Scheme of Solution

Internal Assessment Test 3 – November 2019

Sub:	COMPUTER ORGANIZATION							Code:	18CS34
Date:	18/10/2019	Duration	90 mins	Max Marks:	50	Sem:	III	Branch:	**ISE**

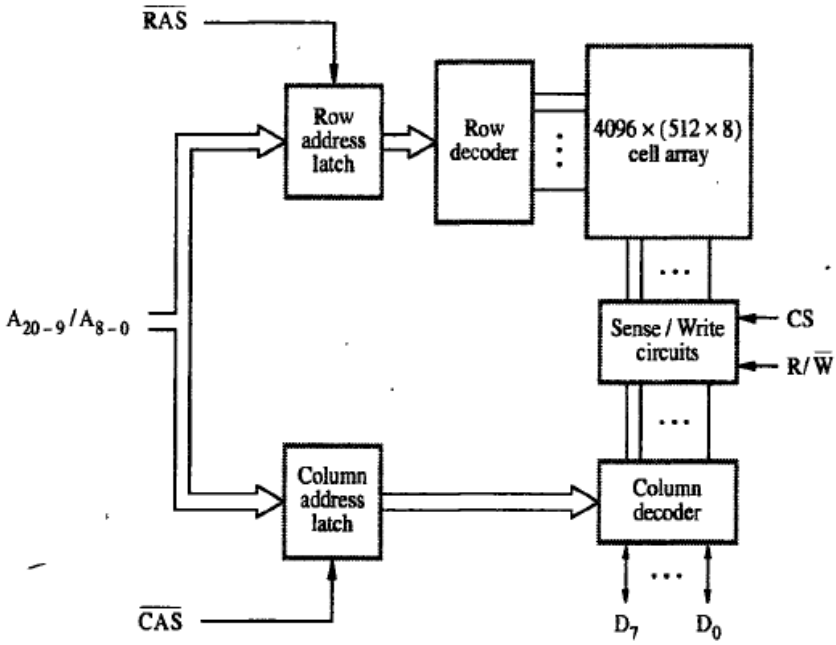
Answer Any **FIVE FULL** Questions

1(a) With a neat diagram, explain the organization of 2M X 8 dynamic memory chip.

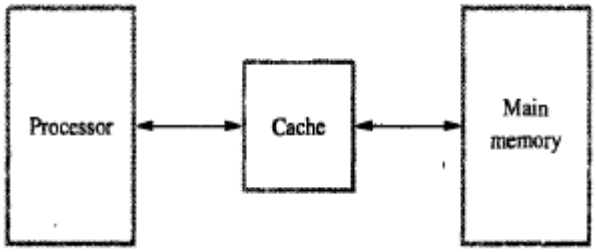
Organized as 4kx4k array. 4096 cells in each row are divided into 512 groups of 8. Each row can store 512 bytes. 12 bits to select a row, and 9 bits to select a group of 8 bits in a row. Total of 21 bits. (2 MB). Reduce the number of bits by multiplexing row and column addresses. First apply the row address, RAS signal latches the row address. Then apply the column address, CAS signal latches the address. Timing of the memory unit is controlled by a specialized unit which generates RAS and CAS. This is **asynchronous** DRAM.

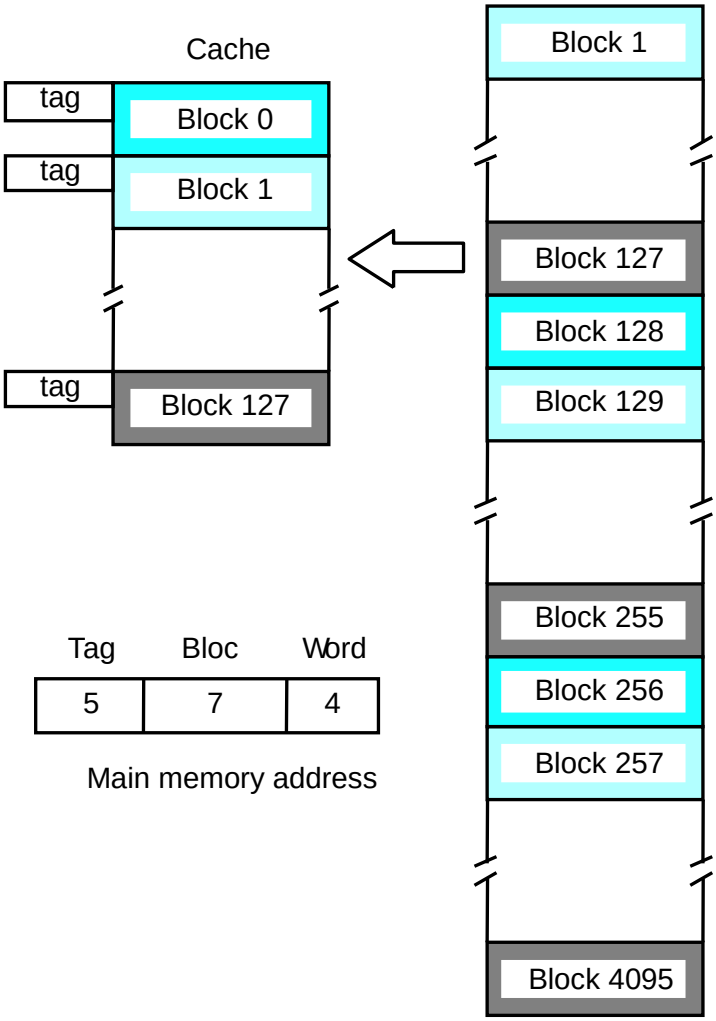
- All the contents of a row are selected based on a row address.Particular byte is selected based on the column address.
- Add a latch at the output of the sense circuits in each row. All the latches are loaded when the row is selected. Different column addresses can be applied to select and place different bytes on the data lines.
- Consecutive sequence of column addresses can be applied under the control signal CAS, without reselecting the row. Allows a block of data to be transferred at a much faster rate than random accesses. A small collection/group of bytes is usually referred to as a block. This transfer capability is referred to as the fast page mode feature.

