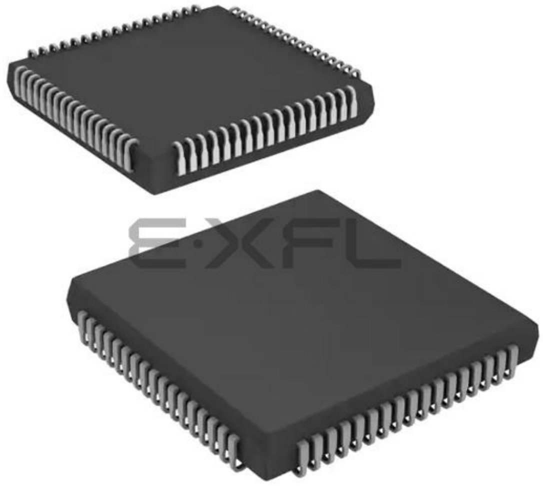
Renesas Electronics America Inc - IS80C286-20 Datasheet



Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Obsolete
Core Processor	80C286
Number of Cores/Bus Width	1 Core, 16-Bit
Speed	20MHz
Co-Processors/DSP	Math Engine; 80C287
RAM Controllers	-
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	-
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	-
Package / Case	68-LCC (J-Lead)
Supplier Device Package	68-PLCC (24.23x24.23)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/is80c286-20

Pin Descriptions

The following pin function descriptions are for the 80C286 microprocessor. **(Continued)**

SYMBOL	PIN NUMBER	TYPE	DESCRIPTION
$M/\overline{IO}$	67	O	MEMORY I/O SELECT: distinguishes memory access from I/O access. If HIGH during T_S, a memory cycle or a halt/shutdown cycle is in progress. If LOW, an I/O cycle or an interrupt acknowledge cycle is in progress. $M/\overline{IO}$ is held at high impedance to the last valid logic state during bus hold acknowledge.
$COD/\overline{INTA}$	66	O	CODE/INTERRUPT ACKNOWLEDGE: distinguishes instruction fetch cycles from memory data read cycles. Also distinguishes interrupt acknowledge cycles from I/O cycles. $COD/\overline{INTA}$ is held at high impedance to the last valid logic state during bus hold acknowledge. Its timing is the same as $M/\overline{IO}$.
$\overline{LOCK}$	68	O	BUS LOCK: indicates that other system bus masters are not to gain control of the system bus for the current and following bus cycles. The $\overline{LOCK}$ signal may be activated explicitly by the "LOCK" instruction prefix or automatically by 80C286 hardware during memory XCHG instructions, interrupt acknowledge, or descriptor table access. $\overline{LOCK}$ is active LOW and is held at a high impedance logic one during bus hold acknowledge.
$\overline{READY}$	63	I	BUS READY: terminates a bus cycle. Bus cycles are extended without limit until terminated by $\overline{READY}$ LOW. $\overline{READY}$ is an active LOW synchronous input requiring setup and hold times relative to the system clock be met for correct operation. $\overline{READY}$ is ignored during bus hold acknowledge. (See Note 1)
HOLD HLDA	64 65	I O	BUS HOLD REQUEST AND HOLD ACKNOWLEDGE: control ownership of the 80C286 local bus. The HOLD input allows another local bus master to request control of the local bus. When control is granted, the 80C286 will float its bus drivers and then activate HLDA, thus entering the bus hold acknowledge condition. The local bus will remain granted to the requesting master until HOLD becomes inactive which results in the 80C286 deactivating HLDA and regaining control of the local bus. This terminates the bus hold acknowledge condition. HOLD may be asynchronous to the system clock. These signals are active HIGH. Note that HLDA never floats.
INTR	57	I	INTERRUPT REQUEST: requires the 80C286 to suspend its current program execution and service a pending external request. Interrupt requests are masked whenever the interrupt enable bit in the flag word is cleared. When the 80C286 responds to an interrupt request, it performs two interrupt acknowledge bus cycles to read an 8-bit interrupt vector that identifies the source of the interrupt. To ensure program interruption, INTR must remain active until an interrupt acknowledge bus cycle is initiated. INTR is sampled at the beginning of each processor cycle and must be active HIGH at least two processor cycles before the current instruction ends in order to interrupt before the next instruction. INTR is level sensitive, active HIGH, and may be asynchronous to the system clock.
NMI	59	I	NON-MASKABLE INTERRUPT REQUEST: interrupts the 80C286 with an internally supplied vector value of two. No interrupt acknowledge cycles are performed. The interrupt enable bit in the 80C286 flag word does not affect this input. The NMI input is active HIGH, may be asynchronous to the system clock, and is edge triggered after internal synchronization. For proper recognition, the input must have been previously LOW for at least four system clock cycles and remain HIGH for at least four system clock cycles.
PEREQ $\overline{PEACK}$	61 6	I O	PROCESSOR EXTENSION OPERAND REQUEST AND ACKNOWLEDGE: extend the memory management and protection capabilities of the 80C286 to processor extensions. The PEREQ input requests the 80C286 to perform a data operand transfer for a processor extension. The $\overline{PEACK}$ output signals the processor extension when the requested operand is being transferred. PEREQ is active HIGH. $\overline{PEACK}$ is active LOW and is held at a high impedance logic one during bus hold acknowledge. PEREQ may be asynchronous to the system clock.
$\overline{BUSY}$ $\overline{ERROR}$	54 53	I I	PROCESSOR EXTENSION BUSY AND ERROR: indicates the operating condition of a processor extension to the 80C286. An active $\overline{BUSY}$ input stops 80C286 program execution on WAIT and some ESC instructions until $\overline{BUSY}$ becomes inactive (HIGH). The 80C286 may be interrupted while waiting for $\overline{BUSY}$ to become inactive. An active $\overline{ERROR}$ input causes the 80C286 to perform a processor extension interrupt when executing WAIT or some ESC instructions. These inputs are active LOW and may be asynchronous to the system clock.

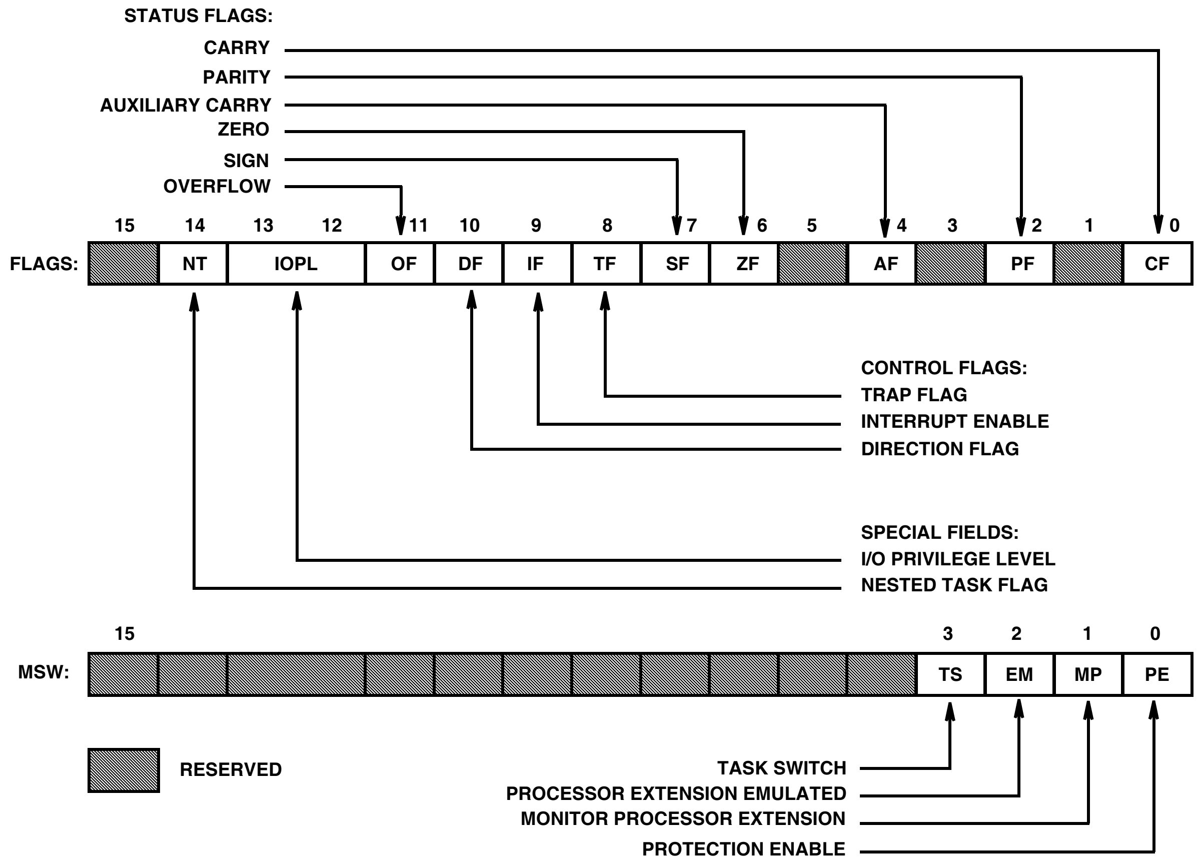


FIGURE 2. STATUS AND CONTROL REGISTER BIT FUNCTIONS

TABLE 1. FLAGS WORD BIT FUNCTIONS

BIT POSITION	NAME	FUNCTION
0	CF	Carry Flag - Set on high-order bit carry or borrow; cleared otherwise.
2	PF	Parity Flag - Set if low-order 8 bits of result contain an even number of 1 bits; cleared otherwise.
4	AF	Set on carry from or borrow to the low order four bits of AL; cleared otherwise.
6	ZF	Zero Flag - Set if result is zero; cleared otherwise.
7	SF	Sign Flag - Set equal to high-order bit of result (0 if positive, 1 if negative).
11	OF	Overflow Flag - Set if result is a too-large positive number or a too-small negative number (excluding sign-bit) to fit in destination operand; cleared otherwise.
8	TF	Single Step Flag - Once set, a single step interrupt occurs after the next instruction executes. TF is cleared by the single step interrupt.
9	IF	Interrupt-Enable Flag - When set, maskable interrupts will cause the CPU to transfer control to an interrupt vector specified location.
10	DF	Direction Flag - Causes string instructions to auto decrement the appropriate index registers when set. Clearing DF causes auto increment.

