

(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(51) Int. Cl.<sup>7</sup>  
G11C 7/00

(45) 공고일자 2000년08월01일

(11) 등록번호 10-0262434

(24) 등록일자 2000년05월02일

(21) 출원번호	10-1997-0043250	(65) 공개번호	특1998-0069917
(22) 출원일자	1997년08월29일	(43) 공개일자	1998년10월26일
(30) 우선권주장	97-037683 1997년02월21일 일본(JP)		
(73) 특허권자	미쓰비시덴키 가부시기가이샤 다니구찌 이찌로오, 기타오카 다카시		
(72) 발명자	일본국 도쿄도 지요다쿠 마루노우치 2초메 2반 3고 코야마 마사유키		
(74) 대리인	일본 도쿄도 지요다쿠 마루노우치 2-2-3 미쓰비시 덴키가부시끼 가이샤 내 코바야시 나오히로 일본 도쿄도 지요다쿠 마루노우치 2-2-3 미쓰비시 덴키가부시끼 가이샤 내 김창세		

**심사관 : 퇴-정해권**

**(54) 쌍방향 전송형 기억장치 및 메모리의 입출력 제어방법**

**요약**

본 발명에 의하면, 데이터를 기억하는 영역을 감소시키면서, 또한 복수조의 데이터에 대하여 연속적으로 FIL0의 기능을 얻는다.

메모리 M0으로부터 입력된 데이터의 열은 메모리 M3까지 쉬프트되고, 스위치백하여 메모리 M0으로부터 판독되기 때문에, 이 열에 대하여 FIL0의 기능을 얻을 수 있다. 한편, 메모리 M3으로부터 입력된 데이터의 열은 메모리 M0까지 쉬프트되고, 스위치백하여 메모리 M3으로부터 판독되기 때문에, 이 열에 대하여도 FIL0의 기능을 얻을 수 있다. 그리고, 이들 2개의 데이터의 열은 서로 푸쉬-풀의 관계를 유지하면서 메모리 M0~M3의 사이를 데이터가 쉬프트하기 때문에, 필요한 메모리의 수는 데이터의 각 열의 요소만 있으면 충분한다.

**대표도**

**도3**

**명세서**

**도면의 간단한 설명**

- 도 1은 도 2와 더불어 실시예 1에 관련된 쌍방향 FIL0의 동작을 도시하는 플로우 차트.
- 도 2는 도 1과 더불어 실시예 1에 관련된 쌍방향 FIL0의 동작을 도시하는 플로우 차트.
- 도 3은 실시예 1에 관한 쌍방향 FIL0의 구성을 예시하는 블록도.
- 도 4는 도 3에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 5는 도 3에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 6은 도 3에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 7은 도 3에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 8은 도 3에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 9는 도 3에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 10은 도 3에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 11은 도 3에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 12는 도 3에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 13은 도 3에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 14는 도 3에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 15는 도 3에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 16은 도 3에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.

- 도 17은 표 1 내지 표 3의 리스트에 기초하여 생성되는 쌍방향 FIL0의 구성을 도시하는 블록도.
- 도 18은 도 19 내지 도 27과 더불어 논리회로(110)의 구성을 상세하게 도시하는 회로도.
- 도 19는 도 18 및 도 20 내지 도 27과 더불어 논리회로(110)의 구성을 상세하게 도시하는 회로도.
- 도 20은 도 18 내지 도 19 및 도 21 내지 도 27과 더불어 논리회로(110)의 구성을 상세하게 도시하는 회로도.
- 도 21은 도 18 내지 도 20 및 도 22 내지 도 27과 더불어 논리회로(110)의 구성을 상세하게 도시하는 회로도.
- 도 22는 도 18 내지 도 21 및 도 23 내지 도 27과 더불어 논리회로(110)의 구성을 상세하게 도시하는 회로도.
- 도 23은 도 18 내지 도 22 및 도 24 내지 도 27과 더불어 논리회로(110)의 구성을 상세하게 도시하는 회로도.
- 도 24는 도 18 내지 도 23 및 도 25 내지 도 27과 더불어 논리회로(110)의 구성의 상세를 나타내는 회로도.
- 도 25는 도 18 내지 도 24 및 도 26 내지 도 27과 더불어 논리회로(110)의 상세를 도시하는 회로도.
- 도 26은 도 18 내지 도 25 및 도 27과 더불어 논리회로(110)의 구성을 상세하게 도시하는 회로도.
- 도 27은 도 18 내지 도 26과 더불어 논리회로(110)의 구성을 상세하게 도시하는 회로도.
- 도 28은 도 29와 더불어 실시예 2에 관한 쌍방향 FIL0의 동작을 도시하는 플로우 차트.
- 도 29는 도 28과 더불어 실시예 2에 관한 쌍방향 FIL0의 동작을 도시하는 플로우 차트.
- 도 30은 실시예2에 관한 쌍방향 FIL0의 구성을 예시하는 블록도.
- 도 31은 도 30에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 32는 도 30에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 33은 도 30에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 34는 도 30에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 35는 도 30에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 36은 도 30에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 37은 도 30에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 38은 도 30에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 39는 도 30에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 40은 도 30에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 41은 도 30에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 42는 도 30에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 43은 도 30에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도.
- 도 44는 도 45 내지 도 52와 더불어 표 4 내지 표 8에서 도시된 HDL의 내용에 기초하여 생성된 쌍방향 FIL0의 구성을 상세하게 도시하는 회로도.
- 도 45는 도 44 내지 도 46 내지 도 52와 더불어 표 4 내지 표 8에서 도시된 HDL의 내용에 기초하여 생성된 쌍방향 FIL0의 구성을 상세하게 도시하는 회로도.
- 도 46은 도 44 내지 도 45 내지 도 52와 더불어 표 4 내지 표 8에서 도시된 HDL의 내용에 기초하여 생성된 쌍방향 FIL0의 구성을 상세하게 도시하는 회로도.
- 도 47은 도 44 내지 도 46 내지 도 52와 더불어 표 4 내지 표 8에서 도시된 HDL의 내용에 기초하여 생성된 쌍방향 FIL0의 구성을 상세하게 도시하는 회로도.
- 도 48은 도 44 내지 도 47 및 도 49 내지 도 52와 더불어 표 4 내지 표 8에서 도시된 HDL의 내용에 기초하여 생성된 쌍방향 FIL0의 구성을 상세하게 도시하는 회로도.
- 도 49는 도 44 내지 도 48 및 도 50 내지 도 52와 더불어 표4 내지 표8에서 도시된 HDL의 내용에 기초하여 생성된 쌍방향 FIL0의 구성을 상세하게 도시하는 회로도.
- 도 50은 도 44 내지 도 49 및 도 51 내지 도 52와 더불어 표 4 내지 표 8에서 도시된 HDL의 내용에 기초하여 생성된 쌍방향 FIL0의 구성을 상세하게 도시하는 회로도.
- 도 51은 도 44 내지 도 50 및 도 52와 더불어 표 4 내지 표 8에서 도시된 HDL의 내용에 기초하여 생성된 쌍방향 FIL0의 구성을 상세하게 도시하는 회로도.
- 도 52는 도 44 내지 도 51과 더불어 표 4 내지 표 8에서 도시된 HDL의 내용에 기초하여 생성된 쌍방향 FIL0의 구성을 상세하게 도시하는 회로도.

도 53은 종래기술을 도시한 블록도.

#### 도면의 주요부분에 대한 부호의 설명

M0~M3, R256 : 메모리

M00, M01~M30, M31 : D플립플롭

MODE : 모드 제어 신호

R\_MODE\_CNT, W\_MODE\_CNT, MODE\_CNT : 모드 제어값

#0~#N : 기억 단위.

#### **발명의 상세한 설명**

##### **발명의 목적**

##### **발명이 속하는 기술분야 및 그 분야의 종래기술**

본 발명은 기억장치에 관한 것으로, 특히 연속적으로 FIL0(First In Last Out)의 기능을 다할 수 있는 쌍방향 FIL0에 관한 것이다.

FIL0의 기능을 하는 기억장치(이하, 이러한 기억장치도 「FIL0」라고 칭한다)는 순차적으로 입력된 데이터를 입력된 순서와는 반대로 데이터를 출력한다.

종래의 FIL0은 푸쉬조작과 팝조작에 의해서 동작하기 때문에, 이것을 하나만 이용했을 경우에는 제 1 조의 데이터에 대하여 FIL0의 처리를 얻은 뒤, 즉시 제 2 조의 데이터에 대하여 FIL0의 처리를 얻을 수는 없다. 제 1 조의 데이터의 판독이 끝날 때까지 다른 조의 데이터를 FIL0에 입력하면 제 1 조의 데이터가 파괴되기 때문이다.

도 53은 종래의 FIL0을 두 개 이용하여 제 1 및 제 2 조의 데이터에 대하여 FIL0의 처리를 연속적으로 얻는 구성을 도시한 블록도이다. FIL01, FIL02는 디멀티플렉서 DMUX0에 의해서 입력 데이터 DIN이 조마다 나뉘어 인가되고, 멀티플렉서 MUX0에 의해서 어느 하나의 출력이 출력 데이터 DOUT로서 출력된다.

예를 들면, 입력 데이터 DIN이 제 1 조의 데이터를 디멀티플렉서 DMUX0에 인가할 때는 FIL01에 제 1 조의 데이터가 순차적으로 입력된다. 그리고, 입력 데이터 DIN이 제 2 조의 데이터를 디멀티플렉서 DMUX0에 인가할 때는 FIL02에 제 2 조의 데이터가 순차적으로 입력되면서, 멀티플렉서 MUX0를 통하여 FIL01에 인가되어 있던 데이터가 역순으로 출력 데이터 DOUT로서 출력된다. 그 후, 멀티플렉서 MUX0를 통하여 FIL02에 인가되어 있던 데이터가 역순으로 출력 데이터 DOUT로서 출력될 때에는 입력 데이터 DIN이 제 3 조의 데이터를 디멀티플렉서 DMUX0에 인가한다.

종래의 기술에서는 교대로 선택되는 한쌍의 FIL0을 이용하지 않으면 복수조의 데이터에 대하여 연속적으로 FIL0의 처리를 실행할 수 없기 때문에, 데이터를 기억하는 영역은 1조의 데이터의 2배분이 필요하다는 문제점이 있었다.

##### **발명이 이루고자 하는 기술적 과제**

본 발명은 상기 문제점을 해결하기 위해서 이루어진 것으로, 복수의 기억소자의 사이에 있어서 데이터 전송방향을 위에서 아래로 향하는 방향과, 아래에서 위로 향하는 방향의 양방향에 대해서 실행하고, 데이터를 기억하는 영역을 감소시키면서도, 연속적으로 FIL0의 기능을 가능하게 하는 쌍방향 FIL0을 제공하는 것을 목적으로 하고 있다.

본 발명 중 청구항 1에 관련된 것은 모두가 소정길이의 데이터를 저장하는 제0 내지 제 (N-1)(N≥2)의 기억 단위를 구비하고, 입력 데이터는 N개마다 조를 이뤄 교대로 상기 제 0 및 제 (N+K-1)의 기억 단위로 입력되며, 상기 입력 데이터가 다른 2개의 상기 조의 선두의 데이터가 K개의 상기 기억 단위를 끼워서 대향하면서, 상기 제 0 내지 제 (N+K-1)의 기억 단위에서 저장되는 데이터에 대하여 순차적으로 (N+K-1)회의 쉬프트 전송이 실행되고, 상기 쉬프트 전송의 방향이 교대로 전향됨으로써 상기 입력 데이터가 스위치백하여 출력된다.

본 발명 중 청구항 2에 관련된 것은 모두가 소정길이의 데이터를 저장하고, 제 0 내지 제 (N+K-1)의 어드레스에 대응하는 제 0 내지 제 (N+K-1)(N≥2, K≥0)의 기억 단위를 가지며, 클럭에 기초하여 동작하는 메모리와, 상기 메모리의 판독 어드레스 및 기입 어드레스를 각각 생성하는 어드레스 생성부를 구비하는 쌍방향 전송형 기억장치로서, 입력 데이터는 N개마다 조를 이루고, 상기 기억 단위가 저장하는 내용을 판독하면서 이것에 따라서 입력 데이터가 기입되고, 상기 입력 데이터의 조의 갱신에 대응하여 상기 판독 어드레스 및 상기 기입 어드레스가 이동하는 방향이 역전된다.

