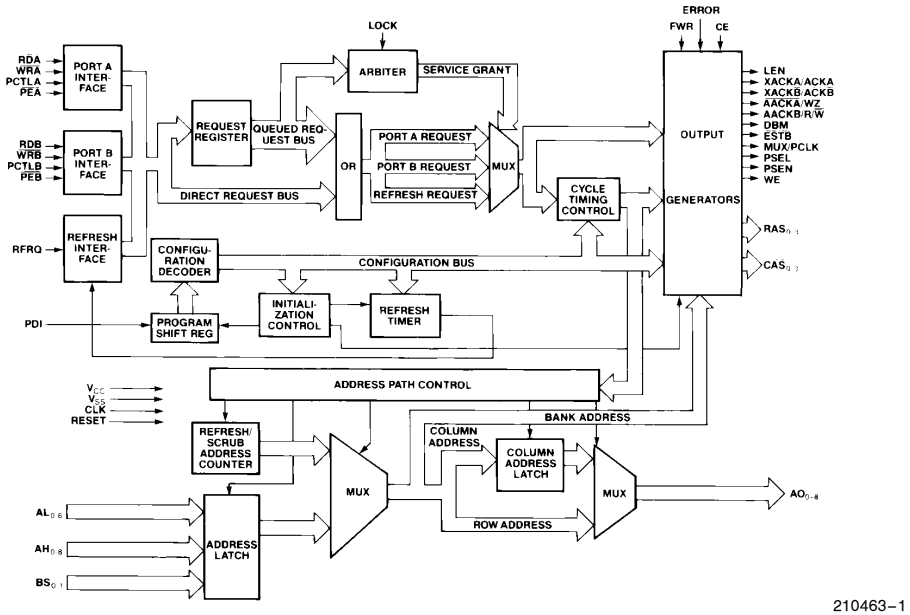
8207
DUAL-PORT DYNAMIC RAM CONTROLLER

- **Provides All Signals Necessary to Control 16K, 64K and 256K Dynamic RAMs**
- **Directly Addresses and Drives up to 2 Megabytes without External Drivers**
- **Supports Single and Dual-Port Configurations**
- **Automatic RAM Initialization in All Modes**
- **Four Programmable Refresh Modes**
- **Transparent Memory Scrubbing in ECC Mode**
- **Fast Cycle Support for 8 MHz 80286 with 8207-16**
- **Slow Cycle Support for 8 MHz, 10 MHz 8086/88, 80186/188 with 8207-8, 8207-10**
- **Provides Signals to Directly Control the 8206 Error Detection and Correction Unit**
- **Supports Synchronous or Asynchronous Operation on Either Port**
- **68 Lead JEDEC Type A Leadless Chip Carrier (LCC) and Pin Grid Array (PGA), Both in Ceramic.**

The Intel 8207 Dual-Port Dynamic RAM Controller is a high-performance, systems-oriented, Dynamic RAM controller that is designed to easily interface 16K, 64K and 256K Dynamic RAMs to Intel and other microprocessor systems. A dual-port interface allows two different busses to independently access memory. When configured with an 8206 Error Detection and Correction Unit the 8207 supplies the necessary logic for designing large error-corrected memory arrays. This combination provides automatic memory initialization and transparent memory error scrubbing.



210463–1

Figure 1. 8207 Block Diagram

Table 1. Pin Description

Symbol	Pin	Type	Name and Function
LEN	1	O	**ADDRESS LATCH ENABLE:** In two-port configurations, when Port A is running with iAPX 286 Status interface mode, this output replaces the ALE signal from the system bus controller of port A and generates an address latch enable signal which provides optimum setup and hold timing for the 8207. This signal is used in Fast Cycle operation only.
$\overline{\text{XACKA}}$/ $\overline{\text{ACKA}}$	2	O	**TRANSFER ACKNOWLEDGE PORT A/ACKNOWLEDGE PORT A:** In non-ECC mode, this pin is $\overline{\text{XACKA}}$ and indicates that data on the bus is valid during a read cycle or that data may be removed from the bus during a write cycle for Port A. $\overline{\text{XACKA}}$ is a Multibus-compatible signal. In ECC mode, this pin is $\overline{\text{ACKA}}$ which can be configured, depending on the programming of the X program bit, as an $\overline{\text{XACK}}$ or $\overline{\text{AACK}}$ strobe. The SA programming bit determines whether the $\overline{\text{AACK}}$ will be an early $\overline{\text{EAACKA}}$ or a late $\overline{\text{LAACKA}}$ interface signal.
$\overline{\text{XACKB}}$/ $\overline{\text{ACKB}}$	3	O	**TRANSFER ACKNOWLEDGE PORT B/ACKNOWLEDGE PORT B:** In non-ECC mode, this pin is XACKB and indicates that data on the bus is valid during a read cycle or that data may be removed from the bus during a write cycle for Port B. $\overline{\text{XACKB}}$ is a Multibus-compatible signal. In ECC mode, this pin is $\overline{\text{ACKB}}$ which can be configured, depending on the programming of the X program bit, as an $\overline{\text{XACK}}$ or $\overline{\text{AACK}}$ strobe. The SB programming bit determines whether the $\overline{\text{AACK}}$ will be an early $\overline{\text{EAACKB}}$ or a late $\overline{\text{LAACKB}}$ interface signal.
$\overline{\text{AACKA}}$/ $\overline{\text{WZ}}$	4	O	**ADVANCED ACKNOWLEDGE PORT A/WRITE ZERO:** In non-ECC mode, this pin is $\overline{\text{AACKA}}$ and indicates that the processor may continue processing and that data will be available when required. This signal is optimized for the system by programming the SA program bit for synchronous or asynchronous operation. In ECC mode, after a RESET, this signal will cause the 8206 to force the data to all zeros and generate the appropriate check bits.
$\overline{\text{AACKB}}$/ R/$\overline{\text{W}}$	5	O	**ADVANCED ACKNOWLEDGE PORT B/READ/WRITE:** In non-ECC mode, this pin is $\overline{\text{AACKB}}$ and indicates that the processor may continue processing and that data will be available when required. This signal is optimized for the system by programming the SB program bit for synchronous or asynchronous operation. In ECC mode, this signal causes the 8206 EDCU to latch the syndrome and error flags and generate check bits.
$\overline{\text{DBM}}$	6	O	**DISABLE BYTE MARKS:** This is an ECC control output signal indicating that a read or refresh cycle is occurring. This output forces the byte address decoding logic to enable all 8206 data output buffers. In ECC mode, this output is also asserted during memory initialization and the 8-cycle dynamic RAM wake-up exercise. In non-ECC systems this signal indicates that either a read, refresh or 8-cycle warm-up is in progress.
$\overline{\text{ESTB}}$	7	O	**ERROR STROBE:** In ECC mode, this strobe is activated when an error is detected and allows a negative-edge triggered flip-flop to latch the status of the 8206 EDCU CE for systems with error logging capabilities. $\overline{\text{ESTB}}$ will not be issued during refresh cycles.
LOCK	8	I	**LOCK:** This input instructs the 8207 to lock out the port not being serviced at the time LOCK was issued.
V_{CC}	9 43	I	**DRIVER POWER:** +5 volts. Supplies V_{CC} for the output drivers. **LOGIC POWER:** +5 volts. Supplies V_{CC} for the internal logic circuits.
CE	10	I	**CORRECTABLE ERROR:** This is an ECC input from the 8206 EDCU which instructs the 8207 whether a detected error is correctable or not. A high input indicates a correctable error. A low input inhibits the 8207 from activating WE to write the data back into RAM. This should be connected to the CE output of the 8206.

Table 1. Pin Description (Continued)

Symbol	Pin	Type	Name and Function
$\overline{\text{ERROR}}$	11	I	**ERROR:** This is an ECC input from the 8206 EDCU and instructs the 8207 that an error was detected. This pin should be connected to the ERROR output of the 8206.
MUX/ PCLK	12	O	**MULTIPLEXER CONTROL/PROGRAMMING CLOCK:** Immediately after a RESET this pin is used to clock serial programming data into the PDI pin. In normal two-port operation, this pin is used to select memory addresses from the appropriate port. When this signal is high, port A is selected and when it is low, port B is selected. This signal may change state before the completion of a RAM cycle, but the RAM address hold time is satisfied.
PSEL	13	O	**PORT SELECT:** This signal is used to select the appropriate port for data transfer. When this signal is high port A is selected and when it is low port B is selected.
PSEN	14	O	**PORT SELECT ENABLE:** This signal used in conjunction with PSEL provides contention-free port exchange on the data bus. When PSEN is low, port selection is allowed to change state.
WE	15	O	**WRITE ENABLE:** This signal provides the dynamic RAM array the write enable input for a write operation.
$\overline{\text{FWR}}$	16	I	**FULL WRITE:** This is an ECC input signal that instructs the 8207, in an ECC configuration, whether the present write cycle is normal RAM write (full write) or a RAM partial write (read-modify-write) cycle.
RESET	17	I	**RESET:** This signal causes all internal counters and state flip-flops to be reset and upon release of RESET, data appearing at the PDI pin is clocked in by the PCLK output. The states of the PDI, PCTLA, PCTLB and RFRQ pins are sampled by RESET going inactive and are used to program the 8207. An 8-cycle dynamic RAM warm-up is performed after clocking PDI bits into the 8207.
$\overline{\text{CAS0}}$–$\overline{\text{CAS3}}$	18–21	O	**COLUMN ADDRESS STROBE:** These outputs are used by the dynamic RAM array to latch the column address, present on the AO0–8 pins. These outputs are selected by the BS0 and BS1 as programmed by program bits RB0 and RB1. These outputs drive the dynamic RAM array directly and need no external drivers.
$\overline{\text{RAS0}}$–$\overline{\text{RAS3}}$	22–25	O	**ROW ADDRESS STROBE:** These outputs are used by the dynamic RAM array to latch the row address, present on the AO0–8 pins. These outputs are selected by the BS0 and BS1 as programmed by program bits RB0 and RB1. These outputs drive the dynamic RAM array directly and need no external drivers.
V_{SS}	26 60	I I	**DRIVER GROUND:** Provides a ground for the output drivers. **LOGIC GROUND:** Provides a ground for the remainder of the device.
AO0–AO8	35–27	O	**ADDRESS OUTPUTS:** These outputs are designed to provide the row and column addresses of the selected port to the dynamic RAM array. These outputs drive the dynamic RAM array directly and need no external drivers.
BS0–BS1	36–37	I	**BANK SELECT:** These inputs are used to select one of four banks of the dynamic RAM array as defined by the program bits RB0 and RB1.
AL0–AL8	38–42 44–47	I	**ADDRESS LOW:** These lower-order address inputs are used to generate the row address for the internal address multiplexer.
AH0–AH8	48–56	I	**ADDRESS HIGH:** These higher-order address inputs are used to generate the column address for the internal address multiplexer.

Table 1. Pin Description (Continued)

Symbol	Pin	Type	Name and Function
PDI	57	I	**PROGRAM DATA INPUT:** This input programs the various user-selectable options in the 8207. The PCLK pin shifts programming data into the PDI input from optional external shift registers. This pin may be strapped high or low to a default ECC (PDI = Logic ''1'') or non-ECC (PDI = Logic ''0'') mode configuration.
RFRQ	58	I	**REFRESH REQUEST:** This input is sampled on the falling edge of RESET. If it is high at RESET, then the 8207 is programmed for internal refresh request or external refresh request with failsafe protection. If it is low at RESET, then the 8207 is programmed for external refresh without failsafe protection or burst refresh. Once programmed the RFRQ pin accepts signals to start an external refresh with failsafe protection or external refresh without failsafe protection or a burst refresh.
CLK	59	I	**CLOCK:** This input provides the basic timing for sequencing the internal logic.
$\overline{\text{RDB}}$	61	I	**READ FOR PORT B:** This pin is the read memory request command input for port B. This input also directly accepts the $\overline{\text{S1}}$ status line from Intel processors.
$\overline{\text{WRB}}$	62	I	**WRITE FOR PORT B:** This pin is the write memory request command input for port B. This input also directly accepts the $\overline{\text{S0}}$ status line from Intel processors.
$\overline{\text{PEB}}$	63	I	**PORT ENABLE FOR PORT B:** This pin serves to enable a RAM cycle request for port B. It is generally decoded from the port address.
PCTLB	64	I	**PORT CONTROL FOR PORT B:** This pin is sampled on the falling edge of RESET. If low after RESET, the 8207 is programmed to accept memory read and write commands, Multibus commands or iAPX 286 status inputs. If high after RESET, the 8207 is programmed to accept status inputs from iAPX 86 or iAPX 186 processors. The $\overline{\text{S2}}$ status line should be connected to this input if programmed to accept iAPX 86 or iAPX 186 status inputs. When programmed to accept commands or iAPX 286 status, it should be tied low or it may be used as a Multibus-compatible inhibit signal.
$\overline{\text{RDA}}$	65	I	**READ FOR PORT A:** This pin is the read memory request command input for port A. This input also directly accepts the $\overline{\text{S1}}$ status line from Intel processors.
$\overline{\text{WRA}}$	66	I	**WRITE FOR PORT A:** This pin is the write memory request command input for port A. This input also directly accepts the $\overline{\text{S0}}$ status line from Intel processors.
$\overline{\text{PEA}}$	67	I	**PORT ENABLE FOR PORT A:** This pin serves to enable a RAM cycle request for port A. It is generally decoded from the port address.
PCTLA	68	I	**PORT CONTROL FOR PORT A:** This pin is sampled on the falling edge of RESET. If low after RESET, the 8207 is programmed to accept memory read and write commands, Multibus commands or iAPX 286 status inputs. If high after RESET, the 8207 is programmed to accept status inputs from iAPX 86 or iAPX 186 processors. The $\overline{\text{S2}}$ status line should be connected to this input if programmed to accept iAPX 86 or iAPX 186 status inputs. When programmed to accept commands or iAPX 286 status, it should be tied low or it may be connected to INHIBIT when operating with Multibus.

GENERAL DESCRIPTION

The Intel 8207 Dual-Port Dynamic RAM Controller is a microcomputer peripheral device which provides the necessary signals to address, refresh and directly drive 16K, 64K and 256K dynamic RAMs. This controller also provides the necessary arbitration circuitry to support dual-port access of the dynamic RAM array.

The 8207 supports several microprocessor interface options including synchronous and asynchronous connection to iAPX 86, iAPX 88, iAPX 186, iAPX 188, iAPX 286 and Multibus.

This device may be used with the 8206 Error Detection and Correction Unit (EDCU). When used with the 8206, the 8207 is programmed in the Error Checking and Correction (ECC) mode. In this mode, the 8207 provides all the necessary control signals for the 8206 to perform memory initialization and transparent error scrubbing during refresh.

FUNCTIONAL DESCRIPTION

Processor Interface

The 8207 has control circuitry for two ports each capable of supporting one of several possible bus structures. The ports are independently configurable allowing the dynamic RAM to serve as an interface between two different bus structures.

Each port of the 8207 may be programmed to run synchronous or asynchronous to the processor clock. (See Synchronous/Asynchronous Mode.) The 8207 has been optimized to run synchronously with Intel's iAPX 86, iAPX 88, iAPX 186, iAPX 188, and iAPX 286. When the 8207 is programmed to run in asynchronous mode, the 8207 inserts the necessary synchronization circuitry for the $\overline{RD}$, $\overline{WR}$, $\overline{PE}$, and PCTL inputs.

The 8207 achieves high performance (i.e., no wait states) by decoding the status lines directly from the iAPX 86, iAPX 88, iAPX 186, iAPX 188 and iAPX 286 processors. The 8207 can also be programmed to receive read or write Multibus commands or commands from a bus controller. (See Status/Command Mode.)

The 8207 may be programmed to accept the clock of the iAPX 86, 88, 186, 188 or 286. The 8207 adjusts its internal timing to allow for the different clock frequencies of these microprocessors. (See Microprocessor Clock Frequency Option.)

Figures 2A and 2B show the different processor interfaces to the 8207 using the synchronous or asynchronous mode and status or command interface.

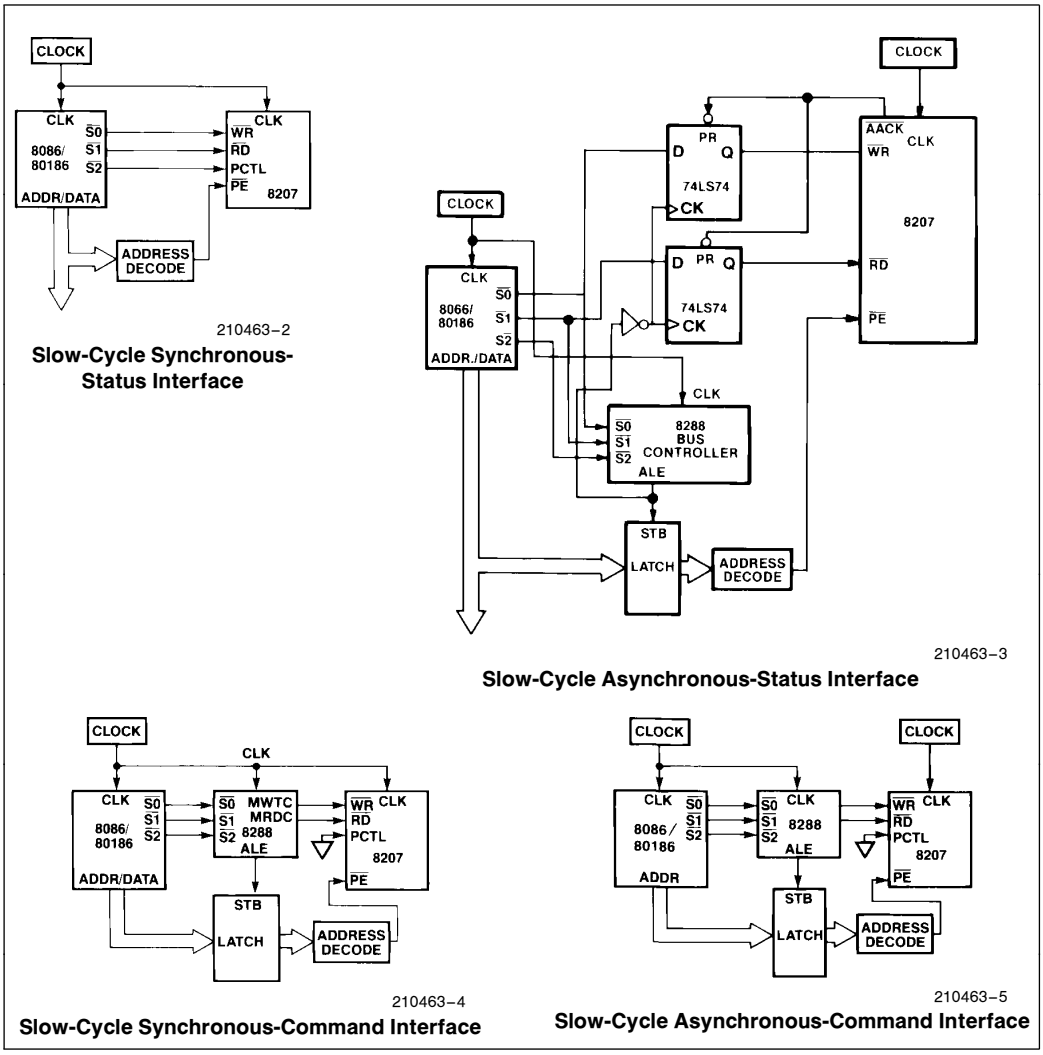


Figure 2A. Slow-Cycle (CFS = 0) Port Interfaces Supported by the 8207

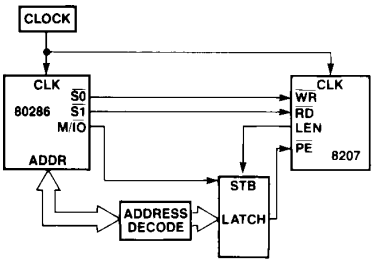
Single-Port Operation

The use of an address latch with the iAPX 286 status interface is not needed since the 8207 can internally latch the addresses with an internal signal similar in behavior to the LEN output. This operation is active only in single-port applications when the processor is interfaced to port A.

