

SCHEME OF VALUTION

(Scoring indicators)

REVISION:2015 COURSECODE:4043				
COURSE TITLE: MICROCONTROLLER AND INTERFACING				
Q. No	Scoring indicator	Splitup mark	Sub Total	Total
I(i)	Timer 0 and Timer 1	1+1	2	
I(ii)	R0 and R1 registers	1+1	2	
I(iii)	TCON register		2	
I(iv)	The conversion time is the primary parameter in deciding the speed of the ADC. The ADC conversion time is the time taken the process to change the input sampled analog to a digital value	2	2	
I(v)	Used in tape drives, floppy disc drives, printers and electric watches. The stepper motor also use in X-Y plotter and robotics.	Any two	2	10

PART B

II(i)

Distinguish between Microprocessor and Microcontroller

S.No	Microprocessor	Microcontroller
1	A microprocessor is a general purpose device which is called a CPU	A microcontroller is a dedicated chip which is also called single chip computer.
2	A microprocessor do not contain onchip I/O Ports, Timers, Memories etc.,	A microcontroller includes RAM, ROM, serial and parallel interface, timers, interrupt circuitry (in addition to CPU) in a single chip.
3	Microprocessors are most commonly used as the CPU in microcomputer systems	Microcontrollers are used in small, minimum component designs performing control-oriented applications.
4	Microprocessor instructions are mainly nibble or byte addressable	Microcontroller instructions are both bit addressable as well as byte addressable.
5	Microprocessor instruction sets are mainly intended for catering to large volumes of data.	Microcontrollers have instruction sets catering to the control of inputs and outputs.
6	Microprocessor based system design is complex and expensive	Microcontroller based system design is rather simple and cost effective
7	The Instruction set of microprocessor is complex with large number of instructions.	The instruction set of a Microcontroller is very simple with less number of instructions. For, ex: PIC microcontrollers have only 35 instructions.
8	A microprocessor has zero status flag	A microcontroller has no zero flag.

**Any 4 points
4x1.5=
6**

6

EVOLUTION OF MICROCONTROLLERS :

<p>II (ii)</p>	<div style="text-align: center;"> <p>(MSB) (LSB)</p> <p>7 6 5 4 3 2 1 0 (bit)</p> <table border="1" style="margin: auto;"> <tr> <td style="width: 20px; height: 20px;">CY</td> <td style="width: 20px; height: 20px;">AC</td> <td style="width: 20px; height: 20px;">FO</td> <td style="width: 20px; height: 20px;">RS1</td> <td style="width: 20px; height: 20px;">RS0</td> <td style="width: 20px; height: 20px;">OV</td> <td style="width: 20px; height: 20px;">-</td> <td style="width: 20px; height: 20px;">P</td> </tr> </table> <p style="margin-top: 10px;"> ↓ carry flag ↓ Auxiliary flag ↓ Register bank status ↓ Overflow flag ↓ parity flag </p> <p style="margin-top: 10px;"> ↓ user defined flag </p> <p>(program/processor status word)</p> </div> <p>Explain each bit</p>	CY	AC	FO	RS1	RS0	OV	-	P	<p>Diagram 3+Exp 3</p>	<p>6</p>	
CY	AC	FO	RS1	RS0	OV	-	P					
<p>II (iii)</p>	<pre> MOV DPTR, #0100H ;Initialize DPTR MOVX A,@DPTR ;Copy number from external memory to A MOV B,A ;copy number to B register MUL AB ;Multiply A and B registers INC DPTR ; Increment memory pointer MOVX @DPTR,A ; Copy Accumulator content to external memory INC DPTR ; Increment memory pointer MOV A,B ;Copy B register content to A MOVX @DPTR,A ; Copy Accumulator content to external memory HERE: SJMP HERE ;Stop </pre>	<p>6</p>	<p>6</p>									
<p>II (iv)</p>	<p>Arithmetic instructions: ALU in the 8051 is capable of performing several basic operations such as addition, subtraction, division, multiplication etc. After execution, the result is stored in the first operand.</p> <p>For example: ADD A,R1 - The result of addition (A+R1) will be stored in the accumulator.</p> <p>Arithmetic instructions of 8051:ADD,ADDC,SUBB,INC,DEC,MUL,DIV,DA A</p>	<p>Definition 2mark+ any 4 examples 1mark each</p>	<p>6</p>									

II
(v)

SCON: Serial Port Control Register (Bit Addressable)

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

- SM0 SCON.7 Serial Port mode specifier (NOTE 1).
- SM1 SCON.6 Serial Port mode specifier (NOTE 1).
- SM2 SCON.5 Enables the multiprocessor communication feature in mode 2 & 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0 (See table 9).
- REN SCON.4 Set/Cleared by software to Enable/Disable reception.
- TB8 SCON.3 The 9th bit that will be transmitted in modes 2 & 3. Set/Cleared by software.
- RB8 SCON.2 In modes 2 & 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
- TI SCON.1 Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes. Must be cleared by software.
- RI SCON.0 Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes (except see SM2). Must be cleared by software.

