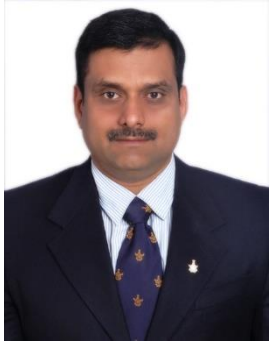
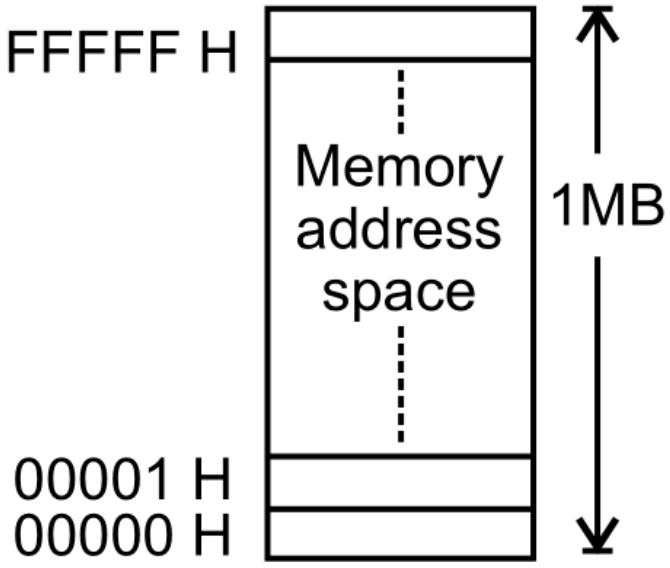
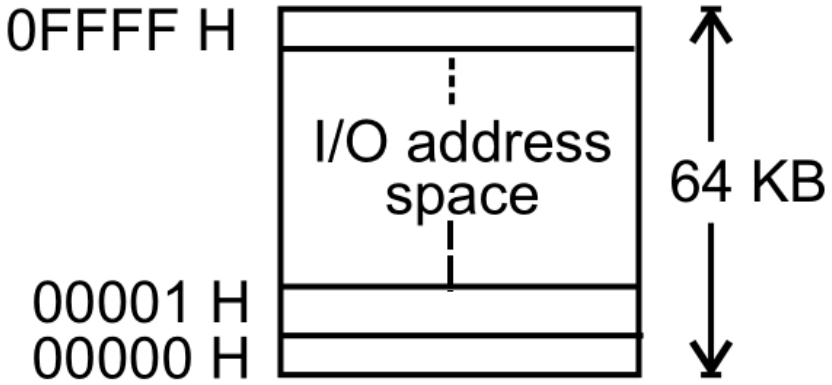
Microprocessors and Interfaces: 2021-22
Lecture 26 :
I/O Interfacing

By Dr. Sanjay Vidhyadharan



IO-Mapped & Memory-Mapped





IO-Mapped & Memory-Mapped

Isolated I/O	Memory mapped I/O
1. I/O devices are treated separate from memory.	1. I/O devices are treated as part of memory.
2. Full 1 MB address space is available for use as memory.	2. Full 1 MB cannot be used as memory since I/O devices are treated as part of memory.
3. Separate instructions are provided in the instruction set to perform isolated I/O input-output operations. These maximise I/O operations.	3. No separate instructions are needed in this case to perform memory mapped I/O operations. Hence, the advantage is that many instructions and addressing modes are available for I/O operations.
4. Data transfer takes place between I/O port and AL or AX register only. This is certainly a disadvantage.	4. No such restriction in this case. Data transfer can take place between I/O port and any internal register. Here, the disadvantage is that it somewhat slows the I/O operations.

Modes of I/O Instructions

- Direct I/O
- Indirect I/O
- String

8086 I/O Instructions

IN and OUT transfer data between an I/O device and the microprocessor's accumulator (AL, AX or EAX).

The I/O address is stored in:

Register DX as a 16-bit I/O address (variable addressing/**Indirect)**.
The byte, p8, immediately following the opcode (fixed address/**Direct**).

