WINTER – 19EXAMINATION

Subject Name: Microcontroller & Application **Model Answer** Subject Code: 22426

Important Instructions to examiners:

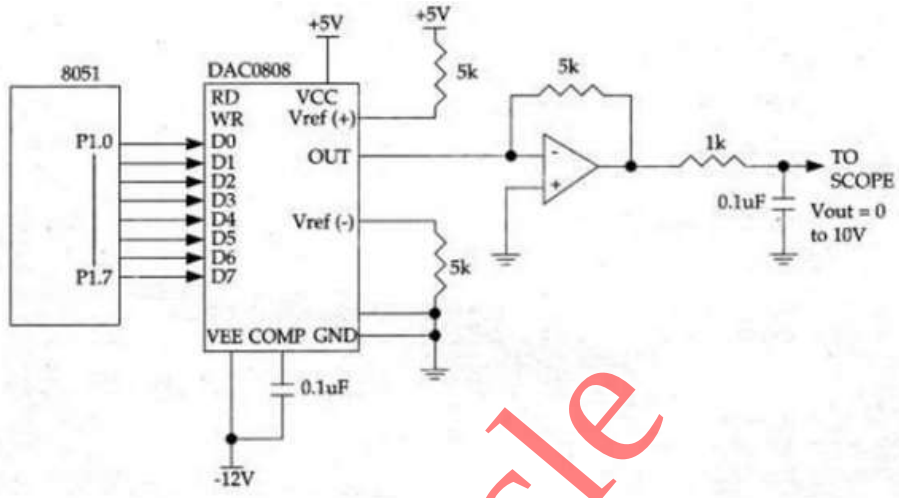
1) The answers should be examined by key words and not as word-to-word as given in themodel answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may tryto assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given moreImportance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in thefigure. The figures drawn by candidate and model answer may vary. The examiner may give credit for anyequivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constantvalues may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

<table>
<tr><th>Q. No.</th><th>Sub Q. N.</th><th>Answer</th><th>Marking Scheme</th></tr>
<tr><td>Q.1</td><td></td><td>Attempt any FIVE of the following:</td><td>10M</td></tr>
<tr><td></td><td>a)</td><td>Compare address bus and data bus used in 8051.</td><td>2M</td></tr>
<tr><td></td><td>Ans:</td><td>
<table>
<tr><th>Sr. No.</th><th>Address Bus</th><th>Data Bus</th></tr>
<tr><td>1</td><td>A bus that is used to specify a physical address in memory</td><td>A bus that is used to transmit data among components</td></tr>
<tr><td>2</td><td>Unidirectional</td><td>Bidirectional</td></tr>
<tr><td>3</td><td>Helps to transfer memory address of data and I/O</td><td>Helps to send and receive data</td></tr>
<tr><td>4</td><td>16 bit address bus in 8051</td><td>8 bit data bus in 8051</td></tr>
</table>
</td><td>1M each (Any 2 points)</td></tr>
<tr><td></td><td>b)</td><td>Calculate the number of address lines required to access 16 kB ROM.</td><td>2M</td></tr>
<tr><td></td><td>Ans:</td><td>14 address lines required to access 16 KB of ROM as
2^{14} = 16KB</td><td>2M</td></tr>
<tr><td></td><td>c)</td><td>State features of ADC 0808.</td><td>2M</td></tr>
<tr><td></td><td>Ans:</td><td>1. Easy to interface with all Microprocessors or works Stand alone.
2. Eight channel 8-bit ADC module.
3. Can measure up to 8 Analog values.
4. On chip Clock not available, external Oscillator is needed (Clock).
5. Digital output various from 0 to 255, operating power is 15mW, conversion time 100us.</td><td>1M each (Any 2 points)</td></tr>
<tr><td></td><td>d)</td><td>List specifications of 8051 microcontroller.</td><td>2M</td></tr>
<tr><td></td><td>Ans:</td><td>1) 8- bit data bus and 8- bit ALU.
2) 16- bit address bus – can access maximum 64KB of RAM and ROM.
3) On- chip RAM -128 bytes (Data Memory)
4) On- chip ROM – 4 KB (Program Memory)</td><td>1M each (Any 2 points)</td></tr>
</table>

		5) Four 8-bit bi- directional input/output ports Four 8-bit bi- directional input/ output ports. 6) Programmable serial ports i.e. One UART (serial port) 7) Two 16- bit timers- Timer 0& Timer 1 8) Works on crystal frequency of 11.0592 MHz 9) Has power down and idle mode in microcontroller when no operation is performed. 10) Six interrupts are available.	
	e)	**List any two instructions which makes accumulator zero individually.**	**2M**
	Ans:	MOV A,#00H CLR A	**1M each**
	f)	**Compare data memory and program memory.**	**2M**
	Ans:	**Sr.No.** / **Program Memory** / **Data Memory** **1** / It is used for storing the hexadecimal codes of the program to be executed i.e. instructions. / It is used for storing temporary variable data and intermediate results. **2** / Program Memory of 8051 is 4kB / Data Memory of 8051 is 128 bytes	**1M each**
	g)	**List SFR in 8051. (any four)**	**2M**
	Ans:	• ACC and B registers – 8 bit each • DPTR : [DPH:DPL] – 16 bit combined • PC : Program Counter – 16 bits • Stack pointer SP – 8 bit • PSW : Program Status Word • Port Latches • Serial data buffer, serial control • Timer Registers (TCON,TMOD,TL0/1,TH0/1) • Power control • Interrupt Enable, Interrupt Priority	**½ M each**

