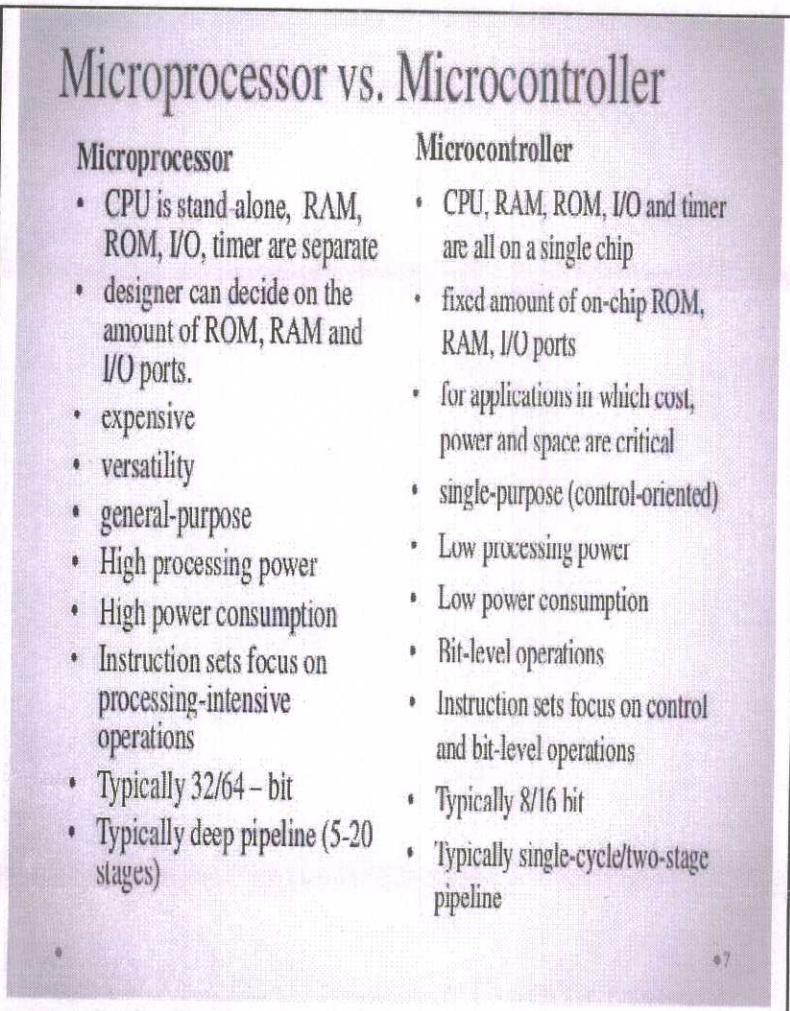
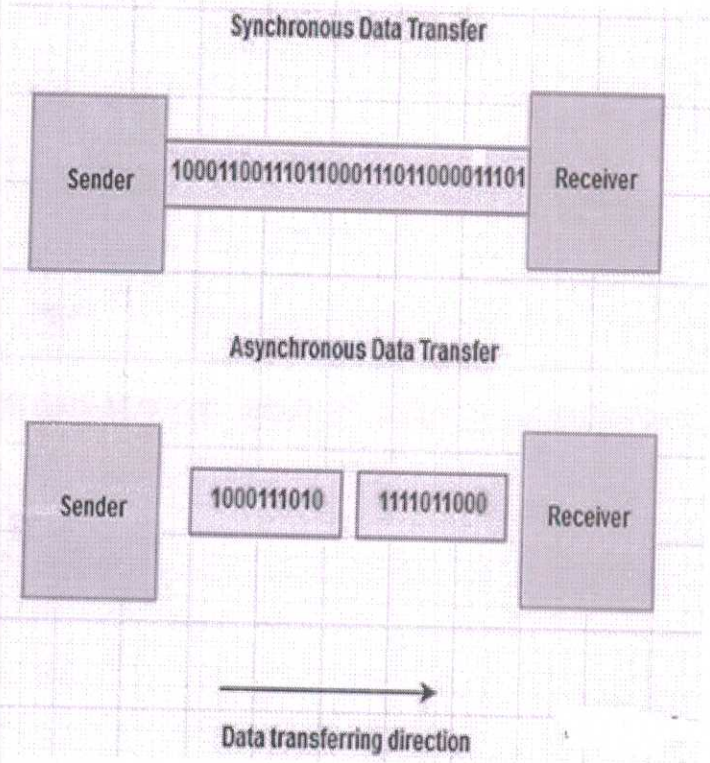
ScoringIndicators

