

# 실험 1. 논리회로 기초

2022. 03. 22

20210054 정하우

## 1. 개요

기초 논리 게이트로 회로를 구성하고, Verilog로 구현한다. 세부적인 학습 목표는 다음과 같다.

- Xilinx Vivado HDL 통합 개발 환경 적응
- Verilog 문법 습득
- 회로의 작동을 확인하기 위한 Testbench 작성 및 시뮬레이션 사용 방법 습득
- Functionally complete 개념 이해

## 2. 이론적 배경

### 1) Positive Logic / Negative Logic

논리식을 실제 전자 회로로 구현할 때 불 대수의 참과 거짓을 High (높은 전압)와 Low (낮은 전압)에 임의로 대응할 수 있다. 예를 들어 Positive logic (Active high)의 경우 참을 High, 거짓을 Low로 표현하고, Negative logic (Active-low)의 경우 참을 Low, 거짓을 High로 표현한다. 이 둘은 논리 회로를 전기적으로 표현하는 방법만 다를 뿐 기능적인 차이는 없다. 단, 실제 부품을 사용하여 회로를 구성할 때는 Active-high와 Active-low 부품이 섞여 있을 수 있으므로 주의해야 한다.

### 2) HDL (Hardware Description Language)

Hardware Description Language는 디지털 시스템의 논리 회로 구조를 표현하기 위한 언어다. 주로 사용되는 언어로는 Verilog와 VHDL이 있으며, 이번 디지털 시스템 설계 강의에서는 Verilog를 사용한다. System Verilog와는 다른 언어로 지원하는 기능에 차이가 있으므로 주의가 필요하다.

### 3) Functionally complete set

어떤 논리 연산의 집합으로 모든 불 대수식을 표현할 수 있을 때 이 집합은 Functionally complete이다. 예를 들어, 기초 연산만으로 이루어진 {AND, OR, NOT}은 Functionally complete이다. 따라서 어떠한 집합이 Functionally complete임을 증명할

때, 포함된 연산을 조합하여 AND, OR, 그리고 NOT 연산을 만들어 낼 수 있음을 보이면 된다. 이번 실험에서 보이는 4가지 경우({OR, NOT}, {AND, NOT}, {NAND}, {NOR})들 모두 Functionally complete 하다.

