

Open-It FPGA training course
Reference for practice (Vivado tool/Nexys-4 board)

Tomohiss Uchida (KEK IPNS Esys)

Rev. 4.2
2017/10/19

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

1 The HDL code written in 2.2 (TEST.v)

```
'timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2014/07/31 18:02:58
// Design Name:
// Module Name: TEST
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

module TEST(
    input SW_A,
    input SW_B,
    output LED0
);

assign LED0 = SW_A & SW_B;

endmodule
```

2 The test bench HDL code written in 2.3 (TEST_TB.v)

```
'timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2014/08/01 11:44:10
// Design Name:
// Module Name: TEST_TB
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

module TEST_TB;

reg SW_A;
reg SW_B;
wire LED0;

TEST uut(
    .SW_A(SW_A),
    .SW_B(SW_B),
    .LED0(LED0)
);

initial begin
    SW_A = 1'b0;
    SW_B = 1'b0;
    #100 SW_A = 1'b1;
    #300 SW_B = 1'b1;
    #200 SW_A = 1'b0;
end

endmodule
```

3 The pin list used in 2.4

Name	Package Pin	I/O Std.	Drive Str.	Off Chip termination	Pull type	Stew
SW_A	U9	LVCMOS33		NONE	NONE	
SW_B	U8	LVCMOS33		NONE	NONE	
SW_C	R7	LVCMOS33		NONE	NONE	
LED0	T8	LVCMOS33	12	NONE	NONE	Slow
LED1	V9	LVCMOS33	12	NONE	NONE	Slow
LED2	R8	LVCMOS33	12	NONE	NONE	Slow

Setup for those used only. For the blanks, no need to input.

4 The HDL code written in 4.2 (TEST.v)

```
'timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2014/08/01 17:17:00
// Design Name:
// Module Name: TEST
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

module TEST(
    input OSC,
    input RST_SWn,
    input SW_A,
    input SW_B,
    output LED0,
    output LED15
);

assign LED0 = SW_A & SW_B;
reg [31:0] sync_counter;

always @(posedge OSC or negedge RST_SWn)begin
    if(!RST_SWn)begin
        sync_counter[31:0] <= 32'd0;
    end else begin
        sync_counter[31:0] <= sync_counter[31:0] + 32'd1;
    end
end

assign LED15 = sync_counter[28];

endmodule
```

5 The test bench HDL code written in 4.2 (TEST_TBv)

```
'timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2014/08/01 17:17:36
// Design Name:
// Module Name: TEST_TB
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////

module TEST_TB;

reg CLK100M;
reg RST_SWn;
reg SW_A;
reg SW_B;
wire LED0;
wire LED15;

TEST uut(
    .OSC(CLK100M),
    .RST_SWn(RST_SWn),
    .SW_A(SW_A),
    .SW_B(SW_B),
    .LED0(LED0),
    .LED15(LED15)
);

initial begin
    SW_A = 1'b0;
    SW_B = 1'b0;
    #100 SW_A = 1'b1;
```

```
        #300 SW_B = 1'b1;
        #200 SW_A = 1'b0;
end

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
```

6 The pin list used in 4.3

Name	Package Pin	I/O Std.	Drive Str.	Off Chip termination	Pull type	Stew
OSC	E3	LVC MOS33		NONE	NONE	
RST_SWn	C12	LVC MOS33		NONE	NONE	
SW_A	U9	LVC MOS33		NONE	NONE	
SW_B	U8	LVC MOS33		NONE	NONE	
SW_C	R7	LVC MOS33		NONE	NONE	
LED0	T8	LVC MOS33	12	NONE	NONE	Slow
LED1	V9	LVC MOS33	12	NONE	NONE	Slow
LED2	R8	LVC MOS33	12	NONE	NONE	Slow
LED15	P2	LVC MOS33	12	NONE	NONE	Slow

Setup for those used only. For the blanks, no need to input.

7 The HDL code written in 6 (TEST_SYNC_COUNTER.v)

```
'timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2015/06/26 09:03:02
// Design Name:
// Module Name: TEST_SYNC_COUNTER
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////

module TEST_SYNC_COUNTER(
    input CLK,
    input RSTn,
    output LED
);

reg [31:0] sync_counter;

always @(posedge CLK or negedge RSTn)begin
    if(!RSTn)begin
        sync_counter[31:0] <= 32'd0;
    end else begin
        sync_counter[31:0] <= sync_counter[31:0] + 32'd1;
    end
end

assign LED = sync_counter[28];

endmodule
```

8 The HDL code written in 6 (TEST.v)

```
'timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2015/06/23 14:49:45
// Design Name:
// Module Name: TEST
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////

module TEST(
    input OSC,
    input RST_SWn,
    input SW_A,
    input SW_B,
    output LED0,
    output LED15
);

assign LED0 = SW_A & SW_B;

TEST_SYNC_COUNTER U1(
    .CLK (OSC),
    .RSTn (RST_SWn),
    .LED (LED15)
);

endmodule
```

9 The HDL code written in 7 (TEST.v)

```
'timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2015/07/09 15:51:39
// Design Name:
// Module Name: TEST
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////

module TEST(
    input OSC,
    input RST_SWn,
    input SW_A,
    input SW_B,
    output LED0,
    output LED15
);

wire CLK200M;
wire sys_reset;

sys_mmcm mmcm_1
(
    // Clock out ports
    .clk_out1(), // output clk_out1
    .clk_out2(CLK200M), // output clk_out2
    // Status and control signals
    .resetn(RST_SWn), // input resetn
    .locked(sys_reset), // output locked
    // Clock in ports
    .clk_in1(OSC)); // input clk_in1
```

```
assign LED0 = SW_A & SW_B;
```

```
TEST_SYNC_COUNTER U1(  
    .CLK (CLK200M),  
    .RSTn (sys_reset),  
    .LED (LED15)  
);
```

```
endmodule
```
