

PLD

林 滿 稷

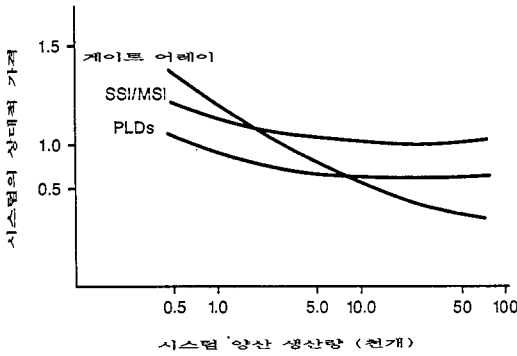
韓國 MJL(株) 代表理事

I. 서 론

PLD(programmable logic devices, 프로그램 가능형 논리 디바이스)란 PAL, PLA, FPLA, PLD, EPLD, EEPLD, LCA, 그리고 FPGA등을 총칭하며, 이는 일정한 하드웨어적 IC(integrated circuit, 集積回路)에 주문형 논리회로를 프로그램하여 탑재시킬 수 있는 구조상의 융통성을 가진 디바이스이다. 따라서, 표준형 부품을 사용하여 특수용도의 회로를 구성할 수 있게 하여, 기능상 ASIC의 역할을 담당케 한다. 게이트 어레이나 표준셀(standard cell) 같은 ASIC을 제조하는데 필요한 오랜 개발시간, 높은 툴링(tooling) 경비, 복잡한 구매상의 경로, 그리고 재고비치등의 문제점을 제거하고, 특정의 논리기능을 갖는 이 디바이스는 PC 혹은 워크스테이션을 이용하여 사내(in-house)에서 손쉽게 설계하여 제작할 수 있다.

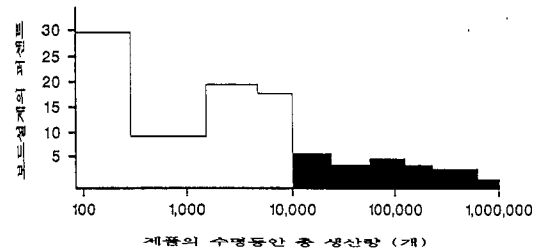
그림 1에서는 몇가지 논리설계 기술의 가격대 시스템 생산량을 표시한다. 이 그림에 의하면, 시스템 생산량이 약 10,000개 이하의 경우, PLD가 가장 경제적이고, 그보다 생산량이 큰 경우에는 게이트 어레이가 더 경제적으로 나타나 있다.

그림 2에서는, 프린트 서킷보드의 수명기간(lifetime) 중 그 총생산량이 10,000개 이하인 경우가 전체 보드설계의 약 75%를 차지하고 있다. 따라서, 이 부분은 PLD가 가장 경제적인 영역이다.



자료출처 : ICE Corporation, 1988

그림 1. 논리 기술 방법의 가격 대 생산량



자료출처: The Technology Research Group, Inc., 1988

>75% 가 10,000 이하

그림 2. 프린트 서킷 보드의 생산량

논리 집적도가 높은 PLD를 CPLD(complex PLD, 복합형 PLD)라고도 칭하며, 이는 IC중 가장 생산량 성장률이 큰 품목으로^[1] 나타나 있다. CPLD는 게이트 어레이의 낮은 논리집적도 부분에 접하고 있다. 1990년도에 게이트 어레이 설계 시작의 30%는 6000 gate 이하로 CPLD가 대체할 수 있음을 시사한다.(그림 3 참조)

PLD가 사용할 수 있는 작동속도(usable speed, f_{cnt})는 약 100MHz에 달하고, 현재 나와 있는 사용할 수 있

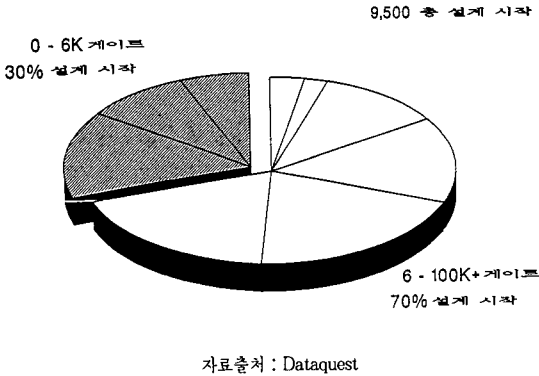


그림 3. 북미에서의 MOS 게이트 어레이 설계 시작 분포(1990년)

는 논리집적도(usable density)는 디바이스당 약 20,000 gate에 달한다. 현재 연간 PLD 시장은 9억불로서,^[1] 만일 PLD가 게이트 어레이보다 세배 혹은 그 이하의 가격인 경우, 설계자는 게이트 어레이보다 PLD를 선호한다고 한다.^[2]

이와 같이 ASIC에서 CPLD는 중요한 위치를 차지하고 있다. 제 2장에서는 PLD를 ASIC의 일환으로 본 관점에서의 분류를 논하고, 제 3장에서는 PLD의 대표적인 2가지 부류를 비교한다. 제 4장에서는 현시점에서 집적도, 성능, 핀수등이 가장 많은 MAX 7000 EPLD를 예로 들어 PLD의 구조, 작동, 설계 방법을 논하고, 제 5장에서는 PLD의 장래 전망을 논한다.

