Electronic Circuits

Week 14: Randomization

Fenerbahçe University

Randomization

- Doğrulamada rasgele veri içeren girişlerin üretilmesi önemli olabilmektedir.

- System Verilog dilince bu işlem için
  - rand
  - ranc

isminde iki değişken türü tanımlanabilmektedir.

Randomization

- Rand: Eşit dağılımda rasgele sayı üretir.
- Veri türünün alabileceği sayı aralığı ne ise, o sayı aralığındaki tüm sayılar eşit olasılık ile üretilebilir.
- Ard arda aynı sayılar üretilebilir.

Randomization

- Randc: Ard arda aynı sayıların üretilmesinin önüne geçilmesi için, tekrarsız rasgele sayı üretir. Üretilmiş bir sayı, ancak diğer üretilebilecek tüm sayılar rasgele olarak üretildikten sonra tekrar üretilebilir.

Örn: 2 bitlik bir rasgele sayı üretiminde;

01 11 00 10; 00 11 10 01; 11 00 11 01

gibi bir üretim olabilir.

Üretilen sayı tekrarı, diğer sayıların üretiminden sonra olabilir.

Randomization

Her sınıfın “randomize” isminde sanal bir fonksiyonu vardır.

İçerisinde rand ve ranc olan ifadeli sınıflarda yeniden rasgele değer üretilmesini sağlar.

Randomization

- Fonksiyon başarılı çalıştığında geriye 1 değilse 0 döndürmektedir.
- Sınıfın içinde tekrar bir randomize isminde fonksiyon tanımlanamaz.

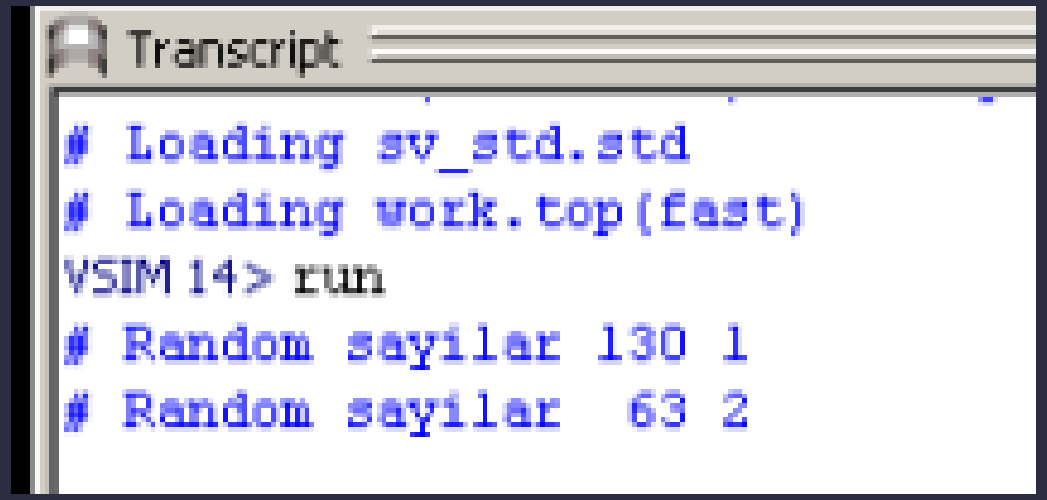
Randomization

Verilen randomUretici isminde bir sınıf tanımlanmış ve içerisine iki adet rasgele sayı oluşturması için rand ve randc türlerinde değişken tanımlanmıştır.

Sınıf iki kere çağrılarak iki farklı rasgele sayı üretilmesi sağlanmıştır.

```
timeunit 1ns; timeprecision 1ns;

module top;

class randomUretici;
    rand bit[7:0] random1;
    randc bit[1:0] random2;
endclass

randomUretici randClass1 = new();
int sonuc;

initial begin
    sonuc = randClass1.randomize();
    if(!sonuc)
        $display("Random Uretilemedi");
    else
        $display("Random sayilar %d %d",
randClass1.random1, randClass1.random2);

    sonuc = randClass1.randomize();
    if(!sonuc)
        $display("Random Uretilemedi");
    else
        $display("Random sayilar %d %d",
randClass1.random1, randClass1.random2);

end

endmodule: top
//`end_keywords
```

Randomization



Oluşturulan rasgele sayılar ekrana bastırılmıştır.

