(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl. [7]
G06F 13/00
G06F 9/38

(45) 공고일자 2001년11월26일
(11) 등록번호 10-0290325
(24) 등록일자 2001년03월02일

(21) 출원번호 10-1993-0030204
(22) 출원일자 1993년12월28일

(65) 공개번호 특1994-0015852
(43) 공개일자 1994년07월21일

(30) 우선권주장 998,080 1992년12월29일 미국(US)

(73) 특허권자 코닌클리케 필립스 일렉트로닉스 엔.브이.
요트.게.아. 롤페즈
네델란드왕국, 아인드호펜, 그로네보드스베그 1

(72) 발명자 게리트라이슬라벤부르그
미국,캘리포니아94022,로스앨토스,랭톤애비뉴304

심사관 : 오흥수

# (54) 긴명령워드를갖는처리기

요약

VLIW의 코스트/수행은 명령 이슈 래지스터에서 슬롯 수와 래지스터 파일에 대한 포트 수를 감소시킴으로써 개선된다.

감소된 명령수는 기능 유닛의 각 명령의 의도되는 것을 확인하는 명령으로 코드에 따라 선택된 기능 유닛에 기인한다. 상기 명령은 또한 판독 포트가 활성기능 유닛에 할당하는 것을 확인한다.

대표도
도 1

## 명세서

[발명의 명칭]

긴 명령 워드를 갖는 처리기

[도면의 간단한 설명]

제1도는 종래 기술의 VLIW CPU를 도시한 도면.

제2도는 종래 기술의 VLIW 명령 워드의 포맷을 도시한 도면.

제3도는 본 발명에 따른 VLIW CPU를 도시한 도면.

제4도는 본 발명에 따른 VLIW 명령 포맷을 도시한 도면.

제5도는 스위칭 매트릭스(401)를 더욱 상세히 도시한 도면.

제6a도는 레지스터 파일로부터 입력과 출력을 도시한 도면.

제6b도는 입력 스위치 제어로부터 입력과 출력을 도시한 도면.

제7도는 오피코드 디스패치 로직을 도시한 도면.

제8도는 출력 매트릭스(402)를 상세히 도시한 도면.

제9a도는 레지스터 파일 기록 제어 유닛을 도시한 도면.

제10도는 레지스터 파일 기록 제어 유닛을 더욱 상세히 도시한 도면.

제11도는 제10도의 SLOT 블록을 상세히 도시한 도면.

< 도면의 주요부분에 대한 부호의 설명>

101: 레지스터 파일 402: 스위칭 매트릭스

801, 802, 803, 804: 멀티플렉서

[발명의 상세한 설명]

본 발명은 a)서로 병렬로 각각의 동작을 실행하는 적어도 3개의 제1수의 기능 유닛과, b)적어도 2개이거나 제1수보다는 적은 제2수의 명령을 홀딩하고 상기 명령을 기능 유닛에 대해 지정하는 코드를 홀딩 하는 명령 레지스터와, c)명령 레지스터의 명령을 병렬로 수행하기 위해 상기 기능 유닛중 하나를 적당히 실행하는 실행 제어 수단을 구비하는 처리기에 관한 것이다.

그러한 처리기는 A, De Gloria에 의해 " VLIW 구조에 대한 프로그램 가능한 명령 포맷 확장" 으로부터 공지되어 있으며 IEEE 컴퓨터 소사이어티 프레스에 의해 1992년 CompEuro에서 " 컴퓨터 시스템 및 소프트웨어 엔지니어링" 에 개시되어 있다.

VLIW는 " 매우 긴 명령 워드" 를 의미한다. VLIW 구조로된 처리기는 멀티플라이어, 가산기, ALU, 상수 발생 유닛, 메모리 등과 같은 기능 유닛을 포함한다. VLIW 구조를 가진 처리기는 컴파일러에 의해 계획될 명령의 확장시 병렬화를 요구한다.

이는 아래에 따라 실현된다. 각 클락 사이클에서, 다수의 명령은 명령 레지스터에서 병렬로 프로그램 메모리로부터 로드된다. 래지스터 내의 명령은 모든 기능 유닛에 대해 복잡한 오피코드와 오퍼랜드 명세를 포함한다. 상기 레지스터로부터의 명령은 병렬로 기능 유닛에 공급된다.

그러므로, 특별한 응용에서 기능 유닛은 드물 때만 모두 함께 활성화된다. 이러한 수행은 약간의 오피 코드와 오퍼랜드 명세가 모든 기능 유닛에 함께 필요할 때보다는 각 명령에서 제공되는 구조를 유도한다. 멀티플렉서를 사용함으로써,

각각의 명령은 활성화되어 선택된 기능 유닛에 대해 조정된다. 이 방법에서 상기는 수행이 어렵게 되는 동안 명령 메모리 사이즈 상에 저장하는 것이 가능해진다. 명령은 원래 VLIW 구조로서 매우 길지는 않지만, 그 중 하나가 이 " not-so-VLIW" 구조를 호출한다.

이러한 구조에서, 상기 명령 레지스터는 다른 클락 싸이클에서 다른 기능 유닛에 대해 이슈 명령을 가져야만 한다.

이를 위해, 종래 기술에 따른 구조는 명령이 기능 유닛에 전달되고 그 오피코드가 이슈되고 그 오퍼랜드 명세가 선택된 기능 유닛에 선택하는 멀티플렉서를 제공한다.

상기 오퍼랜드 명세는 오퍼랜드 그 자체(중간 모드) 또는 레지스터 파일(레지스터 모드)에 오퍼랜드 어드레스를 포함한다. 후자 모드에 대해서 각각의 기능 유닛은 레지스터 파일에 적어도 하나의 판독 포트를 가진다. 상기 명령이 레지스터 모드에 있을 때 오퍼랜드 명세는 판독 포트의 어드레스 입력에 공급된다. 응답시, 어드레스된 레지스터의 내용은 판독 포트에서 출력이 된다.

동작의 병렬 실행을 유지하기 위해, 상기 레지스터는 상기 레지스터로부터 병렬 판독을 허용하는 판독 포트를 가지고, 멀티-포트 레지스터 파일에 수집될 수 있다. 상기 레지스터 파일은 각 기능 유닛의 각 오퍼랜드 입력에 대해 각각의 포트를 가진다. 예로 각 멀티플라이어 기능 유닛은 2개의 판독 유닛에 각각 접속되어 있다. 즉, 대비 시에 명령에 의해 지정된 상수를 공급하는 " 대비" 유닛은 어떤 판독 포트에 접속되지 않았다.

적분 처리기에서, 레지스터 파일은 큰 영역의 실리콘을 요구하며, 비교적 전력 소모가 작다.

따라서, 본 발명의 목적은 실리콘 영역을 감소시키고 처리기의 전력 소모를 감소시키기 위한 것이다.

본 발명을 따른 처리기는 명령 레지스터에서 명령에 관계한 오퍼랜드를 기억하기 위한 멀티포트 레지스터 파일, 오퍼랜드에 근접하여 병렬로 제공하기 위한 제3수의 판독 포트를 구비하며, 상기 판독 포트의 제3수는 제2수의 기능 유닛으로 이루어지는 기능 유닛의 서브세트에 의해 레지스터 파일로부터 요구된 최소수의 오퍼랜드와 적어도 동일하며, 상기 제3 수는 기능 유닛과 함께 요구된 전체수의 오퍼랜드보다 작고, 또한, 상기 처리기는

e)상기 판독 포트에 대해 병렬 액세스로 명령을 실행하기 위해 발생된 기능 유닛을 선택적으로 제공하는 판독 액세스 제공 수단을 구비하는 것을 특징으로 한다.

상기 레지스터 파일의 영역과 전력 소모는 포트수에 비례하며, 포트 수가 감소되는 경우에 요구된 영역과 전력 소비를 감소시키는 것이 가능하다. 병렬로 사용된 포트의 수는 병렬로 시작하기 위해 실행되는 명령의 수에 의해 결정되며, 모든 기능 유닛과 함께 요구된 오퍼랜드의 전체수 보다는 작다.

기능 유닛과 래지스터 파일에 대한 포트 사이에 프로그램가능한 접속을 제공함으로써, 래지스터 파일에 대한 포트의 수는 처리기에 의해 성취할 수 있는 병렬화의 양 감소없이 포트의 전체수 보다 작다.

본 발명의 실시예에서 판독 포트는 판독 포트의 그룹으로 구성되며, 적어도 하나의 특별한 그룹은 적어도 2개의 판독 포트를 구비하며, 상기 판독 액세스 제공수다은 그룹마다 기능 유닛에 제공된 판독 포트를 선택하기 위해 배열되어 있으며, 상기 기능 유닛중 적어도 하나는 각각의 오퍼랜드를 동시로 수신하기 위한 적어도 2개의 입력을 가지며, 상기 입력은 특별한 그룹이 상기 기능 유닛중 하나에 대해 선택될 때 매번 특별한 그룹으로부터 선정된 판독 포트 각각에 연결되어 있다. 프로그램에서의 명령이 상기 그룹만을 확인하는 것을 필요로 하면 상기 명령의 작은 포트는 각 포트가 명령에서 개별적으로 확인되는 경우 필요한 것보다는 포트 선택에 의해서 점령된다.

본 발명의 실시예에서 명령 래지스터는 상기 명령을 이슈하기 위한 각각의 이슈 슬롯의 제2수로 구성되며, 각각의 이슈

슬롯은 적어도 하나의 판독 포트의 각 그룹에 대응하고, 특별한 기능 유닛은 상기 특별한 이슈 슬롯으로부터 명령을 실행할 때 매번 특별한 이슈 슬롯에 대응하는 그룹으로부터 판독 포트나 포트에액세스 되어 제공된다. 이 방법에서, 이슈 슬롯의 분리 확인이 되지 않고 상기 포트 그룹은 기능 유닛에 접속될 필요가 있으며, 작은 명령은 상기 명령이 이슈 슬롯과 그룹의 분리 명령을 포함하는 경우보다 선택하는데 충분하다는 것을 의미한다.

본 발명의 실시예에서 각각 특별한 그룹은 제 4 수의 선정된 포트를 포함하며, 제4수는 상기 기능 유닛중 하나에 의해 요구된 오퍼랜드의 최대 수와 동일하다. 이에 있어 각 그룹은 어떤 기능 유닛을 공급하는 것이 가능하며, 각 그룹에서 포트의 수는 소정 기능 유닛이 래지스터 파일로부터 많은 오퍼랜드를 실제로 요구하지 않을 때 조차도, 어떤 기능 유닛에 의해 필요한 최대수의 오퍼랜드와 동일하다.

예로, 요구된 오퍼랜드의 최대 수가 2이면(멀티플라이어의 경우) 기능 유닛에 제공된 그룹내의 포트수는 2이다. 이러한 포트의 수는 또한 포트가 전혀 필요치 않은 곳에서, 상수를 공급하는데 배열된 기능 유닛에 이용 가능하다.

본 발명의 다른 실시예에서 각각 특별한 이슈 슬롯은 판독 포트에서 어드레스된 오퍼랜드에 액세스를 제공하는 특별한 이슈 슬롯에 대응하는 그룹의 판독 포트의 각각의 어드레스 입력에 연결된 같은 소정수의 어드레스 필드를 포함하며, 상기 특별한 이슈 슬롯으로부터 명령을 실행하는 기능 유닛과 무관하게 어드레스된 오퍼랜드를 요구한다. 따라서, 어드레스 필드는 래지스터 파일에 직접 전달되며, 포트에 액세스될 기능 유닛과는 무관하고, 상기 기능 유닛이 포트에 사용될 때 조차도 무관하다.

이는 래지스터 파일에 대해 어드레스 디코딩 선택이 상기 이슈 슬롯에서 명령을 실행하는 기능을 유닛으로 이루어지기 전에 시작되기 때문에 동작을 가속시킨다.

양호하게도, 상기 명령 래지스터의 구성은, 고정된 포맷, 즉, 상기 명령 래지스터의 각 지정부의 동작의 기능 (오피코드, 오퍼랜드 어드레스, 기능 유닛이 명령등으로 실행되는 것을 확인하는 코드)은 같으며 명령을 실행하는데 사용된 기능 유닛과도 무관하며, 기능 유닛과 명령의 수에 대해 최소한의 즉, 병렬로 실행하는데 선택된 기능 유닛과 명령의 수에 대해 최소한의 즉, 병렬로 실행하는데 선택된 기능 유닛과 명령 그 자체의 조합과도 무관하다. 이는 어드레스 디코딩은 명령이 디코드되기 전에 시작되며, 또한, 다른 이유로, 명령 래지스터와 상기 명령을 통괏키니는 다른 유닛간에 제공될 스위칭 회로를 간단히 한다는 유리한 장점을 갖고 있다.

