디지털시스템설계 실습 보고서

6. 순차회로와 계수기

20210716 최대현

1. 실습 개요

이번 실습에서는 순차회로와 계수기를 다룬다. 순차회로(sequential logic)의 대표적인 예시 중 하나인 계수기(Counter)의 특성을 알아보고 다양한 계수기를 구현해 보는 것이 목표이다.

2. 이론적 배경

1) 순차회로

반가산기(half adder)는 이진수의 한자리수를 연산하고, 자리올림수 출력(carry out)에 따라 자리올림수를 출력하는 연산을 수행한다. Input은 1bit 수 A와 B를 받고, output으로 C(carry-out)와 S(Sum)을 출력한다.

2) 계수기

계수기(Counter)는 Clock Pulse를 세어서 수치를 처리하기 위한 논리 회로이다. 출력 패턴을 달리 하여 사용 목적에 맞게 구현할 수 있다. 동기 계수기(synchronous counter) 순차 계수기(ripple counter)로 나눌 수 있고, 일반적으로 동기 계수기가 순차 계수기보다 구현은 복잡하지만 작동 속도가 빠르고 일반화하여 만들 수 있기 때문에 선호된다.

3) 플립플롭

플립플롭은 Latch와 달리 입력이 바뀔 때 출력이 Clock에 맞추어 반영되는 동기 회로이다. JK 플립플롭과 그것에서 파생된 D 플립플롭, T 플립플롭 등이 있다. 플립플롭은 Clock 신호를 받아 작동한다. Clock의 state에 작동 여부가 달라지고, Clock이 1일 때 작동하는 positive-clock flipflop과 clock이 0일 때 작동하는 negative-clock flipflop 두 종류가 존재한다.

이번 실험에서는 주어진 JK 플립플롭을 가지고 D 플립플롭을 구현한다. JK 플립플롭의 K에 inverter을 적용한 뒤 두 input을 묶기만 하면 되는, 아주 간단한 방법으로 구현할 수 있다.

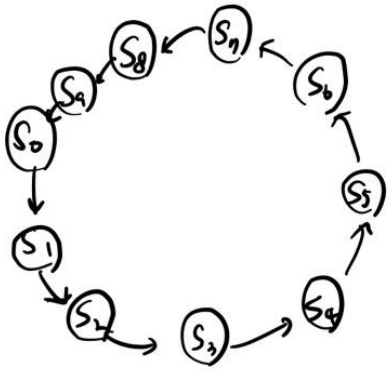
4) 상태 전이도 및 전이표

상태 전이도와 상태 전이표(state transition diagram, state transition table)는 순차회로의 상태 변화를 그림 혹은 표로 나타낸 것이다. 계수기 등 순차회로를 구현하기 위해서는 단순화된 상태 전이도와 전이표를 잘 구성하는 것이 유리하다.

