



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2007-0092093
(43) 공개일자 2007년09월12일

(51) Int. Cl.

G06F 9/46 (2006.01) G06F 9/38 (2006.01)
G06F 9/28 (2006.01)

(21) 출원번호 10-2006-0119678

(22) 출원일자 2006년11월30일

심사청구일자 없음

(30) 우선권주장

095107574 2006년03월07일 대만(TW)

(71) 출원인

파덕 테크놀로지 코., 엘티디.

대만, 신-쥬, 동-다 로드 섹션 2, 넘버 1, 10층-2

(72) 발명자

중-린 창

대만, 신-쥬, 동-다 로드 섹션 2, 넘버 1, 10층-2

찬-포 링

대만, 신-쥬, 동-다 로드 섹션 2, 넘버 1, 10층-2

쉬-유 린

대만, 신-쥬, 동-다 로드 섹션 2, 넘버 1, 10층-2

(74) 대리인

김경희

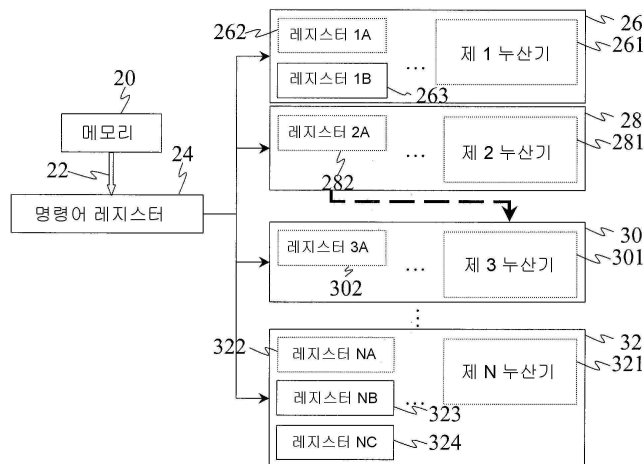
전체 청구항 수 : 총 15 항

(54) 다중 마이크로컨트롤러 시스템, 명령어, 및 그의 명령어실행 방법

(57) 요약

다중 MCU 코어 로직들을 포함하는 다중 마이크로컨트롤러 시스템에 있어서, 다중-MCU-코어-로직 선택 오퍼랜드가 본 발명에 따른 명령어에 제공된다. 다중-MCU-코어-로직 선택 오퍼랜드는 상기 시스템내의 대응하는 MCU 코어 로직을 특정하거나 선택하고, 또 필요한 경우, MCU 코어 로직의 서브-유닛을 특정하거나 선택한다. 상기 시스템내의 어떤 MCU 코어 로직은 이 명령어를 폐칭 및 디코딩하기 위해 주 연산 마이크로컨트롤러 유닛으로서 사용될 수 있다. 상기 명령어내의 연산 코드에 대응하는 연산 동작이 다중-MCU-코어-로직 선택 오퍼랜드에 대응하는 선택된 MCU 코어 로직에 대해 수행된다. 그러므로 상기 다중 마이크로컨트롤러 시스템은 MCU 코어 로직, 및 필요하다면, 원하는 동작이 수행되는 그것의 서브-유닛을 직접 특정할 수 있다. 더욱이, 다중 마이크로컨트롤러 시스템내의 상이한 MCU들은 리소스들을 공유할 수 있고 서로를 제어할 수 있다. 따라서 진정한 의미의 병렬 처리가 달성된다.

대표도 - 도2a



특허청구의 범위

청구항 1

복수의 MCU 코어 로직들을 포함하는 다중 마이크로컨트롤러 시스템에 사용하기 위한 명령어에 있어서,

상기 명령어의 연산 동작(operation action)을 정의하는 연산 코드(operation code); 및

상기 연산 코드에 의해 정의된 상기 연산 동작이 실행되는 상기 MCU 코어 로직들 중 하나를 선택하는 적어도 하나의 다중-MCU-코어-로직 선택 오퍼랜드(multiple-MCU-core-logic selection operand);를 포함하는, 다중 마이크로컨트롤러 시스템에 사용하기 위한 명령어.

청구항 2

제 1 항에 있어서, 상기 명령어는 N개의 연속 또는 불연속 비트들의 2진 코드의 형태로 표현되는 MCU 코어 로직 ID를 포함하고, 여기서 N은 자연수인, 다중 마이크로컨트롤러 시스템에 사용하기 위한 명령어.

청구항 3

제 2 항에 있어서, 상기 MCU 코어 로직 ID는,

상기 오퍼랜드에 포함되는 방식;

상기 연산 코드에 포함되는 방식;

상기 오퍼랜드에 부분적으로 포함되는 방식;

상기 연산 코드에 부분적으로 포함되는 방식;으로 구성되는 방식들 중 하나로 구성되는, 다중 마이크로컨트롤러 시스템에 사용하기 위한 명령어.

청구항 4

제 1 항에 있어서, 상기 명령어는 상기 MCU 코어 로직 ID를 저장하는 위치로부터 상기 MCU 코어 로직 ID를 검색하는, 다중 마이크로컨트롤러 시스템에 사용하기 위한 명령어.

청구항 5

제 1 항에 있어서, 상기 다중-MCU-코어-로직 선택 오퍼랜드는 MCU 코어 로직의 ID 및 상기 MCU 코어 로직과 연결된 서브-유닛을 나타내는 서브-오퍼랜드(sub-operand)를 포함하는, 다중 마이크로컨트롤러 시스템에 사용하기 위한 명령어.

