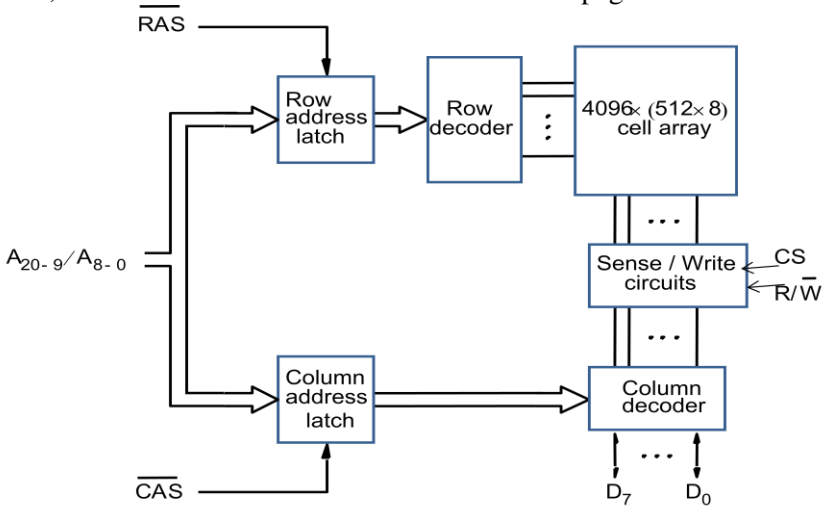
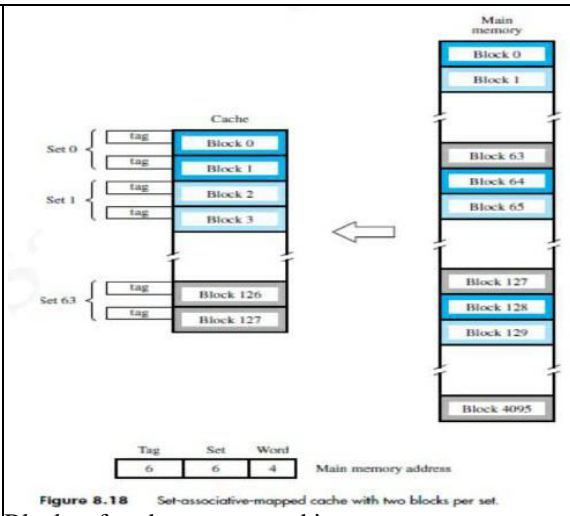
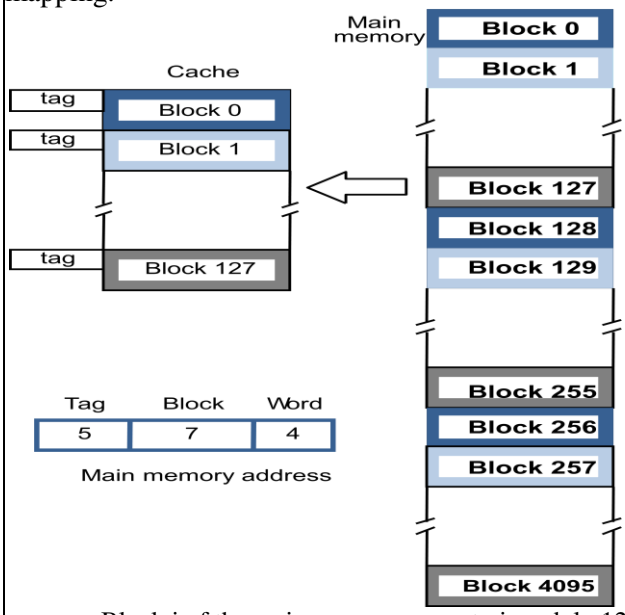
USN | | | | | | | | | | |