Note 1:

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	SMOD REGISTER	Fosc/12
0	1	1	5MHz UART	Variable
1	0	2	8MHz UART	Fosc/64 or Fosc/32
1	1	3	8MHz UART	Variable

6

Diagram
3+Exp 3

II
(vi)

1. Steps to program in mode 2

To generate a time delay using the timer's mode 2, take the following steps.

1. Load the TMOD value register indicating which timer (Timer 0 or Timer 1) is to be used, and select the timer mode (mode 2).
2. Load the TH registers with the initial count value.
3. Start the timer.
4. Keep monitoring the timer flag (TF) with the "JNB TFx, target" instruction to see whether it is raised. Get out of the loop when TF goes high.
5. Clear the TF flag.
6. Go back to Step 4, since mode 2 is auto-reload.

6

6

II
(vii)

In the DAC0808, the digital inputs are converted to current (I_{out}), and by connecting a resistor to the I_{out} pin, we convert the result to voltage. The total current provided by the I_{out} pin is a function of the binary numbers at the DO – D7 inputs of the DAC0808 and the reference current (I_{ref}), and is as follows:

$$I_{out} = I_{ref} \left(\frac{D7}{2} + \frac{D6}{4} + \frac{D5}{8} + \frac{D4}{16} + \frac{D3}{32} + \frac{D2}{64} + \frac{D1}{128} + \frac{D0}{256} \right)$$

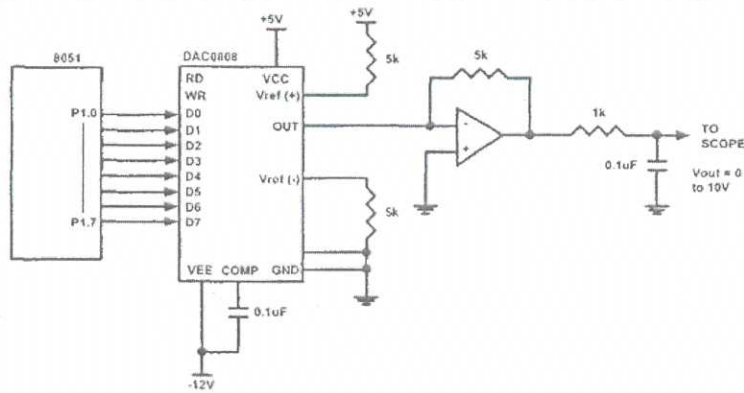


Diagram
3+Exp
3

6

where DO is the LSB, D7 is the MSB for the inputs, and I_{ref} is the input current that must be applied to pin 14. The I_{ref} current is generally set to 2.0 mA. Figure shows the generation of current reference (setting $I_{ref} = 2 \text{ mA}$) by using the standard 5-V power supply and 1K and 1.5K-ohm standard resistors.

Any
5
5x6=
30

III
(a)

PART -C

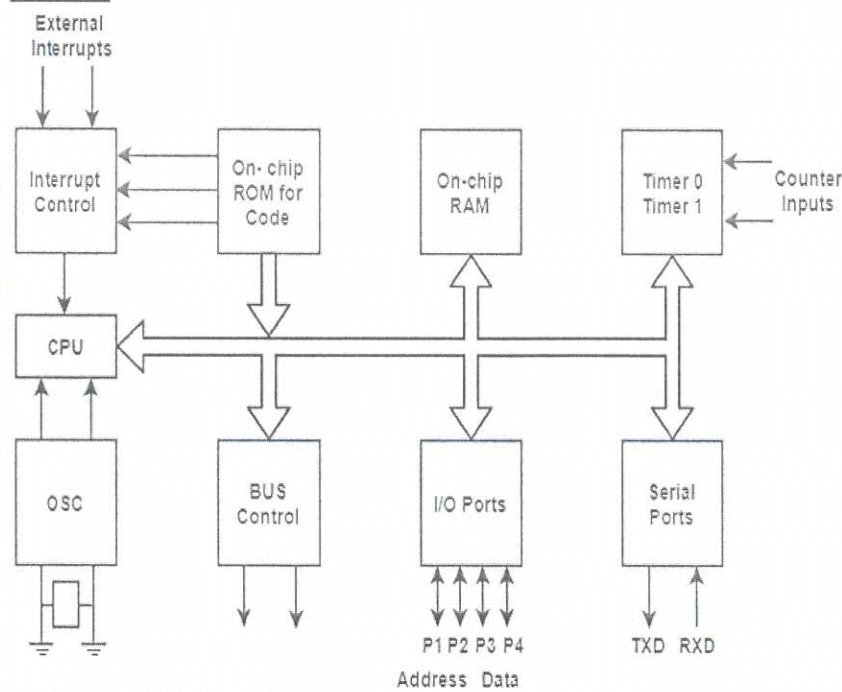


Diagram
4+
Exp 4

8

An 8051 microcontroller has the following 12 major components:

1. ALU (Arithmetic and Logic Unit)
2. PC (Program Counter)
3. Registers
4. Timers and counters

5. Internal RAM and ROM
6. Four general purpose parallel input/output ports
7. Interrupt control logic with five sources of interrupt
8. Serial data communication
9. PSW (Program Status Word)
10. Data Pointer (DPTR)
11. Stack Pointer (SP)
12. Data and Address bus.

