SCHEME OF VALUATION

Revision: 2015

Course Title: MICROCONTROLLER AND INTERFAING(4043)

Qst. No	Scoring Indicators	Split up score	Sub Total	Total
I	PART A			
1.	1. 4 I/O Ports	2	2	
2.	2. The Data Pointer (DPTR) is the 8051's only user-accessible 16-bit (2-byte) register.	2	2	
3.	RET is used to return from a subroutine previously called by LCALL or ACALL. RETI is used to return from an interrupt service routine	2	2	
4.	C/T selects pulses to be counted up by the timer/counter: • 1 – Timer counts pulses brought to the Tx(Timer) pin. • 0 – Timer counts pulses from the internal oscillator.	2	2	
5.	Interfacing is the process of connecting devices together so that they can exchange the information and that proves to be easier to write the programs	2	2	
			2	10

(Scoring Indicators)

	PART-B			
II 1.	The program status word (PSW) register is an 8-bit register. It is also referred to as the *flag register*. Although the PSW register is 8 bits wide, only *6* bits of it are used by the 8051. The two unused bits are user-definable flags. Four of the flags are called *conditional flags,* meaning that they indicate some conditions that result after an instruction is executed. These four are CY (carry), AC (auxiliary carry), P (parity), and OV (overflow). The bits PSW.3 and PSW.4 are designated as RSO and RSI, respectively, and are used to change the bank registers. The PSW.5 and PSW 1 bits are general-purpose status flag bits and can be used by the programmer for any purpose. ie, they are user definable. Figure shows the bits of the PSW register.	6	6	

CY	AC	F0	RS1	RS0	OV	–	P

CY	PSW.7	Carry flag.
AC	PSW.6	Auxiliary carry flag.
F0	PSW.5	Available to the user for general purpose.
RS1	PSW.4	Register Bank selector bit 1.
RS0	PSW.3	Register Bank selector bit 0.
OV	PSW.2	Overflow flag.
--	PSW.1	User-definable bit.
P	PSW.0	Parity flag. Set/cleared by hardware each instuction cycle to indicate an odd/even number of 1 bits in the accumulator.

RS1	RS0	Register Bank	Address
0	0	0	00H - 07H
0	1	1	08H - 0FH
1	0	2	10H - 17H
1	1	3	18H - 1FH

II.

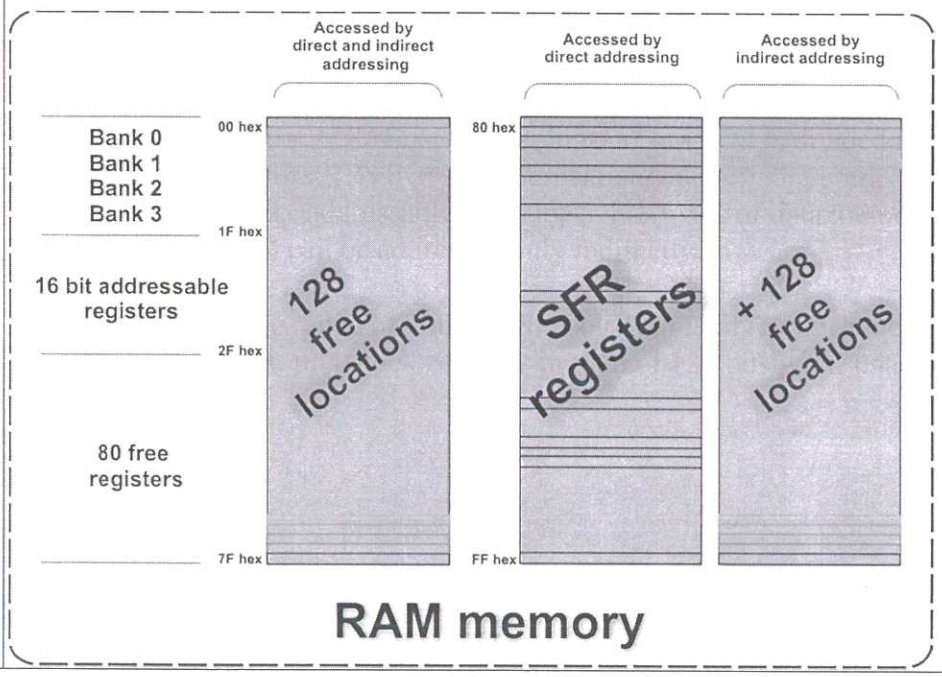
2.

Internal Data Memory

Up to 256 bytes of internal data memory are available depending on the 8051 derivative. Locations available to the user occupy addressing space from 0 to 7Fh, i.e. first 128 registers and this part of RAM is divided in several blocks. The first 128 bytes of internal data memory are both directly and indirectly addressable. The upper 128 bytes of data memory (from 0x80 to 0xFF) can be addressed only indirectly.

