## Write two features of Pentium Microprocessor not present in 80486.

- **64-bit data bus:** The Pentium microprocessor has a 64-bit data bus, which allows it to transfer data more quickly than the 80486, which has a 32-bit data bus.
- **Superscalar architecture:** The Pentium microprocessor is a superscalar processor, which means that it can execute multiple instructions in parallel.

## Describe the effect that a control word of 10010000 sent to 8254 will have

It will configure the 8254 to generate an interrupt after a specified amount of time. This can be used to generate delays, measure time intervals, or trigger other events.

## Write 8086 instructions to set the trap flag.

OR WORD PTR[BP+0],0100H

## Write 8086 instructions to reset the trap flag.

AND WORD PTR[BP+0],0FEFFH

## Differentiate between RET and IRET instructions

| Feature | RET | IRET |
|---|---|---|
| Purpose | Returns from a subroutine | Returns from an interrupt |
| Pops from stack | Return address | Return address, code segment selector, flags register |
| May cause privilege level change | No | Yes |
| When to use | To return from a subroutine | To return from an interrupt, or to return from a subroutine that needs to modify the flags register |

## Differentiate between carry and overflow flags.

| Feature | Carry Flag (CF) | Overflow Flag (OF) |
|---|---|---|

| | | |
|---|---|---|
| Purpose | Indicates a carry out of the MSB | Indicates an overflow |
| Affected bits | MSB | Entire result |
| When to use | To debug programs | To optimize code |

## What logic levels would you find on BHE and $A_0$ when an 8086 is writing a byte to address 04274H? When it is writing a word to 04274H?

hen an 8086 is writing a byte to address 04274H, the logic levels on the BHE and A0 signals would be: **BHE: Low, $A_0$: Low**

This is because the address 04274H is a byte address, and the BHE signal is used to select between byte and word addresses. When the BHE signal is low, the 8086 will access the lower 8 bits of the address.

When an 8086 is writing a word to address 04274H, the logic levels on the BHE and A0 signals would be: **BHE: High, $A_0$: Low**

This is because the address 04274H is a word address, and the BHE signal is used to select between byte and word addresses. When the BHE signal is high, the 8086 will access the upper 8 bits of the address.

## You are required to transfer data bytes to and from a floppy disk controller interfaced to 8086 via 8255. Which mode of operation of 8255 you will use and why?

**Mode 0** is a good choice for transferring data to and from a floppy disk controller. It is simple to use, it provides a good balance of performance and flexibility, and it is compatible with most floppy disk controllers. Mode 0 allows the 8255 to be used as a bidirectional port, which means that it can be used to both send and receive data.

In mode 0, the 8255 has three ports: Port A, Port B, and Port C. Port A is used to send data to the floppy disk controller, Port B is used to receive data from the floppy disk controller, and Port C is used for control signals.

## Which interrupt has higher priority DIVIDE BY ZERO or NMI and why?

The NMI interrupt has a higher priority than the divide by zero interrupt. This is because the NMI interrupt is a non-maskable interrupt, which means that it cannot be ignored by the processor. The divide by zero interrupt, on the other hand, is a maskable interrupt, which means that it can be ignored by the processor if the interrupt flag is not set.

## Differentiate between memory mapped and isolated input-output

| Isolated I/O | Memory Mapped I/O |
|---|---|
| Memory and I/O have separate address space | Both have same address space |
| All address can be used by the memory | Due to addition of I/O addressable memory become less for memory |
| Separate instruction control read and write operation in I/O and Memory | Same instructions can control both I/O and Memory |
| In this I/O address are called ports. | Normal memory address are for both |
| More efficient due to separate buses | Lesser efficient |
| Larger in size due to more buses | Smaller in size |
| It is complex due to separate logic is used to control both. | Simpler logic is used as I/O is also treated as memory only. |