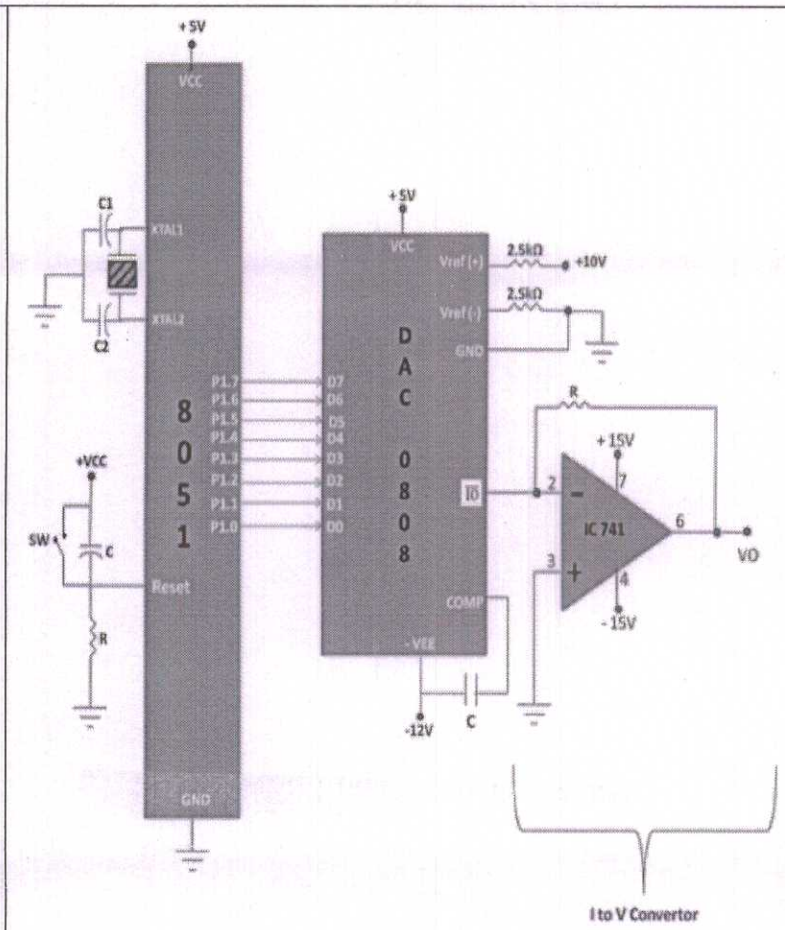
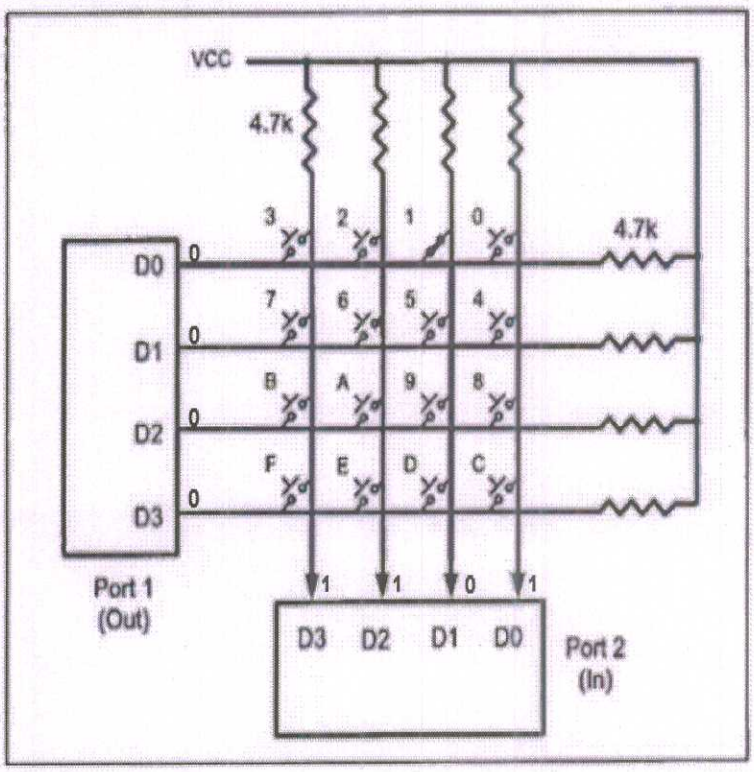
COURSENAME:MICROCONTROLLER AND APPLICATIONS

