Q.1 a. What do you understand by the term Computer organization, design and architecture?

Answer:

Computer architecture is concerned with the structure and behavior of the various functional modules of the computer and how they interconnected to provide the processing needs of user. **Computer organization** is concerned with the way the hardware components are connected together to form a computer system. **Computer design** is concerned with the development of the hardware for the computer taking into consideration a given set of specifications. **4**

b. Explain the differences between hardwired and Micro-programmed control unit.

Answer:

Hardwired control unit: It is designed for specific operation. If one time control unit is designed, it cannot be changed for other operation. It is the hardwired component to decode the instruction and generate the control signals. Hardwired control unit is faster and less costly compare to micro programmed control unit. Less number of instructions. For RISC processor. **4**

Micro-programmed control unit: It is designed for general-purpose applications. It is programmable; it means the control unit can be reprogrammed for other operations or applications. Micro-programmed are stored in control memory. Control signals are generated by decoding the programmed. It is slower and costlier compare to hardwired control unit. For example CISC processor.

c. Explain the instruction pipelining in the RISC processor.

Answer:

In the basic RISC processor the instruction cycle can be divided into three sub-operations and implemented in three segments: Instruction fetch (I), ALU operation (A), and Execute instruction (E). The Instruction fetch unit fetches the instruction from program memory. The instruction is decoded and an ALU operation is performed in the A unit. The ALU is used for three different functions, depending on the decoded instruction. It performed an operation for a data manipulation instruction, it evaluates the effective address for a load or store instruction or it calculates the branch address for a program control instruction. The execution instruction unit directs the output of the ALU to one of three destinations, depending on the decoded instruction. Destination may be register, memory or branch address. **4**

Instruction fetch unit (I)	ALU operation unit (A)	Execute instruction unit (E)

In some instruction pipeline system there may be more than three segments, Instruction fetch (IF), Instruction decode (ID), Operand fetch (OF), Execute the instruction (EI) and Write back (WB).

d. What is the maximum positive and negative value if number is represented using 8 bits including the sign bit in
(i) signed magnitude representation
(ii) 1's complement representation and
(iii) 2's complement representation

Answer:

8 bits are used for representation of number including the sign bit. It means 7 bits for magnitude and 1 bit for sign. Maximum positive and negative number that can be represented will be:

Sign magnitude and 1's complement representation : $\pm 2^7-1 = \pm 127$

2's complement representation: $+2^7-1$ to $-2^7 = +127$ to -128

4

e. What is the stack memory and what is the roll of stack in subroutine?

Answer:

Stack is memory unit that can be used for temporary data storage. It can be unit of memory or set of register s. It is based on LIFO (Last in First Out). The register that holds the address for the stack is called a stack pointer (SP). SP always point to the top location of stack. Push and pop can be used to insert and delete the data from the stack.

After pop, data are not physically removed from the memory. In PUSH following set of micro-operation are performed.

SP ← SP−1
M [SP] ← DR

Stack pointer is decremented by one so that it points the next location. We have assumed that memory write is perform through DR only, so data available in DR will be place in push operation. Similarly in the POP following set of micro-operation will be performed.

DR ← M [SP]
SP ← SP +1

Data pointer by SP will be read from the memory and paced in DR. SP is incremented by one now that will point the top of the stack.

In case of call subroutine, the return address is stored in stack memory before program counter is loaded by subroutine location. For return in to main program, return address stored in stack memory will be loaded in to program counter.

