3주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20191619 이름: 이동석

**1. FPGA**

FPGA는 다음과 같이 실습시간에 배운 방법을 통해 동작 시킬 수 있다.

1.1 Verilog 코딩

우선, 앞선 주차에서 설명한 것 처럼 HDL언어를 통해 코딩을 해야한다. 여기서의 코딩은 C언어나 Java와 같은 코딩과는 조금 다르며, 디지털 회로의 동작이나 구조를 기술하기 위함이다. 즉, 하드웨어에 대한 이해가 무조건적으로 필요하다. Verilog언어를 통해 코딩을 작성한다.

1.2 Device assignment

강의자료에서는 run synthesis를 하라고 되어있지만, 이는 pin assingment 이후 실행하는 것으로 알고 있다. 코딩을 제대로 했다면, 실습에서 사용할 device를 연결한다. 사용하는 device는 xc7a75tfgg484이며, 이는 왼쪽 상단 setting > project device에서 확인 가능하다.

1.3 Pin assignment

이제 사용할 pin을 지정해주어야 한다. 코드에서 input과 output을 보드의 어느 핀에 할지 직접 지정해주는 과정이다. 예로, input1에 1번 스위치를 지정하고 output1에는 1번 LED를 지정해주는 식이다. 이때, 지정할 Pin의 port는 1주차 강의자료를 참고한다. 확장자가 xdc파일인 constraint 파일을 생성해주어야 하며, add sources 메뉴에서 생성 가능하다. 이후, I/O Ports 메뉴로 들어가 직접 지정한다.

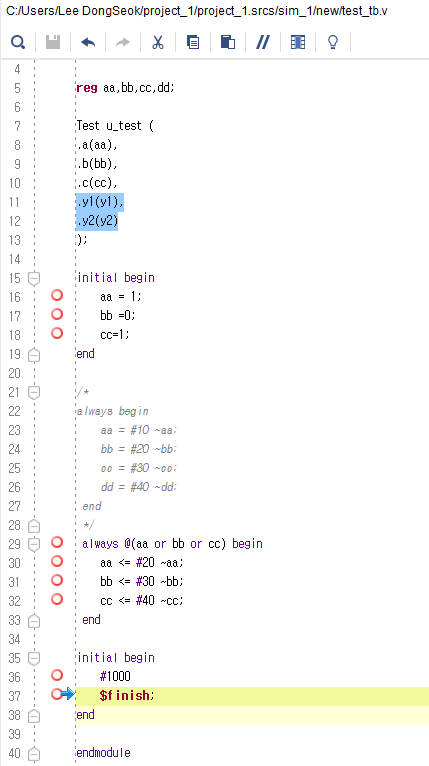
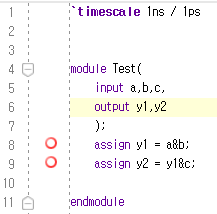
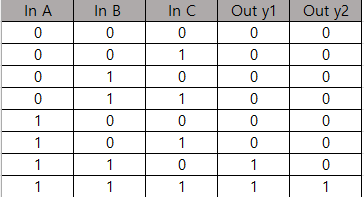
1.4 Synthesis / Implemnet

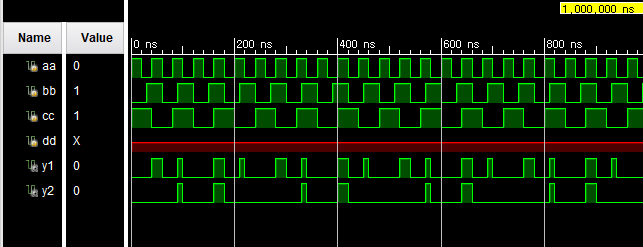
왼쪽 메뉴바를 보면 Run Synthesis와 Run Implement 메뉴가 보일 것이다. 반드시 Synthesis 과정을 먼저 해주어야한다. 합성과정은 앞서 말했듯 우리가 작성한 Verilog언어를 하드웨어가 이해할 수 있도록 컴파일 하는 과정이다. 이 합성과정에서 netlist를 만들고, implement를 통해 FPGA 보드에 place & route 하게 된다. (자일링스 홈페이지 참조) 이후, 비트 스트림 파일까지 생성한다.

1.5 Device Configuration

이제 Open Hardware 메뉴를 통해 컴퓨터와 연결되어 있는 FPGA를 자동으로 연결시켜준다. FPGA에서 input으로 설정한 pin을 동작시키면서 output이 제대로 나오는지 확인한다.

