

# I/O port structure & I/O Transactions

principle function of an I/O port is

- ① serve as a way station for data in transit between the computer & external world
- ② provide the control logic and signals between the computer & outside world that is necessary for data transfer.

structure

- data Reg → for data in transit
- contr Reg → hold commands from uprocessor to port
- status Reg → accessible by uproc to tell it what is happening (i.e monitor I/O activity)

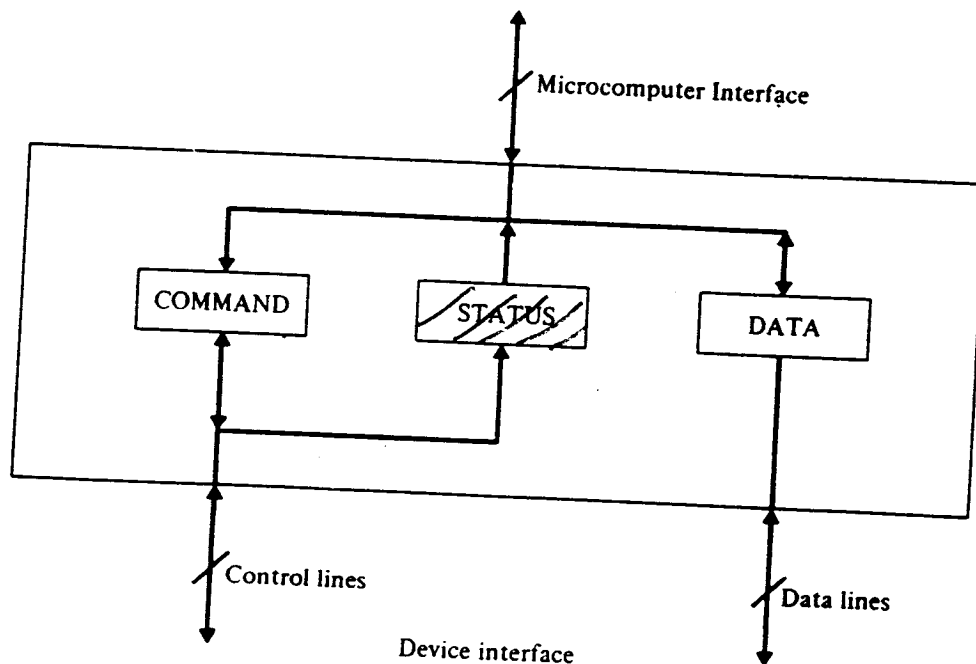


FIGURE 1.8 The structure of a typical I/O port.

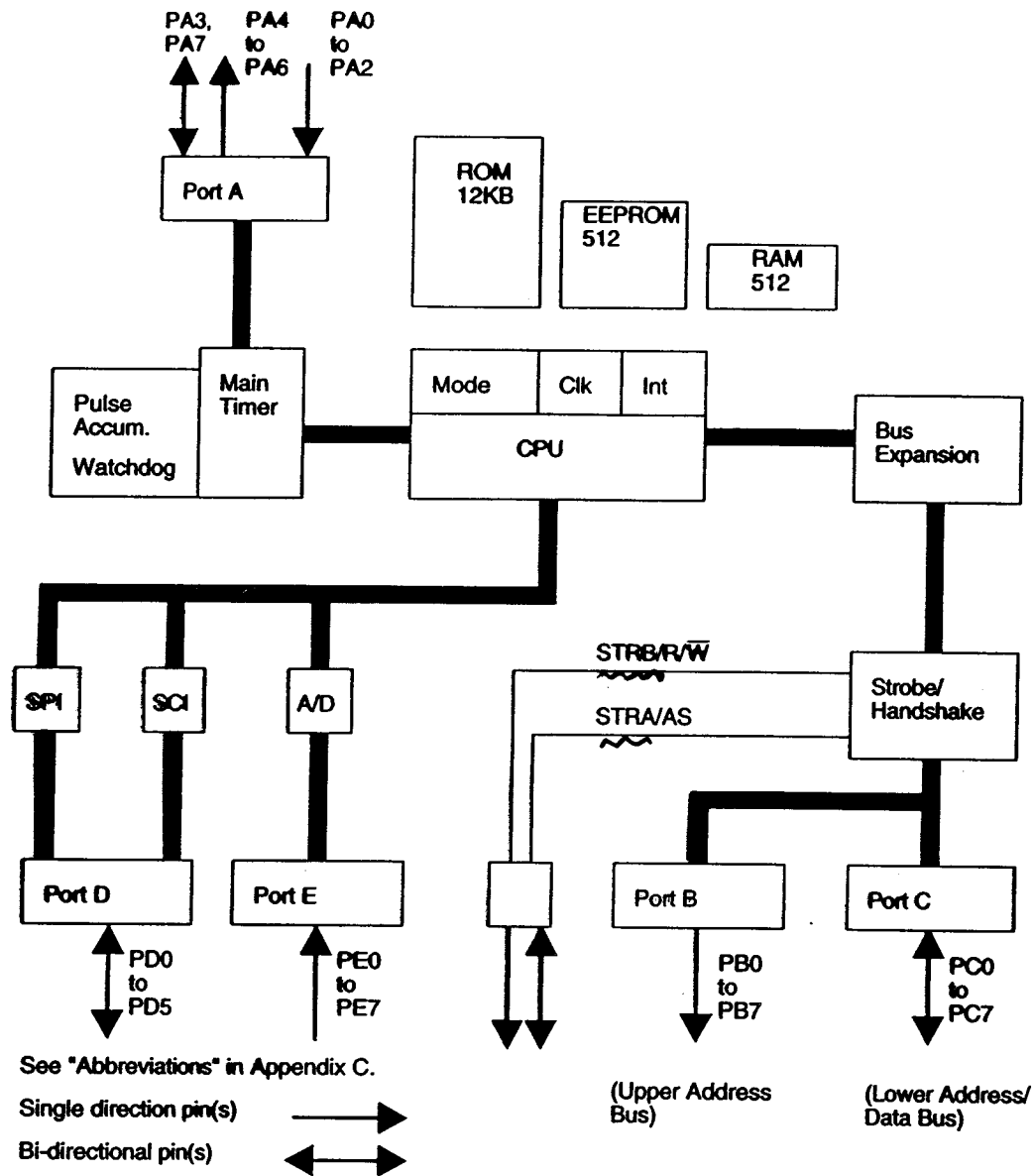


Figure B.1 MC68HC11E9 Block Diagram

STRA  $\rightarrow$  Read peripheral uses STRA to inform 6811 that it has driven parallel comm lines with read data ("I sent data")

STRB  $\rightarrow$  6811 uses STRB to tell a write peripheral that it has driven comm lines with write data ("I am writing data")

## Using parallel ports

parallel ports are often used for simple I/O such as turning ON LED's, seven segment displays or reading switch inputs

## steps for using ports

- ① identify the address of an I/O port & its Data Direction Register DDR
- ② program → "write a value to DDR x of port x" so that the value written reflects the appropriate setting for the port
- ③ store a value to the address of I/O port

TABLE B.1A

M68HC11 E Series Registers (1 of 2)

The 128-byte register block can be remapped to any 4K boundary.

