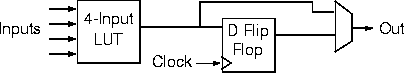
전공: 컴퓨터공학 학년: 2학년 학번: 20161603 이름: 신민준

1. FPGA란 무엇인지 조사하시오(활용법 사용법 포함)

FPGA는 Field Programmable Gate Array의 약자로, 프로그래밍 가능한 내부 회로나 PLD(Programmable Logic Device)가 존재하는 반도체 소자이다. PLD는 AND, OR, XOR 등의 논리 게이트 기능이나 더 복잡한 디코더 등의 조합 기능을 복제하여 프로그래밍할 수 있다. 보통 FPGA는 논리 블록에 간단한 Flip-Flop, 또는 기타 메모리 요소들을 포함한다. FPGA의 논리 블록은 원하는 대로 브레드보드와 같이 내부연결이 가능하다. 이러한 논리블록, 내부선은 소비자나 설계자가 제조공정 이후 프로그램할 수 있기에 어떠한 논리 기능도 수행할 수 있고, 이 특성을 현장 프로그래머블한 특성이라 한다. 대다수의 현대 FPGA는 이렇게 ‘동작 시간’ 내에 재프로그래밍이 가능하며, Reconfigurable Computing/System의 중앙 처리 장치에 대한 아이디어를 이끌었다.

일반적인 기본 구조는 Configurable Logic Blocks 배열과 라우팅 채널로 구성된다. FPGA의 Logic Block은 일반적으로 다음과 같이 입력 룩업 테이블(Lookup Table) 4개와 플립플롭의 조합으로 구성된다.



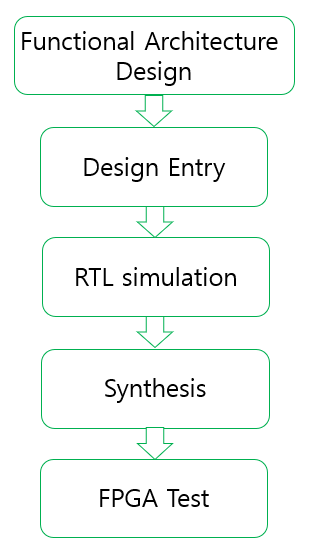
현대로 오면서 반도체 공정 기술이 진보하게 되면서 이에 따라 현대의 FPGA 계열은 높은 수준의 기능들을 지니게 되어, 초기 버전보다 그 속도와 기능이 크게 향상되었다. 예를 들자면, 곱셈지, 임베디드 프로세서, 임베디드 메모리 등이 있다. 또한, FPGA는 시스템 유효성 검사에 널리 사용되고 있다. 최근에는 칩이 공장에서 생산되기 전 사전에 칩의 설계를 검증하는 회사들이 설립되면서 개발 기간도 짧아지는 추세이다.

최근의 FPGA 개발 경향은 Coarse-grained(성긴) 구조적 접근을 채택하는 추세이다. 즉, 전통적인 FPGA의 논리블록과 내부 선을 인베디드 프로세서와 그 주변기기들에 연결해, ‘프로그래머블’한 칩으로 이루어진 시스템을 만드는 방식이다. 또 다른 접근 방법으로는 내부 FPGA 논리로 구현한 소프트 프로세서 코어를 사용하는 방식이 있다. 하드 중앙 처리장치 코어는 중앙 처리 장치 시스템이 물리적으로 내장되어있기 때문에, 소프트코어 중앙 처리 장치보다 일반적으로 성능이 우월하다.

FPGA를 설계/프로그래밍 할 때, FPGA의 동작을 정의하기 위해 하드웨어 기술 언어(HDL)나, 도면 설계를 제공한다. 일반적으로 널리 쓰리는 하드웨어 기술 언어에는 VHDL, 그리고 Verilog가 있다.

넷리스트를 생성해 설계를 기술적으로 매핑하려면 전자 설계 자동화 도구를 사용한다. 배치 – 배선이라 불리는 작업을 거치며 넷리스트를 실제 FPGA에 더 적합하게 변경할 수 있으며, 일반적으로 FPGA 회사에서 제공하는 자산인 배치 – 배선 소프트웨어로 이를 수행한다. 이 결과를 사용자는 시뮬레이션, 맵, 타이밍 분선을 통한 배치와 배선 등의 다른 검증 방법론으로 검증을 하면, 이렇게 검증 과정이 완료되어 생성된 이진 파일을 FPGA의 재설정에 다시 사용한다.

FPGA를 사용해 실험을 할 때에는 다음의 도식에서 보이는 것 과 같은 실험 과정을 거친다.



