

Name: Ankit Sapkota

RollNo: PAS077 BEI005

Subject: Embedded System.

2074 Bhadra

1. Define Embedded System. Clarify the Statement 'Digital Camera is a good example of an Embedded System? In RTOS, explain context switching with suitable diagram.

=> An embedded system is a combination of hardware and software designed to perform a specific function, often with real-time computing constraints

Digital Camera is a good example of an embedded system due to following key characteristics:

i) Dedicated Functionality: A digital camera is specialized for capturing, processing & storing digital image. It serves a specific purpose.

i) Real-time Constraints: Digital camera often have real-time requirements, they need to respond quickly to user input and efficiently process image data in real time.

ii) Tightly Constrained Design: The design of a digital camera is tightly constrained in terms of size, power consumption and cost.

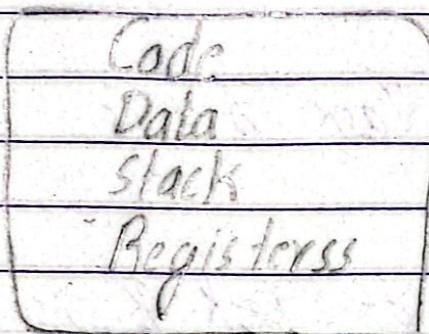
In summary, dedicated functionality, compact form factor and real time operation clarify, Digital Camera to be an good example of an Embedded System.

Content Switching : In Real time Operating Systems (RTOS), content switching allows the system to efficiently switch the from executing the content of one task to another.

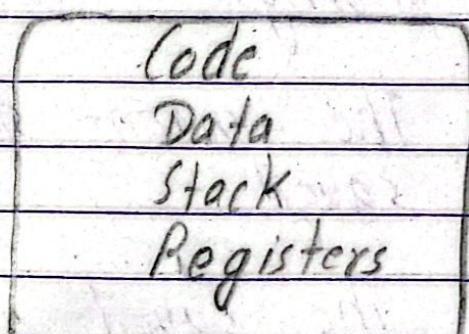
↳ It is particularly important in multitasking

environments where multiple tasks need to be executed concurrently.

Task A



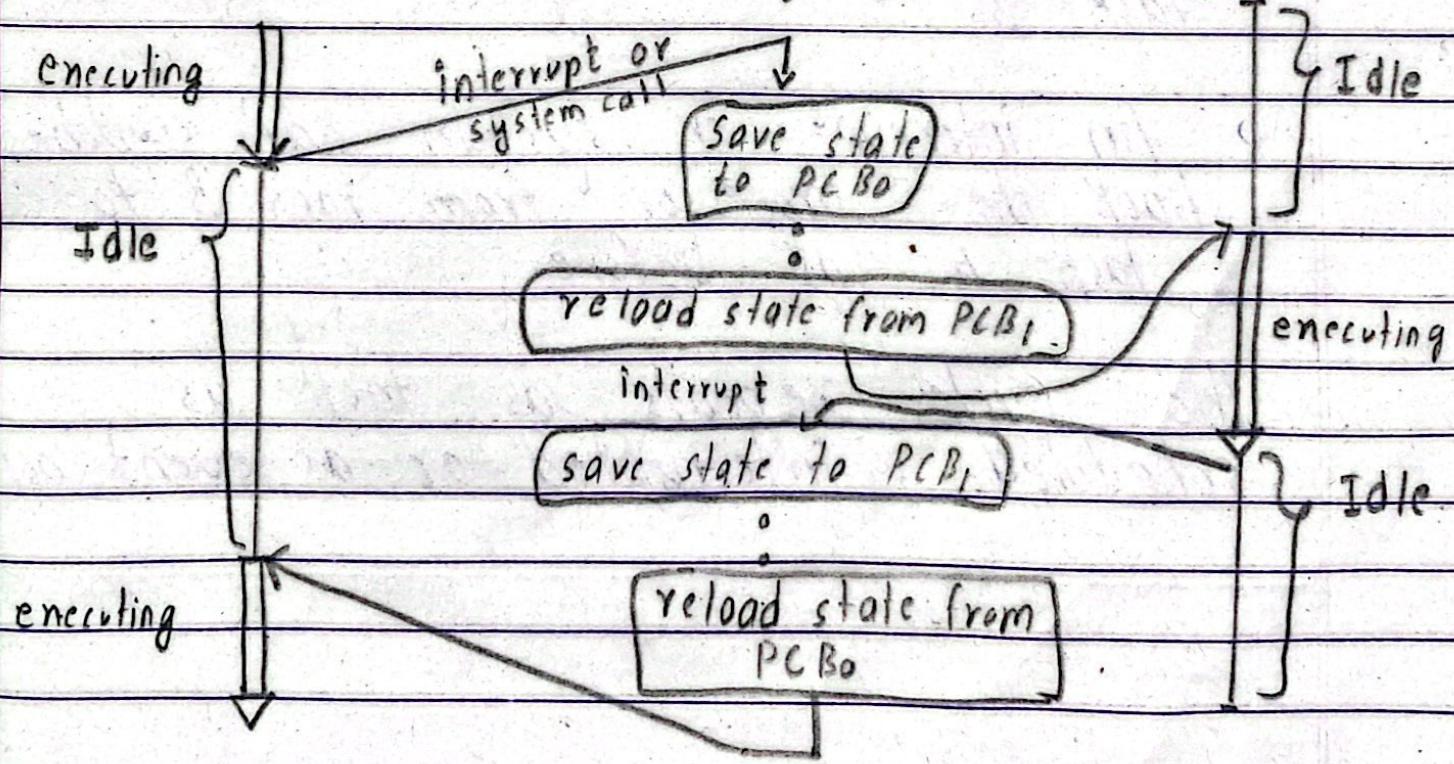
Task B



Task A

Operating System

Task B



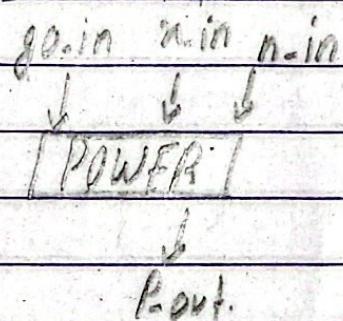
- ↳ Task A is running & its code, data, stack & registers
- ↳ An interrupt occurs, or the scheduler decides to switch task based on priority.
- ↳ The current state of Task A is saved
- ↳ The saved content of Task B is loaded into the processor.
- ↳ The processor now executes Task B.
- ↳ An interrupt or system call switches back the processor from task B to task A as before.

The cycle repeats as task as scheduled, interrupted or as event occur.

Qn.7 Design the a single processor to determine the value of n to the power n . Start the design from the function computing the desired result, FSMD, datapath and controller.

=> Solution

A. Black Box View



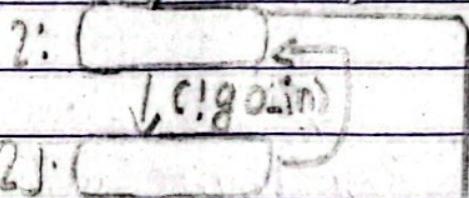
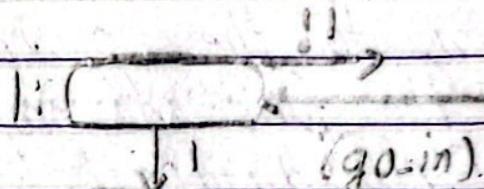
B. Functionality Code.

```

int n, p, n;
while (1) {
    while (!go-in);
    n = n-in;
    n = n-in;
    m = 1;
    while (n > 0) {
        m = m * n
        n = n - 1
    }
}
    
```

$P_{out} = m;$

C. FSMD

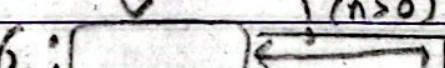


2J:

3: ($n = n_{in}$)

4: ($n = n_{in}$)

5: ($m = 1$)

6:  (n > 0)

7: ($m = m * n$)

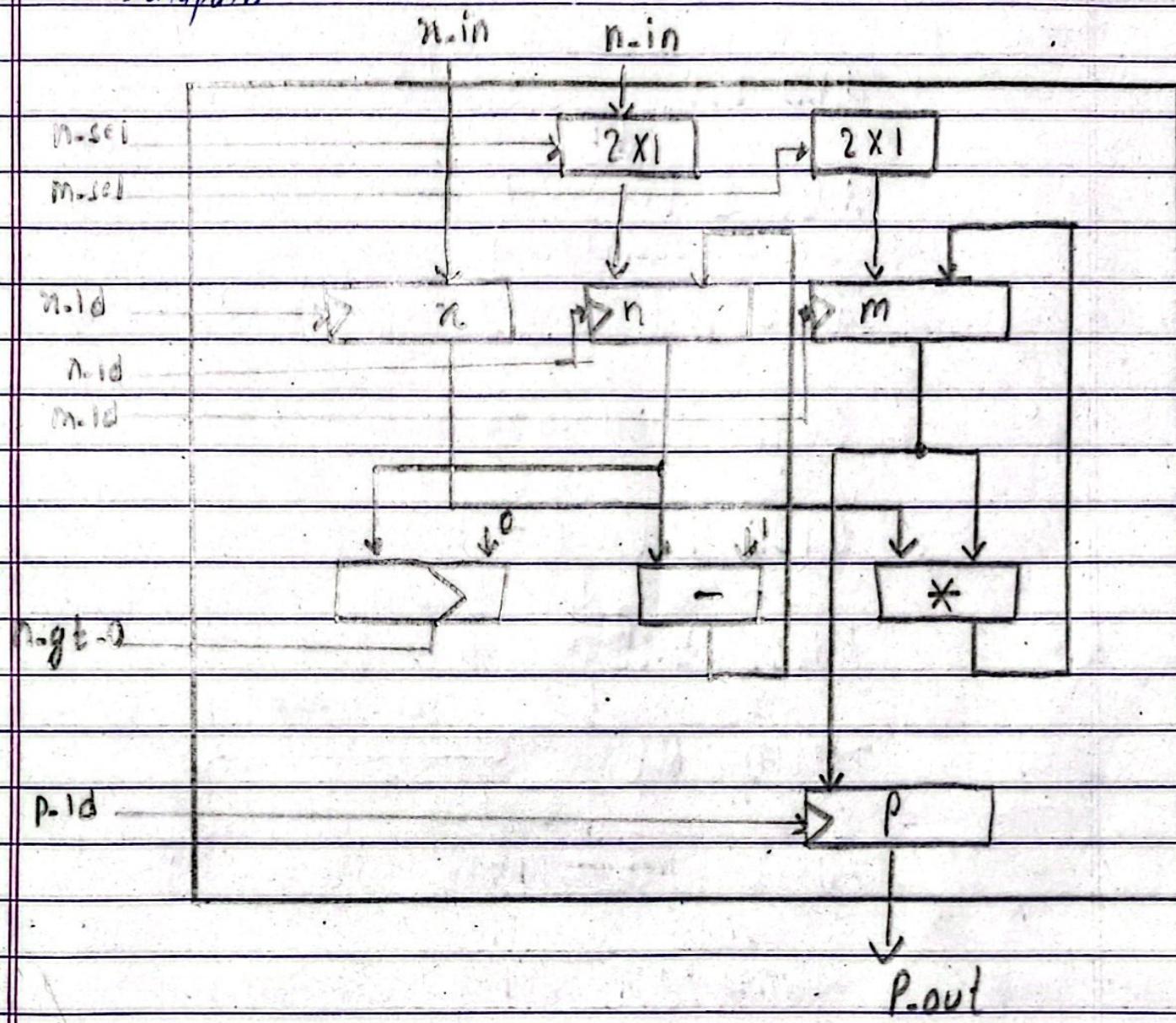
8: ($n = n - 1$)

6J: 

9: ($P_{out} = m$)

1J: 

D. Datapath



E. FSM

0000 1: [] $\xrightarrow{\downarrow}$

0001 2: [] $\xrightarrow{\downarrow \text{!} \text{!} \text{!} \text{!} \text{!} \text{!}}$

0010 27: []

0011 3: ($n_{\cdot}1d=1$)

0100 4: ($n_{\cdot}1d=1, n_{\cdot}sc1=0$)

0101 5: ($m_{\cdot}1d=1, m_{\cdot}sc1=0$)

0110 6: [] $\xleftarrow{\downarrow n_{\cdot}gt=0}$

0111 7: ($m_{\cdot}1d=1, m_{\cdot}sc1=1$)

1000 8: ($n_{\cdot}1d=1, n_{\cdot}sc1=1$)

1001 6J: []

1010 9: ($P_{\cdot}1d=1$)

1011 1J: []

3 Explain the design flow of embedded software development. Explain in brief about programmer's view for general purpose processor.

