1. **Define computer peripheral interfacing with example.**

**Ans:** Computer Peripheral Interfacing is the interactions between computer processor and computer peripherals. Any input or output process that occurs through a peripheral device is known as computer peripheral interfacing. Peripheral devices are those devices that are linked either internally or externally to a computer. These devices are commonly used to transfer data. The most common processes that are carried out in a computer are entering data and displaying processed data. Several devices can be used to receive data and display processed data. The devices used to perform these functions are called peripherals or I/O devices. Peripherals read information from or write in the memory unit on receiving a command from the CPU. Example: Keyboard, Printer, Magnetic Tape, Magnetic Disk.

1. **Compare port-addressed I/O and memory-mapped I/o on basis of address length, control signal, and instruction.**

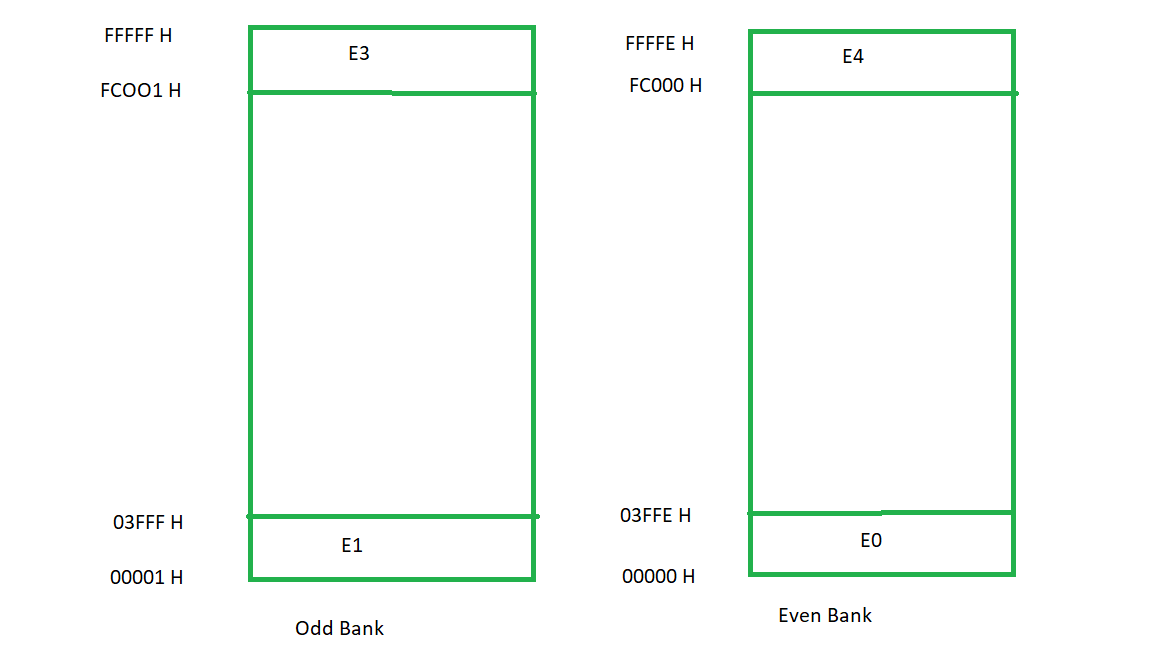
**Ans:**

|  |  |  |
| --- | --- | --- |
| Features | Memory Mapped IO | IO Mapped IO |
| Addressing | IO devices are accessed like any other memory location. | They cannot be accessed like any other memory location. |
| Address Size | They are assigned with 16-bit address values. | They are assigned with 8-bit address values. |
| Instructions Used | The instruction used are LDA and STA, etc. | The instruction used are IN and OUT. |
| Cycles | Cycles involved during operation are Memory Read, Memory Write. | Cycles involved during operation are IO read and IO writes in the case of IO Mapped IO. |
| Registers Communicating | Any register can communicate with the IO device in case of Memory Mapped IO. | Only Accumulator can communicate with IO devices in case of IO Mapped IO. |
| Space Involved | 216 IO ports are possible to be used for interfacing in case of Memory Mapped IO. | Only 256 I/O ports are available for interfacing in case of IO Mapped IO. |
| IO/M` signal | During writing or read cycles (IO/M` = 0 ) in case of Memory Mapped IO. | During writing or read cycles (IO/M` = 1) in case of IO Mapped IO. |
| Control Signal | No separate control signal required since we have unified memory space in the case of Memory Mapped IO. | Special control signals are used in the case of IO Mapped IO. |
| Arithmetic and Logical operations | Arithmetic and logical operations are performed directly on the data in the case of Memory Mapped IO. | Arithmetic and logical operations cannot be performed directly on the data in the case of IO Mapped IO. |

1. **What do you mean by computer peripheral interfacing? (see 1)**
2. **Discuss port-addressed I/O and memory-mapped I/O with control signal, and instruction.**

**Ans:** **SEE 2**

1. **What do you mean by odd address bank and even address bank of 8086 processor?**

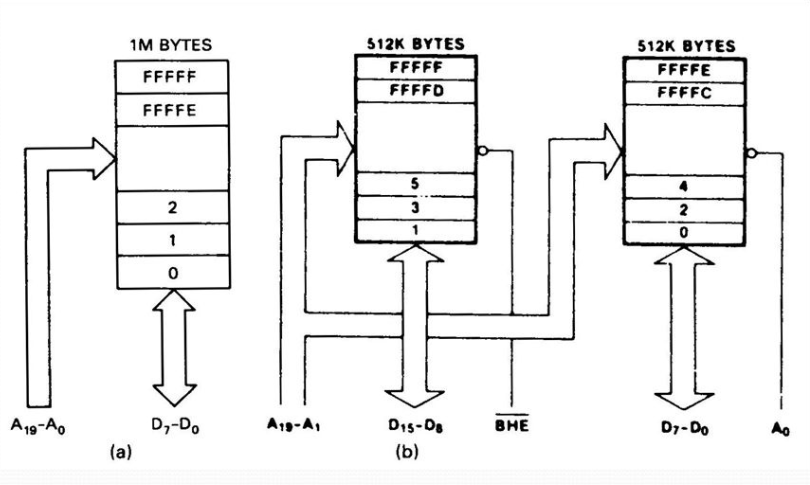
**Ans:** The memory chip is equally divided into two parts(banks). One of the banks contains even addresses called **Even bank** and the other contains odd addresses called **Odd bank**. Even bank always gives lower byte So Even bank is also called **Lower bank** (LB) and Odd bank is also called **Higher bank** (HB). This banking scheme allows to access two aligned memory locations from both banks simultaneously and process 16-bit data transfer. Memory banking doesn’t make it compulsory to transfer 16 bits, it facilitates the 16-bit data transfer.

