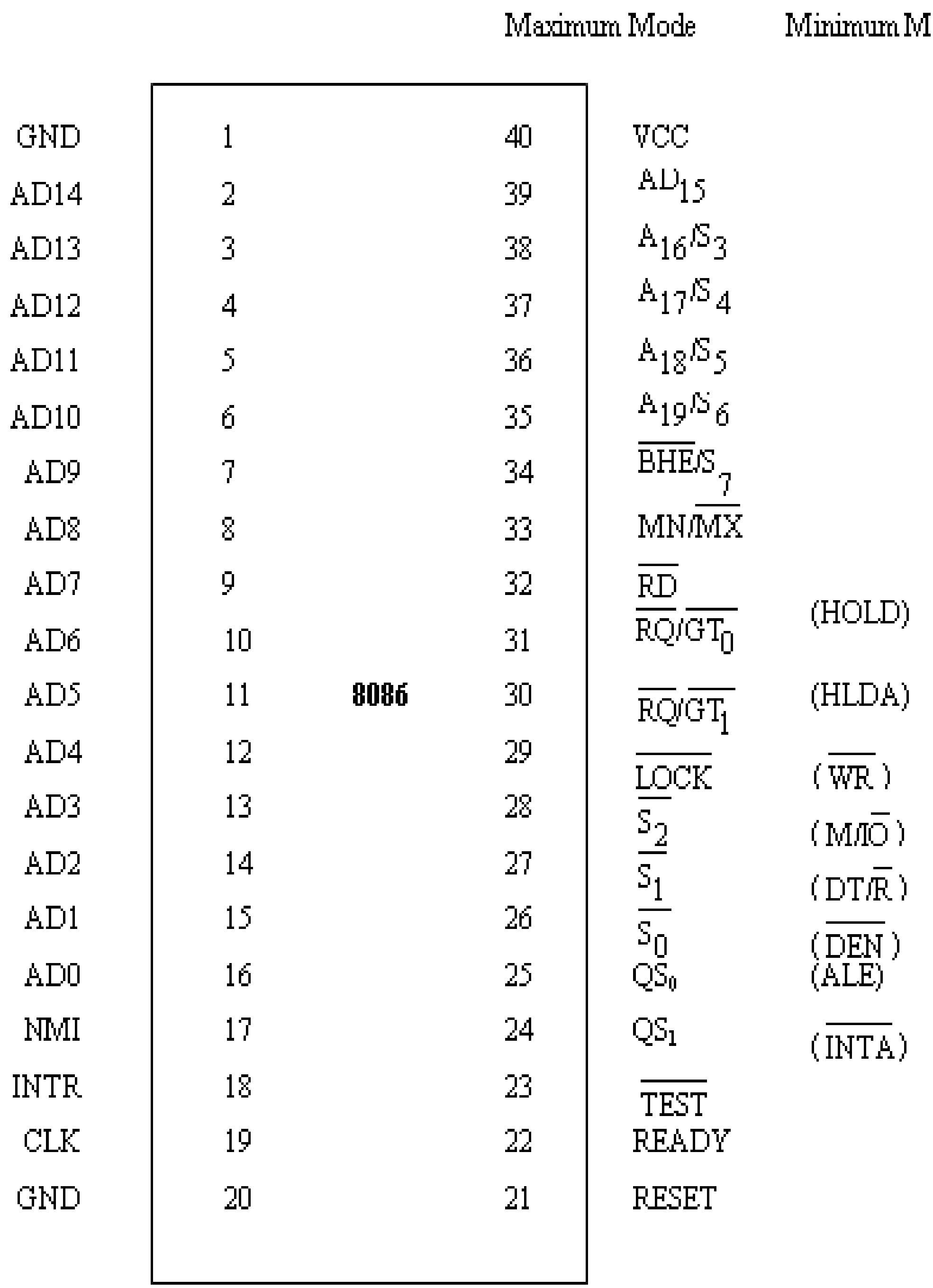
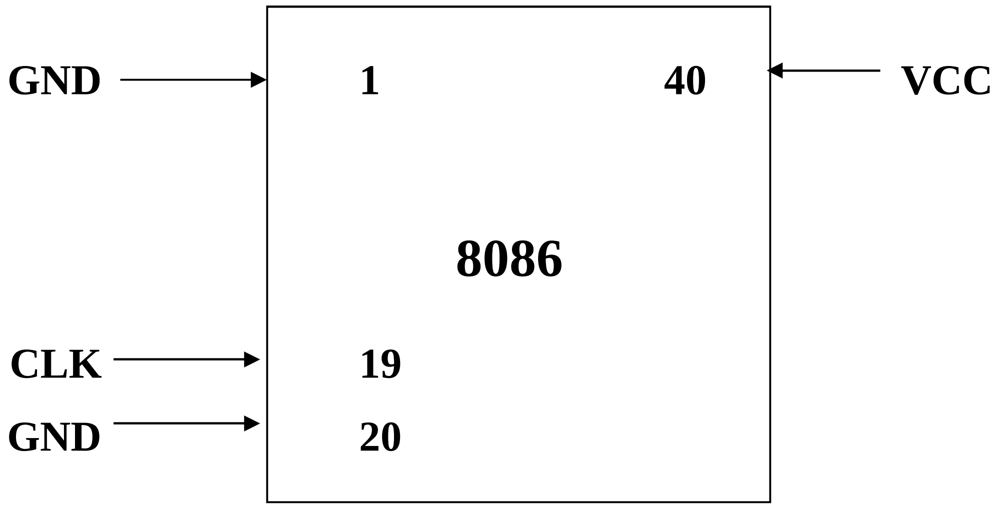
8086 Pin diagram



8086 is a 40 pin DIP using MOS technology. It has 2 GND's as circuit complexity demands a large amount of current flowing through the circuits, and multiple grounds help in dissipating the accumulated heat etc. 8086 works on two modes of operation namely, Maximum Mode and Minimum Mode.

Power Connections



Pin Description:

GND – Pin no. 1, 20

Ground

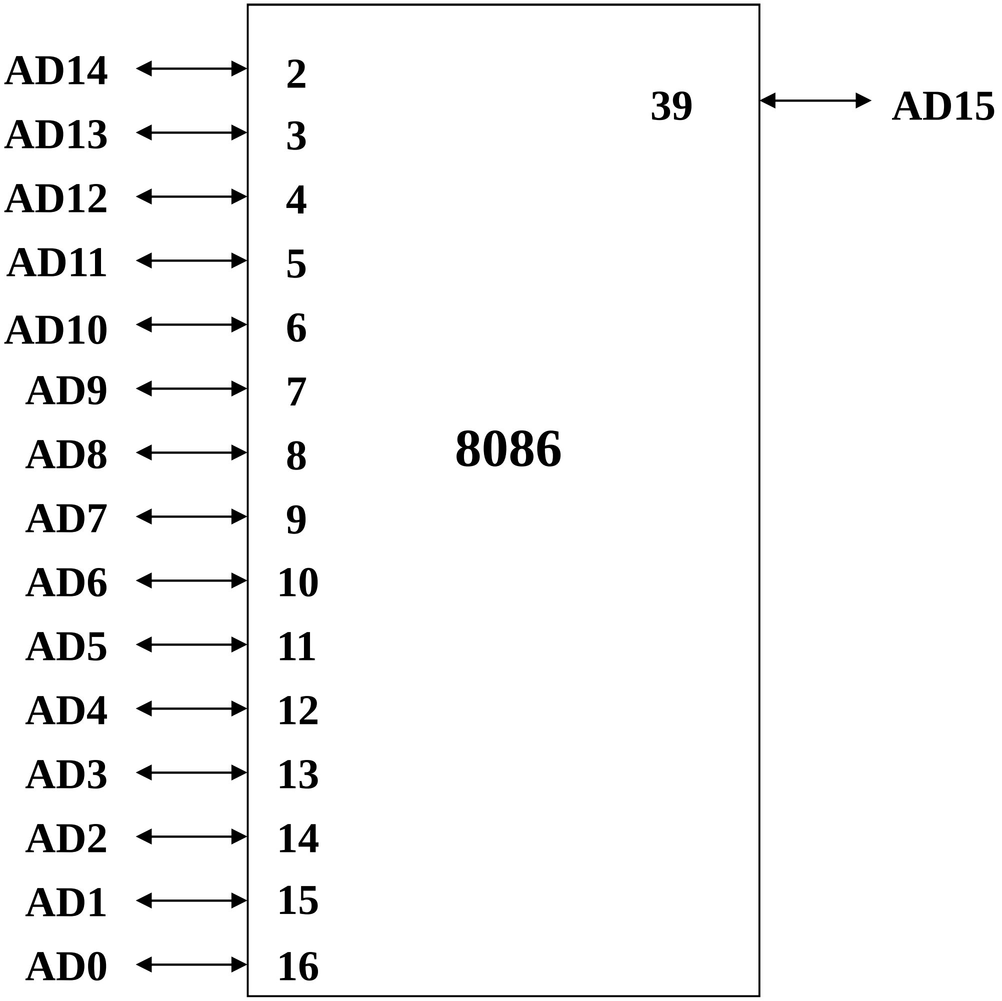
CLK – Pin no. 19 – Type I

Clock: provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.

VCC – Pin no. 40

VCC: +5V power supply pin

Address/ Data Lines



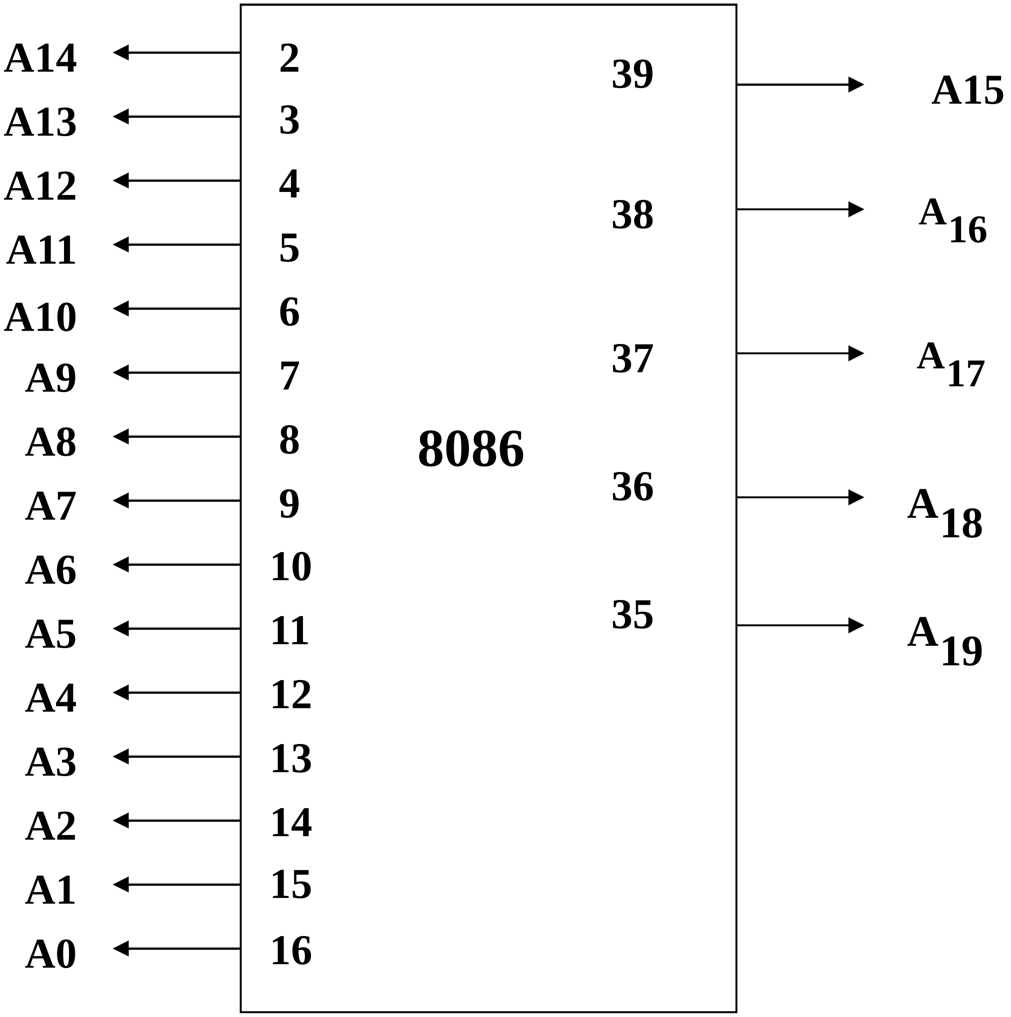
Continued…

Pin Description

AD_{15}-AD_0 – Pin no. 2-16, 39 – Type I/O

Address Data bus: These lines constitute the time multiplexed memory/ IO address (T_1) and data (T_2, T_3, T_W, T_4) bus. A_0 is analogous to BHE* for the lower byte of the data bus, pins D7-D0. It is low when a byte is to be transferred on the lower portion of the bus in memory or I/O operations. Eight –bit oriented devices tied to the lower half would normally use A_0 to condition chip select functions. These lines are active HIGH

Address Lines



Continued…

A_{19}/S_6, A_{18}/S_5, A_{17}/S_4, A_{16}/S_3 – Pin no. 35-38 – Type O

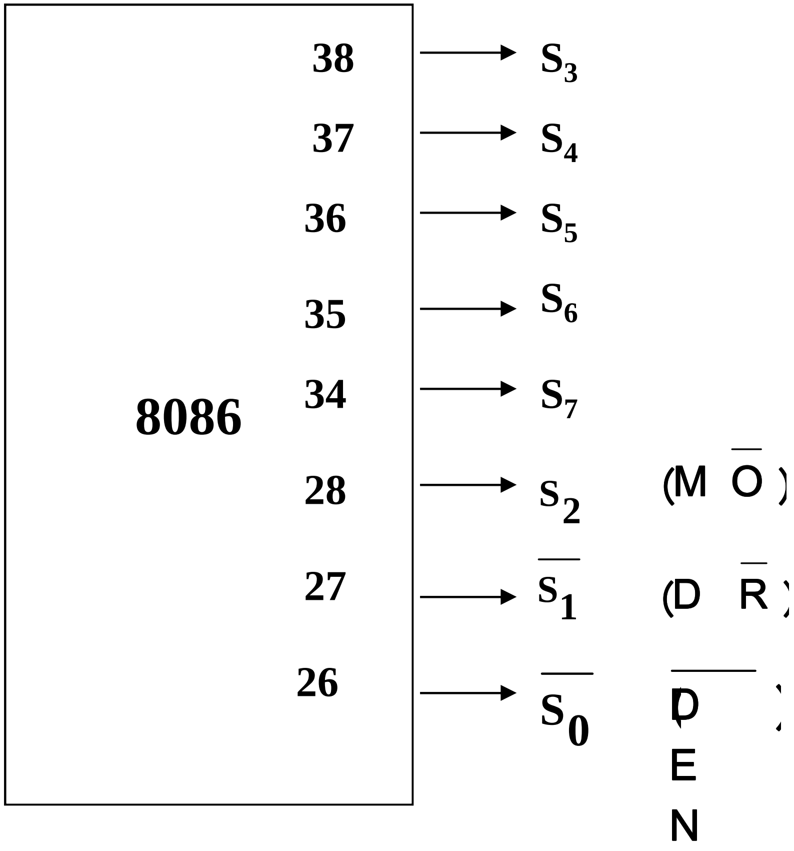
Address / Status: During T_1 these are the four most significant address lines for memory operations. During I/O operations these lines are low. During memory and I/O operations, status information is available on these lines during T_2, T_3, T_W and T_4. The status of the interrupt enable FLAG bit (S_5) is updated at the beginning of each CLK cycle. A_{17}/S_4 and A_{16}/S_3 are encoded as shown.

Continued…

A_{17}/S_4	**A_{16}/S_3**	**Characteristics**
0 (LOW)	0	Alternate Data
0	1	Stack
1(HIGH)	0	Code or None
1	1	Data
S_6 is 0 (LOW)		

This information indicates which relocation register is presently being used for data accessing.

Status Pins S_0-S_7



Continued…

Pin Description

$\overline{S_2}$, $\overline{S_1}$, $\overline{S_0}$ - Pin no. 26, 27, 28 – Type O

Status: active during T_4, T_1 and T_2 and is returned to the passive state (1,1,1) during T_3 or during TW when READY is HIGH. This status is used by the 8288 Bus Controller to generate all memory and I/O access control signals. Any change by , or during T_4 is used to indicate the beginning of a bus cycle and the return to the passive state in T_3 or T_W is used to indicate the end of a bus cycle.

Continued…

These signals float to 3-state OFF in “hold acknowledge”. These status lines are encoded as shown.

S_2*	S_1*	S_0*	**Characteristics**
0(LOW)	0	0	Interrupt acknowledge
0	0	1	Read I/O Port
0	1	0	Write I/O Port
0	1	1	Halt
1(HIGH)	0	0	Code Access
1	0	1	Read Memory
1	1	0	Write Memory
1	1	1	Passive

Continued…

Status Details

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	**Indication**
0	0	0	Interrupt Acknowledge
0	0	1	Read I/O port
0	1	0	Write I/O port
0	1	1	Halt
1	0	0	Code access
1	0	1	Read memory
1	1	0	Write memory
1	1	1	Passive

Continued...

S_4	S_3	**Indications**
0	0	Alternate data
0	1	Stack
1	0	Code or none
1	1	Data

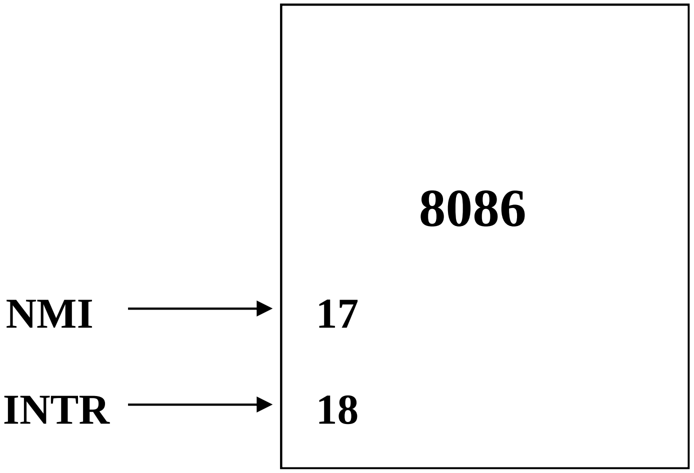
Continued…

$\overline{S_5}$ ----- Value of Interrupt Enable flag

$\overline{S_6}$ ----- Always low (logical) indicating 8086 is on the bus. If it is tristated another bus master has taken control of the system bus.

$\overline{S_7}$ ----- Used by 8087 numeric coprocessor to determine whether the CPU is a 8086 or 8088

Interrupts



Pin Description:

NMI – Pin no. 17 – Type I

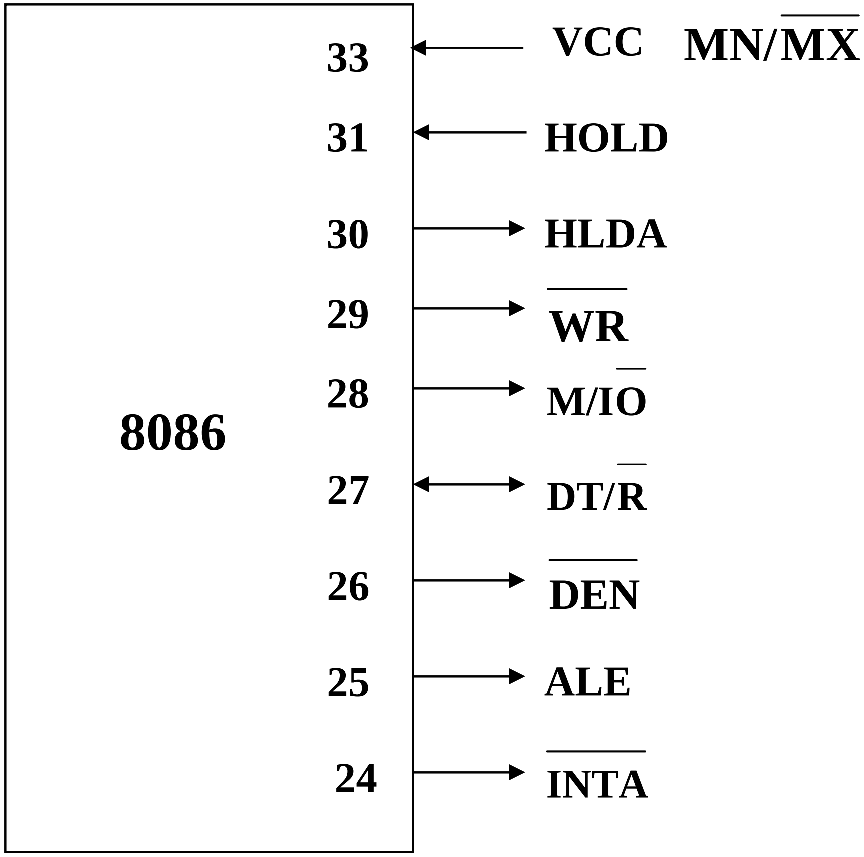
Non – Maskable Interrupt: an edge triggered input which causes a type 2 interrupt. A subroutine is vectored to via an interrupt vector lookup table located in system memory. NMI is not maskable internally by software. A transition from a LOW to HIGH initiates the interrupt at the end of the current instruction. This input is internally synchronized.

