

0. 문제

Show a State Diagram and Logic Design(using D-FF) for Sequential Logic controller with Simulation

If input(X) is a charger button,

Apollo Battery has Up(Charge"1")/Down(Discharge "0") as Counter with Clock-time(100ns) based, when X-Charger input

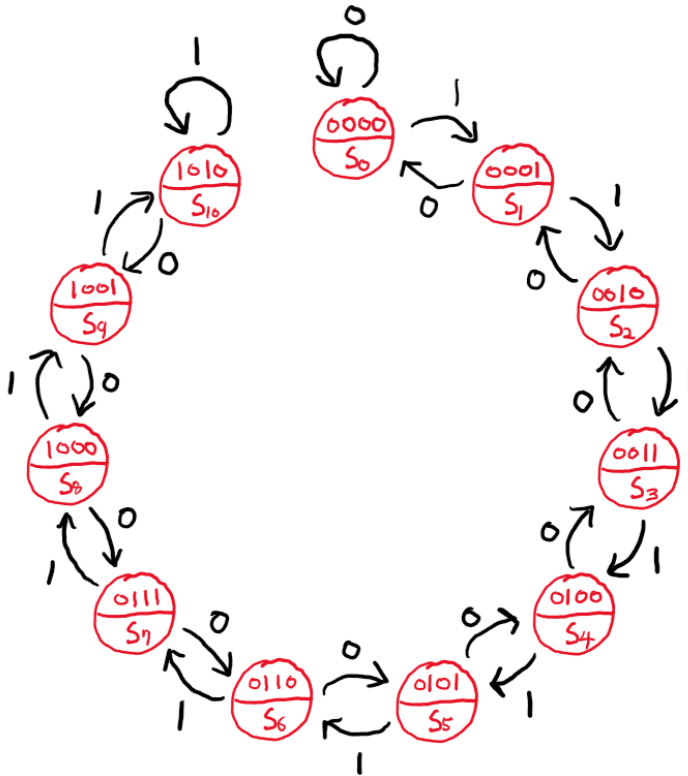
- Charger & Discharge operate as 1-step up or down
- State(Green/ Yellow / Red)
- Input(X)

A logic Design that the output(Green) is greater than 6-V corresponding to battery voltage in step of 1V, Also Output(Yellow) has an alarm signal when between 7 to 10V. And produce the Output(Red) when below 6V.

- Output display(7-Segment) the battery Range of the state(G/Y/R)

~~ Happy and Great for your Study, Thank You.~~~

1. 상태도



방전하한(0V)과 충전상한(10V)에 도달한 경우 차기 상태는 각 하한과 상한으로 설정하여 주어진 범위 내에서 회로가 작동하도록 설계하였다.

2. 전체 회로 설계 (D 플립플롭을 이용한 로직 설계)

상태도를 참고하여 상태표를 만들면 다음과 같은 표를 얻을 수 있다.

현재 상태					값	입력	차기 상태					값	현재 상태					값	입력	차기 상태					값
D	C	B	A	S	X	D	C	B	A	S			D	C	B	A	S	X	D	C	B	A	S		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1		
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	2		
0	0	1	0	2	0	0	0	0	1	1	0	0	1	0	2	1	1	0	0	1	1	1	3		
0	0	1	1	3	0	0	0	1	0	2	0	0	1	1	3	1	1	0	1	0	0	0	4		
0	1	0	0	4	0	0	0	1	1	3	0	1	0	0	4	1	1	0	1	0	1	1	5		
0	1	0	1	5	0	0	1	0	0	4	0	1	0	1	5	1	1	0	1	1	0	0	6		
0	1	1	0	6	0	0	1	0	1	5	0	1	1	0	6	1	1	0	1	1	1	1	7		
0	1	1	1	7	0	0	1	1	0	6	0	1	1	1	7	1	1	1	0	0	0	0	8		

