컴퓨터공학실험2 2주차 결과 보고서

전공: 컴퓨터공학 학년: 2학년 학번: 20201635 이름: 전찬

**1. 목적**

연속 할당문, 절차형 할당문의 차이를 비교한다. Blocking 및 nonblocking 문법의 차이를 simulation을 통해 설명한다. Verilog의 for, if, while, case문을 C언어와 비교한다. Verilog의 net 자료형을 알아본다.

**2. Verilog의 net 자료형**

할당문에 대해 알아보기 전에, Verilog에 존재하는 net 자료형에 대해 알아보자. net 자료형이란 실제로 전기 회로에서 구현될 물리적 연결을 표현하는 변수이다. 대표적인 Net 자료형으로는 wire이 존재한다. wire은 부품간의 단순한 연결을 표현해주는 변수인데, wire A, wire B 가 존재할 때 A의 값을 B에 assign B=A와 같은 표현으로 대입(구동, drive)할 수 있다.

**3. 연속 할당문과 절차형 할당문의 차이**

객체에 값을 대입하는 할당문의 종류에는 연속 할당문과 절차형 할당문이 존재한다. 연속 할당문은 우변 수식이 변화할 때 좌변 수식이 그에 따라서 계속적으로 변화하는 것을 의미한다. 실제로 net 자료형에서 사용되는데, 위에서 설명한 assign B = A 라는 표현에서 우변에 존재하는 A가 변화함에 따라서 wire B 또한 계속해서 A와 같은 값으로 변화하는 것이 연속 할당문이다.

위와 다르게 절차형 할당문은 우변에 값의 변화와 관계없이 단 한번만 실행되는데, C에서 int a = b; 라는 표현을 사용하는 것을 예시로 들 수 있다. a에는 b가 가지고 있는 값이 대입되며 이후에 b의 변화와는 무관하게 변수 a의 값은 유지되는 형태를 절차형 할당문이라고 할 수 있다. Verilog에서는 always, initial문을 통해 절차형 할당문의 실행을 조절할 수 있다.

**4. Blocking 및 nonBlocking 문법의 차이**

절차형 할당문은 Blocking / nonBlocking 할당문으로 세분화할 수 있다. 먼저 Blocking 할당문은 ‘=’ 표현을 사용한다. C언어에서 코드를 실행하는 경우 위에서부터 아래로 한 줄씩 실행되는데, Blocking 할당문은 이와 같이 위의 할당이 종료된 이후 아래의 할당이 실행되는 것을 의미한다.

이와 반대로 nonBlocking 할당문은 ‘<=’ 표현을 사용하며, 위와 아래 할당문의 우변이 먼저 평가된 이후에 객체에 대입이 실행되는 형태를 의미한다. 실제로 Simulation은 아래와 같다.

**5. Verilog의 for, if, while, case문의 C언어와의 비교**

Verilog를 처음 설명할 때, 많은 사람들이 C와 비슷하게 회로를 프로그래밍 한다고 설명한다. 이처럼 Verilog는 실제로도 다양한 문법에서 C와 Verilog는 유사성을 보인다. 각각 각각 for, if, while, case문은 C언어와 Verilog에서 비슷한 형태로 적용된다. 약간의 차이점이 있다면 Verilog에서는 begin, end 표현을 통해 C에서 중괄호와 비슷한 역할을 수행해준다는 점이 있다. 또한 case문에서 break를 기술하지 않는다는 점에서 차이가 존재하기도 한다.