

## 컴퓨터공학실험 2 12 주차 결과보고서

전공: 컴퓨터공학과

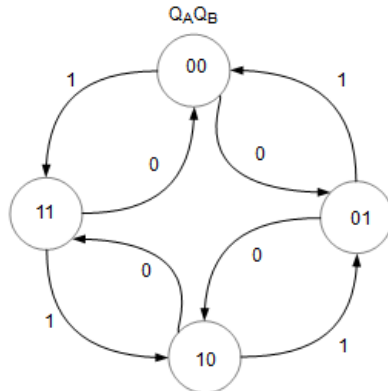
학년: 2

학번: 20191559

이름: 강상원

### 1. 2-bit counter의 결과 및 Simulation 과정에 대해서 설명하시오.

(verliog source, 출력 예시, 과정 상세히 적을것!)



2-Bit Counter 의 상태도는 위와 같고, 상태표는 아래와 같다.

Present state	Next state / Output	
	x = 0	x = 1
a=0, b=0	A=0, B=0 / 0	A=0, B=1 / 0
a=0, b=1	A=0, B=1 / 0	A=1, B=0 / 0
a=1, b=0	A=1, B=0 / 0	A=1, B=1 / 0
a=1, b=1	A=1, B=1 / 0	A=0, B=0 / 1

상태표를 참고하여 카르노 맵을 다음과 같이 그릴 수 있다.

ab \ x	0	1
00	0	0
01	0	1
11	1	0
10	1	1

$$A = ax' + ab' + a'bx$$

ab \ x	0	1
00	0	0
01	0	0
11	0	1
10	0	0

$$Z = abx$$

ab \ x	0	1
00	0	0
01	1	0
11	1	0
10	0	1

$$B = bx' + b'x$$

위와 같은 2-Bit Counter를 JK Flip-Flop을 통해 나타낼 수 있다.

Verilog Design Code는 아래와 같다.

```
1 `timescale 1ns / 1ps
2
3 module binary_counter(
4     inout clk,reset, j, k,
5     output wire [1:0] q
6 );
7
8     wire[1:0] qc;
9     wire [1:0] tmpq;
10    jk_ff b1(j, k, clk, tmpq[0], qc[0]);
11    jk_ff b2(tmpq[0], tmpq[0], clk, tmpq[1], qc[1]);
12
13    and(q[0], ~reset, tmpq[0]);
14    and(q[1], ~reset, tmpq[1]);
15 endmodule
16
17 module jk_ff(
18     input j, k, clk,
19     output reg q, qc
20 );
21     initial begin
22         q = 0;
23         qc = 1;
24     end
25     always@(negedge clk) begin
26         if(j==0 && k==0) begin
27             q <= q;
28             qc <= qc;
29         if(j==0 && k==1) begin
30             q = 0;
31             qc = 1;
32         end
33         end
34         if(j==1 && k==0) begin
35             q = 1;
36             qc = 0;
37         end
38         if(j==1 && k==1) begin
39             q <= qc;
40             qc = q;
41         end
42     end
43 endmodule
```

다음은 2-Bit Counter의 시뮬레이션 코드, Constraints File이다.

```

1 `timescale 1ns / 1ps
2
3 module binary_counter_tb;
4 reg clk;
5 reg reset, j, k;
6 wire[1:0] q;
7 binary_counter func(
8     .clk(clk),
9     .j(j),
10    .k(k),
11    .reset(reset),
12    .q(q)
13 );
14
15     initial begin
16         clk = 1'b1;
17         reset = 1'b0;
18         j = 1'b1;
19         k = 1'b1;
20     end
21     always begin
22         clk = #1~clk;
23         clk = #5~clk;
24     end
25     initial begin
26         #48
27         reset = ~reset;
28         #10
29         $finish;
30     end
31 endmodule
32 |

