CMR Institute of Technology

Department of Electronics and Communication Engineering

ACADEMIC YEAR 2025-26

SEMESTER-III

SUBJECT NAME: COMPUTER ORGANIZATION AND ARCHITECTURE

SUBJECT CODE: BEC306C

Internal Assessment Test – 2									
Sub:	Computer Organization and Architecture							**Code:**	BEC306C
Date:	07/01/2026	**Duration:**	90 mins	**Max Marks:**	50	**Sem:**	III	**Branch:**	ECE

Answer any Five Questions

		Marks	CO	RBT
1.	a) Distinguish between memory-mapped I/O and standard I/O. b) Develop a program segment to read a line of text from a keyboard and display it.	[03] [07]	CO3	L3
2.	Define interrupt. Point out and explain the various ways of enabling and disabling interrupts.	[10]	CO3	L2
3.	With a neat diagram, explain the internal organization of 1k X 1 memory chip.	[10]	CO4	L3
4.	Describe three-bus organization of the Datapath with a neat diagram.	[10]	CO5	L2
5.	Discuss Hardwired Control Unit Organization with relevant diagram.	[10]	CO5	L2
6	Develop and construct the complete control sequence for the execution of instruction ADD (R3), R1	[10]	CO5	L3
7	Draw and organize the internal structure of a 2M X 8 asynchronous DRAM chip	[10]	CO4	L3

IAT-2 SOLUTION

1. a) Distinguish between memory-mapped I/O and I/O mapped I/O. [03] CO3 L3

   b) Develop a program segment to read a line of text from a keyboard and display it. [07]

Sol: a)

1) In Memory-mapped I/O approach, each I/O device is treated similar to any other memory Location with a common address space for Memory and I/O devices Where as in I/O mapped I/O approach, the I/O devices are given a separate address space different from the address space available for Memory
2) In Memory mapped I/O approach, there is no additional control information required to separate memory address information from I/O device address
3) In I/O mapped I/O approach, there is a need to specify additional control information specifying whether the address available on Address bus is a Memory-address or whether the address is pointing to any I/O device interfaced to the processor. This control Information is mostly available on Memory/$\overline{\text{Input} - \text{output}}$ (M/$\overline{IO}$) , a control signal provided by the Processor to all external devices.
4) Interfacing is easy with Memory-Mapped IO approach and is the general adopted approach.
5) Using IO mapped IO approach requires additional control lines which makes the interfacing more complex in comparison
6) In IO mapped IO approach the memory location as well as the IO devices can have the same address
7) In memory-mapped IO approach, none of the memory locations or the IO devices can have the same address.

b)

```
        Move      #LINE,R0        Initialize memory pointer.
WAITK   TestBit   #0,STATUS       Test SIN.
        Branch=0  WAITK           Wait for character to be entered.
        Move      DATAIN,R1       Read character.
WAITD   TestBit   #1,STATUS       Test SOUT.
        Branch=0  WAITD           Wait for display to become ready.
        Move      R1,DATAOUT      Send character to display.
        Move      R1,(R0)+        Store charater and advance pointer.
        Compare   #$0D,R1         Check if Carriage Return.
        Branch≠0  WAITK           If not, get another character.
        Move      #$0A,DATAOUT    Otherwise, send Line Feed.
        Call      PROCESS         Call a subroutine to process the
                                    the input line.
```

2. Define interrupt. Point out and explain the various ways of enabling and disabling interrupts. [10] CO3 L2

Sol: **Interrupts** are signals generated by hardware devices or software programs to temporarily halt the normal execution of the CPU and divert it to a special routine called an **Interrupt Service Routine (ISR)** to handle an event that requires immediate attention.

After servicing the interrupt, the CPU resumes execution of the previously interrupted program from where it left off.

Enabling and disabling the interrupts

The enabling and disabling interrupts must be available to the programmer so that itcan determine whether to continue a task or execute the interrupting device's ISR routine depending on priority of the tasks at hand. A simple way is to provide machine instructions such as Interrupt-enable and Interrupt-disable that performs these functions.

