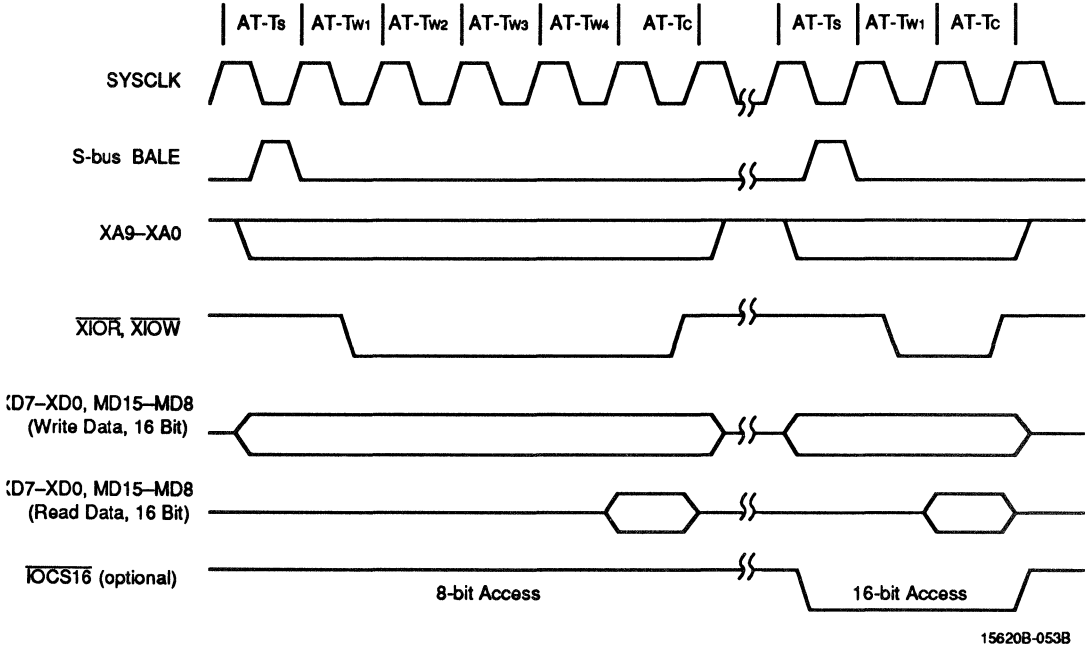
High byte connection is made to the M-bus High byte data lines, MD15–MD8. The optional hardware signaling to IOCS16 of the bus size is also shown. The 8-bit DMA device shows the use of the standard DRQ and DACK lines for DMA accesses.

Bus Cycles

Data transfer cycles to and from the X-bus I/O devices are always controlled by the ATSM. Therefore, the access cycle will generally be timed the same as an S-bus 8- or 16-bit I/O cycle. Figure 6-10 shows two typical data transfer cycles to or from X-bus I/O peripherals: the first is 8 bit and the second is 16 bit. The actual command delays and wait states used are dependent upon the 8- and 16-bit I/O cycle settings in the Command Delay Register (CDR, index 04H) and the Wait State Register (WSR, index 05H).

An extra CPU clock cycle of data recovery time can optionally be appended to the end of an X-bus peripheral transfer cycle, to allow for a relatively long command to data three-state time specified for some I/O devices. See the Enhanced Bus Controller section in Chapter 3 for the configuration options relative to recovery time.

Figure 6-10 X-Bus Peripheral Access Cycle



CHAPTER 7

MEMORY BUS (M BUS)

DRAM INTERFACE

Connection (2/4 Bank)

The Am286ZX/LX integrated processor DRAM interface was designed to support two basic configurations. Signals are provided for the connection of up to four 16-bit banks of DRAM with parity (totaling 18 bits per bank). However, due to output driver loading considerations, it is possible that only two banks can be supported without the addition of external drivers on the memory address lines. DRAM controller programmable options support individual bank sizes of 512 kb, 2 Mb, and 8 Mb. Thus, banks are usually made up of a pair of 256 kb x 9, 1 Mb x 9, or 4 Mb x 9 modules. Shown in Figure 7-1 is a typical two bank configuration. A four bank configuration is shown in Figure 7-2. The connection details of the DRAM array itself are shown in Figure 7-3. The figures show maximum bank count for the given configuration. Empty banks in these configurations are allowed.

"Bit-wide" DRAMs are not a requirement, and use of 4-bit wide (i.e., 1 Mb x 4) DRAMs significantly reduce MA address loading, possibly allowing more DRAM banks on the bus without address buffering.

During internal DMA and external master accesses to local memory, and the generation of staggered $\overline{RAS}$ signals during refresh cycles, a delay line is used to guarantee proper timing. DMA accesses are non-page mode accesses whether or not the DRAM controller has been programmed for page mode operation. The determining factors in delay circuit parameter selection are found in the basics for the external master DMA access to local memory. In this case, one has to examine the S-bus command to Memory command ($\overline{RAS}$ and DLYOUT) propagation delays as well as S-bus address to memory address propagation delay. These delays, in combination with the DRAM access time and memory data to S-bus data propagation delays, must total to a sum less than the overall master cycle length. The result of worst and best case analysis shows an ideal DL0 delay of 20 ns and DL1 delay of 50 ns. However, a DL0 delay of 25 ns is acceptable and probably less expensive. The following constraint equations can be used for calculation of acceptable DL0 and DL1 delay times:

CONSTRAINT EQUATIONS

Column Access Time on Bus Master Read

$$D0 < T_{AM} - T_{CBUF} - t_{139} - t_{140} - t_{47} - T_{DBUF} - T_{CAA}$$

Row Address Hold Time on Bus Master Access

$$D0 > T_{RAH} - t_{139} - t_{140} + t_{141}$$

CAS Access Time on Bus Master Access

$$D1 < T_{AM} - T_{CBUF} - t_{139} - t_{143} - t_{47} - T_{DBUF} - T_{CAC}$$

Data Setup to CAS on Bus Master Access

$$D1 > T_{DDEL} + t_{138} + T_{DBUF} - T_{CBUF} - t_{139} + T_{DS}$$

Column Address Setup on Bus Master Access

$D1 > D_0 + t_{140} + T_{ASC} - t_{143}$

Variable Definitions

D0	DL0 delay from DLYOUT
D1	DL1 delay form DLYOUT
T_{AM}	Bus master read access time, command to data valid
T_{CBUF}	Command buffer delay, 0 if unbuffered S Bus
T_{DBUF}	Data buffer delay, 0 if unbuffered S Bus
T_{DDEL}	Data delay after command on write
T_{CAA}	DRAM access time from column address
T_{RAH}	DRAM row address hold time after $\overline{RAS}$
T_{CAC}	DRAM access time from $\overline{CAS}$
T_{DS}	DRAM data setup time
T_{ASC}	DRAM column address setup before $\overline{CAS}$
t_{47}	SD valid from MD valid (read)
t_{138}	SD to MD valid delay (write)
t_{139}	S-bus command to DLYOUT delay
t_{140}	DL0 to MA valid delay
t_{141}	S-bus command to $\overline{RAS}$ delay
t_{143}	DL1 to $\overline{CAS}$ delay

DRAM Cycles

The DRAM controller has several modes of operation, each being optimized to provide the best possible performance for the mode. Illustrated in this section are typical cycles for each mode and access type. The internal CPU clock is given as a performance reference.

Non-page mode, zero wait state read and write cycles are shown in Figures 7-4 and 7-5.

Page mode cycles are illustrated, starting with page hit zero wait state write cycles with $\overline{RAS}$ inactive in Figure 7-6. The same conditions for a read cycle are shown in Figure 7-7. Page hit zero wait state cycles, both writes and reads, are detailed in Figure 7-8. Page miss cycles are depicted in Figure 7-9.

External Master access cycles are similar to internally initiated DMA cycles, with the added complications of system bus to memory bus address and data propagation issues. This type of cycle is shown in Figure 7-10.

Normal and Staggered refresh cycles are illustrated for reference in Figures 7-11 and 7-12.

OTHER CONNECTIONS

The memory data bus, MD15–MD0, would ideally be a private bus to eliminate all possibility of corruption of data signal integrity. However, in the interest of package size and pin count, the Am286ZX/LX integrated processor must share this bus between memory, ROM, 16-bit X-bus peripherals, and the coprocessor, in some configurations. Since there are no buffer control signals for MD15–MD0, any system including more than memory on this bus must carefully consider the load driving capabilities of all devices on the bus. As a guide for possible configurations, a diagram is provided showing possible MD bus connections in Figure 7-13.

Figure 7-1 Am286ZX/LX Integrated Processor Two Bank Configuration

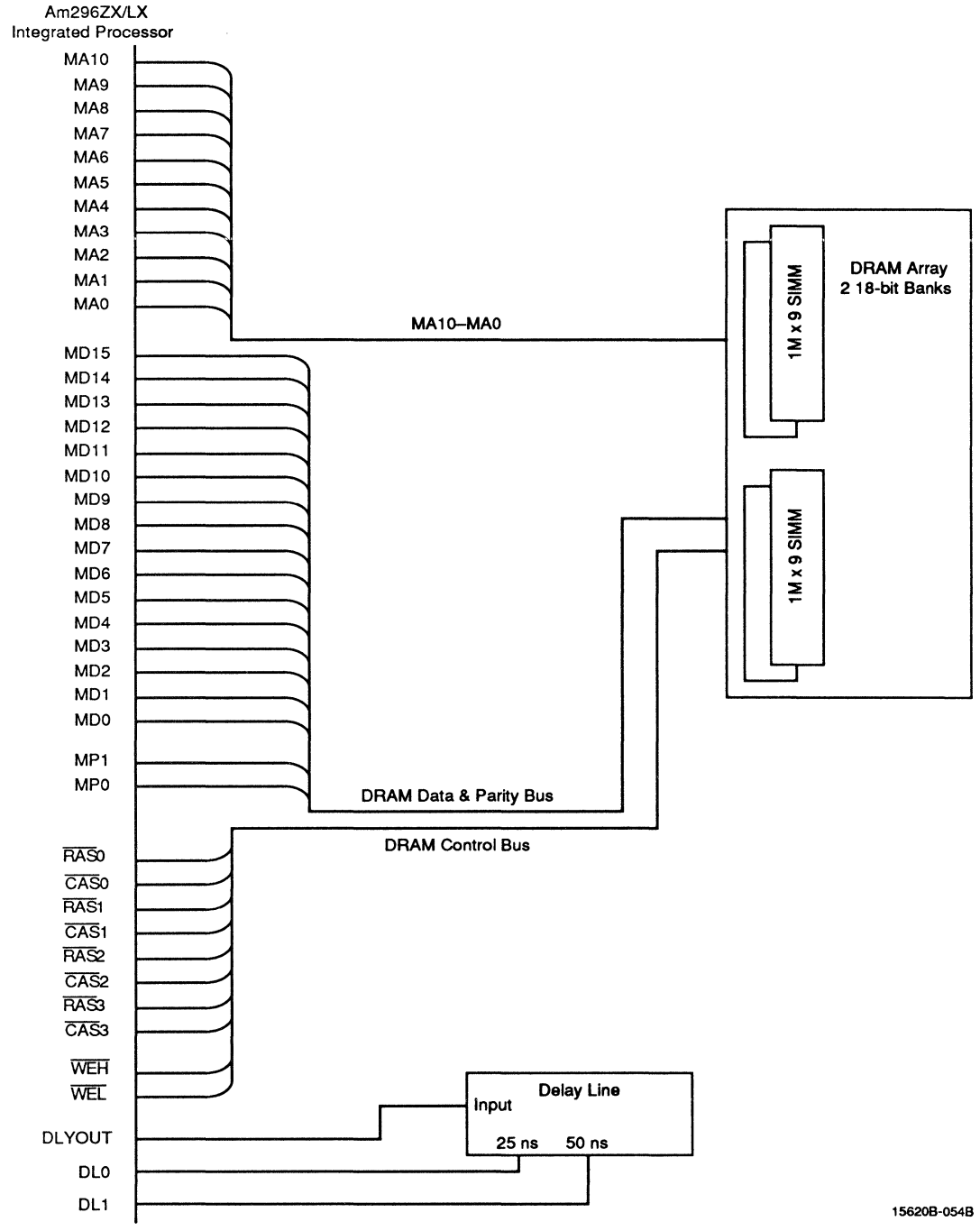


Figure 7-2 Am286ZX/LX Integrated Processor Four Bank Configuration

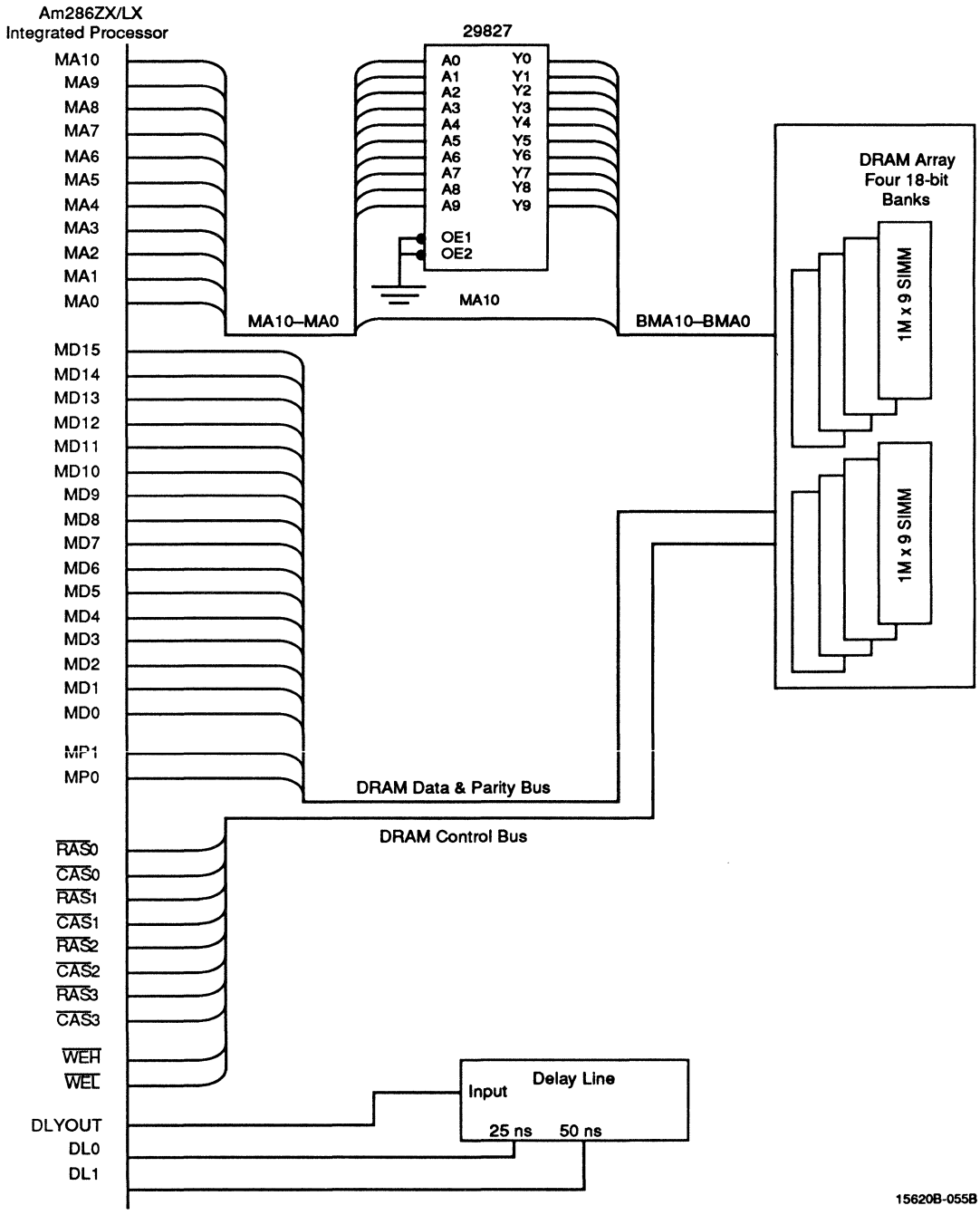
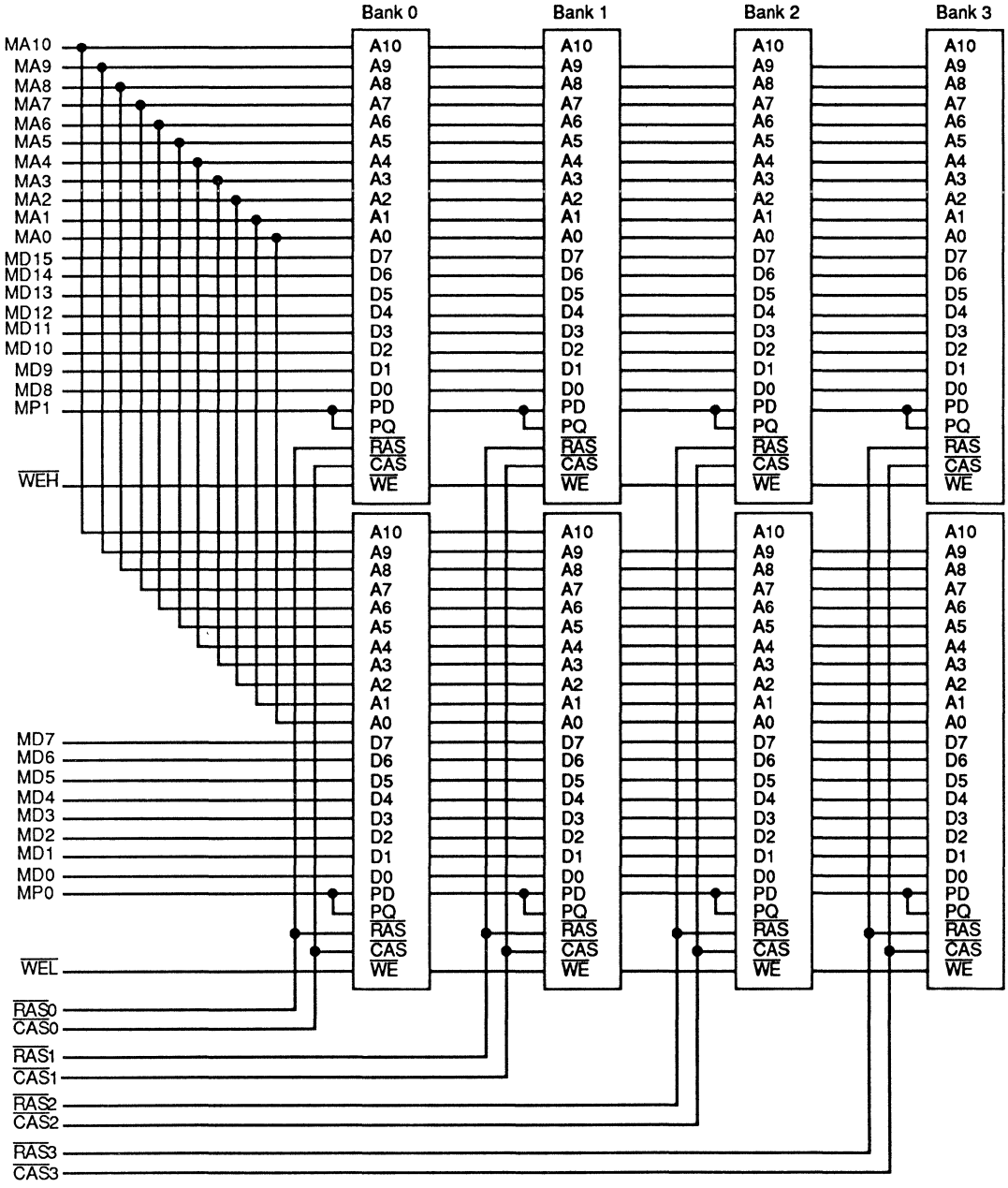


Figure 7-3 DRAM Array Connection Diagram Four Banks (Each Bank Consists of Two 8-Bit SIMM Modules)



15620B-056B

Figure 7-4 Non-Page Mode, Zero Wait State Read and Write Cycles

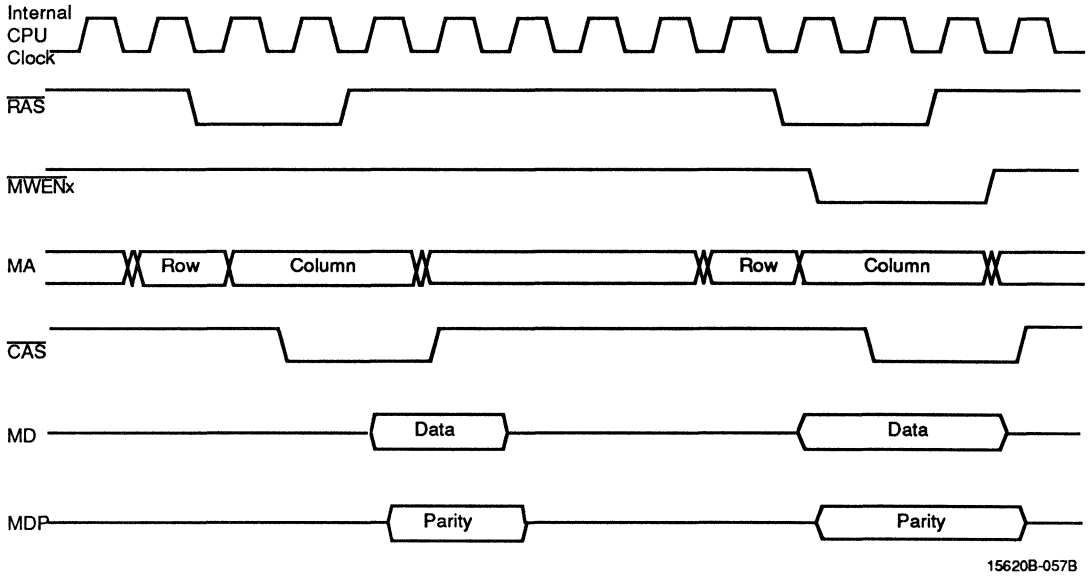


Figure 7-5 Non-Page Mode, One Wait State Read and Write Cycles

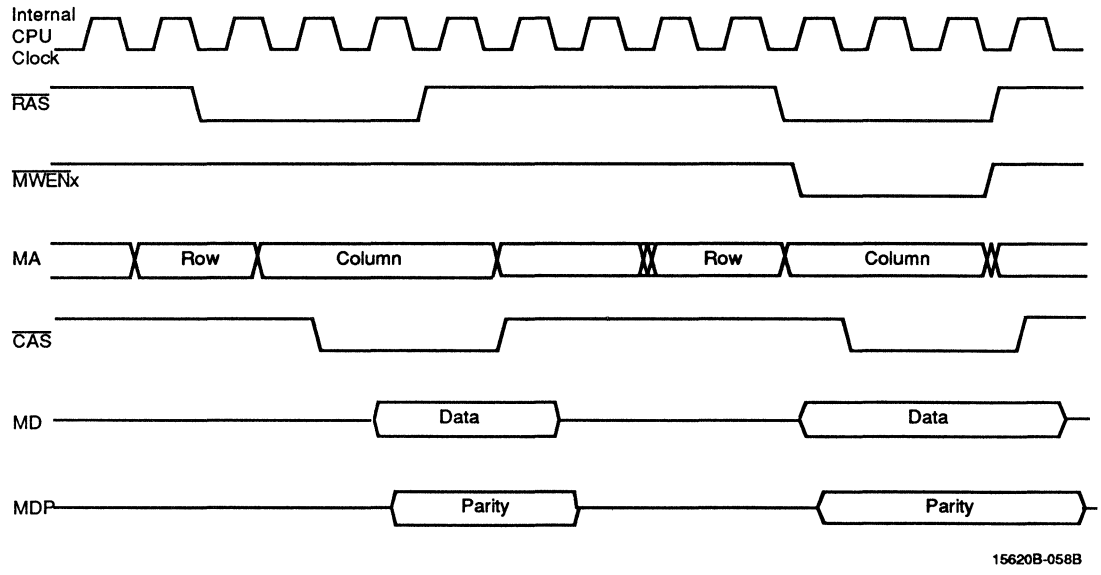
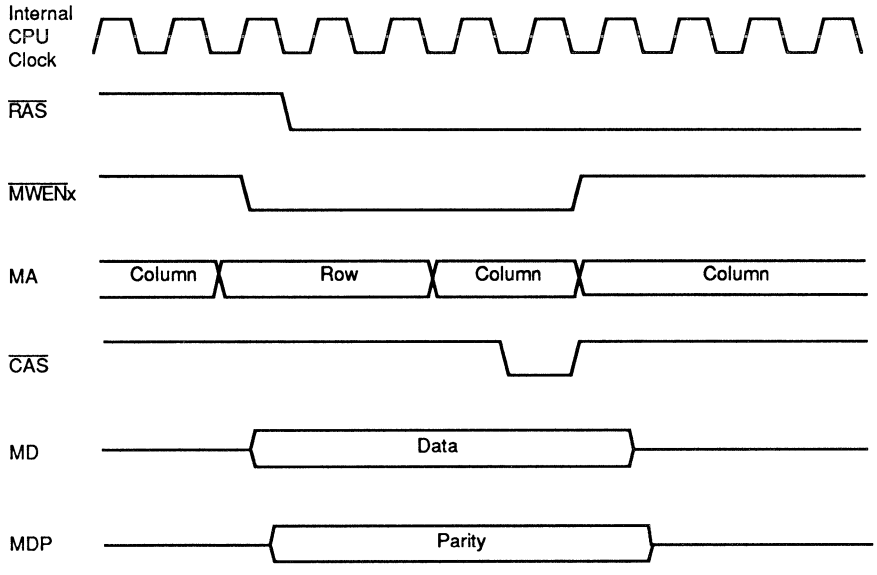
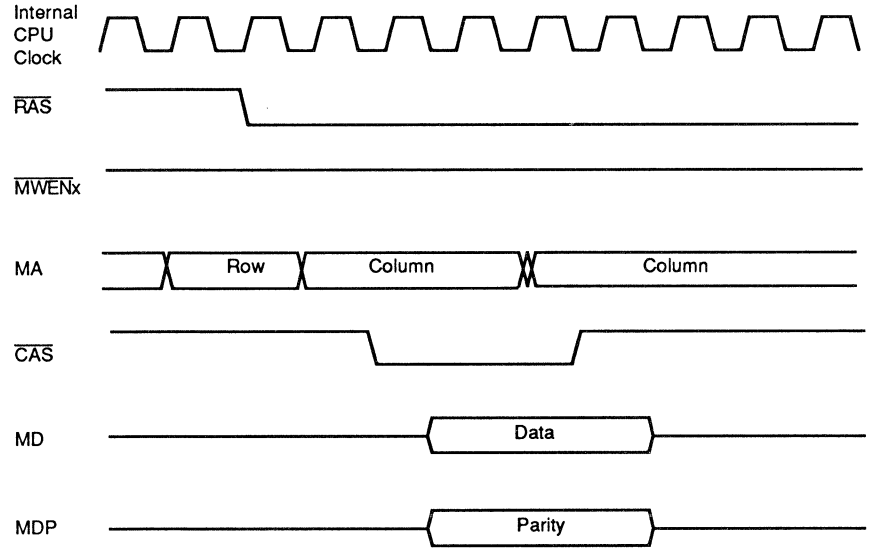


Figure 7-6 Page Mode Page-Hit, Zero Wait State Write Cycle with $\overline{RAS}$ Inactive



15620B-059B

Figure 7-7 Page Mode Page-Hit, Zero Wait State Read Cycle with $\overline{RAS}$ Inactive



