UNIT 6

EMBEDDED SYSTEM IMPLEMENTATION AND TESTING

PART-A
SHORT QUESTIONS WITH SOLUTIONS

Q1. Define,

(i) Pre-processor

(ii) Interpreter.

Ans:

(i) Pre-processor

Pre-processor is a translation tool which pre-processes the source code.

(ii) Interpreter

An interpreter is also a program that reads the source program and executes line by line.

Q2. Define and list the examples of linker, compiler and interpreter.

Ans:

Model Paper-I, Q1(f)

Linker

A linker is a program used to join several small object files into a single large object file. When large programs are to be written, it becomes efficient to divide a large program into smaller modules. Each of these modules can be then separately written, tested and debugged. When all the modules are in working condition, they can be linked together to form a large functioning program.

Example

BL51 from Keil software.

Compiler

A compiler is a program that converts a source program to a target program. It converts or translates any high level language program to machine code. It doesn't produce output. A program which is compiled needs to be run by using another program to produce output. It may need to scan the program twice.

A compiler is user friendly as it does not disturb the user with errors line after line. It gives a complete set of errors in an understandable way to the user. It is not portable as it produces target code which is specific to machine. A compiled program needs more additional programs like linker/loader, assembler. They can be implemented easily in any programming language. Programmer can use trial and error methods to correct the errors.

Example

Microsoft visual studio.

Interpreter

For answer refer Unit-VI, Q1, Topic: Interpreter.

Example

Python, LISP, Ocamle.

Q3. Explain the need of editor. April-18, Set-4, Q1(f)

Ans:

In an embedded system, editor is used for the following reasons,

1. To write C code or assembly mnemonics of a program.
2. It supports addition, deletion, insertion, appending and entry of a program.
3. It allows merging of files or records at any required place.
4. It generates a source file which inturn holds an edited file.

Q4. List the different hardware debugging tools.

Ans:

Following are the different types of hardware debugging tools,

1. In-circuit emulator
2. ROM emulator
3. Background debug mode
4. Oscilloscope
5. Ohm meter
6. Voltmeter
7. Multimeter
8. Logic analyzer.

Q5. List the different software and manual debugging tools. Model Paper-II, Q1(f)

Ans:

The various software debugging tools are as follows,

1. Debugger
2. Monitor
3. Profiler
4. Instruction set simulator.

The different types of tools used for manual debugging are as follows,

1. Print Statements
2. Counters
3. Dumps
4. Output Ports
5. Fast Display.

Q6. State the difference between the quality assurance and testing and debugging.

Ans:

Differences between quality assurance and testing and debugging are mentioned below,

	Quality Assurance and Testing		Debugging
1.	The purpose of quality assurance and testing is to defect the bugs and confirm whether they are fixed.	1.	The purpose of debugging is to fix the bugs that are already detected.
2.	It needs to be done when the developer tries to break the system and encounters some problems.	2.	It needs to be done when the developer tries to to finish any part of design and encounters some problems.
3.	It also needs to be done when the developer tries to carry out test-to-pass and test-to-fail.	3.	It also needs to be done when the developer tries to carry out test-to-pass the bug fix.
4.	It makes use of different testing models such as static black-box and white box testing and dynamic black box and white box testing.	4.	It makes use of different automation and testing tools and techniques such as load tools, stress tools, inference, injectors, noise generators, analysis tools etc.

Q7. What is a target system? How does the target system differ from the final embedded system?

Ans: April/May-17, Set-2, Q1(f)

Target system is a system where the required hardware and software gets tested, debugged and finalized during the development of embedded system. This system is connected with a computer. Once the code is finalized, either the target system itself is considered to be as the final product (embedded system) or different copies of it are generated that acts as embedded system.

Target system differs from final system in its goal i.e., target system aims at recompiling the test programs, whereas final embedded systems aims to expose the defects to ensure that the softwares logic is working properly.

Q8. What is the use of host machine for embedded system?

Ans: April-18, Set-1, Q1(f)

In embedded systems, host machines are used to,

1. Develop the software
2. Test the initial stages
3. Test the software that is independent of hardware and
4. Run simulators.

Q9. Define instruction set simulator related to embedded system design.

Ans: Dec.-13, Set-3, Q7(a)(ii)

An instruction set simulator runs on host. It copies the hardware and simulates the master processor and memory.

The different characteristics of an instruction set simulator are as follows,

1. It estimates the response and throughput times by considering the difference of speed between the host and target.
2. The code generated by the simulator can easily be ported to the target hardware.
3. It keeps track of the variable, register and memory values.
4. It cannot simulate the behaviour of actual hardware in a precise manner.
5. It is cheaper compared to other tools.

Q10. List the various simulators used for embedded system testing. April-18, Set-2, Q1(f)

(or)

What are the various simulators used for embedded system testing? April/May-17, Set-1, Q1(f)

Ans:

There are two types of simulators used for testing embedded systems namely,

1. Instruction set simulator
2. HDL simulator.

1. **Instruction Set Simulator (ISS)**

ISS simulates the general purpose or single purpose processor at instruction level. It is a software based simulator.

2. **HDL Simulator**

Hardware Description Language Simulator (HDLS) simulates either of the processors at gate-level or behaviour level or RT level. It is a hardware based simulator.

Hardware or software co-simulator is used to protect the information regarding integration of an ISS and HDL simulator.

Q11. List the limitations of simulators.

Ans:

The limitations of simulators are as follows,

1. A simulator does not solve the problems such as hardware dependent and timing issues.
2. It does not support sufficient mapping of target processor speed with the host processor speed, required to calculate time responses, output instances and throughputs.
3. It is unsuccessful to demonstrate a bugs within the shared data that may occur from an interrupt only in certain situations.
4. It does not have the ability of simulating the ASCI's and IP's of target system.
5. It does not consider some internal devices available at the target system.
6. It cannot consider the portability problems.

Q12. What are laboratory tools? (Model Paper-III, Q1(f) | April/May-17, Set-4, Q1(f))

Ans:

The commonly used laboratory instruments for testing the embedded system are,

(i) **Oscilloscope**

An oscilloscope can also be referred to as scope. This is an analog device that is mainly used for determining whether the signal is low or high. It is also used to measure the exact voltage of a signal.

(ii) **Bit Rate Meter**

It is a measuring device that helps in determining the count of 0's and 1's within a given time span.