	Bit 7	6	5	4	3	2	1	Bit 0	
\$1000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA
\$1001									Reserved
\$1002	STAF	STAI	CWOM	HNDS	OMN	PLS	EGA	INVB	PIOC
\$1003	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORTC
\$1004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB
\$1005	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0	PORTCL
\$1006									Reserved
\$1007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
\$1008	—	—	PD5	PD4	PD3	PD2	PD1	PD0	PORTD
\$1009	—	—	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
\$100A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	PORTE
\$100B	FOC1	FOC2	FOC3	FOC4	FOC5	—	—	—	CFORC
\$100C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	—	—	—	OC1M
\$100D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	—	—	—	OC1D
\$100E	Bit 15	14	13	12	11	10	9	Bit 8	TCNT (Hi)
\$100F	Bit 7	6	5	4	3	2	1	Bit 0	TCNT (Lo)
\$1010	Bit 15	14	13	12	11	10	9	Bit 8	TIC1 (Hi)
\$1011	Bit 7	6	5	4	3	2	1	Bit 0	TIC1 (Lo)
\$1012	Bit 15	14	13	12	11	10	9	Bit 8	TIC2 (Hi)
\$1013	Bit 7	6	5	4	3	2	1	Bit 0	TIC2 (Lo)
\$1014	Bit 15	14	13	12	11	10	9	Bit 8	TIC3 (Hi)
\$1015	Bit 7	6	5	4	3	2	1	Bit 0	TIC3 (Lo)
\$1016	Bit 15	14	13	12	11	10	9	Bit 8	TOC1 (Hi)
\$1017	Bit 7	6	5	4	3	2	1	Bit 0	TOC1 (Lo)
\$1018	Bit 15	14	13	12	11	10	9	Bit 8	TOC2 (Hi)
\$1019	Bit 7	6	5	4	3	2	1	Bit 0	TOC2 (Lo)
\$101A	Bit 15	14	13	12	11	10	9	Bit 8	TOC3 (Hi)
\$101B	Bit 7	6	5	4	3	2	1	Bit 0	TOC3 (Lo)
\$101C	Bit 15	14	13	12	11	10	9	Bit 8	TOC4 (Hi)
\$101D	Bit 7	6	5	4	3	2	1	Bit 0	TOC4 (Lo)
\$101E	Bit 15	14	13	12	11	10	9	Bit 8	TI4/O5 (Hi)
\$101F	Bit 7	6	5	4	3	2	1	Bit 0	TI4/O5 (Lo)
\$1020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1
\$1021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2

Timer ← port

status polling ←

portc ↑↓

port B output

latched port C

portc control

PIOC

① configure 6811 for parallel I/O operations

② STAF bit serves as a status flag during simple strobe operation

Copyright of Motorola, Used by Permission

TABLE B.1B

	B7	6	5	4	3	2	1	B0	
\$1022	OCN	OCN	OCN	OCN	MOSI	ICN	ICN	ICN	TMSK1
\$1023	OCF	OCF	OCF	OCF	MOSF	ICF	ICF	ICF	TFLG1
\$1024	TOI	RTI	PAOM	PAI	—	—	PR1	PR0	TMSK2
\$1025	TOF	RTF	PAOVF	PAF	—	—	—	—	TFLG2
\$1026	DDRA7	PAEN	PAMOD	PEDE	DDRA3	MOS	RTR1	RTR0	PACTL
\$1027	B7	6	5	4	3	2	1	B0	PACNT
\$1028	SPIE	SPE	ONOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
\$1029	SPIF	WCOL	—	MOOF	—	—	—	—	SPSR
\$102A	B7	6	5	4	3	2	1	B0	SPDR
\$102B	TCLR	SCP2 <sup>1</sup>	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD
\$102C	F8	T8	—	M	WAKE	—	—	—	SCCR1
\$102D	TIE	TCIE	ME	IE	TE	ME	RWU	SBK	SCCR2
\$102E	TDRE	TC	RDRF	IDLE	OR	MF	FE	—	SCSR
\$102F	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0	SCDR
\$1030	COF	—	SCAN	MULT	OD	CC	CB	CA	ADCTL
\$1031	B7	6	5	4	3	2	1	B0	ADR1
\$1032	B7	6	6	4	3	2	1	B0	ADR2
\$1033	B7	6	5	4	3	2	1	B0	ADR3
\$1034	B7	6	5	4	3	2	1	B0	ADR4
\$1035	—	—	—	PTCON	BPRT3	BPRT2	BPRT1	BPRT0	BPROT
\$1036	MBE	—	ELAT	ENDL	ENROW	TI	TO	PGM	EPROG <sup>2</sup>
\$1037									Reserved
\$1038									Reserved
\$1039	ADPU	CSEL	IRGE	OLY	OME	—	OR1	CRO	OPTION
\$103A	B7	6	5	4	3	2	1	B0	COPRST
\$103B	ODD	EVEN	ELAT <sup>3</sup>	BYTE	ROW	ERASE	EELAT	EPGM	PPROG
\$103C	RBOOT	SMOD	MDA	IRVNE	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO
\$103D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT
\$103E	TLOP	—	OOCR	OBYP	DISR	FCM	FOOP	TOON	TEST1
\$103F	EE3 <sup>4</sup>	EE2 <sup>4</sup>	EE1 <sup>4</sup>	EE0 <sup>4</sup>	NOSEC	NOCOP	ROMON <sup>5</sup>	EEON	CONFIG

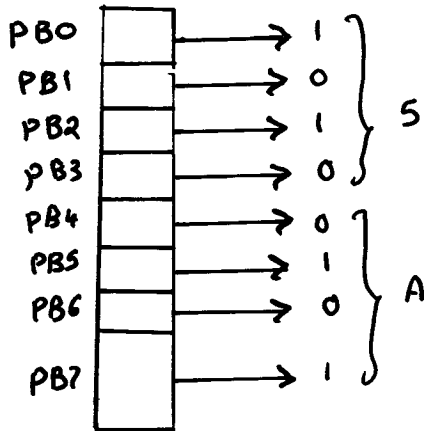
1. MC68HC(7)11E20 only
2. MC68HC711E20 only
3. MC68HC711E9 and MC68S711E9 only
4. MC68HC811E2 only
5. Not applicable to MC68HC811E2. For devices with disabled ROM array (MC68HC11E0, MC68HC11E1, MC68L11E0, or MC68L11E1) this bit must never be set to one.

Copyright of Motorola, Used by Permission

# The parallel I/O subsystem

port B ; port C

\$1004 ≈ port B

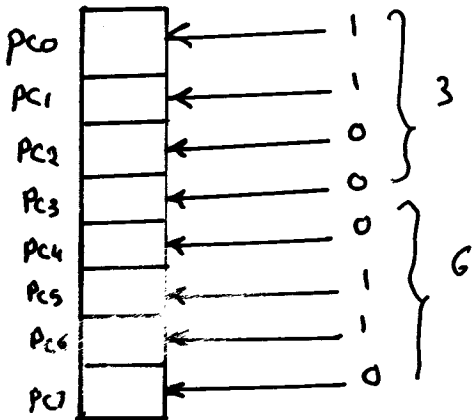


Example LDAA # \$A5

STAA \$1004

output port only  
unidirectional

\$1003 ≈ port C

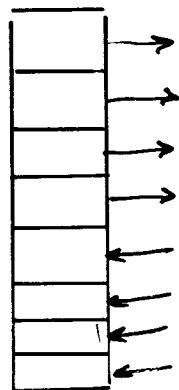
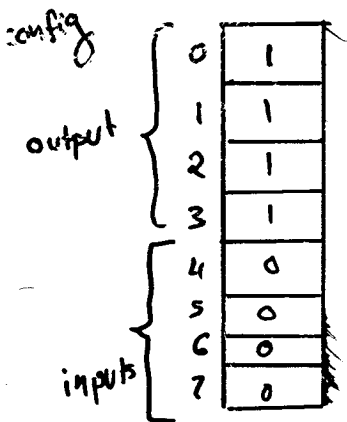