3 Marks



2 Marks

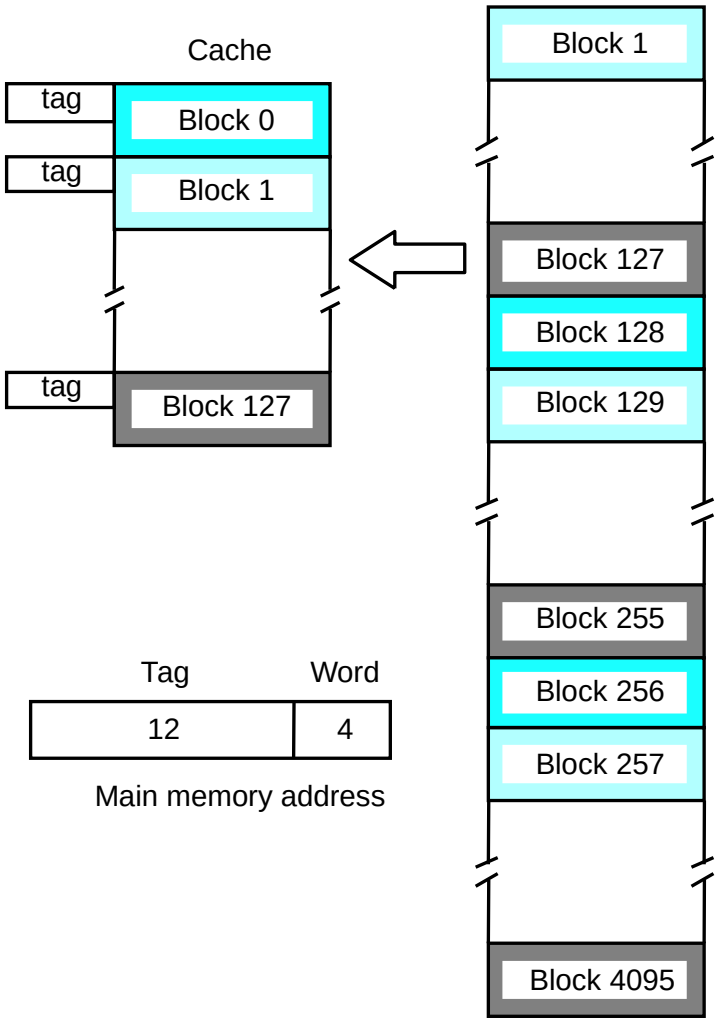
1 (b)	What is ROM memory? What are the different types of ROM. Explain? Read-Only Memory: Data are written into a ROM when it is manufactured. Programmable Read-Only Memory (PROM): Allow the data to be loaded by a user. Process of inserting the data is irreversible. Storing information specific to a user in a ROM is expensive. Providing programming capability to a user may be better. Erasable Programmable Read-Only Memory (EPROM): Stored data to be erased and new data to be loaded. Flexibility, useful during the development phase of digital systems. Erasable, reprogrammable ROM. Erasure requires exposing the ROM to UV light. Electrically Erasable Programmable Read-Only Memory (EEPROM): To erase the contents of EPROMs, they have to be exposed to ultraviolet light. Physically removed from the circuit. EEPROMs the contents can be stored and erased electrically Flash memory: It is similar approach to EEPROM. Read the contents of a single cell, but write the contents of an entire block of cells. Flash devices have greater density. Higher capacity and low storage cost per bit. Power consumption of flash memory is very low, making it attractive for use in equipment that is battery-driven. Single flash chips are not sufficiently large, so larger memory modules are implemented using flash cards and flash drives. 1 X 5 = 5 Marks
2(a)	What is a Cache? Explain different cache memory mapping functions with the help of diagrams. Cache is used to improve the speed of data processing and placed in between CPU and Main memory. Block diagram is given below.  **Direct Mapping.** Assume a simple processor *example*: *Cache* consisting of 128 blocks of 16 words each. Total size of cache is 2048 (2K) words. Main memory is addressable by a 16-bit address. Main memory has 64K words. Main memory has 4K blocks of 16 words each. Consecutive addresses refer to consecutive words.



