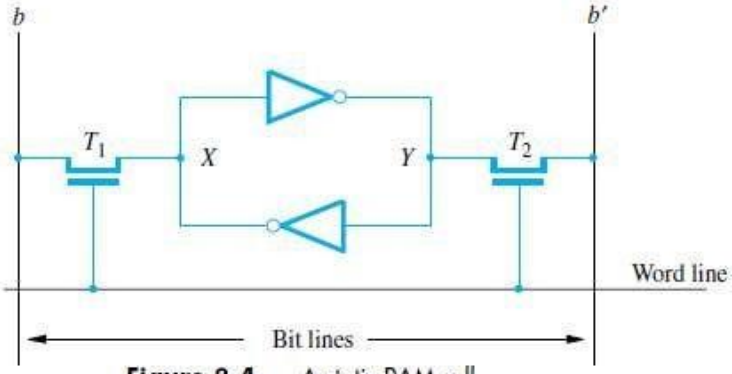
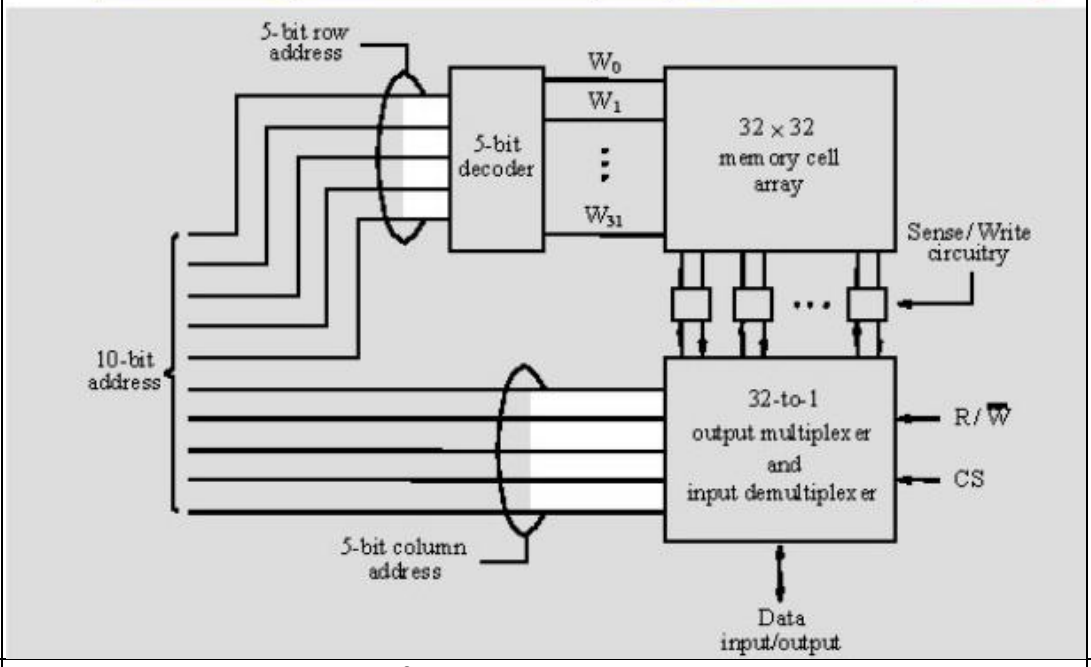
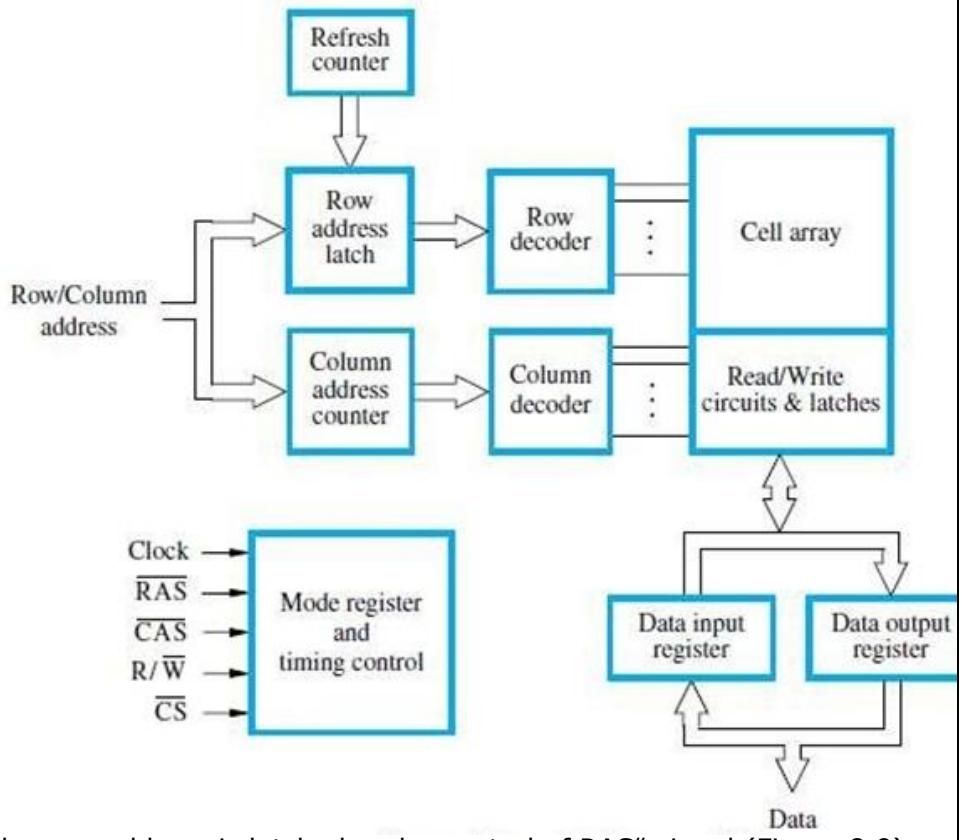
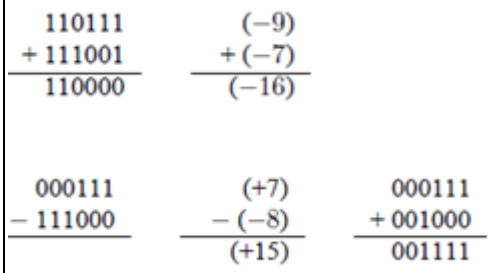
USN | | | | | | | | | | |

