Input-Output Interface

Input-Output Device Interface

- Using I/O devices data can be transferred between the microprocessor and the outside world.
- This can be done in groups of 8 bits using the entire data bus. This is called parallel I/O.
- The other method is serial I/O where one bit is transferred at a time using the SID and SOD pins on the Microprocessor.
- Memory address space- 1,048,576 bytes long (1M-byte): 00000H-FFFFFH
- Input/output address space- 65,536 bytes long (64K-bytes):0000H-FFFFH
- I/O ports are 8 bits in width a 16-bit port is actually two consecutive 8-bit ports being addressed a 32-bit I/O port is actually four 8-bit ports.
- I/O devices can be interfaced in two ways
    - ❑Isolated mapped I/O
    - ❑Memory mapped I/O

Isolated Input/output

- It treats them separately from memory.
- I/O devices are assigned a "port number" within the 8-bit address range of 00H to FFH.
- The 16-bit address is called a ***variable address*** because it is stored in DX, and then used to address I/O device.
- The user in this case would access these devices using the IN and OUT instructions only.
- Advantages of isolated I/O
  - ❑ Complete memory address space available for use by memory.
  - ❑ Special instructions have been provided in the instruction set of the 8086 to perform isolated I/O operation. This instructions tailored to maximize performance.
- Disadvantage of Isolated I/O
  - ❑ All inputs/outputs must take place between an I/O port and accumulator (AL or AX or EAX) register

