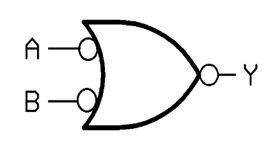
|  |  |
| --- | --- |
| **실험 결과 보고서**  **(3주차)** | 학 번 : 122201856  이 름 : 김다영  제출일 : 2021.09.18  분 반 : 002  실험조 : 개인 |

1. **실험 제목 : AND,NAND,OR,NOR logic Gates**
2. **실험 목적**

* 논리 게이트인 AND, OR, NOT 게이트의 동작특성을 이해한다.
* AND, OR, NOT 게이트의 진리표와 논리식을 실험을 통해 확인한다.

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 입력(A, B)이 2개, 출력(Y)이 1개 존재하며, Logic gate의 구성은 다음과 같다.



[그림 1] 실험 1-1 논리 회로도 – 각 입력에 NOT gate가 있는 NOR gate

텍스트, 거울이(가) 표시된 사진

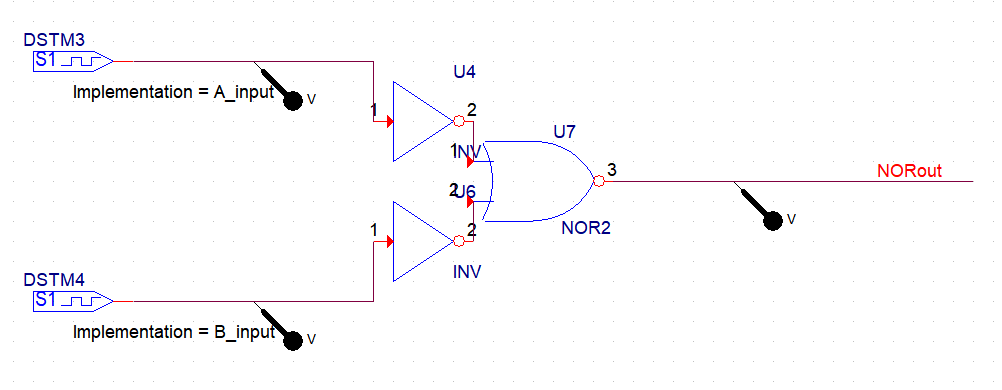
자동 생성된 설명

[그림 2] 실험 1-2 논리 회로도 – 각 입력에 NOT gate가 있는 NAND gate

* + - * 1. **Logic gate IC**

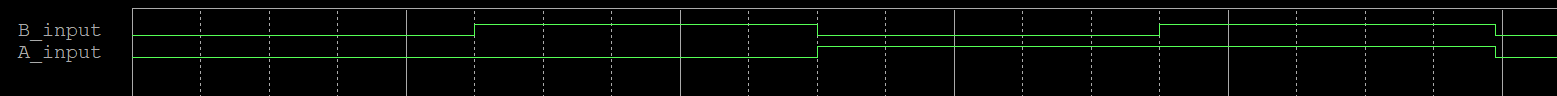
**실험 1-1**

Simulation을 위해 OrCAD pspice로 [그림 1]과 같이 논리 회로를 구성하면 [그림 3]과 같은 결과를 얻을 수 있다.

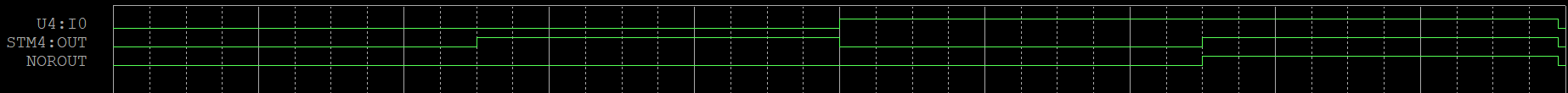
****

[그림 3] 실험 1-1 OrCAD 논리 회로도

논리 회로의 input을 A\_input과 B\_input로 설정한 뒤 [그림 4]와 같이 A\_input과 B\_input의 파형을 설정한다. 이때 2개의 input이 존재하므로 4가지 경우의 입력 파형이 존재한다. 각 wire에서의 입출력 파형을 확인하기 위해 Voltage/Level Marker를 설치한다.