Example LDAA \$1003

Then ACCA has value \$63

1007\$  
DDRC

1003\$  
portC



\* configure

LDAA # \$0F

STAA \$1007

\* Drive PC1 High & PC3 low

LDAA # \$02

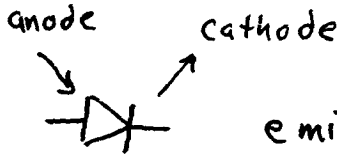
STAA \$1003

\* Read inputs

LDAA \$1003

# Driving LED'S

① AN LED



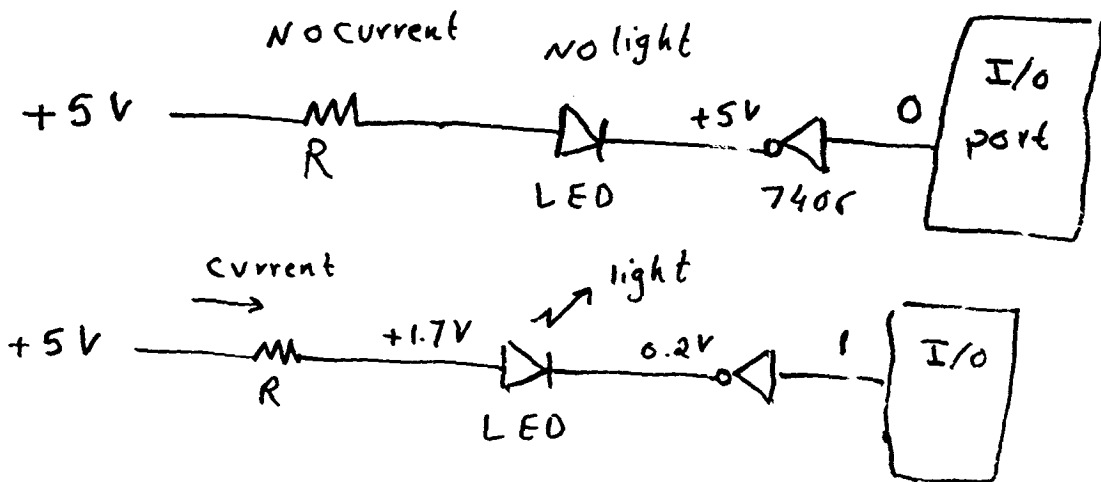
emits light when current

flows through it in the positive direction

" i.e when the voltage on the anode side is made higher than the voltage on the cathode side "

- The forward voltage across the LED is typically about  $\approx 1.5$  volts

circuits for interfacing LED with HC11/12

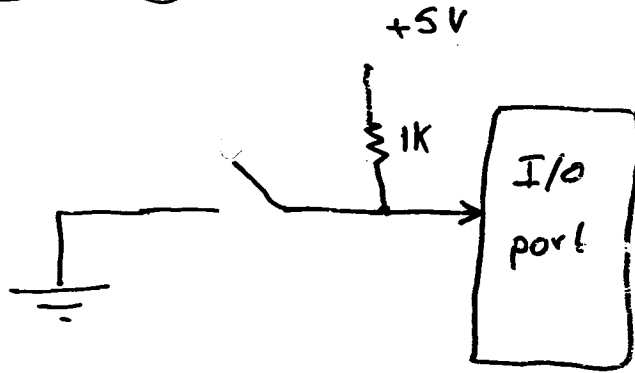


R ??

$$R = \frac{V_{CC} - V_{OL} - V_{LED}}{I_{LED}} = \frac{5 - 0.3 - 1.5}{10mA} = 320\Omega$$

valves 200-400  $\Omega$

# Interfacing a switch



- When switch is **open** the value at the I/O port will be pulled up to 5 volts by the 1k pull up resistor
- when switch is **closed**, the value at I/O port will be zero.

problems?

# Switch debouncing

When mechanical switches are opened or closed there are brief oscillation due to mechanical bouncing (switch bounce)

\* sequential logic circuit can provide an output that is free from multiple transitions

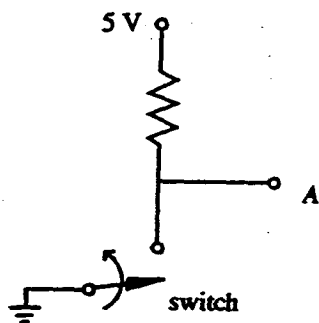


FIGURE 6.21  
Switch bounce.

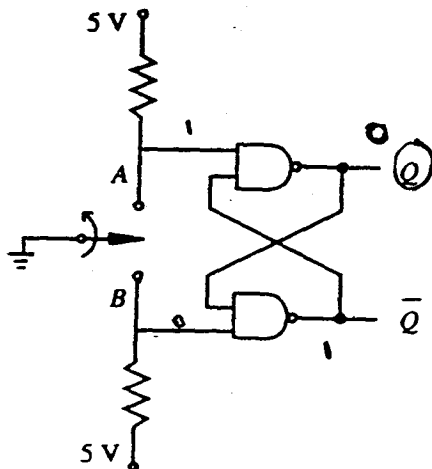
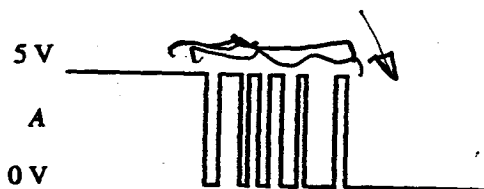
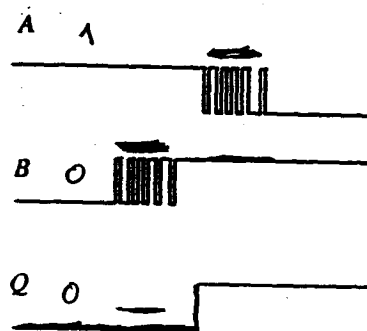
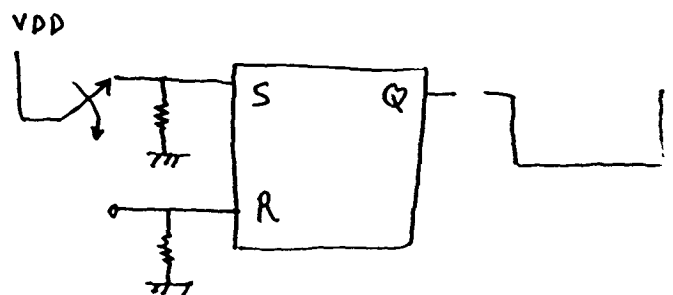


FIGURE 6.22  
Switch debouncer circuit.

