

8254: (PIT)

Programmable Interval timer

- Times are used to produce delays.
 - Times produce delay by counting.
 - 8254 has 3 16 bit down counters.
(counter 0, counter 1, counter 2)
 - delay produced by a program is called software delay whereas delay produced through 8254 is called hardware delay because it is produced by a chip.
- cheap,
 more flexible
 gives simple wait.
- Then why
 do we
 use hardware
 delay?

One massive advantage

- during a delay, CPU can do other operations since 8254 will be busy in giving the delay. so, this saves CPU's time.

→ 8254 has 3 - 16 bit down counts

↳ counts inversely
 $(0, 1, 2, 3, \dots)$

Can produce 3 delays simultaneously

→ 16 bit count (can count upto FFFF)

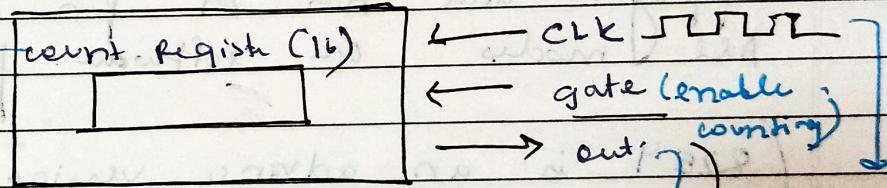
(Given by 8-bit data bus)

(Requires 2 cycles to give the count)

$\frac{1}{(65,000)}$

up gives
count to
8254.

To hold the
count, we need count Register (16 bits).
Many count \rightarrow FFFF H.



If gate = 1,

counting starts

If gate = 0, counting stops.

Many count \rightarrow FFFF H.

when counting
is over, 8254 sends
an interrupt 'OUT'.

* 8254 can also be used as a counter.

(Theory)

Introduction to 8254

- (1) 8254 is a device designed to solve the timing control problems in a microprocessor.
- (2) It has 3 independent counters, each capable of handling clock inputs upto 10 MHz, and size of each counter is 16 bits.
- (3) It operates in +5 regulated power supply and has 24 pin signals. All modes are software programmable.

(8254 is an advance version of 8253 which did not offer the feature of read back command.)

Why to use 8254?

In a microprocessor based system, we come across two important modes -

- ↳ Timer → to provide delay
- ↳ Counter → to count incoming pulses

(1) delay:

- 8085 can provide delays of any value but it uses software to implement the delay.
- The instructions are arranged to waste time.
- The main disadvantage of this scheme is that 8085 is executing some instructions, it means that is it is busy & is not doing work.

(2) Counter:

- The 8085 can count no. of pulses arriving at a port.
- To implement this, 8085 goes on checking port, if it is active it increases counter by 1 and again goes on checking port.
- The same disadvantage, 8085 will have to execute instructions.

Thus,

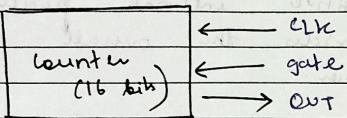
- In a large system when wastage time is critical, a separate timer 8254 can be used.
- 8254 programmable interval timer consists of 3 identical 16 bit counters.
- These counters can work as counters or can provide accurate time delays.
- To operate as a counter, a 16 bit count is loaded in the desired mode of operation is selected.
- The counters will work independently and generate the desired output.
- Now, the 8085 microprocessor job is to initialize and load counters.

Features offered by 8254 PIT

- 1) When an operating system is working on timing sharing basis, then the timer enables smooth operation while switching programs using an interrupt.

- 2) The timer behaves as a counter that serves the purpose of an external event \Rightarrow counter that counts the repetitive internal operations and provides the overall count to the processor.
- 3) It is used to measure the time difference between two external events.
- 4) It acts as baud rate generator (free running counters that generates an internal clock)
- 5) The timer enables the processor to initiate a new operation after a sequence of external events have occurred.
- 6) Through timer, timing signals can be sent to I/O devices at periodic intervals.
- 7) It has a powerful command called READ BACK COMMAND which allows the user to check the count values, programmed mode and current mode and the status of the count.

Working of counters:



7) The OUT pin acknowledges that the counting has been completed. Output is obtained as according to the mode of operation.

Architecture of 8254.

1) There are 3 independent, 16-bit down counters.

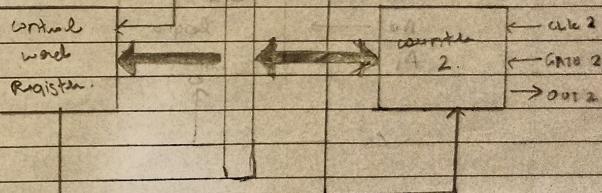
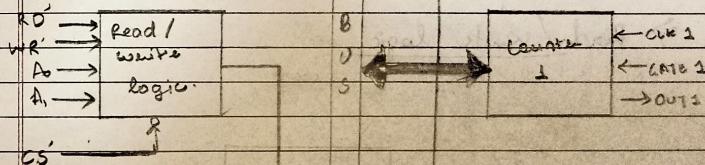
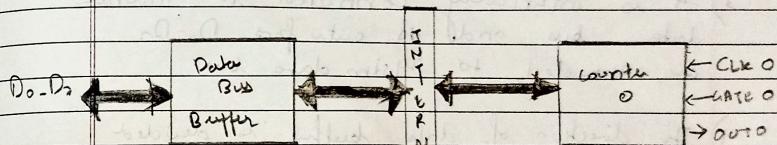
2) They can be programmed separately through control word register to decide mode of counters.

3) It has 2 inputs: CLK (clock) and gate while 1 output: counter OUT.

4) Clock is used to input to counter. Since, it is 16 bits, it can input upto FFFFH.

5) Gate is used to control the counter. If gate = 1, counting starts. If gate = 0, counting stops.

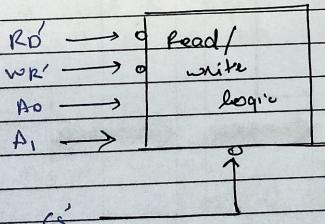
6) And since, 8254 is a down counter, thus after each cycle of the clock input, the count decrements the value by 1.



2) Data Bus Buffer

- 1) It is tri-state, bidirectional, 8 bit data bus buffer.
- 2) It is used to interface 8254 data bus with system data bus.
- 3) It is internally connected to internal data bus and its outputs D₀-D₇ are connected to system data bus.
- 4) The direction of data buffer is decided by read and write control signals.

3) Read / Write logic



- 4) This block accepts data inputs from system control bus and address bus.
- 2) In I/O mapped I/O, the signals RD' and

(4)

WR are connected to ZO and ZW

- b) In memory mapped I/O, RD and WR are connected to MEMR and MEMW.

A₀ and A₁ are directly connected to address lines A₀ and A₁.

CS is connected to address decoder. When this chip is selected, only is active, only then this chip is selected and then the data bus will transfer data.

A₀, A₁ selects a specific part. WR, RD decides writing data to 8254 or reading data from 8254.

The control word register and the counts are selected according to the signals on the line A₀ and A₁.

A ₁	A ₀	Selection
0	0	Count 0
0	1	Count 1
1	1	Count 2
1	1	Control word register

Control word Register :

- 1) The register of 8254 programmable interval timer gets selected when $A_0 = 1$ and $A_1 = 1$.
- 2) It is used to specify the BCD or binary counter to be used, its mode of operation and the data transfer to be used i.e., read or write the data bytes.
- 3) If the CPU performs a write operation, the data is stored in the control word register and is prefixed to an control word. It is used to define counter to operation.
- 4) The data can only be written into control word register, no read operation is allowed. Status information is available with the help of \rightarrow read back command.

PIN DESCRIPTION : 8254

Symbol	Name	function.
D ₇ -D ₀	Data Bus	There are 8 bit, Bidirectional data bus lines connected to the system data bus for data transfer between 8085 and 8254.
CS	chip select	This is an active low input signal, used to select the 8254 IC. If CS = 0, 8254 will be active and take part in the data transfer. If CS = 1, 8254 will be in inactive state.
RD	Read	This is an active low, input signal, used in coordination with A ₀ , A ₁ to send data from appropriate counter lines to data lines D ₇ -D ₀ .
WR	Write	This is an active low input signal used in coordination with A ₀ , A ₁ to load counter or to initialize counters.

A₀-A₁ Address lines There are **input address lines** used to distinguish different parts of 8264 such as counter 0, 1, 2, control word register.

Ck₀-2 clock input There are **clock inputs** to 3 independent counters. The pulse applied at this pin will be counted by the respective counters.

GATE₀₋₂ gate control These are **active high, input signals** used to allow the external hardware to control the respective counter. The function of gate input is dependent on operating mode.

OUT₀₋₂ Output: These lines are **active high, output lines**. The output is dependent on operating modes.

Programming of 8264:

- At the time of programming, it is necessary that each individual counter of 8264 is to be programmed separately using control word and count value.
- The figure shown below shows the format of the control word.

B ₂	B ₁	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀
S _{C1}	S _{C0}	RW ₁	RW ₀	M ₂	M ₁	M ₀	BCD

00	Select counter 0	000 - Mod 0	Ld: BCD count (decimal)
01	Select counter 1	001 - Mod 1	
10	Select counter 2	X10 - Mod 2	Ld: Binary count
11	Read	X11 - Mod 3	(Hexade-
	Back count	100 - Mod 4	cimal)
		101 - Mod 5	

00: Counter load command.

01: Read/write LSB only

10: Read/write MSB only

11: Read/write LSB then MSB

A) To select the BCD or Binary count.

2) B_1, B_2 and B_3 are used to select one of the modes of operation for a specific counter.

3) B_6, B_7 specify the counter.

CL! For the function to take place, the control word is needed to be sent to each separate 16 counter at the same control address register. The identification of the control word of the particular counter is done using bits B_6 and B_7 by the control word format.