Dual-Port Operation

The 8207 provides for two-port operation. Two independent processors may access memory controlled by the 8207. The 8207 arbitrates between each of the processor requests and directs data to or from the appropriate port. Selection is done on a priority concept that reassigns priorities based upon past history. Processor requests are internally queued.

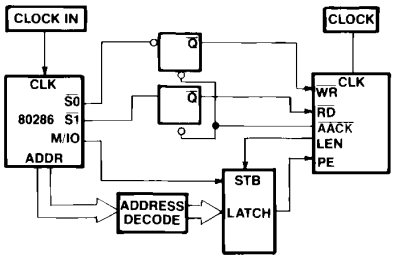
Figure 3 shows a dual-port configuration with two iAPX 86 systems interfacing to dynamic RAM. One of the processor systems is interfaced synchronously using the status interface and the other is interfaced asynchronously also using the status interface.



210463–6

NOTE:
Address latch not required in single-port mode.

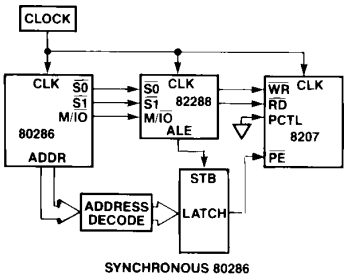
Fast-Cycle Synchronous-Status Interface



210463–7

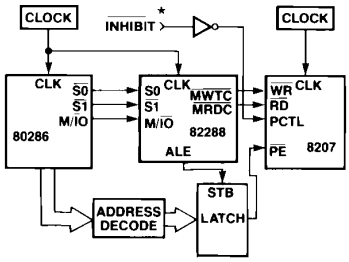
NOTE:
Address latch not required in single-port mode.

Fast-Cycle Asynchronous-Status Interface



210463–8

Fast-Cycle Synchronous-Command Interface



210463–9

*MULTI-BUS Option

Fast-Cycle Asynchronous-Command Interface

Figure 2B. Fast-Cycle (CFS = 1) Port Interfaces Supported by the 8207

Dynamic RAM Interface

The 8207 is capable of addressing 16K, 64K and 256K dynamic RAMs. Figure 4 shows the connection of the processor address bus to the 8207 using the different RAMs.

The 8207 divides memory into as many as four banks, each bank having its own Row ($\overline{RAS}$) and Column ($\overline{CAS}$) Address Strobe pair. This organization permits RAM cycle interleaving and permits error scrubbing during ECC refresh cycles. RAM cycle interleaving overlaps the start of the next RAM cycle with the RAM Precharge period of the previous cycle. Hiding the precharge period of one RAM cycle behind the data access period of the next RAM cycle optimizes memory bandwidth and is effective as long as successive RAM cycles occur in alternate banks.

Successive data access to the same bank will cause the 8207 to wait for the precharge time of the previous RAM cycle.

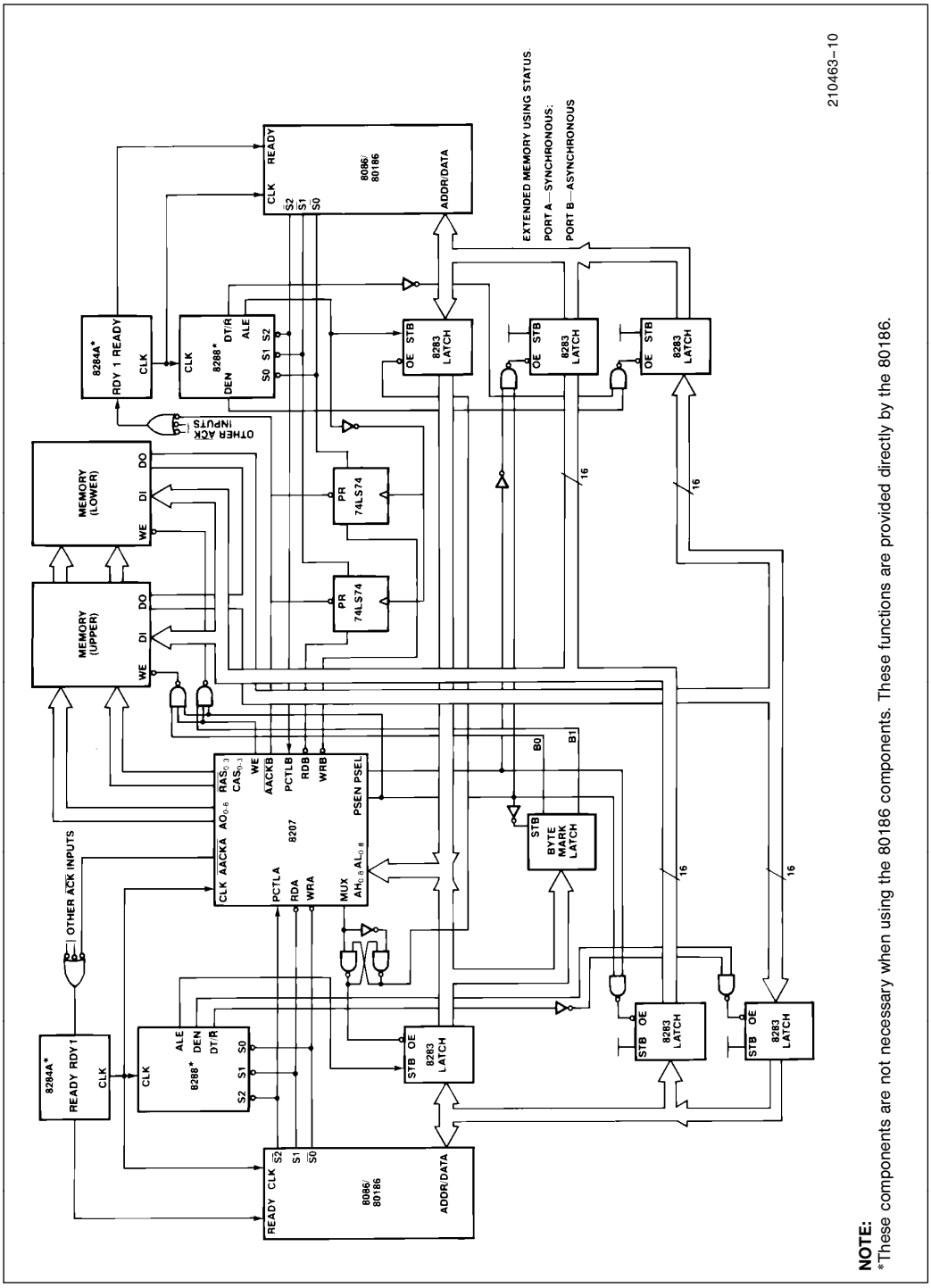
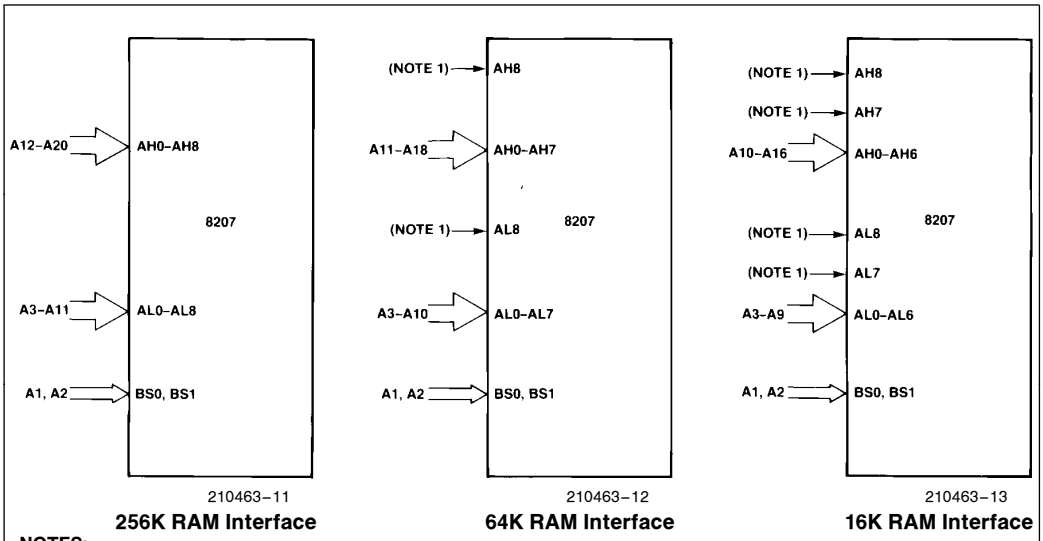


Figure 3. 8086/80186 Dual Port System

8



NOTES:
1. Unassigned address input pins should be strapped high or low.
2. A0 along with BHE are used to select a byte within a processor word.
3. Low order address bits are used as bank select inupts so that consecutive memory access requests are to alternate banks allowing bank interleaving of memory cycles.

Figure 4. Processor Address Interface to the 8207 Using 16K, 64K, and 256K RAMs

If not all RAM banks are occupied, the 8207 reassigns the $\overline{RAS}$ and $\overline{CAS}$ strobes to allow using wider data words without increasing the loading on the $\overline{RAS}$ and $\overline{CAS}$ drivers. Table 2 shows the bank selection decoding and the word expansion, including $\overline{RAS}$ and $\overline{CAS}$ assignments. For example, if only two RAM banks are occupied, then two $\overline{RAS}$ and two $\overline{CAS}$ strobes are activated per bank. Program bits RB1 and RB0 are not used to check the bank select inputs BS1 and BS0. The system design must protect from accesses to ''illegal'', non-existent banks of memory, by deactivating the PEA, PEB inputs when addressing an illegal bank.

The 8207 can interface to fast or slow RAMs. The 8207 adjusts and optimizes internal timings for either the fast or slow RAMs as programmed. (See RAM Speed Option.)

Memory Initialization

After programming, the 8207 performs eight RAM ''warm-up'' cycles to prepare the dynamic RAM for proper device operation. During ''warm-up'' some RAM parameters, such as tRAH, tASC, may not be met. This causes no harm to the dynamic RAM array. If configured for operation with error correction, the 8207 and 8206 EDCU will proceed to initialize all of memory (memory is written with zeros with corresponding check bits).

Table 2. Bank Selection Decoding and Word Expansion

Program Bits		Bank Input		RAS/CAS Pair Allocation
RB1	RB0	BS1	BS0	
0	0	0	0	RAS_{0-3}, CAS_{0-3} to Bank 0
0	0	0	1	Illegal
0	0	1	0	Illegal
0	0	1	1	Illegal
0	1	0	0	$RAS_{0,1}$, $CAS_{0,1}$ to Bank 0
0	1	0	1	$RAS_{2,3}$, $CAS_{2,3}$ to Bank 1
0	1	1	0	Illegal
0	1	1	1	Illegal
1	0	0	0	RAS_0, CAS_0 to Bank 0
1	0	0	1	RAS_1, CAS_1 to Bank 1
1	0	1	0	RAS_2, CAS_2 to Bank 2
1	0	1	1	Illegal
1	1	0	0	RAS_0, CAS_0 to Bank 0
1	1	0	1	RAS_1, CAS_1 to Bank 1
1	1	1	0	RAS_2, CAS_2 to Bank 2
1	1	1	1	RAS_3, CAS_3 to Bank 3

Because the time to initialize memory is fairly long, the 8207 may be programmed to skip initialization in ECC mode. The time required to initialize all of memory is dependent on the clock cycle time to the 8207 and can be calculated by the following equation:

$$T_{INIT} = (2^{23})\ T_{CLCL} \quad (1)$$

if $T_{CLCL} = 125$ ns then $T_{INIT} \approx 1$ sec.

8206 ECC Interface

For operation with Error Checking and Correction (ECC), the 8207 adjusts its internal timing and changes some pin functions to optimize performance and provide a clean dual-port memory interface between the 8206 EDCU and memory. The 8207 directly supports a master-only (16-bit word plus 6 check bits) system. Under extended operation and reduced clock frequency, the 8207 will support any ECC master-slave configuration up to 80 data bits, which is the maximum set by the 8206 EDCU. (See Extend Option.)

Correctable errors detected during memory read cycles are corrected immediately and then written back into memory.

In a synchronous bus environment, ECC system performance has been optimized to enhance processor throughput, while in an asynchronous bus environment (the Multibus), ECC performance has been optimized to get valid data onto the bus as quickly as possible. Performance optimization, processor throughput or quick data access may be selected via the Transfer Acknowledge Option.

The main difference between the two ECC implementations is that, when optimized for processor throughput, RAM data is always corrected and an advanced transfer acknowledge is issued at a point when, by knowing the processor characteristics, data is guaranteed to be valid by the time the processor needs it.

When optimized for quick data access, (valid for Multibus) the 8206 is configured in the uncorrecting mode where the delay associated with error correction circuitry is transparent, and a transfer acknowledge is issued as soon as valid data is known to exist. If the $\overline{\text{ERROR}}$ flag is activated, then the transfer acknowledge is delayed until after the 8207 has instructed the 8206 to correct the data and the corrected data becomes available on the bus. Figure 5 illustrates a dual-port ECC system.

Figure 6 illustrates the interface required to drive the $\overline{\text{CRCT}}$ pin of the 8206, in the case that one port (PORT A) receives an advanced acknowledge (not Multibus-compatible), while the other port (PORT B) receives $\overline{\text{XACK}}$ (which is Multibus-compatible).

Error Scrubbing

The 8207/8206 performs error correction during refresh cycles (error scrubbing). Since the 8207 must refresh RAM, performing error scrubbing during refresh allows it to be accomplished without additional performance penalties.

Upon detection of a correctable error during refresh, the RAM refresh cycle is lengthened slightly to permit the 8206 to correct the error and for the corrected word to be rewritten into memory. Uncorrectable errors detected during scrubbing are ignored.

Refresh

The 8207 provides an internal refresh interval counter and a refresh address counter to allow the 8207 to refresh memory. The 8207 will refresh 128 rows every 2 milliseconds or 256 rows every 4 milliseconds, which allows all RAM refresh options to be supported. In addition, there exists the ability to refresh 256 row address locations every 2 milliseconds via the Refresh Period programming option.

The 8207 may be programmed for any of four different refresh options: Internal refresh only, External refresh with failsafe protection, External refresh without failsafe protection, Burst Refresh mode, or no refresh. (See Refresh Options.)

It is possible to decrease the refresh time interval by 10%, 20% or 30%. This option allows the 8207 to compensate for reduced clock frequencies. Note that an additional 5% interval shortening is built-in in all refresh interval options to compensate for clock variations and non-immediate response to the internally generated refresh request. (See Refresh Period Options.)

