Microprocessor Architecture (April 2019) Solution Set

1 Attempt any three of the following:

a. Explain different constituents of microprocessor system. Draw a neat diagram showing microprocessor based system with bus architecture.

The microprocessor is a semiconductor device (Integrated Circuit) manufactured by the VLSI (Very Large Scale Integration) technique. It includes the ALU, register arrays and control circuit on a single chip. To perform a function or useful task we have to form a system by using microprocessor as a CPU and interfacing memory, input and output devices to it. A system designed using a microprocessor as its CPU is called a microcomputer. The Microprocessor based system (single board microcomputer) consists of microprocessor as CPU, semiconductor memories like EPROM and RAM, input device, output device and interfacing devices. The memories, input device, output device and interfacing devices are called peripherals. The popular input devices are keyboard and floppy disk and the output devices are printer, LED/LCD displays, CRT monitor, etc.

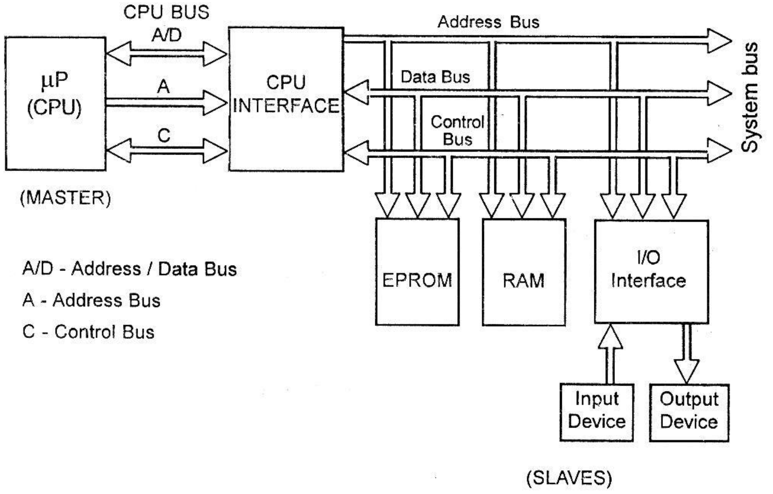
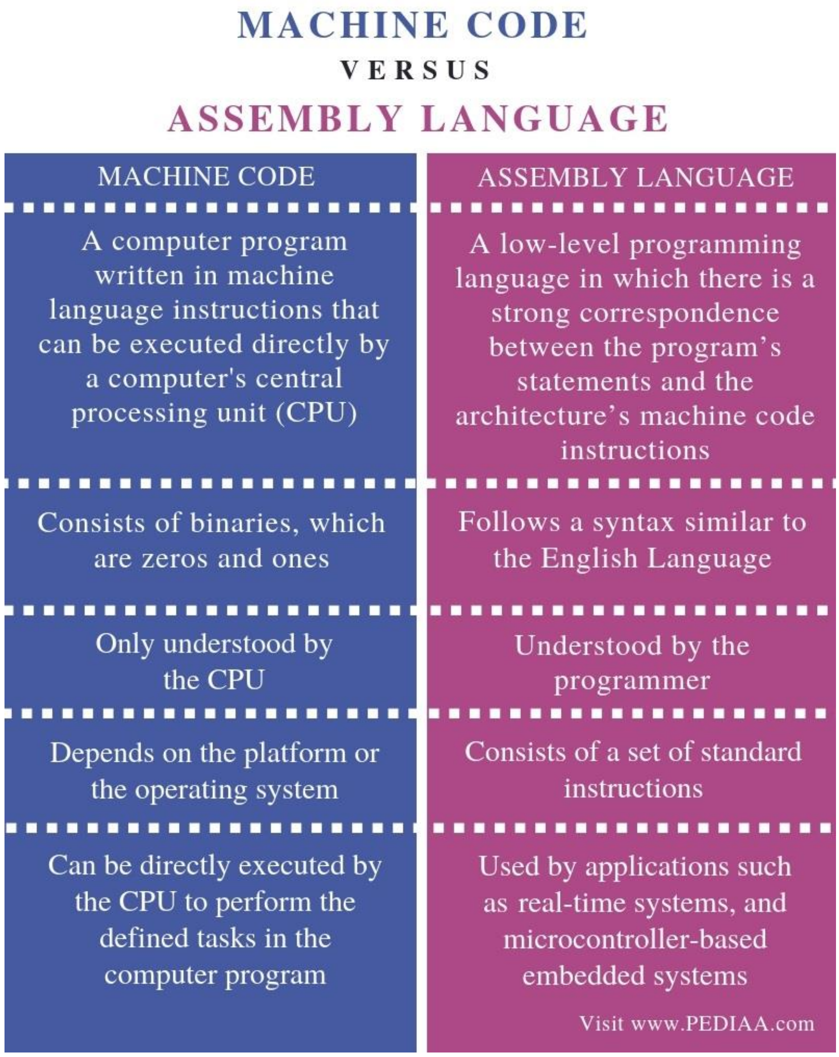


Fig : 1.1 Microprocessor Based System (organisation of microcomputer)

b. Explain the difference between 8085 machine language and 8085 assembly language.



MACHINE CODE VERSUS ASSEMBLY LANGUAGE

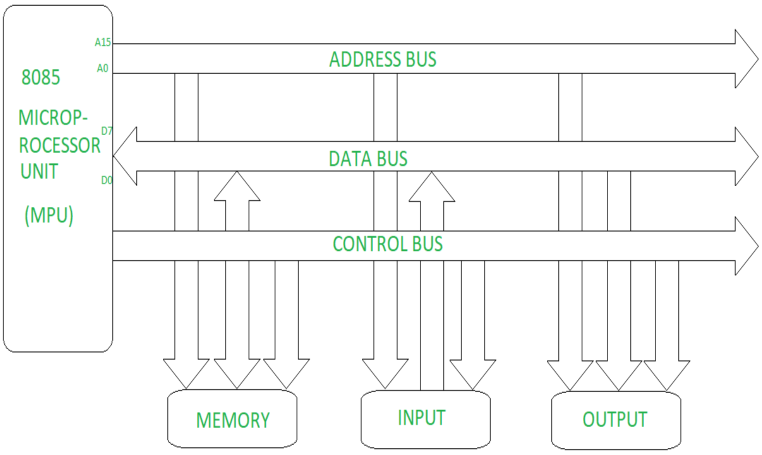
MACHINE CODE	ASSEMBLY LANGUAGE
A computer program written in machine language instructions that can be executed directly by a computer's central processing unit (CPU)	A low-level programming language in which there is a strong correspondence between the program's statements and the architecture's machine code instructions
Consists of binaries, which are zeros and ones	Follows a syntax similar to the English Language
Only understood by the CPU	Understood by the programmer
Depends on the platform or the operating system	Consists of a set of standard instructions
Can be directly executed by the CPU to perform the defined tasks in the computer program	Used by applications such as real-time systems, and microcontroller-based embedded systems

Visit www.PEDIAA.com

c. With neat labelled diagram explain how 8085 system bus is divided into three different sets of communication lines.

