5주차 결과보고서

전공: 아트앤테크놀로지 학년: 3학년 학번: 20191098 이름: 백승주

**1.**

- De-Morgan의 제 1,2법칙을 Verilog 코드로 구현하고 해당 법칙들이 성립함을 확인한다.

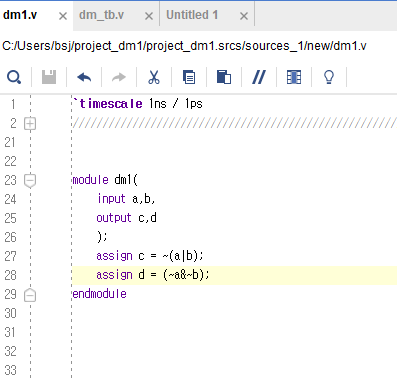
- (A’+B’)\*C’ = ((A\*B)+C)’, (A’\*B’)+C’ = ((A+B)\*C)’를 Verilog 코드로 구현하고 식이 성립함을 확인한다.

- 1 Bit 비교기를 Verilog 코드로 구현하고 그 결과를 토대로 진리표를 작성한다. 또한 FPGA를 통해 회로의 동작을 직접 확인해본다.

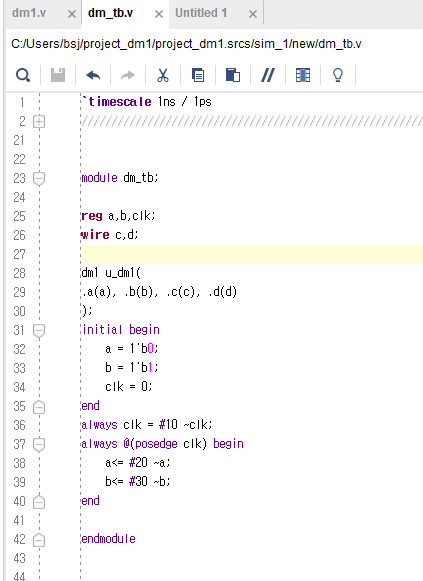
**2-1. 드 모르간 1 법칙**

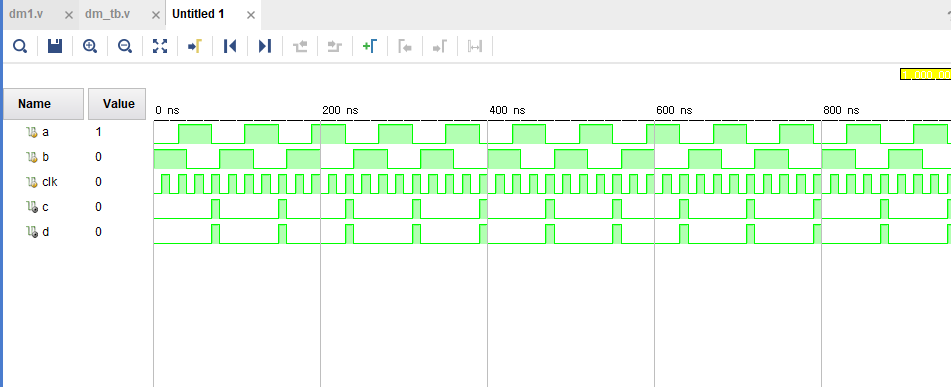
먼저 위의 두 게이트는 모두 2개의 입력을 받는 게이트다. 첫 번째 게이트는 Nor 게이트를 이용한 게이트로 A와 B를 OR 게이트의 입력으로 받고 NOT을 취해 C라는 출력 값을 출력해주는 게이트다. 두 번째 게이트는 AND 게이트를 이용한 게이트로 A와 B에 NOT을 취한 값을 AND 게이트로 받아 D라는 출력 값을 출력해주는 게이트다. 드 모르간 1법칙에 의하면 위 두 게이트의 출력 값 C, D는 같아야 한다. 이 두 게이트의 Verilog 코드는 다음과 같다.



