Roll No. ☐☐☐☐☐☐☐☐☐☐☐☐ **Total No. of Pages : 02**

Total No. of Questions : 18

B.Tech. (Electronics & Communication Engineering) (2018 Batch) (Sem.–4)

MICROPROCESSORS AND MICROCONTROLLERS

Subject Code : BTEC-402-18

M.Code : 77566

Time : 3 Hrs. **Max. Marks : 60**

INSTRUCTIONS TO CANDIDATES :

1. **SECTION-A is COMPULSORY consisting of TEN questions carrying TWO marks each.**
2. **SECTION-B contains FIVE questions carrying FIVE marks each and students have to attempt any FOUR questions.**
3. **SECTION-C contains THREE questions carrying TEN marks each and students have to attempt any TWO questions.**

SECTION-A

Write briefly :

1. What is the need of interrupts in microprocessor?
2. How microcontroller is different from microprocessor?
3. Differentiate RISC and CISC architectures.
4. Give memory organization of 8051 microcontrollers.
5. Give logical instructions with examples.
6. Give significance of program counter.
7. What is bit addressability?
8. For a time delay of 25ms, what value do you need to load into the timer registers? (Assume XTAL = 11.0592 MHz)
9. Discuss the different flags of 8085 microprocessor.
10. Discuss the steps of SUBB instruction with the help of example.

SECTION-B

11. Write a program to generate 2 KHz square wave on pin P1.0 of 8051 by using timer 1 in mode 1. Assume XTAL = 20 MHz.

12. Discuss RAM organization for 8051.

13. Describe PSW and TMOD registers of 8051 microcontrollers.

14. What is addressing mode? Explain the different addressing modes with suitable examples for 8085 microprocessor.

15. Draw and explain the 8085 architecture.

SECTION-C

16. Draw and explain the interfacing of LCD with microcontroller.

17. Discuss various sources of interrupt in 8051. Also discuss various SFR's associated with interrupts.

18. Draw and explain pin configuration of 8085 microprocessor.

NOTE : Disclosure of Identity by writing Mobile No. or Making of passing request on any page of Answer Sheet will lead to UMC against the Student.

MPMC PAPER 1 SOLUTION

SECTION A

1. Interrupts are the signals generated by the external devices to request the microprocessor to perform a task. There are 5 interrupt signals, i.e. TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR. Vector interrupt − In this type of interrupt, the interrupt address is known to the processor. For example: RST7.

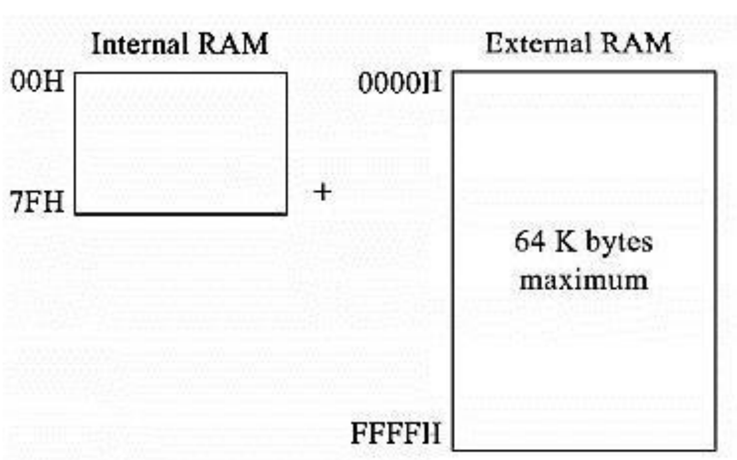
2.

- **Microprocessor consists of only a Central Processing Unit, whereas Micro Controller contains a CPU, Memory, I/O all integrated into one chip.**

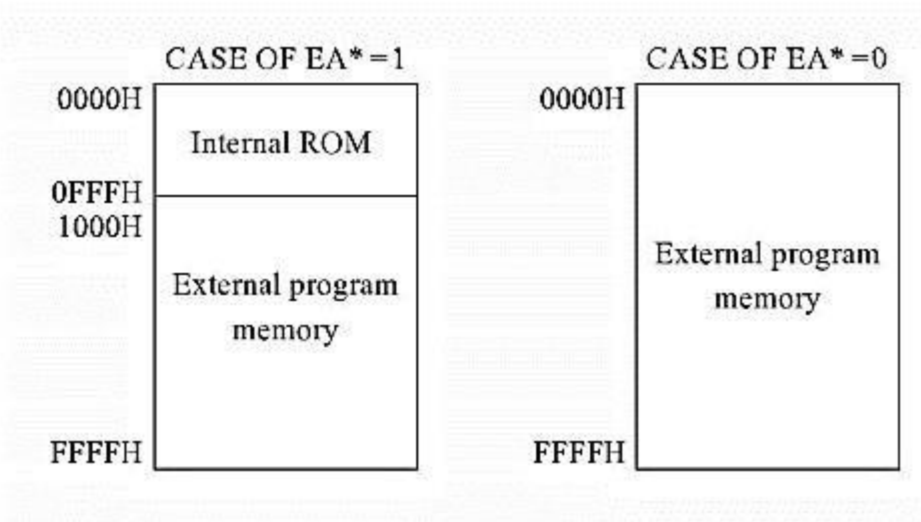
- **Microprocessor is used in Personal Computers whereas Micro Controller is used in an embedded system.**
- **Microprocessor uses an external bus to interface to RAM, ROM, and other peripherals, on the other hand, Microcontroller uses an internal controlling bus.**
- **Microprocessors are based on Von Neumann model Micro controllers are based on Harvard architecture**
- **Microprocessor is complicated and expensive, with a large number of instructions to process but Microcontroller is inexpensive and straightforward with fewer instructions to process.**

3. One of the major differences between RISC and CISC is that RISC emphasizes efficiency in cycles per instruction and CISC emphasizes efficiency in instructions per program. ... RISC needs more RAM, whereas CISC has an emphasis on smaller code size and uses less RAM overall than RISC

4. The 8051 has 128 bytes of On-Chip RAM. So for accessing that RAM area, the address space is 00H to 7FH. When we need more data memory, we can use external RAM. The address space of external RAM is 0000H to FFFFH.



We have already discussed that the 8051 microcontroller has the internal program memory. In this chip, there is EA pin. It indicates the External Access. So by using this pin, we can check whether the internal program memory is used or not. In this case, the internal memory is accessed by the address 0000H to 0FFFH. And also the external memory is accessed from location 1000H to FFFFH.



