2주차 예비보고서

전공: 아트앤테크놀로지 학년: 3학년 학번: 20191098 이름: 백승주

**1.**

* 1. **HDL은 무엇인가**

HDL은 Hardware Description Language의 약자로 디지털 하드웨어를 설계하는 언어다. 디지털 시스템의 논리적 구조를 표현할 수 있는 HDL은 하드웨어의 특징을 추상화해 직관적으로 접근할 수 있게 만들어주어 복잡한 하드웨어를 만들 때 유용하다.

HDL을 사용하면 디지털 회로의 구조 및 동작을 상위 레벨에서 설계가 가능해지고 시뮬레이션을 통해 그 결과가 바로 확인 가능해 손쉽게 오류를 고칠 수 있게 해준다. HDL을 사용하면 디지털시스템의 모델링, 설계, 등 다양한 분야에서 사용이 가능해 현대의 대규모 회로들을 제작하는 데에 쓰이고 있다.

* 1. HDL의 종류

1. VHDL(Very High Speed Integrated Circuit HDL) :

미 국방부에서 디지털 시스템을 문서화해 설명하려는 군사적인 목적으로 개발되었다. 정보 교환이 쉬워져 설계 비용을 절감할 수 있게 만들어주는 VDHL은 이후 IEEE스탠다드로 지정되어 현대에는 FPGA나 디지털 회로를 처리하는 설계 자동화에 사용되고 있음.

1. AHDL은 Altera Hardware Description Language:

미국 알테라 사의 자신인 하드웨어 기술 언어로 알테라 사가 자사의 제품을 설계하기 위해 개발되었다. FPGA나 복합 프로그래머블 논리 소자를 설계할 때 쓰인다.

**2**.

**2-1. Verilog의 역사와 발전과정**

-1984년 : Prabhu Goel이 설립한 미국의 Gateway Design Automation 라는 미국의 회사를 설립하고 개발했다.

-1985년 : 이전까지는 표준화가 되지 않았기에 새 버전인 Verilog-XL과 그 시뮬레이터를 공개

-1987년 : VHDL이 IEEE 스탠다드 HDL로 지정

-1990년 : Gateway Design Automation 회사는 Cadence Design System이라는 회사에 인수되고 회사는 VHDL을 경계하며 Verilog를 공개하는 데 노력을 기울인다.

1991년 : Cadence 사는 OVI(Open Verilog Internation)를 만드는데 성공함.

1995년 : Verilog는 IEEE 스탠다드 1364-1995로 등록되어 공개됨

2001년 : 개정되어 IEEE 스탠다드 1364로 등록되고 2’s complement를 이용한 부호 변수가 사용 가능하게 됨.

이후로도 2005년, 2009년 계속 개정이 이루어졌고 Verilog는 현재까지도 전자 회로 설계 및 검증 구현에 있어 유용하게 쓰이는 언어이다.

**3.**

**3-1. C언어와 비교한 Verilog의 구조와 문법**

Verilog는 기본적으로 C언어랑 문법이 비슷하다. 예를 들어 if나 while문의 제어 구조도 같고 출력 연산자들도 비슷하다. 덕분에 쉽게 접근 가능하지만 블록을 중괄호 대신 Begin과 End를 이용해 구분하고 시간 개념이 포함된다는 차이점도 존재한다.

**3-2. Verliog module**

크게 머리부,선언부, 몸체부 등 3가지로 구성된다.

![텍스트이(가) 표시된 사진

자동 생성된 설명]()

머리부 : 키워드인 module을 선언하고 후 module의 이름과 port 목록을 선언하고 세미콜론을 통해 끝낸다. 주의할 점으로는 모듈의 이름으로 키워드 사용이 불가능하다는 것이다.

선언부 : module에서 필요로 하는 요소들을 선언하는 부분이다.register, parameter, wire 선언 뿐만 아니라 비트 폭이나 포트의 방향도 선언한다.

몸체부 : 회로의 기능과 구조를 나타내는 코드로 구성되어 있다.

**3-3. Verilog의 Data type**

1. Register : 데이터를 저장하기 위한 추상적인 저장 장치로 다른 값을 선언하기 전까지는 기존 값을 유지한다. Always나 initial 같은 절차형 할당문에 의해 값을 받는다.

-Integer : 정수형 변수

-Real : 실수형 변수

-time. real time : 시간 체크가 필요한 상황에서 simulation 시간을 처리하는 시간형 변수이다.

2) Net : 구성요소 사이를 물리적으로 연결하는 부분

- wire : 변수들이 모듈 내에서 어떻게 연결되었는지 나타내주는 변수

- tri : 선을 연결할 때 사용

**3-4. Verilog의 상수 선언**

크게 두 가지 방식의 선언 방식이 존재한다.

1. Size를 정하지 않았을 때

선언 방식은 (비트수)’(입력형식)(입력값)이고 앞의 입력 형식은 입력값이 몇진수로 표기되었는지 나타내는 부분이다. 입력 형식은 16진수는 h, 8진수는 o, 2진수는 b, 10진수는 b로 나타낸다.

간단히 예를 들어보면 다음과 같다.  
h18A; -> hexademical, 즉 16진수 수 18A

1. Size를 정했을 때

선언 방식은 (비트수)’(입력형식)(입력값)이고 비트수는 10진수로 표현된다.

간단히 예를 들어보면 다음과 같다.

4’h13B; -> 4비트의 16진수 수 13B

1. 음수를 표현할 때

선언 방식은 (부호)(비트수)’(입력형식)(입력값)이다. 단순히 -를 비트수 앞에 붙이면 된다.

간단히 예를 들어보면 다음과 같다.

-8’d12; -> 8비트의 10진수 -12

**3-5. Verilog의 연산자**

산술 연산자 : +,-,\*,/

관계 연산자: >,>=,<,<=

논리 연산자 : &&, &, ||, !, ~, |

결합 반복 연산자: {},{{}}

기존의 C언어와 상당 부분 일치한다. 그러나 c언어에서는 볼 수 없었던 중괄호와 겹중괄호로 이루어진 결합, 반복연산자가 있다. 결합 연산자는 중괄호 내의 변수들을 이어주는 역할을 한다. 그리고 겹중괄호는 {상수{변수}}와 같은 형태로 쓰이는데 변수를 상수 숫자만큼 반복해 결합해준다.

**3-6. Verilog의 Timescale**

파일 내의 시간단위와 정밀도를 결정하는 Timescale 구문의 형식은 ‘timescale (시간단위)/(정밀도)이다. 여기서 시간단위는 파일 안의 모든 시간 단위를 해당 시간 단위로 바꾸고 정밀도는 주어진 시간 단위로 구성 가능한 가장 작은 지연을 의미하며 시간 단위의 소수점 범위를 보여준다.