How microcontroller is different to microprocessor?

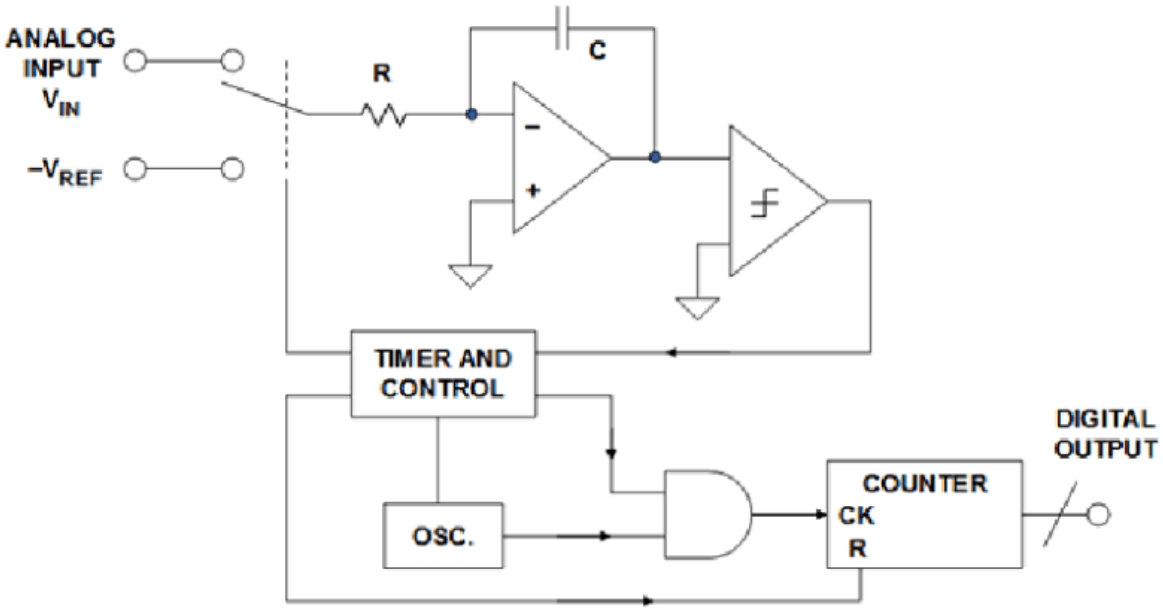
Microprocessor	Microcontroller
Memory and I/O component are connected externally.	Memory and I/O output component are present internally.
Complex circuit	Less complex circuit
Can’t be used in compact system.	Can be used with a compact system.
Not efficient.	Efficient.
Have zero status flag.	Doesn’t have zero status flag.
Less number of registers.	More number of registers.
Generally used in personal computers.	Generally used in washing machines, and air conditioners.

Explain the difference between machine cycle and instruction cycle?

Machine Cycle	Instruction Cycle
Represents a basic unit of operation in the CPU.	Represents the entire process of processing instructions.
Includes steps like fetch, decode, execute, and store.	Includes steps like fetch, decode, execute, and update.
Focuses on the low-level, hardware-oriented aspects of instruction execution.	Focuses on higher-level, process of instruction execution.
Provides a less comprehensive view of instruction execution	Provides a more comprehensive view of instruction execution

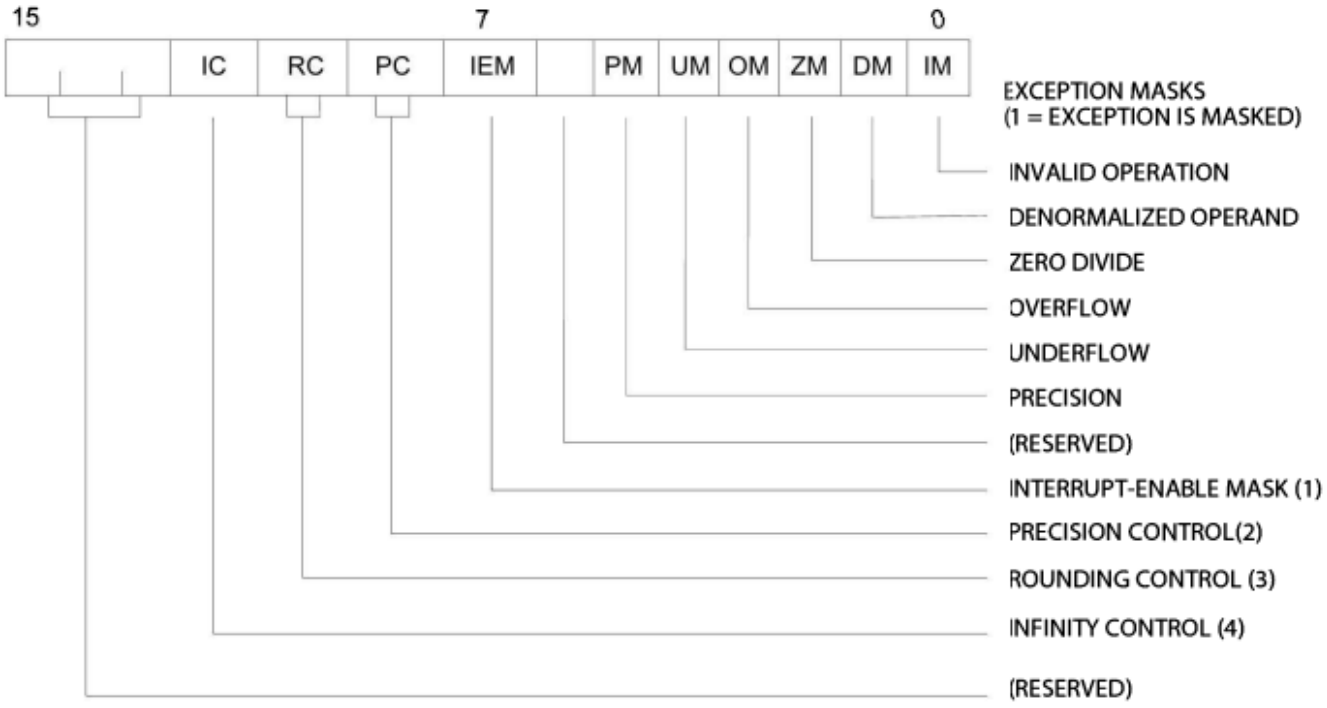
Explain Dual slope A/D convertor.