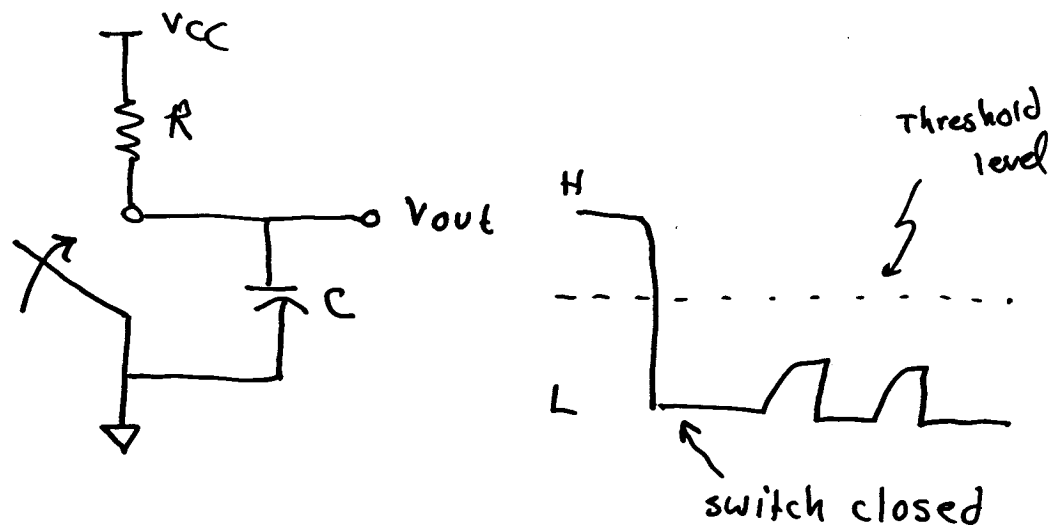


NAND

i1	i2	out
0	0	1
0	1	1
1	0	1
1	1	0



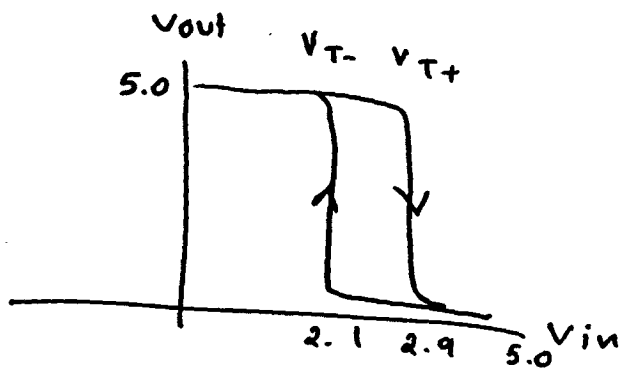
\* Integrating debouncer: (smoothing filter)



The  $RC$  constant of the integrator determines the rate at which the capacitor charges up towards the supply voltage once the Ground connection via the switch has been removed.

→ "As long as the capacitor voltage does not exceed the logic zero threshold value, the  $V_{out}$  signal will continue to be recognized as a logic zero"

# Schmitt trigger



switching threshold is different

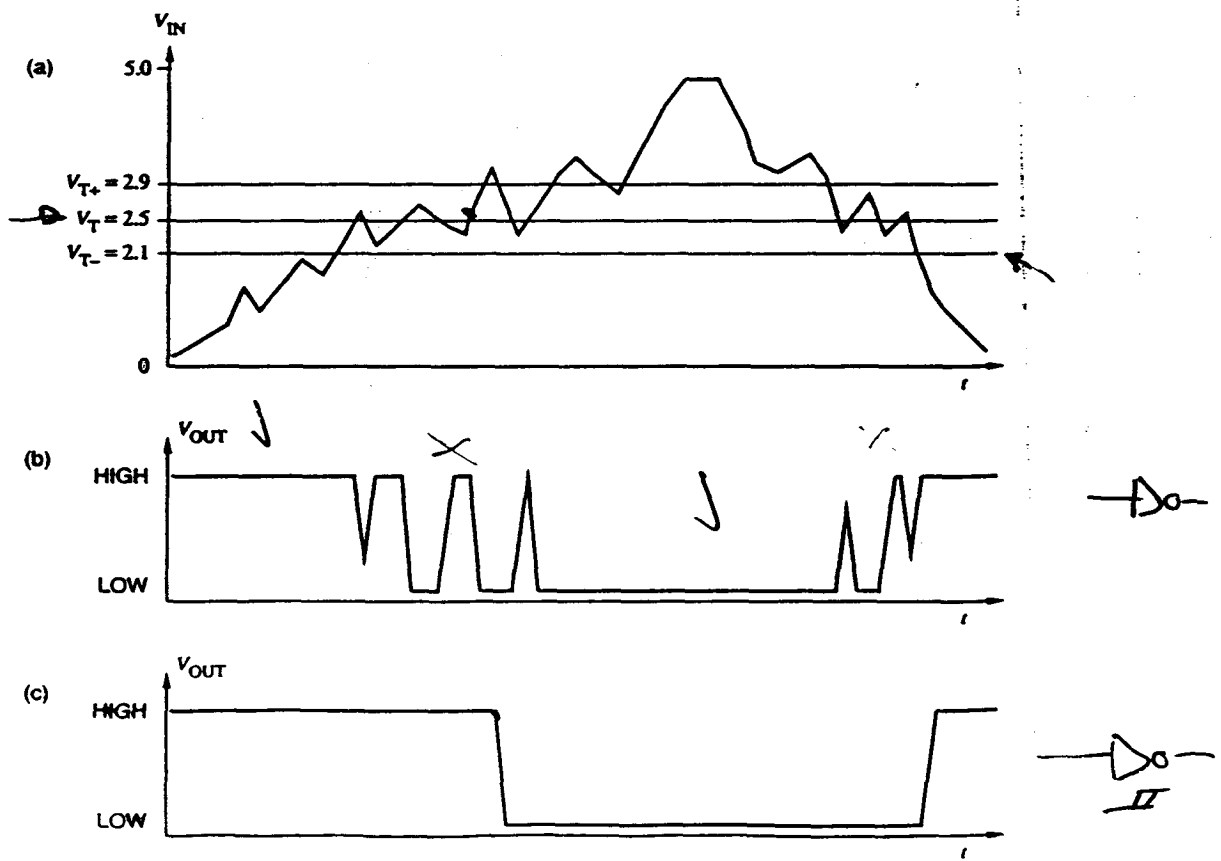


A Schmitt trigger is a special circuit that uses feedback internally to shift the switching threshold depending on

whether the input is changing from Low to High or High to Low

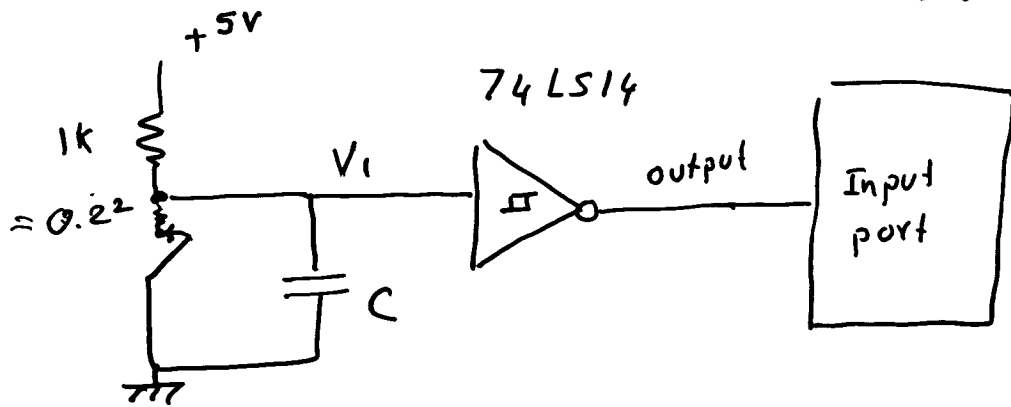
The difference between  $V_{T+}$ ,  $V_{T-}$  is called hysteresis  
→ useful for noise

\* Schmitt-trigger inputs have better noise immunity than ordinary gate inputs for signals that contain transmission line reflections



**Figure 3-45** Device operation with slowly changing inputs: (a) a noisy, slowly changing input; (b) output produced by an ordinary inverter; (c) output produced by an inverter with 0.8 V of hysteresis.