1	0	0	0	8	0	0	1	1	1	7	1	0	0	0	8	1	1	0	0	1	9
1	0	0	1	9	0	1	0	0	0	8	1	0	0	1	9	1	1	0	1	0	10
1	0	1	0	10	0	1	0	0	1	9	1	0	1	0	10	1	1	0	1	0	10
1	0	1	1	11	0	x	x	x	x	x	1	0	1	1	11	1	x	x	x	x	x
1	1	0	0	12	0	x	x	x	x	x	1	1	0	0	12	1	x	x	x	x	x
1	1	0	1	13	0	x	x	x	x	x	1	1	0	1	13	1	x	x	x	x	x
1	1	1	0	14	0	x	x	x	x	x	1	1	1	0	14	1	x	x	x	x	x
1	1	1	1	15	0	x	x	x	x	x	1	1	1	1	15	1	x	x	x	x	x

상태표를 참조하여 D 플립플롭 D, C, B, A 에 대한 차기 상태 $D(t+1)$, $C(t+1)$, $B(t+1)$, $A(t+1)$ 을 구하면 다음과 같다.

$$D(t+1)=(!X\wedge D\wedge !C\wedge !B\wedge A)+(!X\wedge D\wedge !C\wedge B\wedge !A)+(X\wedge !D\wedge C\wedge B\wedge A)+(X\wedge D\wedge !C\wedge !B\wedge !A)+(X\wedge D\wedge !C\wedge !B\wedge A)+(X\wedge D\wedge !C\wedge B\wedge !A)$$

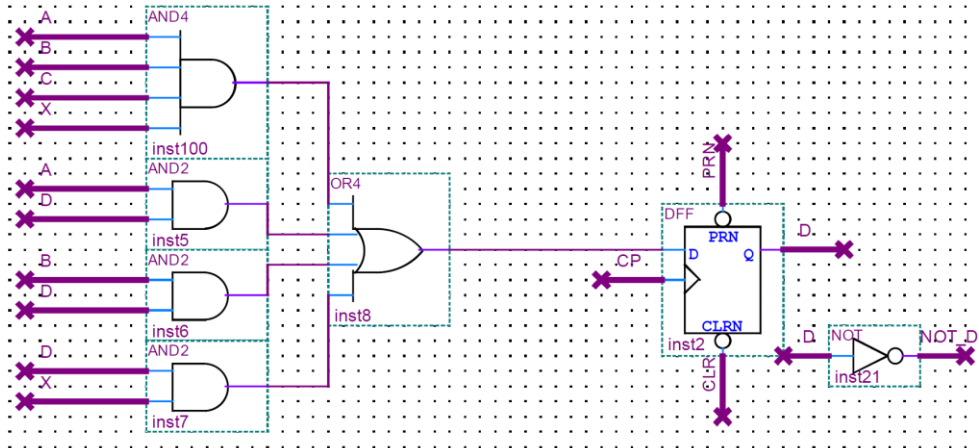
$$C(t+1)=(!X\wedge !D\wedge C\wedge !B\wedge A)+(!X\wedge !D\wedge C\wedge B\wedge !A)+(!X\wedge !D\wedge C\wedge B\wedge A)+(!X\wedge D\wedge !C\wedge !B\wedge !A)+(X\wedge !D\wedge !C\wedge B\wedge A)+(X\wedge !D\wedge C\wedge !B\wedge !A)+(X\wedge !D\wedge C\wedge !B\wedge A)+(X\wedge !D\wedge C\wedge B\wedge !A)$$

$$B(t+1)=(!X\wedge !D\wedge !C\wedge B\wedge A)+(!X\wedge !D\wedge C\wedge !B\wedge !A)+(!X\wedge !D\wedge C\wedge B\wedge A)+(!X\wedge D\wedge !C\wedge !B\wedge !A)+(X\wedge !D\wedge !C\wedge !B\wedge A)+(X\wedge !D\wedge !C\wedge B\wedge !A)+(X\wedge !D\wedge C\wedge !B\wedge A)+(X\wedge !D\wedge C\wedge B\wedge !A)+(X\wedge D\wedge !C\wedge !B\wedge A)+(X\wedge D\wedge !C\wedge B\wedge !A)$$