II. PLD의 분류

PLD는 관점에 따라 여러가지로 분류할 수가 있다. IC를 디바이스 하드웨어 관점에서는 크게 나누어 표준형과 ASIC(application specific IC, 특정응용형 IC)으로 분류할 수 있고, ASIC이 주문형 IC에 반해 PLD는 표준형 IC이다. 그러나, 일단 특정응용을 프로그램하여 탑재한다는 소프트웨어 관점에서는, 프로그램후의 PLD는 ASIC의 역할을 한다고 볼 수 있다. "ASIC"은 흔히 논리집적도가 최소한 수천개에서 10만개 수준의 게이트 어레이나 표준셀을 의미한다. 따라서, PLD를 ASIC에 분류시키는데는 게이트 수가 수천개에서 2만개 수준에 달하는 EPLD나 EEPLD 같은 CPLD와 FPGA(field

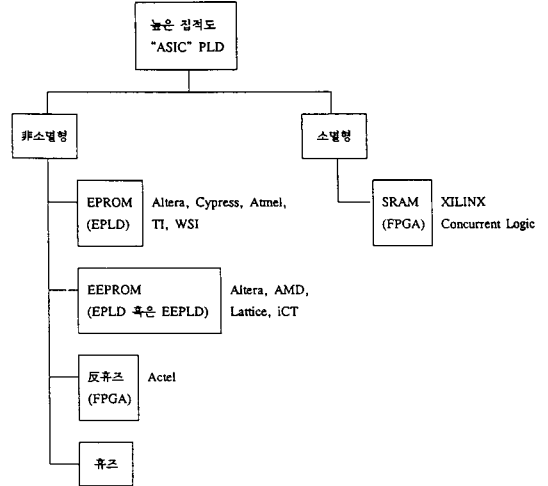


그림 4. 높은 집적도 PLD의 구분

programmable gate array, 현장 프로그램 가능형 게이트 어레이)의 두가지를 이 글에서는 의미한다. 이들 PLD를 프로그램시 연결 방법에 따라 분류한다면, 非소멸(non-volatile)형과 소멸(volatile)형으로 나눌 수도 있다.(그림 4 참조)

非소멸형에서는 전원을 제거하여도 프로그램이 소멸되지 않게 하는 EPROM, EEPROM, 퓨즈, 反퓨즈(anti-fuse) 등을 사용하며, 소멸형에서는 전원을 제거하였을 때 프로그램이 소멸되는 SRAM을 사용한다. 이들 제품은 EPLD, EEPLD, FPGA의 디바이스 품목별로 구분되며, 각 구분당 제품 제작 회사도 그림 4에 제시한다.

Altera사는 EPROM 기술을 처음 CMOS 공정과 결합시켜 EPLD^{[3][4]} (erasable programmable logic device, 소각성 프로그램 가능형 논리 디바이스)를 최초로 발명하여 1984년에 상품화 한후 EPLD와 EEPLD형 PLD의 기술과 시장에 주도 역할을 해 왔으며, 이러한 종류의 PLD를 "EPLD"라 총칭한다. 이중 특히 논리 집적도가 높은 MAX^[3,5,6] (multiple array matrix, 다중 배열 매트릭스) 계열이 응용상 ASIC으로 분류하는 PLD에 적합하다. 소멸형인 SRAM을 사용하는 기술과 시장은 Xilinx사가 주도해 왔으며, 이 PLD는 FPGA의 한 종류이다.

고집적도 PLD 시장의 대부분을 Altera의 MAX와 Xilinx의 FPGA가 비슷한 비율로 장악하고 있으며, 두 회사의 PLD의 구조가 근본적으로 다르다. 다음 장에서는 이 두가지 PLD의 근본적 차이점을 비교한다.

III. MAX와 FPGA의 비교

적은 집적도의 TTL, GAL 그리고 PAL을 사용하는 논리 설계에 비하여, 높은 집적도의 PLD, 즉 MAX와 FPGA를 사용하는 설계에서는 시스템의 성능(즉 작동 속도), 디바이스의 효율적 사용, 설계와 오류 수정의 용이성이 중요하게 고려되어야 한다. 시스템의 성능은 고집적도 디바이스의 형태 선택 뿐만 아니라, 그 디바이스의 기능을 얼마나 잘 구현시켰는가에도 좌우된다. 특히 논리 설계자는 그 디바이스에 구현시키는 기능의 타이밍 성격에 특별히 주의를 기울여야 하며, 이는 설계 주기의 초기에 해야 한다. 그 이유는 설계의 구현이 시스템 인터페이스 타이밍 때문에 고쳐져야 할 수도 있기 때문이다.

