Set-2075 Bhadra

Q1. What is embedded system? Differentiate it with non-embedded systems. with suitable example. In RTOS, describe mutual exclusion through sleep & wake for task synchronization.

Ans → An embedded system is a combination of computer hardware and software, interfacing of memory and peripheral devices for a specific function.

Embedded system	Non-embedded system
1. It is a combination of special purpose hardware and embedded OS for executing specific set of applications.	1. It is a combination of generic hardware and a general purpose OS for executing a variety of applications.
2. It may or may not contain OS.	2. It contains general purpose OS.
3. Non-alterable applications.	3. Alterable applications by user.
4. Assembly language programming.	4. High level progamming language
5. One dedicated task.	5. Many dedicated task.
6. Example: Digital Camera, digital wristwatches, mp3 player, Washing machine, etc.	6. Example: Desktop PC, Laptop, dedicated server, etc.

Nancy

→ In RTOS, mutual exclusion is a technique used to synchronize access to shared resources between tasks. It is used to ensure that only one task can access a shared resource at a time, preventing conflicts or data corruption.

Task synchronization by "sleep & wake func"

When a task needs to access a shared resource, it can call the sleep func^n to suspend its execution and wait for the resource to become available. Once the resource is available, the task can call wake func^n to resume its execution and access the resource.

Example:

```
void task1(void){
  while(1){
  sleep(); // suspend task execution
    // Access shared resource
    wake(); // Resume task execution
  }}

void task2(void){
{ while(1){
    sleep(); // suspend task execution
    // Access shared resource
    wake(); // Resume task execution
  }}
```

Page:
Date: / /

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

Ans Optimization is a process of improving efficiency of a program in time (speed) or space (size). Embedded system optimization and trouble-shooting are carefully managed processes that can correct defects, improve system performance, enhance reliability or add new functionality.

Parameters:

1. Original program

Optimizing → Replace subtraction operation with modulo operation in order to speed up program

2. FSMD

Optimizing → Areas of possible improvements
- merge states
- separate states
- scheduling

3. FSM

optimizing → - State encoding
- State minimization

4. Datapath

Optimizing → - sharing of functional units
- Multifunctional units

Nancy

Q (3). Define datapath and controller of a general purpose processor. Explain AsIP with its types.

Ans:

A general purpose processor is a type of CPU that can execute wide range of executions and is designed to perform a variety of tasks. It is a key component of a computer system and is responsible for executing instructions and controlling the flow of data within the system.

A controller of a general purpose processor is a part of CPU that fetches instructions from memory, decodes them, and generates the control signals needed to execute the instructions. It is responsible for coordinating the operations of the datapath & ensuring that the instructions are executed in the correct order.

An AsIP is a type of processor that is designed for a specific application or class of applications.

Types:

1. Custom AsIP: These are designed from scratch for a specific application.

2. Configurable AsIP: These can be customized / configured to meet the specific needs of an application.

Nancy

3. Conf Domain-specific ASIP: These are designed for a specific domain or class of applications.

4. Hybrid ASIP: These combine elements of custom, configurable, and domain-specific ASIPs.

Q.4. Define write ability & storage permanence of memory. Design a ROM to store following info:

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

Ans

Write ability refers to the ability of a memory device to store new data. It determines whether or not the device can be used to store new information.

Storage performance refers to the ability of a memory device to retain data over a long time period of time, even when the device is not being used or is powered off.

Nancy

Designing of ROM:

An 8X4 ROM is a type of digital storage device that has 8 address lines & 4 data lines. This means that it can store up to 256 (2^8) different values, each of which is 4 bits wide. We need 8-bit address & the value stored at that address would then be output on the data lines. Eg:

Address = 00010010,

value stored would be output on the data lines.

Here, ROM can store up to 256 different 4-bit values, for a total of (256 X 4) 1024 bits. The number of words in an 8X4 ROM is equal to the number of values it can store, which in this case is 256

