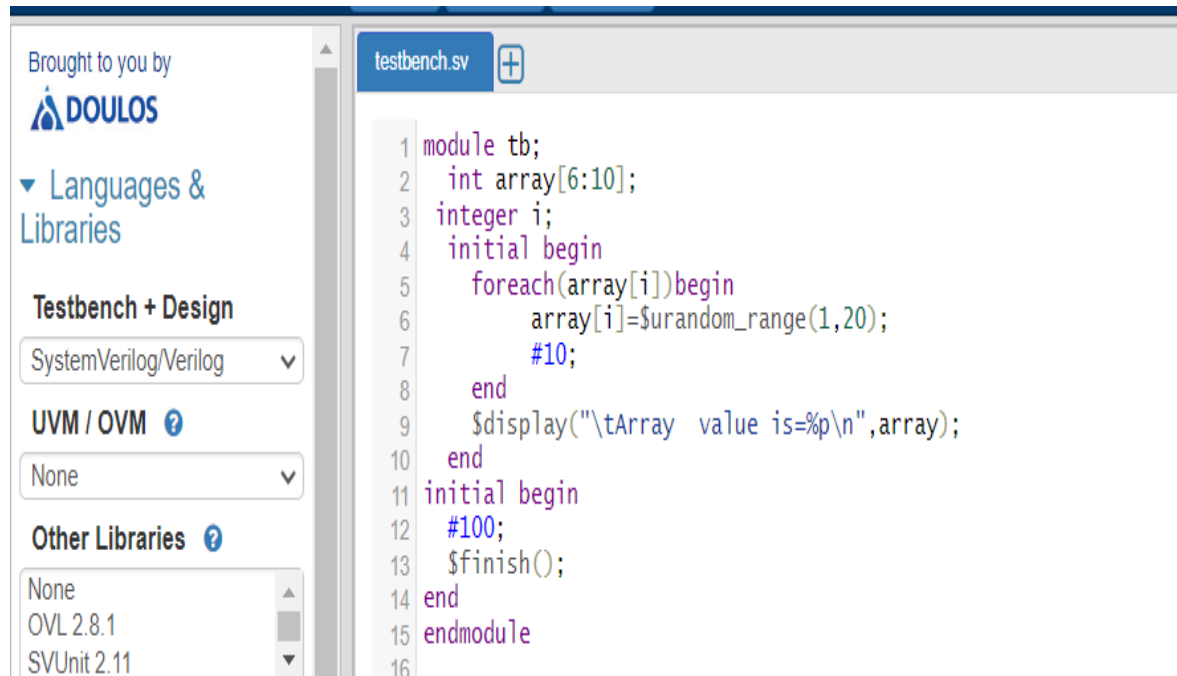
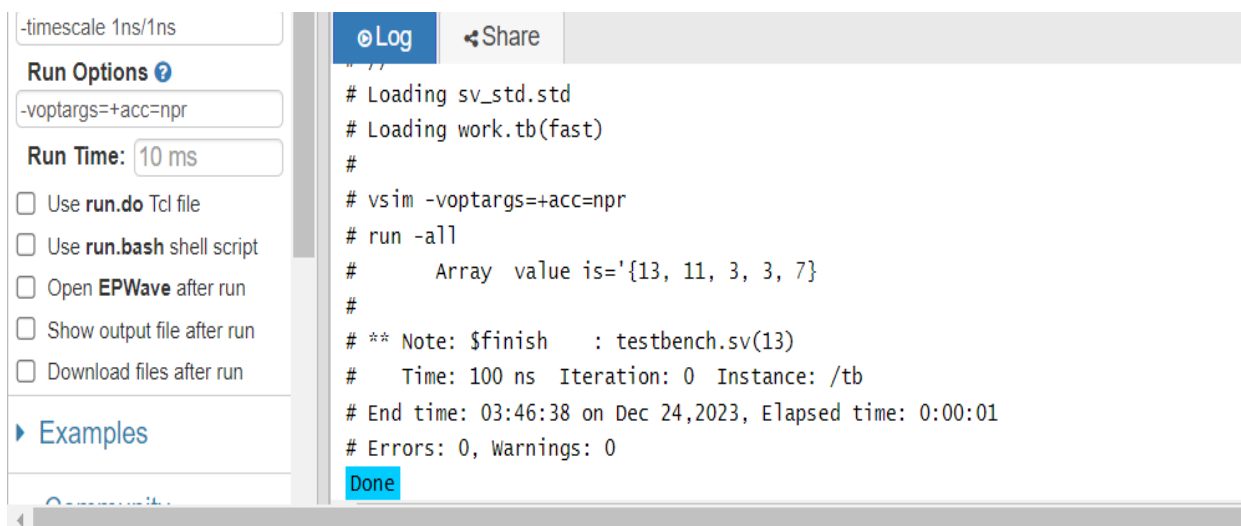


