

Open-It FPGA training course
Practice C1 manual (Nexys-4 board)

Tomohisa Uchida (KEK IPNS Esys)

Rev. 3.0
2017/03/15

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.

2. Procedures of C1

When it is time for the exercise, please proceed by following the steps below. If you have any questions or concerns, please discuss with your neighbors. Sometimes the content of the material might be wrong, so feel free to ask the instructor for any questions.

The example source codes are also put below. You can refer to them to proceed.

2.1 Circuit design

Step 1.1 Finish the truth table below.

Notice: The LED is ON when inputting 0.

I[3]	I[2]	I[1]	I[0]	CA	CB	CC	CD	CE	CF	CG
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1							
0	0	1	0							
0	0	1	1							
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							
1	0	1	0							
1	0	1	1							
1	1	0	0							
1	1	0	1							
1	1	1	0							
1	1	1	1							

Step 1.2 Make the logic expression for the output based on the truth table.

EX: AN[7]:

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

Below, please write down CA to CG:

assign CA =

assign CB =

assign CC =

assign CD =

assign CE =

assign CF =

assign CG =

2.2 Create Vivado project

Refer to the lecture material "2.2 HDL input and RTL analysis" to create new Vivado project.
Use the following setups to create the project:

- Project name
 - Project name: C1
 - Project location: C:\Temp\FPGA_Seminar

- Device selection
 - xc7a100tcsq324-1

After the project is created, proceed to make source code.

2.3 Create new source code

Refer to the lecture material "2.2 HDL input and RTL analysis" to make new Verilog sources.

- Items to set up
 - File type: Verilog
 - File name: C1
 - Module name: C1

Refer to below for the definitions of the I/O ports:

Port name	Direction	Bus	MSB	LSB
I	input	check	3	0
AN	output	check	7	0
CA	output			
CB	output			
CC	output			
CD	output			
CE	output			
CF	output			
CG	output			

2.4 Create code and RTL analysis

Refer to the lecture material "2.2 HDL input and RTL analysis" to make new C1 module, input the logic expressions, and do RTL analysis..

2.5 Logic simulation

Refer to the lecture material "2.3 Functional simulation" to make a test bench for the C1 module, and perform simulation.

- Info of test bench:
 - File type: Verilog
 - File name: C1_TB
 - Module name: C1_TB

In the test bench, change the input value to the I[3:0] port from 0x0 to 0xF, and check if the output is consistent with the truth table.

Supplemental info: When incorporating C1 into the test bench, connect the port as below.

```
C1 uut(  
  .I  (I[3:0] ),  
  .AN (AN[7:0] ),  
  .CA (CA ),  
  .CB (CB ),  
  .CC (CC ),  
  .CD (CD ),  
  .CE (CE ),  
  .CF (CF ),  
  .CG (CG )  
);
```

Notice how the signals I and AN in C1.v are connected. Although it is a multi-bit port, do not specify bits in the port name of the called module.

Proceed to the next step is simulation is working correctly.

2.6 Check the operation from synthesis

Step 1 Refer to the lecture manual "2.4 Implementation on FPGA" to do synthesis, and check the produced circuit diagram (RTL analysis).

Step 2 Please assign the pints as below while referring to the lecture materials. The constraint file name is "C1".

Name	Package Pin	I/O Std.	Drive Str.	Off Chip termination	Pull type	Stew
I[0]	U9	LVCMOS33		NONE	NONE	
I[1]	U8	LVCMOS33		NONE	NONE	
I[2]	R7	LVCMOS33		NONE	NONE	
I[3]	R6	LVCMOS33	12	NONE	NONE	Slow
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

Step 3 Do synthesis again.

Step 4 Proceed place & route, and data generation.

Step 5 Download the data file to FPGA, change the SW, and see the operation result.

3. 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.