5.

Sr.No	**Instruction & Description**
1	**ANL A, R5** **This is an example of type ANL A, Rn. In this instruction, the content of R5 will be ANDed with register A and store the result into A. Similarly the OR (ORL A, Rn) and Ex-OR(XRL A, Rn) also works.**
2	**CLR A** **It is clear instruction. Using this instruction the content of register A will be 00H. We cannot clear other registers content by using this instruction.**
3	**CPL A** **This instruction is used to complement each bit of register A. This instruction also cannot complement other registers data.**
4	**RL A** **This instruction is used for rotating the A register to the Left. This is equivalent to the RLC instruction of 8085. No flags are affected by this instruction.**

Sr.No	Instruction & Description
5	**RLC A** **This is similar to the RL A instruction, but it rotates through the Carrying flag. The RAL instruction of 8085 is similar to this instruction. By this instruction only carry flag is affected.**
6	**RR A** **This instruction can perform the reverse operation of RL A instruction. It rotates the A register content to the right. In 8085, the equivalent instruction was RRC.**
7	**RRC A** **This is Rotate Right through carrying. It affects the Carry Flag only. The equivalent instruction of this in 8085 was RAR.**
8	**SWAP A** **This instruction is used to swap LS Hex digit and MS Hex digits. This instruction is functionally same as executing RL A/RR A four times**

6. A program counter is a register in a computer processor that contains the address (location) of the instruction being executed at the current time. As each instruction gets fetched, the program counter increases its stored value by 1.

7. Bit-addressable objects are objects that may be addressed as words or as bits. Only data objects that occupy the bit-addressable area of the 8051 internal memory fall into this category.Of the 128-byte internal RAM of the 8051, only 16 bytes are bit-addressable. The rest must be accessed in byte format. The bit-addressable RAM locations are 20H to 2FH. These 16 bytes provide 128 bits of RAM bit-addressability since 16×8= 128.

8.

9. In 8085 microprocessor, the flags register can have a total of eight flags. Thus a flag can be represented by 1 bit of information. But only five flags are implemented in 8085. And they are:

- **Carry flag (Cy),**
- **Auxiliary carry flag (AC),**
- **Sign flag (S),**
- **Parity flag (P), and**
- **Zero flag (Z).**

7	6	5	4	3	2	1	0 ← bit number
S	Z	x	AC	x	P	x	Cy

10. In 8085 Instruction set, SBB R is mnemonic used for multi-Byte subtraction. Let us consider the following example on such subtraction

```
                                         2'compl.
  -1                   -1                --------> 1111 1111
 44H 62H---> 0100 0100  0110 0010--------> 0100 0100  0110 0010
                                         2'compl.
-13H F1H--->-0001 0011 -1111 0001-------->+1110 1101 +0000 1111
 -------     ---------  ---------          ---------  ---------
 30H 71H---> 0011 0000  1111 0001          0011 0000  0111 0001(30H 71H)
```

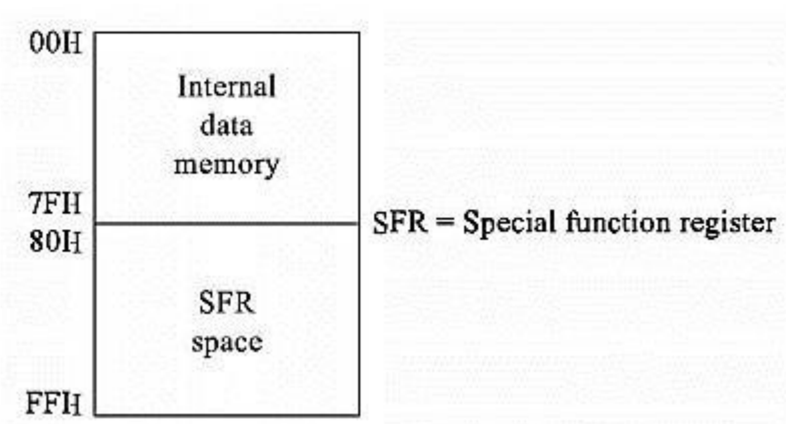
In this above example, the subtraction of 62H and F1H will result in 71H with a borrow of 1. Next, we have to subtract 44H and 13H along with this borrow value of 1. In the above tracing we have shown you how the internal calculations are being done. Now in 8085, to facilitate such an operation, SBB instruction has been provided to subtract two numbers along with the borrow value.

SECTION B

11.

12. The internal data memory of 8051 is divided into two groups. These are a set of eight registers and a scratch pad memory. These eight registers are R0 toR7. The address range 00H to 07H is used to access the registers, and the rest are scratch pad memory.

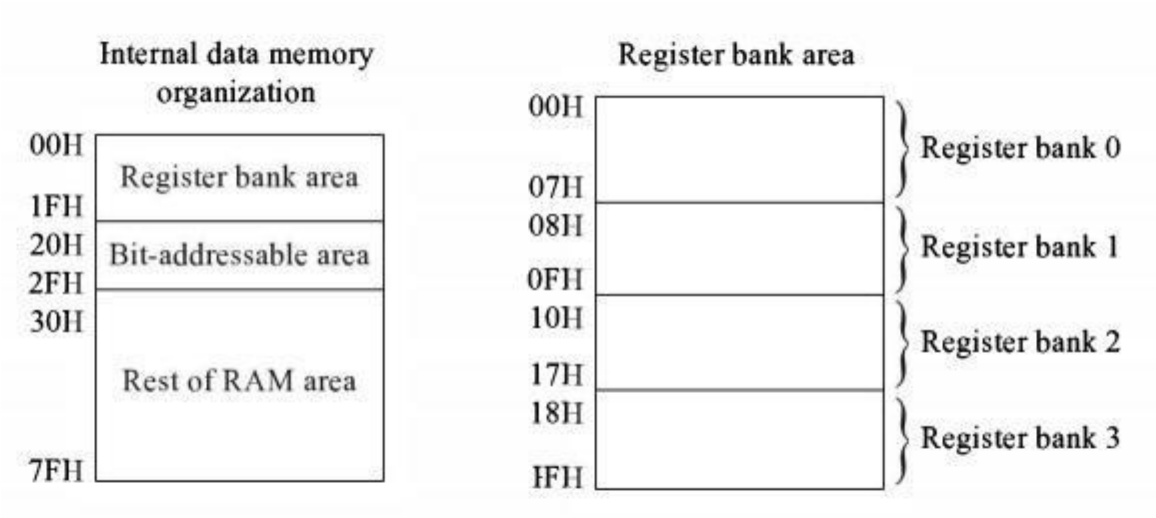
8051 Provides four register bank, but only one register bank can be used at any point in time. To select the register bank, two bits of PSW (Program Status Word) are used.