⇒ Design of embedded software development constitutes of two phases : 1) Implementation phase 2) Verification phase.

The process starts with Implementation phase.

a) Implementation phase : In this phase, the process of editing, compiling, assembling & linking the program code is done.

⇒ Development processor use cross compilers or cross assembler.

b) Verification Phase : Debuggers, Emulator & programmers are deployed for verification.

⇒ It involves confirming the correct implementation of software meets the specified requirements.

⇒ It is crucial for ensuring quality.

A programmer's view of a general-purpose processor involves understanding the architecture and features provided by the processor for software development.

Key aspects include :

- a) Instruction set : The set of instruction supported by the processor including format & addressing modes.
 - b) Registers : The number & types of registers available for general and specific purpose.
 - c) Interrupts : The types of interrupts supported by processor and must write Interrupt Service routine when required.
 - d) Input Output Facility : The number of input output pins available and their functions.
- ↳ Understanding Communication Protocols.

4 Define write ability and storage performance of memory. Explain associative cache mapping technique with its merits and demerits.

⇒ Write ability : The speed at which a particular memory can be written is known as write ability.

It ranges from high end memory, which can be written quickly to low end, which only can be written during fabrication.

Storage ability : Ability of memory to hold its stored bits after those bits have been written.

It ranges from memory which hold data until it is damaged to memory which begins to loose its bits almost immediately.

Associative cache Mapping: In associative cache mapping, any block from main memory can be placed in any block of the cache.

- ↳ There is no restriction on where a block can be placed.
- ↳ This flexibility eliminates conflicts but requires additional hardware to search the entire cache for a particular block during lookup operations.
- ↳ It is more complex and expensive but can provide better cache utilization and Performance.
- ↳ Appropriate replacement algorithm must be utilized for efficient cache implementation.

Merits:

- ↳ It is fast
- ↳ Easy to implement

Demerits:

- ↳ Expensive because it needs to store address along with data

5 Describe two-level bus architecture in detail. Describe priority arbitration method and compare it with daisy-chain arbitration.

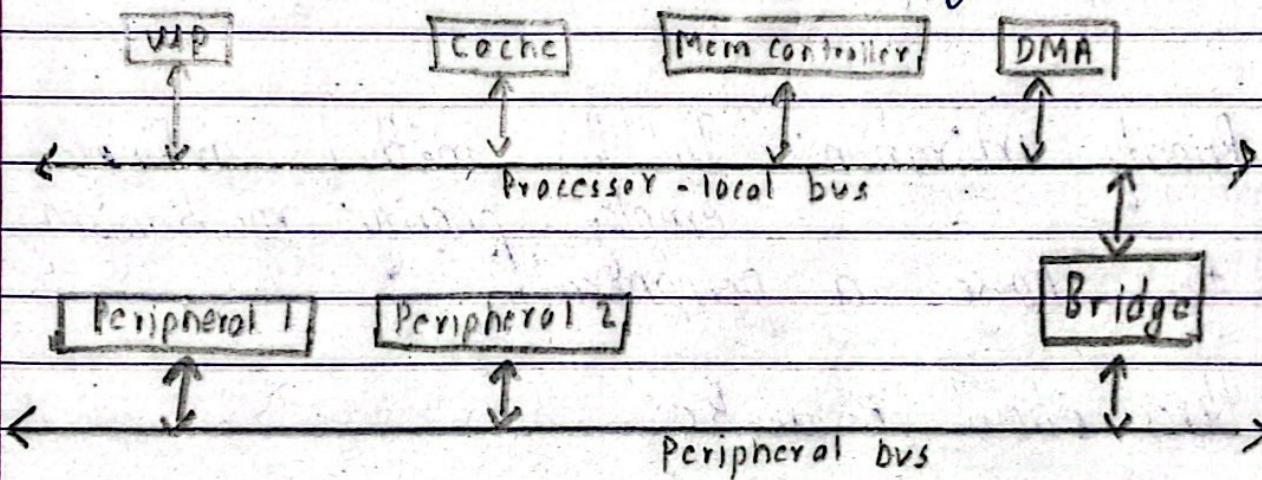


fig: two level bus architecture.

Two level bus architecture consists of:

- 1) High speed processor local bus
- 2) Low-speed peripheral bus
- 3) Bridge

- 1) Processor local bus connects high speed devices such as microprocessor, cache memory.
- 2) Peripheral bus connects to those peripheral which do not have access to processor local bus.
 - ↳ It is slower than Processor local bus.
- 3) Bridge connects two buses and performs speed synchronization for smooth data flow.

Priority arbitration: In this method, arbiter employs priority mechanism to choose a peripheral.

How it works:

- ↳ Let's say two Peripheral A & B assert request to arbiter.
- ↳ Arbiter responds based on priority.
- ↳ Processor is interrupted as soon as any peripheral asserts request.

- ↳ Processor stops its operation & responds to arbiter.
- ↳ Arbiter acknowledges peripheral based on Priority
- ↳ Peripheral communicates with processor.
- ↳ Arbiter switches to next priority peripheral as soon as the task of earlier peripheral is done.

Priority Arbitration

a) Determine access based on priorities

b) Each device has specific priority level

c) Device operate independently

d) Selection involves choosing highest priority

Daisy-chain Arbitration

Manages access in daisy chain structure

Device are connected in a chain

Devices send request down the chain

Selection involves position in the chain

6 Write any four differences between thread and process. Three processes P₁, P₂ and P₃ with estimated completion time 4, 10, 5 ms and priorities 1, 3, 2 respectively enters the ready queue together. A new process P₄ with estimated completion time 3ms and priority 0 enters the ready queue after 5ms of start of operation. Calculate WT, TAT for each process & calculate AW_T and ATAT using preemptive priority based scheduling algorithms.

\Rightarrow	Thread	Process
a)	It is a single unit of execution, a part of Process.	It is a program in execution, combines one or more threads.
b)	(cannot live independently)	Process contains atleast one thread.
c)	Inexpensive to create	Expensive to create.
d)	Context switching is fast & easy.	Context switching is complex.

Solution :

Given information is tabulated below:

Process	Entry time (ms)	Completion time (ms)	Priority
P1	0	9	1
P2	0	10	3
P3	0	5	2
P4	After 5ms of operation	3	0

Using Preemptive priority based scheduling:

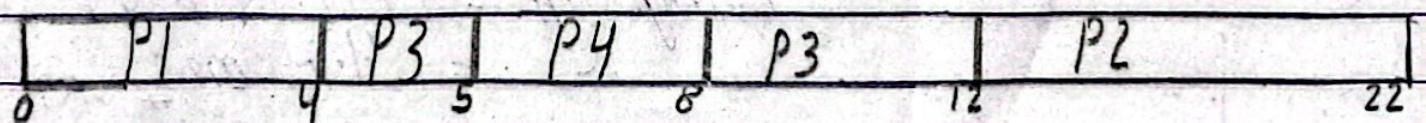


fig: Gantt Chart

Turn Around time (TAT = Completion point - Entry point)

$$P_1 = 9 - 0 = 4 \text{ ms}$$

$$P_2 = 22 - 0 = 22 \text{ ms}$$

$$P_3 = 12 - 0 = 12 \text{ ms}$$

$$P_4 = 8 - 5 = 3 \text{ ms}$$

$$\begin{aligned} \text{Average TAT} &= (4 + 22 + 12 + 3) / 4 \\ &= 10.25 \text{ ms} \# \end{aligned}$$

Waiting time (W.T = TAT - completion time)

$$P_1 = 4 - 4 = 0 \text{ ms}$$

$$P_2 = 22 - 10 = 12 \text{ ms}$$

$$P_3 = 12 - 5 = 7 \text{ ms}$$

$$P_4 = 3 - 3 = 0 \text{ ms}$$

$$\begin{aligned} \text{Average TAT} &= (0 + 12 + 7 + 0) / 4 \\ &= 4.75 \text{ ms} \# \end{aligned}$$

Name: Ankit Sapkota

Rolln: PAS077 BEI005

Subject: Embedded System

2075 Bhadra

1. What is an embedded system? Differentiate it with non-embedded system with suitable example.
In RTOS, describe mutual exclusion through sleep/wake for task synchronization.

⇒ An embedded system is a combination of hardware and software designed to perform a specific function, often with real-time computing constraints.

Embedded System

Non-Essential System

- | | |
|---|--|
| 1) Dedicated to specific task or functions | 1) General Purpose computing systems |
| 2) Operates in resource-constrained environment | Typically has more resources available |

3) Often requires real-time operation 3) Real-time constraints are less critical

4) Examples include car engine control systems

4) Examples include personal computers, laptops, servers.

In real-time operating systems (RTOS), mutual exclusion is a synchronization mechanism used to ensure that only one task at a time can access a shared resource or critical section of code. Sleep and wake mechanisms are often employed for task synchronization in the context of mutual exclusion. Let's delve into how it is typically implemented:

1) Sleep: Consider Task A and Task B both needing access to a shared resource. If Task A finds the resource in use, it may enter a sleep state, allowing Task B to execute.

2) Wake: When the task holding the shared

resource completes its critical section or releases the resource, it notifies the waiting tasks that they can proceed.

Wake and sleep is often done through a synchronization primitive such as semaphores:

- Semaphore is a synchronization mechanism that can be used to implement mutual exclusion. It typically has a count associated with it.
- When a task enters a section, it may decrement the semaphore count (locking the semaphore)
- If the semaphore count is already zero (indicating that the critical section is in use), the task may go to sleep.
- When the task exits the critical section, it increments the semaphore count (unlocking the semaphore, potentially waking up a waiting task).

Qn.2 What is optimization? What are the parameter you consider for optimization of single purpose processors.

⇒ Optimization is the task of making design metric values the best possible. Optimization can be done by simplifying the resulting design of any system utilizing various techniques. Different states in the FSM can be removed which does nothing and are redundant. Also, we can share a component for same operations in different states, and hence minimizing the size of the system as well as its cost.

The parameters to consider for optimization can vary based on process of optimization, here are some of the common parameters to consider for the optimization of single purpose processor.

a) Original Program: We should analyze different program attributes and try to develop alternative algorithm that are more efficient. We can analyze the algorithm in terms of time complexity and space complexity. Number of computations can be a form of time complexity whereas the size of variables required corresponds space complexity.

b) FSMD : Each state in an FSMD is assigned with operations from the desired program; this process is also termed as scheduling. The scheduling process can be improved by.

- ↳ Merging states
- ↳ Eliminating states
- ↳ Separating states

c) Datapath : During the datapath design, the task of selecting a RT components for particular operation is termed as allocation. Whereas the task of mapping operations from the FSMD to allocated

components is termed as binding. The optimization in datapath can be done by:

↳ Sharing functional units

↳ Use of multi-functional units.

d) FSM: Optimization in FSM can be done by:

i) State encoding: It is a task of assigning a unique bit pattern to each state in an FSM.

ii) State minimization: It is the task of merging equivalent states into a single state.

3) Define datapath and controllers of a general purpose processor. Explain ASIP with its types.

=> Datapath : The datapath is a general-purpose processor refers to the collection of functional units, such as ALU (Arithmetic Logic Units), registers and busses that perform operations on data. It represents the execution core of the processor where arithmetic and logical operations take place. The datapath contains the necessary components to process and manipulate data according to the instructions fetched from memory.

Controllers : The controller is responsible for managing the operations and control signals within the processor. It decodes the instructions fetched from memory, generates the necessary control signals to coordinate the activities of the datapath, and ensures the correct sequencing of operations. The controller plays a crucial role in directing

the flow of data and control signals to execute instructions accurately.

An ASIP is a processor designed for a specific application domain, optimizing both hardware and instruction set to meet the requirements of particular applications. It strikes the balance between the flexibility of general purpose processors and the efficiency of application-specific circuits. ASIP's are tailored to execute specific types of algorithms or task efficiently.

Types

- a) Microcontrollers : They are specific to applications that perform a large amount of control oriented tasks.
- b) Digital Signal Processors : These are processors which are specific to applications that process large amounts of data.

c) Less-General ASIP : These are developed to perform some very domain specific processing while allowing some degree of programmability.

