



Simulation Assignment #2



과목명: 디지털공학

교수: 김우일

학과: 컴퓨터공학부

학번: 202201479

이름: 박지원

제출일: 2023년 05월 17일

목차

I . A

- i . Truth table
- ii. Design the logic circuit
- iii. Display the last two digits of my student ID

II. B

- i . 2-bit comparator
- ii. 3-to-8 line decoder

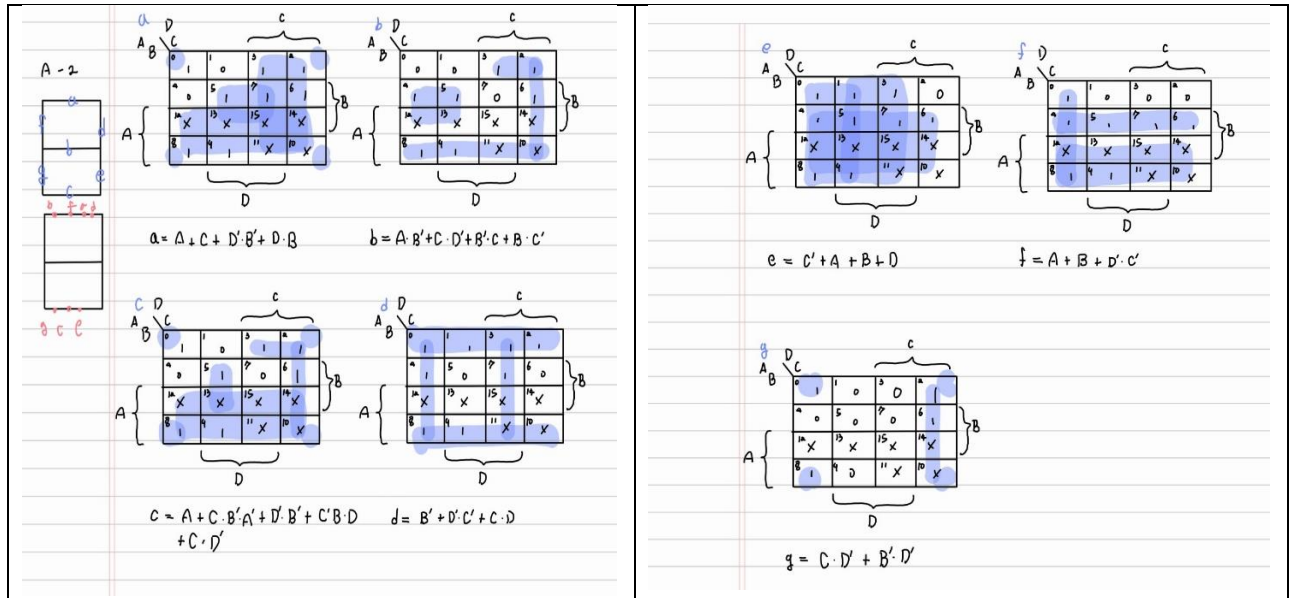
III. 마무리

I . A _ i . Truth table

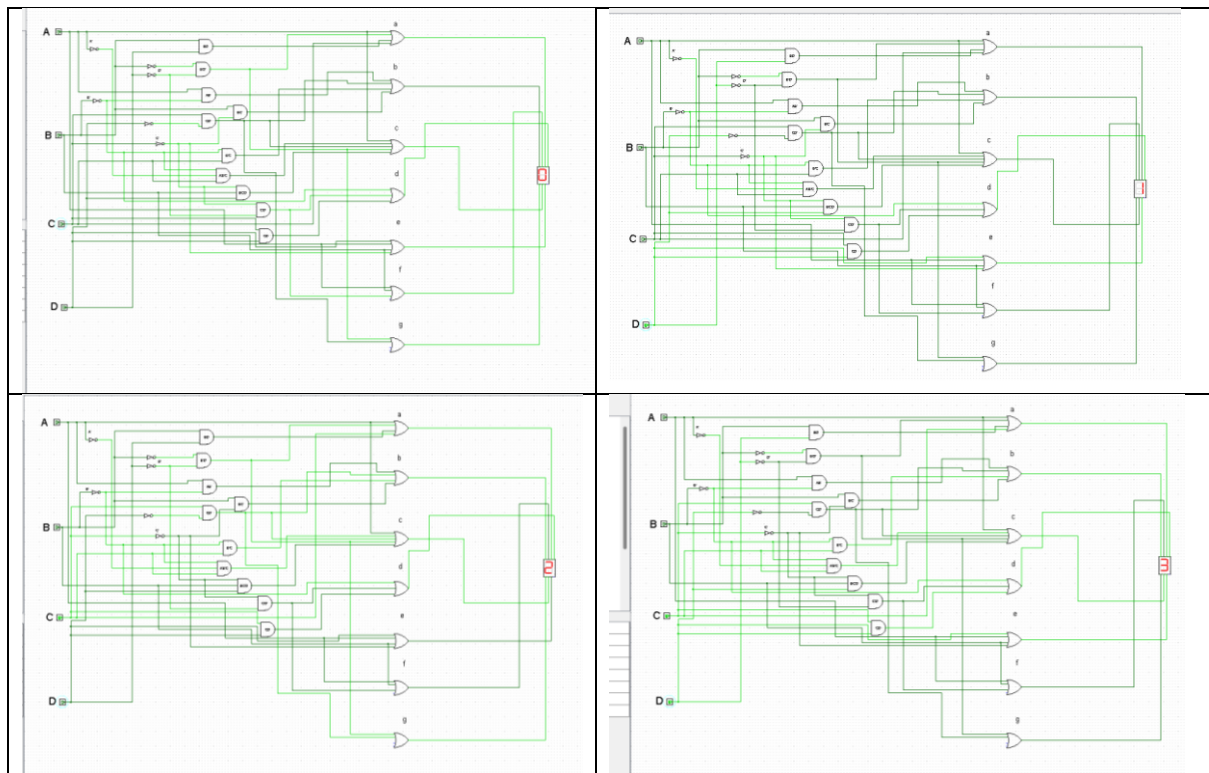
