18IT301

Hall Ticket Number:

II/IV B.Tech DEGREE EXAMINATION

Dec, 2019

Second Semester **Computer Organization & Architecture**

Time: Three Hours **Maximum :** 50 Marks

Answer Question No.1 compulsorily. (10X1 = 10 Marks)
Answer ONE question from each unit. (4X10 = 40 Marks)

1 Answer all questions (10X1=10 Marks)

a) List the various computer types
1. Supercomputer. 2.Mainframe Computer. 3.Minicomputer. 4.Microcomputer.

b) Write basic performance equation?
.
$$T = \frac{N \times S}{R}$$
T – processor time required to execute a program
N – number of actual machine language instructions needed to complete the execution
S – average number of basic steps needed to execute one machine instruction. Each step completes in one clock cycle
R – clock rate

c) Write the steps in execution of complete instruction
1.Fetch instruction 2.Decode information 3.Perform ALU operation 4.Access memory
5.Update register file 6.Update the Program Counter (PC)
.

d) What is Mean by Multi-Phase clocking?
When edge-triggered flip flops are not used, two or more clock signals may be needed to guarantee proper transfer of data. This is known as multiphase clocking.

e) Difference between RAM and ROM

RAM	ROM
1.Temporary Storage	1.Permanent Storage
2.Stored data in MBs	2. Stored data in GBs
3.Volatile	3.Non Volatile
4.Writeing data is Faster	4. .Writing data is Slower

f) State the meaning of locality of reference. List the types
It is the tendency of aprocessor to access the same set of memory locations repetitively over a short period of time **Types:** 1. Temporal 2,Spatial
.

g) What is the role of interface
Interface provides a method for transferring information between internal storage (such as memory and CPU registers) and external I/O devices
.

h) Define is port.
A port is a physical docking point using which an external device can be connected to the computer

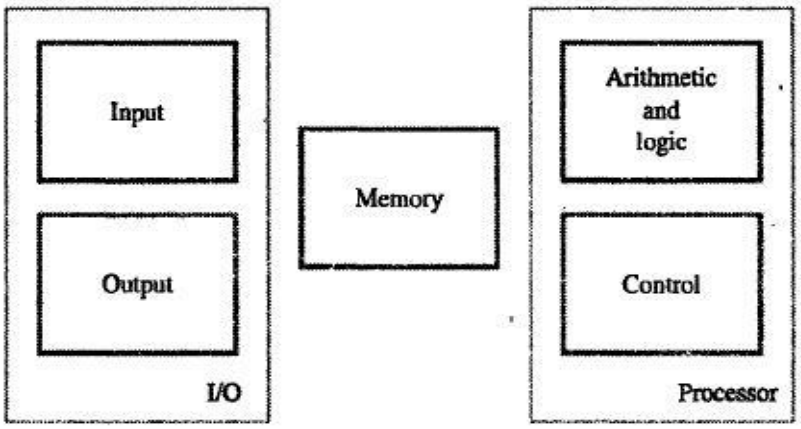
i) What is memory mapped I/O?
Memory-mapped I/O uses the same address space to address both memory and I/O devices. The memory and registers of the I/O devices are mapped to (associated with) address values

j) What is the full form of DMA?
Direct Memory Accesses

UNIT I

2 a) Explain functional units in a computer 7M

A computer consists of five functionally independent main parts: input, memory, arithmetic and logic, output, and control units.



Input Unit:-

Computer accept coded information through input units, which read the data. The most well-known input device is the keyboard. Whenever a key is pressed, the corresponding letter or digit is automatically translated into its corresponding binary code and transmitted over a cable to either the memory or the processor.

Memory Unit:-

The function of the memory unit is to store programs and data. There are two classes of storage, called primary and secondary.

Primary storage is a fast memory that operates at electronic speeds. Programs must be stored in the memory while they are being executed. The memory contains a large number of semiconductor storage cells, each capable of storing one bit of information. These cells are rarely read or written as individual cells but instead are processed in groups of fixed size called words. The memory is organized so that the contents of one word, containing n bits, can be stored or retrieved in one basic operation.

Instructions and data can be written into the memory or read out under the control of the processor. It is essential to be able to access any word location in the memory as quickly as possible. Memory in which any location can be reached in a short and fixed amount of time after specifying its address is called random access memory (RAM). The time required to access code word is called the memory access time. This time is fixed, independent of the location of the word being accessed.

secondary storage is used when large amounts of data and many programs have to be stored, articularly for information that is accessed infrequently. A wide selection of secondary storage devices is available, including magnetic disks and tapes and optical disks (CD-ROMs).

Arithmetic and Logic Unit:-

Any other arithmetic or logic operation, for example, multiplication, division, or comparison of numbers, is initiated by bringing the required operands into the processor, Where the operation is performed by the ALU. When operands are brought into the processor, they are stored in high-speed storage elements called registers. Each register can store one word of data. Access times to registers are somewhat faster than access times to the fastest cache unit in the memory hierarchy.

The control and the arithmetic and logic units are many times faster than other devices connected to a computer system. This enables a single processor to control a number of external devices such as keyboards, displays, magnetic and optical disks, sensors, and mechanical controllers.

Output Unit:-

Its function is to send processed results to the outside world. Some units, such as graphic displays, provide both an output function and an input function.