1. **Distinguish between absolute and partial address decoding.**

**Ans:**

|  |  |
| --- | --- |
| Absolute decoding | Partial decoding |
| Used all the address lines [A0-A7] for decoding of IO devices | Few address lines are not used for decoding IO devices |
| High cost | Less Cost |
| Happen in case of memory interface | Happen in case of memory interface |
| More Hardware is required to design decoding logic | Less hardware is required and sometimes it can be eliminate |
| Used large systems | Used in small syatems |

1. **Discuss how a byte and a word data is read from the odd boundary address and even boundary address of 8086 processor.**

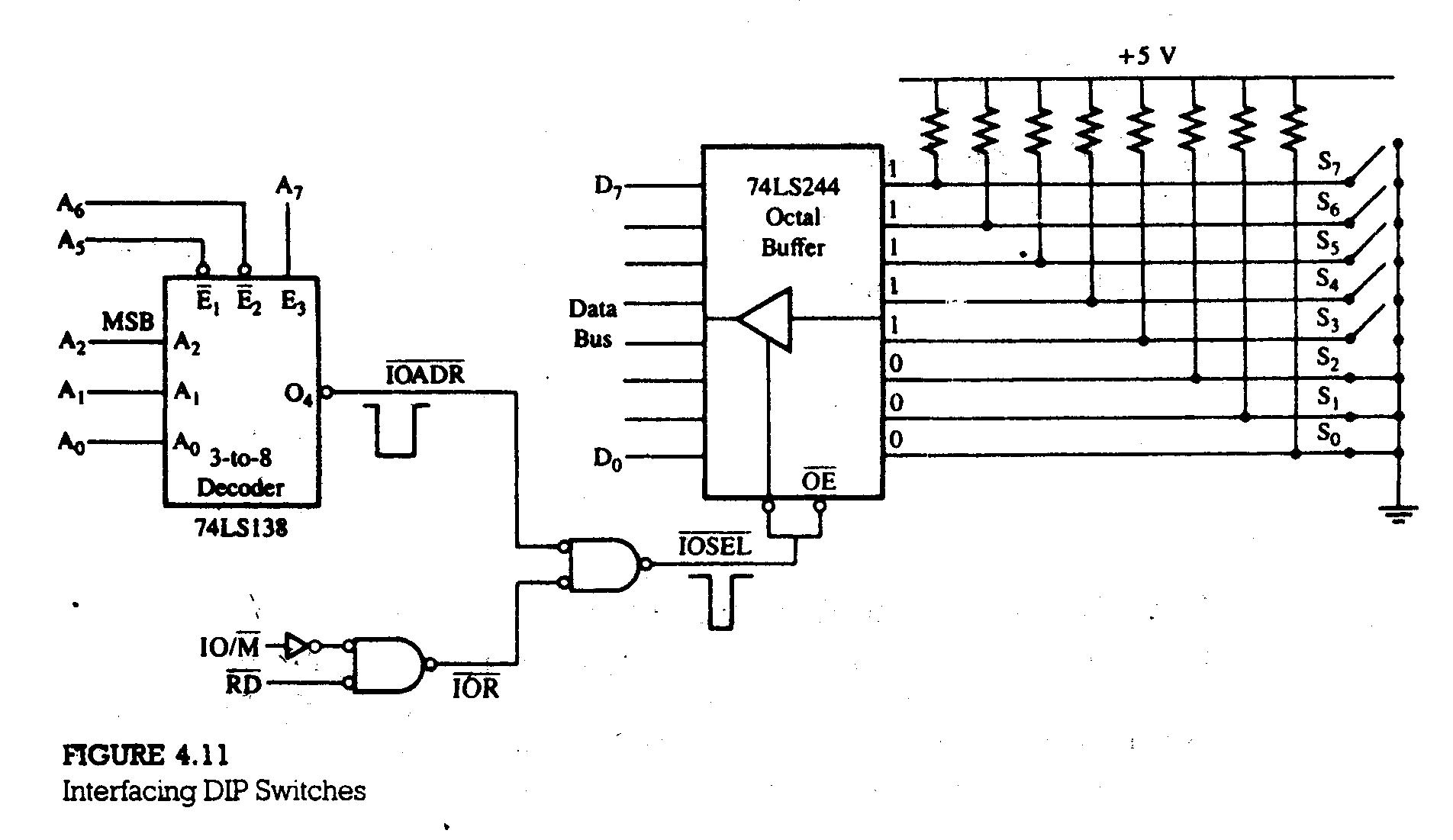
**Ans:** 

* In Figure a memory addresses are not divided into odd and even this figure is given only for comparison.
* Figure b and c show when memory is divided into equal odd and even Bank.
* Since each even and odd bank constitutes 2^19 address, only 19 bits can be used for addressing, a1 to 19 are used for addressing, and a0 is used for enabling / disabling the even bank, another signal, BHE is used for enabling / disabling odd bank. When signal is low a bank is enabled and high bank is disabled.
* A 20-bit address is placed on internal address bus of CPU but a1 to a19 are connected to memory bank for selection of one particular Byte.
* We can conclude from above when even memory address is placed on internal address bus, for example 4 , addressed memory and next memory is addressed (4 and 5) match and when odd memory address is placed on internal address bus , say 5 , addressed memory and the previous memory is addressed (5 and 4) are matched.

1. **Discuss how address buses and data buses of 8086 microprocessor family are multiplexed and how they are de-multiplexed?**
2. **Distinguish between absolute address decoding and partial address decoding. See- 6**
3. **Discuss how device select pulse is generated prior to transfer data in between processor and I/O device.**

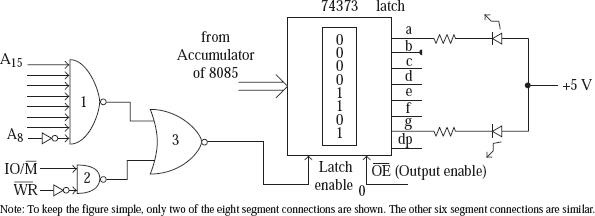
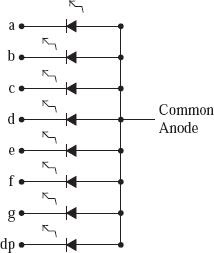
**Ans:**

1. **Draw the circuitry to interface eight DIP switches using a 74LS138 decoder and a 74LS244 buffer.**

**Ans:** The circuit used for interfacing eight DIP switches, as shown in Figure 4.11. The circuit shows the 74LS138 3-to-8 decoder to decode the low-order bus and tri-state octal buffer (74LS244) to interfaces the switches to the data bus. The port can be accessed with the address 84H. The 74LS244 tri-state octal buffer used as an interfacing device. When OE signal goes low, the input data show up on the output lines (connected to the data bus). 