//You have an array, int array[6:10] Randomize array between 1 to 20 without using randc.



The screenshot shows the DOULOS IDE interface. On the left, the 'Languages & Libraries' sidebar is visible, with 'Testbench + Design' selected. The 'SystemVerilog/Verilog' dropdown is set to 'SystemVerilog/Verilog'. Under 'Other Libraries', 'SVUnit 2.11' is selected. The main editor displays a file named 'testbench.sv' with the following code:

```
1 module tb;
2   int array[6:10];
3   integer i;
4   initial begin
5     foreach(array[i])begin
6       array[i]=$urandom_range(1,20);
7       #10;
8     end
9     $display("\tArray value is=%p\n",array);
10  end
11  initial begin
12    #100;
13    $finish();
14  end
15 endmodule
16
```



The screenshot shows the 'Log' tab in the DOULOS IDE. The 'Run Options' sidebar on the left shows '-timescale 1ns/1ns', 'Run Time: 10 ms', and several checkboxes for run options. The log output is as follows:

```
# Loading sv_std.std
# Loading work.tb(fast)
#
# vsim -voptargs=+acc=npr
# run -all
#   Array value is='{13, 11, 3, 3, 7}'
#
# ** Note: $finish    : testbench.sv(13)
#   Time: 100 ns Iteration: 0 Instance: /tb
# End time: 03:46:38 on Dec 24,2023, Elapsed time: 0:00:01
# Errors: 0, Warnings: 0
Done
```

//Ques:3 it is possible to call a function constraint?if yes explain 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 ?

☐ Enable VUnit ?

▼ **Tools & Simulators** ?

Mentor Questa 2021.3 ▼

testbench.sv +

```
1 //Ques: it is possible to call a function constraint?if yes explain with example?
2
3 class sample;
4     rand bit [2:0]data;
5     rand bit [2:0]addr;
6
7     constraint data_c{
8         data == call(10);
9     }
10
11     function bit[2:0] call(bit[2:0] var1);
12         call=var1;
13     endfunction
14 endclass
15
16 module tb;
17     sample s;
18     initial begin
19         repeat(3)begin
20             s=new();
21             s.randomize();
22             $display("\tdata=%0d, addr=%0d", s.data, s.addr);
23         end
24     end
25 endmodule
```

-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

Log Share

```
# Loading work.testbench_sv_unit(fast)
# Loading work.tb(fast)
#
# vsim -voptargs=+acc=npr
# run -all
#     data=2,  addr=7
#     data=2,  addr=1
#     data=2,  addr=4
# exit
```

//Ques:4 write a constraint for 8-bit variable that provide distribution 70% for range 0-50 and the remaining 30%for range 50-255.

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 ?
☐ Enable VUnit ?

▼ Tools & Simulators ?

testbench.sv

```
1 // write a constraint for 8 bit variable that provide distribution 0-50 and the remaining 30%for range 50-255.
2
3 class sample;
4     rand bit[7:0]var1;
5
6     constraint var_c{
7         var1 dist {[0:50]:/70, [51:255]:/30};
8     }
9 endclass
10
11 module tb;
12     sample s;
13     initial begin
14         s=new();
15         repeat(10)begin
16             s.randomize();
17             $display("Variable is=%0d",s.var1);
18         end
19     end
20 endmodule
21
22
```

-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

```
# run -all
# Variable is=101
# Variable is=33
# Variable is=14
# Variable is=35
# Variable is=2
# Variable is=30
# Variable is=193
# Variable is=4
# Variable is=46
# Variable is=21
# exit
```

//Ques:5 Drive odd number within the range of 10 to 30 using sv constraint.

