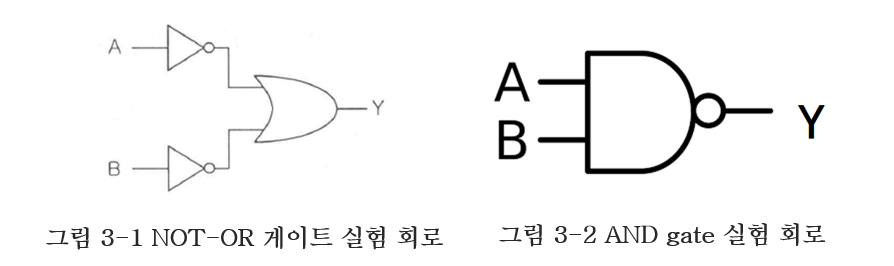
|  |  |
| --- | --- |
| **실험 결과 보고서**  **(4주차)** | 학 번 : 122201856  이 름 : 김다영  제출일 : 2021.10.03  분 반 : 002  실험조 : 개인 |

1. **실험 제목 : Boolean Algebra, 드모르간의 정리**
2. **실험 목적**

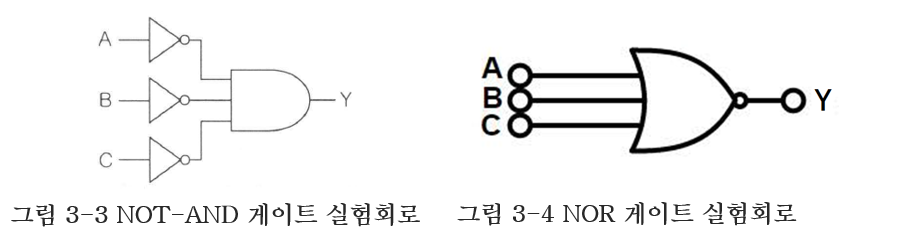
* Boolean Algebra와 드모르간의 정리를 OrCAD Pspice를 이용해 결과를 확인해본다.
* 드 모르간 법칙을 이용하여 Boolean Algebra 변환 및 논리회로를 간소화하는 능력을 익힌다.

1. **실험준비**
2. 장비 셋팅 : 개인 노트북에 OrCAD Pspice 실행
3. 사용기구 및 부품 : OrCAD Pspice Program
4. 팀원 역할 분담 내역 : 개인별로 실험을 진행하였다.
5. **실험결과**

Simulation 해야 할 Logic gate는 아래 [그림1], [그림2]와 같다.



