4주차 결과보고서

전공 : 컴퓨터공학과 학년 : 3학년 학번 : 20211558 이름 : 윤준서

**1. 실험 목적**

• NAND/NOR/XOR/AOI Gate의 동작을 이해하고 확인한다.

• Verilog를 사용하여 다중입력 NAND/NOR/XOR/AOI Gate를 구현한다.

• 입력 신호 생성 후 Simulation을 통해 각 구현된 Gated의 동작을 확인한다.

• FPGA를 통해 Verilog로 구현된 회로의 동작을 확인한다.

**2. 4-input NAND gate**

4-input 3-output NAND gate의 Blooean 식과 Schematic은 다음과 같다.

**E = ~(A • B)**

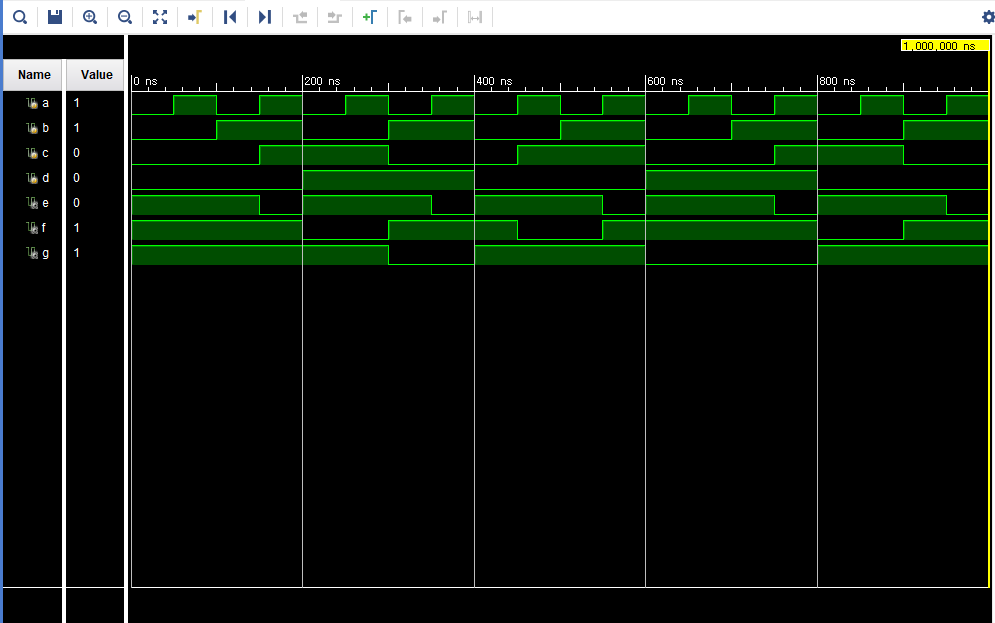
**F = ~(C • E)**

**G = ~(D • F)**

도표, 텍스트, 평면도이(가) 표시된 사진

자동 생성된 설명

Vivado를 통해 작성한 NAND gate의 Simulation 결과와 진리표는 다음과 같다.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IN A** | **IN B** | **IN C** | **IN D** | **OUT E** | **OUT F** | **OUT G** |
| **0** | **0** | **0** | **0** | **1** | **1** | **1** |
| **0** | **0** | **0** | **1** | **1** | **1** | **1** |
| **0** | **0** | **1** | **0** | **1** | **1** | **1** |
| **0** | **0** | **1** | **1** | **1** | **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** | **0** | **1** | **1** |
| **1** | **1** | **0** | **1** | **0** | **1** | **1** |
| **1** | **1** | **1** | **0** | **0** | **0** | **1** |
| **1** | **1** | **1** | **1** | **0** | **0** | **0** |

**3. 4-input NOR gate**

4-input 3-output NOR gate의 Blooean 식과 Schematic은 다음과 같다.

**E = ~(A + B)**

**F = ~(C + E)**

**G = ~(D + F)**

도표이(가) 표시된 사진

자동 생성된 설명

Vivado를 통해 작성한 NOR gate의 Simulation 결과와 진리표는 다음과 같다.

스크린샷, 멀티미디어 소프트웨어, 그래픽 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IN A** | **IN B** | **IN C** | **IN D** | **OUT E** | **OUT F** | **OUT G** |
| **0** | **0** | **0** | **0** | **1** | **1** | **1** |
| **0** | **0** | **0** | **1** | **1** | **1** | **0** |
| **0** | **0** | **1** | **0** | **1** | **0** | **0** |
| **0** | **0** | **1** | **1** | **1** | **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** | **0** | **0** | **0** |
| **1** | **1** | **0** | **1** | **0** | **0** | **0** |
| **1** | **1** | **1** | **0** | **0** | **0** | **0** |
| **1** | **1** | **1** | **1** | **0** | **0** | **0** |

**4. 4-input XOR gate**

4-input 3-output XOR gate의 Blooean 식과 Schematic은 다음과 같다.

**E = A ⊕ B**

**F = C ⊕ E**

**G = D ⊕ F**

도표, 라인이(가) 표시된 사진

자동 생성된 설명

Vivado를 통해 작성한 XOR gate의 Simulation 결과와 진리표는 다음과 같다.