▼ 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 ?
☐ Enable VUnit ?

▼ Tools & Simulators ?

Mentor Questa 2021.3 ▼

Compile Options ?

```
1 //Ques: Drive odd number within the range of 10 to 30 using sv con
2 class sample;
3     randc bit[19:0]odd_num;
4
5     constraint num_c{
6         odd_num inside {[10:30]};
7     }
8
9     constraint odd_num_c{
10        odd_num%2 != 0;
11    }
12
13 endclass
14
15 module tb;
16     sample s;
17     initial begin
18         s=new();
19         repeat(10)begin
20             s.randomize();
21             $display("odd number is=%0p",s.odd_num);
22         end
23     end
24 endmodule
```

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

▼ Community

Log

Share

vsim -voptargs=+acc=npr
run -all
odd number is=21
odd number is=29
odd number is=17
odd number is=23
odd number is=19
odd number is=11
odd number is=25
odd number is=27
odd number is=15
odd number is=13

//Ques:6 write a constraint divisible by 5.

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

☐ Enable VUnit

testbench.sv

```
1 // write a constraint divisible by 5.
2 class sample;
3     randc bit[4:0]num;
4
5     constraint num_c{
6         num%5 == 0;
7     }
8
9 endclass
10
11 module tb;
12     sample s;
13     initial begin
14         s=new();
15         repeat(5)begin
16             s.randomize();
17             $display("\tNumber is=%0p",s.num);
18         end
19     end
20 endmodule
```

-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

Log

Share

```
#
# vsim -voptargs=+acc=npr
# run -all
#
# Number is=30
#
# Number is=10
#
# Number is=25
#
# Number is=5
#
# Number is=20
# exit
```

//Ques:7 Write a Constraint to detect odd number of 1's in an 8 bit Sequence.

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

☐ Enable VUnit

```
1 //Write a Constraint to detect odd number of 1's in an 8 bit Sequence.
2 class sample;
3     randc bit[4:0]num;
4
5     constraint num_c{
6         $countones(num)%2 != 0;
7     }
8
9 endclass
10
11 module tb;
12     sample s;
13     initial begin
14         s=new();
15         repeat(5)begin
16             s.randomize();
17             $display("\t Count_ones is=%0b",s.num);
18         end
19     end
20 endmodule
```

-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

Log

Share

```
#
# vsim -voptargs=+acc=npr
# run -all
#
# Count_ones is=10110
#
# Count_ones is=11111
#
# Count_ones is=111
#
# Count_ones is=1110
#
# Count_ones is=10101
# exit
```

//Ques:8 How to disable Constraint.



▼ 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 ?
☐ Enable VUnit ?

▼ Tools & Simulators ?

Mentor Questa 2021.3 ▼

Compile Options ?

-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

```
1 //How to disable Constraint.
2 class packet;
3     rand bit [7:0] addr;
4
5     constraint addr_c {
6         addr>4;
7         addr<15;
8     }
9 endclass
10
11 module tb;
12     packet pkt;
13     initial begin
14         pkt = new();
15         $display("\tbefor constraint disabled");
16         pkt.addr_c.constraint_mode(0);
17         $display("\t\taddr value is= %0d", pkt.addr);
18
19         $display("\tAfter constraint disabled");
20         repeat(2)begin
21             pkt.randomize();
22             $display("\t\t addr value is= %0d", pkt.addr);
23         end
24     end
25 endmodule
```

Log

Share

```
#
# vsim -voptargs=+acc=npr
# run -all
#
# before constraint disabled
#
# addr value is= 0
#
# After constraint disabled
#
# addr value is= 249
#
# addr value is= 233
# exit
```