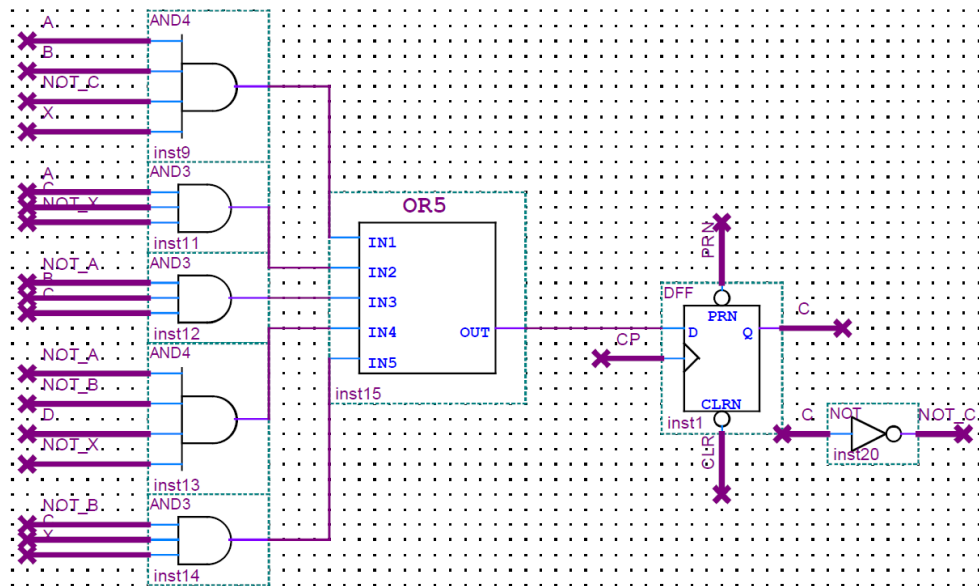
$$A(t+1)=(!X\wedge !D\wedge !C\wedge B\wedge !A)+(!X\wedge !D\wedge C\wedge !B\wedge !A)+(!X\wedge !D\wedge C\wedge B\wedge !A)+(!X\wedge D\wedge !C\wedge !B\wedge !A)+(!X\wedge D\wedge !C\wedge B\wedge !A)+(X\wedge !D\wedge !C\wedge !B\wedge !A)+(X\wedge !D\wedge !C\wedge B\wedge !A)+(X\wedge !D\wedge C\wedge !B\wedge !A)+(X\wedge !D\wedge C\wedge B\wedge !A)+(X\wedge D\wedge !C\wedge !B\wedge !A)+(X\wedge D\wedge !C\wedge B\wedge !A)$$

D 플립플롭의 특성 방정식은 $Q(t+1)=D$ 로 표현되므로 $D_D=D(t+1)$, $D_C=C(t+1)$, $D_B=B(t+1)$, $D_A=A(t+1)$ 로 표현할 수 있고 위의 식을 무관항을 고려한 카르노 맵을 이용하여 정리하고, 회로로 나타내면 다음과 같다.

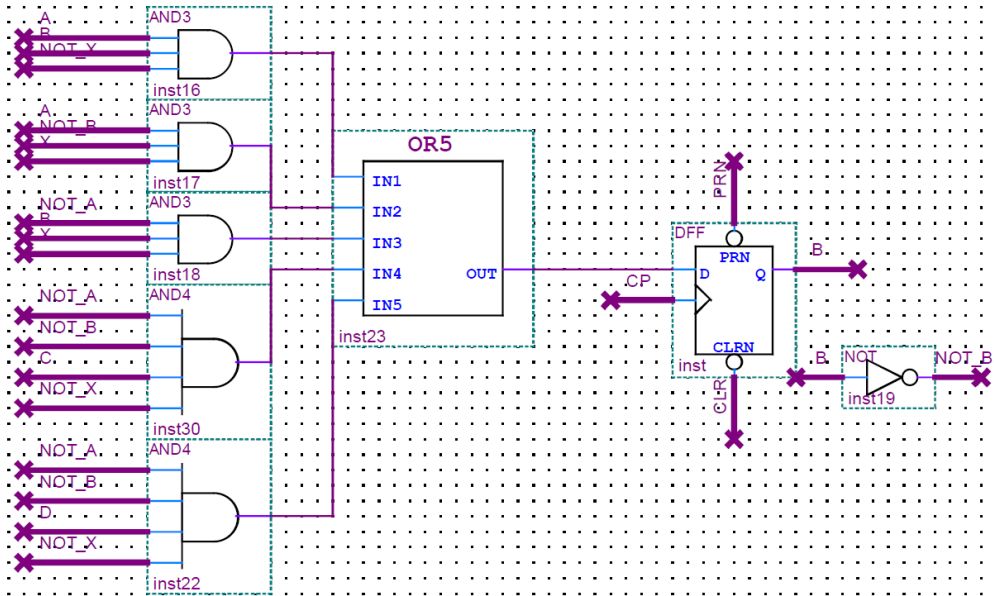
$D_D=D(t+1)$, $D_C=C(t+1)$, $D_B=B(t+1)$, $D_A=A(t+1)$ 로 표현할 수 있고 해당 식을 바탕으로 설계한 회로의 입출력 결과를 확인하면 다음과 같다.



