< 실습과제 7 >

2017253041 홍성우

1.

통합모듈 부분의 reset을 ~reset으로 바꿔 설계하는 걸 까먹어서 1번문제만 키트의 reset 신호를 누르면서 진행했습니다. 모든 문제의 핀 번호는 과제에 제시된 그대로 지정해서 사용했습니다.

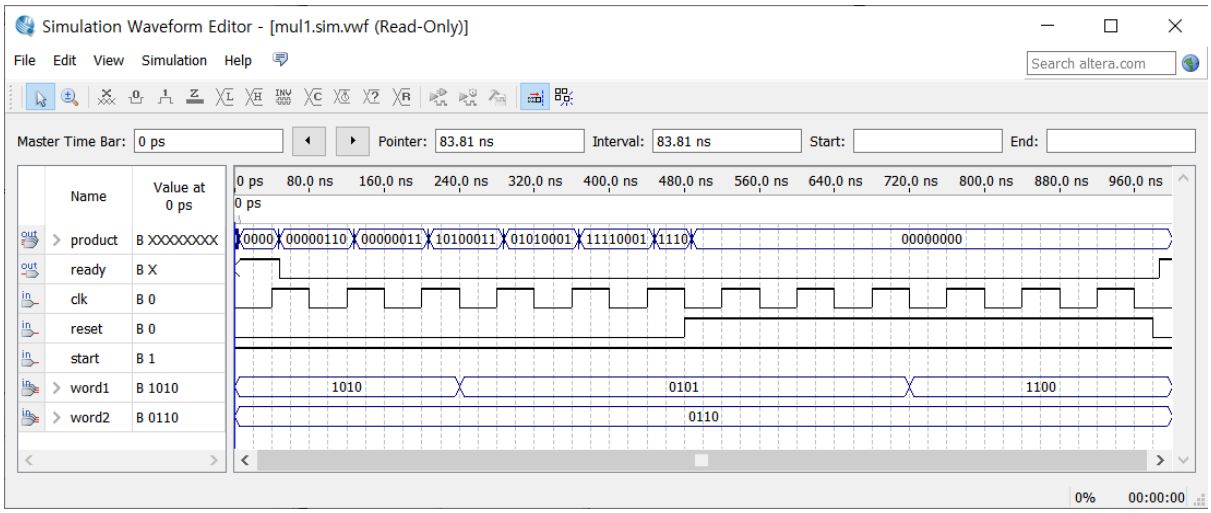
(1)

```
1   module mul1(clk,reset,start,word1,word2,product,ready);
2      input clk,reset,start;
3      input [3:0] word1,word2;
4      output [7:0] product;
5      output ready;
6
7      wire m0,load,shift,add;
8
9      datapath u1(clk,reset,load,shift,add,word1,word2,product,m0);
10     controller u2(clk,reset,start,m0,load,shift,add,ready);
11  endmodule
12
13  module datapath(clk,reset,load,shift,add,word1,word2,product,m0);
14     input clk,reset,load,shift,add;
15     input [3:0] word1,word2;
16     output [7:0] product;
17     output m0;
18
19     reg [7:0] product;
20     reg carry;
21     reg [3:0] multiplicand;
22     wire [4:0] sum;
23
24     assign m0 = product[0];
25     assign sum = product[7:4] + multiplicand;
26
27     always @(posedge clk or posedge reset) begin
28        if(reset) begin multiplicand <= 0; product <=0; end
29        else if(load) begin
30           multiplicand <= word1;
```

