

## Xilinx FPGA 구현 실습 예제

### 1) 8-b 링 계수기

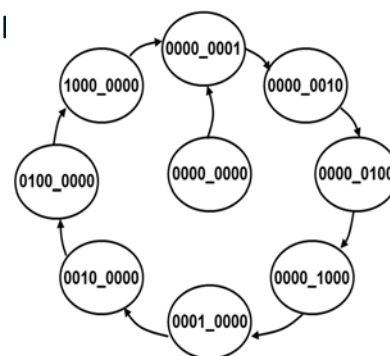
#### □ 링 계수기 (Ring Counter)

❖ 하나의 1이 순환하며 이동하는 계수기

- rst=0일 때, 링 계수기로 동작
- rst=1 또는 레지스터의 값이 0000\_0000일 때, 초기상태 0000\_0001로 복귀
- active-low 비동기식 리셋

❖ 코딩

- ① 시프트 연산자 << 를 사용하는 방법
- ② for 반복문을 사용하는 방법
- ③ 결합 연산자 { } 를 사용하는 방법



8비트 링 계수기의 상태천이도

## 1) 8-b 링 계수기

8

```
module ring_counter (clk, rst, cnt);
    input      clk, rst;
    output [7:0] cnt;
    reg [7:0] cnt;
    reg [18:0] cnt_fdiv;
    reg clk_50;

    always @(posedge clk_50 or negedge rst) begin
        if (!rst)
            cnt <= 1;
        else begin
            if ((cnt == 0) || (cnt == 128))
                cnt <= 1;
            else
                cnt <= cnt << 1;
        end
    end
end
```

코드 1

## 1) 8-b 링 계수기

9

```
// ① clk(50MHz)을 clk_50(50Hz)로 분주, 50% duty cycle
always @(posedge clk or negedge rst) begin
    if(!rst) begin
        cnt_fdiv <= 0;
        clk_50 <= 0;
    end
    else begin
        !! coding !!
    end
end
endmodule
```

코드 1

## 1) 8-b 링 계수기

10

```
module tb_ring_counter;
    reg clk; // Inputs
    reg rst; // Inputs
    wire [7:0] cnt; // Outputs

    // Instantiate the Unit Under Test (UUT)
    ring_counter uut (
        .clk(clk),
        .rst(rst),
        .cnt(cnt));

    initial begin // Initialize Inputs
        clk = 0;
        rst = 0;
        #10 rst = 1;
    end

    always
        #10 clk = ~clk;

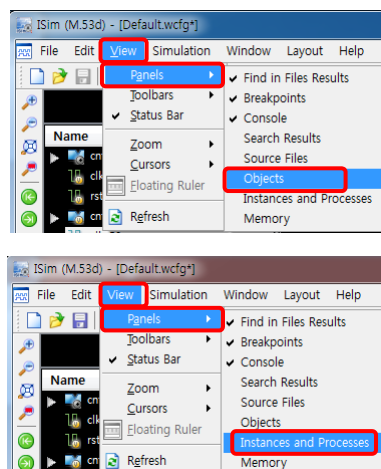
endmodule
```

