3주차 결과보고서

전공 : 컴퓨터공학과 학년 : 2학년 학번 : 20201597 이름 : 신동준

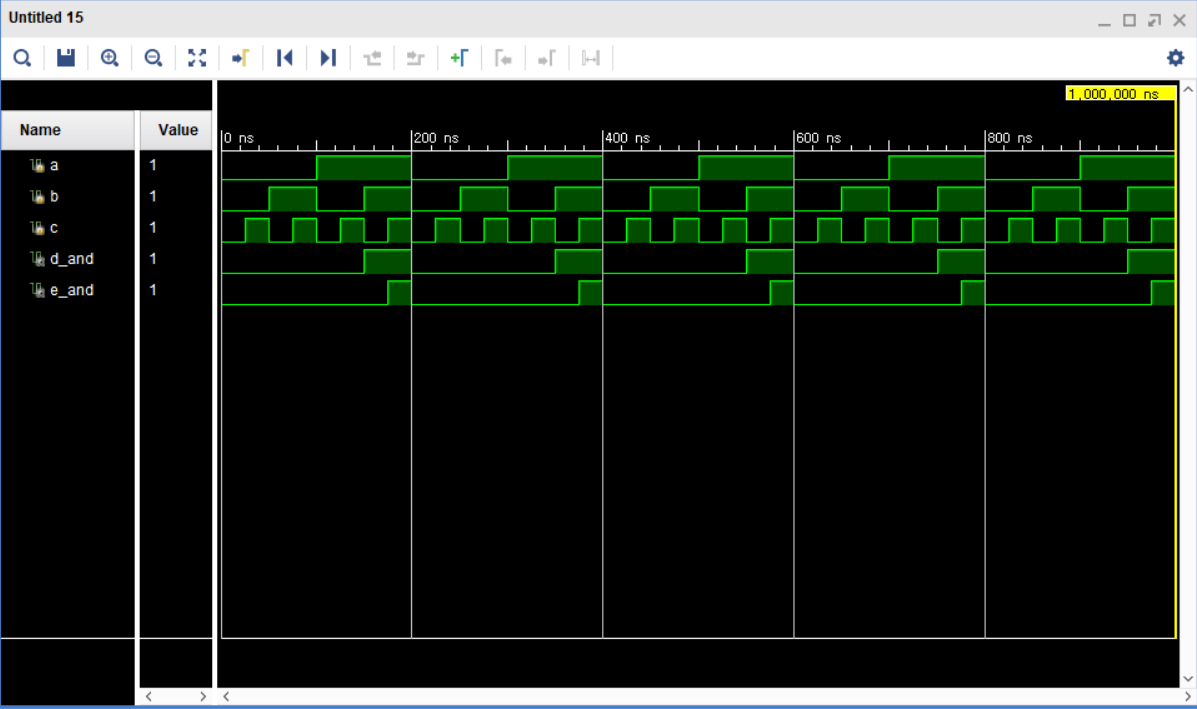
1. **FPGA 동작법을 설명하시오.**

Verilog로 코딩을 하고, run synthesis를 해준다. 이후 프로젝트의 device를 설정해주고, pin을 특정해준다. 이때 Verilog 코드 상의 port와 fpga의 pin을 link해준다. Device와 본인의 컴퓨터 파일을 connect해주고, project를 fpga에 넣어준다.

1. **3-input AND gate의 simulation 결과 및 과정에 대해서 설명하시오.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| In A | In B | In C | Out D | Out 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 |