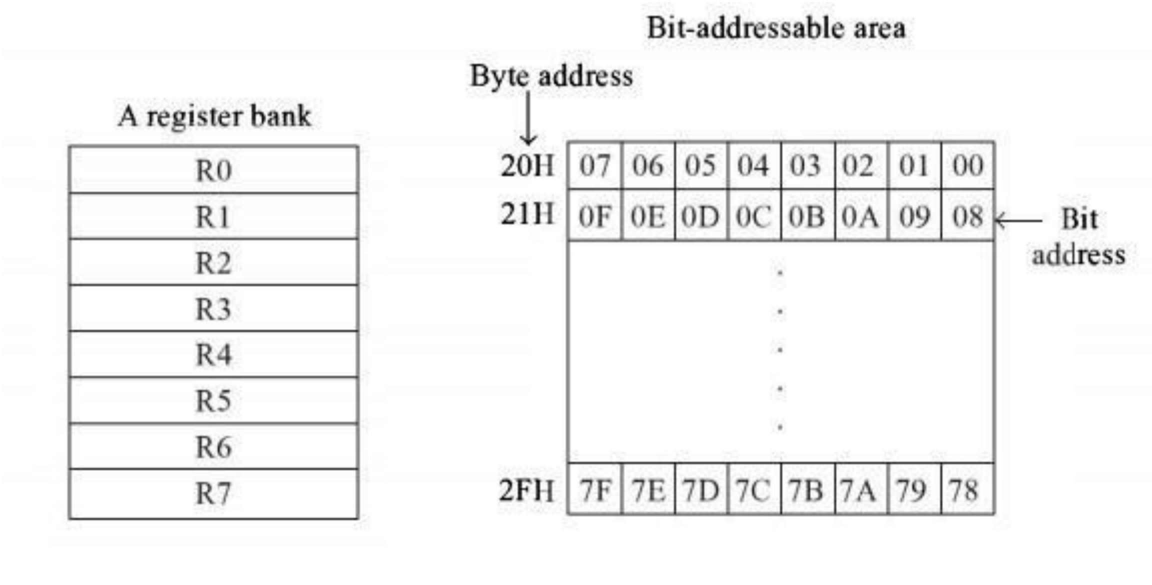
So the following addressing can be used to select register banks.

Address Range	Register Bank
00H to 07H	**Register Bank 0**
08H to 0FH	**Register Bank 1**
10H to 17H	**Register Bank 2**
18H to 1FH	**Register Bank 3**

The concept of four register banks is very useful. For servicing the interrupts, this feature is very good. The interrupt program can use one bank, and the interrupt Service Subroutine (ISS) can access another bank for better performance. As there are four banks, so for nested interrupts these can be used

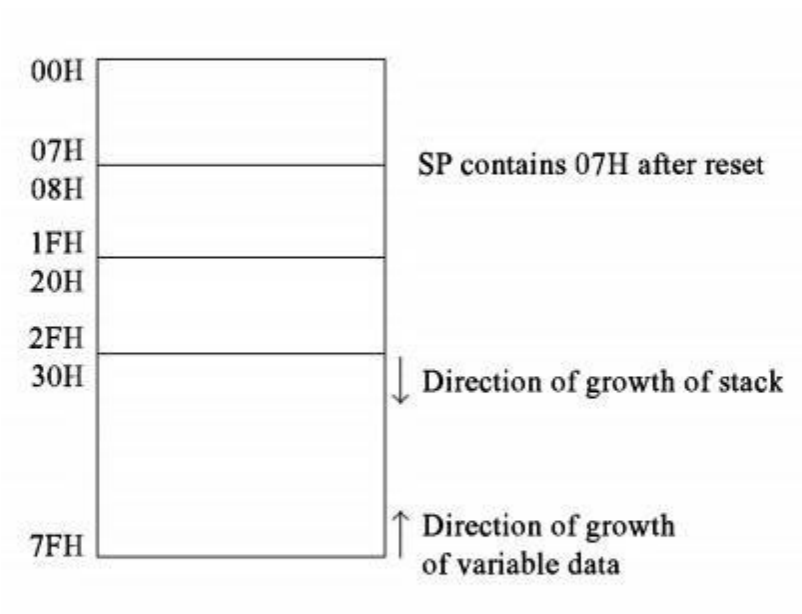


When all of the register banks are being used, the scratch pad area will be 20H to 7FH. But from 20H to 2FH (16 bytes or 128 bits) can be used as bit addressable RAM. By using some simple instructions with 8-bit memory address we can check the bit addressing. For an example the instruction CLR 6FH, using this instruction it clears the location 6FH. As we know the8-bit address can locate 256 different locations, but here only128-bits are addressable. Another section of bit addressable locations is 80H to FFH. The remaining locations (30H to 7EH) of the RAM can be used to store variable data and stack.



Stack Area

The stack area in 8051always can be implemented in the internal data memory. Here the stack pointer (SP) is an only 8-bit register, because the internal RAM area is only in range 00H to 7FH, and when all register banks are being used, the stack location will be in range 30H to 7FH. So in such a case, the SP will be initialized with 2FH.



The stack pointer SP increases before each PUSH operation and decreases after each pop instruction.

When the 8051 is reset, the Stack Pointer will point to 07H. It means the location 08H to 7FHcan be used as a stack. We are assuming that the register bank 0 is in use and 20H to 27H are not like bit-addressable area.

13. The PSW or Program Status Word Register is also called as Flag Register and is one of the important SFRs. The PSW Register consists of Flag Bits, which help the programmer in checking the condition of the result and also make decisions.

PSW: PROGRAM STATUS WORD. BIT ADDRESSABLE.

CY	AC	F0	RS1	RS0	OV	—	P

CY	PSW.7	Carry Flag.
AC	PSW.6	Auxiliary Carry Flag.
F0	PSW.5	Flag 0 available to the user for general purpose.
RS1	PSW.4	Register Bank selector bit 1 (SEE NOTE 1).
RS0	PSW.3	Register Bank selector bit 0 (SEE NOTE 1).
OV	PSW.2	Overflow Flag.
—	PSW.1	User definable flag.
P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of '1' bits in the accumulator.