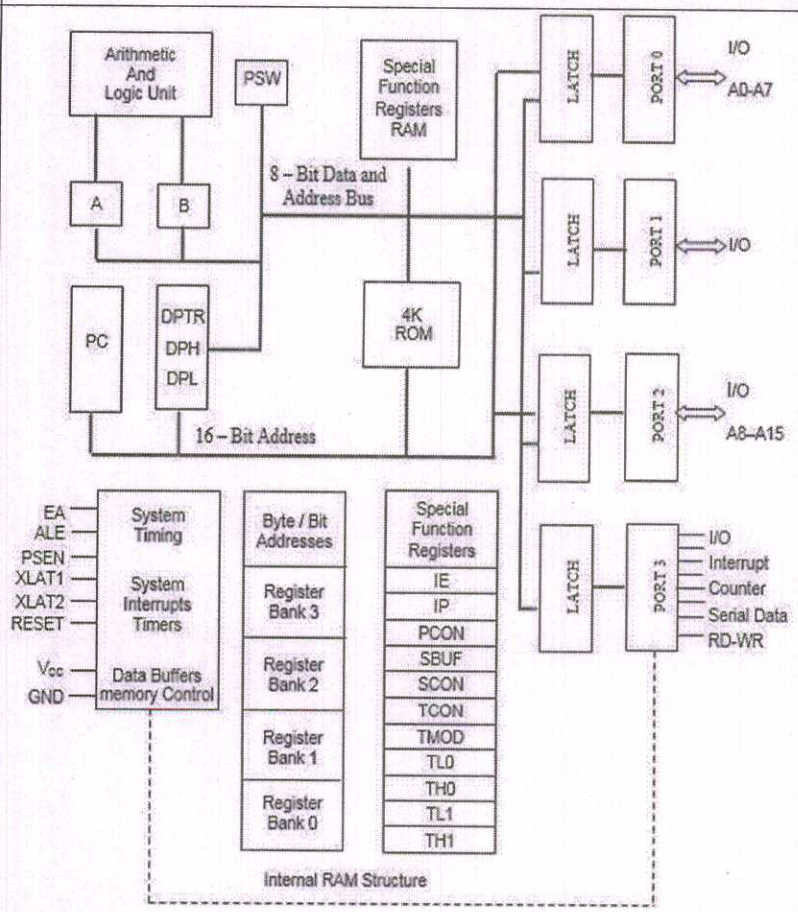
COURSECODE: REV (21)-4041 **QID: 2103230127**

QNo	Scoring Indicators	Split score	SubTotal	Totals core
	PARTA			**9**
I.1	Operand		1	
I.2	128 Bytes		1	
I.3	PSW (Program Status Word)		1	
I. 4	JC		1	
I.5	Bits/Sec		1	
I.6	0000H		1	
I.7	Interfacing		1	
I.8	Register Select (RS), Enable (E), Read/Write (R/W)		1	
I.9	RS232		1	
	PARTB			24

II.1			3	
II.2	Address Bus: 16 Bit, Unidirectional Data Bus : 8 bit, Bidirectional	2x1.5	3	
II.3	4 Register Banks. Bank 0, Bank 1, Bank 2, Bank 3	1+2	3	
II.4	Data Transfer Instructions Arithmetic Instructions Logical instructions Jump Instructions Rotate Instructions Loop Instructions	3x1	3	
II.5	JC, JNC, CPL C	3x1	3	
II.6	MOV: It is a data transfer instruction. It copies content of one register to another. MOVX: It is a data transfer instruction. Copies content of	2x1.5	3	

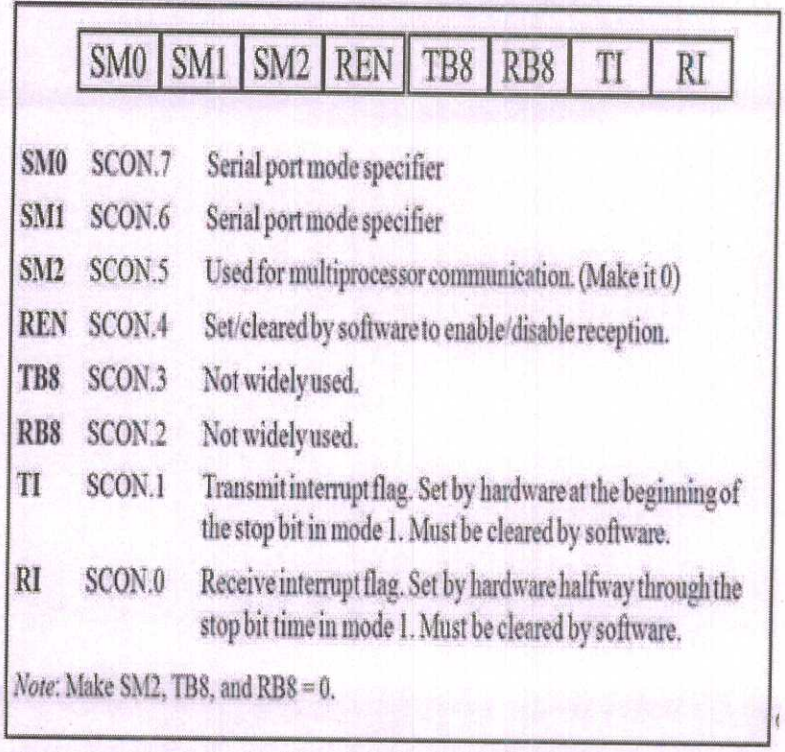
	external memory to Accumalator or from Accumalator to external memory.			
II.7	INT0, INT1, TF0,TF1, TI/RI		3	
II.8	**Asynchronous communication:** transfers a single byte at a time. The clocks of sende and receiver are not synchronized. **Synchronous communication**: transfers a block of data at a time.The clocks of sender and receiver are synchronized.  	2x1.5	3	