Continued…

INTR – Pin No. 18 – Type I

Interrupt Request: is a level triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. A subroutine is vectored to via an interrupt vector lookup table located in system memory. It can be internally masked by software resetting the interrupt enable bit. INTR is internally synchronized. This signal is active HIGH.

Min mode signals



Continued…

Pin Description

HOLD, HLDA – Pin no. 31, 30 – Type I/O

HOLD: indicates that another master is requesting a local bus “hold”. To be acknowledged, HOLD must be active HIGH. The processor receiving the “hold” request will issue HLDA (HIGH) as an acknowledgement in the middle of a T1 clock cycle. Simultaneous with the issuance of HLDA the processor will float the local bus and control lines. After HOLD is detected as being LOW, the processor will LOWer the HLDA, and when the processor needs to run another cycle, it will again drive the local bus and control lines.

The same rules as apply regarding when the local bus will be released.

HOLD is not an asynchronous input. External synchronization should be provided if the system can not otherwise guarantee the setup time.

Continued…

WR* - Pin no. 29 – Type O

Write: indicates that the processor is performing a write memory or write I/O cycle, depending on the state of the M/IO* signal. WR* is active for T_2, T_3 and T_W of any write cycle. It is active LOW, and floats to 3-state OFF in local bus "hold acknowledge".

M/IO* - Pin no. 28 – type O

Status line: logically equivalent to S_2 in the maximum mode. It is used to distinguish a memory access from an I/O access. M/IO* becomes valid in the T_4 preceding a bus cycle and remains valid until the final T_4 of the cycle (M=HIGH), IO=LOW). M/IO* floats to 3-state OFF in local bus "hold acknowledge".

Continued…

DT/R* - Pin no. 27 – Type O

Data Transmit / Receive: needed in minimum system that desires to use an 8286/8287 data bus transceiver. It is used to control the direction of data flow through the transceiver. Logically DT/R* is equivalent to S_1* in the maximum mode, and its timing is the same as for M/IO*. (T=HIGH, R=LOW). This signal floats to 3-state OFF in local bus "hold acknowledge".

DEN* - Pin no. 26 – Type O

Data Enable: provided as an output enable for the 8286/8287 in a minimum system which uses the transceiver. DEN* is active LOW during each memory and I/O access and for INTA cycles. For a read or INTA* cycle it is active from the middle of T_2 until the middle of T_4, while for a write cycle it is active from the beginning of T_2 until the middle of T_4. DEN* floats to 3-state OFF in local bus "hold acknowledge".

Continued…

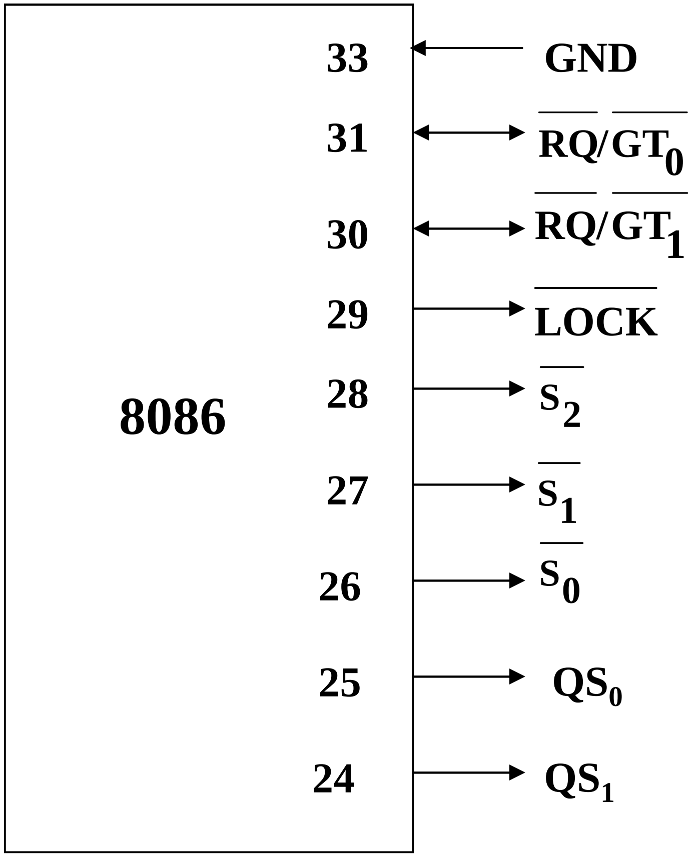
ALE – Pin no. 25 – Type O

Address Latch Enable: provided by the processor to latch the address into the 8282/8283 address latch. It is a HIGH pulse active during T1 of any bus cycle. Note that ALE is never floated.

INTA* - Pin no. 24 – Type O

INTA* is used as a read strobe for interrupt acknowledge cycles. It is active LOW during T_2, T_3 and T_W of each interrupt acknowledge cycle.

Max mode signals



Continued…

Pin Description:

RQ*/GT0*, RQ*/GT1* - **Pin no. 30, 31 – Type I/O**

Request /Grant: pins are used by other local bus masters to force the processor to release the local bus at the end of the processor’s current bus cycle. Each pin is bidirectional with RQ*/GT0* having higher priority than RQ*/GT1*. RQ*/GT* has an internal pull up resistor so may be left unconnected. The request/grant sequence is as follows:

Continued…

1. A pulse of 1 CLK wide from another local bus master indicates a local bus request (“hold”) to the 8086 (pulse 1)
2. During a T4 or T1 clock cycle, a pulse 1 CLK wide from the 8086 to the requesting master (pulse 2), indicates that the 8086 has allowed the local bus to float and that it will enter the “hold acknowledge” state at the next CLK. The CPU’s bus interface unit is disconnected logically from the local bus during “hold acknowledge”.
3. A pulse 1 CLK wide from the requesting master indicates to the 8086 (pulse 3) that the “hold” request is about to end and that the 8086 can reclaim the local bus at the next CLK.

Continued…

Each master-master exchange of the local bus is a sequence of 3 pulses. There must be one dead CLK cycle after each bus exchange. Pulses are active LOW.

If the request is made while the CPU is performing a memory cycle, it will release the local bus during T_4 of the cycle when all the following conditions are met:

- Request occurs on or before T_2.
- Current cycle is not the low byte of a word (on an odd address)
- Current cycle is not the first acknowledge of an interrupt acknowledge sequence.
- A locked instruction is not currently executing.

Continued…

LOCK* - Pin no. 29 – Type O

LOCK* : output indicates that other system bus masters are not to gain control of the system bus while LOCK* is active LOW. The LOCK* signal is activated by the “LOCK” prefix instruction and remains active until the completion of the next instruction. This signal is active LOW, and floats to 3-state OFF in “hold acknowledge”.

QS_1, QS_0 – Pin no. 24, 25 – Type O

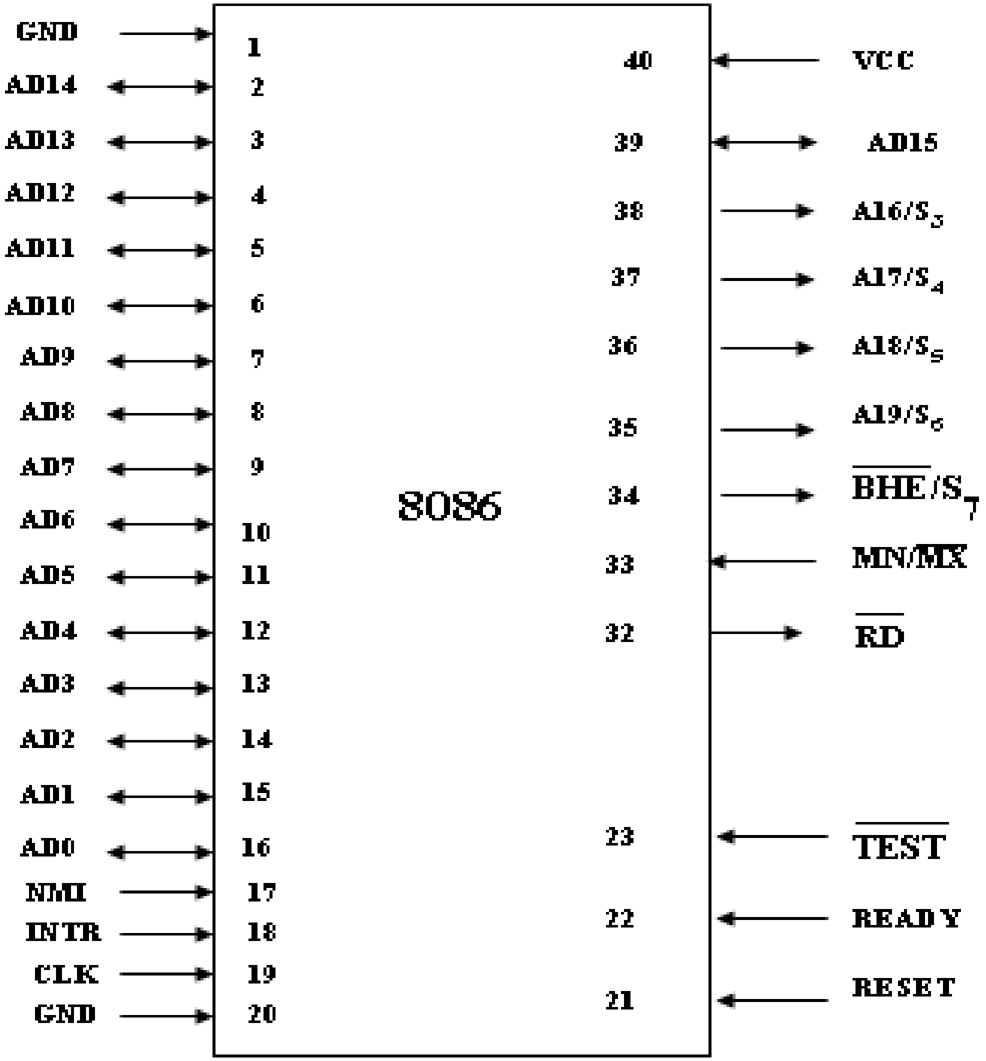
Queue Status: the queue status is valid during the CLK cycle after which the queue operation is performed.

QS_1 and QS_0 provide status to allow external tracking of the internal 8086 instruction queue.

Continued...

QS_1	**QS_0**	**Characteristics**
0(LOW)	0	No operation
0	1	First Byte of Op Code from Queue
1 (HIGH)	0	Empty the Queue
1	1	Subsequent byte from Queue

Common Signals



Continued…

Pin Description:

RD* - Pin no. 34, Type O

Read: Read strobe indicates that the processor is performing a memory of I/O read cycle, depending on the state of the S2 pin. This signal is used to read devices which reside on the 8086 local bus. RD* is active LOW during T_2, T_3 and T_W of any read cycle, and is guaranteed to remain HIGH in T_2 until the 8086 local bus has floated.

This signal floats to 3-state OFF in “hold acknowledge”.

READY – Pin no. 22, Type I

READY: is the acknowledgement from the addressed memory or I/O device that it will complete the data transfer. The READY signal from memory / IO is synchronized by the 8284A Clock Generator to form READY. This signal is active HIGH. The 8086 READY input is not synchronized. Correct operation is not guaranteed if the setup and hold times are not met.

Continued…

TEST* - Pin No 23 – Type I

TEST* : input is examined by the “Wait” instruction. If the TEST* input is LOW execution continues, otherwise the processor waits in an “idle” state. This input is synchronized internally during each clock cycle on the leading edge of CLK.

RESET – Pin no. 21 – Type I

Reset: causes the processor to immediately terminate its present activity. The signal must be active HIGH for at least four clock cycles. It restarts execution, as described in the instruction set description, when RESET returns LOW. RESET is internally synchronized.

Continued…

BHE*/S7- Pin No. 34 – Type O

Bus High Enable / Status: During T1 the Bus High Enable signal (BHE*) should be used to enable data onto the most significant half of the data bus, pins D15-D8. Eight bit oriented devices tied to the upper half of the bus would normally use BHE* to condition chip select functions. BHE* is LOW during T1 for read, write, and interrupt acknowledge cycles when a byte is to be transferred on the high portion of the bus. The S,7 status information is available during T2, T3 and T4. The signal is active LOW and floats to 3-state OFF in “hold”. It is LOW during T1 for the first interrupt acknowledge cycle.

BHE*	**A_0**	**Characteristics**
0	0	Whole word
0	1	Upper byte from / to odd address
1	0	Lower byte from / to even address
1	1	None

Continued…

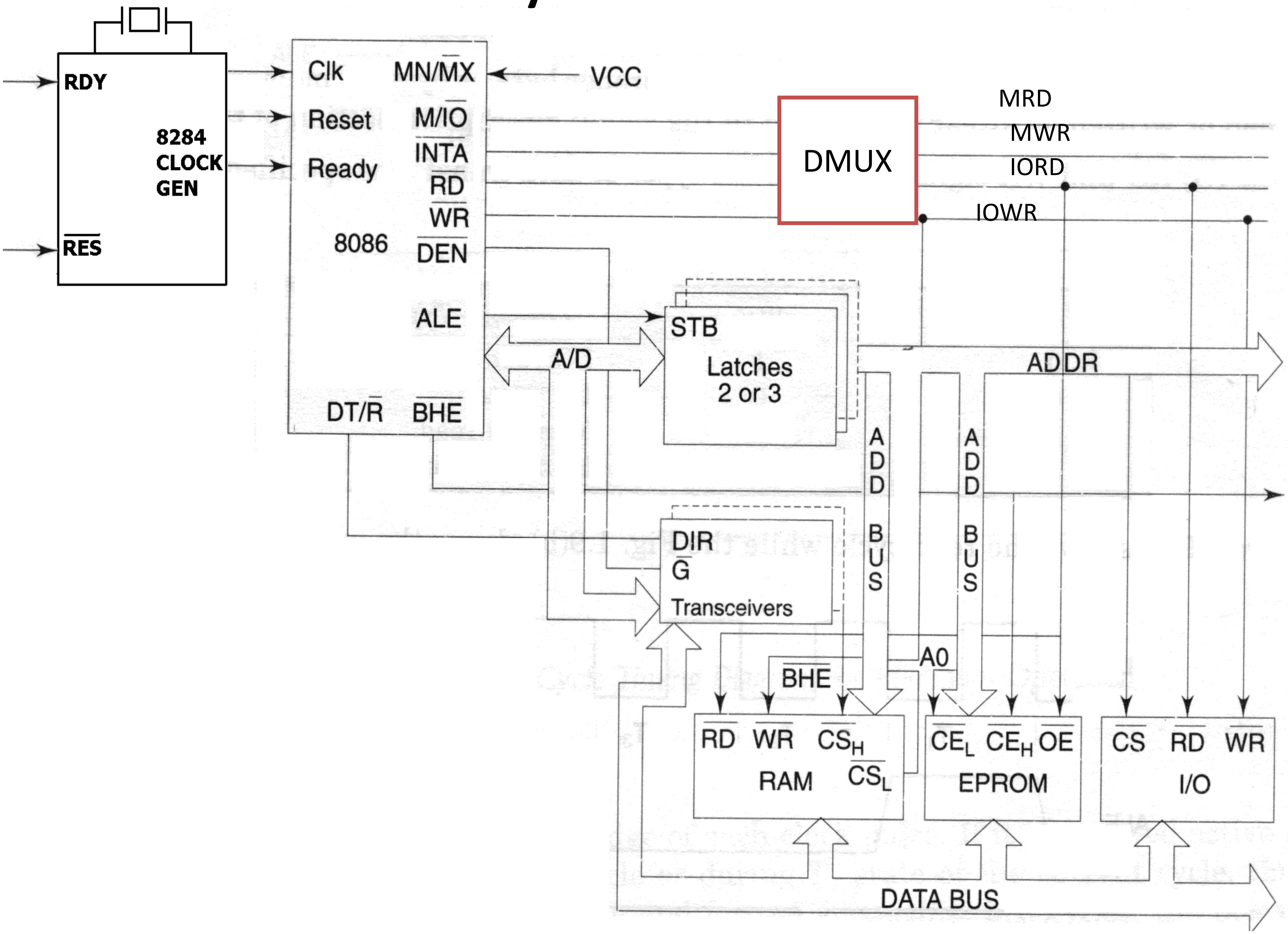
MN/MX* - Pin no. 33 – Type - I

Minimum / Maximum: indicates what mode the processor is to operate in.

If the local bus is idle when the request is made the two possible events will follow:

- ➢ Local bus will be released during the next clock.
- ➢ A memory cycle will start within 3 clocks. Now the four rules for a currently active memory cycle apply with condition number 1 already satisfied.

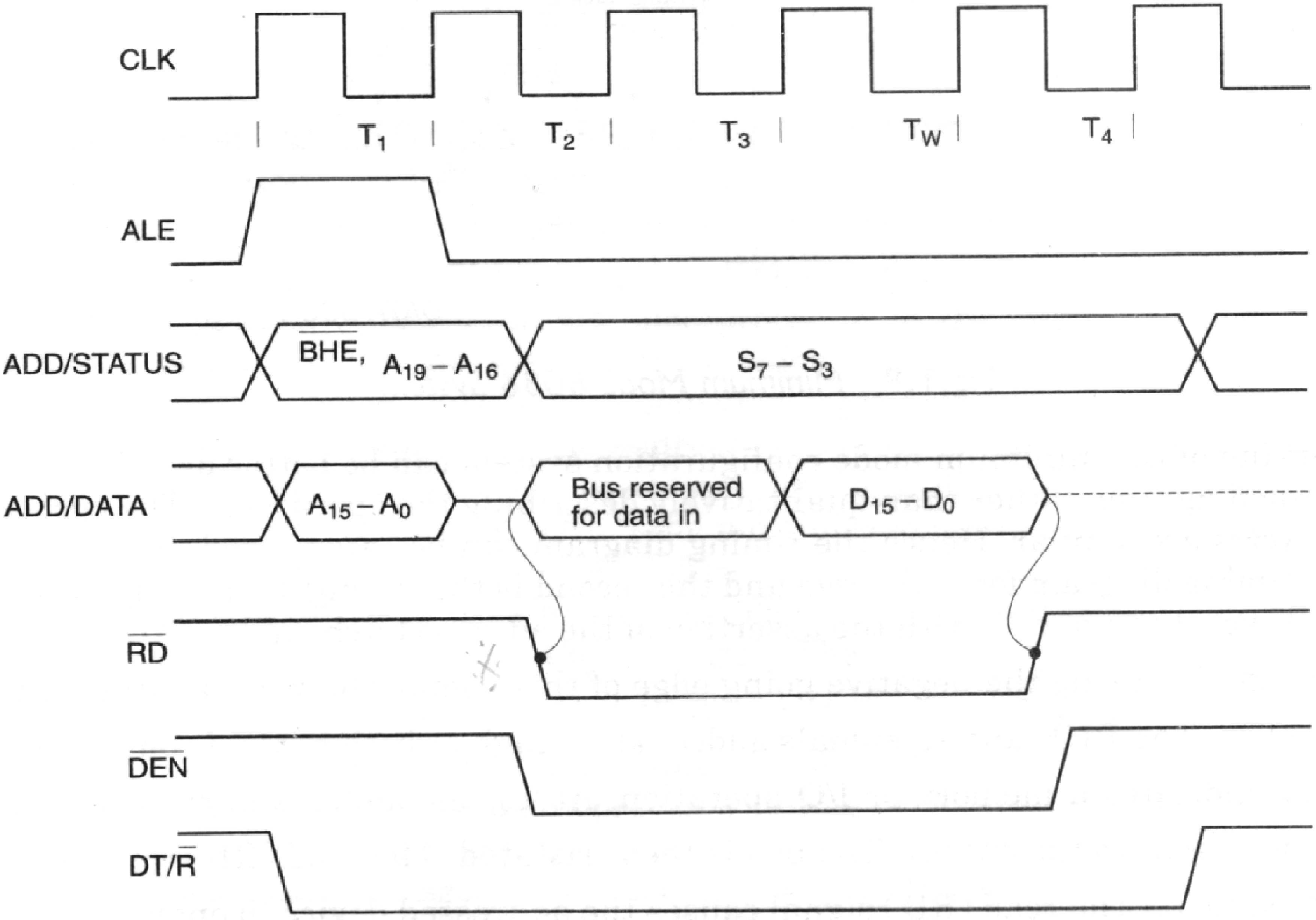
Minimum Mode 8086 System



A minimum mode of 8086 configuration depicts a stand alone system of computer where no other processor is connected. This is similar to 8085 block diagram with the following difference.