Control Unit:-

The memory, arithmetic and logic, and input and output units store and process in formation and perform input and output operations. The operation of these units must be coordinated in some way. This is the task of the control unit. The control unit is effectively the nerve center that sends control signals to other units and senses their states.

I/O transfers, consisting of input and output operations, are controlled by the in structions of I/O programs that identify the devices involved and the information to be transferred. However, the actual timing signals that govern the transfers are generated by the control circuits. Timing signals are signals that determine when a given action is to take place. Data transfers between the processor and the memory are also controlled by the control unit through timing signals. It is reasonable to think of a control unit as a well-defined, physically separate unit that interacts with other parts of the machine.

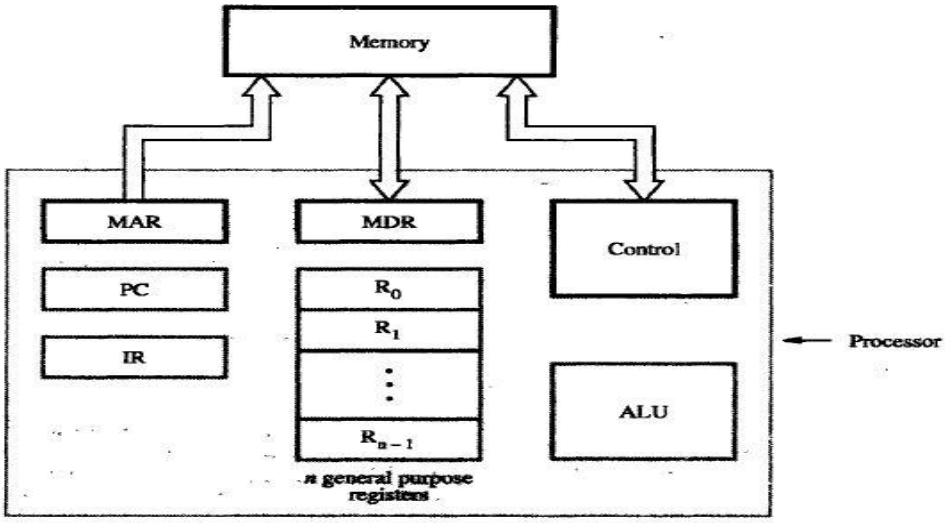
b) List the operations of computer. 3M

1. The computer accepts the information in the form of programs and data through an input unit and stores in the memory.
2. Information stored in the memory is fetched, under the program control, into an arithmetic and logic unit, where it is processed.
3. Processed information leaves the computer Through an output unit.
4. All the activities inside the machine are directed by control Unit

(OR)

3 a) Describe the communication between Processor and Memory unit 7M

The processor contains a number of registers used for several different purposes. The instruction register (IR) holds the instruction that is currently being executed. Its output is available to the control circuits, which generate the timing signals that control the various processing elements involved in executing the instruction The program counter (PC) is another specialized register It keeps track of the execution of a program. It contains the memory address of the next instruction to be fetched and executed During the execution of an instruction, the contents of the PC are updated to correspond to the address of the next instruction to be executed. It is customary to say that the PC points to the next instruction that is to be fetched from the memory. Besides the IR and PC, Figure shows n general-purpose registers through R1



Finally, two registers facilitate communication with the memory. These are the memory address register (MAR) and the memory data register (MDR). The MAR holds the address of the location to be accessed The MDR contains the data to be written into or read out of the addressed location.

Let us now consider some typical operating steps. Programs reside in the memory and usually get there through the input unit. Execution of the program starts when the PC is set to point to the first instruction of the program. The contents of the PC are transferred to the MAR and a Read control signal is sent to the memory. After the time required to access the memory lapses, the addressed word (in this case, the first instruction of the program) is read out of the memory and loaded into the MDR Next, the contents of the MDR are transferred to the IR. At this point, the instruction is ready to be decoded and executed.

If the instruction involves an operation to be performed by the ALU, is necessary to obtain the

required operands. If an operand resides in the memory (it could also be in a general-purpose register in the processor), it has to be fetched by sending its address to the MAR and initiating a Read cycle. When the operand has been read from the memory into the MDR, it is transferred from the MDR to the ALU. After one or more operands are fetched in this way, the ALU can perform the desired operation. If the result of this operation is to be stored in the memory, then the result is sent to the MDR. The address of the location where the result is to be stored is sent to the MAR, and a Write cycle is initiated. At some point during the execution of the current instruction, the contents of the PC are incremented so that the PC points to the next instruction to be executed. Thus, as soon as the execution of the current instruction is completed, a new instruction fetch may be started.

b) Write the steps needed to execute the machine instruction ADD R1, R2, R3. 3M

1. Transfer the contents of register PC to register MAR
2. Issue a Read command to memory, and then wait until it has transferred the requested word into register MDR
3. Transfer the instruction from MDR into IR and decode it
4. Transfer contents of R1 and R2 to the ALU
5. Perform addition of two operands in the ALU and transfer answer into R3
6. Transfer contents of PC to ALU
7. Add 1 to operand in ALU and transfer incremented address to PC

UNIT II

4 a) Explain the use of Interrupts in OS 7M

The operating system (OS) is responsible for coordinating all activities with in the computer.It makes extensive use of interrupts to perform I/O operations and communicate with and control execution of programs.The encrypt mechanism enables the OS to assign priorities,switch from one user to another, implement ,security and the production feautures and coordinate I/O activities.