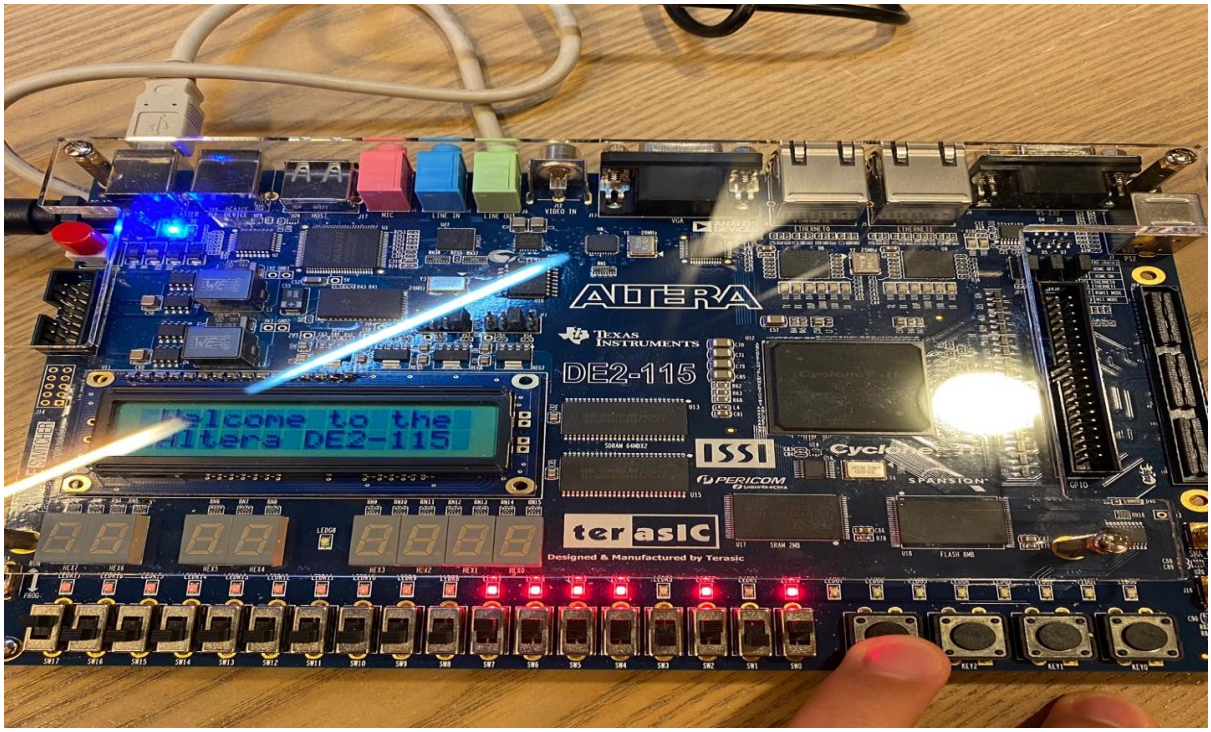
```
30          multiplicand <= word1;
31          product <= {4'b0,word2};
32       end
33       else if(shift) begin
34          product <= {carry,product[7:1]}; carry <= 0;
35       end
36       else if(add) {carry,product[7:4]} <= sum;
37       end
38 endmodule
39
40 module controller(clk,reset,start,m0,load,shift,add,ready);
41    input clk,reset,start,m0;
42    output load,shift,add,ready;
43
44    reg[3:0] state;
45    localparam s0=0,s1=8,s2=9,s3=10,s4=11,s5=12,s6=13,s7=14,s8=15;
46
47    always @(posedge clk or posedge reset) begin
48       if(reset) state <= s0;
49       else
50          case(state)
51             s0: if(start) begin state <= s1; end
52             s1: if(m0) state <= s2; else state <= s3;
53             s2: state <= s3;
54             s3: if(m0) state <= s4; else state <= s5;
55             s4: state <= s5;
56             s5: if(m0) state <= s6; else state <= s7;
57             s6: state <= s7;
58             s7: if(m0) state <= s8; else state <= s0;
59             s8: state <= s0;
```

```
59             s8: state <= s0;
60             endcase
61    end
62
63    assign load = (state==s0) && start;
64    assign shift = (state==s1||state==s3||state==s5||state==s7) && ~m0 ||
65                   (state==s2||state==s4||state==s6||state==s8);
66    assign add = (state==s1||state==s3||state==s5||state==s7) && m0;
67    assign ready = (state==s0) && ~reset;
68 endmodule
69
```

