3rd 13 mark

Designing with computing platforms involves several key steps, each of which plays a crucial role in creating a functional and efficient system. Here are the steps with the requested side headings:

1. System Architecture:
- Define the system's architecture, including the high-level structure of components and their interactions.
- Select the appropriate computing platform (e.g., microcontrollers, CPUs, GPUs) based on the system's requirements.
- Choose communication protocols and interfaces for inter-component communication.
- Consider factors like scalability, reliability, and power efficiency in the architecture.

2. Hardware Design:
- Design the physical hardware components of the computing platform, which can include PCBs (Printed Circuit Boards), integrated circuits, sensors, actuators, and power supplies.
- Consider factors like form factor, size, and power consumption when designing hardware.
- Ensure that the hardware components are compatible with the selected computing platform.
- Conduct prototyping and testing of hardware components to validate their functionality.

3. Development Environments:
- Set up the development environment, which includes selecting appropriate software tools, compilers, and programming languages.
- Develop or configure the software framework and libraries needed for the computing platform.
- Implement the software stack, including operating systems (if applicable), drivers, and application software.
- Establish version control and collaboration tools to manage the development process efficiently.

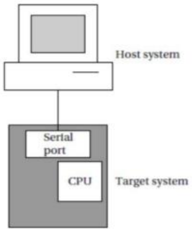


Fig.Connecting a host and a target system

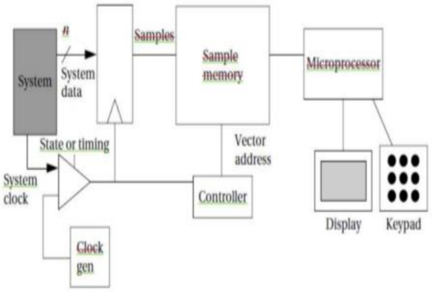
4. Debugging Techniques:

- Use debugging techniques like breakpoint debugging, logging, and code analysis tools to identify and resolve software issues.
- Employ hardware debugging tools such as oscilloscopes, logic analyzers, and JTAG (Joint Test Action Group) interfaces to diagnose hardware problems.
- Implement unit testing and integration testing to catch software bugs early in the development cycle.
- Employ simulation and emulation environments to test and debug components before deploying on the actual hardware.

5. Debugging Challenges:

- Addressing hardware-software integration issues, which can be complex and time-consuming.
- Handling real-time constraints and debugging problems that occur within strict timing requirements.
- Dealing with intermittent and non-reproducible bugs that may arise due to hardware-specific factors.
- Ensuring that debugging tools are compatible with the selected computing platform and can access necessary debugging interfaces.
- Managing debugging in resource-constrained environments, where memory and processing power are limited.

Effective system design and debugging are critical to the success of computing platform-based projects. By following these steps and considering the associated challenges, designers can create reliable and efficient systems that meet their intended goals.


n
System
System data
Samples
Sample memory
Microprocessor
State or timing
System clock
Vector address
Controller
Display
Keypad
Clock gen