VTU Model Question Paper

Subject: Microcontrollers

code: 18EC46

Semester: 04

By:

Prof. Prasann D. Kulkarni

Head of the Department
Dept. of Electronic & Communication Engg.
KLS V.D.I.T., HALIYAL (U.K.)

Model Question Paper-2 with effect from 2019-20 (CBCS Scheme)

USN | | | | | | | | | | |

Fourth Semester B.E. Degree Examination
Subject Title: MICROCONTROLLER

TIME: 03 Hours **Max. Marks: 100**

Note: Answer any **FIVE** full questions, choosing at least **ONE** question from each **MODULE**.

		Module -1	*Bloom's Taxonomy Level	Marks
Q.01	a	With a neat diagram explain the architectural features of 8051 microcontroller.	L1	8
	b	Bring out the difference between Microprocessor and Microcontroller.	L1	4
	c	With diagrams explain the internal RAM structure of 8051 microcontroller.	L2	8
		OR		
Q.02	a	With simple block diagram explain the features of 8051 microcontroller.	L1	8
	b	Define microcontroller, mention its applications.	L1	4
	c	Interface 4k bytes RAM and 8k bytes ROM to 8051 microcontroller in such a way that starting address of RAM is 1000H and ROM is C000H.	L2	8
		Module-2		
Q. 03	a	Explain 5 different addressing modes with examples.	L1	8
	b	Check the correctness of the following instruction. If wrong correct them: 1. `CJNE @RI, #D_ADDRESS, REL8` 2. `ADDC @RI, A` 3. `DJNZ #DATA, REL8` 4. `MOVX @DPTR, R1`	L2	8
	c	Write an ALP to convert a packed BCD number into two ASCII numbers. Store the result in R5 and R6 respectively.	L3	4
		OR		
Q.04	a	Define assembler directives. With example explain all the assembler directives supported by 8051 microcontroller.	L1	8
	b	Explain the following instructions, also mention how many bytes it takes to store in ROM: 1. `DJNZ Rn, R_ADDRESS` 2. `JNC R_ADDRESS` 3. `DA A` 4. `MOVX A, @A+<BASE-REG>`	L2	8
	c	Write an ALP to convert a Binary number to packed BCD number (hexadecimal to decimal). The binary number is stored at 40h location. Store the converted packed BCD number at 50h and 51h internal RAM location.	L3	4
		Module-3		

Karnatak Law Society's
Vishwanathrao Deshpande Institute of Technology, Haliyal
(Formerly Known as KLS Vishwanathrao Deshpande Rural Institute of Technology, Haliyal)
(Approved by AICTE, New Delhi. Affiliated to VTU, Belagavi)
Udyog Vidya Nagar, Haliyal – 581329, Dist: Uttar Kannada
Phone: 08284-220861, 220334, 221409, Fax: 08284-220813
Web: www.vdrit.org email: klsvdrit@yahoo.com

Ref : KLS/VDIT/ F13/ 1820 /2019-20 Dated:11.11.2019

BONAFIDE CERTIFICATE

This is to certify that, **Mr. Vitthal Dyamappa Malledi** bearing University Seat No. 2VD16CV060 is a bonafide student of this college presently studying in B.E VII semester Civil Engineering Branch. for the academic year 2019-20.

This certificate is issued as per the request of **Mr. Vitthal Dyamappa Malledi** vide his letter dated 11.11.2019 for the purpose of **Scholarship** only.

Principal

Principal
KLS Vishwanathrao Deshpande
Institute of Technology, Haliyal

Q. 05	a	With diagrammatical representation explain how stacks plays its role in subroutine operations.	L1	4
	b	Write an assembly language program to sort an array of n= 5 bytes of data in ascending order stored from location 30h. (Use bubble sort algorithm)	L3	8
	c	Write an assembly language program to count the number of 1's and 0's in an 8-bit data received from port P1. Store the count of 1's and 0's in 30h and 31h.	L3	8
		OR		
Q. 06	a	Write a note on subroutine instructions.	L1	4
	b	Write an assembly language program to sort an array of n= 5 bytes of data in descending order stored from location 30h. (Use bubble sort algorithm)	L3	8
	c	Assume a push button switch is connected to port pin P1.2, Write an assembly language program to monitor the switch and turn ON the LED's connected to port P2 as long as the switch is pushed.	L3	8
		Module-4		
Q. 07	a	Explain the bit contents of TCON and TMOD registers.	L1	8
	b	Write an assembly language program to transfer multi-byte data serially with 9600 baud rate.	L3	8
	c	Explain how timers are programed in mode 1.	L2	4
		OR		
Q. 08	a	Explain the bit contents of SCON and PCON registers.	L1	8
	b	Write an assembly language program to generate a square wave on port pin P1.2 of frequency 5k Hz.	L3	8
	c	Write a note on Asynchronous serial communication and data framing.	L2	4
		Module-5		
Q. 09	a	Explain the bit contents of IE register.	L1	4
	b	Explain how programming of external hardware interrupts is done in 8051 microcontrollers with a code snippet.	L2	6
	c	With neat diagram write an assembly language program to interface DAC to 8051 microcontroller.	L3	10
		OR		
Q. 10	a	Explain how interrupt priority can be changed using IP register. Also explain the default priorities assigned to interrupts in 8051 microcontroller.	L1	5
	b	Write a C program using interrupts to generate a square wave on port pin P1.2 of 1kHz using timer-0 in mode 2.	L2	5
	c	With neat diagram write an assembly language program to interface ADC-0804 to 8051 microcontroller.	L3	10

*Bloom's Taxonomy Level: Indicate as L1, L2, L3, L4, etc. It is also desirable to indicate the COs and POs to be attained by every bit of questions.

From Vitthal. Dyamappa. Malledi
Vitthal. Dyamappa Malledi

Branch - civil
Sem — 7th

USN : 2VD16CV060
Mobile No : 9738263092

To,
The Principal
VDRIT,
Haliyal

Respected Sir,

Sub: Issue of Character Certificate/Bonafide Certificate

I may please be issued Character certificate /Bonafide certificate for (Tick mark, which is applicable of the following.)

1. For obtaining passpost
2. For higher studies
3. For Future reference
4. Any other (please specify) scholarship.

Photocopy of the admission challan enclosed for your kind perusal vide challan No._____dated_____for Rs._____

Encl: Photo copy of the admission Challen

Vitthal
(Signature of the Candidate)

HOD remaks (applicable only for character certificate)

Mr/Miss. _____bears good moral character.

Signature of HOD

Office Remarks (Applicable only for B.E. Regular students)

Mr./Miss. _____ is bonafide student of this college studying in_____ year for the year _____admission taken vide Challan No._____dated_____

(Signature of the staff)

OFFICE NOTE (Examination section)

(Applicable only for those students who have completed B.E.)

Mr./Miss _____ was bonafide student of this college and completed his/her B.E. in the year _____ in CSE/ECE/EEE/MEE/CEE branch with FCD/FC/SC/Pass class. The said Office note is submitted with through verification of the office records.