(iii) **Logic Analyzer**

A logic analyzer is an electronic device that displays signals in a digital circuit, that are very fast to be observed through human vision. It provides the signal to a user so that the user can keep track of the correct operation of the digital system.

Q13. What is logic analyzer? April/May-17, Set-3, Q1(e)

Ans:

A device which simultaneously captures and tracks the multiple signals and prepares a graph for them is referred as logic analyzer.

Q14. Define emulator related to embedded system design. Dec.-13, Set-3, Q7(a)(i)

Ans:

In-circuit emulator is abbreviated as 'ICE'. It is also called an 'emulator'. ICE is a hardware device, which is usually used for debugging the software of an embedded system. An emulator takes the place of a microprocessor in a target circuit i.e., microprocessor is replaced by an emulator. In a target circuit, an emulator plays exactly the same role as that played by the microprocessor.

PART-B
ESSAY QUESTIONS WITH SOLUTIONS

6.1 THE MAIN SOFTWARE UTILITY TOOL, CAD AND THE HARDWARE

Q15. Explain about main software utility tool.

April/May-17, Set-4, Q7(a)

(or)

What are the main software utility tools used for writing the code?

Ans:

There are two different types of tools available using which source code is written. They are,

1. ASCII text editor
2. IDE.

1. ASCII Text Editor

ASCII text editor is a tool using which any type code can be written without depending upon any language or platform. It allows the code to be edited and saved. It does not have much of the features other than word-wrap save etc,. But, they prove to be excellent for bug fixing and job completion if the user has complete knowledge about the code written.

2. IDE (Integrated Development Environment)

IDE is a tool used to write source code. It is platform dependent. It integrates a set of tools for editing, compiling and debugging, along with ASCII text editor in order to manage the software projects. It is provided by different vendors such as vendors of IDE, OS, language and any hardware manufacturer that provides the IDE along with its hardware kit.

Q16. Explain in detail about CAD tools and hardware.

Ans:

Model Paper-I, Q7(a)

The hardware engineers make use of the CAD tools to imitate the circuits from electrical view in order to know the functionality of the circuit under different conditions. One of the familiar standard circuit simulator called pspice which is the PC version of spice is shown in figure (1).

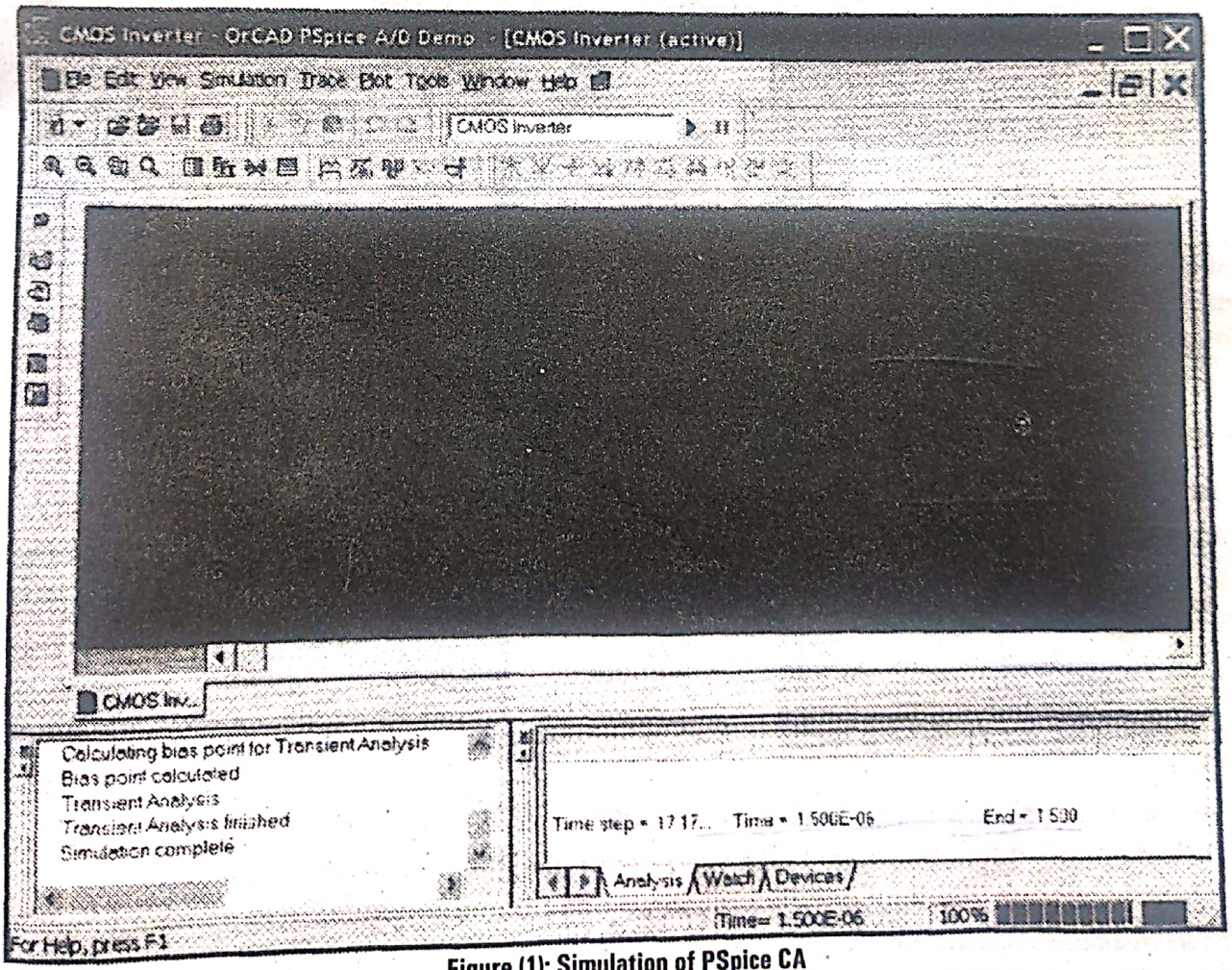


Figure (1): Simulation of PSpice CA

The above shown simulator is a modified version of some other circuit simulator called spice which was actually designed at university of California, Berkley. It is responsible for performing different circuit analysis such as non-linear transient, linear A.C, noise, distortion, non-linear D.C etc,. The circuits which are created in this simulator constitute various active and passive elements. Figure (2) shows a sample of pspice CAD circuit.