4

f. A computer system uses the memory unit with 256K words of 32 bits each. A binary instruction code is stored in one word of memory. The instruction has four parts: an indirect bit, an operation cjode, a register code part to specify one of 64 registers, and an address part. Draw the instruction format and indicate the number of bits in each part

Answer:

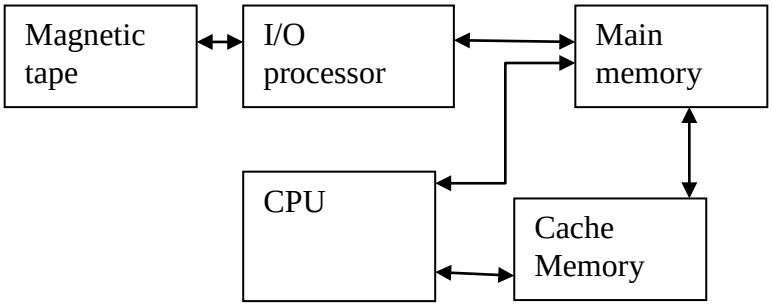
A computer system uses the memory unit with 256K words of 32 bits each. A binary instruction code is stored in one word of memory that means size of instruction is 32 bits. Size of memory is 256K, required the 18 bits in address field. There are 64 registers; therefore, 6 bits are required in register code field to select one register. One bit is for indirect addressing, therefore (32-18-6-1 = 7) 7 bits are there in the operation code field. The instruction format is given bellow. **4**

I (1 bit)	Operation code (7 bits)	Register code (6 bits)	Address (18 bits)

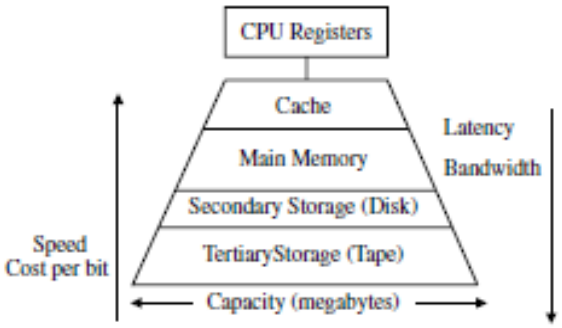
g. Draw the memory hierarchy system. Clearly mention the speed, capacity and access type at each level of memory hierarchy. (7 × 4)

Answer:

Memory hierarchy is given bellow.



Or **4**



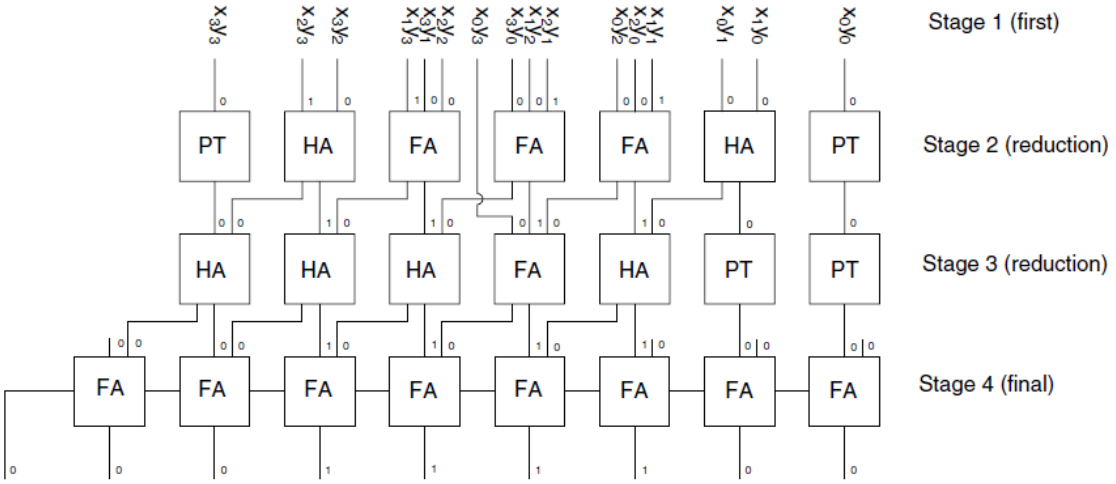
	Access type	Capacity	Latency	Bandwidth	Cost/MB
CPU registers	Random	64–1024 bytes	1–10 ns	System clock rate	High
Cache memory	Random	8–512 KB	15–20 ns	10–20 MB/s	$500
Main memory	Random	16–512 MB	30–50 ns	1–2 MB/s	$20–50
Disk memory	Direct	1–20 GB	10–30 ms	1–2 MB/s	$0.25
Tape memory	Sequential	1–20 TB	30–10,000 ms	1–2 MB/s	$0.025