When an application programs needs an Input/Output operation it points to the data to be transferred and ask the OS to perform the operation.Most process have several diff. s/w interrupt instructions,each with it own interrupt vector.They can be used to call diff. parts of OS ,depending on the service being requested.

The processor switches its operation to supervisor mode at the time it accepts an interrupt request.It does so by setting a bit in the processor status register after saving the old contents of that register on the stack.when an application program calls the OS by a s/w interrupt instruction , the processor automatically will switches to super visor mode,giving the OS complete access to the computers resources.when the OS executes a return-form-interrupt instruction, the process status word belonging to the app. Program is store from the stack.As a result, the processor back to the user mode.

Multitasking is the mode of operation in which a processor executes several user programs at a same time.A common OS technique that makes this possible is called **time – slicing.** With this technique each program runs for a short period is called Time-Slice,then another program runs for its time-slice.

b) List the steps of initialization process of Interrupt. 3M

1. Load the starting address of the interrupt-service routine in location INTVEC.
2. Load the address LINE in a memory location PNTR. The interrupt-service routine will use this location as a pointer to store the input characters in the memory.
3. Enable keyboard interrupts by setting bit 2 in register CONTROL to 1.
4. Enable interrupts in the processor by setting to 1 the IE bit in the processor status register, PS.

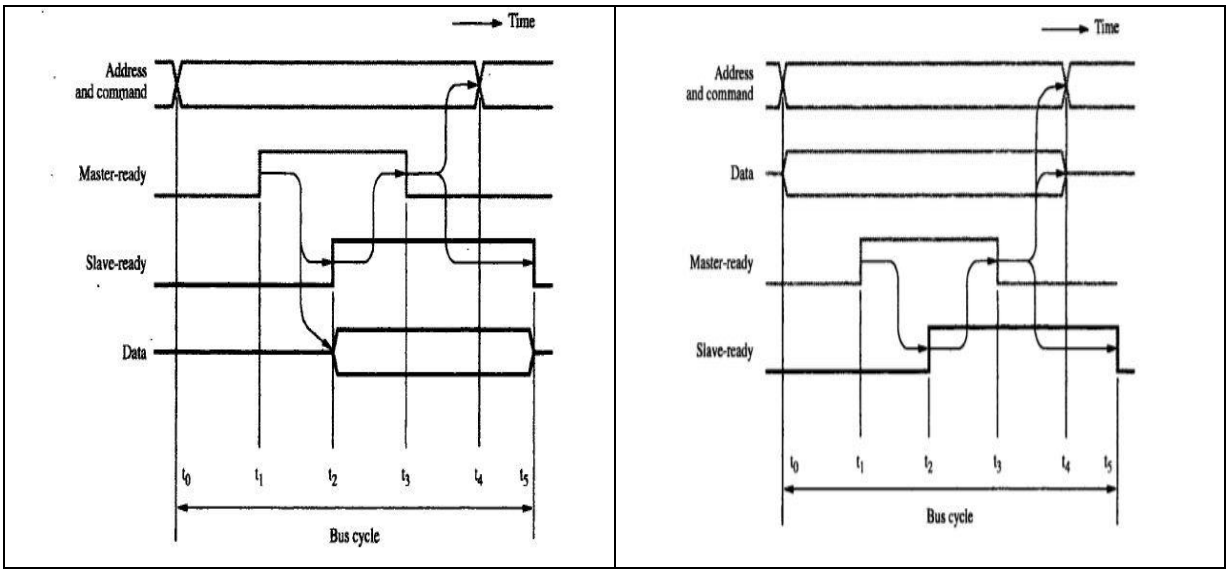
Once this initialization is complete, typing a character on the keyboard will cause an interrupt request to be generated by the keyboard interface. The program being executed at that time will be interrupted and the interrupt-service routine will be executed. This routine has to perform the following tasks:

1. Read the input character from the keyboard input data register. This will cause the interface circuit to remove its interrupt request.
2. Store the character in the memory location pointed to by PNTR, and increment PNTR
3. When the end of the like is reached, disable keyboard interrupts and inform program Main.
4. Return from interrupt.

(OR)

5 Draw the timing diagram of an input data transfer using the handshake scheme and describe the sequence of events 10

The first is asserted by the master to indicate that it is ready for a transaction, and the second is a response from the slave. In principle, a data transfer controlled by a handshake protocol proceeds as follows. The master places the address and command information on the bus. Then it indicates to all devices that it has done so by activating the Master-ready line. This causes all devices on the bus to decode the address. The selected slave performs the required operation and informs the processor it has done so by activating the Slave-ready line. The master waits for Love-ready to become asserted before it removes its signals from the box.In the case of a read operation, it also strobes the data into its input buffer. An example of the timing of an input data transfer using the handshake scheme is given in Figure, which depicts the following sequence of events:



t0- The master places the address and command information on the box, and all devices on the bus begin to decode this information.

t1 - The master sets the Mast teddy line to 1 to inform the 10 devices that the address and command information is ready. The delay-is intended to allow for any skew that may occur on the bus. Skew occur when two signals simultaneously transmitted from one source arrive at the destination different times. This happens because different lines of the bus may have different. propagation speeds Do to guarantee that the Master ready signal does not arrive any device head of the address and command information the delay.

