SystemVerilog Assignment

Name: Shefin Muhammed MS

Systemverilog Lab5 : Randomization

---

1. Randomization

1.1. Basic Randomization

Code:

```
// Code your testbench here
// or browse Examples
//*****************************************
// (c) SION Semiconductors (P) Ltd, Bangalore
// All rights reserved.
//*****************************************
// Filename    : 1_Basic_Randomization.sv
// Description : Declare Random Varable using rand
//               Randomize whole object using .randomize()
//*****************************************

class Basic_Rand ;

// Declaration of rand properties
randc bit [3:0] Address;
randc int Data;
      int Z;

// Write a method to print variables
function void print ();
  $display ("Address=%d,Data=%d,Z=%d",Address,Data,Z);
endfunction

endclass

program test;

// Declaring object handle
Basic_Rand obj;

initial
begin
  obj=new();
  repeat(20) begin
    void'(obj.randomize()); // Apply randomization to object
    obj.print();
  end
end

endprogram
```

Output:

```
Address=14,Data=-1593209748,Z=          0
Address= 4,Data= -242398630,Z=          0
Address= 3,Data=-1991512201,Z=          0
Address=15,Data=  371768801,Z=          0
Address=10,Data= 1503204522,Z=          0
Address= 0,Data= -582893553,Z=          0
Address= 1,Data= 1009625693,Z=          0
Address=13,Data= 1285267316,Z=          0
Address= 2,Data= -675248672,Z=          0
Address= 7,Data=-1704130390,Z=          0
Address= 8,Data=-1128542705,Z=          0
Address= 6,Data=  215002461,Z=          0
Address=12,Data= 1424301300,Z=          0
Address=11,Data= -603995104,Z=          0
Address= 5,Data=-1670207414,Z=          0
Address= 9,Data=-1078714753,Z=          0
Address= 3,Data=  223823205,Z=          0
Address= 4,Data= 1411836136,Z=          0
Address= 0,Data= -610386386,Z=          0
Address= 6,Data=-1673374387,Z=          0
```

1.2. Basic Constraints

Code:

```
1  //*****************************************
2  // (c) SION Semiconductors (P) Ltd, Bangalore
3  // All rights reserved.
4  //*****************************************
5  // Filename    : 2_Basic_Constraints.sv
6  // Description : Demonstrates Constraint Randomization
7  //*****************************************
8
9  // Class definition
10 class Const_Rand ;
11
12 // Declaration of rand properties
13 randc bit [7:0] Address;
14 rand  int       Data;
15       int       Z;
16
17 // Defining constraints
18 constraint adr_c  { (Address inside {5,6,[100:150]});}
19 constraint data_c {Data > 50; Data< 250;}
20
21 // Write a method to print variables
22 function void print ();
23   $display ("Address=%d,Data=%d,Z=%d",Address,Data,Z);
24 endfunction
25
26 endclass
27
28 // Program block
```

```
29 program Basic_Constraints;
30
31 Const_Rand obj;
32
33 initial
34 begin
35   obj=new();
36   repeat(10) begin
37     // Apply Randomization to object using assert
38     assert (obj.randomize());
39     obj.print();
40   end
41 end
42
43 endprogram
```

Output:

```
Address=145,Data=        237,Z=          0
Address=102,Data=         57,Z=          0
Address=126,Data=        195,Z=          0
Address=128,Data=        149,Z=          0
Address=141,Data=        134,Z=          0
Address=148,Data=        171,Z=          0
Address=121,Data=        241,Z=          0
Address=150,Data=         78,Z=          0
Address=130,Data=        190,Z=          0
Address=116,Data=        150,Z=          0
```