P=1 Odd parity

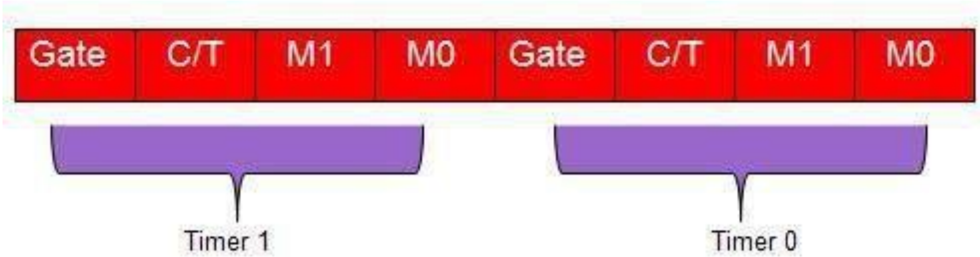
P=0 even Parity

NOTE:
1. The value presented by RS0 and RS1 selects the corresponding register bank.

RS1	RS0	Register Bank	Address
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

TMOD (Timer Mode) Register

Both Timer 0 and Timer 1 use the same register to set the various timer operation modes. It is an 8-bit register in which the lower 4 bits are set aside for Timer 0 and the upper four bits for Timers. In each case, the lower 2 bits are used to set the timer mode in advance and the upper 2 bits are used to specify the location.



Gate − When set, the timer only runs while INT(0,1) is high.

C/T − Counter/Timer select bit.

M1 − Mode bit 1.

M0 − Mode bit 0.

GATE

Every timer has a means of starting and stopping. Some timers do this by software, some by hardware, and some have both software and hardware controls. 8051 timers have both software and hardware controls. The start and stop of a timer is controlled by software using the instruction SETB TR1 and CLR TR1 for timer 1, and SETB TR0 and CLR TR0 for timer 0.

The SETB instruction is used to start it and it is stopped by the CLR instruction. These instructions start and stop the timers as long as GATE = 0 in the TMOD register. Timers can be started and stopped by an external source by making GATE = 1 in the TMOD register.

C/T (CLOCK / TIMER)

This bit in the TMOD register is used to decide whether a timer is used as a delay generator or an event manager. If C/T = 0, it is used as a timer for timer delay generation. The clock source to create the time delay is the crystal frequency of the 8051. If C/T = 0, the crystal frequency attached to the 8051 also decides the speed at which the 8051 timer ticks at a regular interval.

Timer frequency is always 1/12th of the frequency of the crystal attached to the 8051. Although various 8051 based systems have an XTAL frequency of 10 MHz to 40 MHz, we normally work with the XTAL frequency of 11.0592 MHz. It is because the baud rate for serial communication of the 8051.XTAL = 11.0592 allows the 8051 system to communicate with the PC with no errors.

M1 / M2

aM1	M2	Mode
0	0	13-bit timer mode.

0	**1**	**16-bit timer mode.**
1	**0**	**8-bit auto reload mode.**
1	**1**	**Spilt mode.**

14.

These can be guided by addressing modes. Addressing modes in 8085 can be classified into 5 groups −

- **Immediate addressing mode**
- **Register addressing mode**
- **Direct addressing mode**
- **Indirect addressing mode**
- **Implied addressing mode**

Immediate addressing mode

In this mode, the 8/16-bit data is specified in the instruction itself as one of its operands. For example MVI E, ABH means ABH is copied into register A.

- **MVI E ABH**

Register addressing mode

In this mode, the data is copied from one register to another. For example, MOV A, B: means data in register B is copied to register A.

- **MOV E, H**

Direct addressing mode

In this mode, the data is directly copied from the given address to the register. For example LDA 3000H: means the data at address 3000H is copied to register A.

- **LDA 4050H**

Let us consider LDA 4050 Has an example instruction of this type. It is a 3-Byte instruction.

Indirect addressing mode

In this mode, the data is transferred from one register to another by using the address pointed by the register. For example, MOV A, M: means data is transferred from the memory address pointed by the register pair HL to the register A.

- **MOV E, M**

It occupies only 1-Byte in memory. MOVE, M is an example instruction of this type. It is a 1-Byte instruction.

Implied addressing mode

This mode doesn't require any operand; the data is specified by the opcode itself. For example: CMA, CMP.

- **CMP E**

Let us consider one sample instruction CMPE falling in this category. It is a 1-Byte instruction so during execution of this instruction it will occupy only a single Byte in memory.

15. 8085 is an 8-bit microprocessor designed by Intel in 1977 using NMOS technology.

It has the following configuration −

- **8-bit data bus**
- **16-bit address bus, which can address upto 64KB**
- **A 16-bit program counter**
- **A 16-bit stack pointer**
- **Six 8-bit registers arranged in pairs: BC, DE, HL**
- **Requires +5V supply to operate at 3.2 MHZ single phase clock**

It is used in washing machines, microwave ovens, mobile phones, etc.

8085 Microprocessor – Functional Units

8085 consists of the following functional units −

Accumulator

It is an 8-bit register used to perform arithmetic, logical, I/O & LOAD/STORE operations. It is connected to internal data bus & ALU.

Arithmetic and logic unit

As the name suggests, it performs arithmetic and logical operations like Addition, Subtraction, AND, OR, etc. on 8-bit data.

General purpose register

There are 6 general purpose registers in 8085 processor, i.e. B, C, D, E, H & L. Each register can hold 8-bit data.

These registers can work in pair to hold 16-bit data and their pairing combination is like B-C, D-E & H-L.

Program counter

It is a 16-bit register used to store the memory address location of the next instruction to be executed. Microprocessor increments the program whenever an instruction is being executed, so that the program counter points to the memory address of the next instruction that is going to be executed.

Stack pointer

It is also a 16-bit register works like stack, which is always incremented/decremented by 2 during push & pop operations.

Temporary register

It is an 8-bit register, which holds the temporary data of arithmetic and logical operations.

Flag register

It is an 8-bit register having five 1-bit flip-flops, which holds either 0 or 1 depending upon the result stored in the accumulator.