Q.2 a. Draw the block diagram of Wallace tree multiplier to multiply two 4-bit numbers. Explain the operation performed by multiplier with example. (9)

Answer:

Wallace multiplier designs capitalize on the concept of carry-save adders. It is tree-like multiplier designs in the sense that it generates and sum partial products in parallel, but improves the critical path through the multiplier. For *n*-bit inputs, it constructed from a number of stages: an initial stage to generate the partial products; one or more reduction stages which act to reduce the partial products into two 2*n*-bit values; and a final stage which adds the 2*n*-bit values to produce the result.

4 + 5



4bit Wallace multiplier

The Wallace multiplier basic approach is as follow.

1. In the first stage, multiply together (i.e., AND together) each x_i with each y_j to produce a total of n^2 intermediate wires. Each wire is said to have a weight, for example $x_0 \cdot y_0$ has weight 1 as $2^0 \cdot 2^0 = 1$, $x_1 \cdot y_2$ has weight 8 as $2^1 \cdot 2^2 = 8$.
2. Reduce the number of intermediate wires using additional stages composed of full-adders and half-adders:
   - Combine any three wires with same weight using a full-adder; result in next stage is one wire of the same weight (i.e., the sum) and one wire a higher weight (i.e., the carry).
   - Combine any two wires with same weight using a half-adder; result in next stage is one wire of the same weight (i.e., the sum) and one wire a higher weight (i.e., the carry).
   - If there is only one wire with a given weight, just pass it through to the next stage.
3. In the final stage, all weights have just one or two wires in them: combine the wires into two 2*n*-bit values and add them with a standard adder.

b. An instruction of computer system has two parts: Op-code and Operand. Instruction is stored at two consecutive locations of memory as given below. It has one general purpose register (R1) and index register (XR). Content of

memory with their addresses are given below. Contents of Register R1 and index register XR are 400 and 100 respectively. What will be value of AC after the execution of instruction, in following addressing modes? (9)

(i) Register addressing mode.
(ii) Immediate addressing.
(iii) Direct Addressing
(iv) Indirect Addressing
(v) Register indirect
(vi) Indexed Addressing

Addr.	***Contents***
200	*Load to AC*
201	*500*
:	
399	*450*
400	*700*
:	
500	*800*
:	
600	*900*
:	
702	*325*
:	
800	*800*

Answer:

Content of R1= 400 and XR = 100.

Content of AC after:

Addr.	***Contents***
200	*Load to AC*
201	*500*
399	*450*
400	*700*
500	*800*
:	
600	*900*
702	*325*
800	*800*

1.5x6

(1) Register addressing: AC will be loaded by content of register R1 that is **400**.

(2) Immediate addressing: AC will be loaded by the immediate data **500**.

(3) Direct addressing: 500 is directly point the data (500 is effective address), therefore AC is loaded by the data from memory location 500, which is **800**.

(4) Indirect addressing: 500 does not point the data directly (it is not effective address), effective address is stored at location 500, which is 800. Therefore, data from memory location 800, will be loaded in accumulator, which is **800.**

(5) Register indirect: Register R1 has the effective store of data. Therefore, effective address of data is 400. Data from memory location 400, will be loaded in AC, which is **700.**

