FROM BASICS TO ADVANCED:

30

SYSTEM VERILOG INTERVIEW QUESTIONS FOR ENGINEERS

www.linkedin.com/in/jhansi-bendi-2061b5206 jhansibendi1711@gmail.com

1. **What is SystemVerilog?**
   SystemVerilog is an extension of Verilog that adds support for design and verification. It includes advanced constructs for object-oriented programming, assertions, randomization, and interfaces.

2. **What are the differences between Verilog and SystemVerilog?**
   - SystemVerilog adds logic data type (replaces reg and wire).
   - Supports OOP concepts like classes, inheritance, and polymorphism.
   - Adds constructs like interface, assertions, and randomization.

3. **What is the logic data type in SystemVerilog?**
   The logic data type is a 4-state variable (0, 1, X, Z) used for both combinational and sequential logic, replacing reg and wire.

4. **What is a module?**
   A module is a structural element in SystemVerilog used to describe hardware. It can contain variables, instances of other modules, and procedural blocks like always.

5. **What is a class in SystemVerilog?**
   A class is a blueprint for creating objects and encapsulates data (properties) and methods (functions/tasks).

6. **What is inheritance in SystemVerilog?**
   Inheritance allows a class (child) to derive properties and methods from another class (parent).

```
1 class Parent;
2   int a;
3 endclass
4 class Child extends Parent;
5   int b;
6 endclass
```

7. **What is polymorphism?**
   Polymorphism enables a base class handle to point to derived class objects, allowing runtime method binding.

```
class Base;
virtual function void show();
endfunction
endclass
class Derived extends Base;
function void show();
    $display("Derived");
endfunction
endclass
```

8. **What are virtual methods?**
   Virtual methods allow runtime overriding of methods in derived classes, enabling polymorphism.

9. **What is randomization in SystemVerilog?**
   Randomization generates random values for variables to improve test coverage. SystemVerilog uses rand or randc variables for this.

10. **What are constraints in SystemVerilog?**
    Constraints specify rules or conditions for randomization.

```
class Example;
rand int a;
constraint c { a > 5; }
endclass
```

11. **What is the difference between rand and randc?**

- rand: Generates random values within a range.
- randc: Generates cyclic random values, ensuring no repeats until all possibilities are exhausted.

12. What is constraint overriding?

Constraint overriding is changing or modifying constraints in a derived class or test.

13. What is an assertion in SystemVerilog?

Assertions are used to validate design behavior during simulation. There are two types:

- **Immediate Assertions**: Checked immediately during execution.
- **Concurrent Assertions**: Monitors signal behavior over time.

14. What is the difference between assert and assume?

- assert: Verifies a condition during simulation.
- assume: Assumes a condition is true during formal verification.

15. How are covergroups used in SystemVerilog?

Covergroups measure functional coverage by tracking variable values during simulation.

```
1 covergroup cg;
2   coverpoint var;
3 endgroup
```

16. What is UVM?

Universal Verification Methodology (UVM) is a standardized framework for functional verification using SystemVerilog.

17. What is a virtual interface?

A virtual interface is a pointer to a physical interface, used to decouple testbenches from DUTs.

18. What is a factory in UVM?

The UVM factory allows dynamic creation and substitution of objects during simulation.

19. What is a sequence in UVM?

A sequence generates and drives stimulus to the DUT, often through a sequencer.

20. What is a phase in UVM?

Phases manage simulation flow in a UVM testbench, such as build_phase, run_phase, and shutdown_phase.

21. What is the difference between always_ff, always_comb, and always_latch?

- always_ff: Sequential logic.
- always_comb: Combinational logic.
- always_latch: Latches.

22.What is the difference between interface and modport?

- interface: Groups related signals together.
- modport: Specifies access directions (input/output/inout) for the signals.

23.What is the initial block used for?
Executes code once at the start of simulation. It is not synthesizable.

24.What are coverage bins?
Bins in covergroups define the range or specific values of variables to track for coverage.

25.What is DPI in SystemVerilog?
The Direct Programming Interface (DPI) allows integration of C/C++ code with SystemVerilog simulations.

26.How are queues different from arrays?
Queues are dynamic and grow/shrink, while arrays are fixed in size.

27.What is the typedef keyword?
Defines a new name for an existing data type.

```
1 typedef logic [7:0] byte_t;
```

28.What are program blocks?
Used in testbenches for synchronization with clocking events and to avoid race conditions.

29.What is functional coverage?
Functional coverage ensures that all possible scenarios or features in a design have been exercised.

30.What is the use of clocking blocks?
Clocking blocks group signals driven or sampled relative to a clock edge, simplifying timing in testbenches.