본 발명의 실시예에서 상기 처리기는 상기 명령중 하나를 실행하기 위해 발생될 때, 상기 기능 유닛중 적어도 하나가 상기 명령중 하나를 이슈하는 이슈 슬롯내의 어드레스 필드로부터 정보를 수신하도록 배열되어 있으며, 상기 기능 유닛중 적어도 하나는 실행하는 동안 오퍼랜드로서 정보를 사용하다. 따라서 오퍼랜드는 또한 중간 오퍼랜드를 요구하는 동작에 대해 중간 오퍼랜드(즉, 상수)로서 사용된다. 오피코드가 디코드되기 전에, 또는 중간 오퍼랜드를 요구하는 동작에 대해서 중간 오퍼랜드(즉, 상수)로서 사용된다. 오피코드가 디코드되기 전에, 또는 적어도 오피코드가 의도되어 기능 유닛의 확인이 디코드 되기 전에는, 오퍼랜드가 상수인지 어드레스인지를 알지 못한다. 그러므로, 동시에 오퍼랜드가 기능 유닛에 공급되는 바와 같이, 상기 래지스터 파일은 어드레스로서 오퍼랜드를 디코딩하기 시작한다. 디코딩을 위한 대기는 천천히 하향으로 처리된다. 상기 오퍼랜드가 중간이면, 래지스터 파일로부터 액세스 가능한 데이타가 무시된다.

본 발명의 실시예에서 처리기, 멀티포트 래지스터 파일은 적어도 하나의 기록 포트를 구비하며, 상기 처리기는 병렬로 명령을 실행하는데 발생된 기능 유닛에 대해 기록 포트나 포트에 액세스를 선택적으로 제공하는 기록 액세스 제공 수단을 구비하며 그의 결과를 형성한다. 이 방법에서, 기능 유닛에 존재하는 것보다 래지스터 파일에 대해 더적은 기록 포트를 사용하는 것이 가능하며, 영역과 전력 소비를 감소시킬 수 있다. 기록 포트의 선택은 이슈 슬롯으로부터 복잡한 선택

없이 계획된다. 이는 오피코드 이슈와 상기 결과 발생간에 이용가능한 소정 시간이 존재하기 때문에 처리 속도를 감소시킬 필요가 없다.

본 발명의 실시예에서 기록 액세스 제공 수단은 기능 유닛중 하나의 출력을 결합하기 위해 배열되며, 기능 유닛이 명령을 실행한 후 잠복 주기와 기록 포트에 상기 발생되는 어드레스 필드의 내용과 조합한다. 이 방법에서 상기 래지스터 파일에 대한 기록 포트의 수는 기능 유닛의 수보다 작게 유지된다.

본 발명의 실시예에서 명령 래지스터는 명령을 이슈하기 위한 각각의 이슈 슬롯의 제2수로 구성되며, 반면 각각의 이슈 슬롯은 발생된 어드레스 필드와 같은 수를 포함하며, 기록 액세스 제공 수단은 기록 포트 또는 포트의 어드레스 입력 또는 입력에 어드레스 내용을 루팅하기 위해 배열되어 있다.

본 발명의 실시예에서 상기 기록 액세스 제공 수단은 상기 기능 유닛이 명령을 실행한 후 잠복 주기와 기록 포트에 발생되는 어드레스 필드 내용을 조합하는, 기능 유닛중 하나의 출력을 결합하기 위해 배열되어 있다. 기능 유닛이 동작을 실행할 때 동작의 시작과 결과가 이용 가능하게 되는 모멘트 간에 하나의 클럭 사이클보다 더큰 잠복이 종종 존재한다.

본 발명의 실시예에서 래지스터 파일을 액세스하는 기록은 적어도 2개의 기록 포트를 갖고 있다. 즉, 기록 선택 수단은 상기 재할당후의 클럭 사이클에 대해 기능 유닛중 하나에 특별한 기록 포트를 재할당하기 위해 배열된 재-할당 수단을 구비하며, 상기 기능 유닛중 하나의 실행 결과를 이용할 수 있으며, 특별한 기록 포트에 기능 유닛중 하나의 출력을 결합하고 상기 클럭 사이클 동안 특별한 기록포트에 발생되는 어드레스 필드 내용을 전달하기 위해 배열된다. 본 실시예에 따르면 처리기는 이용가능한 결과의 트랙을 유지하며 결과가 기록되기전에 기록 포트의 할당을 계획한다.

본 발명의 다른 실시예에서 상기 재할당 수단은 이 특별한 기록 포트가 클럭싸이클에 대해 기능 유닛중 하나에 이전에 할당되지 않는 억제하에 기록 포트로부터 특별한 기록 포트를 선택하기 위해 배열된다. 이 방법에서 대립은 현재 결과를 제공하는 기능 유닛 사이에서 회피할 수 있으며, 동작 발생을 통해 이들 결과는 충분한 시간을 갖고 시작된다. 본 발명의 이들 및 다른 장점들은 첨부한 도면을 참고로 더욱 상세히 설명된다.

VLIW CPU는 여러가지 응용에서 사용된다. 즉, 슈퍼 컴퓨터에서 작동 스테이션까지 및 작동 스테이션에서 처리기를 표시하거나 프로그램 가능하게 할때 조차도, 개인용 컴퓨터, 상용의 비디오 또는 오디오로 사용된다.

제1도는 종래 기술의 VLIW 장치를 도시한다. 명령은 명령 메모리(102)에서 명령 이슈 래지스터 IIR(Instruction Issue Register)까지 로드된다. 각 클럭 싸이클에서, 새로운 매우 긴 명령은 명령 이슈 래지스터 IIR로부터 전송된다.

이러한 명령은 VLIW CPU에서 각각의 기능 유닛(CONTROL, CONST, ALU1, ALU2, MUL, EPU, MEM)에 대해 이슈 슬롯을 포함한다.

상기 VLIW 머신은 기능 유닛의 유용한 조합을 포함하며, 예에서는 하나의 그러한 조합만 도시되어 있다. 여기에서는 요구된 동작에 따라 여러 다른 형태의 약간의 기능 유닛이 존재한다.

상기 기능 유닛에 대한 오퍼랜드는 배분된, 다중 포트 래지스터 파일(101)로 부터 판독된다. 기능 유닛으로부터의 결과는 또한 이 파일에 기록된다.

상기 이슈 슬롯(issue slot)은 기능 유닛상에서 현재 클럭 싸이클로 시작된 동작을 지정한다. 제2도는 CONTROL 동작, CONST 동작, ALU1 동작, ALU2 동작, MLL 동작, FPU 동작, MEM 동작을 포함하는 VLIW 명령(201)을 도시한다. 한편 VLIW 명령은 제1도의 VLIW CPU에서 각 기능 유닛에 대해 하나의 이슈 슬롯을 포함한다.

202는 CONTROL 동작을 지정하는 이슈 슬롯 내용을 도시한다. 이러한 이슈 슬롯은 CONTROL 오피코드와 소스 래지스터(1) 및 소스 래지스터(2)인 Rsrc1 및 Rsrc2, 즉 2개의 래지스터 명세를 포함한다.

203은 CONST 동작을 지정하는 이슈 슬롯의 내용을 도시한다. 이 이슈 슬롯은 일정값과 목적 래지스터인 래지스터 명세, Rdest 를 포함한다.

204는 ALU 동작을 포함하는 이슈 슬롯 내용을 도시한다. 상기 이슈 슬롯은 ALU 오피코드와 래지스터 명세서, Rsrc1, Rsrc2, Rdest 를 포함한다.

205는 MLL 동작을 포함하는 이슈 슬롯의 내용을 도시한다. 상기 이슈 슬롯은 MUL 오피코드와 3개의 래지스터 명세, Rsrc1, Rsrc2, 및 Rdest 를 포함한다.

206은 FPU 동작을 포함하는 이슈 슬롯의 내용을 도시한다. 상기 이슈 슬롯은 FPU 오피코드 및 3개의 래지스터 명세, Rsrc1, Rsrc2, 및 Rdest 를 포함한다.

207은 MEM 동작을 포함하는 이슈 슬롯의 내용을 도시한다. 상기 이슈 래지스터는 MEM 오피코드와 2개의 래지스터 명세, Rsrc1, Rsrc2 또는 Rdest 를 구비한다.

다시 이들 이슈 슬롯의 내용은 보기화되고 기능 유닛의 유용한 구성에 대해 조절된다.

대부분 종래 기술의 머신에서, 동작은 각 싸이클에서 모든 기능 유닛상에서 시작된다. 싸이클 " i" 에서 시작된 동작은 하나의 싸이클로 완성되거나 또는 여러 싸이클로 완성된다.

완성은 목적 래지스터에서의 동작 결과를 기록함으로써 입증된다. 결과없는 동작에 대해서, ('기억' 동작과 같은) 완성은 동작과 관련된 상태 변화가 발생되는 시간이다.

제1도의 대부분의 기능 유닛은 CONST 와 같은 것이 존재한다(상수 발생 유닛). 이러한 유닛은 입력된 상수를 발생하여 목적 래지스터로 집어넣는다. 상기 ALU, MUL 및 FPU 유닛은 하나 이상의 장치로 대수적, 논리적인 시프트 동작을 수행하며 목적 래지스터에서 단일의 결과를 발생시킨다.

상기 CONTROL 및 MEM 유닛은 다소 다르다.

상기 CONTROL 유닛은 명령이 이슈된 순차를 결정한다.

NOP(동작안되는)는 CONTROL 유닛에 이슈되며, 명령은 명령 메모리로부터 순차적 순서로 이슈된다. CJMPF 또는 CJMPT (Conditional JuMP False 및 Conditional JuMP True) 동작은 제어 유닛상에 이슈되며, Rsrc1 래지스터 내용은 참, 즉, 블런(boolean) 값을 가지는 것으로 설명되며, Rsrc2 래지스터 내용은 지정된 상태가 부합되는 경우에만, 다른말로 하면, 명령 이슈가 순차적으로 진행될때만 명령 이슈가 계속된다.

상기 MEM 유닛은 로드 및 기억 동작을 수행한다.

한편, 상기 래지스터 파일과 시스템 주 메모리 사이에서 데이타 워드를 이동한다. 로드 동작은 로드될 데이타 워드의 주 메모리에서 어드레스로서 Rsrc1 을 사용하고 Rdest는 로드된 값이 기억되는 래지스터를 식별한다. 기억 동작은 어드레스로서는 Rsrc1의 내용이 기억될 값으로서는 Rsrc2의 내용을 사용한다.

물론, 많은 변화의 로드 및 기억 동작이 존재한다. 따라서 로드 명령은 Rsrc2 를 요구하지 않으며 기억 명령은 Rdest 를 요구하지 않고, 이슈 슬롯은 2 래지스터 필드를 포함하는 것만을 필요로한다.

VLIW 머신상에서 동작하기 위한 소프트웨어 프로그램에대한 순서로, " 병렬의 미세 이득" 또는 " 병렬의 명령 레벨" 변환이 발견된다. 이는 ANSIC 와 같은 종래의 고-레벨 프로그래밍 언어를 VLIW 머신에 대한 명령으로 변환하는 컴파일러에 의해 수행된다. VLIW 머신에 사용하기 위한 컴파일러는 죤.알.일씨 등에 의해 MIT 프레스 1985년 ISBN 0-262-5034-X의 BULLDOG: VLIW 구조용 컴파일러에 기술되어 있다.

피크 처리비로 제1도의 VLIW 를 동작시키기 위해, 1 CONTROL, 1 CONSTANT, 2 INTEGER ALU, 1 INTEGER MULTIPLY, 1 FLOATING POINT 및 1 MEMORY 동작은 매 싸이클 마다 이슈된다.

고 레벨 언어로 표시된 실제 프로그램의 성질로 인해, 이 피크 수행시 유지되는 적당한 믹스 동작을 알기 어렵다. 컴파일링 후에, 주어진 클럭 싸이클에서 병렬로 수행될 동작 셋은 이용가능한 기능 유닛 형태와 어울리지 않는 믹스 형태로 이루어진다. 어떤 경우에, 프로그램은 머신에서 유닛 번호보다 병렬 동작이 적은 페이스(phases)를 통해 진행한다.