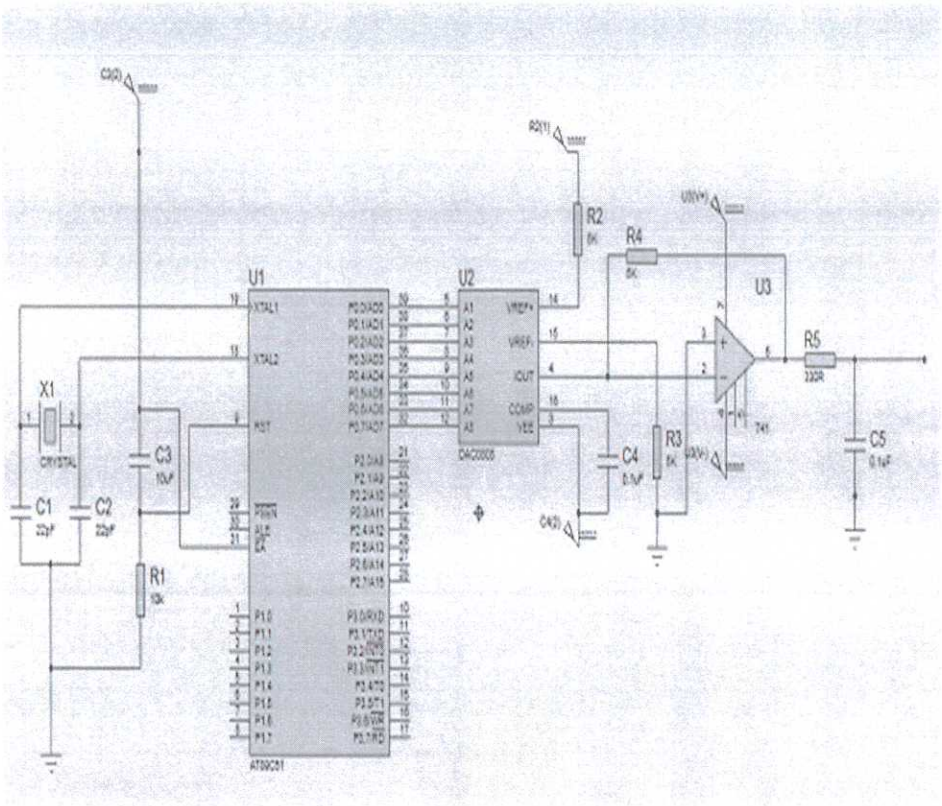
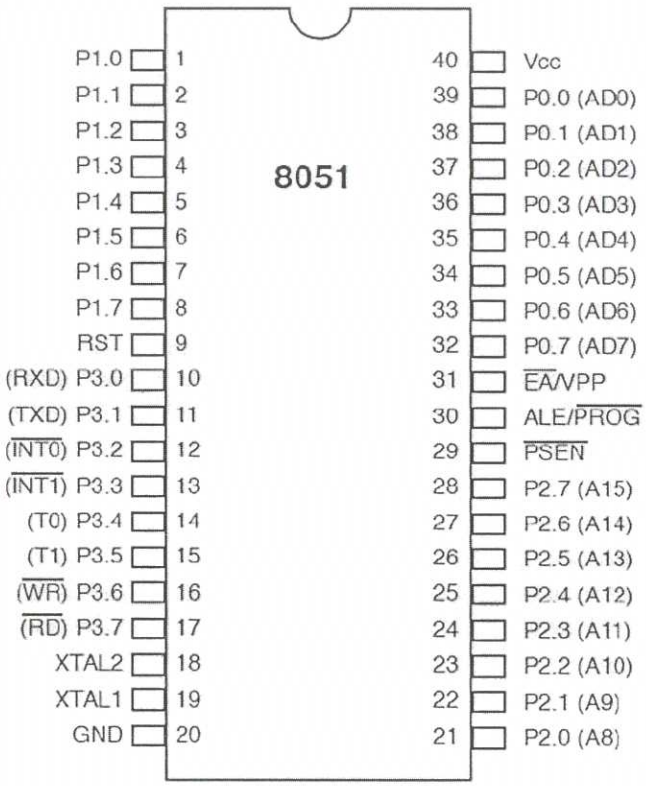
Memory block in the range of 20h to 2Fh is bit-addressable, Since there are 16 such registers, this block contains in total of 128 bits with separate addresses.

