# Interview questions for system verilog

1. What is clocking block?

Ans: Clocking block can be declared using the keywords clocking and endclocking. A clocking block is mainly used in the testbench in order to avoid race conditions. Clocking blocks are used to assemble all the signals. They are useful in separating clocking activities from its main data activities. The declaration and instantiation of clocking block can both occur inside the module.

2. What are modports?

Ans: Modports define directions of ports as if they are declared inside the module. To restrict interface access within module modports are used. Modports can have inputs, outputs and inout ports also.

3. What are interfaces?

Ans: Interfaces enables the connectivity between the modules. It allows the smooth flow of design between the modules. The design can also be reused using interfaces. Using interfaces, we can replace a group of names with a single name which is helpful in reducing the size of program.

4. What are virtual interfaces? How can it be used?

Ans: A virtual interface can be used a reference to existing interface. It allows the subprogram to operate in parts of design.

5. What is a class?

Ans: Class is composed of set of members that describe how an instance of class or object is constructed and how it behaves.

```
Eg: class class name{
member1;
---------------}
object name;
```

6. What is program block?

Ans: Program block is mainly used in the writing of testbench. It is defined with a set of keywords, program and endprogram. It separates the testbench from the device under test. This block works well in the hardware description of the modules. Program block is also used to avoid race conditions.

7. What is a mailbox?

Ans: Mailbox is communication mechanism that allows the exchange of the messages between the processes. Mailboxes are helpful in the transmission and receiving of the data in a systematic way.

8. What are semaphores?

Ans: A semaphore is like a bucket and used in synchronization. They are inbuilt in systemveriolg. When a semaphore is allocated, the keys are allocated to each and every bucket. The number is fixed. The keys are not identical. Processes using semaphores must definitely have a key from bucket before they start the execution process.

9. Why is reactive scheduler used?

Ans: Code specified in program blocks and pass/fail code from property expressions are scheduled in reactive scheduler.

10. What are rand and randc?

Ans: The variables in the class can be declared random using the keywords: rand and randc. Dynamic and associative arrays can be declared using rand or randc keywords.

11. What is the difference between keywords: rand and randc?

Ans: Variables declared with rand keywords are standard random variables. Their values are uniformly distributed over their range. Values declared with randc keyword are randomly distributed. They are cyclic in nature. They support only bit or enumerated data types. The size is limited.

12. What is the use of always_ff?

Ans: The always_ff can be used to model sequential logic behavior. The always_ff is always synthesizable.

13. What are static and automatic functions?

Ans: For overriding the values in the class, static function is used. Whereas in automatic, when one task is called, two separate memories will be allocated.

14. What is the procedure to assign elements in an array in systemverilog?

Ans: Assigning arrays in systemverilog uses the replicate operators. Eg: int n[1:2][1:3]={{0,1,2),{3{4}}};

15. What are the types of arrays in systemverilog?

Ans: There are two terminologies associated with the arrays in systemverilog. Packed and unpacked arrays. When the dimensions of arrays are declared before the object name is referred to as ?packed arrays?. The ?unpacked array? term is used to refer when the dimensions are declared after the object name. The memory allocation of both packed and unpacked arrays also differs.
E.g.: int [7:0] c1; //packed array
reg r[7:0] //unpacked array

16. What are assertions?

Ans: An assertion specifies a behavior of the system. They a re primarily used to validate a behavior of design. Assertions can also be used toprovide functional coverage and generate input stimulus for validation.

17. What is the syntax for ## delay in assertion sequences?

Ans: Is called as cycle delay syntax: ## 2; fifo.wdata < =8?hAA; //wait for 2 default clocking cycles, then drive wdata.

18. What are virtual classes?

Ans: When we use the same function name in both the base and derived classes, the function in the base class is declared as virtual. The class ispreceded using the keyword virtual before its normal declaration.

19. Why are assertions used?

Ans: Assertions are mainly used to check the behavior of the design whether it is working correctly or not. They are useful in providing thefunctional coverage information .i.e. how good the test is and whether the design meets all the requirements for testing. There are mainly two types of assertions in systemverilog. They are: immediate assertions and concurrent assertions.

