4. RTOS (Real-Time operating system:)

- RTOS is a multitasking operating system intended for real time application.
- It is a special-purpose as used in computers that has strict time contraints for any job to be performed.
- The schedular in RTOS is designed to provide a predictable execution pattern. In an embedded system a certain event must be carried across strictly defined time.
- OS which have a schedular with predictions execution pattern is called RTOS.

Features:

1. Context switching latency should be short
2. Interrupt dispatch latency should be short
3. Support kernel pre-emption.
4. Interrupt latency should be short.

RTOS Types

1. **Hard real time:**
   - System where it is absolutely impreative that responses occur within the required deadline.
   Eg: Flight control system.
2. **Soft real-time:**
   System where deadlines are imp but which will still fun correctly if deadlines are occasionally missed.
   Eg: Data acquistion system.

3. Real Real-time

System which are hard real time and which the response time are very short.

Eg: Missible guidance System.

4. Firm real-time

Systems which are soft real-time but in which there is no benefit from late delivery of Service.

OS object:

Task Events, Counters, Schedular, Resources, Alaram & Hook Fun are OS objects.

Task:

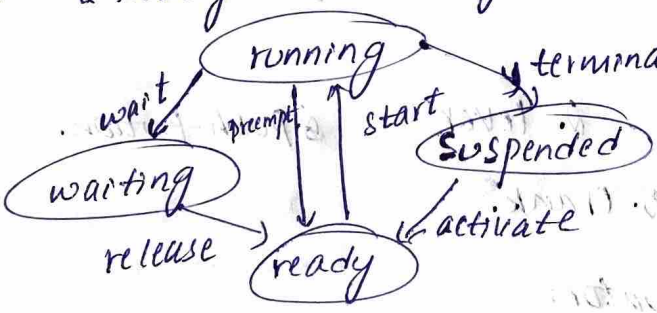
A Task provides the framework for the execution of fun Complex Control Slw can Conventiently be sub divided in parts.

- parts executed according to their real time requirements.
- Two different task concepts are provided by the OSEk operating System.

Extended task:

Extended tasks have four task start.

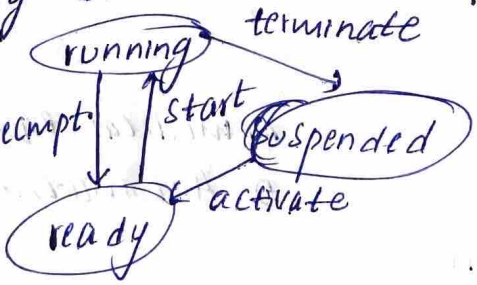
1. Running 2. Ready 3. waiting 4. Suspended.



Basic Task:

Basic task have four task States

1. Running 2. Ready 3. Suspended Basic tasks do not have a waiting state.



Q6. Decentralized systems →

- Cyber Physical Production Systems (CPPS) are complex manufacturing systems that aim to integrate and synchronize machine world and manufacturing facility to the cyber computational space.
- Intensive interconnectivity and computational platform is necessary for the real world implementation of CPPS.

Features of decentralized cyber physical systems →

→ Real time interaction

→ Security

→ Low power consumption

→ Automated operation

→ Scalable architecture

→ Decentralization

→ Robustness

→ Self correction

→ Node status

→ Fault detection

Decentralized System architecture →

The system architecture for Cyber Physical production system (CPPS) consists of three distinct phases :

1. Automatic Production Plan generation →

We automatically generate the production plan based on a system description and formalized specifications.

