100 Verilog Practice Questions (Beginner to Advanced)

Level 1: Beginner

1. Write a Verilog module for a 2-input AND gate.

2. Write a Verilog module for a 2-input OR gate.

3. Write a Verilog module for a 2-input XOR gate.

4. Write a Verilog module for a 2-to-1 multiplexer.

5. Write a Verilog module for a 4-to-1 multiplexer.

6. Write a Verilog module for a 1-bit full adder.

7. Write a Verilog module for a D flip-flop.

8. Write a Verilog module for a T flip-flop.

9. Write a Verilog module for a JK flip-flop.

10. Write a Verilog module for a 4-bit ripple carry adder.

11. Write a Verilog module for a 2-bit comparator.

12. Write a Verilog module for a 8-bit register with synchronous reset.

13. Write a Verilog module for a simple counter with enable.

14. Write a Verilog module for a binary to gray code converter.

15. Write a Verilog module for a gray to binary converter.

16. Write a Verilog module for a 8x1 multiplexer.

17. Write a Verilog module for a decoder (2-to-4).

18. Write a Verilog module for a encoder (4-to-2).

19. Write a Verilog module for a priority encoder.

20. Write a Verilog module for a half subtractor.

Level 2: Intermediate

21. Design a Verilog module that implements a 4-bit up-down counter.

100 Verilog Practice Questions (Beginner to Advanced)

22. Design a Verilog module that detects a sequence 1011.

23. Design a Verilog module that implements a finite state machine (FSM) with 3 states.

24. Design a Verilog module that counts the number of 1s in 8-bit input.

25. Design a Verilog module that rotates an 8-bit register left by 2 bits.

26. Design a Verilog module that counts the number of rising edges on input signal.

27. Design a Verilog module that debounces a button press.

28. Design a Verilog module that implements a 3-bit LFSR.

29. Design a Verilog module that performs 8-bit signed multiplication.

30. Design a Verilog module that detects overflow in a 4-bit adder.

31. Design a Verilog module that implements a barrel shifter.

32. Design a Verilog module that converts BCD to binary.

33. Design a Verilog module that converts binary to BCD.

34. Design a Verilog module that detects parity in 8-bit data.

35. Design a Verilog module that has parameterized bus width.

36. Design a Verilog module that models a traffic light controller.

37. Design a Verilog module that implements a 4-bit ALU.

38. Design a Verilog module that supports 4 instructions of a custom processor.

39. Design a Verilog module that has dual edge triggering logic.

40. Design a Verilog module that calculates population count of 16-bit input.

Level 3: Advanced

41. Design a Verilog system that implements a pipelined 4-stage processor.

42. Design a Verilog system that performs UART communication.

43. Design a Verilog system that implements a simple cache (direct-mapped).

100 Verilog Practice Questions (Beginner to Advanced)

44. Design a Verilog system that supports interrupt handling.

45. Design a Verilog system that implements an FSM-based vending machine.

46. Design a Verilog system that performs matrix multiplication (2x2).

47. Design a Verilog system that uses clock gating for power optimization.

48. Design a Verilog system that synchronizes signals between clock domains.

49. Design a Verilog system that implements an I2C master module.

50. Design a Verilog system that implements an SPI slave module.

51. Design a Verilog system that models a simple RISC-V ALU.

52. Design a Verilog system that has AXI-lite interface.

53. Design a Verilog system that has Wishbone bus interface.

54. Design a Verilog system that contains self-checking testbench.

55. Design a Verilog system that uses assertion-based verification.

56. Design a Verilog system that implements a CORDIC algorithm.

57. Design a Verilog system that performs FFT on 8-point data.

58. Design a Verilog system that schedules tasks using round-robin logic.

59. Design a Verilog system that implements a real-time clock with alarms.

60. Design a Verilog system that communicates with a temperature sensor (digital protocol).