

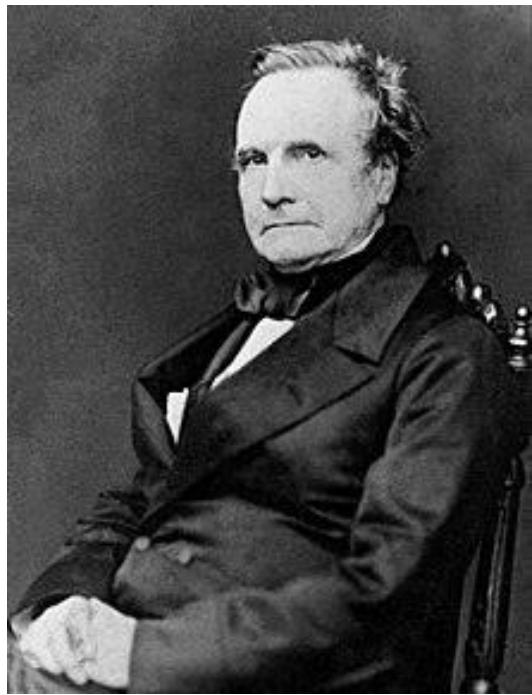
Microprocessors and Microcontrollers

By
Dr. VENKATA REDDY KOTA
Associate Professor of EEE

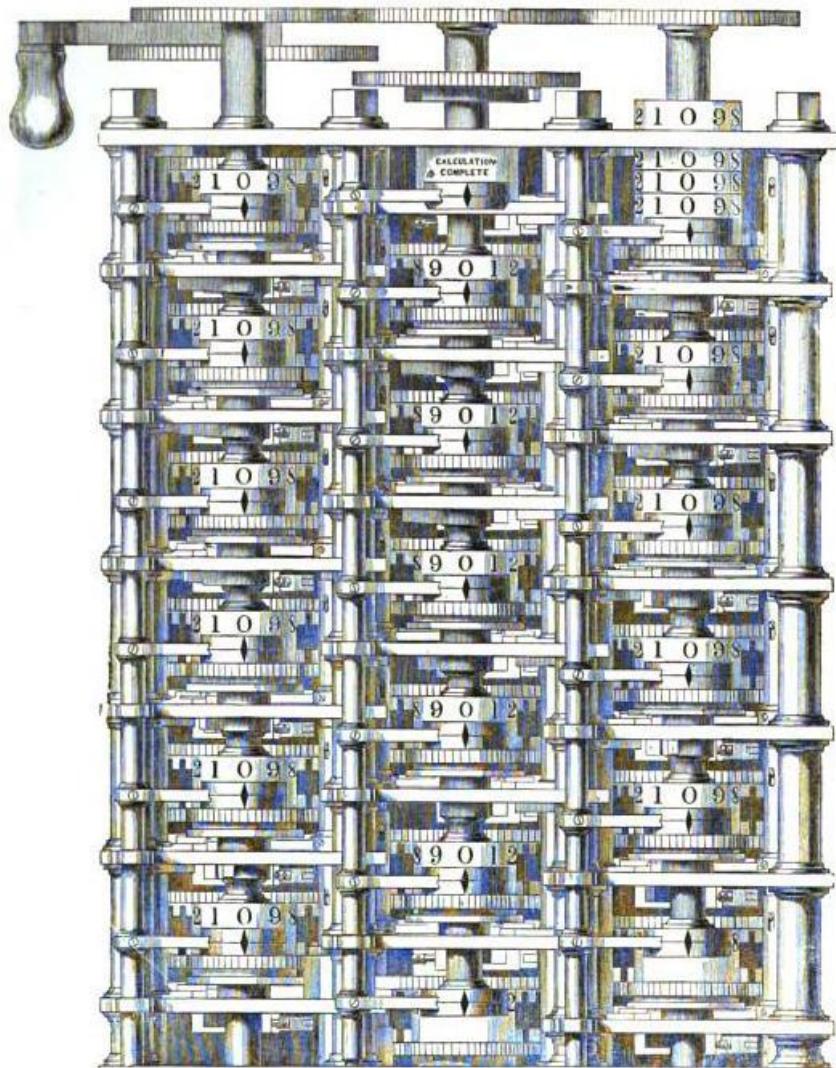


Department of Electrical & Electronics Engineering
University College of Engineering Kakinada (Autonomous)
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
Kakinada – 533003, Andhra Pradesh, India, www.jntuk.edu.in

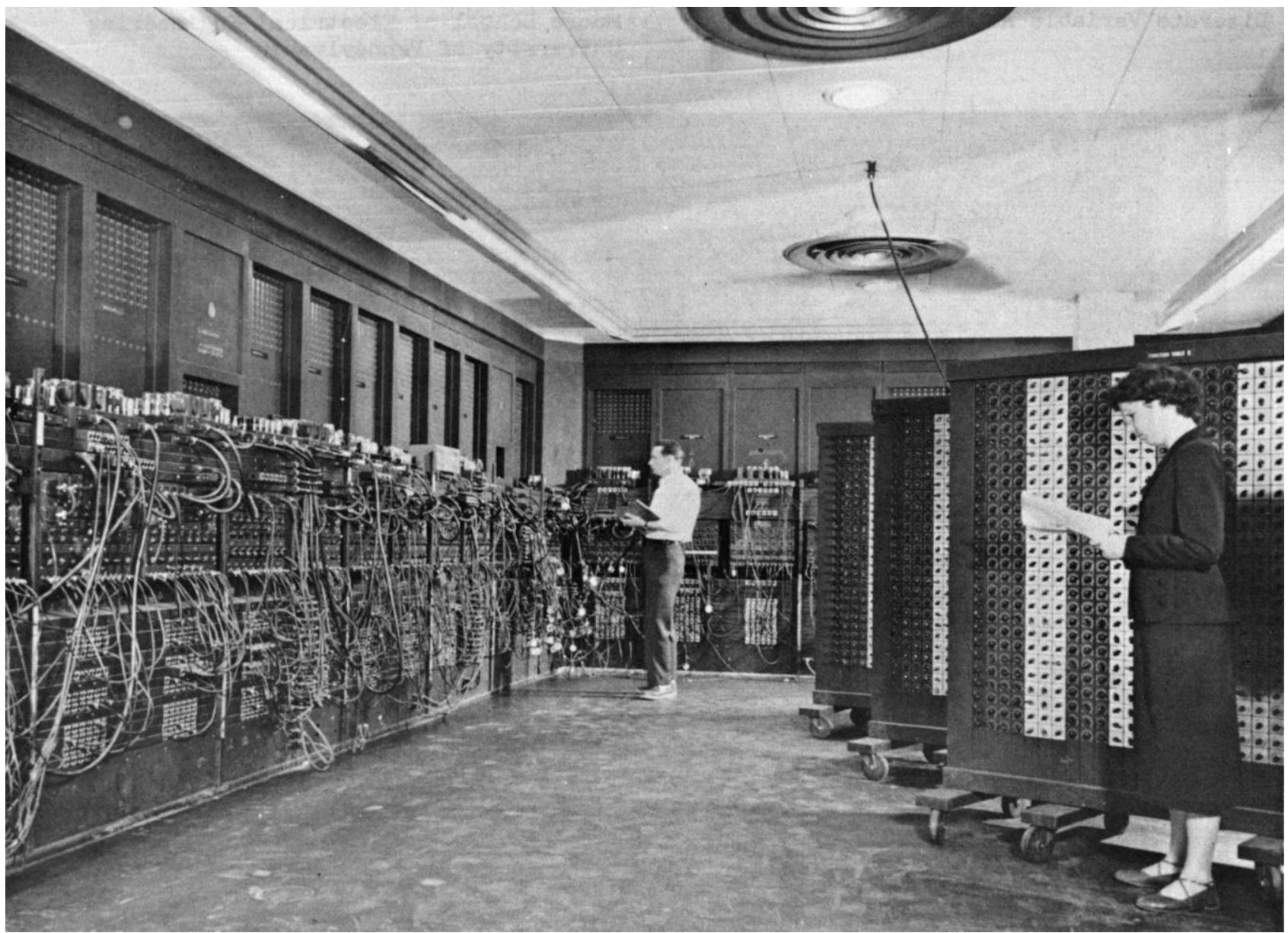
UNIT-I: Introduction to Microprocessor Architecture Introduction and evolution of Microprocessors– Architecture of 8086–Register Organization of 8086–Memory organization of 8086– General bus operation of 8086–Introduction to 80286–80386 and 80486 and Pentium.



The computer as we know it today had its beginning with a 19th century English mathematics professor name **Charles Babbage**.



A portion of Babbage's Difference engine, assembled in 1833, exhibited 1862.



ENIAC (Electronic Numerical Integrator And Computer), 1945

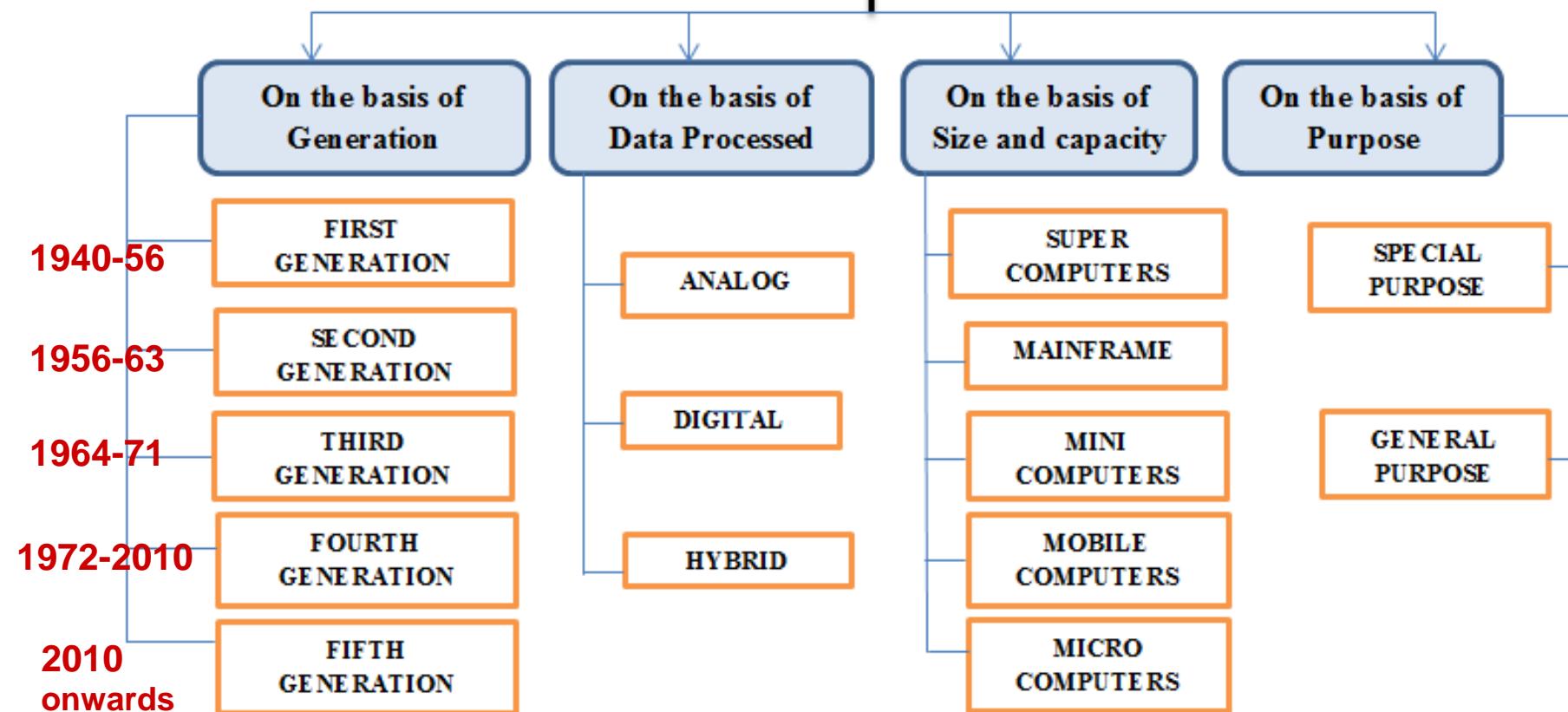
Dr. Venkata Reddy Kota @ JNTUK, Kakinada

EDVAC (Electronic Discrete Variable Automatic Computer) was one of the earliest electronic computers. Unlike its predecessor the ENIAC, it was binary rather than decimal, and was designed to be a stored-program computer.

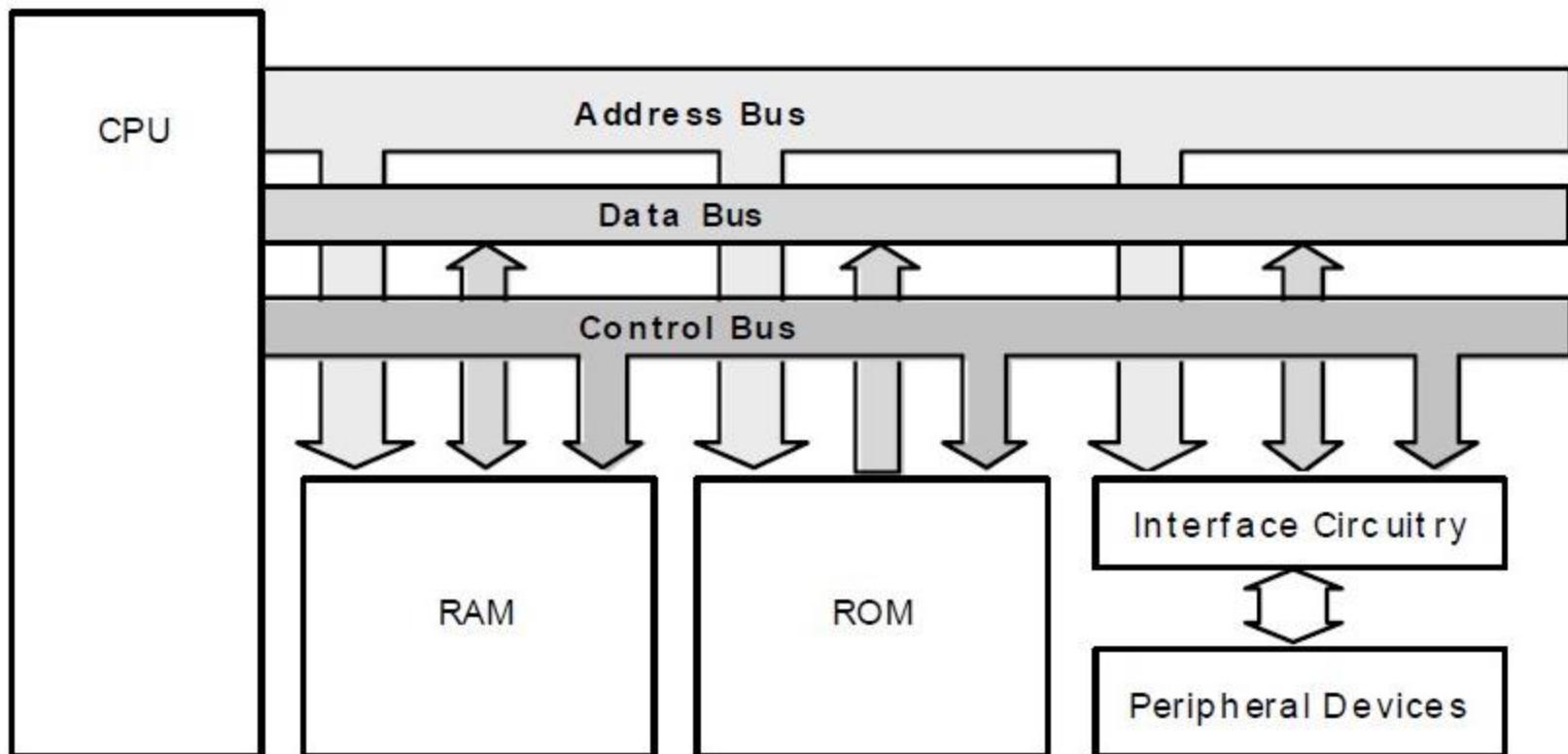
ENIAC inventors John Mauchly and J. Presper Eckert proposed the EDVAC's construction in August 1944 and it was delivered to the Ballistics Research Laboratory in 1949.



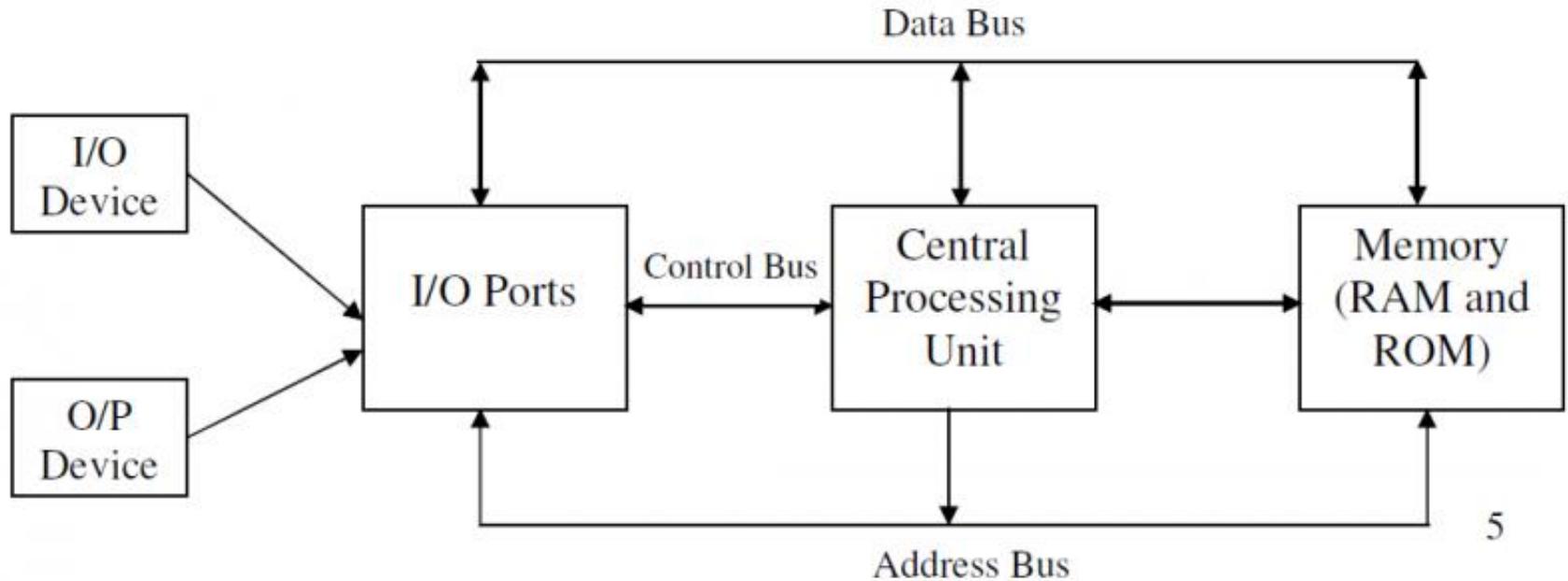
CLASSIFICATIONS OF COMPUTERS



Basic Block Diagram of a Microcomputer



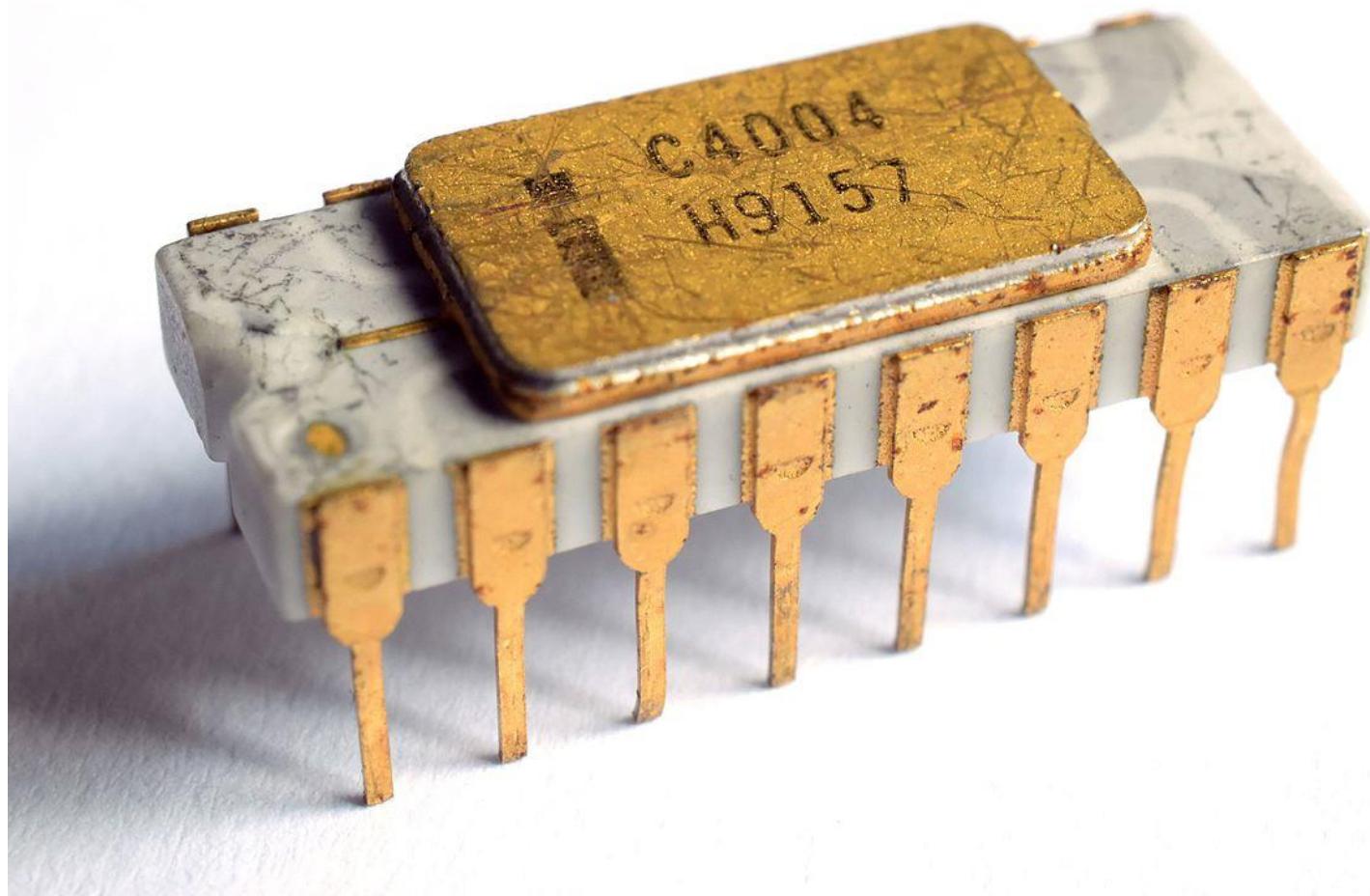
Basic Block Diagram of a Microcomputer



Difference between
microprocessors and
Microcontrollers

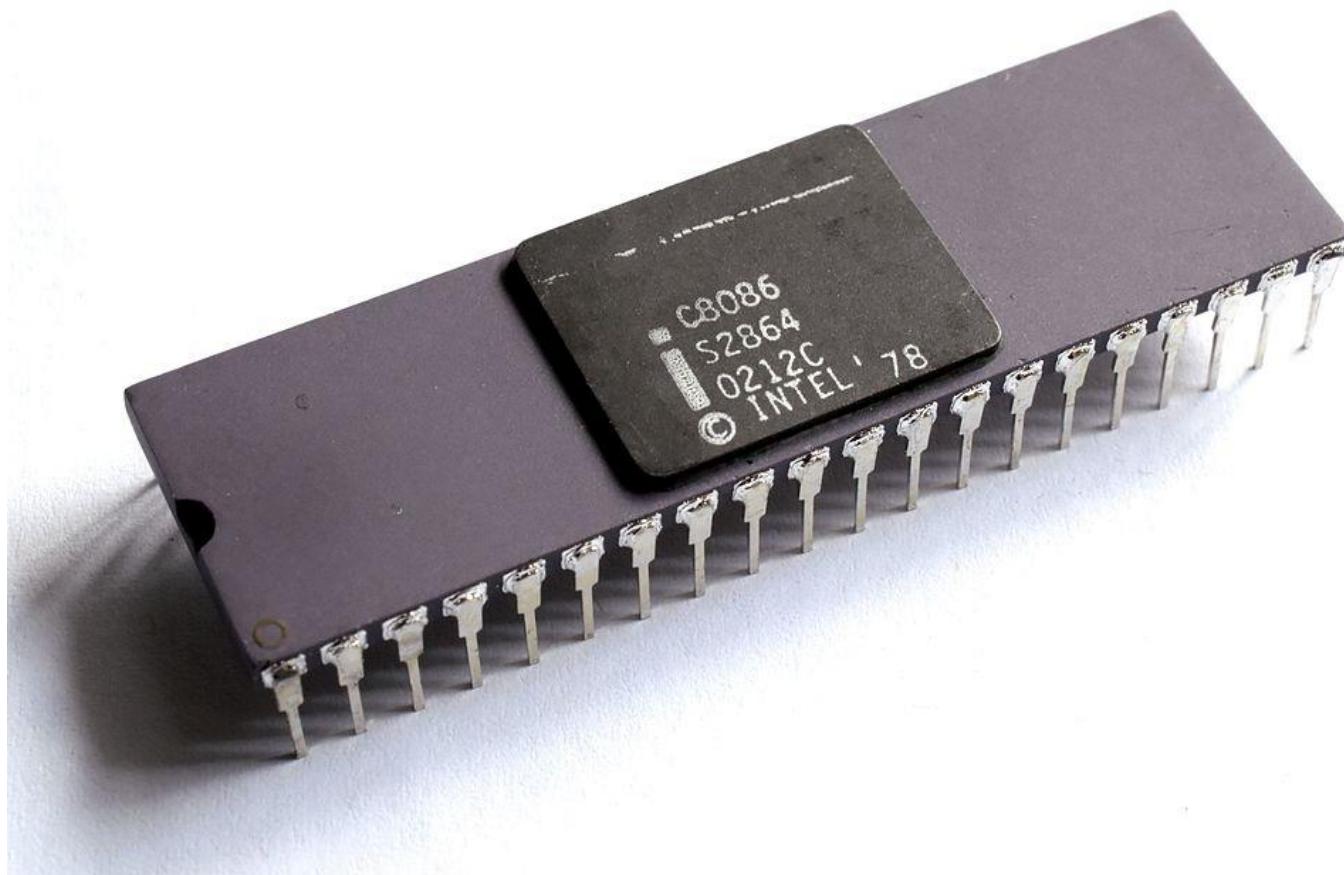
Evolution of Intel Microprocessors

MP	Introduction	Data Bus	Address Bus
4004	1971	4	8
8008	1972	8	8
8080	1974	8	16
8085	1977	8	16
8086	1978	16	20
80186	1982	16	20
80286	1983	16	24
80386	1986	32	32
80486	1989	32	32
Pentium	1993 onwards	32	
Core solo	2006	32	
Dual Core	2006	32	
Core 2 Duo	2006	32	
Core to Quad	2008	32	
i3,i5,i7	2010	64	



Intel 4004:

2300 transistors, 16pin DIP, 740 kHz, 4bit data and 12bit address bus



Intel 8086

Dr. Venkata Reddy Kota @
JNTUK, Kakinada

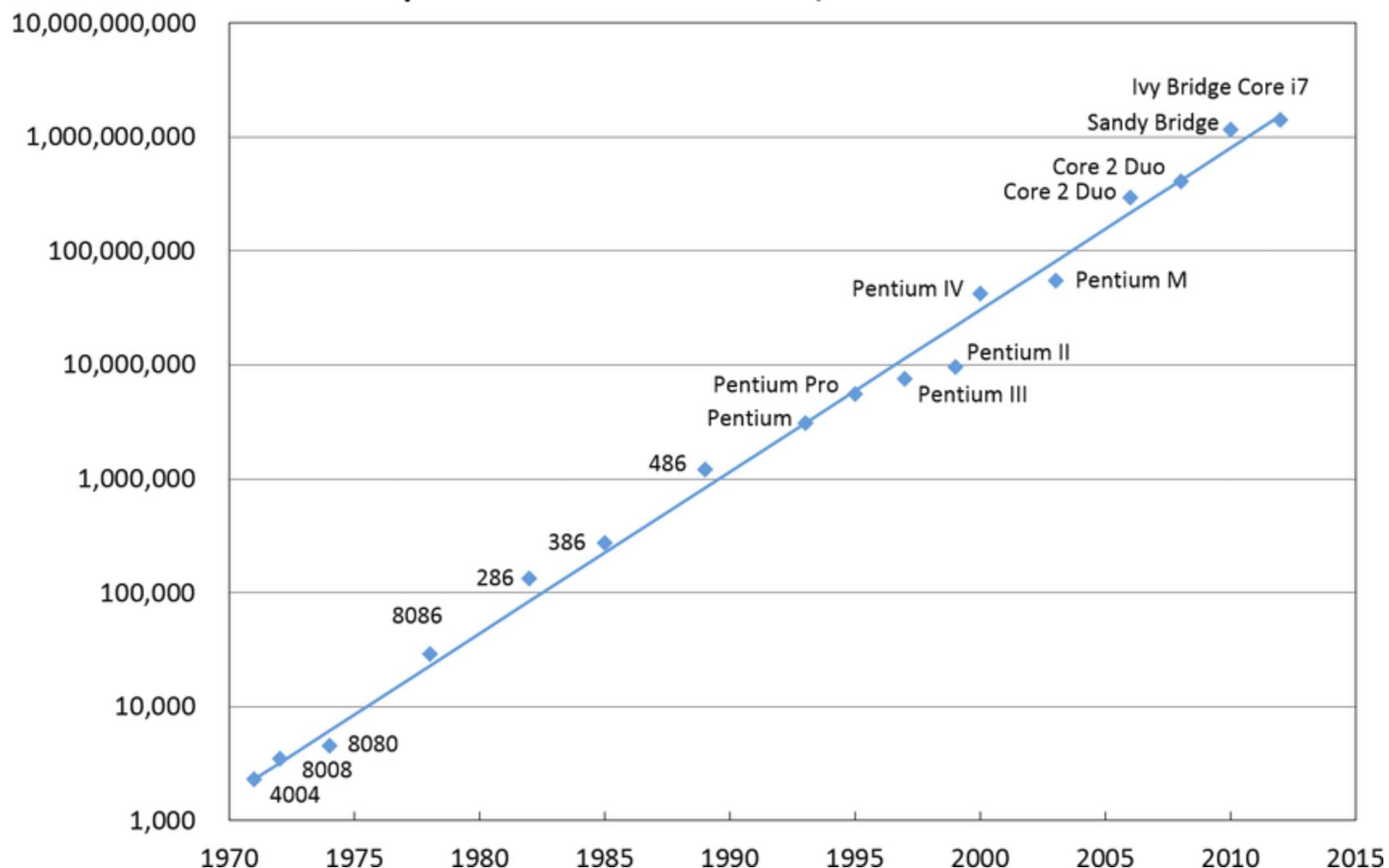


Intel 80386

Dr. Venkata Reddy Kota @
JNTUK, Kakinada



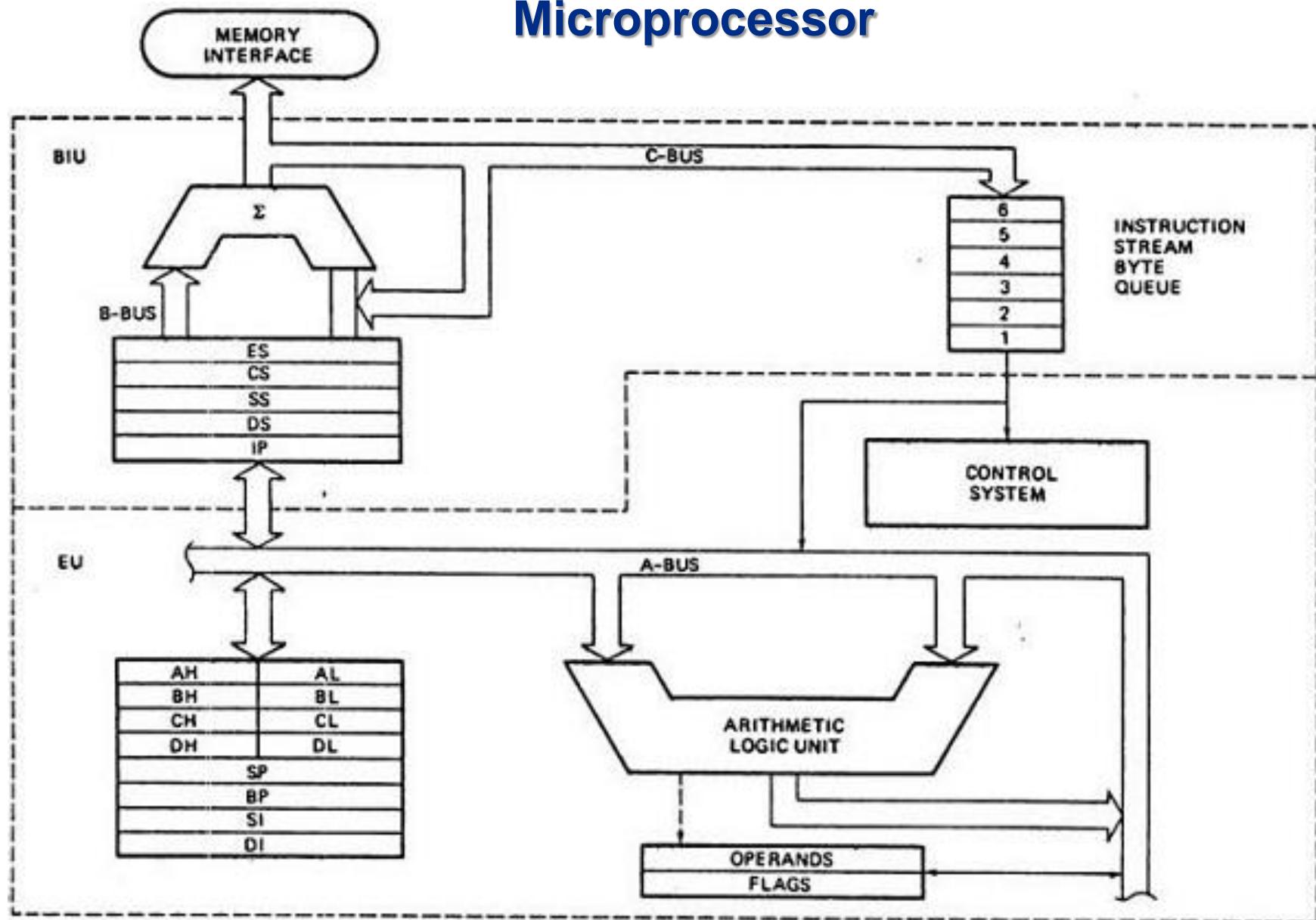
Microprocessor Transistor Count, 1971-2012 & Moore's Law



NAME	YEAR	TRANSISTORS	DATA WIDTH	CLOCK SPEED
8080	1974	6,000	8 bits	2 MHz
8085	1976	6,500	8 bits	5 MHz
8086	1978	29,000	16 bits	5 MHz
8088	1979	29,000	8 bits	5 MHz
80286	1982	134,000	16 bits	6 MHz
80386	1985	275,000	32 bits	16 MHz
80486	1989	1,200,000	32 bits	25 MHz
PENTIUM	1993	3,100,000	32/64 bits	60 MHz
PENTIUM II	1997	7,500,000	64 bits	233 MHz
PENTIUM III	1999	9,500,000	64 bits	450 MHz
PENTIUM IV	2000	42,000,000	64 bits	1.5 GHz

Speed vs Clock frequency

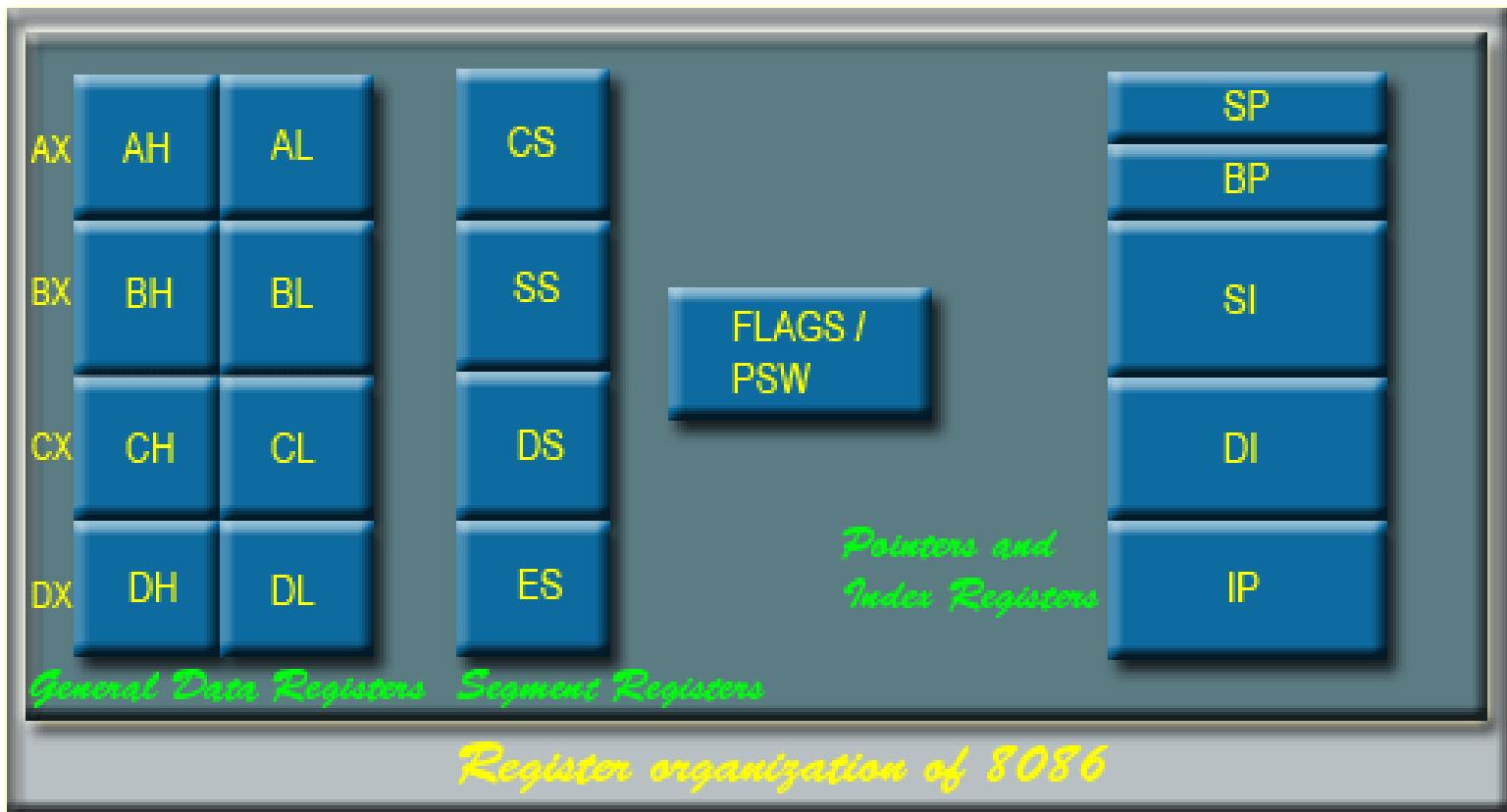
Internal Architecture of 8086 Microprocessor



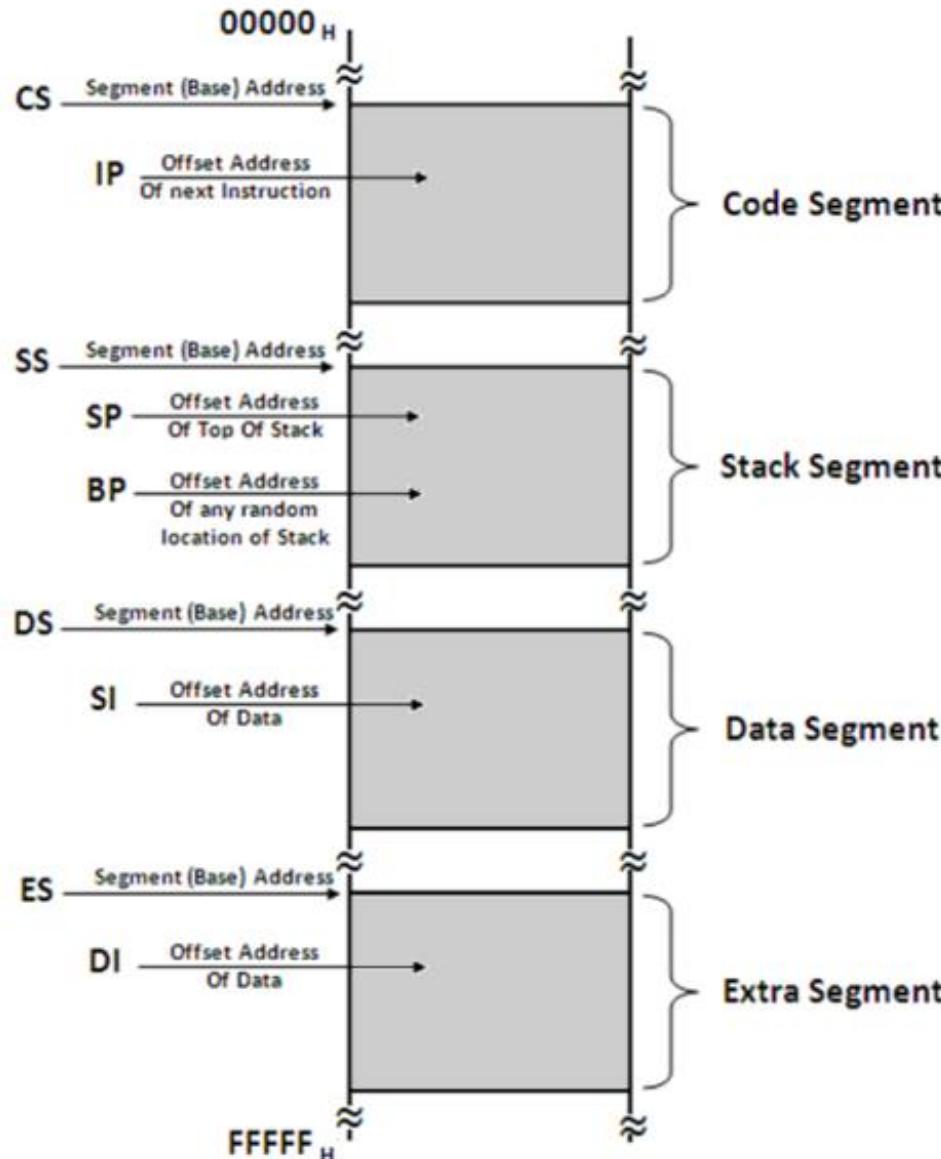
Features of 8086 Microprocessor

- The **8086** is a 16-bit microprocessor chip released by Intel in 1978. It has 16 bit ALU, 16 bit registers and internal data bus and 16 bit external data bus.
- 8088 microprocessor is a 40 pin IC which operate on 5 V power supply.
- The clock frequency of 8086 was originally limited to 5 MHz. It has three version based on the frequency of operation:
 - a) 8086 -> 5MHz
 - b) 8086-2 ->8MHz
 - c) 8086-1 ->10 MHz
- 8086 has 20 bit address lines to access memory. Hence it can access $2^{20} = 1$ MB memory locations. This address space was addressed by means of internal 'segmentation'.
- **8086 uses memory banks:-**The 8086 uses a memory banking system. It means entire data is not stored sequentially in a single memory of 1 MB but memory is divided into two banks of 512KB.
- 8086 has 16-bit address lines to access I/O devices, hence it can access $2^{16} = 64K$ I/O location.
- The 16-bit data bus was multiplexed with the address bus in order to fit a standard 40-pin dual in-line package.

- The 8086 has eight general 16-bit registers (AX, BX, CX, DX, BP, SI, DI, SP) and the instruction pointer, flag register and segment registers. The general purpose registers AX, BX, CX, DX could also be accessed as twice as many 8-bit registers while the other four, BP, SI, DI, SP were 16-bit only.



Memory Segmentation in 8086



Advantages of memory segmentation

- Allow the memory capacity to be 1Mb even though the addresses associated with the individual instructions are only 16 bits wide.
- Facilitate the use of separate memory areas for the program, its data and the stack.
- Permit a program and/or its data to be put into different areas of memory each time the program is executed.
- Multitasking becomes easy.

- 8086 has a 16-bit flags register. Nine of these condition code flags are active, and indicate the current state of the processor: Carry flag (CF), Parity flag (PF), Auxiliary carry flag(AF), Zero flag (ZF), Sign flag (SF), Trap flag (TF), Interrupt flag (IF), Direction flag (DF), and Overflow flag (OF).
- **Pipelining**:-8086 uses two stage of pipelining. First is Fetch Stage and the second is Execute Stage. Fetch stage that prefetch upto 6 bytes of instructions store them in the queue. Execute stage that executes these instructions. Pipelining improves the performance of the processor so that operation is faster.
- **Interrupts**:- 8086 has 256 vectored interrupts.
- **Operates in two modes**:- 8086 operates in two modes:
 - a) Minimum Mode: A system with only one microprocessor.
 - b) Maximum Mode:- A system with multiprocessor.

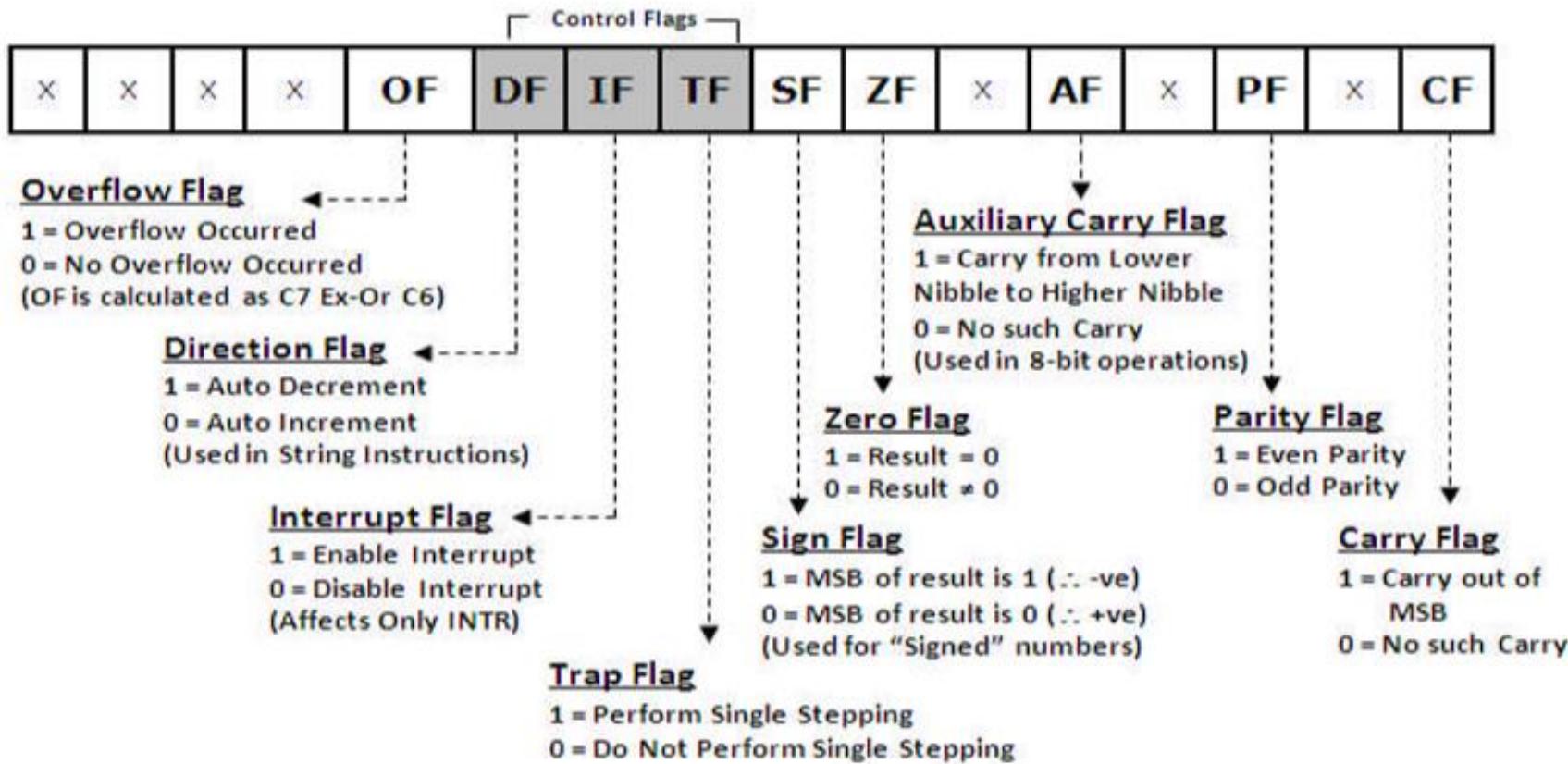
Flag Register

U	U	U	U	OF	DF	IF	TF	SF	ZF	U	AF	U	PF	U	CF
---	---	---	---	----	----	----	----	----	----	---	----	---	----	---	----

1.	CF	CARRY FLAG
2.	PF	PARITY FLAG
3.	AF	AUXILIARY CARRY
4.	ZF	ZERO FLAG
5.	SF	SIGN FLAG
6.	OF	OVERFLOW FLAG
7.	TF	TRAP FLAG
8.	IF	INTERRUPT FLAG
9.	DF	DIRECTION FLAG

Conditional Flags
**(Compatible with 8085,
except OF)**

Control Flags

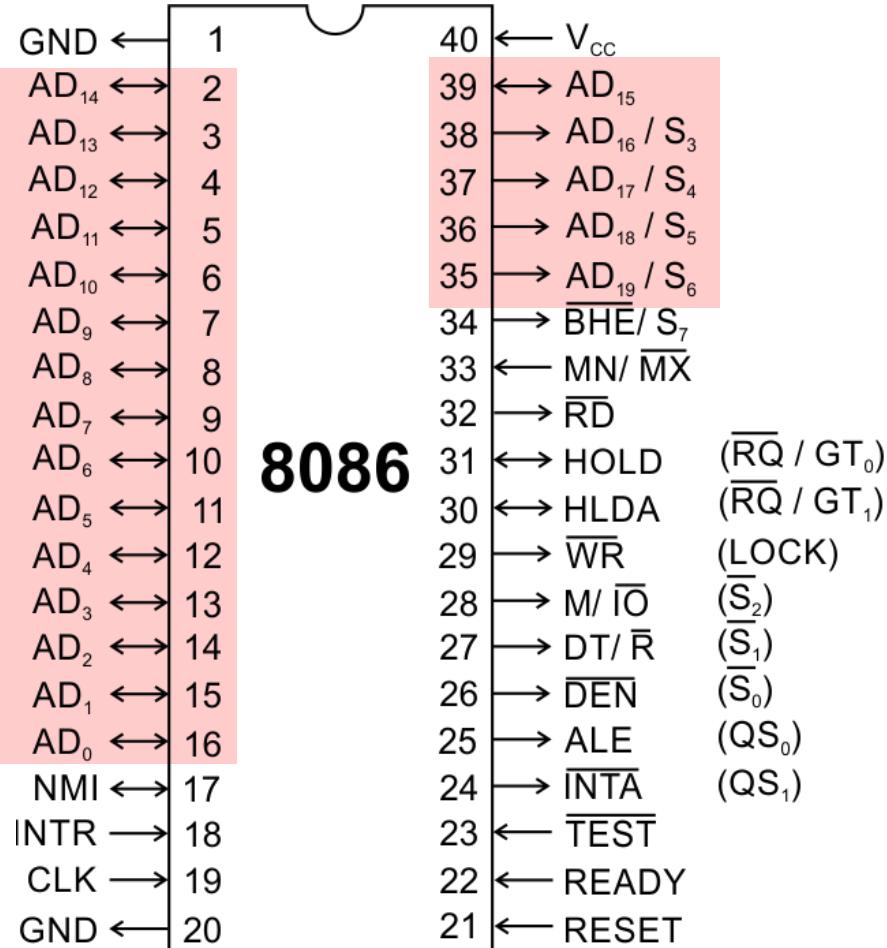


Pin Diagram of 8086 microprocessor

			MAX MODE	MIN MODE
Vss (GND)	1	40	Vcc (5P)	
AD14	2	39	AD15	
AD13	3	38	A16/S3	
AD12	4	37	A17/S4	
AD11	5	36	A18/S5	
AD10	6	35	A19/S6	
AD9	7	34	BHE/S7	
AD8	8	33	MN/MX	
AD7	9	32	RD	
AD6	10	31	RQ/GT0	HOLD
AD5	11	30	RQ/GT1	HLDA
AD4	12	29	LOCK	WR
AD3	13	28	S2	M/I \bar{O}
AD2	14	27	S1	DT/R
AD1	15	26	S0	DEN
AD0	16	25	QS0	ALE
NMI	17	24	QS1	INTA
INTR	18	23	TEST	
CLK	19	22	READY	
Vss (GND)	20	21	RESET	

Functional
pin diagram

Pins and Signals



Common signals

AD_0-AD_{15} (Bidirectional)

Address/Data bus

Low order address bus; these are multiplexed with data.

When AD lines are used to transmit memory address the symbol A is used instead of AD, for example A_0-A_{15} .

When data are transmitted over AD lines the symbol D is used in place of AD, for example D_0-D_7 , D_8-D_{15} or D_0-D_{15} .

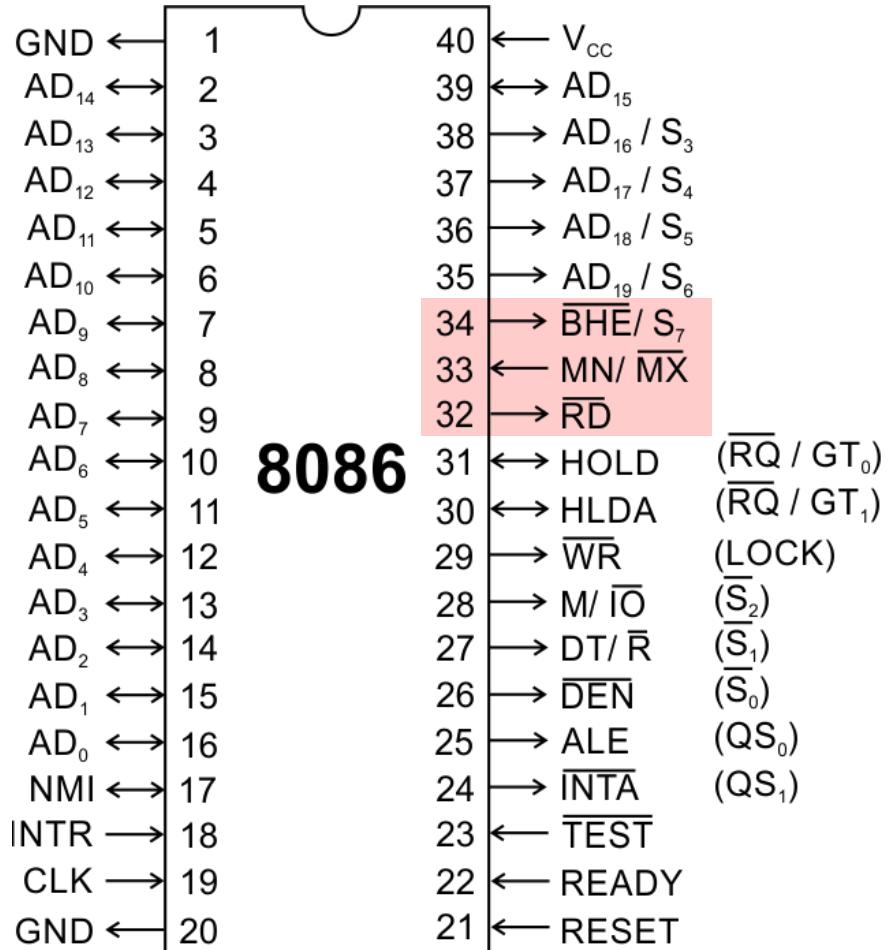
A_{16}/S_3 , A_{17}/S_4 , A_{18}/S_5 , A_{19}/S_6

High order address bus. These are multiplexed with status signals

S_3	S_4	STATUS
0	0	ES
0	1	SS
1	0	CS or idle
1	1	DS

S5 indicates the status of IF flag
S6 is always '0'

Pins and Signals



Common signals

BHE (Active Low)/ S_7 (Output)

Bus High Enable/Status

It is used to enable data onto the most significant half of data bus, D_8-D_{15} . 8-bit device connected to upper half of the data bus use BHE (Active Low) signal. It is multiplexed with status signal S_7 .