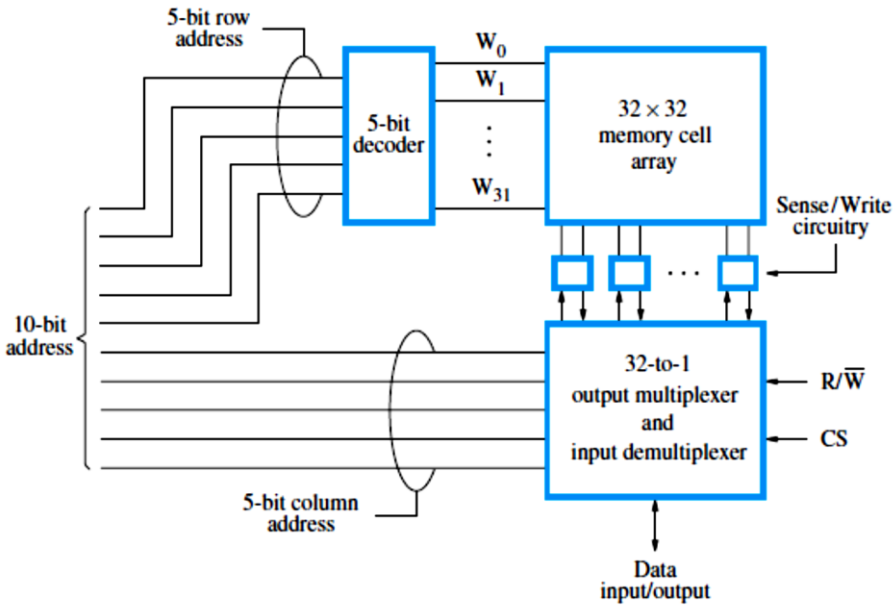
When the device activates interrupt request signal, it keeps this signal activated until it learns that the processor has accepted its request. It is essential to ensure that this active signal does not lead to successive interruptions causing the system to enter an infinite loop from which it cannot recover. There are three ways to avoid this problem.

The first option is that processor must ignore the interrupt request line until the execution of first instruction of interrupt service routine (ISR) has been completed. Then by using the Interrupt disable instruction as the first instruction in the ISR, programmer can ensure that no further interruption will occur.

The second option is to have processor automatically disable the interrupts before staring the execution of Interrupt service routine. After saving the contents of PS on the stack with Interrupt enable bit equal to 1, the processor clears the Interrupt enable bit in PS register thus disabling further interrupts. When a return from interrupt instruction is executed, the contents of PS are restored from the stack setting the Interrupt enable bit back to 1. Hence interrupts are again enabled.

In the third option, processor has a special interrupt request line for which the interrupt handling circuit responds only to leading edge of the signal. Such a line is said to edge triggered.

3. With a neat diagram, explain the internal organization of 1k X 1 memory chip. [10] CO4 L3



1) 1k*1 = 1024 *1 memory
2) Only One data Line is Available
3) 1024 memory cells (each cell storing 1-bit information) is arranged in 32 * 32 format i.e. a matrix of 32 row by 32 columns
4) Since only one data line is available, each bit (cell) of memory will have a unique address.
5) Hence, 1024 cell memory requires 1024 unique addresses to address each cell individually.
6) Hence, we need 10 address lines to generate 1024 addresses (2^{10} =1024)
7) The construction of our required 1k* 1 memory is facilitated using the above diagram
8) 10 address lines are segregated in to two groups of 5-address lines each so that one set of 5-address lines can select one of the 32 rows (2^5 =32) while other group of 5-address lines can be used to select one of the 32 columns of the 32 * 32 matrix format of memory.
9) Hence when any address is available on 10-bit address lines, only one row and one column will get activated using the 5-bit decoder for row identifier and the 5-bit demultiplexer shown for column activation which is connected to a 32:1 multiplexer used for connecting the single data line to the active column
10) Hence the combination of the Active column and active row selects only one memory cell which becomes active facilitate either the read operation of write operation as confirmed by the signal level on the $\mathbf{R/\overline{W}}$ control input line.
11) The sense/Write circuitry connected on each column shown in figure above facilitates the reading or writing of the memory cells activated by the 10-bit address line information.

4. Describe three-bus organization of the Datapath with a neat diagram. [10] CO5 L2

Sol: **MULTIPLE BUS ORGANIZATION**

In single bus structure only one data can be transferred over the bus in one clock cycle. Hence some processors have multiple internal paths.