두 게이트의 결과값을 비교하기 위해 assign문을 활용해 output c, d에 게이트의 결과값을 할당했다. 이 결과값들은 각각 드 모르간 1법칙 (A+B)’ = (A’\*B’)의 좌항과 우항에 해당하는 결과값들이다. 코드의 결과를 simulation하기 위한 testbench코드는 다음과 같다

.

이 코드에 의한 시뮬레이션 결과는 clk 값이 positive edge일 때마다 a,b의 input값들이 서로 다른 시간 간격을 두고 바뀌기 때문에 a, b의 값들에 따른 output c, d 값들의 시뮬레이션 결과는 다음과 같다.

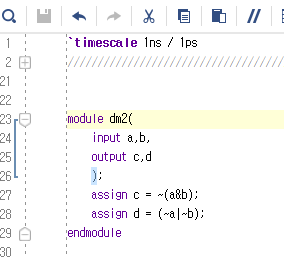


결과값 c는 NOR 게이트와 같아 input이 모두 0일 때에는 1을 결과값으로 출력하고 그 외의 경우에는 0을 출력한다. 반면에 결과값 d는 AND 게이트를 이용한 게이트였는데 시뮬레이션 결과를 보면 a, b 에 따른 출력 값이 모두 c와 같다는 것을 확인할 수 있다. 즉 드 모르간 제 1법칙이 성립한다는 것을 위 시뮬레이션 결과를 통해 확인할 수 있었다.

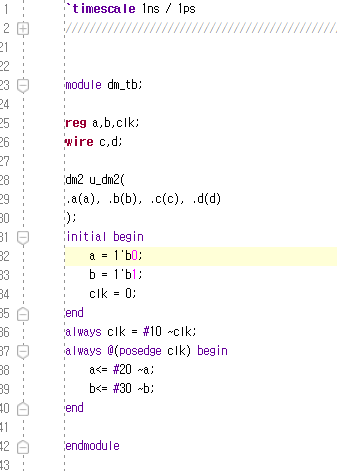
**2-2. 드 모르간 2 법칙**



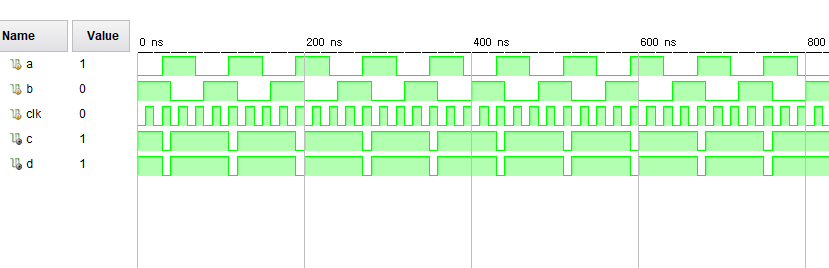
드 모르간 제 2법칙을 나타낸 위의 두 게이트는 모두 2개의 입력을 받는 게이트다. 첫 번째 게이트는 Nand 게이트를 이용한 게이트로 a와 b를 AND 게이트의 입력으로 받고 NOT을 취해 C라는 출력 값을 출력해주는 게이트다. 두 번째 게이트는 OR 게이트를 이용한 게이트로 a와 b에 NOT을 취한 값을 OR 게이트로 받아 a라는 출력 값을 출력해주는 게이트다. 드 모르간 제 2법칙에 의하면 위 두 게이트의 출력 값 c, d는 같아야 한다. 이 두 게이트의 Verilog 코드는 다음과 같다.



이 Verilog 코드에 대한 simulation 코드는 다음과 같다.



