3주차 결과보고서

전공: 컴퓨터공학과 학년: 3학년 학번: 20191599 이름: 송경호

**1.**

Vivado와 Verilog 언어를 통해 FPGA 동작을 검증 가능하다.

**1-1) Verilog Coding**

게이트의 동작을 제어하는 디자인 코드와 테스트벤치 코드를 vivado에서 작성한다. 코드 상의 변수 값은 FPGA의 switch로 구현된다. switch가 켜진 경우를 1로, 꺼진 경우를 0으로 표현한다.

**1-2) Assigning Device**

FPGA 동작을 위해 프로젝트에 device를 할당한다. Vivado 상에서 Settings > Project Device 메뉴를 통해 device를 설정할 수 있다. 컴퓨터공학실험II의 경우 xc7a75tfgg484-1(Artix7) 을 사용한다. Artix7 FPGA는 로직, 신호 처리, 임베디드 메모리 등에 최적화된 성능을 제공한다.

**1-3) Pin Assignment**

Pin은 Verilog 코드의 입력과 출력 변수에 연결된다. .xdc 파일을 생성하여 FPGA의 핀 목록에 제시된 핀을 Verilog 소스의 포트와 매핑한다.

**1-4) Synthesis & Implement / Device Configuration**

Vivado에서 Synthesis > Run Synthesis와 Implementation > Run Implementation 메뉴를 통해 디자인 합성 및 구현을 진행한다. 이 과정을 거쳐 디자인은 FPGA에 맞게 변환된다. 이후에 Program and Debug > Generate Bitstream 및 Open Hardware Manager를 통해 FPGA 디바이스를 구상한다.

**1-5) FPGA 동작**

Open Hardware Manager에서 Auto Connect를 통해 연결한다. 또한 FPGA 하드웨어의 연결을 끝마친다. Project > Project\_runs > impl\_1 폴더에 debug\_nets.ltx 파일을 추가하고 Program and Debug > Program Device > Debug Probes File에서 추가한 debug\_nets.ltx 파일을 선택한다.

**2.**

**스케치, 도표, 그림, 라인이(가) 표시된 사진

자동 생성된 설명**

**2-1) simulation 과정**

(1) **텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

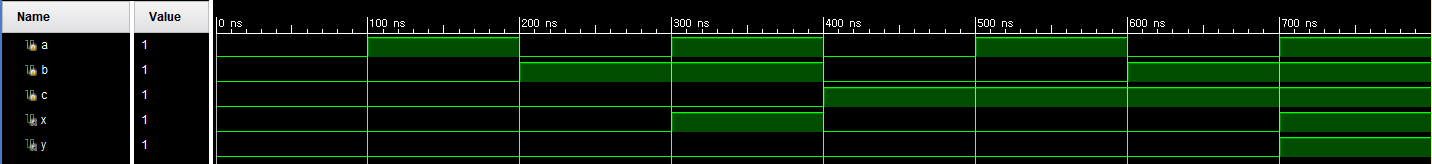
자동 생성된 설명** (2) **텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명** 텍스트, 스크린샷, 폰트, 대수학이(가) 표시된 사진

자동 생성된 설명

(1)이 디자인 코드이며 (2)가 디자인 코드이다. 출력과 입력을 구분하기 위하여 입력을 a,b,c 출력을 x,y로 구분하였다. output x의 경우 a와 b를 입력으로 갖는 AND gate의 출력이며 y는 c와 x를 입력으로 갖는 AND gate의 출력이다. 입력이 세 개이므로 각 입력의 상태를 모두 보기 위해 a, b, c를 각 각 100, 200, 400ns마다 변화를 주었다. 결과적으로 800ns까지 확인하여 모든 상태의 변화를 확인한다.

**2-2) simulation 결과와 진리표**

****

텍스트, 번호, 스크린샷, 낱말맞추기 퍼즐이(가) 표시된 사진

자동 생성된 설명

3 input AND의 경우 세 input이 모두 1인 경우에만 output이 1이 되며, 나머지 경우에는 모두 0인 것을 확인할 수 있다.

