

5주차 결과보고서

전공: 컴퓨터공학

학년: 2학년

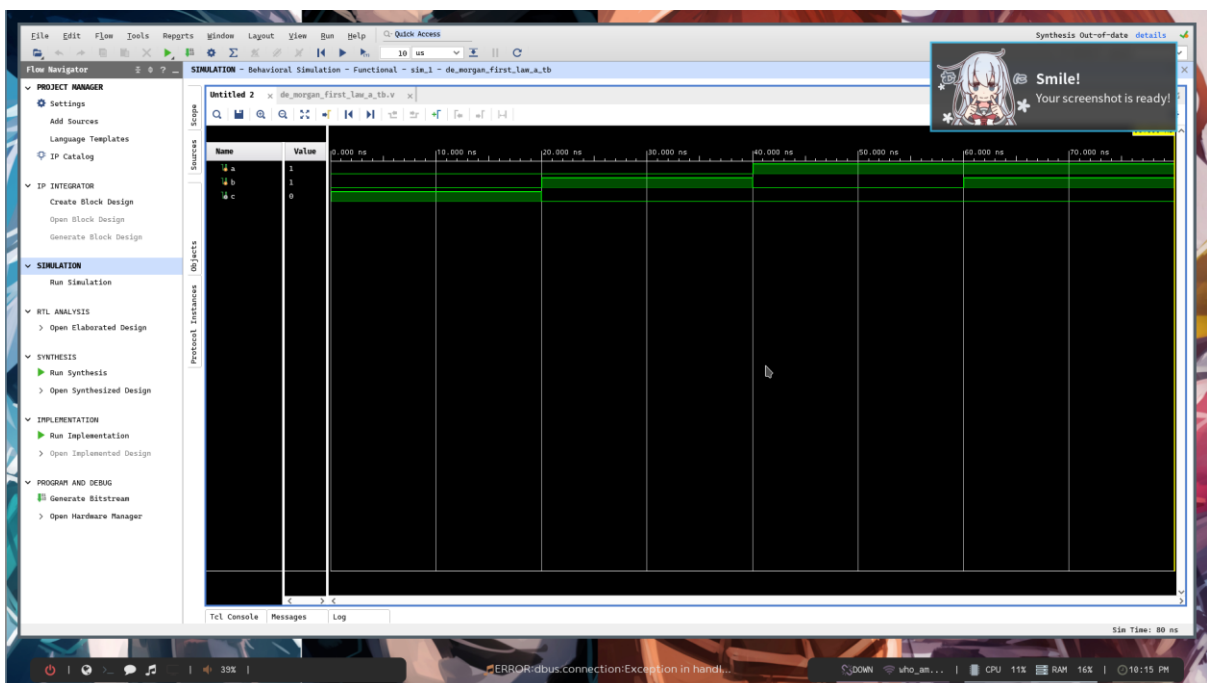
학번: 20191629

이름: 이주현

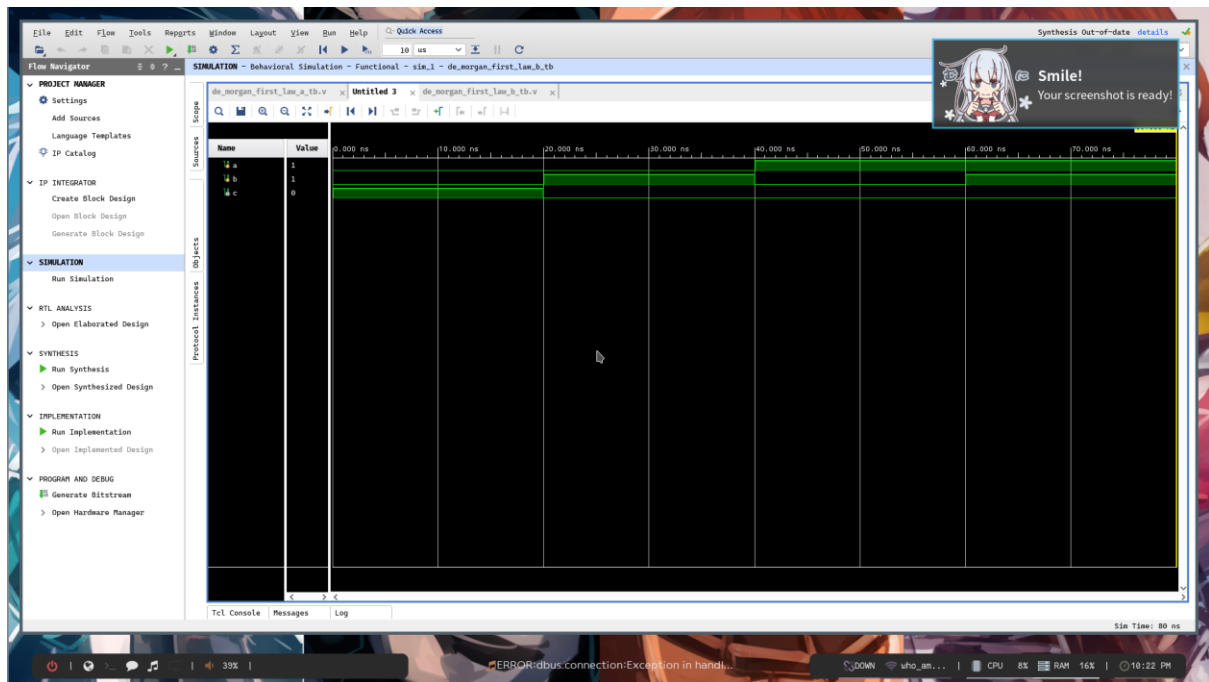
1. De Morgan의 제1, 제2법칙의 simulation 결과 및 과정에 대해서 설명하시오. (NAND, NOR과 비교 포함)

드모르간의 제1법칙은 기본적으로 다음 논리 등식이 성립한다는 것을 나타낸다.