GA) The read/write operations are performed using bits B_6 and B_7 .

Through these bits, the 16-bit count value is read and written in orderly sequence.

C) Also, each time the read operation will take place, the count value is to be read by stopping the counter. Here, basically the count value is latched to an internal latch present at the output of each counter before the read operation.

(e) Write a program to produce a square wave of 1KHz from an input frequency of 16.535 MHz using 8254 counter 0.

Input frequency: 16.535 MHz
Output frequency: 1KHz.

$$\text{O/p frequency} = \frac{\text{i/p frequency}}{\text{count}}$$

$$\text{count} = \frac{\text{i/p}}{\text{o/p}} = \frac{16.535 \times 10^6}{1 \times 10^3}$$

$$\text{count} = (16625)_{10}$$

$$\Rightarrow 4080 \text{ H}$$

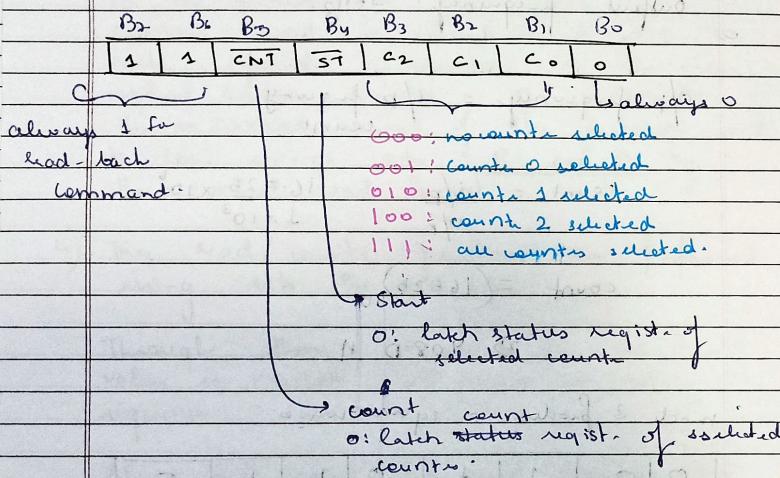
mode 3 produce a square wave.

0	0	1	1	0	1	1	0
SC ₁	SC ₀	RW ₁	RW ₀	M ₂	M ₁	M ₀	BCD

(6)

- ⇒ Unlike 8253, there is a separate read-back control word format in 8254 through which the count value is latched.

The format of the read back control word of 8254:

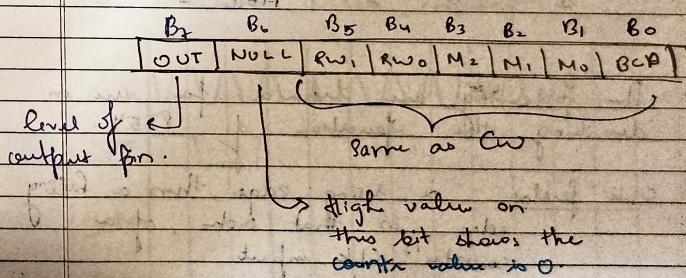


- 1) In order to latch the latched the count value before the read operation takes place, the control word is sent to the same control register address.

- 2) The control word is identified using bit values of B₆ and B₇ by the control register.

- 3) By sending one control word, one or all counters can be latched by the read-back control word.
- (4) The status register is also latched to the output latch of the counters through the control word, this allows reading the status register by the respective addresses of the counters.
- (5) It is to be noted that at any specific instant, either one can't latch the count value by programming the bit B₆ as zero or the latch the status regist. by programming the bit B₇ as zero.

- ⇒ Status register of each counter:



- 2) The programmed status of the counter is judged by the states (end of the counter and through this, one can also check what whether the count value has become zero or not).

8254 PLT Timer Modes

- 2) 3 bits of M_2, M_1 , and M_0 are used to decide programming modes.

M_2	M_1	M_0	Operating Mode
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

- The following are defined for use in describing the operation of 8254.

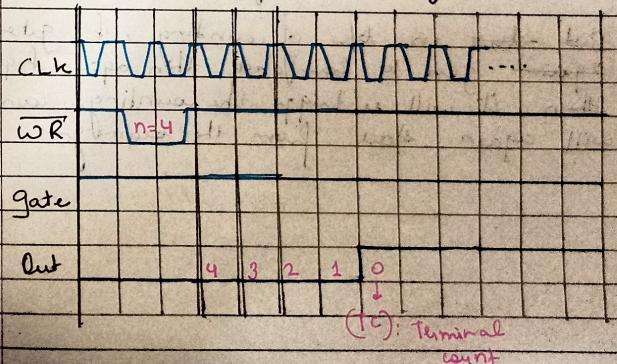
- (a) Clk pulse: a rising edge, then a falling edge, in that order, of a counter's Clk input.

- (b) Trigger: A rising edge of a counter's gate input.

- (c) counter loading: the transfer of a count from the CR to the CE (refer to functional description).

I] MODE 0: INTERRUPT ON TERMINAL COUNT

- This is used for event counting.
- After writing the control word, OUT is low at first. It will remain low until the counter reaches 0, it is decremented by 1 after each clock cycle.
- Then the OUT goes high, and remains high until a new count is there or a new Mode 0 control word is written into the counter.
- The gate = 1 indicates enabling counting, and 0 indicates disable counting.



II) MODE I : MONOSTABLE MULTIVIBRATOR

HARDWARE REGR RETRIGGABLE ONE SHOT

- Out will be high at first, it will go low on the clock pulse following a trigger to begin the one shot pulse.
- It will remain 0 until the counter reaches 0.
- In every other mode, gate is an enable signal - in this mode, gate is a trigger signal.
- The rising edge of gate remaining high won't trigger counting. The falling edge trigger of the gate will enable counting.
- If gate goes low, the counting won't stop - gate only gate going high triggers the counting.
- But there is b/w of counting, if gate again goes to low (and then again high), then it will re-trigger the counting. counting will again start from the beginning.

X

CLK

WR

gate

OUT

4 3 2 1 0
(C)

III) MODE II: RATE GENERATOR

- Initially, out is low.
- When counting is enabled, it goes high.
- This process repeats periodically.
- This mode is used as frequency divider.
- gives rectangular wave.

CLK

WR

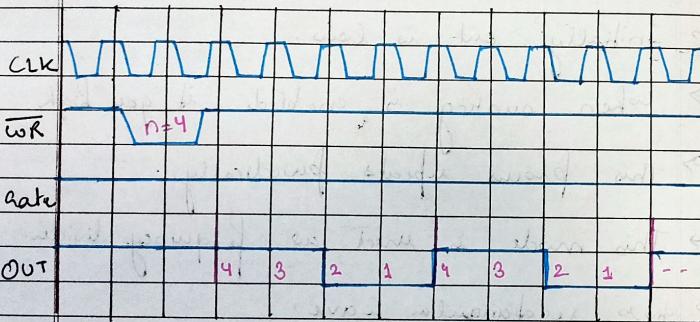
gate

OUT

4 3 2 1 0/4 3 2 1 0/4 3 2 1

IV] Mode 3: SQUARE WAVE GENERATOR

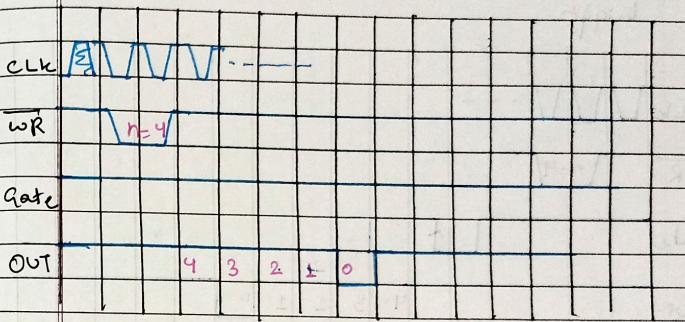
- If the gate is 1, then the counting is enabled, otherwise it is disabled.
- This mode is used to generate the square wave.
- The time period is equal to count. If count is even, the on-time of wave is $\frac{\text{count}}{2}$. Otherwise on-time is $\frac{(\text{count}+1}){2}$ and off time is $\frac{(\text{count}-1)}{2}$.



V] Mode 4: Software TRIGGERED STROBE

- If the gate is 1, then the counting is enabled, otherwise it is disabled.
- Initial OUT value is high, and goes low when count is at the last stage.

→ The count is reloaded again for subsequent clock pulse.



VI] Mode 5: Hardware Triggered strobe

Mode 0 gives a continuous output whereas, Mode 4 gives a strobe output.

VI] Mode 5: Hardware Triggered strobe

- Initially, the OUT is high.
- The counting is triggered by the rising edge of the GATE.
- When initial count is expired the OUT becomes low for one clock pulse, then high again.
- After writing the control word and the

II initial count, the counter will not be loaded until clock pulse after one trigger.

