

실험 1. 논리회로 기초

이름 : 오승준
학번 : 20210661

1. 개요

실험 1은 기초 논리 게이트를 학습 후 회로를 구성하고, Verilog로 구현하는 것이다. 이번 실험에서의 세부 목표는 아래와 같다.

- Xilinx Vivado HDL 통합 개발 환경에 익숙해진다.
- Verilog 기초 문법을 이해하고 직접 코드 작성을 진행한다.
- Testbench를 작성하고 Simulation 사용법을 습득한다.
- Functionally complete를 이해한다.

2. 이론적 배경

1) Positive logic, Negative logic

- 논리연산을 전자 회로에 대입할 때, True와 False를 임의로 High(높은 전압), Low(낮은 전압)에 대응할 수 있다. 이때 논리값의 참값을 High로 잡느냐, Low로 잡느냐에 따라 Positive logic과 Negative logic으로 나뉜다. Positive logic의 경우 참을 High, 거짓을 Low로 표현하고, Negative logic의 경우 참을 Low, 거짓을 High로 표현한다.

2) HDL

HDL은 Hardware Description Language의 약자로, 디지털 시스템 논리 회로 구조를 표현하기 위한 언어다. 대표적인 HDL로는 Verilog와 VHDL이 있으며, 이 실험에서는 Verilog를 이용하였다.

3)Functionally complete set

Functionally complete란, 모든 불 대수식을 표현할 수 있는 어떤 논리 연산의 집합을 말한다. 예를 들어, {AND, OR, NOT} 은 Functionally complete set이다. Functionally complete set을 증명하기 위해서는, set에 포함된 연산들을 통해 AND, OR, NOT 연산을 할 수 있음을 보이면 된다.

3. 실험 준비

- 1) Lab Orientation 영상을 바탕으로 Vivado 사용법과 Verilog 문법을 학습한다.
- 2) NOT, OR, AND, NOR, NAND 연산의 진리표를 작성한다.
 - 각 연산에 대한 진리표는 아래와 같다.

NOT		OR			AND			NOR			NAND		
inA	out	inA	inB	out	inA	inB	out	inA	inB	out	inA	inB	out
0	1	0	0	0	0	0	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0	0	1	0	0	1	1
		1	0	1	1	0	0	1	0	0	1	0	1
		1	1	1	1	1	1	1	1	0	1	1	0

3) Lab1_1

- AND 게이트 작성

AND 연산 진리표는 3.2)에 있으며, 이를 Verilog에서 구현시 아래와 같은 함수를 사용한다.

and(output, input1, input2)

문제에서 input의 값을 1, 2초마다 변경하라는 조건의 경우,

testbench에서 다음과 같은 방식으로 조건을 구현한다.

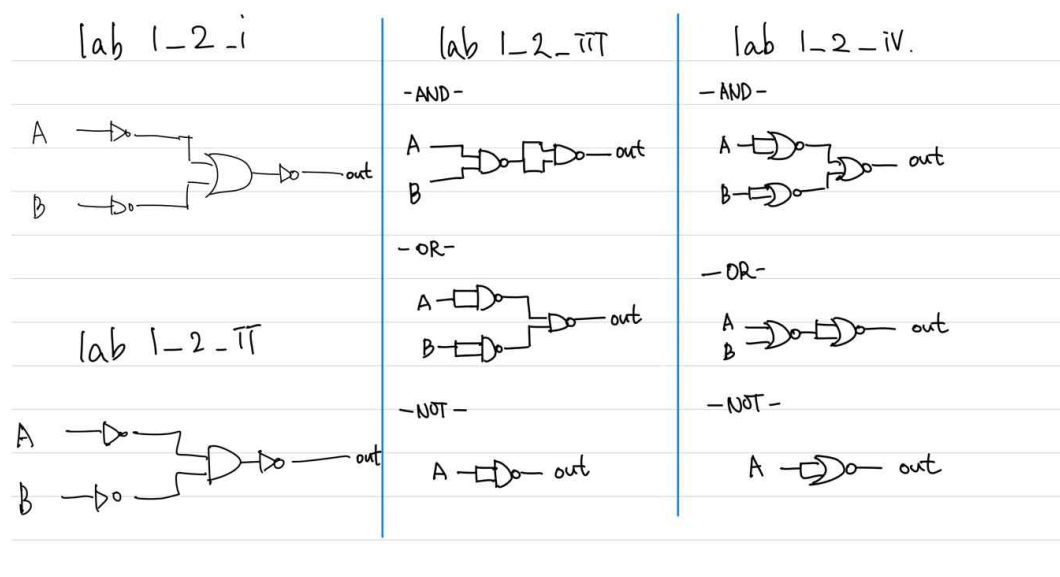
#1 A <= !A

#2 B <= !B

4)Lab1_2

- Functionally complete 집합 구현

집합 구현에 있어 활용되는 연산의 진리표는 3.2)에 있으며, 이를 통하여 각 문제에서 요구하는 회로도를 수기로 작성 시 아래와 같다.