20. Explain the difference between data type?s logic and reg and wire.

Ans: Wire is basic data type which does not drive strength
a) wire is used for designing combinational logic. We can assign a single value to a wire by using the assign statement. Wire cannot store any data.
b) reg data type can be used for storage purpose. reg is used for designing both sequential and combinational circuits. reg data types can be driven from initial and always block .
c) logic data types are similar to reg data types .

21. What is callback?

Ans: A callback is a built in systemverilog task/function. Suppose if we are transmitting a data using mailbox and you are sending data fromdesign to transmitter. If you want to get back the data and you need the same data to put back in the scoreboard for comparison, this is called callback. Any change in the transmitted data can be achieved using the callback routine. Generally callbacks are called bfore transmission and after receiving the data.

22. What are the ways to avoid race condition between testbench and RTL using SystemVerilog?

Ans: Using program block and clocking block.

23. Explain event regions in systemverilog?

Ans: Active region: simulation of design codes in modules.
Observed region: checking the system verilog assertions.
Reactive region: executing the testbench code.
Postponed region: sampling the design signals for test bench inputs.

24. What are the types of coverages available in systemverilog?

Ans: Functional coverage and code coverage.

25. How can you detect a deadlock condition in FSM?

Ans: It can be detected using a property checking with assertion.

26. What is mutex?

Ans: A Mutex object only allows one thread into a controlled section. They cannot be executed by more than one thread in a concurrentmanner. A mutex can be released only through thread.

27. What is the significance of seed in randomization?

Ans: If we want to produce the same simulation results, seed is used. But the seed number should be same to perform the operation.

28. What is the difference between code coverage and functional coverage?

Ans: Functional coverage: Functional coverage determines how much functionality of the design has been exercised. Functional assertions areused to check whether each and every corner of the design is explored and functions properly. Code coverage: This will give

information about how many lines are executed, how many times each expression is executed. It describes the level up to which the code has been tested.

29. If the functional coverage is more that code coverage, what does it means?

Ans: code: High functional: High - You are most likely done
code: High functional: Low - Still need to run more simulation or improve test bench
code: Low functional: High - Probably missing functional coverage points
code: Low functional: Low - Still need to run more simulation or improve test bench

30. How we can have #delay which is independent of time scale in system verilog?

Ans: We can mention it as #5ns.

31. What are constraints in systemverilog?

Ans: By specifying constraints, users can easily create tests that are very difficult to reach corners of the design. Sytemverilog allows users tospecify constraints in compact and declarative way. Random values are generated to meet the constraints.

32. What are the different types of constraints in systemverilog?

Ans: Random constraints, user defined constraints, inline constraints, if-else constraints and global constraints.

33. What is an if-else constraint?

Ans: If the given expression is true, all the constraints in the first constraint block should be satisfied, otherwise all of the constraints in theelse constraint block will be satisfied.

34. What is inheritance and give the basic syntax for it?

Ans: A derived class by default has the properties and methods of the base class or parent class. This is called inheritance.
Syntax: class Derived extends BaseClass;
// method declarations.
endclass

35. What is the difference between program block and module?

Ans: The module is the basic building block in verilog which is used in creating a design. Systemverilog adds a new block called program blockwhich can be declared using the keywords program and endprogram. The program block separates the design and testbench. The program block and module differs in syntax also.

36. What is final block?

Ans: Systemverilog adds a final block that executes the end of simulation. There is nothing to be executed after the final block. It is the end ofsimulation. Final block should not have any delays.

37. What are dynamic and associative arrays?

Ans: When array is declared, memory is allocated for the elements of array when the program starts. Memory is allocated to the array only whenthe arrays are declared in the program. The memory remains allocated during the lifetime of the program. This is called static array. It may sometimes happen that we don?t know how large an array we will need. In this case, it is convenient to allocate an array while the program is running. This is called dynamic array. Dynamic array is a one dimensional array.

38. What is an abstract class?

Ans: Abstract classes are those which can be used for creation of handles. Their methods and constructors and can be used by the class orextended class. They are used to generalize the super class from which child classes can share its methods.

