

## 디자인 코너

# FPGA 기반 임베디드 시스템 디자인의 플로어플래닝

By Paul Glover

Product Applications Engineer

Xilinx Inc.

프로그래머블 시스템을 위한 플랫폼인 FPGA(field programmable gate arrays)에 임베디드 프로세서 코어의 주입 기술이 또 한 단계 발전함에 따라, 첨단 프로그래머블 로직의 혁신적인 이점들이 드러나고 있다. 전자 장비 제조업체들은 저가, 고성능의 임베디드 프로세서 코어들을 맞춤화 할 수 있는 FPGA 회로와 결합시킴으로써 자신들의 특정 애플리케이션을 위한 주문형 칩들을 보다 적은 비용으로 보다 빠르게 출시할 수 있게 되었다.

이 최첨단 기술을 제대로 사용하기 위해서는 디자이너들이 설계 요건들을 충족시키면서 디자인을 구현할 수 있게 해주는 강력한 설계 툴들이 필요하다. 플로어플래닝은 이러한 설계 요건들을 충족시킬 수 있도록 도와주는 많은 툴들 가운데 하나이다. 플로어플래닝은 디자이너가 임베디드 프로세서와 관련 IP 및 이들의 주문형 로직에 대한 배치 및 그룹화를 통제할 수 있도록 해줌으로써 복잡한 시스템온칩 개발 과정을 단순화해 주고, 전체적인 시스템 성능을 향상시켜 준다.

여기서는 PowerPC 코어를 DDR 메모리 컨트롤러 및 LCD 컨트롤러와 통합시키는 임베디드 시스템의 개발에 있어서 플로어플래너(Floorplanner)를 사용하는 방법을 보여 주고자 한다.

다이어그램 "A"에서 볼 수 있듯이, DDR 메모리 컨트롤러는 고속 PLB(Processor Local Bus)를 통해 PPC에 연결되어 있어 명령어 및 데이터 트랜잭션을 위해 메모리를 고속으로 액세스할 수 있게 해준다.

LCD 컨트롤러의 대역 요건들은 보다 낮아서 상대적으로 속도가 느린 OPB(On-Chip Peripheral Bus)를 통해 PPC에 연결시킬 수 있도록 해준다. OPB는 보다 느린 Peripheral Core들을 PLB2OPB 브리지를 통해 PLB에 연결시켜 준다.

우리의 예제는 두 개의 온칩 메모리 컨트롤러와, Xilinx Virtex-II Pro 디바이스의 인기 있는 특징인 블록 RAM의 사용에 대해서도 보여 준다.

시스템 블록도에서 볼 수 있듯이, 우리의 임베디드 시스템 디자인은 다른 버스 컨트롤러들의 추가도 쉽게 지원할 수 있다.

### 시스템 생성

임베디드 시스템의 설계 및 작성은 시스템 생성기를 사용하여 원하는 프로세서 (와 관련 컨트롤러들)의 패러미터들을 정의하고 임베디드 시스템—일반적으로 프로세서 코어와 버스 구조물들은 물론 기존의 IP도 포함한다—의 소스 디자인