3. 실험 전 준비

1) JK 플립플롭을 이용한 Synchronous decade BCD counter

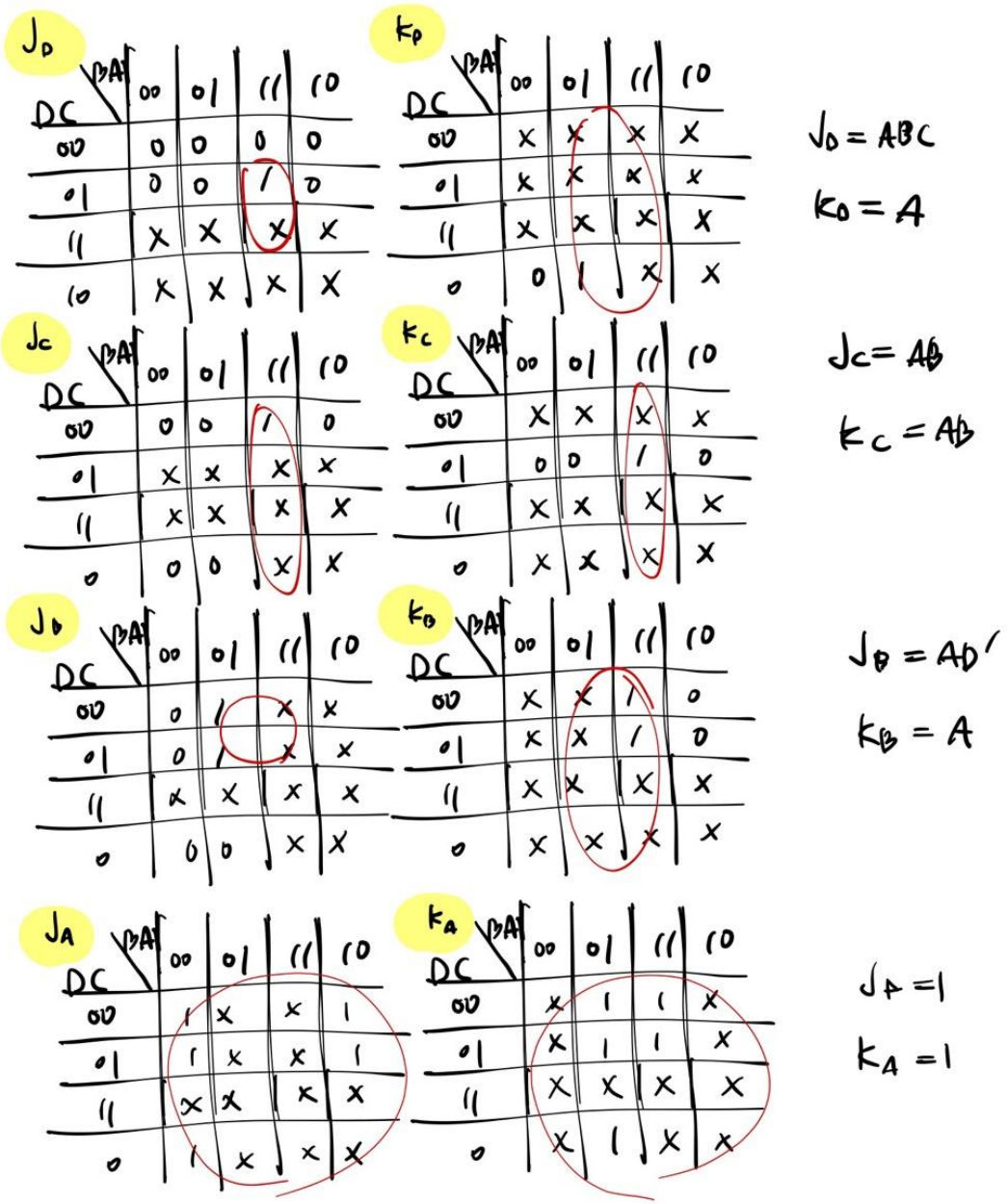
<state transition diagram>



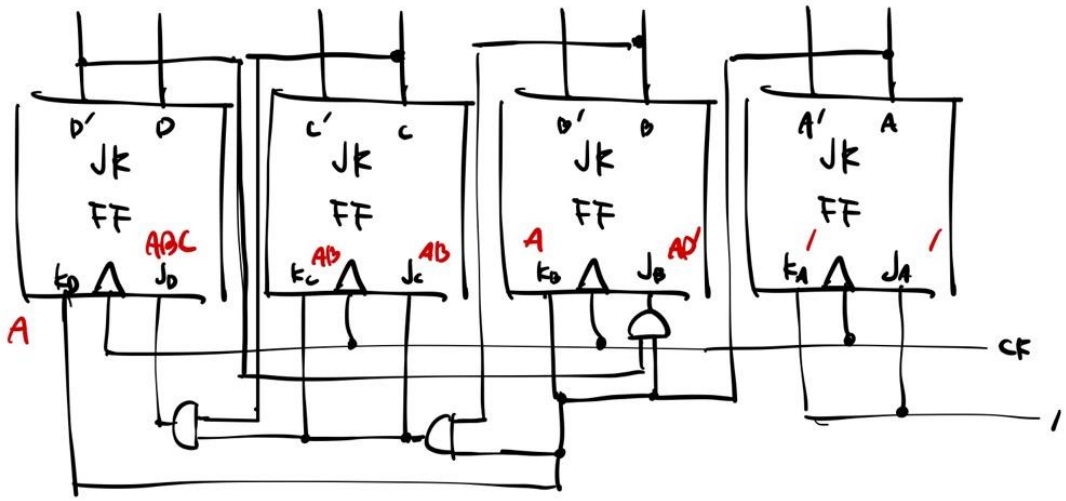
<state transition table>

Present D C B A	Next $D^+ C^+ B^+ A^+$	J-K input $J_D K_D$	$J_C K_C$	$J_B K_B$	$J_A K_A$
S_0 0 0 0 0	S_1 0 0 0 1	0 X	0X	0X	1X
S_1 0 0 0 1	S_2 0 0 1 0	0X	0X	1X	X1
S_2 0 0 1 0	S_3 0 0 1 1	0X	0X	X0	1X
S_3 0 0 1 1	S_4 0 1 0 0	0X	1X	X1	X1
S_4 0 1 0 0	S_5 0 1 0 1	0X	X0	0X	1X
S_5 0 1 0 1	S_6 0 1 1 0	0X	X0	1X	X1
S_6 0 1 1 0	S_7 0 1 1 1	0X	X0	X0	1X
S_7 0 1 1 1	S_8 1 0 0 0	1X	X1	X1	X1
S_8 1 0 0 0	S_9 1 0 0 1	X0	0X	0X	1X
S_9 1 0 0 1	S_0 0 0 0 0	X1	0X	0X	X1
S_{10}~S_{15}		XX	XX	XX	XX