BHE	A0	Action
0	0	Access 16-bit word
0	1	Access odd byte to D_8-D_{15}
1	0	Access even byte to D_0-D_7
1	1	No action

MN / MX

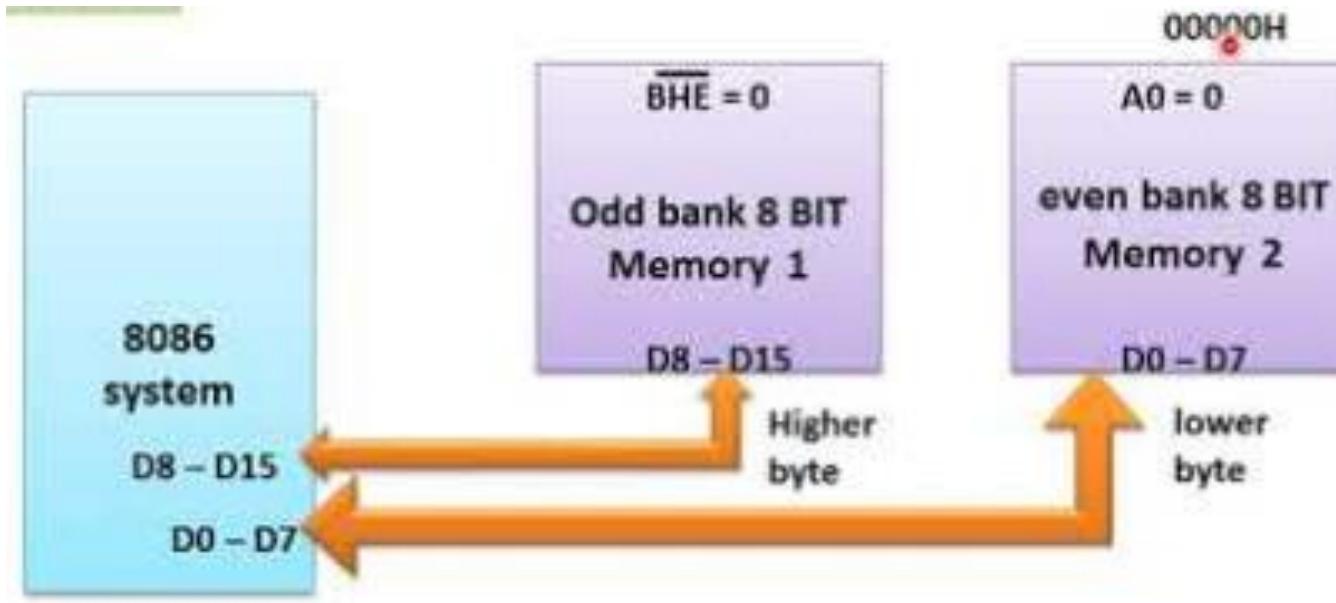
MINIMUM / MAXIMUM

This pin signal indicates what mode the processor is to operate in.

RD (Read) (Active Low)

The signal is used for read operation.
It is an output signal.
It is active when low.

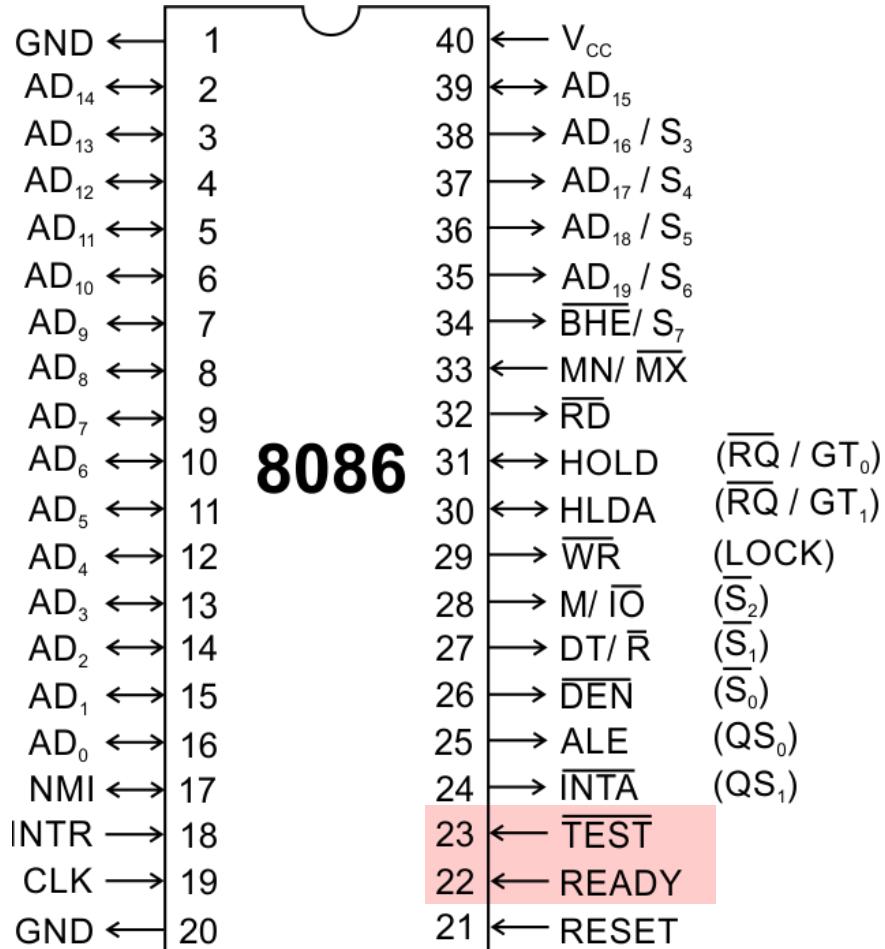
Physical Memory of 8086 microprocessor



BHE	A_0	Operation
0	0	R/W 16-bit from both banks
0	1	R/W 8-bit from higher bank
1	0	R/W 8-bit from lower bank
1	1	No Operation(Idle)

Pins and Signals

Common signals



TEST

TEST input is tested by the 'WAIT' instruction.

8086 will enter a wait state after execution of the WAIT instruction and will resume execution only when the TEST is made low by an active hardware.

This is used to synchronize an external activity to the processor internal operation.

READY

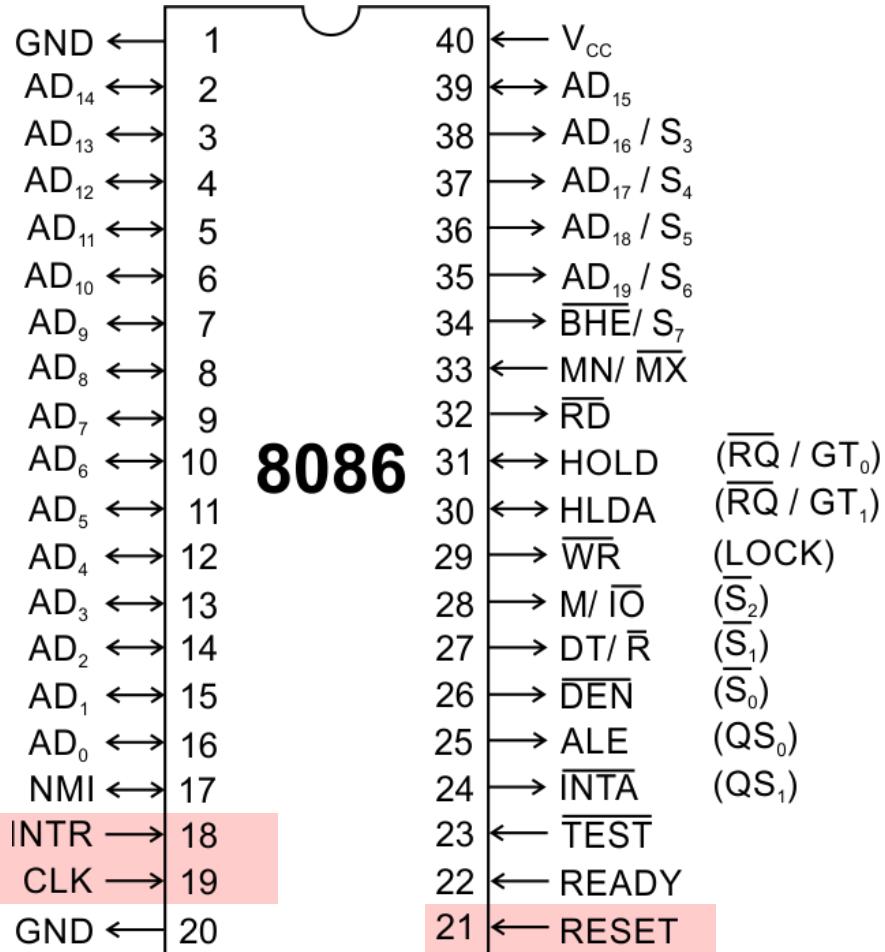
This is the acknowledgement from the slow device or memory that they have completed the data transfer.

The signal made available by the devices is synchronized by the 8284A clock generator to provide ready input to the 8086.

The signal is active high.

Pins and Signals

Common signals



RESET (Input)

Causes the processor to immediately terminate its present activity.

The signal must be active HIGH for at least four clock cycles.

CLK

The clock input provides the basic timing for processor operation and bus control activity. Its an asymmetric square wave with 33% duty cycle.

INTR Interrupt Request

This is a triggered input. This is sampled during the last clock cycles of each instruction to determine the availability of the request. If any interrupt request is pending, the processor enters the interrupt acknowledge cycle.

This signal is active high and internally synchronized.

Interrupt Vector Table of 8086

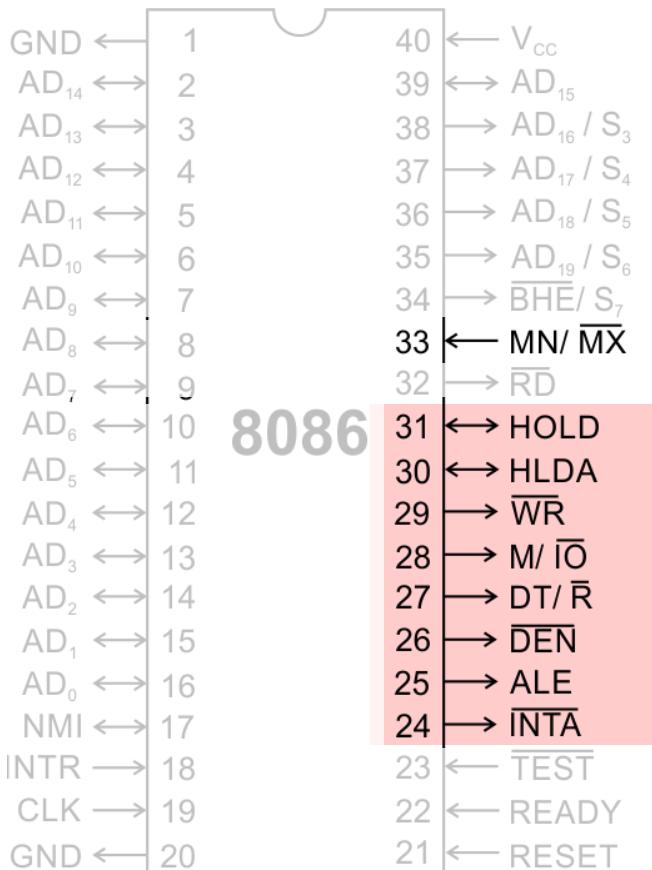
080H	32-255 User defined
	14-31 Reserved
040H	Coprocessor error
03CH	Unassigned
038H	Page fault
034H	General protection
030H	Stack seg overrun
02CH	Segment not present
028H	Invalid task state seg
024H	Coproc seg overrun
020H	Double fault
01CH	Coprocessor not avail
018H	Undefined Opcode
014H	Bound
010H	Overflow (INTO)
00CH	1-byte breakpoint
008H	NMI pin
004H	Single-step
000H	Divide error

The interrupt vector table is located in the first 1024 bytes of memory at addresses 000000H through 0003FFH.

There are 256 4-byte entries (segment and offset in real mode).

Seg high	Seg low	Offset high	Offset low
Byte 3	Byte 2	Byte 1	Byte 0

Pins and Signals



The 8086 microprocessor can work in two modes of operations : **Minimum mode** and **Maximum mode**.

In the minimum mode of operation the microprocessor do not associate with any co-processors and can not be used for multiprocessor systems.

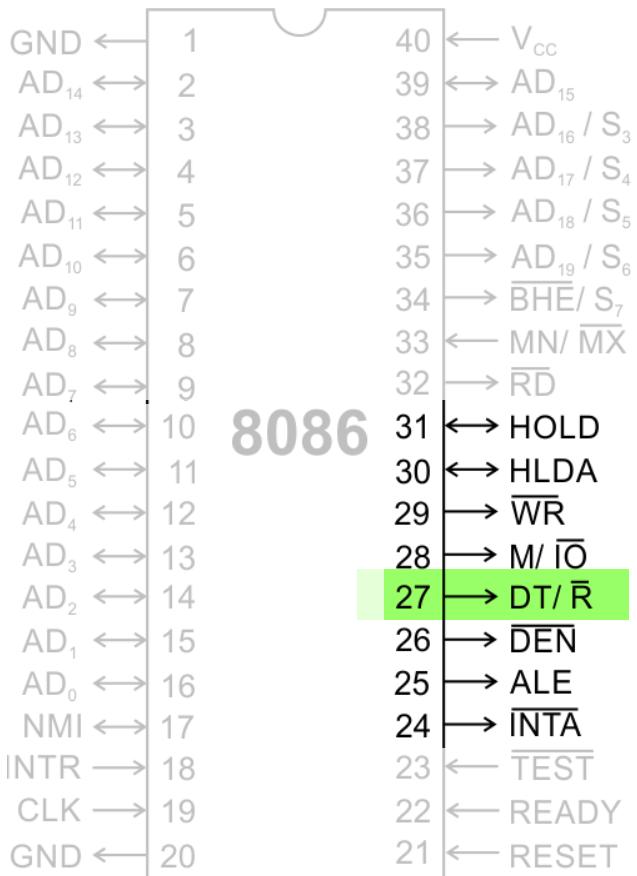
In the maximum mode the 8086 can work in multi-processor or co-processor configuration.

Minimum or maximum mode operations are decided by the pin MN/ MX(Active low).

When this pin is high 8086 operates in minimum mode otherwise it operates in Maximum mode.

Pins and Signals

Minimum mode signals



Pins 24 -31

For minimum mode operation, the MN/ \overline{MX} is tied to VCC (logic high)

8086 itself generates all the bus control signals

DT/̄R

(Data Transmit/ Receive) Output signal from the processor to control the direction of data flow through the data transceivers

DEN

(Data Enable) Output signal from the processor used as output enable for the transceivers

ALE

(Address Latch Enable) Used to demultiplex the address and data lines using external latches

M/̄IO

Used to differentiate memory access and I/O access. For memory reference instructions, it is high. For IN and OUT instructions, it is low.

WR

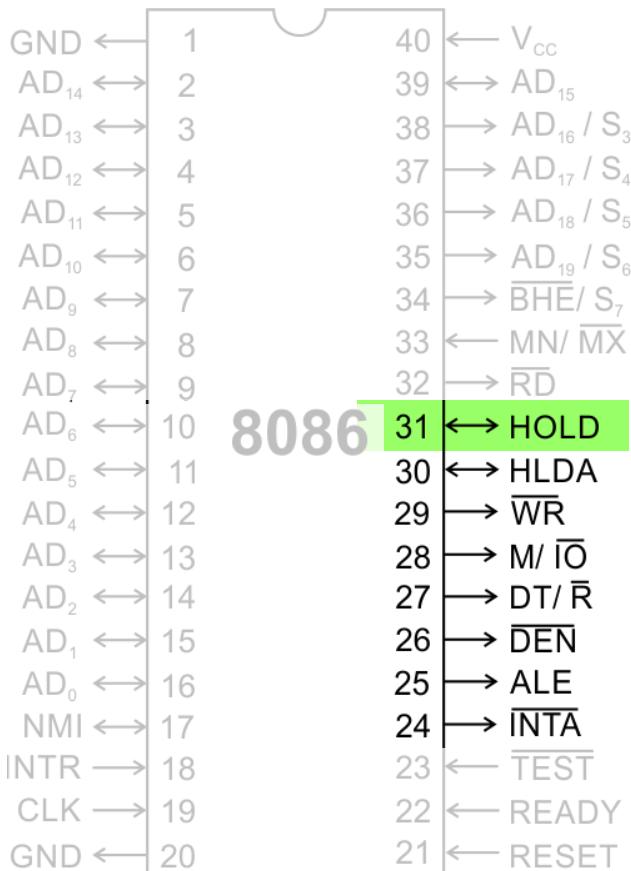
Write control signal; asserted low Whenever processor writes data to memory or I/O port

INTA

(Interrupt Acknowledge) When the interrupt request is accepted by the processor, the output is low on this line.

Pins and Signals

Minimum mode signals



Pins 24 -31

For minimum mode operation, the MN/ \overline{MX} is tied to VCC (logic high)

8086 itself generates all the bus control signals

HOLD

Input signal to the processor from the bus masters as a request to grant the control of the bus.

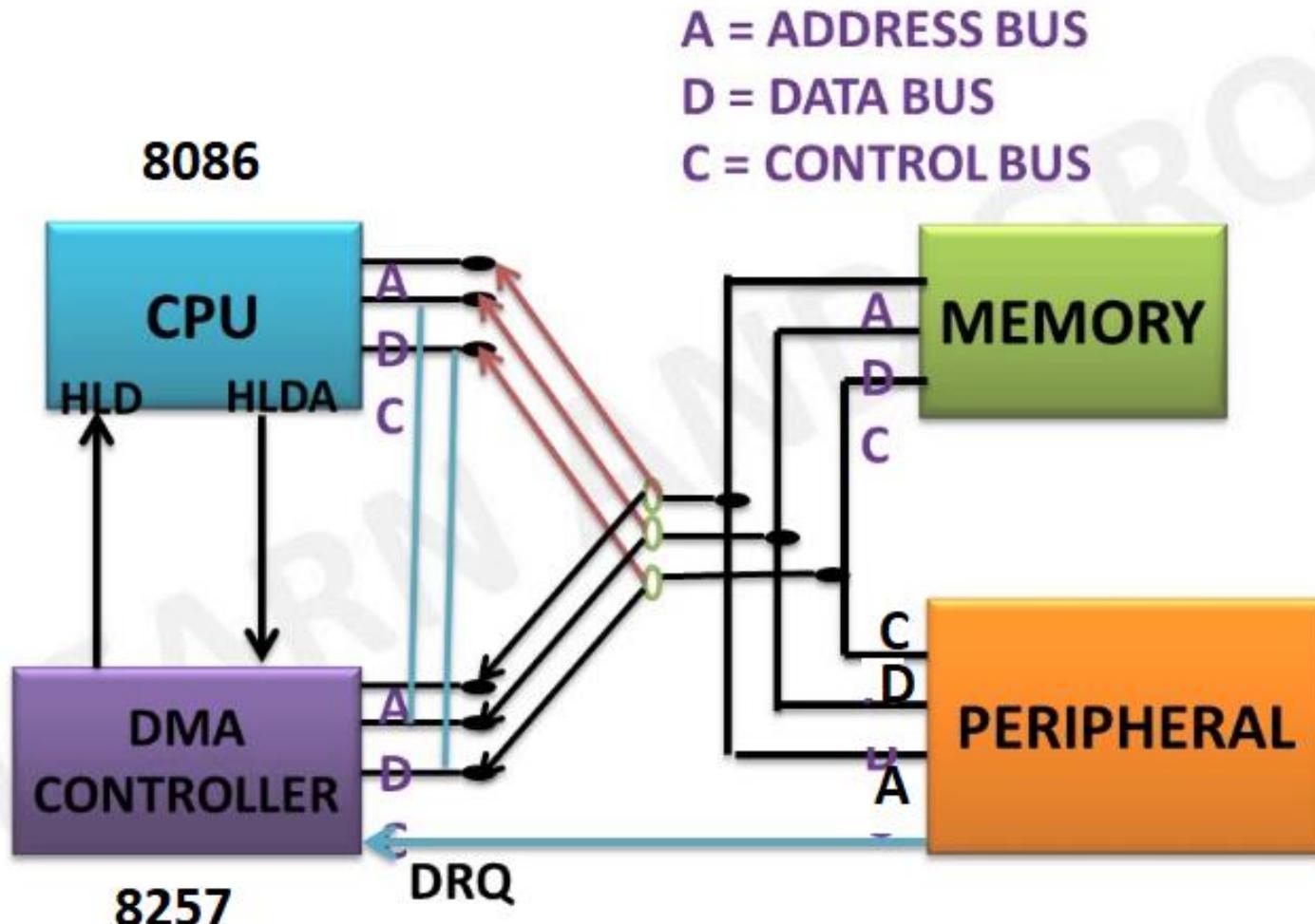
Usually used by the DMA controller to get the control of the bus.

HLDA

(Hold Acknowledge) Acknowledge signal by the processor to the bus master requesting the control of the bus through HOLD.

The acknowledge is asserted high, when the processor accepts HOLD.

Interfacing 8257DMA controller with 8086

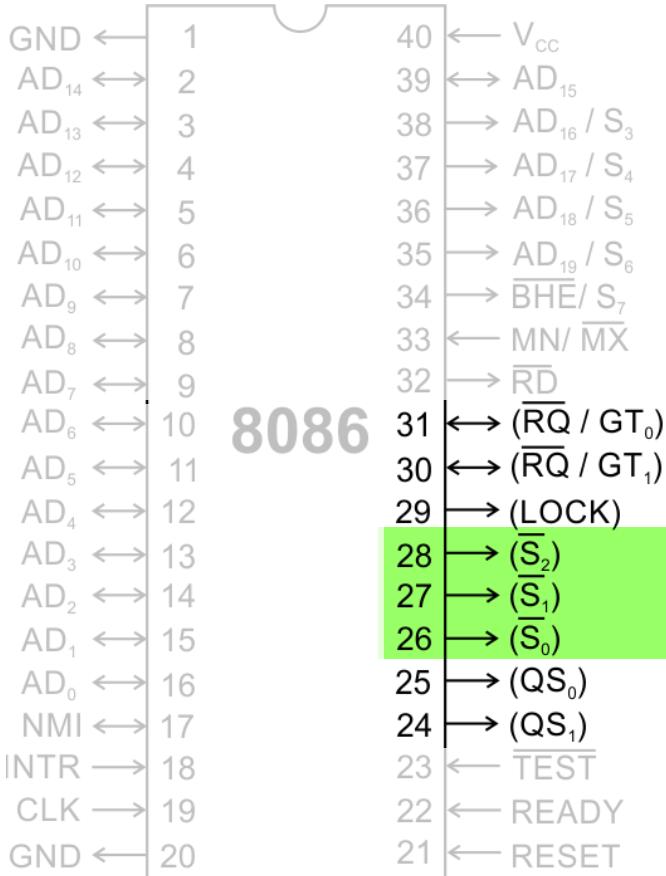


Pins and Signals

Maximum mode signals

During maximum mode operation, the MN/ \overline{MX} is grounded (logic low)

Pins 24 -31 are reassigned



\overline{S}_0 , \overline{S}_1 , \overline{S}_2

Status signals; used by the 8086 bus controller to generate bus timing and control signals. These are decoded as shown.

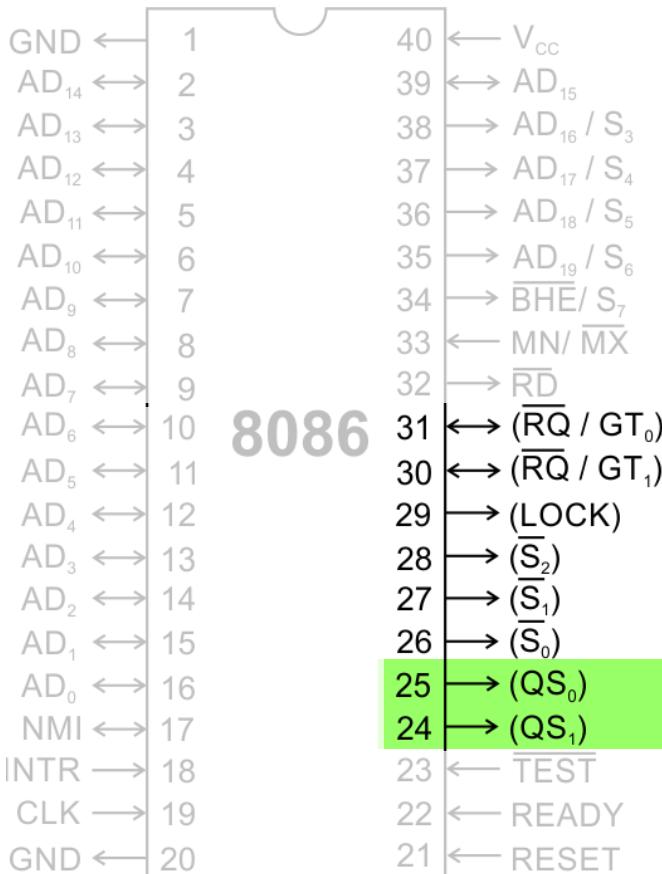
Status Signal			Machine Cycle
\overline{S}_2	\overline{S}_1	\overline{S}_0	
0	0	0	Interrupt acknowledge
0	0	1	Read I/O port
0	1	0	Write I/O port
0	1	1	Halt
1	0	0	Code access
1	0	1	Read memory
1	1	0	Write memory
1	1	1	Passive/Inactive

Pins and Signals

Maximum mode signals

During maximum mode operation, the MN/ \overline{MX} is grounded (logic low)

Pins 24 -31 are reassigned



$\overline{QS_0}$, $\overline{QS_1}$

(Queue Status) The processor provides the status of queue in these lines.

The queue status can be used by external device to track the internal status of the queue in 8086.

The output on QS₀ and QS₁ can be interpreted as shown in the table.

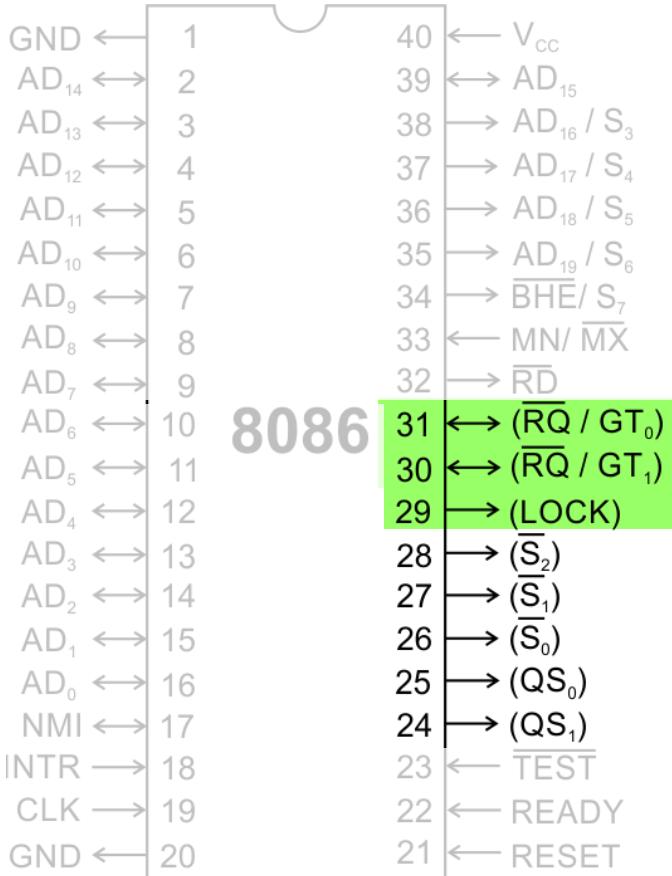
Queue status		Queue operation	
QS ₁	QS ₀		
0	0	No operation	
0	1	First byte of an opcode from queue	
1	0	Empty the queue	
1	1	Subsequent byte from queue	

Pins and Signals

Maximum mode signals

During maximum mode operation, the MN/ \overline{MX} is grounded (logic low)

Pins 24 -31 are reassigned



$\overline{RQ}/ \overline{GT_0},$
 $\overline{RQ}/ \overline{GT_1}$

(Bus Request/ Bus Grant) These requests 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.