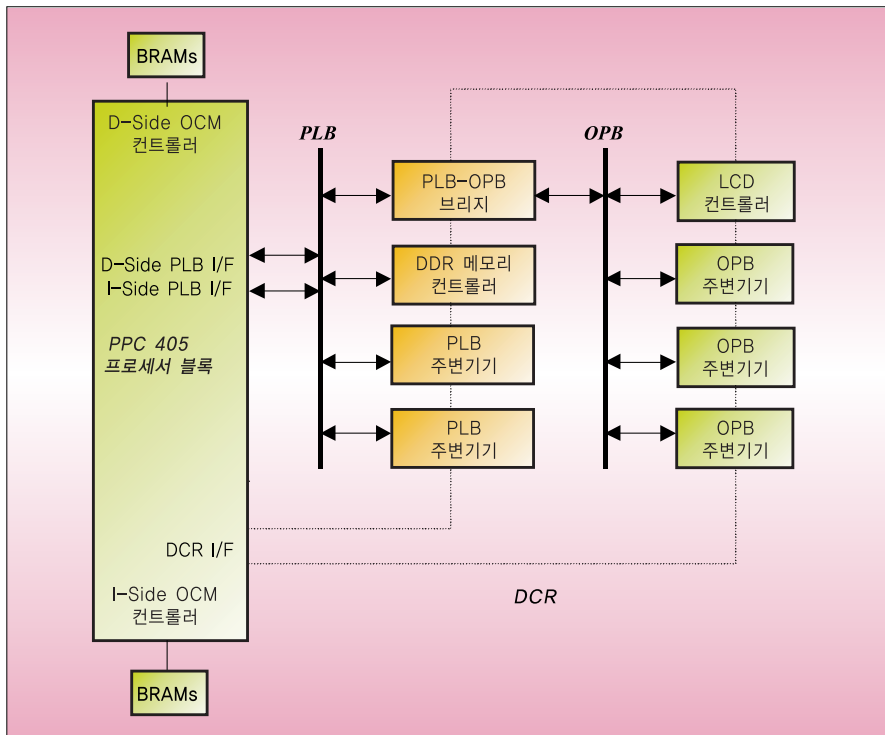


그림 1: DDR 메모리 컨트롤러를 PPC와 연결시킴으로써 메모리를 고속으로 액세스할 수 있다.

을 생성함으로써 간단하게 할 수 있다.

시스템 생성기는 처음 시작할 때 필요한 소프트웨어 헤더 파일들도 생성한다.

이러한 툴들은 하드웨어와 소프트웨어 디자이너들이 동시에 임베디드 시스템의 개발에 착수할 수 있도록 해준다.

하드웨어 디자이너의 경우, 시스템 생성기는 시스템의 모든 주요 부품들을 조립하고 주변기기의 정의, 주변기기의 생성, 그리고 프로세서, 주변기기 및 시스템 버스 상의 수백 개에 달하는 핀들의 연결 작업을 자동화하는 데 필요한 모든 파일들을 생성해 준다.

소프트웨어 개발자들의 경우, 시스템 생성기는 하드웨어와 소프트웨어의 시스템 절충을 수행하는 데 있어서 많은 도움을 주며, 하드웨어와 소프트웨어의 최적화를 원활하게 해준다.

시스템 생성기를 사용할 수 없을 경우에는 시스템 생성을 수작업으로 해야만 하기 때문에 전체적인 시스템 설계 시간과 복잡성이 증가하게 된다.

## 플로어플래닝

많은 EDA 벤더들이 공급하고 있는 또 다른 툴은 Floorplanner이다. Floorplanner는 하드웨어 디자이너가 중요한 모든 기능들과 관련 있는 로직의 배치를 제어하고, 이것이 일단 디바이스 내에 구현되고 난 뒤에는 그 레이아웃을 살펴볼 수 있도록 해준다. 우리의 임베디드 시스템 예에서, Floorplanner는 디자이너가 프로세서와 관련 주변기기들의 배치를 살펴보고 제어하도록 도와줄 수 있다. 어떤 경우에는 이러한 주변기기들의 배치가 디자인의 성능 조건들을 충족시키는 데 있어서 매우 중요한 역할을 할 수도 있다.

앞서 언급했듯이, PowerPC 405 코어는 온칩 메모리(OCM) 컨트롤러를 포함하고 있다. OCM 컨트롤러는 64 비트의 ISBRAM(Instruction-Side Block RAM)과 32 비트의 DSBRAM(Data-Side Block RAM) 양쪽에 대한 인터페이스를 제공한다. OCM 컨트롤러는 최대 16 MB의 DSBRAM과 16 MB의 ISBRAM을 어드레스할 수 있다. OCM의 주된 장점들 가운데 하나는 이것이 고정된 실행시 지연 시간을 보장한다는 데 있다. FPGA의 이러한 타이밍 조건을 충족시키기 위해서 하드웨어 디자이너는 OCM 컨트롤러 인터페이스에 관해 OCM 블록 RAM(BRAM)의 배치를 제어하지 않으면 안 된다. DSBRAM은 PPC 405 코어 위에, ISBRAM은 그 밑에 배치되어야만 한다. 플로어플래너는 이러한 상대적인 레이아웃을 제어할 수 있는 한 가지 방법으로서, 어쩌면 가장 쉬운 방법일지도 모른다.

플로어플래너는 로직을 살펴보고 배치할 수 있는 능력을 제공하지만, 성공적인 플로어플래닝을 위해서는 사용자가 지켜야 할 사항들이 몇 가지 있다. 잘 된 플로어플래닝은 임베디드 시스템의 성능을 향상시킬 수 있지만, 잘못 된 플로어플래닝은 피해를 가져올 수 있다. 다음의 지침 몇 가지를 살펴 보자:

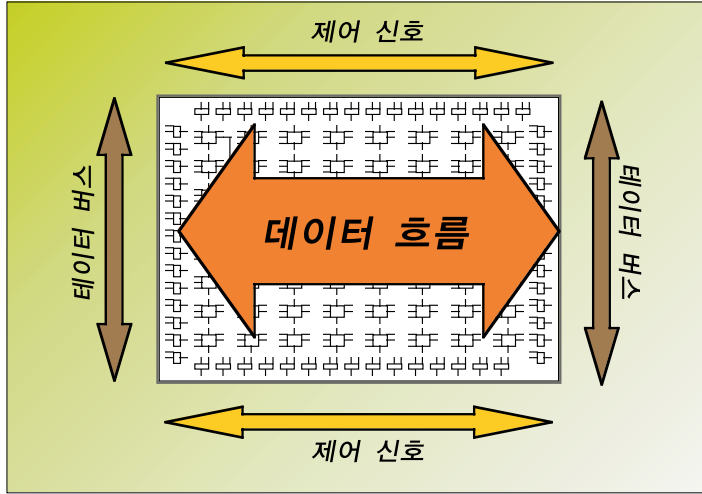


그림 3: 제어 신호들을 위한 I/O는 FPGA의 상단이나 바닥에, 그리고 데이터 버스들을 위한 I/O는 좌측이나 우측에 배치되어야 한다.

**지침 1:** 하드웨어 디자이너로 하여금 자신의 디자인에 대한 플로어플랜을 레이아웃 하기 전에 그 디자인과 목표 아키텍처에 대한 상세한 지식을 갖도록 만들자. 이러한 지식은 올바른 디자인의 배치와 하드웨어 이용을 결정하는 데 있어서 매우 중요하다.

식은 올바른 디자인의 배치와 하드웨어 이용을 결정하는 데 있어서 매우 중요하다.

**지침 2:** FPGA의 물리적 레이아웃은 I/O 레이아웃에 최적이다. 제어 신호들을 위한 I/O는 FPGA의 상단이나 바닥에, 그리고 데이터 버스들을 위한 I/O는 좌측이나 우측에 배치되어야 한다. 이는 FPGA의 배선 자원들을 가장 효과적으로 사용함으로써 최적의 성능을 얻도록 해준다.

**지침 3:** 산술 기능들은 일반적으로 전용의 캐리 체인(carry-chain)들을 이용한다. 캐리 체인은 특정한 수직 방향으로 놓인다. 예를 들어, 10 비트의 카운터는 디바이스의 바닥에서 상부를 향해 수직 방향으로 놓인 캐리 체인을 갖는다. 버스의 최하위 비트를 바닥에 배치하고 최상위 비트는 상단에 배치하면 디자인이 이러한 방향을 이용하게 된다.

예를 들어, 10 비트의 카운터는 디바이스의 바닥에서 상부를 향해 수직 방향으로 놓인 캐리 체인을 갖는다. 버스의 최하위 비트를 바닥에 배치하고 최상위 비트는 상단에 배치하면 디자인이 이러한 방향을 이용하게 된다.

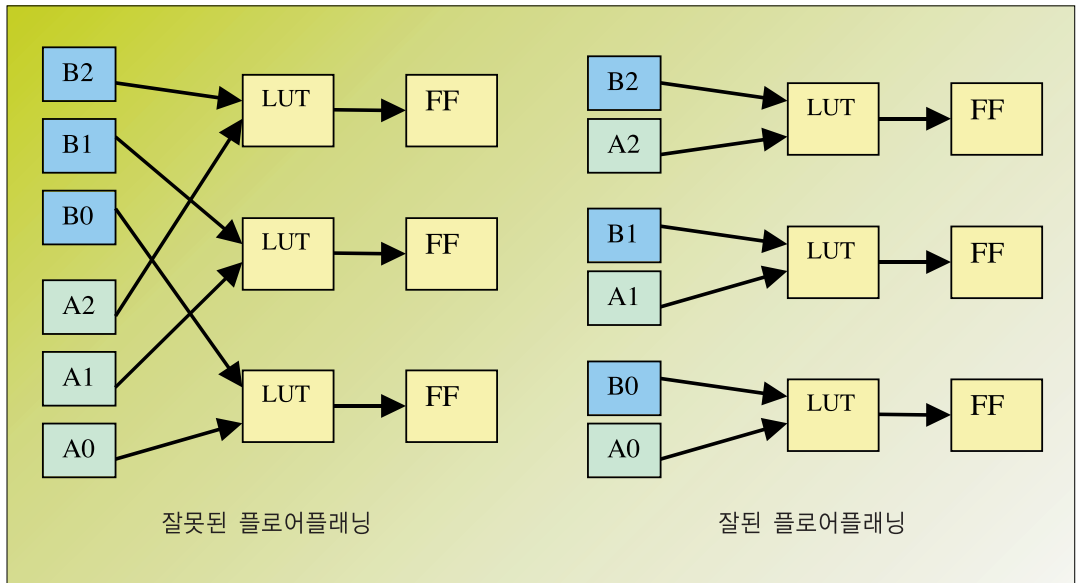


그림 6: 잘 된 플로어플래닝은 임베디드 시스템의 성능을 향상시킬 수 있지만, 잘못 된 플로어플래닝은 피해를 가져올 수 있다.