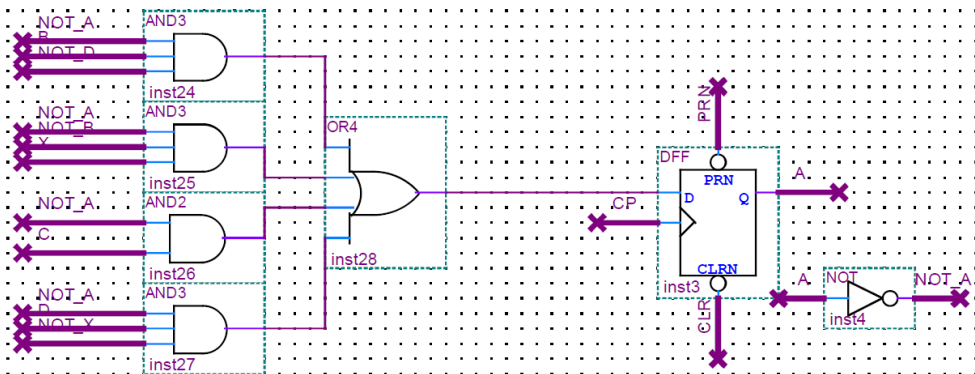
$$D_D = (A \wedge B \wedge C \wedge X) + (A \wedge D) + (B \wedge D) + (D \wedge X)$$



$$D_C = (A \wedge B \wedge !C \wedge X) + (A \wedge C \wedge !X) + (!A \wedge B \wedge C) + (!A \wedge !B \wedge D \wedge !X) + (!B \wedge C \wedge X)$$



$$D_B = (A \wedge B \wedge !X) + (A \wedge !B \wedge X) + (!A \wedge B \wedge X) + (!A \wedge !B \wedge C \wedge !X) + (!A \wedge !B \wedge D \wedge !X)$$



$$D_A = (!A \wedge B \wedge !D) + (!A \wedge !B \wedge X) + (!A \wedge C) + (!A \wedge D \wedge !X)$$

문제의 조건에 부합하는 R, Y, G LED Indicator 에 대한 진리표를 만들면 다음과 같다.

현재 상태				색깔 상태			현재 상태				색깔 상태		
D	C	B	A	R	Y	G	D	C	B	A	R	Y	G
0	0	0	0	1	0	0	1	0	0	0	0	1	1
0	0	0	1	1	0	0	1	0	0	1	0	1	1
0	0	1	0	1	0	0	1	0	1	0	0	1	1