이는 여러 문제점을 발생시킨다.

처음에, 래지스터 파일 포트는 하부에서 활용된다.

상기 실리콘 영역과 중앙 래지스터 파일의 전력 소비는 포트의 전체 수에 비례한다. 따라서 그러한 포트의 활용이 높게 되는 것이 중요하다.

두번째로, 명령 대역폭은 VLIW CPU를 유지하거나 피크 수행이 높게될 필요성이 있다. NOP 코드를 포함하는 빈 슬롯(empty slots)은 이 대역폭에 기인한다. 상기 대역폭은 I-Cache (명령 캐시) 크기 및 버스들의 코스트 및 다른 메모리 시스템 성분으로 직접 변형된다.

세번째로, VLIW에 대해 변형된 프로그램용 코드의 크기는 RISC 형 CPU에 대해 변형될 때 같은 프로그램 크기보다 더 크다. VLIW의 수행이 RISC 보다 높을 때 조차도, VLIW의 코스트 수행비는 RISC 보다 적다.

제3도는 본 발명에 따른 VLIW CPU 를 도시한다.

IIR은 머신내의 기능 유닛에 존재하는 이슈 슬롯보다 적다.

예로 7개의 기능 유닛을 가진 머신에 대해서는 3개의 이슈 슬롯이면 충분하다. 예로, 주어진 수에 대한 이슈 슬롯의 수와 기능 유닛의 형태는 평균 수행 및 코스트 사이에서 평균을 취하며, N 유닛을 가진 머신은 N/4 및 2N/3 이슈 슬롯 사이라는 것을 예측할 수 있다.

상기 CPU는 다수 그룹의 판독 포트($(R1_a, R_{1b})$, $(R_{2a}, R_{2b})$, $(R_{3a}, R_{3b})$)를 가지는 래지스터 파일(403)을 구비한다. 적당한 래지스터 파일의 예는 인텔 iWARP 및 i960 칩에서 알 수 있다. IIR의 각 슬롯에 대해서 한 그룹의 판독 포트가 존재한다. 상기 판독 포트는 래지스터 파일에서 래지스터 내용을 판독하기 위해 존재한다. 각 그룹내의 포트수는 임의 기능 유닛에 의해 필요한 소스 래지스터의최대수와 같다. 이경우, 기능 유닛은 2개의 소스 래지스터 이상일 필요가 없으며, 따라서 판독 포트의 각 그룹은 2개의 판독 포트를 구비한다. 따라서 3개 그룹의 2개의 판독 포트, 소정의 6개의 판독 포트가 존재한다.

입력 스위치 제어(902)에 의해 제어된, 스위칭 매트릭스 (401)는 판독 포트를 기능 유닛의 입력에 접속시키는 것을 결정한다.

상기 래지스터 파일은 또한 다수 그룹의 기록 포트(W1, W2, W3, W4)를 가진다. 각 IIR 슬롯에 대해서는 적어도 한 그룹의 기록 포트가 존재한다. 각 그룹에서 기록 포트의 수는 어떤 기능 유닛에 의해 필요한 목적 래지스터의 최대수이다.

이 예에서의 기능 유닛은 하나의 목적 래지스터에서 보다 더 필요하지는 않다. 그러므로, 각 그룹의 기록 포트에서 하나의 기록 포트가 존재한다.

제2스위칭 매트릭스(402)는 기능 유닛의 출력이 기록 제어 유닛(RFWC)의 제어하에서 기록 포트에 접속한다.

제4도는 새로운 VLIW 명령(410)의 포맷을 도시한다.

이 명령은 동작용으로 3개의 슬롯을 가진다. 각각의 슬롯은 소정 형태의 동작을 포함한다. 전에 상술한 바와 같이, 각각의 기능 유닛에 대해 6개의 동작 형태가 존재한다. 즉, CONTROL, CONST, ALU, MUL, FPU, MEM 이다.

상기 이슈 슬롯은 여러면에서 이전의 이슈 슬롯과는 다르다.

첫째, 이슈 슬롯은 모두 같은 길이를 가진다. 상기 이슈 슬롯 내용의 포맷은 CONTROL, CONST, ALU, MUL, FPU, MEM 유닛 각각에 대해 420, 430, 440, 450, 460, 470 으로 도시된다.

상기 슬롯의 일정 길이는 CONTROL 형 동작(420)의 단부에서 어떤 블랭크 공간의 삽입을 위해 필요하다.

두번째로, 상기 모든 이슈 슬롯은 기능 유닛이 동작을 실행하는 것을 확인하는 어떤 소정 방법을 가진다. 도면에서, 모든 이슈 슬롯은 동작에 사용키 위해 기능 유닛을 확인하는 UNIT ID 필드를 가지며, 이 UNIT ID 필드는 3개의 비트 폭만을 필요로 한다. 기술에 숙련된 사람은 유닛을 확인하는 다른 방법으로 설계할 수 있다. 하나의 대안은 유닛을 의미하는 오피 코드를 가진다. 다른 대안은 유닛 형태를 의미하는 오피 코드를 가지며 하드웨어는 유닛의 형태를 할당한다.

제3도의 VLIW CPU 를 참고로, 각 싸이클 클럭에서, 3개의 동작 명령이 IIR 로 부터 이슈되다. 상기 이슈 슬롯에서 소스 래지스터 필드(Rsrc1, Rscr $_2$ 2)는 래지스터 파일 판독 ($R1_a$, $R_{1b}$), ($R_{2a}$, $R_{2b}$), ($R_{3a}$, $R_{3b}$))를 어드레스 하는데 사용된다. 유닛 확인 필드를 근거로하여, 상기 오피코드 제어는 목적 기능 필드에 대한 루트가 된다.

상기 모든 이슈 슬롯의 목적 래지스터 필드는 각 기능 유닛의 잠복(lalenecy)를 알려주고 결과가 발생될 때 트랙을 유지하는 기록 제어 유닛(RFWC)에 대한 루트가 된다. 이 기록 제어 유닛은 래지스터 파일내의 기록 결과에 적당한 버스들을 할당한다. 3개의 이슈 슬롯 머신에 대해, 다른 잠복으로 인해 발생하는 트래픽 버스트를 조정하기 위해 3개 이상의 기록 버스가 존재한다.

위의 예는 단지 하나의 실시예이다. 설계자는 예측된 작동 로드를 기초로하여 기능 유닛의 수와 형태를 선택하는 것이 필요하다. 상기 설계자는 또한 IIR에서 많은 이슈 슬롯의 수와 코스트/수행을 최적화 하기 위한 많은 기록 버스들을 선택하는 것을 필요로 한다.

상기 양호한 실시예의 구조는 호환성이 있다. 만약, 높은 수행 CPU 를 설치하는 것을 요구하면, 첨가적으로 FPU, ALU, CONST 를 포함하여 IIR 내의 슬롯수를 4 로 증가시킬 수 있다. 상기 새로운 머신은 구형의 유닛이 새로운 머신내에 있는 같은 형태의 유닛에 대해 필드값 맵을 확인하는 한, 구형의 머신으로부터 2 진 유닛 ID 를 계속 사용한다. 상기 구형 머신에 대한 코드 영상은 프로그램 로드 단계 동안, 첨가의 NOP 동작을 가산함으로써 새로운 머신으로 동작할 수 있다. 상기 유닛 ID 필드들은 새로운 머신의 경우에 첨가적으로 제로를 채워넣을 때만 필요하며 구형 머신보다는 더 넓은 필드를 가진다.

그러므로, 코드의 재컴파일링, 재스케쥴링, 재어셈브리가 필요치 않으며, 모든 호환성 이슈들은 프로그램 로드 시간에서 해결될 수 있다.

제5도는 래지스터 파일 판독 포드(($R1_a$, $R_{1b}$), ($R_{2a}$, $R_{2b}$), ($R_{3a}$, $R_{3b}$)) 및 기능 유닛 사이에서 스위칭 매트릭스 실행의 예를 도시한다. 당업자는 이 스위칭 매트릭스에 대해 몇몇의 대안적 실시예를 설계할 수 있다.

각각의 래지스터 파일 판독 포트는 대응하는 기능 유닛 입력 포트에 관련된 모든 멀티플렉서(801)의 입력에 접속한다.

즉, 포트 $R_{1b}$ 는 모든 기능 유닛의 모든 B' 입력 멀티플렉서에 접속한다. 명령이 이슈 상태에 도달하면, 상기 IIR은 상기 명령에 대해 2 진 코드를 포함한다. 명령이 이슈되자마자, 각 이슈 슬롯의 Rsrc1 및 Rsrc2 필드는 래지스터 파일 판독 포트(($R1_a$, $R_{1b}$), ($R_{2a}$, $R_{2b}$), ($R_{3a}$, $R_{3b}$))에 대해 판독 어드레스로서 사용된다. 이는 총 6개의 래지스터가 판독되게 한다.

이들 래지스터 모두는 실제적으로는 필요치 않다. 예로, 슬롯 1은 일정 형태 동작을 이슈할 수 있으며, 이경우 래지스터 파일 어드레스는 무의미하게 된다. 그러므로, 이슈된 Rsrc1, Rsrc2 또는 모두가 필요한 동작이면, 적당한 래지스터 값이 기능 유닛의 모든 MUX 입력, 특히, 동작이 실행되는 기능 유닛의 MUX에 이용할 수있다.

제6a도는 판독 포트와 어드레싱 포트를 가진 래지스터 파일(403)을 도시한다. 제6b도는 입력 스위치 제어 로직 블럭(902)을 도시한다.

도시한 바와 같이, 래지스터 파일(403)에 대한 판독 어드레스 입력(($AR1_a$, $AR_{1b}$), ($AR_{2a}$, $AR_{2b}$), ($AR_{3a}$, $AR_{3b}$))은 모두 6 비트 폭을 가지며, 명령 이슈 래지스터 IIR의 래지스터 소스 필드, 즉, IIR. 이슈 1.Rsrc1, IIR. 이슈 1, Rsrc2, IIR. 이슈 2, Rsrc1, IIR. 이슈 2.Rsrc2, IIR. 이슈 3.

Rsrc1, 및 IIR. 이슈 3. Rsrc2 로부터 공급된다. 이들 어드레스를 근거로 하여, 오퍼랜드는 기능 유닛에서 판독 포트(($R1_a$, $R_{1b}$), ($R_{2a}$, $R_{2b}$), ($R_{3a}$, $R_{3b}$))에 필요한 바와 같이 공급된다.

기록 어드레스 입력(W1.wa, W2.wa, W3.wa, W4.wa)) 및 기록 가능 입력(W1.we, W2.we, W3.we, W4.we)은 제9a도의 RFWC 로부터 공급된다. 동작 결과는 제 8도의 멀티플렉서(801), (802), (803), (804)로부터 W1, W2, W3, 및 W4 로 공급된다.

입력 스위치 제어(902)에 대한 어드레싱 입력, 즉, IIR. 이슈 1, UNITID, IIR. 이슈 2. UNITID, IIR. 이슈 3. UNITID는 명령 이슈 래지스터의 유닛 id 필드로부터 공급된다.

이들 어드레싱 입력으로부터, 입력 스위치 제어(902)는 2비트 신호들 $M_0C$, $M_1C$, $M_2C$, $M_3C$, $M_4C$, $M_5C$, $M_6C$ 을 결정한다. 이들 멀티플렉서 제어 신호들은 래지스터 파일로부터 판독 버스들이 각각의 기능 유닛 입력에 공급되는 오퍼랜드를 선택하기 위해, 멀티플렉서(801)에 공급된다.

제7도는 모든 기능 유닛에 대해 모든 오피코드를 발생하는 오피코드 형성과 전송 로직의 예를 도시한다.

상기 입력 스위치 제어(902)는 표준 셀 로직, PLA, 또는 ROM 조사표와 같은, 여러 공지된 방법으로 실행되는 기능적 로직 블럭이다. 논의를 목적으로, 아래의 유닛 ID 할당은 다음과 같이 취한다. 즉, 유닛 0=제어 ; 1=CO, 2=ALU1, 3=ALU2, 4=MUL, 5=FPU, 6=MEM, 또한 1의 멀티플렉서 제어 필드는 왼쪽의 대부분의 입력을 선택하며, 2는 중간 입력을 선택하고, 3은 오른쪽의 대부분 입력을 선택한다. 이들로부터 변환 표는 유닛 ID 할당을 어떤 조합으로부터 멀티플렉서 제어 필드의 필요한 조합으로 변환하는 구성으로 된다. 이들 표로부터 소정 예의 입력 및 출력 벡터는 아래에 주어져 있다.

