



SUMMER – 13 EXAMINATION

Subject Code: **12187**

Model Answer

Important Instructions to examiners:

- 1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
- 2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
- 3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills).
- 4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
- 5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
- 6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
- 7) For programming language papers, credit may be given to any other program based on equivalent concept.

1. Attempt any five of the following:

(20 Marks)

- a. Draw the transmission format for asynchronous transmission. State four important features of parallel data transfer.

Ans a. (Asynchronous Transmission Format -- 2 Marks)

(any four features of parallel data transfer -- 2 Marks)

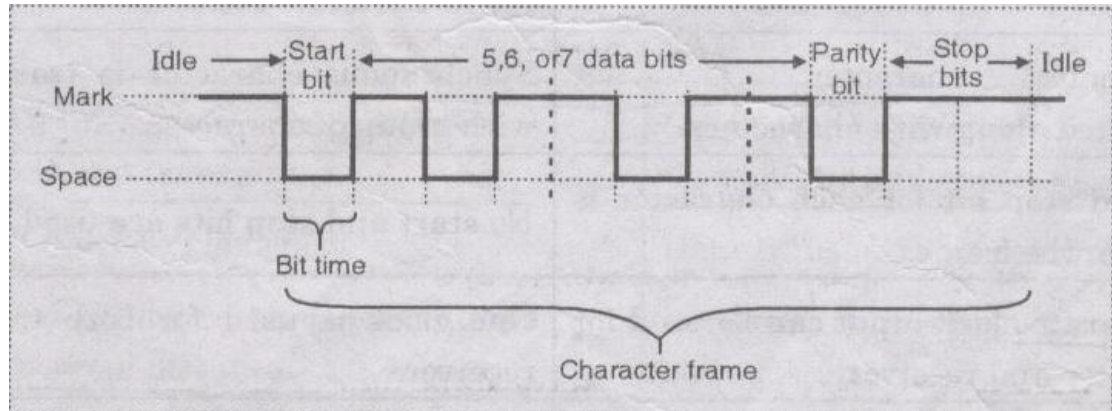


Fig: Transmission format for asynchronous transmission

Four important features of parallel transfer are as follows:-

1. All bits of data can be transferred at a time in a single clock pulse.
2. Normally the data is 8 bit and separate line is required for each bit in data byte. Hence 8 wires or lines are required to transfer 8 bit data and one additional line i.e. 9th to transfer parity bit along with data byte for error detection.
3. Data transfer rate is faster as compared to serial data transfer.
4. Used only if data transfer is required within small distance.
5. Installation cost is more.

b. Give the control word format of 8255. Find the 8255 control word to

- i) Set up all ports as an input in mode 0
- ii) Set up all ports as an output in mode 0.

Ans b. (control word format of 8255 --2 Marks)

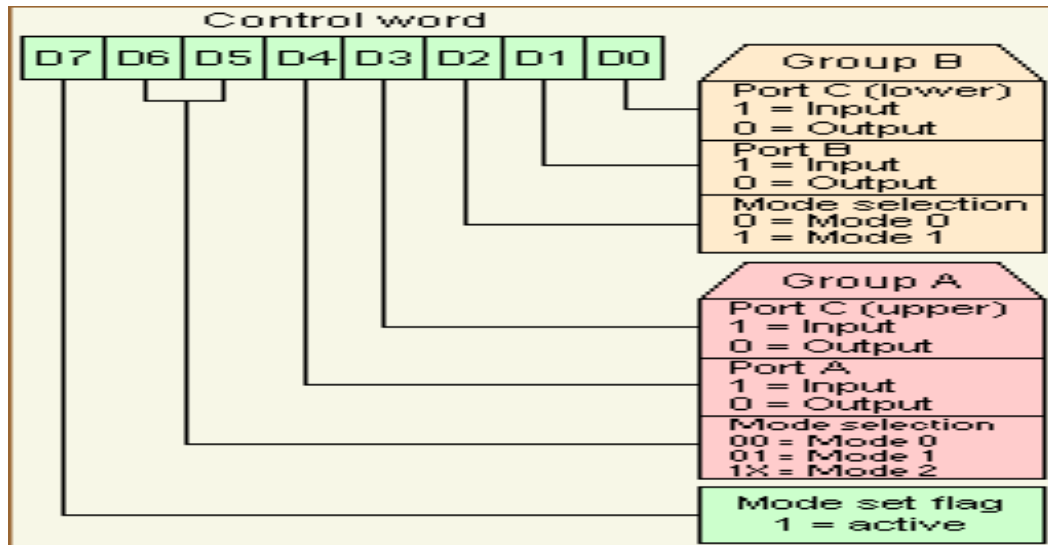
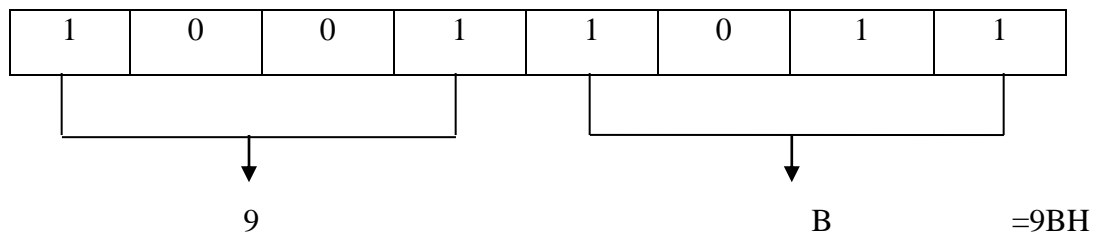


Fig. Control word format of 8255

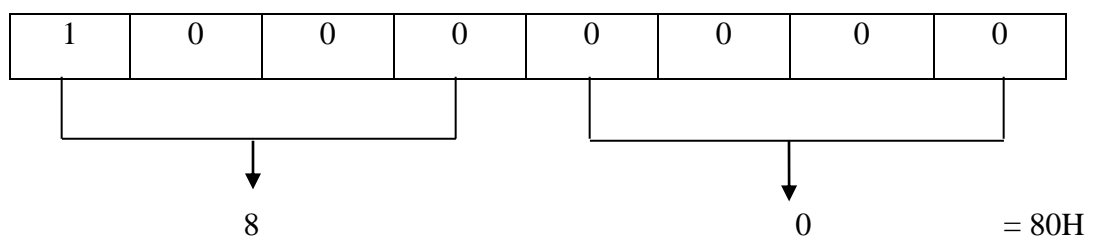
(Correct answer --1Mark)

- i) Set up all ports as an input in mode 0



(Correct answer --1Mark)

- ii) Set up all ports as an output in mode 0.



- c. State difference between Harvard and John Von Neumann's architecture with suitable diagram.

Ans c. Difference between Harvard and John Von Neumann's architecture is as follows:-

Harvard Architecture diagram -- 1 Mark

Van Neumann's Architecture diagram -- 1 Mark

Any 2 difference points --2 marks

Sr. No	Harvard Architecture	Van Neumann's Architecture
1.		
2.	The Harvard architecture uses physically separate memories for their instructions and data.	The Van Neumann's architecture uses single memory for their instructions and data.
3.	Requires separate & dedicated buses for memories for instructions and data	Requires single bus for instructions and data.
4.	Its design is complicated	Its design is simpler.
5.	Instructions and data can be fetched simultaneously as there is separate buses for instructions and data which increasing operation bandwidth.	Instructions and data have to be fetched in sequential order limiting the operation bandwidth.

- d. Explain the flag register of 8051 microcontroller, with its format.

Flag register format --2 Marks

Explanation --2 Marks

Ans d.

CY	AC	F0	RS1	RS0	OV	--	P
----	----	----	-----	-----	----	----	---

CY PSW.7 Carry Flag.

AC PSW.6 Auxiliary carry flag.

F0 PSW.5 Available to the user for general purpose.

RS1 PSW.4 Register bank selector bit 1.

RS0 PSW.3 Register bank selector bit 0.

OV PSW.2 Overflow flag.

-- PSW.1 User- definable bit.

P PSW.0 Parity flag. Set/cleared by hardware each instruction cycle to indicate and Odd/ even number of 1 bit in the accumulator.