이를 구현하기 위해서, Verilog에서 아래의 함수들을 활용하였다.

not(output, input)

and(output, input1, input2)

or(output, input1, input2)

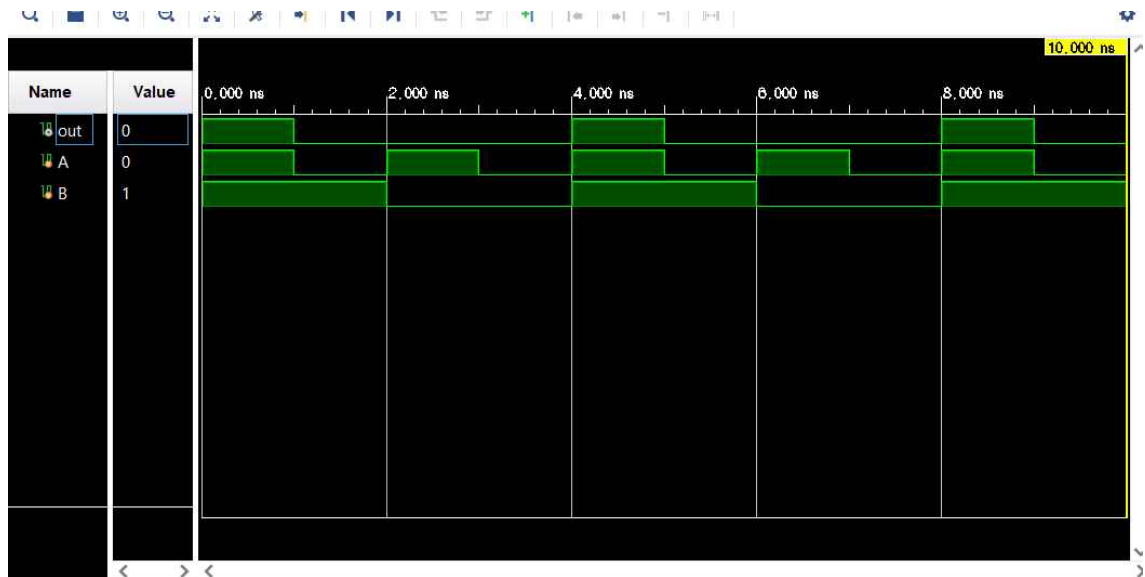
nand(output, input1, input2)

nor(output, input1, input2)

4. 실험 결과

1) AND 게이트 구현

AND 게이트를 구현한 후 Simulation을 실행한 결과 아래와 같은 그림을 얻을 수 있다.



이때,

0ns, 4ns, 8ns에서 A=1, B=1일 때 out=1.

1ns, 5ns, 9ns에서 A=0, B=1일 때 out=0,

2ns, 6ns에서 A=1, B=0일 때 out=0,

3ns, 7ns에서 A=0, B=0일 때 out=0임을 확인 할 수 있으며, 이는 AND 연산의 진리값과 같음을 확인할 수 있다.

(마지막에 Simulation과 진리표 비교를 통해 값이 제대로 나온 확인해야함.)

2) Functionally complete 집합 구현

각 문제에서 요구하는 회로를 Verilog를 통해 작성 후 Schematic 기능을 통해 회로 작성 시 아래와 같은 결과를 얻었다.