1.3. Rand Distribute

```
// Code your design here
//*****************************************
// (c) SION Semiconductors (P) Ltd, Bangalore
// All rights reserved.
//*****************************************
// Filename    : 3_Rand_Distribute.sv
// Description : Distribute Value in Specific range using inside
//               Exclusive Value over specific range using !
//*****************************************


program Rand_Distribute;

  class Dist_Constraint;
    rand bit [7:0] A;
    rand bit [7:0] B;
    rand bit [7:0] C[];
    //Distribute over a specific range
    constraint c {
         A inside { [3:7], 10,15 }; // inclusive
       !(B inside { [10:15] });     // exclusive
        (C.size() inside {[2:4]});
    }

    function void pre_randomize(); //pre_randomize();
      begin
        $display("A : %0d",A);
        $display("B : %0x",B);
        $display("C : %p",C);
      end
    endfunction

  endclass

  initial begin
    Dist_Constraint obj= new();
    repeat(10) begin
      obj.randomize(); // Randomize whole object
      $display("Object values: %p", obj);
    end
  end

endprogram
```

Output:

```
A : 0
B : 0
C : '{}
Object values: '{A:'h3, B:'hbb, C:'{'ha3, 'h8f, 'h11} }
A : 3
B : bb
C : '{'ha3, 'h8f, 'h11}
Object values: '{A:'h3, B:'h7, C:'{'hd6, 'h97, 'h16} }
```

```
A : 3
B : 7
C : '{'hd6, 'h97, 'h16}
Object values: '{A:'hf, B:'ha2, C:'{'h1, 'hde, 'hf3} }
A : 15
B : a2
C : '{'h1, 'hde, 'hf3}
Object values: '{A:'ha, B:'h6f, C:'{'hc9, 'h4f, 'hb5} }
A : 10
B : 6f
C : '{'hc9, 'h4f, 'hb5}
Object values: '{A:'ha, B:'h6f, C:'{'h2d, 'h9f, 'hf8} }
A : 10
B : 6f
C : '{'h2d, 'h9f, 'hf8}
Object values: '{A:'h3, B:'h23, C:'{'ha7, 'h8, 'h41} }
A : 3
B : 23
C : '{'ha7, 'h8, 'h41}
Object values: '{A:'ha, B:'h17, C:'{'hf0, 'h61, 'h67} }
A : 10
B : 17
C : '{'hf0, 'h61, 'h67}
Object values: '{A:'h4, B:'h60, C:'{'h32, 'h15} }
A : 4
B : 60
C : '{'h32, 'h15}
Object values: '{A:'ha, B:'h8a, C:'{'h3d, 'h4d, 'h3f} }
A : 10
B : 8a
C : '{'h3d, 'h4d, 'h3f}
Object values: '{A:'h6, B:'hce, C:'{'h9a, 'h76, 'h78, 'he7} }
```

1.4. Constraint Order Rand

```
//*****************************************
// (c) SION Semiconductors (P) Ltd, Bangalore
// All rights reserved.
//*****************************************
// Filename    : 4_Constraint_Order_Rand.sv
// Description : Create a class with methods
//               Create a object and initialize memory
//               Call methods using object
//*****************************************

program Constraint_Order_Rand;

class Order_Rand;

  rand bit [4:0] lo, med, hi;

  // Declare Order of Randomization Low to High
  constraint good {lo < med; med < hi;}

  function void print();
   $display("low=%d,medium=%d,high=%d",lo,med,hi);
  endfunction

endclass


Order_Rand pkt;

initial begin
  pkt=new();
  repeat(10)
  begin
   // randomize using assert
   assert(pkt.randomize())
   pkt.print();
  end
end

endprogram
```

Output:

```
low=12,medium=14,high=17
low= 7,medium=19,high=29
low= 4,medium=16,high=30
low= 6,medium=16,high=23
low= 8,medium=10,high=16
low= 2,medium= 4,high=10
low= 4,medium= 8,high=30
low= 1,medium= 5,high= 7
low=14,medium=19,high=31
low= 2,medium= 8,high=28
```

