

Open-It FPGA トレーニングコース 実習時の参考資料(Vivado ツール/Nexys-4 DDR 版)

内田 智久 (KEK IPNS Esys)

Rev. 4.3

2025年 4 月 22 日

Translated by Yun-Tsung Lai (KEK IPNS Esys)

2025/04/22

1 2.2 で入力する HDL コード (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 2.3 で入力するテストベンチHDLコード(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 2.4 で使用するピンリスト

Name	Package Pin	I/O Std.	Drive Str.	Off chip termination	Pull type	Slew
SW_A	J15	LVC MOS33		NONE	NONE	
SW_B	L16	LVC MOS33		NONE	NONE	
SW_C	M13	LVC MOS33		NONE	NONE	
LED0	H17	LVC MOS33	12	NONE	NONE	Slow
LED1	K15	LVC MOS33	12	NONE	NONE	Slow
LED2	J13	LVC MOS33	12	NONE	NONE	Slow

使用している信号のみ設定してください。空欄は設定不要です

4 4.2 で入力する HDL コード(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 4.2 で入力するテストベンチHDLコード(TEST_TB.v)

```
'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 2.4 で使用するピンリスト

Name	Package Pin	I/O Std.	Drive Str.	Off chip termination	Pull type	Slew
OSC	E3	LVC MOS33		NONE	NONE	
RST_SWn	C12	LVC MOS33		NONE	NONE	
SW_A	J15	LVC MOS33		NONE	NONE	
SW_B	L16	LVC MOS33		NONE	NONE	
SW_C	M13	LVC MOS33		NONE	NONE	
LED0	H17	LVC MOS33	12	NONE	NONE	Slow
LED1	K15	LVC MOS33	12	NONE	NONE	Slow
LED2	J13	LVC MOS33	12	NONE	NONE	Slow
LED15	V11	LVC MOS33	12	NONE	NONE	Slow

使用している信号のみ設定してください。空欄は設定不要です

7 6. で入力する HDL コード(TEST SYNC COUNTERv)

```
*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 6. で入力する HDL コード(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 7. で入力する HDL コード(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
```

10 The HDL code written in 7. Second part (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,
    output [31:0] counter_out
);
    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];
    assign counter_out = sync_counter;

endmodule
```

11 The HDL code written in 7. Second part (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;

wire [31:0] counter;
wire vio_reset;

vio_0 vio_inst (
    .clk(CLK200M),           // input wire clk
    .probe_out0(vio_reset)   // output wire [0 : 0] probe_out0
);

TEST_SYNC_COUNTER U1(
    .CLK (CLK200M),
    .RSTn (sys_reset & vio_reset),
    .LED (LED15),
    .counter_out (counter)
);

ila_0 ila_inst (
    .clk(CLK200M),           // input wire clk
    .probe0({16'd0, vio_reset, sys_reset, counter}) // input wire [49:0] probe0
);

endmodule

```
