intel®

AP-468

APPLICATION NOTE

Quick Upgrade from the 80C186 to the 80C186EA

LARRY BATES
80C186 APPLICATIONS ENGINEERING

December 1991

Order Number: 272157-001

Information in this document is provided in connection with Intel products. Intel assumes no liability whatsoever, including infringement of any patent or copyright, for sale and use of Intel products except as provided in Intel's Terms and Conditions of Sale for such products.

Intel retains the right to make changes to these specifications at any time, without notice. Microcomputer Products may have minor variations to this specification known as errata.

*Other brands and names are the property of their respective owners.

†Since publication of documents referenced in this document, registration of the Pentium, OverDrive and iCOMP trademarks has been issued to Intel Corporation.

Contact your local Intel sales office or your distributor to obtain the latest specifications before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature, may be obtained from:

Intel Corporation
P.O. Box 7641
Mt. Prospect, IL 60056-7641

or call 1-800-879-4683

Quick Upgrade from the 80C186 to the 80C186EA

CONTENTS

PAGE

CONTENTS

PAGE

The 80C186EA/80C188EA is the second member of the 80C186 modular core family. It retains the standard 80C186's versatile DMA capabilities and adds two power management modes. The 80C186EA is well-suited for a broad spectrum of embedded data control designs where power consumption is a concern. The 80C186EA is ideal for compact, portable designs, including those operated from batteries.

1.0 PURPOSE OF THIS APPLICATION NOTE

This application note has two purposes. First, it highlights the features and benefits of the 80C186EA. Second, it shows how easy it is to upgrade an existing standard 80C186 application to the new 80C186EA. Upgrades require no software modification and little or no hardware modification in most cases.

2.0 THE 80C186EA/80C188EA DEBUTS

The 80C186EA is a high integration embedded processor belonging to the 80C186 modular core family. It is 100 percent code-compatible with the standard 80C186 and includes the following standard 80C186 integrated feature set:

- 16-Bit Execution Unit (EU) using familiar x86 instruction set and 1 Mbyte address space.
- Bus Interface Unit (BIU) with a full complement of control signals and support for alternate bus masters.
- Timer/Counter Unit (TCU) with three programmable 16-bit timers and internal or external clocking options.
- Refresh Control Unit (RCU) to simplify the interface to dynamic memory.
- Chip-Select Unit (CSU) with up to 13 external pins to support memory and I/O devices, including programmable wait states.
- Interrupt Control Unit (ICU) which allows complex interrupt priority schemes.
- Direct Memory Access Unit (DMU) with two channels for efficient data transfers.
- ONCE high-impedance test mode.

Intel offers the 80C188EA with an 8-bit wide data bus for economical product design. Intel also offers 3V (rated 2.7V–5.5V) versions designated as the 80L186EA and the 80L188EA. With the emphasis on low power, 3V operation and power management, Intel commands a leadership position with the 80C186EA family.

With its 80C186 heritage, the 80C186EA automatically arrives with considerable hardware and software development tools already in place.

3.0 SIMILARITIES AND DIFFERENCES BETWEEN THE 80C186EA AND THE STANDARD 80C186

The 80C186EA, with its 80C186 modular core, is 100 percent code-compatible to the original 80C186. The 80C186EA's Chip-Select, Refresh Control, Interrupt Control, Timer/Counter and DMA Units are identical to the original 80C186, too.

There are AC and DC specification differences. Before embarking on your design, obtain the latest specifications from the following data sheets:

80C186EA Data Sheet	#272019
80C188EA Data Sheet	#272020
80L186EA Data Sheet	#272021
80L188EA Data Sheet	#272022

To gain the most advantage from this application note, refer to the *80C186EA User's Manual* (#270950). This note highlights differences between the 80C186EA and the 80C188EA where applicable.

3.1 Program Execution

All existing 80C186 programs execute correctly on the 80C186EA without modification. All 80C186 control registers have the same offsets in the 80C186EA Peripheral Control Block. Although the register functions are identical, many register and bit names differ on the 80C186EA to conform to other 80C186 modular core family members.

3.2 Technology Comparison

The 80C186EA design is based on the modular core implementation of the Intel 80C186. The modular core executes instructions with complete clock-to-clock compatibility to the original 80C186 and is completely static.

Intel manufactures the 80C186EA using its 1 μm CHMOS-IV process. The smaller topology and static design results in power savings of over a third relative to the original 80C186. The 80C186EA has power management features for additional power savings, but you do not have to use the advanced features if you are performing a simple upgrade and do not need them.

3.3 Available Speeds

The 80C186 modular core family uses faster transistor technology than the standard 80C186, allowing Intel to offer faster speed selections. Intel markets the standard 80C186 at speeds of 10, 12.5 and 16 MHz. 80C186EA speeds currently are 12.5, 16 and 20 MHz, except the 3V versions at 8 MHz. The 80C186EA has no minimum frequency requirement. Its clock may stop and restart any time and the processor retains its state.

3.4 Available Packages

The 80C186EA comes only in economical plastic packages: a 68-lead Plastic Leaded Chip Carrier (PLCC) and an 80-lead EIAJ Quad Flat Pack (QFP). Except as discussed in Section 3.9.1, the PLCC 80C186EA pinout is equivalent to the standard 80C186 PLCC pinout. The QFP (EIAJ) 80C186EA pinout differs somewhat from its standard 80C186 counterpart.

The pinout of a socketed PLCC 80C186EA is equivalent to the pinout of a Pin Grid Array (PGA) standard 80C186 and the pinout of a socketed ceramic Leadless Chip Carrier (LCC) standard 80C186.

3.5 Available Temperature Selections

Intel offers the standard 80C186 with either commercial (0°C–70°C ambient) or extended (−40°C to +85°C ambient) temperature ratings. 80C186EA operation is specified for case temperatures, but the commercial and extended temperature ranges do exactly match the 0°C–70°C and −40°C to +85°C ambient temperature ranges, respectively.

3.6 Operating Modes

The concept of operating mode is different between the 80C186 and the 80C186EA.

The standard 80C186 leaves reset in either compatible mode or enhanced mode, depending on the $\overline{\text{TEST}}$/BUSY pin state. Compatible mode derived its name because of its likeness to the NMOS 80186. The standard 80C186 requires enhanced mode operation for power-save mode, the refresh control unit and the 80C187 math coprocessor interface. Enhanced mode changes three Mid-range Chip Select ($\overline{\text{MCS}}$) pins to handshaking pins for the math coprocessor.

The 80C186EA leaves reset with either numerics mode or regular operation. The $\overline{\text{TEST}}$/BUSY pin must be high on the rising edge of Reset Input ($\overline{\text{RESIN}}$) and low four clocks later to enable numerics mode. Numerics mode changes the three $\overline{\text{MCS}}$ pins to 80C187 handshaking pins. Power-save mode and the refresh control unit are always available on the 80C186EA.

An 80C186EA placed into an unmodified standard 80C186 design responds correctly, whether the original configuration was compatible or enhanced. All execution proceeds identically on a clock-for-clock basis. The processor activates new power management features only if the user programs them.

The 80C188EA, like the 80C188, does not have an interface for the 80C187. If an existing design configured the 80C188 for enhanced mode, the 80C188EA responds with regular operation.

Queue status and ONCE modes are identical between the standard 80C186 and the 80C186EA.

3.7 Modular Core and Peripherals

The 80C186EA execution and bus interface units are 100 percent compatible with the standard 80C186 on a clock-for-clock basis. The TCU, RCU, CSU, ICU, DMU and clock generator (with power-save mode) are also identical to their standard 80C186 counterparts.

Although the programming and operation of these peripherals is 100 percent compatible, some of the register and bit names changed in the data sheets and User's Manual to conform to other modular core products.

3.8 Power Management

The 80C186EA has three power management modes: idle, powerdown and power-save. Power-save mode is a clock generation function, while idle and powerdown modes are clock distribution functions. Power-save mode is identical to the standard 80C186 but powerdown and idle are new features.

3.8.1 IDLE MODE

During idle mode operation the clock signal goes only to the integrated peripheral devices, so the peripherals operate normally. The clocks to the modular core (execution and bus interface units) freeze in a logic low state, so execution stops. CLKOUT continues to toggle. Idle mode reduces current draw about a third, depending on the activity in the peripheral units.

Setting the IDLE bit in the power control register arms idle mode. The processor actually enters idle mode when it executes a HLT (halt) instruction. DMA requests, DRAM refresh requests and HOLD requests temporarily turn on the core clocks. DMA and DRAM refresh cycles running under idle mode are no different from other DMA and DRAM refresh cycles. After the bus activity ends, the core clocks again turn off.

The core clocks also turn on temporarily to issue HLDA (hold acknowledge) in response to a HOLD request. After the external bus master finishes using the bus and deasserts HOLD, the processor deasserts HLDA and again turns off the core clocks. Refresh requests occurring during bus hold force the BIU to deassert HLDA to regain control of the bus for the refresh cycle.

There are three ways to exit idle mode: unmasked interrupt, non-maskable interrupt (NMI) or reset. The processor does not retain any information after reset, so the user must reprogram the power control register and reexecute HLT to resume idle mode operation.

After the processor executes the IRET (interrupt return instruction) in the interrupt service routine, the instruction pointer points to the instruction following the HLT instruction. Interrupt execution does not clear the power control register, so the processor can resume idle mode operation by executing another HLT instruction.

3.8.2 POWERDOWN MODE

Powerdown mode freezes the clocks to the entire device (core and peripherals), disables the crystal oscillator and stops CLKOUT. All internal units (registers, state machines, etc.) maintain their state as long as V_{CC} is applied. The BIU cannot run DMA or DRAM refresh cycles nor honor HOLD requests in powerdown mode because the clocks are off. Current consumption in powerdown mode consists of just transistor leakage, typically less than 100 μA.

The procedure for entering powerdown is similar to the procedure for idle mode. Setting the PWRDN bit in the power control register arms powerdown mode. All clocks stop immediately upon execution of the HLT instruction.

The 80C186EA exits powerdown mode via an NMI or a reset. Unless an external oscillator feeds the CLKIN input, the circuit must allow some time for the crystal oscillator to restart.