The Data transceiver block which helps the signals traveling a longer distance to get boosted up. Two control signals data transmit/ receive are connected to the direction input of transceiver (Transmitter/Receiver) and DEN* signal works as enable for this block.

Read Cycle timing Diagram for Minimum Mode

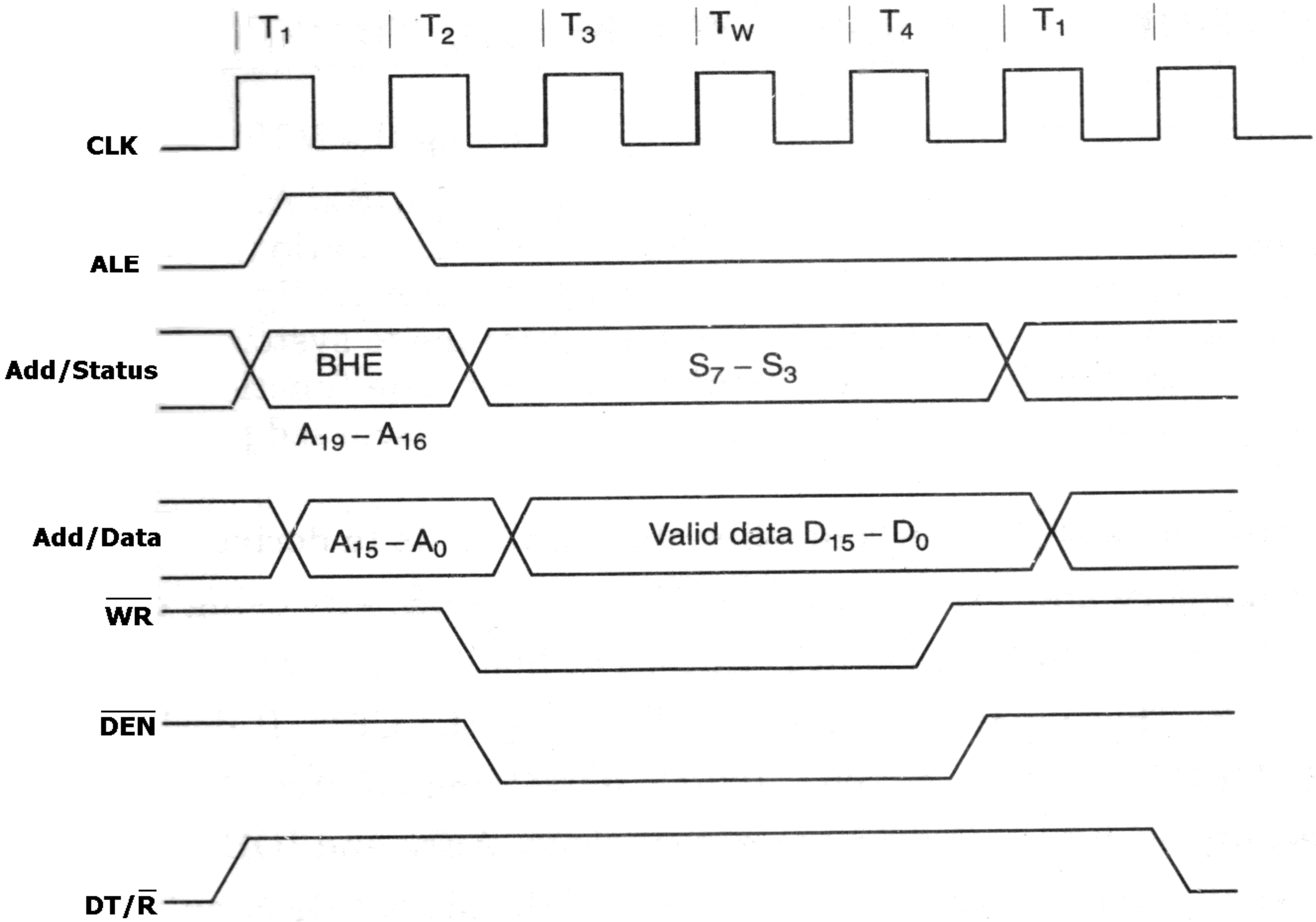


In the bus timing diagram, data transmit / receive signal goes low (RECEIVE) for Read operation. To validate the data, DEN* signal goes low. The Address/ Status bus carries A16 to A19 address lines during BHE* (low) and for the remaining time carries Status information. The Address/Data bus carries A0 to A15 address information during ALE going high and for the remaining time it carries data. The RD* line going low indicates that this is a Read operation. The curved arrows indicate the relationship between valid data and RD* signal.

Continued…

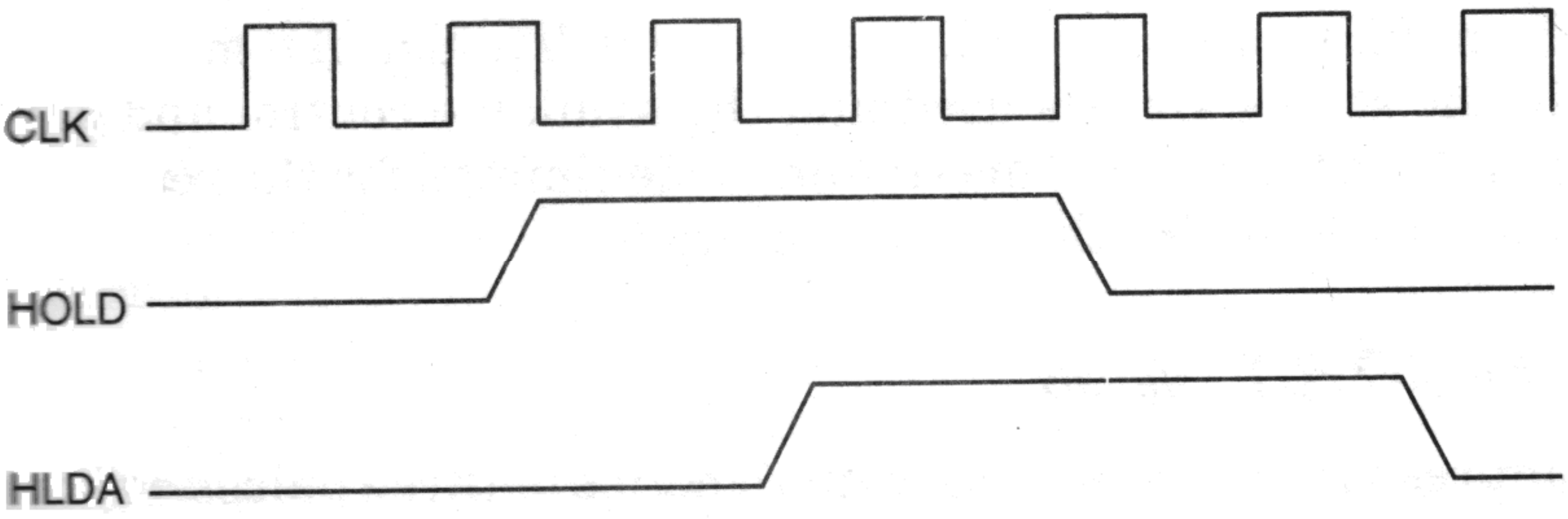
The TW is Wait time needed to synchronize the fast processor with slow memory etc. The Ready pin is checked to see whether any peripheral needs more time for data transmission.

Write Cycle timing Diagram for Minimum Operation



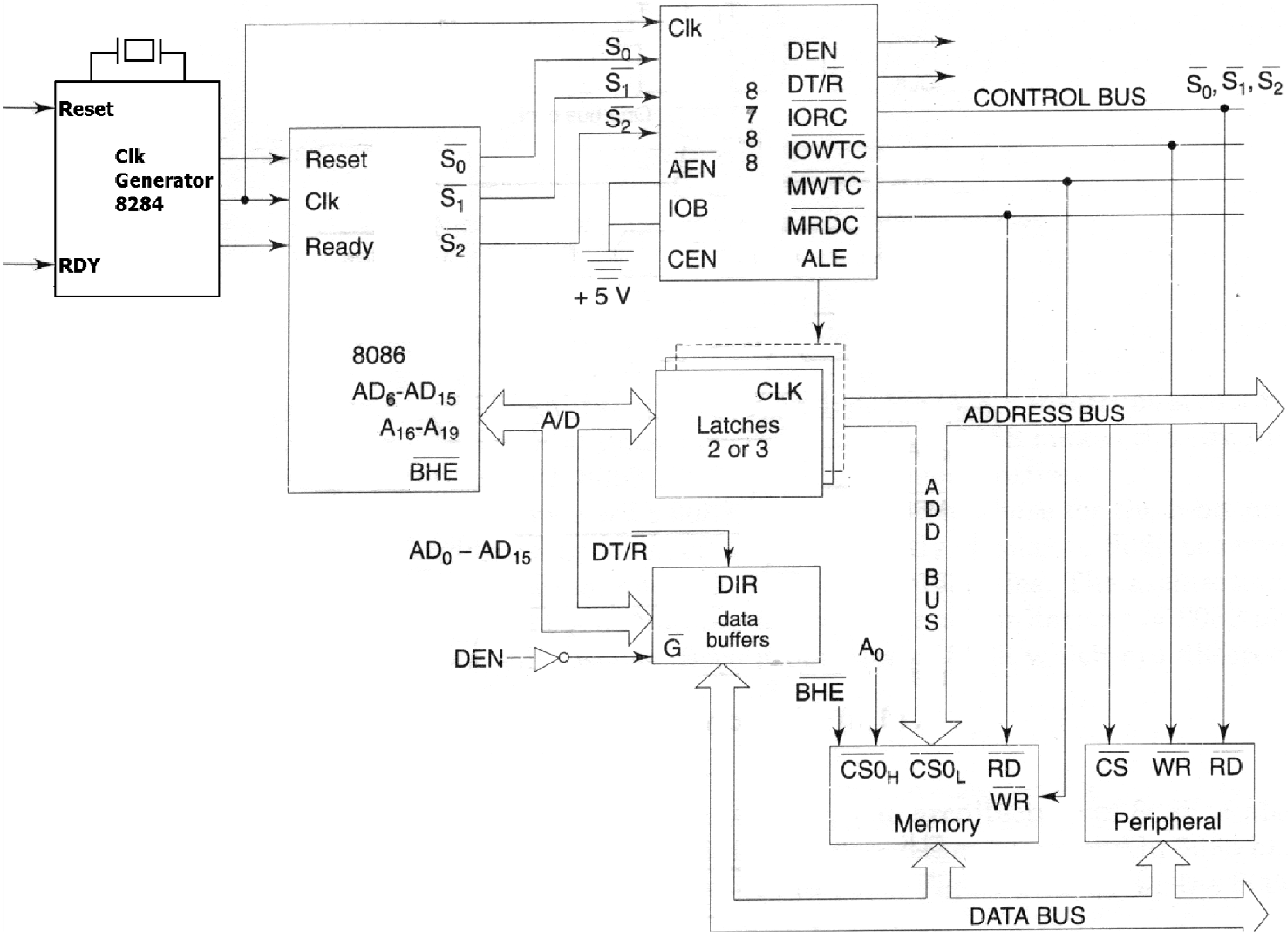
This is the same as Read cycle Timing Diagram except that the DT/R* line goes high indicating it is a Data Transmission operation for the processor to memory / peripheral. Again DEN* line goes low to validate data and WR* line goes low, indicating a Write operation.

Bus Request & Bus Grant Timings in Minimum Mode System



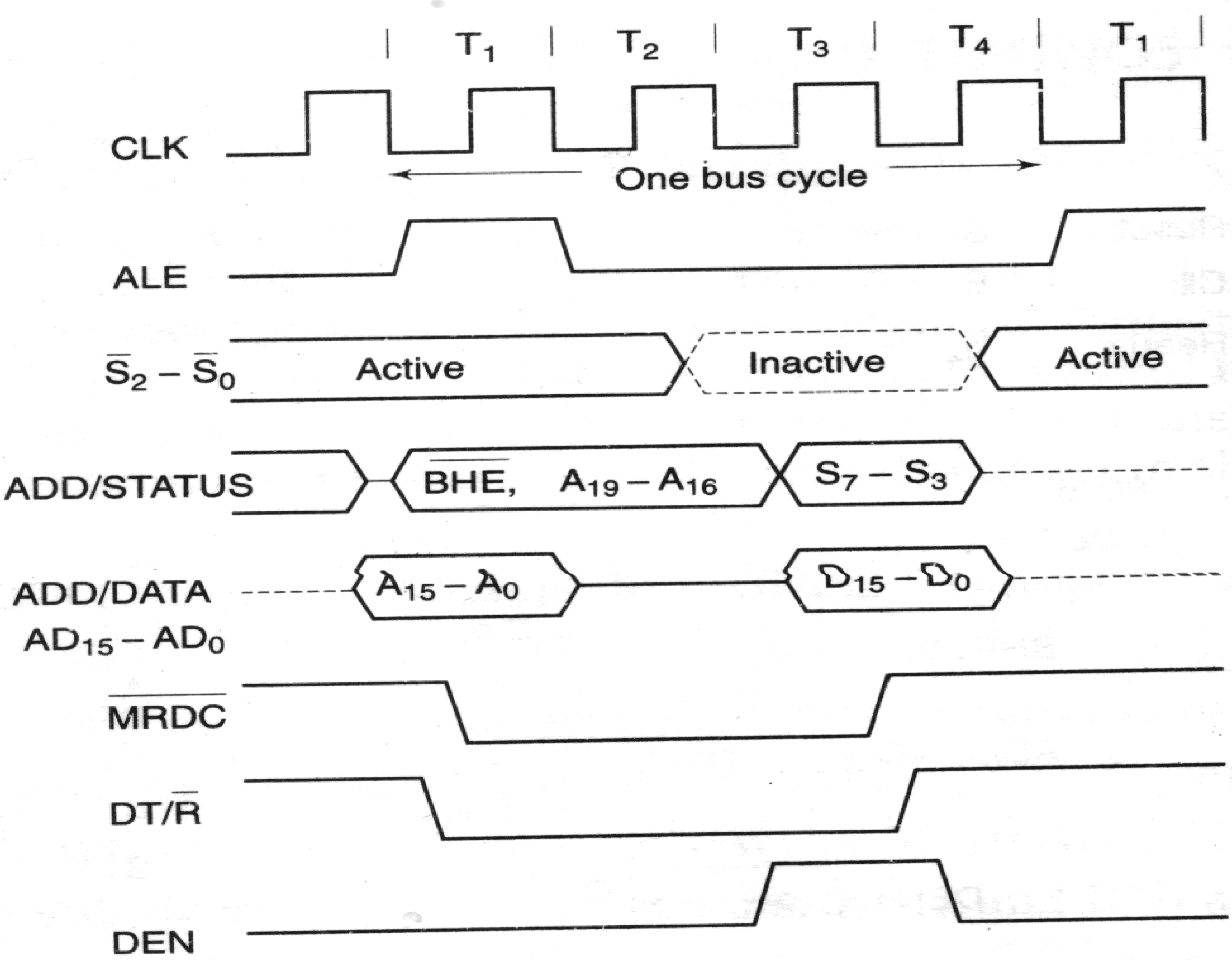
The HOLD and HLDA timing diagram indicates in Time Space HOLD (input) occurs first and then the processor outputs HLDA (Hold Acknowledge).

Maximum Mode 8086 System



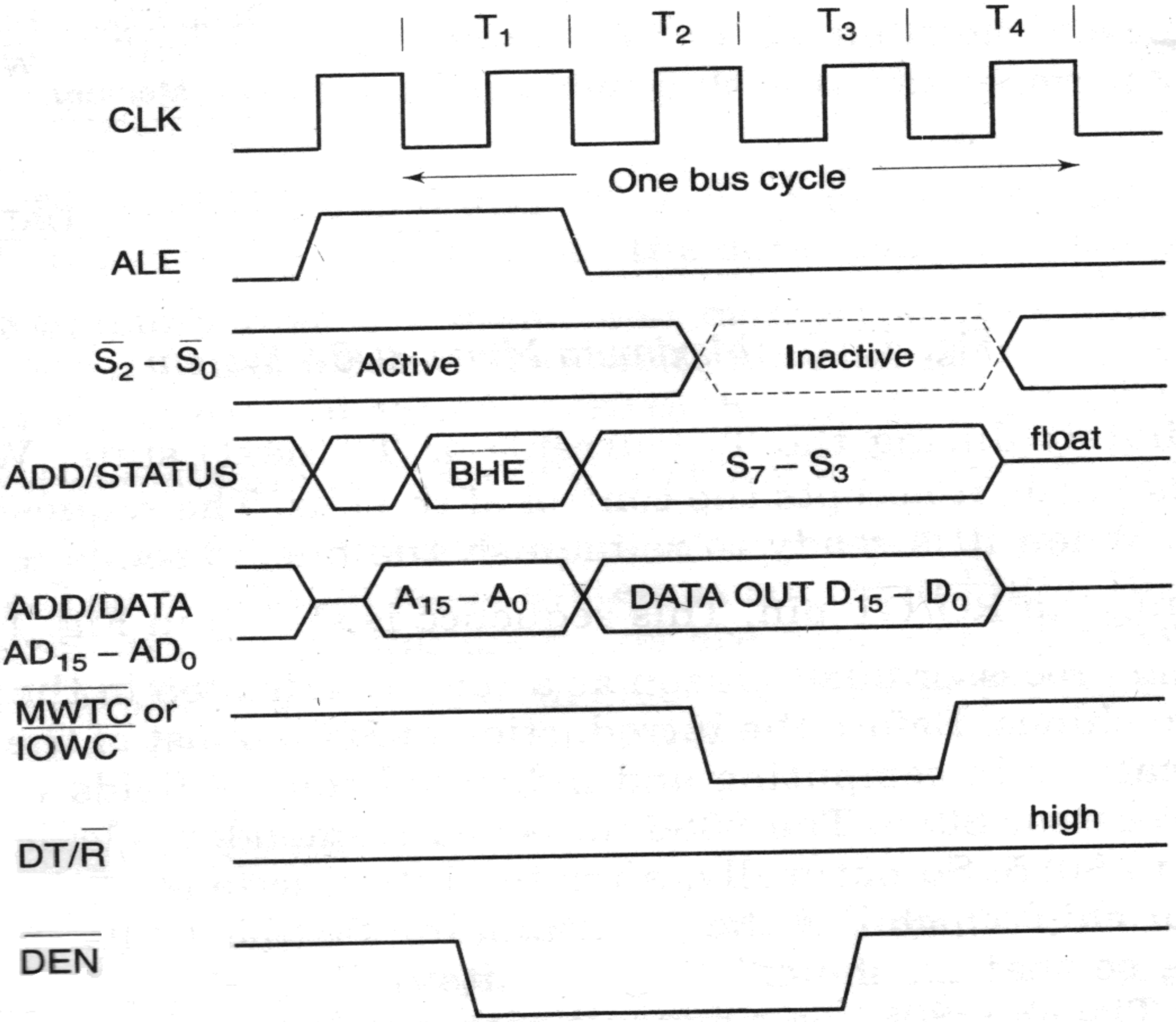
In the maximum mode of operation of 8086, wherein either a numeric coprocessor of the type 8087 or another processor is interfaced with 8086. The Memory, Address Bus, Data Buses are shared resources between the two processors. The control signals for Maximum mode of operation are generated by the Bus Controller chip 8788. The three status outputs S0*, S1*, S2* from the processor are input to 8788. The outputs of the bus controller are the Control Signals, namely DEN, DT/R*, IORC*, IOWTC*, MWTC*, MRDC*, ALE etc. These control signals perform the same task as the minimum mode operation. However the DEN is an active HIGH signal which has to be converted to active LOW by means of an inverter.

