



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2007년12월17일
(11) 등록번호 10-0786418
(24) 등록일자 2007년12월10일

(51) Int. Cl.

G11C 7/00 (2006.01)

(21) 출원번호 10-2001-0058049
(22) 출원일자 2001년09월19일
심사청구일자 2006년09월15일
(65) 공개번호 10-2002-0022618
(43) 공개일자 2002년03월27일
(30) 우선권주장
09/665,892 2000년09월20일 미국(US)
(56) 선행기술조사문헌
JP09016469 A
(뒷면에 계속)

(73) 특허권자

애질런트 테크놀로지스, 인크.

미국 캘리포니아주 95052-8059 산타 클라라 스트리
븐스 크릭 불러바드 5301

(72) 발명자

쿡존에이치3세

미국콜로라도주80526포트콜린스마리골드레인607

싱프리트피

미국캘리포니아주94539

프리몬트사우비그너코트48888

푸엔테에드먼도테라

미국캘리포니아주95014

쿠퍼티노이튼플레이스21815

(74) 대리인

김창세, 장성구

전체 청구항 수 : 총 10 항

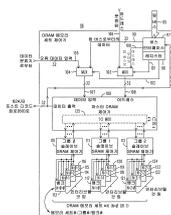
심사관 : 윤난영

(54) DRAM에서의 메모리 동작 수행 방법

(57) 요약

여러 캐치 RAM(32)에서의 DRAM의 동작 속도는 그룹(88)내의 메모리의 상이한 बैं크(113-116)에 대한 신호의 인터리빙과 बैं크의 이들 그룹간의 멀티플렉싱의 조합에 의해 증가될 수 있다. 그룹에 대한 신호를 위한 가요성 4개 인터리빙 방식과 조합된 4개 बैं크 각각의 3개의 그룹간 3-웨이 멀티플렉싱은 12의 계수에 달하는 속도 증가를 발생하는 한편, 3개의 메모리 버스만을 필요로 한다. 12개 बैं크의 각각은 전체 이용 가능한 어드레스 공간을 나타내고, 어떠한 개별적인 기입 사이클도 12개 बैं크의 어떠한 것도 액세스할 것이다. 유틸리티 메카니즘(146)은 어드레스에서의 관독 사이클 동안 모두 12개 बैं크에 대한 결과를 단일화된 결과로 작성한다. 추가의 기입 동작이 또다른 작성 단계에 대한 필요성을 발생할 수 있기 때문에, 작성된 결과의 무결성을 추적하기 위한 메카니즘이 존재한다. 2개의 내부 SRAM(75, 76)과 2개의 외부 DRAM(73, 74)으로 이루어진 4개의 메모리 세트가 존재한다. SRAM은 VLSI 회로의 집적 부품인 한편, DRAM은 그 VLSI에 인접한 개별적인 패키징 부품이다. DRAM의 양은 선택적이다. DRAM 메모리 세트에 대해, 멀티플렉싱과 인터리빙 모드는 최대 100 Mhz 속도에서의 전체 랜덤 액세스를 허용한다. 33 Mhz의 속도를 초과하지 못할 DRAM은 하나의 더 대형 그룹에 관한 인터리빙을 선호하여 멀티플렉싱을 제거함으로써 더 낮은 속도에 대한 보상으로 깊이를 3배로 제공하도록 구성될 수 있으며, 멀티플렉싱의 일부로서 사용된 बैं크 인에이블 비트는 이제 정규 어드레스 비트로 사용되어 잔류하는 하나의 그룹의 어드레스 공간의 크기를 증가시킬 수 있다. DUT에 대한 테스트가 선형 모드의 액세스에 적합화된다면, 테스터가 DUT를 작동시킬 수 있는 최고 속도에서 DUT가 테스트될 때에도, DUT가 메모리 깊이에서의 12배의 증가가 이용 가능하게 된다. 이것은 한 번에 단일 बैं크내의 어드레스를 선호하여 인터리빙 방식을 제거한다. 또다른 재구성은 하나의 메모리 세트에 외부 메모리 세트를 결합하여 다른(즉, 속도 관련) 모드의 동작에 상관없이 비결합된 세트의 2배의 깊이를 갖도록 하는 것이다.

대표도 - 도5



(56) 선행기술조사문헌
JP09017197 A
JP10269799 A
KR1020000033120 A
W09620479 A1
US5905909 A
JP10055694 A

특허청구의 범위

청구항 1

어드레스 공간내의 각각의 어드레스와 관련된 정보의 워드에 대해 DRAM(73)에서 메모리 동작을 수행하는 방법에 있어서,

(a) DRAM의 $(n \times m)$ 개 뱅크(113-124)를 그룹당 m 개 뱅크의 n 개 그룹(88-90)으로 조직하되, 여기서 각각의 뱅크는 어드레스 공간내의 각각의 어드레스에 대한 어드레스 가능한 위치를 가지는 단계와,

(b) 각각의 메모리 동작을 메모리 동작의 정해진 순서의 사이클 시퀀스로 다음 그룹에 순차적으로 전송하는 단계와,

(c) 각각의 그룹내에서 그 안의 각각의 뱅크를 정해진 순서의 사이클 시퀀스로 선택하는 단계와,

(d) 각각의 그룹내에서 그리고 그 그룹에 대해 상기 단계 (b)에 의해 전송된 연속적인 메모리 동작에 대해, 상기 단계 (c)의 정해진 순서의 사이클 시퀀스에 따라 그룹의 m 개 뱅크 사이에 상기 연속적인 메모리 동작을 순차적으로 인터리빙하는 단계와,

(e) 한 그룹내의 각각의 선택된 뱅크에 대해 그리고 상기 단계 (d)에 의해 상기 뱅크로 전송된 인터리빙된 메모리 동작에 대해, 상기 어드레스 공간내의 어드레스에서 상기 단계 (d)의 다음 연속적인 메모리 동작을 수행하는 단계를 포함하는 것을 특징으로 하는

DRAM(73)에서의 메모리 동작 수행 방법

청구항 2

제 1 항에 있어서,

상기 메모리 동작은 기입(writing)이며,

어드레스 가능한 피테스트 소자(14)에 대해 수행된 테스트로부터 기입될 정보를 획득하는 단계를 더 포함하는 것을 특징으로 하는

DRAM(73)에서의 메모리 동작 수행 방법.

청구항 3

제 2 항에 있어서,

상기 피테스트 소자(14)는 메모리이고,

기입될 워드내의 비트는 메모리 테스트 시스템내의 채널을 나타내며,

상기 피테스트 소자에 인가된 어드레스로부터 구해진 어드레스를 가지고 $(n \times m)$ 개 뱅크내의 어드레스 가능한 위치를 어드레스하는 단계를 더 포함하는 것을 특징으로 하는

DRAM(73)에서의 메모리 동작 수행 방법.

청구항 4

제 1 항에 있어서,

상기 메모리 동작은 판독이며,

$(n \times m)$ 개 워드를 발생하기 위해 $(n \times m)$ 개 뱅크(113-124) 전체내의 동일 어드레스에서의 어드레스 가능한 위치로부터 판독하는 단계와,

상기 $(n \times m)$ 개 워드를 하나의 최종 워드로 병합하는 단계(146)와,

상기 최종 워드를 메모리 동작의 결과로서 취하는 단계와,

상기 최종 워드를 동일 어드레스에서 $(n \times m)$ 개 뱅크 전체에 기입하는 단계를 더 포함하는 것을 특징으로 하는

DRAM(73)에서의 메모리 동작 수행 방법.

청구항 5

제 4 항에 있어서,

상기 최종 워드가 (n×m)개 뱅크(113-124) 전체에 기입되는 시점에 플래그(191)를 세트하는 단계와,

다음 연속 메모리 동작이 기입 동작인 단계 (b)의 후속 실행(instance)에서 플래그를 소거하는 단계를 더 포함하는 것을 특징으로 하는

DRAM(73) 메모리 동작 수행 방법.

청구항 6

그룹 선택 부분 및 뱅크내 어드레스 부분을 포함하는 어드레스 공간 내의 각각의 어드레스와 관련된 정보의 워드에 대해 DRAM(73)에서 메모리 동작을 수행하는 방법에 있어서,

(a) DRAM의 (n×m)개 뱅크(113-124)를 그룹당 m개 뱅크의 n개 그룹(88-90)으로 조직하되, 여기서 각각의 그룹은 상기 그룹 선택 부분(167, 169, 171)에 의해 선택 가능하고, 그룹내의 각각의 뱅크는 상기 뱅크내 어드레스 공간(176)에 의해 어드레스 가능한 위치를 가지는 단계와,

(b) 각각의 다음 메모리 동작을 상기 그룹 선택 부분에 의해 식별된 그룹으로 전송하는 단계와,

(c) 각각의 그룹내에서 그 안의 각각의 뱅크를 정해진 순서의 사이클 시퀀스로 선택하는 단계와,

(d) 각각의 그룹내에서, 그리고 그 그룹에 대해 상기 단계 (b)에 의해 전송된 연속적인 메모리 동작에 대해, 각각의 뱅크가 상기 단계 (c)의 정해진 순서의 사이클 시퀀스에 따라 선택될 때 그룹의 m개 뱅크 사이에 상기 연속적인 메모리 동작을 순차적으로 인터리빙하는 단계(161)와,

(e) 상기 그룹내의 각각의 선택된 뱅크에 대해 그리고 상기 단계 (d)에 의해 상기 뱅크로 전송된 인터리빙된 메모리 동작에 대해, 상기 뱅크내 어드레스 부분에 의한 선택된 뱅크의 위치에서 상기 단계 (d)의 다음 연속 메모리 동작을 수행하는 단계를 포함하는 것을 특징으로 하는

DRAM(73)에서의 메모리 동작 수행 방법.

청구항 7

제 6 항에 있어서,

상기 메모리 동작은 기입이며,

어드레스 가능한 피테스트 소자(14)에 대해 수행된 테스트로부터 기입될 정보를 획득하는 단계를 더 포함하는 것을 특징으로 하는

DRAM(73)에서의 메모리 동작 수행 방법

청구항 8

제 7 항에 있어서,

상기 피테스트 소자(14)는 메모리이고,

기입될 워드내의 비트는 메모리 테스트 시스템내의 채널을 나타내며,

상기 피테스트 소자에 인가된 어드레스로부터 구해진 어드레스를 가지고 m개 뱅크의 n개 그룹(88-90)내의 어드레스 가능한 위치를 어드레스하는 단계를 더 포함하는 것을 특징으로 하는

DRAM(73)에서의 메모리 동작 수행 방법.

청구항 9

제 6 항에 있어서,

상기 메모리 동작은 판독이며,

m개 워드를 발생하기 위해 상기 그룹 선택 부분에 의해 식별된 그룹의 m개 뱅크 전체내의 동일 어드레스에서

의 어드레스 가능한 위치로부터 관독하는 단계와,
 상기 m개 워드를 하나의 최종 워드로 병합하는 단계(146)와,
 상기 최종 워드를 메모리 동작의 결과로서 취하는 단계와,
 상기 최종 워드를 상기와 같이 식별된 그룹의 동일 어드레스에서 m개 뱅크 전체에 기입하는 단계를 더 포함하는 것을 특징으로 하는

DRAM(73)에서의 메모리 동작 수행 방법.

청구항 10

제 9 항에 있어서,
 상기 최종 워드가 m개 뱅크 전체에 기입되는 시점 무렵에 플래그(191)를 세트하는 단계와,
 다음 연속 메모리 동작이 기입 동작인 단계 (b)의 후속 실행에서 플래그를 소거하는 단계를 더 포함하는 것을 특징으로 하는

DRAM(73)에서의 메모리 동작 수행 방법.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

- <17> 본 발명은 크기와 속도에 대해 조정 가능한 SDRAM 메모리 세트를 갖는 메모리 테스터용 에러 캐치 RAM에 관한 것이다.
- <18> 전자 장치와 그 전자 장치로 얻을 수 있는 성능은 일상 생활에서 극히 보편적인 것이 되었다. 가정의 개인용 컴퓨터와 함께, 많은 개별 장치들이 다양하고 잡다한 목적을 위해 한 가지 이상의 생산성 도구의 기능을 수행한다. 대부분의 개인용 생산성 전자 장치는 여러 형태의 비휘발성 메모리를 포함한다. 휴대 전화는 전원이 턴오프될 때 사용자 입력 전화 번호 및 구성을 저장하고 유지하기 위해 비휘발성 메모리를 사용한다. PCMCIA 카드는 컴퓨터내의 슬롯에서 제거될 때에도 정보를 저장 및 유지하기 위해 비휘발성 메모리를 사용한다. 많은 다른 보편적인 전자 장치 또한 전원이 공급되지 않은 상태에서 비휘발성 메모리의 장기간 저장 성능의 이점을 얻는다.
- <19> 전자 설비 제조업체에 비휘발성 메모리를 판매하는 비휘발성 메모리 제조업체는 그들이 생산하는 메모리의 적합한 동작을 검사하여 검증하기 위해 테스터를 필요로 한다. 대량의 비휘발성 메모리가 제조되어 점점 더 낮은 가격으로 판매되고 있기 때문에, 단일 파트(part)를 테스트하는데 소요되는 시간을 최소화하는 것이 매우 중요하다. 비휘발성 메모리의 구입업체는 메모리 장치를 최소의 테스트만으로 또는 테스트하지 않고서도 더 고가의 조립 장치에 통합함으로써 얻어지는 비용 절감 때문에 메모리 제조업체에게 높은 출하율(shipment yield)을 제공하도록 요구한다. 따라서, 메모리 테스트 공정은 부합하지 않는 파트(non-conforming part)의 상당 퍼센트, 바람직하게는 부합하지 않는 파트의 전부를 단일 테스트 공정으로 식별하기에 충분할 정도로 효율적이어야 한다.
- <20> 비휘발성 메모리가 더욱 커지고 집적도가 더 높아지며 복잡하게 됨에 따라, 테스터는 메모리의 테스트에 소요되는 시간을 현저하게 증가시키지 않고서 증가된 크기나 복잡성을 처리할 수 있어야 한다. 메모리 테스터는 흔히 지속적으로 동작하며, 테스트 시간은 최종 파트의 비용에 있어 주요 요소로 고려된다. 메모리가 발전 및 향상됨에 따라, 테스터는 소자에 대해 이루어진 변화를 용이하게 수용할 수 있어야 한다. 비휘발성 메모리의 테스트에 특징되는 다른 사항은 메모리 셀에의 반복 기입이 파트의 전체적인 수명 성능을 저하시킬 수 있다는 점이다. 비휘발성 메모리 제조업체는 메모리 장치에 대해 특수한 테스트 모드를 구축함으로써 다수의 테스트 사항에 응답하고 있다. 이들 테스트 모드는 메모리 구입업체에 의해서는 전혀 사용되지 않지만, 가능한 한 적은 시간으로 그리고 가능한 한 효율적으로 메모리의 전부분 또는 주요 부분을 테스트하기 위해 제조업체에 의해서는 다루어질 것이다. 일부 비휘발성 메모리는 테스트 공정 동안 수리될 수도 있다. 따라서,

테스터는 수리의 필요성, 수리 지점, 필요한 수리의 유형을 식별할 수 있어야 하고, 적절한 수리를 수행할 수 있어야 한다. 이러한 수리 공정은 메모리의 특정한 부합하지 않는 파트를 검출하여 고립시킬 수 있는 테스터를 필요로 한다. 수리 기능뿐만 아니라 특수한 테스트 모드의 전체적인 장점을 취하기 위해서는 소자로부터의 응답에 기초하여 조건부 분기(conditional branching)를 지원하는 테스트 프로그램을 테스터가 실행할 수 있도록 하는 것이 좋다.

<21> 개념적 견지에서, 메모리를 테스트하는 공정은 알고리즘적인 공정이 된다. 예를 들어, 통상적인 테스트는 메모리 셀에 0과 1을 기입하면서 순차적으로 증가 또는 감소하는 메모리 어드레스를 포함한다. 일반적으로 메모리 사이클 동안에 기입 또는 관독되는 1과 0의 집합(collection)을 "벡터"로 지칭하는 한편, "패턴"이라는 용어는 벡터의 시퀀스를 지칭한다. 체커보드(checkerboard)와 같은 메모리 공간에 패턴을 기입하는 단계와 1과 버터플라이 패턴(butterfly pattern)을 조금씩 이동시키는 단계를 포함하는 것이 테스트에는 통상적이다. 테스트 개발자는 알고리즘 구조의 도움으로 이들 패턴을 작성하기 위한 프로그램을 더욱 용이하고 효율적으로 발생할 수 있다. 알고리즘적으로 작성된 테스트 패턴은 또한 기대된 바대로 수행하지 않는 패턴의 일부분을 고립시키기 위한 논리적 방법을 디버그하여 사용하는 것이 더욱 용이하다. 프로그래밍 루프에서 반복되는 명령과 커맨드를 사용하여 알고리즘적으로 작성된 테스트 패턴은 테스트 메모리에서 더 적은 공간을 차지한다. 따라서, 메모리 테스터가 알고리즘적인 테스트 패턴 발생 성능을 갖도록 하는 것이 바람직하다.

<22> 정확한 신호 에지 배치와 검출 또한 비휘발성 메모리 테스터의 효율성에 있어 고려 사항이 된다. 중앙에서는 전반적으로 부합하지만 특정된 가장자리에서는 부합하지 않는 파트를 포착하기 위해, 비휘발성 메모리 테스터는 각각의 신호 에지를 다른 신호 에지에 대해 시간적으로 정확하게 배치할 수 있어야 한다. 또한, 신호 에지가 어느 시점에서 수신되는지를 정확하게 측정할 수 있어야 하는 것도 중요하다. 따라서, 비휘발성 메모리 테스터는 자극과 테스트하의 소자(메모리)로부터의 응답의 타이밍 및 배치에 대해 충분한 가요성과 제어를 가져야 한다.

<23> 메모리 테스터는 피테스트 소자(DUT : Device Under Test)에 인가되는 전송 벡터(자극)를 발생하고, 되돌아올 것으로 예측되는 벡터(응답)를 수신한다. 이들 벡터를 발생하는 알고리즘 로직은 일반적으로 벡터내의 특정 비트가 어떻게 DUT내의 특정 신호 패드에 주어지거나 특정 신호 패드로부터 얻어지는지에 대해서는 문제삼지 않고서도 상기와 같이 행할 수 있다. 이러한 레벨에서는 벡터 내 인접 비트가 DUT 상에서 물리적으로 인접한 신호로서 귀결(end up)될 것이 기정 사실인 양 여겨진다. 그러나 불행히도 그렇지 않다.

<24> 실제로, "개념적인 레벨"에 있는 벡터내의 비트와 DUT에서의 실제 신호간의 대응은 오히려 제멋대로인 경향이 있다. 이를 방지하기 위해 어떠한 조치도 행하지 않는다면, DUT와 접촉하기 위해 주변부로부터 하강하는 하나 이상의 프로브 와이어를 교차시키는 것이 필요할 것이다. 이러한 교차(crossing)는 대부분 바람직하지 않으며, 전송 벡터내의 비트 포지션(bit position)이 DUT에 가해지기 전에 전송 벡터내의 비트 포지션을 재배열하도록 전송 벡터의 경로에 맵핑 메카니즘(mapping mechanism)을 통합시켜 물리적인 접촉을 이루는 작업이 교차의 부담을 지지않도록 하는 것이 통상적이다. 대응적으로 벡터 수신에는 역 맵핑(reverse mapping) 메카니즘이 적용되고 그런 다음 이러한 벡터들이 고려된다. 이러한 방식으로, 알고리즘 벡터 생성 및 비교 메카니즘이 이 전체적인 이슈를 무시할 수 있게 된다. 이러한 맵퍼(mapper)와 역맵퍼가 수행할 수 있는 또 다른 예로써, 상이한 예의 동일 유형의 DUT가 동일 웨이퍼 상에 웨이퍼 상의 공간 낭비를 방지하기 위해 회전 또는 소정의 미러형 대칭성(mirrored symmetry)으로 놓여져 있는 경우를 고려해 보자. 이들 실행예(practice)는 벡터 비트 포지션(vector bit position)과 물리적 신호 위치간의 대응에 관한 효과를 갖지만, 이것은 적합한 맵핑과 역맵핑에 의해 가려질 수 있다. 이러한 상황에 요구된 맵핑과 역맵핑은 특정 DUT에 대해 한 번 식별되고, 그대로 유지되어 그 특정 DUT를 테스트하는 과정 동안 변경이 필요치 않다는 것을 알 수 있을 것이다.

<25> 앞에서 DUT가 수리될 수도 있다고 언급한 바 있다. 이것은 아직 웨이퍼의 일부분인 다이싱되지 않은(undiced) 메모리칩에 대해 간혹 실현된다. 회로 수준에서 이것이 실제로 어떻게 달성되는지에 관해서는 이러한 소자를 제조하는 사람들에 의해 잘 이해되는 것이므로, 본 명세서에서는 선택 가능하게 파괴할 수 있는 소정의 엘리먼트(element)가 이들 소자에 통합되어 이들 엘리먼트를 파괴함으로써 관련 회로의 내부 로직을 변경할 수 있다는 정도만으로 충분하다. 이 능력은 결함을 갖고 있는 회로를 대체하기 위한 대체 회로로 내부 신호를 라우팅하기 위해 사용된다. 이 성능은 새로운 파트를 구성하는데 요구될 수도 있는 시간과 노력을 덜 들이고서 수리가 이루어질 수 없다면 불량 파트를 단순히 폐기통(scrap barrel)에 버리는 것이 비용면에서 더욱 효율적이기 때문에 경제적으로 가치가 있지 않은 것일 수도 있다. 특히, 대량의 불량 파트에서 특정 장애를 이해하고 이들의 수리 방법에 대한 결정을 담당하는 공정에 기술 인력을 수반하는 것은 바람직하지 않다. 그 대신, 메모리 테스터내의 알고리즘적 메카니즘(프로그램)이 장애를 분석하여 수리를 시도하기 위해

개발될 수 있다. 수리된 부품은 그 자리에서 재테스트되어 폐기 여부가 결정될 수 있다.