$$\neg(A + B) = \neg A \cdot \neg B$$



드모르간의 제1법칙의 좌변을 시뮬레이션한 모습

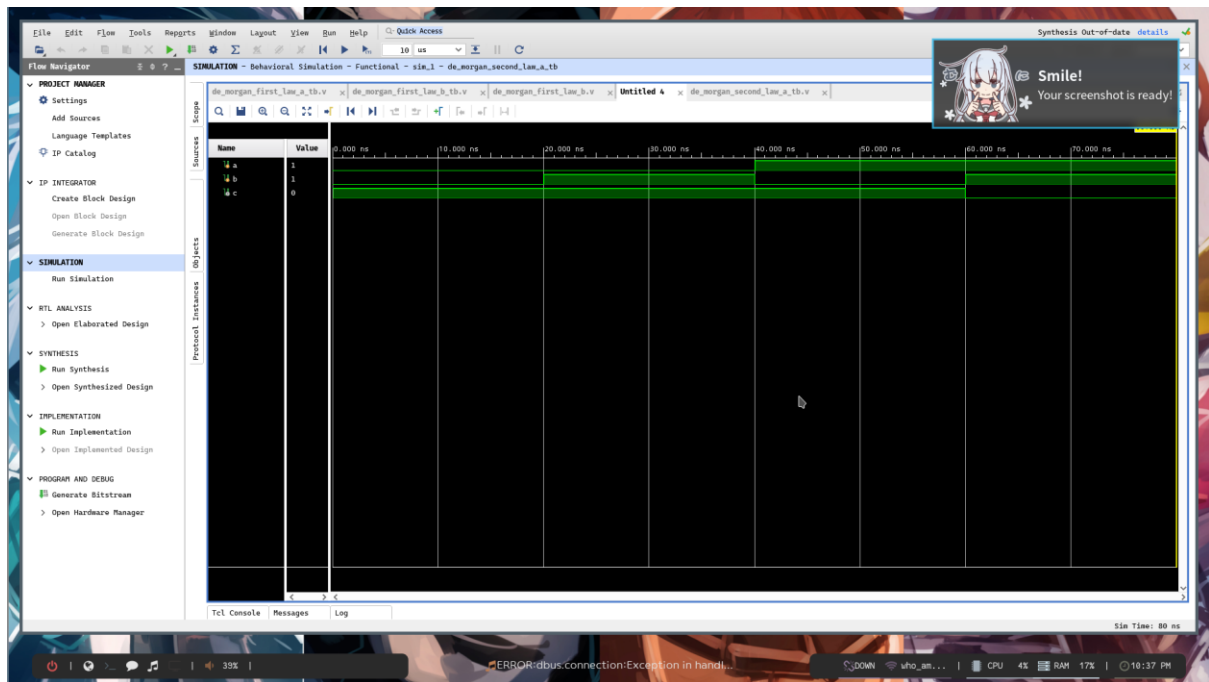


드모르간의 제1법칙의 우변을 시뮬레이션한 모습

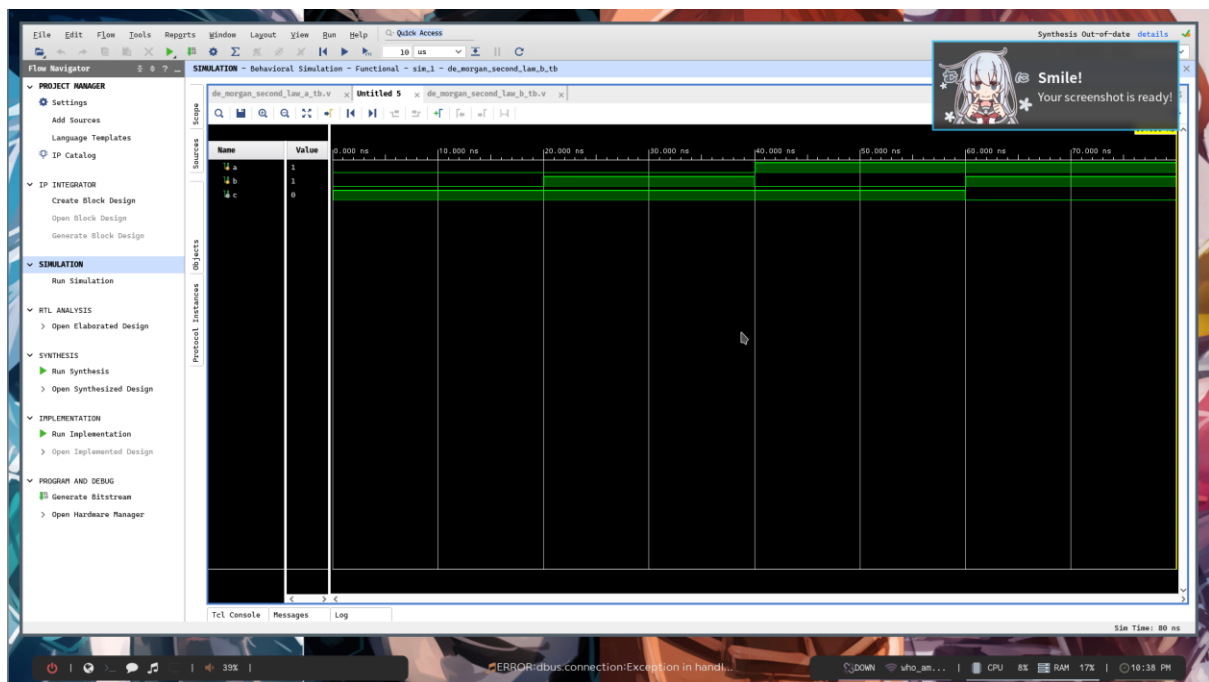
먼저 드모르간의 제1법칙의 좌변은 A와 B, 두 입력을 받아 논리합으로 묶은 뒤, 인버터를 사용해서 결과를 반전시켰다. 우변의 경우는 A와 B 각 입력을 인버터를 이용해 반전시킨 뒤, 논리곱 연산을 이용하여 최종 결과를 산출하였다. 두 시뮬레이션 결과 모두 완전히 동일한 모양을 가지고 있으므로 드모르간의 제1법칙이 성립한다는 것을 알 수 있다.

