

컴퓨터 공학 기초 실험2 보고서

실험제목: [Lab01] 2 to 1 MUX

실험일자: 2020년 09월 11일 (금)

제출일자: 2020년 09월 17일 (목)

학 과: 컴퓨터공학과

담당교수: 이준환 교수님

실습분반: 금요일 5, 6, 7

학 번: 2019202009

성 명: 서여지

1. 제목 및 목적

A. 제목

2 to 1 Multiplexer

B. 목적

기초적인 베릴로그 문법을 이해한다. 레지스터와 와이어, 게이트 등을 이용하여 간단한 기능을 하는 모듈을 설계한다. 베릴로그에서의 수 표현과 연산자를 이해하고, 응용한다. 쿼터스와 모델심 프로그램의 사용법을 익힌다.

2. 원리(배경지식)

1. 베릴로그

베릴로그는 하드웨어를 표현하는 언어인 HDL의 일종으로, C언어와 파이썬 등의 소프트웨어를 제작하는 프로그래밍 언어와 다르게 하드웨어를 설계하는 것에 사용된다. 특정한 기능을 구성하는 포트와 명령어 등을 모아 모듈을 구성하고, 모듈을 연결하여 논리회로를 제작한다. 논리회로를 모두 제작한 이후 testbench를 적절하게 작성하면 제작한 논리회로에 다양한 입력 값을 넣어 시뮬레이션 결과를 바로 얻을 수 있다.

2. 멀티플렉서

멀티플렉서는 여러 개의 input 신호 중에서 select 신호에 따라 한 가지 신호를 선택하여 output으로 출력하는 논리회로이다. 간단하게 MUX라고 부르기도 한다. 2 to 1 MUX는 두 개의 input을 가진다. 따라서 select는 두 가지 값을 가질 수 있는 1bit 신호이다.

2 to 1 MUX의 진리표와 k-map은 다음과 같다.

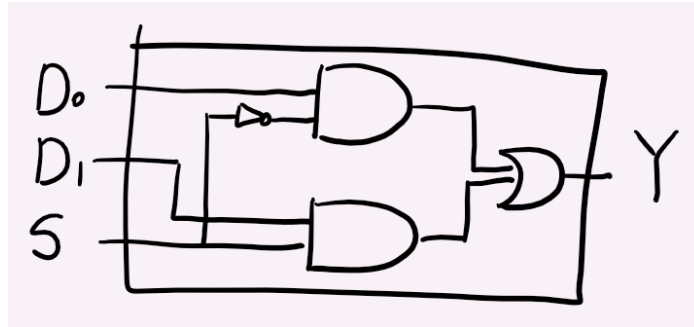
S	D0	D1	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

D1D0 \ S	00	01	11	10
0	0	1	1	0
1	0	0	1	1

따라서 Y는 다음과 같이 정리할 수 있다.

$$Y = D_0\bar{S} + D_1S$$

이 식을 논리회로로 만들면 다음과 같다.



이것은 2 to 1 MUX이지만, 문제에서 제시된 것과 다르다.

3. NAND를 이용해 설계한 MUX

문제에서 제시된 2 to 1 MUX는 세 개의 NAND 게이트와 하나의 인버터로 구성되어있다. 이것은 위에서 설계한 회로의 OR게이트에 드모르간 법칙을 적용하여 유도할 수 있는 형태이다. 따라서 두 회로는 같은 기능을 한다. 두 회로 중 NAND 게이트를 이용한 회로를 더 많이 사용하는 이유는 AND 게이트와 OR 게이트를 함께 이용하는 것 보다 더 경제적 이기 때문이다. 여러 종류의 게이트를 사용하는 것 보다 한 종류의 게이트만 이용하는 것이 유리하고, NAND게이트의 회로집적도가 높기 때문이다.

3. 설계 세부사항

이번 과제는 구체적인 설계의 내용이 주어졌으므로 주어진 조건에 맞게 프로그램을 작성 하였다. MUX를 구성하는 NAND 모듈과 인버터 모듈을 assign과 연산자를 이용하여 만들 었다. 2 to 1 MUX를 구현한 부분인 mx2모듈에서는 필요한 포트와 와이어를 선언한 뒤, 앞 에서 만든 NAND모듈과 인버터 모듈의 인스턴스를 생성하여 사용한다. d0, d1 포트는 MUX의 data input이고, s포트는 select 신호이다. 결과는 y포트를 이용해 출력된다.