IN AL,19H; 8-bits data are saved to AL from I/O port 19H
IN AL, DX; 8-bits data are saved to AL from I/O port [DX]
IN AX, DX; 16-bits are saved to AX.
IN AX, 20H 16-bits data are saved to AX from I/O port 20H
OUT DX, AX; 16-bits are written to port DX from AX
OUT 19H, AL; 8-bits are written to I/O port 0019H.
Only 16-bits (A0 to A15) are decoded.
Address connections above A15 are undefined for I/O instructions.
OUTS

80x86 I/O Instructions

INSB ; inputs a byte from the I/O port specified in DX and stores it at [ES:DI] . It then increments or decrements (depending on the direction flag: increments if the flag is clear, decrements if it is set) DI.

INSW

INSD

OUTSB ; Output byte from memory location specified in DS:(E)SI or RSI to I/O port specified in DX**. It then increments or decrements (depending on the direction flag: increments if the flag is clear, decrements if it is set) SI.

OUTSW

OUTSD

STRING : INS and OUTS, found except the 8086/8088

80x86 I/O 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
INSB	8	A byte is input from the port addressed by DX and stored into the extra segment memory location addressed by DI, then DI = DI ± 1
INSW	16	A word is input from the port addressed by DX and stored into the extra segment memory location addressed by DI, then DI = DI ± 2
INSD	32	A doubleword is input from the port addressed by DX and stored into the extra segment memory location addressed by DI, then DI = DI ± 4
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
OUTSB	8	A byte is output from the data segment memory location addressed by SI into the port addressed by DX, then SI = SI ± 1
OUTSW	16	A word is output from the data segment memory location addressed by SI into the port addressed by DX, then SI = SI ± 2
OUTSD	32	A doubleword is output from the data segment memory location addressed by SI into the port addressed by DX, then SI = SI ± 4

Modes of I/O Instructions

- Direct I/O – the port address is one of the operands.
  - Address must be 00-FFh.
    - IN AL, 27h
  - Data flows through the accumulator
    - MOV AX, BX
    - OUT 26h, AX ; move 16-bit data from AX to port
      ; 26h (AL to 26h and AH to 27h)

Modes of I/O Instructions

- Indirect I/O – the port address is preloaded into DX
  - Address can be 0000-FFFFh

- String I/O – allows data to pass directly through the accumulator (from I/O device to memory)

I/O Interface

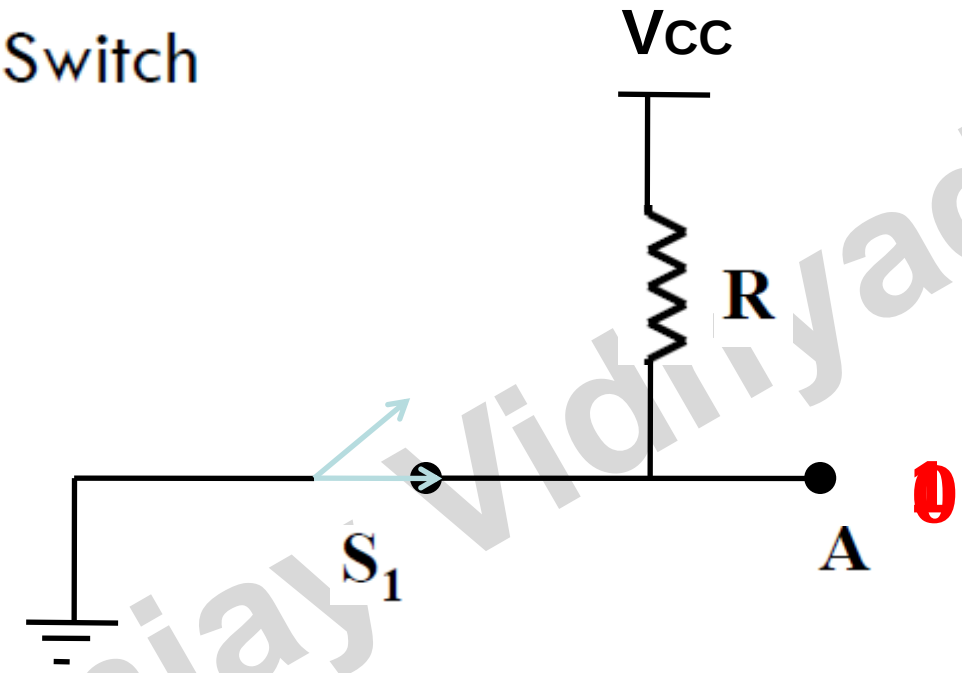
- I/O devices connect to processor through PORTS