1. **Discuss how to interface common anode seven-segment LED with 8085 with proper circuit diagram. Also write the program to display sequentially digit 0-9 in the seven-segment LED.**

**Ans:**  An output device which is very common is, especially in the kit of 8085 microprocessor and it is the Light Emitting Diode consisting of seven segments. Moreover, we have eight segments in a LED display consisting of 7 segments which includes ‘.’, consisting of character 8 and having a decimal point just next to it. We denote the segments as ‘a, b, c, d, e, f, g, and dp’ where dp signifies ‘.’ which is the decimal point.

****

here are two types of 7-segment LED: They are the common anode type and the common cathode type. We have discussed the common anode-type which is 7 segmented Light Emitting Diode. In the LED which is common anode and is 7-segmented, here we connect all the eight LED anodes together and the eight external pin is brought to display. And this pin gets connected to a DC supply of +5 Volt. The cathode ends of the eight segments are brought out on the pins of the display.

1. **What happens when a microprocessor is interrupted?**

**Ans:**  An interrupt is an event or signal that requests the CPU’s attention. This halt allows peripheral devices to access the microprocessor. Whenever an interrupt occurs, the processor completes the current instruction and starts the implementation of an Interrupt Service Routine (ISR) or Interrupt Handler.

In the 8086 microprocessor following tasks are performed when the microprocessor encounters an interrupt:

1. The value of the flag register is pushed into the stack. It means that first, the value of SP (Stack Pointer) is decremented by two then the value of the flag register is pushed to the memory address of the stack segment.
2. The value of starting memory address of CS (Code Segment) is pushed into the stack.
3. The value of IP (Instruction Pointer) is pushed into the stack.
4. IP is loaded from word location (Interrupt type) \* 04.
5. CS is loaded from the following word location.
6. Interrupt, and Trap flags are reset to 0.
7. **Design an interfacing circuit to interface a 256-byte memory module externally using peripheral mapped I/O. Also write the code to read and write data from this memory module.**

**Ans:**

1. **Define instruction cycle, machine cycle and T-state.**

**Ans:** Time required to execute and fetch an entire instruction is called instruction cycle.

The time required by the microprocessor to complete an operation of accessing memory or input/output devices is called machine cycle.

One time period of frequency of microprocessor is called t-state. A t-state is measured from the falling edge of one clock pulse to the falling edge of the next clock pulse.

1. **Why serial data transfer is mostly preferred over parallel data transfer? Give reasons.**

**Ans:** In Serial Transmission, data-bit flows from one computer to another computer in bi-direction. In this transmission, one-bit flows at one clock pulse. In Serial Transmission, 8 bits are transferred at a time having a start and stop bit.

In Parallel Transmission, many bits are flow together simultaneously from one computer to another computer. Parallel Transmission is faster than serial transmission to transmit the bits. Parallel transmission is used for short distance.

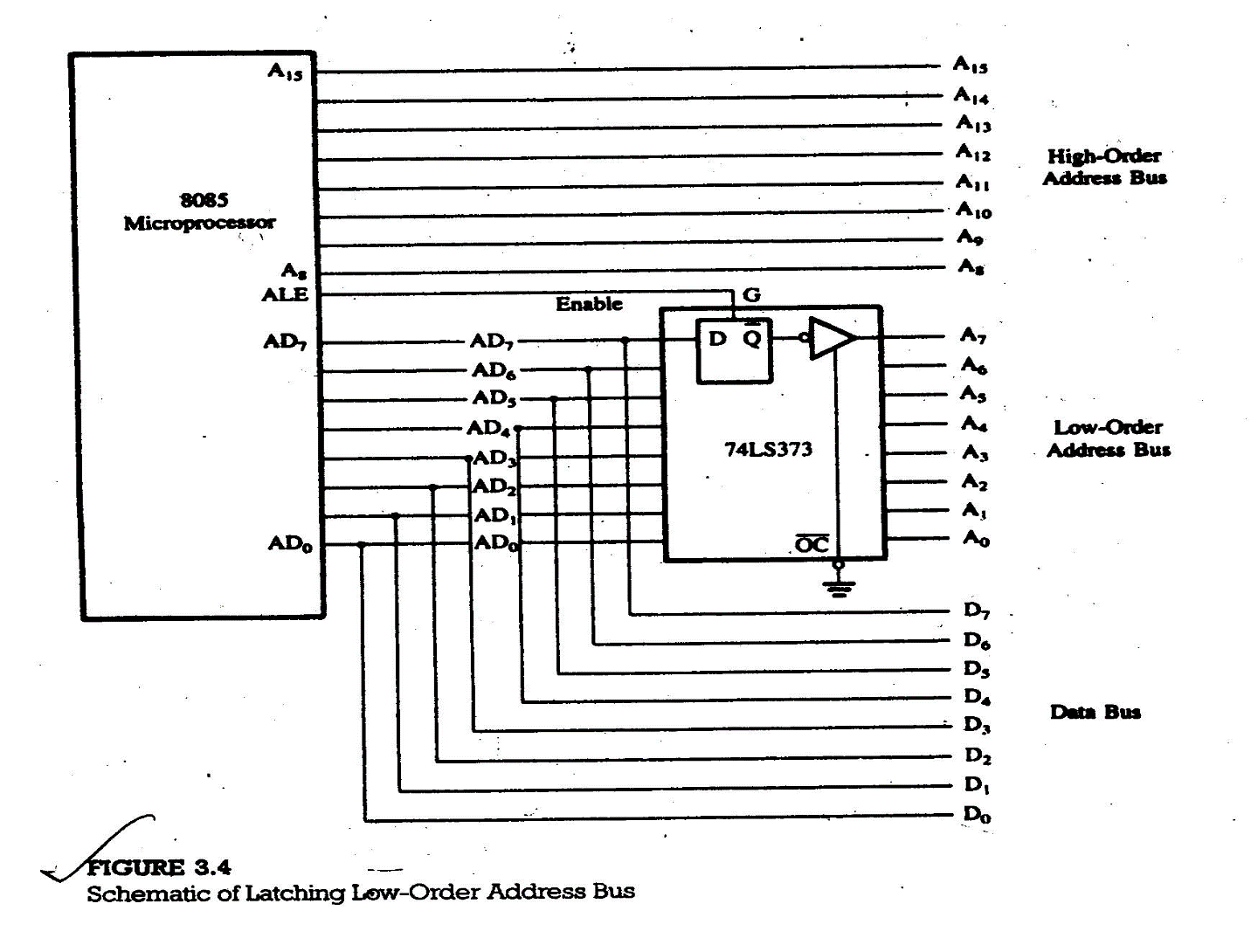
Generally, Serial Transmission is used for long-distance. Serial Transmission is full duplex as sender can send as well as receive the data. Serial transmission is reliable and straightforward.