<JK FF simplification>



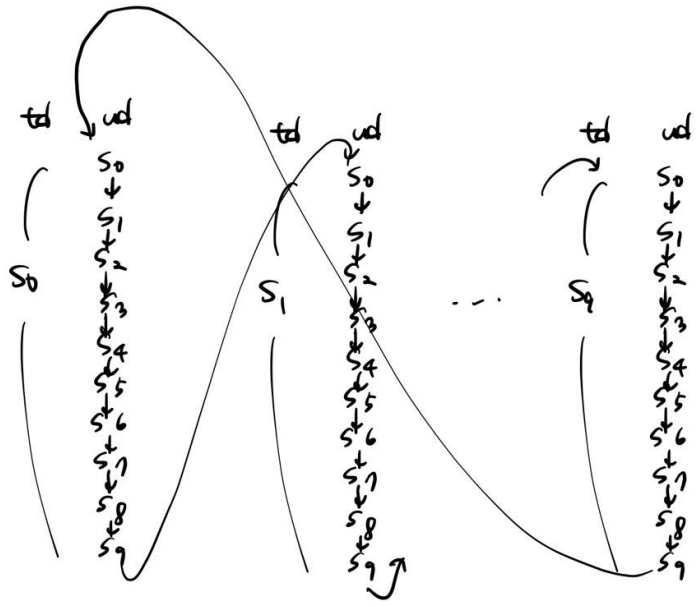
<회로도>



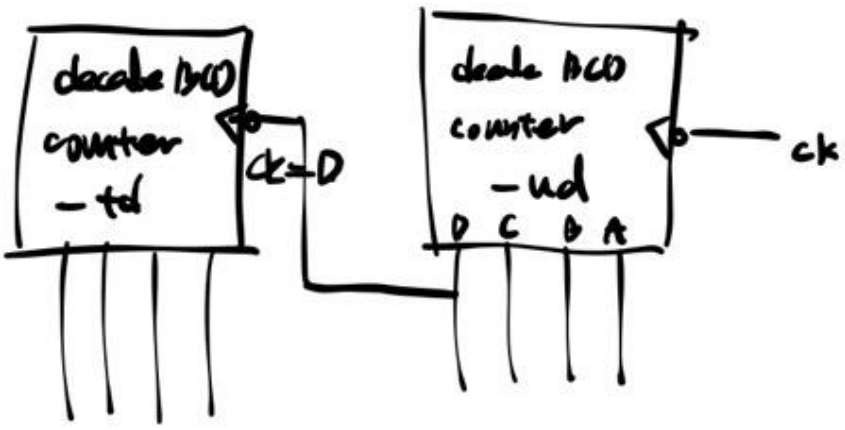
2) JK 플립플롭을 이용한 두 자릿수 Decade BCD counter (0~99)

Negative clock JK flipflop이므로, clock이 1->0일 때 다음 state로 간다. 자릿수가 증가하는 때를 체크해보면, 자릿수가 증가하는 시점은 일의 자릿수의 D값이 1->0으로 바뀌는 과정이다. 즉, clock이 작동해야 하는 경우와 같으므로 십의 자릿수의 Clock = D이다.

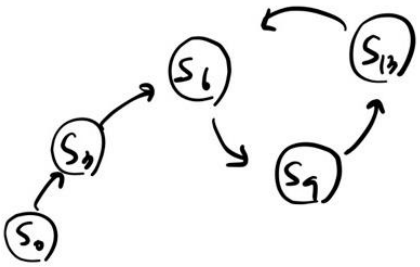
. **<state transition diagram>**



<회로도>



3) D 플립플롭을 이용한 3, 6, 9 계수기

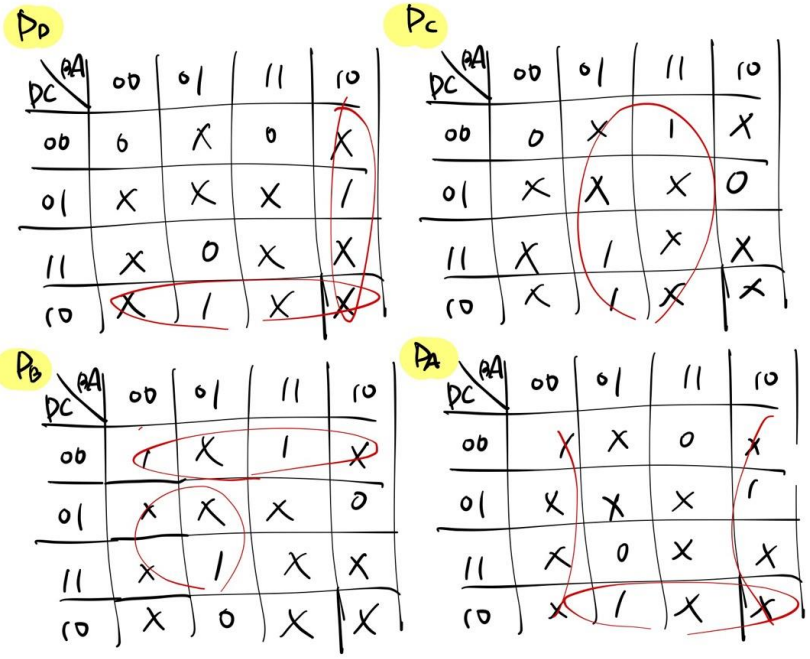


(S0: 0000, S3:0011, S6:0101, S9:1001, S13:1101)

<state transition diagram>

	Present $D\,C\,B\,A$	Next $D\,C\,B\,A$	$D_D\,D_C\,D_B\,D_A$
S_0	0 0 0 0	0 0 1 1	0 0 1 1
S_3	0 0 1 1	0 1 1 0	0 1 1 0
S_6	0 1 1 0	1 0 0 1	1 0 0 1
S_9	1 0 0 1	1 1 0 1	1 1 0 1
S_{13}	1 1 0 1	0 1 1 0	0 1 1 0
		- - - -	x x x x