39. What is the difference between $random and $urandom?

Ans: $random and $random are both system functions.
1. $random returns a 32-bit signed random number whenever it is added.
2. $urandom returns a 32-bit unsigned random number whenever it is added. It a newly added system function in systemverilog.

40. What is the use of $cast?

Ans: Systemverilog provides $cast system task to assign values to variables that may not be valid because of differing data type. $cast can becalled as either a task or a function. $cast is basically used in dynamic casting. For more accurate checking of values, $cast is used.
Syntax: $cast(des_val, eval_expr)

41. What is the difference between mailbox and queue?

Ans: Mailbox is similar to a queue, which allows only atomic operations. They can be bounded/unbounded. Get/put task is used to suspend abounded mailbox. That?s why mailbox is used more for communication between threads. Queues are large structures. Inserting data in queues is very difficult.

42. What are bidirectional constraints?

Ans: Constraints in systemverilog are bidirectional by default. That means they do not follow any sequence in which the constraints arementioned. All the variables are looked simultaneously.

43. What is circular dependency and how to avoid this problem?

Ans: Over specifying the solving order might result in circular dependency. There is no solution for circular dependency. The constraint solvermight give error/warning or no constraining.

44. What is the significance of super keyword?

Ans: The super keyword is used from the derived class to refer to the members of the derived class. If we want to access members of derivedclass that are overridden by the derived class, super keyword is used. Super keyword cannot be accessed directly.

45. What is the significance of this keyword?

Ans: This keyword is used to call the parameters of same class.

46. What are input and output skews in clocking block?

Ans: Skews are numbers that are indicated before the input and output ports. A skew number indicated before an input defines that the inputis sampled before the clocking event occurs, ie a posedge or negedge occurs. A skew number in the output indicates that the output is synchronized and then sent to the clocking blocks. A skew must be a constant expression. It can also be specified as a parameter.

47. What is a scoreboard?

Ans: Dynamic data types and dynamic memory allocations in systemverilog makes us easy to write scoreboards. Scoreboard is used to store theexpected output of the device under test. It implements the same functionality as DUT. It uses higher level of constructs.

48. Mention the purpose of dividing time slots in systemverilog?

Ans: The main purpose of dividing the time slots in systemverilog is to provide interaction between the design and the testbench.

49. What is static variable?

Ans: In systemverilog we can create a static variable inside the class. But this variable has a limited scope. The variable declared static is sharedamong the other variables in the class. A static variable is usually instantiated inside the declaration.

50. In simulation environment under what condition the simulation should end?

Ans: 1) Packet count match
2) Error
3) Error count
4) Interface idle count
5) Global Time out

51. What is public declaration?

Ans: Objects in systemverilog can be declared as public and private. Public declarations are accessible from outside that object, i.e. they areaccessible by the users. By default declarations in systemverilog are public.

52. What is the use of local?

Ans: In order to make the declarations private, local attribute is used. Once the data is declared as local, it can be accessed on in the particularclass.

53. Difference b/w logic & bit.

Ans: Logic is 4 state data type where as bit two state.

54. How to take an asynchronous signal from clocking block?

Ans: By Using modports.

55. What is fork-join, types and differences?

Ans : There are three types of fork.. join blocks
fork.. join: The parent process blocks until all the processes spawned by this fork complete.
fork .. join_any: The parent process blocks until any one of the processes spawned by this fork Complete.
fork?join_none: The parent process continues to execute concurrently with all the processes Spawned by the fork.
The spawned processes do not start executing until the parent thread executes a blocking statement.

56. Difference between final and initial blocks?

Ans: Initial block we can schedule an event but in final block we can not schedule an event. Initial block we can insert delays but in final block we cant insert delays.

