2주차 예비보고서

전공: 생명과학과 학년: 4학년 학번: 20182186 이름: 김승원

1. Hardware description language는 전자 회로(하드웨어)의 원하는 기능이나 동작을 할 수 있는 회로를 기술하는데 사용하는 언어다. 점점 하드웨어는 규모가 커지고 있고, 설계의 크기가 만약 5만 gate이상이 되면 설계가 어려워진다. 따라서 결선 도식 하드웨어 표현을 언어로 표현하고자 하여 만들어진 언어다. HDL을 이용해서 디지털 시스템을 모델링하고, 시뮬레이션 하며 설계하여 검증을 하고 게이트 레벨에서뿐 만 아니라 동작이나 구조적 레벨에서도 표현이 가능하다.

또한 Verilog 이외에도 VHDL과 같은 언어도 있다. 앞의 V는 VHSIC(Very-high-Speed integrated circuits)으로 초고속 직접회로를 뜻한다. Verilog에 비해서 더 high-level로 모델링이 가능하고, pascal과 유사한 문법을 가지고 있다는 특징이 있다. HDL과 같이 VHDL도 디지털 회로를 표현하는 하드웨어 기술 중 하나인 것이다.

VHDL은 본래 미 국방부에서 ASIC의 문서화에 이용하기 위해 만들었다. 복잡한 매뉴얼로 회로의 동작을 설명하는 것 보다 회로의 동작 내용을 문서화를 시키는 것이 더 편했기 때문이다. 하지만 이러한 문서화된 문서를 디자인하는 과정에서 시뮬레이션에 사용하게 됐고, VHDL 파일을 읽어 논리 합성을 해주고 다음 실제 회로의 형태를 출력해주는 기능을 붙였다.

2. Verilog는 1983년에 Prabhu Goel이 설립한 Gateway design automation company에서 공개했다. 이후 1985년에 새로운 version인 Verilog-XL을 공개했다. 이후 1989~95년에 cadence 회사가 인수했고, Verilog HDL을 공개했다. 이때 사용자가 많이 늘어났고, IEEE표준이 만들어지면서 Verilog-95로 불리게 됐다. 2001년에 업그레이드가 있었고, 2의보수와 변수 지원 기능이 추가됐다. 또한 파일 입출력 작업이 새롭게 업그레이드됐다. 2005년에 업그레이드가 또 한 번 있었으며, 디지털 시스템의 설계에서 검증 작업의 중요성이 점점 커짐에 따라 업그레이드를 진행한 것이다.

3. Verilog는 머리, 선언, 몸체부로 크게 세 부분으로 구성된다. 머리부에서는 우선 module을 시작으로 module name과 포트 목록과 마지막으로 세미콜론을 달아준다. 그리고 맨 마지막에 endmodule로 module을 끝내준다. 선언부의 경우 module에서 필요한 것들을 선언해주는데, CD언어에서 변수를 선언해주는 것과 유사하다고 볼 수 있다. Port(port의 방향과 비트 폭), reg, wire과 parameter가 있다. 몸체부는 회로의 기능과 동작 그리고 구조를 표현하는 구문으로 구성되어 있다.

Verilog의 자료형은 Register과 net으로 구분된다. Register은 추상적인 저장장치로 reg, time, integer, real이 있다. Reg는 always나 initial과 같은 절차형 할당문에 의해서 값을 받아오는 객체이고, time은 시간형 변수, integer은 정수형 변수, real은 실수형 변수다. Net은 디바이스의 물리적인 연결로 wire은 변수들이 모듈에서 어떻게 연결됐는지 나타내는 변수이고, tri는 선을 연결할 때 사용하고 wire와는 달리 tri-state net에 사용한다.

상수를 선언할 때는 bit 수 제한이 있는 reg값을 선언할 때 (bit수)’(입력 형식)(입력 값)의 형식으로 선언한다.

Bit(size)를 정한 경우

- 4’b1111; -> binary 1111 (4bit)

Bit(size)를 정하지 않은 경우

- 214; -> integer 214

부호가 있는 수를 처리할 경우

- 8’d6; -> decimal -6 (8bit)

Verilog의 연산자는 +, -, \*, %, /과 같은 산술 연산자가 있고, 또한 >=, >, !=, ==과 같은 관계연산자도 있다. 그리고 논리적 AND인 &&, 논리적 OR인 ||, 논리적 NOT인 !가 있고, 비트 AND, OR, NOT, XOR, XNOR인 &, |, ~, ^, ^~ 또는 ~^가 있다. Shift 연산자로 >>, <<가 있고, 이 밖에도 {}인 결합 연산자와 {{}}인 반복연산자도 있다. 그리고 조건연산자로 exp1 ? exp2 :exp3도 있다.

Timescale은 ‘timescale <시간단위>/<정밀도>로 선언한다. 시간단위를 선언하면 그 파일 내 모든 시간단위는 선언한 값으로 바뀌며, 정밀도는 주어진 시간 단위로 구성 가능한 가장 작은 지연을 나타내 준다.

Assign은 피연산자의 값이 변화할 때 마다 우변의 식이 평가되면서, 그 결과 값이 할당문의 주변 net을 drive해준다. Always는 시뮬레이션이 실행될 때 반복적으로 실행되는 것인데, always @(sensitivity\_list) begin에서 sensitivity\_list는 always 문의 실행을 제어하는 역할을 하고, sensitivity\_list에 나열된 신호들 중에서 변화가 생겼을 때 always 내부의 begin-end block이 실행된다. 마지막으로 initial은 시뮬레이션이 진행될 때 딱 한 번만 실행된다.