

디지털 시스템 설계 Lab1 보고서 – 20230251 정채현

1. Lab 1_1

이번 과제는 verilog 를 활용해 and 게이트를 만드는 것이 목표인 과제이다. Lab1_1에서는 Verilog 를 활용하는 방법과 testbench 를 이용해 각 신호의 주기를 조절하는 법 등을 학습하는 것이 목표이며, 최종적으로는 and 게이트를 거친 두 신호의 파형 을 기록하는 것이 목표이다.

B. 이론적 배경

HDL (Hardware Description Language)는 디지털 시스템의 논리 회로 구조를 표현하기 위한 언어다. 주로 사용되는 언어로는 Verilog 와 VHDL 이 있으며, 이번 디지털 시스템 설계 강의에서는 Verilog 를 사용한다. SystemVerilog 와는 다른 언어로 지원하는 기능에 차이가 있으므로 주의가 필요하다.

And 게이트는 논리 게이트의 일종으로써, and(논리곱)을 구현하는 게이트이다. 논리곱은 명제가 모두 참인지를 나타내는 논리 연산을 의미한다.

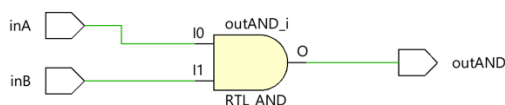
C. 실험 준비

And 게이트를 작성하기 위해서는 and(outAND, inA, inB)를 사용해야 한다.

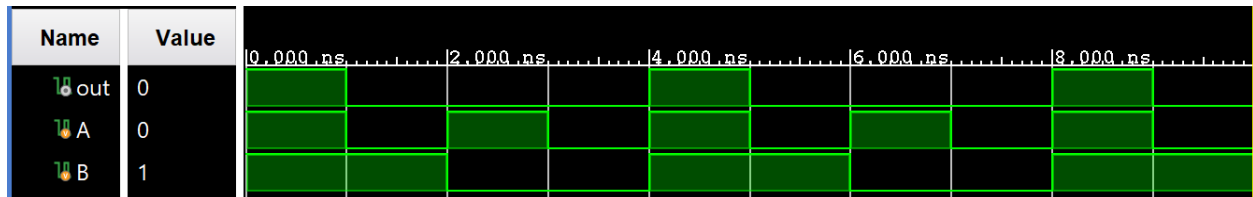
Testbench 를 이용해 1ns 와 2ns 에서 신호를 변환하기 위해서는 #1 A <= !A; 와 같은 형식으로 작성할 수 있다.

D. 결과

Verilog 로 작성한 회로도 는 다음 사진과 같다.



완성된 회로를 simulation 했을 때의 파형은 다음과 같다.



A 와 B 가 모두 1 일 경우에는 out 도 1, A 와 B 둘 중 하나라도 1 이 아닌 경우에는 out 이 0 이
나옴을 확인할 수 있다.

2. Lab1_2

A. 개요

1 번 문제는 AND 게이트를 OR 게이트, NOT 게이트 만을 활용해서 구현하는 문제이다. 2 번
문제는 OR 게이트를 AND 게이트, NOT 게이트 만을 활용해서 구현하는 문제이다. 3 번 문제는
NAND 게이트만으로 NOT, AND, OR 게이트를 각각 구현하는 문제이다. 4 번 문제는
NOR 게이트만으로 NOT, AND, OR 게이트를 각각 구현하는 문제이다.

B. 이론적 배경

NAND 게이트는 AND 게이트에 NOT 연산을 합친 것이다. 즉, 입력이 모두 1 일 때만 0 을
출력하고, 나머지 입력에서는 모두 1 을 출력하는 게이트이다. NOR 게이트는 OR 게이트에 NOT
연산을 합친 것이다. 입력이 모두 0 일 때만 1 이 출력되고, 나머지 경우에는 0 이 출력되는
게이트이다.