- 4) Define write ability and storage permanence of memory. Design a ROM to store the following information.

X	Y	Z	F1	F2	F3	F4
0	0	0	0	0	1	0
0	0	1	1	1	0	0
0	1	0	0	1	0	1
0	1	1	1	1	1	1
1	0	0	0	0	1	1
1	0	1	0	1	0	1
1	1	0	1	0	1	0
1	1	1	0	0	1	1

⇒ Write Ability: It refers to the manner and speed that a particular memory can be written

↳ It also represents the number of times a memory can be programmed or written into.

Storage Performance: It refers to the ability of memory to hold its stored bits after those bits have been written

↳ Volatile and Non-volatile attributes are commonly used to divide memory types into two categories along the storage performance axis.

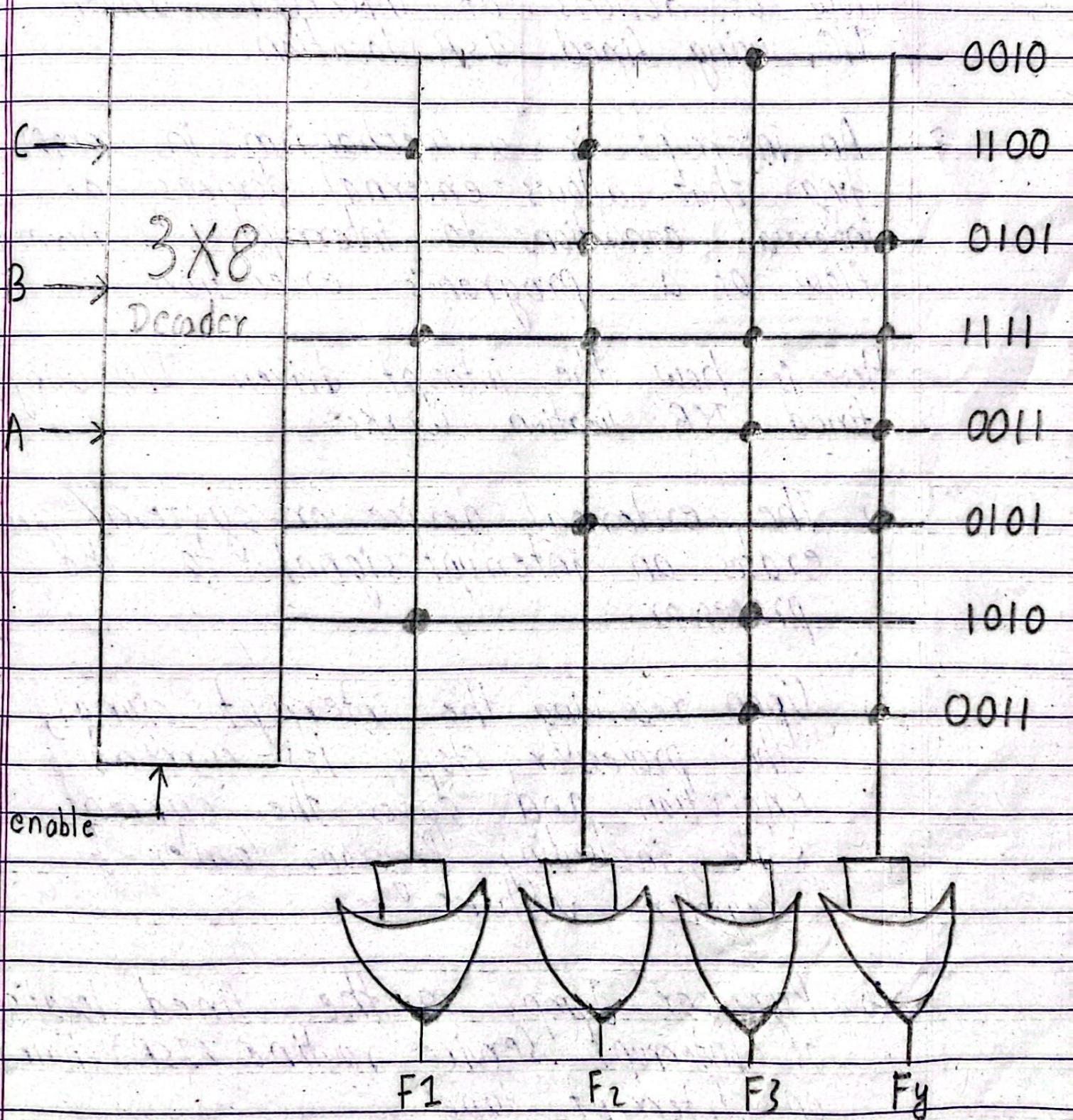


fig: ROM Design

5) a) What is interrupt? Explain summary of flow of actions of interrupt driven I/O using fixed ISR location.

⇒ An interrupt is a mechanism in computing that allows external devices or internal conditions to interrupt the normal flow of a program's execution.

Here is how the interrupt driven I/O using fixed ISR location works.

- ↳ The external device or system generates an interrupt signal to the processor.
- ↳ Upon receiving the interrupt signal, the processor stops its current execution and saves the current state, including Program Counter & register values.
- ↳ Processor Jumps to the fixed location of Interrupt Service routine (ISR) based on Interrupt type.

- ↳ ISR performs the necessary operations to handle the interrupt.
- ↳ After completing IRS, processor restores its previous states.
- ↳ Processor resumes the normal program execution from where it was interrupted.

b) What is arbitration? With neat diagram explain Daisy-chain arbitration.

⇒ Arbitration in computing refers to the process by which multiple devices or components contend for control of a shared resource such as bus or a communication channel.

Here's an explanation, how Daisy-chain arbitration works.



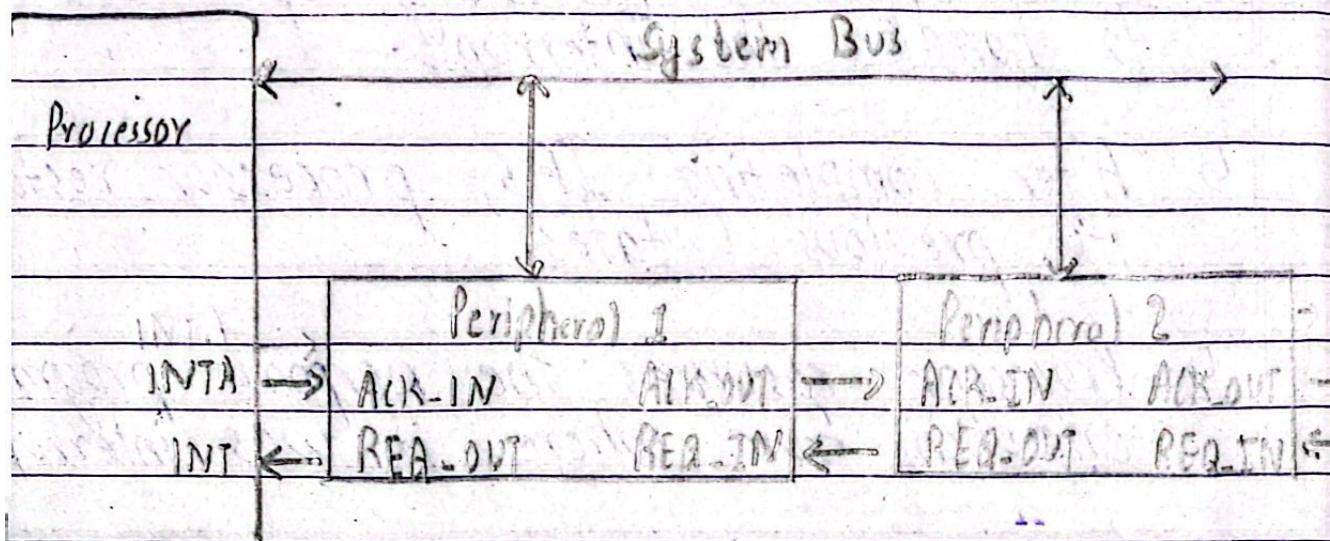


fig: Daisy chain configuration.

In Daisy chain method, devices are connected in linear or daisy-chain fashion.

- ↳ Each device is assigned a unique priority level.
- ↳ The device with the highest priority gains control of the bus or resource.
- ↳ Each device in the chain propagates the control request in the chain.
- ↳ Other devices in chain can

see that the access has been granted and reset their request signals.

- 6) The device that was granted access can now use the shared resources
- 7) After completing the task, the device releases control & acknowledge signal continues circulating through the daisy chain

In this way arbitration is formulated using Daisy chain method

Q6 Explain the condition favoring deadlock situation. Three Process P₁, P₂, P₃ with estimated completion time 5, 8, 7 ms respectively enters the ready que together. Calculate WT, TAT for each process and calculate AWT and ATAT using Round Robin Pre-emptive scheduling algorithms with time slice of 2ms.

⇒ Deadlock occurs under following four conditions:

Mutual Exclusion

- i) Hold & wait: Atleast one resource is held at non-shareable mode.
- ii) Hold & wait: A process must be holding at least one resource and waiting to acquire additional resource.
- iii) Non preemption: Resource can't be forcibly taken away from a process
- iv) Circular wait: A circular chain of processes exists, where each process is waiting for the resource held by next process in chain

Solution:

$$\text{time slice} = 2 \text{ ms}$$

Completion time of $P_1, P_2, P_3 = 5, 8, 7 \text{ ms resp}$

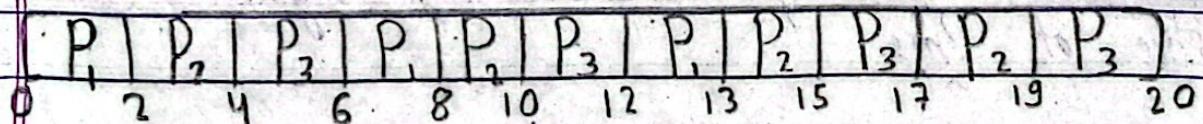


fig: Gantt chart

Turn Around time Calculation

$$P_1 = 13 - 0 = 13 \text{ ms}$$

$$P_2 = 19 - 0 = 19 \text{ ms}$$

$$P_3 = 20 - 0 = 20 \text{ ms}$$

$$\begin{aligned} \text{Average Turn Around Time} &= (13 + 19 + 20) / 3 \\ &= 17.33 \text{ ms} \end{aligned}$$

Waiting time calculation [$W.T = \text{Turn around time} - \text{completion time}$]

$$P_1 = 13 - 5 = 8 \text{ ms}$$

$$P_2 = 19 - 8 = 11 \text{ ms}$$

$$P_3 = 20 - 7 = 13 \text{ ms}$$

$$\begin{aligned} \text{Average Waiting Time} &= (8 + 11 + 13) / 3 \\ &= 10.66 \text{ ms} \end{aligned}$$

Name: Ankit Sapkota Rolln: PAS077BEI005

DATE	/	/
PAGE No.		

2076 Bhadra

1. Define Embedded System & Classify the embedded system based on generations. What is task synchronization and explain the task synchronization using 'Busy/Wait'.

=> An embedded system is a combination of hardware & software designed to perform a specific function, often with real-time computing constraints.

Embedded systems can be classified into different generations based on their complexity, architecture and capabilities.

1. First Generation:

- ↳ Designed using 8 bit up.
- ↳ Simple hardware & limited power
- ↳ Eg: Motor controllers

2. Second Generation:

- ↳ Designed using 16-bit up
- ↳ More complex hardware and more power
- ↳ Eg: Data Acquisition system.

3. Third Generation:

- ↳ Built on more advance 32 bit 16-bit up's.
- ↳ Enhanced functionality & computing
- ↳ Eg: Digital Signal Processor.

4. Fourth Generation:

- (System-on-chip)
- ↳ Reconfigurable processors & multicore processors
 - ↳ Use high performance real time embedded OS.

Coordination of operations performed by multiple tasks or threads to ensure correct & orderly execution of a system is known as Task synchronization

Busy / Wait Task Synchronization: Consider Task A and Task B, sharing a common resource

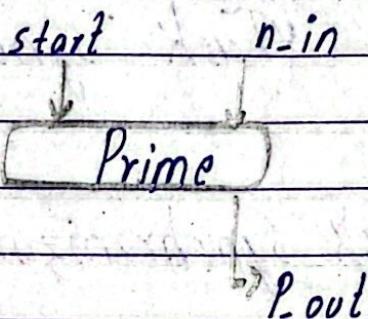
- ↳ Task A when attempting to access the resource, repeatedly checks if resource is available in Busy / Wait Loop.
- ↳ The loop keep executing until resources become available.
- ↳ Once resource is available, the task exits the busy / wait loops.
- ↳ Busy / Wait can lead to high CPU utilization during waiting periods.