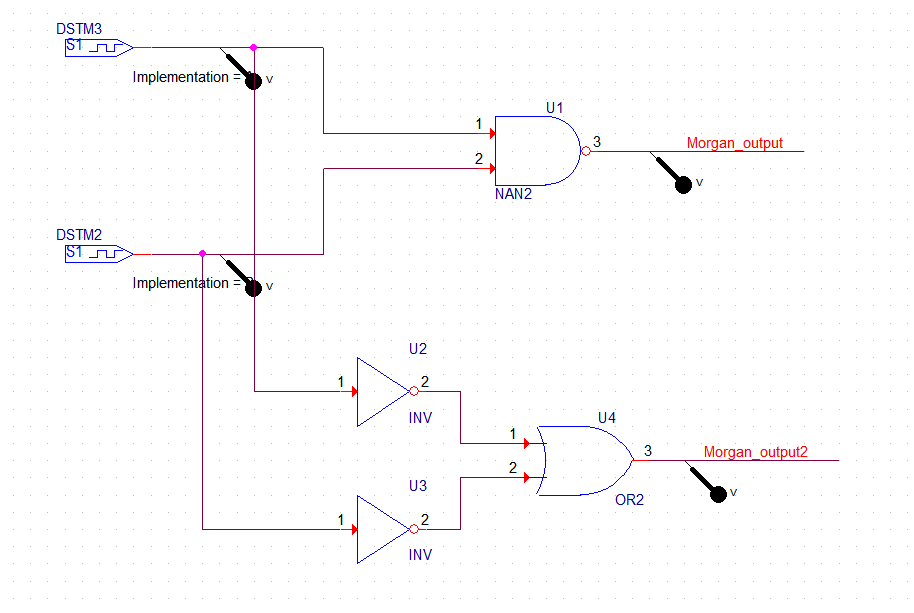
[그림 1] NOT-OR / NAND gate 실험 회로



[그림 2] NOT-AND / NOR gate 실험 회로

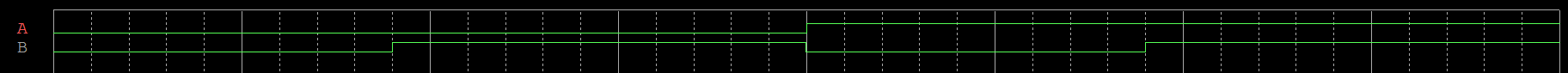
* + - * 1. **실험 1-1**

Simulation을 위해 OrCAD pspice로 [그림 1]과 같이 논리 회로를 구성하면 [그림 3]과 같은 결과를 얻을 수 있다. [그림 1]의 Logic gate는 입력이 2개이므로 입력을 A, B 두 개로 설정한다.

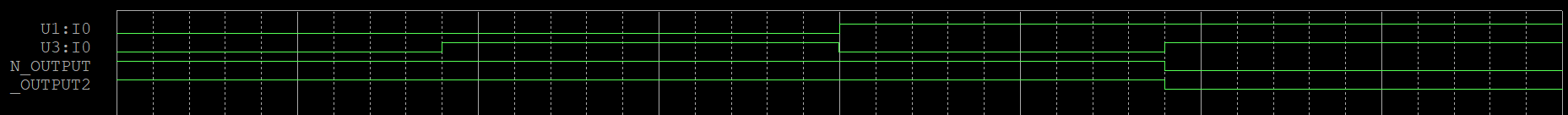


[그림 3] 실험 1-1 OrCAD 논리 회로도

논리 회로의 input을 A와 B 로 설정한 뒤 [그림 4]와 같이 A와 B의 파형을 설정한다. X축 값인 시간의 범위는 0ms부터 40ms로 설정한다. 이때 2개의 input이 존재하므로 22=4가지 경우의 입력 파형이 존재한다. 각 wire에서의 입출력 파형을 확인하기 위해 Voltage/Level Marker를 설치한다.

 [그림 4] 실험 1-1 입력 파형

이후 simulation file을 생성하고 Run To Time을 입력 신호의 시간 범위와 동일한 40ms로 설정한 뒤 RUN하면 [그림 5]와 같이 A과 B의 파형에 따른 출력 파형이 발생하는 것을 볼 수 있다.

 [그림 5] 실험 1-1 출력 파형

출력 결과를 바탕으로 실험 1-1 논리 회로도의 진리표를 작성하면 다음 [표 1]과 같다.

Morgan\_Output은 두 입력 A, B가 NAND gate에 입력되어 출력되는 결과이다. NAND gate는 입력에 0이 하나라도 존재하면 1이, 존재하지 않으면 0이 출력되는 gate이다. 따라서 0이 하나라도 들어가는 경우(⓵ A=0, B=0 / ② A=0, B=1 / ③ A=1, B=0)에는 출력이 1인 것을 확인할 수 있다. 반면에 0이 하나도 존재하지 않는 경우(④ A=1, B=1)에는 출력이 0인 것을 확인할 수 있다.