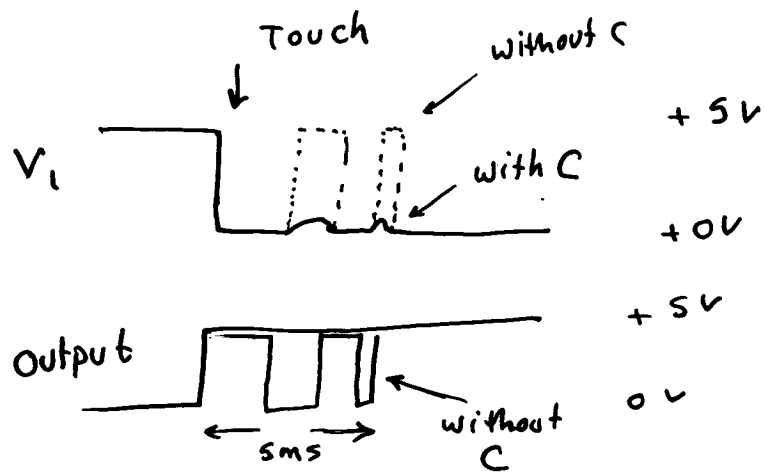
\* Switch bounce removed with a capacitor



- Capacitor value is chosen large enough so that the input voltage does not exceed 0.7V threshold of NOT gate while it is bouncing

\* Switch touch  
Bounce removed  
by cap

No delay



\* switch release  
Bounce is removed  
by cap

significant delay from release  
until fall of switch



\* Timing used to calculate capacitor value

The voltage rise during a bounce interval when switch is open is given by

$$V \geq 5 - 5e^{-t/RC}$$

assume  $R = 1k$  bounce time = 5ms

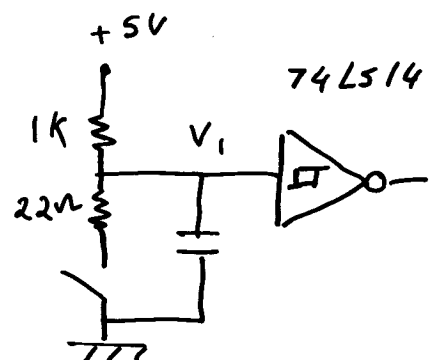
$$0.7 \geq 5 - 5e^{-5ms/(1k \cdot C)}$$

$$0.86 \leq e^{-5ms/(1k \cdot C)}$$

$$\ln(1.16) \geq \frac{5ms}{(1k \cdot C)}$$

$$C \geq \frac{5ms}{1k \cdot \ln(1.16)} = 33\mu F$$

(imp) one problem with above interface is the instantaneous current that occurs when switch bounces closed. (spark)  
(0.1  $\Omega$  contact Resistance of switch)

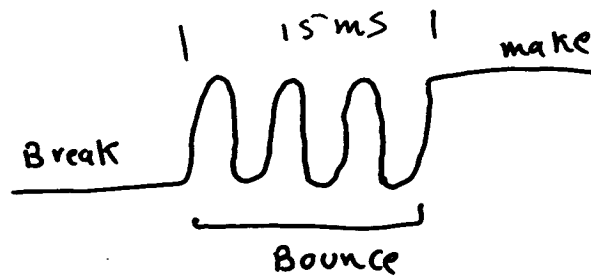


# Software Debouncing

approach #1

To debounce a switch, a software time delay is used that provides a time delay (usually 20-30ms) longer than the duration of the switch bouncing action.

wait & see



so if switch goes low wait for longer than 10ms  $\approx 20\mu$  and then test for the switch still being low.

approach #2

integrating debouncer

idea

initialize a counter with a value of 10 and after the first logic low level is detected to poll the switch every millisecond if the switch output is low, decrement the counter. If the switch output is high increment the counter.

\* when the counter reaches zero, we know the switch output has been low for at least 10ms

\* If on the other hand, the counter reaches 20, we know that the switch has been open for at least 10ms.

# \* pseudocode design for integrating switch debouncer

Initialize count = 0

while count > 0 and < 20

```
Do
    Delay 1 millisecond
    Get switch input
    If switch is closed
        Then Decrement count
        Else increment count
    Endif switch closed
END-DO
```

IF count = 0

then switch is closed

else switch is open

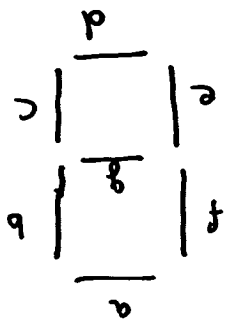
ENDIF count = 0.

\* Seven segment display :

(a, b, c, d, e, f, g)

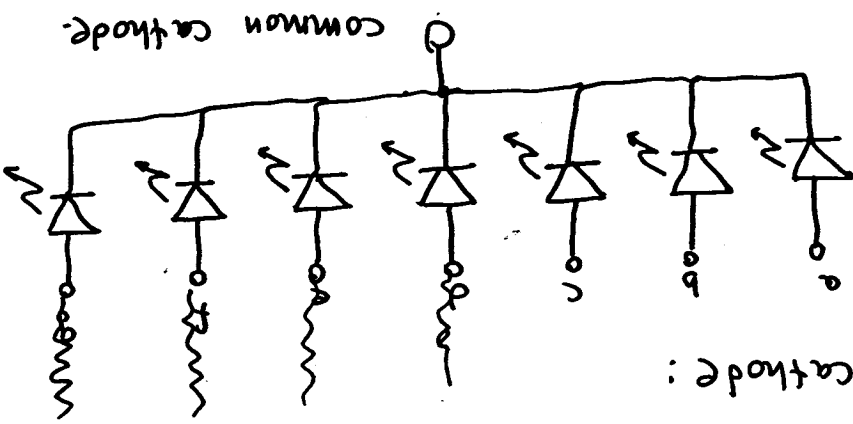
LED segments

consists of seven



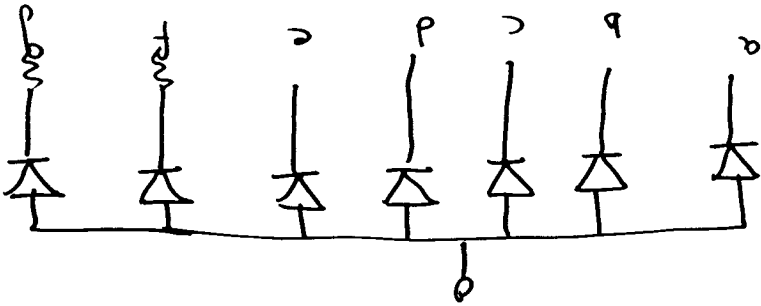
There are two types of seven segment displays

① common cathode :



segment will be lit whenever a high voltage is applied

② common anode : all anodes are tied in common



segment will be lit whenever a low voltage is applied.