57. What are the different layers in Testbench?

Ans: In SystemVerilog based verification environment, the test environment is divided into multiple layers as shown below.
Test layer ? Tests
Scenario layer ? Generator
Functional layer ? Transactor, Self checker, Checker

Command layer ? Driver, Assertions, Monitor
Signal layer ? DUT

58. What is the use of modports?

Ans: An interface can have multiple view points. For example master, slave etc..
These can be specified using modports.
Ex: modport ahb_slave_mp (clocking ahb_slave_cb);
modport ahb_master_mp (clocking ahb_monitor_cb);

59. What is the use of package?

Ans: In Verilog data/function/task within modules are specific to the module only. They can't be shared between two modules. Package is aconstruct of SystemVerilog aims in solving this problem. It allows global data/task/function declarations which can be used across modules.

60. What is the difference between bit [7:0] and byte?

Ans: byte is signed whereas bit [7:0] is unsigned.

61. What is chandle in systemverilog ?

Ans. Chandle is a data type used to store pointers passed from DPI. Size of the chandle is machine dependent. Initialized value of the chandle isnull. It can be used to pass arguments to functions or tasks.

62. What are the features added in systemverilog for function and task?

Ans: Begin and end not required.Function can have any number of input, output and inouts including none. Return can be used in task. Function return can have void return type.

63. What is DPI in systemverilog?

Ans: DPI (Direct Programming Interface) is used to call C, C++, System C functions.

64. What is inheritance?

Ans: Extending the functionality of the existing class. It is mainly to reuse the existing code. Common properties can be grouped into one class.

65. What is polymorphism?

Ans: Wait until run time to bind data with functions.

66. What is Encapsulation?

Ans: Binds data and function together.

67. How to count number of elements in mailbox?

Ans: Mailbox is used for communication between two processes.
Ex: mailbox mbx;
mbx = new(); // Allocate mailbox
mbx.put (data); //Put data object in mailbox
mbx.get (data); // Data updated with new data from FIFO
mbx.try_get(data); // Non-Blocking Version
mbx_peek(data); // Looks for data but wont remove
count = mbx.num(); // To count number of elements is mailbox

68. What is covergroup?

Ans: Captures result from a random stimulation.
Encapsulates the coverage specification.
Ex: enum { red, green, blue } color;
bit [3:0] pixel;
covergroupg1 @ (posedgeclk);
coverpoint color;
coverpoint pixel;
AxC: cross color, pixel;
endgroup

69. What are super, abstract and concrete classes?

Ans: Super class is the class from which child classes can share its methods.Abstract class is the class for which we create handles. Sub classes of abstract classes are concrete classes.

70. Explain some coding guidelines you followed in your environment ?

Ans: Allocate memory for mailbox at environment level.

71. What is Verification plan? What it contains?

Ans : Creating the real time environment for the (DUT ) to check its performance in real time. The verification plan closely tied with the hardware specification and contains a description of what features need to be verified.
Contains:
Directed testing
Random testing
Assertion
Hardware and software co-verification
Use of verification IP

72. Explain how messages are handled?

Ans: Using mailbox.

73. What is difference between define and parameter?

Ans: The parameter value can be changed during execution but not in the case of define construct.

74. Why ?always? block not allowed inside program block?

Ans : always block might execute on every posedge of clock. Even when program block is terminated, an always block which runs continuously isredundant. To stop an always block, $exit should be explicitly called. But initial forever could be used instead.

75. How too implement clock in program block?

Ans : initialforever #5 clk <= ~clk;

76. How to kill process in fork/join ?

Ans : Threads can be terminated at any time using a keyword disable.
Disable will be used with a label assigned to fork statement.

77. Difference between Associative and dynamic arrays?

Ans: Associative arrays basically used for very large memories. They have feature of string indexing. They executes at compile time.

Used for large memories. Value is assigned to dynamic array at run time. New constructor is used to initilize the size of dynamic array.

78. How to check whether randomization is successful or not?

Ans : if (!obj.randomize())
begin
$display(?Error in randomization?);
$finish;
end

79. What is property in SVA?

Ans: 1. Number of sequences can be combined logically or sequentially to create more complex sequences. SVA provides a key word torepresent these complex sequential behaviors called "property."
2. The basic syntax of a property is as follows.
**property** name_of_property;
< test expression > **or**
< complex **sequence** expressions >
**endproperty**;

80. What advantages of Assertions?

Ans: In traditional verification approach we will inject random stimulus into the DUT and checks result at output. For complex designs coverageand debugging is harder.
Assertions come here to improve the verification process.
Increases bug detection possibility at RTL level. Reduces time to develop.
Great help in debug for large nightmare design random tests.