2. Design a custom single-purpose processor that display 1 or 0 if the input integer is prime or not showing all the steps.

DATE _____
PAGE No. _____

⇒ Solution :

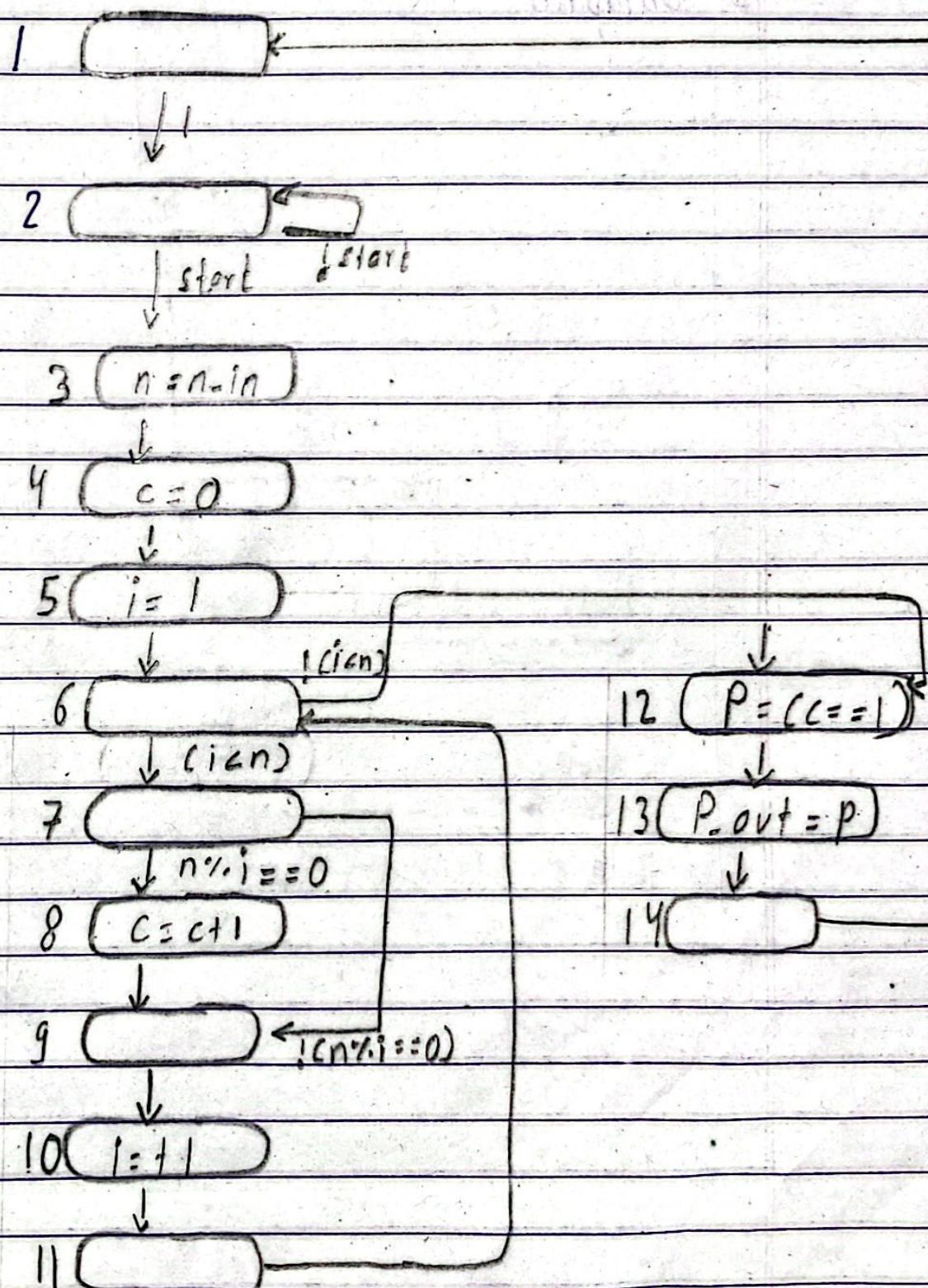
A. Black Box



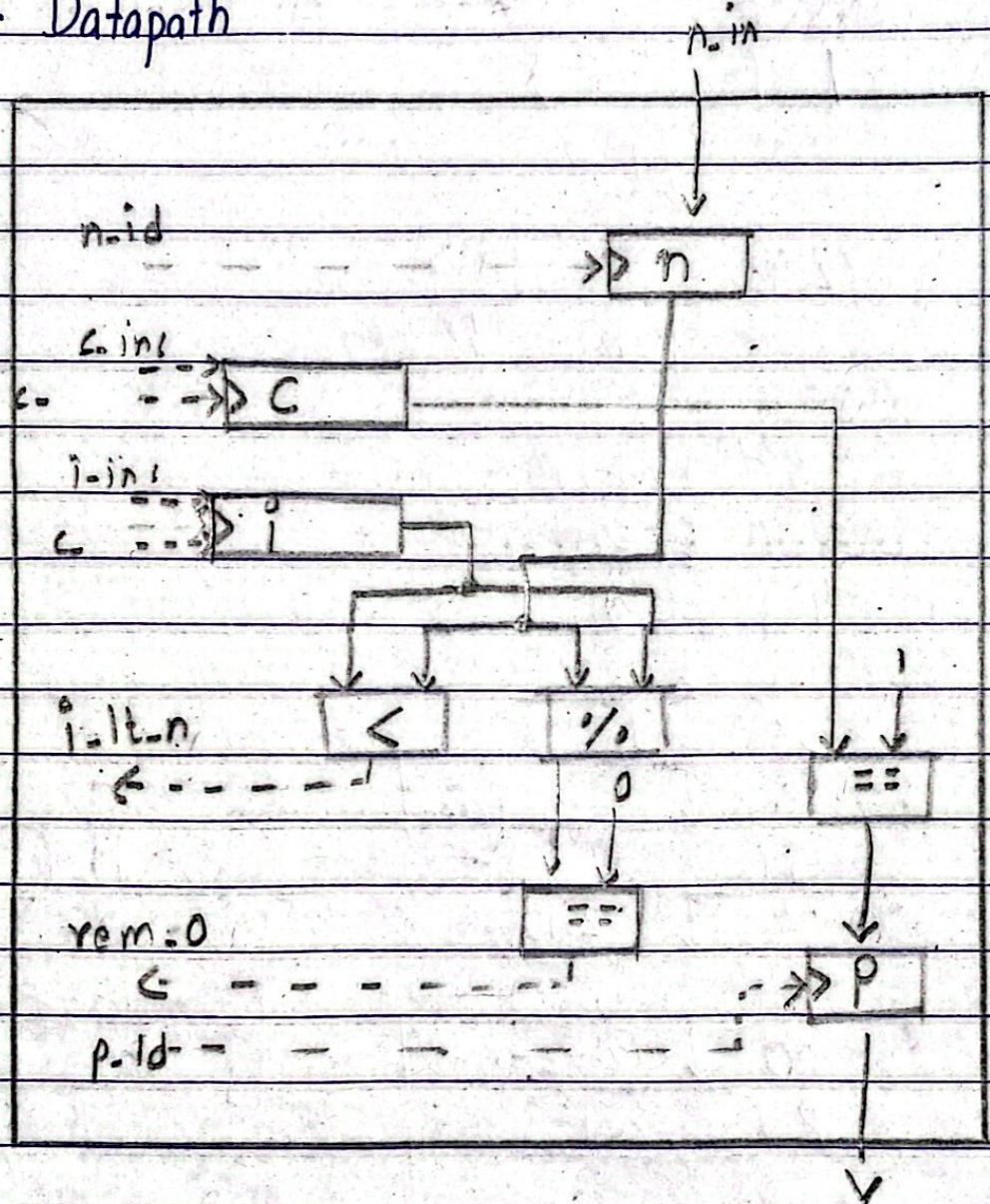
B. Functionality Code

```
int n, i, c
while(1){
    while(!start);
    n = n - in;
    c = 0;
    i = 1;
    while(i < n) {
        if (n % i == 0):
            c = c + 1;
            i = i + 1;
    }
    p = (c == 1);
    P-out = p;
```

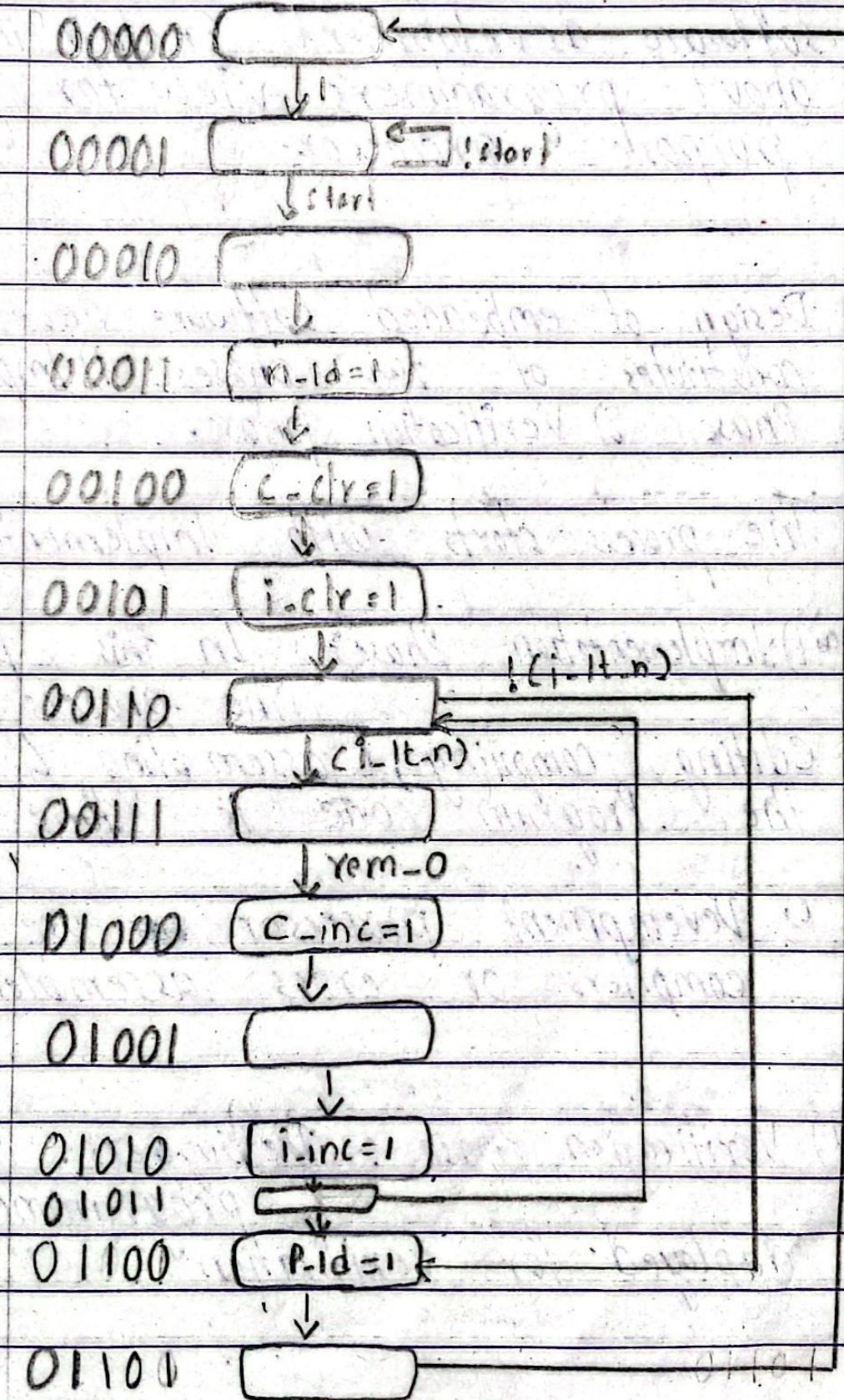
C. FSMD



D. Datapath



E. FSM.



3. Explain the design flow of embedded software development. Explain in brief about programmer's view for general purpose processor.