1. **Mention the functions of READY and HOLD signal of 8086-processor.**

**Ans:**

1. Ready: If the signal of 8085/8080A READY pin is low, the microprocessor enters into a Wait state. The pin is used to synchronize with slower peripherals with microprocessor.
2. Hold: When the HOLD is activated by an external signal, the microprocessor relinquishes control of system bus for external peripheral to use them. For example, HOLD signal is used in DMA data transfer.
3. **What is ALE? Discuss how the low-order address buss of 8086 is demultiplexed using ALE signal.**

**Ans: ALE (Address Latch Enable)** : This is a positive going pulse and generated every time 8085 begins an operation; it indicates that bits on AD7 - AD0 are address bits. This signal are used to latch the low-order address bus an generate a separate set of eight address lines; A7 - A0 . The low-order address (05H) is lost after the first clock period. The address needs to be latched for identifying the memory address. Otherwise, the address 2005H will change to 204FH after the first clock period.

Figure 3.4 shows that the uses of a latch (74LS373) and the ALE signal demultiplexes the low-order bus. 

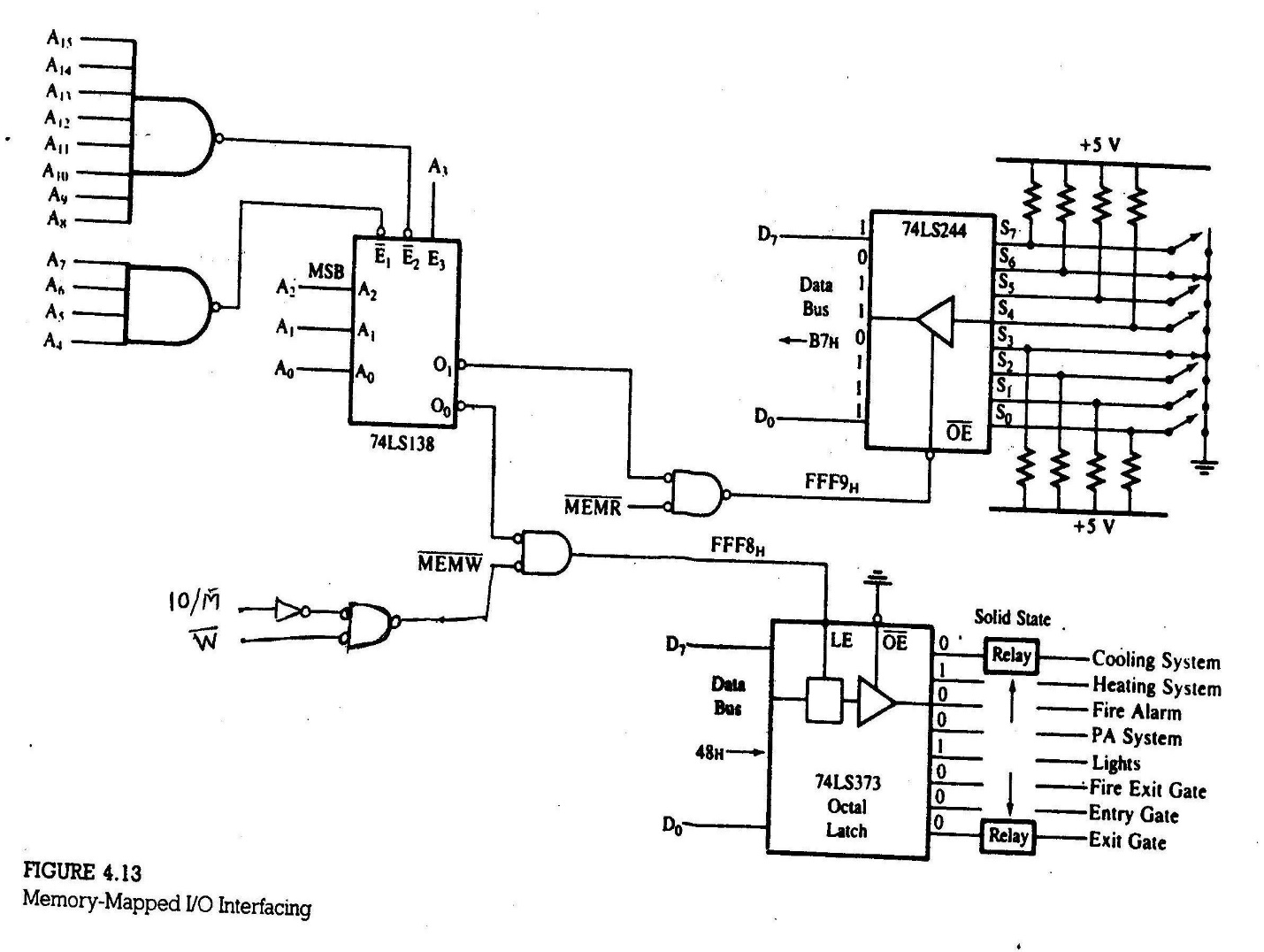
1. **Safety Control System Using Memory-Mapped I/O Technique**

**Ans: The various process control devices are connected to the data bus through the latch 74LS373 and solid state relays. When LE is high, the data enter the latch and when LE goes low, data are latched. The latched data are available on the output lines enabled by OE. To assert the I/O select pulse, the output port address should be FFF8H, as shown below:**