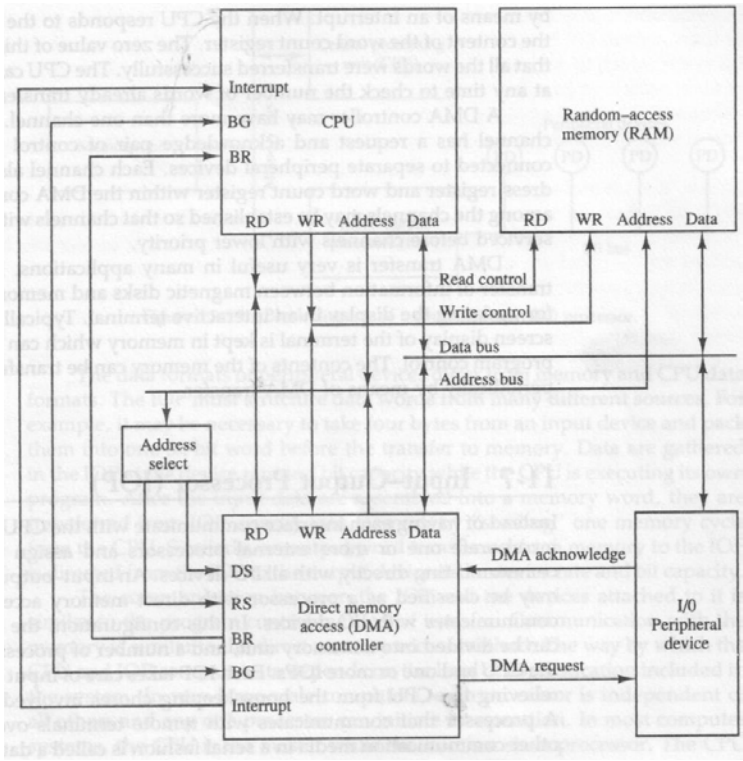
(6) Indexed addressing: Second byte of instruction 500 (operand part) is added with content of index register (100) to find the effective address of data. Therefore, data from memory location 600 will be loaded in AC, which is **900.**

Q.3 a. What are the steps required for data transfer using DMA controller? Explain the interfacing diagram of DMA controller with computer system. (9)

Answer:

When the peripheral device sends a DMA request, the DMA controller activates the BR line, informing the CPU to relinquish the buses. The CPU responds with its BG line, informing the DMA that its buses are disabled. The DMA then puts the current value of its address register into the address bus, initiates the RD or WR signal, and send a DMA acknowledgement to the peripheral device. The direction of transfer depends on the status of the BG line. When BG = 0, the RD and WR are input lines allowing the CPU to communicate with the internal DMA registers. When BG =1, the RD and WR are output lines from the DMA controller to the memory to specify the read or write operation for the data. When the peripheral device received a DMA acknowledgement, it puts a word in the data bus or receives a word from the bus.

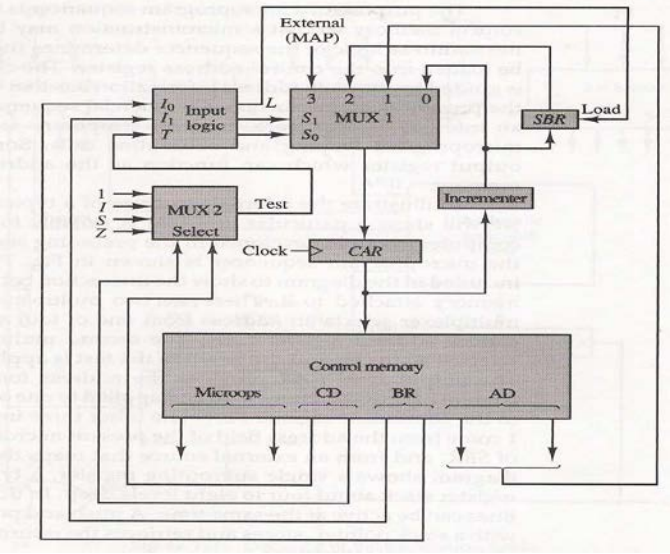
4 + 5



b. A computer has the memory capacity of 2048 x 16. It is based on the micro-programmed control unit. Size of control memory is 128 x 20. Draw the block diagram of micro-program sequencer for given computer. (9)

Answer:

Micro-program sequencer block diagram.



9

Q.4 a. What do you understand by general register organised CPU? Draw the block diagram of bus organization for general register organized CPU, which has following control word (instruction format).