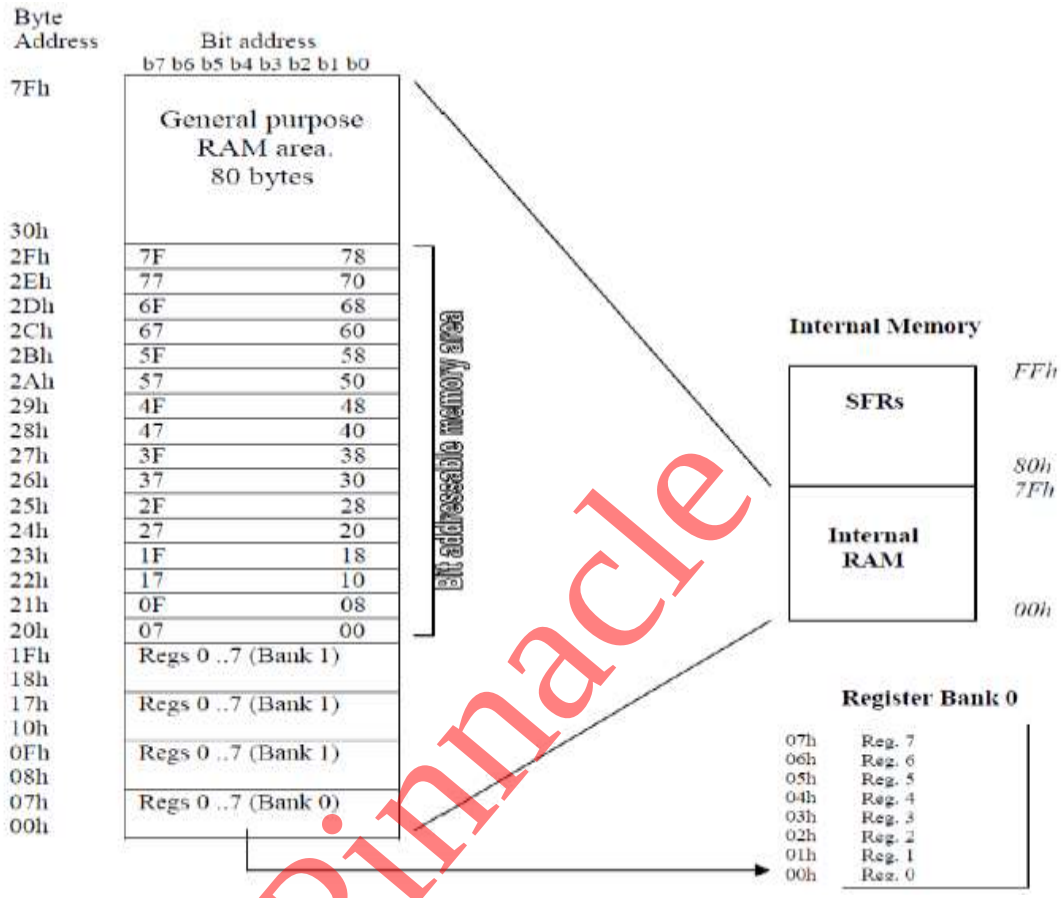
Q.2		**Attempt any THREE of the following:**	**12- Total Marks**
	a)	**Compare any three derivatives of 8051 microcontroller on the basis of RAM,ROM,Timer and Interrupts.**	**4M**
	Ans:	**Features** / **8051** / **8052** / **89c52** / **8031** / **8751** / **89v51 RD2** **RAM** / 128 / 256 / 256 / 128 / 128 / 1k **ROM** / 4K (mask / 8K (EPROM) / 8K (Flash) / 0 / 4K (UV-EPROM) / 64KB (FLASH)	**1M each (Any 4 Points)**

