1주차 예비보고서

전공: 컴퓨터공학과/생명과학과 학년: 9학기 학번: 20181435 이름: 박다희

**1.**

FPGA란 Field-Programming Logic Array이고 디지털 논리 회로를 구현하기 위하여 1985년 Xilinx사에 의해 소개된 하드웨어 장치이다. 이것은 사용자가 의도에 맞게 프로그래밍하여 여러 디지털 논리 기능을 설정할 수 있는 유연한 논리 소자 배열의 일동이다. 보통 설계가 끝난 하드웨어의 성능이나 그 동작 결과를 확인하기 위해 제작의 중간 개발물의 집적 회로(IC)이다. 다른 보통 반도체보다는 그 크기가 크지만 중간에 하드웨어를 검증하면서 고치고 업데이트한 논리회로를 새로 확인해 볼 수 있기 때문에 보통 반도체를 개발하는 단계의 초반에서 중반으로 넘어갈 때 주로 사용한다. FPGA를 구성하는 주된 요소는 논리 게이트, 래치 및 레지스터 그리고 링크 및 연결로 크게 나눌 수 있다. FPGA는 수천, 수백만개의 AND, NOR, OR, XOR등의 다른 여러 논리 게이트들을 포함하고 있고 데이터를 저장하고 처리하기 위한 레지스터나 래치같은 저장소 역할을 하는 요소도 포함하고 있으며 이러한 논리 게이트와 저장소를 연결해주는 많은 배선이 포함되어 있다.

대부분의 FPGA는 프로그래밍이 가능한 논리 요소에 메모리 요소와 간단한 flip-flop을 가지고 있는데 FPGA를 위한 프로그램을 작성하기 위해서는 HDL(Hardware Description Language)를 사용해야 한다. 대표적인 종류로는 Verilog와 VHDL이 있는데 이를 사용하여 코드를 작성하고 이 코드를 비트 파일로 변환하여 FPGA를 구성하게 하면 된다.

FPGA를 사용하는 법은 다음과 같다. 먼저 초기 설계 입력 단계로, 그래픽적인 입력과 문자적인 입력으로 나눌 수 있다. 그래픽적인 입력에서는 schematic capture tool로 논리 회로를 입력하는데 이 tool의 내부 라이브러리는 FPGA의 칩을 바탕으로 이미 설계된 회로들이 구성하고 있기 때문에 효율적이고 안정적이다. 문자적인 입력은 앞서 언급한 VHDL이나 Verilog HDL을 이용하는 입력인데 전체적인 논리는 파악하기 위해 자주 사용된다. 다음은 Register-transfer level simulation (RTL simulation)단계이다. 이 단계는 앞 단계에서 작성한 소스 코드를 simulation을 통해 검증하는 단계로 register가 데이터를 읽고 쓰며, flip-flop으로 구성되어 있어서 회로를 clock 신호를 통해 동기화한다. 단지 HDL로만 보면 디지털 회로가 실제로 어떻게 구현되는지 파악하기가 어렵지만 register 단계에서는 대략 알 수 있다. 보통 synthesis 단계는 디버깅도 어렵고 시간이 오래 걸리기 때문에 synthesis 전에 simulation 검증 단계를 거친다. 다음 단계는 synthesis단계인데 사용자가 구현한 high-level 코드를 low-level인 netlist형태의 데이터로 변환하여 FPGA가 이해할 수 있게 하는 단계이다. 이렇게 synthesis과정이 끝나면 place&route 단계로 넘어가는데 회로가 실제적으로 구현된 것이 칩에서 이루어지는 단계이다. Place는 FPGA의 특정 logic block위에 mapping된 block들을 배치해주는 것이고, route는 회로가 연결된 정보에 따라서 전기적 스위치나 연결선을 사용하여 배치 완료된 block들을 상호 연결하는 것을 의미한다. 여기까지의 단계가 끝나면 마지막으로 설계를 검증하고 그 성능을 추정하는 단계로 넘어간다. 앞선 단계인 place와 route 후에 회로의 실제적인 형태를 가지고 있는 상태에서 timing simulation이 이루어진다. 이 과정은 place와 route후 cell들을 연결했을 때 wire의 길이가 달라져서 delay가 될 수 있는 상황이 생길 수 있는데 이러한 상황에서 timing 오류 없이 회로가 잘 작동되는지 확인하는 과정이다. 이 과정을 통해 우리는 검증과 확인 뿐만 아니라 특정 조건의 상황에서 성능은 어떻게 되는지 확인하고 후에 어떤 성능을 가질 수 있을지도 추정할 수 있다.