1. CY: the carry flag.

This flag is set whenever there is a carry out from the D7 bit.

The flag bit is affected after an 8 bit addition or subtraction.

It can also be set to 1 or 0 directly by an instruction such as “SETB C” and CLR C” where “SETB C” stands for “set bit carry” and “CLR C” for “clear carry”.

2. AC: the auxiliary carry flag

If there is a carry from D3 and D4 during an ADD or SUB operation, this bit is set; it is cleared. This flag is used by instructions that perform BCD (binary coded decimal) arithmetic.

3. F0: Available to the user for general purposes.

4. RS0, RS1: register bank selects bits

These two bits are used to select one of the four register banks in internal RAM in the table. By writing zeroes and ones to these bits, a group of registers R₀- R₇ can be used out of four registers banks in internal RAM.

RS1	RS0	Space in RAM
0	0	Bank 0 (00H- 07H)
0	1	Bank 1 (08H-0FH)
1	0	Bank2 (10H-17H)
1	1	Bank3 (18H-1FH)

5.OV: the overflow flag

This flag is set whenever the result of a signed number operation is too large, causing the high- order bit to overflow into the sign bit. In general, the carry flag is used to detect errors in unsigned arithmetic operations. The overflow flag is only used to detect errors in signed arithmetic operations.

6. P: Parity flag

The parity flag reflects the number of 1s in the A (accumulator) register only. If the A register contains an odd number of 1s, then P=1.

P=0 if A has an even number of 1s.

- e. State four features of 8051 micro controller.

Ans e . **Any four features -- 4 marks**

Features of 8051 micro controller are as follows:-

- 1) 8- bit data bus and 8- bit ALU.
- 2) 16- bit address bus – 64KB of RAM and ROM.
- 3) On- chip RAM -128 (256) bytes (“ Data Memory”)
- 4) On- chip ROM – 4 KB (“Program Memory”)
- 5) Four 8-bit bi- directional input/output ports Four 8-bit bi- directional input/ output ports.
- 6) Programmable serial ports i.e. One UART (serial port)
- 7) Two 16- bit timers- Timer 0& Timer 1
- 8) Six interrupts are available: Reset, Two interrupts Timers i.e. Timer 0 and Timer 1, Two external hardware interrupts- INT0 and INT1, Serial communication interrupt for both receive and transmit

- f. Draw the format of SCON register and explain the function of each bit.

Ans f. **SCON register format** **-- 2 Marks**
Explanation of each bit **-- 2 Marks**

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

SM0	SCON.7	Serial port mode specifier
SM1	SCON.6	Serial port mode specifier.
SM2	SCON.5	Used for multiprocessor communication (Make it 0.)
REN	SCON.4	Set/ cleared by software to enable/ disable reception.
TB8	SCON.3	Not widely used.
RB8	SCON.2	Not widely used.

TI	SCON.1	Transmit interrupt flag. Set by hardware at the beginning of the stop Bit in mode 1. Must be cleared by software.
RI	SCON.0	Receive interrupt flag. Set by hardware halfway through the stop bit time in mode 1. Must be cleared by software.

Note: Make SM2, TB8 and RB8 = 0.

SM0	SM1	
0	0	Serial Mode 0
0	1	Serial Mode 1, 8-bit data, 1 stop bit, 1 start bit
1	0	Serial Mode 2
1	1	Serial Mode 3

SM2: SM2 is the D5 bit of the SCON register.

This bit enables the multiprocessing capability of the 8051. Make SM2= 0 since we are not using the 8051 in a multiprocessor environment.

REN: The REN (receive enable) bit is D4 of the SCON register. The REN bit is also referred to as SCON.4 since SCON is a bit addressable register.

When the REN =1, it allows the 8051 to receive data on the RxD pin of the 8051. As a result if we want the 8051 to both transfer and receive data, REN must be set to 1.

By making REN=0, the receiver is disabled. Making REN=1 or REN=0 can be achieved by the instructions “SETB SCON.4” and “CLR SCON.4”, respectively.

This bit can be used to block any serial data reception and is an extremely important bit in the SCON register.

TB8: TB8 (transfer bit 8) is bit D3 of SCON. It is used for serial modes 2 and 3. We make TB8=0 since it is not used in our applications.

RB8: RB8 (receive bit 8) is bit D2 of the SCON register. In serial mode 1, this bit gets copy of the stop bit when an 8 bit data is received. This bit (as is the case for TB8) is rarely used anymore. In all our applications we will make RB8=0. Like TB8, the RB8 bit is also used in serial modes 2 and 3.

TI: TI (transmit interrupt) is bit D1 of the SCON register.

This is an extremely important flag bit in the SCON register.

When the 8051 finishes the transfer of the 8 bit character, it raises the T1 flag to indicate that it is ready to transfer another byte. The TI bit is raised at the beginning of the stop bit.

RI: RI (receive interrupt) is the D0 bit of the SCON register. This is another extremely important flag in the SCON register.

When the 8051 receives data serially via RxD, it gets rid of the start and stop bits and places the byte in the SBUF register.

Then it raises the RI flag bit to indicate that a byte has been received and picked up before it is lost. RI is raised halfway through the stop bit.

g. Explain the following 8051 micro controller instructions.

i) CJNE A, direct, rel

ii) XCHD A, @ Ri

Ans g.

i) Explanation of CJNE A, direct, rel -- (2 Marks)

Function: Compare and jump if not equal

Addressing mode: Direct

Bytes: 3

Cycles: 2

Encoding:

1 0 1 1	0 1 0 1	Direct address	rel. address
---------	---------	----------------	--------------

Operation: (PC) \leftarrow (PC) + 3
IF (A) < > (direct)
THEN
 (PC) \leftarrow (PC) + relative offset
IF (A) < (direct)
THEN
 (C) \leftarrow 1
ELSE
 (C) \leftarrow 0

Example: CJNE A, 40H, NEXT ; jump if A NOT = with the value held by RAM location 40H.

ii) Explanation of XCHD A, @ Ri –2 Marks

Function: Exchange digits

Flags: None

The XCHD instructions exchanges only the lower nibble of A with the lower nibble of the RAM location pointed to by Ri while leaving the upper nibbles in both places intact.

Example: Assuming RAM location 40H has the value 97H, find its content after the following instructions

; 40H= (97H)

MOV A, #12H ; A= 12H (0001 0010 binary)

MOV R1, # 40 ; R1= 40H, load pointer

XCHD A, @R1 ; exchange the lower nibble of
; A and RAM location 40H

After execution of the XCHD instruction, we have A= 17H and RAM location 40H has 92H.

2. Attempt any four of the following:

(16 Marks)

a. Explain why peripheral is needed.

Ans a. Need of peripheral– (4 Marks)

- The microcomputer consists of different components such as microprocessor, memory and different I/O devices.
- The I/O devices are required to interface different external I/O peripherals for data transfer.
- The microcomputer can read data from external peripheral or can write data to the external peripherals.
- The external peripheral may be keyboard, switches data acquisition system, sensors, magnetic tape, hard disk drive, scanner, printer etc.
- These external peripheral are required to input data to microcomputer and the output devices are required to get the soft or hard copy of the data.
- So, Microcomputer reads data from input peripheral such as sensors, data acquisition system, keyboards, mouse, scanner, switches etc. and write data to monitor (display device), printer, storage devices, controllers and actuators or other microcomputer in network.
- While transferring data to/ from peripheral, the speed of the peripheral should be same to achieve faster data transfer rate.
- But practically, the speed of the peripheral is slow as they have some electro- mechanical components which take some time to perform desired action.
- So, the slower peripheral cannot responds immediately. Hence, there are different data transfer technique have been developed to solve this speed problem.

b. Compare between microprocessor and microcontroller based on number of instructions used, memory, register and applications.