- <26> 이러한 모드의 조작은 메모리 테스터의 설계에 대해 특정한 암시를 포함한다. 테스트는 적합한 것으로 간주되는 어떠한 속도(때때로 이것은 부품이 작동될 수 있는 최고 속도)에서도 수행되어야 한다. 장애의 실시간 검출은 장애에 대한 이해를 세밀히 구별하기 위해 플래그를 설정하고 실험 알고리즘을 변경하기 위해 사용될 수 있다. 즉, 적절한 작동을 검증하기 위해 수행된 테스트가 부품이 최초 장소에서 고장나는 이유를 찾는 데 최적화된 테스트기여서는 안될 것이다. 최종적으로, 메모리 테스터는 수리를 시도해야 하는지의 여부와 만일 그렇다면 수리를 하기 위해 어떠한 행동을 취해야 하는지를 결정하는 자동화된 분석(즉각적으로 행해지는 더 큰 테스트 공정의 결론으로 이루어진 간에)을 위해 테스트 데이터의 트레이스(즉, 사용 가능한 레코드)를 작성할 수 있어야 한다.
- <27> 통상적으로, 수리의 시도는 적어도 예비적인 테스트가 가능한 장애의 범위와 수를 밝혀낼 때까지는 연기된다. 이용 가능한 대체 회로의 수는 제한되며(이를테면, 1/2 다스(dozen) 또는 홀수 유도 손익 분석(odd-driven cost benefit analysis)에 의해 결정된 바와 같은 정도), 이용 가능하기 보다는 더 많은 도움을 필요로 하는 것으로 나타낼 수 있는 파트를 고정(fix)하려는 시도에는 제한이 없다. DUT의 테스트가 고속으로 불필요한 일시중지(pause)없이 수행되어야 한다면, 장애를 기술하는 트레이스를 작성하기 위해 사용된 테스터의 메모리가 DUT를 테스트하기 위해 사용된 속도와 동일한 속도로 작동해야 한다. 본 명세서에서 설명될 메모리 테스터에서, 메모리는 ECR(에러 캐치 RAM)로 지칭된다.
- <28> 동작시, ECR은 일반적으로 DUT에 가해지는 것과 동일한 어드레스로 어드레스되고, DUT의 가장 작은 데이터 워드 폭인 비트 단위의 데이터 워드 폭을 갖는다. 워드 폭은 2의 거듭제곱(8, 16, 32)에 따라 조정 가능하며, 이러한 조정 가능성(adjustability)이 어드레스가능성(addressability)에 있어서의 대응하는 역 변경(inverse change)에 의해 달성되어, 어드레스 가능한 위치의 수를 곱한 워드 폭은 몇몇 상수와 동일하게 된다.
- <29> DUT에 대한 테스트 채널(출력 워드내의 비트 또는 대상으로 하는 일부 다른 신호)이 예측된 결과를 비교하거나 비교를 실패할 때, ECR내의 그 어드레스에서의 대응 비트가 사용 협의에 따라 설정 또는 소거된다. 이렇게 조직된 바와 같이, ECR은 각각의 어드레스/채널 조합에 대해 다수 비트 값을 취하지 않으며, 대신에 테스트동안 그 조합이 얼마나 많이 액세스될 것인지에는 상관없이 각각의 이러한 조합에 대한 정보의 단일 비트의 가치를 저장할 수 있다. 테스트 방안에는 비트가 무엇을 의미하는지와 비트가 어떻게 유지되는지가 포함된다. 비트는 전체적인 다수 액세스 테스트에 대해 "결코 실패하지 않았다/적어도 한번은 실패하였다"의 이분법을 나타내거나, 더 앞서서의 테스트와 편차가 있는 경우에도 최종 액세스(즉, 테스트)의 결과만을 나타낼 것이다. 특정 어드레스/채널에 대한 장애에 대해 정량 정보(quantity information)가 요구되면, 일부 추가 자원(카운터)이 이를 기록하기 위해 할당되어야 한다.
- <30> 종래의 메모리 테스터는 ECR로서 SRAM을 사용하였다. SRAM은 하나의 단일화된 어드레스를 사용하여 액세스되며, 일의적으로 어드레스될 때 DRAM보다 빠르지만 현저히 더 비싸다. 덜 비싼 DRAM은 CAS(Column Address Strobe)를 가지고 어드레스된 "열"을 특정함으로써 후속된 RAS(Row Address Strobe)를 가지고 어드레스된 "행"의 긴 선충전(lengthy pre-charging)을 요구하도록 내부적으로 조직된다. DRAM은 행이 선충전되었을 때 추가의 어드레싱이 그 행을 따라 행에 한정될 수 있다면(즉, CAS의 추가 요구, 그러나 RAS는 없음), 충분할 정도로 빠르다. 그러나, 테스터 동작에 관한 이러한 알고리즘적 제한(DUT를 일의적으로 어드레스하는 능력을 방해하는)은 종종 받아들여질 수 없으며, 때때로 유용하다 하더라도 고속 ECR 동작을 제공하는 것으로 신뢰될 수 없다. DRAM을 사용함으로써 ECR의 크기가 증가될 수 있고 그 비용이 감소된다면 더 비싼 SRAM에 대해 공통적으로 기대되는 것과 동일한 비율에서 임의적 어드레싱으로 DRAM을 동작시킬 수 있는 경우 더 유리할 수 있다.
- <31> 시판 부품을 구입하는 소비자로서, 우리는 기존의 개별적인 DRAM 파트를 요구된 크기로 또는 더 빠르게 하는 방법을 갖고 있지 않다. 우리가 할 수 있는 것은 소정의 요구되는 SRAM 양만큼 DRAM을 이용하게되는 지점까지 더 많은 DRAM을 이용하는 것이다. SRAM이 DRAM보다 현저하게 더 비싸므로 이것은 매혹적이다. 멀티플렉싱이 꺼려지기는 하지만, n개 부분의 멀티플렉싱 방식은 사용시의 메모리 사업의 수에서 관련된 n배로 증가한다. 버스당 50 내지 60 핀에서, 10-웨이 멀티플렉서(ten-way multiplexer)는 요구된 물리적인 팬-아웃(fan-out)을 실현하기 위해 증명된 악몽(a certified nightmare)일 것이다. 더욱이, 모든 그 메모리를 파일(pile)로 쌓아올리고 에러 캐치 RAM으로 사용할 만큼 고속으로 메모리에 기입하는 방법을 찾는다면, 그 메모리를 다른 용도, 이를테면 랜덤 액세스 속도가 더 낮아야 되는 것으로 알려진 곳이나, 또는 파트에 고유하고 기본 어드레싱 모드가 열 어드레스에서의 변화로 한정될 것을 전제로 하는 단순한 방법을 사용하여 고속으로 관독

및 기록 모두가 가능하기를 희망하는 경우 이용하기 위해 용이하게 재구성할 수 있을 것이다.

발명이 이루고자 하는 기술적 과제

<32> 여러 캐치 RAM에서의 사용을 위해 DRAM 동작의 속도를 증가시키는 것의 문제점은 그 그룹내의 메모리의 상이한 बैं크를 위한 인터리빙 신호의 조합과 이들 그룹의 बैं크간의 멀티플렉싱에 의해 해소될 수 있다. 그룹에 대한 신호 트래픽을 위한 가용성 4-배 인터리빙 방식으로 결합된 4개 बैं크 각각의 3개의 그룹간의 3-웨이 멀티플렉싱은 12의 계수에 달하는 동작 속도에서의 증가를 발생하는 한편, 오직 3개의 메모리 버스만을 필요로 한다. 멀티플렉서에 대한 다음 그룹의 선택을 위한 라운드 로빈 방식(round robin strategy)은 간단하며, 각각의 그룹에 대한 인터리빙 메카니즘이 가장 최근에 할당된 태스크를 완료하는데 필요한 시간을 갖는다는 것을 보증한다. 그룹내의 다음 어드레스가 동일한 것인지, 인접하거나 거의 인접한 정도의 것인지 또는 그 그룹내에서 액세스된 이전의 어드레스로부터 먼 것인지의 여부에 상관없이, 그룹내의 모든 인터리브된 액세스는 बैं크내 소재지를 얻기 위한 시도에서 실시간 고속 어드레스 분석을 불필요하게 수행하기 보다는 라운드 로빈 선택에 의해 선택된 다음 बैं크(그 그룹내의)에 대해 수행된다. 이 구성에서, 12개 बैं크의 각각은 전체 이용 가능한 어드레스 공간을 나타내고, 어떠한 개별적인 기입 사이클도 12개 बैं크의 어느 하나의 액세스를 완료할 것이다. 테스트의 결론 단계에서, 대상으로 하는 임의의 어드레스의 히스토리 또는 어드레스의 집합이 전체 12개의 बैं크에 걸쳐 펼쳐질 것이므로, 12개 बैं크 모두가 DUT의 테스트 동안 어떠한 장애가 발생되었는지를 검토하기 위해 조사되어야 한다. 즉, 어드레스에서의 채널이 통과되었는지 아니면 실패되었는지를 판정하기 위해, 12개 बैं크의 각각에 대해 판독 동작을 수행하고(여기서, 각각의 बैं크는 어드레스에 의해 결정됨), 그와 같이 하여 생성된 비트의 12개의 집합의 의미를 평가할 필요가 있다. 특정 채널이 12개의 비트에 의해 표현된다(각각의 बैं크로부터의 1 비트와 그 बैं크에 대한 워드내에서의 그 비트 포지션은 채널에 의해 결정됨).

<33> 그러나, 장애 정보를 찾아내기 위해 모두 12개의 बैं크를 개별적으로(이를테면, 수동으로) 참고해야만 하는 것은 번거로울 것이므로, 어드레스에서의 ECR 판독 사이클 동안의 모두 12개 बैं크의 결과를 단일화된 결과로 자동적으로 "작성(compose)"(병합)하기 위해 유틸리티 메카니즘(utility mechanism)이 제공된다. 즉, 어드레스/채널 조합에서의 0는 비교의 장애를 나타내는 것으로 가정한다. 그런 다음, 작성된 결과의 i번째 비트는 12개 워드의 적어도 하나(12개 बैं크의 각각으로부터의 하나)가 0인 경우에만 0이다. 작성은 모든 채널에 대해 한번 발생하지만 한 어드레스씩에 기초하여 이루어진다. 필요하다면, 작성 결과는 추후의 참조를 위해 선택된 बैं크 또는 모든 बैं크에 동시적으로 저장될 수 있다. 대상으로 하는 전체 어드레스 범위(이를테면, 테스트되는 전체 어드레스 공간)에 걸쳐 이 동작을 수행하는 루프에 포함될 때, 이것은 후속 장애 분석 메카니즘에 현저한 편리를 제공한다. 작성된 데이터가 모든 बैं크에 저장되면, 임의적 순서로 최고 속도에서 재생될 수 있다. 작성된 결과의 무결성의 트랙을 유지하는데 도움을 주는 메카니즘도 존재하며, 그리하여 추가의 테스트가 발생하는 경우(즉, 12개의 बैं크 모두에 일치하여 동시적으로 가해지지 않는 기입 동작이 존재함), 다른 작성 단계가 필요로 된다(거의 확실하게)고 말할 수 있을 것이다. 이들 메카니즘은 기입되는 최고 어드레스와 최저 어드레스의 트랙, 분류된 플래그 및 ECR 어드레스 공간의 범위를 어느 프로그램이 "소유"하는지에 대한 운영 체제 레벨 정보를 유지하는 레지스터를 포함할 것이며, 그리하여 작성의 관리가 유연하게 이루어지고, 필요에 따라 최소 또는 포괄적으로 이루어질 것이다.

<34> ECR은 또한 4개의 메모리 세트르 분할되며, 이중 2개는 "내부" SRAM이고 나머지 2개는 "외부" DRAM이다. 이 메모리 모두는 메모리 테스터 내부에 있으며, "내부" 및 "외부"라는 표현은 집적의 레벨에 더 관련된다. SRAM은 ECR과 관련된 VLSI(초대규모 집적) 회로의 집적 파트인 반면, DRAM은 VLSI 재료에 인접 장착된 개별적인 패키지 파트이다. SRAM의 양은 상당히 작음(이를테면, 메모리 세트당 메가비트 정도) 반면, DRAM의 양은 상당하고 선택 가능하다(이를테면, 메모리 세트당 128 내지 1024 메가비트의 범위). SRAM 메모리 세트는 항상 존재하며, ROM인 DUT의 예측된 내용을 저장하는 것과 같은 어떠한 적합한 용도를 위해 사용될 수 있다. DRAM 메모리 세트는 실제적으로 옵션이며, 많은 다른 용도가 있지만 통상적으로 수리를 이끌어 내기 위한 후속 분석을 위한 트레이스를 작성하기 위해 사용된다. 테스터는 SRAM과 DRAM 메모리 세트가 사용될 수 있는 다른 용도에 관해 SRAM과 DRAM 메모리 세트간의 구별을 강요하지 않는다. 이들 구별은 대부분 크기 문제를 야기한다. SRAM 메모리 세트는 작음 반면, DRAM 메모리 세트는 크다. 테스트 프로그래밍을 작성하는 사람은 다양한 메모리 세트가 어떻게 사용되는지에 관한 결정을 한다. 그러나, SRAM이 이미 고속으로 랜덤하게 액세스 가능한 것이므로, 다수 बैं크를 위한 멀티플렉싱/인터리빙 메카니즘을 포함하지 않으며, 각각의 SRAM 메모리 세트는 단순히 단일 बैं크가 된다. 이와 같이, SRAM은 항상 작성되어 별도의 작성 메카니즘의 필요성이 없다.

<35> 4개 메모리 세트의 각각은 자산의 제어기를 가지며, 이들의 동작은 상이한 모드의 ECR 동작을 지원하도록 구

성할 수 있다. 이것의 일특징은 메모리 세트 제어기가 지원하는 메모리 관련 트랜잭션(transaction)의 타입에 관련한다. 메모리 세트 제어기의 메모리측 상에서 개별적인 메모리 사이클은 본래 "관독" 또는 "기입"으로 분류된다. 그러나, 그 시스템측 상에서 메모리 세트 제어기는 소정의 상이한 스타일의 메모리 트랜잭션을 인정한다. 메모리 트랜잭션은 다음을 포함한다; (A) 상이한 시간에 기입된 데이터에 대해 "스틱키 제로(sticky zero)"를 실행하는 오버레이 기입 동작. 어드레스에서의 임의의 비트 포지션이 이 모드 동안의 어떤 시간에 기입된 0을 갖는다면, 그 어드레스에서의 그 비트 포지션은 0의 기입에 후속하여 그 어드레스에서의 그 비트 포지션에 1의 기입이 이루어지는 경우에도 관독시에 0을 발생할 것이다. (B) 공급된 데이터에 의한 어드레스된 데이터의 엄격한 대체(즉, 정규의 기입)인 오버라이트 기입. (C) 메모리 세트가 외부 메모리라면 모든 뱅크에 동일 데이터를 기입하는 시스템 기입. (D) 메모리 세트가 외부 메모리라면 모든 뱅크로부터 작성하는 분석 관독. (E) 메모리 세트가 외부 메모리라면 선택된 뱅크로부터 데이터를 관독하는 버퍼 메모리 관독. 이들 스타일의 메모리 트랜잭션은 각각 내부 SRAM과 외부 DRAM 모두에 실행 가능하다. 유일한 실제 차이는 달성하는데 얼마나 오래 걸리느냐는 것이며, 상기 설명은 "모든 뱅크" 또는 "선택된 뱅크"를 인용하고 타겟 메모리 세트가 내부 SRAM이라면 그 타겟 메모리 세트가 "모든" 뱅크와 "선택된" 뱅크 모두에 동시에 해당되는 한 뱅크의 메모리라는 인식이다. 그러므로, 모든 메모리 세트가 소프트웨어를 사용함으로써 등가적으로 처리(여하튼, 원칙적으로)될 수 있지만, 그들이 SRAM인 경우에도 다양한 메모리 세트 제어기의 내부 동작에서의 일부 차이를 예측하는 이유가 존재한다는 것이 명백하다.

<36> DRAM 메모리 세트와 관련된 일부 추가적인 구성 특성이 있다. 외부 DRAM 메모리 세트에 대해, 전술된 멀티플렉싱과 인터리빙 모드는 최대 100Mhz의 속도에서 풀 랜덤 액세스를 허용한다. 속도가 33Mhz를 초과하지 않을 것으로 알려져 있다면, ECR의 외부 DRAM 메모리 세트의 내부 동작은 더 낮은 속도에 대한 보상으로 3배의 깊이(depth)를 제공하도록 구성될 수 있다. 이것은 하나의 더 큰 그룹에 관한 인터리빙을 선호하여 그룹들간의 멀티플렉싱을 제거함으로써 달성되며, 멀티플렉싱의 일부로서 사용되던 뱅크 인에이블 비트는 남아있는 하나의 그룹의 어드레스 공간의 크기를 증가시키기 위해 정규 어드레스 비트로서 사용될 수 있다. 최종적으로, DUT에 대한 테스트가 "선형" 모드의 액세스(하나의 RAS, 다수의 CAS)를 적합화시킨다면, 테스트가 동작할 수 있는 최고 속도로 DUT가 테스트될 때에도 메모리 깊이(memory depth)에서의 12배 증가가 이용 가능하다. 이것은 동시의 단일 뱅크내의 어드레스를 선호하여 인터리빙 방식을 제거하며, 선형 어드레스를 가지고 사용될 때의 DRAM의 특정 성질 때문에 가능하다.

<37> 허용되는 또다른 가요성 재구성은 다른(즉, 속도 관련된) 모드의 동작에 상관없이 결합되지 않은 세트의 깊이의 2배를 갖는 하나의 메모리 세트로 외부 DRAM 메모리 세트를 결합하는 것이다. 이것은 또한 내부 SRAM 메모리 세트에 대해서도 행해질 것이다.

발명의 구성 및 작용

<38> 도 1을 참조하면, 본 발명의 원리에 따라 구성된 비휘발성 메모리 테스트 시스템의 개략적 블록도(1)가 도시되어 있다. 특히, 도시된 시스템은 36개까지의 개별적 DUT의, 64개의 많은 테스트 포인트(test points) 각각에서, 동시에 테스트할 수 있으며, 64개보다 더 많은 테스트 포인트를 가지는 DUT를 테스트하도록 테스트 리소스 집합의 요소들이 서로 분당되도록 하는 재구성을 제공한다. 이들 테스트 포인트는 아직 다이싱(diced)되지 않고 패키징(packaged)되지 않은 집적 회로 웨이퍼 상의 일부 위치가 있을 수 있고, 또는 패키징된 부품의 편일 수도 있다. "테스트 포인트"라는 용어는 신호(예컨대, 전원, 클럭, 데이터 입력)가 인가되거나 신호(예컨대, 데이터 출력)가 측정될 수 있는 전기적 위치를 지칭한다. 테스트 포인트를 "채널(channel)"이라고 지칭하는 업계의 관행을 따르기로 한다. 앞서 언급된 "서로 분당될 테스트 리소스의 모음"이란 36개만큼의 테스트 사이트라고 이해될 수 있는데, 각 테스트 사이트는 테스트 사이트 제어기(Test Site Controller)(4), (64개 채널) DUT 테스터(6) 및 DUT(14)에 대하여 실제 전기적 연결을 이루는 (64개 채널) 핀 전자 장치(Pin Electronics) 모음을 포함한다. DUT 테스트가 64개 또는 그 이하의 채널을 요구하는 경우에, 하나의 단일 테스트 사이트가 그 DUT에 대하여 테스트를 수행하기에 충분하며, 말하자면, 예컨대, 테스트 사이트(#1)(도 1에 도시된 바에 따르면)는 "단일 사이트 테스트 스테이션"으로서 형성되고 동작한다. 한편, 전술된 소정 형태의 재구성이 유효한 경우에는, 두 개(또는 그 이상의) 테스트 사이트가 서로 분당되어 128개의 채널을 가지는 하나의 더 큰 등가의 테스트 사이트로서 기능한다. 따라서, 도 1에 도시된 예를 다시 참조하면, 이른바 테스트 사이트(#35, #36)는 "2-사이트 테스트 스테이션"을 형성한다.

<39> 반대의 경우를 간단히 생각해보면, 전체 테스트 사이트가 단일 DUT를 테스트하여야 한다거나, 단일 테스트 사

이트가 단지 하나의 단일 DUT만을 테스트할 수 있다고 생각해서는 안 된다. 웨이퍼가 두 개의(아마도, 그러나 반드시 그러해야 하는 것은 아니며, 그 정도의) 다이(die)를 가지며, 그 테스트 채널 요청의 합이 64개의 채널 또는 그 이하라고 가정한다. 양쪽 DUT 모두는 단일 테스트 사이트에 의하여 테스트될 수 있다. 이를 가능하게 하는 것은 각 테스트 사이트의 범용 프로그램 가능성(general purpose programmability)이다. 테스트 사이트에 의하여 실행되는 테스트 프로그램은 테스트 사이트 리소스 중 일부가 DUT중 하나를 테스트하는데 이용되는 동안 또 다른 부분은 다른 DUT를 테스트하는데 이용되도록 작성될 수 있다. 결국, 처음 두 개의 논리적 유니언(union)인 제 3 DUT를 가진다면, 단일 테스트 사이트로 그 제 3 DUT를 테스트할 수 있다고 할 것이며, 따라서 소위 그 "구성요소 DUT(component DUTs)"를 유사하게 테스트할 수 있어야 한다. 유일한 차이는, "제 3" DUT의 경우 통합된 답이 도출되는 것과 달리 두 개의 "구성요소 DUT"가 합격(pass)인지 불합격(fail)인지를 개별적으로 추적한다는 점이다(즉, 제 3 DUT의 어떠한 부분이 불합격인지에 관한 문제가 있다). 이러한 "단일-사이트 다수-테스트 스테이션" 능력은 종래 기술에 가깝지만, 명료성을 위하여 또한 두 개 이상의 테스트 사이트를 서로 분당하는 개념과 비교하는 경우 있을 수 있는 잠재적 혼란 및 오해를 피하기 위하여 본 명세서에서 이를 거론하는 것이다.

