

3주차 결과보고서

전공: 컴퓨터공학 학년: 2학년 학번: 20191629 이름: 이주현

1. FPGA 동작법을 설명하시오.

비록 Verilog는 FPGA 또는 ASIC을 설계하고 검증하기 위해 만들어진 언어이지만, 서로 다른 종류의 FPGA는 가지고 있는 핀의 개수, 또는 사용하고 있는 논리 소자의 구조가 서로 다르다. 따라서, Verilog 코드를 합성해서 FPGA에 올리려면 목표 FPGA 칩에 맞게 설정을 조절해야 한다.

먼저 합성 컴파일러에게 목표 FPGA 칩이 무엇인지 알려주어야 한다. 만약 올바르게 않은 칩을 선택한다면, 목표 FPGA에 최적화되지 않은 결과물이 나올 가능성이 있을 뿐만 아니라, 아예 업로드가 불가능한 결과물이 나올 가능성이 있다. 우리가 사용하는 Libero FPGA Starter Kit III는 Xilinx의 Artix-7 계열 FPGA인 xc7a75tfgg484를 탑재하고 있다.

그 다음, Verilog에서 정의한 입력 또는 출력 핀을 FPGA 보드에 존재하는 물리적 핀과 연결하는 과정이 필요하다. 보드 패키지의 설명서를 참고하여 보드의 핀과 Verilog의 핀을 연결해야 하는데, 다음 명령을 사용하면 Verilog의 포트와 보드의 핀을 연결할 수 있다.

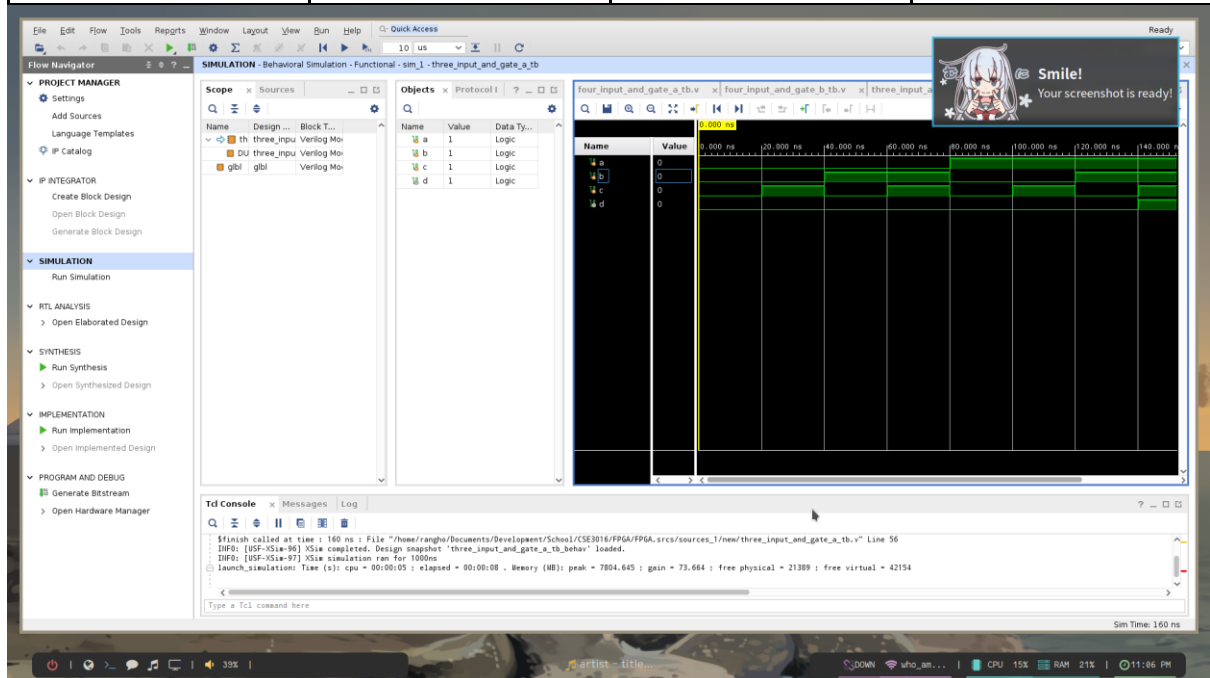
```
set_property -dict {PACKAGE_PIN <pin name> IOSTANDARD LVCMOS33}  
[get_ports <port name>]
```

마지막으로, FPGA 보드와 컴퓨터를 연결하고 미리 합성해 둔 비트스트림 파일을 IDE를 통해 FPGA에 업로드하면 된다.

