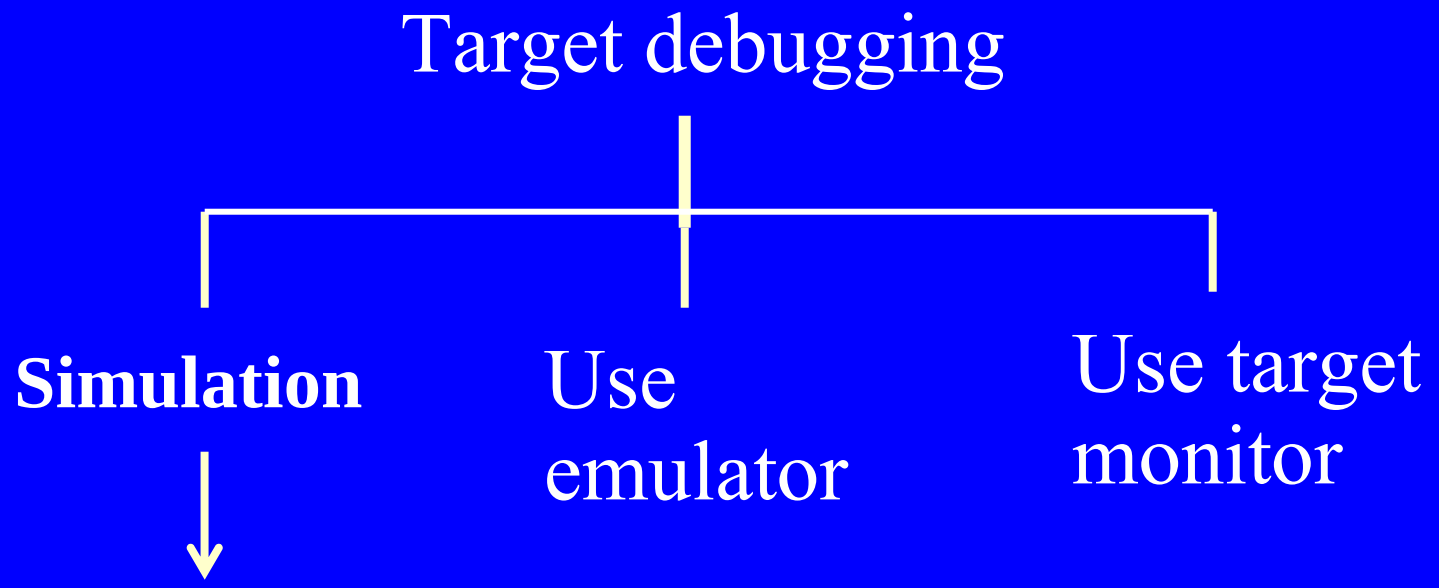
Embedded Software development Process and Tools:

