(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl.[6]
G06F 13/00

(45) 공고일자 2004년10월06일
(11) 등록번호 10-0440655
(24) 등록일자 2004년07월07일

(21) 출원번호 10-1997-0018087
(22) 출원일자 1997년05월06일

(65) 공개번호 10-1997-0076247
(43) 공개일자 1997년12월12일

(30) 우선권주장 643,647 1996년05월06일 미국(US)

(73) 특허권자 모토로라 인코포레이티드
미국, 일리노이 60196, 샤움버그, 이스트 앨공퀸 로드 1303

(72) 발명자 랜건 죤 에이
미국, 텍사스 78737, 오스틴, 그래나다 힐즈 드라이브9126

드레이크 필립 비.
미국, 텍사스 78749, 오스틴, 폰챠 패스 6707

레이서 레베카 에이
미국, 텍사스 78704, 오스틴, 빅토리 드라이브 #D202

(74) 대리인 이병호

심사관 : 임영희

(54) 데이타프로세서동작방법및집적회로

요약

'특수' 모드에서 감지 제어 비트들(sensitive control bits: 34,36,38,40)에 대한 제1 기록을 무시하는 레지스터 액세스 스킴(scheme)을 갖는 데이타 처리 시스템(18). 사용자 모드에서 감지 제어 비트들(34,36,38,40)은 시스템 열화를 피하기 위해 한 번의 기록만을 허용한다. 에뮬레이션(emulation) 및 테스트는 감지 제어 비트들들(34,36,38,40)에 대한 제1 기록을 무시하지만, 후속되는 모든 기록들을 허용하는 '특수' 모드를 이용한다. 어플리케이션 코드는 에뮬레이터 모드에서 작동함으로써, 사용자가 유연성있고 정확한 설계를 할 수 있도록 한다.

대표도

도 1

명세서

도면의 간단한 설명

도1은 본 발명의 일실시예에 따른 데이타 프로세서의 블록도.
도2는 본 발명의 일실시예에 따른 제어 레지스터를 레지스터 형태로 도시하는 도면.
도3은 본 발명의 일실시예에 따른 버스 인터페이스 장치의 블록도.
도4는 본 발명의 일실시예에 따른 특수 기록 제어 회로의 회로도.

도5는 본 발명의 일실시예에 따른 에뮬레이터의 블록도.
도6은 본 발명의 일실시예에 따른 에뮬레이터의 블록도.
도면의 주요부분에 대한 부호의 설명
2 : 데이타 프로세서 4 : CPU
6 : 메모리 장치 8 : 버스
10 : 버스 인터페이스 장치 12 : 제어 레지스터
34 : NECLK 36 : LSTRE
38 : IVIS 40 : 포트 제어
80 : 모드 논리 81 : 특수 모드
82 : 정상 모드

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

(관련 출원)
본 출원은 Jay A. Hartvigsen 등이 발명하고, 1995년 6월 2일 출원되고, 그의 양수인에게 양도된, 변리사 정리번호 SC-02421A의 'Method and Apparatus for Controlling Show Cycles in a Date Processing System' 및 Jay A. Hartvigsen 등이 발명하고, 1995년 6월 2일 출원되고, 그의 양수인에게 양도된, 변리사 정리번호 SC-02422A의 'Method and Apparatus for Providing an External Indication of Internal Cycles in a Date Processing System'의 두 가지 미국 특허와 관련된다.
(발명의 분야)
본 발명은 일반적으로 데이타 처리 시스템에 관한 것으로서, 특히 데이타 프로세서 내의 레지스터를 액세스하는 방법에 관한 것이다.
(발명의 배경)
보호된 제어 레지스터들
마이크로프로세서 시스템에서, 임의의 '제어(control)' 레지스터들 및 비트들은 시스템 구성(system configuration)을 결정하기 위해 사용된다. 많은 데이타 프로세서들은 특수 상황들일 때 외에는 기록들에 대해 보호되는 감지(sensitive) 제어 레지스터들과 비트들을 갖는다. 전형적인 보호 메커니즘들은 리셋 후 특정 기간 내에만 이들 비트들을 기록하는 능력과 각 리셋 후 한 번 그것들을 기록하는 능 력을 포함한다. 이들 비트들은 데이타 프로세서의 기본 구성을 제어하고, 우연한 기록이 심각한 시스템 문제들을 유발 할 수 있었다. 일단 셋팅되면, 그 부분이 리셋될 때까지 제어 비트들이 변경되지 않고 유지된다. 보호들은 소프트웨어 제어된 특징들을 포함하는 것을 실재적으로 만들 수 있고, 이들은 그렇지 않으면 배제될 수 있는 것이다. 데이타 프로세서가 복잡할수록 가능한 제어 레지스터들 및 비트들의 수는 증가한다. 이것은 특정 기간동안 제어 레지스터에 모든 기록들이 행해지도록 요구하는 레지스터 액세스 스킴들(schemes)에 문제를 가져온다.
작동 모드들
시스템 일체성을 유지하기 위해서, '정상' 또는 '사용자' 작동 모드에서, 감지 제어 비트들이 통상 한 번만 기록될 수 있도록 허용된다. 보호 메커니즘들은 '특수' 작동 모드들에서 오버라이드(override)될 수 있으며, 여기서 비트들은 레셋 시퀀스를 통하지 않고 반복적으로 변경될 수 있다. 오버라이드 특징은 특히 테스트동안 유용하며, 많은 가능한 구성들은 최소량의 시간 내에 달성되어야 한다. 에뮬레이터 작동 모드 동안, 데이타 프로세서는 소정의 구성을 유지하고, 감지 제어 레지스터들에 대한 기록들을 금지하는 것이 바람직하다. 예를 들어, 핀 구성은 포트 교체 장치를 허용하도록 유지되어야 한다. 다양한 모드들의 요구에 적합하도록 많은 스킴들이 개발되어왔다.

발명이 이루고자 하는 기술적 과제

에뮬레이터 모드들은 데이타 프로세서의 실제 동작 내로 사용자 가시성을 허용하도록 설계되어있다. 설계 복잡성이 증가함에 따라, 에뮬레이션은 응용개발에 필수적이 된다. 실용상, 에뮬레이션 셋업을 수용하기 위해서는 추가의 코드가 기록되어야 한다. 이러한 추가의 코드는 복잡성의 정도를 더하고, 에뮬레이션 프로세스에 불확실성을 도입한다. 사용자 코드에 의해 에뮬레이션이 손상(corrupt)되지 않는 일정한 제어 스킴을 정의하는 것이 바람직하다.