코드 1

## 1) 8-b 링 계수기

11

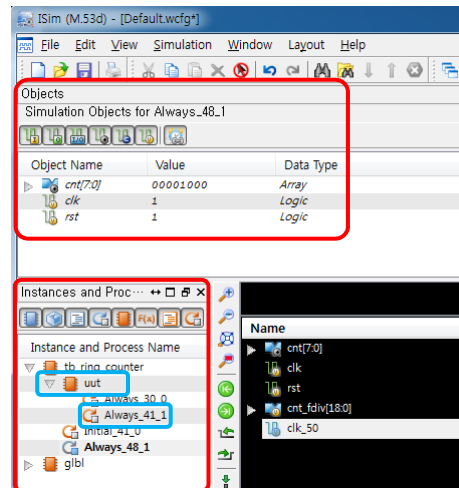
### ❖ ISIM에서 시뮬레이션 결과 object 추가하기



## 1) 8-b 링 계수기

12

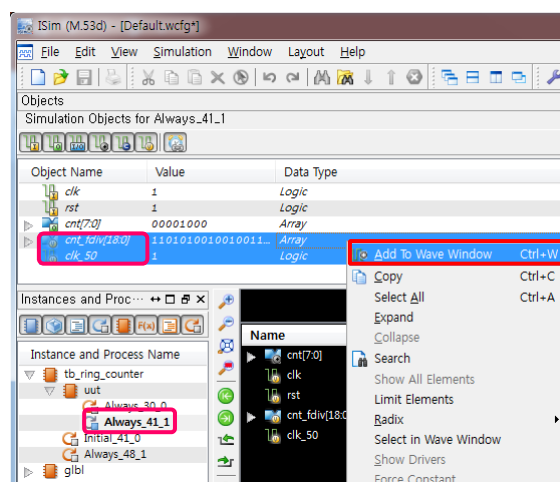
### ❖ ISIM에서 시뮬레이션 결과 object 추가하기



## 1) 8-b 링 계수기

13

### ❖ ISIM에서 시뮬레이션 결과 object 추가하기



## 1) 8-b 링 계수기

14

```
module ring_counter_conop (clk, rst, cnt);
    input      clk, rst;
    output [7:0] cnt;
    reg [7:0] cnt;
    reg [18:0] cnt_fdiv;
    reg      clk_50;

    always @(posedge clk_50 or negedge rst) begin
        * 결합 연산자 { }를 사용하여 코드를 완성한다.
    end
    always @(posedge clk or negedge rst) begin

        // 주파수 분주기

    end

endmodule
```

코드 2

과제-1) 결합연산자를 이용한 링 계수기 (Simulation)

과제-2) 결합연산자를 이용한 링 계수기 (FPGA 동작 확인)

## 1) 8-b 링 계수기

15

신호 이름	핀 번호	
	XC3S400	XC3S200
CLK	T9	P79
RST	F4	P205
LED[0]	N5	P2
LED[1]	P7	P3
LED[2]	T5	P4
LED[3]	T8	P5
LED[4]	T3	P7
LED[5]	R3	P9
LED[6]	T4	P10
LED[7]	R4	P11

## 2) 좌우 시프팅 링 계수기

16

- 좌/우 시프팅 기능을 갖는 링 계수기를 **시프트 연산자 << 와 >>**를 사용하는 방법으로 모델링 (Active-low 비동기식 리셋을 가짐)

rst	mode	cnt	동작
1	0	0000 0000, 0000 0001	1000 0000으로 복귀
		—	오른쪽으로 시프팅
	1	0000 0000, 1000 0000	0000 0001로 복귀
		—	왼쪽으로 시프팅
0	0	—	0000 0001으로 초기화
	1	—	

과제-3) 좌우 시프팅 링 계수기 (FPGA 동작 확인)

## 2) 좌우 시프팅 링 계수기

17

```
module ring_counter2_sop (clk, rst, mode, cnt);
    input      clk, rst, mode;
    output [7:0] cnt;
    reg [7:0] cnt;
    reg [18:0] cnt_fdiv;
    reg      clk_50;
```

\*Active-low 비동기식 리셋을 가짐  
\*rst=1인 정상동작 모드에서 if(mode) 조건문에 의해

- mode=1인 경우;
  - cnt=0 또는 cnt=128이면, 초기상태(0000\_0001)로 복귀
  - 왼쪽 시프트 연산자 <<를 사용하여 시프팅 동작을 구현
- mode=0인 경우;
  - cnt=0 또는 cnt=10이면, 초기상태(1000\_0000)로 복귀
  - 오른쪽 시프트 연산자 >>를 사용하여 시프팅 동작을 구현

endmodule

코드 3

## 2) 좌우 시프팅 링 계수기

18

신호 이름	핀 번호		비고
	XC3S400	XC3S200	
CLK	T9	P79	
RST	F4	P205	
PUSH_1	P2	P18	mode 신호
LED[0]	N5	P2	
LED[1]	P7	P3	
LED[2]	T5	P4	
LED[3]	T8	P5	
LED[4]	T3	P7	
LED[5]	R3	P9	
LED[6]	T4	P10	
LED[7]	R4	P11	