Ref. No.: Ex/IEE/PC/B/T/314/2024(S)

BIEE 3rd Year 1st Semester Examination

Subject: Microcontrollers

Full Marks: 100

Time: 3 hours

Sl. No	Answer all questions Assume that the 8051 microcontroller uses a 12 MHz crystal	Marks
	Unit-1	
1	With examples explain the bit-wise implementation of the PSW register.	10
2	Briefly describe the on-chip memory organization of the 8051 microcontroller.	10
3	With examples explain the implementation of the SJMP and LJMP instructions.	10
	Unit-2	
4	Write an assembly language program that adds the two unsigned numbers stored in R0 and R1, computes the integer average of the sum and stores it into R0.	10
5	R0-R1 contains a 2-byte unsigned integer. R2 contain another unsigned integer. Write an assembly language program that multiplies R0-R1 by R2 and stores the result in R0-R1-R2.	10
6	R0-R1 and R2-R3 contain two 2-byte unsigned integers. Write an assembly language program that sets/resets the CY flag if the content of R0-R1 is greater than/lesser than or equal to that of R2-R3.	10
	Unit-3	
7	Explain the bit-wise implementation of the TMOD register.	5
	Explain the working of the timers in auto-reload mode. What is its advantage over mode-1 operation?	5
	Write a stretch of code that uses timer-0 overflow interrupt to generates a 5 kHz clock at P1.0	5
8	Explain the bit-wise implementation and functioning of the IP and IE registers.	6
	How does the microcontroller handle concurrent interrupt requests?	4
9	Explain the bit-wise implementation of the SCON register.	5
	Derive the working formula for calculating the baud rate in mode-1 and mode-3 of UART operation.	3
	Write a stretch of code that configures the UART in mode-1, sets the SMOD bit in PCON register and configures timer-1 for a baud rate of 9600.	5
	Calculate the %error in the baud rate setting	2

80C51 Instruction Set

Arithmatic operations

mnemonic	byte	m/c cycle
ADD A, Rn/@Ri	1	1
ADD A, direct,#data	2	1
ADDC A, Rn/@Ri	1	1
ADDC A, direct/#data	2	1
SUBB A, Rn	1	1
SUBB A, direct	2	1
SUBB A, @Ri	1	1
SUBB A, #data	2	1
INC A/Rn/@Ri	1	1
INC direct	2	1
DEC A/Rn/@Ri	1	1
DEC direct	2	1
INC DPTR	1	2
MUL AB	1	4
DIV AB	1	4
DA A	1	1

Logical operations

mnemonic	byte	cycle
ANL A, Rn/@Ri	1	1
ANL A, direct/#data	1	1
ANL A, #data	2	1
ANL direct, A	2	1
ANL direct, #data	3	2
ORL A, Rn/@Ri	1	1
ORL A, direct/#data	2	1
ORL direct, A	2	1
ORL direct, #data	3	2
XRL A, Rn/@@Ri	1	1
XRL A, direct/#data	2	1
XRL direct, A	2	1
XRL direct, #data	3	2
RL A	1	1
RLC A	1	1
RR A	1	1
RRC A	1	1
SWAP A	1	1

Program branching

mnemonic	byte	m/c cycle
ACALL addr11	2	2
LCALL addr16	3	2
RET	1	2
RETI	1	2
AJMP addr11	2	2
LJMP addr16	3	2
SJMP add8	2	2
JZ/JNZ rel	2	2
CJNE A, direct, rel	3	2
CJNE A, #data, rel	3	2
CJNE Rn, #data, rel	3	2
CJNE @Ri, #data, rel	3	2
DJNZ Rn, rel	3	2
DJNZ direct, rel	3	2
NOP	1	1

Data transfer

mnemonic	byte	m/c cycle
MOV A, Rn/@Ri	1	1
MOV A, direct/#data	2	1
MOV Rn, A	1	1
MOV Rn, direct	2	2
MOV Rn, #data	2	1
MOV direct, A	2	1
MOV direct, Rn	2	2
MOV direct, direct	3	2
MOV direct, @Ri	2	2
MOV direct, #data	3	2
MOV @Ri, A	1	1
MOV @Ri, direct	2	2
MOV @Ri, #data	2	1
PUSH direct	2	2
POP direct	2	2
XCH A, Rn	1	1
XCH A, @Ri	1	1
XCHD A, @Ri	1	1

Boolean variable manipulation

mnemonic	byte	m/c cycle
CLR C	1	1
CLR bit	2	1
SETB C	1	1
SETB bit	2	1
CPL C	1	1
CPL bit	2	1
ANL C, bit	2	2
ANL C, /bit	2	2
ORL C, bit	2	2
ORL C, /bit	2	2
MOV C, bit	2	1
MOV bit, C	2	2
JC/JNC rel	2	2
JB/JNB bit, rel	3	2
JBC bit, rel	3	2