

컴퓨터 공학 기초 실험2 보고서

실험제목: 2-to-1 MUX

실험일자: 2022년 09월 13일 (화)

제출일자: 2022년 09월 17일 (토)

학 과: 컴퓨터정보공학부

담당교수: 공영호 교수님

실습분반: 화요일 0,1,2

학 번: 2019202050

성 명: 이강현

1. 제목 및 목적

A. 제목

2-to-1 MUX

B. 목적

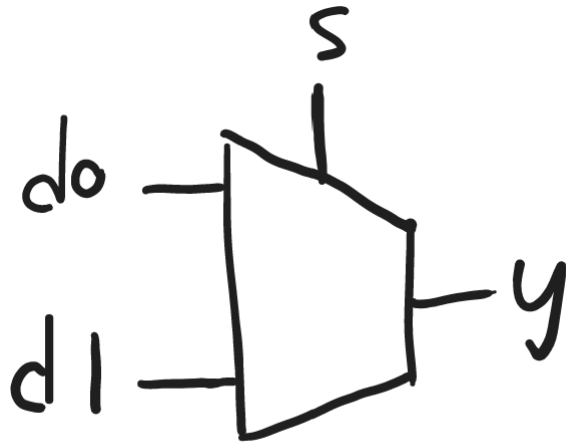
Inverter와 nand게이트를 모듈화시켜 활용할 줄 안다. 모듈에서 다른 모듈을 호출해 사용하여 mux를 설계한다. 설계한 mux를 synthesis, compile하여 코드가 잘 구현되었는지 확인하고 RTL viewer와 testbench를 통해 잘 작동하는지 확인한다.

2. 원리(배경지식)

Verilog란 하드웨어의 기능을 컴퓨터에서 기술하게끔 하는 언어다. 이 언어를 통해 하드웨어의 기능을 확인하고 시뮬레이션 할 수 있다. 구성은 C언어와 비슷하나 end, endmodule, begin등의 구문들을 제외하고 모든 문장의 마지막에는 세미콜론을 붙여야 한다는 특징이 있다. Ctrl + K를 눌러 컴파일을 진행하고 오류가 발생하지 않는다면 testbench의 waveform을 통해 시간의 따른 값의 전달을 확인할 수 있고 RTL viewer를 통해 하드웨어적으로 어떻게 구성되어 있는지 확인이 가능하다. Module, port, net, register를 기본적으로 사용하는데 module은 C언어의 함수나 클래스같이 블록단위로 프로그램을 구성한다. Port는 모듈에서 input, output, inout 중 무엇인지 나타내고 net은 모듈과 모듈사이의 연결 즉 wire를 나타내어 값이 이동은 하나 저장되진 않는다는 특징을 가지고 register는 값의 저장이 필요할 때 사용하며 새로운 값이 할당되기 전엔 기존의 값을 유지한다.

컴파일 후 RTL viewer을 통해 설계된 회로도를 확인하고 시뮬레이션 후에 waveform으로 입력값이 출력값으로 잘 전달되는지를 확인할 수 있다.

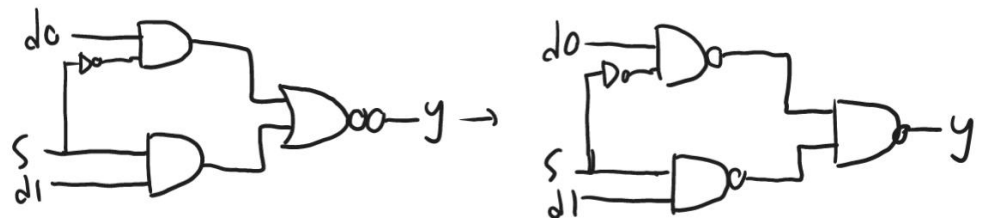
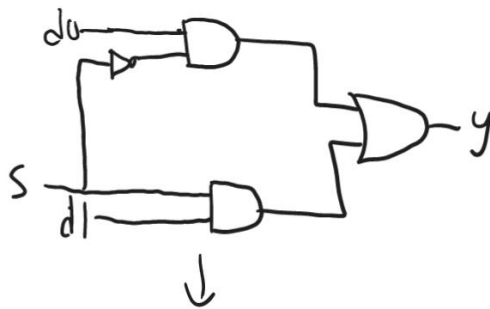
2 to 1 MUX란 두가지 입력중 select에 의해 하나를 출력하는 장치를 말하고 아래와 같은 기호를 사용하며 카르노맵은 아래와 같다.



s \ d _{1:0}	00	01	11	10
0	0	1	1	0
1	0	0	1	1

위 카르노맵을 통해 논리식을 얻으면 $y = \bar{s}d_0 + sd_1$ 가 된다.

그리고 위의 논리식에서 짜여진 회로에서 bubble push를 사용해 2개의 and 1개의 or가 필요한 회로에서 3개의 nand를 필요로 하는 회로로 바꿀 수 있다.



3. 설계 세부사항

mx2.v 구현

mx2.v에서는 gates.v에서 생성한 모듈 nand와 inverter를 입력값 d0,d1,s를 주어 mux가 되게끔 하였다. Nand 게이트 3개와 select에 인버터를 사용하여 구현했다.

gates.v 구현

gates.v에서는 비트단위 연산자를 통해 nand와 inverter를 구현했다.

tb_mx2.v 구현

tb_mx2.v에서는 입력값을 000~111까지 10ns간격으로 주어 확인한다.