- Ports are:
  - registers (part of the I/O interface)
  - 8, 16, or 32 bits wide
  - Addressed in the range 0000-FFFFh
  - Accessed with 2 instructions – IN, OUT

I/O Interfacing

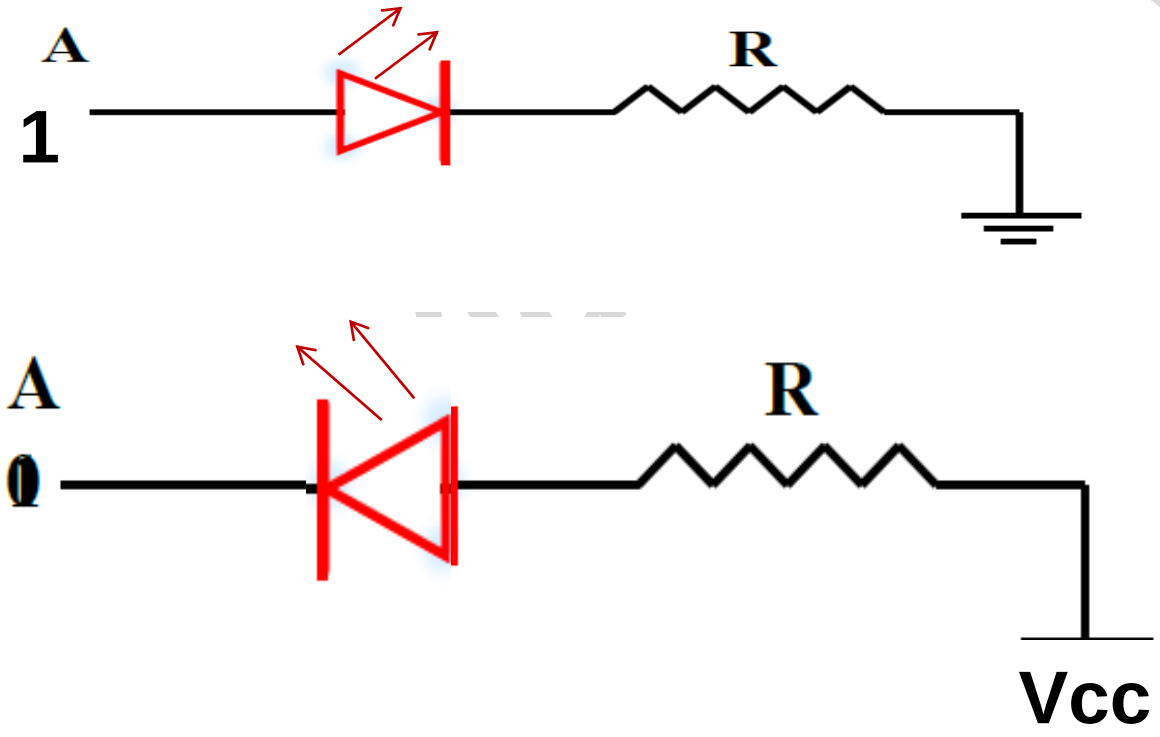
Input Device

- E.g. Switch



I/O Interfacing

Output Device
- E.g. LED



Why Buffers

Input devices must be isolated from the global data bus

Else unwanted data (garbage) may be transferred on to the data bus

Tri-state buffers – provide isolation as well as strengthen the signal

I/O Design in 8086

Any μP-based system when data is sent out by μP, the data on the data-bus must be latched by the receiver/output device