2. 3-input AND gate의 simulation 결과 및 과정에 대해서 설명하시오.

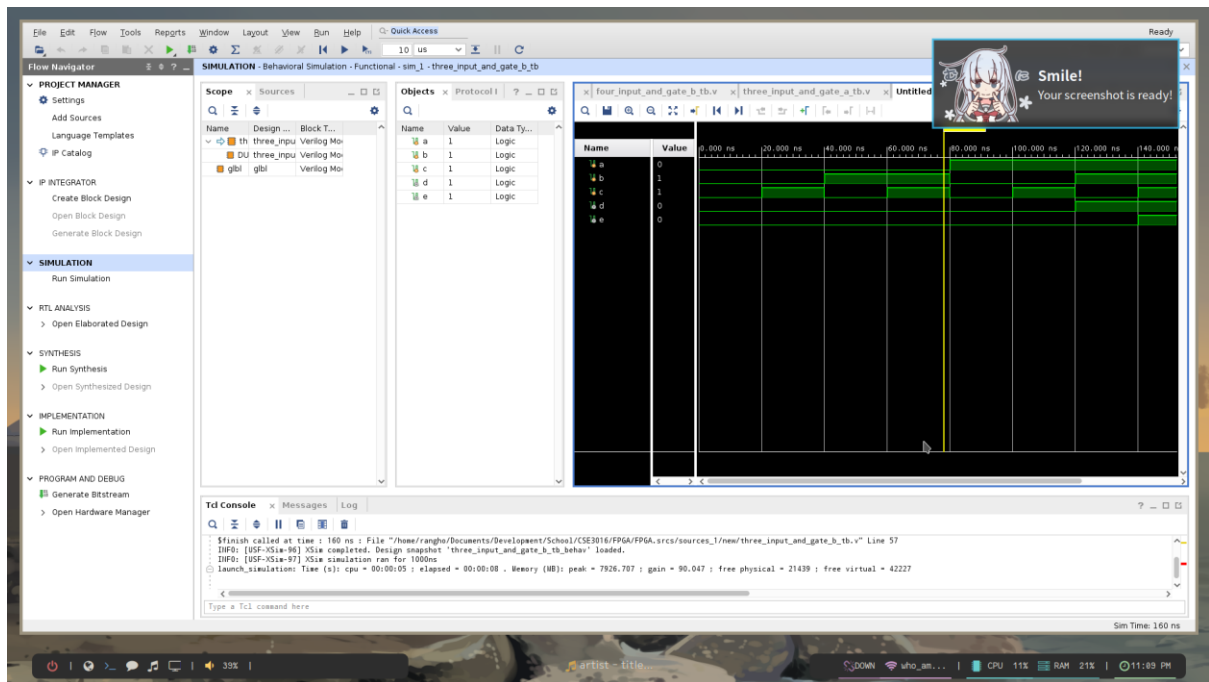
Input A	Input B	Input C	Output D
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0

1	1	0	0
1	1	1	1



(A)의 진리표와 시뮬레이션 결과

Input A	Input B	Input C	Output D	Output E
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