Functional Architecture Design에서는 이론과 Schematic을 사용해 디자인을 설계한다. Design Entry 단계에서는 HDL(Hardware Description Language)를 사용해 설계하고, 이 단계에서 사용자가 구현한 것들을 시뮬레이션으로 통해 검증하는 RTL Simulation 단계를 거친다. Synthesis, 즉 합성 과정은 유저가 구현한 High level 디자인을 FPGA 보드가 이해 가능한 Low level 디자인으로 바꿔주는 단계이다. 마치 C 언어에서 소스 코드를 실행파일로 바꿔주는 Compiling 단계가 있듯이, FPGA에서는 High Level의 디자인을 Low Level 디자인으로 바꾸는 Synthesis 단계가 있다고 볼 수 있다. 이후, 실제 FPGA 보드를 사용해 이 디자인을 테스트 하는 FPGA Test 과정을 거친다.

FPGA에서 디지털 회로를 구현하는 것에 있어서, 가장 기초적인 구성요소 두 가지는 Lookup Table(LUT – [럿])과 Flip-flop이다.

예를 들어, 3-input LUT이 있다 가정할 때, 3개의 input인 A, B, C로 조합 가능한 모든 Boolean Function을 구현할 수 있다. 왜냐하면, 이름의 Table이 보여주듯이, 각 input들로 조합할 수 있는 모든 진리표-Truth table을 내장하고 있기 때문이다. 따라서, 이렇게 LUT들의 조합으로 AND gate, XOR gate 등의 모든 디지털 회로를 구현할 수 있다.

예를 들자면, Xilinx 사의 FPGA인 Vertex 5 LX85에는 6-input LUT가 약 5만개 정도 들어있다. VHDL로 구현할 코드를 작성하면, 이 FPGA의 LUT들이 이 코드를 수행하기 위해 조합된다.

다음은 FPGA에서 흔히 사용되는 용어들이다.

* Synthesis(합성) – HDL로 짠 코드를 마치 컴파일하듯이 실제 디지털 회로로 칩에 구현하는 과정이다. 수많은 LUT들은 이 때 사용된다.
* Netlist – synthesis 이후 나오는 결과 파일이다.
* Delay – 총 delay = cell\_delay + net\_delay로 계산되며, cell은 하나의 AND-gate나 Flip-flip을 칭하고, net은 이들을 서로 연결하는 와이어를 일컫는다. 하나의 cell의 상태가 0에서 1이 될 때 까지의 Delay를 rise\_delay, 그리고 반대로 1에서 0이 될 때 까지의 Delay를 fall\_delay라고 한다.
* Skew – 위에서 설명한 여러 delay들의 총 합을 말한다. 예를 들어, FPGA 내부에 여러 Flip-flip들과 이들로 가는 clock signal이 있다면, 가장 빨리 도착한 clock과 상기한 여러 delay들로 인해 늦게 도착한 clock 사이의 차이를 말한다.
* Slack – Delay를 제외한 후 남는 여유 시간이다. 예를 들자면, 100ns 주기의 clock이 있는 시스템에서 10ns의 delay가 존재한다면, 여기서의 slack은 100ns에서 10ns를 뺀 값인 90ns가 된다.
* Slew – 디지털 신호가 0에서 1, 1에서 0으로 변화할 때의 그 기울기를 말한다. 이 기울기가 너무 크면, 즉 전압이 급변하면, EMI 노이즈가 생기거나 순간적으로 강한 전력이 사용되므로 다른 FPGA 소자들에게 간섭이 일어날 수 있기 때문에, 이 slew rate를 적당한 값으로 조절할 필요가 있다.

1. FPGA의 장단점 및 활용 분야를 조사하시오

FPGA는 주문형 반도체(ASIC)보다 느리고, 설계가 복잡한 경우 사용할 수 없으며, 소비전력도 비교적 크다는 단점이 있다. 그러나, 개발하는 데 드는 시간이 비교적 짧고, 오류를 곧바로 현장에서 수정하는 것이 가능하며, 초기에 드는 개발비가 저렴하다는 장점이 있다. ASIC는 프로그래밍이 불가능한 양산형 집적회로이고, FPGA에 비해 성능, 전력소모, 칩 가격 등 모든 면에서 FPGA보다 뛰어나지만, 개발에 드는 비용이 앞서 말했듯 어마어마하기 때문에 수요가 크지 않은 이상 일반적으로 FPGA를 ASIC에 비해 일반적으로 선호한다.

FPGA는 알파고 같은 로봇제어, 우주선, 인공위성, 가상화폐 대두에 따라오는 가상화폐 채굴기, 현대전 군사 무기, 프로토타입 테스트 등 압도적으로 빠른 하드웨어 프로그래밍을 요구하는 현장에서 주로 사용된다.