Date:
Place : Haliyal

Signature of the case worker
(Examination Section)

Character Certificate fees of Rs.25/- received vide challan no._____dated_____

Bonafide Certificate fees of Rs. 25/- received vide challan no. 2938 dated 11/11/19

(Signature of the staff)

Issue Bonafide /Character Certificate

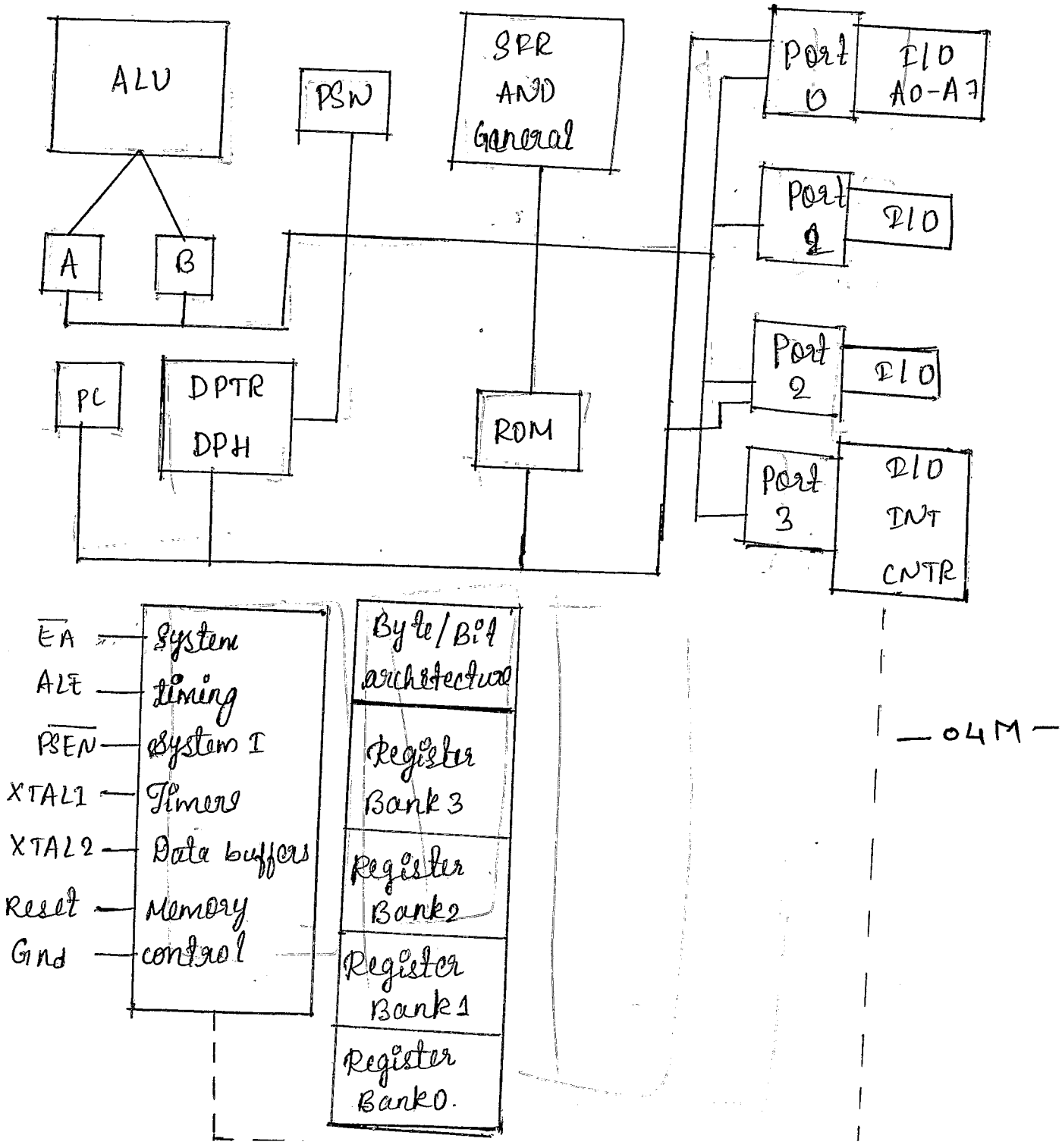
PRINCIPAL

Principal
KLS Vishwanathrao Deshpande
Institute of Technology, Haliyal

MODULE -1

Q 01 a] With a neat diagram explain the architectural features of 8051 microcontroller. — 08M —

⇒



— 04M —

- The 8051 consists of 8-bit arithematic and logic unit to perform arithematic operations like addition, substraction, multiplication and division. And also logical operations like AND, OR, clear etc. using registers A, B, SP, PSW, DPTR and PC
- Internal RAM is of 128 bytes organized into three distinct areas:
  i) 32 bytes of working registers from 00h to 1FH. Four banks with eight registers in each bank.
  ii) 16 bytes of bit addressable area.
  iii) General purpose RAM from 30h to 7Fh

02 M

- Internal ROM is of 4KB
- Registers :- Accumulator is general purpose 8-bit registers used for data transfer and arithematic operation.
  'B' register is used for arithematic operations and for data storage. 'SP' is stack pointer register which holds the address of the "top of stack". 'DPTR' is a 16-bit register to hold the 16-bit address used while receiving accessing external memory. 'PC' is a 16-bit register which holds the address of the next instruction to be executed. The PC is automatically incremented after every instruction byte is fetched. 'PSW' is an 8-bit flag register holding flags like carry flag, overflow flag, parity flag axillary carry flag etc.

02 M

- 8051 has two timers/counters to generate delay/counts events happening outside the microcontroller using registers TCON, TMOD TL0, TH0, TL1, TH1.

- The I/O ports :- 8051 has 32 I/O ports pins configured as 4 ports, port0, port1, port2, port3.

- 8051 has one serial port used for serial communication using SBUF & SCON registers.

Q.01 b) Bring out the difference between Microprocessor and Microcontroller. — 04M

⇒

Microprocessor	Microcontroller.
• A microprocessor is a general purpose digital computer central processing unit (CPU)	• A Microcontroller is a programmable digital processor unit with necessary peripherals.
• As it is only a processor, so memory and I/O components need to connected externally.	• Microcontroller has a processor along with external memory and I/O components.
• Microprocessor is equipped with mass storage devices, I/O peripherals and others.	• Microcontroller incorporate the features such as ALU, PC, SP registers. storage devices such as ROM, RAM, parallel I/O, serial I/O, counters & clock circuits.
• We find applications such as, performing extensive calculations, Accounting system and Military applications.	• Microcontrollers are mainly used in devices like, Mobile phone, Automobiles, cameras etc.