External Refresh Requests after RESET

External refresh requests are not recognized by the 8207 until after it is finished programming and preparing memory for access. Memory preparation includes 8 RAM cycles to prepare and ensure proper

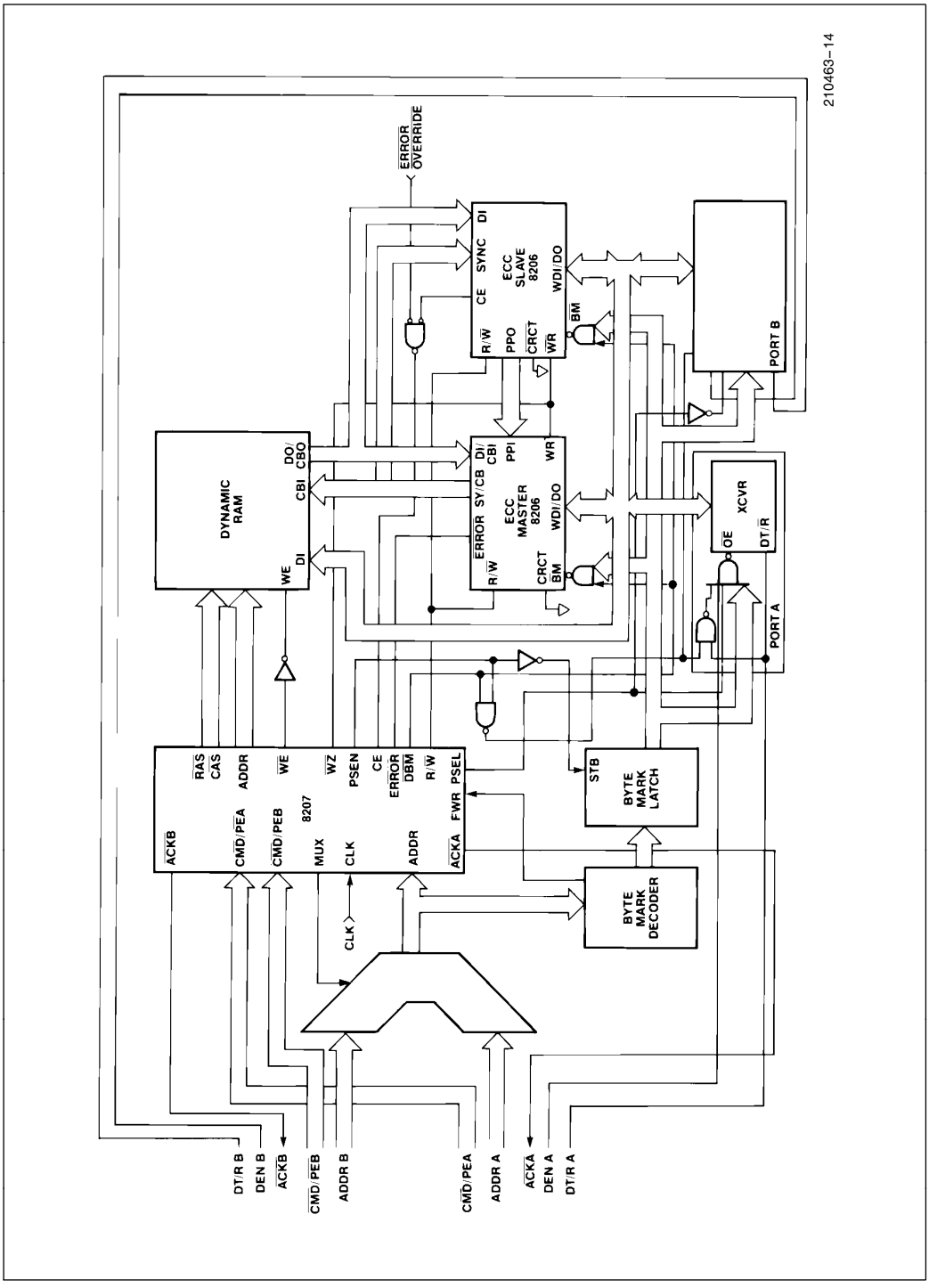


Figure 5. Two-Port ECC Implementation Using the 8207 and the 8206

11

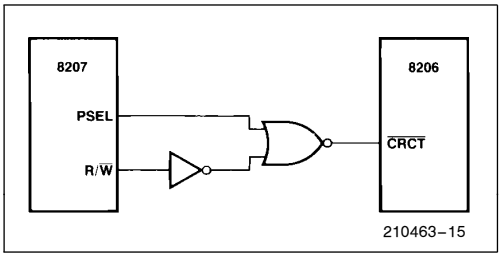


Figure 6. Interface to 8206 $\overline{\text{CRCT}}$ Input when Port A Receives $\overline{\text{AACK}}$ and Port B Receives $\overline{\text{XACK}}$

dynamic RAM operation, and memory initialization if error correction is used. Many dynamic RAMs require this warm-up period for proper operation. The time it takes for the 8207 to recognize a request is shown below.

Non-ECC Systems:

$$T_{RESP} = T_{PROG} + T_{PREP} \quad (2)$$

where:

$$T_{PROG} = (66)\ (T_{CLCL}) \quad (3)$$

which is programming time

$$T_{PREP} = (8)\ (32)\ (T_{CLCL})$$

which is the RAM warm-up time

if $T_{CLCL} = 125$ ns then $T_{RESP} \approx 41\ \mu s$

ECC Systems:

$$T_{RESP} = T_{PROG} + T_{PREP} + T_{INIT} \quad (5)$$

if $T_{CLCL} = 125$ ns then $T_{RESP} \approx 1$ sec.

RESET

RESET is an asynchronous input, the falling edge of which is used by the 8207 to directly sample to logic levels of the PCTLA, PCTLB, RFRQ, and PDI inputs. The internally synchronized falling edge of RESET is used to begin programming operations (shifting in the contents of the external shift register into the PDI input).

Until programming is complete the 8207 registers but does not respond to command or status inputs. A simple means of preventing commands or status from occurring during this period is to differentiate the system reset pulse to obtain a smaller reset pulse for the 8207. The total time of the reset pulse and the 8207 programming time must be less than the time before the first command in systems that alter the default port synchronization programming bits (default is Port A synchronous, Port B asynchronous). Differentiated reset is unnecessary when the default port synchronization programming is used.

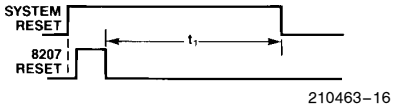
The differentiated reset pulse would be shorter than the system reset pulse by at least the programming period required by the 8207. The differentiated reset pulse first resets the 8207, and system reset would reset the rest of the system. While the rest of the system is still in reset, the 8207 completes its programming. Figure 7 illustrates a circuit to accomplish this task.

Within four clocks after RESET goes active, all the 8207 outputs will go high, except for PSEN, WE, and AO0–2, which will go low.

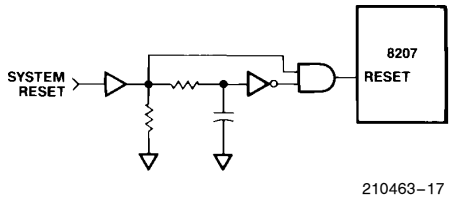
OPERATIONAL DESCRIPTION

Programming the 8207

The 8207 is programmed after reset. On the falling edge of RESET, the logic states of several input pins are latched internally. The falling edge of RESET actually performs the latching, which means that the logic levels on these inputs must be stable prior to that time. The inputs whose logic levels are latched at the end of reset are the PCTLA, PCTLB, REFRQ, and PDI pins. Figure 8 shows the necessary timing for programming the 8207.



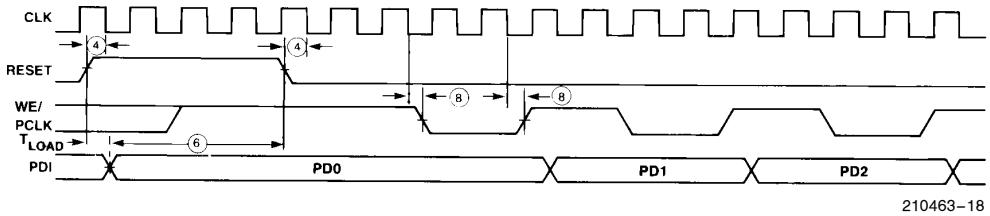
t_I Programming Time of 8207



Differentiated Reset

NOTES:
1. Required only when the port synchronization options (SA & SB) are altered from their initial default values.
2. V_{CC} must be stable before system reset is activated when using this circuit.

Figure 7. 8207 Differentiated Reset Circuit



NOTES:
TRTVCL—Reset is an asynchronous input, if reset occurs before T_1, then it is guaranteed to be recognized.
TPGVCL—Minimum PDI valid time prior to reset going low.
TCLPC—MUX/PCLK delay.
TLOAD—Asynchronous load data propagation delay.

Figure 8. Timing Illustrating External Shift Register Requirements for Programming the 8207

Status/Command Mode

The two processor ports of the 8207 are configured by the states of the PCTLA and PCTLB pins. Which interface is selected depends on the state of the individual port's PCTL pin at the end of reset. If PCTL is high at the end of the reset, the 8086 Status interface is selected; if it is low, then the Command interface is selected.

The status lines of the 80286 are similar in code and timing to the Multibus command lines, while the status code and timing of the 8086 and 8088 are identical to those of the 80186 and 80188 (ignoring the differences in clock duty cycle). Thus there exists two interface configurations, one for the 80286 status or Multibus memory commands, which is called the Command interface, and one for 8086, 8088, 80186 or 80188 status, called the 8086 Status interface. The Command interface can also directly interface to the command lines of the bus controllers for the 8086, 8088, 80186 and the 80286.

The 8086 Status interface allows direct decoding of the status of the iAPX 86, iAPX 88, iAPX 186 and the iAPX 188. Table 3 shows how the status lines are decoded. While in the Command mode the iAPX 286 status can be directly decoded. Microprocessor bus controller read or write commands or Multibus commands can also be directed to the 8207 when in Command mode.

Refresh Options

Immediately after system reset, the state of the REFRQ input pin is examined. If REFRQ is high, the 8207 provides the user with the choice between self-refresh or user-generated refresh with failsafe protection. Failsafe protection guarantees that if the

Table 3A. Status Coding of 8086, 80186 and 80286

Status Code			Function	
$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	8086/80186	80286
0	0	0	Interrupt	Interrupt
0	0	1	I/O Read	I/O Read
0	1	0	I/O Write	I/O Write
0	1	1	Halt	Idle
1	0	0	Instruction Fetch	Halt
1	0	1	Memory Read	Memory Read
1	1	0	Memory Write	Memory Write
1	1	1	Idle	Idle

Table 3B. 8207 Response

8207 Command			Function	
PCTL	$\overline{RD}$	$\overline{WR}$	8086/80186 Status Interface	80286 Status or Command Interface
0	0	0	Ignore	Ignore*
0	0	1	Ignore	Read
0	1	0	Ignore	Write
0	1	1	Ignore	Ignore*
1	0	0	Read	Ignore
1	0	1	Read	Inhibit
1	1	0	Write	Inhibit
1	1	1	Ignore	Ignore

*Illegal with CFS = 0

user does not come back with another refresh request before the internal refresh interval counter times out, a refresh request will be automatically generated. If the REFRQ pin is low immediately after a reset, the 8207 is programmed in a non-failsafe refresh mode. In this mode the refresh cycle is initiated only upon receipt of an external refresh request. The user has the choice of a single external refresh cycle, burst refresh or no refresh.

Internal Refresh Only

For the 8207 to generate internal refresh requests, it is necessary only to strap the REFRQ input pin high.

External Refresh with Failsafe

To allow user-generated refresh requests with failsafe protection, it is necessary to hold the REFRQ input high until after reset. Thereafter, a low-to-high transition on this input causes a refresh request to be generated and the internal refresh interval counter to be reset. A high-to-low transition has no effect on the 8207. A refresh request is not recognized until a previous request has been serviced.

External Refresh without Failsafe

To generate single external refresh requests without failsafe protection, it is necessary to hold REFRQ low until after reset. Thereafter, bringing REFRQ high for one clock period causes refresh request to be generated. A refresh request is not recognized unitl a previous request has been serviced.

Burst Refresh

Burst refresh is implemented through the same procedure as a single external refresh without failsafe (i.e., REFRQ is kept low until after reset). Thereafter, bringing REFRQ high for at least two clock periods causes a burst of up to 128 row address locations to be refreshed.

The ECC-configured systems, 128 locations are scrubbed. Any refresh request is not recognized until a previous request has been serviced (i.e., burst completed).

No Refresh

It is necessary to hold REFRQ low until after reset. This is the same as programming External Refresh without Failsafe. No refresh is accomplished by keeping REFRQ low.

Option Program Data Word

The program data word consists of 16 program data bits, PD0–PD15. If the first program data bit shifted into the 8207 (PD0) is set to logic 1, the 8207 is configured to support ECC. If it is logic 0, the 8207 is configured to support a non-ECC system. The remaining bits, PD1–PD15, may then be programmed to optimize a selected configuration. Figures 9 and 10 show the Program words for non-ECC and ECC operation.

Using an External Shift Register

The 8207 may be configured to use an external shift register with asynchronous load capability such as a 74LS165. The reset pulse serves to parallel load the shift register and the 8207 supplies the clocking signal to shift the data in. Figure 11 shows a sample circuit diagram of an external shift register circuit.

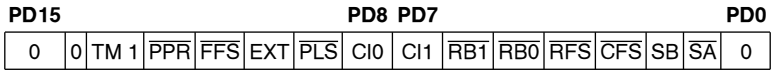
Serial data is shifted into the 8207 via the PDI pin (57), and clock is provided by the MUX/PCLK pin (12), which generates a total of 16 clock pulses. After programming is complete, data appearing at the input of the PDI pin is ignored. MUX/PCLK is a dual-function pin. During programming, it serves to clock the external shift register, and after programming is completed, it reverts to a MUX conrol pin. As the pin changes state to select different port addresses, it continues to clock the shift register. This does not present a problem because data at the PDI pin is ignored after programming. Figure 8 illustrates the timing requirements of the shift register circuitry.

ECC Mode (ECC Program Bit)

The state of PDI (Program Data In) pin at reset determines whether the system is an ECC or non-ECC configuration. It is used internally by the 8207 to begin configuring timing circuits, even before programming is completely finished. The 8207 then begins programming the rest of the options.

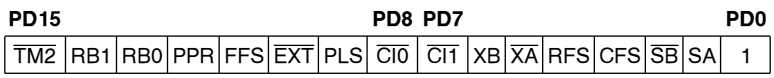
Default Programming Options

After reset, the 8207 serially shifts in a program data word via the PDI pin. This pin may be strapped either high or low, or connected to an external shift register. Strapping PDI high causes the 8207 to default to a particular system configuration with error correction, and strapping it low causes the 8207 to default to a particular system configuration without error correction. Table 4 shows the default configurations.



Program Data Bit	Name	Polarity/Function
PD0	ECC	ECC = 0 For Non-ECC Mode
PD1	$\overline{SA}$	$\overline{SA}$ = 0 Port A is Synchronous $\overline{SA}$ = 1 Port A is Asnychronous
PD2	SB	SB = 0 Port B is Asynchronous SB = 1 Port B is Synchronous
PD3	$\overline{CFS}$	$\overline{CFS}$ = 0 Fast-Cycle iAPX 286 Mode $\overline{CFS}$ = 1 Slow-Cycle iAPX 86 Mode
PD4	$\overline{RFS}$	$\overline{RFS}$ = 0 Fast RAM $\overline{RFS}$ = 1 Slow RAM
PD5 PD6	$\overline{RB0}$ $\overline{RB1}$	RAM Bank Occupancy See Table 2
PD7 PD8	CI1 CI0	Count Interval Bit 1; see Table 6 Count Interval Bit 0; see Table 6
PD9	$\overline{PLS}$	$\overline{PLS}$ = 0 Long Refresh Period $\overline{PLS}$ = 1 Short Refresh Period
PD10	EXT	EXT = 0 Not Extended EXT = 1 Extended
PD11	$\overline{FFS}$	$\overline{FFS}$ = 0 Fast CPU Frequency $\overline{FFS}$ = 1 Slow CPU Frequency
PD12	$\overline{PPR}$	$\overline{PPR}$ = 0 Most Recently Used Port Priority $\overline{PPR}$ = 1 Port A Preferred Priority
PD13	TM1	TM1 = 0 Test Mode 1 Off TM1 = 1 Test Mode 1 Enabled
PD14	0	Reserved, Must be Zero
PD15	0	Reserved, Must be Zero

Figure 9. Non-ECC Mode Program Data Word



Program Data Bit	Name	Polarity/Function
PD0	ECC	ECC = 1 ECC Mode
PD1	SA	SA = 0 Port A Asynchronous SA = 1 Port A Synchronous
PD2	$\overline{SB}$	$\overline{SB}$ = 0 Port B Synchronous $\overline{SB}$ = 1 Port B Asynchronous
PD3	CFS	CFS = 0 Slow-Cycle iAPX 86 Mode CFS = 1 Fast-Cycle iAPX 286 Mode
PD4	RFS	RFS = 0 Slow RAM RFS = 1 Fast RAM
PD5	$\overline{XA}$	$\overline{XA}$ = 0 MULTIBUS-Compatible ACKA $\overline{XA}$ = 1 Advanced ACKA Not Multibus-Compatible
PD6	XB	Advanced ACKB Not Multibus-Compatible XB = 1 Multibus-Compatible ACKB
PD7 PD8	$\overline{CI1}$ $\overline{CI0}$	Count Interval Bit 1; see Table 6 Count Interval Bit 0; see Table 6
PD9	PLS	PLS = 0 Short Refresh Period PLS = 1 Long Refresh Period
PD10	$\overline{EXT}$	$\overline{EXT}$ = 0 Master and Slave EDCU $\overline{EXT}$ = 1 Master EDCU Only
PD11	FFS	FFS = 0 Slow CPU Frequency FFS = 1 Fast CPU Frequency
PD12	PPR	PPR = 0 Port A Preferred Priority PPR = 1 Most Recently Used Port Priority
PD13 PD14	RB0 RB1	RAM Bank Occupancy See Table 2
PD15	$\overline{TM2}$	$\overline{TM2}$ = 0 Test Mode 2 Enabled $\overline{TM1}$ = 1 Test Mode 2 Off

Figure 10. ECC Mode Program Data Word

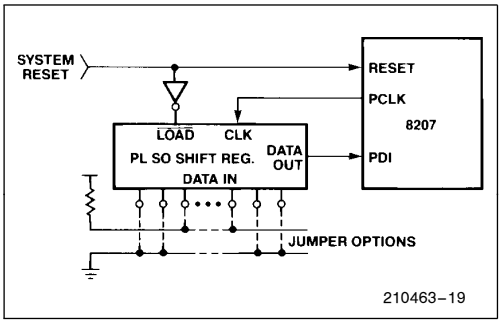


Figure 11. External Shift Register Interface

Table 4A. Default Non-ECC Programming, PDI Pin (57) Tied to Ground

Port A is Synchronous ($\overline{\text{EAACKA}}$ and $\overline{\text{XACKA}}$)
Port B is Asynchronous ($\overline{\text{LAACKB}}$ and $\overline{\text{XACKB}}$)
Fast-Cycle Processor Interface (iAPX 286)
Fast RAM
Refresh Interval uses 236 Clocks
128 Row Refresh in 2 ms; 256 Row Refresh in 4 ms
Fast Processor Clock Frequency (16 MHz)
''Most Recently Used'' Priority Scheme
4 RAM banks occupied

Table 4B. Default ECC Programming, PDI Pin (57) Tied to V_{CC}

Port A is Synchronous
Port B is Asynchronous
Fast-Cycle Processor Interface (iAPX 286)
Fast RAM
Port A has $\overline{\text{EAACKA}}$ strobe (non-multibus)
Port B has $\overline{\text{XACKB}}$ strobe (multibus)
Refresh interval uses 236 clocks
128 Row refresh in 2 ms; 256 Row refresh in 4 ms
Master EDCU only (16-bit system)
Fast Processor Clock Frequency (16 MHz)
''Most Recently Used'' Priority Scheme
4 RAM banks ocuppied

If further system flexibility is needed, one or two external shift registers can be used to tailor the 8207 to its operating environment.

