

Open-It FPGA トレーニングコース 演習 S1 手順書

内田 智久 (KEK IPNS Esys)

Rev. 1.2

2015 年 12 月 4 日

1 演習 S1 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。

2 演習 S1 手順

演習時間になったら下の手順に従って演習を進めてください。分からない事、不明な事があれば隣近所の方と相談して解決するか、講師に質問してください。手順書の内容が間違っていることもあります。疑問があれば気楽に質問してください。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。

2.1 回路設計

カウンター回路は既に実習しましたので、回路設計はありません。講義資料「4.2 論理シミュレーション」で記述したカウンターと演習 C1 の内容を組み合わせてソースコードを作成します。次へ進んでください。

2.2 Vivado プロジェクト作成

講義資料「2.2 HDL 入力と RTL 解析」を見ながら Vivado プロジェクトを作成してください。プロジェクト作成時に以下の内容を設定してください。

- プロジェクト名
 - Project name: S1
 - Project location: C:¥Temp¥FPGA_Seminar
- デバイス選択
 - xc7a100tcs324-1

プロジェクトナビゲータ画面が開いたら次のソースコード作成へ進んでください。

2.3 新しいソースコード作成

講義資料「2.2 HDL 入力と RTL 解析」を見ながら新しい Verilog ソースを作成してください。

- 設定項目
 - File type: Verilog
 - File name: S1
 - Module name: S1

I/O ポート定義は下を設定してください。同じ作業を繰り返しますが、練習ですので再度入力お願いします。

Port Name	Direction	Bus	MSB	LSB
OSC	input			
RST_SWn	input			
AN	output	チェックを入れる	7	0
CA	output			
CB	output			
CC	output			
CD	output			
CE	output			
CF	output			
CG	output			

2.4 コード作成と RTL 解析

手順 1 講義資料「4.2 論理シミュレーション」を見ながら新しく作成した「S1」モジュールに 28bit カウンターを記述してください。ビット数に変更になる以外は実習時と同じです。

- クロックは OSC 入力
- リセットは RST_SWn 入力、0 の時リセット有効
- カウンター信号名は sync_counter

手順 2 演習 C1 で作成したソースコードを組み込みます。以下の手順に従い進めてください。

- カウンター記述の下で演習 C1 で作成した回路の入力 I とカウンター信号 sync_counter を接続してください。ヒントは下です。
 - wire 記述を使って内部信号 I を宣言
 - assign 文を使って I[3:0] と sync_counter[27:24] を接続

手順 3 演習 C1 で作成したコードの一部をコピーする。

- Vivado の上部メニュー [File] から [Open File] を選択
- ファイル選択画面が表示されるので C:\Temp\FPGA_Seminar\C1..srcs\srcs_1\new\C1.v を選択
- Vivado の右側の窓に C1.v が表示されるので、ポートリストの下から endmodule の手前まで選択、Ctrl+C または右クリックで開くメニューから copy を選択してコピーする
- Vivado の右側の窓の表示を S1.v へ切り替え、endmodule の手間にペーストする
- Vivado の右側の窓の C1.v は閉じる（間違っても変更してはいけません）

以上でソースコードが完成しました。RTL 解析を行ってください。

2.5 論理シミュレーション

講義資料「4.2 論理シミュレーション」を見ながら、「S1」モジュールのテストベンチを新たに作成し論理シミュレーションを行ってください。

- テストベンチの名前等

- File type: Verilog
- File name: S1_TB
- Module name: S1_TB

先ほどの実習で登場した LED15 と同様に I が変化するまでシミュレーションするのは時間がかかり大変です。先ほどの実習と同じように sync_counter の下位ビットが変化していることを確認する事でシミュレーションを終了してください。シミュレーション画面で波形が青色になっている信号がない事を確認してください。また、リセット解除後もしばらく赤色の線になっている場合も何か問題があります。このような事を見つけたらソースコードを見直して修正してください。