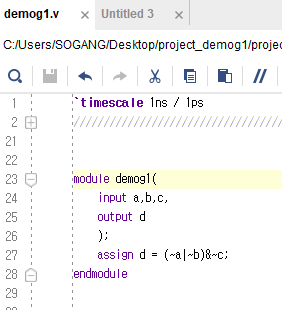
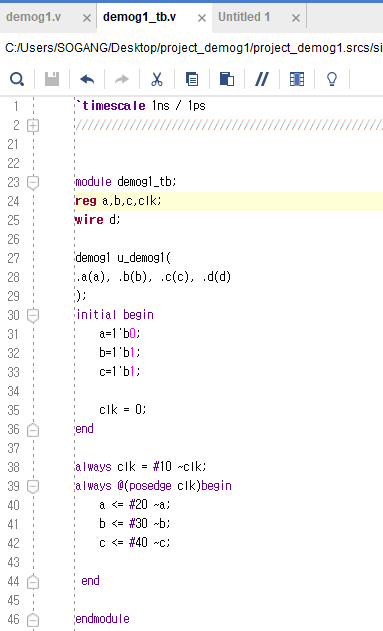
이에 따른 시뮬레이션 결과는 다음과 같다.



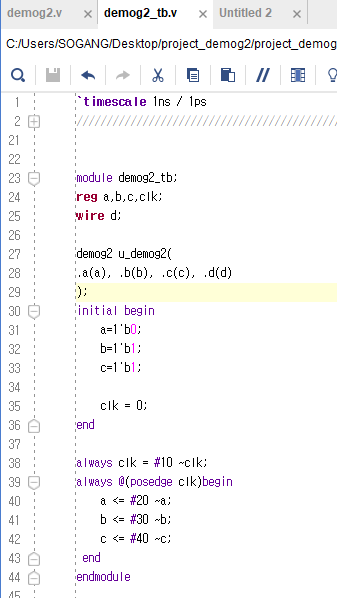
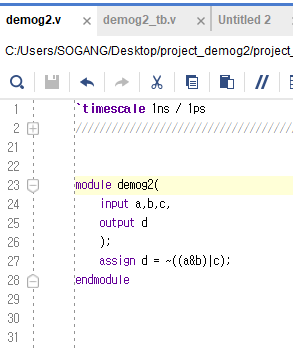
결과값 c는 Nand 게이트와 같아 input이 모두 1일 때에는 0을 결과값으로 출력하고 그 외의 경우에는 0을 출력한다. 반면에 결과값 d는 AND 게이트를 이용한 게이트였는데 시뮬레이션 결과를 보면 a, b 에 따른 출력 값이 모두 c와 같다는 것을 확인할 수 있다. 즉 드 모르간 제 2법칙이 성립한다는 것을 위 시뮬레이션 결과를 통해 확인할 수 있다.

**3-1. (A’+B’)\*C’ = ((A\*B)+C)’**

드 모르간 법칙에 의해 위의 식도 성립함을 알 수 있다. 우변의 (a\*b)를 하나의 항으로 보면 드 모르간 제 1법칙에 의해 (A\*B)’\*C’이 됨을 알 수 있고 이 식에서 (A\*B)’에서 다시 드 모르간 제 2법칙을 이용하면 A’+B’이 된다. 즉 좌변과 우변의 식이 같음을 알 수 있다. 이 수식이 성립함을 Verilog 코드를 통해서도 알 수 있는데 먼저 좌변의 식에 대한 Verilog 코드와 testbench 코드를 구현해보면 다음과 같다.