이슈 1 이슈 2 이슈 3 출 력 주의

유닛 1 유닛 4 유닛 6 $M_1C=1$ $M_4C=2$ $M_6C=3$ 다른것=0 (1)

유닛 0 유닛 2 유닛 3 $M_0C=1$ $M_2C=2$ $M_3C=3$ 다른것=0 (2)

주의(1) 이슈 슬롯(1)은 CONST(상수) 유닛 동작을 이슈한다. 따라서 CONST 유닛은 입력을 갖고있지 않으며, 멀티플렉서 제어도 하지않는다. 그러므로, 제7도의 오피코드 루팅 로직은 CO 유닛 오피코드 발생 멀티플렉서를 제어하기 위해 $M_1C$ 신호를 필요로 한다. 이슈 슬롯(2)은 MUL 동작을 이슈한다.

따라서 $M_4C=2$는 $R_2$ 포트값을 선택하기 위해 설정될 MUL (멀티플라이어) 유닛중 2개의 멀티플렉서를 야기한다. 모든 다른 멀티플렉서들은 위치 0 으로 설정되며, 오피코드 상에서 전송 로직을 특별하게 사용한다. 이슈 슬롯(3)에 의해 이슈된 유닛(6)동작이 로드되면, MEM 유닛에 루트된 오피코드는 $R_{3b}$ BUS의 값을 무시하게 하며, 이에 따라서 $Rsrc_2$ 오퍼랜드는 로드 동작에 의해 사용되지 않는다.

주의 (2) : 동작이 이슈된 유닛에 대한 모든 멀티플렉서들은 그들의 동작을 이슈하는 이슈 슬롯에 대응하는 래지스터 포트를 선택한다. 이는 입력 스위치 제어 블럭의 로직 식을 구성하기 위한 일반적인 규칙이다.

래지스터 판독 포트값 스위칭에 대한 대응하게, 제7도의 회로는 개별의 기능 유닛 오피코드를 발생시킨다. 상기 같은 멀티플렉서 제어 신호들은 제5도에서 처럼 이 회로에 사용된다.

이 규칙은 이슈 슬롯 X 가 Y 유닛에 대한 동작을 포함하는 경우이며, 유닛 Y 는 멀티플렉서 제어 신호값 X 를 수신하며, 여기서 Y'S 멀티플렉서는 그 동작용 오피코드로서 이슈 슬롯 X 내의 오프코드 필드를 선택한다.

아무것도 이슈되지 않은 어떤 유닛은 멀티플렉서 선택 신호를 수신한다. 이는 상기 유닛에 대해 비동작(NOP) 오피코드 발생을 초래한다. 이는 비동작 코드(도시하지 않음)를 수신하는 멀티플렉서에 특별한 입력을 가산하거나 오피코드 멀티플렉서 유닛의 로직 식을 변화시킴으로써 수행된다.

제7도에서, CONST 유닛이 특별하게 된다. 이는 모조 유닛이 된다. 이는 오피코드에 집중되는 2 진 값을 출력하는 결과를 발생하며, 즉, CO 유닛 동작이 이슈되는 이슈 슬롯의 Rsrc1 및 Rsrc2 필드를 발생시킨다.

당업자는 제5도 내지 제7도에 대해 대안적인 실시예를 설계할 수 있다. 예를들어, 고수행 머신은 적당히 파이프라인된다. 그런경우 파이프라인 래지스터 및 바이패스 회로는 이러한 베이직 설계에 가산된다. 또한 동작시 UNITID 및 오피코드 필드들은 필드 사이에서 이동 가능 분리부를 갖는 단일 필드로 조합된다. 그러한 이동 가능 분리는 다른 기능 유닛에 대해 요구된 다른 수의 오피코드 비트를 수용할 수 있다. 예로, ALU는 MUL 또는 MEM 유닛보다 더 많은 오피코드 비트를 필요로 하게 된다. 따라서, UNITID 필드들은 더짧게 되며 MUL 및 MEM 유닛에 대한 것보다 ALU에 대해 오피코드 필드가 더 길어진다.

그경우, 입력 스위치 제어 블럭은 유닛 ID 를 표시하는 비트의 가장 긴 가능한 셋을 수신하며, 동시에 오피코드 전송 로직은 오피코드를 표시하는 비트의 가장 긴 가능한 셋을 수신한다.

다른 가능한 변화가 존재한다. UNITID 필드는 사용될 필요성이 없다. 상기 하드웨어는 기능 유닛 형태와 할당 유닛 ID 필드 동적으로 오피코드가 진행되는 맵에 표를 사용할 수 있다.

제8도는 출력 매트릭스(402)를 상세히 설명한다.

상기 기능 유닛은 출력 래지스터(806, 807, 808, 809, 810)에 수행된 동작의 결과를 기록한다. 상기 래지스터(806, 807, 808, 809, 810)은 32비트 버스 입력을 가진 기록 포트 멀티플렉서 (801, 802, 803, 804)에 접속된다. 상기 CONST 유닛 오피코드를 직접 멀티플렉서(801, 802, 803, 804)에 접속되며, 따라서 제7도에 도시한 바와 같이 CONST 유닛에서 오피코드가 이용가능한 결과 간에 시간이 경과하지 않는다. 멀티플렉서 (801, 802, 803, 804)는 제어 입력 $W_1$MC, $W_2$MC, $W_3$MC 및 $W_4$MC에 근거하여, 그들의 입력을 그들의 출력 W1, W2, W3, W4에 대한 루트를 선택한다. 상기 출력 W1, W2, W3, W4는 각각 32비트이며 래지스터 파일(402)의 기록 포트에 접속된다.

제어 입력 $W_1$MC, $W_2$MC, $W_3$MC, $W_4$MC는 래지스터 파일 기록 제어 유닛(RFWC)에 의해 공급된다.

제9a도는 래지스터 파일 기록 제어 유닛(RFWC)를 도시한다. RFWC의 입력은 IIR 로부터 유닛 id's(이슈 1 unitid, 이슈 2. unitid, 이슈 3. unitid) 및 목적 래지스터 어드레스(이슈 1. rdest, 이슈 2. rdest, 이슈 3. rdest)이다.

상기 RFWC는 출력으로서 4개의 멀티플렉서 제어 신호($W_1$MC, $W_2$MC, $W_3$MC, $W_4$MC)와, 래지스터 파일을 어드레싱하기 위한 4개의 기록 어드레스(W1.wa, W2.wa, W3.wa, W4.wa) 및 래지스터 파일에 직접 4개의 기록 포트 기록 가능 신호(W1.we, W2.we, W3.we, W4.we)를 발생시킨다.

제9b도는 유닛 ID'S 및 오피코드를 실행하는 RFWC/WE 인코더를 도시하며, RFWC 용 기록 인에이블 입력을 발생시킨다.

상기 인코더는 ROM, PLA 또는 로직일 수 있다. 상기 인코더는 다음의 동작, 즉, COnST, ALU, MUL, FPU 및 MEM 로드에 대한 결과를 갖는 동작을 표시하는 " 1" 을 발생시킨다. 상기 인코더는 CONTROL 및 MEM에 대한 결과를 기억하지 않는 것을 표시하며 " 0" 을 발생시킨다.

제10도는 RFWC 블럭에 대한 실행의 예를 도시한다.

이 실행은 2개의 서브블럭, SLOT 블럭(1001) 및 LAT 블럭 (102)를 구비한다. 또한 SLOT의 4개의 로우와 4개의 컬럼이 존재한다. 4개의 로우는 위의 표1에서 보는 바와 같이, 기능 유닛의 4개의 가능한잠복에 대해 필요하다. 상기 기능 유닛의 잠복은 상기 유닛상의 동작 시작과 기록 멀티플렉서 입력에서 동작의 결과 이용성 사이의 클럭 싸이클 수이다.

표1

| 유닛 | 잠복 |
|---|---|
| CONTROL | N/A |
| CONST | 0 |
| ALU1, ALU2 | 1 |
| MEM | 1 |
| MUL | 2 |
| FPU | 3 |

각 유닛 즉, MUL에 대해, 상기 대응한 입력 기록 인에이블 신호가 TRUE 이며, 잠복은 RFWC의 LAT 블럭에 이해 알 수 있다. MUL 유닛의 경우에, 잠복은 2 이다. 따라서 PFWC는 알려진 잠복에 대응하는 잠복 로우에 대해 기록 인에이블 플랙을 설정한다. MUL 기능 유닛에 대해, 자유 SLOT는 잠복 로우(2)에 위치되며, 여기서 unitid 기록 어드레스 및 기록 인에이블 정보가 기억된다. 아래의 2개 클럭싸이클 동안에, 이 정보는 컬럼에 따라 잠복 로우 0 으로 통과한다.

각 SLOT 로우는 특별한 잠복에 대해 사용된다. 상기 상부 로우는 잠복(3)에 대해 사용되며 하부 로우는 잠복 0에 대해 사용된다. 각 로우는 매 클럭 싸이클에서 밑에 있는 로우로 데이타를 통과시킨다. 상기 통과된 데이타는 unitid, 즉, 기록 어드레스 및 기록 인에이블 표시이다.

잠복 로우 0의 출력은 RFWC 블럭의 출력이다.

각 로우는 LAT 서브-출력(1002)에 의해 발생된, 3개의 기록 인에이블 제어 신호를 수여한다. 로우에서 가장 왼쪽의 SLOT는 이들 제어 입력을 수여한다. 어떤 로직 동작후에, 이 가장 오른쪽 SLOT는 같은 로우내의 왼쪽 다음의 SLOT 블럭에 3개의 제어 출력을 통과시킨다. SLOT(33)의 제어 출력은 슬롯(23)으로 통과된다. SLOT(33)의 데이타 출력은 SLOT(32)로 통과되다.

각 LAT 유닛은 IIR에 이슈 슬롯으로부터 유닛 id(unitid) 및 기록 인에이블 신호(we)를 제공한다. 본 예의 IIR에서 3개의 슬롯이 존재하기 때무에, 3개의 LAT 유닛이 존재한다. 상기 LAT 유닛은 4개의 기록 인에이블 신호들 (We0, We1, We2, We3)을 발생시킨다. 상기 유닛 id가 CONST (잠복=0)이면 대응한 we 입력은 TRUE 이며, 따라서 We 0가 참(true)를 이룬다. 유닛 id가 ALU1, ALU2 또는 MEM(잠복=1)이고 대응의 we 입력이 TRUE 이면, we1 이 TRUE이 된다. 상기 유닛 i.d, 가 FPU(잠복=3)이고 이슈. we 가 TRUE 이면 we3가 TRUE가 된다.

제11도는 SLOT 서브 블럭의 하나에 대한 실행의 예를 도시한다. 각 SLOT는 IIR 로부터 4개의 데이타 입력(in1, in2, in3, in4)를 제공하며 하나의 데이타 출력(out)을 발생시킨다.

in1, in2, in3, in4 와 각각은 9 와이어를 가지며, 3개를 유닛 id에 통과시키고 6개는 기록 어드레스(wa)에 통과시킨다.

제1 입력(in1)은 IIR. 이슈 1. unitid 및 IIR. 이슈 1. Rdest 이다. 상기 제 2 입력(in2)는 IIR. 이슈 2.Rdest 및 IIR. 이슈 2. unitid 이다. 상기 제 3 입력(in3)는 IIR. 이슈 3. Rdest 및 IIR. 이슈 3. unitid 이다. 상기 제 4 입력(in4)는 이 입력이 모두 제로를 수신하게 접속된 상부 로우를 제외하고, 로우 슬롯의 출력에 접속된다.

각 SLOT은 4개의 제어 신호들(" we 표시기 in" i1.we, i2.we, 및 i3.we)를 제공한다. 상기 신호들 i1.we, i2.we, 및 i3.we는 로우에서 오른쪽 대부분 블럭에 대한 것을 제외하고, 다음 블럭에서 오른쪽까지 공급되며 여기서 이들 신호들은 대응 LAT 블럭에서 공급된다. 상기 신호 " we 표시 in" 은 상기가 와이어 " 0" 인 상부 블럭에 대한 것을 제외하고 다음 블럭에서 공급된다.

덧붙여, 하나의 로우 슬롯의 " we 표시기 in" 은 상기 SLOT의 " we 표시기 out" 에 접속된다. 상부 로우에서의 SLOT는 " 0" " we 표시기 in" 을 수신한다.

각 슬롯 SLOT는 4개의 제어 신호(01.we, 02we.