In a dual slope ADC, the input is integrated on to a capacitor for 2^N cycles of the counter. In the next phase, the counter is reset and the reference is integrated to discharge the capacitor. The value of the counter when the capacitor is totally discharged is the digitized output. As there are two integrations involved, there are two slopes thus giving the name.

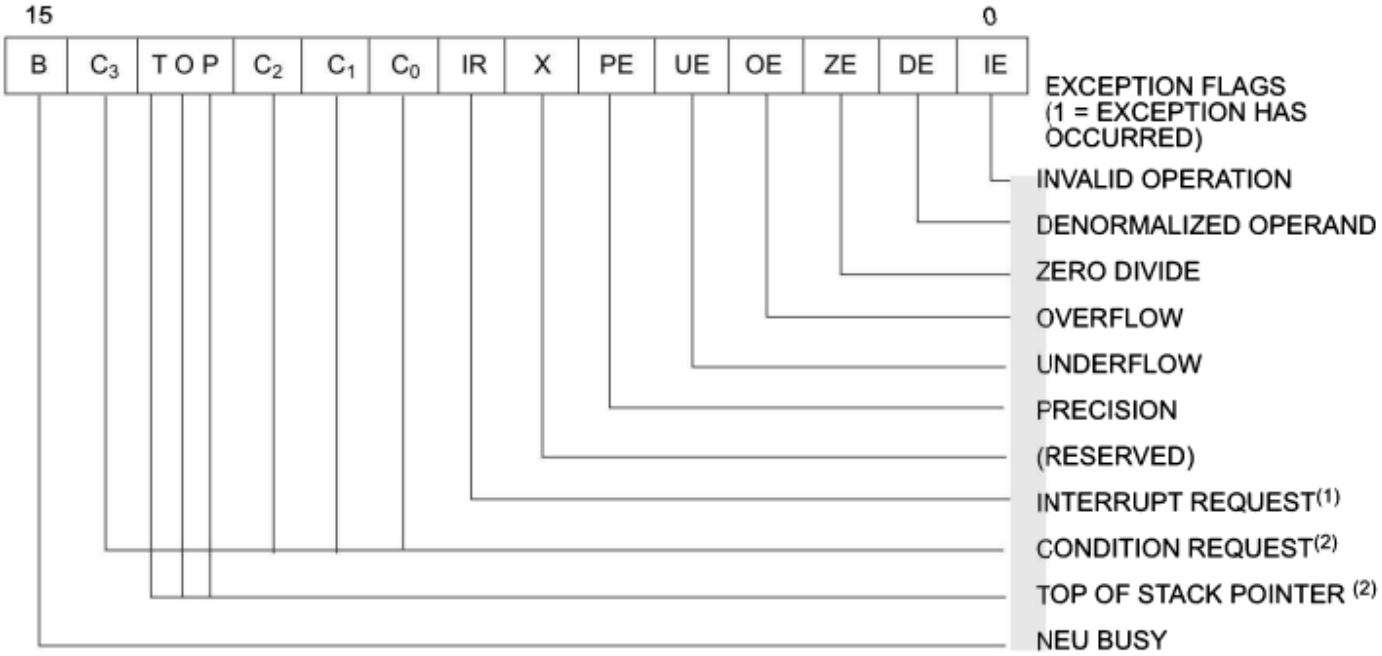


Write 8087 control and status word formats.

Control Word Format



Status Word Format



Name the categories in which Instruction set of 8086 is categorized

Data Copy/Transfer Instructions These types of instructions are used to transfer data from source operand to destination operand. e.g. mov ax, bx

Arithmetic and Logical Instructions All the instructions performing arithmetic, logical, increment, decrement, compare and scan instructions belong to this category. e.g. add ax, bx

Branch Instructions These instructions transfer control of execution to the specified address. e.g. jmp 0x1000

Loop Instructions These are useful to implement different loop structures. e.g. loop label

Machine Control Instructions These instructions control the machine status. e.g. cli

Flag Manipulation Instructions All the instructions which directly affect the flag register, come under this category. e.g. clc

Shift and Rotate Instructions These instructions involve the bitwise shifting or rotation in either direction, with or without a count in CX. e.g. shl al, 1

String Instructions These instructions involve various string manipulation operations. e.g. movsb

Compare the 8085 and 8086 microprocessor.

8085	8086
8 bit microprocessor	16 bit microprocessor
16 bit address line	20 bit address line
8 bit data bus	16 bit data bus
Clock speed is 3MHz	Clock speed varies between 5,8 and 10MHz
5 flags	9 flags
Does not support pipelining	Supports pipelining
50% duty cycle	33% duty cycle
Does not support memory segmentation	Supports memory segmentation

6500 transistors	29000 transistors
Accumulator based processor	General purpose register based processor
No minimum or maximum mode	Has minimum and maximum mode
Only one processor is being used	More than one processor is being used
Only 64 KB memory is used	1MB memory is used

Explain the 80268, Pentium Processors and microcontrollers.

Feature	80268	Pentium Processors	Microcontrollers
Data bus width	32 bits	32 bits or 64 bits	8 bits or 16 bits
Address bus width	16 bits	32 bits or 64 bits	8 bits or 16 bits
Maximum addressable memory	64 MB	4 GB or more	64 KB or more
Clock speed	8 MHz	200 MHz or more	1 MHz or more
Number of registers	16	32 or more	8 or more
Instruction set	230 instructions	500 or more instructions	100 or more instructions
Addressing modes	14	16 or more	8 or more
Interrupts	16	32 or more	8 or more
Pipelining	Yes	Yes	No
Multitasking	No	Yes	No
Applications	Embedded systems	Personal computers, servers, workstations	Appliances, toys, industrial control

What are the functions of the following pins of 8085 microprocessor

(i) READY (ii) ALE (iii) HOLD (iv) TRAP

Ready: If it is high during a read or write cycle, it indicates that the memory or peripheral is ready to send or receive data. If it is low, the CPU will wait for Ready to go high before completing the read or write cycle.