Bus is a group of conducting wires which carries information, all the peripherals are connected to microprocessor through Bus.

Diagram to represent bus organization system of 8085 Microprocessor.



Bus organization system of 8085 Microprocessor

There are three types of buses.

Address bus –

It is a group of conducting wires which carries address only.Address bus is unidirectional because data flow in one direction, from microprocessor to memory or from microprocessor to Input/output devices (That is, Out of Microprocessor).

Length of Address Bus of 8085 microprocessor is 16 Bit (That is, Four Hexadecimal Digits), ranging from 0000 H to FFFF H, (H denotes Hexadecimal). The microprocessor 8085 can transfer maximum 16 bit address which means it can address 65, 536 different memory location.

The Length of the address bus determines the amount of memory a system can address.Such as a system with a 32-bit address bus can address 2^32 memory locations.If each memory location holds one byte, the addressable memory space is 4 GB.However, the actual amount of memory that can be accessed is usually much less than this theoretical limit due to chipset and motherboard limitations.

Data bus –

It is a group of conducting wires which carries Data only.Data bus is bidirectional because data flow in both directions, from microprocessor to memory or Input/Output devices and from memory or Input/Output devices to microprocessor.

Length of Data Bus of 8085 microprocessor is 8 Bit (That is, two Hexadecimal Digits), ranging from 00 H to FF H. (H denotes Hexadecimal).

When it is write operation, the processor will put the data (to be written) on the data bus, when it is read operation, the memory controller will get the data from specific memory block and put it into the data bus.

The width of the data bus is directly related to the largest number that the bus can carry, such as an 8 bit bus can represent 2 to the power of 8 unique values, this equates to the number 0 to 255.A 16 bit bus can carry 0 to 65535.

Control bus –

It is a group of conducting wires, which is used to generate timing and control signals to control all the associated peripherals, microprocessor uses control bus to process data, that is what to do with selected memory location. Some control signals are:

Memory read
Memory write
I/O read
I/O Write

Opcode fetch
If one line of control bus may be the read/write line.If the wire is low (no electricity flowing) then the memory is read, if the wire is high (electricity is flowing) then the memory is written.

d. Illustrate the memory address range of a memory chip with 256 bytes of memory. Draw a neat diagram to show the memory map and explain how this memory chip is accessed by 8085 microprocessor.

Memory interfacing is used to provide more memory space to accommodate complex programs for more complicated systems. Types of memories which are most commonly used to interface with 8085 are RAM, ROM, and EEPROM. 8085 can access 64kB of external memory. It can be explained as- total number of address lines in 8085 are 16, therefore it can access 2^16 = 65535 locations i.e. 64kB.

Note: *2^n=number of memory locations. Where, n = number of address lines*

Some of the RAM IC's are given as:
1. IC 2114 -> 1k x 4bits
2. IC 6116 -> 2k x 8bits
3. IC 6264 -> 8k x 8bits

Some of the ROM IC's are given as:
1. IC 2708 -> 1k x 8bits
2. IC 2716 -> 2k x 8bits
3. IC 2732 -> 4k x 8bits
4. IC 2764 -> 8k x 8bits
5. IC 27128 -> 16k x 8bits
6. IC 27256 -> 32k x 8bits
7. IC 2708 -> 64k x 8bits

e. Explain how lower order data and address bus of 8085 microprocessor are multiplexed.

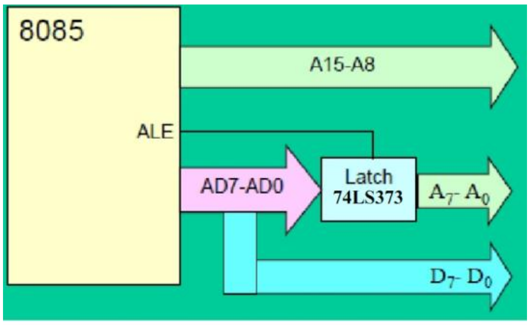


Fig: Demultiplexing of address data pins of 8085

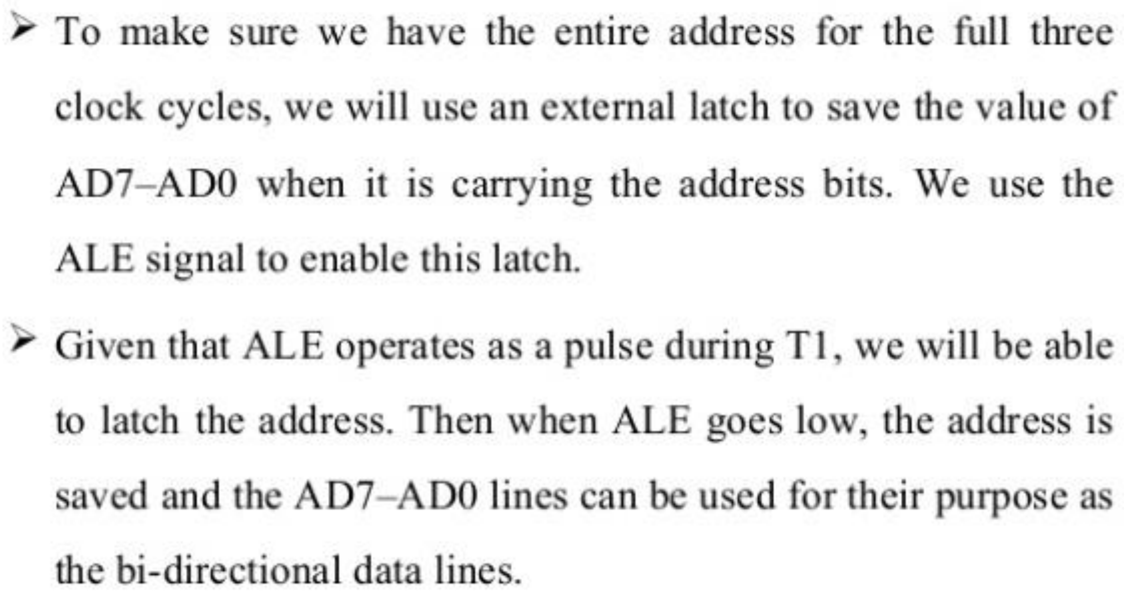
- The address bus has 8 signal lines A8 – A15 which are unidirectional.

