3주차 결과보고서

전공: 기계공학과 학년: 3학년 학번: 20191820 이름: 김형준

**1.**

1-1. Verilog Coding

Vivado 상에서 HDL 언어를 사용하여 회로의 동작과 구조를 기술하는 design source code와 testbench code를 작성한다. Simulation -> Run Simulation을 사용해 작성한 코드가 시뮬레이션 상에서 제대로 동작하는지를 확인할 수 있다.

1-2. Device assignment

실제로 컴퓨터에 연결할 보드(Device)를 FPGA의 Project에 assign한다. 이 실험에서 사용하는 Device는 xc7atfgg484-1이며, Project Manager -> Settings -> Project Device에서

Device명을 검색 후 설정한다.

1-3. Pin assignment

source code상에 정의된 input과 output 신호(변수)를 보드에서 어느 pin과 연결할

지를 결정하는 과정이다. 보드의 장치들의 pin list는 1주차 강의자료 P23~P26에서 확인할

수 있다. Add Sources -> Add or create constraints로 .xdc파일을 생성한 후,

FPGA pin list에서 할당하고 싶은 pin과 Verilog 소스의 port를 링크한다. 이후,

Window -> I/O Ports를 클릭해 하단에서(I/O Ports 메뉴) 설정된 port들의 세부사항을

직접 설정할 수 있다.

1-4. Synthesis / Implementation

Systhesis -> Run Synthesis를 눌러 합성을 먼저 수행하고 (소스코드 컴파일 및 net list 생성), Implementation -> Run Implementation을 눌러 만들어진 net list를 사용하여

place and route 과정을 진행한다.

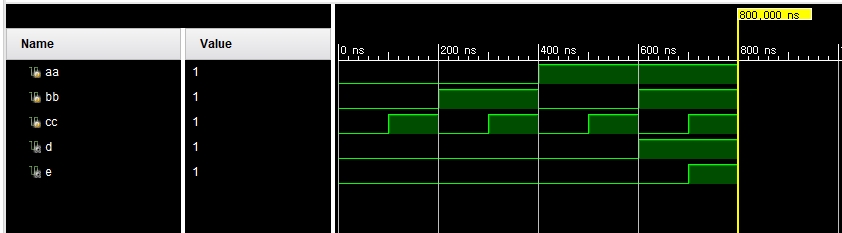
1-5. Device Configuration

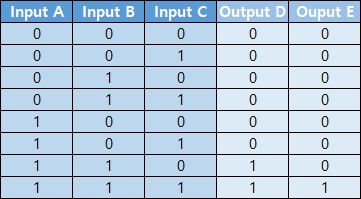
Program and debug -> Generate Bitstream -> Open Hardware Manager를 눌러 비트 스트림 파일을 생성한다.

1-6. FPGA 동작

Open Hardware Manager -> Auto Connect를 눌러서 컴퓨터에 연결된 Device를 불러오고, Program Device -> 연결한 보드(xc7a75t\_0)를 클릭하고 보드에 전송할 비트스트림 파일을 선택한 뒤 Program 버튼을 눌러 FPGA를 작성했던 코드대로 동작시킨다.

**2.**

[3-input AND gate의 simulation 결과]

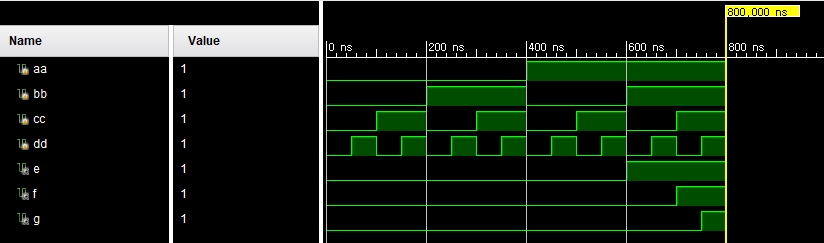


[3-input AND gate의 진리표]

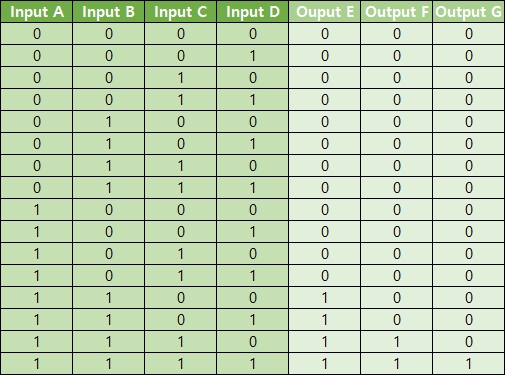
3-input AND gate에서 입력으로 A, B, C를, 출력으로 D, E를 갖는다.