TABLE 8. RECOMMENDED MSW ENCODINGS FOR PROCESSOR EXTENSION CONTROL

TS	MP	EM	RECOMMENDED USE	INSTRUCTION CAUSING EXCEPTION 7
0	0	0	Initial encoding after RESET. 80C286 operation is identical to 80C86/88.	None
0	0	1	No processor extension is available. Software will emulate its function.	ESC
1	0	1	No processor extension is available. Software will emulate its function. The current processor extension context may belong to another task.	ESC
0	1	0	A processor extension exists.	None
1	1	0	A processor extension exists. The current processor extension context may belong to another task. The exception 7 on WAIT allows software to test for an error pending from a previous processor extension operation.	ESC or WAIT

TABLE 9. REAL ADDRESS MODE ADDRESSING INTERRUPTS

FUNCTION	INTERRUPT NUMBER	RELATED INSTRUCTIONS	RETURN ADDRESS BEFORE INSTRUCTION
Interrupt table limit too small exception	8	INT vector is not within table limit	Yes
Processor extension segment overrun interrupt	9	ESC with memory operand extending beyond offset FFFF(H)	No
Segment overrun exception	13	Word memory reference with offset = FFFF(H) or an attempt to execute past the end of a segment	Yes

80C286 Real Address Mode

The 80C286 executes a fully upward-compatible superset of the 80C86 instruction set in real address mode. In real address mode the 80C286 is object code compatible with 80C86 and 80C88 software. The real address mode architecture (registers and addressing modes) is exactly as described in the 80C286 Base Architecture section of this Functional Description.

Memory Size

Physical memory is a contiguous array of up to 1,048,576 bytes (one megabyte) addressed by pins A_0 through A_{19} and $\overline{BHE}$. A_{20} through A_{23} should be ignored.

Memory Addressing

In real address mode physical memory is a contiguous array of up to 1,048,576 bytes (one megabyte) addressed by pin A_0 through A_{19} and $\overline{BHE}$. Address bits A_{20}-A_{23} may not always be zero in real mode. A_{20}-A_{23} should not be used by the system while the 80C286 is operating in Real Mode.

The selector portion of a pointer is interpreted as the upper 16-bits of a 20-bit segment address. The lower four bits of the 20-bit segment address are always zero. Segment addresses, therefore, begin on multiples of 16 bytes. See Figure 6 for a graphic representation of address information.

All segments in real address mode are 64K bytes in size and may be read, written, or executed. An exception or interrupt can occur if data operands or instructions attempt to wrap around the end of a segment (e.g. a word with its low order byte at offset FFFF(H) and its high order byte at offset 0000(H)). If, in real address mode, the information contained in a segment does not use the full 64K bytes, the unused end of the segment may be overlaid by another segment to reduce physical memory requirements.

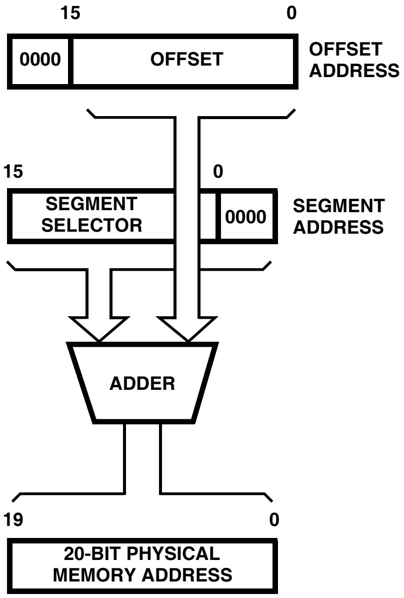


FIGURE 6. 80C286 REAL ADDRESS MODE ADDRESS CALCULATION

Protected Virtual Address Mode

The 80C286 executes a fully upward-compatible superset of the 80C86 instruction set in protected virtual address mode (protected mode). Protected mode also provides memory management and protection mechanisms and associated instructions.

The 80C286 enters protected virtual address mode from real address mode by setting the PE (Protection Enable) bit of the machine status word with the Load Machine Status Word (LMSW) instruction. Protected mode offers extended physical and virtual memory address space, memory protection mechanisms, and new operations to support operating systems and virtual memory.

All registers, instructions, and addressing modes described in the 80C286 Base Architecture section of this Functional Description remain the same. Programs for the 80C86, 80C88, and real address mode 80C286 can be run in protected mode; however, embedded constants for segment selectors are different.

Memory Size

The protected mode 80C286 provides a 1 gigabyte virtual address space per task mapped into a 16 megabyte physical address space defined by the address pins A_{23}-A_0 and $\overline{BHE}$. The virtual address space may be larger than the physical address space since any use of an address that does not map to a physical memory location will cause a restartable exception.

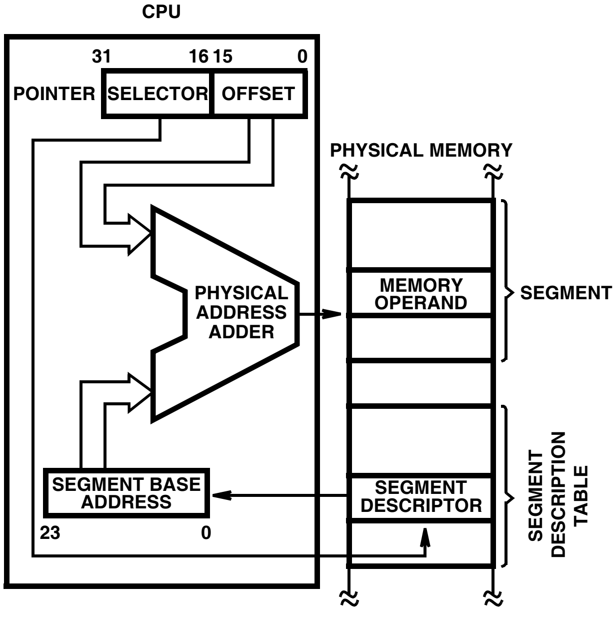


FIGURE 8. PROTECTED MODE MEMORY ADDRESSING

Memory Addressing

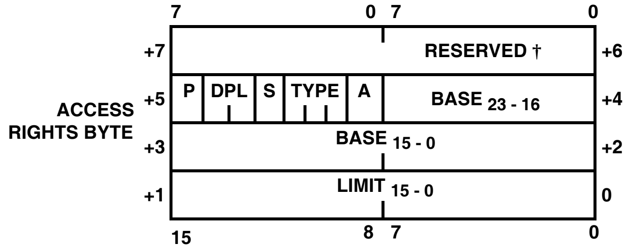
As in real address mode, protected mode uses 32-bit pointers, consisting of 16-bit selector and offset components. The selector, however, specifies an index into a memory resident table rather than the upper 16-bits of a real memory address. The 24-bit base address of the desired segment is obtained from the tables in memory. The 16-bit offset is added to the segment base address to form the physical address as shown in Figure 8. The tables are automatically referenced by the CPU whenever a segment register is loaded with a selector. All 80C286 instructions which load a segment register will reference the memory based tables without additional software. The memory based tables contain 8 byte values called descriptors.

Descriptors

Descriptors define the use of memory. Special types of descriptors also define new functions for transfer of control and task switching. The 80C286 has segment descriptors for code, stack and data segments, and system control descriptors for special system data segments and control transfer operations. Descriptor accesses are performed as locked bus operations to assure descriptor integrity in multi-processor systems.

Code and Data Segment Descriptors (S = 1)

Besides segment base addresses, code and data descriptors contain other segment attributes including segment size (1 to 64K bytes), access rights (read only, read/write, execute only, and execute/read), and presence in memory (for virtual memory systems) (See Table 10). Any segment usage violating a segment attribute indicated by the segment descriptor will prevent the memory cycle and cause an exception or interrupt.



† MUST BE SET TO 0 FOR COMPATIBILITY WITH FUTURE UPGRADES

FIGURE 9. CODE OR DATA SEGMENT DESCRIPTOR

Code and data (including stack data) are stored in two types of segments: code segments and data segments. Both types are identified and defined by segment descriptors (S = 1). Code segments are identified by the executable (E) bit set to 1 in the descriptor access rights byte. The access rights byte of both code and data segment descriptor types have three fields in common: present (P) bit, Descriptor Privilege Level (DPL), and accessed (A) bit. If P = 0, any attempted use of this segment will cause a not-present exception. DPL specifies the privilege level of the segment descriptor. DPL controls when the descriptor may be used by a task (refer to privilege discussion below). The A bit shows whether the segment has been previously accessed for usage profiling, a necessity for virtual memory systems. The CPU will always set this bit when accessing the descriptor.

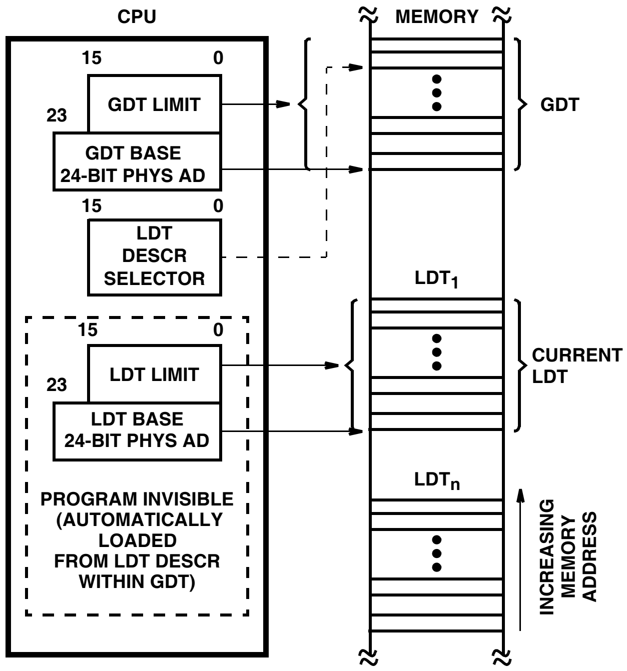


FIGURE 14. LOCAL AND GLOBAL DESCRIPTOR TABLE DEFINITION

The LGDT and LLDT instructions load the base and limit of the global and local descriptor tables. LGDT and LLDT are privileged, i.e. they may only be executed by trusted programs operating at level 0. The LGDT instruction loads a six byte field containing the 16-bit table limit and 24-bit physical base address of the Global Descriptor Table as shown in Figure 15. The LDT instruction loads a selector which refers to a Local Descriptor Table descriptor containing the base address and limit for an LDT, as shown in Table 11.

