2주차 예비보고서

전공: 심리학과 학년: 3학년 학번: 20190345 이름: 김동현

**1.**

HDL은 Hardware Description Language의 약자로, 전자회로의 동작과 구조를 기술하는 언어이다. C언어, 자바와 같은 언어가 인간이 기계어를 직접 기술하기 어렵다는 점을 고려하여 인간의 언어와 문법을 사용하여 만든 언어라는 점과 마찬가지로, HDL 또한 하드웨어와 관련된 정보들을 인간의 언어와 문법을 사용하여 표현할 수 있도록 하는 언어이다. HDL을 사용하여, 회로의 동작과 구조를 설계할 수 있으며, 시뮬레이션 기능을 통해 회로를 검증할 수 있다. HDL은 특정 집적회로를 설계하고 생성하기 이전, 해당 회로의 동작을 예측하고, 검증하여 오류를 수정하기 위한 모델을 생성할 때 사용된다. 뿐만 아니라 FPGA와 같은 설계 가능 논리 소자를 프로그래밍하기 위해 사용된다.

Verilog 이외의 HDL은 다음과 같다. VHDL(VHSIC Hardware Description Language)는 베릴로그와 함께 많이 상용화된 대표적인 HDL이다. 디지털 회로를 설계하는 데에 주로 사용되며, 전자회로의 기능을 구현할 수 있다. 뿐만 아니라 디지털과 아날로그 신호를 포함하는 혼합 신호를 표현할 수도 있다. AHDL(Altera Hardware Description Language)는 complex programmable logic device와 FPGA를 프로그래밍 하기 위한 언어로, c언어 문법과 AHDL의 문법이 비슷하다는 특징을 가지고 있다. 이외에도 자바를 기반으로 한 “JHDL”, 파이썬을 기반으로 한 “MyHDL”, 루비를 기반으로 한 “RHDL” 등이 있다.

**2.**

베릴로그는 1983년에 “Gateway Design Automation”에서 만들어진 HDL이다. 초기에 베릴로그는 시뮬레이션을 설명하기 위한 언어로 만들어졌다. 초기 버전은 “Verilog-95”로 IEEE 표준 1364-1995로 지정되었다. 다음 버전으로는 “Verilog-2001”이 출시되었다. IEEE 표준 1364-2001로 지정되었다. 특징으로는 “+,-,\*,/,>>”의 연산자를 사용하여 비트 연산이 가능하게 되었으며, if문과 case문이 추가되어 구문 제어에 편리함을 더했다. Always구문 역시 이 버전에서 추가 되었다. 이후 “Verilog-2005”이 IEEE 표준 1364-2005로 출시되었다. “System Verilog”가 2009년 출시되어 Verilog 언어의 표준이 되었다.

**3.**

텍스트이(가) 표시된 사진

자동 생성된 설명

베릴로그의 구조는 머리부, 선언부, 몸체부 총 3가지 부분으로 나뉘어져 있다. 머리부는 “module”로 시작하는 구문으로, 모듈의 이름과 포트 목록을 작성하며, 세미콜론(;)으로 구문의 끝을 나타낸다. 선언부는 모듈에서 필요한 내용을 선언하는 부분으로, 머리부에서 명시한 포트 목록에 있는 포트의 방향, 비트 폭, register와 net 자료형을 구성하는 reg, wire, parameter 등의 자료형을 선언한다. 몸체부는 회로의 기능과 동작, 구조를 표현하는 부분으로, always, initial, assign문, function, task의 정의와 호출 등 베릴로그 구문을 사용하여 기술한다.

베릴로그에서 사용하는 자료형은 register, net형 두 가지로 나뉜다. Register 자료형은 추상적인 저장 장치로 reg(절차형 할당문을 통해 값을 할당받는 변수), integer(정수형), time(시간형 변수), real(실수형)으로 구성된다. Net 자료형은 디바이스의 물리적인 연결을 표현하는 변수로, wire(모듈 내 변수들의 연결을 나타내는 변수), tri(3형을 나타내는 변수)로 구성된다.

베릴로그에서의 상수는 “(비트 수)’(입력형식)(입력값)”과 같이 선언한다. 비트 수는 상수가 차지하는 비트를 나타내며, 비트 수를 기재하지 않으면 입력값에 맞는), 16 비트수가 자동으로 설정된다. 입력형식은 진법을 나타내며, 2진법은 “b”, 8진법은 “o”, 10진법은 “d”(default 진법은 “h”로 표현한다.

베릴로그에서의 연산자는 c언어 연산자와 유사하다. “+,-,\*,/”는 산술연산자로, “<,>,<=,>=,==,!=”는 관계연산자로, “&,|,^,<<,>>”는 비트연산자, “?”는 조건연산자, “&&,||,!”는 논리연산자로 사용한다. C언어와 다른 점은 결합 및 반복 연산자로 중괄호({,})를 사용한다는 점이다. 결합연산자는 변수값을 이어주는 역할을 하며, 반복 연산자는 변수들을 원하는 배수로 복사한다. 베릴로그의 구문 중 timescale은 “'timescale시간단위/정밀도”로 표현하여, 프로그램을 동작을 검증할 때, 변수들의 변화 등을 시간단위로 변화시키거나 연산할 수 있도록 한다. 시간 단위는 선언 시, 파일 내에서 기준이 될 시간 단위를 결정한다. 정밀도는 시간 단위로 설정된 시간보다 작은 시간에 대한 오차 허용 범위를 나타낸다. 베릴로그의 할당문은 연속 할당문과 절차형 할당문이 있다. 연속 할당문은 net 자료형에 논리 값을 지정하기 위해 사용하는 구문으로, “assign”을 통해 할당하며, “deassign”을 통해 제거한다. 절차형 할당문은 변수 값을 갱신하는 구문으로, 모든 문장이 동시에 저장되는 “nonblocking”구문과 한 문장씩 저장되는 “blocking”구문으로 나뉜다. Always문은 시뮬레이션 내에서 반복적으로 실행되며, 시간의 제어를 받는 구문을 표현할 때에 사용된다. “always @(sensitivity\_list)” 문으로 사용되며, sensitivity\_list내에 있는 변수에 변화가 발생했을 때 always문 내의 구문들이 실행된다. Always문은 “begin”, “end”로 시작과 끝을 표현한다. Initial문은 시뮬레이션 내에서 한 번 실행되며, 절차형 구문으로 주로 구성되어 있다. 변수의 값을 갱신하는 절차형 구문의 특성으로 인해 변수 값을 갱신하기 위해 사용된다. Always문과 동일하게 “begin”, “end”로 시작과 끝을 표현한다.

**4. 참고문헌**

컴퓨터공학실험2 교재

G. Gannot and M. Ligthart, "Verilog HDL based FPGA design," International Verilog HDL Conference, 1994, pp.