발명의 구성 및 작용

일반: 다음 설명에서, 여러 가지 특정의 세부사항들이 본 발명의 완전한 이해를 제공하기 위한 특정 제어 레지스터 비트 길이들과 같은 것이 기재된다. 그러나, 그러한 특정의 세부사항들 없이 본 발명이 실행될 수 있다는 것이 당업계의

숙련자들에게는 자명할 것이다. 다른 예들에서, 불필요한 세부사항으로 본 발명을 불명료하게 하지 않기 위해서 블록도 형태로 회로를 나타내었다. 대부분의 경우에 타이밍 고려들 등에 관한 세부사항들 설명 등은 본 발명의 완전한 이해를 얻기 위해서 불필요하기 때문에 그에 대한 설명들은 생략하였고, 관련 기술의 당업자의 기술에 속한다.
용어 정의: 용어 '버스'는 데이타, 어드레스들, 제어, 또는 상태와 같은 하나 이상의 다양한 정보를 전달하기 위해 사용될 수 있는 다수의 신호들 또는 컨덕터들(conductors)을 가리키는데 사용될 것이다. 용어 '어서트(assert)' 또는 '부정(negate)'은 각각 신호, 상태 비트, 또는 유사한 장치를 논리적으로 '참' 또는 논리적으로 '거짓' 상태를 만드는 것을 가리킬 때 사용될 것이다. 논리적으로 참 상태가 로직 레벨 1이면, 논리적으로 거짓 상태는 논리 레벨 0일 것이다. 논리적으로 참 상태가 논리 레벨 0이면, 논리적으로 거짓 상태는 로직 레벨 1일 것이 다.
본 발명: 본 발명은 제어 선택 비트들이 유효하지 않게 한 번 기록될 수 있고, 그 다음에 유효하게 기록될 수 있는 데이타 프로세서 내의 레지스터를 액세스하는 방법을 제공한다. 본 발명의 일 실시예에서, 사용자 모드는 제어 선택 비트들이 한 번 기록될 수 있도록 한다. 테스트 및 에뮬레이션을 위해서 사용된 '특수' 모드에서를 제외하고, 후속 기록들이 무효이며, 초기 기록은 유효하지 않지만, 모든 후속 기록들은 유효하다.
시스템 구성 및 제어에 관해서는, 예시적인 실시예는 레지스터 비트에 기록을 이용한다. 대안의 실시예들은 레지스터 비트들을 렌더링(rendering)(즉, 어서트 또는 부인)하기 위해 임의의 여러 가지 방법들을 사용하지만, 논리 연산들 및 소프트웨어 명령들에 한정되지 않는다. 각각의 경우에, '특수' 모드로 들어갈 때, 제어비트들은 초기 렌더링에서는 유효하지 않지만, 후속 렌더링들에는 유효이다.
본 발명은 시스템 구성의 확실성과 안정성을 보장하면서 어플리케이션 코드가 수정 없이 에뮬레이터 모드에서 동작할 수 있도록 한다. 본 발명은 데이타 프로세서 에뮬레이션에 유연성을 더한다. 또한, 본 발명은 제어 레지스터에 다중 기록들을 검출하는 코드 검증방법을 제공한다.
이제 도면들을 참고해서 본 발명이 설명되는데, 도면에서, 도시된 구성요소들은 크기에 맞춰 나타내지 않았으며, 유사하거나 동일한 구성요소들은 여러 개의 도면을 통해서 동일한 번호로 표시하였다.

도면들의 설명

도1은 본 발명의 일 실시예를 설명하는 것으로서, 중앙 처리장치(CPU)(4), 버스(8), 버스 인터페이스 장치(10), 제어 레지스터(12), 및 메모리 장치(6)은 모두 데이타 프로세서(2)의 구성요소이다. CPU(4), 메모리 장치(6), 및 버스 인터페이스 장치(10)는 각각 다중의 2-방향 컨덕터들에 의해서 버스(8)에 결합되어있다. 버스 인터페이스 장치(10)는 다중의 2-방향 컨덕터들을 통해서 다른 모듈과 주변장치들과 통신한다. 버스 인터페이스 장치(10)는 데이타 프로세서(2)의 시스템 구성을 실행하기 위해 사용되는 제어 레지스터(12)를 포함한다.
제어 레지스터(12)의 상세한 설명은 도2에 도시되어있다. 본 발명의 일 실시예에서, 제어 레지스터(12)는 보류된 상위 4 비트를 갖는 8 비트 길이이다. 하위 4 비트들은 'No E-Clock'(NECLK)(34), '로우 스트로브 인에이블(Low Strobe Enable)'(LSTRE)(36), 내부 가시성(Internal Visibility: IVIS)(38), 포트 제어(40)로 표시되어있다. 각 비트의 중요성은 작동 모드에 의존한다.
도3을 참고하면, 버스 인터페이스 장치(10)의 일례를 도3에 설명하였다. 본 발명의 실시예에서, 버스(8), 버스 인터페이스 장치(10), 및 모드 논리(80)는 데이타 프로세서(2) 내에 포함되어있다. 버스 인터페이스 장치(10)는 제어 레지스터(12), 특수 기록 제어 회로(75), 및 레지스터 복호 회로(50)를 포함한다.
데이타 정보와 통신하기 위해서, 버스(8)는 데이타(9)의 2-방향 컨덕터들을 통해서 제어 레지스터(12)를 따라 버스 인터페이스 장치(10)에 결합되어있고, 여기서 제어 레지스터(12)는 수 비트의 정보를 포함한다. 본 발명의 일 실시예에서, 비트0(70)은 특수 기록 제어 회로(Special Write Control Circuity:75)로 이용가능하 지만, 비트1(but Bit1: 71), 비트2(Bit2: 72)는 제어 레지스터(12) 내의 다수의 다른 비트들의 예시이며, 특수 기록 제어 회로(75)에 의해서 제어된다.
본 발명의 일 실시예에서, 신호 모드 논리(80)는 '정상' 또는 '특수' 모드로서 동작 모드를 나타낸다. '특수' 모드에서, 특수 기록 제어 회로(75)는 리셋 후, 제어 레지스터(12) 내 비트들의 초기 기록을 방지하고, 이들은 특수 기록 제어 회로(75)에 의해서 제어되지만, 모든 후속 기록들을 허용한다. 특수 기록 제어 회로(75)에 의해 제어된 비트들은 감지 제어 비트들로서 언급된다.
어드레스 정보를 통신하기 위해서, 버스(8)는 어드레스(11)의 컨덕터들을 통해서 레지스터 복호 회로(Register Decode Circuitry:50)에 결합되어있다. 레지스터 복호 회로(50)는 또한, 신호 모듈 선택(56), 기록(52), 및 판독(54)을 수신한다. 본 발명의 일 실시예에서, 이들 신호들은 각 컨덕터들을 경유해서 수신되지만, 대안의 실시예는 임의의 수의 컨덕터들 또는 다중 컨덕터들을 사용할 수 있다. 또한, 제어 복호 회로(50)는 기록 또는 판독으로서의 동작을 나타내는 신호들을 수신한다. 신호 기록(52)은 동작이 기록일 때를 나타낸다. 어드레스(11)는 레지스터 복호 회로(50)에 의해서 수신된다. 본 발명의 실시예에서, 어드레스(11) 및 모듈 선택(56)은 제어 레지스터(12)를 나타낼 때, 신호 판독(54) 및 기록(52)에 의해서 제공된 판독 및 기록 정보들은 각각 레지스터 판독(58) 및 레지스터 기록(60)으로 건네진다. 판독(54) 및 기록(52) 중의 하나만이 버스 사이클 동안 액티브이므로, 어드레스(11)가 제어 레지스터(12)를 나타낼 때, 레지스터 판독(58) 및 레지스터 기록(60) 중의 하나만이 유효하다.
또한, 레지스터 복호 회로(50)는 다음번 동작, 기록 또는 판독을 나타내는 신호들을 수신한다. 레지스터 기록(60)은 제어 레지스터(12)에 대한 데이타 기록을 선택하고, 제어 레지스터(12) 내의 다른 비트(들)뿐만 아니라 감지 제어 비트(들)(즉, 특수 제어 회로(75)에 의해서 제어된 비트(들)) 양자에 대한 기록에 영향을 미치도록 사용된다. 마찬가지로, 신호 레지스터 판독(58)은 제어 레지스터(12)의 데이터 판독을 선택하고, 제어 레지스터(12)의 감지 제어 비트(들) 및 다른 비트(들)로 판독을 실행할 수 있다.
도3을 더 참고하면, 레지스터 복호 회로(50)는 신호 레지스터 판독(58)을 통해서 제어 레지스터(12)의 데이타 판독을 개시하고, 신호 레지스터 기록(60)을 통해서 데이타 기록을 개시한다. 레지스터 기록(60)은 또한, 특수 기록 제어 회