청구항 6

제 1 항에 있어서, 상기 선택된 MCU 코어 로직은 상기 명령어를 통해 임의의 다른 MCU 코어 로직 또는 임의의 다른 주변 유닛과 통신할 수 있는, 다중 마이크로컨트롤러 시스템에 사용하기 위한 명령어.

청구항 7

다중 마이크로컨트롤러 시스템의 명령어 실행 방법에 있어서,

상기 방법은:

연산 코드 및 적어도 하나의 다중-MCU-코어-로직 선택 오퍼랜드를 구비하는 명령어를 주 연산 MCU 코어 로직에 의해 페칭하는 단계; 및

상기 명령어를 디코딩하는 단계;를 포함하고,

상기 주 연산 MCU 코어 로직은 상기 다중-MCU-코어-로직 선택 오퍼랜드에 의해 선택된 적어도 하나의 MCU 코어 로직에 대한 상기 연산 코드에 의해 정의된 연산 동작을 수행하는, 다중 마이크로컨트롤러 시스템의 명령어 실행 방법.

청구항 8

제 7 항에 있어서, 상기 다중 마이크로컨트롤러 시스템은 복수의 MCU 코어 로직들을 포함하고, 상기 MCU 코어 로직들 중 어느 하나는 상기 주 연산 MCU로서의 역할을 할 수 있는, 다중 마이크로컨트롤러 시스템의 명령어 실행 방법.

청구항 9

제 8 항에 있어서, 상기 다중-MCU-코어-로직 선택 오퍼랜드는 2^N 개 이하의 MCU 코어 로직들 중 대응하는 MCU 코어 로직을 선택하는 적어도 N개의 연속 또는 불연속 연속 또는 불연속 비트들을 포함하고, 여기서 N은 자연수인, 다중 마이크로컨트롤러 시스템의 명령어 실행 방법.

청구항 10

제 9 항에 있어서, 상기 적어도 N개의 비트들은,

상기 오퍼랜드에 포함되는 방식;

상기 연산 코드에 포함되는 방식;

상기 오퍼랜드에 부분적으로 포함되는 방식;

상기 연산 코드에 부분적으로 포함되는 방식;으로 구성되는 방식들 중 하나로 구성되는, 다중 마이크로컨트롤러 시스템의 명령어 실행 방법.

청구항 11

제 7 항에 있어서, 상기 다중-MCU-코어-로직 선택 오퍼랜드는, MCU 코어 로직을 식별하는 것 외에 또한 상기 MCU 코어 로직과 연관된 서브-유닛(sub-unit)을 식별하는, 다중 마이크로컨트롤러 시스템의 명령어 실행 방법.

청구항 12

제 11 항에 있어서, 상기 식별된 MCU 코어 로직은 상기 명령어를 통해 임의의 다른 MCU 코어 로직 또는 임의의 다른 주변 유닛과 통신할 수 있는, 다중 마이크로컨트롤러 시스템의 명령어 실행 방법.

청구항 13

복수의 MCU 코어 로직들을 포함하는 다중 마이크로컨트롤러 시스템에 있어서,

각각의 상기 MCU 코어 로직은 2진 코드의 형태로 표현되는 대응하는 MCU 코어 로직 ID를 가지며,

상기 시스템은 상기 MCU 코어 로직들 중 하나와 관련된 명령어를 실행할 때, 상기 명령어는 상기 하나의 MCU 코어 로직에 대응하는 상기 MCU 코어 로직 ID를 포함하는, 다중 마이크로컨트롤러 시스템.

청구항 14

제 13 항에 있어서, 상기 명령어는 연산 코드를 나타내는 비트들 및 오퍼랜드를 나타내는 비트들을 포함하고,

상기 MCU 코어 로직 ID는:

상기 오퍼랜드를 나타내는 상기 비트들에 포함되는 방식;

상기 연산 코드를 나타내는 상기 비트들에 포함되는 방식;

상기 오퍼랜드를 나타내는 상기 비트들에 부분적으로 포함되는 방식;

상기 연산 코드를 나타내는 상기 비트들에 부분적으로 포함되는 방식;으로 구성된 방식들 중 하나로 구성되는, 다중 마이크로컨트롤러 시스템.

청구항 15

복수의 MCU 코어 로직들을 포함하는 다중 마이크로컨트롤러 시스템에 있어서,

각각의 상기 MCU 코어 로직은 2진 코드의 형태로 표현되는 대응하는 MCU 코어 로직 ID를 가지며,

상기 시스템은 상기 MCU 코어 로직들 중 하나와 관련된 명령어를 실행할 때, 상기 명령어는 상기 MCU 코어 로직

ID를 저장하는 위치로부터 상기 하나의 MCU 코어 로직에 대응하는 상기 MCU 코어 로직 ID를 검색하는, 다중 마이크로컨트롤러 시스템.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

