

CSED273 Lab6 Report

20210643 김현준

1. 개요

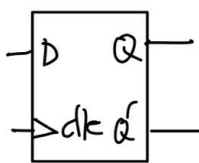
순차회로의 대표적인 예시 중 하나라고 볼 수 있는 Counter(계수기)의 특성을 알아보고, 다양한 종류의 계수기를 Verilog로 구현해본다. 이를 통해, 순차 회로와 counter에 대한 개념을 이해하고, 순차회로의 회로 구현에 대해 배운다.

2. 이론적 배경

1) flip flop

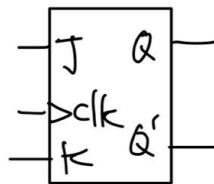
플립플롭은 1비트의 정보를 보관, 유지할 수 있는 회로로, 순차 회로의 기본 요소가 된다. 이번 lab에서는 D플립플롭과 JK플립플롭이 이용되는데, D플립플롭은 clock 신호에 따라 입력 D가 Q에 바로 반영되는 flip flop이다. D플립플롭의 excitation table은 아래와 같다. 또한 JK 플립플롭은 set, reset, hold, toggle의 기능을 가진 플립플롭으로 아래와 같이 나타난다.

D FlipFlop



D	Q(t+1)
0	0
1	1

JK FlipFlop



J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Q'(t)

2) 계수기(Counter)

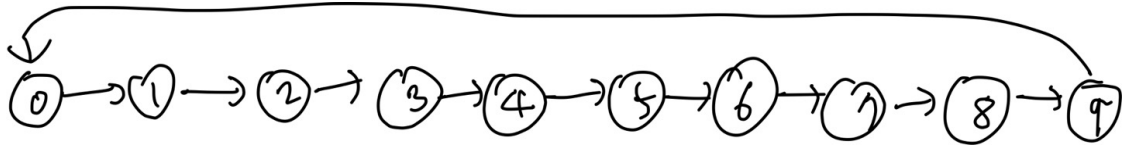
계수기는 순차회로의 일종으로, clock에 따라 순차적으로 지정된 패턴에 따라 숫자를 저장 및 출력하는 회로이다. 사용 목적에 따라 출력 패턴을 다양하게 설정하여, 다양한 종류의 계수기를 만들 수가 있다.

2-1) 동기 계수기(Synchronous Counter)

동기 계수기는 combinational circuit을 통해 clock 신호를 카운터의 각 플립플롭에 동시에 입력하는 계수기이다. 비회로가 비동기 계수기에 비해 다소 복잡하지만, clock이 지연되지 않아 작동 속도는 더욱 빠른 장점이 있다.

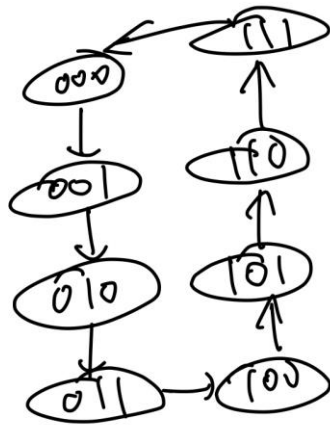
2-2) 십진 계수기(Decade Counter)

십진 계수기는 십진수를 순차적으로 counting하는 회로이다. 0부터 9까지 반복하여 순차적으로 카운팅한다. State transition diagram은 아래와 같다.



3) 상태 전이도 및 전이표

상태 전이도는 상태 변화를 그림으로 나타낸 것이고, 전이표는 상태 변화를 표로 나타낸 것이다. 예를 들어 3bit 이진 계수기는 아래와 같이 전이도와 전이표를 구성할 수 있다.

