4주차 결과보고서

전공: 영미어문/컴퓨터공학과 학년: 3학년 학번: 20200185 이름: 박정주

**1.**

NAND, NOR, XOR, AOI 게이트의 동작을 이해하고, Verilog를 통해 다중입력 NAND, NOR, XOR, AOI 게이트를 구현한 뒤 FPGA를 통해 회로의 동작을 확인하는 것을 이번 실험의 목적으로 한다.

**2.**

4 input NAND gate를 구현하는 과정은 다음과 같다. 먼저 Design source를 만들어 4개의 input, 3개의 out을 선언하고 아래와 같이 코딩한다.

텍스트이(가) 표시된 사진

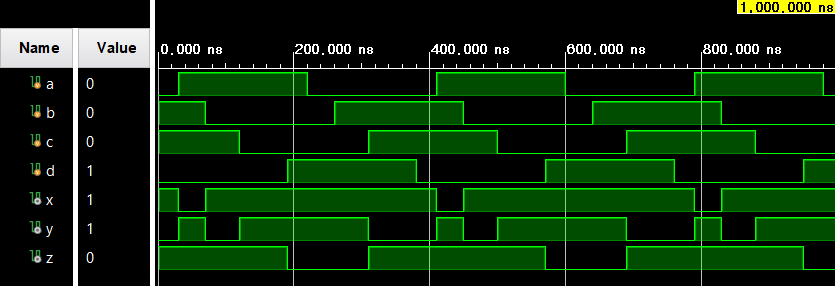
자동 생성된 설명

아래와 같이 simulation source를 코딩한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

4 input NAND gate의 simulation 결과는 다음과 같다.



4 input NAND gate의 진리표는 다음과 같다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| in a | in b | in c | in d | out x | out y | out z |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

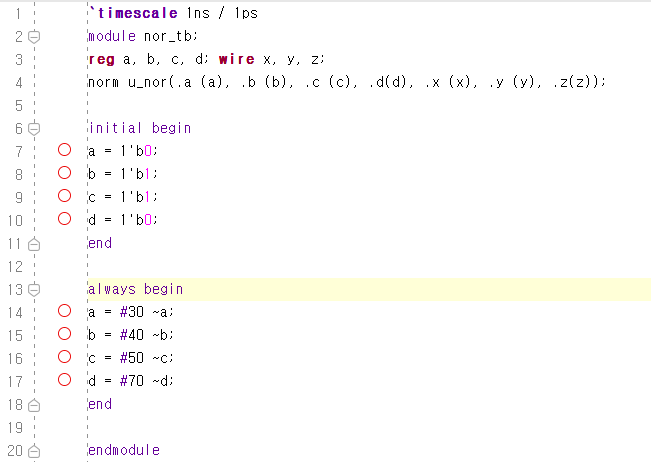
**3.**

4 input NOR gate를 구현하는 과정은 다음과 같다. 먼저 Design source를 만들어 4개의 input, 3개의 out을 선언하고 아래와 같이 코딩한다.

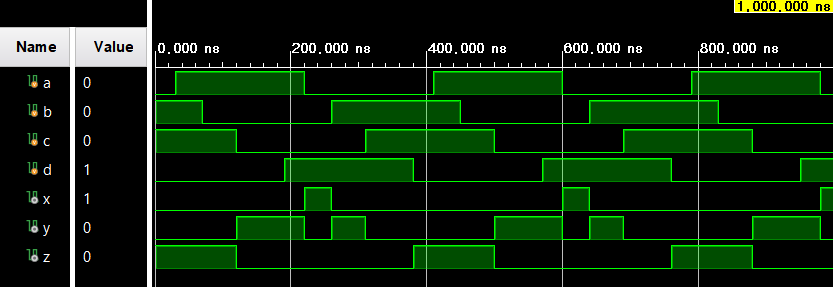
텍스트이(가) 표시된 사진

자동 생성된 설명

아래와 같이 simulation source를 코딩한다.



4 input NOR gate의 simulation 결과는 다음과 같다.



4 input NOR gate의 진리표는 다음과 같다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| in a | in b | in c | in d | out x | out y | out z |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 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 gate를 구현하는 과정은 다음과 같다. 먼저 Design source를 만들어 4개의 input, 3개의 out을 선언하고 아래와 같이 코딩한다. exclusive or 연산자인 ^를 사용한다.