Memories have internal latches–store data

Latching System must be designed for ports

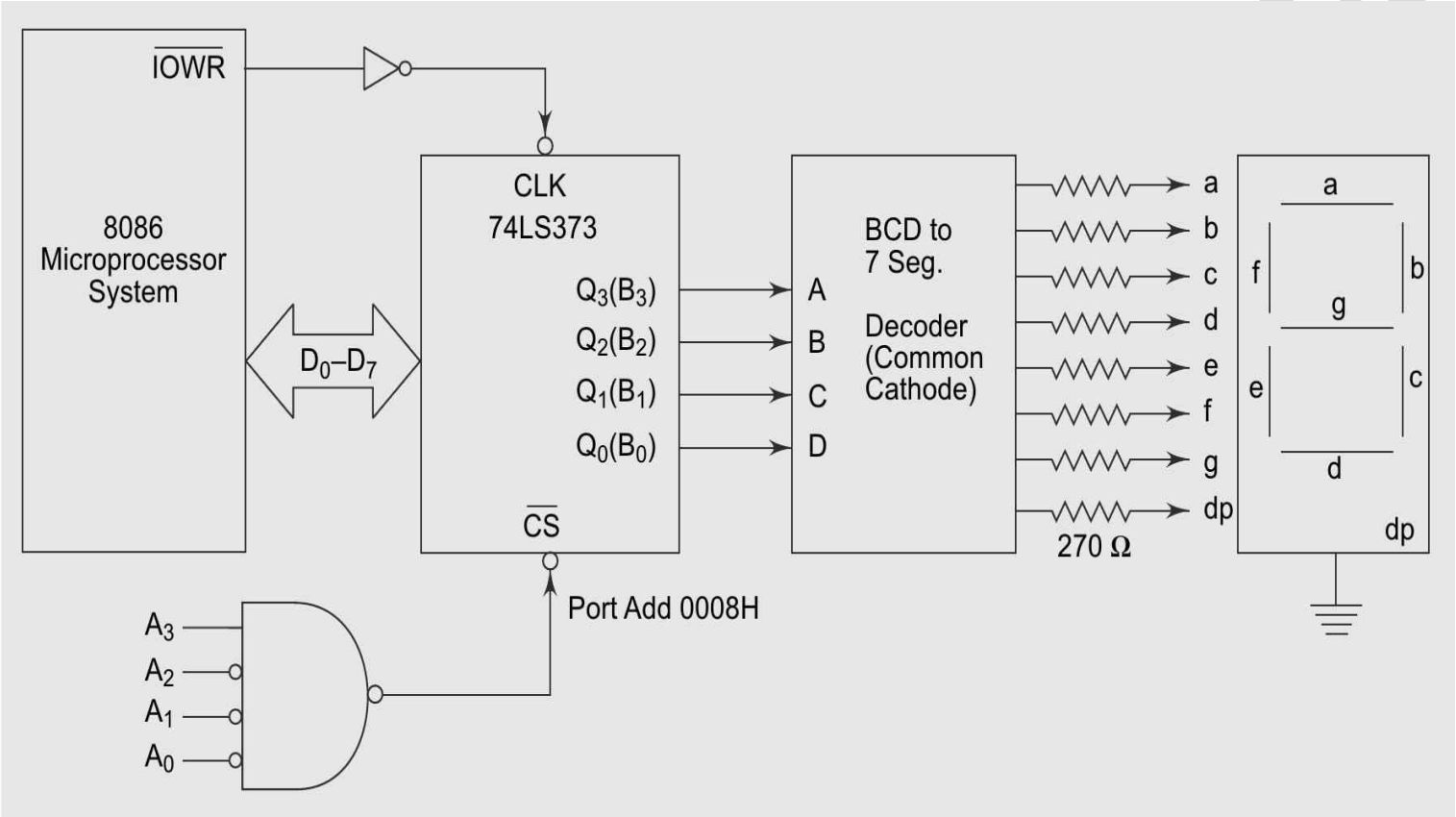
Data provided by the μP is available only for short period of time (50-1000ns) data must be latched else it will be lost

When data comes in from a port/memory, data must be input through a tri-state buffer