For a reset, follow the usual procedure for a powerup (cold) reset. A simple RC network at the $\overline{\text{RESIN}}$ input pin fulfills this requirement.

For an NMI, an external Powerdown Timer (PDTMR) pin provides the timing delay. First, determine the startup time for your exact circuit over its full operating range using a storage oscilloscope. Be careful to compensate for scope loading. Then multiply the startup time by the constant C_{PD}, given in the 80C186EA data sheet, to obtain a capacitance value. With typical oscillator startup on the order of a couple of milliseconds, the PDTMR capacitor value will be less than 1 μF. After the processor executes the IRET in the NMI routine, the instruction pointer points to the instruction following the HLT. Remember that the PWRDN bit in the power control register remains set and that the processor can re-enter powerdown via HLT.

Driving the CLKIN input with a continuous external frequency source presents a special case. The processor doesn't need to wait for a crystal to start vibrating, so leave the PDTMR pin unconnected. The 80C186EA can exit powerdown mode without a timing delay.

3.8.3 POWER-SAVE MODE

Power-save mode is not a new feature. It is identical on the standard 80C186 and the 80C186EA. However, power-save mode deserves special attention on the 80C186EA because you can now use it to greater advantage.

Power-save mode enables a programmable clock divisor in the clock generation circuit. This divider operates in addition to the usual divide-by-two counter. Possible clock divisor settings are 1, 4, 8 and 16 (1 has no effect). The divided frequency feeds the core, the integrated peripherals and CLKOUT. The processor operates at the divided clock rate as if you used a lower frequency crystal or external oscillator. Unlike the standard 80C186, the 80C186EA has no lower frequency limit. You can use the maximum divisor, 16, regardless of the undivided frequency.

Operation of *both* the core and the integrated peripherals continues in power-save mode. It may be necessary to reprogram units such as the timer counter unit and the refresh control unit to compensate for the overall reduced clock rate.

The procedure for entering power-save mode is to program the divisor bits and the PSEN (Power-Save Enable) bit in the power-save register. The choices for leaving power-save mode are clearing the PSEN bit, an unmasked (hardware) interrupt or an NMI. Power-save changes go into effect only at the falling edge of T_3 bus cycle states.

Since current is linear with respect to frequency, power-save mode reduces current draw at least half. If the processor can run background tasks at reduced speed, power-save mode is well-suited, although the overhead might be considerable. Concurrent power management (entering power-save mode *and* entering idle mode) is an attractive option because the power reductions multiply.

3.9 Pin Compatibility

The functionality of the 80C186EA is directly compatible to the standard 80C186. The degree of pin compatibility depends on the package being used. Table 1 lists all pin names and their numbers, highlighting all changes. Functional differences noted for the 80C188 also apply to the 80C188EA.

Table 1. 80C186 to 80C186EA Pin Comparison Chart

Standard 80C186 Name	80C186EA Name	Standard C186 68-Lead Package	C186EA 68-Lead Package	Standard C186 80-Lead Package	C186EA 80-Lead Package	Notes
AD0	AD0	17	17	64	64	
AD1	AD1	15	15	66	66	
AD2	AD2	13	13	68	68	
AD3	AD3	11	11	70	70	
AD4	AD4	8	8	74	74	
AD5	AD5	6	6	76	76	
AD6	AD6	4	4	78	78	
AD7	AD7	2	2	80	80	
AD8	AD8	16	16	65	65	A8 (C188)
AD9	AD9	14	14	67	67	A9 (C188)
AD10	AD10	12	12	69	69	A10 (C188)
AD11	AD11	10	10	71	71	A11 (C188)
AD12	AD12	7	7	75	75	A12 (C188)
AD13	AD13	5	5	77	77	A13 (C188)
AD14	AD14	3	3	79	79	A14 (C188)
AD15	AD15	1	1	1	1	A15 (C188)
A16	A16	68	68	3	3	
A17	A17	67	67	4	4	
A18	A18	66	66	5	5	
A19/$\overline{S6}$	A19/$\overline{S6}$	65	65	6	6	
ALE/QS0	ALE/QS0	61	61	10	10	
$\overline{BHE}$	$\overline{BHE}$	64	64	7	7	$\overline{RFSH}$ (C188)
$\overline{S0}$	$\overline{S0}$	52	52	23	23	
$\overline{S1}$	$\overline{S1}$	53	53	22	22	
$\overline{S2}$	$\overline{S2}$	54	54	21	21	
$\overline{RD}$/$\overline{QSMD}$	$\overline{RD}$/$\overline{QSMD}$	62	62	9	9	
$\overline{WR}$/QS1	$\overline{WR}$/QS1	63	63	8	8	
ARDY	ARDY	55	55	20	20	
SRDY	SRDY	49	49	27	27	

Table 1. 80C186 to 80C186EA Pin Comparison Chart (Continued)

Standard 80C186 Name	80C186EA Name	Standard C186 68-Lead Package	C186EA 68-Lead Package	Standard C186 80-Lead Package	C186EA 80-Lead Package	Notes
$\overline{DEN}$	$\overline{DEN}$	39	39	38	39	
DT/$\overline{R}$	DT/$\overline{R}$	40	—	37	37	See PDTMR
$\overline{LOCK}$	$\overline{LOCK}$	48	48	28	28	
HOLD	HOLD	50	50	26	26	
HLDA	HLDA	51	51	25	25	
$\overline{RES}$	$\overline{RESIN}$	24	24	55	55	Name Change
RESET	RESOUT	57	57	18	18	Name Change
X1	CLKIN	59	59	16	16	Name Change
X2	OSCOUT	58	58	17	17	Name Change
CLKOUT	CLKOUT	56	56	19	19	
$\overline{TEST}$/BUSY	$\overline{TEST}$/BUSY	47	47	29	29	$\overline{TEST}$ (C188)
—	PDTMR	—	40	—	38	See DT/$\overline{R}$
NMI	NMI	46	46	30	30	
INT0	INT0	45	45	31	31	
INT1	INT1/$\overline{SELECT}$	44	44	32	32	Name Change
INT2/$\overline{INTA0}$	INT2/$\overline{INTA0}$	42	42	35	35	
INT3/$\overline{INTA1}$	INT3/$\overline{INTA1}$/IRQ	41	41	36	36	Name Change
$\overline{UCS}$	$\overline{UCS}$	34	34	45	45	
$\overline{LCS}$	$\overline{LCS}$	33	33	46	46	
$\overline{MCS0}$/PEREQ	$\overline{MCS0}$/PEREQ	38	38	39	40	$\overline{MCS0}$ (C188)
$\overline{MCS1}$/$\overline{ERROR}$	$\overline{MCS1}$/$\overline{ERROR}$	37	37	40	41	$\overline{MCS1}$ (C188)
$\overline{MCS2}$	$\overline{MCS2}$	36	36	41	42	
$\overline{MCS3}$/$\overline{NPS}$	$\overline{MCS3}$/$\overline{NCS}$	35	35	42	43	Name Change $\overline{MCS3}$ (C188)
$\overline{PCS0}$	$\overline{PCS0}$	25	25	54	54	
$\overline{PCS1}$	$\overline{PCS1}$	27	27	52	52	
$\overline{PCS2}$	$\overline{PCS2}$	28	28	51	51	
$\overline{PCS3}$	$\overline{PCS3}$	29	29	50	50	
$\overline{PCS4}$	$\overline{PCS4}$	30	30	49	49	
$\overline{PCS5}$/A1	$\overline{PCS5}$/A1	31	31	48	48	
$\overline{PCS6}$/A2	$\overline{PCS6}$/A2	32	32	47	47	
TMR OUT 0	T0OUT	22	22	57	57	Name Change
TMR IN 0	T0IN	20	20	59	59	Name Change
TMR OUT 1	T1OUT	23	23	56	56	Name Change
TMR IN 1	T1IN	21	21	58	58	Name Change

Table 1. 80C186 to 80C186EA Pin Comparison Chart (Continued)

Standard 80C186 Name	80C186EA Name	Standard C186 68-Lead Package	C186EA 68-Lead Package	Standard C186 80-Lead Package	C186EA 80-Lead Package	Notes
DRQ0	DRQ0	18	18	61	61	
DRQ1	DRQ1	19	19	60	60	
V_{SS}	V_{SS}	26, 60	26, 60	12, 13, 53	12, 13, 24, 53, 62	
V_{CC}	V_{CC}	9, 43	9, 43	33, 34, 72, 73	2, 33, 34, 44, 72, 73	
NO CONNECT	NO CONNECT	—	—	2, 11, 14, 15, 24, 43, 44, 62, 63	11, 14, 15, 63	

3.9.1 68-LEAD PLASTIC LEADED CHIP CARRIER (PLCC)

Upgrades to existing 80C186 PLCC designs may not require a new layout. The only PLCC pin difference is lead #40, which is the Data Transmit/Receive ($DT/\overline{R}$) lead on the standard 80C186 and the PDTMR lead on the 80C186EA. If the original design does not use $DT/\overline{R}$ to control the data direction in bus transceivers, no layout changes are needed. If you wish to use the 80C186EA powerdown mode, you must connect a capacitor at PDTMR to time the processor's exit from powerdown mode (unless you also use an external oscillator).

If $DT/\overline{R}$ is needed, reconstruct it by latching status signal $\overline{S1}$ with a transparent latch gated by Address Latch Enable (ALE). The timing difference between the processor's $DT/\overline{R}$ (if it were available) and synthesized $DT/\overline{R}$ equals the propagation delay through the latch, since $DT/\overline{R}$, bus cycle status ($\overline{S2:0}$) and ALE all change state at the falling edge of T_4.

3.9.2 80-LEAD QUAD FLAT PACK (QFP-EIAJ)

The 80C186EA fine pitch QFP package does provide the $DT/\overline{R}$ signal. The 80C186EA has four more power and ground pins than the standard 80C186 and several pin functions changed position. You must use a new board layout for the 80C186EA QFP package.