**지침 4:** 플로어플래너는 계층 구조에 의해 로직을 표시하므로 로직을 공통의 그룹이나 영역으로 한데 묶기가 쉽다. 계층 구조에 의한 그룹화는 로컬 배선 자원들을 이용함으로써 배선 지연을 줄이고 회로 성능을 향상시킨다.

**지침 5:** 관련 I/O 버스들을 인터리브 시킴으로써 하드웨어 엔지니어는 배선 지연을 줄일 수 있다. 이는 버스들이

추가적인 로직에 연결되지 않는 한 가치 있는 기술이다. 레이아웃 상의 결정을 내릴 경우에는 반드시 디자인을 전체로서 바라보도록 하자.

이러한 요령들을 임베디드 시스템 디자인에 적용함으로써 하드웨어 엔지니어는 디자인을 손쉽게 조직 및 레이아웃 할 수 있다. 임베디드 시스템의 플로어플래닝에 착수하기 위해 디자이너는 먼저 서로 다른 그룹들에 로직을 할당하지 않으면 안 된다.

전형적인 그룹은 설계 계층 구조를 토대로 한다. 계층 구조는 임베디드 시스템에서 자연스러운 경계를 제공하며, 플로어플래너가 그룹을 생성하는 데 손쉽게 사용할 수 있다. 시스템 생성기는 자신이 생성하는 시스템의 각 블록을 독자적인 특유의 계층 구조 리퍼런스를 이용하여 생성하므로 임베디드 시스템 디자인의 그룹 정의가 간단해진다.

일단 플로어플래닝 할 그룹들이 생성되고 나면, 디자이너는 그룹들을 배치할 물리적 영역들을 할당할 수 있다. 그룹의 배치는 앞서 설명했던 요령들과 설계 레이아웃을 토대로 이루어져야 한다.

## 레이아웃의 제약 사항들

플로어플래너는 다음과 같이 개별 로직 블록들이나 그룹들을 물리적 위치에 제약 시키기 위한 세 가지 방법을 제공한다:

1) 첫 번째 방법에는 로직 부품을 특정한 물리적 위치나 부품에 배치하는 작업이 수반된다. 예를 들어, 디자이너는 데이터 버스의 출력인 비트 10을 디바이스의 핀 37에 할당할 수 있다. 이런 유형의 할당은 어떠한 융통성도 허용하지 않는다. 배치 및 배선 툴들은 비트 10을 위한 출력 드라이버를 핀 37에 배치하지 않으면 안 된다. 이 방법은 로직 부품들의 그룹에도 적용할 수 있다.

2) 두 번째 방법에는 로직이나 로직 그룹을 물리적 영역에 할당하는 작업이 수반된다. 이 방법이 흔히 사용되는 이유는 배치 및 배선 툴들이 영역 그룹 내에 있는 로직을 이리저리 움직여 봄으로써 최적의 배치 및 배선 결과를 얻을 수 있도록 해주기 때문이다. 이 방법에 대해서는 나중에 설명할 것이다. 시스템 디자인의 최적화를 위해서 보다 강력한 플로어플래너들은 중첩되는 그룹들을 지원하며, 플로어플래닝을 수행한 영역 내에 플로어플래닝을 수행하지 않은 로직을 구현하는 기능도 지원한다.

3) 세 번째 방법에는 RPM(Relationally Placed Macros)의 생성 작업이 수반된다. RPM은 로직 구성 요소들의, 서로에 대해 상대적인 레이아웃을 정의한다. 그러나 이 로직의 정확한 배치는 배치 및 배선 툴들에 의해 결정되므로 디자인의 성능을 최적화할 수 있다. 예를 들어, 룩업 테이블(LUT)은 플립플롭과 동일한 슬라이스에 배치할 수 있다. 슬라이스는 다른 슬라이스 바로 위에 배치할 수 있다. RPM은 디자이너에게 특정 모듈들의 레이아웃을 통제할 수 있도록 해주지만, 배치 및 배선 툴들을 완전히 제한하지는 않는다.

