|  |
| --- |
| **Architecture All Access: Modern FPGA Architecture** |
| * 동영상 내용 요약:   - 강의 개요  fpga는 프로그래밍이 가능한 회로로 언제든지 기능을 업데이트 혹은 완전히 변경할 수 있다. 80년대 중반에 개발되어 모든 응용 분야에 사용되고 있다. 지난 30년 동안 기술은 나선형, 모바일 빅데이터 및 고속 처리 등으로 변화하며 집중되어 왔다. 이는 FPGA을 활용하기에 알맞는 설정이라 할 수 있다.  - FPGA 개요  Fpga 개발자는 CPU를 설계하는 사람에 가깝다고 볼 수 있다. 흔히 아는 소프트웨어 개발자는 파이썬, c++ 등의 언어를 사용하여 특정 행위를 수행하게 하는 명령을 한다. 따라서 CPU는 해당 명령을 통해 기존의 회로를 활용하여 결과값을 얻는다. 하지만 FPGA 개발자는 이와 다르게 이 행위를 수행할 수 있는 디지털 회로를 제작하는 행위를 수행시키는 명령을 한다.  - 디지털 로직 개요  모든 디지털 논리 회로는 and게이트, or게이트, 인버터/not 게이트 이렇게 세 가지의 회로로 구성되어있다. 그리고 게이트는 반도체 재료로 만들어진 트랜지스터로 구성되어 있다. fpga를 위한 여러 개념 중 첫 번째는 클럭이다. 클럭은 설정 주파수의 상승과 하강하는 전압니다. 주기를 알면 특정 시간 동안 이 클록이 몇 번 상승 및 하강하는지 알 수 있다. 두번째는 플립플랍이다. D플립플랍은 클록이 0에서 1이 될 때, 즉 상승엣지에만 값을 저장한다. D의 입력이 달라지더라도 클록이 다시 상승하기 전까지는 출력이 대체되지 않는다.  - ASIC  한 회사에서만 사용하는 맞춤형 칩, ASIC는 우리 주변에서 흔히 볼 수 있는 다양한 전자기기에 포함되어 있다. 이는 기기에 최적화된 설계를 할 수 있지만 거액이 필요하고 시간이 오래 걸린다는 점, 그리고 실수나 수정 사항이 있을 경우 이를 다시 반복해야 한다는 단점이 있다. 따라서 FPGA는 ASIC만큼 빠르거나 낮은 전력을 소모하는 것은 아니지만 새로운 기능이나 새로운 애플리케이션에 대해 빠르게 최적화가 가능하다. 따라서 중간 규모의 애플리케이션에서 높은 비용적 이득을 얻을 수 있다.  - FPGA 빌딩 블록  FPGA의 기본 구성 요소는 LUT이라 불리는 조회 테이블로 플립플랍에 연결되어 있다. 이 테이블은 특정한 입력에 대해 특정한 출력을 제공한다. LUT과 플립플랍은 ALM이라는 구조로 통합된다. FPGA가 발전함에 따라 LUT도 더 작은 규모로 쪼개거나 단일로 작동하는 등 향상되었다. FPGA의 다른 구성 요소로는 라우팅 기능이 있다. ASIC에서는 플립플랍과 논리 게이트를 연결하는 라우팅이 물리적으로 구축된다. FPGA에서는 LUT과 플립플랍을 연결하는 프로그래밍을 할 수 있는 어레이로 설계되어있다. FPGA는 도시로 치자면 각 건물들을 놓고 해당 건물로 가기 위한 경로를 특정 도로를 활용하여 설정해놓은 것이라 볼 수 있다.  - FPGA 발전  FPGA가 발전함에 따라 다양한 하드웨어 기능이 추가되었다. On chip으로 write가 가능한 메모리 블록, 디지털 신호 프로세서, 주변장치와의 호환성 증대 등이 있다. 즉 위에서 설명한 도시가 여러개가 있고 해당 도시들을 연결하는 고속도로를 놓은 것이다. 도시 가까이에는 보관을 위한 창고를 두고 손쉽게 사용할 수 있다. FPGA는 다양한 기능을 갖추고 있지만 그만큼 중요한 프로그래머블 로직을 많이 사용하여 시장경쟁력을 놓칠 수 있다.  FPGA는 VHDL 혹은 Verilog 이 두가지 언어로 수행되는 경우가 많다. 하지만 하드웨어 설계 후 FPGA와의 연결을 위해 설계해야 하는 소프트웨어도 존재하며 FPGA에 CPU가 내장된 경우 CPU용 소프트웨어도 개발해야 한다. 최근에는 FPGA를 더욱 추상화하려는 노력을 보이고 있다.   * 느낀점: FPGA가 프로그래밍 언어인 줄 알았는데 강의를 통해 회로에 더 가깝다는 것을 알게 되었고, 이를 다룰 수 있는 언어가 verilog라는 것도 알았다. verilog를 들어본 적은 있지만 어떤 언어인지에 대해 더 자세히 알 수 있었고 VHDL과 같은 선상의 언어라는 것도 알게 되었다. 이런 개념들을 수업시간에 들어 알고는 있었으나 각자의 세세한 역할을 알게 된 것이 처음이라 많은 도움이 되었다. 또한, 그동안 개별적으로 배운 회로에 대한 지식과 프로그래밍에 대한 지식을 결합하여 새로운 개념을 배울 수 있어 다음 수업이 기대된다. |