Block j of the main memory maps to j modulo 128 of the cache.
Memory Blocks 0,128,256, …maps to cache block 0.
Memory Blocks 1, 129, 257, …maps to cache block 1. and so on.
More than one memory block is mapped onto the same position in the cache.
May lead to contention cache blocks even if the cache is not full.
Resolve the contention by allowing new block to replace the old block, leading to a trivial replacement algorithm.
Memory address is divided into three fields:
- Low order 4 bits determine one of the 16 words in a block.
- When a new block is brought into the cache, the next 7 bits determine which cache block this new block is placed in.
- High order 5 bits determine which of the possible 32 set of blocks of memory is currently present in the cache.
These are tag bits. = No of Blocks of memory / No of Blocks of cache = 4096/128 = 32 = 2^5
Simple to implement but not very flexible.

3 Marks

Associative Mapping.



Main memory block can be placed into any cache position.
Memory address is divided into two fields:

- Low order 4 bits identify the word within a block.
- High order 12 bits or tag bits identify a memory block when it is resident in the cache. Flexible, and uses cache space efficiently. Replacement algorithms can be used to replace an existing block in the cache when the cache is full.

Cost is higher than direct-mapped cache because of the need to search all 128 patterns to determine whether a given block is in the cache.

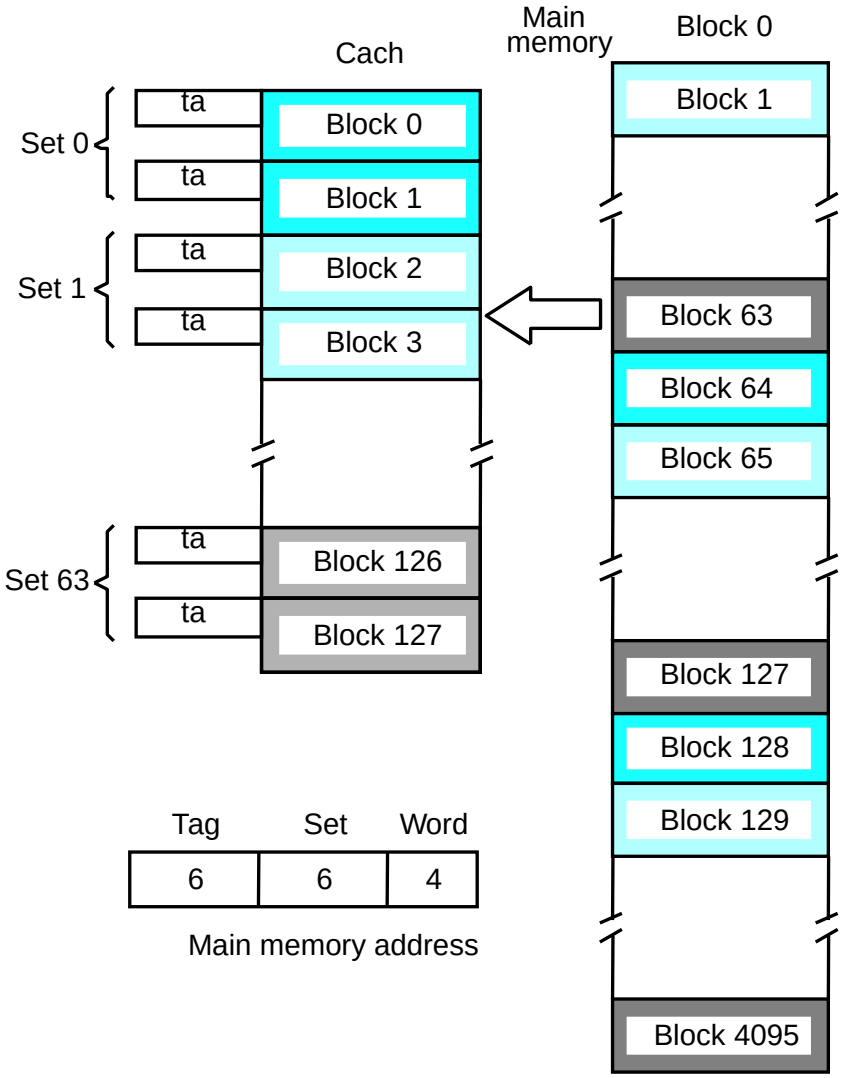
3 Marks

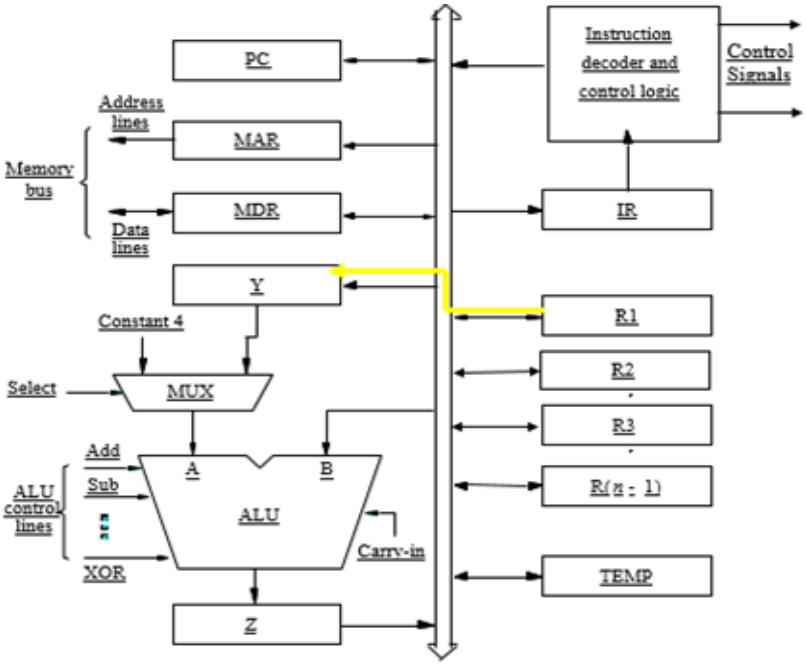
Set Associative mapping.

