CMR
INSTITUTE OF
TECHNOLOGY

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

Internal Assessment Test - 2

Sub:	**Microcontrollers**							Code:	18EE52
Date:	17/12/2021	Duration:	90 mins	Max Marks:	50	Sem:	5th	Branch:	EEE

Answer **Any FIVE FULL** Questions

		Marks	OBE	
			CO	RBT
1	Explain different jump and call instructions of 8051 microcontroller with their jump ranges with the help of figure	10	CO2	L2
2	Write an ALP of 8051 to read the content of Port 0 and send it to Port 1 after inversion, continuously.	10	CO2	L4
3	Explain the functions TMOD & TCON register with its bit pattern.	10	CO3	L1
4	Write a C program to convert ASCII digits ‘2’ and ‘7’ to packed BCD and display it on Port1.	10	CO3	L4
5	Write a C Program to bring in a byte of data serially one bit at a time via P1.0. The LSB should come first.	10	CO3	L4
6	Explain with suitable example, the various bit level instructions available in 8051in detail.	10	CO2	L4
7	Write an ALP to find the average of ten 8 bit numbers stored in internal RAM location starting from 30 H to onwards store result at 60 H.	10	CO2	L3
8	a) Explain the difference between sbit, sfr and bit declarations. b) Write the steps involved in programming Mode1 of 8051 Timer.	5 5	CO2 CO3	L4 L2

Signature of CCI

Signature of CI

1. Explain different jump and call instructions of 8051 microcontroller with their jump ranges with the help of figure.

   There are two unconditional jumps in 8051 : LJMP and SJMP

   LJMP (long jump) − LJMP is 3-byte instruction in which the first byte represents opcode, and the second and third bytes represent the 16-bit address of the target location. The 2-byte target address is to allow a jump to any memory location from 0000 to FFFFH.

   SJMP (short jump) − It is a 2-byte instruction where the first byte is the opcode and the second byte is the relative address of the target location. The relative address ranges from 00H to FFH which is divided into forward and backward jumps; that is, within –128 to +127 bytes of memory relative to the address of the current PC (program counter). In case of forward jump, the target address can be within a space of 127 bytes from the current PC. In case of backward jump, the target address can be within –128 bytes from the current PC.

   CALL Instructions
   CALL is used to call a subroutine or method. Subroutines are used to perform operations or tasks that need to be performed frequently. This makes a program more structured and saves memory space. There are two instructions − LCALL and ACALL.

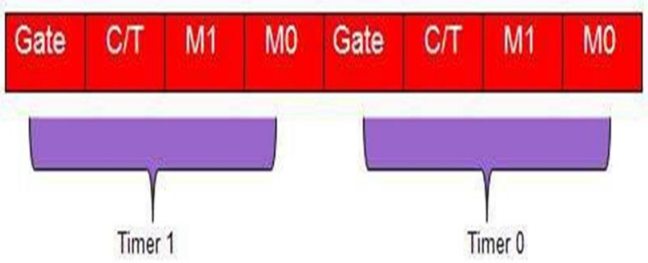
   LCALL (Long Call)
   LCALL is a 3-byte instruction where the first byte represents the opcode and the second and third bytes are used to provide the address of the target subroutine. LCALL can be used to call subroutines which are available within the 64K-byte address space of the 8051.
   To make a successful return to the point after execution of the called subroutine, the CPU saves the address of the instruction immediately below the LCALL on the stack. Thus, when a subroutine is called, the control is transferred to that subroutine, and the processor saves the PC (program counter) on the stack and begins to fetch instructions from the new location. The instruction RET (return) transfers the control back to the caller after finishing execution of the subroutine. Every subroutine uses RET as the last instruction.
   ACALL (Absolute Call)
   ACALL is a 2-byte instruction, in contrast to LCALL which is 3 bytes. The target address of the subroutine must be within 2K bytes because only 11 bits of the 2 bytes are used for address. The difference between the ACALL and LCALL is that the target address for LCALL can be anywhere within the 64K-bytes address space of the 8051, while the target address of CALL is within a 2K-byte range.