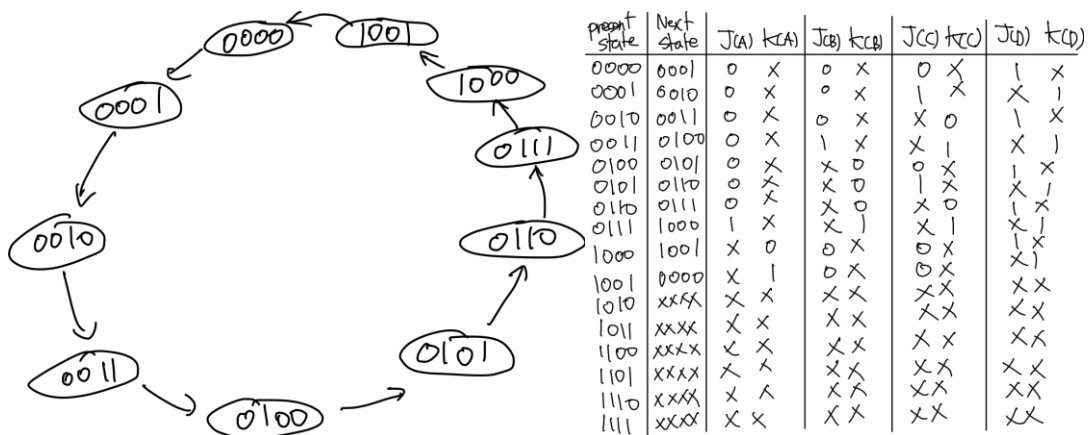


present state			next state		
A	B	C	A ⁺	B ⁺	C ⁺
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

3. 실험 준비

1) JK 플립플롭을 이용한 Synchronous decade BCD counter

우선 BCD counter는 BCD 0000부터 1001까지 decade 수를 1씩 count하는 계수기이다. 이를 상태 전이도로 표현하면 아래와 같고, 전이표는 아래와 같이 그릴 수 있다.



이를 바탕으로 A, B, C, D 각각 JK flip flop 회로를 구성하여 연결할 수 있다. 각 J, K에 대해 카노 맵으로 minimize한 식을 얻으면 아래와 같다.

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	X	X	X	X

$J_A = BCD$

AB \ CD	00	01	11	10
00	X	X	X	X
01	X	X	X	X
11	X	X	X	X
10	0	1	X	X

$K_A = D$

AB \ CD	00	01	11	10
00	0	0	1	0
01	X	X	X	X
11	X	X	X	X
10	0	0	X	X

$J_B = CD$

AB \ CD	00	01	11	10
00	X	X	X	X
01	0	0	1	0
11	X	X	X	X
10	X	X	X	X

$K_B = CD$

AB \ CD	00	01	11	10
00	0	1	X	X
01	1	X	X	X
11	X	X	X	X
10	0	0	X	X

$J_C = A'D$

AB \ CD	00	01	11	10
00	X	X	1	0
01	X	X	X	X
11	X	X	X	X
10	X	X	X	X

$K_C = D$

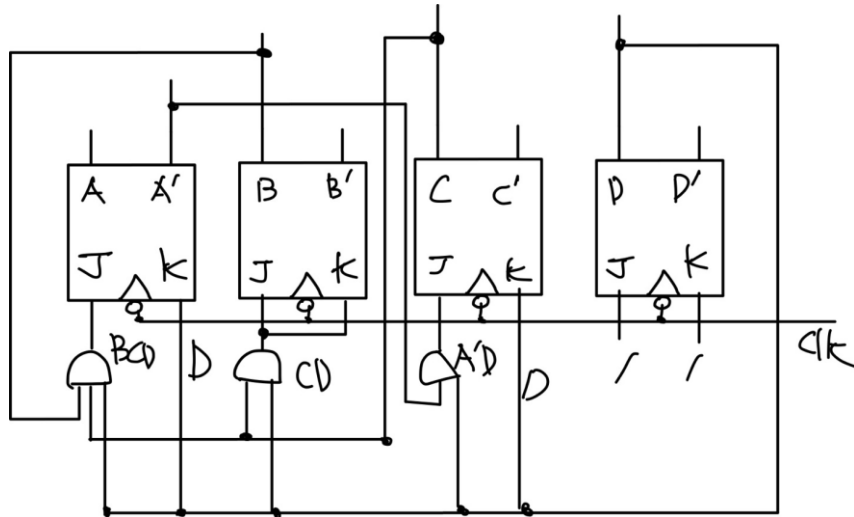
AB \ CD	00	01	11	10
00	1	X	X	X
01	1	X	X	X
11	1	X	X	X
10	1	X	X	X