본 발명중, 청구항 3에 관련된 것은 모두가 소정길이의 데이터를 저장하는 제 1 내지 제 N+K(N≥2, K≥0)의 기억 단위를 가지며, 클럭에 기초하여 동작하는 메모리의 입출력 제어방법으로서, 제어신호가 제 1 값을 취할 경우에는 상기 메모리의 판독 어드레스가 기입 어드레스에 대하여 상기 K만큼 크게 설정되면서, 상기 클럭에 기초하여 상기 판독 어드레스 및 상기 기입 어드레스 모두가 1 증가하고, 상기 제어신호가 제 2 값을 취할 경우에는 상기 판독 어드레스가 상기 기입 어드레스에 대하여 상기 K만큼 작게 설정되면서, 상기 클럭에 기초하여 상기 판독 어드레스 및 기입 어드레스 모두가 1 감소하며, 상기 제어신호는 상기 기입 어드레스가 상기 제 N의 기억 단위를 지정한 경우에 대응하여 상기 제 2 값으로, 상기 제 K+1의 기억 단위를 지정한 경우에 대응하여 상기 제 1 값으로 각각 설정되고, 상기 제 K의 기억 단위에 대하여 상기 기입이 행해진 뒤, 상기 기입 어드레스 및 상기 판독 어드레스는 각각 상기 제 0 어드

레스, 및 상기 제 K 어드레스에 각각 설정된다.

### 발명의 구성 및 작용

#### (실시예 1)

실시예 1에서는 쉬프트 레지스터형의 쌍방향 FIL0에 대하여 설명한다. 도 1 및 도 2는 양도면이 합쳐져 실시예 1에 관련된 쌍방향 FIL0의 동작을 도시한 플로우 차트이다. 양도면은 접속자 J1~J4에 의해서 서로 결합된다. 또한, 도 3은 실시예 1에 이러한 쌍방향 FIL0의 구성을 예시하는 블록도이다. 그리고, 도 4 내지 도 16은 도 3에 도시된 쌍방향 FIL0의 동작을 순서대로 도시하는 모식도이다.

도 3에 도시된 쌍방향 FIL0은 쉬프트 레지스터를 구성하는 메모리 M0~M3 및, 이들 사이에서 순차적으로 데이터를 전송하는 제어부 CNT1~CNT3을 구비하고 있다. 구체적으로는, 메모리 M0, M1의 사이의 데이터의 전송은 제어부 CNT1이, 메모리 M1, M2의 사이의 데이터의 전송은 제어부 CNT2가, 메모리 M2, M3의 사이의 데이터의 전송은 제어부 CNT3이 각각 실행한다. 제어부 CNT1~CNT3은 클럭 CLK에 동기하여 동작한다. 데이터가 전송되는 방향은 모드신호 MODE의 값이 "0", "1"을 취하는 것에 대응하여, 각각을 위에서 아래로 향하는 방향(M0으로부터 M3으로 향하는 방향), 아래에서 위로 향하는 방향(M3으로부터 M0으로 향하는 방향)이 된다.

디멀티플렉서 DMUX 및 멀티플렉서 MUX는 각각 모드신호 MODE의 값에 기초하여 출력 및 입력을 실행한다. 구체적으로는, 디멀티플렉서 DMUX는 입력 데이터 DIN을 입력하고, 모드신호 MODE의 값이 "0", "1"을 취하는 것에 대응하여 각각 "0", "1"로 부기된 출력단(이하 「"0" 출력단」, 「"1" 출력단」으로 한다)에 입력 데이터 DIN을 출력한다. 또한, 멀티플렉서 MUX는 모드신호 MODE의 값이 "0", "1"을 취하는 것에 대응하여 각각 "0", "1"로 부기된 입력단(이하 「"0" 입력단」, 「"1" 입력단」으로 한다)에 입력되는 데이터 DOUT1, DOUT2를 출력 데이터 DOUT로서 출력한다. 여기서, 데이터 DOUT1, DOUT2는 각각 메모리 M3, M0에 저장된 데이터이다.

모드신호 MODE는 카운터 COUNT1, COUNT2, 및 비교기 CMP에 의해서 생성된다. 카운터 COUNT1은 클럭 CLK를 카운트하여 모드 제어값 MODE\_CNT를 생성하고, 비교기 CMP는 이것을 소정의 값, 예를 들면, 2비트의 값 "11" (10진수에서 말하는 "3")과 비교한다. 모드 제어값 MODE\_CNT는 입력 데이터 DIN을 저장하는 메모리가 데이터의 이동방향을 따라서 몇번째에 있는지를 지시한다. 모드 제어값 MODE\_CNT가 소정의 값과 일치했을 경우에는 (이것은 모드신호 MODE가 "0"을 취한다면 메모리 M3이, "1"을 취한다면 메모리 M0이, 각각 입력 데이터 DIN을 저장하는 것을 의미한다), 카운터 COUNT1을 리셋하여 모드 제어값 MODE\_CNT의 값을 "0"으로 하고, 또한 카운터 COUNT2에 카운트시킨다. 카운터 COUNT2는 클럭 CLK를 카운트하는 것은 아니고, 비교기 CMP의 출력을 카운트하는 1비트의 카운터이며, 그 출력인 모드신호 MODE의 값이 변화하는 타이밍이 클럭 CLK에 동기한다.

카운터 COUNT1은 비교기 CMP의 출력에 의해서 리셋되기 위한 리셋단R2 외에, 쌍방향 FIL0의 리셋에 이용되는 리셋신호RST에 의해서도 리셋되기 때문에 리셋단R도 가지고 있다. 마찬가지로, 카운터 COUNT2에도 리셋신호RST에 의해서 리셋되기 위한 리셋단R이 마련되어 있다. 또한, 각 메모리 M0~M3에도 리셋신호RST에 의해서 리셋되기 위한 리셋단R이 마련되어 있다.

도 1에 도시된 플로우 차트에서, 단계 S101은 클럭 CLK에 관계없이 실행된다. 이 단계 S101에서, 리셋 명령이 인가되었는지 아닌지가 판단된다. 도 3에 의거해서 말하면 리셋신호RST가 하강하였는지 아닌지가 판단된다. 리셋 명령이 인가되었을 경우(리셋신호RST가 하강했을 경우), 접속자 J2를 거쳐서 도 2에 도시된 단계 S113~S115가 실행되고, 접속자 J3을 거쳐서 다시 단계 S101로 되돌아간다.

단계 S113~S115에서 쌍방향 FIL0의 리셋을 위한 처리가 실행된다. 단계 S113에서는 제 0 번째 내지 제 (N-1) 번째의 메모리, 즉 모든 메모리의 내용이 초기화된다. 도 3에 의거해서 말하면 N=4이고, 메모리 M0~M3의 내용은 이들 리셋단R에 "H"로부터 "L"로 천이가 인가됨으로써 클리어된다. 단계 S114에서는 동작모드가 제 1 모드로 설정된다. 이것은 도 3에 의거해서 말하면 모드신호 MODE의 값이 "0"으로 설정되는 것을 나타내며, 카운터 COUNT2의 리셋단R에 "H"로부터 "L"로 천이가 인가됨으로써 실현된다. 단계 S115에서는 마찬가지로, 카운터 COUNT1의 리셋단R에 "H"로부터 "L"로 천이가 인가됨으로써 모드 제어값 MODE\_CNT가 "0"으로 설정된다.

도 1에 도시된 단계 S102, S100에 관해서는 도 3에 도시된 쌍방향 FIL0에서는 특히 대응하는 기구가 도시되어 있지 않지만, 대기(wait)명령WT, 인에이블(enable)명령EN을 이용하여 메모리를 관리하는 것은 통상 자주 실행되고 있고, 그 기구도 주지의 기술을 이용하여 실현할 수 있다.

단계 S103에서, 동작모드가 제 1 모드인지 제 2 모드인지가 판단된다. 여기서, 제 1 및 제 2 모드는 각각 모드신호 MODE의 값이 "0", "1"인 것에 대응하고, 제어부 CNT1~CNT3이 데이터를 전송하는 방향은 이것에 대응하여 위에서 아래로 향하는 방향, 아래에서 위로 향하는 방향이 된다. 단계 S113~S115가 실행되어 처음으로 단계 S103이 실행되는 경우에는, 단계 S114가 실행된 결과, 모드신호 MODE의 값이 "0"으로 되고, 단계 S104로 진행한다.

단계 S104에서, 종단의 메모리인 제 (N-1) 번째의 메모리의 내용이 출력 데이터로서 채용된다. 이것은 멀티플렉서 MUX가 모드신호 MODE의 값이 "0"인 것에 대응하여, 출력 데이터 DOUT로서 데이터 DOUT1을 채용함으로써 실현된다. 도 4는 이 상태를 도시하는 도면이고, 모든 메모리 M0~M3이 클리어되어 "0"을 저장하고 있는 상태에서, 출력 데이터 DOUT는 메모리 M3이 저장하는 내용(데이터 DOUT1)을 취하고 있다.

단계 S106, S108, S111, S112는 클럭 CLK에 동기하여, 예를 들면 그 상승의 타이밍을 계기로 하여 실행된다. 우선 단계 S106에 있어서, 선두의 메모리인 제 0 번째의 메모리 내지 끝에서 2번째의 메모리인 제 (N-2) 번째의 메모리의 내용을 선두로부터 2번째의 메모리인 제 1 번째의 메모리 내지 종단의 메모리인 제 (N-1) 번째의 메모리로 전송한다. 이것은 제어부 CNT1~CNT3이 메모리 M0~M3의 사이에서 데이터를 위에서 아래로 향하는 방향으로 전송함으로써 실현된다. 따라서, 도 4에 있어서 메모리 M0~M2에 저장되어 있던 데이터 "0", "0", "0"이 각각 메모리 M1~M3에 저장되게 된다. 또한, 단계 S108에서 선두의 메모리

인 제 0 번째의 메모리에 입력 데이터의 값이 저장된다. 예를 들면 입력 데이터 DIN의 값으로서 A1이 인가되어 있으면, 메모리 M0에는 데이터A1이 저장된다. 도 5는 이렇게 하여 얻어진 각 메모리 M0~M3의 저장내용을 나타내고 있다.

단계 S110에서는 모드 제어값 MODE\_CNT의 값이 판독된다. 도 5에 도시된 상태에서는 입력 데이터 DIN의 값이 저장되는 대상은 메모리 M0이고, 모드 제어값 MODE\_CNT가 "0"이기 때문에 단계 S112로 진행된다. 단계 S112에 있어서, 모드 제어값 MODE\_CNT는 "1" 증가한다.

이것은 입력 데이터 DIN을 저장하는 메모리가 순차적으로 이동하는 것을 도시한다. 그리고, 해당 메모리가 선두 또는 종단의 메모리에 대응하게 되었을 경우(즉 (N-1)과 같아졌을 경우)에는, 단계 S110으로부터 단계 S111로 진행되어 모드신호 MODE의 값을 전환하여 동작모드를 변경하고, 모드 제어값의 값도 "0"으로 설정한다. 해당 메모리가 선두 또는 종단의 메모리에 대응하지 않는 동안은 접속자J1을 거쳐서 단계 S103으로 되돌아간다.

도 6은 단계 S114가 실행되고 나서 2회째에 단계 S104가 실행되고 있는 상태를 도시하고 있다. 그리고, 클럭 CLK의 상승을 계기로 하여, 도 4 내지 도 6에서 메모리 M0~M2에 저장되어 있던 데이터A1, "0", "0"이 각각 메모리 M1~M3에 저장되고(단계 S106), 입력 데이터 DIN의 새로운 값 A2가 메모리 M0으로 저장된다(단계 S108, 도 7). 그 후의 단계 S112에 있어서 모드 제어값 MODE\_CNT의 값은 증가하지만, 그 값은 10진수에서 2로서 N-1(=3)과 같지는 않고, 단계 S111로부터 접속자J1, 단계 S103을 거쳐서 단계 S104로 되돌아간다. 도 8은 이 상태를 도시하고 있다.

이러한 처리가 반복되어 순차적으로 갱신되는 입력 데이터 DIN의 값 A3, A4가 메모리 M0으로 저장된다. 그리고, 데이터A4가 메모리 M0으로 저장될 때의 클럭 CLK의 상승에 대응하여 실행되는 단계 S110에 있어서는, 이미 모드 제어값 MODE\_CNT가 "3"으로 되어 있었기 때문에 단계 S111의 처리를 받는다. 즉 모드 제어값 MODE\_CNT가 "0"이 되고, 모드신호 MODE가 "1"로 된다. 이것은 도 3에 있어서, 비교기 CMP가 모드 제어값 MODE\_CNT의 값을 2진수"11"과 비교하여 그 결과 카운터 COUNT1을 그 리셋단 R2를 거쳐서 리셋하고, 카운터 COUNT2를 카운트시킴으로써 실현된다. 도 9는 이 상태를 도시하고 있다.

단계 S111의 처리를 받은 직후에서는, 단계 S103에서 동작모드는 제 2 모드라고 판단되고, 단계 S103으로부터 단계 S105로 처리가 진행된다. 그리고, 선두의 메모리인 제 0 번째의 메모리의 내용이 출력 데이터 DOUT으로써 채용된다. 이것은 멀티플렉서 MUX가 모드신호 MODE의 값이 "1"인 것에 대응하여, 출력 데이터 DOUT로서 데이터 DOUT2를 채용함으로써 실현된다. 도 10은 이 상태를 도시하고 있고, 출력 데이터 DOUT는 메모리 M0이 저장하는 데이터A4를 취한다.