These pins are bidirectional.

The request on $\overline{GT_0}$ will have higher priority than $\overline{GT_1}$

An output signal activated by the LOCK prefix instruction.

Remains active until the completion of the instruction prefixed by LOCK.

The 8086 output low on the LOCK pin while executing an instruction prefixed by LOCK to prevent other bus masters from gaining control of the system bus.

The 8086 signals can be categorized in three groups.

The first are the signal having common functions in minimum as well as maximum mode.

AD15-AD0 : These are the time multiplexed memory I/O address and data lines. – Address remains on the lines during T1 state, while the data is available on the data bus during T2, T3, Tw and T4. These lines are active high and float to a tristate during interrupt acknowledge and local bus hold acknowledge cycles

A19/S6,A18/S5,A17/S4,A16/S3 : These are the time multiplexed address and status lines. During T1 these are the most significant address lines for memory operations.

During I/O operations, these lines are low. During memory or I/O operations, status information is available on those lines for T2,T3,Tw and T4.

The status of the interrupt enable flag bit is updated at the beginning of each clock cycle.

The **S4** and **S3** combinely indicate which segment register is presently being used for memory accesses as in below fig.

S4	S3	Segment used
0	0	Extra segment
0	1	Stack segment
1	0	Code segment
1	1	Data segment

These lines float to tri-state off during the local bus hold acknowledge. The status line **S6 is always low**

The address bit are separated from the status bit using latches controlled by the ALE signal

BHE/S7 : The bus high enable is used to indicate the transfer of data over the higher order (D15-D8) data bus . It goes low for the data transfer over D15-D8 and is used to derive chip selects of odd address memory bank or peripherals. BHE is low during T1 for read, write and interrupt acknowledge cycles, whenever a byte is to be transferred on higher byte of data bus. The status information is available during T2, T3 and T4. The signal is active low and tristated during hold. It is low during T1 for the first pulse of the interrupt acknowledge cycle.

RD – Read : This signal on low indicates the peripheral that the processor is performing memory or I/O read operation. RD is active low and shows the state for T2, T3, Tw of any read cycle. The signal remains tristated during the hold acknowledge.

READY : This is the acknowledgement from the slow device or memory that they have completed the data transfer. The signal made available by the devices is synchronized by the 8284A clock generator to provide ready input to the 8086. The signal is active high

INTR-Interrupt Request : This is a triggered input. This is sampled during the last clock cycles of each instruction to determine the availability of the request. If any interrupt request is pending, the processor enters the interrupt acknowledge cycle. This can be internally masked by resulting the interrupt enable flag. This signal is active high and internally synchronized.

TEST : This input is examined by a ‘WAIT’ instruction. If the TEST pin goes low, execution will continue, else the processor remains in an idle state. The input is synchronized internally during each clock cycle on leading edge of clock.

CLK- Clock Input : The clock input provides the basic timing for processor operation and bus control activity. Its an asymmetric square wave with 33% duty cycle.

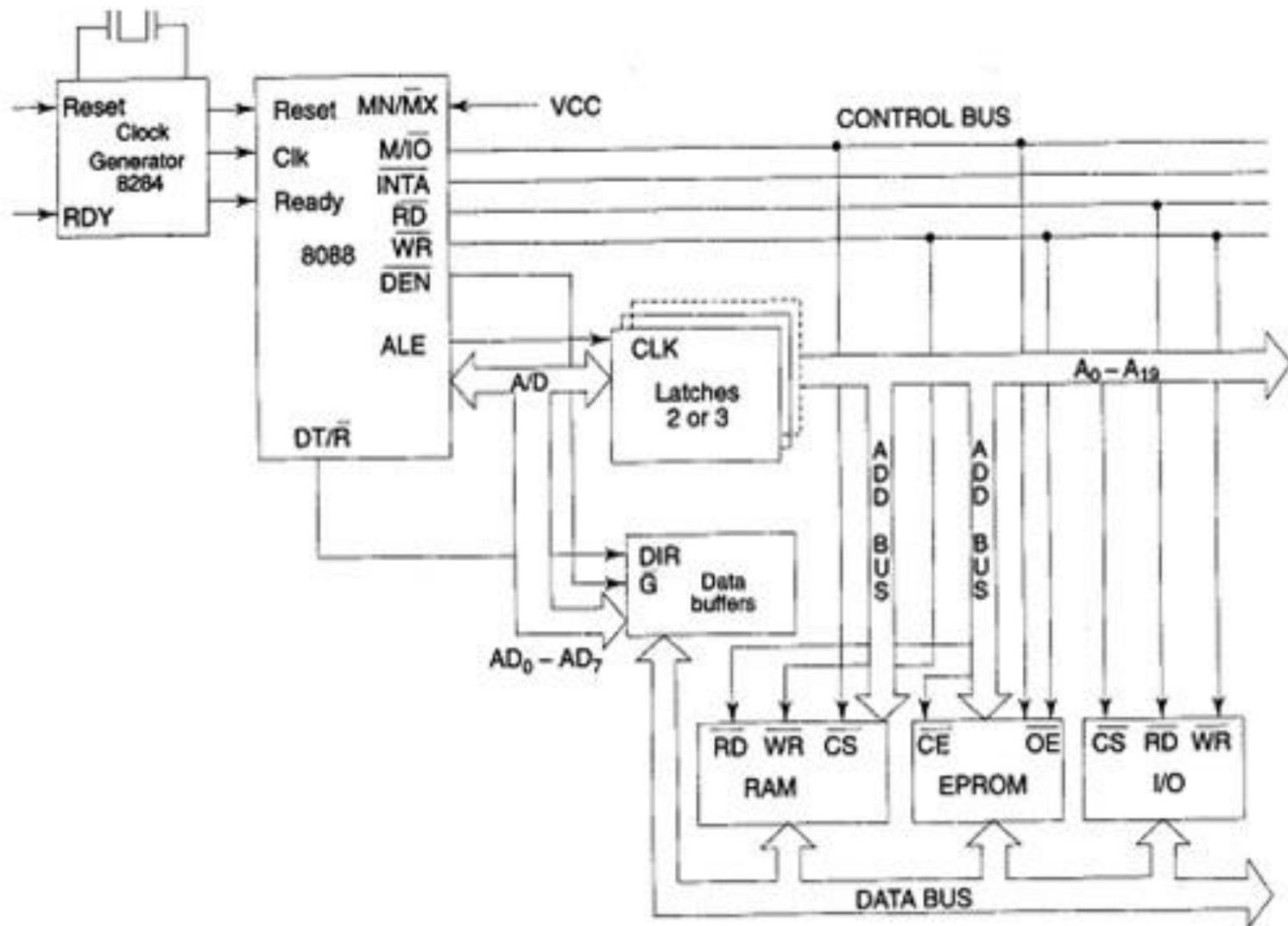
The second are the signals which have special functions for minimum mode.

The third are the signals having special functions for maximum mode.

MINIMUM MODE OF 8086

- In a minimum mode 8086 system, the microprocessor 8086 is operated in minimum mode by strapping its MN/MX pin to logic 1.
- In this mode, all the control signals are given out by the microprocessor chip itself. There is a single microprocessor in the minimum mode system.
- Latches are generally buffered output D-type flip-flops like 74LS373 or 8282. They are used for separating the valid address from the multiplexed address/data signals and are controlled by the ALE signal generated by 8086.
- Transreceivers are the bidirectional buffers and sometimes they are called as data amplifiers. They are required to separate the valid data from the time multiplexed address/data signals.
- They are controlled by two signals namely, DEN and DT/R.
- The DT/R signal indicates the direction of data, i.e. from or to the processor.

M / \overline{IO}	\overline{RD}	\overline{WR}	Transfer Type
0	0	1	I / O read
0	1	0	I/O write
1	0	1	Memory read
1	1	0	Memory write



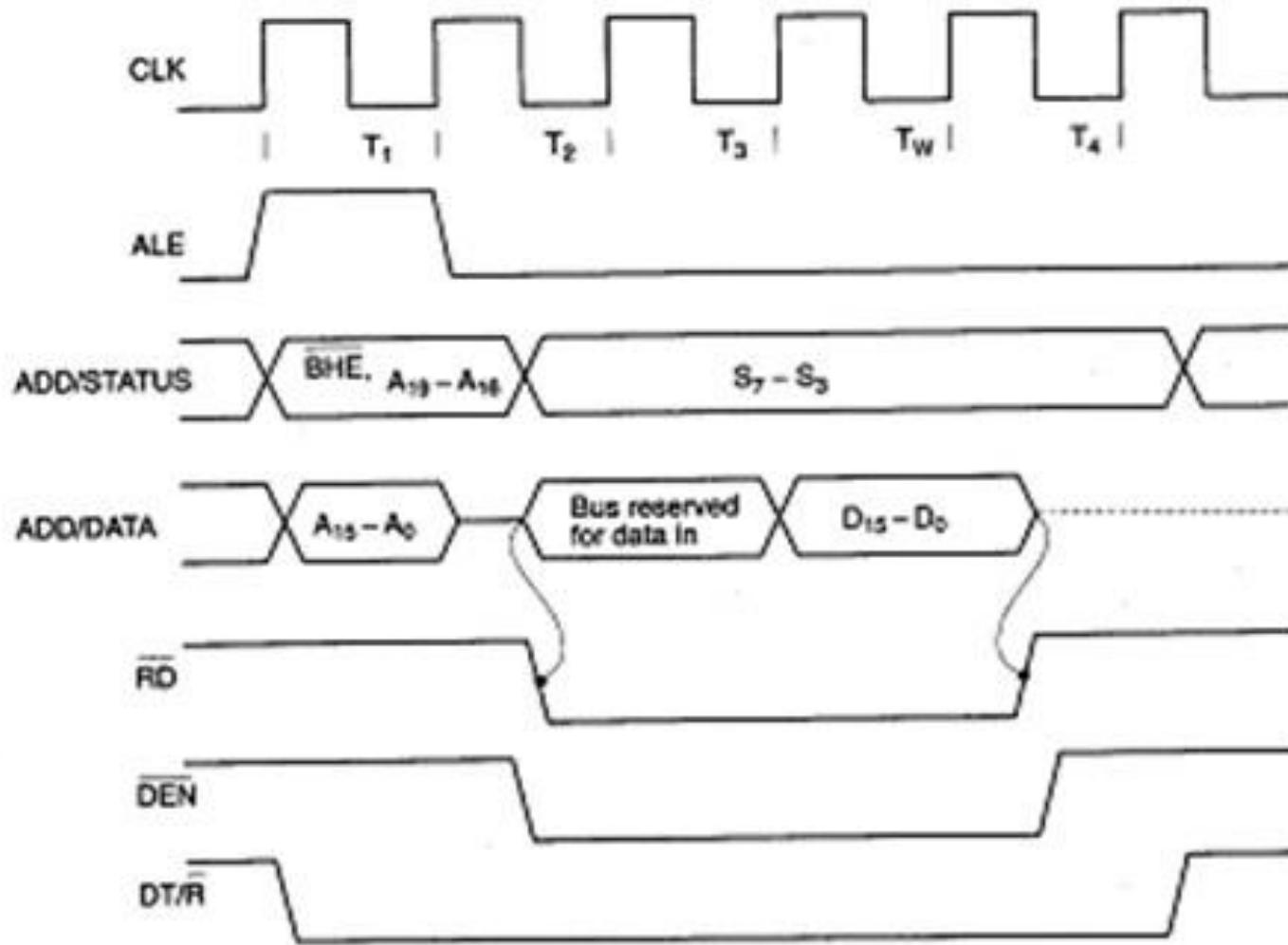
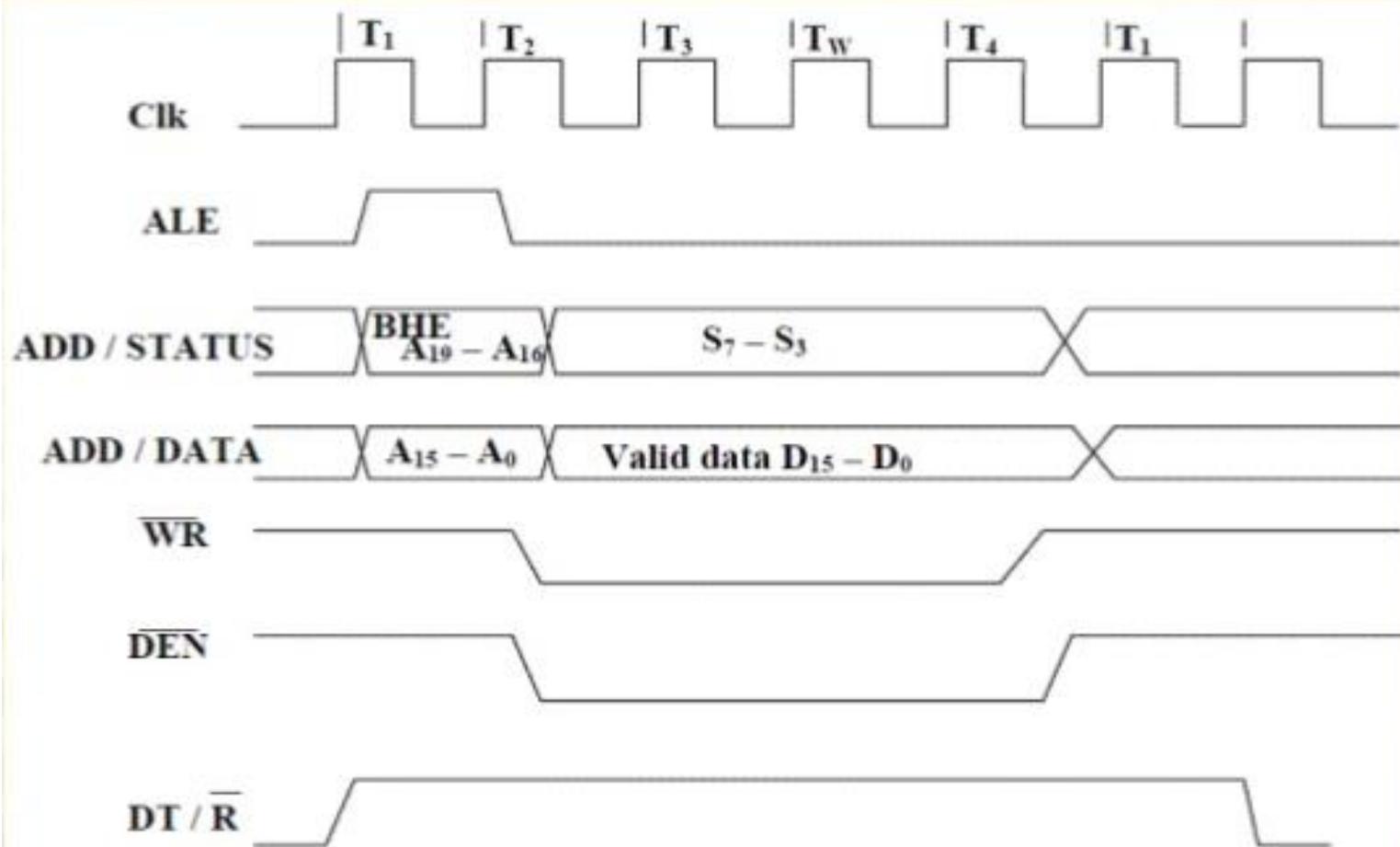


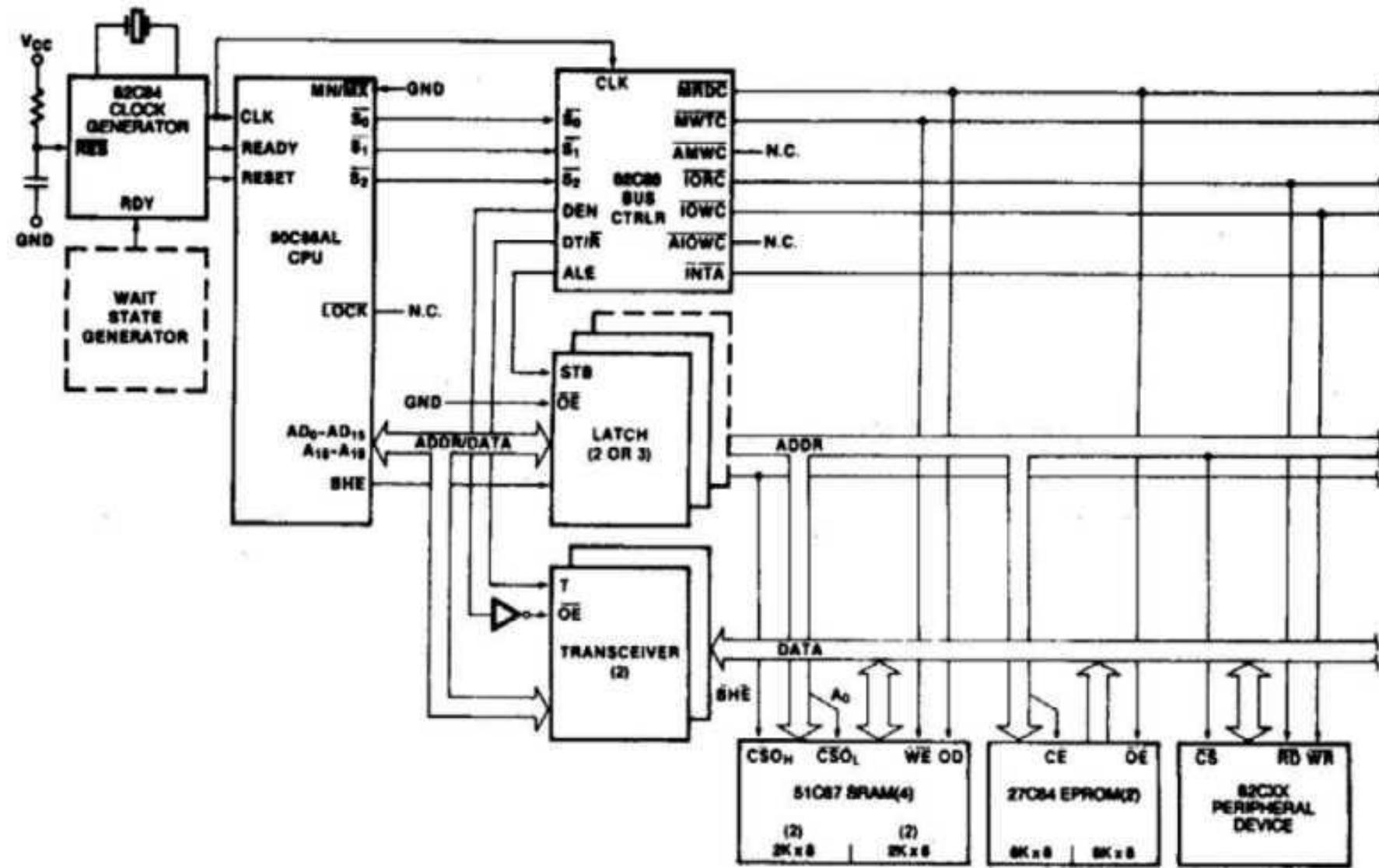
Fig: Read Cycle Timing Diagram in minimum mode



Write Cycle Timing Diagram for Minimum Mode

MAXIMUM MODE OF 8086

- In the maximum mode, the 8086 is operated by strapping the MN/MX pin to ground.
- In this mode, the processor derives the status signal S2, S1, S0. Another chip called bus controller derives the control signal using this status information .
- In the maximum mode, there may be more than one microprocessor in the system configuration.
- The basic function of the bus controller chip IC8288, is to derive control signals like RD and WR (for memory and I/O devices), DEN, DT/R, ALE etc. using the information by the processor on the status lines.
- The bus controller chip has input lines S2, S1, S0 and CLK. These inputs to 8288 are driven by CPU.
- It derives the outputs ALE, DEN, DT/R, MRDC, MWTC, AMWC, IORC, IOWC and AIOWC. The AEN, IOB and CEN pins are specially useful for multiprocessor systems.
- IORC, IOWC are I/O read command and I/O write command signals respectively . These signals enable an IO interface to read or write the data from or to the address port.
- The MRDC, MWTC are memory read command and memory write command signals respectively and may be used as memory read or write signals.



Maximum Mode 80C86AL Typical Configuration

Dr. Venkata Reddy Kota @ JNTUK, Kakinada

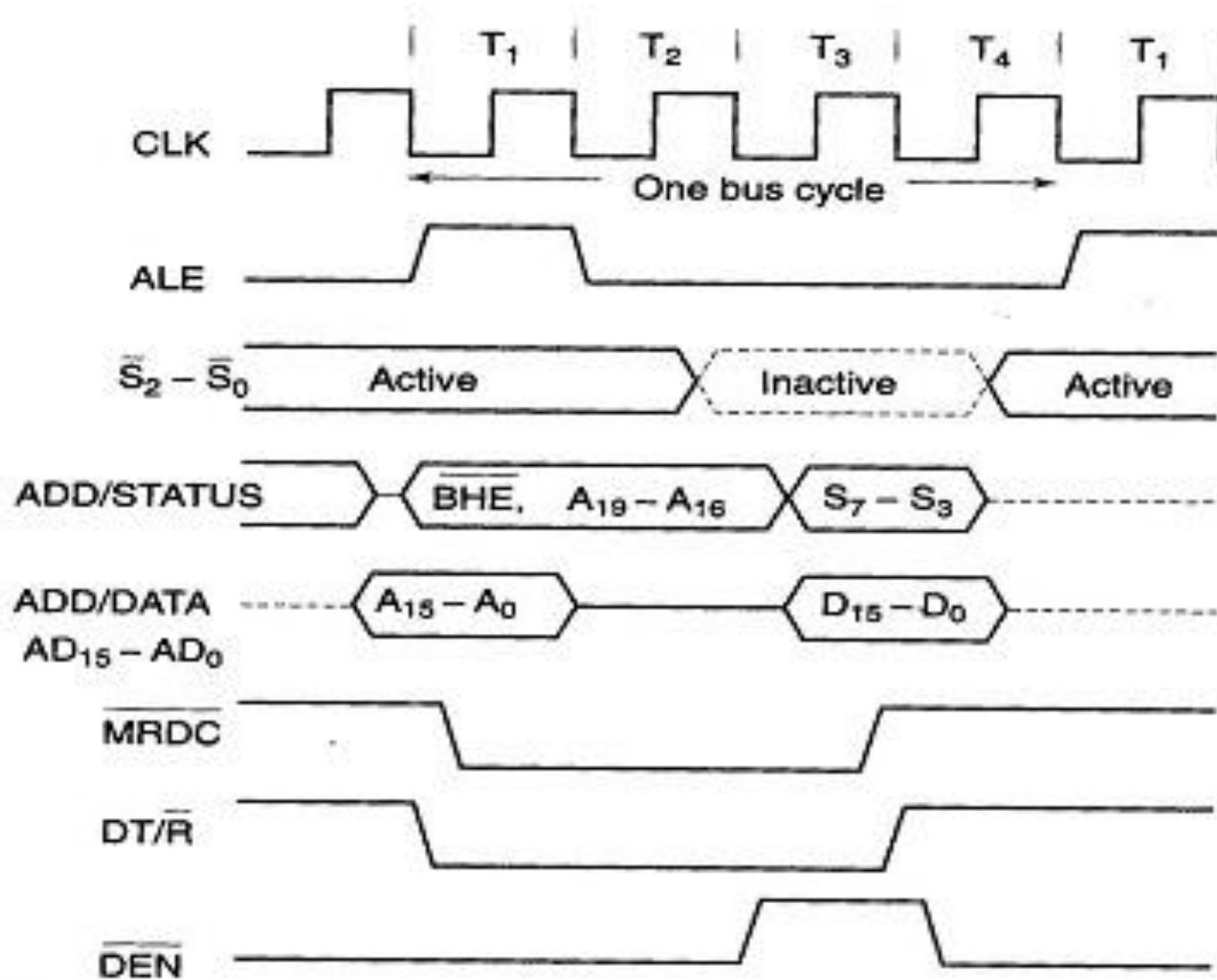


Fig: Read Cycle Timing Diagram in maximum mode

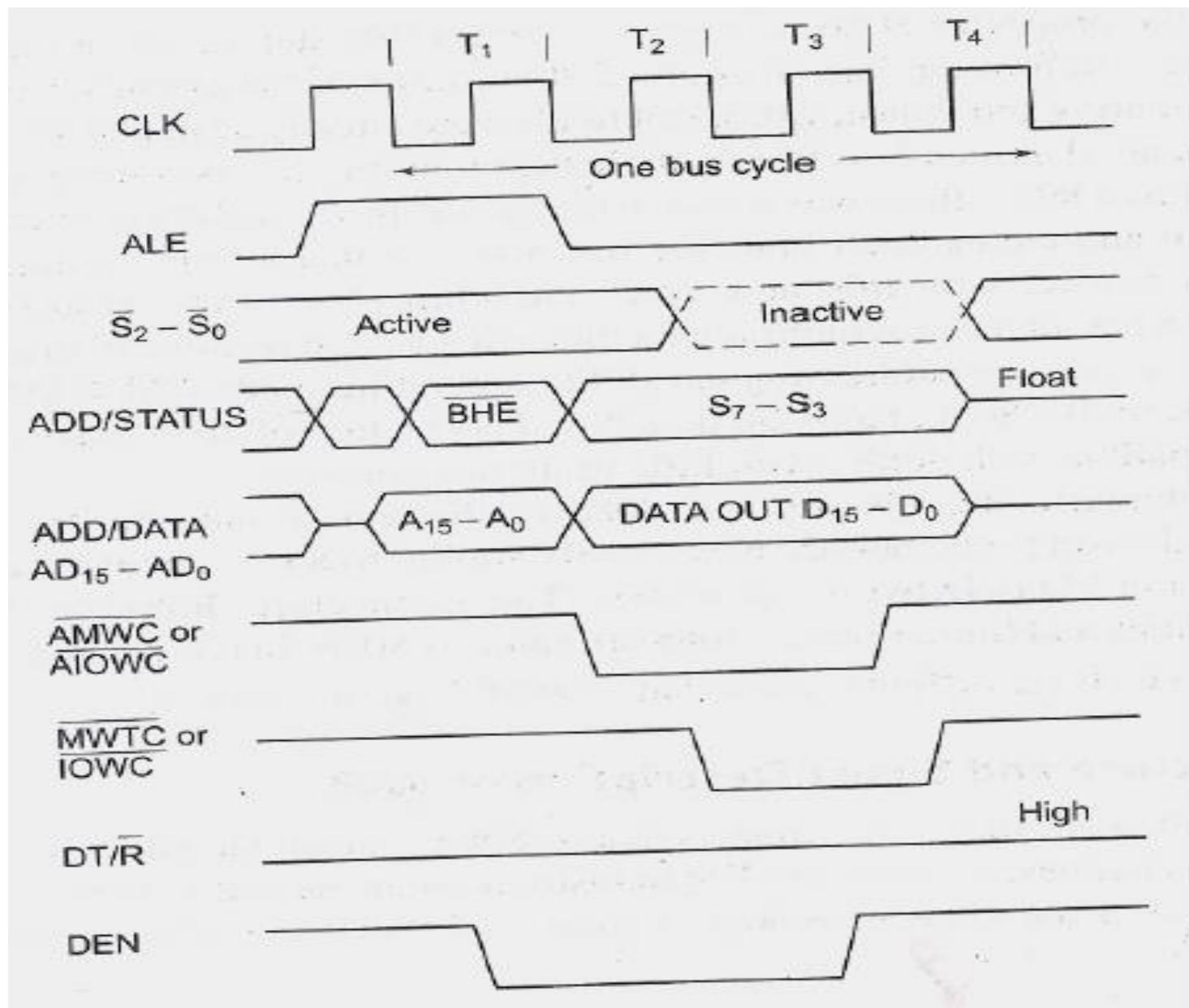


Fig: Write Cycle Timing Diagram in maximum mode

Dr. Venkata Reddy Kota @ JNTUK, Kakinada

80186 Basic Features

- The 80186 contains 16 – bit data bus
- The internal register structure of 80186 is virtually identical to the 8086
- Only difference is that the 80186 contain additional reserved interrupt vectors and some very powerful built-in I/O features.