<state transition table>



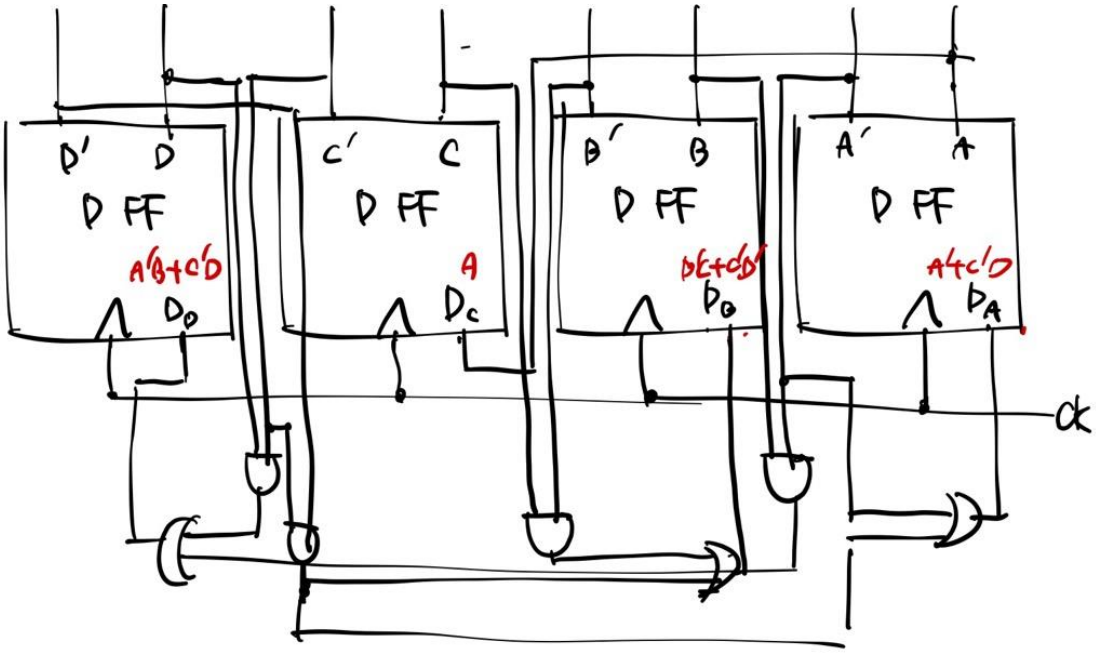
$D_D = A'B + C'D$

$D_C = A$

$D_B = B'C + C'D'$

$D_A = A' + C'D$

<D FF simflipication>



<회로도>

4. 실험

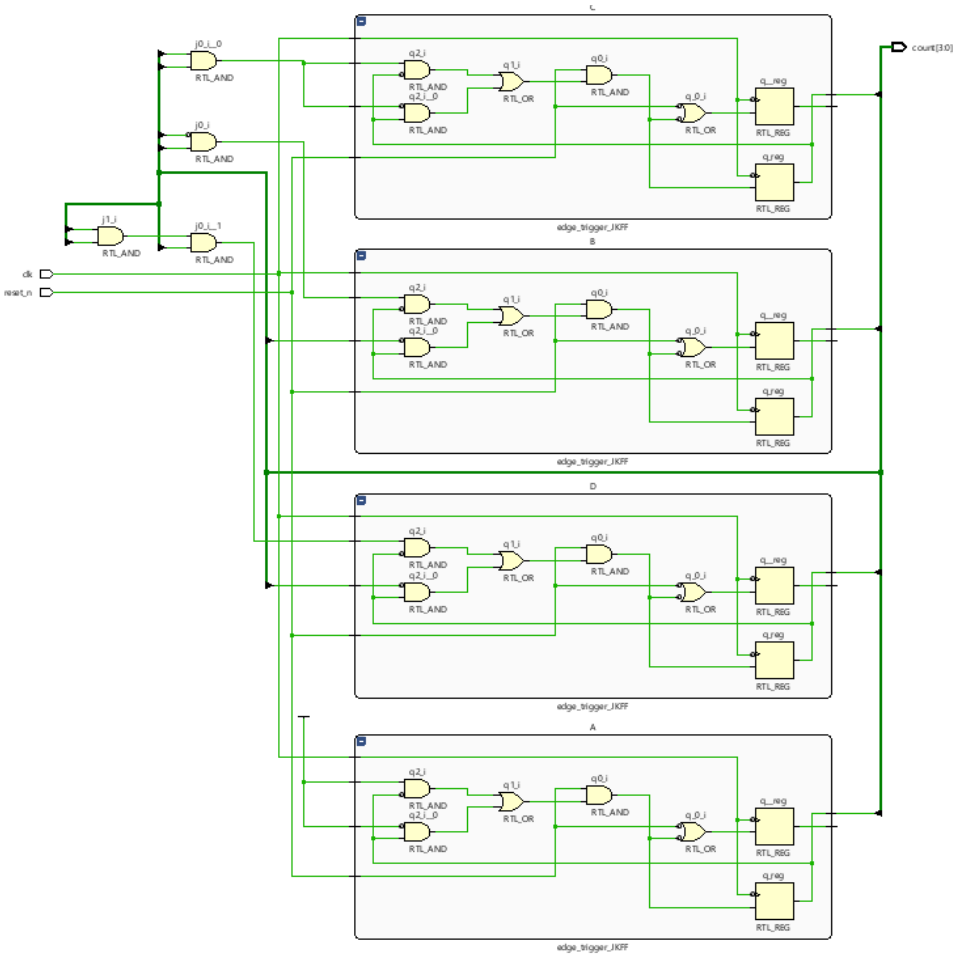
본 실험은 Verilog 프로그래밍으로 회로를 구현하며 진행되었다.

6-1. lab6_1.v

JK 플립플롭을 이용한 Synchronous decade BCD counter를 구현한 코드이다.

```
module decade_counter(input reset_n, input clk, output [3:0] count);

    ////////////////////////
    /* Add your code here */
    wire [3:0] ncount;
    edge_trigger_JKFF D(reset_n,count[2]&count[1]&count[0],count[0],clk,count[3],ncount[3]); // J_D = ABC , K_D = A
    edge_trigger_JKFF C(reset_n,count[1]&count[0],count[1]&count[0],clk,count[2],ncount[2]); // J_C = AB , K_C = AB
    edge_trigger_JKFF B(reset_n,~count[3]&count[0],count[0],clk,count[1],ncount[1]); // J_B = AD' , K_B = A
    edge_trigger_JKFF A(reset_n,1'b1,1'b1,clk,count[0],ncount[0]); // J_D = 1 , K_D = 1

    ////////////////////////

endmodule
```