단계 S107, S109는 클럭 CLK에 동기하여, 예를 들면 그 상승의 타이밍을 계기로 하여 실행된다. 우선 단계 S107에서, 선두로부터 2번째의 메모리인 제 1 번째의 메모리 내지 종단의 메모리인 제 (N-1) 번째의 메모리의 내용을, 선두의 메모리인 제 0 번째의 메모리 내지 끝에서 2번째의 메모리인 제 (N-2) 번째의 메모리로 전송한다. 이것은 제어부 CNT1~CNT3이 메모리 M0~M3의 사이에서 데이터를 아래에서 위로 향하는 방향으로 전송함으로써 실현된다. 따라서, 도 10에 있어서 메모리 M1~M3에 저장되어 있던 데이터 A3, A2, A1이, 각각 메모리 M0~M2에 저장되게 된다. 또한 단계 S109에 있어서, 종단의 메모리인 제 (N-1) 번째의 메모리에 입력 데이터의 값이 저장된다. 예를 들면 입력 데이터 DIN의 값으로서 B1이 인가되어 있으면, 메모리 M3에는 데이터B1이 저장된다. 도 11은 이렇게 해서 얻어진 각 메모리 M0~M3의 저장내용을 도시하고 있다.

단계 S109의 실행후에는 제 1 모드의 경우와 마찬가지로 단계 S110~S112가 실행된다. 따라서 도 12 내지 도 14에 도시된 바와 같이 해서 출력 데이터으로써 순차적으로 A3, A2, A1을 얻을 수 있고, 그 한쪽에서 종단의 메모리 M3의 내용은 잇달아 새로운 입력 데이터의 값 B2, B3, B4로 갱신된다.

그리고, 단계 S109의 실행에 의해서 도 14에 표시된 상태가 얻어진 후, 클럭 CLK의 상승시에 단계 S111이 실행된다. 제 2 모드에 있어서 모드 제어값 MODE\_CNT의 값이 "3"을 취하는 것은 메모리 M0을 지정하게 되기 때문이다. 동작모드가 제 1 모드로 옮겨지고, 모드 제어값 MODE\_CNT의 값도 "3"으로부터 "0"으로 리셋된다. 그 후, 도 15에 도시된 바와 같이 입력 데이터 DIN의 새로운 값이 C1로 갱신되고, 출력 데이터 DOUT는 메모리 M3의 내용인 데이터B4를 출력한다(단계 S104). 그리고, 메모리 M0~M2에 저장되어 있던 데이터 B1, B2, B3이 각각 메모리 M1~M3에 저장되고(단계 S106), 그리고, 데이터C1이 메모리 M0으로 저장된다(단계 S108, 도 16).

이렇게 하여, N개(도 3 내지 도 16에 의거하여 말하면 4개)의 값을 조(組)로 한 데이터(A1, A2, A3, A4), (B1, B2, B3, B4), ...를 순차적으로 입력 데이터 DIN으로서 취하면, 출력 데이터 DOUT는 (A4, A3, A2, A1), (B4, B3, B2, B1), ...으로 얻어진다. 즉, N개의 메모리를 구비하는 쌍방향 FIFO에서는 N개의 데이터로 이루어진 조의 복수에 대하여, FIFO의 기능을 연속적으로 발휘할 수 있다. 이것은 N개의 메모리 사이에 있어서의 데이터의 전송방향을 위에서 아래로 향하는 방향과, 아래에서 위로 향하는 방향의 쌍방향을 채용하고, 또한 말하자면 데이터가 스위치백의 형태로써 입출력되기 때문이다. 이하, 데이터의 N개의 조를 패킷이라는 단위로 취급하는 경우도 있다.

또, 쉬프트 레지스터형의 쌍방향 FIFO에서는 도 11 내지 도 13에 도시된 바와 같이, 다른 조의 데이터의 선두가 인접하는 형태를 도시하였지만, 물론 K( $\geq 0$ )개의 기억 단위를 끼워 이들이 대향하는 형태로 하여도 좋은 것은 명백하다. 그 경우에, 필요로 하는 기억소자는 (N+K)개로 되고, 단계 S104, S106, S107, S109, S113에 있어서의 N의 값을 N+K로 전환하여 판독하면 상기의 효과를 얻을 수 있다. 단, 스텝S110의 N의 값은 이 경우에도 유지된다.

쉬프트 레지스터의 일부를 구성하는 메모리 M0~M3의 사이에서 어느 한쪽의 방향에 데이터의 전송이 행해지는지, 입력 데이터를 선두의 메모리 및 종단의 메모리중 어느쪽에 저장할지, 출력 데이터가 선두의 메모리 및 종단의 메모리 중 어디에서 판독되는지가 각각 적절하게 선택되면 상기 쌍방향 FIFO는 실현될 수 있다. 따라서, 제어부 CNT1~CNT3과 디멀티플렉서 DMUX를 구분할 필요는 없다. 이하에서 이러한 구분이 이루어지지 않은 경우의 회로를 도시한다. 단, 정논리("1", "0"이 각각 "H", "L"에 대응한다)가 채용된다.

표 1 내지 표 3은 이 순서대로 결합하여 쌍방향 FIFO를 기술하는 하드웨어 기술언어(HDL)에 의한 리스트를 도시한다. 여기에서는 예로써 Cadence사의 Verilog-HDL을 이용하고 있다.

**[표 1]**

```

module bififo1(clk, rst, en, wt, din, dout );
input          clk, rst, en, wt;
input  [1:0]   din;
output [1:0]   dout;
wire  [1:0]   dout1, dout2;
reg     [1:0]   filo[0:3];
reg                      mode;
reg     [1:0]   mode_cnt;
integer        i;

```

## [표 2]

```

always @(posedge clk or negedge rst) begin
    if(!rst) begin
        for (i = 3; i >= 0; i=i-1) begin
            filo[i] <= 2'h0;
        end
        mode <= 1'b0;
        mode_cnt <= 2'h0;
    end else begin
        if(!wt && en) begin
            case(mode)
            1'b0:begin
                for (i = 3; i > 0; i=i-1) begin
                    filo[i] <= filo[i-1];
                end
                filo[0] <= din;
                mode_cnt <= mode_cnt + 1'b1;
                if(mode_cnt == 2'd3) begin
                    mode <= 1'b1;
                    mode_cnt <= 2'h0;
                end
            end
            1'b1:begin
                for (i = 3; i > 0; i=i-1) begin
                    filo[i-1] <= filo[i];
                end
                filo[3] <= din;
                mode_cnt <= mode_cnt + 1'b1;
                if(mode_cnt == 2'd3) begin
                    mode <= 1'b0;
                    mode_cnt <= 2'h0;
                end
            end
            endcase
        end else if(!en) begin
            mode_cnt <= 2'h0;
        end
    end
end

```

## [표 3]

```

assign dout1 = filo[3];
assign dout2 = filo[0];
assign dout  = (mode == 1'b0) ? dout1 : dout2;

endmodule

```

표 1은 해당 리스트의 제 1 행 내지 제 9 행을, 표 2는 해당 리스트의 제 10 행 내지 제 47 행을, 표 3는 해당 리스트의 제 48 행 내지 제 51 행(최종행)을 도시하고 있다. 표 1에서는 리스트의 제 1 행에서 모듈(module)의 개시가 도시되고, 제 2 행 내지 제 9 행에서 각종의 선언이 실행되고 있다. 여기서는 패킷 하나당 데이터는 4개이고, 각 데이터의 비트폭은 2비트이다.

도 17은 해당 리스트에 기초하여 생성되는 쌍방향 FIL0의 구성을 도시하는 블록도이다. 해당 리스트에

있어서 소문자로 표시된 변수는 이하의 도면에서 대문자로 표시되는 신호, 데이터에 상당한다.

논리회로(110)에는 클럭 CLK, 2비트의 입력 데이터 DIN <1:0>, 리셋신호 RST, 대기(wait)신호 WT, 인에이블 신호 EN이 입력되고, 모드신호 MODE, 메모리 M0이 저장하는 데이터에 대응하는 2비트의 데이터 FILO <0> <0>, FILO <0> <1> (이들은 데이터 DOUT2를 구성한다), 메모리 M3이 저장하는 데이터에 대응하는 2비트의 데이터 FILO <3> <0>, FILO <3> <1> (이들은 데이터 DOUT1을 구성한다)가 출력된다.

인버터G1, G2, 복합게이트 G3, G4는 멀티플렉서 MUX에 대응하고, 모드신호 MODE가 "0"을 취하는 것에 대응하여 FILO <3> <0>, FILO <3> <1>의 쌍이, "1"을 취하는 것에 대응하여 FILO <0> <0>, FILO <0> <1>의 쌍이, 각각 2비트의 출력 데이터 DOUT <1:0>로서 출력된다.

이러한 구성은 표 3에 도시된 리스트의 제 48 행 내지 제 50 행에 있어서의 어사인(assign)문에 의해서 규정되고, 도 1의 단계 S103~S105에 해당한다.

표 2에 도시된 내용은 논리회로(110)를 규정하고 있고, 도 18 내지 도 27은 더불어 논리회로(110)의 구성을 상세하게 도시하는 회로도이다.

리스트 제 10 행의 올웨이즈(always)문은 클럭 CLK의 상승, 또는 리셋신호의 하강에 의해서 제 11 행 내지 제 47 행의 처리가 이루어지는 것을 도시한다. 클럭 CLK에 상관없이, 리셋신호 RST가 하강한 것을 인식하는 제 11 행의 기재는 도 1에 있어서의 단계 S101에 상당하고, 리셋처리인 단계 S113~S115(도 2)는 각각 제 12 행 내지 제 14 행, 제 15 행, 제 16 행에 상당한다.

도 1에 도시된 단계 S102는 표 2에 표시된 리스트 제 18 행에 상당하고, 단계 S102의 판단이 "Y"인 경우에 처리되는 단계 S103~S112는 제 19 행 내지 제 42 행에 상당한다. 그리고, 제 17 행 내지 제 46 행에 있어서 대기(wait)신호 WT가 "1"인 경우에 대한 처리가 기재되어 있지 않기 때문에, 대기(wait)신호 WT가 "1"이면, 단계 S103~S112의 모든 처리가 대기되게 된다. 이것은 단계 S102의 판단이 "N"인 경우에 단계 S100을 거쳐서 단계 S101로 되돌아가는 것에 대응하고 있다. 또한, 제 43 행 내지 제 45 행에 있어서, 인에이블 신호 EN이 "L"인 경우에는 모드 제어값 MODE\_CNT에 2비트의 값 "0"이 인가되는 것이 규정되어 있다. 이것은 단계 S100에 대응하고 있다.

리스트의 제 19 행은 단계 S103에 대응하고 있고, 단계 S106 이후의 처리는 제 20 행 내지 제 30 행에, 단계 S107 이후의 처리는 제 31 행 내지 제 41 행에 규정되어 있다.

구체적으로는 동작모드가 제 1 모드인 경우(모드신호 MODE=0)에는 단계 S106은 제 21 행 내지 제 23 행에, 단계 S108은 제 24 행에, 단계 S110은 제 26 행에, 단계 S111은 제 27 행 내지 제 28 행에, 단계 S112는 제 25 행에, 각각 대응한다. 또, 모드 제어값 MODE\_CNT에 대한 절차 대입(procedure assignment)이 행해지는 제 25 행 및 제 28 행에서는 논블록킹(non-blocking) 대입문이 이용되고 있기 때문에, 리스트에 있어서의 기재의 순서에 상관없이 도 2에 도시된 플로우 차트가 대응된다.

마찬가지로 해서, 동작모드가 제 2 모드인 경우(모드신호 MODE=1)에는 단계 S107은 제 32 행 내지 제 34 행에, 단계 S109는 제 35 행에, 단계 S110은 제 37 행에, 단계 S111은 제 38 행 내지 제 39 행에, 단계 S112는 제 36 행에, 각각 대응한다. 이 동작모드에 있어서도, 모드 제어값 MODE\_CNT에 대한 절차 대입이 행해지는 제 36 행 및 제 39 행에서는 논블록킹(non-blocking) 대입문이 이용되기 때문에 리스트에 있어서의 기재순서에 상관없이 도 2에 도시된 플로우 차트가 대응된다.

도 18은 모드 제어값 MODE\_CNT의 제 0 번째 비트의 값 MODE\_CNT <0>, 및 통상 동작 신호 WTEN을 생성하는 부분을 도시한다. 통상 동작 신호 WTEN은 대기신호 WT가 "L"이고, 인에이블 신호 EN이 "H"인 경우에만 "H"로 되고, 플로우 차트에 의거해서 말하면 그 단계 S102에 있어서 "Y"로 판단되는 것에 대응하고, 또한 리스트에 의거해서 말하면 그 제 19 행 내지 제 42 행에 기재된 처리가 행해지는 것에 대응한다.