Address Latch Enable (ALE): It occurs during the first clock cycle of a machine state and enables the address to get latched into the on chip latch of peripherals. ALE can also be used to strobe the status information

HOLD indicates that another Master is requesting the use of the Address and Data Buses. The CPU, upon receiving the Hold request. Will relinquish the use of buses as soon as the completion of the current machine cycle. Internal processing can continue. The processor can regain the buses only after the Hold is removed.

TRAP interrupt is a non-maskable restart interrupt. It is recognized at the same time as INTR. It is unaffected by any mask or Interrupt Enable. It has the highest priority of any interrupt.

Tell meaning and operation of operand, instruction word size, machine cycle, T-state for the following opcode used in 8085 microprocessor:

(i) LDA (ii) JMP (iii) POP (iv) DAD (v) CPI (vi) SHLD

Opcode	Meaning	Operand	Instruction word size	Machine Cycle	T state
LDA	Loads the accumulator with the contents of the operand.	Register, Memory location or Immediate value	2 bytes	3 T-states	1 fetch, 1 decode, 1 execute
JMP	Transfers program control to the operand address.	Register or Memory location	2 bytes	3 T-states	1 fetch, 1 decode, 1 execute
POP	Removes the top of the stack and stores it in the accumulator.	None	2 bytes	3 T-states	1 fetch, 1 decode, 1 execute
DAD	Adds the contents of the operand register to the accumulator and stores the result in the accumulator.	Register or Memory location	2 bytes	4 T-states	1 fetch, 1 decode, 1 read operand, 1 execute
CPI	Compares the accumulator with the immediate operand and sets the flags accordingly.	Immediate value	2 bytes	3 T-states	1 fetch, 1 decode, 1 execute
SHLD	Stores the accumulator and the H register into the memory location specified by the operand.	Memory location	3 bytes	7 T-states	1 fetch, 1 decode, 1 read operand, 1 write accumulator, 1 write H, 1 wait

Explain the register organization diagram of 8086.

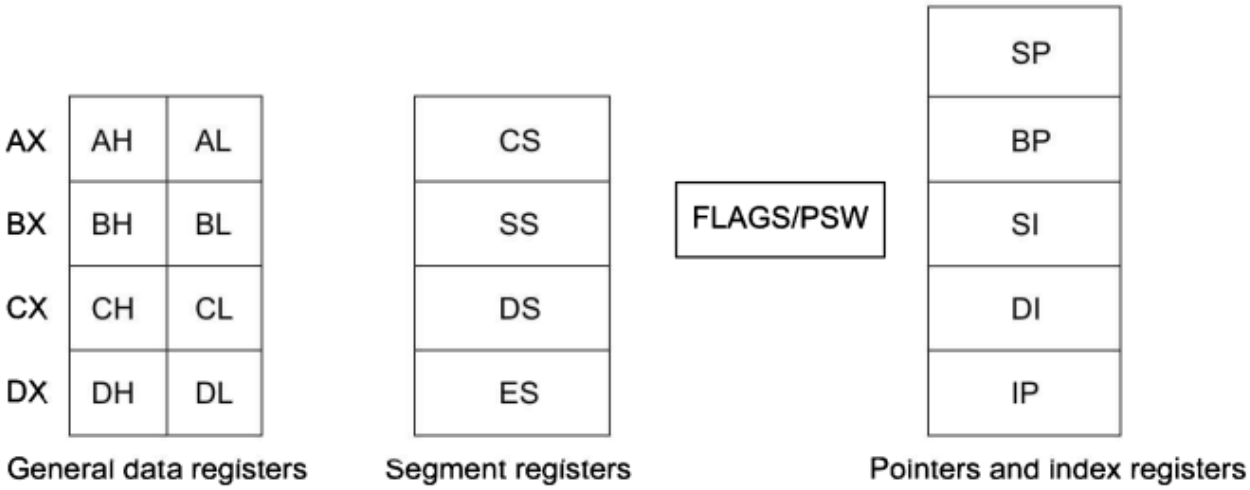


Fig. 1.1 *Register organisation of 8086*

- **General register:** The registers: AX (Accumulator), BX (Base), CX (Counter) and DX (Data) are general purpose 16 bit registers.
- **Segment register:** The complete 1MB memory is divided into 16 logical segments, Each segment contains 64KB memory. There are 4 segment registers: CS (Code Segment), DS (Data Segment), SS (Stack Segment), and ES (Extra Segment).
- **Flag register:** The flag register indicates the result of computation in ALU. It also contains some flag bits to control CPU operations.
- **Pointer register:** Pointer registers are used to store addresses in memory. The 8086 has two pointer registers: SP (Stack Pointer) and BP (Base Pointer).
- **Index register:** Index registers are used to store offsets from a base address. The 8086 has two index registers: SI (Source Index) and DI (Destination Index).

Explain BIU and EU in 8086 microprocessor.

Bus Interface Unit: It provides a full 16-bit bi-directional data bus and 20-bit address bus. The bus interface unit is responsible for performing all external bus operations, as listed below:

1. It sends address of the memory or I/O.
2. It fetches instruction from memory.
3. It reads data from port/memory.
4. It writes data into port/memory.
5. It supports instruction queuing.
6. It provides the address relocation facility.

Execution Unit: The execution unit of 8086 tells the BIU from where to fetch instructions or data, decodes instructions and executes instructions. It contains:

• Control Circuitry

• Instruction Decoder

• Arithmetic Logic Unit (ALU)

• Flag Register

• General Purpose Registers

• Pointers and Index Registers

Compare and explain Maximum Mode and Minimum microprocessor with their timing Diagrams.