EPLD의 MAX 계열은 적은 수의 LAB(logic array block, 논리 배열 블록)이라는 비교적 큰 논리 기능의 집단이 넓은 통로인 PIA(programmable interconnect array, 프로그램 가능형 상호 연결 배열)로 연결되어 있다.(그림 5참조)

FPGA는 CLB(configurable logic blocks)이라는 많은 수의 비교적 작은 논리 기능의 집단이 좁은 통로인 채널로 연결되어 있다. 이러한 근본적 구조와 신호 전달의 차이점은 MAX와 FPGA의 상호 다른 장단점의 근원이 된다.

1. 성능

성능이란 신호 전달 속도가 얼마나 빠른가를 의미한다. 집적도가 낮은 설계에서는, 제조회사의 출판된 데이터를 보고 논리 지연을 신속히 산정할 수 있다.

프린트된 서킷 보드 (printed circuit board)에서는 보드의 연결 회로가 기여하는 신호 지연은 무시할 정도이기 때문에, 논리 디바이스를 통한 신호 지연만이 중요하다. 이 지연은 정해져 있고, 완전히 예측할 수가 있으며, 핀과 핀 사이의 지연이 받아들일 만한 경우, 패키지 내에서 어떻게 특정 기능이 구현되었는가를 따질 필요가 없이 각 신호가 지나갈 통로를 통한 논리 전달 지연을 계산하기 위해서는, 단순히 각 패키지의 지연을 합산하면 된다.

그러나, 높은 논리 집적도를 가진 PLD 내에서의 설계에서는 상호연결(interconnect) 지연과 기능(function) 지연의 두가지를 더 첨가해서 생각해야 된다. 상호 연결이란 이 논리 기능블락간의 신호전달을 의미한다. 전체 신호 지연중, 고집적도 PLD에서 이 상호연결 신호 지연이 중요한 요소가 된다. 기능적 지연은 논리 블락내에서의 신호 전달 지연이다.

FPGA에서는 그림 5에서 보는 바와 같이, 상호 연결 신호 지연이 누가적(累加的, cumulative)이고, 라우팅 때마다 변하고(variable), 라우팅 경로(routing path)에 따라 다르다. 블락간의 통로 자체가 좁기 때문에, 집적도가 큰 논리를 다룰때, 그 통로 자체가 모자라 라우팅이 어렵게 된다. 통로가 좁다는 의미에서, 이 통로를 흔히 채널(channel) 이라고도 부른다. 이 구조에서 또 하나의 문제는 논리 블락이 작기 때문에 어느 복잡한 기능을 수행하는 데는 많은 블락을 사용하게 되며, 몇개나 블락이 필요한 지를 예상하기가 어렵다. 몇개의 블락을 신호가 지나가느냐에 따라 지연시간에 직접 영향을 주기 때문에, 설계자가 지연시간을 예측하기 어려운 문제점이 된다.

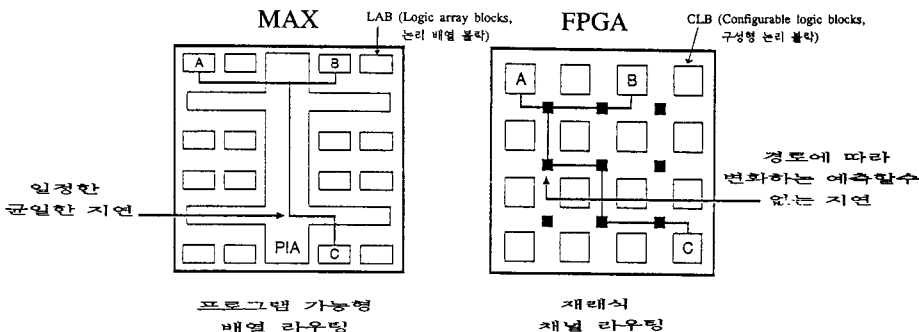


그림 5. 고 집적도의 PLD의 비교

그림 5에서 신호가 A에서 B로 갈때 보다 A에서 C로 갈 때 2배의 경로를 거쳐가기 때문에, 지연시간도에 비해 약 2배 더 걸린다. 설계를 변화시키면 라우팅도 변하기 때문에, 설계가 끝나기 전에는 이 지연시간을 미리 예측할 수가 없다. 그러나 설계가 끝날 때까지 지연시간의 계산을 미루는 것은 시간의 낭비를 초래할 수 있다. 즉 라우팅이 자동으로 되었을 때 선택한 디바이스가 필요한 지연 시간 조건을 만족시키지 못했을 경우, 원하는 결과를 얻기 위해서는 엄청난 시간을 들여 수동으로 라우팅 작업을 해야 한다. 뿐만 아니라, 이러한 노력후에 비로서 필요한 지연시간 조건을 그 디바이스로는 성취할 수 없음을 알게 될 수도 있다.