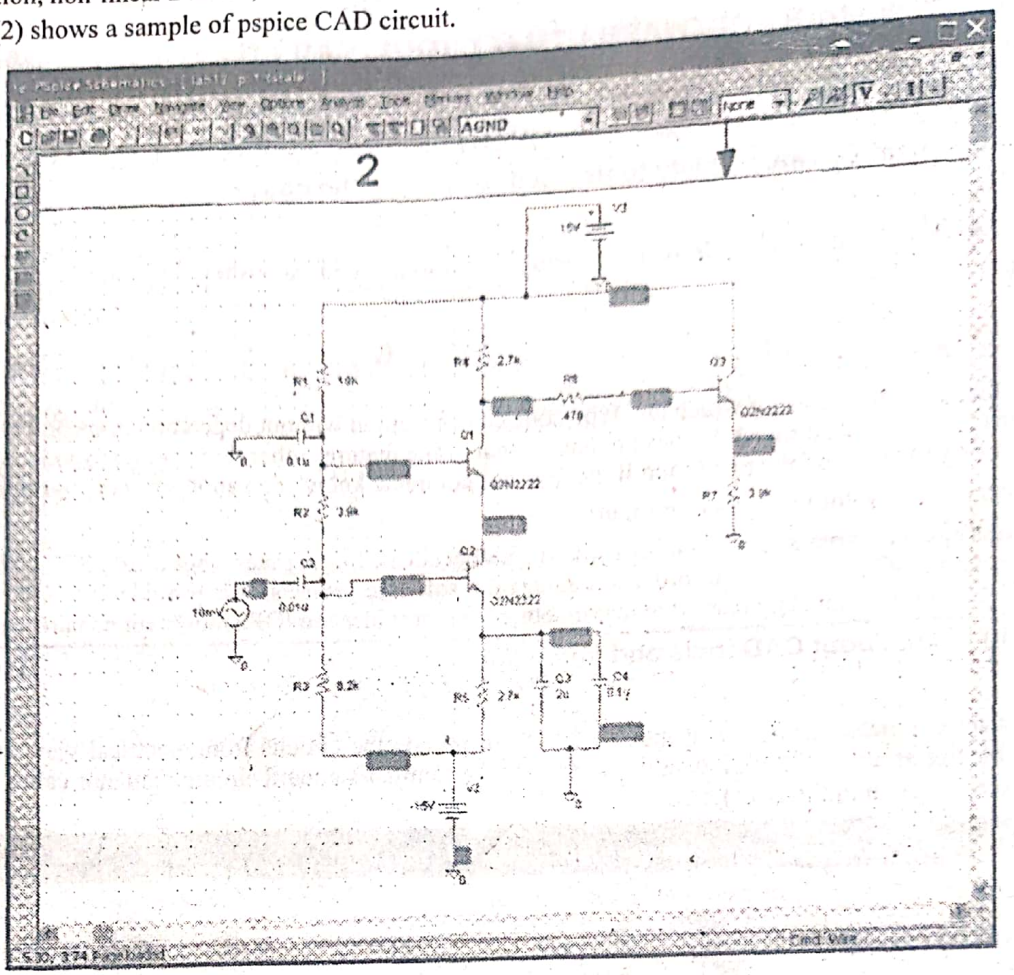


Figure (2): Circuit of Pspice CAD

Many industrial techniques are available which make use of CAD tools for simulating a circuit due to the significance of hardware designing and its costs. A set of related simulators and models are used to perform simulation on the design in case of complex circuits of a processor or board in difficult situations. But this is difficult to be done in hardware designing without considering the simulators accuracy or efficiency.

Q17. Write a short note on OrCAD EDA tools.

Ans: (Nov./Dec.-14, Set-1, Q8(a) | Nov./Dec.-14, Set-2, Q8(a))

OrCAD EDAtools

OrCAD is a software tool for electronic design automation which is used to design electronic schematics and electronic prints for manufacturing printed circuit boards. It consists of a schematic editor (capture) a circuit simulator (pspice) and a PCB designer.

OrCAD Capture

It is a schematic capture application containing component information system to link the data of component package footprint (or simulation behaviour) with schematic circuit symbol.

It is used to export netlist data to the simulator (orCADEE) and hardware description of the circuit schematic to verilog or VHDL. It also exports netlists to designs of circuit board like orCAD layout, allegro etc.

Capture supports TCL/TK scripting functionality to write scripts for the users. It's market place allow the customers to share and sell add-ons and design resources.

OrCAD EE PSPICE

PSPICE abbreviates personal simulation program with integrated circuit emphasis. orCAD EE pspice is a SPICE circuit simulator application used to simulate and verify analog and mixed-signal circuits. It is used to run simulations for orCAD capture defined circuits and also provides an interface to integrate with MATLAB.

PSPICE is a simulator which creates an output file to store simulation results. orCAD EE interface display these results graphically. Hence, orCAD EE PSPICE is also called as an improved version of pspice simulator. It includes automatic circuit optimization, viewing, recording, curve-fitting and post processing of waveform.

orCAD EE consists of a library of models for physical components and mathematical functions. It also supports model editor, parameterized models, auto-convergence and checkpoint restart and magnetic part editor.

6.2 TRANSLATION TOOLS – PRE-PROCESSORS, INTERPRETERS, COMPILERS AND LINKERS

Q18. Explain the important features of compilers and linkers that are relevant to embedded system?

(Model Paper-II, Q7(a) | April-18, Set-2, Q7(a))

(or)

List and describe the translation tools used in an embedded system. April-18, Set-2, Q7(b)

(or)

Explain in detail translation tools-pre-processors.

(Refer Only Pre-processor)

April/May-17, Set-3, Q7(a)

(or)

Explain the important features of the compiler and linkers.

(Refer Only Complier and Linker)

Ans: Dec.-13, Set-4, Q8

The translation tools used in embedded systems are,

1. Pre-processor
2. Compiler
3. Linker
4. Loader
5. Interpreter.

1. Pre-processor

Pre-processor is a translation tool which pre-processes the source code. The preprocessing includes the translation and reconstruction the source code file by freeing it from directives (begin with #), expanding its macros and substituting the text, etc,. The resultant file obtained is called preprocessed source code file which will become input to the compiler. This whole processing is done before the source code gets compiled or interpreted.