RAM memory internally divided into a set of storage locations such as register banks, bit-addressable area, and scratch-pad area.

III
(b)

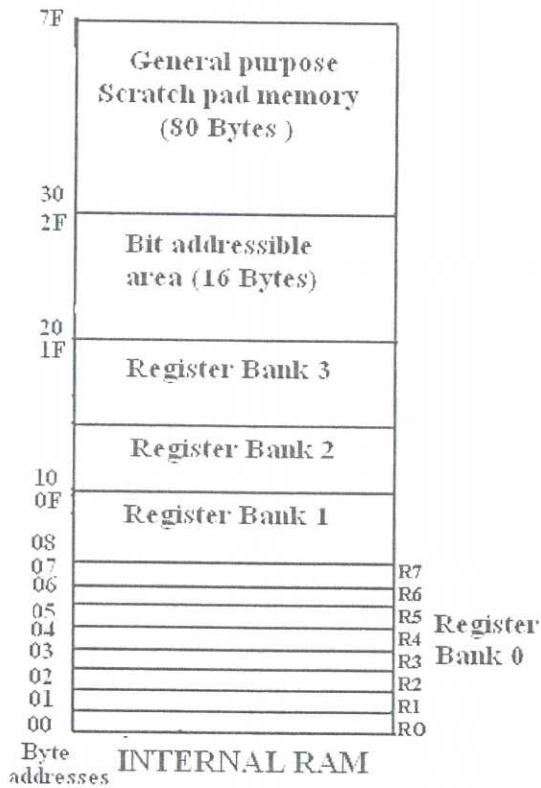


Diagram
m 3+
Exp 4

7

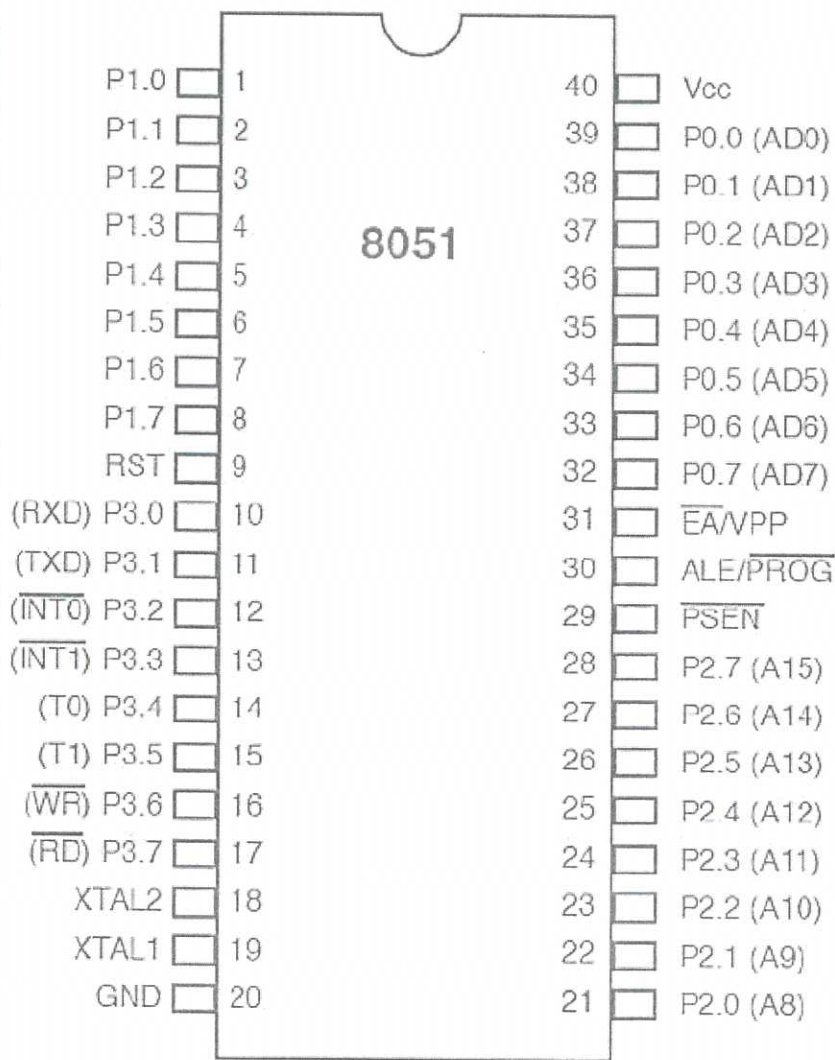
REGISTER BANKS:

The banks contain 8 general purpose registers such as R0-R7, and all such registers are byte-addressable registers that store or remove only 1-byte of data. The banks are divided into four different banks such as;

- Bank0
- Bank1
- Bank2
- Bank3

These can be selected by using the values of PSW register (i.e ,RS1,

<p>IV (a)</p>	<p>RS0).</p> <p>Bit Addressable Area: This area has total 128 addresses starting from 00h to 07Fh which represent for data storage location. Bit addressable area is formed close to the register banks. They are designed from address 20H to 2FH. Bit addressable area mainly used to store bit variables from an application program, like device output status, such as LEDs or motors (ON and OFF) etc.</p> <p>A Scratch Pad Area:</p> <p>Scratch pad area consists of byte addressable registers that store or remove only 1-bit of data. It is formed from 30H to 7FH. Scratch pad area mainly used to store byte variables from an application program, like print a device output status, such as motor directions (forward and backward) etc. Scratch pad area consists 80 bytes of memory.</p> <p>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.</p> <p>Pin 9 – It is a RESET pin, which is used to reset the microcontroller to its initial values.</p> <p>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.</p> <p>Pins 18 & 19 – These pins are used for interfacing an external crystal to get the system clock.</p> <p>Pin 20 – This pin provides the power supply to the circuit.</p> <p>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.</p> <p>Pin 29 – This is PSEN pin which stands for Program Store Enable. It is used to read a signal from the external program memory.</p>	<p>Diagram 4+ Exp 4</p>	<p>8</p>	
-------------------	--	------------------------------------	-----------------	--



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.

IV
(b)

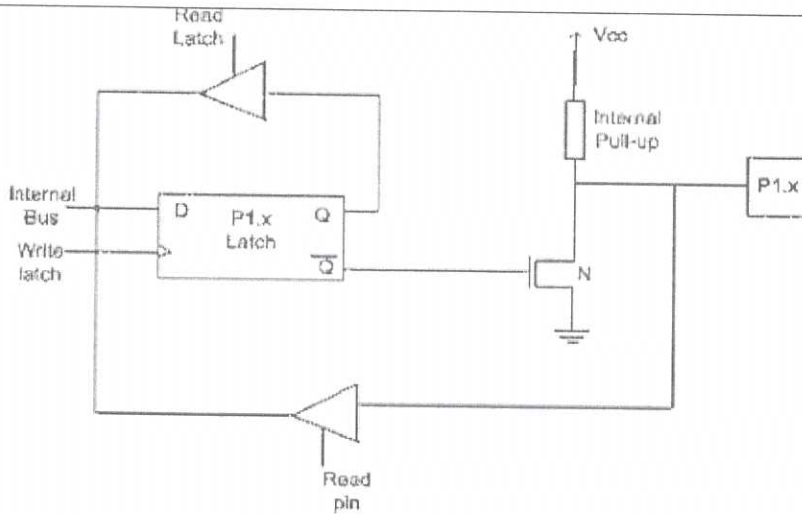


Diagram
m 3+
Exp 4

7

- Port 1 dedicated only for I/O interfacing when used as output port not needed to connect additional pull up register like port 0.
- It have provided internally pull up resistor the pin is pull up or down through internal pull-up when we want to initialize as an output port.
- To use port-1 as input port 1 has to be written to the latch., Input mode when '1' is written to the pin by the external device then it read fine. But when '0' is written to the pin by the external device then the external source.
- Must sink current due to internal pull-up if the external device is not able to sink the current the pin voltage may rise leading to a possible.
- The internal FET pull up is used to help the port-1 to speed up when it is used as an output port the internal FET pull up has another FET that is in parallel to lower FET.
- FET is turned on for two clock periods when the transition on the pin is low to high arrangement provides low impedance path to the positive supply voltage.

V(a)

Addressing modes of 8051

1. Immediate addressing mode: In this type, the operand is specified in the instruction along with the opcode. In simple way, it means data is provided in instruction itself. Ex: MOV A,#05H -> Where MOV stands for move, # represents immediate data. 05h is the data. It means the immediate date 05h provided in instruction is moved into A register.

2.Register addressing mode: Here the operand in contained in the specific register of microcontroller. The user must provide the name of register from where the operand/data need to be fetched. The

Any 4
4 x 2=8

8

permitted registers are A, R7-R0 of each register bank. Ex: MOV A,R0
content of R0 register is copied into Accumulator.

3. Direct addressing mode: In this mode the direct address of memory location is provided in instruction to fetch the operand. Only internal RAM and SFR's address can be used in this type of instruction.
Ex: MOV A, 30H => Content of RAM address 30H is copied into Accumulator.

4. Register Indirect addressing mode: Here the address of memory location is indirectly provided by a register. The '@' sign indicates that the register holds the address of memory location i.e. fetch the content of memory location whose address is provided in register.
Ex: MOV A,@R0 => Copy the content of memory location whose address is given in R0 register.