A-1	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	0	1	1	1	1	1
1	0	0	0	1	0	0	0	1	1	0	0
2	0	0	1	0	1	1	1	1	0	0	1
3	0	0	1	1	1	1	1	1	1	0	0
4	0	1	0	0	0	1	0	1	1	1	0
5	0	1	0	1	1	1	1	0	1	1	0
6	0	1	1	0	1	1	0	1	1	1	1
7	0	1	1	1	1	0	0	1	1	1	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	1	1	0

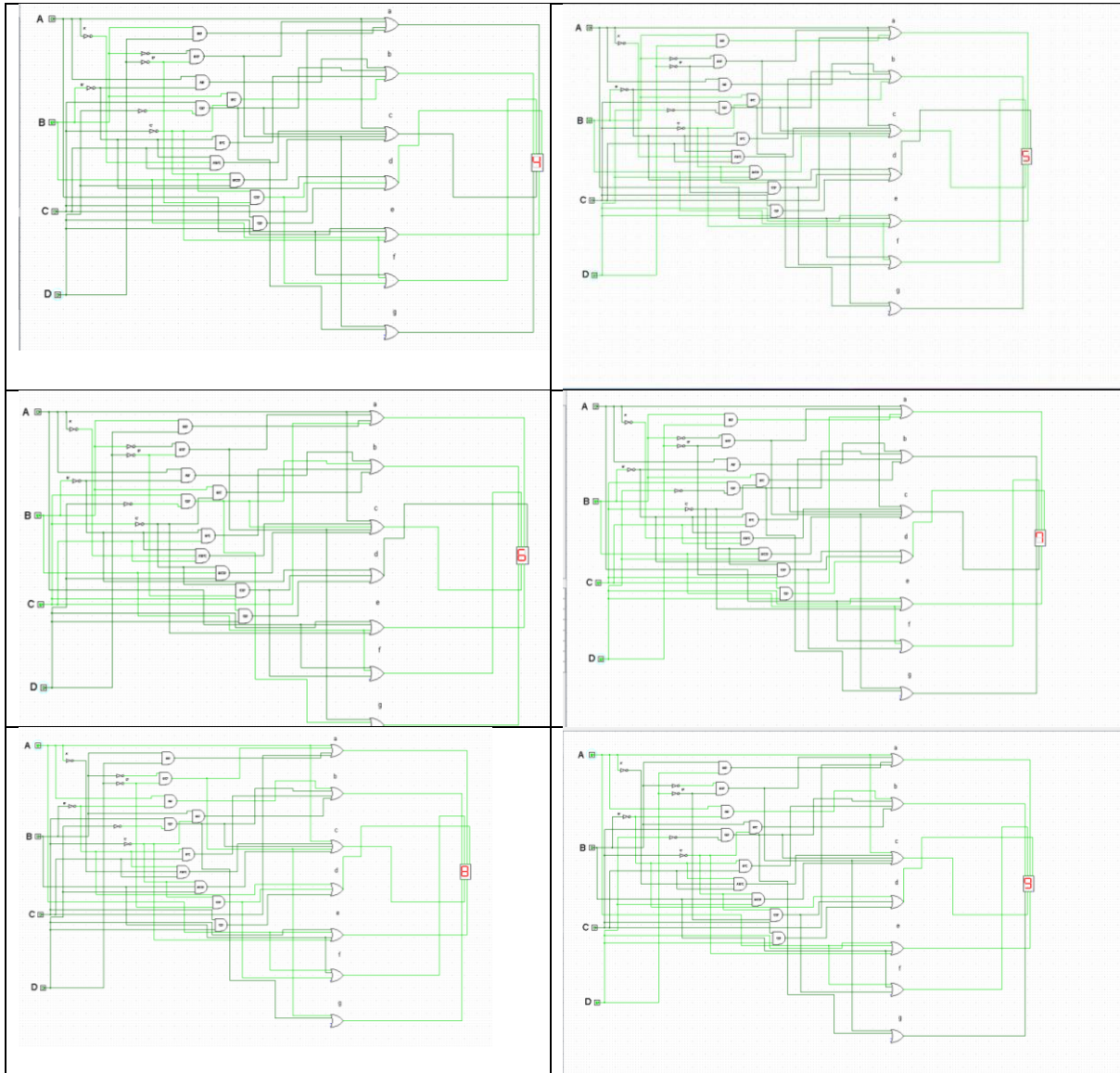
위에 표처럼 왼쪽에는 0 에서 9 까지를 BCD 코드로 나타내 주었고, 숫자에 맞게 켜져야 하는 부분에 1 을, 켜지지 않아야 하는 부분에 0 을 써주었습니다.