6 6



II
3.
	Mov r7, #0Ah ; initialize counter by 10d
	Mov r0, #20h ; get initial source location
	Mov dptr, #1020h ; get initial destination location
	Nxt: Mov a, @r0 ; get first content in acc
	Movx @dptr, a ; move it to external location
	Inc r0 ; increment source location
	Inc dptr ; increase destination location
	Djnz r7, nxt ; decrease r7. if zero then over otherwise move next
		6	6	

II 4.	**Interrupt priority upon reset** When the 8051 is powered up, the priorities are assigned according to Table below. From Table we see that if external hardware interrupts 0 and 1 are activated at the same time, external interrupt 0 (INTO) is responded to first. Only after INTO has been serviced is INT1 serviced, since INT1 has the lower priority. **Table 11-3: 8051/52 Interrupt Priority Upon Reset** **Highest to Lowest Priority** External Interrupt 0 (INT0) Timer Interrupt 0 (TF0) External Interrupt 1 (INT1) Timer Interrupt 1 (TF1) Serial Communication (RI + TI) Timer 2 (8052 only) TF2	6	6	
II 5.	TI (transfer interrupt) is raised when the last bit of the framed data, the stop bit, is transferred; indicating that the SBUF register is ready to transfer the next byte. RI (received interrupt), is raised when the entire frame of data, including the stop bit, is received. In other words, when the SBUF register has a byte, RI is raised to indicate that the received byte needs to be picked up before it is lost (overrun) by new incoming serial data. In the 8051 only one interrupt is set aside for serial communication. This interrupt is used to both send and receive data. If the interrupt bit in the IE register (IE.4) is enabled, when RI or TI is raised the 8051 gets interrupted and jumps to memory address location 0023H to execute the ISR. In that ISR we must examine the TI and RI flags to see which one caused the interrupt and respond accordingly. .	6	6	
II 6.	**Steps to program in mode 1** To generate a time delay, using the timer's mode 1, the following steps are taken. 1. 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. 1. Load registers TL and TH with initial count values. 2. Start the timer. 1. Keep monitoring the timer flag (TF) with the "JNB TFx, target" instruc tion to see if it is raised. Get out of the loop when TF becomes high. 3. Stop the timer. 4. Clear the TF flag for the next round. 5. Go back to Step 2 to load TH and TL again.	6	6	

II 7.	 	6	6	6x5 =30
III (a)	**PART C**  • **Pins 1 to 8** – These pins are known as Port 1. This port doesn't serve any other functions. It is internally pulled up, bi-directional I/O port.	5	5	

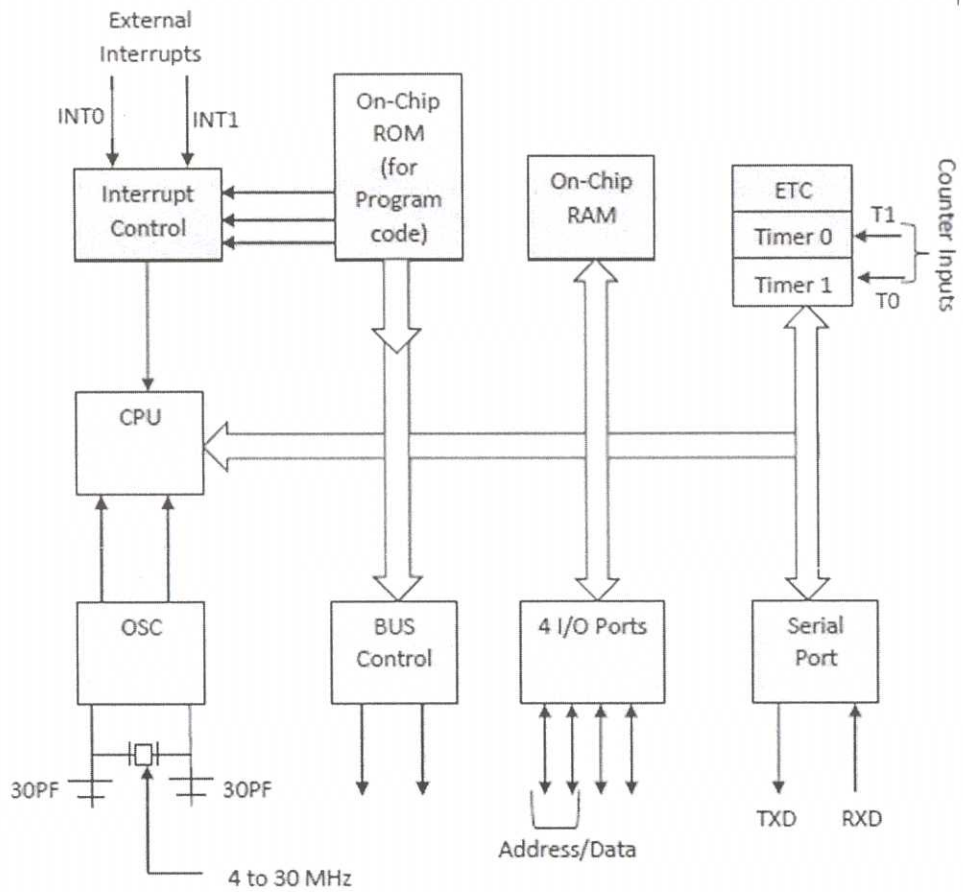
	• **Pin 9** – It is a RESET pin, which is used to reset the microcontroller to its initial values. • **Pins 10 to 17** – These pins are known as Port 3. This port serves some functions like interrupts, timer input, control signals, serial communication signals RxD and TxD, etc. • **Pins 18 & 19** – These pins are used for interfacing an external crystal to get the system clock. • **Pin 20** – This pin provides the power supply to the circuit. • **Pins 21 to 28** – These pins are known as Port 2. It serves as I/O port. Higher order address bus signals are also multiplexed using this port. • **Pin 29** – This is PSEN pin which stands for Program Store Enable. It is used to read a signal from the external program memory. • **Pin 30** – This is EA pin which stands for External Access input. It is used to enable/disable the external memory interfacing. • **Pin 31** – This is ALE pin which stands for Address Latch Enable. It is used to demultiplex the address-data signal of port. • **Pins 32 to 39** – These pins are known as Port 0. It serves as I/O port. Lower order address and data bus signals are multiplexed using this port. **Pin 40** – This pin is used to provide power supply to the circuit	5	10	
III (b)	**Features of 8051** 8-bit CPU. 16-bit Program Counter. 8-bit Processor Status Word (PSW) 8-bit Stack Pointer. (Any five) Internal RAM of 128bytes. Special Function Registers (SFRs) of 128 bytes. 32 I/O pins arranged as four 8-bit ports (P0 - P3) Two 16-bit timer/counters : T0 and T1.	5	5	15
IV (a)	It is an 8-bit microcontroller. It is built with 40 pins DIP (dual inline package), 4kb of ROM storage and 128 bytes of RAM storage, 2 16-bit timers. It consists of are four parallel 8-bit ports, which are programmable			