These are the set of 5 flip-flops −

- **Sign (S)**
- **Zero (Z)**
- **Auxiliary Carry (AC)**
- **Parity (P)**
- **Carry (C)**

Its bit position is shown in the following table −

D7	D6	D5	D4	D3	D2	D1	D0
S	Z		AC		P		CY

Instruction register and decoder

It is an 8-bit register. When an instruction is fetched from memory then it is stored in the Instruction register. Instruction decoder decodes the information present in the Instruction register.

Timing and control unit

It provides timing and control signal to the microprocessor to perform operations. Following are the timing and control signals, which control external and internal circuits −

- **Control Signals: READY, RD’, WR’, ALE**
- **Status Signals: S0, S1, IO/M’**
- **DMA Signals: HOLD, HLDA**

- **RESET Signals: RESET IN, RESET OUT**

Interrupt control

As the name suggests it controls the interrupts during a process. When a microprocessor is executing a main program and whenever an interrupt occurs, the microprocessor shifts the control from the main program to process the incoming request. After the request is completed, the control goes back to the main program.

There are 5 interrupt signals in 8085 microprocessor: INTR, RST 7.5, RST 6.5, RST 5.5, TRAP.

Serial Input/output control

It controls the serial data communication by using these two instructions: SID (Serial input data) and SOD (Serial output data).

Address buffer and address-data buffer

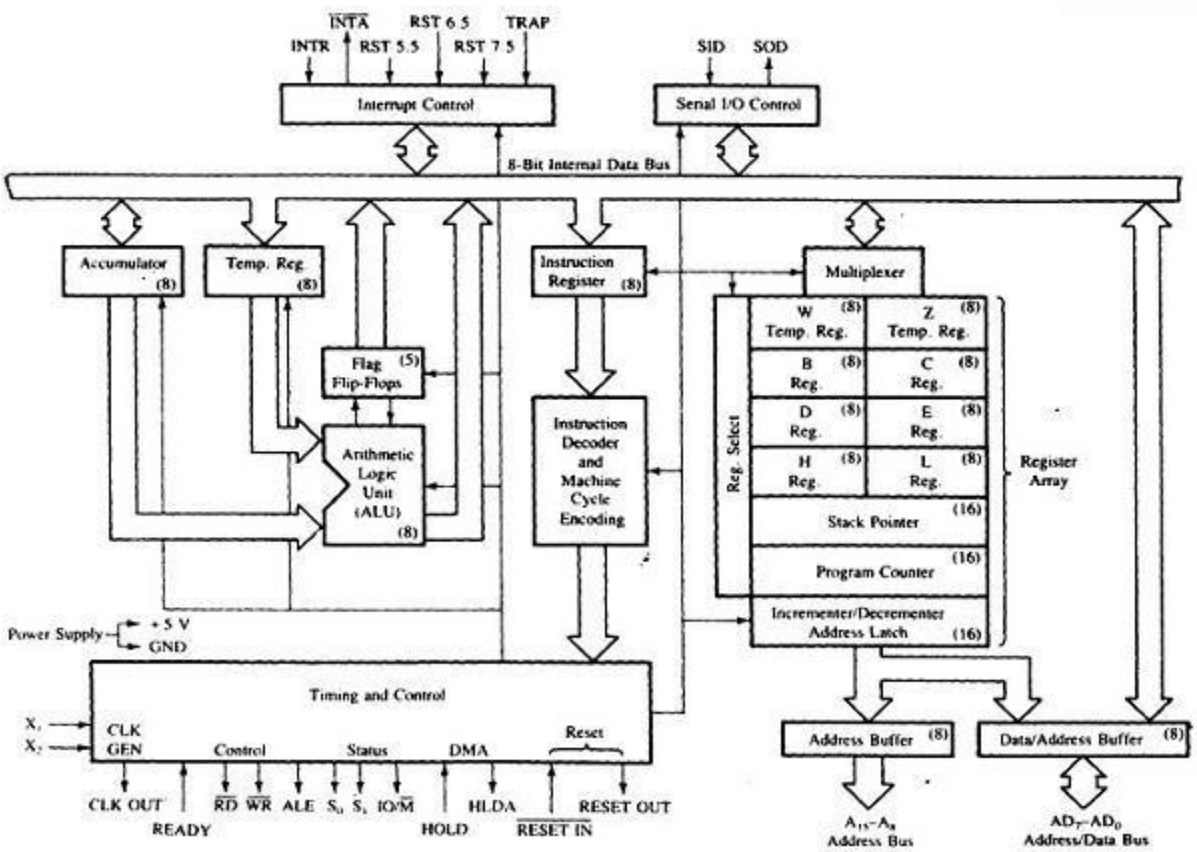
The content stored in the stack pointer and program counter is loaded into the address buffer and address-data buffer to communicate with the CPU. The memory and I/O chips are connected to these buses; the CPU can exchange the desired data with the memory and I/O chips.