Ans b. (1 Mark for each point)

Sr. No	Parameter	Microprocessor	Microcontroller
1.	No. of instructions used	Many instructions to read/ write data to/ from external memory.	Few instruction to read/ write data to/ from external memory
2.	Memory	Do not have inbuilt RAM or ROM.	Inbuilt RAM or ROM
		Program and data are stored in same memory.	Separate memory to store program and data

3.	Registers	Microprocessor contains general purpose registers, Stack pointer register, Program counter register	Microcontroller contains general purpose registers, Stack pointer register, Program counter register additional to that it contains Special Function Registers (SFRs) for Timer , Interrupt and serial communication etc.
4.	Applications	General purpose, Computers and Personal Uses.	Single purpose(dedicated application), Automobile companies, embedded systems, remote control devices.

c. Explain the function of $\overline{\text{PSEN}}$ and ALE pins of microcontroller 8051.

Ans c. **Function of $\overline{\text{PSEN}}$ -- (2Marks)**

- PSEN stands for “program store enable.” In an 8031-based system in which an external ROM holds the program code, this pin is connected to the OE pin of the ROM.
- In other words, to access external ROM containing program code, the 8031/51 uses the PSEN signal.
- When the EA pin is connected to GND, the 8031/51 fetches opcode from external ROM by using PSEN.
- In systems based on the 8751/89C51/ DS5000 where EA is connected to VCC, these chips do not activate the PSEN pin. This indicates that the on- chip ROM contains program code.

Function of ALE -- (2Marks)

- ALE stands for address latch enable. It is an output pin and is active high for latching the low byte of address during accesses to external memory.
- The ALE pin is used for demultiplexing the address and data by connecting to the G pin of the 74LS373 chip.

d. Explain the Boolean processor of 8051 micro controller

Ans d. **Explanation – (4Marks)**

The 8051 instruction set is optimized for the one bit operations so often desired in real – world, real- time control applications.

- The Boolean processor is an internal part of 8051 architecture. It is an independent bit processor with its own accumulator and its own bit addressable RAM and I/O.

- The Boolean processor provides direct support for bit manipulation .This leads to more efficient programs that needs to deal with binary input and output conditions inherent in digital control problems.
- Bit addressing can be used for test pin monitoring or program control flags. For example, instructions for Boolean function are as given below.
 - (a) ORL P0, #1 ; Set P0.0
 - (b) XRL P0, #1 ; Toggle P0.0
 - (c) ANL C, P1.4 ; AND the bit on P1.4 with carry.
 - (d) ANL C,! (P1.4) ; AND inverted bit on P1.4 with carry.

e. Describe the baud rate in UART OF 8051

Ans e. **Explanation – (4Marks)**

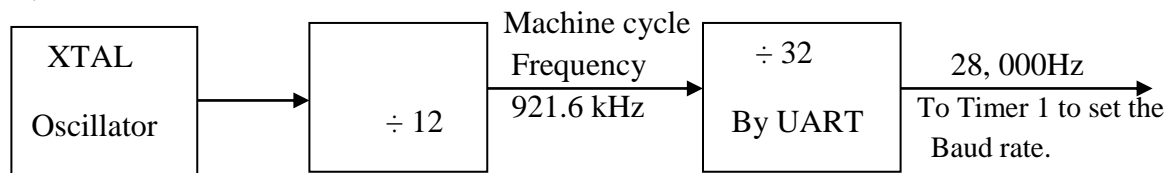
- The rate of data transfer in serial data communication is stated in bps (bits per second). Another widely used terminology for bps is baud rate.
- The 8051 divides the crystal frequency by 12 to get the machine cycle frequency. In the case of XTAL = 11.0592 MHz, the machine cycle frequency is 921.6k Hz ($11.0592 \text{ MHz}/12 = 921.6 \text{ kHz}$).
- The 8051's serial communication UART circuitry divides the machine cycle frequency of 921.6 kHz by 32 once more before it is used by Timer 1 to set the baud rate. Therefore, 921.6 kHz divided by 32 gives 28,800 Hz.
- When Timer 1 is used to set the baud rate it must be programmed in mode 2, which is 8- bit, auto- reload. To get baud rates compatible with the PC, we must load TH1 with the values shown in table.

TABLE: Timer 1 TH1 Register values for Various Baud Rates

Baud rate	TH1 (Decimal)	Th1 (Hex)
9600	-3	FD
4800	-6	FA
2400	-12	F4
1200	-24	E8

Note: XTAL= 11.0592 MHz

11, 0592 MHz



Ans f. Correct block- schematic of temperature control using 8085 μ P—(4Marks)



3. Attempt any four of the following:

(16 Marks)

a) Explain the timer section of 8155.

Ans a. **Explanation -- (4 marks)**

The Timer is a 14 bit down-counter that counts the “timer input “pulses and provides either a square wave or pulse when the terminal count is reached.

Address for the Low order byte of register: XXXXX100

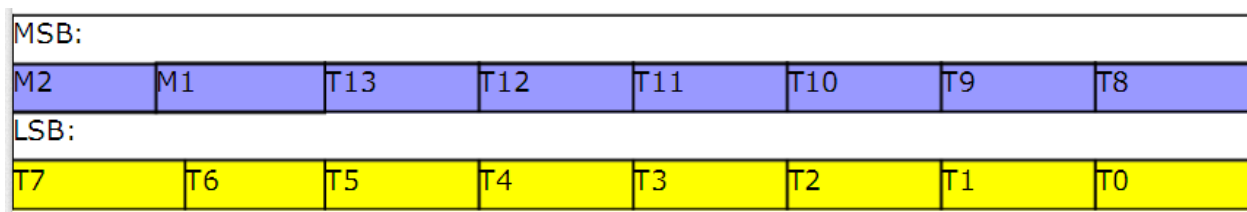
Address for the high order byte of register: XXXXX101

To Program the Timer, the COUNT LENGTH REGISTER is loaded first, one byte at a time, by selecting the timer addresses.

Bits 0~13 specify the count of next count

Bits 14~15 specify the Timer Output Mode.

The value of the count length register may have any value from 2H through 3FFFH on bits 0~13



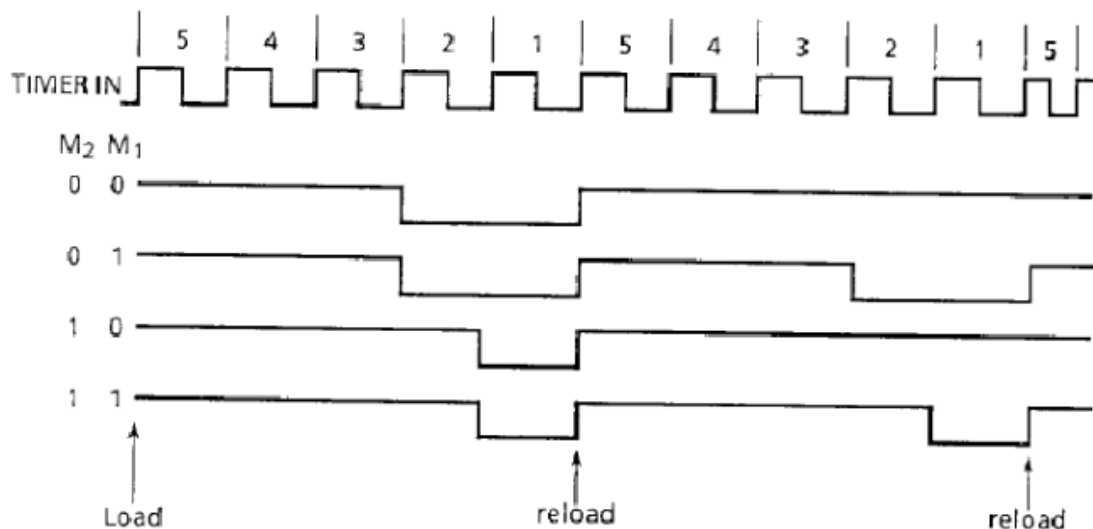
M2 M1

0 0 ——— Put out low during second half of count.

0 1 ——— Continuous square wave; The period of the square-wave equals the count length programmed with automatic reload at terminal count.

1 0 ——— Single pulse upon TC being reached.

1 1 ——— Continuous pulses.



b. Draw and explain the format of timer control register TCON.