#### 4) 진리표

A	B	AND	OR	NOT(~A)	NOT(~B)	NAND	NOR
0	0	0	0	1	1	1	1
0	1	0	1	1	0	1	0
1	0	0	1	0	1	1	0
1	1	1	1	0	0	0	0

### 3. 실험

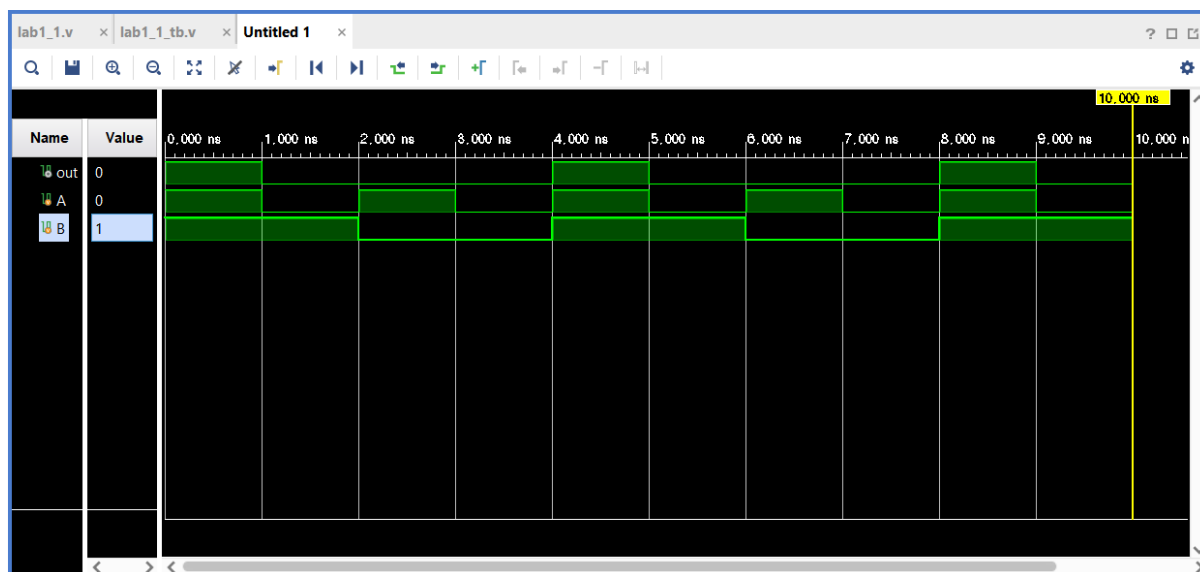
#### 1) AND 게이트 구현

ㄱ. Verilog로 lab1\_1.v 파일에 AND 게이트를 작성했다.

ㄴ. Verilog로 lab1\_1\_tb.v 파일에 Testbench를 작성했다.

- 총 10ns 동안 시뮬레이션을 수행했다.
- AND 게이트의 두 입력을 각각 1ns와 2ns마다 한 번씩 반전시켰다.
- 0ns일 때 입력의 초기값은 모두 1로 설정했다.

ㄷ. 시뮬레이션 결과를 확인한다.

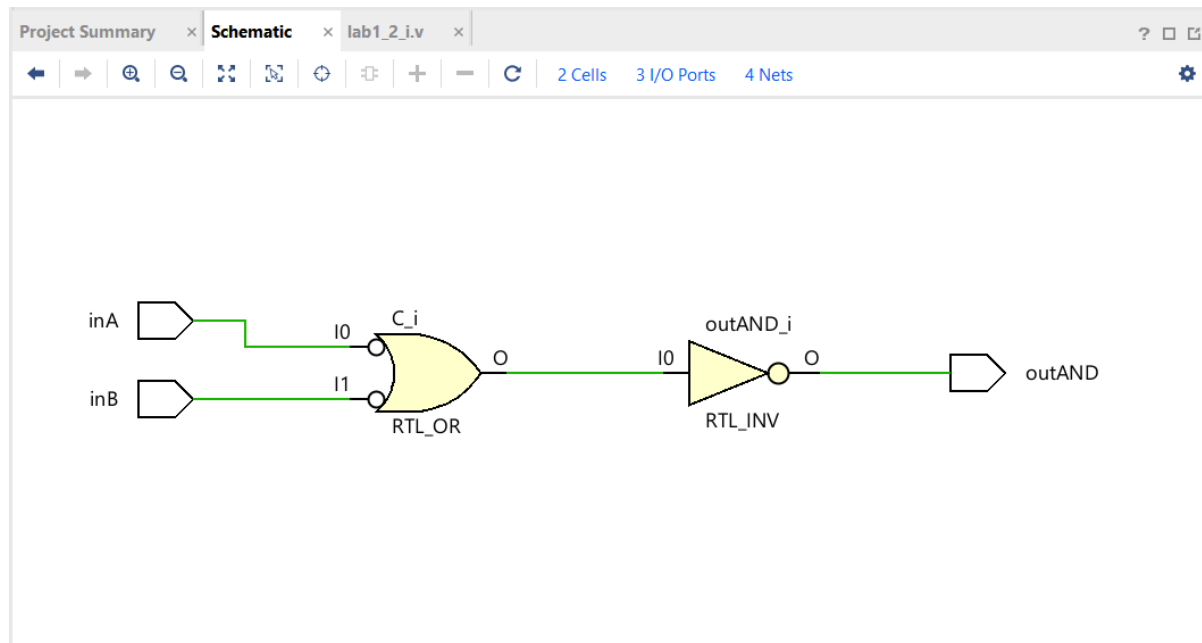


#### 2) Functionally complete 집합 구현

ㄱ. 주어진 연산만으로 코드에 제시된 모듈을 구현하여 {AND, OR, NOT} 집합을 완성했다.