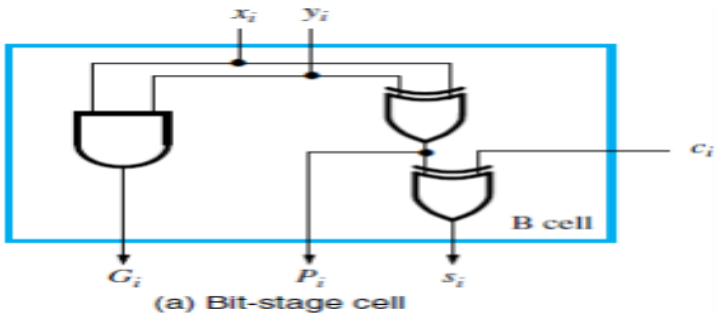
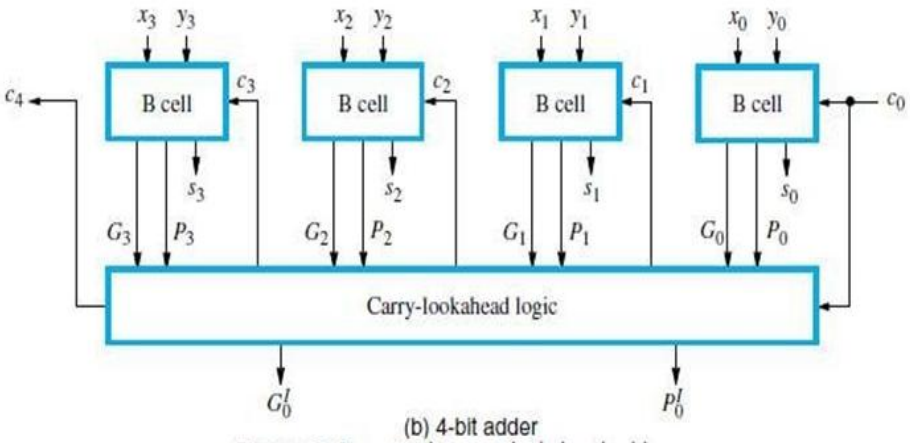
Internal Assessment Test 3 – March 2022

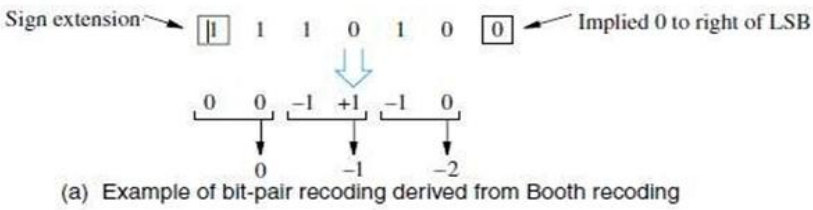
Sub:	Computer Organization					Sub Code:	18CS34	Branch:	CSE	
Date:	04/03/22	Duration:	90 minutes	Max Marks:	50	Sem/Sec:	III /A,B,C		OBE	