81. What are immediate Assertions?

Ans: Immediate assertion is basically a statement that something must be true, similar to if statement.
If assert evaluates to X, Z or 0, then the assertion fails and the simulator writes an error message.
my_assert:**assert**(condition1 & condition2)
$display("Passed..");
**else**
$error("Failed..");

82. What are Assertion severity system level task? What happens if we won?t specify these tasks?

Ans : When we set property and if we wont specify failure case of the property, then by default language dictates simulator should give error as$error severity level.
$fatal - Run time fatal (quit Simulation)
$error - Run time error. Default according to LRM 3.1a. Vendor specific line commands can change this behavior.
$warning ? Run time warning
$info ? Means this assertion carries no specific severity.

83. What is difference between Concurrent and Immediate assertions?

Ans; Immediate assertion describes a logic behavior at an instant of time, where as concurrent assertion detects behavior over time to bespecified.

84. In which event region concurrent assertions will be evaluated?

Ans: The variables used in a concurrent assertion are sampled in the Preponed region of a time slot and the assertions are evaluated during theObserve region. Both these regions occur immediately before a clock edge.

85. What are the main components in Concurrent Assertions?

Ans: In concurrent assertion there are three main components.
Sequence
Property
Assert ? property

86. What is Consecutive Repetition Operator in SVA?

Ans : Consecutive Repetition Operator [* ]
It is to specify that a signal or sequence to match continuously for the number of specified clocks.
Syntax: signal or sequence [* n]
Where "n" is the number of times the expression should match repeatedly.

87. What is goto Replication operator in SVA?

Ans : This operator specifies that an expression will match the number of times specified not necessarily on continuous clock cycles.
Ex: x [->4:7]
x has been true 4, 5, 6 or 7 times, not necessarily on consecutive clocks.

88. What is difference between x [->4:7] and x [=4:7] in SVA?

Ans :

| | |
|---|---|
| x [->4:7] | x has been true 4, 5, 6 or 7 times, not necessarily on consecutive clocks |
| x [=4:7] | x has been true 4,5,6 or 7 times, once again not necessarily on consecutive clocks, and with possible additional clocks after words when x is not true. |

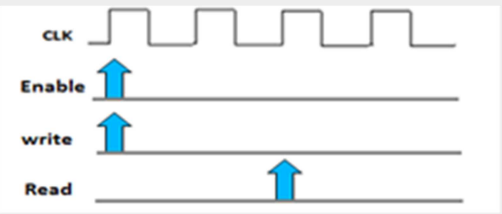
89. What are implication operators in Assertions?

Ans : Implication operators only used inside the property.
Two types of operators

1. Overlapping ( )

```
property pr1;
  always @(posedge clk) enable |-> (write ##2read);
endproperty

asr1:assert property(pr1);
```



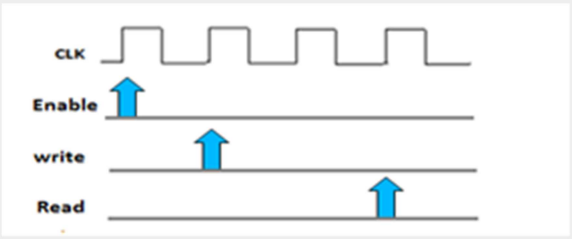
2. Non Overlapping ( )

```
property pr1;

  always @(posedge clk) enable |=> (write ##2read);

endproperty


asr1:assert property(pr1);
```




90. Can a constructor qualified as protected or local in systemverilog?

Ans : local

91. What are advantages of Interfaces?

Ans : a. Interface is ideal for design reuse.
b. It reduces the possibility of signal misconnection when the number of signals are large.
c. Easy to add new signals to the design

92. How automatic variables are useful in Threads?

Ans : Some times we use loop that spawns threads and we don?t save variable values before the next iteration.
We should use the automatic variables inside a fork?.join statement to save the copy of a variable.
Key word automatic create a copy of variable in each loop.