- The other 8 address bits are multiplexed(time shared) with the 8 data bits. So, the bits AD0 –AD7are bi-directional and serve as A0 –A7and D0 –D7at the same time. During the execution of the instruction, these lines carry the address bits during the early part, then during the late parts of the execution, they carry the 8 data bits.

- In order to separate the address from the data, we can use a latch to save the value before the function of the bits changes.

- From the above description, it becomes obvious that the AD7–AD0 lines are serving a dual purpose and that they need to be demultiplexed to get all the information.

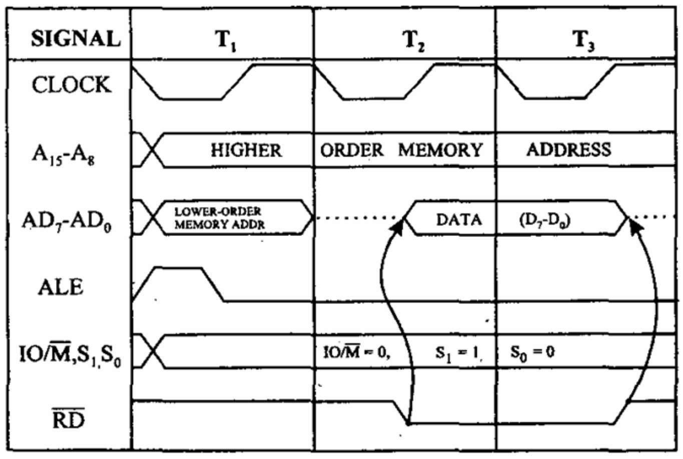
- The high order bits of the address remain on the bus for three clock periods. However, the low order bits remain for only one clock period and they would be lost if they are not saved externally. Also, notice that the low order bits of the address disappear when they are needed most.



- To make sure we have the entire address for the full three clock cycles, we will use an external latch to save the value of AD7–AD0 when it is carrying the address bits. We use the ALE signal to enable this latch.
- Given that ALE operates as a pulse during T1, we will be able to latch the address. Then when ALE goes low, the address is saved and the AD7–AD0 lines can be used for their purpose as the bi-directional data lines.

f. With proper timing diagram explain memory read cycle of 8085 microprocessor.
Memory Read Machine Cycle of 8085:

The memory read machine cycle is executed by the processor to read a data byte from memory. The processor takes 3T states to execute this cycle.
The instructions which have more than one byte word size will use the machine cycle after the opcode fetch machine cycle.



2 Attempt any three of the following:

a Explain how eight DIP switches are interfaced with 8085 microprocessor using a decoder.
.

Below figure gives the decoding circuitry for inputting 8-bit data from 8-toggle switches isolated I/O. PORT address allocated 16 AB 8 BDB D7 D0 IPTS A0 +5V 1 k SW7 SW0 RD IO/M BCB A15 A14 A13 A12 being A0H to AFH with A0H as primary address and rest as fold back addresses. In the design, we have used 8205 (3 line to 8 line decoder). It could be 74LS138 another 3 line to 8 line decoder.

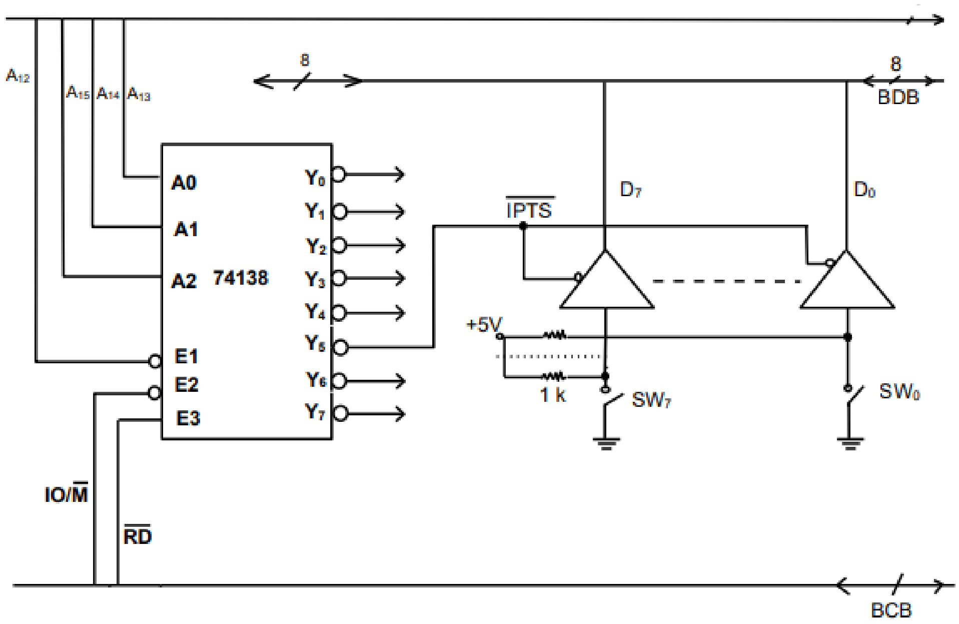


Fig.7.10 Interfacing of 8-Toggle Switches Using 3 Line to 8Line Decoder

b. How is testing and troubleshooting of I/O interfacing circuit is done?

There are two types of interfacing in context of the 8085 processor.
Memory Interfacing.
I/O Interfacing.

Memory Interfacing:

While executing an instruction, there is a necessity for the microprocessor to access memory frequently for reading various instruction codes and data stored in the memory. The interfacing circuit aids in accessing the memory.

Memory requires some signals to read from and write to registers. Similarly the microprocessor transmits some signals for reading or writing a data.