	7 0	7 0	
+5	RESERVED †	BASE $_{23-16}$	+4
+3	BASE $_{15-0}$		+2
+1	LIMIT $_{15-0}$		0
	15 8	7 0	

† MUST BE SET TO 0 FOR COMPATIBILITY WITH FUTURE UPGRADES

FIGURE 15. GLOBAL DESCRIPTOR TABLE AND INTERRUPT DESCRIPTOR TABLE DATA TYPE

Interrupt Descriptor Table

The protected mode 80C286 has a third descriptor table, called the Interrupt Descriptor Table (IDT) (see Figure 16), used to define up to 256 interrupts. It may contain only task gates, interrupt gates and trap gates. The IDT (Interrupt Descriptor Table) has a 24-bit physical base and 16-bit limit register in the CPU. The privileged LIDT instruction loads these registers with a six byte value of identical form to that of the LGDT instruction (see Figure 16 and Protected Mode Initialization).

References to IDT entries are made via INT instructions, external interrupt vectors, or exceptions. The IDT must be at least 256 bytes in size to allocate space for all reserved interrupts.

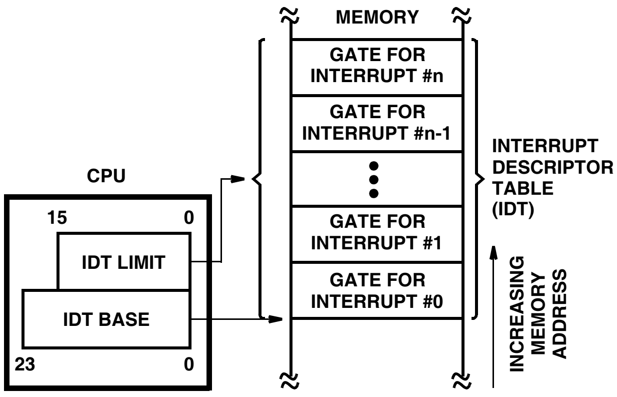
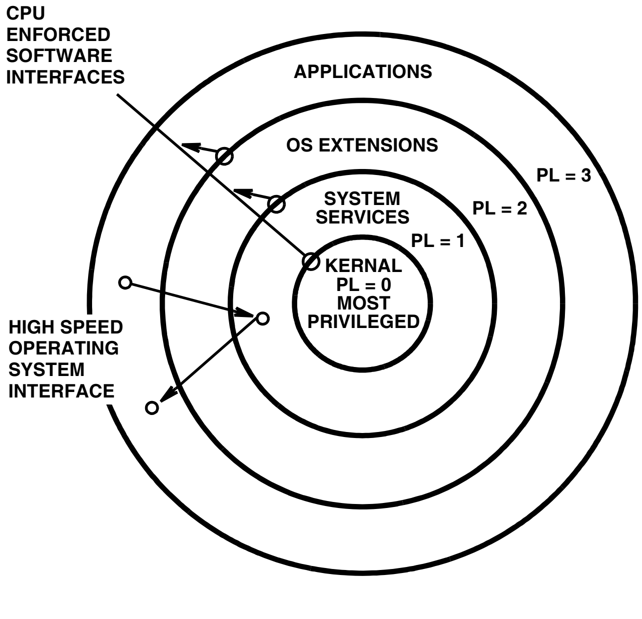


FIGURE 16. INTERRUPT DESCRIPTOR TABLE DEFINITION

Privilege

The 80C286 has a four-level hierarchical privilege system which controls the use of privileged instructions and access to descriptors (and their associated segments) within a task. Four-level privilege, as shown in Figure 17, is an extension of the users/supervisor mode commonly found in minicomputers. The privilege levels are numbered 0 through 3. Level 0 is the most privileged level. Privilege levels provide protection within a task. (Tasks are isolated by providing private LDT's for each task.) Operating system routines, interrupt handlers, and other system software can be included and protected within the virtual address space of each task using the four levels of privilege. Each task in the system has a separate stack for each of its privilege levels.

Tasks, descriptors, and selectors have a privilege level attribute that determines whether the descriptor may be used. Task privilege affects the use of instructions and descriptors. Descriptor and selector privilege only affect access to the descriptor.


CPU ENFORCED SOFTWARE INTERFACES
APPLICATIONS
OS EXTENSIONS
PL = 3
SYSTEM SERVICES
PL = 2
PL = 1
KERNAL PL = 0 MOST PRIVILEGED
HIGH SPEED OPERATING SYSTEM INTERFACE

NOTE: PL becomes numerically lower as privilege level increases.

FIGURE 17. HIERARCHICAL PRIVILEGE LEVELS

TABLE 15. OPERAND REFERENCE CHECKS

ERROR DESCRIPTION	EXCEPTION NUMBER
Write into code segment	13
Read from execute-only code segment	13
Write to read-only data segment	13
Segment limit exceeded (See Note)	12 or 13

NOTE: Carry out in offset calculations is ignored.

TABLE 16. PRIVILEGED INSTRUCTION CHECKS

ERROR DESCRIPTION	EXCEPTION NUMBER
CPL ≠ 0 when executing the following instructions: LIDT, LLDT, LGDT, LTR, LMSW, CTS, HLT	13
CPT > IOPL when executing the following instructions: INS, IN, OUTS, OUT, STI, CLI, LOCK	13

The IRET and POPF instructions do not perform some of their defined functions if CPL is not of sufficient privilege (numerically small enough). Precisely these are:

- The IF bit is not changed if CPL is greater than IOPL.
- The IOPL field of the flag word is not changed if CPL is greater than 0.

No exceptions or other indication are given when these conditions occur.

Exceptions

The 80C286 detects several types of exceptions and interrupts in protected mode (see Table 17). Most are restartable after the exceptional condition is removed. Interrupt handlers for most exceptions can read an error code, pushed on the stack after the return address, that identifies the selector involved (0 if none). The return address normally points to the failing instruction including all leading prefixes. For a processor extension segment overrun exception, the return address will not point at the ESC instruction that caused the exception; however, the processor extension registers may contain the address of the failing instruction.

These exceptions indicate a violation to privilege rules or usage rules has occurred. Restart is generally not attempted under those conditions.

All these checks are performed for all instructions and can be split into three categories: segment load checks (Table 14), operand reference checks (Table 15), and privileged instruction checks (Table 16). Any violation of the rules shown will result in an exception. A not-present exception causes exception 11 or 12 and is restartable.

SPECIAL OPERATIONS

Task Switch Operation

The 80C286 provides a built-in task switch operation which saves the entire 80C286 execution state (registers, address space, and a link to the previous task), loads a new execution state, and commences execution in the new task. Like gates, the task switch operation is invoked by executing an inter-segment JMP or CALL instruction which refers to a Task State Segment (TSS) or task gate descriptor in the GDT or LDT. An INT instruction, exception, or external interrupt may also invoke the task switch operation by selecting a task gate descriptor in the associated IDT descriptor entry.

The TSS descriptor points at a segment (see Figure 18) containing the entire 80C286 execution state while a task gate descriptor contains a TSS selector. The limit field of the descriptor must be greater than 002B(H).

Each task must have a TSS associated with it. The current TSS is identified by a special register in the 80C286 called the Task Register (TR). This register contains a selector referring to the task state segment descriptor that defines the current TSS. A hidden base and limit register associated with TR are loaded whenever TR is loaded with a new selector. The IRET instruction is used to return control to the task that called the current task or was interrupted. Bit 14 in the flag register is called the Nested Task (NT) bit. It controls the

TABLE 17. PROTECTED MODE EXCEPTIONS

INTERRUPT VECTOR	FUNCTION	RETURN ADDRESS AT FALLING INSTRUCTION?	ALWAYS RESTARTABLE?	ERROR CODE ON STACK?
8	Double exception detected	Yes	No (Note 7)	Yes
9	Processor extension segment overrun	No	No (Note 7)	No
10	Invalid task state segment	Yes	Yes	Yes
11	Segment not present	Yes	Yes	Yes
12	Stack segment overrun or stack segment not present	Yes	Yes (Note 6)	Yes
13	General protection	Yes	No (Note 7)	Yes

NOTES:

6. When a PUSHA or POPA instruction attempts to wrap around the stack segment, the machine state after the exception will not be restartable because stack segment wrap around is not permitted. This condition is identified by the value of the saved SP being either 0000(H), 0001(H), FFFE(H), or FFFF(H).
7. These exceptions indicate a violation to privilege rules or usage rules has occurred. Restart is generally not attempted under those conditions.