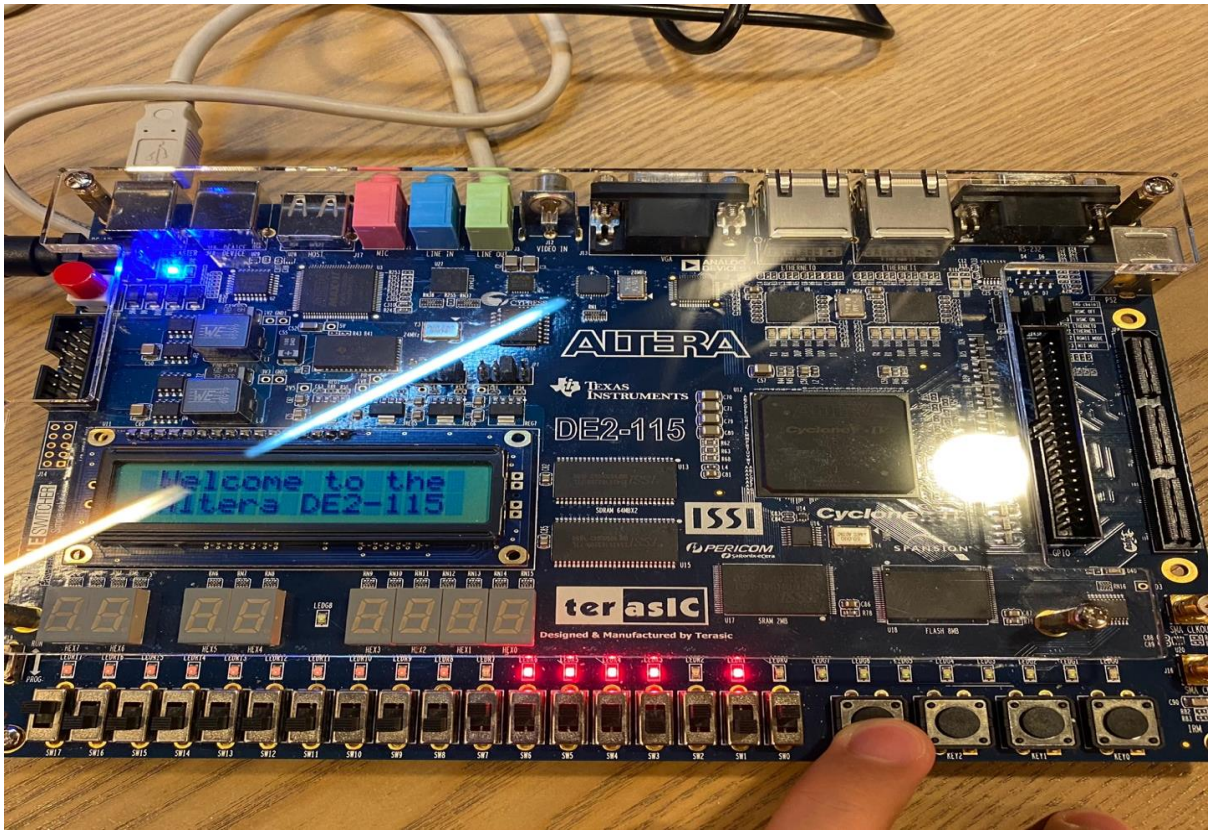
(2)