반면 드모르간의 제2법칙의 등식은 제1법칙의 등식과 비슷하지만 논리합이 논리곱으로, 논리곱이 논리합으로 전환된 모양을 가진다.

$$\neg(A + B) = \neg A \cdot \neg B$$



드모르간의 제2법칙의 좌변을 시뮬레이션한 모습



드모르간의 제2법칙의 우변을 시뮬레이션한 모습

드모르간의 제2법칙의 좌변은 A와 B, 두 입력을 받아 논리곱으로 묶은 뒤, 인버터를 사용해서 결과를 반전시켰다. 우변의 경우는 A와 B 각 입력을 인버터를 이용해 반전시킨 뒤, 논리합 연산을 이용하여 최종 결과를 산출하였다. 두 시뮬레이션 결과 모두 완전히 동일한 모양을 가지고 있으므로 드모르간의 제2법칙 역시 성립한다는 것을 알 수 있다.

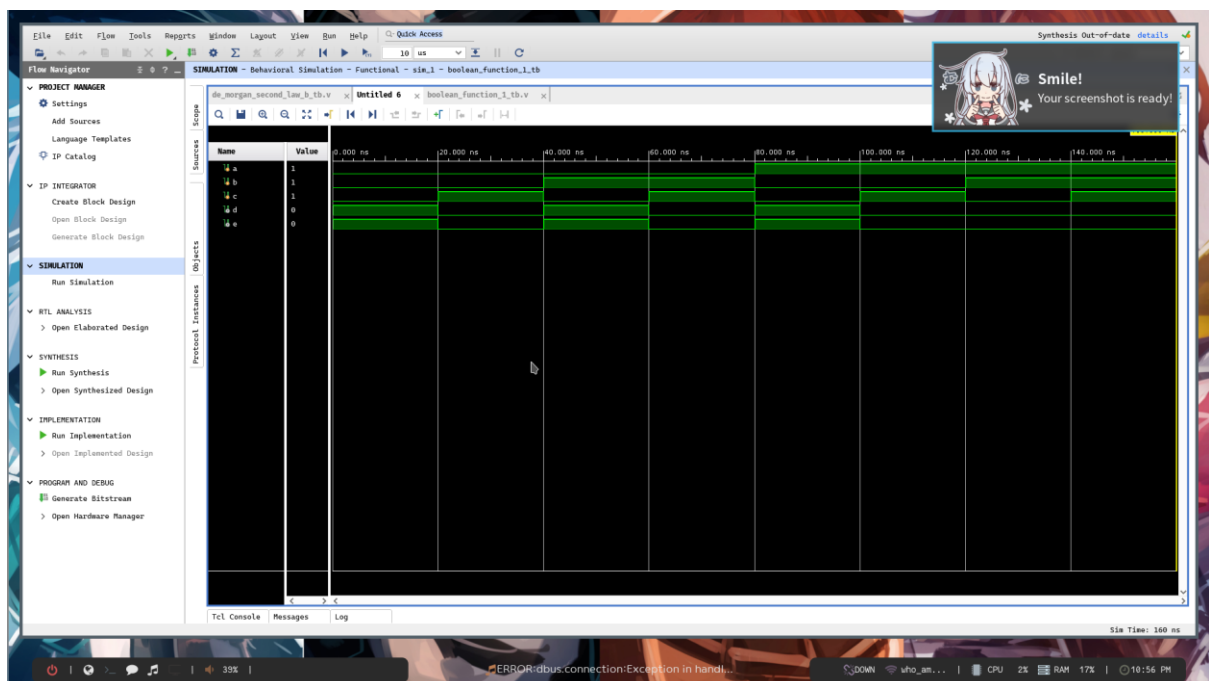
위 시뮬레이션 결과로부터 다음과 같은 진리표를 도출할 수 있다.

A	B	A B	A & B	~A	~B	~(A B)	~A & ~B	~(A & B)	~A ~B
T	T	T	T	F	F	F	F	F	F
T	F	T	F	F	T	F	F	T	T
F	T	T	F	T	F	F	F	T	T
F	F	F	F	T	T	T	T	T	T

위 진리표를 통하여 드모르간의 제1법칙과 제2법칙이 모두 성립함을 다시 한번 확인할 수 있다. 이 사실을 이용하면 NAND 게이트를 OR 게이트와 NOT 인버터, NOR 게이트를 AND 게이트와 NOT 인버터만을 사용해서 나타낼 수 있는 방법이 생긴다. 결과적으로, 드모르간의 제1법칙과 제2법칙을 한 문장으로 요약하면 다음과 같이 말할 수 있다.