	ROM)					
TIMER	2	3	3	2	2	3
INTERRUPTS	6	8	8	6	6	8

b)	**Draw and explain the interfacing of DAC to 8051.**	**4M**
Ans:	**Diagram:** 	**2M**
	• Microcontroller generates output which is in digital form but many controlling system requires analog signal as they don't accept digital data thus making it necessary to use DAC which converts digital data into equivalent analog voltage. • In the figure shown, we use 8-bit DAC 0808. This IC converts 8 bit digital data into equivalent analog current. Hence we require an I to V converter to convert this current into equivalent voltage.	**2M Expla-nation**
c)	**Describe 8051 microcontroller as boolean processor.**	**4M**
Ans:	• 8051 processor is a CPU that can perform some operation on a data and gives the output. • The 8051 processor contains a complete Boolean processor for single-bit operations. • The internal RAM contains 128 addressable bits, and the SFR space supports up to 128 other addressable bits. • All port lines are bit-addressable, and each can be treated as a separate single-bit port. • The instructions that access these bits are not only conditional branches but also a complete set of move, set, clear, complement, OR, and AND instructions. • The 8051 instruction set is optimized for the one bit operations. The Boolean processor provides direct support for bit manipulation and testing of individual bit allows the use of single bit variable to perform logical operations therefore 8051 can be used to solve Boolean expression. Bits may be set or cleared in a single instruction. • Eg: CLR C means clear the carry bit SETB 20h means set the memory bit with bit address 20h.	**4M**
d)	**Explain function of following pins of 8051** **(i) Pin 31** **(ii) Pin 29** **(iii) Pin 21-28**	**4M**
Ans:	i) **Pin 31-EA :** It is and active low I/P to 8051 microcontroller. When (**EA**)= 0, then 8051 microcontroller access from external program memory (ROM) only. When (**EA**) = 1,	**1M- EA**

		then it access internal and external program memories (ROMS). **ii) Pin 29- PSEN :** This is an output pin. **PSEN** stands for "program store enable." It is active low O/P signal. It is used to enable external program memory (ROM). When [PSEN(bar)]= 0, then external program memory becomes enabled and micro controller read content of external memory location. Therefore it is connected to (OE) of external ROM. **iii) Pin 21-28: $A_8 – A_{15}$:** These pins are known as Port 2. It serves as I/O port. Each pin is bidirectional Input /Output with internal pull – up resistors. Besides the Input /Output, when external memory is interfaced, PORT 2 pins act as the higher-order address bus. (A8-A15)	**1M-PSEN** **2M-Pin 21-28** **1M Port 2 & 1M A8 - A15**
Q.3		**Attempt any THREE of the following:**	**12- Total Marks**
	a)	**Develop Assembly Language program (ALP) to find the largest number in a block of 10 numbers stored at location 40H onwards in internal RAM.**	**4M**

	Ans:	**(NOTE: Marks to be given for any other correct logic used by students.)**
		ORG 0000H
		MOV R1, #0AH ; Initialize Byte Counter
		MOV R0, #40H ; Initialize source pointer R0 to 40H
		DEC R1 ; decrement counter by one
		MOV 60H, @R0 ;Read First Byte
		UP: INC R0 ; Increment the contents of R0
		MOV A, @R0 ; Read second number
		CJNE A, 60H, DN ;compare the first two numbers, if not equal go to DN
		AJMP LARGE ;else go to LARGE
		DN: JC LARGE ;check carry
		MOV 60H, A ;Store largest number to 60H
		LARGE: DJNZ R1, UP ;decrement the counter by one, if count ≠ 0, then go to UP
		END
		Largest No. is saved in memory 60H. Assume any location to store the result.
		OR
		MOV R1, #0AH ; initialize the counter
		MOV R0, #40H ; initialize the memory pointer
		DEC R1 ; decrement counter by one
		MOV A,@R0 ; load number in accumulator
		MOV B, A ; move that number to register B
		UP: INC R0 ; increment the memory pointer
		MOV A,@R0 ; read the next number in A
		CJNE A, B, DOWN ; compare the first two numbers, if not equal go to DOWN
		AJMP NEXT ; else go to NEXT
		DOWN: JC NEXT ; if number in A is greater then go to NEXT
		MOV B, A ; else move the number in register B
		NEXT: DJNZ R1, UP ; decrement the counter by one, if count ≠ 0, then go to UP
		INC R0 ; increment the memory pointer
		MOV A,B
		MOV 50H, A ; store result at memory location 50H(Assume any location)
		HERE: SJMP HERE
			4M for correct program

	b)	**Sketch the internal memory organization in 8051.**	**4M**
	Ans:	**Daigram:**  	**4M for neat Sketch with label**
	c)	**Explain processes of interrupt enabling and disabling in 8051.**	**4M**
	Ans:	Interrupts are the events that temporarily suspend the main program, pass the control to the external sources and execute their task. It then passes the control to the main program where it had left off.8051 has 5 interrupt signals, i.e. INT0, TF0, INT1, TF1, RI/TI. Each interrupt can be enabled or disabled by setting bits of the IE register and the whole interrupt system can be disabled by clearing the EA bit of the same register. **IE (Interrupt Enable) Register:** This register is responsible for enabling and disabling the interrupt. EA bit is set to 1 for enabling interrupts and set to 0 for disabling the interrupts. Its bit sequence and their meanings are shown in the following figure. EA | _ | _ | ES | ET1 | EX1 | ET0 | EX0	**2M format** **2M function of each**

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

bit

d) **Explain following instructions of 8051.**
(i) ADDC
(ii) L CALL — **4M**

Ans:

(i) ADDC: The **ADDC** instruction adds a byte value and the value of the carry flag to the accumulator. The results of the addition are stored back in the accumulator. Several of the flag registers are affected.