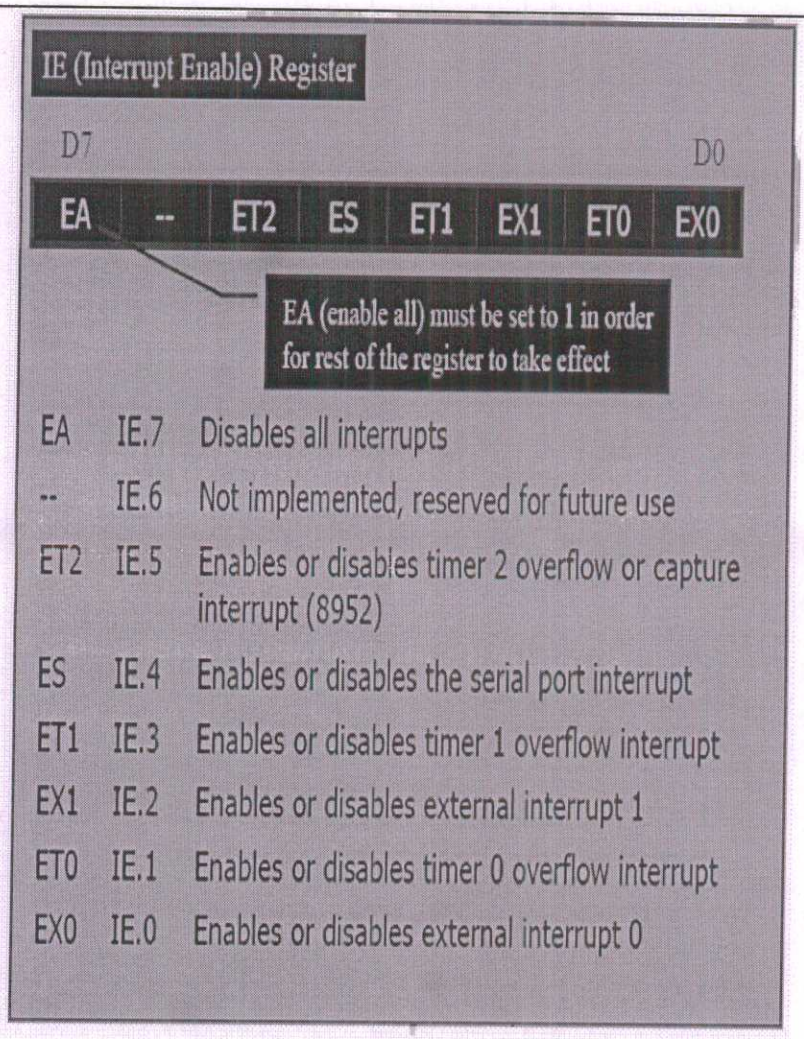
II.9	 +5V VCC C1 XTAL1 XTAL2 C2 8051 P1.7 P1.6 P1.5 P1.4 P1.3 P1.2 P1.1 P1.0 GND +VCC SW C Reset R DAC0808 D7 D6 D5 D4 D3 D2 D1 D0 Vref(+) Vref(-) GND COMP -VEE -12V C 2.5kΩ +10V 2.5kΩ R +15V IC 741 -15V VO I to V Convertor		3	
II.10	 VCC 4.7k 4.7k D0 D1 D2 D3 Port 1 (Out) D3 D2 D1 D0 Port 2 (In)		3	

	PARTC			42
III	➨ i. 4 KB on chip program memory (ROM or EPROM)). ➨ ii. 128 bytes on chip data memory(RAM). ➨ iii. 8-bit data bus ➨ iv. 16-bit address bus ➨ v. 32 general purpose registers each of 8 bits ➨ vi. Two -16 bit timers T0 and T1 ➨ vii. Five Interrupts (3 internal and 2 external). ➨ ix. Four Parallel ports each of 8-bits (PORT0, PORT1,PORT2,PORT3) with a total of 32 I/O lines. ➨ x. One 16-bit program counter and One 16-bit DPTR (data pointer) ➨ xi. One 8-bit stack pointer ➨ xii. One Microsecond instruction cycle with 12 MHz Crystal. ➨ xiii. One full duplex serial communication port.		7x1	7
IV			Fig 5 Exp 2	7