Memory Read timing in Maximum Mode



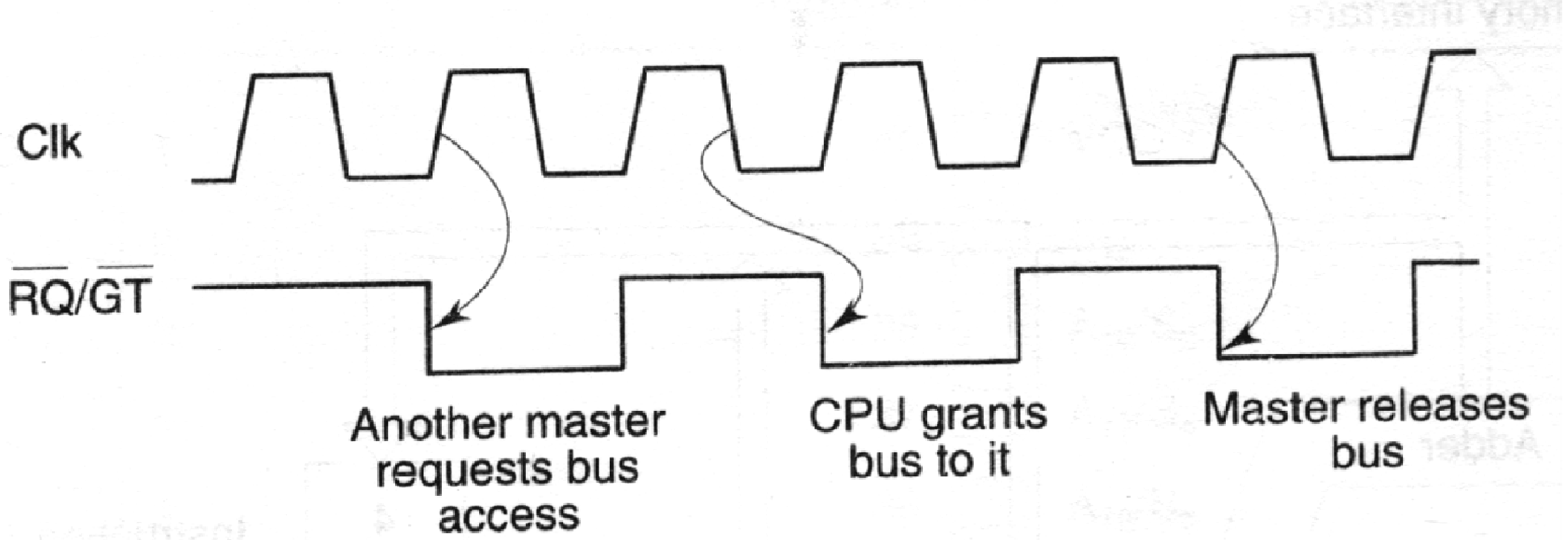
Here MRDC* signal is used instead of RD* as in case of Minimum Mode S0* to S2* are active and are used to generate control signal.

Memory Write Timing in Maximum Mode



Here the maximum mode write signals are shown. Please note that the T states correspond to the time during which DEN* is LOW, WRITE Control goes LOW, DT/R* is HIGH and data output in available from the processor on the data bus.

RQ*/ GT* Timings in Maximum Mode


Clk
$\overline{RQ}/\overline{GT}$
Another master requests bus access
CPU grants bus to it
Master releases bus

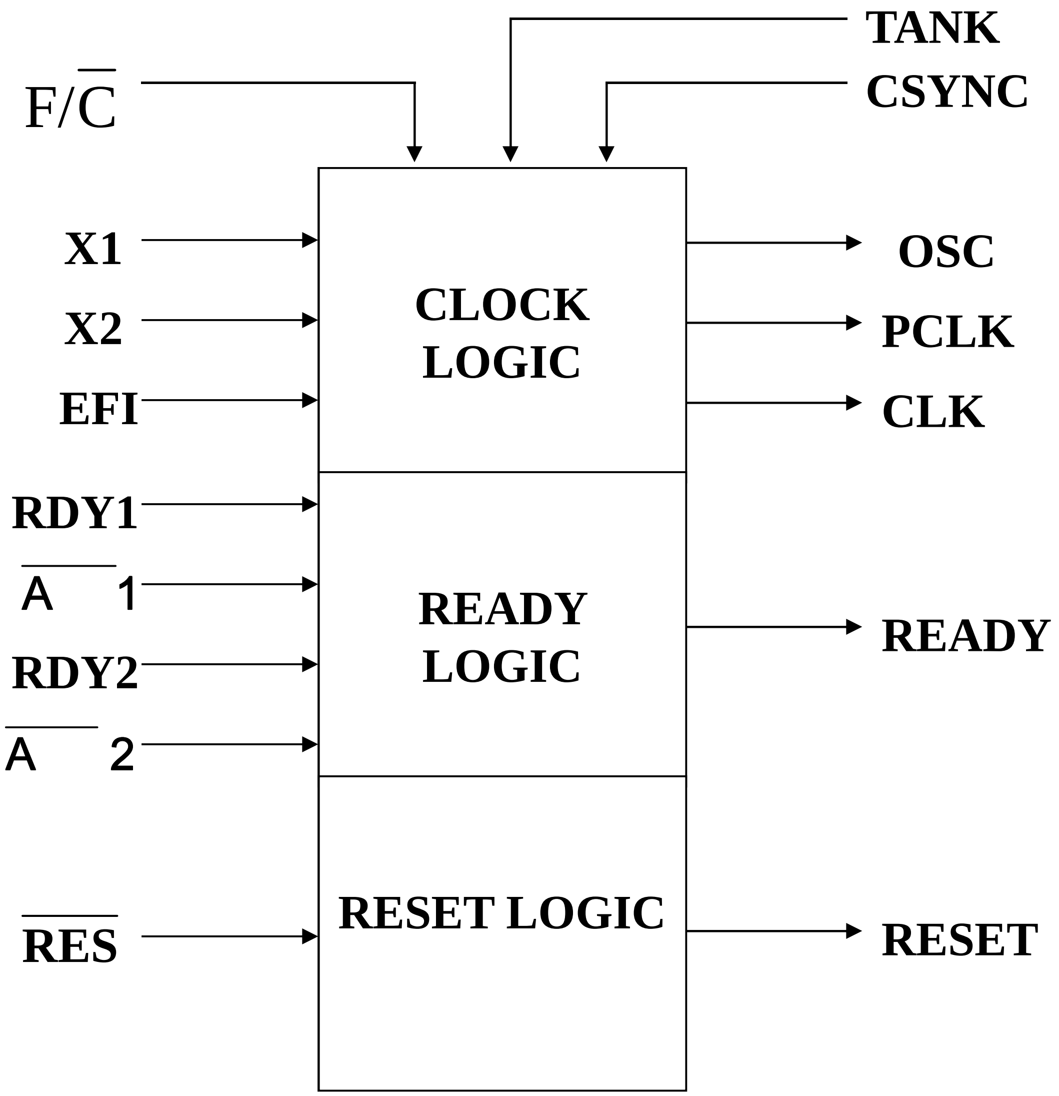
Request / Grant pin may appear that both signals are active low. But in reality, Request signal goes low first (input to processor), and then the processor grants the request by outputting a low on the same pin.

8284 Clock Generator

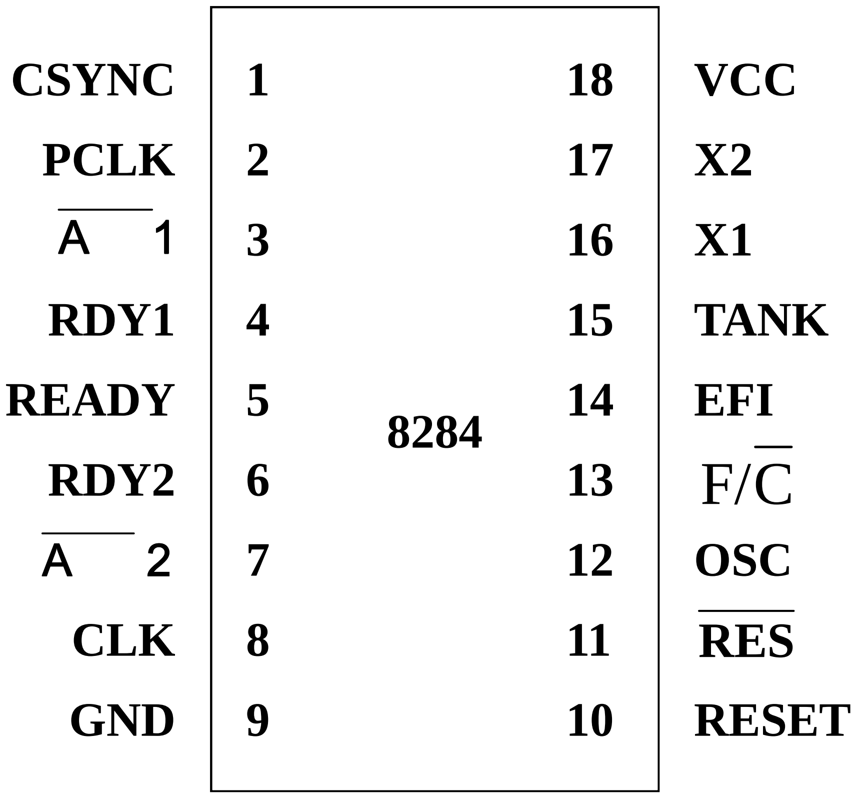
The clock Generator 8284 performs the following tasks in addition to generating the system clock for the 8086/8088.

- ✓ Generating the Ready signal for h 8086/8088
- ✓ Generating the Reset signal for h 8086/8088

8284 Block Diagram

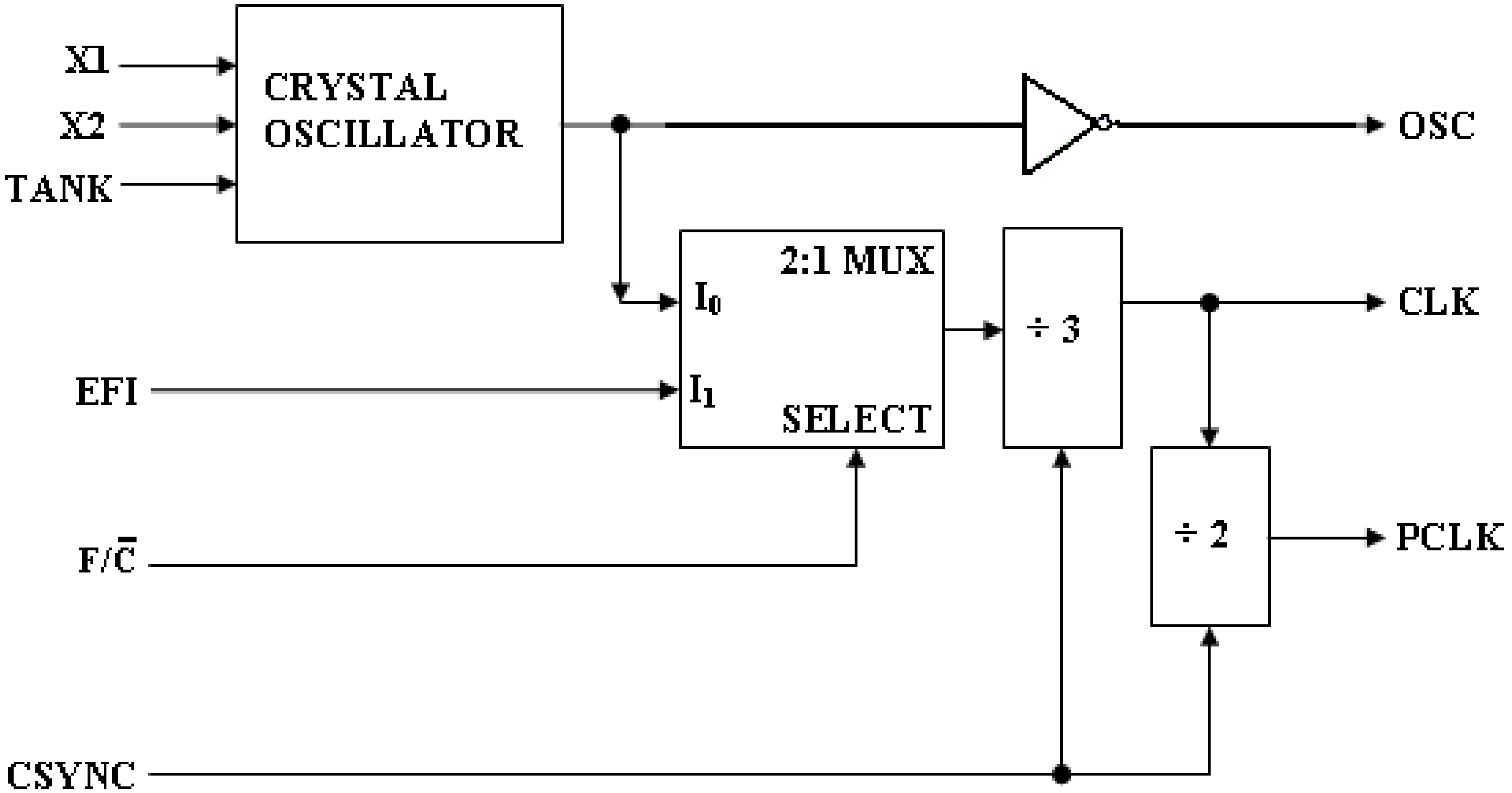


8284 Pin Diagram



CSYNC	1	18	VCC
PCLK	2	17	X2
$\overline{\text{A 1}}$	3	16	X1
RDY1	4	15	TANK
READY	5	14	EFI
RDY2	6	13	$F/\overline{C}$
$\overline{\text{A 2}}$	7	12	OSC
CLK	8	11	$\overline{\text{RES}}$
GND	9	10	RESET

Clock Logic



Continued…

The clock logic generates the three output signals OSC, CLOCK, and PCLK.

OSC:

OSC is a TTL clock signal generated by the crystal oscillator in 8284. Its frequency is same as the frequency of the crystal connected between X1 and X2 pins of 8284. In a PC, a crystal of 14.31 MHz is connected between X1 and X2. thus OSC output frequency will be 14.31MHz. This signal is used by the Color Graphics Adapter (CGA). The Tank input is used by the crystal oscillator only if the crystal is an overtone type crystal. An LC circuit is connected to the TANK input to tune the oscillator to the overtone frequency of the crystal. Generally, in PCs, the TANK input is connected to ground, as fundamental type crystal is used in a PC.

Continued…

Clock:

The Clock output of 8284 is used as the system clock for the 8086/8088, 8087, and the bus controller 8288. It is having a duty cycle of 33%. It is derived from the OSC frequency generated by the crystal oscillator, or from an External Frequency Input (EFI). These two signals are inputs to a multiplexer. The F/C* (external frequency/crystal) input to the multiplexer decides this aspect. If F/C*=0, OSC frequency is used for deriving Clock. If F/C*=1, EFI input is used for deriving clock. The output of the multiplexer, which is OSC or EFI, is divided by 3 to provide the Clock output. Thus, if F/C*=0, clock frequency will be 14.31MHz/3=4.77MHz.

Continued…

Turbo PCs use 30MHz crystal oscillator circuit for generating EFI input. With F/C*=1, they allow turbo clock speed of 10MHz. Such PCs provide a choice of switching between 4.77MHz and 10MHz using a toggle switch or manual operation. The switching can also be controlled by software using an output port.

The CSYNC input is a synchronization signal for synchronizing multiple 8284s in a system. In a PC, CSYNC is tied to ground, as there is a single 8284.

Continued…

PCLK:

PCLK frequency output is obtained by dividing clock frequency by 2. PCLK is used by dividing clock frequency by 2. PCLK is used by support chips like 8254 timer, which need a lower frequency for their operation.

Continued…

Pin functions of 8284A:

X1 and X2 The **Crystal Oscillator** pins connect to an external crystal used as the timing source for the clock generator and all its functions.

EFI The **External Frequency** input is used when the F/C is pulled high. EFI supplies the timing whenever the F/C* pin is high.

F/C* The **Frequency/Crystal select** input results the clocking source for the 8284A. If this pin is held high, an external clock is provided to the EFI input pin, and if it is held low, the internal crystal oscillator provides the timing signal.

Continued…

CSYNC The **clock synchronization** pin is used whenever the EFI input provides synchronization in systems with multiple processors. When the internal crystal oscillator is used, this pin must be grounded.

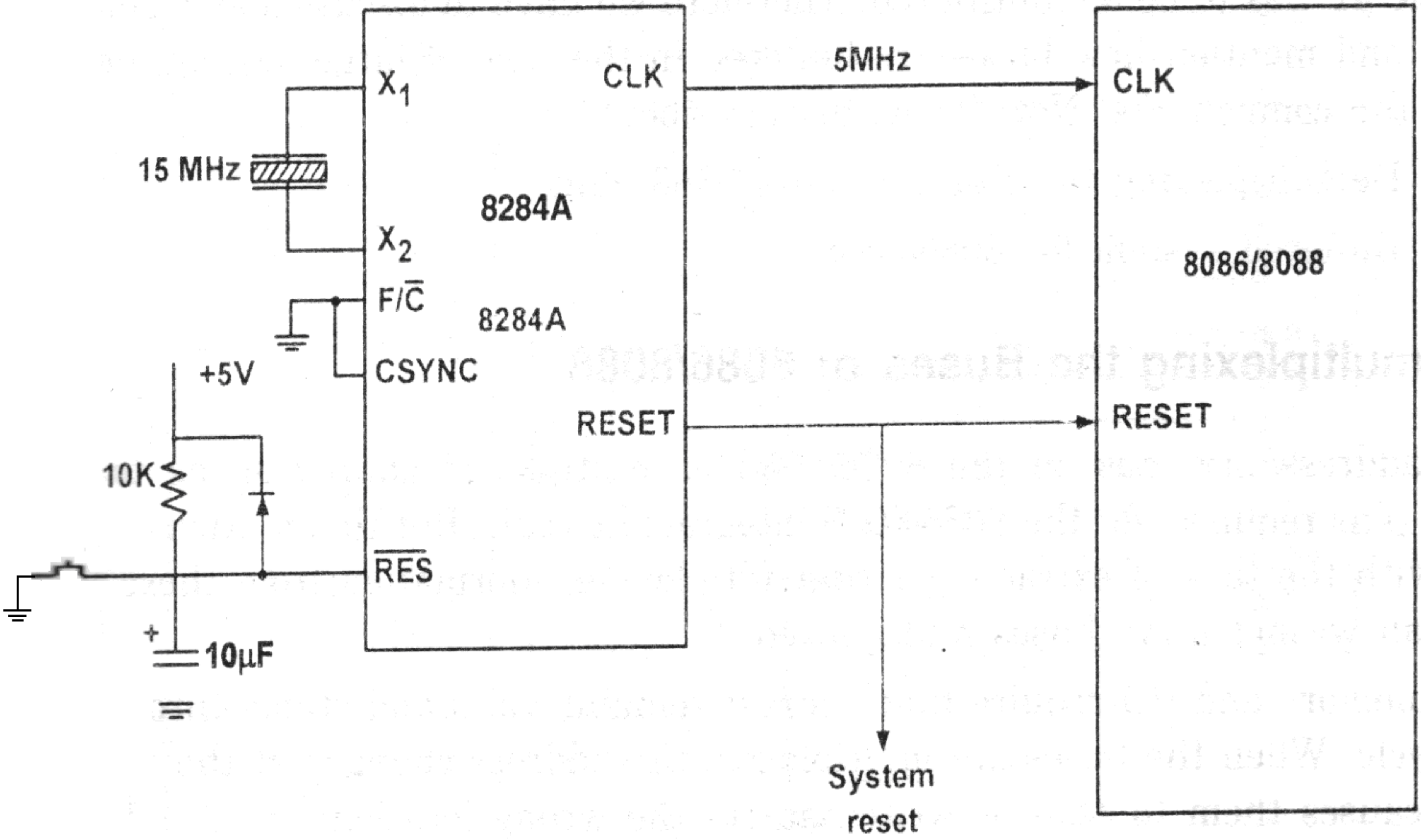
OSC The **Oscillator** output is a TTL level signal that is at the same frequency as the crystal or EFI input. (The OSC output provides and EFI input to other 8284A clock generators in some multiple processor systems).