- <26> 발명의 분야
- <27> 본 발명은 마이크로컨트롤러 시스템의 명령어 및 그의 실행 방법에 관한 것으로서, 특히 다중 마이크로컨트롤러 시스템의 명령어 및 그의 실행 방법에 관한 것이다.
- <28> 관련 기술의 설명
- <29> 과학 및 기술의 진보에 따라, 컴퓨터가 전보다 더 강력해 지고 있다. 컴퓨터는 컴퓨터의 심장과 유사한 CPU를 구비하며, 이것은 사용자에게 의해 할당된 태스크(task)를 끝내기 위해서 뿐만 아니라 사용의 문제를 해결하기 위해, 컴퓨터의 아이디어 및 논리와 유사한 컴퓨터 프로그램을 통해 주변 하드웨어 장치를 제어한다.
- <30> 컴퓨터가 일련의 처리 절차를 실행하는 컴퓨터 프로그램은 명령어들(instructions) 및 명령문들(statements)의 조합이다. 기계 코드(machine code)는 명령어의 가장 기본적인 형태이며, 이것은 연산 코드와 오퍼랜드(또는 오퍼랜드들)로 구성된다. 기계 코드의 오퍼랜드(들)은 기계 코드내의 연산 코드에 따라 동작을 수행한다. 프로그램의 기록을 단순화하기 위해, 명령어들 및 명령문들은 명령 코드(mnemonic)의 형태로 기록된다. 특히, 오퍼랜드는 어드레스 또는 순수 데이터일 수 있고, 연산 코드는 오퍼랜드들의 동작, 예를 들어 데이터 전달, 산술 연산, 논리 연산, 비트 연산, 또는 프로그램 브랜칭(program branching)을 정의한다. 식 (1)은 연산 코드 및 2개의 오퍼랜드를 구비하는 종래의 명령어를 나타낸다:
- <31> MOV A, REGA (1)
- <32> 상기에서, MOV는 데이터 전달을 나타내는 연산 코드이고, A 및 REGA는 제어기에서의 누산기 및 레지스터를 각각 나타내는 오퍼랜드들이다. 따라서, 이 명령어의 의미는 레지스터 A내의 데이터를 제어기내의 누산기에 저장한다는 것이다.
- <33> 컴퓨터가 명령어를 실행하기 전에, 일련의 디코딩 및 분석 태스크가 오퍼랜드(들) 및 연산 코드로 표현된 데이터 및 동작을 인터럽트한 후 이들을 실행시키기 위해 적절한 유닛을 할당해야 한다. 각각 종래의 제어 시스템의 블록도 및 종래의 제어 시스템에 따른 명령어 실행 방법의 플로차트인 도 1(a) 및 도 1(b)를 참조하여, 식 (1)에서의 명령어가 이하에 설명된다. 도 1(a) 및 도 1(b)에 나타난 것과 같이, 제어기(16)는 먼저 데이터 버스(12)를 통해 메모리(10)내의 데이터를 판독하고 그것을 명령어 레지스터(14)에 임시 저장한다(단계 S01). 이후 제어기(16)는 식 (1)에서의 명령어의 의미가 레지스터 A(162)내의 데이터를 제어기(16)의 누산기(161)에 저장하는 것이라고 결정하기 위해 식(1)에서의 명령어를 디코딩한다(단계 S02). 다음에, 상기 해석(interpretation)에 따라, 제어기(16)는 레지스터 A(162)내의 데이터를 검색하고 그 데이터를 누산기(161)에 저장하여(단계 S03), 식 (1)에서의 명령어 실행을 완료한다.
- <34> 상기 제어 시스템 및 그 명령어 실행 방법으로부터 용이하게 알 수 있는 것과 같이, 이들 모든 명령어들의 포맷은 시스템이 제어기(또는 메인 제어기)만을 갖고, 명령어들의 페치, 디코딩 및 실행은 이 유일한 제어기에 의해서 수행된다는 전제하에서 설계된다. 프로그램들을 실행하는 하나 이상의 마이크로컨트롤러 유닛(MCU) 코어 로직들을 가진 다중 마이크로컨트롤러 시스템에 있어서(각각의 MCU 코어 로직은 대응하는 레지스터들 또는 다른 연관된 회로 유닛들을 가질 수 있음), 그것은 병렬 및 멀티-태스크(multi-task) 처리를 달성하기 위해 상기 명령어들을 사용하는 것이 극히 곤란하고, 불가능하지 않으면 복잡해진다. 다중 마이크로컨트롤러 시스템에 있어서, 특히 시스템이 복잡한 프로그램을 처리할 때, 모든 MCU들이 프로그램 실행 상태를 판독할 수 있고, 서로를 제어할 수 있도록 내부 리소스들을 공유하는 것이 매우 중요하다. 상기 능력들에 의해, 다중 마이크로컨트롤러 시스템은 진정한 병렬 처리 및 실시간 응답을 달성할 수 있고, 상기 시스템 비용이 감소될 수 있다.

발명이 이루고자 하는 기술적 과제

<35> 그러므로, 본 발명은 종래 기술의 상기 문제들을 해결할 수 있는 포맷을 갖는 명령어들을 채용하는 다중 마이크로컨트롤러 시스템을 제공하고, 또 상기 명령어들을 실행하는 방법을 제공하고, 그럼으로써, 병렬 및 멀티-태스크 처리를 달성하여 다중 마이크로컨트롤러 시스템의 처리 성능을 효과적으로 향상시킨다.

발명의 구성 및 작용

<36> 발명의 요약

<37> 본 발명의 주 목적은 다중 마이크로컨트롤러 시스템에 사용하기 위한 명령어를 제공하는 것이다. 상기 명령어는 연산 코드 및 적어도 하나의 다중-MCU-코어-로직 선택 오퍼랜드(MCU-core-logic selection operand)를 포함한다. 2^N 이하의 MCU 코어 로직들이 있을 경우, 적어도 N 비트들이 이들 MCU 코어 로직들로부터 하나를 선택하는데 요구되고, 여기서 이들 N 비트들은 연속 또는 불연속일 수 있다. 상기 연산 코드는 명령어의 연산 동작을 정의한다. 상기 다중-MCU-코어-로직 선택 오퍼랜드는 다중 마이크로컨트롤러 시스템에서 특정 MCU 코어 로직을 특정하고(또는 선택하고), 다중-MCU-코어-로직 선택 오퍼랜드는 서브-위치(sub-location)를 특정하기 위해 서브-오퍼랜드를 더 포함할 수 있다. 이와 같은 명령어에 의해, 다중 마이크로컨트롤러 시스템에서 선택될 MCU 코어 로직(있다면, 서브-오퍼랜드에 의해 정의된 그것의 서브-위치)은 용이하고 정확하게 식별될 수 있고, 따라서 어떠한 복잡한 프로그램 코드들을 더 이상 기록할 필요가 없다.