**2 3-input AND gate (b)**

**** 

****

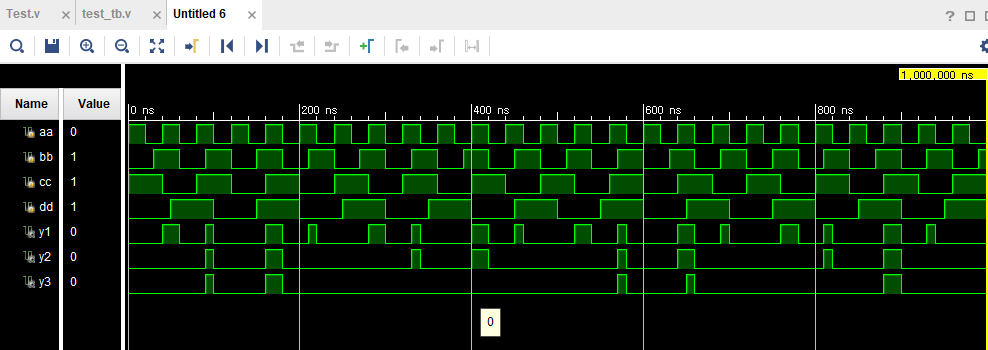
시뮬레이션 결과는 위와 같다. Reg에 dd를 선언해놓고 시뮬레이션을 돌려서 dd의 벨류는 x가 되어버렸다. 이제, 결과를 확인해보면 AND게이트 이므로 aa와 bb가 모두 1일 때, y1이 1이 됨을 알 수 있으며, y1과 cc가 모두 1일 때 y2가 1이 됨을 알 수 있다. 따라서, y2의 경우 나머지 4개 변수의 값이 1이어야 1값을 가지는 것을 알 수 있다.

**3 4-input AND gate (b)**

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

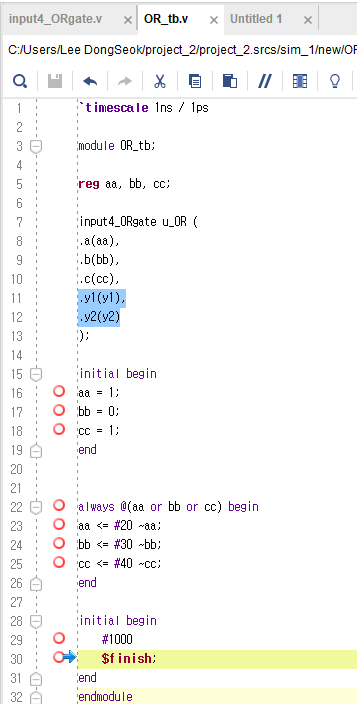
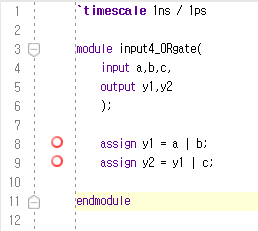
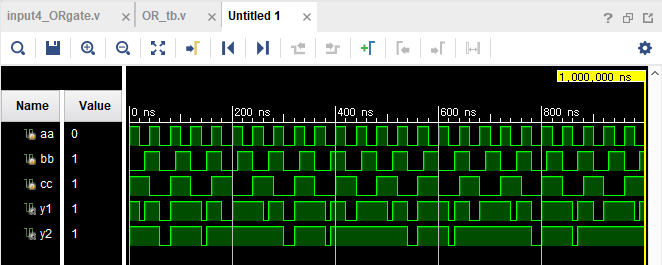
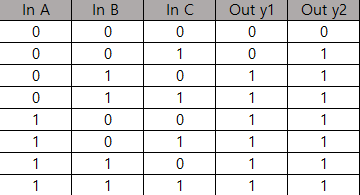


시뮬레이션 결과는 위 그림과 같다. 시뮬레이션을 보면 aa와 bb가 모두 1일 때 y1이 1의 값을 가지게 되며, y2는 y1과 cc가 모두 1일 때 1을가지고 마지막으로 y3는 y2와 dd가 1일 때 1의 값을 가지게 된다. 이외에는 모두 0이 되게 된다. 즉, y3가 1일 때 aa, bb, cc, dd, y1, y2 모두 1의 값을 가진다는 말이며 시뮬레이션으로 쉽게 확인이 가능하다. 진리표는 다음과 같이 작성할 수 있다.