01 x 4 = 04M

Q.01 c) With diagrams explain the internal RAM structure of 8051 microcontroller. — 08M

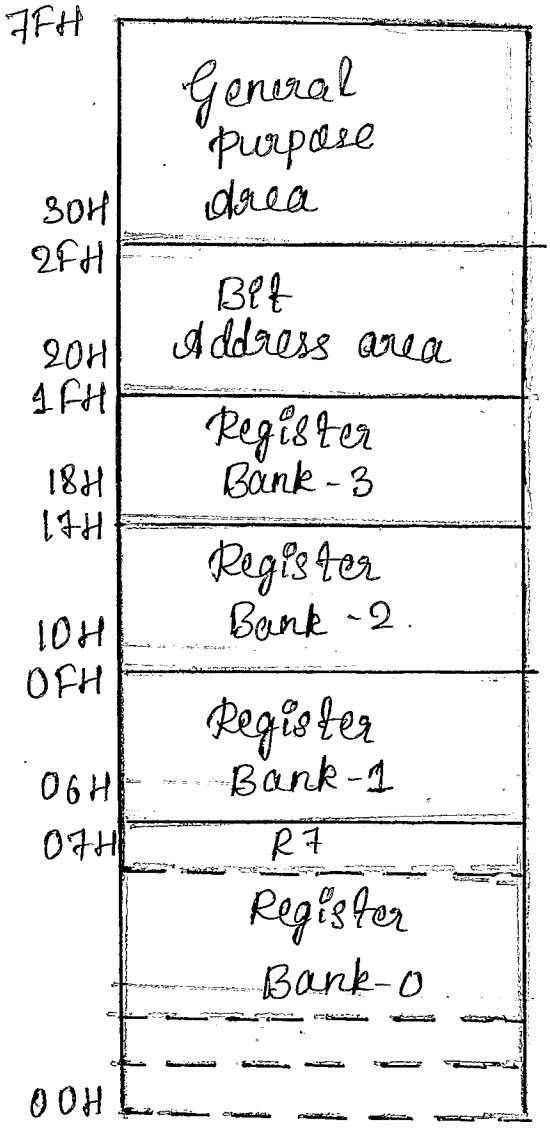
⇒ The 128 bytes internal RAM is organized into 3 distinct areas.

1] 32 bytes from address 00h to 1fh that make up 32 working registers organized as 4 memory banks of 8 registers each. The 4 register banks are numbered 0 to 3 and are made up of 8 registers named R0 to R7. Each register can be addressed by name or by its RAM address. Thus R0 of bank 3 is R0 or address 18h.

04M

Bits RS0 & RS1 in the PSW determine which bank of register is

currently in use at any time when program is running. Register banks not selected can be used as general purpose RAM. Bank0 is selected by default on reset.



16 bytes

32 bytes = 5 × 4 by

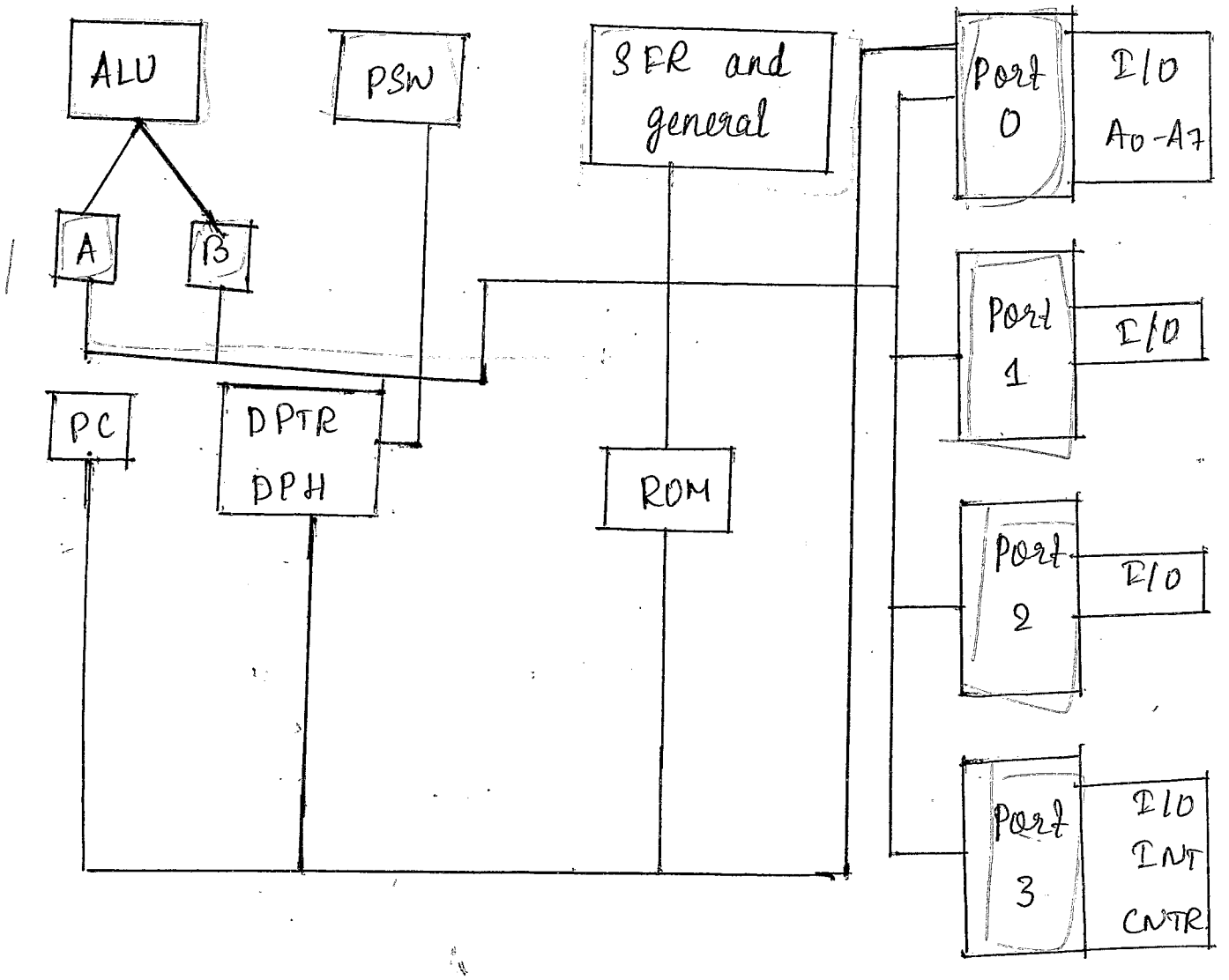
2) A bit addressable area of 16 bytes occupies RAM byte addresses 20h to 2fh, forming total 128 bits. An addressable bit may be specified by its bit address of 00h to 7fh or 8 bits may form any byte address from 20h to 2fh. For example bit address 4fh is also bit 7 of byte address 29h. Addressable bits are useful when the program need only remember a binary event.

3] d general purpose RAM area above the bit area from 30h to 7fh addressable as byte.

OR

Q. 02

(a) With simple diagram explain the features of 8051 micro controller — 08M —



04M

- The 8051 contains 8-bit arithematic and logic unit to perform arithematic operations like addition, subtraction, multiplication, division and logical operations like AND, OR clear etc. using registers A, B, SP, PSW, DPTR & PC

- Internal RAM is of 128 bytes organized into 3 distinct areas.

i) 32 bytes of working registers from 00h to 1Fh. Four banks with eight registers in each bank. (ii) 16 bytes of bit addressable area (iii) General purpose RAM from 30h to 7Fh.

- Internal ROM is of 4 KB
- Registers: Accumulator is a general purpose 8-bit register used for data transfer and arithematic operations.

  'B' register is used for arithematic operations like multiplication division and for data storage.

  'SP' is stack pointer register which holds the address of the 'Top of stack'. stack refers to area of internal RAM used to store and retrieve data quickly.

  'DPTR' is a 16-bit register to hold the 16-bit address used while accessing external memory.

  'PC' is a 16-bit register which holds the address of the next instruction to be executed.

  'PSW' is an 8-bit flag register holding flags like carry flag overflow flag, parity flag axillary carry flag etc.

- 8051 has two timers / counters to generate delay / count events happening outside the microcontroller using register TCON, TMOD, TL0, TH0, TL1, TH1

- I/O ports:- 8051 has 32 I/O pins configured as 4 ports, port 0 port 1, port 2, port 3.