텍스트이(가) 표시된 사진

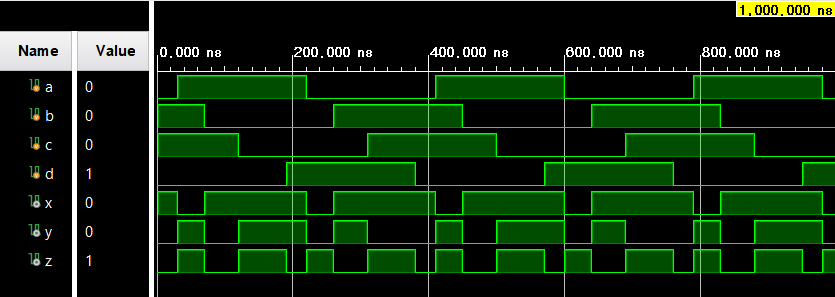
자동 생성된 설명

아래와 같이 simulation source를 코딩한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

4 input XOR gate의 simulation 결과는 다음과 같다.



4 input XOR gate의 진리표는 다음과 같다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| in a | in b | in c | in d | out x | out y | out z |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 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 gate를 구현하는 과정은 다음과 같다. 먼저 Design source를 만들어 4개의 input, 3개의 out을 선언하고 아래와 같이 코딩한다.

텍스트이(가) 표시된 사진

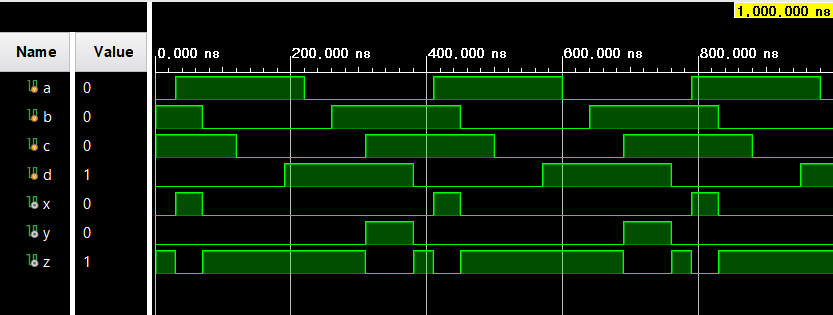
자동 생성된 설명

아래와 같이 simulation source를 코딩한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

4 input AOI gate의 simulation 결과는 다음과 같다.



4 input AOI gate의 진리표는 다음과 같다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| in a | in b | in c | in d | out x | out y | out z |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 |

**6.**

실험 결과, 4 input NAND 게이트에서 input a와 b의 값이 모두 1인 경우에만 x값으로 0이 출력되었다. input c와 output x가 모두 1인 경우에만 y값으로 0이 출력되었으며, d와 z의 값이 모두 1인 경우에만 최종적으로 z값이 0으로 출력되었다.

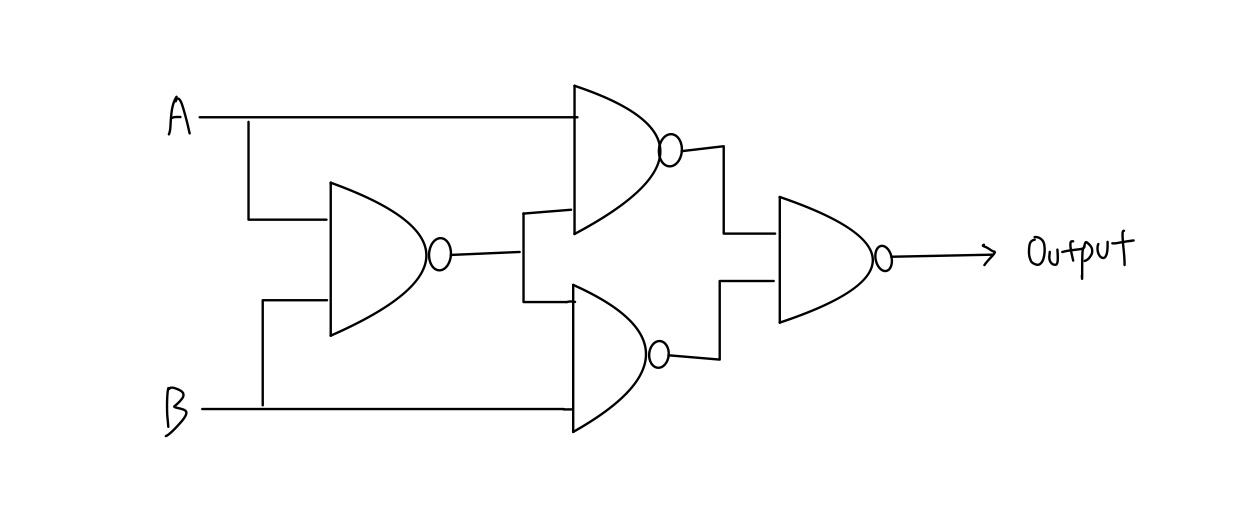
4 input NOR gate에서 input a와 b의 값이 모두 1인 경우에만 x값으로 0이 출력되었다. input c와 output x가 모두 1인 경우에만 y값으로 0이 출력되었으며, d와 z의 값이 모두 1인 경우에만 최종적으로 z값이 0으로 출력되었다.

