

- Please maintain academic integrity.
- Show your work to get credits and state any assumptions you make.

1. [14 marks] The parts relate to 8051, but are independent.

- [2 marks] Give code to swap contents of registers R0 and R1. For full credit, do it in least number of cycles.
- [1 mark] If a '1' is written to a pin of Port 0 in 8051 (with no external resistors) the output will still be '0'. True or False. Justify.
- [1 mark] Is, MOV R1, R2; a valid 8051 instruction ? Yes or No. Justify.
- [1 mark] What is the maximum number of clock-cycles taken by an instruction in 8051 ?
- [1 mark] Which register(s) in 8051 always stores an address ?
- [3 marks] What is the role of these signals in 8051 ?
 - ALE
 - EA
 - PSEN
- [1 mark] Give an equivalent instruction to achieve the following in fewer cycles.

MOV OEOH, #50H

- (h) [1 mark] Consider that the following instructions are executed in 8051. The accumulator will have the content of which memory location ?

SETB RS1
SETB RSO
MOV A, R7

- (i) [1 mark] What does the following instruction do ?

MOV C, 6

- (j) [1 mark] When using AJMP instruction, what limits the destination location ? How far (in terms of memory locations) can the execution jump ?

- (k) [1 mark] What is the minimum interrupt latency in 8051, which uses a 12 MHz clock ? Justify.

2. [4 marks] Write code to add two 4 digit BCD numbers. The first number is stored in 40H and 41H. The second number is in 42H and 43H. The higher order bits are in 41H and 43H, respectively. Store the BCD result in 40H and 41H, with higher order bits in 41H.

3. [3 marks] Consider the 8051 assembly program shown below.

```
FUN: MOV R0,#50H
LOOP: MOV @R0,#0
      INC R0
      CJNE R0,#80H,LOOP
      END
```

- [1 mark] What is the purpose of the subroutine FUN ?
- [2 marks] If you consider a clock frequency of 12 MHz, what is the execution time of the subroutine ?

4. [5 marks] Consider a person detector sensor that generates a signal transitions from 1-to-0 on detecting a person. This sensor is to be connected to the 8051, so that you can use Timer 1 to check if 1000 people have entered a room. Once 1000 people enter, a LED connected to port P1.7 is to be turned ON.
- [1 mark] Provide a brief description of the mode and connections that are needed to do this.
 - [4 marks] Provide your 8051 code to realize this aspect of the application, with necessary comments.
5. [5 marks] We are interested in measuring the pulse width (duration) of an external signal. How do we achieve this using the Timer-1 in 8051 ?
- [1 mark] Mention which pin(s) are to be used to achieve this.
 - [3 marks] Provide lines of code for appropriate configurations and mode settings
 - [1 mark] Is there a limit on the pulse width that can be measured ? Justify.
6. [2 marks] This is about using serial communication ports in Mode 0.
- [1 mark] What do we gain by using the Serial Ports (Rxd, Txd) in this mode ?
 - [1 mark] What is the disadvantage of using these pins for data transfer, when compared to use of other port pins (e.g., Port 1) ?

7. [7 marks] We are interested in transmitting a character using the serial port.

- [3 marks] Consider the program MAIN which uses subroutines INIT and OUTCHAR to use the serial port of 8051 as an 8-bit UART at 2400 baud rate to transmit a character. The CPU operates using a 12MHz clock and it is to output an ASCII character (7-bits and 1-bit to indicate parity) on the serial port. Complete the code to achieve this functionality with relevant comments. If needed add other missing line(s) of code.

Q.1 Q.2.
 (1) (1)

```

INIT: MOV SCON, #40H; (mode 1 ; #50 also is fine)
      MOV TMOD, #20H; ( auto reload)
      MOV TL1, #f3H; ( or #00H)
      MOV TH1, #f3H; ( baw 243 → baud rat 2400)
      SETB TRI; ; ( start baud rate)
      RET
  
```

OUTCHAR: CLR T1; → MOV SBUF, A.

 HERE: JNB T1, HERE; # wait for A to tx.

 CLR T1;

 MOV X, X

 RET

MAIN: LCALL INIT

 MOV A, #'X'

 LCALL OUTCHAR

 END

with ISR.