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	Write short notes on the working of a static RAM cell and Explain the internal organization of 1K x 1 Memory chip. • Memories consist of circuits capable of retaining their state as long as power is applied are known.  **Figure 8.4** A static RAM cell. • Two inverters are cross connected to form a latch (Figure 8.4). • The latch is connected to 2-bit-lines by transistors T_1 and T_2. • The transistors act as switches that can be opened/closed under the control of the word-line. • When the word-line is at ground level, the transistors are turned off and the latch retain its state. ***Read Operation*** • To read the state of the cell, the word-line is activated to close switches T_1 and T_2. • If the cell is in state 1, the signal on bit-line b is high and the signal on the bit-line b" is low. • Thus, b and b" are complement of each other. • Sense/Write circuit → monitors the state of b & b" and → sets the output accordingly. ***Write Operation*** • The state of the cell is set by → placing the appropriate value on bit-line b and its complement on b" and → then activating the word-line. This forces the cell into the corresponding state. • The required signal on the bit-lines is generated by Sense/Write circuit. Large memory circuit that has 1k that is 1024 memory cells. The circuit can be organized as a 128 into 8 memory, requiring a total of 19 external connections. Alternatively, the same number of cells can be organized into a 1K X 1 format. In this case, a 10 bit address is needed but there is only one data line resulting in a total of 15 external connections. Such an organization requires 10-bit address being divided into two groups of five bits each to form the row address and the column address of the cell array. A row address selects a row of 32 cells all of which are accessed in parallel. However, according to the column address only one of these cells is connected to the external data line by the output multiplexer and the input demultiplexer.	[10]	3	L2

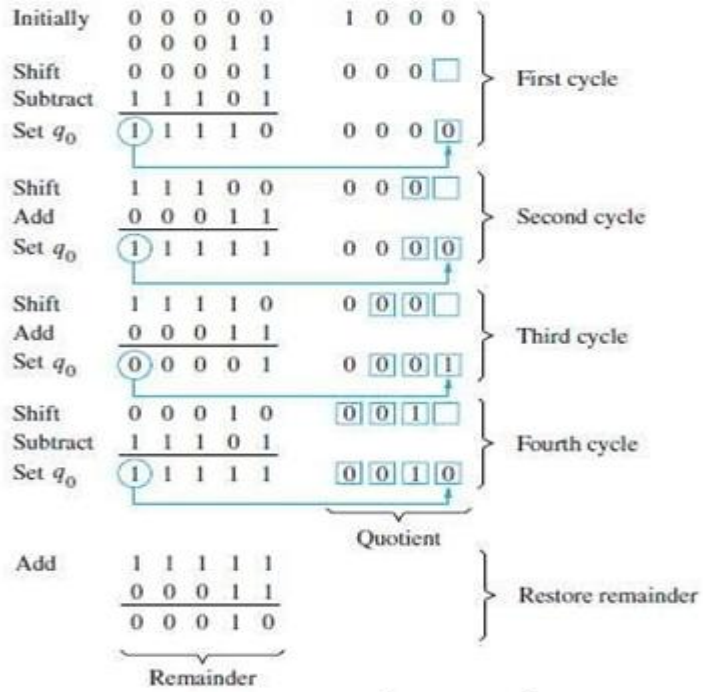
	Internal organization of memory chips size=1k bits (1024x1)  			
2	Explain the internal structure of Synchronous DRAM with neat diagrams. • The operations are directly synchronized with clock signal as in Figure. • The address and data connections are buffered by means of registers. • The output of each sense amplifier is connected to a latch. • A Read-operation causes the contents of all cells in the selected row to be loaded in these latches. • Data held in latches that correspond to selected columns are transferred into data-output register. • Thus, data becoming available on the data-output pins.  • First, the row-address is latched under control of RAS“ signal (Figure 8.9). • The memory typically takes 2 or 3 clock cycles to activate the selected row. • Then, the column-address is latched under the control of CAS“ signal. • After a delay of one clock cycle, the first set of data bits is placed on the data-lines.	[10]	3	L2