1.5. Weighted Constraint Rand

```
//*****************************************
// (c) SION Semiconductors (P) Ltd, Bangalore
// All rights reserved.
//*****************************************
// Filename    : 5_Weighted_Constraint_Rand.sv
// Description : Distribute elements specific rand
//               By using dist keyword
//*****************************************

program Dist_rand;

  class frame;
    rand bit [7:0] Address;
    rand bit [7:0] Data;
    rand bit [15:0] MEM;

    // Initializing Value for specific rang of MEM
    constraint len {MEM dist {
        [64   :  127 ] := 10,
        [128 :  511 ] := 10,
        [512 :  2048] := 10
      };
    }

    // Initializing Value for specific rang of Adress
    constraint Adres {
      Address dist {
        0   := 1,
        1   := 1,
        2   := 5,
        4   := 1
      };
    }

    // Initializing Value for specific rang of data
    constraint DATA {
      Data dist {
        [0   : 5   ] :/ 5,
        [6   : 100 ] := 1,
        [101 : 200 ] := 1,
        [201 : 255 ] := 1
      };
    }

    function void post_randomize();
      begin
        $display("Addres : %0x",Address);
        $display("Data : %0b"  ,Data);
        $display("MEM   : %0d" ,MEM);
      end
    endfunction

  endclass

  initial begin
    frame obj = new();
```

```
57      integer i,j = 0;
58
59     // Reprat execution 4 times using for loop
60      for (j=0;j < 4; j++) begin
61        i = obj.randomize();
62      end
63    end
64
65  endprogram
```

Output:

```
Addres : 4
Data : 110011
MEM   : 138
------------------
Addres : 2
Data : 10000110
MEM   : 772
------------------
Addres : 2
Data : 100111
MEM   : 735
------------------
Addres : 1
Data : 110100
MEM   : 1640
------------------
```

1.6. Arrays Constraint

```
class ArrayConst;

 rand int f;
      int fib[5] = '{1,2,3,5,8};

 // Initialize constraint for Array
 constraint c_fibonacci {
  f inside fib;
 }

endclass


program Array_Const;

ArrayConst pkt;

initial begin

  // Creating object
  pkt=new();

  // Assert Based Randomization
  for(int i=0;i<5;i++) begin
    assert(pkt.randomize())
    $display("%d",pkt.f);
  end

end

endprogram
```

Output:

```
1
8
1
3
2
```

1.7. Arrays Rand

```
1
2  //*****************************************
3  // (c) SION Semiconductors (P) Ltd, Bangalore
4  // All rights reserved.
5  //*****************************************
6  // Filename    : 7_Arrays_Rand.sv
7  // Description : Randomization with Arrays
8  //*****************************************
9
10 class Arrays_Rand;
11
12 randc bit [15:0] index;    // Index into array
13       int         array[];  // Values to choose
14
15 // Construct & initialize
16 function new(int a[]);
17   array = a;
18 endfunction
19
20 // Return most recent pick
21 function int pick;
22   return array[index];
23 endfunction
24
25 // Constrain index of array using .size()
26 constraint c_size {index < array.size();}
27
28 endclass
29
30 // Top level program block
31 program ArraysRand;
32
33 Arrays_Rand pkt;
34
35 initial begin
36   pkt = new('{1,3,5,7,9,11,13});// Initialize Elements
37   repeat (pkt.array.size()) begin
38    assert(pkt.randomize());
39    $display("Picked %d [%0d]", pkt.pick(), pkt.index);
40   end
41 end
42
43 endprogram
```

Output:

```
Picked           5 [2]
Picked           7 [3]
Picked           3 [1]
Picked           9 [4]
Picked          13 [6]
Picked          11 [5]
Picked           1 [0]
```