Address bus and data bus

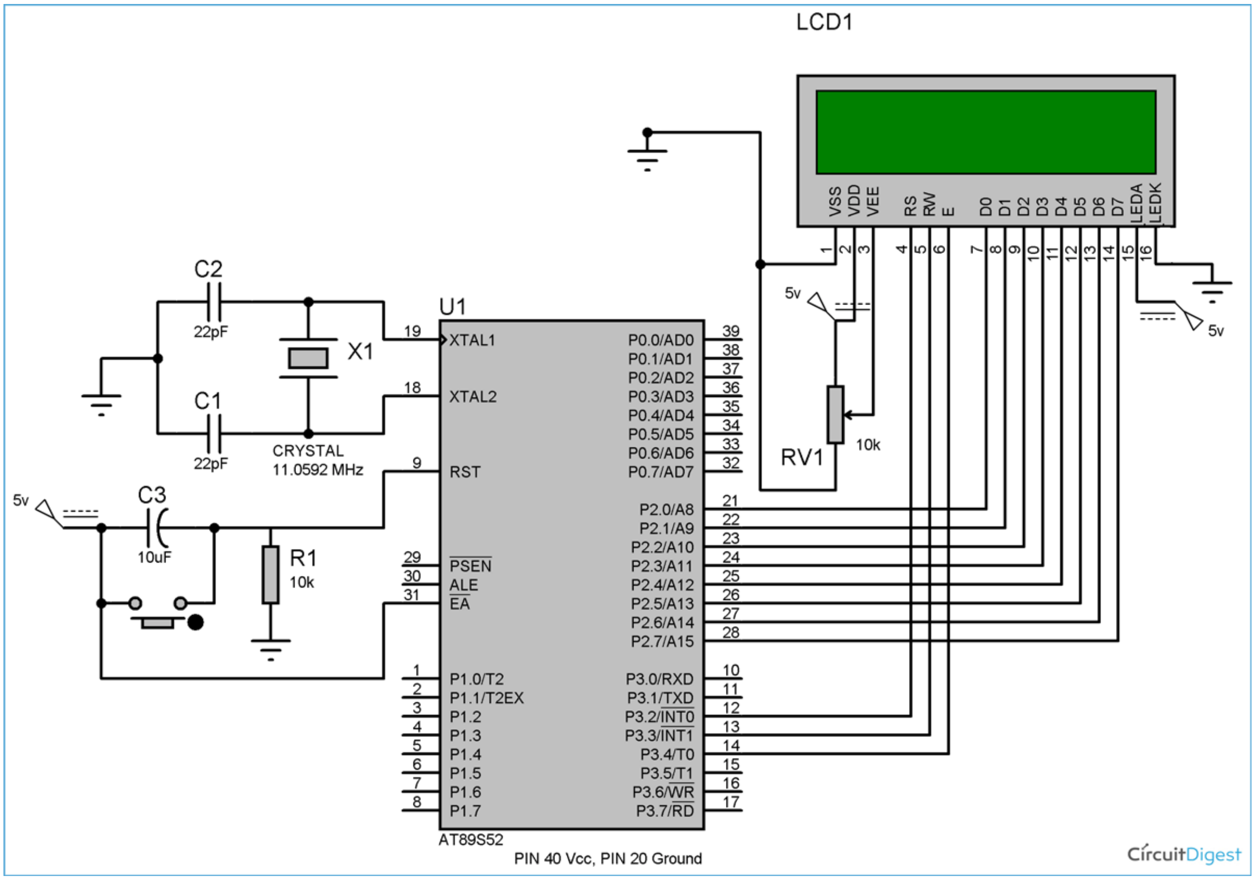
Data bus carries the data to be stored. It is bidirectional, whereas address bus carries the location to where it should be stored and it is unidirectional. It is used to transfer the data & Address I/O devices.

8085 Architecture

We have tried to depict the architecture of 8085 with this following image −


INTA
RST 6.5
TRAP
INTR
RST 5.5
RST 7.5
SID
SOD
Interrupt Control
Serial I/O Control
8-Bit Internal Data Bus
Accumulator (8)
Temp. Reg. (8)
Flag (5) Flip-Flops
Arithmetic Logic Unit (ALU) (8)
Instruction Register (8)
Instruction Decoder and Machine Cycle Encoding
Multiplexer
W (8) Temp. Reg.
Z (8) Temp. Reg.
B (8) Reg.
C (8) Reg.
D (8) Reg.
E (8) Reg.
H (8) Reg.
L (8) Reg.
Reg. Select
Stack Pointer (16)
Program Counter (16)
Incrementer/Decrementer Address Latch (16)
Register Array
Power Supply
+5 V
GND
Timing and Control
X1
X2
CLK GEN
Control
Status
DMA
Reset
CLK OUT
READY
RD
WR
ALE
S0
S1
IO/M
HOLD
HLDA
RESET IN
RESET OUT
Address Buffer (8)
Data/Address Buffer (8)
A15–A8 Address Bus
AD7–AD0 Address/Data Bus

Section C



We can divide it in five categories, Power Pins, contrast pin, Control Pins, Data pins and Backlight pins.

Category	**Pin NO.**	**Pin Name**	**Function**
Power Pins	**1**	**VSS**	**Ground Pin, connected to Ground**
	2	**VDD or Vcc**	**Voltage Pin +5V**
Contrast Pin	**3**	**V0 or VEE**	**Contrast Setting, connected to Vcc thorough a variable resistor.**
Control Pins	**4**	**RS**	**Register Select Pin, RS=0 Command mode, RS=1 Data mode**
	5	**RW**	**Read/ Write pin, RW=0 Write mode, RW=1 Read mode**

	6	**E**	**Enable, a high to low pulse need to enable the LCD**
Data Pins	**7-14**	**D0-D7**	**Data Pins, Stores the Data to be displayed on LCD or the command instructions**
Backlight Pins	**15**	**LED+ or A**	**To power the Backlight +5V**
	16	**LED- or K**	**Backlight Ground**

All the pins are clearly understandable by their name and functions, except the control pins, so they are explained below:

RS: RS is the register select pin. We need to set it to 1, if we are sending some data to be displayed on LCD. And we will set it to 0 if we are sending some command instruction like clear the screen (hex code 01).

RW: This is Read/write pin, we will set it to 0, if we are going to write some data on LCD. And set it to 1, if we are reading from LCD module. Generally this is set to 0, because we do not have need to read data from LCD. Only one instruction "Get LCD status", need to be read some times.

E: This pin is used to enable the module when a high to low pulse is given to it. A pulse of 450 ns should be given. That transition from HIGH to LOW makes the module ENABLE.

There are some preset command instructions in LCD, we have used them in our program below to prepare the LCD (in lcd_init() function). Some important command instructions are given below:

Hex Code	**Command to LCD Instruction Register**
0F	**LCD ON, cursor ON**
01	**Clear display screen**
02	**Return home**
04	**Decrement cursor (shift cursor to left)**
06	**Increment cursor (shift cursor to right)**
05	**Shift display right**
07	**Shift display left**
0E	**Display ON, cursor blinking**

80	**Force cursor to beginning of first line**
C0	**Force cursor to beginning of second line**
38	**2 lines and 5×7 matrix**
83	**Cursor line 1 position 3**
3C	**Activate second line**
08	**Display OFF, cursor OFF**
C1	**Jump to second line, position 1**
OC	**Display ON, cursor OFF**
C1	**Jump to second line, position 1**
C2	**Jump to second line, position 2**

data pins (D0-D7) of LCD to the Port 2 (P2_0 – P2_7) microcontroller. And control pins RS, RW and E to the pin 12,13,14 (pin 2,3,4 of port 3) of microcontroller respectively.

PIN 2(VDD) and PIN 15(Backlight supply) of LCD are connected to voltage (5v), and PIN 1 (VSS) and PIN 16(Backlight ground) are connected to ground.

Pin 3(V0) is connected to voltage (Vcc) through a variable resistor of 10k to adjust the contrast of LCD. Middle leg of the variable resistor is connected to PIN 3 and other two legs are connected to voltage supply and Ground.