The preprocessor can be used separately or can even be incorporated into either translation unit or interpretation unit.

2. Compiler

It converts or translates any high level language program to machine code. It doesn't produce output. A program which is compiled needs to be run by using another program to produce output. It may need to scan the program twice.

A compiler is user friendly as it does not disturb the user with errors line after line. It gives a complete set of errors in an understandable way to the user. It is not portable as it produces target code which is specific to machine. A compiled program needs more additional programs like linker/loader, assembler. They can be implemented easily in any programming language. Programmer can use trial and error methods to correct the errors.

3. Linker

When large programs are to be written, it becomes efficient to divide a large program into smaller modules. Each of these modules can be then separately written, tested and debugged. When all the modules are in working condition, they can be linked together to form a large functioning program.

The linker creates a 'link file' which includes the binary codes for all the combined modules. It also creates a 'link map file' which holds the information about the link files. A linker does not provide absolute addresses to a program, but assigns only relative addresses starting from zero. Linker program is relocatable as it can be placed anywhere in the memory to run.

4. Loader

A loader reallocates the available addresses, and places address into the memory at some particular physical address. Binary code must be loaded into different addresses available and then the program is allowed to run on the system. A loader is a part of the operating system and performs multiple operations on host machines.

5. Interpreter

It reads the line, checks the errors if any and reports it to the user. If no errors are detected then gives the output. It takes the source program along data and produces output line by line.

It scans the program and disturbs the user after checking each line if there is any error. Even error reporting is done in simple English.

Interpreters are portable as they produce code which is machine independent. The programmer needs to correct the incorrect line and only then the interpreter checks the next line.

6.3 DEBUGGING TOOLS

Q19. What are the different types of hardware debugging tools available? Explain.

Ans:

Following are the different types of hardware debugging tools,

1. In-circuit emulator
2. ROM emulator
3. Background debug mode
4. Oscilloscope
5. Ohm meter
6. Voltmeter
7. Multimeter
8. Logic analyzer.

1. In Circuit Emulator

In-Circuit Emulator (ICE) is a hardware device, which is usually used for debugging the software of an embedded system. An emulator takes place of a microprocessor in a target circuit microprocessor is replaced by an emulator. It is used to monitor and modify the internal registers, memory, variables and caches etc. It consists of an overlay memory used for simulating ROM. It debugs the code by creating the breakpoints or by performing single stepping process. In-circuit emulator is the most expensive hardware debugging tool.

2. ROM Emulator

This tool simulates ROM by replacing the ROM with cables connected to RAM (dual port). ROM emulator serves as an intermediate hardware device which is connected to the target and host using different ports. It is used to modify the data present in the ROM It is used to view the ROM code dynamically and to create the breakpoints within it. It is not capable to support the devices such as on-chip ROM, custom ASIC's etc. It behaves as a universal tool and can be integrated with other debuggers.

3. Back ground Debug Mode

Debugging using a BDM (Back ground Debug Mode) is also known as on-chip debugging. It creates the break points to stop the execution of any software. It provides the facility to read and write the I/O ports, RAM and registers. It allows the observation of software execution in real time. It depends upon the processor. Although it is cheaper than ICE, more flexible than it.

4. Oscilloscope

Oscilloscope is an analog device that is mainly used to measure the exact voltage of a signal with respect to any given time. It is used for the verification of circuit functioning. It is able to keep track of maximum of two signals concurrently. It is used to observe the changes occurring in signals on any I/O port. This observation helps in verifying running softwares, time taken for changing one signal to other and so on. It is an independent of processor and can sometimes be used as voltmeter.

5. Ohmmeter

It is a device used for measuring the resistance between any two points on a circuit. It is used to determine the variations in voltage/current with respect to resistance. It is the cheapest hardware debugging tool.

6. Voltmeter

This device is used to measure the voltages between any two points on a circuit. It can also be used to determine whether power is available in circuits chip. This device is also quite cheap when compared to other tools.

7. Multimeter

Multimeter is mainly used for measuring the voltage and electrical resistance. The working of multimeter is similar to that of ohmmeter and voltmeter.

8. Logic Analyzer

A logic analyzer helps in simultaneous capturing and tracking of multiple signals and prepare a graph for them. It operates on two modes viz timing mode and state mode. Similar to an oscilloscope, it determines any change in signals on I/O ports. It can access only the data that is passed to slash from the processor externally. It is not able to access the data of internal memory, registers, caches etc. It is processor independent. The captured data can be stored. It is an expensive hardware debugging tool.

Q20. List and describe the debugging tools used in an embedded system.

(Model Paper-III, Q7(a) | April-18, Set-3, Q7(a))

(or)

List and explain different software debugging tools.

Ans:

The various software debugging tools are as follows,

1. Debugger
2. Monitor
3. Profiler
4. Instruction set simulator.

1. Debugger

It is a functional debugging tool and its function depends upon the type of debugger used. However, the following are the functions of debugger.

(i) It creates breakpoints to stop the execution of software.

(ii) It also creates conditional breakpoints which helps in stopping the software execution if any problem is occurred during execution.

(iii) It can load, trace out or can perform stepping on target.

(iv) It can be used for modifying the RAM data, but not the ROM contents.

2. Monitor

Monitor is a debugging interface. It consists of debug software that runs on both the target and the host system. The debug software on host communicate with the debug software on target through serial port or ethernet.

3. Profiler

A profiler gathers the timing history of particular registers and variables. It stores both the behaviours (time-dependent) and execution pattern of the software being executed.

4. Instruction Set Simulator

An instruction set simulator runs on host. It copies the hardware and simulates the master processor and memory.

Q21. Discuss different types of manual debugging tools.

Ans:

The different types of tools used for manual debugging are as follows,

1. Print Statements

These are the functional debugging tools and are used within the code to print the status of variables, locations and so on. The register values and the variable output can be viewed even when the code is running. Print statements are used for verifying the execution of any segment of code and also decreases the execution time. However, it may result in missed deadline in real-time systems.

2. Counters

Counters are the performance and efficiency debugging tools wherein the timers can be reset and incremented in between the code. They determines the execution times by counting bus cycles or with the help of system clock.