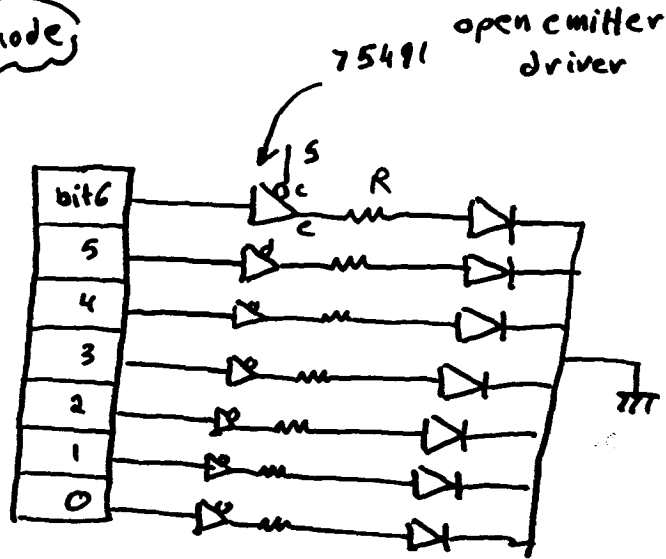
# Seven segment LED Interfaces

→ Common cathode  
→ common Anode

Common cathode

Seven cathodes are connected

75491 "source current from supply"



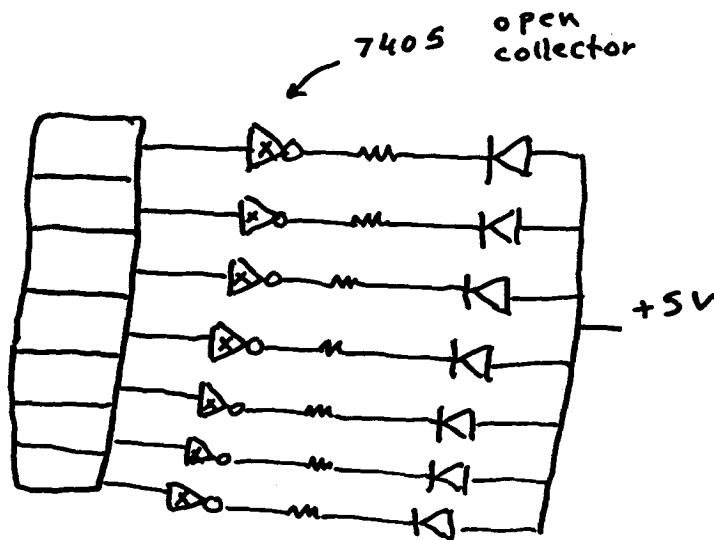
$$R = \frac{5 - V_{ce} - V_d}{I_d}$$

$$= \frac{5 - 0.79 - 2}{10\text{mA}} = 221\Omega$$

Common anode

Seven anodes are connected

"A current sink is required to interface a common anode display"



$$R = \frac{5 - V_{OL} - V}{I_d}$$

$$= \frac{5 - 0.4 - 2}{10\text{mA}} = 260\Omega$$

Common error

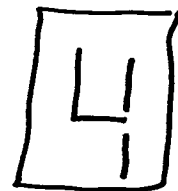
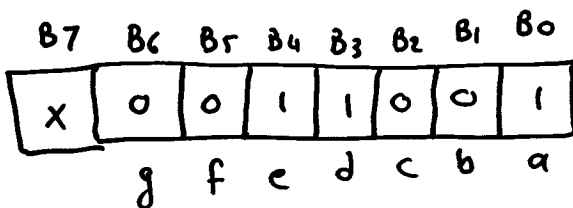
IF you try to replace the seven individual resistors in either of two circuits above with a single resistor on common side, then brightness of each LED will be a function of how many LEDs are on!

# Driving the 7-seg Display

**imp** current limiting resistors must be included  
if they are omitted, a forward biased condition  
will short-circuit the LED and cause it to  
draw too much current

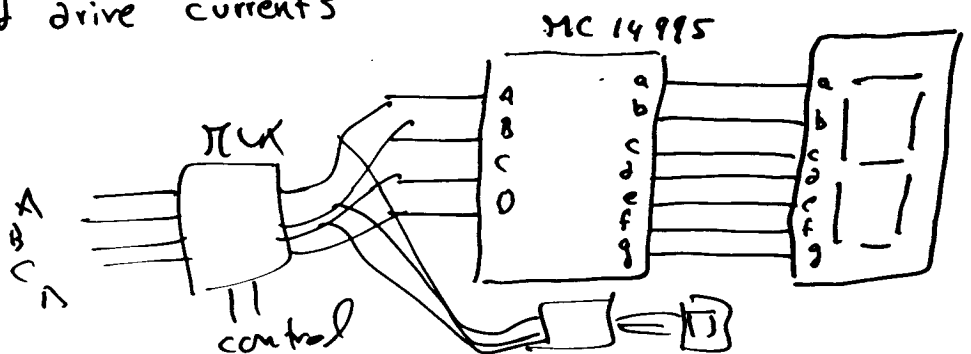


## ① Use port B



Hex code = \$19

② Hardware decoder some ICs are specially designed  
to drive 7-seg display. They contain buffers to  
supply required drive currents



# Keypad interfacing Techniques:

A keypad consists of a number of key switches arranged in a matrix of rows and columns