이와는 대조적으로, MAX의 배열 구조에서는 논리 블록 사이의 상호연결 신호지연은 일정하고 균일하다. 즉 그림 5에서 A와 B간과 A와 C간의 지연시간이 같고 일정하다. 논리 블록간 연결하는 신호는 그 통로 자체가 프로그램형으로 한 LAB에서 다른 LAB으로 신호를 전달하는데 필요한 통로만 PIA 내에서 프로그램되며, PIA 통로 자체가 대단히 넓은 배열로 구성되어 있기 때문에 통과할 신호에 언제나 통로가 부여된다. 즉 PIA로는 전용적 입력신호, 입출력핀의 귀환신호, 마크로셀의 귀환신호가 다 전달되나, 각 마크로셀이 필요로 하는 신호만을 각 LAB에 전달한다. 또한 논리 블록이 크기 때문에, 어느 기능을 구현하는데 몇개의 논리블록이 필요한가를 예측하기가 쉽고, 대개는 한 층의 논리 지연내에서 구현시킬 수가 있다. 따라서 배열 구조가 채널 구조에 비해, 성능면에서 장점은 자동 라우팅이 반드시 단시간에 성공적으로 되며, 신호 지연시간이 짧고 일정하며 예측할 수 있다는 점이다.

2. 디바이스의 효율적 사용

디바이스 제조회사가 제공하는 마크로 기능 라이브러리를 참조하면, 그 마크로 사용에 얼마만한 논리 자원이 필요한가를 예측할 수 있다. 이 논리 자원의 효율적 사용이 고집적도 PLD를 이용한 논리 설계에서는 중요한 역할을 한다.

작은 논리 블록의 집단으로 된 FPGA 구조는 레지스터가 많고 입력 단자수가 적은 경우에 더 효율적으로 쓰이는 경향이 있다. 그 반대로 큰 스테이트머신이나 카운터 같이 많은 입력 단자가 필요하고 논리 설계 위주의 경우는 작은 논리 블록 구조의 FPGA는 적합하지 않거나, 그 자원을 효과있게 사용하기 어렵다. 특히 큰 설계에서는, 모든 신호의 경로를 상호 연결하는데 필요한 연결 자원이 부족되는 수가 많다. 이는 디바이스가

거의 완전히 사용되려고 하면, 상호 연결하는 길이 많기 때문이다.

배열 구조의 MAX는 논리 블록이 크기 때문에 크고 복잡한 논리 기능을 구현하는 데는 효율적으로 사용된다. 또한 상호 연결 자원이 풍부하여 반드시 라우팅이 자동으로 완성되며 논리 재원을 충분히 이용하게 한다. 그러나, 대단히 간단한 논리를 구현할 때는 재원을 낭비하는데가 있다. 레지스터를 주로 사용하는 경우가 이러한 경우에 해당한다. 새로운 MAX 7000 계열 구조에서는 공유형과 평행형의 연장논리 텀 구조를 넣으므로, 더욱 효율적으로 자원을 사용케 한다. 뿐만 아니라, 이 연장 논리텀이 논리텀만 아니라 레지스터도 구현하므로서, 레지스터를 많이 쓰는 응용에 적응시키고 있다.

3. 설계와 오류 제거

개발도중에, 시스템 오류 제거(debug)를 얼마나 신속히 할 수 있는가는 오류의 신속하고 쉬운 감지와, 설계를 얼마나 빨리 콤파일하고 재프로그램 할 수 있는가에 달린다. MAX에서는 오류가 자동으로 입력 화면에 표시되며, 자동 라우팅이 100% 보증되고, 또한 5천개의 게이트가 길어도 약 30분내에 콤파일된다. FPGA는 라우팅이 보증되지 못하고, 5천개의 게이트일 경우 보통 10시간 이상 콤파일 시간이 걸려, 전체 설계 시간의 대조를 이룬다. 라우팅이 빠른 것은, 오류를 제거하고 반복 콤파일시켜 제품을 개발할 때, 그만큼 개발시간이 절약되어 상품화 시간을 줄이는 결과를 초래한다.

RAM에 근거하거나 E²C MOS에 근거한 디바이스는 보드내에서 거의 실시간 대화식(real-time interactive)인 오류제거를 허용하여, 설계자가 변화를 시도케 하는 장점이 있다. MAX 7000 계열은 EPROM 제품과 EEPROM 제품을 제공하여, 비소각성이면서도 집적도가 더 높은 EPROM의 장점과, 저렴한 가격과 실시간 대화식 오류 제거의 장점을 가진 EEPROM의 선택을 제공한다.

4. PLD의 중요 제품과 성능

Altera의 MAX계열 EPLD와 Xilinx의 3000과 4000 계열 FPGA를 마크로셀 단위에 따라 그림 6에 보인다. 논리 집적도는 두회사가 비슷한 영역을 포함한다. Actel사에서 Xilinx와 비교하려고 고안한 데이터패스, 타이머/카운터, 스테이트머신, 산술등 4가지 벤치마크 설계에 의해 Altera사가 XC-4005-7과 EPM7256-2를 비교한데 의하면 MAX가 모든 범주에서 성능이 더 빠르며, 평균 2.7배^[7] 빠르게 나와 있다.