ADDC
Function: Add with Carry
Syntax: ADDC A, source byte
Flags affected: OV,AC,CY
Description: ADDC simultaneously adds the byte variable indicated, the carry flag and the Accumulator contents, leaving the result in the Accumulator (A = A + byte +CY). The carry and auxiliary-carry or bit flags are set, respectively. If CY = 1 prior to this instruction, CY is also added to A.
Addressing modes supported for ADDC instruction :

- Immediate: ADDC A,#data
- Register: ADDC A, Rn
- Direct: ADDC A, address
- Register Indirect: ADDC A, @Ri

(ii) LCALL

Function: Long call, Transfers control to a subroutine
Syntax: LCALL 16 bit addr
Flags affected : None
No. of bytes used: 3 byte(1 byte is opcode and other two bytes are the 16 bit address of the target subroutine)
Description: This instruction is used to transfers control to a subroutine To reach the target address in the 64 Kbytes maximum ROM space of the 8051, LCALL instruction is used. For calling a subroutine, the PC register (which has the address of the instruction after the LCALL) is pushed onto the stack, and the stack pointer (SP) is incremented by 2. Then the program counter is loaded with the new address and control is transferred to the subroutine.

2M each instruction

Q.4		**Attempt any THREE of the following :**	**12 Marks**
	a)	**Draw the format of TCON register of 8051 and describe the function of each bit of it.**	**4M**
	Ans:	**TCON: TIMER/COUNTER CONTROL REGISTER.BIT ADDRESSABLE** **TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0** TF1 TCON. 7 Timer 1 overflows flag. Set by hardware when the Timer/Counter 1 Overflows. Cleared by hardware as processor vectors to the interrupt Service routine. TR1 TCON. 6 Timer 1 run control bit. Set/cleared by software to turn Timer/Counter1 ON/OFF. TF0 TCON. 5 Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 Overflows. Cleared by hardware as processor vectors to the service routine. TR0 TCON. 4 Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF. IE1 TCON. 3 External Interrupt 1 edge flag. Set by hardware when External Interrupt edge is detected. Cleared by hardware when interrupt is processed. IT1 TCON. 2 Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt. IE0 TCON. 1 External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed. IT0 TCON. 0 Interrupt 0 type control bit. Set/cleared by software to Specify falling edge/low level triggered External Interrupt	**2M format** **2M Function of each bit**
	b)	**Describe serial communication in 8051. Explain the use of SCON register.**	**4M**
	Ans:	8051 micro controller communicate with another peripheral device through RXD and TXD pin of port3.controller have four mode of serial communication. **1. Serial Data Mode-0 (Baud Rate Fixed)** In this mode, the serial port works like a shift register and the data transmission works synchronously with a clock frequency of fosc /12. Serial data is received and transmitted through RXD. 8 bits are transmitted/ received at a time. Pin TXD outputs the shift clock pulses of frequency fosc /12, which is connected to the external circuitry for synchronization. The shift frequency or baud rate is always 1/12 of the oscillator frequency. **2. Serial Data Mode-1 (standard UART mode)(baud rate is variable)** In mode-1, the serial port functions as a standard Universal Asynchronous Receiver Transmitter (UART) mode. 10 bits are transmitted through TXD or received through RXD. The 10 bits consist of one start bit (which is usually '0'), 8 data bits (LSB is sent first/received first), and a stop bit (which is usually '1'). Once received, the stop bit goes into RB8 in the special function register SCON. The **baud rate is variable** **3. Serial Data Mode-2 Multiprocessor (baud rate is fixed)** In this mode 11 bits are transmitted through TXD or received through RXD. The various bits are as follows: a start bit (usually '0'), 8 data bits (LSB first), a programmable 9 th (TB8 or RB8)bit and a stop bit (usually '1'). While transmitting, the 9 th data bit (TB8 in SCON) can be assigned the value '0' or '1'. For example, if the information of parity is to be transmitted, the parity bit (P) in PSW could be moved into TB8.On reception of the data, the 9 th bit goes into RB8 in 'SCON',	**2M mode description in short** **(½ mark for each mode)** **& 2M format with function**

while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency.

f baud = (2 SMOD /64) fosc

4. Serial Data Mode-3 - Multi processor mode(Variable baud rate)

In this mode 11 bits are transmitted through TXD or received through RXD.The various bits are: a start bit (usually '0'), 8 data bits (LSB first), a programmable 9 th bit and a stop bit (usually '1'). Mode-3 is same as mode-2, except the fact that the baud rate in mode-3 is variable (i.e., just as in mode-1).

$f_{baud} = (2^{SMOD}/32) * (f_{osc}/ 12 (256-TH1))$

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0 SCON.7 Serial port mode specifier

SM1 SCON.6 Serial port mode specifier.

SM0 SM1

0 0 Serial Mode 0

0 1 Serial Mode 1, 8-bit data, 1 stop bit, 1 start bit

1 0 Serial Mode 2

1 1 Serial Mode 3

SM2 SCON.5 Used for multiprocessor communication

REN SCON.4 Set/ cleared by software to enable/ disable reception.