<source code>



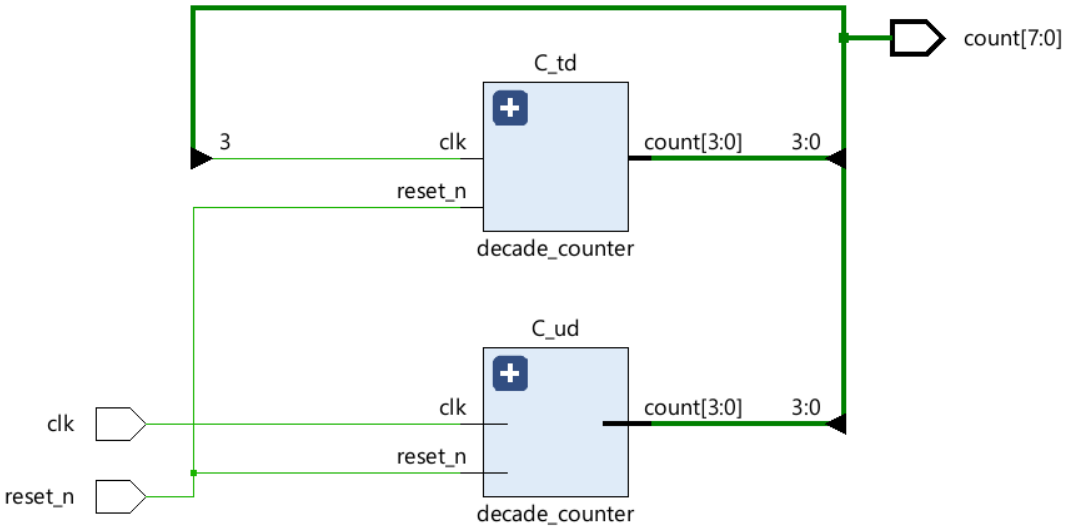
<Schematic 기능으로 출력한 회로도>

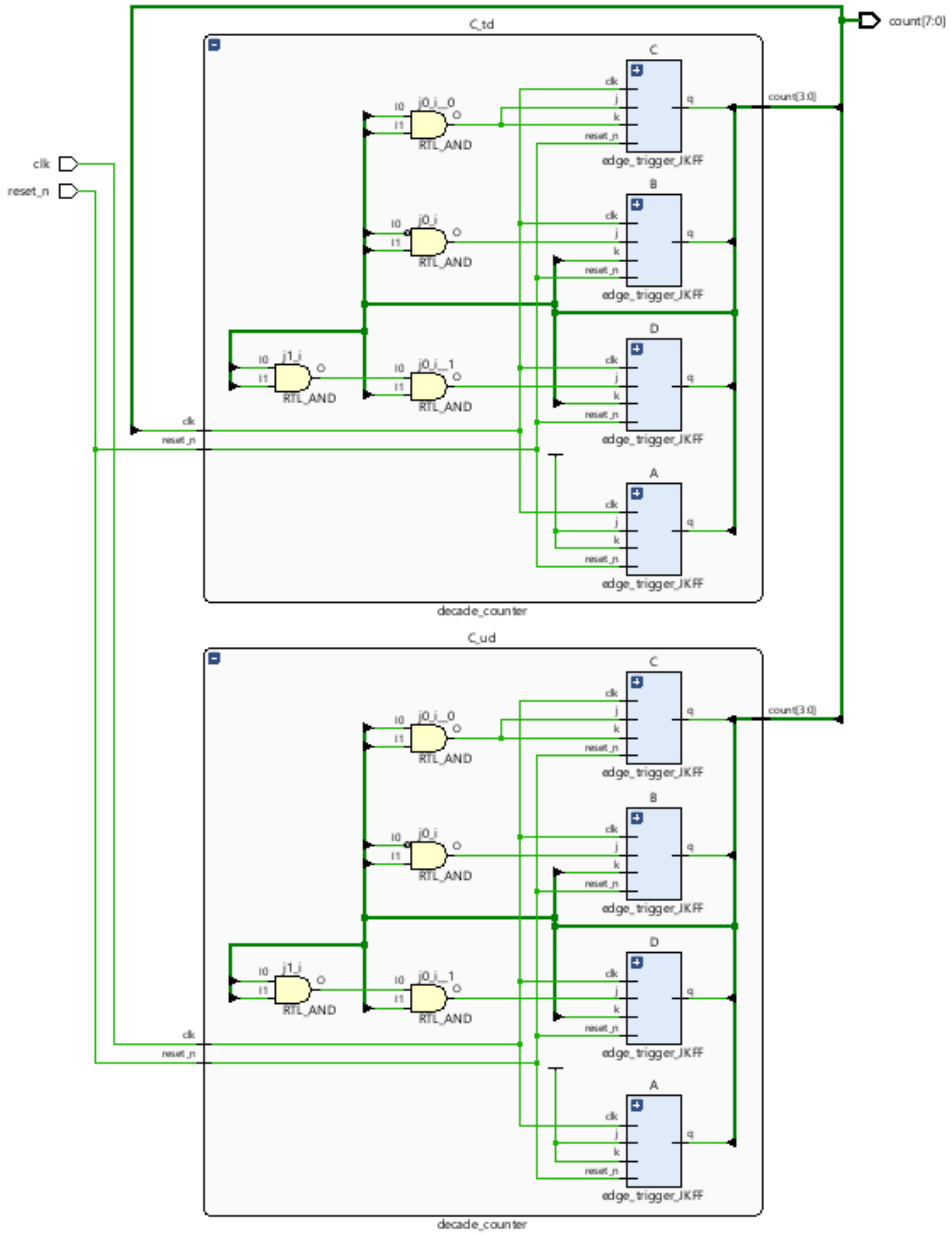
6-2. lab6_2.v

JK 플립플롭을 이용한 두 자릿수 Decade Counter를 구현한 코드이다.

```
/* CSED273 lab6 experiment 2 */
/* lab6_2.v */

`timescale 1ps / 1fs

/* Implement 2-decade BCD counter (0-99)
 * You must use decade BCD counter of lab6_1.v */
module decade_counter_2digits(input reset_n, input clk, output [7:0] count);

    ////////////////////////
    /* Add your code here */
    decade_counter C_ud(reset_n,clk,count[3:0]); // ud, 일의 자리
    decade_counter C_td(reset_n,count[3],count[7:4]); // td, 십의 자리, count[3]이 1->0 으로 바뀔 때 마다 신호를 받아서 다음 state로 넘어감.
    ////////////////////////

endmodule
```

