4주차 결과보고서

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

**1. 실험 목적**

NAND gate, NOR, XOR gate의 동작을 이해해보고 확인해보는 시간을 갖는다. Verilog를 이용해서 다중으로 입력되는 NAND와 NOR gate 그리고 XOR gate를 구현해보고 입력신호를 생성하여 simulation을 실행해본 뒤 구현한 각각의 gate의 동작을 확인해본다. 또한 FPGA를 통해 Verilog로 구현된 회로가 어떻게 동작되는지 확인한다.

**2. 4-input NAND gate**

1) simulation 결과



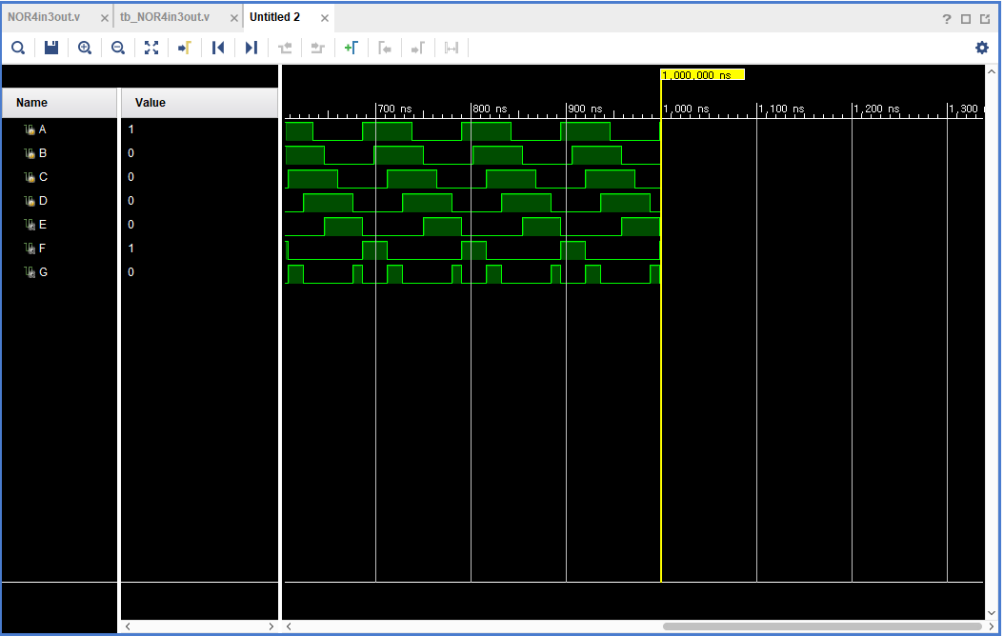
NAND gate의 결과로 A, B 모두 high로 입력됐을 때, E는 low로 출력됐다. 그 이외의 경우는 모두 high로 출력됐다. 그리고 C와 E를 입력 받을 때도 동일하게 모두 high로 입력됐을 때만 F가 low 출력됐다. 이후 D와 F를 입력 받았을 때, G의 결과도 동일했다.

<Truth table>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

**3. 4-input NOR gate**

<simulation 결과>



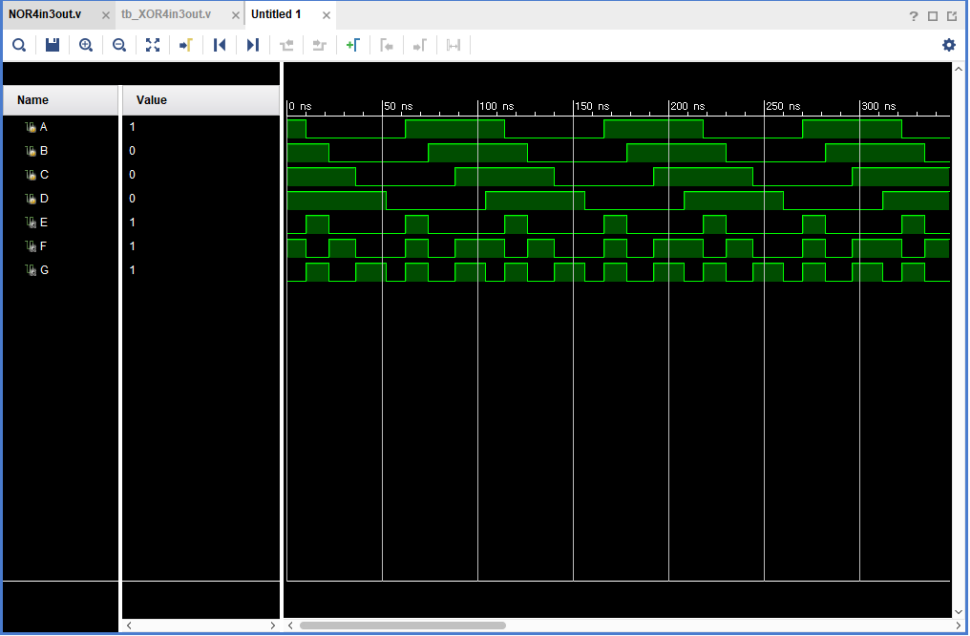
NOR의 시뮬레이션결과를 확인해보면, input 값들이 모두 low인 경우에만 high로 출력된 것을 확인할 수 있다. 위의 결과는 A와 B를 input으로 했을 때, e가 출력되도록 했다. 또한 C와 E를 입력 받아서 F를 출력했고, D와 F를 입력 받아 G를 출력했다.

