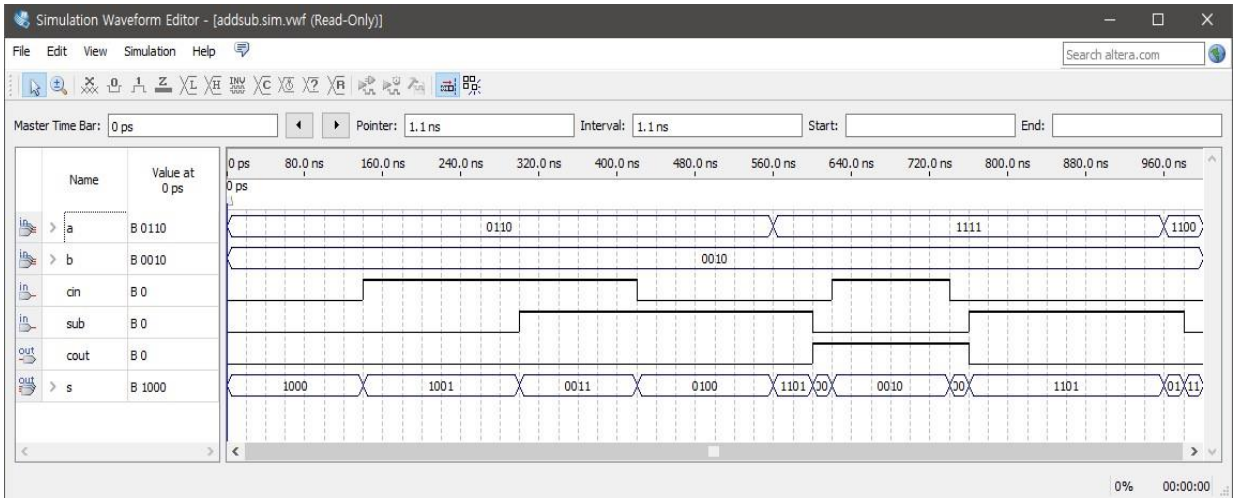
< 실습과제 6 >

2017253041 홍성우

1.

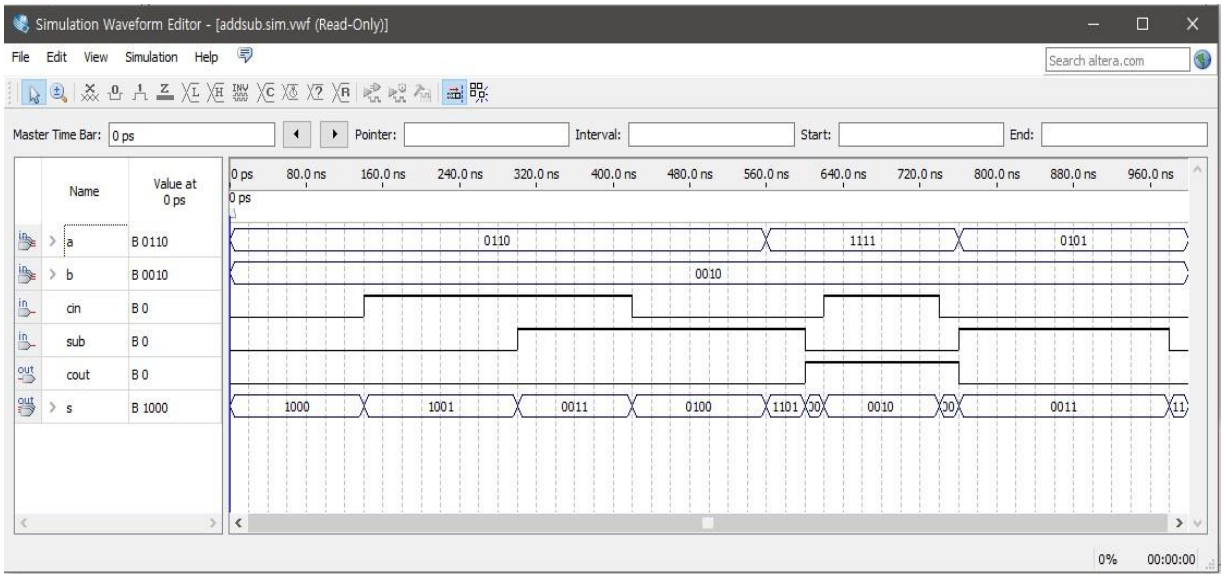
(1)

```
module addsub(sub, a, b, cin, s, cout);
    input sub;
    input [3:0]a, b;
    input cin;
    output [3:0]s;
    output cout;

    wire [3:0] b2;
    wire cin2, cout2;

    assign b2 = sub? ~b: b;
    assign cin2 = sub ? ~cin : cin;
    assign {cout2, s} = a + b2 + cin2; //adder
    assign cout = sub ? ~cout2 : cout2;
endmodule
```



