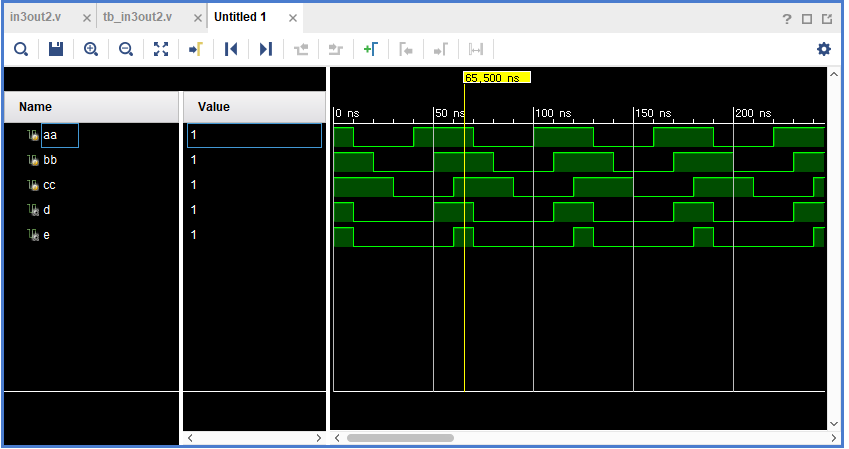
3주차 결과보고서

전공: 생명과학과 학년: 4학년 학번: 20182186 이름: 김승원

**1. FPGA 동작법**

우선 Verilog 코딩을 한다. FPGA를 동작하기 위해서 FPGA 보드와 컴퓨터를 연결한다. 그 다음 Device를 assignment해준다. Device는 sc7a75tfgg484-1을assignment해준다. 이후 add sources에서 constraints source를 추가한다. 파일명을 적고 FPGA pin list에서 할당하고 싶은 verilog source의 port와 pin을 연결한다. 이후 synthesis와 implementation을 진행하고, device configuration을 진행한다. 하드웨어 매니저를 열고 open target에서 기기를 connect한다. 이후 program device를 진행하고 동작결과를 확인한다

**2. 3-input AND gate의 시뮬레이션 결과 및 과정**



**<fig1. 3-input AND gate의 시뮬레이션 결과>**

a, b를 input으로 받은 결과를 d를 통해 출력했고, 그 d를 c와 input으로 사용해 e를 출력했다. 결과적으로 output d의 경우 a, b모두 input 신호를 모두 high로 받을 때만 high신호가 출력됐다. 또한 output e의 경우 d와 c모두 high 신호를 받을 때 high 신호를 출력한 것을 확인했다.

Input a, b -> output d

Input d, c -> output e

**<table 1. 3-input AND gate truth table>**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b | c | d | 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 |

**3. 4-input AND gate simulation 결과 및 과정**

텍스트, 전자기기, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**<fig2. 4-input AND gate의 시뮬레이션 결과>**

a, b를 입력했을 때 e를 출력했고, c와 e를 입력했을 때 F를 출력했고, f와 d를 입력했을 때 g가 출력되도록 했다. 결과적으로 a와 b모두 high신호를 받았을 때, high 신호가 출력됐고, c와 e 모두 high 신호를 입력 받았을 때, f에서 high신호가 출력됐다. 그리고 마지막으로 f와 d모두 high 신호일 때 g가 high 신호로 출력됐다.