Minimum mode	Maximum mode
There can be only one processor.	There can be multiple processors.
Performance is slower.	Performance is faster.
The circuit is simple.	The circuit is complex.
Multiprocessing cannot be performed.	Multiprocessing can be performed.
MN/MX is 1 to indicate the minimum mode.	MN/MX is 0 to indicate the maximum mode
The system is more affordable.	The system costs more money.
It is used for small systems.	It is used for large systems.
The multiprocessor setup is not supported.	The multiprocessor setup is supported

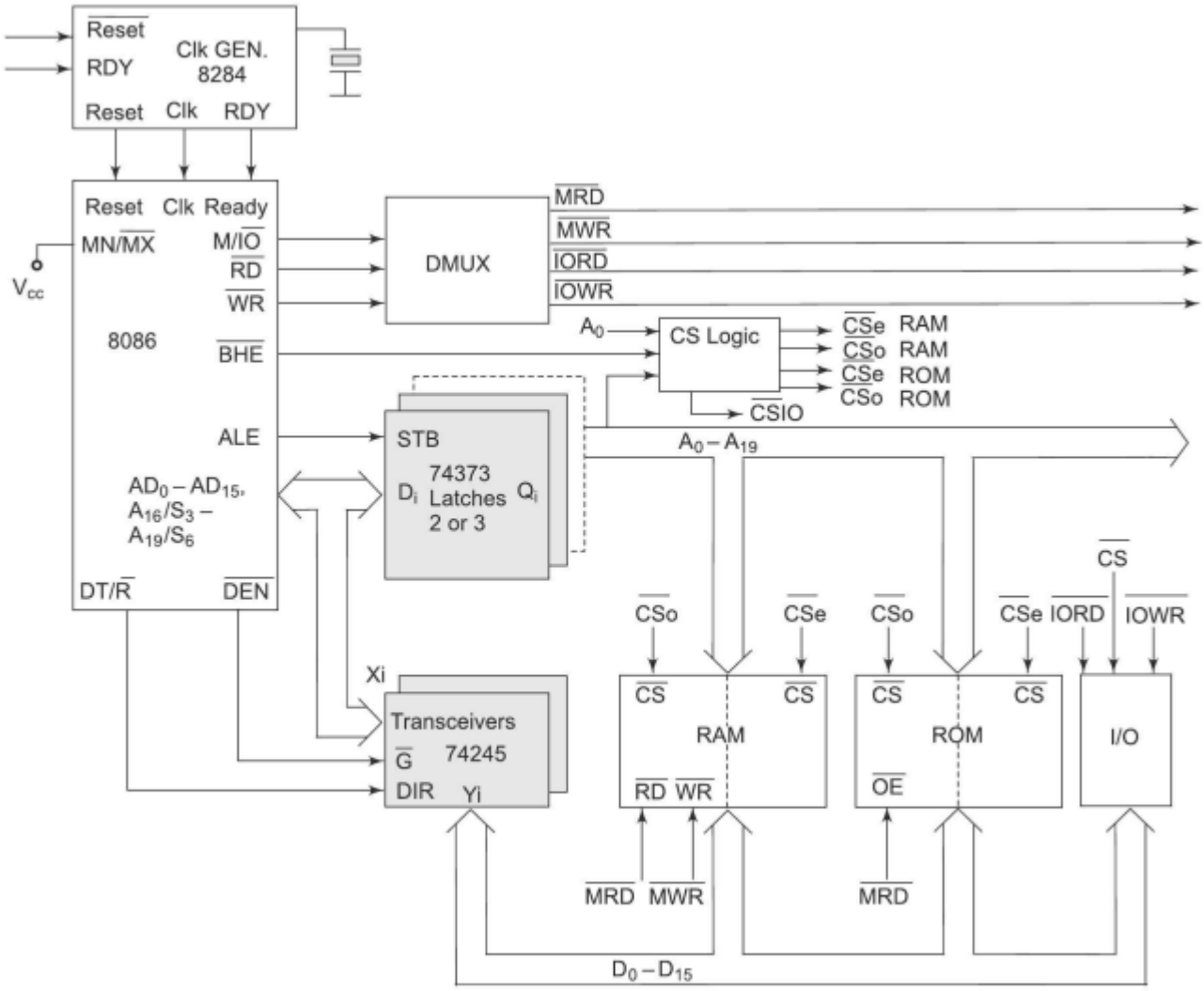


Fig. 1.13 *Minimum Mode 8086 System*

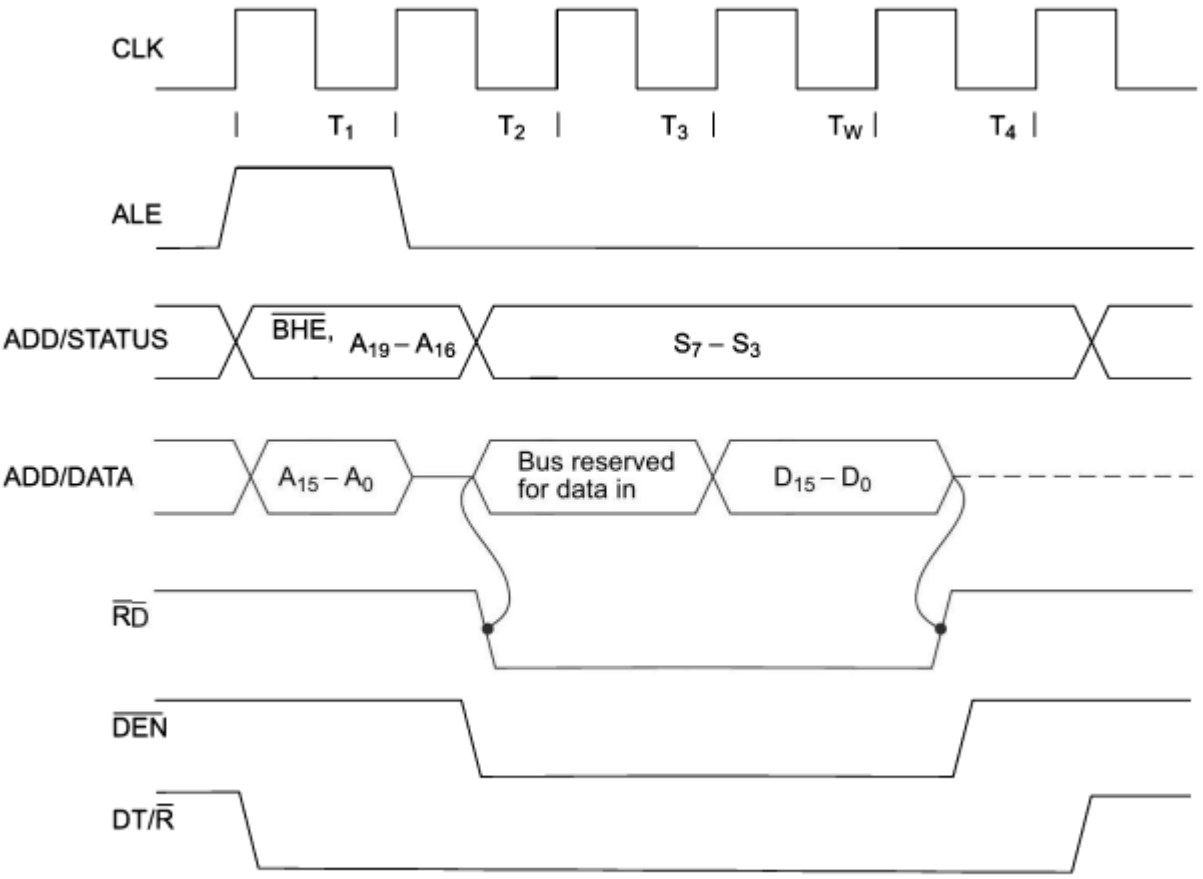


Fig. 1.14(a) *Read Cycle Timing Diagram for Minimum Mode*

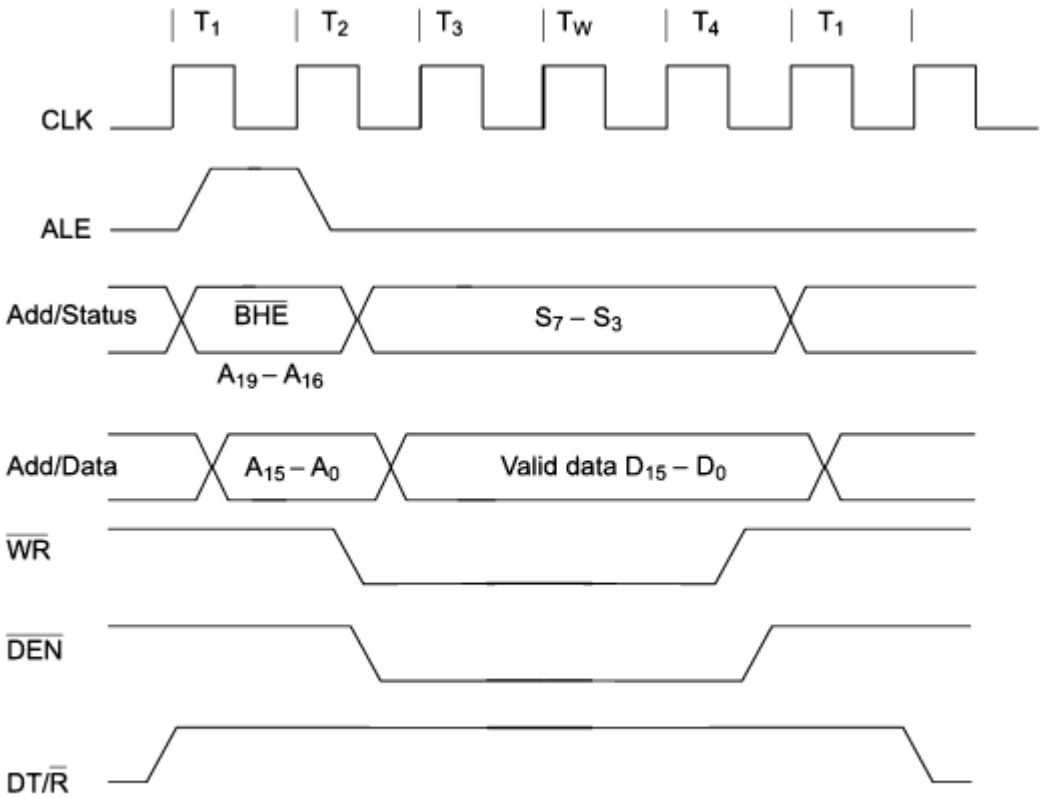


Fig. 1.14(b) *Write Cycle Timing Diagram for Minimum Mode Operation*

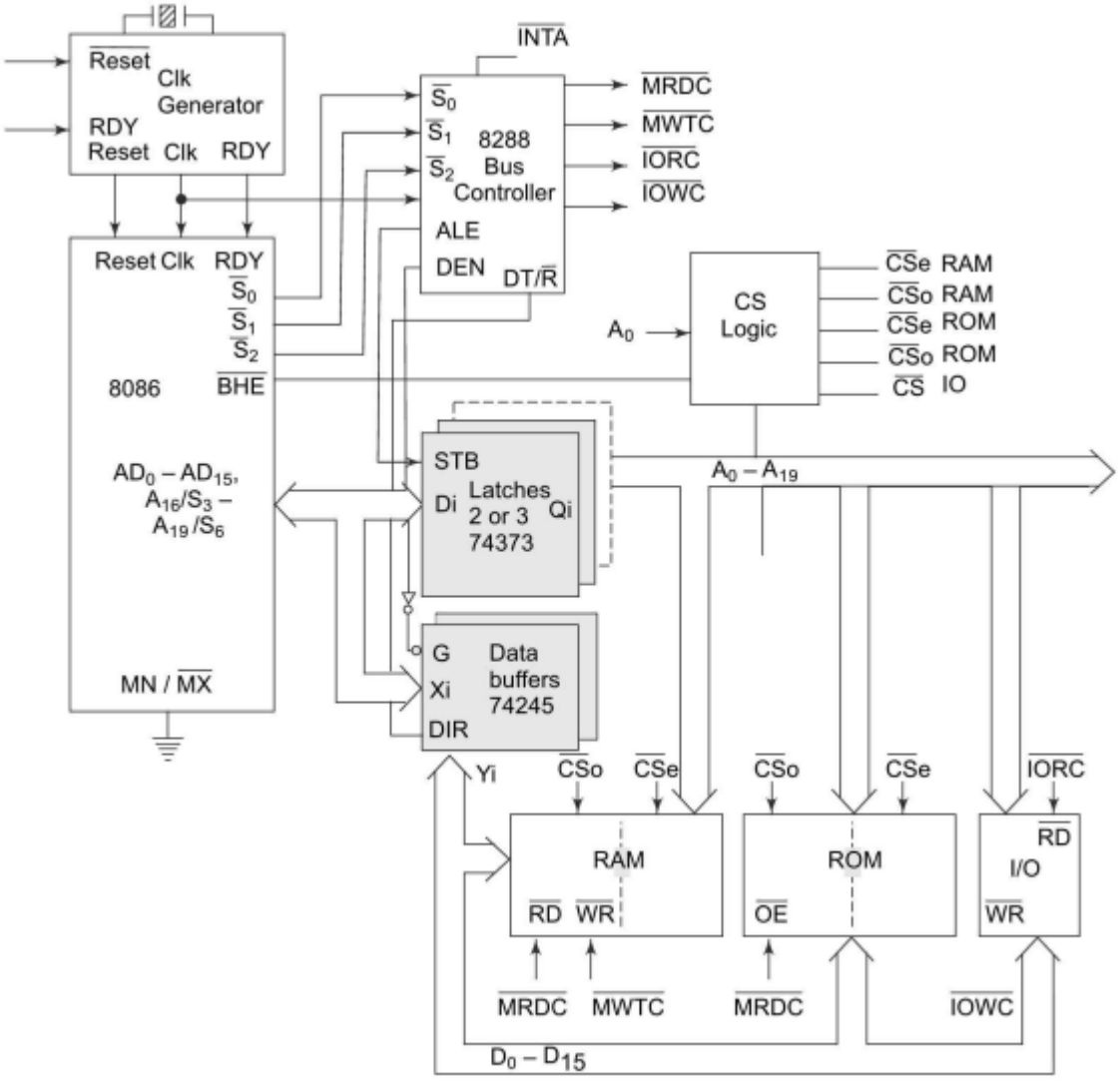


Fig. 1.15 *Maximum Mode 8086 System*

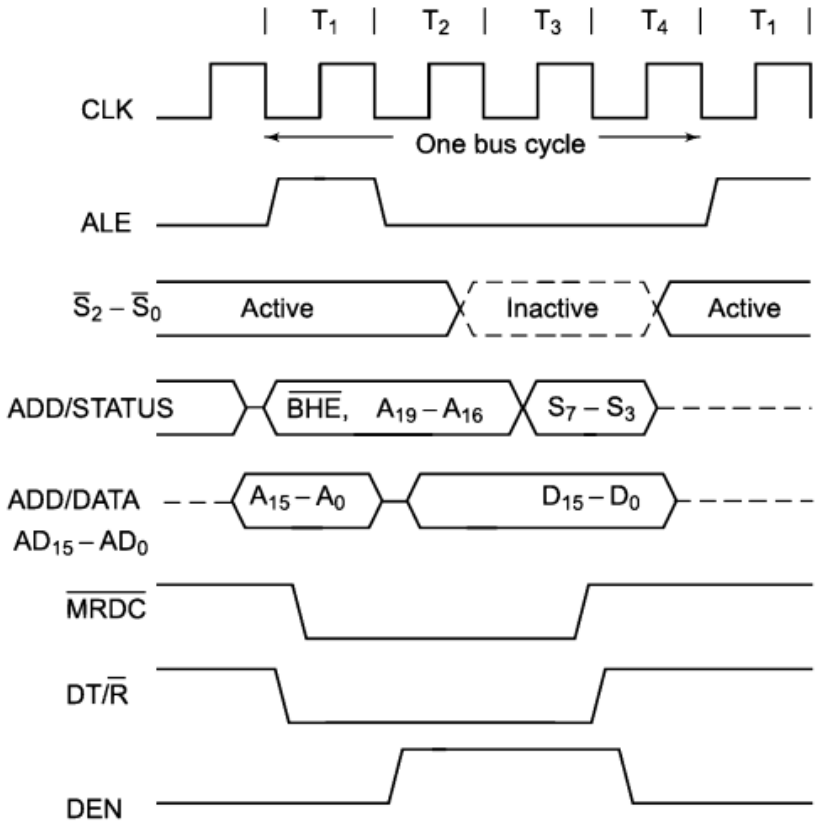


Fig. 1.16 (a) *Memory Read Timing in Maximum Mode*

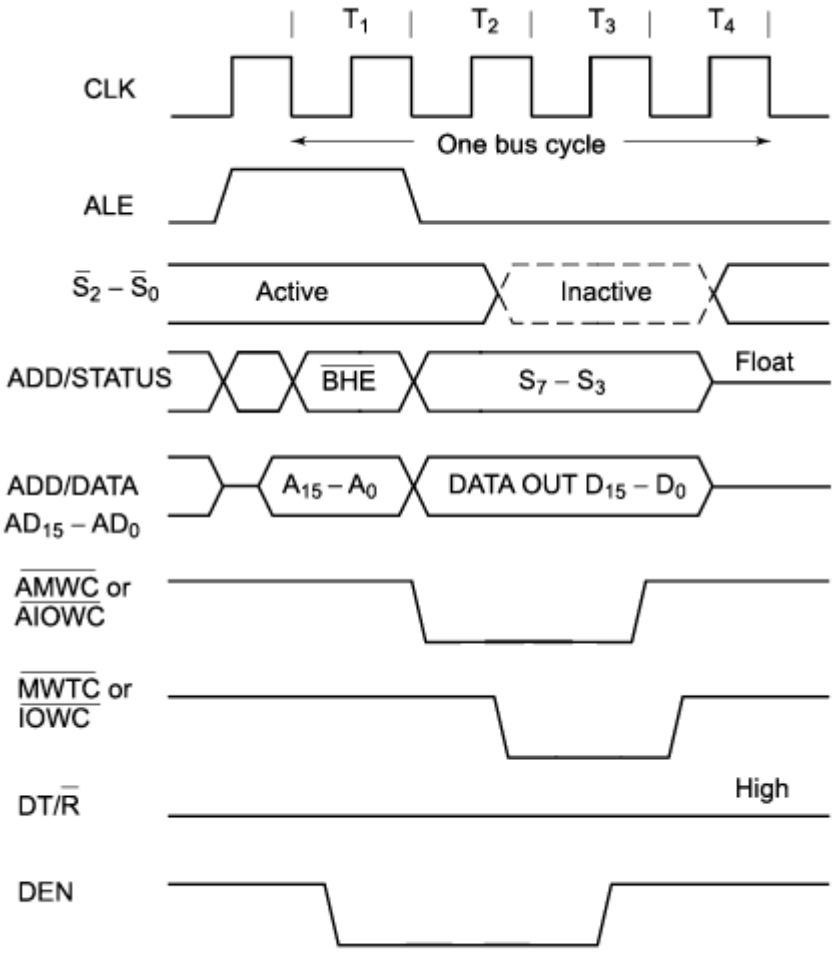


Fig. 1.16(b) *Memory Write Timing in Maximum Mode*

What is the difference between DAA and AAD instructions?

Feature	DAA	AAD
Purpose	Adjusts the accumulator after addition or subtraction operations.	Adjusts the accumulator after division operations.
How it works	Adds or subtracts 6 to the accumulator, depending on the value of the carry flag and the status of the lower nibble of the accumulator.	Multiplies the quotient by 10 and adds the remainder to the accumulator.

