2주차 예비보고서

전공: 아트엔테크놀로지 학년: 3학년 학번: 20191172 이름: 함승우

1.HDL이 무엇인지 조사하고 Verilog이외의 HDL에 대하여 조사하시오.

HDL(Hardware Description Language)로 디지털 하드웨어를 설계하는 언어이다. 전자회로를 정밀하게 기술하는데 사용하는 컴퓨터 언어이다. 보통 디지털 회로 생성을 위해 사용되기 때문에 FPGA와 함께 사용된다. HDL을 사용하면 디지털 회로의 구조 및 동작을 High-Level에서 설계를 할 수 있다. 코드를 짜고, HDL을 사용하여 시뮬레이션을 설계한 논리 구조가 제대로 동작이 되는지 확인할 수 있고 바로 오류를 고칠 수 있다.

1-1. Verilog이외의 HDL

HDL에는 Verilog를 제외하고, VHDL, AHDL이 있다. 특히 VHDL, Verilog는 IEEE에서 표준으로 공인되었다. VHDL은 미 국방부에서 주문형 회로를 문서화하기 위해서 만든 언어이지만, 이러한 특성으로 현재 디지털 회로의 설계, 검증, 구현 등의 용도로 사용되고 있다. 그 이외에는 AHDL이 있다. AHDL은 altera hardware description language로 알테라 사가 개발한 하드웨어 기술 언어이다. VHDL과 비슷하게 작동한다.

2.Verilog의 역사와 발전 과정을 조사하시오

- 1983년 Gateway Design Automation사에서 하드웨어 기술언어인 Hilo와 c언어의 특징을 기반으로 Verilog를 개발했다.

- 1991년에 Cadence Design Systems에 인수된 verilog는 Open Verilog International라는 이름의 조직을 구성하고, Verilog HDL을 공개했다.

-1995년, IEEE 표준화로 결정되었다.

-IEEE 1800~2005, 2009로 개정되는 등 확장이 지속적으로 이루어지고 있다.

3.Verilog의 기본적인 구조와 문법에 관하여 조사하시오.

Verilog은 기본적으로 C언어 기반으로 개발되었다. 따라서, C언어와 문법이 유사하다. C언어와 마찬가지로 변수를 선언하고, if문이나 while문 같은 반복문의 구조도 동일하고, 연산자들 매우 비슷하다. 하지만 C언어와는 다르게 begin~end로 구분한다. (c언어는 중괄호 {, }로 구분함). 또한, HDL의 가장 큰 특징점인 시간에 대한 개념이 포함되었다.

* Verilog module에 대하여

Verilog의 module에 대해서 설명하자면, verilog은 크게 머리부, 선언부, 몸체부 등 세 부분으로 나눌 수 있다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| 머리부: 회색으로 표시된 부분이 머리부이다.  이때 module은 Verilog의 기본 구성단위이다.  Module {모듈 이름} ({fport 목록}); 이해 맨 마지막에 endmodule로 module의 끝을 알려야 한다.  또한, module의 이름은 대소문자 구분,\_ 가 사용이 가능하다. |
| 선언부: 머리부 아래의 모서리가 둥근 박스 안에 설명되어 있는 부분인데, module에서 필요한 것들을 선언한다. 앞에서 언급했듯이, c언어의 변수 선언하는 것과 매우 유사하다. |
| 몸체부: 설계할 FPGA의 기능, 동작, 구조등을 표현한다. |

* Verilog Data Type

Verilog의 데이터 타입으로는 Register와 연결하는 선인 Net으로 이루어져 있다. Register는 값을 받아서 저장할 때 사용되고, Net은 디바이스의 물리적인 연결을 나타낼 때 나타낸다. Register는 다른 값을 선언하기 전까지는 기존 값을 유지한다. Net은 신호를 연결할 때 사용한다. Register 자료형에는 reg, integer, time, real 등이 있다. reg은 일반적으로 사용하는 register로, input을 표현할 때 주로 사용된다. Integer는 정수 표현을 하는 register이다. time은 HDL의 특징인 시뮬레이션 시간을 저장하는 레지스터이다. real은 실수 표현을 하는 레지스터이다. 레지스터 time을 제외하고는 C언어와 매우 유사해 보인다. Net에는 wire, tri가 있다. wire은 변수들이 module 내에서 어떻게 연결되었는지 나타내 주는 변수로 보통 output을 나타낼 때 쓰인다. tri는 선을 연결할 때 사용한다.

* Verilog의 상수 선언

비트수 제한이 있는 reg의 값 선언 시 형식 (비트 수)’(입력형식)(입력값)

예를 들어 16’b0000\_1000\_1000\_0000의 의마는 <16비트><이진수><이진 값>을 의미한다.

이때, 입력형식을 살펴보면 d는 10진수, h는 16진수, o는 8진수, b는 2진수, s는 부호가 있는 수로 해석하면 된다. 한 가지 더 예시를 들어보면 12’H3aD는 16진수인 수 3aD를 12비트로 나타내는 것이다. (비트 수)에 값을 생략해도 된다, Size를 정하지 않아도 선언이 가능하다. 또한 부호가 있는 경우에도 표현 가능하다, -8’d6의 경우에도, 10진수로 표현된 -6을 8비트로 표현 가능하다. 음의 값은 2의 보수로 처리 되어 저장된다.

* Verilog의 연산자

산술 연산자에는 +, -, \*, /, \*\*, % 등이 있고, 관계 연산자에는 >. >=, <, <=, 결합 연산자에는 {},{{}}가,shift 연산자 또한 있다. C언어와 매우 비슷하다.

|  |  |  |  |
| --- | --- | --- | --- |
| 연산자 | 기능 | 연산자 | 기능 |
| {}, {{}} | 결합, 반복 | ^ | Bit xor |
| +, -, \*, /, \*\* | 산술 | ^~ , ~^ | Bit xnor |
| % | 나머지 | & | And |
| >, >=, <, <= | 관계 | ~& | Not and |
| ! | 부정 | | | Or |
| && | 논리 and | ~! | Nor |
| || | 논리 or | ^ | Xor |
| == | 논리 등가 | ^~ , ~^ | 축약 xnor |
| != | 논리 부등 | << | 왼쪽 시프트 |
| === | Case 등가 | >> | 논리 오른쪽 시프트 |
| !== | Case 부등 | <<< | 산술 왼쪽 시프트 |
| ~ | 비트 부정 | >>> | 산술 오른쪽 시프트 |
| & | 비트 and | ? : | 조건 |
| | | 비트 inclusive or | Or | Event or |

* Verilog의 Time

파일 내의 시간 단위와 정밀도를 결정하는 Timescale 구문형식은 ‘timescale <시간 단위>/<정밀도>’이다. 시간 단위를 선언하면, 그 파일 내의 모든 시간 단위가 정해진다, 정밀도는 주어진 시간 단위로 구성할 수 있는 가장 작은 지연을 나타낸다. 만약 timescale 1ns/10ps라는 구문이 있다고 가정하고, 다음 구문을 살펴 보자

assign #5 d = a ; 이 구문의 경우에는 기본 단위가 1ns임을 timescale 구문을 통해서 알 수 있었는데, 이 구문에 걸리는 딜레이가 5ns임을 알 수가 있다.