D플립플롭 MDCT0은 제 0 번째 비트의 모드 제어값 MODE\_CNT <0>을 출력하여 그 논리반전으로서 신호 X1이 생성된다. 통상 동작 신호 WTEN이 "H"이면 신호 X1이 D플립플롭 MDCT0의 D입력단에 인가되기 때문에, 클럭 CLK가 D플립플롭 MDCT0의 클럭단 T에 인가될 때마다 제 0 번째 비트의 모드 제어값 MODE\_CNT <0>의 값은 "0", "1"을 교대로 출력하게 된다. D플립플롭 MDCT0에는 리셋신호RST의 하강에 의해서 리셋되기 위한 리셋단R도 설치되어 있다.

도 19는 제 1 번째 비트의 모드 제어값 MODE\_CNT <1>를 생성하는 부분을 도시한다. D플립플롭 MDCT1은 제 1 번째 비트의 모드 제어값 MODE\_CNT <1>를 출력하고, 그 값이 "0", "1"을 취하는 것에 대응하여 셀렉터SEL이 각각 A입력단에 인가된 데이터, B입력단에 인가된 데이터를 논리반전시켜 출력하고, D플립플롭 MDCT1의 D입력단에 인가된다. 통상 동작 신호 WTEN이 "H"이면 셀렉터SEL의 A입력단 및 B입력단에는 각각 제 0 번째 비트의 모드 제어값 MODE\_CNT <0>의 논리반전 및 신호 X1의 논리반전(즉, 제 0 번째 비트의 모드 제어값 MODE\_CNT <0>)이 인가된다.

이상과 같이, 도 18 및 도 19로 도시된 구성은 도 3에서 도시된 카운터 COUNT1에 거의 대응하고 있다. 단, 카운터 COUNT1과는 달리, 도 18 및 도 19로 도시된 구성은 비교기 CMP의 피드백을 필요로 하지 않은 4진 카운터이다.

도 20은 모드신호 MODE를 생성하는 부분을 도시한다. 통상 동작 신호 WTEN이 "H"이면 모드 제어값 MODE\_CNT <0>, MODE\_CNT <1>의 논리곱을 취함으로써 모드 제어값 MODE\_CNT가 10진수에서 말하는 값 "3"과 일치하는지 아닌지의 판정을 한다(단계 S110의 처리에 상당함). 또한 단계 S111의 처리를 실행하기 위해, 모드 제어값 MODE\_CNT <0>, MODE\_CNT <1>의 논리적과, 모드신호 MODE와의 배타적 논리합이, D플립플롭 MD의 D입력단에 인가되어, D플립플롭 MD가 모드신호 MODE를 출력한다.

도 21은 메모리 M0의 제 0 번째 비트 및 제 1 번째 비트에 각각 대응하는 D플립플롭 M00, M01을 포함하는 구성을 도시하고 있다. 각각의 D입력단에는 복합게이트G5, G6의 출력이 인가되어 있다. D플립플롭 M00, M01의 각각의 출력FILO <0> <0>, FILO <0> <1>은 메모리 M0이 저장하는 2비트의 정보에 상당한



다.

2비트의 입력 데이터 DIN <0>, DIN <1>은 모두 논리반전되어 각각 복합게이트G5, G6으로 입력된다. 이들의 입력 데이터 DIN <0>, DIN <1>이 논리반전되는 것은 복합게이트G5, G6의 구성상, 그 최종출력단이 NAND게이트로 되어있는 것에 기인한다.

도 22는 메모리 M1의 제 1 번째의 비트에 대응하는 D플립플롭 M11을 포함하는 구성을, 도 23은 메모리 M1의 제 0 번째의 비트에 대응하는 D플립플롭 M10을 포함하는 구성을 각각 도시하고 있다. 각각의 D입력단에는 복합게이트G7, G8의 출력이 인가되고 있다. D플립플롭 M10, M11의 각각의 출력 FILO <1> <0>, FILO <1> <1>은 메모리 M1이 저장하는 2비트의 정보에 상당한다.

도 24는 메모리 M2의 제 1 번째의 비트에 대응하는 D플립플롭 M21을 포함하는 구성을, 도 25는 메모리 M2의 제 0 번째의 비트에 대응하는 D플립플롭 M20을 포함하는 구성을 각각 도시하고 있다. 각각의 D입력단에는 복합게이트 G9, G10의 출력이 인가되어 있다. D플립플롭 M20, M21의 각각의 출력 FILO <2> <0>, FILO <2> <1>은 메모리 M2가 저장하는 2비트의 정보에 상당한다.

도 26은 메모리 M3의 제 1 번째의 비트에 대응하는 D플립플롭 M31을 포함하는 구성을, 도 27은 메모리 M3의 제 0 번째의 비트에 대응하는 D플립플롭 M30을 포함하는 구성을 각각 도시하고 있다. 각각의 D입력단에는 복합게이트G11, G12의 출력이 인가되어 있다. D플립플롭 M30, M31의 각각의 출력FILO <3> <0>, FILO <3> <1>은 메모리 M3이 저장하는 2비트의 정보에 상당한다. 신호 X13, X12는 각각 입력 데이터 DIN <0>, DIN <1>의 논리반전이고, 복합게이트 G12, G11에 입력된다.

복합게이트 G5~G12는 인버터(통상 동작 신호 WTEN이 "H"로 되어 인버터로서 기능하는 NAND게이트를 포함함)와 더불어 도 3에 도시된 제어부CNT1~CNT3과 멀티플렉서 DMUX를 통합한 기능을 실현하고 있다.

또, 이들의 도면에 있어 통상 동작 신호 WTEN이 "H"이면 신호 X2, X4는 각각 모드신호 MODE와 동일한 논리를 취한다. 또한 통상 동작 신호 WTEN의 값에 따르지 않고, 신호 X5, X3, X9, X7, X11, X10, X8, X6은 각각 FILO <0> <0>, FILO <0> <1>, FILO <1> <0>, FILO <1> <1>, FILO <2> <0>, FILO <2> <1>, FILO <3> <0>, FILO <3> <1>의 논리반전이다.

대기신호 WT가 "H"로 되어 대기명령이 인가된 경우에는 통상 동작 신호 WTEN이 "L"로 되고, 복합게이트G5~G12의 동작으로부터 모든 D 플립플롭의 내용은 변경되지 않는다. 또한, 인에이블 신호 EN이 "L"로 되어 인에이블 상태가 아닌 경우에도 통상 동작 신호 WTEN이 "L"로 되고, D플립플롭 MDCT0, MDCT1의 D입력단에 "0"이 인가되어 단계 S100이 실행된다. 또한, 리셋신호RST가 하강해서 리셋명령을 받았을 경우에는 단계 S113~S115의 처리가 이루어진다.

#### (실시예 2)

실시예 2에서는 어드레스 포인터형의 쌍방향 FILO에 대하여 설명한다. 도 28 및 도 29는 양도면이 함께 실시예 2에 대한 쌍방향 FILO의 동작을 도시하는 플로우 차트이다. 양도면은 접속자J5, J7에 의해서 서로 결합된다. 또한, 도 30은 실시예2에 대한 쌍방향 FILO의 구성을 예시하는 블록도이다. 그리고, 도 31 내지 도 43은 도 30에 도시된 쌍방향 FILO의 동작을 순서대로 도시하는 모식도이다.

도 30에 도시된 쌍방향 FILO는 RAM인 메모리R256, 멀티플렉서 MUX1, MUX2, 카운터 COUNT10, COUNT11, COUNT2, 비교기 CMP1, CMP2를 구비하고 있다.

카운터 COUNT10, COUNT11은 카운터 COUNT2에 의해서 생성된 모드신호 MODE의 값이 "0", "1"을 취하는 것에 대응하여 그 카운트방향이 증대방향, 감소방향으로 되도록 제어되고, 각각 8비트의 모드 제어값 W\_MODE\_CNT, R\_MODE\_CNT를 생성한다. 모드 제어값 W\_MODE\_CNT, R\_MODE\_CNT는 메모리R256에 있어서는 각각 기입 어드레스 및 판독 어드레스로 된다.

카운터 COUNT10, COUNT11, COUNT2는 모두 클럭 CLK를 수신하는 클럭단 T를 갖고 있고, 카운터 COUNT10, COUNT11은 클럭 CLK의 상승을 카운트한다. 한편, 카운터 COUNT2는 실시예 1의 경우에 이용된 것과 마찬가지로 클럭 CLK를 카운트하는 것이 아니라 비교기 CMP1, CMP2의 출력을 카운트하는 1비트의 카운터이고, 그 출력인 모드신호 MODE의 값의 변화하는 타이밍이 클럭 CLK에 동기한다. 또한, 이들의 인에이블단 E 모두에 리셋신호RST, 혹은 인에이블 신호 EN이 인가되고, 이들의 신호에 의해서 동작의 여부가 제어된다.

카운터 COUNT10, COUNT11의 초기값은 각각 멀티플렉서 MUX1, MUX2의 출력으로부터 얻어진다. 멀티플렉서 MUX1에는 10진수로 0과 204를 나타내는 값의 8비트신호가 입력되고, 멀티플렉서 MUX2에는 10진수로 203과 1을 나타내는 값의 8비트신호가 입력된다. 멀티플렉서 MUX1, MUX2가 각각 수신한 2개의 입력 중 어느 것을 출력하는지는 비교기 CMP1, CMP2의 출력에 의해서 결정된다.

비교기 CMP1에는 10진수로 1을 나타내는 값과, 모드 제어값 W\_MODE\_CNT가 인가되고, 양자가 같은 경우에는 멀티플렉서 MUX1, MUX2가 각각 "0", "1"을 출력하도록 제어한다. 또한 비교기 CMP2에는 10진수로 203을 나타내는 값과, 모드 제어값 W\_MODE\_CNT가 인가되고, 양자가 같은 경우에는 멀티플렉서 MUX1, MUX2가 각각 "204", "203"을 출력하도록 제어한다.

메모리R256의 기입클럭단 WT 및 판독클럭단RT에는 공통적으로 클럭 CLK가 인가되고, 클럭 CLK의 상승 및 하강에 동기해서 각각 메모리R256에 대한 기입 및 판독이 행해진다.

기입단DI에는 입력 데이터DIN이 인가되고, 판독단DO로부터 메모리R256의 판독데이터를 얻을 수 있다. 도 30에서는 멀티플렉서 MUX3이 추가로 마련되고, 리셋신호RST가 비활성시("1")인 경우에는 메모리R256으로부터의 판독데이터가 쌍방향 FILO의 출력 데이터 DOUT로서 채용되지만, 리셋신호RST가 활성시("0")인 경우에는 10진수에서 말하는 0이 쌍방향 FILO의 출력 데이터 DOUT로서 채용된다. 이 이유에 대해서는 후술된다.

도 31 내지 도 43에서는 (N+1)개의 어드레스0~N에 대응하는 기억 단위 #0~#N을 도시하고 있다. 메모리 R256에 있어서는 기억 단위는 256개가 있고, N=204로 설정되어 있다. 물론, N≤255의 관계가 있으면 다른 값이어도 좋지만, 그 경우에는 멀티플렉서 MUX1, MUX2에 인가되는 값은 각각 204, 203의 대신에 N, N-

1로 된다.

도 28에 도시된 플로우 차트에 있어서, 단계 S201~S204는 클럭 CLK에 상관없이 실행된다. 단계 S201에 있어서는 리셋명령을 수신하였는지 여부(리셋신호RST가 활성화되어 있는지 비활성인지)가 판단되고, 수신했을 경우에는 단계 S215~S219의 리셋처리가 이루어져, 접속자J8을 거쳐서 단계 S201로 되돌아간다. 리셋명령을 수신하지 않으면 단계 S202에서 인에이블 상태인지의 여부(인에이블 신호 EN이 활성화되어 있는지 비활성인지)가 판단되고, 인에이블 상태가 아닌 경우에는 단계 S218~S219의 처리를 받아 단계 S201로 되돌아간다. 단계 S203에서는 대기명령이 인가되어 있는지의 여부(대기신호 WT가 활성화되어 있는지 비활성인지)가 판단되고, 대기명령이 걸려 있지 않은 경우에는 단계 S204로 진행된다.

리셋처리에 대하여 설명하면, 단계 S215에 있어서 RAM인 메모리 R256의 초기화명령을 받았는지의 여부가 판단되고, 초기화 명령을 수신하고 있으면 RAM의 초기화가 실행된다. 구체적으로는 후술하겠지만, 메모리R256의 초기화단S에 인가되는 초기화신호SRAM0이 활성화("0")됨으로써 클럭 CLK의 상승에 동기하여 메모리R256의 모든 어드레스에 있어서 특정한 값, 예를 들면 "0"이 인가된다. 단, 이 때에 메모리R256이 출력하는 판독데이터는 출력 데이터 DOUT로서의 의미를 갖지 않기 때문에, 멀티플렉서 MUX3에 의해서 별도로 특정한 값(도 30에서는 "0"으로 설정되어 초기화에 이용되는 값과 때때로 일치함)가 출력 데이터 DOUT로서 채용된다(단계 S217).