INIT: MOV SCON, #40H

 MOV TMOD, #20H

 MOV TL1, #f3H

 MOV TH1, #f3H

 SETB TRI

 MOV IE, #40H

 RET

SERIAL-ISR:

CLEAR

 CLR S T1;

 RETI

HAIN: LLALLINIT

 MOV SBUF, #X

 DONE: LJMP , DONE

- [3 marks] You are to realize the same functionality as in previous part, using interrupt driven serial communication. The vector address for serial interrupt is 0023 H. Provide the INIT and ISR code to achieve this.

- [1 mark] Which of the above approaches (part (a) or part(b)) is preferable if we require continuous transmission ? Why ?

for cont.

 MAIN: LCALL INIT

 RETI: MOV SBUF, A

 JNB 2 of 2

Port P3 of 8051 is a multi-function port.

Additional functions of Port 3 lines

Port Line	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
Function	RD	WR	T1 in	T0 in	INT1	INT0	TxD	RxD

Lines P3.5 and P3.4 can be used as inputs to Timers T1 and T0 respectively.

TCON register at BYTE address 88H								
Bit No.	7	6	5	4	3	2	1	0
Bit Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Bit Addr	8F	8E	8D	8C	8B	8A	89	88
TMOD register at BYTE address 89H								
Bit No.	7	6	5	4	3	2	1	0
Timer:	T1				T0			
Bit Name	G1	C/T1	T1M1	T1M0	G0	C/T0	T0M1	T0M0

Reg. Name	Address	Function
TCON	88H	Timer status
TMOD	89H	Timer modes and Config
TL0	8AH	T0 count register: Low
TL1	8BH	T1 count register: Low
TH0	8CH	T0 count register: High
TH1	8DH	T1 count register: High

SFR IE at byte address A8H

Bit No.	7	6	5	4	3	2	1	0
Bit Addr	AF	AE	AD	AC	AB	AA	A9	A8
Bit Name	EA	-	-	ES	ET1	EX1	ET0	EX0
Interrupt on	IE	U	U	SI	TF1	Ex1	TF0	Ex0

SFR IP at byte address B8H

Bit No.	7	6	5	4	3	2	1	0
Bit Addr	BF	BE	BD	BC	BB	BA	B9	B8
Bit Name	U	U	U	PS	PT1	PX1	PT0	PX0

Note: The IP bits are in the polling order of interrupts.

SFR SCON at byte address 98H

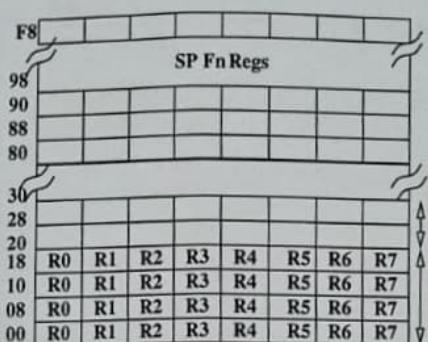
Bit No.	7	6	5	4	3	2	1	0
Bit Addr	9F	9E	9D	9C	9B	9A	99	98
Bit Name	SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SM1	Serial Mode
0	0	0
0	1	1
1	0	2
1	1	3

PSW is located at the address D0H in SFR memory area.

Bit No.	7	6	5	4	3	2	1	0
Bit Name	CY	AC	F0	RS1	RS0	OV	F1	P

SFR range accessed only by direct addressing



8052 upper memory range accessed only by indirect addressing.



Bit Addressable

Register Banks

(Active bank selected by bits in PSW)

90

Figure 1: Layout of internal RAM of 8051

F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8									CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87

Figure 2: Layout of SFRs in 8051

- Please maintain academic integrity.
- Show your work to get credits and state any assumptions you make.

1. [14 marks] The parts relate to 8051, but are independent.

(a) [2 marks] Give code to swap contents of registers R0 and R1. For full credit, do it in least number of cycles.

Solution:

```
MOV A, R0
XCH A, R1
MOV R0, A
```

```
MOV A, R1
MOV B, R0
MOV R1, B
MOV R0, A
```

(or)

```
MOV A, R1
MOV R1, OOH
MOV R0, A
```

Possible changes: This is more or less it. A can be replaced with B.

There is another solution in which you can do it by adding and subtracting ($a = a+b$; $b = a-b$; $a = a-b$;) but it will be too tedious in 8051 and I assume nobody would have done it, and if they have check the code and give marks accordingly)

(b) [1 mark] If a '1' is written to a pin of Port 0 in 8051 (with no external resistors) the output will still be '0'. True or False. Justify.