스크린샷, 멀티미디어 소프트웨어, 그래픽 소프트웨어, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IN A** | **IN B** | **IN C** | **IN D** | **OUT E** | **OUT F** | **OUT 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**

4-input 3-output AOI gate의 Blooean 식과 Schematic은 다음과 같다.

**E = A • B**

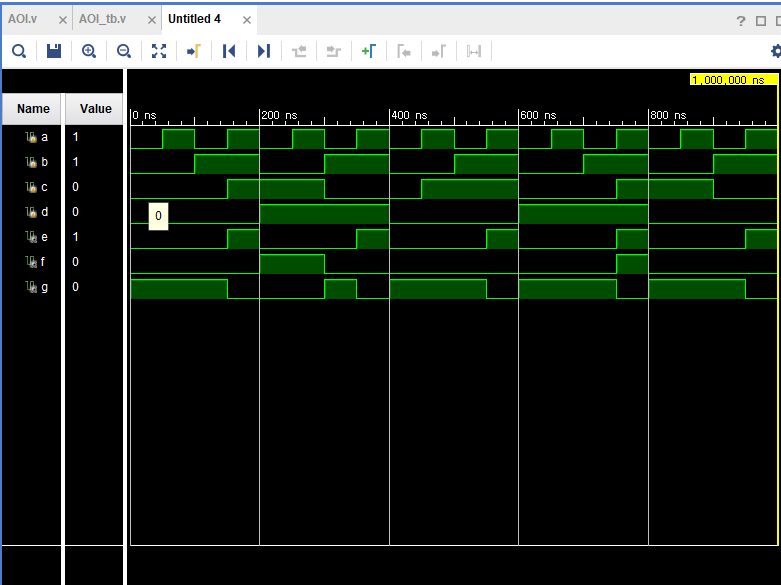
**F = C • D**

**G = ~(E + F)**

도표, 라인, 텍스트, 그래프이(가) 표시된 사진

자동 생성된 설명

Vivado를 통해 작성한 AOI gate의 Simulation 결과와 진리표는 다음과 같다.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IN A** | **IN B** | **IN C** | **IN D** | **OUT E** | **OUT F** | **OUT 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** | **0** | **1** |
| **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** | **0** | **1** |
| **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** | **0** | **1** |
| **1** | **1** | **0** | **0** | **1** | **1** | **0** |
| **1** | **1** | **0** | **1** | **1** | **1** | **0** |
| **1** | **1** | **1** | **0** | **1** | **1** | **0** |
| **1** | **1** | **1** | **1** | **1** | **1** | **0** |

**6. 결과 검토 및 논의사항**

NAND 게이트는 두 입력 값이 모두 1인 경우 0이 출력되는 논리 게이트다.

NOR 게이트는 두 입력 값이 하나라도 1인 경우 0이 출력되는 논리 게이트다.

XOR 게이트는 두 입력 값 중 1이 홀수개인 경우 1이 출력되는 논리 게이트다.

AOI 게이트는 AND-OR-Invert의 약자로, AND/OR/NOT 게이트를 조합하여 연산하는 게이트다. 하나의 자체적인 게이트이기 때문에 AND/OR/NOT 게이트를 축약하여 연산 속도를 높일 수 있다.

NAND/NOR 게이트는 이전에 다루었던 AND/OR 게이트와 같이 Boolean 식을 각각 나눠서 계산하는 경우와 연속적으로 한번에 계산하는 경우에 답이 같게 나온다. 즉 NOR 게이트의 경우,

**• E = ~(A + B), F = ~(C + E), G = ~(D + F)**

**• G = ~(~(~(A + B) + C) + D)**

위의 두 식이 같은 input일 경우 output의 값이 같다. 이는 NAND 게이트와 XOR 게이트에도 성립한다.

Verilog의 회로에선 input 값들을 차례로 연산하여 시뮬레이션을 실행하는 것을 위의 Schematic을 통해 알 수 있다.

**7. 추가이론**

Verilog에서 XOR 게이트의 논리식을 **Y = A ^ B** 로 표현하지만, AND/OR/NOT 게이트를 사용하여 표현할 수 있다. 실제 XOR의 논리식은 (A' AND B) OR (A AND B'), 즉 Y = (~A • B) + (A • ~B) 이므로 Verilog에서 **Y = (~A & B) | (A & ~B)** 로 표현할 수 있다.

하지만 Schematic의 형태가 여러 게이트를 사용하므로 훨씬 복잡해 진다.

NAND/NOR/AOI 게이트는 사용되는 트랜지스터 수가 적어 회로의 크기를 줄이는 장점이 있다. NAND/NOR 게이트의 경우 각각 AND/OR 게이트와 NOT 게이트를 합친 느낌이라 오히려 트랜지스터 수가 많을 것 같지만, 실제로 NAND/NOR 게이트에 사용되는 트랜지스터 수는 4개이며, AND/OR 게이트에 사용되는 트랜지스터 수는 6개이다.