MICROPROCESSORS AND MICROCONTROLLERS

PRACTICAL

1. **Generate number of Fibonacci series.**

LXI H,3050h

MVI C,08

MVI B,00

MVI D,01

MOV M,B

INX H

MOV M,D

jump: MOV A,B

ADD D

MOV B,D

MOV D,A

INX H

MOV M,A

DCR C

JNZ jump

HLT

1. **Clear all flags without using any data transfer instruction**

start: nop

XRA A ;resets S,C and AC.

INR A ;resets Z and P.

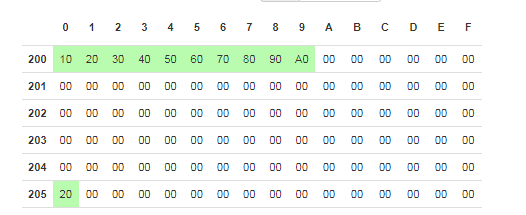
STC ;sets C=1

CMC ;compliments C

Hlt

1. **Program to search for a number in a list.**

LXI H,2050H ;Loads number to be found

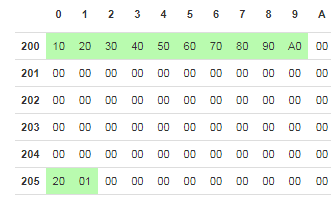
MOV B,M ;Stores it in B

LXI H,2000H ;Goes to start of list

MVI C,0AH ;Counter for list traversal

LOOP: MOV A,M

CMP B

JZ END ;Jumps if number found

INX H ;If not jumps to next num

DCR D ;Decrements counter

JNZ LOOP ;Goes back to comparing

LXI H,2051H ;Loads mem where o/p to be stored

MVI M,00H ;Stores zero if not found

HLT

END:LXI H,2051H

MVI M,01H ;Stores 1 if found

HLT

1. **Program to sort a list.**

;code

START: nop;