Continued…

CLK The **clock** output pin provides CLK input signal to the 8086/8088 microprocessors (and other components in the system). The CLK pin has an output signal that is one-third of the crystal or EFI input frequency and has a 33 percent duty cycle, which is required by the 8086/8088.

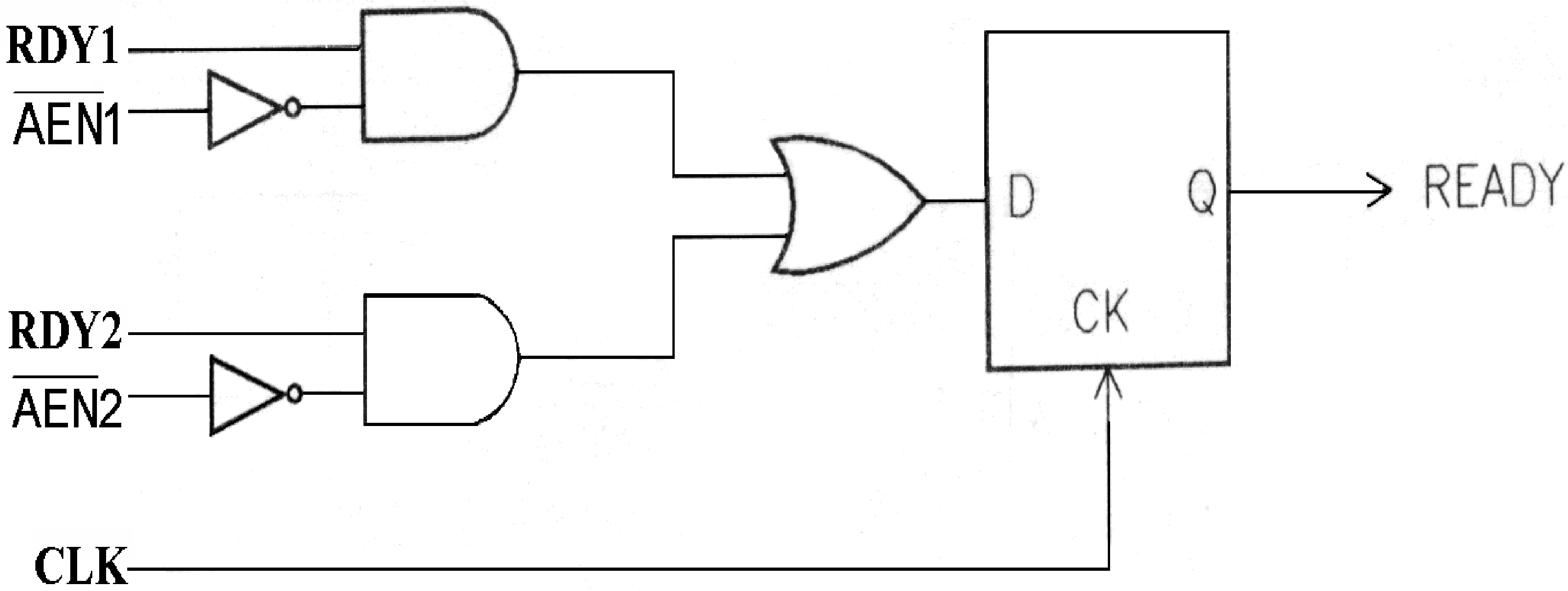
PCLK The **Peripheral Clock** signal is one-sixth the crystal or EFI input frequency and has a 50 percent duty cycle. The PCLK output provides a clock signal o the peripheral equipment in the system.

Clock Generator (8284A and the 8086/8088 microprocessor illustrating the connection for the clock and reset signals (A 15MHz crystal provides the 5MHz clock for the microprocessor)



Ready Logic

The Ready Logic generates the Ready signal for the 8086/8088. If the Ready signal is made low by this circuit during T2 state of a machine cycle, the microprocessor introduces a wait state between T3 and T4 states of the machine cycle.

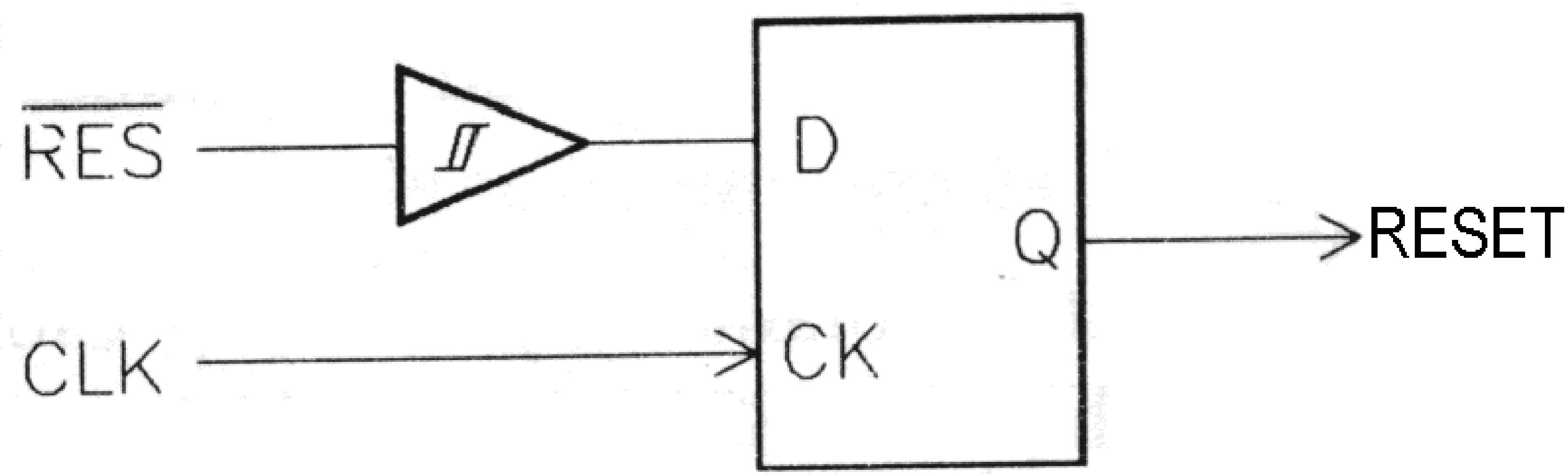


Continued…

The Ready logic is indicated in the figure. There are two pairs of signals in 8284 which can make the Ready output of 8284 to go low. If (RDY1=0 or SEN1*=1) and (RDY2=0 or AEN2*=1), the Ready output becomes low when the next clock transition takes place.

In PCs, RDY2 and AEN2* are not used, and as such RDY2 is tied to Ground and /or AEN2* is tied to +5V. AEN1* is used for generating wait states in the 8086/8088 bus cycle, and RDY1 is used for generating wait state in the DMA bus cycle.

Reset Logic



Continued…

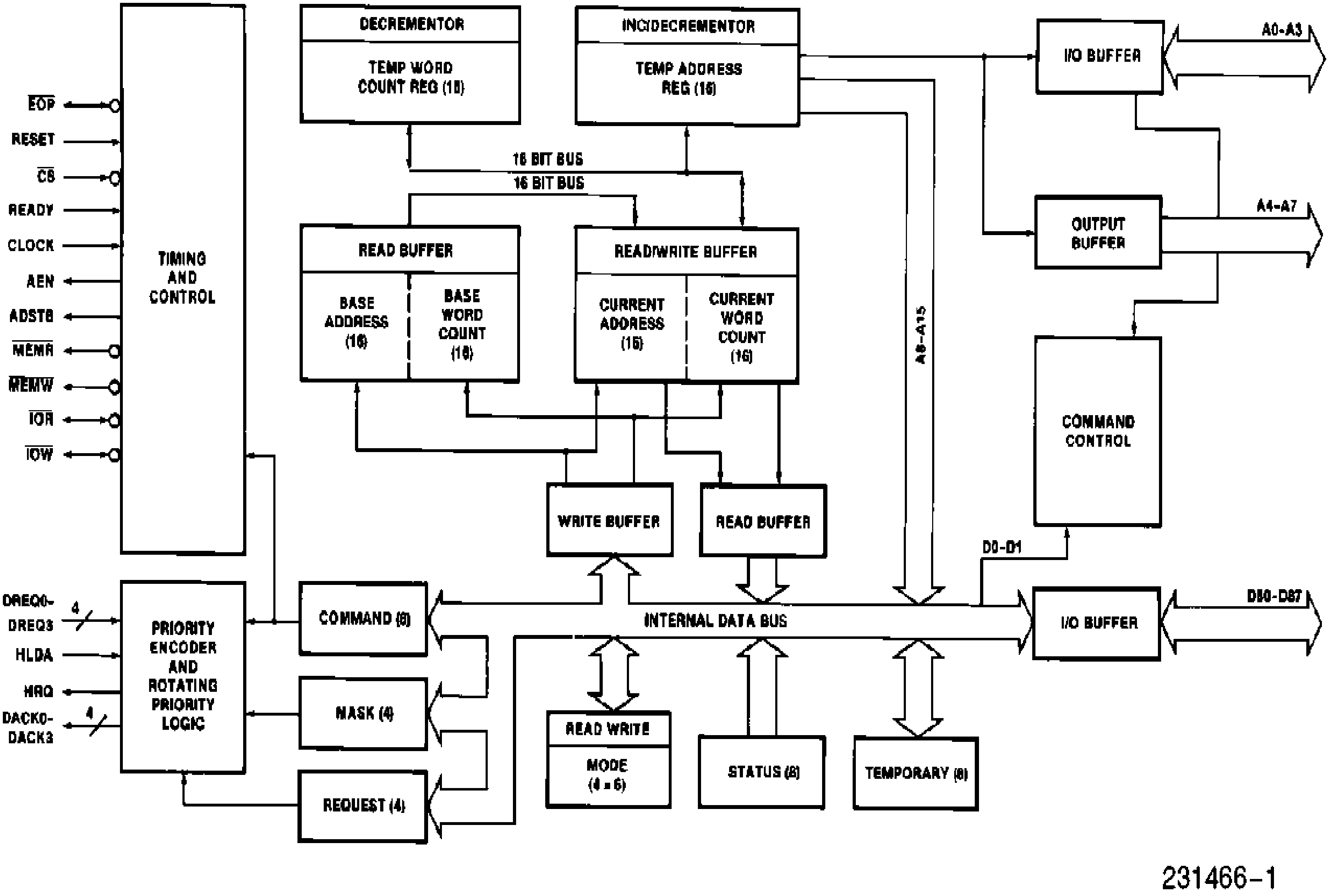
Reset Logic

The Reset logic generates the Reset input signal for the 8086/8088. When the RESET* pin goes low, the Reset output is generated by the 8284 when the next clock transition takes place.

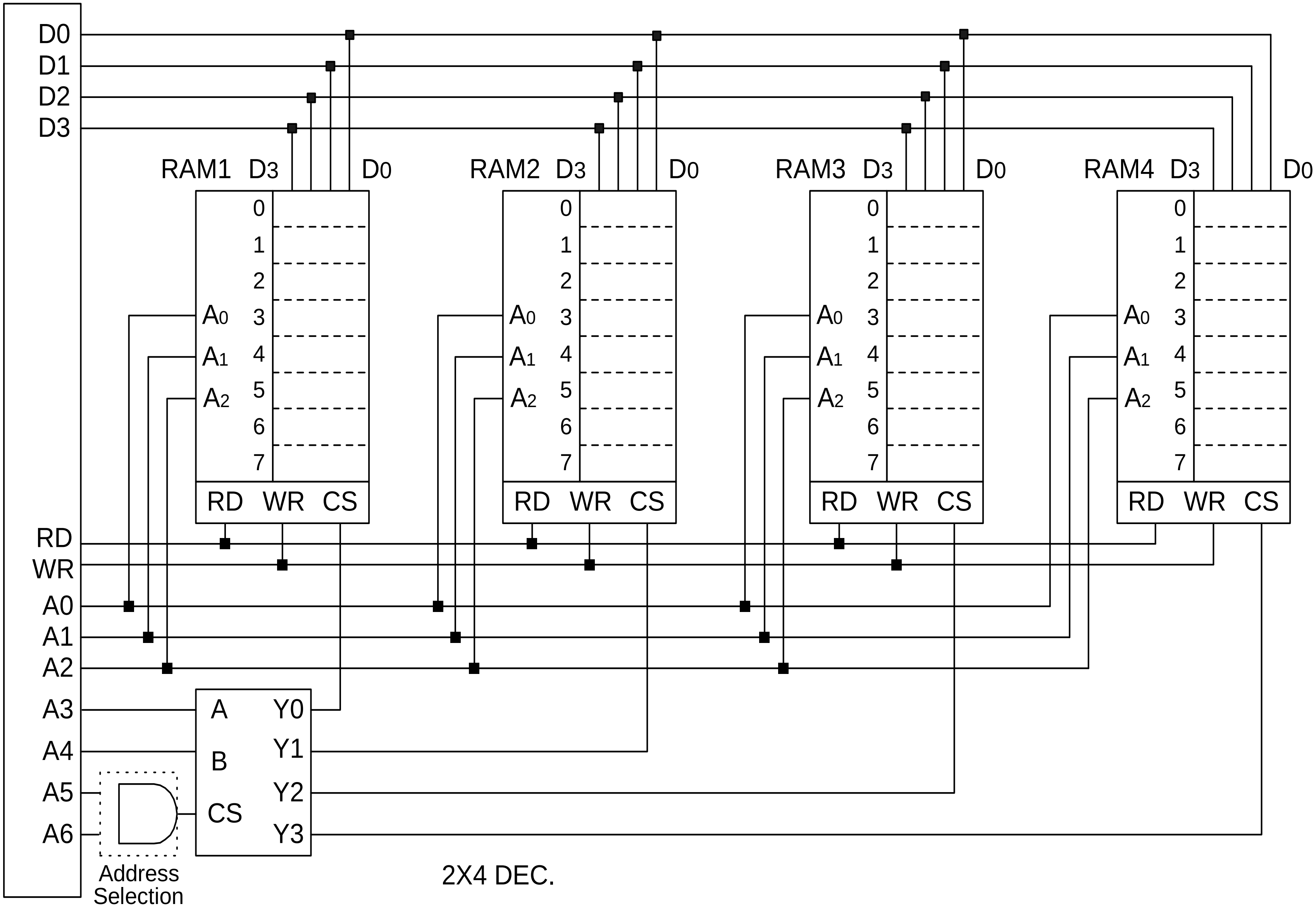
In PCs, the RES* input is activated by one of the following.

- ✓ From the manual Reset button on the front panel.
- ✓ From the ‘Power on Reset’ circuit, which uses RC components.
- ✓ If the ‘Power Good’ signal from the SMPS is not active.

8237 DMA Controller



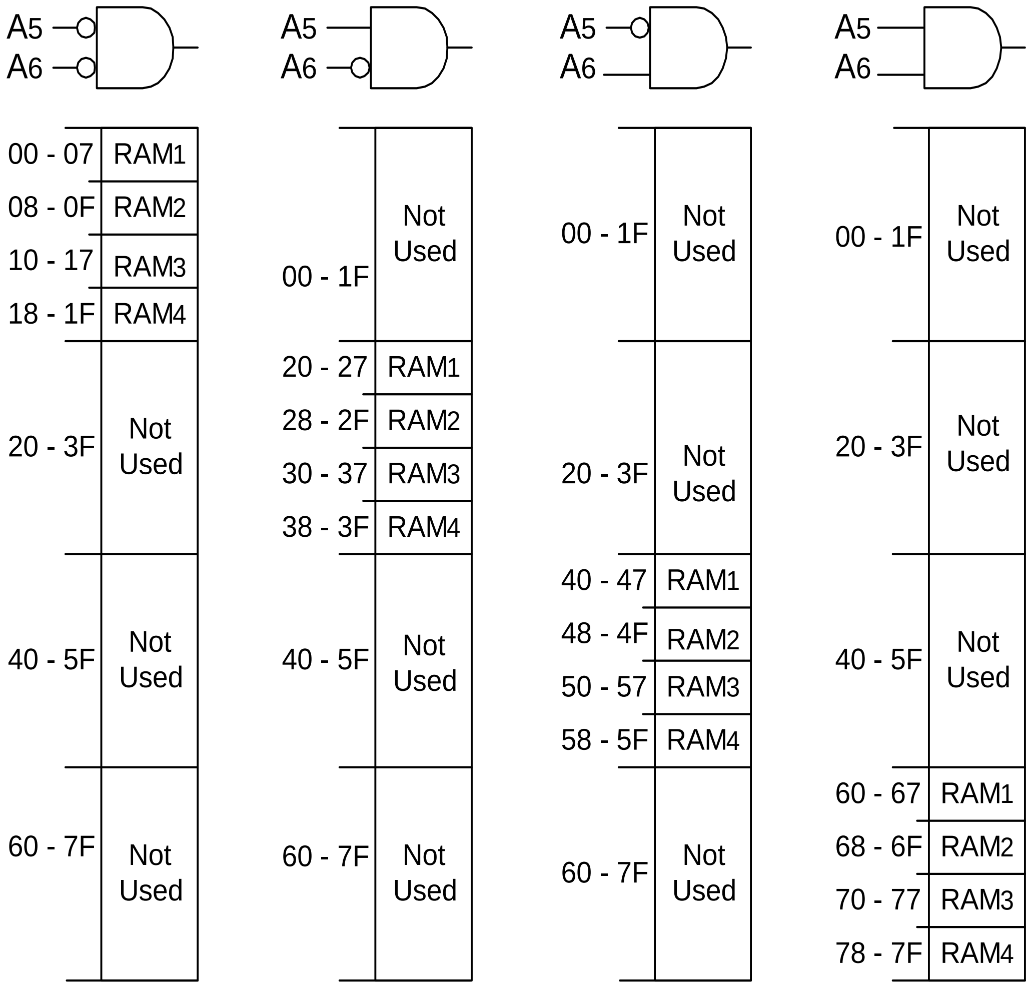
Address Size Expansion: (32X4 RAM module using 8X4 RAM chips)



Memory Maps

- Tables that show the addresses occupied by each memory device in a system.

- In the previous example it is assumed that the processor has only 7 address line, thus it can address 128 memory locations.
- The size of the RAM memory module is 32 bytes, thus the module can be mapped to occupy one out of the four available memory blocks in the memory map.
- The memory block occupied by the memory module depends on the connection of the address selection circuit (AND gate) that enables the decoder.



