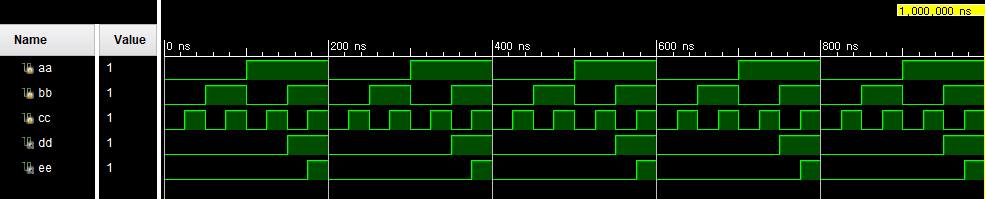
3주차 결과보고서

전공: 컴퓨터공학과 학년: 3학년 학번: 20191612 이름: 윤기웅

**1.**

**Vivado 프로그램에서 프로젝트를 새로 생성하고 이름과 프로젝트 위치를 설정한다. 이후 default part를** xc7a75tfgg484-1로 지정해주고 **디자인 설계 코드와 테스트 코드를 만든다. 그리고 나서 .xdc 파일(constraint file)을 만들어서 입력과 출력 변수에 각각 pin을 설정해주면 된다. Simulation으로 시뮬레이션을 작동시킬 수 있고 Run synthesis와 Implementation을 해주고 Bitstream을 생성하여 회로 기기에 코드를 올려준다. 그런데 실행해주기 전에 프로젝트 폴더의 프로젝트제목\_runs inpl\_1 폴더에다가 debug\_nets.ltx 파일을 등록해주면 된다.**