I/O Design in 8086

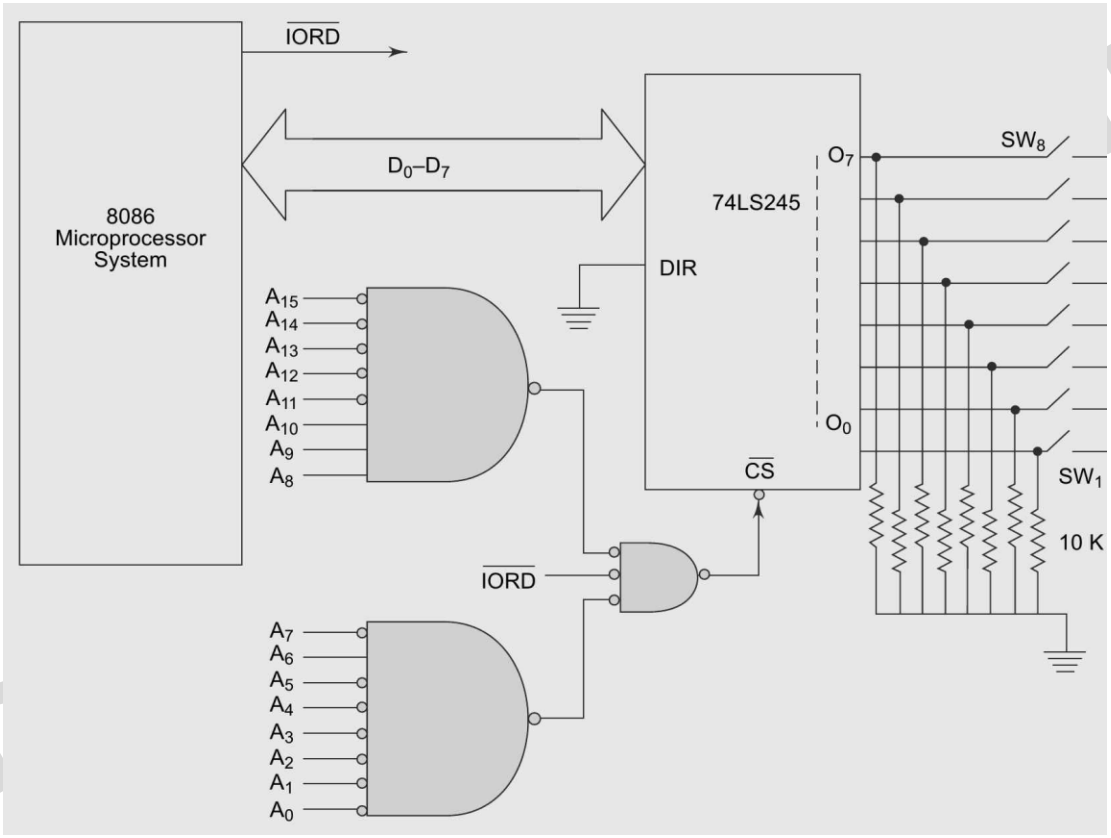
- Interfacing *input devices* like switches require *buffers*.
- Interfacing *output devices* like LEDs require *latches*.
- **Programmable Peripheral Interface (PPI)** device provides these features