- 8051 has one serial port used for serial communication using SUBF & SCON registers.
- Full Duplex serial Data receiver / transmitter.
- Two External and Internal Interrupt sources.

Q.No2

(b) Define Microcontroller, mention its applications. — 04M

⇒ A microcontroller is a programmable digital processor with necessary peripherals. It is a small computer on a single integrated chip with one or more CPUs along with memory and I/O peripherals.

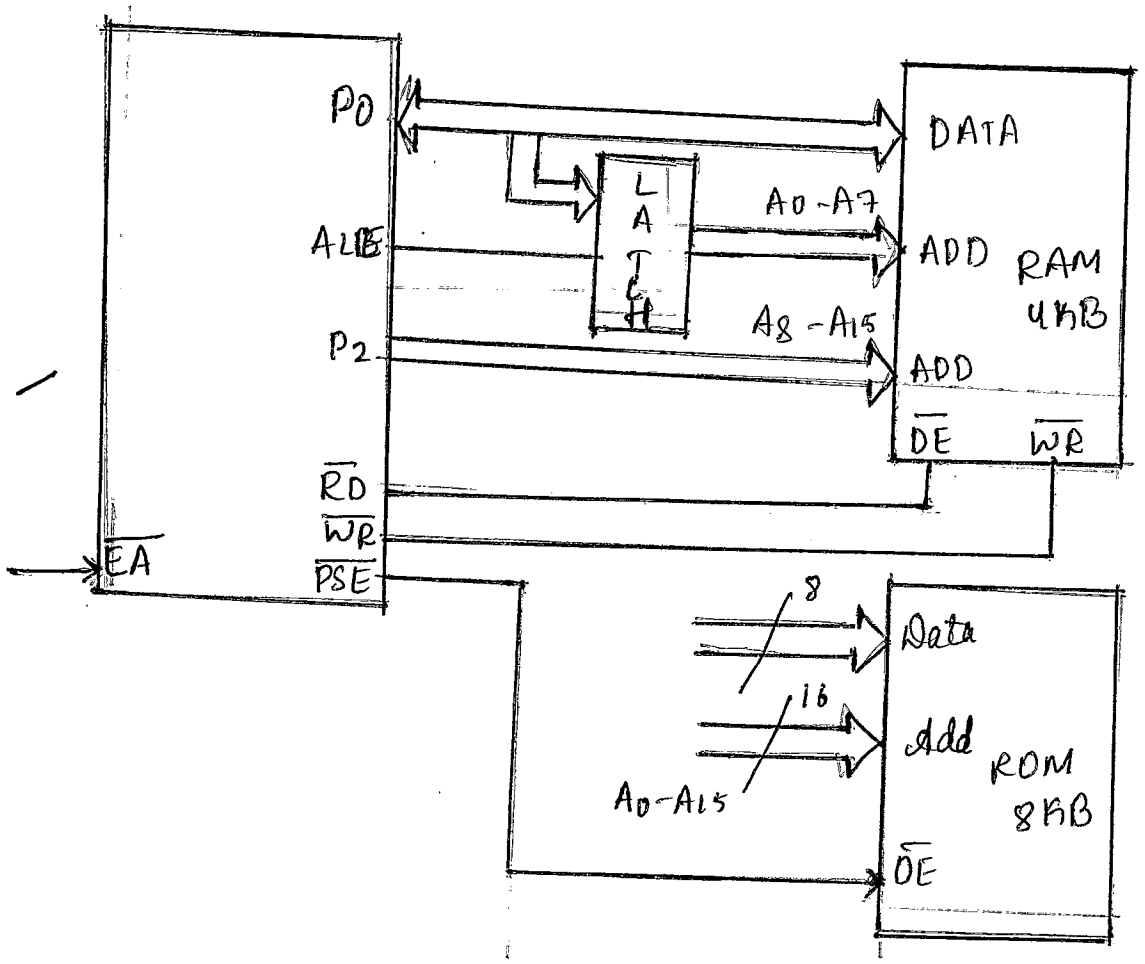
Applications of Microcontroller includes in devices like,

- Mobile phones, Automobiles, keyboard controllers, watches etc.
- The prime use of microcontroller is to control the operation of machine using a fixed program that is stored in ROM & that does not change over the lifetime of the system.
- It is also meant to read data, perform limited calculations on that data and control its environment based on those calculations.

01×4=04M

2(c) Interface 4K bytes RAM and 8K bytes ROM to 8051 microcontroller in such a way that starting address of RAM is 1000H and ROM is C000H. — 08M

⇒



— 04M —

External program memory is fetched if either of the following two conditions are satisfied.

$\overline{EA}$ - enable address is low. The microcontroller by default starts searching from program from external program memory.

PC is higher than FFFH

$\overline{PSEN}$ tells the outside world whether the external memory fetched is program memory or data memory.

The circuit design decides wheather the code is found totally in internal RAM, totally in external ROM or combination of both.

MODULE - 2

Q.NO

3(a) Explain 5 different addressing modes with examples. — 08M —

⇒ (i) Immediate Addressing mode.

- This addressing mode is named as immediate because it transfers an 8-bit data immediately to the accumulator.
- The immediate data must be preceded by the sign '#'
- It can load information into any registers including 16-bit DPTR register.

Eg:- MOV A, #n copy the immediate data-type n to A register.

(ii) Register addressing mode.

- Certain registers may be used as part of the opcode mnemonic as sources or destinations of data. — 02M —
- Registers A, DPTR and R0 to R7 may be named as part of the opcode. Other registers in the 8051 may be addressed using the direct addressing mode.
- Some assemblers can equate many of the direct address to the register name, so that register names may be used in lieu of registers address.

```
Eg:- MOV A, R0      ; copy contents of R0 into A.
     MOV R2, A      ; copy contents of A into A.
     ADD A, R5      ; add contents of R5 into A.
```

(iii) Direct Addressing mode.

- All 128 bytes of internal RAM and the SFRs may be addressed directly using the single byte address assigned to each RAM location and each special function register. — 02M —
- Internal RAM uses addresses from 00 to 7Fh to address each byte. The SFR addresses exist from 80h to FFh.
- It is used to access RAM locations 30-7Fh.

Eg:- MOV A, add copy data from direct address add to register A.

(iv) Indirect Addressing mode.

- The indirect addressing mode uses a register to hold the actual address that will finally be used in the data move.
- The register itself is not the address, but rather the number in the register.
- Indirect addressing for MOV opcodes uses register R0 or R1 often called 'data pointers' to hold the address of one of the data locations, which could be a RAM or an SFR address.
- The Eg:- MOV @Rp, #n ; copy the immediate byte n to address in Rp.

  MOV @Rp, add copy the contents of add to address in Rp

  MOV @Rp, A copy the data in A to the address in Rp.

  MOV add, @Rp copy the contents of the address in Rp to add.

  MOV A, @Rp copy the contents of the address in Rp to A.

-02M-

(v) Register Indirect addressing mode.

- In this addressing mode, address of the data is given in the register operand.
- Here, the value inside R0 is considered as an address, which holds the data to be transfered to accumulator.