Input a, b -> output e

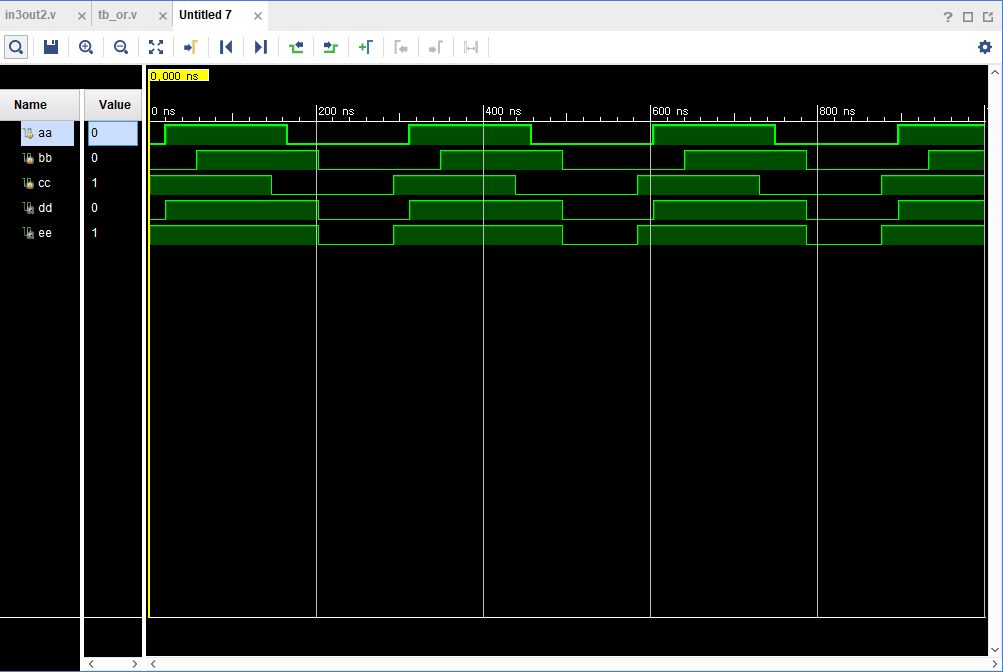
Input c, e -> output f

Input d, f -> output g

**<table 2. 4-input AND gate truth table>**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 |

**4. 3-input OR gate simulation 결과 및 과정**



**<fig3. 3-input OR gate의 시뮬레이션 결과>**

위는 a, b신호를 입력하면 d가 출력되도록 했고, c와 d를 입력했을 때 e가 출력되도록 했다. a, b 중 한 신호라도 high 신호로 입력하면 출력결과 d는 high 신호가 출력됐다. 마찬가지로 d와 c 중 한 신호라도 high신호가 입력되면 출력결과 e는 high 신호로 출력된 것을 확인했다.

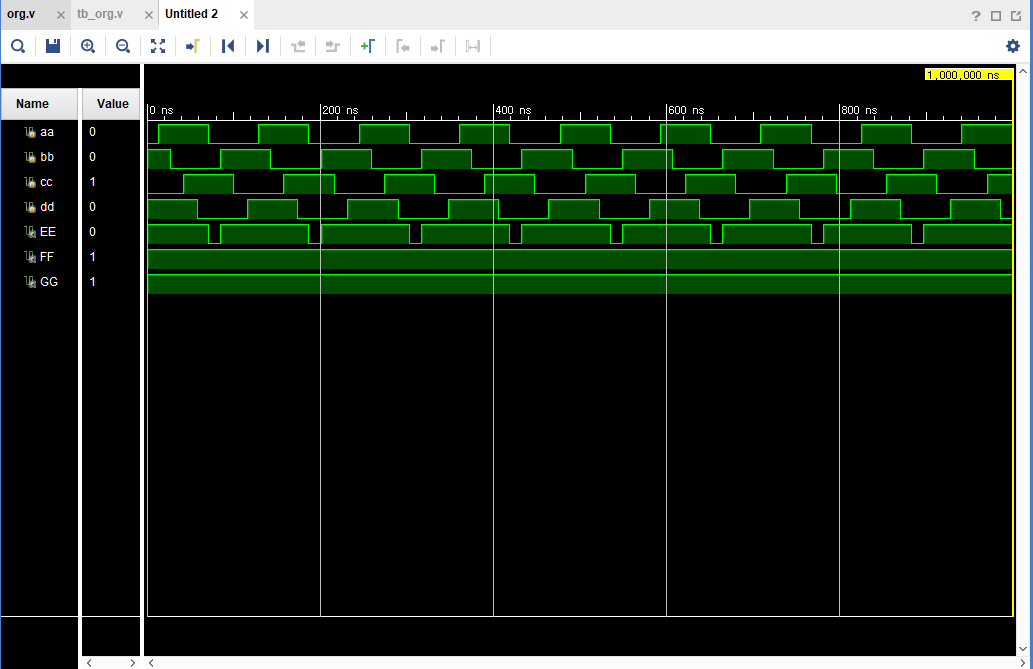
Input a, b -> output d

Input d, c -> output e

**<table 3. 3-input OR gate truth table>**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b | c | d | 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 |

**5. 4-input OR gate simulation 결과 및 과정**



**<fig4. 4-input OR gate의 시뮬레이션 결과>**

a, b를 입력했을 때 e가 출력되도록 했다. c와 e를 입력했을 f가 출력되도록 했다. 마지막으로 d와 f를 입력했을 때 g가 출력되도록 했다. 결과적으로 a, b 중 1신호라도 high 신호를 입력 받으면 e는 high 신호로 출력됐고, c와 e중 한 신호라도 high 신호를 입력 받으면 f는 high 신호로 출력됐다. 마지막으로 d와 f의 경우에도 입력한 신호 중 한 신호라도 high 신호면 출력은 high로 출력됐다.

Input a, b -> output e

Input c, e -> output f

Input d, f -> output g

**<table 4. 3-input OR gate truth>**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 |

5. AND gate의 경우 2개 이상의 신호를 입력 받아서 그 때 모든 신호가 high 신호여야 high신호를 출력하고 그 이외의 경우에는 Low를 출력하는 특징이 있다. 3-input AND gate와 4-input AND gate에서 최종 output을 확인했을 때 모두 high신호를 입력 받은 경우만 high를 출력했고, 그 이외의 경우에는 Low를 출력하는 결과를 확인했다. OR gate의 경우 2개 이상의 신호를 입력 받은 다음 두 신호 중 한 신호라도 high인 경우에 high 신호를 출력한다. 즉 둘 신호가 모두 Low 인 경우 Low 신호를 출력한다. 3-input OR gate와 4-input OR gate 모두에서 확인할 수 있었듯이 한 신호라도 high 신호가 입력됐을 때 output은 high신호가 출력됐다.

6. AND gate와는 다른 Negative-AND gate가 있다. NAND gate라고 하며, 모든 입력이 참인 경우 출력이 거짓이 되는 논리 게이트이다. 또한 모든 입력이 거짓일 때 출력이 참이 되는 게이트인 NOR gate도 있다. NOR gate는 OR gate와 반대로 동작하는 gate이다.