D는 A와 B의 AND 연산의 결과이다. (D=A&B) 따라서 D는 A, B가 모두 1일때만 1을 갖고, 나머지는 0을 갖게 된다. E는 D와 C의 AND 연산의 결과이다. (E=D&C) 따라서 E는 D, C가 모두 1일때만 1을 갖고, 나머지는 0을 갖게 된다. 결과적으로 E는 A,B,C 모두 1일때만 1을 갖고, 나머지는 0을 갖게 된다.

**3.**



[4-input AND gate의 simulation 결과]

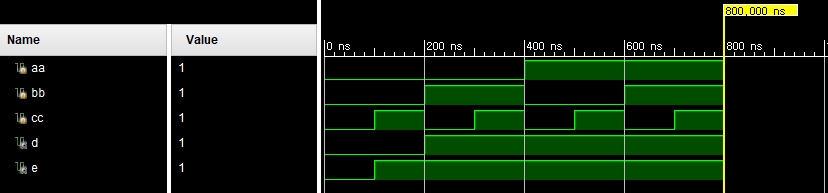


[4-input AND gate의 진리표]

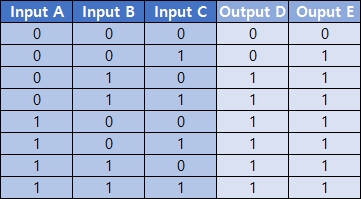
4-input AND gate에서 입력으로 A, B, C, D를, 출력으로 E, F, G를 갖는다.

E는 A와 B의 AND 연산의 결과이다. (E=A&B) 따라서 E는 A, B가 모두 1일때만 1을 갖고, 나머지는 0을 갖게 된다. F는 E와 C의 AND 연산의 결과이다. (F=E&C) 따라서 F는 E, C가 모두 1일때만 1을 갖고, 나머지는 0을 갖게 된다. G는 F와 D의 AND 연산의 결과이다. (G=F&D) 따라서 G는 F, D가 모두 1일때만 1을 갖고, 나머지는 0을 갖게 된다. 결과적으로 G는 A,B,C,D 모두 1일때만 1을 갖고, 나머지는 0을 갖게 된다.

**4.**



[3-input OR gate의 simulation 결과]

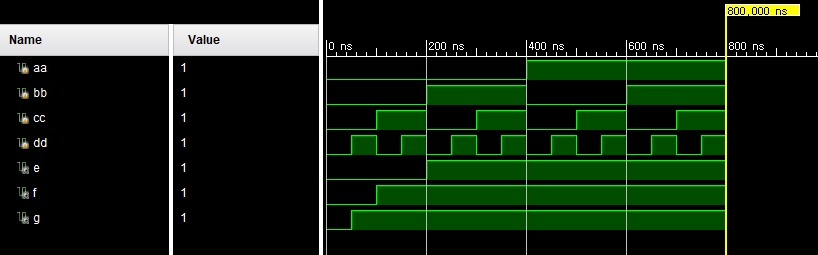


[3-input OR gate의 진리표]

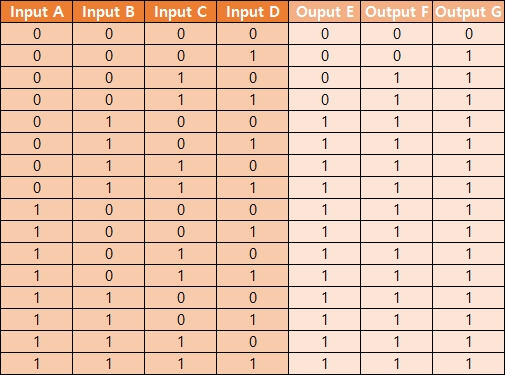
3-input OR gate에서 입력으로 A, B, C를, 출력으로 D, E를 갖는다.

D는 A와 B의 OR 연산의 결과이다. (D=A|B) 따라서 D는 A, B가 모두 0일때만 0을 갖고, 나머지는 1을 갖게 된다. E는 D와 C의 OR 연산의 결과이다. (E=D|C) 따라서 E는 D, C가 모두 0일때만 0을 갖고, 나머지는 1을 갖게 된다. 결과적으로 E는 A,B,C 모두 0일때만 0을 갖고, 나머지는 1을 갖게 된다.