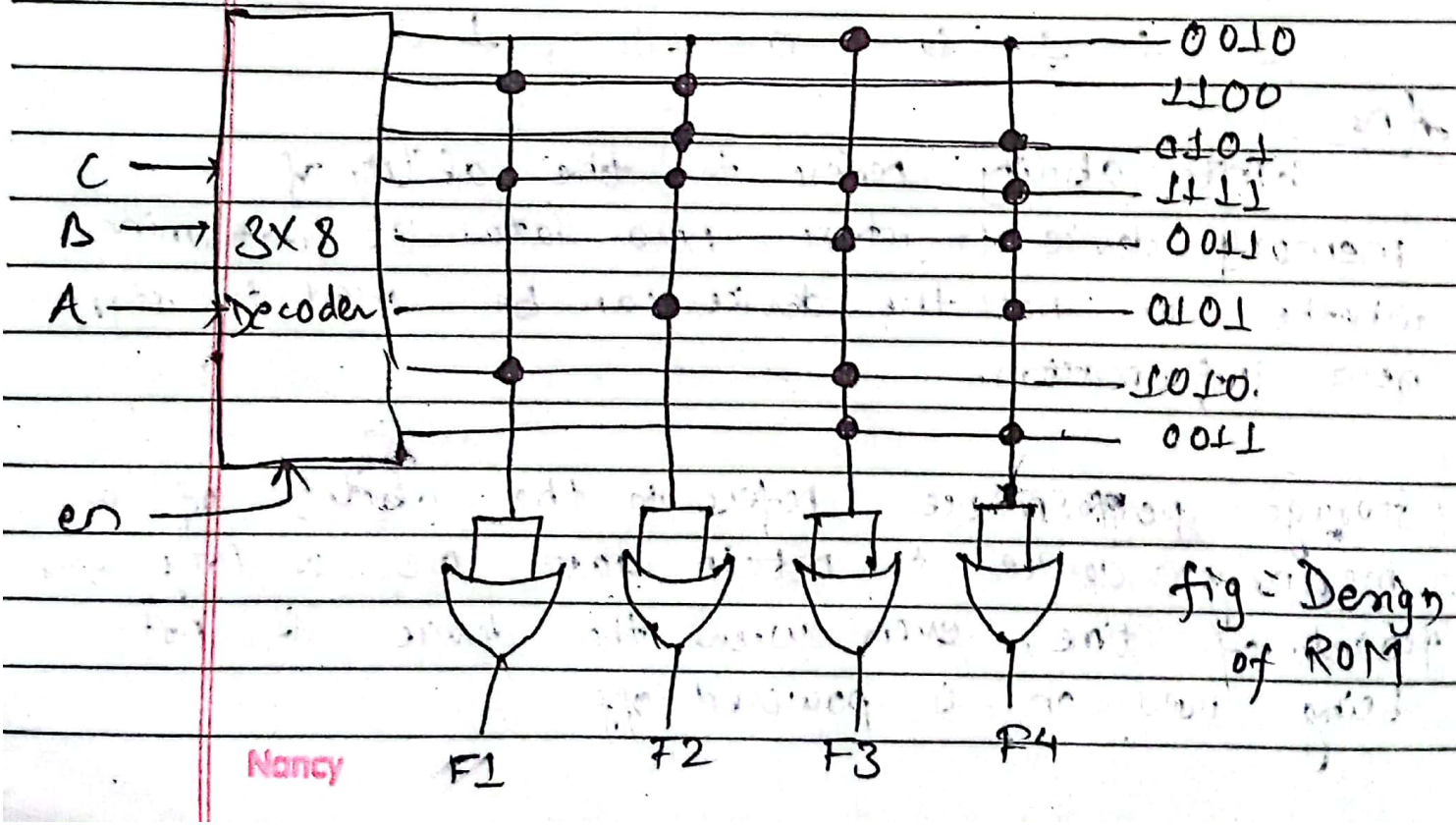


fig: Design of ROM

Nancy

Page:
Date: / /

Q(5 What is the interrupt? Explain summary of flow of actions of interrupt driven using fixed ISR location.

→ An interrupt is a signal to the processor emitted by hardware or software indicating an event that needs immediate attention.

Interrupt-driven I/O is a method of transferring data between a computer and external device, such as keyboard, mouse or printer.

Flow of actions:

1. The external device initiates the transfer of data by seriding an interrupt ~~service~~ request (IRQ) to the processor.

2. The processor stops executing the current instruction and saves its current state (e.g. the value of the PC, register values, etc.).

3. The processor jumps to a fixed location in memory, known as interrupt service routine, where it begins executing the code to handle the interrupt.

4. The ISR code determines the source of the interrupt and takes the appropriate action, such as reading the data from the device

Nancy

or writing data to the device.

5. When the ISR code has completed its task, it sends an interrupt acknowledge signal to the device to indicate that interrupt have been serviced.

6. The processor restores its saved state and resumes execution of the interrupted instruction.

Q no. 6. Explain the conditions of favoring deadlock situation. 3 processes P1, P2, P3 with estimated completion time 5, 8, 7 ms resp. enters the ready queue together. Calculate WT, TAT for each process & calculate AWT and ATAT using Round Robin Preemptive scheduling algorithms with time slice of 2 ms.

Ans →

4 favoring conditions:

1. Mutual Exclusion: Non-shareable resources.
2. Hold and wait: A process is holding at least one resource at a time and is waiting to acquire other resources held by some other process.
3. No preemption: The resource can be released by a process voluntarily i.e.

Nancy

Page:
Date: / /

after execution of the process.

4. Circular wait: A set of processes are waiting for each other in a circular fashion.

Gantt chart:

A	B	C	A	B	C	A	B	C	B	C	
0	2	4	6	8	10	12	13	15	17	19	20

Job	Arrival Time	Burst Time	Finish Time	Turnaround Time	Waiting Time
A	0	5	13	13	8
B	0	8	19	19	11
C	0	7	20	20	13
Average				52/3 = 17.333	32/3 = 10.667

2075 Bhadra

Page:
Date: / /

Q. (1.) Define embedded system and classify based on generations. What is task synchronization & explain task synchronization using 'Busy/Wait'.

→ Embedded system ✓

→ Different generations of embedded system:

1. 1st Generation (1970s-1980s):
   → Simple microcontrollers ~~and~~ were used for basic tasks such as controlling appliances and automating industrial process

2. 2nd Generation (1980s-1990s):
   → Advanced microcontrollers & microprocessors were used for wider range of applications such as automotive systems, medical devices and military systems.

3. 3rd Generation (1990s-2000s):
   → Powerful microprocessors had advanced features such as RTOS, internet connectivity. Applications are industrial control, telecoms & consumer electronics.

4. 4th Generation (2000s - present):
   → High-speed networking & multimedia capabilities
   → Multimedia devices, smart home systems, and self driving cars.

Nancy

Page :
Date : / /

→ Task synchronization is the process of coordinating the execution of tasks in a concurrent or parallel system, such as computer with multiple processors or a system with multiple threads.

'Busy-Wait'

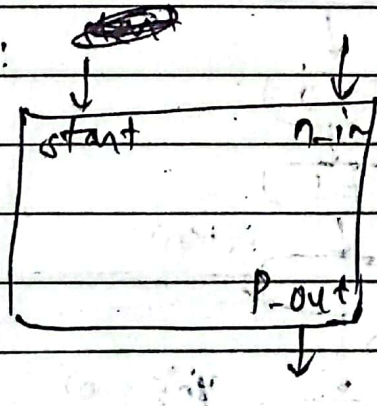
→ A task waits for another task to complete by continuously checking the status of other task. This can be implemented using a loop that checks the status of other task & waits until it is completed,

```
while (task_B status != completed)
{
  // check status of task B
}
// continue execution of task A
```

Q. no. 2 Design a custom single-purpose processor that display 1/0 if the input integer is prime or not showing all the steps:

Ans:

a. Black Box:

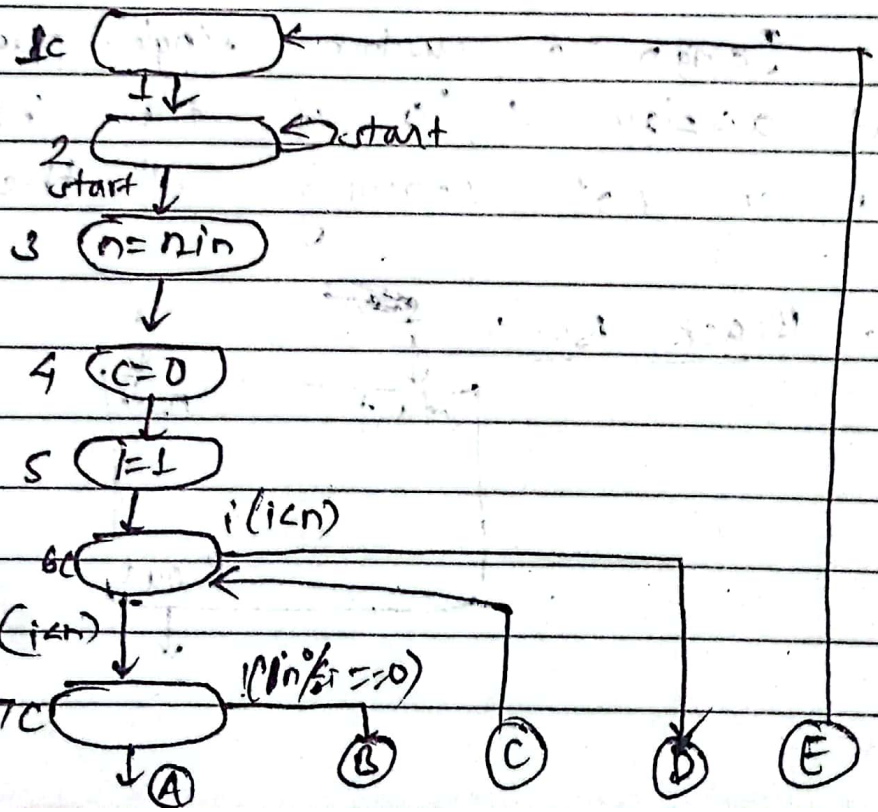


Nancy

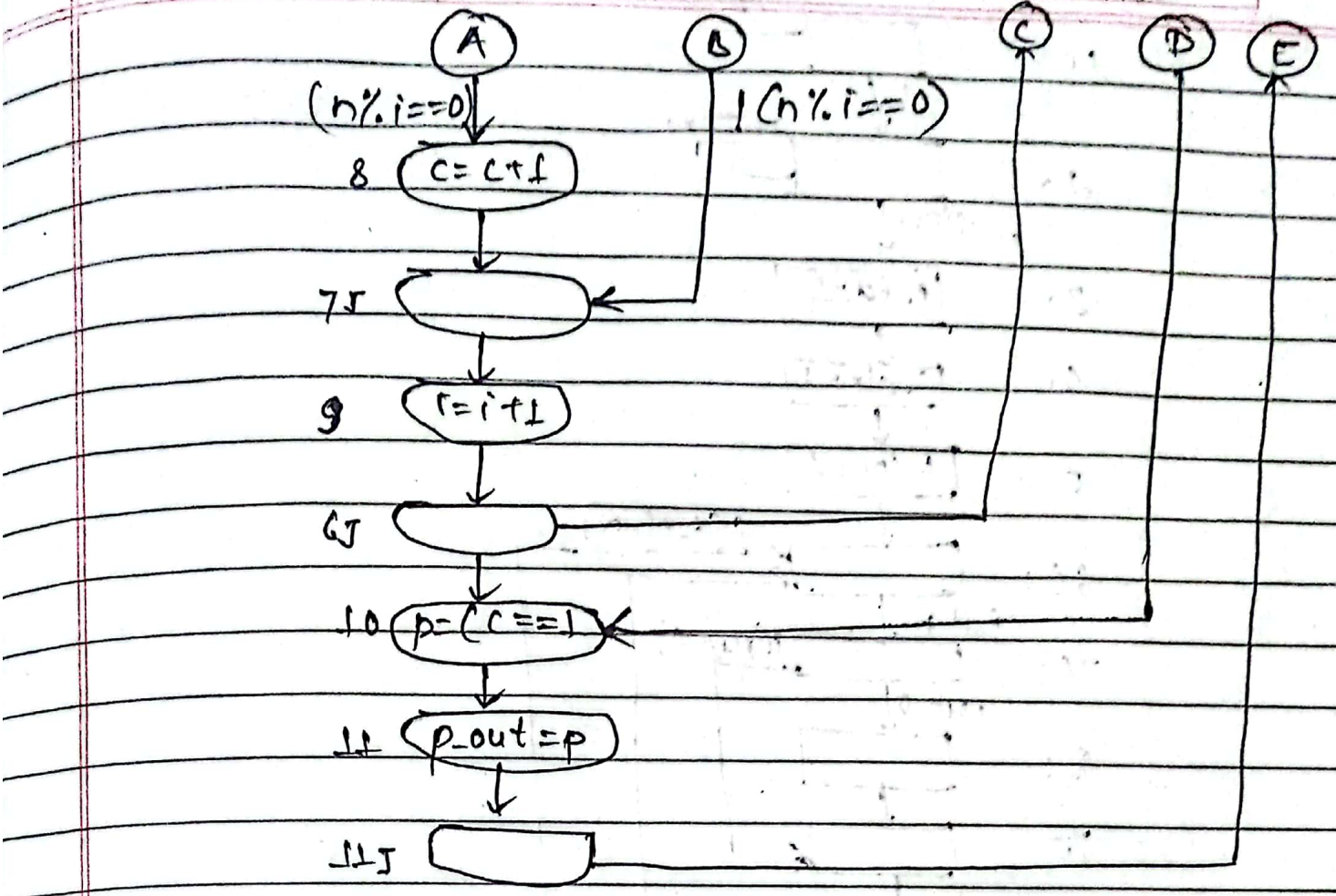
b. Functionality Code:

```
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;
}
```

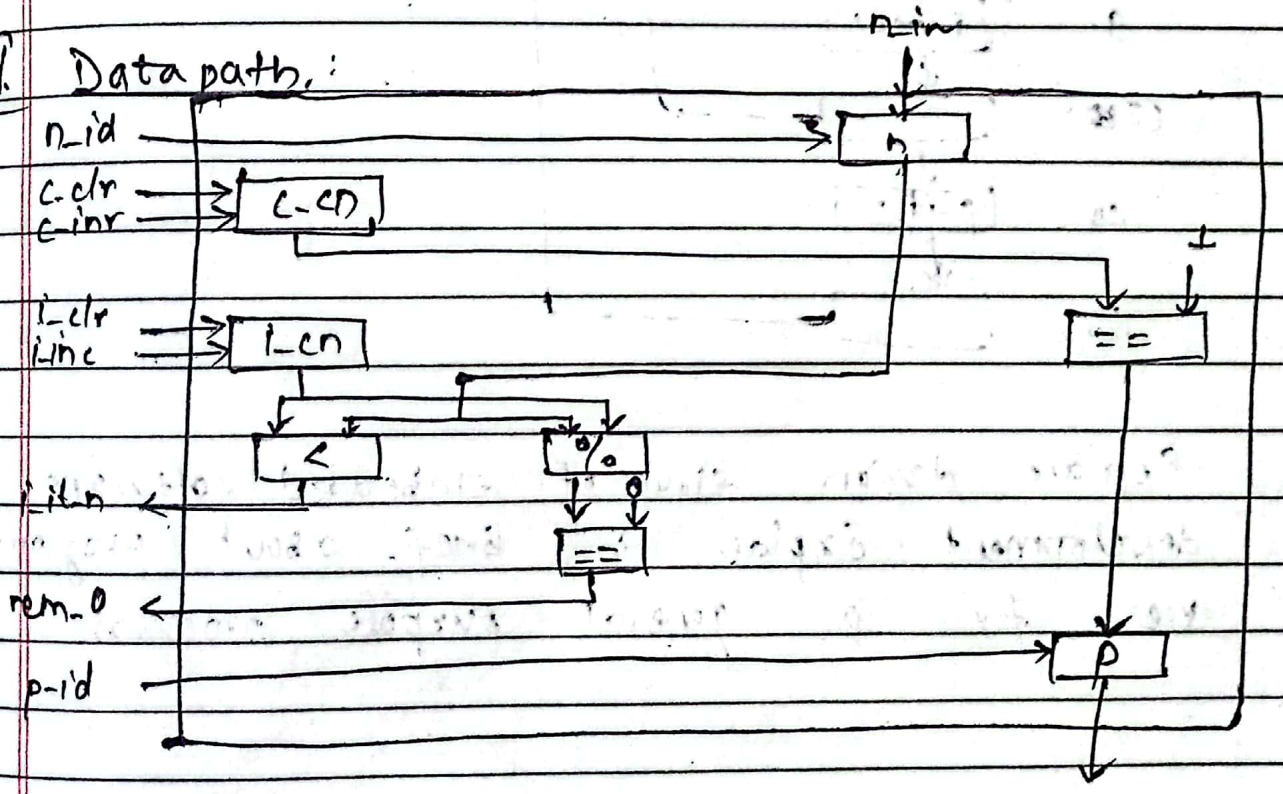
c. FSMD:



Nancy

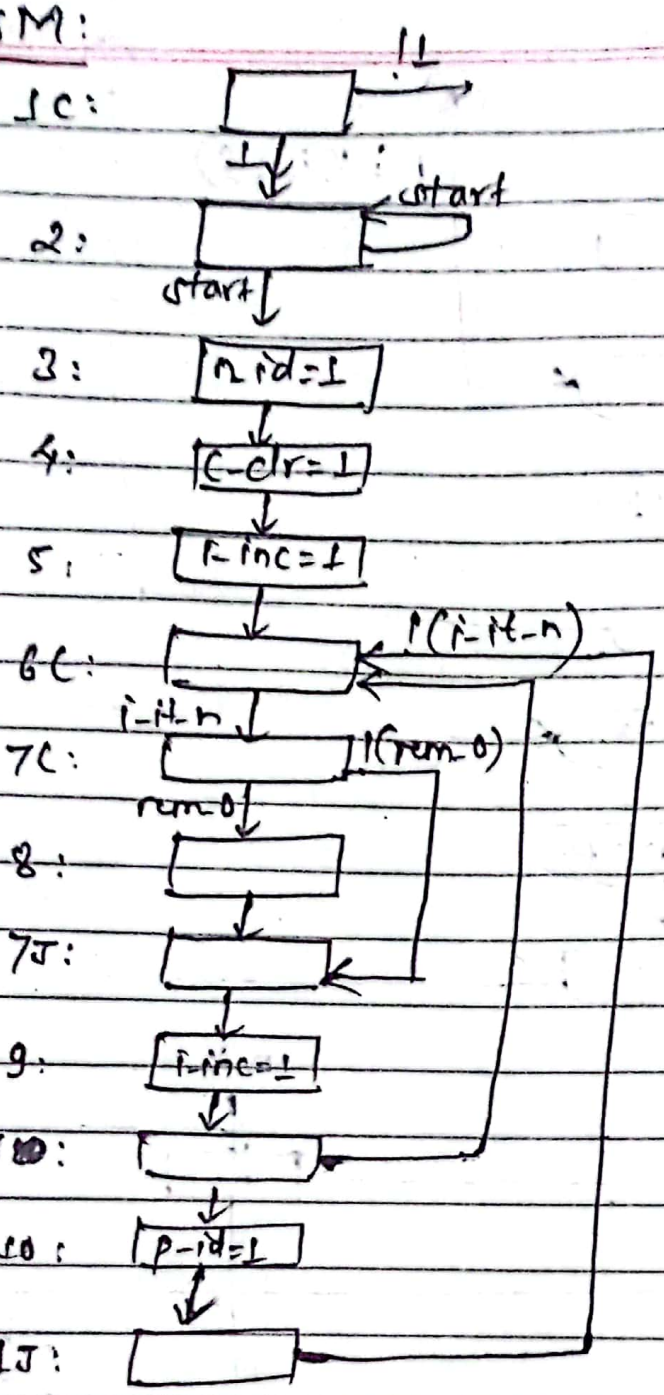


d. Data path:



Nancy

(e) FSM:



Q. no (3) Explain design flow of embedded software development. Explain in brief about programmer's view for a general purpose processor.

Page:
Date: / /

Ans Design-flow:

(i) Requirements gathering & analysis:
- Software & hardware components
- Performance requirements
- Constraints / Limitations

(ii) System design:
- Overall architecture : components & their interactions

(iii) Hardware design:
- Designing & implementing hardware components
- Processor, memory, peripherals interaction.

(iv) Software design:
- Designing & implementing software components
- OS, drivers, application software interaction.

(v) Testing & debugging:
- Ensuring functions work correctly.

(vi) Integration & deployment:
- Integration of hardware & software components
- Deploying it in target environment.

→ A programmer's view for a general purpose processor is typically focused on writing software programs that can be executed on

Nancy

the processor. This involves writing code in high level programming language, C/C++, and using tools like compilers & debuggers. Programmers needs to consider issues such as memory management, performance optimization and power consumption when writing code for a general purpose processor.

Q. (4.) Define & categories of memory; write ability & storage ~~performance~~ permanence with their different levels. Explain replacement algorithm used in cache memory.

→

Write ability: → Ability of memory device to store new data.

Levels:

1. ROM: Memory is permanently written with data & can't be modified by user.

2. PROM: Memory can be written with data once using a special programming device; But can't be modified by user once programmed.

3. ~~RAM~~ RW Memory: Memory can be written by both read & user. Eg: RAM, flash memory.

Nancy

Page:
Date: / /

Storage permanence: Ability of a memory device to retain stored data when power is removed.

Levels:

1. Volatile memory: Memory requires a constant power supply to retain stored data. When power is removed, stored data is lost. Eg: DRAM, SRAM.

2. Semi-permanent memory: Memory retains stored data for a certain ~~permanent~~ period of time after power is removed, but eventually loses the stored data. Examples: battery backed RAM, ferroelectric RAM.

3. Non-volatile memory: Memory retains stored data indefinitely, even when power is removed. Eg, ROM, PROM, Flash memory.