<source code>





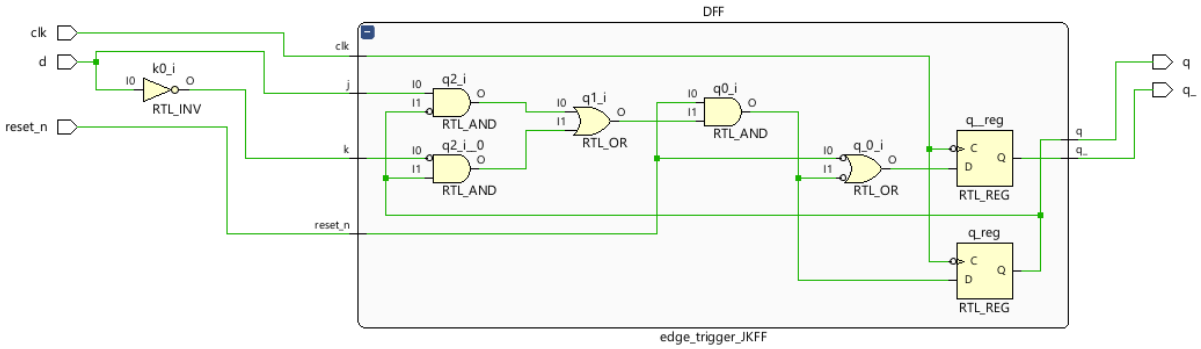
<Schematic 기능으로 출력한 회로도>

6-3. lab6_3.v

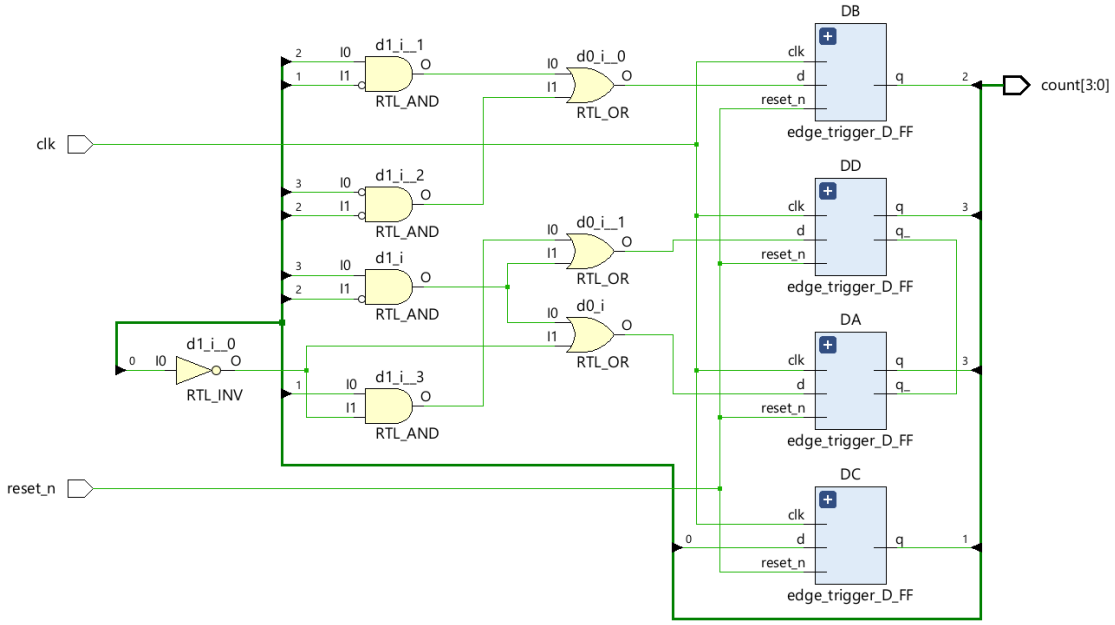
D 플립플롭을 이용한 3, 6, 9 계수기를 구현한 코드이다.

```
//////////////////////
/* Add your code here */
wire [3:0] ncount;
edge_trigger_D_FF DD(reset_n,(count[1]&~count[0])|(count[3]&~count[2]),clk,count[3],ncount[3]);
edge_trigger_D_FF DC(reset_n,count[0],clk,count[2],ncount[2]);
edge_trigger_D_FF DB(reset_n,(count[2]&~count[1])|(~count[3]&~count[2]),clk,count[1],ncount[1]);
edge_trigger_D_FF DA(reset_n,(count[3]&~count[2])|~count[0],clk,count[0],ncount[0]);
//////////////////////

dmodule
```