	• SDRAM automatically increments column-address to access next 3 sets of bits in the selected row. • A good indication of performance is given by 2 parameters: 1) Latency 2) Bandwidth. **Latency** • It refers to the amount of time it takes to transfer a word of data to or from the memory. • For a transfer of single word, the latency provides the complete indication of memory performance. • For a block transfer, the latency denotes the time it takes to transfer the first word of data. **Bandwidth** • It is defined as the number of bits or bytes that can be transferred in one second. • Bandwidth mainly depends on 1) The speed of access to the stored data & 2) The number of bits that can be accessed in parallel.			
3 (a)	Perform the following operations on the 5 bit signed numbers using 2's compliment representation system. Also indicate whether overflow has occurred. (i) (-9) + (-7) overflow (ii) (+7) – (-8) (iii) (+13) + (+14) (overflow)  01101 +13 01110 +14 11011	[5]	4	L3
3 (b)	Explain 4bit carry-look ahead adder (5)	[5]	4	L3

	• The logic expression for s_i(sum) and c_{i+1}(carry-out) of stage i are $s_i=x_i+y_i+c_i$ ------(1) $c_{i+1}=x_iy_i+x_ic_i+y_ic_i$.....(2) • Factoring (2) into $c_{i+1}=x_iy_i+(x_i+y_i)c_i$ we can write $c_{i+1}=G_i+P_iC_i$ where $G_i=x_iy_i$ and $P_i=x_i+y_i$ • The expressions G_i and P_i are called generate and propagate functions (Figure 9.4). • If $G_i=1$, then $c_{i+1}=1$, independent of the input carry c_i. This occurs when both x_i and y_i are 1. Propagate function means that an input-carry will produce an output-carry when either $x_i=1$ or $y_i=1$. • All G_i and P_i functions can be formed independently and in parallel in one logic-gate delay. • Expanding c_i terms of i-1 subscripted variables and substituting into the c_{i+1} expression, we obtain $c_{i+1}=G_i+P_iG_{i-1}+P_iP_{i-1}G_{i-2}.....+P_1G_0+P_iP_{i-1}.....P_0c_0$ • Conclusion: Delay through the adder is 3 gate delays for all carry-bits & 4 gate delays for all sum-bits. • Consider the design of a 4-bit adder. The carries can be implemented as $c_1=G_0+P_0c_0$ $c_2=G_1+P_1G_0+P_1P_0C_0$ $c_3=G_2+P_2G_1+P_2P_1G_0+P_2P_1P_0C_0$ $c_4=G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1G_0+P_3P_2P_1P_0C_0$ • The carries are implemented in the block labeled carry-lookahead logic. An adder implemented in this form is called a **Carry-Lookahead Adder**. • Limitation: If we try to extend the carry-lookahead adder for longer operands, we run into a problem of gate fan-in constraints.  (a) Bit-stage cell  (b) 4-bit adder **Figure 9.4** A 4-bit carry-lookahead adder.			
4 (a)	Perform signed multiplication of numbers (-12) and (-11) using Booth's Algorithm.	[5]	4	L3

	10100(-12) X10101(-11) → 1 0 1 0 0 -1 1 -1 1 -1 (recoded multiplier) 0 0 0 0 0 1 1 0 0 1 1 1 1 0 1 0 0 0 0 0 1 1 0 0 1 1 0 1 0 0 0 1 1 0 0 0 1 0 0 0 0 1 0 0 (+132)			
4 (b)	Perform multiplication of (+13) and (-6) using Bit Pair recoding technique.  (a) Example of bit-pair recoding derived from Booth recoding 0 1 1 0 1 (+13) × 1 1 0 1 0 (−6) 0 1 1 0 1 0 −1 −2 1 1 1 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 0 0 1 0	[5]	4	L3

5 (a)	Given A = 10101 and B = 00100. Perform A/B using Restoring Division method. • Procedure: Do the following n times 1) Shift A and Q left one binary position (Figure 9.22). 2) Subtract M from A, and place the answer back in A 3) If the sign of A is 1, set q_0 to 0 and add M back to A(restore A).If the sign of A is 0, set q_0 to 1 and no restoring done. **Solution:**  	[5]	4	L3

