**2020년도 2학기 컴퓨터공학실험Ⅱ**

**2주차 예비보고서**

**20191621 이민영**

1. **HDL이 무엇인지 조사하고 Verilog 이외의 HDL에 대하여 조사하시오.**

HDL이란 디지털 회로의 설계 자동화에 사용하는 하드웨어 기술 언어로 전자회로를 정밀하게 기술하는 데 사용하는 컴퓨터 언어이다. HDL은 프로세서와 같은 특정한 집적회로를 설계하기 위해서 사용되거나, FPGA와 같은 PLD를 프로그램하기 위해서 사용된다.

HDL은 소프트웨어 프로그래밍 언어와 비슷하지만 약간의 차이가 있다. 두 언어 모두 컴파일러에 의해서 처리가 되는 공통점이 있지만, HDL은 고유의 포맷으로 칩의 logic을 기술한 파일을 생성하고, 산업표준인 EDIF 포맷으로 변환한다. 마지막으로 JEDEC 포맷 파일로 변환한다. 소프트웨어 컴파일어의 경우에는 마이크로프로세서에서 직접 실행되는 명령어를 생성한다.

<Verilog> 이외의 HDL

ABEL : ABEL은 프로그래머블 논리 소자 (PLD)를 프로그래밍하는데 사용되는 프로그래밍 언어와 관련된 설계도구이다. 논리를 기술하기 위해서 사용되며 하드웨어 프로그래머에 다운로드 할 수 있는 테스트 벡터를 생성하는데도 사용된다.

AHDL : AHDL은 CPLD와 FPGA를 프로그램 하기 위한 알테라 사의 자산 하드웨어 기술 언어이다.

VHDL : VHDL은 디지털 회로 및 혼합 신호를 표현하는 하드웨어 기술 언어이다. FPGA나 집적회로 등의 전자공학 회로를 처리하는 설계 자동화에 사용한다. 기존의 심볼에 의한 회로도 작성 대신 언어적 형태로 전자회로의 기능을 표현한다. 아날로그 신호로 주로 표현하지만 실제 활용면에서는 디지털회로에 많이 적용되어 사용한다.

1. **Verilog의 역사와 발전 과정을 조사하시오.**

Verilog는 1981년 프라부 고엘이 미국의 Gateway Design Automation회사를 설립한 후 1983년에이 언어와 시뮬레이터가 공개되면서 알려졌다. 1985년에 새 버전인 Verilog-XL과 그 시뮬레이터를 공개하였고 1987년에는 VHDL 언어가 IEEE 1076 표준 HDL로 공개되었다. 이후 1989년에 Gateway회사가 Cadence Systems에 인수되어서 업계 표준 HDL로 자리잡았다. 1995년에 IEEE표준이 만들어졌고, 2001년 변경이 있었다. 그러나 디지털 시스템의 설계에서 검증의 필요성이 커지면서 이를 지원하기 위해서 확장이 필요해졌고, 이로 인해서 2005년 System Verilog가 IEEE표준으로 만들어졌다. 현재 VHDL과 Verilog는 Accellera라는 단체가 유지, 관리하고 있다.

1. **Verilog의 기본적인 구조와 문법에 관하여 조사하시오.**

Verilog HDL는 하드웨어를 설계하는 기본 방식인 top-down 방식을 따르고, C언어와 유사한 문법의 일반 목적 하드웨어 표현 언어이다

Verilog는 if문, while문 등과 같은 제어구조, 출력 루틴 및 연산자들이 C언어의 구조와 비슷하다. C언어와 달리 블록의 시작과 끝은 중괄호 기호를 사용하지 않고 begin과 end를 사용하여 구분한다. 또한 대소문자를 구분한다. (Case0sensitive)

첫번째 머리부는 module로 시작해서 모듈이름, 포트리스트, 세미콜론으로 끝난다.

선언부에서는 모듈에서 필요한 것들을 선언하고, 몸체부는 회로의 기능, 동작, 구조 등을 표현하는 구문들로 구성된다.

Register는 추상적인 저장장치로 reg, integer, time, real로 구성되어 있다. Reg는 절차형 할당문에서 값을 받는 객체이고 integer는 정수형 변수, time은 시간형 변수, real는 실수형 변수이다.

verilog에서 사용하는 data type에는 네트(nets), 레지스터(register), 파라미터(parameter)가 있다. 네트는 디바이스의 물리적인 연결, 레지스터는 추상적인 저장 장치, 파라미터는 상수를 선언한다.

네트에는 wire,tri,triand 등이 있는데 대부분 wire를 사용한다.

위와 같은 형태에서는 물리적으로 a와 b랑 연결되어 있다는 의미이다.

Wire a;

Wire b;

Assign a=b

레지스터에서는 주로 reg 데이터 타입을 사용한다. Reg는 always문 안에 사용되는 변수들을 선언할 때 사용한다.

파라미터의 경우는 상수로 변하지 않는 숫자를 의미한다.

Verilog에서 공통적으로 반복해서 사용하는 코드는 function이나 task를 이용할 수 있다. Function은 하나의 값만 받고 시간지연을 포함할 수 없는 반면에 task는 여러 개의 값을 받고 타이밍 제어문을 통해서 시간지연을 포함할 수 있다.

Verilog의 연산자들을 살펴보면, ^는 비트단위 exclusive or 기능을, &는 축약 and 기능 등을 수행하며, C언어와 비슷한 부분이 많다.

Verilog 구문 중 always문은 시뮬레이션이 실행되는 동안 반복적으로 실행되며, 타이밍 제어와 연관된 표현에 유용하게 사용된다. initial구문은 always구문과 달리 시뮬레이션이 실행되는 동아 한번만 실행된다. Verilog의 절차형 할당문의 경우, 문장이 나열된 순서대로 실행되어서 할당문 좌변의 변수 값을 갱신하는 특징을 가지며 always구문, initial 구문 안에서 사용이 된다.