<40> 이러한 재구성의 개념이 없다면 테스트 사이트와 테스트 스테이션간에는 어떠한 차이도 없을 것이고, 그 용어들 중 하나는 없어도 될 것이다. 그러나, 사실, 테스트 스테이션의 수가 테스트 사이트의 수와 동일할 필요는 없다는 것을 쉽게 알게 될 것이다. 과거에는, 테스트 사이트가 분할되어 더 많은 테스트 스테이션을 생성하였기 때문에(DUT는 전체 테스트 사이트를 사용할 만큼 복잡하지 않았다) 그 수가 서로 달랐을 수 있다. 그러나, 이제 그 차이는 테스트 사이트가 서로 분당되어 다수-사이트 테스트 스테이션을 형성하는(DUT는 단일 테스트 사이트로 프로세싱하기에 너무 복잡하다) 테스트 사이트에 또한 기인할 수 있다.

<41> 그런 다음, 이어서, 테스트 시스템 제어기(2)는 시스템 버스(3)에 의하여 36개만큼의 테스트 사이트 제어기(그 이름이 접미사(#1) 내지 접미사(#36)로 끝남)(4a-4z)로 연결된다. (a-z까지의 접자는 26개뿐이며 36개까지가 가능하지 않다는 것은 사실이다. 그러나 이러한 불일치에도 불구하고 숫자로 된 접자를 붙이는 것(잠재적으로 매우 복잡할 수 있음)보다는 숫자 참조 문자를 사용하는 것이 바람직한 것으로 보인다.) 테스트 시스템 제어기(2)는 비휘발성 메모리를 테스트하는 작업을 포함하는 적합한 테스트 시스템 제어 프로그램을 실행하는 컴퓨터(예컨대, NT를 실행하는 PC)이다. 테스트 시스템 제어 프로그램은 바람직한 테스트를 달성하기 위하여 작업 (및 복잡도)을 계층적으로 분할하였을 때 최상위 레벨에 존재하는 추상화 혹은 축약화(abstraction)를 나타낸다. 테스트 시스템 제어기는 어떠한 프로그램이 서로 다른 테스트 사이트에서 수행되고 있는지 뿐만 아니라 어떠한 프로그램이 필요에 따라 DUT와 테스트 프로브를 바꾸는 로봇 시스템(도시되지 않음)을 검사하는지를 결정한다. 테스트 시스템 제어기(2)는 어떤 테스트 사이트가 단일-사이트 테스트 스테이션으로서 수행하도록 프로그램되고, 반면 다른 것들은 서로 분당되어 다수-사이트 테스트 스테이션을 형성한다는 개념을 지지하는 방식으로 기능할 수 있다. 분명히, 그러한 환경에서는, 테스트되어야 하는 상이한 부품이 있으며, 상이한 부품에 대하여는 상이한 테스트가 수행되는 것이 가장 바람직하다. 유사하게, 모든 단일-사이트 테스트 스테이션이 동일한 유형의 부품을 테스트하도록 요구하는 경우는 없으며, 다수-사이트 테스트 스테이션에 대하여도 그러한 요구를 하는 경우는 없다. 따라서, 테스트 시스템 제어기(2)는 필요한 테스트 사이트 분당을 달성하는 명령을 발행하고 그런 다음 다양한 테스트 스테이션에서 이용될 적절한 테스트 프로그램을 호출하도록 프로그램된다. 테스트 시스템 제어기(2)는 또한 그 테스트로부터 얻어진 결과에 관한 정보를 수신하여, 잘못된 부품을 폐기하기 위하여 적절한 행동을 취할 수 있으며 소위 공장 설정에 있어서의 생산 프로세스를 제어하는데 이용될 수 있는 다양한 분석에 관한 기록(log)을 유지할 수 있다.

<42> 테스트 시스템 그 자체는 상당히 크고 복잡한 시스템이며, 핀 전자장치(Pin Electronics)(9)에 연결된 프로브 하에 하나 이상의 미래 다이(웨이퍼가 아직 다이싱되지 않음)를 순차적으로 배치하는 스테이지에서 웨이퍼를 적재하는데 로봇 서브시스템을 이용하는 것이 일반적이며, 여기에서 그 미래 다이(웨이퍼는 아직 다이싱되지 않음)가 테스트된다. 테스트 시스템은 또한 적절한 캐리어 상에 적재된 패키지 부품을 테스트하는데 이용될 수도 있다. 얼마나 많은 테스트 사이트가 테스트 스테이션 형성에 이용되는지, 또는 얼마나 많은 테스트 스테이션이 테스트 사이트 상에 있는지에 관계없이, 각 테스트 스테이션에 관련되어 이용되는 적어도 하나의 테스트 사이트 제어기가(이하에서 설명되는 바와 같이) 있을 것이다. 테스트 사이트 제어기는 내장형 시스템으로서 VOS(VersaTest O/S)라 불리는 전용 오퍼레이팅 시스템을 실행하는 36 내지 64MB의 프로그램 및 데이터 결합형 메모리를 구비한 Intel i960 프로세서일 수 있으며, 이는 비휘발성 메모리를 테스트하기 위해 이전의 상품에도 이용되었다(예컨대, Agilent V1300 또는 V3300). 우선은, 단일-사이트 테스트 스테이션의 상황만을 고려한다. 명확한 예를 위하여, 테스트 사이트(#1)는 테스트 스테이션(#1)으로 기능하여, WHIZCO 부품 no.0013을 테스트한다고 가정한다. 테스트 형태는 백 개 정도의 다양한 유형의 테스트(전압 레벨(voltage

level), 펄스 폭(pulse width), 에지 위치(edge position), 지연(delay), 그리고 대규모 단순 저장 및 선택된 정보 패턴의 검출을 변화시키고 모니터링함)에 관련되며, 각 유형의 테스트는 DUT마다 수백만의 개별적 메모리 사이클과 관련된다. 최상위 레벨에서, 테스트 시스템의 오퍼레이터는 테스트 시스템 제어기(2)에 대하여 테스트 스테이션(#1)을 이용하여 WHIZCO 0013을 테스트하기 시작하도록 지시한다. 결국 테스트 시스템 제어기(2)가 테스트 사이트 제어기(#1)(4a)(내장형 [컴퓨터] 시스템임)에 대하여 관련된 프로그램, 즉 TEST_WHIZ_13을 실행하도록 지시한다. 테스트 사이트 제어기(#1) 환경에서 이미 프로그램이 이용 가능하다면, 단순히 그것을 실행한다. 그렇지 않다면, 테스트 시스템 제어기(2)에 의하여 프로그램이 공급된다.

<43> 원리상, 프로그램 TEST_WHIZ_13은 완전 자립형(self-contained)일 수 있다. 그러나 만약 그렇다면, 상당히 클 것이 거의 확실할 것이며, 테스트 사이트 제어기(4a)내의 내장형 시스템의 프로세서가 바람직한 속도로 테스트를 실시하기에 충분히 빨리, 심지어 하나의 DUT 메모리 사이클로부터 다음 사이클까지 일정한 속도로 실행하도록 하는 것은 매우 어려울 수 있다. 어드레스 시퀀스 및 기록되어질 또는 관독 동작으로부터 예상되는 어드레스 시퀀스 및 관련 데이터를 생성하는 하위 레벨의 서브루틴 유형 동작은 필요에 따라서 DUT 테스트(6)내에 위치한 프로그램 가능 알고리즘 메카니즘에 의해 발생되지만, 이는 테스트 사이트 제어기(4)내의 내장형 시스템에 의해서 실행되는 프로그램과 동기하여(in synchrony with) 동작한다. 이것을, 소정의 하위 레벨 서브루틴 유사 동작(subroutine-like activity) 및 DUT 메모리 사이클을 개시하는 작업을 DUT(14)의 하드웨어 환경에 보다 가까운 메카니즘(DUT 테스터)으로 전달(export)하는 것으로 생각하기 바란다. 일반적으로 말해서, 테스트 시스템 제어기(2)는 테스트 사이트 제어기에 대하여 테스트 프로그램을 갖추도록 하는 경우마다, 테스트 사이트 제어기용 프로그래밍이 서술(describe) 혹은 필요로 하는 전체 동작을 달성하는데 필요한 적절한 하위 레벨 구현 루틴(아마도 테스트될 메모리에 특정한 루틴)을 관련 DUT 테스터에게 공급한다. 하위 레벨 구현 루틴은 "패턴"이라는 용어로서, 일반적으로 이름 붙여진다(상위 레벨 프로그래밍 언어에서 함수 및 변수가 이름을 가지는 것과 같이).

<44> 각각의 테스트 사이트 제어기(#n)(4)는 사이트 테스트 버스(#n)(5)에 의하여 그 관련된 DUT 테스터(#n)(6)로 연결된다. 테스터 사이트 제어기는 사이트 테스트 버스를 이용하여 DUT 테스터의 동작을 제어하고 그로부터 테스트 결과에 관한 정보를 수신한다. DUT 테스터는 테스트 형태에 관련된 다양한 DUT 메모리 사이클을 고속으로 생성할 수 있으며, "관독" 메모리 사이클의 결과가 예상되는바와 같은 지를 관정한다. 사실, 관독 및 기록 DUT 메모리 사이클의 대응하는 유용한 시퀀스를 개시함으로써 테스트 사이트 제어기로부터 전송된 명령 또는 작동 명령어("패턴이라고 명명됨")에 응답한다(즉, 대응하는 패턴을 실행한다). 개념적으로, DUT 테스터(6)의 출력은 DUT로 인가될 자극 정보이며, 또한 그로부터 응답 정보를 수용한다. 이러한 자극/응답 정보(7a)는 DUT 테스터(6a) 및 핀 전자 장치(#1) 어셈블리(9a) 사이를 지난다. 핀 전자 장치 어셈블리(9a)는 DUT(14)로 인가될 수 있는 64개까지의 프로브를 지원한다.

<45> 진술한 자극 정보는 DUT 테스터에서 이용되는 소정의 논리 장치군의 전압 레벨에 따라서 표현되는 병렬 비트 패턴의 시퀀스(즉, "송신 벡터" 및 예상되는 "수신 벡터"의 시퀀스)이다. 자극/응답내의 비트 위치와 다이상의 프로브 사이에 구성 가능한 맵핑(configurable mapping)이 있으며, 이러한 맵핑은 DUT 테스터(6)에 의하여 이해된다. 개별적 비트는 그 타이밍 및 에지 위치에 관하여 올바르지만, 맵핑에 더하여 DUT로 적용될 수 있을 때까지 전압 레벨 시프팅을 필요로 할 수 있다. 이와 유사하게, 자극에 후속하여 DUT에서 발생하는 응답은 DUT 테스터로 다시 입력되기 전에 버퍼링 및 (반전)레벨 시프팅을 할 필요가 있을 수 있다. 이런 레벨 시프팅 작업은 핀 전자 장치(9a)의 범주이다. WHIZCO 0013을 테스트하기 위하여 요구되는 핀 전자 장치 구성은 ACMIE사로부터의 부품을 테스트하는데 유효하지 못할 것이고, 아마도 심지어 다른 WHIZ사 부품에 대하여도 그러하지 못할 것이다. 그러므로, 핀 전자 장치 어셈블리가 역시 구성 가능할 것(그러한 구성 가능성은 PE Config 라인(8a)의 기능임)을 요구한다는 점이 이해될 것이다.

<46> 이상으로 단일 테스트 사이트가 DUT 테스트를 위하여 어떻게 구성되는지에 관한 간단한 구조적 개요를 설명하였다. 이제 작동할 많은 테스트 사이트가 있는 경우 일어날 수 있는 문제에 관하여 논의한다. 준비 단계로서, 다수의 테스트 사이트를 가지는 테스트 시스템을 구성하는 바람직한 실시예를 설명할 것이다. 많은 측면에 있어서, 이제 설명하고자 하는 정보 중 어떤 것은 고객 선호 및 비용 편익 분석의 시장 조사를 기초로 하는 선택의 문제이다. 어찌되었건, 이들 중 하나를 구성하기 위해 일정한 선택을 하여야하며, 일단 선택이 이루어지면 전체 시스템을 통하여 명백한 특정 결과가 발생한다. 적어도 일반적으로는, 테스트 시스템의 하드웨어 특성에 관한 더 많은 개요를 설명하는 것이 유용할 것이다. 이러한 특성들 중 일부는 우연에 의한 것이라고 할지라도, 그럼에도 불구하고 이러한 특성들을 아는 것이 본 발명을 설명하는데 이용되는 다양한 예들을

이해하도록 도울 것이다.

- <47> 네 개의 다소 큰 카드 케이지(card cages)를 고려하는 것으로 시작한다. 각 카드 케이지는, 전원(power supplies) 및 워터 쿨링(water cooling)(청정실 환경에서 팬(fan)은 오염원이 될 수 있음)에 더하여, 본체 기판(mother board), 프론트 플레인(front plane) 및 백 플레인(back plane)을 가진다. 각 카드 케이지에 대하여 9개까지의 어셈블리가 배치될 수 있다. 각각의 어셈블리는 테스트 사이트 제어기, DUT 테스터 및 핀 전자 장치를 포함한다. 테스트 사이트 제어기가 서로 본딩되는 방법의 일반적 개요를 설명할 것이며, 이는 데이지 체인(daisy chain)을 생성하는데 이용되는 몇몇 버스를 연루시킬 것이다.
- <48> 본론에서 벗어나 "데이지 체인"이란 용어에 관하여 간단히 설명하겠다. 시스템 요소 A, B, C 및 D를 생각해 보자. 이들이 그러한 순서로 데이지 체인되어야 한다고 가정한다. A를 떠나 B로 가는 정보 및 제어 경로가 있고, 그런 다음 B는 B를 떠나 C로 가도록 트래픽을 선택적으로 전달할 수 있으며, 그런 다음 C는 C를 떠나 D로 가도록 트래픽을 선택적으로 전달할 수 있다고 말할 수 있다. 이러한 동일한 종류의 장치는 반대쪽 방향으로의 트래픽에 관하여도 존재할 수 있다. 데이지 체인은 종종 우선 순위 방안을 생성하기 위하여 이용된다. 여기서는 다양한 테스트 사이트 제어기들 사이에 마스터/슬레이브 관계를 생성하는데 이들을 이용할 것이다. 이들 데이지 체인 유형의 통신 장치는 "BUS" 대신에 접미사 "DSY"로 표시할 것이다. 그러므로, 명령어/데이터 버스 대신에 명령어/데이터 DSY라고 말할 것이다. 정보가 "B로 입력되어 선택적으로 전달된다"는 개념은 트래픽이 전달되기 전에 별도의 도전체 세트 상으로 복제되어야 한다는 것을 암시할 수 있다. 그러한 방법은, 성능 이유가 아니라면 이는 어드레스 가능한 엔티티를 가지는 정규 버스와 더욱 유사한 것일 수 있다. 프로그램 가능 어드레스 맵핑 장치 및 다운 스트림 테스트 사이트 제어기의 일부가 "수면(to sleep)" 상태에 있도록 하는 능력에 의하여, 단일 버스는 논리적으로 다수의 데이지 체인처럼 보이게(즉, 그렇게 기능하게) 구성될 수 있다. 결국, 데이지 체인은 명령어 및 제어 정보에 대한 고성능 경로라고 할 수 있고, 만약 그렇지 않다면 마스터/슬레이브 조합(다수-사이트 테스트 스테이션)이 단일 테스트 사이트만큼 빨리 작동한다고 예상할 수 없다. 데이지 체인 성능의 이점을 위하여, 다양한 DSY는 그 각각의 카드 케이지를 벗어나지 않는다. 이러한 결정의 효과는 어떠한 테스트 사이트가 서로 본딩될 수 있는지(또한 얼마나 많은 테스트 사이트가 서로 본딩될 수 있는지) 어떤 한계를 설정하게 한다. 원리상, 이러한 한계가 반드시 요구되는 것은 아니며, 또한 (그러한 경우에) 관련된 것이 기술적으로 실현 가능성이 없는 것도 아니다. 다만, 카드 케이지에 이미 9개의 테스트 사이트가 있으므로, DSY를 확장하는 것은 상대적으로 적은 이점에 비하여 너무나 많은 부가 비용을 추가하는 것처럼 보인다.
- <49> 도 1의 논의를 다시 시작하여, 네 개의 카드 케이지에 있는 다양한 테스트 사이트 제어기(4a-4z)를 고려한다(각각의 카드 케이지는 9개의 테스트 사이트 제어기를 구비함). 이들을 참조 번호(4a-4f), 참조 번호(4g-4m), 참조 번호(4n-4t) 및 참조 번호(4u-4z)라고 표시한다. (이전에 설명했던 바와 같이, 이들은 명목상 26개의 첨자뿐임에 - 독자는 또 다른 10개의 첨자 심볼이 그곳 어딘가에 있을 것이라고 생각하기 바란다 - 주의할 필요는 없다.) CMD/DAT DSY(17a)(명령어 및 데이터 데이지 체인)는 하나의 카드 케이지에 있는 테스트 사이트 제어기(4a-4f)를 상호 연결하는 반면, 다른 CMD/DAT DSY(17b)는 또 다른 카드 케이지에서 테스트 사이트 제어기(4g-4m)를 상호 연결한다. 남아있는 카드 케이지에 대하여도 각각 동일한 장치, 테스트 사이트 제어기(4n-4t) 및 테스트 사이트 제어기(4u-4z)가 존재한다. 앞서 DSY는 카드 케이지를 벗어나지 않으며, DSY를 실제로 형성하는 버스의 "테일 엔드(tail end)"가 카드 케이지를 벗어나지 않고 또 다른 카드 케이지에서 다음 세그먼트의 헤드(head)가 된다고 설명한바 있다. 대신에, 테스트 시스템 제어기(2)로부터의 시스템 버스(3)는 테스트 사이트 제어기 전체로 연결되며, 각각은 카드 케이지를 벗어나지 않는 DSY 세그먼트의 헤드에서 마스터가 될 수 있다.
- <50> 지금까지 논의해 온 CMD/DAT DSY(17a-17d)는 다양한 테스트 사이트 제어기(4a-4z) 사이에 존재한다. SYNC/ERR DSY(18a-18d) 및 DYT 테스터(6a-6z)에 대하여도 유사한 장치가 있다. SYNC/ERR DSY(18)에 의하여 운반된 동기화 및 에러 정보는 DUT 테스터가 일치하여 작동할 수 있게 해준다. 이들 두 개의 데이지 체인(17,18)은 약간씩 다른 유형의 정보를 전송하지만, 각각은 하나 이상의 테스트 사이트를 함께 하나의 테스트 스테이션으로 본딩하는 동일한 일반 메카니즘의 일부로서 존재한다.
- <51> 이제 도 2를 참조하면, 도 1의 DUT 테스터의 단순화한 확장 블록도가 도시되어(36개까지 있을 수 있음) 있다. 그 하나의 경우만을 설명하는 것으로 충분하다. 도 2를 언뜻 보면, 상당히 조밀하다고 느낄 것이다. DUT 테스터(6)에서 블록도에 도시되고 있는 일부는 기능적으로 상당히 복잡하고, "오프 더 션elf(off the shelf)" 형태로는 이용가능하지 않다. 여기서 두 가지 포인트를 지적하는 것이 중요하다. 첫 번째, 도면에 도 2를 포함한 주요 목적은 전체적 비휘발성 메모리 테스트 시스템(1)내의 중요한 작동 환경의 기본 특성을 설명하는

것이다. 도 3 및 후속 도면들에 의하여 충분히 설명되는 본 발명은 도 2의 다음 설명에서 시작되는 메카니즘의 확장이거나 또는 도 2로부터 동기(motivation)가 부여되는 새로운 메카니즘일 것이다. 어느 쪽이나, 본 명세서를 작성함에 있어 이들 중 어떤 것이 독자 앞에 있는지는 정확히 알지 못한다. 현재의 목표는 차후의 여러 가지 바람직한 실시예의 방대한 상세한 설명을 위한 단순화되고 정보적인 시작 포인트를 제공하여, 차후의 설명 각각이 적절히 간결해지도록 하는 것이다(각각의 서로 다른 발명에 관하여 모든 것을 설명하는 하나의 "점보" 명세서가 되지 않도록). 두 번째는, 확대된 또는 확장된 요소가 일반적으로 전체적으로는 도 2와 일치하지만 도 2의 단순화 버전과 정확하게 "매치(match-up)"되지 않는 정보를 포함할 수 있다는 것이다. 이는 예러가 있다는 것을 의미하거나 또는 그것들이 치명적으로 일치하지 않는다는 것을 의미하는 것은 아니며, 이는 때때로 무엇인가를 단순화하여 미니어처로 그 정확한 이미지를 나타내는 것이 곤란하거나 불가능하기 때문에 발생하는 것이다. 이러한 상황은 지도와 유사하다. 표준 사이즈의 콜로라도 지도는 I-70상에서 동쪽으로 가는 경우 덴버에서 I-25를 따라 북쪽으로 갈 수 있다는 것을 보여줄 것이다. 이는 좌회전처럼 보인다. 그리고 이는 과거에는 실제로 좌회전이었으나 지금은 그렇지 않고, 그 교차로에 대한 상세한 지도는 일련의 회전과 사이에 있는 사거리를 보여줄 것이다. 그러나 누구도 표준 사이즈 지도가 잘못되었다고 말할 수 없으며, 그 추상화 혹은 축약화의 레벨에 있어서는 올바른 것이다. 유사하게, 그리고 그 상당히 복잡한 외관에도 불구하고, 도 2는 실제로 중간 레벨의 추상화 혹은 축약화로 동작하는 간이한 것이지만, 일부 외관상 좌회전이 전혀 단순한 좌회전이 아닌 경우도 있다.