3. Dumps

This tool dumps the data in any other storage device at run-time. Dumps are functional debugging tools and work similar to a print statement but with faster execution. These tools also estimate the overrun of stacks or heap by checking the contents of memory during run-time.

4. Output Ports

Output ports behave as performance efficiency and functional debugging tools. They are toggled i.e., inserted at different points in a software. They help us in studying the behaviour of multi-tasking system by allowing us to assign different ports to different tasks. If used with an oscilloscope (or logic analyzer), they can determine the execution time between various port toggling.

5. Fast Display

Fast display is also a functional debugging tool and work similar to a print statement. It consists of LEDs or LCDs that display the data. It works faster, it is less intrusive and can be applied to real-time deadlines. It also checks and confirms whether all the parts of the code are running or not.

6.4 QUALITY ASSURANCE AND TESTING OF THE DESIGN

Q22. What is quality assurance and testing of the design? Explain in detail.

April/May-17, Set-4, Q7(b)

(or)

Write a short note on quality assurance and testing of the design.

(Nov./Dec.-14, Set-4, Q8(a) | Nov./Dec.-14, Set-2, Q8(c))

(or)

Write short notes on quality assurance and testing of the embedded system design.

Ans: (April-18, Set-1, Q7(b) | Dec.-13, Set-2, Q8(b))

The purpose of quality assurance and testing is to defect the bugs and confirm whether they are fixed. It needs to be done when the developer tries to break the system and encounters some problems. It also needs to be done when developers tries to carry out test-to-pass and test-to-fail. It makes use of different testing models such as static black-box testing and dynamic black box and white box testing, box and white box testing.

Testing is a process in which software failures are detected in order to uncover and correct the defects and to ensure the software quality. To fix the defects each and every component along with the prototypes which are used further as designs foundation, also need to tested otherwise the system to is very poor. The defects such as hardware, software and also measure defect rate are to be considered. It is difficult to debug or replace the component that are costly.

Q23. Write a short note on embedded system testing.

Ans: (Model Paper-I, Q7(b) | Nov./Dec.-14, Set-1, Q8)

Testing is a process in which software failures are detected in order to uncover and correct the defects and to ensure the software quality. To fix the defects each and every component along with the prototypes which are used further as designs foundation, also need to tested otherwise the system results to be very poor. The defects such as hardware, software and also measure defect rate are to be considered. It is difficult to debug or replace the components that are costly.

Approaches of Testing

Basically the two approaches of testing are,

1. Static testing
2. Dynamic testing.

1. Static Testing

This approach involves the testing of software while the system is not being run.

2. Dynamic Testing

This approach involves the testing of software while the system is being run.

Method of Testing

There are two different software testing methods,

(i) Black box testing

(ii) White box testing.

(i) Black Box Testing

In this type of testing method, the tester do not have access to internal data structures and algorithms.

(ii) White Box Testing

In this type of testing method, the tester can have access to internal data structures and algorithms.

Q24. List and explain different types of testing techniques.

Ans:

Formation of Testing Models

There are basically four types of testing model that include several testing techniques. These models are formed when the testing methods (black box, white box) follow any of the testing approaches (i.e., static or dynamic). The four testing models are discussed below,

1. Static black box testing
2. Static white box testing
3. Dynamic black box testing
4. Dynamic white box testing.

1. Static Black Box Testing

Static black box testing is performed by considering two criterias.

(i) High Level Specification Testing

High level specification testing includes tracing out of basic problems at high level, failures and breach of research existing guidelines/standards.

(ii) Low Level Specification Testing

Low level specification testing includes checking accuracy, completeness, relevance, feasibility, preciseness, consistency.

2. Static White Box Testing

Static white box testing includes the process of examining and reviewing the errors in code (or hardware) that is not executing.

3. Dynamic Black Box Testing

Dynamic black box testing includes the process of testing the executing code (or hardware) without knowing the internal details i.e., only entering the input, obtaining the outputs and verifying the results.

(i) Data Testing

It verifies data about the input and output.

(ii) Boundary Conditions Testing

It involves the testing of bugs and errors that usually occur at boundary values (conditions) such as maximum, minimum, inside or outside boundaries.

(iii) Internal Boundary Testing

It is a testing situation which perform tests on powers of 2 i.e., 2^2, 2^4 including ASCII table.

Input Testing

It is a testing situation which performs test to check invalid and null data.

State Testing

It is a testing technique which performs tests on software modes and transitions present within state variables. For instance starving software, feed software.

4. Dynamic White Box Testing

Dynamic white box testing includes the process of testing the executing code with the full knowledge of internal details such as data structures, algorithms, etc. It directly perform tests on high-level specifications and low-level specifications.

Q25. Write a short notes on the following,

1. Test cases

2. Code inspections.

Ans:

1. Test Cases

Test cases can be defined as the set of conditions based on which an application or a software is tested to find out whether it is working properly, meeting the architectural specification. Moreover these also ensure that the system meets the actual requirements with irrespective of whether they discussed properly or not at all in the documentation.

The results obtained from these test cases are managed differently by different organizations. Inaddition to this the results are also based on formal and informal design reviews. The informal design reviews also called peer reviews follow the procedures such as, walk throughs, inspections etc., in order to exchange the information whereas the formal design reviews do not follows any particular procedure.

2. Code Inspections

Code inspection can be defined as testing strategy which is conducted to improve the code and reduce the bugs. In addition to this, code inspection also checks whether or not the programming language best practices have been followed.

Code inspection will be effective and efficient if it is performed by right type of team members.

The following type of roles must be present in the code inspection team,

(i) Author

A person who is responsible for explaining source, discussing errors present in the system, as well as the future rework.

(ii) **Reader**

A person responsible for reading out the source code and the respective specification for the operational investigation loudly.

(iii) **Moderator**

A person who is responsible to carry out the code inspections and conduct the meetings.

(iv) **Recorder**

A person responsible for recording checklist report and documents of the code inspection.

6.5 TESTING ON HOST MACHINE

Q26. Briefly discuss about testing on host machine.

(Model Paper-II, Q7(b) | April-18, Set-3, Q7(b))

(or)

Discuss the steps in testing at host machine.

Ans:

The steps involved in testing at host machine are as follows,

Step- 1

Initial Test

Each module or segment is tested in this step on host machine.