그리고, 단계 S218에 있어서 동작모드가 제 1 모드로 초기 설정된다. 또한 단계 S219에 있어서 기입 어드레스, 판독 어드레스가 각각 "0", "1"로 초기 설정된다. 즉, 카운터 COUNT10, COUNT11은 리셋신호RST의 활성화 혹은 인에이블 신호 EN의 비활성화에 의해서 각각 초기값이 "0", "1"로 설정된다. 단계 S218, S219는 리셋명령이 인가된 경우에도, 인에이블 상태가 아닌 경우에도 실행되고, 그 내용은 기입 어드레스 및 판독 어드레스에 관한 한 동일하기 때문에 카운터 COUNT2, COUNT10, COUNT11의 인에이블단 E에 인가되는 것으로서는 리셋신호RST도 인에이블 신호 EN도 공통적으로 이용된다.

도 31은 리셋직후의 기억 단위 #0~#N의 상태를 보이고 있다. 이들 모두에 대하여 "0"이 저장되어 있다.

단계 S204에 의해서 동작모드가 제 1 모드, 제 2 모드 중 어느 쪽인지가 판단되고, 각각의 경우에 따라 단계 S205, 혹은 접속자J5를 거쳐서 단계 S210으로 분기한다.

제 1 모드란 모드신호 MODE가 "0"인 경우에 상당한다. 단계 S205에서 입력 데이터 DIN이 RAM으로의 기입 데이터로서 채용된다. 기입동작은 클럭 CLK의 상승에 동기하여 실행된다.

단계 S206에서 기입 어드레스가 N-1인지의 여부가 판단된다. 이것은 모드 제어값 W\_MODE\_CNT와 204-1=203을 입력하는 비교기 CMP2에 의해서 실행된다. 기입 어드레스가 N-1이 아닌한 단계 S208에 의해서 기입 어드레스, 판독 어드레스 모두가 1만큼 증가된다. 이것은 카운터 COUNT10, 11에 의해서 실행된다. 단계 S208도 클럭 CLK의 상승을 계기로 하여 실행되지만, 단계 S205보다 나중에 실행된다.

도 32는 리셋처리가 행해진 후에 처음에 클럭 CLK가 상승한 경우의 기억 단위 #0~#N에 대한 처리가 도시되어 있다. 단계 S219에 있어서 기입 어드레스가 "0"으로 설정되어 있었기 때문에, 우선 단계 S205에 따라서 입력 데이터 DIN의 값 A1이 기억 단위 #0에서 기억된다. 그리고, 단계 S208에 따라서 판독 어드레스 및 기입 어드레스가 각각 1 증가하여, 각각 값 1, 2를 취한다. 도 32에 있어서의 파선은 단계 S205의 기입동작을, 흰 화살표는 클럭 CLK의 상승에 의한 어드레스의 변화를 하는 단계 S208의 동작을 각각 도시하고 있다.

단계 S209는 RAM의 판독데이터를 출력 데이터로서 채용하는 단계이고, 클럭 CLK의 하강에 동기하여 실행된다. 단계 S208에 의해서 판독 어드레스는 2로 되어 있기 때문에, 기억 단위 #2에 저장되어 있던 리셋후의 값 "0"이 판독되게 된다. 도 33은 이 상태를 도시하고 있고, 기입 어드레스는 값 1을 취하기는 하지만, 클럭 CLK의 하강에 의해서는 기입동작은 실행되지 않기 때문에, 기억 단위 #1이 저장하는 값은 이 시점에서는 변화하지 않는다.

단계 S201~S205를 거쳐서 단계 S206, S208, S209가 반복되고, 단계 S208에 의해서 기입 어드레스, 판독 어드레스가 각각 N-1, N으로 되었을 경우를 고려한다. 이미 단계 S205에 의해서 (N-1)개의 값 A1~A(N-1)이 순차적으로 기억 단위 #0~#N-1에 저장되어 있다. 도 34는 이 상태에서부터 단계 S209가 실행된 시점의 동작을 도시하고 있다. 이 시점에서는 클럭 CLK가 하강한 직후로, 어드레스#N-1에는 아직 새로운 입력 데이터 DIN은 저장되어 있지 않다. 한편, 출력 데이터 DOUT는 리셋후의 값"0"을 출력한다.

이 후, 접속자J6, 단계 S201~S204를 거쳐서 단계 S205가 실행되고, 기억 단위 #N-1에 값 AN이 기입된다. 도 35의 파선은 이 상태를 도시하고 있다. 기입 어드레스가 (N-1)이기 때문에 단계 S206으로부터 단계 S207로 진행되고, 동작모드가 제 2 모드로 변경되며, 또한 기입 어드레스 및 판독 어드레스가 각각 N, N-1로 설정된다. 도 35의 실선 및 흰 화살표는 이 어드레스의 변경을 나타내고 있다.

이러한 어드레스의 설정은 도 30에서 도시된 비교기 CMP2가 멀티플렉서 MUX1, MUX2에 각각 204, 203을 출력시키는 것에 대응한다. 이에 따라 카운터 COUNT10, 11의 초기값이 다시 설정되게 된다.

단계 S207도 단계 S208과 마찬가지로, 클럭 CLK의 상승을 계기로 하여 단계 S205보다 나중에 실행된다. 도 35에 도시된 바와 같이, N개의 데이터 A1~AN은 순차적으로 기억 단위 #0~#N-1에 저장되어 있다.

이 후, 단계 S209에 있어서는 기억 단위 #N-1에 저장되어 있던 값의 판독이 행해진다. 따라서, 출력 데이터 DOUT로서는, 연속하여 "0"이 얻어진 후의 초기값으로서 값 AN이 얻어지게 된다(도 36).

또한, 접속자J6, 단계 S201~S203을 거쳐서 단계 S204에 도달하면, 동작모드가 제 2 동작모드로 전환되었기 때문에, 단계 S210이 실행된다. 단계 S210에 도달할 때까지는 기입 어드레스에 변동은 없기 때문에, 입력 데이터 DIN의 값B1이 기억 단위 #N에 기입된다. 도 37의 파선은 이 상태를 나타내고 있다.

또한, 단계 S211로 진행되어 기입 어드레스가 1인지의 여부가 판단된다. 이것은 모드 제어값 W\_MODE\_CNT와 "1"을 입력하는 비교기 CMP1에 의해서 실행된다. 기입 어드레스가 1이 아닌 한, 단계 S213에 의해서 기입 어드레스, 판독 어드레스 모두가 1만큼 줄어든다. 이것은 카운터 COUNT10, COUNT11에 의해서 실행된다.

행된다. 단계 S211도 클럭 CLK의 상승을 계기로 하여 실행되지만, 단계 S210보다 나중에 행해진다. 단계 S211의 시점에서는 기입 어드레스는 N이기 때문에, 단계 S213에 의해서 기입 어드레스 및 판독 어드레스는 각각(N-1), (N-2)로 설정된다(도 37의 흰 화살표 및 실선참조).

다음에 단계 S214로 진행되어 단계 S209와 마찬가지로 하여 출력 데이터 DOUT를 얻는다. 도 38은 클럭 CLK의 하강에 동기하여 기억 단위 #N-2로부터 값 A(N-1)을 판독하는 상태를 도시하고 있다.

이렇게 하여 접속자 J7, 단계 S201~S204를 거쳐서 단계 S210~S213, S214가 반복되고, 출력 데이터 DOUT는 값 AN, A(N-1), ...을 순차적으로 출력한다.

단계 S213에 의해서 기입 어드레스, 판독 어드레스가 각각 1, 0이 되었을 경우를 고려한다. 이 후에 클럭 CLK가 하강해서 단계 S214가 실행되면, 도 39에 도시된 바와 같이, 출력 데이터 DOUT로서 값 A1이 얻어진다. 단, 기억 단위 #1에서 저장되어 있는 값은 A2인 채로써 경신되어 있지 않다.

이 후, 접속자 J7, 단계 S201~S204를 거쳐서 단계 S210이 실행되고, 기억 단위 #1에 값BN이 기입된다(도 40의 파선참조). 기입 어드레스가 1이기 때문에 단계 S211로부터 단계 S212로 진행되고, 동작모드가 제 1 모드로 변경되며, 또한, 기입 어드레스 및 판독 어드레스가 각각 0, 1로 설정된다(도 40의 흰 화살표 및 실선참조). 이와 같은 어드레스의 설정은 도 30에서 도시된 비교기 CMP1이, 멀티플렉서 MUX1, MUX2에 각각 0, 1을 출력시키는 것에 대응한다. 이에 따라 카운터 COUNT10, 11의 초기값이 다시 설정되게 된다.

단계 S212도 단계 S213과 마찬가지로, 클럭 CLK의 상승을 계기로 하여 단계 S210보다 나중에 실행된다. 도 40은 이 시점에서의 기억 단위 #0~#N의 상태를 도시하고 있고, N개의 데이터 B1~BN은 순차적으로 기억 단위 #N~#1에 저장되어 있다.

이 후, 단계 S214에 있어서는 기억 단위 #1이 저장되어 있던 값의 판독이 행해진다. 따라서, 출력 데이터 DOUT로서는 값 AN, A(N-1), ..., A1의 후에는 값 BN이 얻어지게 된다(도 41).

또한, 접속자 J7, 단계 S201~S203을 거쳐서 단계 S204에 도달하면, 동작모드가 제 1 동작모드로 전환되었기 때문에, 단계 S205가 실행된다. 단계 S205에 도달할 때까지는 기입 어드레스에 변동은 없기 때문에, 입력 데이터 DIN의 값C1이 기억 단위 #0에 기입된다. 도 42의 파선은 이 상태를 나타내고 있다.

그리고, 단계 S206으로부터 단계 S208로 진행되어 기입 어드레스 및 판독 어드레스가 각각 1, 2로 설정된다. 도 42의 흰 화살표는 단계 S208의 동작을 도시하고 있다.

이상과 같이 하여 (N+1)개의 기억소자를 이용함으로써, N개(도 30에 의거해서 말하면 204개)의 값을 한 조로 한 데이터(A1, A2, ..., AN), (B1, B2, ..., BN), ...을 순차적으로 입력 데이터 DIN으로서 입력하면, 출력 데이터 DOUT는 (AN, ..., A2, A1), (BN, ..., B2, B1), ...으로 얻어진다. 즉, 적어도 (N+1)개의 어드레스를 갖는 RAM을 이용하여, N개의 데이터로 이루어진 조의 복수개에 대하여, FILO의 기능을 연속적으로 발휘시킬 수 있다.

이것은, 기억 단위가 저장하는 내용을 판독하면서 이것에 따라서 입력 데이터가 기입되고, 입력 데이터의 조의 갱신에 대응하여 판독 어드레스 및 기입 어드레스가 이동하는 방향이 위에서 아래로 향하는 방향과, 아래에서 위로 향하는 방향으로 역전되기 때문이다.

이상과 같이, 실시예 2에 의하면, 종래의 RAM을 이용하여 쌍방향 FILO를 실현할 수 있다.

또한, 본 실시예에 있어서는 모드 제어값 W\_MODE\_CNT, R\_MODE\_CNT가 하나 어긋나는 형태를 나타냈지만, 물론 0 이상의 정수 K만큼 어긋나도 좋은 것은 명백하다. 그 경우에는, 필요로 하는 기억소자 (N+K)이고, 단계 S207에 있어서는 기입 어드레스가 (N+K-1)로 설정되며, 단계 S211에 있어서는 기입 어드레스가 K인지의 여부가 판단되고, 단계 S212, S219에 있어서는 판독 어드레스가 K로 설정된다. K=0이라고 해도 기입 및 판독의 타이밍이 각각 클럭 CLK의 상승 및 하강에서 실행되기 때문에 상기의 효과를 얻을 수 있다.

표 4 내지 표 8은 이 순서대로 결합하여 실시예 2에 있어서는 쌍방향 FILO를 기술하는 하드웨어 기술언어(HDL)에 의한 리스트를 도시한다. 여기에서도 표 1 내지 표 3과 마찬가지로 Cadence사의 Verilog HDL을 이용하고 있고, N=204, K=1로 설정되어 있다.