→ Cache memory is a type of high speed memory that is used to store frequently accessed data from main memory. It uses replacement algorithm to determine which data should be removed from the cache when new data needs to be stored.

Replacement algorithms:

1. LRU: It removes the data that has not been accessed for the longest period of time.

Nancy

(ii) MRU: It removes the data that has been accessed most recently.

(iii) LFU: It removes data that has been accessed the fewest no. of times.

Q. (5) What is interrupts? How interrupt needed in digital devices? Write a summary of flow of actions for interrupts devices I/O using ISR location.

→ Interrupts

→ In terrupts are needed in digital services because they allow the device to respond to external events in timely manner.

→ flow of actions

Q no. (6) Explain process life cycle. 3 process (P_1, P_2, P_3) with estimated completion time 8, 6, 10 ms & priority 0, 3, 2 (0 - high, 3 - low) enters a ready queue in P_1, P_2, P_3. Now a process P_4 with estimated completion 6ms time with priority 1 enters the ready queue after 5 ms of execution of P_1. Calculate WT, TAT, AWT, TAT.

Nancy

Ans

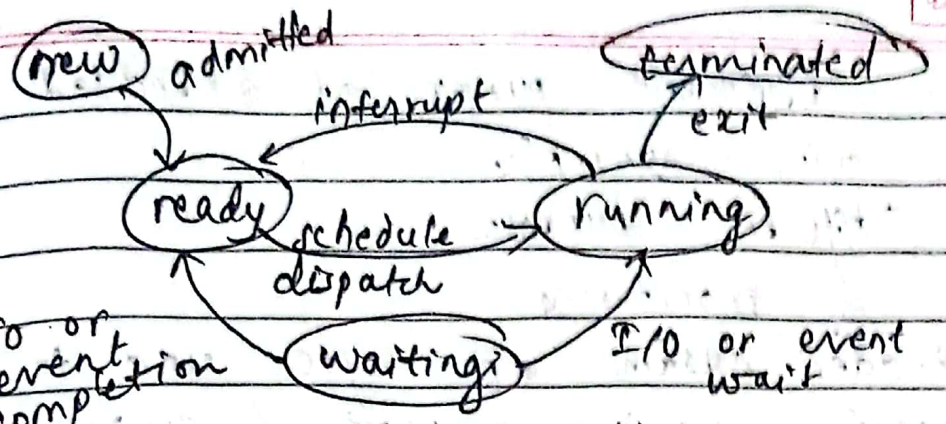


fig: process state diagram

A process life cycle refers to the sequence of stages that a process goes through, from its creation to its completion. Process state diagram is a visual representation of different states that a process can be during its life cycle.

Processes:

1. New
2. Ready
3. Running
4. Waiting
5. Terminated

So in:

Gantt chart

P1	P4	P3	P2
0	8	14	24	30

Job	Arrival Time	Burst Time	Finish Time	Terminated Time	Waiting time
P1	0	8	8	8	0
P2	0	6	30	30	24
P3	0	10	24	24	14
P4	5	6	14	9	3
			Avg:	71/4	41/4

2077 Chaitra

Page:
Date: / /

Q.1. Explain different purpose of embedded system with examples.

→ Different purposes:

1. Control & monitoring:
Embedded systems are often used to control & monitor various processes & devices, such as motors, sensors, and actuators. Ex: They are used to control operation of the machine, monitor its performance, and detect and diagnose any problems in industries.

2. Data acquisition & monitoring:
Embedded systems are often used to acquire & process data from various source, such as sensors, instruments or networks. Ex: They are used to collect data from temp., pressure, and humidity sensors and to process and transmit that data to a central location for analysis in weather station.

1. Real-time processing: Embedded systems are used to perform real-time processing, which requires fast response times & high reliability. Ex: It is used to control flight systems, navigate aircrafts, and detect & respond to any emergencies in real time in aircraft.

Nancy

4. Communication & networking: Embedded systems are used to faciliate communication & networking between devices & systems. Example: It is used in mobile phones to support wireless communication, data transfer, internet connectivity.

Q. (2.) What are optimization opportunities in single-purpose processors (SPP)

→ Single purpose processors (SPP), also known as application-specific processors, are microprocessors that are designed and optimized for a specific application or task.

Optimization opportunities

1. Instruction set optimization: (custom & reduced)
SPP can be optimized for specific instructions & operations required by the application to improve efficiency & power

2. Memory hierarchy optimization:
SPP can be optimized to reduce size and organization of cache and main memory.

3. Pipeline optimization:
SSP can be optimized for specific pipeline stages & latencies required by application.

4. Power management optimization:

Nancy

DSP can be optimized as dynamic voltages and frequency scaling, power gating, and sleep modes.

Q. 3. Explain the architecture of general purpose processor. What are the criteria to select this processor?

→ Architecture of general purpose processor:

1. ALU: It is responsible for executing arithmetic and logical operations. It receives operands from registers or memory and produces a result, which is stored in a register or memory location.

2. CU: It is responsible for fetching instructions from memory, decoding them and issuing control signals to other components of the processor.

3. Registers: These are small, high-speed memory locations within the processor that are used to store operands, intermediate results, and other data.

4. Cache memory: It is a type of high speed memory that is used to store frequently accessed data and instructions.

5. Bus interface: It is responsible for transferring

Nancy

data and instructions between the processor and main memory, as well as between the processor and other devices on the system bus.

6. Instruction pipeline: It is a series of stages that instructions pass through as they are executed.

→ Criteria to select the processor:

1. Performance:
   → Processor should have sufficient processing power to handle the tasks required by the system.

2. Memory:
   → Processor should have sufficient memory to store program code and data required by the system.

3. I/O capabilities:
   → Processor should have necessary I/O capabilities to interface with the peripherals and devices required by the system.

4. Power consumption:
   → Processor should have low power consumption to extend the battery life of the system or to reduce heat generation.

and many more.