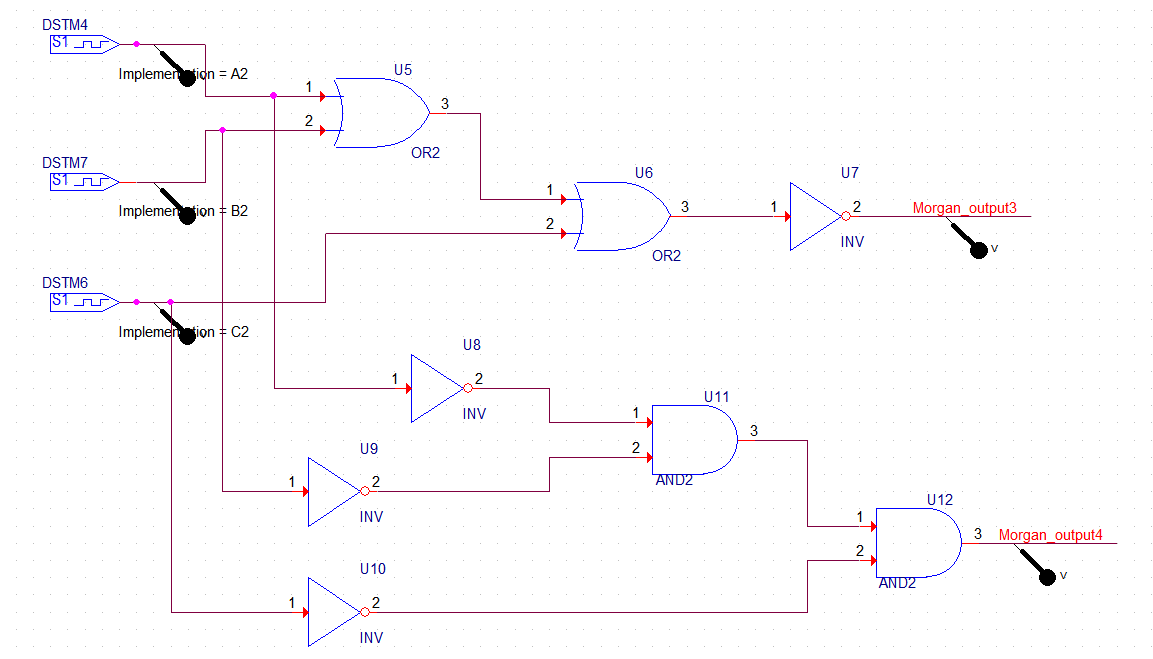
Morgan\_Output2는 두 입력 A, B가 NOT gate에 의해 반전된 값이 OR gate에 입력되어 출력되는 결과이다. OR gate는 입력에 1이 하나라도 존재하면 1이, 존재하지 않으면 0이 출력되는 gate이다. 따라서 1이 하나라도 들어가는 경우(⓵ A’=1, B’=1 / ② A’=1, B’=0 / ③ A’=0, B’=1)에는 출력이 1인 것을 확인할 수 있다. 반면에 0이 하나도 존재하지 않는 경우(④ A’=0, B’=0)에는 출력이 0인 것을 확인할 수 있다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **Y(계산)** | **AB(계산)** | **(실험)** | **(계산)** | **(계산)** | **(실험)** |
| ⓵ **0** | **0** | **1** | **0** | **1** | **1** | **1** | **1** |
| ② **0** | **1** | **1** | **0** | **1** | **1** | **0** | **1** |
| ③ **1** | **0** | **1** | **0** | **1** | **0** | **1** | **1** |
| ④ **1** | **1** | **0** | **1** | **0** | **0** | **0** | **0** |

[표 1] 실험 1-1 진리표

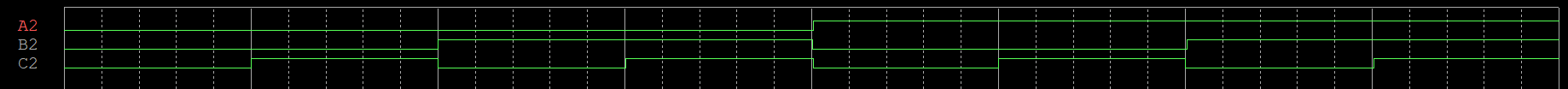
* + - * 1. **실험 1-2**

Simulation을 위해 OrCAD pspice로 [그림 2]과 같이 논리 회로를 구성하면 [그림 6]과 같은 결과를 얻을 수 있다. [그림 2]의 Logic gate는 입력이 3개이므로 입력을 A2, B2, C2 세 개로 설정한다.