## Compare the features of Pentium II, Pentium III and Pentium IV Processors

| Feature | Pentium 2 | Pentium 3 | Pentium 4 |
|---|---|---|---|
| Clock speed | 233 MHz to 450 MHz | 450 MHz to 1.4 GHz | 1.3 GHz to 3.8 GHz |
| Front-side bus (FSB) | 66 MHz to 100 MHz | 100 MHz to 133 MHz | 400 MT/s to 1066 MT/s |
| Power consumption | 35 W to 70 W | 35 W to 100 W | 55 W to 115 W |
| Architecture | Intel's sixth generation | P6 microarchitecture | Netburst microarchitecture |
| Max. Clock | 450 MHz | 450~1400 MHz | 800~3000 MHz |

| | | | |
|---|---|---|---|
| Introduced | 1997 | 1999 | 2000 |

## Compare Microprocessors and Microcontrollers with respect to their architecture and specific applications

| Parameter | Microprocessor | Microcontroller |
|---|---|---|
| **Definition** | Microprocessors can be understood as the heart of a computer system. | Microcontrollers can be understood as the heart of an embedded system. |
| **What is it?** | A microprocessor is a processor where the memory and I/O component are connected externally. | A microcontroller is a controlling device wherein the memory and I/O output component are present internally. |
| **Circuit complexity** | The circuit is complex due to external connection. | Microcontrollers are present on chip memory. The circuit is less complex. |
| **Memory and I/O components** | The memory and I/O components are to be connected externally. | The memory and I/O components are available. |
| **Compact system compatibility** | Microprocessors can’t be used in compact system. | Microcontrollers can be used with a compact system. |
| **Efficiency** | Microprocessors are not efficient. | Microcontrollers are efficient. |
| **Zero status flag** | Microprocessors have a zero status flag. | Microcontroller doesn’t have a zero status flag. |
| **Number of registers** | Microprocessors have less number of registers. | Microcontrollers have more number of registers. |
| **Applications** | Microprocessors are generally used in personal computers. | Microcontrollers are generally used in washing machines, and air conditioners. |

## With respect to 8086 explain the following signals: i) HOLD and HLDA ii) NMI iii) INTR

**HOLD input, HLDA output:** A HIGH or HOLD pin indicates that another master (DMA) is requesting to take over the system bus. On receiving HOLD signal processor outputs HLDA signal HIGH as an acknowledgement. At the same time, processor tristates the system bus. A low on HOLD gives the system bus control back to the processor. Processor then outputs low signal on HLDA.

**NMI:** It is a positive edge triggered nonmaskable interrupt request.

**INTR:** It is a level triggered maskable interrupt request. It is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt service routine.

## Explain the following terms with respect to 8086:i) Physical address ii) Offset address iii) Segment Address iv) Segment override Prefix v) Default Segment

**Physical address:** The physical address is the actual address of a memory location in the computer's memory. The physical address is a 20-bit number that is used to access memory.

**Offset address:** The offset address is a 16-bit number that is used to address a location within a segment. The offset address is added to the base address of the segment to get the physical address of the memory location.

**Segment address:** The segment address is a 16-bit number that is used to identify a segment in memory. The segment address is used together with the offset address to get the physical address of a memory location.

**Segment override prefix:** The segment override prefix is a special byte that can be used to override the default segment for an instruction. The segment override prefix is used when the programmer wants to access a memory location in a different segment than the default segment.

**Default segment**: The default segment is the segment that is used by the processor when no segment override prefix is used. The default segment is typically the code segment, but it can be changed by the programmer.