80286 Basic Features

- The 80286 microprocessor is an advanced version of the 8086 microprocessor that is designed for multi user and multitasking environments.
- The 80286 addresses 16M Byte of physical memory and 1G Bytes of virtual memory by using its memory-management system.
- The 80286 is basically an 8086 that is optimized to execute instructions in fewer clocking periods than the 8086.
- Like the 80186, the 80286 doesn't incorporate internal peripherals; instead it contains a memory management unit (MMU)
- The 80286 operates in both the real and protected modes.
- In the real mode, the 80286 addresses a 1MByte memory address space and is virtually identical to 8086.
- In the protected mode, the 80286 addresses a 16MByte memory space.
- The clock is provided by the 82284 clock generator, and the system control signals are provided by the 82288 system bus controller.
- The 80286 contains the same instructions except for a handful of additional instructions that control the memory-management nit.

80386 Basic Features

- The 80386 microprocessor is an enhanced version of the 80286 microprocessor and includes a **memory-management unit** is enhanced to provide memory paging.
- The 80386 also includes **32-bit extended registers** and a **32-bit address and data bus**
- The 80386 has a physical **memory size of 4GBytes** that can be addressed as a **virtual memory with up to 64TBytes**.
- The 80386 is operated in the **pipelined mode**, it sends the address of the next instruction or memory data to the memory system prior to completing the execution of the current instruction.
- This allows the memory system to begin fetching the next instruction or data before the current is completed.
- This increases access time, thus reducing the speed of the memory.
- The I/O structure of the 80386 is almost identical to the 80286, except that I/O can be inhibited when the 80386 is operated in the protected mode through the I/O bit protection map.
- The register set of the 80386 contains **extended versions of the registers** introduced on the 80286 microprocessor. These extended registers include EAX, EBX, ECX, EDX, EBP, ESP, EDI, ESI, EIP and EFLAGS.

80386 Basic Features contd...

- The instruction set of the 80386 is enhanced to include instructions that address the **32-bit extended register set**.
- Interrupts, in the 80386 microprocessor, have been expanded to include additional predefined interrupts in the interrupt vector table.
- The 80386 memory manager is similar to the 80286, except the physical addresses generated by the MMU are 32 bits wide instead of 24-bits.
- The 80386 is also capable of **paging**.
- The 80386 is operated in the real mode (i.e. 8086 mode) when it is reset.
- The **real mode** allows the microprocessor to address data in the first **1MByte** of memory.
- In the **protected mode**, 80386 addresses any location in its **4G bytes** of physical address space.

80486 Basic Features

- The 80486 microprocessor is an improved version of the 80386 microprocessor that contains an **8K-byte cache** and an **80387 arithmetic co processor**; it executes many instructions in one clocking period.
- The 80486 microprocessor executes a few new instructions that control the internal cache memory.
- A new feature found in the 80486 in the BIST (builtin self-test) that tests the microprocessor, coprocessor, and cache at reset time v If the 80486 passes the test, EAX contains a zero.
- Additional test registers are added to the 80486 to allow the cache memory to be tested.
- These new test registers are TR3 (cache data), TR4 (cache status), and TR5 (cache control).

Pentium Basic Features

- The Pentium microprocessor is almost identical to the earlier 80386 and 80486 microprocessors.
- The main difference is that the Pentium has been modified internally to contain **a dual cache (instruction and data)** and a dual integer unit.
- The Pentium also operates at a higher clock speed of **66 MHz**
- The data bus on the Pentium is **64 – bits** wide and contains eight byte-wide memory banks selected with bank enable signals
- Memory access time, without wait states, is only about 18 ns in the 66 MHz Pentium
- The superscalar structure of the Pentium contains three independent processing units: **a floating point processor and two integer processing units**
- A new mode of operation called the System Memory Management (SMM) mode has been added to the Pentium. It is intended for high-level system functions such as power management and security
- The Built-in Self-test (BIST) allows the Pentium to be tested when power is first applied to the system
- Allows 4MByte memory pages instead of the 4KByte pages

Pentium Pro Basic Features

- The Pentium Pro is an enhanced version of the Pentium microprocessor that contains not only the level 1 caches found inside the Pentium, but the level 2 cache of 256 K or 512K found on most main boards
- The Pentium Pro operates using the same 66 MHz bus speed as the Pentium and the 80486
- It uses an internal clock generator to multiply the bus speed by various factors to obtain higher internal execution speeds
- The only significant software difference between the Pentium Pro and earlier microprocessors is the addition of **FCMOV** and **CMOV** instructions
- The only hardware difference between the Pentium Pro and earlier microprocessors is the addition of **2M paging** and **four extra address lines** that allow access to a memory address space of 64G Bytes

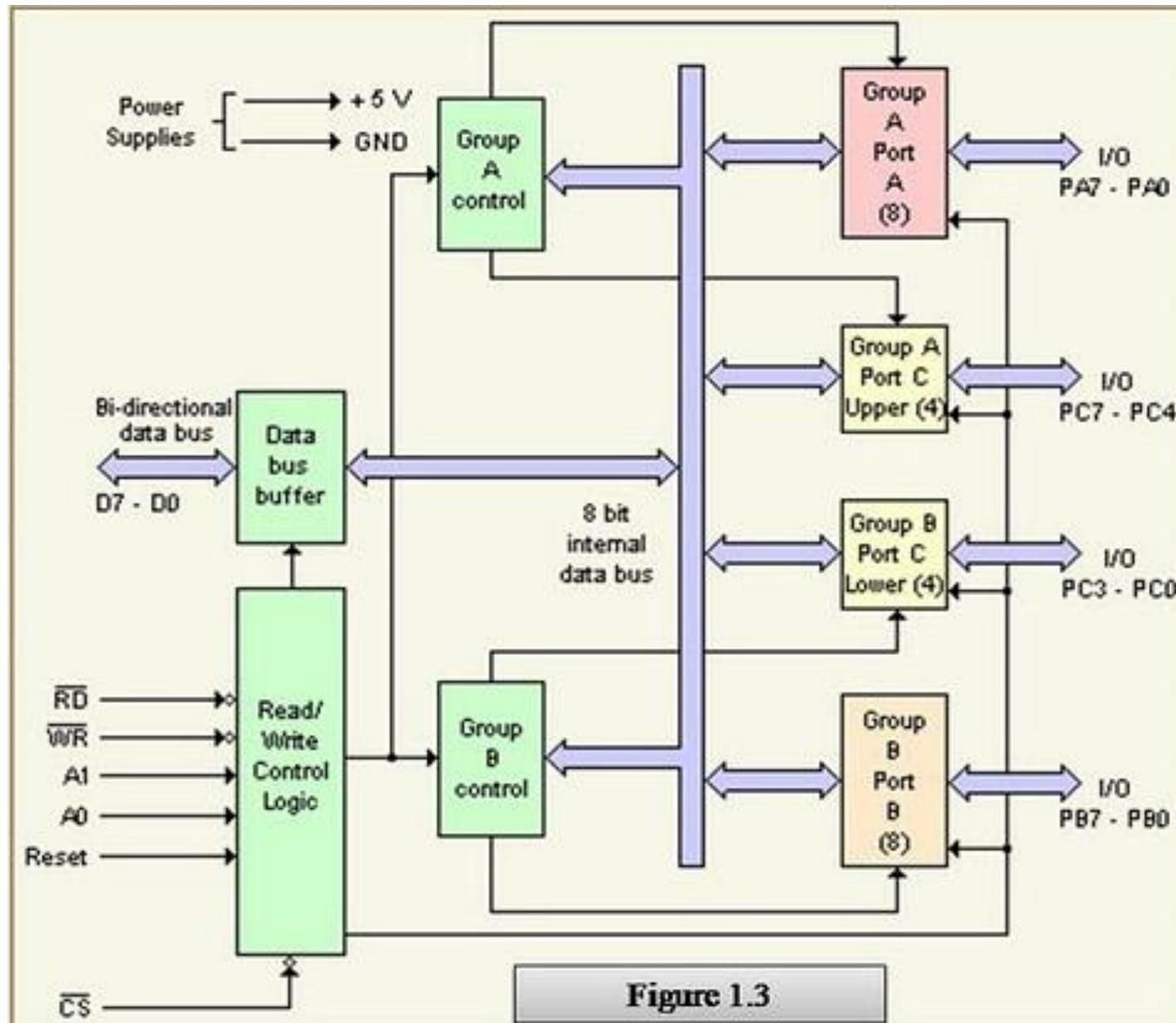
8255 - Programmable Peripheral Interface

- The **Intel 8255** (or **i8255**) Programmable Peripheral Interface (PPI) chip was developed and manufactured by Intel in the first half of the 1970s for the Intel 8080 microprocessor.
- The 8255 provides 24 parallel input/output lines with a variety of programmable operating modes.
- It is an I/O port chip used for interfacing I/O devices with microprocessor. It is a very commonly used peripheral chip.
- PPI has 24 pins for I/O that are programmable in 2 groups of 12 pins each and has three distinct modes of operation.
- The 8255 was widely used in many microcomputer/microcontroller systems and home computers such as the SV-328 and all MSX models. The 8255 was used in the original IBM-PC, PC/XT, etc.

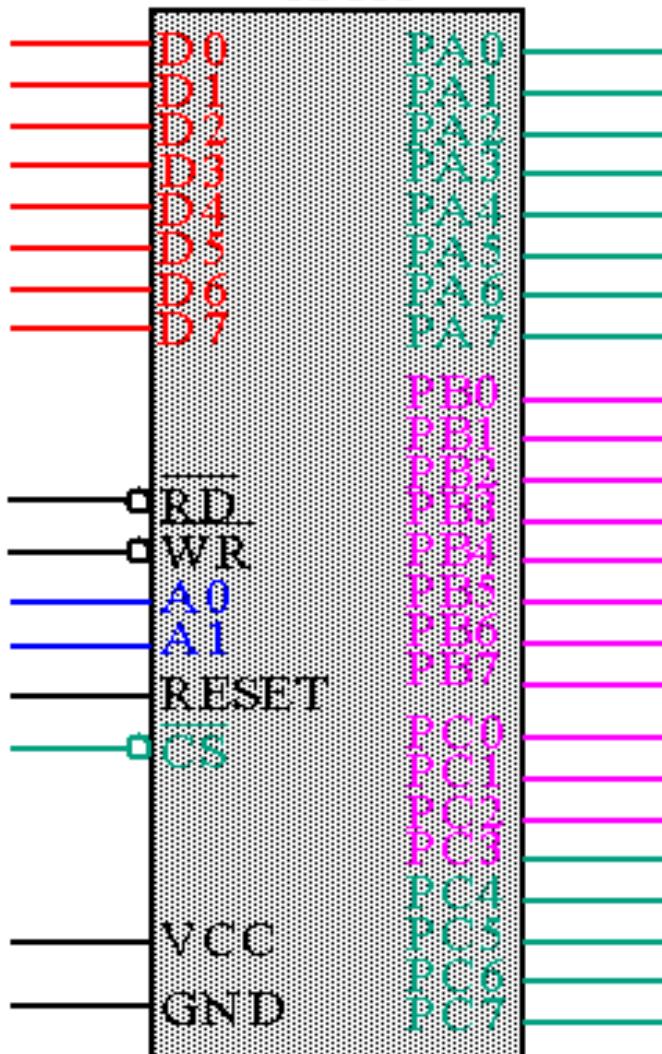
8255 Pin Diagram

PA3	1	40	PA4
PA2	2	39	PA5
PA1	3	38	PA6
PA0	4	37	PA7
<u>RD</u>	5	36	WR
<u>CS</u>	6	35	RESET
gnd	7	34	D0
A1	8	33	D1
A0	9	32	D2
PC7	10	8255	31
PC6	11	PPI	30
PC5	12		D5
PC4	13		D6
PC0	14		D7
PC1	15	26	Vcc
PC2	16	25	PB7
PC3	17	24	PB6
PB0	18	23	PB5
PB1	19	22	PB4
PB2	20	21	PB3

Internal Architecture of 8255



82C55



Group A

Port A (PA7-PA0) and upper half of port C (PC7 - PC4)

Group B

Port B (PB7-PB0) and lower half of port C (PC3 - PC0)

I/O Port Assignments

A ₁	A ₀	Function
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Command Register

- **Data bus(D₀-D₇):**These are 8-bit bi-directional buses, connected to 8085 data bus for transferring data.
- **CS:** This is Active Low signal. When it is low, then data is transfer from 8086.
- **Read:** This is Active Low signal, when it is Low read operation will be start.
- **Write:** This is Active Low signal, when it is Low Write operation will be start.

RESET: This is used to reset the device. That means clear control registers.

PA₀-PA₇:It is the 8-bit bi-directional I/O pins used to send the data to peripheral or to receive the data from peripheral.

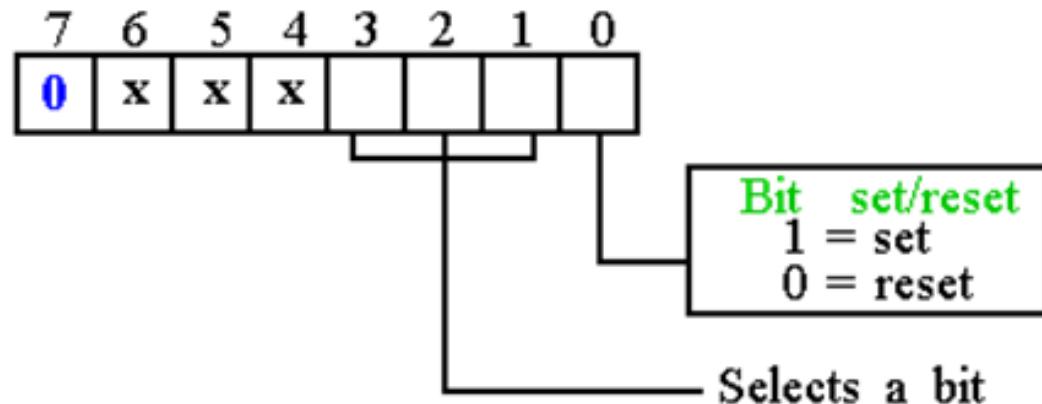
PB₀-PB₇:Similar to PA

PC₀-PC₇:This is also 8-bit bidirectional I/O pins. These lines are divided into two groups.

1. PC₀ to PC₃(Lower Groups)
2. PC₄ to PC₇ (Higher groups)

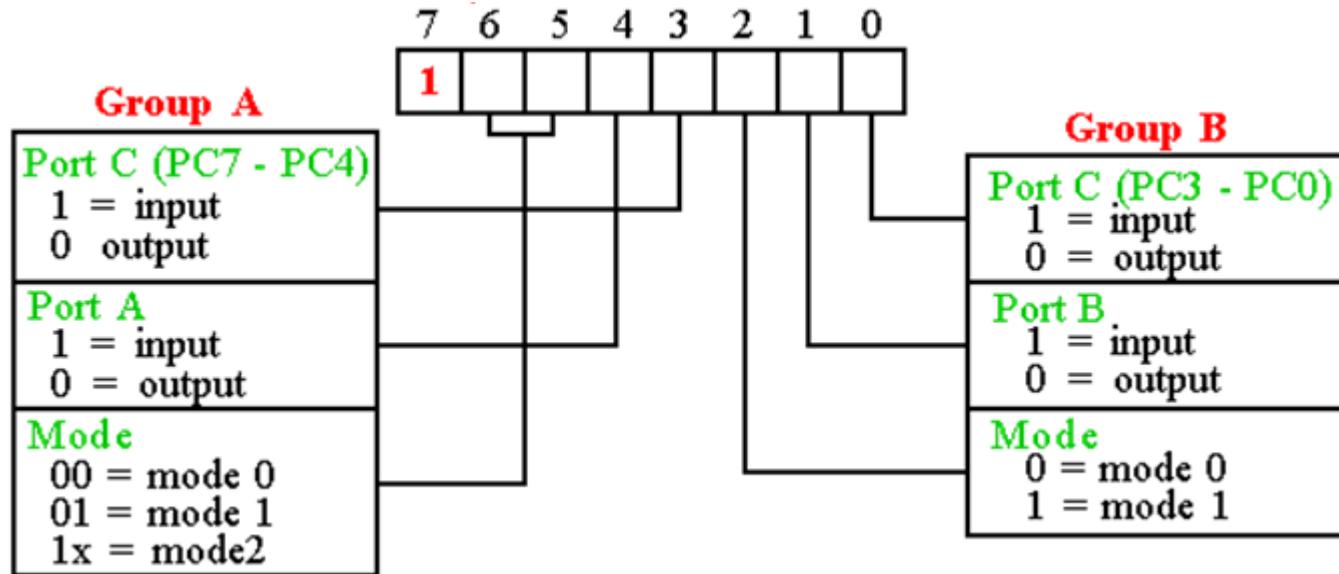
Modes of Operation of 8255

- BSR Mode
- IO modes: *mode 0, mode 1, and mode 2*

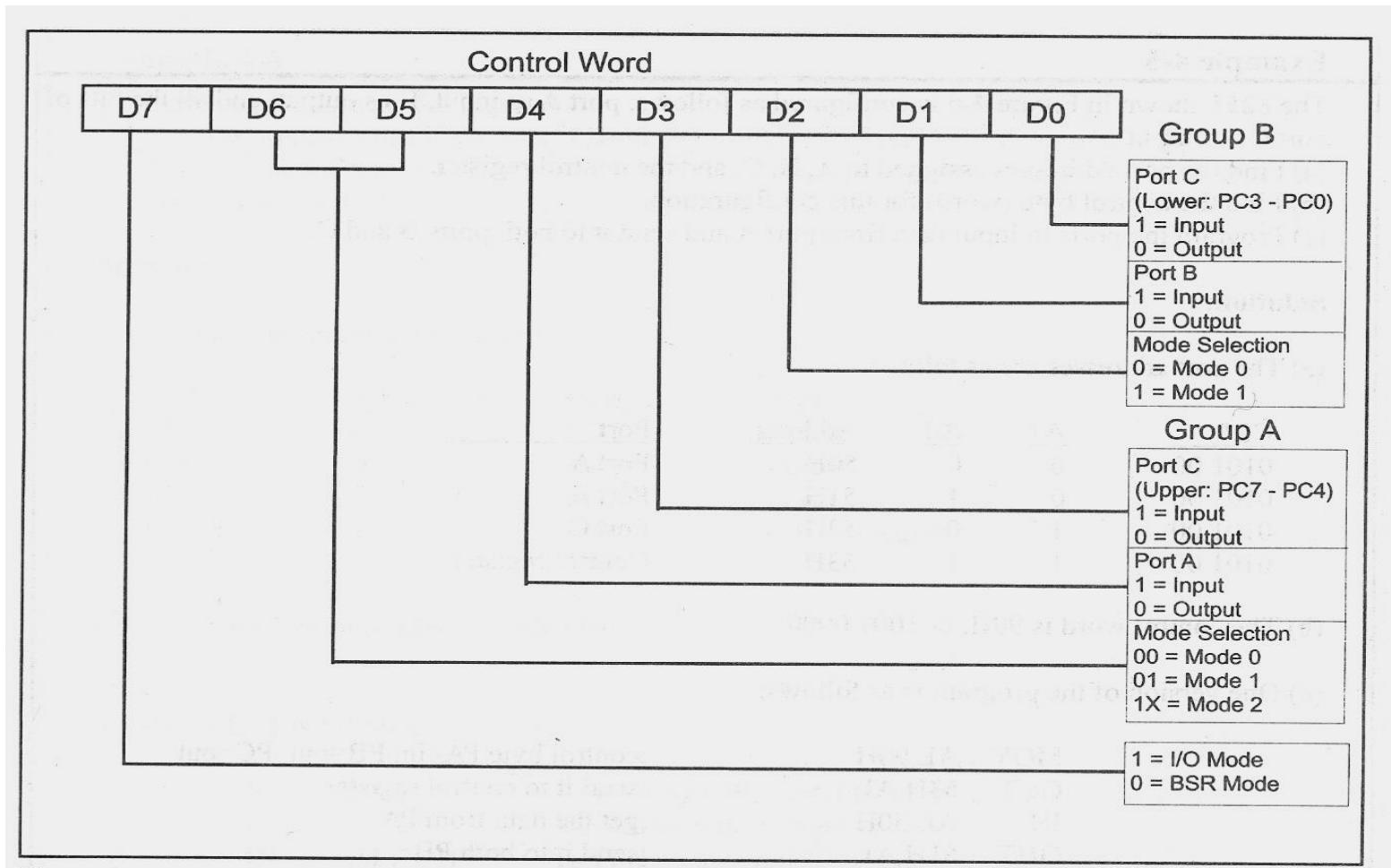


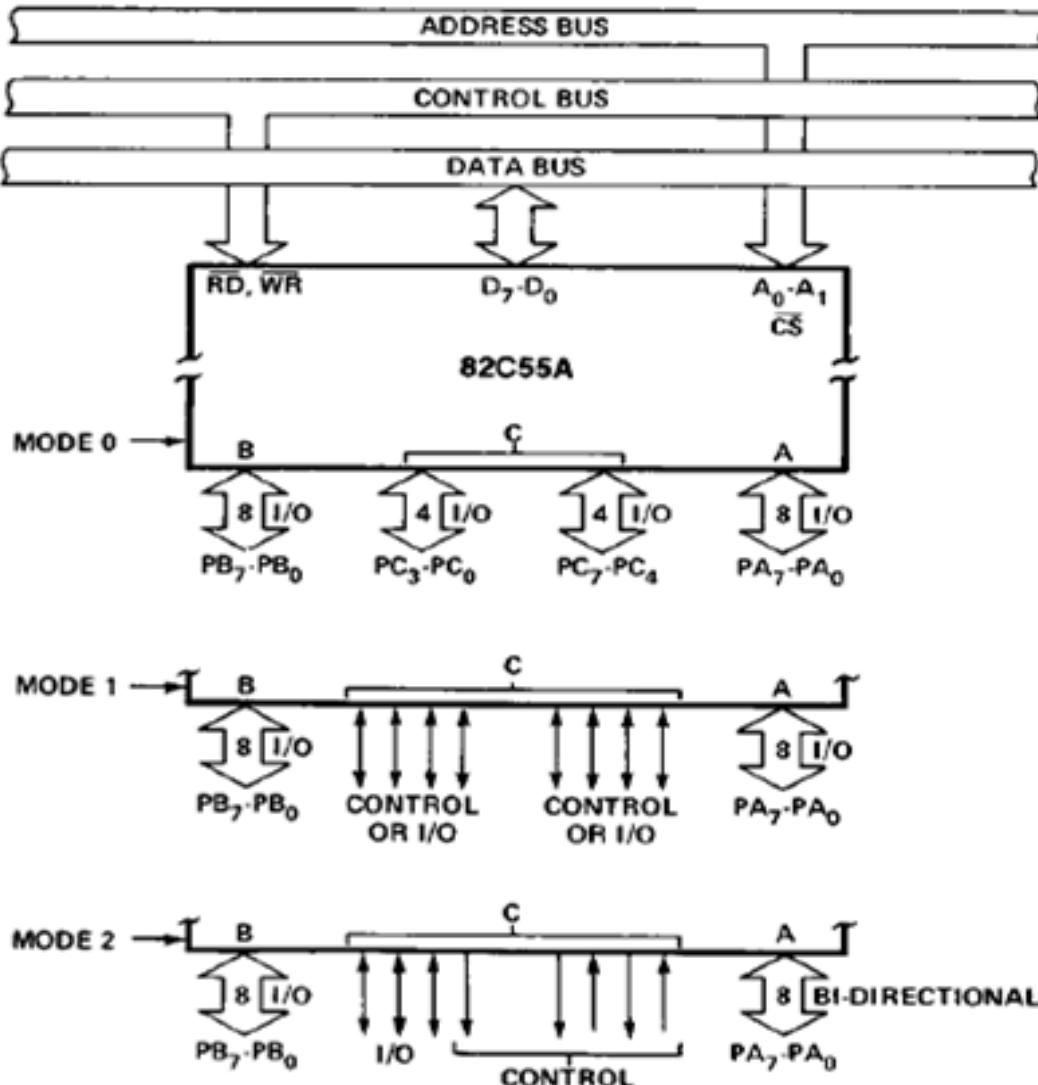
Control Word Register (CWR) format for BSR mode

- 8255 has three operation modes: *mode 0, mode 1, and mode 2*



Control Word Register (CWR) format for IO mode





- Mode 0
 - Basic I/O
- Mode 1
 - Strobe I/O
- Mode 2
 - Bi-Dir Bus

Mode 0: Simple Input or Output

In this mode, ports A, B are used as two simple 8-bit I/O ports
port C as two 4-bit ports.

Each port can be programmed to function as simply an input port or an output port. The input/output features in Mode 0 are as follows.

1. Outputs are latched.
2. Inputs are not latched.
3. Ports don't have handshake or interrupt capability.

Mode 1: Input or Output with Handshake

In this mode, handshake signals are exchanged between the MPU and peripherals prior to data transfer.

The features of the mode include the following:

1. Two ports (A and B) function as 8-bit I/O ports.
They can be configured as either as input or output ports.
2. Each port uses three lines from Port C as handshake signals.
 $\text{PC}_0 - \text{PC}_2$ for port B and $\text{PC}_3 - \text{PC}_5$ for port A.
The remaining two lines of Port C, PC_6 and PC_7 , can be used for simple I/O operations.
3. Input and Output data are latched.
4. Interrupt logic is supported.

Mode 2: Bidirectional Data Transfer

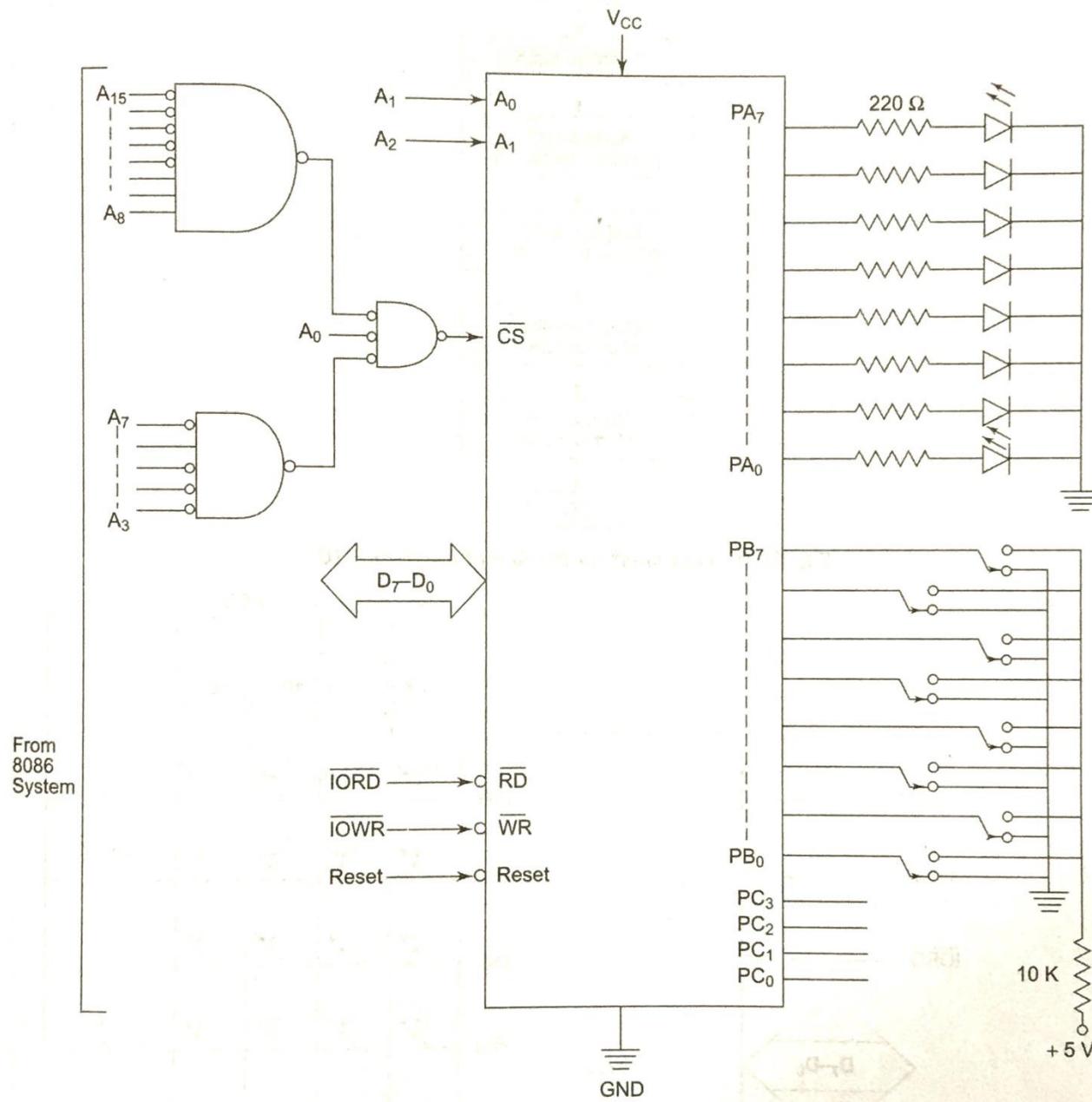
This mode is used primarily in applications such as data transfer between two computers.

