2주차 결과보고서

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

**1. 연속 할당문과 절차형 할당문**

연속 할당문은 조합 논리 회로를 설계하기에 적합하다.

절차형 할당문은 순차 논리 회로와 레지스터 기반 회로를 설계하기에 적합하다.

다음은 각 할당문의 특징이다.

**연속 할당문**

- assign 키워드를 사용한다.

- wire 자료형의 연결형 신호에 값을 할당한다.

- 입력 값이 변하면 즉시 계산을 실행한다.

- 타이밍 제어를 사용하지 않는다.

**절차형 할당문**

- always 블록 내부에서 사용한다.

- reg 자료형의 저장형 변수에 값을 할당한다.

- event에 따라 순차적으로 실행한다.

- 타이밍 제어를 사용한다.

**2. Blocking과 NonBlocking의 Simulation**

Blocking과 NonBlocking는 계산, 할당 순서에서 차이가 발생한다.

**Blocking**의 경우 한 줄 한 줄 순차적으로 계산 후 값을 할당한다.

**NonBlocking**의 경우 Begin ~ end 내부의 모든 계산을 먼저 실행한 후 값을 동시에 할당한다. 이 차이에 의해 서로 다른 값이 출력된다.

다음은 각각 Blocking과 NonBlocking에 해당하는 코드와 그 시뮬레이션 이미지다.

**Blocking**

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명스크린샷이(가) 표시된 사진

자동 생성된 설명

**NonBlocking**

텍스트, 스크린샷, 폰트, 대수학이(가) 표시된 사진

자동 생성된 설명 스크린샷이(가) 표시된 사진

자동 생성된 설명

코드를 보면 '='과 '<='의 차이에 의해 Blocking과 NonBlocking으로 나뉘는 것을 알 수 있다.

Blocking의 경우 a의 값이 b로 할당한 후 b의 값을 a로 할당하기에 a와 b의 값이 같다.

반면 NonBlocking의 경우 두 할당이 동시에 이뤄지기 때문에 서로 값이 바뀐 것을 확인할 수 있다.

**3. Verilog의 조건문**

Verilog의 조건문은 C언어와 마찬가지로 for문, if문, while문, case문이 모두 존재한다. 하지만 그 목적은 C언어와 차이를 보인다. 다음은 각 조건문에 대한 C언어의 간단한 특징과 Verilog와의 차이점이다.

**for문**

**C** - 특정 조건이 만족될 때까지 코드를 반복 실행한다.

**Verilog** - C와 달리 Verilog의 for문은 순차적으로 실행되는 것이 아니라 여러 하드웨어의 구성 요소를 병렬적으로 생성한다.

**if문**

**C** - 특정 조건이 True일 경우 코드를 실행한다.

**Verilog** - C와 달리 Verilog의 if문은 하드웨어의 동작을 기술하는데, 조건에 따라 다른 경 로로 기술한다.

**while문**

**C** - 반복 제어 구조로, 특정 조건이 True인 동안 코드를 반복 실행한다.

**Verilog** - Verilog의 while문은 시뮬레이션과 test bench에서만 사용된다. for문과 if문과 달 리 하드웨어 생성과 관련이 없다.

**case문**

**C** - 논리 흐름 제어 구문으로, 특정 변수의 값에 따라 한 코드를 선택하여 실행한다.

**Verilog** - C와 달리 Verilog의 case문은 하드웨어를 제어할 때 사용된다. 특정 변수의 값에 따라 나뉘는 흐름은 비슷하지만 기본값인 default를 꼭 지정해야 한다.

**4. net 형 자료형**

net 형자료형은 디바이스의 물리적인 연결을 의미하는 자료형이다. 변수들 간이 연결과 모듈 간의 신호 흐름을 표현하며, 모듈의 입력과 출력 간에 신호가 전달될 때 사용한다. net 형 자료형은 reg와 달리 값을 저장할 수도, 직접 변경할 수 없기에 다른 소스에서 값을 불러온다.

net 형 자료형의 종류는 다음과 같다.

**wire** - 변수들이 모듈 내에서 어떻게 연결돼 있는지 나타내는 자료형이다. 출력으로 연결 되거나, 논리적 연산을 통해 신호를 전달한다.

**tri** - 3상태 net에 사용되는 자료형으로, 선을 서로 연결할 때 사용한다.

**wand** - 여러 변수들의 값을 AND로 결합하는 자료형이다.

**wor** - 여러 변수들의 값을 OR로 결합하는 자료형이다.

**supply0**, **supply1** - 상수 값을 갖는 자료형으로, supply0은 0(False), supply1은 1(True)를 나타낸다.