테이블이(가) 표시된 사진

자동 생성된 설명

**4. 3-input OR gate (b)**

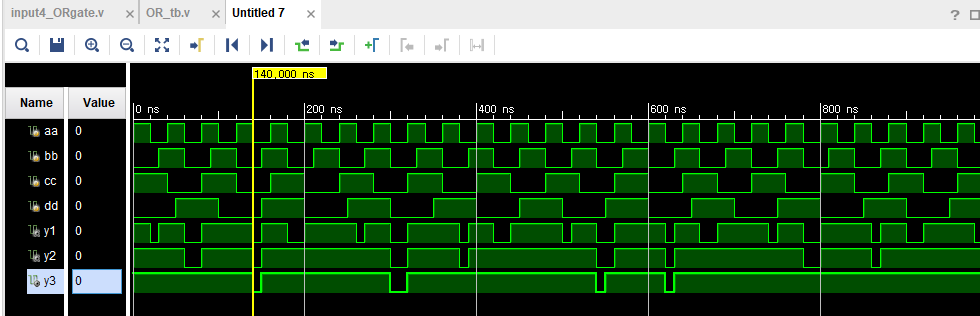
**** ****

AND와는 다른 양상을 띈다. 일단, or 게이트는 알다시피 두 논리값 중 하나만 참(1, high)이어도 1을 출력한다. 따라서, 여기서도 aa가 1이라면, y1과 y2는 항상 1이 됨을 알 수 있다. 또한, aa, bb, cc가 모두 0일 때 y2는 0값을 가진다. y1의 경우엔 당연하게도 aa와 bb가 0일 때 0을 가진다.

**5. 4-input OR gate (b)**

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

시뮬레이션 결과는 위 그림과 같다. 시뮬레이션을 보면 aa와 bb가 모두 0일 때 y1이 0의 값을 가지게 되며, 이를 제외하면 1의 값을 가진다. y2는 y1과 cc가 모두 0일 때 0을가지고 마지막으로 y3는 y2와 dd가 0일 때 0의 값을 가지게 된다. 이외에는 모두 1이 되게 된다. 즉, y3가 0일 때 aa, bb, cc, dd, y1, y2 모두 0의 값을 가진다는 말이며 시뮬레이션으로 쉽게 확인이 가능하다. 진리표는 다음과 같이 작성할 수 있다.

**테이블이(가) 표시된 사진

자동 생성된 설명**

**6. 결과 검토 및 논의 사항**

시뮬레이션을 돌릴 때, always문에 들어가는 시간을 지연시키는 문법 작성시 주의사항이 있다. 초기 작성시엔 모든 변수에 대해 딜레이 되는 시간을 동일하게 작성했다. 하지만, 이럴 경우 시뮬레이션의 결과를 알아보기 힘들다. 변수들은 모두 같은 시간에 0과 1이 바뀌어 스스로가 짠 코드가 맞는지 알기 어렵다. 따라서 이럴 경우 지연되는 시간이 서로소가 되도록 하는 것이 좋다. 직접 시뮬레이션을 해보니 #20, #30 .. 이런식으로 짤 경우 무조건적인 공백이 생기기 때문에 값의 변화를 쉽게 알아차릴 수 있었다. 또한, 2,4,6,8 처럼 최소공배수가 같은 수를 설정하여도 괜찮을 것 같다.

또한, 코드를 작성하고 보통 스키마틱을 확인하게 되는데 실습 때 강의자료와 다르게 나와 코드를 잘못 작성한 줄 알았다. 하지만, 직접 스키마틱을 살펴보니 동일한 회로였다. 같은 회로여도 배치를 살짝 바꿈으로서 복잡해 보일 수 있다.

**7. 추가 이론 조사 및 작성**

Boolean 식을 제대로 알지 못해서 그에 관한 법칙이 궁금해 조사해보았다.

우선 불 대수는 모든 입출력(논리 값)을 이진 수(0,1)로 한다. 논리연산은 AND, OR, NOT이며 여기서 확장하면 NAND, XOR 등의 연산자가 있다. 이런 논리적 관계를 식으로 표현한 것을 Boolean 식으로 부른다. ( ∙을 \*로 대체해서 적었다. '도 마찬가지로 `로 대체해서 정리했다.)

대수로 불리는 만큼 공리와, 법칙이 있다. 공리는 다음과 같다.

1. X = 0 or X = 1
2. 0 \* 0 = 0 , 1 \* 1 = 1
3. 0 + 0 = 0, 1 + 1 = 1
4. 0 \* 1 = 1 \* 0 = 0 , 1 + 0 = 0 + 1 = 1
5. 1` = 0, 0` = 1 ( not을 `을 이용해 표현하기도 한다.)

법칙의 경우엔 집합에서 사용되는 드모르간이나, 일반 대수에서의 교환, 결합, 분배 법칙이 모두 성립한다. 다음은 기본법칙이다.

X + 0 = 0 + X = X

X \* 1 = 1 \* X = X

X + 1 = 1 + X = 1

X + X = X

X \* X = X

X + X` = 1

X \* X` = 0

X`` = X

모두 소개하기에는 너무 많아, 드모르간과 흡수법칙만 정리한다.

드모르간

(X + Y)` = X` \* Y` , ( X \* Y)` = X` + Y`

흡수법칙

X + X \* Y = X

X \* ( X + Y ) = X

X + X` \* Y = X + Y

X \* ( X` + Y ) = X \* Y

**8. 참고문헌**

<https://docs.xilinx.com/r/2021.2-English/ug892-vivado-design-flows-overview/Placement-and-Routing>

[부울 대수란 무엇입니까? - 테코피디아의 정의 (techopedia.com)](https://www.techopedia.com/definition/6234/boolean-algebra)