Program for LCD Interfacing

```
#include<reg51.h>
#define display_port P2      //Data pins connected to port 2 on microcontroller
sbit rs = P3^2;  //RS pin connected to pin 2 of port 3
sbit rw = P3^3;  // RW pin connected to pin 3 of port 3
sbit e =  P3^4;  //E pin connected to pin 4 of port 3

void msdelay(unsigned int time)  // Function for creating delay in milliseconds.
{
   unsigned i,j ;
   for(i=0;i<time;i++)
   for(j=0;j<1275;j++);
}
void lcd_cmd(unsigned char command)  //Function to send command instruction to LCD
{
   display_port = command;
```

```
   rs= 0;
   rw=0;
   e=1;
   msdelay(1);
   e=0;
}

void lcd_data(unsigned char disp_data)  //Function to send display data to LCD
{
   display_port = disp_data;
   rs= 1;
   rw=0;
   e=1;
   msdelay(1);
   e=0;
}

 void lcd_init()    //Function to prepare the LCD  and get it ready
{
   lcd_cmd(0x38);  // for using 2 lines and 5X7 matrix of LCD
   msdelay(10);
   lcd_cmd(0x0F);  // turn display ON, cursor blinking
   msdelay(10);
   lcd_cmd(0x01);  //clear screen
   msdelay(10);
   lcd_cmd(0x81);  // bring cursor to position 1 of line 1
   msdelay(10);
}
void main()
{
   unsigned char a[15]="CIRCUIT DIGEST";    //string of 14 characters with a null
terminator.
   int l=0;
   lcd_init();
   while(a[l] != '\0') // searching the null terminator in the sentence
   {
      lcd_data(a[l]);
      l++;
      msdelay(50);
   }
}
```

17.

Interupts are basically the events that temporarily suspend the main program, pass the control to the external sources and execute their task. It then passes the control to the main program where it had left off.

8051 has five interrupts. These interrupts are INT0, INT1,TO ,T1 , TI/RI. All of the interrupts can be enabled or disabled by using the IE (interrupt enable) register.

The interrupt addresses of these interrupts are like below −

Interrupt	Address
INT0	0003H
INT1	000BH
T0	0013H
T1	001BH
TI/RI	0023H

Interrupt Enable (IE)Register

This register can be used to enable or disable interrupts programmatically. This register is an SFR. The address is A8H. This byte is bit addressable. So it can be programmed by the user. The bits in this register has a different meaning. The register structure is looking like this:

BitAddress	AF	AE	AD	AC	AB	AA	A9	A8
Bit Details	EA	X	X	ES	ET1	EX1	ET0	EX0

Now, let us see the bit details and different operations when the value is low (0) and high(1).

Bit Details	High Value(1)	Low Value(0)
EA	Least significant 5 bits can decide enable or disable of these five interrupts.	Disable all five interrupts. It just ignores the rest five bits.
ES	Enable Serial Port Interrupt	Disable Serial Port Interrupt
ET1	Enable Timer1 interrupt	Disable Timer1 interrupt
EX1	Enable external interrupt 1 (INT1)	Disable external interrupt 1 (INT1)
ET0	Enable Timer0 interrupt	Disable Timer0 interrupt
EX0	Enable external interrupt 0 (INT0)	Disable external interrupt 0 (INT0)

Interrupt Priority (IP) Register

All of these five interrupts can be in one or two interrupt level. The priority levels are level 1 and level 0. Priority level 1 indicates the higher priority, and level 0 indicates lower priority. This IP register can be used to store the priority levels for each interrupt. This is also a bit addressable SFR. Its address is B8H.

BitAddress	BF	BE	BD	BC	BB	BA	B9	B8
Bit Details	X	X	X	PS	PT1	PX1	PT0	PX0

Now, let us see the bit details and different operations when the value is low (0) and high(1).

Bit Details	High Value(1)	Low Value(0)
PS	Set 1 level priority of Serial port interrupt	Set 0 level priority of Serial port interrupt
PT1	Set 1 level priority of Timer1 interrupt	Set 0 level priority of Timer1 interrupt
PX1	Set 1 level priority of external interrupt 1 (INT1)	Set 0 level priority of external interrupt 1 (INT1)
PT0	Set 1 level priority of Timer0 interrupt	Set 0 level priority of Timer0 interrupt
PX0	Set 1 level priority of external interrupt 0 (INT0)	Set 0 level priority of external interrupt 0 (INT0)

When all of the five interrupts are in same priority level, and if all of the interrupts are enabled, then the sequence of interrupts will be INT0, T0, INT1, T1, TI/R I.

Some specific priority register value can be used to maintain the priorities of the interrupts. Let the value of Priority register is xxx00101 indicates the sequence INT0, INT1, TI/RI, T1, T0. But all of the sequences are not feasible. Like INT0, INT1, TI/RI, T1, T0 is not valid.

External Interrupt

The external interrupts of 8051 are INT0and. INT1 These interrupts can be programmed to either edge-triggered or level triggered. The TCON register can be used top rogram external interrupts to edge or level triggered. The TCON isTimer Control. TCON is another bit addressable SFR. Here the address is 88H.

BitAddress	8F	8E	8D	8C	8B	8A	89	88
Bit Details	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Now, let us see the bit details and different operations when the value is low (0) and high(1).

Bit Details	High Value(1)	Low Value(0)
IT0	Set (INT0) as negative edge triggeredinput.	Set (INT0) as active low level triggered input.
IT1	Set (INT1) as negative edge triggeredinput.	Set (INT1) as active low level triggered input.
IE0	This will be 1, when INT0is activated as level triggered.	This will be 0, when INT0is activated as edge triggered.
IE1	This will be 1, when INT1 is activated as level triggered.	This will be 0, when INT1 is activated as edge triggered.
TR0	Set Timer0 as run mode	Set Timer0 as stop mode.
TR1	Set Timer1 as run mode	Set Timer1 as stop mode.
TF0	High when Timer T0 overflow occurs.	After resetting the timer T0 thiswill also be changed to 0 state
TF1	High when Timer T1 overflow occurs.	After resetting the timer T1 this will also be changed to 0 state.

The IT0 and IT1 are stands for Interrupt Type. These bits are used to decide whether the INT0 and INT1 will be level trigged or edge triggered.