이러한 요령과 방법들을 임베디드 시스템 디자인에 적용함으로써 하드웨어 디자이너들은 특정 부품들의 배치를 제어하고, 전반적인 디바이스 레이아웃(영역 그룹들)을 개선하며, 각 부품들 서로에 대해 특정 부품들의 배치를 제어할 수 있다(RPM).

이것은 여러 개의 UART와 하

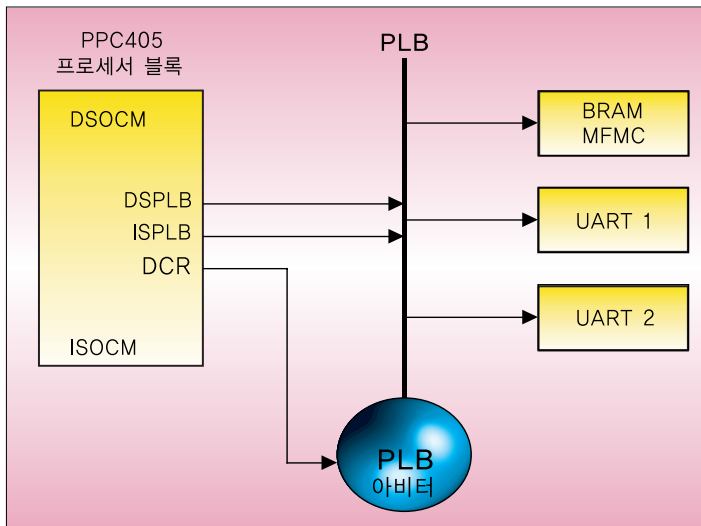


그림 7: BRAM과 PPC 405는 디자인 성능을 극대화하기 위해 특정 위치들로 제한된다.

여기서는 PPC 405를 이용하여 앞서 살펴 보았던 몇몇 나이의 BRAM 컨트롤러로 구성된 임베디드 디자인이다.

이 디자인에서 그룹들은 시스템 생성기에 의해 생성된 계층구조적인 경계 상에 생성된다. PLB Arbitrator, UART 및 BRAM 메모리 컨트롤러에 대해 영역 그룹을 생성할 수 있다. BRAM과 PPC 405는 디자인의 성능을 극대화하기 위해 특정 위치들로 제한된다.

이 디자인의 플로어플래닝을 보여주고자 포함시킨 예제를 참조하기 바란다.

영역 그룹의 할당은 그 영역으로 제한된 로직에게 충분한 자원들을 허용해야만 한다는 점에 주목하자.

보다 강력한 플로어플래너들은 사용자에게 각 영역 그룹에 필요한 자원들의 수를 알려줄 수 있다.

예제의 영역 그룹들은 그 구현에 실제로 필요한 것보다 두 배에 해당하는 자원들을 포함하고 있다.

이는 플로어플래닝 된 영역 내에 로직(그룹 내에 포함되지 않은)을 추가로 배치할 수 있도록 함으로써 디자인을 한층 더 최적화할 수 있게 해준다.

BRAM 메모리 컨트롤러는 다이의 중앙부에 할당되어 있어 모든 BRAM 구성 요소들과 PPC 405 코어를 똑같이 액세스할 수 있도록 해준다.

이 디자인 내의 Block RAM은 PPC 405를 위한 데이터와 명령어들을 포함하고 있다. UART1과 UART2는 I/O 가 사이에 위치해 있어 입력에서 클럭, 그리고 클럭에서 출력까지의 시간을 최소화 시킬 수 있도록 해준다.

PLB 마스터는 CPU Instruction Side PLB 인터페이스와 CPU Data Side PLB 인터페이스의 두 개가 있다.

이 때문에 PLB Arbitrator가 PPC 405 코어의 가까이 배치되어 있다. 그 구현 결과는 다음과 같다.

각 영역 그룹에 할당되는 로직은 그 영역의 제약 사항에 따라서 배치된다.

또한 디자인의 성능을 높이기 위해 추가적인 로직이 혼합된다. 이는 각 영역 그룹의 할당시 여분의 공간을 남겨

뒤야 하는 또 다른 이유를 보여준다.

## 플로어플래닝의 가치

저가의 고성능 프로세서를 프로그래머블 로직 디바이스에 임베디드 시킴으로써, 칩 개발자는 총 시스템 성능을 향상시킬 뿐만 아니라 프로그래머블 로직이 갖는 전통적인 적시 출시의 이점도 누리게 된다.

앞서 개관한 요령과 절차들을 따른다면, 플로어플래닝은 자원의 활용, 디자인 플래닝, 그리고 FPGA 내의 임베디드 시스템 구현에 부가 가치를 부여해 줄 것이다. □