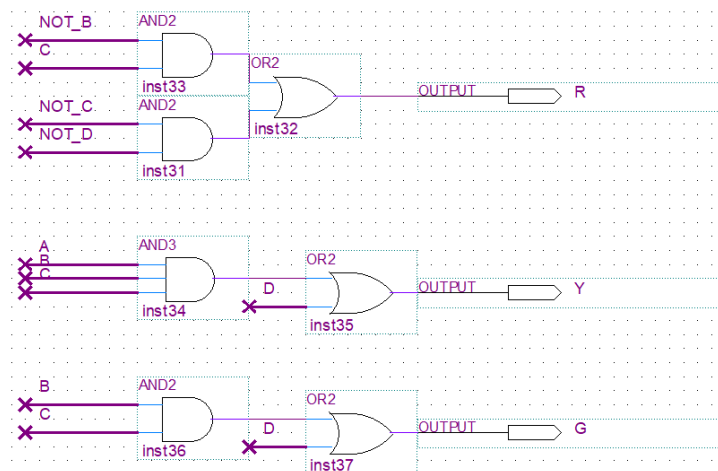
0	0	1	1	1	0	0	1	0	1	1	x	x	x
0	1	0	0	1	0	0	1	1	0	0	x	x	x
0	1	0	1	1	0	0	1	1	0	1	x	x	x
0	1	1	0	0	0	1	1	1	0	0	x	x	x
0	1	1	1	0	1	1	1	1	1	0	x	x	x

무관항을 고려한 카르노 맵을 이용하여 진리표를 SOP form 으로 정리하면 다음과 같다.

$$R = (!B \wedge C) + (!C \wedge !D)$$

$$Y = (A \wedge B \wedge C) + D$$

$$G = (B \wedge C) + D$$



(R, Y, G Indicator circuit)

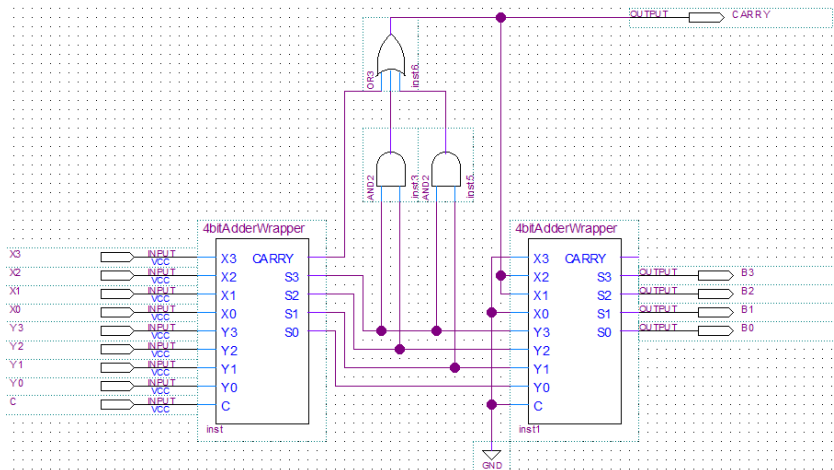
조건에 제시된 전압을 7448 드라이버(BCD to 7-Segment Decoder)를 이용하여 FND로 표시하기 위해 BCD 입력을 받아 BCD Carry와 B3, B2, B1, B0의 이진 BCD 값을 출력하는 BCDScaler를 설계 및 활용하였다. 이는 BCD 값이 10 이상일 때 1의 BCD Carry 값과 입력 BCD 값에서 10을 뺀 값을 반환하는 회로이다..

D Value	ED CARRY	aled BCD Value
0	0	0
1	0	1
2	0	2
3	0	3

4	0	4
5	0	5
6	0	6
7	0	7
8	0	8
9	0	9
10	1	0
11	1	1
12	1	2
13	1	3
14	1	4
15	1	5

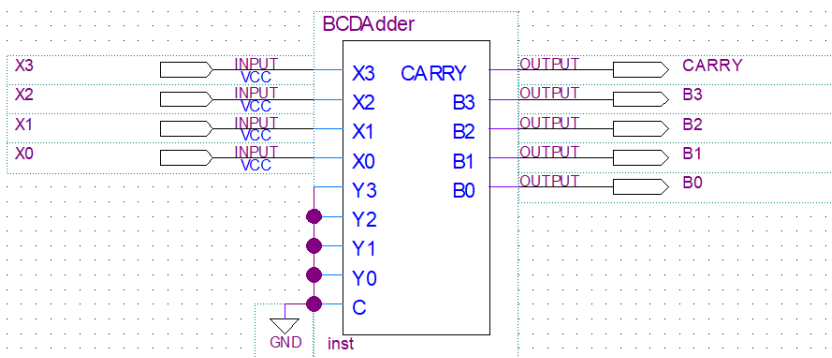
<BCDScaler의 출력표>

해당 회로를 만들기 위해서는 BCDAdder를 활용할 수 있다. BCDAdder는 두 BCD 값을 더한 BCD 값을 출력하는 회로이다. 기존 과제에서 활용한 RCA 4bitAdder를 이용하여 BCDAdder를 설계하였다.