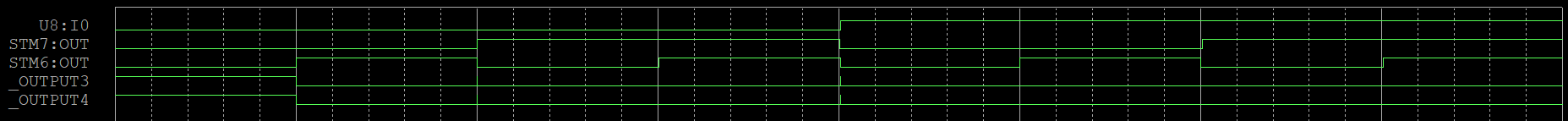
[그림 6] 실험 1-2 OrCAD 논리 회로도

논리 회로의 input을 A, B, C 로 설정한 뒤 [그림 7]와 같이 A와 B, C의 파형을 설정한다. X축 값인 시간의 범위는 0ms부터 80ms로 설정한다. 이때 3개의 input이 존재하므로 23=8가지 경우의 입력 파형이 존재한다. 각 wire에서의 입출력 파형을 확인하기 위해 Voltage/Level Marker를 설치한다.



[그림 7] 실험 1-2 입력 파형

이후 simulation file을 생성하고 Run To Time을 입력 신호의 시간 범위와 동일한 80ms로 설정한 뒤 RUN하면 [그림 5]와 같이 A, B, C의 파형에 따른 출력 파형이 발생하는 것을 볼 수 있다.



[그림 8] 실험 1-2 출력 파형

출력 결과를 바탕으로 실험 1-2 논리 회로도의 진리표를 작성하면 [표 2]와 같다.

Morgan\_Output3은 세 입력 A, B, C가 NOR gate에 입력되어 출력되는 결과이다. NOR gate는 입력에 1이 존재하지 않으면 1이, 하나라도 존재하면 0이 출력되는 gate이다. 따라서 1이 존재하지 않는 경우(⓵ A=0, B=0, C=0)에는 출력이 1인 것을 확인할 수 잇다. 반면에 1이 존재하는 경우(② A=0, B=0, C=1 / ③ A=0, B=1, C=0 / ④ A=0, B=1, C=1 / ⑤ A=1, B=0, C=0 / ⑥ A=1, B=0, C=1 / ⑦ A=1, B=1, C=0 / ⑧ A=1, B=1, C=1)에는 출력이 0인 것을 확인할 수 있다.

Morgan\_Output4는 세 입력 A, B, C가 NOT gate에 의해 반전된 값이 AND gate에 입력되어 출력되는 결과이다. AND gate는 입력이 모두 1이면 1이, 입력에 0이 하나라도 존재하면 0이 출력되는 gate이다. 따라서 입력이 모두 1인 경우(⓵ A’=1, B’=1, C’=1)에는 출력이 1인 것을 확인할 수 있다. 반면에 0이 하나라도 존재하는 경우(② A’=1, B’=1, C’=0 / ③ A’=1, B’=0, C’=1 / ④ A’=1, B’=0, C’=0 / ⑤ A’=0, B’=1, C’=1 / ⑥ A’=0, B’=1, C’=0 / ⑦ A’=0, B’=0, C’=1 / ⑧ A’=0, B’=0, C’=0)에는 출력이 0인 것을 확인할 수 있다.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **Y** | **A+B+C** | **(실험)** |  |  |  | **(실험)** |
| **0** | **0** | **0** | **0** | **0** | **1** | **1** | **1** | **1** | **1** |
| **0** | **0** | **1** | **1** | **1** | **0** | **1** | **1** | **0** | **0** |
| **0** | **1** | **0** | **1** | **1** | **0** | **1** | **0** | **1** | **0** |
| **0** | **1** | **1** | **1** | **1** | **0** | **1** | **0** | **0** | **0** |
| **1** | **0** | **0** | **1** | **1** | **0** | **0** | **1** | **1** | **0** |
| **1** | **0** | **1** | **1** | **1** | **0** | **0** | **1** | **0** | **0** |
| **1** | **1** | **0** | **1** | **1** | **0** | **0** | **0** | **1** | **0** |
| **1** | **1** | **1** | **1** | **1** | **0** | **0** | **0** | **0** | **0** |