t1-t0 should be larger than the maximum possible bas skew (Note that in the synchronous case bus is

accounted for as a part of the maximum propagation delay Wen the address information arrives at any device, it is decoded by the interface circuitry Sufficient time should be allowed for the interface circuitry to decode addresses. The delay needed can be included in period t1-t0.

t2 - The selected slave, having decoded the address and command information, performs the required input operation by placing the data from its data register on the data lines. At the same time, it sets the Slave ready signal to 1. If extra delays are introduced by the interface circuitry before it places the data on the bus, the slave must delay the Slave-ready signal accordingly. The period – depends on the distance between the master and the slave and on the delays introduced by the slave's circuitry. It is this variability that gives the bus its asynchronous nature.

t3-The Slave-ready signal arrives at the master, indicating that the input data are available on the bus. However, since it was assumed that the device interface transmits the Slave-ready signal at the same time that it places the data on the bus, the master should allow for bus skew. It must also allow for the setup time needed by its input buffer. After a delay equivalent to the maximum bus skew and the minimum setup time, the master strobes the data into its input buffer. At the same time, it drops the Master-ready signal, indicating that it has received the data.

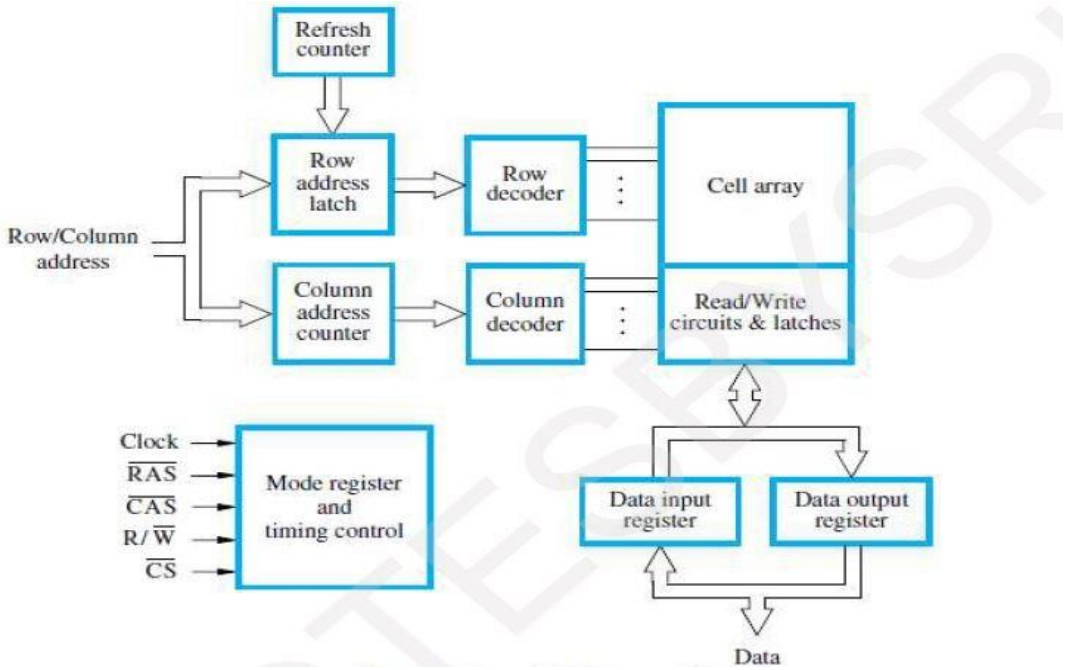
t4-The master removes the address and command information from the bus. The delay between 1, and is again intended to allow for bus skew. Erroneous addressing may take place if the address, as seen by some device on the bus, starts to change while the Master-ready signal is still equal to 1.

t5-When the device interface receives the 1 to 0 transition of the Master-ready signal, it removes the data and the Slave ready signal from the bus. This completes the input transfer.

UNIT III

6 a) Draw the block diagram of SDRAM and Explain. 7M

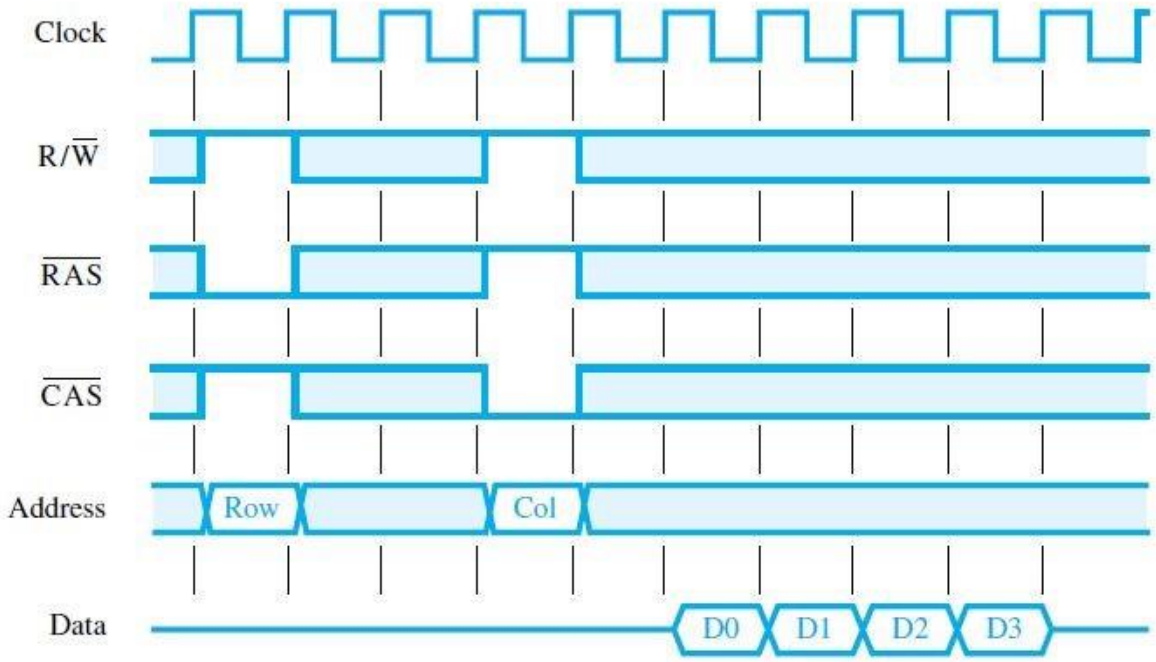
In the early 1990s, developments in memory technology resulted in DRAMs whose operation is synchronized with a clock signal. Such memories are known as synchronous DRAMs (SDRAMs).The address and data connections are buffered by means of registers.