Set-associative mapping combination of direct and associative mapping.
Blocks of cache are grouped into sets. Mapping function allows a block of the main memory to reside in any block of a specific set.
Divide the cache into 64 sets, with two blocks per set. Memory block 0, 64, 128 etc. map to set 0, and they can occupy either of the two positions.
Memory address is divided into three fields:

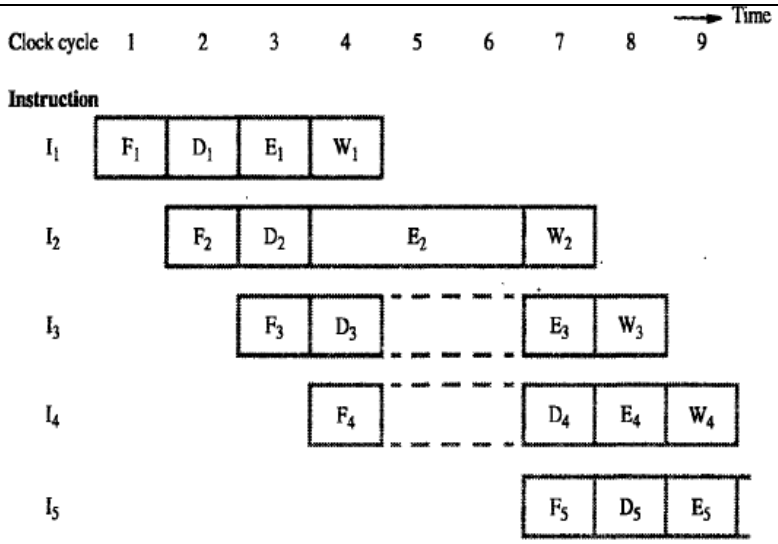
- Low order 4 bits identify the word within a block.
- 6 bit field determines the set number.
- High order 6 bit fields are compared to the tag fields of the two blocks in a set.

Tag size = No of Blocks of memory / No of sets of cache = 4096/64 = 64= 2^6

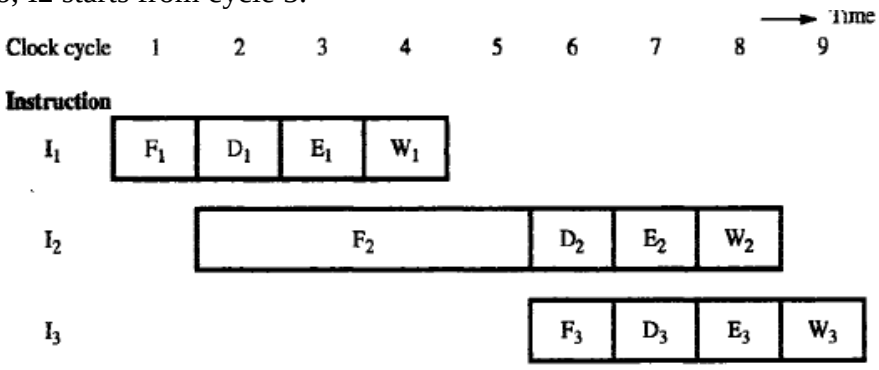
	- Number of blocks per set is a design parameter. - One extreme is to have all the blocks in one set, requiring no set bits (fully associative mapping - Other extreme is to have one block per set, is the same as direct mapping.  4 Marks
3(a)	Draw and explain a single bus organization of the data path inside a processor. - Single bus organization contains MAR, MDR registers. MDR has two inputs and two outputs. Data may be loaded into MDR wither from the memory bus or from the internal processor bus. Data stored in MDR may be place on either bus. - The input of MAR is connected to the internal bus, and its output is connected to the external bus. - The control lines of the memory bus are connected to the instruction decoder and control logic block. - Three registers: Y, Z and Temp are used in this design. - ALU must have only one input connection from the bus. The other input must be stored in a holding register called Y register. - A multiplexer selects among register Y and 4 depending upon select line. - One operand of a two-operand instruction must be placed into the Y register before the other operand must be placed onto the bus. Identical reasoning tells us that there

	must be an output register Z which collects the output of the ALU at the end of each cycle. - This way, there can be one operand in the Y register, one operand on the bus and the result stored in the Z register - The register, ALU and the interconnecting bus are collectively referred as *datapath.* - The following sequence are considered for instruction execution - Transfer a word of data from one processor to another or to the ALU - Perform an arithmetic or logic operation and store the result - Fetch the contents of a given memory location and load them into processor register - Store a word of data from a processor register into a given memory location. 5 Marks  5 Marks
4(a)	Explain the different actions required for execution of instruction ADD (R3), R1.Write the control sequence of it. To execute the instruction we must execute the following tasks: 1. Fetch the instruction. 2. Fetch the operand (contents of the memory location pointed to by *R3*.) 3. Perform the addition. 4. Load the result into *R1*. 3 Marks **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 3 Marks