3.10 DC Compatibility

Intel manufactures both the standard 80C186 and the 80C186EA in CMOS logic. However, the standard 80C186 has TTL-compatible inputs while the 80C186EA has CMOS-compatible inputs. TTL logic in existing 80C186 designs must change to CMOS logic if the outputs drive the 80C186EA. For all-CMOS circuits, use devices such as those from the HC and AC logic families. Corresponding members of the HCT and ACT families have TTL-compatible inputs. Pullup resistors are inconsistent with choosing the 80C186EA for low power because of added current draw.

CMOS-level inputs have several advantages. The main advantage is improved noise margin. The standard 80C186 has a V_{OH} minimum of 2.4V and a V_{IH} minimum of 0.2 V_{CC} + 0.9V, for a noise margin of 0.5V (with 5V operation). The 80C186EA has a V_{OH} minimum of V_{CC} − 0.5V and a V_{IH} minimum of 0.7 V_{CC}, for a noise margin of 1.0V (with 5V operation).

The standard 80C186 data sheet references AC timings to 1.5V (the TTL switchpoint). The 80C186EA data sheet references AC timings to $V_{CC}/2$ (the CMOS switchpoint). Reducing the operating voltage (80L186EA/80L188EA) directly scales the specified reference point.

The V_{CC} tolerance of the 16 MHz standard 80C186 was ±5%. The V_{CC} tolerance of the 16 MHz 80C186EA improved to ±10%.

3.11 AC Compatibility

Intel specifies 80C186EA AC timings in a simplified format consistent with other 80C186 modular core family members. Table 2 lists all standard 80C186 AC timing mnemonics and their 80C186EA equivalents.

Table 2. 80C186 to 80C186EA AC Timing Mnemonic Comparison

Standard 80C186 AC Timing Mnemonic	Parameter	Equivalent 80C186EA AC Timing Mnemonic
T_{DVCL}	Data in Setup (A/D)	T_{CLIS}
T_{CLDX}	Data in Hold (A/D)	T_{CLIH}
T_{CHSV}	Status Active Delay	T_{CHOV1}
T_{CLSH}	Status Inactive Delay	T_{CLOV2}
T_{CLAV}	Address Valid Delay	T_{CLOV1} (A19:16, $\overline{DEN}$), T_{CLOV2} (AD15:0)
T_{CLAX}	Address Hold	T_{CLOV1} (A19:16), T_{CLOV2} (AD15:0)
T_{CLDV}	Data Valid Delay	T_{CLOV1} (A19:16), T_{CLOV2} (AD15:0)
T_{CHDX}	Status Hold Time	Eliminated
T_{CHLH}	ALE Active Delay	T_{CHOV1}
T_{LHLL}	ALE Width	T_{LHLL}
T_{CHLL}	ALE Inactive Delay	T_{CHOV1}
T_{AVLL}	Address Valid to ALE Low	T_{AVLL}
T_{LLAX}	Address Hold from ALE Inactive	T_{LLAX}
T_{AVCH}	Address Valid to Clock High	Eliminated
T_{CLAZ}	Address Float Delay	T_{CLOF}
T_{CLCSV}	Chip-Select Active Delay	T_{CLOV2}
T_{CXCSX}	Chip-Select Hold from Command Inactive	T_{RHPH} ($\overline{RD}$), T_{WHPH} ($\overline{WR}$)
T_{CHCSX}	Chip-Select Inactive Delay	T_{CHOV2}
T_{DXDL}	$\overline{DEN}$ Inactive to DT/$\overline{R}$ Low	T_{DXDL}
T_{CVCTV}	Control Active Delay 1	T_{CHOV1} ($\overline{DEN}$), T_{CLOV2} ($\overline{WR}$, $\overline{INTA}$)
T_{CVDEX}	$\overline{DEN}$ Inactive Delay	T_{CLOV1}
T_{CHCTV}	Control Active Delay 2	T_{CHOV1}
T_{CLLV}	$\overline{LOCK}$ Valid/Invalid Delay	T_{CLOV1}
T_{AZRL}	Address Float to Read Active	T_{AFRL}
T_{CLRL}	$\overline{RD}$ Active Delay	T_{CLOV2}
T_{RLRH}	$\overline{RD}$ Pulse Width	T_{RLRH}
T_{CLRH}	$\overline{RD}$ Inactive Delay	T_{CLOV2}
T_{RHLH}	$\overline{RD}$ Inactive to ALE High	T_{RHLH}
T_{RHAV}	$\overline{RD}$ Inactive to Address Active	T_{RHAV}
T_{CLDOX}	Data Hold Time	T_{CLOV2}
T_{CVCTX}	Control Inactive Delay	T_{CLOV2} ($\overline{WR}$, $\overline{INTA}$), T_{CHOV1} ($\overline{DEN}$)

Table 2. 80C186 to 80C186EA AC Timing Mnemonic Comparison (Continued)

Standard 80C186 AC Timing Mnemonic	Parameter	Equivalent 80C186EA AC Timing Mnemonic
T_{WLWH}	$\overline{WR}$ Pulse Width	T_{WLWH}
T_{WHLH}	$\overline{WR}$ Inactive to ALE High	T_{WHLH}
T_{WHDX}	Data Hold after $\overline{WR}$	T_{WHDX}
T_{WHDEX}	$\overline{WR}$ Inactive to $\overline{DEN}$ Inactive	T_{WHDEX}
T_{CKIN}	CLKIN Period	T_C
T_{CLCK}	CLKIN Low Time	T_{CL}
T_{CHCK}	CLKIN High Time	T_{CH}
T_{CKHL}	CLKIN Fall Time	T_{CF}
T_{CKLH}	CLKIN Rise Time	T_{CR}
T_{CICO}	CLKIN to CLKOUT Skew	T_{CD}
T_{CLCL}	CLKOUT Period	T
T_{CLCH}	CLKOUT Low Time	T_{PL}
T_{CHCL}	CLKOUT High Time	T_{PH}
T_{CH1CH2}	CLKOUT Rise Time	T_{PR}
T_{CL2CL1}	CLKOUT Fall Time	T_{PF}
T_{SRYCL}	Synchronous Ready (SRDY) Transition Setup Time	T_{CLIS}
T_{CLSRY}	SRDY Transition Hold Time	T_{CLIH}
T_{ARYCH}	ARDY Resolution Transition Setup Time	T_{CHIH}
T_{CLARX}	ARDY Active Hold Time	T_{CLIH}
T_{ARYCHL}	ARDY Inactive Holding Time	T_{CHIH}
T_{ARYLCL}	Asynchronous Ready (ARDY) Setup Time	T_{CLIS}
T_{INVCH}	INTx, NMI, $\overline{TEST}$/BUSY, TMR IN Setup Time	T_{CHIH}
T_{INVCL}	DRQ0, DRQ1 Setup Time	T_{CLIH}
T_{CLTMV}	Timer Output Delay	T_{CLOV1}
T_{CHQSV}	Queue Status Delay	Eliminated
T_{RESIN}	$\overline{RES}$ Setup	T_{CLIS}
T_{HVCL}	HOLD Setup	T_{CLIS}
T_{CLRO}	Reset Delay	T_{CLOV1}
T_{CLHAV}	HLDA Valid Delay	T_{CLOV1}
T_{CHCZ}	Command Lines Float Delay	T_{CHOF}
T_{CHCV}	Command Lines Valid Delay (after Float)	T_{CHOV1} (A19:16, $\overline{BHE}$, DT/$\overline{R}$, $\overline{S2:0}$, $\overline{LOCK}$), T_{CHOV2} ($\overline{RD}$, $\overline{WR}$)

Since the 80C186EA can run faster than the standard 80C186, compare data sheet specifications carefully before using the 80C186EA in your design. Please note DATA SHEET SPECIFICATIONS ARE SUBJECT TO CHANGE. If timing margins appear very tight in your new 80C186EA design, be sure to consider the difference in reference points for AC timings as well as the timing values.

80C186EA input specifications generally show equivalent or better setup and hold timings. 80C186EA output specifications generally show equivalent or better maximum timings. However, several minimum timings for signals such as Byte High Enable ($\overline{BHE}$) and $\overline{S2:0}$ are 3 ns for the 80C186EA but 5 ns for the standard 80C186. This difference might cause a problem if your application has synchronous (clocked) logic derived from 80C186 control signals. See Section 4.4 for more information on "glue logic" for the 80C186EA.

4.0 SYSTEM DESIGN CONSIDERATIONS

Microprocessor-based hardware falls into four broad categories: CPU, memory, peripheral logic and miscellaneous control logic. This section contains information applicable to each portion of your 80C186EA design. This section may be particularly helpful if you are considering a speed upgrade to an existing standard 80C186 design.

4.1 Designing for High-Speed CPUs

High-speed design techniques are vital for all 80C186EA circuits. Anticipate significant switching noise on V_{CC} and V_{SS}. To reduce the impedance in power and ground connections, use a circuit board with power and ground planes and avoid sockets. Use a bypass capacitor for every logic device package and at least four for the 80C186EA.

Pay careful attention to board layout. Minimize trace lengths for clocks, buses and other high-frequency signals and terminate long bus lines wherever practical.

At 20 MHz, you may find that more memory devices and peripherals have trouble floating the data bus fast enough after a read operation. Compare the 80C186EA T_{RHAV} specification with the output float time given for the memory or peripheral. If the memory or peripheral cannot meet the timing requirement, you need to place it on a buffered bus to prevent contention.

4.2 Memory Speed Requirements

The following sections show access time requirements for various memory types. Note that 8 MHz calculations use 80L186EA timings and 10 MHz calculations use 80C186EA12 timings.

4.2.1 EPROM, FLASH MEMORY AND SRAM DEVICES

Access time from address valid to data valid for EPROM, flash memory and SRAMs is calculated as $3 * T - T_{CLOV2} - T_{PD} - T_{CLIS}$. T is the CLKOUT period, T_{CLOV2} is the maximum 80C186EA address output delay, T_{PD} is the maximum propagation delay through an address latch, and T_{CLIS} is the data setup time to the falling edge of T_4 in an 80C186EA read cycle. Table 3 lists commonly available component access times which satisfy the timing requirements with at least a small amount of timing margin. At high frequencies, you may need faster memory than shown in the table to satisfy bus float time (T_{RHAV}) requirements.