1.8. Enum Rand

```
//*****************************************
// (c) SION Semiconductors (P) Ltd, Bangalore
// All rights reserved.
//*****************************************
// Filename    : 8_Enum_Rand.sv
// Description : Randomization with enum type
//*****************************************


class Enum_Rand;

// Declaration of enumarated datatype
typedef enum {SUN, MON, TUE, WED, THU, FRI, SAT} days_e;

// Queue of Enum
days_e choices[$];

rand days_e choice;

constraint cday {choice inside choices;}
endclass


// Top level program block
program EnumRand;

Enum_Rand days;

initial begin

  days = new();

  // Initializing Range of Enum_Rand
  days.choices = {Enum_Rand::SUN, Enum_Rand::SAT};

  // Assertion based randomization
  assert (days.randomize());
  $display("Random weekend day %s\n", days.choice.name);

  // Modify Enum_Rand
  days.choices = {Enum_Rand::MON, Enum_Rand::TUE, Enum_Rand::WED,
     Enum_Rand::THU, Enum_Rand::FRI};
  assert (days.randomize());
  $display("Random week day %s", days.choice.name);

end

endprogram
```

Output:

```
Random weekend day SAT

Random week day WED
```

1.9. Event Constraint

```
//*****************************************
// (c) SION Semiconductors (P) Ltd, Bangalore
// All rights reserved.
//*****************************************
// Filename    : 9_Event_Constraint.sv
// Description : Describe Condition based Constraint
//               Initialize Range of Element using Event ->
//*****************************************

class EventConst;

randc bit [1:0]addr_range;
randc int addr;

function print();
 $display ("addr_range=%d,addr=%d",addr_range,addr);
endfunction

// Initialize Emenent range using Event (->) with condition using
constraint limit{
   (addr_range==0)-> addr inside {[0:100]};
   (addr_range==1)-> addr inside {[101:1000]};
   (addr_range==2)-> addr inside {[1001:10000]};
 }
endclass

// Top level program block
program Event_Cond_Const;

EventConst obj;

initial begin
  obj=new();
  repeat(5) begin
  obj.randomize();
  obj.print();
  end
end

endprogram
```

Output:

```
addr_range=1,addr=        906
addr_range=2,addr=       3173
addr_range=0,addr=         56
addr_range=3,addr=  231368803
addr_range=3,addr=  652288026
```

1.10. Conditional Constraint

```
//*****************************************
// (c) SION Semiconductors (P) Ltd, Bangalore
// All rights reserved.
//*****************************************
// Filename    : 10_Conditional_Constraint.sv
// Description : Demonstration of Conditional Constraints
//*****************************************

class Conditional_Constraint;

randc bit [1:0] addr_range;
randc int       addr;

function print();
 $display ("addr_range=%d,addr=%d",addr_range,addr);
 endfunction

// Conditional Constraints
constraint limit{
  if (addr_range==0) addr inside {[0:100]};
  if (addr_range==1) addr inside {[101:1000]};
  if (addr_range==2) addr inside {[1001:10000]};
 }
endclass

// Top level program file
program CondConstraint;

Conditional_Constraint obj;

initial begin
  obj=new();
  repeat(5) begin
  obj.randomize();
  obj.print();
  end
end

endprogram
```

Output:

```
addr_range=1,addr=        906
addr_range=2,addr=       3173
addr_range=0,addr=         56
addr_range=3,addr=  231368803
addr_range=3,addr=  652288026
```

1.11. Constraint Mode

```
//*****************************************
// (c) SION Semiconductors (P) Ltd, Bangalore
// All rights reserved.
//*****************************************
// Filename    : 11_Constraint_Mode.sv
// Description : Demonstration of Constraint mode
//*****************************************

program Constraint_Mode;

  class Moderandom;
  rand int a;
  rand int b;
  function  void print ();
    $display ("a=%d,b=%d",a,b);
  endfunction

  // Initialize Range of Elements
  constraint limit1 {
                    a inside {[0:100]};
                    b inside {[0:50]};
                  }
  constraint limit2 {a<200;
                     b<100;
                     }
endclass

Moderandom obj;

initial begin

  obj=new();
  // Call the constraint using mode
  obj.limit1.constraint_mode(1);
  obj.randomize(a);
  obj.randomize(b);
  obj.print();
end

endprogram
```

Output:

```
a=75,b=19
```