V	ORG 4100H
	CLR C
	MOV DPTR,#4300H
	MOVX A,@DPTR
	MOV B,A
	INC DPTR
	MOVX A,@DPTR
	MUL AB
	INC DPTR
	MOVX @DPTR,A
	INC DPTR
	MOV A,B
	MOVX @DPTR,A
	END
			7	7

VI	ORG 4100H
	MOV DPTR,#4500H
	MOVX A,@DPTR
	MOV B,A
	INC DPTR
	MOVX A,@DPTR
	ADD A,B
	INC DPTR
	MOVX @DPTR,A
	HERE:SJMP HERE
			7	7

VII	**SCON Serial Port Control Register (Bit Addressable)**  SM0 SM1 SM2 REN TB8 RB8 TI RI SM0 SCON.7 Serial port mode specifier SM1 SCON.6 Serial port mode specifier SM2 SCON.5 Used for multiprocessor communication. (Make it 0) REN SCON.4 Set/cleared by software to enable/disable reception. TB8 SCON.3 Not widely used. RB8 SCON.2 Not widely used. TI SCON.1 Transmit interrupt flag. Set by hardware at the beginning of the stop bit in mode 1. Must be cleared by software. RI SCON.0 Receive interrupt flag. Set by hardware halfway through the stop bit time in mode 1. Must be cleared by software. *Note*: Make SM2, TB8, and RB8 = 0.		7	7
VIII	When an interrupt event occurs, the microcontroller executes the following sequence of steps: 1. The CPU completes the execution of the instruction in progress, 2. The address of the next instruction present in the program counter is stored into the stack memory. 3. The current status of the interrupts are internally saved by the CPU. 4. The CPU suspends the normal code execution and the program branches to the preset memory location called the interrupt vector address. 5. The interrupt flag corresponding to the interrupt sources are cleared by the hardware. 6. The CPU executes the interrupt service subroutine that is located at the interrupt vector address. 7. Upon executing an RETI instruction that signifies the end of interrupt subroutine, the CPU restores the contents of the		7	7

	program counter from the stack. 8. The CPU thus branches back to the program address location where it was branched off when an interrupt occurred and continues the execution from the next instruction onwards.			
IX	 • Upon reset, all the interrupts in the microcontroller are disabled. • The interrupts must be enabled by the software for the microcontroller to respond. • The SFR called IE register allows the programmer to enable interrupts as needed. • 0 means disabled • 1 means enabled		7	7

X			7	7

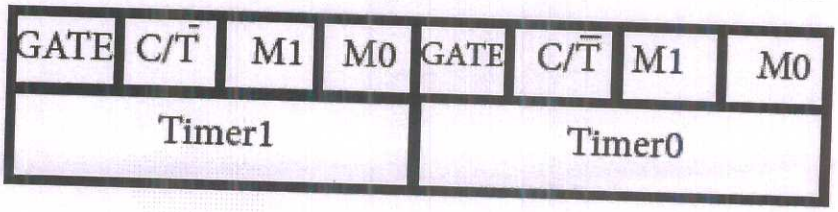


Fig. TMOD Register

This is an 8-bit register which is used by both timers 0 and 1 to set the various timer modes.

In this TMOD register, lower 4 bits are set aside for timer0 and the upper 4 bits are set aside for timer1.

In each case, the lower 2 bits are used to set the timer mode and upper 2 bits to specify the operation

Every timer has a means of starting and stopping.

Some timers do this by software, some by hardware, and some have both software and hardware controls.

The hardware way of starting and stopping the timer by an external source is achieved by making GATE=1 in the TMOD register.

If we change to GATE=0 then we do not need external hardware to start and stop the time

Bit 7,3 – GATE:

1 = Enable Timer/Counter only when the INT0/INT1 pin is high and

TR0/TR1 is set.

0 = Enable Timer/Counter when TR0/TR1 is set.