Synchronous/Asynchronous Mode (SA and SB Program Bits)

Each port of the 8207 may be independently configured to accept synchronous or asynchronous port commands ($\overline{\text{RD}}$, $\overline{\text{WR}}$, PCTL) and Port Enable ($\overline{\text{PE}}$) via the program bits SA and SB. The state of the SA and SB programming bits determine whether their associated ports are synchronous or asynchronous.

While a port may be configured with either the Status or Command interface in the synchronous mode, certain restrictions exist in the asynchronous mode. An asynchronous Command interface using the control lines of the Multibus is supported, and an asynchronous 8086 interface using the control lines of the 8086 is supported, with the use of TTL gates as illustrated in Figure 2. In the 8086 case, the TTL gates are needed to guarantee that status does not appear at the 8207's inputs too much before address, so that a cycle would start before address was valid.

Microprocessor Clock Frequency Option (CFS and FFS Program Bits)

The 8207 can be programmed to interface with slow-cycle microprocessors like the 8086, 8088, 81088 and 80186 or fast-cycle microprocessors like the 80286. The CFS bit configures the microprocessor interface to accept slow or fast cycle signals from either microprocessor group.

The FFS bit is used to select the speed of the microprocessor clock. Table 5 shows the various microprocessor clock frequency options that can be programmed.

Table 5. Microprocessor Clock Frequency Options

Program Bits		Processor	Clock Frequency
CFS	FFS		
0	0	iAPX 86, 88, 186, 188	$\leq$ 6 MHz
0	1	iAPX 86, 88, 186, 188	$>$ 6 MHz
1	0	iAPX 286	$\leq$12 MHz
1	1	iAPX 286	$>$12 MHz

The external clock frequency must be programmed so that the failsafe refresh repetition circuitry can adjust its internal timing accordingly to produce a refresh request as programmed.

RAM Speed Option (RFS Program Bit)

The RAM Speed programming option determines whether RAM timing will be optimized for a fast or slow RAM.

Refresh Period Options (CI0, CI1 and PLS Program Bits)

The 8207 refreshes with either 128 rows every 2 milliseconds or 256 rows every 4 milliseconds. This translates to one refresh cycle being executed approximately once every 15.6 microseconds. This rate can be changed to 256 rows every 2 milliseconds or a refresh approximately once every 7.8 microseconds via the Period Long/Short, program bit PLS, programming option. The 7.8 microsecond refresh request rate is intended for those RAMs, 64K and above, which may require a faster refresh rate.

In addition to PLS program option, two other programming bits for refresh exist: Count Interval 0 (CI0) and Count Interval 1 (CI1). These two programming bits allow the rate at which refresh requests are generated to be increased in order to permit refresh requests to be generated close to the same 15.6 or 7.8 microsecond period when the 8207 is operating at reduced frequencies. The interval between refreshes is decreased by 0%, 10%, 20%, or 30% as a function of how the count interval bits are programmed. A 5% guardband is built-in to allow for any clock frequency variations. Table 6 shows the refresh period options available.

The numbers tabulated under Count Interval represent the number of clock periods between internal refresh requests. The percentages in parentheses represent the decrease in the interval between refresh requests. Note that all intervals have a built-in 5% (approximately) safety factor to compensate for minor clock frequency deviations and non-immediate response to internal refresh requests.

Extend Option (EXT Program Bit)

The Extend option lengthens the memory cycle to allow longer access time which may be required by the system. Extend alters the RAM timing to compensate for increased loading on the Row and Column Address Strobes, and in the multiplexed Address Out lines.

Port Priority Option and Arbitration (PPR Program Bit)

The 8207 has to internally arbitrate among three ports: Port A, Port B and Port C—the refresh port. Port C is an internal port dedicated to servicing refresh requests, whether they are generated internally by the refresh interval counter, or externally by the user. Two arbitration approaches are available via

Table 6. Refresh Count Interval Table

Ref. Period (μs)	CFS	PLS	FFS	Count Interval CI1, CI0 (8207 Clock Periods)			
				00 (0%)	01 (10%)	10 (20%)	11 (30%)
15.6	1	1	1	236	212	188	164
7.8	1	0	1	118	106	94	82
15.6	1	1	0	148	132	116	100
7.8	1	0	0	74	66	58	50
15.6	0	1	1	118	106	94	82
7.8	0	0	1	59	53	47	41
15.6	0	1	0	74	66	58	50
7.8	0	0	0	37	33	29	25

NOTE:
Refresh period = clock period × refresh count interval.

the Port Priority programming option, program bit PPR. PPR determines whether the most recently used port will remain selected (PPR = 1) or whether Port A will be favored or preferred over Port B (PPR = 0).

A port is selected if the arbiter has given the selected port direct access to the timing generators. The front-end logic, which includes the arbiter, is designed to operate in parallel with the selected port. Thus a request on the selected port is serviced immediately. In contrast, an unselected port only has access to the timing generators through the front-end logic. Before a RAM cycle can start for an unselected port, that port must first become selected (i.e., the MUX output now gates that port's address into the 8207 in the case of Port A or B). Also, in order to allow its address to stabilize, a newly selected port's first RAM cycle is started by the front-end logic. Therefore, the selected port has direct access to the timing generators. What all this means is that a request on a selected port is started immediately, while a request on an unselected port is started two to three clock periods after the request, assuming that the other two ports are idle. Under normal operating conditions, this arbitration time is hidden behind the RAM cycle of the selected port so that as soon as the present cycle is over a new cycle is started. Table 7 lists the arbitration rules for both options.

Port LOCK Function

The LOCK function provides each port with the ability to obtain uninterrupted access to a critical region of memory and, thereby, to guarantee that the opposite port cannot ''sneak in'' and read from or write to the critical region prematurely.

Only one LOCK pin is present and is multiplexed between the two ports as follows: when MUX is high, the 8207 treats the LOCK input as originating at PORT A, while when MUX is low, the 8207 treats LOCK as originating at PORT B. When the 8207 recognizes a LOCK, the MUX output will remain pointed to the locking port until LOCK is deactivated. Refresh is not affected by LOCK and can occur during a locked memory cycle.

Table 7. The Arbitration Rules for the Most Recently Used Port Priority and for Port A Priority Options Are As Follows:

1.	If only one port requests service, then that port—if not already selected—becomes selected.
2a.	When no service requests are pending, the last selected processor port (Port A or B) will remain selected. (Most Recently Used Port Priority Option.)
2b.	When no service requests are pending, Port A is selected whether it requests service or not. (Port A Priority Option.)
3.	During reset initialization only Port C, the refresh port, is selected.
4.	If no processor requests are pending after reset initialization, Port A will be selected.
5b.	If Ports A and B simultaneously(*) request service while Port C is selected, then the next port to be selected is Port A. (Port A Priority Option.)
6.	If a port simultaneously requests service with the currently selected port, service is granted to the selected port.
7.	The MUX output remains in its last state whenever Port C is selected.
8.	If Port C and either Port A or Port B (or both) simultaneously request service, then service is granted to the requester whose port is already selected. If the selected port is not requesting service, then service is granted to Port C.
9.	If during the servicing of one port, the other port requests service before or simultaneously with the refresh port, the refresh port is selected. A new port is not selected before the presently selected port is deactivated.
10.	Activating LOCK will mask off service requests from Port B if the MUX output is high, or from Port A if the MUX output is low.

NOTE:
*By ''simultaneous'' it is meant that two or more requests are valid at the clock edge at which the internal arbiter samples them.

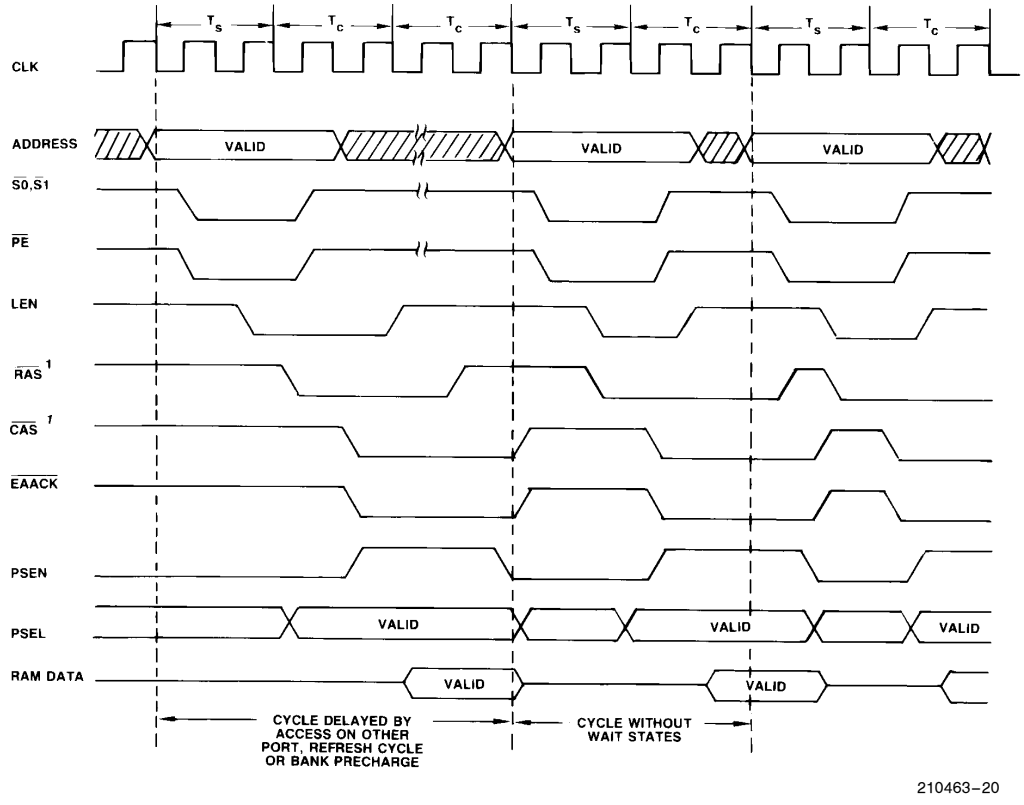
Dual-Port Considerations

For both ports to be operated synchronously, several conditions must be met. The processors must be the same type (Fast or Slow Cycle) as defined by Table 8 and they must have synchronized clocks. Also when processor types are mixed, even though the clocks may be in phase, one frequency may be twice that of the other. So to run both ports synchronous using the status interface, the processors must have related timings (both phase and frequency). If these conditions cannot be met, then one port must run synchronous and the other asynchronous.

Figure 3 illustrates an example of dual-port operation using the processors in the slow cycle group. Note the use of cross-coupled NAND gates at the MUX output for minimizing contention between the two latches, and the use of flip flops on the status lines of the asynchronous processor for delaying the status and thereby guaranteeing RAS will not be issued, even in the worst case, until address is valid.

Processor Timing

In order to run without wait states, $\overline{AACK}$ must be used and connected to the $\overline{SRDY}$ input of the appropriate bus controller. $\overline{AACK}$ is issued relative to a point within the RAM cycle and has no fixed relationship to the processor's request. The timing is such, however, that the processor will run without wait states, barring refresh cycles, bank precharge, and RAM accesses from the other port. In non-ECC fast cycle, fast RAM, non-extended configurations (80286), $\overline{AACK}$ is issued on the next falling edge of



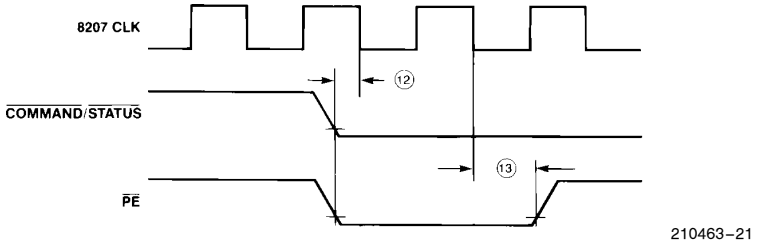
NOTE:
1. The RAS and CAS shown in figure are different banks being accessed.

Figure 14. iAPX 286/8207 Synchronous-Status Timing Programmed in non-ECC Mode, C0 Configuration (Read Cycle)

the clock after the edge that issues RAS. In non-ECC, slow cycle, non-extended, or extended with fast RAM cycle configurations (8086, 80188, 80186), $\overline{AACK}$ is issued on the same clock cycle that issues RAS. Figure 14 illustrates the timing relationship between $\overline{AACK}$, the RAM cycle, and the processor cycle for several different situations.

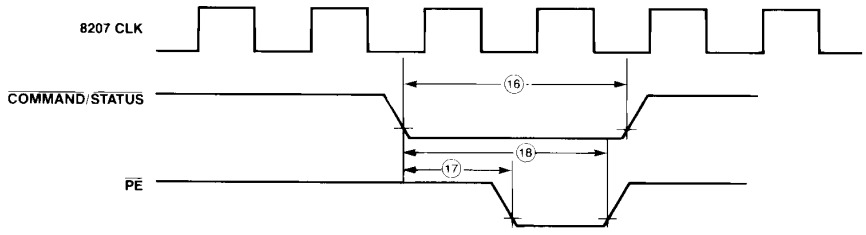
Port Enable ($\overline{PE}$) setup time requirements depend on whether the associated port is configured for synchronous or asynchronous fast or slow cycle operation. In a synchronous fast cycle configuration, $\overline{PE}$ is required to be setup to the same clock edge as the status or commands. If $\overline{PE}$ is true (low), a RAM cycle is started; if not, the cycle is aborted. The memory cycle will only begin when both valid signals ($\overline{PE}$ and $\overline{RD}$ or WR) are recognized at a particular clock edge. In asynchronous operation. $\overline{PE}$ is required to be setup to the same clock edge as the internally synchronized status or commands. Externally, this allows the internal synchronization delay to be added to the status (or command)-to-$\overline{PE}$ delay time, thus allowing for more external decode time that is available in synchronous operation.

The minimum synchronization delay is the additional amount that $\overline{PE}$ must be held valid. If $\overline{PE}$ is not held valid for the maximum synchronization delay time, it is possible that $\overline{PE}$ will go invalid prior to the status or command being synchronized. In such a case the 8207 aborts the cycle. If a memory cycle intended for the 8207 is aborted, then no acknowledge ($\overline{AACK}$ or $\overline{XACK}$) is issued and the processor locks up in endless wait states. Figure 15 illustrates the status (command) timing requirements for synchronous and asynchronous systems. Figures 16 and 17 show a more detailed hook-up of the 8207 to the 8086 and the 80286, respectively.



210463–21

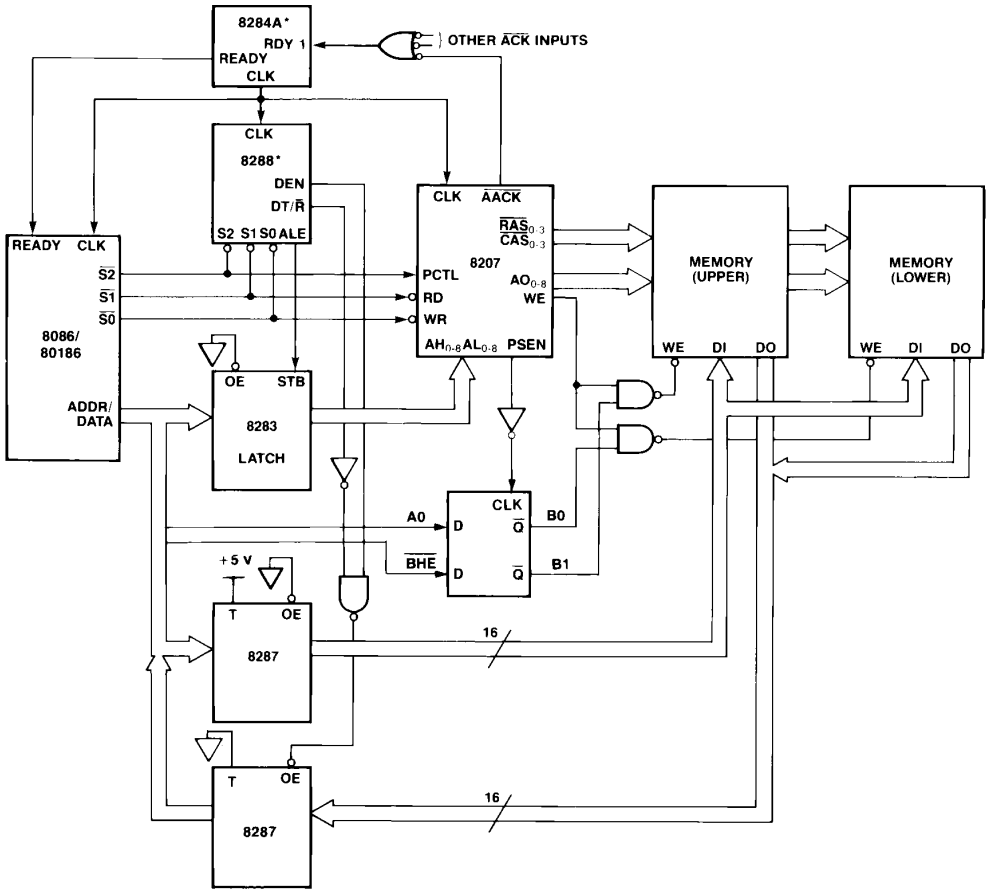
(A) $\overline{PE}$ Set-Up and Hold Time Requirements for Fast Cycle, Synchronous Operation (80286 CMD/Status)



210463–22

(B) $\overline{PE}$ Timing Requirements for Fast or Slow Cycle Asynchronous Operation

Figure 15



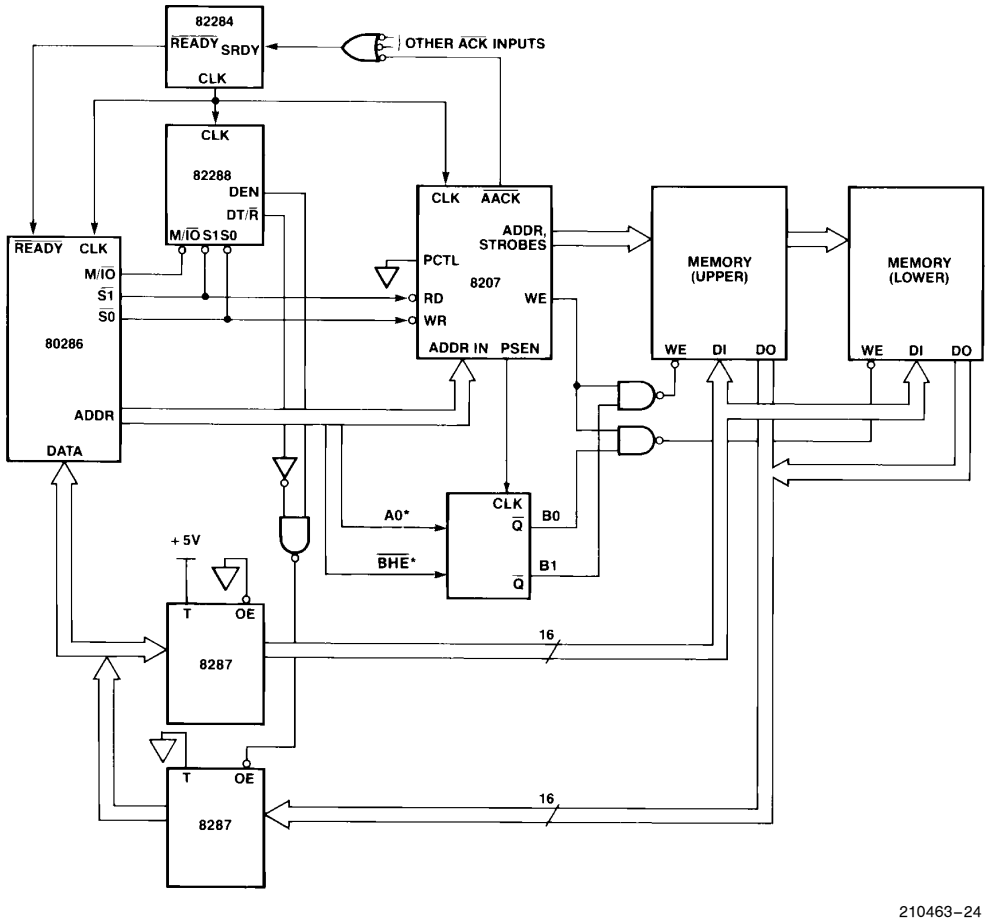
NOTE:
*These components are not necessary when using the 80186. These functions are provided directly by the 80186.