<38> 본 발명의 다른 목적은 상기 명령어에 따라 연산들을 수행할 수 있는 다중 마이크로컨트롤러 시스템을 제공하는 것이다.

<39> 본 발명의 또 다른 목적은, 메인 MCU가 명령어를 페치 및 디코딩하고, 상기 명령어내의 다중-MCU-코어-로직 선택 오퍼랜드에 의해, 선택된 MCU 코어 로직이 메인 MCU와 협력하여 연산 코드에 의해 정의된 동작을 수행할 수 있고, 게다가 선택된 MCU 코어 로직내의 데이터가 판독되어 다른 선택된 MCU 코어 로직의 지정된 위치에 직접 기록될 수 있는 다중 마이크로컨트롤러 시스템의 명령어 실행 방법을 제공하는 것이다. 이와 같은 명령어 및 방법을 이용하여, 다중 마이크로컨트롤러 시스템내의 상이한 MCU 코어 로직들은 리소스들을 공유하고 서로를 제어하고, 그럼으로써 진정한 병렬 및 멀티-태스크 처리를 달성하여 성능을 향상시킬 수 있다.

<40> 본 발명의 또 다른 목적은 명령어 실행 방법에 따라 명령어를 실행할 수 있는 다중 마이크로컨트롤러 시스템을 제공하는 것이다.

<41> 상기 목적을 달성하기 위해, 본 발명은 다중 마이크로컨트롤러 시스템에 사용하기 위한 명령어 및 그것의 실행 방법을 제공한다. 본 발명의 다중 마이크로컨트롤러 시스템에 사용하기 위한 명령어는 연산 코드 및 적어도 하나의 다중-MCU-코어-로직 선택 오퍼랜드를 포함한다. 다중-MCU-코어-로직 선택 오퍼랜드는 다중 마이크로컨트롤러 시스템에서 특정 MCU 코어 로직을 특정하거나 선택하고, 다중-MCU-코어-로직 선택 오퍼랜드는 서브-위치를 특정하기 위해 서브-오퍼랜드를 더 포함할 수 있다. 상기 연산 코드에 대응하는 연산 동작이 선택된 MCU 코어 로직에 대해 수행된다. 더욱이, 본 발명은 또한 다중 마이크로컨트롤러 시스템에서 이 명령어의 실행 방법을 제공한다. 다중 마이크로컨트롤러 시스템에서 주 연산 MCU를 미리 결정한 후(주 연산 MCU는 임의의 MCU 또는 전용의 MCU일 수 있음), 명령어가 페치된 후 주 연산 MCU에 의해 디코딩되어 명령어에서 연산 코드의 정보 및 적어도 다중-MCU-코어-로직 선택 오퍼랜드를 획득한다. 연산 코드에 의해 정의된 연산 동작은 다중-MCU-코어-로직 선택 오퍼랜드에 의해 특정된 적어도 하나의 MCU에서 수행된다.

<42> 본 발명에 따르면, 명령어들이 단순한 포맷으로 기록될 수 있어 다중 마이크로컨트롤러 시스템내의 상이한 MCU 코어 로직들은 리소스들을 공유할 수 있고 서로를 제어할 수 있다. 다중 마이크로컨트롤러 시스템내의 임의의 또는 전용의 MCU 코어 로직은 명령어들을 페치 및 디코딩하기 위한 주 연산 MCU가 되도록 필요에 따라 선택될 수 있다. 상기 명령어들에 있어서, 상기 다중-MCU-코어-로직 선택 오퍼랜드는 MCU 코어 로직을 연산 동작의 대상(object)으로 선택하고(다중-MCU-코어-로직 선택 오퍼랜드내의 서브-오퍼랜드는, 있다면, 그것의 서브-위치를 선택한다), 상기 연산 코드는 그 동작을 정의하고, 상기 다중 마이크로컨트롤러 시스템은 그에 상응하게 동작한다.

<43> 본 발명의 여러 목적 및 이점은 첨부 도면과 관련하여 읽을 경우 다음의 상세한 설명으로부터 더 용이하게 이해될 수 있을 것이다.

<44> 최선의 실시예들의 상세한 설명

<45> 본 발명에 따른 다중 마이크로컨트롤러 시스템은 다중 MCU 코어 로직들간의 상호 액세스 및 제어가 크게 향상된

처리 능력을 제공한다. 종래의 마이크로컨트롤러 시스템과 비교되는 본 발명의 하나의 개선점은, 이 시스템에 사용하기 위한 명령어들이 발명성이 있는 포맷을 채용하여 임의의 MCU 코어 로직이 다중 마이크로컨트롤러 시스템에서 임의의 다른 MCU 코어 로직들내의 데이터를 액세스할 수 있다는 점이다. 이하, 그 상세가 기술된다.