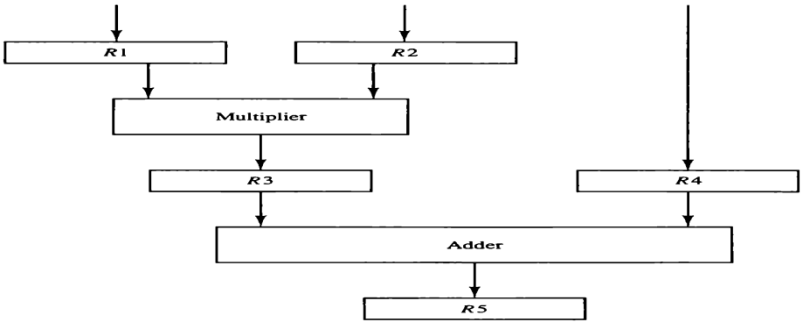
Internal Assessment Test 3–February 2023

Scheme & Solution

Sub:	Computer Organization and Architecture					SubCode:	21CS34	Branch:	ISE	
Date:	09/02/2023	Duration:	90min's	MaxMarks:	50	Sem/Sec:	III A, B &C			OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	Explain the internal organization of a 16 Megabits DRAM chip configured as 2MX8 cells. Also, describe how it can be made to work in fast page mode.  ■ Each row can store 512 bytes. 12 bits to select a row, and 9 bits to select a group in a row. Total of 21 bits. • 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 ■ Suppose if we want to access the consecutive bytes in the selected row. ■ This can be done without having to reselect the row. ▪ 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.	10(Diagram-3+ Explanation-7)	CO3	L2
2a	Explain Set Associative Memory in detail.		CO3	L2

	 Figure 8.18 Set-associative-mapped cache with two blocks per set. 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 block 0, and they can occupy either of the two positions. Memory address is divided into three fields: - 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. Set-associative mapping combination of direct and associative mapping. 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.	5(diagram-2,explanation-3)		
2b	Give detailed explanation of direct mapping and how is it different from associative mapping.  • Block j of the main memory maps to j modulo 128 of the cache. 0 maps to 0, 129 maps to 1. • More than one memory block is mapped onto the same position in the cache. • May lead to contention for 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:	5(Diagram-2,explanation-2,1-differences)	CO3	L2

	- Low order 4 bits determine one of the 16 words in a block. - When a new block is brought into the cache, the the next 7 bits determine which cache block this new block is placed in. - High order 5 bits determine which of the possible 32 blocks is currently present in the cache. These are tag bits. • Simple to implement but not very flexible. • 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	Explain pipelining with the necessary diagram and example. • Pipelining is a technique of decomposing a sequential process into suboperations, with each subprocess being executed in a special dedicated segment that operates concurrently with all other segment. • A pipeline can be visualized as a collection of processing segments through which binary information flows. • Each segment performs partial processing dictated by the way the task is partitioned. • The result obtained from the computation in each segment is transferred to the next segment in the pipeline. • The final result is obtained after the data have passed through all segments. • It is characteristic of pipelines that several computations can be in progress in distinct segments at the same time. • The overlapping of computation is made possible by associating a register with each segment in the pipeline. • The registers provide isolation between each segment so that each can operate on distinct data simultaneously. • Each segment consists of an input register followed by a combinational circuit. • The register holds the data and the combinational circuit performs the suboperation in the particular segment. • The output of the combinational circuit in a given segment is applied to the input register of the next segment. • A clock is applied to all registers after enough time has elapsed to perform all segment activity. • In this way the information flows through the pipeline one step at a time. • Ai* Bi + Ci for i = 1, 2, 3, . . . , 7  • Each suboperation is to be implemented in a segment within a pipeline. • Each segment has one or two registers and a combinational circuit. • R 1 through RS are registers that receive new data with every clock pulse. • The multiplier and adder are combinational circuits. • The five registers are loaded with new data every clock pulse. $R1 \leftarrow A_i,\quad R2 \leftarrow B_i$ Input A_i and B_i $R3 \leftarrow R1 * R2,\quad R4 \leftarrow C_i$ Multiply and input C_i $R5 \leftarrow R3 + R4$ Add C_i to product	10(Diagram-3,Explanation-5,eg.-2)	CO5	L2