In this mode, Port A can be configured as the bidirectional port. Port B either in Mode 0 or Mode 1.

Port A uses five signals from Port C, $\text{PC}_3\text{-}\text{PC}_7$, as handshake signals for data transfer.

The remaining three signals from port C, $\text{PC}_0\text{-}\text{PC}_2$ can be used either as simple I/O or as handshake for port B.

Interface an 8255 with 8086 to work as an I/O port. Initialize port A as output port, port B as input port and port C as output port. Port A address should be 0740H. Write a program to sense switch positions SW0-SW7 connected at port B. The sensed pattern is to be displayed on Port A, to which 8 LEDs are connected, while the port C lower displays number of ON switches out of the total eight switches.



Analog to Digital Converters

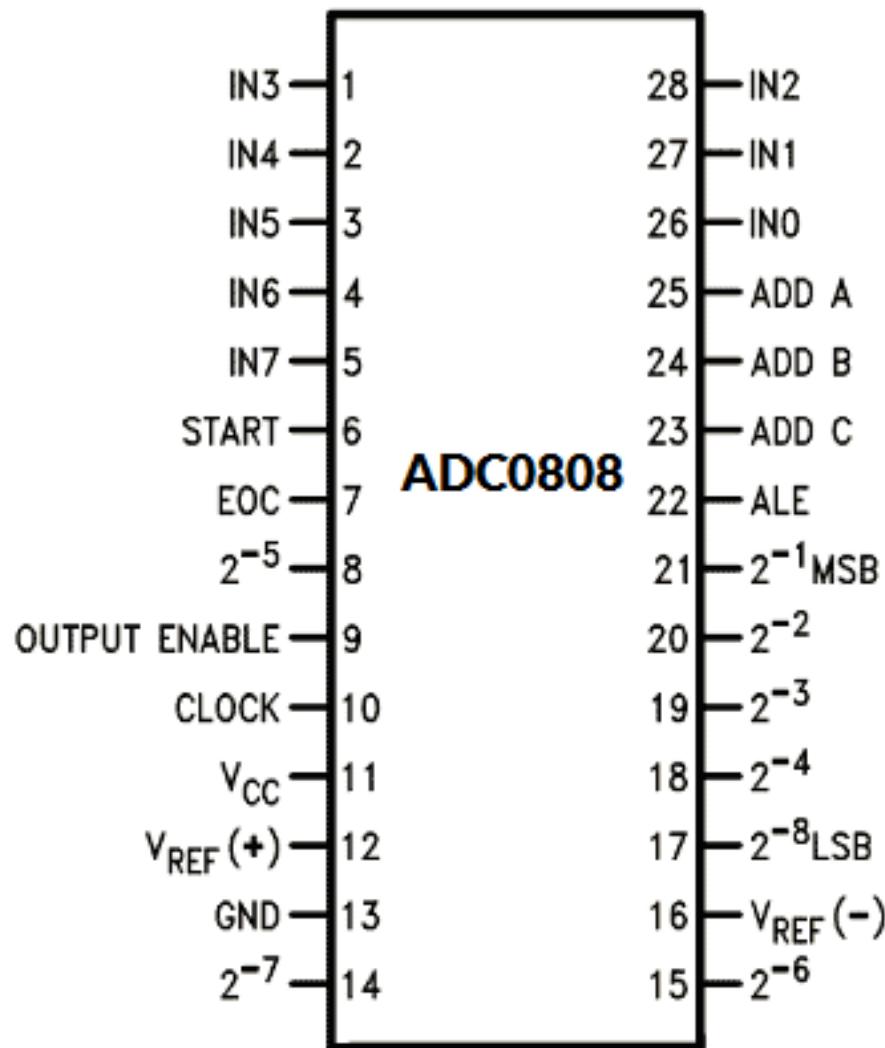
- In most of the cases, the 8255 PPI is used for interfacing the analog to digital converters with microprocessor.
- The analog to digital converters is treated as an input device by the microprocessor, that sends an initialising signal to the ADC to start the analogy to digital data conversation process. The start of conversation signal is a pulse of a specific duration.
- The process of analog to digital conversion is a slow process, and the microprocessor has to wait for the digital data till the conversion is over. After the conversion is over, the ADC sends end of conversion EOC signal to inform the microprocessor that the conversion is over and the result is ready at the output buffer of the ADC. These tasks of issuing an SOC pulse to ADC, reading EOC signal from the ADC and reading the digital output of the ADC are carried out by the CPU using 8255 I/O ports.

- The time taken by the ADC from the active edge of SOC pulse till the active edge of EOC signal is called as the conversion delay of the ADC.
- It may range anywhere from a few microseconds in case of fast ADC to even a few hundred milliseconds in case of slow ADCs.
- The available ADC in the market use different conversion techniques for conversion of analog signal to digital. Successive approximation techniques and dual slope integration techniques are the most popular techniques used in the integrated ADC chip.
- General algorithm for ADC interfacing contains the following steps:
 - Ensure the stability of analog input, applied to the ADC.
 - Issue start of conversion pulse to ADC
 - Read end of conversion signal to mark the end of conversion processes.
 - Read digital data output of the ADC as equivalent digital output.

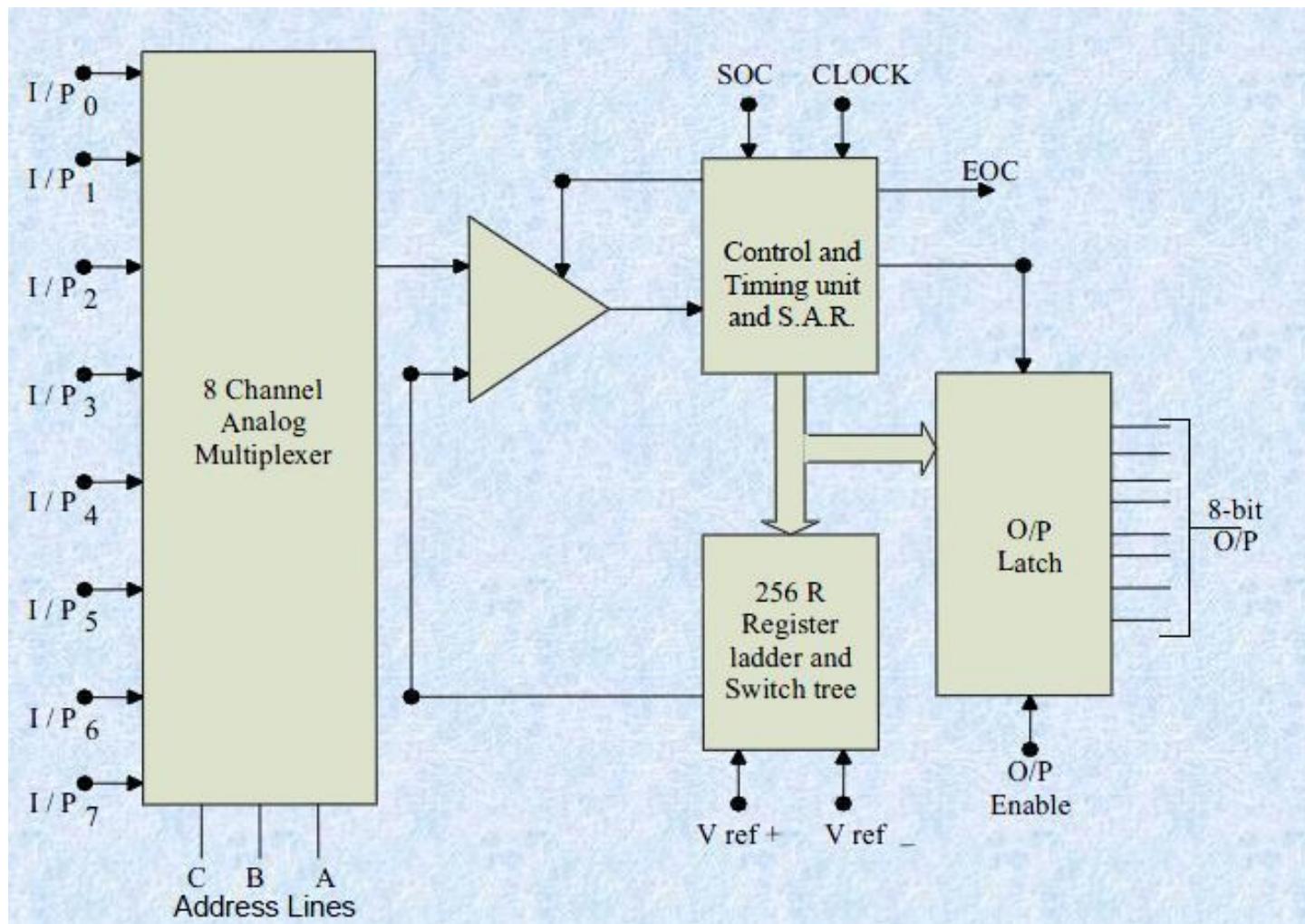
ADC 0808/0809

- The analog to digital converter chips 0808 and 0809 are 8- bit CMOS, successive approximation converters.
- This technique is one of the fast techniques for analog to digital conversion.
- The conversion delay is $100\mu s$ at a clock frequency of 640 KHz, which is quite low as compared to other converters.
- These converters internally have a 3:8 analog multiplexer so that at a time eight different analog conversion by using address lines -ADD A, ADD B, ADD C. Using these address inputs, multichannel data acquisition system can be designed using a single ADC. The CPU may drive these lines using output port lines in case of multichannel applications. In case of single input applications, these may be hardwired to select the proper input.
- There are unipolar analog to digital converters, i.e. they are able to convert only positive analog input voltage to their digital equivalent.
- These chips do no contain any internal sample and hold circuit.

ADC 0808/0809 Pin Diagram

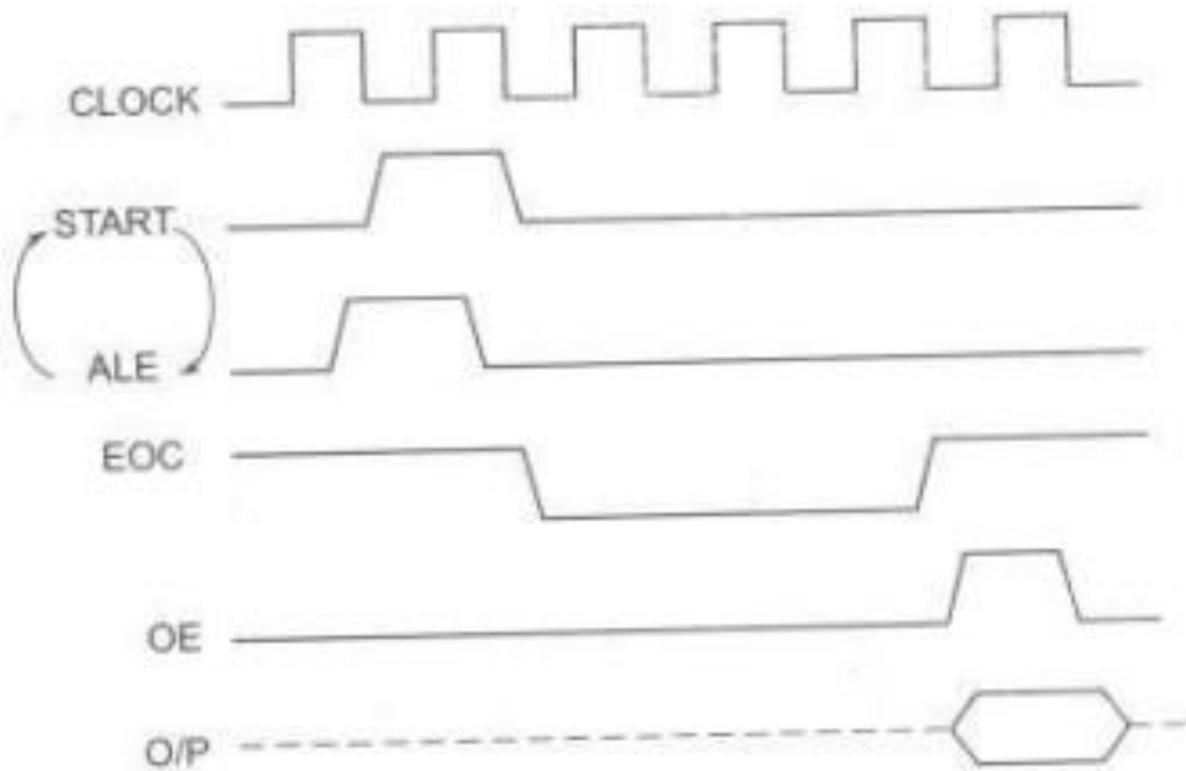


Block Diagram of ADC 0808/0809

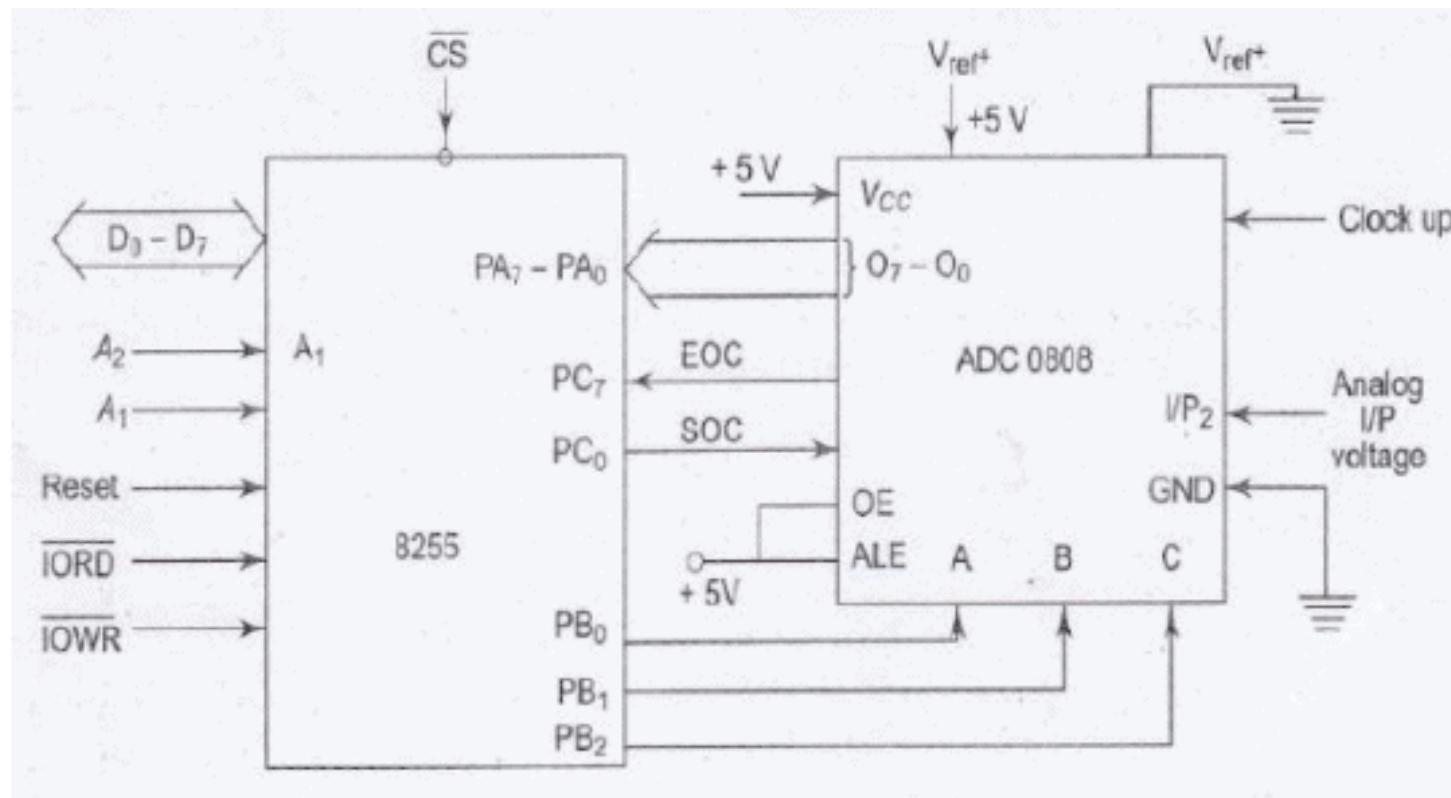


Address lines			Input Selected
C	B	A	
0	0	0	Input 0
0	0	1	Input 1
0	1	0	Input 2
0	1	1	Input 3
1	0	0	Input 4
1	0	1	Input 5
1	1	0	Input 6
1	1	1	Input 7

- **V_{cc}** Supply pins +5V
- **GND** GND
- **V_{ref+}** Reference voltage positive +5 Volts maximum.
- **V_{ref-}** Reference voltage negative 0Volts minimum.
- **I/P₀ - I/P₇** Analog inputs
- **ADD A,B,C** Address lines for selecting analog inputs.
- **O₇ - O₀** Digital 8-bit output with O₇ MSB and O₀ LSB
- **SOC** Start of conversion signal pin
- **EOC** End of conversion signal pin
- **OE** Output latch enable pin, if high enables output
- **CLK** Clock input for ADC



Interface ADC0808 with 8086 using 8255 ports. Use Port A of 8255 for transferring digital output of ADC to the CPU and Port C for control signals. Assume that an analog input is present at input2 of the ADC and a clock input of suitable frequency is available for ADC. Draw the schematic and write required assembly language program.



**MOV AL, 98h
OUT CWR, AL**

**MOV AL, 02h
OUT Port B, AL**

**MOV AL, 00h
OUT Port C, AL
MOV AL, 01h
OUT Port C, AL
MOV AL, 00h
OUT Port C, AL**

**WAIT: IN AL, Port C
RCL AL,01
JNC WAIT**

**IN AL, Port A
HLT**

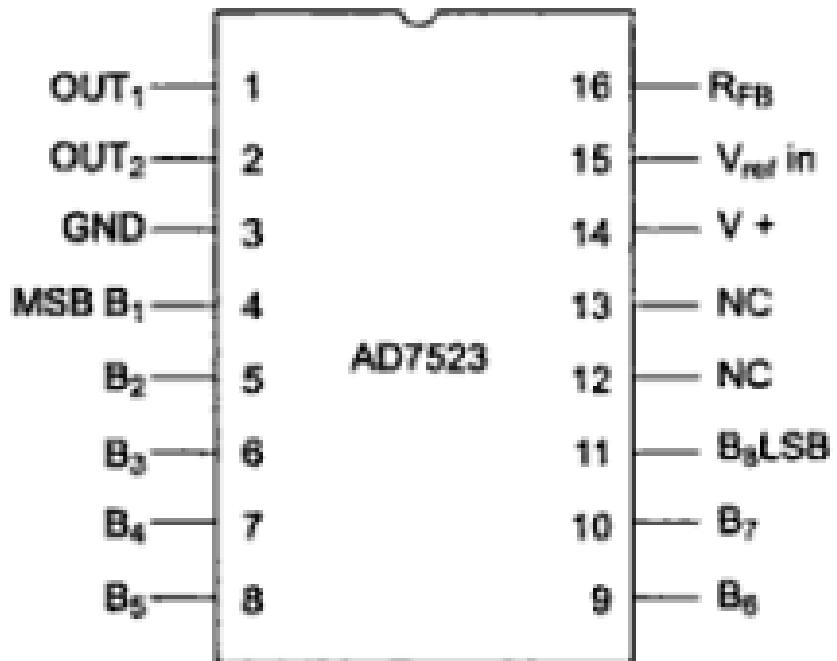
Dr. Venkata Reddy Kota @
JNTUK, Kakinada

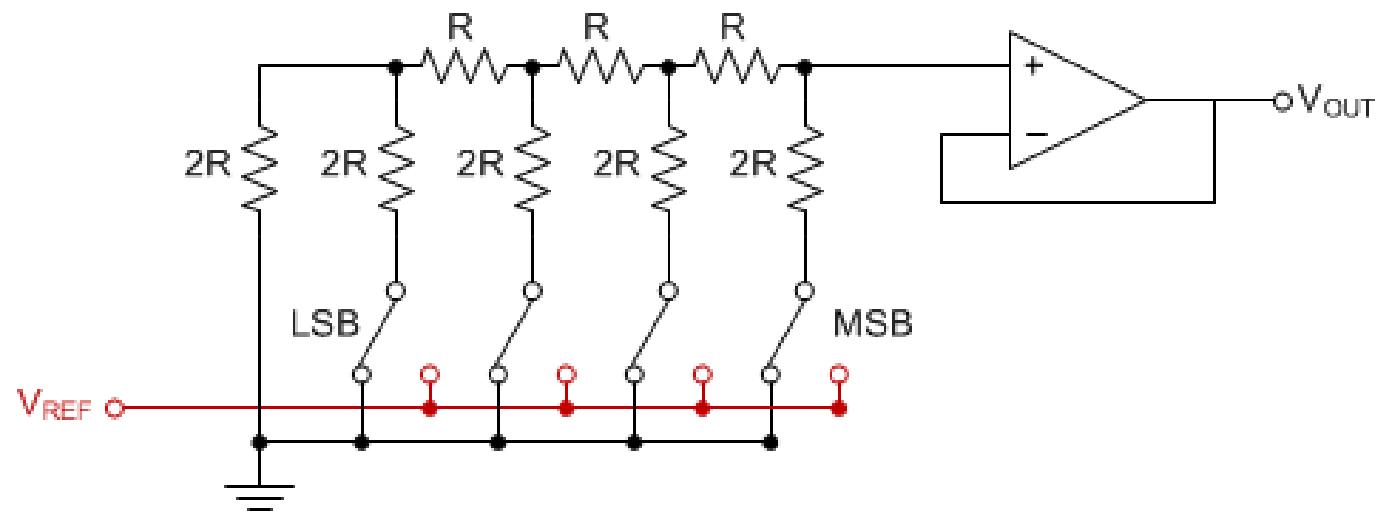
Digital to Analog Converters

- Digital to analog converters convert binary numbers into their analog equivalent voltages.
- The DACs find applications in areas like digitally controlled gains, motor speed control, programmable gain amplifiers, etc.

AD7523

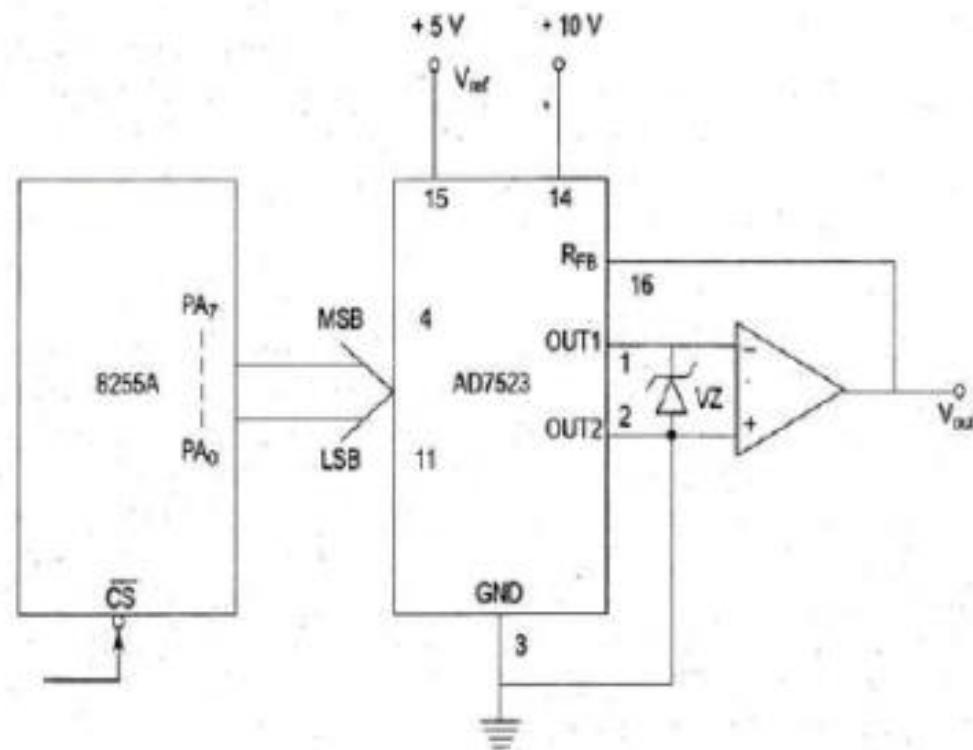
- Intersil's AD7523 is a 16 pin chip with multiplying digital to analog converter, containing R-2R ladder ($R=10K$).
- The supply range extends from +5V to +15V, while V_{ref} may be any where between -10V to +10V.
- The maximum analog output voltage will be +10V.
- An operation amplifier is used as a current-to-voltage converter at the output of AD7523.





Interface DAC AD7523 with an 8086 CPU running at 8 MHz and write an assembly language program to generate a sawtooth waveform of period 1ms with Vmax 5V.

CWR = ?