- The fig below shows three bus structure used to connect registers and ALU.
- All general purpose registers are combined into a single block called **register file**.
- The register file has 3 ports. The two output ports allow the contents of two different registers to be placed on bus A and B. The input port allows the contents of bus C to be placed into a third register during the same clock cycle.
- Buses A and B are normally used for passing input operands to ALU. Bus C passes the result to the destination. This arrangement avoids the need for registers Y and Z. If needed ALU may simple pass one of its two operands unmodified to bus C by using control signals R=A or R=B.
- The second feature in the figure below is Incrementer unit which is used to increment PC by 4. This eliminates the need to add 4 to PC everytime using the main ALU.
- Consider a three operand instruction

ADD R4,R5,R6

- The control sequence for executing this instruction is given in the figure 7.9 below.

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.

- The contents of PC are passed through the ALU using R=B control signal and loaded into the MAR to start the memory read operation. At the same time PC is incremented by 4. (MAR has original contents of PC. PC is incremented at the end of the clock cycle. This will not affect the contents of MAR).
- In step 2 the processor waits for MFC
- In step 3 the processor loads the data from MDR to IR.

- In step 4 all the remaining execution operations can be carried out thus reducing the number of steps.

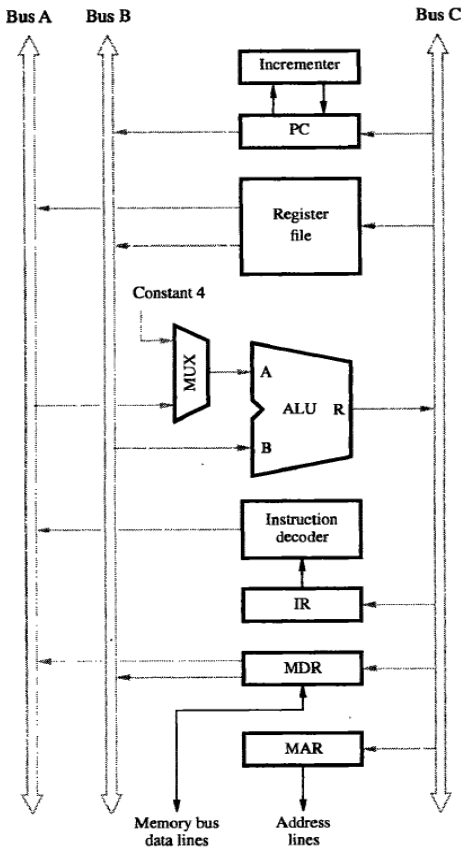


Figure 7.8 Three-bus organization of the datapath.

5. Discuss Hardwired Control Unit Organization with relevant diagram. [10] CO5 L2

Sol: **HARDWIRED CONTROL**

To execute instructions, the processor must have some means of generating the control signals needed in the proper sequence. This can be done by hardwired control or microprogrammed control. The figure below shows the Control Unit Organization.

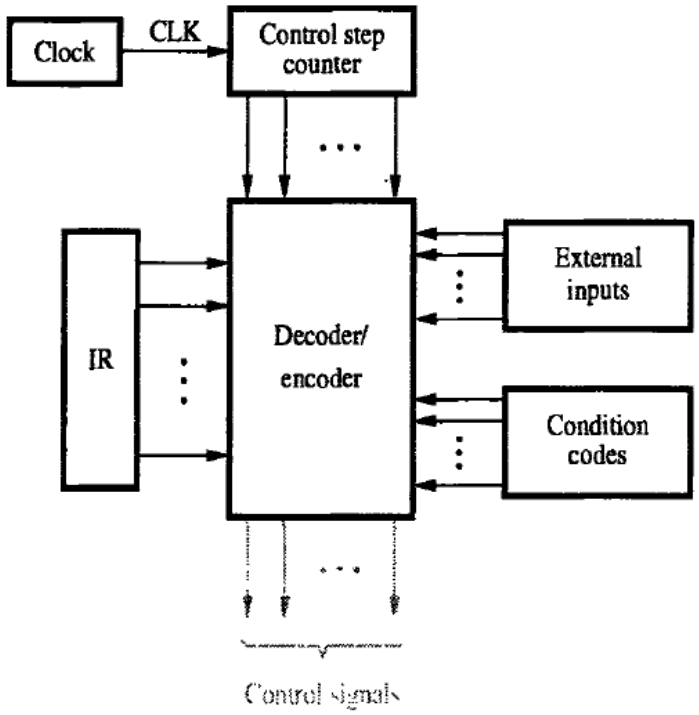


Figure 7.10 Control unit organization.

The decoder/encoder block in the above figure is a combinational circuit that generates the required control outputs depending on the state of all its inputs. A more detailed block diagram by separating decoder and encoder blocks is shown in the fig. below.