CLK

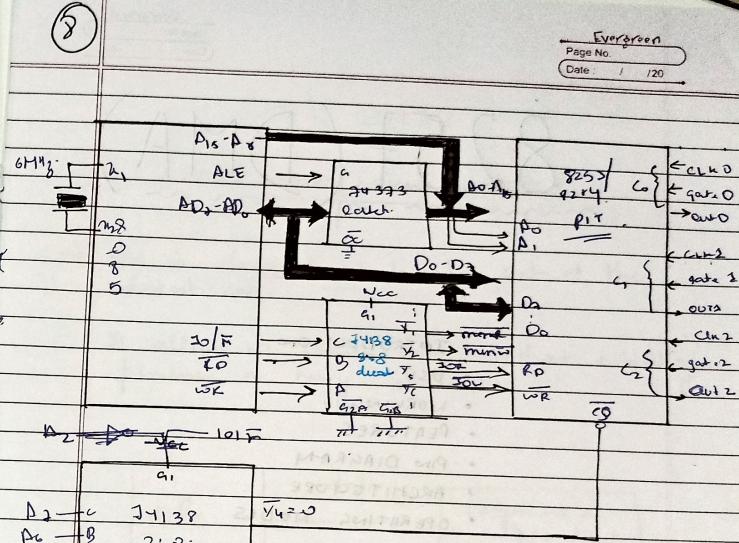
$$\overrightarrow{wR} \quad n=4$$

gate

OUT -

4 3 2 1 0

Intel Intel 8254 PIT interfacing with 8085



$\Delta_2 - c$ 74138
 $\Delta_6 - B$ 3 : 8
 $\Delta_5 - A$ $\overline{C2A}$ $\overline{G2B}$

▷ J/O May, d₁ 8254

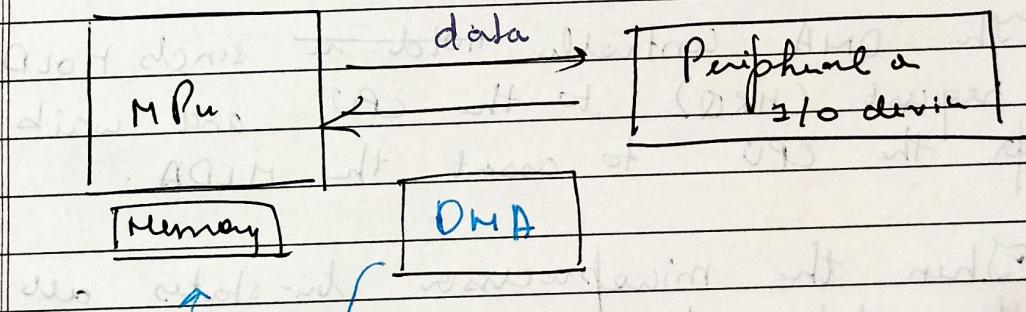
Assuming the address to be : 80.

	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A
C ₀	1	0	0	0	0	0	0	0
C ₁	1	0	0	0	0	0	1	81
C ₂	1	0	0	0	0	0	10	82
CW	1	0	0	0	0	0	11	83

8257 (DMA)

(direct memory access controller)

- It is designed to transfer data at the fastest rate.
- It allows to transfer data without subjecting processor to a heavy overhead.



Using DMA controller:

- Using a DMA controller, the device requests the CPU to hold its data, address and control bus, so the device can be free to transfer data directly to/ from the memory.

need of DMA controller:

→ whenever we need to transfer a large amount of data from one device to another device.

How DMA operations are performed?

- 1) Initially when any device has to send data between the device and the memory, the device has to send DMA request (DRQ) to DMA controller.
- 2) The DMA controller sends HOLD request (HREQ) to the CPU and waits for the CPU to assert the HLDA.
- 3) Then the microprocessor tri-states all the data bus, address bus and control bus. The CPU leaves the control over bus and acknowledges the HOLD request through HLDA signal.
- 4) Now the CPU is in HOLD state and the DMA controller has to manage the operations over bus between the CPU, memory, and I/O devices.

(9)

Features of 8257

- 1) It has four channels which can be used over four I/O devices.
- 2) Each channel has 16 bit address and 14-bit counter.
- 3) Each channel can transfer data up to upto 64 kb.
- 4) Each channel can be programmed independently.
- 5) Each channel can perform read transfer, write transfer and parity transfer operations.
- 6) It generates MARK signal to the peripheral device that 128 bytes have been transferred.
- 7) It requires a single phase clock.
- 8) Its frequency ranges from 250 Hz to 3 MHz.
- 9) It operates in two modes i.e., Master mode and Slave mode.

- 10) In Master Mode, the control of the buses is with the DMA and DMA is going to transfer the data from the CPU to I/O device or vice versa.

(control of the buses are w/ DMA)

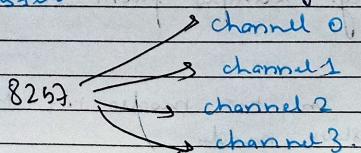
- 11) In slave mode, the control of the buses is w/ microprocessor and CPU is going to send status of register and address to DMA, etc. Thus, DMA is acting as a slave to the CPU.

PIN DIAGRAM : 8257

IOR	1	40	$\rightarrow A_7$
IOW	2	39	$\rightarrow A_6$
MENR	3	38	$\rightarrow A_5$
MENW	4	37	$\rightarrow A_4$
MARL	5	36	$\rightarrow TC$
READY	6	2	$\rightarrow A_3$
ULDA	7	35	$\leftrightarrow A_2$
ADSTB	8	5	$\rightarrow A_1$
AEN	9	34	$\leftrightarrow A_0$
HRO	10	7	$\rightarrow V_{CC}$
CS	11	30	$\leftrightarrow D_0$
CLK	12	29	$\leftrightarrow D_1$
RESET	13	21	$\leftrightarrow D_2$
DACK2	14	28	$\leftrightarrow D_3$
DACK3	15	26	$\leftrightarrow D_4$
DRQ3	16	25	$\rightarrow DACK_0$
DRQ2	17	24	$\rightarrow DACK_1$
DRQ1	18	23	$\leftrightarrow D_5$
DRQ0	19	22	$\leftrightarrow D_6$
IND	20	21	$\leftrightarrow D_7$

Pin diagram and architecture of DMA 8257 DMA controls

- 4) DMA manages several DMA channels, each of which can be programmed to perform a sequence of three DMA transfers.



2) DRQ₀ - DRQ₃:

→ These are the four individual channels DMA request inputs, which are used by the peripheral device for using DMA service.

→ When the fixed priority mask is selected, then DRQ₀ has the highest priority and DRQ₃ has the lowest priority among them.

2) DACK₀ - DACK₃:

→ These are the active low DMA acknowledge lines, which update the requesting peripheral about the status of their request by the CPU.

→ These lines can also act as sticker lines for the requesting device.

3) D₀ - D₇:

→ There are bidirectional data lines which are used to interface the system bus w/ the internal data bus of DMA controller.

→ In the slave mode, it carries command words to 8257 and status word from 8257.

- In the master mode, these lines are used to send higher byte of the generated address to the latch.
- The address is further latched using ADR558 signal.

4) IOR:

→ It is an active-low bidirectional tri-state input line, which is used by the CPU to read internal registers of 8257 in the slave mode.

→ In the master mode, it is used to read data from the peripheral devices during a memory write cycle.

5) IOW:

→ It is an active low bi-directional tri-state line, which is used to load the contents of the data bus to the 8 bit mode register or upper/lower byte of a 16 bit DMA address register or terminals count register.

→ In the master mode, it is used to load the data to the peripheral devices during DMA memory read cycle.

6) CLK

- It is a clock frequency signal which is required for the internal operation of 8257.

7) RESET:

- This signal is used to reset the DMA controller by disabling all the DMA channels.

8) A₀-A₃:

- These are the four least significant address lines.

→ In the slave mode, they act as an input, which selects one of the registers to be read or written.

→ In the master mode, they are the four least significant memory address output lines generated by 8257-12.

9) CS:

- It is an active-low chip select line.

→ In the slave mode, it enables the read/write form operations to/ from 8257.

→ In the master mode, it disables the write/read operations to/ from 8257.

10) A₄-A₇:

- These are the higher nibble of the lower byte address generated by DMA in Master Mode.

11) Ready:

- It is an active-high asynchronous input signal, which makes DMA ready by inserting wait states.

12) HRQ:

- This signal is used to receive the read request signal from the output device.

→ In the Slave mode, it is connected w/ an ORP input line 8257.

→ In Master mode, it is connected w/ HOLD input of the CPU.

13) ALDA:

→ It is the held acknowledgement signal which indicates the DMA controller that the bus has been granted to the requesting peripheral by the CPU when it is set to 1.

14) MEMR:

→ It is the low memory read signal, which is used to read the data from the addressed memory locations during DMA read cycles.

15) MEMW:

→ It is the active-low write state signal which is used to write the data to the addressed memory locations during DMA read cycles (write operation).

16) ADST:

→ This signal is used to convert the high byte of the memory address generated by the DMA controller into the latches.

17) AEN:

→ This signal is used to disable the address bus / data bus.

18) TC:

→ It stands for 'Terminal count' which indicates the present DMA cycle to the present peripheral devices.

19) MARK:

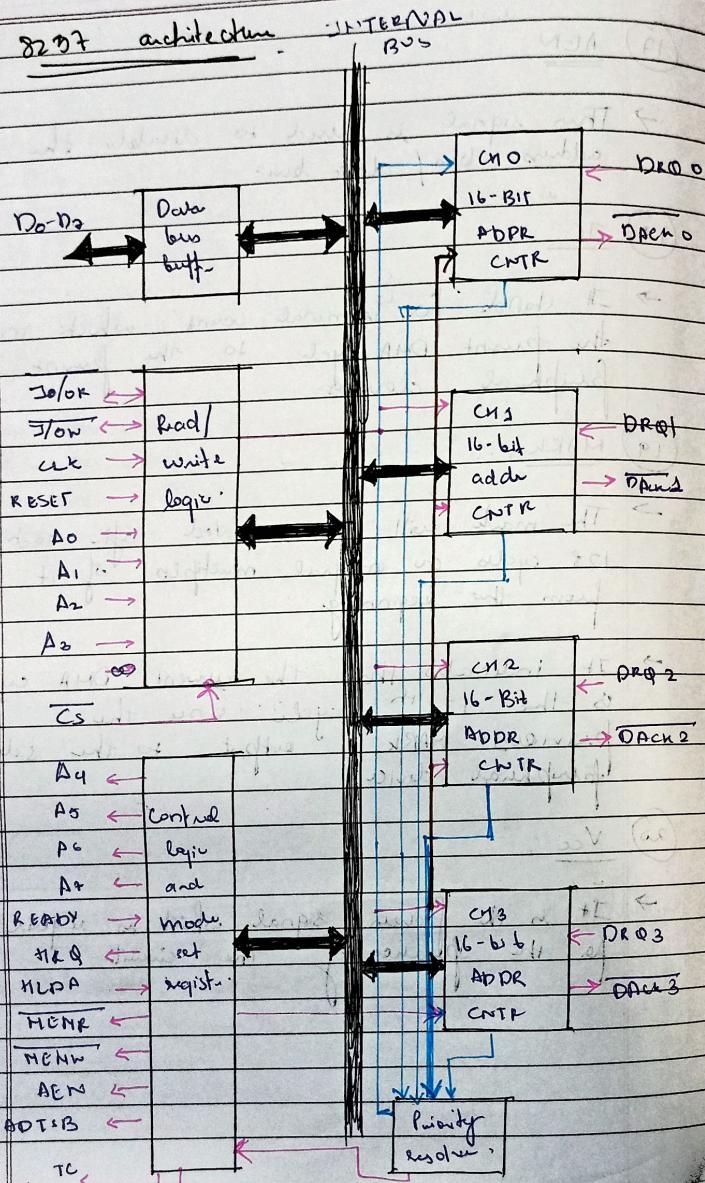
→ The mark will be activated after each 128 cycles or integral multiple of it from the beginning.