Functionally complete set 은 어떤 논리 연산의 집합으로 모든 불 대수식을 표현할 수 있는
set 을 의미한다. 예를 들어, 기초 연산만으로 이루어진 {AND, OR, NOT}은 Functionally
complete 이다. 따라서 어떠한 집합이 Functionally complete 임을 증명할 때, 포함된 연산을
조합하여 AND, OR, 그리고 NOT 연산을 만들어 낼 수 있음을 보이면 된다. 1_2_iii 와 1_2_iv 에서
각각 NAND, NOR 가 functionally complete set 임을 증명할 것이다.

드모르간 법칙은 $(X_1 + X_2 + \dots + X_n)' = X_1' X_2' \dots X_n'$, $(X_1 X_2 \dots X_n)' = X_1' + X_2' + \dots + X_n'$
라는 것이다.

C. 실험 준비

논리 연산의 원리를 이용해 적절히 코드에 제시된 모듈들을 구현한다. 구현한 후, RTL Analysis -> Elaborated Design -> Schematic 기능으로 작성한 회로를 확인한다.

1_2_i 는 OR, NOT 만으로 AND 를 만드는 문제로, $(A' + B')' = AB$ 인 점을 이용하면 된다.

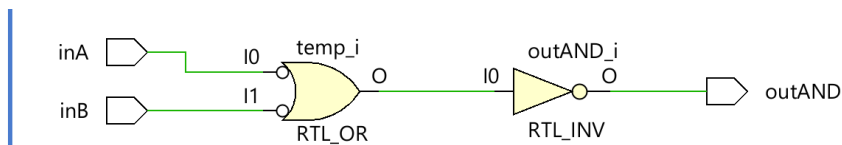
1_2_ii 는 AND, NOT 만으로 OR 를 만드는 문제로, $(A'B')' = A+B$ 인 점을 이용하면 된다.

1_2_iii 는 NAND 만으로 AND, OR, NOT 를 구현하는 것이다. NOT A 는 A 와 A 를 NAND 게이트에 넣으면 된다. AND 는 A,B 에 NAND 연산을 해 주고, 그 값에 NOT 연산을 해주는 방식으로 했다. OR 는 A 와 B 를 각각 NOT 연산한 후 NAND 를 해서 만들었다.

1_2_iv 는 NOR 만으로 AND, OR, NOT 를 구현하는 것인데, 1_2_iii 와 상당히 유사하다. NOT A 는 A 와 A 를 NOR 게이트에 넣으면 된다. OR 는 A,B 에 NOR 연산을 해 주고, 그 값에 NOT 연산을 해주는 방식으로 했다. AND 는 A 와 B 를 각각 NOT 연산한 후 NOR 를 해서 만들었다.

D. 결과

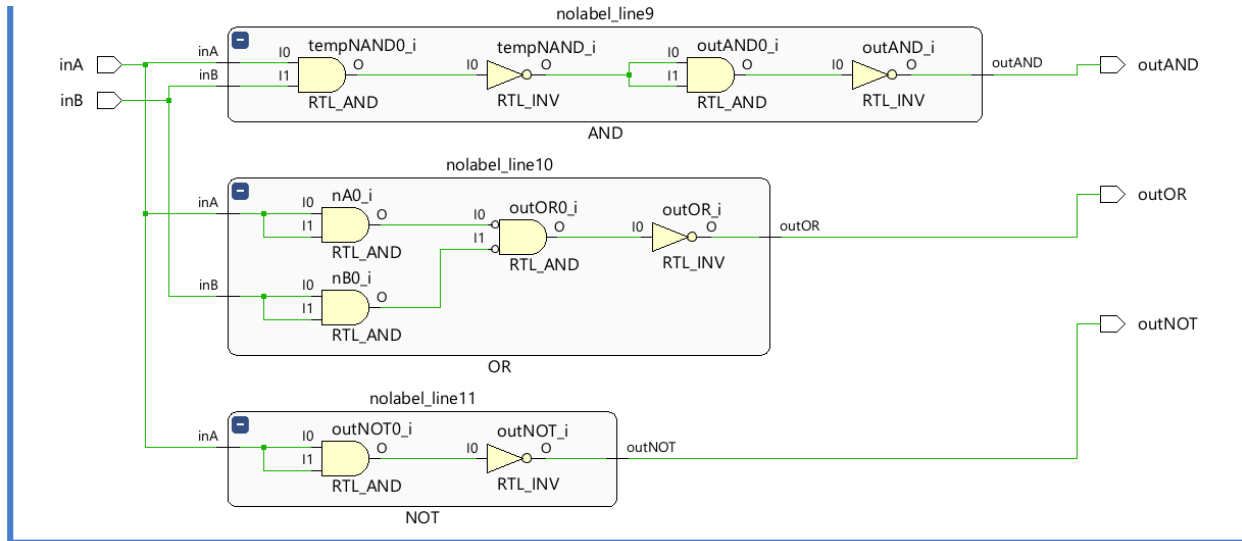
1_2_i 의 회로는 다음과 같다.