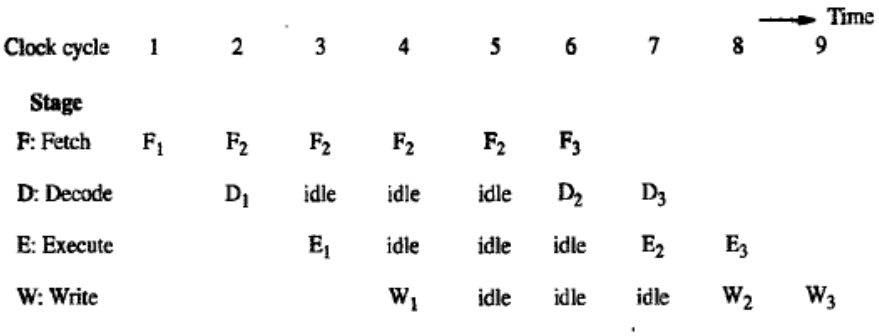
	Recall that: *PC* holds the address of the memory location which has the next instruction to be executed. *IR* holds the instruction currently being executed. Step 1 - Load the contents of *PC* to *MAR*. - Activate the *Read* control signal. - Increment the contents of the *PC* by 4. **- PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}.** Step 2 - Update the contents of the *PC*. - Copy the updated *PC* to Register *Y* (useful for Branch instructions). - Activate the control signal to load data from external bus to *MDR* - Wait for *MFC* from memory. **- Z_{out}, PC_{in}, Y_{in}, MDR_{inE}, WMFC** Step 3 - Place the contents of *MDR* onto the bus. - Load the *IR* with the contents of the bus. **- MDR_{out}, IR_{in}** Step 4: - Place the contents of Register *R3* onto internal processor bus. - Load the contents of the bus onto *MAR*. - Activate the *Read* control signal. **- $R3_{out}$, MAR_{in}, Read** Step 5: - Place the contents of *R1* onto the bus. - Load the contents of the bus into Register *Y (*Recall one operand in *Y).* - Wait for *MFC.* **- $R1_{out}$, Y_{in}, MDR_{inE}, WMFC** Step 6: - Load the contents of *MDR* onto the internal processor bus. - Select *Y*, and perform the addition. - Place the result in *Z.* **- MDR_{out}, SelectY, Add, Z_{in}.** Step 7: - Place the contents of Register *Z* onto the internal processor bus. - Place the contents of the bus into Register *R1*. **- Z_{out}, $R1_{in}$** 4 Marks
5(a)	What is instruction pipelining ?What are the different types of hazards involved in it Pipeline is a particularly effective way of organizing concurrent activity in a computer systems. Simple in design with assembly line operation. - *Pipeline Performance:* i) Sometimes pipeline stages may not be able to complete its processing task for a given instruction in the allotted time. - As per the following example, I2 requires three cycle to complete, from cycle 4 through cycle 6. Thus, in cycle 5 and 6, the write stage must be told to do nothing, because it has no data to work with. This stalled clock cycle known as *hazard.*



ii) Cache miss on pipeline is given as, I1 is fetched from the cache in cycle 1, and its execution proceeds normally. But in I2, the result data missed in cache. Thus instruction fetch unit must now suspend any further fetch request and wait for I2 to arrive. So, I2 starts from cycle 5.