Table 3. 80C186EA Access Time Requirements for EPROMs, Flash Memory and SRAMs

Speed (MHz)	EPROM, SRAM, Flash Access Time	@ Wait States
8	250	0
10	250	1
	200	0
12.5	200	1
	170	0
16	170	1
	120	0
20	120	1
	85	0

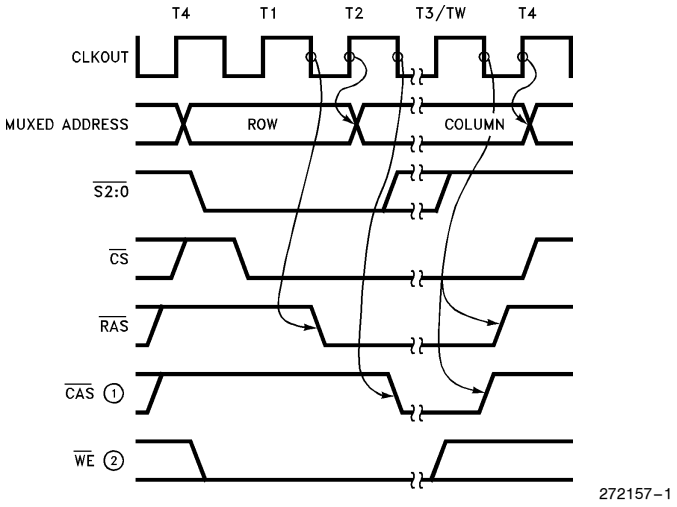
4.2.2 DRAM DEVICES

DRAMs have numerous critical timing parameters, so one computes DRAM speeds differently. The key parameters to consider are the DRAM's row access time and cycle time. Cycle time relates directly to the minimum length of a processor bus cycle. The timing analysis must also consider delays introduced by the DRAM controller, a necessary circuit for any DRAM memory system.

The DRAM control timing relationships shown in Figure 1 are well-proven compromises between DRAM requirements and the 80C186 Modular Core's four-state bus cycle. The DRAM controller generates Row Address Strobe ($\overline{RAS}$) approximately on the falling edge of T_2. Access time from $\overline{RAS}$ is calculated as $2 * T - T_{PD} - T_{CLIS}$. T is the CLKOUT period, T_{PD} is the propagation delay from falling CLKOUT to $\overline{RAS}$ active due to the DRAM controller, and T_{CLIS} is the data setup time to the falling edge of T_4. If the DRAM controller cannot meet specifications for the $\overline{RAS}$ to $\overline{CAS}$ (Column Address Strobe) delay or the $\overline{RAS}$ to column address delay, you can compute access time differently. Table 4 lists common component access times which satisfy the timing requirements with at least a small amount of timing margin. At high frequencies, you may need faster DRAMs than shown in the table to satisfy bus float time requirements. Section 4.4.2 and Appendix A discuss using DRAMs in more detail.

Table 4. 80C186EA Access Time Requirements for DRAMs

Speed (MHz)	DRAM Access Time	@ Wait States
8–10	150	0
12.5	150 120	1 0
16	150 80	1 0
20	120 70	1 0



NOTES:
1. $\overline{CAS}$ is unnecessary for refresh cycles only.
2. $\overline{WE}$ is necessary for write cycles only.

Figure 1. Suggested DRAM Control Signal Timing Relationships

4.3 Peripheral Logic

The application usually dictates the choice of peripherals. If a peripheral is too slow or lacks the necessary interface signals, the user adds extra components to generate wait states, buffer the bus or form control logic.

Among the peripheral devices used with the 80C186EA is Intel's 80C187 numerics coprocessor. Intel does not market a 20 MHz version of the 80C187. However, the interface is asynchronous and all speed combinations meet all timing requirements. This includes a 20 MHz 80C186EA and a 16 MHz 80C187.

4.4 Miscellaneous Control Logic

This broad category includes all the "glue logic" necessary to make the rest of the system work, including such circuits as:

- DRAM Controllers
- Controllers for pseudo-static RAMs and dual-ported memory
- Wait State Generators (extra chip-selects, too)
- Bus Control Signal Decoders
- DMA Acknowledge Generators
- Bus Arbiters
- Peripheral Interfaces
- Control and Diagnostic Registers

In some cases, this logic must synthesize signals not directly provided by the processor to "glue on" peripherals. In other cases, this circuitry must overcome timing mismatches or errata. Some of these logic subsystems must synchronously track the operation of the 80C186EA and its control signals. With higher CPU speeds and shorter minimum AC timings, the 80C186EA may require faster logic for bus tracking than its standard 80C186 predecessor.

The remainder of Section 4.4 describes situations which call for special control logic.

4.4.1 CHIP-SELECTS AND WAIT STATE GENERATION

The 80C186EA chip-select unit is identical to the standard 80C186 chip-select unit. It is not possible to cover the entire one Mbyte address space with the chip-selects provided. The maximum number of programmable wait states is three, which may be a limitation at high frequencies. 80C186EA users may need additional decoding circuitry to generate chip-selects and control the Asynchronous Ready (ARDY) or Synchronous Ready (SRDY) input pins to generate wait states. (The 80C186EB and the 80C186EC have an improved chip-select unit which overcomes such limitations.)

Figure 2 is an 80C186EA bus cycle with five wait states. You can build a synchronous wait state generator using a programmable logic device (PLD) clocked by the CPU's CLKOUT. The circuit samples ALE and a programmed chip-select pin. At the rising edge of T_2, the wait state generator drives the SRDY input low. The processor samples SRDY low at the falling edge of T_3 and each falling CLKOUT edge during the wait states. When the wait state generator counts up to the fourth wait state, it drives SRDY high on rising clock. After the 80C186EA samples SRDY high, it runs the fifth wait state before continuing with T_4.

There are several timings to check carefully for this wait state controller. The PLD flip-flops must need a hold time less than 3 ns because that is the minimum value for the 80C186EA's ALE valid (T_{CHOV1}) and chip-select valid (T_{CHOV2}) specifications. The PLD must also be fast enough so that the minimum CLKOUT High Time (T_{PH}) minus the maximum PLD Clock to Output Valid Time (T_{CO}) is greater than the minimum SRDY setup time (T_{CLIS}).

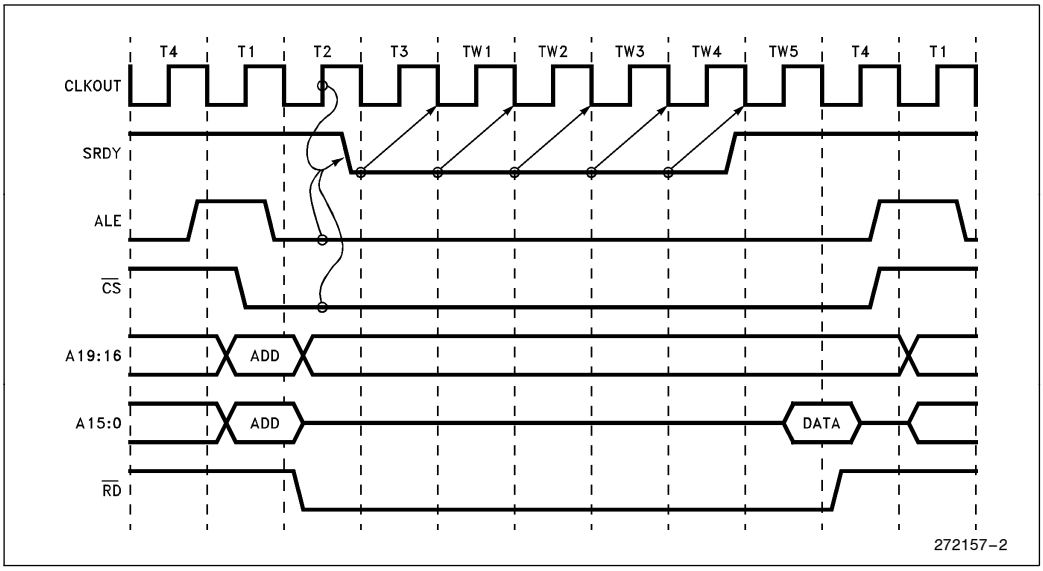


Figure 2. 80C186EA Bus Cycles with Five Wait States

4.4.2 DRAM CONTROL

All members of the 80C186 modular core family contain a refresh control unit. A complete DRAM controller requires additional circuitry outside the processor even in the simplest configurations.

There are four basic functions which a DRAM controller must provide:

- Map 80C186EA-generated addresses into the DRAM address pins.
- Translate processor control signals into DRAM control signals.
- Refresh the DRAMs at regular intervals.
- Overcome loading and transmission line effects.

Turn back to Figure 1 and examine the DRAM bus cycle. A general DRAM cycle has six phases:

1. First, the controller supplies a row address to the DRAM (meeting setup and hold times).
2. Next, the controller asserts $\overline{RAS}$, which latches the row address inside the DRAMs.
3. Then the controller supplies a column address to the DRAMs (meeting setup and hold times). Row and column addresses are multiplexed on the same DRAM pins to minimize the pin count.
4. The controller asserts a $\overline{CAS}$, which latches the column address inside the DRAMs.
5. Data is written to the DRAM if Write Enable ($\overline{WE}$) is asserted. Otherwise, the DRAM drives data on the bus for a read operation.
6. After the strobes go inactive, the controller must provide a recovery interval ($\overline{RAS}$ precharge) before the next cycle.

In the simplest DRAM refresh scheme, the controller provides only row addresses and $\overline{RAS}$ to perform DRAM refresh on a row-by-row basis. Appendix A describes an 80C186EA DRAM controller in detail.

4.4.3 82C59A INTERFACE

A popular peripheral for the 80C186EA is the 82C59A programmable interrupt controller. Unfortunately, the speed selections of the 82C59A have not kept up with the increasing speed capability of the 80C186 family.

For the 80C186 family, find the fastest 82C59A you can. At least one vendor manufactures a "10 MHz" version which avoids frustrating interface problems. The 82C59A is not a clocked device; "10 MHz" refers to a microprocessor speed to which it is matched. The following recommendations are based on using a "10 MHz" 82C59A device with a 16 MHz or 20 MHz 80C186EA.