-02M-

- Eg:- MOV A, @R0

If R0 holds the value 20H, and we have a data 2FH stored at the address 20H, then the value 2FH will get transferred to accumulator after executing this instruction. Hence opcode for this instruction is E6H

S.No

3 b) Check the correctness of the following instruction. If wrong correct them. — 08M

1. CJNE @RI , #D_ADDRESS
2. ADDC @RI , A
3. DJNZ #DATA , REL8
4. MOVX @DPTR , R7

02X4 = 08M

⇒ 1. CJNE @RI , #D_ADDRESS
This instruction is correct.

2. ADDC @RI , A
This instruction is correct

3. DJNZ #DATA , REL8
This instruct is incorrect
It should be register instead of DATA

DJNZ Rn DJNZ Rn, Rel8

4. MOVX @DPTR , R1
This instruction is correct.

Q.No 3(c) Write an ALP to convert a packed BCD number into two ASCII numbers. Store the result in R5 and R6 respectively. — 04M

⇒

```
MOV  A, #29H      ; A = 29H packed BCD
MOV  R5, A        ; keep a copy of BCD data in R5
ANL  A, #0FH      ; mask upper nibble (A=09)
ORL  A, #30H      ; make it an ASCII, A = 39H
MOV  R6, A        ; save it (R6 = 39H ASCII char)
MOV  A, R5        ; A = 29H, get the original data
ANL  A, #0F0H     ; mask the lower nibble (A=20)
RR   A            ; rotate right
RR   A            ; rotate right
RR   A            ; rotate right
RR   A            ; rotate right, (A=02)
ORL  A, #30H      ; A=32H, ASCII char '2'
MOV  R2, A        ; save ASCII char in R5.
END.
```

— 02M — — 02M —

Q.No. 4 a] Define assembler directives. With example explain all the directives supported by 8051 microcontroller. — 08M

⇒ Assembler directive tells the assembler to do something other than creating the machine code for an instruction.

— 02M —

(i) ORG (Origin)

The ORG directive is used to indicate the starting address. It can be used only when the program counter needs to be changed. The number that comes after ORG can be either in hex or in decimal.

- If the number is not followed by H. it is decimal and the assembler will convert it to hex.
- Some assemblers use ".ORG" instead of "ORG" for origin directive

Eg:- ORG 0000H ; set PC to 0000.

(ii) EQU (equate):-

- This directive is used to define a constant without occupying a memory location. The EQU directive does not set aside storage for a storage for a data item but associate a constant value with a data label so that content the label appears in program. Its constant value will be substituted for a label. We use EQU for the counter constant and then the constant is used to load R3 register. 02M

- Eg:- COUNT EQU 05
  MOV R3, #COUNT

(iii) DB (Define Byte)

The DB directive is used to define an 8 bit data. DB directive initializes memory with 8 bit value. The numbers can be in decimal, binary, hex or in ASCII formats. For decimal, the 'D' after the decimal number is optional, but for binary and hexadecimal, 'B' and are

Eg:- DATA 1 : DB 40H; Hex
DATA 2 : DB 01011100B ; binary 02M
DATA 3 : DB 48; decimal.
DATA 4 : DB 'go corona' ; ASCII

(iv) END

The end directive signals the end of the assembly module. It indicates the end of the program to the assembler. Any text in the assembly file that appears after the END directive is ignored. If the END statement is missing, the assembler will generate an error message. 02M

Q.NO 4 (b) Explain the following instructions, also mention how many bytes it takes to store in ROM. — 08M

1. DJNZ Rn, R_ADDRESS .. 02×4 = 08M
2. JNC R_ADDRESS 2byte
3. DA A
4. MOVX A, @A+<BASE-REG>.

⇒ 1. DJNZ Rn, R_ADDRESS

→ DJNZ is a jump instruction. ie., decrement register and Jump if not zero.

→ R_Address indicates a label and which initializes the loop. When initial value of register is 1, loop initializes and executes. When value of register becomes 0, the loop ends and program flow continues with further instruction. Everytime the loop runs, value of register is decremented by one.

→ It takes 2 bytes to store in ROM.

2. JNC R_ADDRESS

→ JNC branches to the address indicated by reladdr if the carry bit is not set. If the carry bit is set program execution continues with the following the instructions of the program.

→ It takes 2bytes to store

3. DA A → Decimal-adjust Accumulator for Addition.

→ DA Adjusts the contents of the accumulator to correspond to a BCD number after two BCD numbers have been added by ADD or ADDC instruction. If the carry bit is set or if the value of bits 0-3 exceed 9, 0x06 is added to the accumulator. It takes 1 byte to store.

4. MOVX A, @A+<BASE-REG>

→ The registers A and base registers are added. The data value obtained is the address. The value at this address is moved to A, i.e., Accumulator. It takes 1 byte to store in ROM

Q.No 4(c) Write an ALP to convert a binary number to packed BCD number (Hexadecimal to decimal). The binary number is stored at 40h location. Store the converted packed BCD number at 50h and 51h internal RAM location. — 04M

⇒

```
org 00h
mov dptr,#40h
mov 0f0h, #064
div ab
inc dptr
movx @dptr,a
mova, 0f0h
mov 0f0h,#0ah
divab;
inc dptr;
movx @dptr,a
in dptr
mov a, 0f0h
movx @dptr,a
simp here
end.
```

-02M

-02M

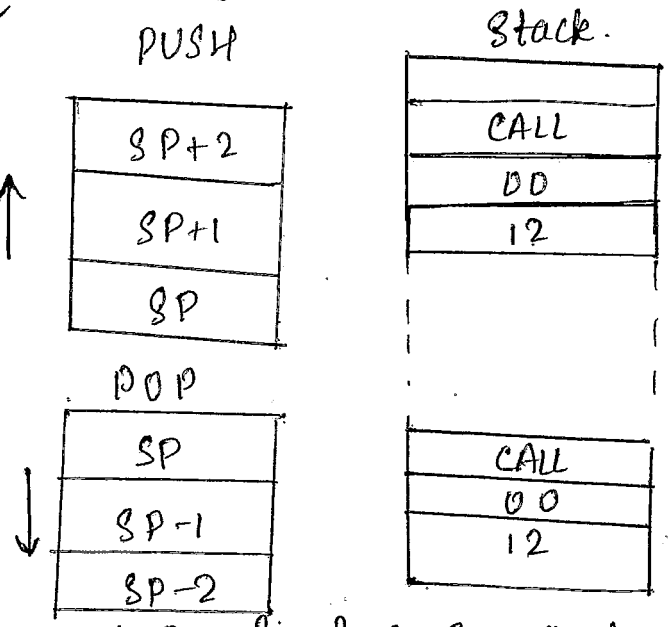
MODULE - 3

Q.NO.

5a) With diagramatical representation explain how stacks plays role in subroutine operations. — 04M

=> Subroutines are often used to perform tasks that need to be performed frequently. This makes a program more structured in addition to saving memory space.

Call instructions is used to call subroutine. ACALL & LCALL when a subroutine is called, control is transferred to that subroutine, the processor saves on stack the address of instruction immediately below the LCALL.



— 02M

Main role stack plays is for saving data when a subroutine is called. For example, assume your main program uses registers R0-R4. You call a subroutine that is going to calculate some value and pass it back to program. After subroutine is called, all the data is pushed onto the stack before executing any instructions in the subroutine. The subroutine can then use all the registers for its internal use and store the data that the main program needs in one memory location. At the end of subroutine, the data is popped off the stack and then return control to main program. So the stack data structure is the most efficient way to store the return address of the ~~sub~~ subroutines.