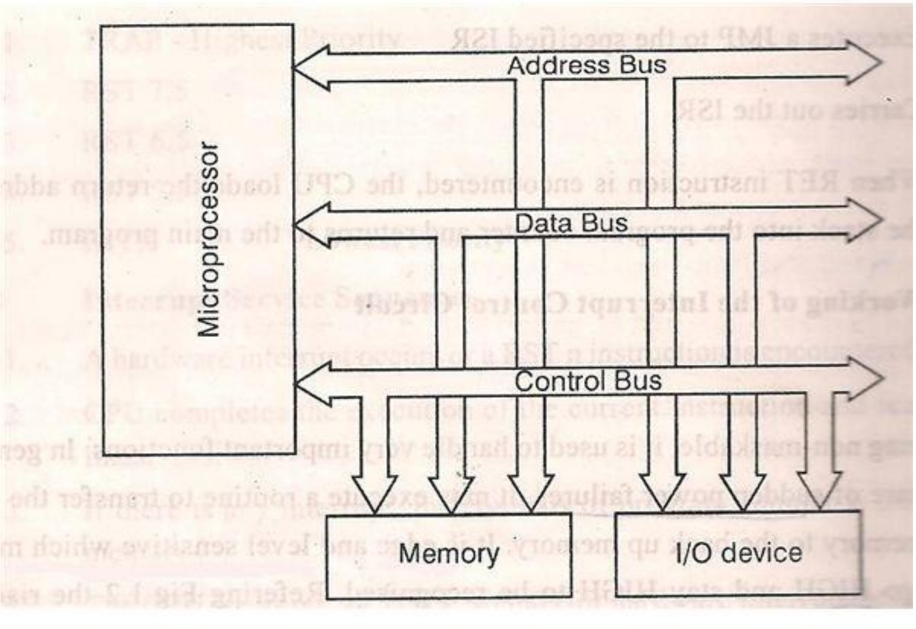
But what is the purpose of interfacing circuit here?

The interfacing process involves matching the memory requirements with the microprocessor signals. The interfacing circuit therefore should be designed in such a way that it matches the memory signal requirements with the signals of the microprocessor. For example for carrying out a READ process, the microprocessor should initiate a read signal which the memory requires to read a data. In simple words, the primary function of a memory interfacing circuit is to aid the microprocessor in reading and writing a data to the given register of a memory chip.

I/O Interfacing:

We know that keyboard and Displays are used as communication channel with outside world. So it is necessary that we interface keyboard and displays with the microprocessor. This is called I/O interfacing. In this type of interfacing we use latches and buffers for interfacing the keyboards and displays with the microprocessor.

But the main disadvantage with this interfacing is that the microprocessor can perform only one function. It functions as an input device if it is connected to buffer and as an output device if it is connected to latch. Thus the capability is very limited in this type of interfacing.



c. Discuss in brief the programming model of 8085 microprocessor.

PROGRAMMING MODEL OF 8085 MICROPROCESSOR

accumulator	flag
B	C
D	E
H	L
STACK POINTER	
PROGRAM COUNTER	

IN 8085 microprocessor there are six 8 bit register (B,C,D,E,H,L)
,one accumulator ,one flag register
and two 16 bit registers that is stack pointer and program counter.

THE programming model contain 6 general purpose register(B,C,D,E,H,L), and this register can perform 16 bit operation by combining as BC,DE,HL.

Here the work of program counter is to store or copy data into register by using copy data instruction.

Accumulator -THE accumulator is a 8 bit register which is a part of arithmetic and logical unit(ALU).we use this register to store 8 bit data and perform arithmetic and logical operation on it.

Stack Pointer-THE use of stack pointer is to locate the top most location of the of stack or memory.

Program Counter-THE funtion of programe counter is to point the the memory address from which the next byte is to be fetched.

FLAG-THE most commonly used flags are zero,carry,sign.flag register always be either zero or one.

e. Explain any one arithmetic and any one logical group one byte instruction from the instruction set of 8085 microprocessor.

CMP: - (compare register or memory with accumulator) The contents of the operand register or memory are M compared with the contents of the accumulator. Both contents are preserved . The result of the comparison is shown by setting the flags of the PSW as follows:

if (A) < reg/mem: carry flag is set.
if (A) = reg/mem:zero flag is set.
if (A) > reg/mem: carry and zero flags are reset.

Eg: - CMP B
CMP M

ANI: - (AND operation with immediate data) AND operation is performed between accumulator and specified immediate data.

Eg: - ANI 30H

f. Write an assembly language program to add two 8 bit numbers stored at memory locations D2000 H and D3000 H. Store the answer at memory location D400H.

MEMORY ADDRESS	MNEMONICS	COMMENT
2000	LDA D200	A<-[d200]
2003	MOV H, A	H<-A
2004	LDA D300	A<-[D300]
2007	ADD H	A<-A+H
2006	MOV L, A	L←A

2007	MVI A 00	A←00
2009	ADC A	A←A+A+carry
200A	MOV H, A	H←A
200B	SHLD D400	H→D401, L→D400
200E	HLT	

3 Attempt any three of the following:

a. What are different available conditional loops in the assembly language programming for 8085?

LOOPING:

The programming technique used to instruct the microprocessor to repeat tasks is called looping.

This task is accomplished by using jump instructions.

CLASSIFICATION OF LOOPS:

1.continuous loop

2.Unconditional loop

CONTINUOUS LOOP:

repeats a task continuously.

A continuous loop is set up by using the unconditional jump instruction

A program with a continuous loop does not stop repeating the tasks until the system is reset.

CONDITIONAL LOOP:

A conditional loop is set up by a conditional jump instructions.

These instructions check flags(Z,CY,P,S) and repeat the tasks if the conditions are satisfied.

These loops include counting and indexing.

CONDITIONAL LOOP AND COUNTER:

A counter is a typical application of the conditional loop.

A microprocessor needs a counter,flag to accomplish the looping task.

Counter is set up by loading an appropriate count in a register.

Counting is performed by either increment or decrement the counter.

Loop is set up by a conditional jump instruction.

End of counting is indicated by a flag.

CONDITIONAL LOOP,COUNTER AND INDEXING:

Another type of loop which includes counter and indexing .

INDEXING:

pointing of referencing objects with sequential numbers.

Data bytes are stored in memory locations and those data bytes are referred to by their memory locations.

Example:

Steps to add ten bytes of data stored in memory locations starting ata given location and display the sum.

The microprocessor needs

a counter to count 10 data bytes.

an index or a memory pointer to locate where data bytes are stored.

to transfer data from a memory location to the microprocessor(ALU)
to perform addition
registers for temporary storage of partial answers
a flag to indicate the completion of the stack
to store or output the result.

b. Explain following logical instructions-

i) RAL
ii) RAR

Rotate accumulator left through carry (RAL) –

In this instruction, each bit is shifted to the adjacent left position. Bit D7 becomes the carry bit and the carry bit is shifted into D0. Carry flag CY is modified according to the bit D7. For example:
A = D7 D6 D5 D4 D3 D2 D2 D0

//before the instruction
A = 10101010; CY=0

//after 1st RAL
A = 01010100; CY=1

//after 2nd RAL
A = 10101001; CY=0

Rotate accumulator right through carry (RAR) –

In this instruction, each bit is shifted to the adjacent right position. Bit D0 becomes the carry bit and the carry bit is shifted into D7. Carry flag CY is modified according to the bit D0. For example:
A = D7 D6 D5 D4 D3 D2 D2 D0

//before the instruction
A = 10000001; CY=0

//after 1st RAR
A = 01000000; CY=1

//after 2nd RAR
A = 10100000; CY=0

c. What is time delay? Why is time delay needed in a program? What are different ways of generating a time delay in an assembly language program for 8085 microprocessor?