Simple Output Port



74373 : Latch

Simple Input Port



74245 : Trans-receive Tristate Buffer

Simple Input & Output Ports

Simple Input & Output Ports

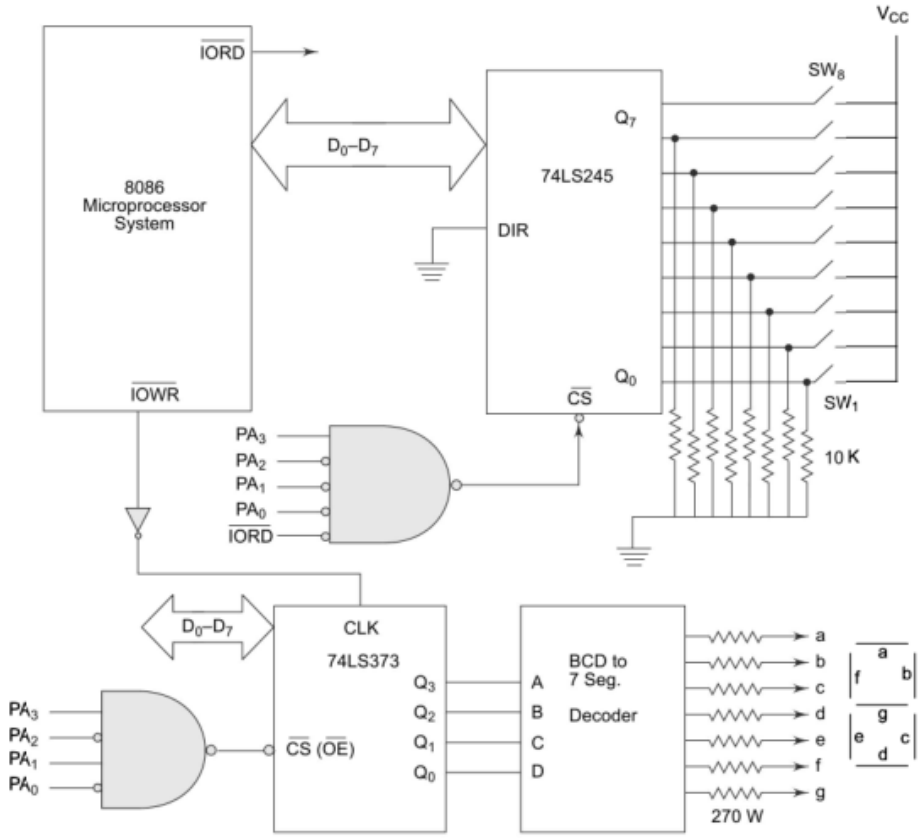
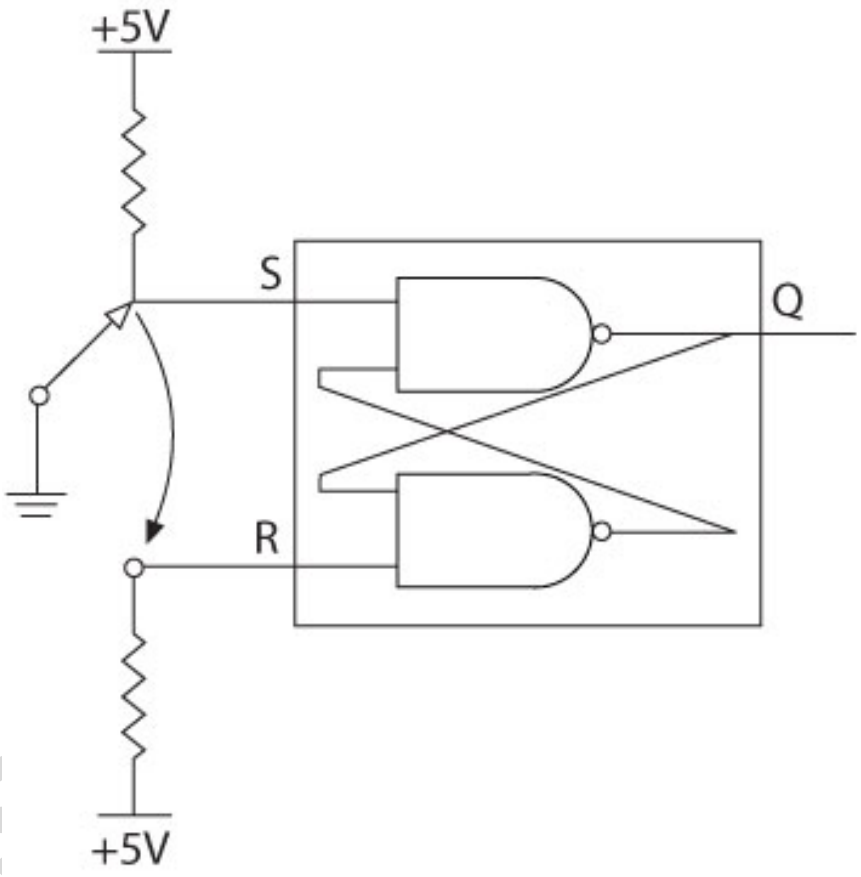
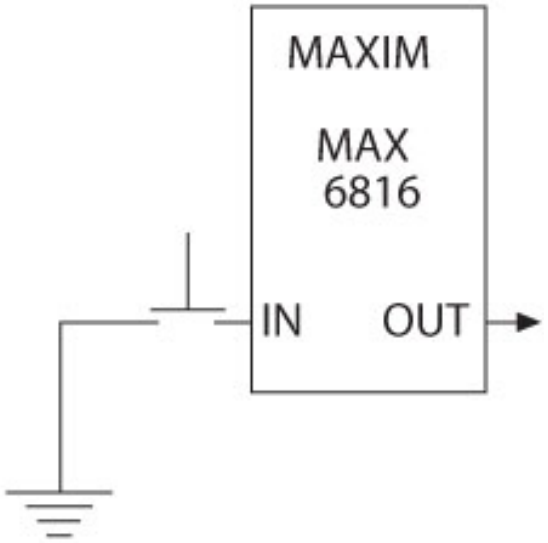


Fig. 5.13 *Interfacing Switches and Displays for Problem 5.7*

64K I/P & 64K O/P

Key Debouncing Circuits





Thank You