We should particularly note that the output of each sense amplifier is connected to a latch. A Read operation causes the contents of all cells in the selected nkw to loaded into these latches. But, if an access is made for refreshing purposes only, it will not change the contents of these latches; it will merely refresh the contents of the cells. Data held in the latches that correspond to the selected column(s) are transfened into the data output register, thus becoming available on the data output pins.

SDRAMs have several different modes of operation, which can be selected by writing

control information into a mode register. For example, bunt operations of different lengths can be specified. The burst operations use the block transfer capability described above as the fast page mode feature. In SDRAMs, it is not necessary to provide externally generated pulses on the CAS line to select successive columns. The necessary control signals are provided internally using a column counter atid the clock signal. New data can be placed on the data lines in each clock cycle. All actions are triggered by the rising edge of the clock.

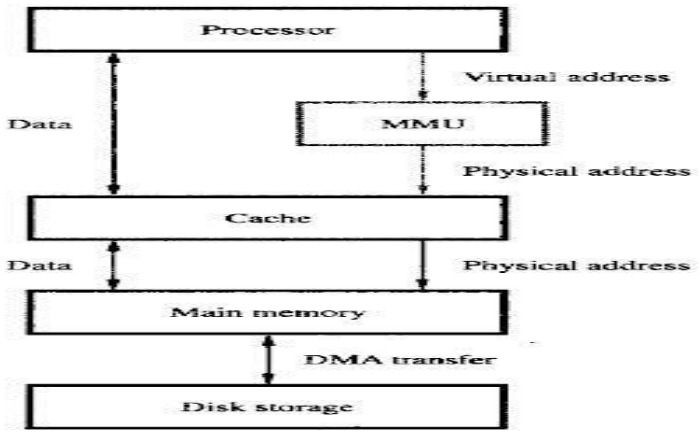


b) Write the uses of Cache memory 3M

1.Cache memory is used to increase the performance of the PC.
2.It holds data and instructions retrieved from RAM to provide faster access to the CPU
3.It holds frequently requested data and instructions so that they are immediately available to the CPU when needed.

(OR)

7 a) Write how Virtual Memory Translated into physical address. 7M



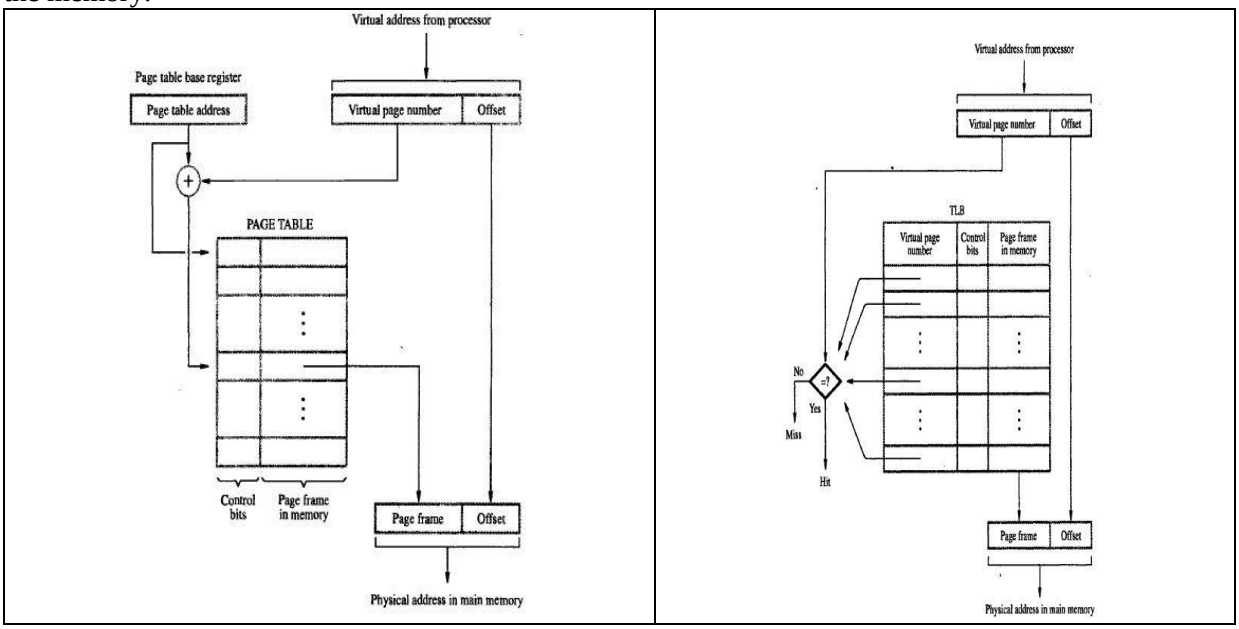
The typical organiztion that implements virtual memory. A special hardware unit, called the Memory Management Unit(MMU) translates virtual addresses into physical addresses .When the desired data are in the main memory. If the data are not in the main memory, the MMU causes the operating system to bring the data into the memory from the disk. Transfer of data between the disk and the main memory is performed using the DMA scheme.