5. Indexed Addressing mode: This addressing mode is basically used for accessing data from look up table. Here the address of memory is indexed i.e. added to form the actual address of memory.Ex: MOVC A,@A+DPTR => here 'C' means Code. Here the content of A register is added with content of DPTR and the resultant is the address memory location from where the data is copied to A register.

V
(b)

Interrupts of 8051

Interrupt	Flag	Interrupt vector address
Reset	-	0000H
INT0 (Ext. int. 0)	IE0	0003H
Timer 0	TF0	000BH
INT1 (Ext. int. 1)	IE1	0013H
Timer 1	TF1	001BH
Serial	TI/RI	0023H

Reset

Reset is the highest priority interrupt, upon reset 8051 microcontroller start executing code from 0x0000 address.

External Interrupts:

Two interrupts are triggered by external signals provided by circuitry connected to the corresponding interrupts. External hardware

7

7

interrupt 0 and External interrupt interrupt1.

Internal Interrupts (Timer Interrupt):8051 has two internal interrupts namely timer0 and timer1. Whenever timer overflows, timer overflow flags (TF0/TF1) are set. Then the microcontroller jumps to their vector address to serve the interrupt. For this, global and timer interrupt should be enabled.

Serial interrupt:8051 has serial communication port and have related serial interrupt flags (TI/RI). When the last bit (stop bit) of a byte is transmitted, TI serial interrupt flag is set and when last bit (stop bit) of receiving data byte is received, RI flag get set.

VI
(a)

Rotate instructions of 8051

The ability to rotate Accumulator register data is useful to allow examination of individual bits of a byte. Register A can be rotated one bit position to left or right with or without including carry flag in the rotation.

RL A –Rotate Accumulator Left

RLC A –Rotate Accumulator Left through the carry.

RR A –Rotate Accumulator Right

RRC A –Rotate Accumulator Right through the carry.

4x2=8

8

VI
(b)

IE REGISTER

		(MSB)								(LSB)
	EA		ET2	ES	ET1	EX1	ET0			EX0
Symbol	Position	Name and Significance								
EA	IE.7	Enable All control bit. Cleared by software to disable all interrupts, independent of the state of IE.4-IE.0.								
	IE.6	(Reserved)								
ET2	IE.5	Timer 2 interrupt enable bit.								
ES	IE.4	Enable Serial port control bit. Set/cleared by software to enable/disable interrupts from TI or RI flags.								
ET1	IE.3	Enable Timer 1 control bit. Set/cleared by software to enable/disable interrupts from timer/counter 1								
EX1	IE.2	Enable External interrupt 1 control bit. Set/cleared by software to enable/disable interrupts from INT1.								
ET0	IE.1	Enable Timer 0 control bit. Set/cleared by software to enable/disable interrupts from timer/counter 0.								
EX0	IE.0	Enable external interrupt 0 control bit. Set/cleared by software to enable/disable interrupts from INT0.								

Fig. 12.29 IE-Interrupt enable register

Diagram 4+
Exp 3

7

Interrupt Enable (IE) Register: This register is responsible for enabling and disabling the interrupt. It is a bit addressable register in which EA must be set to one for enabling interrupts. The corresponding bit in this register enables particular interrupt like timer, external and serial inputs. In the below IE register, bit corresponding to 1 activates the interrupt and 0 disables the interrupt.

VII
(a)

Programming the 8051 to transfer character bytes serially, the following steps must be taken:

1. TMOD register is loaded with the value 20H, indicating the use of Timer1 in mode 2 (8 bit auto reload) to set baud rate.
2. TH1 is loaded with one of the values in the table to set the baud rate for serial data transfer.
3. The SCON register is loaded with the value indicating serial mode.
4. TR1 is set to 1 to start timer.
5. TI is cleared by the CLR TI instruction.
6. The character byte to be transferred serially is written in to the SBUF register.
7. The TI flag bit is monitored with the use of instruction JNB TI to see if the character has been transferred completely.
8. To transfer next character go to step 5.

8

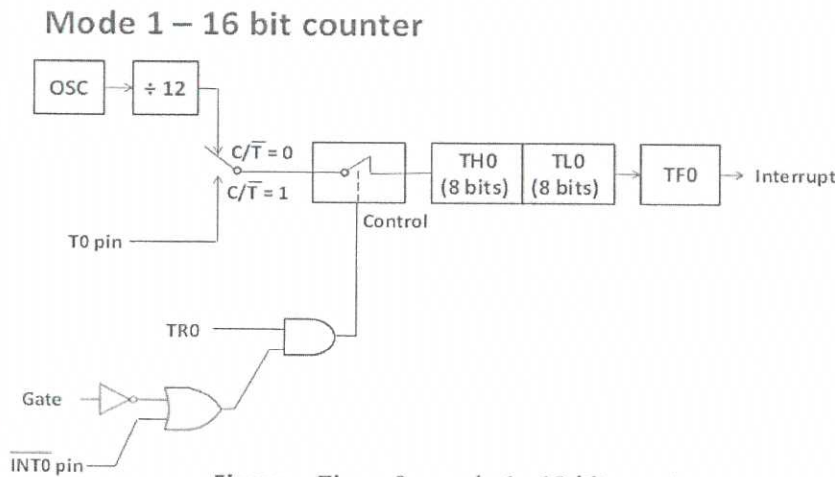
8

VII
(b)

MODE 1 is similar to MODE 0 except it is a 16 bit mode. In this mode the THx & TLx both acts as an 8 bit timer. The TLx counts upto 255 & then resets to 00 & increment THx by 1. Since this is a full 16 bit timer we can get maximum of 2^{16} i.e. 65536 Machine cycle before the timer overflows.