<source code>



<D flip flop - Schematic 기능으로 출력한 회로도>

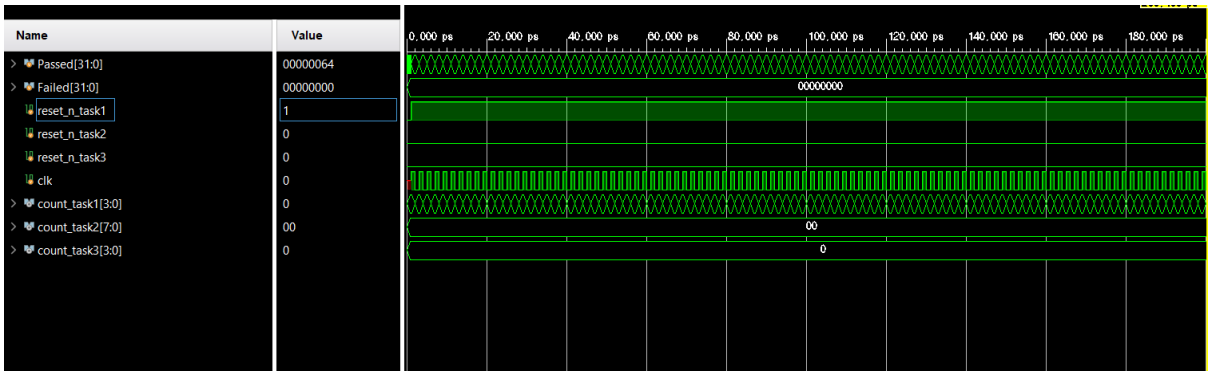


<Schematic 기능으로 출력한 회로도>

Test 결과

이번 실험의 경우 모든 testbench file이 하나로 묶여 있으므로, testbench에 대한 결과를 한 번에 정리했다.

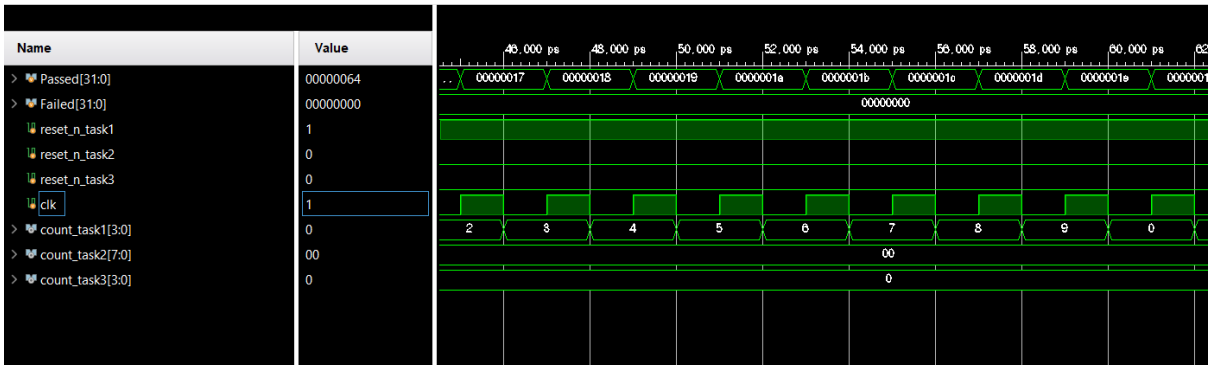
1. Lab6_1



<simulation 결과>

우선, 2중 포문에서 바깥과 안쪽 포문을 10회씩 수행하고, 수행 1회 당 2 timescale만큼 작동하므로 20ps까지의 결과를 캡쳐해서 첨부했다. 이 동안 passed는 계속해서 증가하는 반면 failed는 0을 유지하므로 계속해서 작동이 원활하게 발생했음을 알 수 있다. 또한 clk(clock)이 지속적

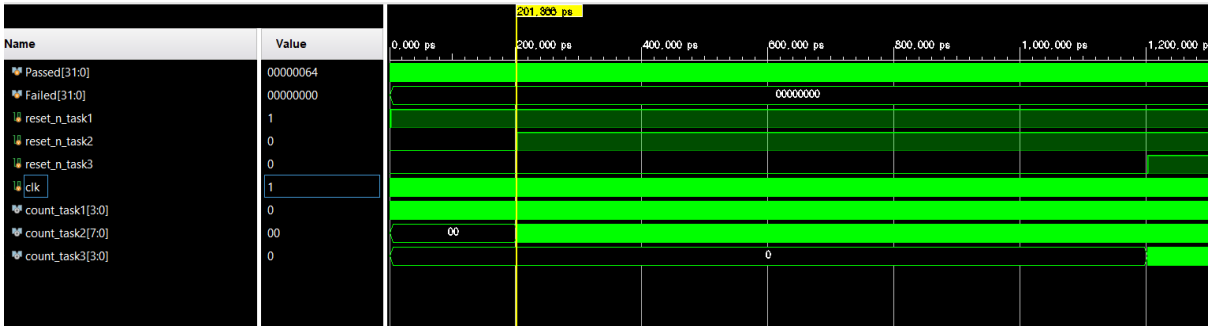
으로 진동하고, 이 때마다 count_task값이 0~9까지를 반복하는 것을 확인할 수 있다.