Ans b. **Format of timer control register TCON** -- 2marks

Explanation -- 2marks

TCON: TIMER/COUNTER CONTROL REGISTER. BIT ADDRESSABLE.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

Fig: Format of timer control register TCON.

TF1 TCON. 7 Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.

TR1 TCON. 6 Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ON/OFF.

TF0 TCON. 5 Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.

TR0 TCON. 4 Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.

IE1 TCON. 3 External Interrupt 1 edge flag. Set by hardware when External Interrupt edge is detected. Cleared by hardware when interrupt is processed.

IT1 TCON. 2 Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

IE0 TCON. 1 External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.

IT0 TCON. 0 Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

c) Explain power saving option of 8051 microcontroller

Ans c. **Suitable Diagram** -- 2Marks

Explanation -- 2Marks

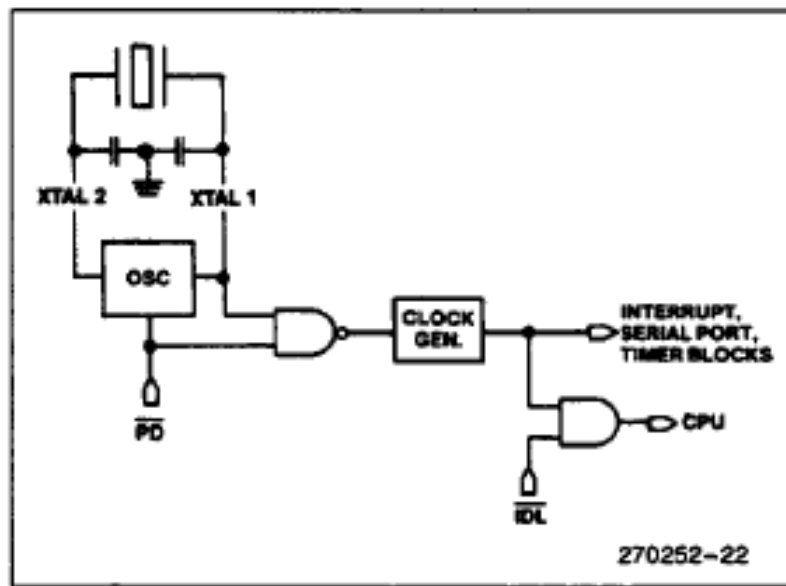


Fig: Power saving option 8052 microcontroller.

PCON: POWER CONTROL REGISTER. NOT BIT ADDRESSABLE

SMOD	--	--	--	GF1	GF0	PD	IDL
------	----	----	----	-----	-----	----	-----

Fig: Format of PCON

SMOD	Double Baud rates bit. When set to 1 and Timer 1 is used to generate baud rate, and the serial port is used in modes 1, 2, or 3
PD	Power Down bit. Setting this bit activates power down operation
IDL	Idle mode bit. Setting this bit activates idle mode operation

IDLE MODE

In the Idle mode, the internal clock signal is gated off to the CPU, but not to the Interrupt, Timer, and Serial Port functions.

The CPU status is preserved in its entirety, the Stack Pointer, Program Counter, Program Status Word, Accumulator, and all other registers maintain their data during Idle. The port pins hold the logical state they had at the time idle mode was activated. ALE and PSEN hold at logic high levels.

There are two ways to terminate the Idle mode.

- Activation of any enabled interrupt will cause PCON.O to be cleared and idle mode is terminated.
- Hard ware reset: that is signal at RST pin clears IDEAL bit IN PCON register directly. At this time, CPU resumes the program execution from where it left off.

POWER DOWN MODE

An instruction that sets PCON.1 causes that to be the last instruction executed before going into the Power Down mode. In the Power Down mode, the on-chip oscillator is stopped. With the clock frozen, all functions are stopped, but the on-chip RAM and Special Function Register are maintained held. The port pins output the values held by their respective SFRs. ALE and PSEN are held low.

Termination from power down mode: an exit from this mode is hardware reset. Reset defines all SFRs but doesn't change on chip RAM.

d) Write a program to add two BCD numbers stored in register bank 1.

Ans d. **Correct program – 4Marks**

```
CLR PSW.4      ;Select Register Bank 1 of internal RAM
SETB PSW.3
MOV A,R2       ;Load first number in Accumulator
ADD A,R3       ;Add first number with second number
DAA            ;Decimal adjust Accumulator after addition
MOV R4,A       ;Store result in internal memory
LOOP:AJMP LOOP ;Stop
```

e) Draw and explain format of IP register of 8051 microcontroller.

Ans e. **Format of IP register –2Marks**

Explanation --2Marks

D7							D0
--	--	PT2	PS	PT1	PX1	PT0	PX0

Priority bit = 1 assigns high priority. Priority bit = 0 assigns low priority.

--	IP.7	Reserved
--	IP.6	Reserved
PT2	IP.5	Timer 2 interrupt priority bit (8052 only)
PS	IP.4	Serial port interrupt priority bit
PT1	IP.3	Timer 1 interrupt priority bit
PX1	IP.2	External interrupt 1 priority bit
PT0	IP.1	Timer 0 interrupt priority bit
PX0	IP.0	External interrupt 0 priority bit

User software should never write 1s to unimplemented bits, since they may be used in future products.

f) Draw interfacing schematic of D/A converter with 8085 microprocessor.

Ans f. **Correct interfacing schematic of D/A converter – 4 Marks**

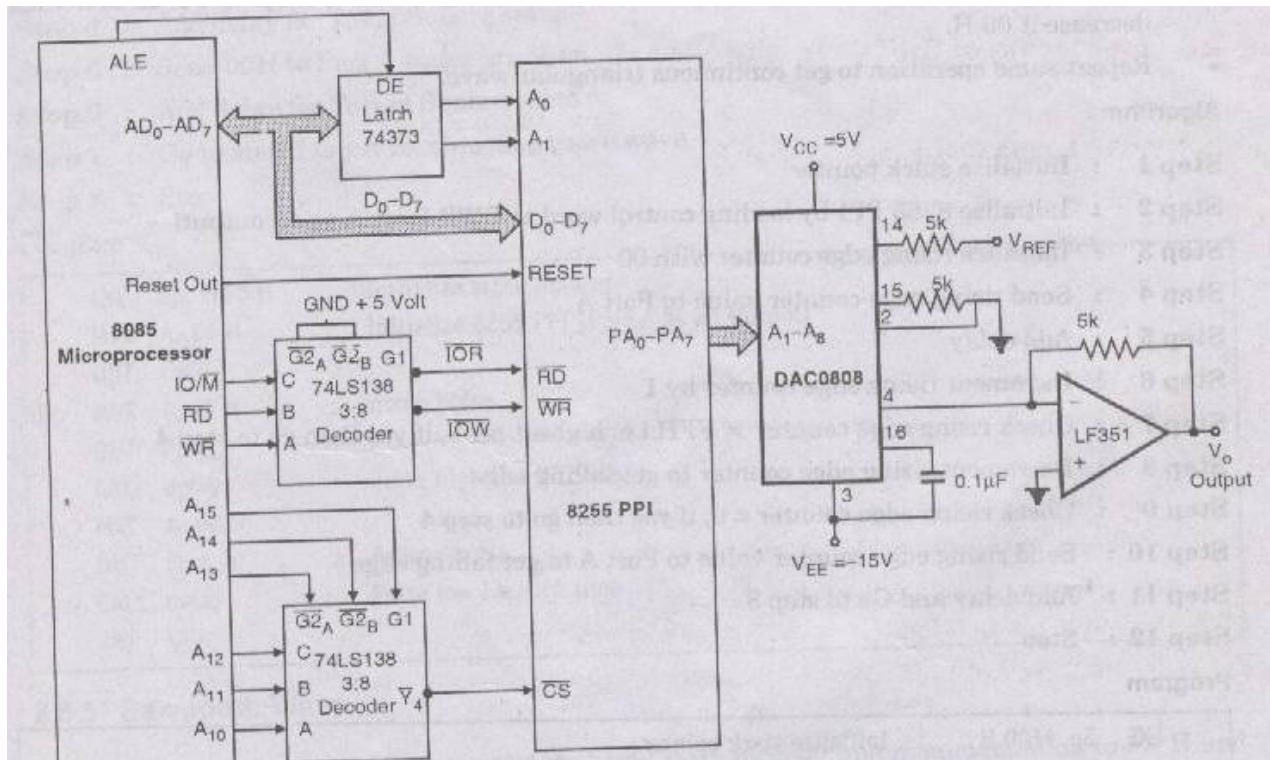


Fig: Interfacing schematic of D/A converter with 8085 microprocessor

Q4) Attempt any two of the following:

(16 Marks)

a) Draw and explain interfacing of A to D converter with 8085.