```

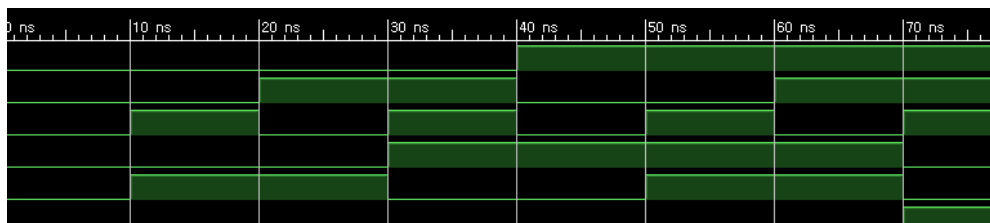
```

1 |set_property IOSTANDARD LVCMOS18 [get_ports clk]
2 set_property IOSTANDARD LVCMOS18 [get_ports reset]
3 set_property IOSTANDARD LVCMOS18 [get_ports j]
4 set_property IOSTANDARD LVCMOS18 [get_ports k]
5 set_property IOSTANDARD LVCMOS18 [get_ports q[1]]
6 set_property IOSTANDARD LVCMOS18 [get_ports q[0]]
7 set_property PACKAGE_PIN J4 [get_ports clk]
8 set_property PACKAGE_PIN L3 [get_ports reset]
9 set_property PACKAGE_PIN K3 [get_ports j]
10 set_property PACKAGE_PIN M2 [get_ports k]
11 set_property PACKAGE_PIN F15 [get_ports q[1]]
12 set_property PACKAGE_PIN F13 [get_ports q[0]]
13
14 set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets {clk_IBUF}]

```

FPGA의 스위치와 LED로 실제 동작을 확인하였다.

위와 같이 작성한 후의 시뮬레이션 결과는 아래와 같다.

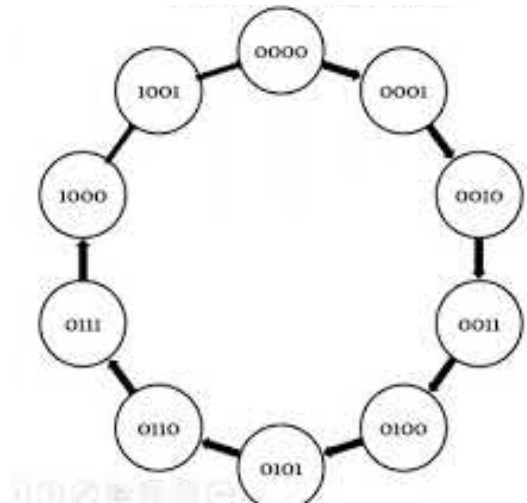


시뮬레이션에서 보듯이 x가 4번 1이 나온 후 z (맨 밑 값)이 1이 된다.

## 2. 4-bit decade counter의 결과 및 Simulation 과정에 대해서 설명하시오.

(verilog source, 출력 예시, 과정 상세히 적을것!)

다음은 4-Bit decade Counter의 상태도이다.



상태표는 아래와 같다.

n of $q_n$	Present state	Next state / Output	
		x=0	x=1
0	0000	q0 / 0	q1 / 0
1	0001	q1 / 0	q2 / 0
2	0010	q2 / 0	q3 / 0
3	0011	q3 / 0	q4 / 0
4	0100	q4 / 0	q5 / 0
5	0101	q5 / 0	q6 / 0
6	0110	q6 / 0	q7 / 0
7	0111	q7 / 0	q8 / 0
8	1000	q8 / 0	q9 / 0
9	1001	q9 / 0	q0 / 1

상태표를 참고하여 카르노 맵을 다음과 같이 그릴 수 있다.

- A

x = 0

ab \ cd	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	d	d	d	d
10	1	1	d	d

$x = 1$

ab \ cd	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	d	d	d	d
10	1	0	d	D

$$A = ax' + bcdx + ad'x$$

- B

$x = 0$

ab \ cd	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	d	d	d	d
10	0	0	d	d

$x = 1$

ab \ cd	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	d	d	d	d
10	0	0	d	d

$$B = bx' + bd'x + b'cdx + bc'x$$

- C

$x = 0$

ab \ cd	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	d	d	d	d
10	0	0	d	d

$x = 1$

ab \ cd	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	d	d	d	d

10	0	0	d	d
----	---	---	---	---

$$C = cd'x + c'x + a'c'dx$$

- D

$$x = 0$$

ab \ cd	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	d	d	d	d
10	0	1	d	d

$$x = 1$$

ab \ cd	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	d	d	d	d
10	1	0	d	d

$$D = d'x + dx'$$

- Z

$$x = 0$$

ab \ cd	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	d	d	d	d
10	0	0	d	d

$$x = 1$$

ab \ cd	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	d	d	d	d
10	0	1	d	d

$$Z = adx$$

위와 같은 4-Bit Decade Counter를 JK Flip-Flop을 통해 나타낼 수 있다.