<52> 도 1에 도시된 바와 같이, DUT 테스트(6)로의 주 입력은 테스트 사이트 버스(5)가 한 예로, 이 테스트 사이트 버스(5)는 관심 DUT 테스트(6)의 예와 연관된 테스트 사이트 제어기(4)로부터 시발된다. 테스트 사이트 버스(5)는 특별한 목적의 마이크로프로세서에 유사할 수 있는 마이크로-제어기 시퀀서(19)와 결합한다. 테스트 사이트 버스(5)는 마이크로-제어기 시퀀서(19) 내부(PGM SRAM(20)) 혹은 마이크로-제어기 시퀀서(19) 외부(EXT. DRAM(21)) 중 하나일 수 있는 프로그램 메모리에 저장된 프로그램으로부터의 명령을 페치(fetch)한다. 이러한 두 개의 메모리가 본질적으로 프로그램 카운터로 역할하는 논리 공통 어드레스(63)(혹은 명령어 페치 어드레스)인 것에 의해 어드레싱되고 두 개의 메모리 중 어느 하나가 수행될 프로그래밍의 소스일 수 있지만, (1)메모리 중 오직 하나만이 임의의 시간 주기 동안에 명령어 페치 메모리 사이클을 수행하고 (2)실제로 두 개의 메모리가 전기적으로 상이한 신호에 의해 어드레싱된다는 점에 유의해야 한다. SRAM은 고속이고 진정한 랜덤 액세스를 허용하지만, 마이크로-시퀀스 제어기(19)(대형 IC인)내의 유용한 공간을 소모하여, SRAM의 크기는 제한된다. 외부 DRAM은 대용량의 조절가능한 양으로 제공될 수 있지만, 선행 수행을 포함하고 어떠한 브랜칭도 없는 순차 청크(chunk)로 액세스될 때에만 고속이다. 집중 알고리즘인 SRAM(20) 프로그래밍이 가장 흔하지만, EXT. DRAM(21)은 초기화 루틴 및 랜덤 혹은 비정규 데이터와 같이 알고리즘 프로세스에 의해 쉽게 생성되지 않는 요소에 가장 적합하다.

<53> 마이크로-제어기 시퀀서(19)에 의해 수행되는 명령어 워드는 상당히 길다, 즉 208비트이다. 명령어 워드는 13개의 16비트 필드로 구성된다. 이러한 필드는 흔히 정규 마이크로-제어기 시퀀서 외부의 메카니즘에 관한 페치 명령 정보를 나타낸다. 이런 필드는 이와 연관된 메카니즘 전용이다. 한 세트의 ALU 명령(22)은 8개의 16비트 ALU(24)의 콜렉션에 제공되고, 다른 세트 ALU 명령은 DUT 테스트 도처에 분포된 다양한 다른 메카니즘에 분배된다. "다양한 제어값 & 명령"(42) 범례(legend) 및 라인이 이런 후자의 경우를 표시한다.

<54> 8개의 16 비트 ALU(24) 각각은 연관된 16비트 결과 레지스터 주변에 구축된 전형적인 레퍼토리의 산술 명령을 갖는다(각각의 ALU는 또한 여러 다른 레지스터를 갖는다). 이러한 결과 레지스터 중 3개의 레지스터 및 3개의 레지스터와 연관된 ALU는 DUT에 제공될 완결 어드레스로 다양하게 결합되는 X, Y 및 Z 어드레스 성분(27)을 생성하기 위한 것이다. 8개의 ALU/레지스터 중 2개(DH & DL) 이상이 최대 유효부(most significant portion : DH)와 최소 유효부(least significant portion : DL) 사이에서 분할되는 32비트 데이터 패턴(28)의 알고리즘 생성을 지원하는데 제공된다. 마지막 3개의 ALU/레지스터(A, B, C)는 카운터로서 사용되고 어떤 프로그램 지정 횟수의 반복 혹은 다른 횟수 조건에서 완결하기 위해 프로그램 제어 및 브랜칭으로 지원하는 다양한 프로그램 제어 플래그(25)의 생성에 기여한다. 이러한 프로그램 제어 플래그(25)는 마이크로-제어기 시퀀서(19)에 되전송되는데, 마이크로-제어기 시퀀서(19)에서 플래그(25)는 마이크로프로세서에서와 같이 명령어 페치 어드레스의 값에 영향을 미친다. 또한 프로그램 브랜칭에 영향을 미치는데 사용될 수 있는 다양한 기타 플래그(55)가 존재한다. 이러한 플래그(55)는 페치 명령어 워드의 상이한 필드에 의해 제어되는 DUT 테스트(6)내의 다양한 기타 메카니즘과 관련된다. 하나의 특정한 추가 플래그가 분리 아이템, 즉 VEC_FIFO_FULL(26)로 명시적으로 도시되어 있다. 다소 덜 세부적인 사항을 갖는 다른 도면에서, 이 하나의 특정한 추가 플래그는 기타 플래그(55)와 함께 총괄될 수 있다. 본 명세서에서는 하나의 특정한 추가 플래그를 분리하여 마이크로-제어기 시퀀서(19) 동작의 한 측면의 설명을 용이하게 하고자 한다.

<55> VEC_FIFO_FULL이 수행하는 것은 마이크로-제어기 시퀀서(19)에 의한 추가 프로그램 수행을 (일시) 중지시키는 것이다. 마이크로-제어기 시퀀서(19)에 의해 폐치되는 명령과 DUT에 적용될 테스트 벡터를 최종적으로 핸드 오프 하는 메카니즘 사이에 많은 단계의 파이프라인이 존재한다. 또한, DUT에 제공될 수화물(baggage) 일부가 앞으로 진행함에 따라서 벡터를 수반하는 수화물 일부는 궁극적인 벡터 애플리케이션의 속도 혹은 각각의 벡터 지속 시간에 관한 정보이다. 따라서, DUT로의 벡터 애플리케이션의 속도는 상수일 필요가 없으며, 특히 어떤 그룹의 벡터는 생성하는 것보다 제공하는데 더 오래 걸릴 수 있다. 마이크로-제어기 시퀀서는 단지 자신의 최대 속도로 프로그래밍을 수행한다. 그러나 명백하게, 평균적으로 "벡터 소모"의 속도는 파이프라인이 거의 제한 없이 탄성체처럼 들쭉 날쭉 할 필요가 없도록 "벡터 생성"의 속도와 동일해야 한다. 후술할 어드레스 맵퍼(29)의 출력에 벡터 FIFO(45)가 존재하고, 벡터 FIFO(45)는 파이프라인에서 탄성체 기능을 수행한다. VEC_FIFO_FULL 신호는 파이프의 헤드 엔드(head end)에서 새로운 벡터의 생성을 일시 중지시켜 파이프라인에서 제한된 수의 단계를 초과하는 것을 방지하는데 사용된다.

<56> 계속하여, (16비트의 3배인 48비트의) X, Y 및 Z 어드레스 성분(27)은 어드레스 맵퍼(29)에 제공되는데, 어드레스 맵퍼(29)의 출력은 순서화된 48비트 어드레스 공간에서 거의 임의 재구성된 어드레스 값으로 사전 선택된다. 이것을 인식하는 출발점으로서, 어드레스 맵퍼(29)가 완전히 찬 48비트 어드레스 공간인 메모리이고 각각의 어드레스에서 48비트 값을 보유한다고 우선 가정하자(이런 메모리가 오늘날 대형 냉장고 크기가 될 것임을 잠시 고려하지 않기 바란다). 이런 메모리가 주어졌을 때, 탐색 테이블은 어느 제공 어드레스를 대체 어드레스로 사용될 수 있는 임의 선택된 다른 48비트 값에 맵핑하게 구현될 수 있다. 이런 어드레스 맵핑이 바람직한 이유는 X, Y 및 Z 어드레스 성분이 (일반적으로 하나의 큰 선행 디코더로는 구현되지 않을 가능성이 가장 높은 특정한 DUT 내부 아키텍처의 관점에서 유용한 의미를 갖는다는 점이다. 행, 열과 레이어, 블록 혹은 페이지의 개념은 테스트 엔지니어에게 매우 유용할 수 있고, 물리적으로 서로 가까운 위치에서 발생하는 오류는 그들의 X, Y 및 Z 어드레스에서 대응하는 유사성을 포함할 수 있다. 테스트 결과에서의 이런 패턴은 무엇이 오류인지를 인식하고 오류 섹션 동작을 예비 섹션 동작으로 회피하도록 일부를 재프로그래밍하는 생성 레벨에서 이런 혹은 설계 레벨에서 이런 오류를 정정하는데 유용할 수 있다. 이 점에서 두 가지 문제가 발생한다. 첫 번째는 48비트를 DUT에 제공될 실제수의 비트(32비트 혹은 16비트)로 줄이는 것이다. 본 발명자들은 줄이는 방법을 간략히 언급할 것인데, 줄이는 방법은 대개 X로부터 얼마의 비트, Y로부터 얼마의 비트 및 Z로부터 그 나머지를 취하는 문제이다. 진부는 아니지만, 이것은 두 번째 문제인데, 이는 소정 어드레스가 회로의 다른 섹션의 좌-우(left-for-right)(혹은 좌-우 및 상-하) 거울 이미지로 회로내에 놓여 있을 수 있기 때문이다. 이것은 어떤 순차 어드레스 값이 이 회로내에서 물리적 순서대로 존재하는 한 비트가 의미하는 것을 재구성하는 효과를 갖는다. 이런 칩 레이아웃 특성은 수차례 발생할 수 있고, 한 그룹의 비트, 즉 Y가 해석되는 방법이 어떤 다른, 즉 Z비트의 수반값에 의존할 수 있다는 것은 당연하다. 어드레스 맵퍼(29)는 원래 X, Y 및 Z 어드레스가 "재패킹"되게 제공되어 이런 내부 아키텍처 구성을 갖는 메모리를 테스트할 사람이 이런 유형의 일을 반영하게 한다. 이와 같은 것을 실제 수행하기 위해, 어드레스 맵퍼(29)는 상당수의 상호 접속 멀티플렉서를 포함한다. 어드레스 맵퍼(29)는, 설명을 위해 앞서 일시 가정한 바와 같이, 완전히 찬 메모리 디코드 기법(fully populated memory decode scheme)의 완전 임의 탐색 테이블 행동(completely arbitrary look-up table behavior)을 달성할 수는 없다. 그러나, 어드레스 맵퍼(29)는 필요한대로 X, Y 및 Z 어드레스 성분의 서브-필드를 재구성할 수 있는데, 이는 48비트를 필요한 실제수로 감소시킬 다른 메카니즘이 여전히 존재하기 때문이다. 어드레스 맵퍼(29)는 또한 3개의 16비트(어드레스) 탐색 테이블을 포함하여 로컬 범위내에서 제한된 임의 맵핑을 수행한다.

<57> 어드레스 맵퍼(29)의 맵핑 어드레스 출력(30)은 Aux RAM(31) 및 에러 캐치 RAM(32)에 어드레스로서 제공되는데, Aux RAM(31) 및 에러 캐치 RAM(32)은 별개 기능을 수행하지만 하나의 대형인 전체 RAM에서 선택가능한 부분으로 구현될 수 있다. 맵핑 어드레스 출력(30)은 또한 후술할 어드레스 비트 선택 회로(37)에 하나의 입력으로 제공된다.

<58> Aux RAM(31)을 고려해 보자. Aux RAM(31)의 기능은 DUT에 제공될 수 있는 데이터 패턴(33)과 어드레스(34)를 보유하는 것이다. 데이터 패턴(33)과 어드레스(34)는 Aux RAM(31)으로부터의 논리적으로 별개인 출력인데, 이는 데이터 패턴(33)과 어드레스(34)가 다소 상이하게 프로세싱되고 상이한 장소에서 사용되기 때문이다(Aux RAM(31)은 이중 "포트 메모리"가 아니지만, 출력이 멀티플렉서에 제공되는 여러 뱅크인 것이 바람직하다). 이와 같은 구현시, 저장 데이터(33)는 Aux RAM(31) 어드레스의 하나의 뱅크 혹은 범위에 보존되고, 저장 어드레스(34)는 Aux Ram(31)의 다른 뱅크 혹은 범위에 보존될 수 있다. 또한, 본 발명자들은 Aux RAM(31)에 기록하기 위한 명시적인 메카니즘을 도시하지는 않는다. 이것은 수행 프로그램 명령에서 테스트 사이트 제어기(4)가 개시하는 어드레싱 버스 동작에 의해 달성된다(도 2에서의 거의 모든 부분으로 향하는 "under the

floorboards", 즉 "링 버스"로 지칭되는 "유틸리티 서비스" 버스[도면을 매우 복잡하게 하기 때문에 도시하지 않음]가 존재한다).

<59> 에러 캐치 RAM(32)은 Aux RAM(31)에 제공되는 동일한 어드레스로 어드레싱되고, 에러 캐치 RAM(32)은 에러에 관한 정보를 저장하거나 혹은 검색하는데, 이 동작은 후술할 포스트 디코드 회로와 결합하여 수행된다. Aux RAM(31)으로부터의 경로(33 및 34)에서와 같이, 경로(61)(에러 캐치 RAM으로) 및 경로(62)(에러 캐치 RAM으로부터)는 링 버스(도시되지 않음)에 의해 분배된 구성 정보에 따라서 멀티-뱅크 메모리(에러 캐치 RAM(32))로부터 멀티플렉싱된 출력이 바람직하다.

<60> 데이터 멀티플렉서(35)는 ALU의 수집(24)에서의 레지스터(DH 및 DL)로부터의 데이터(28)뿐만 아니라 Aux RAM(31)으로부터의 저장 데이터 출력(33)을 입력으로 갖는다. 데이터 멀티플렉서(35)는 이러한 입력(28, 32) 중 어느 것을 자신의 출력(38)으로 제공할지를 선택하는데, 이 출력(38)은 전송 벡터 맵퍼/직렬변환기/수신 벡터 비교 데이터 회로(40)에 두 개의 벡터 성분 중 하나로서 제공된다(다른 성분은 어드레스 비트 선택 회로(37)의 출력(39)이다).

회로(40)는 세 가지 기능, 즉 벡터 성분(38, 39)을 DUT에 제공(전송)될 전체 벡터의 순서화된 논리 표현으로 어셈블링하고, 전송 벡터 논리 표현의 순서화된 비트와 이 신호(즉, 이 벡터내의 비트) 대신에 DUT와 접촉할 핀 전자 장치(즉, 어느 프로브 팁)의 실제 물리 채널 번호 사이에 임의 동적 대응(맵핑)을 적용하며, 컴파일러와 협력하여 전체 논리 벡터를 DUT(이를 허용하는 DUT)에 개별적으로 또한 순서대로 제공될 부분(pieces)으로 분할하는 기능을 수행할 수 있다. 이러한 기능 중 어느 기능이 수행될지는 마이크로-제어기 시퀀서(19)에 의해 폐치되는 208비트 명령어에서의 필드에 따라서 또한 어드레싱되는 SRAM(41)으로부터의 제어 신호에 의해 결정된다. 회로(40)의 출력은 VEC_FIFO_FULL 신호(26)를 완전히 생성하는 벡터 FIFO(45)에 제공될 64비트 벡터(44)에 달하는데, VEC_FIFO_FULL 신호(26)의 의미와 이용은 전술하였다. 벡터 FIFO(45)의 상위 벡터는 주기 생성기(49)(간략히 설명될)에서 발생하는 VEC_FIFO_UNLOAD 신호(47)의 수신시 벡터 FIFO(45)에서 제거된다. 이 제거 벡터(46)는 핀 전자 장치(9)의 연관 동작을 통해 DUT에 연결되는 타이밍/포매팅 및 비교 회로(52)에 제공된다. 즉, 핀 전자 장치(9)의 각각의 동작은 핀 전자 장치(9)와 연관된 타이밍/포매팅 및 비교 회로(52)로부터 전송 & 수신 벡터(7) 및 핀 전자 장치 구성 정보(8)를 수신한다.

<61> 타이밍/포매팅 및 비교 회로(52)는 마이크로-제어기 시퀀서(19)의 프로그램 SRAM(20)에서와 같이 동일한 명령어 어드레스("원내부의 A")에 의해 어드레싱되는 내부 SRAM(54)을 갖는다(외부 DRAM(53)이 내부 SRAM(54) 대신에 사용될 수 있다). 내부 SRAM(54)(혹은 외부 DRAM(53))은 구동 및 비교 사이클의 생성을 지원한다. 구동 사이클은 전송 벡터를 DUT에 제공한다. 비교 사이클은 DUT가 제공하는 벡터를 수신하고 이를 검사하여 이전에 제공된 비교 데이터와 매칭하는지를 판정한다. 구동 사이클 및 비교 사이클 모두는 그들의 지속 시간, 로드가 인가되는지 여부 및 언제 인가되는지 데이터가 언제 래칭 혹은 스트로빙되는지에 관해 조절될 수 있다. 전술한 비교는 수신 벡터 역맵퍼/직렬 변환기(57)에 제공되는 64비트 값(56)을 생성하는데, 이 변환기(57)의 기능은 회로(40)의 논리적인 반전으로 생각하면 된다(회로(57)의 동작은 SRAM(41)에 의한 회로(40)의 제어에 대응하여 SRAM(58)에 의해 제어된다). 이어서, 회로(57)의 출력(59)은 포스트 디코드 회로(60)에 제공된다. 포스트 디코드 회로(60)는 프로그램 표준을 통해 입력 에러 정보(59) 및 (이전의) 저장 에러 정보(60)(에러 캐치 RAM에 저장된) 모두를 검사하여 후에 경로(61)를 통해 에러 캐치 RAM(32)에 되저장될 압축되고 쉽게 해석가능한 에러 정보를 생성할 수 있다. 한 가지 예로 에러가 몇 차례나 특정한 범위의 어드레스내에 존재하는지에 관한 정보가 생성될 수 있는데, 이 정보는 대체 회로를 구동하여 온-칩 수리 시도할 때를 결정하는데 유용할 수 있다.

<62> 이제 주기 생성기(49) 및 이와 연관된 타이밍 SRAM(51)을 설명하겠다. 주기 생성기(49)와 타이밍 SRAM(51)은 마이크로-제어기 시퀀서(19)에 의해 폐치되는 각 208비트 명령어에 대해 타이밍/포매팅 및 비교 회로(52)의 연관 동작에 관한 지속 시간을 결정하는 8비트 신호 T_SEL(43)에 응답한다. T_SEL(43)은 폐치 명령어내의 상이한 필드에 의해 표시되는 다양한 제어값 및 명령(42)의 멤버이다. T_SEL(43)은 8비트 값으로 256 상이한 경우를 표시하거나 혹은 인코딩할 수 있다. 이 경우에, 이러한 "경우들"은 타이밍 SRAM(51)에 저장된 28비트 값이고 T_SEL에 의해 어드레싱된다. 각각의 어드레싱된 28비트 값(23)은 19.5 피코초 해상도를 갖는 원하는 지속 시간을 지정한다. 액세스된 28비트 지속 시간 값(23)의 시퀀스는 이 시퀀스의 개별 멤버가 벡터 FIFO(45)에 저장된 목적 대응 벡터의 검색과 동시에 검색되고 제공될 수 있도록 주기 FIFO(50)에 저장된다.

<63> FIFO(50)에서 최초 엔트리에서의 거친 타이밍 값 필드는 5 nsec의 해상도를 갖는 지속 시간 정보를 운반하고 이 정보로부터 벡터 FIFO(45)로부터의 후속 전송 벡터를 타이밍/포매팅 및 비교 회로(52)에 전송하는 VEC_FIFO_UNLOAD 신호(47)를 생성한다. 비교 신호 타이밍 리마인더(48)가 또한 회로(52)에 제공되는데, 이

회로(52)에서 최종 19.5 피코초 해상도가 달성된다.