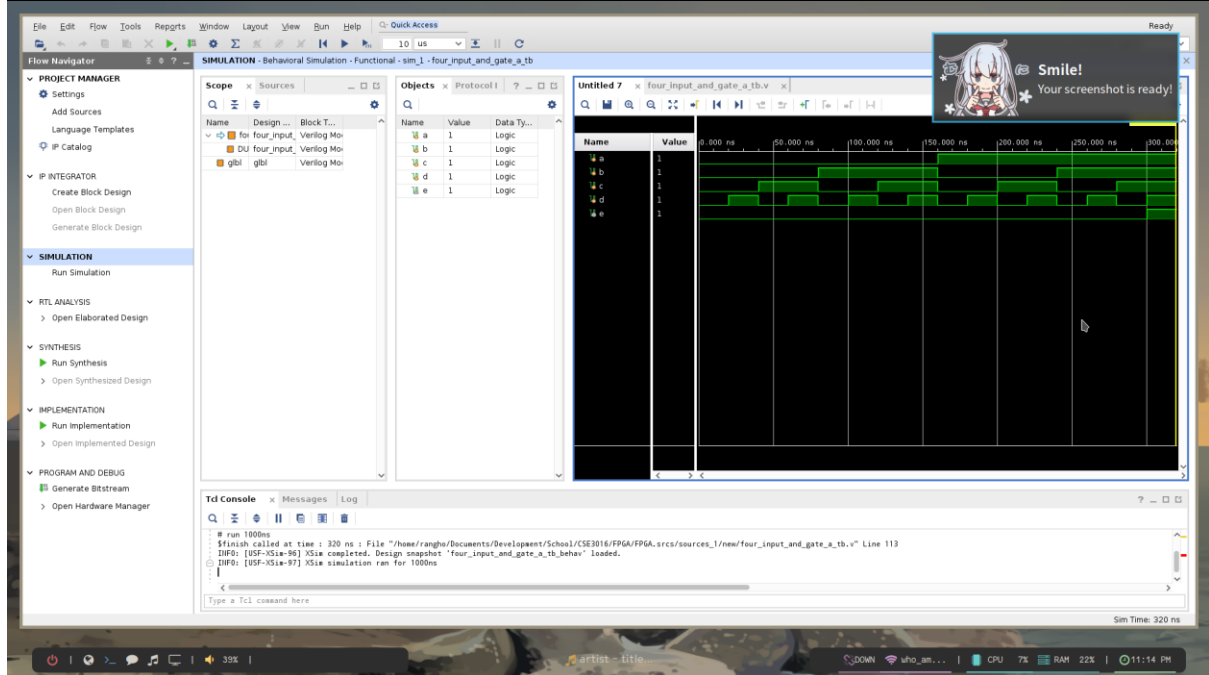
(B)의 진리표와 시뮬레이션 결과

(A)의 경우는 A & B & C와 같이 세 개의 입력 신호를 한꺼번에 연결하였고, (B)의 경우는 A와 B를 먼저 엮은 다음, 그 결과인 D를 C와 엮었다. 두 경우 모두 (중간 단계 출력이 있다는 점을 제외하면) 동작은 완전히 동일했다.

3. 4-input AND gate의 simulation 결과 및 과정에 대해서 설명하시오.

Input A	Input B	Input C	Input D	Output E
0	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	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0

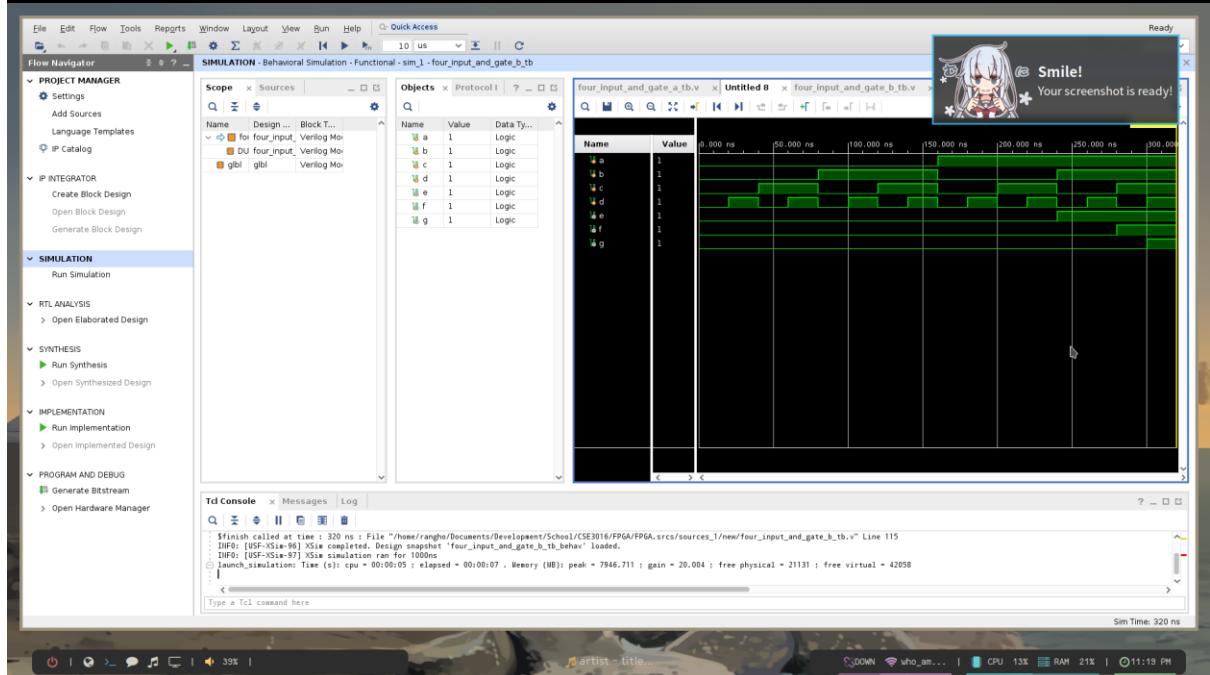
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1