正しく動作していることを確認したら次の「論理合成から動作確認」へ進んでください。

2.6 論理合成から動作確認

手順1 講義資料「4.3 FPGA への実装」を見ながら論理合成を行い、生成された回路の回路図を確認してください (RTL 解析)。

手順2 講義資料を見ながら下のようにピン指定を行ってください。制約ファイル名は「S1」とします。

Name	Site	I/O Std.	Drive Str.	Off chip termination	Pull type	Slew
OSC	E3	LVCMOS33		NONE	NONE	
RST_SWn	C12	LVCMOS33		NONE	NONE	
AN[0]	N6	LVCMOS33	12	NONE	NONE	Slow
AN[1]	M6	LVCMOS33	12	NONE	NONE	Slow
AN[2]	M3	LVCMOS33	12	NONE	NONE	Slow
AN[3]	N5	LVCMOS33	12	NONE	NONE	Slow
AN[4]	N2	LVCMOS33	12	NONE	NONE	Slow
AN[5]	N4	LVCMOS33	12	NONE	NONE	Slow
AN[6]	L1	LVCMOS33	12	NONE	NONE	Slow
AN[7]	M1	LVCMOS33	12	NONE	NONE	Slow
CA	L3	LVCMOS33	12	NONE	NONE	Slow
CB	N1	LVCMOS33	12	NONE	NONE	Slow
CC	L5	LVCMOS33	12	NONE	NONE	Slow
CD	L4	LVCMOS33	12	NONE	NONE	Slow
CE	K3	LVCMOS33	12	NONE	NONE	Slow
CF	M2	LVCMOS33	12	NONE	NONE	Slow
CG	L6	LVCMOS33	12	NONE	NONE	Slow

手順3 再度論理合成を行ってください。

手順4 配置配線、データファイルの生成を行ってください。

手順5 最後にデータを FPGA に書き込み動作確認してください (リセットスイッチを押下して動作がリセットされることも確認してください)。

3 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。

3.1 Verilog ソースコード

```
'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
);

    reg [27:0] sync_counter;
```

```

always @(posedge OSC or negedge RST_SWn)begin
    if(!RST_SWn)begin
        sync_counter[27:0] <= 28'd0;
    end else begin
        sync_counter[27:0] <= sync_counter[27:0] + 28'd1;
    end
end

end

wire [3:0] I;

assign I[3:0] = sync_counter[27:24];

assign AN[7:0] = 8'b1111_1110;

assign CA = (~I[3] & ~I[2] & ~I[1] & I[0]) |
            (~I[3] & I[2] & ~I[1] & ~I[0]) |
            ( I[3] & ~I[2] & I[1] & I[0]) |
            ( I[3] & I[2] & ~I[1] & I[0]) ;

assign CB = (~I[3] & I[2] & ~I[1] & I[0]) |
            (~I[3] & I[2] & I[1] & ~I[0]) |
            ( I[3] & ~I[2] & I[1] & I[0]) |
            ( I[3] & I[2] & ~I[1] & ~I[0]) |
            ( I[3] & I[2] & I[1] & ~I[0]) |
            ( I[3] & I[2] & I[1] & I[0]) ;

assign CC = (~I[3] & ~I[2] & I[1] & ~I[0]) |
            ( I[3] & I[2] & ~I[1] & ~I[0]) |
            ( I[3] & I[2] & I[1] & ~I[0]) |
            ( I[3] & I[2] & I[1] & I[0]) ;

assign CD = (~I[3] & ~I[2] & ~I[1] & I[0]) |
            (~I[3] & I[2] & ~I[1] & ~I[0]) |
            (~I[3] & I[2] & I[1] & I[0]) |
            ( I[3] & ~I[2] & I[1] & ~I[0]) |
            ( I[3] & I[2] & I[1] & I[0]) ;

assign CE = (~I[3] & ~I[2] & ~I[1] & I[0]) |
            (~I[3] & ~I[2] & I[1] & I[0]) |
            (~I[3] & I[2] & ~I[1] & ~I[0]) |
            (~I[3] & I[2] & ~I[1] & I[0]) |
            (~I[3] & I[2] & I[1] & I[0]) |
            ( I[3] & ~I[2] & ~I[1] & I[0]) ;

assign CF = (~I[3] & ~I[2] & ~I[1] & I[0]) |
            (~I[3] & ~I[2] & I[1] & ~I[0]) |
            (~I[3] & ~I[2] & I[1] & I[0]) |
            (~I[3] & I[2] & I[1] & I[0]) |
            ( I[3] & I[2] & ~I[1] & I[0]) ;

assign CG = (~I[3] & ~I[2] & ~I[1] & ~I[0]) |