2. Production plan validation →

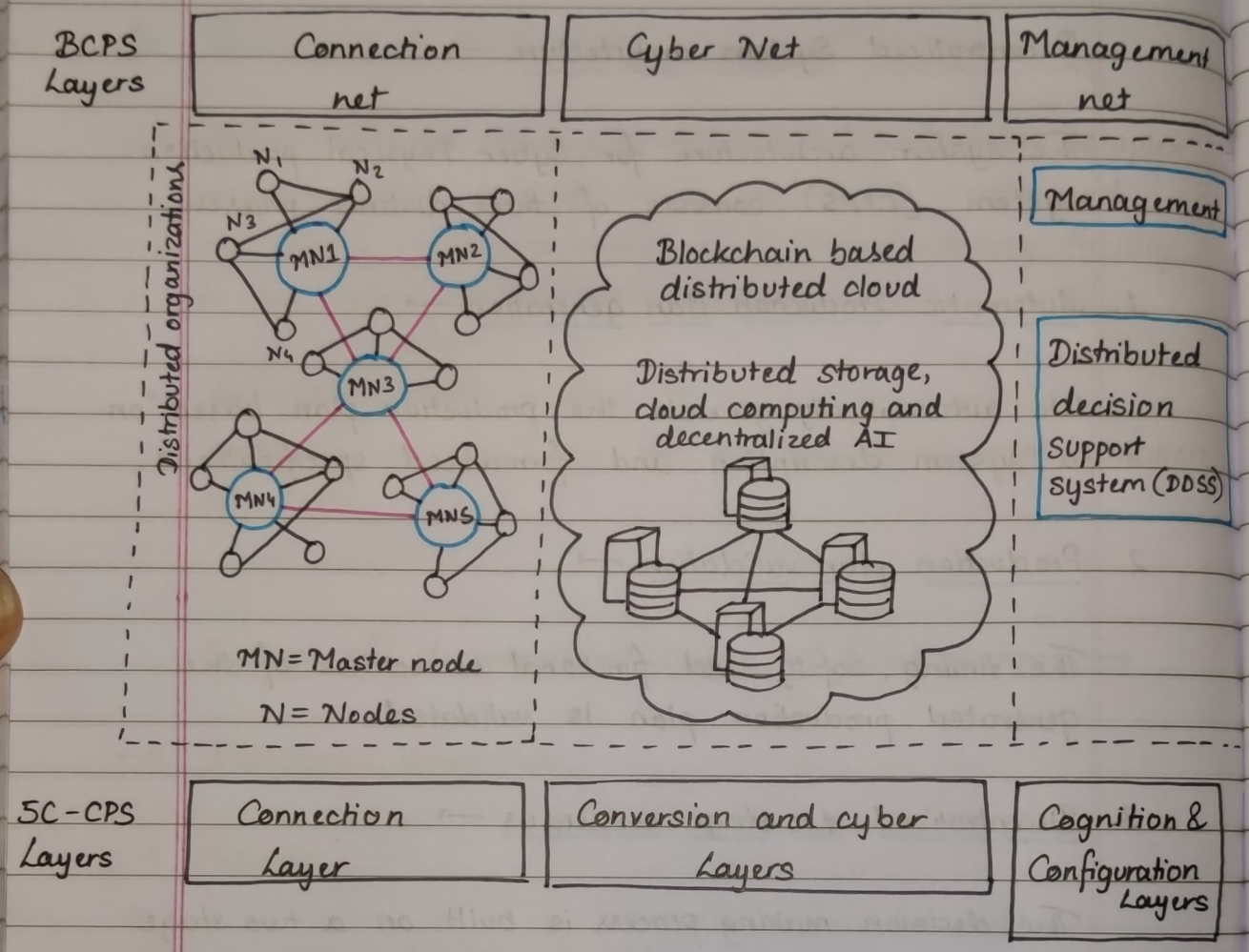
The timing, safety and functional correctness of the generated production plan is validated.

3. Decentralized two stage consensus →

The decision making process is built on a two stage consensus algorithm, featuring →

- A majority agreement on the safety and optimality.
- A unanimous agreement of all executing devices on the feasibility and authenticity of the plan.