Step- 2

Test Data

A single test data is constructed in this step by performing each possible combination of data.

Step- 3

Exception Condition Test

All possible combination of exceptions are considered while performing test.

Step- 4

Test-1

Hardware independent codes are tested in this step.

Step-5

Test-2

Scaffold software is tested in this step.

Step -6

Test on interrupt service routines and hardware independent part.

Step-7

Testing the Hardware Dependent Interrupt Service Routine

Hardware dependent interrupt service routines are called and tested in this step.

Step-8

Testing the Timer Routines

Hardware dependent codes containing timing routines like clock tick set, counts get, counts put delay are tested in this step.

Step-9

Assert Macro Test

Assert macro test is performed at the host machine. In this technique the assert macros are inserted at different places within the program to check whether a condition/parameter becomes true/false. If the result of assert macro test comes to false, the program immediately halts.

6.6 SIMULATORS

Q27. Write short notes on simulator and its features.

(or)

What do you meanby application software for a target system?

(Refer Only Simulator)

Ans: April-18, Set-3, Q1(f)

Simulator

Simulator is available at the host system such as PC, laptop or workstation (that run the windows as their operating system). It is the model of target system i.e., it has the appearance and characteristics of the target system's processor/ microcontroller and architecture. It cross-compiles the source code (written in C/assembly)and places it at the RAM (of host system) along with the behaviour of the registers (of the target system). It also has linker/locator that allocates the memory for the cross-compiled code, links it to the external references and allows it to run as if it is running on the actual target system. It is used during the development phase of application software. It is independent of the target system. It produces the results at the host system similar to the results that would have to be produced at the target system.

Features of Simulator

The simulator is supported by the systems (such as PC) running on windows as their operating system. It possess the following features,

1. It has processors of different families and is available in various versions.
2. It maintains all the details of the source code part such as labels and symbolic constants during each step of the execution.
3. It maintains the status of RAMs and ports of the target system during each step of execution.
4. It maintains all the information of the peripherals of the target system during each step of execution.
5. It also maintains the details of registers during each step of execution or during each single module of execution.
6. It keeps track of the system response and also estimates the throughput.

7. It supports the drivers for network and devices.
8. It synchronizes the internal peripherals and delays.
9. It supports the assembler, disassembler, interpreters, mouse selected macros for *C* language and assembly language expressions.
10. It maintains all the information of simulator commands that are either selected from the menu or entered through the keyboard.

Q28. Write a short notes on V X Sim simulator tool.

Ans:

V X Sim is a simulating software tool used within the virtual target system to generate and debug the code. It is also used to ignore the code that is repeated within the target hardware system. It is more useful in initial phases of development. Since the simulation of V X Works RTOS task scheduling can be completely done prior to its implementation into the target system.

This tool possess the following features,

1. It supports applications development tools like UML and 'Rouge Wave' which enables short design cycles.
2. It is capable of debugging and can thus determine the presence of flaws/faults easily.
3. It is capable of simulating the devices as well as behaviour of device driver.
4. It simulates the user interface (For instance simulation of setup box interface).
5. It simulates the network to model the complex multinode network system. Hence, acting as a virtual test bed.
6. It has native development environment
7. It also simulates the APIs of the RTOS.

6.7 LABORATORY TOOLS

Q29. Explain about Laboratory instruments for testing the embedded system.

(April-18, Set-1, Q7(a) | Dec.-13, Set-1, Q8(b) | Dec.-13, Set-2, Q8(a))

(or)

Explain about laboratory tool.

Ans:

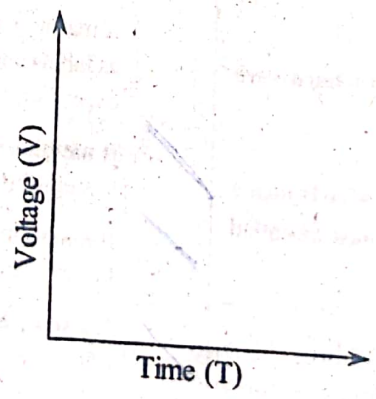
April/May-17, Set-3, Q7(b)

The commonly used laboratory instruments for testing the embedded system are,

1. **Oscilloscope**

An oscilloscope can also be referred to as scope. This is an analog device that is mainly used for determining whether the signal is low or high. It is also used to measure the exact voltage of a signal.

Graph Representing Voltage Versus Time



Figure

This device consists of a display, two probes with leads attached to it and control sets (such as knobs and buttons).

On oscilloscope screen the graph is displayed. Leads are wires which are used for connecting the CRO input to the observation point and at the other end have a crocodile (alligator) clip for connecting to the electronic circuit.

The probes have finely sharpened ends which acts as a means to get connected to the electronic device and its respective signal can be viewed on the oscilloscope screen. To cover those metallic pointed ends, a small plastic black colored cap is fitted which is called as hitches cap. Oscilloscope consists of two probes, each probe having a ground lead. Leads are often short (convenient length) wires extending from probe's head for connecting CRO and a circuit. The ground lead has an alligator clip for easily connecting to the electronic circuit.

Basic controls are,

ON-OFF

This switch turns ON or OFF the oscilloscope.

Intensity

This controls the brightness of the light produced by beam spot. Actually, it controls the number of electrons (i.e., magnitude of omission of the electronic beam) that are bombarding the screen.

Focus

This controls the sharpness of the spot.

Astigmatism

This is another focus control with the help of both focus and astigmatism, a very sharp spot can be obtained in the centre and as well as at the edges of the screen. This adjustment is made to obtain the roundest spot on the screen.

X-shift or Horizontal Position Control

By varying the voltages between the horizontal plates, the X-position of the spot is adjusted. When the spot is at the centre, the two horizontal plates have the same potential.

Y-shift or Vertical Position Control

By varying the voltages between the vertical plates, the Y-position of the spot is adjusted. When the spot is at the centre, the two vertical plates have the same potential.

Time Base Control

It is used to observe the waveform of time varying signals. If a signal has time period of 20 msec. Following are the two different time base control selections by which it can be represented.