로(75)에 제공되며, 이는 리셋 후 초기 기록을 억제하고, 특수 기록(76)을 통해서 제어 레지스터(12) 내의 감지 제어 비트(들)로 후속 기록을 허용한다. 본 발명의 실시예에서, 특수 기록(76)은 특수 기록 제어 회로(75)를 제어 레지스터(12)에 결합하는 단일 컨덕터로 보내진다. 대안의 실시예들이 단일 컨덕터 또는 임의 다수의 컨덕터들을 이용할 수 있다.

본 발명의 실시예에서, 특수 기록 제어 회로(75)는 레지스터 복호 회로(50)로부터 컨덕터를 통해서 신호 레지스터 기록(60)을 수신한다. 신호 리셋(84)은 리셋된 데이타 프로세서(2)를 나타내는 컨덕터를 경유해서 특수 기록 제어 회로(75)에 공급되며, 모드 로직(80)은 데이타 프로세서(2)의 작동 모드를 결정하고, 특수 모드(81) 및 정상 모드(82) 신호를 컨덕터들을 통해서 특수 기록 제어 회로(75)에 공급한다. 본 발명의 일 실시예는 각 신호를 위한 각 컨덕터를 갖는다. 대안의 실시예들은 컨덕터 및 다중의 컨덕터들의 조합 또는 다중의 컨덕터들을 사용하기도 한다. 일 실시예에서, 특수 제어 기록 회로(75)는 컨덕터를 통해서 리셋(84) 신호를 수신한다.

신호 특수 모드(81)는 데이타 프로세서가 '특수' 모드에서 동작하는 것을 나타낸 것이다. '특수' 모드에서의 데이타 프로세서(2) 동작에 대해, 특수 기록 제어 회로(75)는 제어 레지스터(12) 내의 감지 제어 비트에 대한 초기 기록을 억제하도록 작용하고, 그것은 기록을 유효하게 하지 않는다. 특수 기록 제어 회로(75)는 감지 제어 비트들이 아닌 비트들에 대한 기록들을 유효하게 하지 않는다. 또한, '특수' 모드에서, 특수 기록 제어 회로(75)는 감지 제어 비트들에 대한 후속 기록들을 인에이블하도록 작용하여, 후속 기록들을 유효하게 한다.

신호 정상 모드(82)는 정상 모드 또는 사용자 모드에서 동작하는 것을 나타낸다. 정상 모드에서의 데이타 프로세서(2) 동작에 대해, 특수 기록 제어 회로(75)는 제어 레지스터(12) 내의 감지 제어 비트들로 초기 기록을 억제하도록 작용하지 않는다. 또한, 정상 모드에서, 특수 기록 제어 회로(75)는 감지 제어 비트로 후속 기록들을 인에이블하도록 작용하지 않는다. 본 발명의 실시예에서, 정상 모드에서 데이타 프로세서(2) 동작에 대해, 특수 기록 제어 회로(75)는 제어 레지스터(12)내의 감지 제어 비트들에 초기 기록을 인에이블하도록 작용하는 반면, 그 비트들에 대한 모든 후속 기록들을 억제한다.

본 발명의 대안의 실시예에서, 제어 레지스터(12)는 감지 제어 비트들만을 포함하며, 감지 제어 레지스터로 고려된다. 본 발명의 실시예에서, 제어 레지스터(12)는 감지 제어 비트들, 다른 제어 비트들, 및 제어와 관련되지 않은 비트들을 포함한다. 대안의 실시예들은 제어 레지스터(12) 내의 비트들의 임의 조합을 가질 수 있으며, 다중의 레지스터들을 포함할 수도 있다.

도4는 특수 기록 제어 회로(75)의 일 실시예를 설명하며, 각 컨덕터들을 통해서 신호 특수 모드(81), 정상 모드(82), 및 리셋(84)을 수신한다. 신호 레지스터 기록(60)은 레지스터 복호 회로(50)와 결합된 컨덕터를 통해서 제공된다. 본 발명의 실시예에서, 특수 기록 제어 회로(75)는 여러 개의 컨덕터들을 통해서 제어 레지스터(12)에 연결되어있다. 본 발명의 실시예에서, 특수 기록 제어 회로(75)는 두개의 플립 플롭들, DFF2(92) 및 DFF1(90), 및 로직 게이트들(94,96,98,100,101,102 및 103)을 포함한다.

본 발명의 일실시예는 에뮬레이션 및 테스트용으로 사용된 '특수' 작동 모드를 갖는다. 시스템 에뮬레이터(16)는 도5에 설명된 데이타 처리 시스템(18)을 에뮬레이션하기 위해 '특수' 모드를 사용한다. 본 발명의 실시예에서, 데이타 처리 시스템(18), 포트 교체 장치(PRU)(26), 및 메모리(30)는 모두 시스템 에뮬레이터(16)의 부품이다. 데이타 처리 시스템(18)은 2-방향 컨덕터들, 어드레스(20)를 통해서 PRU(26) 및 메모리(30)에 어드레스 정보를 제공하고, 대안의 실시예에서는 1-방향 컨덕터들을 통해서 제공한다. 데이타 처리 시스템(18), PRU(26) 및 메모리(30)는 각각 2-방향 컨덕터들 데이타(22)를 통해서 데이타 정보를 송신 및 수신한다. 마찬가지로, 데이타 처리 시스템(18), PRU(26) 및 메모리(30)는 각각 2-방향 컨덕터들 제어(24)를 통해서 제어 정보를 송신 및 수신한다. 본 발명의 실시예에서, 사용자는 시스템 에뮬레이터(16)를 사용하여 수정 없이 어플리케이션 코드를 사용하는 데이타 처리 시스템(18)을 에뮬레이터할 수 있다. 시스템 에뮬레이터(16)는 PRU(26)를 통해서 사용자의 어플리케이션과 통신한다. PRU(26)는 제공된 포트들의 각각에 할당된 2-방향 컨덕터들에 의해서 사용자의 어플리케이션에 신호를 송신 및 수신한다. 본 발명의 실시예에서, 데이타 처리 시스템(18)은 집적 회로임에 유의한다.