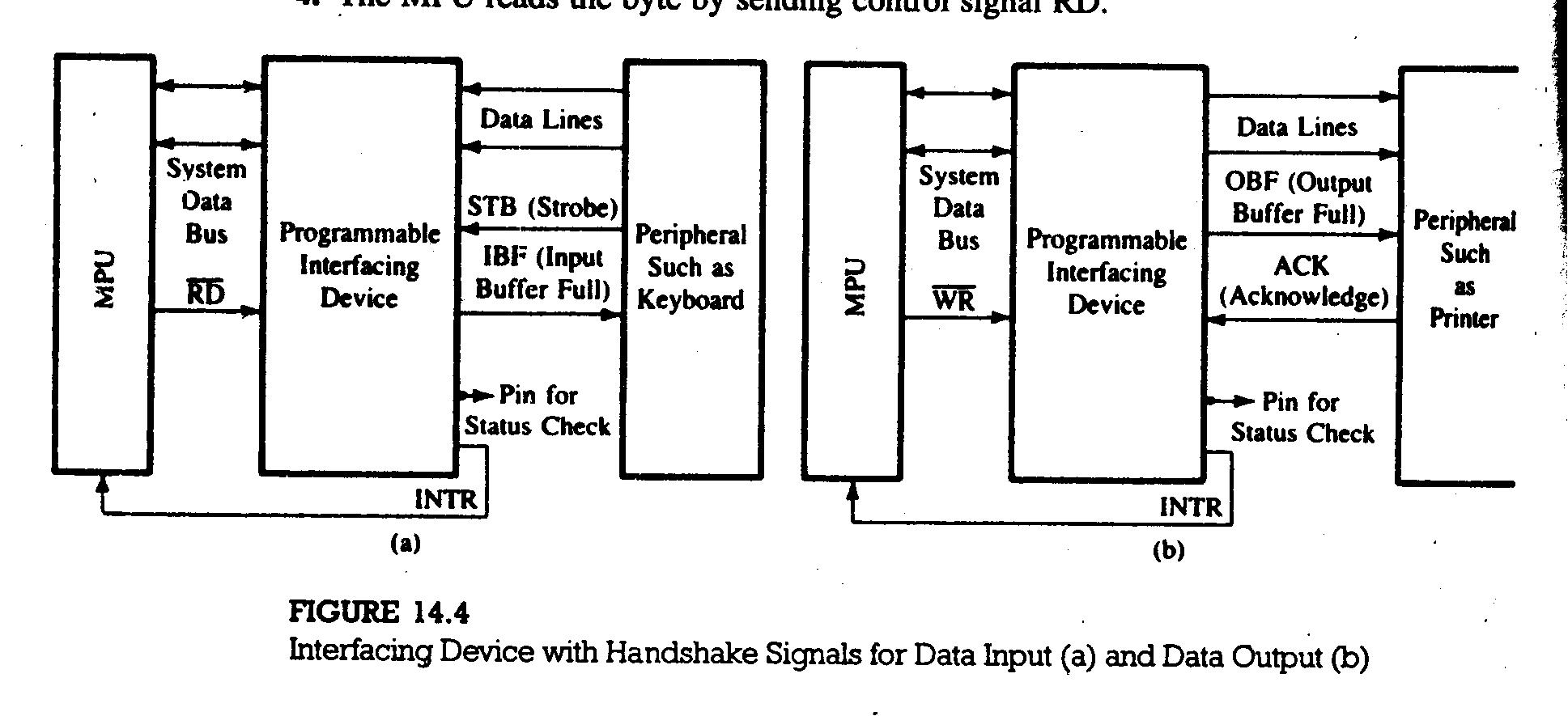
A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0

1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0

= FFF8H

****

1. **Programmable Device with a Handshake Signals**
2. **The MPU and peripherals operate at different speeds; therefore, signals are exchanged prior to data transfer between fast-responding MPU and slow-responding peripherals such as printers and data converters. These signals are called handshake signals. These signals are generally provided by programmable devices.**

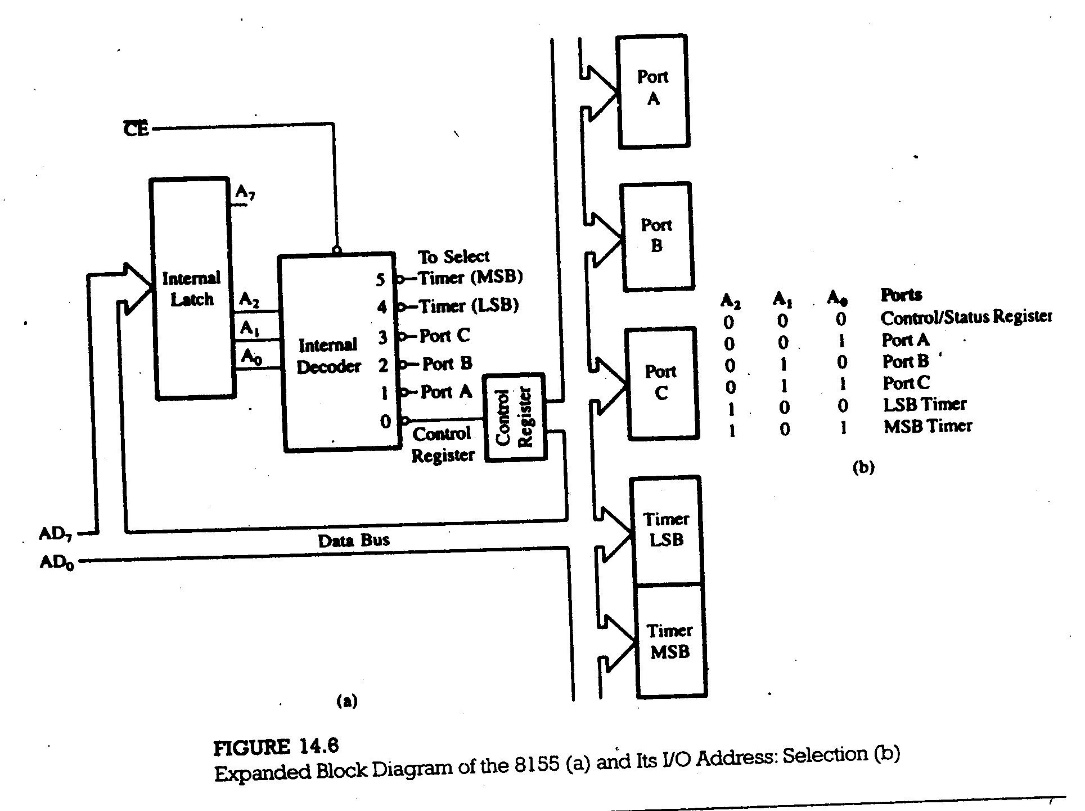
****

1. **Discuss the I/O ports and Timer of 8155.**

**Ans:** The 8155 is a device with two sections: the first is a 256 bytes R/W memory, and the second is a programmable I/O. Functionally, these two sections can be viewed as two independent chips.

The I/O section includes two 8-bit parallel I/O ports (A and B), one 6-bit port (C), and a timer (Figure 14.5). All the ports can be configured simply as input/output ports. Ports A and B also can be programmed in the handshake mode, each port using three signals from port C. The timer is a 14-bit down-counter and has four modes.

THE 8155 I/O PORTS:



The I/O section of the 8155 includes a control register, three I/O ports, and two registers for the timer (Figure 14.6).

To communicate with the peripherals through the 8155 the following steps are necessary:

1. Determine the address (port numbers of the registers and I/O’s) based on

the Chip Enable logic and address lines AD0, AD1, and AD2.