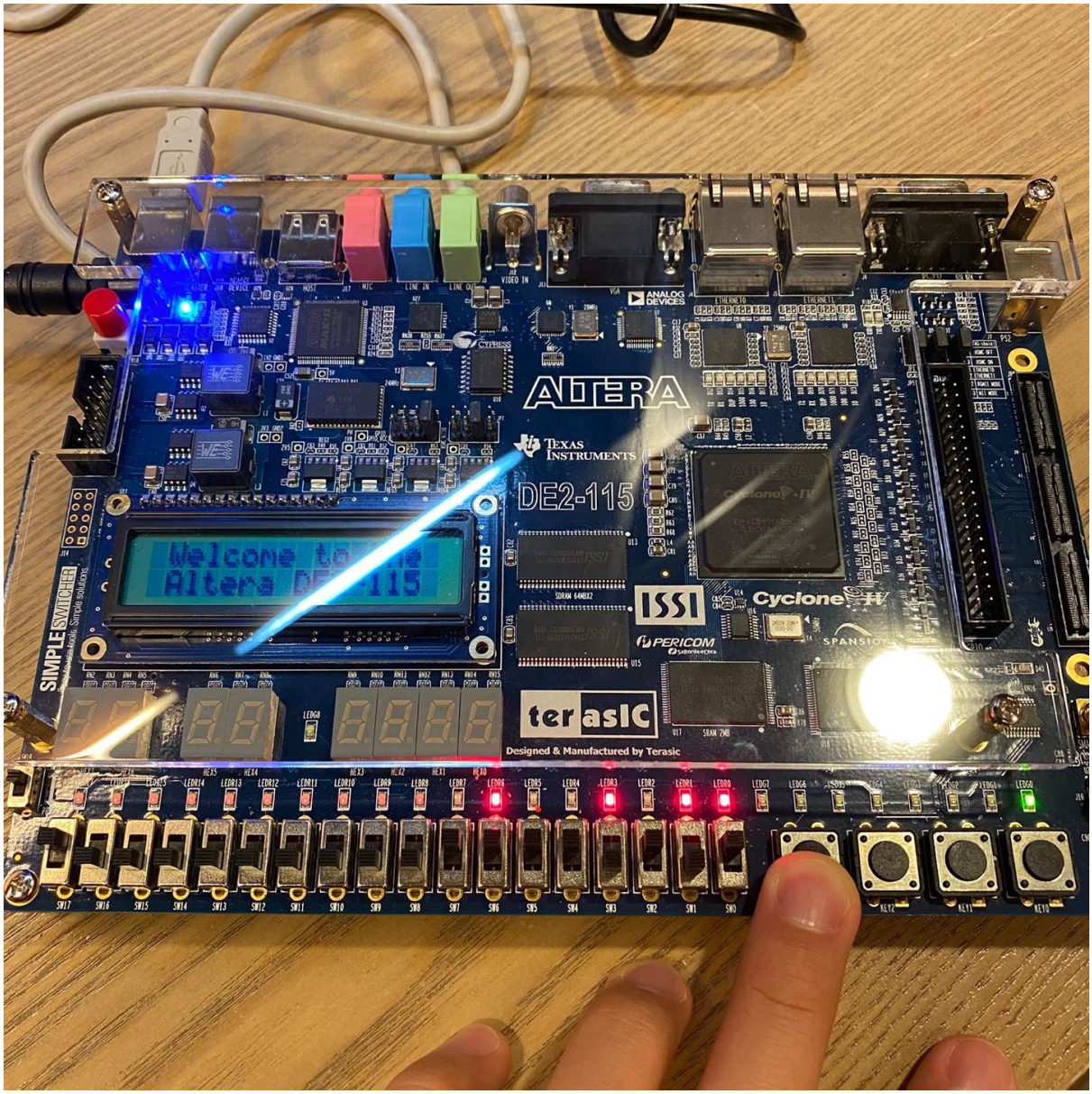
(3)



1111과 0101의 곱셉이다



clk 신호를 주니 shift 하는 것을 확인할 수 있다.



계산결과가 01001011로 올바르게 출력됐다.

2.

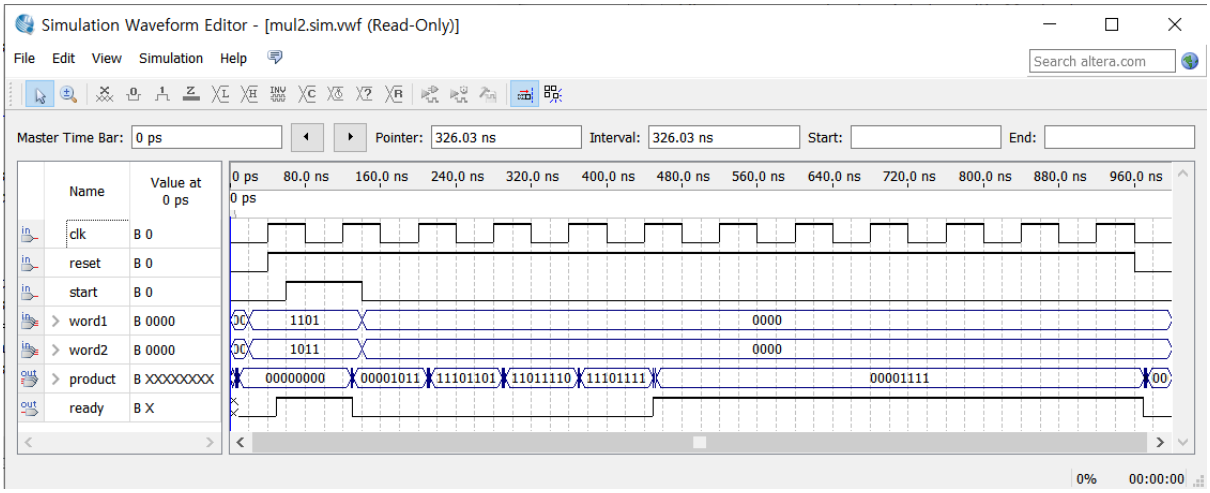
(1)

```
1  module mul2(clk,reset,start,word1,word2,product,ready);
2     input clk,reset,start;
3     input [3:0] word1,word2;
4     output [7:0] product;
5     output ready;
6
7     wire m0,load,shift,addshift,sub;
8
9     datapath u1(clk,~reset,load,shift,addshift,sub,word1,word2,product,m0);
10    controller u2(clk,~reset,start,m0,load,shift,addshift,sub,ready);
11 endmodule
12
13 module datapath(clk,reset,load,shift,addshift,sub,word1,word2,product,m0);
14    input clk,reset,load,shift,addshift,sub;
15    input [3:0] word1,word2;
16    output [7:0] product;
17    output m0;
18
19    reg [7:0] product;
20    reg [3:0] multiplicand;
21    wire [4:0] sum;
22    wire [4:0] eproduct,emcand;
23
24    assign m0 = product[0];
25    assign eproduct = {product[7],product[7:4]};
26    assign emcand = {multiplicand[3],multiplicand};
27    assign sum = sub ? (eproduct - emcand) : (eproduct + emcand);
28
29    always @(posedge clk or posedge reset) begin
30       if(reset) begin multiplicand <= 0; product <=0; end
```

