

Open-It FPGA トレーニングコース

演習 I2 手順書

内田 智久 (KEK IPNS Esys)

Rev. 1.5

2018 年 8 月 16 日

1 演習内容説明

最初に講師が演習内容について説明しますので、説明を聞き演習で設計する回路の内容について理解してください。何を設計すれば良いのか理解していないと演習を正しく進める事ができません。分からないことがあれば些細な事でも構いませんので質問してください。演習時間になったら手順に従って演習を進めてください。設計やソースコードの記述で調べても分からない時は最後にソースコードを解答例として載せましたので参考にしながら先へ進んでください。

1.1 概要

演習 I2 は演習 I1 の結果を修正する事で進めます。終わっていない方は終わらせてからこの演習を行ってください。

1.2 回路設計

Pattern generator を設計します。

手順 1 下の真理値表を完成させてください。AN[7:4] は 1 に固定なので省略します。

Q[1]	Q[0]	AN[3]	AN[2]	AN[1]	AN[0]
0	0	1	1	1	0
0	1				
1	0				
1	1				

手順 2 作成した真理値表を見ながら、各出力毎に論理式を書いてください。

AN[7:4] は固定値なので下のようになります。

```
assign AN[7:4] = 4'b1111;
```

以下、AN[0] から AN[3] についての式を書いてください。

```
assign AN[0] =
```

```
assign AN[1] =
```

```
assign AN[2] =
```

assign AN[3] =

新しいモジュールを追加して設計した回路を記述します。

1.3 S1 プロジェクトを開く

講義資料「4.2 論理シミュレーション」を見ながら演習 S1 の Vivado プロジェクトを開いてください。

1.4 モジュール作成

Pulse genrator のモジュール「S1_PG」を新たに作ります。

手順 1 講義資料「5 階層構造設計」を見ながら以下の内容で新しいモジュール「S1_PG」を作成してください。

- ファイル名
 - File type: Verilog
 - File name: S1_PG
 - File location: Local to Project
- Define Module (IO ポート)

Port Name	Direction	Bus	MSB	LSB
Q	input	チェックを入れる	1	0
AN	output	チェックを入れる	7	0

手順 2 Project Manager の Source 窓の Hierachy 画面で新しいモジュールが追加された事を確認してください。

手順 3 S1_PG.v を開いてください。

手順 4 設計した論理式を S1_PG.v モジュールの endmodule の前に書いてください。

手順 5 修正が終わったら保存してください (エラー等ないことを確認、エラー時は修正の事)。

手順 6 S1.v に S1_PG モジュールを組み込んでください。インスタンス名は自由に名付けてください。適当な名前が思いつかない時は U3 としてください。

PG モジュールポート名	S1 モジュールでの接続信号
Q	U1 の Q[29:28]
AN	S1 モジュールの出力ポート AN に直接接続

組み込む際に S1 モジュールの AN に関する assign 文の削除を忘れないでください (PG モジュールの信号と二重定義されるのでエラーになる)。

手順 7 修正が終わったら保存してください (エラーが無い事を確認)。

手順 8 「Source」画面で S1 モジュールの下に新しいモジュールが組み込まれていることを確認してください。

以上でコード作成は終了です。

1.5 論理合成から実機確認

データを生成し FPGA にデータをダウンロードして動作確認してください。期待通りの動作をするか確認してください。ここでは、シミュレーションは省略しても構いません。興味がある方はシミュレーションしてください。

2 解答例

以下の回答は一つの例です。皆さんが設計した回路や記述と異なっているかもしれませんが、異なってもシミュレーションと実機でも正しく動作しているのなら、それも正解です。以下の例は参考程度に見てください。

2.1 Verilog ソースコード

2.1.1 S1.v

```
'timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2015/06/25 10:45:42
// Design Name:
// Module Name: S1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////

module S1(
    input OSC,
    input RST_SWn,
    output [7:0] AN,
    output CA,
    output CB,
    output CC,
    output CD,
    output CE,
    output CF,
    output CG
);

    wire [31:0] Q;
```

```
S1_SYNC_COUNTER U1(  
    .CLK    (OSC),  
    .RSTn   (RST_SWn),  
    .Q      (Q[31:0])  
);  
  
S1_ENCODER U2(  
    .I  (Q[27:24]),  
    .CA (CA),  
    .CB (CB),  
    .CC (CC),  
    .CD (CD),  
    .CE (CE),  
    .CF (CF),  
    .CG (CG)  
);  
  
S1_PG U3(  
    .Q (Q[29:28]),  
    .AN (AN[7:0])  
);  
  
endmodule
```

2.1.2 S1_PG.v

```
'timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2015/06/26 16:25:10
// Design Name:
// Module Name: S1_PG
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////
```

```
module S1_PG(
    input [1:0] Q,
    output [7:0] AN
);

    assign AN[7:4] = 4'b1111;

    assign AN[0] = ~(~Q[1] & ~Q[0]);
    assign AN[1] = ~(~Q[1] & Q[0]);
    assign AN[2] = ~(Q[1] & ~Q[0]);
    assign AN[3] = ~(Q[1] & Q[0]);

endmodule
```