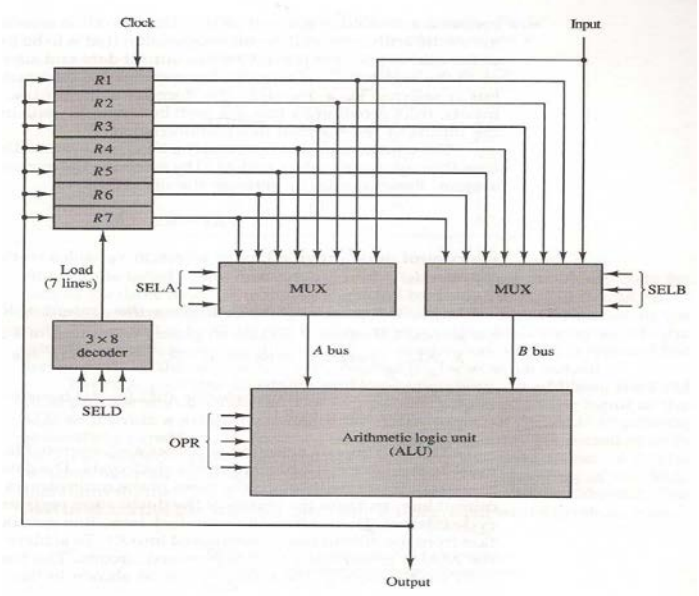
SEL A	SEL B	SEL D	ORP
(3 bits)	(3 bits)	(3 bits)	(5 bits)

Where SEL A and SEL B are two source register and SEL D is destination register field. ORP is 5-bit operation code field. (9)

Answer:

Because the memory access in the time consuming operation, so it is more efficient to store the intermediate data in the processor register. When large numbers of registers are there then it is convenient to connect them through common bus operations. These registers communicate each other directly for data transfer and micro-operation.

Block diagram of register organized CPU.



4 + 5

Control word or instruction format is given bellow.

3	3	3	5
SELA	SELB	SELD	OPR

SELA and SELB used to select the two source data registers and SELD is used to select the destination register to store the result coming from the ALU. Operation may be perform on the operand may coming from the input (external data from the input). Only seven output lines of 3 x 8 decoder are use to activate the load pin of one of the seven registers.

b. Explain the IEEE-754 single precision floating point representation and its format. Represent -1.5_{10} in single precision IEEE-754 floating-point representation. (9)

Answer:

IEEE 754 (floating point repersentation) standard 32 bit format is given below. It comprises 23 bit mantisa field *M*, an 8 bit exponent field *E*, and a sign bit S. The base is two. The exponent is repersented in 8 bit excess-127 code.

Sign (S)	Exponenet (E)	Mantisa (M)
1 bit	(8 bit excess-127 code)	23 bit

If N is a real number then in 32- bit floating point (IEEE754) repersentation it can repersented as:

$$N = (-1)^S 2^{E-127}(1.M) \quad \text{-------(1)}$$

4 + 5

Decimal $-1.5_{10} = -1.1_2$ (binary) = $(-1)^1 2^{127-127}(1.1)$ compare this with equestion No. 1 that gives