small matrices → directly connected to MCU  
 large matrices → use decoders

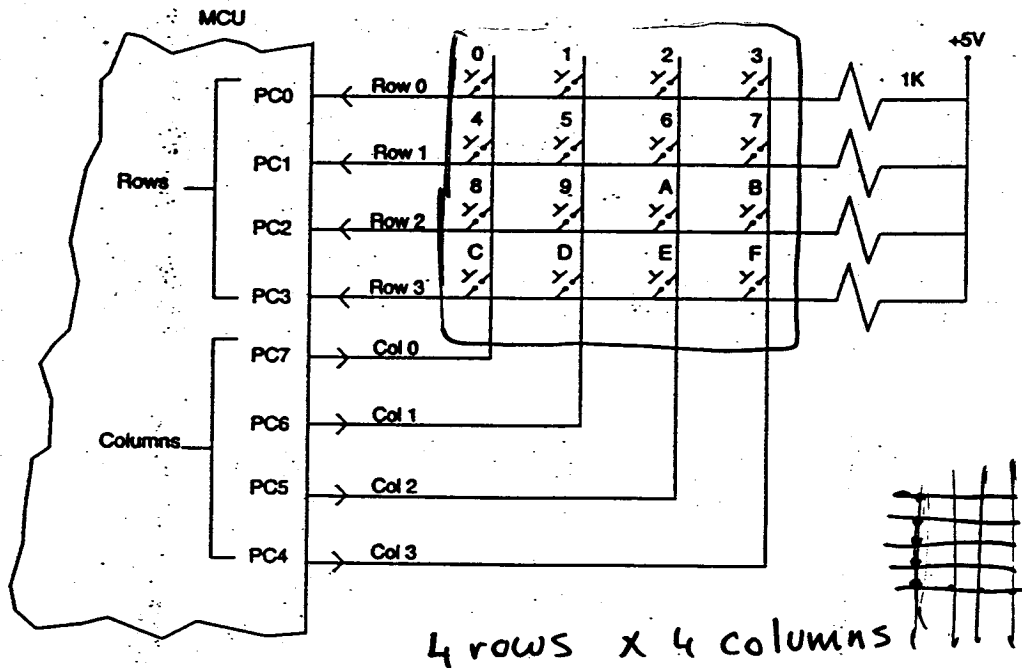


Figure 6.25



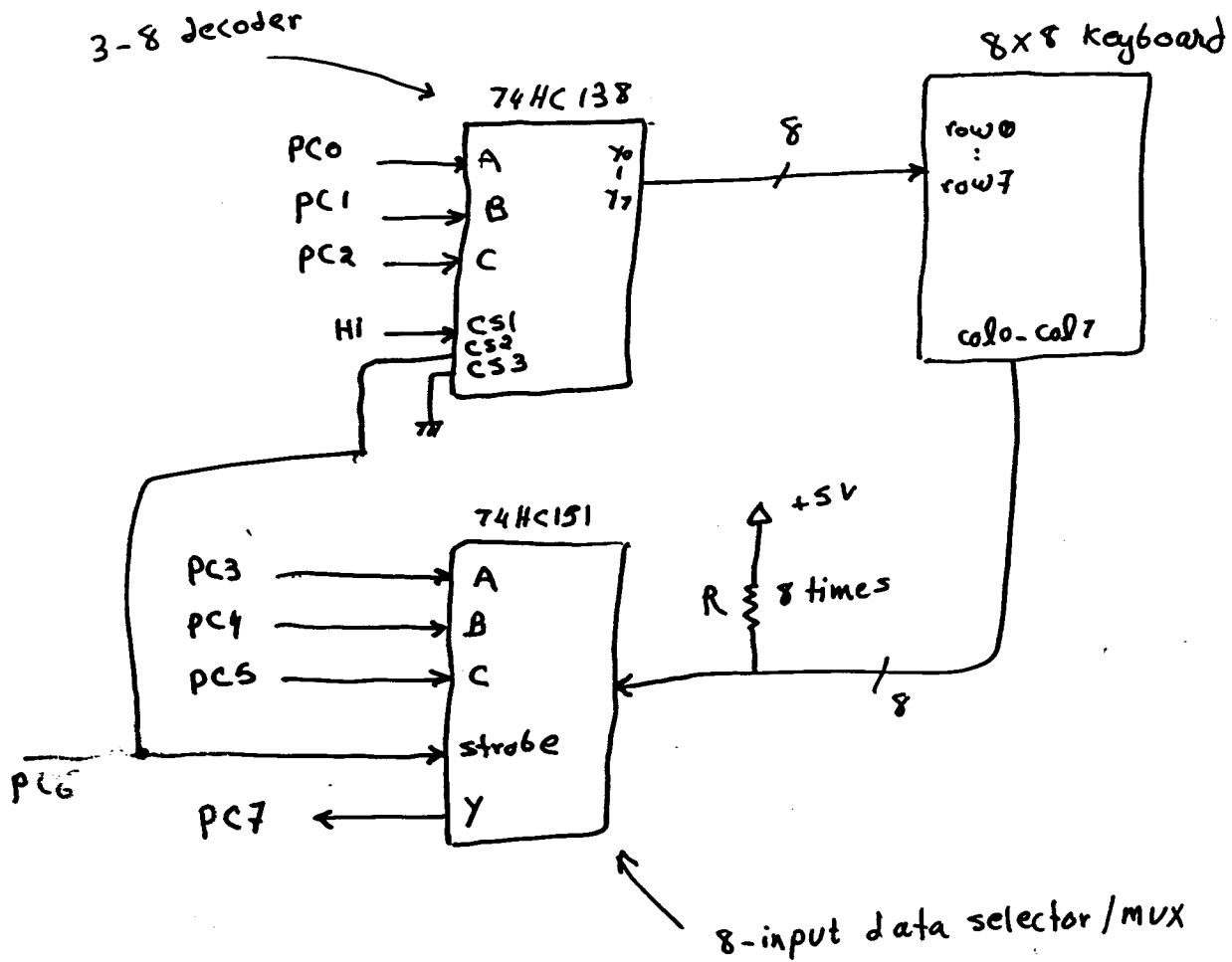
- The rows have high logic since they are connected → +5
- To identify a pressed key, the technique is to send a logic "0" on each column, one at a time, then find out which row has a logic low.

The combination of both low-logic column and low logic row will identify which key is pressed.

How about 20 keys?

# Using Hardware Decoding chips

example 8x8 matrix keyboard with 64 keys?



**Note** Transfer of output Y of mux occurs only when strobe is low  
 same applies for decoder signal CS2

\* Sending an output % 00010011 will drive row 3 low and select input from column 2.

# Liquid crystal Displays

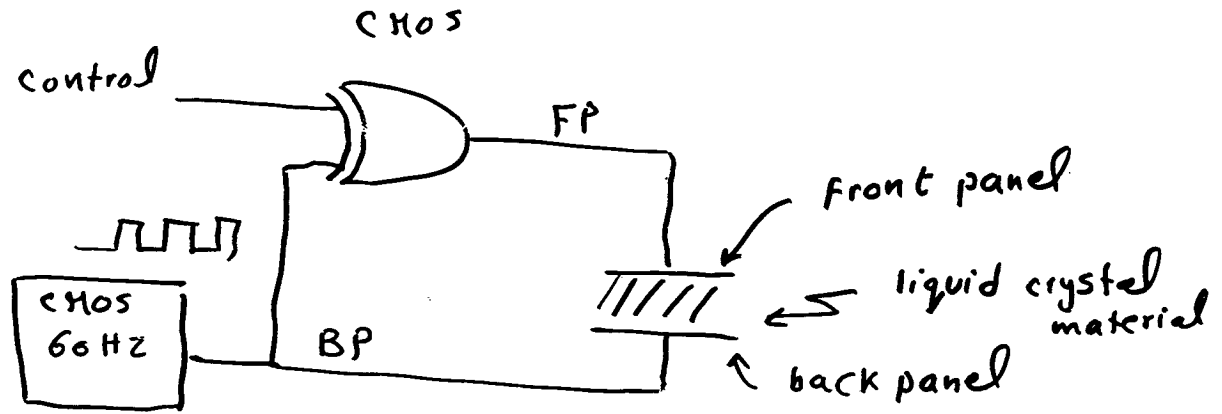
LCD's are widely used in microcomputer systems

- ① Low power consumption w.r.t LEDs  
this allows the display and even the entire computer system to be battery operated
- ② LCDs are more flexible in their sizes & shapes  
so this permits the combination of numbers, letters, words and graphics to be driven with relatively simple interface

Disadvantage Low response

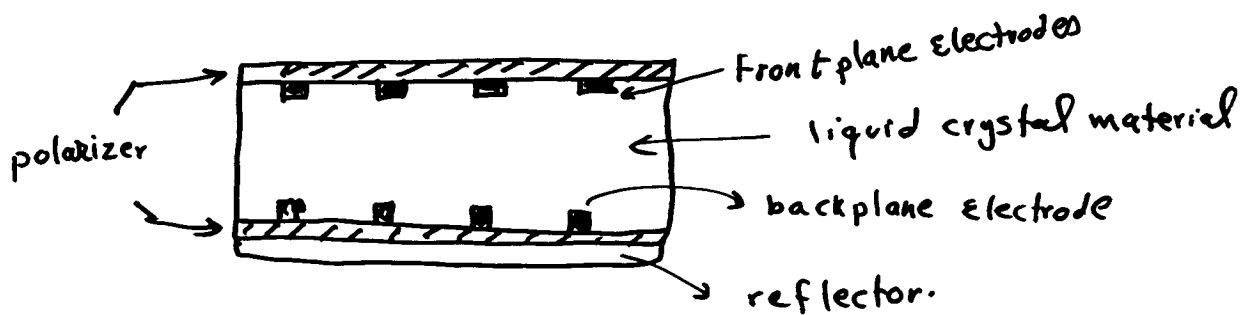
imp unlike LEDs that convert electric power to optical power, an LCD uses AC voltage to change the light { reflectivity, absorption }  
i.e light energy is supplied by the room or separate backlight

Basic idea



(imp) A constant (DC) excitation signal will polarize and destroy the crystal

(So) An LCD display requires an alternating excitation wave applied to selected electrodes to change selected areas



- reflective type
- The excitation wave develops an electrostatic field to align the liquid crystal molecules in these selected areas.
  - when crystals are aligned, they allow light to pass through the mirror
  - In the charged area the mirror reflects more light than the uncharged area.