[표 2] 실험1-2 진리표

* + - * 1. **실험 결과 분석**

실험 1-1에서 이용된 두 Logic gate의 Logic Equation을 나타내면 각각 **,** 와 같다. 실험 1-1을 통해 두 Logic gate의 출력 결과가 동일하다는 것을 알 수 있었으므로 다음이 성립함을 알 수 있다

**=**

위 식은 드모르간의 법칙으로, 실험을 통해 드모르간의 법칙이 성립함을 알 수 있었다.

또 실험 1-2에서 이용된 두 Logic gate의 Logic Equation을 나타내면 **,**와 같다. 실험 1-2을 통해 두 Logic gate의 출력 결과가 동일하다는 것을 알 수 있었으므로 다음이 성립함을 알 수 있다.

**=**

이는 입력이 2개일 때 뿐만 아니라 입력이 3개일 때도 드모르간의 법칙이 성립한다는 것을 나타낸다. 드모르간의 법칙은 변수가 여러 개인 경우에도 적용 가능하다.

1. **고찰**

이번 실험은 Boolean Algebra와 드모르간의 정리에 관한 실험으로, 논리합의 부정식과 논리곱의 부정식에 대해 강의노트에 제시된 Logic gate를 OrCAD Pspice를 통해 Simulation함으로써 드모르간의 법칙이 성립함을 알 수 있었다. 드모르간의 법칙을 이용할 때는 OR는 AND로, AND는 OR로 바꾸고, 각각의 입력에 NOT을 해주면 된다는 것을 알 수 있었다.

실험 1-2의 NOR gate 실험 회로의 Logic gate의 입력은 A, B, C로 3개인데 OrCAD Pspice에서 제공하는 gate의 입력은 2개로 제한이 되어있어 입력이 3개인 회로를 어떻게 설계해야 하는지에 대해 의문이 들었다. 이번 수업을 들으면서 3개의 입력 A, B, C를 받는 OR gate를 만들기 위해선, OR gate를 2개를 이용하면 된다는 것을 새롭게 알 수 있었다.

또 입력의 개수와 경우가 동일한 경우에는 똑같은 입력을 여러 개 만들지 않고 기존에 만들어놓은 하나의 입력을 wire만 다르게 하여 다른 gate에 연결해 이용하여 simulation할 수 있다는 사실을 이번 실험을 통해 알 수 있었다.

|  |  |
| --- | --- |
| **예비실험 보고서**  **(5주차)** | 학 번 : 122201856  이 름 : 김다영  제출일 : 2021.10.03  분 반 : 002 |

1. **실험 제목 : Karnaugh Map**
2. 실험 목적

* Karnaugh Map을 이용하여 Bool Algebra 함수를 단순화할 수 있다.
* Karnaugh Map을 이용해 단순화 한 함수를 OrCAD Pspice를 이용해 simulation한다.

1. 실험 이론
   1. Karnaugh Map
   * 카르노 맵은 Bool Algebra의 함수를 단순화하는 방법이다.
   * 단순화 되지 않은 불대수 표현에서 불 변수 A,B,C에 대해 f(A,B,C) = 라고 할 때, 의 숫자는 최소항 전개(minterm)의 진리표에서 출력이 1인 것을 의미한다.
   * 최소항(minterm)은 Truth table에서 1의 값을 가지는 변수를 모두 찾아서 SOP으로 표시할 때 각각의 곱의 항을 의미한다.
   1. Karnaugh Map 간소화 규칙

⓵ Truth table에서 1이 나오는 경우(최소항)를 추려내 카르노 맵에 표현한다.