<46> 본 발명에 따른 다중 마이크로컨트롤러 시스템에 있어서, MCU 코어 로직에 대해 수행될 태스크를 정의하는 명령어는 연산 코드 및 적어도 하나의 다중-MCU-코어-로직 선택 오퍼랜드로 구성된다. 상기 명령어는 다음의 조합들, 즉 연산 코드와 다중-MCU-코어-로직 선택 오퍼랜드, 연산 코드와 2개의 다중-MCU-코어-로직 선택 오퍼랜드, 연산 코드, 다중-MCU-코어-로직 선택 오퍼랜드, 및 다른 공통 오퍼랜드 ("공통(common)"은 오퍼랜드가 다중-MCU-코어-로직 선택 오퍼랜드가 아님을 의미) 등의 어느 하나를 포함할 수 있다. 다중-MCU-코어-로직 선택 오퍼랜드의 내용(content)은, 연산 코드에 대응하는 연산 동작이 독립적으로 행해지는, 다중 MCU 코어 로직들 중에서 하나의 선택된 MCU 코어 로직을 나타내는 정보(및 필요하다면, 그의 서브-위치)를 특정하기 위해 사용될 수 있다. 또는, 다중-MCU-코어-로직 선택 오퍼랜드의 내용은 연산 코드에 대응하는 연산 동작이 협력적으로 행해지는 2개의 선택된 MCU 코어 로직들을 나타내는 정보(및 필요하다면 이들의 서브-위치들)를 특정하기 위해 사용될 수 있다.

<47> 본 발명에 따른 제 1 예는 다음 식 (2)에 나타난 것과 같이, 하나의 연산 코드 및 2개의 다중-MCU-코어-로직 선택 오퍼랜드들을 포함한다:

<48> MOV M3A, M2REG2A (2)

<49> 상기에서, MOV는 "데이터 전달(data transfer)"을 나타내는 연산 코드이고, M3A 및 M2REG2A는 2개의 상이한 MCU 코어 로직들 및 이들의 서브-위치를 각각 나타내는 오퍼랜드들이다. 오퍼랜드 M3A는 제 3 MCU 코어 로직 M3 및 그 내부 서브-위치 A를 나타내는 2개의 섹터, 즉 M3 및 A를 구비한다. 서브-위치 A는 제 3 누산기에 대응한다. 따라서 다중 MCU 코어 로직들간에서 선택하기 위한 이 오퍼랜드 M3A는 제 2 MCU 코어 로직의 제 3 누산기에 대응한다. 오퍼랜드 M2REG2A는 제 3 MCU 코어 로직 M2 및 그 내부 서브-위치 REG2A를 나타내는 2개의 섹터, 즉 M2 및 REG2A를 구비한다. 서브-위치 REG2A는 레지스터 2A에 대응한다. 따라서 다중 MCU 코어 로직들 중에서 선택하기 위한 이 오퍼랜드 M2REG2A는 제 2 MCU 코어 로직의 레지스터 2A에 대응한다. 그러므로, 이 명령어에 의해 정의된 동작은 "데이터를 제 2 MCU 코어 로직의 레지스터 2A로부터 제 3 MCU 코어 로직의 제 3 누산기로 전달하는 것"으로서 해석될 수 있다.

<50> 위에 기술한 명령어는 본 발명의 단지 하나의 예이다. 식 (2)에서의 데이터 전달의 연산 동작 외에, 동일한 아이디어가 산술 연산, 논리 연산, 비트 연산 및 프로그램 브랜칭과 같은 다른 연산 동작에 적용가능하다. MCU 코어 로직들에서 또는 MCU 코어 로직들에 의해 실행될 임의의 연산 동작은 본 발명의 명령어로 표현될 수 있다.

<51> 하나의 다중-MCU-코어-로직 선택 오퍼랜드와 하나의 공통 오퍼랜드가 있는 경우에, 공통 오퍼랜드는 소스 어드레스, 목적지 어드레스, 또는 심지어 순수 데이터일 수 있다. 예를 들면, 다음의 식 (3)은 연산 코드 및 2개의 오퍼랜드를 구비하는 명령어를 나타낸다:

<52> MOV M1REG1B, #55H (3)

<53> 상기에서, MOV는 또한 "데이터 전달"을 나타내는 연산 코드이고 M1REG1B 및 #55H는 각각 다중-MCU-코어-로직 선택 오퍼랜드 및 공통 오퍼랜드이다. 식 (2)의 상기 명령어와 유사하게, 다중 MCU 코어 로직들 중에서 선택하기 위한 오퍼랜드 M1REG1B는 제 1 MCU 코어 로직의 레지스터 1B에 대응한다. 오퍼랜드 #55H에 대해, 그것은 상수 "55H"를 나타내는 순수 데이터(pure number data)이다. 그러므로, 이 명령어에 의해 정의된 동작은 "55H를 제 1 MCU 코어 로직의 레지스터 1B로 전달하는 것"으로서 해석될 수 있다.

<54> 식 (2)의 명령어는 연산 코드와 다중 MCU 코어 로직들 중에서 선택하기 위한 2개의 오퍼랜드의 조합이고, 식 (3)의 명령어는 연산 코드, 다중-MCU-코어-로직 선택 오퍼랜드, 및 다른 공통 오퍼랜드의 조합이다. 이들 2개의 형식 이외에, 물론 다음 식 (4)에 나타난 것과 같이, 연산 코드 및 단지 하나의 다중-MCU-코어-로직 선택 오퍼랜드를 포함하는 명령어가 있을 수 있다:

<55> POP M3PC (4)

<56> 상기에서, POP는 "특정 위치로의 스택 메모리(stack memory) 내용의 전달"을 나타내는 연산 코드이고, M3PC는 단지 제 3 MCU 코어 로직의 프로그램 카운터를 나타내는 오퍼랜드(다중-MCU-코어-로직 선택 오퍼랜드)이다. 그러므로, 이 명령어에 의해 정의된 동작은 "제 3 MCU 코어 로직의 프로그램 카운터로의 스택 메모리 내용의 전송"로서 해석될 수 있다.