5 (b)	Perform 8/3 using Non Restoring Division method. **NON-RESTORING DIVISION** • Procedure: Step 1: Do the following n times i) If the sign of A is 0, shift A and Q left one bit position and subtract M from A; otherwise, shift A and Q left and add M to A (Figure 9.23). ii) Now, if the sign of A is 0, set q_0 to 1; otherwise set q_0 to 0. Step 2: If the sign of A is 1, add M to A (restore).  **Figure 9.25** A non-restoring division example.	[5]	4	L3
6	Discuss with neat diagram, single bus organization of data path inside a processor. • ALU and all the registers are interconnected via a **Single Common Bus** (Figure 7.1). • Data & address lines of the external memory-bus is connected to the internal processor-bus via MDR& MAR respectively. (MDR→ Memory Data Register, MAR → Memory Address Register). • **MDR** has 2 inputs and 2 outputs. Data may be loaded → into MDR either from memory-bus (external) or → from processor-bus (internal). • **MAR**"s input is connected to internal-bus; • MAR"s output is connected to external-bus. • **Instruction Decoder & Control Unit** is responsible for → issuing the control-signals to all the units inside the processor. → implementing the actions specified by the instruction (loaded in the IR). • Register R0 through R(n-1) are the **Processor Registers**. The programmer can access these registers for general-purpose use. • Only processor can access 3 registers **Y**, **Z** & **Temp** for temporary storage during program-execution.The programmer cannot access these 3 registers. • In **ALU**, 1) „A" input gets the operand from the output of the multiplexer (MUX). 2) „B" input gets the operand directly from the processor-bus. • There are 2 options provided for „A" input of the ALU. • MUX is used to select one of the 2 inputs. • **MUX** selects either → output of Y or	[10]	5	

→ constant-value 4(which is used to increment PC content).

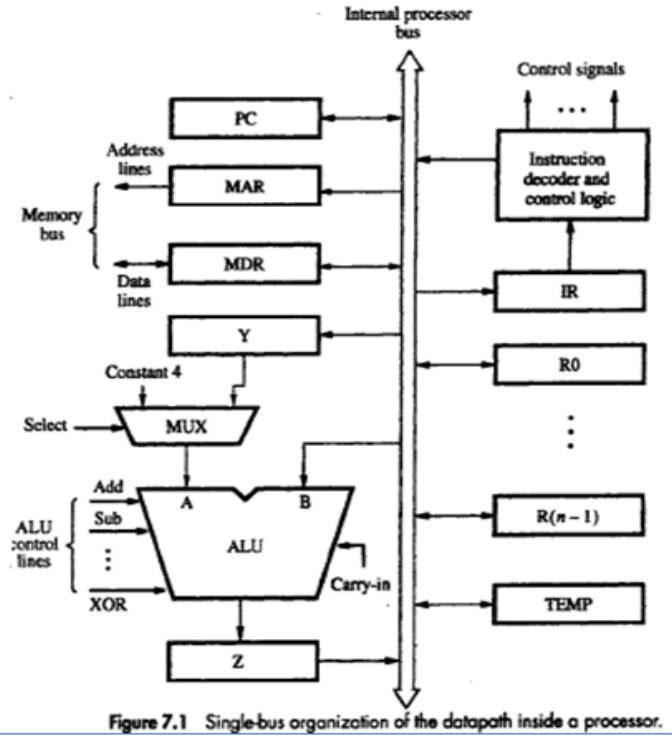


Figure 7.1 Single-bus organization of the datapath inside a processor.

- An instruction is executed by performing one or more of the following operations:
  1) Transfer a word of data from one register to another or to the ALU.
  2) Perform arithmetic or a logic operation and store the result in a register.
  3) Fetch the contents of a given memory-location and load them into a register.
  4) Store a word of data from a register into a given memory-location.
- **Disadvantage:** Only one data-word can be transferred over the bus in a clock cycle.

  Solution: Provide multiple internal-paths. Multiple paths allow several data-transfers to take place inparallel.

6

(ii) Explain the control sequence for the execution of 1 complete instruction Add (R3), R1.

- 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.
  3) Perform the addition &
  4) Load the result into R1.

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

Figure 7.6 Control sequence for execution of the instruction Add (R3),R1

- Instruction execution proceeds as follows:

  Step1--> The instruction-fetch operation is initiated by

  → loading contents of PC into MAR &

→ sending a Read request to memory.

The Select signal is set to Select4, which causes the Mux to select constant 4. This valueis added to operand at input B (PC"s content), and the result is stored in Z.

Step2--> Updated value in Z is moved to PC. This completes the PC increment operation andPC will now point to next instruction.

Step3--> Fetched instruction is moved into MDR and then to IR.The step 1 through 3 constitutes the **Fetch Phase**.

At the beginning of step 4, the instruction decoder interprets the contents of the IR. Thisenables the control circuitry to activate the control-signals for steps 4 through 7.

The step 4 through 7 constitutes the **Execution Phase**.

Step4--> Contents of R3 are loaded into MAR & a memory read signal is issued.Step5--> Contents of R1 are transferred to Y to prepare for addition.

Step6--> When Read operation is completed, memory-operand is available in MDR, and theaddition is performed.

Step7--> Sum is stored in Z, then transferred to R1.The End signal causes a new instructionfetch cycle to begin by returning to step1.