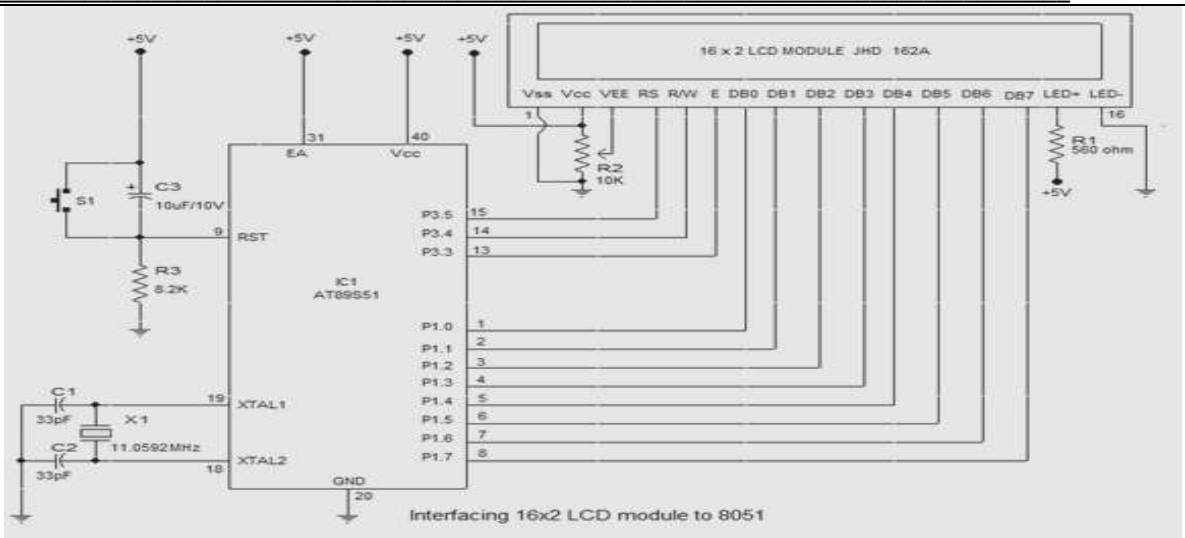
TB8 SCON.3 – the 9th bit that will be transmitted in mode 2/3 set/clear by software.

RB8 SCON.2– in mode 2/3 it is the 9th bit that was received .

TI SCON.1 Transmit interrupt flag. Set by hardware at the beginning of the stop Bit in mode 1.

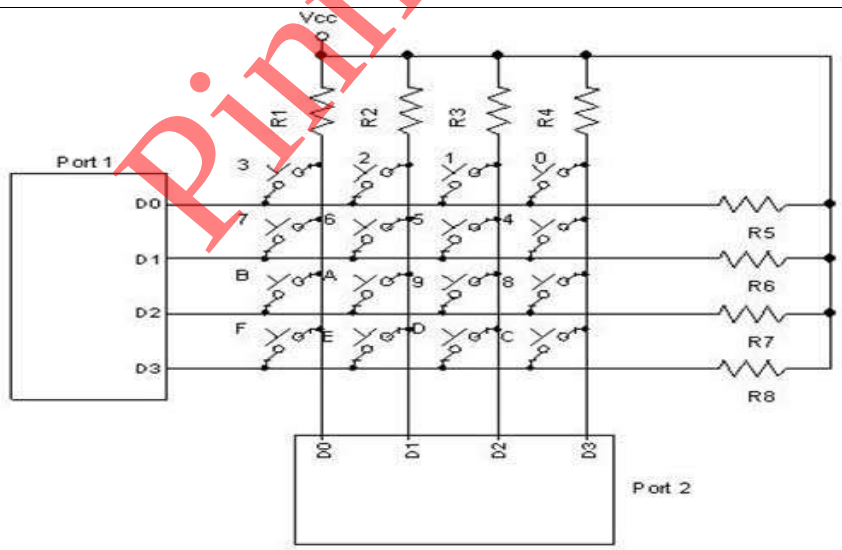
RI SCON.0 Receive interrupt flag. Set by hardware halfway through the stop bit time in mode 1.

c)	**Draw interfacing of 16 × 2 LCD with 8051 and state the function of EN and RS of LCD**	**4M**
Ans:	**Diagram:**	**2M for diagram** **2Marks for**

		 Interfacing 16x2 LCD module to 8051 **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 instructions during the initializing sequence like clear the screen etc. **EN:** The enable pin is used by the LCD to latch information presented to its data pins. When data is supplied to the data pins, a high-to-low pulse must be applied to this pin in order for the LCD to latch in the data present at the data pins. This pulse must be a minimum of **450ns** wide.	**function of two pins(1Mark each pin function)**
	d)	**Explain the use of following assembler directives.** **(i) EQU** **(ii) ORG**	**4M**
	Ans:	**(i) EQU**: Equate It is used to define constant without occupying a memory location. Syntax: Label EQU Numeric value By means of this directive, a numeric value is replaced by a symbol. For e.g. MAXIMUM EQU 99 After this directive every appearance of the label MAXIMUM in the program, the assembler will interpret as number 99 (MAXIMUM=99). **(ii) ORG:-**ORG stands for Origin Syntax: ORG Address The ORG directive is used to indicate the beginning of the address.The *origin directive* tells the assembler where to load instructions and data into memory. It changes the program counter to the value specified by the expression in the operand field. The number thatcomes after ORG can be either in hex or in decimal. If the number is notfollowed by H, it is decimal and the assembler will convert it to hex.	**2 Marks for each directive**
	e)	**State the alternate pin functions of port 3 of 8051.**	**4M**