본 발명의 실시예에서, 도6에 설명되어있는 바와 같이, 시스템 에뮬레이터(40)는 데이타 처리 시스템(18)의 확장된 동작을 에뮬레이트한다. 데이타 처리 시스템(18), 포트 교체 장치(PRU)(28), 및 사용자의 메모리(32)는 모두 시스템 에뮬레이터(40)의 부품이다. 데이타 처리 시스템(18)은 2-방향 컨덕터들(어드레스 42, 대안의 실시예에서는 1-방향 컨덕터들일 수 있음)을 통해서 PRU(28) 및 사용자의 메모리(32)에 어드레스 정보를 제공한다. 데이타 처리 시스템(18), PRU(28), 및 사용자의 메모리(32) 각각은 2-방향 컨덕터들(데이타(44))을 통해서 데이타 정보를 송신 및 수신한다. 마찬가지로, 데이타 처리 시스템(18), PRU(28) 및 사용자의 메모리(32) 각각은 2-방향 컨덕터들(제어(46))을 통해서 제어 정보를 송신 및 수신한다. 본 발명의 실시예에서, 사용자는 시스템 에뮬레이터(40)를 사용하여 수정없이 어플리케이션 코드를 사용하는 데이타 처리 시스템(18)을 에뮬레이터할 수 있다. 시스템 에뮬레이터(40)는 PRU(28)를 통해서 사용자의 어플리케이션과 통신한다. PRU(28)는 확장된 포트에 할당된 2-방향 컨덕터들에 의해서 사용자의 어플리케이션 및 사용자의 메모리(32)에 정보 신호들 및 제어 신호들을 송신 및 수 신한다.

바람직한 실시예의 동작

제어: 많은 데이타 프로세서들은 특수 상황들을 제외하고는 기록들에 대해 보호되는 감지 제어 레지스터들 및 비트들을 갖는다. 도1을 보면, 본 발명의 일실시예에서 데이타 프로세서(2)는 정상 모드라고 칭하는 사용자 모드에서 동작되거나, 테스트 및 에뮬레이션을 위한 '특수' 모드에서 동작된다. 사용자 모드에서, 데이타 프로세서(2)의 내부 동작은 사용자가 볼 수 없다. 사용자 모드 동작의 경우, 어플리케이션 코드는 메모리 장치(6) 내에 저장된다.

본 발명의 일 실시예에서, 데이타 프로세서(2)는 제어 레지스터(12) 내에 제어 정보를 저장한다. 제어 정보는 CPU(4)의 동작을 정의한다. 다시 말하면, 제어정보는 CPU(4) 동작 조건들에 영향을 주는 제어 파라미터들을 나타낸다. 제어 파라미터들은 제어 레지스터(12)의 각 비트들을 액세스함으로써 변경 또는 검증될 수 있다.

예를 들어, 본 발명의 실시예에서, 제어 파라미터는 인터럽트 감지성(interrupt sensitivity)이다. 인터럽트 감지성이란 인터럽트를 트리거할 입력 조건을 가리키며, 예를 들어 상승 에지가 그의 정상 동작으로부터 CPU(4) 인터럽트를 발생시킬 것이다. 인터럽트 감지성은 선택적이고 제어 레지스터(12) 내의 연관된 비트(들)를 갖는다. 인터럽트 감지성은 그러한 비트(들)에 의해서 제어된다. 마찬가지로 다른 제어 파라미터들은 제어 레지스터(12) 내의 연관된 비트(들)를 갖는다. 대안의 실시예들은 임의의 수의 제어 파라미터 또는 제어 파라미터들의 조합을 가질 수도 있다. 하나의 제어 레지스터는 통상 여러 개의 제어 파라미터들에 대해 이용할 수 있는 것에 유의한다.

작동 모드와 일치하지 않거나, 오동작 또는 불안정한 데이타 프로세서(2) 동작을 일으킬 제어 파라미터는 감지 제어 비트로 고려된다. 정확하고 확실하기 위해서, 데이타 처리 시스템 내의 불안정한 상황을 유발할 감지 제어 비트에 대한 부적합한 액세스를 방지하는 것이 바람직하다. 대안의 실시예들에서, 임의의 수의 결정들은 감지 제어 비트로서 제어 비트를 정의하기 위해서 사용되는 것에 유의한다.

도2에 설명되어 있는 바와 같이, 본 발명의 실시예에서, 제어 레지스터(12)는 제어 비트들 NECLK(34), LSTRE(36), IVIS(38) 및 PORT CONTROL(40)을 포함하며, 여기서 각각은 에뮬레이션에서 제어에 영향을 미치고, 모두는 감지 제어 비트들로서 정의된다. 어플리케이션 코드는 원하는 시스템 구성에 따라서 각 비트를 렌더링한다. 감지 제어 비트들이 일단 정의되면, 어플리케이션 코드는 또 다른 리셋이 발생될 때까지 정의를 변경시킬 수 없다. NECLK(34), LSTRE(36), IVIS(38) 및 PORT CONTROL(40)은 임의의 수의 가능한 감지 비트들의 예이고, 대안의 실시예들은 어플리케이션에 의해서 감지성이 결정되고, 다른 중요도의 비트들을 포함할 수도 있다.(예를 들면 제어 비트가 없다)

사용자 모드나 정상 모드는 사용자의 동작 환경이므로, 유연성 및 확실성을 의미한다. 본 발명의 실시예에서, 사용자가 데이타 프로세서(2)의 리셋 후 한 번만 특정 제어 비트를 선택할 수 있도록 함으로써 두 가지 목적들이 달성된다. 사용자는 상황 보증으로서 다양한 제어들의 변화가 유연성을 갖지만, 데이타 프로세서(2) 의 리셋 없이, 한 번 셋팅된 제어들을 변경시키지 않도록 한다.

'특수'모드 : 본 발명의 실시예에서, '특수' 모드는 데이타 프로세서(2)의 내부 동작의 가시성을 허용한다. 다른 용도 중에서, '특수' 모드는 데이타 프로세서(2)의 사용자 에뮬레이션을 위해 사용된다. 에뮬레이션은 통상 데이타 프로세서를 교체하기 위해 사용되는 에뮬레이션 시스템을 필요로 한다. 본 발명의 일 실시예의 시스템 에뮬레이터(40)는 도6에 설명되어있다. 시스템 에뮬레이터(40)는 '특수'모드에서 기능하는 데이타 프로세서(2)를 포함하며, 여기서 어드레스(42), 데이타(44) 및 제어(46)와 같은 내부 신호들은 데이타 프로세서(2)로 외부에 제공된다. 외부 신호들을 제공하는 것은 통상 사용자 모드의 다양한 포트들에 할당되는 데이타 프로세서(2)의 자원들을 필요로 한다. 시스템 에뮬레이터(40)는 다양한 포트를 재구축하는 PRU(28)를 포함한다. 어드레스(42), 데이타(44) 및 제어(46)로부터 수신한 입력들에 기초해서, PRU(28)는 외부 일치성을 허용하는 데이타 프로세서(2)의 포트 자원들로서 작용한다. 이런 식으로, 데이타 프로세서(2)는 시스템 에뮬레이터(40)를 인에이블링하는 성분으로서 작용하고, 시스템 에뮬레이터(40)는 사용자의 어플리케이션에서 데이타 프로세서(2)로서 작용한다.