논리곱과 논리합의 역은 각각 역의 논리합과 역의 논리곱과 같다.

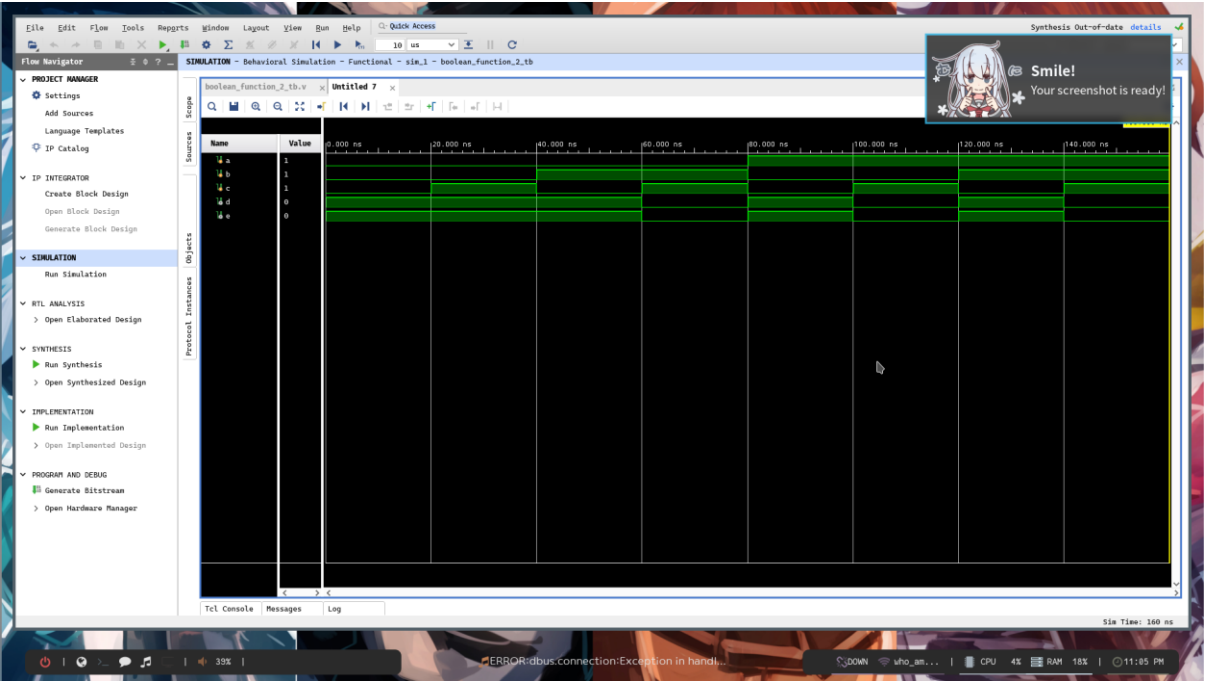
2. $(A' + B') \cdot C' = ((A \cdot B) + C)'$ 의 simulation 결과 및 과정에 대해서 설명하시오. (+ 및 ·의 위치를 바꾼 모양도 포함)



$(A' + B') \cdot C' = ((A \cdot B) + C)'$ 를 시뮬레이션한 모양. d가 좌변이고 e가 우변의 값이다.

위 시뮬레이션 그래프에서 볼 수 있듯이, 좌변과 우변의 결과값이 완전히 동일한 모양을 가지고 있다. 이 시뮬레이션 결과를 진리표로 그리면 다음과 같다.