S	D0	D1	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

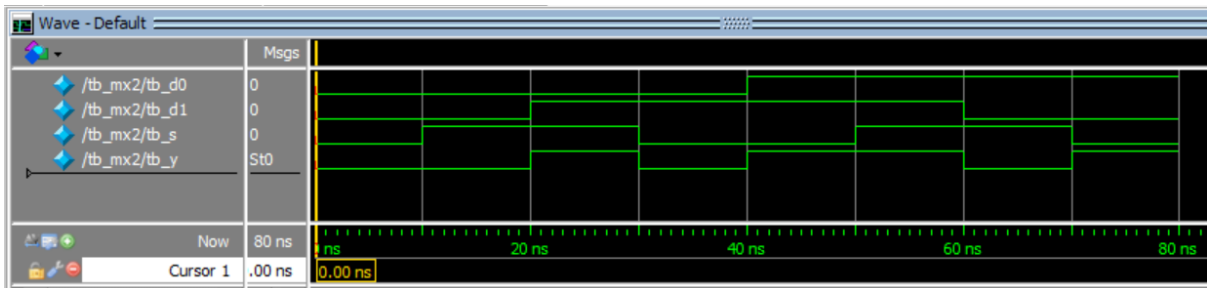
mx2 모듈은 위에서 보였던 2 to 1MUX의 진리표와 같이 동작 해야 한다. 즉, s가 0인 경우 d0의 값을 y로 출력하고, s가 1일 때 d1의 값을 y로 출력해야 한다. Mx2 모듈을 제작하는데 NAND 모듈 3개와 인버터 하나가 사용되었다.

4. 설계 검증 및 실험 결과

A. 시뮬레이션 결과

Testbench의 내용은 다음과 같다. 가장 먼저 사용할 레지스터와 와이어를 선언하고, mx_top이라는 이름의 mx2 모듈의 인스턴스를 생성한다. 이후 레지스터의 모든 값을 0으로 지정하고, 10ns의 간격으로 값을 변경한다. 변경하는 값은 3bit gray code를 이용하여 정했다.

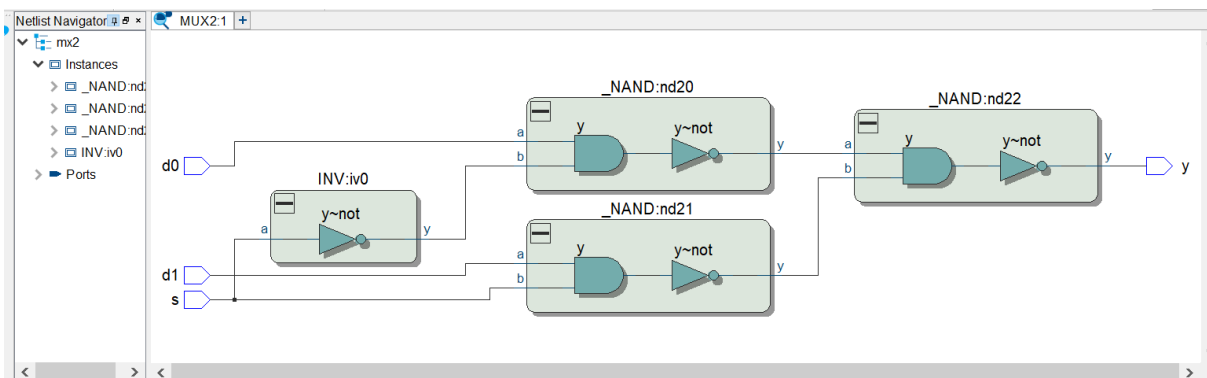
Waveform



시뮬레이션 결과는 위와 같이 나타났다. 진리표에서 나타난 것과 같이 s 값이 0일 때 d0의 값이, s가 1일 때 d1의 값이 그대로 나타나는 것을 볼 수 있다. 모든 경우에 대해서 진리표와 같은 결과가 나타난 것을 알 수 있다.

B. 합성(synthesis) 결과

RTL Viewer



RTL Viewer 화면은 위와 같이 나타났다. Input 레지스터인 d0,d1,s와 y가 회로도에 나타나고, 모듈은 사각형으로 표시되었다. 직접 작성한 NAND모듈이 AND와 인버터로 나누어진 것을 볼 수 있다.

5. 고찰 및 결론

A. 고찰

NAND모듈을 직접 만들어서 사용하는 과정에서 컴파일이 되지 않았다. 라이브러리에 존재하는 nand모듈과 이름이 겹치는 것이 문제였던 것 같다. 직접 만든 모듈의 이름을 _NAND로 변경한 뒤 정상적으로 실행되었다. 또한 모듈에 레지스터를 작성할 때 순서대로 레지스터를 입력하는 방법을 사용했다가 순서를 잘못 입력해서 정상적으로 작동하지 않았었다. 이 문제를 확실히 해결하기 위해 named mapping 방식을 사용하는 것으로 수정하였다.

B. 결론

Quartus 환경에서 베릴로그를 이용하여 기초적인 프로그래밍을 하는 실험이었다. 낯선 환경에서 새로운 언어를 이용하여 문제를 해결하는 전반적인 과정이 흥미로웠다. Quartus와 ModelSim에서 사용할 수 있는 다른 도구도 다양한 것 같다. 어떤 도구가 있는지 궁금하다. 프로젝트를 생성하는 과정을 자료의 설명대로 그대로 따라해서 어떤 것을 의미하는지 모르는 부분이 있었다. 베릴로그 프로그래밍과 Quartus환경에 더 익숙해진 후 그것이 어떤 의미인지 이해하는 시간을 갖고싶다.

6. 참고문헌

이준환교수님, 디지털논리회로2 강의자료, 광운대학교 컴퓨터정보공학과,2020

David Money Harris 외 1인, Digital Design and Computer Architecture, Elsevier, 2013