[표 4]

```
module bifilol(clk, rst, en, wt, din, dout, sram);
    input          clk, rst, en, wt, sram;
    input  [7:0]    din;
    output [7:0]    dout;
    wire  [7:0]    dout1, dout2, dout3, memo;
    reg   [7:0]    filo[0:203];
    reg                               mode;
    reg   [7:0]    w_mode_cnt, r_mode_cnt;
    reg   [7:0]    dout;
    integer        i;
```

[H 5]

```

always @(posedge clk) begin
    if(!rst) begin
        mode <= 1'b0;
        w_mode_cnt <= 8'h0;
        r_mode_cnt <= 8'h1;
    end else begin
        if(!wt && en) begin
            case(mode)
            1'b0:begin
                w_mode_cnt <= w_mode_cnt + 1'b1;
                r_mode_cnt <= r_mode_cnt + 1'b1;
                if(w_mode_cnt == 8'd203) begin
                    mode <= 1'b1;
                    w_mode_cnt <= 8'd204;
                    r_mode_cnt <= 8'd203;
                end
            end
            1'b1:begin
                w_mode_cnt <= w_mode_cnt - 1'b1;
                r_mode_cnt <= r_mode_cnt - 1'b1;
                if(w_mode_cnt == 8'd1) begin
                    mode <= 1'b0;
                    w_mode_cnt <= 8'h0;
                    r_mode_cnt <= 8'd1;
                end
            end
            endcase
        end else if(!en) begin
            w_mode_cnt <= 8'h0;
            r_mode_cnt <= 8'h1;
            mode <= 1'b0;
        end
    end
end

```

[H 6]

```

always @(posedge clk) begin
    if(!rst) begin
        dout <= 8'h0;
    end else begin
        if(!wt) begin
            dout <= memo;
        end
    end
end

end
RAM256 mem256 ( .TO(clk), .TI(clk), .AO(w_mode_cnt), .AI(r_mode_cnt),
                .DIO(din), .D01(memo), .rst(rst), .sram(sram) );
endmodule

```

[H 7]

```

module RAM256 ( T0, T1, A0, A1, DIO, D01, rst, sram )

    input          T0;          // WRITE CLOCK
    input          T1;          // READ CLOCK
    input[7:0]     A0;          // WRITE ADDRESS
    input[7:0]     A1;          // READ ADDRESS
    input[7:0]     DIO;         // WRITE DATA
    input          rst, sram;
    output[7:0]    D01;         // READ DATA
    reg[7:0]       WP, RP;

always @(posedge T0) begin
    if(!rst && sram) begin
        {WP,RP} <= 16'h0;
    end else if(!sram) begin
        RP <= 8'h0;
        if(WP == 8'd255) begin
            WP <= 8'h0;
        end else begin
            WP <= WP+1'b1;
        end
    end
end

end

wire[7:0] RPRAM = (sram == 1'b0) ? RP : A1;
wire[7:0] WPRAM = (sram == 1'b0) ? WP : A0;
    RAM256C r256(T0, T1, WPRAM, RPRAM, DIO, D01);
endmodule

```

## [표 8]

```

module RAM256C ( TO, T1, AO, A1, DIO, D01 );
    input          TO;      // WRITE CLOCK
    input          T1;      // READ CLOCK
    input[7:0]     AO;      // WRITE ADDRESS
    input[7:0]     A1;      // READ ADDRESS
    input[7:0]     DIO;     // WRITE DATA
    output[7:0]    D01;     // READ DATA
    reg[7:0]       D01;
    reg[7:0]       MEMORY[255:0];
    always @(negedge T1)
        D01 <= MEMORY[A1];
    always @(posedge TO)
        MEMORY[AO] <= DIO[7:0];
endmodule

```

표 4는 해당 리스트의 제 1 행 내지 제 10 행을, 표 5는 해당 리스트의 제 11 행 내지 제 44 행을, 표 6은 해당 리스트의 제 45 행 내지 제 56 행을, 표 7은 해당 리스트의 제 57 행 내지 제 81 행을, 표 8은 해당 리스트의 제 82 행 내지 제 95 행을 각각 도시하고 있다.

표 4 내지 표 6에서는 "module bifilo1"이, 표 7에서는 "module RAM256"이, 표 8에서는 "module RAM256C"가 각각 기재되어 있다.

표 4에서는 리스트의 제 1 행에 있어서 모듈의 개시가 도시되고, 제 2 행 내지 제 10 행에 있어서 각종의 선언이 실행되고 있다. 표 7에서는 리스트의 제 57 행에 있어서 모듈의 개시가 도시되고, 제 58 행 내지 제 65 행에 있어서 각종의 선언이 실행되고 있다. 표 8에서는 리스트의 제 82 행에 있어서 모듈의 개시가 도시되고, 제 83 행 내지 제 90 행에 있어서 각종의 선언이 실행되고 있다.

표 8에 도시되는 "module RAM256C"는, 클럭 CLK의 상승에서 기입이, 하강에서 판독이 각각 실행되는 RAM의 코어부분(도 31 내지 도 43에 도시된 기억 단위에 상당하는 부분을 포함함)을 규정하고, 표 7은 리셋처리의 일부 및 초기화처리를 규정하고 있다.

단계 S201을 거쳐서 실행되는 단계 S215~단계 S216은 리스트 제 69 행 내지 제 76 행에 도시된 if-else 문에 있어서의 처리와, 변수 RP, WP에 의해 각각 지정되는 판독 어드레스 및 기입 어드레스를 거쳐서 초기화를 하는 리스트 제 78 행 내지 제 80 행이 상당한다.

리셋처리에 관해서는 이 밖에, 단계 S217이 리스트 제 47 행(표 6)에 상당하고, 단계 S201을 경유하여 실행되는 단계 S218~S219가 리스트 제 12 행 내지 제 15 행(표 5)에 상당하고 있다.

또한, 단계 S202를 경유하여 실행되는 단계 S218~S219는 리스트 제 38 행 내지 제 41 행(표 5)에 상당하고 있다. 또한, 단계 S203에 있어서 "Y"라고 판단되는 경우는 리스트 제 49 행 내지 제 51 행(표 6)에 상당하여 출력 데이터가 갱신되지 않고, "N"이라고 판단되는 경우는 리스트 제 18 행 내지 제 37 행의 case문(표 5)에 규정되는 처리가 실행된다.

리스트 제 18 행 내지 제 37 행의 case문에 대해서는 리스트 제 18 행이 단계 S204에 상당하고, 리스트 제 19 행 내지 제 27 행이 단계 S206~S208에 상당하며, 리스트 제 28행 내지 제 36행이 단계 S211~S213에 상당한다. 단계 S205, S209, S210, S214에 있어서의 RAM으로의 판독·기입은 리스트 제 78 행 내지 제 79 행에 있어서의 대입문과, 제 80 행에서 참조되는, 표 8에 도시된 "module RAM256C"에 의해서 규정되고 있다.

리스트 제 19 행 내지 제 27 행에 대해서는 단계 S206이 리스트 제 22 행에, 단계 S207이 리스트 제 23 행 내지 제 25 행에, 단계 S208이 리스트 제 20 행 내지 제 21 행에 각각 상당한다.

리스트 제 28 행 내지 제 36 행에 대해서는 단계 S211이 리스트 제 31 행에, 단계 S212가 리스트 제 32 행 내지 제 34 행에, 단계 S213이 리스트 제 29 행 내지 제 30 행에 각각 상당한다.

도 44 내지 도 52는 더불어서 표 4 내지 표 8에서 도시된 HDL의 내용에 기초하여 생성된 쌍방향 FIFO의 구성을 상세하게 도시하는 회로도이다.

도 44에 도시되는 회로는 인에이블 신호 EN, 리셋신호RST, 대기신호 WT, 8비트의 모드 제어값 W\_MODE\_CNT <0:7>, 및 도 45에 도시되는 회로에서 생성되는 신호 Y3을 입력하고, 신호 Y1, Y2, Y5, Y6, W 1~W8을 출력한다. 여기에서 신호 W2는 리셋신호RST와 상보적인 값을 취하고, 신호 W4는 W\_MODE\_CNT <3> 과 상보적인 값을 취하며, 신호 W5는 W\_MODE\_CNT <1> 과 상보적인 값을 취하고, 신호 W6은 W\_MODE\_CNT <

6> 과 상보적인 값을 취하고, 신호 W7은 W\_MODE\_CNT <7> 과 상보적인 값을 취하며, 신호 W8은 W\_MODE\_CNT <0> 과 상보적인 값을 취하고, 신호 Y1은 인에이블 신호 EN0이 "0", "1"의 값을 취하는 데에 따라서 각각 "1", /RST+/WT를 (/는 논리반전을 나타냄)를 취하고, 신호 Y2는 인에이블 신호 EN0이 "0", "1"의 값을 취하는 데에 따라서 각각 "0", RST·/WT를 취한다는 특징이 있다.

도 45에 도시되는 회로는 도 44에 도시되는 회로에서 생성되는 신호 Y1, Y2, Y5, W3~W7, 및 8비트의 모드 제어값 W\_MODE\_CNT <0:7> 및 클럭 CLK를 입력하고, 신호 Y3, Y4, Y7, Y8, 및 8비트의 데이터 DT1 <0:7> 및 모드신호 MODE를 출력한다.

이 도면에 있어서, 0플립플롭 MD는 모드신호 MODE를 출력하고, 도 30에 있어서의 COUNT2에 상당한다. 가감산기(201)는 모드신호 MODE에 의존하여 모드 제어값 W\_MODE\_CNT에 대하여 1 가산할지 감산할지가 제어된다. 또한, 신호 Y7은 모드신호 MODE와 상보적인 값을 취한다.

도 46에 도시되는 회로는 도 44에 도시되는 회로에서 생성되는 신호 Y6, 도 45에 도시되는 회로에서 생성되는 신호 Y4, Y7 및 모드신호 MODE 및 8비트의 모드 제어값 R\_MODE\_CNT <0:7> 을 입력하고, 신호 Y11 및 8비트의 데이터 DT2 <0:7> 을 출력한다. 이 도면에 있어, 가감산기(202)는 모드신호 MODE에 의존하여 모드 제어값 R\_MODE\_CNT에 대하여 1 가산할지 감산할지가 제어된다.

도 47에 도시되는 회로는 신호 Y1, Y2, Y6~Y9, W4~W8, 및 모드 제어값 W\_MODE\_CNT <3:5> 및 데이터 DT1 <0:7> 을 입력하고, 신호 Y9, Y10, N397~N404를 출력한다. 또한 도 48에 도시되는 회로는 신호 Y1, Y2, Y9~Y11, 및 모드 제어값 R\_MODE\_CNT <0:7> 및 데이터 DT2 <0:7> 를 입력하고, 신호 N405~N412를 출력한다.

도 49에 도시되는 회로는 RAM인 메모리 R256과, 신호 N397~N404로부터 모드 제어값 W\_MODE\_CNT를 생성하는 회로 W\_CNT와, 신호 N405~N412로부터 모드 제어값 R\_MODE\_CNT를 생성하는 회로 R\_CNT로 구성되어 있다. 모드 제어값 W\_MODE\_CNT, R\_MODE\_CNT는 각각 메모리 R256의 기입 어드레스단 WA 및 판독 어드레스단 RA에 인가된다. 또한 8비트의 입력 데이터 DIN <7:0> 은 기입 단에 인가되고, 판독단 D0로부터 8비트의 신호 MEMO <7:0> 이 판독된다. 클럭 CLK는 회로 R\_CNT, W\_CNT 및 메모리 R256의 기입클럭단 WT 및 판독클럭단 RT에 인가된다. 또한 리셋신호 RST 및 초기화신호 SRAM이 R256의 리셋단 R 및 초기화단 S에 각각 인가된다.

도 50에 도시된 회로는 도 30에 도시된 멀티플렉서 MUX3에 상당하고, 출력 데이터 DOUT <7:0> 을 출력하는 0플립플롭 F0~F7을 구비하고 있다. 이들은 메모리 R256으로부터 얻어진 신호 MEMO <7:0> 과, 클럭 CLK와, 대기신호 WT와, 신호 W2에 의해서 제어된다.

도 51은 회로 W\_CNT의 구성을 도시하고 있고, 0플립플롭 W0~W7을 구비하고 있다. 0플립플롭 Wp는 모두 클럭 CLK에 의해서 제어되고, 신호 N(404-p)을 수신하여 모드 제어값 W\_MODE\_CNT <p> 를 출력한다(p=0~7).

도 52는 회로 R\_CNT의 구성을 도시하고 있고, 0플립플롭 R0~R7을 구비하고 있다. 0플립플롭 Rq는 모두 클럭 CLK에 의해서 제어되고, 신호 N(412-q)을 모드 제어값 R\_MODE\_CNT <q> 를 출력한다(q=0~7).

### 발명의 효과

본 발명 중 청구항 1에 대한 쌍방향 전송형 기억장치에 의하면, 한 쌍의 입력 데이터가 스위치백하여 출력되는 방향은 서로 역방향이기 때문에, (N+K)개의 기억 단위를 사용하여 N개의 데이터로 이루어지는 조의 복수개에 대하여 FIFO의 기능을 연속적으로 발휘할 수 있다.

본 발명 중 청구항 2에 대한 쌍방향 전송형 기억장치에 의하면, 어느 방향을 따라서 기억 단위에 데이터가 기입되면, 역방향으로부터 판독되고, 또한 이 판독에 따라서 새로운 데이터의 기입이 행해진다. 따라서, N개의 데이터로 이루어지는 조의 복수개에 대하여 FIFO의 기능을 연속적으로 발휘할 수 있다.

