

DAY 57 CHALLENGE: 100 DAYS VERIFICATION CHALLENGE

Topic: System Verilog Data types – Arrays, *typedef*, *struct*, *enum*, *union*, *String*, *Event*

1. What are fixed size arrays in SV? Explain it's applications.
2. How to find indexes associated with associative array items?
3. Why is reactive scheduler used?
4. Explain the use of *typedef* with an example
5. What is *struct*?
6. Explain *Union* with an example.
7. Explain *enum* with an example
8. What are different functions in System Verilog for an *enum*?
9. What is a *String* in System Verilog?
10. What are different functions in System Verilog for *String*?
11. What is the difference between *time* & *realtime*?
12. How to write floating point & exponential numbers in System Verilog?
Explain with an example?
13. Explain *void* data type with an example.
14. How can I convert a *real* data type to an *int*?
15. What is the use of *Event* data type in SV?
16. What is the use of *parameter* data type in SV?
17. Create a *struct* packet with following fields:
 - 32-bit address
 - 64-bit data
 - 1 bit valid

Topic: System Verilog DATA Types: ARRAYS,

Typeclaf, struct, enum, union, string, Event

Solⁿ ① what is fixed size array in SV? Explain it's application?

- fixed size array is same as verilog array or fixed size array is nothing but the size of array is fixed & is unpacked array type.
- Data storage & external buffer implementations are some of the application of fixed size array.
- Once the array is declared no need to create it. By default array will be initialized with value '0'.

Solⁿ ② How to find indexes associated with associative array items?

- The Syntax of associative array is "data type array name [indexes of array items]" inside bracket the indexes represents the data types of array items.

Solⁿ ③ why is reactive Scheduler used?

- The reactive Scheduler is used to:
 - i) execute all program Blocking assignments.
 - ii) Execute the pass/fails code from Concurrent assertions.
 - iii) Evaluate the RHS of all program non-Blocking assignments.
 - iv) Execute all program Continuous assignments.
 - v) Execute the \$exit & implicit \$exit Commands.

Solⁿ ④ Explain the use of typedef with an example.

- Basically typedef is used to create a user defined data type.
- typedef datatype is used to convert a variable into specific datatype.

Solⁿ⑤ What is struct?

- Struct is a collection of elements having different data types.
- These elements can be accessed as a whole or individually, like enum. These can be used as to make user-defined data types.
- When a struct is declared, the memory is allocated for all the elements.
- There are two types of struct packed & unpacked struct.

Solⁿ⑥ Explain union with an example.

- Union are like struct, but in union, we can only access one element at a time.
- When unions are declared the memory is allocated only for the largest data type.
- Not all simulators supports unions. check the simulator manual before using unions.

Solⁿ⑦ Explain enum with an example.

- enum is a special data type introduced in System Verilog.
- This is user-defined data type which means that a new data type can be created using enum.
- enum is a data-type which assigns names to the integer constants.
- enum is used to simplify the code & make it easier to read & manage.

Solⁿ⑧

What are different functions is SV for an enum?

- enum has different built in functions like that can be used iterate within elements of the enum:
- | | | |
|-----------|----------|----------|
| ① first() | ② last() | ③ prev() |
| ④ next() | ⑤ num() | ⑥ name() |

Solⁿ ⑨ what is a string in System Verilog?

- in Verilog that there is no data type to store string.
- instead, reg data type was used to store string.
- in System Verilog, The string data type can be used to store string.
- string is a particular data type whose size changes dynamically during run time.
- an ordered of characters is called a string data type.

Solⁿ ⑩ what are different functions in System Verilog for string?

- String methods are System functions that specifically work with the string data type. These methods can be used to perform various operations on a string.

- ① len()
- ② putc
- ③ getc
- ④ Substring()
- ⑤ toupper()
- ⑥ tolower()

Solⁿ ⑪ what is the difference Btw time & real time?

