2주차 예비보고서

전공: 수학과 학년: 4학년 학번: 20171273 이름: 심현우

1. **HDL이 무엇인지 조사하고 Verilog이외의 HDL에 대하여 조사하시오.**

HDL이란 Hardware Description Language의 약자로 말그대로 하드웨어의 전자회로를 기술하기 위해 사용하는 언어입니다. 앞서 1주차에서 조사했던 FPGA의 회로를 설계하기 위한 프로그래밍을 하는 데 쓰이는 언어입니다. 회로의 동작과 구조를 기술하고 FPGA가 잘 작동하는지 검증할 수 있습니다.

HDL은 소프트웨어 프로그래밍 언어, 즉 C언어나 Python과 달리 하드웨어를 설명하기 위해 시간 개념을 포함하고 있습니다. 따라서 시스템의 구조를 나타내고 clock수에 따른 동작에 대해 기술합니다. Verilog가 도입되기 전 초기 HDL은 이 소프트웨어 프로그래밍 언어와의 차이를 줄이지 못하고 HDL의 자체의 언어를 다룰 수 있어야 했습니다. 하지만 Verilog는 C++을 simulator에 통합하여 객체 지향 프로그래밍을 사용하고 시간을 다룰 수 있게 되었습니다.

HDL을 이용하여 프로그래밍, 회로 설계는 합성 단계에서 마무리됩니다. 개발자는 전체적인 틀을 잡은 후 HDL을 사용하여 제어 및 연산을 구성하고 회로를 설명합니다. 그 후, HDL 코드를 통해 코드 및 회로를 검토합니다. 이 검토하는 과정에서 오류 및 오류를 유발할 수 있는 코드를 식별하여 개발자가 수정할 수 있도록 합니다. 검토가 마무리되면 합성을 진행합니다.

1. **Verilog의 역사와 발전 과정을 조사하시오.**

Verilog는 HDL의 한 종류로 1번에서 조사했듯 하드웨어를 기술하는 언어입니다. 초기 HDL과 다르게 verilog는 그 자체로 시간과 신호 강도를 기술할 수 있기 때문에 C언어와 유사하게 소프트웨어 프로그래밍 하듯 사용할 수 있습니다. Verilog는 1984년 초에 만들어졌습니다. 초기 Verilog는 시뮬레이션을 기술하기 위해 사용되었습니다. 현재의 verilog의 장점인 자동으로 합성하는 과정은 verilog가 만들어지고 사용된 이후입니다.

초기 Verilog 이후 1995년에 Verilog-95가 만들어졌습니다. 이는 Verilog 사용량이 증가함에 따라 범용성을 늘리기 위하여 Verilog를 표준화하기 위해 만들어졌습니다. 이후 Verilog 2001이 만들어졌는데 이는 95의 단점을 보완하기 위해 만들어졌습니다. 이 때, 95는 연산 과정에서 변수와 연산자 등 여러 부분에 단점이 많았습니다. 이를 보완하기 위하여 변수를 더 명시적으로 표현할 수 있고 연산자 또한 기술하기 쉽도록 만들었습니다. 연산자는 +, -, /, \* 등을 사용하였습니다. 이 후, 현재 사용되는 systemverilog가 만들어졌습니다. 이는 위에서 언급했듯 시간과 신호 강도를 포함하여 소프트웨어 프로그래밍 하듯 하드웨어를 프로그래밍하고 검증할 수 있는 언어가 되었습니다. 설계와 검증 단계에서 기능이 발달하고 범용적으로 사용되고 있습니다.

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

Verilog의 기본적인 구조는 모듈, 포트, 인스턴스로 구성됩니다. 간단하게 설명하면 모듈은 기본적으로 변수와 함수를 설명하는 부분입니다. 포트는 장치들 사이의 인터페이스 역할입니다. 인스턴스는 객체를 의미하며 모듈과 함께 프로그래밍에 쓰이게 됩니다.

모듈은 머리부, 선언부, 몸체부로 구성하는데 머리부에서는 모듈의 이름과, 포트목록으로 구성됩니다. 선언부에서는 머리부에서 있는 포트 목록의 포트들을 세부적으로 방향, 비트폭 reg와 wire을 각각 선언합니다. 몸체부는 말그대로 몸체로 회로를 직접적으로 설계하는 부분이며 변수, 연산등을 통해 함수를 구성하고 회로의 기능과 동작을 선언합니다.