Solution: True. Port 0 has no Internal pull-up resistors so we can not have 1 as an output. It requires an external resistor to operate it as a normal IO pin.

(c) [1 mark] Is, MOV R1, R2; a valid 8051 instruction ? Yes or No. Justify.

Solution: No. MOV cannot be used to move directly between registers using their names.

(d) [1 mark] What is the maximum number of clock-cycles taken by an instruction in 8051 ?

Solution: An instruction takes atleast 1 machine cycle, which is 12 clock-cycles. Instructions such as MUL, DIV take the maximum number of machine cycles (4), and hence 48 clock-cycles.

(e) [1 mark] Which register(s) in 8051 always stores an address ?

Solution: Program counter (PC), Stack Pointer (SP), Data Pointer (DPTR)

(f) [3 marks] What is the role of these signals in 8051 ?

i. *ALE*

Solution: Address latch enable is an output signal that demultiplexes address and data bus (indicates the microcontroller that the content on the bus corresponds to Address, when the ALE is high)

ii. *EA*

Solution: External Access is an input signal. When this is low, the internal ROM is disabled and program executes from external ROM.

iii. *PSEN*

Solution: Program store enable is an output signal, which enables external program (code) memory.

- (g) [1 mark] Give an equivalent instruction to achieve the following in fewer cycles.

MOV OEOH, #50H

Solution:

MOV A, #50H

- (h) [1 mark] Consider that the following instructions are executed in 8051. The accumulator will have the content of which memory location ?

SETB RS1
SETB RSO
MOV A, R7

Solution: The accumualator will have the content of 1FH.

- (i) [1 mark] What does the following instruction do ?

MOV C, 6

Solution: Places the bit in location 26H in Carry of PSW (PSW.7)

- (j) [1 mark] When using AJMP instruction, what limits the destination location ? How far (in terms of memory locations) can the execution jump ?

Solution: When using AJMP, the 5 MSBs of the location pointed by PC is used as such and the remaining 11 bits come from the instruction. Hence, the jump an be 2^{11} bytes (or 2K locations).

- (k) [1 mark] What is the minimum interrupt latency in 8051, which uses a 12 MHz clock ? Justify.

Solution: The interrupts are sampled in S5P2 (i.e., 5th state and phase 2 within a machine cycle) and polled in the following machine cycle to decide where the program should branch to. After this 2 machine cycles are required to store the return address in PC. Hence, a minimum latency of 3 machine cycles (or $3\mu s$) in this case is needed. More precisely $3.25\mu s$.

2. [4 marks] Write code to add two 4 digit BCD numbers. The first number is stored in 40H and 41H. The second number is in 42H and 43H. The higher order bits are in 41H and 43H, respectively. Store the BCD result in 40H and 41H, with higher order bits in 41H.

Solution:

```
MOV A, 40H
ADD A, 42H
DA A
MOV 40H, A
MOV A, 41H
ADDC A, 43H
DA A
```

3. [3 marks] Consider the 8051 assembly program shown below.

```
FUN: MOV R0, #50H
LOOP: MOV @R0, #0
      INC R0
      CJNE R0, #80H, LOOP
      END
```

- (a) [1 mark] What is the purpose of the subroutine FUN ?

Solution: This routine clears the 48 memory locations from 50H to 80H.

- (b) [2 marks] If you consider a clock frequency of 12 MHz, what is the execution time of the subroutine ?

Solution: Number of machine cycles: $1 + 48 * (1 + 1 + 2) = 193$, time = $193\mu s$

4. [5 marks] Consider a person detector sensor that generates a signal transitions from 1-to-0 on detecting a person. This sensor is to be connected to the 8051, so that you can use Timer 1 to check if 1000 people have entered a room. Once 1000 people enter, a LED connected to port P1.7 is to be turned ON.

- (a) [1 mark] Provide a brief description of the mode and connections that are needed to do this.

Solution: The Timer 1 needs to be in mode 1 and act as 'counter'. The TH1 and TL1 are loaded with value equivalent to 10000 and the timer is to be started by setting TR1. The sensor output is connected to port pin P3.5. The counter increments for each person entering (or 1-to-0 transition) and when it overflows, then the LED is to be turned on. The LED is connected to port P1.7.

- (b) [4 marks] Provide your 8051 code to realize this aspect of the application, with necessary comments.