시스템 구성은 에뮬레이션 동안 적합한 동작에 중요하다. 사용자가 그러한 구성을 변경시키면, 시스템 에뮬레이터(40)는 적절하지 못하게 작용하거나 기능을 완전히 멈출 수도 있다. 어플리케이션 코드는 에뮬레이터의 구성을 변경시키는 문제가 있다. 어플리케이션 코드가 종종 에뮬레이션 목적을 위해 수정되어 제어 비트들의 임의의 렌더링을 제거한다. 명백한 단점은 사용자 모드를 위한 것 하나와 및 에뮬레이션을 위한 것 하나인, 두 셋트들의 어플리케이션 코드에 대한 필요성이다. 본 발명은 제어 비트에 대한 초기 기록을 무시(disregard or ignore)함으로써, 어플리케이션 코드가 에뮬레이션 동안 사용될 수 있도록 한다. 본 발명의 다른 실시예에서, PRU(28)는 포트들을 셋업하고 재구축하기 위한 초기 기록을 이용한다.

본 발명의 일실시예에서, 도3에 도시된 바와 같이, 특수 기록 제어 회로(75)는 제어 레지스터(12) 내의 감지 제어 비트의 적합한 렌더링을 실행 억제하는 작용을 한다. 여기에서, 제어 레지스터(12)는 감지가 아닌 제어 비트들과 감지 제어 비트들(예를 들면 비트1(71)), 및 제어 비트들이 아닌 비트들(예를 들면 비트0(70))을 포함한다. '특수' 모드 동안, 특수 기록 제어 회로(75)는 이 실시예에서 감지 제어 비트인 비트(71)의 제1 렌더링을 억제하지만, 감지 제어 비트가 아닌 비트0(70)의 제1 렌더링을 억제하지 않을 것이다. 정상 모드 동안, 특수 기록 제어 회로(75)는 비트1(71)의 초기 렌더링을 가능하게 하는 작용을 하지만, 비트(71)에 후속 기록을 억제하며, 반면, 레지스터 복호 회로(50)는 비트0(70)의 초기 및 후속 렌더링에 영향을 미친다. 한 레지스터 내의 제어 비트(들), 비 제어 비트(들), 및 감지 제어 비트(들)를 조합함으로써, 데이타 프로세서(2)는 보류되어 미사용된 비트(들)를 갖는 과도한 레지스터들을 갖지 않는다. 도4는 논리 게이트들과 다른 로직 디바이스들을 포함하는 본 발명의 실시예를 설명한다. 대안의 실시예들은 감지 제어 비트(들)의 유사한 제어에 영향을 미치는 임의의 종류의 회로를 사용할 수도 있다.

어플리케이션 코드 : 본 발명의 일실시예에서, 어플리케이션 코드는 정확히 한 번 각 제어 비트에 기록할 필요가 있다. 각 비트가 기록되면, 시스템 구성은 정의되고, 다음번 리셋까지 유지된다. 그러면, 어플리케이션 코드는 감지 제어 비트에 제1 기록이 무시되는 에뮬레이션으로 실행 될 수 있다. 감지 제어 비트에 대한 후속 기록은 유효하다. 감지 제어 비트에 다중 기록들을 포함하는 어플리케이션 코드가 있으면, 에뮬레이터는 대부분 오동작하여, 어플리케이션 코드가 감지 제어 비트에 다중 기록들을 포함한다는 경고를 사용자에게 제공한다.

본 발명의 일실시예에서, 특정 제어 비트들은 그들의 기능이 에뮬레이션에 영향을 미칠 때에만 정의된 감지 제어 비트들이다. 본 발명의 실시예에서, 감지 제어 비트들은 제어 레지스터(12)에서 조합된다. 본 발명의 대안의 실시예는 감지 제어 비트들로서 모든 제어 비트들을 정의한다. 다른 실시예들에서, 반드시 제어 비트들일 필요는 없는 임의의 레지스터 비트들이 감지(sensitive)로서 정의된다.

본 발명의 실시예에서, NECLK(34)는 제어 레지스터(12) 내의 비트이며, 데이타 프로세서(2)가 외부 클록 신호를 제공할 것인지를 결정하기 위해서 사용된다. 사용자 모드에서, 버스 인터페이스 장치(10) 및 주변장치들은 외부 클록 신호들을 사용한다. '특수' 모드에서, 외부 클록 신호는 내부 명령 경로들을 트래킹하기 위해서 사용되고, 버스 인터페이

스 장치(10)에서처럼 버스 인터페이스에 영향을 미치게 하기 위해 사용된다. 시스템 구성은 이 결정에 의존한다. 또 다른 자원들은 외부 클록 신호가 제공되지 않을 때 포트 기능들을 위해서 유용할 것이다. '특수' 모드는 사용자가 유연하게 이 결정을 하도록 해준다. 통상, 외부 클록 신호는 사용자 모드에서 볼 수 없는 신호들(예컨대, 어드레스와 데이터 신호들)이 제공되는 확장 모드에서 사용된다. 그러나, 일단 결정이 되면, 사용자는 이것이 시스템 구성에 필수 결정이므로 한번만 NECLK(34)를 렌더링해야 한다.

어플리케이션 코드는 각 리셋 후 NECLK(34)에 한번만 기록하는 것을 포함한다. 본 발명의 실시예에서, 사용자 모드는 모든 후속 기록들이 무시되는 NECLK(34)에 한 번 기록만을 허용하는 반면, '특수' 모드는 NECLK(34)에 제1 기록을 무시하고, 후속되는 모든 기록들에 영향을 미친다. 사용자는 NECLK(34)에 한 번만 기록하는 사용자 모드 요건에 따르는 어플리케이션 코드를 기록한다. 어플리케이션 코드는 다음에 '특수' 모드에서 에뮬레이션을 위해 유용하며, 여기서 제1 기록은 무시될 것이다. 이것은 시스템 구성이 에뮬레이터에 의해 미리 정해지는 에뮬레이션에서 중요하다. 또한, 이들 비트들의 다중 기록을 실수로 포함하는 어플리케이션 코드가 있으면, 에뮬레이션은 에러를 드러낸다.

도2에 도시된 바와 같이, 본 발명의 실시예는 제어 레지스터(12) 내에서 조합된 NECLK(34), LSTRE(36), IVIS(38) 및 PORT CONTROL(40) 제어 비트들을 포함한다. 여기서, NECLK(34), LSTRE(36), IVIS(38) 및 PORT CONTROL(40)는 각 연관된 제어 기능이 에뮬레이션에 영향을 미치므로 감지 제어 비트로서 정의된다.

LSTRE(36)는 로우 스트로브 신호를 허용하도록 사용되고, 이것은 데이타 프로세서(2)에 의해서 제공된다. 로우 스트로브 신호는 시스템 자원들을 이용하고, 이는 그렇지 않았다면 포트 기능들에 유용했을 것이고, LSTRE(36)을 인에이블하기 위한 결정은 시스템 구성 고려에 필수적이다.

마찬가지로, 내부 가시성 기능들을 인에이블하는 IVIS(38), 및 특정 포트 구 성을 인에이블하는 PORT CONTROL(40)는 시스템 자원들과 시스템 구성에 영향을 미친다. IVIS(38) 및 PORT CONTROL(40)은 시스템 구성 고려들에 필수적이다. 본 발명의 한 실시예에서, 모두 네 개의 제어 신호들은 시스템 구성 및 시스템 자원 할당에 영향을 미치지만, 사용자는 이들 결정들에 의해서 영향을 받지 않는 시스템 구성을 결정할 수 있음을 유의한다. 그러나, 통상 시스템 구성에 영향을 미치는 제어 기능들은 데이타 프로세서(2) 어플리케이션에서 중요하다.