00 - 07	RAM1
08 - 0F	RAM2
10 - 17	RAM3
18 - 1F	RAM4
20 - 3F	Not Used
40 - 5F	Not Used
60 - 7F	Not Used

00 - 1F	Not Used
20 - 27	RAM1
28 - 2F	RAM2
30 - 37	RAM3
38 - 3F	RAM4
40 - 5F	Not Used
60 - 7F	Not Used

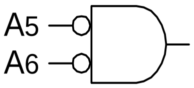
00 - 1F	Not Used
20 - 3F	Not Used
40 - 47	RAM1
48 - 4F	RAM2
50 - 57	RAM3
58 - 5F	RAM4
60 - 7F	Not Used

00 - 1F	Not Used
20 - 3F	Not Used
40 - 5F	Not Used
60 - 67	RAM1
68 - 6F	RAM2
70 - 77	RAM3
78 - 7F	RAM4

Effect of the Address Selection Circuit

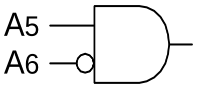
- The memory block occupied by the memory module depends on the connection of the address selection circuit (AND gate) that enables

Address Selection Circuit



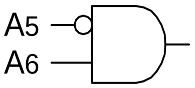
A6	A5	A4	A3	A2	A1	A0	Mem. Map	
0	0	0	0	0	0	0	00	RAM1
0	0	0	0	1	1	1	07	
0	0	0	1	0	0	0	08	RAM2
0	0	0	1	1	1	1	0F	
0	0	1	0	0	0	0	10	RAM3
0	0	1	0	1	1	1	17	
0	0	1	1	0	0	0	18	RAM4
0	0	1	1	1	1	1	1F	
0	1	0	0	0	0	0	20	Not Used
1	1	1	1	1	1	1	7F	

Address Selection Circuit



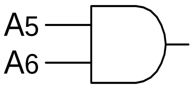
A6	A5	A4	A3	A2	A1	A0	Mem. Map	
0	0	0	0	0	0	0	00	Not Used
0	0	1	1	1	1	1	1F	
0	1	0	0	0	0	0	20	RAM1
0	1	0	0	1	1	1	27	
0	1	0	1	0	0	0	28	RAM2
0	1	0	1	1	1	1	2F	
0	1	1	0	0	0	0	30	RAM3
0	1	1	0	1	1	1	37	
0	1	1	1	0	0	0	38	RAM4
0	1	1	1	1	1	1	3F	
1	0	0	0	0	0	0	40	Not Used
1	1	1	1	1	1	1	7F	

Address Selection Circuit



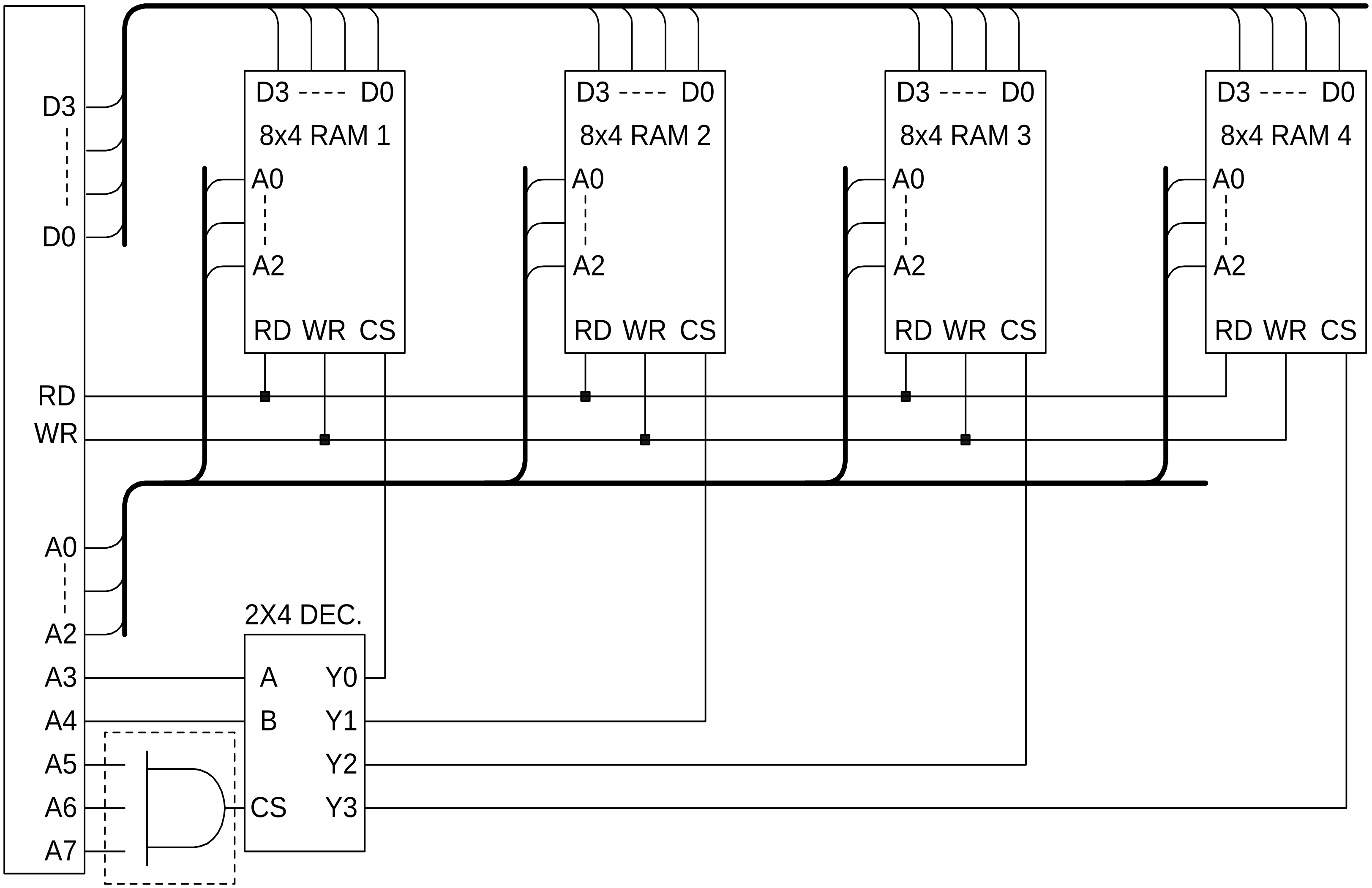
A6	A5	A4	A3	A2	A1	A0	Mem. Map	
0	0	0	0	0	0	0	00	Not Used
0	1	1	1	1	1	1	3F	
1	0	0	0	0	0	0	40	RAM1
1	0	0	0	1	1	1	47	
1	0	0	1	0	0	0	48	RAM2
1	0	0	1	1	1	1	4F	
1	0	1	0	0	0	0	50	RAM3
1	0	1	0	1	1	1	57	
1	0	1	1	0	0	0	58	RAM4
1	0	1	1	1	1	1	5F	
1	1	0	0	0	0	0	60	Not Used
1	1	1	1	1	1	1	7F	

Address Selection Circuit



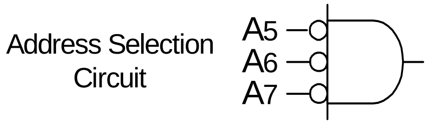
A6	A5	A4	A3	A2	A1	A0	Mem. Map	
0	0	0	0	0	0	0	00	Not Used
1	0	1	1	1	1	1	5F	
1	1	0	0	0	0	0	60	RAM1
1	1	0	0	1	1	1	67	
1	1	0	1	0	0	0	68	RAM2
1	1	0	1	1	1	1	6F	
1	1	1	0	0	0	0	70	RAM3
1	1	1	0	1	1	1	77	
1	1	1	1	0	0	0	78	RAM4
1	1	1	1	1	1	1	7F	

Example: (32X4 RAM module using 8X4 RAM chips - Assume an 8-address line processor)

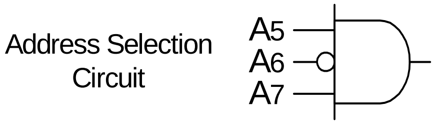


Memory Map for previous example.

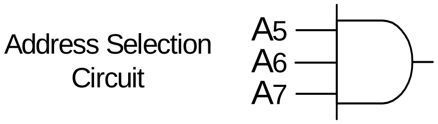
- There are three address lines connected on the address selection circuit. Thus there can be eight different memory map configurations



A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Mem. Map	
0	0	0	0	0	0	0	0	00	RAM1
0	0	0	0	0	1	1	1	07	
0	0	0	0	1	0	0	0	08	RAM2
0	0	0	0	1	1	1	1	0F	
0	0	0	1	0	0	0	0	10	RAM3
0	0	0	1	0	1	1	1	17	
0	0	0	1	1	0	0	0	18	RAM4
0	0	0	1	1	1	1	1	1F	
0	0	1	0	0	0	0	0	20	Not Used
1	1	1	1	1	1	1	1	FF	



A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Mem. Map	
0	0	0	0	0	0	0	0	00	Not Used
1	0	0	1	1	1	1	1	9F	
1	0	1	0	0	0	0	0	A0	RAM1
1	0	1	0	0	1	1	1	A7	
1	0	1	0	1	0	0	0	A8	RAM2
1	0	1	0	1	1	1	1	AF	
1	0	1	1	0	0	0	0	B0	RAM3
1	0	1	1	0	1	1	1	B7	
1	0	1	1	1	0	0	0	B8	RAM4
1	0	1	1	1	1	1	1	BF	
1	1	0	0	0	0	0	0	C0	Not Used
1	1	1	1	1	1	1	1	FF	



A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Mem. Map	
0	0	0	0	0	0	0	0	00	Not Used
1	1	0	1	1	1	1	1	DF	
1	1	1	0	0	0	0	0	E0	RAM1
1	1	1	0	0	1	1	1	E7	
1	1	1	0	1	0	0	0	E8	RAM2
1	1	1	0	1	1	1	1	EF	
1	1	1	1	0	0	0	0	F0	RAM3
1	1	1	1	0	1	1	1	F7	
1	1	1	1	1	0	0	0	F8	RAM4
1	1	1	1	1	1	1	1	FF	

Design Example:

Design an 8KX8 RAM module using 2KX8 RAM chips. The module should be connected on an 8-bit processor with a 16-bit address bus, and occupy the address range starting from the address A000. Show the circuit and the memory map.

- Number of memory devices needed = 8K/2K = 4
- Decoder needed = 2X4
- Number of address lines on each 2KX8 memory chip = 11

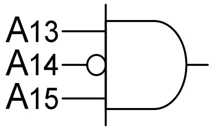
  $2^m = 2K = 2^1 \times 2^{10} = 2^{11} \Rightarrow$ (A0..A10)
- Decoder needed = 2X4

  $\Rightarrow$ 2 address lines are needed for the decoder. $\Rightarrow$ (A11..A12)
- Number of address lines needed for the address selection circuit

  = 16 - 11 - 2 = 3 $\Rightarrow$ (A13, A14 A15)

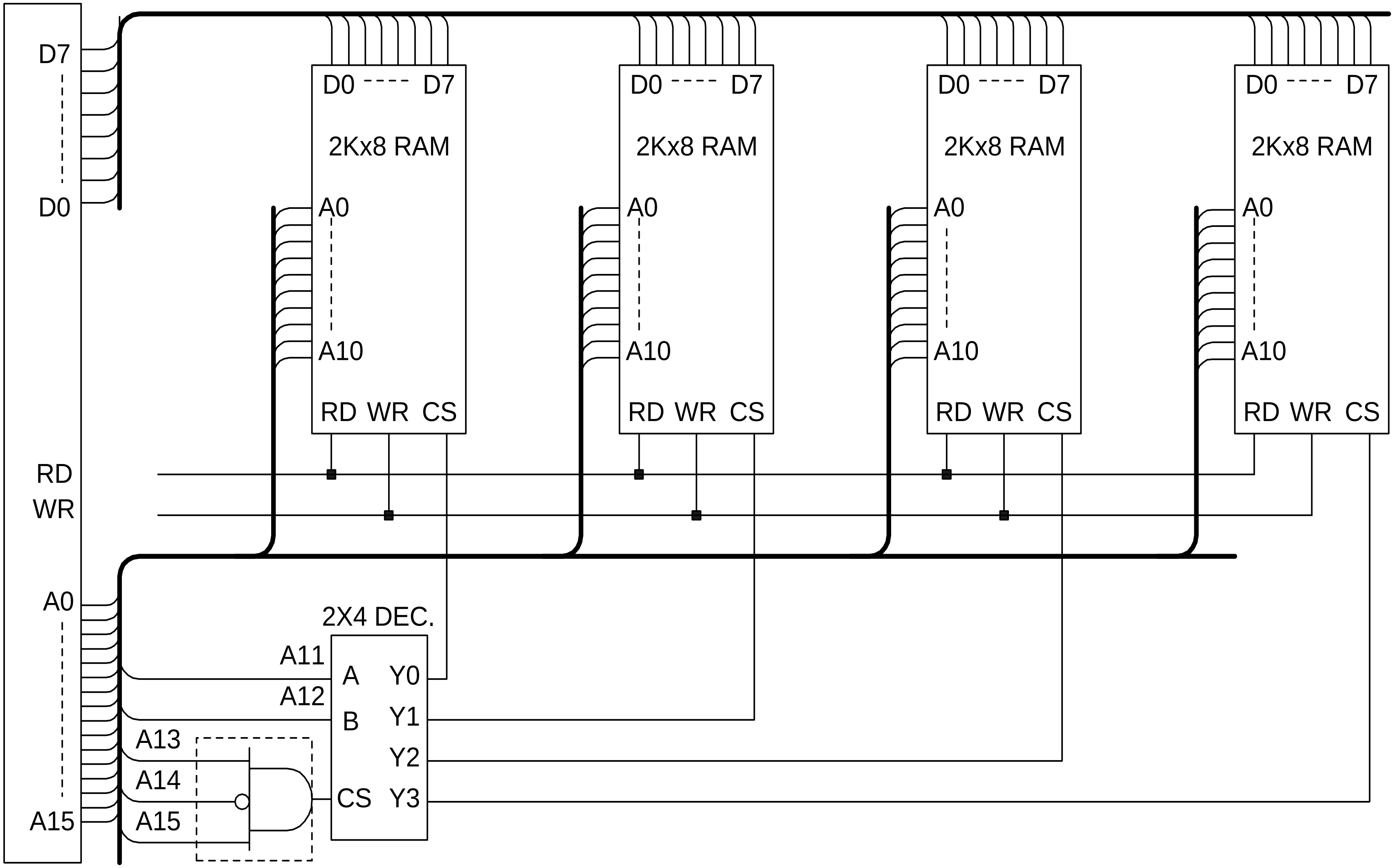
Starting Address = A000 = 1010-0000-0000-0000
==> A15 = 1, A14 = 0 and A13 = 1

Address Selection Circuit



A15	A14	A13	A12	A11	A10	......	A0	Mem. Map	
0	0	0	0	0	0	......	0	0000	Not Used
1	0	0	1	1	1	......	1	9FFF	
1	0	1	0	0	0	......	0	A000	RAM1
1	0	1	0	0	1	......	1	A7FF	
1	0	1	0	1	0	......	0	A800	RAM2
1	0	1	0	1	1	......	1	AFFF	
1	0	1	1	0	0	......	0	B000	RAM3
1	0	1	1	0	1	......	1	B7FF	
1	0	1	1	1	0	......	0	B800	RAM4
1	0	1	1	1	1	......	1	BFFF	
1	1	0	0	0	0	......	0	C000	Not Used
1	1	1	1	1	1	......	1	FFFF	

Circuit Diagram



Address Decoding

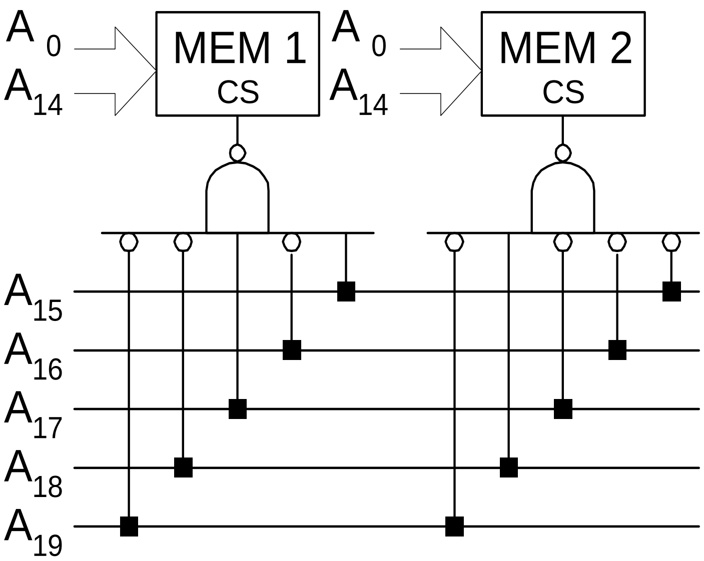
- The physical address space, or memory map, of a microprocessor refers to the range of addresses of memory location that can accessed by the microprocessor. The size of the address space depends on the number of address lines of the microprocessor.
- At least two memory devices are required in a microprocessor system: one for the ROM and one for the RAM.
- In an 8088/8086 the high addresses in the memory map should always be occupied by a ROM, while the low addresses in the memory map should always be occupied by a RAM.
- Address decoding is required in order to enable the connection of more than one memory devices on the microprocessor. Each device will occupy a unique area in the memory map.
- A memory system is not fully decoded if some of the address lines are not used by the address decoding circuit or memory. In this case a memory device will occupy more than one sections in the memory map. This is referred as memory mirroring or memory imaging.

Address Decoding Circuits

- A number of types of address decoding circuits can be used in a microprocessor system.The main issues related to the selection of an address decoding circuit are:
  - The time delays introduced by the address decoding circuit. This delays are added to the access time of the memory devices, and might yield to the insertion of wait states.
  - The number of chips required by the address decoding circuit, as well as the complexity of the circuit (number of tracks required on the board.
- An address decoding circuit must ensure that an address section is occupied by only one memory device. If two or more devices occupy the same addresses then bus contention will occur. Bus contention occurs if two of more devices drive the bus at the same time. Bus contention can be either static or dynamic.
  - Static bus contention occurs when two or more devices drive a bus for a prolonged time period. This might damage some of the components of the system. Static bus contention might be caused by improper address decoding design, or by other faults in the system such as a short circuit of the CS of a device to the ground.
  - Dynamic bus contention occurs when two or more devices drive a bus for a short period of time. This might change the logic levels on the bus and cause system malfunctions.Dynamic bus contention might be caused by improper address decoding design, or by wrong memory timing analysis.

Address decoding circuits using Only NAND gates

- A single NAND gate is used to decode each memory device. The inputs of the NAND gate can be connected on the address lines either directly, or through inverters, according to the required memory map.
- This decoding circuit has the advantage that it adds a short time delay in the memory path. (t_d = 2 X gate delay <10ns)
- The disadvantage of this circuit is that too many gates (NAND and NOT) are needed for memory systems that have a few memory chips. This increases the cost of the system, adds to the complexity of the PCB board (too many chips and lines) and might create fan-out problems.



A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	• • •	A_0	Memory Map	
0	0	1	0	1	0	• • •	0	28000H	MEM1
0	0	1	0	1	1	• • •	1	2FFFFH	
0	1	0	0	0	0	• • •	0	40000H	MEM2
0	1	0	0	0	1	• • •	1	47FFFH	

Direct memory access

Direct memory access (DMA) is a process in which an external device takes over the control of system bus from the CPU.

DMA is for **high-speed data transfer** from/to mass storage peripherals, e.g. harddisk drive, magnetic tape, CD-ROM, and sometimes video controllers.

For example, a hard disk may boasts a transfer rate of 5 M bytes per second, i.e. 1 byte transmission every 200 ns. To make such data transfer via the CPU is both undesirable and unnecessary.

The basic idea of **DMA** is to transfer *blocks of data* directly between memory and peripherals. The data don’t go through the microprocessor but the data bus is occupied.

“Normal” transfer of one data byte takes up to 29 clock cycles. The DMA transfer requires only 5 clock cycles.

Nowadays, DMA can transfer data as fast as 60 M byte per second. The transfer rate is limited by the speed of memory and peripheral devices.

Basic process of DMA

For 8086 in maximum mode:

The RQ/GT1 and RQ/GT0 pins are used to issue DMA request and receive acknowledge signals.

Sequence of events of a typical DMA process

1) Peripheral asserts one of the request pins, e.g. RQ/GT1 or RQ/GT0 (RQ/GT0 has higher priority)
2) 8088 completes its current bus cycle and enters into a HOLD state
3) 8088 grants the right of bus control by asserting a grant signal via the same pin as the request signal.
4) DMA operation starts
5) Upon completion of the DMA operation, the peripheral asserts the request/grant pin again to relinquish bus control.

For 8086 in minimum mode:

The HOLD and HLDA pins are used instead to receive and acknowledge the hold request respectively.

Normally the CPU has full control of the system bus. In a DMA operation, the peripheral takes over bus control temporarily.