Read access time for the 82C59A is calculated as 3 * T − T_{CLOV2} − T_{PD} − T_{CLIS}. The 80C186EA's minimum read, write and interrupt acknowledge ($\overline{INTA}$) pulse width is (2 * T) − 5. Bus accesses to the 82C59A must have at least one wait state. The wait state gives the 82C59A longer to return data to the 80C186EA and it ensures the widths of the $\overline{RD}$, $\overline{WR}$ and $\overline{INTA}$ strobes are wide enough for the 82C59A. Program a peripheral chip-select to provide wait states during 82C59A read and write cycles. Wait states for 82C59A INTA cycles require an external wait state generator (see Figure 3) because programmable chip-selects do not apply to INTA cycles.

Figure 4 shows 80C186EA Cascade Mode INTA cycles with wait state control furnished by the 74AC109 J-$\overline{K}$ Flip-Flop circuit of Figure 3. Inverted ALE asynchronously sets the flip-flop on every bus cycle, driving the $\overline{Q}$ output to a logic 0. The $\overline{Q}$ output connects to the processor's $\overline{SRDY}$ pin. At the rising edge of T_2, the J input from $\overline{INTA}$ is a logic 0 and the $\overline{K}$ input from $\overline{DEN}$ is a logic 1, so the flip-flop state does not change. The CPU asserts $\overline{DEN}$ upon rising T_2. On the falling edge of T_3, the 80C186EA recognizes SRDY as low and prepares to insert one wait state. At the rising edge of T_3, both inputs to the flip-flop are low, clearing the flip-flop. At the falling edge of T_W, the processor recognizes SRDY as high and prepares to terminate the bus cycle.

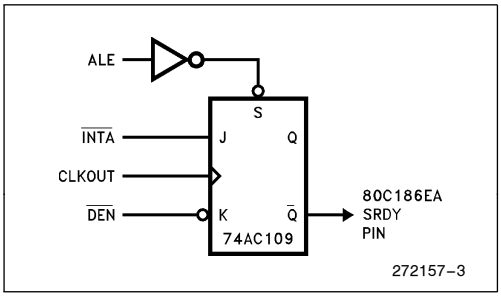


Figure 3. Wait State Generator for 82C59A Interface (1 Wait State)

This simple circuit drives the SRDY input signal to logic 0 on non-INTA cycles. Provide bus ready for other bus cycle types through chip-select programming or the ARDY pin. Also note that the timing margin for the SRDY setup time T_{CLIS} disappears as the processor speed approaches 20 MHz.

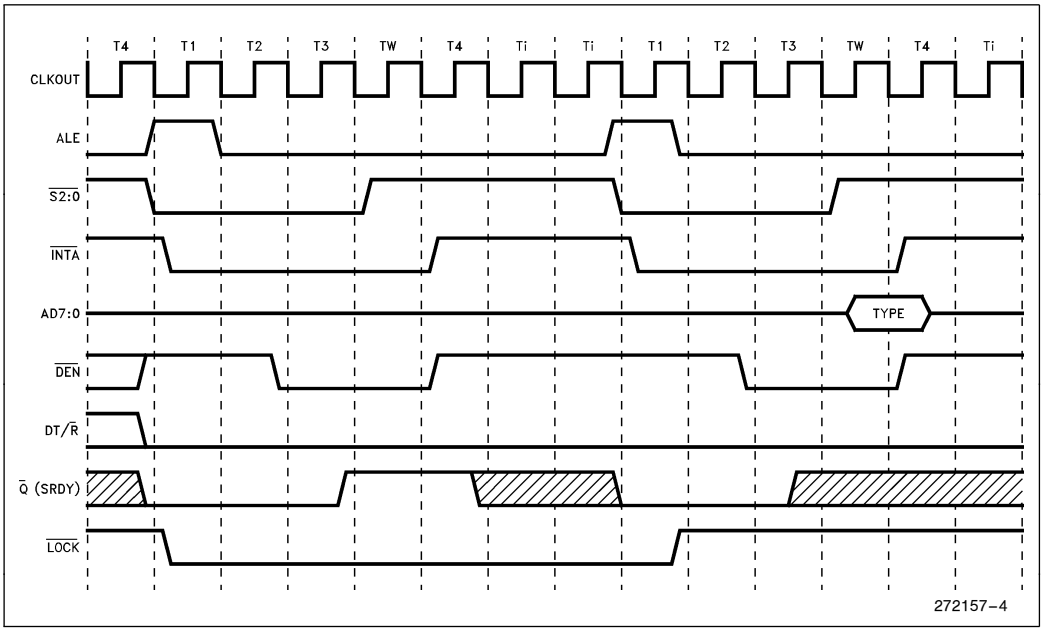


Figure 4. 80C186EA Cascade Mode INTA Cycles with One Wait State

Lock all reads and writes to the control registers of the 82C59A with the LOCK instruction prefix. This precaution ensures meeting the 82C59A Command Recovery Time for dissimilar cycles by inserting two idle T-states at the end of reads and writes. The 80C186EA automatically adds two idle states to each INTA bus cycle (see Figure 4), ensuring the system meets a similar recovery time specification for back-to-back INTA cycles.

Finally, be sure to place the 82C59A on a buffered data bus. At high CPU speeds the interrupt controller cannot float its data buffers fast enough after read cycles to avoid contention with the microprocessor.

5.0 OTHER UPGRADE CONSIDERATIONS

The following is a rundown of all known standard 80C186 errata and their disposition on the 80C186EA:

- **Non-Contiguous INTA Cycles ($\overline{LOCK}$/INTA Cycles)**—If a standard 80C186 received an interrupt during a locked bus cycle, the bus interface unit did not correctly lock subsequent INTA cycles. 80C186EA INTA cycles are correctly locked and contiguous.
- **V_{IH} on SRDY and ARDY**—The standard 80C186 required a higher than normal voltage to guarantee a logic 1 on these pins, making them sensitive to noise. The voltage thresholds of the 80C186EA SRDY and ARDY pins are the same as for other input pins.
- **Bus Preemption**—If the standard 80C186 attempts to perform a DRAM refresh cycle during bus hold, the processor can hang. This behavior depends on pending instruction execution. The 80C186EA always arbitrates the bus correctly to run refresh cycles.
- **80C188 $\overline{RFSH}$ Pin**—The standard 80C188 $\overline{RFSH}$ pin goes active and inactive prematurely in the T_4 bus cycle phase. The 80C188EA asserts and deasserts its $\overline{RFSH}$ pin correctly on T_1 of its bus cycles.
- **FWAIT/$\overline{ERROR}$**—During execution of the 80C187 FWAIT instruction, 80C186 family processors do not test the $\overline{ERROR}$ pin. 80C187 errors go undetected unless another numerics instruction follows, or you route errors to an external interrupt pin. This situation continues to exist on the 80C186EA but Intel declassified it as an errata. See the 80C187 data sheet.
- **Interrupt Status Register**—The processor may ignore a timer interrupt request during a write to this standard 80C186 register or redirect it to the wrong interrupt vector. Usually this problem happens during a write to clear the DHLT bit. This situation continues to exist on the 80C186EA but Intel declassified it as an errata. Continue to disable time interrupts while writing to the Interrupt Status Register, or use the DMA Control Registers to turn off DMA.

The 80C186EA has two registers not found on the standard 80C186. The Power Control Register (PWRCON) is located at offset F2H. It enables and disables idle mode and powerdown (see Section 3.8). Check existing software for spurious writes to the power control register location. Accidentally programming the register with either 0000H or FFFFH disables both modes. The Stepping ID (STEPID) Register at offset F6H is a read-only register to indicate the silicon stepping of the device.

APPENDIX A
80C186EA DRAM CONTROLLER EXAMPLE

This section demonstrates how to incorporate the 80C186EA DRAM Refresh Unit into a fully functional DRAM control system.

A.1 Design Requirements

Designing DRAM controllers is not as intuitive as designing other types of digital logic. DRAMs pose particularly fussy timing problems, and the choices of access time, wait states and refresh method can be overwhelming. This design applies only to the basic DRAM access method described in Section 4.2.2. DRAM systems employing the basic access method are suitable for most 80C186EA designs.

A successful controller must respond correctly to all 80C186EA bus cycles. It must discriminate between bus cycles intended for the DRAMs and other accesses. For DRAM bus cycles, it must generate all the necessary outputs, including chip-selects, Write Enables and Output Enables, with timing relationships similar to Figure 1. It must also be fast enough to not require excessive wait states.

A.2 What the 80C186EA Provides

The 80C186EA bus interface unit provides 20-bit addresses for routine read and write cycles. The integrated RCU provides similar 20-bit addresses for refresh cycles. The benefits of the DRAM refresh feature come from how the 80C186EA generates them and encodes them in the control signals.

Figure 5 is a block diagram of the RCU. It consists of a nine-bit timing counter, a nine-bit address counter, three control registers and interface logic. When the RCU is enabled, the timing counter decrements every CLKOUT cycle until it reaches a count of 1. At that time a refresh bus request is initiated, the timing counter reloads, and the operation continues. A refresh bus cycle has high priority, and runs as soon as the 80C186EA bus is free.