Diagram
m 4+
Exp 3

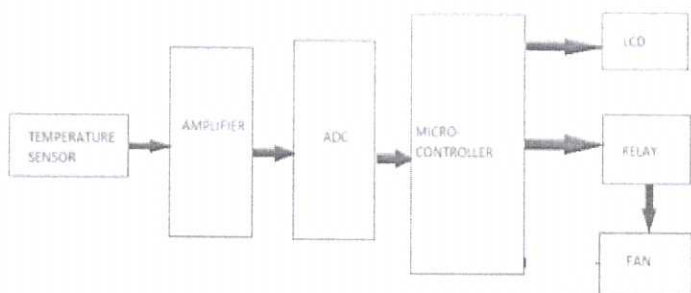
7



VIII (a)	<p>Mode 0: Only synchronous mode. Data transferred on RXD clock on TXD. Clock is fixed at 1/12 of the oscillator frequency</p> <p>Mode 1: Asynchronous mode. 10 bit data frame (start bit, 8 data bit and stop bit). Variable baud rate. Stop bit is placed in RB 8 bit of SCON register on reception of a frame.</p> <p>Mode 2: Asynchronous mode. 11 bit data frame (start bit, 8 data bit, programmable 9 bit and stop bit). On transmission 9th bit is TB 8 bit of SCON. On reception 9th bit is placed in RB 8 bit of SCON. 9th bit may be used for data or as a parity bit. Baud rate may be 1/32 or 1/64 of oscillator frequency.</p> <p>Mode 3: Same as mode 2 but with programmable baud rate.</p>	4x2=8	8	Diagram 3+ Exp 4	7																																				
VIII (b)	<p>TMOD Register This register contains bits controlling the operation of timer 0 & 1. To select the operating mode and the timer/counter operation of the timers we use TMOD register. Timer 0 and timer 1 are two timer registers in 8051. Both of these registers use the same register called TMOD to set various timer operation modes.</p> <ul style="list-style-type: none"> • TMOD is an 8-bit register. • The lower 4 bits are for Timer 0. • The upper 4 bits are for Timer 1. • In each case, <ul style="list-style-type: none"> ○ The lower 2 bits are used to set the timer mode. ○ The upper 2 bits to specify the operation. <p>TMOD : Timer/Counter Mode Control Register (Not Bit Addressable)</p> <table border="1" data-bbox="360 1305 1002 1395"> <tr> <td>GATE</td><td>C/T</td><td>M1</td><td>M0</td><td>GATE</td><td>C/T</td><td>M1</td><td>M0</td> </tr> <tr> <td colspan="4">TIMER 1</td><td colspan="4">TIMER 0</td> </tr> </table> <p>GATE When TRx (in TCON) is set and GATE = 1, TIMER/COUNTERx will run only while INTx pin is high (hardware control). When GATE = 0, TIMER/COUNTERx will run only while TRx = 1 (software control).</p> <p>C/T Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).</p> <p>M1 Mode selector bit (NOTE 1).</p> <p>M0 Mode selector bit (NOTE 1).</p> <p>Note 1 :</p> <table border="1" data-bbox="252 1664 1098 1843"> <thead> <tr> <th>M1</th><th>M0</th><th colspan="2">OPERATING MODE</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>12-bit Timer</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>16-bit Timer/Counter</td></tr> <tr> <td>1</td><td>0</td><td>2</td><td>8-bit Auto-Reload Timer/Counter</td></tr> <tr> <td>1</td><td>1</td><td>3</td><td>(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits, TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.</td></tr> <tr> <td>1</td><td>1</td><td>3</td><td>(Timer 1) Timer Counter 1 stopped</td></tr> </tbody> </table>					GATE	C/T	M1	M0	GATE	C/T	M1	M0	TIMER 1				TIMER 0				M1	M0	OPERATING MODE		0	0	0	12-bit Timer	0	1	1	16-bit Timer/Counter	1	0	2	8-bit Auto-Reload Timer/Counter	1	1	3	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits, TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.
GATE	C/T	M1	M0	GATE	C/T	M1	M0																																		
TIMER 1				TIMER 0																																					
M1	M0	OPERATING MODE																																							
0	0	0	12-bit Timer																																						
0	1	1	16-bit Timer/Counter																																						
1	0	2	8-bit Auto-Reload Timer/Counter																																						
1	1	3	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits, TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.																																						
1	1	3	(Timer 1) Timer Counter 1 stopped																																						