Time delay: A register is loaded with a number , depending on the time delay required and then the register is decremented until it reaches zero by setting up a loop with conditional jump instruction.

When the delay subroutine is executed, the microprocessor does not execute other tasks. For the delay we are using the instruction execution times. executing some instructions in a loop, the delay is generated. There are some methods of generating delays. These methods are as follows.

- Using NOP instructions
- Using 8-bit register as counter
- Using 16-bit register pair as counter.

Using NOP instructions:

One of the main usage of NOP instruction is in delay generation. The NOP instruction is taking four clock pulses to be fetching, decoding and executing. If the 8085 MPU is working on 6MHz clock frequency, then the internal clock frequency is 3MHz. So from that we can easily determine that each clock period is 1/3 of a microsecond. So the NOP will be executed in 1/3 * 4 = 1.333µs. If we use the entire memory with NOP instruction, then 64K NOP instructions will be executed. Then the overall delay will be 216 * 1.333µs = 87359.488µs, though the time is not so large and the program size is also large. So this type of NOP instruction can be used to generate a short time delay for few milliseconds.

Using 8-bit register as counter:

Counter is another approach to generate a time delay. In this case the program size is smaller. So in this approach we can generate more time delay in less space. The following program will demonstrate the time delay using 8-bit counter.

```
      MVI B,FFH
LOOP: DCR B
      JNZ LOOP
      RET
```

Here the first instruction will be executed once, it will take 7 T-states. DCR C instruction takes 4 T-states. This will be executed 255 (FF) times. The JNZ instruction takes 10 T-states when it jumps (It jumps 254 times), otherwise it will take 7 T-States. And the RET instruction takes 10 T-States. 7 + ((4*255) + (10*254)) + 7 + 10 = 3584. So the time delay will be 3584 * 1/3µs = 1194.66µs. So when we need some small delay, then we can use this technique with some other values in the place of FF.

This technique can also be done using some nested loops to get larger delays. The following code is showing how we can get some delay with one loop into some other loops.

```
    MVI B,FFH
L1: MVI C,FFH
L2: DCR C
    JNZ L2
    DCR B
    JNZ L1
    RET
```

From this block, if we calculate the delay, it will be nearly 305µs delay. It extends the time of delay.

Using 16-bit register-pair as counter:

Instead of using 8-bit counter, we can do that kind of task using 16-bit register pair. Using this method more time delay can be generated. This method can be used to get more than 0.5 seconds delay. Let us see and example.

Program	Time (T-States)
LXI B,FFFFH LOOP: DCX B MOV A,B	10 6 4

ORA C **JNZ LOOP** **RET**	4 10 (For Jump), 7(Skip) 10

In the above table we have placed the T-States. From that table, if we calculate the time delay, it will be like this:
10 + (6 + 4 + 4 + 10) * 65535H – 3 + 10 = 17 + 24 * 65535H = 1572857. So the time delay will be 1572857 * 1/3µs = 0.52428s. Here we are getting nearly 0.5s delay.
In different program, we need 1s delay. For that case, this program can be executed twice. We can call the Delay subroutine twice or use another outer loop for two-time execution

d. Write an assembly language program for 8085 microprocessor to count continuously from FFH to 00H in a system with 0.05 micro second clock period. Set up a delay of 1 millisecond between two values.

ADDRESS	LABEL	MNEMONICS
2000H		MVI B, FFH
2002H	NEXT	DCR B
2003H		MVI C, COUNT
2005H	DELAY	DCR C
2006H		JNZ DELAY
2009H		MOV A, B
200AH		OUTPORT#
200CH		JMP NEXT

The C register is the time delay register which is loaded by a value COUNT to produce a time delay of 1ms.
To find the value of COUNT we do-

$T_D = T_L + T_O$
where- T_D = Time Delay
T_L = Time delay inside loop
T_O = Time delay outside loop

The **delay loop** includes two instructions- DCR C (4 T-states) and JNZ (10 T-states)
So T_L = 14*Clock period*COUNT
=> $14*(0.5*10^{-6})*COUNT$

=> $(7*10^{-6})$*COUNT

Delay outside the loop includes-
DCR B : 4T
MVI C, COUNT : 7T
MOV A, B : 4T
OUTPORT : 10T
JMP : 10T
Total : 35T
T_O= 35*Clock period => 17.5 microseconds
So, 1ms= (17.5+ 7*COUNT)microsecond

Therefore, COUNT=$(140)_{10}$

e. What is stack? How is stack used both by microprocessor and user?

A stack is just a space for storing stuff and follows the concept of the stack data structure. In the stack data structure, insertion and deletion can always occur at one end only.
So, let us say you want to store something in memory and then delete it from memory and you want to be confident that it is the value which you want, you can use a stack for that.
Now,
You might be wondering what the use of a stack in a microprocessor is. There are many uses of a stack. Also, as far as I know, stacks are used in every microprocessor.
Some of the uses in 8085 microprocessor are-
1. for storing flags and the contents of accumulator when a CALL is encountered- so that the current state can be saved and then the control can be transferred to the (subroutine) address location which the CALL statement references. After returning from the subroutine, these flags and contents of the accumulator are removed from the stack and the values in flag registers and accumulator are restored.
2. same as above in case of an interrupt to transfer the control to the ISR.
Also, the use of stacks depends on the programmer. He can use it as per his choice. It is just a way of storing and retrieving contents from memory.
Now, a stack can be present anywhere. It is no more than a way of accessing a memory location.
In 8085, the stack pointer is used to set the memory location from which stack starts, ie- stack pointer points to that memory location and decrements from that location on push and increments on the pop operation. This memory is the RAM or Main Memory.

f. Explain the following instructions for 8085 microprocessor-
i) Restart
ii) Conditional call and return

In 8085 Instruction set, RSTn is actually standing for “Restart n”. And in this case, n has a value from 0 to 7 only. Thus the eight possible RST instructions are there, e.g. RST 0, RST 1, …, RST 7. They are 1-Byte call instructions.

Conditional Call Instructions: Only if the condition is satisfied, the instructions executes.