**5.**



[4-input OR gate의 simulation 결과]



[4-input OR gate의 진리표]

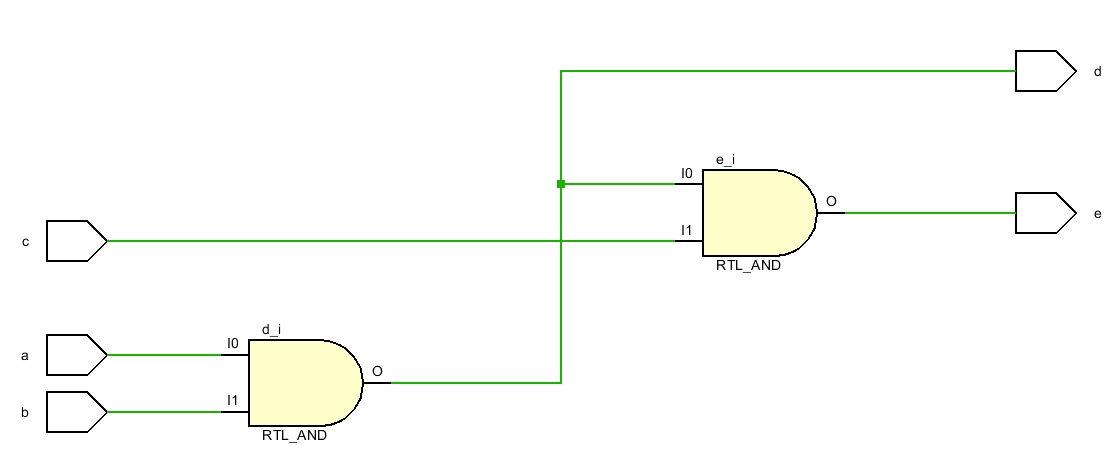
4-input OR gate에서 입력으로 A, B, C, D를, 출력으로 E, F, G를 갖는다.

E는 A와 B의 OR 연산의 결과이다. (E=A|B) 따라서 E는 A, B가 모두 0일때만 0을 갖고, 나머지는 1을 갖게 된다. F는 E와 C의 OR 연산의 결과이다. (F=E|C) 따라서 F는 E, C가 모두 0일때만 0을 갖고, 나머지는 1을 갖게 된다. G는 F와 D의 OR 연산의 결과이다. (G=F)D) 따라서 G는 F, D가 모두 0일때만 0을 갖고, 나머지는 1을 갖게 된다. 결과적으로 G는 A,B,C,D 모두 0일때만 0을 갖고, 나머지는 1을 갖게 된다.

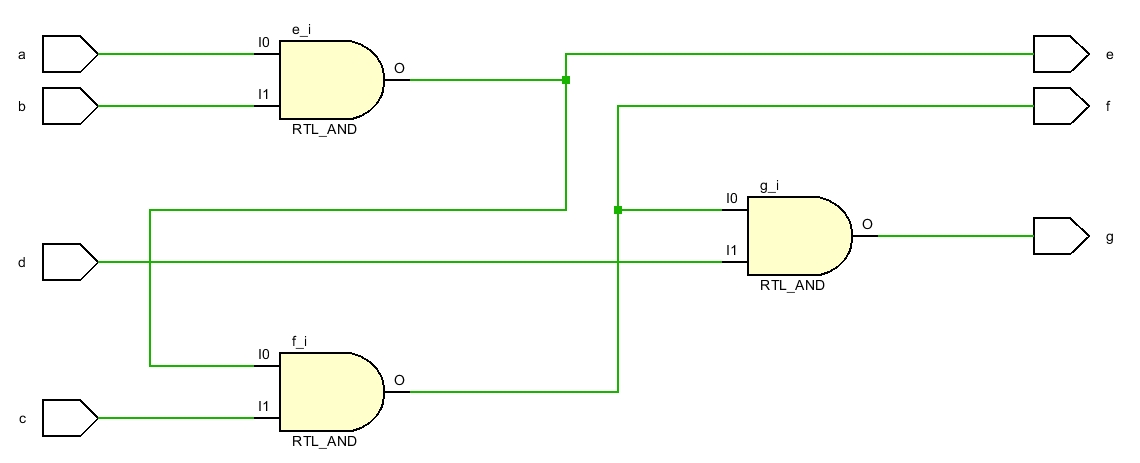
**6.**

2. ~ 5.까지 각각의 시뮬레이션 결과와 진리표를 비교했을 때 차이가 없으므로 의도한 대로 코드가 작성되었음을 확인 할 수 있다. 이 실험을 통해 다중입력 AND 게이트는 모든 입력이 1일때만 1을 반환하고 아닌 경우에는 0을 반환하는 것을 알 수 있었고, 다중입력 OR 게이트는 모든 입력이 0일때만 0을 반환하고 아닌 경우에는 1을 반환하는 것을 알 수 있었다.