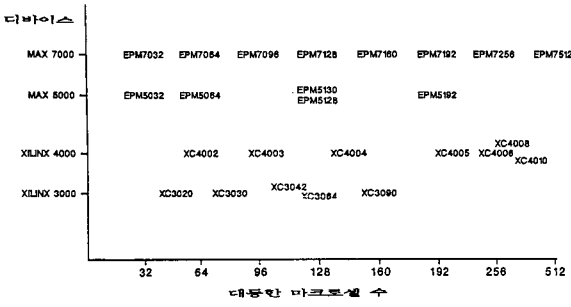


그림 6. Altera EPLD와 Xilinx FPGA제품

EPM7256-2가 ACT1240-1 보다 1.3배^[7] 빠르게 보고 되어 있다. 따라서 EPM7256-2는 현시점에서 생산되는 모든 CPLD중 가장 I/O 핀수가 많고 집적도가 높으며 속도가 빠른 디바이스로 간주된다. 그림 7에서는 Altera의 MAX계열과 FPGA계열의 논리 집적도, 속도, 핀수의 비교^[3,7]를 보인다.

MAX 계열이 가장 광범위한 속도, 핀수 및 집적도를 갖고 있다. 다음 장에서는 MAX 7000 계열에 관해 좀 더 자세히 논한다.

IV. MAX 7000 계열 EPLD

이 장에서는 MAX 7000 계열의 내부구조, 작동원리, 설계방법, 현재 나와있는 제품등을 논한다.

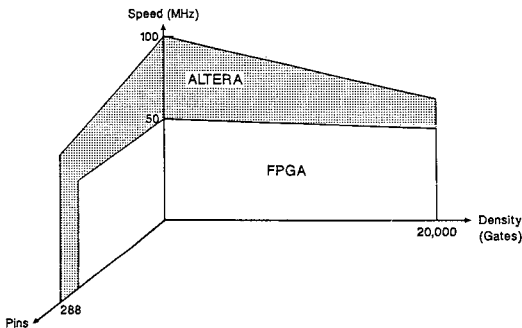


그림 7. Altera PLD와 가장 광범위한 속도, 핀수, 집적도

1. MAX 7000의 블락 도면

MAX 7000의 블락 도면을 그림 8에 제시한다. MAX 5000에서와 마찬가지로 모듈화된 구조를 갖고 있다. 이 구조는 입출력 제어 블락(I/O control block)으로 둘러 쌓여 있어서, LAB 속에 있는 마이크로셀을 I/O 핀에 연결시켜 주는 역할을 하여 많은 핀의 수를 허용한다. 한 디바이스에 LAB이 2 내지 16개(현재) 있으며, 각 LAB에는 16개의 마이크로셀과 연장 프로덕트 텀(expander product term)이 있다. LAB은 PIA로

Anti-fuse인 Actel과의 비슷한 비교에서, Altera의

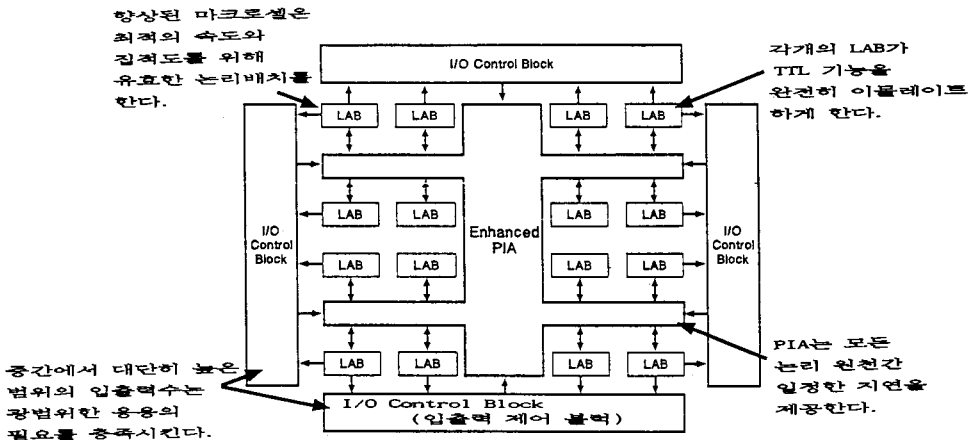


그림 8. MAX 7000 블락 도면

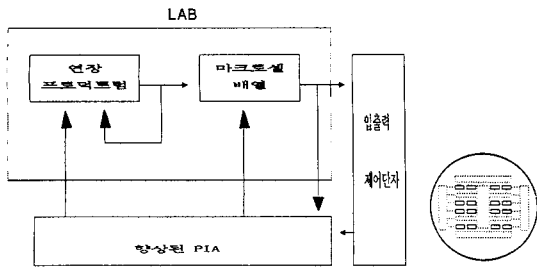


그림 9. MAX 7000의 심장부(logic array block 논리 배열 블록)

연결되어 있다.

2. Logic Array Block

LAB은(logic array block, 논리 배열 블록)은 MAX 구조의 심장부이다.(그림 9 참조)

여기에는 16개의 마크로셀이 있고, 마크로셀에는 PIA와 연장프로덕트텀에서 신호가 입력된다. 입출력 제어 블록은 PIA에 입력을 보내고, 마크로셀로부터의 입력을 출력 핀에 보낸다.