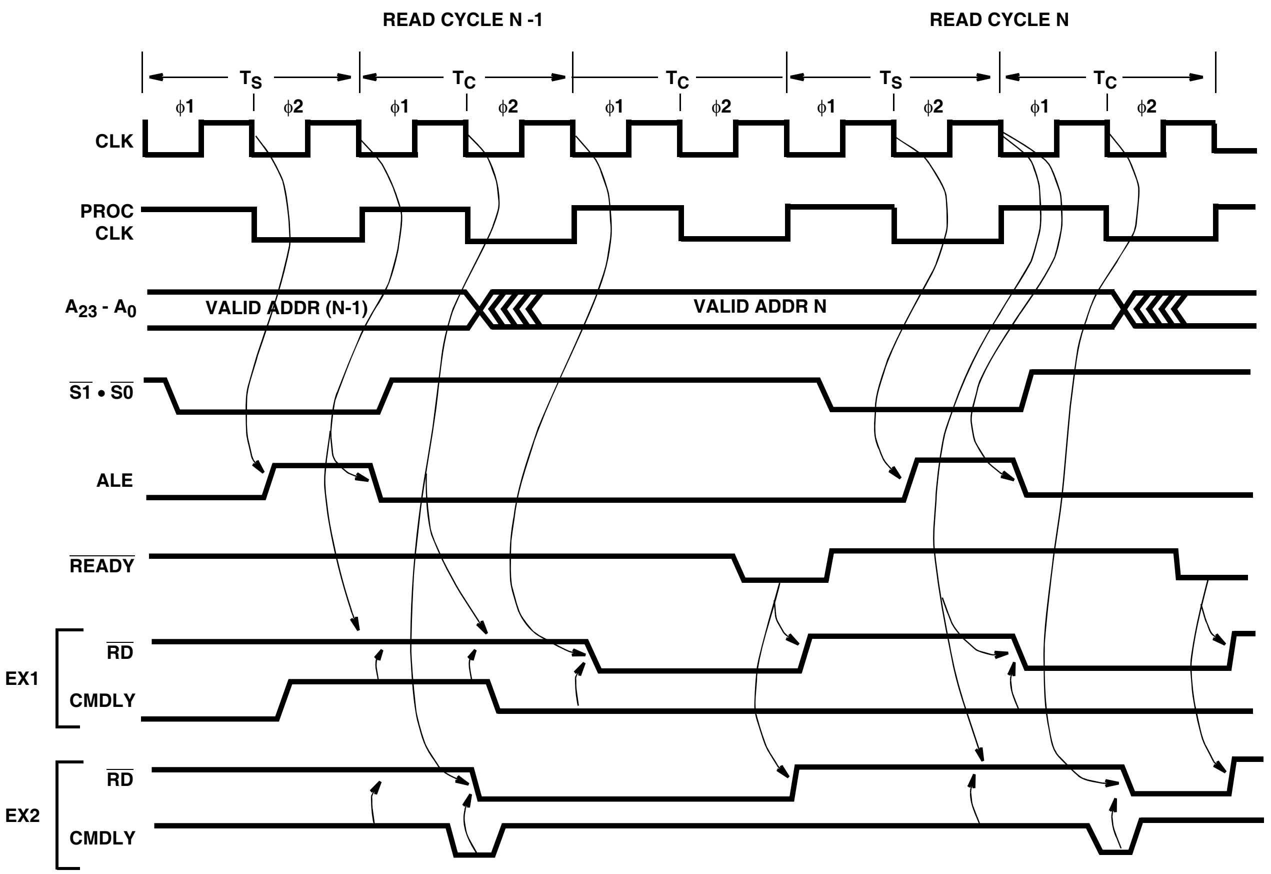


FIGURE 23. CMDLY CONTROLS THE LEADING EDGE OF COMMAND SIGNAL

Bus Cycle Termination

At maximum transfer rates, the 80C286 bus alternates between the status and command states. The bus status signals become inactive after T_S so that they may correctly signal the start of the next bus operation after the completion of the current cycle. No external indication of T_C exists on the 80C286 local bus. The bus master and bus controller enter T_C directly after T_S and continue executing T_C cycles until terminated by the assertion of $\overline{\text{READY}}$.

$\overline{\text{READY}}$ Operation

The current bus master and 82C288 bus controller terminate each bus operation simultaneously to achieve maximum bus operation bandwidth. Both are informed in advance by $\overline{\text{READY}}$ active (open-collector output from 82C284) which identifies the last T_C cycle of the current bus operation. The bus master and bus controller must see the same sense of the $\overline{\text{READY}}$ signal, thereby requiring $\overline{\text{READY}}$ to be synchronous to the system clock.

Synchronous Ready

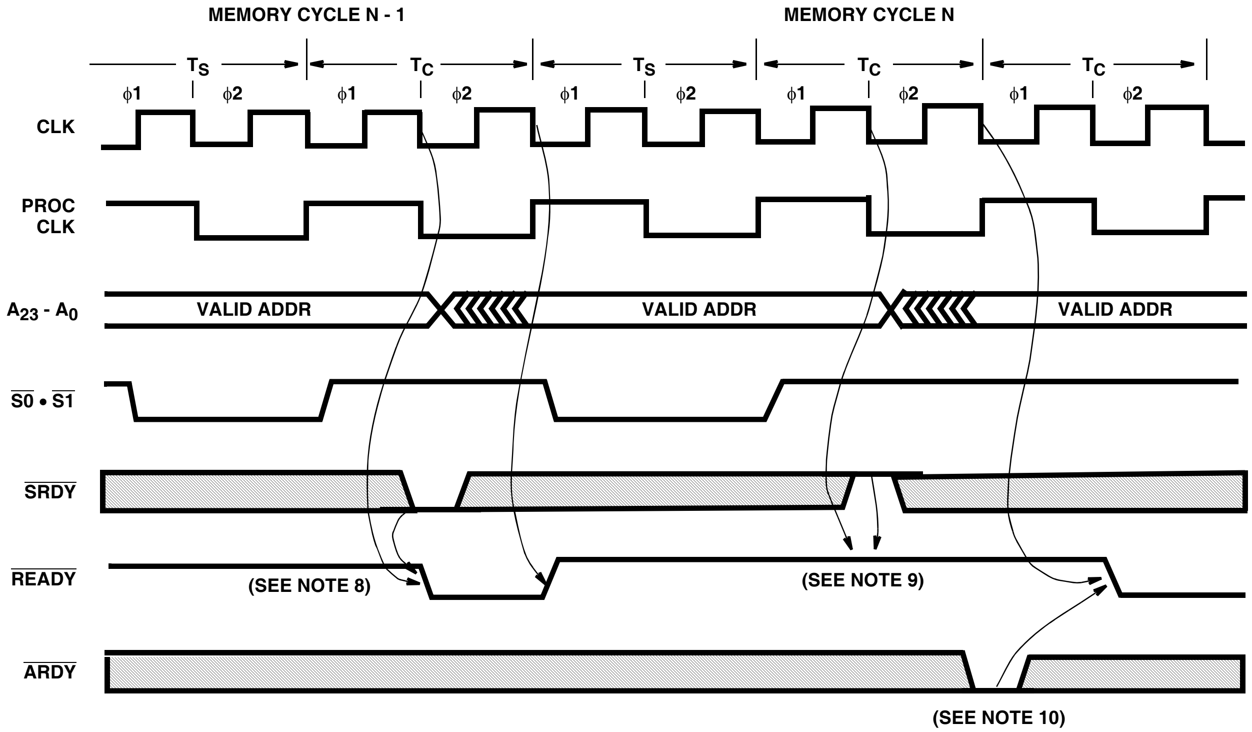
The 82C284 clock generator provides $\overline{\text{READY}}$ synchronization from both synchronous and asynchronous sources (see Figure 24). The synchronous ready input ($\overline{\text{SRDY}}$) of the clock generator is sampled with the falling edge of CLK at the end of phase 1 of each T_C. The state of $\overline{\text{SRDY}}$ is then broadcast to the bus master and bus controller via the $\overline{\text{READY}}$ output line.

Asynchronous Ready

Many systems have devices or subsystems that are asynchronous to the system clock. As a result, their ready outputs cannot be guaranteed to meet the 82C284 $\overline{\text{SRDY}}$ setup and hold time requirements. But the 82C284 asynchronous ready input ($\overline{\text{ARDY}}$) is designed to accept such signals. The $\overline{\text{ARDY}}$ input is sampled at the beginning of each T_C cycle by 82C284 synchronization logic. This provides one system CLK cycle time to resolve its value before broadcasting it to the bus master and bus controller.

$\overline{\text{ARDY}}$ or $\overline{\text{ARDYEN}}$ must be HIGH at the end of T_S. $\overline{\text{ARDY}}$ cannot be used to terminate the bus cycle with no wait states.

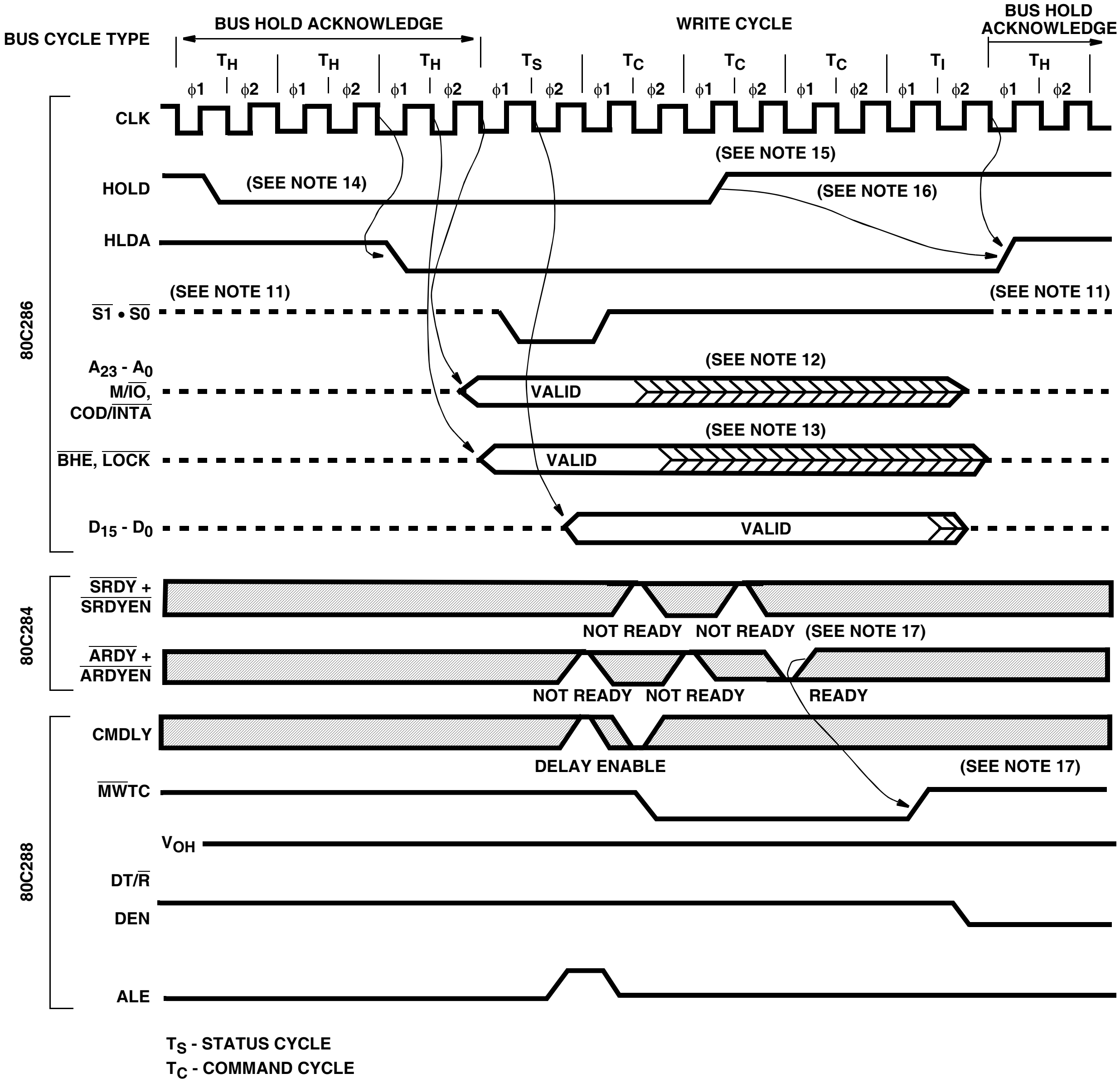
Each ready input of the 82C284 has an enable pin ($\overline{\text{SRDYEN}}$ and $\overline{\text{ARDYEN}}$) to select whether the current bus operation will be terminated by the synchronous or asynchronous ready. Either of the ready inputs may terminate a bus operation. These enable inputs are active low and have the same timing as their respective ready inputs. Address decode logic usually selects whether the current bus operation should be terminated by $\overline{\text{ARDY}}$ or $\overline{\text{SRDY}}$.