- <64> 도 3은 도 2의 블록도내의 ECR(32)의 간략화된 블록도(64)를 도시하고 있다. EXR은 다양한 어드레스 분류기(77, 78, 79)에 인가되는 어드레스 맵퍼(29)로부터의 48비트 맵핑된 어드레스(30)를 수신한다. 어드레스 분류기는 메모리 세트(73-76)와 관련되고, 이들 메모리 세트는 각각 관련 ECR 기능을 개별적으로 수행할 수 있는 완전한 메모리 메카니즘이다. 이들 메모리 세트 중의 2개(73, 74)가 외부 DRAM으로 구성되는 반면, 나머지 2개는 내부 SRAM으로 구성된다. 2개의 외부 DRAM 메모리 세트는 항상 효과면에서 동일한 어드레스 분류 기능을 가질 것이며, 하나의 공통 어드레스 분류기(77)를 공유한다. 내부 SRAM 메모리 세트(75, 76)는 각각 자신의 관련된 어드레스 분류기(78, 79)를 갖는다. 이들 어드레스 분류기는 통상적인 것일 것이며, 종래 기술에 널리 알려진 용도 및 원리에 따라 어드레스를 변경시킬 수 있다. 이들은 완벽성을 위해 그리고 본 출원과 앞으로의 관련 출원간의 호환성을 촉진하기 위해 도시되었다. 어드레스 분류기가 유용한 기능을 수행하지만, 어드레스에 대한 변경을 수행하지 않는 것으로 단순히 가정함으로써 무시될 것이다.
- <65> 각각의 메모리 세트는 메모리 세트 제어기를 포함하며, 외부 DRAM 메모리 세트(73, 74)는 각각 DRAM 메모리 세트 제어기(65, 66)를 갖는 한편, 내부 SRAM 메모리 세트(75, 76)는 각각 SRAM 메모리 세트 제어기(67, 68)를 갖는다. DUT의 테스트 동안, 이들 메모리 세트의 임의의 세트로 전송된 메모리 트랜잭션을 위한 어드레스는 각기 관련된 어드레스 분류기로부터 관련 메모리 세트 제어기에 도달한다. DUT의 테스트 동안, 포스트 디코드 회로(60)로부터 도달하고 ECR내에 기입될 오류 데이터(61)가 먼저 데이터 분류기(80-83)에 인가되며, 그 중의 하나가 각각의 메모리 세트와 관련된다. 데이터 분류기의 기능은 현재의 관심 대상이 아니며, 완벽성을 위해 그리고 본 출원과 차후 기대되는 관련 출원간의 호환성을 촉진하기 위해 원칙적으로 도시한 것이다. 어드레스 분류기에 관해, 데이터 분류기(80-83)는 데이터를 수정없이 단순히 통과시키도록 함으로써 무시될 것이다. 어드레스 및 데이터 분류기는 각각 어드레스 및 데이터를 위한 고속 경로를 제공하고, 필요한 최고 속도로 동작할 것이다. 링 버스(도시 생략)가 어드레스와 데이터를 메모리 세트로 전송하기 위한 또다른 방식을 제공하는 것에 대해 간략하게 살펴볼 것이다.
- <66> 이 시점에서, 각각 유입 어드레스와 데이터를 갖는 4개의 메모리 세트 제어기(65-68)가 있다. 이들 메모리 세트 제어기의 각각은 관련 메모리에 접속되며, DRAM 메모리 세트 제어기(73, 74)는 각각 외부 DRAM(69, 70)에 접속되는 한편, SRAM 메모리 세트 제어기(75, 76)는 각각 내부 SRAM(71, 72)에 접속된다. 이들 배열은 4개의 메모리 세트(73-76)를 구성하며, 이들 중의 2개(75, 76)는 적당한 양의 고속 SRAM을 갖고, 나머지 2개(73, 74)는 대량의 보다 느린 DRAM을 갖는다. 현재 주요 관심 사항은 DRAM 메모리 세트를 어떻게 SRAM 메모리 세트만큼 빠르게 할 수 있느냐 하는 것과 사용자 기호 및 테스트 프로그램 전략에 의거하여 SRAM의 구성에 관한 특정 대안을 어떻게 통합하느냐 하는 것이다. 그러므로, DRAM 메모리 세트 제어기(65, 66)가 구성 가능하고, 상이한 유형의 메모리 트랜잭션을 수행하며, 더 단순한 SRAM 메모리 세트 제어기(67, 68)와 완전히 동일하지 않다는 것을 설명할 것이다. 간략화를 위해, 도 3은 이 가요성을 제공하는 구조를 도시하지 않으며, 각각의 메모리 세트 제어기가 링 버스(도시 생략)에 접속되어 요구되는 특정 모드의 동작과 구성에 대해 이 버스로부터 명령을 받는다. 이들 모드 중의 일부는 데이터가 어떻게 저장되는지를 수반하며, 일부는 데이터를 꺼내어 다시 되돌리는 처리를 해야 한다. DRAM 메모리 세트의 모드와 구성에만 원칙적으로 관심을 가질 것이다. 결론을 내리기 위해, 각각의 메모리 세트가 추가의 처리를 위해 포스트 디코드 메카니즘(60)에 보내지는 관련된 Data Out(62A-62D)를 갖는다는 점에 유의하기 바란다.
- <67> 도 4에는 도 3과 관련하여 설명되는 ECR(32)의 더욱 상세한 블록도(84)가 도시되어 있다. 이 도면은 거의 동일하며, 유사 구성요소는 공통 참조 부호가 부여되어 있다. 현재의 목적을 위해, 도 4에서 나타나는 차이를 밝히는 것으로 충분하다. 특히, 어드레스 분류기(78-79)의 각각은 관련 MUX(85-87)가 선행되어 있다는 점에 유의하기 바란다. 이들 MUX는 어드레스 구별의 처리에 도움을 주고, 특히 48비트를 32비트로 하향된 어드레스의 크기를 얻는데 도움을 준다. 즉, 이들 MUX에 관한 상황은 어드레스 분류기 및 데이터 분류기에 대한 상황과 유사하다는 것이 이해될 것이며, MUX에 대해 현재 언급되지 않은 유용한 이유로 이 곳에 MUX가 존재한다면, 이들 MUX를 주로 완벽성을 위한 것(및 어드레스를 32비트로 하향하기 위한 것)으로 보아야 할 것이다. 더욱이, 링 버스(85)가 메모리 세트 제어기(65-68)의 각각에 접속되는 것에 유의하여야 할 것이다. 메모리 세트 제어기로부터의 Data Out(62A-62D)은 포스트 디코드 회로(60)에 도달할 때 4:1 MUX(94)에 인가되며, 이 MUX는 링 버스에 의해 설정된 제어 레지스터(95)에 따라 추가 처리를 위해 어느 출력이 선택되는지를 결정한다.
- <68> 도 4에서의 주요 관심은 다양한 메모리 세트 제어기(65-68)에 의해 제어된 메모리의 총체적인 조직을 처리하여야 한다는 것이다. 메모리 세트2(75) 및 메모리 세트3(76)의 경우, 그 메모리는 단순히 단일 어드레스 공

간(그 메모리 세트내의)으로서 배치되고 종래의 방식으로 동작하는 SRAM이다. 그러나, 메모리 세트0(73) 및 메모리 세트1(74)의 경우, 각각을 위한 메모리는 각각 4개 뱅크의 3 그룹이며, 어드레스를 위한 포맷은 이들 메모리 세트에 현재 적용 가능한 모드 및 구성 정보에 좌우된다.

<69> 그래서, 예를 들어, 메모리 세트0(73)는 3개의 그룹(88, 89, 90)을 갖는 한편, 메모리 세트1(74)는 그룹(91, 92, 93)을 갖는다. 랜덤 어드레싱을 위한 하나의 고속 모드에서, 연속되는 메모리 트랜잭션은 자동적으로 상이한 그룹에 보내지고(멀티플렉싱), 이들의 각각은 어드레스 및 데이터를 위한 자신의 하드웨어 경로를 갖는다. 이들 그룹의 각각은 4개의 뱅크(어드레스 공간의 4개의 예)로 구성되며, 이들을 위한 메모리 동작은 본 기술 분야에서 이해된 원리 및 실시예에 따라 인터리브될 수 있다. 특히, 현 상황에서의 사용을 위해 바람직한 DRAM의 유형은 SDRAM이며, 이것은 인터리빙을 위한 특정한 방식을 암시한다. 아래에 설명되는 것 외에도 다른 유형의 DRAM이 존재하고 인터리빙을 위해 다른 메카니즘이 가능하다는 점을 이해할 수 있을 것이다. 더 느린 랜덤 어드레싱을 위한 또다른 모드에서, 그룹은 자동적으로 선택되는 대신 어드레스 가능하다. 이러한 더 느린 모드에서, 그룹을 위한 여분의 어드레스 비트가 하드웨어 경로를 선택하기 위해 사용된다. 이 모드에서, 어드레스는 그룹과 인터리브된 뱅크의 그 그룹내의 뱅크 어드레스를 특정한다. 어드레싱을 수행하는 또다른 고속 모드에서, 멀티플렉싱과 인터리빙 모두 턴오프되고, 어드레스는 그룹 선택 비트, 뱅크 선택 비트 및 뱅크내 어드레스 비트(within-Bank address bits)를 갖는다. 협폭 워드 모드의 동작에서, 메모리 트랜잭션의 타겟인 전체 워드내에서 필드를 특정하기 위해 추가의 어드레싱 비트가 사용된다.

<70> 도 5에는 도 3 및 도 4에 도시된 DRAM 메모리 세트 제어기(65, 66)의 간략화된 블록도(96)가 도시되어 있다. DRAM 메모리 세트 제어기는 분류된 어드레스(CLASSIFIED ADDRESS)(106)와, 링 버스(85)로부터의 모드 및 구성 정보와, 관련 데이터 분류기로부터의 ERROR DATA IN(105)을 입력으로서 수신한다. 앞에서와 같이, DRAM 메모리 세트 제어기는 DATA OUT(62A/B)을 발생한다.

<71> 그러나, 어드레스 및 데이터의 다른 소스가 링 버스(85)를 통해 이들 데이터와 어드레스를 제공하는 것을 알 수 있다. 즉, 링 버스를 DRAM 메모리 세트 제어기에 접속시키는 버스 인터페이스(97)가 존재하고, 그 인터페이스를 통해 링 버스로부터의 데이터(DATA FROM RING BUS)(99) 및 링 버스로부터의 어드레스(ADDRESS FROM RING BUS)(100)가 이용 가능하게 된다. MUX(104)는 ERROR DATA IN(105) 또는 DATA FROM RING BUS(99) 중의 어느 것이 데이터(107)로서 전방으로 전송되어 마스터 DRAM 제어기(109)의 DATA IN 단자에 인가될지를 선택한다. 마찬가지로, MUX(103)는 분류된 어드레스(106)와 링 버스로부터의 어드레스(100)간에 선택하여 마스터 DRAM 제어기의 어드레스 단자에 공급되는 어드레스(108)를 발생한다. 그 내용이 링 버스 상의 트래픽에 의해 설정되는 하나 이상의 레지스터(98)의 집합은 각각 MUX(103, 104)에 의해 이루어질 선택을 나타내는 제어 신호(101, 102)를 발생한다.

<72> DATA OUT(62A/B)이 버스 인터페이스(97)에 추가로 인가되어 DATA OUT이 링 버스를 통해 보내지도록 하는 것을 이해할 것이다.

<73> DRAM 메모리 세트 제어기의 기본적인 기능은 3개의 그룹 중에서의 다양한 메모리 트랜잭션의 할당 또는 분배이다. 고속 모드의 동작에서, DRAM 메모리 세트 제어기는 1:3 MUX(125)(의 등가물)를 사용하는 라운드 로빈 방식으로 이 할당을 수행한다. MUX(125)는 점선으로 도시되어 있으며, 그 이유는 그 곳에 MUX가 실제로 존재될 수 있지만 본 바람직한 실시예에서는 그 위치에 실제 MUX가 존재하지 않는다는 것이 명백해질 것이기 때문이다. 그 대신, 그리고 도 6과 관련하여 명백하게 되는 바와 같이, 복잡한 규칙 추종 메카니즘(스테이트 머신)의 제어 하에 어드레스 가능한 데이터 소스의 다수의 예가 존재한다.

<74> 도 5에 대한 설명을 완료하기 위해, 1:3 MUX(125)가 3개의 슬레이브 SDRAM 제어기(110-112)를 분할하고 그룹 0, 그룹1 및 그룹3의 각각에 대해 하나의 이러한 슬레이브 SDRAM 제어기가 존재한다는 것에 유의하기 바란다. 각각의 SDRAM 슬레이브 제어기는 그 그룹으로서 SDRAM의 4개 뱅크의 집합을 갖는다. 예를 들어, 그룹0에 대한 SDRAM 제어기(110)는 뱅크(113, 114, 115, 116)에 접속된다. 유사한 방식으로, 그룹1은 뱅크(117-120)를 갖는 한편, 그룹2는 뱅크(121-124)를 갖는다. 결과는 각각의 DRAM 메모리 세트에 대해 총 12개의 뱅크가 있다.

<75> 각각의 그룹의 SDRAM은 여러 모드 또는 구성으로 동작하도록 배열될 수 있다. 고속에서의 랜덤 어드레싱 동작을 위해 구성될 때, 그룹간의 멀티플렉싱은 최고 레이트로 발생하고, 연속적인 메모리 동작은 항상 그리고 자동적으로 그 사이클 시퀀스로 다음 그룹으로 전송된다. 그룹 내에서, 메모리 연산은 4개의 뱅크 중에서 이들을 균등하게 분배하기 위해 인터리브된다. 여기서는 정규 사이클 시퀀스가 바람직하다. 인터리빙은 속도에서 4배 증가를 발생하며, 멀티플렉싱에 의해 제공된 3배 증가와 결합될 때 12의 계수만큼 속도에서의 증가

가 이루어진다. 이 방법의 동작은 각각의 뱅크를 전체 어드레스 공간으로서 취급하고, 12개의 뱅크 중의 어느 것이 특정 메모리 트랜잭션을 위한 타겟이 되는 시간보다 먼저 제어를 시도하지 않는다. 즉, 저장될 데이터는 12개의 뱅크 중의 임의의 한 뱅크에서 종료할 것이고, 메모리로부터의 단순한 판독은 12개 뱅크 중의 임의의 한 뱅크로부터 내용을 복구할 것이다. 한 뱅크에서의 어드레스의 내용이 또다른 뱅크에서의 그 동일 어드레스의 내용과 동일하다라고 기대할 이유가 없다. 그러나, 이 시점에서 데이터가 DRAM의 단일 뱅크를 위한 레이트의 최소 10배의 레이트에서 저장될 수 있다는 것이 명백하다. 우리가 지불하는 가격은 당연히 어드레스에서 데이터 아웃을 판독하기 위해 모두 12개 뱅크에서 그 어드레스의 내용을 조사하는데 필요한 정도이다 (엄격히 말하면, 이것은 항상 사실이지는 않다. 오직 4개 뱅크의 내용이 조사되도록 요구되는 동작 모드가 존재한다. 이에 대해서는 아래에 간략하게 더 설명될 것이다).

<76> 이 시점에서, DRAM 메모리 세트가 지원할 수 있는 다양한 모드 및 구성의 일부에 대한 약간의 전문 용어와 간단한 설명을 설정할 수 있다. 이들은 다음과 같다.

<77> 랜덤 100 Mhz(R100)

<78> 깊이에 있어서 한 뱅크에 동일한 어드레스 공간에 대한 랜덤하게 어드레스된 기입을 허용하기 위해 4개의 인터리브된 뱅크의 각각의 3개의 멀티플렉싱된 그룹을 사용하는 전체 속도 기입 동작. 멀티플렉싱과 인터리빙 모두가 사용되고 있음. 데이터가 '다음' 그룹내의 '다음' 뱅크에 기입될 것이고, 유입 어드레스가 특정 그룹 또는 특정 뱅크를 식별하기 위한 비트를 갖지 않음.

<79> 랜덤 33 Mhz(R33)

<80> 깊이에 있어서 3개의 그룹에 동일한 어드레스 공간, 즉 3개의 뱅크 깊이에 상당하는 어드레스 공간에 대한 랜덤하게 어드레스된 기입을 허용하기 위해 4개의 인터리브된 뱅크의 어드레스된 그룹을 사용하는 감소된 속도 기입 동작. 인터리빙이 사용되고 멀티플렉싱은 사용되지 않음. 데이터가 어드레스된 그룹내의 '다음' 뱅크에 기입될 것이고, 어드레스는 그룹 선택 비트를 갖지만 뱅크 선택 비트는 갖지 않음. 그룹 선택 비트는 최상위 어드레스 비트이며, 그래서 그룹 경계를 제외하고는 연속 어드레스가 자동 인터리빙에 의해 결정된 뱅크에서만 연속 뱅크내 위치가 됨.

<81> 로컬라이즈드 100 Mhz(L100)

<82> 깊이에 있어서 12개의 뱅크와 동일한 어드레스 공간에 대한 최소 행 어드레스를 갖는 판독 및 기록 모두를 허용하기 위해 어드레스된 그룹내의 어드레스된 뱅크를 사용하는 전체 속도 동작. 유입 어드레스는 그룹 선택 비트, 뱅크 선택 비트 및 뱅크내 위치를 특정하는 어드레스 비트를 가짐. 멀티플렉싱과 인터리빙 모두 없으며, 뱅크내 어드레싱이 장소 조건(locality requirement)을 충족시킬 것임. 장소의 부족은 자동적으로 검출되며, 필요로되는 리어드레싱이 수행됨. 장소의 부족이 치명적이지는 않지만, 지속적으로 위반되면, 결과는 느린 성능으로 나타남.

<83> 스택 메모리 세트

<84> 임의의 전술한 모드에서, 어드레스 공간의 깊이를 2배로 제공하기 위해 메모리 세트0과 메모리 세트1이 결합될 수 있고, 메모리 세트2와 메모리 세트3도 마찬가지로. 유입 어드레스는 메모리 세트 선택 비트를 가짐.

<85> 협폭의 워드

<86> 임의의 전술한 모드에서, 메모리 세트가 2의 거듭제곱이면서 32 이하의 워드 폭을 갖도록 구성될 수 있음. 이러한 협폭의 워드는 어드레스된 워드의 전체 워드 폭 이내에서의 필드이고, 2의 거듭제곱 경계(power-of-two boundaries) 상에 위치되며, 어드레스된 워드내에 필드를 위치시키기 위해 여분의 어드레스 비트를 사용함. R100 및 R33과는 동작하지만 L100과는 동작하지 않음.

<87> 작성(Composition)

<88> 전술한 "랜덤" 액세스 모드에서, 판독을 위한 데이터를 작성할 때의 다수 뱅크를 판독하기 위한 필요성은 R33

에 대해서는 적합한 그룹의 4개의 뱅크로부터의 어드레스에서의 결과치를 그리고 R100에 대해서는 모두 12개 뱅크로부터의 어드레스에서의 결과치를 통합시키는 하드웨어 보조(hardware assist)에 의해 지원됨. 그 결과치를 모든 뱅크 또는 알려진 뱅크에 저장하면서 뱅크내 어드레스를 이동하는 루프에 작성 동작을 집어넣음으로써, 메모리의 전체 영역이 테스트 결과치의 분석 동안 더 빠른 액세스를 허용하도록 미리 작성될 수 있음.

<89> 작성 무결성(Composition Integrity)

<90> 메모리의 작성 영역에 대한 작성의 손실에 대한 검출이 하드웨어 보조를 가짐.

<91> 히든 리프레시(Hidden Refresh)

<92> 다양한 뱅크에서의 DRAM의 리프레시 동작은 전술된 모드 또는 구성의 어느 것에도 간섭없이 자동적으로 수행됨.

<93> 전술한 집합의 성능과 협력하여, 개별적인 메모리 트랜잭션은 다음 카테고리 중의 하나에 속하는 것으로서 설명될 수 있다.

<94> 오버라이트 기입(OWW)

<95> 한 어드레스에서의 1개, 4개 또는 12개 뱅크 모두(1개, 4개 또는 12개 뱅크의 각각)에 대한 엄격한 대체 기입: 이전의 내용은 상실됨. R100(12개 뱅크), R33(4개 뱅크) 및 L100(1개 뱅크)과 작업하지만, 랜덤 어드레싱을 위한 정격 속도 동작을 보존하지는 못함. 더 오래된 메모리 테스터를 위한 테스트 프로그램과의 호환성이 유지됨. 분류된 어드레스 및 분류된 데이터를 사용함.

<96> 오버레이 기입(OLW)

<97> 1개, 4개 또는 12개 뱅크의 '다음'에 대한 "스티키 제로"를 지원하는 판독-수정-기입 동작(read-modify-write operation). 비트 포지션에서의 0은 1로 오버라이트되지 않을 것이지만, 1은 0로 오버라이트될 수 있다. R100(12뱅크), R33(4개 뱅크) 및 L100(1개 뱅크)과 정격 속도로 작업함. 이것은 테스트 동안 데이터를 기입하기 위한 기본적인 방법이며, DUT 어드레스에서의 반복된 테스트를 통해 비트 포지션내의 장애의 포획(trapping)을 허용함. L100에서 데이터가 기입되지 않는다면(부담 조건(an onerous requirement)) 여전히 작성될 필요가 있을 것이므로, 작성과 혼동되지 않음. 분류된 데이터와 분류된 어드레스를 사용함.

<98> 시스템 기입(SYW)

<99> L100에서는 단일 뱅크에 기입함. R33에서는 어드레스된 그룹의 모두 4개의 뱅크에 그리고 R100에서는 모두 12개의 뱅크에 동일한 것을 기입하지만, 정격 속도를 보존하지 않음. 어드레스와 데이터를 위한 소스는 링 버스임.

<100> 시스템 판독(SYR)

<101> L100에서는 단일 뱅크로부터 판독함. R33에서는 어드레스된 그룹의 모두 4개의 뱅크로부터 그리고 R100에서는 모두 12개의 뱅크로부터 작성된 판독을 어드레스에서 수행하지만, 규정 속도를 보존하지는 않음. 어드레스 소스와 데이터 목적지는 링 버스임.

<102> 분석 판독(ANR)

<103> R33에서는 어드레스된 그룹의 모두 4개의 뱅크로부터 그리고 R100에서는 모두 12개의 뱅크로부터 작성된 판독을 어드레스에서 수행하지만, 규정 속도를 보존하지는 않음. 분류된 어드레스를 사용하고, 데이터를 경로(62A-62D)를 통해 포스트 디코드에 보냄.

