4주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20231609 이름: 정희선

**1.**

**NAND, NOR, XOR, AOI gate** 각각의 특성을 알아보고, Verilog를 이용하여 각 게이트를 구현해 보는 것이 이 실험의 목적이다.

**2.**

4-input NAND gate

1. **Boolean식:** ,

**2) Verilog 코딩**

*`timescale 1ns / 1ps*

*module FourNand(*

*input a,b,c,d,*

*output e,f,g*

*);*

*assign e=~(a&b);*

*assign f=~(c&e);*

*assign g=~(d&f);*

*endmodule*

**3) Simulation 결과**

스크린샷, 소프트웨어, 멀티미디어 소프트웨어, 라인이(가) 표시된 사진

자동 생성된 설명

**4) FPGA 동작**

각 게이트마다 신호 전달 시간이 필요하기 때문에 단계마다 지연 시간이 발생한다. 이러한 지연이 누적되어 전체 시스템 속도에 영향을 준다.

**5) 진리표**

NAND 게이트는 모든 입력이 1인 경우에만 0을 출력한다.

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

1. **Boolean식:**

**2) Verilog 코딩**

*`timescale 1ns / 1ps*

*module FourNor(*

*input a,b,c,d,*

*output e,f,g*

*);*

*assign e=~(a||b);*

*assign f=~(c||e);*

*assign g=~(d||f);*

*endmodule*

**3) Simulation 결과**

**스크린샷, 소프트웨어, 텍스트, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

**4) FPGA 동작**

각 게이트마다 신호 전달 시간이 필요하기 때문에 단계마다 지연 시간이 발생한다. 이러한 지연이 누적되어 전체 시스템 속도에 영향을 준다.

**5) 진리표**

NOR gate는 입력이 모두 0일때만 1을 출력한다.

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

**1) Boolean식:**

**2) Verilog 코딩**

*`timescale 1ns / 1ps*

*module FourXor(*

*input a,b,c,d,*

*output e,f,g*

*);*

*assign e=a^b;*

*assign f=c^e;*

*assign g=d^f;*

*endmodule*

**3) Simulation 결과**

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

자동 생성된 설명

**4) FPGA 동작**

각 게이트마다 신호 전달 시간이 필요하기 때문에 단계마다 지연 시간이 발생한다. 이러한 지연이 누적되어 전체 시스템 속도에 영향을 준다.

**5) 진리표**

XOR gate는 홀수 개의 1을 입력으로 받은 경우 1을 출력한다.

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

**1) Boolean식:**

**2) Verilog 코딩**

*`timescale 1ns / 1ps*

*module FourAOI(*

*input a,b,c,d,*

*output e,f,g*

*);*

*assign e=a&b;*

*assign f=c&d;*

*assign g=~(e||f);*

*endmodule*

**3) Simulation 결과**

**스크린샷이(가) 표시된 사진

자동 생성된 설명**

**4) FPGA 동작**

AND, OR, NOT 게이트의 조합으로 구현하므로, 여러 논리 연산이 한꺼번에 수행되면서 지연이 발생할 수 있다.

**5) 진리표**

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

**6.**

* NAND gate는 NOT과 AND 연산이 결합된 게이트로, 모든 입력이 1일 때만 0을 출력하고, 나머지는 1을 출력한다.
* NOR gate는 NOT과 OR 연산이 결합된 게이트로, 모든 입력이 0일 때만 1을 출력하고, 나머지는 0을 출력한다.
* XOR gate는 서로 다른 입력에 대해서만 1을 출력하는 게이트다.
* AOI gate는 AND-OR-Inverter 연산을 결합한 회로다. 두 AND 연산 결과를 OR 연산 한 후, 그 결과를 NOT으로 반전하여 출력한다.
* XOR 연산은 입력의 비대칭성을 처리하는 연산이 포함되어 신호 지연이 발생할 수 있다.
* AOI 연산은 논리 게이트 여러 개가 결합된 형태이므로, 구현이 복잡해 신호 지연이 발생할 수 있다.
* 따라서, NAND와 NOR은 비교적 간단한 논리 게이트임에 반해 XOR과 AOI는 복잡하므로 비교적 느리게 실행될 수 있다.

**7.**

- 카르노 맵: 논리식을 간소화하는 방법이다. 다중 입력 논리 회로에서 중복된 항을 제거하여 최소 형태의 논리식을 얻을 수 있다. 논리 회로를 최적화하는 데 주로 사용된다.

- 표현 가능성의 완전성: 주어진 게이트 만으로 모든 디지털 논리 회로를 구현할 수 있는 성질이다. 예를 들어, NAND 게이트 만으로 AND, OR, NOT 게이트를 구현할 수 있다.