NOTES:

8. SRDYEN is active low.
9. If SRDYEN is high, the state of SRDY will not effect READY.
10. ARDYEN is active low.

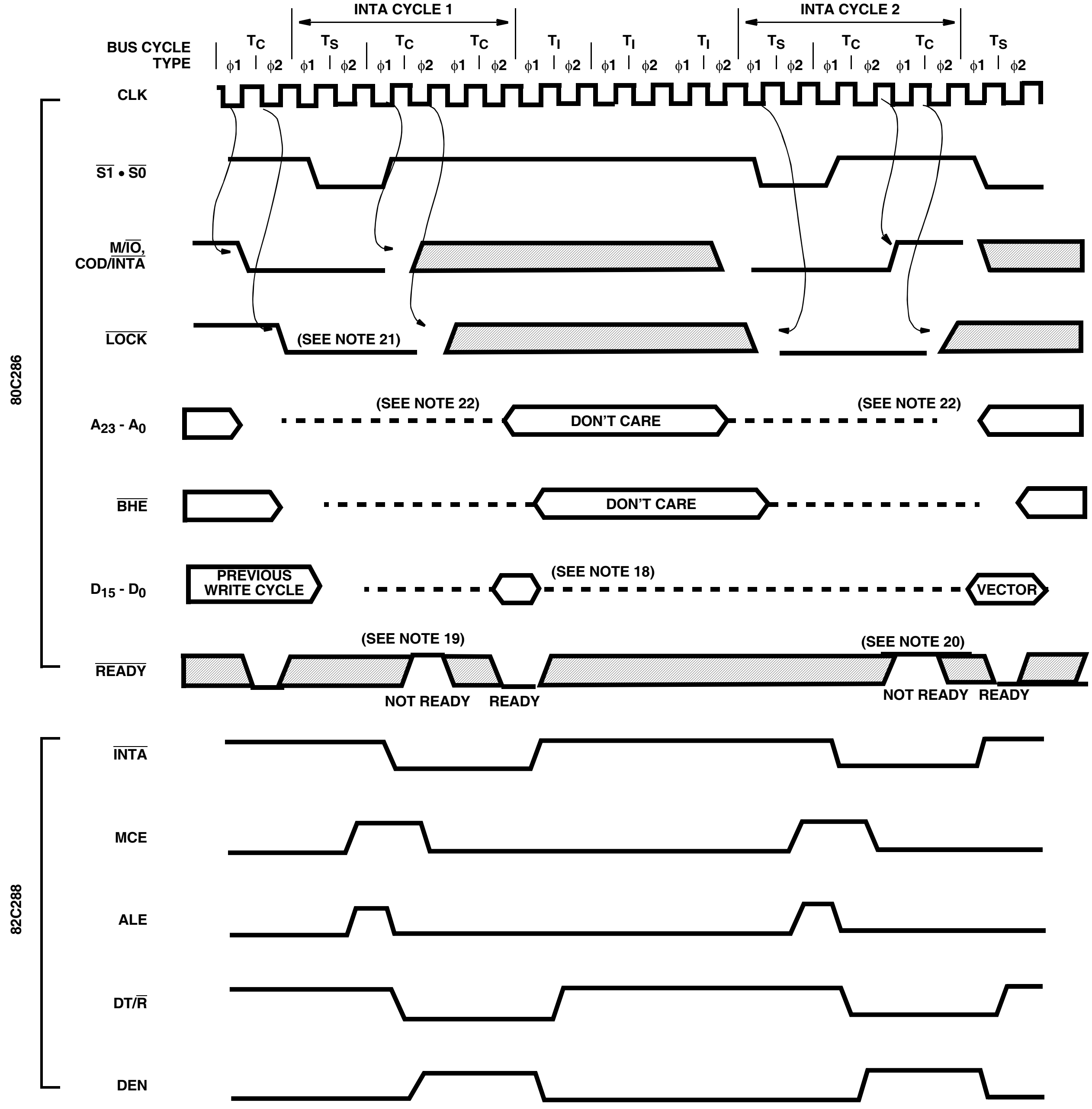
FIGURE 24. SYNCHRONOUS AND ASYNCHRONOUS READY



NOTES:

11. Status lines are held at a high impedance logic one by the 80C286 during a HOLD state.
12. Address, M/$\overline{IO}$ and COD/$\overline{INTA}$ may start floating during any T_C depending on when internal 80C286 bus arbiter decides to release bus to external HOLD. The float starts in φ2 of T_C.
13. $\overline{BHE}$ and $\overline{LOCK}$ may start floating after the end of any T_C depending on when internal 80C286 bus arbiter decides to release bus to external HOLD. The float starts in φ1 of T_C.
14. The minimum HOLD to HLDA time is shown. Maximum is one T_H longer.
15. The earliest HOLD time is shown. It will always allow a subsequent memory cycle if pending is shown.
16. The minimum HOLD to HLDA time is shown. Maximum is a function of the instruction, type of bus cycle and other machine state (i.e., Interrupts, Waits, Lock, etc.).
17. Asynchronous ready allows termination of the cycle. Synchronous ready does not signal ready in this example. Synchronous ready state is ignored after ready is signaled via the asynchronous input.

FIGURE 28. MULTIBUS WRITE TERMINATED BY ASYNCHRONOUS READY WITH BUS HOLD



NOTES:

18. Data is ignored.
19. First INTA cycle should have at least one wait state inserted to meet 82C59A minimum INTA pulse width.
20. Second INTA cycle must have at least one wait state inserted since the CPA will not drive A_{23}-A_0, $\overline{BHE}$, and $\overline{LOCK}$ until after the first T_C state. The CPU imposed one/clock delay prevents has contention between cascade address buffer being disabled by MCE ↓ and address outputs.
21. Without the wait state, the 80C286 address will not be valid for a memory cycle started immediately after the second INTA cycle. The 82C59A also requires one wait state for minimum INTA pulse width.
22. $\overline{LOCK}$ is active for the first INTA cycle to prevent the 82289 from releasing the bus between INTA cycles in a multi-master system. $\overline{LOCK}$ is also active for the second INTA cycle.
23. A_{23}-A_0 exits three-state OFF during $\phi 2$ of the second T_C in the INTA cycle.

FIGURE 29. INTERRUPT ACKNOWLEDGE SEQUENCE

Absolute Maximum Ratings

Supply Voltage	±8.0V
Input, Output or I/O Voltage Applied	GND -1.0V to V_{CC} +1.0V
Storage Temperature Range	-65^{o}C to +150^{o}C
Junction Temperature, PGA	+175^{o}C
PLCC	+150^{o}C
Lead Temperature (Soldering, 10s) (PLCC - Lead Tips Only)	+300^{o}C

Thermal Information

Thermal Resistance (Typical)	θ_{JA} (^{o}C/W)	θ_{JC} (^{o}C/W)
PDIP Package	35	6
CERDIP Package	33	9

Maximum Package Power Dissipation	
PGA Package	1.22W
PLCC Package	2.2W
Gate Count	22,500

CAUTION: Stresses above those listed in “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

Operating Conditions

Operating Voltage Range	
80C286-10, -12	+4.5V to +5.5V
80C286-16, -20, -25	+4.75V to +5.25V

Operating Temperature Range	
I80C286-10, -12, -16, -20	-40^{o}C to +85^{o}C
C80C286-12, -16, -20, -25	0^{o}C to +70^{o}C

DC Electrical Specifications

V_{CC} = +5V ± 10%, T_A = 0^{o}C to +70^{o}C (C80C286-12), V_{CC} = +5V ± 5%, T_A = 0^{o}C to +70^{o}C (C80C286-16, -20, -25), V_{CC} = +5V ± 10%, T_A = -40^{o}C to +85^{o}C (I80C286-10, -12), V_{CC} = +5V ± 5%, T_A = -40^{o}C to +85^{o}C (I80C286-16, -20)