**ASSUME CS:CODE
CODE SEGMENT**

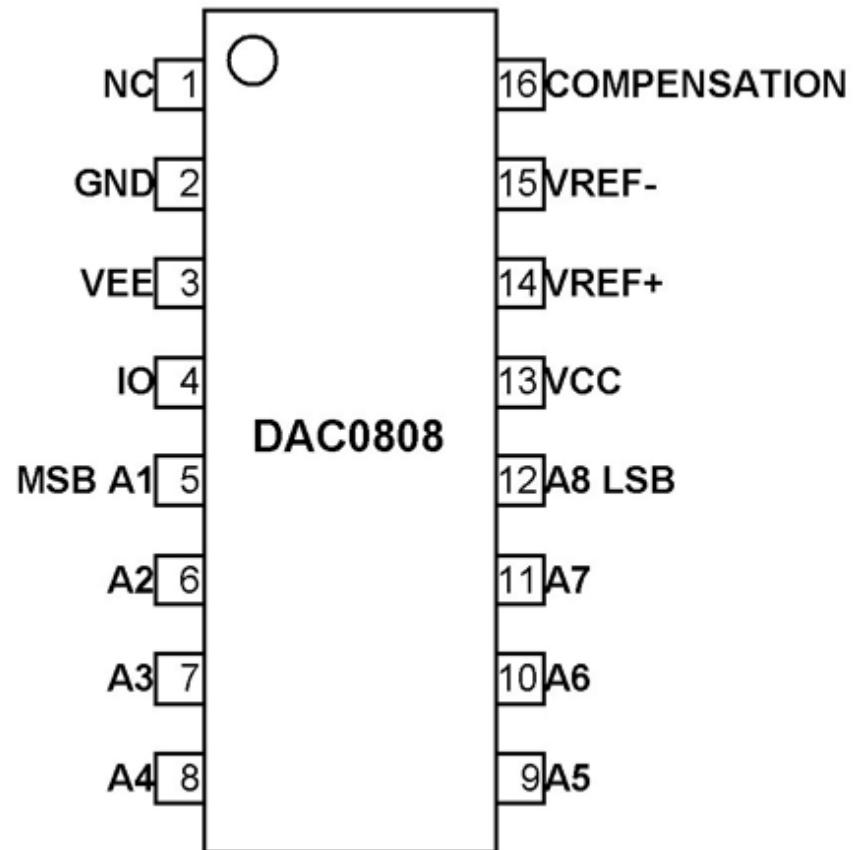
**MOV AL, 80H
OUT CWR, AL**

**AGAIN: MOV AL, 00H
BACK: OUT PORT A, AL
INC AL
CMP AL, 0F2H
JB BACK
JMP AGAIN**

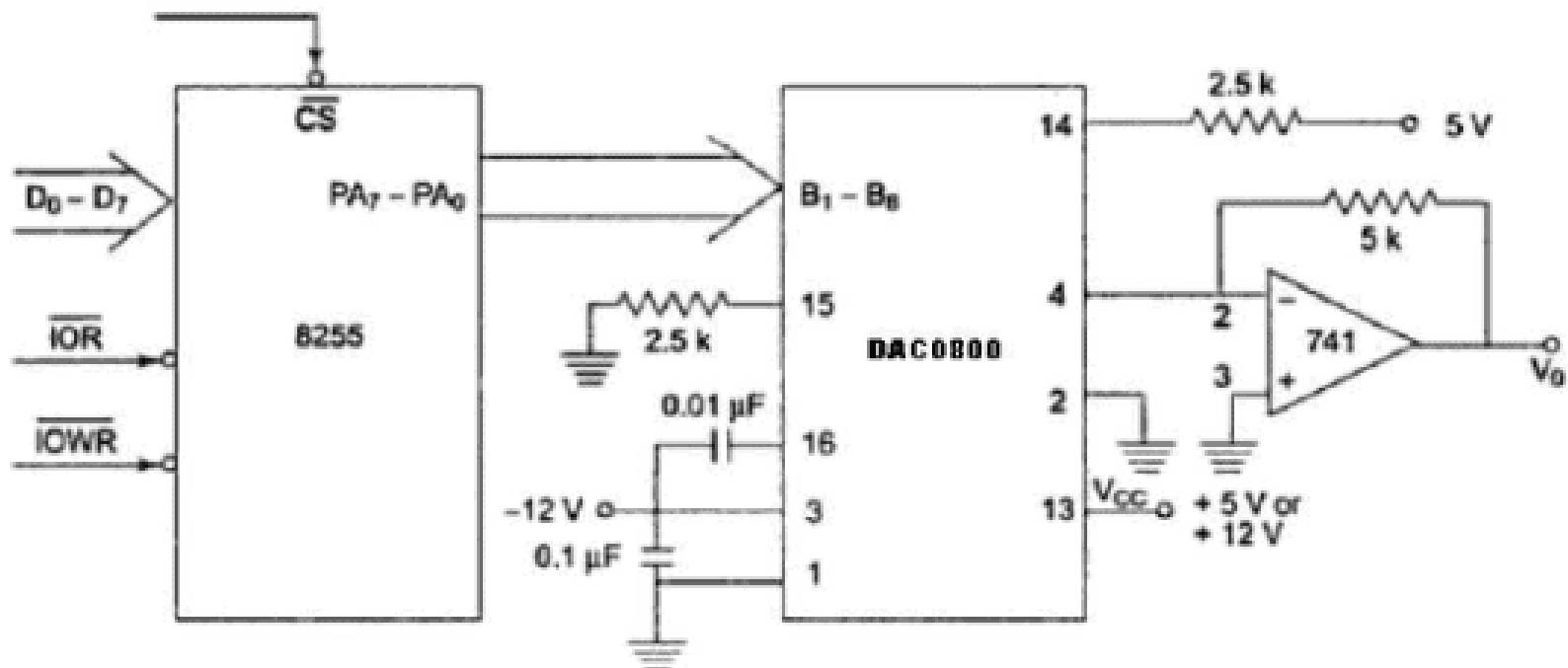
**CODE ENDS
END**

DAC0800

- The DAC 0800 is a monolithic 8-bit DAC manufactured by National Semiconductor.
- It has a settling time around 100ms.
- It can operate from 4.5V to +18V.
- The supply V+ is 5V or 12V and V- pin is kept at -12V.



Write an assembly language program to generate a triangular wave of frequency 500Hz using the interfacing circuit. The 8086 system operates at 8MHz. The amplitude of triangular wave should be +5V.



**ASSUME CS:CODE
CODE SEGMENT**

**MOV AL, 80H
OUT CWR, AL**

RISE: **MOV AL, 00H
OUT PORT A, AL
INC AL
CMP AL, 0FFH
JB RISE**

FALL: **OUT PORT A, AL
DEC AL
CMP AL, 00H
JA FALL**

JMP RISE

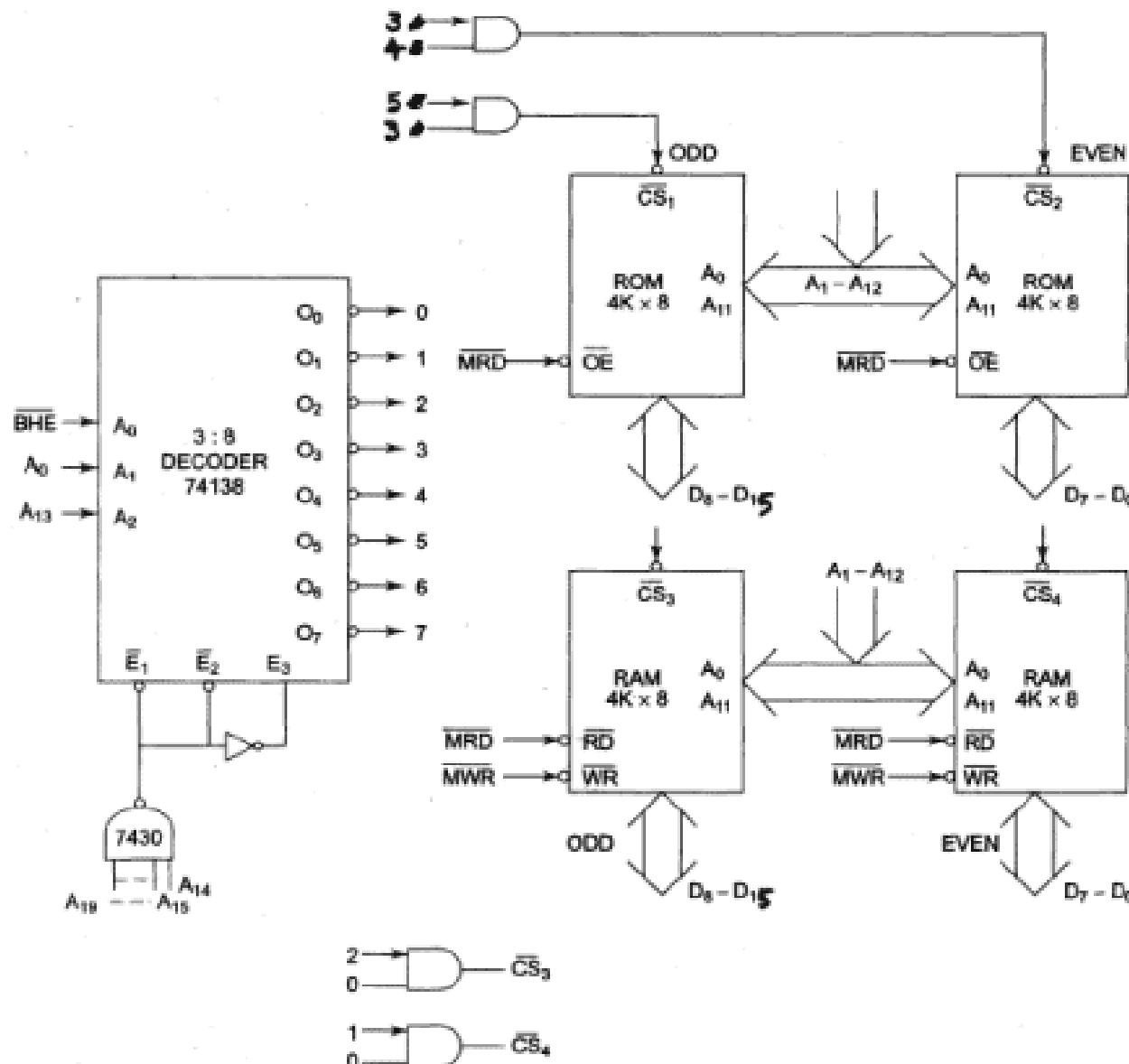
**CODE ENDS
END**

Dr. Venkata Reddy Kota @
JNTUK, Kakinada

Static Memory Interfacing with 8086

Q. Interface two 4K x 8 EPROMs and two 4K x 8 RAM chips with 8086 microprocessor. Select suitable mapping.

Address	A ₁₉	A ₁₈	A ₁₇	A ₁₆	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₀₉	A ₀₈	A ₀₇	A ₀₆	A ₀₅	A ₀₄	A ₀₃	A ₀₂	A ₀₁	A ₀₀	
FFFFFH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
EPROM											8K x 8										
FE000H	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FDFFFFH	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
RAM											8K x 8										
FC000H	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



A2 (A13)	A1 (A0)	A0 (BHE*)	RAM / EPROM	Chip Selected	Decoder Output
0	0	0	RAM	3 and 4	O ₀
0	0	1	RAM	4	O ₁
0	1	0	RAM	3	O ₂
0	1	1	RAM	--	O ₃
1	0	0	EPROM	1 and 2	O ₄
1	0	1	EPROM	2	O ₅
1	1	0	EPROM	1	O ₆
1	1	1	EPROM	--	O ₇

8259A Programmable Interrupt Controllers

Interrupts

- Definition
 - An event external to the currently executing process that causes a change in the normal flow of instruction execution; usually generated by hardware devices external to the CPU.
 - Key point is that interrupts are asynchronous w.r.t. current process
 - Typically indicate that some device needs service

Why interrupts?

- MCUs have many external peripherals
 - Keyboard, mouse, screen, disk drives, scanner, printer, sound card, camera, etc.
 - These devices occasionally need CPU service
 - But we can't predict when
 - We want to keep the CPU busy (or asleep) between events
 - Need a way for CPU to find out devices need attention

Possible Solution: Polling

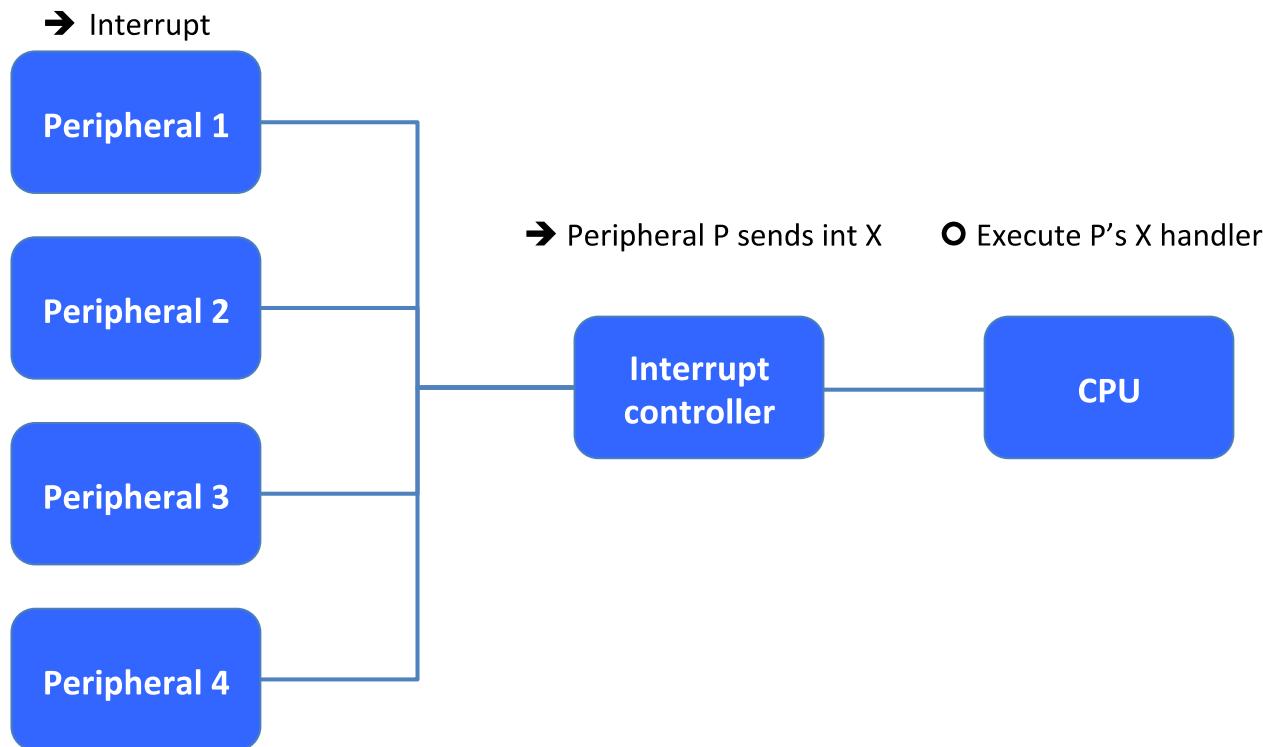
- CPU periodically checks each device to see if it needs service
 - “Polling is like picking up your phone every few seconds to see if you have a call.”

Possible Solution: Polling

- CPU periodically checks each device to see if it needs service
 - “Polling is like picking up your phone every few seconds to see if you have a call.”
 - Cons: takes CPU time even when no requests pending
 - Pros: can be efficient if events arrive rapidly

Alternative: Interrupts

- Give each device a wire (interrupt line) that it can use to signal the processor

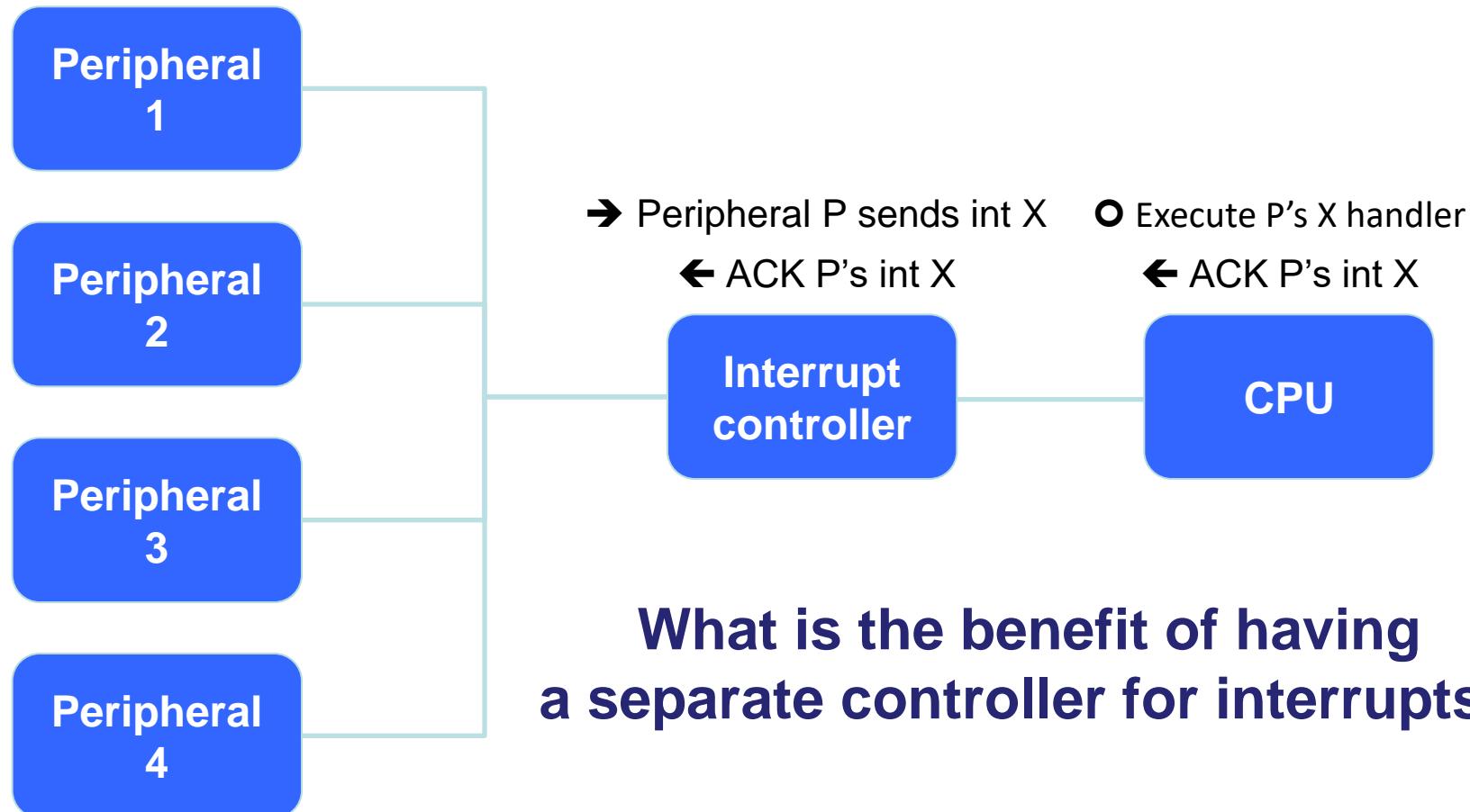


Alternative: Interrupts

- Give each device a wire (interrupt line) that it can use to signal the processor
 - When interrupt signaled, processor executes a routine called an interrupt handler to deal with the interrupt
 - No overhead when no requests pending

How do interrupts work?

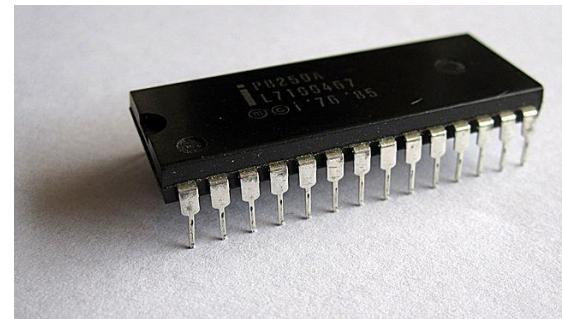
- ➔ Interrupt
- Clear interrupt



The Interrupt controller

- **Handles simultaneous interrupts**
 - Receives interrupts while the CPU handles interrupts
- **Maintains interrupt flags**
 - CPU can poll interrupt flags instead of jumping to a interrupt handler
- **Multiplexes many wires to few wires**
 - CPU doesn't need a interrupt wire to each peripheral

Fun fact: Interrupt controllers used to be separate chips!



Intel 8259A IRQ chip

Image by Nixdorf - Own work

How to use interrupts

1. Tell the peripheral which interrupts you want it to output.
2. Tell the interrupt controller what your priority is for this interrupt.
3. Tell the processor where the interrupt handler is for that interrupt.
4. When the interrupt handler fires, do your business then clear the int.

CPU execution of interrupt handlers

INTERRUPT

1. Wait for instruction to end
2. Push the program counter to the stack
3. Push all active registers to the stack
4. Jump to the interrupt handler in the
interrupt vector
5. Pop the program counter off of the
stack

Intel's “reserved” interrupts

- Intel had reserved interrupt-numbers 0-31 for the processor's various exceptions
- But only interrupts 0-4 were used by 8086
- Designers of the early IBM-PC ROM-BIOS disregarded the “Intel reserved” warning
- So interrupts 5-31 got used by ROM-BIOS code for its own various purposes
- This created interrupt-conflicts for 80286+

Exceptions in Protected-Mode

- The interrupt-conflicts seldom arise while the processor is executing in Real-Mode
- PC BIOS uses interrupts 8-15 for devices (such as timer, keyboard, printers, serial communication ports, and diskette drives)
- CPU uses this range of interrupt-numbers for various processor exceptions (such as page-faults, stack-faults, protection-faults)

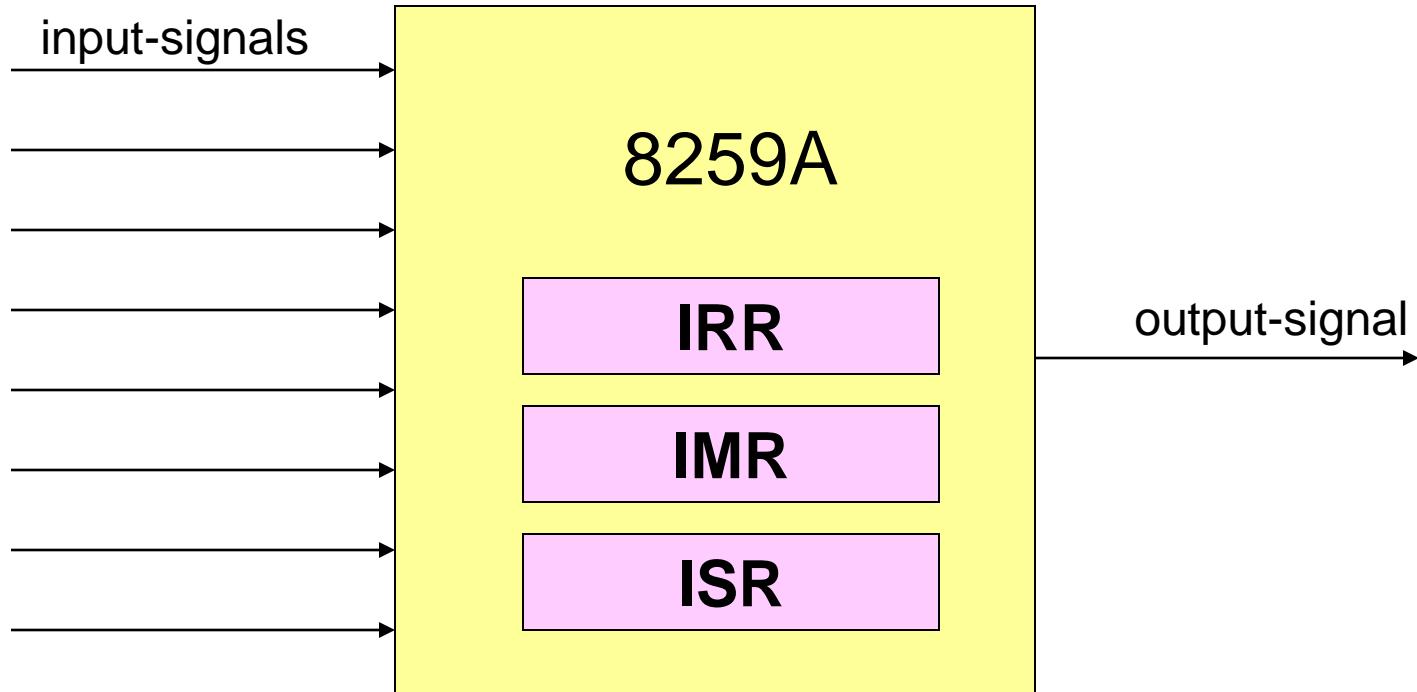
Handling these conflicts

- There are two ways we can ‘resolve’ these interrupt-conflicts when we write ‘handlers’ for device-interrupts in the ‘overlap’ range
 - We can design each ISR to query the system in some way, to determine the ‘cause’ for the interrupt-condition (i.e., a device or the CPU?)
 - We can ‘reprogram’ the Interrupt Controllers to use non-conflicting interrupt-numbers when the peripheral devices trigger their interrupts

Learning to program the 8259A

- Either solution will require us to study how the system's two Programmable Interrupt Controllers are programmed
- Of the two potential solutions, it is evident that greater system efficiency will result if we avoid complicating our interrupt service routines with any “extra overhead” (i.e., to see which component wished to interrupt)

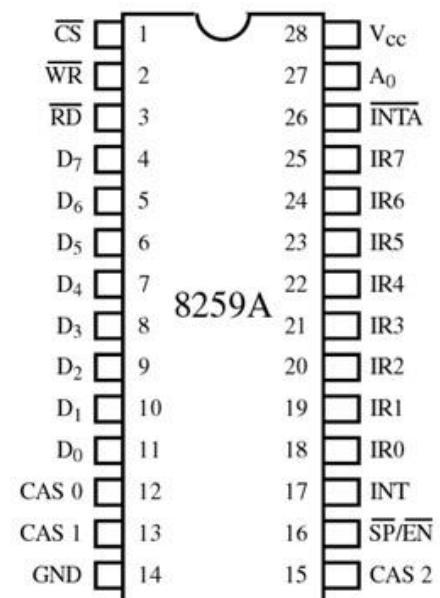
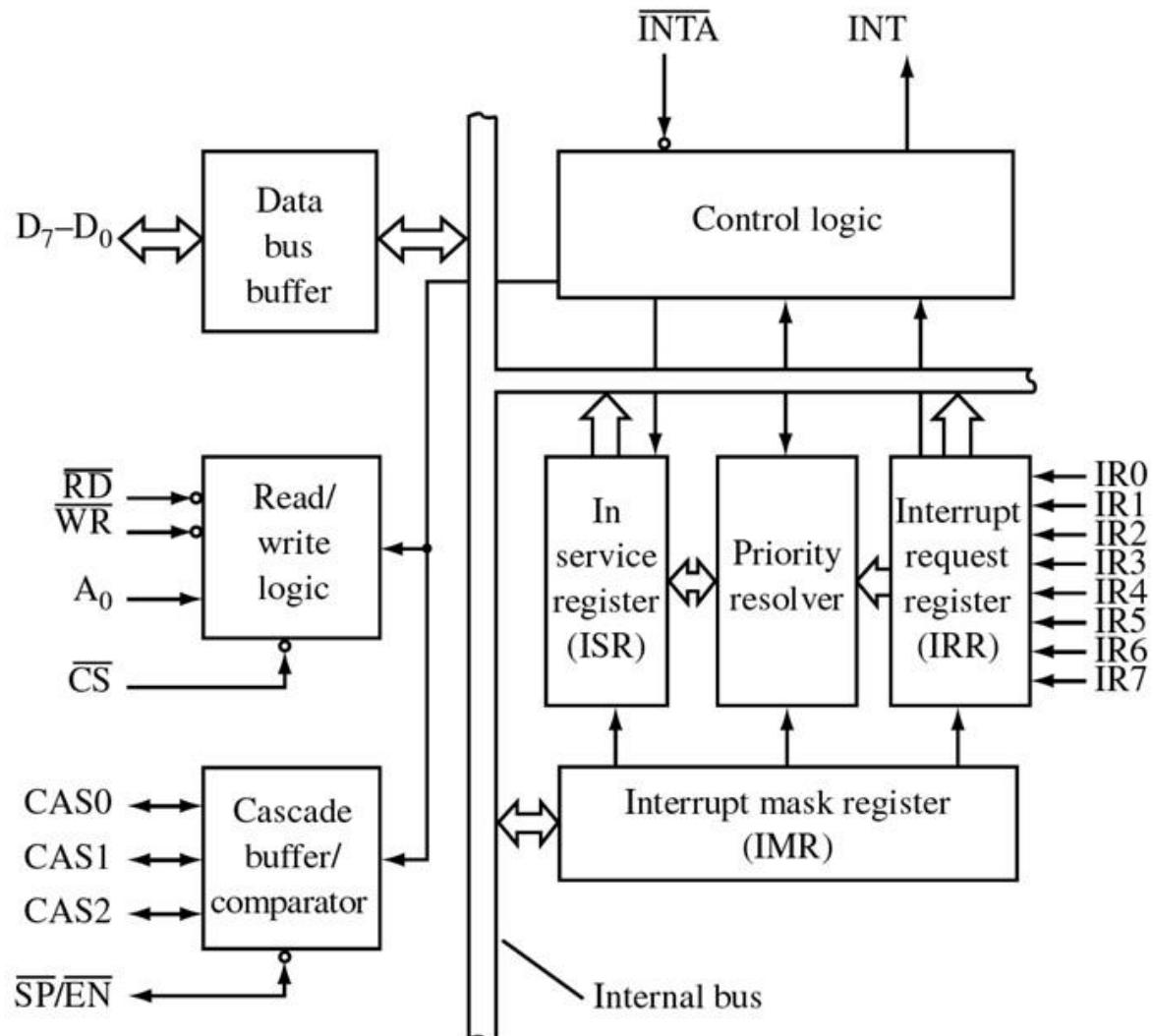
Three internal registers



IRR = Interrupt Request Register

IMR = Interrupt Mask Register

ISR = In-Service Register



Interrupt Request Register (IRR) and In-Service Register (ISR)

The interrupts at the IR input lines are handled by two registers in cascade, the Interrupt Request Register (IRR) and the In- Service Register (ISR). The IRR is used to indicate all the interrupt levels which are requesting service, and the ISR is used to store all the interrupt levels which are currently being serviced .