```

```
(~I[3] & ~I[2] & ~I[1] & I[0]) |  
(~I[3] & I[2] & I[1] & I[0]) |  
( I[3] & I[2] & ~I[1] & ~I[0]) ;
```

endmodule

3.2 テストベンチコード

```
'timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2015/06/25 11:20:39
// Design Name:
// Module Name: S1_TB
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

module S1_TB;

    reg CLK100M;
    reg RST_SWn;

    wire [7:0] AN;

    wire CA;
    wire CB;
    wire CC;
    wire CD;
    wire CE;
    wire CF;
    wire CG;

    // ちなみに上はこんな風にも書けます
    // wire CA, CB, CC, CD, CE, CF, CG;

    S1 uut(
        .OSC(CLK100M),
        .RST_SWn(RST_SWn),
        .AN(AN),
```

```
.CA(CA),
.CB(CB),
.CC(CC),
.CD(CD),
.CE(CE),
.CF(CF),
.CG(CG)
);

parameter PERIOD = 10;

always begin
    CLK100M = 1'b0;
    #(PERIOD/2);
    CLK100M = 1'b1;
    #(PERIOD/2);
end

initial begin
    RST_SWn = 1'b0;
    #700 RST_SWn = 1'b1;
end

endmodule
```

3.3 制約ファイル (XDC ファイル)

XDC ファイル内の行順は結果に影響しません。

```
set_property PACKAGE_PIN N6 [get_ports {AN[0]}]
set_property PACKAGE_PIN M6 [get_ports {AN[1]}]
set_property PACKAGE_PIN M3 [get_ports {AN[2]}]
set_property PACKAGE_PIN N5 [get_ports {AN[3]}]
set_property PACKAGE_PIN N2 [get_ports {AN[4]}]
set_property PACKAGE_PIN N4 [get_ports {AN[5]}]
set_property PACKAGE_PIN L1 [get_ports {AN[6]}]
set_property PACKAGE_PIN M1 [get_ports {AN[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[0]}]
set_property PACKAGE_PIN L3 [get_ports CA]
set_property PACKAGE_PIN N1 [get_ports CB]
set_property PACKAGE_PIN L5 [get_ports CC]
set_property PACKAGE_PIN L4 [get_ports CD]
set_property PACKAGE_PIN K3 [get_ports CE]
set_property PACKAGE_PIN M2 [get_ports CF]
set_property PACKAGE_PIN L6 [get_ports CG]
set_property PACKAGE_PIN E3 [get_ports OSC]
set_property PACKAGE_PIN C12 [get_ports RST_SWn]

set_property IOSTANDARD LVCMOS33 [get_ports CA]
set_property IOSTANDARD LVCMOS33 [get_ports CB]
set_property IOSTANDARD LVCMOS33 [get_ports CC]
set_property IOSTANDARD LVCMOS33 [get_ports CD]
set_property IOSTANDARD LVCMOS33 [get_ports CE]
set_property IOSTANDARD LVCMOS33 [get_ports CF]
set_property IOSTANDARD LVCMOS33 [get_ports CG]
set_property IOSTANDARD LVCMOS33 [get_ports RST_SWn]
set_property IOSTANDARD LVCMOS33 [get_ports OSC]

set_property OFFCHIP_TERM NONE [get_ports AN[0]]
set_property OFFCHIP_TERM NONE [get_ports AN[1]]
set_property OFFCHIP_TERM NONE [get_ports AN[2]]
set_property OFFCHIP_TERM NONE [get_ports AN[3]]
set_property OFFCHIP_TERM NONE [get_ports AN[4]]
set_property OFFCHIP_TERM NONE [get_ports AN[5]]
set_property OFFCHIP_TERM NONE [get_ports AN[6]]
```

```
set_property OFFCHIP_TERM NONE [get_ports AN[7]]
create_clock -period 10.000 -name OSC -waveform {0.000 5.000} [get_ports OSC]
```