$J_D = 1$

AB \ CD	00	01	11	10
00	X	1	1	X
01	X	1	1	X
11	X	X	X	X
10	X	1	X	X

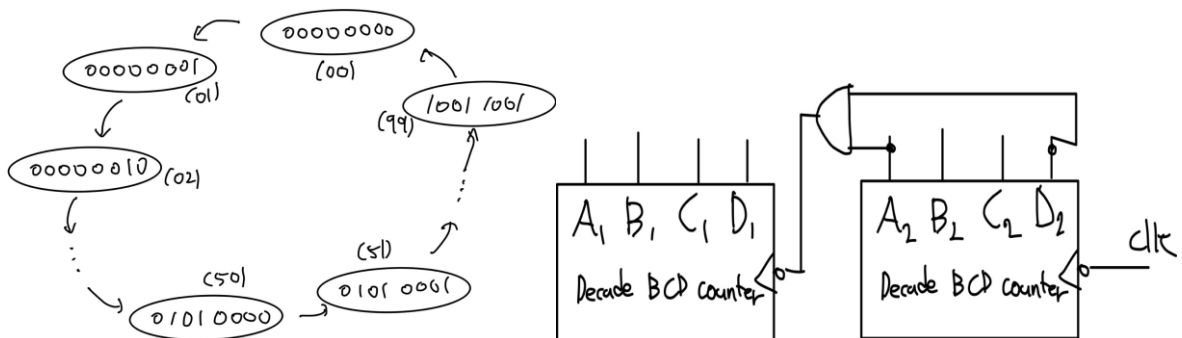
$K_D = 1$

이렇게 $J_A=BCD$, $K_A=D$, $J_B=CD$, $K_B=CD$, $J_C=A'D$, $K_C=D$, $J_D=1$, $K_D=1$ 의 값을 얻을 수 있었다. 이를 바탕으로 회로를 구성하면 아래와 같이 나뉜다.



2) JK 플립플롭을 이용한 두 자릿수 Decade BCD counter (0~99)

우선, 앞선 counter와 비슷한데, 0부터 99까지, 즉 두 decade 2자릿수의 수를 모두 counting하는 계수기이다. 따라서 각 자릿수를 하나의 Decade BCD counter가 담당하면 된다. 우선 전이도를 대략 그리면 아래와 같이 나타나고, 실험 준비 1)에서 구상한 Decade BCD Counter를 이용해 회로를 그리면 아래와 같이 구성이 가능하다.



이때, 2번째 Decade BCD counter의 clock에 A2D2를 연결하는 것은, BCD에서 9가 1001이기 때문에, 1001마다 다음 counter가 1씩 count되도록 하는 것이다. 이로써, 2자리 수의 십진수 계수기를 한 자리수의 counter로 구성할 수 있게 되었다.

3) D플립플롭을 이용한 3, 6, 9 counter

3, 6, 9 계수기는 0, 3, 6, 9, 13, 6 순으로 count하는 계수기이다. 따라서 이 계수기의 전이도를 그리면 아래와 같고, 전이표를 그리고 D flip flop으로 나타내면 아래와 같이 나타난다.

present state	Next state	D _A	D _B	D _C	D _D
0000	0011	0	0	1	1
0001	XXXX	X	X	X	X
0010	XXXX	X	X	X	X
0011	0110	0	1	1	0
0100	XXXX	X	X	X	X
0101	XXXX	X	X	X	X
0110	1001	1	0	0	1
0111	XXXX	X	X	X	X
1000	XXXX	X	X	X	X
1001	1101	1	1	0	1
1010	XXXX	X	X	X	X
1011	XXXX	X	X	X	X
1100	XXXX	X	X	X	X
1101	0110	0	1	1	0
1110	XXXX	X	X	X	X
1111	XXXX	X	X	X	X

이 전이표를 바탕으로 각 D플립플롭의 minimum 식을 카노 맵으로 구하면 아래와 같이 나타난다. 이를 이용해서 회로를 구성할 수 있다.

