3주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20211522 이름: 김정환

**1.**

.................

프로젝트를 만들면서 사용할 Device를 정해주거나 Vivado 내에서도 Device를 변경할 수 있는데, 이때 Device는 xc7a75tfgg484-1를 선택한다. Verilog로 원하는 동작에 대한 코딩을 진행한 후, pin list에서 할당하고 싶은 Pin과 Verilog 소스의 port를 연결한다. 이는 .v 파일이 아닌 create constraints를 통해 생성한 .xdc 파일에 작성한다. 그 후에 Run Synthesis를 진행한 후에 Run Implementation을 실행하고, Generate Bitstream을 진행한다. 그 후에 보드가 연결되어 있다면 Open Hardware Manager를 통해서 열 수 있다. 그 뒤, Open target -> auto connect로 연결한다.

................

**2.**

.......................

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

3-input AND Gate에서 각각 d와 e는 a&b, d&c로 나타낼 수 있다. Simulation 결과는 아래와 같다.

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

자동 생성된 설명

진리표는 아래와 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| In A | In B | In C | Out D | Out E |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

input a, b, c의 값이 모두 1이 되어야 최종 output e가 1이 된다.

.........................

**3.**

.......................

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

4-input AND Gate에서 각각 e, f, g는 a&b, e&c, f&d로 나타낼 수 있다. Simulation 결과는 아래와 같이 나타난다.

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

자동 생성된 설명

진리표는 아래와 같다.

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

input a, b, c, d의 값이 모두 1이 되어야 최종 output g가 1이 된다.

.........................

**4.**

.......................

텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

3-input OR Gate에서 d, e는 각각 a|b, d|c로 나타낼 수 있다. Simulation 결과는 아래와 같다.

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

자동 생성된 설명

진리표는 아래와 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| In A | In B | In C | Out D | Out E |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

input a, b, c 중 하나라도 1이면 최종 output e가 1이 된다.

.........................

**5.**

.......................

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

4-input OR Gate에서 e, f, g는 각각 a|b, e|c, f|d로 나타낼 수 있다. Simulation 결과는 아래와 같다.

스크린샷, 도표, 그래픽 소프트웨어, 사각형이(가) 표시된 사진

자동 생성된 설명

진리표는 아래와 같다.

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

input a, b, c, d 중 하나라도 1이면 최종 output g는 1이 된다.

.........................

**6.**

.......................

3-input과 4-input인 경우에 대해서 각각 AND와 OR Gate를 Verilog로 코딩하여 시뮬레이션을 확인하였다. AND를 중첩으로 사용할 경우에 있어서는 모든 입력 값이 1, 즉 High여야 최종 output이 1로 출력되고, OR을 중첩으로 사용할 경우에는 입력 값 중 1이 하나라도 있으면 최종 output이 1로 출력되는 결과가 실제로 나오는 것을 확인할 수 있었다. 또한 16가지의 경우가 simulation에서 모두 출력되도록 하여 진리표의 결과를 확인할 수 있었다. 이는 이처럼 같은 gate를 사용한 case 외에도 다른 종류의 gate가 섞인 경우에 대해서도 simulation으로 결과를 확인할 수 있음을 나타낸다.

.........................

**7.**

.......................

우선 이번에 실습한 gate 외의 gate를 정리하면, not, nand, xor, nor, xnor gate가 있다. NOT gate는 결과 값을 반대로 하는 gate이다. 이는 ~로 표현한다. 즉 입력이 1인 경우 0으로, 입력이 0인 경우 1로 출력한다. NAND gate는 AND gate에 NOT을 처리한 것으로 AND의 결과에 NOT을 씌운 것과 같다. 즉, input을 (a, b)로 나타내면, (0, 0), (1, 0), (0, 1)일 때 1, (1, 1)일 때 0의 결과를 출력한다. 이는 ~&로 표현한다. NOR gate는 OR gate에 NOT을 처리한 것으로 입력이 (0, 0)일 때만 1을 출력하고 이외에는 0을 출력한다. 이는 ~|로 표현한다. XOR gate는 값이 서로 같을 때는 0, 서로 다를 때 1을 출력하는 gate이다. 즉, (0, 0), (1, 1)일 때는 0을 (1, 0), (0, 1)일 때는 1을 출력한다. 이는 ^로 표현한다. XNOR gate는 XOR gate에 NOT을 처리한 것으로 입력값이 같을 때 1을, 서로 다른 값일 때 0을 출력한다. 이는 ~^나 ^~로 표현한다.

다음으로 Verilog에서 모델링 기법에 대해 정리해보았다. 우선 구조적 모델링으로, 이는 하위 모듈의 호출 등으로 동일한 기능을 여러 곳에서 만들 때 간결히 할 수 있도록 구현한다. 다음으로 데이터플로우 모델링으로 이는 실습에서 진행한 형태의 방식으로 구현하는 것을 의미한다. 여기서는 operator로 데이터의 흐름을 표현한다. 다음으로는 동작적 모델링으로 조건을 구분하거나 진리표를 이용하는 방식으로 구현하는 방식이 있다.

.........................