Randomization

Sınıfta çağrılan randomization fonksiyonu otomatik olarak çağırdığı 2 fonksiyon vardır. Bunlar;

- pre_randomize(): Randomization işlemi yapılmadan önce çağrılmaktadır
- post_randomize(): Randomization işlemi yapıldıktan sonra çağrılmaktadır.

Tablo'da pre ve post fonksiyonları örneği verilmiştir.

Randomization

```
  timeunit 1ns; timeprecision 1ns;

module top;

class randomUretici;
    rand bit[7:0] random1;
    randc bit[1:0] random2;
    bit [1:0] parity;

    function void pre_randomize();
        $display("Random Cagrilmadan Once");
    endfunction

    function void post_randomize();
        $display("Random Cagrildiktan Sonra");
        parity = random1[1:0] ^ random2[1:0];
    endfunction
endclass

randomUretici randClass1 = new();
int sonuc;
```

```
initial begin
    sonuc = randClass1.randomize();
    if(!sonuc)
        $display("Random Uretilemedi");
    else
        $display("Random sayilar %d %d",
randClass1.random1, randClass1.random2);

    sonuc = randClass1.randomize();
    if(!sonuc)
        $display("Random Uretilemedi");
    else
        $display("Random sayilar %d %d",
randClass1.random1, randClass1.random2);

end

endmodule: top
```

Randomization

```
# Random Cagrilmadan Once
# Random Cagrildiktan Sonra
# Random sayilar 130 1
# Random Cagrilmadan Once
# Random Cagrildiktan Sonra
# Random sayilar  63 2
```

Pre ve post fonksiyonu çıktıları gösterilmektedir.

Randomization

Her bir random değişken türünün randMode isminde çağrılabileceği metodu vardır. Bu metod ile istenirse rasgele değişkenin rasgele değer üretmesi iptal edilebilmektedir.

Rasgele üretimi iptal etmek için randMode metoduna 0 argümanı verilmelidir. Metoda 1 verildiğinde tekrar aktif edilecektir.

Randomization

```
timeunit 1ns; timeprecision 1ns;

module top;

class randomUretici;
    rand bit[7:0] random1;
    randc bit[1:0] random2;
    rand bit[7:0] random3;
endclass

randomUretici randClass1 = new();
int durum;
int sonuc;
```

```
initial begin
    randClass1.random1.rand_mode(0);
    sonuc = randClass1.randomize();
    durum = randClass1.random1.rand_mode();
    $display("Durum %d, Random sayilar %d %d %d",
durum, randClass1.random1, randClass1.random2,
randClass1.random3);

    sonuc = randClass1.randomize();
    durum = randClass1.random1.rand_mode();
    $display("Durum %d, Random sayilar %d %d %d",
durum, randClass1.random1, randClass1.random2,
randClass1.random3);

end

endmodule: top
```

Randomization

randmode'in 0 yapıldığı ilk çıktının tekrar 0 ürettiği gözlenmektedir.

```
VSIM 22> run
# Durum              0, Random sayilar    0 1 122
# Durum              0, Random sayilar    0 2 196
```

Randomization

Kısıtlar, rasgele data üretilirken sayıları kısıtlamak amacıyla kullanılmaktadır.

Randomization

Verilen randomUretici sınıfında random2 olarak tanımlanmış değişkenin değerinin aslı 2'b11 üretilmemesi sağlanmıştır.

```
timeunit 1ns; timeprecision 1ns;

module top;

class randomUretici;
    rand bit[7:0] random1;
    randc bit[1:0] random2;
    rand bit[7:0] random3;

    constraint const1 {random2 != 2'b11;}
endclass

randomUretici randClass1 = new();
int sonuc;

initial begin


    for(int i=0;i<10;i++) begin
        sonuc = randClass1.randomize();
        $display("Sonuc %d, Random sayilar %d %d %d",
sonuc, randClass1.random1, randClass1.random2,
randClass1.random3);
    end

end

endmodule: top
//`end_keywords
```