**3.**

**스케치, 도표, 그림, 화이트이(가) 표시된 사진

자동 생성된 설명**

**3-1) simulation 과정**

(1) 텍스트, 폰트, 스크린샷이(가) 표시된 사진

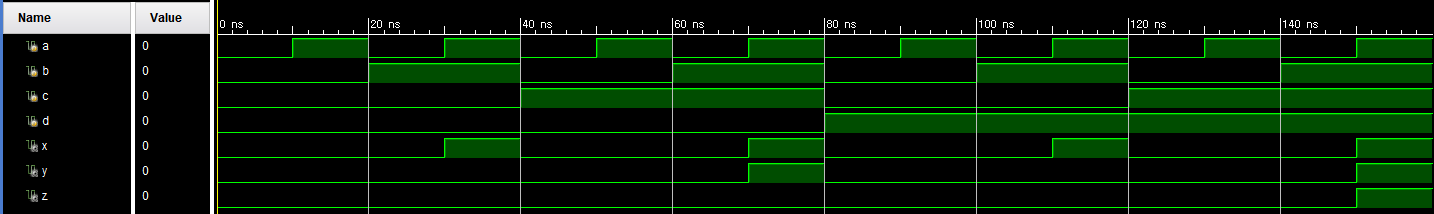
자동 생성된 설명 (2) 텍스트, 폰트, 스크린샷, 친필이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

3 input AND와 동일하게 (1)이 디자인 코드이며 (2)가 디자인 코드이다. 출력과 입력을 구분하기 위하여 입력을 a,b,c,d 출력을 x,y,z로 구분하였다. output x의 경우 a와 b를 입력으로 갖는 AND gate의 출력이며 y는 c와 x를 입력으로 갖는 AND gate의 출력, z는 y와 d를 입력으로 갖는 AND gate의 출력이다. 입력이 네 개이므로 각 입력의 상태를 모두 보기 위해 a, b, c, d를 각 각 10, 20, 40, 80ns마다 변화를 주었다. 결과적으로 160ns까지 확인하여 모든 상태의 변화를 확인한다.

**3-2) simulation 결과와 진리표**

****

텍스트, 번호, 평행이(가) 표시된 사진

자동 생성된 설명

4 input AND의 경우 네 input이 모두 1인 경우에만 output이 1이 되며, 나머지 경우에는 모두 0이 되는 것을 알 수 있다.

**4.**

**도표, 스케치, 그림, 라인이(가) 표시된 사진

자동 생성된 설명**

**4-1) simulation 과정**

(1) 텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

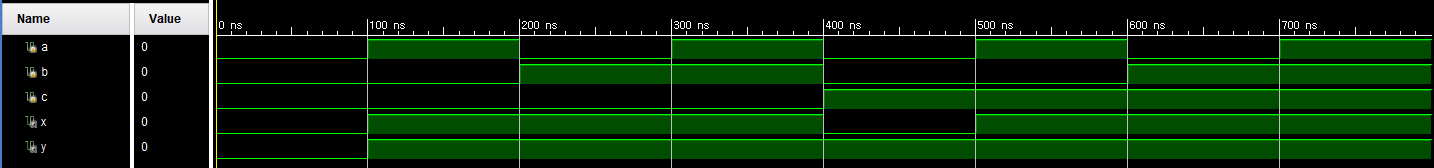
자동 생성된 설명 (2) 텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트, 대수학이(가) 표시된 사진

자동 생성된 설명

3 input AND와 동일하게 (1)이 디자인 코드이며 (2)가 디자인 코드이다. 출력과 입력을 구분하기 위하여 입력을 a,b,c 출력을 x,y로 구분하였다. output x의 경우 a와 b를 입력으로 갖는 OR gate의 출력이며 y는 c와 x를 입력으로 갖는 OR gate의 출력이다. 3 input AND 동일하게 입력이 세 개이므로 각 입력의 상태를 모두 보기 위해 a, b, c를 각 각 100, 200, 400ns마다 변화를 주었다. 결과적으로 800ns까지 확인하여 모든 상태의 변화를 확인한다.

**4-2) simulation 결과와 진리표**

****

텍스트, 번호, 낱말맞추기 퍼즐이(가) 표시된 사진

자동 생성된 설명

3 input OR의 경우 세 input이 모두 0인 경우에만 output이 0이 되며, 나머지 경우에 Out E는 모두 1인 것을 확인할 수 있다.