Ans a. **Suitable diagram -- 4Marks**

Explanation -- 4Marks

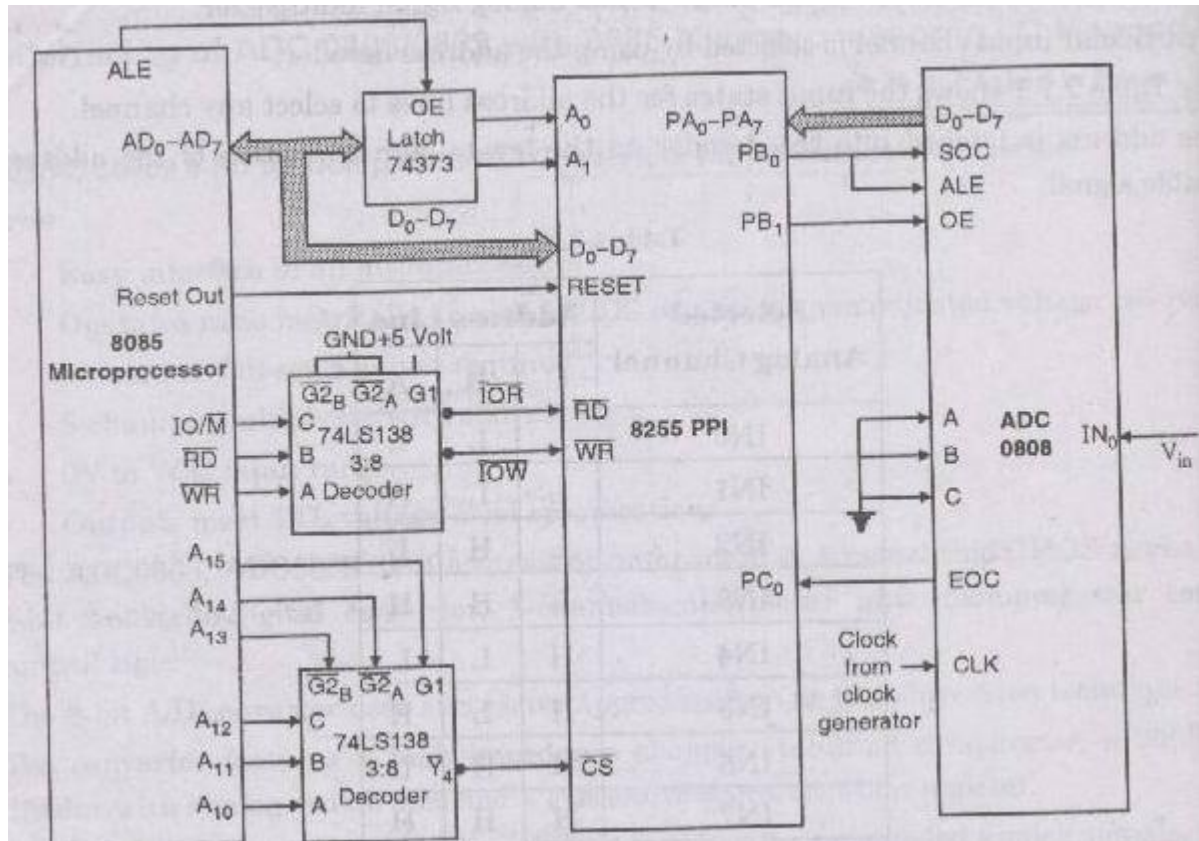


Fig: Interfacing A to D convertor with 8085

- The complete interfacing of 0808/0809ADC with microprocessor 8085 through 8255 PPI using interrupt mode is shown in above diagram. Interfacing of 8255 PPI with microprocessor is in I/O mapped I/O.
- D0-D7 pins of ADC 0808 are connected to port A pins PA0-PA7.
- The ALE and SOC are connected to PB0 through which we can issue Start of Conversion command and enable internal address latch.
- The OE is connected to PC5 so that output from ADC0809 after conversion can make available on Port A lines by issuing Output Enable command.
- The end of conversion EOC signal can be read from port C; hence the EOC pin is connected to PC4 of the port C of 8255PPI.
- The input selection pin A, B and C of ADC 0809 are grounded to select only IN0.
- The PC3 pin of 8255 is connected to RST5.5 interrupt of 8085, so that 8085 can be interrupted to read data from ADC 0808.
- When SOC and ALE signals are received by ADC0808 from 8085 through 8255 PPI, the ADC starts the conversion.
- After the conversion ADC0808 generates the EOC signal which will be received by 8255 on PC4 as a STROBE signal.

- After receiving EOC signal 8255 generates OE signal on PC5 and interrupt signal on PC3 which is connected to RST5.5 interrupt of 8085.
- The 8085 receives interrupt signal on RST5.5 and executes ISR to read digital data from port A.
- The control word to initialize 8255 is given when Port A is configured as an input, Port B as an output, Port C lower as input in mode 1
- Clock to the ADC 0808 is connected to external clock generating circuitry. The control word is

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	
1	0	1	1	1	0	0	1	=B9H

The start of conversion command is given in figure where D₀ bit is 1 i.e PB0 =1 and then a D₀ bit is 0 to get one high pulse to start conversion.

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	
0	0	0	0	0	0	0	1	=01H

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	
0	0	0	0	0	0	0	0	=00H

b) Explain the four addressing modes of 8051 microcontroller with suitable example.

Ans b. **Any four Addressing modes with suitable examples – 8Marks**

There are a number of addressing modes available to the 8051 instruction set, as follows:

1. Immediate Addressing mode
2. Register Addressing mode
3. Direct Addressing mode
4. Register Indirect addressing mode
5. Relative Addressing mode
6. Absolute addressing mode
7. Long Addressing mode
8. Indexed Addressing mode

1) Immediate Addressing mode:

Immediate addressing simply means that the operand (which immediately follows the Instruction op. code) is the data value to be used.

For example the instruction:

MOV A, #25H ; Load 25H into A

Moves the value 25H into the accumulator The # symbol tells the assembler that the immediate addressing mode is to be used.

2) Register Addressing Mode:

One of the eight general-registers, R0 to R7, can be specified as the instruction Operand. The assembly language documentation refers to a register generically as Rn.

An example instruction using register addressing is :

ADD A, R5 ; Add the contents of register R5 to contents of A (accumulator)

Here the contents of R5 are added to the accumulator. One advantage of register addressing is that the instructions tend to be short, single byte instructions.

3) Direct Addressing Mode:

Direct addressing means that the data value is obtained directly from the memory location specified in the operand.

For example consider the instruction:

MOV R0, 40H; Save contents of RAM location 40H in R0.

The instruction reads the data from Internal RAM address 40H and stores this in the R0. Direct addressing can be used to access Internal RAM, including the SFR registers.

4) Register Indirect Addressing Mode:

Indirect addressing provides a powerful addressing capability, which needs to be appreciated.

An example instruction, which uses indirect addressing, is as follows:

MOV A, @R0; move contents of RAM location whose address is held by R0 into A

Note the @ symbol indicated that the indirect addressing mode is used. If the data is inside the CPU, only registers R0 & R1 are used for this purpose.

5) Relative Addressing Mode:

This is a special addressing mode used with certain jump instructions. The relative address, often referred to as an offset, is an 8-bit signed number, which is automatically added to the PC to make the address of the next instruction. The 8-bit signed offset value gives an address range of + 127 to -128 locations.

Consider the following example:

SJMP LABEL_X

An advantage of relative addressing is that the program code is easy to relocate in memory in that the addressing is relative to the position in memory.