Bit 6,2 - C/(Counter/Timer): Timer or Counter select bit

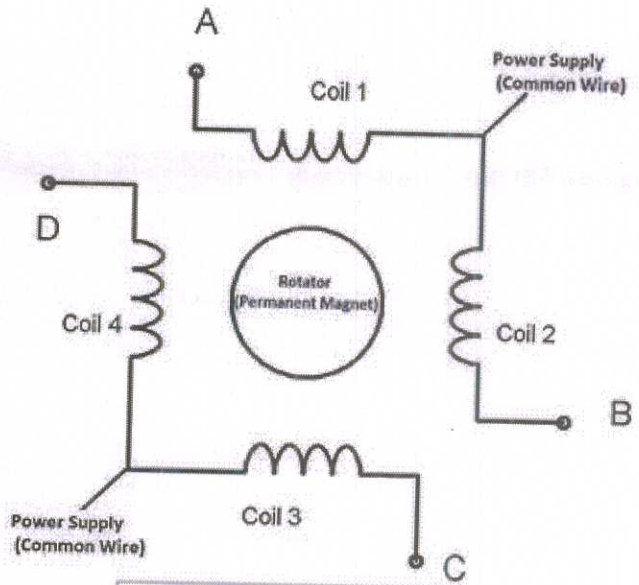
1 = Use as Counter

0 = Use as Timer

Bit 5:4 & 1:0 - M1:M0: Timer/Counter mode select bit

M1	M0	Mode	Operation
0	0	0 (13-bit timer mode)	13-bit timer/counter, 8-bit of THx & 5-bit of TLx
0	1	1 (16-bit timer mode)	16-bit timer/counter, THx cascaded with TLx
1	0	2 (8-bit auto-reload mode)	8-bit timer/counter (auto-reload mode), TLx reload with the value held by THx each time TLx overflow
1	1	3 (split timer mode)	Split the 16-bit timer into two 8-bit timers i.e. THx and TLx like two 8-bit timer

XI



Unipolar Stepper Motor

Whenever the coils energised by applying the current, the electromagnetic field is created, resulting the rotation of rotator (permanent magnet).

Coils should be energised in a particular sequence to make the rotator rotate.

The working method of Unipolar stepper motor is divided into three modes: Wave drive mode, full step drive mode and half step drive mode

Wave drive mode: In this mode one coil is energised at a time, all

7 7

four coil are energised one after another.

It produces less torque in compare with Full step drive mode but power consumption is less.

Give Logic 1 to the coils in the sequential manner.

Steps	A	B	C	D
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Full Drive mode: In this, two coil are energised at the same time producing high torque.

Power consumption is higher.

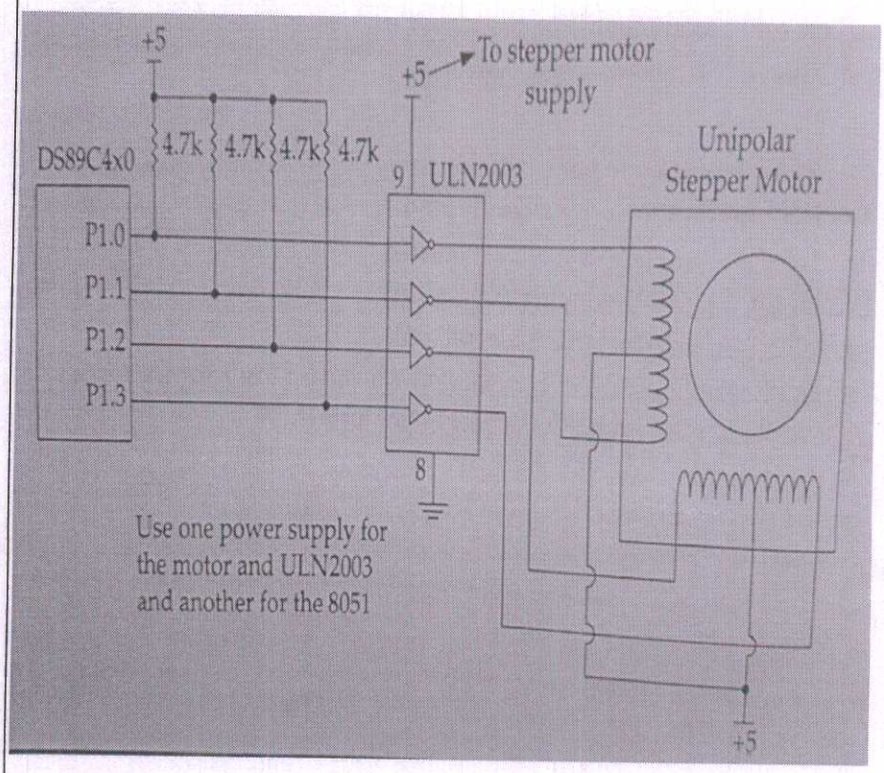
Give Logic 1 to two coils at the same time, then to the next two coils and so on.