	Ans:		**4 Marks for 8 pins(½ mark for each pin function)**

Pin	Name	Alternate Function
P3.0	RXD	Serial input line
P3.1	**TXD**	Serial output line
P3.2	$\overline{INT0}$	External interrupt 0
P3.3	$\overline{INT1}$	External interrupt 1
P3.4	**T0**	Timer0 external input
P3.5	**T1**	Timer1 external input
P3.6	$\overline{WR}$	External data memory write strobe
P3.7	$\overline{RD}$	External data memory read strobe

Q.5		**Attempt any TWO of the following**	**12 Total Marks**
	(a)	**Explain with sketch the interfacing of 4 ×4 matrix keypad with 8051 microcontroller.**	**6M**
	Ans:		**sketch -3M**

Interfacing keypad

Fig. shows how to interface the **4 X 4 matrix keypad** to two ports in **microcontroller**. The rows are connected to an output port and the columns are connected to an input port.

To detect a pressed key, the **microcontroller** grounds all rows by providing 0 to the output latch, and then it reads the columns. If the data read from the columns is D3-D0=1111, no key has been pressed and the process continues until a key press is detected. However, if one of the

Explanation – 3M

column bits has a zero, this means that a key press has occurred. For example, if D3-D0=1101, this means that a key in the D1 column has been pressed.

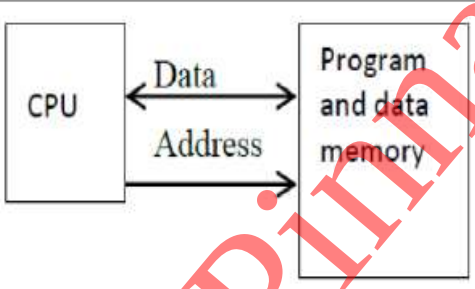
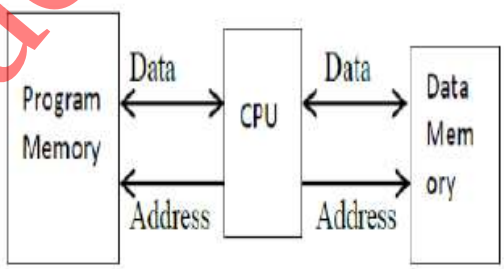
After a key press is detected, the **microcontroller** will go through the process of identifying the key. Starting with the top row, the **microcontroller** grounds it by providing a low to row D0 only; then it reads the columns.

If the data read is all 1s, no key in that row is activated and the process is moved to the next row. It grounds the next row, reads the columns, and checks for any zero. This process continues until the row is identified. After identification of the row in which the key has been pressed, the next task is to find out which column the pressed key belongs to.

(b) **Differentiate between**
(i) Harvard and Von-neuman architecture
(ii) Microprocessor and Microcontroller **6M**

Ans:

i) Harvard Architecture and Von-neuman architecture

Sr.No	Von Neumann architecture	Harvard architecture
1		
2	The Van Neumann architecture uses single memory for their instructions and data.	The Harvard architecture uses physically separate memories for their instructions and data.
3	Requires single bus for instructions and data	Requires separate & dedicated buses for memories for instructions and data.
4	Its design is simpler	Its design is complicated
5	Instructions and data have to be fetched in sequential order limiting the operation bandwidth.	Instructions and data can be fetched simultaneously as there is separate buses for instruction and data which increasing operation bandwidth.
6	Program segments & memory blocks for data & stacks have separate sets of addresses.	Vectors & pointers, variables program segments & memory blocks for data & stacks have different addresses in the program.

Von Neumann Harvard 3 M (any three points)

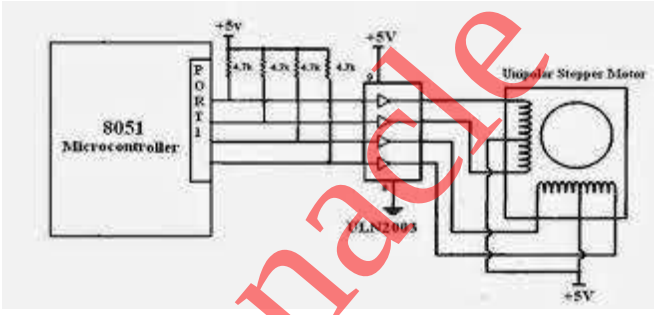
ii) Microprocessor and Microcontroller

Micro proces

			sor , Micro controller – 3M (any three points)