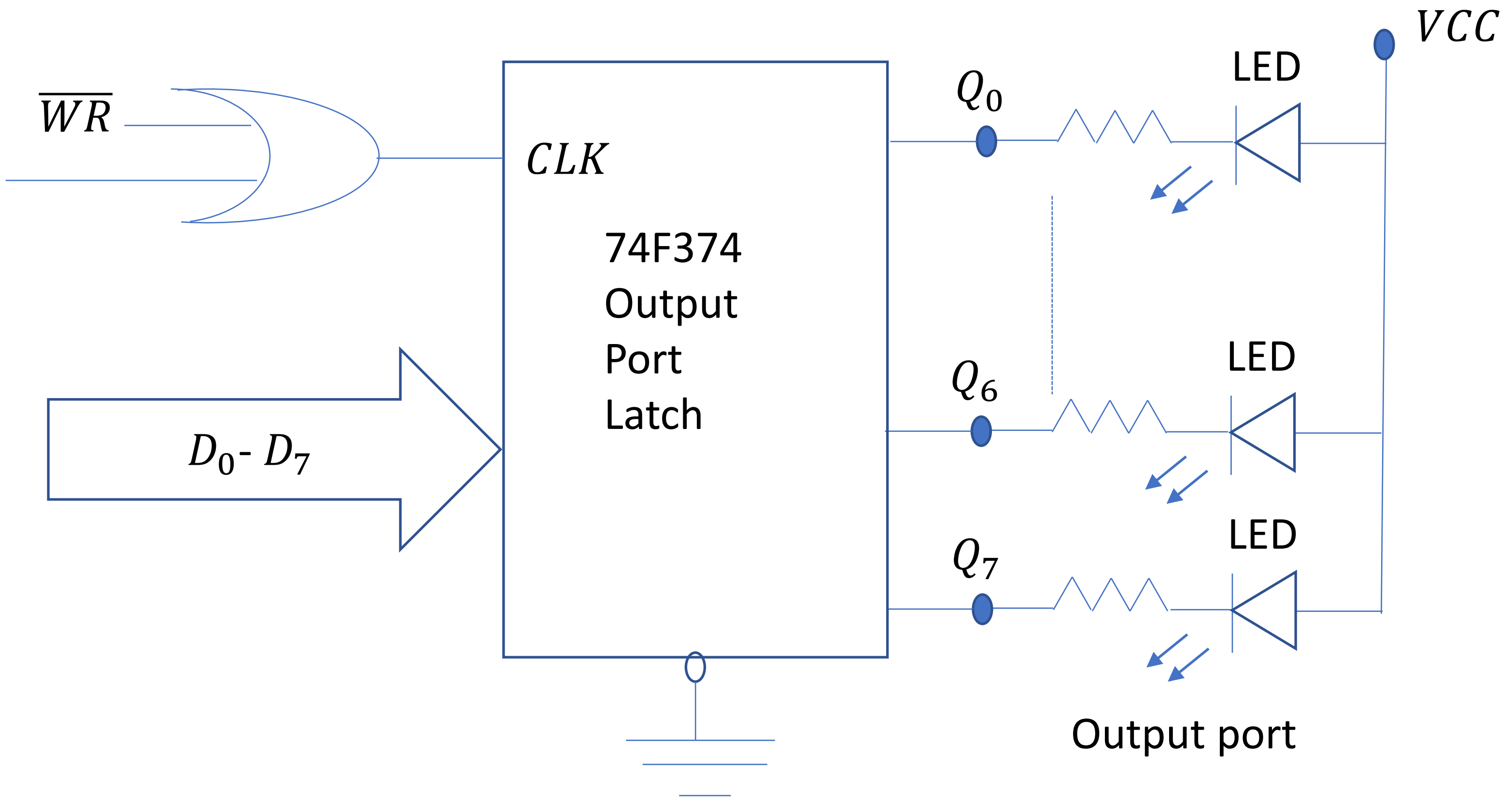
Memory Mapped Input Output

- It considers them like any other memory location.
- They are assigned a 16-bit address within the address range of the 8086.
- The exchange of data with these devices follows the transfer of data with memory. The user uses the same instructions used for memory.
- Advantages of memory mapped input output
    - ❑Simpler decoding circuits, no special instructions required.
- disadvantages of memory mapped input output
    - ❑A portion of the memory system is used as the I/O map, reducing the memory available to applications.

Isolated I/O	Memory Mapped I/O
Isolated I/O used separated memory space	Memory mapped I/O uses memory from the main memory
Limited instructions can be used such IN, OUT, INS, OUTS	Any instructions which references to memory can be used. (MOV, AND, XCHG, SUB, ….)
Faster because I/O instruction is specifically designed to run faster than memory instruction	Slower because memory instruction execute slower than the special I/O instructions
The memory address space is not affected	Part of the memory address space is lost
The addresses for isolated I/O devices are called ports	Memory mapped I/O devices are treated as memory locations on the memory map.

The interfacing of the output devices

- Output devices are usually slow. Also, the output is usually expected to continue appearing on the output device for a long period of time.
- Given that the data will only be present on the data lines for a very short period (microseconds), it has to be latched externally.
- To do this the external latch should be enabled when the port's address is present on the address bus, the $IO/\overline{M}$ signal is set high and $\overline{WR}$ is set low.
- The resulting signal would be active when the output device is being accessed by the microprocessor.
- Decoding the address bus (for memory-mapped devices) follows the same techniques discussed in interfacing memory.



$\overline{WR}$
CLK
74F374
Output
Port
Latch
D_0- D_7
Q_0
Q_6
Q_7
LED
LED
LED
VCC
Output port

Figure (1)

The interfacing of Input Devices

- The basic concepts are similar to interfacing of output devices.
- The address lines are decoded to generate a signal that is active when the particular port is being accessed.
- An $\overline{IORD}$ signal is generated by combining the $IO/\overline{M}$ and the $\overline{RD}$ signals from the microprocessor.
- A tri-state buffer is used to connect the input device to the data bus.
- The control (Enable) for these buffers is connected to the result of combining the address signal and the signal $\overline{IORD}$.