- <104> 버퍼 메모리 판독(BMR)
- <105> 어느 뱅크가 '다음'이 되던지 간에 그 뱅크에 대한 랜덤 어드레스에서의 전체 속도(100MHz) 판독. L100(동일 뱅크가 항상 '다음'이 됨), R33(어드레스된 그룹내의 4개의 뱅크의 '다음 것') 및 R100(3개 그룹내의 4개 뱅크의 '다음 것')과 작업한다. 분류된 어드레스를 사용하고, 데이터를 경로(62A-62D)를 통해 포스트 디코드에 보냄.
- <106> 명백하게, 상기 내용의 일부는 더욱 상세히 하술될 멀티플렉싱 및 인터리빙 방식으로부터 발생한다. 멀티플렉싱 및 인터리빙 방식은 물론 DRAM 메모리 세트에 제한된다(SRAM 메모리 세트는 스타트가 빠르다). 그러나, 이것은 이들 동일한 능력 또는 동작 모드가 SRAM 메모리 세트에 의해 지원될 수 없다는 것을 의미하지는 않는다. 일반적으로, 한 메모리 세트에 가해질 수 있는 메모리 트랜잭션은 임의의 다른 메모리 세트에도 가해질 수 있으며, 단지 크기 제한만을 받는다. SRAM 메모리 세트는 DRAM 메모리 세트가 행할 임의의 스타일의 동작을 존중할 것이다. 메모리 세트 제어가 요구된 트랜잭션을 어떻게 내부적으로 실시하느냐 하는 것만이 상이하다. 예를 들어, 분석 판독(작성)의 경우, SRAM 메모리 세트는 그 데이터가 이미 첫번째 장소에서 작성되었기 때문에 단순한 판독을 행하는 것 이상으로 괴롭힐 필요가 없다.
- <107> 이러한 다양한 스타일의 메모리 트랜잭션은 테스트 프로그램내에서 필요에 따라 결합될 것이다. 예를 들어, 테스트 후, 루프가 특정 메모리 세트에 대한 어드레스 범위내의 모든 데이터를 작성할 수 있다. 그리고나서, 임의의 랜덤한 순서로 데이터에 도달하기 위해 BMR이 사용될 수 있다(고속으로). 이것은 모든 어드레스에서(작성된 범위내의) 각각의 뱅크가 동일 데이터를 갖기 때문에 이루어진다.
- <108> 실험에 의해 전술한 메모리 모드/구성 및 트랜잭션 모드가 상당한 양의 시간을 단축시킬 수 있는 것으로 판명되었다. 다음의 표 1은 단축 처리에 도움을 주기 위한 함축된 개요로써 제공된다. W 1/4라는 표기는 4개의 뱅크 중의 하나에 대한 기입을 나타내고, R은 판독을 나타낸다.

표 1

메모리 트랜잭션

소스/ 목적지	작동	이름	설명	모드: L100 R33 R100		
링 버스	시스템 기입 시스템 판독	SYW SYR	전체 (1, 4, 12) 뱅크 전체 (1, 4, 12) 뱅크	W 1/1 R 1/1	W 4/4 R 4/4	W 12/12 R 12/12
H/S ADDR.	패턴 기입 AND데이터 패턴 판독	OWW ANR	전체 (1, 4, 12) 뱅크 전체 (1, 4, 12) 뱅크	W 1/1 R 1/1	W 4/4 R 4/4	W 12/12 R 12/12
H/S ADDR.	패턴 기입 AND데이터 패턴 판독	OLW BMR	다음 (1, 4, 12) 뱅크 다음 (1, 4, 12) 뱅크	W 1/1 R 1/1	W 1/4 R 1/4	W 1/12 R 1/12

- <109>
- <110> 설명을 재개하기 위해, 도 5에 도시된 마스터 DRAM 제어기(109)의 블록도(126)가 도시되어 있는 도 6을 참조한다. 관련 메모리 세트 제어기로부터 인가된 기입 데이터(WRITE DATA)(107) 및 ADDRESS(108)는 각각의 FIFO(127, 128)에 접속된다. FIFO(127)의 출력단측 상에서는 상이한 그룹에 관련되는 추가의 FIFO(137, 139, 141)에게 WRITE DATA(131)가 인가된다. 이들의 출력단(166, 168, 170)은 각각 그룹 0 내지 그룹 2에 대한 실제 기입 데이터 버스이다. 유사한 형태로, FIFO(128)의 출력은 FIFO(138, 140, 142)에 인가되고, 이들 FIFO의 출력단은 상기한 그룹의 어드레스 버스(167, 169, 171)가 된다.
- <111> 마스터 DRAM 제어기(109)는 ADDRESS(132) 및 링 버스(85)에 접속되는 스테이트 머신(193)을 포함한다. 다른 것들 중에, 각종의 모드 제어 레지스터(130)가 요구된 모드 및 구성을 나타내도록 설정될 수 있다. 스테이트 머신(193)은 또한 실행시의 모드와 구성에 따라 어떤 그룹이 다음 메모리 트랜잭션을 수신할지에 대한 선택을 담당한다. 즉, 스테이트 머신은 유입 어드레스내의 그룹 선택 비트의 필드를 존중하거나 아니면 자동적으로 다음 그룹을 선택할 것이다. 그룹에 대한 메모리 동작을 발생하기 위해, 스테이트 머신은 적합한 그룹 사이클 제어(GROUP CYCLE CONTROL) 신호(133)를 발행한다. 이들 신호는 그 그룹에 대한 FIFO(그룹 0에 대해서는 143, 그룹 1에 대해서는 144, 그룹 2에 대해서는 145)에 래치되며, 그 그룹에 대한 메모리 사이클 제어 버스

(172, 173, 174) 를 통해 정해진 순서에 따라 래치로부터 출현한다. 스테이트 머신(193)을 ADDRESS(132)에 접속시키는 다른 이유는 ECR의 DRAM 메모리 세트에 대한 어드레싱 방식에 관한 아래의 설명 동안에 명백하게 될 것이다.

<112> 판독 동작은 그룹 0, 그룹 1 및 그룹 2에 대해 각각 그룹 판독 데이터(GROUP READ DATA)(134, 135, 136)를 발생한다. 이들 결과는 작성기 회로(146) 및 MUX(148)에 인가된다. MUX(148)는 개개의 그룹 판독 데이터 중의 하나와 그 어드레스에 대한 데이터의 작성된 버전(COMPOSED GROUP READ DATA(147)) 간에 선택한다. 작성 동작이 진행중이면, GROUP READ DATA 버스의 각각이 데이터의 4개의 워드를 제공할 필요가 있을 것이며, 이는 4개의 연속적인 READ에 대한 필요성(ADDRESS FIFO(138, 140, 142) 및 CYCLE CONTROL FIFO(143, 144, 145)를 수반함)을 암시한다는 점에 유의하기 바란다. 이러한 내부적 동작의 관리는 스테이트 머신(193)의 제어하에 이루어지고, 특정될 수 있는 다양한 메모리 트랜잭션의 성질에 응답하여 발생한다. 임의의 경우에, 선택된 데이터(모드 제어 레지스터(130)의 제어 하에 MUX(148)에 의해 선택된)는 그룹 판독 데이터(149)로서 나타나며, 이 그룹 판독 데이터(149)는 FIFO(150)에 래치되어 판독 데이터(62A/B)가 되거나 또는 기입 데이터(107)인 것처럼 인가되어 메모리(의 모든 뱅크)에 다시 기입된다. 이러한 다양한 메모리 트랜잭션을 관리하는 것 또한 스테이트 머신(193)이다.

<113> 슬레이브 DRAM 제어기를 설명하기 위해, 작성기 회로(146)의 블록도가 도시되어 있는 도 7을 참조한다. 앞에서, 작성기 회로는 3개의 그룹(R100으로서 저장된 데이터)에 걸친 모두 12개 뱅크 또는 1개 그룹(R33으로서 저장된 데이터)의 모두 4개의 뱅크에 대해 동일 어드레스로 판독하고, 그 내용을 12개(4개)에서의 다른 워드가 그 비트 포지션에 1을 가질 경우에도 비트 포지션에 0을 유지하는 한 워드에 병합하는 기능을 갖는다고 설명한 바 있다. 이를 위해, 도 6의 마스터 DRAM 제어기는 각각의 적합한 그룹에서의 모두 4개의 뱅크가 동일 어드레스를 사용하여 판독되도록 할 것이다. 그러므로, R100의 경우에 대해서는, 뱅크 0, 3, 6, 9(이 예에서는 R100 동작에 대해 이렇게 이름이 붙여진 0 내지 11개의 뱅크의 시퀀스에서--임의의 그룹내에서는 그 안의 뱅크가 단순히 0 내지 3으로 이름 붙여짐)가 그룹0 판독 데이터(GROUP 0 READ DATA)(134)에 관한 시퀀스로 나타날 것이고, 뱅크 1, 4, 7 및 10은 그룹1 판독 데이터(GROUP 1 READ DATA)(135)에 관한 시퀀스로 나타날 것이며, 뱅크 2, 5, 8 및 11은 그룹2 판독 데이터(GROUP 2 READ DATA)(136)에 관한 시퀀스로 나타날 것이다. 뱅크 0, 1 및 2에 대한 데이터는 한 사이클 동안에 동시에 나타날 것이며, 한편 뱅크 3, 4 및 5에 대한 데이터는 다음 사이클 동안에 동시에 나타날 것이며, 나머지 뱅크에 대한 데이터도 이와 같이 이루어질 것이다. 이들 데이터(134-136)내의 각각의 비트에 대해, 모두 3개 그룹내의 비트가 세트될 때에만 그 출력이 참(true)인 대응 AND 게이트(예를 들어, 151 및 152)가 존재하며, 그 어드레스에서 어떠한 기록된 장애(0들)가 있다면, AND 게이트의 출력 또한 0이 될 것이다. 이것을 4회 시행하여(뱅크 대 뱅크 인터리빙의 각각의 단계에 대해 한번) AND 게이트의 출력을 관련 래치(예를 들어, AND 게이트(151)에 대해서는 래치(156), AND 게이트(152)에 대해서는 래치(157))에 포착해두어야 한다. 래치의 상태는 AND 게이트에 피드백되고, 이에 의해 래치가 0을 포착한다면 AND 게이트의 출력은 여전히 검사를 필요로 하는 뱅크가 있음에도 불구하고 0을 유지한다(이후부터는 아래에 설명되고 도 5에 설명된 포맷에 일치하는 더욱 유용한 포맷을 사용함으로써 뱅크를 0 내지 11로 이름을 붙이는 번거로움을 덜 것이다. 0 내지 11로 이름을 붙이는 것의 유용한 제시성에도 불구하고, 이것은 R100 동작에 국한되는 느낌을 주며, 그 스타일을 R33까지 연장하는 것은 혼동을 줄 것이다).

<114> 이 시점 작업까지 설명된 바와 같은 체계는 작성이 시작되기 전에 래치가 세트되어야 한다는 것을 전제로 한다. 그러나, 이것은 시간이 소모되며, 래치(156-157)의 초기 상태에 상관없이 체계 작업을 갖도록 하는 것이 바람직하다. 이것은 래치 출력을 입력으로서 수신하는 2-입력 OR 게이트(153, 154)에 제1 사이클(FIRST CYCLE) 신호(155)를 입력으로서 인가함으로써 해결된다. OR 게이트(153-154)의 출력은 AND 게이트(151-152)에 대한 "재순환" 입력으로 인가된다. 제1 사이클 신호(155)는 마스터 DRAM 제어기(109)내의 스테이트 머신(193)에 의해 발생되고, 각각의 그룹의 제1 뱅크의 판독 동안에만 참이다. 2개의 DRAM 메모리 세트(0과 1)에 대해, 이들은 뱅크 0:0:0, 0:1:0, 0:2:0, 0:3:0 및 1:0:0, 1:1:0, 1:2:0, 1:3:0(여기서의 포맷은 메모리 세트#:그룹#:뱅크# 임)이다. 신호 제 1 사이클(155)의 효과는 래치(156-157)의 상태를 제1 사이클 동안 돈케어(don't care) 상태로 하고, AND 게이트(151-152)가 정확한 결과를 발생하도록 한다. 이들 결과는 래치(156-157)를 정확하게 설정하며, 제 1 사이클(155) 후에 작성 동작의 듀레이션 동안 거짓이 된다.

<115> 모두 4개 세트의 뱅크가 판독된 후, 32개의 래치(156-157)는 작성된 그룹 판독 데이터(COMPOSED GROUP READ DATA)(147)을 포함하며, 이 데이터는 이전에 설명된 방식으로 도 6의 마스터 DRAM 제어기(109)에 의해 사용된다. 이 동일 메카니즘이 R33 동작을 위해 어떻게 작업하는지를 이해하기 위해 추가의 정보가 필요하다. 사용되지 않는 또는 비활동의 그룹 N 판독 데이터 버스(GROUP N READ DATA BUS)(R33 동안 134, 135 및 136 중의 2개임)는 모두의 것으로서 나타난다. 이것은 R100에 대해 작업하는 동일 메카니즘이 R33에 대해서도 정확하

게 기능하도록 한다.

- <116> 도 8에는 도 5의 슬레이브 SDRAM 제어기(110, 111, 112)의 블록도(158)가 도시되어 있다. 슬레이브 SDRAM 제어기의 중앙 구성요소는 스테이트 머신(161)이며, 이 스테이트 머신은 링 버스(85)에 접속시킴으로써 세트되는 약간의 제어 레지스터(180)를 포함한다. 그룹 N 어드레스(GROUP N ADDRESS)(167, 169 또는 171)는 레지스터(160)에 의해 포착되고 스테이트 머신(161)에도 접속되는 FIFO(159)에 인가된다(N에 대한 값은 0, 1 또는 2 이 될 것임). 레지스터(160)로부터는 GROUP N ADDRESS(175)가 대상으로 하는 그룹을 구성하는 SDRAM 칩에 인가된다. 스테이트 머신(161)은 또한 마스터 SDRAM 제어기(109)로부터 그룹 사이클 제어(GROUP CYCLE CONTROL) 정보(172, 173 또는 174)를 수신한다. 이 정보로부터, 어떤 모드의 동작과 구성이 유효한지를 알뿐만 아니라, 스테이트 머신(161)이 대상으로 하는 그룹에 대한 SDRAM 제어 신호(176)(RAS, CAS, 칩 인에블 신호 등을 포함)의 적합한 시퀀스를 작성할 수 있다. 이들 제어 신호(176)를 발생하는 방식에 의해 인터리빙을 실제로 달성하는 것은 스테이트 머신(161)이다.
- <117> 스테이트 머신(161)은 또한 리프래시 타이머(도시 생략)를 포함하며, 이 타이머는 시간 종료시에(대략 40 μ s 후) 일부의 일회분의 리프래시(some installment of refresh)가 수행되는 동안 외부로부터의 추가의 동작을 차단한다. 리프래시는 한번에 한 행씩 수행되고 그 행내의 모든 열에 대해 수행된다. 각각의 다음 일회분의 리프래시는 다음 행에 수행된다. 이 방식의 동작을 용이하게 하기 위해, 슬레이브 DRAM 제어기 및 그 외부 환경은 파이프라인 연결되고(이들 FIFO 모두), 슬레이브 SDRAM 제어기 동작의 본래 속도가 143 Mhz 이어서 리프래시에 전용된 그 시간의 대략 7%는 합계 100 Mhz 속도로 응답하기 위한 시간으로 남겨진다.
- <118> 한편, 그룹 N 기입 데이터(GROUP N WRITE DATA)(166, 168 또는 170)는 FIFO(162)에 인가되고, 이 FIFO의 출력은 (2:1) \times 32 MUX(163)에 대한 입력으로서 인가된다. MUX(163)의 출력은 대상으로 하는 그룹을 위한 GROUP N DATA(178)인 신호에 결합된다. 이들 라인(178) 상에는 기입될 데이터와 관독된 데이터가 모두 나타날 것이다. 기입될 데이터는 MUX(163)로부터 보내지는 것이며, FIFO(162)를 통해 그룹 N 관독 데이터(GROUP N WRITE DATA)를 가지고 또는 방금 관독되어 레지스터(164)에 저장된 데이터로부터 발생된다. 경로(179)는 이후자의 경우를 나타내며, 관독-수정-기입 스타일 동작이 수행될 때 발생한다("수정" 파트는 복잡할 것이고, MUX를 가지고 행해지며, 여기서는 관심 대상이 아니다). 슬레이브 SDRAM 제어기의 외부에서 사용하기 위해 관독되는 데이터는 레지스터(165)내에 추가로 래치되고, GROUP N READ DATA(134, 135 또는 136)가 된다.
- <119> 메모리 세트의 다양한 동작 모드 및 구성은 다음의 표 2 내지 표 10에 의해 예시된 어드레싱 체계에 의해 용이하게 된다. 도시된 것은 SDRAM 메모리 세트에 대한 것이다. SRAM 메모리 세트에 대한 어드레싱 체계는 이와 유사하지만 다소 더 간단하다(그 이유는 SRAM 메모리 세트가 그룹 및 뱅크를 갖지 않기 때문이다).

표 2

기호 정의

기호	의미
M	메모리 세트를 스택하기 위한 메모리 세트 선택 비트
G	메모리 세트내의 그룹을 스택하기 위한 그룹 선택 비트
B	그룹내의 뱅크를 스택하기 위한 뱅크 선택 비트
R	행 어드레스 비트
C	열 어드레스 비트
F	협폭 워드 동작을 위한 필드 선택 비트
E	칩 인에이블

<120>

표 7

1-비트 모드 단일 메모리 세트‡

비트	3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0						
Pos	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0							
유형	-----																																						
512M	G	G	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	F	F	F	F	F
256M	G	G	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	F	F	F	F	F
128M	G	G	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	F	F	F	F	F
64 M	G	G	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	F	F	F	F	F

‡G는 R100 동작 동안 없거나 무시됨

<125>

표 8

32-비트 모드 단일 메모리 세트‡

비트	3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0				
Pos	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0						
유형	-----																																					
512M	G	G	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	C	C	C	C
256M	G	G	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	C	C	C	C
128M	G	G	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	C	C	C	C
64 M	G	G	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	C	C	C

‡G는 R100 동작 동안 없거나 무시됨

<126>

표 9

단일 메모리 세트 L100(항상 32-비트)

비트	3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0				
Pos	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0						
유형	-----																																					
512M	G	G	B	B	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	C	C	C	
256M	G	G	B	B	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	C	C	C	C
128M	G	G	B	B	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	C	C	C	C
64 M	G	G	B	B	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	C	C	C

<127>

표 10

스택 메모리 세트 L100(항상 32-비트)

비트	3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0				
Pos	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0						
유형	-----																																					
512M	G	G	B	B	E	M	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	C	C	C	
256M	G	G	B	B	E	M	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	C	C	C	C
128M	G	G	B	B	E	M	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	C	C	C	C
64 M	G	G	B	B	E	M	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	C	C	C

<128>

<129>

표 2는 표 5 내지 표 10에 사용된 각종 기호의 의미를 나타낸다. 즉, 상이한 ECR 모드의 구성 및 동작 하에 어드레스내의 각종 비트 포지션에 어떤 해석이 제공되는지를 말해준다. 표 5 내지 표 10에서 설명하는 어드레싱 체계는 메모리 부품 자체에 직접 적용되기 보다는 마스터 및 슬레이브 메모리 제어기내의 스테이트 머신에 적용되는 것으로서 생각되어야 한다.

<130>

표 3은 ECR이 거주될 수 있는 각종 메모리 부품을 나타내며, 그들의 기대된 조직의 일부를 말해주고 있다. 특히, 도 9의 간략화된 그룹 SDRAM 블록도(181)를 참조하여, 이들이 16개의 데이터 비트를 갖는 부품(182-185)이어서 전체 폭 워드(32 비트)를 얻기 위해 2개의 부품이 앞뒤로 연결(182 & 183, 184 & 185)되어 어드레스될 것이고, 그들의 출력 비트가 하나의 커다란 필드(178)로 결합된다는 점에 유의하기 바란다. 또한, 각각의 부품은 4개의 뱅크를 갖고 이들 뱅크의 각각이 부품(R100 및 R33 동작 동안 이용된)의 어드레스 공간의 별도의 예를 실시한다는 점에 유의하기 바란다. 대안적인 시각은 관련된 2개의 뱅크 선택 비트를 추가의 어드레스 비트로서 해석하는 것이며, 이는 L100 동작에 유용하다. 2쌍의 메모리 부품이 존재하며, 각각의 쌍은 칩 선택(CHIP SELECT) 신호의 공통적인 예(부품 182 및 183에 대해서는 신호 188, 부품 184 및 185에 대해서는 신호 187)에 의해 선택된다. 이들 CHIP SELECT 신호는 어드레싱 시점에서 마치 행 어드레스 비트와 동등한 것처럼 기능한다. 이 점에서, 부품(182, 184)은 데이터의 32비트의 절반을 위한 저장 공간을 제공하기 위해 결합하는 한편, 부품(183, 185)은 나머지 절반을 저장한다. 그래서, 부품들이 512 Mbit 부품들이면, 도시

된 배열은 4개 뱅크의 각각에서의 16M 어드레스 × 32비트 데이터 워드 메모리이다. E 비트(메모리 세트내의 어드레스당 하나로 존재)는 메모리 세트에서 각각의 그룹에 공통적인 상보형 CHIP SELECT 신호(187, 188)로 바뀐다. 도 9가 2개의 DRAM 메모리 세트 중의 하나에서 3개 그룹 중의 단지 하나를 다루고 있다는 점에 유의하기 바란다. 표 4는 테스트 프로그램(DUT 아키텍처와 내부 조직에 초점이 맞춰진)의 로직과 ECR 및 그 메모리 세트를 위한 모드와 구성간에 나타날 수 있는 관계를 이해하는데 유용하다. 명심해야 할 중요한 개념은 DUT와 ECR 모두 이들이 매우 상이한 내부 조직과 작동 방식을 갖는다 하더라도 동일 순서 시퀀스의 비트를 어드레스로서 해석할 수 있다는 점이다. 그러므로, 표 4는 32 오더 어드레스 비트에 대한 상당히 많은 상이한 가능 DUT 관련 해석 중의 2개만을 나타내고 있다. 표 4 내지 표 9는 이들 동일한 32 어드레스 비트의 ECR 관련 해석을 나타내고 있고 있으며, 2개의 해석은 서로 거의 관계가 없다. 매우 높은 레벨에서, 테스트 프로그램의 작성자가 원하는 원칙적인 것은 그(즉, 프로그램적 제어하의 메모리 테스트)가 ECR에 데이터를 기입하고 차후 분석을 위해 다시 그 데이터를 취할 수 있는 지식을 가지고 DUT에 인가되는 어드레스에 대응하는 ECR에 어드레스를 갖도록 하는 능력이다. 한편으로는, 테스트 엔지니어는 DUT 아키텍처에 관련된 의미를 포착하기 위한 방식을 연구한다. 다른 한편으로는, ECR 어드레싱 비트가 어떠한 신형의 DUT에도 상관없이 그 내부 동작에 관련한다. 단순한 예를 제시하기 위해, ECR 에서의 Z 어드레스에 대한 개념이 없으며, 그렇다면 DUT 에서 특정 Z 어드레스 또는 Z 어드레스의 범위에 대응하는 어드레스 또는 어드레스의 범위를 ECR 메모리가 갖도록 하는 방법을 어떻게 해결할 수 있을까? 당분간은, 어드레스 분류기 및 데이터 분류기가 단순히 그들의 내용을 변경되지 않은채로 넘겨주고, ECR이 적합한 모드의 동작(충분한 속도, 충분한 어드레스)으로 발생하는 한, DUT에 충분히 우수한 어떠한 어드레스도 ECR에 충분히 우수하다라고 가정하자는 충고를 가지고 이 문제를 무시하자(사실, 그 방법을 지속적으로 생각하는 것은 어떠한 묘수를 쫓는 것이다. 그러나, 그 동안 다른 주제에 관심을 두어왔고, 간결함을 위해 적어도 일시적으로 생각할 것이다).