A simple method for translating virtual addresses into physical addresses is to assume that all programs and data are composed of fixed – length units called pages,each of which consists of a block

of words that occupy contiguous locations in the main memory.Pages commonly range from 2k to 16k bytes of length . Pages should not be too small,because the access time of a magnetic disk is much longer than the access time of the main memory.

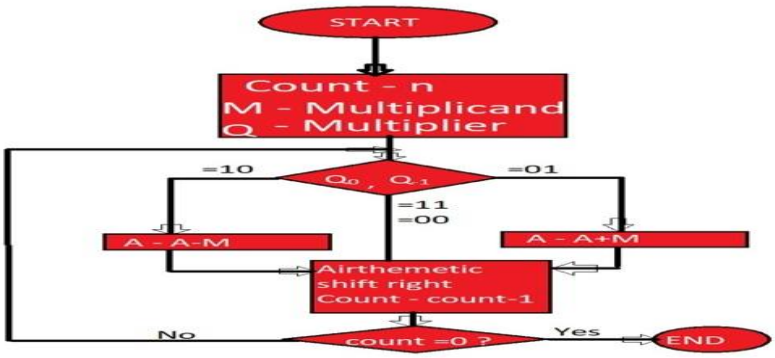
Each virtual address generated by the processor, whether it is for an instruction fetch or an operand fetch or an operand fetch/store operation, is interpreted as a virtual page number (high – order bits) followed by an offset (low- order bits) that specifies the location of a particular byte(or word) with in a page. Information about the main memory location of each page is kept in a page table. This information includes the main memory address where the page is stored and the current status of the page. An area in the main memory that can hold one page is called a page frame. The starting address of the page table is kept in a page table base register . By adding the virtual page number to the contents of this register, the address of the corresponding entry in the page table is obtained . The contents of this location give the starting address of the page if that page currently resides in the main memory.

Each entry in the page table also includes some control bits that describe the status of the page while it is in the main memory. One bit indicates the validity of the page, that is whether the page is actually loaded in the main memory. This bit allows the operating system to invalidate the page without actually removing it. Another bits indicates whether the page has been modified during its residency in the memory.



The page table information is used by the MMU for every read and write access , so ideally the page table should be situated with in the MMU.Unfortunately , the page table may be rather large , and since the MMU is normaly implemented as part of the processor clip ,it is impossible to include a complete page table on this chip.Therefor,the page table is kept in the main memory .Hwever a copy of a small portion of the page table can be accommodated with the MMU.This portion consists of the page table entires that correspond to the most recently accessed pages.A small cache , usually called the Translation Lookaside Buffer (TLB) is incorporated into the MMU for this purpose.

b) Draw the flow chart for Booth's algorithm to multiply two binary numbers. 3M



8 a) Explain the execution of instruction Add (R3), R1 5M

Let us now put together the sequence of elementary operations required to execute one instruction. Consider the instruction

Add (R3),R1

Which adds the contents of a memory location pointed by R3 to register R1.Executing this instruction requires the following actions:

1.Fetch the instruction .

2 . Fetch the first operand (the contents of the memory location pointed to by R3).

3.Perform the addition.

4.Load the result into R1.

The below figure gives the sequence of control steps required to perform these operations for the single - bus architecture .

In step 1,the instruction fetch operation is initiated by loading the contents of the PC into the MAR and sending a Read request to the memory . The select signal is set to Select 4,which causes the multiplexer MUX to select the constant 4. This value is added to the operand at input B, which is the contents of PC, and the result is stored in register Z. The updated value is moved from register Z back into the PC during step 2,while waiting for the memory to respond. In step 3,the word fetched from the memory is loaded into the IR.

Steps 1 through 3 constitute the instruction fetch phase, which is the same for all instructions. The instruction decoding circuit interprets the contents of the IR at the beginning of step4.The contents of register R3 are transferred to the MAR in step 4,and a memory read operation is indicated. Then

Step	Action
1	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}
2	Z_{out}, PC_{in}, Y_{in}, WMFC
3	MDR_{out}, IR_{in}
4	$R3_{out}$, MAR_{in}, Read
5	$R1_{out}$, Y_{in}, WMFC
6	MDR_{out}, SelectY, Add, Z_{in}
7	Z_{out}, $R1_{in}$, End

the contents of R1 are transferred to register Y in step 5,to prepare for the addition operation. When the Read operation is completed , the memory operand is availablein register MDR, and the addition operation is performed in step 6.The contents of MDR is gated to the bus, and thus also to the B input of the ALU ,and register Y is selected as the second input to the ALU by choosing Select Y .The Sum is stored in register Z, then transferred to R1 in step 7 .The End signal causes a new instruction fetch cycle to begin by returning to step1.