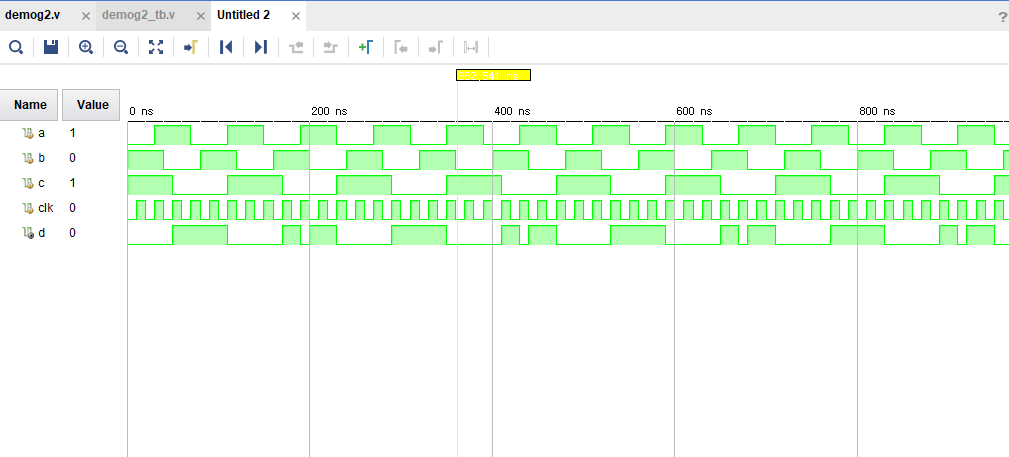
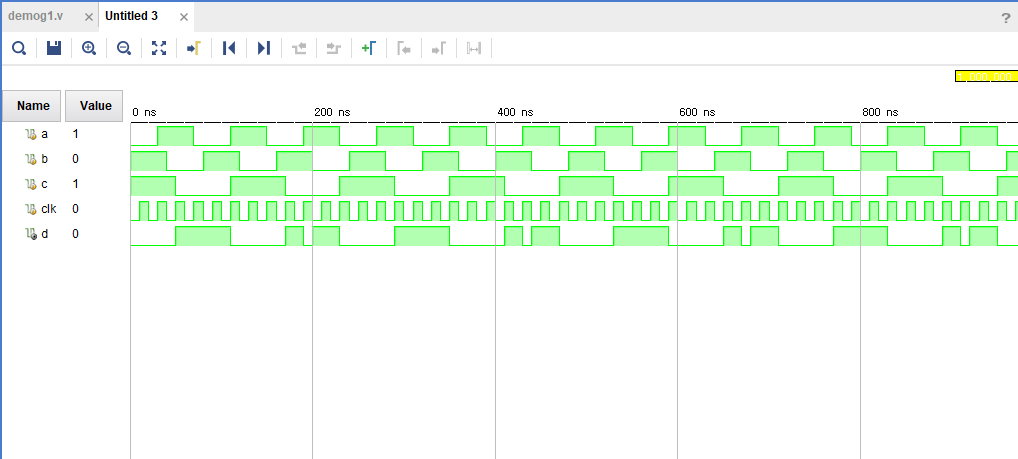
 

Output d를 이용해 좌변의 식의 결과값을 받아줬고 시뮬레이션 결과는 clk 값이 positive edge일 때마다 a,b,c의 input값들이 서로 다른 시간 간격을 두고 바뀌게 해 모든 경우의 수에 따른 좌변의 값을 알 수 있도록 했다. 마찬가지로 우변의 식에 대한 Verilog 코드와 testbench 코드는 다음과 같다.



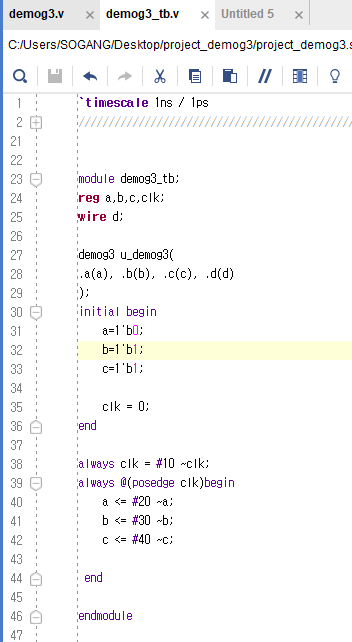
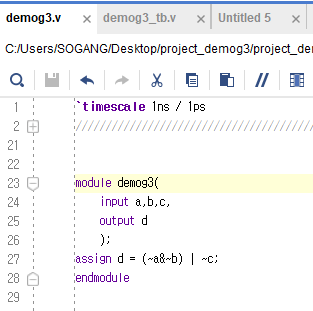
Output d를 설정해 우변의 식의 결과값을 받아줬고 시뮬레이션 결과는 clk 값이 positive edge일 때마다 a,b,c의 input값들이 서로 다른 시간 간격을 두고 바뀌게 해 모든 경우의 수에 따른 우변의 값을 알 수 있도록 했다.

좌변의 식과 우변식의 시뮬레이션 결과들은 각각 다음과 같다.