SYMBOL	PARAMETER	MIN	MAX	UNITS	TEST CONDITIONS
V_{IL}	Input LOW Voltage	-0.5	0.8	V	
V_{IH}	Input HIGH Voltage	2.0	V_{CC} +0.5	V	
V_{ILC}	CLK Input LOW Voltage	-0.5	1.0	V	
V_{IHC}	CLK Input HIGH Voltage	3.6	V_{CC} +0.5	V	
V_{OL}	Output LOW Voltage	-	0.4	V	I_{OL} = 2.0mA
V_{OH}	Output HIGH Voltage	3.0 V_{CC} -0.4	- -	V	I_{OH} = -2.0mA, I_{OH} = -100μA
I_I	Input Leakage Current	-10	10	μA	V_{IN} = GND or V_{CC} Pins 29, 31, 57, 59, 61, 63-64
I_{SH}	Input Sustaining Current on $\overline{BUSY}$ and $\overline{ERROR}$ Pins	-30	-500	μA	V_{IN} = GND (See Note 28)
I_{BHL}	Input Sustaining Current LOW	38	200	μA	V_{IN} = 1.0V (See Note 24)
I_{BHH}	Input Sustaining Current HIGH	-50	-400	μA	V_{IN} = 3.0V (See Note 25)
I_O	Output Leakage Current	-10	10	μA	V_O = GND or V_{CC} Pins 1, 7-8, 10-28, 32-34
I_{CCOP}	Active Power Supply Current	-	185	mA	80C286-10 (See Note 27)
		-	220	mA	80C286-12 (See Note 27)
		-	260	mA	80C286-16 (See Note 27)
		-	310	mA	80C286-20 (See Note 27)
		-	410	mA	80C286-25 (See Note 27)
I_{CCSB}	Standby Power Supply Current	-	5	mA	(See Note 26)

Capacitance

T_A = +25^{o}C, All Measurements Referenced to Device GND

SYMBOL	PARAMETER	TYP	UNITS	TEST CONDITIONS
C_{CLK}	CLK Input Capacitance	10	pF	FREQ = 1MHz
C_{IN}	Other Input Capacitance	10	pF	
$C_{I/O}$	I/O Capacitance	10	pF	

NOTES:

24. I_{BHL} should be measured after lowering V_{IN} to GND and then raising to 1.0V on the following pins: 36-51, 66, 67.
25. I_{BHH} should be measured after raising V_{IN} to V_{CC} and then lowering to 3.0V on the following pins: 4-6, 36-51, 66-68.
26. I_{CCSB} tested with the clock stopped in phase two of the processor clock cycle. V_{IN} = V_{CC} or GND, V_{CC} = V_{CC} (Max), outputs unloaded.
27. I_{CCOP} measured at 10MHz for the 80C286-10, 12.5MHz for the 80C286-12, 16MHz for the 80C286-16, 20MHz for the 80C286-20, and 25MHz for the 80C286-25. V_{IN} = 2.4V or 0.4V, V_{CC} = V_{CC} (Max), outputs unloaded.
28. I_{SH} should be measured after raising V_{IN} to V_{CC} and then lowering to GND on pins 53 and 54.

AC Electrical Specifications V_{CC} = +5V ±10%, T_A = 0°C to +70°C (C80C286-12), T_A = -40°C to +85°C (I80C286-10, -12) V_{CC} = +5V ±5%, T_A = 0°C to +70°C (C80C286-16), T_A = -40°C to +85°C (I80C286-16) AC Timings are Referenced to 0.8V and 2.0V Points of the Signals as Illustrated in Data Sheet Waveforms, Unless Otherwise Specified

SYMBOL	PARAMETER	10MHz		12.5MHz		16MHz		UNIT	TEST CONDITION
		MIN	MAX	MIN	MAX	MIN	MAX		
TIMING REQUIREMENTS									
1	System Clock (CLK) Period	50	-	40	-	31	-	ns	
2	System Clock (CLK) LOW Time	12	-	11	-	7	-	ns	At 1.0V
3	System Clock (CLK) HIGH Time	16	-	13	-	11	-	ns	At 3.6V
17	System Clock (CLK) RISE Time	-	8	-	8	-	5	ns	1.0V to 3.6V
18	System Clock (CLK) FALL Time	-	8	-	8	-	5	ns	3.6V to 1.0V
4	Asynchronous Inputs SETUP Time	20	-	15	-	5	-	ns	(Note 29)
5	Asynchronous Inputs HOLD Time	20	-	15	-	5	-	ns	(Note 29)
6	RESET SETUP Time	19	-	10	-	10	-	ns	
7	RESET HOLD Time	0	-	0	-	0	-	ns	
8	Read Data SETUP Time	8	-	5	-	5	-	ns	
9	Read Data HOLD Time	4	-	4	-	3	-	ns	
10	$\overline{\text{READY}}$ SETUP Time	26	-	20	-	12	-	ns	
11	$\overline{\text{READY}}$ HOLD Time	25	-	20	-	5	-	ns	
20	Input RISE/FALL Times	-	10	-	8	-	6	ns	0.8V to 2.0V
TIMING RESPONSES									
12A	Status/$\overline{\text{PEACK}}$ Active Delay	1	22	1	21	1	18	ns	1, (Notes 31, 35)
12B	Status/$\overline{\text{PEACK}}$ Inactive Delay	1	30	1	24	1	20	ns	1, (Notes 31, 34)
13	Address Valid Delay	1	35	1	32	1	27	ns	1, (Notes 30, 31)
14	Write Data Valid Delay	0	40	0	31	0	28	ns	1, (Notes 30, 31)
15	Address/Status/Data Float Delay	0	47	0	32	0	29	ns	2, (Note 33)
16	HLDA Valid Delay	0	47	0	25	0	25	ns	1, (Notes 31, 36)
19	Address Valid to Status SETUP Time	27	-	22	-	16	-	ns	1, (Notes 31, 32)

NOTES:

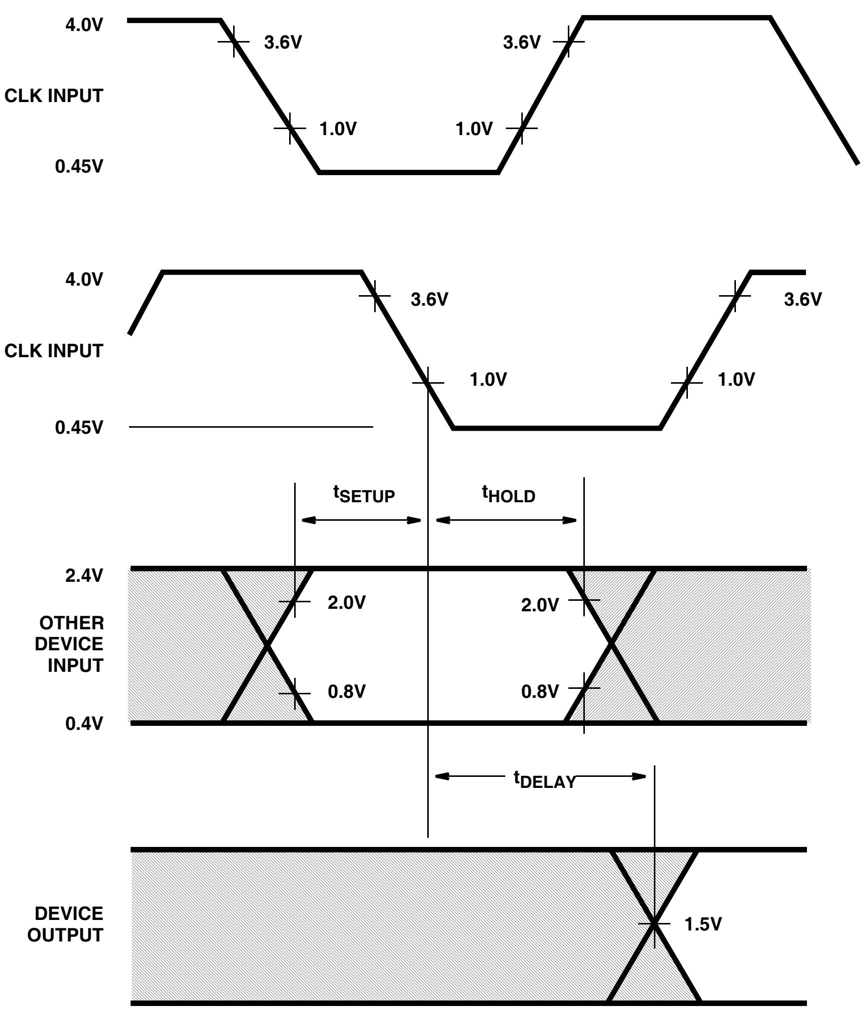
29. Asynchronous inputs are INTR, NMI, HOLD, PEREQ, ERROR, and BUSY. This specification is given only for testing purposes, to assure recognition at a specific CLK edge.
30. Delay from 1.0V on the CLK to 0.8V or 2.0V.
31. Output load: C_L = 100pF.
32. Delay measured from address either reaching 0.8V or 2.0V (valid) to status going active reaching 0.8V or status going inactive reaching 2.0V.
33. Delay from 1.0V on the CLK to Float (no current drive) condition.
34. Delay from 1.0V on the CLK to 0.8V for min. (HOLD time) and to 2.0V for max. (inactive delay).
35. Delay from 1.0V on the CLK to 2.0V for min. (HOLD time) and to 0.8V for max. (active delay).
36. Delay from 1.0V on the CLK to 2.0V.

AC Test Conditions

TEST CONDITION	I_L (CONSTANT CURRENT SOURCE)	C_L
1	|2.0mA|	100pF
2	-6mA (V_{OH} to Float) 8mA (V_{OL} to Float)	100pF

AC Specifications (Continued)

C80C286-20, -25
I80C286-20
AC DRIVE AND MEASURE POINTS - CLK INPUT



NOTE: Typical Output Rise/Fall Time is 6ns. For AC testing, input rise and fall times are driven at 1ns per volt.

FIGURE 33.

AC Electrical Specifications

82C284 and 82C288 Timing Specifications are given for reference only and no guarantee is implied.

