INTERVIEW QUESTIONS

VERILOG

PART 3

JAIRAJ MIRASHI
DESIGN VERIFICATION ENGINEER

1. **What is the difference between $stop and $finish tasks in functions?**

The $stop task immediately terminates the simulation and can be used to halt the simulation at a particular point. The $finish task also terminates the simulation, but it allows any remaining simulation time to be completed before the simulation stops. This can affect the execution of other scheduled tasks and processes.

2. **Difference between parameter and define?**

Parameters are used to declare values that can be set when instantiating modules, and they can be overridden at different levels of hierarchy. Defines, on the other hand, are simple text substitutions and are often used to declare constants that are replaced during preprocessing. Parameters offer more flexibility and can be redefined at different levels, while defines are global and lack this hierarchical scoping.

3. **What is the difference between the following two statements?**

@(val == 2)

wait(val == 2)

@(val == 2) is a procedural statement that waits for the condition val to become equal to 2 before proceeding. It's typically used in testbenches for synchronization.

wait(val == 2) is also a procedural statement used in SystemVerilog's wait statement construct. It suspends the execution of the current process until the condition val is satisfied.

4. **What are the differences while specifying parameters using the defparam construct vs. specifying during instantiation?**

The defparam construct is used to override parameter values after module instantiation, and it applies to all instances of the module. Specifying parameters during instantiation sets the parameter values during instantiation and is more modular, allowing different instances to have different parameter values.

5. Difference between vectored and scalar nets?

Vectored nets represent multiple signals as a single entity, allowing efficient representation of bus signals. Scalar nets represent a single signal, making them suitable for individual signal connections. Vectored nets are denoted with [ ] brackets, while scalar nets are not.

6. Difference between real and realtime?

real is a data type used to represent floating-point values. realtime is a SystemVerilog type used for simulation time expressions. It ensures that the operations involving realtime values are performed in the order specified by simulation time, regardless of the order of execution in the procedural code.

7. What is the difference between arithmetic and logical shift registers?

Arithmetic shift registers perform shifts while considering the sign bit, which means they preserve the sign of the number during shifts. Logical shift registers perform shifts without considering the sign bit, effectively shifting in zeros.

Let's consider the binary number 1101 (13 in decimal) and perform both arithmetic and logical right shifts by 1 bit.

Arithmetic Right Shift:

In an arithmetic right shift, the sign bit is preserved. If the original number is positive (sign bit 0), zeros are shifted in from the left. If the original number is negative (sign bit 1), ones are shifted in from the left.

Original: 1101

After Arithmetic Right Shift by 1: 1110

Logical Right Shift:

In a logical right shift, only zeros are shifted in from the left, regardless of the sign bit.

Original: 1101

After Logical Right Shift by 1: 0110

8. What is the difference between the following two registers?

reg [1:n] rega;

reg mema [1:n];

reg [1:n] rega; declares a single register named rega that is n bits wide. This single register can hold an n-bit binary value.

reg mema [1:n]; declares an array of registers named mema, where each element of the array is 1 bit wide. There are n separate 1-bit registers named mema[1], mema[2], ..., mema[n].

9. How are the above two handled in assignments, ports, functions, and tasks?

The way the two types of register declarations (reg [1:n] rega; and reg mema [1:n];) are handled in assignments, ports, functions, and tasks depends on their nature as a single register and an array of registers. Here's how they are treated in different contexts:

- Assignments:

For reg [1:n] rega;: You can assign an n-bit value to the entire register rega using standard assignment operators, e.g., rega = 4'b1100;.

For reg mema [1:n];: You can assign 1-bit values to individual elements of the array, like mema[1] = 1'b1; or mema[2] = 1'b0;.

- Ports:

For reg [1:n] rega;: In module ports, rega can be used as a single port of width n.

For reg mema [1:n];: Each element of the array mema can be used as separate ports in module declarations.

- Functions and Tasks:

For reg [1:n] rega;: You can use rega as an argument to functions or tasks, treating it as a single value.

For reg mema [1:n];: You can pass individual elements of the array mema as arguments to functions or tasks.

10. What is the difference between parameters and specparams?

Parameters are used for general values that can be overridden during instantiation. Specparams are used in assertion constructs like SystemVerilog's cover property to provide specific values for coverage analysis, separate from regular parameter overrides.

11. Is it possible to synthesize a for loop?

Yes, a for loop with fixed limits can be synthesized into hardware. It represents a repeated operation that can be unrolled during synthesis to generate multiple instances of the loop's body.

12. How is time advanced in simulation?

In simulation, time is advanced by executing events and processes in chronological order. Events scheduled at different simulation times are executed in the correct sequence to simulate the passage of time.

13. Name three methods of timing control?

Three methods of timing control are:

- Blocking Assignments (=): In a blocking assignment, the right-hand side expression is evaluated immediately, and the value is assigned to the left-hand side variable. The simulation process is blocked until the assignment is complete. This method is often used for combinational logic modeling.

Example:

a = b & c;

- Non-blocking Assignments (<=): Non-blocking assignments schedule the assignment to occur at the end of the time step. Multiple non-blocking assignments within a procedural block are executed in parallel, allowing modeling of sequential behavior. This method is often used for modeling flip-flops and sequential logic.

Example:

q <= d;

- Delay Statements: Verilog provides delay statements to control the timing of events. The delay value specifies how much time should elapse before the statement is executed. The delay can be specified using a numerical value followed by a time unit (such as #10ns for a 10 nanosecond delay).

Example:

#5ns a = b & c;

14. What is behavioral modeling used for?

Behavioral modeling is used to describe the functionality and behavior of a digital system without specifying the low-level implementation details. It focuses on what the system does rather than how it's implemented.

15. How do you define the states for an FSM?

Finite State Machines (FSMs) have states defined as discrete values. They are usually declared using an enumerated data type or binary encoding. Each state represents a unique condition that the FSM can be in.

16. What is the difference between force/release and assign/deassign?

force and release are used to drive values onto nets in a continuous assignment, and they are mainly used for testbench purposes. assign and deassign are used for modeling combinational logic and allow signals to be assigned or deassigned at different points in time.

17. What is the difference between posedge and negedge?

In edge-sensitive constructs like always @(posedge clk) and always @(negedge rst), posedge triggers on the rising edge (transition from 0 to 1), while negedge triggers on the falling edge (transition from 1 to 0).

18. What is the difference between $display and $write?

$display adds a newline character to its output by default, while $write does not. When used without arguments, $display prints a newline character, but $write without parameters prints nothing.

19. What is the difference between $display and $monitor?

Both $display and $monitor are used for printing information during simulation. However, $monitor automatically monitors variables and expressions for changes and displays them whenever they change, while $display requires explicit calls and arguments to display values.

20. What is the difference between $display and $strobe?

$display is the normal display, which executes its parameters wherever it is present in the code.

$strobe executes only once in a time instant, when all processes in that time instant have executed.