	• The first clock pulse transfers A1 and B1 into R 1 andR2. • The second dock pulse transfers the product of R 1 and R2 into R3 and C1 into R4. • The same clock pulse transfers A2 and B2 into R 1 and R2. The third clock pulse operates on all three segments simultaneously. **TABLE 9-1** Content of Registers in Pipeline Example • • It places A1 and B1 into R1 and R2, transfers the product of R1 and R2 into R3, transfers C, into R4, and places the sum of R3 and R4 into RS. • It takes three clock pulses to fill up the pipe and retrieve the first output from R5. • From there on, each dock produces a new output and moves the data one step down the pipeline. • This happens as long as new input data flow into the system. When no more input data are available, the clock must continue until the last output emerges out of the pipeline.			
4	Calculate the Speedup ratio of non pipeline to pipeline, if t_p=20 ns, no of tasks, n =100, and no of segments, k=4. • the time it takes to process a sub-operation in each segment be equal to tp = 20 ns. • K=4, n=100 tasks, • Pipeline= (k + n - 1)*tp=(4+99)*20=2060 ns • If tn=ktp=4*20=80 ns • Non pipeline=n*ktp=100*80=8000 ns. • Speedup ratio=8000/2060=3.88	10	CO5	L3
5	In detail describe Arithmetic pipeline. • Pipeline arithmetic units are usually found in very high speed computers. • They are used to implement floating-point operations, multiplication of fixed-point numbers, and similar computations encountered in scientific problems. • Example:-The inputs to the floating-point adder pipeline are two normalized floating-point binary numbers. • X = A X 2^a • Y = B X 2^b A and B are two fractions that represent the mantissas and a and b are the exponents. • The floating-point addition and subtraction can be performed in four segments, as shown in Fig. 9-6. • The registers labeled R are placed between the segments to store intermediate results. The sub operations that are performed in the four segments are: • 1. Compare the exponents. • 2. Align the mantissas. • 3. Add or subtract the mantissas. • 4. Normalize the result.	10(example-3,diagram-4,explanation-3)	CO5	L2

Clock Pulse Number	Segment 1		Segment 2		Segment 3
	R1	R2	R3	R4	R5
1	A_1	B_1	—	—	—
2	A_2	B_2	$A_1 * B_1$	C_1	—
3	A_3	B_3	$A_2 * B_2$	C_2	$A_1 * B_1 + C_1$
4	A_4	B_4	$A_3 * B_3$	C_3	$A_2 * B_2 + C_2$
5	A_5	B_5	$A_4 * B_4$	C_4	$A_3 * B_3 + C_3$
6	A_6	B_6	$A_5 * B_5$	C_5	$A_4 * B_4 + C_4$
7	A_7	B_7	$A_6 * B_6$	C_6	$A_5 * B_5 + C_5$
8	—	—	$A_7 * B_7$	C_7	$A_6 * B_6 + C_6$
9	—	—	—	—	$A_7 * B_7 + C_7$