- Time data type is used to represent simulation time means \$time returns the current simulation time as a 32-bit unsigned integer.
- The real time data type represents time as a floating point number. This is valuable when there's a need to represent fractional time values, especially useful in mixed signal simulations where analog values evolve continuously.

Solⁿ ⑫ Explain void data type with an example.

- The void data type represents non-existent data.
- If the function doesn't return any data type when its called, then it could be a void function.
- mean void is used to indicate null return from the function.

Solⁿ④ How can i convert a real data type to an int?

- It can be done using the static casting into;

`a = int(3.14);`

- `$toi` : Converts a real number to an integer. It truncates the real numbers to form an integer.

Eg: `reg [63:0] van1;`

`real van2;`

`van2 = $bits to real (van1);`

integer

Solⁿ⑤ What is the use of event data type in System Verilog?

- In System Verilog, basic operation of event remains the same as that of the Verilog, but some enhancement are done in SV.
- Event are a static object which can be used to synchronize 2 running process parallelly.
- The handle of the event can be used to trigger the event.
→ is used to trigger an event whereas @ or wait() to wait for the event to be triggered.

Eg: `event e;`
 `→ e;`

- The main purpose is to use the event synchronize the process.

Solⁿ⑥ What is use of parameter data type in SV?

- Parameter data types is used to declare the constant variable inside the local class we can override the parameter using `defparam` keyword or when instantiation of class can override it.

Ques:17 Create a "struct" packet with following fields

Brought to you by
DOULOS

▼ Languages & Libraries

Testbench + Design

SystemVerilog/Verilog

UVM / OVM

None

Other Libraries

None
OVL 2.8.1
SVUnit 2.11

Enable TL-Verilog

testbench.sv

```
1 // Code your testbench here
2 // or browse Examples
3 module tb;
4     typedef struct {
5         bit [31:0]address;
6         bit [63:0]data;
7         bit valid;
8     }packet;
9
10    packet pkt;
11
12    initial begin
13        pkt = '{ 45, 52, 21};
14        $display("\tpacket data=%p",pkt);
15    end
16 endmodule
17
```

-timescale 1ns/1ns

Run Options

-voptargs+=acc=npr

Run Time: 10 ms

Use run.do Tcl file

Use run.bash shell script

Open EPWave after run

Show output file after run

Download files after run

Examples

Done

eLog Share

```
# // 18 U.S.C. Section 1905.
# //
# Loading sv_std.std
# Loading work.tb(fast)
#
# vsim -voptargs+=acc=npr
# run -all
#     packet data='{address:45, data:52, valid:1}
# exit
# End time: 13:48:42 on Dec 17,2023, Elapsed time: 0:00:01
# Errors: 0, Warnings: 0
```

Ques:1 Example of Static Array in System Verilog .

- In Verilog we have seen that only static arrays can be created.
- Static arrays has a major drawback as the size of the arrays once defined cannot be changed. This wasted a lot of memory space as at times the entire size of the array is not used.

Brought to you by
DOULOS

Languages & Libraries

Testbench + Design

SystemVerilog/Verilog

UVM / OVM

None

Other Libraries

None

```
1 module tb;
2     bit[4:0] Array1[4:0];
3     bit[4:0] Array2[4:0];
4
5 initial begin
6     Array1 = {2, 4, 5, 8, 9};
7     $display("\tArray1=%p",Array1);
8     Array2 = Array1;
9     $display("\tArray2=%p",Array2);
10 end
11 endmodule
```

Compile Options

Run Options

Run Time: 10 ms

Use run.do Tcl file

Use run.bash shell script

Open EPWave after run

Show output file after run

Download files after run

Examples

Log

```
# //
# Loading sv_std.std
# Loading work.tb(fast)
#
# vsim -voptargs=-acc=npr
# run -all
#      Array1='{2, 4, 5, 8, 9}
#      Array2='{2, 4, 5, 8, 9}
# exit
# End time: 11:46:26 on Dec 17, 2023, Elapsed time: 0:00:01
# Errors: 0, Warnings: 0
```

Ques:4 Example of "typedef"