The hardware design of the architecture is visualized in the following diagram

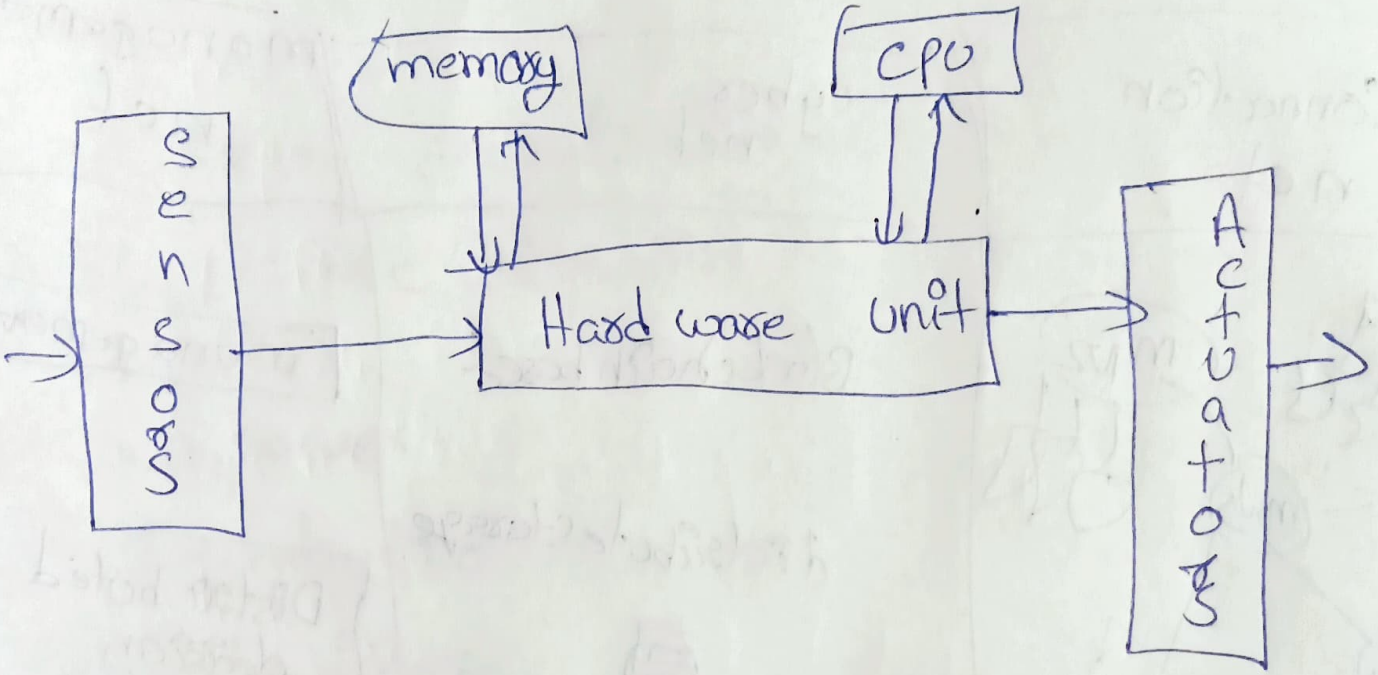


DECENTRALIZED COMPUTER ARCHITECTURE FOR CPPS

We consider a mesh network of ~~homogenous~~ heterogenous devices with ~~vario~~ varying functionalities and computational capabilities.

The generation of production plans is outsourced to cloud devices, whereas the functional validation can be performed on the computationally stronger devices in the mesh network as well.

Architecture of Embedded System



① typical system has two main parts.
  ① Embedded software
  ② Embedded hardware.

① Embedded hardware includes micro processors, microcontrollers, memory, controllers.

① Embedded software includes · operating system

The Design process

It consists of

1. Modeling
2. Design
3. Analysis

modeling: It is a process of gaining deeper understanding of a system, through the properties of system and what system does

Design: It is a structure creation of artifacts. It specifies what a system does

Analysis: It is a process of gaining a deeper understanding of a system through dissection ~~It specifies why~~.

Components of embedded system

1. hardware
2. software
3. Real time OS

Important steps in developing Embedded system

1. Requirement defination
2. System specification
3. functional Design
4. Architectural Design
5. prototyping design

Elements in Embedded system

1. processor
2. time counter
3. memory
4. Communication ports

Programing languages

1. C
2. Embedded-C
3. C++
4. Python
5. Java
6. B#

Examples

1. fitness tracker
2. medical devices
3. GPS
4. ATM
5. factory Robots
6. Automotives

Advantages

1. fast in performance
2. Small in size
3. less power

Disadvantage

1. Difficult to backup
2. maintanance is difficult.