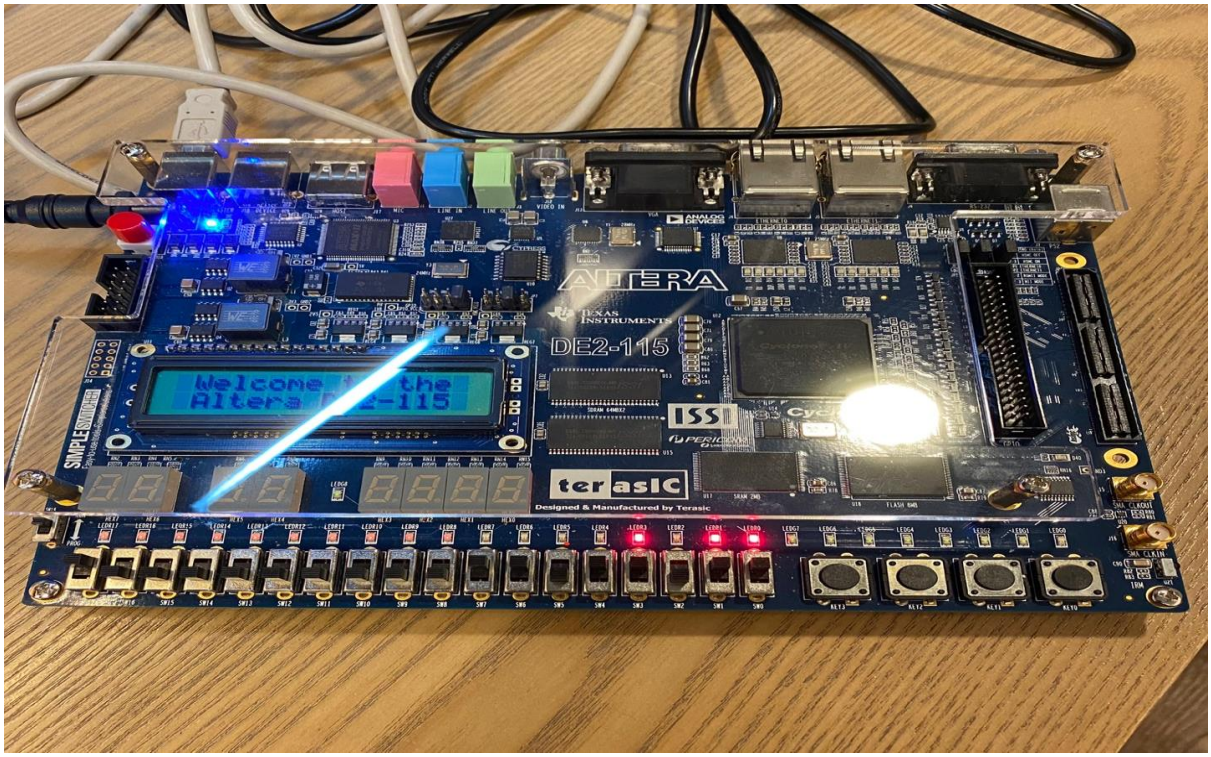
```
30          if(reset) begin multiplicand <= 0; product <=0; end
31          else if(load) begin
32             multiplicand <= word1;
33             product <= {4'b0,word2};
34          end
35          else if(shift)
36             product <= {product[7],product[7:1]};
37          else if(addshift) product <= {sum,product[3:1]};
38       end
39    endmodule
40
41    module controller(clk,reset,start,m0,load,shift,addshift,sub,ready);
42       input clk,reset,start,m0;
43       output load,shift,addshift,sub,ready;
44
45       reg state;
46       reg [1:0] count;
47       localparam s0=0,s1=1;
48
49       always @(posedge clk, posedge reset) begin
50          if(reset) begin state <= s0; count <= 0; end
51          else
52             case(state)
53                s0: if(start) begin state <= s1; count <= 3; end
54                s1: if(count ==0) state <= s0; else count <= count - 1;
55                endcase
56       end
57
58       assign load = (state==s0) && start;
59       assign shift = (state==s1) && ~m0;
```

```
59       assign shift = (state==s1) && ~m0;
60       assign addshift = (state==s1) && m0;
61       assign sub = (state==s1) && (count==0);
62       assign ready = (state==s0) && ~reset;
63    endmodule
64
```