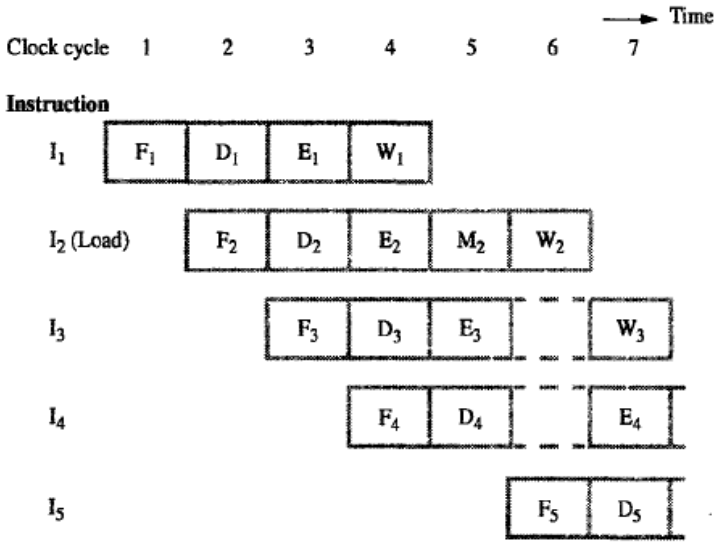


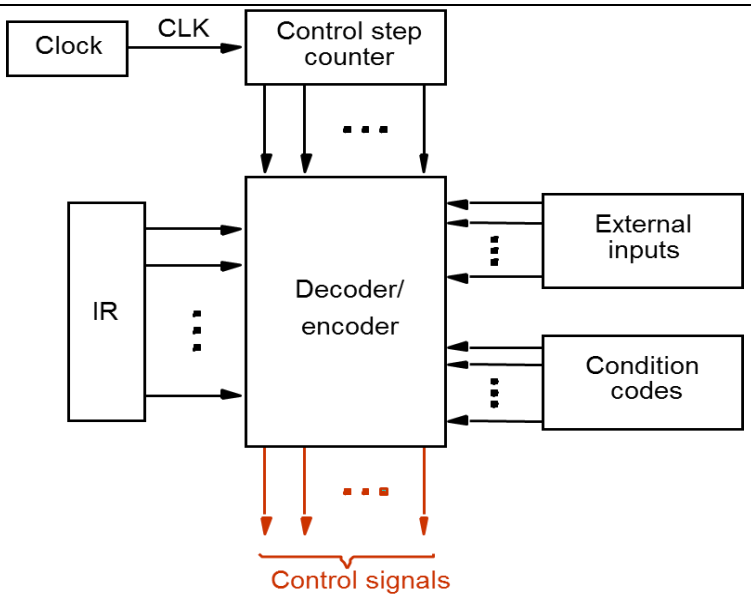
(a) Instruction execution steps in successive clock cycles

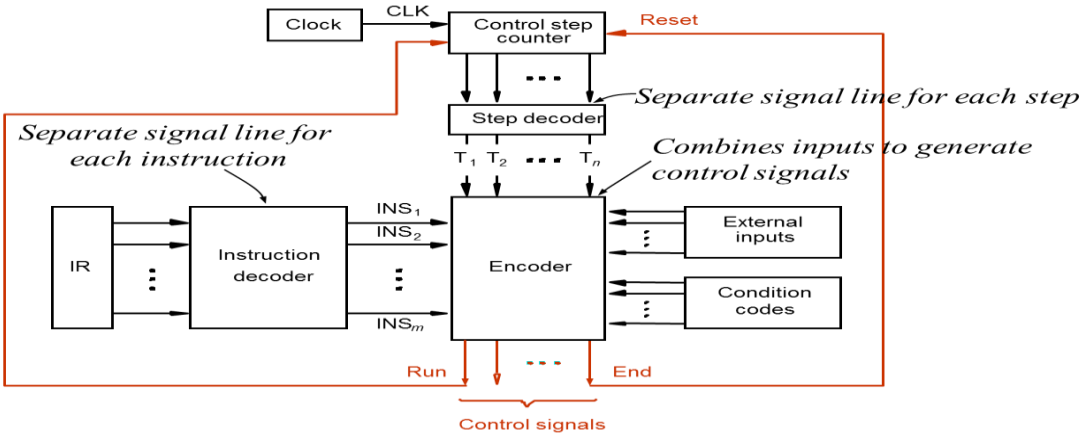


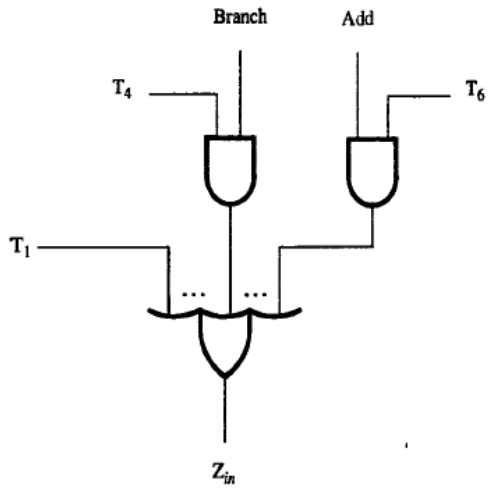
(b) Function performed by each processor stage in successive clock cycles

iii) Structural hazard – This situation will occur when two instruction require the use of a given hardware resource at the same time.
Example: LOAD X (R1), R2
- Four stage pipeline is used for this example. The memory address, X + [R1], is computed in step E2 in cycle 4, then memory access (M) takes place in cycle 5. The

	operand read from memory is written into R2 in cycle 6.  5 Marks
5(b)	Differentiate between SRAM and DRAM. Static RAMs (SRAMs): - Consist of circuits that are capable of retaining their state as long as the power is applied. - Volatile memories, because their contents are lost when power is interrupted. - Access times of static RAMs are in the range of few nanoseconds. - Capacity is low (each cell consists of 6 transistors) - However, the cost is usually high. Dynamic RAMs (DRAMs): - Do not retain their state indefinitely. - Contents must be periodically refreshed. - Contents may be refreshed while accessing them for reading. - Access time is longer than SRAM - Capacity is higher than SRAM (each cell consists of 1 transistor) - Cost is lower than SRAM 2.5 X 2 = 5 Marks
6(a)	Describe the function of Hardwired control unit with block diagram, decoding / encoding diagram and control signals. - Required control signals are determined by the following information: i) Contents of the control step counter. Determines which step in the sequence. ii) Contents of the instruction register. Determines the actual instruction iii) Contents of the condition code flags. Used for example in a BRANCH instruction. iv) External input signals such as MFC. - Control unit consists of a decoder/encoder block to accept the following inputs: - Control step counter. - Instruction Register. IR - Condition codes - External inputs. 3 Marks