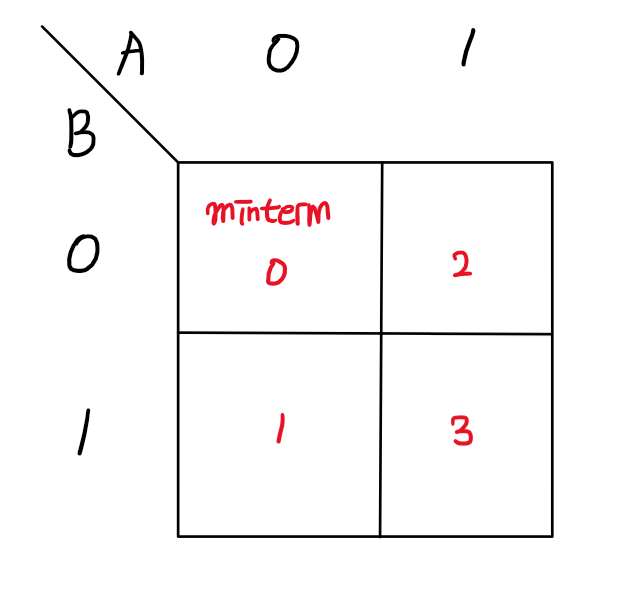
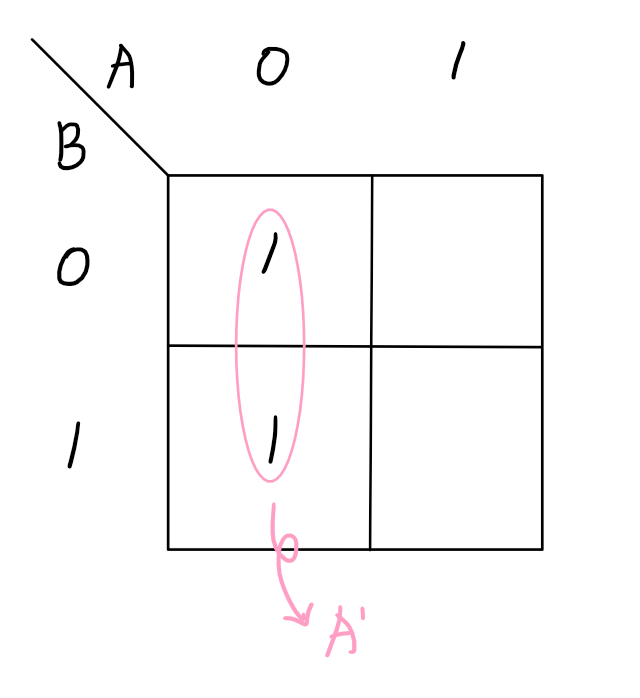
② 이웃을 2의 지수 승(1개, 2개, 4개, 8개 …)로 묶는다.

③ 반드시 사각형 형태로 묶어야 하며, 크게 묶을 수 있는한 크게 묶는다.

④ 각각의 사각형에서 묶인 변수를 식으로 표현하여 더한다.

* 1. Karnaugh Map 예시(2변수)

: f(A,B) =

왼쪽 그림은 2변수에서의 minterm 위치를 나타낸다. 해당 함수는 minterm 0과 1을 가지고 있으므로 0과 1 자리에 1을 채워 넣으면 오른쪽 그림과 같이 그릴 수 있다.

이후 이웃한 1에 대해 2의 지수 승 개로 가능한 크게 묶으면 다음 분홍색 동그라미처럼 묶을 수 있다. 모두 묶은 후에 묶은 결과를 식으로 나타내면 A’라는 결과를 얻을 수 있다.

- 참고문헌 -

[1] [외국] Charles Petzold. (2010). CODE, The Hidden Language of Computer Hardware and Software

[2] [Alan B. Marcovitz](http://www.yes24.com/SearchCorner/Result?domain=ALL&author_yn=Y&query=Alan+B.+Marcovitz), Introduction to Logic Design, 3/E, [McGraw-Hill College](javascript:void(0);) , 0070164908