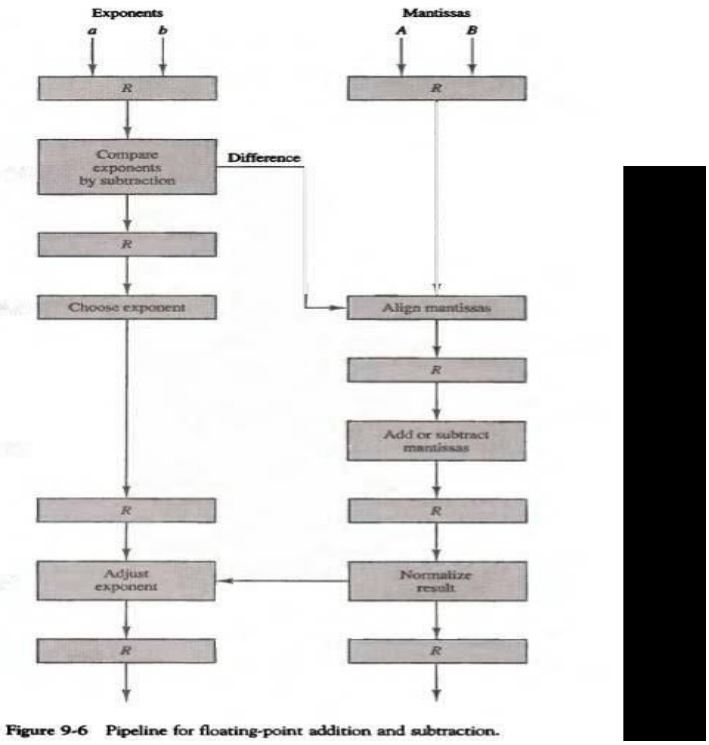


Figure 9-6 Pipeline for floating-point addition and subtraction.

- Consider the two normalized floating-point numbers:

$$X = 0.9504 \times 10^3$$

$$Y = 0.8200 \times 10^2$$

- The two exponents are subtracted in the first segment to obtain 3 - 2 = 1. The larger exponent 3 is chosen as the exponent of the result. The next segment shifts the mantissa of Y to the right to obtain

$$X = 0.9504 \times 10^3$$

$$Y = 0.0820 \times 10^3$$

- This aligns the two mantissas under the same exponent. The addition of the two mantissas in segment 3 produces the sum

$$Z = 1.0324 \times 10^3$$

- The sum is adjusted by normalizing the result so that it has a fraction with a nonzero first digit. This is done by shifting the mantissa once to the right and incrementing the exponent by one to obtain the normalized sum.

$$Z = 0.10324 \times 10^4$$

6	Illustrate the concept of Virtual memory. With a neat diagram, explain the translation of a virtual address to a physical address. ■ Each virtual or logical address generated by a processor is interpreted as a virtual page number (high-order bits) plus an offset (low-order bits) that specifies the location of a particular byte within that page. ■ Information about the main memory location of each page is kept in the page table. ■ Main memory address where the page is stored. ■ Current status of the page. ■ Area of the main memory that can hold a page is called as page frame. Starting address of the page table is kept in a page table base register. ■ Virtual page number generated by the processor is added to the contents of the page table base register. ■ This provides the address of the corresponding entry in the page table. ■ The contents of this location in the page table give the starting address of the page if the page is currently in the main memory.	10(Diagram-4,explanation-6)	CO3	L2

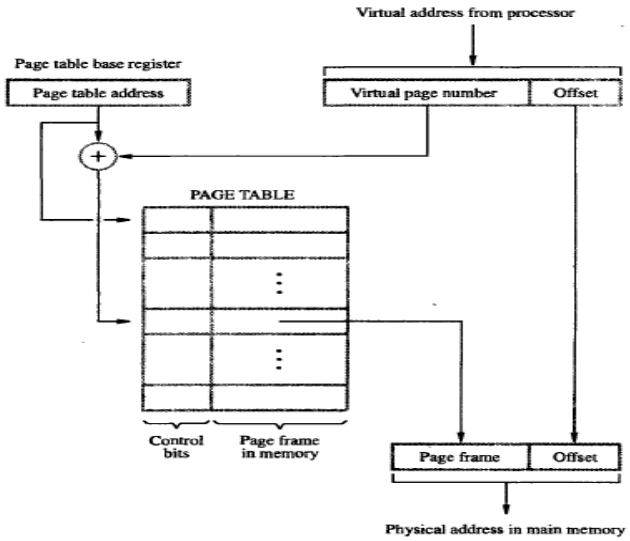


Figure 5.27 Virtual-memory address translation.

- Page table entry for a page also includes some control bits which describe the status of the page while it is in the main memory.
- One bit indicates the validity of the page.
  - Indicates whether the page is actually loaded into the main memory.
  - Allows the operating system to invalidate the page without actually removing it.
- One bit indicates whether the page has been modified during its residency in the main memory.
  - This bit determines whether the page should be written back to the disk when it is removed from the main memory.
  - Similar to the dirty or modified bit in case of cache memory.
- Other control bits for various other types of restrictions that may be imposed.
  - For example, a program may only have read permission for a page, but not write or modify permissions.

Faculty Signature CCI Signature HOD Signature