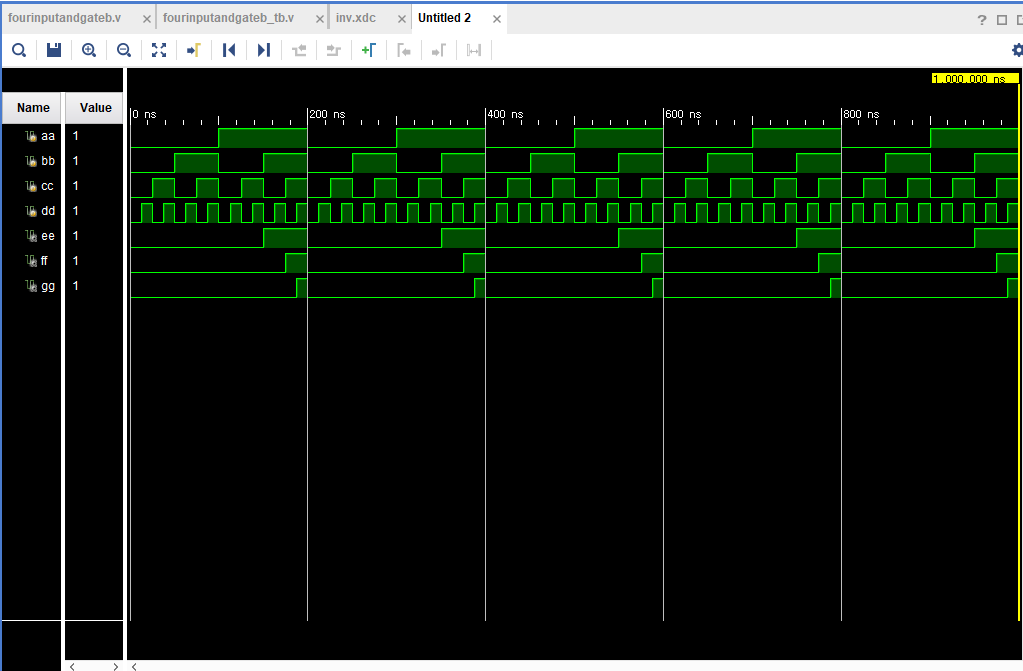
**2.**

****

**3 Input And Gate : and 연산을 하기 때문에 들어오는 값이 모두 1이 아니면 결과값이 0이 된다. 여기의 회로에서도 대부분의 결과값이 0이고 input이 모두 1인 경우에만 결과값 2개 모두 1이 된다. D는 A,B의 결과값이라서 A,B에만 영향을 받는다.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A (input)** | **B (input)** | **C(input)** | **D(output)** | **E(output)** |
| **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **0** | **0** |
| **0** | **1** | **0** | **0** | **0** |
| **0** | **1** | **1** | **0** | **0** |
| **1** | **0** | **0** | **0** | **0** |
| **1** | **0** | **1** | **0** | **0** |
| **1** | **1** | **0** | **1** | **0** |
| **1** | **1** | **1** | **1** | **1** |

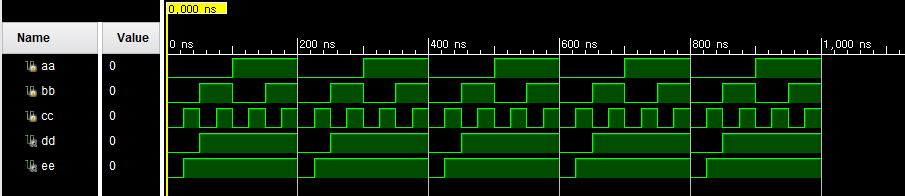
**3.**

****

**4 Input And Gate : E는 A,B의 and 연산의 결과값이고 F는 C와 E의 and연산을 한 값이다. G D 와 F를 and 연산을 실행한 값이다. G는 모든 input이 1인 경우에 1의 값을 갖게 되고 하나라도 0이면 0의 값을 갖는다.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **A (input)** | **B (input)** | **C(input)** | **D(input)** | **E(output)** | **F(output)** | **G(output)** |
| **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **0** | **1** | **0** | **0** | **0** |
| **0** | **0** | **1** | **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **1** | **0** | **0** | **0** |
| **0** | **1** | **0** | **0** | **0** | **0** | **0** |
| **0** | **1** | **0** | **1** | **0** | **0** | **0** |
| **0** | **1** | **1** | **0** | **0** | **0** | **0** |
| **0** | **1** | **1** | **1** | **0** | **0** | **0** |
| **1** | **0** | **0** | **0** | **0** | **0** | **0** |
| **1** | **0** | **0** | **1** | **0** | **0** | **0** |
| **1** | **0** | **1** | **0** | **0** | **0** | **0** |
| **1** | **0** | **1** | **1** | **0** | **0** | **0** |
| **1** | **1** | **0** | **0** | **1** | **0** | **0** |
| **1** | **1** | **0** | **1** | **1** | **0** | **0** |
| **1** | **1** | **1** | **0** | **1** | **1** | **0** |
| **1** | **1** | **1** | **1** | **1** | **1** | **1** |

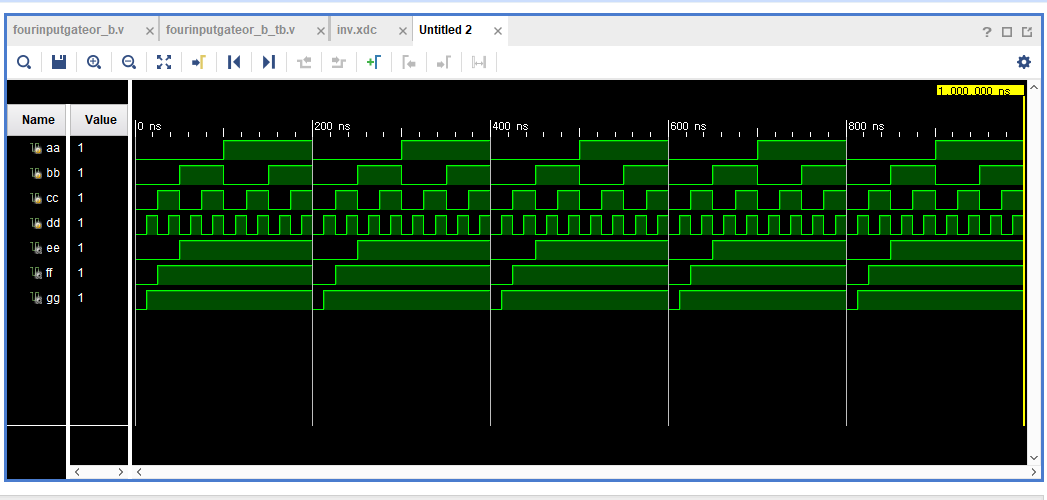
4.