⇒ Design of embedded software development constitutes of two phases: 1) Implementation Phase 2) Verification phase.

The process starts with Implementation phase.

• 1) Implementation Phase : In this phase, the process of editing, compiling, assembling & linking the program code is done

↳ Development processor ~~use~~ cross compilers or cross assembler.

2) Verification Phase : Debuggers, emulators & programmers are deployed for verification.

- It involves confirming the correct implementation of software meets the specified requirements.
- It is crucial for ensuring quality.

A programmer's view of a general-purpose processor involves understanding the architecture and features provided by the processor for software development.

Key aspects include:

- Instruction set: The set of instructions supported by the processor including format & addressing modes.
- Registers: The number and types of registers available for storing general and specific purpose.

c. Interrupts: The types of interrupts supported by processor and must write Interrupt Service routine when required.

d. Input Output facility: The number of input, output pins available and their functions. Understanding communication protocols

4. Define two characteristics of memory: write ability and storage performance with their different levels. Explain replacement algorithms used in cache memory.

=) Write ability: The speed at which a particular memory can be written is known as write ability.

Different levels of write ability are:

- a. High End: Processor can simply & quickly write.
- b. Middle range: It's a bit slower than high End.
- c. Lower Range: Special device: Programmer is used to write.
Low
- d. High ^ End: Bit are stored only one during fabrication.

Storage Performance: Ability of memory to hold its stored bits after those bits have been written.

Different levels of storage Performance are:

- a. Low End: Memory begins to loose its bits almost immediately after those bits are written
- b. Lower Range: Memory hold bits as long as Power is applied

c. Middle Range : Memory hold bits for days, month, or even years.

d. High End : Memory hold bits until it is damaged.

Different replacement algorithms used in cache memory are:

a. Random replacement : Replaces the block randomly without any specific algorithm.

b. Least Recently used (LRU) : Replaces the block not used for the longest time.

c. First In First Out (FIFO) : Replaces the first entered block.

d. Least Frequently used (LFU) : The least used block of memory is replaced.

5 What is interrupt? How interrupt needed in digital device? Write a summary of flow of actions for interrupt driven I/O using fixed ISR location.

⇒ First & last Part of Question is already answered in 2075 Bhadra: Qn. 5a

Interrupt are needed in digital device for several reasons:

- a. Asynchronous events: Digital device need to respond to asynchronous events, such as external hardware signal.
- b. Real-time response: Interrupt allow for real time responsiveness to time sensitive task.
- c. Peripheral Devices: Interrupt are crucial for maintaining & managing interactions with peripheral devices

6. Explain process life cycle with process state diagram. Three processes - - - scheduling.

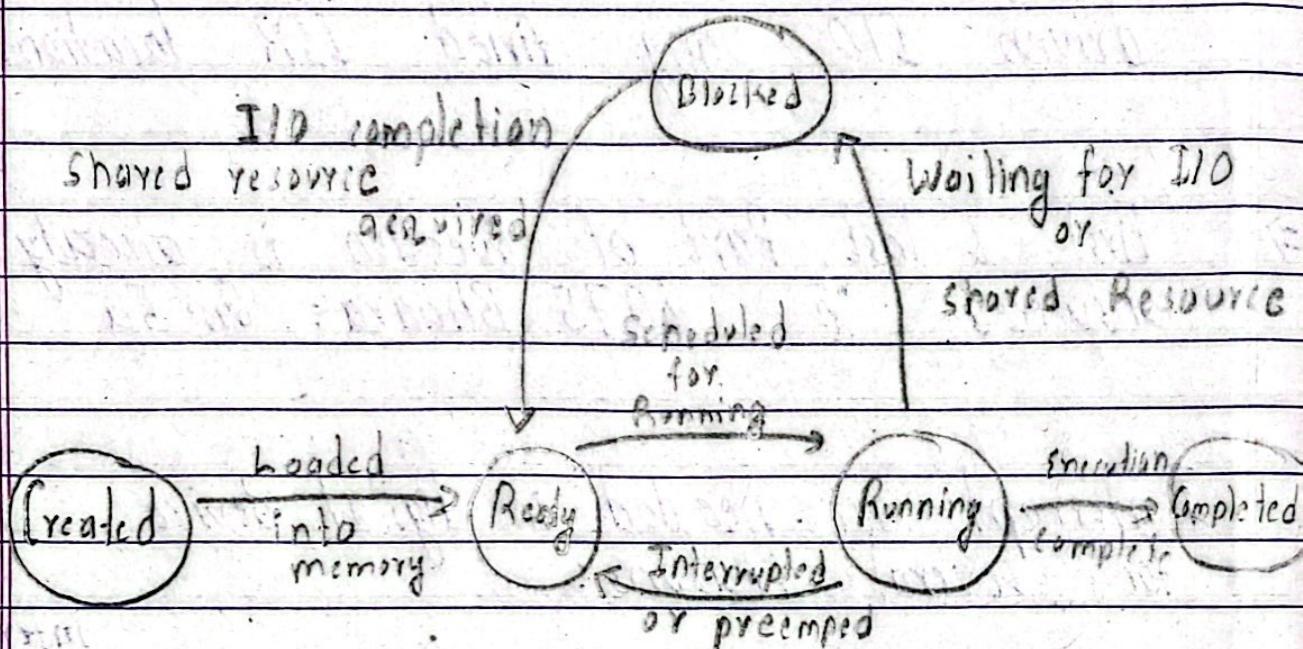


fig: Process life cycle.

- ↳ First stage is the creation of a process
- ↳ Process is loaded into memory and awaiting for Processor time at Ready state
- ↳ Process is executed at Running stage
- ↳ Temporarily suspended task are sent

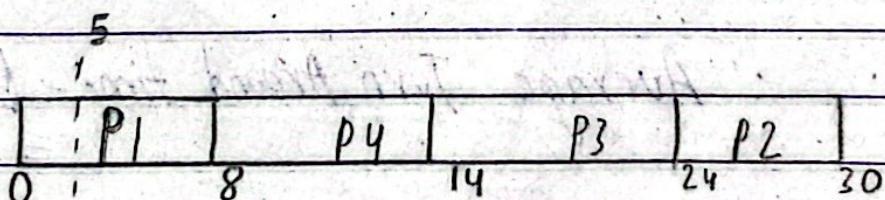
to blocked state

- After complete execution process ends with complete state.

Solution

Given information is tabulated below:

Process	Entry time	Completion time	Priority
P1	0	8	0
P2	0	6	3
P3	0	10	2
P4	5ms after P1 starts	6	1



Execution Sequence of Process

Waiting time calculation ($WT = \text{Execution starting Point} - \text{Entry Point}$)

$$P_1 = (0-0) = 0 \text{ ms}$$

$$P_2 = (24-0) = 24 \text{ ms}$$

$$P_3 = (14-0) = 14 \text{ ms}$$

$$P_4 = (8-5) = 3 \text{ ms}$$

$$\therefore \text{Average Waiting time} = (0+24+14+3)/4 \\ = 10.25 \text{ ms}$$

Turn Around Time calculation ($TAT = \text{Completion point} - \text{Entry Point}$)

$$P_1 = (8-0) = 8 \text{ ms}$$

$$P_2 = (30-0) = 30 \text{ ms}$$

$$P_3 = (24-0) = 24 \text{ ms}$$

$$P_4 = (14-5) = 9 \text{ ms}$$

$$\therefore \text{Average Turn Around time} = (8+30+24+9)/4 \\ = 17.75 \text{ ms}$$

2077
Chaitra

DATE

PAGE No.

1. Explain Different purpose of embedded system with examples.

⇒ Purpose of Embedded Systems:

a) Control : Used to control applications to regulate and manage devices

Eg: Engine Control Units (ECUs)

b) Data Communication : Used to connect two or more devices which maybe in close vicinity

Eg: Modems and routers.

c) Data Collection : Used to collect & process various range of datas based on functionality of embedded system

Eg: Camera

d) Application specific user Interface :

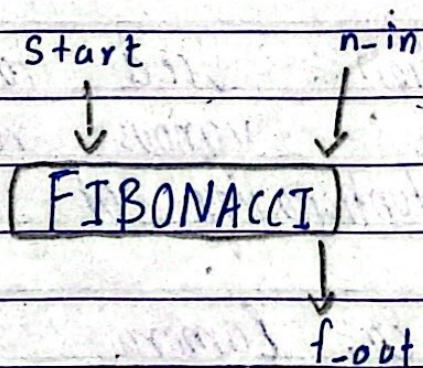
Used to provide better user interface based on application.

Eg: Touch pad, speakers, etc.

2. Design a single phase processor that output Fibonacci numbers upto ' n ' places. What are the optimization opportunity in single purpose processors.

Solution:

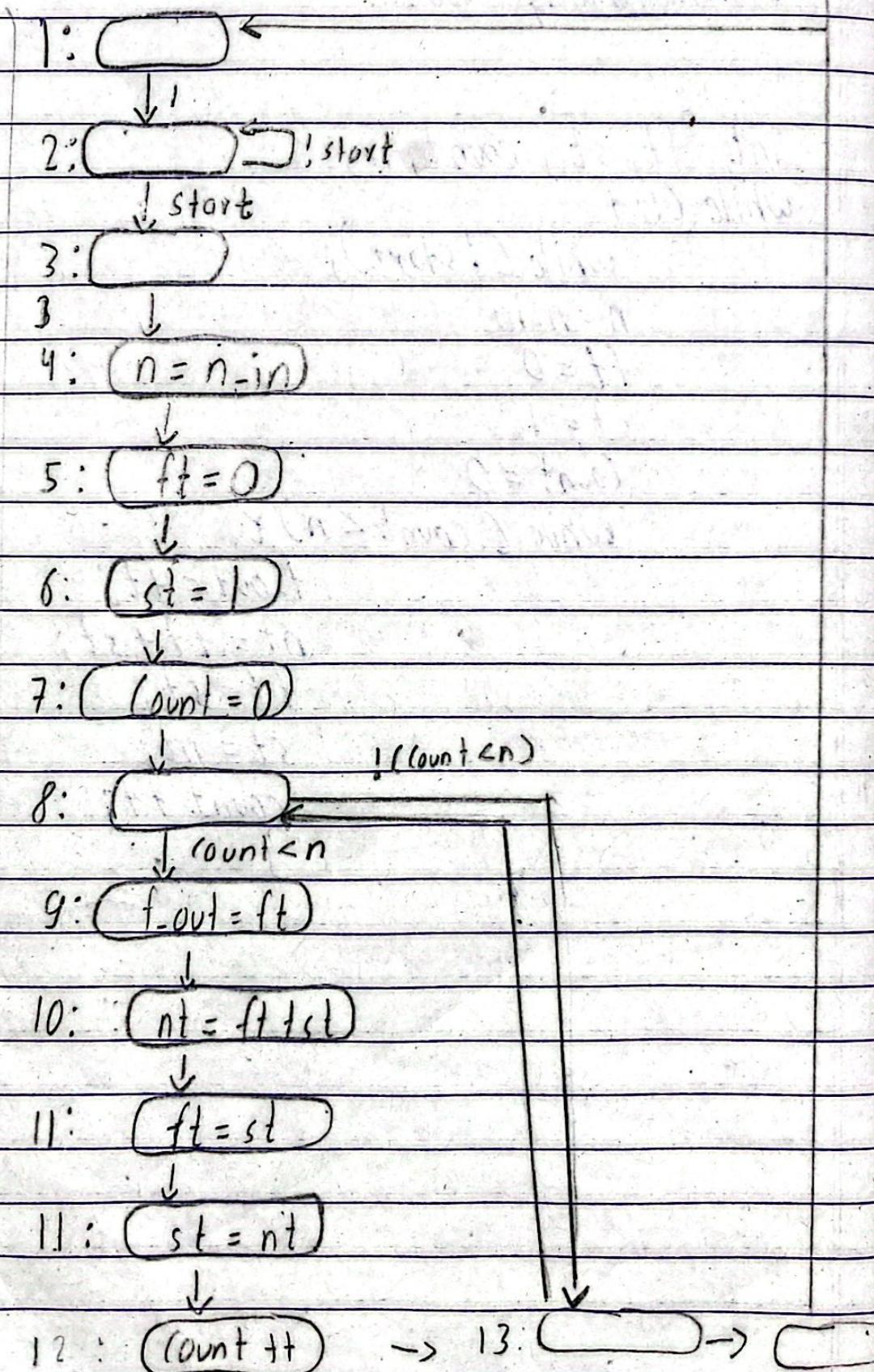
A. Block Box view



B. Functionality Code:

```
int ft, st, count, n;  
while (1){  
    while (C:'start');  
    n = n-in  
    ft = 0  
    st = 1  
    count = 0  
    while (count < n){  
        f-out = ft;  
        nt = ft + st;  
        ft = st;  
        st = nt;  
        count = ft;  
    }  
}
```

C. FSMD



PA077BEI005

DATE	/	/
PAGE No.		