본 발명 중 청구항 3에 대한 메모리의 입출력 제어방법에 의하면, 제어신호가 제 1 값을 취하고 있는 경우에는 제 1 내지 제 N의 기억 단위로, 어느 조의 데이터 A1, A2, ..., A(N-1), AN이 순차적으로 기억된다. 그리고, 제어신호는 제 2 값으로 변경되고, 제 N 내지 제 1 기억 단위로부터 데이터 AN, A(N-1), ..., A2, A1이 순차적으로 판독되면서, 제 (N+K) 내지 제 (K+1)의 기억 단위에는 다음 조의 데이터 B1, B2, ..., B(N-1), BN이 순차적으로 입력된다. 그리고, 제어신호는 제 1 값으로 변경되고, 제 (K+1) 내지 제 (N+K)의 기억 단위로부터 데이터 BN, B(N-1), ..., B2, B1이 순차적으로 판독된다.

이와 같이 기억 단위 사이에 있어서의 데이터의 입출력을 어드레스의 조작에 의해서 서로 반대의 쌍방향으로 행하기 때문에, (N+K)개의 기억 단위를 갖는 통상의 메모리를 이용하여, N개의 데이터로 이루어지는 조의 복수개에 대하여 FIFO의 기능을 연속적으로 발휘할 수 있다.

### (57) 청구의 범위

#### 청구항 1

모두가 소정길이의 데이터를 저장하는 제 0 내지 제 (N+K-1)(N≥2, K≥0)의 기억 단위를 포함하고,

입력 데이터는 N개마다 조를 이루어 교대로 상기 제 0 내지 제 (N+K-1)의 기억 단위로 입력되고,

상기 입력 데이터의 서로 다른 2개의 상기 조의 선두의 데이터가 K개의 상기 기억 단위를 사이에 두고 대향하면서, 상기 제 0 내지 제 (N+K-1)의 기억 단위에서 저장되는 데이터에 대하여 순차적으로 (N+K-1)회의 쉬프트 전송이 실행되며,

상기 쉬프트 전송의 방향이 교대로 역방향으로 전향됨으로써 상기 입력 데이터가 스위치백하여 출력되는 쌍방향 전송형 기억장치.



**청구항 2**

모두가 소정길이의 데이터를 저장하고, 제 0 내지 제  $(N+K-1)$ 의 어드레스에 대응하는 제 0 내지 제  $(N+K-1)$  ( $N \geq 2, K \geq 0$ )의 기억 단위를 가지며, 클럭에 기초하여 동작하는 메모리와,

상기 메모리의 판독 어드레스 및 기입 어드레스를 각각 생성하는 어드레스 생성부를 포함하며,

입력 데이터는 N개마다 조를 이루어 상기 기억 단위에 저장되는 내용을 판독하면서 이것에 따라서 입력 데이터가 기입되고, 상기 입력 데이터의 조의 갱신에 대응하여 상기 판독 어드레스 및 상기 기입 어드레스가 이동하는 방향이 역전하는 쌍방향 전송형 기억장치.

**청구항 3**

모두가 소정길이의 데이터를 저장하는 제 1 내지 제  $(N+K)$  ( $N \geq 2, K \geq 0$ )의 기억 단위를 가지며, 클럭에 기초하여 동작하는 메모리의 입출력 제어방법에 있어서,

제어신호가 제 1 값을 취할 경우에는 상기 메모리의 판독 어드레스가 기입 어드레스에 대하여 상기 K만큼 크게 설정되면서, 상기 클럭에 기초하여 상기 판독 어드레스 및 상기 기입 어드레스 모두가 1 증가하고,

상기 제어신호가 제 2 값을 취할 경우에는 상기 판독 어드레스가 상기 기입 어드레스에 대하여 상기 K만큼 작게 설정되면서, 상기 클럭에 기초하여 상기 판독 어드레스 및 기입 어드레스 모두가 1 감소하며,

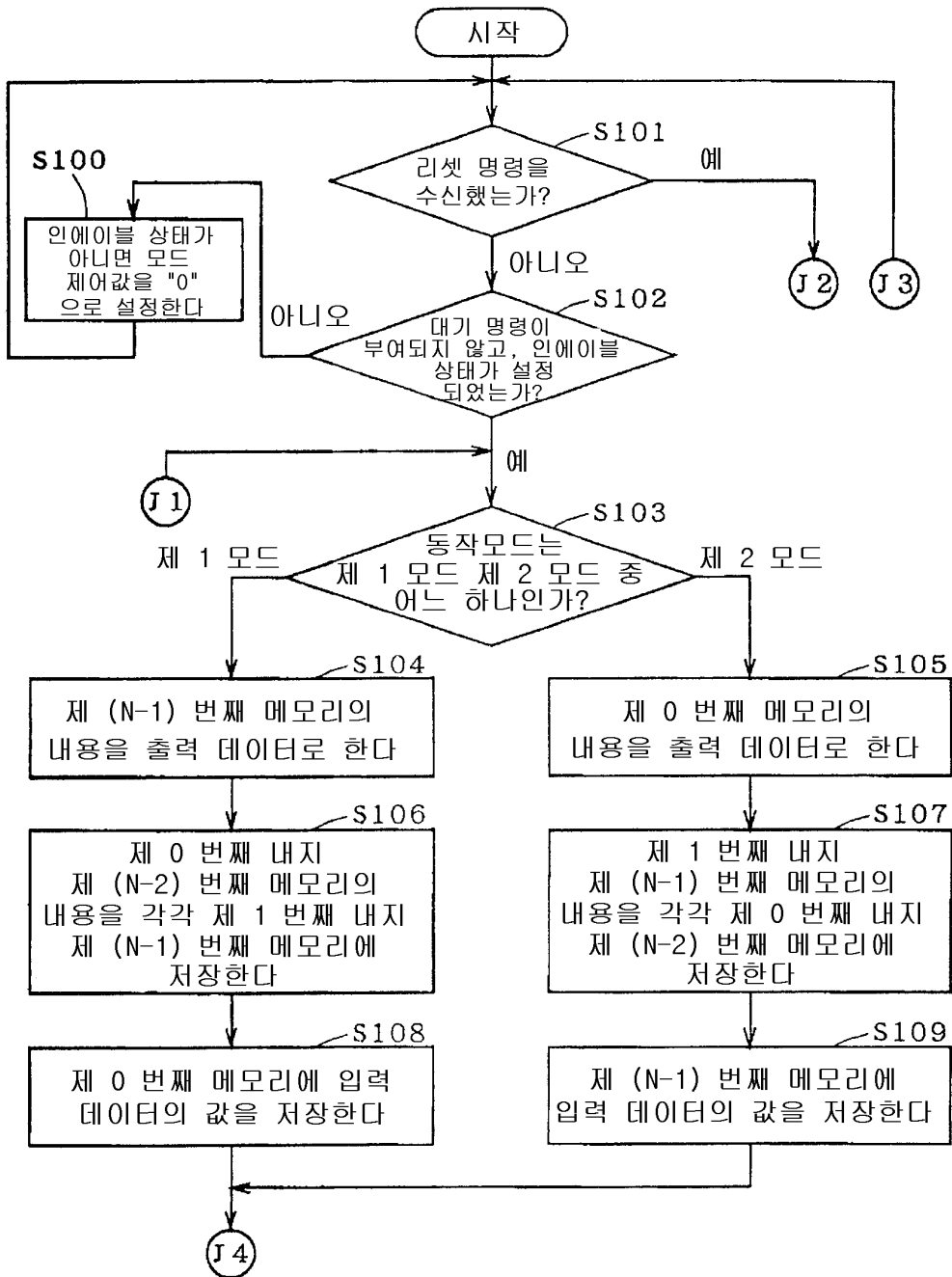
상기 제어신호는 상기 기입 어드레스가 상기 제 N의 기억 단위를 지정한 경우에 대응하여 상기 제 2 값으로, 상기 제  $(K+1)$ 의 기억 단위를 지정한 경우에 대응하여 상기 제 1 값으로 각각 설정되고,

상기 제  $(N-1)$ 의 기억 단위에 대하여 상기 기입이 실행된 후, 상기 기입 어드레스 및 상기 판독 어드레스는 각각 상기 제  $(N+K-1)$ 의 어드레스, 및 상기 제  $(N-1)$ 의 어드레스에 각각 설정되며,

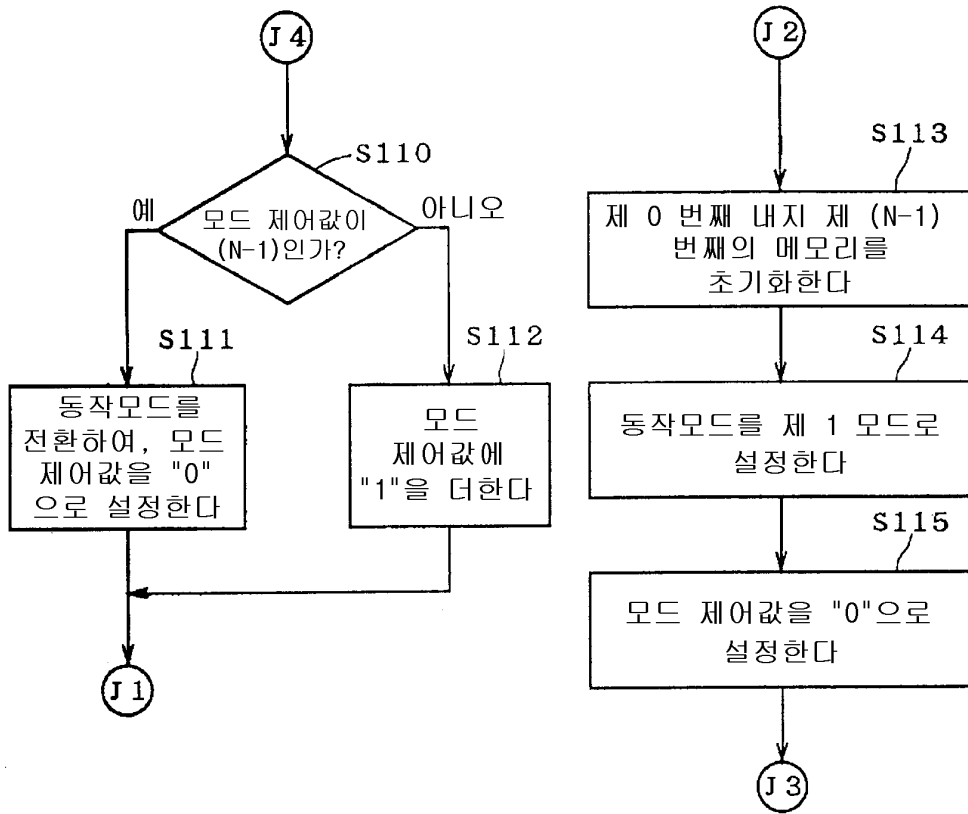
상기 제 K의 기억 단위에 대하여 상기 기입이 실행된 후, 상기 기입 어드레스 및 상기 판독 어드레스는 각각 상기 제 0 어드레스, 및 상기 제 K 어드레스에 각각 설정되는 메모리의 입출력 제어방법.

