

Open-It FPGA training course
Practice I2 manual

Tomohisa Uchida (KEK IPNS Esys)

Rev. 1.5
2018/08/16

Translated by Yun-Tsung Lai (KEK IPNS Esys)
2024/06/20

1. Explanation on the content of C1

First, the content will be explained during the lecture, so please listen to the explanation and understand the content of the circuit for the exercise. Please feel free to ask any questions, even if they are trivial. Otherwise, you cannot proceed to design if you don't understand what to do. The source codes as an example answer are included at the end. You can refer to it if you don't understand the content.

1.1 Introduction

I2 is based on your result on I1, so please finish I1 first to proceed.

1.2 Circuit design

Design the pattern generator.

Step 1 Finish the below truth table. AN[7:4] is ignored since they are fixed as 1.

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

Step 2 Refer to the truth table to write down the logic expression below.

AN[7:4] is fixed:

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

Please fill in from AN[0] to AN[3] below:

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

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

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

```
assign AN[3] =
```

1.3 Open the project for S1

Refer to the lecture material "4.2 Logic simulation" to open the Vivado project for S1.

1.4 Create module

Create a module "S1_PG" for pulse generator.

Step 1 Refer to the lecture material "6 Hierarchical structure design" to make a new module "S1 PG" based on the following content.

- File name
 - File type: Verilog
 - File name: S1_PG
 - File location: Local to project
- Define module (I/O port)

Port Name	Direction	Bus	MSB	LSB
Q	input	Check	1	0
AN	output	Check	7	0

Step 2 Confirm that these two new modules have been added in the Hierarchy screen of the Project Manager's Source window.

Step 3 Open S1_PG.v

Step 4 Put the logic expressions finished above in the code of S1_PG.v above the endmodule line.

Step 5 Save the files when you finish them. (Check if there is no error.)

Step 6 Incorporate S1_PG.v into S1.v. The instance can be named freely. You can simply use U3.

Port name of PG module	Connected signal at S1 module
Q	Q[29:28] of U1
AN	Connect to AN (output port of A1) directly

When incorporating, do not forget to delete the assign statement related to AN in the S1 module (an error will occur because it is defined twice with the signal in the PG module).

Step 7 Save the files when you finish them. (Check if there is no error.)

Step 8 Check the Source window to see if the two modules are incorporated under S1 or not.

1.5 Synthesis to implementation

Generate the data and test it in FPGA. The operation should be the same as the one from S1. It is fine to ignore simulation for now. If you are interested in it, you can also do it.

2. Example of answers

The answers below are just one of the example. It might be different from yours. Please just refer to them. If yours can be working in simulation and FPGA, it is no problem.

2.1 Verilog source codes

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
```