15620B-060B

Figure 7-8 Page Mode Page-Hit, Zero Wait State Write and Read Cycles

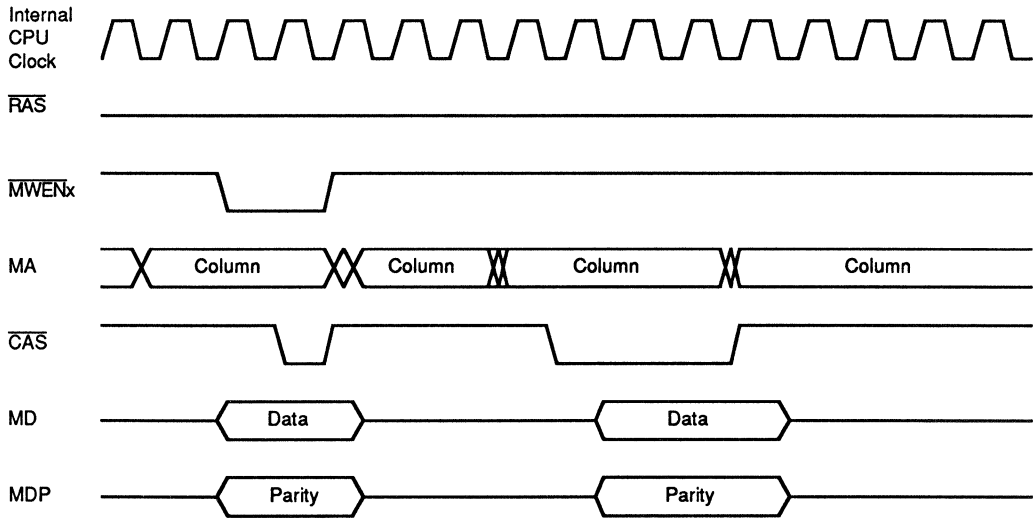
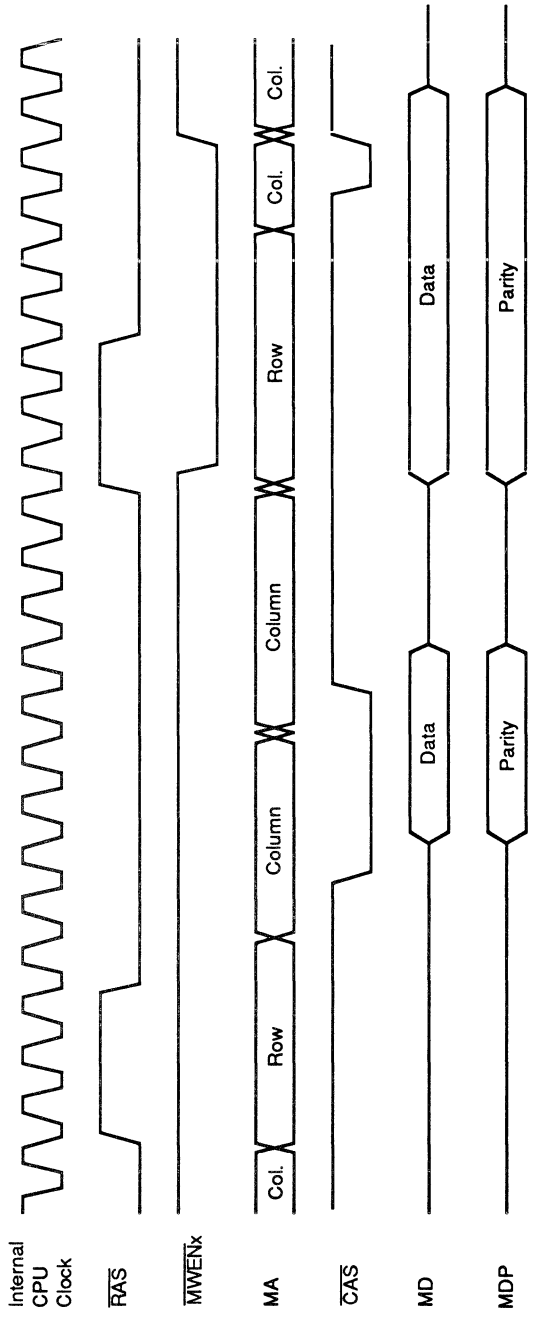


Figure 7-9 Page Mode Page Miss, Zero Wait State Read and Write Cycles



15620B-062B

Figure 7-10 S-Bus Master/DMA Access to Local Memory

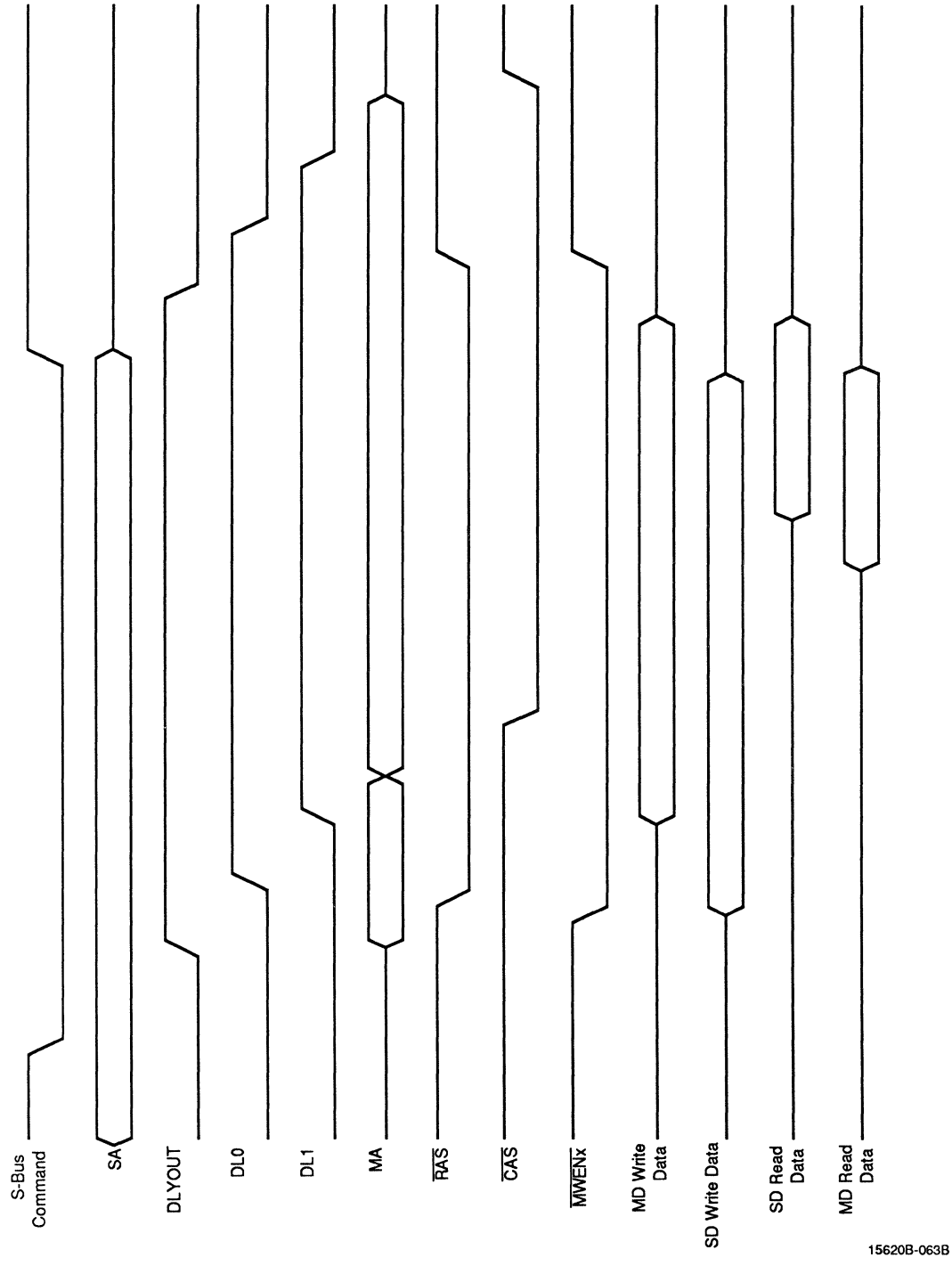


Figure 7-11 Normal Refresh

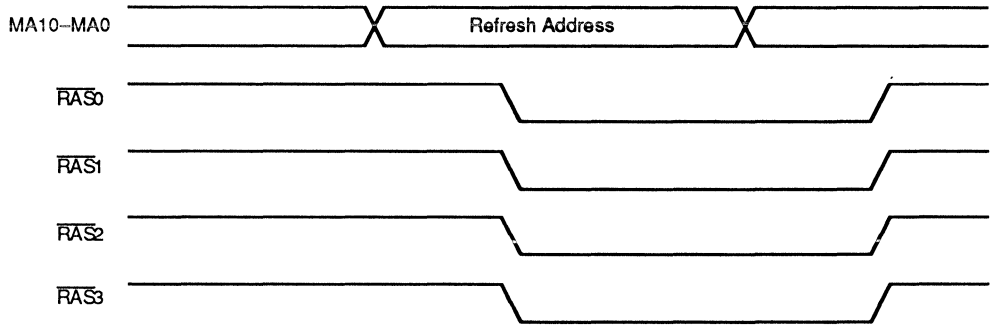


Figure 7-12 Staggered Refresh

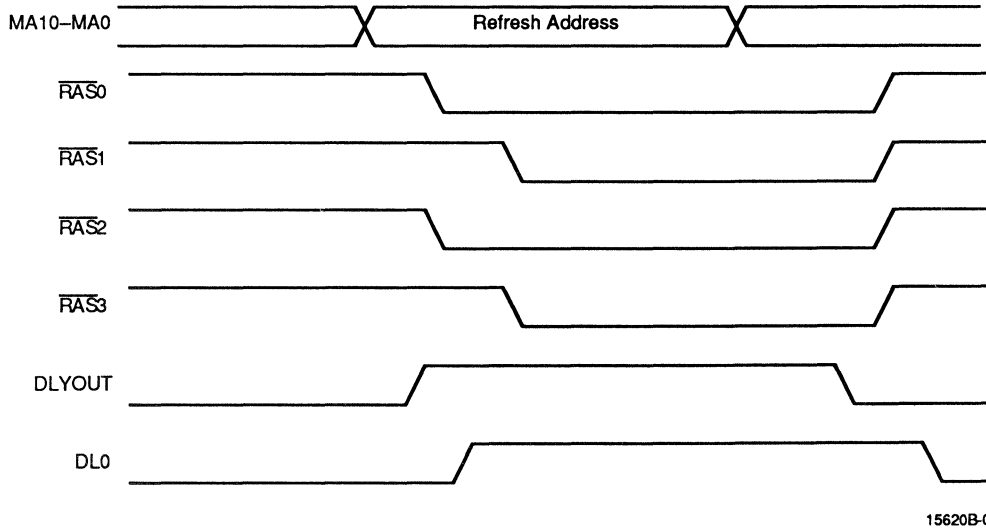
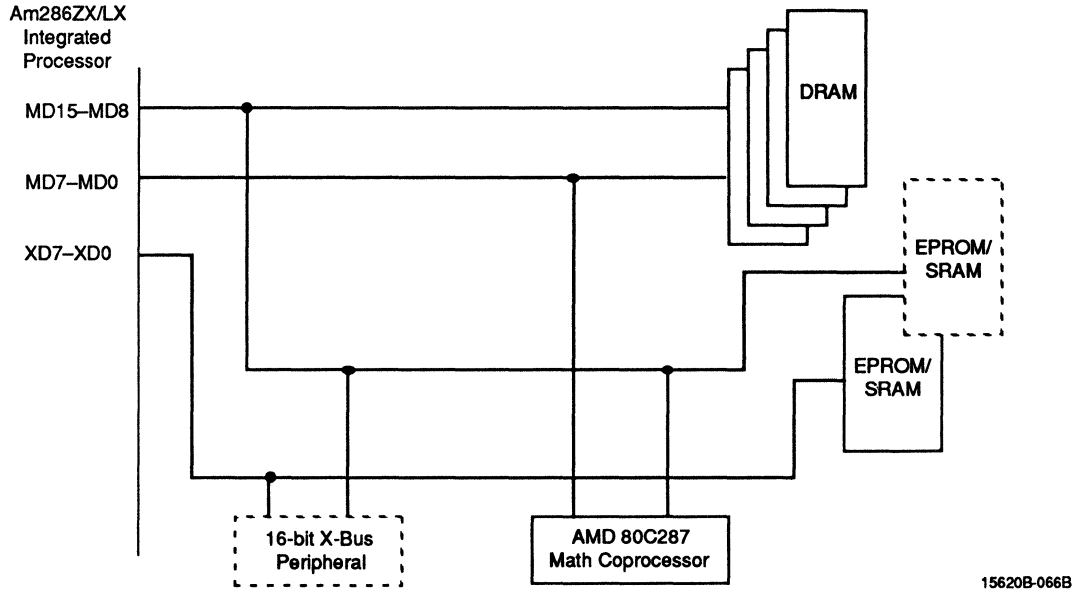


Figure 7-13 M-Bus Total Possible Connections



CHAPTER 8

SYSTEM CONFIGURATIONS

SYSTEM BLOCK DIAGRAMS

Introduction

The programmability and multi-bus structure of the Am286ZX/LX integrated processor allows flexible configuration of a broad range of system designs. Figures 8-1 and 8-2 illustrate two different system solutions using the Am286ZX/LX integrated processor. Figure 8-1 is a block diagram of an AT-compatible system that shows the integrated processor's ability to directly interface on all buses, and support 8-bit ROMs and optional X-bus mapping of additional peripherals. The second figure shows an expanded design that includes external buffering and a 16-bit ROM interface. The designs can be divided into the following six sub-blocks:

- Keyboard interface,
- ROM interface,
- AMD 80C287 math coprocessor interface,
- Local memory interface,
- ISA expansion bus interface, and
- Clock, reset, and delay line circuitry.

Keyboard Interface

The device is designed to connect directly to a 8042 or compatible keyboard controller. There are six specific pins that are provided by the integrated processor to achieve this. In addition, $\overline{\text{XIOR}}$, $\overline{\text{XIOW}}$, and XA2 must be used to control data transfers between the processor and keyboard controller. Since the 8042 is an 8-bit X-bus device, data is transferred on the XD7–XD0 bus.

ROM Interface

Both 8- and 16-bit ROM configurations are supported. The default, shown in Figure 8-1, is 8 bit. Figure 8-2 shows the open collector buffer connecting $\overline{\text{RESET}}$ and $\overline{\text{IOCS16}}$ to enable 16-bit ROM fetches. This memory is addressed on the X Bus with the low data byte attached to XD7–XD0. When a 16-bit ROM configuration is used, the upper data byte is transferred on MD15–MD8.

AMD 80C287 Math Coprocessor Interface

The standard interface to an AMD 80C287 math coprocessor or compatible is provided. This processor interface includes clock, reset, and 80C286 control signals, as well as a dedicated chip select line. Addressing for the AMD 80C287 math coprocessor is from the X Bus, while the data is transferred along the M-bus data lines. The industry standard I/O command decodes for ports 0F0h and 0F1h that are associated with the math coprocessor are handled internally by the integrated processor.

Figure 8-1 Am286ZX/LX Integrated Processor System Design with Direct Interface to Busses, 8-Bit ROM Support, and X-Bus Mapping of Peripherals

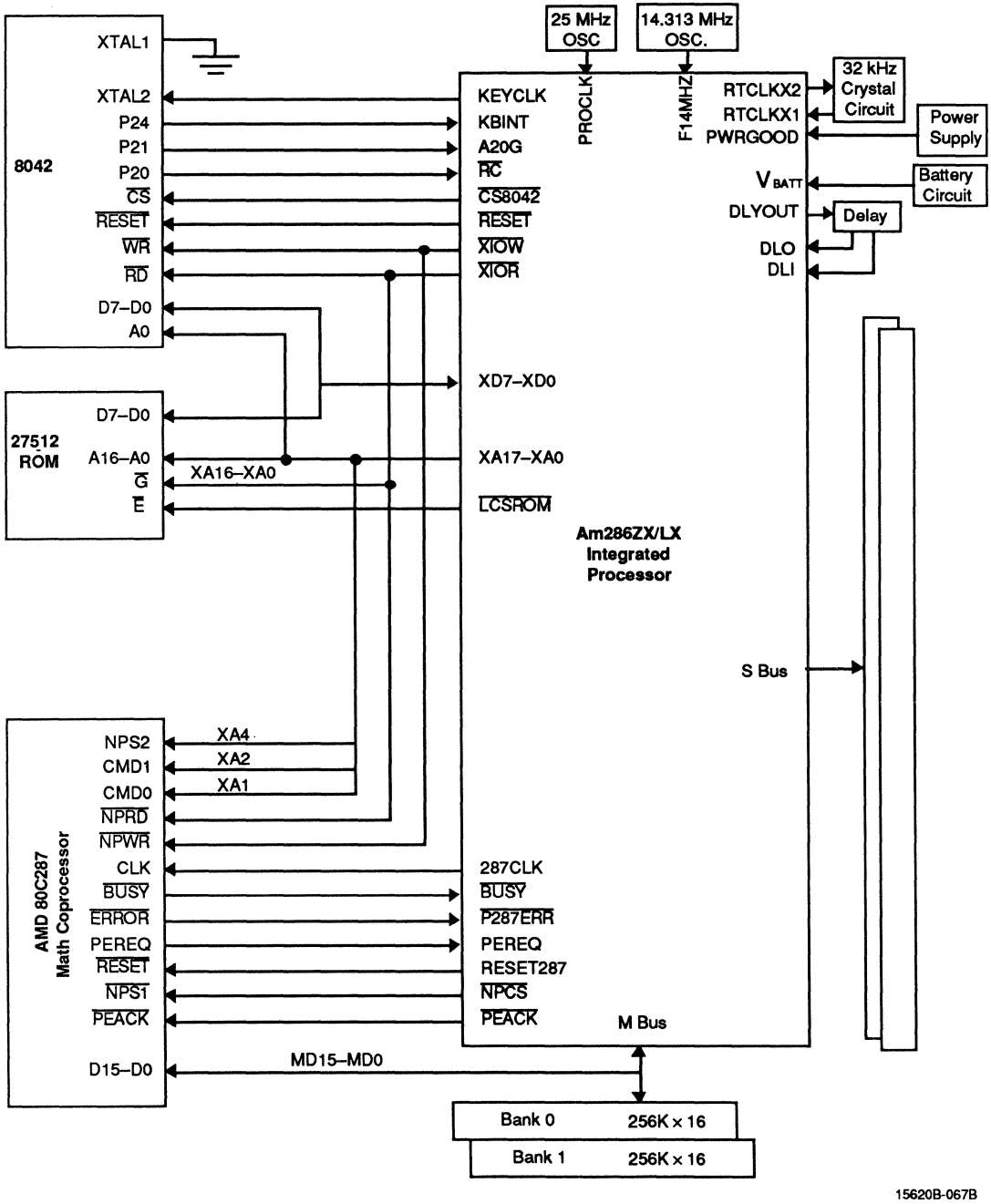
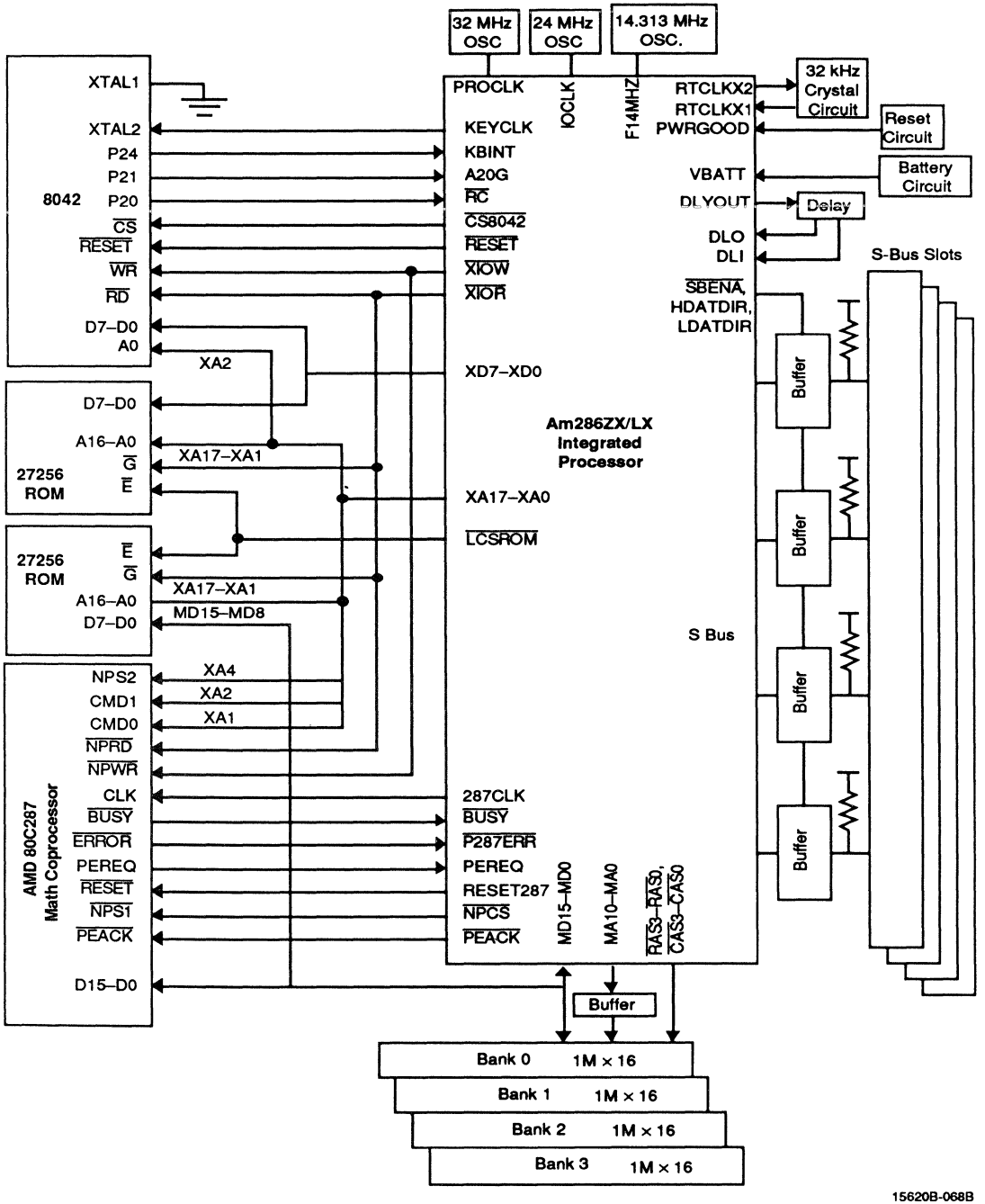


Figure 8-2 Am286ZX/LX Integrated Processor Expanded System Design with External Buffering and 16-Bit ROM Support



Local Memory Interface

As seen in Figure 8-1, the processor can directly drive two 16-bit (18 bit with optional parity bits) banks of DRAM. If 4-Mb DRAMs are used, the total physical addressing range of the 80C286 would be implemented with two banks (16 Mb). Buffering MA9–MA0 (Figure 8-2) allows the processor to control a full four banks of RAM. This buffering can be accomplished with a single Am29827 10-bit buffer. Separate $\overline{RAS}$ and $\overline{CAS}$ control lines are provided for each bank, as well as write enable controls for both data bytes.

ISA Expansion Bus Interface

The integrated processor provides the address, data, and control signals that comprise the ISA expansion bus (S Bus). A number of expansion slots can be directly connected to the integrated processor. If a larger number of AT bus slots are required, buffers can be added, as illustrated in Figure 8-2. Pull-up resistors should be added on the slot-side of the buffers.

Clock, Reset, and Delay Line Circuitry

The Am286ZX/LX integrated processor has four clock inputs. Three require TTL compatible input, while the fourth, used to clock the internal RTC circuitry, can use a crystal network input. Either IOCLK or PROCLK oscillators may be left out, if all required frequencies can be derived from just one of these inputs. Figure 8-1 shows a 25-MHz oscillator connected to PROCLK. This would allow the processor to run at 12.5 MHz, with an S-bus speed of 6.25 MHz.

If an external reset switch is not required (Figure 8-1), the PWRGOOD input of the processor can simply be connected to the PWRGOOD signal of the power supply.

A delay line with taps of 25 and 50 ns is needed for DMA and Bus Master accesses to local memory.

CONFIGURATION REGISTER INITIALIZATION

Introduction

Tables 8-1 and 8-2 are examples of how configuration registers could be initialized to support the systems modeled in figures 8-1 and 8-2. The tables list only those registers whose default values are altered. These registers may be divided into three groups: clock control, cycle control, and general setup.

Clock control consists of the two clock configuration registers, CCR1 and CCR2. These registers select the source and divisors for the AT state machine (ATSM clock), processor (CPU clock), keyboard (KEYCLK), and coprocessor (287CLK) clocks. Cycle control is composed of those registers that define the command delays and wait states required for most bus cycles. These registers are PSR, CDR, WSR, and IPCD. The remaining configuration registers are classified as general setup.

The order in which these groups are loaded is critical. The decision on which group of registers to write first depends on the new clock values to be loaded. In the examples of Tables 8-1 and 8-2, these values will be written after a hardware reset (via PWRGOOD), when system clock frequencies will be increasing from their defaults. In this case, the cycle control group must be written prior to the clock control registers, so that the higher clock frequencies do not cause problems with accessing ROM or internal peripherals.

Table 8-1 Am286ZX/LX Integrated Processor Configuration Register Values for Figure 8-1

Order	Name	Index	Default Value	Programmed Value
1	MWS	012H	001H	002H
2	IPCD	008H	000H	0FFH
3	CDR	004H	0FFH	015H
4	WSR	005H	0FFH	066H
5	CCR1	001H	000H	042H
6	CCR2	002H	0C0H	024H
7	RABR	015H	000H	005H
8	DCR	016H	000H	001H
9	ROBR2	014H	0C0H	0F0H
10	XBE1	019H	000H	0FEH
11	XBD	021H	000H	004H

Table 8-2 Am286ZX/LX Integrated Processor Configuration Register Values for Figure 8-2

Order	Name	Index	Default Value	Programmed Value
1	MWS	012H	001H	003H
2	IPCD	008H	000H	0FFH
3	CDR	004H	0FFH	015H
4	WSR	005H	0FFH	066H
5	CCR1	001H	000H	052H
6	CCR2	002H	0C0H	064H
7	RABR	015H	000H	0A5H
8	DCR	016H	000H	036H
9	ROBR2	014H	0C0H	0F0H

Cycle Control Group

Both systems program these registers with similar values. These values determine the command and data timings for the five programmable types of bus transfers that may occur:

- Internal I/O,
- Local memory,
- AT-bus I/O,
- AT-bus memory, and
- X-bus ROM accesses.

INTERNAL I/O

These locations are the ports associated with the internal peripherals of the integrated processor (DMA, RTC, etc.). The command delays for all accesses are programmed (IPCD) to three clock delays. The default of four wait states (IPWS1, IPWS2) is unaltered.

LOCAL MEMORY

Optional wait states may be added (MWS), but these design examples are zero or near zero (page mode) memory schemes, and therefore, the default of no wait states is unaltered.

AT-BUS I/O

These 8- and 16-bit cycles not only reference ports located on the AT Bus (S Bus), but also any devices that have been mapped (via the X-Bus I/O Range Enable Registers) to the X Bus. Both 8- and 16-bit transfers have been programmed (CDR) to one command delay. Wait states are modified (WSR) to an AT compatible—four for 8-bit cycles, and one for 16-bit accesses.

AT-BUS MEMORY

These accesses are defined as memory transfers to any address not assigned to X-bus ROM (ROBR1, ROBR2) or local memory (RABR) space. Command delays for these cycles are configured (CDR) to one for 8-bit accesses and none for 16-bit transfers. Like AT I/O cycles, the wait states are set (WSR) to one for 16-bit accesses and four for 8-bit transfers.

X-BUS ROM

There are no programmable command delays associated with X-Bus ROM cycles. The wait states for Figure 8-1 have been set to two, and those associated with Figure 8-2 to three. These extra cycle times have been added to allow for a ROM data access time of up to 200 ns. If faster ROMs are used, these wait states should be reduced accordingly.

Clock Control Group

The two block diagrams illustrate two different solutions to generating the various system clocks used in the AT architecture. Figure 8-1 has only a single 25-MHz oscillator (attached to PROCLK) to source the four clocks that synchronize system events. The 25-MHz signal is routed undivided (CCR1) to the CPU (CPU clock), to classify the system as a 12.5-MHz design.

This PROCLK source is divided by two to provide a 12.5-MHz KEYCLK (CCR2), 287CLK (CCR2), and ATSM clock (CCR1). The latter is always divided by two to provide the AT-bus clock, SYSCLK. In this case, its frequency is 6.25 MHz.

The system modeled in Figure 8-2 has two clock sources available for the above mentioned clocks. A 32-MHz oscillator is driving PROCLK, and a 24-MHz oscillator is connected to IOCLK. Internal register CCR1 has been modified to provide an IOCLK sourced ATSM clock of 12 MHz (SYSCLK of 6 MHz) and a CPU clock of 32 MHz (16 MHz design). The new values written to CCR2 halve both the IOCLK and PROCLK. The former to generate a KEYCLK of 12 MHz, the latter to supply a 16-MHz frequency for 287CLK.

General Setup

Default values must be changed for ROM size (ROBR2), along with local memory size (RABR) and accessing mode (DRC). In addition, the design of Figure 8-1 requires the floppy disk I/O space to be mapped to the X Bus (XBE1), along with the associated DMA channel (XBD).

The value of RABR in Table 8-2 enables all four banks of local memory. Banks 0 and 1 for 256-kb DRAMs and Banks 2 and 3 for 1-Mb devices. This will give the system a total of 5-Mb of local memory. The new value in DCR configures this memory for page mode, interleave accesses. Both parity and $\overline{RAS}$ timeout are also enabled. DRAMs with an access time of at most 100 ns are required for reliable accesses to this memory.