Page :
Date : / /

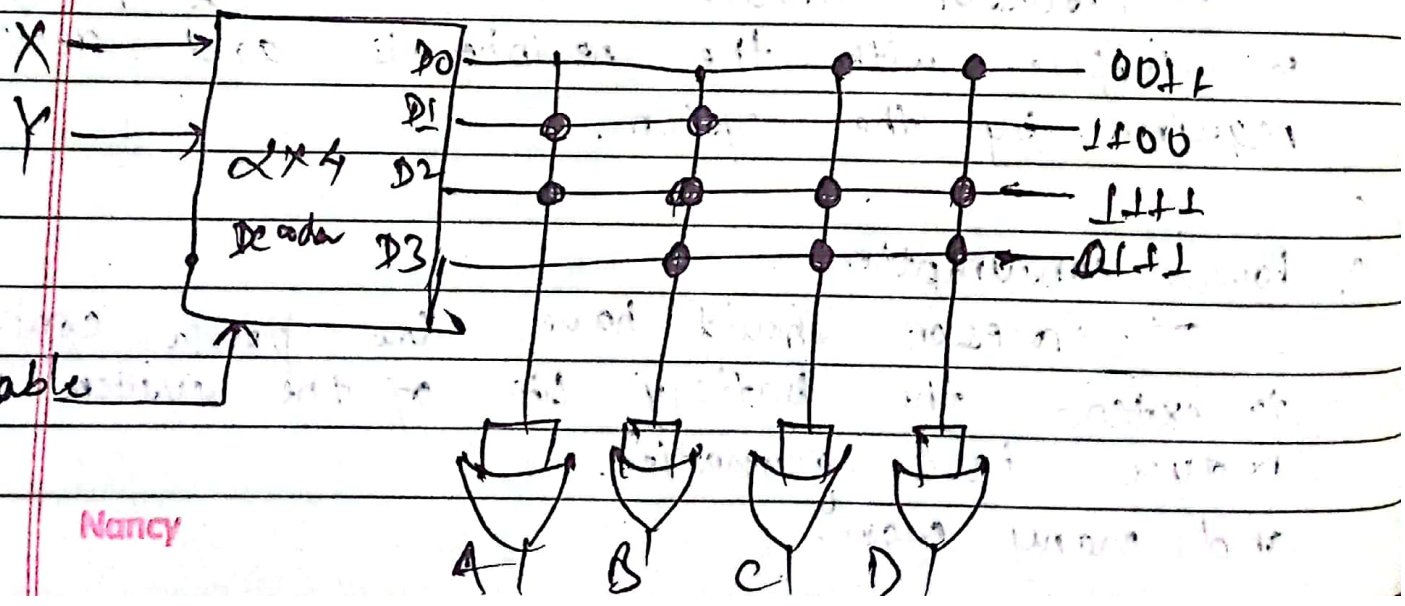
Q. 4. Design the internal architecture of 4x4 ROM. Explain memory write ability and storage permanance with suitable example.

→ Following steps

1. Construct truth table for inputs & outputs.
2. Based on total no: of address in the ROM and the length of their content, decide the decoder as well as OR gates to be used.
3. Now, program the intersection between the & lines, as per the truth table, so that the output of ROM is in accordance with the truth table.

example:

Inputs		Outputs			
X	Y	A	B	C	D
0	0	0	0	1	1
0	1	1	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1



Nancy

→ Write ability & storage permanence ✓

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

→ Interfacing: The process of connecting 2 or more devices or systems in order to exchange information or data.

→ Needs of interfacing:

1. Data exchange: Interfacing allows devices & systems to exchange data with each other.

2. Control & communication: It allows to control & communicate and collaborate with each other, enabling to coordinate actions and respond to changing conditions.

3. Extension & expansion:
   → Expand and extend capabilities, enabling to interact with a wider range of peripherals & devices.

4. Compatibility & interoperability
   → Work together seamlessly, regardless of their specific technologies and protocols.

Nancy

Page :
Date : / /

Priority Arbitration:

It is a method of resolving conflicts that arise when multiple devices or processes try to access a shared resource at the same time.

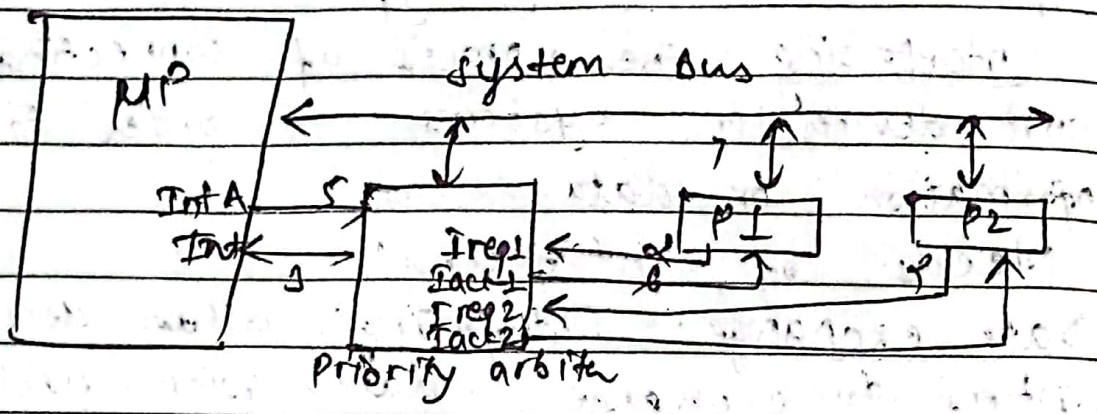


fig: Priority arbitration

Each device has a priority level, with higher priority levels indicated by a lower position in the diagram. When a device wants to access the resource, it sends a request to the priority arbiter, which grants access to the device with highest priority level that has made request.

Types:

1. Static priority arbitration
2. Dynamic priority arbitration

Q. 6. Compare task, process and threads. 5 process with : IDs P1, P2, P3 with estimated completion times 6, 8, 2 ms resp. enters the ready, queue together. Process P4 with estimated

Nancy

execution completion time 4ms. enters the ready queue after 1ms. Calculate WT & AWT for each process and AWT & TAT in non-preemptive shortest job first scheduling. Describe basic func^{n} of real time kernel.

→ Difference betⁿ Def? Exec^{n}, Resource Usage, Task Schedule

- A unit of work performed by a computer system
- Executed sequentially / concurrently
- Use shared resources such as memory & files
- Scheduled by system / user

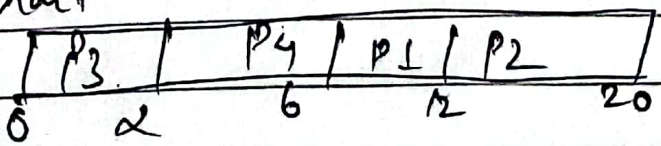
Process

- An instance of a running program / application
- Executed sequentially / concurrently
- Have own memory & file space
- Scheduled by system / user

Threads

- A unit of execution within a process
- Executed concurrently within a process
- Share resource within a process
- Scheduled by system / process.