- The step decoder provides a separate signal line for each step/time slot.
- The output of Instruction Decoder consists of separate lines for each machine instruction.
- Depending on the instruction any one of INS_1 to INS_m is set to 1 and all other lines are set to 0.
- The input signals to the encoder block are combined to generate individual control signals Y_{in}, PC_{out}, Add, End and so on.

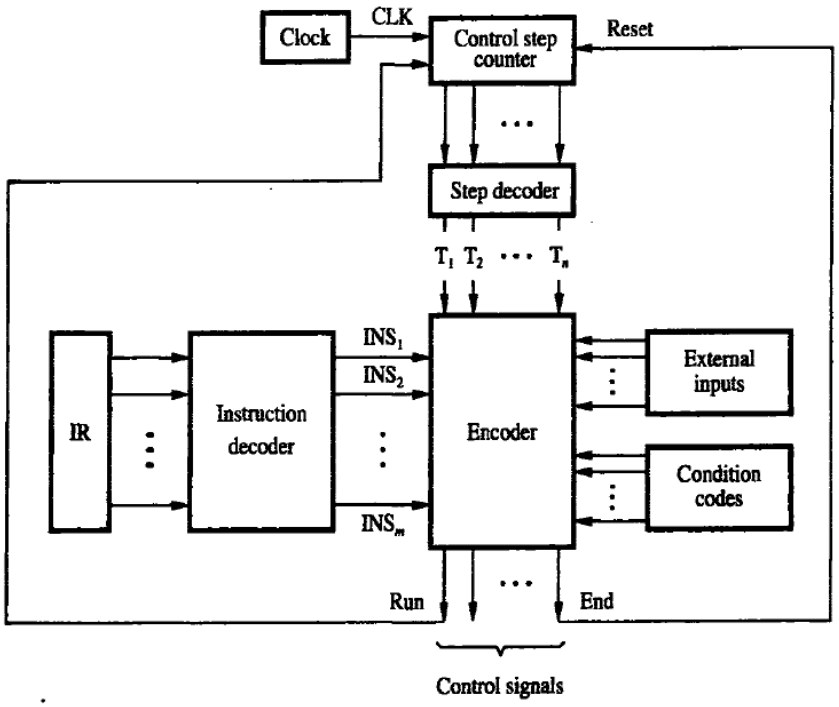


Figure 7.11 Separation of the decoding and encoding functions.

- **Example on how encoder generates Zin control signal**. This signal is asserted during time slot T1 for all instructions, during T4 for an unconditional branch instruction and so on. This is derived from the control sequences in fig. 7.6 and fig. 7.7

$$Z_{in} = T_1 + T_6 \cdot ADD + T_4 \cdot BR + \cdots \quad [7.1]$$

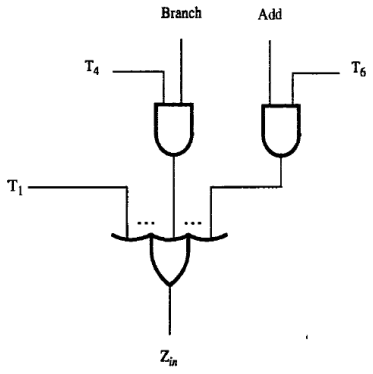


Figure 7.12 Generation of the Z_{in} control signal for the processor in Figure 7.1.

- Example on how to generate End control signal. This signal starts a new instruction fetch cycle by resetting the control step counter to its starting value.

$$End = T_7 \cdot ADD + T_5 \cdot BR + (T_5 \cdot N + T_4 \cdot \overline{N}) \cdot BRN + \cdots \quad [7.2]$$

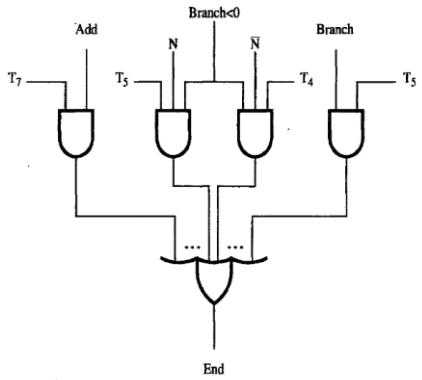


Figure 7.13 Generation of the End control signal.

- There is another control signal RUN which is incremented by one at the end of every clock cycle. When RUN is equal to 0, the counter stops counting. This is needed whenever the processor has to wait for WMFC signal.
- Figures 7.10 and 7.11 can be viewed as state machine that changes from one state to another in every clock cycle depending on the contents of IR, condition codes and external inputs. The sequence of operations carried out by this machine is determined by the wiring of the logic elements. Hence the name "hardwired".