as well as addressable as per the requirement. An on-chip crystal oscillator is integrated in the microcontroller having crystal frequency of 12 MHz

4

4

The architecture of 8051 Microcontroller is shown in the diagram. The system bus connects all the support devices to the CPU. The system bus consists of an 8-bit data bus, a 16-bit address bus and bus control signals. All other devices like program memory, ports, data memory, serial interface, interrupt control, timers, and the CPU are all interfaced together through the system bus.



6

10

IV (b)

Comparison of 8051 family members:

Features	8051	8052	80
RAM(bytes)	128	256	12
ROM	4K	8K	0K
Timers	2	3	2
Serial port	1	1	1
I/O pins	32	32	32
Interrupt sources	6	8	6

5

15

15

V (a)

In 8051 There are five types of addressing modes.

Immediate Addressing Mode

Register Addressing Mode

Direct Addressing Mode

Register Indirect Addressing Mode

Indexed Addressing Mode

Immediate addressing mode

In this Immediate Addressing Mode, the data is provided in the instruction itself. The data is provided immediately after the opcode.

Examples of Immediate Addressing Mode.

2

10

```
MOVA, #0AFH;
MOVR3, #45H;
MOVDPTR, #FE00H;
```

In these instructions, the # symbol is used for immediate data. In the last instruction, there is DPTR. The DPTR stands for Data Pointer. Using this, it points the external data memory location. In the first instruction, the immediate data is AFH, but one 0 is added at the beginning. So when the data is starting with A to F, the data should be preceded by 0.

Register addressing mode

In the register addressing mode the source or destination data should be present in a register (R0 to R7). These are some examples of Register Addressing Mode.

```
MOVA, R5;
MOVR2, #45H;
MOVR0, A;
```

2

Direct Addressing Mode

In the Direct Addressing Mode, the source or destination address is specified by using 8-bit data in the instruction. Only the internal data memory can be used in this mode. Here some of the examples of direct Addressing Mode.

```
MOV80H, R6;
MOVR2, 45H;
MOVR0, 05H;
```

2

The first instruction will send the content of registerR6 to port P0 (Address of Port 0 is 80H). The second one is forgetting content from 45H to R2. The third one is used to get data from Register R5 (When register bank RB0 is selected) to register R5.

Register indirect addressing Mode

In this mode, the source or destination address is given in the register. By using register indirect addressing mode, the internal or external addresses can be accessed. The R0 and R1 are used for 8-bit addresses, and DPTR is used for 16-bit addresses, no other registers can be used for addressing purposes. Let us see some examples of this mode.

```
MOV0E5H, @R0;
MOV@R1, 80H
```

2

In the instructions, the @ symbol is used for register indirect addressing. In the first instruction, it is showing that theR0 register is used. If the content of R0 is 40H, then that instruction will take the data which is located at location 40H of the internal RAM. In the second one, if the content of R1 is 30H, then it indicates that the content of port P0 will be stored at location 30H in the internal RAM.

```
MOVXA, @R1;
MOV@DPTR, A;
```

In these two instructions, the X in MOVX indicates the external data memory. The external data memory can only be accessed in register indirect mode. In the first instruction if the R0 is holding 40H, then A will get the content of external RAM location40H. And in the second one, the content of A is overwritten in the location pointed by DPTR.