— 02M

NO

b] Write an assembly language program to sort an array of n=5 bytes of data in ascending order stored from location 30h. (use bubble sort algorithm). — 08M

⇒

```
org     org 0000h
        num equ 040h
back1 : mov r0, #30h    ; store n elements (n=5) from 30h
        mov a, r0       ; r0 and r1 are used as pointers
        mov r1, a
back2 : MOV A, @DPTR
        MOV B, A
        INC DPTR
        MOVX A, @DPTR
        CLRC
        MOV R2, A
        SUBB A, B
        JNC NOEXCHG
        MOV A, B
        MOVX @DPTR, A
        DEC DPL
        MOV A, R2
        MOVX @DPTR, A
        INC DPTR
        DJNZ R1, L2
        DJNZ R0, L1
        SJMP
        ORG 30h
        END.
```

04M

02M

02M

Q.NO. 5 c) Write a assembly language program to sort count the number of 1's and 0's in an 8-bit data received from port P_1. Store the count of 1's and 0's in 30h and 31h. — 08M

⇒

ORG 0000H
SJMP 30H
ORG 30H
MOV

— 02M

ORG 0000H
SJMP 30H
ORG 30H
MOV R4, #08H
MOV R0, #30H
MOV A, @P1

— 04M

LOOP2 : RRC A
JNC LOOP1
INC R2
JMP NEXT
LOOP1 : INC R3
DJNZ R4, LOOP2

— 02M

MOV 30H, R2
MOV 31H, R3
END

OR

Q NO 6 a] Write a note on subroutine instructions.

⇒ Subroutines are often used to perform tasks that need to be performed frequently. This makes a program more structured in addition to saving memory space. — 04M —

Call instructions are used to call subroutine. Such as LCALL and ACALL

LCALL (long call)

- It is a 3-byte instruction
- First byte is a opcode
- Second and third bytes are used for address of target subroutine.

— 02M — ACALL (absolute call)

- 2-byte instruction.
- 11 bits are used for address within 2K byte range.

Subroutine is located anywhere within 64K byte address space. When a subroutine is called, control is transferred to that subroutine, the processor saves on the stack the address of the instruction immediately below LCALL. and begins to fetch instructions from the new location. After finishing execution of the subroutine, the instruction RET transfers control back to the caller. Every subroutine needs RET as the last instruction.

— 02M —

6 b] Write an assembly language program to sort an array of n = 5 byte of data in descending order stored from location 30h. (use bubble sort algorithm). — 08M —

⇒

2)

```
    ORG  0H
    MOV  R0, #05
L1: MOV DPTR, #30h
    MOV A, R0
    MOV R1, A
L2: MOV X A,@DPTR
    MOV B, A
    INC DPTR
    MOV X A, @DPTR
    CLR C
    MOV A, @DPTR
    CLR C
    MOV R2, A
    SUBB A, B
    JC NOEXCHG
    MOV A, B
    MOVX @DPTR, A
    DEC DPL
    MOV A, R2
    MOVX ,@DPTR, A
    INC DPTR
    DJNZ R1, L2      ; NOEXCHG
    DJNZ R0, L1
    SJMP HERE
    ORG 30H
    END.
```

_02M _02M _02M _02M _02M

6 C] Assume a push button switch is connected to port pin P1.2, write an assembly language program to monitor the switch and turn ON the LED's connected to port P_2 as long as the switch is pushed. — 08M —

⇒

~~SETB P1.2; make P1.2 an input~~

~~AGAIN: JB P1.2, OVER; JUMP TURN ON LED when P1.2 = 1~~

~~MOV P2, #~~

```
MOV P0, #3H        ; initialising push button
READSW: MOV A, P0  ; Moving the port value to accumulator
RRC A              ; checking the value of Port 0 to know if
                     SW1 is ON or not.
JC NXT             ; if switch 1 is OFF then jump to NXT to
                     check switch 2 is ON.
CLR P0.7
SJMP READSW        ; Read switch status again.
NXT: RRC A         ; checking the value of Port 0 to know if
                     switch 2 is ON or not.
JC READSW          ; jumping to READSW to check status of
                     switch 1 again
SETB P1.2          ; turn ON LED as switch 2 is ON
SJMP READSW        ; monitor the status
END
```

02M — 02M — 04M

MODULE - 4

Q.NO 7 a] Explain the contents of TCON and TMOD register. — 08M —

⇒ TCON :- Timer / counter Control Register (Bit addressable).

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

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	TCON6	Timer 1 run control bit. set/cleared by software to turn timer / counter ON/OFF
TF0	TCON5	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.
IT0	TCON.0	Interrupt 0 type control bit. set/cleared by software.

— 04M —

TMOD : Timer/Counter Mode Control Register (Not bit addressable)

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

TIMER 1 TIMER 0

GATE — When TRx (in TCON) is a set and GATE=1, TIMER/COUNTERx will run only while INTx pin is high (hardware control) When GATE=0, TIMER/COUNTERx will run only while TRx=1

C/$\overline{T}$ — Timer or counter selector. Cleared for Timer operation (input from internal system clock). Set for counter operation

M1 — Mode selector bit (NOTE 1).

M0 — Mode selector bit (NOTE 1.)

M1	M0	operating mode
0	0	0
0	1	1
1	0	2
1	1	3.

7.b] Write an assembly language program to transfer multi-byte data serially with 9600 baud rate.

⇒ As multibyte is given we can take (say =100 bytes) more than one byte.

⇒

```
MOV TMOD, #20      ; Timer 1, mode 2
MOV TH1, #-3       ; 9600 baud rate
MOV SCON, #50H     ; 8 bit, REN enabled.
SETB TR1           ; start timer 1.
MOV R0, #0H        ; starting address
MOV R1, #100
LABEL: MOV A, @R0  ; reads the value at the address pointed by R0
INC R0             ; Increment
ACALL TRANS
DJNZ R1, LABEL     ; loop runs until R1=0.
TRANS: MOV SUBF, A ; load SUBF
HERE: JNB TI, HERE ; wait for last bit.
CLR TI             ; get ready for next byte
RET
```

-04M-

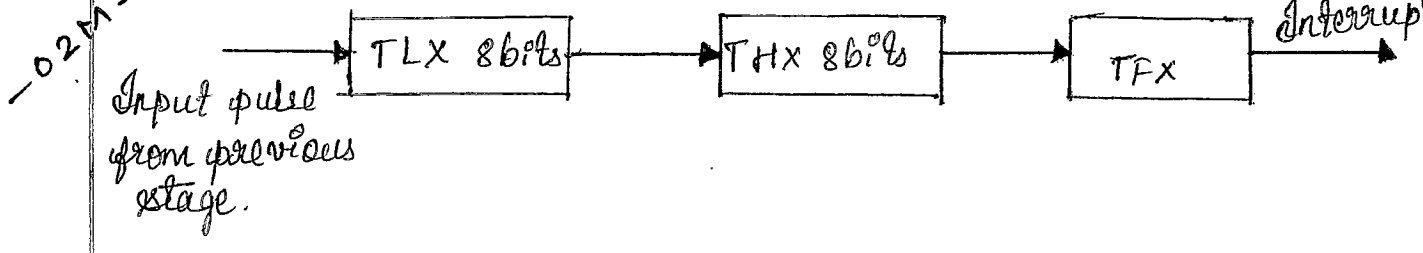
-04M-

7(C) Explain how timers are programed in mode 1. -04M-

⇒ In mode1, the timer is used as a 16-bit UP counter. The lower 8 bits of TLX and 8 bits of THX are used for the 16-bit count. Upper 8 6 bits of TLx are ignored. When the counter rolls over from 0's to all 1's, TFX flag is set and an interrupt is generated. The input pulse is obtained from the previous stage. If TR1/0 bit is 1 and Gate bit is 0, the counter continues counting up. If TR1/0 bit is 1 and gate bit is 1, then the operation of the counter is controlled by input. This mode is useful to measure the width of given pulse fed to input.

