



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: 1.4,2025.11.05

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。

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

例題－3 互換性に関する問題

- 以下のように変数が定義されているとします。以下の問に答えて下さい。

```
typedef logic [15:0] ushort_t;  
ushort_t          a;  
string           s;
```

- ① 以下の代入文は正しいですか？正しくなければ訂正して下さい。

```
a = "JP";  
s = "JP";
```

- ② 以下の代入文は正しいですか？正しくなければ訂正して下さい。

```
a = s;
```

- ③ 以下の代入文は正しいですか？正しくなければ訂正して下さい。

```
s = a;
```

例題の解答と解説

- ① 何れの代入文も正しい記述法です。
- ② 誤りです。integral型の変数aにstring型の値を割り当てる事はできません。以下のように型を揃えなければなりません。

```
_____
a = ushort_t'(s);
_____
```

- ③ 誤りです。以下のようにしなければなりません。

```
_____
s = string'(a);
_____
```

例題 - 4 オペレータの書き換え

- 以下のように変数が定義されているとします。以下の問に答えて下さい。

```
logic [7:0] a, z;
```

- ① 以下の記述におけるシフトオペレータを使用しないで同じ機能を実現して下さい。

```
z = a >> 1;
```

- ② 以下の記述におけるシフトオペレータを使用しないで同じ機能を実現して下さい。

```
z = a << 1;
```

例題の解答と解説

- シフトオペレータの代わりにパートセレクトを使用できます。

- ① パートセレクトで同じ機能を実現できます。

```
z = a[7:1];
```

- ② パートセレクトと `{}` オペレータで同じ機能を実現できます。

```
z = {a[6:0], 1'b0};
```

例題－5 簡略表現

- 以下のように変数が定義されているとします。以下の問に答えて下さい。

```
logic [7:0]      a, z;
```

- 以下の記述を一つの代入文で表現し直して下さい。

```
z[7] = 1'b0;  
z[6] = a[6];  
z[5] = a[6]&a[5];  
z[4] = a[5]&a[4];  
z[3] = a[4]&a[3];  
z[2] = a[3]&a[2];  
z[1] = a[2]&a[1];  
z[0] = ~a[0];
```

例題の解答と解説

- {} オペレータを使用すれば、以下のように一行で記述できます。

```
z = {1'b0, a[6], a[6:2]&a[5:1], ~a[0]};
```

例題－6 書式を揃える問題

- string型の変数ieeeが以下のように定義されているとします。

```
string ieee = "ieee";
```

- 変数ieeeの内容を以下のように、左詰め、中央揃え、右詰めでプリントするための手順をSystemVerilogで記述して下さい。なお、[から]までのプリント幅は20文字です。

```
[ieee  
[ ieee  
[ ieee]
```

例題の解答と解説

- 以下のようにすれば、正しい結果を得られます。

```
string  ieee = "ieee";
int     left_spaces, right_spaces;

$display("[%18s]", ieee);
left_spaces = (18-ieeee.len)/2;
right_spaces = 18-ieeee.len-left_spaces;
$display("[%s]", {left_spaces{" "}, ieee, right_spaces{" "}});
$display("[%18s]", ieee);
```

- このような技術を習得するには、文献[5]をすすめます。

例題ー7 case文を式で表現する問題

- 以下のようなデザインが与えられたとします。

```
module design(input a,b,ci,output logic co);
  always_comb
    casex ({a,b,ci})
      3'b11x,
      3'bx11,
      3'b1x1: co = 1'b1;
      default co = 1'b0;
    endcase
endmodule
```

- 連続代入文を使用してこのデザインを書き直して下さい。

例題の解答と解説

- 以下のような記述をできます。insideオペレータは、文献[3,4]に詳しく解説されています。

```
module design(input a,b,ci,output co);  
  assign co = {a,b,ci} inside {3'b11x,3'bx11,3'b1x1};  
endmodule
```

- あるいは、以下のようにしても良いです。

```
module design(input a,b,ci,output co);  
  assign co = a&b | b&ci | ci&a;  
endmodule
```

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[2-5]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。初心者におすすめです。

[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.