Indexed addressing mode

In the indexed addressing mode, the source memory can only be accessed from program memory only. The destination operand is always the register A. These are some examples of Indexed addressing mode.

```
MOVCA, @A+PC;
```

2

	MOVCA, @A+DPTR;			

V (b)	The C in MOVC instruction refers to code byte. For the first instruction, let us consider A holds 30H. And the PC value is1125H. The contents of program memory location 1155H (30H + 1125H) are moved to register A.
	Mov a, r0 ; get the content of r0 and r1
	Mov b, r1 ; in register A and B
	Div ab ; divide A by B
	Mov r2, a ; store result in r2
	Mov r3, b ; and reminder in r3
	Mov b, r1 ; again get content of r1 in
			5	15

VI (a)	Types of Interrupts in 8051 Microcontroller The 8051 microcontroller can recognize five different events that cause the main program to interrupt from the normal execution. These five sources of interrupts in 8051are: 1. Timer 0 overflow interrupt- TF0 2. Timer 1 overflow interrupt- TF1 3. External hardware interrupt- INT0 4. External hardware interrupt- INT1 5. Serial communication interrupt- RI/TI	5 5	5	
VI (b)	Upon activation of an interrupt, the microcontroller goes through the following steps; 1. It finishes the instruction it is executing and saves the address of the next instruction (PC) on the stack. 2. It also saves the current status of all the interrupts internally. 3. It jumps to a fixed location in the memory called the interrupt vector table that holds the address of the interrupt service routine. 4. The microcontroller gets the address of the ISR from the interrupt vector table and Jumps to it. It starts to execute the interrupt service routine until it reaches the last Instruction of the subroutine, which is RETI (return from interrupt). 5. Upon executing the RETI instruction, the microcontroller returns to the place where it was interrupted. First, it gets the program counter (PC) address from the stack by popping the top two bytes of the stack in to the PC. 6. Then it starts to execute from that address. IE (INTERRUPT ENABLE) REGISTER • This register is responsible for enabling and disabling the interrupt. EA register is set to one for enabling interrupts and set to 0 for disabling the interrupts. Its bit sequence and their	5 5	10	

meanings are shown in the following figure.

EA	-	-	ES	ET1	EX1

•

EA IE.7 It disables all interrupts. When EA = 0 no interrupt will acknowledged and EA = 1 enables the interrupt individu

- IE.6 Reserved for future use.

- IE.5 Reserved for future use.

ES IE.4 Enables/disables serial port interrupt.

ET1 IE.3 Enables/disables timer1 overflow interrupt.

EX1 IE.2 Enables/disables external interrupt1.

ET0 IE.1 Enables/disables timer0 overflow interrupt.

EX0 IE.0 Enables/disables external interrupt0.

2

3

5 15

VII (a)

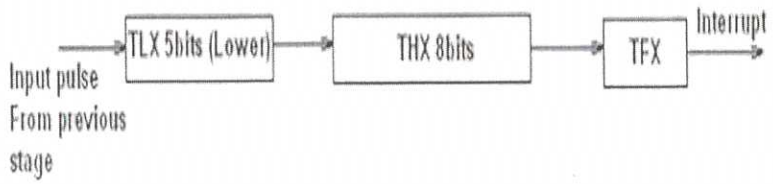
Timers can operate in four different modes. They are as follows

Timer Mode-0:

In this mode, the timer is used as a 13-bit UP counters as follows.

Operation of Timer on Mode-0

The lower 5 bits of TLX and 8 bits of THX are used for the 13 bit count. Upper 3 bits of TLX are ignored. When the counter rolls over from all 0's to all 1's, TFX flag is set and an interrupt is generated.



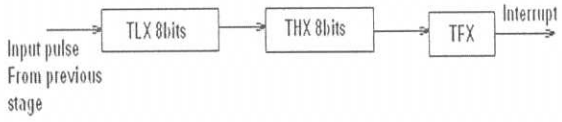
2.5

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 $\overline{INTX}$ input. This mode is useful to measure the width of a given pulse fed to $\overline{INTX}$ input.

10

Timer Mode-1:

This mode is similar to mode-0 except for the fact that the Timer operates in 16-bit mode.



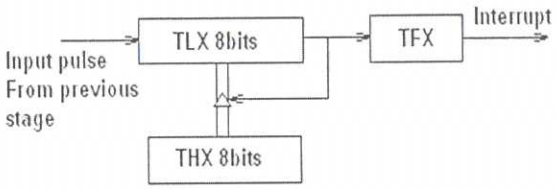
Operation of Timer in Mode 1

2.5

Timer Mode-2: (Auto-Reload Mode)

This is a 8 bit counter/timer operation. Counting is performed in TLX while THX stores a constant value. In this mode when the timer overflows i.e. TLX becomes FFH, it is fed with the value stored in THX. For example if we load THX with 50H then the timer in mode 2 will count from 50H to FFH. After that 50H is again reloaded. This mode is useful in applications like fixed time sampling.

