Artgraphics

SystemVerilog基礎講座

ランダムスティミュラスの生成

篠塚一也
アートグラフィックス
Document Revision: 1.0,2025.04.29

www.artgraphics.co.jp

注意事項（Caveat）

- SystemVerilogの知識を個人的に習得する目的として本資料を活用して下さい。本資料を通して、業務（実践）で必要となるSystemVerilogに関する知識を習得して頂くのが本来の目的です。
- 転用目的（本来の目的と違った他の用途に使う事）で本資料を使用する事はご遠慮下さい。また、本資料から学んだ知識を転載する場合等は出典が本資料である事を明記して下さい。但し、他の著者の文書にも書かれている内容は、この限りではありません。本注意事項は現在及び過去に於ける弊社からの全てのフリーダウンロード資料に適用されます。
- 本注意事項に合意出来ない場合には、本資料を速やかに抹消して下さい。尚、ダウンロード記録は、依然として残ります。

Copyright © 2025 Artgraphics. All rights reserved.

SystemVerilogの
ランダムスティミュラス生成機能

- 整数型だけでなく実数型の変数もランダム変数に指定できます。ランダム変数でない変数を状態変数（state variable）と呼びます。
- 制約を与えて乱数を発生できます。
- ランダム変数にはビット長の制限はありません。
- 制約を一時的に有効・無効にできます。
- クラスのrandomize()メソッドを呼ぶことにより乱数を発生できるので、ユーザが乱数を発生するタイミングを自由に制御できます。
- ランダム変数を一時的に無効にする事ができます。
- 状態変数を一時的にランダム変数にできます。
- クラス内にランダム変数を定義するのが最も簡単ですが、クラス外でも変数を一時的にランダム変数に指定できます。

Copyright © 2025 Artgraphics. All rights reserved.

例題ー1　ランダム変数の定義法

- 4ビット符号なしの加算器をテストするためのトランザクションを定義してください。ただし、以下の条件を満たすようにしてください。

① 加算器は入力aとbが必要で、出力はcoとsumであるとします。ここで、coはcarry-out、sumは和を求めるために使用されます。

② aとbをランダム変数に定義してください。

Copyright © 2025 Artgraphics. All rights reserved.

例題の解答と解説

- 例えば、以下のようにトランザクションを定義できます。

```
class simple_item_t;
rand logic [3:0] a, b;
logic [3:0]      sum;
logic            co;
endclass
```

Copyright © 2025 Artgraphics. All rights reserved.

例題－2　制約の定義

- 例題－1で定義したトランザクションに以下の条件を満たすような制約を定義してください。

① a ＞ b

② bは2、8、10の何れかの値だけを取ります。

Copyright © 2025 Artgraphics. All rights reserved.

例題の解答と解説

- 二つの制約が必要ですが、個別に定義した方が汎用性があります。

```
class simple_item_t;
rand logic [3:0] a, b;
logic [3:0]      sum;
logic            co;
constraint C1 { a > b; }
constraint C2 { b inside {2,8,10}; }
endclass
```

Copyright © 2025 Artgraphics. All rights reserved.

例題ー3　乱数の発生

- 例題ー2で定義したトランザクションを使用して乱数を発生する例を書いてください。

Copyright © 2025 Artgraphics. All rights reserved.

例題の解答と解説

- 乱数を発生するためには、以下のようにクラスのビルトインメソッドrandomize()を呼び出さなければなりません。

```
simple_item_t     item;
item = new;
assert( item.randomize() );
```

- 乱数発生が成功するとは限らないので、上記のようにrandomize()の戻り値を確認する習慣を付けておくと安全です。

Copyright © 2025 Artgraphics. All rights reserved.

例題ー4　制約を一時的に無効にする

- 例題ー2で定義したbに関する制約を一時的に無効にするにはどうすれば良いですか？

Copyright © 2025 Artgraphics. All rights reserved.

例題の解答と解説

- 制約に定義されているconstraint_mode()メソッドを呼べば制約を有効・無効に制御できます。
- 以下のようにすれば、制約C2を無効にできます。

```
simple_item_t    item;
item = new;
item.C2.constraint_mode(0);
```

Copyright © 2025 Artgraphics. All rights reserved.

例題ー5　クラスを使用せずに乱数を発生する

- 変数a、b、cは4ビットの符号なし整数であるとします。クラスを使用せずに、これらの変数に乱数を発生してください。ただし、以下の条件を満たしてください。

① a+b==c

② a > b

Copyright © 2025 Artgraphics. All rights reserved.

例題の解答と解説

- クラスを使用せずに乱数を発生するためには、std::randomize()を使用できます。制約条件があれば、withクローズで指定できます。
- この場合には、以下のようになります。

```
module test;
logic [3:0]       a, b, c;

initial begin
        repeat( 5 ) begin
                assert( std::randomize(a,b,c) with {a+b==c; a > b;} );
                $display("%2d %2d %2d",a,b,c);
        end
end

endmodule
```

Copyright © 2025 Artgraphics. All rights reserved.

参考文献

文献[1]は最新版の仕様書です。是非一読下さい。SystemVerilogに関する知識の確認には、文献[2-4]を参照して下さい。文献[4]は設計分野で必要とされるSystemVerilogの基礎知識を非常に詳しく解説しています。ランダムスティミュラスの生成に関する詳しい解説は文献[2,3]にあります。文献[5]にはランダムスティミュラスの生成に関する基礎知識が例題とともに簡潔に要約されています。初心者におすすめします。

[1] IEEE Std 1800-2023: IEEE Standard for SystemVerilog – Unified Hardware Design, Specification and Verification Language.

[2] 篠塚一也、SystemVerilogによる検証の基礎、森北出版 2020.

[3] 篠塚一也、SystemVerilog入門, 共立出版 2020.

[4] 篠塚一也、SystemVerilog超入門, 共立出版 2023.

[5]篠塚一也、検証のためのSystemVerilogプログラミング、森北出版 2022.

Copyright © 2025 Artgraphics. All rights reserved.