2. Write control word in the control register to specify I/O functions of the ports

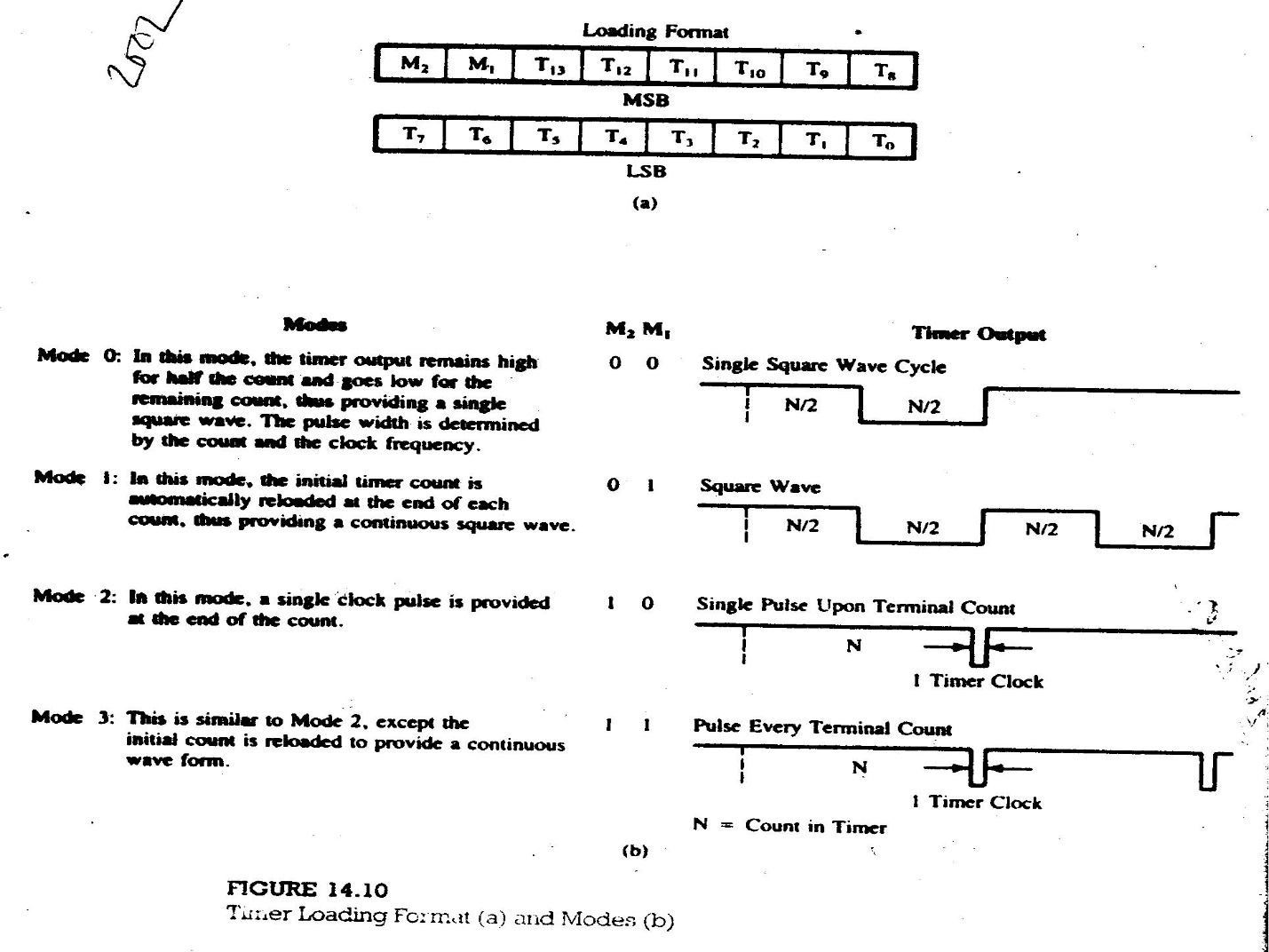
and the timer characteristics.

3. Write I/O instructions to port addresses to communicate with peripherals.

4. Read the status register, if necessary, to verify the status of the I/O ports

and the timer. In simple applications, this step is not necessary.

1. **Design a square-wave generator with a pulse width of 100µs by using the 8155 timer. Set up the timer in mode1 if the clock frequency is 3MHz.**

**Ans: Timer Count:** The pulse width required is 100 µs; therefore, the count should be calculated for the period of 200 µs. The timer output stays high for only half the count. 

Clock Period = 1/f = 1/3 x 106 = 330 ns

Timer Count = Pulse Period / Clock Period = 200 x 10-6 / 330 x 10-9 = 606

Count = 025EH

The port addresses for the timer registers are

Timer LSB = 24H

Timer MSB = 25H

The least significant byte, 5EH (of the count 025EH), should be loaded in the timer register with address 24H. The most significant byte is determined as follows:

M2 M1 T13 T12 T11 T10 T9 T8

0 1 0 0 0 0 1 0 = 42H

Therefore, 42H should be loaded in the timer register with the address 25H.

1. **Discuss the control signals in handshake mode for input and output of 8155 ports with timing waveform.**

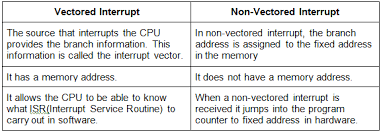
**Ans:**

1. **Difference between Maskable and non-maskable interrupt &**

**Vectored and non-vectored interrupt.**

**Ans:**

| SR.NO. | Maskable Interrupt | Non Maskable Interrupt |
| --- | --- | --- |
| 1 | Maskable interrupt is a hardware Interrupt that can be disabled or ignored by the instructions of CPU. | A non-maskable interrupt is a hardware interrupt that cannot be disabled or ignored by the instructions of CPU. |
| 2 | When maskable interrupt occur, it can be handled after executing the current instruction. | When non-maskable interrupts occur, the current instructions and status are stored in stack for the CPU to handle the interrupt. |
| 3 | Maskable interrupts help to handle lower priority tasks. | Non-maskable interrupt help to handle higher priority tasks such as watchdog timer. |
| 4 | Maskable interrupts used to interface with peripheral device. | Non maskable interrupt used for emergency purpose e.g power failure, smoke detector etc . |
| 5 | In maskable interrupts, response time is high. | In non maskable interrupts, response time is low. |
| 6 | It may be vectored or non-vectored. | All are vectored interrupts. |
| 7 | Operation can be masked or made pending. | Operation Cannot be masked or made pending. |
| 8 | RST6.5, RST7.5, and RST5.5 of 8085 are some common examples of maskable Interrupts. | Trap of 8085 microprocessor is an example for non-maskable interrupt. |

****

1. **8085 VECTORED INTERRUPTS**

**Ans :**

**12.2 8085 VECTORED INTERRUPTS**

**The 8085 has five interrupt inputs (Figure 12.5). One is called INTR, three are called RST 5.5, RST 6.5, and RST 7.5 respectively, and the fifth is called TRAP, a nonmaskable interrupt. These last four (RSTs and TRAP) are automatically vectored (transferred) to a specific locations on memory page 00H without any external hardware. The do not require the INTA signal or an input port; the necessary hardware is already implemented inside the 8085. These interrupts and their call locations are as follows:**

**Interrupts Call Locations**

**1. TRAP 0024H**

**2. RST 7.5 003CH**

**3. RST 6.5 0034H**

**4. RST 5.5 002CH**

**The TRAP has highest priority, followed by RST 7.5, 6.5, 5.5, and INTR, in that order; however, the TRAP has lower priority than the Hold signal used for DMA.**

1. ***Can the microprocessor be interrupted again before the completion of the first interrupt service routine?***