03.we, 및 " we 표시기 out" )를 발생시킨다. 상기 신호들 01.we, 02.we, 및 03.we는 그들이 접속되지 않는 가장 좌측 슬롯을 제외하고, 우측에 대해 다음 블럭에 공급된다. 상기 신호 " we 표시기 out" 은 이 신호가 기록 포트 인에이블 신호(W1.we, W2.we, W3.we, W4.we)로서 래지스터 파일에 공급되는 하부의 대부분 슬롯을 제외하고, 하부의 다음 블럭에 공급된다. 하부 로우의 out.unitid 신호들은 제8도의 멀티플렉서(801, 802, 803, 804)에 공급된 $W_1$MC, $W_2$MC, $W_3$MC 및 $W_4$MC 신호이다. 상기 하부 로우의 out.wa 신호들은 제6b도의 래지스터 파일(403)에 고급된 래지스터 파일 기록 어드레스 (W1.wa, W2.wa, W3.wa, W4.wa) 이다.

각 슬롯은 4개의 서브-블럭을 가진다. 즉, We1, We2, We3 및 we의 제어하에 in1, in2, in3 및 in4 중 하나를 선택하는 4:1 멀티플렉서, 입력 we, i1.we, i2.we, i3.we 로부터 we1, we2, we3, we, 01.we, 02.we 를 발생하는 우선 인코더, 입력으로서 we1, we2, we3 we 를 가지는 4개의 입력 논리 OR 게이트를 가진다. 상기 MUX는 래지스터 REG1에 출력하여 OR는 래지스터 REG2에 출력한다. 래지스터의 출력은 블럭의 출력이다. 최하위 로우의 슬롯은 기억이 필요하지 않기때문에 이들 래지스터가 부족하다.

상기 PREN 유닛은 모든 잠복용 기록 인에이블의 모든 가능한 조합에 대해 자유 슬롯이 요구된 잠복 로우에서 알 수 있는 것을 보증한다. PREN에 대한 진리표는 아래의 표II에 주어져 있다.

표II

PREN 로직 진리표

| i1.we | i2.we | i3.we | we지시자 | we1 | we2 | we3 | o1.we | o2.we | o3.we |
|---|---|---|---|---|---|---|---|---|---|
| x | x | x | 1 | 0 | 0 | 0 | i1.we | i2.we | i3.we |
| 1 | x | x | 0 | 1 | 0 | 0 | 0 | i2.we | i3.we |
| 0 | 1 | x | 0 | 0 | 1 | 0 | 0 | 0 | i3.we |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

제11도의 MUX는 상기 we 표시기 in 신호가 활성되는 경우 out에 in4 를 접속한다. we3 가 활성되면 in3 을 out에 접속하고 we2 가 활성이면 in2 을 접속하고 we1 이 활성이면 in1 을 접속한다. 상기 PREN 유닛은 이들중 단지 하나가 활성(" 1" )인 것을 보증한다.

제9-11도에 기술된 RFWC 기록 제어 유닛 하드웨어는 컴파일러 또는 프로그래머가 동시에 발생하는 4기록 이상을 방지하는 것을 요구한다. 예로 아래의 동작 순차가 방지한다.

시간 i : FPU ... ...

i+1 : MUL ... ...

i+2 : ALU $_1$ ALU$_2$ ....

i+3 : CO ... ...

이 예의 순차는 5기록 버스를 래지스터 파일에 기록하는 것을 요구하며 싸이클 i+3 를 완성하는 CO 동작의 결과를 기록하는데는 하나, " i+2" 로 시작된 ALU $_1$ 및 ALU$_2$ 에 대해서는 2, " i+3" 에 대해서는 MUL 동작에 대해서는 하나, FPU 동작에 대해서 하나를 요구한다.

동작의 어떤 다음의 순차방지는 시스템 수행에 약간 영향을 미치며 하드웨어 가격을 낮춘다. 특정예에서, 4개의 기록 포트를 가진 래지스터 파일은 5개의 기록 포트를 가진 더비싼 래지스터 파일에 충분히 대신한다.

## (57) 청구의 범위

### 청구항 1.

a)서로 병렬로 각각의 동작들을 실행하는 적어도 3개인 제1수의 기능 유닛들과,

b)적어도 2개이면서 제1수 보다는 적은 제2수의 명령들을 홀딩하고 명령을 기능 유닛에 대해 지정하기 위한 코드를 홀딩하는 명령 래지스터와,

c)기능 유닛들 중 적당한 유닛들이 명령 래지스터 내의 명령들을 병렬로 실행하게 하는 실행 제어 수단을 구비하는 처리기에 있어서,

상기 처리기는 명령 래지스터의 명령들에 관계하는 오퍼랜드들을 기억시키기 위한 멀티포트 래지스터 파일을 구비하며, 상기 멀티포트 레지스터 파일은 제3수의 판독 포트들을 포함하여, 상기 오퍼랜드들에 병렬 액세스를 제공하며, 상기 제3수의 판독 포트들은 상기 제2수의 기능 유닛들로 이루어지는 기능 유닛들 중 임의의 서브세트에 의해 래지스터 파일로부터 요구된 최대 수의 오퍼랜드들과 적어도 동일하며, 상기 제3수는 상기 기능 유닛들과 함께 요구되는 총수의 오퍼랜드들 보다 더 작고, 상기 처리기는,

e)판독 포트들에 병렬 액세스로 명령을 실행시키게 되는 기능 유닛들을 선택적으로 제공하는 판독 액세스 제공 수단을 구비하는 처리기.

### 청구항 2.

제1항에 있어서, 상기 판독 포트들은 판독 포트들의 그룹으로 이루어지며, 적어도 하나의 특정 그룹은 적어도 2개의 판독 포트들을 포함하며, 상기 판독 액세스 제공 수단은 그룹마다 기능 유닛들에 제공된 판독 포트들을 선택하도록 배열되며, 기능 유닛들 중 적어도 하나는 각각의 오퍼랜드를 동시로 수신하는 적어도 2개의 입력부들을 가지며, 상기 입력부들은 특정 그룹이 상기 기능 유닛들 중 하나에 대해 선택될 때 매번 상기 특정 그룹으로부터 각각의 소정 판독 포트들에 연결되어 있는 처리기.

청구항 3.

제1항 또는 제2항에 있어서, 상기 명령 래지스터는 상기 명령들을 이슈하는 제2수의 각각의 이슈 슬롯들로 이루어지며, 각각의 이슈 슬롯은 판독 포트들 중 적어도 하나의 각각의 그룹에 대응하며, 특정 기능 유닛은 상기 특정 이슈 슬롯으로부터의 명령을 실행할 때 매번 특정 이슈 슬롯에 대응하는 그룹으로부터 판독 포트 또는 포트들에 대한 액세스가 제공되는 처리기.

청구항 4.

제2항에 있어서, 각각의 특정 그룹은 제4수의 소정 포트들을 포함하고 상기 제4수는 기능 유닛들 중 임의의 하나에 의해 요구된 최대 수의 오퍼랜드들 동이한 처리기.

청구항 5.

제3항에 있어서, 각각의 특정 이슈 슬롯은 상기 특정 이슈 슬롯에 대응하는 그룹의 판독 포트들의 각각의 어드레스 입력부들에 연결되어 상기 특정 이슈 슬롯으로부터의 명령을 실행하는 기능 유닛이 어드레스된 오퍼랜드들을 필요로하는지의 여부와 상관 없이 상기 판독 포트들에서 어드레스된 오퍼랜드들에 액세스를 제공하는 동일한 수의 어드레스 필드를 포함하는 처리기.

청구항 6.

제5항에 있어서, 상기 기능 유닛들 중 적어도 하나는 상기 명령들 중 하나늘 실행하게 될 때, 상기 명령들 중 하나를 이슈하는 이슈 슬롯의 어드레스 필드로부터 정보를 수신하도록 배열되어 있으며, 상기 기능 유닛들 중 상기 적어도 하나는 실행하는 동안 오퍼랜드로서 정보를 사용하는 처리기.

청구항 7.

제1항에 있어서, 상기 멀티포트 래지스터 파일은 적어도 하나의 기록 포트를 구비하며, 상기 처리기는 명령들을 병렬로 실행하게 되고 그로부터의 결과를 형성하는 기능 유닛들에 대해 기록 포트 또는 포트들로의 액세스를 선택적으로 제공하는 기록 액세스 제공 수단을 구비하는 처리기.

청구항 8.

제7항에 있어서, 상기 명령 래지스터는 명령들을 이슈하는 제2수의 각각의 이슈 슬롯들로 이루어지며, 상기 각각의 이슈 슬롯은 동일한 수의 각각의 수의 결과 어드레스 필드들을 포함하며, 상기 기록 액세스 제공 수단은 기록 포트나 포트들의 어드레스 입력 또는 입력부들에 어드레스 필드 내용을 루팅하기 위해 배열되어 있는 처리기.

청구항 9.

제8항에 있어서, 상기 기록 액세스 제공 수단은 상기 기능 유닛들중 하나의 출력을 상기 기능 유닛들이 명령을 실행하게 된 후 잠복 주기 동안 기록 포트에 상기 결과 어드레스 필드의 내용과 조합하여, 커플링하도록 배열되어 있는 처리기.

청구항 10.

제9항에 있어서, 상기 래지스터 파일은 적어도 2개의 기록 포트들을 가지며, 상기 기록 선택수단은 사전 할당 후에 클럭 싸이클 동안 상기 기능 유닛들 중 하나에 특정 기록 포트를 사전 할당하기 위해 배열되어 있으며, 상기 클럭 싸이클 동안 상기 기능 유닛들 중 하나의 실행 결과는 유효하며, 상기 특정 기록 포트에상기기능 유닛들 중 하나의 출력을 커플링하고 클럭 싸이클 동안 상기 특정 기록 포트에상기 결과 어드레스 필드의 내용을 제공하도록 배열된 사전 할당 수단을 포함하는 처리기.

청구항 11.

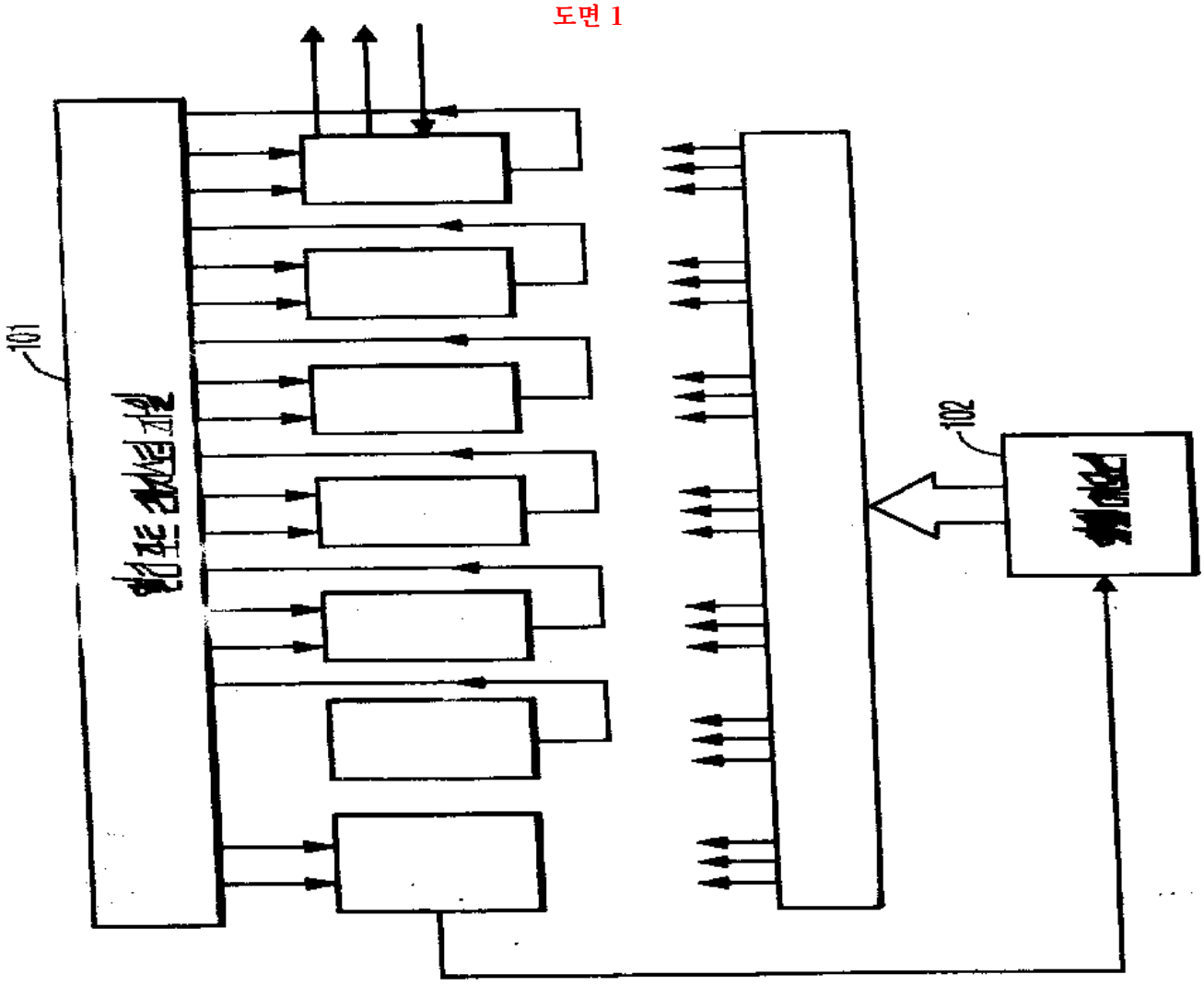
제10항에 있어서, 상기 사전 할당 수단은 특정 기록 포트가 클럭 사이클 동안 상기 기능 유닛들 중 하나에 미리 할당되지 않는다는 제한하에 기록 포트들로부터 특정 기록 포트를 선택하도록 배열되어 있는 처리기.