(A)의 진리표와 시뮬레이션 결과

Input A	Input B	Input C	Input D	Output E	Output F	Output G
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



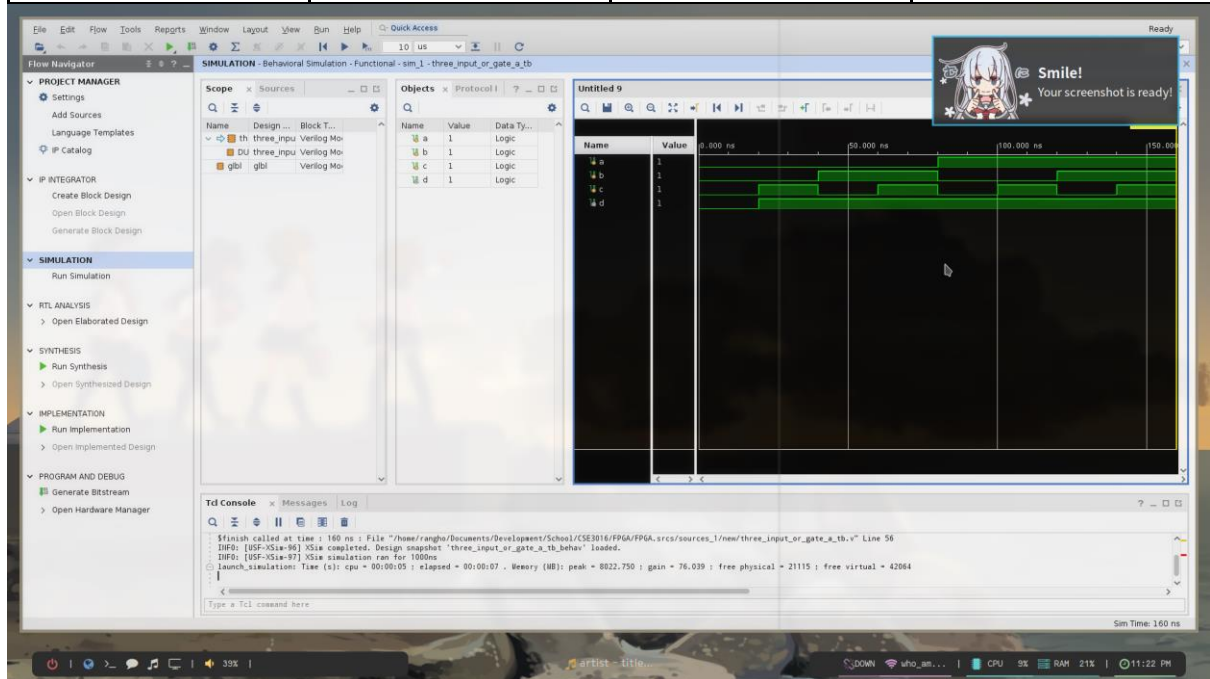
(B)의 진리표와 시뮬레이션 결과

위와 같이 (A)의 경우는 A & B & C & D와 같이 네 개의 입력 신호를 한꺼번에 연결하였고, (B)의 경우는 A와 B를 먼저 엮어 E에 저장한 다음, E와 C를 엮어 F에 저장하고, 마지막으로 F와 D를 엮어 마지막으로 출력한다. 두 경우 모두 (중간 단계 출력이 있다는 점을 제외하면) 동작은 완전히 동일했다.