2.5

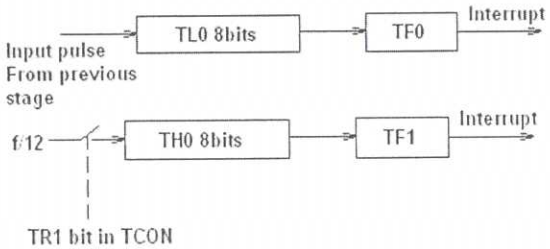


Operation of Timer in Mode 2

Timer Mode-3:

Timer 1 in mode-3 simply holds its count. The effect is same as setting TR1=0. Timer0 in mode-3 establishes TL0 and TH0 as two separate counters.

2.5

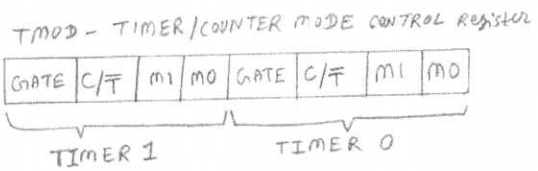


Operation of Timer in Mode 3

Control bits TR1 and TF1 are used by Timer-0 (higher 8 bits) (TH0) in Mode-3 while TR0 and TF0 are available to Timer-0 lower 8 bits (TL0).

VII (b)

Microcontroller TMOD register (Not bit addressable)



GATE- When GATE=1 and TRx(in TCON) is set, Timer/Counter-x will run while INT-x pin is set high(Hardware Control). When GATE=0, Timer/Counter-x will run only

5

5

15

while TRx=1(software control).

C/(T)- Set 0 for timer operation and Set 1 for counter operation

M1- Mode Selector bit

M0- Mode Selector bit

VIII (a) **Serial Communication: Different modes**

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

o ***Mode - 0*** **Shift register mode.**
Serial data centers and exists through RXD. 8-bits are transmitted/received. Pin *TXD* is connected to the internal *shift frequency* pulse source to supply shift pulses to external circuits. The shift frequency or baud rate is fixed at 1/2 of the oscillator frequency.

o ***Mode - 1*** **Standard UART**
10 bits are transmitted (through TXD) or received through (RXD), a start bit(0), 8 data bits (LSB first), and a stop bit(1). Once received, the stop bits goes into RB8 in special function register SCON.The baud rate is variable.

o ***Mode - 2*** **Multiprocessor Mode.**
11 bits are transmitted through TXD or received through RXD, a start bit (0), 8 data bits (LSB first), a programmable 9th bit and a stop bit(1). On transmission, the 9th data bit (TB8in SCON) can be assigned the value 0 or 1. Or, for example, the parity bit (P in the PSN) could be moved into TB8. On receive; the 9th bit goes into RB8 in SFR SCON, which the stop bit is ignored. The bandwidth is programmable to either 1/32 or 1/64 of oscillator frequency.

o ***Mode - 3***
11 bits are transmitted through TXD or received through RXD: a start bit, 8 data bits (LSBfirst), a programmable 9th bit, and a stop bit (1). In fact, Mode 3 is same as Mode 2 in all respects except the baud rate. The baud rate in *Mode 3 is variable.*

2.5*4

10

VIII (b) Register PCON controls processor power down, sleep modes and serial data rate. Only one bit of PCON is used with respect to serial communication seventh bit (b7)(SMOD) is used to generate the baud rate of communication.

Address: 87H

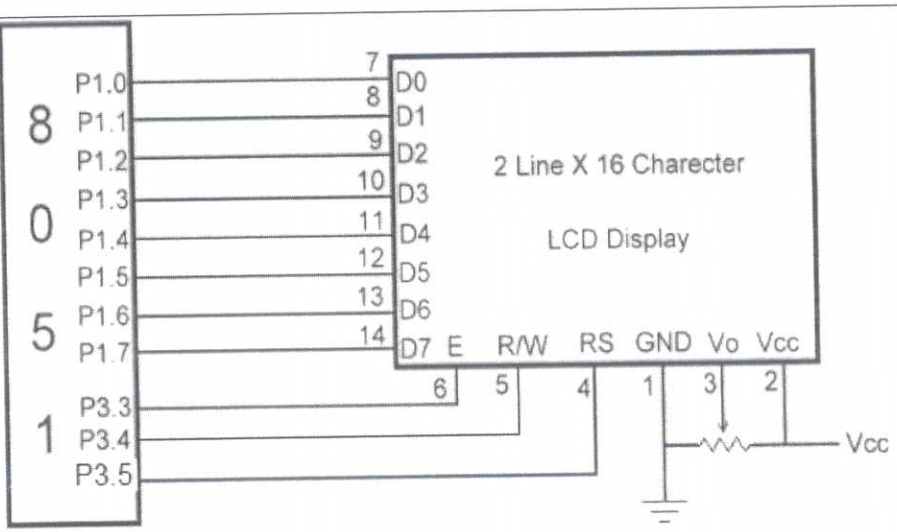
b7							b0
SMOD	—	—	—	GF1	GF0	PD	IDL

SMOD: Serial baud rate modify bit
GF1: General purpose user flag bit 1
GF0: General purpose user flag bit 0
PD: Power down bit
IDL: Idle mode bit