→ It indicates that the current DMA cycle is the 128th cycle since the previous MARK output to the selected peripheral device.

20) Vcc:

→ It is the power signal which is required for the operation of the circuit.

8237 architecture



Operating Modes of 8257 (DMA)

2) There are two types of register:

1) Address Register (16 bit)

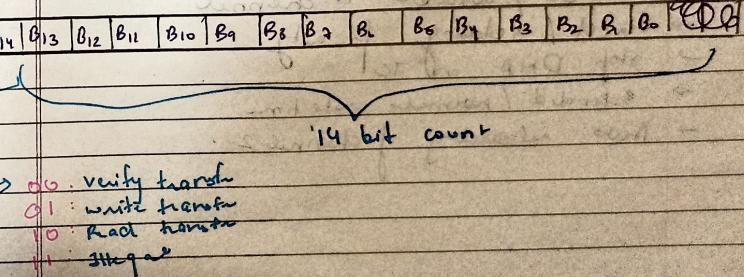
Address register is used to store the starting address of memory location for DMA (data transfer).

The address in address register is automatically incremented after every read/write/verify transfer operation.

means that data has reached to its designated destination.

2) Count register : (16 bit)

used to count the no. of byte or word transferred by DMA.



- $B_0 - B_{13}$: 14 bit count value.
- $B_{14} - B_{15}$: used to indicate the type of DMA transfer (Read / write / verify).
- In read transfer → data is transferred from memory to I/O device.
- In write transfer → data is transferred from I/O device to memory.
- Verification operations generate the DMA address without generating the DMA memory and I/O control signals.

→ 8 bit register of 8257 :

(i) Mode set register :

B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
AL	ICS	EW	RP	EN3	EN2	EN1	ENO

uses :

- Enable / disable a channel
- Fixed / rotating priority
- Stop DMA on TC
- Extended / normal write time
- Auto re-loading of channel 2

(12)

→ B_7 : AL

auto re-loading of channel 2

- 1: enable auto reload
- 0: disable auto reload

(Auto re-loading means that when the bit controls 1 is 1, then the contents of channel no. 3, count register and the status register, they are loaded in channel no. 2 whenever the channel 2 reaches terminal count.)

(no. of channels are reduced)

→ B_6 : TCS (Terminal count stop)

- 1: stop DMA on TC.

→ used to stop the operation of DMA when terminal count is reached.

→ B_5 : EW

→ when it is set to 1, then the timing of low write signals (MW and ZONE) will be extended.

$B_3: 1 \rightarrow$ extended write selection
 $0 \rightarrow$ normal write selection.

$\rightarrow B_4: RP$

used to
if bit B_4 is set to 1; then the channels will have rotating priority.

If it is 0, then the channels will have fixed priority.

\rightarrow In rotating priority: after servicing a channel its priority is made at lowest.

\rightarrow In fixed priority: channel 0: highest priority
channel 1: medium priority
channel 2: lowest priority.

$\underline{B_5, B_6, B_7, B_8}$:

used to enable / disable the four channels

$B_3: EN_3$

1: enables channel 3
0: disable channel 3

$B_2: EN_2$

1: enable channel 2
0: disable channel 2

$B_1: EN_1$

1: enable channel 1
0: disable channel 1

$B_0: EN_0$

1: enable channel 0
0: enable channel 0.

(2) Status Register:

B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
0	0	0	up	T_{C3}	T_{C2}	T_{C1}	T_{C0}

$\rightarrow B_0:$

1: channel 0 has reached terminal count.

$\rightarrow B_1:$

1: channel 1 has reached TC

$\rightarrow B_2:$

1: channel 2 has reached TC

$\rightarrow B_3:$

1: channel 3 has reached TC.

→ B₄:

1: channel 2 selected from channels

⇒ status bits are cleared after a read operation by RP.

8279 : INTRODUCTION

- 8279 programmable keyboard / display keyboard controller is designed by Intel that interfaces a keyboard / w/ the CPU.
- The keyboard first scans the keyboard and identifies if any key has been pressed.
- It then sends the response of the pressed key to the CPU and vice-a-versa.

Methods of interfacing keyboard w/ CPU:

- Mainly there are two ways by which keyboard can be interfaced with the CPU.

1) Interrupt Mode :

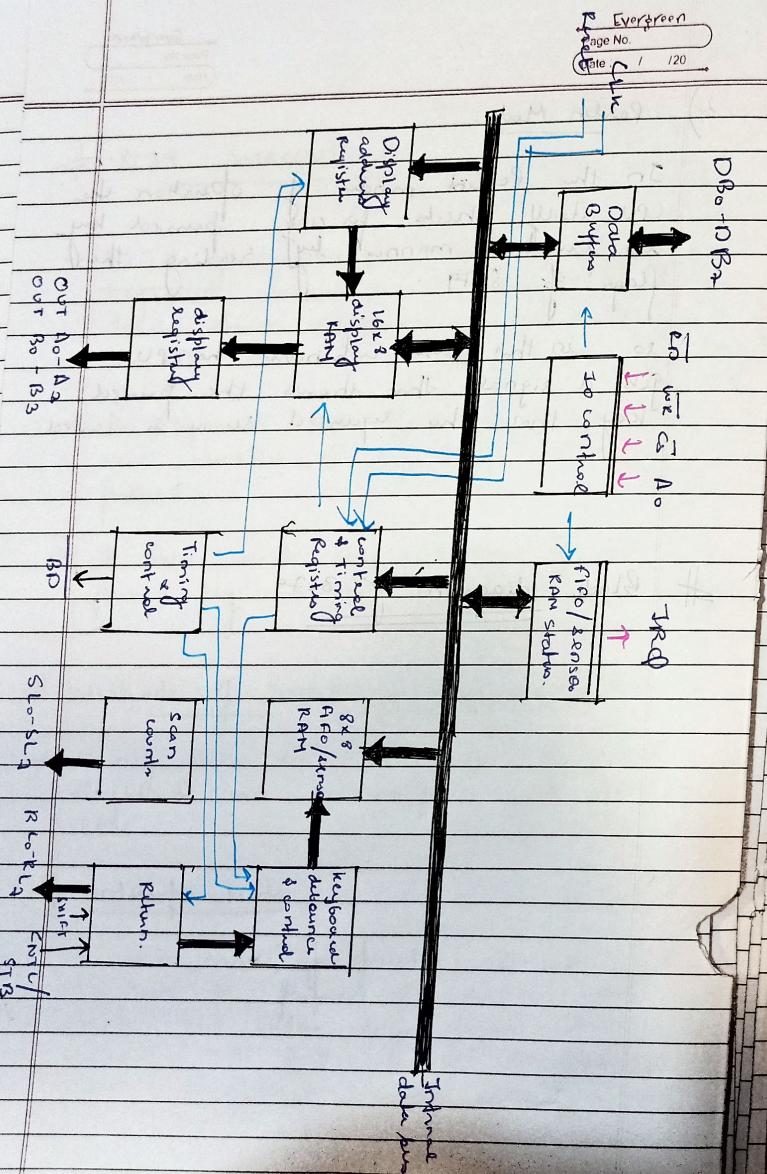
In this mode of operation, the CPU continues to perform its original task until and unless any request is generated to that a key is pressed and is required to be serviced.

2) Polling Mode:

In the Polling mode of operation the CPU itself checks for any pressed key in periodic manner by reading the flags of 8279.

so, in this case whenever the CPU gets a signal that shows the pressed key then the requested service is served.

Block diagram of 8279 :



(14)

- I/O control & Buff data buffer
- This unit controls the flow of data through the up.
- It is enabled only when D is low.
- The data buffer interfaces the external buses of the system w/ the internal buses of the up.
- The pins Ao, RD and WR are used for command, status or data read/write operations.

Control and timing register and Timing control :

- This unit contains registers to store the keyboard, display mode, and other operations as programmed by the CPU.
- The timing & control unit handles the timing for the operation of the circuit.
- Scan counter :
- It has two modes :

1) Encoded Mod.

The counter provides the binary code count that is to be entirely decoded to provide scan lines for the keyboard & display.

2) Decoded Scan Mod.

The counter internally divides the most significant 2 bits and provides a decoded 1 out of 4 scan on $SL_0 - SL_3$.

3) Return Buffers, keyboard debouncing and control:

→ This unit first scans the key closure row-wise, if found then the keyboard debouncing unit debounces the key entry.

→ In case, the same key is detected, then the code of that key is directly transferred to the sense RAM along w/ SHIFT & control key status.

