Reg. No.									

MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

(A constituent unit of MAHE, Manipal)

IV SEMESTER B.TECH (ELECTRICAL & ELECTRONICS ENGINEERING)

GRADE IMPROVEMENT EXAMINATIONS, AUGUST 2021

MICROCONTROLLERS [ELE 2254]

REVISED CREDIT SYSTEM

Time: 2 Hours **Date: 9 August 2021** **Max. Marks: 40**

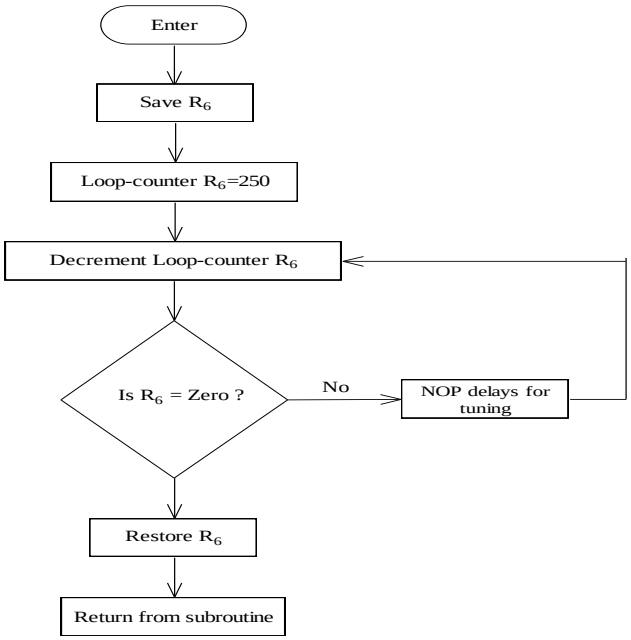
Instructions to Candidates:

- Answer **any four** full questions.
- Missing data may be suitably assumed.

1A. Discuss how External Access (EA) is used to select between internal and external program memory. ***(04)***

1B. Write an 8051 ALP to find the address of given byte '64H' in an array of numbers stored in external RAM locations starting at address 2018H. Size of the Array is store in memory location 2017H. If the number 64H is found in the array, then display the address of first occurrence in P0 and P1 respectively and store the count of such occurrences in memory location 2000H. If the number 64H is not found in the array, then display AAH in both P0 and P1 as an error code. Also include program comments appropriately. ***(06)***

2A. Develop an 8051 instructions to realise the flow chart given. Add required number of NOP instructions to generate a delay of 1 millisecond. Substantiate your answer with proper justification. Assume crystal frequency is 12 MHz



(05)

2B. 8051 microcontroller-based fire alarm system clocked at 12MHz is connected to a loud speaker through transistor based amplified and is controller using a switch as shown in **Figure 2B**. Write an 8051 ALP to sound the loud speaker at a frequency of 500Hz when the switch is turned ON. Use Timer 1 in mode 2 to obtain the required delay. ***(05)***

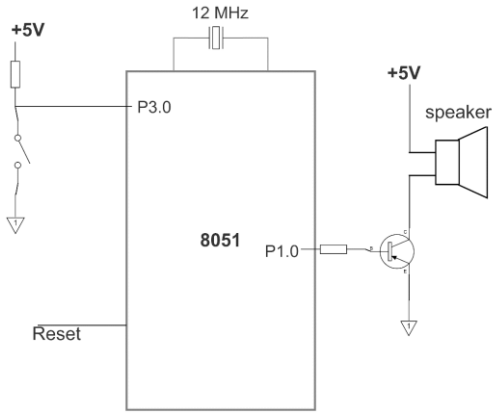


Figure 2B

3A. Develop a system with 8051 to interface 4 LEDs and 4 switches. Write an ALP such that operation of each switch would turn on only one corresponding LED. Support the program with appropriate comments ***(05)***

3B. As part of an industrial automation system two wheels are driven by two separate motors, motor A and motor B. The rotation sensors give a logic low level as the wheel magnet passes the sensor. Each motor can be turned on or off by providing a logic signal as indicated in **Figure 3B**. An 8051 is to be used to control these motors, where a motor can be turned ON and allowed run for N rotations and then turned OFF. The sensor signals will cause timer/counter interrupts. Write an 8051-assembly language program which will turn on the two motors at the same time. Motor A will do 20 rotations and will then be stopped. Motor B will do 200 rotations and will then be stopped.