## Draw the minimum mode time diagram of 8086 for read and write operations.

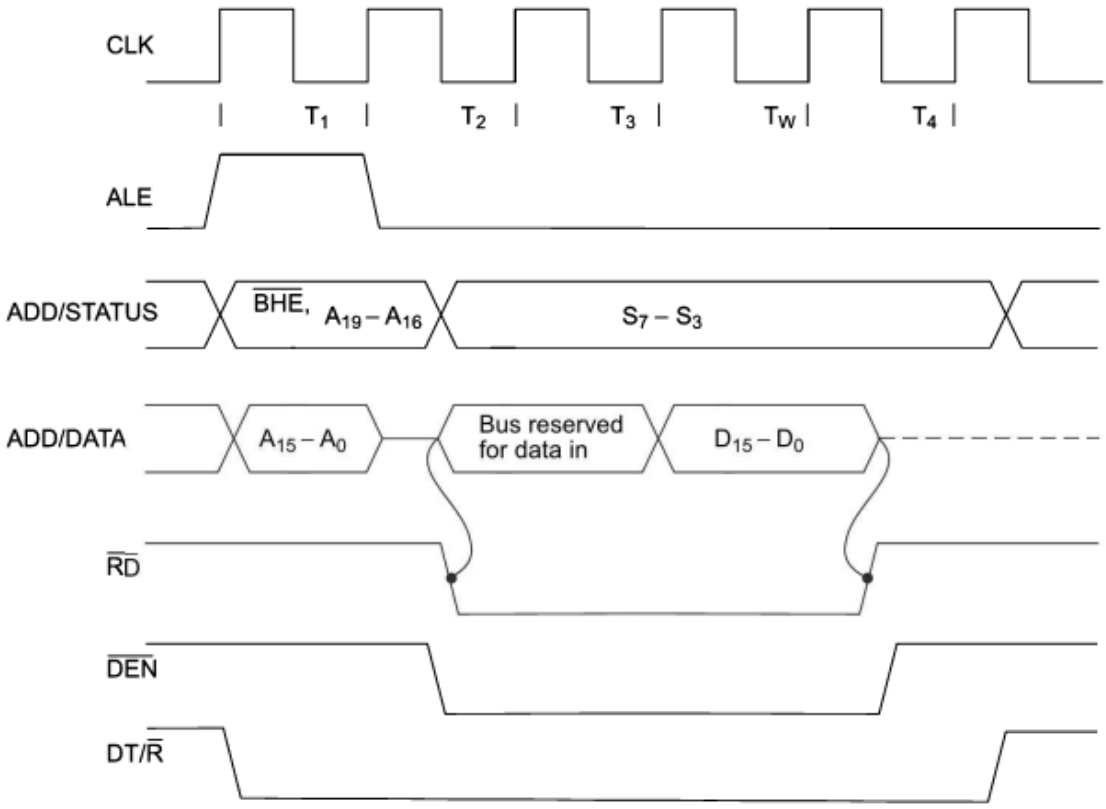


**Fig. 1.14(a)** *Read Cycle Timing Diagram for Minimum Mode*

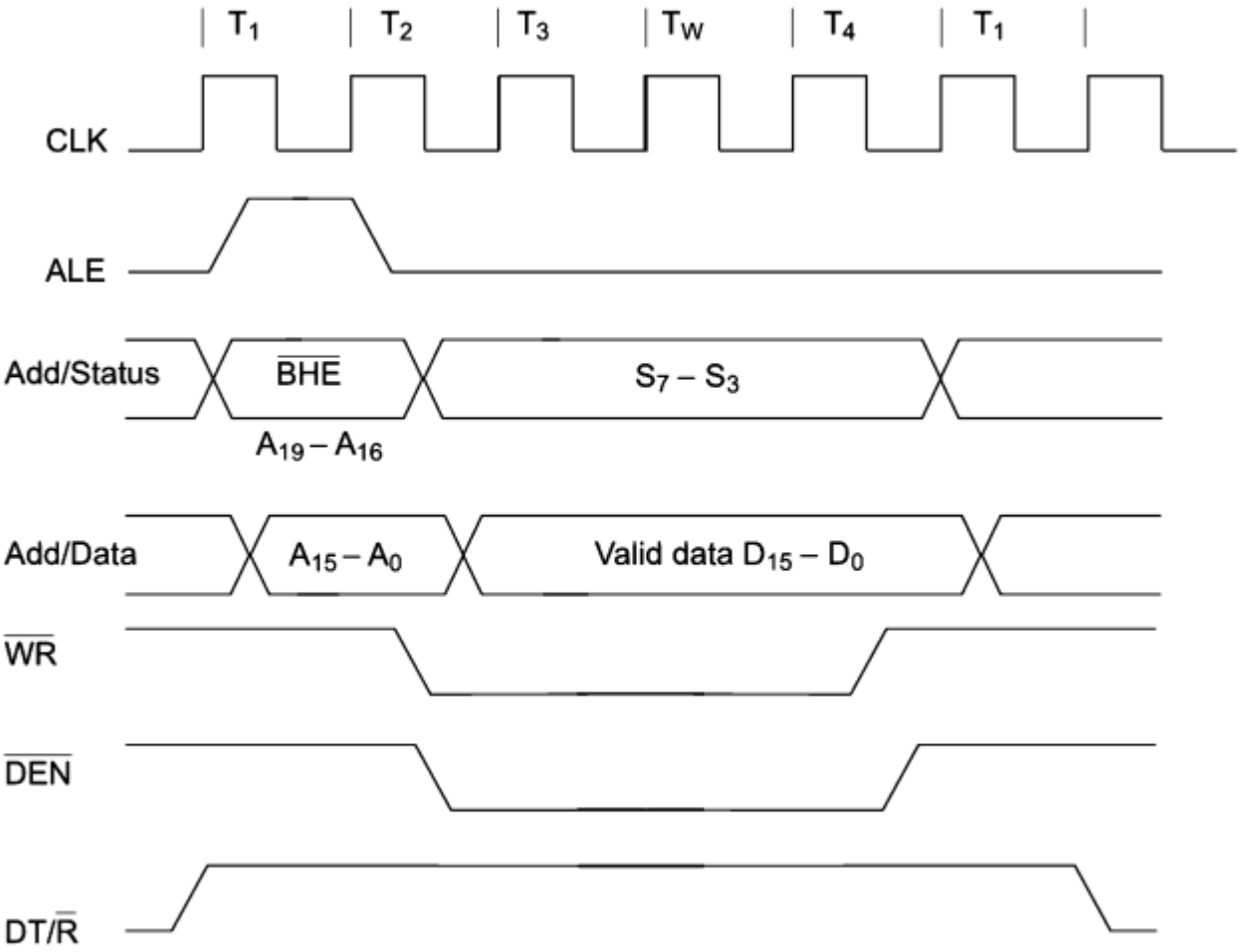


**Fig. 1.14(b)** *Write Cycle Timing Diagram for Minimum Mode Operation*

## What are Assembler directives? Explain the following assembler directives DPTR ii) OFFSET EVEN ii) PUBLIC iv) ASSUME

Assembler directives are instructions that are used by the assembler to control the assembly process. They do not represent instructions that are executed by the processor.

- **DPTR:** The DPTR directive is used to define the data pointer register. The data pointer register is a 16-bit register that is used to store the address of a memory location. The DPTR directive is typically used in assembly language programs that access memory indirectly.
- **OFFSET EVEN:** The OFFSET EVEN directive is used to ensure that the offset of a memory location is even. This is important for some instructions, such as the MOVS instruction, which can only access even-numbered memory locations.
- **PUBLIC:** The PUBLIC directive is used to make a symbol visible to other modules. This means that other modules can access the symbol by its name. The PUBLIC directive is typically used for global variables and functions.
- **ASSUME:** The ASSUME directive is used to specify the default segment registers for code and data. This is useful for assembly language programs that use multiple segments.

## The binary code of an instruction is 10001010 00010101. Write the corresponding assembly language instruction. What are the addressing modes used in this instruction?

**MOV DL, [BX+SI]**

In this instruction, the addressing modes used are:

1. **Register Direct:** DL is the destination register where the data is being moved.

2. **Base Register with Index Register:** The source operand [BX+SI] combines the contents of the base register BX and the index register SI to form the effective address from which the data is read.

## What do you understand by REP prefix used with string instructions?

The REP prefix is used with string instructions to repeat the instruction a specified number of times. The number of times the instruction is repeated is stored in the CX register.

For example, the following code will compare the contents of memory location 1000h to memory location 2000h 100 times.

```
MOV CX, 100

REPE CMPS [1000h], [2000h]
```

## Explain how inter-processor communication is handled in loosely coupled systems.

Inter-processor communication (IPC) in loosely coupled systems is handled through message passing. Message passing system is a software layer that provides a way for processors to send and receive messages.