AB \ CD	00	01	11	10
00	0	X	0	X
01	X	X	X	1
11	X	0	X	X
10	X	1	X	X

$D_A = AB' + CD'$

AB \ CD	00	01	11	10
00	0	X	X	X
01	X	X	X	0
11	X	1	X	X
10	X	1	X	X

$D_B = D$

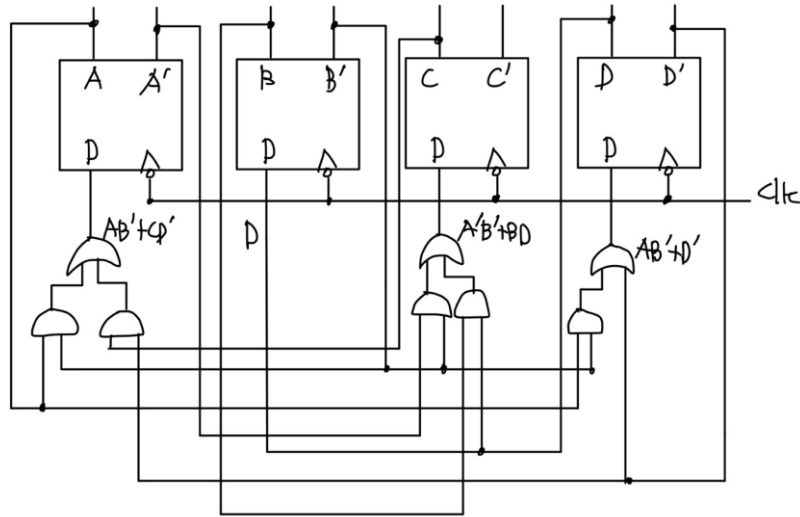
AB \ CD	00	01	11	10
00	1	X	1	X
01	X	X	X	0
11	X	1	X	X
10	X	0	X	X

$D_C = A'B' + BD$

AB \ CD	00	01	11	10
00	1	X	0	X
01	X	X	X	1
11	X	0	X	X
10	X	1	X	X

$D_D = AB' + D'$

이렇게 $D_A = AB' + CD'$, $D_B = D$, $D_C = A'B' + BD$, $D_D = AB' + D'$ 으로 D플립플롭을 구현할 수 있다. 이를 바탕으로 D플립플롭 4개를 이용하여 counter 회로를 구현할 수 있다. 아래와 같이 회로 구성을 구상하였다.

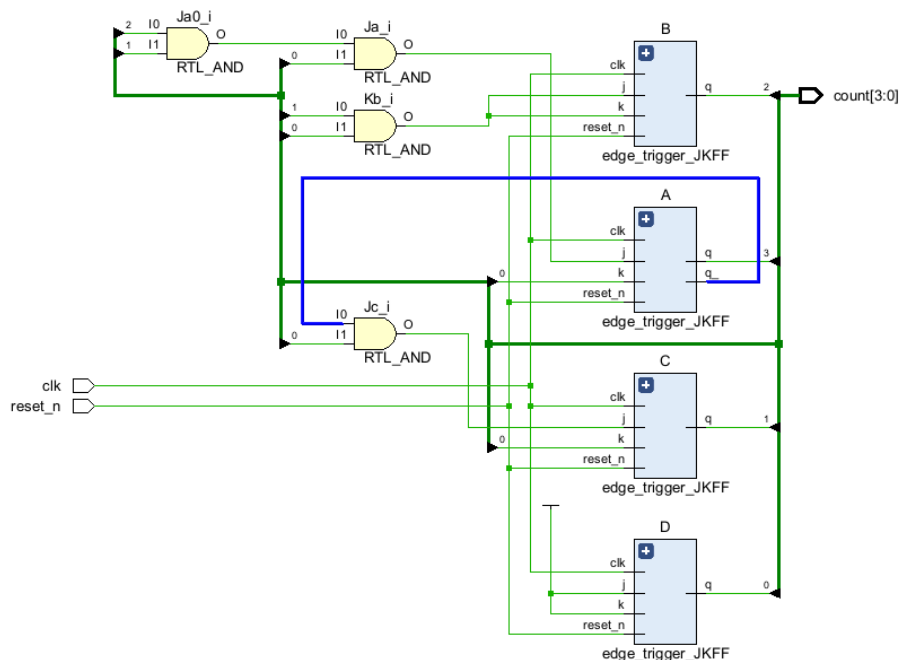


회로에 따라 0부터 99까지 clock에 따라 count up 되는 계수기가 동작할 수 있을 것이다. 이제, 실험 준비 단계가 끝났으니 vivado 환경에서 실험을 직접 진행한다.