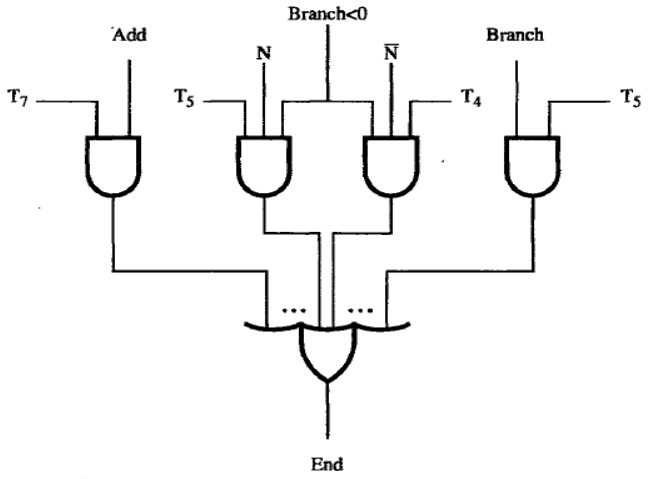


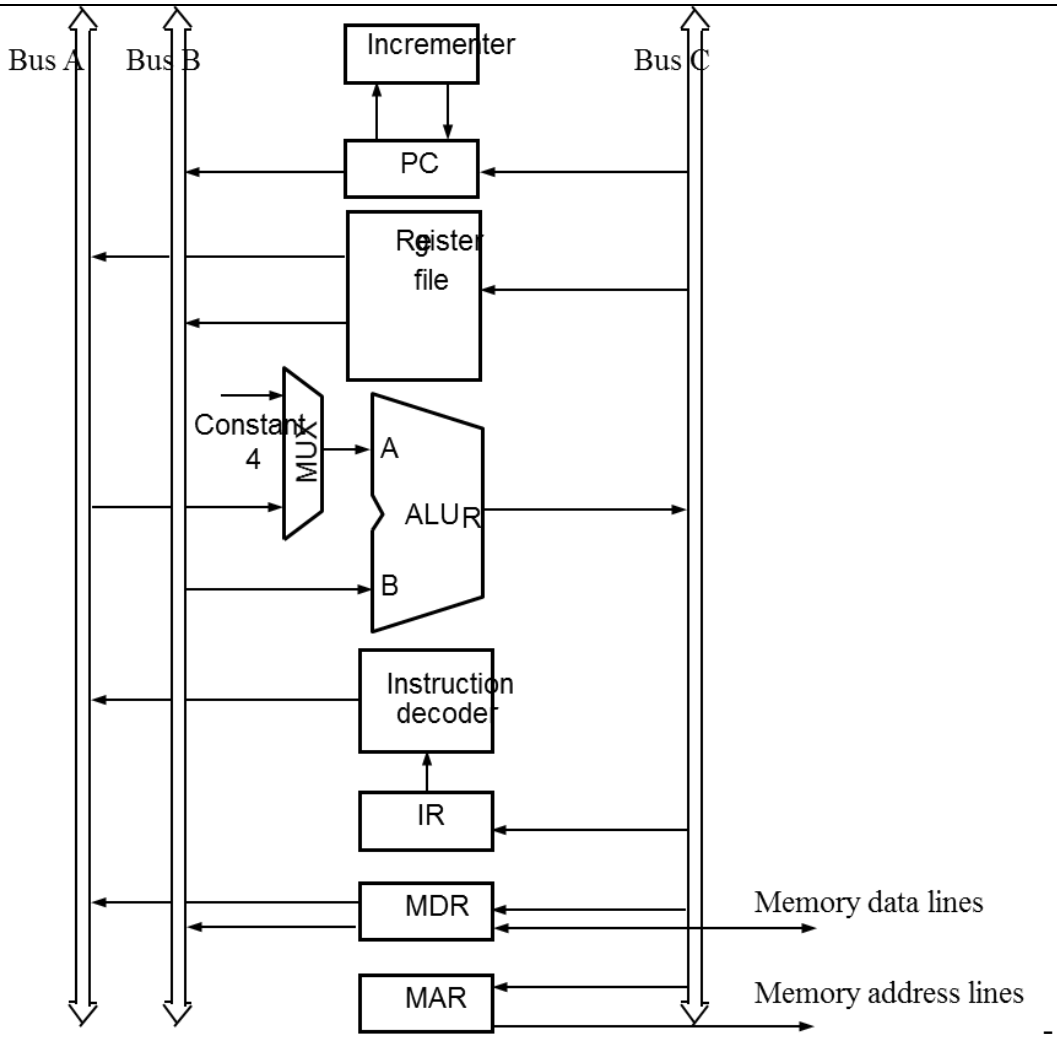




$$Z_{in} = T_1 + T_6 \cdot ADD + T_4 \cdot BR + \cdots$$

4 Marks

	- Suppose if Z_{in} is asserted: - During T_1 for all instructions. - During T_6 for *ADD* instruction. - During T_4 for unconditional *BRANCH* instruction  $\text{End} = T_7 \cdot \text{ADD} + T_5 \cdot \text{BR} + (T_5 \cdot N + T_4 \cdot \overline{N}) \cdot \text{BRN} + \cdots$ - In the above diagram, End signal starts a new instruction fetch cycle by resettling the control step counter to its starting value. It contains another signal called RUN. If it is active, the counter is incremented by one at the end of every clock cycle. - When RUN = 0, the counter stops counting. This is needed whenever the WMFC signal is issued, to cause the processor to wait for the reply from the memory. - Control hardware can be viewed as a state machine. Changes state every clock cycle depending on the contents of the instruction register, condition codes, and external inputs. - Outputs of the state machine are control signals. Sequence of control signals generated by the machine is determined by wiring of logic elements, hence the name "hardwired control". - Speed of operation is one of the advantages of hardwired control is its speed of operation. - Disadvantages include: Little flexibility. Limited complexity of the instruction set it can implement. 3 Marks
7(a)	With a suitable diagram, explain about Three-bus organization of the data path. Provide control sequence for ADD R4, R5, R6. - In simple single-bus structure, the results in long control sequences, because only one data item can be transferred over the bus in a clock cycle. - Most commercial processors provide multiple internal paths to enable several transfers to take place in parallel. - Three-bus organization to connect the registers and the ALU of a processor. All general-purpose registers are combined into a single block called register file. - Register file has three ports. Two outputs ports connected to buses A and B, allowing the contents of two different registers to be accessed simultaneously, and placed on buses A and B. One input port allows the data on bus C to be loaded into a third register during the same clock cycle. - Inputs to the ALU and outputs from the ALU - Buses A and B are used to transfer the source operands to the A and B inputs of the ALU. Result is transferred to the destination over bus C. 3 Marks



4 Marks

Example: Add R4, R5, R6

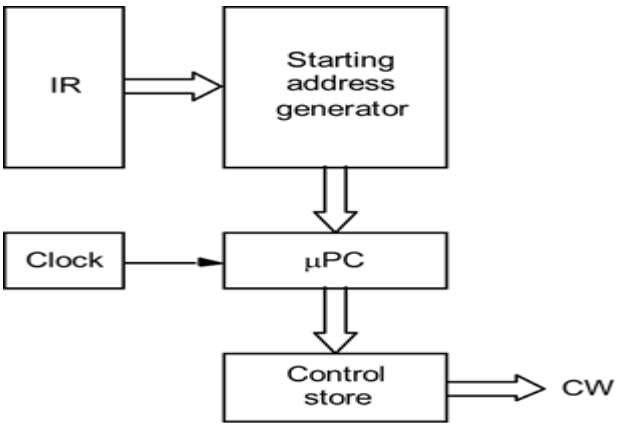
Step	Action
1	PC_{out}, R=B, MAR_{in}, Read, IncPC
2	WMFC
3	MDR_{outB}, R=B, IR_{in}
4	$R4_{outA}$, $R5_{outB}$, SelectA, Add, $R6_{in}$, End

- Control signals for such an operation are *R=A* or *R=B*. Three bus arrangement obviates the need for Registers *Y* and *Z* in the single bus organization.
- Incremental unit - Used to increment the PC by 4. Source for the constant 4 at the ALU multiplexer can be used to increment other addresses such as the memory addresses in multiple load/store instructions.

	1. Pass the contents of the *PC* through ALU and load it into *MAR*. Increment *PC*. 2. Wait for *MFC*. 3. Load the data received into *MDR* and transfer to *IR through ALU.* 4. Execution of the instruction is the last step. 3 Marks