3. MAX 7000 마크로셀

MAX 7000 마크로셀은 5개의 프로덕트텀이 있어서, 선택 매트릭스(select matrix)에 연결된다. (그림 10 참조) 선택 매트릭스는 프로덕트텀을 OR 게이트, XOR 게

트, preset, clear, clock 혹은 clock enable에 보낸다. 따라서, 5개 프로덕트텀의 논리가 낭비되는 적이 없이 필요한 곳으로 배치된다. 마크로셀에서 해결할 수 있는 논리보다 더 복잡한 경우, 연장 프로덕트 텀(expander product term)이라 부르는 추가의 프로덕트텀을 사용할 수 있다. 레지스터는 D, T, JK, 그리고 SR로 프로그램할 수 있다. 레지스터는 preset과 clear를 갖고 있어서 진정한 비동기(asynchronous) 기능을 생성할 수 있다. 이 결과는 모든 TTL 논리를 에뮬레이트 할 수 있게 한다. 레지스터에는 또한 clock enable이 있어서, 동기적 크락킹을 포기하지 않으면서도, 논리에 의해 크락을 제한하도록 한다. 마크로셀의 출력은 조합적(combinatorial)이건 레지스터적이건, PIA에 귀환시키고, 또한 입출력 제어 블록으로 내보낸다.

4. MAX의 향상된 PIA와 라우팅

PIA란 LAB 사이를 프로그램하여 연결하는 통로이다.(그림 11 참조)

PIA는 모든 원천 신호가 모든 디바이스 내의 마크로셀에 접속하는 것을 허용한다. 효율적으로 하기 위해, PIA는 각 LAB가 필요로 하는 신호만을 라우팅(routing) 한다. MAX 7000의 PIA는 MAX 5000의 PIA에 비해 훨씬 향상되었다. PIA를 통해 신호가 건너가는데에 단지 2-3 나노초밖에 지연되지 않는다.

PIA는 MAX EPLD내에서 어느 시작점으로부터 어

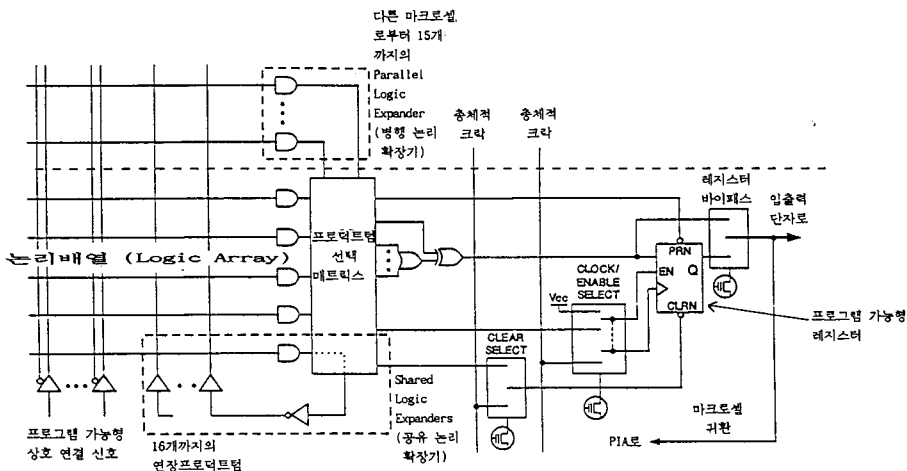
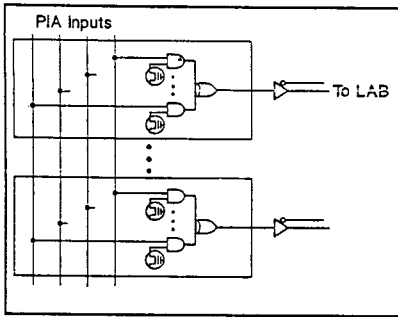


그림 10. MAX 7000 마크로셀



- LAB간의 프로그램 가능한 접속
- 디바이스내의 모든 신호 원천에 연결
- 각 LAB이 필요로 하는 신호란 라우팅
- 어느 두 LAB간이나 일정한 전달 연장
 - 라우팅 전달 연장에 변화가 없음
 - 논리 신호 스큐의 제거

그림 11. PIA(programmable interconnect array, 프로그램 가능형 상호 연결 배열)

느 목적지에는 신호를 연결시킬수 있다는 것을 보장한다. 제 III 장에서 논한 바와 같이 MAX의 PIA는 FPGA 같은 구조에서 사용하는 채널 라우팅(channel routing)의 문제점을 극복하고 있다. FPGA의 채널 라우팅은 두가지 문제점을 갖고 있다. 첫째는 복잡한 기능을 통합하려고 할 때 라우팅 자원(routing resources)이 소진되기가 대단히 쉽다. 이는 마치 좁은 길로 많은 차를 보낼 때, 통로의 부족에서 오는 것과 비유된다. 둘째로는, 팬인(fan in), 팬아웃(fan out) 그리고 신호 경로에 따른 신호전달 속도의 변화때문에, 라우팅이 된다고 해도 회로에서의 지연시간을 예측할 수가 없다. PIA는 이들 제한을 제거한다. PIA는 팬인이나 팬아웃 제한에서 오는 속도의 지연이 없으며, 논리 신호의 스큐(skew)나 신호의 경로에 따른 속도의 변화가 없기 때문에, 예측할 수 있는 균일한 지연을 허락한다. MAX 디바이스는 또한 충분한 라우팅 재원을(마치 큰 하이웨이에 적은 대수의 차가 지나가는 것으로 비유) 갖고 있어서 라우팅은 물론 핀 지정등 프레이징(placing)도 자동으로 대단히 신속히 수행시킨다. 이런 점은 FPGA에서 흔히 필요한 밤새 걸리는 라우팅시간, 자주 일어나는 수동라우팅과 대조적으로, 라우팅시간이 몇분에서 몇십분이면 충분하다.