2. Write an ALP of 8051 to read the content of Port 0 and send it to Port 1 after inversion, continuously.

```
        ORG 0000H
        MOV A, #0FFH
        MOV P0, A
BACK:   MOV A,P0
        CPL A
        MOV P1,A
        SJMP BACK
        END
```

3. Explain the functions TMOD & TCON register with its bit pattern.

TMOD (timer mode) Register:
Both timers 0 and 1 use this same register to set the various timer operation modes
TMOD is an 8-bit register. The lower 4 bits are for Timer 0 and the upper 4 bits are for Timer 1.



M1	M0	Mode
0	0	0 (13 bit Timer Mode)
0	1	1 (16 bit Timer Mode)
1	0	2 (8 bit auto reload)
1	1	3 (split Timer Mode)

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.

TCON (Timer Control Register)

This register is used to initialize the counting and to indicate when the timers have reached their counting limit.TCON stands for timer control and is used to send the control signals for the functioning of the timer. The TCON register is bit addressable and is places at the address 88H in the ROM. It is an 8-bit register that starts the timer(TR) and also contains the flag which gets updated when the timer overflows(TF).

4. Write a C program to convert ASCII digits '2' and '7' to packed BCD and display it on Port1.

```
#include<reg51.h>
void main()
{       unsigned char myBCD;
        unsigned char x = '2',y='7';
        x = x & 0x0F; //mask lower 4 bits
       x = x <<4;  //shift 4 left
       y = y & 0x0F;
myBCD = x | y;  //combine to make BCD
        P1 = myBCD;
}// end of main
```

5. Write a C Program to bring in a byte of data serially one bit at a time via P1.0. The LSB should come first.

```
#include <reg51.h>
sbit P1b0=P1^0;
sbitregALSB=ACC^0;
void main(void)
{
unsigned char conbyte=0x44;
```

```
unsigned char x;
ACC=conbyte;
for (x=0;x<8;x++)
{
P1b0=regALSB;
ACC=ACC>>1;
}
}
```

6. Explain with suitable example, the various bit level instructions available in 8051in detail.

Mnemonic	Operation
ANL C,b	AND C and the addressed bit; put the result in C
ANL C,/b	AND C and the complement of the addressed bit; put the result in C; the addressed bit is not altered.
ORL C,b	OR C and the addressed bit; put the result in C
ORLC,/b	OR C and the complement of the addressed bit; put the result in C; the addressed bit is not altered
CPLC	Complement the C flag
CPL b	Complement the addressed bit
CLRC	Clear the C flag to zero
CLR b	Clear the addressed bit to zero
MOV C,b	Copy the addressed bit to the C flag
MOV b,C	Copy the C flag to the addressed bit
SETB C	Set the flag to one
SETB b	Set the addressed bit to one

7. Write an ALP to find the average of ten 8 bit numbers stored in internal RAM location starting from 30 H to onwards store result at 60 H.

```
        ORG 0000H
        MOV R1, #0A      ;R1 stores the count of total 8 bit numbers
        MOV B, #0A       ;B is used as divisor for average
        MOV R0, #30H     ;R0 acts as pointer to the data
        MOV A, #00H      ;Clear A
BACK :  ADD A, @R0       ; Add the data to A register
        INC R0           ; Increment the pointer
        DJNZ R1,BACK     ; repeat addition until R1=0
        DIV AB           ; divide sum to get average
                         ; quotient is in A and remainder in B
        MOV 60H,A        ;Ignore remainder and store average result in 60h
HERE:   SJMP HERE        ; wait
        END
```

8. (a)Explain the difference between sbit, sfr and bit declarations. (5)

sbit (Single bit):

The 8 bit keyword is a widely used 8051 C data types which is used to access single-bit addressable register.

It allows access to the single bits of the SFR registers.

Among the SFRs that are widely used are also bit addressable ports P0-P3.

bit and sfr:

The bit data type allows access to single bits of bit-addressable memory spaces 20 – 2FH.

To access the byte-size SFR registers, the sfr data type can used.

(b)Write the steps involved in programming Mode1 of 8051 Timer.(5)

1. To generate a time delay, using timer in mode 1, following are the steps:
2. Load the TMOD value register indicating which timer (timer 0 or timer 1) is to be used and which timer mode (0 or 1) is selected.
3. Load registers TL and TH with initial count value.
4. Start the timer.

5. Keep monitoring the timer flag (TF) with the JNB TFx, target instruction to see if it is raised. Get out of the loop when TF becomes high.
6. Stop the timer.
7. Clear the TF flag for the next round.
8. Go back to Step 2 to load TH and TL again.