Sr. No	Parameter	Microprocessor	Microcontroller
1.	No. of instructions used	Many instructions to read/ write data to/ from external memory.	Few instruction to read/ write data to/ from external memory
2.	Memory	Do not have inbuilt RAM or ROM.	Inbuilt RAM /or ROM
3.	Registers	Microprocessor contains general purpose registers, Stack pointer register, Program counter register	Microcontroller contains general purpose registers, Stack pointer register, Program counter register additional to that it contains Special Function Registers (SFRs) for Timer , Interrupt and serial communication etc.
4.	Timer	Do not have inbuilt Timer.	Inbuilt Timer
5.	I/O ports	I/O ports are not available requires extra device like 8155 or 8255.	I/O ports are available
6.	Serial port	Do not have inbuilt serial port, requires extra devices like 8250 or 8251.	Inbuilt serial port
7.	Multifunction pins	Less Multifunction pins on IC.	Many multifunction pins on the IC
8.	Boolean Operation	Boolean operation is not possible directly.	Boolean Operation i.e. operation on individual bit is possible directly
9.	Applications	General purpose, Computers and Personal Uses.	Single purpose(dedicated application), Automobile companies, embedded systems, remote control devices.

	(c)	**Develop an ALP to generate square wave of 3 KHz using 8051 microcontroller on port pin P2.3 (Assume X_{tal}freq^n=12 MHz)**	**6M**
	Ans:	Crystal frequency= 12 MHz I/P clock = (12*10^6)/12= 1 MHz T_{in} = 1µ sec For 3 kHz square wave F_{out} = 3 KHz T_{out} = 1/ (3X 10^3) = 0.3msec =333 µ sec So T_{ON} = T_{OFF} = 333/2 = 166.5 µ sec N = T_{ON} / T_{in} = 166.5 µ sec /1 µ sec = 166.5 167 65535 – 167+1 = $(65369)_{10}$ = $(FB71)_{16}$ Program:- MOV TMOD, # 01H ; Set timer 0 in Mode 1, i.e., 16 bit timer	**Count calculation – 2M, Correct program – 4M**

		L2: MOV TL0, # 71 H ; Load TL register with LSB of count MOV TH0, # 0FB H ; load TH register with MSB of count SETB TR0 ; start timer 0 L1: JNB TF0, L1 ; poll till timer roll over CLR TR0 ; stop timer 0 CPL P2.3 ; complement port 2.3 line to get high or low CLR TF0 ; clear timer flag 0 SJMP L2 ; re-load timer with count as mode 1 is not auto reload	
Q.6		**Attempt any TWO of the following:**	**12Total Marks**
	(a)	**Draw interfacing of stepper motor with 8051 and write an ALP to rotate it in clockwise direction.**	**6M**
	Ans:	**Diagram:**  	**3M**

Step no	Winding A	Winding B	Winding C	Winding D	Clockwise
1	1	0	0	1	↓
2	1	1	0	0	
3	0	1	1	0	
4	0	0	1	1	

Program-

```
            MOV  A,#66H     ;load step sequence
BACK:       MOV  P1,A       ;issue sequence to motor
            RR   A          ;rotate right clockwise
            ACALL DELAY     ;wait
            SJMP BACK       ;keep going

            ...
DELAY
            MOV  R2,#100
H1:         MOV  R3,#255
H2:         DJNZ R3,H2
            DJNZ R2,H1
            RET
```

(Other programs with similar logic can be given marks)

3M

(b) **Describe with sketches the procedure to troubleshoot the traffic light controller.** **6M**

Ans:

Considerations of Traffic Signal

1) Traffic light may have sensors integrated to provide real time traffic information
2) Based on the traffic information provided by the sensor, the duration of the green/Red LED light for each direction may vary so that the traffic for both the directions are roughly balanced. Time left for the green light should be displayed
3) When the traffic light for one signal is green, then the traffic for the other directions should be red (with duration displayed in red)
4) The red light will be switched to yellow when the timer value is 5 sec before switching to red.

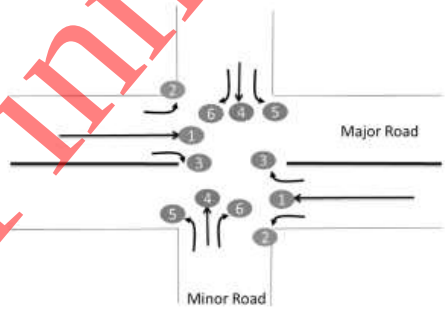
5) Violations happen when user expectancy is not met. A user like pedestrian does not expect to stand for more than a minute or two at a signal, when this user expectancy is not met, the pedestrian tries to venture out and violate the signal
6) The smooth movement of conflicting vehicles is determined by the availability of gaps in traffic. This is true for both pedestrians and vehicular traffic. Understanding of gaps is important for justifying the type of traffic control device, including a traffic signal.