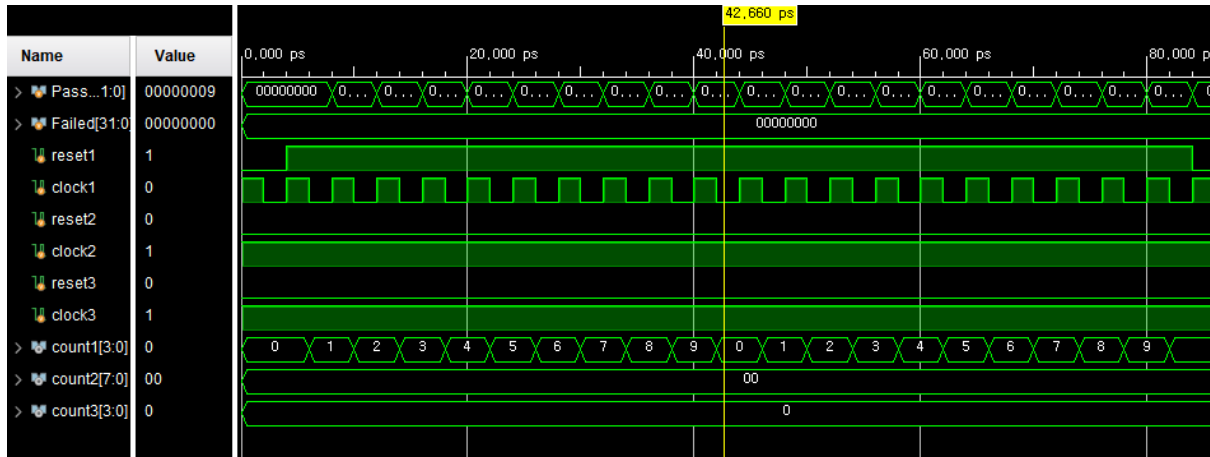
4. 결과

1) JK 플립플롭을 이용한 Synchronous decade BCD counter

우선, 실험 준비 과정에서 구상한 회로도대로, lab6_ff.v에 제공된 negative edge triggered JK flipflop을 사용해 Synchronous decade BCD counter 회로를 완성하였다. 그 결과 schematic을 확인하니 아래와 같이 나오는 것을 볼 수 있었다.



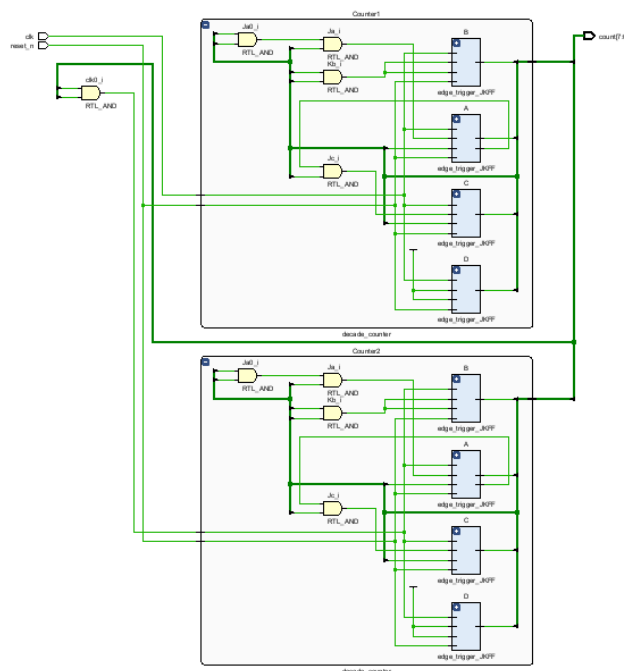
이는 실험 준비 단계에서 구성한 것과 같은 형태의 회로도였고, 즉 verilog로 회로를 적절하게 구성하였음을 알 수 있었다. 다음으로, 이 BCD counter의 동작을 확인하기 위해 주어진 testbench를 완성하고 시뮬레이션을 진행하니 아래와 같은 결과를 얻을 수 있었다.