4. 3-input OR gate의 simulation 결과 및 과정에 대해서 설명하시오.

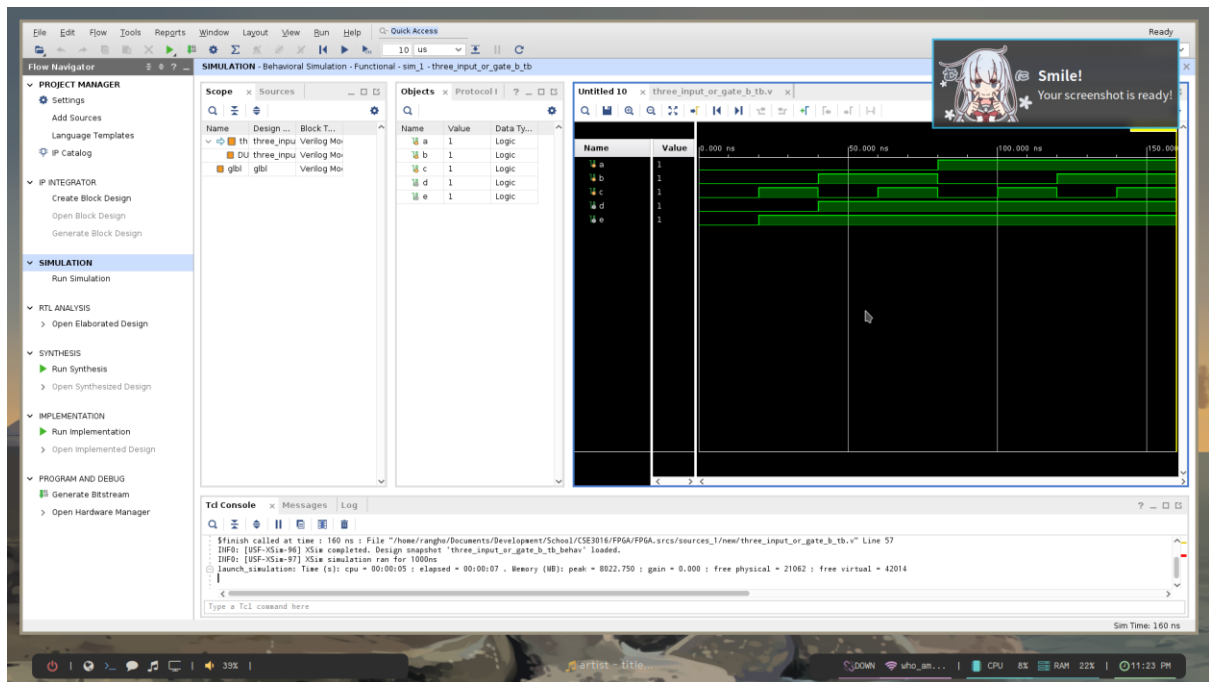
Input A	Input B	Input C	Output D
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1

1	0	1	1
1	1	0	1
1	1	1	1



(A)의 진리표와 시뮬레이션 결과

Input A	Input B	Input C	Output D	Output E
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