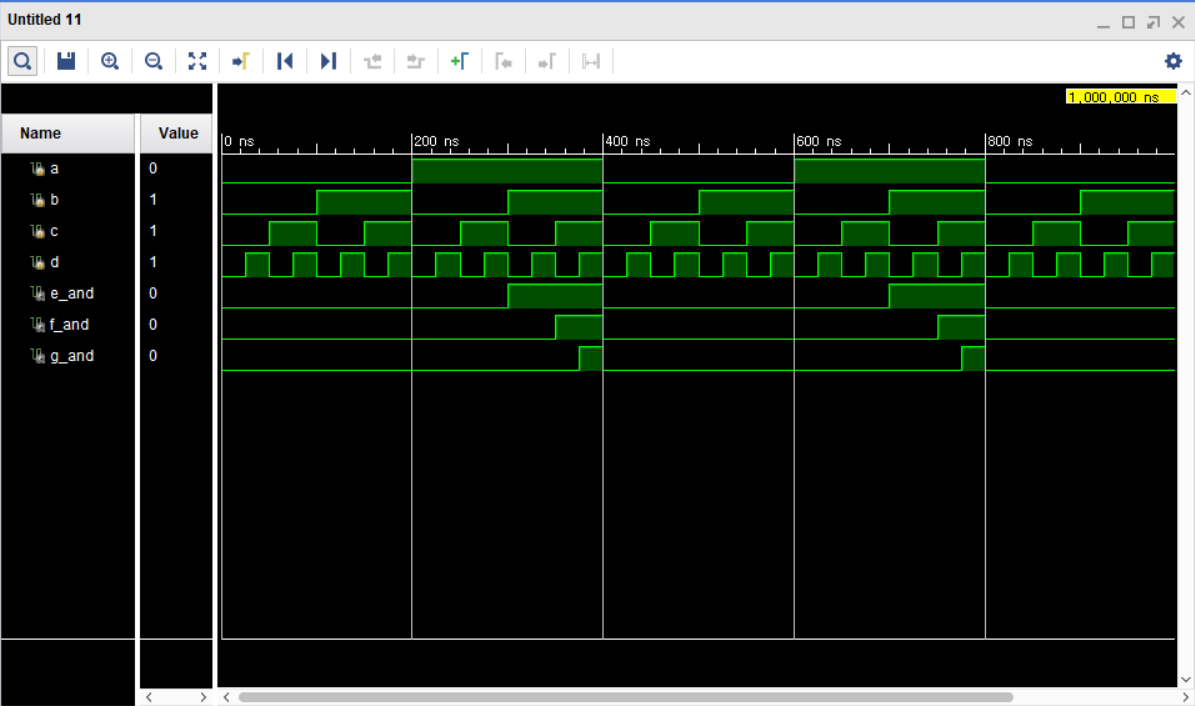
D를 먼저 and 연산으로 계산해주고, C, D를 다시 and 연산으로 E를 구한다.



1. **4-input AND gate의 simulation 결과 및 과정에 대해서 설명하시오.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | 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 |

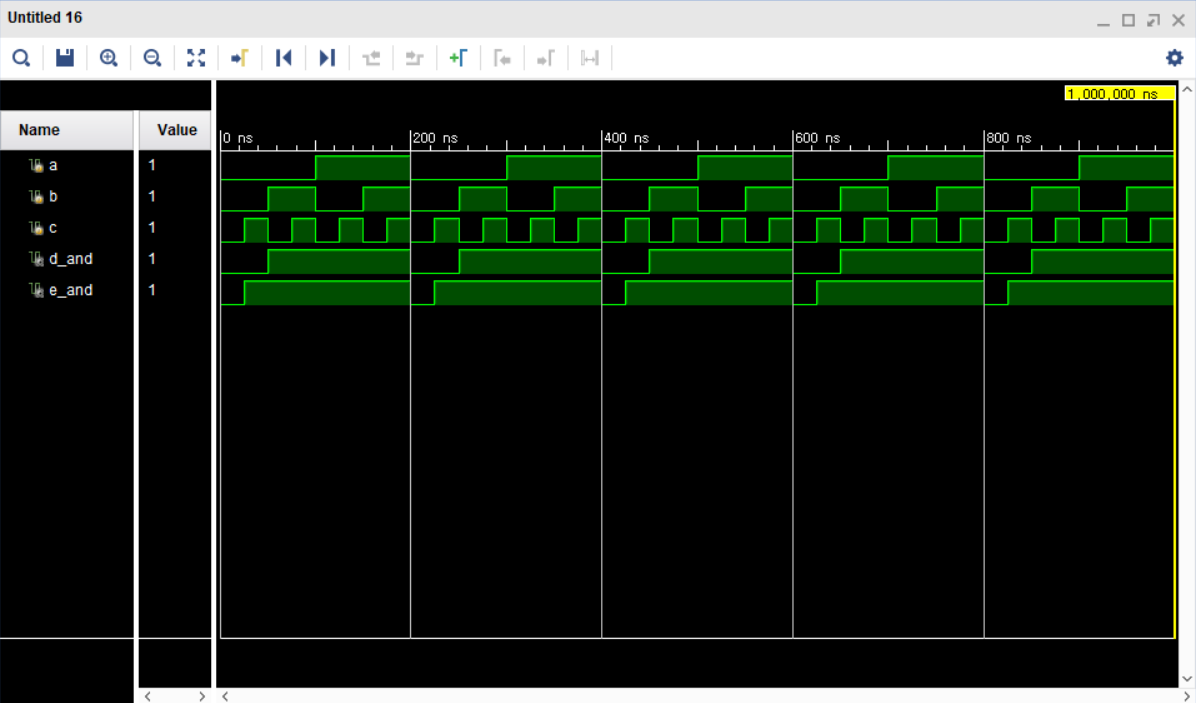
E를 먼저 and 연산으로 계산해주고, C, E를 다시 and 연산으로 F를 구한 뒤, D, F를 and 연산으로 G를 구해준다.