Brought to you by
DOULOS

Languages & Libraries

Testbench + Design

SystemVerilog/Verilog

UVM / OVM

-voptargs=-acc=npr

Run Time: 10 ms

Use run.do Tcl file

Use run.bash shell script

Open EPWave after run

Show output file after run

Download files after run

Examples

Done

```
1 module type_def;
2     typedef bit [2:0]packet;
3     packet pkt =3'd7;
4     initial
5         begin
6             $display("\tpacket=%b",pkt);
7         end
8     endmodule
```

```
# //
# Loading sv_std.std
# Loading work.type_def(fast)
#
# vsim -voptargs=-acc=npr
# run -all
#      packet=111
# exit
# End time: 12:04:37 on Dec 17, 2023, Elapsed time: 0:00:02
# Errors: 0, Warnings: 0
```

Ques:5 Example of Struct:

- Struct and unions are special data types which are used to group variables having different data types.
- Similarly, arrays also provide the grouping of various variables or elements. But in arrays, the data type of each element is the same,
- whereas struct and unions can have elements having different data types.

```
typedef struct {           example struct {  
    element1;             int roll;  
    element2;             string name;  
}name;                      } student;
```

The screenshot shows the DOULOS Testbench + Design software interface. On the left, there's a sidebar with 'Brought to you by DOULOS' and sections for 'Languages & Libraries' (Testbench + Design, SystemVerilog/Verilog, UVM / OVM, Other Libraries), and 'Run Options' (timescale, -voptargs, Run Time: 10 ms, checkboxes for run.do, run.bash, EPWave, output file, download files, Examples). The main area is titled 'testbench.sv' and contains the following SystemVerilog code:

```
module tb;
  typedef struct{
    int roll;
    string name;
    int mark;
  }student;
  student s1,s2;
  initial begin
    s1 = '{ 5,"Arti tyagi",50};
    $display("Student Record s1=%p",s1);
    s2 = s1;
    $display("\tStudent Record s2=%p",s1);
    $display("\tStudent Record s2=%p",s2);
  end
endmodule
```

The screenshot shows the DOULOS Testbench + Design software interface with a 'Log' tab selected. The log window displays the following text from the simulation:

```
# Loading sv_std.std
# Loading work.tb(fast)
#
# vsim -voptargs=-acc=npr
# run -all
# Student Record s1='{roll:5, name:"Arti tyagi", mark:50}
#     Student Record s2='{roll:5, name:"Arti tyagi", mark:50}
#     Student Record s2='{roll:5, name:"Arti tyagi", mark:50}
# exit
# End time: 12:46:15 on Dec 17,2023, Elapsed time: 0:00:01
# Errors: 0, Warnings: 0
```

Ques:6 Union with example.

Brought to you by
DOULOS

▼ Languages & Libraries

Testbench + Design

SystemVerilog/Verilog

UVM / OVM

None

Other Libraries

None
OVL 2.8.1
SVUnit 2.11

Enable TL-Verilog

Enable Easier UVM

-timescale 1ns/1ns

Run Options

-voptargs=+acc=npr

Run Time: 10 ms

Use run.do Tcl file

Use run.bash shell script

Open EPWave after run

Show output file after run

Download files after run

Examples

Log

Share

```
1 module tb;
2   union {
3     int num1;
4     byte num2;
5   } packet;
6
7 initial begin
8   packet.num1 = 'habcd;
9   $display("\tnum1= %0h",packet.num1);
10  $display("\tnum2= %0h",packet.num2);
11
12  packet.num2 = 'h43;
13  $display("\tnum1= %0h",packet.num1);
14  $display("\tnum2= %0h",packet.num2);
15
16 end
17 endmodule
18
```

```
# d = ff
# n = ff48
# d = 48
# test = '{n:0000ff48, d:48}
#       num1= abcd
#       num2= cd
#       num1= ab43
#       num2= 43
# exit
# End time: 13:13:55 on Dec 17, 2023, Elapsed time: 0:00:01
# Errors: 0, Warnings: 0
```

Done