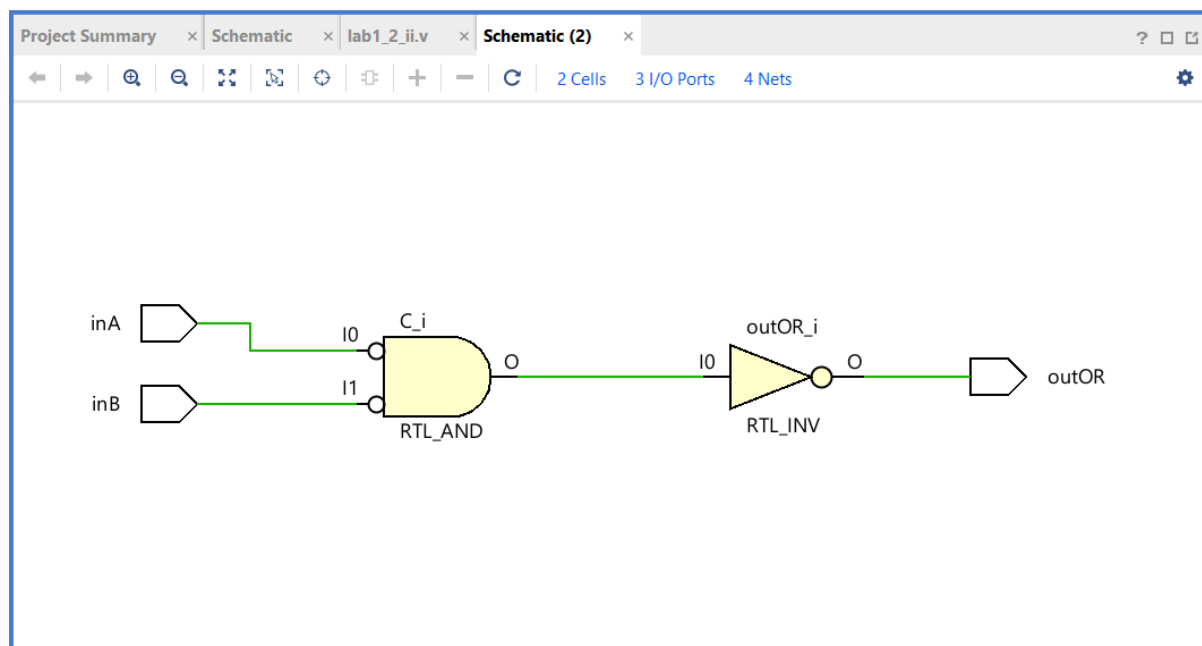
i. {OR, NOT} – AND

$$(A' + B')' = AB$$



ii. {AND, NOT} – OR

$$(A'B')' = A + B$$

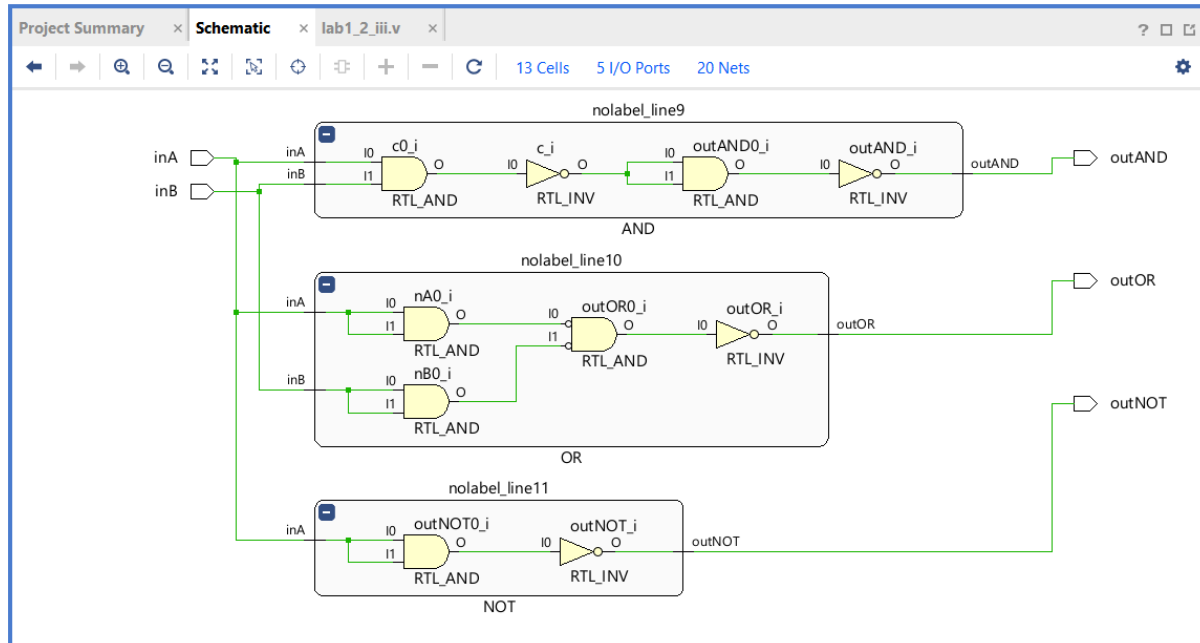


iii. {NAND} – AND, OR, NOT

$$AND - ((AB)' + (AB)')' = AB$$

$$OR - ((AA)'(BB)')' = A + B$$

$$NOT - (AA)' = A'$$

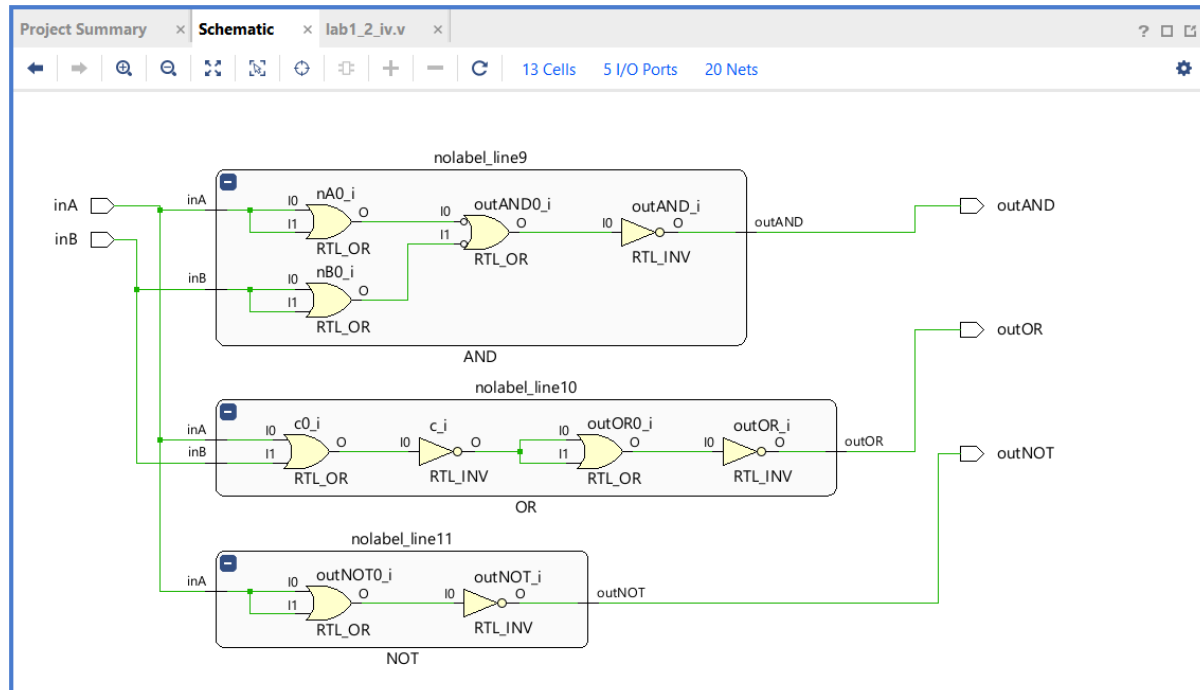


iv. {NOR} – AND, OR, NOT

$$AND - ((A + A)' + (B + B)')' = AB$$

$$OR - ((A + B)' + (A + B)')' = A + B$$

$$NOT - (A + A)' = A'$$



#### 4. 토론 및 결론

코드를 작성하는 부분에서는 기본 틀이 있어서 큰 어려움 없이 해 낸 것 같다. 이번 실험을 통해서 수업시간에 배운 4가지 Functionally complete 집합을 직접 AND OR NOT 게이트를 구현 해 보면서 복습을 한 시간이 되었다.