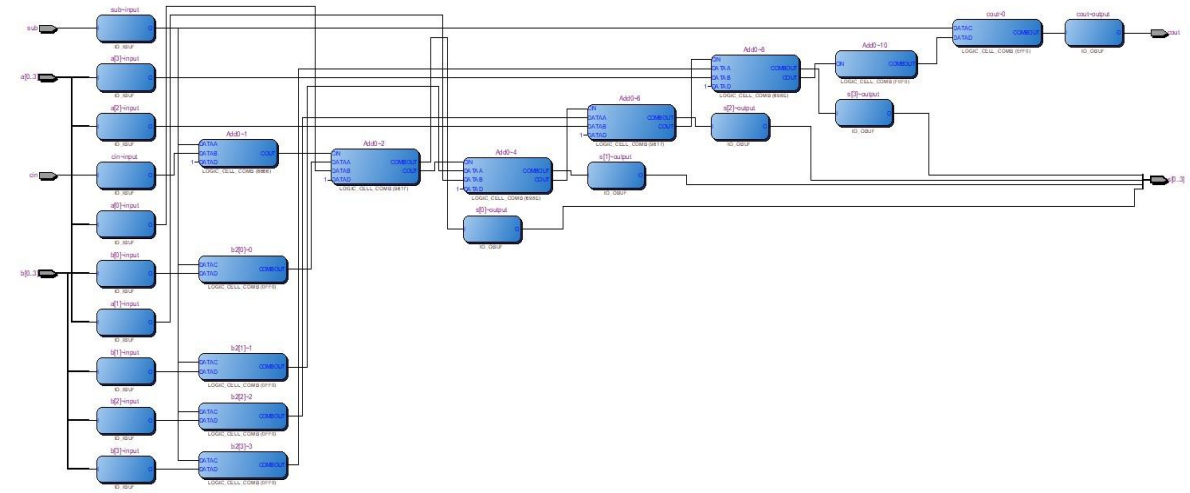
(2)

```
module addsub(sub, a, b, cin, s, cout);
    input sub;
    input [3:0]a, b;
    input cin;
    output [3:0]s;
    output cout;

    assign {cout, s} = sub ? (a - b - cin) : (a + b + cin);
endmodule
```



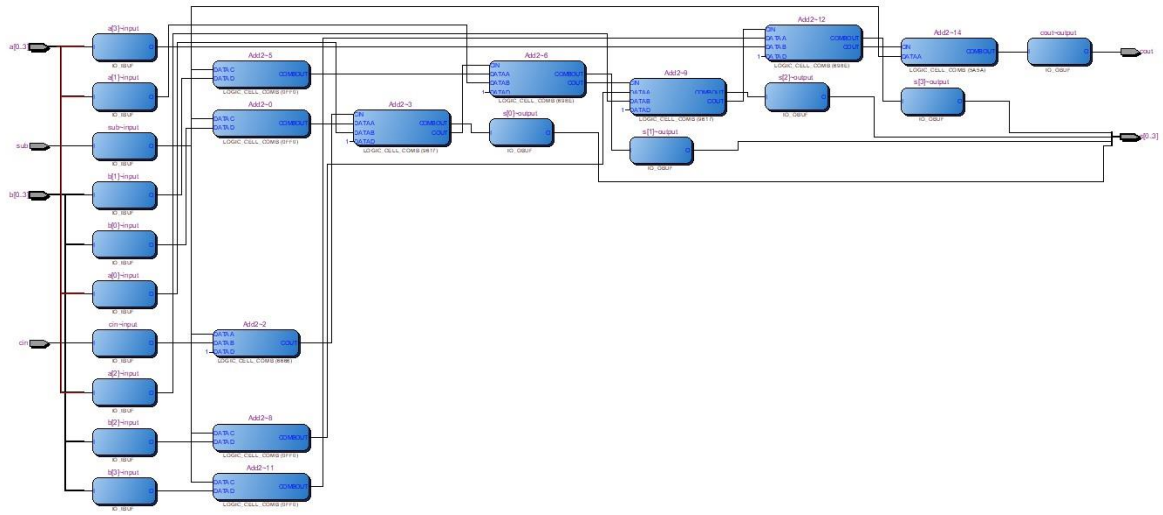
(3)

- 1번 -



Total logic elements	11 / 114,480 (< 1 %)
Total combinational functions	11 / 114,480 (< 1 %)
Dedicated logic registers	0 / 114,480 (0 %)

- 2번 -



Total logic elements	10 / 114,480 (< 1 %)
Total combinational functions	10 / 114,480 (< 1 %)
Dedicated logic registers	0 / 114,480 (0 %)

Technology Map Viewer는 유사하나 Logic Elements에선 2번이 더 적게 사용한다.

2.