6) Absolute addressing Mode:

Absolute addressing within the 8051 is used only by the AJMP (Absolute Jump) and ACALL (Absolute Call) instructions.

7) Long Addressing Mode:

The long addressing mode within the 8051 is used with the instructions LJMP and LCALL. The address specifies a full 16 bit destination address so that a jump or a call can be made to a location within a 64KByte code memory space ($2^{16} = 64K$).

An example instruction is:

LJMP 5000h; full 16 bit address is specified in operand

8) Indexed Addressing Mode:

With indexed addressing a separate register, either the program counter, PC, or the data pointer DPTR, is used as a base address and the accumulator is used as an offset address. The effective address is formed by adding the value from the base address to the value from the offset address. Indexed addressing in the 8051 is used with the JMP or MOVC instructions. Look up tables are easy to implement with the help of index addressing.

Consider the example instruction:

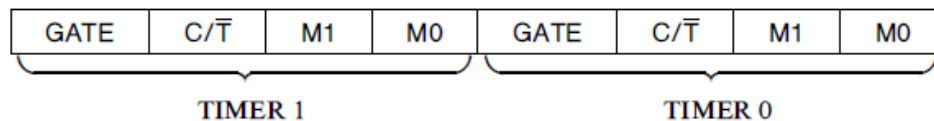
MOVC A, @A+DPTR

MOVC is a move instruction, which moves data from the external code memory space. The address operand in this example is formed by adding the content of the DPTR register to the accumulator value. Here the DPTR value is referred to as the base address and the accumulator value is referred to as the index address.

c) Draw and explain format of TMOD register and explain different timer modes of operation of 8051.

Ans c. **Format of TMOD register** -- 4Marks
Different timer modes operation of 8051 -- 4Marks

TMOD: TIMER/COUNTER MODE CONTROL REGISTER. NOT BIT ADDRESSABLE.



GATE When TR_x (in TCON) is set and GATE = 1, TIMER/COUNTER_x will run only while INT_x pin is high (hardware control). When GATE = 0, TIMER/COUNTER_x will run only while TR_x = 1 (software control).

C/ \bar{T} Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).

M1 Mode selector bit. (NOTE 1)

M0 Mode selector bit. (NOTE 1)

NOTE 1:

M1	M0	Operating Mode
0	0	0 13-bit Timer (MCS-48 compatible)
0	1	1 16-bit Timer/Counter
1	0	2 8-bit Auto-Reload Timer/Counter
1	1	3 (Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits, TH0 is an 8-bit Timer and is controlled by Timer 1 control bits. (Timer 1) Timer/Counter 1 stopped.

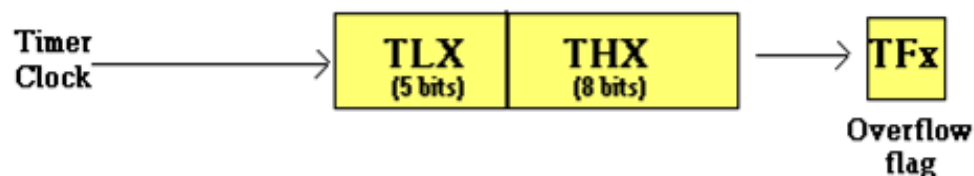
Fig: Format of TMOD register

To operate the timer/counter the bit TR1/TR0 in TCON register must be set and gate bit in TMOD register must be reset or INT1/INT0 i/p pin should be 1

Operating modes of Timer: The timer may operate in any of the four modes that are determined by M1 and M0 bit in TMOD register.

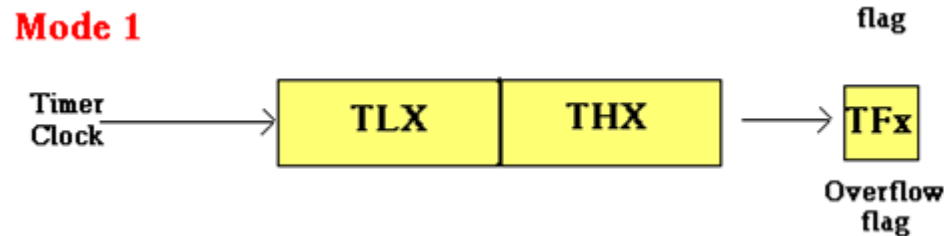
Mode 0:

Mode 0



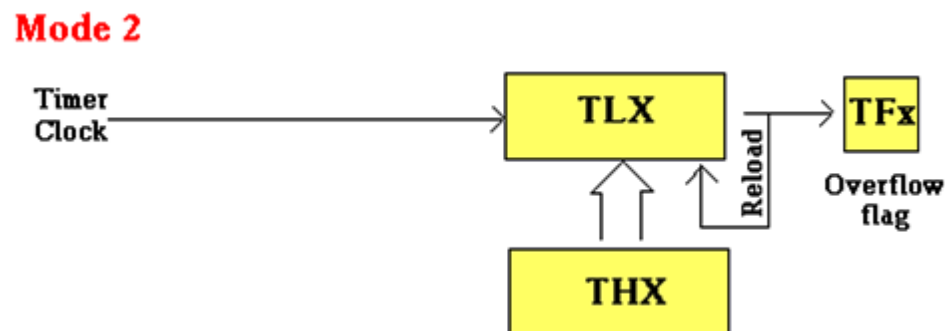
In mode 0 the register THX is used as 8 bit counter and TLX is used as 5 bit counter. The pulse i/p is divided by $(32)_{10}$ so that TH counts. Hence original oscillator frequency is divided by $(384)_{10}$. The timer flag is set when THX rolls over from FF to 00H.

Mode 1:



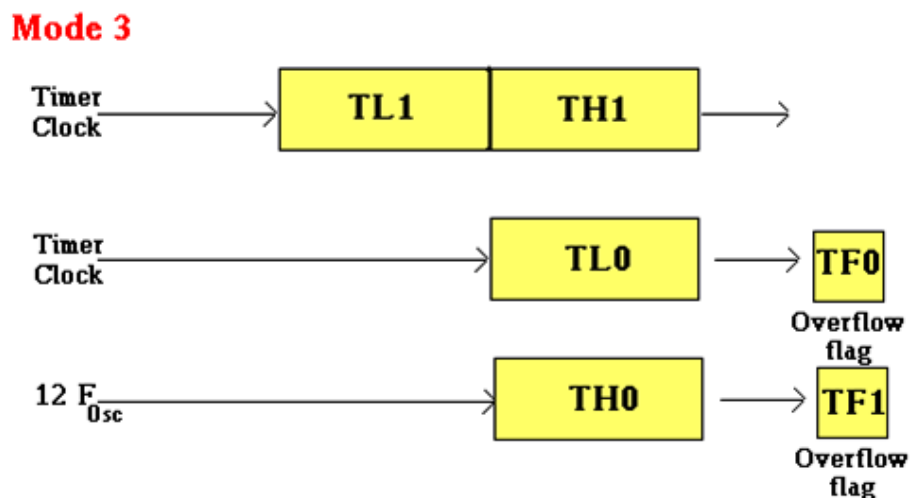
It is similar to Mode 0 except TLX is configured as a full 8-bit counter. Hence pulse input is divided by 256_{10} so that TH counts the timer flag is set when THX rolls over from FF to 00H.

Mode 2:



In this mode only TLX is used as 8-bit counter. THX is used to hold the value which is loaded in TLX initially. Everytime TLX overflows from FFH to 00H the timer flag is set and the value from THX is automatically reloaded in TLX register.

Mode 3:



In this mode, timer 0 becomes two completed separate 8-bit timers. TL0 is controlled by gate arrangement of timer 0 and sets timer 0 flag when it overflows. TH0 receives the timer clock under the

control of TR1 bit and sets TF1 flag when it overflows. Timer 1 may be used in mode 0, 1 and 2 with one important exception that no interrupt will be generated by the timer when the timer 0 is using TF1 overflow flag.

5. Attempt **any four** of the following:

(16 Marks)

a) State the difference between 8255 PPI and 8155 (Four points).