청구항 12.

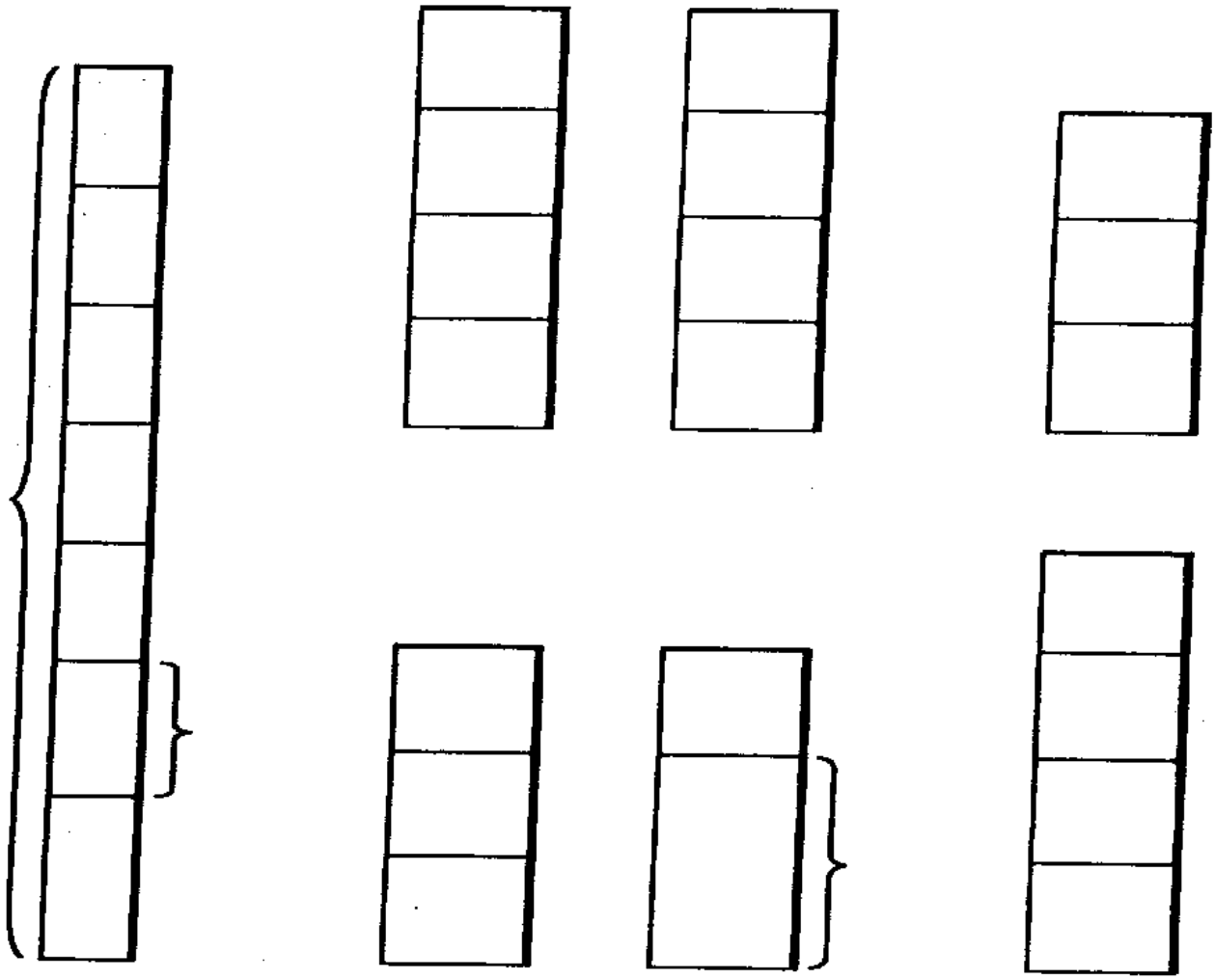
제11항에 있어서, 상기 사전 할당 수단은 래지스터들의 매트릭스를 포함하는 처리기.