The values of Table 8-1 enable (RABR) only the first two banks for 256-kb DRAMs for a total local memory size of 1 Mb. This memory is programmed (DCR) for normal mode accesses without parity checking. In addition, the 384 kb of memory located between 0A0000H and 0FFFFFH have been located to begin at 0100000H. DRAMs of 80 ns or faster are necessary for this true zero wait state memory configuration.

The mapping of S-bus I/O space 03F0H–03F7H to the X Bus is necessary for the floppy disk controller to be addressed correctly. A value of 0FEH in XBE1 allows all accesses to the 8-byte segment to be routed to the X Bus. The corresponding DMA channel, Channel 2, is also directed to the X Bus with XBD's configuring.

CHAPTER 9

BIOS PROGRAMMING GUIDELINES

INTRODUCTION

The Am286ZX/LX integrated processor provides the BIOS programmer with several new ways to implement low level services common to most systems. The following sections show example code for BIOS areas that should be modified to take advantage of the integrated processor's full feature set.

INITIALIZATION

The initialization code examples in Figures 9-1 and 9-2 need to be executed only after a hardware reset. Figure 9-1 illustrates how the integrated processor's internal registers would be initialized using a ROM-based table. The code in Figure 9-2 is for detecting the size of RAM in each bank of local memory. This information is then programmed into the RAM bank configuration register (RABR). The code in Figure 9-2 should be executed after that of Figure 9-1 and the starting of DRAM refresh. The RAM sizing routine will invalidate all local memory.

SPEED SWITCHING

Any of the clocks controlled by the integrated processor can be modified by a simple write to the appropriate control register. Consideration should be given to assuring adequate cycle timing (command delays and wait states) whenever a clock frequency is increased. However, this issue can be handled by initializing the cycle timing for the fastest frequency possible. Command delays and wait states that work at the highest speed will also work at slower speeds. The code examples of Figure 9-3 show how the processor clock speed can be changed "on-the-fly."

FAST RESET AND GATE A20

The Am286ZX/LX integrated processor supports both "Fast Reset" and "Fast Gate A20" functions through the industry standard port 092H. Figure 9-4 is an example of a procedure to trigger a fast reset. Figure 9-5 shows code to set and clear the Gate A20 function.

X-BUS PERIPHERALS

I/O devices that usually reside on the S Bus (AT expansion bus) can be mapped to the X Bus. Figure 9-6 shows the procedures needed to map a floppy disk controller's I/O space and DMA to the X Bus.

POWER SAVE FEATURES

The Am286LX processor has two power saving clock modes that can be programmed "on-the-fly." The first, called CPU Stop Clock, stops only the CPU clock. The second mode,

where all system clock functions except refresh are inhibited, is called System Standby. Figure 9-7 shows code examples that enable these modes.

EMS 4.0 REGISTER PROGRAMMING

The EMS page registers, including the PGE (Page Enable Bit) bits, are undefined on power-up and must be initialized before translations are enabled. The code in Figure 9-8 initializes both the primary and secondary sets of EMS registers to zero (including the PGE bits), then enables global translations (GTE bit).

XT KEYBOARD INTERFACE PROGRAMMING

The integrated processor supports a direct connection to a XT style keyboard. In general, the programming interface is the same as a normal XT BIOS, except that port 061H functions (PB7 and PB6) are supported through I/O port 064H. The code for basic initialization and the reading of scan codes is shown in Figure 9-9.

Figure 9-1 Initialize Configuration Registers

```
        cli
        mov     al,080H             ;Load NMI mask
        out     070H,al             ;Turn of NMI's
        cld

        mov     ax,cs               ;set CS and DS equal
        mov     ds,ax
        mov     bx,OFFSET reg_tbl   ;set start of pointer
tblrd:  mov     al,BYTE PTR [bx]    ;move contents to accum.
        or      al,00h              ;see if end of table and set zero flag
        jz      tblend              ;continue code if end of table
        out     22h,al              ;INDEX port
        inc     bx                  ;increment pointer
        mov     al,BYTE PTR [bx]
        out     23h,al              ;DATA associated with this port
        inc     bx
        jmp     tblrd               ;repeat loop until end of table

;-TABLE OF CONFIG REGISTERS--
;INDEX,DATA

reg_tbl DB      012h,003h           ;MWS - RAM 0w/s, ROM 3 w/s
        DB      008h,0FFh           ;IPCD - 3 CD for all
        DB      004h,015h           ;CDR - 0 CD M16, 1 CD M8,I16,I8
        DB      005h,066h           ;WSR - 1 w/s 16 bit M/IO, 4 w/s 8 bit M/IO
        DB      001h,052h           ;CCR1 - AT state machine clock = IOCLK/2
                                    ;CCR1 - Processor clock= PROCLK
        DB      002h,064h           ;CCR2 - KBDCLK = IOCLK/2, 287CLK = PROCLK/2
        DB      016h,036h           ;DCR - Page mode interleaved DRAM accesses
                                    ;DCR - Parity and RAS timeout enabled
        DB      014h,0F0h           ;ROBR2 - ROM enabled from 0F0000H to 0FFFFFH
        DB      000h,000h           ;Indicates end of list

tblend: nop
      ;continue
```

Figure 9-2

```
        ;START REFRESH AND PRECHARGE RAM BEFORE THIS CODE

        ;SIZE AND CONFIGURE LOCAL MEMORY
        ;This code will detect what size DRAM (if any) each local memory bank
        ;contains. This information will then be loaded into RABR to configure
        ;the system's local memory size.

                mov             al,015H    ;point configuration index to RABR
                out             022H,al
                xor             ax,ax
                mov             ds,ax      ;all memory operations to lower 64k
                mov             ax,0C0C0H  ;initial RABR mask, bank 3 for 4Mb
DRAMs
                mov             cx,4;initialize loop counter for 1 pass/bank
                xor             dl,dl      ;initial RABR value
ramsz:  out     023H,al         ;configure current bank to 4Mb DRAMs
                mov             si,00200H
                mov             WORD PTR [si],0H ;clear 000200H
                mov             si,00600H
                mov             WORD PTR [si],0H ;clear 000200H
                mov             si,00E00H
                mov             [si],0AA55H      ;if 256kb DRAMs then written to
00200H
                                    ;if 1Mb DRAMs then written to 00600H
                                    ;if 4Mb DRAMs then written to 00E00H
                mov             si,0000H
                mov             WORD PTR [si],0  ;clear mdbus of phantom data
                mov             si,00200H
                cmp             WORD PTR [si],0AA55H   ;Does the bank have 256kb
DRAMs
                jne             chk_1M
                and             al,001010101B    ;256kb mask
                or              dl,al      ;update value to program RABR with
                jmp             lp_end
chk_1M: mov     si,00600H
                cmp             WORD PTR [si],0AA55H   ;Does the bank have 1Mb
DRAMs
                jne             chk_4M
                and             al,010101010B    ;1Mb mask
                or              dl,al      ;update value to program RABR with
                jmp             lp_end
chk_4M: mov     si,00E00H
                cmp             WORD PTR [si],0AA55H   ;Does the bank have 4Mb
DRAMs
                jne             lp_end
                or              dl,al      ;update value to program RABR with
lp_end: mov     al,ah           ;update current bank to 4 Mb DRAMs
                ror             ax,2
                loop            ramsz      ;loop for four banks
                mov             al,dl
                out             023H,al    ;configure RABR with final value
        ;CONTINUE WITH BIOS INITIALIZATION
```

Figure 9-3

```
;These procedures modify the CCR1 register to adjust the processor clock
;to operate at either full or half the frequency of the source oscillator.

full_CPU_speed  equ   000000010B ;mask for full speed processor clock
half_CPU_speed  equ   000000100B ;mask for half speed processor clock

fast_CPU   proc uses ax
        mov  al,001H
        out   022H,al             ;point the configuration INDEX to CCR1
        in    al,023H             ;read current value of CCR1
        or    al,full_CPU_speed   ;set the clock divisor to one
        out   023H,al             ;write out new value to CCR1
        ret
fast_CPU      endp

slow_CPU   proc uses ax
        mov   al,001H
        out   022H,al             ;point the configuration INDEX to CCR1
        in    al,023H             ;read current value of CCR1
        or    al,full_CPU_speed   ;set the clock divisor to one
        out   023H,al             ;write out new value to CCR1
        ret
fast_CPU      endp
```

Figure 9-4

```
;This procedure generates a fast reset through port 092H.  The reset
;is rising edge triggered so a 0 is written first followed by a 1.

fast_reset proc uses ax
        in    al,092H             ;read in current value
        and   al,011111101B       ;clear reset bit
        out   092H,al             ;clear reset line
        or    al,000000010B       ;set reset bit
        out   092H,al             ;set reset line
hlt_lp: hlt                       ;wait for reset
        jmp hlt_lp
fast_reset    endp
```

Figure 9-5

```
;This procedure gates A20 to allow A20 to toggle as needed.

set_gate_a20 proc uses ax
        mov    al,02H
        out    092H,al            ;set a20 gate
        ret
set_gate_a20    endp

;This procedure clears the gate A20 to force A20 to always be 0.

clear_gate_a20 proc uses ax
        mov    al,00H
        out    092H,al            ;clear a20 gate
        ret
clear_gate_a20    endp
```

Figure 9-6

```
;This procedure maps an 8 byte I/O range to the X Bus.  On entry AL
;contains the index for which XBE register to use, and AH holds address
;bits 9 through 3 of the starting I/O address space.

;In the case of a floppy disk controller, AH will have a value of 07EH.

map_xbus_io proc uses ax
        out  022H,al            ;program INDEX register
        or   ah,080H            ;set enable bit
        xchg al,ah              ;move I/O range into AL
        out  023H,al            ;enable X Bus I/O range
        ret
map_xbus_io    endp

;This procedure maps a DMA channel (0,1,2 or 3) to the X Bus.  On entry CL
;contains the DMA channel to be mapped.

;In the case of a floppy disk controller, AX will have a value of 00002H.

map_xbus_dma proc uses ax
        mov  ah,1
        rol  ah,cl              ;set correct mask bit
        mov  al,021H
        out  022H,al            ;program INDEX register for XBD
        in   al,023H            ;read in current register settings
        or   al,ah              ;set new X-Bus DMA bit
        out  023H,al            ;write new value
        ret
map_xbus_dma    endp
```

Figure 9-7

```
;This procedure sets the CPU clock stop bit in the Power Save Function
;Register (PSFR).

cpu_stop_clock proc uses ax
        mov  al,023H
        out  022H,al            ;point INDEX register to PSFR
        xor  al,al
        out  023H,al            ;clear cpu stop clock bit
        inc  al
        out  023H,al            ;set cpu stop clock bit. CLOCK STOPS!
        nop                     ;system continues here on interrupt
        ret
cpu_stop_clock   endp

;This procedure sets the System Standby Mode bit in the Power Save Function
;Register (PSFR).

system_standby proc uses ax
        mov  al,023H
        out  022H,al            ;point INDEX register to PSFR
        xor  al,al
        out  023H,al            ;clear system standby bit
        mov  al,002H
        out  023H,al            ;set system standby bit. CLOCKS STOP!
        nop                     ;system continues here on interrupt
        ret
system_standby    endp
```

Figure 9-8

```
;This procedure initializes (to 00000H) both sets of EMS registers
;and enables global translations (GTE bit).

init_enable_ems proc uses ax
        mov     al,018H
        out     022H,al         ;point INDEX register to ARL
        mov     al,010H
        out     023H,al         ;enable I/O decoding for 0208H & 0209H
        xor     ax,ax
        mov     bx,ax           ;initialize bx and ax
ems_lp: mov     dx,00209H
        out     dx,al           ;enable current 4 register set
        xor     ax,ax
        mov     dx,00208H       ;clear first register of set
        out     dx,ax
        mov     dx,04208H       ;clear second register of set
        out     dx,ax
        mov     dx,08208H       ;clear third register of set
        out     dx,ax
        mov     dx,0C208H       ;clear fourth register of set
        out     dx,ax
        add     bx,4            ;move to next 4 register set
        mov     ax,bx
        cmp     bl,010000000B   ;have both sets been written?
        jne     ems_lp          ;no, write the next four registers
        mov     dx,00209H       ;yes, enable global translations
        out     dx,al           ;AL has 010000000B, which sets GTE
        ret

init_enable_ems    endp
```

Figure 9-9

```
kybd_data       equ     060H            ;data port for keyboard
kybd_cntl       equ     061H            ;control port for keyboard
                                        ;only bits 7 & 6 functional
;This procedure initializes both the Am286ZX/LX processor and XT keyboard.

xt_kybd_initialization proc uses ax
        call    mask_off_kybd_int       ;calls a procedure to program the
                                        ;interrupt controllers to not ignore
                                        ;keyboard interrupts
        mov     al,003H
        out     022H,al                 ;point INDEX register to PSR
        in      al,023H                 ;read in current PSR settings
        or      al,000000100B           ;set XTE bit
        out     023H,al                 ;enable XT keyboard interface

        mov     al,010000000B           ;clear interrupt logic
        out     kybd_cntl,al            ;and begin reset command

        call    pause_12_5_ms           ;pause for 12.5mS

        mov     al,011000000B           ;enable keyboard communications
        out     kybd_cntl,al            ;

        call    pause_500_ms            ;pause for 500mS for keyboard basic
                                        ;assurance test (BAT) to finish
        in      al,kybd_data            ;read BAT results
        cmp     al,0AAH                 ;0AAH means BAT passed
        jne     kybd_error              ;handle keyboard error

        call    setup_kybd_buffer       ;setup keyboard buffers
        call    setup_kybd_int          ;initialize for keyboard ISR
        call    enable_kybd_int         ;calls a procedure to program the
                                        ;interrupt controllers to service
                                        ;keyboard interrupts
        mov     al,011000000B           ;enable keyboard communications
        out     kybd_cntl,al
        ret                             ;return from procedure

kybd_error:
;code to report error to system.
;leave keyboard interface as is and exit
        ret                             ;return
xt_kybd_initialization    endp

;This procedure reads in a scan code form the keyboard.
;Returns code in AH

xtkb_read proc
        in      al,kybd_data            ;read in data from port 060H
        mov     ah,al                   ;save scan code in AH
        in      al,kybd_cntl            ;read current control port settings
        or      al,010000000B           ;set PB7
        out     kybd_cntl,al            ;clear logic to enable next scan code
        ret
xtkb_read endp
```

CHAPTER 10

SYSTEM SUPPORT PRODUCTS

HEWLETT-PACKARD IN-CIRCUIT EMULATOR (ICE)

Description

The HP 64700 Series Emulation/Analyzers provide real-time transparent emulation and analysis for the AMD 80C286, Am286ZX and Am286LX microprocessors. These stand-alone emulators/analyzers are self-contained vehicles which can be integrated into the HP 64000-PC Personal Integration Environment by hosting the emulator on an IBM PC-compatible, such as the HP Vectra PC. For large team-oriented designs, the emulators can be integrated into the even more powerful HP 64000-UX Advanced Integration Environment hosted on HP 9000 Series 300 computers. Two RS-232-C serial ports are provided for connection to PC, workstation, or basic terminal. One of the RS-232-C ports can be reconfigured to an RS-422 port for high-speed code download.

The emulators have either 128 kb or 512 kb of high-speed, dual-port emulation memory. The emulators run at a maximum processor speed of 20 MHz with no wait states in target memory or in emulation memory. Memory can be mapped in 1-kb blocks, and configured as either emulation or target RAM, emulation or target ROM, or guarded memory. The emulator checks and optionally breaks on writes to ROM or guarded memory.

For designs involving multiple microprocessors, a Coordinated Measurement Bus allows synchronization and cross triggering of up to 32 HP 64700 Series emulators/analyzers. Each emulator has an internal emulation analyzer for tracing the address bus, data bus, and status lines of the processor. Up to eight hardware resources, each consisting of address, data, and status event comparators, can be combined in several fashions. Analysis features allows triggering on a sequential specification, an address or data range with flexibility for positioning the trigger in the trace listing.

Features

BASIC

- 128-kb or 512-kb dual-port emulation memory mappable in 1-kb blocks
- 32 software breakpoints
- Synchronized operation and cross triggering between multiple emulators for designs involving multiple processors
- Software code coverage analysis
- 80-channel emulation analyzer
- 1024-state trace buffer
- AMD 80C287 math coprocessor support
- Optional HP 9000 Series 300 softkey user interface which supports symbolic debug
- Optional PC user interface, which provides symbolic debug and a friendly windowed user environment

PROCESSOR SPECIFIC

- Disassembly of 80286/80287 instruction set
- Real- and protected-mode support
- Dequeued trace list
- Real-time operation to 20 MHz
- Zero wait state operation to 20 MHz in target or emulation memory

Accessories Supplied

Each HP 64700 Series Emulator/Analyzer includes a demo board for out-of-circuit operation, a sticker describing the datacomm switches on the rear of the unit, power cord, and operator manuals.

PROCESSOR COMPATIBILITY

HP 64700 Series Emulator/Analyzer is compatible with AMD 80C286, Am286LX or Am286ZX processors.

Ordering Information

Model	Description
64766A/B	80C286 Emulator/Analyzer
64766S	80C286 Hosted User Interface
Opt 004	HP 9000/300 Interface
Opt 006	PC Interface
647xxA	Am286ZX/LX Emulator/Analyzer
647xxS	Am286ZX/LX Hosted User Interface
Opt 004	HP 9000/300 Interface
Opt 006	PC Interface

For More Information

Contact:
Hewlett-Packard Company
Attn: Marketing, Sales Development
P. O. Box 617
Colorado Springs, CO 80901-0617
USA
Telephone: 719-590-5801 (International)
Telephone: 1-800-447-3282 (North America)

PHOENIX TECHNOLOGIES SYSTEM BIOS AND EMS 4.0 DRIVER

BIOS Support for Processor/Chip Set Integration

New-generation processor/chip set integrations will play an increasingly vital role in PC system design. PhoenixBIOS supports these advanced "mother-boards-on-a-chip" fast and effectively. With a new chip set interface, independent chip set support modules can be developed quickly and "plugged" into an insulated BIOS core code. This design reduces

BOS-build time, while raising quality control because of its unchanging core code. The CONSTRUCT utility lets engineers modify the BIOS without source code.

AMD Am286ZX and Am286LX Integrated Processor Support

The PhoenixBIOS A286ZX supports the highly integrated features of the AMD Am286ZX/LX integrated processor. The Am286ZX/LX integrated processor is an AMD proprietary PC-AT motherboard-on-a-chip plus enhancements. The Am286ZX/LX integrated processor includes the 80C286 microprocessor core, bus controller, DMA controller, interrupt controller, clock generation, power management functions, counter/timer functions, and real-time clock—all on a single CMOS device. It directly interfaces to DRAM, the AMD 80C287 math coprocessor, BIOS ROM, keyboard controller, and drives two bus slots without buffers or latches. The Am286ZX/LX integrated processor is ideal for the design of hand-held, laptop, notebook, and other industry-standard AT personal computers where performance, size, power consumption, and cost are critical factors.

PhoenixBIOS Features

- AMD 80C287 math coprocessor support
- 6- to 16-MHz with BIOS timings designed for clock independence
- Adjustable bus clock speed for total flexibility
- BIOS SETUP user interface
- ROM or diskette-based configuration tool with new SETUP system information page
- Chip set configuration utility (CCU) for development and debugging
- BIOS code size: 64 kb or 126 kb (Full source code available)
  choice of configuration: Two 27256 EPROMs
  Four 27128 EPROMs
  One 27512 EPROM
  Two 27512 EPROMs (for 128-kb BIOS)
- Diskette drive available: 5.25-inch (160 kb, 180 kb, 320 kb, 360 kb, 720 kb, 1.2 Mb), and 3.5-inch (720 kb and 1.44 Mb)
- PC manufacturers specify up to 47 fixed-disk drive types in BIOS disk tables. Users can specify up to two drive types in CMOS RAM
- Support for ST506 (17 sectors per track), RLL (26 sectors per track), ARLL (36 sectors per track), and ESDI (34/36 sectors per track) devices
- 101/102-key and 83/84-key keyboard support with keyboard controller speed of 6–12 MHz. Keyboard-based CPU clock-speed switching. 8-level adjustable key-click volume. Support for international "hot keys." Supports auto-sensing keyboards, including Maxiswitch or BTC. Phoenix keyboard controller firmware (8042-compatible) available
- MDA and CGA standard video support
- Phoenix VGA, EGA optional video available. Video BIOS can be included in system BIOS ROMs to save EPROM cost and board space
- User-terminated diagnostic RAM test and IBM compatible beep-cost POST diagnostic system boot and run-time messages

- Four serial and four parallel I/O ports
- Standard utilities have been expanded to include SETUP, CONSTRUCT, ROMINFO, and HDFORM
- Standard chip set feature support includes: Hardware EMS, ROM shadowing, Fast A20 switching, Dynamic memory configuration, speed switching, video memory backfill
- Enhanced NMI handling to prevent system lockup
- Server configurations available without keyboard and video
- Network workstation configurations available without disk or diskette drives
- Custom non-standard I/O device support available
- PS/2-style mouse operation and multilevel password security

Contact For More Information

Eastern Region Office
Phoenix Technologies, LTD.
846 University Avenue
Norwood, MA 02062-3950
Tel: 617-551-4000
FAX: 617-551-3750

Western and Pacific Region Office
Phoenix Technologies Ltd.
40 Airport Parkway
San Jose, CA 95110
Tel: 408-452-6500
FAX: 408-452-1985

Asian Region Office
Phoenix Technologies Ltd.
Walsin Lihwa Buildilng, 6th Floor
675 Min Sheng East Road
Taipei, Tiawan, ROC
Tel: 886 2 718 8956
FAX: 886 2 718 8955

European Region Office
Phoenix Technologies Ltd.
Les Algorithmes ARISTOTE B
2000 Route des Lucioles
Sophia Antipolis
96560 Valbonne, France
Tel: 33-93-65-4600
FAX: 33-93-95-8102

AMP INCORPORATED 216-POSITION MCR TAPEPAK PRODUCTION SOCKET

Introduction

Measuring only 5.74 mm (x224") above the printed circuit board, this new production socket was specifically designed by AMP for JEDEC-approved TapePak molded carrier ring (MCR) packages. It also accommodates other leaded packages (including TAB) up to 28 mm square, with unformed leads trimmed to 30.35 ± .127 mm (1.195 ± .005") tip-to-tip.

Component leads are formed when installed in the socket. Installation consists of orienting the IC package on the socket base (insulating barriers assure accurate lead placement) and pressing on the cover until it snaps into place. Once in place, the cover provides package retention.

Thermal expansion/contraction between the package and printed circuit board are accommodated by flexure at the contact base to avoid micromotion at the socket/package-lead interface.

The two scalloped areas in the base of each contact helps form strong solder fillets during reflow soldering. Hot bar reflowing is recommended, but vapor phase reflowing can be used.

Features

- Low profile
- Surface mount
- Reliable electrical contact
- Forms strong solder fillets
- Compatible with:
  - —JEDEC approved TapePak
  - —Tape AutomatedBonding (TAB) packages
  - —Any JEDEC flatpack less than 28 mm (1.1") square with unformed leads trimmed to 30.35 ± .127 mm (1.195 ± .005") tip-to-tip

Contact For More Information

AMP Incorporated
Harrisburg, PA 17105
Phone: 717-564-0100
TWX: 510-657-4110
Product Information Center
Phone: 1-800-522-6752

ADVANCED MICRO DEVICES MERCURY/286ZLX DEVELOPMENT KIT

General Description

The Mercury/286ZLX is a 16-MHz, IBM-compatible PC-AT system development kit. It provides the hardware, software, documentation, and manufacturing information necessary

to evaluate the Am286ZX processor or the Am286LX (Am286ZX/LX) processor for use in a system design. The Mercury/286ZLX development kit consists of a PC-AT demonstration board and a full set of board design and manufacturing documentation to reduce the system manufacturer's development cycle and provide a cost effective design using the Am286ZX/LX PC-AT motherboard-on-a-chip.

The Mercury/286ZLX utilizes Surface Mount Technology (SMT) in its manufacturing process to minimize board space. The low component count coupled with the small footprint makes this design a unique example of a small form-factor PC-AT. The AT components exist in a 5.4 x 4.0 inch area and the expansion slots occupy a 3.1 x 8.0 inch area. The small component count is a direct result of the high integration level of the Am286ZX/LX processor.

The Mercury/286ZLX development board is based on AMD's first integrated processor, the Am286ZX/LX processor. For additional performance, an AMD 80C287 coprocessor is provided and is directly interfaced to the Am286ZX processor. The Mercury/286ZLX motherboard drives three 16-bit expansion slots directly, which eliminates the need for external buffers and greatly reduces the component count. The Mercury/286ZLX also directly interfaces to two banks of DRAM, an 8-bit EPROM, and an 8742 Keyboard Controller. These features are all provided in a mini (5.4 x 4.0) AT form factor.

The Am286ZX/LX processor provides the page mode/interleaved memory control for the on-board DRAM. Four single in-line memory modules (SIMMs) support up to 16 Mb of on board memory when 4-Mbit x 9 DRAMS are used. The Am286ZX/LX processor supports the Lotus/Intel/Microsoft Expanded Memory Specification, LIM EMS 4.0.

The Mercury/286ZLX PC-AT development kit provides the tools for system designers to develop and manufacture a highly integrated, small form-factor PC-AT system.

The Mercury/286ZLX is a 286 based PC-AT with minimal chip count and maximum performance. This design is ideal for small footprint PC-AT's, laptops, notebooks, and "embedded computer" applications.

Note: The Am286ZX/LX processor will drive two 16-bit AT expansion slots without the need for external buffers. Three 16-bit AT expansion slots, depending on the loading of the add-in cards, may also be driven without the need for the buffers or latches. Please consult the Mercury Technical Reference Manual for more information.

Features

- 16-MHz Am286ZX/LX integrated processor-based IBM PC-AT Compatible System Development Board
- Am286ZX/LX integrated processor
- 4-Mb of on-board page mode/interleaved DRAM, expandable to 16 Mb using 4-Mb SIMMs
- Supports Lotus/Intel/Microsoft Expanded Memory Specification, LIM EMS 4.0
- Flexible clock speeds up to 16 MHz
- AMD 80EC287 math coprocessor in PLCC package
- PhoenixBIOS A286ZX BIOS from Phoenix Technologies, LTD
- Single 8-bit EPROM (BIOS) using Am27C512 or Am27C010 incorporating "Shadow RAM Mapping"

- 5.4 x 4.0 inch PWB aligns with AT motherboard mounting holes to fit into a "Baby AT" PC case
- Extended output drive capability eliminates the need for external buffers and latches
- Three expansion slots, four SIMM sockets, reset switch, speaker, and standard LED PC-front-end indicators on board
- Uses all Surface Mount Technology (SMT)

Functional Description

The Mercury/286ZLX PC-AT System Board is a design prepared by AMD to provide system designers with an illustrative Am286ZX/LX integrated processor design example. The Am286ZX/LX integrated processor allows the design of systems with minimal space requirements. This reduces the board cost dramatically while maintaining processing power and all the features of a 286-based PC-AT.

An AMD 80C287 numeric coprocessor is provided on board to enhance performance, and is interfaced synchronously to the Am286ZX/LX integrated processor.

Two banks of DRAM with 4 Mb are supported by the Mercury/286ZLX. These banks will support up to 16-Mb of memory through the use 4M x 9 SIMMs. DRAM memory is configured as 640 kb of system memory, with the remainder configured as extended and/or expanded memory (LIM EMS 4.0).

The DRAM is organized as banks consisting of 16 bits of data and 2 bits of parity information. The 16 bits of data are separated into High and Low order bytes with 1 odd parity bit for each byte. The Mercury/286ZLX board implements this configuration with four 30 pin DRAM single in-line memory modules (SIMMs). Each memory bank consists of two SIMMs, which are available in 256K x 9, 1M x 9, or 4M x 9 configurations. The Mercury/286ZLX board uses a DRAM page mode/interleaved scheme when configured with 1 Mb, 4 Mb or 16 Mb of memory. Board performance is maximized in this scheme where the DRAM $\overline{RAS}$ signal is held active and the $\overline{CAS}$ signals of two memory banks are interleaved so that the $\overline{CAS}$ active time of one bank occurs during the $\overline{CAS}$ precharge time of the other bank.

Mercury/286ZLX supports an 8-bit EPROM directly via 64K x 8 (512K) or 128K x 8 (1 M) devices for the Phoenix Technologies' A286ZX BIOS. A 32-pin socket is provided to accommodate either the 28-pin 512K (Am27C512) or 32-pin 1 M (Am27C010) EPROM. The "Shadow RAM Mapping" feature of the Am286ZX/LX processor may be used to enhance performance by executing the ROM-based code through faster DRAM.

The Mercury/286ZLX may be operated in a low-power mode through the low- power features of the Am286LX integrated processor. The 80C286 core of the Am286LX processor is comprised of completely static circuitry. The clock to the processor may be stopped at any point and held there indefinitely or placed in standby mode. The Power Save Function Register provided in the Am286LX processor enables the CPU Stop Clock Mode or the System Shutdown Mode.