Gantt chart



Nancy

Job	Arrival Time	Burst Time	Finish Time	Turnaround Time	Waiting Time
P1	0	6	12	12	6
P2	0	8	20	20	12
P3	0	2	2	2	0
P4	1	4	6	5	1
	Average			39/4 = 9.75	19/4 = 4.75

A real-time Kernel is a type of OS Kernel that is designed to provide deterministic response times to events. The main function is to manage the resources of a computer system in such a way as to ensure that time-critical tasks are completed within specified time constraints.

Nancy

2078 Poush

Page:
Date: / /

Q. (1) Define embedded system. Explain unique characteristics of embedded system.

→ Embedded System: ✓

Characteristics:

1. Specialized func^n: To perform specific task.
2. Real-time constraints: Respond to inputs and events within a specific time frame.
3. Limited resources: Memory & processing power.
4. Embedded within a larger system: car / washing machine
5. Customization: specific applications & environments
6. Variety of technologies: Microcontrollers, microprocessors, FPGAs.

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

→ ✓

Nancy

Q.3 Explain the basic operations carried by controller of a general purpose processor. Describe different forms of ASIP.

→ Basic operations:

1. Instruction fetch: The controller retrieves instruction from memory & loads them into the processor.

2. Instruction decode: To determine operations to be performed.

3. Execution: Perform required operations.

4. Memory Access: To read and write data as part of execution.

5. Interrupt handling: To respond to external event of peripherals.

→ ASIP is a type of processor that is customized for a specific application or set of applications.

Different forms:

1. CISP: Custom instruction set tailored to specific requirements of application.

Nancy

2. Reconfigurable processors: Hardware or software techniques to achieve reconfigurability to perform different tasks.

3. Hybrid processors: Combined elements of general-purpose and application-specific processors.

4. Domain-specific processors: To work on specific domain, audio processing or image processing.

a. 4. Define write ability with its different levels. Design a ROM to store the following info:

A	B	C	F1	F2	F3	F4
0	0	0	0	1	1	1
0	0	1	0	0	1	1
0	1	0	0	1	1	0
0	1	1	0	0	0	0
1	0	0	1	0	1	0
1	0	1	1	1	1	1
1	1	0	1	0	0	0
1	1	1	0	1	1	1

→ Write ability ✓

- Designing a ROM : steps ✓

- 8X4 ROM, 8 address lines, 4 data lines
- It can store 256 different values

Nancy

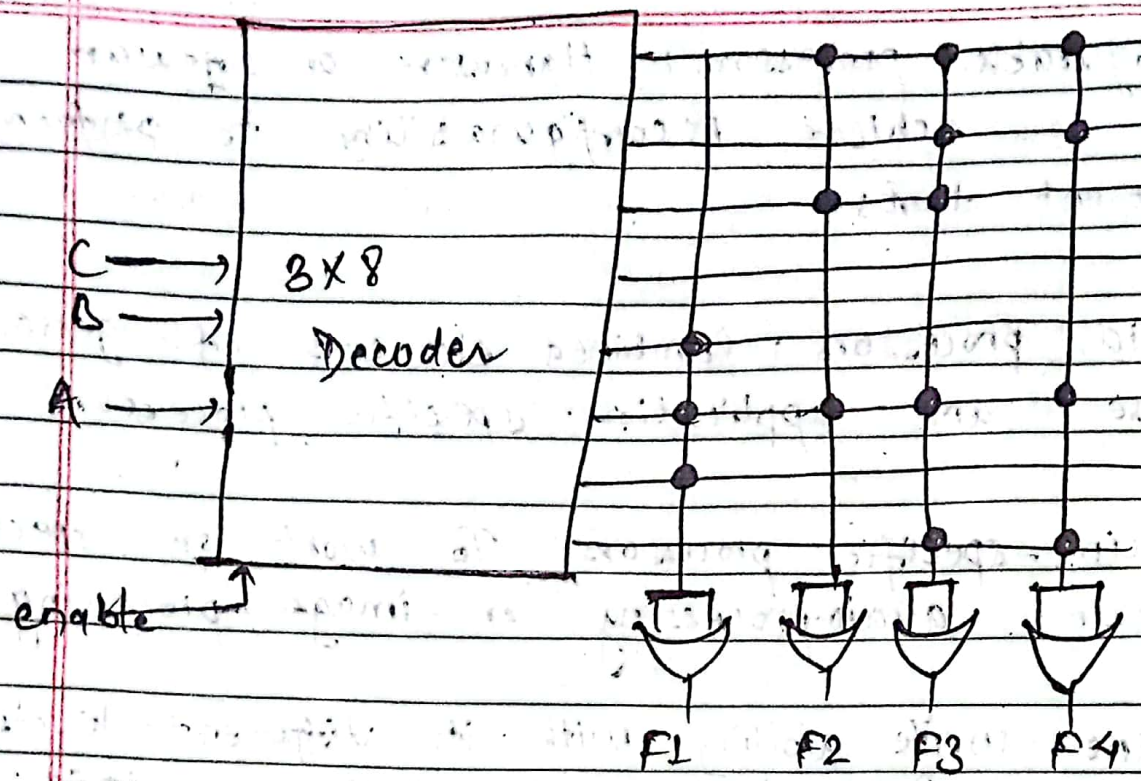


fig: Design of ROM

Q. (5.) Define interfacing and explain needs of interfacing. Explain Daisy chain arbitration with the help of block diagram & steps.

→ Interfacing ✓

Needs of interfacing ✓

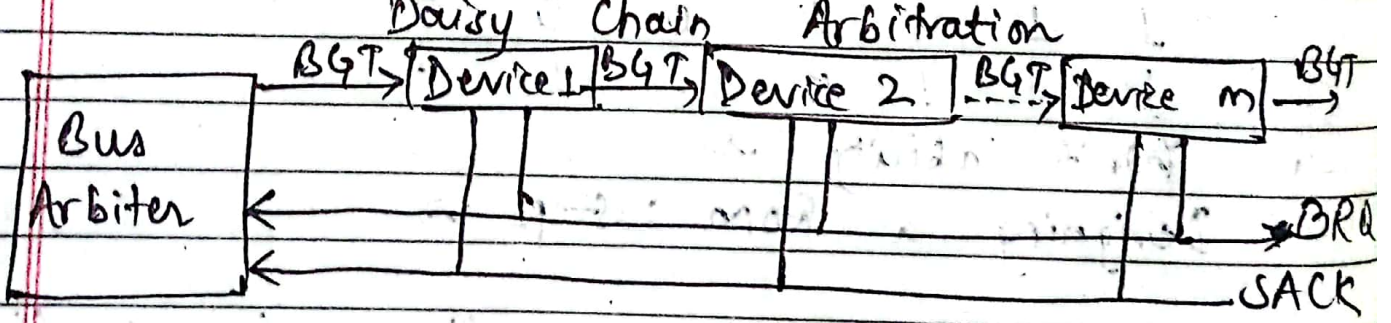


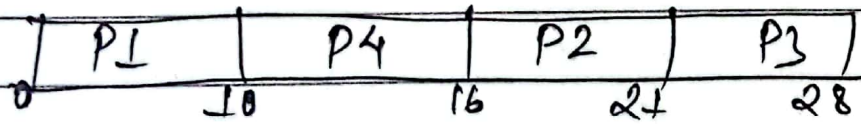
fig: Daisy chained bus arbitration

Nancy

In this diagram, devices are connected in a daisy chain configuration, with each device connected to the bus through a request/grant line. The arbiter is responsible for granting access to the shared resource based on request signals it receives from the devices. The shared resource is accessed through a shared memory or other shared resource component.