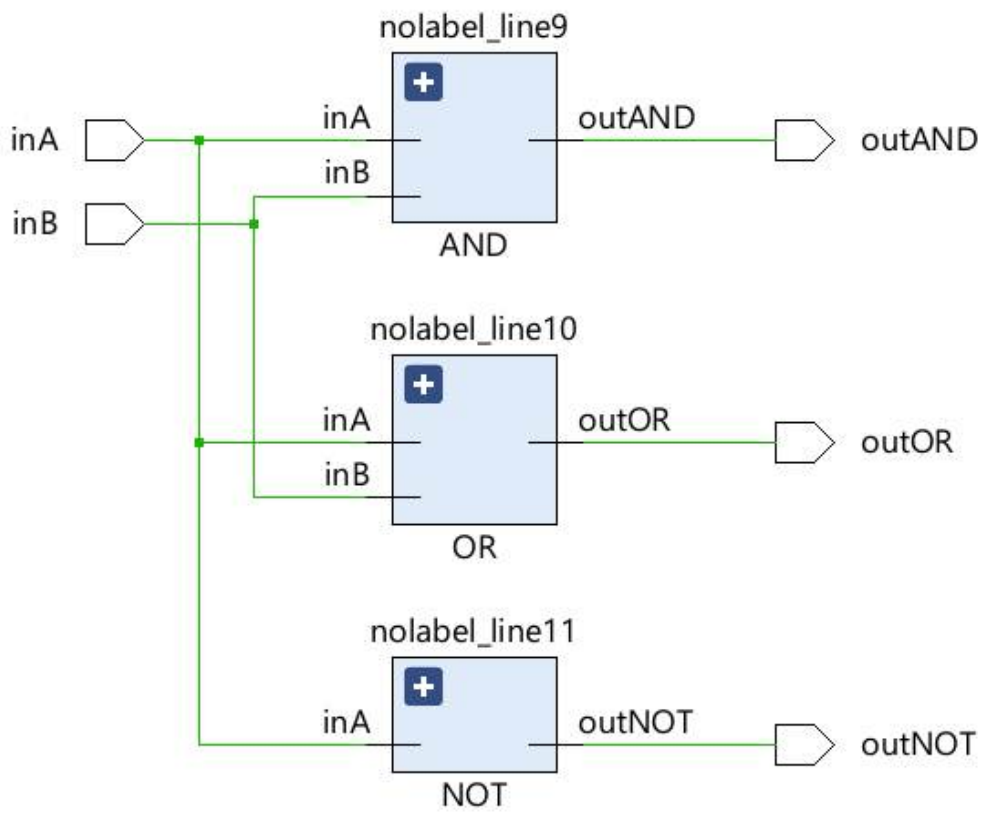
- Lab1_2_i

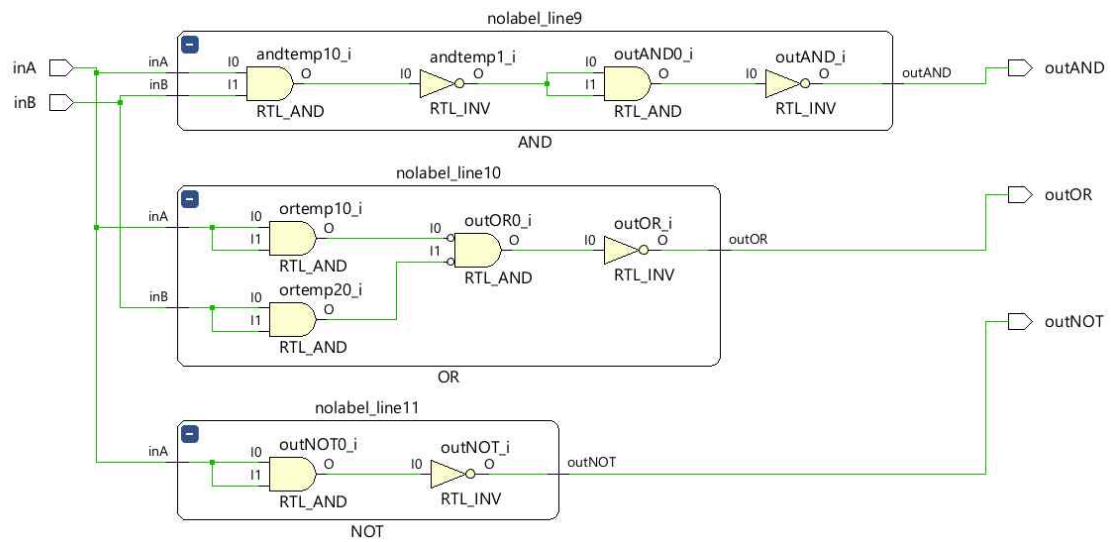


- Lab1_2_ii

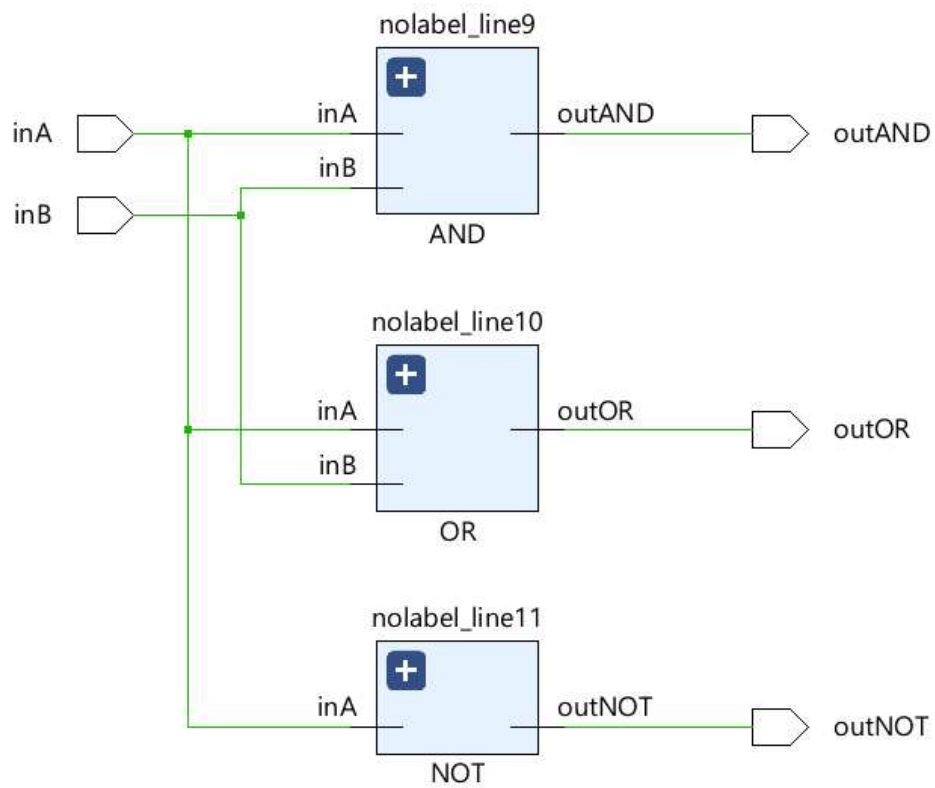


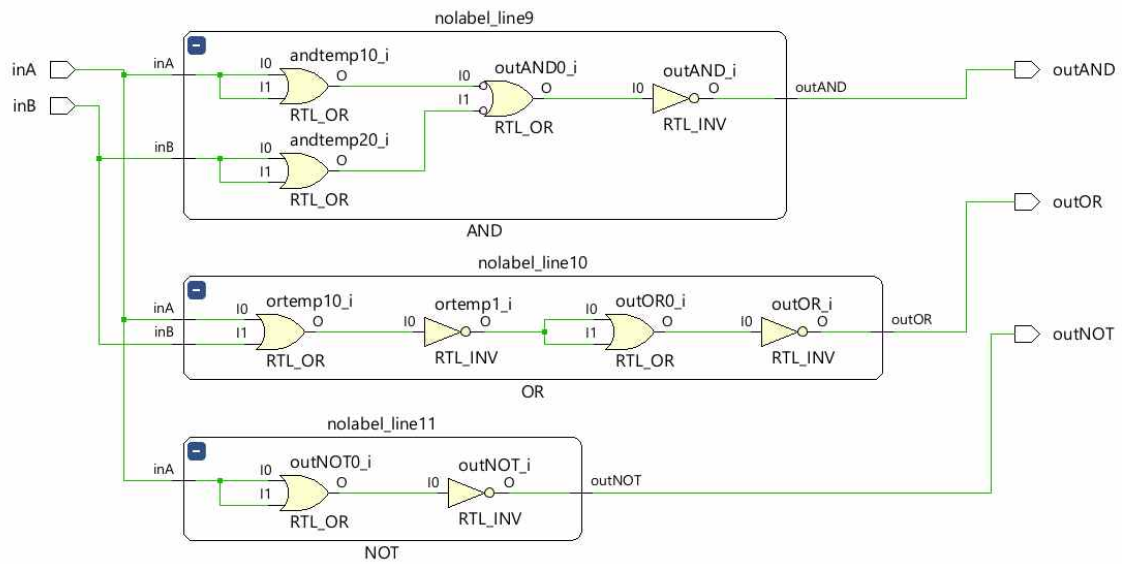
-Lab1_2_iii





-Lab1_2_iv





이때, 각 문제에 작성된 회로가 실험 준비 과정 중 수기로 작성한 회로와 일치하므로, 구현이 정상적으로 되었음을 확인할 수 있다.

5. 논의

실험 1을 진행하며 Vivado라는 새로운 작업환경과 Verilog라는 HDL 언어를 처음 접할 수 있었다. 코드가 간단하게 구성되어있었지만 Vivado에 친숙하지 않아 많은 시간이 소요되었다. 추후 Lab들과 실습을 통하여 Vivado라는 환경과 Verilog라는 언어에 적응하여야 함을 깨달을 수 있었다.