3.1 Truth table

I[3]	I[2]	I[1]	I[0]	CA	CB	CC	CD	CE	CF	CG
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	0	1	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	1	0	0
1	0	1	0	0	0	0	1	0	0	0
1	0	1	1	1	1	0	0	0	0	0
1	1	0	0	0	1	1	0	0	0	1
1	1	0	1	1	0	0	0	0	1	0
1	1	1	0	0	1	1	0	0	0	0
1	1	1	1	0	1	1	1	0	0	0

3.2 Verilog source code

```
'timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2015/06/23 12:00:24
// Design Name:
// Module Name: C1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
module C1(
    input [3:0] I,
    output [7:0] AN,
    output CA,
    output CB,
    output CC,
    output CD,
    output CE,
    output CF,
    output CG
);

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

```

3.3 Test bench code

```
'timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2015/11/05 16:59:42
// Design Name:
// Module Name: C1_TB
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////

module C1_TB;

reg [3:0] I ;

wire [7:0] AN ;
wire CA ;
wire CB ;
wire CC ;
wire CD ;
wire CE ;
wire CF ;
wire CG ;

C1 uut(
    .I (I[3:0] ),
    .AN (AN[7:0] ),
    .CA (CA ),
    .CB (CB ),
    .CC (CC ),
    .CD (CD ),
```

```
.CE (CE ),  
.CF (CF ),  
.CG (CG )  
);  
  
initial begin  
    I[3:0] = 4'b0000;  
    #50;  
    I[3:0] = 4'b0001;  
    #50;  
    I[3:0] = 4'b0010;  
    #50;  
    I[3:0] = 4'b0011;  
    #50;  
    I[3:0] = 4'b0100;  
    #50;  
    I[3:0] = 4'b0101;  
    #50;  
    I[3:0] = 4'b0110;  
    #50;  
    I[3:0] = 4'b0111;  
    #50;  
    I[3:0] = 4'b1000;  
    #50;  
    I[3:0] = 4'b1001;  
    #50;  
    I[3:0] = 4'b1010;  
    #50;  
    I[3:0] = 4'b1011;  
    #50;  
    I[3:0] = 4'b1100;  
    #50;  
    I[3:0] = 4'b1101;  
    #50;  
    I[3:0] = 4'b1110;  
    #50;  
    I[3:0] = 4'b1111;  
end  
  
endmodule
```

3.4 Constraint file (XDC file)

The order of lines in XDC file does not matter.

```
set_property PACKAGE_PIN U9 [get_ports {I[0]}]
set_property PACKAGE_PIN U8 [get_ports {I[1]}]
set_property PACKAGE_PIN R7 [get_ports {I[2]}]
set_property PACKAGE_PIN R6 [get_ports {I[3]}]
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 IOSTANDARD LVCMOS33 [get_ports {I[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {I[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {I[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {I[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports CA]
set_property IOSTANDARD LVCMOS33 [get_ports CB]
set_property IOSTANDARD LVCMOS33 [get_ports CD]
set_property IOSTANDARD LVCMOS33 [get_ports CC]
set_property IOSTANDARD LVCMOS33 [get_ports CE]
set_property IOSTANDARD LVCMOS33 [get_ports CF]
set_property IOSTANDARD LVCMOS33 [get_ports CG]
set_property OFFCHIP_TERM NONE [get_ports CA]
set_property OFFCHIP_TERM NONE [get_ports CB]
set_property OFFCHIP_TERM NONE [get_ports CC]
set_property OFFCHIP_TERM NONE [get_ports CD]
set_property OFFCHIP_TERM NONE [get_ports CE]
set_property OFFCHIP_TERM NONE [get_ports CF]
set_property PACKAGE_PIN M1 [get_ports {AN[7]}]
set_property PACKAGE_PIN L1 [get_ports {AN[6]}]
set_property PACKAGE_PIN N4 [get_ports {AN[5]}]
set_property PACKAGE_PIN N2 [get_ports {AN[4]}]
set_property PACKAGE_PIN N5 [get_ports {AN[3]}]
set_property PACKAGE_PIN M3 [get_ports {AN[2]}]
set_property PACKAGE_PIN M6 [get_ports {AN[1]}]
set_property PACKAGE_PIN N6 [get_ports {AN[0]}]
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]}]
```