I . A ii. Design the logic circuit



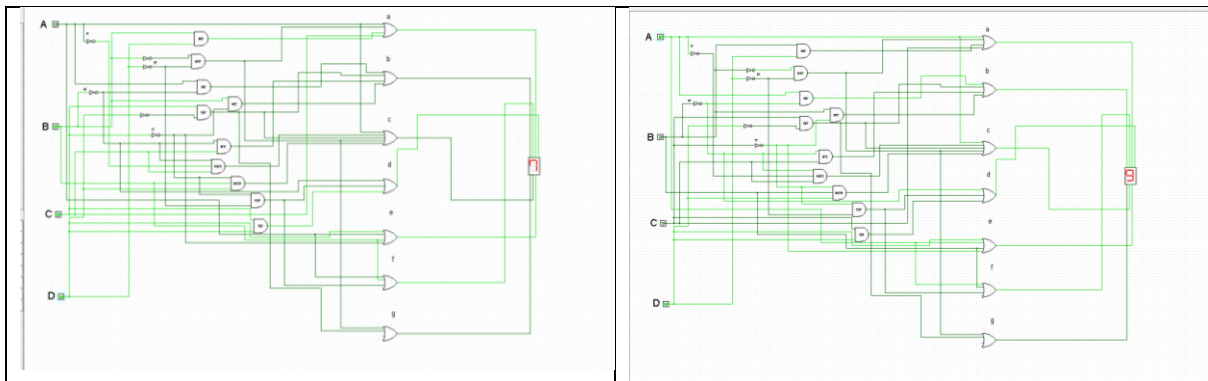
Logism을 이용하여 논리회로를 작성하기전에 위의 두 사진처럼 k map을 사용하여 각각의 a,b,c,d,e,f,g를 A,B,C,D에 대한 논리식으로 간단하게 정리해줬습니다. 그래서 이 식들을 토대로 논리회로를 아래 사진과 같이 작성해줬습니다.





K map을 이용하여 정리한 식을 통해 logism 프로그램으로 논리회로를 작성하여 위와 같이 0에서 9까지 결과가 제대로 나온 것을 확인할 수 있습니다.

I . A_ iii. Display the last two digits of my student ID



위의 논리회로를 이용하여 제 학번의 뒷자리 2개의 숫자를 나타내면 위의 사진과 같습니다.

II. B_ i . 2-bit comparator

2-bit comparator 는 이진수의 크기를 비교하여 결과를 나타내야 하므로, 아래 사진과 같이 True table 을 작성했습니다.