<131> 표 5는 512M SDRAM이 단일 메모리 세트의 부품으로서 구성될 때 이 SDRAM이 어떻게 어드레스되는지를 나타내고 있다. 워드 폭이 전체 32 비트인 경우(표의 가장 밑의 행)를 고려해보자. 이것은 2개의 상이한 모드의 동작, R100(멀티플렉싱과 인터리빙을 사용하는 100 Mhz에서의 랜덤 어드레싱) 및 R33(멀티플렉싱없이 인터리빙만을 갖는 33 Mhz 랜덤 어드레싱) 하에서 발생할 수 있다. 각각의 경우, 8M의 본래 어드레스공간이 13개의 R과 10개의 C에 의해 제공된다. 칩 인에이블 비트 E가 포함될 때 16M의 어드레스 공간(4개 뱅크의 각각에서)이 된다. R100에서, 별도의 사용자 개시 그룹 어드레싱(G 비트)이 사용되지 않으며, 마스터 DRAM 제어기의 스테이트 머신은 그룹에 대응하는 대응하는 임의의 어드레싱 비트를 필요로 하지 않고서도 다음 메모리 트랜잭션을 단순하게 다음 그룹의 버스로 전송한다. 이것은 자신의 버스를 갖는 각각의 그룹으로부터 직접적으로 후속하며, 하나의 버스 상의 어드레스는 또다른 버스 상의 어드레스와 완전히 별도의 어드레스 공간의 예이다.

<132> 단지 동작 모드가 R33 이라면, 이 모드에서는 멀티플렉싱이 이루어지 않으며, 그럼에도 불구하고 33 Mhz의 더 낮은 속도에서이기는 하기만 어드레스된 메모리 트랜잭션을 3개의 그룹의 각각에 (동시에) 전송할 수 있다. 각각의 그룹을 그룹 어드레싱 비트(G)로 설비하면, 깊이가 3배(48M의 어드레스가능성)인 결합된 어드레스 공간을 형성하기 위해 3개의 그룹을 스택할 수 있다. 이 모드에서, 어드레스된 그룹만이 메모리 트랜잭션에 응답할 것이다. 이러한 이유로 슬레이브 SDRAM 제어기의 스테이트 머신(161)이 그룹 N 어드레스(GROUP N ADDRESS)에 접속되며, 그 슬레이브 SDRAM 제어기는 어드레스된 그룹용이거나 어드레스된 그룹용이 아닐 수도 있다.

<133> 표 5에 도시된 어드레싱 체계는 또한 협폭 워드 구성을 지원한다. 이것은 전체 워드내에서 작은 필드로의 부분할(subdivision)을 위한 계산에 추가의 어드레스 가능성을 제공하는 경우이다. 이들 작은 필드는 0(전체 32 비트 워드) 내지 5(32개의 1비트 필드)의 범위에 있는 F 비트이다. 이 모드의 동작을 실시하기 위한 또다른 이유는 무엇 때문에 슬레이브 SDRAM 제어기의 스테이트 머신(161)이 그룹 N 어드레스(GROUP N ADDRESS)를 수신하느냐하는 것이다. 또한 무엇 때문에 MUX(163) 및 데이터 경로(179)(도 8 참조)가 제공되는가하는 것도 이유가 된다. F 비트가 DRAM 부품 자체에 가지 않으며 전체 32비트 워드를 위한 것처럼 어드레스된다는 것을 이해할 것이다. 슬레이브 SDRAM 제어기가 이 추가의 협폭 워드 성능을 제공하며, F 비트가 SDRAM 제어기에서 사라져 적합한 제어기 행동에 의해 대체될 것이다.

<134> 표 6은 표 5와 유사하다. 표 5가 스택되거나 또는 스택되지 않을 그룹을 다루는 반면, 표 6은 스택되는 메모리 세트를 다룬다. 예로서 512M 부품을 사용하며, 표 6의 축소 버전은 더 작은 메모리 부품에 대해 존재한다(또한 표 5의 축소 버전도 마찬가지임). 이들 다른 테이블들은 간략화를 위해 생략되었다. 2개의 메모리 세트를 스택하기 위해서는, 2개의 메모리 세트가 서로 동일하게 구성되도록 하고 어드레스 공간을 두배로 하기 위해 하나의 여분의 비트(M 비트)을 제공하여 단지 하나를 사용하는 대신에 2개의 메모리 세트를 일치시켜야

할 필요가 있다. 이 여분의 비트를 제어하는 것은 테스트 프로그램이다. 이 테스트 프로그램에 응답하는 것은 마스터 DRAM 제어기이다(이것이 도 6의 스테이트 머신(193)이 ADDRESS(132)에 접속되어야 하는 이유이다). M 비트는 그 시점에서 어드레싱 비트로서 소멸하여 그 경우의 스테이트 머신(193)을 갖는 메모리 세트가 어드레스된 메모리 세트인지의 여부에 좌우되는 메모리 활성화의 존재 또는 부재에 의해 대체될 것이다.

<135> 표 7 및 표 8을 참조하면, 이들 표는 지금까지 설명되어 왔던 것으로부터 충분히 유추될 수 있다. 즉, 표 7의 첫번째 라인은 표 5의 첫번째 라인과 동일하고, 표 8의 첫번째 라인은 표 5의 마지막 라인과 동일하다. 표 6과 표 7의 수직축이 부품 크기인 반면 표 4의 축이 협폭 워드 모드라는 점이 상이하다.

<136> 표 9와 표 10은 L100 구성을 다루고 있다. 앞에서 L100은 "선형" 또는 로컬러티 어드레싱(행 어드레스에서의 최소 변경)이라고 설명한 바 있다. 여기에서는 멀티플렉싱도 인터리빙도 존재하지 않는다. 어드레스 공간의 12개의 별도의 예를 가지고 사용했던 것에 비해, 지금은 하나의 어드레스 공간을 갖고 있지만, 이 어드레스 공간은 깊이가 12배이다. 이것은 마스터 DRAM과 슬레이브 SDRAM 메모리 제어기에 인가하기 위해 4개 더 많은 어드레스 비트에 대한 필요성을 발생시킨다. 이들 여분의 비트는 표 9 및 표 10에 도시된 GG 및 BB 비트이다. 앞에서와 같이, 이들 비트는 제어기내로 사라진다. 즉, 스마트 스테이트 머신과 그룹이 되는 메모리의 집합을 위한 별도의 버스에 의해 이루어질 수 있는 대응 기능성(corresponding functionality)에 의해 대체될 것이다. 2개의 표간의 차이점은 표 6과 관련하여 이전에 설명된 바와 같이 동작하는 M 비트이다. 또한, L100 동작이 협폭 워드 모드의 개념을 배제시켜 이를 지원하기 위한 시스템 레벨에서 충분한 어드레스 비트가 존재하지 않는다는 점에 유의하기 바란다.

<137> 메모리 제어기내의 스테이트 머신에 대해 간략하게 설명할 것이다. 먼저, 마스터 DRAM 제어기(109)내의 스테이트 머신(193)을 고려해보면, 스테이트 머신의 기본적인 관련 사항은 다음과 같다. (A) 메모리 세트의 동작 모드가 R100이면, 그룹 중에서의 멀티플렉싱이 요구된다. 스테이트 머신은 어느 그룹이 그룹의 라운드 로빈 스퀀스에서 "다음"이 되는지 알고 있고, 그 다음 그룹에 대해 버스의 집합(GROUP N ADDRESS, WRITE DATA, CYCLE CONTROL & READ DATA, N=0, 1, 2)에 관해 메모리 트랜잭션을 시행한다. (B) 메모리 세트의 동작 모드가 R100 이외의 것이라면, 멀티플렉싱이 사용되지 않는다. 이러한 경우, 사용하기 위한 그룹은 표 5 내지 표 10과 관련하여 진술된 바와 같이 유입 어드레스 비트에 의해 결정되며, 메모리 트랜잭션은 그룹 어드레스에 의해 선택된 바와 같은 버스의 집합(ADDRESS, WRITE DATA, CYCLE CONTROL & READ DATA)에 관해 시행된다. 이제, 경우 (A)와 (B)는 R100과 R33에 대해 어드레스에 대한 관독 동작이 그 어드레스에 기입된 최종 데이터를 발생시킬 것이라는 것을 조금도 보장하지 못한다는 것을 용이하게 이해할 수 있다하더라도 항상 기입 동작이 되는 것으로 고려될 수 없다(이것을 작성 동작이 고려된 것으로 다룬다). 그럼에도 불구하고, 관독 동작이 가능하며, 메모리가 작성된 것으로 제시한다. 그러므로, 문제가 없다. 또한, 경우 (B)는 단지 "NOT R100"으로 불리워지며, L100 동작 동안 획득할 것이다. 여기에서는 멀티플렉싱이 존재하지 않으며, 인터리빙 또한 존재하지 않기 때문에, 관독 동작은 예측 가능하게 행동하도록 기대될 수 있다.

<138> 그러나, 경우 (A)와 (B)가 유일한 경우라고 가정해서는 안된다. 또한 작성에 대한 경우 (C)가 있다, 작성 동작은 R100이 유효하지 아니면 R33이 유효한지에 좌우되어 그 자신의 특별한 활동을 발생한다(L100 기입은 관독 이전에 작성을 위한 필요성을 발생하지 않는다). R100 경우에는 동일 어드레스에 대한 4개의 연속적인 관독이 이루어질 필요가 있다(모든 뱅크에 걸쳐 인터리빙이 단계를 밟아 가도록 하기 위해). 이들 4개의 관독은 각각의 그룹에 관해 동시에 수행된다. 그러므로, 4개의 기입 동작(각각의 그룹에 관해 동시에 행해진)이 작성된 결과를 모두 12개 뱅크내로 되돌려주도록 할 필요가 있다. 이 활동 모두는 어드레스당 100 Mhz 속도로 진행하지는 않을 것이라 하더라도 단일 코멘드의 결과로 될 수 있다(이것은 한 그룹내의 모두 4개의 뱅크에 동시에 기입하기 위한 방법(사용하는 SDRAM 부품을 가지고)이 없기 때문이다).

<139> 작성은 R33 모드의 동작에서 가능하다. 4개의 관독 동안 2개의 어드레스되지 않은 그룹이 "셧다운(shut down)"되어야 하고 기입이 인터리빙을 통해 단계를 밟아 나가야한다는 점에서 사소한 차이가 있다. 관독 동안, 그룹 N 관독 데이터 버스(GROUP N READ DATA BUSES)(134, 135, 136) 중의 2개의 어드레스되지 않은 것은 어드레스된 것이 평소처럼 수행하는 동안 나타나는 것 모두를 가져야 한다. 이것은 작성기(146)에서 적합한 결과를 발생시킬 것이며, 이 결과는 작성된 그룹에 다시 기입된다. 다른 그룹에서의 대응 어드레스가 사용중인 어드레스 공간에서 실제로 상당히 상이한 위치이기 때문에(스택 그룹), 그룹의 4개의 뱅크만이 기입된다. R33에서의 그룹에 걸친 작성은 상이한 크레딧 카드를 위해 계좌 번호를 추가하는 것과 같을 것이며, 그 결과는 유용한 계좌 번호가 되지 않는다. 마스터 DRAM 제어기에서의 스테이트 머신은 어느 그룹이 그들의 관련 버스를 통해 그룹 사이클 제어(GROUP CYCLE CONTROL) 정보를 취하느냐를 결정함으로써 이 모두를 해결할 수 있다. 또한, 그룹 N 관독 데이터(GROUP N READ DATA) 버스가 비활성인 때를 나타내거나 하이로 될 수 없을

수도 있다. 그 경우가 아니라면, 작성에 참여하지 않은 그룹으로부터의 비트를 가리기 위해(mask out) AND 게이트(151-154)에 대한 입력에 걸친 추가 제어가 필요할 것이다.

<140> 이제는 도 8의 슬레이브 SDRAM 제어기의 스테이트 머신(161)을 고려해보자. 여기에는 주의해야 할 중요한 것이 있다. 먼저, 스테이트 머신은 리프레시 타이머를 유지한다. 타이머가 제로로 떨어질 때, 스테이트 머신은 스테이트 머신이 제어하는 SDRAM에 리프레시 사이클을 발행한다. 이 시간 동안, 어떠한 유입 메모리 트랜잭션도 파이프라인에 발생되지 않아야 한다. 특정 행내의 모든 열은 발행된 리프레시 사이클에 의해 리프레시된다. 상태 머신(161)은 어느 행이 다음에 행해질지를 알고 있다. 리프레시 동작이 진행되지 않는다면, 정규 메모리 트랜잭션이 수행될 수 있다. 인터리빙이 유효하다면(R100 또는 R33), 트랜잭션은 다음 बैं크에서 시행된다. 인터리빙이 유효하지 않다면, 트랜잭션은 어드레스된 बैं크에서 시행된다. 어느 쪽의 경우에도, 메모리 사이클 제어 신호(176)의 정확한 시퀀스의 전송을 관리하는 것은 스테이트 머신의 태스크이다. 스테이트 머신의 태스크에는 상이한 환경을 위한 모든 방식의 인터리빙도 포함된다.

<141> 슬레이브 SDRAM 제어기에 의해 수행된 인터리빙은 SDRAM을 사용하는 기술에 익숙한 사람에게는 용이하게 이해될 수 있는 것이다. 이하에는, 대상으로 하는 다양한 메모리 트랜잭션의 일부와 그들의 관련된 인터리빙 체계간의 대응에 대한 함축된 버전을 구성하는 표 11 내지 표 15가 제공된다.

표 11

정의

- a = 활성화 행(& 행 어드레스)
- r = 판독(& 열 어드레스)
- w = 기입(& 열 어드레스)
- P = 선충전(& बैं크 선택)
- = 클럭 사이클
- Bn = AC 또는 D 상의 बैं크#에 대한 트래픽
- D = (별도의) 데이터 버스; i = (입력) 기입 데이터, 0 = (출력) 판독 데이터
- AC = (별도의) 어드레스 & 제어 버스

<142>

표 12

SDRAM 제어

SDRAM 핀	작동			
	p	a	r	w
행 어드레스 선택(RAS)	1	1	0	0
열 어드레스 선택(CAS)	1	0	1	1
기입/판독	0	0	0	1

<143>

표 13

오버레이 기입 및 오버라이트 기입

B0	p---a---r--o-wp---a---r--o-w
B1	p---a---r--o-wp---a---r--o-w
B2	p---a---r--o-wp---a---r--o-w
B3	p---a---r--o-wp---a---r--o-w
D	o-i-o-i-o-i-o-i-o-i-o-i-o-i
AC	p---pa--par-parwparwparwparwparw-arw--rw----

<144>

그룹 내의 하나의 OLW에 대해 16 사이클

<152> FCP를 고속으로 만드는 것은, 첫째, FCP가 L100과 동일한 방식으로 동작한다는 것이다. 즉, FCP는 또다른 선 충전(p) 및 또다른 활성화 행(a)을 발행하기 위한 필요성을 방지하고 후속의 r 및 w 동안 열 선택을 간단하게 변경하는 능력인 로컬러티의 장점을 취한다. 당연히, 때때로 슬레이브 SDRAM 제어기는 또다른 p와 a의 발행을 필요로 할 것이다. 이를 행하기 위한 필요성은 행 선택이 다음 어드레스를 변경하기 때문에 또는 최종 활성화 행 a가 다음 어드레스를 요구한 이후 시간이 경과되었기 때문에 발생할 것이다. 그러나, 대부분은 12 클럭 사이클에서 다수의 FCP가 발생할 것이다. 두번째로, FCP는 뱅크내의 각각의 위치에 대한 어드레싱의 하나의 예만을 사용하여 판독 및 기입 동작을 행하기 때문에 고속이 된다. 이것은 각각 자신의 어드레싱을 행하는 2개의 동작 대신 하나의 단일화된 동작이 존재하는 결과이다.

<153> 최종적으로, 작성된(COMPOSED) 플래그(CMP_FLG_MS#N)(190)가 어떻게 제어될 수 있는지에 관한 간략화된 블록도(189)를 도시하고 있는 도 10을 참조한다. 각각의 DRAM 메모리 세트에 대해 하나의 이러한 플래그가 존재하며, 이 플래그는 메모리 영역의 작성 무결성을 나타내기 위해 사용된다. 플래그가 세트되면, 메모리의 관련 영역을 작성된 것으로서 처리하는 것이 안전하며, 작성된 채로 유지한다. 그 메모리 영역에 대한 기입 동작은 기입된 어드레스에서의 작성을 망칠 가능성이 있을 것이며, 플래그를 소거하기 위해 사용된다. 플래그 자체는 (1) 링 버스를 통해 발행될 수 있는 래치(SET_CMP_FLG_MS#N)를 세트하기 위한 명시 명령어(explicit instruction)(197)와 (2) 관련 메모리 세트에 대한 FCP 동작이 수행되었다는 것을 나타내는 신호(196)(FCP_MS#N)의 논리 OR인(OR 게이트(194)에 의해 발생된) 신호(193)에 의해 세트되는 플립플롭 또는 래치(191)의 상태에서부터 발생한다. 옵션 (2)는 FCP가 작성을 달성하기 위해 사용되지 않는다 하더라도 테스트 프로그램이 플래그 세트를 취할 수 있도록 한다. 메모리 세트에 행해진 임의의 OLW(OLW_MS#N)(195)는 작성된 결과의 무결성에 대한 잠재적인 위협이 되며, 플래그를 소거하기 위해 사용된다. 플래그의 상태는 링 버스를 사용하여 점검될 수 있다

발명의 효과

<154> 전술한 특징에 따른 본 발명에 의하면, 메모리 테스터용의 에러 포착 RAM을 크기와 속도에 대해 구성할 수 있는 SDRAM 메모리 세트를 가지고 구성할 수 있게 된다.

도면의 간단한 설명

- <1> 도 1은 본 발명에 따라 구성된 광범위하게 재구성 가능한 비휘발성 메모리 테스터의 간략화된 블록도,
- <2> 도 2는 도 1의 DUT 테스터(6)의 간략화된 확대 블록도,
- <3> 도 3은 도 2의 블록도에서 도시된 ECR(에러 캐치 RAM) 메카니즘의 간략화된 기능 블록도,
- <4> 도 4는 도 3의 ECR 메카니즘의 더욱 상세한 블록도,
- <5> 도 5는 도 3 및 도 4의 블록도에서 도시된 DRAM 메모리 세트 제어기 메카니즘의 간략화된 블록도,
- <6> 도 6은 도 5의 블록도에서 도시된 마스터 DRAM 제어기 메카니즘의 블록도,
- <7> 도 7은 도 6의 블록도에서 도시된 작성기(composer) 회로의 블록도,
- <8> 도 8은 도 5의 블록도에서 도시된 슬레이브 SDRAM 제어기 메카니즘의 블록도,
- <9> 도 9는 도 6 및 도 8의 제어기에 의해 제어된 SDRAM의 그룹의 간략화된 블록도,
- <10> 도 10은 "작성된" 플래그의 활동성에 속하는 간략화된 블록도.