OPCODE	OPERAND	EXPLANATION	EXAMPLE

CC	address	Call if carry flag is 1	CC 2050
CNC	address	Call if carry flag is 0	CNC 2050
CZ	address	Calls if zero flag is 1	CZ 2050
CNZ	address	Calls if zero flag is 0	CNZ 2050
CPE	address	Calls if carry flag is 1	CPE 2050
CPO	address	Calls if carry flag is 0	CPO 2050
CM	address	Calls if sign flag is 1	CM 2050
CP	address	Calls if sign flag is 0	CP 2050

Conditional Return Instruction: The program sequence is transferred unconditionally from the subroutine to the calling program only is the condition is satisfied.

OPCODE	OPERAND	EXPLANATION	EXAMPLE
RC	none	Return from the subroutine if carry flag is 1	RC
RNC	none	Return from the subroutine if carry flag is 0	RNC
RZ	none	Return from the subroutine if zero flag is 1	RZ
RNZ	none	Return from the subroutine if zero flag is 0	RNZ
RPE	none	Return from the subroutine if parity flag is 1	RPE
RPO	none	Return from the subroutine if parity flag is 0	RPO
RM	none	Returns from the subroutine if sign flag is 1	RM

RP	none	Returns from the subroutine if sign flag is 0	RP

4 Attempt any three of the following:

a. Write a 8085 assembly language to convert a 8-bit binary number to unpacked BCD.

Address	Labels	Mnemonics	Comments
F000		LXI H,8000H	Initialize memory pointer
F003		MVI D,00H	Clear D- reg for Most significant Byte
F005		XRA A	Clear Accumulator
F006		MOV C, M	Get HEX data
F007	LOOP	ADI 01H	Count the number one by one
F009		DAA	Adjust for BCD count
F00A		JNC SKIP	Jump to SKIP
F00D		INR D	Increase D
F00E	SKIP	DCR C	Decrease C register
F00F		JNZ LOOP	Jump to LOOP
F012		MOV L, A	Load the Least Significant Byte
F013		MOV H, D	Load the Most Significant Byte
F014		SHLD 8050H	Store the BCD
F017		HLT	Terminate the program

b. What is meant by look up technique? How is it used for BCD to seven segment LED code conversion?

Look up technique: An array or matrix of data that contains items that are searched. Lookup tables may be arranged as key-value pairs, where the keys are the data items being searched (looked up) and the values are either the actual data or pointers to where the data are located.

BCD to 7 Segment Display Decoder

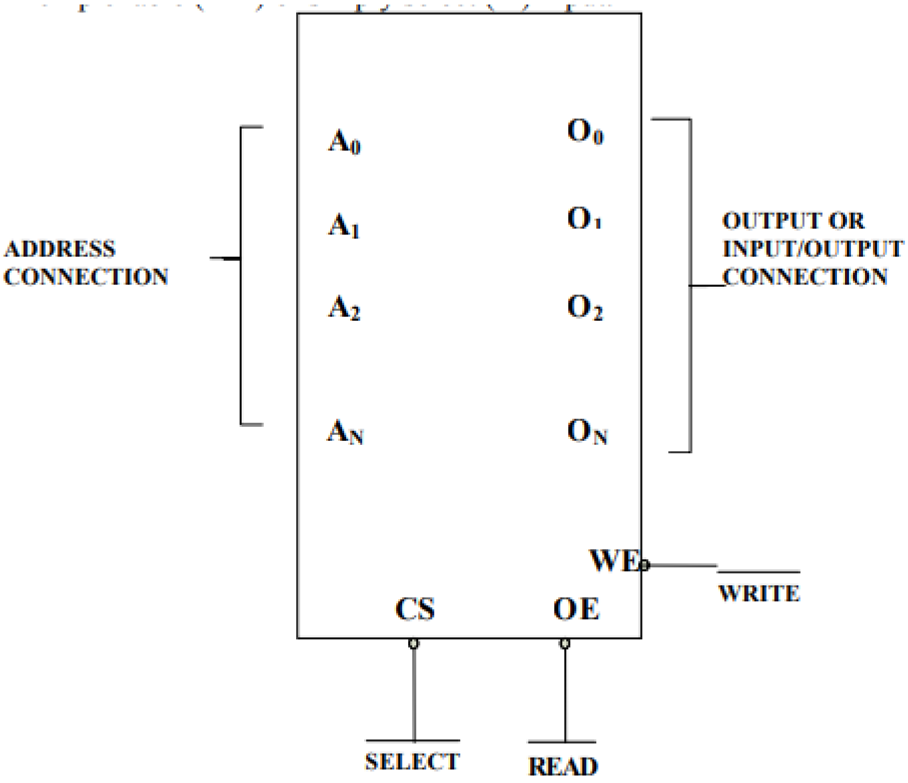
A BCD to Seven Segment decoder is a combinational logic circuit that accepts a decimal digit in BCD (input) and generates appropriate outputs for the segments to display the input decimal digit.
The truth table is extracted from the CD4511 IC datasheet. This truth table is interactive. Click on any row to see the respective 7 segment display output.

Truth Table

Display	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

c. Explain the hardware features of a typical software development system.

Interface
- We have four common types of memory:
- Read only memory (ROM)
- Flash memory (EEPROM)
- Static Random access memory (SARAM)
- Dynamic Random access memory (DRAM).
- Pin connections common to all memory devices are: The address input, data output or input/outputs, selection input and control input used to select a read or write operation.
- Address connections: All memory devices have address inputs that select a memory location within the memory device. Address inputs are labeled from A0 to An.
- Data connections: All memory devices have a set of data outputs or input/outputs. Today many of them have bi-directional common I/O pins.
- Selection connections: Each memory device has an input, that selects or enables the memory device. This kind of input is most often called a chip select (CS), chip enable (CE) or simply select (S) input.



MEMORY COMPONENT ILLUSTRATING THE ADDRESS, DATA AND CONTROL CONNECTIONS

RAM memory generally has at least one CS or S input and ROM at least one CE .
• If the CE , CS , S input is active the memory device perform the read or write.
If it is inactive the memory device cannot perform read or write operation.
• If more than one CS connection is present, all most be active to perform read or write data.

d. What are advantages of an assembler?

1. Programs written in machine language are replaceable by mnemonics which are easier to remember.
2. Memory Efficient.
3. It is not required to keep track of memory locations.
4. Faster in speed.
5. Easy to make insertions and deletions.
6. Hardware Oriented.
7. Requires fewer instructions to accomplish the same result.

e. Discuss various interrupts used by 8085 microprocessor and their priorities

The 8085 microprocessor has five interrupt inputs. They are TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR. These interrupts have a fixed priority of interrupt service. If two or more interrupts go high at the same time, the 8085 will service them on priority basis. The TRAP has the highest priority followed by RST 7.5, RST 6.5, RST 5.5. The priority of interrupts in 8085 is shown below.