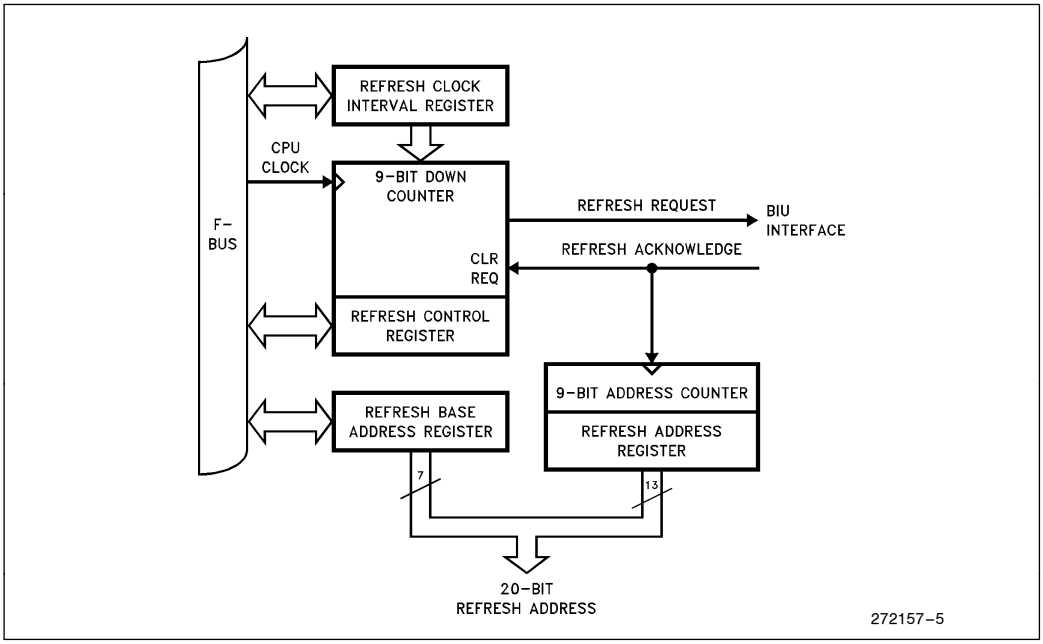


Figure 5. 80C186EA Refresh Control Unit Block Diagram

80C186 users refer to refresh bus cycles as "dummy read" cycles. They appear on the 80C186EA bus like ordinary read cycles except that no data is transferred. The states of $\overline{BHE}/\overline{RFSH}$ (Byte High Enable/Refresh) and A0 identify refresh cycles according to Table 5.

Table 5. Identification of 80C186EA/80C188EA Refresh Cycles

Data Bus Width	$\overline{BHE}/\overline{RFSH}$	A0
80C186EA	1	1
80C188EA	0	1

A.3 The Design Solution

Several vendors offer fully-integrated, single-chip DRAM controllers. Some of these devices are useful with the 80C186EA. However, most designers prefer to design their own DRAM controllers for the following reasons:

- Off-the-shelf DRAM controllers are relatively expensive, rivaling the cost of the 80C186EA.
- Off-the-shelf DRAM controllers may have relatively high pin counts and power needs. Such devices are more appropriate for larger DRAM arrays than those commonly used with the 80C186 family.
- Off-the-shelf DRAM controllers may not be a "best fit" because of slow operation or lack of flexibility for specific applications.
- Off-the-shelf DRAM controllers may offer special features such as nibble, page and static-column modes, which are not commonly used in 80C186 family designs. Therefore, the user may be paying for excess functionality.

DRAM controllers can be implemented using discrete logic, delay lines, programmable logic devices or some combination. Analog delay lines provide accurate timing relationships, but are inflexible and require additional discrete logic. Discrete logic alone takes excessive board space due to complex state logic and decoding.

A PLD solution (with or without additional discrete logic) is a good choice. PLDs have the following advantages:

- PLDs are available in speed selections to match the processor's speed.
- A wide variety of PLDs are available for optimal price versus performance.
- PLDs are ideal for a mixture of synchronous state logic and combinatorial decoding logic.
- PLDs are flexible and allow rapid design changes.

The next section presents a design solution using the Intel 85C060 CMOS Programmable Logic Device. The 85C060 contains all the decoding and state logic. 74AC157s multiplex the row and column addresses.

A.3.1 DESIGN OVERVIEW

Figure 6 is a block diagram of the DRAM controller and memory array. Address decoding within the controller selects the memory devices, so the user can readily change the base address (40000H as tested) or the array size (512 Kbytes as tested). Intel verified this design in the laboratory and incorporated it into the EV80C186EA/XL Evaluation Board. This board runs at 20 MHz, accessing Toshiba TC514256AP-70 DRAMs with zero wait states.

The 85C060 is a general purpose, 16-macrocell device. The 85C060 has several advantages over other programmable devices for this design. It is relatively high-speed, allowing the DRAM controller to work up to 20 MHz with the -15 (T_{PD} = 15 ns) version. It has eight product terms per macrocell. The 85C060 also has an asynchronous clocking option, which is very advantageous in this design.

The DRAM controller uses nine of the 85C060's 24 pins as inputs. It uses nine macrocells, with a maximum of five product terms per macrocell.

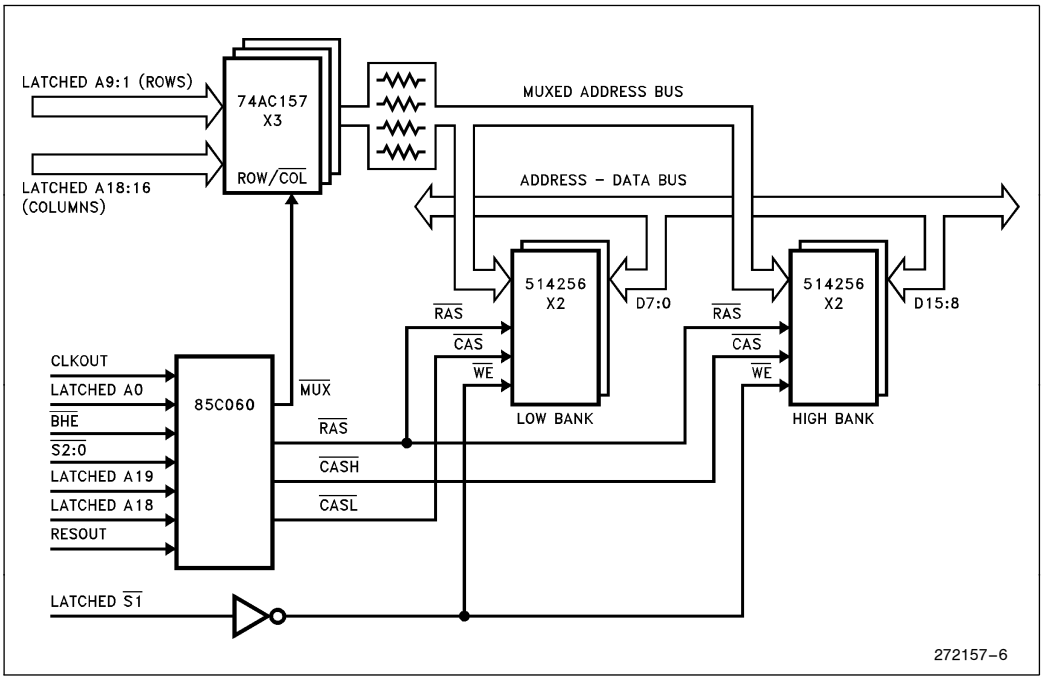


Figure 6. 80C186EA DRAM Memory System Block Diagram

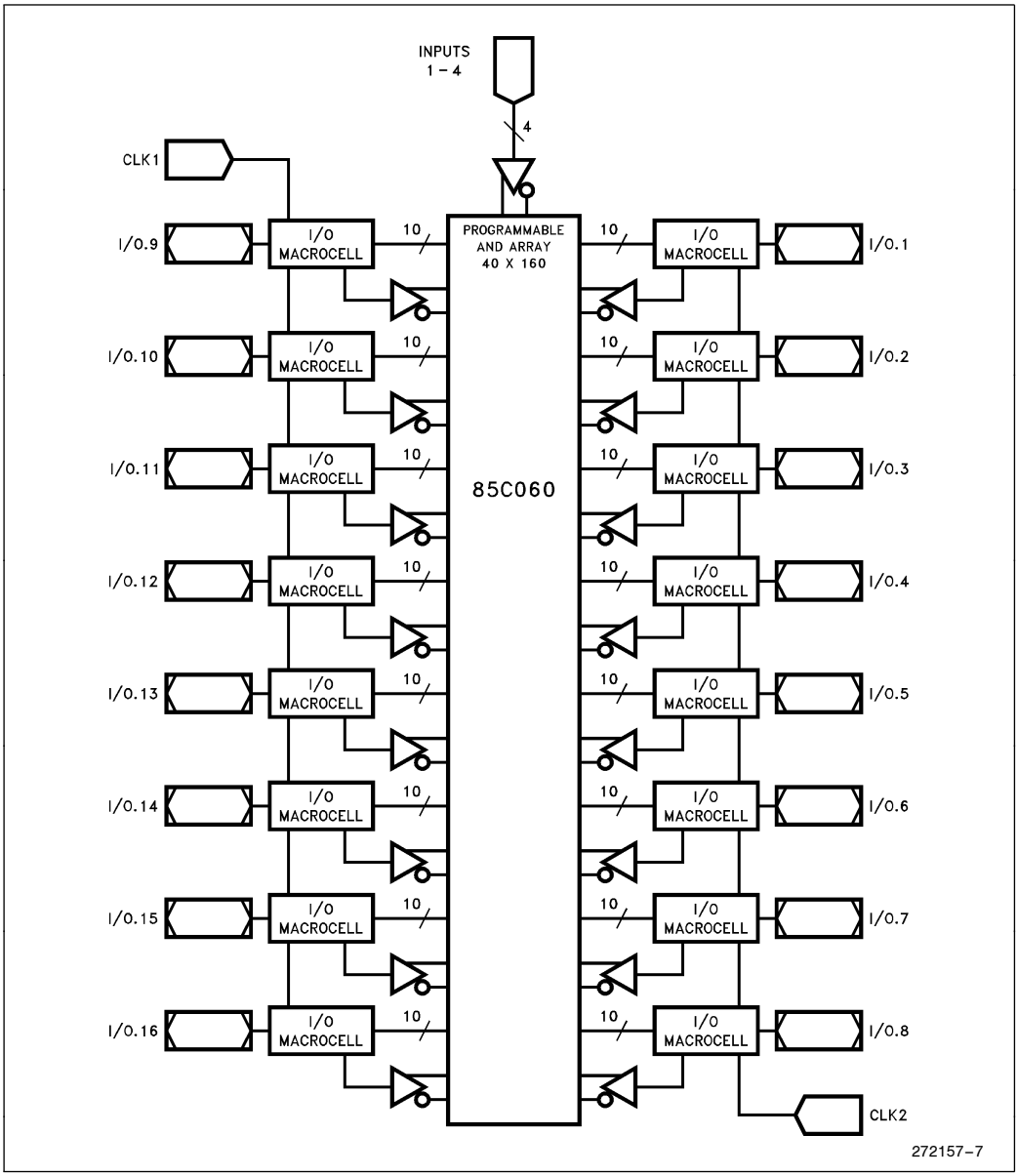


Figure 7. Intel 85C060 Architecture Diagram

A.3.2 HOW THE DESIGN SOLVES THE PROBLEM

Study Figure 1 once more. The 80C186EA's status signals $\overline{S2:0}$ can be conveniently decoded to detect the bus cycle beginning, type and end. However, notice that the status lines go active on a *rising* CLKOUT edge and inactive on a *falling* CLKOUT edge. $\overline{RAS}$ and $\overline{CAS}$ should both go active on *falling* clock edges and our controller should toggle between row and column addresses near the *rising* CLKOUT edge in between.