5. 입력 출력 제어 블록(I/O control block)

MAX 7000의 매크로셀은 입출력 제어 블록과 삼상버퍼(tri-state buffer)로 격리(decoupled)되어 있어서(그림 12 참조) 매크로셀과 입출력핀을 독립해서 쓸 수

있다. 두개의 귀환(feedback) 경로가 있어서, 하나는 매크로셀로부터, 그리고 다른 하나는 삼상버퍼와 입출력핀 사이로부터이다.

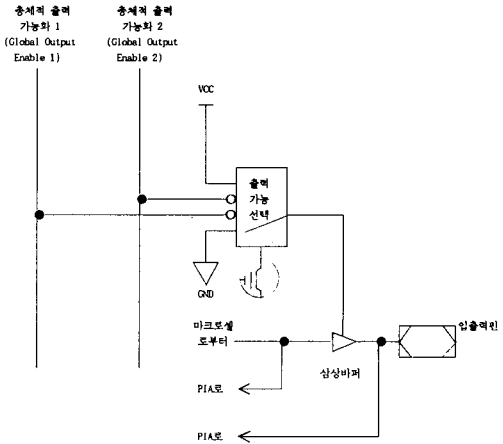


그림 12. 입출력 제어 블록

삼상버퍼를 억제(disable)하므로써 입출력핀을 입력용으로 사용하면서도 매크로셀은 매몰된 논리(buried logic)로 사용할 수가 있다. 삼상버퍼를 작동(enable)시키므로써, 입출력핀을 매크로셀의 출력 단자로 사용할 수가 있다. 혹은 삼상버퍼를 작동 혹은 억제를 선택함에 따라 입출력핀이 양방으로 통신하는 기능을 부여할 수도 있다. 이 입출력 핀은 PIA를 통하여 모든 매크로셀에 연결된다.

6. 프로그램 가능형 속도/전력 제어

MAX 7000 계열의 또하나의 구조적 특성은 속도와 전력의 제어 선택이다. 소프트웨어로 회로를 설계할 때, 각 매크로셀 단위별로 전력 소모를 선택할 수가 있다. 속도를 50% 줄이면, 전력 소모를 75% 감소시킬 수가 있다. 대부분의 회로 설계에서는 단 몇개만의 중요한 경로가 최대 속도를 요하기 때문에, 이러한 매크로셀만 최대 전력을 사용하고, 나머지 매크로셀은 전력을 줄여 전체 MAX 7000의 기능을 최대한으로 사용하면서도 전력의 소모를 흔히 50% 줄일 수가 있다.

7. MAX 7000 EPLD 제품

MAX 7000 계열은 32 매크로셀과 44핀을 가진

EPM 7032에서 1024 마크로셀과 288 핀을 가진 EPM71024가 있다. 현재에 양산되는 부품은 EPM7032, EPM7096, 그리고 EPM7256이며, 각 디바이스는 각종 포장형(package)을 제공한다. 모든 비슷한 포장의 부품은 똑같은 핀아웃(pinouts)을 갖고 있어서, 같은 소켓에 더 높은 밀집도의 MAX 부품을 삽입할 수 있어서, 수직적 통합이 가능하다. 또한, 주어진 밀집도에는 여러 가지 포장의 선택이 가능하여 종적인 통합도 가능하다.

8. MAX+PLUS II 개발 시스템

Altera사가 디바이스에서 계속 새로운 EPLD를 개발해 왔듯이, 개발장비도 계속 발전되어 왔다. 게이트의 밀집도의 증가와 각종 기능의 세련화에 따라 더욱 강력한 소프트웨어 개발장비가 필요하게 되어 MAX+PLUS II가 개발되었고, 이는 MAX 7000 계열뿐만 아니라 크라식, MAX 5000 및 특수기능용 디바이스인 STG EPLD를 다 지원한다. MAX+PLUS II는 스케매틱, 문장(text), 그리고 웨이브폼(waveform)에 의한 설계 입력을 지원한다. 여러개의 칩을 사용할 경우의 논리 콤파일(compilation), 분리(partition), 논리 시뮬레이션, 논리 제작(synthesis), 자동 피팅(fitting)을 지원하여, 한개의 칩에 들어갈 수 없는 큰 설계도 가능하게 되어 있다. 설계의 검증을 위해서는 완전한 논리 기능의 시뮬레이션과 타이밍 시뮬레이션의 기능이 있고, 정적 타이밍 분석 기능과, 논리 기능의 검사(functional test) 기능도 있다. MAX+PLUS II는 386, 486에 근저한 PC에 Window 3.X로 작동되며 motif를 작동시키는 SUN과 Apollo 워크스테이션에서의 콤파일러 버전도 있다.