IE0 and IE1 bits are used to indicate the status of external interrupts. These bit can be set or reset by the microcontroller itself.

The first four bits are the status information about timers. When TR0 and TR1 are 1, it indicates the running mode of the timers. These bits provide software control over the running of timers. Timers can also be controlled by the hardware. The priority of hardware mode is higher than the software mode.

The TF0 and TF1 are used to indicate the overflow of timer T0 and T1 respectively. When over flow occurs these flags are set to 1. When the interrupt is handled by some interrupt service subroutine (ISS), these will be 0.

Serial Port Interrupt

The serial ports can be used either Transmitting mode or reception mode. The interrupt status for the Transmission is provided by TI, and status for Reception is provided by RI. These are two bits of SCON(Serial Control). This is also a bit addressable SFR. The address is98H

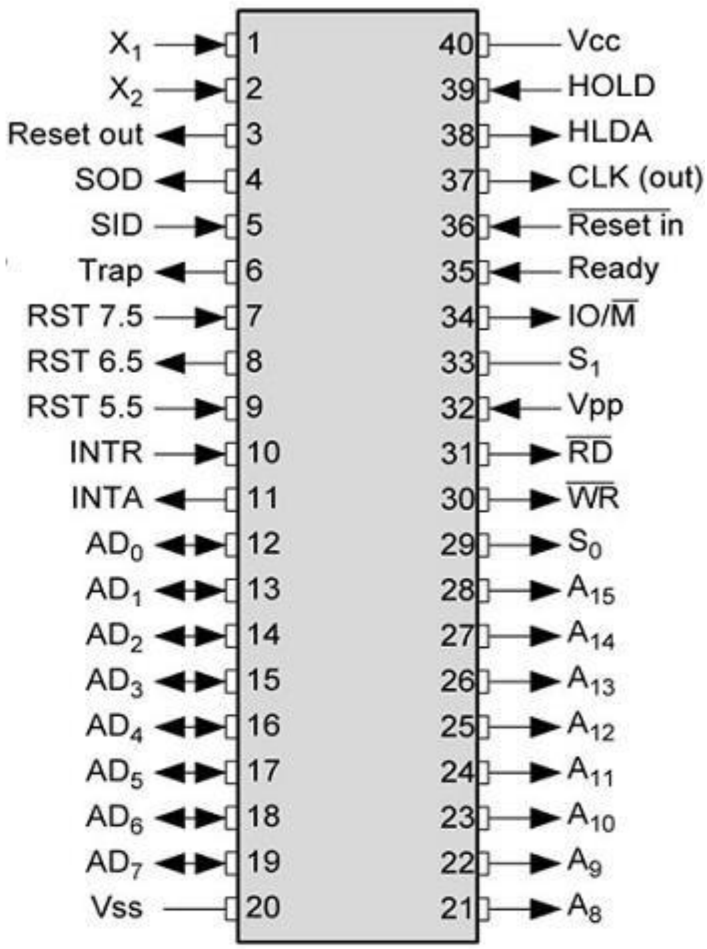
BitAddress	**9F**	**9E**	**9D**	**9C**	**9B**	**9A**	**99**	**98**
Bit Details	**SM0**	**SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TI**	**RI**

The significance of these bits are as follows

Bit Details	**Description**
SM0	**This is Serial Port Mode 0 shift register**
SM1	**This is Serial Port Mode 1 (8-bit UAR + variable)**
SM2	**Enable multiprocessor communication in the mode 2 or 3**
REN	**Set or reset by the software to enable or disable the Reception**
TB8	**It indicates the 9th bit that will be transmitted in mode 2 or 3. It can be set or reset by the software**
RB8	**In mode 2 or 3, the 9th bit was received in mode 1.**
TI	**The transmission interrupt flag. It can be set by hardware.**
RI	**The receiver interrupt flag. It can be set by hardware but must be reset by software.**

18.

The following image depicts the pin diagram of 8085 Microprocessor −



The pins of a 8085 microprocessor can be classified into seven groups −

Address bus

A15-A8, it carries the most significant 8-bits of memory/IO address.

Data bus

AD7-AD0, it carries the least significant 8-bit address and data bus.

Control and status signals

These signals are used to identify the nature of operation. There are 3 control signal and 3 status signals.

Three control signals are RD, WR & ALE.

- **RD − This signal indicates that the selected IO or memory device is to be read and is ready for accepting data available on the data bus.**
- **WR − This signal indicates that the data on the data bus is to be written into a selected memory or IO location.**

- **ALE − It is a positive going pulse generated when a new operation is started by the microprocessor. When the pulse goes high, it indicates address. When the pulse goes down it indicates data.**

Three status signals are IO/M, S0 & S1.

IO/M

This signal is used to differentiate between IO and Memory operations, i.e. when it is high indicates IO operation and when it is low then it indicates memory operation.

S1 & S0

These signals are used to identify the type of current operation.

Power supply

There are 2 power supply signals − VCC & VSS. VCC indicates +5v power supply and VSS indicates ground signal.

Clock signals

There are 3 clock signals, i.e. X1, X2, CLK OUT.

- **X1, X2 − A crystal (RC, LC N/W) is connected at these two pins and is used to set frequency of the internal clock generator. This frequency is internally divided by 2.**
- **CLK OUT − This signal is used as the system clock for devices connected with the microprocessor.**

Interrupts & externally initiated signals

Interrupts are the signals generated by external devices to request the microprocessor to perform a task. There are 5 interrupt signals, i.e. TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR. We will discuss interrupts in detail in interrupts section.

- **INTA − It is an interrupt acknowledgment signal.**
- **RESET IN − This signal is used to reset the microprocessor by setting the program counter to zero.**
- **RESET OUT − This signal is used to reset all the connected devices when the microprocessor is reset.**
- **READY − This signal indicates that the device is ready to send or receive data. If READY is low, then the CPU has to wait for READY to go high.**
- **HOLD − This signal indicates that another master is requesting the use of the address and data buses.**

- **HLDA (HOLD Acknowledge) − It indicates that the CPU has received the HOLD request and it will relinquish the bus in the next clock cycle. HLDA is set to low after the HOLD signal is removed.**

Serial I/O signals

There are 2 serial signals, i.e. SID and SOD and these signals are used for serial communication.

- **SOD (Serial output data line) − The output SOD is set/reset as specified by the SIM instruction.**
- **SID (Serial input data line) − The data on this line is loaded into accumulator whenever a RIM instruction is executed.**