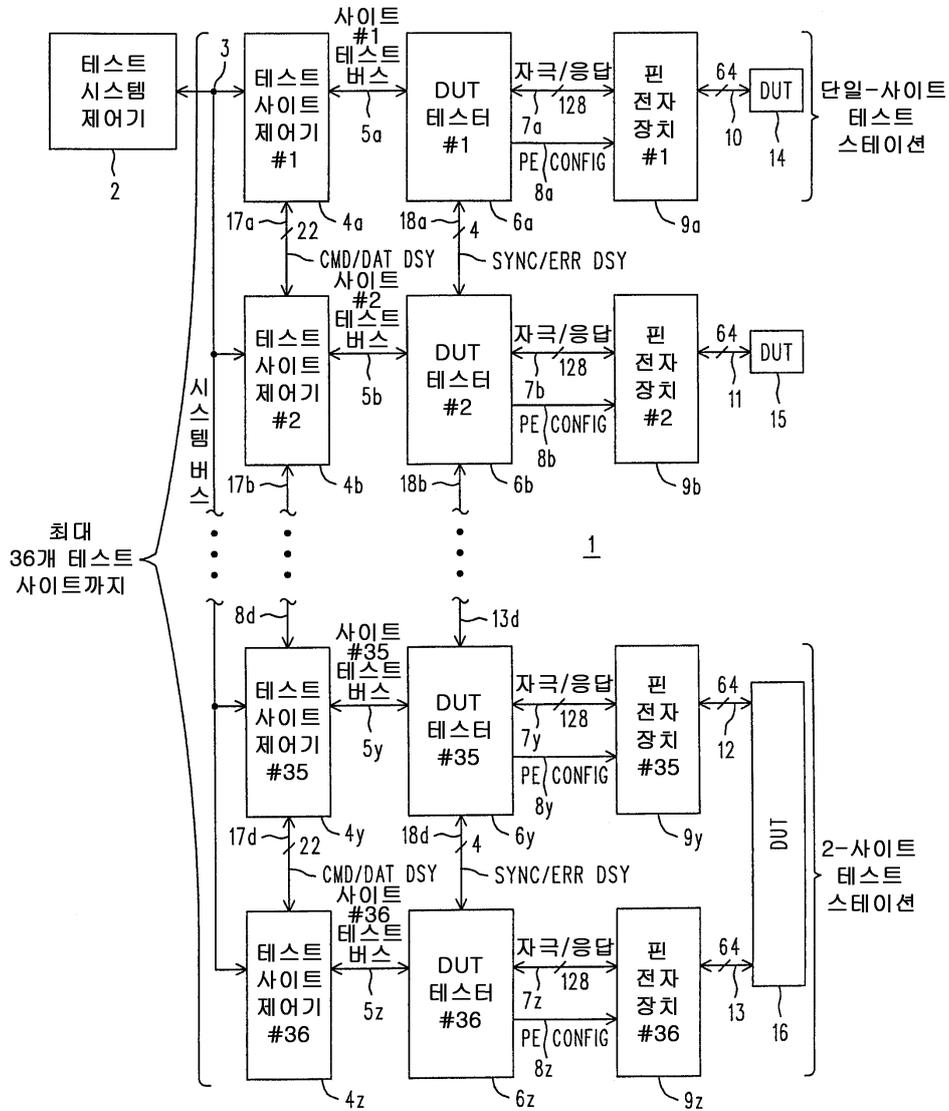
도면의 주요 부분에 대한 부호의 설명

- <12> 109 : 마스터 DRAM 제어기 110, 111, 112 : 슬레이브 SDRAM 제어기
- <13> 113-124 : 뱅크 146 : 작성기
- <14> 161 : 슬레이브 SDRAM 제어기(110, 111, 112)의 스테이트 머신
- <15> 176 : SDRAM 제어 신호

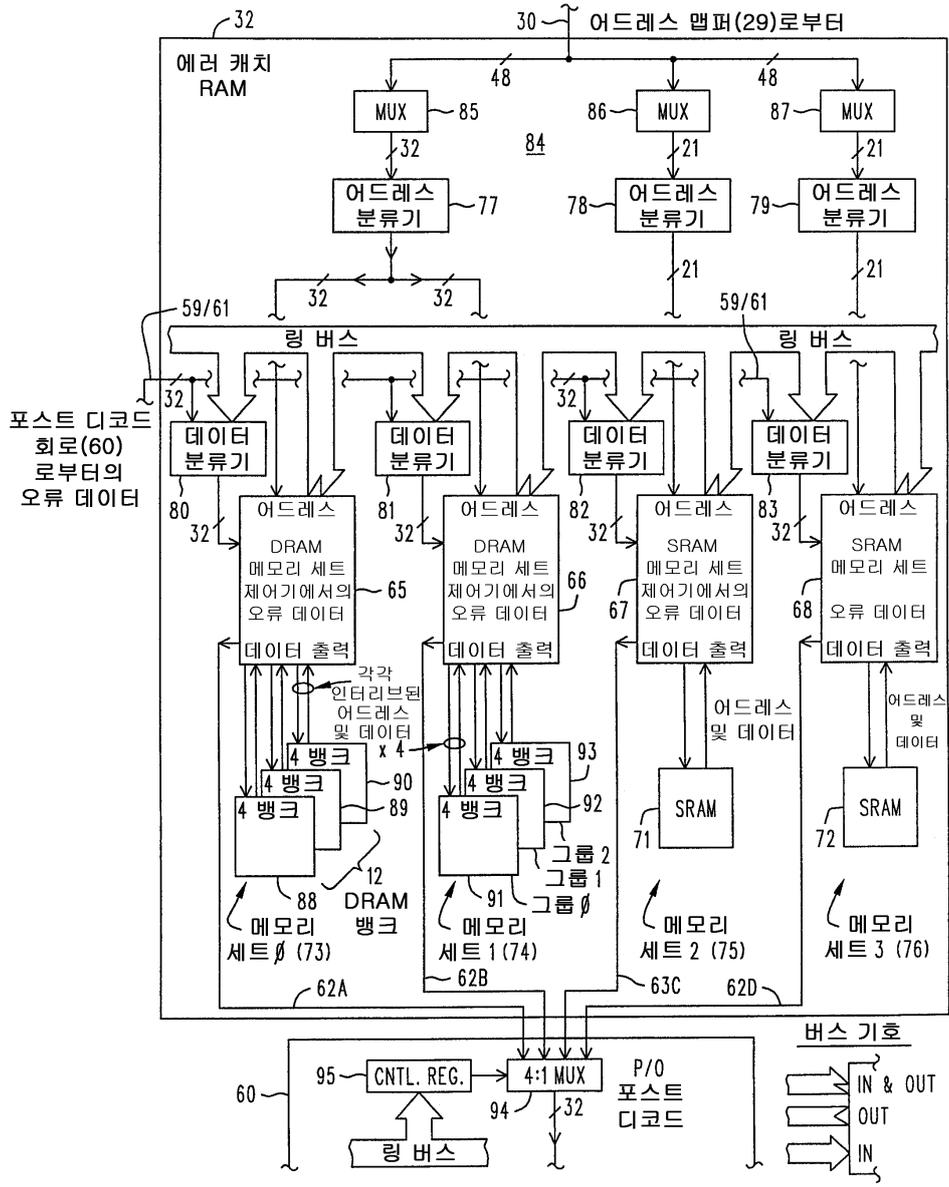
<16> 193 : 마스터 DRAM 제어기(109)의 스테이트 머신

도면

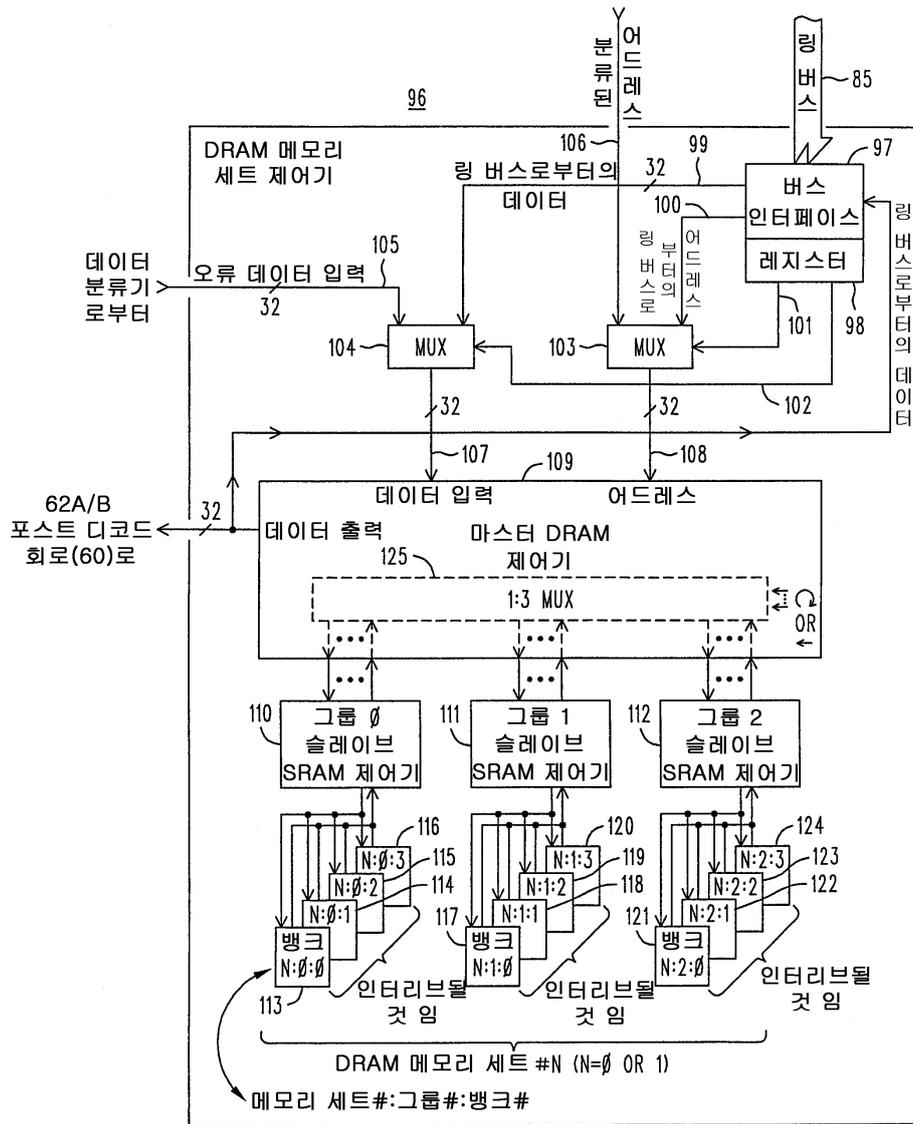
도면1



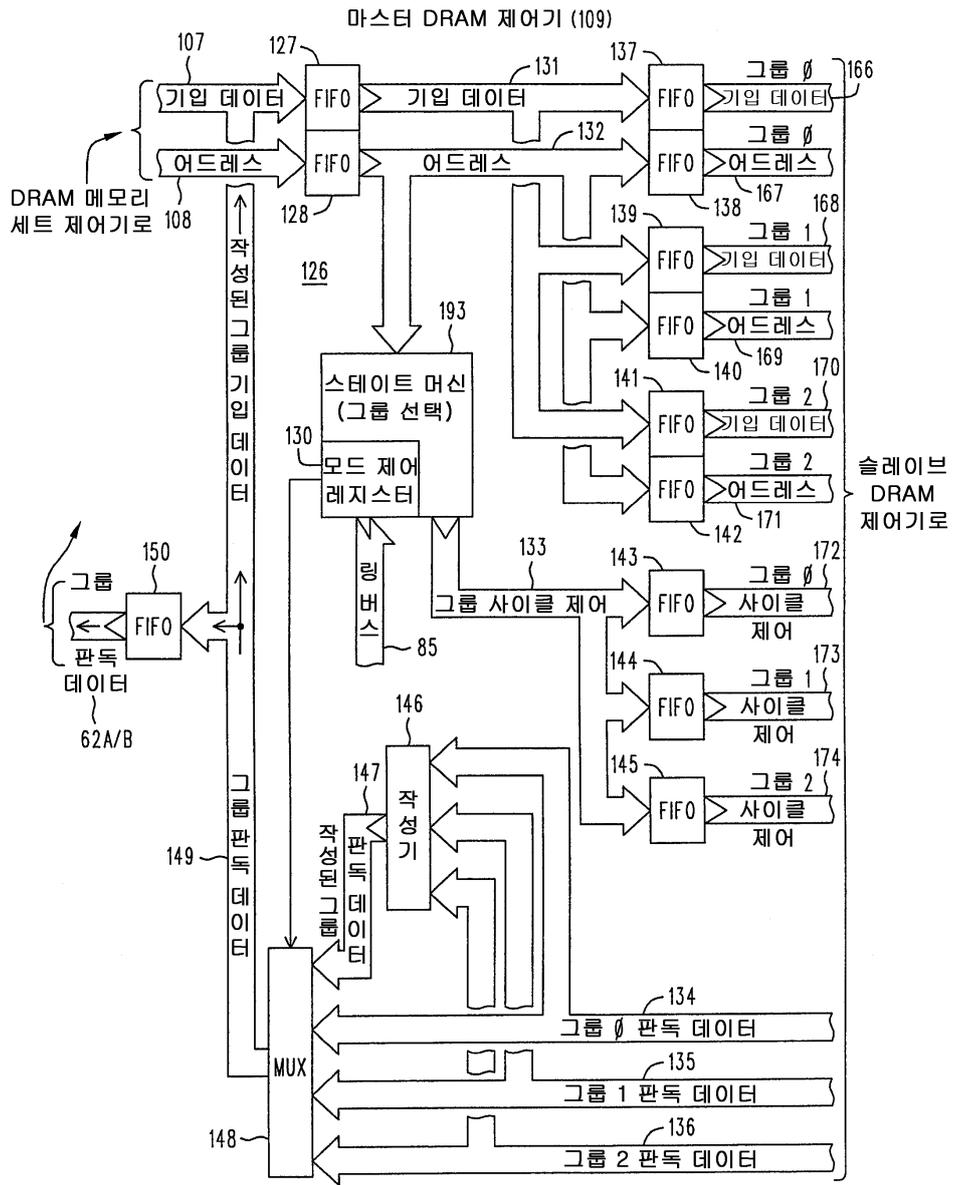
도면4

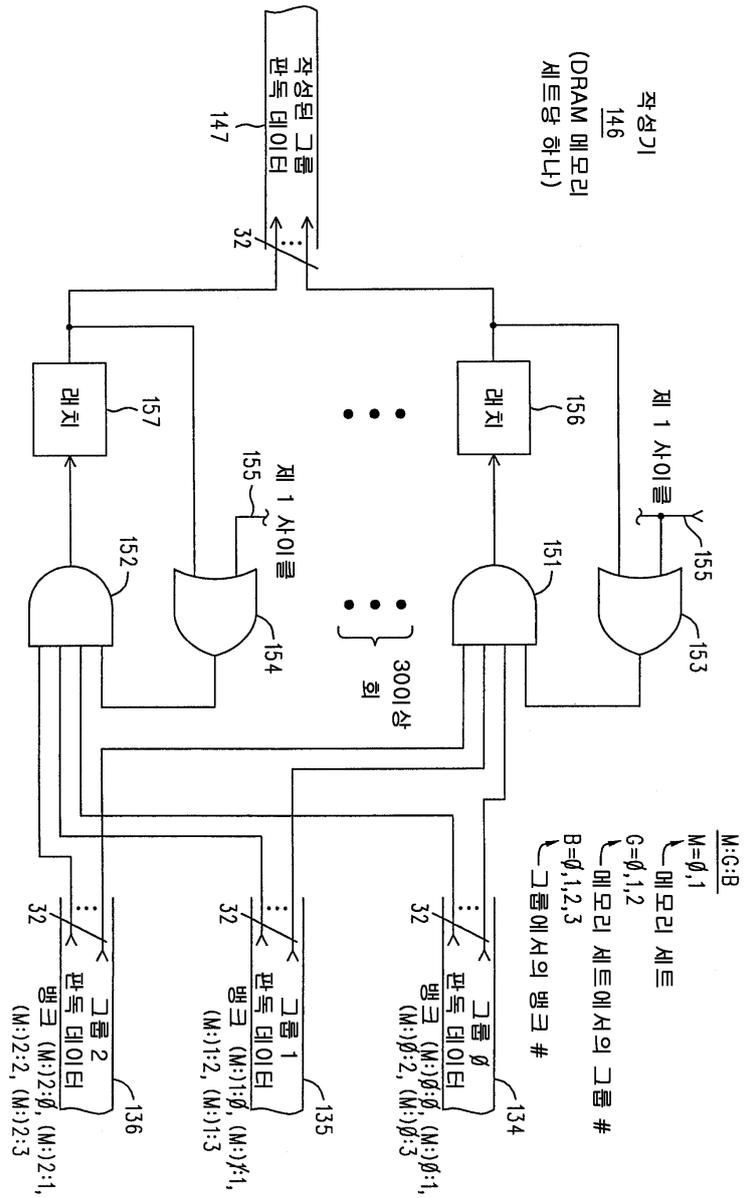


도면5



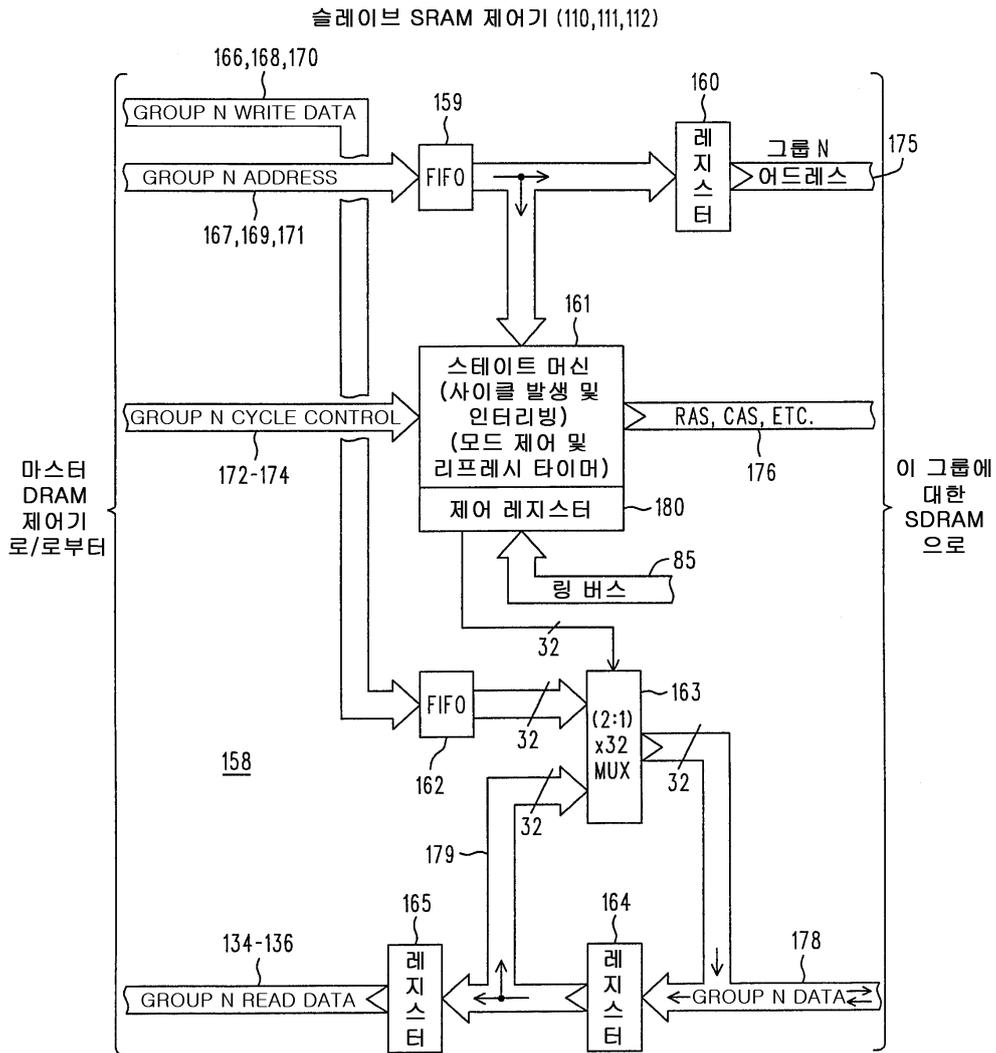
도면6



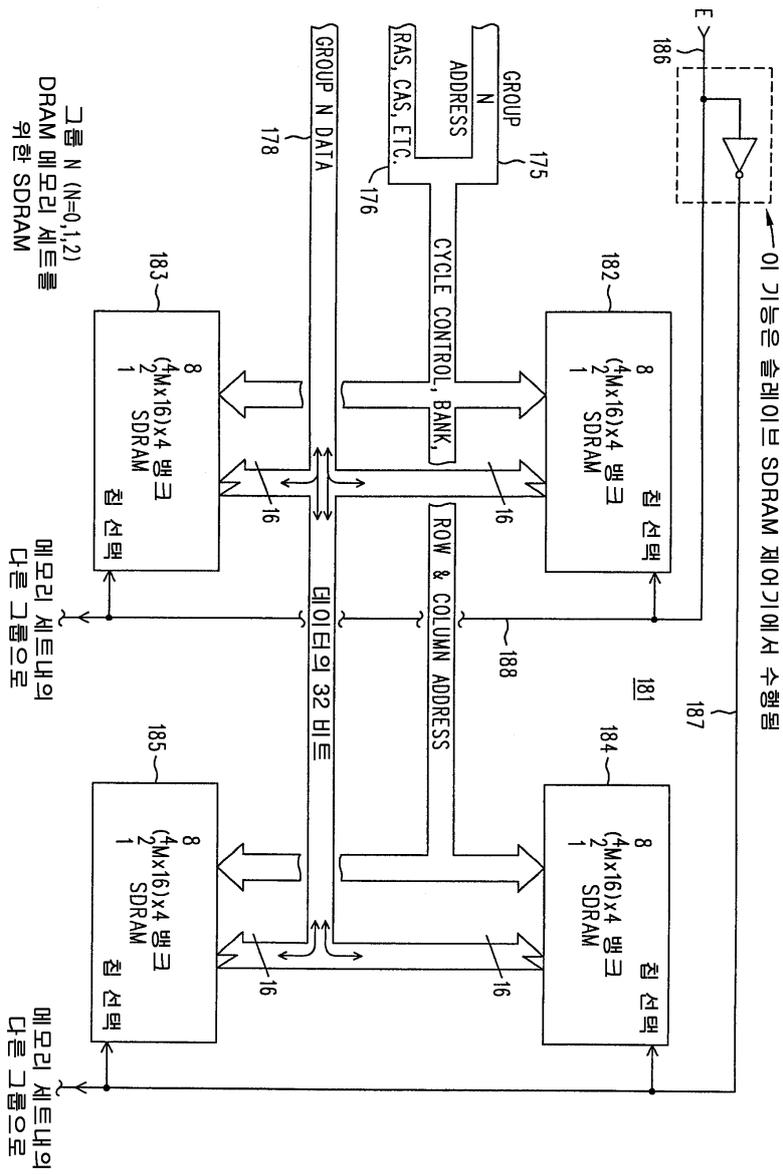


도면7

도면8



도면9



도면10