Input A	Input B	Input C	Output D	Output E
T	T	T	F	F
T	T	F	F	F
T	F	T	F	F
T	F	F	T	T
F	T	T	F	F
F	T	F	T	T
F	F	T	F	F
F	F	F	T	T



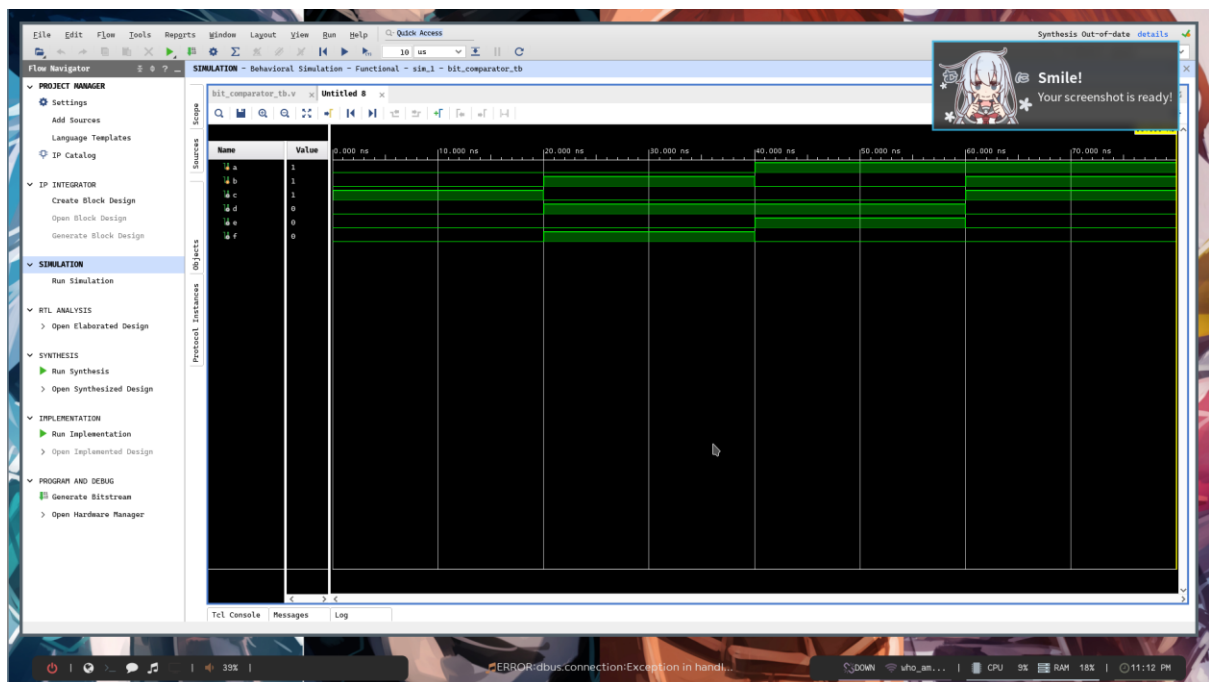
$(A' \cdot B') + C' = ((A + B) \cdot C)'$ 를 시뮬레이션한 모양. d가 좌변이고 e가 우변의 값이다.

위 식에서 논리곱과 논리합을 바꾼 모양인 $(A' \cdot B') + C' = ((A + B) \cdot C)'$ 역시 좌변의 값과 우변의 값이 서로 완전히 동일한 것을 알 수 있다. 이 시뮬레이션도 진리표로 그려보면 다음과 같다.

Input A	Input B	Input C	Output D	Output E
---------	---------	---------	----------	----------

T	T	T	F	F
T	T	F	T	T
T	F	T	F	F
T	F	F	T	T
F	T	T	F	F
F	T	F	T	T
F	F	T	T	T
F	F	F	T	T

3. 1-bit 비교기의 simulation 결과 및 과정에 대해서 설명하시오.



1-bit 비교기를 시뮬레이션한 모습. c는 =, d는 ≠, e는 >, f는 < 연산의 결과이다.

시뮬레이션 결과로부터 도출한 1-bit 비교기의 진리표는 다음과 같다.

A	B	$A = B$	$A \neq B$	$A > B$	$A < B$
T	T	T	F	F	F
T	F	F	T	T	F
F	T	F	T	F	T

F	F	T	F	F	F
---	---	---	---	---	---

이로부터 각 경우에 대한 논리식을 세울 수가 있다.