4) FIFO / sense RAM & status logic:

→ This unit acts as 8-bit first-in-first-out (FIFO) RAM where the key codes of every pressed key is entered into the RAM w/ their sequence.

→ The status logic generates an interrupt request after each FIFO read operation till the FIFO gets empty.

→ In the scanned sense matrix mode this unit acts as sense RAM when its each row is loaded w/ the status of their corresponding row of sensors into matrix.

→ When the sensor changes its state, the JRD line changes to high and interrupts the CPU.

5) Display address register and display RAM:

→ This unit consists of display address registers which holds the address of the word currently read/written by the CPU to/from the display RAM.

PIN DIAGRAM : 8279

Page No. _____
Date / / 120

RL ₂	1	9 ₀	- V _{CC} (6.6V)
RL ₃	2	3 ₉	- RL ₁
CLOCK	3	3 ₀	- RLO
JRQ	4	3 ₂	- CNTL/STB
JKLY	5	3 ₆	- SHIFT
RL ₅	6	8	3 ₅ - SL ₂
RL ₆	7	2	3 ₄ - SL ₂
RL ₇	8	2	3 ₃ - SL ₁
RESET	9	2	3 ₂ - SLO
RD	10	7	3 ₁ - OUT B ₀
WR	11	7	3 ₀ - OUT B ₁
DB ₀	12	9	2 ₉ - OUT B ₂
BB ₁	13	9	2 ₈ - OUT B ₃
DB ₂	14	9	2 ₇ - OUT A ₀
DB ₃	15	9	2 ₆ - OUT A ₁
DB ₄	16	9	2 ₅ - OUT A ₂
DB ₅	17	9	2 ₄ - OUT A ₃
DB ₆	18	9	2 ₃ - B ₀
DB ₇	19	9	2 ₂ - CS
V _{SS} (OV)	20	9	2 ₁ - A ₀

15

Evergreen
Page No. _____
Date / / 120

① Data bus lines , DB₀ - DB₇:

There are 8 bidirectional data bus lines used to transfer the data to / from the CPU.

② CLK:

The clock input is used to generate internal timings required by microprocessor.

③ RESET :

As the name suggests, this pin is used to reset the microprocessor.

④ CS chip select :

When this pin is set low, it allows read / write operations, else this pin should be set to high.

⑤ A₀ :

This pin indicates the transfer of command / status information. When it is low, it indicates the transfer of data.

⑥ RD, WR:

This read / write pin enables the data buffer

to send/receive data over the data bus.

(7) IRO:

This interrupt output line goes high when there is data in the FIFO sense RAM.

The interrupt line goes low w/ each FIFO Ram read operation.

However, if the FIFO RAM further contains any key code entry to be read by the CPU, this pin again goes high to generate an interrupt to the CPU.

(8) V_{ss}, V_{cc}:

These are the ground & power supply lines of the MCP.

(9) S_{LO}-S_{LA}:

These are the scan lines used to scan the keyboard matrix and display the digits.

These lines can be programmed as encoded or decoded, using the mode control register.

(10) R_{L0}-R_{La}:

These are the return lines used to scan the keyboard matrix & display the digits.

These lines can be programmed as encoded or decoded, using the mode control register.

(11) SHIFT:

The shift input line status is stored along w/ every key code in FIFO in the scanned keyboard mode.

Till it is pulled low w/ a key closure, it is pulled up internally to keep it high.

(12) CNTL/STB - CONTROL/STROBED I/P MODE:

In the keyboard mode, this line is used as a control input & stored in FIFO on a key closure.

The line is a strobe line that enters the data into FIFO Ram, in the strobed input mode.

It has an internal pull up. The line is pulled down w/ a key closure.

(13) BD:

It stands for blank display. It is used to blank the display during digit switching.

(14) OUT A₀ - OUT A₃ and OUT B₀ - OUT B₃

These are the output pins for two 16x4 or one 16x8 internal display refresh registers.

The data from these lines is synchronized w/ the scan lines to scan the display & the keyboard.

Operating modes of 8279

(1) Input mode:

The input, which is provided by the keyboard to the system specifies the input mode.

This mode is classified into the following categories:

(16)

(2) Scanned keyboard Mode :

In this operating mode, the key matrix is interfaced w/ either encoded or decoded scan.

In encoded scan, 8x8 keyboard is interfaced while in the decoded scan, 4x8 keyboard is interfaced.

The keycodes are stored in the FIFO RAM.

(3) Scanned sensor matrix:

This helps in interfacing the sensor array w/ 8279 by making use of an encoder or decoder scan.

Similar to scanned keyboard mode, 8x8 sensor matrix for encode scan and 4x8 sensor matrix interfacing for decode scan.

(4) Strobed input:

This mode of operation, if correct is not specified by the processor & the control line shows low signal then the data present on the return lines is stored in FIFO RAM byte by byte.

2) Output Modes

This mode is also known as display mode.

It is further classified into two modes.

This mode helps in selecting the display options.

a) Display scan

The 8229 generates 8 or 16 characters multiplexed displays that are organized in either dual 4-bit or single 8-bit display units.

b) Display entry

The data which gets displayed can be either displayed starting from either the right side or left side.

3) Display Modes

This mode is associated w/ data display and has two further classifications.

a) Left entry mode

→ It is also known as typewriter mode.

→ In this, the first type of character is present at the left-most position while further incoming characters appear successively towards the right.

→ This means data begins to appear from the left side of the display unit.

→ So, the bit value at address 0 in the display RAM will appear at the left-most position whereas the bit value at address 15 will appear at the right-most position.

b) Right entry mode

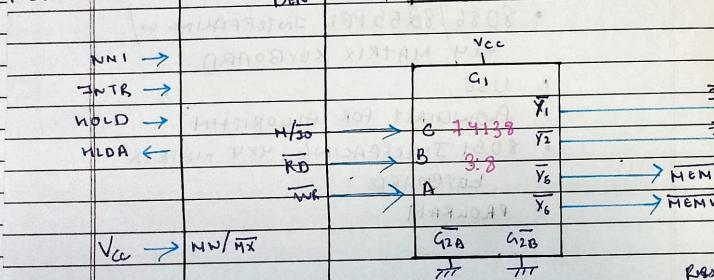
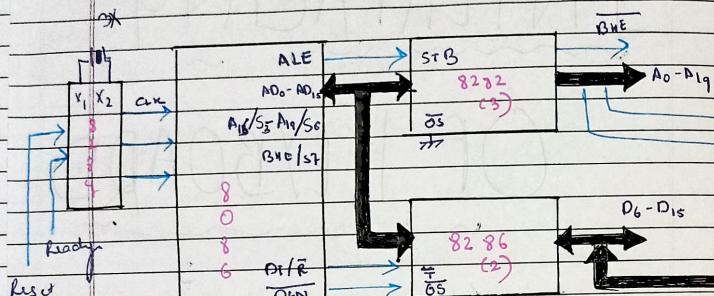
→ This mode is also known as calculator mode.

→ This is so because in the calculator the first entered character appears at the right-most position and then as successively when a new character has entered the position of the former one is shifted towards the left.

→ Thus, in this mode, the first entry

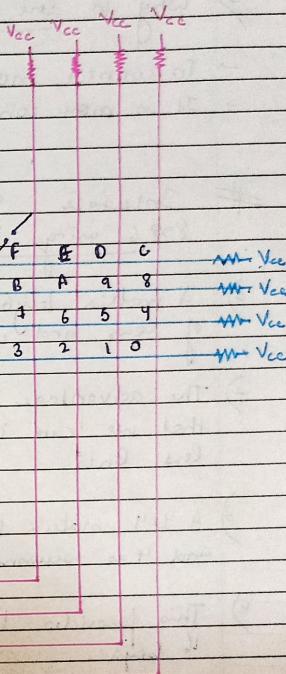
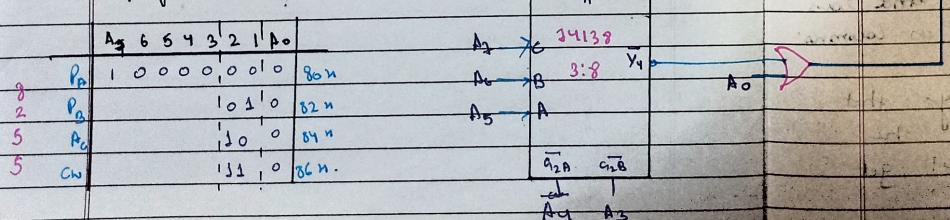
will appear at the eightmost position
but as soon as a new entry is made
then the previous one will get shifted
towards the left by one & the present
entry will take the eightmost pos.

8086 - 80-8255 PPI interfacing with a 4x4 matrix keyboard.



Reset (8/84) → Reset CS

I/O Map of 8255 :



Q) Why to use a Matrix keyboard?

- To connect more keys w/ less no. of lines.
- It is more complex than a normal keyboard.