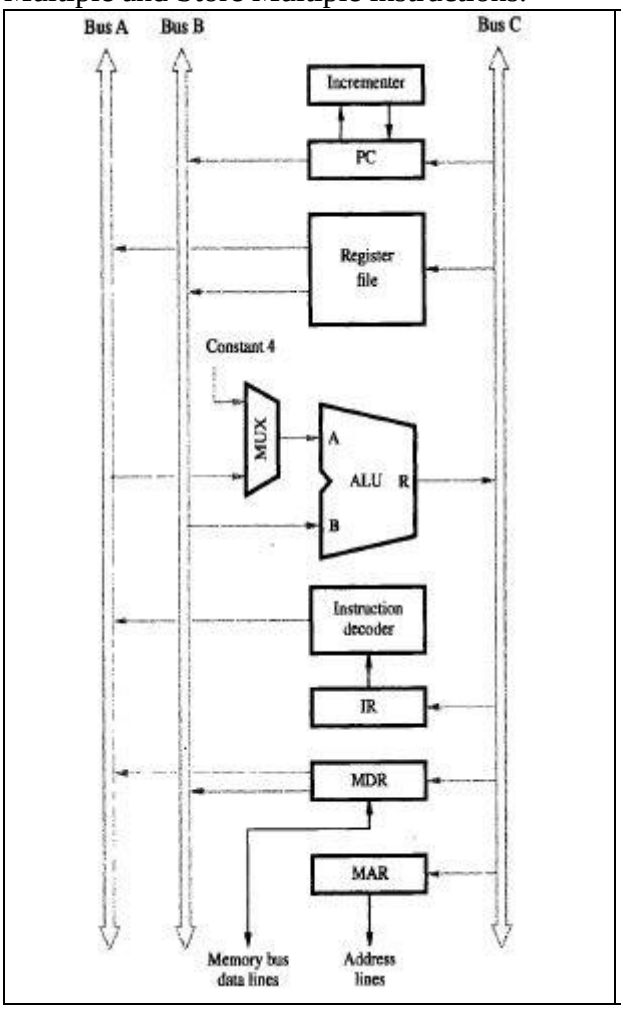
b) Describe multi bus organization . 5M

We used the simple single bus structure to illustrate the basic ideas. The resulting control sequences are quite long because only one data item can be transferred over the bus in clock cycle. To reduce the number of steps needed, most commercial processors provide multiple internal paths that enable several transfers to take place in parallel.

Fig depicts a three-bus structure used to connect the registers and the ALU of a processor . All general purpose registers are combined into a signal block called the register file .In VLSI technology , the most efficient way to implement way to implement a number of registers in the form of an array of memory cells similar to those used in implementation of random-access memories (RAMs).The register file have three ports. There are two outputs ,allowing the contents of two registers to be accessed simultaneously and have their contents places on buses

A and B. The third port allows the data on bus C to be loaded into a third register during the same clock cycle.

A second feature in the introduction of the introduction of the incrementor unit, which is used to increment the PC by 4.Using the incremented eliminates the need to add 4 to the PC using main ALU , as was done .The source for the constant 4 at the ALU input multiplexer is still useful . I can be used to increment other addresses , such as the memory addresses in Load Multiple and Store Multiple instructions.



Step	Action
1	PC_{out}, R=B, MAR_{in}, Read, IncPC
2	WMFC
3	MDR_{outB}, R=B, IR_{in}
4	$R4_{outA}$, $R5_{outB}$, SelectA, Add, $R6_{in}$, End

Figure 7.9 Control sequence for the instruction Add R4,R5,R6 for the three-bus organization in Figure 7.8.

Consider the three operand instruction

Add R4,R5,R6

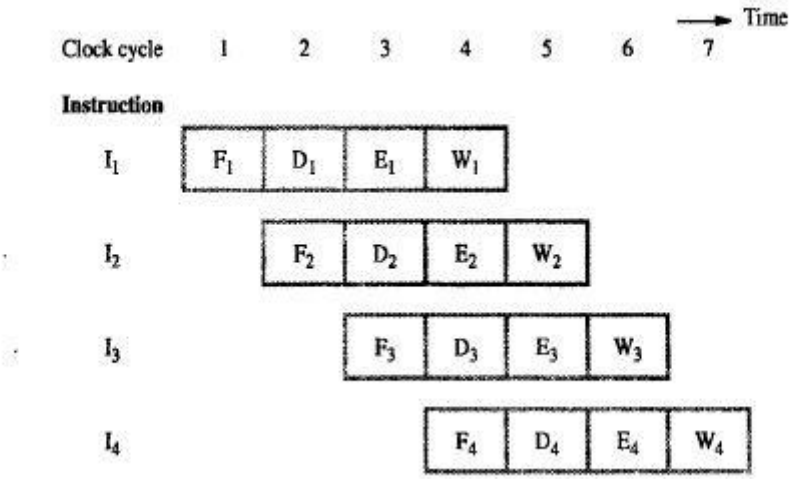
The control sequence for executing this instruction ,the contents of PC are passed through ALU,using the R=B control signal , and loaded into the MAR to start a memory read operation .At the same time of PC is incremented by 4.Note that the value loaded into MAR is the original contents of the PC.The incremented value is loaded into the PC at the end of the clock cycle and will not affect the contents of MAR .In step2 , the processor waits for MFC and loads the data received into MDR , then transfers them to IR in step3.Finally , the execution phase of the instruction requires only one control step to complete step4.