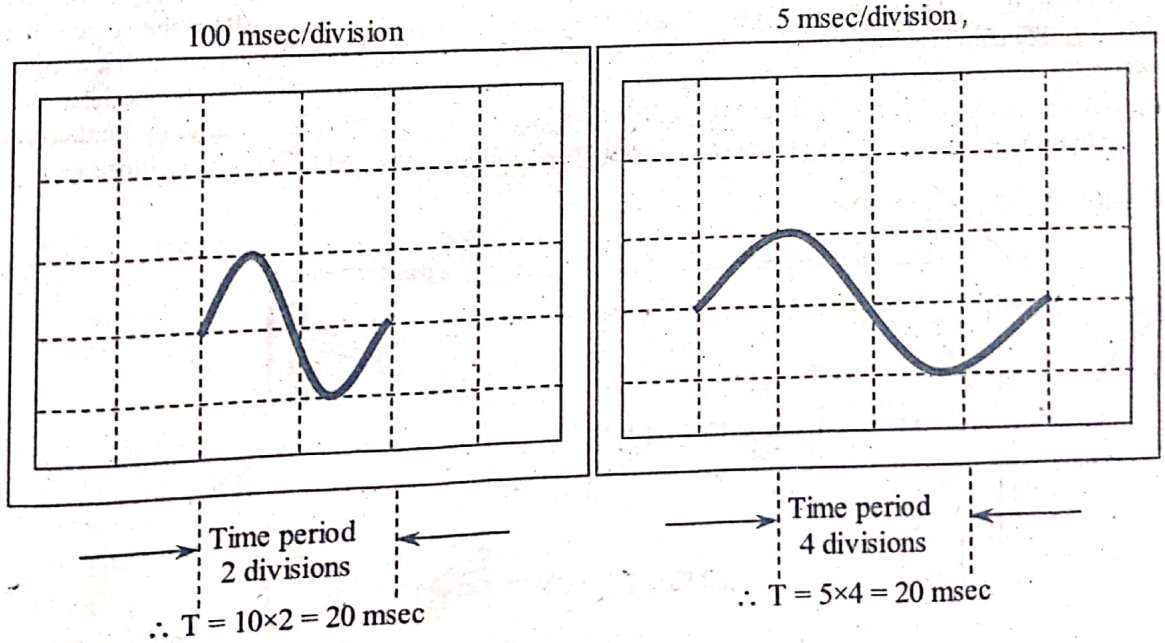


Figure (2)

Synchronous Selector

The signals can be selected using a synchronous selector switch. The various types of signals which can be selected are, internal , line and external.

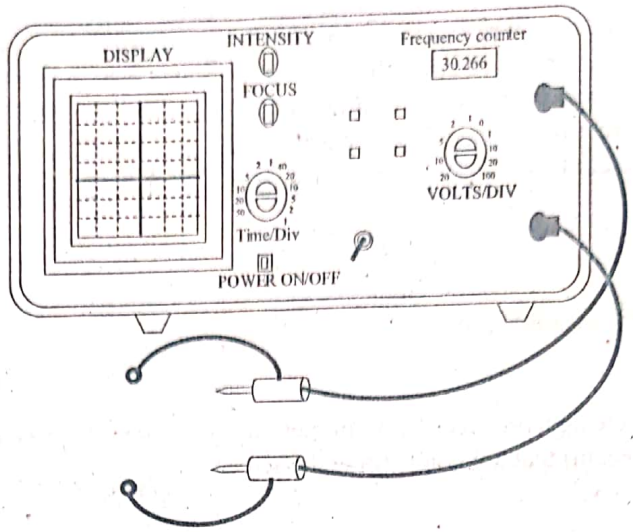


Figure (3): Oscilloscope

A storage oscilloscope (or storage scope) is another kind of oscilloscope which captures a signal by seeing only once and stores it in the oscilloscopes's memory, later it generates the screen display from that memory. This oscilloscope is the most expensive among the other.

2. Bit Rate Meter

It is a measuring device that helps in determining the count of 0's and 1's within a given time span. A bit rate meter is mainly used for the following two reasons,

(i) For measuring the throughput i.e., the total number of bytes/second available within a network.

(ii) For evaluating the bits (0's and 1's the test message and then to match them with the bits of the actual text message.

3. Logic Analyzer

A logic analyzer is an electronic device that displays signals in a digital circuit, that are very fast to be observed through human vision. It provides the signal to a user so that the user can keep track of the correct operation of the digital system.

Oscilloscope, voltmeters, ammeters, multimeter are conventional instruments. These instruments do not respond quickly to digital systems or equipments, microprocessor based systems or computer motherboard circuits while solving measurement problems associated with them. They are very slow to capture the signal and produce its graph on the screen. For example, consider a specific memory chip in a microprocessor based system. The chip has various digital activities that need to be carried out such as formation of digital data, address bus, transferring data to the memory during appropriate clock pulse etc., To obtain a detailed analysis, a logic analyzer is required which display simultaneous waveforms that are large in number. It is used for capturing data in systems that have too many channels that need to be examined with the help of an oscilloscope.

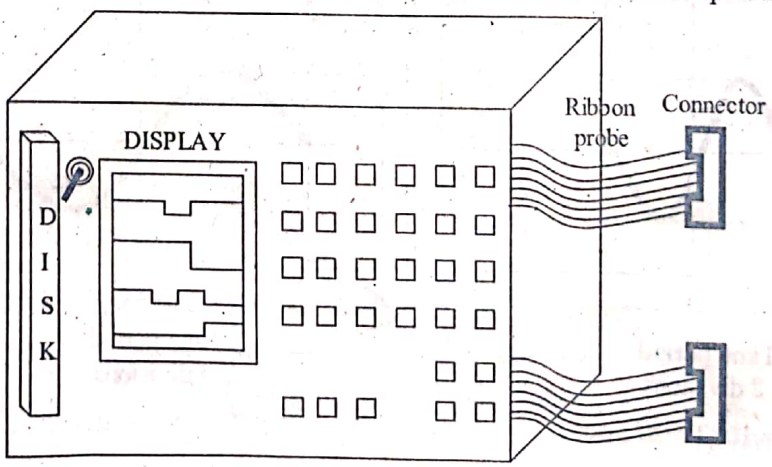


Figure (4): Logical Analyzer

Q30. Explain in detail about in-circuit emulator.