8(a)	Explain about the microprogrammed control unit with conditional branching with the help of block diagram. Write the microinstructions for ADD (R3), R1. - Control signals are generated by a ***program*** similar to machine language programs. - Control Signals: PC_{out}, PC_{in}, MAR_{in}, Read, MDR_{out}, IR_{in}, Y_{in}, SelectY, Select4, Z_{in}, etc. - At every step, a Control Word needs to be generated. Every instruction will need a sequence of CWs for its execution. - Sequence of CWs for an instruction is the microroutine for the instruction. Each CW in this microroutine is referred to as a microinstruction. (SelectY is represented by Select=0, & Select4 by Select=1) - Every instruction will have its own microroutine which is made up of microinstructions. Microroutines for all instructions in the instruction set of a computer are stored in a special memory called Control Store. - Sequentially reading the CWs of the corresponding microroutine from the control store. 3 Marks

Micro-instruction	..	PC_{in}	PC_{out}	MAR_{in}	Read	MDR_{out}	IR_{in}	Y_{in}	Select	Add	Z_{in}	Z_{out}	$R1_{out}$	$R1_{in}$	$R3_{out}$	WMFC	End	:
1		0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	
2		1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	
3		0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	
4		0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	
5		0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	
6		0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	
7		0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	

2 Marks



2 Marks

	- Microprogram counter (mPC) is used to read CWs from control store sequentially. When a new instruction is loaded into IR, Starting address generator generates the starting address of the microroutine. - This address is loaded into the mPC. It is automatically incremented by the clock, so successive microinstructions are read from the control store. - Basic organization of the microprogrammed control unit cannot check the status of condition codes or external inputs to determine what should be the next microinstruction. - Use conditional branch microinstructions. These microinstructions, in addition to the branch address also specify which of the external inputs, condition codes or possibly registers should be checked as a condition for branching. - Starting and branch address generator accepts as inputs. Contents of the Instruction Register IR (as before). External inputs, Condition codes are used. - Generates a new address and loads it into microprogram counter (*mPC*) when a microinstruction instructs it do so. ***mPC*** is incremented every time a microinstruction is fetched except: - New instruction is loaded into IR, *mPC* is loaded with the starting address of the microroutine for that instruction. - Branch instruction is encountered and branch condition is satisfied, *mPC* is loaded with the branch address. - End instruction is encountered, *mPC* is loaded with the address of the first CW in the microroutine for the instruction fetch cycle. 3 Marks