본 발명의 일실시예에서, 각 결정은 시스템 구성에 영향을 미치기 때문에 네개의 제어 기능들의 각각은 리셋 후 한 번만 정의된다. 시스템 구성은 각 제어에 의존하며, 시스템 자원들이 할당되고, 따라서 한 번 결정되면, 시스템 구성은 유지된다.

에뮬레이션: 본 발명은 어플리케이션 코드가 수정 없이 에뮬레이션에 사용된다. 본 발명의 실시예에서, 사용자는 에뮬레이션의 수용을 고려하지 않고, 어플리케이션 코드를 기록하는데 자유롭다. 어플리케이션 코드는 한 번만 감지 제어 비트에 기록하는 것이며, 이것은 사용자가 더 선호하는 동작을 수립한다. 사용자는 다음에, 에뮬레이션을 위해 직접 어플리케이션 코드를 사용할 수도 있다. 에뮬레이션에서, 감지 제어 비트에 대한 초기 기록은 무시됨으로써, 에뮬레이션 시스템 구성을 유지한다. 감지 제어 비트들에 대한 다중 기록들을 포함하는 어플리케이션 코드는 에뮬레이터를 손상시킨다. 어플리케이션 코드는 일반적으로 정상 동작 요건을 만족시키기 위해 기록되고, 제어 비트들에 대한 초기 기록은 유효이지만, 후속 기록들은 유효하지 않다.

도5를 참고하면, 시스템 에뮬레이터(16)는 디버그, 설계 및 조사용으로 사용된다. 시스템 에뮬레이터(16)는 사용자 어플리케이션에 대한 '대화'에 PRU(26)를 사용한다. 시스템 에뮬레이터(16)는 사용자 어플리케이션에 연결되거나 '플러그인'된다. 데이타 처리 시스템(18)은 '특수' 모드에서 실제로 오퍼레이팅하는 동안에도 가상 타입의 사용자 모드로서 수행함으로써, 어플리케이션 동작 동안 내부 가시성을 허용한다. 에뮬레이션은 응용 설계에서 필수적이다. 본 발명의 실시예에서, 시스템 에뮬레이터(16)는 데이타 처리 시스템(18)의 포트 자원들을 사용하고, 데이타 처리 시스템(18)은 데이타 프로세서(2)를 포함함으로써, 실제 포트 자원들을 응용 용도로 이용할 수 없게 만든다. 에뮬레이션에서, 데이타 처리 시스템(18)은 '특수' 모드에서 동작하며, 그의 동작 환경으로 변경하면 오동작을 하게 될 것이다. 데이타 처리 시스템(18)을 위해 기록된 어플리케이션 코드는 감지 제어 비트를 렌더링하는 명령들을 포함한다. 에뮬레이션에서 사용하기 위해서, '특수' 모드는 초기 렌더링을 무시하지만, 모든 후속 렌더링을 허용한다. 어플리케이션 코드가 감지 제어 비트의 단일 렌더링만을 수행하면, 에뮬레이션은 부드럽게 작동할 것이다.

확장된 에뮬레이션에서, 데이타 처리 시스템(18)은 도6에 도시된 바와 같이, '특수' 모드에서 작동하지만, 시스템 에뮬레이터(40)는 확장된 어플리케이션과 인터페이스하기 위해서 PRU(28)을 사용한다. 여기서 포트 자원들은 포트들로서 보다는 확장된 기능들을 위해서 사용된다. 에뮬레이션에서, 데이타 처리 시스템(18) 오퍼레이팅 환경의 변화는 마찬가지로 오동작을 유발할 것이다.

본 발명은 어플리케이션 코드를 수정하기 위한 요구를 제거하면서, 데이타 프로세서(2)의 임의의 수의 에뮬레이션 기술을 가능하게한다. 또 다른 고려는 테스트 조건들 하에서의 동작이다.

테스트: 본 발명자들의 레지스터 액세스 스킴은 디바이스 시험을 위한 또다른 이점을 갖는다. 본 발명의 실시예에서, 데이타 프로세서(2)는 테스트하기 위한 '특수' 모드에 있고, 각 감지 제어 비트에 대한 제1 기록은 무시되지만, 각 후속 기록들은 유효하다. 에뮬레이션과 대조적으로, 테스트는 시스템의 유효한 재구성을 필요로 한다. 후속 기록에 영향을 미치지만 제1 기록을 무시함으로써, 데이타 프로세서(2)는 리셋 시퀀스를 실행하지 않고 다양한 시스템 구성에서 테스트될 수 있다. 이것은 향상된 테스트 시간 및 증가된 생산 유연성을 고려한다.

결론 : 결론적으로, 본 발명은 감지 제어들을 보호하면서 레지스터를 액세스하는 방법을 제공한다. 특정 비트의 초기 렌더링이 무효이지만, 후속 렌더링들은 유효임을 '특수' 모드에서 제공함으로써, 본 발명은 에뮬레이션 및 테스트의 일관성있는 해결책을 제공한다. 감지 제어들은 부적절한 사용으로부터 보호되고, 사용자 요구는 에뮬레이션의 수정된 어플리케이션 코드를 생성함으로써 이중 노력을 할 필요가 없다. 본 발명은 장치 테스트에서 사용하기 위해 일관적이며, 테스트 동안 감지 제어 비트들에 반복 액세스를 허용한다.

특정 실시예들을 참조하여 본 발명을 설명하였지만, 추가의 수정들 및 개선들을 할 수 있다는 것을 당업계의 숙련자들에게는 자명한 일이다. 예를 들어, 본 발명의 몇 가지 용도들을 여기서 설명하였지만, 집적 회로 내에서 '제1 기록은 유효하지 않고, 적어도 하나의 후속 기록은 유효한' 레지스터 비트 또는 메모리 로케 이션을 위한 광범위한 용도가 있다. 본 발명의 대안의 실시예들은 '제1의 N번 기록들'이 유효하지 않으며, 여기서 'N'은 소정의 정수이거나, 사용자에 의해서 프로그램된 변수이거나, 하드웨어에 의해서 결정된 값(예를 들면 계수기에서 증가된 값)임에 유의한다. 따라서, 본 발명은 설명된 특정 형태들에만 한정되지 않으며, 첨부된 청구범위는 본 발명의 사상과 범위로부터 벗어나지 않는 모든 수정들을 포함한다.

(57) 청구의 범위

청구항 1.