Priority Resolver

This logic block determines the priorities of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during the INTA sequence.

Interrupt Mask Register (IMR)

The IMR stores the bits which disable the interrupt lines to be masked. The IMR operates on the output of the IRR. Masking of a higher priority input will not affect the interrupt request lines of lower priority.

Data Bus Buffer

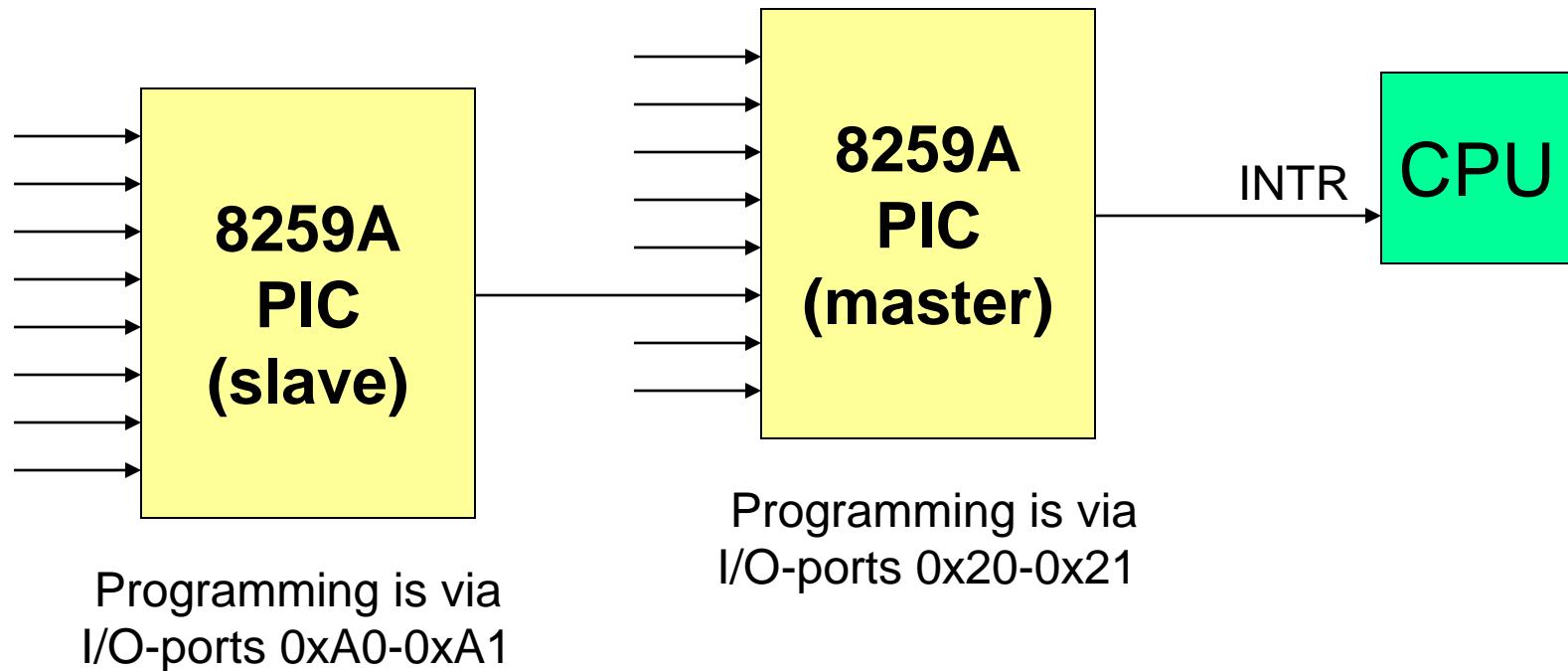
This 3-state, bidirectional 8-bit buffer is used to interface the PIC to the System Data Bus. Control words and status information are transferred through the Data Bus Buffer.

- ▶ **Read/Write Control Logic**
- ▶ The function of this block is to accept output commands from the CPU. It contains the Initialization Command Word (ICW) registers and Operation Command Word (OCW) registers which store the various control formats for device operation. This function block also allows the status of the PIC to be transferred onto the Data Bus. This function block stores and compares the IDs of all PICs used in the system. The associated three I/O pins (CAS0- 2) are outputs when the 8259 is used as a master and are inputs when the 8259 is used as a slave. As a master, the 8259 sends the ID of the interrupting slave device onto the CAS0 - 2 lines. The slave, thus selected will send its preprogrammed subroutine address onto the Data Bus during the next one or two consecutive INTA pulses.

D[7..0] These wires are connected to the system bus and are used by the microprocessor to write or read the internal registers of the 8259.

A[0..0] This pin acts in conjunction with WR/RD signals. It is used by the 8259 to decipher various command words the microprocessor writes and status the microprocessor wishes to read.

PC System Design



Dr. Venkata Reddy Kota @
JNTUK, Kakinada

DIRECT MEMORY ACCESS

- For the applications that require huge amounts of data transfer to memory from a magnetic or optical disk a dedicated hardware device called DMA controller is used.
- The DMA controller temporarily borrows the address bus ,data bus and control bus from the microprocessor and transfers the data bytes directly from the disk controller to a series of memory locations.
- Some DMA devices even can do memory to memory transfers.

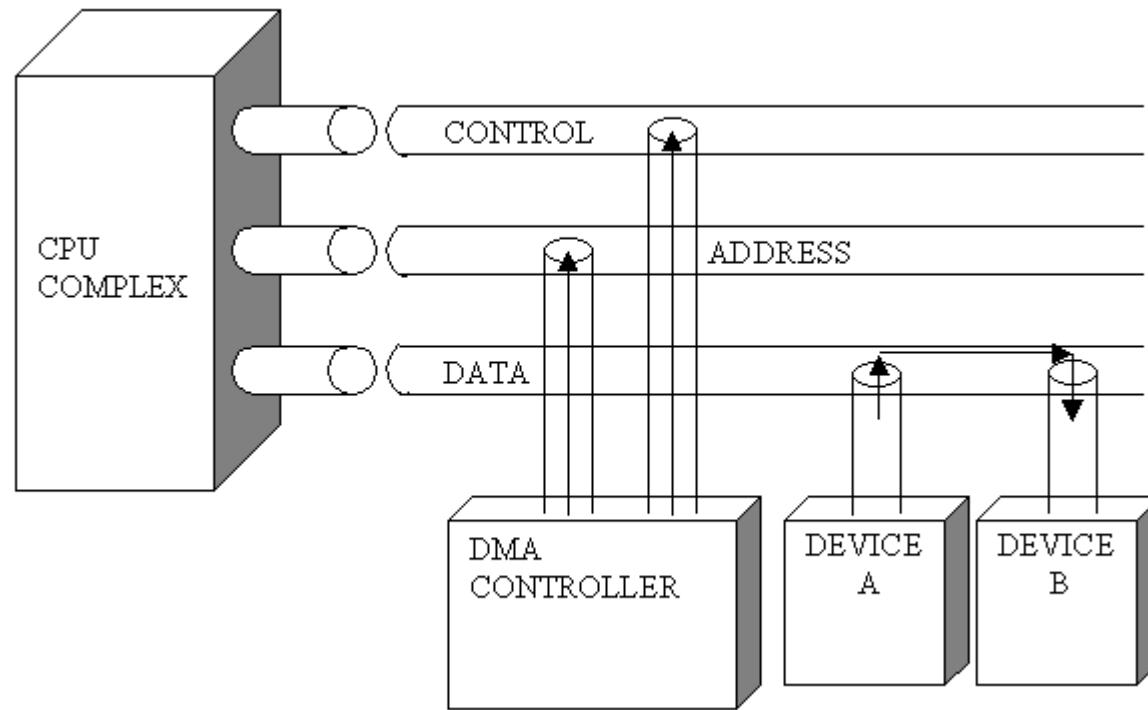


Fig. Block diagram of DMA Interfacing

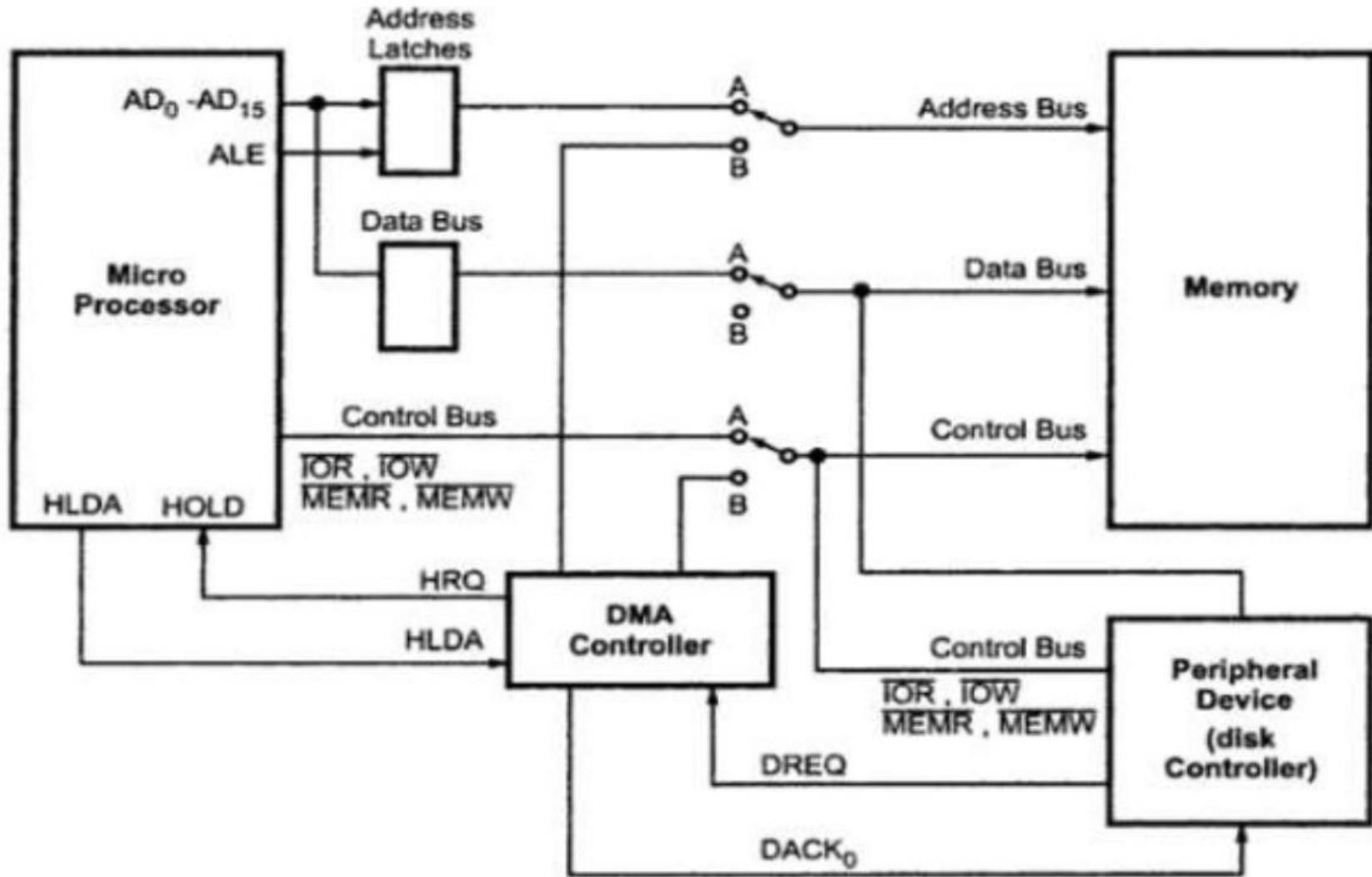


Fig. Block diagram of DMA data transfer scheme

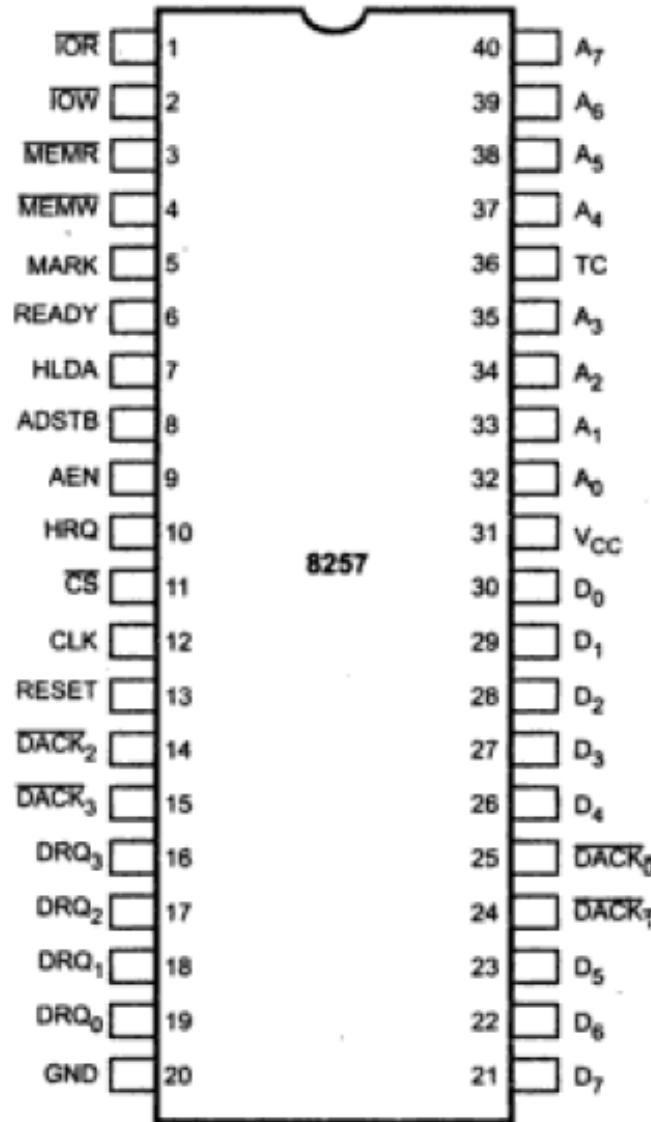


Fig. Pin diagram of 8257 DMA controller

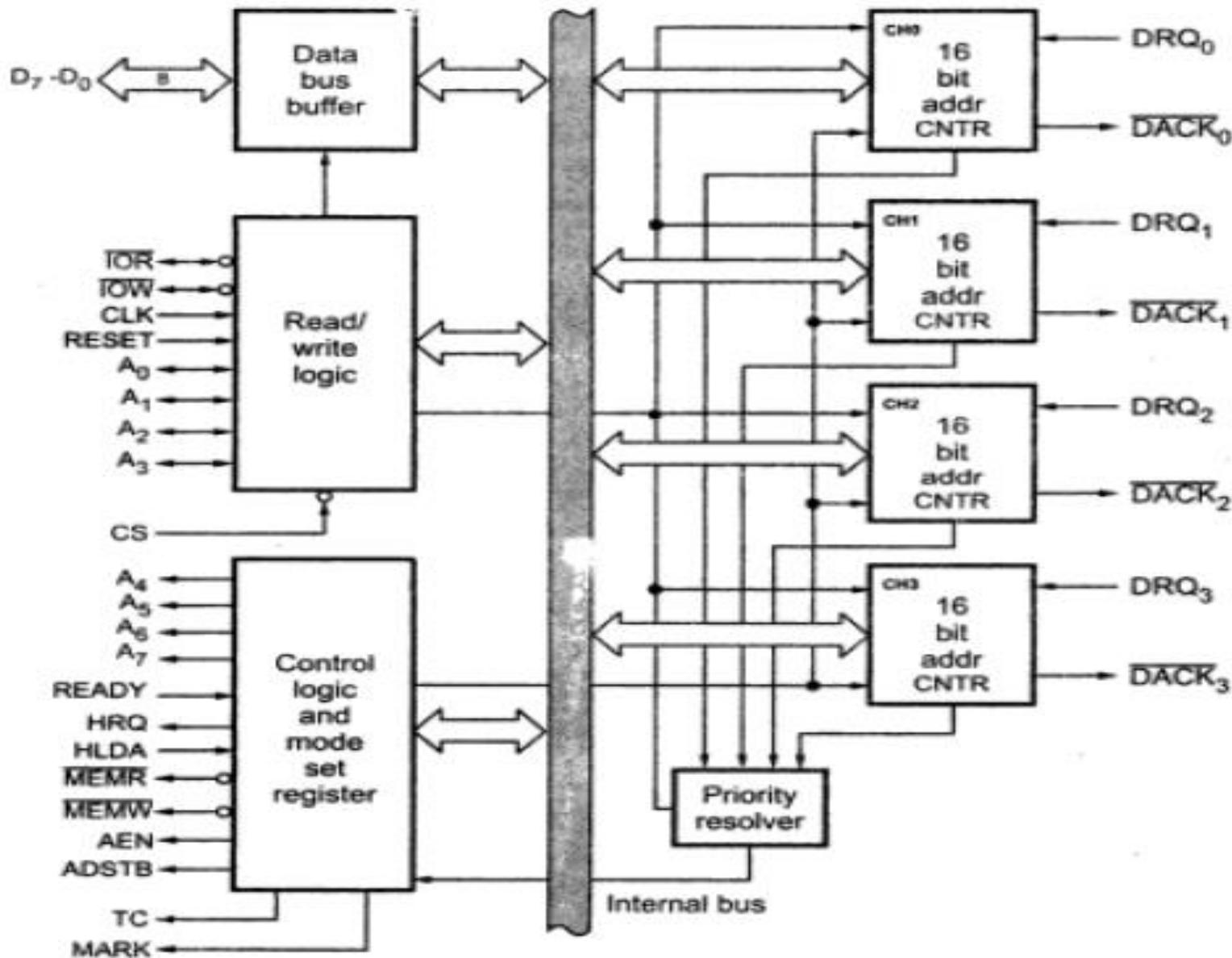
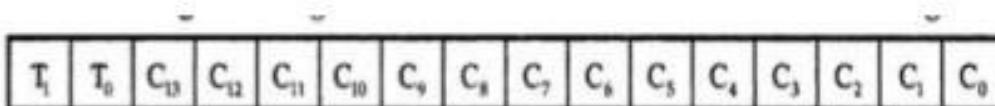


Fig. Functional block diagram of 8257

Format of Terminal Count Register

- The most significant two bits of the terminal count register specify the type of DMA operation for that channel.



T ₁	T ₀	Type of operation
0	0	DMA verify cycle
0	1	DMA Write cycle
1	0	DMA READ cycle
1	1	Illegal

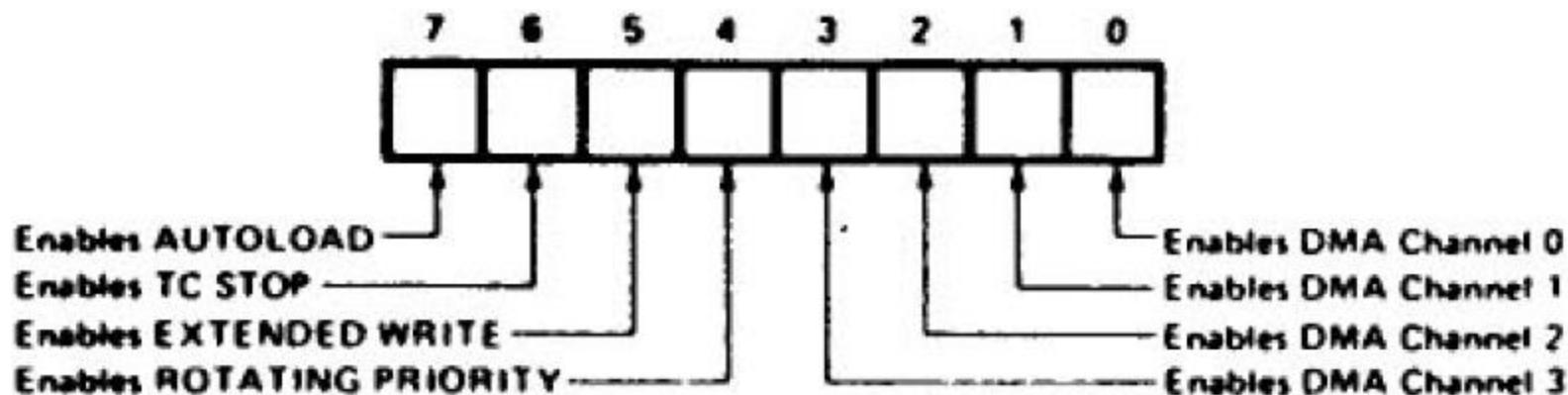


Fig. Mode set register Bit format

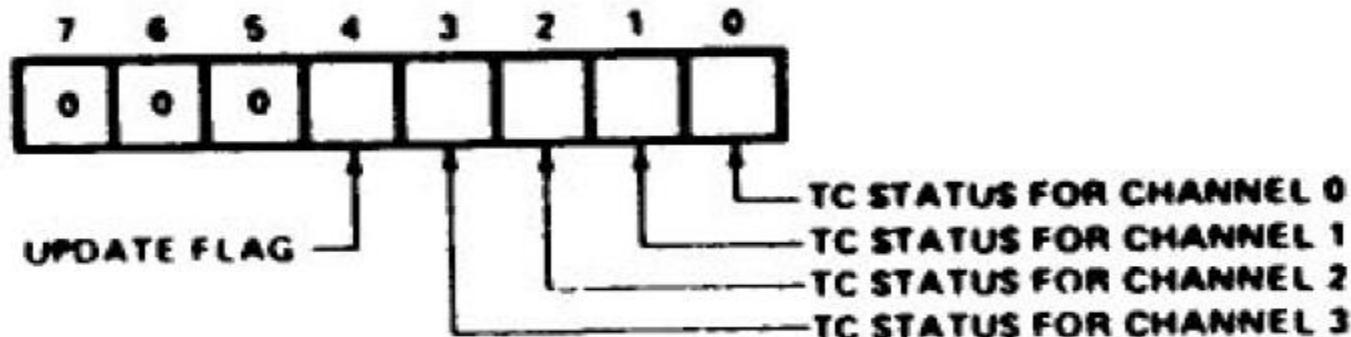


Fig. Status register Bit format

REGISTER	BYTE	ADDRESS INPUTS				F/L	'BI-DIRECTIONAL DATA BUS							
		A ₃	A ₂	A ₁	A ₀		D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
CH-0 DMA Address	LSB	0	0	0	0	0	A ₇	A ₈	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
	MSB	0	0	0	0	1	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈
CH-0 Terminal Count	LSB	0	0	0	1	0	C ₇	C ₆	C ₅	C ₄	C ₃	C ₂	C ₁	C ₀
	MSB	0	0	0	1	1	Rd	Wr	C ₁₃	C ₁₂	C ₁₁	C ₁₀	C ₉	C ₈
CH-1 DMA Address	LSB	0	0	1	0	0	Same as Channel 0							
	MSB	0	0	1	0	1								
CH-1 Terminal Count	LSB	0	0	1	1	0	Same as Channel 0							
	MSB	0	0	1	1	1								
CH-2 DMA Address	LSB	0	1	0	0	0	Same as Channel 0							
	MSB	0	1	0	0	1								
CH-2 Terminal Count	LSB	0	1	0	1	0	Same as Channel 0							
	MSB	0	1	0	1	1								
CH-3 DMA Address	LSB	0	1	1	0	0	Same as Channel 0							
	MSB	0	1	1	0	1								
CH-3 Terminal Count	LSB	0	1	1	1	0	Same as Channel 0							
	MSB	0	1	1	1	1								
MODE SET (Program only)	-	1	0	0	0	0	AL	TCS	EW	RP	EN3	EN2	EN1	ENO
STATUS (Read only)	-	1	0	0	0	0	0	0	0	UP	TC3	TC2	TC1	TC0

*A₀-A₁₅: DMA Starting Address, C₀-C₁₃: Terminal Count value (N-1), Rd and Wr: DMA Verify (00), Write (01) or Read (10) cycle selection, AL: Auto Load, TCS: TC STOP, EW: EXTENDED WRITE, RP: ROTATING PRIORITY, EN3-ENO: CHANNEL ENABLE MASK, UP: UPDATE FLAG, TC3-TC0: TERMINAL COUNT STATUS BITS.

Table. 8257 Register selection

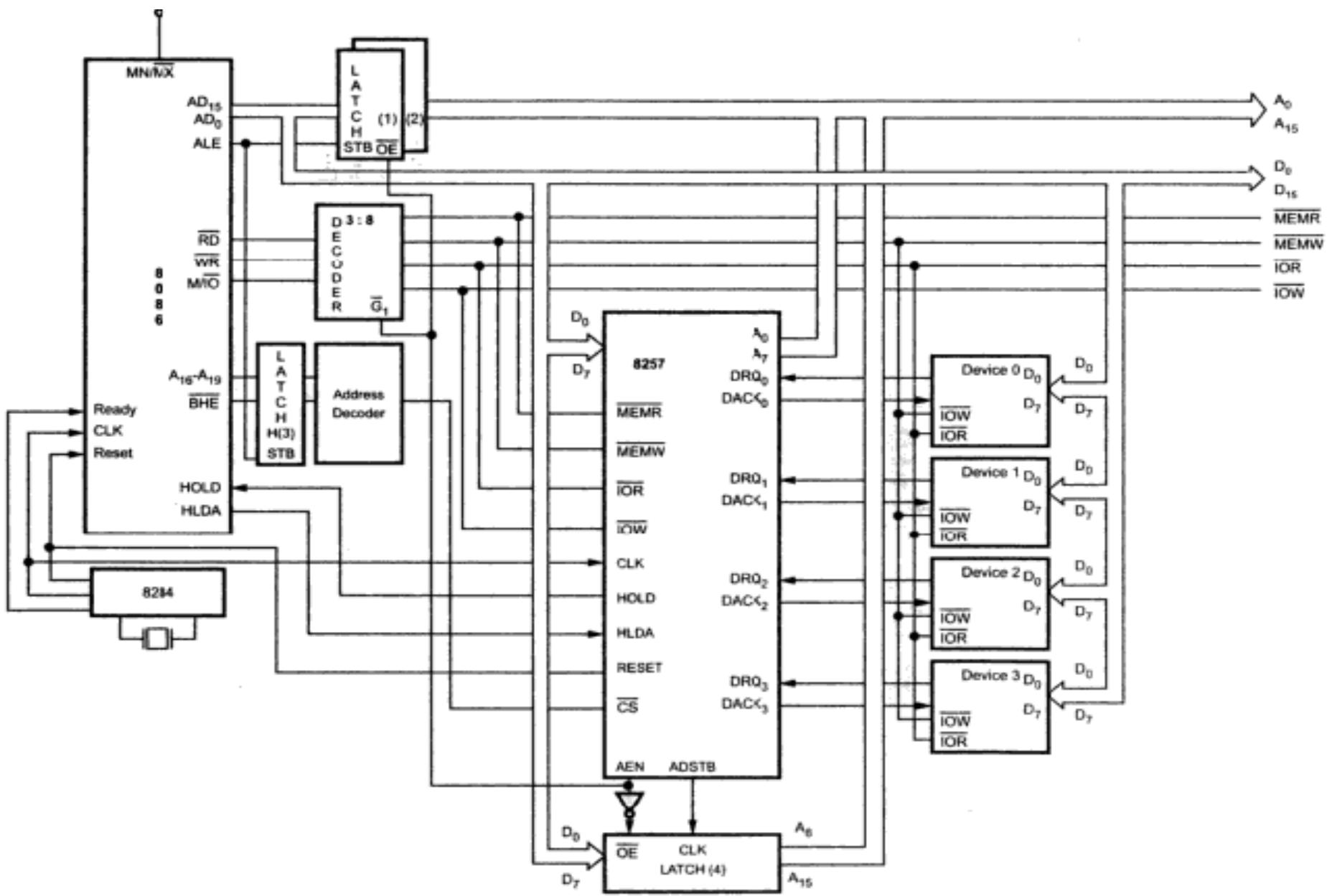


Fig. Interfacing 8257 with 8086

Dr. Venkata Reddy Kota @
JNTUK, Kakinada

8257 BLOCK DIAGRAM