<truth table>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |

**4. 4-input XOR gate**

<simulation 결과>



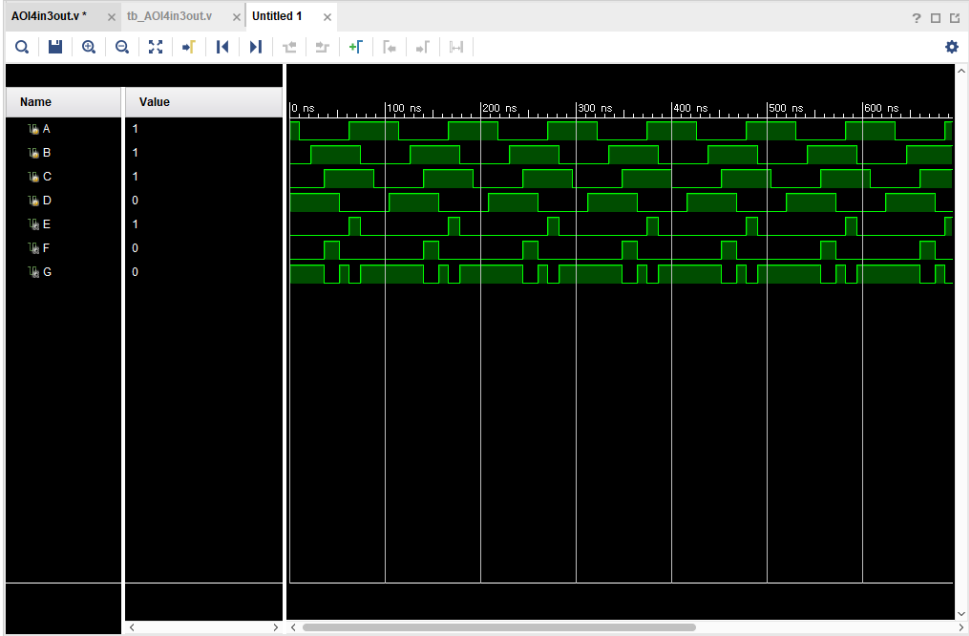
XOR은 두 input값이 서로 다른 경우(high와 low)에만 high로 출력됐다. 위의 결과는 a와 b를 input으로 했을 때, e가 출력되도록 했다. 또한 C와 E를 입력 받아서 F를 출력했고, D와 F를 입력 받아 G를 출력했다.

<truth table>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

**5. 4-input AOI gate**

<simulation 결과>



AOI gate로 A, B를 input으로 AND gate를 통과시켜 E를 출력했다. 그리고 C, D를 input으로 하여 AND gate를 통과시켜 F를 출력시켰다. 마지막으로 E와 F를 NOR gate를 통해서 G를 출력시켰다. 위의 결과는 A, B와 C, D를 input으로 했을 때, AND gate의 결과로 모두 high로 입력 받는 경우에만 high로 출력됐다. G의 경우 NOR gate를 지나며 모두 low일때만 high로 출력되는 결과를 확인했다.

<truth table>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 |

6. NAND gate는 AND gate의 결과에서 역을 취한 값이고, 모든 입력이 high일 때 low를 출력하는 결과를 확인했다. NOR gate는 OR gate를 역을 취한 gate로, 모든 입력이 low일 때, high가 출력됐다. XOR gate는 입력되는 input의 high(1)가 홀수로 입력될 때, high로 출력됐다.

7. 또 다른 logic gate로 XNOR gate가 있다. XOR gate 뒤에 not gate를 붙인 것으로 XOR gate와 출력 값이 반대가 된다. 입력 값이 동일할 때 1이 출력되는 특징이 있다. 따라서 비교 게이트라고 불리기도 한다. AND와 OR로는 NOR, NAND 등의 gate를 만들 수 없다. 하지만 NAND와 NOR을 조합하면 모든 gate를 만들 수 있다. 따라서 NAND, NOR을 많이 사용하여 이 두 gate를 universal gate라고도 한다.