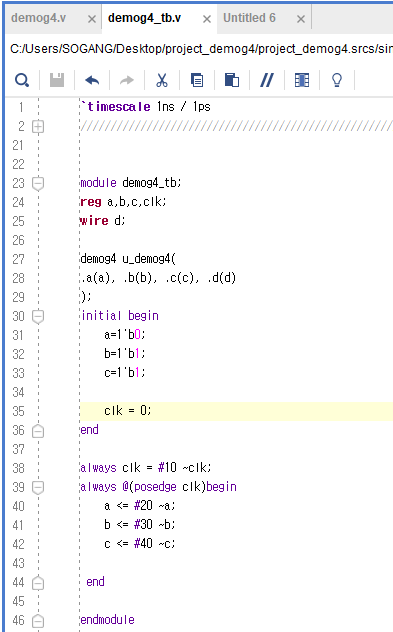
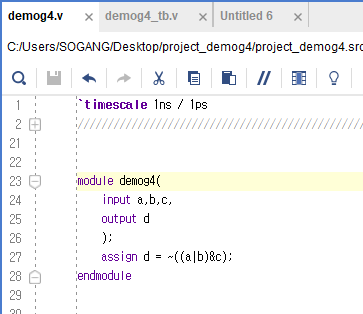


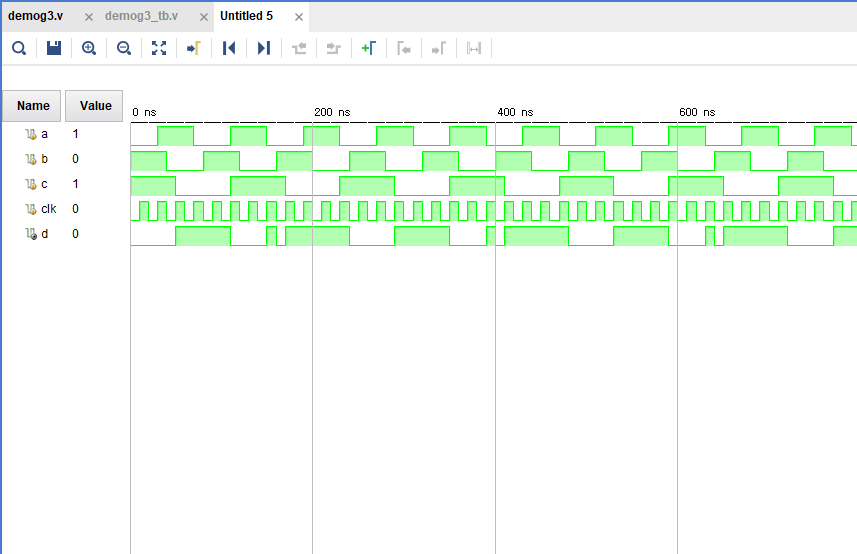
Simulation 결과에서 확인할 수 있듯이 a,b,c에 따른 결과값인 d의 값이 동일하게 나타나는 것을 알 수 있다. 즉 (A’+B’)\*C’ = ((A\*B)+C)’식은 성립함을 알 수 있다.  
**3-2. (A’\*B’)+C’ = ((A+B)\*C)’**