IX
(a)

Temperature control system

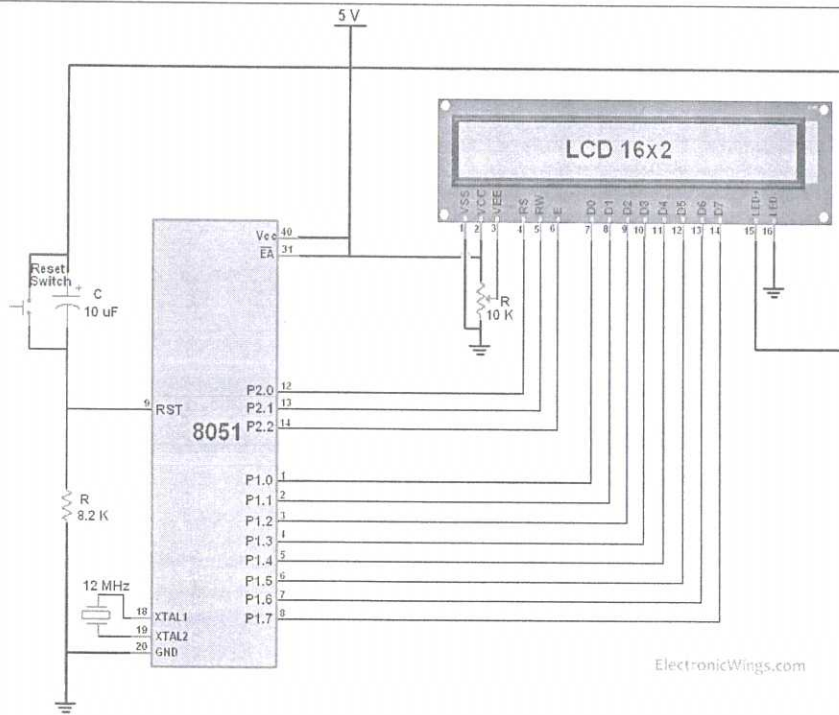


Temperature Sensor: The o/p of this temperature sensor is linearly proportional to the Celsius scale. This IC doesn't need any external calibration to give accuracies. The main function of the temperature sensor in the proposed system is to detect the temperature of external environment of fan. An ADC (analog to digital converter) needs to be interfaced with 8051 microcontroller to allow analog i/p for processing of data. ADC converts the input analog signal into equivalent digital signal by using the reference voltage.

The main function of the microcontroller in the proposed system is to analyze the temperature which is sensed by the temperature sensor. Based on the temperature, the microcontroller should change the speed of fan.

Diagram 4+
Exp 4 8

IX
(b)



Dim 3+
Exp 4

7

All the pins are clearly understandable by their name and functions, except the control pins, so they are explained below:

RS: RS is the register select pin. We need to set it to 1, if we are sending some data to be displayed on LCD. And we will set it to 0 if we are sending some command instruction like clear the screen (hex code 01).

RW: This is Read/write pin, we will set it to 0, if we are going to write some data on LCD. And set it to 1, if we are reading from LCD module. Generally this is set to 0, because we do not have need to read data from LCD. Only one instruction "Get LCD status", need to be read some times.

E: This pin is used to enable the module when a high to low pulse is given to it. A pulse of 450 ns should be given. That transition from HIGH to LOW makes the module ENABLE.

VEE pin 3 adjusts the contrast of the LCD display by varying voltage at this pin. Performed by connecting one end to ground potential and other to VCC (5V).

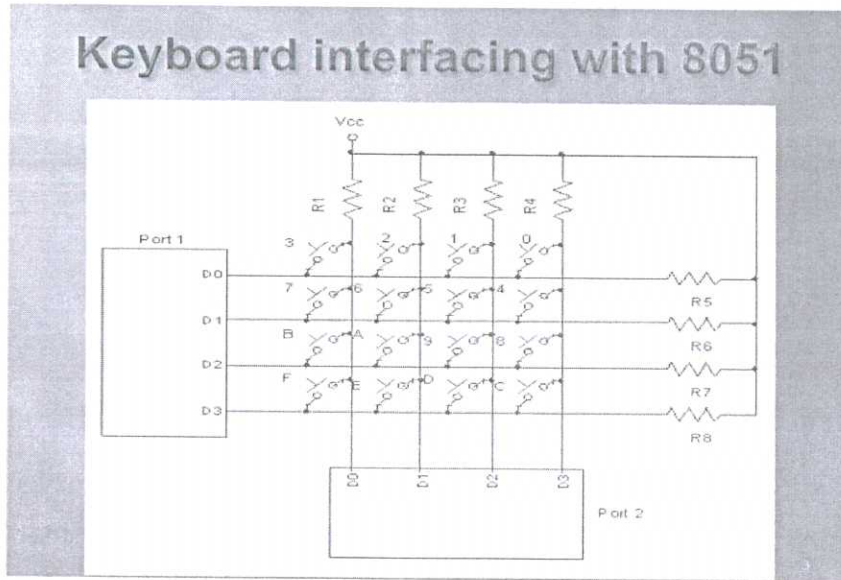
R/W pin 5 selects between read and write modes. LOGIC HIGH at pin facilitates READ mode and LOW level on this pin facilitates WRITE mode.