1. **3-input OR gate의 simulation 결과 및 과정에 대해서 설명하시오.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| In A | In B | In C | Out D | Out 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 |

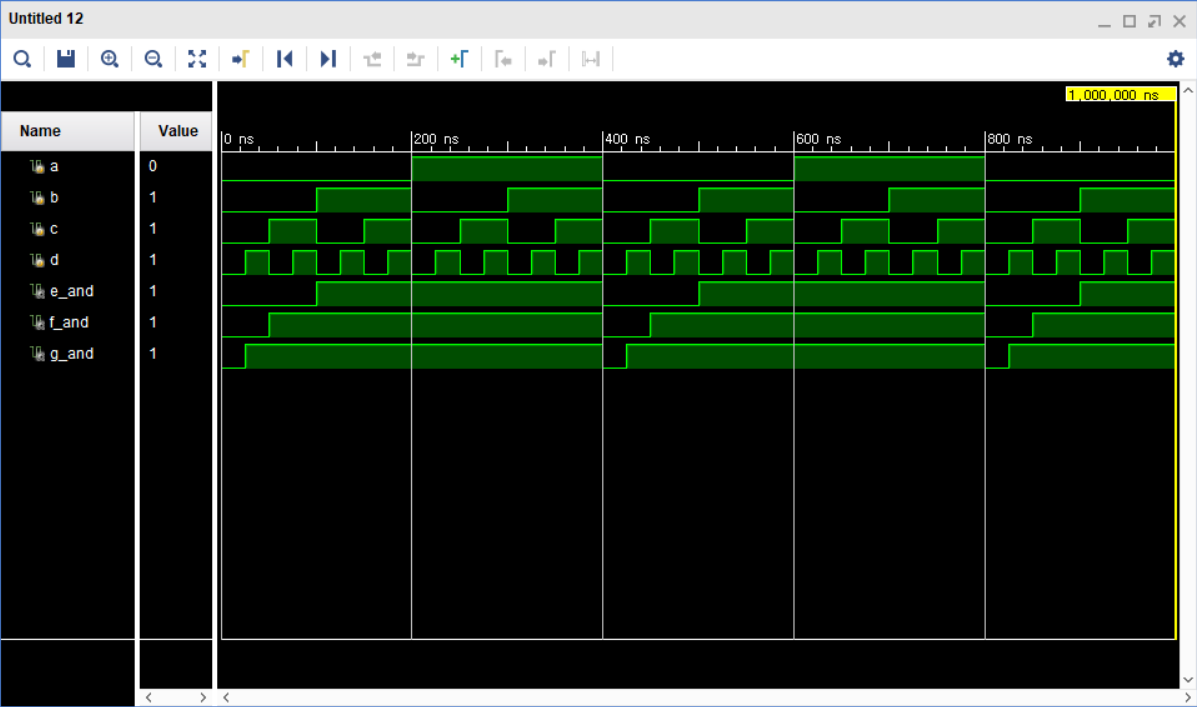
D를 먼저 or 연산으로 계산해주고, C, D를 다시 or 연산으로 E를 구한다.



1. **4-input OR gate의 simulation 결과 및 과정에 대해서 설명하시오.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | 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 |

E를 먼저 or 연산으로 계산해주고, C, E를 다시 or 연산으로 F를 구한 뒤, D, F를 or 연산으로 G를 구해준다.



1. **결과 검토 및 논의사항.**

진리표로 예상할 수 있었던 결과와 wave form이 동일한 형태로 나왔다.

Verilog 상에서는 and(A, B)와 같은 방법과 A & B 로 표현하는 방법 2가지가 가능하다.

1. **추가 이론 조사 및 작성.**

이외에도 기본 gate들의 Verilog 표현법

assign not = ~ a;

assign and = a & b;

assign or = a | b;

assign xor = a ^ b;

assign nor = ~(a | b);

assign nand = ~(a & b);