Randomization

10 defa randomization fonksiyonu çağrılmış ve sonuçlarında random2 değerinin 2b’11 olmadığı gözlemlenmiştir.

```
# Sonuc          1, Random sayilar 130 0 122
# Sonuc          1, Random sayilar  63 1 196
# Sonuc          1, Random sayilar 148 2 218
# Sonuc          1, Random sayilar  81 0 243
# Sonuc          1, Random sayilar 209 2 104
# Sonuc          1, Random sayilar  39 1 233
# Sonuc          1, Random sayilar 147 0  29
# Sonuc          1, Random sayilar 159 2 184
# Sonuc          1, Random sayilar 247 1 255
# Sonuc          1, Random sayilar 130 0 226
```

Randomization

Kısıt koymak için “inside” operatörü kullanılabilir. Inside operatörü ile verilen bir değişkenin alabileceği değerler ve/veya bir değer aralığı ifade edilebilmektedir.

```
class randclass;
  rand bit[7:0] p3;
  constraint c1 {p3 inside {3, 7, [11:20]};}
endclass

randclass myrand = new;

int ok;

initial begin
  ok = myrand.randomize();
  ...
end
```

Randomization

“Inside” operatörü ile tanımlanmış bir aralığın dışında olan sayıları ürettirebilmek için ifadenin başına ünlem işareti konur.

```
class not_inside;
  rand bit[7:0] p3;
  constraint c2 { !( p3 inside {1, 7, [10:255]} ); }
endclass
```

Randomization

Kısıt koymak için “inside” operatörü kullanılabilir. Inside operatörü ile verilen bir değişkenin alabileceği değerler ve/veya bir değer aralığı ifade edilebilmektedir.

```
class randclass;
  rand bit[7:0] p3;
  constraint c1 {p3 inside {3, 7, [11:20]};}
endclass

randclass myrand = new;

int ok;

initial begin
  ok = myrand.randomize();
  ...
end
```

Randomization

```
timeunit 1ns; timeprecision 1ns;

module top;

class randomUretici;
    rand bit[3:0] random1;

    constraint const1 { !(random1 inside{0,1,[5:15]});}
endclass

randomUretici randClass1 = new();
int sonuc;

initial begin

    for(int i=0;i<5;i++) begin
        sonuc = randClass1.randomize();
        $display("Sonuc %d, Random sayilar %d", sonuc,
randClass1.random1);
    end

end

endmodule: top
//`end_keywords
```

```
Sonuc           1, Random sayilar  2
Sonuc           1, Random sayilar  4
Sonuc           1, Random sayilar  2
Sonuc           1, Random sayilar  3
Sonuc           1, Random sayilar  3
```

Kısıtlı Rasgele Sayı Üretimi

Randomization

Rasgele değerler ağırlıklı olarak da üretilebilmektedir.
Bunun için “dist” ifadesi kullanılmaktadır.

Verilmeyen değerlerin ağırlığı default olarak 1’dir.

Randomization

```
class randclass;
  rand bit[7:0] p3;
  constraint c2 { p3 dist { [0:127]:=2, [128:255]:=1 }; }
endclass
```

= işareti eleman veya değer aralığındaki grubun tamamına sağ tarafında verilen sayı kadar ağırlık atar.

Randomization

```
timeunit 1ns; timeprecision 1ns;

module top;

class randomUretici;
    rand bit[3:0] random1;

    constraint const1 { random1  dist {[0:6]:=2, [7:10]:=3, [11:15]};}
endclass

randomUretici randClass1 = new();
int sonuc;

initial begin

    for(int i=0;i<10;i++) begin
        sonuc = randClass1.randomize();
        $display("Sonuc %d, Random sayilar %d", sonuc,
randClass1.random1);
    end

end
endmodule: top
```

```
# Sonuc           1, Random sayilar  7
# Sonuc           1, Random sayilar  9
# Sonuc           1, Random sayilar  7
# Sonuc           1, Random sayilar  8
# Sonuc           1, Random sayilar 10
# Sonuc           1, Random sayilar  7
# Sonuc           1, Random sayilar  6
# Sonuc           1, Random sayilar  7
# Sonuc           1, Random sayilar 11
# Sonuc           1, Random sayilar 10
```

Ağırlık Dağılımı Konsol Çıktısı

Randomization

```
class randclass;
  rand bit[7:0] p3;
  constraint c2 { p3 dist { [0:127]:=2, [128:255]:=1 }; }
endclass
```

= işareti eleman veya değer aralığındaki grubun tamamına sağ tarafında verilen sayı kadar ağırlık atar.