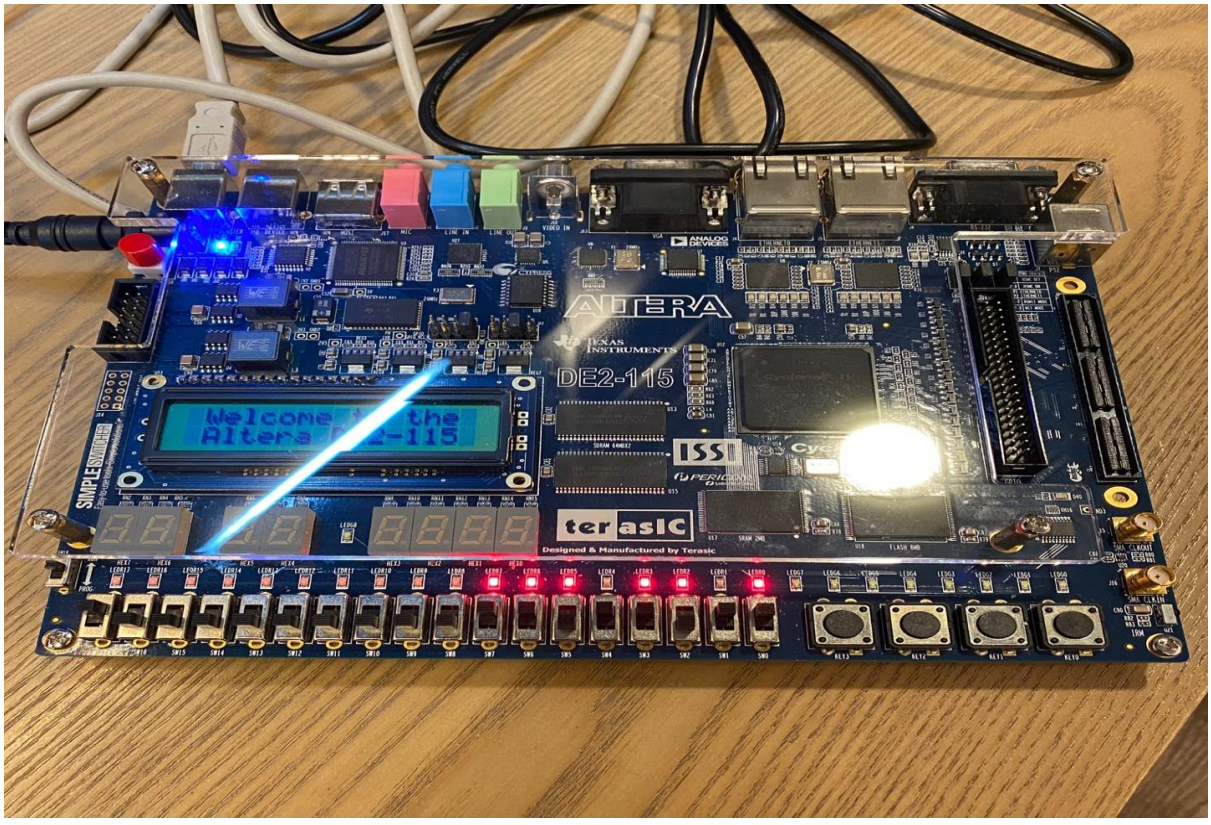
(2)