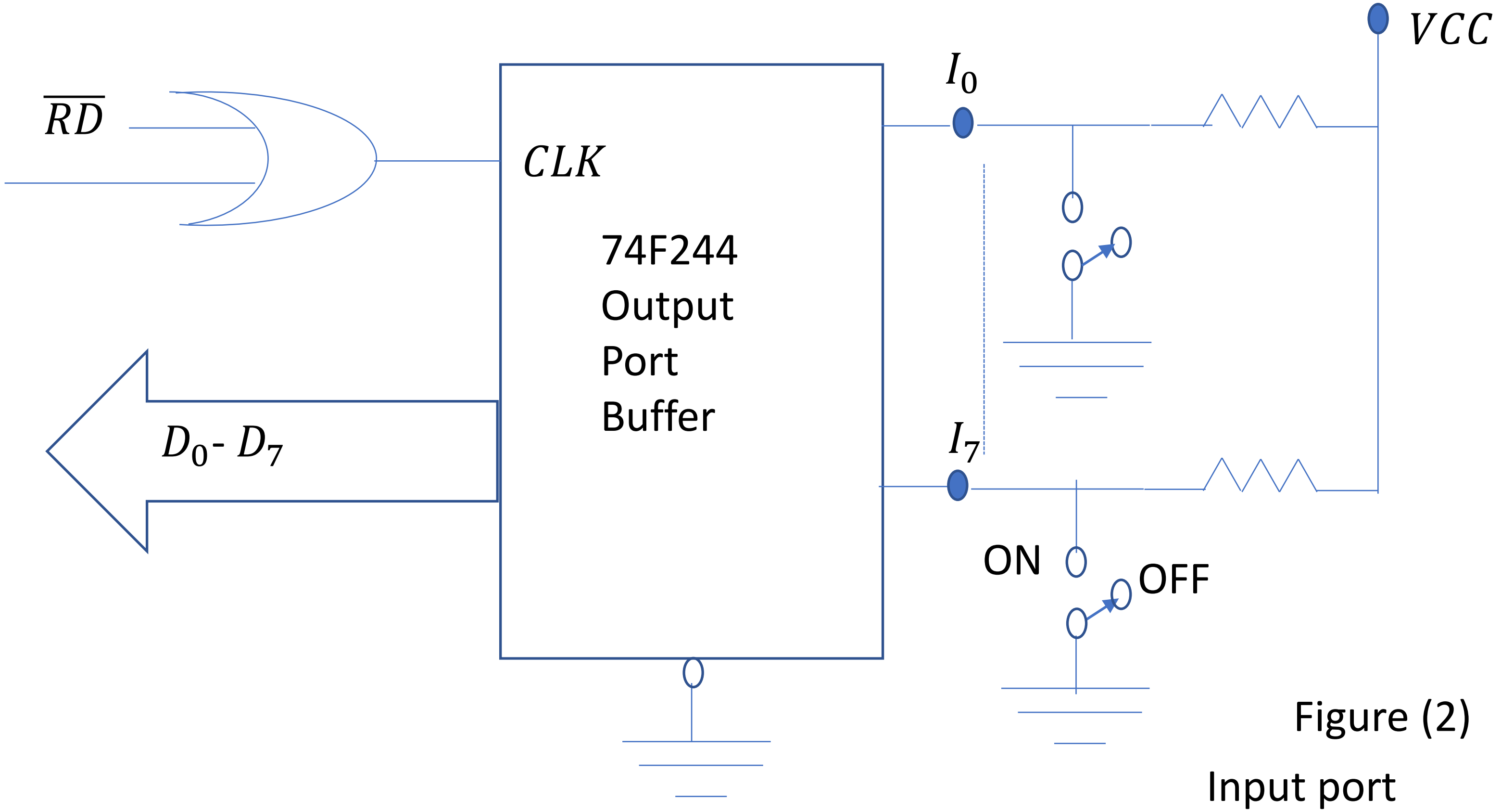


Figure (2)