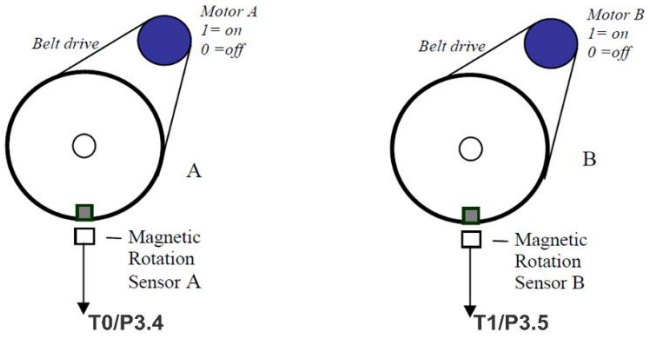


Figure 3B. ***(05)***

4A. Develop a 8051 ALP program to transfer block of 100 data bytes from external memory location 7000H onwards to external memory location 8000 H onwards. If any of those data bytes are 00H then it needs to be changed to FFH and stored in corresponding locations of 8000H onwards. Support your program with comments ***(05)***

4B. A water-level sensor placed in an overhead tank provides analog signals of the current water level to 3rd input channel of an analog to digital converter interfaced with 8051 micro-controller. Draw the interfacing circuit and write an assembly language program to interface an analog to digital converter (ADC 0808) to 8051 and obtain the equivalent digital value in port 1 continuously. Assume Port 0 of 8051 is connected to ADC 0808. ***(05)***

5A. Design a system which contains a 16-key matrix keyboard and 8 LEDs interfaced with 8051. Develop a program to detect the key press (key closure) and key identification. The binary code of the pressed key should be displayed on LEDs. Use P1(rows) and P2(columns) pins for interfacing. ***(06)***

5B. Show the interfacing diagram of 16X2 LCD with the 8051. Write steps to initialize the LCD and instructions executed by the LCD during internal intialisation. ***(04)***

6A. Interface the DAC 0808 with 8051and write a program to generate a sine wave. ***(04)***

6B. The **Figure 6B** shows a D/A (digital to analog) converter which drives a dc motor. Assume the power stage is included in the D/A converter so that the motor can be driven directly by Vout. An 8051 microcontroller has this D/A converter connected to an I/O port P1. The 8051 is connected to a Computer via the serial port. When 8-bit data character is received from the Computer, it interrupts the 8051 through the 8051's serial port interrupt and the received 8-bit data value is passed to the D/A converter so that the Computer is effectively controlling the motor speed. Write an 8051-assembly language program to implement this system. Assume a serial data baud rate of 9,600 baud.

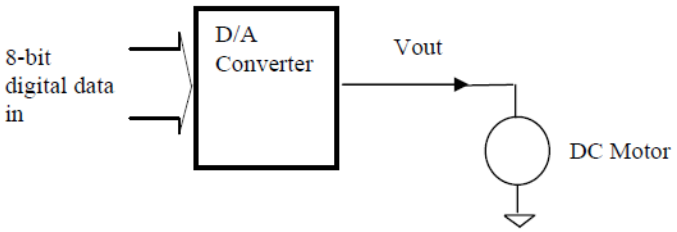


Figure 6B ***(06)***

Arithmetic Instructions			Logical Instructions		
CODE	**DESCRIPTION**	**MC**	**CODE**	**DESCRIPTION**	**MC**
ADD A, #n	A← A + n	1	ANL A, #n	A← A AND n	1
ADD A, Rr	A ← A + Rr	1	ANL A, Rr	A ← A AND Rr	1
ADD A, addr	A ← A + [addr]	1	ANL A, addr	A ← A AND [addr]	1
ADD A, @Rp	A ← A + [Rp]	1	ANL A, @Rp	A ← A AND [Rp]	1
ADDC A, #n	A← A + n + C	1	ANL addr, A	[addr] ← [addr] AND A	1
ADDC A, Rr	A ← A + Rr + C	1	ANL addr, #n	[addr] ← [addr] AND n	2
ADDC A, addr	A ← A + [addr] + C	1	ORL A, #n	A← A OR n	1
ADDC A, @Rp	A ← A + [Rp] + C	1	ORL A, Rr	A ← A OR Rr	1
SUBB A, #n	A← A - n - C	1	ORL A, addr	A ← A OR [addr]	1
SUBB A, Rr	A ← A - Rr - C	1	ORL A, @Rp	A ← A OR [Rp]	1
SUBB A, addr	A ← A - [addr] - C	1	ORL addr, A	[addr] ← [addr] OR A	1
SUBB A, @Rp	A ← A - [Rp] - C	1	ORL addr, #n	[addr] ← [addr] OR n	2
INC A	A ← A + 1	1	XRL A, #n	A← A X-OR n	1
INC Rr	Rr ← Rr + 1	1	XRL A, Rr	A ← A X-OR Rr	1
INC addr	[addr] ← [addr] + 1	1	XRL A, addr	A ← A X-OR [addr]	1
INC @Rp	[Rp] ← [Rp] + 1	1	XRL A, @Rp	A ← A X-OR [Rp]	1
INC DPTR	DPTR ← DPTR + 1	2	XRL addr, A	[addr] ← [addr] X-OR A	1
DEC A	A ← A - 1	1	XRL addr, #n	[addr] ← [addr] X-OR n	2
DEC Rr	Rr ← Rr -1	1	CLR A	A ← 00	1
DEC addr	[addr] ← [addr] - 1	1	CPL A	A← 1's compliment of A	1
DEC @Rp	[Rp] ← [Rp] - 1	1	RL A	$A_0 \leftarrow A_7 \leftarrow A_6 \ldots \leftarrow A_0$	1
MUL AB	BA ← A X B (B – Higher)	4	RLC A	$C \leftarrow A_7 \leftarrow A_6 \ldots \leftarrow A_0 \leftarrow C$	1
DIV AB	BA ← A / B (B – Remainder)	4	RR A	$A_0 \rightarrow A_7 \rightarrow A_6 \ldots \rightarrow A_0$	1
DA A	$A_{DEC} \leftarrow A_{BIN}$	1	RRC A	$C \rightarrow A_7 \rightarrow A_6 \ldots \rightarrow A_0 \rightarrow C$	1
			SWAP A	$A_{LN} \leftarrow\rightarrow A_{HN}$	1
			NOP	PC ← PC + 1	1