B-1						
A	B	C	D	$num1 > num2$	$num1 = num2$	$num1 < num2$
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

True table 에 대해 설명하자면, 왼쪽에는 A,B,C,D 를 두어, A, B 를 num1 로, C, D 도 num2 로 본다고 가정했습니다. 그래서 만일 A=0, B=0 그리고 C=0, D=1 인 경우에는 num2 가 더 크므로 num1 < num2 가 1 이 되도록 표를 작성하였습니다. 위의 표를 토대로 논리 회로를 그리기 위한 k map 을 아래와 같이 작성했습니다.

num1 > num2

		<i>C</i>			
		<i>D</i>	<i>C</i>	<i>B</i>	<i>A</i>
A	B	0	1	3	4
		5	6	7	8
		9	10	11	12
		13	14	15	16

D

num1 > num2

$$= C' \cdot A + B \cdot D' + D' \cdot A \cdot B$$

num1 = num2

		<i>C</i>			
		<i>D</i>	<i>C</i>	<i>B</i>	<i>A</i>
A	B	0	1	3	4
		5	6	7	8
		9	10	11	12
		13	14	15	16

D

num1 = num2

$$= B' \cdot A' \cdot C' \cdot D' + A' \cdot B \cdot C' \cdot D + A \cdot B \cdot C \cdot D + A \cdot B' \cdot C \cdot D'$$