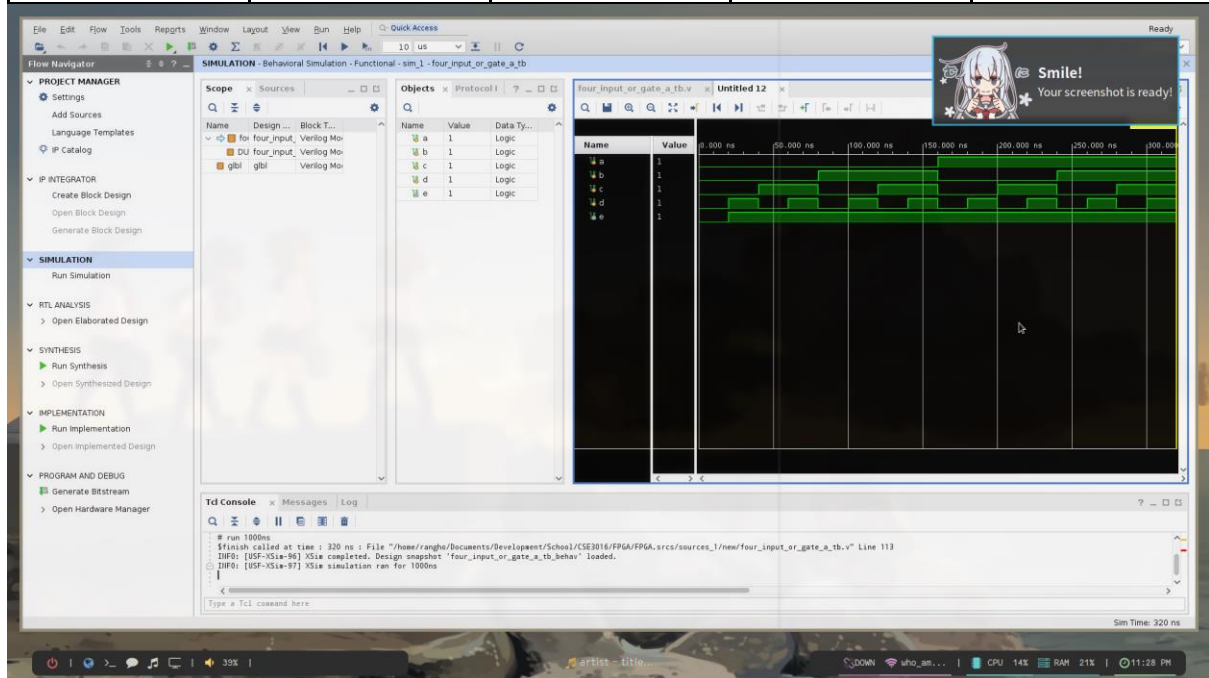
(B)의 진리표와 시뮬레이션 결과

(A)의 경우는 $A | B | C$ 와 같이 세 개의 입력 신호를 한꺼번에 연결하였고, (B)의 경우는 A와 B를 먼저 엮은 다음, 그 결과인 D를 C와 엮었다. 두 경우 모두 (중간 단계 출력이 있다는 점을 제외하면) 동작은 완전히 동일했다.

5. 4-input OR gate의 simulation 결과 및 과정에 대해서 설명하시오.

Input A	Input B	Input C	Input D	Output E
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1

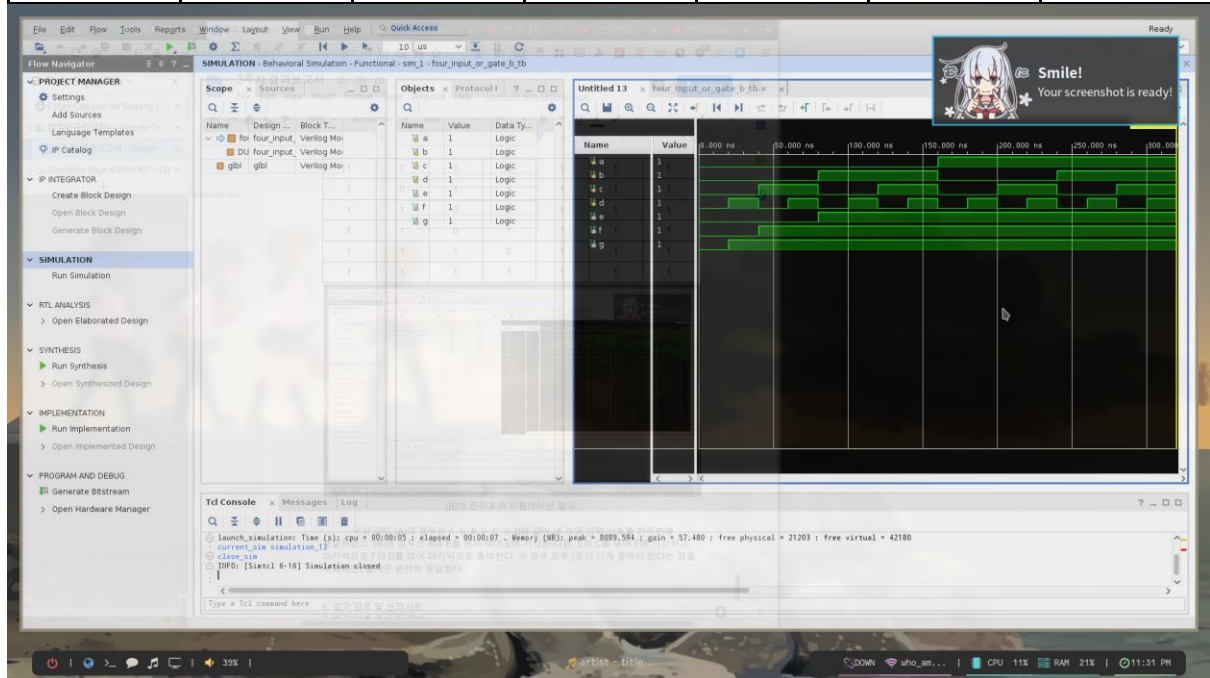
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