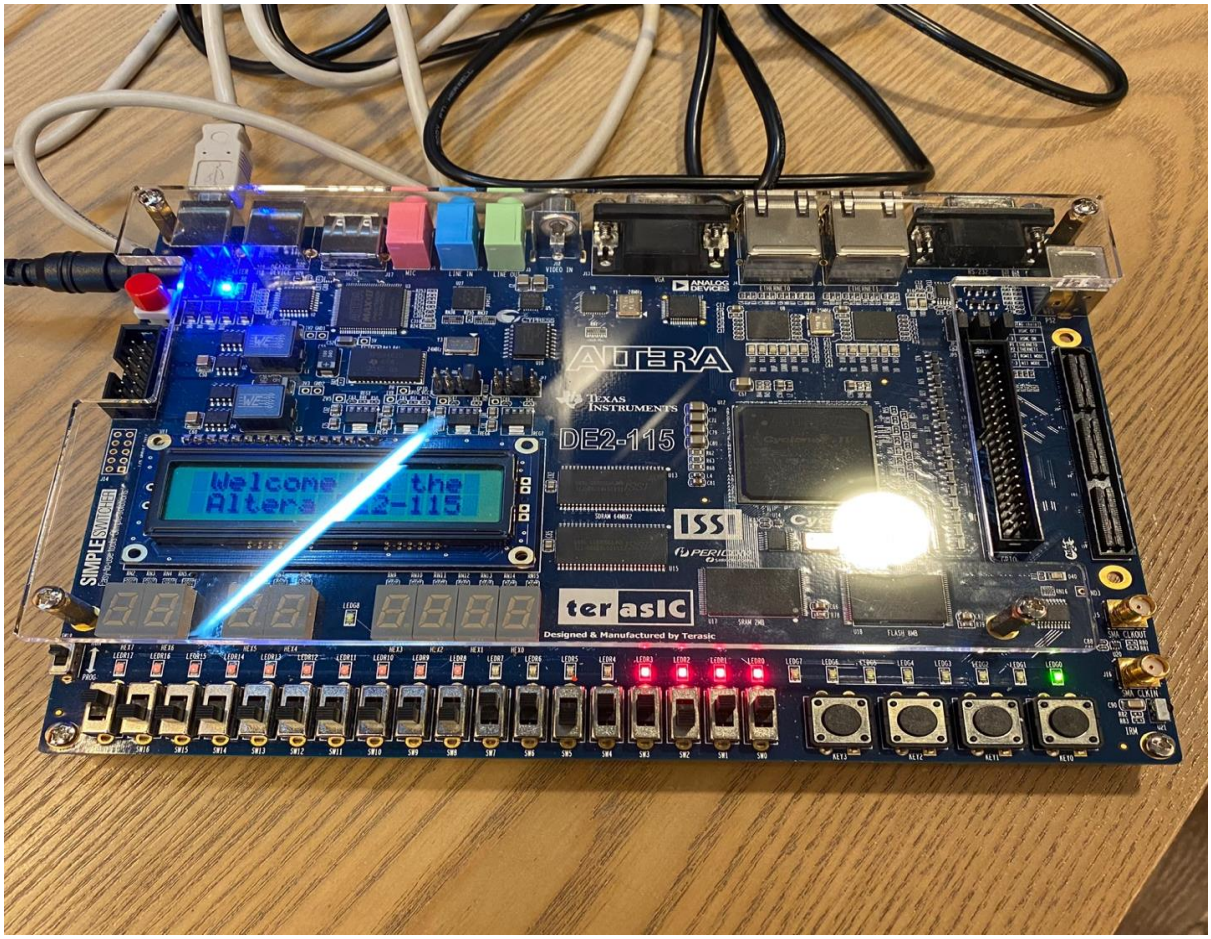
(3)



1101과 1011의 곱셈이다.



clk신호를 주니 addshift 하는 것을 확인할 수 있다.



계산결과가 00001111로 올바르게 출력됐다.