num1 < num2

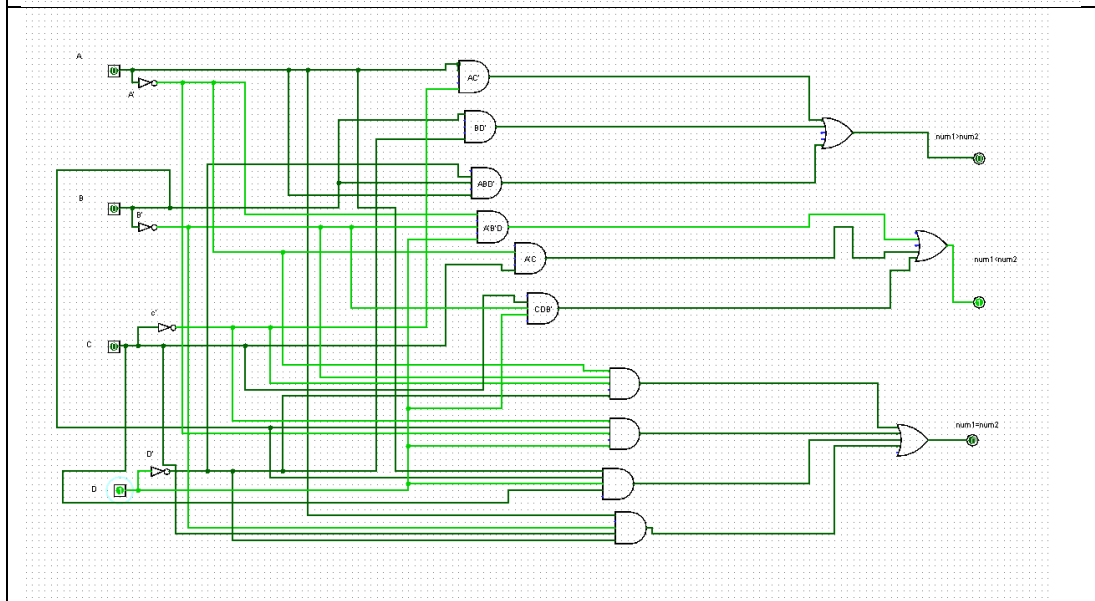
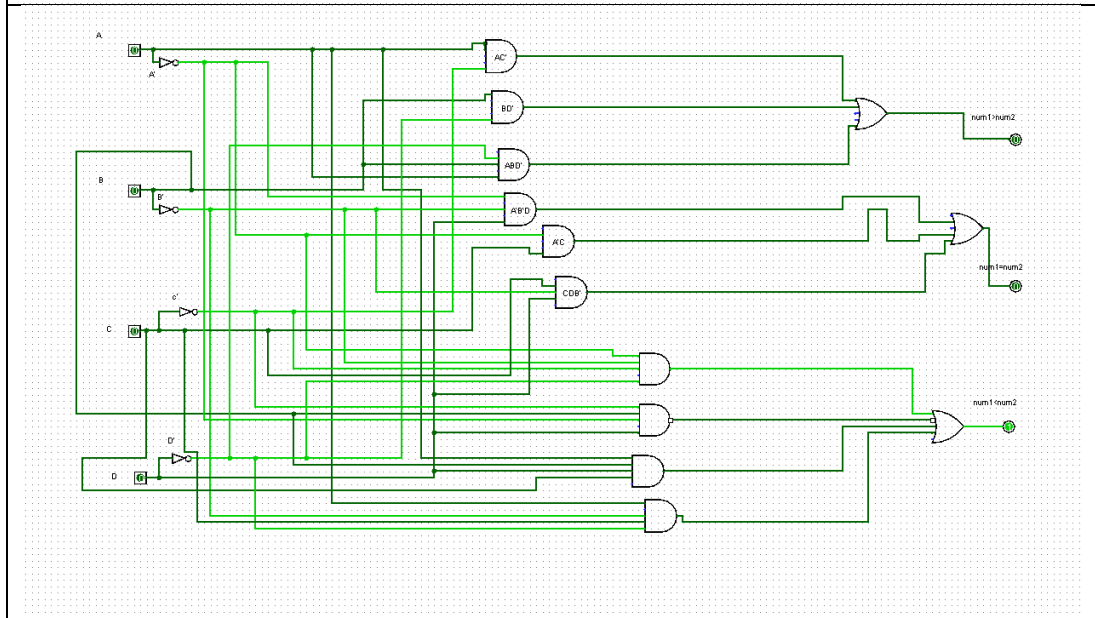
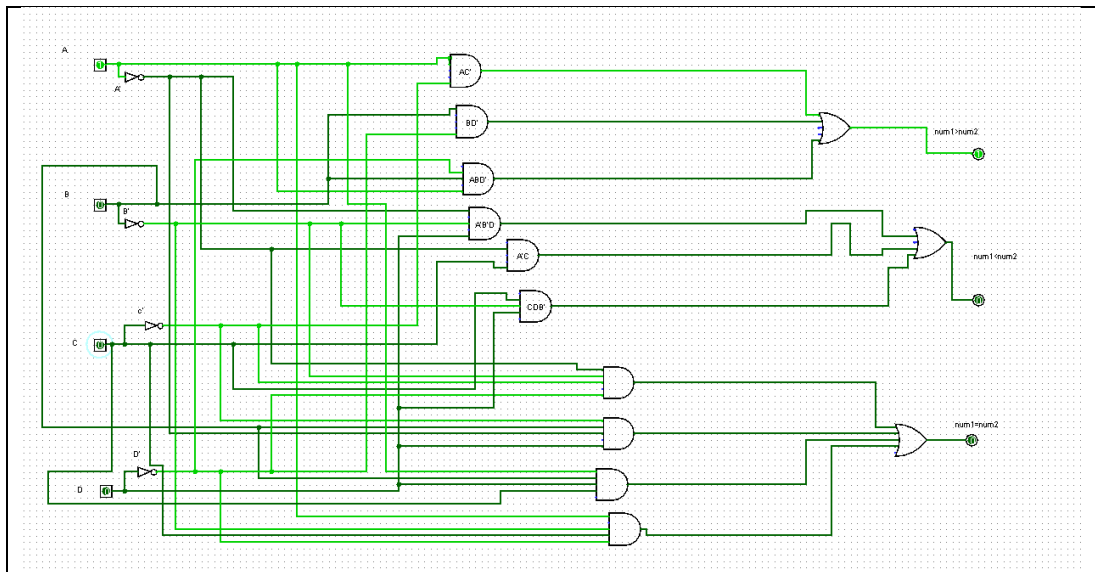
		<i>C</i>			
		<i>D</i>	<i>C</i>	<i>B</i>	<i>A</i>
A	B	0	1	3	4
		5	6	7	8
		9	10	11	12
		13	14	15	16

D

num1 < num2

$$= B' \cdot A' \cdot D + C \cdot A' + C \cdot D \cdot B'$$

위와 같이 k map 을 작성하여 논리식을 정리하고 아래와 같이 logism 에서 직접 논리회로를 작성하여 결과가 제대로 나옴을 확인할 수 있습니다.