# Basic timing of a liquid crystal

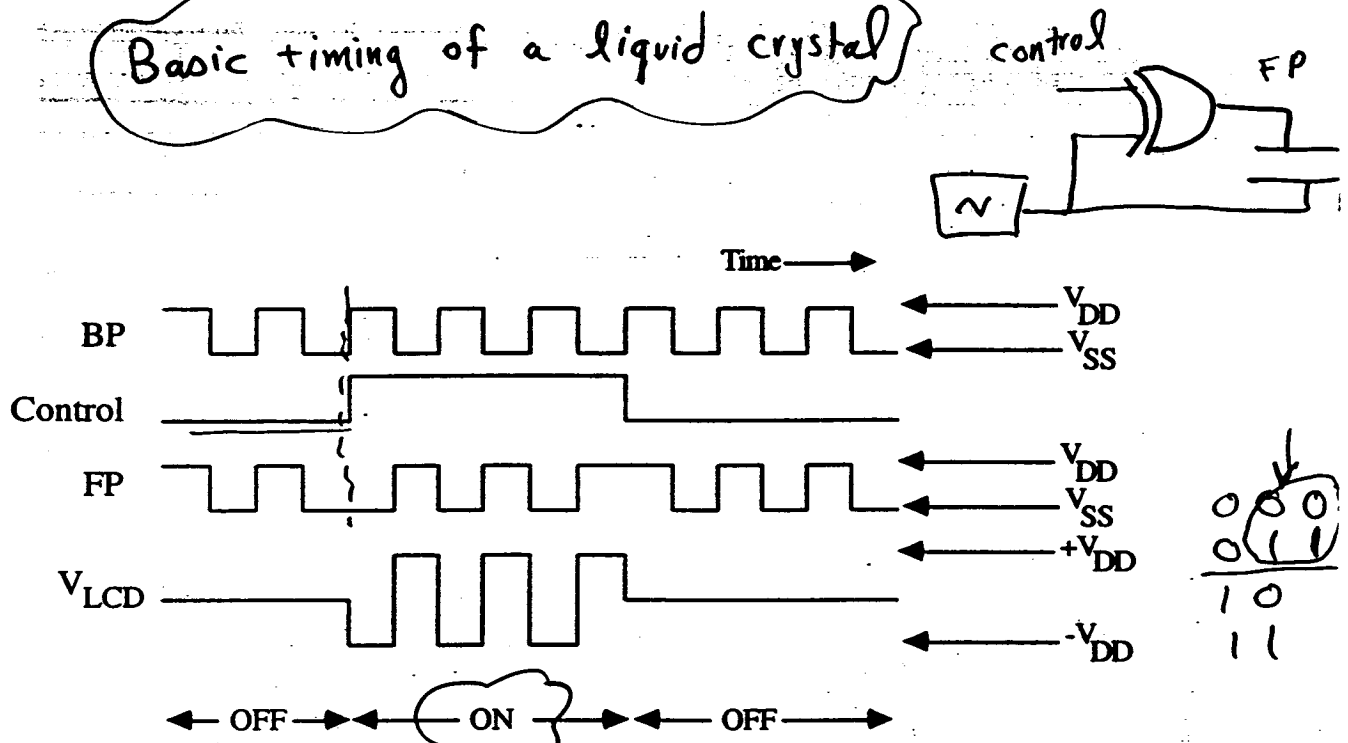
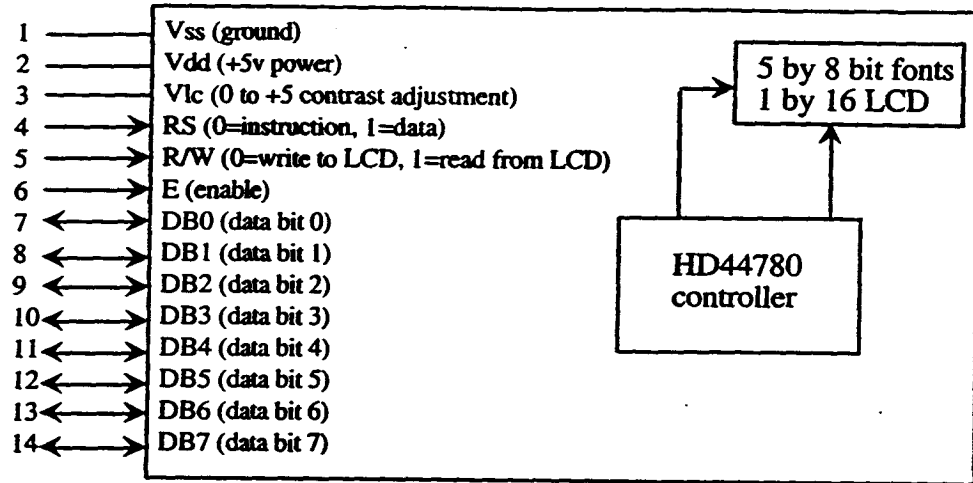


Figure 8.44  
The basic timing of a liquid crystal interface.

- let  $v_{DD}$  be supply voltage of CMOS  
 $v_{SS}$  be ground
- why CMOS  $\rightarrow$  fully restored logic
- for both control = high }  $V_{LCD} = FP - BP$  (i.e. no d.c component)  
                                  low }
- when control signal = low ,  $V_{LCD} = 0$  (no reflection)
- when " " = high  $V_{LCD} = \pm V_{DD} \rightarrow$  reflects light

**Figure 8.55**  
Interface of a HD44780  
LCD controller.



**Table 8.8**  
Two control signals  
specify the type of  
access to the HD44780.

RS	R/W	Cycle
0	0	Write to Instruction Register
0	1	Read Busy Flag (bit 7)
1	0	Write data from microprocessor to the HD44780
1	1	Read data from HD44780 to the microprocessor

\* parallel port LCD interface with HD44780 controller

- There are 4 types of access cycles to HD44780 depending on RS and R/W
- Two types of synchronization can be used blind cycle (unconditional) and gadfly (polling)
- Most operations require 40  $\mu$ s to complete while some require 1.64 ms

\* seven segment LCD

Driver IC is MC 14543B

(BCD-to-seven seg Latch/Decoder/Driver for LC)

It codes only 4-bit nibbles % 0000 - % 1001

\* Dot matrix LCD

MC 14500c

(Serial input Multiplexed LCD driver)

- seg on
- seg off

\* displays graphical & alphanumeric info

Display element is turned on or off by controlling corresponding row (BP) & column (FP) lines.

5x8 Dot matrix

