Date 23/04/22
Page No.

Unit: 2

Interrupts: ★ Interrupts are generated by peripherals, to indicate that the external device is ready for communication and request, the attention of microprocessor.

★ In 8085, there are five pins for interrupts:

→ TRAP

→ RST 7.5

→ RST 6.5

→ RST 5.5

→ INTR.

★ The interrupt process, is controlled by interrupt enable flip flop which is internal to the microprocessor and can be set or reset, using instructions.

★ The instruction EI (Enable Interrupt) is a 1 byte instruction that sets the interrupt enable flip flops and enables the interrupt process.

★ The instruction DI (Disable Interrupt) is a 1 byte instruction that resets the interrupt enabled flip flop and disables the interrupt.

Date / /
Page No.

RST Instructions: The 8085, includes eight RST instructions (Restart instructions) listed as RST 0 - RST 7, these are 1 byte call instructions, that transfer the program execution, to a specific location, denoted by 16 bit address.

Mnemonics	Binary code								Hex code	Call location
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
RST 0	1	1	0	0	0	1	1	1	C7	0000 H
RST 1	1	1	0	0	1	1	1	1	CF	0008H
RST 2	1	1	0	1	0	1	1	1	D7	0010H
RST 3	1	1	0	1	1	1	1	1	DF	0018H
RST 4	1	1	1	0	0	1	1	1	E7	0020H
RST 5	1	1	1	0	1	1	1	1	EF	0028 H
RST 6	1	1	1	1	0	1	1	1	F7	0030H
RST 7	1	1	1	1	1	1	1	1	FF	0038H

Vectored Interrupt: TRAP, RST 7.5, RST 6.5, RST 5.5 are automatically transferred to or vectored to a specific memory location without any external hardware.

These do not require the $\overline{INTA}$ signal, or an Input code. the necessary hardware is already implemented in the microprocessor

Interrupt	Call Location
TRAP	0024 H
RST 7.5	003C H
RST 6.5	0034 H
RST 5.5	002C H

Priority I/p bit Diagram (From pg no. 386 in book)

Date 24/04/24
Page No.

i) Set Interrupt Mask (SIM): This is a 1 byte instruction, i.e. used for 3 different functions.

(FIRST FUNCTION)

★ To set mask for RST 7.5, 6.5 & 5.5:

★ This instruction reads the content of accumulator and enables and disables the interrupts. But D_3 is a control bit, it should be 1 for D_0, D_1 and D_2 to be effective.

★ If D_3 is 0, then D_0, D_1 and D_2 will enable the Interrupt and logic 1 will disable.

(SECOND FUNCTION)

★ The second function is to reset RST 7.5, D_4 is the additional control bit for RST 7.5, If D_4 is 1, RST 7.5, is reset. This is used to override or ignore RST 7.5, without servicing it.

(THIRD FUNCTION)

★ To implement, serial I/o transmission with D_7 and D_6 of the accumulator, are used for this purpose and do not affect the interrupts. If bit D_6 is 1, then bit D_7 is used to transmit serial output.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SOD	SDE	X	R 7.5	MSE	M7.5	M6.5	M5.5

R 7.5 → Reset: If 1, then RST 7.5 is set, else reset

MSE → Mask Set Enable: 0, bit 0-2 ignored; 1, mask is set

M7.5, M6.5 → RST 7.5, RST 6.5

M5.5 → RST 5.5: 0 = available, 1 = masked

RIM {Read Interrupt Mask}: This is a 1 byte instruction that can be used for following functions.

Function 1: To read interrupt mask, this instruction loads the accumulator with 8 bits, indicating the current status of interrupts.

Function 2: To identify pending interrupts, bits D_4, D_5, and D_6 are used for this purpose.

Function 3: To receive serial data, bit D_7 is used for this purpose.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5
↓				↓			
Serial I/P data unit	Pending Interrupts 1 = Pending			Interrupt Enable flag 1: "Enabled"	Interrupt Masked 1 = Masked		

(Data Transfer Set)

1) MVI R, 8 bit
2) MOV Rd, Rs
3) LXI Rp, 16 bit
4) OUT 8 bit
5) IN 8 bit
6) STA 16 bit
7) LDA 16 bit
8) LDAx Rp
9) STAx Rp
10) MOV R, M
11) MOV M, R

Arithmetic Instructions:

1) ADD R
2) ADI 8 bit
3) ADD M
4) SUB R
5) SUI 8 bit
6) SUB M
7) INR M
8) INR R
9) DCP R
10) DCR M
11) INX Rp
12) DCX Rp
13) DAD B
14) DAD D

Logical Instructions:

1) ANA R
2) ANI 8 bit
3) ORA R
4) ORA M
5) ORI 8 bit
6) XRA R
7) XRA M
8) XRI 8 bit
9) CMP R
10) CPI 8 bit
11) RLC
12) RRC
13) CMA
14) RAR
15) RAL

(11, 12: with Carry; 14, 15: without Carry)

Branching Instructions:

JMP (16 bit address)

1) JZ 16 bit address (Jump when Zero)
2) JNZ 16 bit address (Jump Not Zero)
3) JC 16 bit address (Jump if Carry)
4) JNC " (Jump if not carry)
5) JPE " (Jump if Parity Even)
6) JPO " (Jump if Parity Odd)
7) CALL " (Call)
8) RET 16 bit address (Return)
★ HLT (Halt)
★ NOP (No operation) } Machine Control Instructions

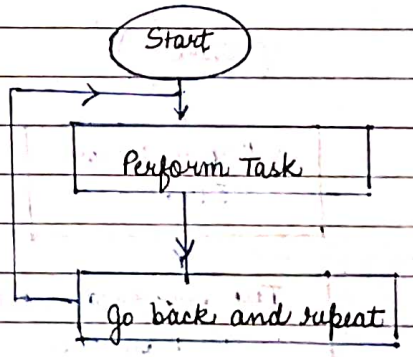
Looping: This programming technique is used to instruct the microprocessor, to repeat task, a loop is setup by using Jump instructions and changing the sequence of Execution, and performing the tasks ahead. Counting and Indexing are used to set up a loop, it can be classified into 2 types:

1) Continuous looping
2) Conditional Looping

1) Continuous Looping: This loop is set up by using the unconditional jump instruction, as shown in the flowchart.

A continuous loop doesn't stop repeating until the system is reset.

Eg: A continuous counter or a continuous monitoring system.



2) Conditional Looping: Conditional loop is set up by conditional loop jump instructions. These instructions checks flags (0, carry etc.) and repeat the specified task, if the conditions are satisfied. These loops include counting & indexing.

★ The counter is set up by loading an appropriate count in a register:

Date / /
Page No.

* Counting is performed either by incrementing or decrementing the counter.
* Loop is setup by conditional jump instruction
* End of counting is indicated by a flag.