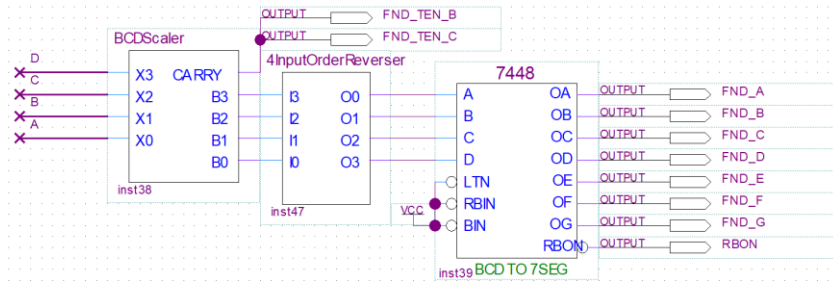


<BCDAdder>

BCDAdder는 두 수를 더한 값이 10 이상이면 CARRY를 1로 처리하기 때문에 한 입력에는 0을, 한 입력에는 스케일할 BCD 값을 주면 BCDScaler로 활용할 수 있다.



<BCDScaler>



(Voltage display circuit using 7448 driver)

과방전과 과충전을 감지하는 ALU 를 설계하기 위해서 출력 Overcharge, Overdischarge 에 대해 과방전, 과충전 조건을 통해 부울식을 세우면 다음과 같다.

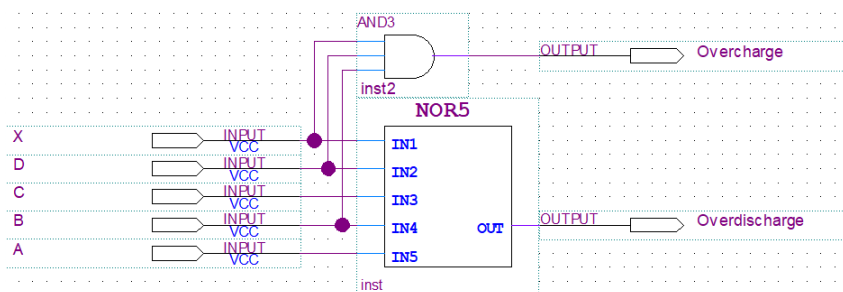
전압이 10V (1010(2) V)로 충전상한에 다다르고 X(충전 입력)가 1 일 때 과충전이므로