82C284 Timing

SYMBOL	PARAMETER	10MHz		12.5MHz		16MHz		UNIT	TEST CONDITION
		MIN	MAX	MIN	MAX	MIN	MAX		
TIMING REQUIREMENTS									
11	$\overline{SRDY}$/$\overline{SRDYEN}$ Setup Time	15	-	15	-	10	-	ns	
12	$\overline{SRDY}$/$\overline{SRDYEN}$ Hold Time	2	-	2	-	1	-	ns	
13	$\overline{ARDY}$/$\overline{ARDYEN}$ Setup Time	5	-	5	-	3	-	ns	(Note 43)
14	$\overline{ARDY}$/$\overline{ARDYEN}$ Hold Time	30	-	25	-	20	-	ns	(Note 43)
TIMING RESPONSES									
19	PCLK Delay	0	20	0	16	0	15	ns	C_L = 75pF, I_{OL} = 5mA, I_{OH} = 1mA

NOTE:

43. These times are given for testing purposes to ensure a predetermined action.

82C288 Timing

SYMBOL	PARAMETER	10MHz		12.5MHz		16MHz		UNIT	TEST CONDITION
		MIN	MAX	MIN	MAX	MIN	MAX		
TIMING REQUIREMENTS									
12	CMDLY Setup Time	15	-	15	-	10	-	ns	
13	CMDLY Hold Time	1	-	1	-	0	-	ns	
TIMING RESPONSES									
16	ALE Active Delay	1	16	1	16	1	12	ns	
17	ALE Inactive Delay	-	19	-	19	-	15	ns	
19	DT/$\overline{R}$ Read Active Delay	-	23	-	23	-	18	ns	C_L = 150pF
20	DEN Read Active Delay	-	21	-	21	-	16	ns	I_{OL} = 16mA Max
21	DEN Read Inactive Delay	3	23	3	21	5	14	ns	I_{OL} = 1mA Max
22	DT/$\overline{R}$ Read Inactive Delay	5	24	5	18	5	14	ns	
23	DEN Write Active Delay	-	23	-	23	-	17	ns	
24	DEN Write Inactive Delay	3	23	3	23	3	15	ns	
29	Command Active Delay from CLK	3	21	3	21	3	15	ns	C_L = 300pF
30	Command Inactive Delay from CLK	3	20	3	20	3	15	ns	I_{OL} = 32mA Max

NOTE:

44. These times are given for testing purposes to ensure a predetermined action.

Waveforms (Continued)

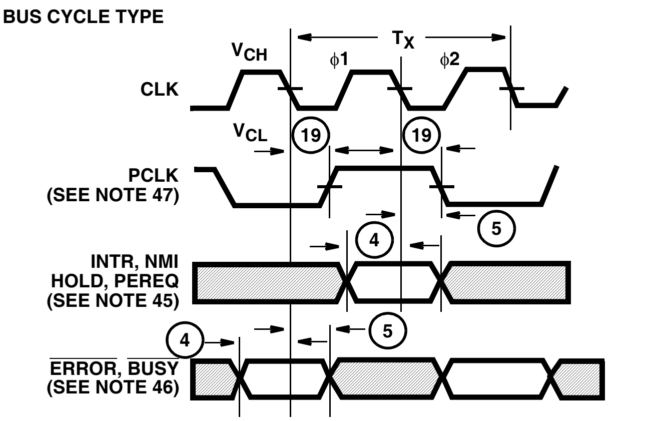


FIGURE 35. 80C286 ASYNCHRONOUS INPUT SIGNAL TIMING

NOTES:

45. PCLK indicates which processor cycle phase will occur on the next CLK, PCLK may not indicate the correct phase until the first cycle is performed.

46. These inputs are asynchronous. The setup and hold times shown assure recognition for testing purposes.

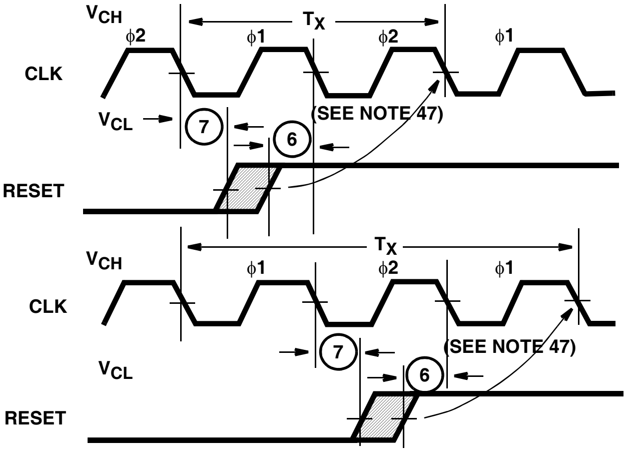
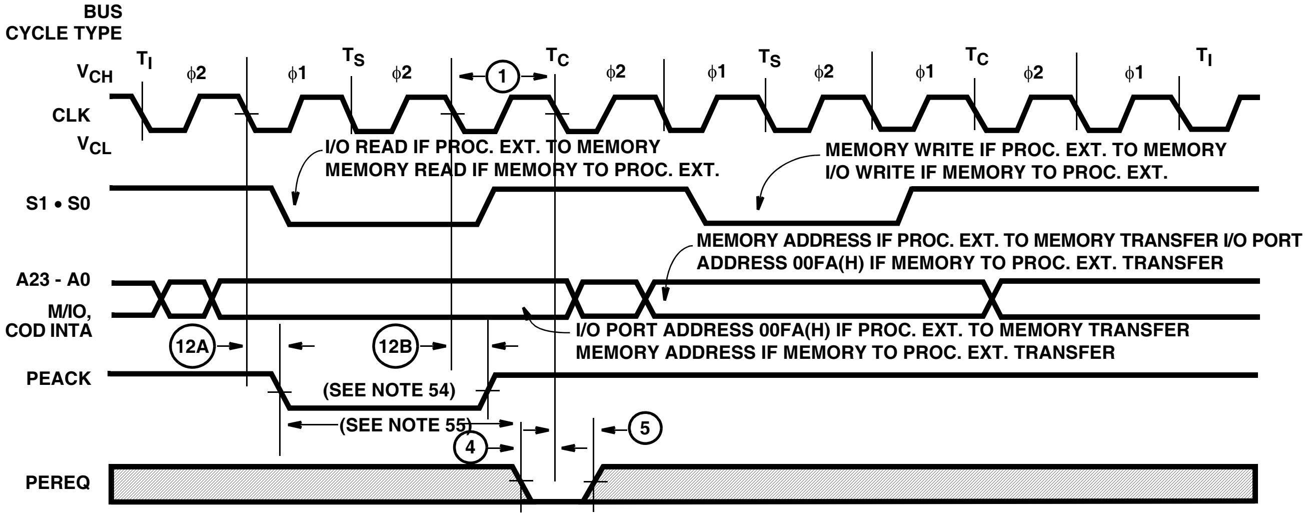


FIGURE 36. 80C286 RESET INPUT TIMING AND SUBSEQUENT PROCESSOR CYCLE PHASE

NOTE:

47. When RESET meets the setup time shown, the next CLK will start or repeat ϕ1 of a processor cycle.

Waveforms (Continued)


BUS CYCLE TYPE
T I
T S
T C
T S
T C
T I
ϕ2
ϕ1
ϕ2
ϕ1
ϕ2
ϕ1
ϕ2
ϕ1
1
V CH
CLK
V CL
I/O READ IF PROC. EXT. TO MEMORY
MEMORY READ IF MEMORY TO PROC. EXT.
MEMORY WRITE IF PROC. EXT. TO MEMORY
I/O WRITE IF MEMORY TO PROC. EXT.
S1 • S0
MEMORY ADDRESS IF PROC. EXT. TO MEMORY TRANSFER I/O PORT
ADDRESS 00FA(H) IF MEMORY TO PROC. EXT. TRANSFER
A23 - A0
M/IO,
COD INTA
I/O PORT ADDRESS 00FA(H) IF PROC. EXT. TO MEMORY TRANSFER
MEMORY ADDRESS IF MEMORY TO PROC. EXT. TRANSFER
12A
12B
PEACK
(SEE NOTE 54)
(SEE NOTE 55)
4
5
PEREQ

ASSUMING WORD-ALIGNED MEMORY OPERAND. IF ODD ALIGNED, 80C286 TRANSFERS TO/FROM MEMORY BYTE-AT-A-TIME WITH TWO MEMORY CYCLES.

FIGURE 38. 80C286 PEREQ/$\overline{\text{PEACK}}$ TIMING FOR ONE TRANSFER ONLY

NOTES:

54. $\overline{\text{PEACK}}$ always goes active during the first bus operation of a processor extension data operand transfer sequence. The first bus operation will be either a memory read at operand address or I/O read at port address 00FA(H).

55. To prevent a second processor extension data operand transfer, the worst case maximum time (Shown above) is 3 x ① - 12A$_{MAX}$ - ④$_{MIN}$. The actual configuration dependent, maximum time is: 3 x ① - 12A$_{MAX}$ - ④$_{MIN}$ + N x 2 x ①. N is the number of extra T$_C$ states added to either the first or second bus operation of the processor extension data operand transfer sequence.