-02M-

-02M-



OR

Q.NO 8(a) Explain the bit contents of SCON and PCON registers. — 08M

⇒ SCON : Serial Control.

(value on reset)	0	0	0	0	0	0	0	0
	SM0	SM1	SM2	REN	TB8	RB8	T1	R1
	MSB							LSB
	Bit							

SM0	7	Serial port Mode selection. Bit 0.
SM1	6	Serial port Mode selection. Bit 1.
SM2	5.	Multiprocessor comm. Bit
REN	4.	receive enable bit.
TB8	3.	transmitted bit 8.
RB8	2.	received Bit 8.
TI	1.	transmit interrupt flag.
RI	0.	Receive interrupt flag.

-04M

The SCON SFR is used to control the Microcontroller's serial port. It is located as an address of 98H. Modes of operation can be controlled using SCON.

PCON Power control.

MSB							LSB
SMOD	--	-	-	GF1	GF0	PD	IDL
Bit							

SMOD	7	serial comm. Baud rate. Modify bit.
GF1	3	General purpose user flag, Bit 1
GF0	2.	General purpose user flag. Bit 0.
PD	1.	Power down Bit, to enter power down mode, set to 1
IDL	0	Idle Mode Bit, to enter idle mode, set to 1.

-04M

The PCON register, as the name says, is used to control the 8051 microcontroller's power modes and is located at 87H of the SFR Memory space. Using two bits in the PCON register, the microcontroller can be set to idle mode and Power Down mode.

8 b] Write an assembly language program to generate a square wave on port pin P1.2 of frequency 5KHz. — 08M —

⇒ (a) $T = \frac{1}{5} = 0.2$ ms, the period of square wave. Or 200μs

(b) $\frac{1}{2}$ of it for high and low portion of the pulse is 0.1ms or 100μs — 04M —

(c) $\frac{0.1ms}{1.085\mu s} = \frac{100\mu s}{1.085\mu s} = 92$, and $65536 - 92 = 65443$ in decimal and FFA3 in hex.

Program :-

```
MOV TMOD, #01 ; Timer 0, 16-bit mode
AGAIN: MOV TL1, #1AH ; TL1 = 1A, low byte of time
SETB TR1 ; start timer 1
BACK: JNB TF1, BACK; until time rolls over
CLR P1.2 ; clear timer flag 1.
CLR TF1 ; clear timer 1 flag.
SJMP AGAIN ; Reload timer.
```

— 04M —

8 c] Write a note on Asychronous serial communication and data framing.

⇒ Asynchronous method transfers a single byte at a time. There are special IC chips made by many manufacturers for serial communication. UART (universal asynchronous receiver-transmitter) USART (universal synchronous-asychronous receiver transmitter). — 02M —

Asynchronous serial data communication is widely used for character oriented transmissions. Each character is placed in between start and stop bits, this is called framing. Block-oriented data transfers. — 02M —

MODULE-5

Q.No.

9a] Explain the bit contents of IE register. — 04M —

⇒ IE (Interrupt Enable) register.

D7							D0
EA	--	ET_2	ES	ET1	EX1	ET0	EX0

02M — EA (enable all) must be set to 1 in order for rest of the register to take effect.

EA	IE.7	Disables all interrupts.
--	IE.6	Not implemented, reserved for future use
ET2	IE.5	Enables or disables timer 2 overflow or capture interrupt
ES	IE.4	Enables or disables the serial port interrupt
ET1	IE.3	Enables or disables timer 1 overflow interrupt.
EX1	IE.2	Enables or disables external interrupt 1
ET0	IE1.	Enables or disables timer 0 overflow interrupt.
EX0	IE0	Enables or disables external interrupt 0.

02M

9b] Explain how programming of external hardware interrupts is done in 8051 microcontrollers with a code snippet. — 06M —

⇒ The 8051 has two external hardware interrupts namely, Pin 12 (P3.2) and Pin 13 (P3.3) of the 8051, designed as INT0 and INT1 are used as external hardware interrupts. The interrupt vector table locations 0003H and 0013H are set aside for INT0 and INT1. There are two activation levels for external hardware interrupts. Level triggered and Edge trigged.

03M

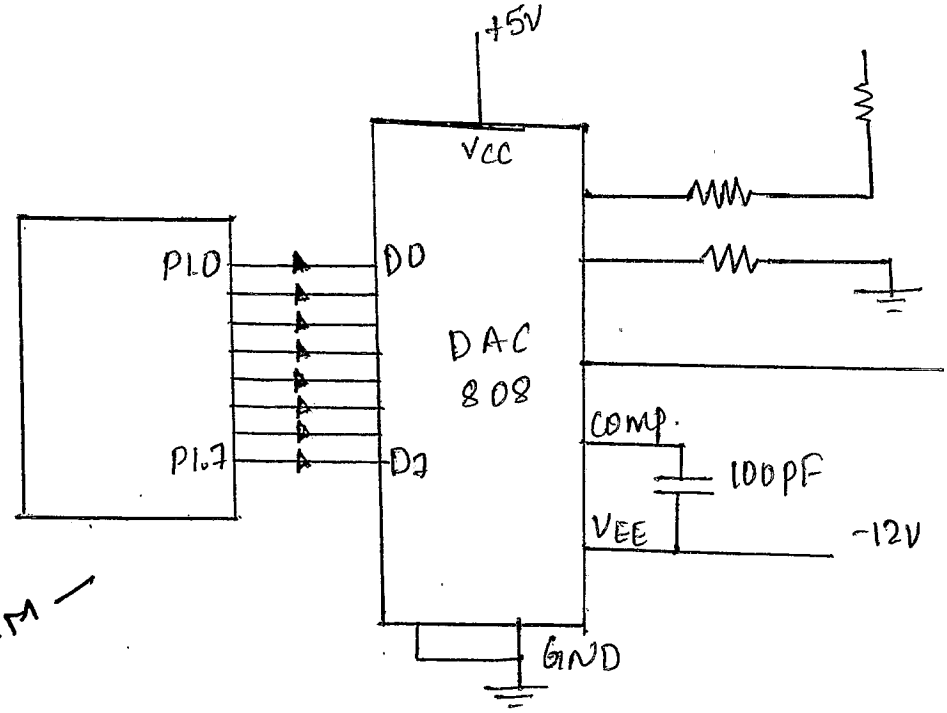
We assume here example of INT1 pin connected to a switch that is normally high. whenever it goes low, it should turn on an LED. The LED is connected to P1.3 and is normally off. When it is turned on it should stay on for a fraction of a second. As long as the switch is pressed low, the LED should stay on.

```
        ORG 0000H
        LJMP MAIN       ; by-pass interrupt vt
        ;vector table
        ; --ISR for INT1 to turn on LED
        ORG 0013H       ; INT1 ISR
        SETB P1.3       ; turn on LED
        MOV R3,#225
BACK:   DJNZ R3,BACK    ; keep LED on for a while
        CLR P1.3        ; turn off the LED
        RETI            ; return from ISR
        ; -- MAIN program for initialization
        ORG 30H
MAIN:   MOV IE,#10000100B ; enable external INT 1
HERE:   SJMP HERE         ; stay here until get interrupted
        END.
```