[그림 4] 실험 1-1 입력 파형

이후 simulation을 RUN하면 [그림 5]와 같이 A\_input과 B\_input의 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.

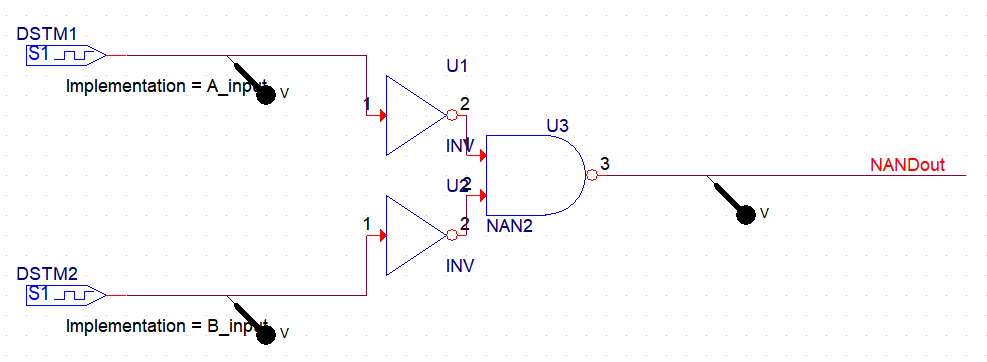
NOR gate는 input이 A와 B가 모두 0이 들어갔을 때 1이 출력되고 나머지 경우는 0이 출력된다. 이 회로의 경우는 각각의 입력이 NOT gate에 의해 반전되어 NOR gate에 들어가므로 input이 A와 B가 모두 1이 들어갔을 때만 1이 출력되고 나머지 경우는 0이 출력됨을 알 수 있다. 모두 1이 들어갔을 때만 1이 출력되고 0이 하나라도 있는 경우엔 0이 출력되므로, 이는 AND gate와 동일한 동작 특성을 가졌다는 것을 알 수 있다.

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **Y** |
| **0** | **0** | **0** |
| **0** | **1** | **0** |
| **1** | **0** | **0** |
| **1** | **1** | **1** |

[표 1] 실험 1-1 진리표

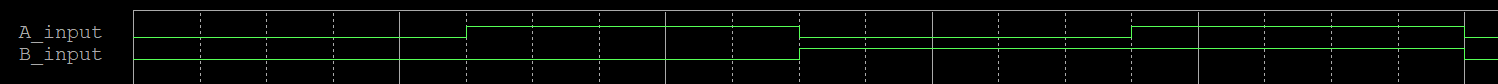
**실험 1-2**

Simulation을 위해 OrCAD pspice로 [그림 2]과 같이 논리 회로를 구성하면 [그림 6]과 같은 결과를 얻을 수 있다.

****

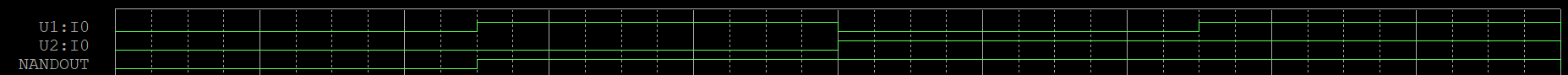
[그림 6] 실험 1-2 OrCAD 논리 회로도

논리 회로의 input을 A\_input과 B\_input로 설정한 뒤 [그림 7]와 같이 A\_input과 B\_input의 파형을 설정한다. 이때 2개의 input이 존재하므로 4가지 경우의 입력 파형이 존재한다. 각 wire에서의 입출력 파형을 확인하기 위해 Voltage/Level Marker를 설치한다.