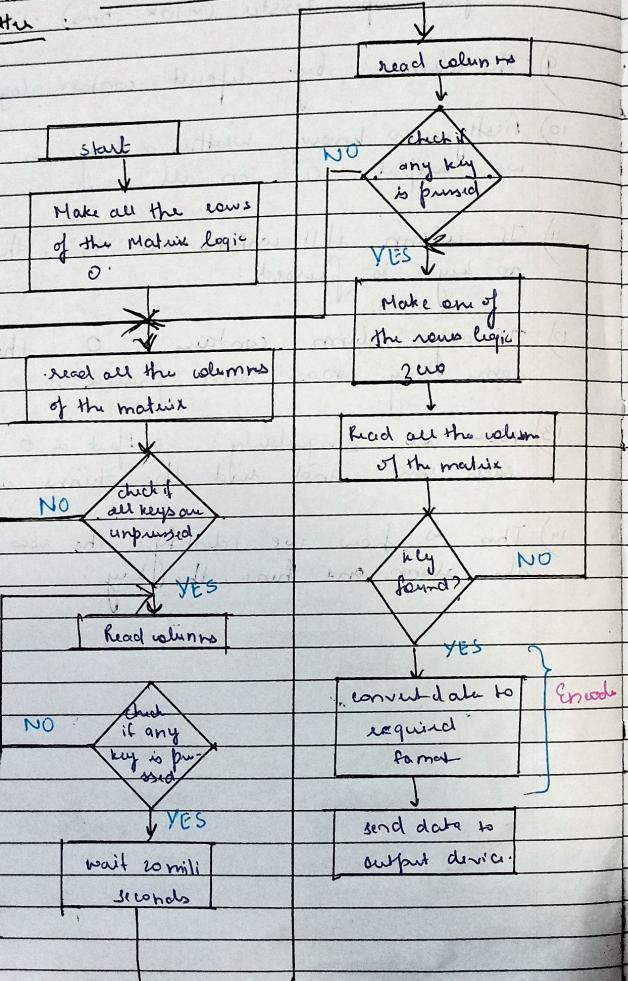
Interface a 4×4 matrix keyboard to 8086 using 8285:

- 1) A matrix keyboard is formed by combination of rows and columns.
- 2) The advantage of a matrix keyboard is that we can interface more keys using less lines.
- 3) A 4×4 matrix keyboard uses 4 lines as rows and 4 as columns.
- 4) This provides 16 intersecting points to connect 16 keys.
- 5) The rows are used as outputs and columns as inputs.
- 6) 4 lines of Port A are used as ~~and~~ rows and 4 lines of Port B are used as columns.
- 7) Keys are connected in such a way that only when a key is pressed, the corresponding row and column will get connected.

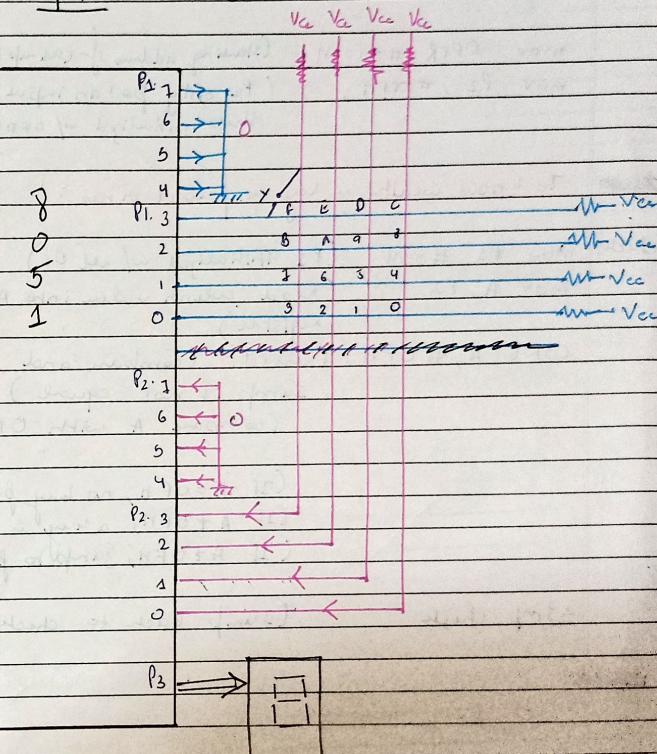
Q) The columns are connected to V_{cc} via pull up resistors ($\sim 10^4$ ohms).

- 8) The columns by default contain logic '1'.
- 9) Firstly, to know whether a key is pressed, we output "0" on all rows.
- 10) If column still contains all '1's, then no key is pressed.
- 11) If any column contains a 0, then some key has been pressed.
- 12) Now we singularly output a 0 on each row and read the column again.
- 13) This is how we identify the row and the column and hence, the key.

A flow-chart to understand the algorithm better:



8051 interfacing 4x4 matrix keyboard + program



Program:

```
mov DPTR, #0400H; (starting address of look-up table)
mov P2, #0FH; (P2: only port an input,
thus initialized w/ 0FH)
```

To know whether a key is pressed or not:

```
check: mov P1, #00H (P1 initialized w/ all 0s)
       mov A, P2 (read column value into A
       register)
       CJNE A, #0FH, Pressed. (compare and
       jump if not equal)
       (compare A with 0FH)
```

- (If A=0FH, no key pressed)
- (If A+OFH, a key is pressed)
- (If A+OFH, jump/no pressed)

Simp check

(Jump back to check)

Pressed:

// checking row 1:

```
Pressed: MOV R0, #00H
         MOV P1, #0EH
         MOV A, P2
         CJNE A, #0F, colcheck.
```

(19)

// checking row 2:

```
MOV R0, #04H
MOV P1, #0DH
MOV A, P2
CJNE A, #0FH, colcheck.
```

// row checking row 3:

```
MOV R0, #08H
MOV P1, #0BH
MOV A, P2
CJNE A, #0FH, colcheck.
```

// checking row 4:

```
MOV R0, #0CH
MOV P1, #07H
MOV A, P2
```

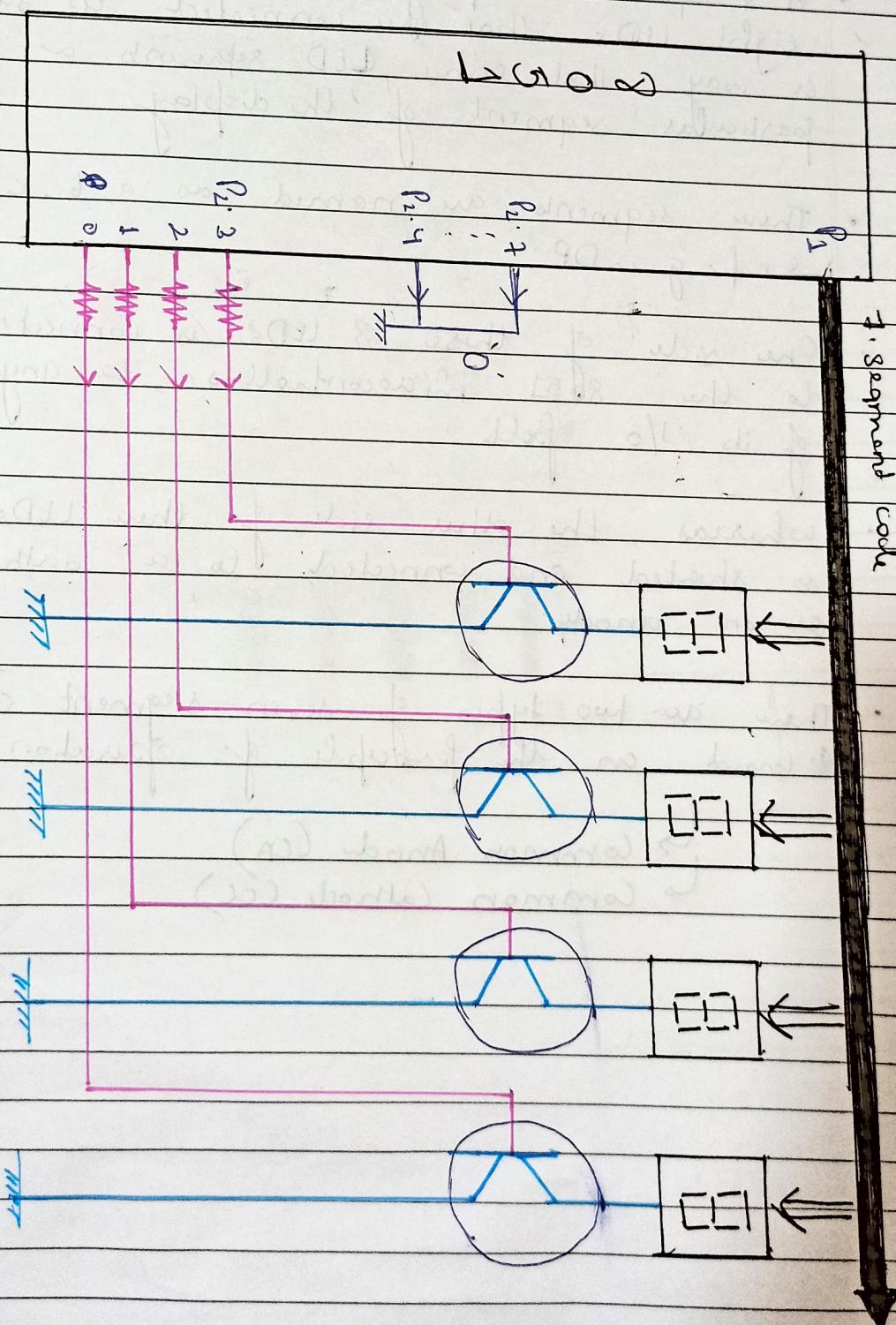
checking comparing A and to know which
column.

```
colcheck: RRC A (Rotate right)
          JNC Display (If CF=0, jump to display)
          JNC R0 (R0 <= R0+i)
          SIMPL colcheck
```

```
display: MOV A, R0
         MOVC A, @A+DPTR
         MOV P3, A
         SJMMP check.
```

END

Intufacing 4 digit 7 segment multiplexed LED Display :

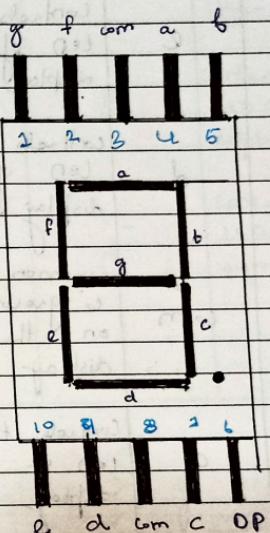


More about 7-segment display:

- A 7 segment display is a combination of eight LEDs that are connected in such a way that each LED represents a particular segment of the display.
- These segments are named as a, b, c, d, e, f, g, DP.
- One side of these 8 LEDs is connected to the 8051 microcontroller, to any of its I/O pins.
- Whereas, the other side of these LEDs is shared and connected to a cathode or an anode.
- There are two types of seven-segment displays based on the principle of operation.
 - Common Anode (CA)
 - Common Cathode (CC)

(20)

PIN diagram of 7-segment LED:



Pin number	Pin Name	Description
1		Controls the left bottom LED of the 7 segment display.
2	d	Controls the bottom most LED of the 7 segment display.
3	com	Common pin connected to ground / Vcc based on the type of display.
4	c	Controls the right bottom LED of the 7 segment display.
5	DP	Controls the decimal point LED of the 7 segment display.
6	b	Controls the top right LED of the 7 segment display.
7	a	Controls the top most LED of the 7 segment display.