6 Develop and construct the complete control sequence for the execution of instruction ADD (R3), R1 [10] CO5 L3

Sol: **EXECUTION OF COMPLETE INSTRUCTION**

Consider an instruction

Add (R3), R1

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

1. Fetch the instruction
2. Fetch the first operand (contents of memory location pointed to by R3)
3. Perform the addition
4. Load the result into R1

Figure below gives the sequence of control steps required to perform these operations for a single bus architecture.

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.

1. In step 1, the instruction fetch operation is initiated by loading the contents of PC into MAR and sending the Read request to the memory. The Select signal is set to Select4 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 the PC and the result is stored in Z.
2. In step 2, the updated value is moved from register Z back into PC. To speed up the execution of Branch instruction, the value of PC is copied into register Y. But this is not necessary while executing Add instruction.
3. In step 3, the word fetched from memory is loaded into the IR.
4. At the beginning of step 4, the instruction decoding circuit interprets the contents of the IR. This enables the control circuitry to activate the control signals for steps 4 through 7. The contents of register R3 are transferred to the MAR and memory read operation is initiated.
5. In step 5, the contents of R1 are transferred to register Y.
6. In step 6 addition operation is performed. The contents of MDR are gated to the bus and also to B input of ALU. The register Y is selected as the second input of ALU by choosing SelectY.
7. In step 7, the sum is stored in register Z and then transferred to R1. Then it returns to step 1.

7 Draw and organize the internal structure of a 2M X 8 asynchronous DRAM chip [10] CO4 L3

Sol:

ASYNCHRONOUS DRAM DESCRIPTION

- The 4 bit cells in each row are divided into 512 groups of 8 (Figure 5.7).
- 21 bit address is needed to access a byte in the memory. 21 bit is divided as follows:
  1) 12 address bits are needed to select a row.
     i.e. $A_{8\text{-}0}$ → specifies row-address of a byte.

2) 9 bits are needed to specify a group of 8 bits in the selected row.
i.e. A_{20-9} → specifies column-address of a byte.

- During Read/Write-operation,
  → row-address is applied first.
  → row-address is loaded into row-latch in response to a signal pulse on **RAS'** input of chip.(RAS = Row-address Strobe CAS = Column-address Strobe)

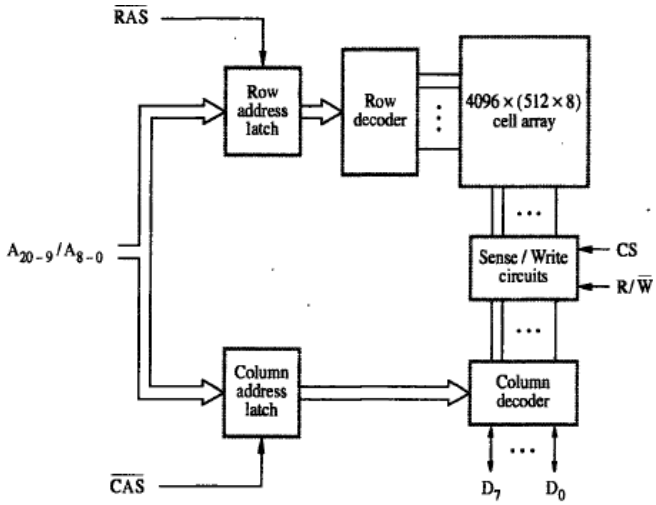


Figure 5.7 Internal organization of a 2M × 8 dynamic memory chip

- When a Read-operation is initiated, all cells on the selected row are read and refreshed.
- Shortly after the row-address is loaded, the column-address is
  → applied to the address pins &
  → loaded into **CAS'.**

- The information in the latch is decoded.
- The appropriate group of 8 Sense/Write circuits is selected.
  R/W'=1(read-operation) → Output values of selected circuits are transferred to data-lines D_0-D_7.
  R/W'=0(write-operation) → Information on D_0-D_7 are transferred to the selected circuits.

- RAS‟ & CAS‟ are active-low so that they cause latching of address when they change from high tolow.
- To ensure that the contents of DRAMs are maintained, each row of cells is accessed periodically.
- A special memory-circuit provides the necessary control signals RAS‟ & CAS‟ that govern the timing.
- The processor must take into account the delay in the response of the memory.