도면

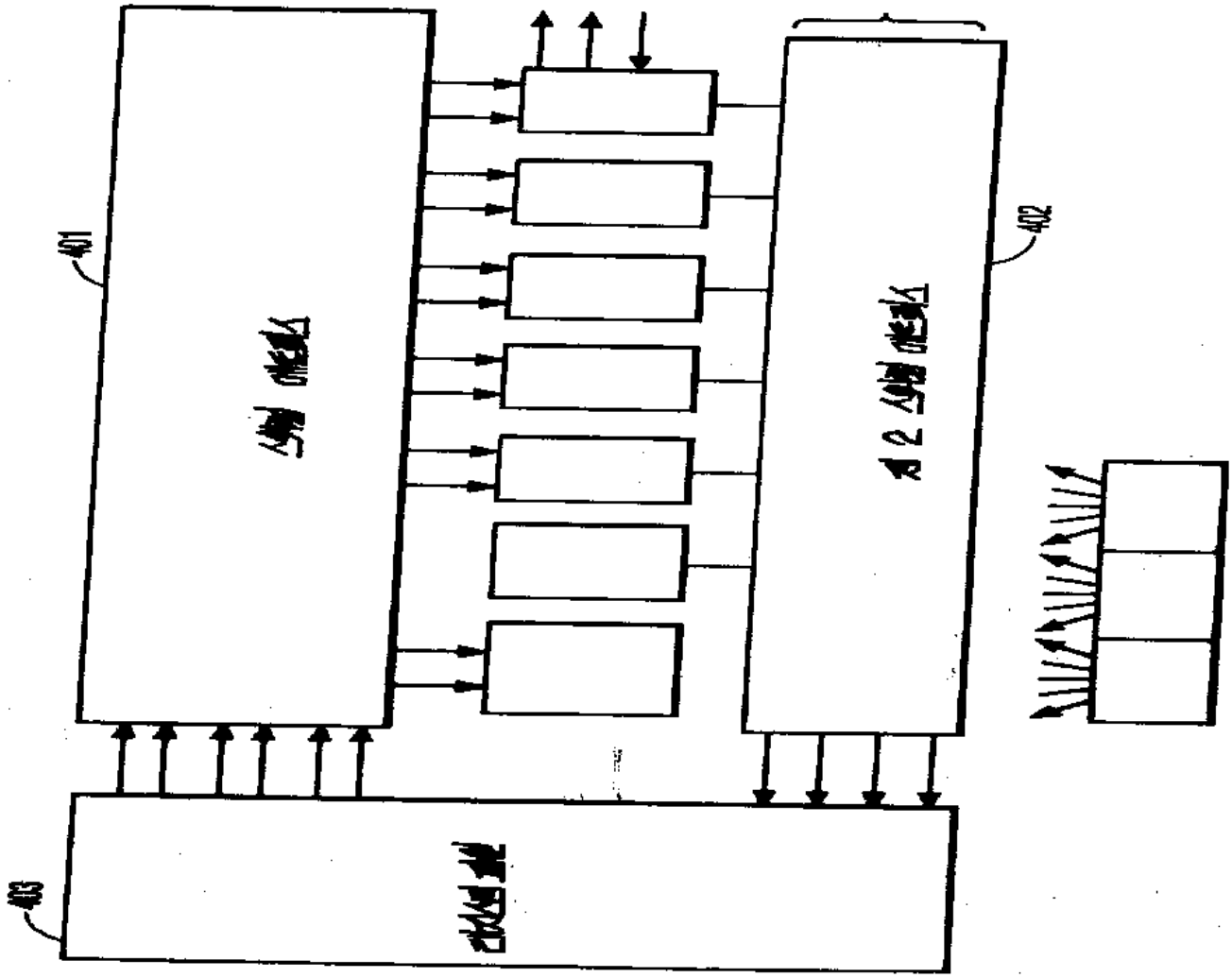
도면 1



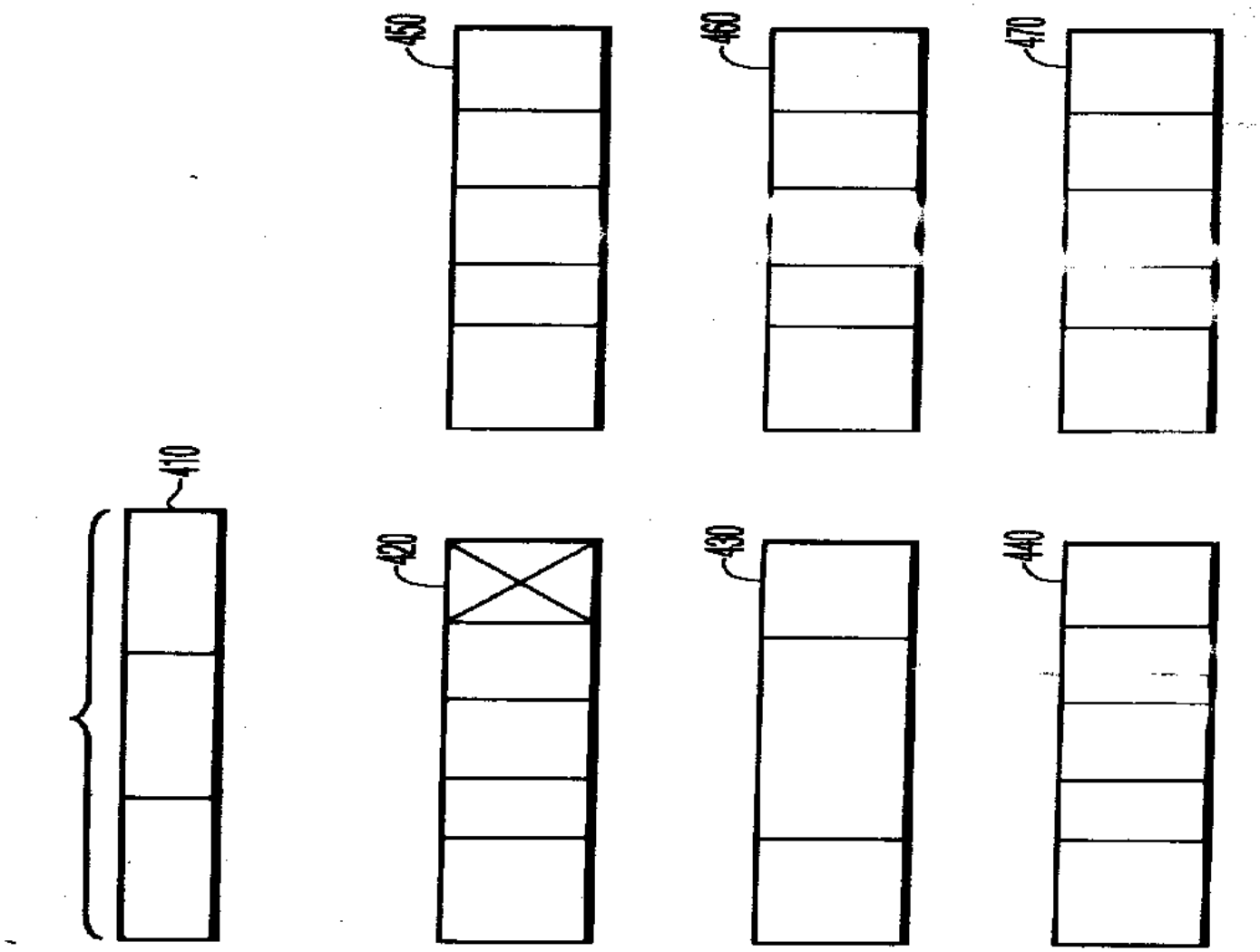
도면 2



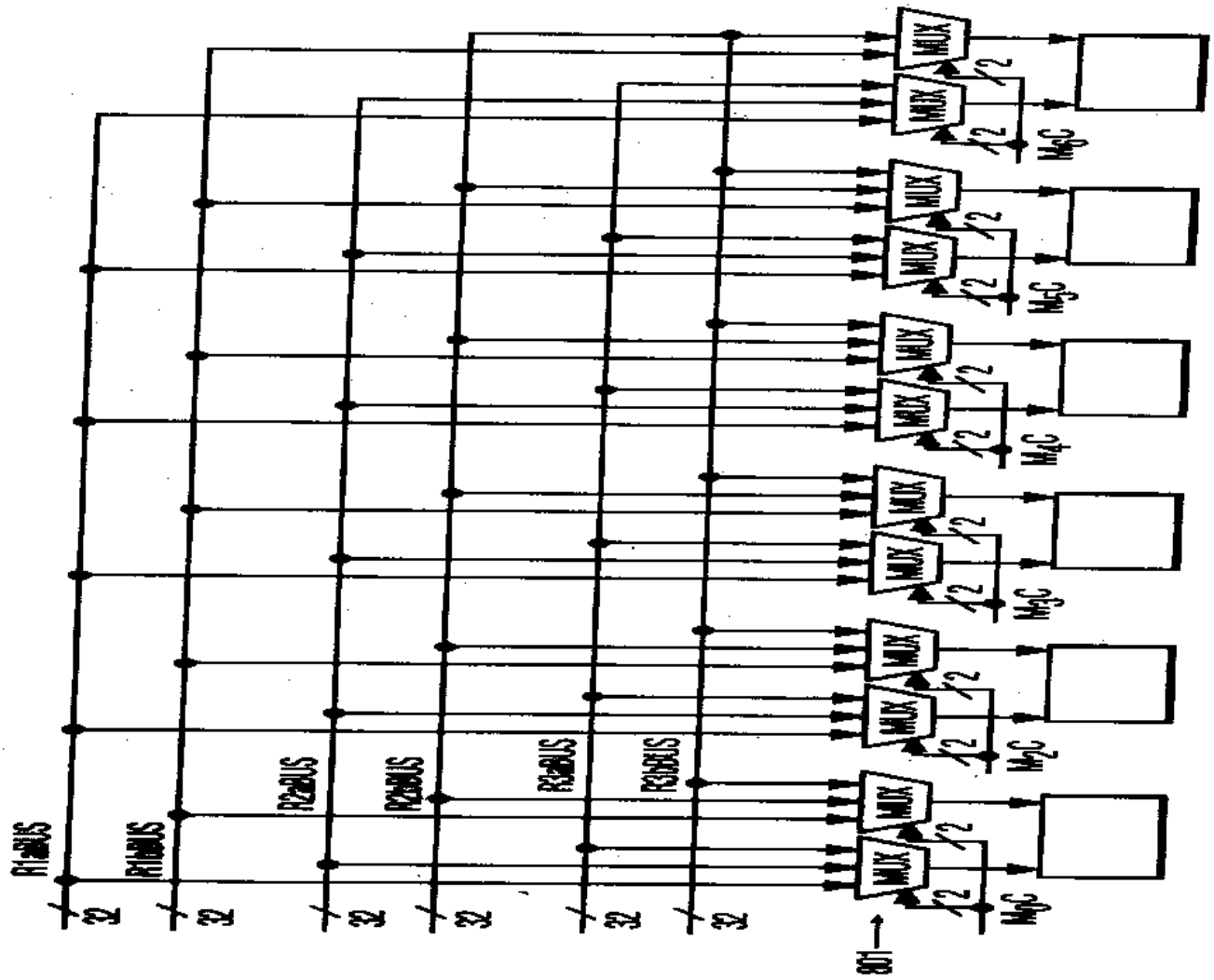
도면 3



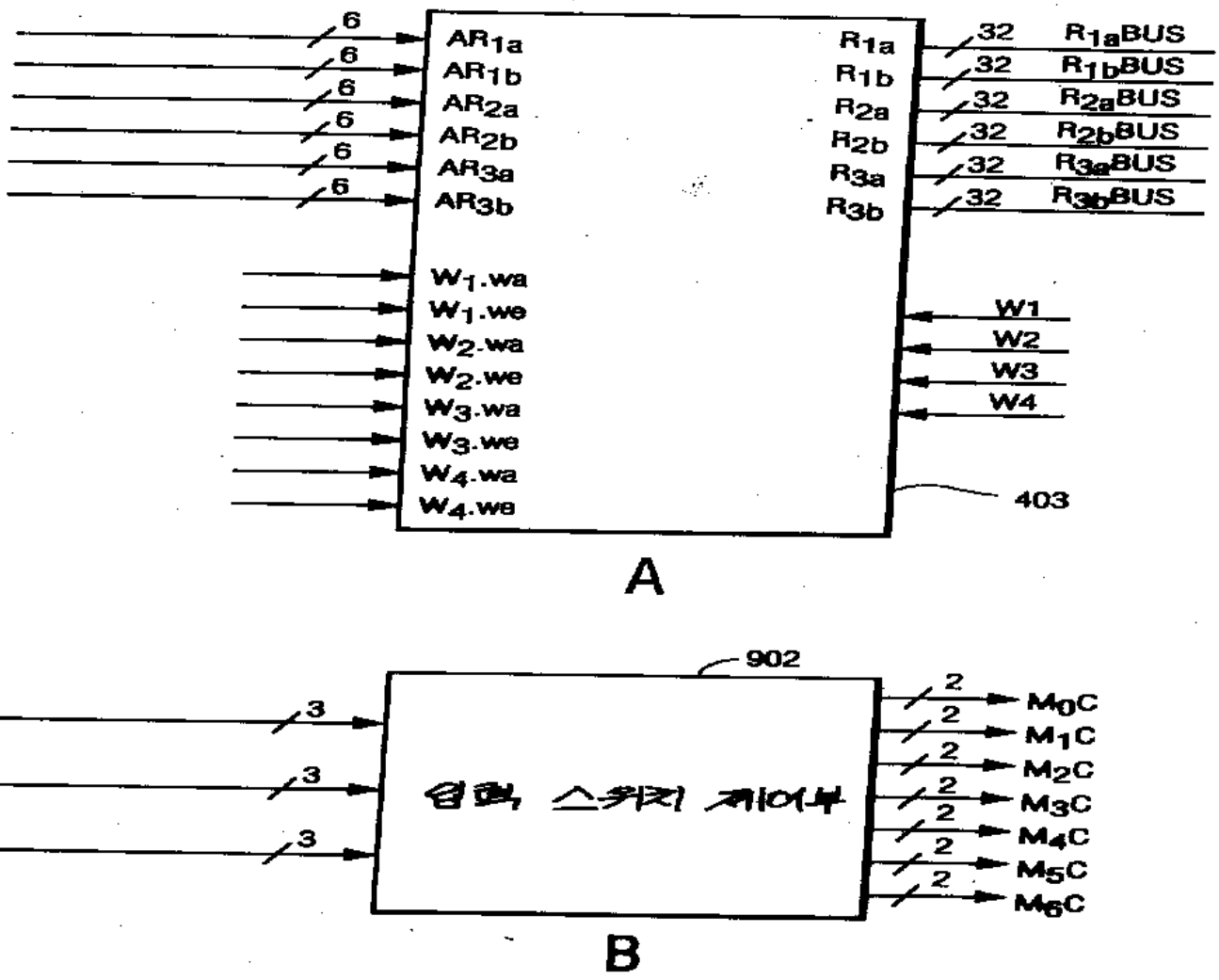
도면 4



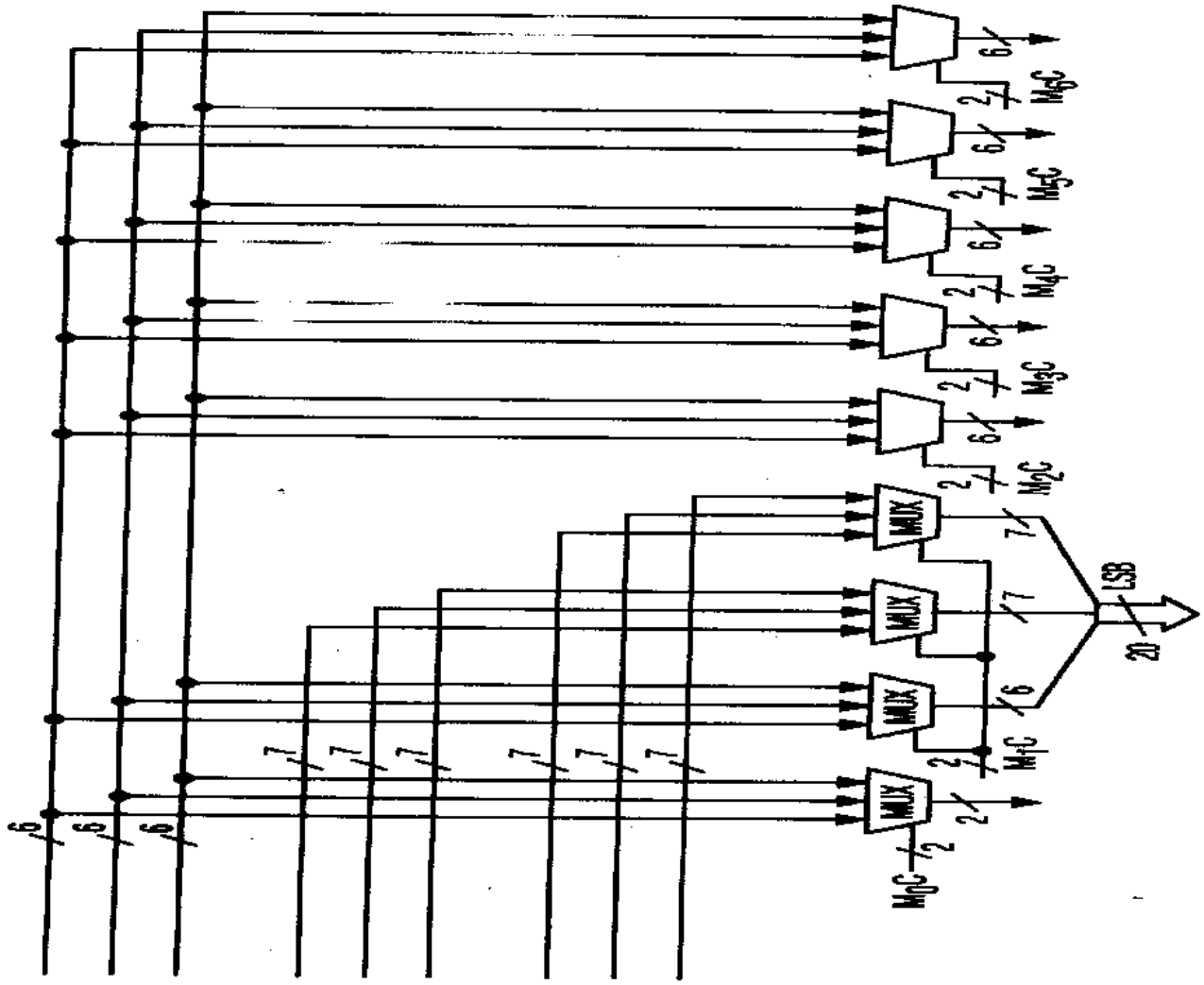
도면 5


R1aBUS
R1bBUS
R2aBUS
R2bBUS
R3aBUS
R3bBUS
32
MUX
1/2
MUX
1/2
MUX
1/2

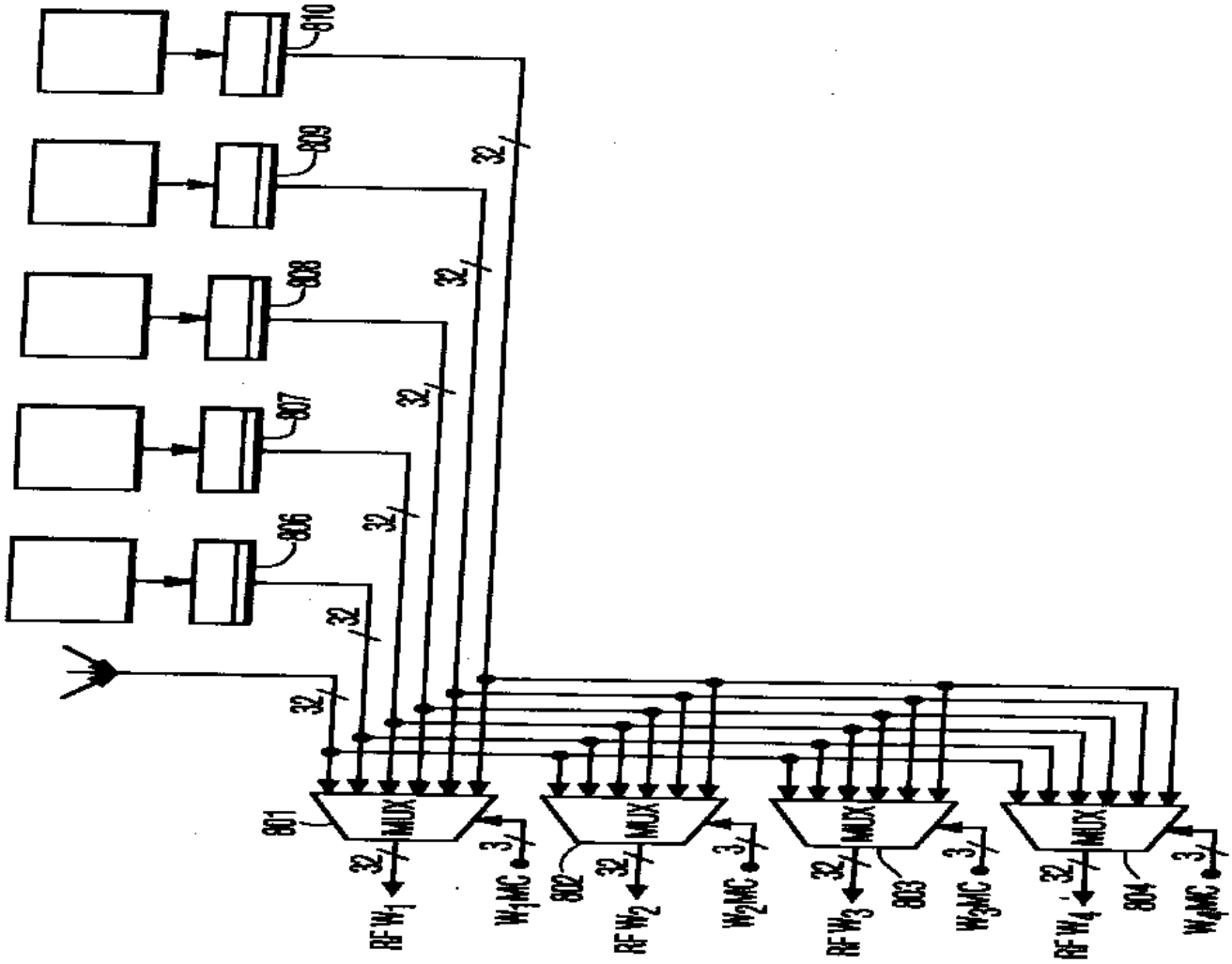
도면 6


6
AR1a
AR1b
AR2a
AR2b
AR3a
AR3b
W1.wa
W1.we
W2.wa
W2.we
W3.wa
W3.we
W4.wa
W4.we
R1a
R1b
R2a
R2b
R3a
R3b
32
R1aBUS
R1bBUS
R2aBUS
R2bBUS
R3aBUS
R3bBUS
W1
W2
W3
W4
403
A
902
3
2