****

[그림 7] 실험 1-2 입력 파형

이후 simulation을 RUN하면 [그림 8]와 같이 A\_input과 B\_input의 파형에 따른 출력 파형이 발생하는 것을 볼 수 있다.

****

[그림 8] 실험 1-2 출력 파형

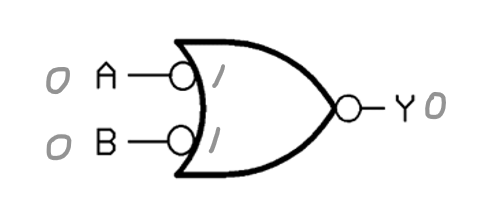
출력 결과를 바탕으로 실험 1-2 논리 회로도의 진리표를 작성하면 [표 2]와 같다.

NAND gate는 input이 A와 B가 모두 1이 들어갔을 때 0이 출력되고 나머지 경우는 1이 출력된다. 이 회로의 경우는 각각의 입력이 NOT gate에 의해 반전되어 NAND gate에 들어가므로 input이 A와 B가 모두 0이 들어갔을 때만 0이 출력되고 나머지 경우는 1이 출력됨을 알 수 있다. 모두 0이 들어갔을 때만 0이 출력되고 1이 하나라도 있는 경우엔 1이 출력되므로, 이는 OR gate와 동일한 동작 특성을 가졌다는 것을 알 수 있다.

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **Y** |
| **0** | **0** | **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **1** |

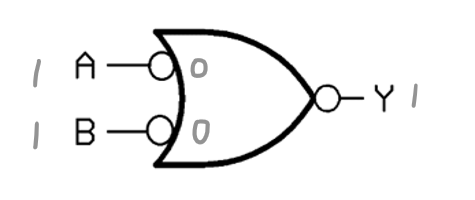
[표 2] 실험1-2 진리표

**실험 결과 검토**

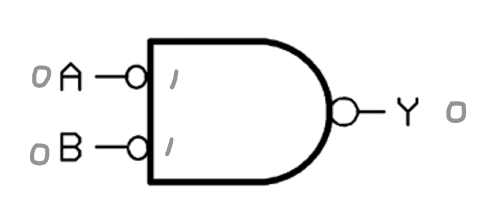
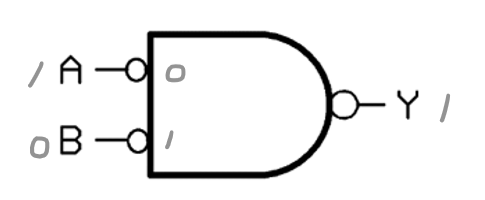
 텍스트, 시계, 게이지이(가) 표시된 사진

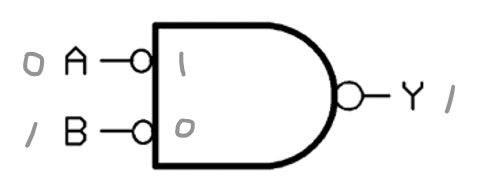
자동 생성된 설명

텍스트, 게이지이(가) 표시된 사진

자동 생성된 설명 

실험 1-1의 논리 회로를 OrCAD가 아닌 수기로 각 input을 변경해가며 회로의 출력 값을 확인한 결과 위의 [표 1]과 같이 모든 input의 1일 때만 1이 출력되고 하나라도 input이 0이 있으면 0이 출력되는 것을 확인할 수 있었다.

 광장이(가) 표시된 사진

자동 생성된 설명

실험 1-2의 논리 회로를 OrCAD가 아닌 수기로 각 input을 변경해가며 회로의 출력 값을 확인한 결과 위의 [표 2]과 같이 모든 input의 0일 때만 0이 출력되고 하나라도 input이 1이 있으면 1이 출력되는 것을 확인할 수 있었다.

1. **고찰**

이번 실험을 통해 OrCAD pspice를 통해 Digital Logic Circuit을 simulation하기 위해 회로를 어떻게 설정하고 input을 어떻게 설정해야 하는지, simulation을 어떻게 RUN시켜야 하는지 등 ORCAD 사용법을 더욱 자세히 알 수 있었다.