LXI H, 2000H ;this location stores size of list

MVI D,00H

MOV C,M

DCR C

INX H ;list begins at 2001

COMPARE: MOV A,M

INX H

CMP M

JC NEXT ;if first num smaller proceed to next

JZ NEXT ;if equal proceed to next

MOV B,M ;if not then swap

MOV M,A

DCX H

MOV M,B

INX H

MVI D,01H

NEXT: DCR C ;decrement after every iteration

JNZ COMPARE

MOV A,D

CPI 01H

JZ START

HLT

1. **Program to copy a list from one part of the memory to another**

START: nop ;Moving five number from 2000H to 2040H

MVI C,05

LXI H,2000H

LXI D,2040H

JUMP: MOV A,M

STAX D ;Stores contents of A in D

INX H ;goes to next in original list

INX D ;goes to next in where to store

DCR C ;Decrements counter

JNZ JUMP

HLT

1. **Multiply two numbers using successive additions**

START: LXI H,2000H ;gets first number

MOV B,M ;B has num 1

INX H ;gets second number

MOV D,M ;D has num 2

XRA A ;A=0

MVI C,00H ;This is to store carry

LOOP: ADD B

JNC SKIP

INR C ;if carry increment c

SKIP: DCR D

JNZ LOOP

LXI H,2050H

MOV M,C ;Store carry

INX H

MOV M,A ;store num

HLT

1. **Program to calculate square root of a number**

START: NOP

MVI D,01

MVI E,01

LDA 2050H

LABEL: SUB D

JZ JUMP

INR D

INR D

INR E

JMP LABEL

JUMP: MOV A,E

STA 2000H

HLT

1. **Program to calculate factorial using recursion**

START: LXI H,2000H

MOV B,M

MVI D,01H

LOOP: CALL LABEL

DCR B

JNZ LOOP

INX H

MOV M,D

HLT

LABEL: MOV E,B

MVI A,00H

LABEL2: ADD D

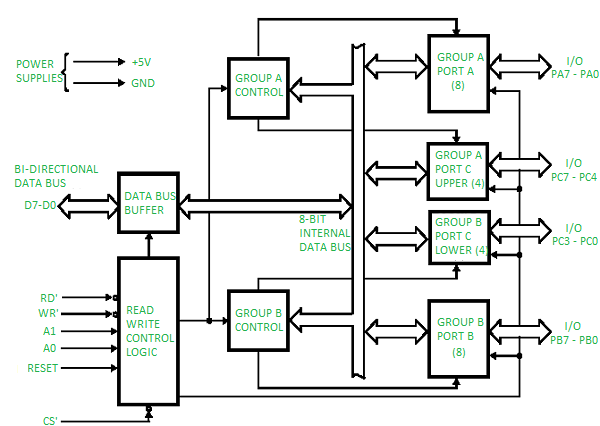
DCR E

JNZ LABEL2

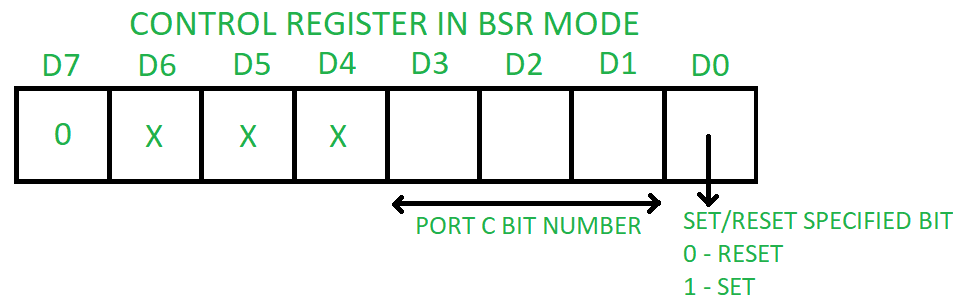
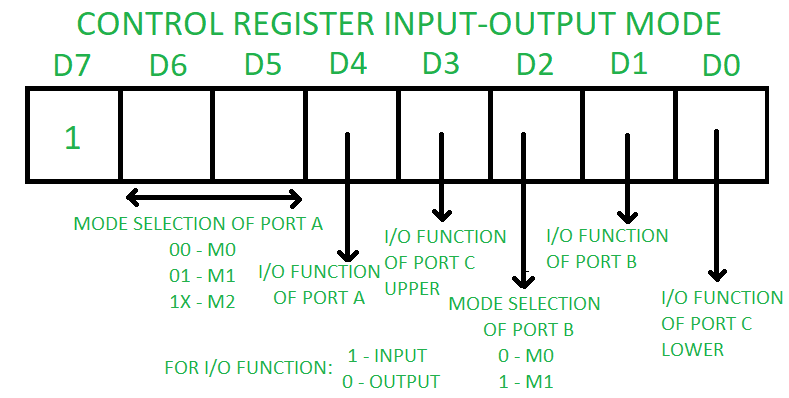
MOV D,A

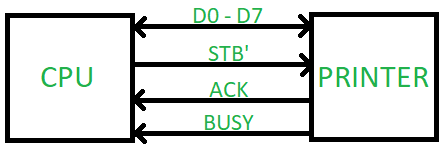
RET

1. **Interface 8255 with a microprocessor and use all its modes.**

****

**Operating modes –**

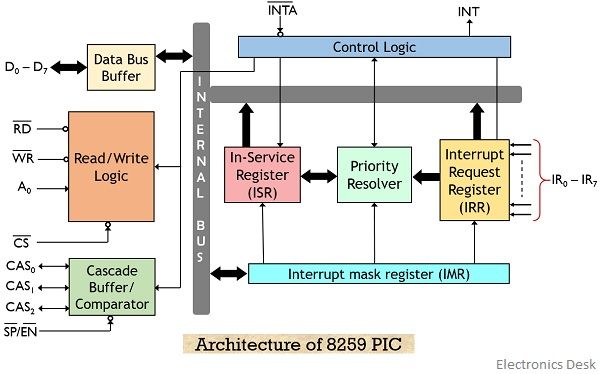
1. **Bit set reset (BSR) mode –**   
   If MSB of control word (D7) is 0, PPI works in BSR mode. In this mode only port C bits are used for set or reset.
2. **Input-Output mode –**   
   If MSB of control word (D7) is 1, PPI works in input-output mode. This is further divided into three modes:
   * **Mode 0 –**In this mode all the three ports (port A, B, C) can work as simple input function or simple output function. In this mode there is no interrupt handling capacity.
   * **Mode 1 –** Handshake I/O mode or strobbed I/O mode. In this mode either port A or port B can work as simple input port or simple output port, and port C bits are used for handshake signals before actual data transmission. It has interrupt handling capacity and input and output are latched.

Example: A CPU wants to transfer data to a printer. In this case since speed of processor is very fast as compared to relatively slow printer, so before actual data transfer it will send handshake signals to the printer for synchronization of the speed of the CPU and the peripherals.

* + **Mode 2 –** Bi-directional data bus mode. In this mode only port A works, and port B can work either in mode 0 or mode 1. 6 bits port C is used as handshake signals. It also has interrupt handling capacity.

| CS’ | A1 | A0 | Selection | Address |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | PORT A | 80 H |
| 0 | 0 | 1 | PORT B | 81 H |
| 0 | 1 | 0 | PORT C | 82 H |
| 0 | 1 | 1 | Control Register | 83 H |
| 1 | X | X | No Selection | X |
|  |  |  |  |  |

1. **Interface 8259 with a microprocessor and use all its features.**

****

**Features of 8259:**

1. The 8259 programmable interrupt controller has 8 interrupt pins thus can handle 8 interrupt inputs.
2. The priority of interrupts in 8259 can be programmed. The priority of interrupts is decided by the different operating modes.
3. a single 8259 can handle 8 interrupt inputs but by cascading multiple 8259, it can handle maximal 64 interrupt inputs.
4. 8259 allows individual masking of each generated interrupt using interrupt mask register.
5. 8259 is programmed in a way that it can handle either edge-triggered or level-triggered interrupt request at a time.
6. If multiple interrupts are generated, then 8259 holds the status of interrupts that are masked, in-service and pending.
7. It reduces the software and real-time overhead generated due to handling multilevel priority interrupts.