TRAP - 1
RST 7.5 - 2
RST 6.5 - 3
RST 5.5 - 4

INTR - 5

f. What is meant by vectored interrupt? Also explain use of SIM instruction.

Vectored Interrupt

The hardware interrupts of 8085 are TRAP, RST 7.5, RST 6.5, RST 5.5 and INTR. In vectored interrupts, the processor automatically branches to the specific address in response to an interrupt.

SIM

The 8085 provide additional masking facility for RST 7.5, RST 6.5 and RST 5.5 using SIM instruction.

The status of these interrupts can be read by executing RIM instruction.

The masking or unmasking of RST 7.5, RST 6.5 and RST 5.5 interrupts can be performed by moving an 8-bit data to accumulator and then executing SIM instruction.

The format of the 8-bit data is shown below.

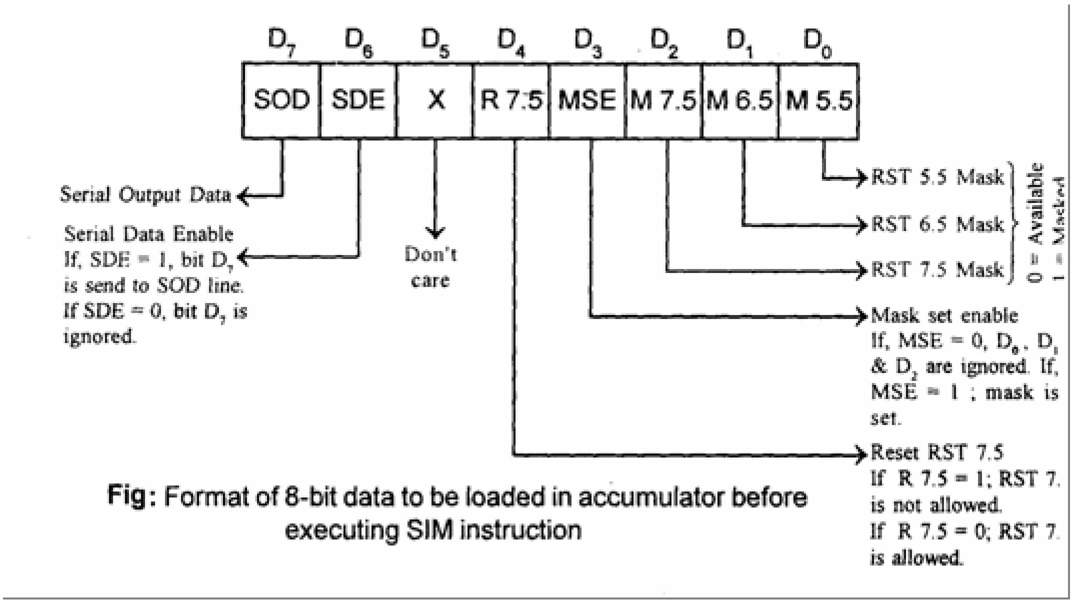

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
SOD SDE X R 7.5 MSE M 7.5 M 6.5 M 5.5
Serial Output Data
Serial Data Enable If, SDE = 1, bit D_7 is send to SOD line. If SDE = 0, bit D_7 is ignored.
Don't care
RST 5.5 Mask
RST 6.5 Mask
RST 7.5 Mask
0 = Available
1 = Masked
Mask set enable If, MSE = 0, D_0, D_1 & D_2 are ignored. If, MSE = 1 ; mask is set.
Reset RST 7.5 If R 7.5 = 1; RST 7. is not allowed. If R 7.5 = 0; RST 7. is allowed.

Fig: Format of 8-bit data to be loaded in accumulator before executing SIM instruction

5 Attempt any three of the following:

a. What are special Pentium Registers? Discuss the architecture of Special Pentium Registers.

The special-purpose registers include EIP, ESP, EFLAGS; and the segment registers CS, DS, ES, SS, FS, and GS.

EIP (instruction pointer)

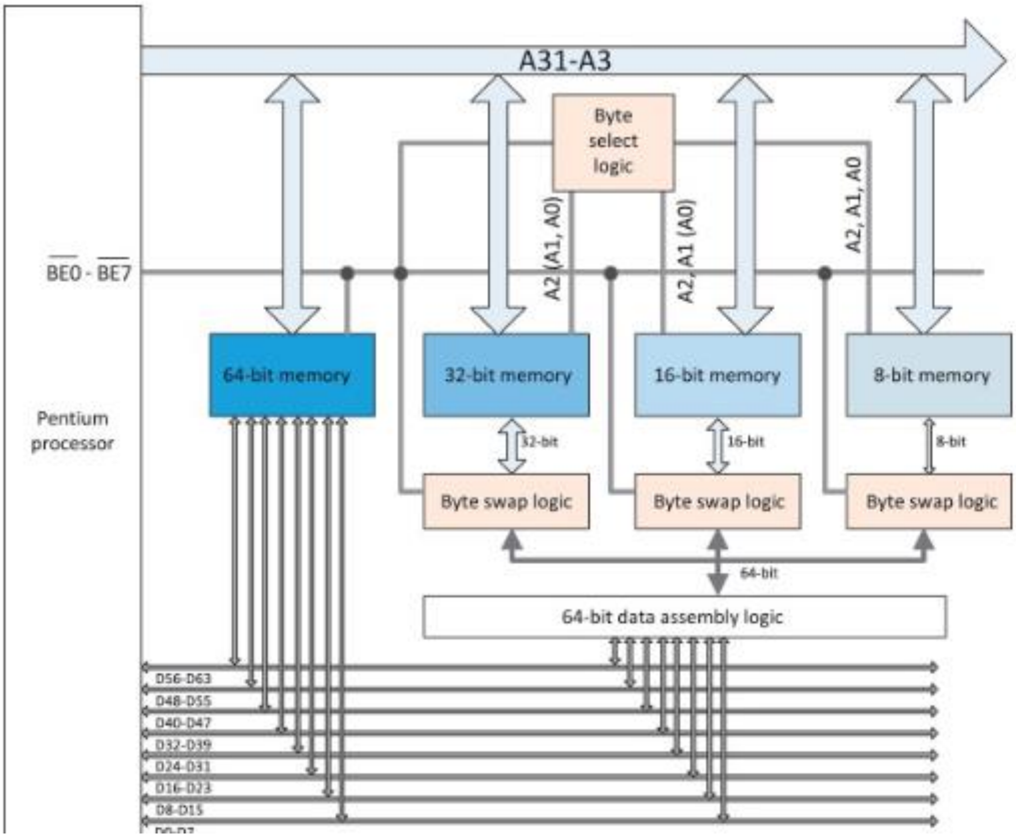
EIP addresses the next instruction in a section of memory defined as a code segment. This register is IP (16 bits) when the microprocessor operates in the real mode and EIP (32 bits) when the 80386 and above operate in the protected mode. Note that the 8086, 8088, and 80286 do contain EIP, and only the 80286 and above operate in the protected mode. The instruction pointer, which points to the next instruction in a program, is used by the microprocessor to find the next sequential instruction in a program located within the code segment. The instruction pointer can be modified with a jump or a call instruction.