## Design a system to interface Analog to digital convertor 0808 with 8086 using 8255 ports. Use port A of 8255 for transferring digital data output of ADC to the CPU and port for control signals. Assume that an analog input is present at I/P2 of the ADC and a clock input of suitable frequency is available for ADC.

**Solution** Figure 5.39 shows the interfacing connections of ADC0808 with 8086 using 8255. The analog input $I/P_2$ is used and therefore address pins A,B,C should be 0,1,0 respectively to select $I/P_2$. The OE and ALE pins are already kept at +5V to select the ADC and enable the outputs. Port C upper acts as the input port to receive the EOC signal while port C lower acts as the output port to send SOC to the ADC. Port A acts as a 8-bit input data port to receive the digital data output from the ADC. The 8255 control word is written as follows:

| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | Control word |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | = 98 H |

The required ALP is given as follows:

```
        MOV AL,98 H              ; Initialise 8255 as
        OUT CWR,AL               ; discussed above
        MOV AL,02H               ; Select I/P2 as analog
        OUT PORT B,AL            ; input
        MOV AL,00H               ; Give start of conversion
        OUT PORT C,AL            ; pulse to the ADC.
        MOV AL,01 H              ;
        OUT PORT C,AL            ;
        MOV AL,00H               ;
        OUT PORT C,AL            ;
WAIT :  IN AL,PORTC              ; Check for EOC by
        RCL                      ; reading port C upper and
        JNC WAIT                 ; rotating through carry.
        IN AL,PORTA              ; If EOC, read digital equivalent in
                                      AL
        HLT                      ; Stop
```

**Program 5.11** *ALP for Problem 5.16*

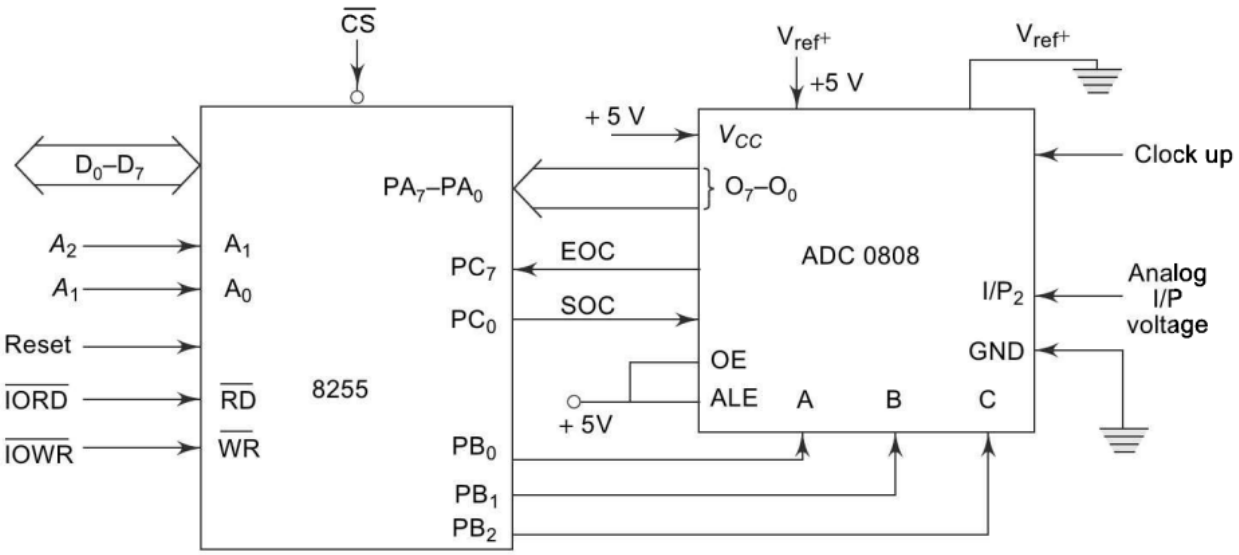


**Fig. 5.39** *Interfacing 0808 with 8086*