Ans a. **Any Four difference points -- 4 Marks**

Sr. No.	8255	8155
1.	Three 8 bit I/O Ports i.e. Port A, Port B and Port C	Two 8 bit I/O ports i.e. Port A, Port B and one 6 bit I/O Ports i.e. Port C
2.	No timer	Inbuilt 13 bit timer
3.	Separate data bus D ₀ -D ₇ and address lines A ₀ and A ₁	Multiplexed AD ₀ -AD ₇ bus
4.	IO/M, ALE signals are not available.	IO/M, ALE signals are available
5.	No memory	Inbuilt 256 byte of RAM
6.	A ₀ , A ₁ address lines are used to select I/O ports, CWR.	A ₀ , A ₁ , A ₂ lines are used to select ports, CWR and timer register.
7.	Port C is 8 bit can be used as two 4 bit ports.	Port C is a 6 bit port

b) What is stack? Describe stack operation in 8051 with one example.

Ans b. **Definition of stack -- 1 Mark**

Stack Operation -- 3 Marks

Stack: The stack is a part of RAM used by the CPU to store information temporarily.

This information may either be data or an address.

The CPU needs this storage area as there are only a limited amount of registers. The register used to access the stack is called the stack pointer which is an 8-bit register.

So, it can take values of 00 to FF H. When the 8051 is powered up, the SP register contains the value 07, i.e., the RAM location value 08 is the first location being used for the stack by the 8051 controller. There are two important functions to handle this stack. One is the PUSH and other is the POP. The loading of data from the CPU registers to the stack is done by the PUSH and loading of the contents of the stack back into a CPU register is done by POP.

EX. MOV R6, #35 H

MOV R1, #21 H

PUSH 6

PUSH 1

In the above instructions the contents of the Registers R6 and R1 are moved to stack and they occupy the 08 and 09 locations of the stack. Now the contents of the SP are incremented by two and it is 0A. Similarly POP 3 instruction pops the contents of the stack into R3 register.

Now the contents of the SP is decremented by 1.

In 8051 the RAM locations 08 to 1F (24 bytes) can be used for the stack. In any program if we need more than 24 bytes of stack, we can change the SP point to RAM locations 30-7F H. This can be done with the instruction MOV SP,#XX.

c) What are the different assembly language program development tools? Explain in brief.

Ans.- **Each program development tool – 1 Mark**

Assembly language programming tools:

- 1) Editor
- 2) Assembler
- 3) Linker
- 4) Object-Hex converter

1) Editor: An editor is a program which helps you to construct your assembly language program in right format so that the assembler will translate it correctly to machine language. So, you can type your program using editor. This form of your program is called as source program and extension of program must be .asm or .src depending on which assembler is used.

The DOS based editor such as EDIT , Wordstar, and Norton Editor etc. can be used to type your program.

2) Assembler: An assembler is programs that translate assembly language program to the correct binary code for each instruction i.e. machine code and generate the file called as Object file with extension .obj and list file with extension .lst extension.

Some examples of assembler are ASEM-51, Keil's A51, AX 51 and C51, Intel PL/M-51 etc.

3) Linker: A linker is a program, which combines, if requested , more than one separately assembled object files into one executable program, such as two or more programs and also generate .abs file and initializes it with special instructions to facilitate its subsequent loading the execution.

Some examples of linker are ASEM-51 BL51, Keil u Vision Debugger, LX 51 Enhanced Linker etc.

4) Object-Hex converter: The Object – HEX converter creates Intel HEX files from absolute object modules i.e. .abs file. Intel HEX files are ASCII files that contain a hexadecimal representation of your program.

They may be easily loaded into a device programmer for writing EPROMS or other memory devices. Example of object to HEX converter is Keil's OH51.

d) Explain the function of following directives:

- i) DB ii) EQU iii) ORG iv) END

Ans d. **Explanation of each directive --1 Mark**

i) DB:- Data Byte

Syntax:

LABEL:	DB	BYTE
--------	----	------

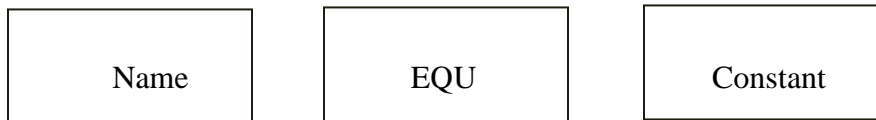
Where byte is an 8-bit number represented in either binary, Hex, decimal or ASCII form. There should be at least one space between label & DB.

The colon (:) must present after label. This directive can be used at the beginning of program. The label will be used in program instead of actual byte. There should be at least one space between DB & a byte.

ii) EQU: Equate

It is used to define constant without occupying a memory location.

Syntax:

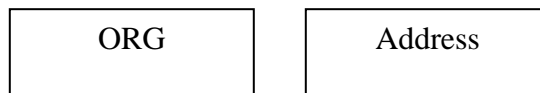


By means of this directive, a numeric value is replaced by a symbol. For e.g. MAXIMUM EQU 99
After this directive every appearance of the label “MAXIMUM” in the program, the assembler will interpret as number 99 (MAXIMUM=99).

iii) ORG:- Origin

It is used to indicate the beginning of address.

Syntax:



The address can be given in either hex or decimal there should be a space of at least one character between ORG & address fields. Some assemblers use ORG should not begin in label field.

iv) END:

This directive must be at the end of every program. meaning that in the source code anything after the END directive is ignored by the assembler.

This indicates to the assembler the end of the source file(asm).

Once it encounters this directive, the assembler will stop interpreting program into machine code.

e.g. END ; End of the program.

e)Write the format of PCON register and explain it.

Ans e. **Format of PCON** -- 2 Marks

Explanation -- 2 Marks

Register PCON

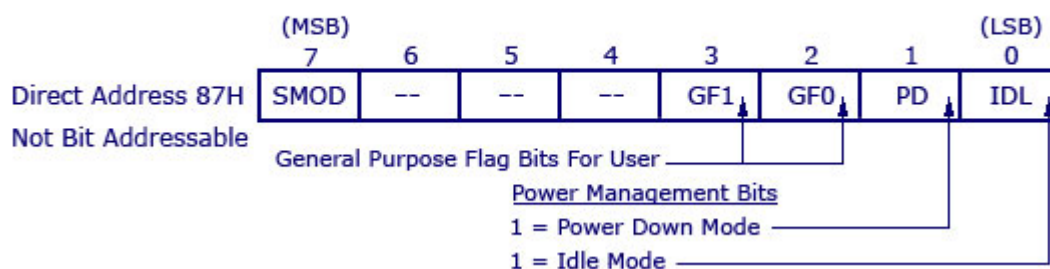


Fig: Format of PCON register

SMOD	Double Baud rates bit. When set to 1 and Timer 1 is used to generate baud rate, and the serial port is used in modes 1, 2, or 3
GF1	General purpose bit (available for use)
GF0	General purpose bit (available for use)
PD	Power Down bit. Setting this bit activates power down operation
IDL	Idle mode bit. Setting this bit activates idle mode operation

f. Classify instruction set of 8051. Give two examples for each instruction set.

Ans f . **Classification of instruction** -- 1Mark
Two examples for each instruction set -- 3Marks

The 8051 microcontroller instructions are divided among five functional groups:

1. Arithmetic
2. Branch
3. Data transfer
4. Logical
5. Boolean

1. **Arithmetic instructions:** perform several basic operations such as addition, subtraction, division, multiplication etc. After execution, the result is stored in the first operand. For example:

Mnemonic	Description
ADD A, Rn	Adds the register to the accumulator
ADD A, direct	Adds the direct byte to the accumulator

2. **Branch instruction:** There are two kinds of branch instructions:

Unconditional jump instructions: upon their execution a jump to a new location from where the program continues execution is executed.