Input port

Port Address

$A_7A_6A_5A_4 \;\; A_3A_2A_1A_0$

1 1 1 1 0 X X X

1 1 1 1 0 0 0 0 F0H

1 1 1 1 0 1 1 1 F7H

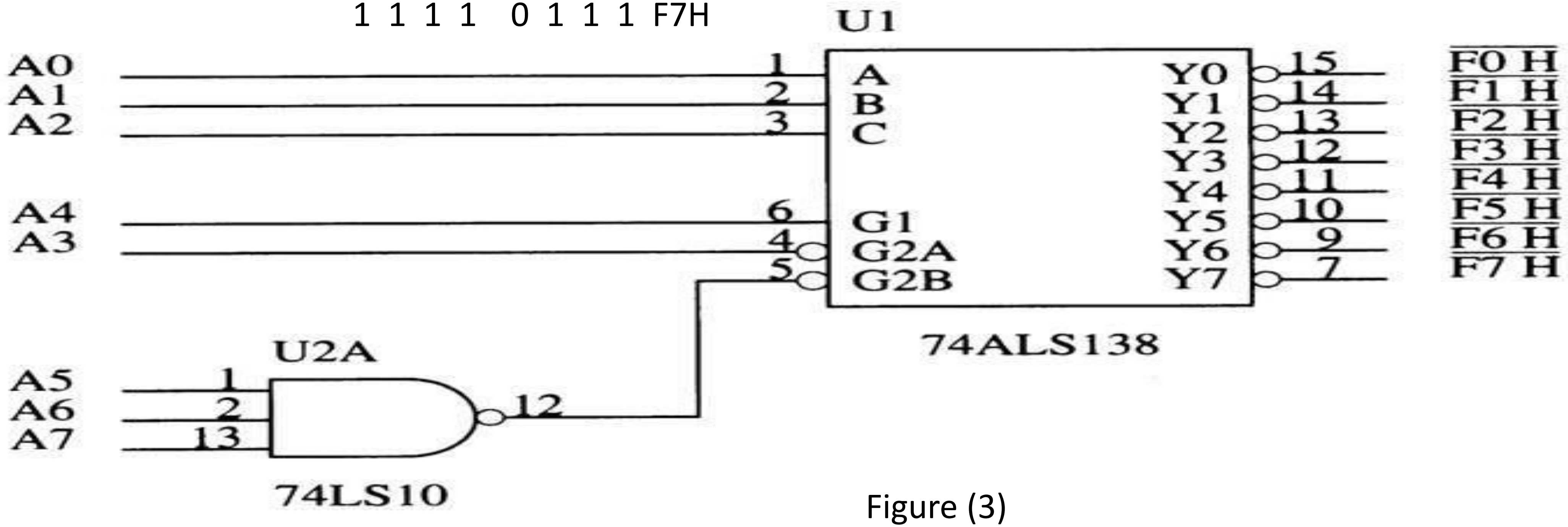


Figure (3)

- Figure (3)shows a 74ALS138 decoder that decodes 8-bit I/O ports F0H -F7H. –identical to a memory address decoder except we only connect address bits A7–A0 to the inputs of the decoder.
- The output of decoder varies from 000 to 111. So the address of ports is varied from F0H to F7H. A3 is always 0, A4, A5, A6, A7 always 1.