A good solution should somehow operate on both edges of the processor's CLKOUT signal and that is exactly what the 85C060 does. With the asynchronous clocking option, each 85C060 macrocell can derive its clock separately from the programmable AND array. The DRAM controller uses CLKOUT as an input. The 85C060 input buffers provide both true and complemented signals to the programmable array (see Figure 8) so each edge of CLKOUT can clock some of the macrocells. The maximum skew between the true and complemented signals is about 500 ps and can be neglected in a timing analysis.

The dual state machine aspect of the 85C060 makes the DRAM controller independent of wait states. It uses the 80C186EA's $\overline{S2:0}$ lines to sense the end of the bus cycle rather than just counting clocks. This dependence on the status lines does have a drawback. The 80C186EA's minimum specifications for T_{CHOV1} (Status Active) and T_{CLOV2} (Status Inactive) are both 3 ns. In asynchronous clock mode, the 85C060 has a minimum hold time T_{AH} of 6 ns. The solution is to avoid using status signals directly. A decoded status signal feeds back to the programmable AND array through a macrocell, adding enough delay time to meet the hold requirement.

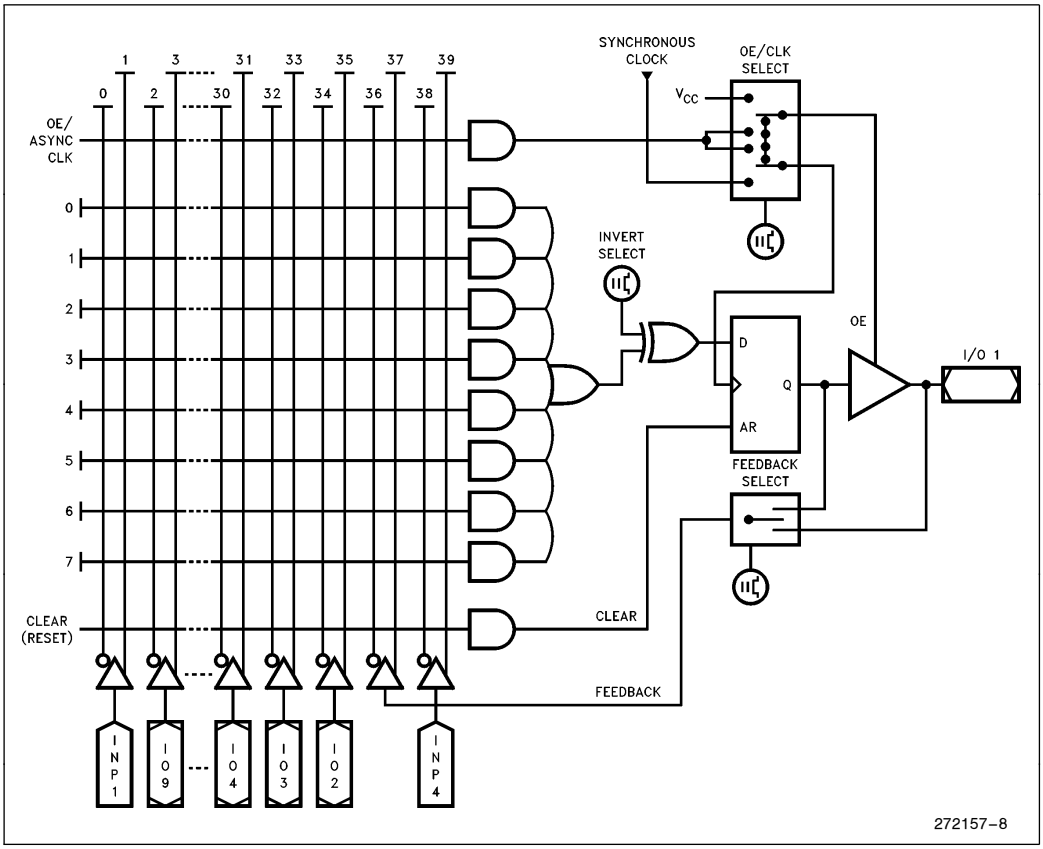


Figure 8. 85C060 Macrocell Architecture with D-Type Flip-Flop Configuration

A.3.3 HOW THE DRAM CONTROLLER WORKS

Figure 9 lists the complete source code for the DRAM controller. It consists mostly of two interconnected state machines. State Machine A and the address multiplexing signal ($\overline{MUX}$) are clocked on rising CLKOUT. State Machine B and the logic for $\overline{RAS}$ and $\overline{CAS}$ are clocked on falling CLKOUT.

```
Name                DRAMCON;
Partno              80C186EA/XL-xxx;
Revision            01;
Date                8/9/91;
Designer            LARRY BATES;
Company             Intel EMD Applications;
Assembly            28-Pin EPLD;
Location            Uxx;
Device              EP600;    /* Compilation switch */

/*************************************************************************/
/*  Target Device Type:  Intel 85C060-15                                 */
/*  File written for the CUPL PLD Design Program.                        */
/*  CUPL is a trademark of Logical Devices Inc.                          */
/*************************************************************************/

/** Inputs **/

pin 3 = nS0;            /* Status pins need pullups because the    */
pin 4 = nS1;            /* processor will not always drive them.   */
pin 5 = nS2;
pin 8 = LA19;
pin 9 = LA18;
pin 10 = UNUSED_IO;
pin 11 = UNUSED_INPUT;
pin 14 = LA0;
pin 7 = nBHE;
pin 23 = RESET;

/** Outputs **/

pin 6 = STATUS;
pin 15 = nCAS1;
pin 16 = nCAS0;
pin 22 = nMUX;
pin 17 = nRAS;
pin [18,19] = [A_CTR0..1];
pin [20,21] = [B_CTR0..1];

FIELD A_CTR = [A_CTR1, A_CTR0];

FIELD B_CTR = [B_CTR1, B_CTR0];

/** Other Pins **/

pin 1 = UNUSED_CLK1;
pin 2 = CLK;
pin 13 = UNUSED_CLK2;
```

272157–9

Figure 9. DRAM Controller PLD Source Code

```
/** Declarations and Intermediate Variable Definitions **/

$define A_0 'b'00
$define A_1 'b'01
$define A_2 'b'11
$define A_3 'b'10

$define B_0 'b'00
$define B_1 'b'01
$define B_2 'b'11
$define B_3 'b'10

/*************************************************************************/
/*  The powerup state of all the registers is LOW, which means the       */
/*  corresponding output pins are LOW.  RESET asynchronously clears      */
/*  the state machines.  The controller outputs correct themselves       */
/*  within a couple of clocks after RESET.                               */
/*************************************************************************/

/** Logic Equations **/

ADDRESS_SELECT =  (!LA19 & LA18) # (LA19 & !LA18);

REFRESH_ONLY =  (LA0 & nBHE);

LO_BYTE = !LA0;

HI_BYTE = !nBHE;

SEQUENCE A_CTR {

present A_0     if  STATUS          next A_1; /* ...T1 sample */
                if !STATUS          next A_0;

present A_1     if !ADDRESS_SELECT  next A_3; /* T2 sample */
                if  ADDRESS_SELECT  next A_2;

/*************************************************************************/
/*  Now control is turned over to B_CTR, which is clocked on falling     */
/*  edge, counts wait states and checks for end of cycle                 */
/*************************************************************************/

present A_2     if  B_CTR:B_2       next A_0; /* T3, T4 sample */
                default             next A_2;

present A_3     if  B_CTR:B_2       next A_0; /* T3, T4 sample */
                default             next A_3;
}

SEQUENCE B_CTR {

present B_0     if  (A_CTR:A_2 # A_CTR:A_3) next B_1; /* ...T3 sample */
                default             next B_0;

present B_1     if !STATUS          next B_2; /* Tw or T4 sample */
                if  STATUS          next B_1;

present B_2                         next B_0; /* End of Cycle */
```

272157-10

Figure 9. DRAM Controller PLD Source Code (Continued)

```
present B_3                          next B_0; /* Not used    */

}

STATUS =       !nS1 & !nS0 & nS2 #    /*  fetch            */
               !nS1 & nS0 & nS2 #     /*  read/rfsh        */
               nS1 & !nS0 & nS2;      /*  write            */
                                      /*  not I/O or INTA  */

                           /*  Makes it combinatorial. We are  */
                           /*  using feedback rather than      */
STATUS.OE =      'b'1;     /*  using status directly to give   */
                           /*  positive margin on hold time.   */

nRAS.D =         !((A_CTR:A_1 # A_CTR:A_2) & ADDRESS_SELECT & STATUS) ;
nRAS.CK =        !CLK;
nRAS.AR =        'b'0;

nMUX.D =         !(!nRAS & !REFRESH_ONLY);
nMUX.CK =        CLK;
nMUX.AR =        'b'0;

nCAS0.D =        !(!nMUX & LO_BYTE & STATUS);
nCAS0.CK =       !CLK;
nCAS0.AR =       'b'0;

nCAS1.D =        !(!nMUX & HI_BYTE & STATUS);
nCAS1.CK =       !CLK;
nCAS1.AR =       'b'0;

A_CTR0.CK =       CLK;
A_CTR0.AR =       RESET;

A_CTR1.CK =       CLK;
A_CTR1.AR =       RESET;

B_CTR0.CK =       !CLK;
B_CTR0.AR =       RESET;

B_CTR1.CK =       !CLK;
B_CTR1.AR =       RESET;

FUSE(6480,1);              /*  TURBO = ON  */
FUSE(6481,1);
```

272157-11

Figure 9. DRAM Controller PLD Source Code (Continued)

Refer to the state diagrams in Figure 10 for the following discussion. The startup conditions for State Machines A and B are A-0 and B-0 respectively. State Machine A initiates the DRAM controller sequence and State Machine B terminates the sequence.