(A)의 진리표와 시뮬레이션 결과

Input A	Input B	Input C	Input D	Output E	Output F	Output G
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	1	0	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



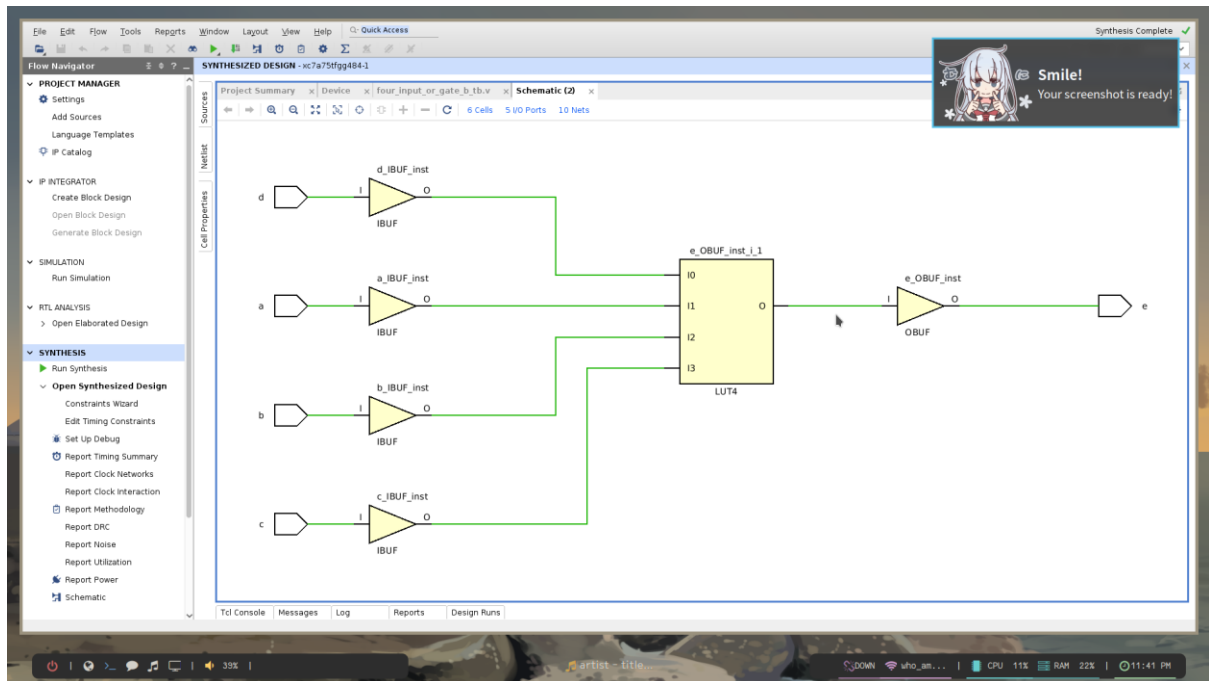
(B)의 진리표와 시뮬레이션 결과

위와 같이 (A)의 경우는 $A \mid B \mid C \mid D$ 와 같이 네 개의 입력 신호를 한꺼번에 연결하였고, (B)의 경우는 A와 B를 먼저 엮어 E에 저장한 다음, E와 C를 엮어 F에 저장하고, 마지막으로 F와 D를 엮어 마지막으로 출력한다. 두 경우 모두 (중간 단계 출력이 있다는 점을 제외하면) 동작은 완전히 동일했다.