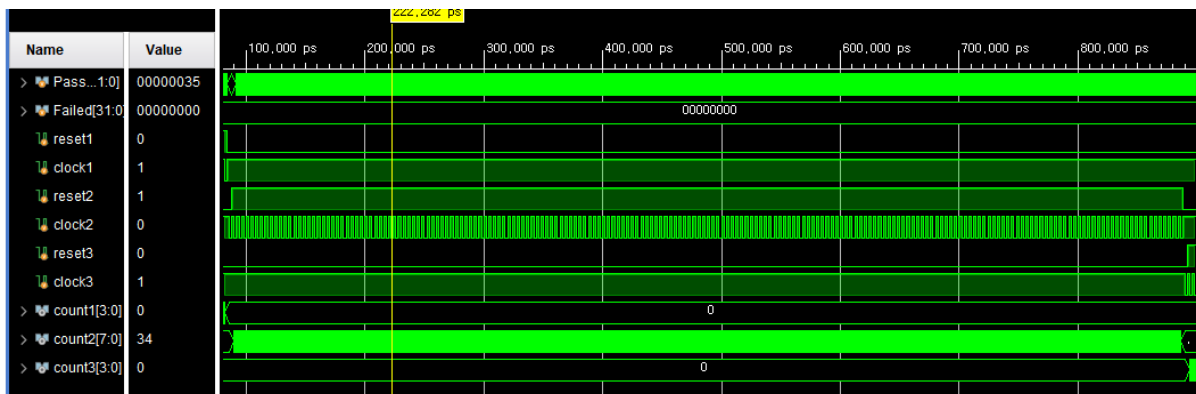
testbench는 우선 reset1이 0에서 시작하고, clock은 4ps마다 진동하도록(2ps마다 바뀌도록) 설계하여, 총 20번 진동하도록 하였다. 시뮬레이션 결과에서 볼 수 있듯, count1, 즉 decade BCD counter의 결과 값이 reset이 1일 때 0부터 9까지 그리고 9에서 다시 0으로 제대로 순환하는 것을 볼 수 있다. 이로써 회로가 제대로 verilog로 구현되었다는 것을 확인할 수 있다. 또한, testbench에서 count_expected값을 구해서 이와 실제 count를 비교하여 맞으면 가산하는 Passed도 제대로 증가하고, Failed는 0인 것을 보아 제대로 작동함을 알 수 있다.

2) JK 플립플롭을 이용한 두 자릿수 Decade BCD counter (0~99)

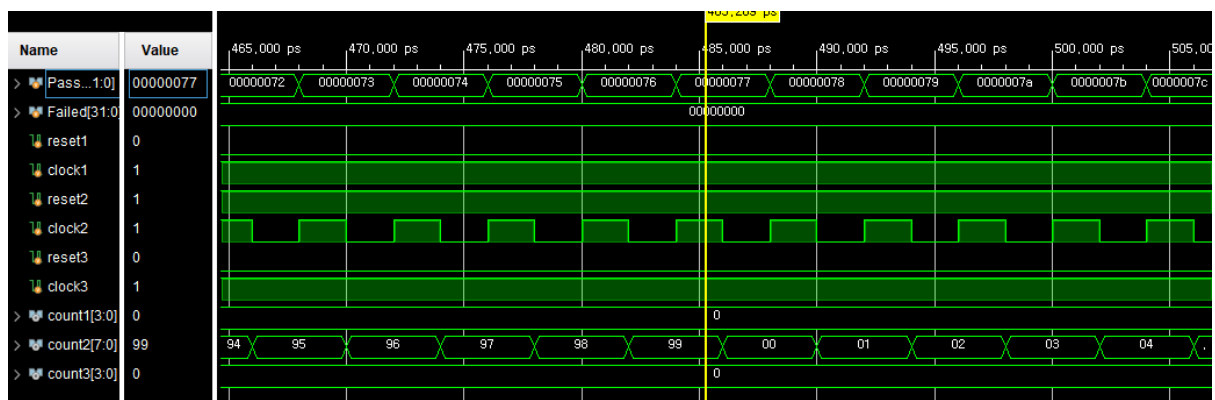
실험 준비 단계에서 구성한 대로, 두 자릿수 Decade BCD counter는 한 자릿수 카운터 두 개를 연결하여 구현할 수 있고, 따라서 실험1에서 구현한 decade_counter 모듈을 이용하여 구현하였다. 이때 두 번째 카운터의 clock에는, 실험 준비 단계에서 구상한 것처럼, count[3]&count[0]을 넣어 주었다. 완성된 코드로 schematic 기능으로 회로를 확인하니 아래와 같이 나오는 것을 확인할 수 있었다.