9. MAX 7000의 요약

MAX 7000 EPLD는 MAX 5000 EPLD에서 한 걸음 더 나간 제 2세대 MAX 디바이스이다. 모듈화된 구조와 향상된 PIA는 시스템내 속도(f_{cm})가 80MHz를 능가케 한다. 논리의 밀집도(density) 또한 대단히 높아서 40,000개의 게이트와 288개의 핀을 제공한다. MAX 7000 계열은 마크로셀 숫자도 여러가지를 제공하고, 프로그램 가능형 속도와 전력소모의 선택을 제공하여, TTL 논리, PAL, PLA, MACH, FPGA 등을 포함하는 광범위의 디바이스를 쉽게 이물레이트하고 통합하게 한다. MAX+PLUS II는 논리 입력에서부터 MAX 디바이스까지 프로그래밍 하게 하는 통합된 소프트웨어 패키지 PC에 쉽게 작동되나, SUN등 워크스테이션에서 제삼 공급자 툴과도 연결하여 작동한다.

V. 맺음말

EPLD는 1984년 처음 300 게이트로 나온 이래, 대단히 빠른 기술과 시장의 발전을 통해 20,000 게이트에 달하는 MAX 7000 계열과 이를 프로그래밍하는 MAX+PLUS II라는 소프트웨어가 나왔다. 따라서, 핀과 논리, 레지스터등의 수에서 PAL, GAL에서 FPGA까지를 다 포함하고 고성능이며 게이트 어레이의 일부에 접하는 프로그래밍 가능형 논리 설계 디바이스로 대두하였다. FPGA는 레지스터가 많이 쓰이고 입력이 적은 분야에 특히 장점이 있어서, 논리 설계 중심의 EPLD와는 서로 보강하는 위치를 차지해 왔다.

PLD 기술은 현재 게이트 어레이의 30%의 설계에 사용할 수 있어, 그 집적도와 성능의 급격한 발전과 시장의 확장이 기대된다. 특히 앞으로의 기술은 PLD를 더 큰 시스템 수준에 통합을 쉽게 하게 하고, 되도록 한 번 한 설계가 여러가지 디바이스를 사용하도록 발전하도록 여러가지 표준이 정해질 것을 사용자는 바라고 있다.

논리 집적도에 주력을 해오던 Altera의 MAX 7000 계열은, 많은 핀수와 집적도 보강으로 레지스터에 주력하는 응용에 박차를 가하고 있으며, 주로 레지스터 중심의 설계에 많이 쓰여오던 FPGA를 생산해 오던 Xilinx사는 Altera EPLD 계열과 비슷한 구조의 EPLD를 개발하던 Plus Logic사를 병합하여, 두 PLD 주류회사가 논리와 레지스터를 다 중요시하는 방향으로 노력하고 있다. 이 PLD 기술이 표준화된 설계를 위한 인터페이스가 필요하며, 이러한 장래의 추세가 PLD의 더욱 집적도가 높은 논리 통합에서 시스템 수준으로 큰 역할을 하여, 더 많은 게이트 어레이의 영역에 잠식하여, 게이트 어레이형 ASIC에서 요구되는 오랜 개발시간, NRE 경비, 주문생산에 의한 재고 비치등의 문제점을 제거하고 상품화 시간(time-to-market)을 줄이는데 공헌할 것이다.


參 考 文 獻

- [1] S. Baker and R. Wilson, "PLD Competition Heats Up", Electronics Times, Issue 676, Jan. 1992.
- [2] L. Farrell, "No One Solution Fits All Problems", Electronic Engineering Times, Technology

Trends/The PLD/FPGA Conundrum, p. T30, March 1992.

- [3] Altera Data Book, Altera Corporation, San Jose, 1991.
- [4] 임만직, “차세대 논리회로의 해결책 EPLD”, 전자과학, 제 30권 제 349호, p.122, 1988년 6월.
- [5] 임만직, “PLD/EPLD와 신기술”, 전자부품, 제 6

권, p.159, 1988년 8월.

- [6] 이경은, “차세대 논리소자로 일컫는 EPLD”, 전자과학, 제 33권 제 387호, p.234, 1991년 8월.
- [7] “Board-Line Programmable Logic Solutions”, Private Communications, p. 38, Altera Corporation, 1992. 

筆者紹介



林 滿 稷

1938年 1月 17日生

1960年 3月 서울대학교 공과대학 화공과 (학사)

1965年 6月 캘리포니아대학(Berkeley)재료공학 (석사)

1967年 6月 캘리포니아대학(Berkeley) Engineering Science (박사)

1967年 4月 ~ 1987年 11月 AT & T Engineering Research Center(Princeton).
책임연구원

1987년 11월 ~ 현재 MJL Corporation(미국) 대표이사

1988년 4월 ~ 현재 한국 MJL(주) 대표이사

주관심 분야 : IC 설계 및 제조, 공장자동화, 데이터 베이스, 실시간 제어, 인공지능, 생산업체 통합관리