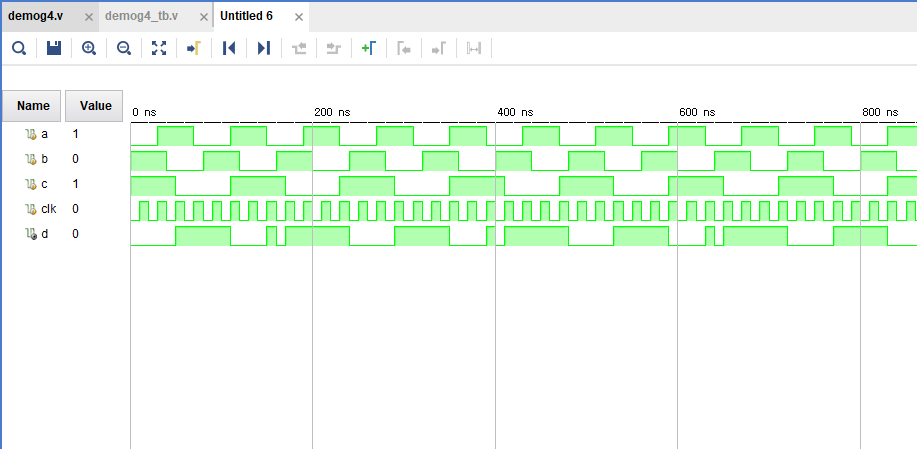
위 식 역시 드 모르간 법칙에 의해 성립함을 먼저 알 수 있다. 우변의 식에서 (A\*B)를 하나의 항으로 보고 드 모르간 제 2법칙을 적용하면 (A+B)’+C’가 되고 (A+B)’에서 다시 드 모르간 제 1법칙을 적용하면 A’+B’가 된다. 즉 드 모르간 법칙을 사용하여 우변을 정리하면 좌변의 식과 일치하는 것을 알 수 있다. 또한 이 식이 일치함을 Verilog 코드를 통해 알아보자. 먼저 좌변의 식에 대해 verilog코드와 simulation 코드를 짜면 다음과 같다.



마찬가지로 우변의 식에 대해 코드를 짜면 다음과 같다.



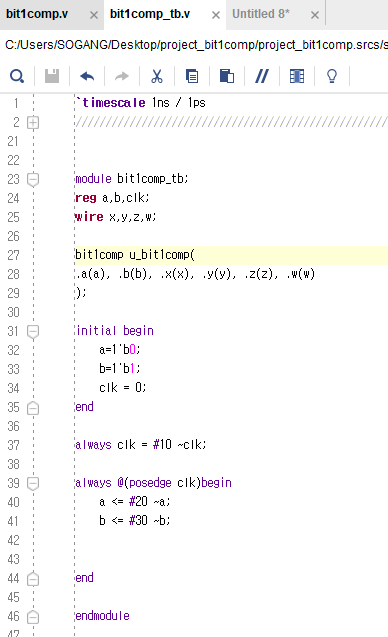
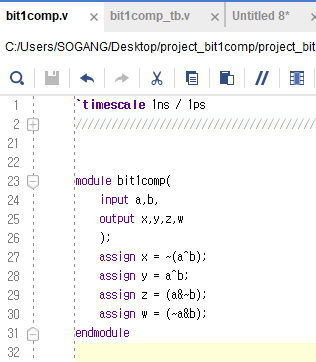
이 코드들에 의한 좌변과 우변의 시뮬레이션 결과는 다음과 같다.

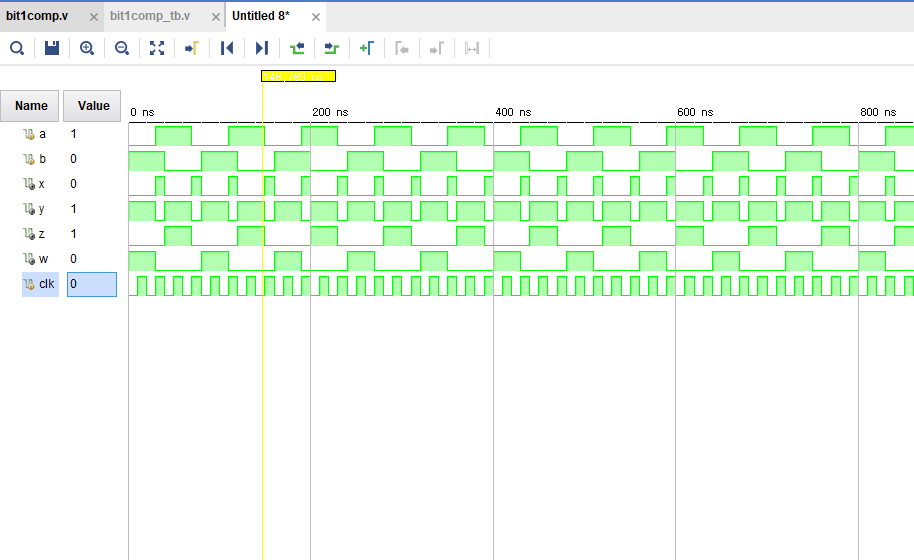


Simulation 결과에서 확인할 수 있듯이 a,b,c에 따른 결과값인 d의 값이 동일하게 나타나는 것을 알 수 있다. 즉 (A’\*B’)+C’ = ((A+B)\*C)’식은 성립함을 알 수 있다.