Steps	A	B	C	D
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

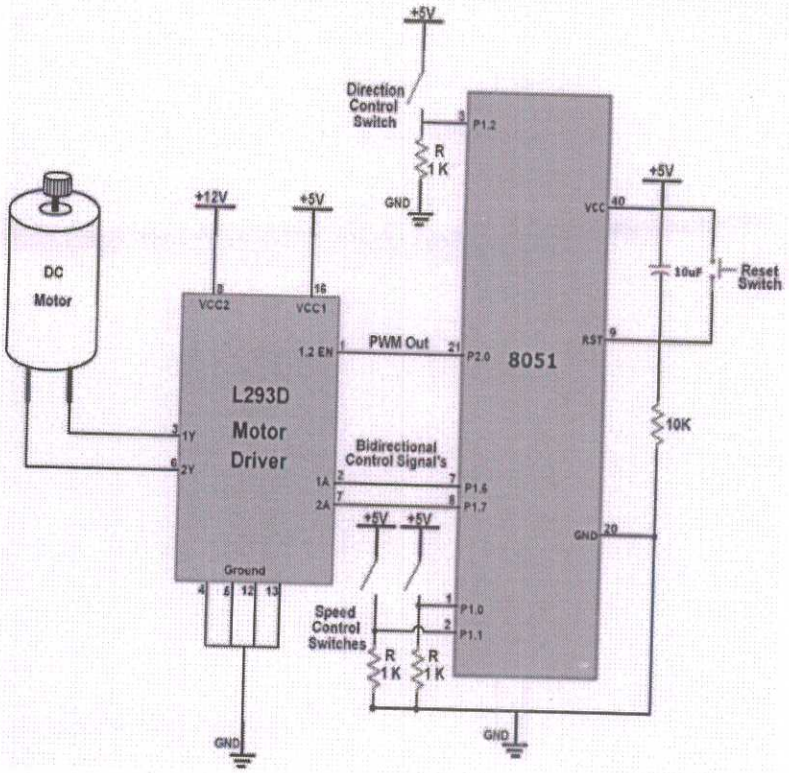
Half Drive mode: In this one and two coils are energised alternatively, ie one coil is energised first then two coils are energised then again one coil is energised then again two, and so on.

This is combination of full and wave drive mode, and used to increase the angular rotation of the motor.

Steps	A	B	C	D
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1



	8051 doesn't provide enough current to drive the coils so we need to use a current driver IC that is ULN2003A. ULN2003A is the array of seven NPN Darlington transistor pairs. Darlington pair is constructed by connecting two bipolar transistors to achieve high current amplification			
XII	DC motor converts electrical energy in the form of Direct Current into mechanical energy. The DC motor speed can be controlled by applying varying DC voltage; whereas the direction of rotation of the motor can be changed by reversing the direction of current through it. For reversing the current, we can make use of H-Bridge circuit or motor driver ICs that employ the H-Bridge technique or any other mechanisms The main purpose of DC interfacing with 8051 microcontroller is for controlling the speed of the motor Interfacing 8051 with DC motor requires a motor driver. L293D is typically used for interfacing DC motor with 8051. L293 is an IC with 16 pins This 16 pin L293D IC can be used for controlling the direction of two DC motors. The IC L293D works based on the H-bridge concept.		7	7



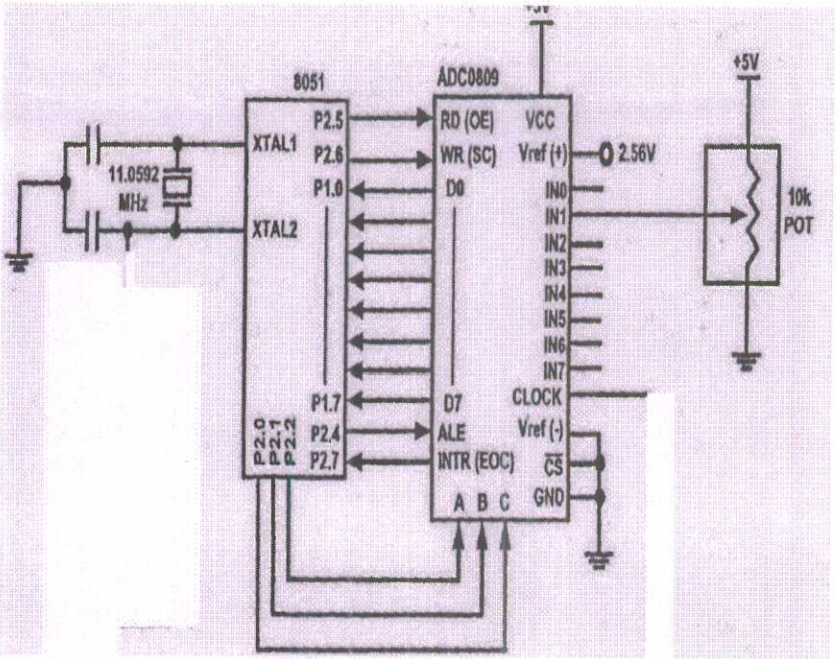
Two toggle switches are connected on P1.0 and P1.1 pin of the microcontroller to change the speed of the DC motor

P1.6 and P1.7 pins are used as output direction control pins.

It provides control to motor1 input pins of L293D motor driver which rotate the motor clockwise and anticlockwise by changing their terminal polarity.

Speed of the DC Motor is varied through PWM Out pin P2.0.