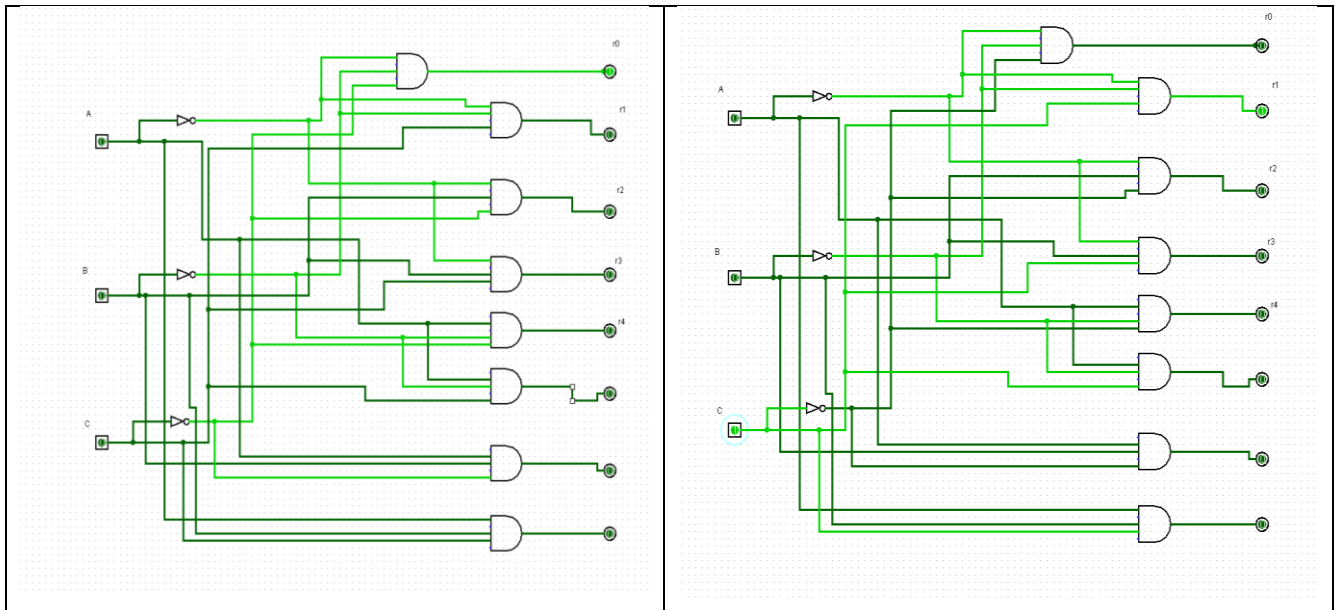


II. B_ii. 3-to-8 line decoder

3-to-8 line decoder는 3개의 입력으로 8개의 결과가 나와야 하므로 True table을 아래와 같이 작성할 수 있습니다.

	A	B	C	r_0	r_1	r_2	r_3	r_4	r_5	r_6	r_7
	0	0	0	1	0	0	0	0	0	0	0
	0	0	1	0	1	0	0	0	0	0	0
	0	1	0	0	0	1	0	0	0	0	0
	0	1	1	0	0	0	1	0	0	0	0
	1	0	0	0	0	0	0	1	0	0	0
	1	0	1	0	0	0	0	0	1	0	0
	1	1	0	0	0	0	0	0	0	1	0
	1	1	1	0	0	0	0	0	0	0	1

위의 식은 k map을 통해 나타내지 않아도 간단하게 바로 논리식을 도출해낼 수 있습니다. 예를 들어 $A = 0, B = 0, C = 0$ 의 경우, $A \cdot B \cdot C = r_0$ 라고 생각할 수 있습니다. 따라서 위의 표를 보고 바로 논리 회로를 작성하면 아래와 같은 결과를 확인 할 수 있습니다.



Ⅲ. 마무리

Logism을 처음 사용해보았는데, 시험기간에 손으로 논리 회로를 하나 하나 그렸던 기억이 납니다. 그때는 선도 구불구불하게 그려져서 고생했습니다. 하지만 이번에 프로그램을 통해 그리니 선도 반듯하고, 보기에든 편하게 그려져서 굉장히 편리하다는 생각을 했습니다. 수업시간에 배웠던 True table, k map등을 이용해 논리식을 도출해내고, 직접 회로를 그리니 그 개념들에 대해 더 이해하기 수월했습니다. 논리 회로를 그리는데 처음에는 구글에도 검색해보고, 시행착오를 겪었지만 익숙해지니 금방 그릴 수 있어 매우 흥미로웠습니다. 첫 번째 A 문제를 풀다가 다른 숫자는 다 잘 나오는데 처음에 6이 제대로 출력되지 않아 당황했는데, True table과 k map이 일치하지 않음을 알고 다시 수정했습니다. 새로운 프로그램을 접하고, 직접 시행착오를 겪어 문제를 해결할 수 있던 기회가 되어 의미가 깊었습니다.