Q. no. 6 3 processes with process IDs P1, P2, P3 with estimated completion time 10, 5, 7 ms and priorities 0, 2, 3 resp., enters ready queue together in order P1, P2, P3. Now a process P4 with estimated completion time 6 ms and priority 1 enters the ready queue after 6 ms of execution of P1. Calculate WT & TAT for each process and average WT & TAT. Assume there is no I/O waiting time for processes and priority-based non-preemptive scheduling.

→ Gantt chart



P.T.O.

Nancy

Job	(AT) Arrival Time	(BT) Burst Time	(FT) Finish Time	(FT-AT) Turn around Time	FT-(AT+BT) Waiting Time
P1	0	10	10	10	0
P2	0	5	21	21	16
P3	0	7	28	28	21
P4	6	6	16	10	4
	Average			69/4	41/4

Nancy

2079 ~~Ashwin~~ Jestha - Regular

Page:
Date: / /

Q. 1. Define & explain embedded system. Describe its various application.

→

Embedded system ✓

Applications:

1. Automotive system: Engine management, navigation and entertainment.

2. Industrial control system: Manufacturing, quality control and energy management.

3. Medical device: Heart monitors, insulin pumps, and imaging systems.

4. Consumer electronics: smartphones, tablets, and home appliances.

5. Military & Aerospace systems: Navigation, communication and weapons systems.

6. Networking: Routers, switches, hubs to control and manage the flow of data.

7. Transportation: Trains, planes, buses to control various functions such as navigation, communication, and safety systems.

Nancy

Page:
Date: / /

Q. (2.) What is optimization? Explain in detail about optimization of custom single purpose processor with a suitable example.

→ Optimization ✓

Optimization of custom single purpose processor

- Reducing the no. of cycles required to execute a task. Eg: matrix multiplication (vectorization in ML).
- Minimizing the no. of instruction needed to perform a task. Eg: image processing tasks; scaling and rotating images (Data Augmentation in ML) → library ~~methods~~ functions.
- Reduced amount of hardware required to implement the processor. Eg: arithmetic operations can be optimized by using fewer hardware elements such as bit-level operations than word level operations.

Q. (3.) Explain concept of pipelining. Differentiate super scalar and VLIW architecture.

→

Pipelining is a technique used in computer architecture to increase the performance of a processor by executing multiple instructions concurrently. It does this by dividing the execution of instructions into multiple stages, each of which can be executed concurrently.

Nancy

Page:
Date: / /

Differences between super scalar and VLIW architecture:

1. Instruction parallelism, 2. hardware complexity, 3. programming complexity, 4. performance potential, 5. power consumption

	Super scalar	VLIW
1.	Dynamically determined based on dependencies	Explicitly specified
2.	High	Low
3.	Low	High
4.	High	High
5.	High	Low

Q. no: 4 What are the basic techniques for cache mapping? How direct mappings differ from fully associative mappings?

→

Cache mapping refers to the method used to determine which memory locations are stored in the cache and how they are associated with the main memory.

Basic techniques

1. Direct mapping
2. fully associative mapping
3. Set Associative mapping

Nancy

Differences between :

Direct Mapping

- Fixed size cache blocks with specific location in a memory.
- Simple to implement
- Minimal hardware
- High cache miss rates

Fully associative Mapping

- Any memory location to be stored in any block in the cache
- Complex implementation
- More hardware
- Less cache miss rates

Q. 5. How interrupts interface with μP? Explain it. Briefly explain operation of peripheral to memory transfer with DMA.

→

In order to interface with a μP, interrupts use a dedicated hardware line or lines that are connected to the processor. When an interrupt event occurs, the interrupt hardware sends a signal to the processor over the dedicated interrupt line, which causes the processor to stop executing its current instructions & execute the interrupt handler. Interrupt handler is a

Nancy

special routine that is designed to handle the interrupt event and may involve reading/writing data to I/O devices, updating system times or performing other tasks as needed.

→ Peripheral to memory transfer with DMA.

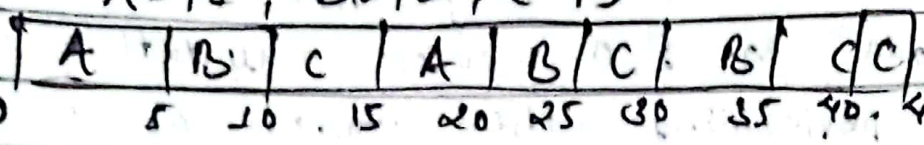
DMA is connected to both peripheral device and system memory, and is able to access both directly, so is responsible for data transfer. When the processor wants to transfer data from a peripheral device to memory, it sends request to the DMA controller. DMA fetches data from peripheral and stores it in the designated memory location, without involving the processor. When the data transfer is complete, DMA sends a signal to the processor. When DMA performs data transfer without involving processor, it is bus mastering.

Q. 6. 3 process P1, P2, P3 arrive at time 0. Their total execution time is 10ms, 15ms and 20ms resp. They spent first 20% of their execution time in doing I/O, 60% in CPU processing & 20% again doing I/O. For what % of time was CPU free? Use Round-robin algorithm with time quantum 5ms.

Nancy

Gantt chart

A = P1, B = P2, C = P3



Job	Arrival Time	Burst Time	Finish Time	Turn around Time	Waiting Time
A	0	10	20	20	10
B	0	15	35	35	20
C	0	20	45	45	25
	Average →			100/3	55/3

For each process, CPU will be busy for 60% of the process's total execution time, 6 ms, 10 ms & 20 ms for processes P1, P2, P3 resp. The CPU will also be busy for the time quantum of 5 ms for each process switch.

In total, CPU will be busy for

$6 + 10 + 12 + (3 \times 5) = 43$ ms

Thus, CPU will be free for a total of

$100 - 43 = 57\%$ of time.

Nancy