Verilog Design Code는 아래와 같다.

```

1  `timescale 1ns / 1ps
2
3  module four_bit_decade_counter(
4      input clk, reset,
5      output wire [3:0] q
6  );
7
8      wire[3:0] qc;
9      wire[3:0] tmpq;
10     wire[6:0] tmpwire;
11     jk_ff b1(1'b1,1'b1,clk,tmpq[0], qc[0]);
12     and(tmpwire[0], tmpq[0], qc[3]);
13     jk_ff b2(tmpwire[0], tmpwire[0], clk, tmpq[1], qc[1]);
14     and(tmpwire[1], tmpq[0], tmpq[1]);
15     jk_ff b3(tmpwire[1], tmpwire[1], clk, tmpq[2], qc[2]);
16     and(tmpwire[2], tmpwire[1], tmpq[2]);
17     and(tmpwire[3], tmpq[0], tmpq[3]);
18     or(tmpwire[4], tmpwire[2], tmpwire[3]);
19     jk_ff b4(tmpwire[4], tmpwire[4], clk, tmpq[3], qc[3]);
20
21     and(q[0], ~reset, tmpq[0]);
22     and(q[1], ~reset, tmpq[1]);
23     and(q[2], ~reset, tmpq[2]);
24     and(q[3], ~reset, tmpq[3]);
25 endmodule
26
27 module jk_ff(
28     input j, k, clk,
29     output reg q, qc
30 );
31     initial begin
32         q=0;
33         qc=1;
34     end
35     always@(negedge clk)begin
36         if(j==0&&k==0) begin
37             q<=q;
38             qc<=qc;
39         end
40         if(j==0&&k==1) begin
41             q=0;
42             qc=1;
43         end
44         if(j==1&&k==0) begin
45             q=1;
46             qc=0;
47         end
48         if(j==1&&k==1)begin
49             q<=qc;
50             qc<=q;
51         end
52     end
53 endmodule

```

다음은 2-Bit Counter의 Constraints File이다.

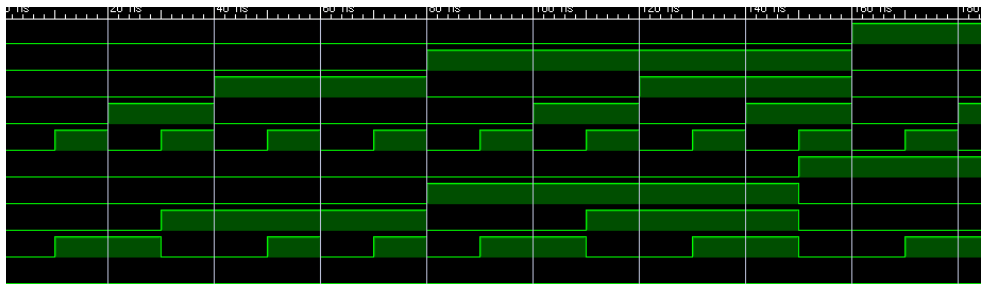
```

1 set_property IOSTANDARD LVCMOS18 [get_ports clk]
2 set_property IOSTANDARD LVCMOS18 [get_ports reset]
3 set_property IOSTANDARD LVCMOS18 [get_ports j]
4 set_property IOSTANDARD LVCMOS18 [get_ports k]
5 set_property IOSTANDARD LVCMOS18 [get_ports q[1]]
6 set_property IOSTANDARD LVCMOS18 [get_ports q[0]]
7 set_property PACKAGE_PIN J4 [get_ports clk]
8 set_property PACKAGE_PIN L3 [get_ports reset]
9 set_property PACKAGE_PIN K3 [get_ports j]
10 set_property PACKAGE_PIN M2 [get_ports k]
11 set_property PACKAGE_PIN F15 [get_ports q[1]]
12 set_property PACKAGE_PIN F13 [get_ports q[0]]
13
14 set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets {clk_IBUF}]

```

FPGA의 스위치와 LED로 실제 동작을 확인하였다.

위와 같이 작성한 후의 시뮬레이션 결과는 아래와 같다.

