1. Explain enum with an example.

An enum in SystemVerilog is a user-defined data type that allows you to define a set of named integer constants. Enums improve code readability and maintainability by using symbolic names instead of raw numeric values.

Example:

typedef enum {IDLE, START, RUN, STOP} state\_t;

state\_t current\_state;

initial begin

current\_state = START; // Assigning an enum value

$display("Current state: %0d", current\_state);

end

1. What are the different functions in System Verilog for an enum?

SystemVerilog provides some functions to work with enums:

* $type(): Returns the name of the type.
* $bits(): Returns the number of bits used by the enum type.
* $isunknown(): Checks if the enum value is unknown (used with uninitialized variables).
* $enum(): Returns the string representation of the enum value (if it's defined).

1. What is a string in System Verilog?

In SystemVerilog, a string is a dynamic array of characters that can hold text data. Strings are used for text manipulation and debugging. Strings in SystemVerilog are null-terminated and can dynamically grow or shrink.

Syntax:

string str = "Hello, World!";

1. What are the different functions in System Verilog for string?

* $sformatf(): Formats a string similarly to sprintf() in C.
* $size(): Returns the length of a string.
* $sscanf(): Reads data from a string using a format string.
* $stohex() and $stoint(): Convert strings to hex or integer values.

1. What is the difference between time & realtime?

* time: A 32-bit unsigned integer representing the simulation time in nanoseconds (from 0 to 2^32-1).
* time is integer-based, representing time in integer nanoseconds.
* realtime: A 64-bit signed real value representing simulation time in seconds (can store both very large and fractional time values).
* realtime is floating-point based, providing greater precision and allowing fractional values.

Example:

time t = 100; // Integer time in nanoseconds

realtime rt = 100.5; // Real time in seconds (with fractional part)

1. How to write floating-point and exponential numbers in System Verilog? Explain with an example.

* Floating-point numbers: real r = 3.14159; // A floating-point number
* Exponential notation: real exp\_num = 1.23e4; // 1.23 \* 10^4

1. Explain void data type with an example.

The void data type represents the absence of a return value. It is often used in functions that do not return any value.

Example:

function void print\_hello();

$display("Hello, World!");

endfunction

initial begin

print\_hello(); // Calls function that does not return a value

end

1. How can I convert a real data type to an int?

* You can use type casting or $int() to convert a real value to an int.

Example:

real r = 3.14;

int i = r; // Implicit conversion

$display("Integer value: %0d", i); // Displays: 3

* Using $int():

Example:

real r = 3.99;

int i = $int(r); // Explicit conversion

$display("Integer value: %0d", i); // Displays: 3

1. What is the use of event data type in SV?

The event data type is used for synchronization between processes. It is a lightweight primitive that allows threads to signal each other and wait for specific events to occur.

Example:

event e;

initial begin

-> e; // Trigger the event

end

initial begin

@e; // Wait for the event to be triggered

$display("Event triggered!");

end

1. What is the use of parameter data type in SV?

A parameter in SystemVerilog is a constant value used at compile-time. It is typically used for configuring hardware behaviour or controlling simulation parameters.

Example:

module example;

parameter WIDTH = 8; // Parameter with a default value

reg [WIDTH-1:0] data; // Use parameter in the module

endmodule

1. Create a struct packet with the following fields:
   1. 32-bit address
   2. 64-bit data
   3. 1-bit valid

typedef struct {

logic [31:0] address; // 32-bit address

logic [63:0] data; // 64-bit data

logic valid; // 1-bit valid flag

} packet\_t;

packet\_t pkt; // Declare a packet instance

initial begin

pkt.address = 32'hA5A5A5A5; // Example address

pkt.data = 64'hDEADBEEF12345678; // Example data

pkt.valid = 1; // Set valid flag to 1

$display("Address: %h, Data: %h, Valid: %0d", pkt.address, pkt.data, pkt.valid);

end