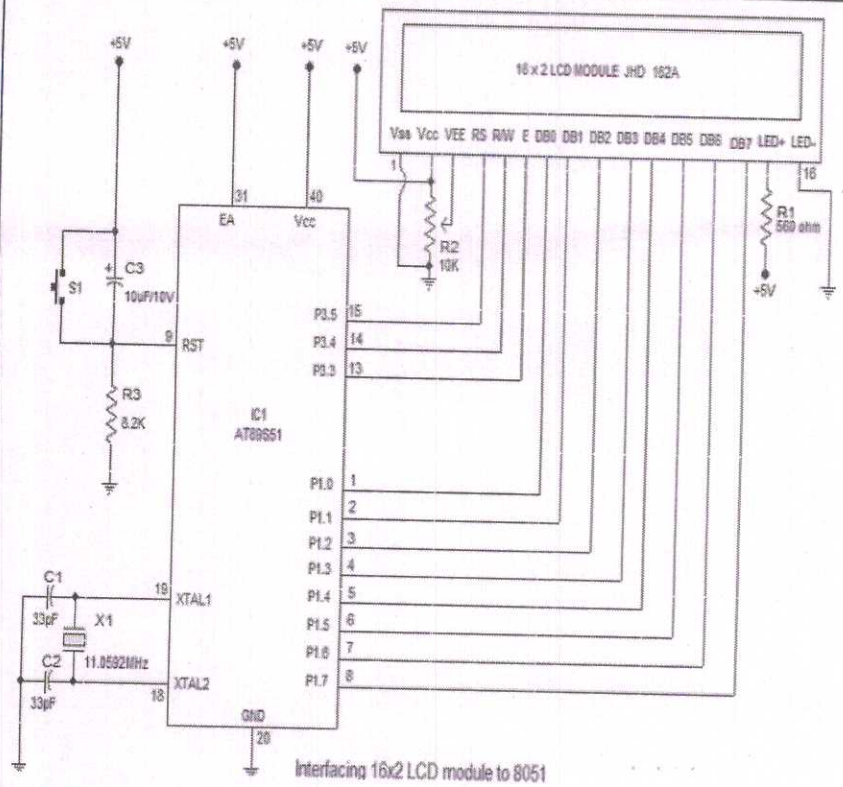
The timer of microcontroller is used to generate PWM

XIII	# 8051 Interfacing ADC  ## Address Lines (A,B,C)		7	7

Selected ADC channel	C	B	A
IN0	0	0	0
IN1	0	0	1
IN2	0	1	0
IN3	0	1	1
IN4	1	0	0
IN5	1	0	1
IN6	1	1	0
IN7	1	1	1

- ADC0808 has 8-bit data output
- 8 analog input channels are multiplexed and selected using three address pins A, B & C.

	Signals for A/D Conversion • **Address Latch Enable (ALE):** A LOW-TO-HIGH signal at this pin will latch the above-selected address and selected the respective channel for ADC conversion. • **START Conversion (SC):** The A/D converter's successive approximation register (SAR) is reset on the positive edge of the start conversion (SC) pulse. Thus we need to generate a LOW-HIGH pulse for starting the ADC conversion. • **End of Conversion (EOC):** Once the conversion is over, this pin is pulled HIGH by ADC0808. This pin needs to be monitored for the conversion to complete and then read the data. • **Output Enable(OE):** ADC0808 does the A/D conversion and holds the data in the internal registers. A HIGH signal on this pin will bring the data on the output lines. • D0-D7 are the digital Data output lines. It uses the principle of successive approximation for calculating digital values, which is very accurate for performing 8-bit analog to digital conversions. The 0808 does not have an internal clock and needs an external clock signal to operate.			
XIV	16x2 LCD has 2 rows and 16 character display Register Select (RS)- LCD has two registers- Instruction Command Code register and Data register. RS=0 selects ICC register RS=1 selects Data register		7	7



Interfacing 16x2 LCD module to 8051

R/W=0 for writing to LCD

Enable or E is used by LCD to latch the information presented to the data pins.

Pin D0-D7 are the data/command bits

Pin15 (+ve pin of the LED): This pin is connected to +5V

Pin 16 (-ve pin of the LED): This pin is connected to GND.

8051 Interfacing LCD (Liquid Crystal Display)

Pin Descriptions for LCD

Pin	Symbol	I/O	Descriptions
1	VSS	--	Ground
2	VCC	--	+5V power supply
3	VEE	--	Power supply to control contrast
4	RS	I	RS=0 to select command register, RS=1 to select data register
5	R/W	I	R/W=0 for write, R/W=1 for read
6	E	I/O	Enable
7	DB0	I/O	The 8-bit data bus
8	DB1	I/O	The 8-bit data bus
9	DB2	I/O	The 8-bit data bus
10	DB3	I/O	The 8-bit data bus
11	DB4	I/O	The 8-bit data bus
12	DB5	I/O	The 8-bit data bus
13	DB6	I/O	The 8-bit data bus
14	DB7	I/O	The 8-bit data bus

- Send displayed information or instruction command codes to the LCD
- Read the contents of the LCD's internal registers

used by the LCD to latch information presented to its data bus