- <57> 상기 식 (2) 내지 (4)에 있어서, 각각의 다중-MCU-코어-로직 선택 오퍼랜드는 MCU 코어 로직 (M1, M2, M3 등)을 나타내는 섹터 및 서브-위치(A, REG2A, REG1B, PC 등)를 나타내는 서브-오퍼랜드를 구비한다. 그러나, 명령어가 그것의 서브-위치를 정의하지 않고 단지 MCU 코어 로직을 정의하는 것도 가능하다. 이 경우에, 다중-MCU-코어-로직 선택 오퍼랜드는 단지 MCU 코어 로직을 나타내는 섹터를 구비하지만 서브-위치를 나타내는 서브-오퍼랜드는 구비하지 않는다. 예를 들면, 명령어 세트내의 명령어 PUSHPC가 스택 메모리로의 프로그램 카운터의 내용을 푸싱(pushing)하는 것으로 정의되면, 다음 식 (5)는 서브-위치를 나타내는 서브-오퍼랜드는 필요로 하지 않는다:
- <58> PUSHPC M4 (5)
- <59> 이 명령어는 스택 메모리로의 제 4 MCU 코어 로직의 프로그램 카운터(program counter)의 내용을 푸싱하는 것을 의미한다.
- <60> 상기 설명에 있어서, 상기 명령어들은 더 높은 수준의 언어(higher level language)로 표현된다. 실제로, 낮은 레벨의 2진 코드(lower-level binary codes)로 변환되면, 상기 식 (2) 내지 (5)는 일련의 2진수(binary digits)로 표현될 수 있고, 여기서 몇몇 자리들은 연산 코드를 나타내고, 다른 자리는 오퍼랜드들을 나타낸다. 예를 들어, 상기 명령어들은 5자리의 연산 코드를 나타내는 최상위 비트(MSB)로부터 시작하고, 다른 비트들은 오퍼랜드들을 나타내는, 16비트 2진 코드로 각각 표현될 수 있다.
- <61> 2진 코딩에 있어서, 각 비트는 2개의 MCU 코어 로직들 중에서 선택하기 위해 사용될 수 있다. 그러므로, 2^N 개 이하의 MCU 코어 로직들이 있을 경우, 적어도 N 비트가 이들 MCU 코어 로직들 사이에서 선택하기 위해 필요로 된다(개념상, 이들 N개의 2진 비트로 구성된 2진수는 대응하는 MCU 코어 로직의 ID(identification)로서 간주될 수 있다). 본 발명에 따르면, 이들 N 비트는 연속(즉, 다수의 N 연속 자리를 형성), 또는 불연속(즉, 명령어를 나타내는 2진 코드로 산재됨(scattered))이다. 대부분의 경우에, 이들 N 비트는 오퍼랜드에 포함되지만, 이들은 연산 코드에, 또는 오퍼랜드에 부분적으로 그리고 연산 코드에 부분적으로 포함된다.
- <62> 더 구체적인 예로서, 명령어가 16자리 2진 코드로 표현되고 명령어 세트로 정의된 24 연산 코드가 있다면, 그것은 연산 코드를 식별하기 위해 5비트를 필요하는 데 그 이유는 $2^4=16<24<32=2^5$ 이기 때문이다. 그러므로, 명령어의 2진 코드는 MSB로부터 시작하는 5개의 비트들이 연산 코드를 식별하도록 사용되고, 다른 비트들이 오퍼랜드를 식별하도록 사용되는 방식으로 구성된다. 더욱이, 본 발명에 따라, 다중 마이크로컨트롤러 시스템이 8개의 MCU 코어 로직들(따라서 3-비트 ID가 MCU 코어 로직을 식별하기 위해 필요로 된다)을 가지는 것으로 가정하면, MCU 코어 로직의 3-비트 ID는 제 6 비트 내지 제 8 비트(연속)로 또는 제 14 비트 내지 제 16 비트(연속)로, 또는 제 6 비트, 제 10 비트 및 제 14 비트(불연속)로 구성될 수 있다. 게다가, 단지 24개의 연산 코드가 있기 때문에, 00000 내지 10111는 제 1 내지 제 23 연산 코드를 표현하기 위해 사용될 수 있고, 11xxx는 제 24 연산 코드를 표현하기 위해 사용될 수 있다. 이러한 구성에 이어, 11000 내지 11111가 제 1 내지 제 8 MCU 코어 로직들의 제 24 연산을 각각 표현하기 위해 사용될 수 있다. 이 경우에, MCU 코어 로직을 식별하기 위한 3-비트 ID는 제 1 내지 제 23 명령어들로 오퍼랜드에 포함되고, MCU 코어 로직을 식별하기 위한 3-비트 ID는 제 24 명령어로 연산 코드에 포함된다.
- <63> 상기 제 24 명령어에 있어서, MCU 코어 로직 ID가 연산 코드에 포함되지만, 개념상, 우리는 또한 상기 연산 코드를 MSB로부터 시작하는 2비트로서 그리고 다중-MCU-코어-로직 선택 오퍼랜드를 제 3 비트로부터 시작하는 비트로서 간주할 수 있다.
- <64> 이하, 본 발명의 최선의 실시예에 따른 명령어 실행 방법을 식 (2)에서의 명령어 및 도 2(a) 및 도 2(b)를 참조하여 설명한다. 도 2(a)는 본 발명의 일 실시예에 따른 제어 시스템의 블록도이다. 도 2(b)는 본 발명의 일 실시예에 따른 명령어 실행 방법의 플로차트이다. 도 2(a)에 나타난 바와 같이, 내부 레지스터들(262, 263, 282, 302, ... 및 322, 323, 324)을 각각 가진 N개의 MCU 코어 로직들(26, 28, 30, ... 32)로 구성된 다중 마이크로컨트롤러 시스템에 있어서, 제 1 MCU 코어 로직(26)은 주 연산 MCU로서 사용되고, 메모리(20)내의 데이터가 데이터 버스(22)를 통해 판독된 다음, 명령어 레지스터(24)에 임시 저장된다(단계 S11). 제 1 MCU 코어 로직(26)은 이어, 식 (2)의 명령어를 디코딩하고 명령어의 내용이 "제 3 코어 로직(30)의 제 3 누산기(301)로의 제 2 MCU 코어 로직(28)의 레지스터 2A(282)내의 데이터의 저장"을 나타낸다고 결정한다(단계 S12). 다음에, 상기 해석에 따라, 제 1 MCU 코어 로직(26)은 제 2 MCU 코어 로직(28)에 명령하여 레지스터 2A(282)내의 데이터를 판독하게 하고, 그 데이터를 제 3 MCU 코어 로직(30)의 제 3 누산기(301)에 기록한다(도 2(a)에 굵은 점선으로 나타난 데이터 전달 방향, 그것에 의해 식 (2)에서의 명령어의 실행을 완료)(단계 S13).