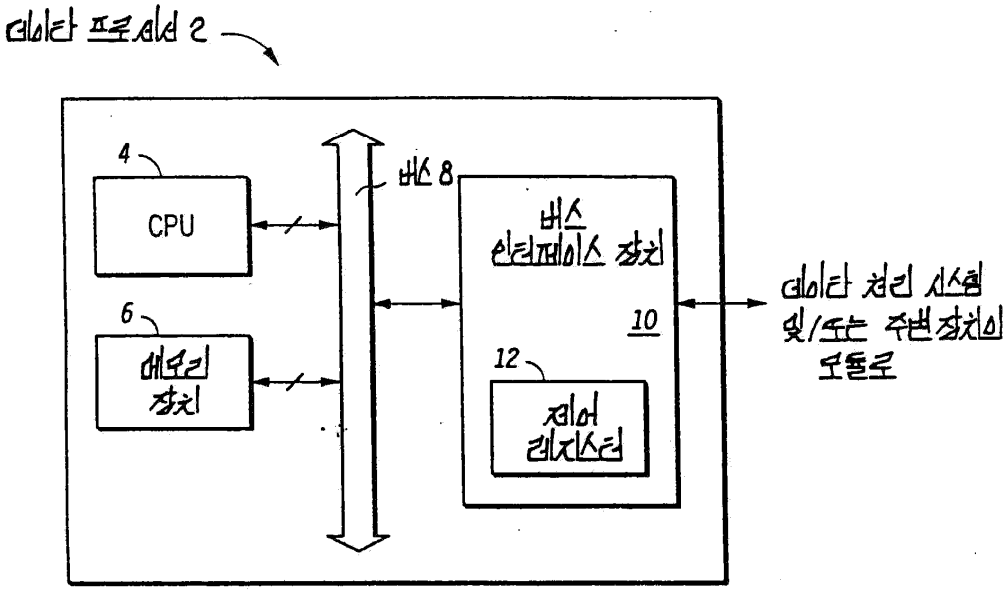
데이타 프로세서를 동작시키기 위한 방법에 있어서,
제1 제어 비트에 대한 제1 기록 액세스(write access)를 수행하는 단계로서, 상기 제1 기록 액세스는 상기 데이타 프로세서의 리셋 후의 초기 기록 액세스이고, 상기 제1 제어 비트는 상기 데이타 프로세서의 적어도 하나의 파라미터를 구성하기 위한 것인, 상기 제1 기록 액세스 수행 단계와,
상기 제1 제어 비트에 대한 상기 제1 기록 액세스가 상기 제1 제어 비트에 영향을 미치지 않게 하는 단계와,
상기 제1 제어 비트에 대한 제2 기록 액세스를 수행하는 단계로서, 상기 제2 제어 기록 액세스는 상기 제1 기록 액세스에 후속하는, 상기 제2 기록 액세스 수행단계와,
제1 제어 비트에 대한 제2 기록 액세스가 상기 제1 제어 비트에 영향을 미치지 않도록 하는 단계와,
상기 제1 제어 비트에 대한 제 3 기록 액세스를 실행하는 단계로서, 상기 제 3 기록 액세스는 상기 제2 기록 액세스에 후속하며, 상기 제1 제어 비트에 대한 상기 제 3 기록 액세스가 상기 제1 제어 비트에 영향을 미치도록 하는 단계를 포함하는, 데이타 프로세서 동작 방법.

청구항 2.

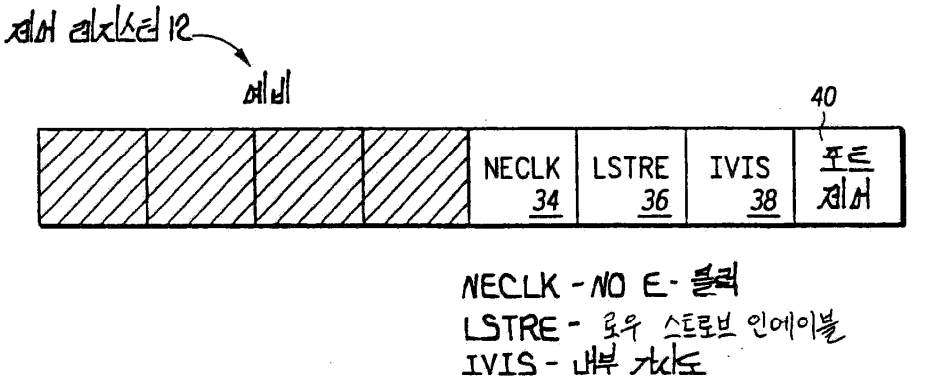
집적 회로에 있어서,
레지스터, 및
상기 레지스터에 대한 현재 기록 액세스가 리셋 후의 제1 기록 액세스인지의 여부를 결정하기 위해 상기 레지스터에 결합된 제어 회로를 포함하고,
상기 레지스터에 대한 현재 기록 액세스가 리셋 후의 상기 레지스터에 대한 상기 제1 기록 액세스이면, 상기 제어 회로는 상기 레지스터에 대한 상기 현재 기록 액세스가 상기 레지스터에 영향을 미치지 않도록 하며,
상기 레지스터에 대한 상기 현재 기록 액세스가 리셋 후의 상기 레지스터에 대한 제1 기록 액세스가 아니면, 상기 제어 회로는 상기 레지스터에 기록함으로써 상기 레지스터에 대한 상기 현재 기록 액세스가 상기 레지스터에 영향을 미치게 하는, 집적 회로.