D. Datapath

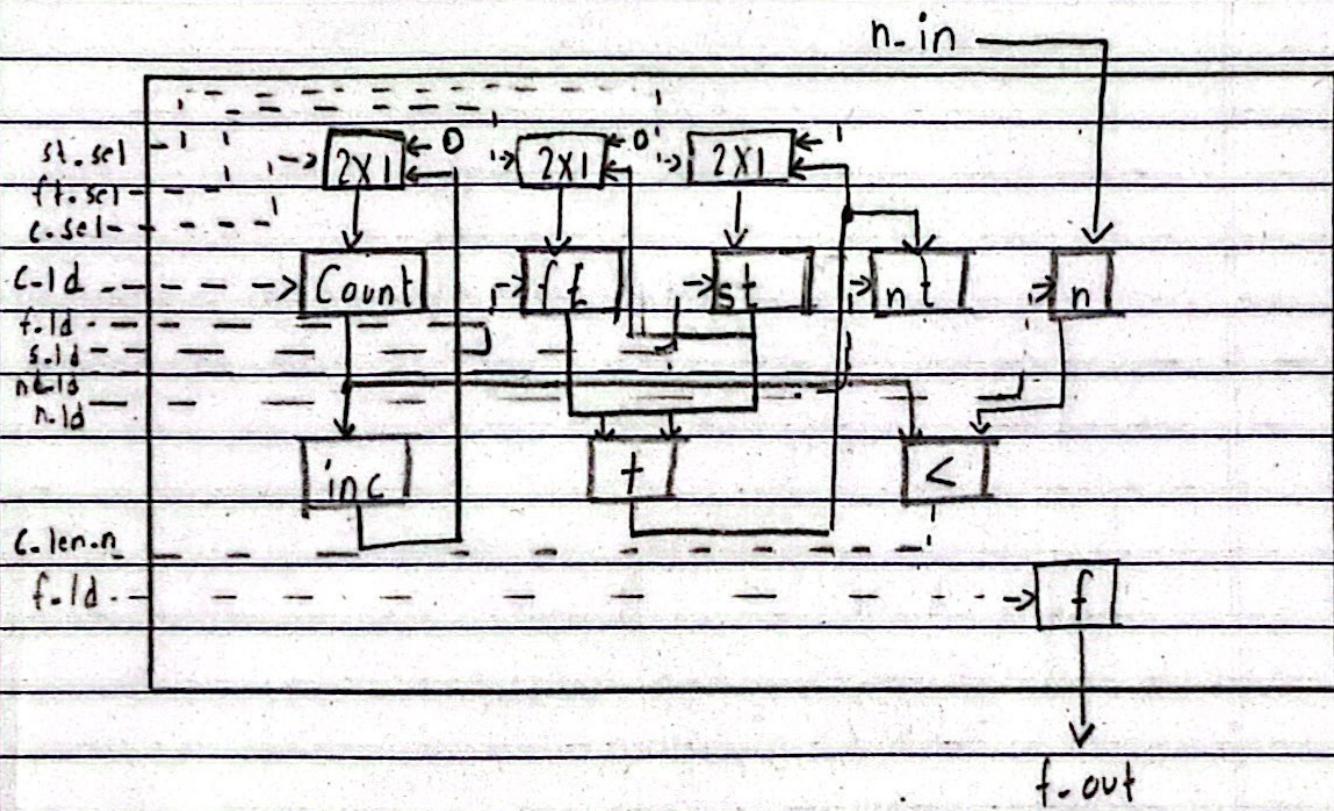
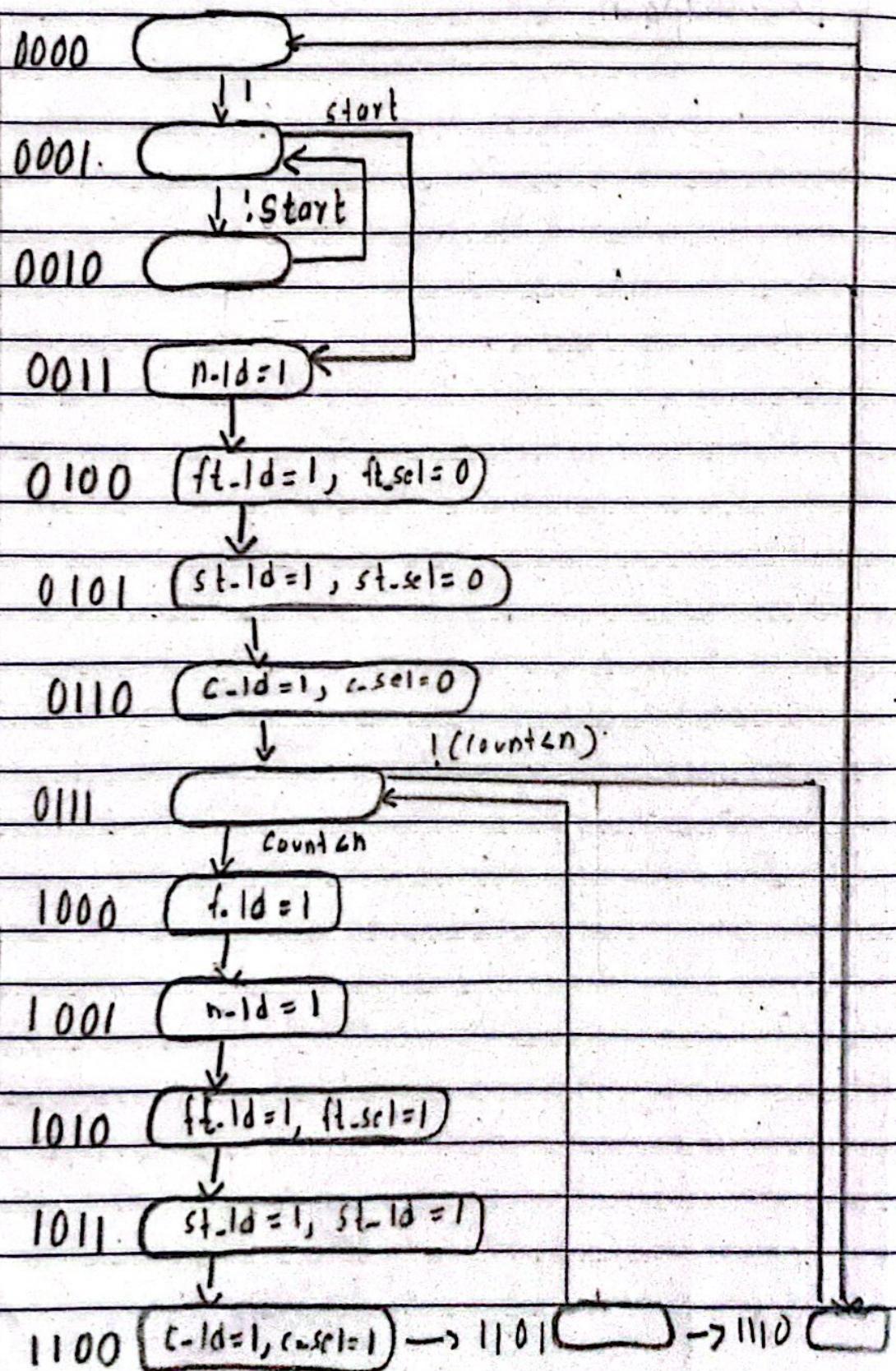


fig: Datapath Diagram of Fibonacci series generator

E. FSM Controller:



Optimizing opportunities in single purpose processor are:

- a) Original Program : Program to can be optimized to have least time and space complexity.
- b) FSMD : Several states can be merged, eliminated & separated for optimal performance.
- c) Datapath : Datapath optimization can be done by
 - ↳ Sharing functional units
 - ↳ Use of multi-functional units
- d) FSM : By state encoding & state minimization, FSM of the system can be optimized.

3

Explain the architecture of general purpose processors. What are the criteria to select the processor?

Every architecture of general purpose processors consists of 3 main components:

a) Datapath: It consists of the circuitry for transforming data and for temporary data storage.

It manipulates data through various operations such as addition, subtraction, AND, OR, etc..

It generates status signal to represent various conditions such as carry, zero, etc.

It contains Registers for temporary data storage.

b) Control Unit: It sequences through the states and generate control signals to read instructions into the instruction register.

- ↳ It controls the flow of data between ALU, registers & memory
- ↳ It determines next value of Program counter.

i) Memory : It is used to store information for medium or long term.

Criteria for selecting the processor are :

- ii) Clock speed : Speed can be compared for selection based on the clock speed if the instruction per clock cycle is same.
- iii) Instruction per second : Number of instruction performed per second.
- ↳ Complexity of instruction must also be considered.
- iv) Dhrystone benchmark : It is a program that runs on different processors and evaluate their performance.
- v) Millions of Instruction Per second (MIPS) : It is a general

measure of computing performance

↳ Can be useful when comparing performance of processors having similar architecture.

4. Design the internal architecture of 4x4 Rom. Explain the memory write ability and storage permanence with suitable example.

⇒ 2nd Part of question already done in 2076 Bhadra Qn.4.

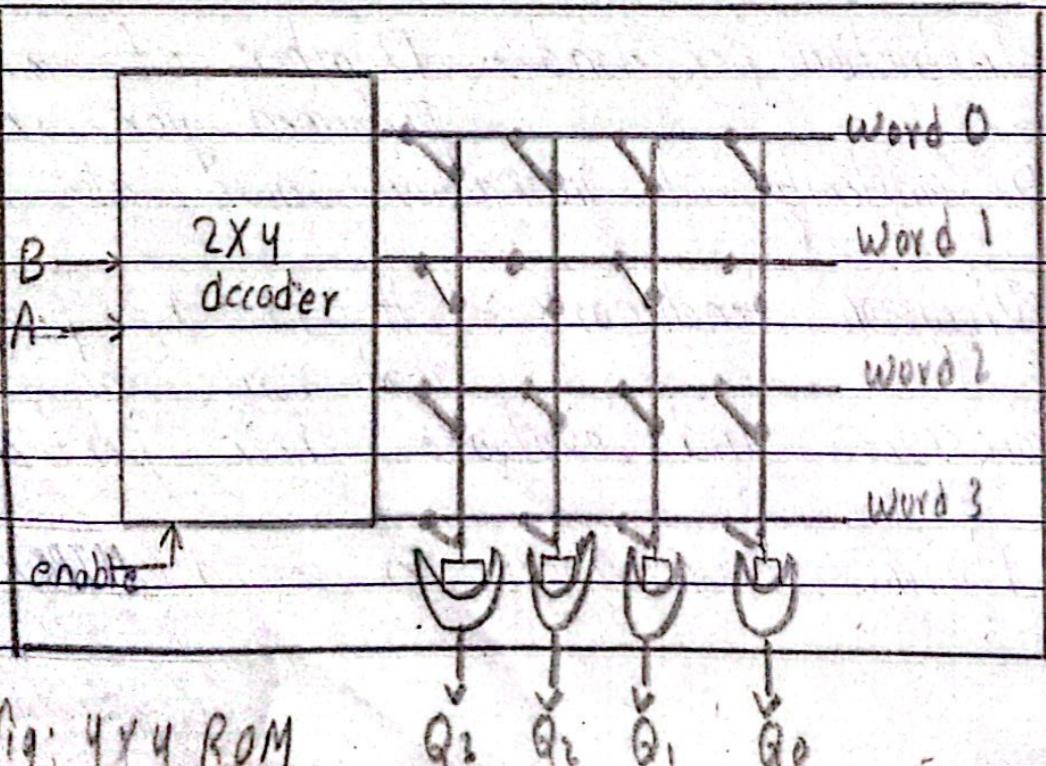


fig: 4x4 ROM

5. Define interfacing and write about needs of interfacing. Explain priority arbitration with proper illustration and types.

⇒ The process of connecting different hardware or software components to enable them to work together is known as interfacing.