이로써 실험 준비 단계에서 구성한 대로 회로가 구현되었음을 알 수 있었고, 주어진 testbench를 완성하여 시뮬레이션을 돌려 작동을 확인하니 아래와 같이 나타났다.



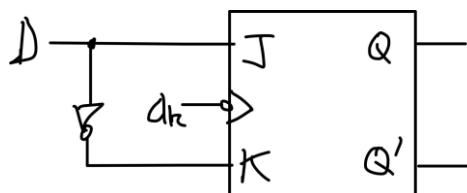
위와 비슷하게 testbench를 clock2가 4ps 주기로 진동하도록 설정하였고, 그에 따라 계수기가 0부터 99까지 그리고 99에서 다시 0으로 잘 작동하는 것을 확인할 수 있었다. 또한, Passed도 제대로 늘어나고, Failed는 0인 것을 보아 구성한 회로가 제 기능을 함을 확인할 수 있다. 시뮬레이션 결과를 확대하면 아래와 같다.



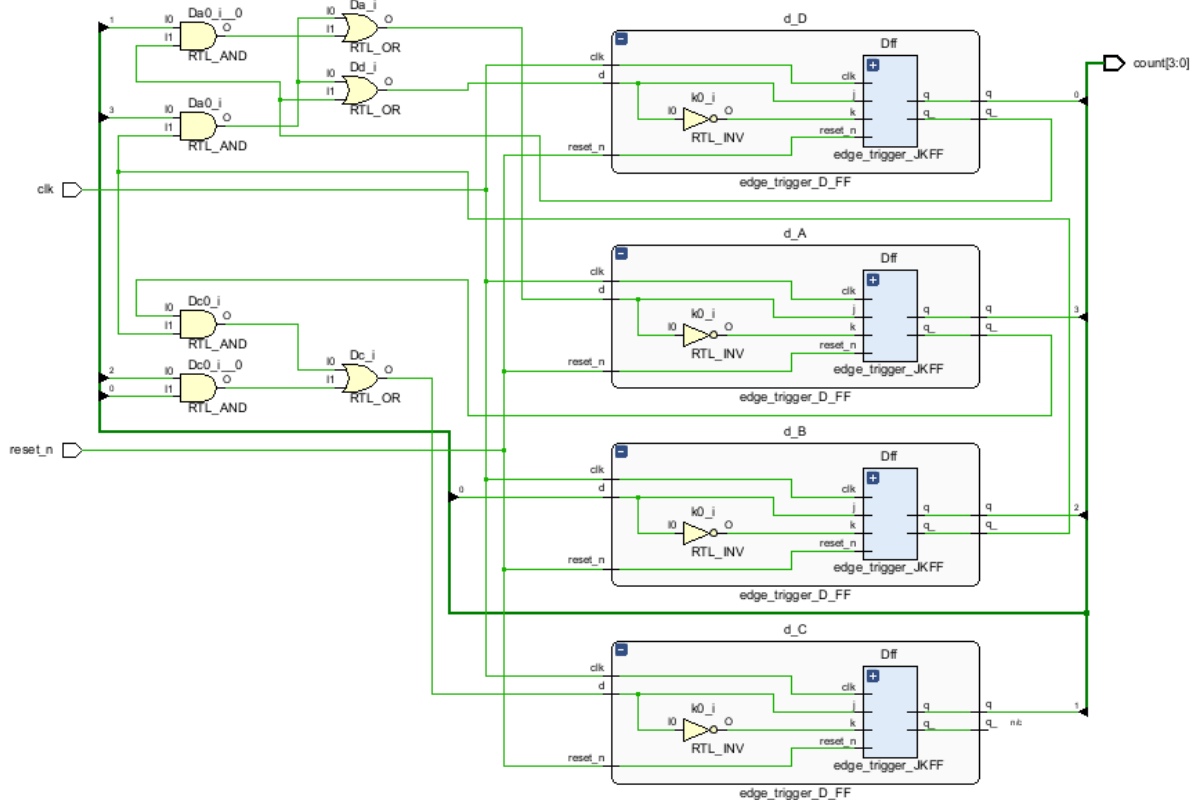
위 시뮬레이션 결과 그림과 같이 99 다음으로는 00으로 잘 되돌아가는 것을 확인할 수 있다. 이로써 두자리 BCD 계수기가 잘 작동함을 확인하였다.