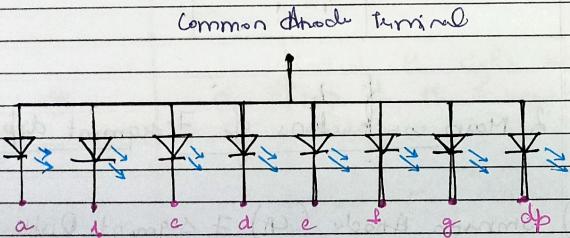
8	com	Common pin connected to ground / Vcc based on the type of display.
9	f	Controls the top left LED of the 7 segment display.
10	g	Controls the middle LED of the 7 segment display.

2 Main constructions of 7-segment display :

(1) Common Anode (CA) 7 segment Display :

→ In the common anode type of 7 segment display, the anodes of all the LEDs are joined together to Vcc supply with a maximum of 10mA current whereas, the cathodes of these LEDs are connected to the I/O pins of the microcontroller.

- we know that an LED turns on when forward biased and off when reverse biased.
- Now since the anode is connected to +5V to make the LED forward biased, the cathode should be at 0V, i.e., the LED turns on when a low pulse from the microcontroller is given.
- In summary, to switch on a common anode type's segment, you've to write a '0' to its corresponding pin.



Ground these pins to enable each segment

(2) Common cathode (cc) 7 segment Display:

In common cathode seven segment display one side, i.e., the cathode part of all of the eight light-emitting diodes is shared together and connected to the ground / GND.

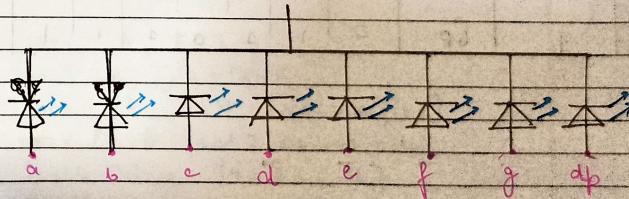
The other side i.e., the anode part, is connected to the microcontroller 10 pins.

When we give a high pulse through the pins of the microcontroller, the LED turns ON.

Hence, we can say that a controller common cathode type display is active high.

In summary, to switch on a common cathode type's segment, you've to write a '1' to its corresponding pin.

Common cathode terminal



Connect these pins to source to enable each segment

7 segment display table:

(common cathode method used)

character	HEX code	cc	g	f	e	d	c	b	a
0	3F	0	0	1	1	1	1	1	1
1	06	0	0	0	0	0	1	1	0
2	6B	0	1	0	1	1	0	1	1
3	4F	0	1	0	0	1	1	1	1
4	66	0	1	1	0	0	1	1	0
5	6D	0	1	1	0	1	1	0	1
6	7D	0	1	1	1	1	1	0	1
7	0A	0	0	0	0	1	1	1	1
8	7F	0	1	1	1	1	1	1	1
9	6F	0	1	1	0	1	1	1	1

→ all these codes are stored in the memory in a form of a table called "look up table".

Look up table

initialized D PTR
to the starting

address of the ← D PTR → 0400
look up table

0400	0'
0401	1'
0402	2'
⋮	⋮
0408	8'
0409	9'

→ address from the internal
ROM.

→ It is stored in ROM because it is a permanent table.

Program to implement the Display of '2013'.

mov D PTR, #0400H.

Again: mov A, #02H

mov P1, @A+D PTR

mov P2, 09H (select digit 1)

mov P3, A (and 7 segment code)

Acq Delay (5 mil second)

} display 2

```
mov A, #00H  
movc A, @A+DPTR  
mov P2, #04H  
mov P1, A  
A call Delay
```

```
mov A, #01H  
movc A, @A+DPTR  
mov P2, 02H  
mov P1, A  
A call Delay
```

```
mov A, #08H  
movc A, @A+DPTR  
mov P2, 01H  
mov P1, A  
A call Delay
```

SJmp Again.

display

display 1

display 2

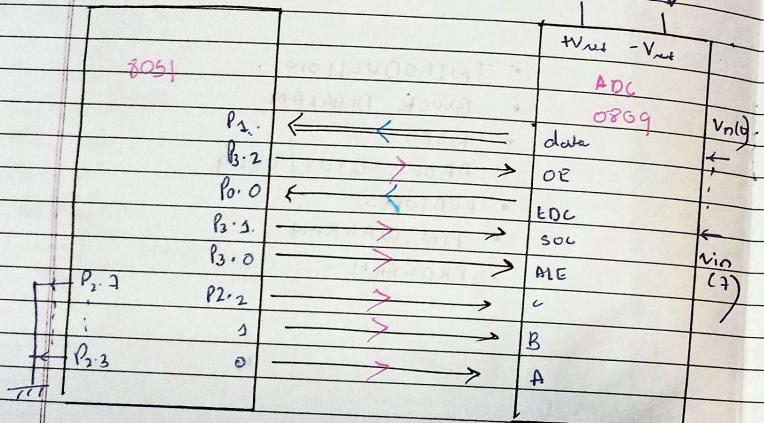
Analog:

Analog means any information represented in the form of a voltage.

Digital :

Digital means any information represented in the form of a code.

ADC :



2) NEED of ADCs :

→ The physical parameters that any microcontroller processes come from sensors. Sensors are transducers that convert a physical parameter like temperature into

Date / / 120
electrical signals the microcontroller can understand.

→ But here is the issue → analog sensors output data in an analog format which a microcontroller cannot understand.

Therefore, to convert this analog date to a digital format, Analog to digital converter or ADC's are used.

ADC is the analogy to digital converter which converts analog data into digital format; usually it is used to convert analog voltage to digital format.

→ An analog signal has a continuously changing amplitude w/ respect to time. A digital signal, on the other hand, is a stream of 0s and 1s.

→ Instead of continuous conversion, ADC converts data periodically, which is usually known as sampling data rate.

► An ADC maps analog signals to their binary equivalents. To do this, ADCs use various methods like Flash conversion, Slope integrators, or successive approximation.

→ Telephone modem is one of the examples of ADC, which is used for internet, it converts analog data into digital data, so that computer can understand because computer can only understand digital data.

→ The major advantage of using ADC is that, noise can be efficiently eliminated from original signal and digital signals can travel more efficiently than analog one.

That is why digital audio is very clear, while listening.

→ In present time, there are lots of microcontrollers in market which have built-in ADC with one or more channels.

but, some IoT applications require an external ADC as they provide better accuracy and higher speeds compared to internal ADCs.

→ When we select 8051 microcontroller family for making any project, in which we need of an ADC conversion, then we use External ADC.

→ Some external ADC chips are: 0803, 0804, 0808, 0809 etc.

About ADC 0808 / 0809 :

→ ADC 0808 / ADC 0809 is an 8-bit successive approximation ADC which is multiplexed among 8 @ input pins.

→ ADC 0809 data acquisition component is a monolithic CMOS device w/ an 8-bit analog-to-digital converter, 8-channel multiplexer & microprocessor compatible control logic.

→ It can measure up to eight ADC values from 0 to 5 volt since it has eight channels.

→ When voltage reference is +5V, its step size will be 19.53 mV. That is, for every increase of 19.53 mV on the input side there will be an increase of 1 bit at the output side.

→ 7 # Features of ADC 0809 :

- 1) Easy interface to all microprocessors
- 2) Operates Ratiometrically or
no zero or full-scale adjustment required
- 3) 8-channel multiplexer with address logic
- 4) OV to VCC input range
- 5) Outputs meet TTL voltage level specifications.

ADC 0809

- 6) 28-pin molded chip carrier package

Pin diagram

In3	IN2
In4	IN5
In5	IN6
In6	IN7
In7	Start
EOC	ALE
2^6	2^7 MSB
Output enable	2^2
Clock	2^{-3}
V _{cc}	2^{-4}
V _{ref(+)}	2^{-8} LSB
GND	V _{ref(-)}
2^3	2^{-6}

1) Address Lines (A, B, C) :

as mentioned earlier, ADC 0809 has 8-multiplexed ADC channels

A particular input channel is selected by using these address bits as shown below:

Evergreen
Page No. / 120
Date / /

ADC channel			
C	B	A	
0	0	0	IN0
0	0	1	IN1
0	1	0	IN2
0	1	1	IN3
1	0	0	IN4
1	0	1	IN5
1	1	0	IN6
1	1	1	IN7

2) Address latch enable (ALE):

a low to high signal at this pin sets latch will latch the above selected address & selected the respective channel for ADC conversion.

3) START conversion (SOC):

The A/D converter's successive approximation register (SAR) is reset on the +ve edges

(24)

of the Start conversion (SC) pulse. Thus, we need to generate a low-to-high pulse for starting the ADC conversion.

4) End of conversion (EOC):

Once the conversion is over, this pin is pulled HIGH by 0809.

This pin needs to be monitored for the conversion to complete & then read the data.