Explain Program development algorithm for Assembly language program

Entering a program: The first step in developing an assembly language program is to enter the program code into a text editor.

Assembling a program: Once the program code has been entered, it needs to be assembled into machine code. This is done using an assembler. The assembler will convert the assembly language instructions into machine code that can be executed by the computer.

Linking a program: The machine code generated by the assembler program is not a complete executable program. It needs to be linked with other libraries and modules before it can be executed. This is done using a linker. The linker will combine the machine code with the libraries and modules to create a complete executable program.

Using a debugger: If the program does not work correctly, it can be debugged using a debugger. It helps you to find and fix the errors in the program code.

State the addressing modes of 80386. Given the various exceptions which occur when operating in the Protected Virtual addressing mode.

Addressing Mode	Description	Example
Immediate	The operand is the immediate value following the instruction.	MOV AX, 10h
Register	The operand is the value of a register.	MOV AX, BX
Direct	The operand is the address of the data in memory.	MOV AX, [1000h]

Register indirect	The operand is the address of the data in memory, as stored in a register.	MOV AX, [BX]
Base indexed	The operand is the address of the data in memory, as calculated by adding the base register and the index register.	MOV AX, [BX + SI]
Base indexed with displacement	The operand is the address of the data in memory, as calculated by adding the base register, the index register, and a displacement value.	MOV AX, [BX + SI + 10h]
Relative	The operand is the address of the data in memory, as calculated by adding the current instruction pointer and a displacement value.	MOV AX, [BP + 10h]
Segment override	The operand is the address of the data in memory, as stored in a segment register.	MOV AX, CS:[1000h]
Scaled index	The operand is the address of the data in memory, as calculated by adding the base register, the index register, and a displacement value, multiplied by a scale factor.	MOV AX, [BX + SI * 2]
Based indexed with displacement and scale factor	The operand is the address of the data in memory, as calculated by adding the base register, the index register, a displacement value, and a scale factor.	MOV AX, [BX + SI * 2 + 10h]

Exception	Description
General Protection Fault	This exception occurs when a program attempts to access a memory location that it is not allowed to access.
Page Fault	This exception occurs when a program attempts to access a page that is not currently loaded into memory.
Alignment Check Fault	This exception occurs when a program attempts to access a memory location that is not aligned on a 16-byte boundary.
Invalid Opcode Fault	This exception occurs when a program attempts to execute an invalid opcode.

Divide by Zero Fault	This exception occurs when a program attempts to divide by zero.
Debug Exception	This exception occurs when a program triggers a breakpoint or a single-step instruction.

Why 8087 is known as coprocessor?

The 8087 is known as a coprocessor because it is an additional processor that works alongside the main processor (the 8088 or 8086). The 8087 is specifically designed to perform floating-point arithmetic, which is a type of math that is often used in scientific and engineering applications. The 8087 can perform floating-point operations much faster than the main processor, so it can offload these operations from the main processor and free it up to do other tasks.

What is the difference Single Handshake I/O and Double-Handshake data transfer for parallel data transfer.

- **Single handshake:** The peripheral sends a strobe signal to the microprocessor to indicate that it is ready to send data. The microprocessor then opens its input port and receives the data. After receiving the data, the microprocessor sends an acknowledge signal to the peripheral to indicate that the transmission has been completed.
- **Double handshake:** The peripheral first sends a strobe signal to the microprocessor. The microprocessor then sends an acknowledge signal to the peripheral to indicate that it is ready to receive data. The peripheral then sends the data to the microprocessor. After the data has been sent, the peripheral sends a strobe signal to indicate that the transmission has been completed. The microprocessor then drops its acknowledge signal, which indicates that the session has been completed.