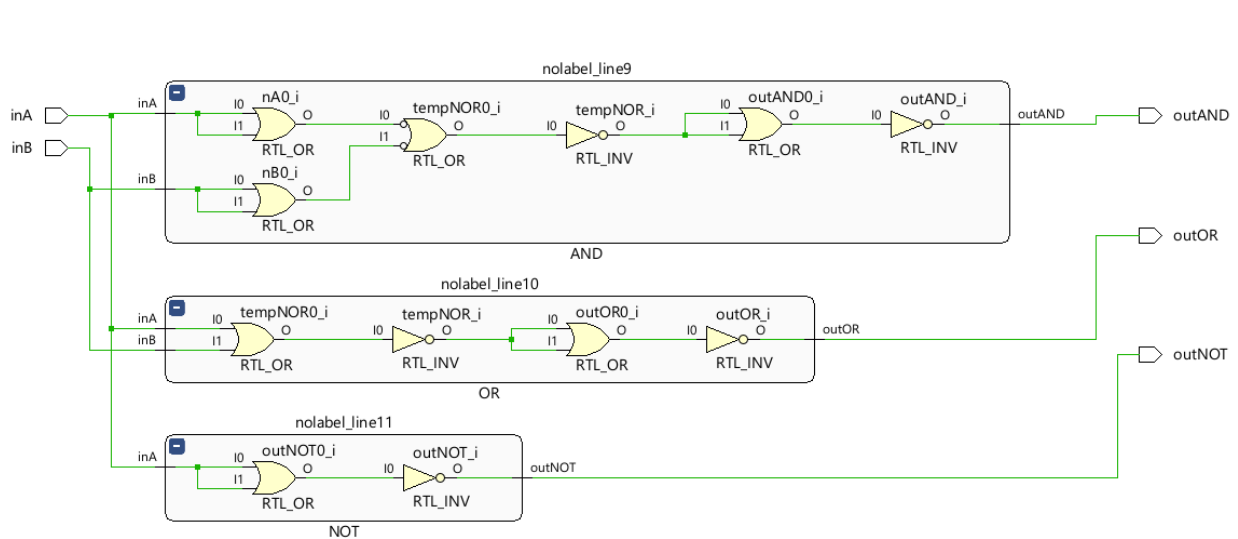
1_2_ii 의 회로는 다음과 같다.



1_2_iii 의 회로는 다음과 같다.



1_2_iv 의 회로는 다음과 같다.



E. 논의

이번 실험을 통해서 각각의 게이트의 사용 방법을 익히고, Verilog 프로그램의 schematic 기능에 대해 공부할 수 있었다. NAND 와 NOR 게이트가 Functionally complete 하다는 것을 실습을 통해 직접 알아볼 수 있었다. Xilinx Vivado 의 사용법이 아직 익숙하지 않아 어려웠지만, 사용법을 익힐 수 있어서 도움이 되었다.