The Mercury/286ZLX incorporates three 16-bit AT bus slots. These expansion slots will support any external hardware that has an 8- or 16-bit PC-AT bus interface. These add in cards (VGA and Disk Controllers for example) are driven directly by the Am286ZX/LX integrated processor.

An 8742AH Universal Peripheral Interface 8-bit Slave Microcontroller is provided for the keyboard interface and may operate at 8 or 12 MHz. The standard AT keyboard is directly

interfaced through the 5-pin DIN connector. The ROM based microcontroller has been programmed to support many front-end features through the use of on-board connectors including:

- Turbo LED indicator
- Turbo LED switch
- Manufacturer's ID (Signature Mode)
- Color/Monochrome switch
- Keyboard inhibit switch

On-board connectors also allow for the attachment of power-on and hard drive access LED indicators and a 2.25" permanent magnet speaker.

The Mercury/286ZLX processor design is based on surface mount technology (SMT). All components are SMT types except for the 8-bit EPROM, oscillators, DRAMs, and connectors. Chip capacitors (SMT) are placed on both sides of the board to reduce the component density.

PC/AT COMPATIBILITY

The Mercury/286ZLX has been designed to be 100% compatible with IBM PC-AT systems. The Mercury/286ZLX will run all PC-AT compatible software and is configurable with all hardware that is compatible with IBM PC-AT systems.

EMI CONSIDERATIONS

AMD has incorporated several PCB layout and design considerations into the Mercury/286ZLX PC-AT board. A range of decoupling capacitor values have been used to supply the high frequency current requirements of ICs when outputs switch. Capacitors are also placed on all power connections that interface to external cables (i.e., keylock, speaker, and power LED connector). Ferrite beads and capacitors to keyboard ground (KGND) have been placed in series with the clock, data, and +5 V lines to the keyboard connector. An isolated KGND has been provided to isolate RF currents on the logic ground plane from the keyboard cable. Additional capacitors were used on many high-frequency clock lines for harmonic frequencies above 200 MHz.

Ordering Information

VALID COMBINATIONS

The Mercury/286ZLX PC-AT Development Kit consists of a development board and documentation kit. The part numbers for each item are listed in the following table. Consult the local AMD sales office for availability and delivery information.

MERCURY/286ZLX PC-AT Development Kit

Part Number: Am286ZLX/HW/MRYB10
Price: $ 995.00
Includes: Mercury/286ZLX System Board
Mercury/286ZLX Data Sheet
Am286ZX/LX Integrated Processor Overview Brochure
Mercury/286ZLX Technical Reference Manual
Mercury/286ZLX Board Schematics (B-size)
Mercury/286ZLX Integrated Processor Bill of Materials
AMD Am286 ZX/LX Integrated Processor Data Sheet
Mercury Board Overlay Drawing (silkscreen layer)
Phoenix A286ZX BIOS Data Sheet
Hewlett-Packard Am286ZX/LX Integrated Processor In-Circuit Emulator Data Sheet
AMP Socket Data Sheet

MERCURY/286ZLX PC/AT Documentation Kit

Order Number 15343 (includes all documentation above).

Warranty Information

HARDWARE WARRANTY

Advanced Micro Devices, Inc. (AMD) warrants that the AMD product enclosed with an AMD Limited Warranty statement, when used in normal operating conditions, will conform to the manufacturer's specifications and be free from defect in workmanship and materials for a period of 90 days from the date of original purchase. This warranty does not extend beyond the first purchaser of said product.

SOFTWARE AND DOCUMENTATION WARRANTY

Definitions:

Software—consists of the software received by Customer in the product package;

Documentation—consists of all manuals and other material which are made available by AMD and received by Customer in conjunction with a hardware or software product. Customer acknowledges that AMD and/or its licensors are the owners of copyright in the Software and Documentation.

CUSTOMER SUPPORT

AMD offers product technical support during 8 a.m. to 5 p.m. (PST) through the Applications Hotline, 800-222-9323. Contact the local AMD sales office to request additional Technical Customer Assistance.

APPENDIX A

STANDARD LOGIC LITERATURE REFERENCES

The following list specifies the Publication Identification (PID) numbers for additional documentation for the standard logic blocks contained in the Am286ZX/LX integrated processor, and for external devices used with the device. This documentation includes data sheets, technical manuals, and programming references.

80C286 PROCESSOR

Data Sheet 11625
Programmer's Reference 14554

AM95C17 DMA CONTROLLER

Data Sheet 11339
Technical Manual 00092

AM82C54 COUNTER/TIMER

Data Sheet 07840

AMD 80C287, AMD 80EC287 MATH COPROCESSORS

AMD 80C287 Data Sheet 11671
AMD 80EC287 Data Sheet 11959
Programmer's Reference 14554

APPENDIX B

CONFIGURATION REGISTER REFERENCE

Lonestar Reserved Register 1

Name: LRR1
Index: 00 (00H)
Default: AAH (read only)

This is a reserved register to be used for silicon identification purposes. The high four bits (7:4) will stay constant. The low four bits (3:0) may change with silicon revisions. **This is a read only register.**

7	6	5	4	3	2	1	0
1	0	1	0	1	0	1	0

Clock Configuration Register 1

Name: CCR1
Index: 01 (01H)
Default: 00H

This register contains the programmable configuration options for AT-bus state machine clocking and CPU clocking.

7	6	5	4	3	2	1	0
FBE	BD1	BD0	BS	0	PD1	PD0	PS

Bit(s)	Name	Description
7	FBE	Fast Bus Enable (default is 0) 1 = This bit enables the echoing of all bus transactions onto the S Bus. This mode can only be enabled if ATSM clock = CPU clock. 0 = The ALE on the AT bus is delayed compared to Local bus, and only S-bus target cycles enable the S-bus buffers.
6:5	BD1:0	AT State Machine Clock Divisor Select (default is 00) These 2 bits select the divisor for the AT state machine clock. 0 0 = Divide by 4 0 1 = Divide by 1 1 0 = Divide by 2 1 1 = Divide by 3
4	BS	AT State Machine Clock Source Select (default is 0) This bit selects the source for the AT state machine clock. 1 = IOCLK 0 = PROCLK
3	Reserved	This bit is reserved; do not program to a 1 (default is 0)
2:1	PD1:0	Processor Clock Divisor Select (default is 0) These 2 bits select the divisor for the internal CPU clock. 0 0 = Divide by 4 0 1 = Divide by 1 1 0 = Divide by 2 1 1 = Divide by 16
0	PS	Processor Clock Source Select (default is 0) This bit selects source for the internal CPU clock. 1 = IOCLK 0 = PROCLK

Clock Configuration Register 2

Name: CCR2
Index: 02 (02H)
Default: C0H

This register contains the programmable configuration options for the keyboard and coprocessor clocks.

7	6	5	4	3	2	1	0
KS1	KS0	KD1	KD0	0	CD1	CD0	CS

Bit(s)	Name	Description
7:6	KS1:0	Keyboard Clock Source Select (default is 11) These 2 bits select the source for the keyboard clock. 0 0 = PROCLK 0 1 = IOCLK 1 0 = Inverted AT SYSCLK 1 1 = 14.318 MHz (F14MHZ input pin)
5:4	KD1:0	Keyboard Clock Divisor (default is 00) These 2 bits select the divisor for the keyboard clock. 0 0 = Divide by 4 0 1 = Divide by 1 1 0 = Divide by 2 1 1 = Divide by 16
3	Reserved	This bit is reserved; do not program to a 1 (default is 0)
2:1	CD1:0	Coprocessor Clock Divisor Select (default is 00) These 2 bits select the divisor for the AMD 80C287 math coprocessor clock. 0 0 = Divide by 4 0 1 = Divide by 1 1 0 = Divide by 2 1 1 = Divide by 3
0	CS	Coprocessor Clock Select (default is 0) This bit selects the source for the 80C287 clock. 1 = IOCLK 0 = PROCLK

Peripheral Selection Register

Name: PSR
Index: 03 (03H)
Default: 00H

This register contains DMA controller configuration options for wait states and clocking. Also, the AT/XT keyboard select option is here.

7	6	5	4	3	2	1	0
D16M1	D16M0	D8M1	D8M0	0	XTE	DMC1	DMC0

Bit(s)	Name	Description
7:6	D16M1:0	16-bit DMA Wait State selection (default is 00) These 2 bits select the number of wait states in cycles generated by the 16-bit DMA controller. 0 0 = 1 wait state 0 1 = 2 wait states 1 0 = 3 wait states 1 1 = 4 wait states
5:4	D8M1:0	8-bit DMA Wait State selection (default is 00) These 2 bits select the number of wait states in cycles generated by the 8-bit DMA controller. 0 0 = 1 wait state 0 1 = 2 wait states 1 0 = 3 wait states 1 1 = 4 wait states
3	Reserved	This bit is reserved; do not program to a 1 (default is 0)
2	XTE	Keyboard Interface selection (default is 0) This bit selects which keyboard interface is enabled. 1 = XT-type keyboard 0 = AT-type keyboard
1:0	DMC1:0	DMA Clock selection (default is 00) These 2 bits select the clock source for both the 8- and 16-bit DMA controllers. 0 0 = SYSCLK/2 0 1 = SYSCLK 1 0 = AT State Machine clock 1 1 = CPU Clock/2 (CPU state clock)

Command Delay Register

Name: CDR
Index: 04 (04H)
Default: FFH

This register specifies the number of command delays for the various types of AT-bus cycles. Note that the delay is specified as the number of AT state machine clocks, which is half the period of SYSCLK.

7	6	5	4	3	2	1	0
M16C1	M16C0	M8C1	M8C0	I16C1	I16C0	I8C1	I8C0

Bit(s)	Name	Description
7:6	M16C1:0	AT-bus 16-bit memory cycle Command Delay (default is 11) 0 0 = 0 clock delay 0 1 = 1 clock delay 1 0 = 2 clock delay 1 1 = 3 clock delay
5:4	M8C1:0	AT-bus 8-bit memory cycle Command Delay (default is 11) 0 0 = 0 clock delay 0 1 = 1 clock delay 1 0 = 2 clock delay 1 1 = 3 clock delay
3:2	I16C1:0	AT-bus 16-bit I/O cycle Command Delay (default is 11) 0 0 = 0 clock delay 0 1 = 1 clock delay 1 0 = 2 clock delay 1 1 = 3 clock delay
1:0	I8C1:0	AT-bus 8-bit I/O cycle Command Delay (default is 11) 0 0 = 0 clock delay 0 1 = 1 clock delay 1 0 = 2 clock delay 1 1 = 3 clock delay

Wait State Register

Name: WSR
Index: 05 (05H)
Default: FFH

This register specifies the number of wait states for the various types of AT-bus cycles.

7	6	5	4	3	2	1	0
M16W1	M16W0	M8W1	M8W0	I16W1	I16W0	I8W1	I8W0

Bit(s)	Name	Description
7:6	M16W1:0	AT-bus 16-bit memory cycle Wait States (default is 11) 0 0 = 0 wait states 0 1 = 1 wait state 1 0 = 2 wait states 1 1 = 3 wait states
5:4	M8W1:0	AT-bus 8-bit memory cycle Wait States (default is 11) 0 0 = 2 wait states 0 1 = 3 wait states 1 0 = 4 wait states 1 1 = 5 wait states
3:2	I16W1:0	AT-bus 16-bit I/O cycle Wait States (default is 11) 0 0 = 0 wait states 0 1 = 1 wait state 1 0 = 2 wait states 1 1 = 3 wait states
1:0	I8W1:0	AT-bus 8-bit I/O cycle Wait States (default is 11) 0 0 = 2 wait states 0 1 = 3 wait states 1 0 = 4 wait states 1 1 = 5 wait states

Internal Peripheral Wait State Register 1

Name: IPWS1
Index: 06 (06H)
Default: 44H

This register defines the number of wait states executed when accessing the internal interrupt or DMA logic groups.

7	6	5	4	3	2	1	0
IIWS3	IIWS2	IIWS1	IIWS0	IDWS3	IDWS2	IDWS1	IDWS0

Bit(s)	Name	Description
7:4	IIWS3:0	Internal Interrupt Wait States (default is 0100, for four wait states) These 4 bits define 0-15 wait states, with a straight binary encoding, for the interrupt logic group.
3:0	IDWS3:0	Internal DMA Wait States (default is 0100, for four wait states) These 4 bits define 0-15 wait states, with a straight binary encoding, for the internal DMA logic group.

Internal Peripheral Wait State Register 2

Name: IPWS2
Index: 07 (07H)
Default: 44H

This register defines the number of wait states executed when accessing the internal counter/timer or RTC/misc logic groups.

7	6	5	4	3	2	1	0
ICWS3	ICWS2	ICWS1	ICWS0	IMWS3	IMWS2	IMWS1	IMWS0

Bit(s)	Name	Description
7:4	ICWS3:0	Internal Counter/Timer Wait States (default is 0100, for four wait states) These 4 bits define 0-15 wait states, with a straight binary encoding, for the counter/timer logic group.
3:0	IMWS3:0	Internal Misc/RTC Wait States (default is 0100, for four wait states) These 4 bits define 0-15 wait states, with a straight binary encoding, for the internal RTC logic group, along with other miscellaneous AT registers.

Internal Peripheral Command Delay Register

Name: IPCD
Index: 08 (08H)
Default: 00H

This register defines the number of command delays executed when accessing the internal peripheral logic groups.

7	6	5	4	3	2	1	0
IMCD1	IMCD0	ICCD1	ICCD0	IICD1	IICD0	IDCD1	IDCD0

Bit(s)	Name	Description
7:6	IMCD1:0	Internal Misc/RTC Command Delays (default is 00) These 2 bits define 0-3 command delays, with a straight binary encoding, for the internal RTC logic group, along with other miscellaneous AT registers.
5:4	ICCD1:0	Internal Counter/Timer Command Delays (default is 00) These 2 bits define 0-3 command delays, with a straight binary encoding, for the counter/timer logic group.
3:2	IICD1:0	Internal Interrupt Command Delays (default is 00) These 2 bits define 0-3 command delays, with a straight binary encoding, for the interrupt logic group.
1:0	IDCD1:0	Internal DMA Command Delays (default is 00) These 2 bits define 0-3 command delays, with a straight binary encoding, for the internal DMA logic group.

Bus Control Register

Name: BCR
Index: 09 (09H)
Default: 00H

This register controls the AT system bus (S-bus) enable and reset functions. Also the strong pull-up control and full I/O decode configuration bits are here.

7	6	5	4	3	2	1	0
KSPUL	0	0	0	FIO	SRST	0	BCON

Bit(s)	Name	Description
7	KSPUL	Kill Strong Pull-ups (default is 0) Setting this bit will turn off the internal strong pull-ups on the following pins: $\overline{\text{MEMCS16}}$, $\overline{\text{IOCS16}}$, $\overline{\text{P0WS}}$, $\overline{\text{IOCHK}}$, IOCHRDY, $\overline{\text{MASTER}}$, and $\overline{\text{REFRESH}}$. 1 = 0 =
6:4	Reserved	These bits are reserved; do not program to a 1 (default is 0)
3	FIO	Full IO Decode (default is 0) This bit controls the IO decode for peripheral chip selects. 1 = Full 16-bit IO decode 0 = Normal AT style 10-bit IO decode
2	SRST	S-Bus RESETDRV control (default is 0) The AT system bus reset pin, RESETDRV, can be forced active under software control through this configuration bit. 1 = Force RESETDRV active 0 = Normal RESETDRV
1	Reserved	This bit is reserved; do not program to a 1 (default is 0)
0	BCON	S-Bus Connect (default is 0) This bit controls the S-bus address, data, and control buffers. 1 = Connect S Bus, enabling the buffers 0 = Disconnect S Bus, three-stating the buffers

Bus Master Control Register

Name: BMCR
Index: 10 (0AH)
Default: 00H

This register contains the configuration bits for Bus Master Mode. A standard motherboard/main CPU implementation will never need to access this register.

7	6	5	4	3	2	1	0
0	0	0	HIE	HME	AS	BA	BM

Bit(s)	Name	Description
7:5	Reserved	These bits are reserved; do not program to a 1 (default is 0)
4	HIE	Host IO Enable (default is 0) 1 = If IO address matches Page Register (BMPR), go to S Bus with access instead of doing local I/O cycle 0 = Normal operation, don't redirect I/O
3	HME	Host Memory Enable (default is 0) 1 = If Memory address matches Page Register (BMPR), go to S Bus with access instead of doing local memory cycle 0 = Normal operation, don't redirect any memory accesses
2	AS	Acquire Status (default is 0, READ ONLY bit) Reading this bit senses the state of bus acquisition in bus master mode. 1 = Host S Bus is acquired, this device has control 0 = Host CPU has control of the S Bus
1	BA	Bus Acquire (default is 0) This bit can initiate a bus request handshake in order to gain control of the host's S Bus. 1 = Acquire S Bus (0 to 1 transition activates DRQ3, then waits for $\overline{\text{DACK3}}$) 0 = Release S Bus
0	BM	Bus Master Mode Select (default is 0) This bit controls the pin driver directions of $\overline{\text{MASTER}}$, DRQ3, and $\overline{\text{DACK3}}$. It also enables all bus master mode operations. 1 = Bus Master Mode 0 = Normal Mode

Bus Master Page Register

Name: BMPR
Index: 11 (0BH)
Default: 00H

This register defines which memory or I/O page accesses should be routed out to the AT system bus when the device is in Bus Master Mode.

7	6	5	4	3	2	1	0
BMP7	BMP6	BMP5	BMP4	BMP3	BMP2	BMP1	BMP0

Bit(s)	Name	Description
7:0	BMP7:0	These 8 bits define a 64-kb page for host memory accesses and a 4-byte page for host I/O accesses. The 8 bits in this register are compared with A23:16 for memory accesses and A9:2 for IO accesses. Address comparison is controlled by the HIE and HME bits in BMCR.

Shadow RAM Enable Register 1

Name: SER1
Index: 12 (0CH)
Default: 00H

This register selects which specific 16-kb blocks of memory are to be "shadowed" with the shadow RAM feature.

7	6	5	4	3	2	1	0
SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0

Bit(s)	Name	Description
7	SE7	Shadow Enable 7 (default is 0) 1 = Enable Shadow RAM for 0BC000 - 0BFFFF 0 = Disabled
6	SE6	Shadow Enable 6 (default is 0) 1 = Enable Shadow RAM for 0B8000 - 0BBFFF 0 = Disabled
5	SE5	Shadow Enable 5 (default is 0) 1 = Enable Shadow RAM for 0B4000 - 0B7FFF 0 = Disabled
4	SE4	Shadow Enable 4 (default is 0) 1 = Enable Shadow RAM for 0B0000 - 0B3FFF 0 = Disabled
3	SE3	Shadow Enable 3 (default is 0) 1 = Enable Shadow RAM for 0AC000 - 0AFFFF 0 = Disabled
2	SE2	Shadow Enable 2 (default is 0) 1 = Enable Shadow RAM for 0A8000 - 0ABFFF 0 = Disabled
1	SE1	Shadow Enable 1 (default is 0) 1 = Enable Shadow RAM for 0A4000 - 0A7FFF 0 = Disabled
0	SE0	Shadow Enable 0 (default is 0) 1 = Enable Shadow RAM for 0A0000 - 0A3FFF 0 = Disabled

Shadow RAM Enable Register 2

Name: SER2
Index: 13 (0DH)
Default: 00H

This register selects which specific 16-kb blocks of memory are to be "shadowed" with the shadow RAM feature.

7	6	5	4	3	2	1	0
SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8

Bit(s)	Name	Description
7	SE15	Shadow Enable 15 (default is 0) 1 = Enable Shadow RAM for 0DC000 - 0DFFFF 0 = Disabled
6	SE14	Shadow Enable 14 (default is 0) 1 = Enable Shadow RAM for 0D8000 - 0DBFFF 0 = Disabled
5	SE13	Shadow Enable 13 (default is 0) 1 = Enable Shadow RAM for 0D4000 - 0D7FFF 0 = Disabled
4	SE12	Shadow Enable 12 (default is 0) 1 = Enable Shadow RAM for 0D0000 - 0D3FFF 0 = Disabled
3	SE11	Shadow Enable 11 (default is 0) 1 = Enable Shadow RAM for 0CC000 - 0CFFFF 0 = Disabled
2	SE10	Shadow Enable 10 (default is 0) 1 = Enable Shadow RAM for 0C8000 - 0CBFFF 0 = Disabled
1	SE9	Shadow Enable 9 (default is 0) 1 = Enable Shadow RAM for 0C4000 - 0C7FFF 0 = Disabled
0	SE8	Shadow Enable 8 (default is 0) 1 = Enable Shadow RAM for 0C0000 - 0C3FFF 0 = Disabled

Shadow RAM Enable Register 3

Name: SER3
Index: 14 (0EH)
Default: 00H

This register selects which specific 16-kb blocks of memory are to be "shadowed" with the shadow RAM feature.

7	6	5	4	3	2	1	0
SE23	SE22	SE21	SE20	SE19	SE18	SE17	SE16

Bit(s)	Name	Description
7	SE23	Shadow Enable 23 (default is 0) 1 = Enable Shadow RAM for 0FC000 - 0FFFFF 0 = Disabled
6	SE22	Shadow Enable 22 (default is 0) 1 = Enable Shadow RAM for 0F8000 - 0FBFFF 0 = Disabled
5	SE21	Shadow Enable 21 (default is 0) 1 = Enable Shadow RAM for 0F4000 - 0F7FFF 0 = Disabled
4	SE20	Shadow Enable 20 (default is 0) 1 = Enable Shadow RAM for 0F0000 - 0F3FFF 0 = Disabled
3	SE19	Shadow Enable 19 (default is 0) 1 = Enable Shadow RAM for 0EC000 - 0EFFFF 0 = Disabled
2	SE18	Shadow Enable 18 (default is 0) 1 = Enable Shadow RAM for 0E8000 - 0EBFFF 0 = Disabled
1	SE17	Shadow Enable 17 (default is 0) 1 = Enable Shadow RAM for 0E4000 - 0E7FFF 0 = Disabled
0	SE16	Shadow Enable 16 (default is 0) 1 = Enable Shadow RAM for 0E0000 - 0E3FFF 0 = Disabled

Shadow RAM Protection Register 1

Name: SPR1
Index: 15 (0FH)
Default: 00H

This register selects which specific 16-kb blocks of shadow RAM should be protected from memory writes.

7	6	5	4	3	2	1	0
SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

Bit(s)	Name	Description
7	SP7	Shadow Protect 7 (default is 0) 1 = Protect Shadow RAM at 0BC000 - 0BFFFF 0 = Disabled
6	SP6	Shadow Protect 6 (default is 0) 1 = Protect Shadow RAM at 0B8000 - 0BBFFF 0 = Disabled
5	SP5	Shadow Protect 5 (default is 0) 1 = Protect Shadow RAM at 0B4000 - 0B7FFF 0 = Disabled
4	SP4	Shadow Protect 4 (default is 0) 1 = Protect Shadow RAM at 0B0000 - 0B3FFF 0 = Disabled
3	SP3	Shadow Protect 3 (default is 0) 1 = Protect Shadow RAM at 0AC000 - 0AFFFF 0 = Disabled
2	SP2	Shadow Protect 2 (default is 0) 1 = Protect Shadow RAM at 0A8000 - 0ABFFF 0 = Disabled
1	SP1	Shadow Protect 1 (default is 0) 1 = Protect Shadow RAM at 0A4000 - 0A7FFF 0 = Disabled
0	SP0	Shadow Protect 0 (default is 0) 1 = Protect Shadow RAM at 0A0000 - 0A3FFF 0 = Disabled

Shadow RAM Protection Register 2

Name: SPR2
Index: 16 (10H)
Default: 00H

This register selects which specific 16-kb blocks of shadow RAM should be protected from memory writes.

7	6	5	4	3	2	1	0
SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8

Bit(s)	Name	Description
7	SP15	Shadow Protect 15 (default is 0) 1 = Protect Shadow RAM at 0DC000 - 0DFFFF 0 = Disabled
6	SP14	Shadow Protect 14 (default is 0) 1 = Protect Shadow RAM at 0D8000 - 0DBFFF 0 = Disabled
5	SP13	Shadow Protect 13 (default is 0) 1 = Protect Shadow RAM at 0D4000 - 0D7FFF 0 = Disabled
4	SP12	Shadow Protect 12 (default is 0) 1 = Protect Shadow RAM at 0D0000 - 0D3FFF 0 = Disabled
3	SP11	Shadow Protect 11 (default is 0) 1 = Protect Shadow RAM at 0CC000 - 0CFFFF 0 = Disabled
2	SP10	Shadow Protect 10 (default is 0) 1 = Protect Shadow RAM at 0C8000 - 0CBFFF 0 = Disabled
1	SP9	Shadow Protect 9 (default is 0) 1 = Protect Shadow RAM at 0C4000 - 0C7FFF 0 = Disabled
0	SP8	Shadow Protect 8 (default is 0) 1 = Protect Shadow RAM at 0C0000 - 0C3FFF 0 = Disabled

Shadow RAM Protection Register 3

Name: SPR3
Index: 17 (11H)
Default: 00H

This register selects which specific 16-kb blocks of shadow RAM should be protected from memory writes.

7	6	5	4	3	2	1	0
SP23	SP22	SP21	SP20	SP19	SP18	SP17	SP16

Bit(s)	Name	Description
7	SP23	Shadow Protect 23 (default is 0) 1 = Protect Shadow RAM at 0FC000 - 0FFFFF 0 = Disabled
6	SP22	Shadow Protect 22 (default is 0) 1 = Protect Shadow RAM at 0F8000 - 0FBFFF 0 = Disabled
5	SP21	Shadow Protect 21 (default is 0) 1 = Protect Shadow RAM at 0F4000 - 0F7FFF 0 = Disabled
4	SP20	Shadow Protect 20 (default is 0) 1 = Protect Shadow RAM at 0F0000 - 0F3FFF 0 = Disabled
3	SP19	Shadow Protect 19 (default is 0) 1 = Protect Shadow RAM at 0EC000 - 0EFFFF 0 = Disabled
2	SP18	Shadow Protect 18 (default is 0) 1 = Protect Shadow RAM at 0E8000 - 0EBFFF 0 = Disabled
1	SP17	Shadow Protect 17 (default is 0) 1 = Protect Shadow RAM at 0E4000 - 0E7FFF 0 = Disabled
0	SP16	Shadow Protect 16 (default is 0) 1 = Protect Shadow RAM at 0E0000 - 0E3FFF 0 = Disabled

Memory Wait States Register

Name: MWS
Index: 18 (12H)
Default: 01H

This register selects the number of wait states for ROM and RAM accesses.

7	6	5	4	3	2	1	0
RCXIO	RCROM	RAMWS 1	RAMWS 0	ROMWS 3	ROMWS 2	ROMWS 1	ROMWS 0

Bit(s)	Name	Description
7	RCXIO	X-Bus I/O Recovery Disable (default is 0) Setting this bit will turn off the extra CPU cycle of recovery time for X-Bus I/O devices. This extra recovery time allows for greater command to data three-state delays and greater address to command hold times. With this bit set on, the next CPU cycle is allowed to start immediately after the end of the current X-Bus I/O cycle. 0 = Enable recovery time 1 = Disable extra recovery time
6	RCROM	ROM I/O Recovery Disable (default is 0) Setting this bit will turn off the extra CPU cycle of recovery time for X-Bus Memory in the ROM address space. This extra recovery time allows for greater command to data three-state delays and greater address to command hold times. With this bit set on, the next CPU cycle is allowed to start immediately after the end of the current X-Bus Memory cycle. 0 = Enable recovery time 1 = Disable extra recovery time
5:4	RAMWS1:0	RAM Wait States (default is 00) These 2 bits set the number of wait states generated during memory cycles to system DRAMs. 0 0 = 0 wait states 0 1 = 1 wait state 1 0 = 2 wait states 1 1 = Reserved
3:0	ROMWS3:0	ROM Wait States (default is 0001) These 4 bits define 0-15 wait states, with a straight binary encoding, for memory cycles to the system ROM. Note that ROM cycles are CPU synchronous.

ROM Bank Configuration Register 1

Name: ROBR1
Index: 19 (13H)
Default: 00H

Each ROM Enable Bit in this register enables an individual 16-kb block for ROM access at a specific address.