Lesson-2

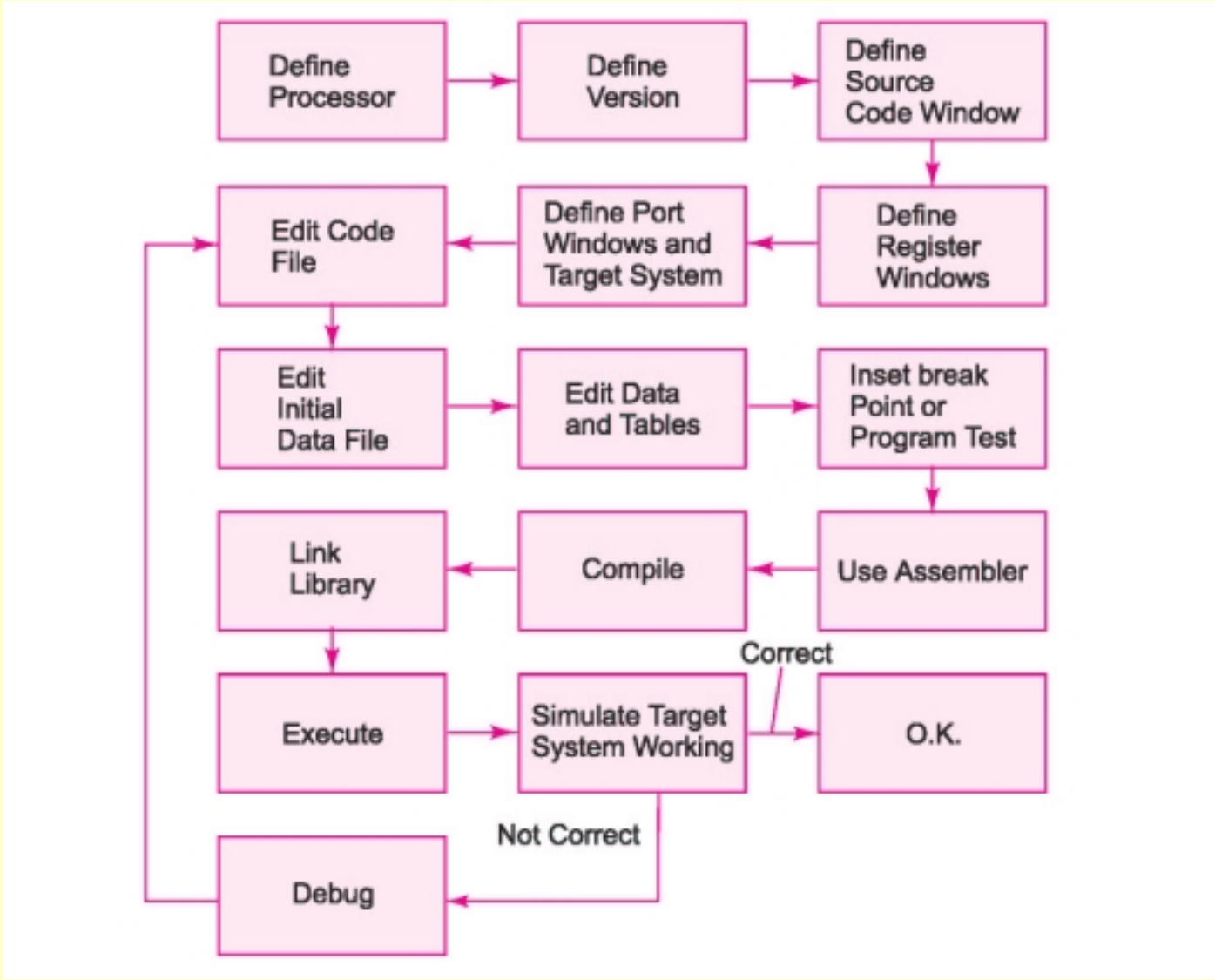
Simulator

1. Development processes using simulator



Code tested for the MCU/ system by simulating it on host computer used for code development

Detailed design development process using the simulator



Simulator

- Uses knowledge of target processor or microcontroller, and target system architecture on the host processor.
- First does cross compilation of the codes and place these into host system RAM.
- Behavior of target system processor registers simulated

Simulator …

- Uses linker and locator to port the cross compiled codes in RAM and functions like the code that would have run at the actual target system.
- Simulates hardware units like emulator, peripherals, network and input-output devices on a host (PC (or workstation or laptop).

Simulator …

- A simulator remains independent of a particular targeted system
- Results expected from codes at target system RAM, peripherals, network and input-output devices obtained at the host system RAM

Simulator …

- Helps in the development of the system before the final target system is ready with only a PC as the tool for development.
- Simulators readily available for different processors and processing devices employing embedded systems

Simulator …

- Helps in the development of the system before the final target system is ready with only a PC as the tool for development.
- Simulators readily available for different processors and processing devices employing embedded systems

2. Simulator Features

Simulator Features

- Defines the processor or processing device family as well as its various versions for the target system.
- Monitors the detailed information of a source code part with labels and symbolic arguments as the execution goes on for each single step.
- Provides the detailed information of the status of RAM and ports (simulated) of the defined target system as the execution goes on for each single step.

Simulator Features…

- Provides the detailed information of the status of peripheral devices (simulated, assumed to be attached) with the defined system.
- Provides the detailed information of the registers as the execution goes on for each single step or for each single module.

Simulator Features…

- Monitors system response and determines throughput.
- The windows on the screen providing details
- Detailed information of the status of stack, devices and ports (simulated) of the defined microcontroller system

Simulator Features…

- Program flow trace as the execution continues.
- Trace─ the output of contents of program counter versus the processor registers.
- Trace of application software means an output of chosen variables in a function of stepping sequence

Simulator Features…

- Helps the window on the screen to provide the detailed meaning of the present command.
- Monitors the detailed information of the simulator commands as these are entered from the keyboard or selected from the menu

Simulator Features…

- Supports the conditions (up to 8 or 16 or 32 conditions) and unconditional breakpoints.
- Breakpoints and the trace together are the important testing and debugging tool.
- Facilitates synchronizing the internal peripherals and delays.

Simulator Features…

- Employs RTOS scheduler that preempts task.
- Simulates the inputs from the interrupts, the timers, ports and peripherals. So it tests the codes for these.
- Provides network driver and device driver support

3. Simulator Possible Inabilities

Simulator Possible Inabilities

- May not resolve timing issues and hardware dependent problems
- Processor speed at target processor may not be adequately mapped with the host for calculating time responses and calculating output instances and throughputs at the target.

Simulator Possible Inabilities…

- May not be able to take into account portability problems.
- For example, target system may have 8-bit data bus between RAM and un-pipelined processor and host may be having pipelined processor and 32-bit bus

Simulator Possible Inabilities…

- A simulator may fail to show a bug from shared data as it arises from an interrupt in some particular situation only.
- For example, a long word in four registers loaded partly, and exactly at that moment the interrupt occurred.

Simulator Possible Inabilities…

- May not be able to simulate the ASICs and IP(s), embedded at the target system
- ASIC or IP core manufacturer may provide an alternative debugging tool
- ICE for the processor ARM7 or ARM9 emulates the ARM functions on host processor and system.

Simulator Possible Inabilities…

- May not be able to take into account internal devices existence.
- For example, the target system may use a Java accelerator, while host system may not have

4. Example of Simulator VxSim

Simulator VxSim

- A simulator tool, which provides a virtual target for developing and debugging the codes.
- Helps in avoiding the repeated code locating in actual target board of the embedded system.

Simulator VxSim…

- VxWorks RTOS task scheduling can be thoroughly simulated before implementation into the target
- Simulation APIs of the RTOS for given hardware

Simulator VxSim…

- Application Development tools — supports the UML and ‘RougeWave’, gives a short design cycle
- Native Development Environment— several native development environments and debugging. Environment may be MS Visual C++ or GNU tools

Simulator VxSim…

- Debugging Capability
- Device Simulation– Simulates devices and device driver behavior
- Network Simulation –Network simulation capabilities make it a virtual test bed, which permits modeling of complex multi-node networked systems.

Simulator VxSim…

- User interface Simulation

Summary

We learnt

- Simulator simulates most functions of a target embedded system circuit including additional memory, peripherals and buses on the host system itself.
- Makes application development independent of prior availability of a particular target system.

We learnt

- Simulates the real processes and outputs on the host system that will execute when the codes execute on the targeted particular processor .

End of Lesson-2 of chapter 14 on Simulator