DATA TRANSFER INSTRUCTIONS			BRANCH INSTRUCTIONS		
CODE	**DESCRIPTION**	**MC**	**CODE**	**DESCRIPTION**	**MC**
MOV A, #n	A ← n	1	ACALL sadd	[SP] ← PC + 2; PC ← sadd	2
MOV A, Rr	A ← Rr	1	LCALL ladd	[SP] ← PC + 3; PC ← ladd	2
MOV A, addr	A ← [addr]	1	CJNE A, add, radd	If (A ≠ [add]); PC ← PC + 3 + radd	2
MOV A, @Rp	A ← [Rp]	1	CJNE A, #n, radd	If (A ≠ n); PC ← PC + 3 + radd	2
MOV Rr, A	Rr ← A	1	CJNE Rr, #n, radd	If (Rr ≠ n); PC ← PC + 3 + radd	2
MOV Rr, #n	Rr ← n	1	CJNE @Rp, #n, radd	If ([Rp] ≠ n); PC ← PC + 3 + radd	2
MOV Rr, addr	Rr ← [addr]	2	DJNZ Rr, radd	If (Rr-1 ≠ 0); PC ← PC + 2 + radd	2
MOV addr, A	[addr] ← A	2	DJNZ add, radd	If ([add]-1 ≠ 0); PC ← PC + 3 + radd	2
MOV addr, #n	[addr] ← n	2	AJMP sadd	PC ← sadd	2
MOV addr, Rr	[addr] ← Rr	2	LJMP ladd	PC ← ladd	2
MOV addr1, addr2	[addr1] ←[addr2]	2	SLMP radd	PC ← PC + 2 + radd	2
MOV addr, @Rp	[addr] ←[Rp]	1	JMP @A+DPTR	PC ← A + DPTR	2
MOV @Rp, A	[Rp] ← A	1	JC radd	If (C=1); PC ← PC + 2 + radd	2
MOV @Rp, #n	[Rp] ← n	1	JNC radd	If (C=0); PC ← PC + 2 + radd	2
MOV @Rp, addr	[Rp] ← [addr]	2	JB b, radd	If (b=1); PC ← PC + 3 + radd	2
MOV DPTR, #nn	DPTR ← nn	2	JNB b, radd	If (b=0); PC ← PC + 3 + radd	2

CODE	DESCRIPTION	MC	CODE	DESCRIPTION	MC
MOVX A, @Rp	A ← [Rp]^	2	JBC b, radd	If (b=1); PC ← PC + 3 + radd; b←0	2
MOVX A, @DPTR	A ← [DPTR]^	2	JZ radd	If (A=00); PC ← PC + 2 + radd	2
MOVX @Rp, A	[Rp]^ ← A	2	JNZ radd	If (A>00); PC ← PC + 2 + radd	2
MOVX @DPTR, A	[DPTR]^ ← A	2	RET	PC ← [SP]	2
MOVC A, @A+DPTR	A ← [A+DPTR]	2	RETI	PC ← [SP]; Enable Interrupts	2
MOVC A, @A+PC	A ← [A+PC]	2	**BOOLEAN INSTUCTIONS**		
PUSH addr	SP ← SP + 1; [SP] ← [addr]	2	CLR C	C←0	1
POP addr	[addr] ← [SP]; SP ← SP - 1	2	CPL C	C← 1's compliment of C	1
XCH A, Rr	A ← → Rr	1	SETB C	C ← 1	1
XCH A, addr	A ← → [addr]	1	CLR b	b←0	1
XCH A, @Rp	A ← → [Rp]	1	CPL b	b← 1's compliment of b	1
XCHD A, @Rp	ALN ← → [Rp]LN	1	SETB b	b ← 1	1
			MOV C, b	C ← b	2
			MOV b, C	b ← C	2
			ANL C, b	C ← C AND b	2
			ORL C, b	C ← C OR b	2

PIN DIAGRAM OF 8051