7	6	5	4	3	2	1	0
RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0

Bit(s)	Name	Description
7	RE7	ROM Enable 7 (default is 0) 1 = Enable ROM access at 0DC000 - 0DFFFF 0 = Disabled
6	RE6	ROM Enable 6 (default is 0) 1 = Enable ROM access at 0D8000 - 0DBFFF 0 = Disabled
5	RE5	ROM Enable 5 (default is 0) 1 = Enable ROM access at 0D4000 - 0D7FFF 0 = Disabled
4	RE4	ROM Enable 4 (default is 0) 1 = Enable ROM access at 0D0000 - 0D3FFF 0 = Disabled
3	RE3	ROM Enable 3 (default is 0) 1 = Enable ROM access at 0CC000 - 0CFFFF 0 = Disabled
2	RE2	ROM Enable 2 (default is 0) 1 = Enable ROM access at 0C8000 - 0CBFFF 0 = Disabled
1	RE1	ROM Enable 1 (default is 0) 1 = Enable ROM access at 0C4000 - 0C7FFF 0 = Disabled
0	RE0	ROM Enable 0 (default is 0) 1 = Enable ROM access at 0C0000 - 0C3FFF 0 = Disabled

ROM Bank Configuration Register 2

Name: ROBR2
Index: 20 (14H)
Default: C0H

Each ROM Enable Bit in this register enables an individual 16 kb block for ROM access at a specific address. Note that the top two blocks are enabled at reset for 32-kb of enabled ROM.

7	6	5	4	3	2	1	0
RE15	RE14	RE13	RE12	RE11	RE10	RE9	RE8

Bit(s)	Name	Description
7	RE15	ROM Enable 15 (default is 1) 1 = Enable ROM access at 0FC000 - 0FFFFF 0 = Disabled
6	RE14	ROM Enable 14 (default is 1) 1 = Enable ROM access at 0F8000 - 0FBFFF 0 = Disabled
5	RE13	ROM Enable 13 (default is 0) 1 = Enable ROM access at 0F4000 - 0F7FFF 0 = Disabled
4	RE12	ROM Enable 12 (default is 0) 1 = Enable ROM access at 0F0000 - 0F3FFF 0 = Disabled
3	RE11	ROM Enable 11 (default is 0) 1 = Enable ROM access at 0EC000 - 0EFFFF 0 = Disabled
2	RE10	ROM Enable 10 (default is 0) 1 = Enable ROM access at 0E8000 - 0EBFFF 0 = Disabled
1	RE9	ROM Enable 9 (default is 0) 1 = Enable ROM access at 0E4000 - 0E7FFF 0 = Disabled
0	RE8	ROM Enable 8 (default is 0) 1 = Enable ROM access at 0E0000 - 0E3FFF 0 = Disabled

RAM Bank Configuration Register

Name: RABR
Index: 21 (15H)
Default: 00H

This register specifies what type/size of DRAM is in each of the four DRAM banks. A null (not installed) code is included.

7	6	5	4	3	2	1	0
D3T1	D3T0	D2T1	D2T0	D1T1	D1T0	D0T1	D0T0

Bit(s)	Name	Description
7:6	D3T1:0	DRAM Bank 3 Type (default is 00) 0 0 = Bank disabled 0 1 = 256-kb DRAMs 1 0 = 1-Mb DRAMs 1 1 = 4-Mb DRAMs
5:4	D2T1:0	DRAM Bank 2 Type (default is 00) 0 0 = Bank disabled 0 1 = 256-kb DRAMs 1 0 = 1-Mb DRAMs 1 1 = 4-Mb DRAMs
3:2	D1T1:0	DRAM Bank 1 Type (default is 00) 0 0 = Bank disabled 0 1 = 256-kb DRAMs 1 0 = 1-Mb DRAMs 1 1 = 4-Mb DRAMs
1:0	D0T1:0	DRAM Bank 0 Type (default is 00) 0 0 = Bank disabled 0 1 = 256-kb DRAMs 1 0 = 1-Mb DRAMs 1 1 = 4-Mb DRAMs

DRAM Configuration Register

Name: DCR
Index: 22 (16H)
Default: 00H

This register contains eight configuration option bits which control the operating mode of the DRAM Controller.

7	6	5	4	3	2	1	0
STR	SRE	PE	RTE	0	PGE	INE	REL

Bit(s)	Name	Description
7	STR	Staggered Refresh Enable (default is 0) 1 = Staggered refresh: activate rows 0 and 2, then 1 and 3 0 = Normal refresh: all $\overline{RAS}$ edges are coincident
6	SRE	Slow Refresh Enable (default is 0) This bit enables a divide by 8 in the refresh request path. 1 = Slow refresh enable 0 = Normal refresh
5	PE	Parity Enable (default is 0) This bit can disable the parity checking logic (16-bit memory banks). 1 = Enable Parity Checking (also requires PortB enable) 0 = Disable Parity Checking (also clears parity error)
4	RTE	$\overline{RAS}$ Timeout Enable (default is 0) This bit enables the $\overline{RAS}$ active timeout feature. 1 = Enable $\overline{RAS}$ timeout feature 0 = Disable $\overline{RAS}$ timeout
3	Reserved	This bit is reserved; do not program to a 1 (default is 0)
2	PGE	Page Mode Enable (default is 0) This bit enables operation of the DRAM controller in page mode. 1 = Enable page mode operation 0 = Disable page mode
1	INE	Interleave Mode Enable (default is 0) This bit enables either two-way or four-way interleaving while in page mode. 1 = Enable interleaved operation 0 = Disable interleaving
0	REL	Relocate 384-kb Memory (default is 0) This bit controls the 384-kb relocate feature for 1-Mb systems. 1 = Enable relocation of 384 kb to above 1 Mb 0 = Disable relocation

ATU I/O Register Location

Name: ARL
Index: 24 (18H)
Default: 00H

This register specifies the placing of the ATU Control Register in the I/O space of the system with an enable bit and an address select field.

7	6	5	4	3	2	1	0
0	0	0	AIE	ARL3	ARL2	ARL1	ARL0

Bit(s)	Name	Description
7:5	Reserved	These bits are reserved; do not program to a 1 (default is 0)
4	AIE	ATU I/O Enable (default is 0) This bit enables decoding of the 02X8h and 02X9h locations for the ATU. If this bit is disabled, the ATU registers do not show up in the system's I/O address space. 1 = Enable I/O decode for ATU registers 0 = Disable decode
3:0	ARL3:0	ATU Register Location (default is 0000, or ATU I/O at 0208h-0209h) These four bits define the second 4-bit nibble of the ATU register's base I/O address.

X-Bus I/O Range Enable Register 1

Name: XBE1
Index: 25 (19H)
Default: 00H

This register can specify the placing of an additional I/O device on the X Bus (as opposed to the default of being on S Bus). The address is specified down to an 8-byte boundary.

7	6	5	4	3	2	1	0
XBEN1	XB1.9	XB1.8	XB1.7	XB1.6	XB1.5	XB1.4	XB1.3

Bit(s)	Name	Description
7	XBEN1	X- bus I/O Range Enable 1 (default is 0) This bit enables the address range specified in bits 6:0 for redirection to the X Bus, as opposed to the S Bus. 1 = Enable X-bus range specified 0 = Disable use of this redirection register
6:0	XB1.9:3	X- bus I/O Address 1 (default is 0000000) These 7 bits define an 8-byte I/O address range for an X-Bus I/O peripheral. These bits are compared to address bits 9 through 3 for the determination of X-bus redirection.

X-Bus I/O Range Enable Register 2

Name: XBE2
Index: 26 (1AH)
Default: 00H

This register can specify the placing of an additional I/O device on the X Bus (as opposed to the default of being on S Bus). The address is specified down to an 8-byte boundary.

7	6	5	4	3	2	1	0
XBEN2	XB2.9	XB2.8	XB2.7	XB2.6	XB2.5	XB2.4	XB2.3

Bit(s)	Name	Description
7	XBEN2	X-bus I/O Range Enable 2 (default is 0) This bit enables the address range specified in bits 6:0 for redirection to the X Bus, as opposed to the S Bus. 1 = Enable X-bus range specified 0 = Disable use of this redirection register
6:0	XB2.9:3	X-bus I/O Address 2 (default is 0000000) These 7 bits define an 8-byte I/O address range for an X-bus I/O peripheral. These bits are compared to address bits 9 through 3 for the determination of X-bus redirection.

X-Bus I/O Range Enable Register 3

Name: XBE3
Index: 27 (1BH)
Default: 00H

This register can specify the placing of an additional I/O device on the X Bus (as opposed to the default of being on S Bus). The address is specified down to an 8-byte boundary.

7	6	5	4	3	2	1	0
XBEN3	XB3.9	XB3.8	XB3.7	XB3.6	XB3.5	XB3.4	XB3.3

Bit(s)	Name	Description
7	XBEN3	X-bus I/O Range Enable 3 (default is 0) This bit enables the address range specified in bits 6:0 for redirection to the X Bus, as opposed to the S Bus. 1 = Enable X-bus range specified 0 = Disable use of this redirection register
6:0	XB3.9:3	X-bus I/O Address 3 (default is 0000000) These 7 bits define an 8-byte I/O address range for an X-bus I/O peripheral. These bits are compared to address bits 9 through 3 for the determination of X-bus redirection.

X-Bus I/O Range Enable Register 4

Name: XBE4
Index: 28 (1CH)
Default: 00H

This register can specify the placing of an additional I/O device on the X Bus (as opposed to the default of being on S Bus). The address is specified down to an 8-byte boundary.

7	6	5	4	3	2	1	0
XBEN4	XB4.9	XB4.8	XB4.7	XB4.6	XB4.5	XB4.4	XB4.3

Bit(s)	Name	Description
7	XBEN4	X-bus I/O Range Enable 4 (default is 0) This bit enables the address range specified in bits 6:0 for redirection to the X Bus, as opposed to the S Bus. 1 = Enable X-bus range specified 0 = Disable use of this redirection register
6:0	XB4.9:3	X-bus I/O Address 4 (default is 0000000) These 7 bits define an 8-byte I/O address range for an X-bus I/O peripheral. These bits are compared to address bits 9 through 3 for the determination of X-bus redirection.

X-Bus I/O Range Enable Register 5

Name: XBE5
Index: 29 (1DH)
Default: 00H

This register can specify the placing of an additional I/O device on the X Bus (as opposed to the default of being on S Bus). The address is specified down to an 8-byte boundary.

7	6	5	4	3	2	1	0
XBEN5	XB5.9	XB5.8	XB5.7	XB5.6	XB5.5	XB5.4	XB5.3

Bit(s)	Name	Description
7	XBEN5	X-bus I/O Range Enable 5 (default is 0) This bit enables the address range specified in bits 6:0 for redirection to the X Bus, as opposed to the S Bus. 1 = Enable X-bus range specified 0 = Disable use of this redirection register
6:0	XB5.9:3	X-bus I/O Address 5 (default is 0000000) These 7 bits define an 8-byte I/O address range for an X-bus I/O peripheral. These bits are compared to address bits 9 through 3 for the determination of X-bus redirection.

X-Bus I/O Range Enable Register 6

Name: XBE6
Index: 30 (1EH)
Default: 00H

This register can specify the placing of an additional I/O device on the X Bus (as opposed to the default of being on S Bus). The address is specified down to an 8-byte boundary.

7	6	5	4	3	2	1	0
XBEN6	XB6.9	XB6.8	XB6.7	XB6.6	XB6.5	XB6.4	XB6.3

Bit(s)	Name	Description
7	XBEN6	X-bus I/O Range Enable 6 (default is 0) This bit enables the address range specified in bits 6:0 for redirection to the X Bus, as opposed to the S Bus. 1 = Enable X-bus range specified 0 = Disable use of this redirection register
6:0	XB6.9:3	X-bus I/O Address 6 (default is 0000000) These 7 bits define an 8-byte I/O address range for an X-bus I/O peripheral. These bits are compared to address bits 9 through 3 for the determination of X-bus redirection.

X-Bus I/O Range Enable Register 7

Name: XBE7
Index: 31 (1FH)
Default: 00H

This register can specify the placing of an additional I/O device on the X Bus (as opposed to the default of being on S Bus). The address is specified down to an 8-byte boundary.

7	6	5	4	3	2	1	0
XBEN7	XB7.9	XB7.8	XB7.7	XB7.6	XB7.5	XB7.4	XB7.3

Bit(s)	Name	Description
7	XBEN7	X-bus I/O Range Enable 7 (default is 0) This bit enables the address range specified in bits 6:0 for redirection to the X Bus, as opposed to the S Bus. 1 = Enable X-Bus range specified 0 = Disable use of this redirection register
6:0	XB7.9:3	X-bus I/O Address 7 (default is 0000000) These 7 bits define an 8-byte I/O address range for an X-bus I/O peripheral. These bits are compared to address bits 9 through 3 for the determination of X-bus redirection.

X-Bus I/O Range Enable Register 8

Name: XBE8
Index: 32 (20H)
Default: 00H

This register can specify the placing of an additional I/O device on the X Bus (as opposed to the default of being on S Bus). The address is specified down to an 8-byte boundary.

7	6	5	4	3	2	1	0
XBEN8	XB8.9	XB8.8	XB8.7	XB8.6	XB8.5	XB8.4	XB8.3

Bit(s)	Name	Description
7	XBEN8	X-bus I/O Range Enable 8 (default is 0) This bit enables the address range specified in bits 6:0 for redirection to the X Bus, as opposed to the S Bus. 1 = Enable X-Bus range specified 0 = Disable use of this redirection register
6:0	XB8.9:3	X-bus I/O Address 8 (default is 0000000) These 7 bits define an 8-byte I/O address range for an X-bus I/O peripheral. These bits are compared to address bits 9 through 3 for the determination of X-bus redirection.

X-Bus DMA Channel Enable Register

Name: XBD
Index: 33 (21H)
Default: 00H

This register specifies which 8-bit DMA channels, if any, are used by peripherals on the X Bus.

7	6	5	4	3	2	1	0
0	0	0	0	XBD3	XBD2	XBD1	XBD0

Bit(s)	Name	Description
7:4	Reserved	These bits are reserved; do not program to a 1 (default is 0)
3	XBD3	DMA Channel 3 on X Bus (default is 0) 1 = Enable X-bus DMA routing for this channel 0 = Disable
2	XBD2	DMA Channel 2 on X Bus (default is 0) 1 = Enable X-bus DMA routing for this channel 0 = Disable
1	XBD1	DMA Channel 1 on X Bus (default is 0) 1 = Enable X-bus DMA routing for this channel 0 = Disable
0	XBD0	DMA Channel 0 on X Bus (default is 0) 1 = Enable X-bus DMA routing for this channel 0 = Disable

X-Bus I/O Size Register

Name: XBSZ
Index: 34 (22H)
Default: 00H

This register defines the size (8 or 16 bit) of each X-bus peripheral defined in the X-bus Enable registers (XBE1-XBE8). However, any 8-bit defined peripheral may drive IOCS16# low to indicate a "hardware-signaled" 16-bit transfer.

7	6	5	4	3	2	1	0
XSZ8	XSZ7	XSZ6	XSZ5	XSZ4	XSZ3	XSZ2	XSZ1

Bit(s)	Name	Description
7	XSZ8	X-bus Range Size 8 (default is 0) 1 = Peripheral in X-bus peripheral address range 8 is 16 bit 0 = Peripheral is 8 bit or hardware signaled 16 bit
6	XSZ7	X-bus Range Size 7 (default is 0) 1 = Peripheral in X-bus peripheral address range 7 is 16 bit 0 = Peripheral is 8 bit or hardware signaled 16 bit
5	XSZ6	X-bus Range Size 6 (default is 0) 1 = Peripheral in X-bus peripheral address range 6 is 16 bit 0 = Peripheral is 8 bit or hardware signaled 16 bit
4	XSZ5	X-bus Range Size 5 (default is 0) 1 = Peripheral in X-bus peripheral address range 5 is 16 bit 0 = Peripheral is 8 bit or hardware signaled 16 bit
3	XSZ4	X-bus Range Size 4 (default is 0) 1 = Peripheral in X-bus peripheral address range 4 is 16 bit 0 = Peripheral is 8 bit or hardware signaled 16 bit
2	XSZ3	X-bus Range Size 3 (default is 0) 1 = Peripheral in X-Bus peripheral address range 3 is 16 bit 0 = Peripheral is 8 bit or hardware signaled 16 bit
1	XSZ2	X-bus Range Size 2 (default is 0) 1 = Peripheral in X-bus peripheral address range 2 is 16 bit 0 = Peripheral is 8 bit or hardware signaled 16 bit
0	XSZ1	X-bus Range Size 1 (default is 0) 1 = Peripheral in X-bus peripheral address range 1 is 16 bit 0 = Peripheral is 8 bit or hardware signaled 16 bit

Power Save Function Register

Name: PSFR
Index: 35 (23H)
Default: 00H

This register contains the general and CPU specific "stop clock" bits which enable reduction of operating power by stopping internal clocks. Either a NMI, or a normal interrupt will take the device out of system standby or CPU stop clock mode.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	SSBY	CSTC

Bit(s)	Name	Description
7:2	Reserved	These bits are reserved; do not program to a 1 (default is 0)
1	SSBY	System Standby Mode Select (default is 0) A 0 to 1 transition in this bit will place the device in system standby, which shuts down the CPU clock along with other clocks in the device which are not associated with maintaining DRAM refresh capability. 1 = Go to standby mode 0 = Normal operation
0	CSTC	CPU Stop Clock Select (default is 0) A 0 to 1 transition in this bit will stop the CPU clock, but will allow DRAM refreshes and DMA cycles to continue. 1 = Stop CPU clock 0 = Normal operation

Map Space Control Register

Name: MSC
Index: 37 (25H)
Default: 00H

This register specifies the amount of local memory physical address space to reserve for use with the address manager's mapping capabilities (EMS 4.0).

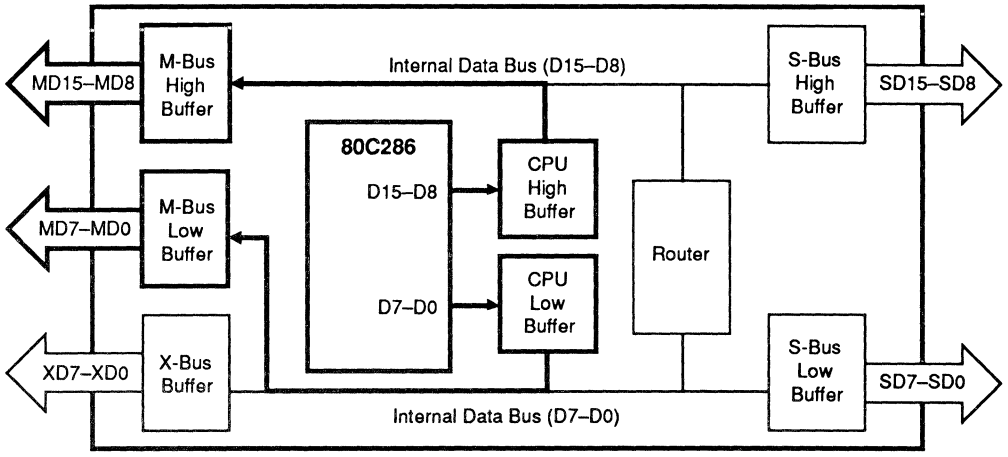
7	6	5	4	3	2	1	0
RSRVD	SEMS	RSRVD	MSS4	MSS3	MSS2	MSS1	MSS0

Bit(s)	Name	Description
7	Reserved	This bit is reserved; do not program to a 1 (default is 0)
6	SEMS	Special EMS Map Space (default is 0) 1 = Enable special 384K map space 0 = Disable special 384K map space
5	Reserved	This bit is reserved; do not program to a 1 (default is 0)
4:0	MSS4:0	Map Space Select (default is 00000, no memory reserved) These bits specify a multiple of 512-kb of local memory to reserve for use with the address manager. The bits are a straight binary encoding of the 512-kb multiple: 00000 indicates no memory reserved, 11111 indicates 32 x 512K, which is 16 Mb, the physical limit of the DRAM controller.

APPENDIX C

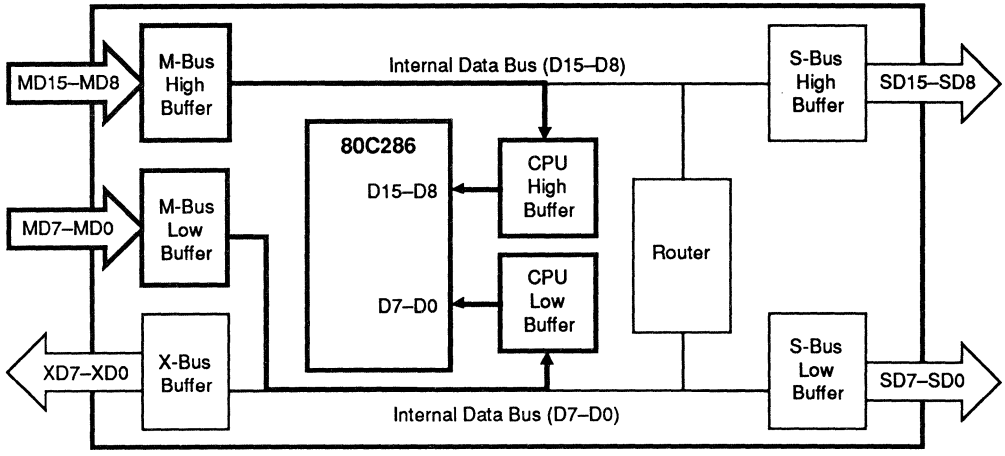
TRANSFER CYCLE REFERENCE

Figure C–1



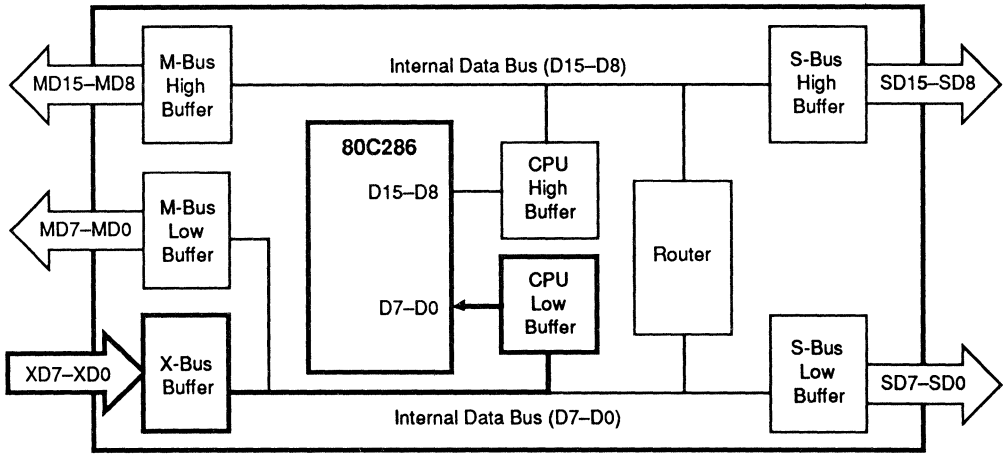
Master: 16-bit CPU
Target: 16-bit Local Memory, AMD 80C287 Math Coprocessor
Cycle Type: Write

Figure C–2



Master: 16-bit CPU
Target: 16-bit Local Memory, AMD 80C287 Math Coprocessor
Cycle Type: Read

Figure C–3



Master: 8-bit CPU
Target: 8-bit ROM Space, X-bus I/O, AT Keyboard Controller
Cycle Type: Even Address Read

Figure C–4

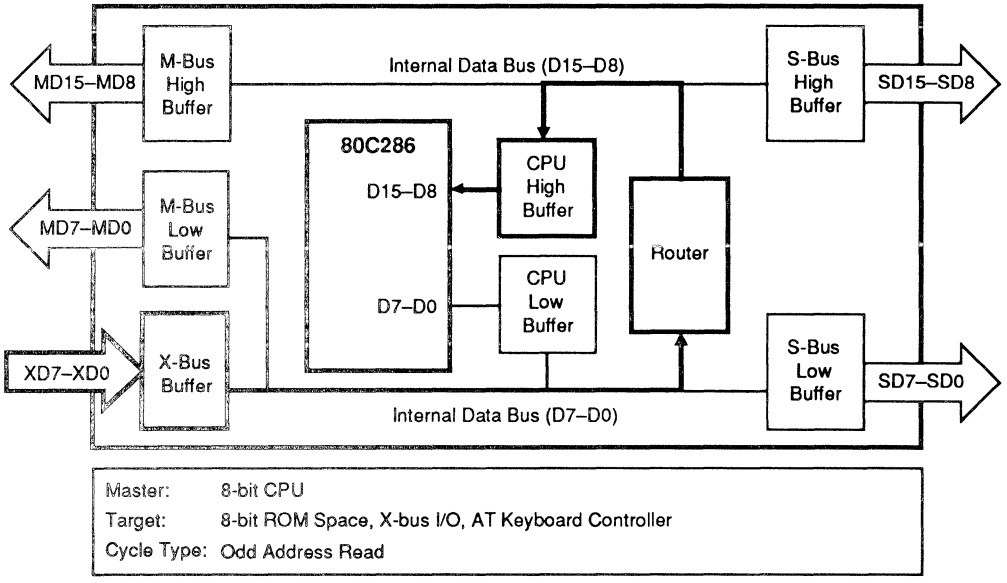


Figure C–5

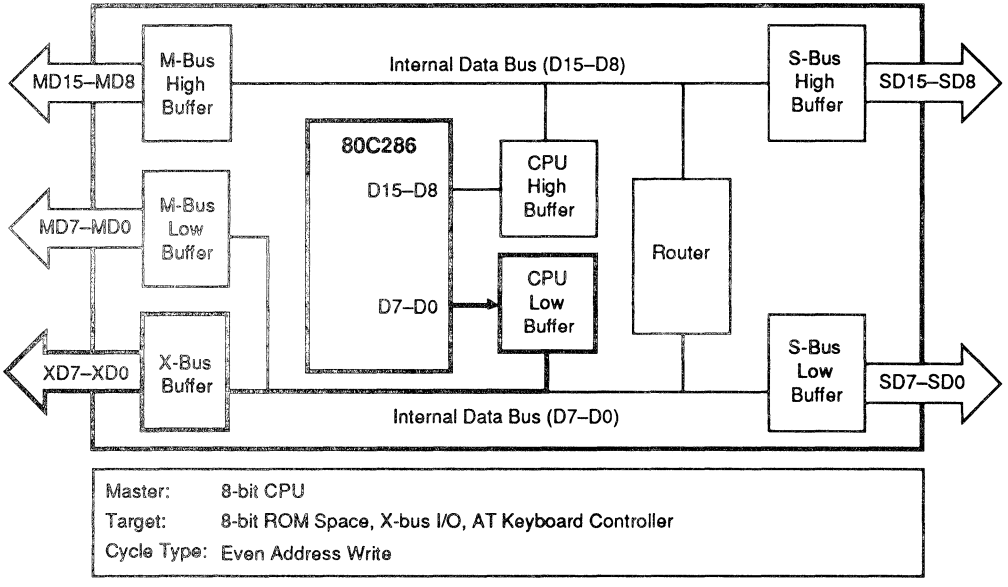
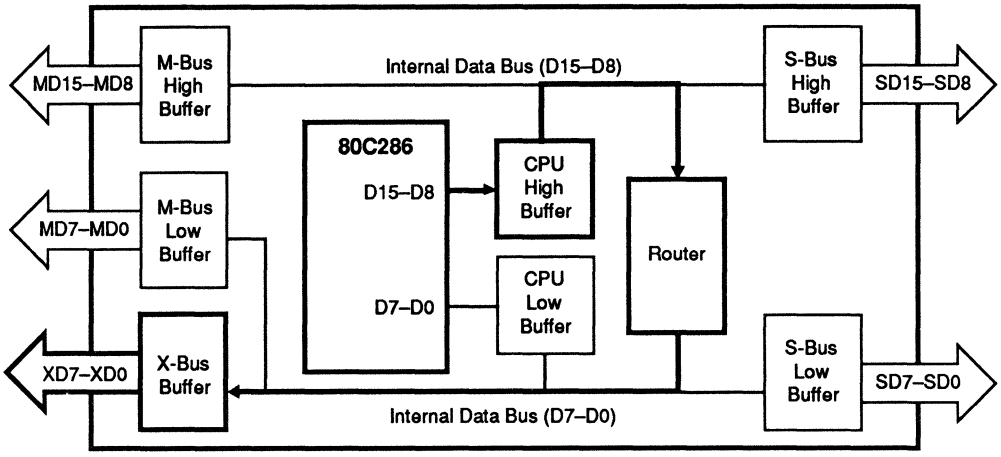
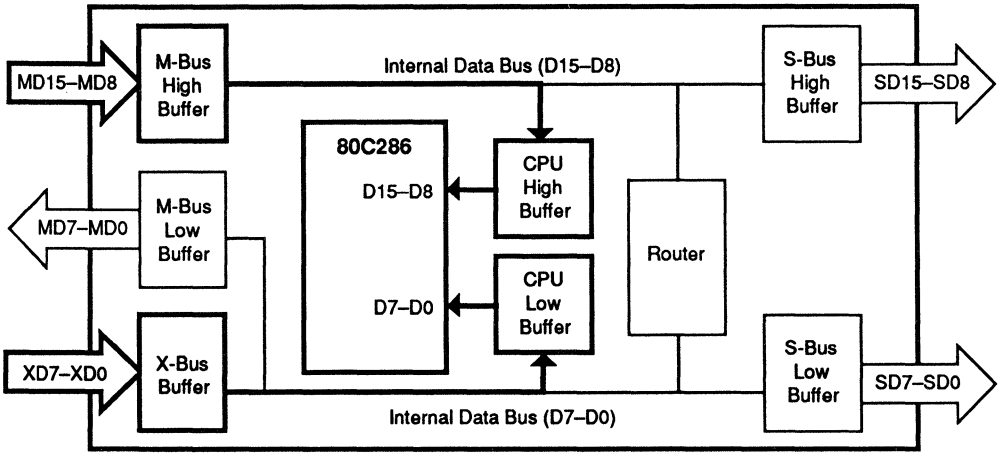