The DRAM control sequence starts at the rising CLKOUT edge in bus state T_1 with Machine A in control. Machine A samples the bus status signals. If Machine A detects a memory read, write or refresh cycle, it proceeds from State A-0 to State A-1. Otherwise, it remains in A-0 and waits for the next bus cycle.

On the falling edge of T_2, the $\overline{RAS}$ logic samples the state of Machine A, the latched address lines and the bus status signals. If Machine A is in A-1 (memory read, write or refresh cycle) and the bus cycle is intended for the DRAMs, the 85C060 asserts $\overline{RAS}$.

On the rising edge of T_2, Machine A also samples the latched address lines. If the bus cycle is intended for the DRAMs, Machine A proceeds from A-1 to A-2. Otherwise, Machine A proceeds to A-3. Now, control passes to State Machine B. The $\overline{MUX}$ logic samples the state of $\overline{RAS}$ and the $\overline{BHE}$ pin. If $\overline{RAS}$ is active (indicating a DRAM access) and the bus cycle is not a refresh cycle, the 85C060 asserts $\overline{MUX}$. $\overline{MUX}$ toggles the 74AC157s between row and column addresses for DRAM reads and writes.

On the falling edge of T_3, Machine B samples the state of Machine A. If Machine A is in state A-2 (DRAM access) or state A-3 (memory read or write but not a DRAM access), Machine B proceeds from B-0 to B-1. If the bus cycle is a DRAM access, the 85C060 continues to assert $\overline{RAS}$. The $\overline{CAS}$ logic samples the state of $\overline{MUX}$, latched address A0, $\overline{BHE}$ and bus cycle status. If $\overline{MUX}$ is active (indicating a DRAM read or write) and processor accesses a low byte, the 85C060 asserts $\overline{CAS0}$. If $\overline{MUX}$ is active and the processor accesses a high byte, the 85C060 asserts $\overline{CAS1}$. The difference between a DRAM read access and a DRAM refresh access is that $\overline{MUX}$ and $\overline{CAS1:0}$ are not necessary for the refresh.

On the rising edge of T_3, Machine A waits for Machine B to terminate the sequence. If $\overline{MUX}$ is active (DRAM read or write), it remains active.

On the next falling edge, Machine B samples the bus status signals. If status is still active, this state is a wait state, T_W, and Machine B remains in B-1. If this state is a wait state and $\overline{RAS}$ is active (DRAM access), $\overline{RAS}$ remains active. If this state is a wait state and $\overline{CAS1:0}$ are active (DRAM read or write), $\overline{CAS1:0}$ remain active.

On the rising edge of a wait state, Machine A continues waiting for Machine B to teminate the sequence. If $\overline{MUX}$ is active (DRAM read or write), it remains active as in a T_3 state.

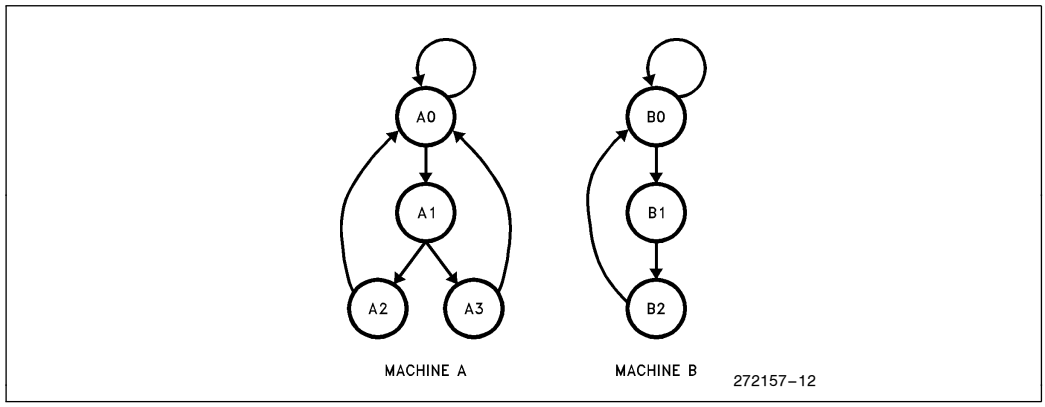


Figure 10. DRAM Controller State Machines

If status is inactive, this state is a T_4 state and Machine B proceeds from B-1 to B-2. If this state is a T_4 state and $\overline{RAS}$ is active (DRAM access), the $\overline{RAS}$ logic also detects inactive status and the 85C060 deasserts $\overline{RAS}$. If this state is a T_4 state and $\overline{CAS1:0}$ are active (DRAM read or write), the $\overline{CAS}$ logic also samples the bus status signals. If status is inactive, the 85C060 deasserts $\overline{CAS1:0}$.

On the rising edge of T_4, Machine A samples the state of Machine B. With Machine B in state B-2 (ready to terminate the sequence), Machine A proceeds from either A-2 (DRAM access) or A-3 (memory read or write but not DRAM access) to A-0. If $\overline{MUX}$ is active, the $\overline{MUX}$ logic examines the state of $\overline{RAS}$. If $\overline{RAS}$ is inactive (indicating an ending cycle), the 85C060 deasserts $\overline{MUX}$.

On the next falling CLKOUT edge, Machine B unconditionally proceeds from state B-2 to B-0, terminating the DRAM sequence. Control passes to State Machine A.

A.3.4 PROGRAMMING THE 80C186EA REFRESH CONTROLLER

The programming value for the refresh clock interval register is easy to calculate. The 514256 DRAM data sheet specifies 512 refresh cycles (2^9 = 512 rows) per 8 ms. Multiplying 0.008 by 20 * 10^6 (for 20 MHz operation), then dividing by 512, yields 312 clocks. To compensate for bus latency for the RCU to gain control over the bus, derate the actual programming value by five percent to about 297.

Program the refresh base address register with the upper seven address bits which specify the location of the DRAM array. Finally, activate the refresh control unit by setting the REN (Refresh Enable) bit in the refresh control register.

Use of the 80C186EA power-save mode requires that you reprogram the device. Divide the value in the refresh clock interval register by the desired clock divisor value before writing to the power-save register.

A.3.5 LOADING AND TRANSMISSION LINE EFFECTS

In the example DRAM system, there are only four DRAM devices so the capacitive load can generally be ignored. In larger systems, with address, data and control lines connected to numerous DRAMs, the total capacitive load can degrade performance. Keep in mind that the controller drives a particularly heavy load during refresh, when it drives all DRAMs simultaneously.

The inductance of long printed circuit board traces makes DRAM subsystems prone to ringing and overshoot/undershoot. Series damping resistors offer relief by increasing the rise and fall times of the signals and thus the instantaneous change in current. The example circuit contains 22Ω resistors in address and control driven by the 85C060 and 74AC157 devices. Determine optimum values for your system empirically.

A.3.6 MODIFICATIONS FOR THE 80C188EA

It is easy to modify the DRAM array for an 80C188EA. Place all the DRAM devices on the low data bus D7:0. Use A0 as a true address bit (instead of low byte select) by connecting latched A0, instead of latched A18, to the 74AC157. Since A0 is fixed as a logic one during 80C188EA refresh cycles (see Table 5), it becomes a column address line, not a row address line.

In the 85C060 logic equations, rename nBHE to nRFSH and change REFRESH_ONLY to REFRESH_ONLY = !nRFSH to detect refresh bus cycles. Change the definition of LO_BYTE to LO_BYTE = !LA19 & LA18 and change HI_BYTE to HI_BYTE = LA19 & !LA18. This arrangement maps 256 Kbytes (two DRAM devices) at 40000H–7FFFFH and 256 Kbytes (the other two DRAM devices) at 80000H–B7FFFH. The 85C060 asserts $\overline{RAS}$ to all the DRAMs on every DRAM access, but $\overline{CAS}$ is active only for the two DRAMs being read or written. The bus cycles are redundant refreshes at the other two DRAMs.

A.4 Adapting the Design to Your System

With DRAM more reasonably priced now, chances are you will consider DRAMs for your system. The example DRAM controller leaves one input, one I/O and two clock pins unused on the 85C060. With minor PLD code changes, you can use these pins to control additional DRAM banks, provide a bus ready output or provide a DMA Acknowledge signal.

The 85C060-based DRAM controller is a general example of the synchronous "glue logic" described in Section 4. Macrocells can derive individual output enables and asynchronous resets from the array of AND gates. In addition, the registers may be configured as D-, T-, JK- or SR-type flip-flops. This allows you to apply the same general bus-tracking technique to a variety of 80C186EA control logic.

The 85C060 uses Intel's CHMOS EPROM technology, making it a suitable low-power companion to the 80C186EA. It is available in both 24-pin DIP and 28-lead PLCC packages. If you need more functions than will fit inside the 85C060, consider the 85C090, a similar device with 24 macrocells.

A.5 Bibliography

1. Advanced Micro Devices. *PAL Device Data Book*, Sunnyvale, CA, 1990.
2. Advanced Micro Devices. *PAL Device Handbook*, Sunnyvale, CA, 1988.
3. Mion, Garry. "The 80C186/80C188 Integrated Refresh Control Unit", Intel Corporation Ap Brief AB-31, Order #270520, Santa Clara CA, 1990.
4. Slater, Michael. *Microprocessor-Based Design*, Mountain View, CA: Mayfield, 1987.
5. Waugh, Thomas. "Programmable Array Serves as a Controller for Dynamic RAMs", *EDN*, February 18, 1988, pp. 169–176.