Continuing Education Course #265
Verilog for Digital Design

1. Verilog can be used for designs that target:
- ○ a. board-level designs using discrete devices,
- ○ b. Field-Programmable Gate Array (FPGA) devices,
- ○ c. Application-Specific Integrated Circuit (ASIC) custom devices,
- ○ d. all of the above.

2. Verilog is a parallel and concurrent computer language:
- ○ a. True,
- ○ b. False.

3. Verilog can be used at several levels of design abstraction, including:
- ○ a. as a general-purpose numerical computation platform,
- ○ b. as a “Register Transfer Level” (RTL) digital design-capture tool,
- ○ c. at the device level, including MOS devices and components,
- ○ d. all of the above.

4. In the signal representations, individual signal bits can be represented as:
- ○ a. a logic “1” or “0,”
- ○ b. “X” (unknown),,
- ○ c. “Z” (un-connected),
- ○ d. any of the above.

5. Verilog program entry requires a schematic diagram tool:
- ○ a. True,
- ○ b. False.

6. Word-Processor software (like uSoft Word) is the best tool for Verilog textual entry:
- ○ a. True,
- ○ b. False.

7. The Icarus FOSS Verilog Compiler is available in multiple environments:
- ○ a. True,
- ○ b. False.

8. Alpha-Numeric Text messages can be inserted in Verilog to support programming:
- ○ a. True,
- ○ b. False.

9. Verilog programs require the keyword “module” to begin and “endmodule” to end:

○ a. True,
○ b. False.

10. Verilog supports program comments on each line:
○ a. True,
○ b. False.

11. The Verilog “initial” declaration signifies a portion of code that executes only once:
○ a. True,
○ b. False.

12. A Verilog Behavioral-Level of Abstraction can be used to provide a “test-bench”:
○ a. True,
○ b. False.

13. A Behavioral Level program must still follow the syntax rules of the Verilog language
○ a. True,
○ b. False.

14. The introduction to Verilog Behavioral Level is illustrated using:
○ a. a text editor,
○ b. the iverilog compiler,
○ c. both of the above
○ d. none of the above

15. A Clock waveform example is viewed using the tool:
○ a. the text editor,
○ b. the iverilog compiler,
○ c. the GTKWave waveform viewer

16. A waveform viewer can only be used with the RTL-Level designs:
○ a. True,
○ b. False.

17. The iVerilog compiler, as demonstrated, can produce:
○ a. a continuous-time waveform,
○ b. a continuous-amplitude waveform,
○ c. digital waveform files for use by a viewer,
○ d. none of the above

18. A repetitive clock signal can be viewed at the level of abstraction:
○ a. Behavioral Level,
○ b. RTL Level,
○ c. Structural Level,
○ d. any of the above

19. More than one level of abstraction can be used together in a program:
○ a. True,
○ b. False.

20. An RTL-Level program must use single-bit registers:

○ a. True,
○ b. False.

21. A group of Behavioral, RTL, and Structural-Level modules can be wired together:
○ a. True,
○ b. False.

22. Signals on wires within a design can be viewed using GTKWave:
○ a. True,
○ b. False.

23. A Structural-Level design can represent individual transistor devices:
○ a. True,
○ b. False.

24. A Structural-Level design can be constructed without needing a clock input:
○ a. True,
○ b. False.

25. The text-editor, compiler, and GTKWave viewer are the only FOSS tools available for Verilog:
○ a. True,
○ b. False.