Figure C–6



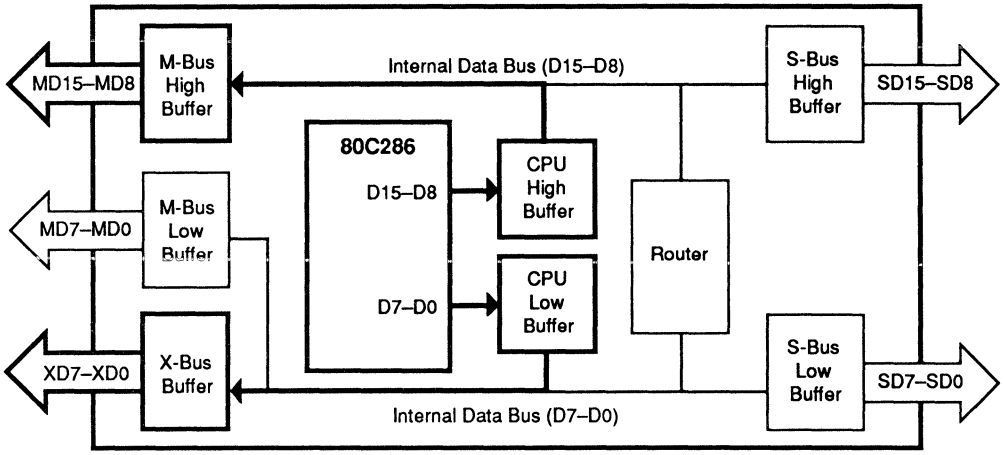
Master:	8-bit CPU
Target:	8-bit ROM Space, X-bus I/O, AT Keyboard Controller
Cycle Type:	Odd Address Write

Figure C–7



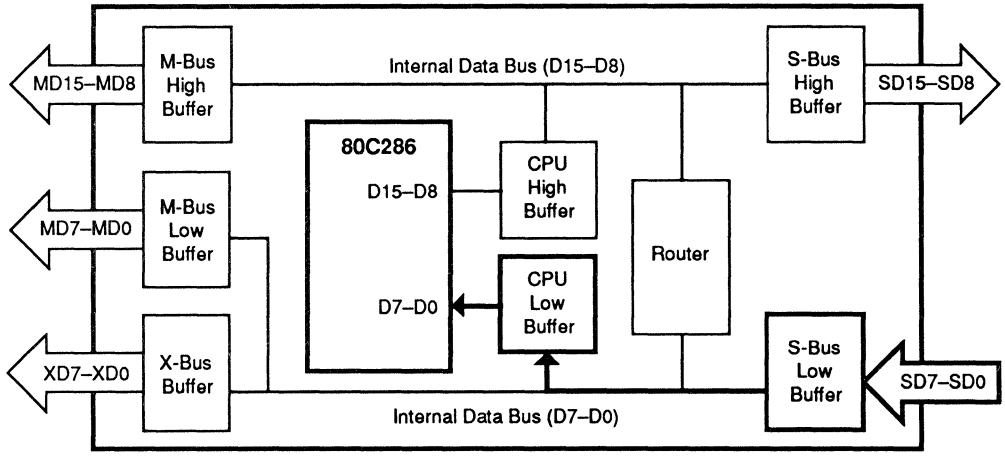
Master:	16-bit CPU
Target:	16-bit ROM Space, X-bus I/O
Cycle Type:	Read

Figure C–8



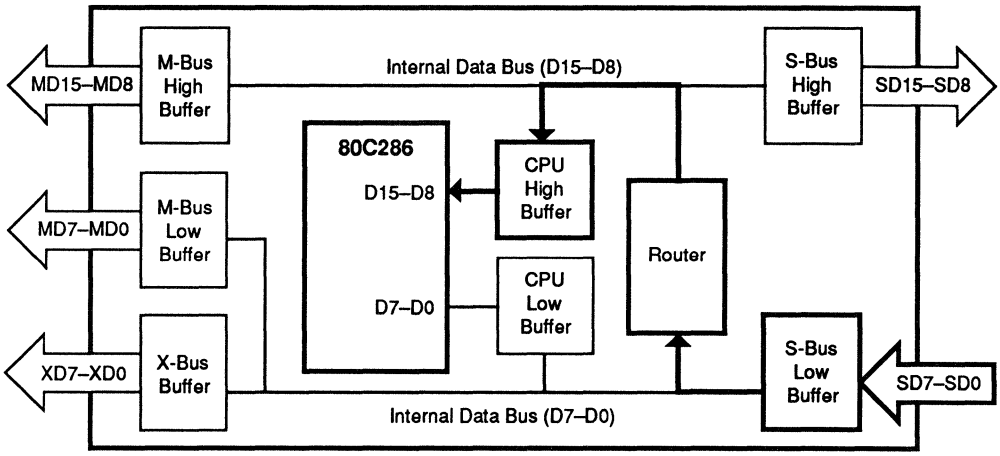
Master:	16-bit CPU
Target:	16-bit ROM Space, X-bus I/O
Cycle Type:	Write

Figure C–9



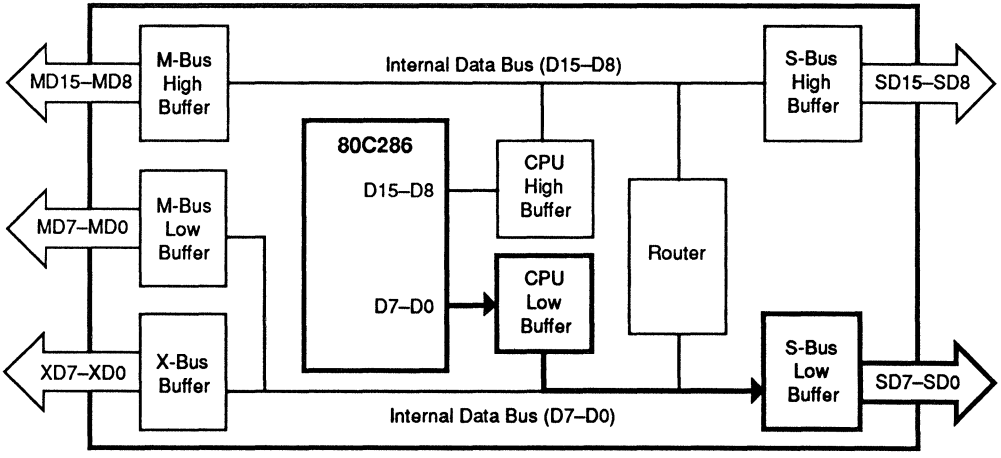
Master:	8-bit CPU
Target:	8-bit S-bus Memory or I/O ($\overline{\text{IOCS16}}$ and $\overline{\text{MEMCS16}}$ High)
Cycle Type:	Even Address Read

Figure C–10



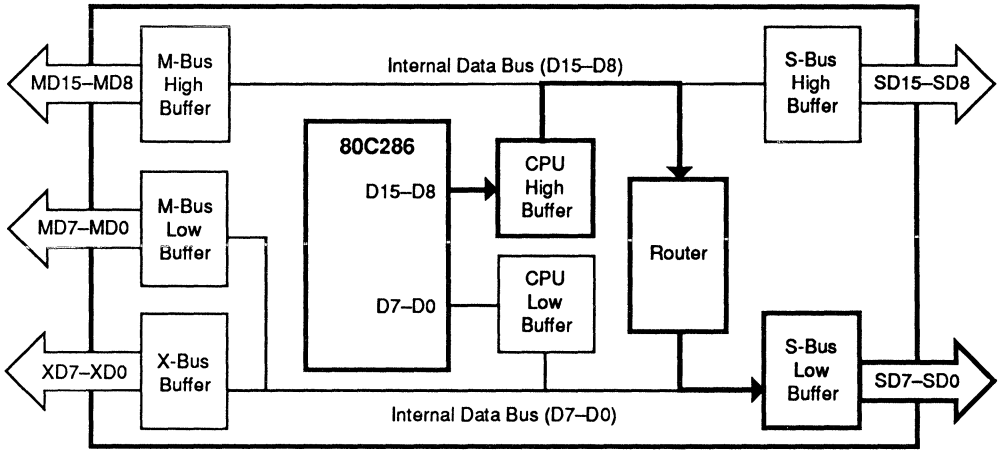
Master: 8-bit CPU
Target: 8-bit S-bus Memory or I/O ($\overline{\text{IOCS16}}$ and $\overline{\text{MEMCS16}}$ High)
Cycle Type: Odd Address Read

Figure C–11



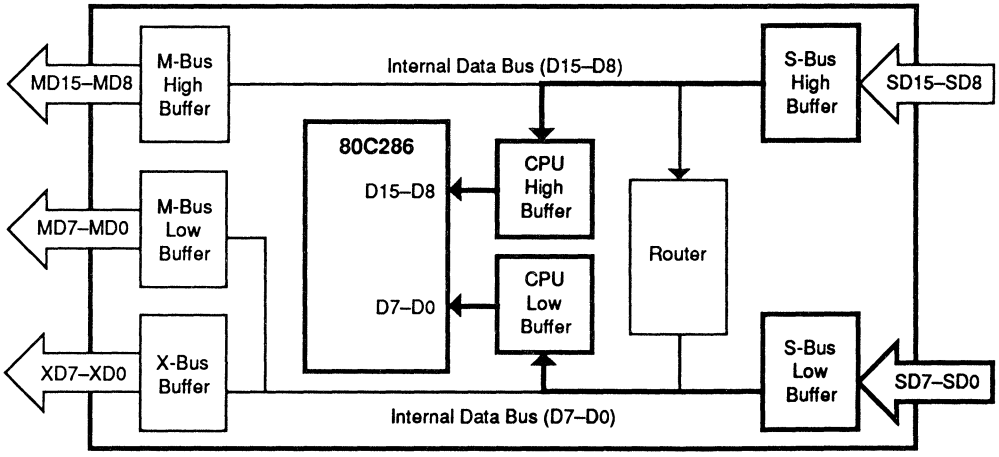
Master: 8-bit CPU
Target: 8-bit S-bus Memory or I/O ($\overline{\text{IOCS16}}$ and $\overline{\text{MEMCS16}}$ High)
Cycle Type: Even Address Write

Figure C–12



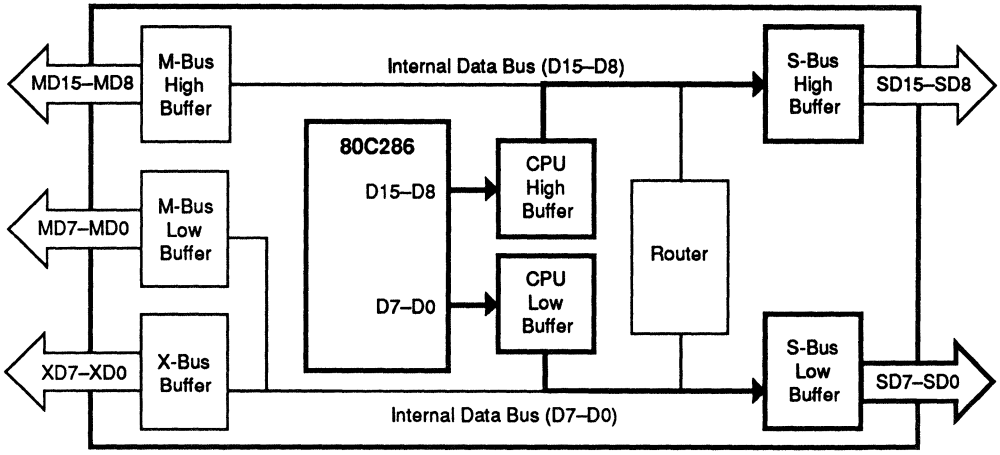
Master: 8-bit CPU
Target: 8-bit S-bus Memory or I/O ($\overline{IOCS16}$ and $\overline{MEMCS16}$ High)
Cycle Type: Odd Address Write

Figure C–13



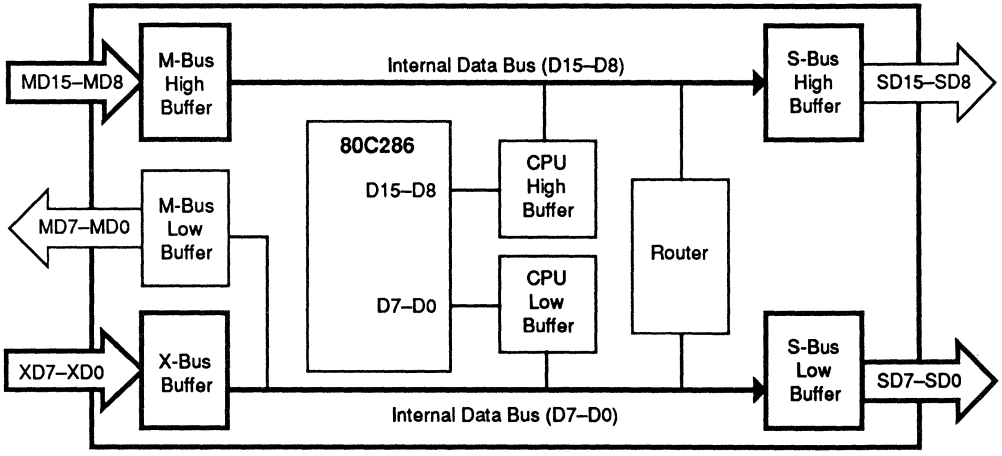
Master: 16-bit CPU
Target: 16-bit S-bus Memory or I/O ($\overline{IOCS16}$ and $\overline{MEMCS16}$ High)
Cycle Type: Read

Figure C–14



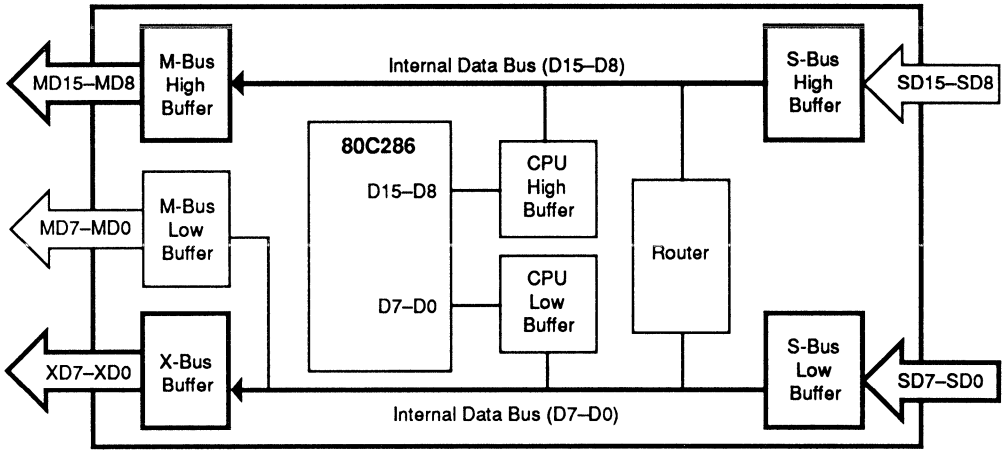
Master:	16-bit CPU
Target:	16-bit S-bus Memory or I/O ($\overline{IOCS16}$ and $\overline{MEMCS16}$ High)
Cycle Type:	Write

Figure C–15



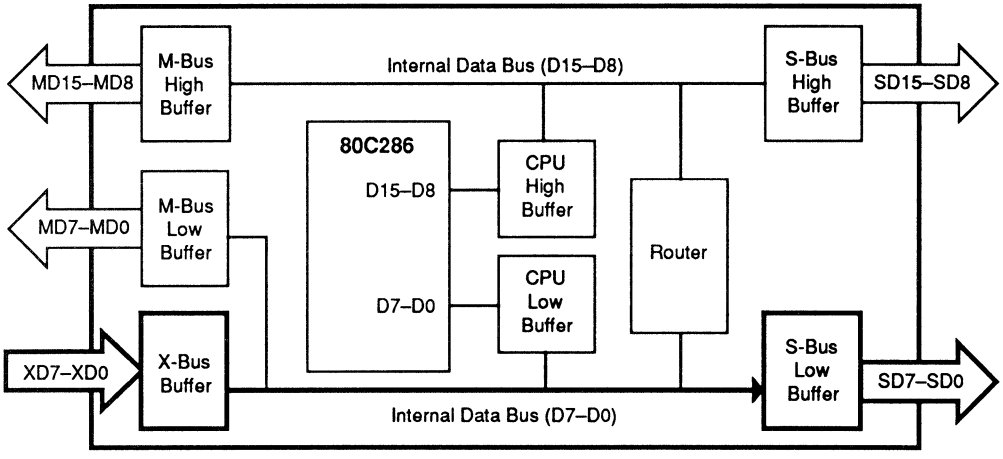
Master:	16-bit Bus Master
Target:	16-bit X-bus I/O
Cycle Type:	Read

Figure C–16



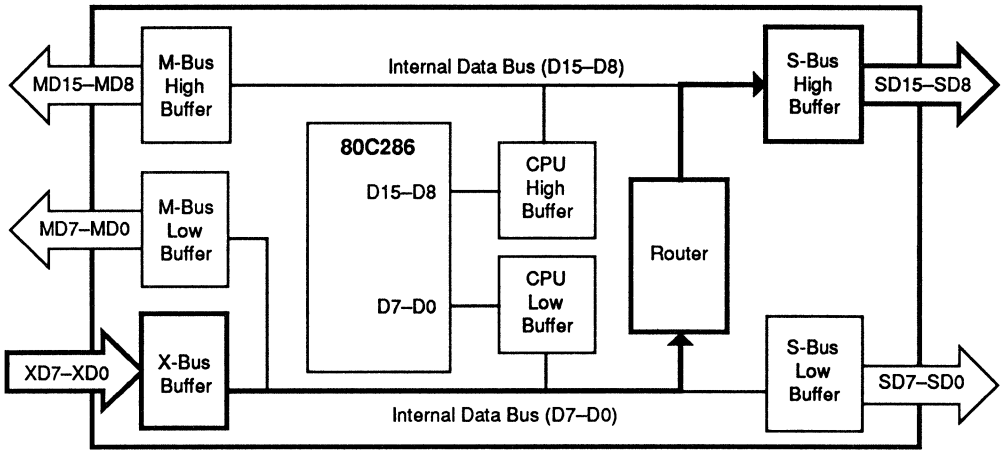
Master:	16-bit Bus Master
Target:	16-bit X-bus I/O
Cycle Type:	Write

Figure C–17



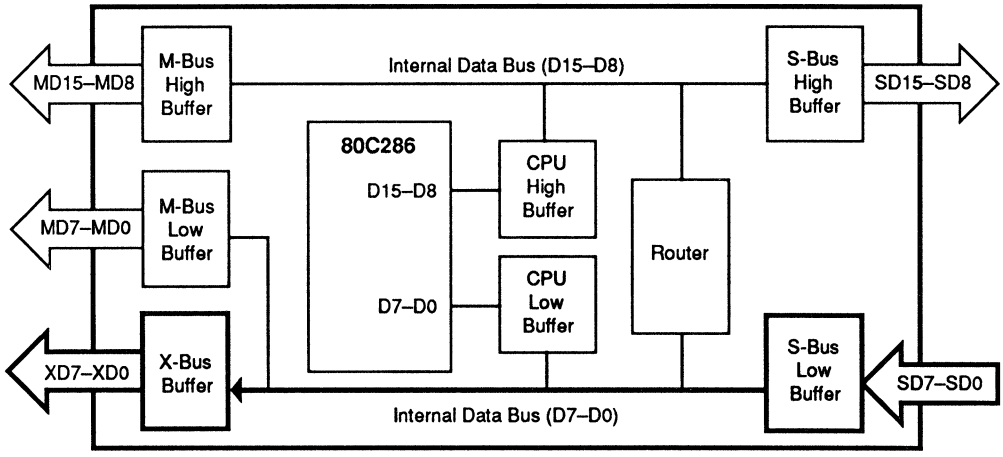
Master:	8-bit Bus Master
Target:	8-bit X-bus I/O, AT Keyboard Controller
Cycle Type:	Even Address Read

Figure C–18



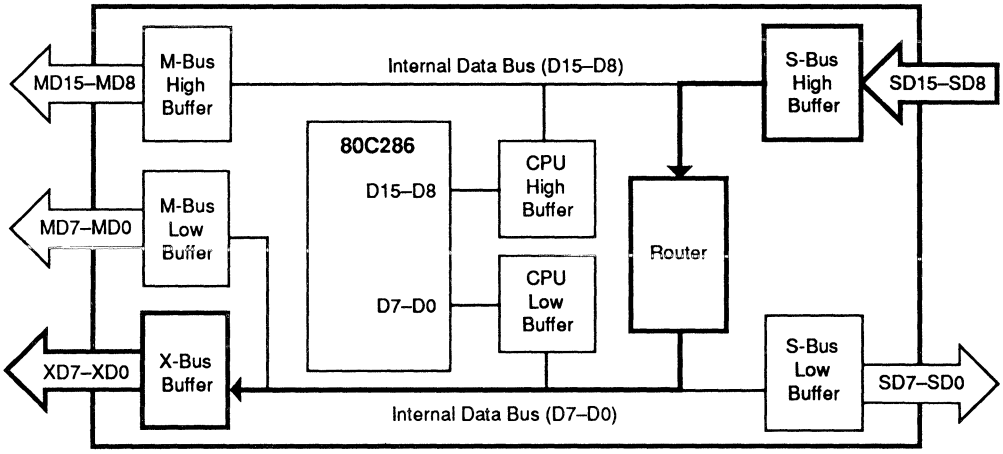
Master: 8-bit Bus Master
Target: 8-bit X-bus I/O, AT Keyboard Controller
Cycle Type: Odd Address Read

Figure C–19



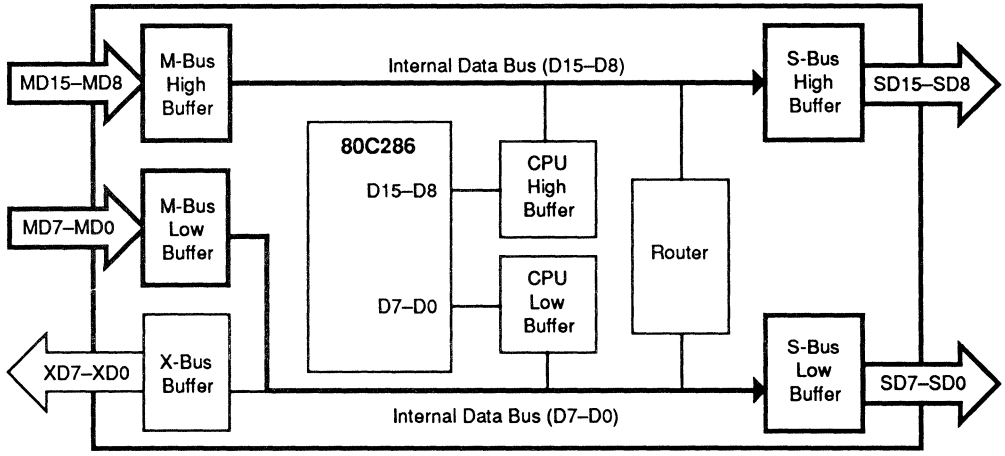
Master: 8-bit Bus Master
Target: 8-bit X-bus I/O, AT Keyboard Controller
Cycle Type: Even Address Write

Figure C–20



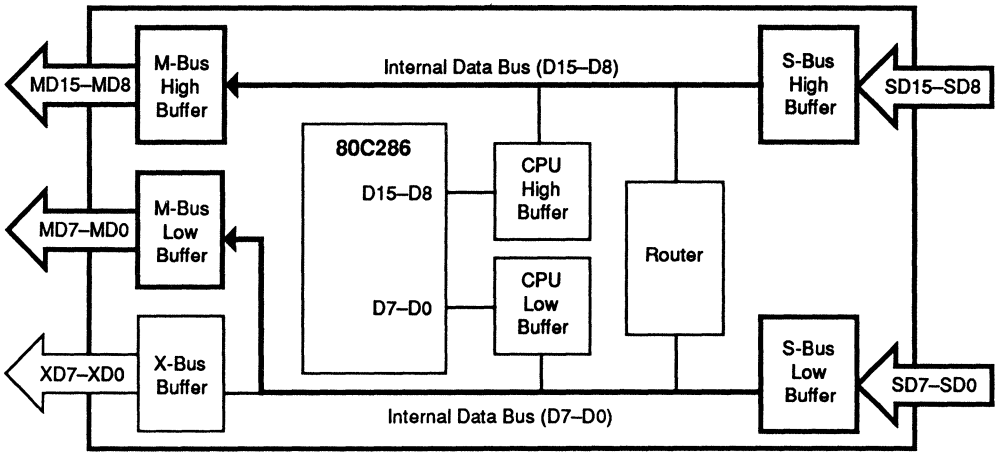
Master: 8-bit Bus Master
Target: 8-bit X-bus I/O, AT Keyboard Controller
Cycle Type: Odd Address Write

Figure C–21



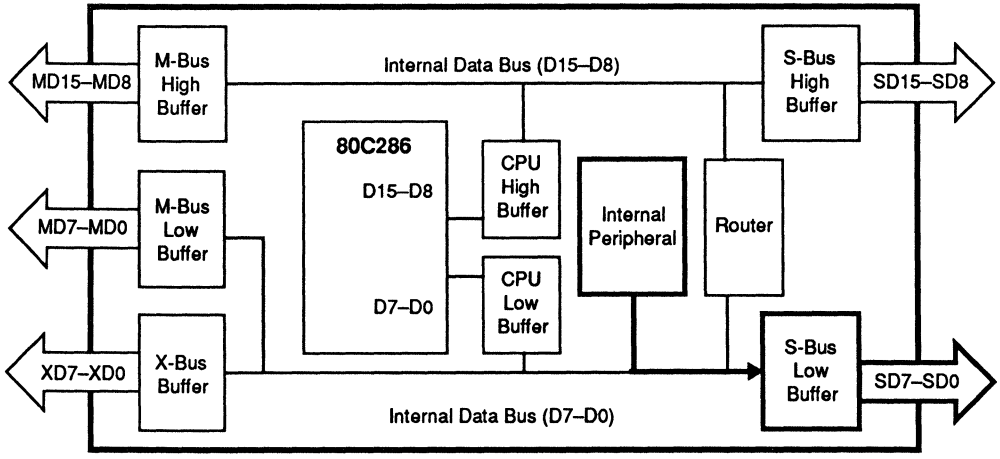
Master: 16-bit Bus Master
Target: 16-bit Local Memory
Cycle Type: Read

Figure C–22



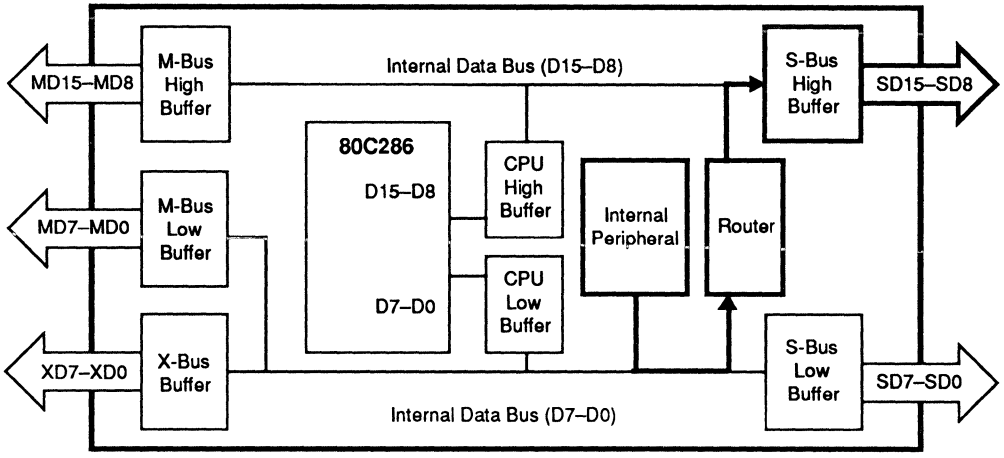
Master: 16-bit Bus Master
Target: 16-bit Local Memory
Cycle Type: Write