Data register- place the data to be displayed

Command register- places the commands. Inserts data in 8 bit data line (DB0 to DB7), the LCD module recognizes it as data to be displayed

DB0 to DB7 are data pins. The data to be displayed and the command instructions are placed on these pins.

X(a)



Dir 4 +
Exp 4

8

Keyboard is organized in a matrix of rows and columns as shown in the figure. The microcontroller accesses both rows and columns through the port.

1. The one of the port P1 that P1 means P1.0 to P1.7) as an I/P port for microcontroller 8051, port P0 as an O/P port of microcontroller 8051 and port P2 is used for displaying the number of pressed key.
2. Make all rows of port P0 high so that it gives high signal when key is pressed.
3. See if any key is pressed by scanning the port P1 by checking all columns for non zero condition.
4. If any key is pressed, to identify which key is pressed make one row high at a time.
5. Initiate a counter to hold the count so that each key is counted.
6. Check port P1 for nonzero condition. If any nonzero number is there in [accumulator], start column scanning by following step 9.
7. Otherwise make next row high in port P1.
8. Add a count of 08h to the counter to move to the next row by repeating steps from step 6.
9. If any key pressed is found, the [accumulator] content is rotated right through the carry until carry bit sets, while doing this increment the

count in the counter till carry is found.

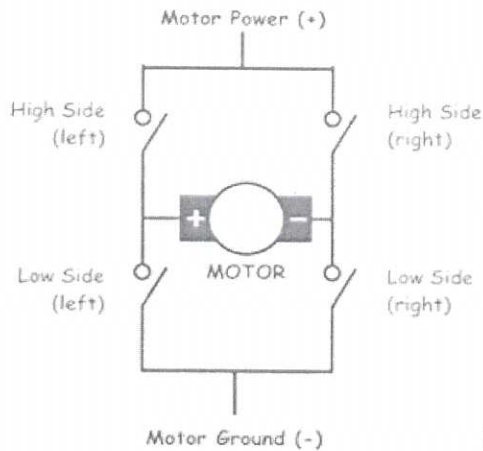
10. Move the content in the counter to display in data field or to memory location

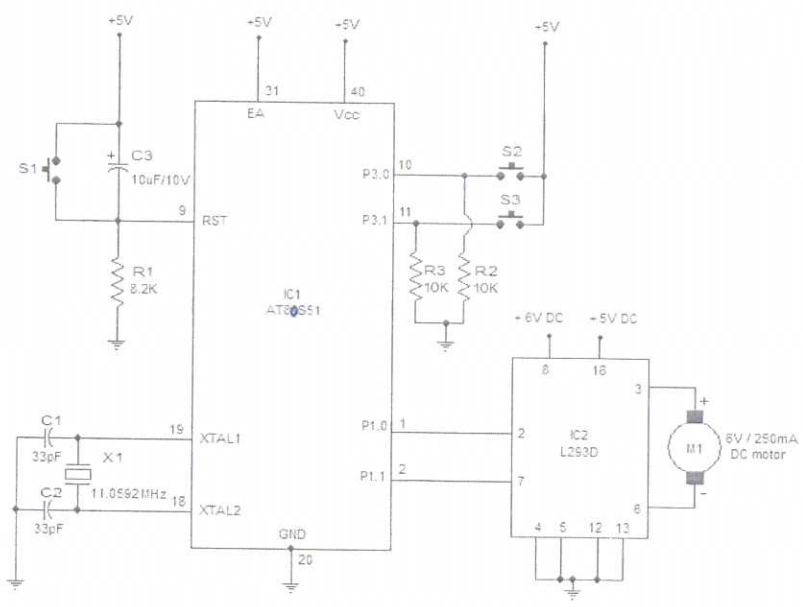
11. To repeat the procedures go to step 2.

X
(b)

The DC motor is another widely used device that translates electrical pulses into mechanical movement. Motor has 2 leads +ve and -ve, connecting them to a DC voltage supply moves the motor in one direction. On reversing the polarity rotates the motor in the reverse direction. Basic difference between Stepper and DC motor is stepper motor moves in steps while DC motor moves continuously. Maximum speed of a DC motor is indicated in rpm.

H-bridge can be made with the help of transistors and MOSFETs etc. It will be cheap but they increase the size of the design and circuit board which is mostly not required, so a small 16 pin IC is preferred for this purpose. Actually, the name "H-Bridge" is derived from the shape of the switching circuit which controls the motion of the motor. Basically there are four switching elements in the H-Bridge as shown in the figure below.





or any equivalent diagram

Diagram 7
m 4+
Exp 3

Any
4x15
=60