5

5

IX (a)



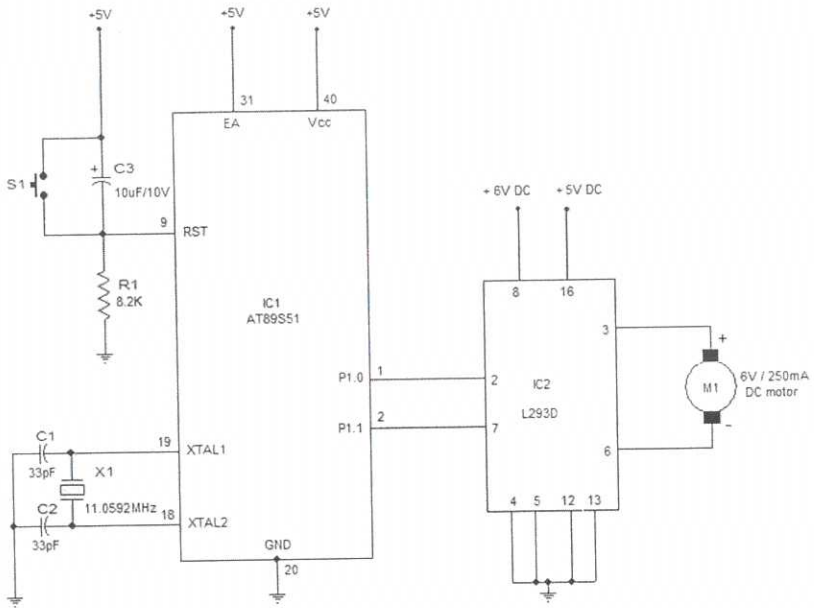
10

10

15

The circuit diagram given above shows how to interface a 16×2 LCD module with 8051 microcontroller. P1.0 to P1.7 pins of the microcontroller is connected to the D0 to D7 pins of the module respectively and through this route the data goes to the LCD module. P3.3, P3.4 and P3.5 are connected to the E, R/W, RS pins of the microcontroller and through this route the control signals are transferred to the LCD module. POT is used for adjusting the contrast of the display.

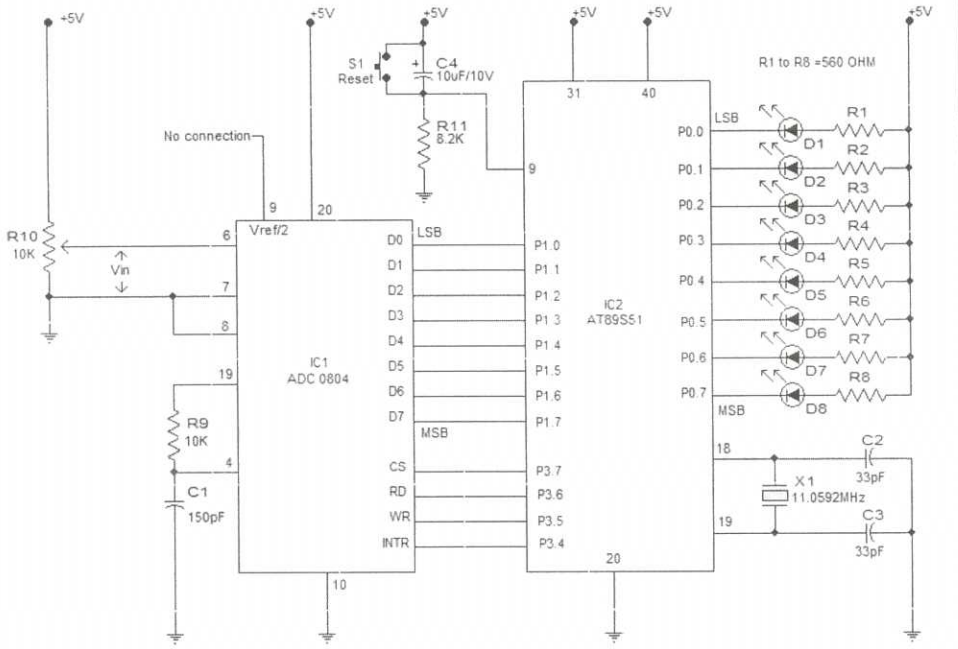
IX (b)