Its needs are:

- a) To exchange data between different components
- b) To establish standards for compatibility between different systems
- c) To provide necessary communication between devices
- d) To integrate various components in a complex system.

Priority Arbiter: It is a single purpose processor, used to arbitrate among various requests from peripherals.

- It employs priority mechanism to choose a peripheral.

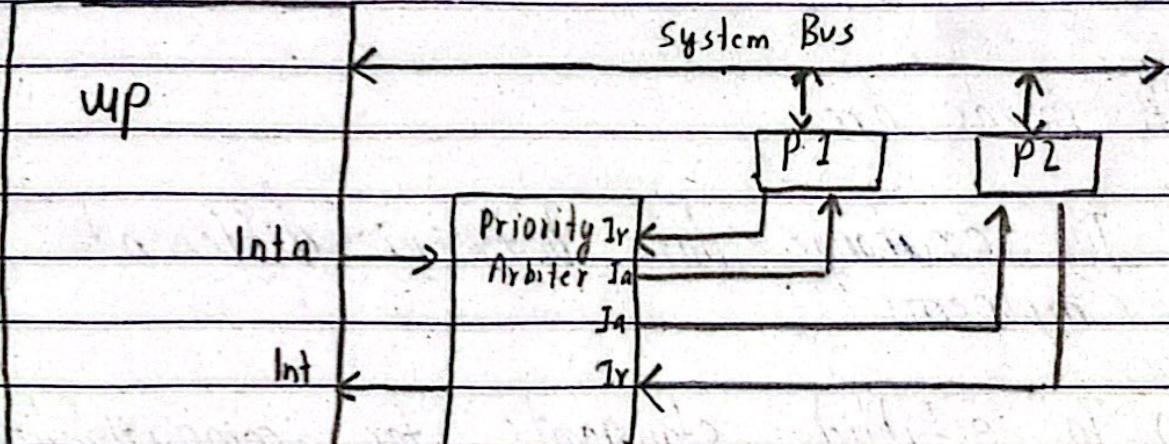


fig: Arbitration using Priority Arbiter

- Initially, MP is busy in its own task.
- Both Peripheral can assert request to arbiter.

- ↳ Processor is interrupted if ~~at~~- atleast one Peripheral asserts request.
- ↳ Processor stops its operation stops & responds to arbitrator.
- ↳ Arbitrator responds / acknowledges based on Priority.
- ↳ Peripheral communicates with processor
- ↳ Processor returns to its old task, ~~on~~ on the completion of ISR.

Types:-

- a) Fixed Priority :- Each Peripheral is assigned with fixed rank & arbitrator choose interrupt based on that rank.
- b) Round robin- Priority :- Each peripheral gets almost equal time service from the arbitrator arbiter. The priority change is

- ↳ The priority change on history of service.
6. Compare task, process and threads. Three Process with — time kernel shortest Job first scheduling. Describe basic function of real time^{kernel}.

Task	Process	Thread
a) Any unit of work	A program in execution	A light weight unit of execution within Process
b) Communication depends on content	Requires inter Process communication mechanism	Easier communication, share memory space
c) Fault tolerance depends on content	More fault tolerance	Less fault tolerance
d) Task contains number of Process	Process consists of atleast one thread	Thread can't exist independent

Solution:

Given information is tabulated below:

Process	Entered At	Completion time (ms)
P ₁	0	6
P ₂	0	8
P ₃	0	2
P ₄	After 1ms of execution	4

Execution Sequence of Processes.

Point

Turn Around time calculation (Completion time - Entry Point)

$$P_1 = 12 - 0 = 12 \text{ ms}$$

$$P_2 = 20 - 0 = 20 \text{ ms}$$

$$P_3 = 2 - 0 = 2 \text{ ms}$$

$$P_4 = 6 - 1 = 5 \text{ ms}$$

$$\begin{aligned} \text{Average turn Around time} &= (12 + 20 + 2 + 5) / 4 \\ &= 9.75 \text{ ms} \end{aligned}$$

Waiting time calculation / Turn around time - completion time

$$P_1 = 12 - 6 = 6 \text{ ms}$$

$$P_2 = 20 - 8 = 12 \text{ ms}$$

$$P_3 = 2 - 2 = 0 \text{ ms}$$

$$P_4 = 6 - 4 = 1 \text{ ms}$$

$$\therefore \text{Average Waiting time} = (6 + 12 + 0 + 1) / 4 \\ = 4.75 \text{ ms}$$

Basic Function of Real time kernel are :-

- ↳ To manage the time of microprocessor & to ensure that time-critical events are processed as efficiently as possible
- ↳ Schedule task or threads based on their priority & deadlines.
- ↳ Provide mechanism for inter-task communication
- ↳ They need to manage memory effectively in the resource constraint environment

4) Handle interrupts and minimize interrupt latency

In summary, a real time operating system kernel provides the essential infrastructure and mechanism for real-time systems.

1. Define design metric and explain various design metrics of embedded system

\Rightarrow A measurable feature of the system's performance, cost, time of implementation and safety, etc is called design metric.

Here are some design metrics:

a) NRE cost (non-recurring engineering cost): It represents the monetary cost for designing the system.

b) Unit cost: Monetary cost of manufacturing each unit, including NRE cost.

c) Size: Physical space req' for the system.

d) Power: Amount of Power consumed by system.

2. Design a single purpose processor that

gives the numbers. So translate if state trans required da

\Rightarrow Solution:

A. I

B. Functionality

while (1){

 while (C

 a = a - i;

 b = b - i;

 c = 2;

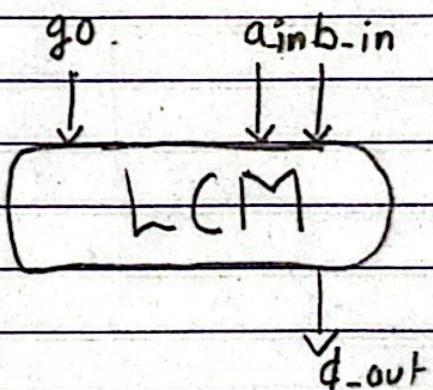
2078
Chaitra

DATE / /
PAGE No.

gives the LCM of two digit 8-bit numbers. Start with the algorithm, translate it into state diagram and state transition table. Also draw the required data-path.

⇒ Solution:

A. Black Box View.



B. Functionality code.

```
while(1){  
    while(!go){};  
    a = a_in;  
    b = b_in;  
    c = ?
```

while !(a mod b == 0) {

a = a * c;

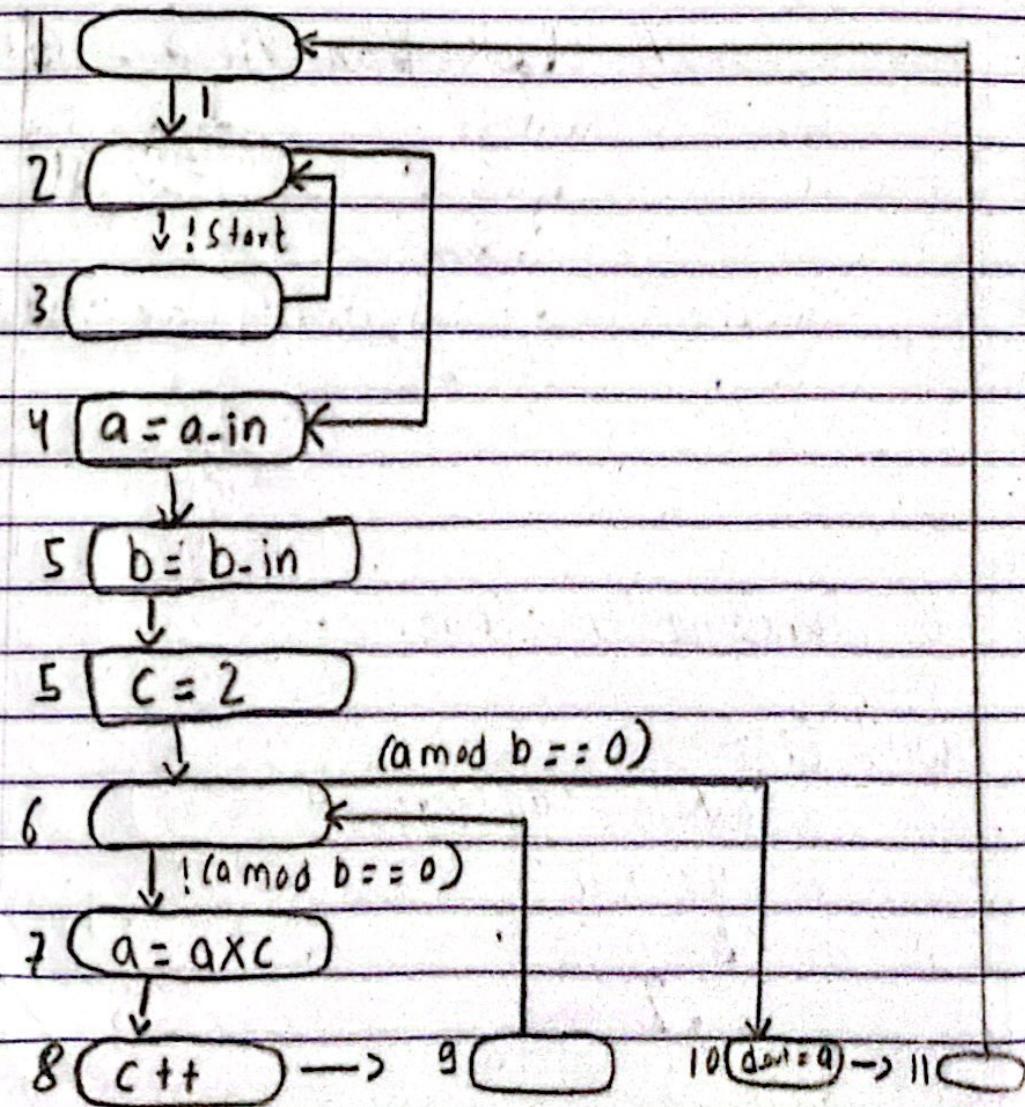
c++;

