

반복문

Kyung-Wook Shin
kwshin@kumoh.ac.kr

School of Electronic Eng.,
Kumoh National Institute of Technology

Verilog HDL

행위수준 모델링

K. W. SHIN

5.5 반복문

2

□ 반복문

- ❖ **forever** 문 : 문장이 무한히 반복적으로 실행
- ❖ **repeat** 문 : 지정된 횟수만큼 문장이 반복 실행
 - 반복 횟수를 나타내는 수식이 **x** 또는 **z**로 평가되면 반복 횟수는 0이 되고, 문장은 실행되지 않음
- ❖ **while** 문 : 조건식의 값이 거짓이 될 때까지 문장이 반복 실행
 - 조건식의 초기값이 거짓이면 문장은 실행되지 않음
- ❖ **for** 문 : 반복 횟수를 제어하는 변수에 의해 문장이 반복 실행

```
forever statement;
| repeat(expression) statement;
| while(expression) statement;
| for(variable_assign); expression; variable_assign) statement;
```

Verilog HDL

행위수준 모델링

K. W. SHIN

5.5 반복문

3

예 5.5.1 repeat 문을 이용한 shift-add 방식의 승산기

```
module multiplier_8b(opa, opb, result);
  parameter SIZE = 8, LongSize = 2*SIZE;
  input  [SIZE-1:0] opa, opb;
  output [LongSize-1:0] result;
  reg    [LongSize-1:0] result, shift_opa, shift_opb;

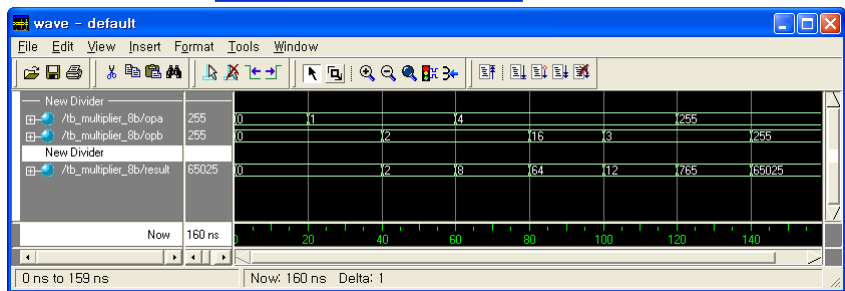
  always @(opa or opb) begin
    shift_opa = opa;      // multiplicand
    shift_opb = opb;      // multiplier
    result = 0;
    repeat(SIZE) begin
      if(shift_opb[0])
        result = result + shift_opa;
      shift_opa = shift_opa << 1;
      shift_opb = shift_opb >> 1;
    end
  end
endmodule
```

코드 5.14

5.5 반복문

4

예 5.5.1 코드 5.14의 시뮬레이션 결과



5.5 반복문

5

예 5.5.2 8비트 입력 rega에 포함된 1을 계수하는 회로

```
module cnt_one(rega, count);
  input  [7:0] rega;
  output [3:0] count;
  reg    [7:0] temp_reg;
  reg    [3:0] count;

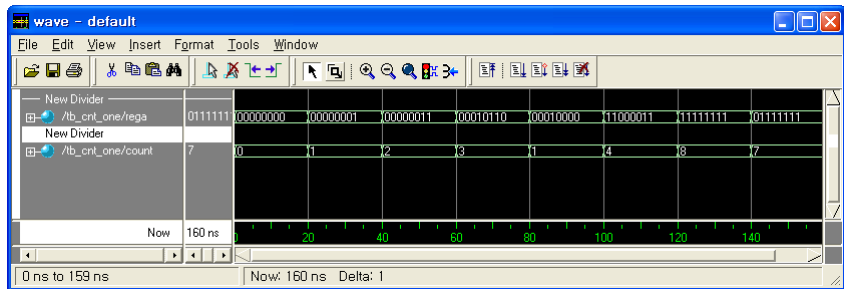
  always @(rega) begin
    count = 0;
    temp_reg = rega;
    while(temp_reg) begin
      if(temp_reg[0])
        count = count + 1;
      temp_reg = temp_reg >> 1;
    end
  end
endmodule
```

코드 5.15

5.5 반복문

6

예 5.5.2 코드 5.15의 시뮬레이션 결과



5.5 반복문

7

예 5.5.3 for 문을 이용한 8비트 우선순위 인코더(priority encoder)

```
module enc_for(in, out);
    input  [7:0] in;
    output [2:0] out;
    reg    [2:0] out;
    integer i;

    always @(in) begin : LOOP
        out=0;
        for(i = 7; i >= 0; i = i-1) begin
            if(in[i]) begin
                out=i;
                disable LOOP;
            end
        end
    end
endmodule
```

입력 in[7:0]	출력 out[2:0]
0000_0001	000
0000_001x	001
0000_01xx	010
0000_1xxx	011
0001_xxxx	100
001x_xxxx	101
01xx_xxxx	110
1xxx_xxxx	111

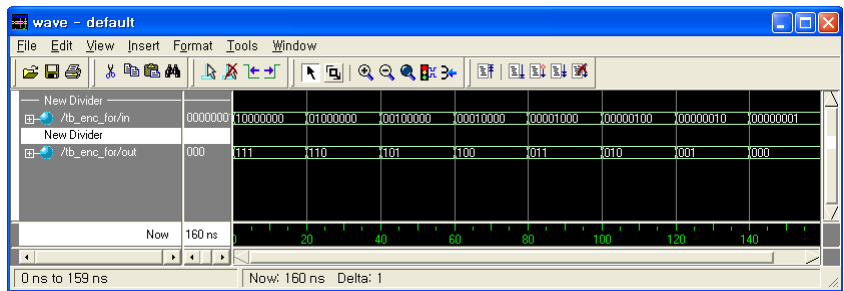
8비트 우선순위 인코더

코드 5.16

5.5 반복문

8

예 5.5.3 코드 5.16의 시뮬레이션 결과



5.5 반복문

9

예 5.5.4

forever 문을 이용한 주기 신호 생성

```
module tb_dff ;  
    reg  clk, din;  
  
    dff U1 (clk, din, q);  
  
    initial begin  
        clk = 1'b0;  
        forever #10 clk = ~clk;  
    end  
  
    initial begin  
        din = 1'b0;  
        forever begin  
            #15 din = 1'b1;  
            #20 din = 1'b0;  
            #30 din = 1'b1;  
            #20 din = 1'b0;  
        end  
    end  
endmodule
```