5

5

X (a)

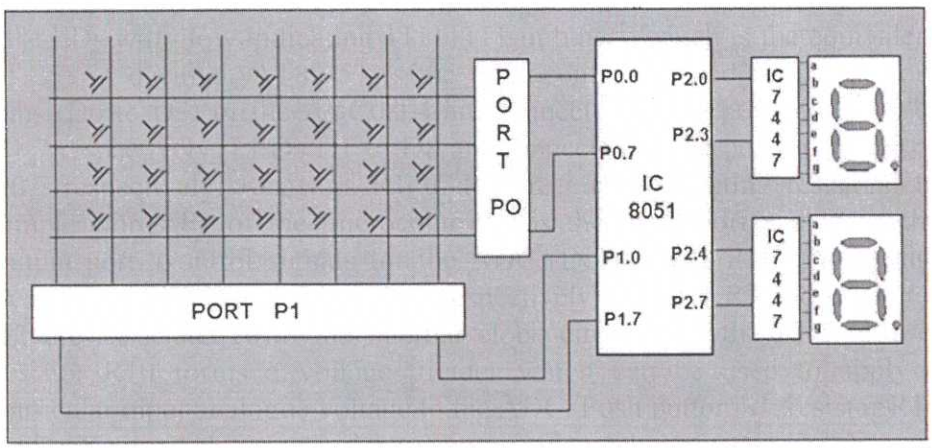


10

10

15

X (b) The figure above shows the schematic for interfacing ADC0804 to 8051. The circuit initiates the ADC to convert a given analogue input, then accepts the corresponding digital data and displays it on the LED array connected at P0. For example, if the analogue input voltage Vin is 5V then all LEDs will glow indicating 11111111 in binary which is the equivalent of 255 in decimal. AT89s51 is the microcontroller used here. Data out pins (D0 to D7) of the ADC0804 are connected to the port pins P1.0 to P1.7 respectively. LEDs D1 to D8 are connected to the port pins P0.0 to P0.7 respectively. Resistors R1 to R8 are current limiting resistors. In simple words P1 of the microcontroller is the input port and P0 is the output port. Control signals for the ADC (INTR, WR, RD and CS) are available at port pins P3.4 to P3.7 respectively. Resistor R9 and capacitor C1 are associated with the internal clock circuitry of the ADC. Preset resistor R10 forms a voltage divider which can be used to apply a particular input analogue voltage to the ADC. Push button S1, resistor R11 and capacitor C4 forms a debouncing reset mechanism. Crystal X1 and capacitors C2,C3 are associated with the clock circuitry of the microcontroller.



5

5

CODE:4043

COURSE: MICROCON TROLLER AND INTERFACING VERSION:2015

BLUE PRINT

SL NO	MODULE	TYPE OF QUESTIONS							
		PART-A		PART-B		PART-C		TOTAL	
		No of questions	score	No of questions	score	No of questions	Score	No of questions	score
1	I	1	2	2	12	4	30	7	44
2	II	2	4	2	12	4	30	8	46
3	III	1	2	2	12	4	30	7	44
4	IV	1	2	1	6	4	30	6	38
Total		5	10	7	42	16	120	28	172

QUESTION WISE ANALYSIS

COURSE :Microcontroler and interfacng

VERSION: 2015

Qn NO	Specific outcome (as per syllabus)	Module	Content details	score	Time in minutes
PART A					
I 1	1.1.8	I		2	3.6m
I 2	1.1.3	I		2	3.6m
I 3	2.2.3	II		2	3.6m
I 4	4.1.1	IV		2	3.6m
I 5	4.1.2	IV		2	3.6m
PART B					
II 1	1.1.3	I		6	10.8m
II 2	1.1.6	I		6	10.8m
II 3	2.1.3	II		6	10.8m
II 4	2.1.6	II		6	10.8m
II 5	3.1.1	III		6	10.8m
II 6	3.1.4	III		6	10.8m
II 7	4.1.3	IV		6	10.8m
PART C					
III (a)	1.1.4	I		10	18m
III (b)	1.1.2	I		5	9m
IV (a)	1.1.3	I		10	18m
IV (b)	1.1.5	I		5	9m
V (a)	2.1.1	II		10	18m
V (b)	2.1.3	II		5	9m
VI (a)	2.2.2,2.2.3	II		10	18m
VI (b)	2.2.4	II		5	9m
VII (a)	3.1.4	III		10	18m
VII (b)	3.1.3	III		5	9m
VIII (a)	3.2.3	III		10	10m

VIII (b)	3.2.5	III		5	18m
IX (a)	4.1.1	IV		10	9m
IX (b)	4.1.5	IV		5	18m
X (a)	4.1.3	IV		10	9m
X (b)	4.1.2	IV		5	18m
Total time					363.6m