**2.**

먼저 FPGA의 장점은 다음과 같다. 첫번째로는 프로그램 가능성이다. FPGA는 원하는 디지털 논리 회로를 만들기 위해 하드웨어를 프로그래밍할 수 있다 이를 통해 하드웨어의 설계를 소프트웨어적으로 바꿀 수 있으며 이는 하드웨어를 필요한 만큼 업그레이드하거나 빠른 개발 속도, 빠른 수정을 가능하게 해준다. 따라서 장기적으로 유지할 수 있게 된다. 두번째 장점으로는 고성능이다. 애초에 FPGA는 높은 성능의 계산을 하기 위해 설계되었고, 병렬 처리를 할 수 있기에 데이터의 양이 많고 계산이 복잡하여도 실시간으로 처리가 가능하다. 따라서 이러한 장점은 고성능 응용분야인 암호화나 딥러닝 분야에서 유용하다. 세번째 장점은 FPGA는 작업에 필요한 전력만 사용하기 때문에 에너지 효율성이 좋다는 점이다. 네번째 장점은 FPGA는 하드웨어 레벨에서 작동하기에 레이턴시가 높지 않고 응용 프로그램이 실시간으로 돌아가야 할 때 적합하다는 점이다. 마지막으로는 프로세서를 바탕으로 하는 시스템은 하나의 작업을 한 번에 실행하고 시간 결정적 작업의 위험한 상태에서 서로가 서로를 선점하기 때문에 하드웨어의 안정성을 높일 수 있다.

다음으로 FPGA의 단점은 다음과 같다. ASIC(응용 특정 통합 회로)와 비교했을 때 많지 않은 양에서 장기적인 관점에서는 FPGA를 사용하는 것이 비용을 절감할 수 있지만 FPGA의 양산 단가가 ASIC의 양산 단가보다 비싸기 때문에 처음 구축할 때 대규모 생산을 한다면 비용적으로 비쌀 수 있다. FPGA의 리소스는 제한적이라서 큰 부피의 논리 회로 같은 작업을 구현하려 할 때에 여러 개의 FPGA가 필요할 수 있다. 또한 FPGA의 칩의 사이즈가 크기 때문에 크기가 작은 제품에는 FPGA로 설계하는 것이 어렵다는 단점이 있다.

마지막으로 FPGA를 활용하는 주된 분야는 디지털 신호 처리, 임베디드 시스템, 고속 트레이딩 그리고 과학 및 의료분야이다. 또한 우리는 어떤 제품의 설계를 완성하고 나서 기능을 업데이트 할 때에 하드웨어를 바꾸지 않아도 업데이트가 가능한데 이 과정은 microprocessor를 사용하는 시스템에서 ROM프로그램을 업데이트하거나 IC칩 기능을 업데이트 시키며 이루어지는데, 이때 IC칩 자체적인 기능을 업데이트 시키는 것을 가능하게 해주는 것이 FPGA이다. 디지털 신호 처리에서는 통신 시스템이나 음성,영상 처리 그리고 레이더와 같은 디지털 신호를 처리하고 응용하는 분야에서 많이 사용되고, 임베디드 시스템에서는 하드웨어의 가속을 위해 FPGA를 많이 사용하여 성능 향상과 전력의 효율성을 높일 수 있다. 또한 고속 트레이딩에서는 FPGA를 사용하면 1초에 수백만 건의 거래를 처리할 수 있고, 과학 및 의료 분야에서는 데이터를 수집하고 분석하기 위한 것으로 많이 사용된다.