Conditional jump instructions: a jump to a new program location is executed only if a specified condition is met. Otherwise, the program normally proceeds with the next instruction. For example,

Branch Instructions	
Mnemonic	Description
ACALL addr 11	Absolute subroutine call
LCALL addr16	Long subroutine call

3. Data Transfer Instructions

Data transfer instructions move the content of one register to another. The register the content of which is moved remains unchanged. If they have the suffix "X" (MOVX), the data is exchanged with external memory. For example,

Data transfer Instructions	
Mnemonic	Description
MOV A,Rn	Moves the register to the accumulator
MOV A,direct	Moves the direct byte to the accumulator

4. Logic instructions

Logic instructions perform logic operations upon corresponding bits of two registers. After execution, the result is stored in the first operand. For example,

LOGIC INSTRUCTIONS	
Mnemonic	Description
ANL A,Rn	AND register to accumulator
ORL A,Rn	OR register to accumulator

5. Boolean Instructions

Similar to logic instructions, bit-oriented instructions perform logic operations. The difference is that these are performed upon single bits. For example,

BIT-ORIENTED INSTRUCTIONS	
Mnemonic	Description
CLR C	Clears the carry flag
CLR bit	Clears the direct bit
SETB C	Sets the carry flag

a) Draw interfacing diagram of stepper motor with 8085 microprocessor using IC 8255. Write a program for rotating stepper motor by 90° in anticlockwise direction. Assume 1.8° step angle.

Ans a. **Interfacing Diagram -- 4 Marks**

Correct Program -- 4 Marks

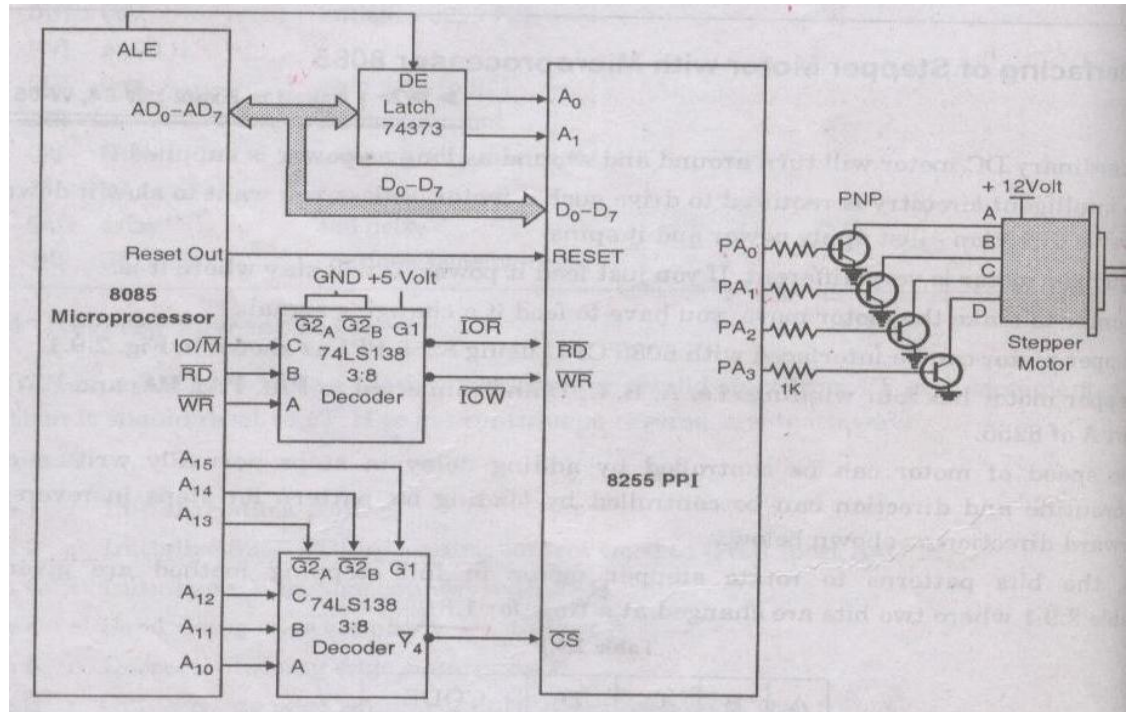


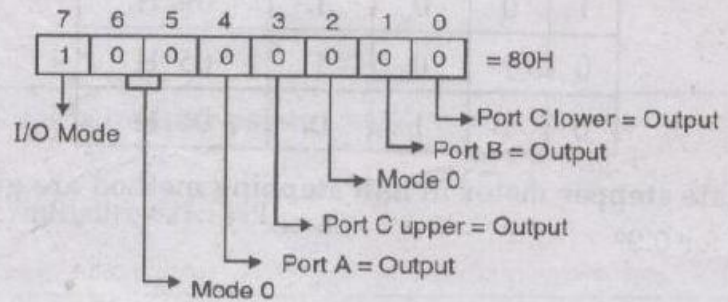
Fig: Interfacing Stepper motor to IC 8085 using 8255

Stepper motor has four windings i.e. A, B, C, D and connected to PA0, PA1, PA2, PA3 of a port of 8255.

The bits pattern to rotate stepper motor in 1.8 degree i.e. in full stepping method is as follows:

A	B	C	D	CODE
1	0	1	0	0AH
1	0	0	1	09H
0	1	0	1	05H
0	1	1	0	06H

Control Word for 8255 PPI



Program:

```

LXI SP, FFE0 H           ; Initialize stack pointer
MVI A, 80H
OUT CWR                   ; Initialize 8255 PPI
UP: LXI H, C200 H         ; Initialize memory pointer for look up table
MVI C, 04 H               ; Initialize byte counter
UP 1: MOV A, M             ; Read bits pattern i.e. step code
OUT PORT_A                ; write to port A
CALL DELAY                ; add delay between steps
INX H                     ; increment memory pointer for look up table
DCR C                     ; DECREMENT BYTE COUNTER BY 1

JNZ UP1                   ; if byte counter ≠ 0 then go to UP1
JMP UP                     ; Continuous rotation

```

b. Write a program to find largest number from the array of ten numbers stored in external RAM.

Ans b. Correct Program -- 8 Marks

Program:

```

CLR PSW.3                 ; Select Bank 0
CLR PSW.3
MOV R1, 0AH               ; Initialize byte counter
MOV DPTR, # 3000H         ; Initialize memory pointer
DEC R1                    ; Decrement byte counter by 1
MOV X A, @DPTR            ; Load number in accumulator
MOV 40 H, A               ; Store number in memory location
UP: INC DPTR               ; Increment memory pointer by 1
MOVXA, @DTPR              ; Read next number

CJNE A, 40 H, DN           ; if number≠ next number, and then go to NEXT
DN: JC NEXT                ; If next number > number then go to NEXT
MOV 40H, A                 ; Else replace NEXT number with number
NEXT: DJNZ R1, UP          ; Decrement byte counter by 1, if byte counter≠ 0 then go to UP
INC DPTR                   ; Increment memory pointer by 1
MOV A, 40H
MOVX@ DPTR, A              ; Store result t external memory location
LOOP: AJMP LOOP            ; Stop

```

- c. Write a program to generate continuous square wave of 2 KHz on P1.4 using model of timer 0. Given crystal frequency = 11.0592MHz.

Ans c. **Calculations --- 2Marks**

Delay --- 2 Marks

Program --- 4 Marks

Program:

Crystal frequency= 11.0592 MHz

I/P clock = $11.0592 \times 10^6 = 11.0592\text{MHz}$

$1/12 \times 11.0592\text{Mhz} = 921.6\text{ KHz}$

$T_{in} = 1.085\mu\text{ sec}$

For 2 kHz square wave

$F_{out} = 2\text{ KHz}$

$T_{out} = 1/2 \times 10^3$

$T_{out} = 500\mu\text{ sec}$

Consider half of it = $T_{out} = 250\mu\text{ sec}$

$N = T_{out} / T_{in} = 250/1.085 = 230$

$65536 - 230 = (65306_{10}) = \text{FF1A}$

```
ORG 0000
MOV TMOD, #01H ; Mode1
MOV TL0, #01AH ; Lower byte Of Timer0
MOV TH0, #0FFH ; higher byte Of Timer0
CPL P1.4 ; Toggle P1.4
ACALL DELAY
SJMP HERE
```

Delay using Timer0

```
DELAY: SETB TR0 ; Start Timer 0
AGAIN: JNB TF0, AGAIN
CLR TR0 ; Stop Timer 0
CLR TF0
RET
END
```