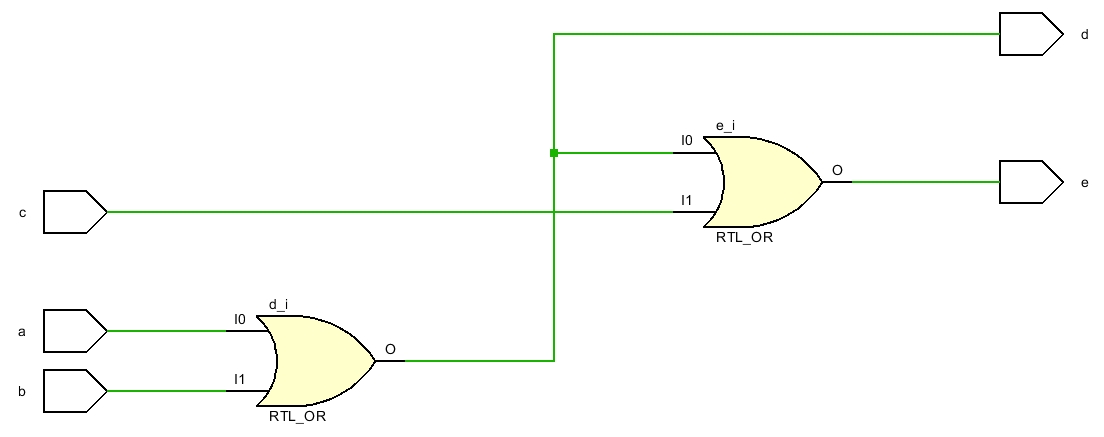
아래의 사진들은 2. ~ 5. 각각의 Schematic이다.

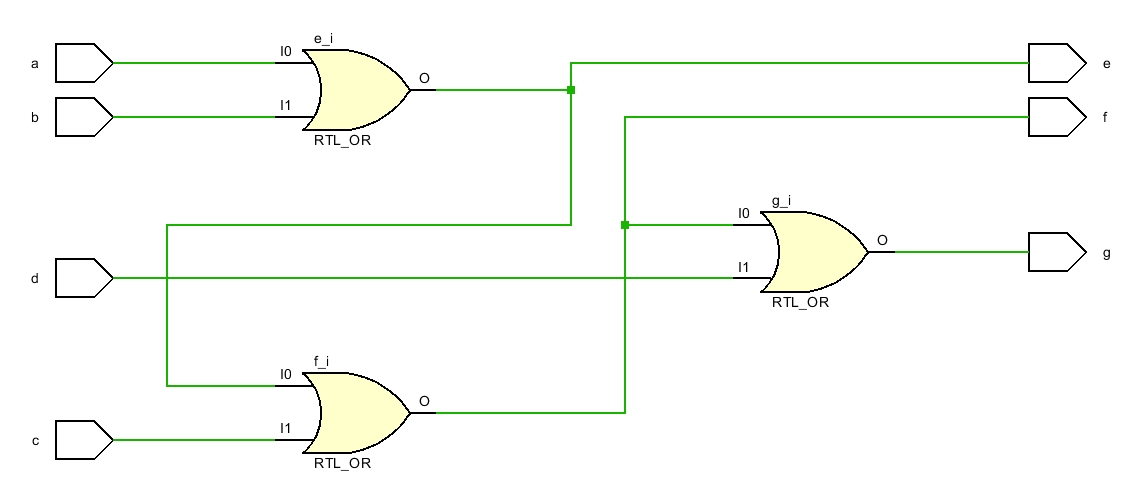


[3-input AND gate의 schematic]



[4-input AND gate의 schematic]

[3-input OR gate의 schematic]



[4-input OR gate의 schematic]

**7.**

다중입력 NAND 게이트는 입력이 2개 이상인 NOR 게이트로, Boolean 식으로는

으로 나타낼 수 있다. 모든 입력이 1인 경우에만 0을, 아닌 경우에는 1을

반환한다. 같은 입력을 가지는 다중입력 AND게이트의 보수(complement)이다.

다중입력 NOR 게이트는 입력이 2개 이상인 NOR 게이트로, Boolean 식으로는

으로 나타낼 수 있다. 모든 입력이 0인 경우에만 1을, 아닌 경우에는 0을

반환한다. 같은 입력을 가지는 다중입력 OR게이트의 보수(complement)이다.

다중입력 XOR 게이트는 입력이 2개 이상인 XOR 게이트로, Boolean 식으로는

으로 나타낼 수 있다. 입력 중 1의 개수가 홀수일 때 1을, 짝수일 때 0을 반환한다.

다중입력 XNOR 게이트는 입력이 2개 이상인 XNOR 게이트로, Boolean 식으로는

으로 나타낼 수 있다. 입력 중 1의 개수가 짝수일 때 1을, 홀수일 때 0을 반환한다. 같은 입력을 가지는 다중입력 XOR게이트의 보수(complement)이다.