$$\text{Overcharge} = D \wedge B \wedge X$$

전압이 0V (0000(2) V)로 방전하한에 다다르고 X(충전 입력)가 0 일 때 과충전이므로

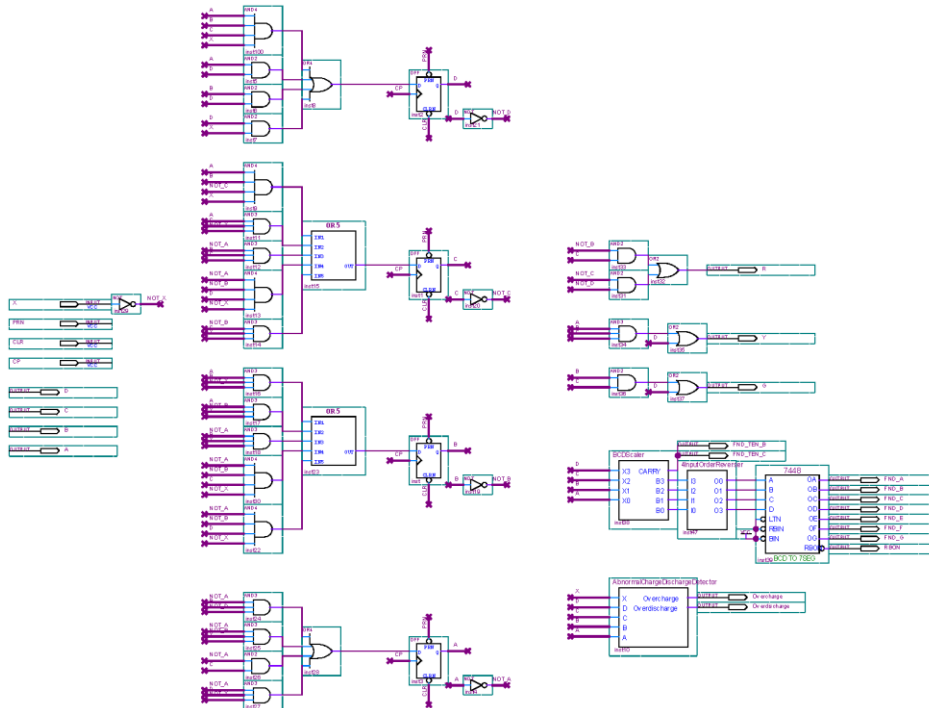
$$\text{Overdischarge} = !D \wedge !C \wedge !B \wedge !A \wedge !X$$

위의 식을 이용하여 과방전, 과충전을 감지하는 ALU 를 설계하면 다음과 같다.

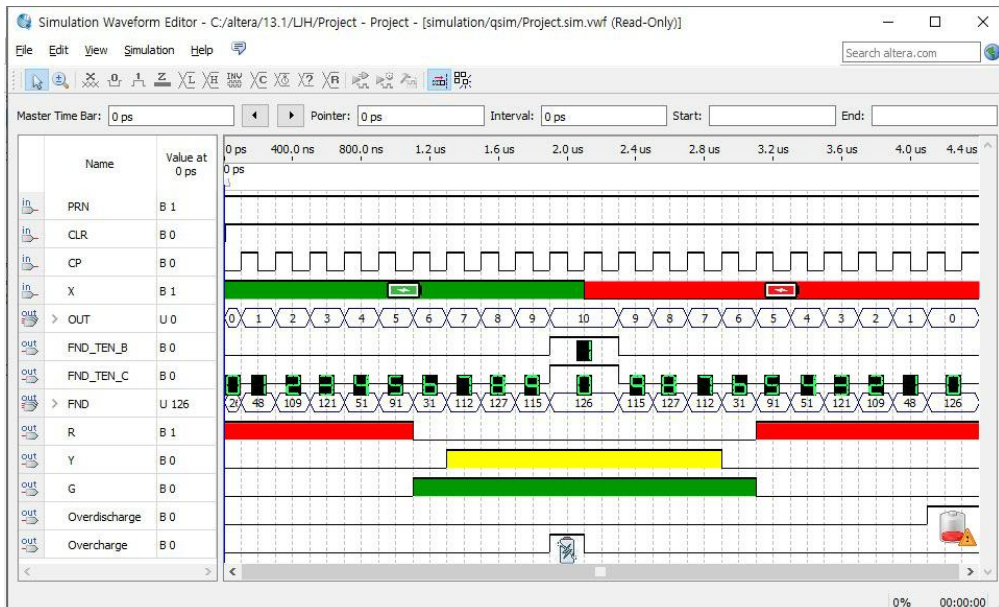


<AbnormalChargeDischargeDetector>

3. 전체 회로와 시뮬레이션



<Full circuit structure>



<Full circuit simulation>

4. 느낀 점

한 학기 동안 배운 것을 총 동원하여 본 문제와 같이 처음 강의에 입문하였을 때에는 생각지도 못했던 복잡한 회로를 설계하는 것을 해내서 정말 뿌듯함을 느꼈습니다.

한 학기 동안 카르노 맵, 조합 논리, 멀티 플렉서, FND, 플립플롭, State Machine, 카운터 등 전기 회로에 대해 많은 것을 공부하고 배울 수 있어서 정말 뜻깊은 시간이었습니다.

좋은 강의 감사합니다! :D

~~ Happy and Great for your Teaching, Thank You. ~~