$E = 127_{10}$ and S = 1 and M = 1

Thefore in IEEE 754 format it will be 1 01111111 10000000000000000000000

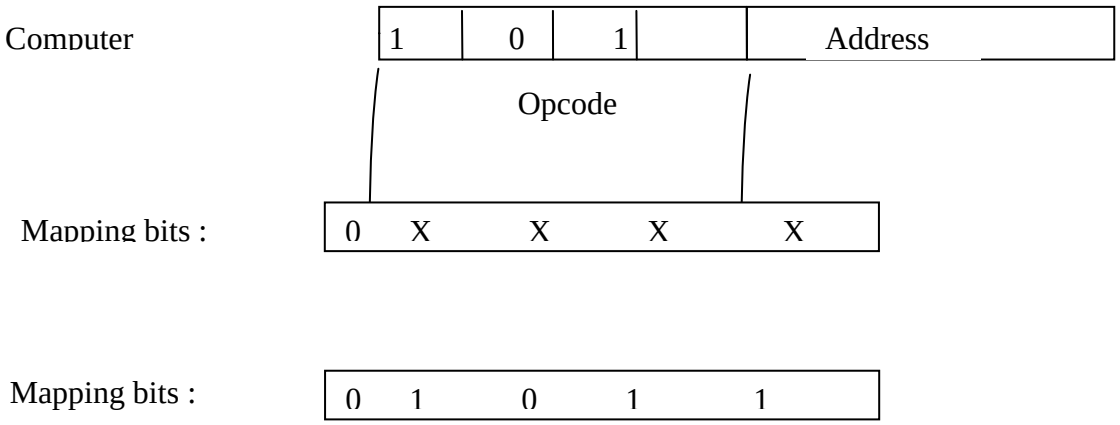
Q.5 a. How mapping from an instruction code to microinstruction address is being performed by Control Address Register? Explain the mapping procedure with suitable diagram. (9)

Answer:

The size of main memory is 2048 X 16 and size of control memory is 128 X 20. The basic instruction format is given below has 1 bit indirect addressing bit, 4 bit opcode field and 11 bits in address field. The control memory has 128 words, therefore it required 7 bit address bits. For each instrcution (operation code) there exists a microprogam routine in control memory that executes the instruction. The simple mapping procedure that convert the 4 bit opcode to 7 bit address for control memory is shown in figure. This mapping consist of placing a 0 in the most significant bit of the address, transferring the four operation code bits, and claering the two least significat bits of the control address register (control address

4 + 5

register). This provides for each computer instruction a microprogram routine with a capacity of four microinstrcution. If the each routinr need more than four microinstrcutions. If it uses fewer than four microinstrcutions, the unused memory locations would be available for other routine.



b. Design an arithmetic circuit to generate the following arithmetic operations. Draw the logic diagram of one typical stage. (9)

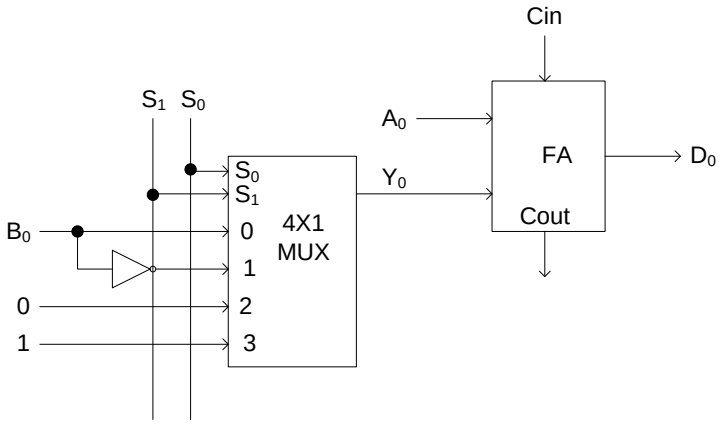
S_1	S_0	$C_{in} = 0$	$C_{in} = 1$
0	**0**	$F = A + B$	$F = A + B + 1$
0	**1**	$F = \overline{B} + A$	$F = \overline{B} + A + 1$
1	**0**	$F = A$	$F = A + 1$
1	**1**	$F = A - 1$	$F = A$

Answer:

Design of ALU:

S_1	S_0	C_{in}	Output of MUX (Y)	Output (D)	Micro-operation performed
0	0	0	B	A+B	Add
0	0	1	B	A+B+1	Add with carry
0	1	0	B’	A+B’	Subtract with borrow
0	1	1	B’	A+B’+1 = A-B	Subtract
1	0	0	0	A	Transfer
1	0	1	0	A+1	Increment
1	1	0	1 (all 1)	A+ all 1 = A+2’ of 1= A-1	Decrement
1	1	1	1 (all 1)	A-1+1= A	Transfer