4 input XOR gate에서는 입력 값들 중 1의 개수가 홀수일 때, 즉 a와 b의 값 둘 중 하나만 1일 때 x의 값으로 1이 출력되었다. 1의 개수가 0개이거나 2개인 경우 1이 출력되었다. x와 c의 값 중 하나만 1일 때 y값이 1로 출력되었고, y와 z의 값 중 하나만 1일 때 최종적으로 z값이 1로 출력되었다.

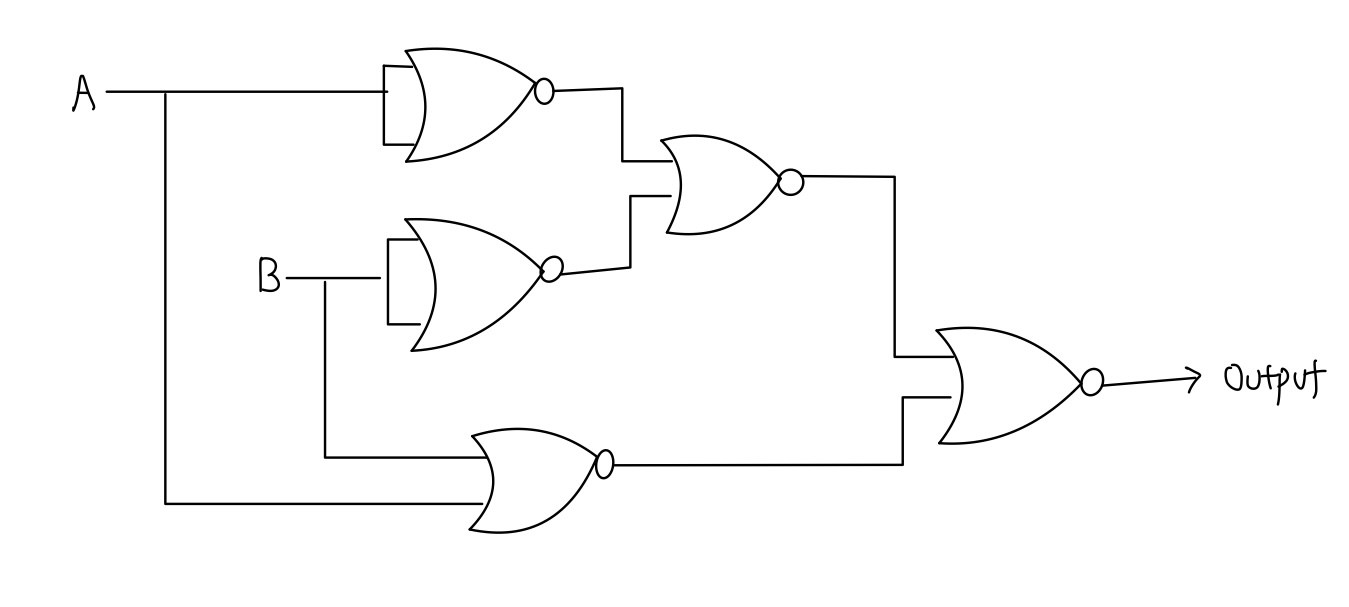
4 input AOI gate에서 input a와 input b의 값이 모두 1일 때에만 x값이 1로 출력되었다. 마찬가지로 input c와 input d의 값이 모두 1일 때 y값이 1로 출력되었다. x와 y가 NOR gate의 입력 값이 되어, x와 y가 모두 0일 때만 최종적으로 output z의 값이 1로 출력되었다.

**7.**

실험에서는 XOR gate를 Verilog의 exclusive or 연산자인 ^를 통해 구현하였다. 그러나 XOR 게이트는 NAND 게이트만을 사용하여 아래와 같이 구현할 수 있다.



아래와 같이, NOR 게이트만을 사용하여 XOR 게이트를 구현할 수도 있다.



위의 두 가지 방법으로 구현한 XOR gate의 simulation을 돌리면 ^를 사용한 XOR gate와 같은 output이 나온다.