- <65>** 본 발명에 따라, 제 1 MCU 코어 로직이 상기 실시예에서 주 연산 MCU로서 기술되지만, 임의의 MCU 코어 로직이, 시스템의 최적 효율을 달성하기 위해, 상이한 응용들이 요구되면, 임의로(randomly) 또는 미리 정해진 규칙에 의해, 주 연산 MCU의 역할을 하도록 선택될 수 있다.
- <66>** 더욱이, MCU 코어 로직을 식별하는 ID 번호가 상기 실시예에서는 명령어에 직접 지정되지만, 동일한 목적이 간접의, 숨겨진 방식(hidden manner)에 의해 달성될 수 있다. 예를 들면, 선택된 MCU 코어 로직의 ID는 먼저 레지스터에 기록되고, 이 레지스터의 내용은 이후 다음 명령어에 의해 호출된다. 이 예에서, MCU 코어 로직을 식별하는 ID 번호가 나중의 명령어에 나타나지 않지만, 그것은 실질적으로 이 발명에 의해 개시된 것과 등가물이고 따라서 본 발명의 범위에 포함되어야 한다.
- <67>** 전술한 바로부터 알 수 있는 것과 같이, 본 발명은 다중 마이크로컨트롤러 시스템, 다중 마이크로컨트롤러 시스템에 사용하기 위한 명령어 및 그의 실행 방법을 제공하고, 그 결과, 상기 시스템내의 각각의 MCU는 용이하게 편집가능한 명령어에 의해 다른 MCU 또는 임의의 다른 내부/주변 유닛들과 직접 통신 및 협력할 수 있고, 그것에 의해, 다중 마이크로컨트롤러 시스템이 최적의 처리 효율을 달성할 수 있게 한다.
- <68>** 이상 본 발명은 그의 최선의 실시예에 관하여 설명되었지만, 본 발명은 그들의 상세들에 한정되지 않는다는 것이 이해될 것이다. 다양한 대체예들 및 변형예들이 다음의 설명에 제시되고, 이 기술분야에서 숙련된 사람은 다른 예를 생각할 수 있을 것이다. 그러므로, 모든 이와 같은 대체예들 및 변형예들은 다음의 청구항들에 정의된 본 발명의 범위내에 포함되도록 의도된다.

발명의 효과

- <69>** 본 발명에 의하면, 다중 마이크로컨트롤러 시스템내의 상이한 MCU 코어 로직들이 리소스들을 공유하고 서로를 제어하므로, 진정한 병렬 및 멀티-태스크 처리를 달성하여 성능을 향상시킬 수 있다.

도면의 간단한 설명

- <1> 도 1(a)는 종래의 제어 시스템의 블록도이다.
- <2> 도 1(b)는 종래의 제어 시스템의 명령어 실행 방법의 플로차트이다.
- <3> 도 2(a)는 본 발명의 일 실시예에 따른 제어 시스템의 블록도이다.
- <4> 도 2(b)는 본 발명의 일 실시예에 따른 명령어 실행 방법의 플로차트이다.