Figure 16. 8086/80186, 8207 Single Port Non-ECC Synchronous Systems

Memory Acknowledge (AACK, XACK)

In system configurations without error correction, two memory acknowledge signals per port are supplied by the 8207. They are the Advanced Acknowledge strobe ($\overline{AACK}$) and the Transfer Acknowledge strobe ($\overline{XACK}$). The CFS programming bit determines for which processor $\overline{AACKA}$ and $\overline{AACKB}$ are optimized, either 80286 (CFS = 1) or 8086/186 (CFS = 0), while the SA and SB programming bits optimize $\overline{AACK}$ for synchronous operation ("early" $\overline{AACK}$) or asynchronous operation ("late" $\overline{AACK}$).

Both the early and late $\overline{AACK}$ strobes are three clocks long for CFS = 1 and two clocks long for CFS = 0. The $\overline{XACK}$ strobe is asserted when data is valid (for reads) or when data may be removed (for writes) and meets the Multibus requirements. $\overline{XACK}$ is removed asynchronously by the command going



NOTE:
While the 8207 does not need the input addresses latched, A0, $\overline{BHE}$ must come from the latched address bus.

Figure 17. 80286 Hook-Up to 8207 Non-ECC Synchronous System-Single Port

inactive. Since in asynchronous operation the 8207 removes read data before late $\overline{AACK}$ or $\overline{XACK}$ is recognized by the CPU, the user must provide for data latching in the system until the CPU reads the data. In synchronous operation, data latching is unnecessary since the 8207 will not remove data until the CPU has read it.

In ECC-based systems there is one memory acknowledge ($\overline{XACK}$ or $\overline{AACK}$) per port and a programming bit associated with each acknowledge. If the X programming bit is active, the strobe is configured as $\overline{XACK}$, while if the bit is inactive, the strobe is configured as $\overline{AACK}$. As in non-ECC, the SA and SB programming bits determine whether the $\overline{AACK}$ strobe is early or late ($\overline{EAACK}$ or $\overline{LAACK}$).

Data will always be valid a fixed time after the occurrence of the advanced acknowledge. Table 9 summarizes the various transfer acknowledge options.

Table 8. Processor Interface/Acknowledge Summary

Cycle	Processor	Request Type	Sync/Async Interface	Acknowledge Type
Fast Cycle CFS = 1	80286	Status	Sync	$\overline{\text{EAACK}}$
	80286	Status	Async	$\overline{\text{LAACK}}$
	80286	Command	Sync	$\overline{\text{EAACK}}$
	80286	Command	Async	$\overline{\text{LAACK}}$
	8086/80186	Status	Async	$\overline{\text{LAACK}}$
	8086/80186	Command	Async	$\overline{\text{LAACK}}$
	Multibus	Command	Async	$\overline{\text{XACK}}$
Slow Cycle CFS = 0	8086/80186	Status	Sync	$\overline{\text{EAACK}}$
	8086/80186	Status	Async	$\overline{\text{LAACK}}$
	8086/80186	Command	Sync	$\overline{\text{EAACK}}$
	8086/80186	Command	Async	$\overline{\text{LAACK}}$
	Multibus	Command	Async	$\overline{\text{XACK}}$

Table 9. Memory Acknowledge Option Summary

	Synchronous	Asynchronous	XACK
Fast Cycle	AACK Optimized for Local 80286	AACK Optimized for Remote 80286	Multibus Compatible
Slow Cycle	AACK Optimized for Local 8086/186	AACK Optimized for Remote 8086/186	Multibus Compatible

Test Modes

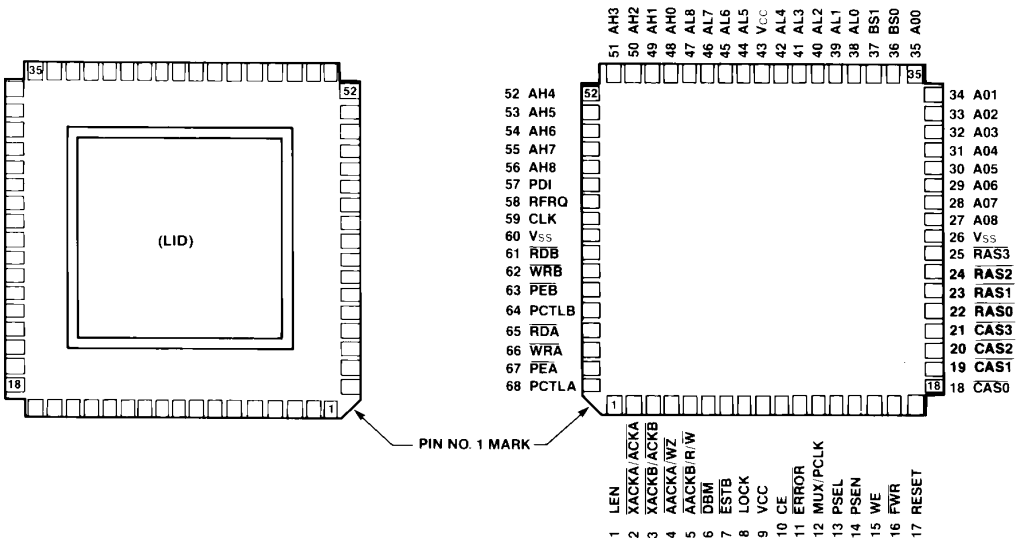
Two special test modes exist in the 8207 to facilitate testing. Test Mode 1 (non-ECC mode) splits the refresh address counter into two separate counters and Test Mode 2 (ECC mode) presets the refresh address counter to a value slightly less than rollover.

Test Mode 1 splits the address counter into two, and increments both counters simultaneously with each refresh address update. By generating external refresh requests, the tester is able to check for proper operation of both counters. Once proper individual counter operation has been established, the 8207 must be returned to normal mode and a second test performed to check that the carry from the first counter increments the second counter. The outputs of the counters are presented on the address out bus with the same timing as the row and column addresses of a normal scrubbing operation. During Test Mode 1, memory initialization is inhibited, since the 8207, be definition, is in non-ECC mode.

Test Mode 2 sets the internal refresh counter to a value slightly less than rollover. During functional testing other than that covered in Test Mode 1, the 8207 will normally be set in Test Mode 2. Test Mode 2 eliminates memory initialization in ECC mode. This allows quick examination of the circuitry which brings the 8207 out of memory initialization and into normal operation.

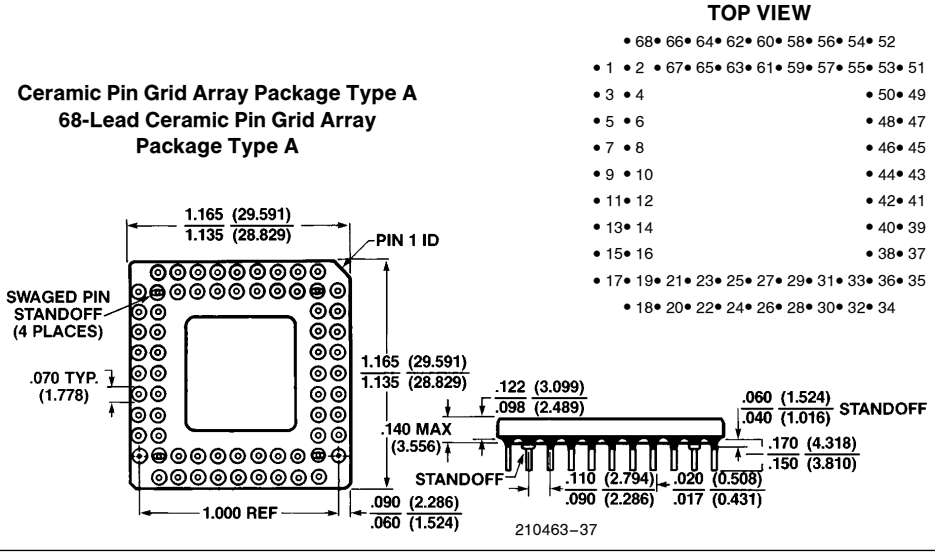
General System Considerations

The RAS_{0-3}, CAS_{0-3}, AO_{0-8}, output buffers were designed to directly drive the heavy capacitive loads associated with dynamic RAM arrays. To keep the RAM driver outputs from ringing excessively in the system environment and causing noise in other output pins it is necessary to match the output impedance of the RAM output buffers with the RAM array by using series resistors and to add series resistors to other control outputs for noise reduction if necessary. Each application may have different impedance characteristics and may require different series resistance values. The series resistance values should be determined for each application. In non-ECC systems unused ECC input pins should be tied high or low to improve noise immunity.



NOTE:
LCC is mounted lid-down into socket.

Figure 19. 8207 Pinout Diagram



8207 Pin Grid Array (PGA) Pin-Out

Packaging

The 8207 is packaged in a 68 lead JEDEC Type A Leadless Chip Carrier (LCC) and in Pin Grid Array (PGA), both in Ceramic. The package designations are R and A respectively.

eg: R 8207-8 LCC, 8 MHz DRAM Controller
eg: A 8207-16 PGA, 16 MHz DRAM Controller

NOTE:
The pin-out of the PGA is the same as the socketed pinout of the LCC.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature
Under Bias . . . . . −0°C to +70°C
Storage Temperature . . . . . −65°C to +150°C
Voltage on Any Pin with
Respect to Ground . . . . . −0.5V to +7V
Power Dissipation (Note 2) 2.5W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

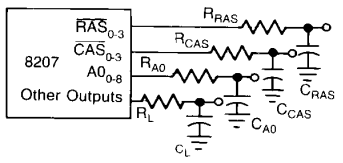
**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS V_{CC} = 5.0V ±10% for 8207-10, 8207-8; T_A = 0°C to 70°C; V_{SS} = GND; V_{CC} = 5.0V ±5% for 8207-16 (Note 2)

Symbol	Parameter	Min	Max	Units	Comments
V_{IL}	Input Low Voltage	−0.5	+0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} +0.5	V	
V_{OL}	Output Low Voltage		0.45	V	(Note 1)
V_{OH}	Output High Voltage	2.4		V	(Note 1)
V_{ROL}	RAM Output Low Voltage		0.45	V	(Note 1)
V_{ROH}	RAM Output High Voltage	2.6		V	(Note 1)
I_{CC}	Supply Current		455	mA	T_A = 0°C
I_{LI}	Input Leakage Current		±10	μA	0V ≤ V_{IN} ≤ V_{CC}
V_{CL}	Clock Input Low Voltage	−0.5	±0.6	V	
V_{CH}	Clock Input High Voltage	3.8	V_{CC} + 0.5	V	
C_{IN}	Input Capacitance		20	pF	fc = 1 MHz(2)

NOTE:
1. I_{OL} = 5 mA and I_{OH} = −0.2 mA (Typically I_{OL} = 10 mA and I_{OH} = −0.88 mA). WE: I_{OL} = 8 mA.
2. Sampled, not 100% tested.

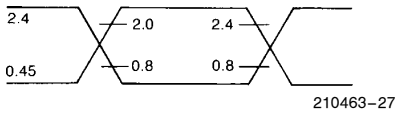
A.C. TESTING LOAD CIRCUIT(2)



210463-26

R_{RAS} = 39Ω
R_{CAS} = 39Ω
R_{A0} = 22Ω
R_L = 39Ω
C_{RAS} = 150 pF
C_{CAS} = 150 pF
C_{A0} = 380 pF
C_L = 100 pF

A.C. TESTING INPUT, OUTPUT WAVEFORM



210463-27

A.C. Testing inputs (except clock) are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0" (clock is driven at 4.0V and 0.45V for Logic "1" and "0" respectively). Timing measurements are made at 2.0V, 2.4V for Logic "1" and 0.8V for Logic "0".

A.C. CHARACTERISTICS

V_{CC} = 5V ±10% for 8207-8; T_A = 0°C to 70°C; V_{CC} = +5V ±5% for 8207-16

Measurements made with respect to RAS_{0-3}, CAS_{0-3}, AO_{0-8}, are a +2.4V and 0.8V. All other pins are measured at 2.0V and 0.8V. All times are ns unless otherwise indicated. Testing done with specified test load.

Ref	Symbol	Parameter	8207-16, -8		8207-10		Units	Notes
			Min	Max	Min	Max		
CLOCK AND PROGRAMMING								
—	tF	Clock Fall Time		10		10	ns	3
—	tR	Clock Rise Time		10		10	ns	3
1	TCLCL	Clock Period 8207-16	62.5	200			ns	1
		8207-10			100	250	ns	2
		8207-8	125	500			ns	2
2	TCL	Clock Low Time 8207-16	15	180			ns	1
		8207-10			TCLCL/2−12		ns	2
		8207-8	TCLCL/2−12				ns	2
3	TCH	Clock High Time 8207-16	20	180			ns	1
		8207-10			TCLCL/3−3		ns	2
		8207-8	TCLCL/3−3				ns	2
4	TRTVCL	Reset to CLK ↓ Setup	20		40		ns	4
5	TRTH	Reset Pulse Width	4TCLCL		4TCLCL		ns	
6	TPGVRTL	PCTL, PDI, RFRQ to RESET ↓ Setup	125		125		ns	5
7	TRTLPGX	PCTL,RFRQ to RESET ↓ Hold	10		10		ns	
8	TCLPC	PCLK from CLK ↓ Delay		45		45	ns	
9	TPDVCL	PDin to CLK ↓ Setup	60		60		ns	
10	TCLPDX	PDin to CLK ↓ Hold	40		40		ns	6
RAM WARM-UP AND INITIALIZATION								
64	TCLWZL	$\overline{WZ}$ from CLK ↓ Delay		40		40	ns	7
SYNCHRONOUS μP PORT INTERFACE								
11	TPEVCL	$\overline{PE}$ to CLK ↓ Setup	27		27		ns	2
12	TKVCL	$\overline{RD}$, $\overline{WR}$, $\overline{PE}$, PCTL to CLK ↓ Setup	20				ns	1
13	TCLKX	$\overline{RD}$, $\overline{WR}$, $\overline{PE}$, PCTL to CLK ↓ Hold	0		0		ns	
14	TKVCH	$\overline{RD}$, $\overline{WR}$, PCTL to CLK ↑ Setup	20		20		ns	2

A.C. CHARACTERISTICS (Continued)

V_{CC} = 5V ±10% for 8207-8; T_A = 0°C to 70°C; V_{CC} = +5V ±5% for 8207-16

Measurements made with respect to RAS_{0-3}, CAS_{0-3}, AO_{0-8}, are a +2.4V and 0.8V. All other pins are measured at 2.0V and 0.8V. All times are ns unless otherwise indicated. Testing done with specified test load.