}

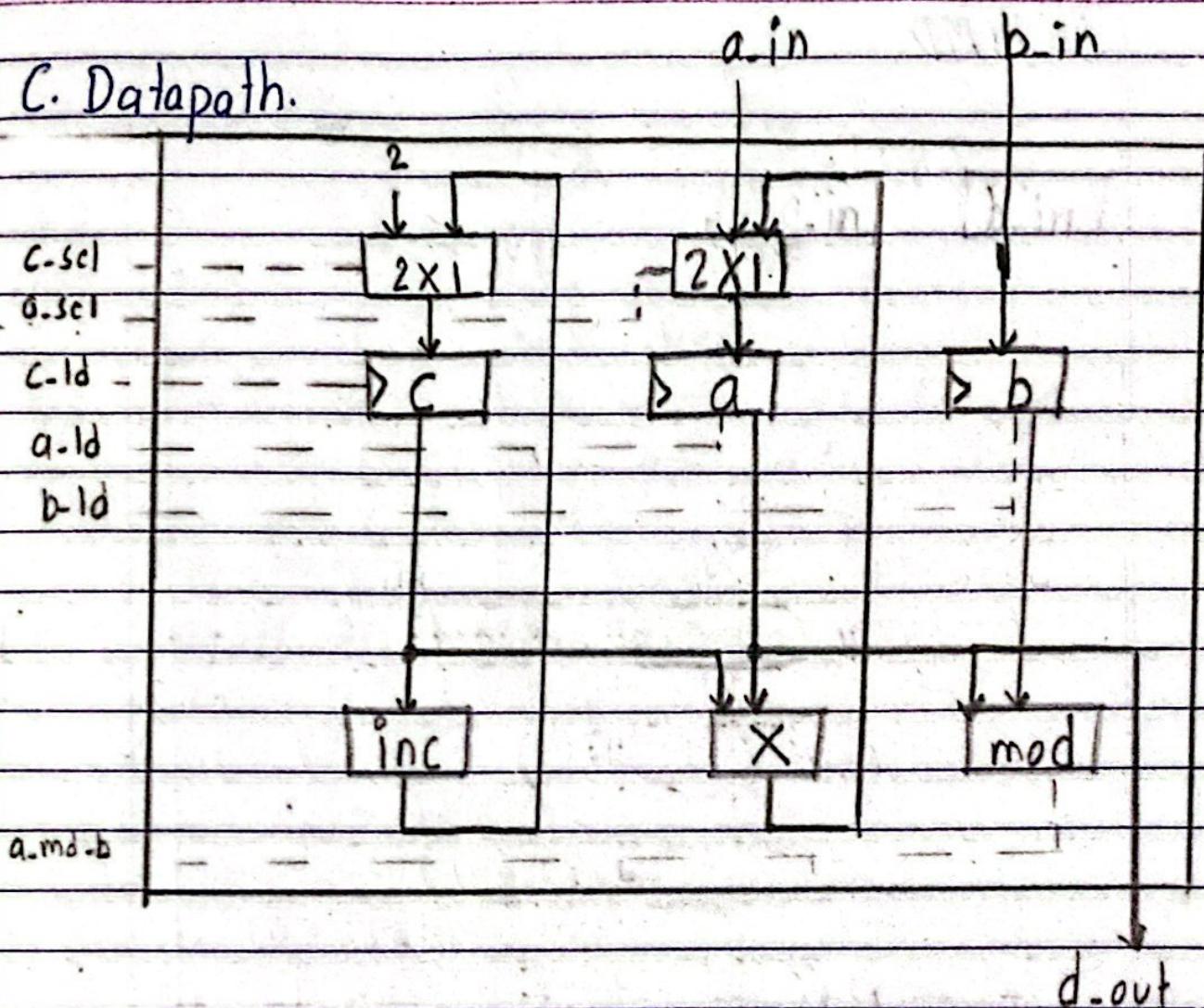
d_out = a

3

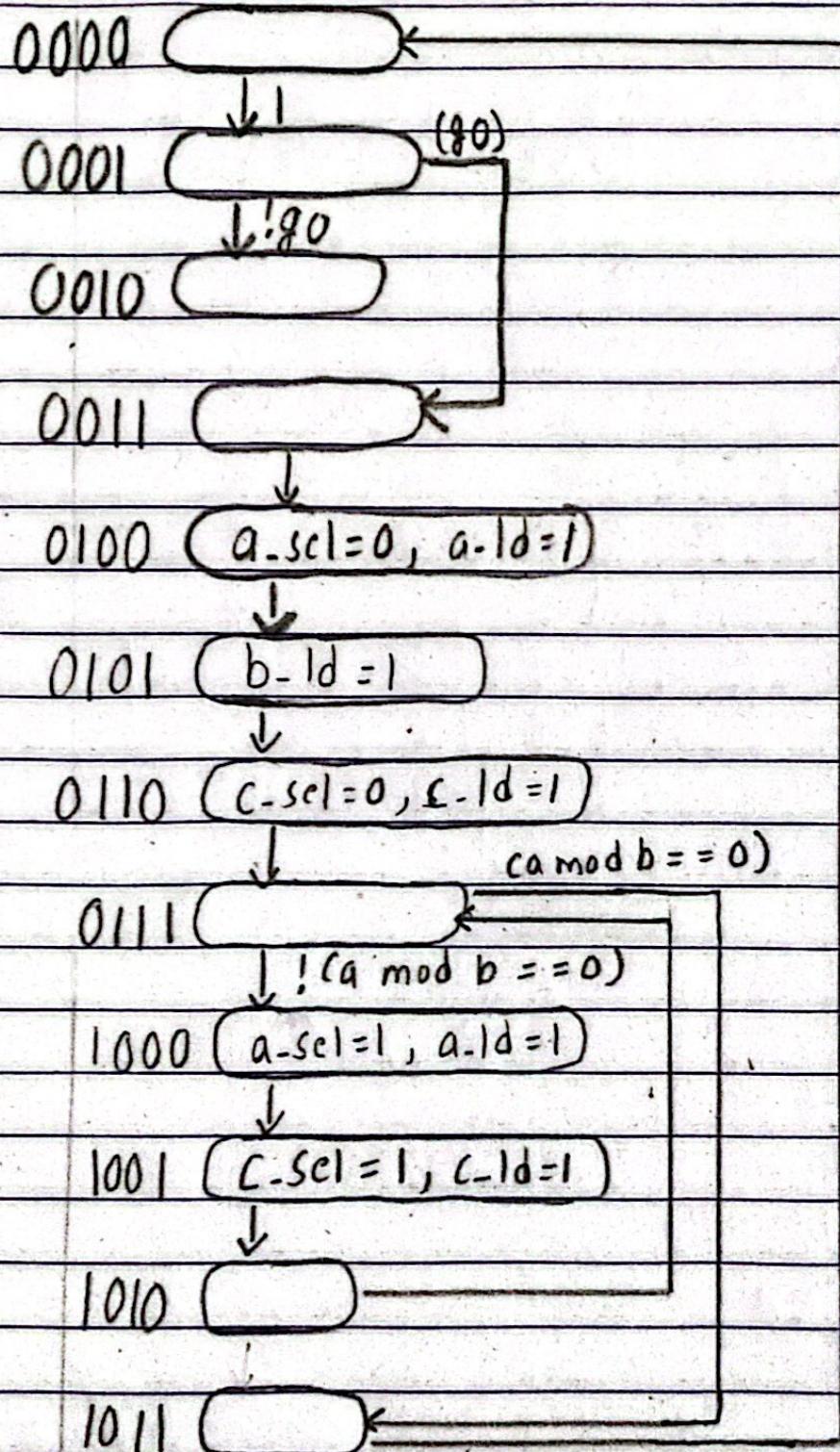
C. FSM D



C. Datapath.



E. FSMD



3 What do you mean by pipeline? Explain 5 stage pipeline. Describe features of DSP.

⇒ Pipeline is a technique where a process is divided into series of stages and each stage of process are carried out simultaneously.

5 stages of pipelining are :-

- a) Fetch :- Fetching the instruction from memory.
- b) Instruction decode :- Decoding instructions and extracting necessary
- c) Operands Fetch :- Extracting the value of operands
- d) Execution :- Executing the operation specified by the instruction.
- e) Memory Access / storage :- Storing the result into the memory or registers

DSP: Digital Signal Processor is a specialized type of microprocessor. Its features are:

- a) It may contain numerous register files, memory blocks, multiplier and other arithmetic units.
- b) It facilitates with instruction that are applicable uniquely to digital signal processing.
- c) Frequently used arithmetic function are implemented using hardware.
- d) DSPs often allow concurrent execution of functions which boost performance.
- e) It incorporates many peripherals specific to signal Processing.

4. Explain the operation of storing data in one Time programmable ROM.

Compose a memory of size $2^{K+1} \times 2^n$ using $2^k \times n$ sized memory.

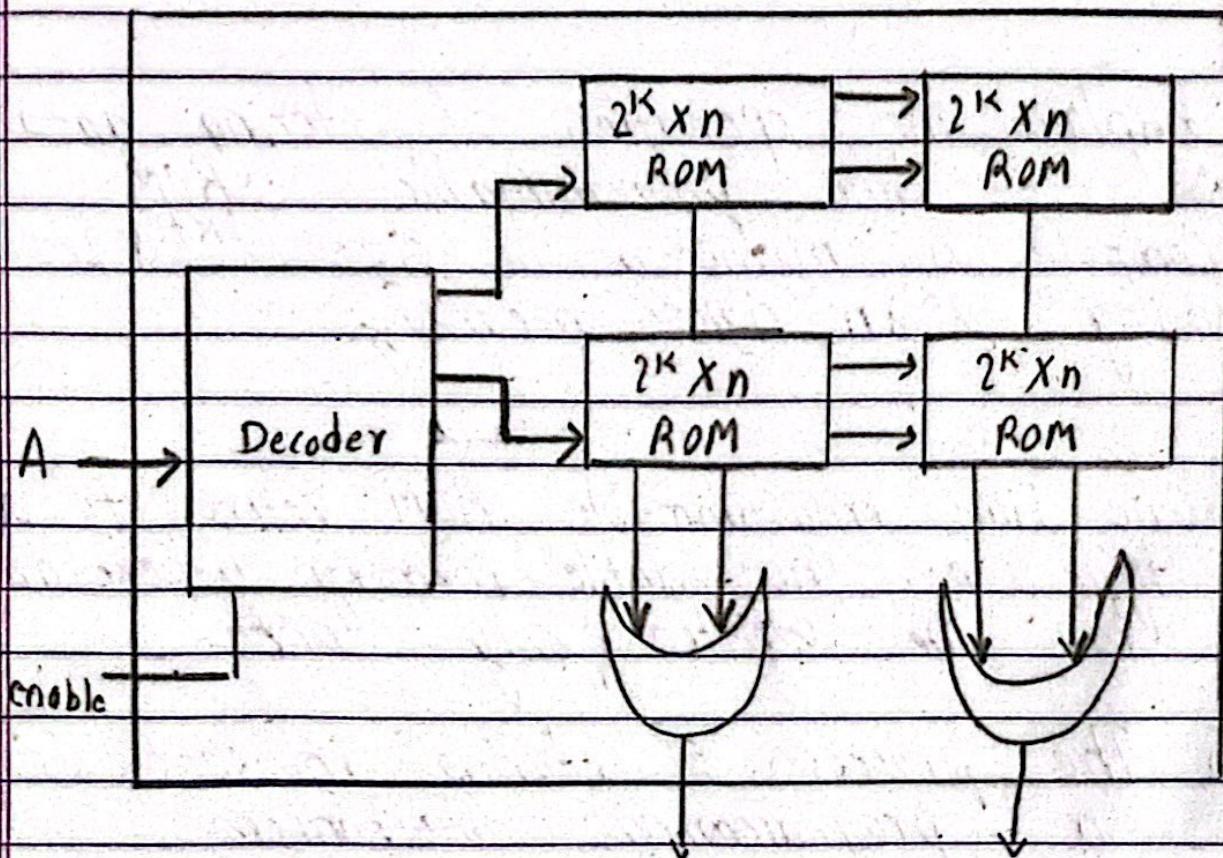
=) One-Time Programmable ROM refers to a type of non-volatile memory where data can be written only once.

↳ The process of storing involves using a programming mechanism.

↳ Programming mechanism permanently sets certain Memory cell to specific states.

↳ Programmer blows fuses by passing large current wherever a connection is not required.

↳ The blown fuses can't be reestablished, hence one time programmable.



Composing
fig: $2^{K+1} \times 2^n$ Memory using $2^K \times n$ memory

5 Explain control methods used for communication in I₂ interfacing. Describe Daisy-chain arbitration with the help of a block diagram and steps.

=> 2nd part of Question already done in 2075 Bhadra Qn° 5 b

Control methods used for communication in interfacing are:

- a) Polling : Controlling device continuously checks the status of the peripheral device to determine they are ready for communication
- b) Interrupts : After Peripheral device send signal to controlling device when they are ready for communication
- c) DMA : Direct Memory Access avoid the involvement of CPU & directly sends data from peripherals to memory.

6 Briefly explain the different states of task. Consider---process. List Coffman condition for deadlock.

⇒ First Part of Question is already done in 2076 Bhadra Qn:6.

Last Part of Question is already done in 2075 Bhadra Qn:6

Solution:

Given information is tabulated below:

Process	Entry time	Completion time
P1	0	53
P2	0	17
P3	0	68
P4	0	24

Time slice = 4ms.

Execution Sequence:

P1	P2	P3	P4												
0	4	8	12	16				32			48				64

P1	P2	P3	P4	P1	P3	P4	P1	P3	P1P3	P1P3	P1P3	P1	P3
68	69			84	88								

~~(P1 P3 P1 P3 P1 P3 P1 P3 P1 P3 P1 P3)~~

Turn around time (Completion point - Entry point) [in ms]

$$P_1 = 145 - 0 = 145 \text{ ms}$$

$$P_2 = 69 - 0 = 69 \text{ ms}$$

$$P_3 = 162 - 0 = 162 \text{ ms}$$

$$P_4 = 88 - 0 = 88 \text{ ms}$$

$$\text{Average TAT} = 116 \text{ ms}$$

Waiting time (Turn around time - Completion time)

$$P_1 = 145 - 53 = 92 \text{ ms}$$

$$P_2 = 69 - 17 = 52 \text{ ms}$$

$$P_3 = 162 - 68 = 94 \text{ ms}$$

$$P_4 = 88 - 24 = 64 \text{ ms}$$

$$\therefore \text{Average waiting time} = (92 + 52 + 94 + 64) / 4 \\ = 75.5 \text{ ms}$$