3주차 결과보고서

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

**1.**

먼저 Verilog를 통해 논리 회로를 코딩해야 한다. 그 후 Settings에서 장치를 xc7a75tfgg484로 설정해준다. 좌측의 Add Sources를 누르고 add or create constraints를 눌러 파일을 생성한 뒤, FPGA의 pin과 Verilog 소스의 port를 연결 짓는다. 좌측의 Run Synthesis와 Run Implementation을 모두 실행시킨다. Generate Bitstream을 눌러 Open Hardware Manager을 선택한다. FPGA 키트가 연결되어 있고 전원이 켜져 있는 상태라면, Open Target – Auto Connect를 선택하면 연결이 완료된다. 이후 동작을 검증하면 된다.

**2.**

3 input AND gate를 만들기 위해서 3개의 input, 2개의 output을 선언하고 아래와 같이 코딩한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

아래와 같이 3 input AND gate의 simulation source를 작성한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Simulation 결과는 아래와 같다.

텍스트, 점수판이(가) 표시된 사진

자동 생성된 설명

진리표는 아래와 같다.

|  |  |  |  |
| --- | --- | --- | --- |
| aa | bb | cc | Output (y) |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

**3.**

4 input AND gate를 만들기 위해서 4개의 input, 3개의 output을 선언하고 아래와 같이 코딩한다.

**텍스트이(가) 표시된 사진

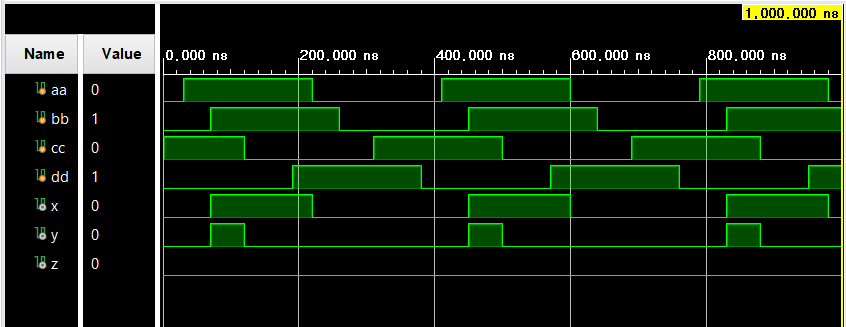
자동 생성된 설명**

아래와 같이 4 input AND gate의 simulation source를 작성한다.

**텍스트이(가) 표시된 사진

자동 생성된 설명**

Simulation 결과는 아래와 같다.

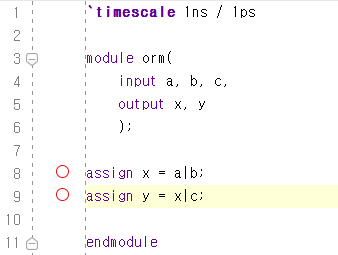
****

진리표는 아래와 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| aa | bb | cc | dd | Output (z) |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

**4.**

3 input OR gate를 만들기 위해서 3개의 input, 2개의 output을 선언하고 아래와 같이 코딩한다.



아래와 같이 4 input OR gate의 simulation source를 작성한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Simulation 결과는 아래와 같다.

텍스트, 점수판이(가) 표시된 사진

자동 생성된 설명

진리표는 아래와 같다.

|  |  |  |  |
| --- | --- | --- | --- |
| aa | bb | cc | Output (y) |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

**5.**

4 input OR gate를 만들기 위해서 4개의 input, 3개의 output을 선언하고 아래와 같이 코딩한다.

테이블이(가) 표시된 사진

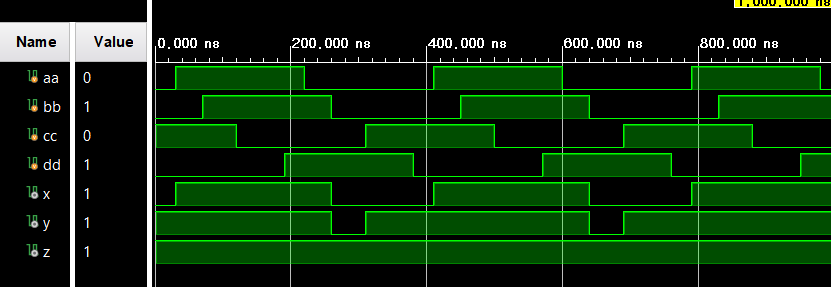
자동 생성된 설명

아래와 같이 4 input OR gate의 simulation source를 작성한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Simulation 결과는 아래와 같다.



진리표는 아래와 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| aa | bb | cc | dd | Output (z) |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**6.**

Simulation 결과, 3 input AND gate와 4 input AND gate에서는 모든 입력 신호가 1이어야만 1이 출력되고, 그 외의 경우에는 0이 출력되었다. 3 input OR gate와 4 input OR gate에서는 모든 입력 신호가 0이어야만 0이 출력되었다. 입력 신호 중 하나라도 1이라면 1이 출력되었다.

**7.**

AND gate, OR gate, NOT gate와 같은 논리 게이트의 결합을 통해 보다 복잡한 논리 회로를 구성할 수 있다. 이 외의 논리 게이트에는 NAND, NOR, XOR이 있다.