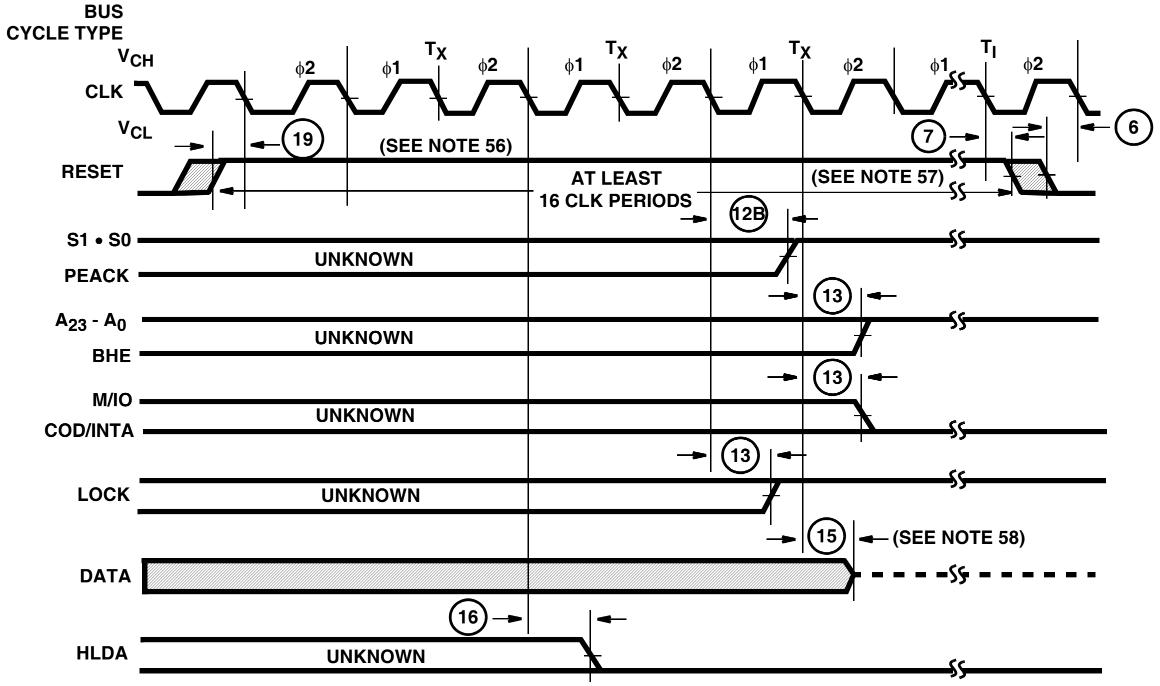


FIGURE 39. INITIAL 80C286 PIN STATE DURING RESET

NOTES:

56. Setup time for RESET ↑ may be violated with the consideration that ϕ1 of the processor clock may begin one system CLK period later.

57. Setup and hold times for RESET ↓ must be met for proper operation, but RESET ↓ may occur during ϕ1 or ϕ2.

58. The data bus is only guaranteed to be in a high impedance state at the time shown.

80C286 Instruction Set Summary **(Continued)**

FUNCTION	FORMAT				CLOCK COUNT		COMMENTS	
					REAL ADDRESS MODE	PROTECTED VIRTUAL ADDRESS MODE	REAL ADDRESS MODE	PROTECTED VIRTUAL ADDRESS MODE
LEA = Load EA to Register	10001101	mod reg r/m			3 (Note 59)	3 (Note 59)		
LDS = Load Pointer to DS	11000101	mod reg r/m	(mod ≠ 11)		7 (Note 59)	21 (Note 59)	2	9, 10, 11
LES = Load Pointer to ES	11000100	mod reg r/m	(mod ≠ 1)		7 (Note 59)	21 (Note 59)	2	9, 10, 11
LAHF Load AH with Flags	10011111				2	2		
SAHF = Store AH into Flags	10011110				2	2		
PUSHF = Push Flags	10011100				3	3	2	9
POPF = Pop Flags	10011101				5	5	2, 4	9, 15
ARITHMETIC								
ADD = Add								
Reg/Memory with Register to Either	000000dw	mod reg r/m			2, 7 (Note 59)	2, 7 (Note 59)	2	9
Immediate to Register/Memory	100000sw	mod 000 r/m	data	data if sw = 01	3, 7 (Note 59)	3, 7 (Note 59)	2	9
Immediate to Accumulator	0000010w	data	data if w = 1		3	3		
ADC = Add with Carry								
Reg/Memory with Register to Either	000100dw	mod reg r/m			2, 7 (Note 59)	2, 7 (Note 59)	2	9
Immediate to Register/Memory	100000sw	mod 010 r/m	data	data if sw = 01	3, 7 (Note 59)	3, 7 (Note 59)	2	9
Immediate to Accumulator	0001010w	data	data if w = 1		3	3		
INC = Increment								
Register/Memory	1111111w	mod 000 r/m			2, 7 (Note 59)	2, 7 (Note 59)	2	9
Register	01000 reg				2	2		
SUB = Subtract								
Reg/Memory and Register to Either	001010dw	mod reg r/m			2, 7 (Note 59)	2, 7 (Note 59)	2	9

80C286 Instruction Set Summary **(Continued)**

FUNCTION	FORMAT				CLOCK COUNT		COMMENTS	
					REAL ADDRESS MODE	PROTECTED VIRTUAL ADDRESS MODE	REAL ADDRESS MODE	PROTECTED VIRTUAL ADDRESS MODE
Register - Byte					13	13		
Register - Word					21	21		
Memory - Byte					16 (Note 59)	16 (Note 59)	2	9
Memory - Word					24 (Note 59)	24 (Note 59)	2	9
IMUL = Integer Multiply (Signed)	1111011w	mod 101 r/m						
Register - Byte					13	13		
Register - Word					21	21		
Memory - Byte					16 (Note 59)	16 (Note 59)	2	9
Memory - Word					24 (Note 59)	24 (Note 59)	2	9
IMUL = Interger Immediate Multiply (Signed)	011010s1	mod reg r/m	data	data if s = 0	21, 24 (Note 59)	21, 24 (Note 59)	2	9
DIV = Divide (Unsigned)	1111011w	mod 110 r/m						
Register - Byte					14	14	6	6
Register - Word					22	22	6	6
Memory - Byte					17 (Note 59)	17 (Note 59)	2, 6	6, 9
Memory - Word					25 (Note 59)	25 (Note 59)	2, 6	6, 9
IDIV = Integer Divide (Signed)	1111011w	mod 111 r/m						
Register - Byte					17	17	6	6
Register - Word					25	25	6	6
Memory - Byte					20 (Note 59)	20 (Note 59)	2, 6	6, 9
Memory - Word					28 (Note 59)	28 (Note 59)	2, 6	6, 9
AAM = ASCII Adjust for Multiply	11010100	00001010			16	16		

80C286 Instruction Set Summary **(Continued)**

FUNCTION	FORMAT				CLOCK COUNT		COMMENTS	
					REAL ADDRESS MODE	PROTECTED VIRTUAL ADDRESS MODE	REAL ADDRESS MODE	PROTECTED VIRTUAL ADDRESS MODE
Immediate Data and Register/Memory	1111011w	mod 000 r/m	data	data if w = 1	3, 6 (Note 59)	3, 6 (Note 59)	2	9
Immediate Data and Accumulator	1010100w	data	data if w = 1		3	3		
OR = Or								
Reg/Memory and Register to Either	000010dw	mod reg r/m			2, 7 (Note 59)	2, 7 (Note 59)	2	9
Immediate to Register/Memory	1000000w	mod 001 r/m	data	data if w = 1	3, 7 (Note 59)	3, 7 (Note 59)	2	9
Immediate to Accumulator	0000110w	data	data if w = 1		3	3		
XOR = Exclusive or								
Reg/Memory and Register to Either	001100dw	mod reg r/m			2, 7 (Note 59)	2, 7 (Note 59)	2	9
Immediate to Register/Memory	1000000w	mod reg r/m	data	data if w = 1	3, 7 (Note 59)	3, 7 (Note 59)	2	9
Immediate to Accumulator	0011010w	data	data if w = 1		3	3		
NOT = Invert Register/Memory	1111011w	mod 010 r/m			2, 7 (Note 59)	2, 7 (Note 59)	2	9
STRING MANIPULATION								
MOVS = Move Byte/Word	1010010w				5	5	2	9
CMPS = Compare Byte/Word	1010011w				8	8	2	9
SCAS = Scan Byte/Word	1010111w				7	7	2	9
LODS = Load Byte/Word to AL/AX	1010110w				5	5	2	9
STOS = Store Byte/Word from AL/A	1010101w				3	3	2	9
INS = Input Byte/Word from DX Port	0110110w				5	5	2	9, 14
OUTS = Output Byte/Word to DX Port	0110111w				5	5	2	9, 14

80C286 Instruction Set Summary **(Continued)**

FUNCTION	FORMAT				CLOCK COUNT		COMMENTS	
					REAL ADDRESS MODE	PROTECTED VIRTUAL ADDRESS MODE	REAL ADDRESS MODE	PROTECTED VIRTUAL ADDRESS MODE
Via Call Gate to Different Privilege Level, X Parameters						86 + 4x + m		8, 11, 12, 18
Via TSS						177 + m		8, 11, 12, 18
Via Task Gate						182 + m		8, 11, 12, 18
Indirect Intersegment	11111111	mod 011 r/m	mod ≠ 11		16 + m (Note 59)	29 + m (Note 59)	2	8, 9, 11, 12, 18
Protected Mode Only (Indirect Intersegment)								
Via Call Gate to Same Privilege Level						44 + m (Note 59)		8, 9, 11, 12, 18
Via Call Gate to Different Privilege Level, No Parameters						83 + m (Note 59)		8, 9, 11, 12, 18
Via Call Gate to Different Privilege Level, X Parameters						90 + 4x + m (Note 59)		8, 9, 11, 12, 18
Via TSS						180 + m (Note 59)		8, 9, 11, 12, 18
Protected Mode Only (Indirect Intersegment) (Continued)								
Via Task Gate						185 + m (Note 59)		8, 9, 11, 12, 18
JMP = Unconditional Jump								
Short/Long	11101011	disp-low			7 + m	7 + m		18
Direct Within Segment	11101001	disp-low	disp-high		7 + m	7 + m		18
Register/Memory Indirect Within Segment	11111111	mod 100 r/m			7 + m, 11 + m (Note 59)	7 + m, 11 + m (Note 59)	2	9, 18
Direct Intersegment	11101010	Segment Offset			11 + m	23 + m		11, 12, 18
Protected Mode Only (Direct Intersegment)		Segment Selector						
Via Call Gate to Same Privilege Level						38 + m		8, 11,12,18
Via TSS						175 + m		8, 11,12,18
Via Task Gate						180 + m		8, 11,12,18
Indirect Intersegment	11111111	mod 101 r/m	mod ≠ 11		15 + m (Note 59)	26 + m (Note 59)	2	8, 9, 11, 12, 18
Protected Mode Only (Indirect Intersegment)								