M0C
M1C
M2C
M3C
M4C
M5C
M6C
B

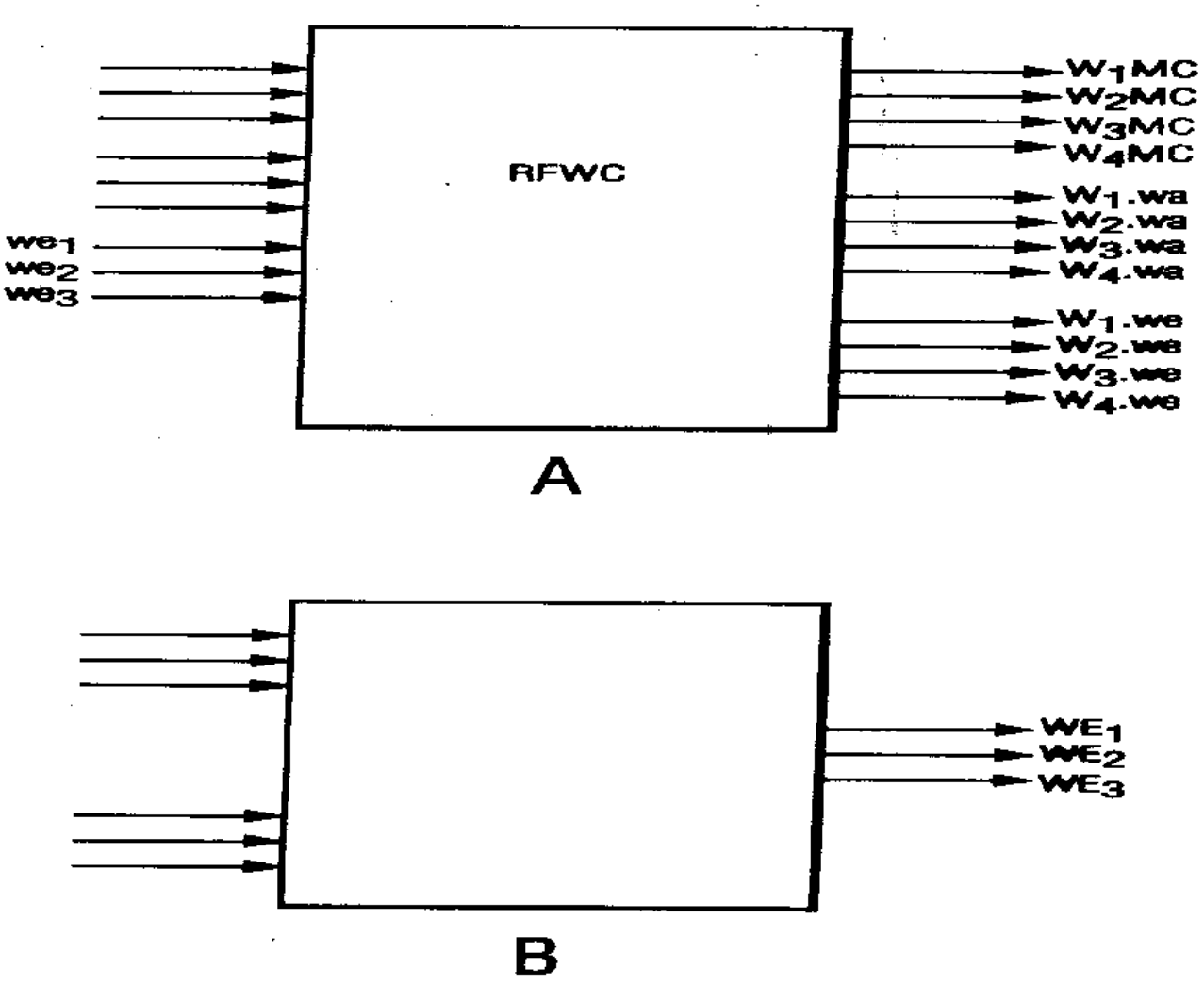
도면 7



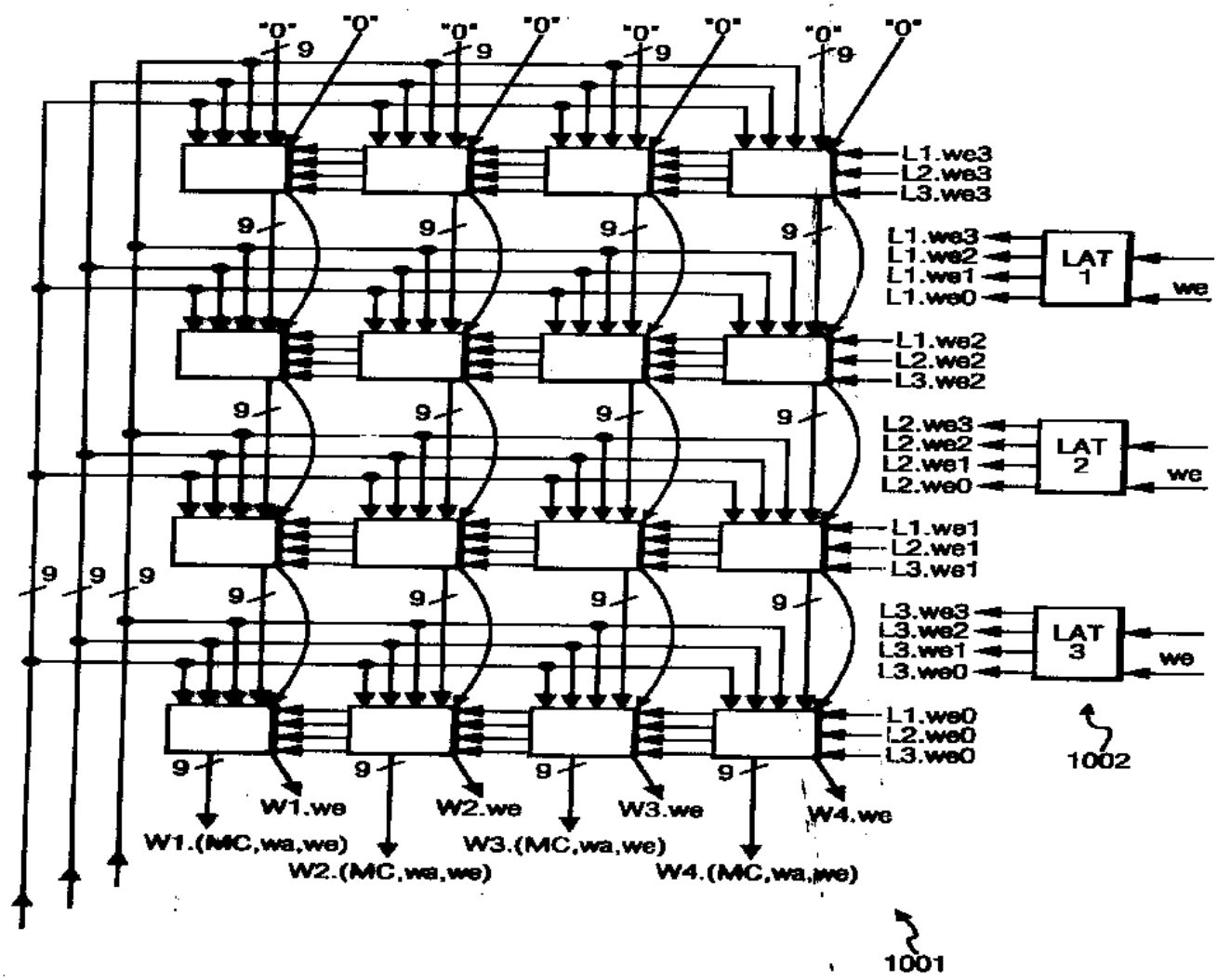
도면 8


810
809
808
807
806
32
801
802
803
804
MUX
3
RF W1
W1MC
RF W2
W2MC
RF W3
W3MC
RF W4
W4MC

도면 9


RFWC
we1
we2
we3
W1MC
W2MC
W3MC
W4MC
W1.wa
W2.wa
W3.wa
W4.wa
W1.we
W2.we
W3.we
W4.we
A
WE1
WE2
WE3
B

도면 10


"0"
9
L1.we3
L2.we3
L3.we3
L1.we2
L2.we2
L3.we2
L1.we1
L2.we1
L3.we1
L1.we0
L2.we0
L3.we0
LAT 1
LAT 2
LAT 3
we
W1.we
W2.we
W3.we
W4.we
W1.(MC,wa,we)
W2.(MC,wa,we)
W3.(MC,wa,we)
W4.(MC,wa,we)
1001
1002

도면 11