By providing more paths for data transfer a significant reduction in the number of clock cycle needed to execute an instruction is archived.

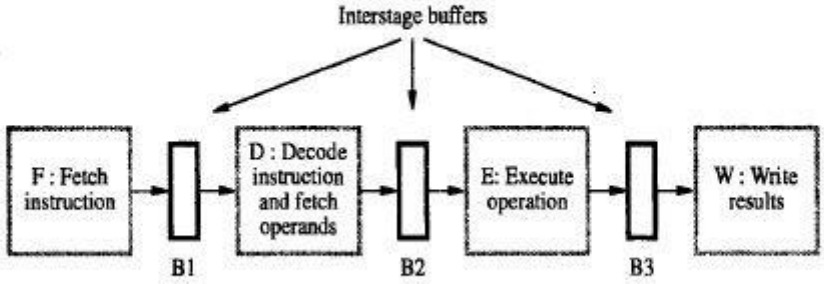
(OR)

9 a) What is the role of cache memory in pipelining. 5M

Each stage in a pipeline is expected to complete its operation in one clock cycle. Hence,the clock period should be sufficiently long to complete the task being performed in any stage.If different units require different amount of time,the clock period must allow the longest task to be completed.A unit that completes its task early is idle for the remainder of the clock period.Hence pipelining is most effective in improving



(a) Instruction execution divided into four steps



(b) Hardware organization

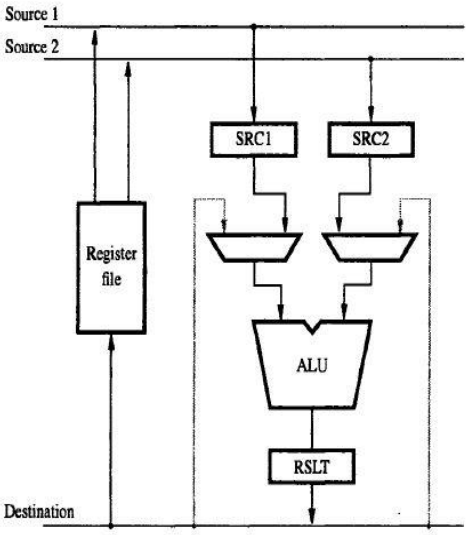
Performance if the task being performed in different stages require about the same amount of time.

This consideration is particularly important for the instruction fetch step, which is assigned one clock period .The clock cycle has to be equal to or greater than the time needed to complete a fetch operation. However, the access time of the main memory may be as much as ten times greater than the time needed to perform basic pipeline stage operations inside the processor, such as adding two numbers. Thus if each instruction fetch required access to the main memory pipelining would be of little value.

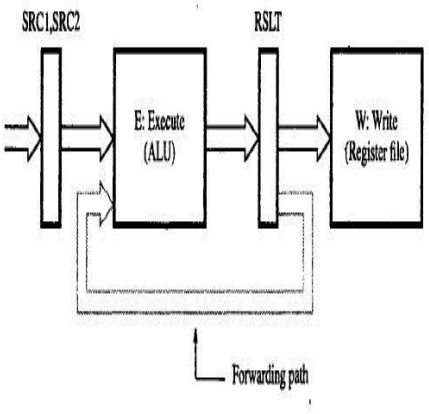
The use of cache memories solves the memory access problem. In particular , when cache is included on the same chip as the processor ,access time to the cache is usually the same as the time needed to perform basic operations inside the processor . This makes it possible to divide instruction fetching and processing into steps that are more or less equal in duration. Each of these steps is performed by a different pipeline stage, and the clock period is chosen to correspond to the longest one.

b) Explain Operand Forwarding. 5M

The data hazard just describes arises because one instruction, instruction I2 ,is waiting for data to be written in the register file. However, these data are available at the output of the ALU once the execute stage completes step E1.Hence , the delay can be reduced , or possibly eliminated, if we arrange for the result of instruction I1 to be forward directly for use in step E2.Fig shows a part of the processor data path involving the ALU and the register file. This arrangement is similar to the three-bus structure, except the registers SRC1,SRC2 and RSLT have been added. These registers constitute the inter stage buffers needed for pipelined operation, as illustrated in fig.



(a) Datapath



(b) Position of the source and result registers in the processor pipeline

With registers SRC1,SRC2 are part of buffer B2 and RSLT is part of B3.The data forwarding mechanism is provided by the blue connection lines. The two multiplexes connected at the inputs to the ALU allow the data to the destination bus to be selected instead of the contents of either the SRC1 or SRC2 register.

When instructions are executed in the data path , the operations performed in each block cycle as follows. In the next cycle , the product produced by instruction I1 is available in register RSLT , and because of the forwarding connection , it can be used in step E2.Hence execution of I2 proceeds without interruption.