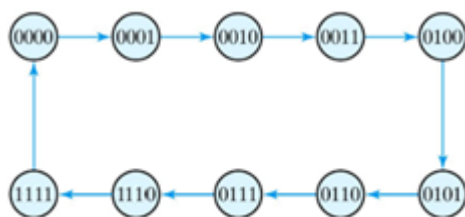


상태도와 같이 0000에서 입력 0일 때 0000이 나오고 출력이 0임을 확인할 수 있다.

### 3. 4-bit 2421 decade counter의 결과 및 Simulation 과정에 대해서 설명하시오.

(verliog source, 출력 예시, 과정 상세히 적을것!)

다음은 4-Bit 2421 decade Counter의 상태도이다. 기존 Counter (8421)과 다르게 왼쪽에서 첫번째 비트를 2로 두어 계산한다. 따라서 10진수 9는 1111로 표현된다.



상태표는 아래와 같다.

n of $q_n$	Present state	Next state / Output	
		x=0	x=1
0	0000	q0 / 0	q1 / 0
1	0001	q1 / 0	q2 / 0
2	0010	q2 / 0	q3 / 0
3	0011	q3 / 0	q4 / 0



4	0100	q4 / 0	q5 / 0
5	0101	q5 / 0	q6 / 0
6	0110	q6 / 0	q7 / 0
7	0111	q7 / 0	q8 / 0
8	1110	q8 / 0	q9 / 0
9	1111	q9 / 0	q0 / 1

상태표를 참고하여 카르노 맵을 다음과 같이 그릴 수 있다.

- A

x = 0

ab \ cd	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	d	d	1	1
10	d	d	d	d

x = 1

ab \ cd	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	D	d	0	1
10	d	d	d	d

$$A = ax' + a'bcdx + abcd'x$$

- B

x = 0

ab \ cd	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	d	d	1	1
10	d	d	d	D

x = 1

ab \ cd	00	01	11	10
00	0	0	1	0
01	1	1	1	1

11	d	d	0	1
10	d	d	d	d

$$B = bx' + a'bx + bcd'x + a'cdx$$

- C

$$x = 0$$

ab \ cd	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	d	d	1	1
10	d	d	d	d

$$x = 1$$

ab \ cd	00	01	11	10
00	0	1	0	1
01	0	1	1	1
11	d	d	0	1
10	d	d	d	d

$$C = cd'x + cx' + a'bcx + cd'x$$

- D

$$x = 0$$

ab \ cd	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	d	d	1	0
10	d	d	d	d

$$x = 1$$

ab \ cd	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	d	d	0	1
10	d	d	d	d

$$D = d'x + dx'$$

- Z

x = 0

ab \ cd	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	d	d	0	0
10	d	d	d	d

x = 1

ab \ cd	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	d	d	1	0
10	d	d	d	d

Z = adx

위와 같은 4-Bit 2421 Decade Counter를 JK Flip-Flop을 통해 나타낼 수 있다.

Verilog Design Code는 아래와 같다.

```

1 `timescale 1ns / 1ps
2
3 module tfto_decade_counter(
4     input clk, reset,
5     output wire [3:0] q
6 );
7     wire [3:0] tmpq;
8     wire check;
9     wire tmpwire;
10    reg [3:0] B;
11    four_bit_decade_counter fourbit(clk, reset, tmpq);
12    assign check = (tmpq[3]&~tmpq[2])|(tmpq[2]&tmpq[0])|(tmpq[2]&tmpq[1]&~tmpq[0]);
13    always @(check)
14        if(check == 1'b1) begin
15            B = 4'b0110;
16        end
17        else begin
18            B = 4'b0000;
19        end
20    adder4bit add0(tmpq, B, 0, q, tmpwire);
21 endmodule
22
23 module four_bit_decade_counter(
24     input clk, reset,
25     output wire [3:0] q
26 );
27
28     wire[3:0] qc;
29     wire[3:0] tmpq;
30     wire[6:0] tmpwire;
31     jk_ff b1(1'b1,1'b1,clk,tmpq[0], qc[0]);
32     and(tmpwire[0], tmpq[0], qc[3]);
33     jk_ff b2(tmpwire[0], tmpwire[0], clk, tmpq[1], qc[1]);
34     and(tmpwire[1], tmpq[0], tmpq[1]);
35     jk_ff b3(tmpwire[1], tmpwire[1], clk, tmpq[2], qc[2]);
36     and(tmpwire[2], tmpwire[1], tmpq[2]);
37     and(tmpwire[3], tmpq[0], tmpq[3]);
38     or(tmpwire[4], tmpwire[2], tmpwire[3]);
39     jk_ff b4(tmpwire[4], tmpwire[4], clk, tmpq[3], qc[3]);
40
41     and(q[0], ~reset, tmpq[0]);
42     and(q[1], ~reset, tmpq[1]);
43     and(q[2], ~reset, tmpq[2]);
44     and(q[3], ~reset, tmpq[3]);
45 endmodule