ESP (stack pointer)

ESP addresses an area of memory called the stack. The stack memory stores data through this pointer. This register is referred to as SP if used as a 16-hit register and ESP if referred to as a 32-bit register.

b. Discuss the memory map of Pentium 2 Processor.

The Pentium processor has a memory space of 4 GB (232 bytes) and a separate I/O space with 64 KB of addressable locations. The memory space is organized as a sequence of 64-bit quantities. Each 64-bit location has eight individually addressable bytes at consecutive memory addresses. The I/O space is organized as a sequence of 32-bit quantities. Each 32-bit quantity has four individually addressable bytes at consecutive memory addresses.
Below illustrates how the Pentium processor interfaces with memory of various width:



c. Explain the CPUID instruction used by Pentium 4.

CPUID--CPU Identification

Opcode	Instruction	Description
0F A2	CPUID	Returns processor identification and feature information to the EAX, EBX, ECX, and EDX registers, according to the input value entered initially in the EAX register.

Description
Provides processor identification information in registers EAX, EBX, ECX, and EDX. This information identifies Intel as the vendor, gives the family, model, and stepping of processor, feature information, and cache information.

IA-32 Processors	Highest Value in EAX		
		Basic Information	Extended Function Information

Pentium 4 processors	2H	80000004H	

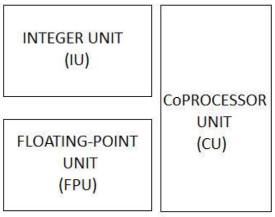
d. Explain the architecture of SPARC.

- SPARC stands for Scalable Processor Architecture.
- developed by Sun Microsystems in the 1980s.
- is based on the RISC structure designed at the University of California at Berkeley in early 1980s.
- The SPARC architecture is a non-proprietary architecture that any person or company can license and use to develop microprocessors and other semiconductor devices based on published industry standards.
- In 1989, Sun Microsystems transferred ownership of the SPARC specifications to an independent, non-profit organization, SPARC International, which administers and licenses the technology and provides conformance testing and other services for its members.

SPARC Architecture

- It is a Load and store architecture. Operations are always done over registers.
- Uses “register window” concept thus offering a large number of registers.
- Uses delay slot to optimize branch instruction.
- Passes arguments using registers and the stack.

e. List the component of SPARC processor. Discuss each in brief.



The Integer Unit (IU)

- Contains the general purpose registers and controls the overall operation of the processor.
- may contain from 64 to 528 general-purpose 64-bit r registers. They are partitioned into 8 global registers, 8 alternate global registers, plus a circular stack of from 3 to 32 sets of 16 registers each, known as register windows.
- Executes the integer arithmetic instructions and computes memory addresses for loads and stores.
- Maintains the program counters and controls instruction execution for the FPU.

The Floating-point Unit (FPU)

- The FPU has 32 32-bit (single-precision) floating-point registers, 32 64-bit (double-precision) floating-point registers, and 16 128-bit (quad-precision) floating-point registers.
- Double-precision values occupy an even-odd pair of single-precision registers.
- Quad-precision values occupy an odd-even number pair of double precision registers.
- Floating-point load/store instructions are used to move data between the FPU and

memory.

- The memory address is calculated by the IU.
- Floating-Point operate (FPop) instructions perform the floating-point arithmetic operations and comparisons

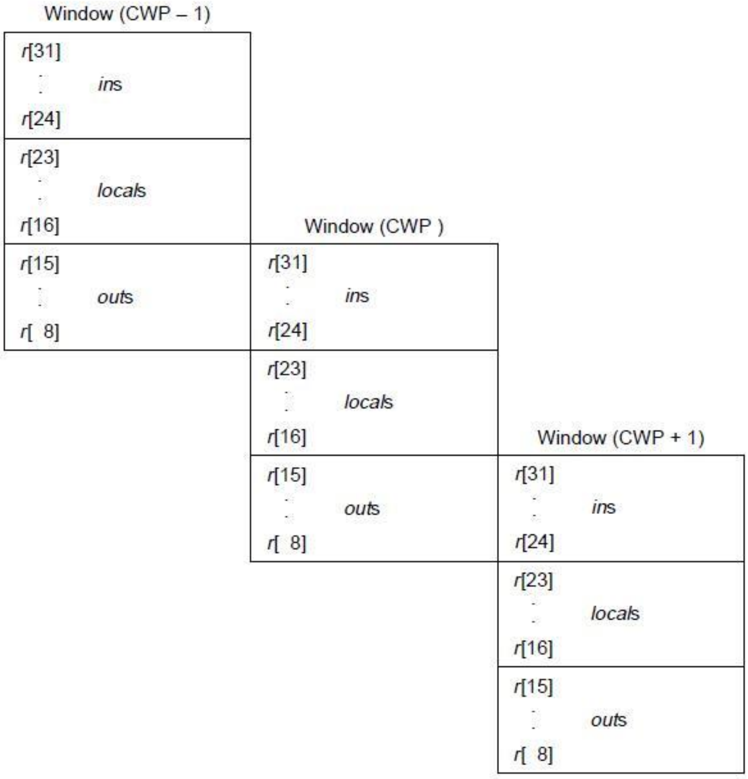
Coprocessor Unit (CU)

- The instruction set includes support for a single, implementation-dependent coprocessor. The coprocessor has its own set of registers.
- Coprocessor load/store instructions are used to move data between the coprocessor registers and memory.
- floating-point instructions mirrors coprocessor instructions.
- Not implemented in SPARC V9.

f. Explain the concept of windowed register of SPARC microprocessor.

The Register Window

□ The current window into the *r registers is given by the current window pointer (CWP) register.*


Window (CWP – 1)
r[31]
ins
r[24]
r[23]
locals
r[16]
r[15]
outs
r[8]
Window (CWP)
r[31]
ins
r[24]
r[23]
locals
r[16]
r[15]
outs
r[8]
Window (CWP + 1)
r[31]
ins
r[24]
r[23]
locals
r[16]
r[15]
outs
r[8]