DMA controller

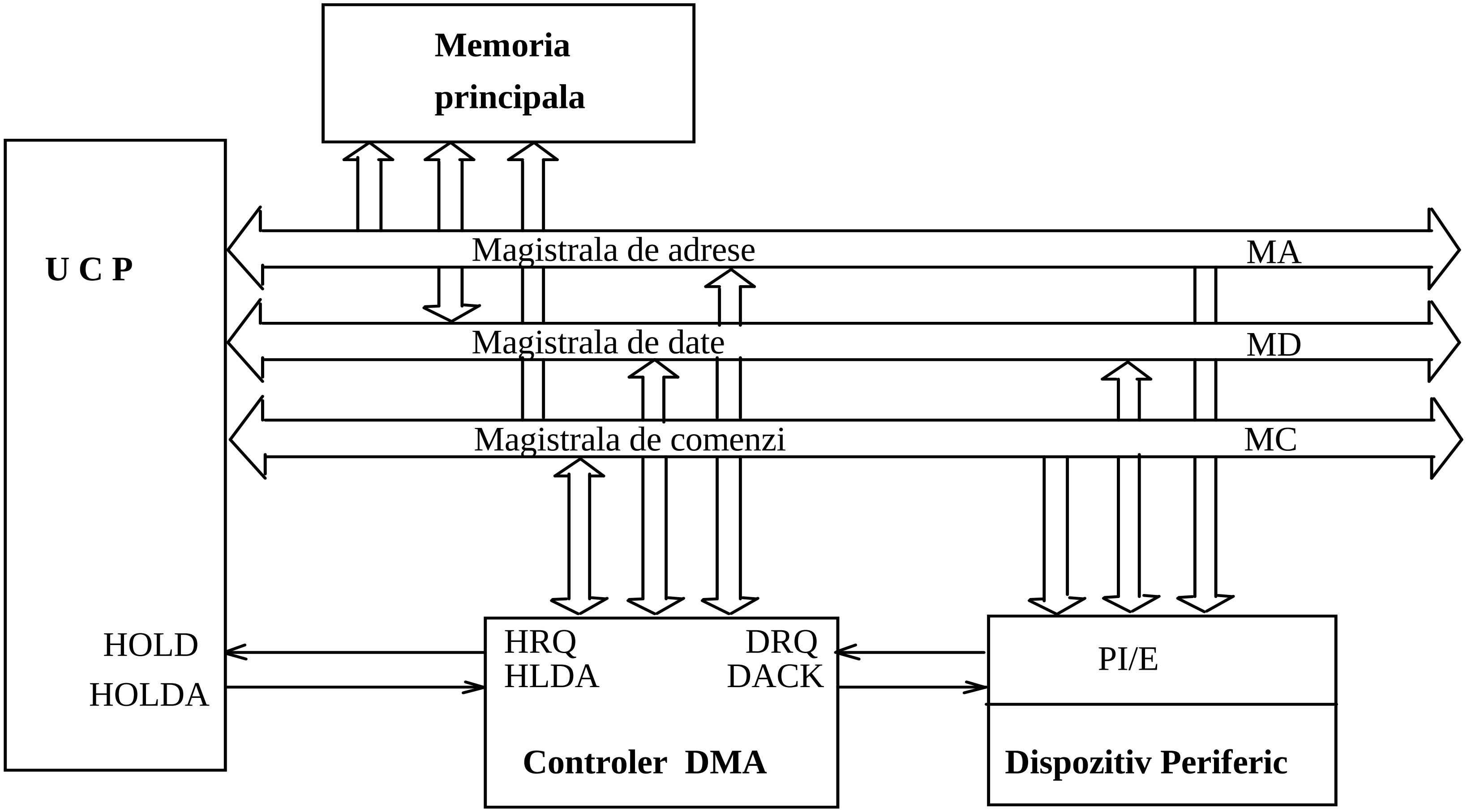
A DMA controller interfaces with several peripherals that may request DMA.

The controller decides the priority of simultaneous DMA requests communicates with the peripheral and the CPU, and provides memory addresses for data transfer.

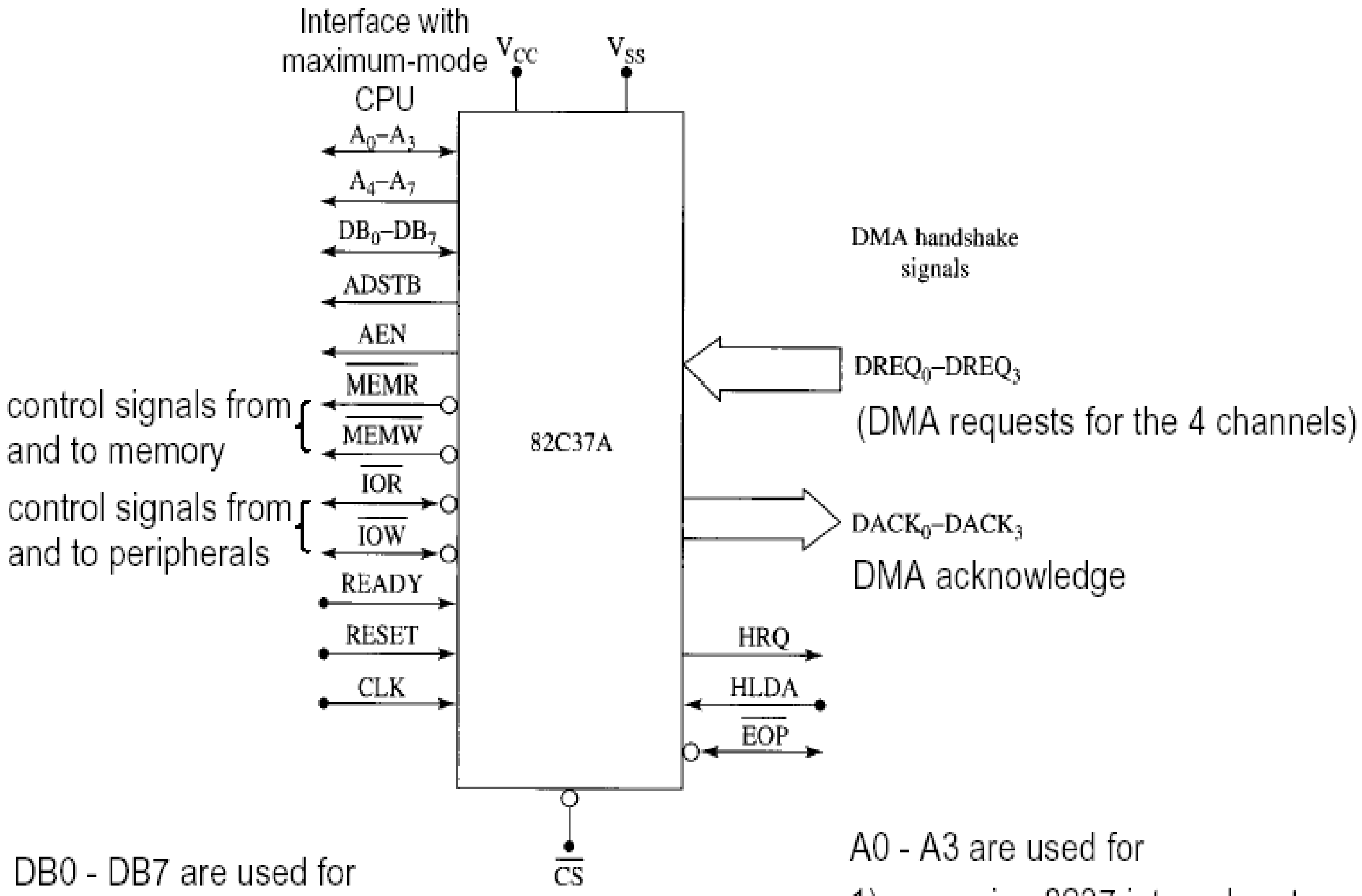
DMA controller commonly used with 8088 is the 8237 programmable device.

The 8237 is in fact a special-purpose microprocessor.
Normally it appears as part of the system controller chip-sets.

The 8237 is a 4-channel device. Each channel is dedicated to a specific peripheral device and capable of addressing 64 K bytes section of memory.


Memoria
principala
U C P
Magistrala de adrese
MA
Magistrala de date
MD
Magistrala de comenzi
MC
HOLD
HOLDA
HRQ
HLDA
DRQ
DACK
Controler DMA
PI/E
Dispozitiv Periferic

8237 DMA controller



Interface with maximum-mode CPU

control signals from and to memory

control signals from and to peripherals

(DMA requests for the 4 channels)

DMA acknowledge

DB0 - DB7 are used for

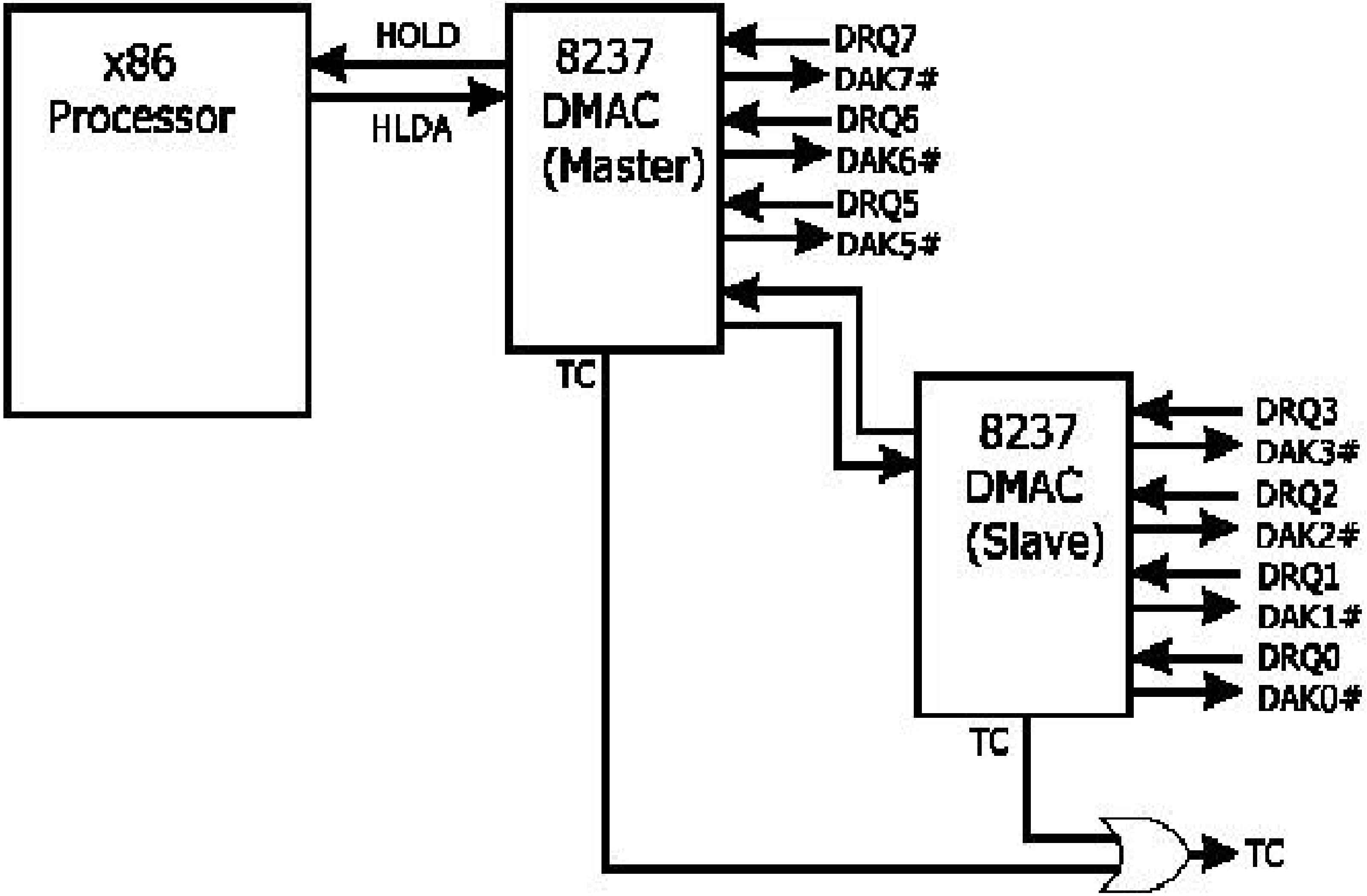
1) transfer of data

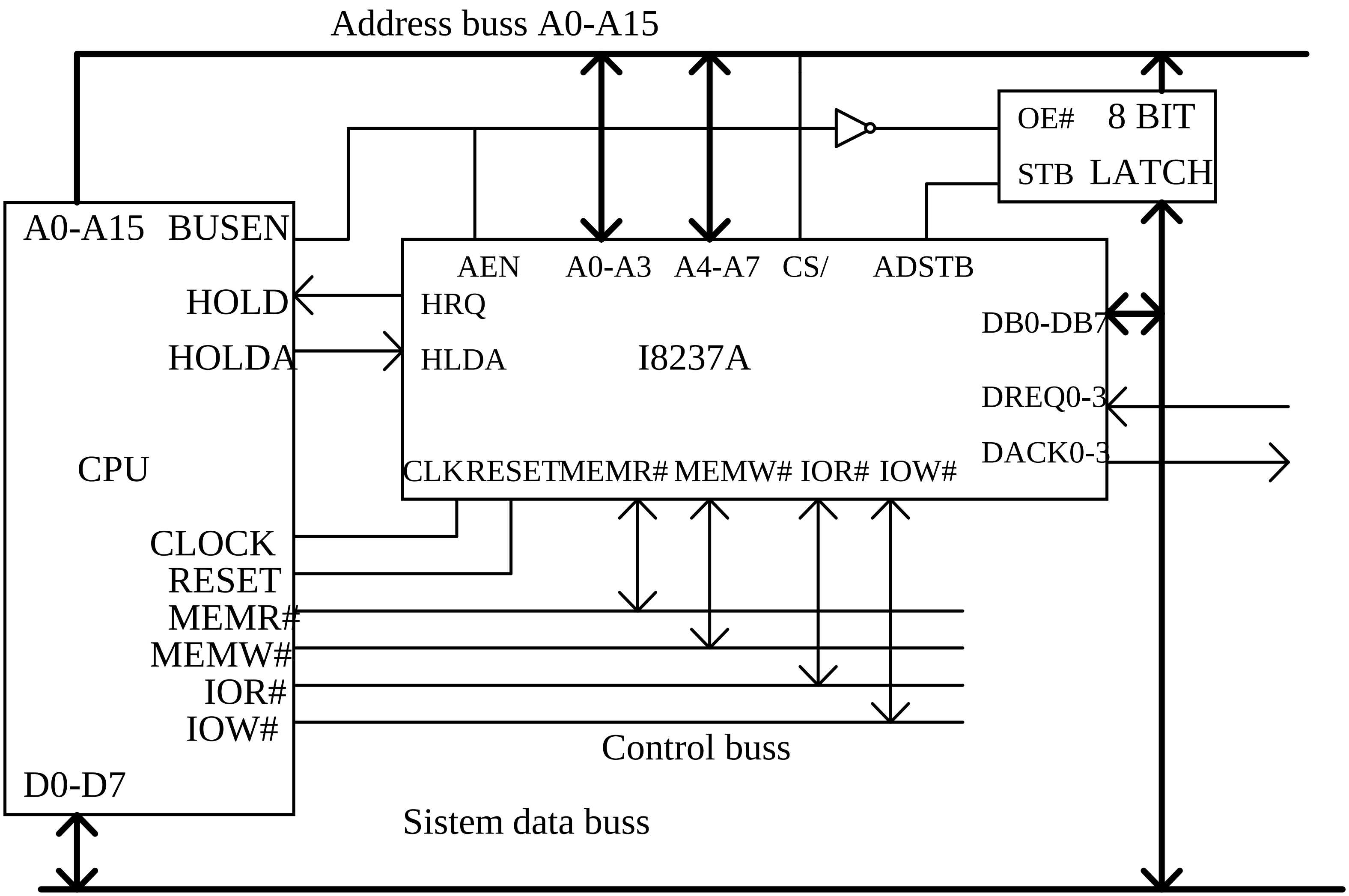
2) 8237 programming

A0 - A3 are used for

1) accessing 8237 internal ports

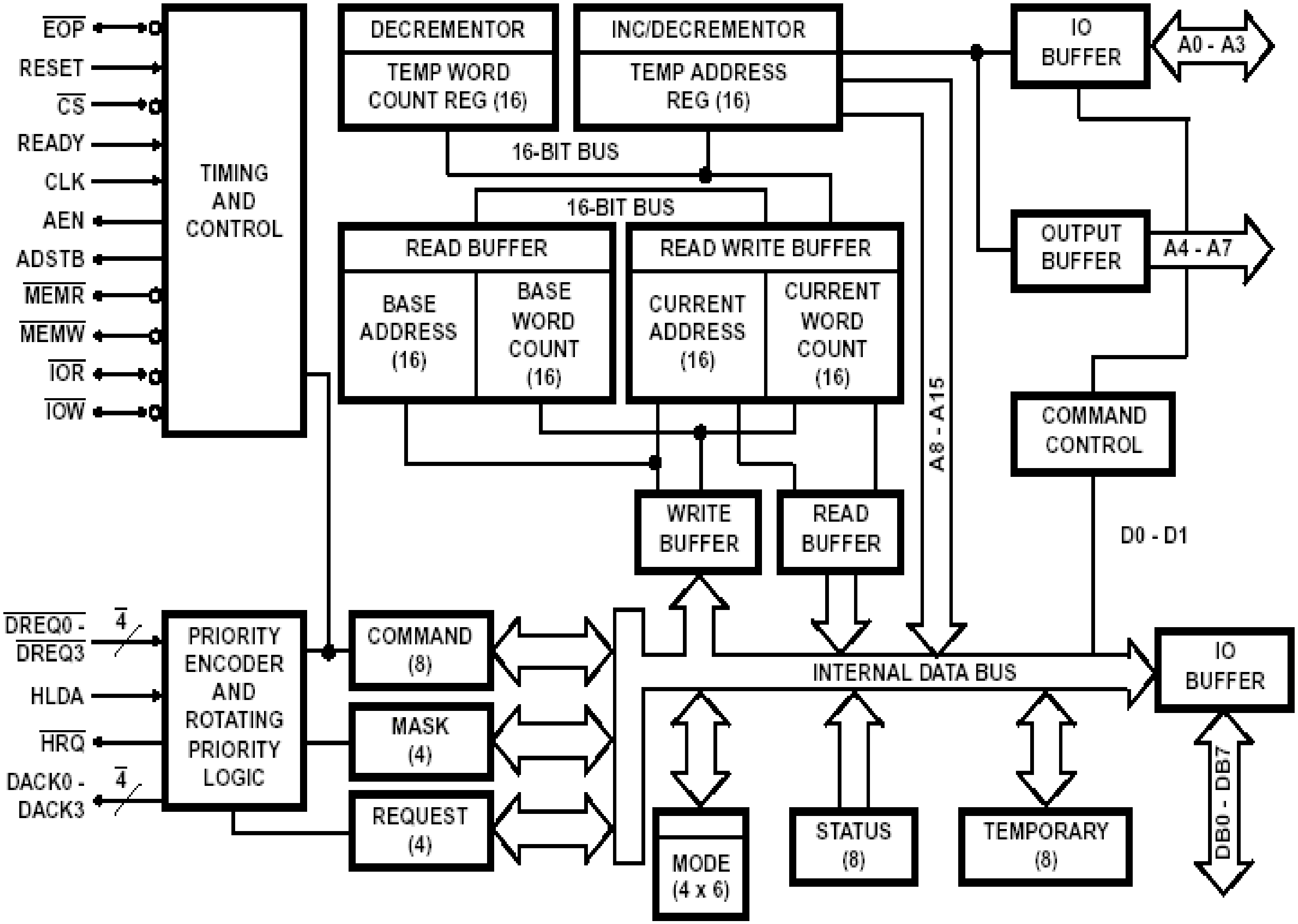
2) carrying memory address in DMA read and write operations


x86
Processor
HOLD
HLDA
8237
DMAC
(Master)
DRQ7
DAK7#
DRQ6
DAK6#
DRQ5
DAK5#
TC
8237
DMAC
(Slave)
DRQ3
DAK3#
DRQ2
DAK2#
DRQ1
DAK1#
DRQ0
DAK0#
TC
TC