```

```

47 module jk_ff(
48     input j, k, clk,
49     output reg q, qc
50 );
51     initial begin
52         q=0;
53         qc=1;
54     end
55     always@(negedge clk)begin
56         if(j==0&&k==0) begin
57             q<=q;
58             qc<=qc;
59         end
60         if(j==0&&k==1) begin
61             q=0;
62             qc=1;
63         end
64         if(j==1&&k==0) begin
65             q=1;
66             qc=0;
67         end
68         if(j==1&&k==1)begin
69             q<=qc;
70             qc<=q;
71         end
72     end
73 endmodule
74
75 module adder1bit(A, B, Ci, S, Co);
76
77     input A, B, Ci;
78     output S, Co;
79     assign S=A^B^Ci;
80     assign Co=(A&B)|((A^B)&Ci);
81
82 endmodule
83
84 module adder4bit(A, B, Ci, S, Co);
85     input [3:0] A, B; input Ci;
86     output [3:0] S; output Co;
87
88     wire [3:0] A, B, S; wire Ci, Co;
89     wire [2:0] C;
90
91     adder1bit add1(A[0], B[0], Ci, S[0], C[0]);
92     adder1bit add2(A[1], B[1], C[0], S[1], C[1]);
93     adder1bit add3(A[2], B[2], C[1], S[2], C[2]);
94     adder1bit add4(A[3], B[3], C[2], S[3], Co);
95
96 endmodule

```

다음은 2-Bit Counter의 Constraints File이다.

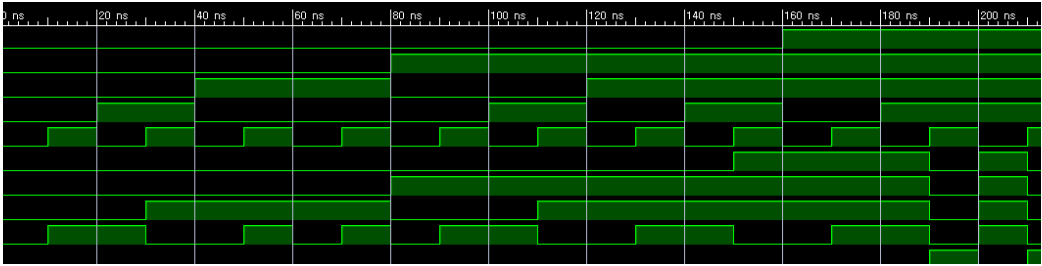
```

1 |set_property IOSTANDARD LVCMOS18 [get_ports clk]
2 set_property IOSTANDARD LVCMOS18 [get_ports reset]
3
4 set_property IOSTANDARD LVCMOS18 [get_ports q[3]]
5 set_property IOSTANDARD LVCMOS18 [get_ports q[2]]
6 set_property IOSTANDARD LVCMOS18 [get_ports q[1]]
7 set_property IOSTANDARD LVCMOS18 [get_ports q[0]]
8
9
10 set_property PACKAGE_PIN J4 [get_ports clk]
11 set_property PACKAGE_PIN L3 [get_ports reset]
12
13 set_property PACKAGE_PIN F15 [get_ports q[3]]
14 set_property PACKAGE_PIN F13 [get_ports q[2]]
15 set_property PACKAGE_PIN F14 [get_ports q[1]]
16 set_property PACKAGE_PIN F16 [get_ports q[0]]
17
18 set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets {clk_IBUF}]

```

FPGA의 스위치와 LED로 실제 동작을 확인하였다.

위와 같이 작성한 후의 시뮬레이션 결과는 아래와 같다.



상태도와 같이 1110에서 입력 0일 때 1110이 나오고 출력이 0임을 확인할 수 있다.