**5.**

**스케치, 도표, 그림, 라인이(가) 표시된 사진

자동 생성된 설명**

**5-1) simulation 과정**

(1) 텍스트, 폰트, 스크린샷이(가) 표시된 사진

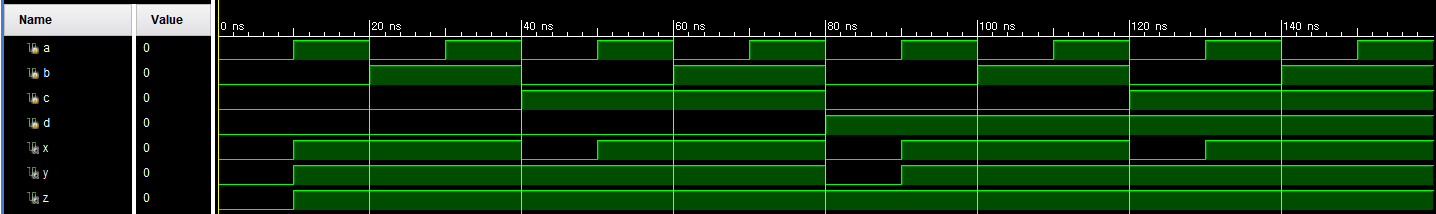
자동 생성된 설명 (2) 텍스트, 폰트, 스크린샷, 친필이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트, 대수학이(가) 표시된 사진

자동 생성된 설명

4 input AND와 동일하게 (1)이 디자인 코드이며 (2)가 디자인 코드이다. 출력과 입력을 구분하기 위하여 입력을 a,b,c,d 출력을 x,y,z로 구분하였다. output x의 경우 a와 b를 입력으로 갖는 OR gate의 출력이며 y는 c와 x를 입력으로 갖는 OR gate의 출력, z는 y와 d를 입력으로 갖는 OR gate의 출력이다. 입력이 네 개이므로 각 입력의 상태를 모두 보기 위해 a, b, c, d를 각 각 10, 20, 40, 80ns마다 변화를 주었다. 결과적으로 160ns까지 확인하여 모든 상태의 변화를 확인한다.

**5-2) simulation 결과와 진리표**

****

텍스트, 번호, 평행, 스크린샷이(가) 표시된 사진

자동 생성된 설명

4 input OR의 경우 네 input이 모두 0인 경우에만 output이 0이 되며, 이를 제외하고 나머지 경우에는 모두 1이 되는 것을 알 수 있다.

**6.**

AND와 OR gate의 input / output 개수를 다르게 하여 시뮬레이션을 시행했다. 결과론적으로 봤을 때, AND gate의 경우 모든 input이 1인 경우에만 output이 1을, OR gate의 경우 모든 input이 0인 경우에만 output이 0을 출력하는 것을 확인했다. 또한 input의 개수에 따라 적절하게 신호 전환 시간을 지정해주어야 직관적으로 output값을 확인할 수 있다.

**7.**

논리 회로의 표현 방법은 크게 세 가지이다.

**7-1) Boolean Algebra**

영국의 수학자 George Boole에 의해 고안된 대수 체계로 논리 회로를 수학적으로 표현할 수 있다. 각 논리 게이트는 수학적 식으로 표시할 수 있다. 예를 들어 ‘x+y’나 ‘abc’와 같은 방식이다. 또한 Boolean Algebra에는 교환 법칙, 결합 법칙, 분배 법칙 등이 존재하여 이를 통해 논리식을 변형할 수 있다. 이 때 각 법칙은 항상 법칙 수행 전과 후가 항상 같은 논리 결과를 갖는다. 예를 들어 ‘a(b+c)’를 분배법칙을 통해 ‘ab+ac’로 표현할 수 있다.

**7-2) Truth Table**

Truth table은 논리 회로의 동작을 나타내는 표이다. 모든 입력 조합에 대한 출력을 열거하는 방식으로 입력이 n인 경우 총 2^n개의 입력 조합이 생긴다. 이에 대한 출력값을 모두 표에 표시하여 입력과 출력간의 관계를 시각화 할 수 있다.

**7-3) Logic Diagram**

논리 회로를 도식화한 것이다. 논리 게이트와 그들 간의 연결을 그림으로 나타낸다. Logic Diagram의 장점은 회로의 구조를 시각화해주는 것으로 이를 통해 설계 및 디버깅이 용이 해진다.