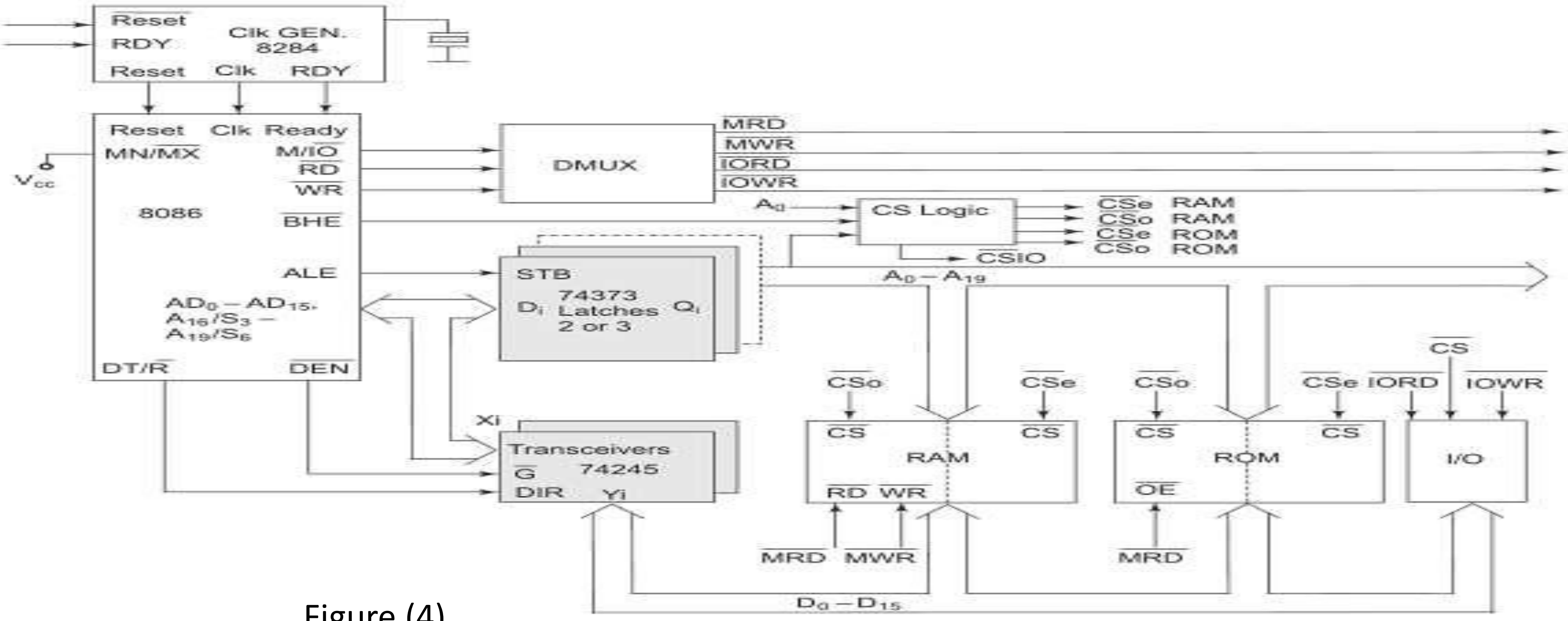


Figure (4)
Minimum Mode

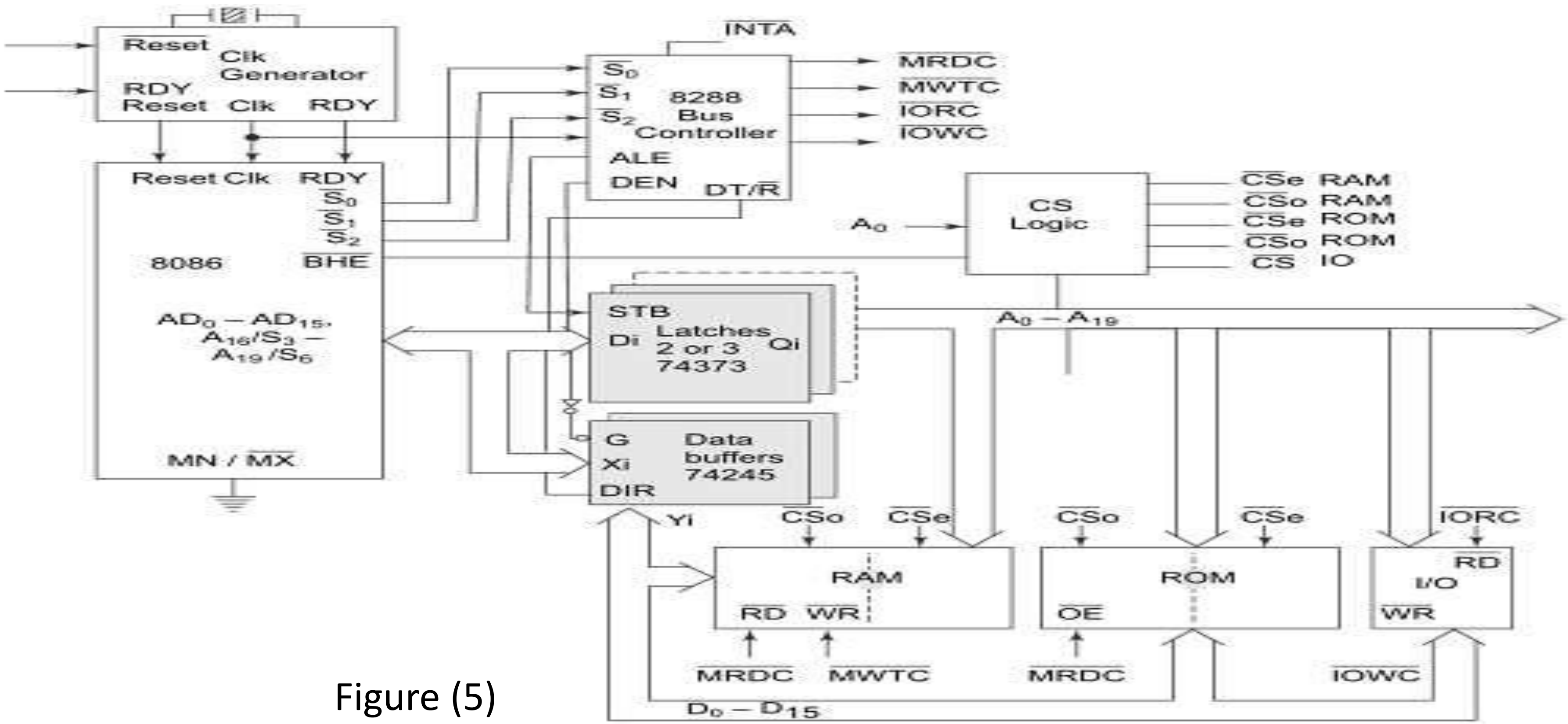


Figure (5)
Maximum Mode

Input / Output Instructions

Instruction	Data width	Function
IN AL, p8	8	A byte is input into AL from port p8
IN AX, p8	16	A word is input into AX from port p8
IN EAX, p8	32	A doubleword is input into EAX from port p8
IN AL, DX	8	A byte is input into AL from the port addressed by DX
IN AX, DX	16	A word is input into AX from the port addressed by DX
IN EAX, DX	32	A doubleword is input into EAX from the port addressed by DX
OUT p8, AL	8	A byte is output from AL into port p8
OUT p8, AX	16	A word is output from AL into port p8
OUT p8, EAX	32	A doubleword is output from EAX into port p8
OUT DX, AL	8	A byte is output from AL into the port addressed by DX
OUT DX, AX	16	A word is output from AX into the port addressed by DX
OUT DX, EAX	32	A doubleword is output from EAX into the port addressed by DX

Example:-

- IN AL,0C8H ;Input a byte from port 0C8H to AL
- MOV BL,AL; Store the value of AL
- IN AX, 34H ;Input a word (two byte) from port 34H, 35H to AX
- OUT 2CH,AX ;Copy the contents of the AX to port 2CH, 2DH
- MOV AL,BL ;Copy the value of BL to AL
- OUT 3BH, AL ;Copy the contents of the AL to port 3Bh
- MOV DX, 0FF78H ;Initialize DX point to port
- IN AL, DX ;Input a byte from a 8 bit port 0FF78H to AL
- IN AX, DX ;Input a word from 16 bit port to 0FF78H,0FF79H to AX.