Ans: Model Paper-III, Q7(b)

In-circuit emulator is abbreviated as 'ICE'. It is also called an 'emulator'. ICE is a hardware device, which is usually used for debugging the software of an embedded system. An emulator takes the place of a microprocessor in a target circuit i.e., microprocessor is replaced by an emulator. In a target circuit, an emulator plays exactly the same role as that played by the microprocessor. ICE is a software developers tool in embedded design and it enables developers to closely observe the development of an embedded system. ICE replaces the processor or microcontroller of the target hardware.

ICE is a computer chip, that is used to emulate a microprocessor by loading a program into the emulator and allowing it to run, so that the software of the embedded system can be tested by the developers.

Advanced features of an emulator are as follows,

1. Performance analysis
2. Coverage analysis
3. A trace buffer and advanced trigger
4. Break point possibilities
5. Overlay memory-an emulator consisting of one or more blocks of memory.

In addition to mirroring the microprocessor, an emulator also supports,

1. Debugging same as the usual desktop software debugger. With this user can set the break points, once the break point is encountered, a user can examine memory contents as well as register contents, can view and even change the source code, execution can be resumed, step through the code. A user can view the register contents and memory contents even if the program crashes all of a sudden with the help of an emulator.
2. A trace can be captured by the emulator same as the data captured by the logic analyzer operating in a state mode. However, the trace is not as flexible as the data captured by the logic analyzer.
3. Emulators do not use the memory of the target system. Rather, the emulator uses its own memory called overlay memory (one or more blocks replacing the blocks in target system's memory). The user specifies the address ranges, the read/write blocks and the read-only blocks for which the emulator should use its overlay memory. Thus, a read or write performed by the emulated processor in these ranges, leads the emulator to use the overlay memory. This helps in easily and efficiently downloading the versions of the debugging software on the target machine.

Q31. Discuss in detail about monitors.

Ans:

Monitors

Among the most advanced debugging tools, monitor is also one of the tool used for debugging the software. Monitors provides an interface for debugging the software which is similar to that of a debugging tool known as in-circuit emulator. An important feature of the monitor is that, it lets you run the software on the real target machine and allows debugging too.

Working of Monitors

A monitor is divided into two parts, one part of the monitor runs on the target machine and the other part runs on the host machine. A monitor is a small program that resides inside a ROM (non-volatile memory) on the target machine. The program knows how to receive a software through network or a serial port, in addition to that it also knows how to copy the software into RAM and run it. This program is used for setting and checking break-points, memory and register. It also performs some other functions too for the sake of application debugger as per the requirements. Vendors usually refer this program by the name target agent, monitor or debugging kernel, but these are not standard names. Another part of the monitor which resides on host machine provides a debugging user interface in order to perform software debugging. This program establishes communication with the target agent or debugging kernel over a serial port or a communication network.

2. Write and compile or assemble all the modules. Based on the monitor specifications, a locator may run or may not. If a user purchases monitor form RTOS vendor, it might be or might not be necessary to link the real time operating system into the system.

3. While downloading the compiled modules, the program which is running on the host machine cooperates with the 'target agent'. During the download, locator functions are also carried out if required.

4. In order to set break-points, run and compile the program, instructions must be given to the monitor. On the host machine user interface will be running and on the target machine 'target agent' will be running. Communication is established between target and host by giving instructions to the 'target agent'.

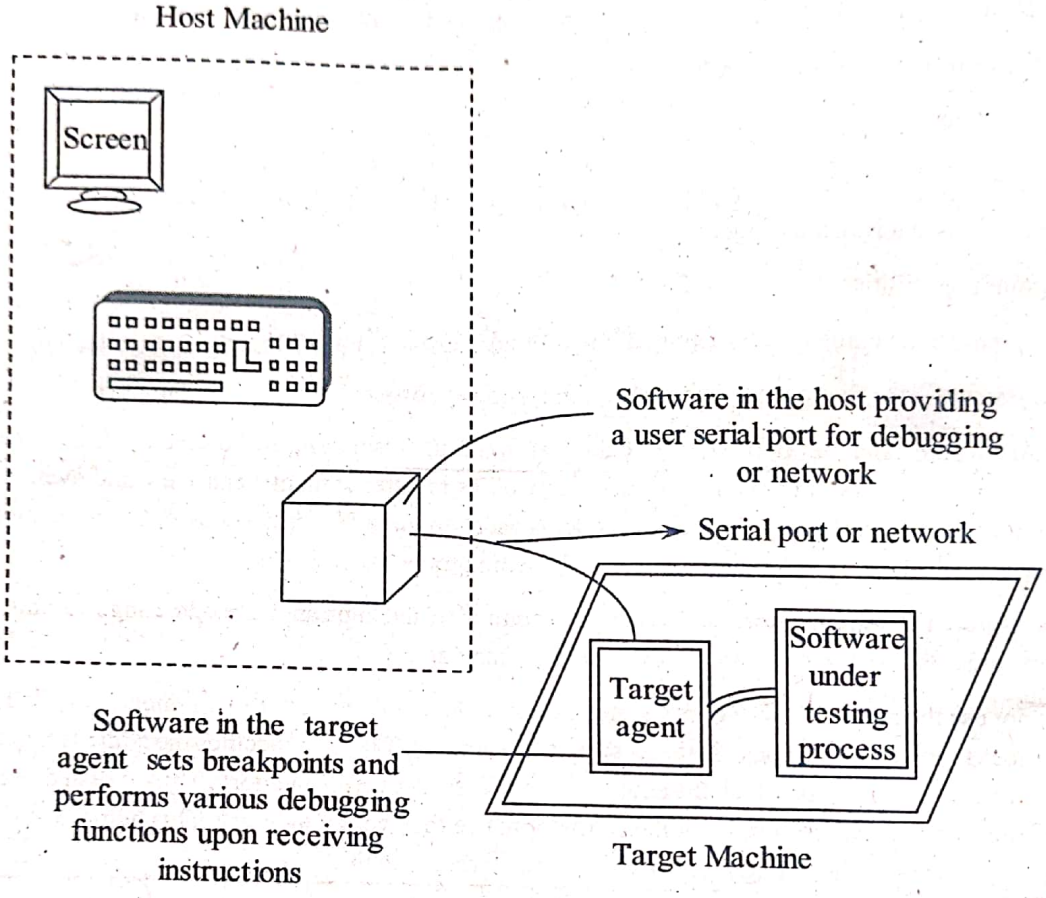


Figure: Software-only Monitor