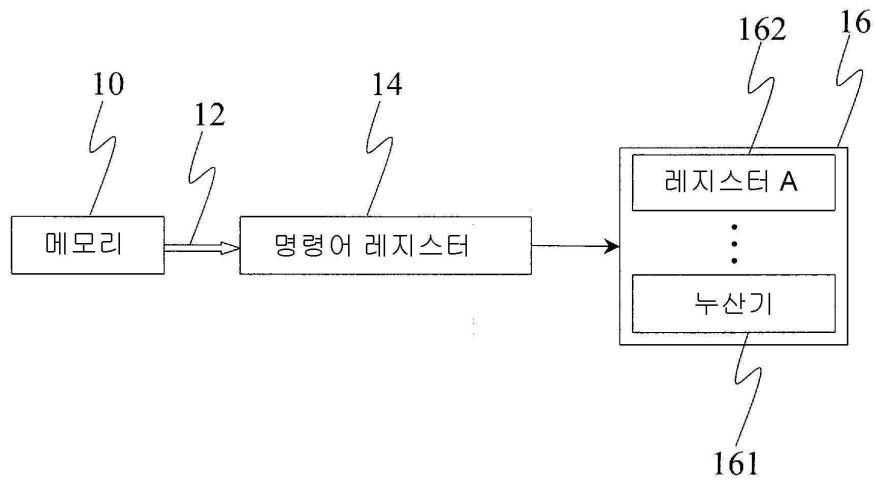
<5> <도면의 주요 부분에 대한 부호의 설명>

- | | | |
|------|-------------------|------------|
| <6> | 10: 메모리 | 12: 데이터 버스 |
| <7> | 14: 명령어 레지스터 | 16: 제어기 |
| <8> | 161: 누산기 | 162: 레지스터 |
| <9> | 20: 메모리 | 22: 데이터 버스 |
| <10> | 24: 명령어 레지스터 | |
| <11> | 26: 제 1 MCU 코어 로직 | |
| <12> | 262: 레지스터 1A | |
| <13> | 263: 레지스터 1B | |
| <14> | 261: 제 1 누산기 | |
| <15> | 28: 제 2 MCU 코어 로직 | |
| <16> | 282: 레지스터 2A | |
| <17> | 281: 제 2 누산기 | |
| <18> | 30: 제 3 MCU 코어 로직 | |
| <19> | 302: 레지스터 3A | |

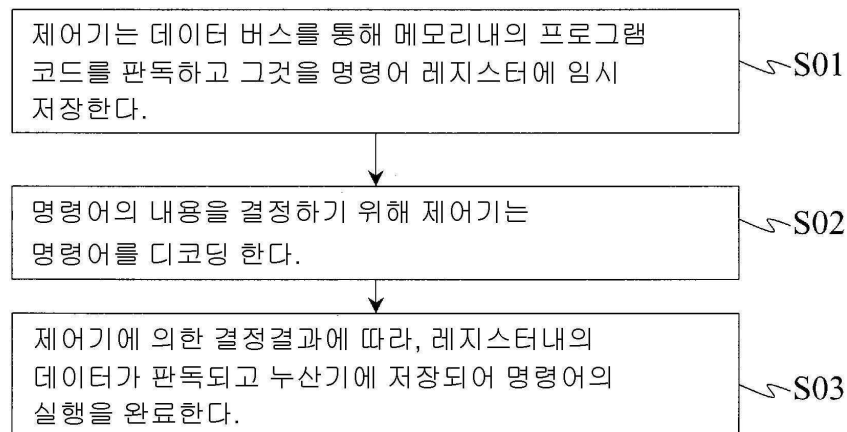
- <20> 301: 제 3 누산기
- <21> 32: 제 N MCU 코어 로직
- <22> 322: 레지스터 NA
- <23> 323: 레지스터 NB
- <24> 324: 레지스터 NC
- <25> 321: 제 N 누산기

도면

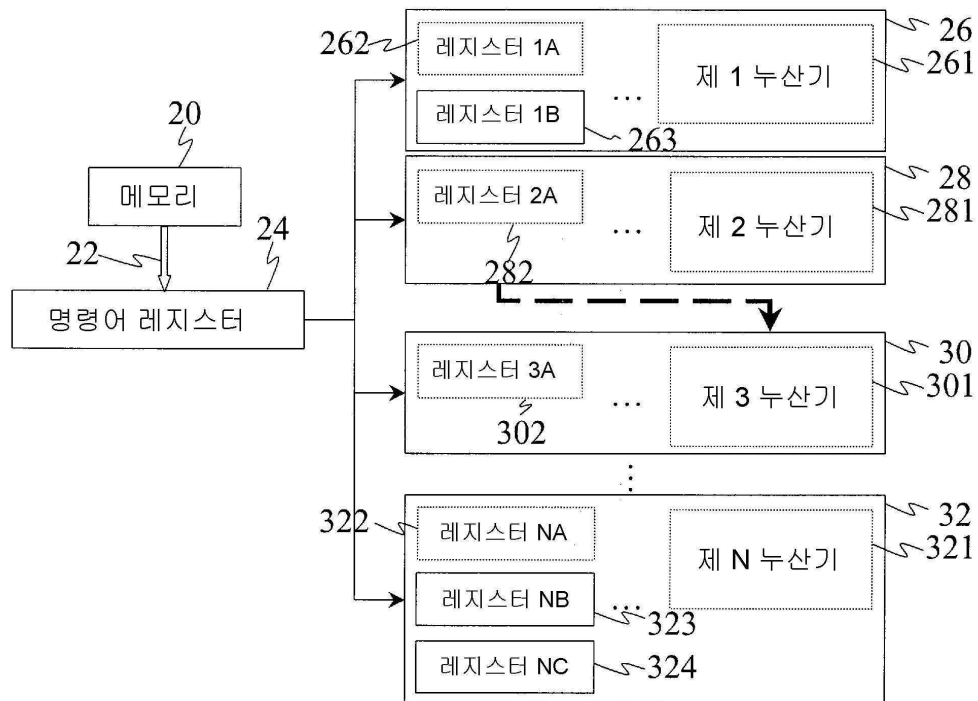
도면1a



도면1b



도면2a



도면2b