Solution:

```

MOV TMOD, #50H      ; 16-bit counter in Timer 1
MOV TL1, #HighCount ; -1000 in Hex is FC18H 18H
MOV TH1, #LowCount  ; FCH
CLR P1.7            ; Turn-off the LED
COUNT: JNB TF1, COUNT ; Start counting
        SETB P1.7       ; Turn-on the LED once 10000 people enter
HERE: SJMP HERE

```

5. [5 marks] We are interested in measuring the pulse width (duration) of an external signal. How do we achieve this using the Timer-1 in 8051 ?

- (a) [1 mark] Mention which pin(s) are to be used to achieve this.

Solution: The external signal is connected to pin P3.3 (INT1).

- (b) [3 marks] Provide lines of code for appropriate configurations and mode settings

Solution: We dont need the entire code. Appropriate settings and a brief description is sufficient.

```

ORG 0013H ; External-1 ISR
JB INT1, LOW
JBC TR1, STOP
SETB TR1      ; start timer-1 to count

LOW: RETI      ; pulse has not started

STOP: CLR EX1 ; clear for next measurement
      RETI      ;

INIT: MOV TMOD, #90H ; Set Timer 1 in Gated mode
      SETB IT1    ; TCON.2 for timer-1 transition trigger
      MOV TL1, #00H

```

```
MOV TH1, #00H
```

```
MOV IE, #84H
```

; Enable global and external-1 interrupts.

You can write additional code to move the values of TH1 and TL1 to get the count and then convert it to pulse-width time.

- (c) [1 mark] Is there a limit on the pulse width that can be measured ? Justify.

Solution: Though the maximum value corresponding to FFFFH (65535) might seem like the maximum width, this is not necessarily the maximum width. We could use a count of number of overflows and measure a large pulse-width.

6. [2 marks] This is about using serial communication ports in Mode 0.

- (a) [1 mark] What do we gain by using the Serial Ports (Rxd, Txd) in this mode ?

Solution:

- We get an extra port pin for communication, but data is sent out serially.
- Fastest mode to communicate, though in a synchronous fashion with external device.

- (b) [1 mark] What is the disadvantage of using these pins for data transfer, when compared to use of other port pins (e.g., Port 1) ?

Solution:

- If we need multi bit output, we need to connect a serial-to-parallel shift register to as there is only one pin to output (or input).

7. [7 marks] We are interested in transmitting a character using the serial port.

- (a) [3 marks] Consider the program MAIN which uses subroutines INIT and OUTCHAR to use the serial port of 8051 as an 8-bit UART at 2400 baud rate to transmit a character. The CPU operates using a 12MHz clock and it is to output an ASCII character (7-bits and 1-bit to indicate parity) on the serial port. Complete the code to achieve this functionality with relevant comments. If needed add other missing line(s) of code.

Solution:

```
INIT: MOV SCON, #40H ;
      MOV TMOD, #20H ;
      MOV TL1, #0F3H ;
      MOV TH1, #0F3H ;
      CLR TI
      RET
```

```
OUTCHAR: CLR TI ;
```

```
        MOV SBUF, A
```

```
HERE:   JNB TI, HERE ;
```

```
        CLR TI ;
```

```

RET

MAIN: LCALL INIT
      MOV A, #'X'
      LCALL OUTCHAR
      END

```

```

INIT: MOV SCON,      ;
      MOV TMOD,     ;
      MOV TL1,      ;
      MOV TH1,      ;

      -----
      RET

OUTCHAR: CLR ___ ;
HERE:   JNB ___ , HERE ;
      CLR ___ ;

      -----
      MOV ___ , ___
      RET

MAIN: LCALL INIT
      MOV A, #'X'
      LCALL OUTCHAR
      END

```

- (b) [3 marks] You are to realize the same functionality as in previous part, using interrupt driven serial communication. The vector address for serial interrupt is 0023 H. Provide the INIT and ISR code to achieve this.

Solution:

```

INIT: MOV SCON, #40H ;
      MOV TMOD, #20H ; Timer-1 in auto-reload mode
      MOV TL1, #0F3H ;
      MOV TH1, #0F3H ;
      MOV IE, #90H ; Enable global and serial port interrupt
      CLR TI
      RET

SER_ISR: CLR TI
          RETI

MAIN: LCALL INIT
      MOV A, #'X'

HERE:  MOV SBUF, A
      JMP HERE

```

- (c) [1 mark] Which of the above approaches (part (a) or part(b)) is preferable if we require continuous transmission ? Why ?

Solution: The interrupt based transmission is preferred for continuous transmission as the processor need not wait to 'poll'.