Ref	Symbol	Parameter	8207-16, -8		8207-10		Units	Notes
			Min	Max	Min	Max		
ASYNCHRONOUS μP PORT INTERFACE								
15	TRWVCL	$\overline{RD}$, $\overline{WR}$ to CLK ↓ Setup	20		20		ns	8, 9
16	TRWL	$\overline{RD}$, $\overline{WR}$ Pulse Width	2TCLCL+30		2TCLCL+30		ns	
17	TRWLPEV	$\overline{PE}$ from $\overline{RD}$, CFS=1 $\overline{WR}$ ↓ Delay CFS=0		TCLCL−20 TCLCL−30		 TCLCL−20	ns ns	1 2
18	TRWLPEX	$\overline{PE}$ to $\overline{RD}$, $\overline{WR}$ ↓ Hold	2TCLCL+30		2TCLCL+30		ns	
19	TRWLPTV	PCTL from $\overline{RD}$, $\overline{WR}$ ↓ Delay		TCLCL−30		TCLCL−30	ns	2
20	TRWLPTX	PCTL to $\overline{RD}$, $\overline{WR}$ ↓ Hold	2TCLCL+30		2TCLCL+30		ns	2
21	TRWLPTV	PCTL from $\overline{RD}$, $\overline{WR}$ ↓ Delay		2TCLCL−20		2TCLCL−30	ns	1
22	TRWLPTX	PCTL to $\overline{RD}$, $\overline{WR}$ ↓ Hold	3TCLCL+30		3TCLCL+40		ns	1
RAM INTERFACE								
23	TAVCL	AL, AH, BS to CLK ↓ Setup	35+tASR		35+tASR		ns	10
24	TCLAX	AL, AH, BS to CLK ↓ Hold	0		0		ns	
25	TCLLN	LEN from CLK ↓ Delay		35			ns	1
26	TCLRSL	$\overline{RAS}$ ↓ from CLK ↓ Delay		35		35	ns	
27	TRCD	$\overline{RAS}$ to $\overline{CAS}$ CFS=1 Delay CFS=0	TCLCL−25 TCLCL/2−25		 25		ns ns ns	1, 14 11, 14
28	TCLRSH	$\overline{RAS}$ ↑ from CLK ↓ Delay		50		50	ns	
29	TRAH	Row A0 to CFS=1 $\overline{RAS}$ Hold CFS=0	TCLCL/2−11 TCLCL/4−11		 18		ns ns	1, 13, 15 11, 15
30	TASR	Row A0 to $\overline{RAS}$ Setup	0		0			10, 18
31	TASC	Column A0 to CFS=1 $\overline{CAS}$ ↓ Setup CFS=0	0 5		 5		ns ns	13, 19, 20 13, 19, 20

A.C. CHARACTERISTICS (Continued)

V_{CC} = 5V ±10% for 8207-8; T_A = 0°C to 70°C; V_{CC} = +5V ±5% for 8207-16

Measurements made with respect to RAS_{0-3}, CAS_{0-3}, AO_{0-8}, are a +2.4V and 0.8V. All other pins are measured at 2.0V and 0.8V. All times are ns unless otherwise indicated. Testing done with specified test load.

Ref	Symbol	Parameter	8207-16, -8		8207-10		Units	Notes
			Min	Max	Min	Max		
RAM INTERFACE (Continued)								
32	TCAH	Column A0 to $\overline{CAS}$ Hold	(See DRAM Interface Tables)					21
33	TCLCSL	$\overline{CAS}$ ↓ from CLK ↓ Delay	TCLCL/4+30	TCLCL/1.8+53	TCLCL/4+30	100	ns	11, 12
34	TCLCSL	$\overline{CAS}$ ↓ from CLK ↓ Delay		35		40	ns	1
35	TCLCSH	$\overline{CAS}$ ↑ from CLK ↓ Delay		50		50	ns	
36	TCLW	WE from CLK ↓ Delay		35		35	ns	
37	TCLTKL	$\overline{XACK}$ ↓ from CLK ↓ Delay		35		35	ns	
38	TRWLTKH	$\overline{XACK}$ ↑ from $\overline{RD}$ ↑, $\overline{WR}$ ↑ Delay		50		50	ns	
39	TCLAKL	$\overline{AACK}$ ↓ from $\overline{CLK}$ ↓ Delay		35		35	ns	
40	TCLAKH	$\overline{AACK}$ ↑ from CLK ↓ Delay		50		50	ns	
41	TCLDL	$\overline{DBM}$ from CLK ↓ Delay		35		35	ns	
ECC INTERFACE								
42	TWRLFV	$\overline{FWR}$ from CFS=1 $\overline{WR}$ ↓ Delay CFS=0		2TCLCL−40 TCLCL+TCL−40		100	ns ns	1, 22 2, 22
43	TFVCL	$\overline{FWR}$ to CLK ↓ Setup	40		30		ns	23
44	TCLFX	$\overline{FWR}$ to CLK ↓ Hold	0		0		ns	24
45	TEVCL	$\overline{ERROR}$ to CLK ↓ Setup	20		20		ns	25, 26
46	TCLEX	$\overline{ERROR}$ to CLK ↓ Hold	0		0		ns	
47	TCLRL	R/$\overline{W}$ ↓ from CLK ↓ Delay		40		40	ns	
48	TCLRH	R/$\overline{W}$ ↑ from CLK ↓ Delay		50		50	ns	
49	TCEVCL	CE to CLK ↓ Setup	20		20		ns	25, 27
50	TCLCEX	CE to CLK ↓ Hold	0		0		ns	
51	TCLES	$\overline{ESTB}$ from CLK ↓ Delay		35		45	ns	

A.C. CHARACTERISTICS (Continued)

V_{CC} = 5V ±10% for 8207-8; T_A = 0°C to 70°C; V_{CC} = +5V ±5% for 8207-16

Measurements made with respect to RAS_{0-3}, CAS_{0-3}, AO_{0-8}, are a +2.4V and 0.8V. All other pins are measured at 2.0V and 0.8V. All times are ns unless otherwise indicated. Testing done with specified test load.

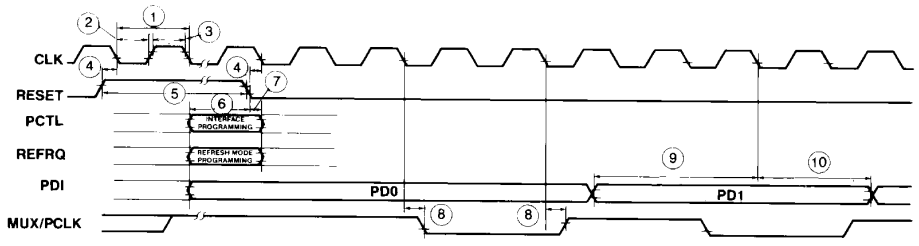
Ref	Symbol	Parameter	8207-16, -8		8207-10		Units	Notes
			Min	Max	Min	Max		
PORT SWITCHING AND LOCK								
52	TCLMV	MUX from CLK ↓ Delay		45		45	ns	
53	TCLPNV	PSEN from CLK ↓ Delay	TCL	TCL+35	TCL	TCL+35	ns	28
54	TCLPSV	PSEL from CLK ↓		35		35	ns	
55	TLKVCL	LOCK to CLK ↓ Setup	30		30		ns	30, 31
56	TCLLKX	LOCK to CLK ↓ Hold	10		10		ns	30, 31
57	TRWLLKV	LOCK from $\overline{RD}$ ↓, $\overline{WR}$ ↓ Delay		2TCLCL−30		2TCLCL−30	ns	31, 32
58	TRWHLKX	LOCK to $\overline{RD}$ ↓, $\overline{WR}$ ↓ Hold	3TCLCL+30		3TCLCL+30		ns	31, 32
REFRESH REQUEST								
59	TRFVCL	RFRQ to CLK ↓ Setup	20		20		ns	
60	TCLRFX	RFRQ to CLK ↓ Hold	10		10		ns	
61	TFRFH	Failsafe RFRQ Pulse Width	TCLCL+30		TCLCL+30		ns	33
62	TRFXCL	Single RFRQ Inactive to CLK ↓ Setup	20		20		ns	34
63	TBRFH	Burst RFRQ Pulse Width	2TCLCL+30		2TCLCL+30		ns	33

NOTES:
1. Specification when programmed in the Fast Cycle processor mode (iAPX 286 mode).
2. Specification when programmed in the Slow Cycle processor mode (iAPX 186 mode).
3. tR and tF are referenced from the 3.5V and 1.0V levels.
4. RESET is internally synchronized to CLK. Hence a set-up time is required only to guarantee its recognition at a particular clock edge.
5. The first programming bit (PD0) is also sampled by RESET going low.
6. TCLPDX is guaranteed if programming data is shifted using PCLK.
7. WZ is issued only in ECC mode.
8. TRWVCL is not required for an asynchronous command except to guarantee its recognition at a particular clock edge.
9. Valid when programmed in either Fast or Slow Cycle mode.
10. tASR is a user specified parameter and its value should be added accordingly to TAVCL.
11. When programmed in Slow Cycle mode and 125 ns $\leq$ TCLCL $<$ 200 ns.
12. When programmed in Slow Cycle mode and 200 ns $\leq$ TCLCL.
13. Specification for Test Load conditions.
14. tRCD (actual) = tRCD (specification) + 0.06 (ΔC_{RAS}) − 0.6 (ΔC_{CAS}) where ΔC = C (test load) − C (actual) in pF (These are first order approximations).
15. tRAH (actual) = tRAH (specification) + 0.06 (ΔC_{RAS}) − 0.022 (ΔC_{A0}) where ΔC = C (test load) − C (actual) in pF. (These are first order approximations.)
18. tASR (actual) = tASR (specification) + 0.06 (ΔC_{A0}) − 0.025 (ΔC_{RAS}) where ΔC = C (test load) − C (actual) in pF. (These are first order approximations.)
19. tASC (actual) = tASC (specification) + 0.06 (ΔC_{A0}) − 0.025 (ΔC_{CAS}) where ΔC = C (test load) − C (actual) in pF. (These are first order approximations.)

20. tASC is a function of clock frequency and thus varies with changes in frequency. A minimum value is specified.
21. See 8207 DRAM Interface Tables 14–18.
22. TWRLFV is defined for both synchronous and asynchronous $\overline{FWR}$. In systems in which $\overline{FWR}$ is decoded directly from the address inputs to the 8207, TCLFV is automatically guaranteed by TCLAV.
23. TFVCL is defined for synchronous $\overline{FWR}$.
24. TCLFV is defined for both synchronous and asynchronous $\overline{FWR}$. In systems in which $\overline{FWR}$ is decoded directly from the address inputs to the 8207 TCLFV is automatically guaranteed by TCLAV.
25. $\overline{ERROR}$ and CE are set-up to CLK ↓ in fast cycle mode and CLK ↑ in slow cycle mode.
26. $\overline{ERROR}$ is set-up to the same edge as R/$\overline{W}$ is referenced to, in RMW cycles.
27. CE is set-up to the same edge as WE is referenced to in RMW cycles.
28. Specification when TCL < 25 ns.
29. Synchronous operation only. Must arrive by the second clock falling edge after the clock edge which recognizes the command in order to be effective.
30. LOCK must be held active for the entire period the opposite port must be locked out. One clock after the release of LOCK the opposite port will be able to obtain access to memory.
31. Asynchronous mode only. In this mode a synchronizer stage is used internally in the 8207 to synchronize up LOCK. TRWLLKV and TRWHLKX are only required for guaranteeing that LOCK will be recognized for the requesting port, but these parameters are not required for correct 8207 operation.
32. TFRFH and TBRFH pertain to asynchronous operation only.
33. Single RFRQ cannot be supplied asynchronously.

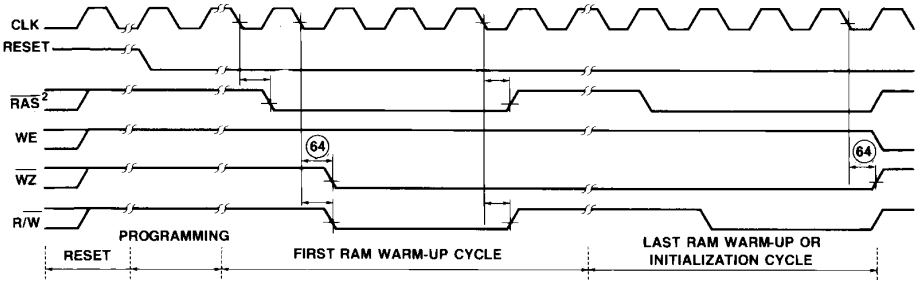
WAVEFORMS

CLOCK AND PROGRAMMING TIMINGS



210463–28

RAM WARM-UP AND MEMORY INITIALIZATION CYCLES

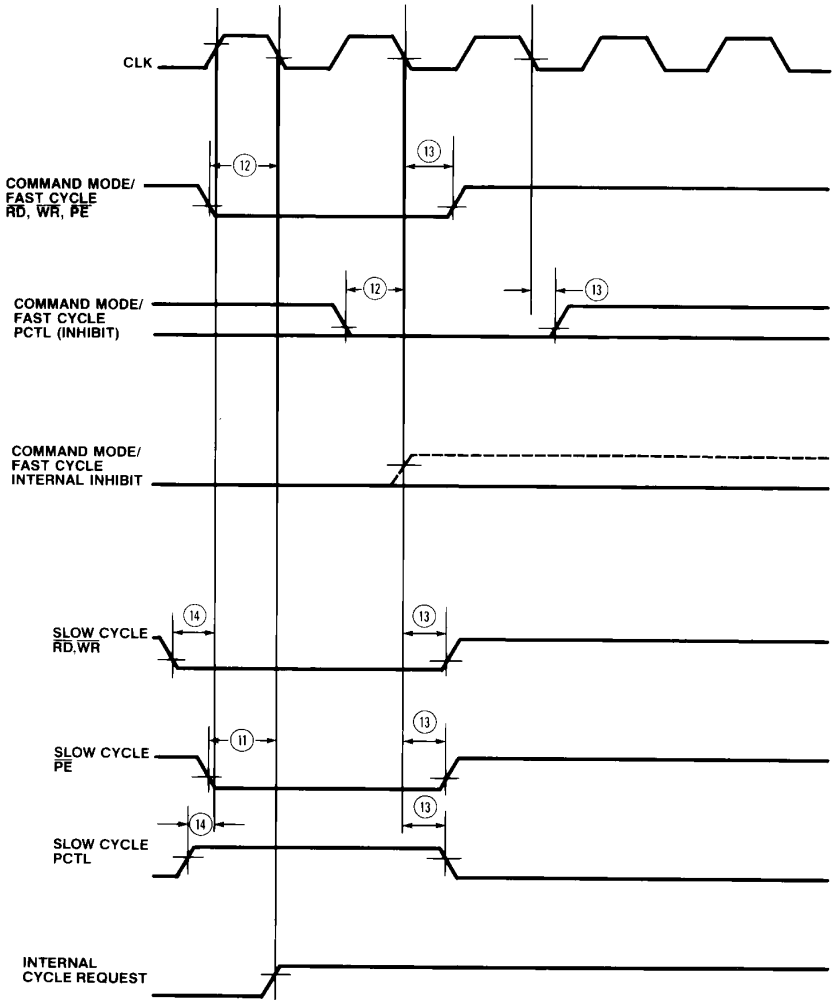


210463–29

NOTES:
1. When in non-ECC mode or in ECC mode with the TM2 programming bit on, there are no initialization cycles, when in ECC mode with TM2 off, the dummy cycles are followed by initialization cycles.
2. The present example assumes a RAS four clocks long.

WAVEFORMS (Continued)

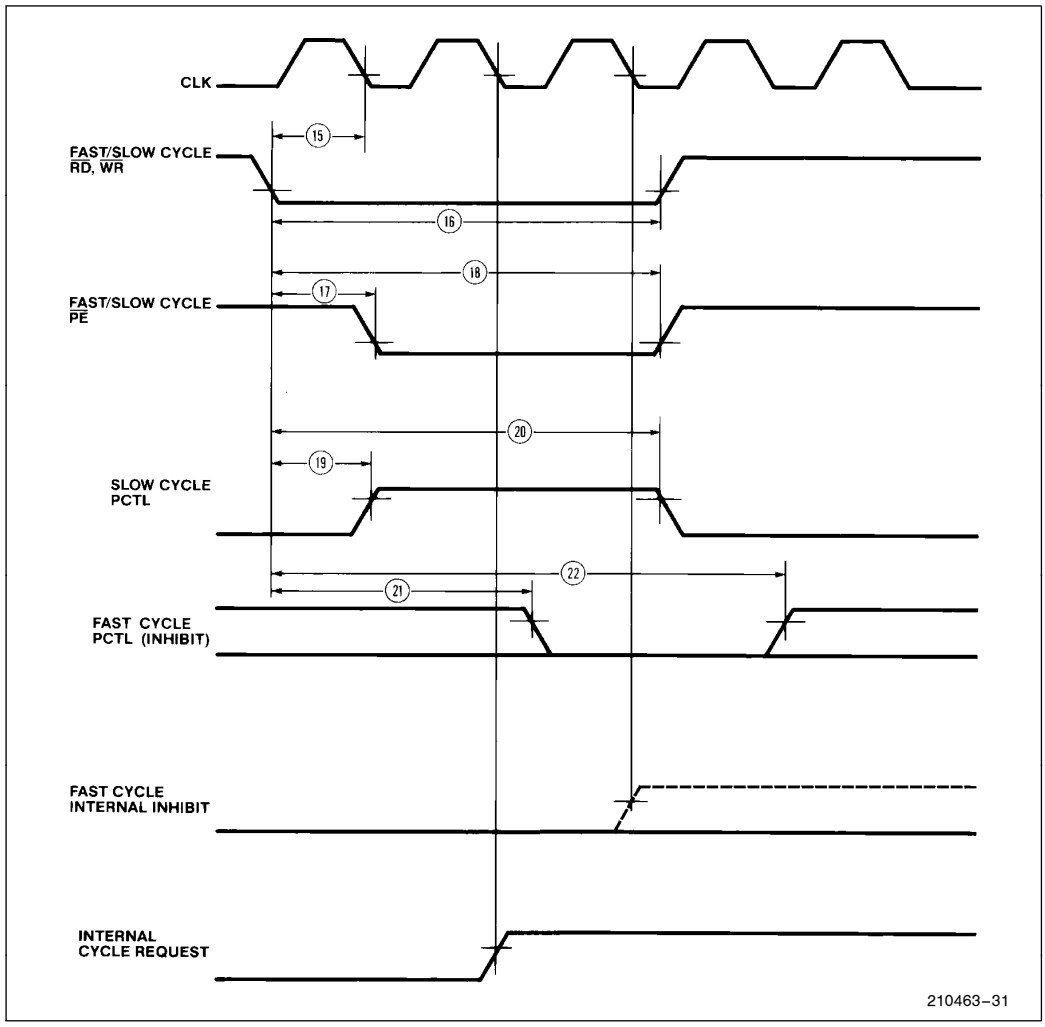
SYNCHRONOUS PORT INTERFACE



NOTE:
Actual transitions are programmable. Refer to Tables 12 and 13.