- $A = B \Rightarrow (A \cdot B) + (\neg A \cdot \neg B) \Rightarrow A \odot B$
- $A \neq B \Rightarrow (A \cdot \neg B) + (\neg A \cdot B) \Rightarrow A \oplus B$
- $A > B \Rightarrow A \cdot \neg B$
- $A < B \Rightarrow \neg A \cdot B$

1비트 "같지 않음" 연산은 A 또는 B 하나만 True일 경우에만 True를 반환한다는 점에서 XOR 연산으로 해결할 수 있고, 1비트 "같음" 연산은 반대로 A 또는 B 하나만 True일 경우에만 False를 반환한다는 점에서 XNOR 연산으로 이해할 수 있다.

4. 결과 검토 및 논의사항

첫 번째 실습에서는 드모르간의 정리를 실험적으로 확인할 수 있었다. 드모르간의 정리는 논리식을 정리할 때 매우 유용하게 활용될 수 있기 때문에 부울 대수에서 매우 중요한 정리 중 하나이다. 그 예시로, 실습 2에서의 등식이 성립함을 드모르간의 법칙을 사용해서 증명할 수도 있다.

$$\begin{aligned}
 (A' + B') \cdot C' &= ((A \cdot B) + C)' & (A' \cdot B') + C' &= ((A + B) \cdot C)' \\
 \Rightarrow (A \cdot B)' \cdot C' &= ((A \cdot B) + C)' & \Rightarrow (A + B)' + C' &= ((A + B) \cdot C)' \\
 \Rightarrow ((A \cdot B) + C)' &= ((A \cdot B) + C)' & \Rightarrow ((A + B) \cdot C)' &= ((A + B) \cdot C)'
 \end{aligned}$$

마지막 실습에서는 1-bit 비교기를 만들었는데, 진리표로부터 AND와 OR만을 사용하여 만들어낸 회로를 직접 간략화하여 하나의 연산자를 사용하도록 프로그래밍하며 논리식 간단화 작업을 연습할 수 있는 기회가 되었다.

5. 추가 이론 조사 및 작성

1비트보다 큰 비트 수의 비교기를 만들려면 고려해야 할 입력의 수가 많이 늘어나기 때문에 1-bit 비교기와 같이 눈대중으로 계산하기 힘들다. 예를 들어, 다음 진리표를 가지는 2비트 비교기를 만든다고 하자.

INPUT				OUTPUT		
A1	A0	B1	B0	A<B	A=B	A>B
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

이 회로는 총 4개의 입력 신호가 있기 때문에 16가지의 경우의 수를 고려해야 한다.
이를 지난 주차에 학습한 카르노 맵으로 풀어 보면 다음과 같다.

		B1B0			
		00	01	11	10
A1A0	00	0	0	0	0
	01	1	0	0	0
	11	1	1	0	1
	00	1	1	0	0

		A = B			
		B1B0	00	01	11
A1A0	00	1	0	0	0
	01	0	1	0	0
	11	0	0	1	0
	00	0	0	0	1

		A < B			
		B1B0	00	01	11
A1A0	00	0	1	1	1
	01	0	0	1	1
	11	0	0	0	0
	00	0	0	1	0

카르노 맵으로부터 논리식을 유도하면 다음과 같다.

- $$A = B \quad \neg A_1 \cdot \neg A_0 \cdot \neg B_1 \cdot \neg B_0 + \neg A_1 \cdot A_0 \cdot \neg B_1 \cdot B_0 + A_1 \cdot A_0 \cdot B_1 \cdot B_0 + A_1 \cdot \neg A_0 \cdot B_1 \cdot \neg B_0$$

$$\Rightarrow (A_0 B_0 + \neg A_0 \neg B_0)(A_1 B_1 + \neg A_1 \neg B_1)$$

$$\Rightarrow (A_0 \odot B_0)(A_1 \odot B_1)$$
- $$A < B \quad \neg A_1 B_1 + \neg A_0 B_1 B_0 + \neg A_1 \neg A_0 B_0$$
- $$A > B \quad A_1 \neg B_1 + A_0 \neg B_1 \neg B_0 + A_1 A_0 \neg B_0$$