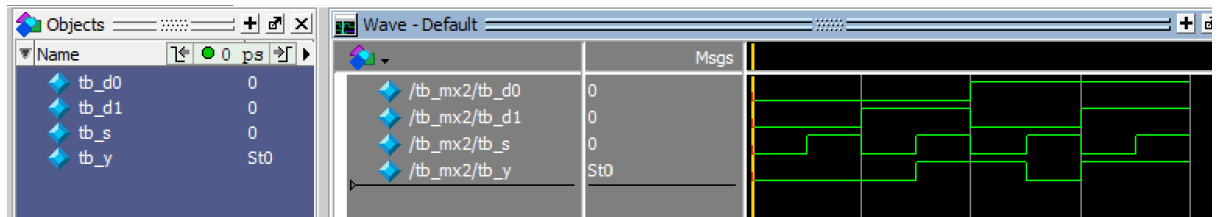
d0	d1	s	sb	w0	w1	y
0	0	0	1	1	1	0
0	0	1	0	1	1	0
0	1	0	1	1	1	0
0	1	1	0	1	0	1
1	0	0	1	0	1	1
1	0	1	0	1	1	0
1	1	0	1	0	1	1
1	1	1	0	1	0	1

<입출력>

위에서 d0,d1 입력값이 s의 값에 따라 d0와 d1이 나올지 결정되는 mux를 구현하였고 입력을 위와 같은 표로 주어 testbench에서 입력값이 잘 전달 되어 출력이 올바르게 나오는지 확인한다.

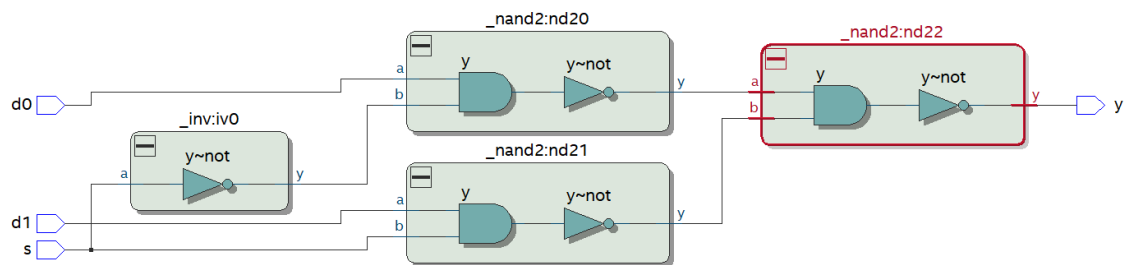
4. 설계 검증 및 실험 결과

A. 시뮬레이션 결과



reg d0,d1,s와 wire y를 선언하여 000~111까지 총 8개의 경우에서 y값이 어떻게 나오는지를 확인할 수 있었다. 위의 입출력 truth table의 값과 동일함으로 보아 잘 구현됨을 알 수 있었다.

B. 합성(synthesis) 결과



RTL viewer를 통해 기존에 구현하려던 MUX와 같은 회로를 얻음을 확인했다.

Flow Summary	
<<Filter>>	
Flow Status	Successful - Sat Sep 17 14:59:47 2022
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	assignment1
Top-level Entity Name	mx2
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	N/A
Total registers	0
Total pins	4
Total virtual pins	0
Total block memory bits	0
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	0

Flow Summary에서는 Flow Status에서 Successful임으로 보아 컴파일이 잘 되었고 이 assignment에 대한 정보들을 확인할 수 있다.

5. 고찰 및 결론

A. 고찰

모듈에서 모듈을 참조할 때 import하는 어떤 조치가 있어야 한다고 생각해서 시간이 좀 걸렸는데 add file을 하면 알아서 쿼터스에서 다른 모듈을 사용할 수 있게끔 해준다는 것을 알았다. 모듈을 참조할 때 현재 파라미터를 그대로 사용하는 것과 불러온 모듈의 파라미터를 점(dot)을 이용하여 사용하는 두가지 방법이 있음을 알았다. 그리고 컴파일은 모듈을 검사하는 거라 성공적으로 error없이 잘 작동하였으나 waveform을 확인하던 도중에 문제가 발생하였는데 모듈의 이름을 잘못 참조하여 생긴 문제였다.

B. 결론

Verilog HDL을 사용하여 하드웨어를 어떻게 설계해야하는지 알고 잘 설계되어있는지 확인하는 방법을 알았다. 쿼터스의 기본적인 사용법을 체득하는 경험이었다. 모듈이름을 작성할 때 nand와 같은 이미 내장되어있는 것들을 커스텀모듈로 사용할 수 없다라는 것을 알게 되었고 컴파일이 잘된다고 해서 올바르게 회로를 구성하였다는 것은 아님을 모듈이름 참조를 실수했던 부분에서 알 수 있었다. 이번 예제에서는 nand와 inverter만을 이용하여 설계했지만 앞으로 XOR, OR등 여러가지 게이트를 직접 구현해볼 수 있고 그를 통해 감가산기나 디코더 같은 회로를 구성해보면 더 능숙하게 모듈 구현을 할 수 있을 것 같다.

6. 참고문헌

공영호 교수님/컴퓨터공학기초실험2/2022