3) D플립플롭을 이용한 3, 6, 9 counter

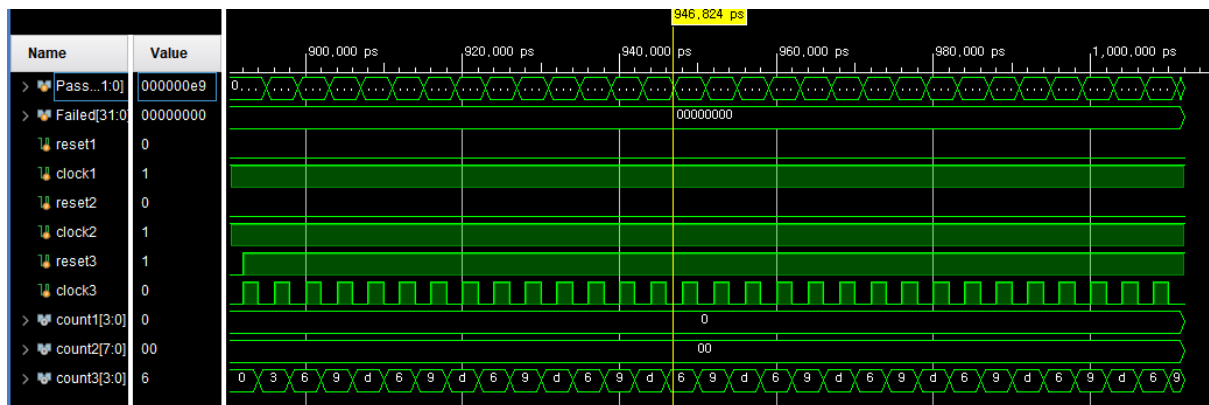
우선, lab6_ff.v 파일에서 제공된 Negative edge triggered JK flipflop으로 Negative edge triggered D flipflop을 구현한다. 아래 회로도도 같이 구현 가능하므로 이를 Verilog 코드로 구현하였다.



이후 이 Negative edge triggered D flipflop을 이용하여 실험 준비 단계에서 구상한 대로 3, 6, 9 계수기를 제작하였다. 그 결과 Schematic으로 회로도도 확인하니 아래의 결과를 얻을 수 있었다.



회로도를 살펴보면, D flipflop도 주어진 JK flipflop으로 잘 구성되었고, 나머지 회로도도 실험 준비 단계에서 구상한 것과 일치함을 확인할 수 있었다. 다음으로 testbench를 완성하여 시뮬레이션을 돌려 작동을 확인해본 결과 아래와 같은 결과를 얻을 수 있었다.



위에서 한 것과 동일하게 clock3 값을 4ps 주기로 진동시켰고, 그에 따라 count3 값이 0, 3, 6, 9, 13(d), 6, 9, ...로 제대로 순환하는 것을 확인할 수 있었다. 또한 마찬가지로 Passed가 제대로 가산된 걸 보아, 카운터가 제대로 작동하였음을 알 수 있다. 3, 6, 9 카운터 또한 제대로 구현되었음을 확인할 수 있다.

5. 논의

이번 lab6를 통해서 카운터의 개념과 JK 플립플롭, D 플립플롭을 이용한 그 구현에 대해서 공부할 수 있었다. 또한, 복잡한 전이표를 그리고 실수 없이 회로도를 그리는 것을 연습할 수 있었다.

또한, 이전 lab에서 자세하게 다루지는 않았던 testbench 코드를 직접 작성해보면서 어떤 식으로 testbench를 작성해야 하는지 알게 되었다. 추가로, lab6을 진행하면서 크게 어려웠던 점은 없이 회로도 까지는 잘 완성했지만, 2자릿수 카운터의 testbench 시뮬레이션에서 약간의 문제가 있었다. 00부터 10까지는 Passed가 나오는데, 그 이후 Failed가 나오고 다시 99에서 00으로 넘어가니 Passed가 나오기 시작하는 문제였다. 사실 이는 BCD 2 자릿수여서 00001001에서 다음으로는 00010000으로 넘어가야 하는데 그냥 단순히 1을 더해서 count_expected와 count가 일치하지 않아서 벌어진 일이었고, 코드를 수정하여 Passed가 뜨도록 바로잡을 수 있었다.