처음에 실험 1-1, 1-2의 논리 회로도를 보고 그냥 NOR, NAND gate가 아니라 왜 input이 NOT gate에 의해 반전돼서 들어가는지에 대해 의문이 들었는데, 실험 결과를 분석해본 결과 두 논리 회로들이 각각 AND gate와 OR gate의 동작 특성과 일치한다는 것을 확인할 수 있었다. 해당 실험에서는 다른 회로도여도 같은 동작을 할 수 있다는 알려주기 위해 input을 반전시킨 것이 아닐까 하고 생각할 수 있었다. 또 실험 결과를 통해 [그림 1], [그림 2]의 회로에 드모르간의 법칙을 이용하면 각각의 회로가 AND gate, OR gate가 된다는 것을 알 수 있었다.

처음에 project 파일을 잘못 만들어서 각 input을 입력할 때 사용하는 Edit Pspice Stimulus 버튼이 보이지 않아 실험하는데 시간이 지체되곤 했다. 다음부턴 이런 실수가 발생하지 않도록 project 파일을 정확한 형식에 맞춰 생성하도록 할 것이다.

|  |  |
| --- | --- |
| **예비실험 보고서**  **(4주차)** | 학 번 : 122201856  이 름 : 김다영  제출일 : 2021.09.18  분 반 : 002 |

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. 실험 이론
   1. Boolean Algebra
      1. 부울대수의 기본 정리와 법칙들은 논리식을 계산하고 논리회로를 간략화시키는 데 사용된다.
      2. 부울대수는 교환법칙, 결합법칙, 분배법칙 등의 기본 법칙들은 통상적인 대수학에서의 법칙들과 거의 같다.
      3. 부울대수는 ‘0’과 ‘1’의 2진수 2개 요소와 3개의 연산자( ·, +, - )를 사용한다. 3개의 연산자는 AND, OR, NOT 게이트의 논리소자에 대응된다.
      4. 논리 연산의 주요 법칙
      * 교환 법칙

AB=BA

A+B=B+A

* + - 결합 법칙

(AB)C=A(BC)

(A+B)+C=A+(B+C)

* + - 분배 법칙

A(B+C)=AB+AC

A+(BC)=(A+B)(A+C)

* + - 항등 법칙

0⋅A=0

0+A=A

1⋅A=A

1+A=1

* + - 반전 법칙



* 1. 드모르간의 정리
     1. NAND 게이트나 NOR 게이트를 AND 게이트나 OR 게이트에 NOT 게이트를 사용하여 등가회로로 구성하여 같은 결과를 얻을 수 있는 방법을 드모르간의 법칙이라 한다.
     2. 드모르간의 법칙을 논리회로 변환에 응용하는 순서는 다음과 같다.

AND 게이트는 OR 게이트로, OR 게이트는 AND 게이트로 대치한다.

각 변수를 invert한다.

마지막 결과 전체에 대하여 invert한다.

가능하면 두 번 invert하여 회로를 간략화 한다.

* + 1. 드모르간의 법칙은 변수의 개수에 무관하게 적용할 수 있으며, Boolean Algebra 전체 또는 일부에만 적용할 수도 있다.
    2. 드모르간의 법칙 응용

논리합의 부정

: 논리합의 부정에 대한 드모르간의 법칙 식은 다음과 같이 나타낼 수 있다.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | A+B |  |  |  |  |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |

[표 1] 논리합의 부정에 대한 진리표

논리곱의 부정

: 논리곱의 부정에 대한 드모르간의 법칙 식은 다음과 같이 나타낼 수 있다.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B |  |  |  |  |  |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |

[표 2] 논리곱의 부정에 대한 진리표

- 참고문헌 -

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

[2] http://www.ktword.co.kr/test/view/view.php?m\_temp1=5543