4 + 5



Q.6 a. A digital computer has a memory unit of 64K x 16 and a cache memory of 1K word. The cache uses direct mapping with a block size of four words.
(i) How many bits are there in the tag, index, block and word field of the address format?
(ii) How many bits are there in each word of cache, and how are they divided in to functions?
(iii)How many blocks can the cache accommodate? (9)

Answer:

Memory mapped Vs Isolated mapped:

In the isolated I/O configuration, the CPU has input and output instructions and each of these instructions is associated with the address of an interface register. When the CPU fetch es and decodes the operation code of an input ouput instruction, it palce the address associated with the instruction into the commond address lines. At the same time, it enables the I/O read(for input) or I/O write (for output) control line. This informs the externa components thet are attached to the common bus that the address in the address lines is for an interface registre not for a memory word. On the other hand , when the CPU is fetching an instruction or an operand from memory, it places the memory address on the address lines and enables the memory read or memory write control line. This informs the external the external componenets that the address is for a memory word and not for an I/O interface.

6 + 3

The Isolated I/O method isolates memory and I/O address so that memory address values are not affcted by interface address assignment since each has its own address space. The other alternative is to use the same address space for both memory and I/O. This is the case in computer that emply only one set of read and write signals and do not distinguish between memory and I/O addresses. This configuration is refred as memory mappped I/O. The computer treats an interface register as part of the memory system. The assigned addtess for interface register cannot be used for memory words, which reduces the addess range available.

In a memory mappped I/O organization there are no specific input or output instruction. The CPU can manipulate I/O data residing in interface registers with the same instructions that are used to manipulate memory words. Ecah interface is orgnized as set if register that responds to read and write request in the normal address space. Typically, a segments of the total address space is reserved for interface registers, but in general , they can be located at any adddress as long as therre is not also a memory word that responds to the same address.

Computer with memory mapped I/O can use memory type instructions to access I/O data. IT alloww the computer to use the same instrurion for either inout-output transfer or for memory trasfre. The advantage is that the load and store instructions used for redaing and writing from memory can be used to input and output data from I/O register. In the tuypical computer there are more memory refrenace instruction than I/O instructions. With memory mapped I/O all instructions that refer to memory are also available for I/O.

b. Explain the isolated (peripheral) mapped I/O and memory-mapped I/O. What are the advantages and disadvantages of each? (9)

Answer:

Memory is 64K X16 : 16 bit address. 16 bit data.

(a) Tag is 6, block is 8 and word is 2, Index will be 8+2 =10 .

Tag(6)	Block (8)	Word (2)

3 x 3

(b) Number of bits in each word of cache is 23. Format is given below.

V (1)	Tag (6)	Data (16)

(c) 2^8 = 256 blocks of 4 word each.

Q.7 a. What is parallel processing and what are the different levels of parallel processing? How parallelism can be achieved in uni-processor system? (9)

Answer:

Parallel processing is techinique that used to provide simultaneous data processing task for the purpose of incresing the computational speed of comuter system. Paraller processing epmhasize the use of concurrent evets in the computing process. Parallel event may occurs in multiple resources, during the same time interval. Concurrent execution of many program in the computer. Cost effective approch to improve the system perfromance.

Parallel processing can be achived at different level:-

3 + 3 + 3

Job or program level

Task or procedure level

Inter instrcuction level

Intrainstruction level

Other mechnism to achive parallelism are:-
Lowest level: (register level): regsiter in parallel in place of serial
Low level: (Instruction level): Instrcution pipel lineing
Higher leve: Multiple fuction units, that perfrom identical or different operations simultanously.
Highest level: separate specialized processor like I/O processor, interrupt controller.
Parallel processing mechanism used in uniprocessor computers:
Multiple functional units:
Parallelism and pipeline within the CPU
Overlapped CPU and I/O operations
Use of a hieracrchical memory system
Balancing of subsytem bandwidhs
Multiprogramming and time sahring.

b. Describe the classification of parallel computer. Draw the block diagram and explain each classification. (9)

Answer:

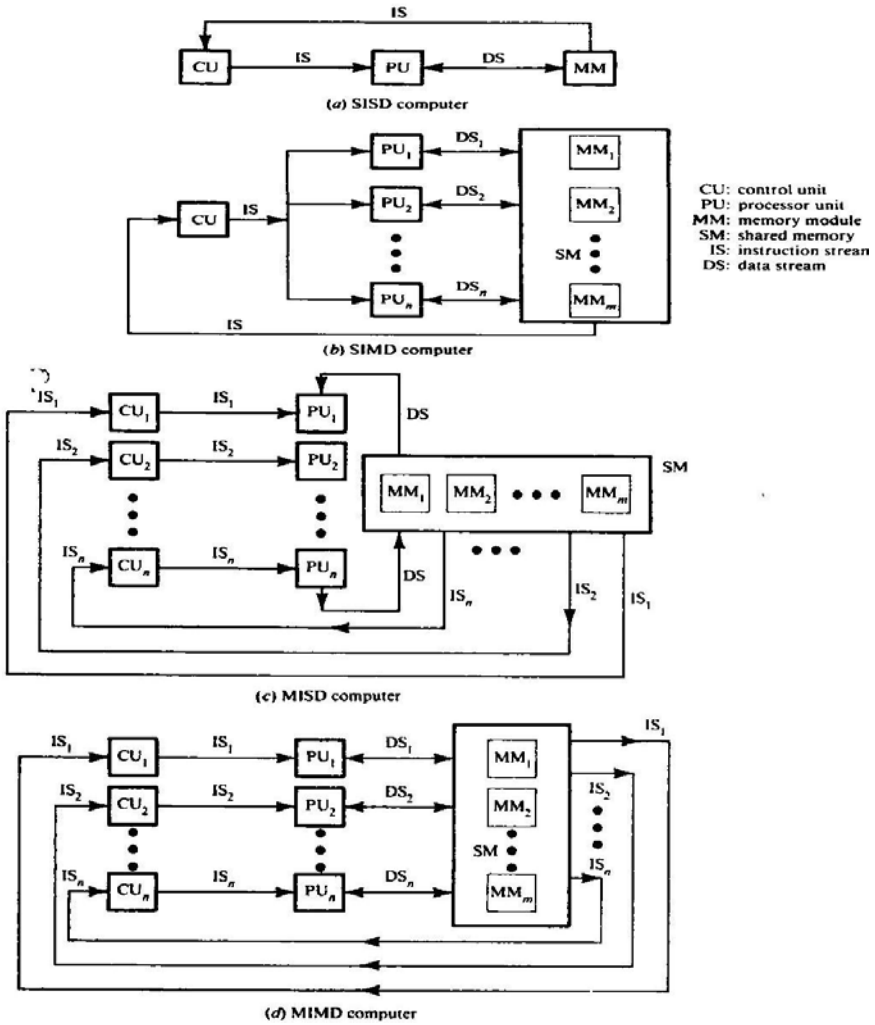
The computer are chracterized by the multiplicity of the hardware provided to service the instruction and data stream. An instruction stream is a sequence of inctructions executed by the machine. Data stream is sequence of data including input, partial or temptorary results, called by the instruction stream. Classification is given below.

SISD(single instruction single data stream): Serial computer, instructions are executed sequentially but may be overlapped in their execution stages. M ost SISD uniprocessor systems are piplined. More than one functional unit in it. All functional units are under the supervision of one control unit.

SIMD(Single instruction multiple data stream) This is array processor. Multiple processing unit in the supervision of one control unit. All PE receive the same instruction from the control unit but operate on different data sets.

MISD(multiple instruction single data stream) There are multiple processing unit , they receive the different instruction and operate on same data stream. No real computer system is this class.

MIMD(multiple instruction multiple data stream): Multiple processing unit receive the multiple instruction and operate on multiple data.



3 + 3 + 3

TEXT BOOK

I. Morris Mano-Computer System Architecture-PHI, Eastern Economy Edition-2001
II. John D. Carpinelli-Computer Systems Organization and Architecture-Pearson Education Asia-1st Edition