Figure C–23



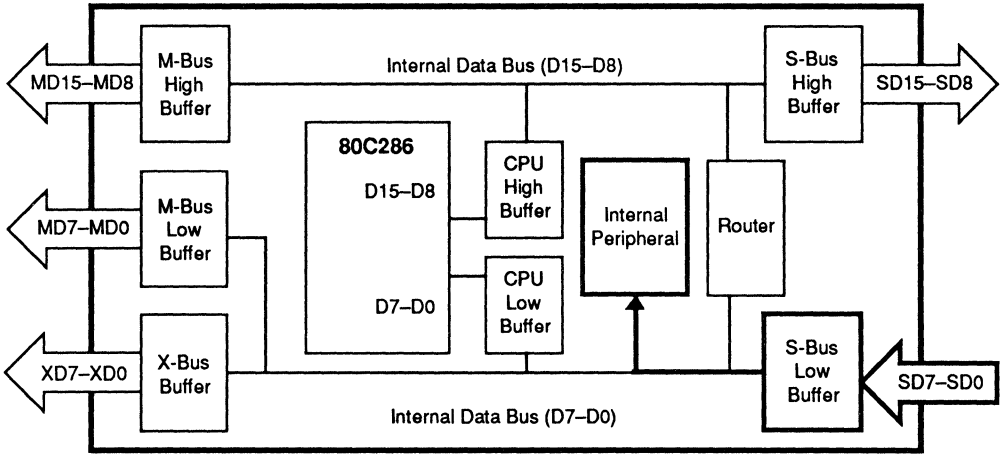
Master: 8-bit Bus Master
Target: 8-bit Internal Peripherals
Cycle Type: Even Address Read

Figure C–24



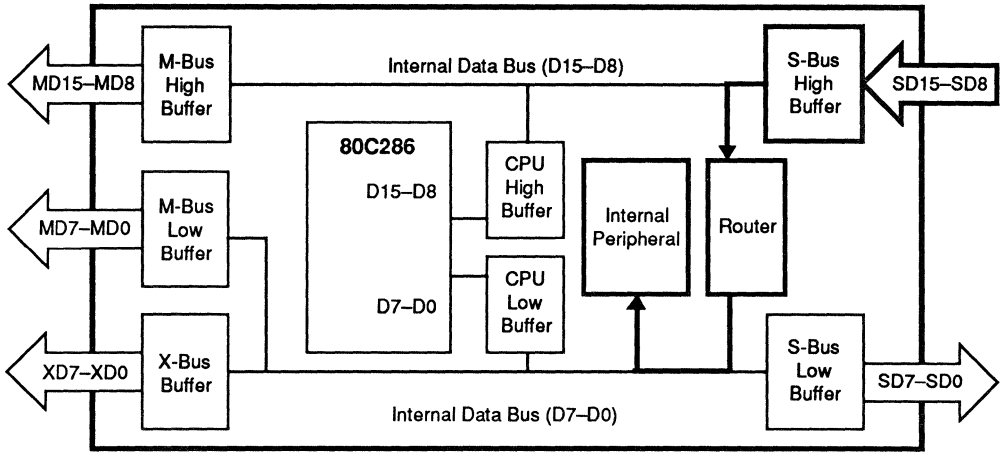
Master: 8-bit Bus Master
Target: 8-bit Internal Peripherals
Cycle Type: Odd Address Read

Figure C–25



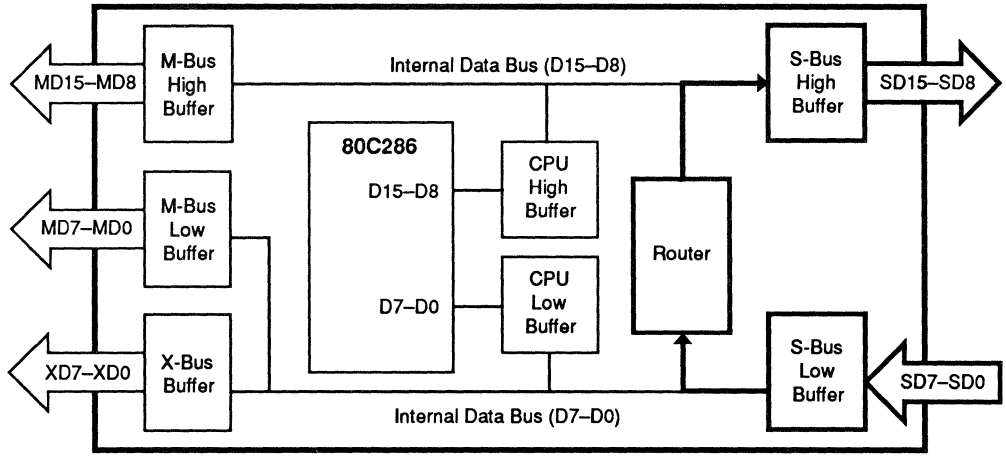
Master: 8-bit Bus Master
Target: 8-bit Internal Peripherals
Cycle Type: Even Address Write

Figure C–26



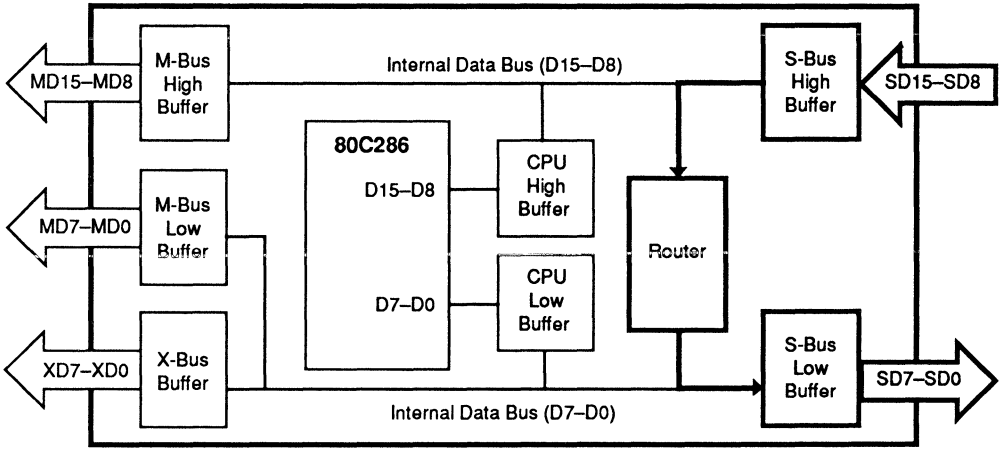
Master:	8-bit Bus Master
Target:	8-bit Internal Peripherals
Cycle Type:	Odd Address Write

Figure C–27



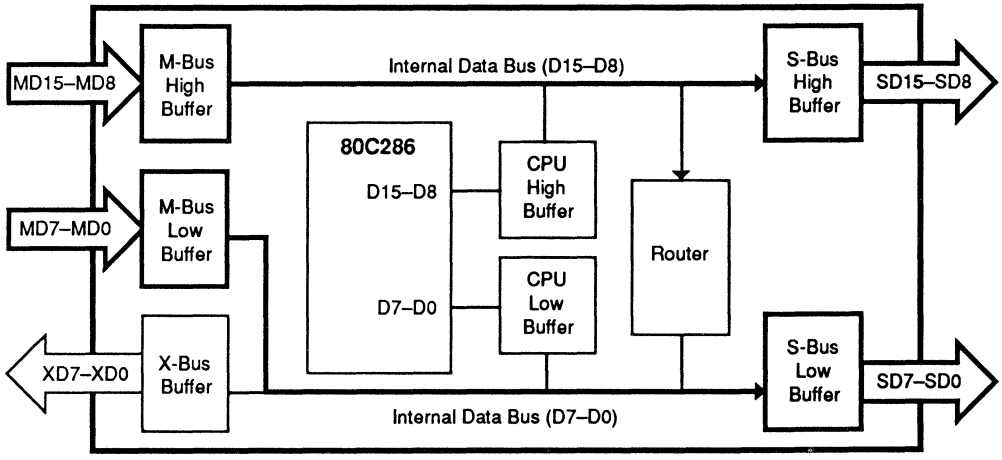
Master:	8-bit Bus Master
Target:	8-bit S-bus Peripheral ($\overline{\text{IOCS16}}$ and $\overline{\text{MEMCS16}}$ Inactive)
Cycle Type:	Odd Address Read

Figure C-28



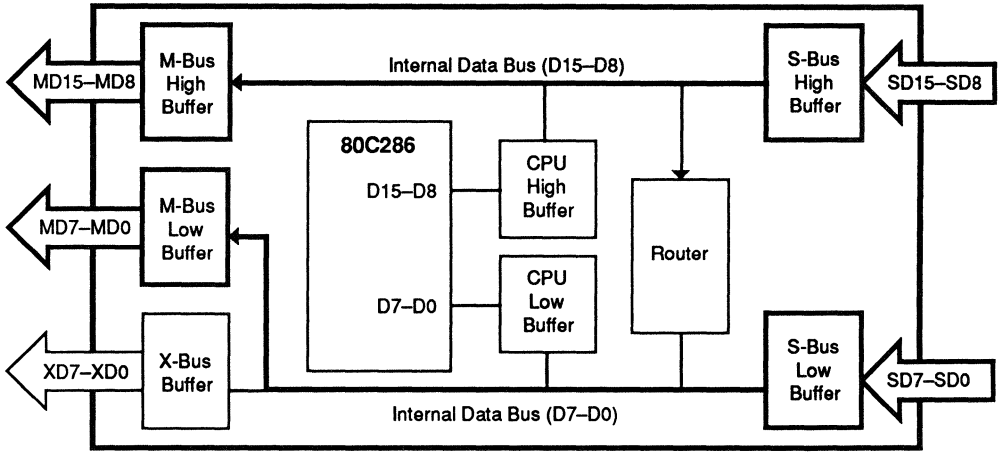
Master:	8-bit Bus Master
Target:	8-bit S-bus Peripheral ($\overline{\text{IOCS16}}$ and $\overline{\text{MEMCS16}}$ Inactive)
Cycle Type:	Odd Address Write

Figure C-29



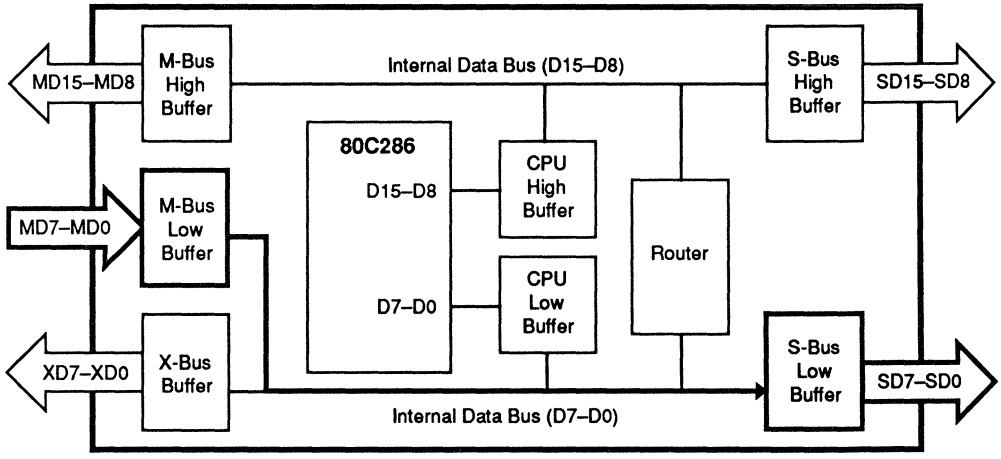
Master:	16-bit DMA
Target:	16-bit S-Bus I/O
Cycle Type:	Read Local Memory, Write S Bus

Figure C–30



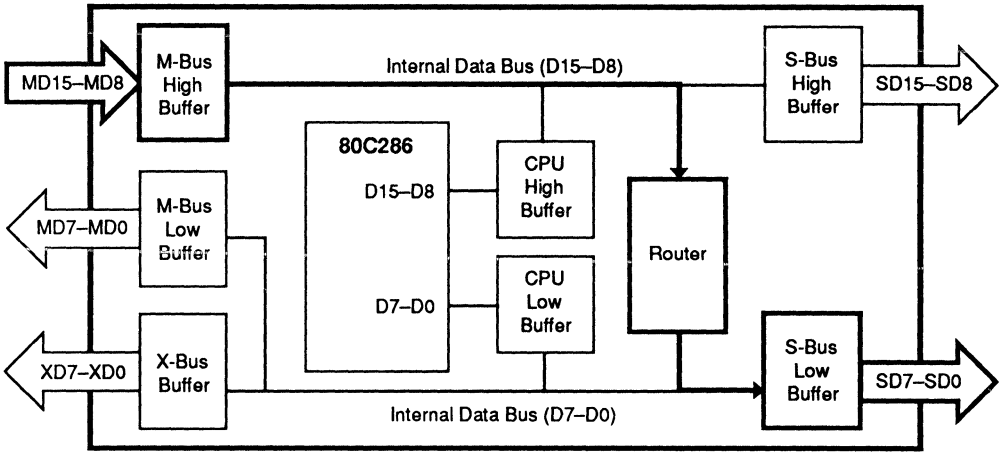
Master:	16-bit DMA
Target:	16-bit S-Bus I/O
Cycle Type:	Write Local Memory, Read S Bus

Figure C–31



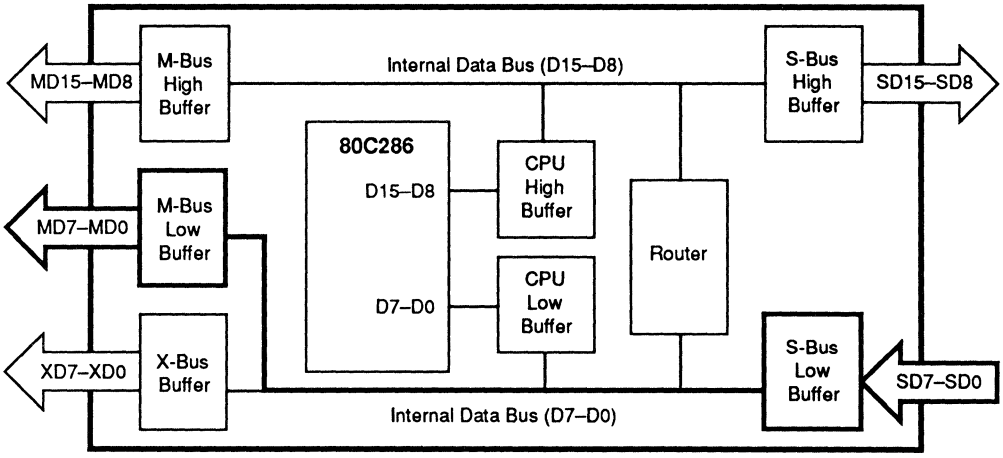
Master:	8-bit DMA1
Target:	8-bit S-bus I/O
Cycle Type:	Transfer from Even Address DRAM to 8-bit S-bus I/O

Figure C–32



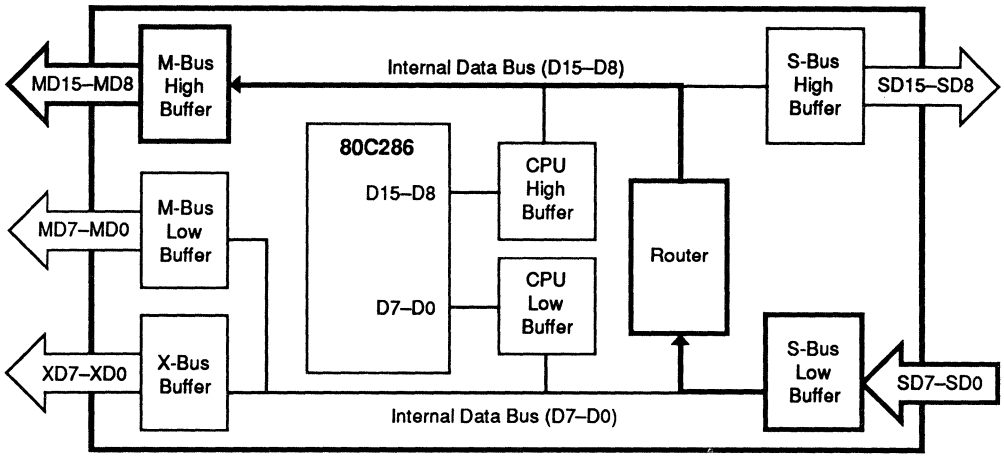
Master: 8-bit DMA1
Target: 8-bit S-bus I/O
Cycle Type: Transfer from Odd Address DRAM to 8-bit S-bus I/O

Figure C–33



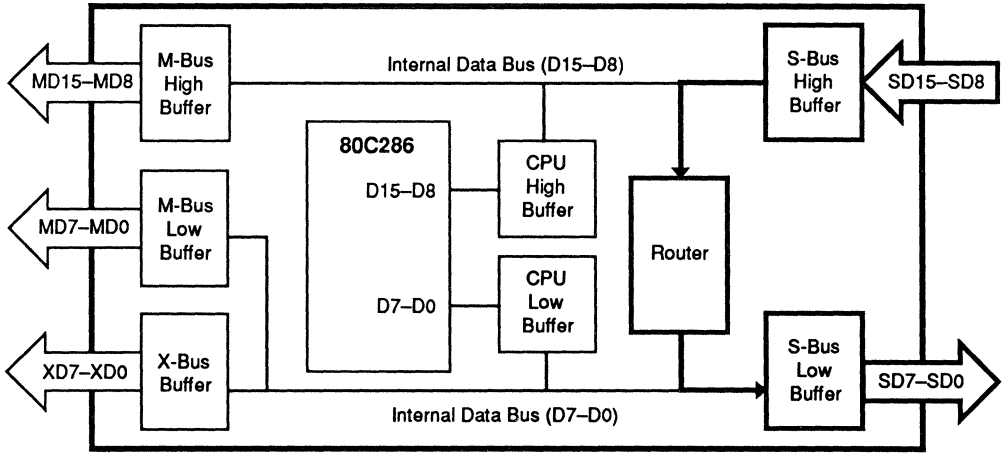
Master: 8-bit DMA1
Target: Even Address DRAM
Cycle Type: Write to Even Address DRAM to 8-bit S-bus I/O

Figure C-34



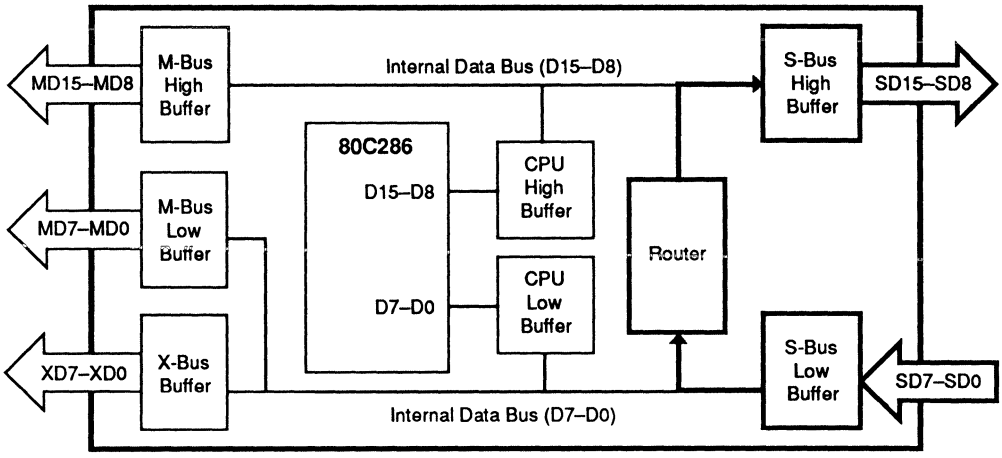
Master:	8-bit DMA1
Target:	Odd Address DRAM
Cycle Type:	Write to Odd Address DRAM from 8-bit S-bus I/O

Figure C-35



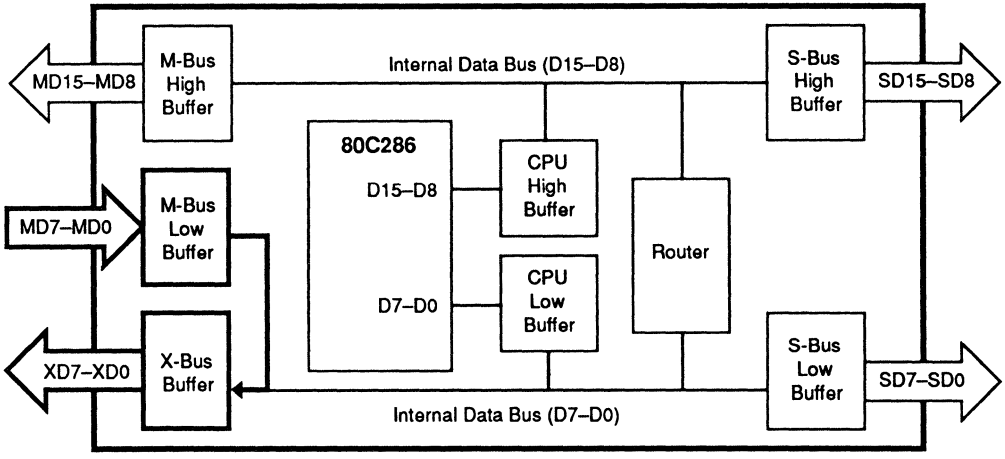
Master:	8-bit DMA1
Target:	8-bit S-bus I/O
Cycle Type:	Transfer from Odd Address 16-bit S-bus Memory to 8-bit S-bus I/O

Figure C–36



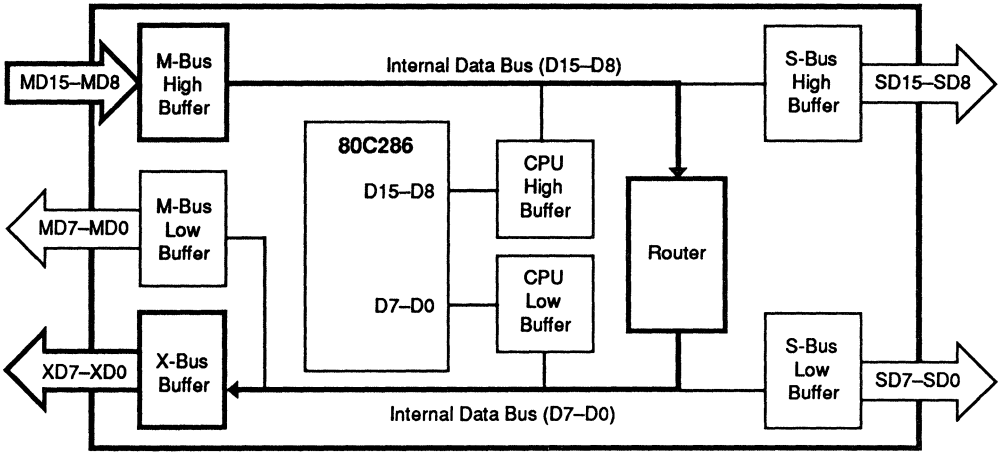
Master: 8-bit DMA1
Target: 8-bit S-bus I/O
Cycle Type: Transfer to Odd Address 16-bit S-bus Memory from 8-bit S-bus I/O

Figure C–37



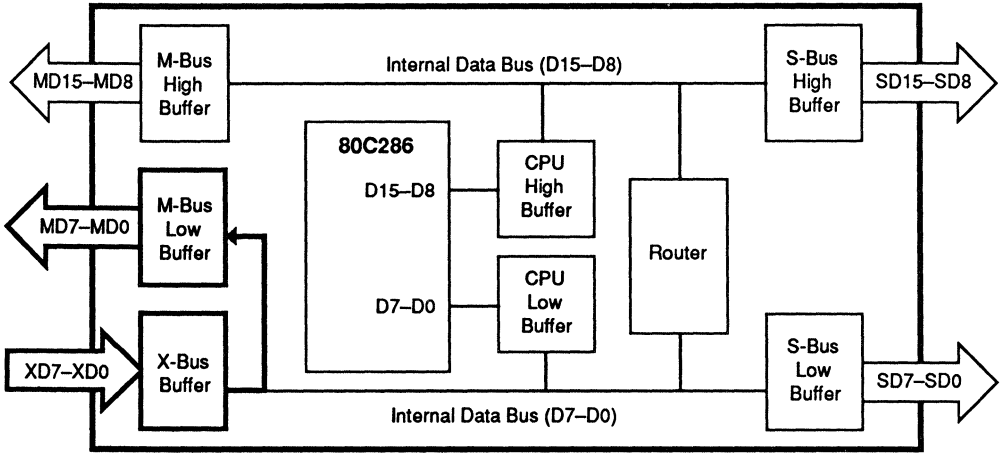
Master: 8-bit DMA1
Target: 8-bit X-bus I/O
Cycle Type: Transfer from Even Address Local Memory to X-bus I/O

Figure C–38



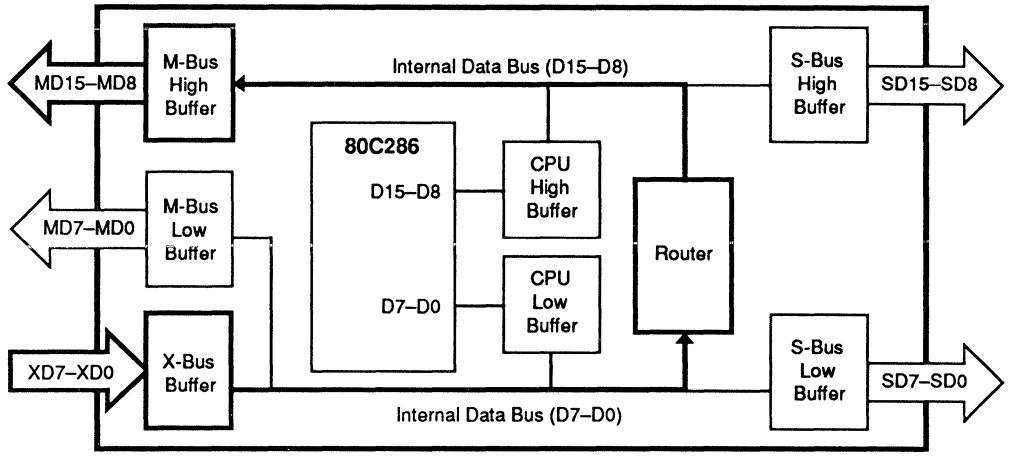
Master:	8-bit DMA1
Target:	8-bit X-bus I/O
Cycle Type:	Transfer from Odd Address Local Memory to X-bus I/O

Figure C–39



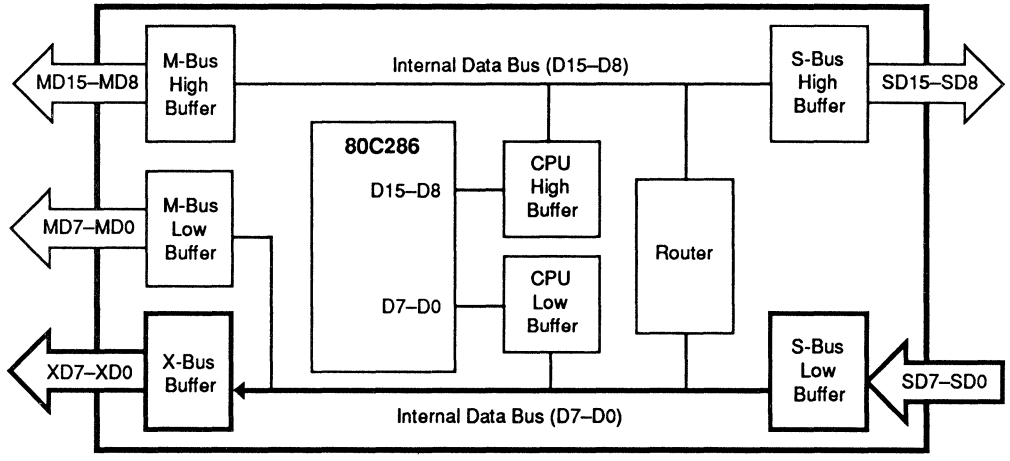
Master:	8-bit DMA1
Target:	8-bit X-bus I/O
Cycle Type:	Transfer to Even Address Local Memory from X-bus I/O