(1)

```
module counterN(clk, reset, enable, qout, tc);
   parameter N = 10;        // mod-N counterN
   parameter M = 4;         // M = bit size = log2(N)
   input clk, reset, enable;
   output reg [M-1:0] qout;
   output tc;               // terminal count

   always @(posedge clk or negedge reset) begin
      if (~reset) qout <= 0;
      else if (enable) begin
         if (qout == N-1) qout <= 0;
         else qout <= qout + 1;
      end
   end

   assign tc = enable && (qout == N-1);

endmodule
```

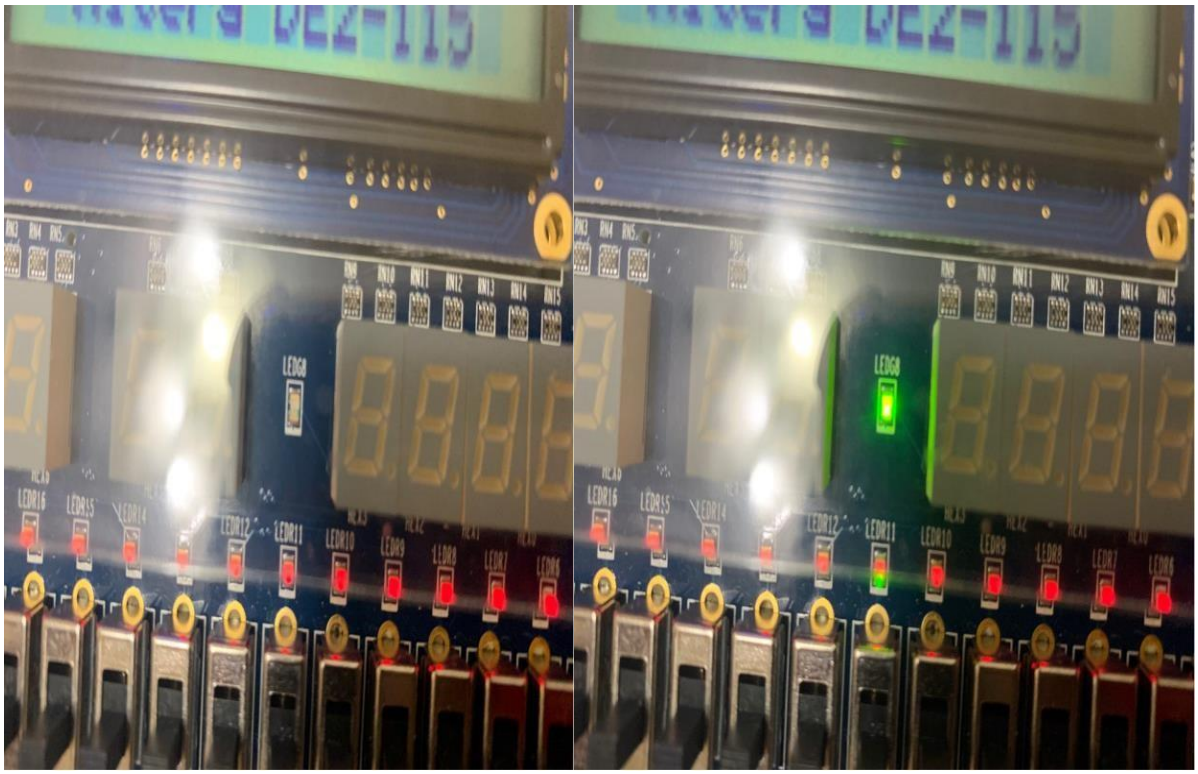
(2)-1

```
module counter25(qout, tc, enable, reset, clk);
    output [32:0] qout;
    output tc;
    input enable, reset, clk;
    reg [32:0] qout;

    always @(posedge clk) begin
        if(~reset) qout <= 0;
        else if(enable) qout <= qout + 1;
    end
    assign tc = (qout == 25000000) & enable;
endmodule
```

(2)-2

```
module counter25(clk50,clk1);
    input wire clk50;
    output reg clk1;
    reg[31:0] cnt;

    assign tc = (cnt == 32'd24999999);

    always @(posedge clk50) begin
        if(tc) begin
            clk1 <= ~clk1;
            cnt <= 32'b0;
        end
        else cnt <= cnt + 32'b1;
    end

endmodule
```

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard
clk1	Output	PIN_F17	7	B7_N2	PIN_F17	2.5 V (default)
clk50	Input	PIN_Y2	2	B2_N0	PIN_Y2	2.5 V (default)
<<new node>>						



3.

4비트 가감산기 설계는 2학년때 배웠던 논리회로설계를 기반으로 해서 비교적 수월했지만 클록 주파수 분주기 설계는 설계 방식을 과제 방식대로 생각해내지 못해서 할 수 있는 최대한 설계했다. 생각보다 어려웠다.