**Ans: The answer to this question is determined by the programmer. After the first interrupt, the interrupt process is automatically disabled. In the Illustrative program in section 12.12, the service routine enables the interrupt at the end of the service routine; in this case, the microprocessor cannot be interrupted before the completion of this routine. If the instruction EI is written at the beginning of the routine, the microprocessor can be interrupted again during the service routine.**

1. **Time Delay Using a Loop within a Loop Technique**

**Ans: Label Opcode Operand T-states**

**MVI B,38H 7**

**LOOP2: MVI C,FFH 7**

**LOOP1: DCR C 4**

**JNZ LOOP1 10/7**

**DCR B 4**

**JNZ LOOP2 10/7**

**DELAY CALCULATIONS**

**The delay in LOOP1 is TL1 = 1783.5 µs. The counter value of LOOP2 is 5610 (38H). Delay for the LOOP2 is calculated as follows:**

**TL2 = 56 (TL1 + 21 T-states x 0.5 µs)**

**= 56 (1783.5 µs + 10.5 µs)**

**= 100.46 ms**

1. **Time Delay Using a Register Pair**

**Ans: Here the counter value can be a 16-bit number and maximum of FFFFH.**

**Label Opcode Operand T-states**

**LXI B,2384H 10**

**LOOP: DCX B 6**

**MOV A,C 4**

**ORA B 4**

**JNZ LOOP 10/7**

**TIME DELAY**

**The loop includes 24 clock periods for execution. The decimal equivalent of counter value is**

**2384H = 909210**

**If the clock period of the system = 0.5 µs, the delay in the loop TL is**

**TL = (T x Loop T-states x N10 )**

**= (0.5 x 24 x 909210)**

**≈ 109 ms (without adjusting for the last cycle)**

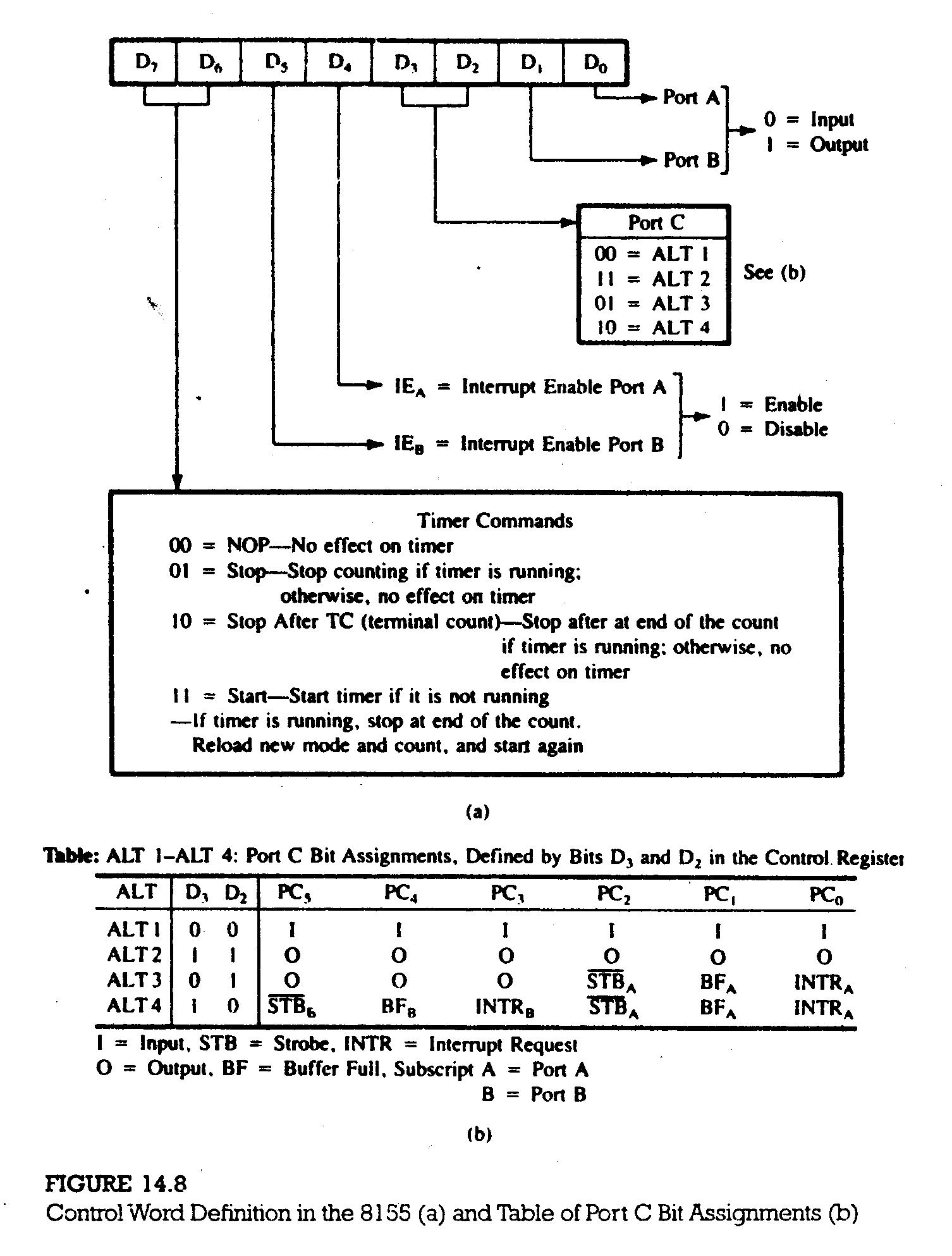
**Total Delay TD = 109 ms + TO**

**≈ 109 ms (the instruction LXI adds only 5 µs)**

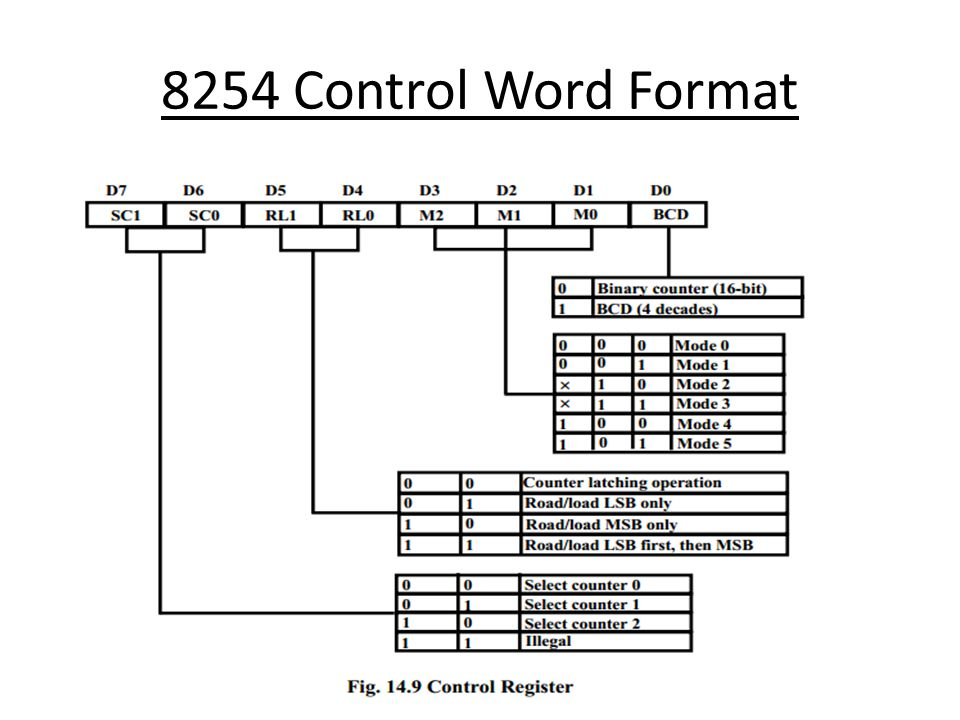
1. **What is timer? Write the control word format of 8253/8155 programmable interval timer.**