Points to consider for determining signal timings

1) The signal operational parameters are reviewed and updated (if needed) on a regular basis to maximize the ability of the traffic control signal to satisfy current traffic demands

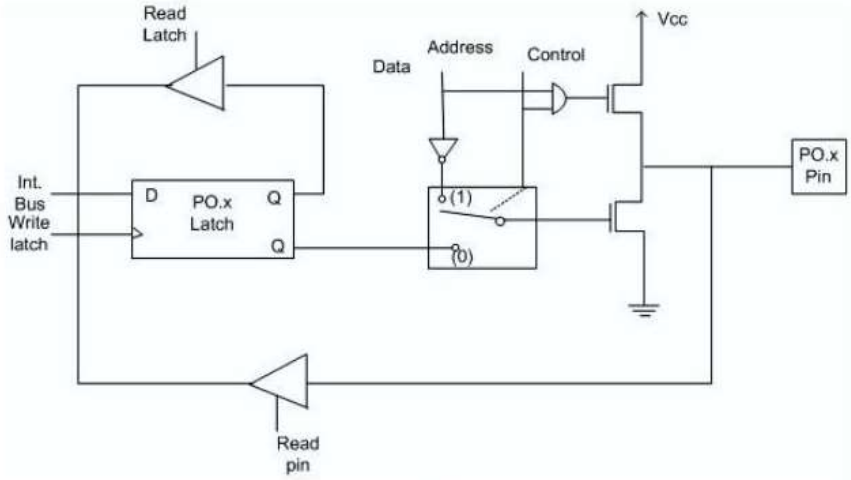
Any other correct trouble shooting procedure may be given marks

2) Geometry of the intersection is the next step in the signal timing process. Determining the lane use (which traffic mode), dedicated vs. shared lane, type of roads interacting (arterial with arterial etc.), type of road infrastructure (ramps, one way streets, etc.) will impact the timing.
3) Basic signal timing parameters comes next. Pedestrian walk times, flashing don't walk, yellow time, all red clearance interval, detector gap times all need to calculated or established.
4) Identify bottlenecks, review conditions, conduct warrant analysis and use engineering judgment in determining traffic signal installation
5) Determine AM and PM peak hour traffic volumes.
6) Condition diagram which includes roadway geometrics, parking, driveways, sidewalks, signing, pavement markings, development of intersection quadrants, and any other features pertinent to the study peak hour delay study
7) A conflict analysis
8) Capacity analysis of the intersection for current and future years using growth



	(c)	**Draw and explain Internal port structure of Port 0 and Port 1 of 8051 microcontroller.**	**6M**

Ans:



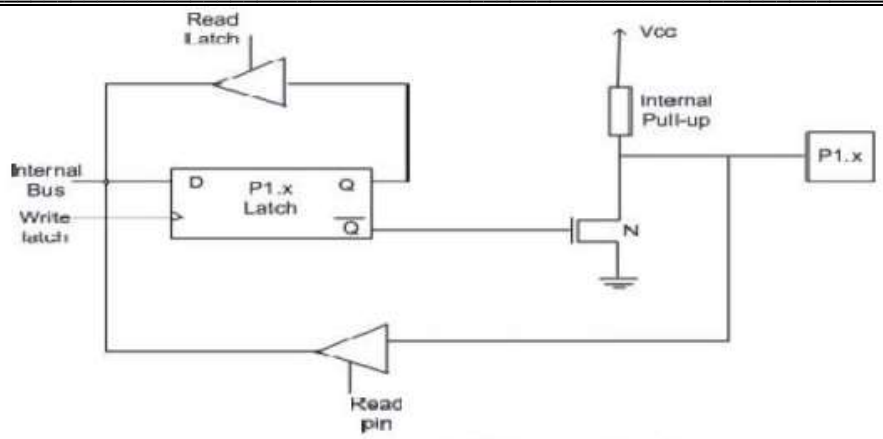
Port-0 Structure

Port-0 can be configured as a normal bidirectional I/O port or it can be used for address/data interfacing for accessing external memory. When control is '1', the port is used for address/data interfacing. When the control is '0', the port can be used as a normal bidirectional I/O port. **Port 0 – 3M**

Let us assume that control is '0'. When the port is used as an input port, '1' is written to the latch. In this situation both the output MOSFETs are 'off'. Hence the output pin floats. This high impedance pin can be pulled up or low by an external source. When the port is used as an output port, a '1' written to the latch again turns 'off' both the output MOSFETs and causes the output pin to float. An external pull-up is required to output a '1'. But when '0' is written to the latch, the pin is pulled down by the lower MOSFET. Hence the output becomes zero. **Port 1 – 3M**

When the control is '1', address/data bus controls the output driver MOSFETs. If the address/data bus (internal) is '0', the upper MOSFET is 'off' and the lower MOSFET is 'on'. The output becomes '0'. If the address/data bus is '1', the upper transistor is 'on' and the lower transistor is 'off'. Hence the output is '1'. Hence for normal address/data interfacing (for external memory access) no pull-up resistors are required.

Port-0 latch is written to with 1's when used for external memory access.



Port 1 Structure

Port-1 does not have any alternate function i.e. it is dedicated solely for I/O interfacing. When used as output port, the pin is pulled up or down through internal pull-up. To use port-1 as input port, '1' has to be written to the latch. In this 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 wrong reading.