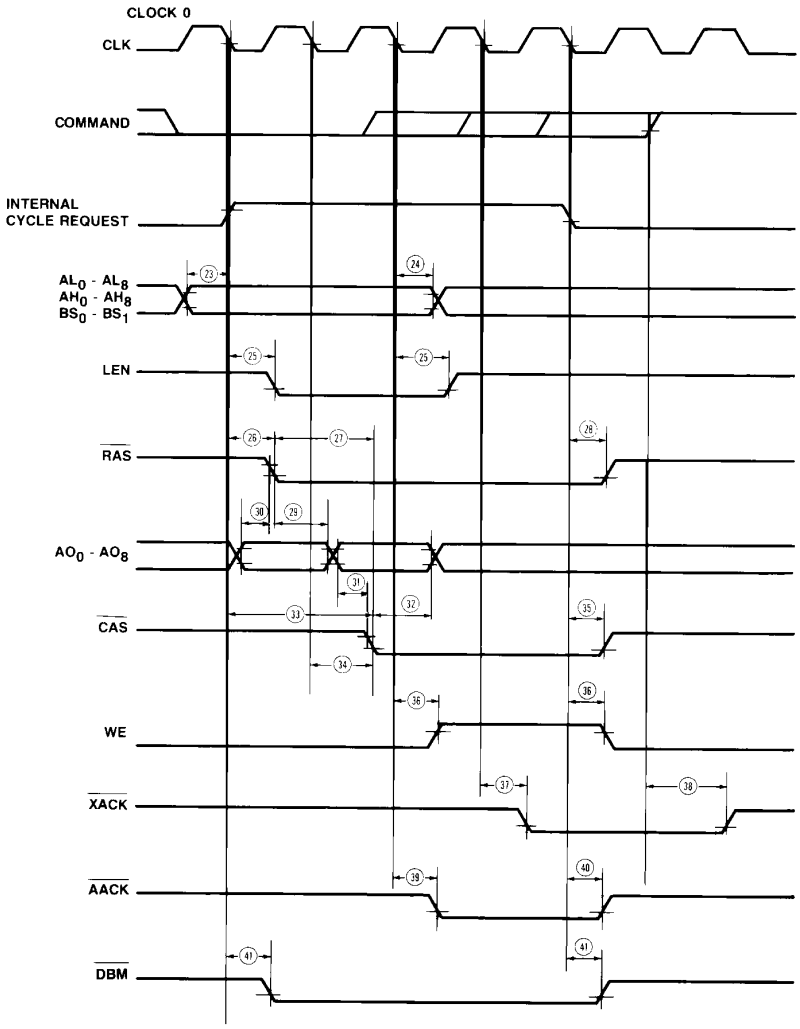
WAVEFORMS (Continued)

ASYNCHRONOUS PORT INTERFACE



WAVEFORMS (Continued)

RAM INTERFACE TIMING
ECC AND NON-ECC MODE

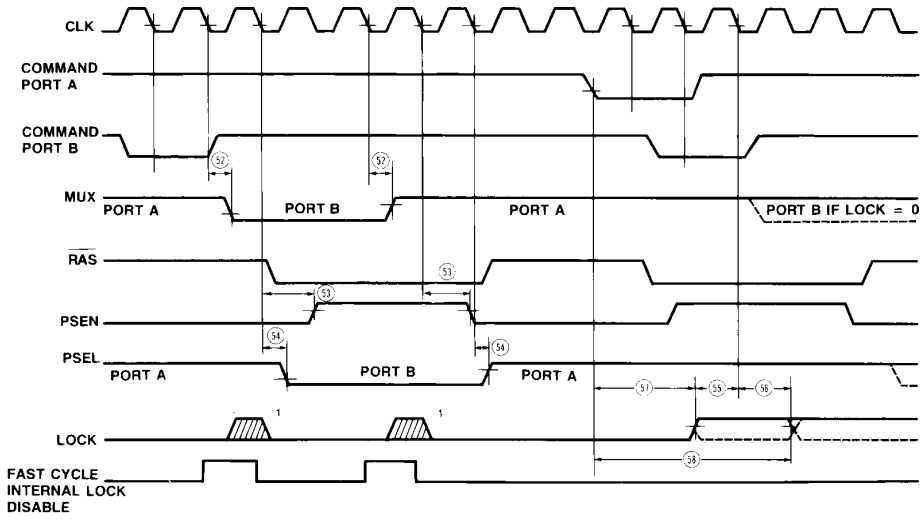


210463–32

NOTE:
Actual transitions are programmable. Refer to Tables 12 and 13.

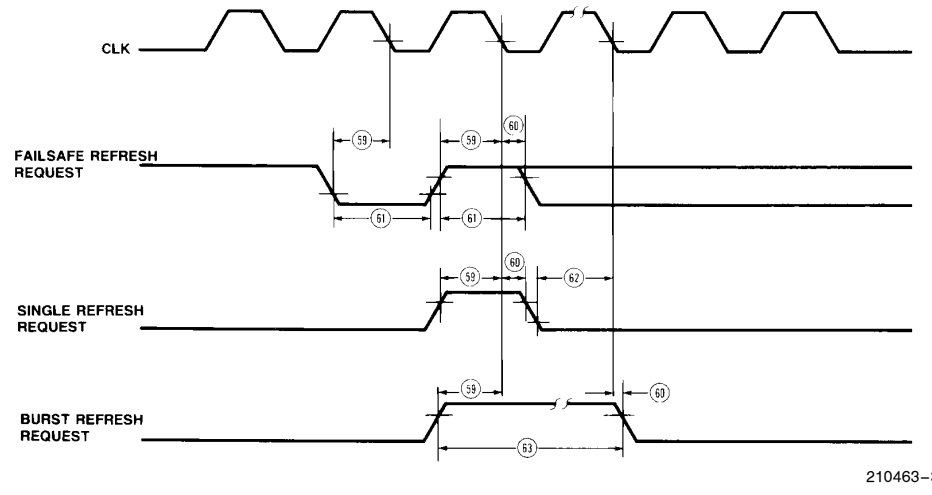
WAVEFORMS (Continued)

PORT SWITCHING AND LOCK TIMING



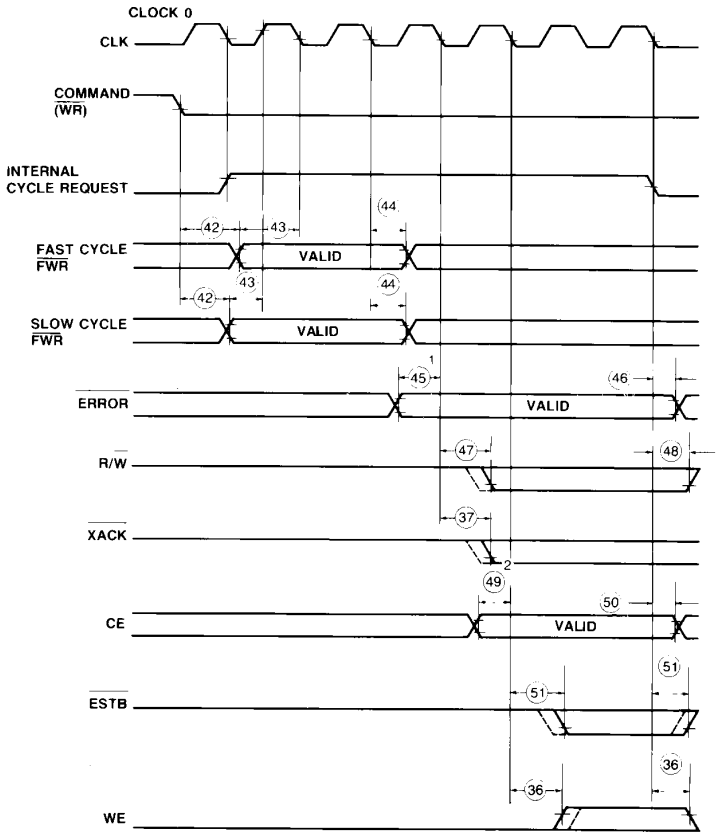
NOTE:
Transients during MUX switching.

REFRESH REQUEST TIMING



WAVEFORMS (Continued)

ECC INTERFACE TIMING



210463–35

NOTES:
1. This parameter is set-up to the falling edge of clock, as shown, for fast cycle configurations. It is set-up to the rising edge of clock if in slow cycle configurations. Table 13A shows which clock and clock edge these signals are set-up in the R/$\overline{W}$ L column.
2. CE is set-up to the same edge as WE is referenced to in RMW cycles.

CONFIGURATION TIMING CHARTS

The timing charts that follow are based on 8 basic system configurations where the 8207 operates.

Tables 10 and 11 give a description of non-ECC and ECC system configurations based on the 8207's PD0, PD3, PD4, PD10 and PD11 programming bits.

Table 10. Non-ECC System Configurations

Non-ECC Mode: PD0 = 0

Timing Conf.	$\overline{CFS}$(PD3)	$\overline{RFS}$(PD4)	EXT(PD10)	$\overline{FFS}$(PD11)
C_0	iAPX286(0)	Fast RAM(0)	Not EXT(0)	12 MHz(1)
C_0	iAPX286(0)	Fast RAM(0)	EXT(1)	12 MHz(1)
C_0	iAPX286(0)	Slow RAM(1)	Not EXT(0)	12 MHz(1)
C_0	iAPX286(0)	Slow RAM(1)	EXT(1)	12 MHz(1)
C_0	iAPX286(0)	Fast RAM(0)	Not EXT(0)	16 MHz(0)
C_1	iAPX286(0)	Slow RAM(1)	Not EXT(0)	16 MHz(0)
C_1	iAPX286(0)	Fast RAM(0)	EXT(1)	16 MHz(0)
C_2	iAPX286(0)	Slow RAM(1)	EXT(1)	16 MHz(0)
C_3	iAPX186(1)	Fast RAM(0)	Not EXT(0)	10, 8 MHz(0)
C_3	iAPX186(1)	Slow RAM(1)	Not EXT(0)	10, 8 MHz(0)
C_3	iAPX186(1)	Fast RAM(0)	EXT(1)	10, 8 MHz(0)
C_3	iAPX186(1)	Fast RAM(0)	Not EXT(0)	6 MHz(1)
C_3	iAPX186(1)	Fast RAM(0)	EXT(1)	6 MHz(1)
C_3	iAPX186(1)	Slow RAM(1)	Not EXT(0)	6 MHz(1)
C_3	iAPX186(1)	Slow RAM(1)	EXT(1)	6 MHz(1)
C_4	iAPX186(1)	Slow RAM(1)	EXT(1)	10, 8 MHz(0)

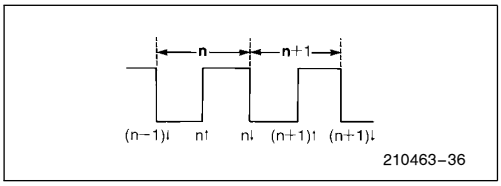
Table 11. ECC System Configurations

ECC Mode: PD0 = 1

Timing Conf.	CFS(PD3)	RFS(PD4)	$\overline{EXT}$(PD10)	FFS(PD11)
C_0	iAPX286(1)	Slow RAM(0)	M/S EDCU(0)	12 MHz(0)
C_0	iAPX286(1)	Slow RAM(0)	M EDCU(1)	12 MHz(0)
C_0	iAPX286(1)	Fast RAM(1)	M/S EDCU(0)	12 MHz(0)
C_0	iAPX286(1)	Fast RAM(1)	M EDCU(1)	12 MHz(0)
C_0	iAPX286(1)	Fast RAM(1)	M EDCU(1)	16 MHz(1)
C_1	iAPX286(1)	Slow RAM(0)	M EDCU(1)	16 MHz(1)
C_2	iAPX286(1)	Fast RAM(1)	M/S EDCU(0)	16 MHz(1)
C_3	iAPX286(1)	Slow RAM(0)	M/S EDCU(0)	16 MHz(1)
C_4	iAPX186(0)	Slow RAM(0)	M/S EDCU(0)	6 MHz(0)
C_4	iAPX186(0)	Fast RAM(1)	M/S EDCU(0)	6 MHz(0)
C_4	iAPX186(0)	Slow RAM(0)	M EDCU(1)	10, 8 MHz(1)
C_4	iAPX186(0)	Fast RAM(1)	M EDCU(1)	10, 8 MHz(1)
C_5	iAPX186(0)	Slow RAM(0)	M/S EDCU(0)	10, 8 MHz(1)
C_5	iAPX186(0)	Fast RAM(1)	M/S EDCU(0)	10, 8 MHz(1)
C_6	iAPX186(0)	Slow RAM(0)	M EDCU(1)	6 MHz(0)
C_6	iAPX186(0)	Fast RAM(1)	M EDCU(1)	6 MHz(0)

Using the Timing Charts

The notation used to indicate which clock edge triggers an output transition is ''n ↑'' or ''n ↓'', where ''n'' is the number of clock periods that have passed since clock 0, the reference clock, and '' ↑'' refers to rising edge and '' ↓'' to falling edge. A clock period is defined as the interval from a clock falling edge to the following falling edge. Clock edges are defined as shown below.



The clock edges which trigger transitions on each 8207 output are tabulated in Table 12 for non-ECC mode and Table 13 for ECC mode. ''H'' refers to the high-going transition, and ''L'' to low-going transition; ''V'' refers to valid, and ''$\overline{V}$'' to non-valid.

Clock 0 is defined as the clock in which the 8207 begins a memory cycle, either as a result of a port request which has just arrived, or of a port request which was stored previously but could not be serviced at the time of its arrival because the 8207 was performing another memory cycle. Clock 0 may be identified externally by the leading edge of RAS, which is always triggered on 0 ↓.

Notes for interpreting the timing charts:

1. **PSEL - valid** is given as the latest time it can occur. It is entirely possible for PSEL to become valid before the time given in a refresh cycle. PSEL can switch as defined in the chart, but it has no bearing on the refresh cycle itself, but only on a subsequent cycle for one of the external ports.
2. **LEN - low** is given as the latest time it can occur. LEN is only activated by port A configured in Fast Cycle iAPX286 mode, and thus it is not activated by a refresh cycle, although it may be activated by port A during a refresh cycle.
3. **ADDRESS - col** is the time column address becomes valid.
4. In non-ECC mode the $\overline{CAS}$, $\overline{EAACK}$, $\overline{LAACK}$ and $\overline{XACK}$ outputs are not issued during refresh.
5. In ECC mode there are really seven types of cycles: Read without error, read with error, full write, partial write without error, partial write with error, refresh without error, and refresh with error. These cycles may be derived from the timing chart as follows:
   A. Read without error: Use row marked 'RD, RF'.
   B. Read with error: Use row marked 'RMW', except for $\overline{EAACK}$ and $\overline{LAACK}$, which should be taken from 'RD, RF'. If the error is uncorrectable. WE will not be issued.
   C. Full write: Use row marked 'WR'.
   D. Partial write without error: Use row marked 'RMW', except that $\overline{DBM}$ and $\overline{ESTB}$ will not be issued.
   E. Partial write with error: Use row marked 'RMW', except that $\overline{DBM}$ will not be issued. If the error is uncorrectable, WE will not be issued.
   F. Refresh without error: Use row marked 'RD, RF', except that $\overline{ESTB}$, $\overline{EAACK}$, $\overline{LAACK}$, and $\overline{XACK}$ will not be issued.
   G. Refresh with error: Use row marked 'RMW', except that $\overline{EAACK}$, $\overline{LAACK}$, $\overline{ESTB}$, and $\overline{XACK}$ will not be issued. If the error is uncorrectable WE will not be issued.
6. **XACK - high** is reset asynchronously by command going inactive and not by a clock edge.
7. **MUX - valid** is given as the latest time it can occur.

Table 12A. Timing Chart—Non-ECC Mode

		PSEN		PSEL		$\overline{DBM}$		LEN		$\overline{RAS}$		$\overline{CAS}$		WE	
C_n	Cycle	H	L	V	$\overline{V}$	L	H	L	H	L	H	L	H	H	L
C_0	RD, RF	0↓	3↓	0↓	4↓	0↓	4↓	0↓	2↓	0↓	3↓	1↓	4↓		
	WR	0↓	4↓	0↓	5↓			0↓	2↓	0↓	5↓	1↓	5↓	2↓	5↓
C_1	RD, RF	0↓	5↓	0↓	6↓	0↓	6↓	0↓	2↓	0↓	4↓	1↓	6↓		
	WR	0↓	4↓	0↓	5↓			0↓	2↓	0↓	5↓	1↓	5↓	2↓	5↓
C_2	RD, RF	0↓	5↓	0↓	6↓	0↓	6↓	0↓	2↓	0↓	4↓	1↓	6↓		
	WR	0↓	4↓	0↓	5↓			0↓	2↓	0↓	5↓	1↓	5↓	2↓	5↓
C_3	RD, RF	0↓	2↓	0↓	3↓	0↓	3↓			0↓	3↓	0↓	3↓		
	WR	0↓	3↓	0↓	4↓					0↓	4↓	0↓	4↓	2↑	4↓
C_4	RD, RF	0↓	3↓	0↓	4↓	0↓	4↓			0↓	4↓	0↓	4↓		
	WR	0↓	3↓	0↓	4↓					0↓	4↓	0↓	4↓	2↑	4↓

Table 12B. Timing Chart—Non-ECC Mode

		Col Addr		$\overline{EAACK}$		$\overline{LAACK}$		$\overline{XACK}$		MUX	
C_n	Cycle	V	$\overline{V}$	L	H	L	H	L	H	V	$\overline{V}$
C_0	RD, RF	0↓	2↓	1↓	4↓	2↓	5↓	3↓	$\overline{RD}$	−2↓	2↓
	WR	0↓	2↓	1↓	4↓	1↓	4↓	3↓	$\overline{WR}$	−2↓	2↓
C_1	RD, RF	0↓	3↓	2↓	5↓	2↓	5↓	4↓	$\overline{RD}$	−2↓	2↓
	WR	0↓	3↓	1↓	4↓	1↓	4↓	3↓	$\overline{WR}$	−2↓	2↓
C_2	RD, RF	0↓	3↓	2↓	5↓	3↓	6↓	4↓	$\overline{RD}$	−2↓	2↓
	WR	0↓	3↓	1↓	4↓	1↓	4↓	3↓	$\overline{WR}$	−2↓	2↓
C_3	RD, RF	0↓	2↓	0↓	2↓	1↓	3↓	2↓	$\overline{RD}$	−1↓	2↓
	WR	0↓	2↓	0↓	2↓	1↑	3↑	2↓	$\overline{WR}$	−1↓	2↓
C_4	RD, RF	0↓	2↓	1↓	3↓	1↓	3↓	3↑	$\overline{RD}$	−1↓	2↓
	WR	0↓	2↓	0↓	2↓	1↑	3↑	2↓	$\overline{WR}$	−1↓	2↓