도면

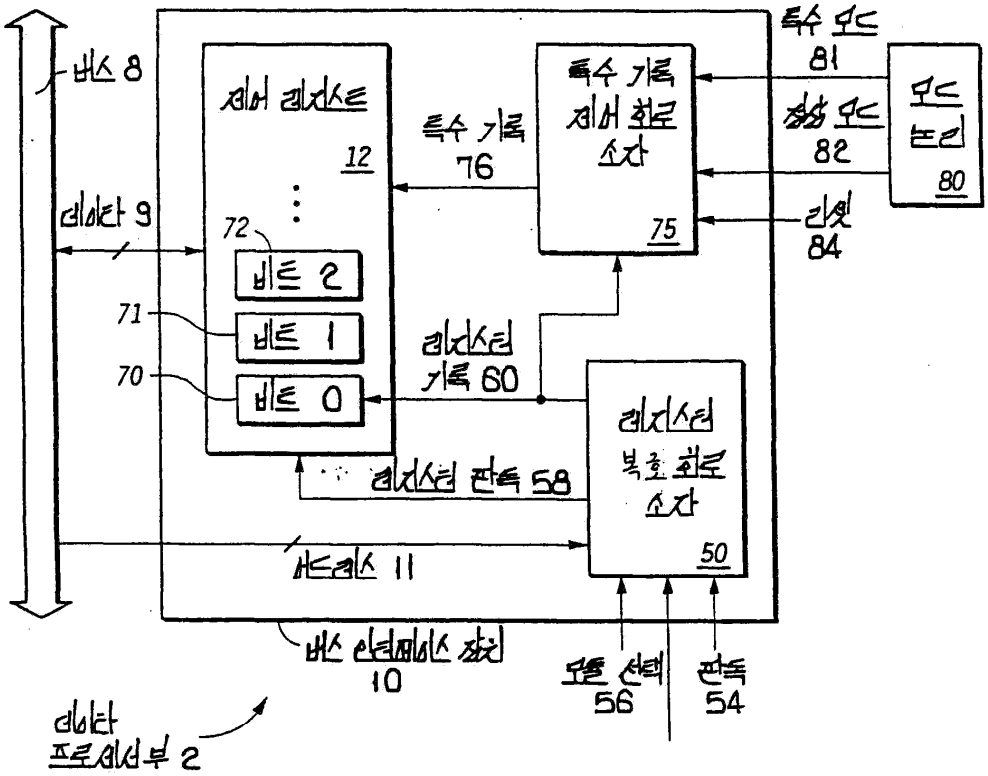
도면1



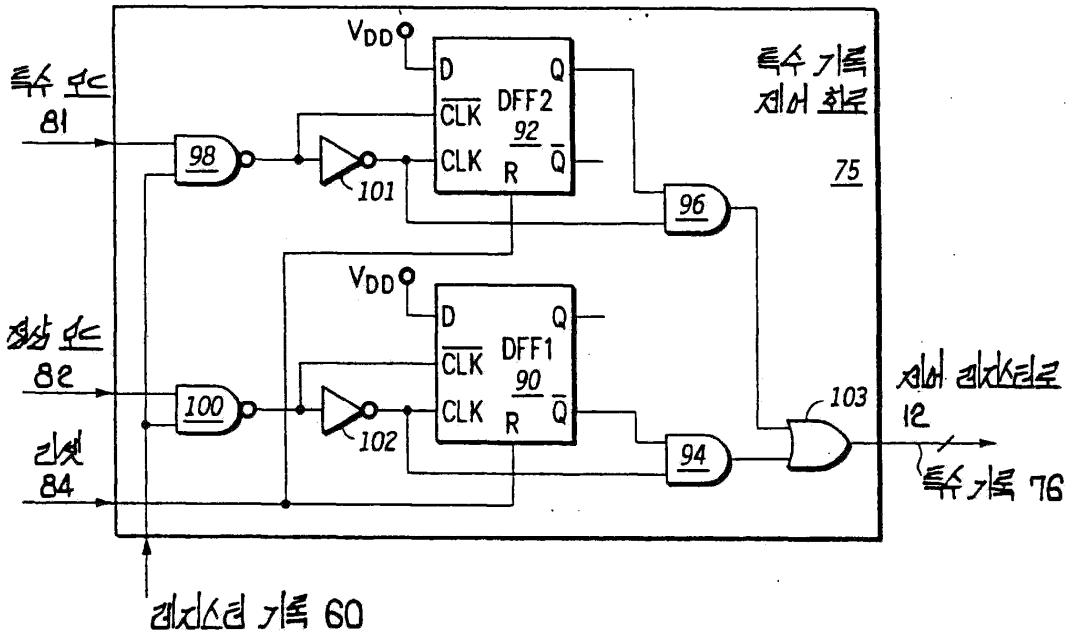
도면2


제어 레지스터 12
예비
40
NECLK 34
LSTRE 36
IVIS 38
포트 제어
NECLK - NO E- 클럭
LSTRE - 로우 스트로브 인에이블
IVIS - 내부 가시도

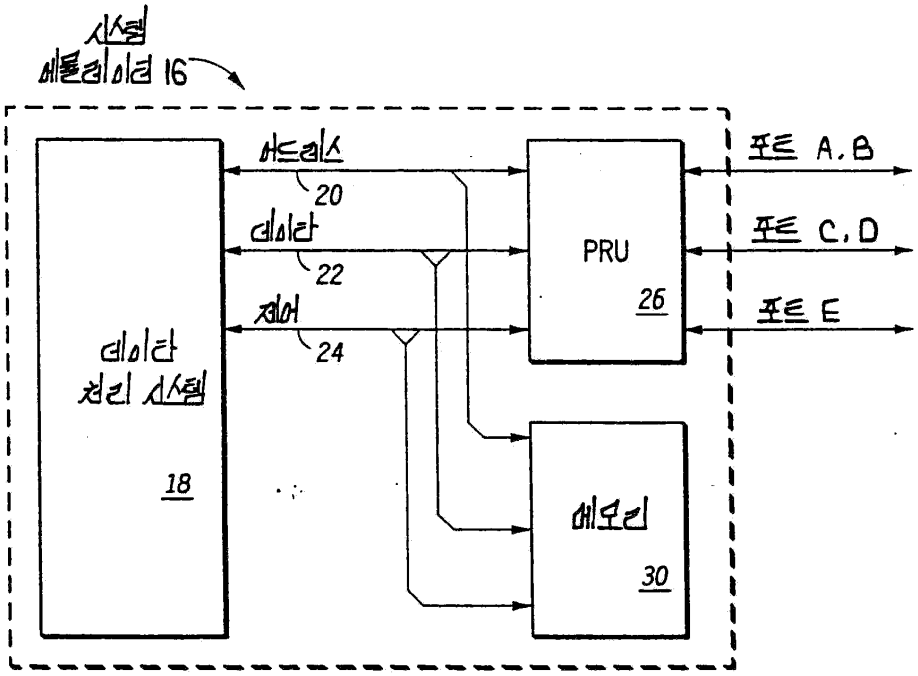
도면3


버스 8
데이타 9
제어 레지스트 12
72
비트 2
71
비트 1
70
비트 0
특수 기록 76
특수 기록 제어 회로 소자 75
특수 모드 81
정상 모드 82
리셋 84
모드 논리 80
레지스터 기록 60
레지스터 복호 회로 소자 50
레지스터 판독 58
어드레스 11
버스 인터페이스 장치 10
모드 선택 56
판독 54
데이타 프로세서부 2

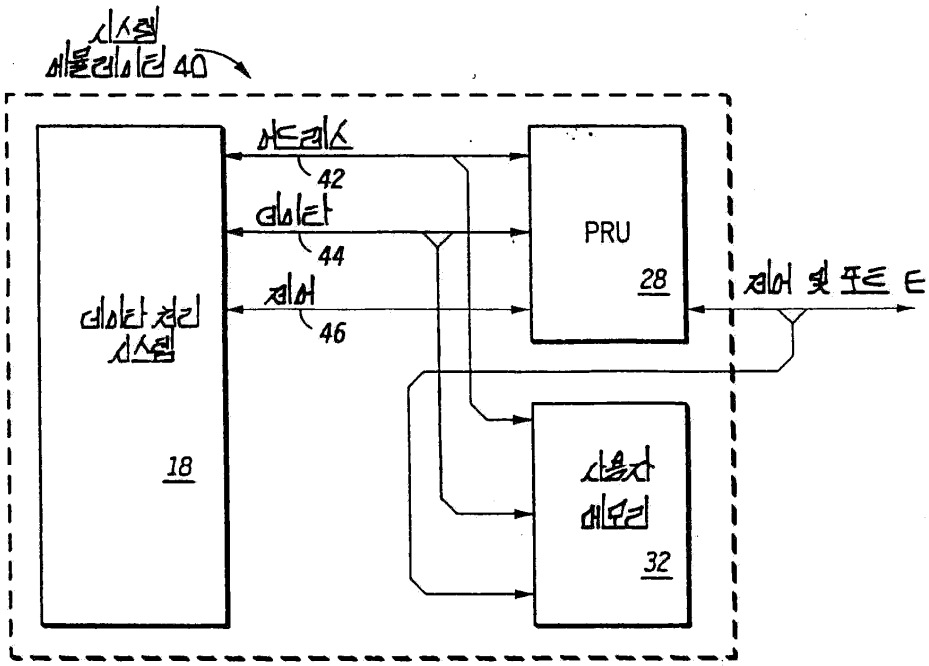
도면4


특수 모드
81
정상 모드
82
리셋
84
98
100
101
102
VDD
D
Q
CLK
DFF2
92
R
DFF1
90
96
94
103
특수 기록
제어 회로
75
제어 레지스터로
12
특수 기록 76
레지스터 기록 60

도면5


시스템
제어기 16
어드레스
20
데이타
22
제어
24
데이타
처리 시스템
18
PRU
26
메모리
30
포트 A, B
포트 C, D
포트 E

도면6


시스템
서플리이터 40
어드레스
42
데이타
44
제어
46
데이타 처리
시스템
18
PRU
28
제어 및 포트 E
사용자
메모리
32