**4. 1 bit 비교기**

1 bit 비교기는 2개의 input을 가지고 4개의 output을 출력해주는 회로다. 구체적으로는 입력값 a, b에 대한 비교연산들에 대한 결과값을 출력해주는 회로로 비교연산의 식이 성립하면 1을 출력하고 성립하지 않으면 0을 출력한다. 그 비교연산으로는 a==b, a!=b, a>b, a<b, 4가지가 있다. 간단히 예를 들어 보면 a, b 가 모두 1일 경우 비교연산 a==b 이 성립하므로 1 bit adder는 1,0,0,0 을 출력한다. 이에 대한 Verilog 코드를 구현하면 다음과 같다.



a==b를 나타낼 때에는 두 입력 값 중 하나만 1일 때에 1을 출력하는 XOR 식에다 NOT을 취하면 동일한 결과를 얻을 수 있다. a!=b를 나타내기 위해서는 a==b와 반대이므로 xor을 그대로 써서 표현이 가능하다. a>b은 a가 1이고 b가 0일 때를 의미하므로 a&~b로 표현 가능하며, a<b 역시 유사하게 a~&b로 표현이 가능하다. 위 식에 대한 시뮬레이션 결과와 그에 따른 진리표는 다음과 같다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input a** | **Input b** | **a==b** | **a!=b** | **a>b** | **a<b** |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

**5.**

**5-1.** NOR 게이트인 (A+B)’와 NAND 게이트 (A\*B)’가 드 모르간 제1, 2법칙에 의해 (A’\*B’), (A’+B’) 와 동일한 출력 결과를 가지기 때문에 NOR 게이트와 NAND게이트는 2가지 방법으로 구현될 수 있었다.

**5-2.** 시뮬레이션을 통해 (A’+B’)\*C’ = ((A\*B)+C)’와 (A’\*B’)+C’ = ((A+B)\*C)’ 식이 성립함을 확인하며 복잡한 식에서도 드 모르간 법칙을 사용해 식을 정리할 수 있음을 확인했다.

**5-3.** 두 개의 인풋 a,b를 a==b, a!=b, a>b, a<b의 비교연산을 논리 연산자로 구현할 수 있음을 확인할 수 있었다. 또한 그러한 1bit 비교기가 어떻게 구현되는 지를 확인하고 그 시뮬레이션 결과를 확인해 진리표를 작성했다.

**6.** 디코더 – 최소항 생성기라 할 수 있는 디코더는 n개의 input을 받고 2^n개의 output을 출력한다. 디코더는 input 값에 따라 하나의 output 값만 1이 되는 디코더를 high decoder라 하고 반대로 input 값에 따라 output 값 하나만 0이 되는 decorder를 low decorer라 한다. 예를 들어 3 to 8 line decoder 란 말은 3개의 input 값과 8개의 최소항을 갖는 디코더라는 뜻이다.

인코더 – 디코더의 반대 개념으로 2^n개의 입력이 주어지면 n개의 output 을 출력한다.

멀티플렉서 – input이 여러 개 주어지면 그 중 하나의 input만 골라 출력한다. 따라서 input의 개수는 2^n개가 되며 그 중 어떤 input을 입력할지 결정하는 select bit가 존재한다. Select bit는 개수가 n개가 된다. 간단히 예를 들어보 a,b,c,d 네 개의 input이 존재하고 select bit로 그 중 원하는 input을 고른다하면 00으로 설정해 첫 번째 input인 a를 고르거나 11으로 설정해 4번째 input인 d를 고를 수 있다. 멀티플렉서 역시 반대의 연산을 하는 디멀티플렉서가 존재한다.