Artgraphics

SystemVerilog基礎講座
センシティビティ

篠塚一也
アートグラフィックス
Document Revision: 1.0,2025.05.02

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　イベント待ち

- 以下のような状況を仮定します。質問に答えて下さい。

```
logic [7:0]       a,
                  b[5];
```

① 下記のイベント待ちは正しいですか？

```
@(posedge a);
```

① 下記のイベント待ちは正しいですか？

```
@(b);
```

例題の解答と解説

- イベント待ちは連続領域を使用しなければなりません。また、posedgeとnegedgeのエッジセンシティブなイベント待ちは、指定された式の値のLSBで判定されます。

① 変数aは8ビットなので、**@(posedge a[0])**として判定されます。したがって、意図した記述と異なると考えられるので、記述は正しくないといえます。

② bはunpackedアレイなので、記述は間違いです。 unpackedアレイの要素同士は不連続に配置されているので、シミュレータはイベント待ちを監視する事はできません。したがって、以下のように訂正しなければなりません。

```
@(b[0],b[1],b[2],b[3],b[4]);
```

例題−2　always_combとセンシティビティ

- 以下の記述は正しいですか？

```
module mux_if(input a,b,c,d,sa,sb,sc,output logic out);

always_comb
        select(a,b,c,d);

task select(v,w,x,y);
        if( sa == 1 )
                out = v;
        else if( sb == 1 )
                out = w;
        else if( sc == 1 )
                out = x;
        else
                out = y;
endtask

endmodule
```

例題の解答と解説

- 与えられた記述は正しくありません。タスク内部はデータフロー解析されないので、always_combが生成するセンシティビティリストは不完全です。
- always_combが生成するセンシティビティリストは、@(a,b,c,d)です。
- タスクの代わりにファンクションを使用するか、タスクに引数を追加しなければなりません。以下のようにファンクションに変更するのが簡単です。ファンクションはデータフロー解析の対象なので、センシティビティリストとして@(sa,sb,sc,a,b,c,d)が生成されます。

```
function void select(v,w,x,y);
        if( sa == 1 )
                out = v;
        else if( sb == 1 )
                out = w;
        else if( sc == 1 )
                out = x;
        else
                out = y;
endfunction
```

例題ー3　非同期信号の判定

- 下記の記述は正しいですか？

```
module design(input clk,set,reset,d,output logic q);

always_ff @(posedge clk,negedge reset,negedge set)
        if( !reset )
                q <= 1'b0;
        else if( set )
                q <= 1'b1;
        else
                q <= d;

endmodule
```

例題の解答と解説

- 与えられた記述は正しくありません。
- (negedge set)が指定されているので、非同期信号setに関するイベントが起きた時には、set==1'b0の筈です。したがって、else if(set)の記述に間違いがあります。以下のように訂正する必要があります。

```
module design(input clk,set,reset,d,output logic q);

always_ff @(posedge clk,negedge reset,negedge set)
        if( !reset )
                q <= 1'b0;
        else if( !set )
                q <= 1'b1;
        else
                q <= d;

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。SystemVerilogに関する知識の確認には、文献[2-4]を参照して下さい。今回の基礎講座の知識を確実に理解するためには文献[4]をすすめます。

[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.