5) Output enable (OE):

ADC 0809 does the ADC conversion & holds the data in the internal registers.

A high signal on this pin will bring the data on the output lines.

#

Program :

mov R₂, #0AH
 mov R₀, #30H
 mov P₀, #0FFH
 mov P₁, #0FFH

(count)

(starting address)

} i/p ports.

(by giving the value
0FFH, the port
becomes an i/p port)

Back } CLR P_{3.0}
 CLR P_{3.1}
 CLR P_{3.2} } deactivating everything

mov P₂, #03H (giving the channel no.)

SETB P_{3.0} } latched the channel
 CLR P_{3.0}
 SETB P_{3.1} } → start conversion

wait: JNB P_{0.0}, ~~wait~~ (JNB : Jump if not Bit)
 (not bit means not set)
 (If B₀.P_{0.0}, jump to wait)
 (wait for conversion to get over)
 SETB P_{3.2} (output enable)
 (ask for data)

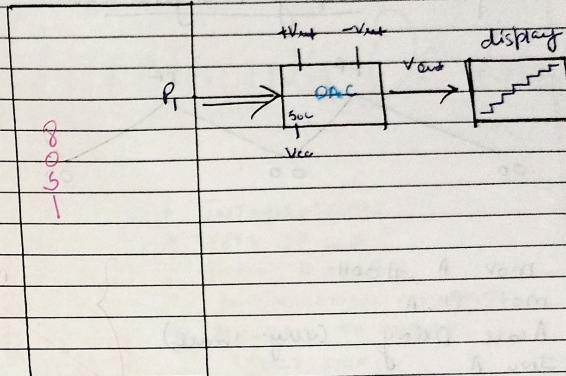
mov A, P₁mov @R₀, AINC R₀DJNZ R₁, Back

Here: SJMP Here

Interfacing with DAC (8031):

- The digital to analog converter (DAC) is a device, that is widely used for converting digital pulses to analog signals.
- There are two methods of converting digital signals to analog signals. These two methods are binary weighted method and R/2R ladder method.
- R/2R ladder method can achieve a much higher degree of precision.
- DACs are judged by its resolution. The bit resolution is a function of ~~also the~~ the no. of binary inputs. The most common input counts are 8, 10, 12 etc.
- No. of data inputs decides the resolution of DAC. So, if there are n digital input pins, then there are 2^n analog levels. So, 8 input DAC has 256 discrete voltage levels.

(25)

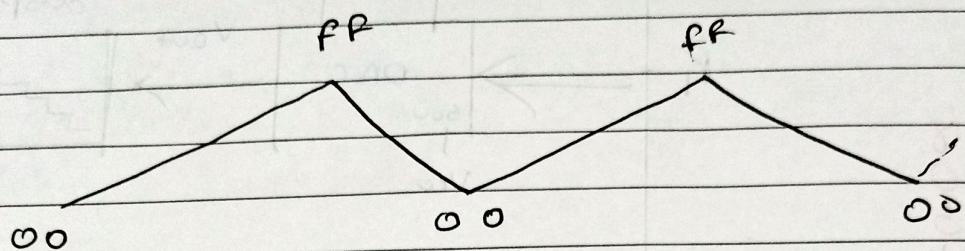


Program to display staircase :

```

mov A, #00H
Back: mov P1, A    (00, 01, 02, ... F2, FF)
      INC A
      ACALL delay
      JNZ Back   (Jump if not zero) { (when A = 0, don't jump)
Here: SJMP Here
  
```

Program to display Triangular waveform



```

Back1:    mov A, #00H
          mov P1, A
          A call Delay (very small)
          INC A
          JNZ Back 1
      
```

```

Back2:    MOV A, #0FEH
          mov P1, A
          ACALL Delay (very small)
          DEC A
          JNZ Back 2
          SJMP Back 1
      
```

LCD introduction

- LCDs are electronic devices that are used to display texts, custom characters and numbers.
- LCD is very simple and easy to understand for beginner and is commonly used in several electronic products.
- LCD (Liquid crystal display) provides a user-friendly interface and can be very useful for debugging purposes.
- The reason LCD is more popular than LED, seven segment displays. Because we can display characters, numbers & custom characters with ease.
- Although, we can also display pictures on LCDs. But result will be not comparable to graphical LCDs (GLCDs). GLCDs are used to display images.
- LCDs, they are available in different sizes & features.

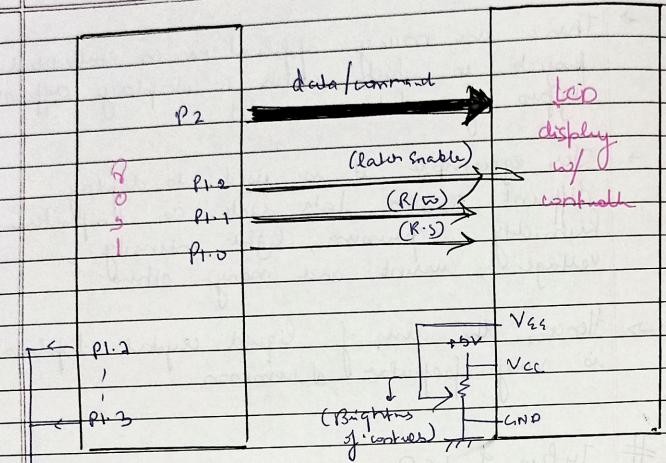
For eg: 16x2, 20x2 or 16x1 are different sizes available.

- There are many applications in embedded projects in which we have to display different types of data on LCD.
- For example, it is used to display different sensors data such as temperature, humidity, pressure, light intensity, voltage, current and many others.
- Hence, the use of liquid crystal displays is very popular & common.

Types of LCDs :

- LCDs come in 2 types:
- (1) serial LCD
 - (2) parallel LCD

- Serial type uses serial communication, to interface w/ 8051.
- They are easier to interface w/ 8051 than ILI LCDs. But they are expensive.
- In contrast, parallel LCDs are exceptionally used and interfacing w/ 8051 is done using digital I/O pins of 8051.



(1) ground: ground (0V)

(2) Vcc: supply voltage : 5V

(3) Vpp: contrast adjustment, through a variable resistor.

(4) RS (Register select): Select command register when low ; and data register when high.

(5) R/W (Read/write): low to write to the register ; High to read from the register.

(6) Enable: sends data to the data pins when a high to low pulse is given.

(7) DB₀ - DB₇: 8 Bit Data Pins

Initialization commands:

(1) 38H : Set up 2 line 5x7 matrix display

(2) 0FH : Display ON cursor ON cursor Blinking

(3) 01H : clear the display

(4) 06H : cursor increment mode (left to right)

(5) 80H : Cursor Home (line 1, position 1)

Steps to give command:

- 1) Put command on port
- 2) make RS = 0 (command)
- 3) make R/W = 0 (write)
- 4) make latch enable 1 0 0
- 5) wait delay : (10 ms)

steps to give date

- 1) Put date on bus
- 2) make R/C = 1 (data)
- 3) make R/W = 0 (write)
- 4) make latch enable = 1
- 5) Call delay (10 ms)

Look up table:

DPTR	Example
1	0400
1	0401
A	L
A	L
1	0
1	W
1	0
1	R
1	4
1	D

(28)

Program to display "HELLO WORLD".

MOV DPTR #0400H.

initialization

```
Cmd: MOV P2, A
      CLR P1.0
      CLR P1.1
      SETB P1.2
      CLR P1.2
      ACALL delay (10 ms)
      RET
```

} Function

initialization: MOV A, #38H

ACALL cmd

Mov A, #0FH

ACALL cmd

Mov A, #01H

ACALL cmd

Mov A, #06H

ACALL cmd

Mov A, #80H

ACALL cmd.

} Initialisation

display: MOV R2, #08H.

data: mov P2, A
SETB P1.0
CLR P1.1
SETB P1.2
CLR P1.3
Acall delay (10 ms)
RET

} Function.

display: Mov R7, #0BH (want to display
Mov RD, #00H 61 characters)

Back: mov A, R0
move A, @ A + DPTR
Acall Data.
Inc R0
D INJ R7, Back.

Here : SJMP Here

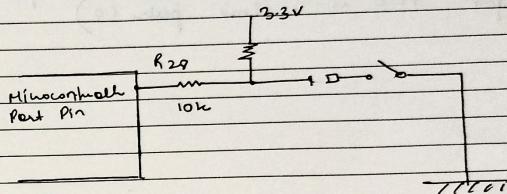
Making a lookup table:

Table: org 0400H (origin: 0400H)
DB 'HELLO WORLD'

A switch :

- The switch is an input device.
- When interfaced w/ microcontroller, it can be used to control other peripherals connected to the microcontroller.
- It basically "makes" the electrical circuit by establishing the flow of current. (or "breaks" it by interrupting the flow of current.)
- A switch, when not in use, has no definite value associated w/ it. Correspondingly, the pin of IC that it is attached to has no definitive value either.
- It is said to be floating b/w 0 and 1.
- Hence a resistor is connected b/w the switch & Vcc or to GND.
- To assign a specific value to it once and for all.
- Based on the position where the resistor is placed, it either acts as a pull up or a pull down resistor i.e., it provides 0 or a finite resistance.

→ Also, the value of this resistance is calculated, keeping in mind that any switch can handle currents in milli-amperes.



P

Positive logic

- When the resistor is connected to the ground, it is called a pull-down resistor.
- When the switch is off (not pressed), the input to pin P2.0 is a low pulse (0).
- When the switch is on (pressed), the input to pin P2.0 is high pulse (1).

Negative logic

When the resistor is connected to Vcc, it is called a pull-up resistor.

- When the switch is off (not pressed), the

input to pin P2.0 is HIGH pulse (1)

- When the switch is ON, the input to pin P2.0 is a low pulse (0)