Feature	Single Handshake I/O	Double Handshake Data Transfer
Complexity	Simpler	More complex
Efficiency	More efficient	Less efficient
Reliability	Less reliable	More reliable
Throughput	Higher	Lower
Applications	Speed-critical applications	Reliability-critical applications

Explain the Architecture and signal descriptions for 8254 chip. Also explain the different operating modes of programmable timer device (8254).

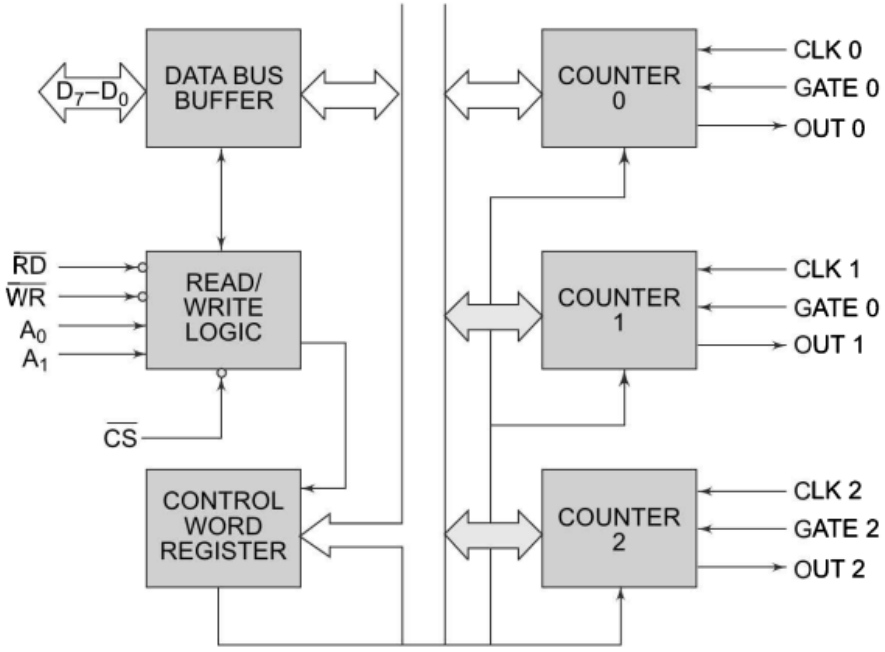


Fig. 6.1(a) *Internal Block Diagram of 8254*

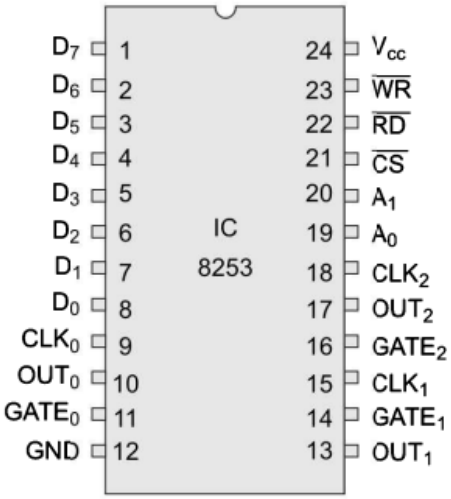


Fig. 6.1(b) *Pin Configuration of 8254*

Architecture of Programmable Timer Device (8254)

- **Data bus buffer:** The data bus buffer is a tri-state, bi-directional, 8-bit buffer that is used to interface the 8254 to the system data bus. It has three basic functions:
  - Programming the modes of the 8254.
  - Loading the count registers.
  - Reading the count values.

- **Read/write logic:** The read/write logic is used to control the flow of data between the 8254 and the system data bus. It includes 5 signals: RD, WR, CS, A1, and A0.
  - RD (Read): This signal is active low and indicates that the CPU is reading data from the 8254.
  - WR (Write): This signal is active low and indicates that the CPU is writing data to the 8254.
  - CS (Chip Select): This signal is active low and indicates that the CPU is accessing the 8254.
  - A1 and A0 (Address): These are two address bits that are used to select the counter that is being accessed.

- **Control word register:** The control word register is a 8-bit register that is used to control the operation of the 8254. It is located at address 0000H in the system address space. The control word register can be written to or read from by the CPU.

- **Counters:** The 8254 has three 16-bit counters. Each counter can be programmed to operate in one of six different modes. The counters are located at addresses 0001H, 0002H, and 0003H in the system address space.

Signal Description of Programmable Timer Device (8254)

- **A0-A1:** These are two address bits that are used to select the counter that is being accessed.
- **D0-D7:** These are eight data bits that are used to read or write data to the 8254.
- **WR:** This is an active low signal that is used to write data to the 8254.
- **CS:** This is an active low signal that is used to select the 8254 chip.
- **RD:** This is an active low signal that is used to read data from the 8254.
- **CLK:** This is a clock signal that is used to clock the 8254.
- **GATE:** This is an active high signal that is used to enable or disable counting.
- **OUT:** This is an output signal that is generated when the counter reaches zero.

Operating Modes of Programmable Timer Device (8254)

- **Mode 0 (Interrupt on Terminal Count):** This mode is typically used for event counting. When the counter reaches 0, an interrupt is generated.
- **Mode 1 (Programmable One Shot):** This mode is used to generate a one-shot pulse. When the counter reaches 0, the output goes low and remains low until the counter is reloaded with a new value.
- **Mode 2 (Rate Generator):** This mode is used to generate a square wave with a period equal to the count value. The output goes high when the counter reaches 0 and goes low when the counter reaches the count value minus 1.
- **Mode 3 (Square Wave Generator):** This mode is similar to mode 2, but the output is always high when the counter is counting down and goes low when the counter reaches 0.
- **Mode 4 (Software Triggered Strobe):** This mode is used to generate a strobe pulse. When the GATE input is asserted, the output goes high and remains high until the GATE input is de-asserted.
- **Mode 5 (Hardware Triggered Strobe):** This mode is similar to mode 4, but the strobe pulse is generated when the GATE input is asserted by an external signal.