**3 Input or Gate: or연산을 하기 때문에 and연산의 회로보다 1의 결과값이 많은 것을 볼 수 있다. 이전과는 반대로 input 3개의 값이 모두 0이여야지만 결과값이 0이 나오고 하나라도 1이면 or연산으로 인해서 1이라는 결과를 얻는다. D는 A,B의 값에만 관련이 있고 E는 모든 입력값의 영향을 받는다.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A (input)** | **B (input)** | **C(input)** | **D(output)** | **E(output)** |
| **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **0** | **1** |
| **0** | **1** | **0** | **1** | **1** |
| **0** | **1** | **1** | **1** | **1** |
| **1** | **0** | **0** | **1** | **1** |
| **1** | **0** | **1** | **1** | **1** |
| **1** | **1** | **0** | **1** | **1** |
| **1** | **1** | **1** | **1** | **1** |

5.



4input or gate에서 3개의 결과값이 있는데 and gate와는 달리 or 연산을 하기 때문에 모든 입력값이 0이여야지만 결과가 0이 나온다. 시뮬레이션 결과를 보면 결과값들이 초반에만 0이고 그 이후에는 1이 되는 것을 알 수 있다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **A (input)** | **B (input)** | **C(input)** | **D(input)** | **E(output)** | **F(output)** | **G(output)** |
| **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **0** | **1** | **0** | **0** | **1** |
| **0** | **0** | **1** | **0** | **1** | **1** | **1** |
| **0** | **0** | **1** | **1** | **1** | **1** | **1** |
| **0** | **1** | **0** | **0** | **1** | **1** | **1** |
| **0** | **1** | **0** | **1** | **1** | **1** | **1** |
| **0** | **1** | **1** | **0** | **1** | **1** | **1** |
| **0** | **1** | **1** | **1** | **1** | **1** | **1** |
| **1** | **0** | **0** | **0** | **1** | **1** | **1** |
| **1** | **0** | **0** | **1** | **1** | **1** | **1** |
| **1** | **0** | **1** | **0** | **1** | **1** | **1** |
| **1** | **0** | **1** | **1** | **1** | **1** | **1** |
| **1** | **1** | **0** | **0** | **1** | **1** | **1** |
| **1** | **1** | **0** | **1** | **1** | **1** | **1** |
| **1** | **1** | **1** | **0** | **1** | **1** | **1** |
| **1** | **1** | **1** | **1** | **1** | **1** | **1** |

6. FPGA 기기로 연결하기 전에 ltx file을 첨부해주어야 하는데 이는 synthesis를 만들 때 생성된다고는 한다. 그런데 만들어주지 않으면 직접 추가해주어야 한다. 실험을 할 때 사용하는 XC7A75T-2FGG484I는 xilinx에서 만든 제품으로 필드 프로그래밍 지원 게이트 어레이 역할을 한다고 한다. 이는 ARTix-7의 한 종류로 로직 신호 처리나 메모리 인터페이스에 적합한 성능을 갖고 있다고 한다.

7.

-FPGA: 여러 유형의 FPGA가 있고 하드코어의 프로세서가 있는 것은 Soc라고 불린다. FPGA는 인공지능과 머신러닝에 사용되고 어떤 경우는 방사선 내성 장치에 이용되는 등 광범위하게 사용된다. 이 외에도 데이터 센터의 스토리지 , 통신과 자동차 산업에서도 사용된다. xilinx의 제품은 다른 업체의 FPGA보다 다양성과 세밀함을 설계 흐름에 넣었다.