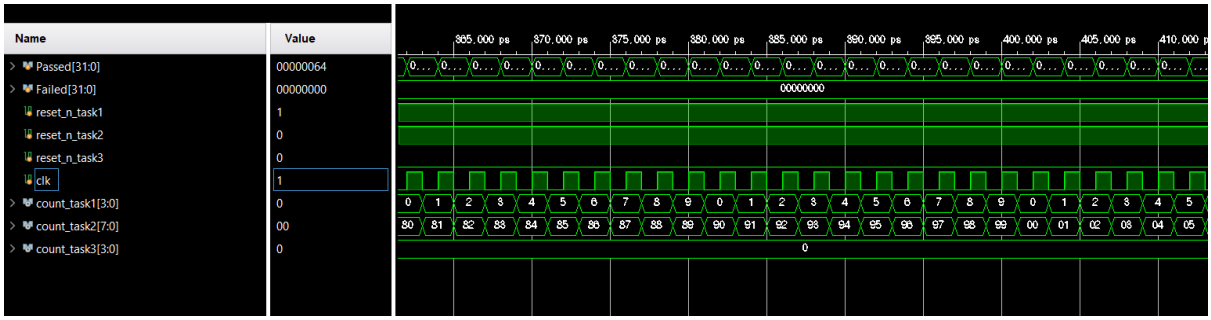


(0부터 9까지 반복하는 모습)

2. Lab6_2

Testbench 코드를 짤 때 3중 포문에서 각 포문을 5, 10, 10회씩 수행하고, 수행 1회 당 2 timescale만큼 작동하므로 20~120ps까지의 결과를 캡쳐해서 첨부했다. 이 동안 passed는 계속해서 증가하는 반면 failed는 0을 유지하므로 계속해서 작동이 원활하게 발생했음을 알 수 있다. 또한 clk(clock)이 지속적으로 진동하고, 이 때마다 count_task값이 0~99까지를 반복하는 것을 확인할 수 있다.

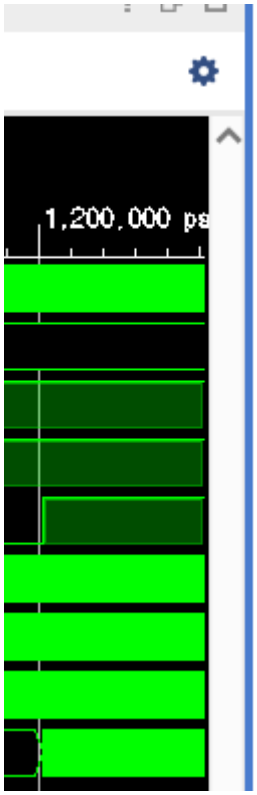


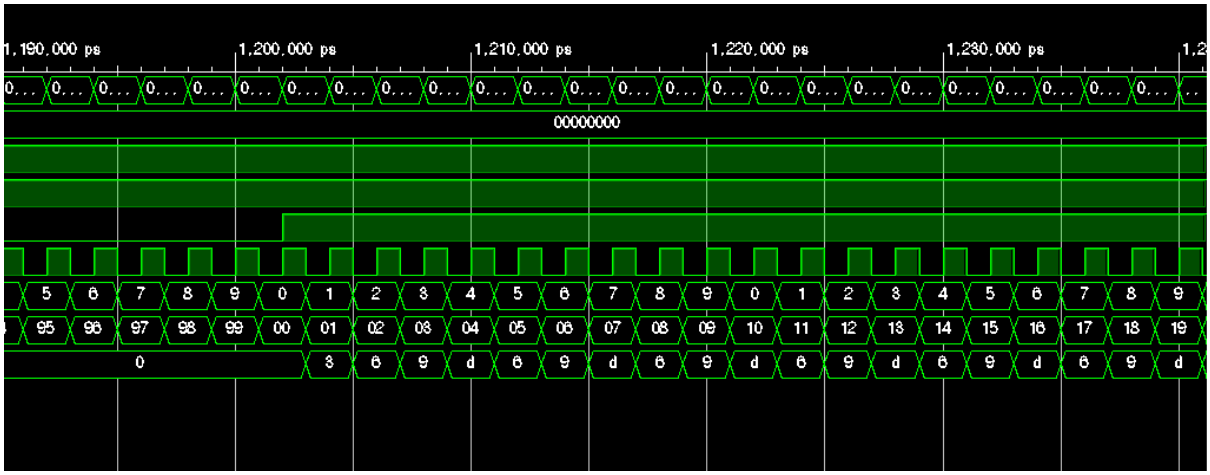


(0부터 99까지 반복하는 모습)

3. Lab6_3

Testbench 코드를 짤 때 단일 포문을 50회수행하고, 수행 1회 당 2 timescale만큼 작동하므로 120~140ps까지의 결과를 캡쳐해서 첨부했다. 이 동안 passed는 계속해서 증가하는 반면 failed는 0을 유지하므로 계속해서 작동이 원활하게 발생했음을 알 수 있다. 또한 clk(clock)이 지속적으로 진동하고, 이 때마다 count_task값이 0,3,6,9,13,6,9...를 반복하는 것을 확인할 수 있다.





(0, 3, 6, 0, 13, 6, 9.. 를 반복하는 모습)

5. 결론

Lab 6_1에서는 주어진 JK 플립플롭 모듈을 이용해서 Synchronous decade BCD counter을 만들었다. 이를 이용하여 Lab 6_2에서는 두 자릿수 Decade Counter를 만들었다. 마지막으로, Lab6_3에서는 제공된 Negative edge triggered JK 플립플롭에서 K input에 inverter을 취하고, 두 input을 묶어서 Negative edge triggered D 플립플롭을 만든 뒤 이를 이용해서 3, 6, 9 계수기를 만들었다.

6. 고찰

이번 실습에서는 이론적으로 학습했던 두 종류의 플립플롭(JK FF, D FF)를 구현해보았다. 또한, 이를 활용하여 계수기(counter)도 만들어보았다. 특히 JK 플립플롭을 이용하여 D 플립플롭을 직접 구현해보는 과정이 가장 인상깊었다. 그리고 테스트벤치를 짜는 방법이 다소 어려웠지만, 이전 랩들을 참고하여 완성했다. 순차회로에서 가장 보편적으로 사용되는 JK 플립플롭을 이용해서 직접 동기 계수기를 회로로 구현해봄으로서 매우 유익한 시간이었다.