



Artgraphics

SystemVerilog基礎講座 プログラミング

篠塚一也

アートグラフィックス

Document Revision: 1.0,2025.05.30

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。

例題－1 ループ処理とアルゴリズムの例

- 以下に示すファンクションはdata_inに含まれるビット1の数を調べています。foreach文を使わずにrepeat文で処理を書き直して下さい。

```
function [3:0] count_ones([7:0] data_in);  
count_ones = 0;  
foreach(data_in[i])  
    count_ones = count_ones + data_in[i];  
endfunction
```

- ただし、変数インデックスのビットセレクトを使用しないで下さい。つまり、data_in[i]のような表現を避けて下さい。

例題の解答と解説

- repeat文による記述は以下のようになります。

```
function [3:0] count_ones([7:0] data_in);
logic [7:0]      tmp;
    count_ones = 0;
    tmp = data_in;
    repeat( 8 ) begin
        count_ones = count_ones + tmp[0];
        tmp = tmp >> 1;
    end
endfunction
```

- data_in[i]のような表現を使えないので、アレイのLSBで判定するようにします。そのためには、data_inを一時的にtmpにコピーし、tmpを右に1ビットシフトし調べたビットをLSBに移動させます。

例題－2 機能的な表現例

- 変数が以下のように定義されているとします。

```
logic a, b, c;
```

- この時、以下の式を簡単にできますか？

```
a+b+c inside {1,3}
```

例題の解答と解説

- $a+b+c$ はフルアダーを示しています。与えられた式は、奇数パリティチェッカー機能です。
- フルアダーにはsumとcoがありますが、sumはパリティチェッカーの機能を持ち、 $sum == 1$ は奇数パリティチェッカーとして機能します。
- したがって、sumを使うことで与えられた式を以下のように簡単に表現できます。

$$\frac{a^b c}{c}$$

参考文献

文献[1]は最新版の仕様書です。是非一読下さい。ここで紹介した基礎知識は本質的に必要な知識です。理解を深めるためには、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。初心者におすすめます。

- [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] SystemVerilogによる効果的実装技術、アートグラフィックス 2025.