Figure C–40



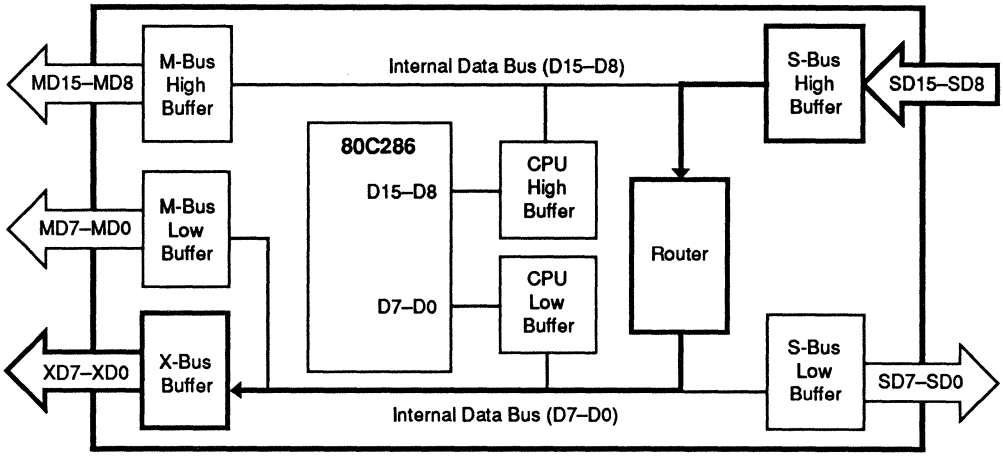
Master:	8-bit DMA1
Target:	8-bit X-bus I/O
Cycle Type:	Transfer to Odd Address Local Memory from X-bus I/O

Figure C–41



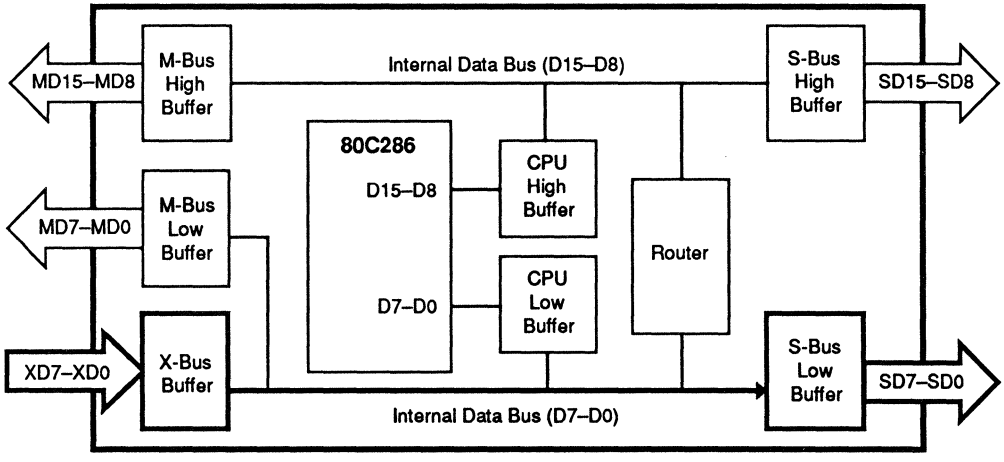
Master:	16-bit DMA1
Target:	8-bit X-bus I/O, AT Keyboard Controller
Cycle Type:	Even Address 16-bit or any 8-bit S-bus Memory to X-bus I/O

Figure C–42



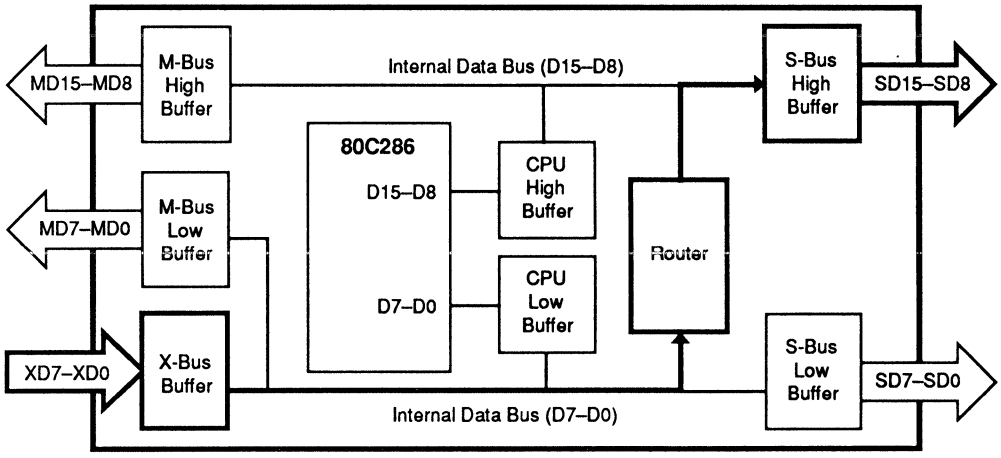
Master:	16-bit DMA1
Target:	8-bit X-bus I/O, AT Keyboard Controller
Cycle Type:	16-bit Odd Address S-bus Memory to X-bus I/O

Figure C–43



Master:	16-bit DMA1
Target:	8-bit S-bus Memory
Cycle Type:	Even Address 16-bit or any 8-bit S-bus Memory from X-bus I/O

Figure C–44



Master:	16-bit DMA1
Target:	8-bit Odd Address S-bus Memory
Cycle Type:	16-bit Odd Address S-bus Memory from X-bus I/O

PIN BUFFER REFERENCE

Pin	Pin Name	Buffer Type	Buffer Enable	Pull-up Enable	Reset State
1	LDATDIR	Output	Always		High
2	$\overline{\text{SBENA}}$	Output	Always		High
3	$\overline{\text{MASTER}}$	Input/Output, Strong Pull-up w/Enable	BM (BMCR.0)	See Note 1	
4	DRQ7	Input, Weak Pull-down		Always	
5	$\overline{\text{DACK7}}$	Output	Always		High
6	DRQ6	Input, Weak Pull-down		Always	
7	$\overline{\text{DACK6}}$	Output	Always		High
8	DRQ5	Input, Weak Pull-down		Always	
9	$\overline{\text{DACK5}}$	Output	Always		High
10	DRQ0	Input, Weak Pull-down		Always	
11	Vcc	Power			
12	Vss	Power			
13	$\overline{\text{DACK0}}$	Output	Always		High
14	$\overline{\text{MEMR}}$	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	High
15	$\overline{\text{MEMW}}$	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	HIgh
16	IRQ14	Input, Weak Pull-up		Always	
17	IRQ15	Input, Weak Pull-up		Always	
18	IRQ12	Input, Weak Pull-up		Always	
19	IRQ11	Input, Weak Pull-up		Always	
20	IRQ10	Input, Weak Pull-up		Always	
21	$\overline{\text{IOCS16}}$	Input, Strong Pull-up w/Enable		$\overline{\text{KSPUL}}$&(BCR.7)	Size Sense
22	$\overline{\text{MEMCS16}}$	Input, Strong Pull-up w/Enable		$\overline{\text{KSPUL}}$ & BCON	
23	$\overline{\text{SBHE}}$	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	HIgh
24	BALE	Output	Always		Low
25	DMATC	Output	Always		Low
26	$\overline{\text{DACK2}}$	Output	Always		HIgh
27	Vcc	Power			
28	Vss	Power			
29	IRQ3	Input, Weak Pull-up		Always	
30	IRQ4	Input, Weak Pull-up		Always	
31	IRQ5	Input, Weak Pull-up		Always	
32	IRQ6	Input, Weak Pull-up		Always	
33	IRQ7	Input, Weak Pull-up		Always	
34	$\overline{\text{REFRESH}}$	Input/Output, Strong Selectable Pull-up		$\overline{\text{KSPUL}}$ (BCR.7)	High
35	DRQ1	Input, Weak Pull-down		Always	
36	SYSCLK	Output	Always		Low
37	$\overline{\text{DACK1}}$	Output	Always		HIgh
38	DRQ3	Input/Output, Weak Pull-down	BM (BMCR.0)	Always	
39	$\overline{\text{DACK3}}$	Input/Output	BCON & $\overline{\text{BM}}$		Float
40	$\overline{\text{IOW}}$	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	High
41	$\overline{\text{IOR}}$	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	HIgh
42	$\overline{\text{SMEMW}}$	Output	Always		High
43	$\overline{\text{SMEMR}}$	Output	Always		High
44	IOCHRDY	Input, Strong Pull-up w/Enable		$\overline{\text{KSPUL}}$ & BCON	

Pin	Pin Name	Buffer Type	Buffer Enable	Pull-up Enable	Reset State
45	AEN	Output	Always		Low
46	$\overline{\text{PoWS}}$	Input, Strong Pull-up w/Enable		$\overline{\text{KSPUL}}$ & BCON	
47	DRQ2	Input, Weak Pull-down		Always	
48	IRQ9	Input, Weak Pull-up		Always	
49	RESETDRV	Output	Always		High
50	$\overline{\text{IOCHK}}$	Input, Strong Pull-up w/Enable		$\overline{\text{KSPUL}}$ & BCON	
51	XA17	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
52	XA16	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
53	XA15	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
54	XA14	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
55	XA13	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
56	Vcc	Power			
57	Vss	Power			
58	XA12	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
59	XA11	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
60	XA10	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
61	XA9	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
62	XA8	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
63	XA7	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
64	XA6	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
65	XA5	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
66	XA4	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
67	XA3	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
68	XA2	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
69	XA1	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
70	XA0	Output DHT, Pull-up (Note 3)	Per Cycle	Always	High
71	Vcc	Power			
72	Vss	Power			
73	MA0	Output	Always		Undefined
74	MA1	Output	Always		Undefined
75	MA2	Output	Always		Undefined
76	MA3	Output	Always		Undefined
77	MA4	Output	Always		Undefined
78	Vcc	Power			
79	Vss	Power			
80	MA5	Output	Always		Undefined
81	Vcc	Power			
82	Vss	Power			
83	MA6	Output	Always		Undefined
84	MA7	Output	Always		Undefined
85	MA8	Output	Always		Undefined
86	MA9	Output	Always		Undefined
87	MA10	Output	Always		Undefined
88	Vcc	Power			
89	Vss	Power			
90	SA0	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
91	SA1	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
92	SA2	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
93	SA3	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
94	SA4	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
95	SA5	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
96	SA6	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
97	SA7	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
98	SA8	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High

Pin	Pin Name	Buffer Type	Buffer Enable	Pull-up Enable	Reset State
99	SA9	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
100	SA10	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
101	SA11	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
102	SA12	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
103	SA13	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
104	SA14	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
105	SA15	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
106	SA16	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
107	SA17	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
108	Vcc	Power			
109	Vss	Power			
110	LA17	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	High
111	SA18	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
112	LA18	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	High
113	SA19	I/O DHT, Pull-up w/Enable (Notes 2,3)	BCON & Per Cycle	BCON (BCR.0)	High
114	LA19	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	High
115	LA20	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	High
116	LA21	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	High
117	LA22	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	High
118	LA23	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	High
119	$\overline{RAS0}$	Output	Always		High
120	$\overline{RAS1}$	Output	Always		High
121	Vcc	Power			
122	Vss	Power			
123	$\overline{RAS2}$	Output	Always		High
124	$\overline{RAS3}$	Output	Always		High
125	$\overline{CAS0}$	Output	Always		High
126	$\overline{CAS1}$	Output	Always		High
127	Vcc	Power			
128	Vss	Power			
129	$\overline{CAS2}$	Output	Always		High
130	$\overline{CAS3}$	Output	Always		High
131	$\overline{MWEL}$	Output	Always		High
132	$\overline{MWEH}$	Output	Always		High
133	DLYOUT	Output	Always		Low
134	DL0	Input			
135	Vcc	Power			
136	Vss	Power			
137	DL1	Input			
138	XD0	Input/Output, Keeper	Per Cycle		Undefined
139	XD1	Input/Output, Keeper	Per Cycle		Undefined
140	XD2	Input/Output, Keeper	Per Cycle		Undefined
141	XD3	Input/Output, Keeper	Per Cycle		Undefined
142	XD4	Input/Output, Keeper	Per Cycle		Undefined
143	XD5	Input/Output, Keeper	Per Cycle		Undefined
144	XD6	Input/Output, Keeper	Per Cycle		Undefined
145	XD7	Input/Output, Keeper	Per Cycle		Undefined
146	Vcc	Power			
147	Vss	Power			
148	MD0	Input/Output, Keeper	Per Cycle		Undefined
149	MD1	Input/Output, Keeper	Per Cycle		Undefined
150	MD2	Input/Output, Keeper	Per Cycle		Undefined
151	MD3	Input/Output, Keeper	Per Cycle		Undefined
152	MD4	Input/Output, Keeper	Per Cycle		Undefined

Pin	Pin Name	Buffer Type	Buffer Enable	Pull-up Enable	Reset State
153	MD5	Input/Output, Keeper	Per Cycle		Undefined
154	MD6	Input/Output, Keeper	Per Cycle		Undefined
155	MD7	Input/Output, Keeper	Per Cycle		Undefined
156	MD8	Input/Output, Keeper	Per Cycle		Undefined
157	MD9	Input/Output, Keeper	Per Cycle		Undefined
158	MD10	Input/Output, Keeper	Per Cycle		Undefined
159	MD11	Input/Output, Keeper	Per Cycle		Undefined
160	Vcc	Power			
161	Vss	Power			
162	MD12	Input/Output, Keeper	Per Cycle		Undefined
163	MD13	Input/Output, Keeper	Per Cycle		Undefined
164	MD14	Input/Output, Keeper	Per Cycle		Undefined
165	MD15	Input/Output, Keeper	Per Cycle		Undefined
166	MDP0	Input/Output, Keeper	Per Cycle		Undefined
167	MDP1	Input/Output, Keeper	Per Cycle		Undefined
168	SD0	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	Undefined
169	SD1	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	Undefined
170	SD2	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	Undefined
171	SD3	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	Undefined
172	SD4	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	Undefined
173	SD5	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	Undefined
174	SD6	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	Undefined
175	SD7	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	Undefined
176	Vcc	Power			
177	Vss	Power			
178	SD8	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	Undefined
179	SD9	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	Undefined
180	SD10	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	Undefined
181	SD11	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	Undefined
182	SD12	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	Undefined
183	SD13	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	Undefined
184	SD14	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	Undefined
185	SD15	Input/Output, Pull-up w/Enable (Note 2)	BCON & Per Cycle	BCON (BCR.0)	Undefined
186	VBATT	Power			
187	RTCLKX1	Oscillator Input			
188	RTCLKX2	Oscillator Output			
189	Vcc	Power			
190	Vss	Power			
191	$\overline{\text{LCSROM}}$	Output	Always		Undefined
192	$\overline{\text{XMEMW}}$	Output	Always		High
193	$\overline{\text{XMEMR}}$	Output	Always		High
194	$\overline{\text{CS8042}}$	Input/Output, Pull-up w/Enable	$\overline{\text{XTE}}$	XTE (PSR.2)	Undefined
195	$\overline{\text{XIOR}}$	Output	Always		High
196	$\overline{\text{XIOW}}$	Output	Always		High
197	$\overline{\text{RC}}$	Input			
198	A20G	Input			
199	$\overline{\text{RESET}}$	Output	Always		Low
200	KEYCLK	Input/Output, Pull-up w/Enable	$\overline{\text{XTE}}$	XTE (PSR.2)	F14MHZ/4
201	KBINT	Input			
202	PWRGOOD	Input			
203	PROCLK	Input			
204	IOCLK	Input			
205	F14MHZ	Input			
206	Vcc	Power			

Pin	Pin Name	Buffer Type	Buffer Enable	Pull-up Enable	Reset State
207	Vss	Power			
208	SPKR	Output	Always		Low
209	P287CLK	Output	Always		PROCLK/4
210	$\overline{\text{NPCS}}$	Output	Always		Undefined
211	$\overline{\text{BUSY}}$	Input, Pull-up		Always	
212	PEREQ	Input, Pull-down		Always	
213	$\overline{\text{PEACK}}$	Output	Always		High
214	RESET287	Output	Always		High
215	$\overline{\text{P287ERR}}$	Input, Pull-up		Always	
216	HDATDIR	Output	Always		High

Notes: 1. $\overline{\text{MASTER}}$ (pin 3) pull-up enable is: KSPUL & BCON & BM
KSPUL is BCR.7
BCON is BCR.0
BM is BMCR.0

2. There are some differences in the S-bus buffer enables when the device is configured in unbuffered Bus Master Mode (BM=1 and BCON=0). In this case, the buffers are enabled only during Bus Master access cycles after the bus acquisition is complete. The buffer enables affected in this way are:

   LA23–LA17
   SA1–SA0
   SD15–SD0
   $\overline{\text{SBHE}}$
   $\overline{\text{MEMW}}$
   $\overline{\text{MEMR}}$
   $\overline{\text{OW}}$
   $\overline{\text{OR}}$

3. A DHT output buffer driver, when instructed to disable the output, first drives the signal High before releasing the drive of the pin. There is also a weak pull-up on these buffers to maintain that High state after the buffer has stopped the strong drive of the pin. (DHT = Drive High, then three-state.)

APPENDIX E

QUICK REFERENCES

I/O MAP

Address Range (in Hex)	I/O Device Selected
000–01F	DMA Controller 1 (95C17)
020–03F, except 022,023	Interrupt Controller 1 (82C59)
022–023	Configuration Register Access, Index/Data
040–05F	Counter/Timer (82C54)
060–06F, except 061	Keyboard Interface: External 8042 (AT) or internal (XT)
061	Port B Register
070–07F	Real Time Clock and Static RAM, Index/Data
070	NMI Enable Register (Bit 7)
080–09F, except 092	DMA Page Register
080	Manufacturing Debug Port (X-bus echo)
092	Port 92 Fast Reset and Fast A20 Gate
0A0–0BF	Interrupt Controller 2 (82C59)
0C0–0DF	DMA Controller 2 (95C17)
0F0	Clear Numeric Coprocessor Busy Latch
0F1	Reset Numeric Coprocessor
0F8–0FF	External Numeric Coprocessor
02X8–02X9	Address Manager ATR and ATCR Registers (programmable)
0100–03FF	Possible X-bus I/O Devices (8 ranges of 8 bytes programmable)

MEMORY MAP

Address Range (in Hex)	Memory Selected
000000–09FFFF	Standard Real Mode Address Space (local or S Bus)
0A0000–0FFFFF	Possible Shadow RAM Address Space (16K enables)
0C0000–0FFFFF	Possible ROM Address Space on X Bus (16K enables)
100000–FEFFFF	Possible Extended Memory Space (local or S Bus)
FF0000–FFFFFF	Fixed High ROM Space (64K)

Am286ZX/LX PROCESSOR CONFIGURATION REGISTER QUICK REFERENCE

Name	Index	Default	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LRR1	00H	AAH	1	0	1	0	1	0	1	0
CCR1	01H	00H	FBE	BD1	BD0	BS	0	PD1	PD0	PS
CCR2	02H	C0H	KS1	KS0	KD1	KD0	0	CD1	CD0	CS
PSR	03H	00H	D16M1	D16M0	D8M1	D8M0	0	XTE	DMC1	DMC0
CDR	04H	FFH	M16C1	M16C0	M8C1	M8C0	I16C1	I16C0	I8C1	I8C0
WSR	05H	FFH	M16W1	M16W0	M8W1	M8W0	I16W1	I16W0	I8W1	I8W0
IPWS1	06H	44H	IIWS3	IIWS2	IIWS1	IIWS0	IDWS3	IDWS2	IDWS1	IDWS0
IPWS2	07H	44H	ICWS3	ICWS2	ICWS1	ICWS0	IMWS3	IMWS2	IMWS1	IMWS0
IPCD	08H	00H	IMCD1	IMCD0	ICCD1	ICCD0	IICD1	IICD0	IDCD1	IDCD0
BCR	09H	00H	KSPUL	0	0	0	FIO	SRST	0	BCON
BMCR	0AH	00H	0	0	0	HIE	HME	AS	BA	BM
BMPR	0BH	00H	BMP7	BMP6	BMP5	BMP4	BMP3	BMP2	BMP1	BMP0
SER1	0CH	00H	SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0
SER2	0DH	00H	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8
SER3	0EH	00H	SE23	SE22	SE21	SE20	SE19	SE18	SE17	SE16
SPR1	0FH	00H	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
SPR2	10H	00H	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8
SPR3	11H	00H	SP23	SP22	SP21	SP20	SP19	SP18	SP17	SP16
MWS	12H	01H	RCXIO	RCROM	RAMWS1	RAMWS0	ROMWS3	ROMWS2	ROMWS1	ROMWS0
ROBR1	13H	00H	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0
ROBR2	14H	C0H	RE15	RE14	RE13	RE12	RE11	RE10	RE9	RE8
RABR	15H	00H	D3T1	D3T0	D2T1	D2T0	D1T1	D1T0	D0T1	D0T0
DCR	16H	00H	STR	SRE	PE	RTE	0	PGE	INE	REL
ARL	18H	00H	0	0	0	AIE	ARL3	ARL2	ARL1	ARL0
XBE1	19H	00H	XBEN1	XB1.9	XB1.8	XB1.7	XB1.6	XB1.5	XB1.4	XB1.3
XBE2	1AH	00H	XBEN2	XB2.9	XB2.8	XB2.7	XB2.6	XB2.5	XB2.4	XB2.3
XBE3	1BH	00H	XBEN3	XB3.9	XB3.8	XB3.7	XB3.6	XB3.5	XB3.4	XB3.3
XBE4	1CH	00H	XBEN4	XB4.9	XB4.8	XB4.7	XB4.6	XB4.5	XB4.4	XB4.3
XBE5	1DH	00H	XBEN5	XB5.9	XB5.8	XB5.7	XB5.6	XB5.5	XB5.4	XB5.3
XBE6	1EH	00H	XBEN6	XB6.9	XB6.8	XB6.7	XB6.6	XB6.5	XB6.4	XB6.3
XBE7	1FH	00H	XBEN7	XB7.9	XB7.8	XB7.7	XB7.6	XB7.5	XB7.4	XB7.3
XBE8	20H	00H	XBEN8	XB8.9	XB8.8	XB8.7	XB8.6	XB8.5	XB8.4	XB8.3
XBD	21H	00H	0	0	0	0	XBD3	XBD2	XBD1	XBD0
XBSZ	22H	00H	XSZ8	XSZ7	XSZ6	XSZ5	XSZ4	XSZ3	XSZ2	XSZ1
PSFR	23H	00H	0	0	0	0	0	0	SSBY	CSTC
MSC	25H	00H	0	SEMS	0	MSS4	MSS3	MSS2	MSS1	MSS0

Sales Offices

North American

ALABAMA	(205) 882-9122
ARIZONA	(602) 242-4400
CALIFORNIA,	
Culver City	(213) 645-1524
Newport Beach	(714) 752-6262
Sacramento(Roseville)	(916) 786-6700
San Diego	(619) 560-7030
San Jose	(408) 452-0500
Woodland Hills	(818) 992-4155
CANADA, Ontario,	
Kanata	(613) 592-0060
Willowdale	(416) 224-5193
COLORADO	(303) 741-2900
CONNECTICUT	(203) 264-7800
FLORIDA,	
Clearwater	(813) 530-9971
Ft. Lauderdale	(305) 776-2001
Orlando (Longwood)	(407) 862-9292
GEORGIA	(404) 449-7920
IDAHO	(208) 377-0393
ILLINOIS,	
Chicago (Itasca)	(708) 773-4422
Naperville	(708) 505-9517
KANSAS	(913) 451-3115
MARYLAND	(301) 381-3790
MASSACHUSETTS	(617) 273-3970
MINNESOTA	(612) 938-0001
NEW JERSEY,	
Cherry Hill	(609) 662-2900
Parsippany	(201) 299-0002
NEW YORK,	
Liverpool	(315) 457-5400
Brewster	(914)279-8323
Rochester	(716) 272-9020
NORTH CAROLINA	
Harrisburg	(704) 455-1010
Raleigh	(919) 878-8111
OHIO,	
Columbus (Westerville)	(614) 891-6455
Dayton	(513) 439-0268
OREGON	(503) 245-0080
PENNSYLVANIA	(215) 398-8006
TEXAS,	
Austin	(512) 346-7830
Dallas	(214) 934-9099
Houston	(713) 376-8084
UTAH	(801) 264-2900

International

BELGIUM, Bruxelles	TEL	(02) 771-91-42
	FAX	(02) 762-37-12
	TLX	846-61028
FRANCE, Paris	TEL	(1) 49-75-10-10
	FAX	(1) 49-75-10-13
	TLX	263282F
GERMANY,		
Bad Homburg	TEL	(49) 6172-24061
München	TEL	(089) 4114-0
	FAX	(089) 406490
	TLX	523883
HONG KONG,	TEL	(852) 865-4525
Wanchai	FAX	(852) 865-1147
	TLX	67955AMDAPHX
ITALY, Milan	TEL	(02) 3390541
		(02) 3533241
	FAX	(02) 3498000
	TLX	843-315286
JAPAN,		
Atsugi	TEL	(0462) 29-8460
	FAX	(0462) 29-8458
Kanagawa	TEL	(0462) 47-2911
	FAX	(0462) 47-1729
Tokyo	TEL	(03) 3346-7550
	FAX	(03) 3342-5196
	TLX	J24064AMDTKOJ
Osaka	TEL	(06) 243-3250
	FAX	(06) 243-3253

International (Continued)

KOREA, Seoul	TEL	(82) 2-784-7598
	FAX	(82) 2-784-8014
LATIN AMERICA,		
Ft. Lauderdale	TEL	(305) 484-8600
	FAX	(305) 485-9736
	TLX	5109554261 AMDFTL
NORWAY, Hovik	TEL	(03) 010156
	FAX	(02) 591959
	TLX	79079
SINGAPORE	TEL	(65) 3481188
	FAX	(65) 3480161
	TLX	55650 AMDMMI
SWEDEN,		
Stockholm	TEL	(08) 733 03 50
(Sundbyberg)	FAX	(08) 733 22 85
	TLX	11602
TAIWAN	TEL	(886) 2-7213393
	FAX	(886) 2-7723422
UNITED KINGDOM,		
Manchester area	TEL	(0925) 828008
(Warrington)	FAX	(0925) 827693
	TLX	851-628524
London area	TEL	(0483) 740440
(Woking)	FAX	(0483) 756196
	TLX	851-859103

North American Representatives

CANADA	
Burnaby, B.C. - DAVETEK MARKETING	(604) 430-3680
Kanata, Ontario - VITEL ELECTRONICS	(613) 592-0060
Mississauga, Ontario - VITEL ELECTRONICS	(416) 676-9720
Lachine, Quebec - VITEL ELECTRONICS	(514) 636-5951
ILLINOIS	
HEARTLAND TECH MKTG, INC	(312) 577-9222
INDIANA	
Huntington - ELECTRONIC MARKETING CONSULTANTS, INC	(317) 921-3450
Indianapolis - ELECTRONIC MARKETING CONSULTANTS, INC	(317) 921-3450
IOWA	
LORENZ SALES	(319) 377-4666
KANSAS	
Merriam – LORENZ SALES	(913) 469-1312
Wichita – LORENZ SALES	(316) 721-0500
KENTUCKY	
ELECTRONIC MARKETING CONSULTANTS, INC	(317) 921-3452
MICHIGAN	
Birmingham - MIKE RAICK ASSOCIATES	(313) 644-5040
Holland – COM-TEK SALES, INC	(616) 392-7100
Novi – COM-TEK SALES, INC	(313) 344-1409
MINNESOTA	
Mel Foster Tech. Sales, Inc.	(612) 941-9790
MISSOURI	
LORENZ SALES	(314) 997-4558
NEBRASKA	
LORENZ SALES	(402) 475-4660
NEW MEXICO	
THORSON DESERT STATES	(505) 883-4343
NEW YORK	
East Syracuse – NYCOM, INC	(315) 437-8343
Woodbury – COMPONENT CONSULTANTS, INC	(516) 364-8020
OHIO	
Centerville – DOLFUSS ROOT & CO	(513) 433-6776
Columbus – DOLFUSS ROOT & CO	(614) 885-4844
Strongsville – DOLFUSS ROOT & CO	(216) 899-9370
OREGON	
ELECTRA TECHNICAL SALES, INC	(503) 643-5074
PENNSYLVANIA	
RUSSELL F. CLARK CO.,INC.	(412) 242-9500
PUERTO RICO	
COMP REP ASSOC, INC	(809) 746-6550
WASHINGTON	
ELECTRA TECHNICAL SALES	(206) 821-7442
WISCONSIN	
HEARTLAND TECH MKTG, INC	(414) 792-0920

Advanced Micro Devices, Inc. 901 Thompson Place, P.O. Box 3453, Sunnyvale, CA 94088, USA
Tel: (408) 732-2400 • TWX: 910-339-9280 • TELEX: 34-6306 • TOLL FREE: (800) 538-8450
APPLICATIONS HOTLINE & LITERATURE ORDERING • TOLL FREE: (800) 222-9323 • (408) 749-5703

4/22/91
EP-9.5M-6/91-0 Printed in USA

ADVANCED MICRO DEVICES, INC.
901 Thompson Place
P.O. Box 3453
Sunnyvale
California 94088-3453
(408) 732-2400
TWX: 910-339-9280
TELEX: 34-6306
TOLL-FREE
(800) 538-8450

APPLICATIONS HOTLINE & LITERATURE ORDERING
(800) 222-9323
(408) 749-5703

RECYCLABLE

Printed in USA
15620B