Table 13A. Timing Chart—ECC Mode

		PSEN		PSEL		$\overline{DBM}$		$\overline{LEN}$		$\overline{RAS}$		$\overline{CAS}$		R/$\overline{W}$		WE	
C_n	Cycle	H	L	V	$\overline{V}$	L	H	L	H	L	H	L	H	L	H	H	L
C_0	RD, RF	0↓	5↓	0↓	6↓	0↓	6↓	0↓	2↓	0↓	4↓	1↓	6↓				
	WR	0↓	5↓	0↓	6↓			0↓	2↓	0↓	6↓	1↓	6↓	1↓	6↓	3↓	6↓
	RMW	0↓	8↓	0↓	9↓	0↓	9↓	0↓	2↓	0↓	9↓	1↓	9↓	4↓	9↓	6↓	9↓
C_1	RD, RF	0↓	5↓	0↓	6↓	0↓	6↓	0↓	2↓	0↓	4↓	1↓	6↓				
	WR	0↓	5↓	0↓	6↓			0↓	2↓	0↓	6↓	1↓	6↓	1↓	6↓	3↓	6↓
	RMW	0↓	8↓	0↓	9↓	0↓	9↓	0↓	2↓	0↓	9↓	1↓	9↓	4↓	9↓	6↓	9↓
C_2	RD, RF	0↓	6↓	0↓	7↓	0↓	7↓	0↓	2↓	0↓	5↓	1↓	7↓				
	WR	0↓	6↓	0↓	7↓			0↓	2↓	0↓	7↓	1↓	7↓	1↓	7↓	4↓	7↓
	RMW	0↓	10↓	0↓	11↓	0↓	11↓	0↓	2↓	0↓	11↓	1↓	11↓	5↓	11↓	8↓	11↓
C_3	RD, RF	0↓	6↓	0↓	7↓	0↓	7↓	0↓	2↓	0↓	5↓	1↓	7↓				
	WR	0↓	6↓	0↓	7↓			0↓	2↓	0↓	7↓	1↓	7↓	1↓	7↓	4↓	7↓
	RMW	0↓	10↓	0↓	11↓	0↓	11↓	0↓	2↓	0↓	11↓	1↓	11↓	5↓	11↓	8↓	11↓
C_4	RD, RF	0↓	3↓	0↓	4↓	0↓	4↓			0↓	3↓	0↓	4↓				
	WR	0↓	4↓	0↓	5↓					0↓	5↓	0↓	5↓	1↑	5↓	3↑	5↓
	RMW	0↓	6↓	0↓	7↓	0↓	7↓			0↓	7↓	0↓	7↓	3↑	7↓	5↑	7↓
C_5	RD, RF	0↓	3↓	0↓	4↓	0↓	4↓			0↓	3↓	0↓	4↓				
	WR	0↓	4↓	0↓	5↓					0↓	5↓	0↓	5↓	1↑	5↓	3↑	5↓
	RMW	0↓	6↓	0↓	7↓	0↓	7↓			0↓	7↓	0↓	7↓	3↑	7↓	5↑	7↓
C_6	RD, RF	0↓	3↓	0↓	4↓	0↓	4↓			0↓	3↓	0↓	4↓				
	WR	0↓	3↓	0↓	4↓					0↓	4↓	0↓	4↓	1↑	4↓	2↑	4↓
	RMW	0↓	4↓	0↓	5↓	0↓	5↓			0↓	5↓	0↓	5↓	2↑	5↓	3↑	5↓

Table 13B. Timing Chart—ECC Mode

		Col Addr		$\overline{\text{ESTB}}$		$\overline{\text{EAACK}}$		$\overline{\text{LAACK}}$		$\overline{\text{XACK}}$		$\overline{\text{MUX}}$	
C_n	Cycle	V	$\overline{\text{V}}$	L	H	L	H	L	H	L	H	V	$\overline{\text{V}}$
C_0	RD, RF	0↓	2↓			2↓	5↓	3↓	6↓	4↓	RD	−2↓	2↓
	WR	0↓	2↓			2↓	5↓	2↓	5↓	4↓	WR	−2↓	2↓
	RMW	0↓	2↓	6↓	8↓	5↓	8↓	5↓	8↓	7↓	WR	−2↓	2↓
C_1	RD, RF	0↓	3↓			3↓	6↓	3↓	6↓	4↓	RD	−2↓	2↓
	WR	0↓	3↓			2↓	5↓	2↓	5↓	4↓	WR	−2↓	2↓
	RMW	0↓	3↓	6↓	8↓	5↓	8↓	5↓	8↓	7↓	WR	−2↓	2↓
C_2	RD, RF	0↓	3↓			4↓	7↓	4↓	7↓	5↓	RD	−2↓	2↓
	WR	0↓	3↓			3↓	6↓	3↓	6↓	5↓	WR	−2↓	2↓
	RMW	0↓	3↓	8↓	10↓	7↓	10↓	7↓	10↓	9↓	WR	−2↓	2↓
C_3	RD, RF	0↓	3↓			4↓	7↓	5↓	8↓	5↓	RD	−2↓	2↓
	WR	0↓	3↓			3↓	6↓	3↓	6↓	5↓	WR	−2↓	2↓
	RMW	0↓	3↓	8↓	10↓	7↓	10↓	7↓	10↓	9↓	WR	−2↓	2↓
C_4	RD, RF	0↓	2↓			1↓	3↓	2↑	4↑	3↑	RD	−1↓	2↓
	WR	0↓	2↓			1↓	3↓	2↑	4↑	3↓	WR	−1↓	2↓
	RMW	0↓	2↓	5↑	6↑	3↓	5↓	4↑	6↑	5↓	WR	−1↓	2↓
C_5	RD, RF	0↓	2↓			2↓	4↓	3↑	5↑	3↑	RD	−1↓	2↓
	WR	0↓	2↓			1↓	3↓	2↑	4↑	3↓	WR	−1↓	2↓
	RMW	0↓	2↓	5↑	6↑	3↓	5↓	4↑	6↑	5↓	WR	−1↓	2↓
C_2	RD, RF	0↓	2↓			1↓	3↓	1↑	3↑	2↑	RD	−1↓	2↓
	WR	0↓	2↓			1↓	3↓	1↑	3↑	2↓	WR	−1↓	2↓
	RMW	0↓	2↓	3↑	4↑	1↓	3↓	2↑	4↑	3↓	WR	−1↓	2↓

8207—DRAM Interface Parameter Equations

Several DRAM parameters, but not all, are a direct function of 8207 timings, and the equations for these parameters are given in the following tables. The following is a list of those DRAM parameters which have NOT been included in the following tables, with an explanation for their exclusion.

READ, WRITE, READ-MODIFY-WRITE & REFRESH CYCLES

tRAC: response parameter.

tCAC: response parameter.

tREF: See ''Refresh Period Options''

tCRP: must be met only if $\overline{CAS}$-only cycles, which do not occur with 8207, exist.

tRAH: See ''A.C. Characteristics''

tRCD: See ''A.C. Characteristics''

tASC: See ''A.C. Characteristics''

tASR: See ''A.C. Characteristics''

tOFF: response parameter.

READ & REFRESH CYCLES

tRCH: WE always goes active after $\overline{CAS}$ goes active, hence tRCH is guaranteed by tCPN.

WRITE CYCLE

tRC: guaranteed by tRWC.

tRAS: guaranteed by tRRW.

tCAS: guaranteed by tCRW.

tWCS: WE always activated after $\overline{CAS}$ is activated, except in memory initialization, hence tWCS is always negative (this is important for RMW only) except in memory initialization; in memory initialization tWCS is positive and has several clocks of margin.

tDS: system-dependent parameter.

tDH: system-dependent parameter.

tDHR: system-dependent parameter.

READ-MODIFY-WRITE CYCLE

tRWD: don't care in 8207 write cycles, but tabulated for 8207 RMW cycles.

tCWD: don't care in 8207 write cycles, but tabulated for 8207 RMW cycles.

Table 14. Non-ECC Mode—RD, RF Cycles

Parameter	Fast Cycle Configurations			Slow Cycle Configurations		Notes
	C_0	C_1	C_2	C_3	C_4	
tRP	3TCLCL−T26	4TCLCL−T26	4TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tCPN	3TCLCL−T35	3TCLCL−T35	3TCLCL−T35	2.5TCLCL−T35	2.5TCLCL−T35	1
tRSH	2TCLCL−T34	3TCLCL−T34	3TCLCL−T34	3TCLCL−T34	4TCLCL−T34	1
tCSH	4TCLCL−T26	6TCLCL−T26	6TCLCL−T26	3TCLCL−T26	4TCLCL−T26	1
tCAH	TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	3TCLCL−T26	3TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tT	3/30	3/30	3/30	3/30	3/30	2
tRC	6TCLCL	8TCLCL	8TCLCL	5TCLCL	6TCLCL	1
tRAS	3TCLCL−T26	4TCLCL−T26	4TCLCL−T26	3TCLCL−T26	4TCLCL−T26	1
tCAS	3TCLCL−T34	5TCLCL−T34	5TCLCL−T34	3TCLCL−T34	4TCLCL−T34	1
tRCS	2TCLCL−TCL −T36−TBUF	2TCLCL−TCL −T36−TBUF	2TCLCL−TCL −T36−TBUF	1.5TCLCL−TCL −T36−TBUF	1.5TCLCL−TCL −T36−TBUF	1

Table 15. Non-ECC Mode—WR Cycle

Parameter	Fast Cycle Configurations			Slow Cycle Configurations		Notes
	C_0	C_1	C_2	C_3	C_4	
tRP	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tCPN	4TCLCL−T35	4TCLCL−T35	4TCLCL−T35	2.5TCLCL−T35	2.5TCLCL−T35	1
tRSH	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	1
tCSH	5TCLCL−T26	5TCLCL−T26	5TCLCL−T26	4TCLCL−T26	4TCLCL−T26	1
tCAH	TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	3TCLCL−T26	3TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tT	3/30	3/30	3/30	3/30	3/30	2
tRWC	8TCLCL	8TCLCL	8TCLCL	6TCLCL	6TCLCL	1
tRRW	5TCLCL−T26	5TCLCL−T26	5TCLCL−T26	4TCLCL−T26	4TCLCL−T26	1
tCRW	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	1
tWCH	3TCLCL+TCL−T34	3TCLCL+TCL−T34	3TCLCL+TCL−T34	3TCLCL+TCL−T34	3TCLCL+TCL−T34	1, 3
tWCR	4TCLCL+TCL−T26	4TCLCL+TCL−T26	4TCLCL+TCL−T26	3TCLCL+TCL−T26	3TCLCL+TCL−T26	1, 3
tWP	2TCLCL+TCL−T36−TBUF	2TCLCL+TCL−T36−TBUF	2TCLCL+TCL−T36−TBUF	2TCLCL−T36−TBUF	2TCLCL−T36−TBUF	1
tRWL	3TCLCL−T36−TBUF	3TCLCL−T36−TBUF	3TCLCL−T36−TBUF	3TCLCL−TCL−T36−TBUF	3TCLCL−TCL−T36−TBUF	1
tCWL	3TCLCL−T36−TBUF	3TCLCL−T36−TBUF	3TCLCL−T36−TBUF	3TCLCL−TCL−T36−TBUF	3TCLCL−TCL−T36−TBUF	1

Table 16A. ECC Mode—RD, RF Cycles

Parameter	Fast Cycle Mode				Notes
	C_0	C_1	C_2	C_3	
tRP	4TCLCL−T26	4TCLCL−T26	4TCLCL−T26	4TCLCL−T26	1
tCPN	3TCLCL−T35	3TCLCL−T35	3TCLCL−T35	3TCLCL−T35	1
tRSH	3TCLCL−T34	3TCLCL−T34	4TCLCL−T34	4TCLCL−T34	1
tCSH	6TCLCL−T26	6TCLCL−T26	7TCLCL−T26	7TCLCL−T26	1
tCAH	TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	1
tT	3/30	3/30	3/30	3/30	2
tRC	8TCLCL	8TCLCL	9TCLCL	9TCLCL	1
tRAS	4TCLCL−T26	4TCLCL−T26	5TCLCL−T26	5TCLCL−T26	1
tCAS	5TCLCL−T34	5TCLCL−T34	6TCLCL−T34	6TCLCL−T34	1
tRCS	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	1

Table 16B. ECC Mode—RD, RF Cycles

Parameter	Slow Cycle Mode			Notes
	C_4	C_5	C_6	
tRP	2TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tCPN	1.5TCLCL−T35	1.5TCLCL−T35	1.5TCLCL−T35	1
tRSH	3TCLCL−T34	3TCLCL−T34	3TCLCL−T34	1
tCSH	4TCLCL−T26	4TCLCL−T26	4TCLCL−T26	1
tCAH	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tT	3/30	3/30	3/30	2
tRC	5TCLCL	5TCLCL	5TCLCL	1
tRAS	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	1
tCAS	4TCLCL−T34	4TCLCL−T34	4TCLCL−T34	1
tRCS	0.5TCLCL−T36 −TBUF	0.5TCLCL−T36 −TBUF	0.5TCLCL−T36 −TBUF	1

Table 17A. ECC Mode—WR Cycle

Parameter	Fast Cycle Mode				Notes
	C_0	C_1	C_2	C_3	
tRP	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	1
tCPN	4TCLCL−T35	4TCLCL−T35	4TCLCL−T35	4TCLCL−T35	1
tRSH	5TCLCL−T34	5TCLCL−T34	6TCLCL−T34	6TCLCL−T34	1
tCSH	6TCLCL−T26	6TCLCL−T26	7TCLCL−T26	7TCLCL−T26	1
tCAH	TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	1
tT	3/30	3/30	3/30	3/30	2
tRWC	9TCLCL	9TCLCL	10TCLCL	10TCLCL	1
tRRW	6TCLCL−T26	6TCLCL−T26	7TCLCL−T26	7TCLCL−T26	1
tCRW	5TCLCL−T34	5TCLCL−T34	6TCLCL−T34	6TCLCL−T34	1
tWCH	5TCLCL−T34	5TCLCL−T34	6TCLCL−T34	6TCLCL−T34	1, 4
tWCR	6TCLCL−T26	6TCLCL−T26	7TCLCL−T26	7TCLCL−T26	1, 4
tWP	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	1
tRWL	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	1
tCWL	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	1

Table 17B. ECC Mode—WR Cycle

Parameter	Slow Cycle Mode			Notes
	C_4	C_5	C_6	
tRP	2TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tCPN	2.5TCLCL−T35	2.5TCLCL−T35	2.5TCLCL−T35	1
tRSH	5TCLCL−T34	5TCLCL−T34	4TCLCL−T34	1
tCSH	5TCLCL−T26	5TCLCL−T26	4TCLCL−T26	1
tCAH	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tT	3/30	3/30	3/30	2
tRWC	7TCLCL	7TCLCL	6TCLCL	1
tRRW	5TCLCL−T26	5TCLCL−T26	4TCLCL−T26	1
tCRW	5TCLCL−T34	5TCLCL−T34	4TCLCL−T34	1
tWCH	5TCLCL−T34	5TCLCL−T34	4TCLCL−T34	1, 4
tWCR	5TCLCL−T26	5TCLCL−T26	4TCLCL−T26	1, 4
tWP	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	1
tRWL	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	1
tCWL	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	1

Table 18A. ECC Mode—RMW

Parameter	Fast Cycle Mode				Notes
	C_0	C_1	C_2	C_3	
tRP	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	1
tCPN	4TCLCL−T35	4TCLCL−T35	4TCLCL−T35	4TCLCL−T35	1
tRSH	8TCLCL−T34	8TCLCL−T34	10TCLCL−T34	10TCLCL−T34	1
tCSH	9TCLCL−T26	9TCLCL−T26	11TCLCL−T26	11TCLCL−T26	1
tCAH	TCLCL−T34	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	3TCLCL−T26	3TCLCL−T26	3TCLCL−T26	1
tT	3/30	3/30	3/30	3/30	2
tRWC	12TCLCL	12TCLCL	14TCLCL	14TCLCL	1
tRRW	9TCLCL−T26	9TCLCL−T26	11TCLCL−T26	11TCLCL−T26	1
tCRW	8TCLCL−T34	8TCLCL−T34	10TCLCL−T34	10TCLCL−T34	1
tRCS	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	TCLCL−T36 −TBUF	1
tRWD	6TCLCL−T26	6TCLCL−T26	8TCLCL−T26	8TCLCL−T26	1, 4
tCWD	5TCLCL−T34	5TCLCL−T34	7TCLCL−T34	7TCLCL−T34	1
tWP	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	1
tRWL	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	1
tCWL	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	3TCLCL−T36 −TBUF	1

Table 18B. ECC Mode—RMW

Parameter	Slow Cycle Mode			Notes
	C_4	C_5	C_6	
tRP	2TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tCPN	2.5TCLCL−T35	2.5TCLCL−T35	2.5TCLCL−T35	1
tRSH	7TCLCL−T34	7TCLCL−T34	5TCLCL−T34	1
tCSH	7TCLCL−T26	7TCLCL−T26	5TCLCL−T26	1
tCAH	2TCLCL−T34	2TCLCL−T34	2TCLCL−T34	1
tAR	2TCLCL−T26	2TCLCL−T26	2TCLCL−T26	1
tT	3/30	3/30	3/30	2
tRWC	9TCLCL	9TCLCL	7TCLCL	1
tRRW	7TCLCL−T26	7TCLCL−T26	5TCLCL−T26	1
tCRW	7TCLCL−T34	7TCLCL−T34	5TCLCL−T34	1
tRCS	0.5TCLCL−T36 −TBUF	0.5TCLCL−T36 −TBUF	0.5TCLCL−T36 −TBUF	1
tRWD	4TCLCL+TCL −T26	4TCLCL+TCL −T26	2TCLCL+TCL −T26	1
tCWD	4TCLCL+TCL −T34	4TCLCL+TCL −T34	2TCLCL+TCL −T34	1
tWP	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	1
tRWL	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	1
tCWL	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	3TCLCL−TCL −T36−TBUF	1

NOTES:
1. Minimum.
2. Value on right is maximum; value on left is minimum.
3. Applies to the eight warm-up cycles during initialization only.
4. Applies to the eight warm-up cycles and to the memory initialization cycles during initialization only.
5. TP = TCLCL
   T26 = TCLRSL
   T34 = TCLCSL
   T35 = TCLCSH
   T36 = TCLW
   TBUF = TTL Buffer delay.

48