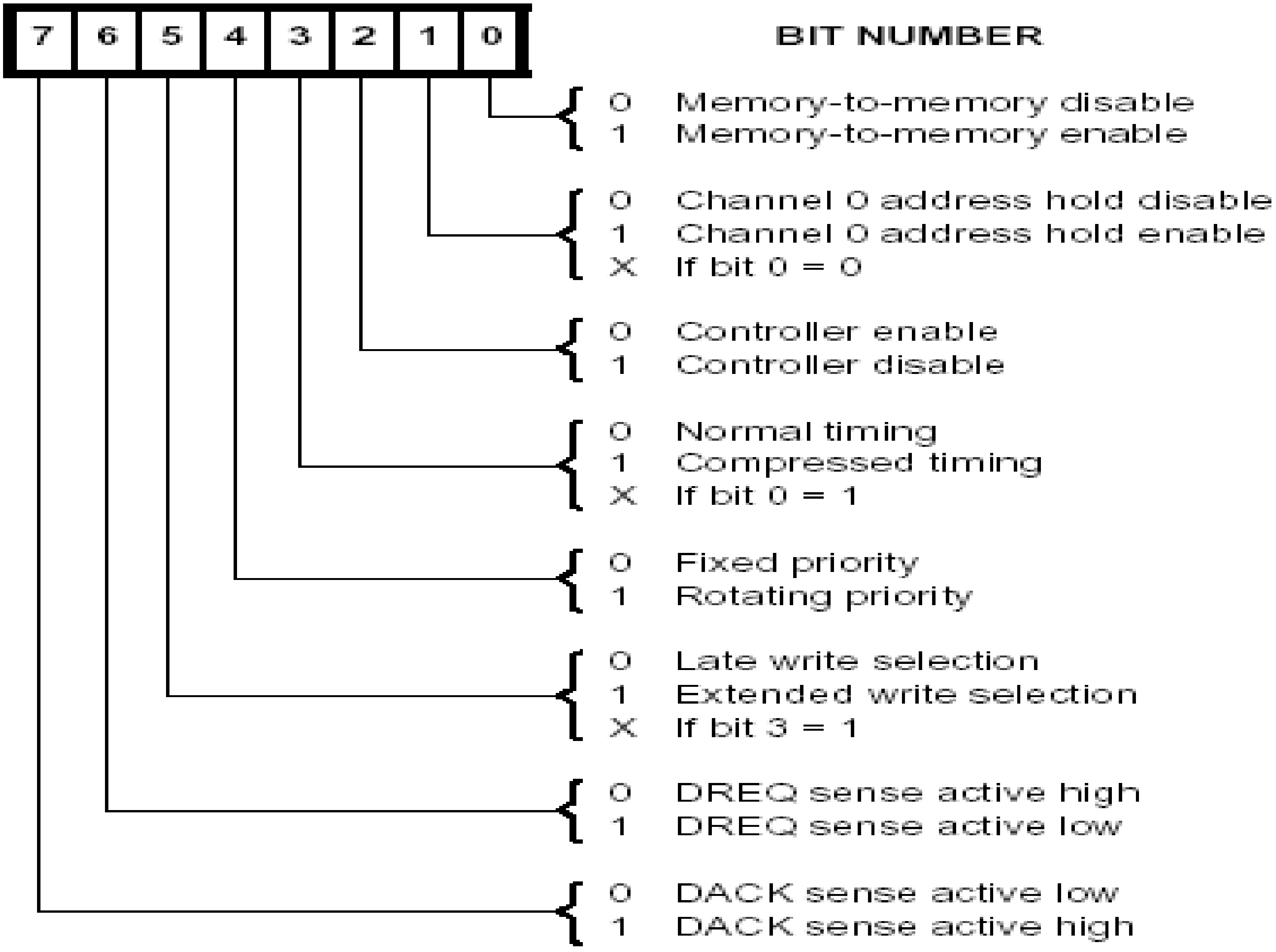


Address buss A0-A15
OE#
8 BIT
STB
LATCH
A0-A15
BUSEN
AEN
A0-A3
A4-A7
CS/
ADSTB
HOLD
HRQ
DB0-DB7
HOLDA
HLDA
I8237A
DREQ0-3
DACK0-3
CPU
CLK
RESET
MEMR#
MEMW#
IOR#
IOW#
CLOCK
RESET
MEMR#
MEMW#
IOR#
IOW#
Control buss
D0-D7
Sistem data buss

Block Diagram



Command Register



0 Memory-to-memory disable
1 Memory-to-memory enable

0 Channel 0 address hold disable
1 Channel 0 address hold enable
X If bit 0 = 0

0 Controller enable
1 Controller disable

0 Normal timing
1 Compressed timing
X If bit 0 = 1

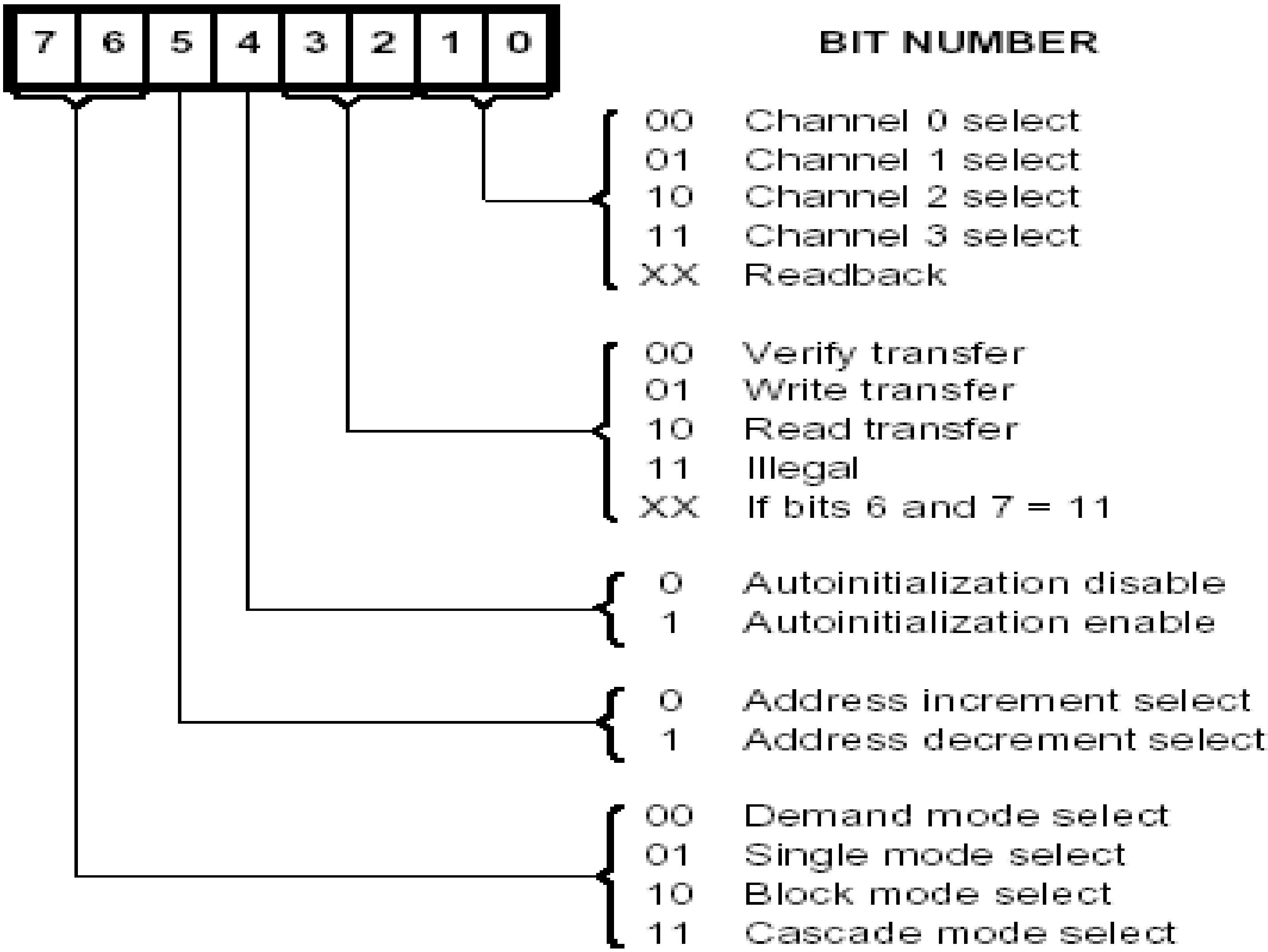
0 Fixed priority
1 Rotating priority

0 Late write selection
1 Extended write selection
X If bit 3 = 1

0 DREQ sense active high
1 DREQ sense active low

0 DACK sense active low
1 DACK sense active high

Mode Register



00 Channel 0 select
01 Channel 1 select
10 Channel 2 select
11 Channel 3 select
XX Readback

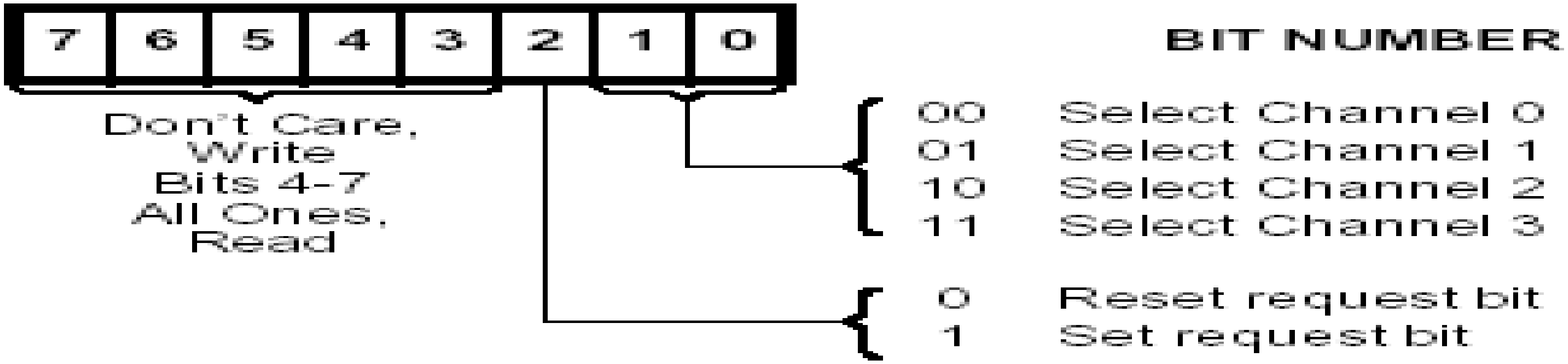
00 Verify transfer
01 Write transfer
10 Read transfer
11 Illegal
XX If bits 6 and 7 = 11

0 Autoinitialization disable
1 Autoinitialization enable

0 Address increment select
1 Address decrement select

00 Demand mode select
01 Single mode select
10 Block mode select
11 Cascade mode select

Request Register



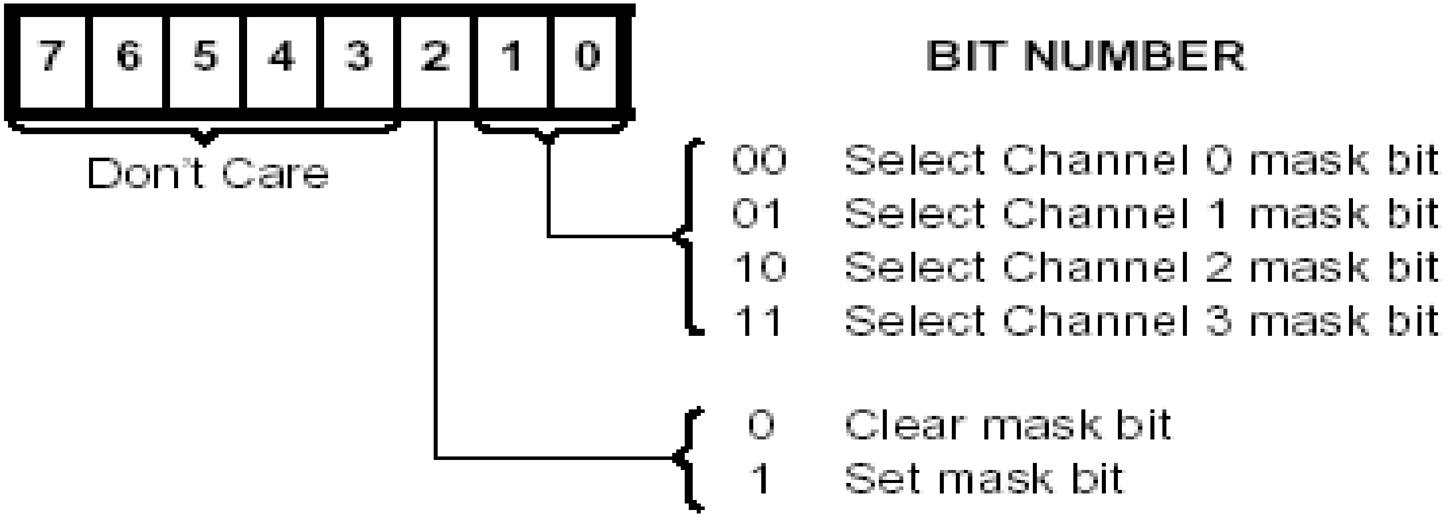
BIT NUMBER

Don't Care,
Write
Bits 4-7
All Ones,
Read

00 Select Channel 0
01 Select Channel 1
10 Select Channel 2
11 Select Channel 3

0 Reset request bit
1 Set request bit

Mask Register

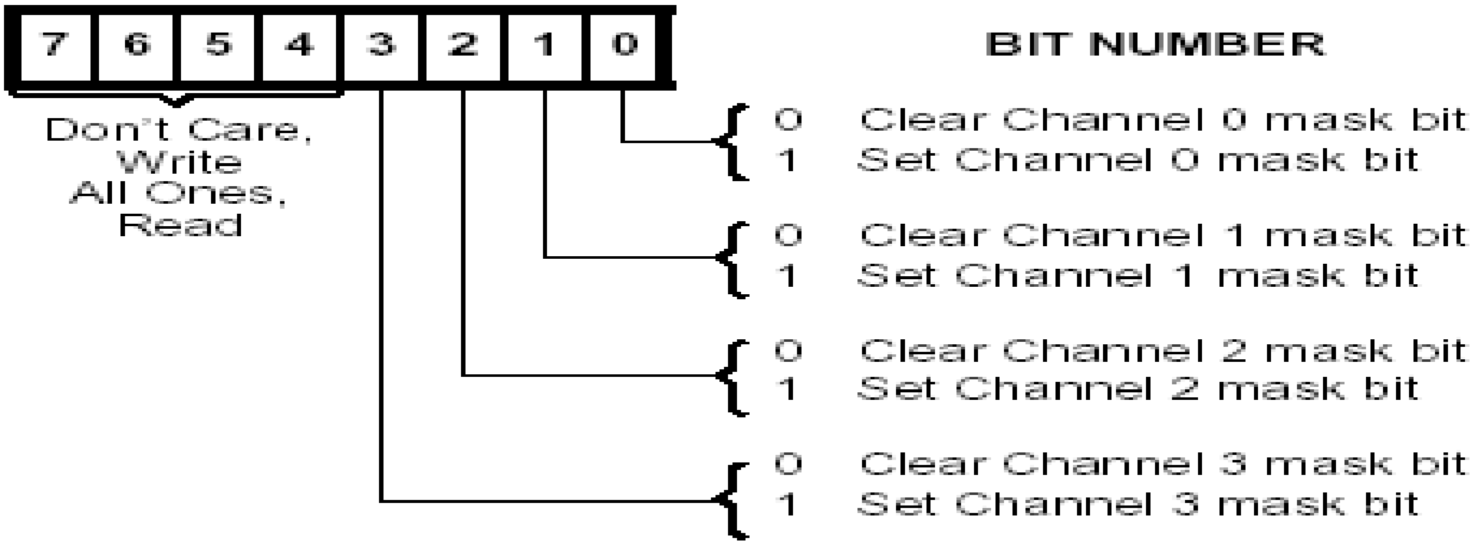


BIT NUMBER

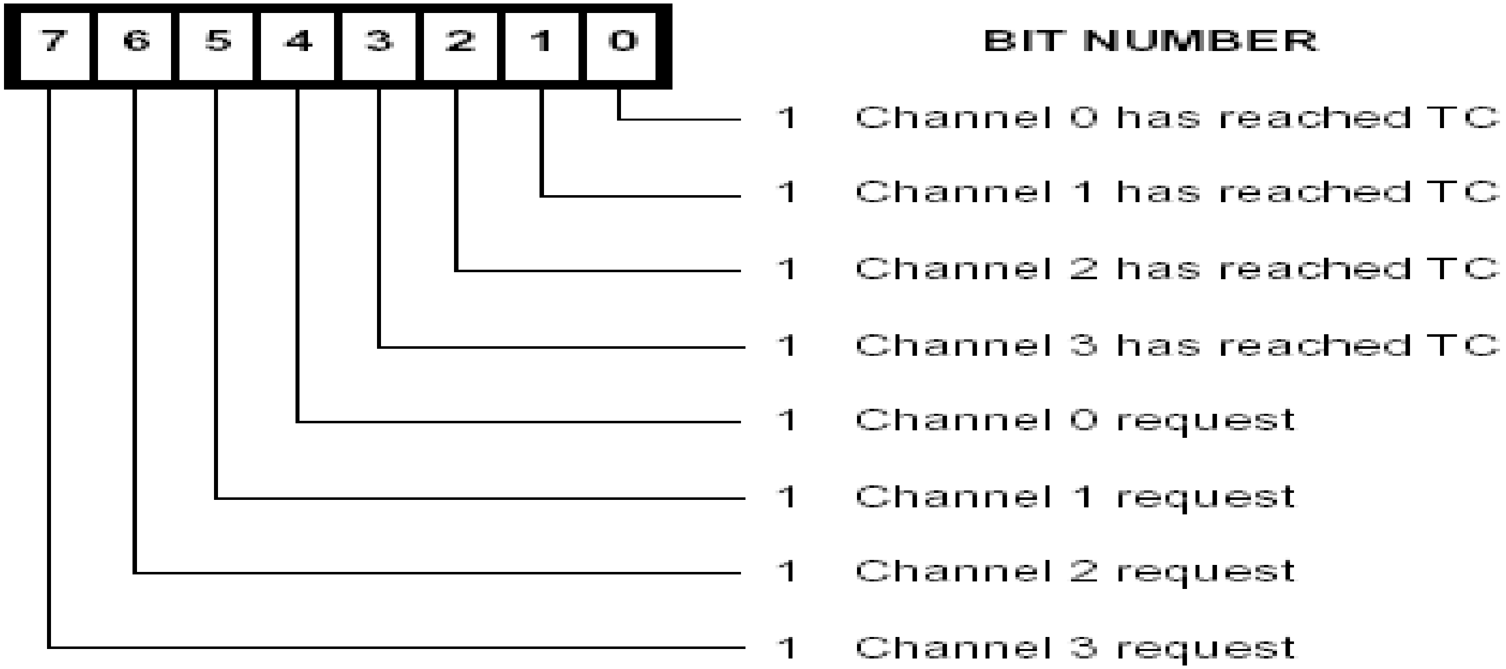
Don't Care

00 Select Channel 0 mask bit
01 Select Channel 1 mask bit
10 Select Channel 2 mask bit
11 Select Channel 3 mask bit

0 Clear mask bit
1 Set mask bit


7 6 5 4 3 2 1 0
BIT NUMBER
Don't Care, Write All Ones, Read
0 Clear Channel 0 mask bit
1 Set Channel 0 mask bit
0 Clear Channel 1 mask bit
1 Set Channel 1 mask bit
0 Clear Channel 2 mask bit
1 Set Channel 2 mask bit
0 Clear Channel 3 mask bit
1 Set Channel 3 mask bit

Status Register


7 6 5 4 3 2 1 0
BIT NUMBER
1 Channel 0 has reached TC
1 Channel 1 has reached TC
1 Channel 2 has reached TC
1 Channel 3 has reached TC
1 Channel 0 request
1 Channel 1 request
1 Channel 2 request
1 Channel 3 request

OPERATION	A3	A2	A1	A0	$\overline{IOR}$	$\overline{IOW}$
Read Status Register	1	0	0	0	0	1
Write Command Register	1	0	0	0	1	0
Read Request Register	1	0	0	1	0	1
Write Request Register	1	0	0	1	1	0
Read Command Register	1	0	1	0	0	1
Write Single Mask Bit	1	0	1	0	1	0
Read Mode Register	1	0	1	1	0	1
Write Mode Register	1	0	1	1	1	0
Set First/Last F/F	1	1	0	0	0	1
Clear First/Last F/F	1	1	0	0	1	0
Read Temporary Register	1	1	0	1	0	1
Master Clear	1	1	0	1	1	0
Clear Mode Reg. Counter	1	1	1	0	0	1
Clear Mask Register	1	1	1	0	1	0
Read All Mask Bits	1	1	1	1	0	1
Write All Mask Bits	1	1	1	1	1	0