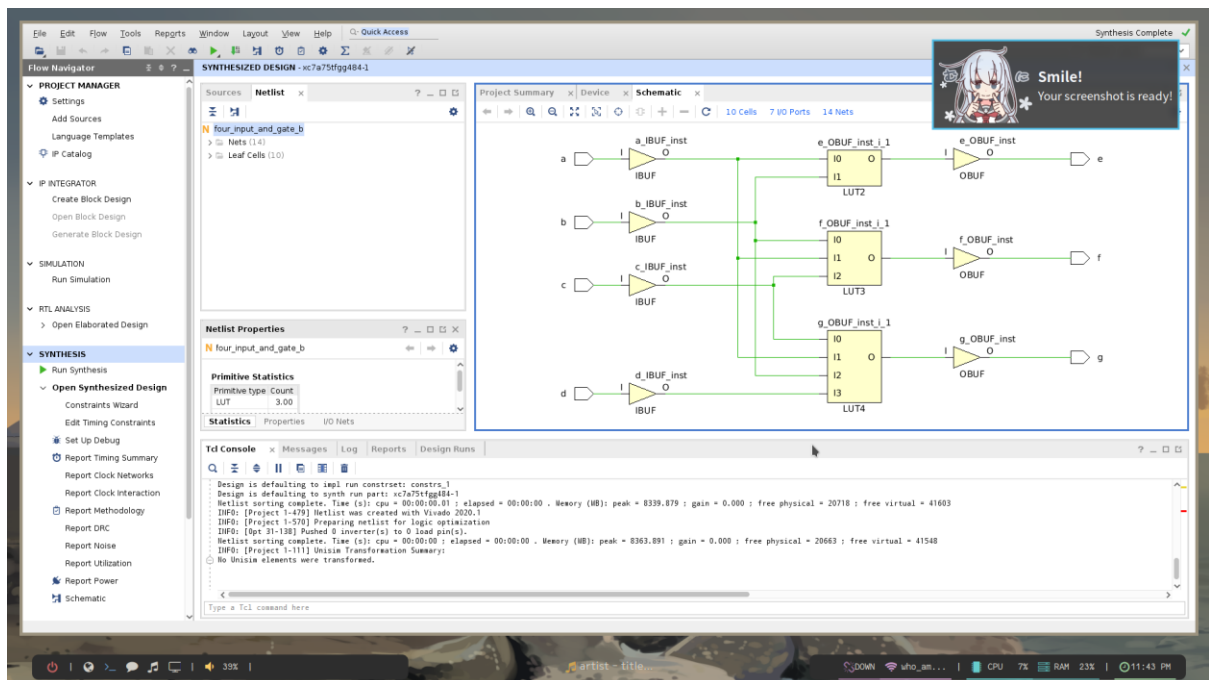
6. 결과 검토 및 논의사항

AND와 OR 게이트 모두 하나의 논리 게이트에 여러 입력을 넣는 것과 각 입력을 두 개씩 나누어 차례로 게이트를 통과하도록 만드는 것 모두 같은 결과를 내는 것을 확인할 수 있었다. 이번 실험은 Verilog의 시뮬레이션 기능을 익힘과 동시에, 논리곱과 논리합의 결합법칙을 눈으로 확인할 수 있는 실험이었다고 생각한다.

7. 추가 이론 조사 및 작성



만약 A와 같이 모든 입력 신호를 하나로 묶게 될 경우 위와 같이 모든 입력이 하나의 LUT로 연결되어 한 번에 연산이 수행되게 된다.



반면 B와 같이 입력 신호를 차례차례 다르게 하게 되면 LUT가 많이 소모될 뿐만 아니라, 하나의 신호가 여러 논리 소자의 입력으로 사용되고 있다.

이런 식의 회로 디자인은 신호의 신뢰성을 떨어뜨릴 수 있고, 전파 지연 시간이 늘어나는 요인이 되기 때문에 중간 출력이 필요하지 않은 이상은 A와 같이 회로를 디자인하는 것이 더 효율적이다.