**Ans: CONTROL WORD 8155**

**The I/O ports and the timer can be configured by writing a control word in the control register. Bit D2 and D3 determine the functions of port C; their combination specifies one of the four alternatives, from simple I/O to interrupt I/O, as shown in Figure 14.8(b). Bits D4 and D5 are used only in the interrupt mode to enable or disable internal flip-flops of the 8155. These bits do not have any effect on the Internal Enable (INTE) flip-flop of the MPU.**

****

**8253/8254**

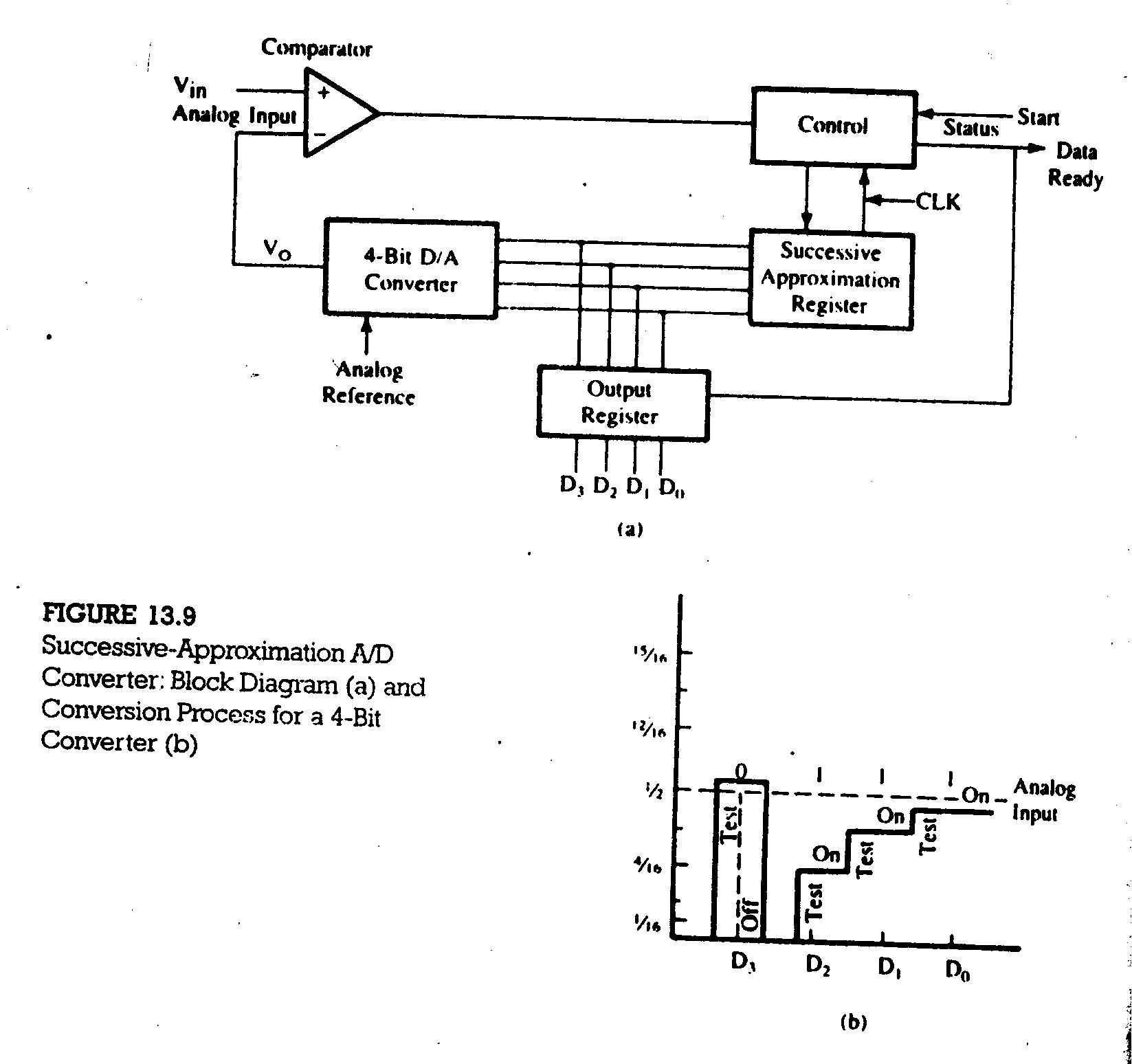
****

1. **Discuss about successive-approximation A/D converter.**

**Ans: Figure 13.9(a) shows the block diagram of a successive approximation A/D converter includes three major elements: the D/A converter, the successive approximation register (SAR), and the comparator. The conversion technique involves the output of the D/A converter Vo with the analog input signal Vin. The digital input to the DAC is generated using the successive-approximation method (explain below). When the DAC output matches the analog signal, the input to the DAC is equivalent digital signal.**

**In the case of a 4-bit A/D converter, bit D3 is turned on first and the output of the DAC is compared with an analog signal. If the comparator changes the state, indicating that the output generated by D3 is larger than the analog signal, bit D3 is turned off in the SAR and bit D2 is turned on. The process continues until the input reaches bit D0.**

**Figure 13.9(b) illustrates a 4-bit conversion process. When bit D3 is turned on, the output exceeds the analog signal and therefore, D3 is turned off. When the next three successive bits are turned on, the output becomes approximately equal to the analog signal.**

****

1. **Draw and describe the functional diagram of 8257 DMA controller.**

**Ans:**

1. **What is DMA? What are uses of DMA?**
2. **Discuss the role of RS232 interface in detail?**
3. **Draw and discuss the block diagram of 8254 programmable interval timer.**
4. **Design a square-wave generator with a pulse width of 100 µs by using the 8155 timer. Set up timer in Mode 1 if the clock frequency is 3 MHz.**

**Ans:**