//Ques:9 How to disable Randomization.

```
1 //How to disable Randomization.
2 class packet;
3     rand bit [4:0] data1;
4     rand bit [4:0] data2;
5
6     constraint data1_c {
7         data1 inside{[10:20]};
8     }
9     constraint data2_c {
10        data2 inside{25,30,40};
11    }
12 endclass
13
14 module tb;
15     packet pkt;
16     initial begin
17         pkt = new();
18         $display("\tbefor randomization disabled");
19         pkt.rand_mode(0);
20         $display("\t\tdata1 value is= %0d,data2 is=%0d", pkt.data1,pkt.data2);
21
22         $display("\tAfter randomization disabled");
23         pkt.rand_mode(1);
24         repeat(2)begin
25             pkt.randomize();
26             $display("\t\tdata1 value is= %0d,data2 is=%0d", pkt.data1,pkt.data2);
27         end
28     end
29 endmodule
```

-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

Community

Log

Share

Loading models (100%)

#

vsim -voptargs=+acc=npr

run -all

before randomization disabled

data1 value is= 0,data2 is=0

After randomization disabled

data1 value is= 19,data2 is=25

data1 value is= 12,data2 is=30

exit

End time: 07:46:26 on Dec 24,2023, Elapsed time: 0:00:02

Errors: 0, Warnings: 1

Done

//Ques:14 Suppose is a 7-bit variable. Write constraint for a condition where for alternate bit position value is 1.

The screenshot displays the DOULOS IDE interface. On the left, a sidebar contains navigation links for 'Languages & Libraries', 'Testbench + Design', and 'Tools & Simulators'. The 'Testbench + Design' section is active, showing 'SystemVerilog/Verilog' as the selected language and 'None' for UVM/OVM and Other Libraries. The main editor window shows a file named 'testbench.sv' with the following code:

```
1 //Suppose is a 7-bit variable .Write constraint for a condition where alternate bit position value is 1.
2 class sample;
3     rand bit[6:0] var1;
4     constraint var1_c{
5         foreach(var1[i])
6         {
7             if(i%2 ==0)
8                 var1[i] == 1;
9         }
10    }
11
12    function void print();
13        $display("\tvariable = %0b",var1);
14    endfunction
15 endclass
16 module tb;
17     sample s = new();
18     initial begin
19         s.randomize();
20         #10
21         s.print();
22     end
23 endmodule
```

Below the editor, the 'Run Options' section is visible, showing a run time of 10 ms and several checkboxes for execution options. The 'Log' tab is selected, displaying the execution output:

```
# //
# Loading sv_std.std
# Loading work.testbench_sv_unit(fast)
# Loading work.tb(fast)
#
# vsim -voptargs=+acc=npr
# run -all
#     variable = 1010101
# exit
# End time: 12:04:20 on Dec 24,2023, Elapsed time: 0:00:02
# Errors: 0, Warnings: 1
Done
```


//Ques:15 a is 32-bit input to Dut with clk and rst Output are 32 bits b and c. when a[0]==1,a=b. when a[0]==0,a=c.write the constraints for this Dut.

```

2 module dut(a,clk,rst,b,c,A);
3   input[31:0] a;
4   input clk,rst;
5   output reg[31:0] b,c,A;
6
7   always@(posedge clk)
8     begin
9       b = 1; c = 0;
10      if(a[0] == 0) begin
11        A = c;
12      end
13      else if(a[0] == 1) begin
14        A = b;
15      end
16      else
17        $display("no result");
18    end
19
20 endmodule

```

tb_sv + (D:\New folder\system_data\desktop_folder\System ver...-64 Constraint Randomization\code11_constrain_for_DUT) - GVIM

File Edit Tools Syntax Buffers Window Help

```

1 class sample;
2   rand bit[31:0] a1;
3   constraint c{
4     foreach(a1[i]){
5       if(i == 0)
6         a1[i] == 0;}}
7   function bit display();
8     $display("a1 = %0d",a1);
9   endfunction
10 endclass
11 module tb;
12   reg[31:0] a;
13   bit clk,rst;
14   wire[31:0] b,c,A;
15   sample s = new();
16   initial begin
17     clk = 0; rst = 1;
18     s.randomize();
19     a = s.a1;
20     s.display();
21   end
22   always #5 clk = ~clk;
23   dut d(.a(a),.clk(clk),.rst(rst),.b(b),.c(c),.A(A));
24   initial begin
25     #10
26     $display("A = %0d",A);
27     #100 $finish;
28   end
29 endmodule
30
31
32

```