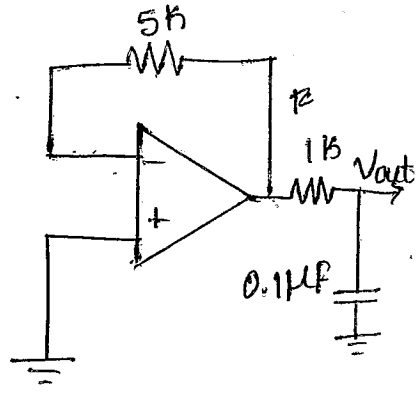
9 c] With neat diagram write an assembly language program to interface DAC to 8051 microcontroller. [10M]

⇒ The DAC is a device widely used to convert digital pulses to analog signals. A DAC consists of a number of binary inputs & a single output. In general, the no. of binary inputs of a DAC will be power of two.

The ~~DAC~~ DAC0808 is a DAC in which the digital inputs are converted to current Iout & by connecting a resistor to Iout pin. we can convert the result to voltage.



02M



Generation of waveforms using DAC0808

Assembly program to generate triangular waveform.

```
    MOV A, #00H
INCR: MOV P1, A
      INC A
      CJNE A, #255, INCR
DECR: MOV P1, A
      DEC A
      CJNE A, #00, DECR
      SJMP INCR
      END
```

04M

Q.No. 10 a]

OR

Explain how interrupt priority can be changed using IP register. Also explain the default priorities assigned to interrupts in 8051 micro controller. — 05M

⇒ We can alter the sequence of interrupt priority by assining a higher priority to any one of the interrupts by programming a register IP (interrupt priority). To give a higher priority to any of the interrupts, we make the corresponding bit in the IP ~~address~~ register high. When two or more interrupt bits in the IP register are set to high. While these interrupts have a higher priority than others, they are serviced according to the sequence.

Priority register table (bit addressable) — 02M

D7							D0
-	-	PT2	PS	PT1	PX1	PT0	PX0

-	IP.7	Reserved
-	IP.6	Reserved
PT_2	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.

— 03M

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

10 b] Write a c program using interrupts to generate a square wave on port pin 1.2 of 1KHz using timer-0 in mode 2. [10M]

⇒ frequency = 1KHz

$1 \text{ pulse} = \frac{1}{1\text{KHz}} = 1\text{ms}$

500μs = ON time, 500μs = OFF time

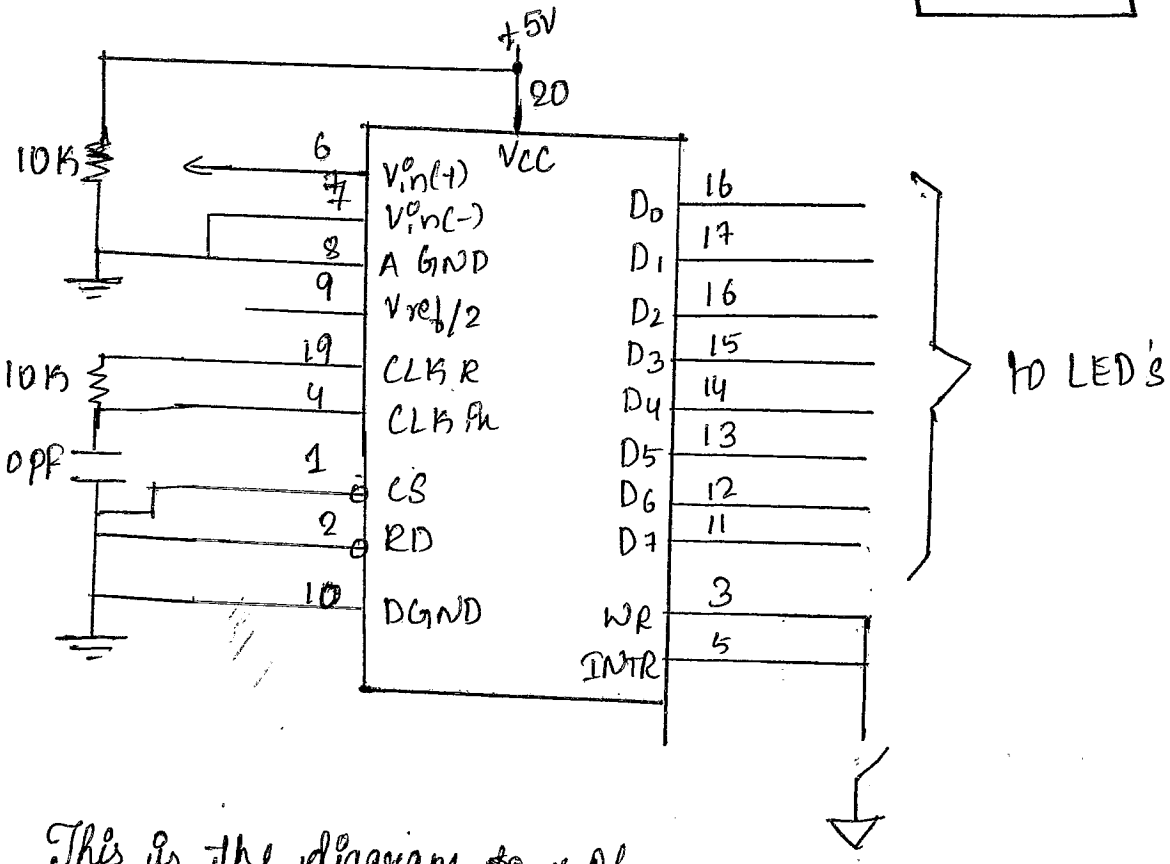
$\text{Count} = \frac{500\mu s}{1\mu s} = 500$

Count initial value = 65536 − 500 = 65036 (decimal) = ⊕
= FE0C (Hex).

```
#include <reg51.h>
 sbit WAVE = P1^2;
 void timer0(void) interrupt 1
{
    WAVE = ~WAVE; // toggle pin
}
 void main()
  {
    SW = 1;  //make switch input.
    TMOD  = 0X02;
    TH0   = 0XA4 ; //TH0 = -92
    IE =0X82 ;  //enable interrupt for timer 0
    while(1)
   {
     IND = SW;
   }
}
```

10 c] With neat diagram, write an assembly language program to interface ADC-0804 to 8051 microcontroller. — 10M

⇒



This is the diagram of ADC-0804. This is also connected to the self clocking with RC component for frequency and the external connected to XTAL2 of 8051.

Assembly program to interface ADC-0804 with self clocking from XTAL2 from of 8051.

```
MY DATA EQU P1
MOV P1, #0FFH SETB P2.7
BACK: CLR P2.6
SETB P2.6
HERE: JB P2.7, HERE
CLR P2.5
MOV A, MYDATA
SETB P2.5
SJMP BACK
END
```