**도면**

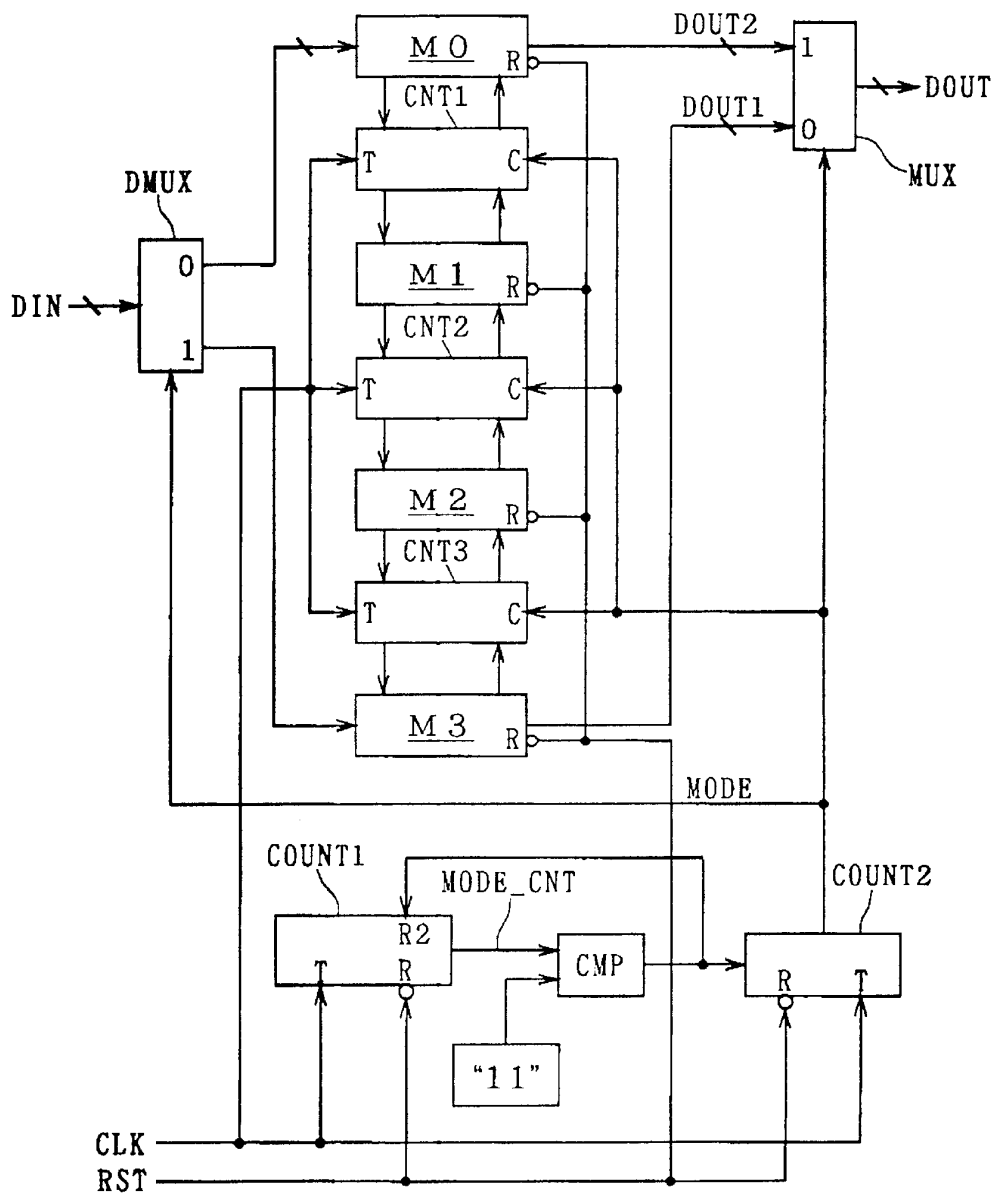
도면1



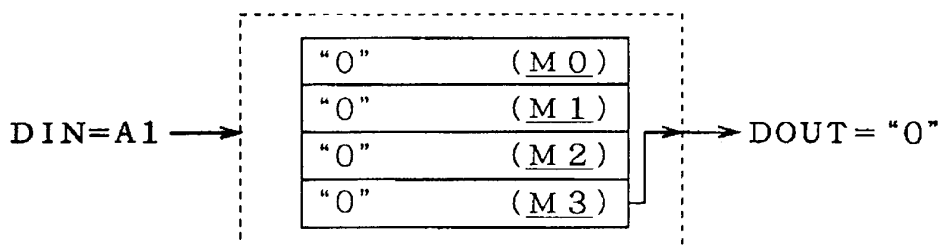
도면2



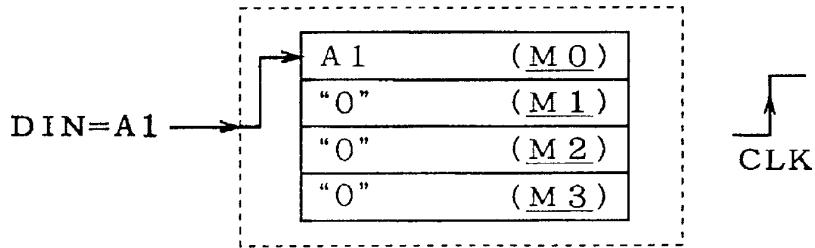
도면3



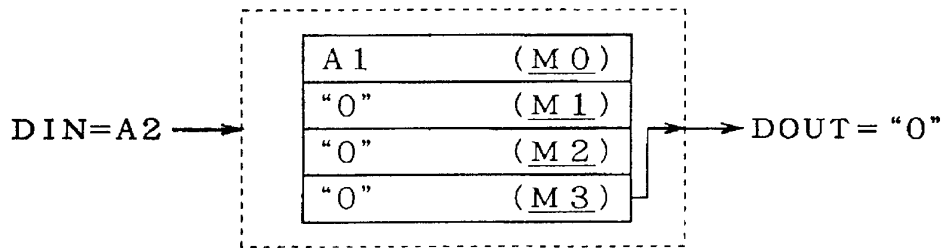
도면4



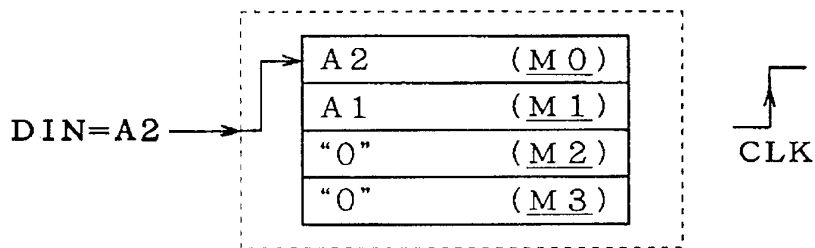
도면5



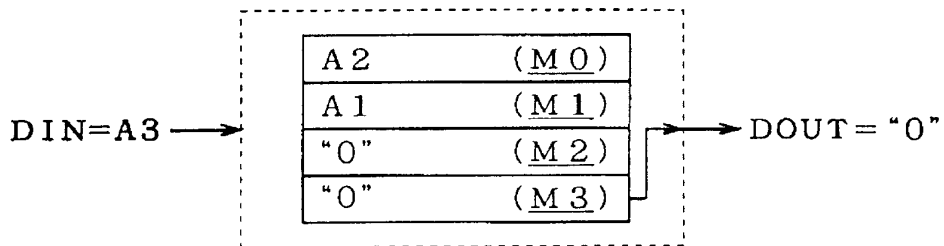
도면6



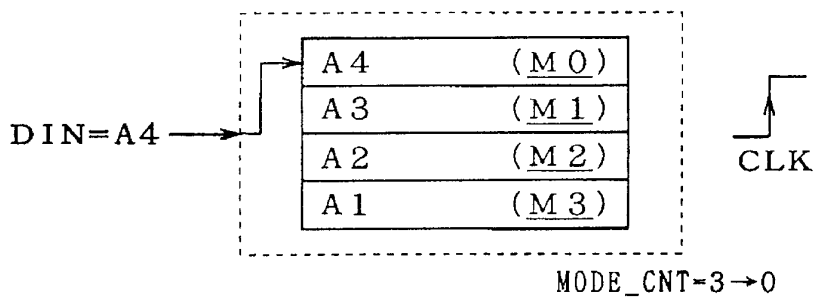
도면7



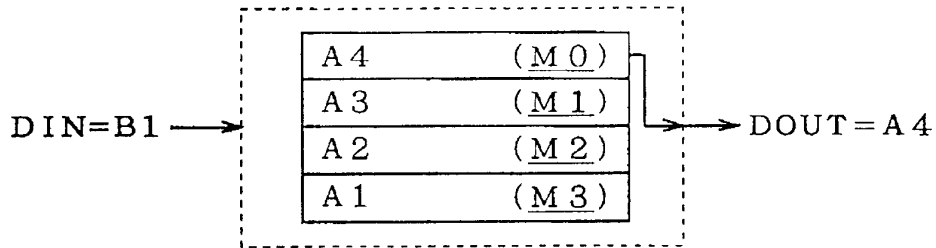
도면8



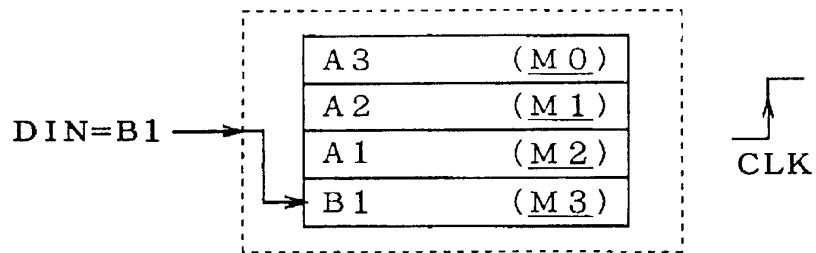
도면9



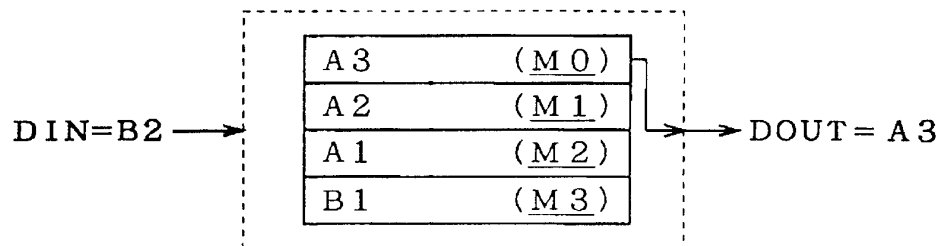
도면10



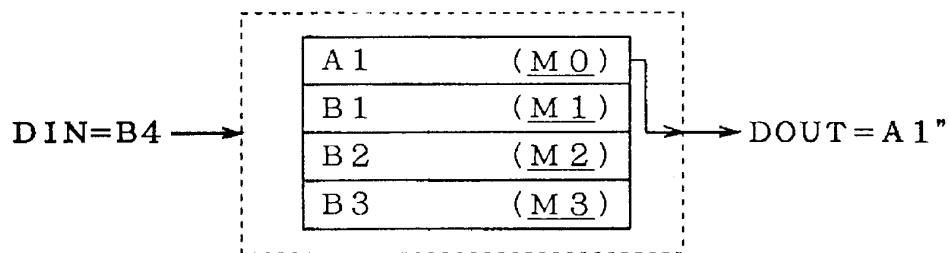
도면11



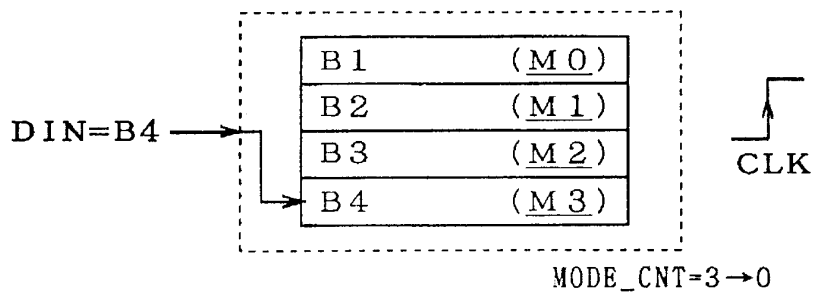
도면12



도면13

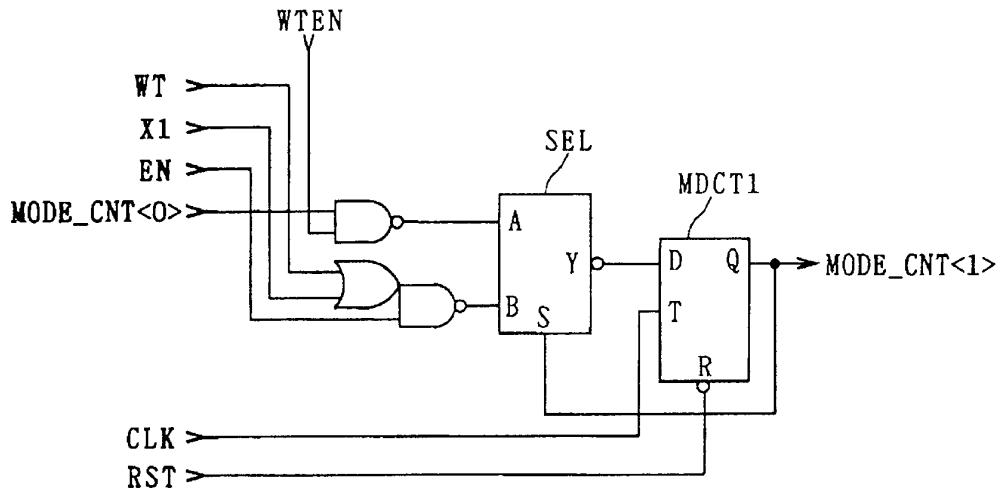


도면14

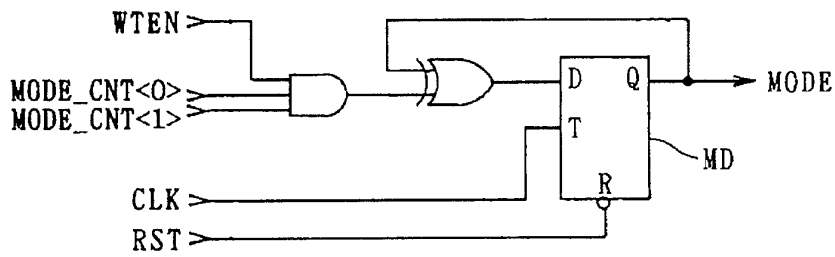




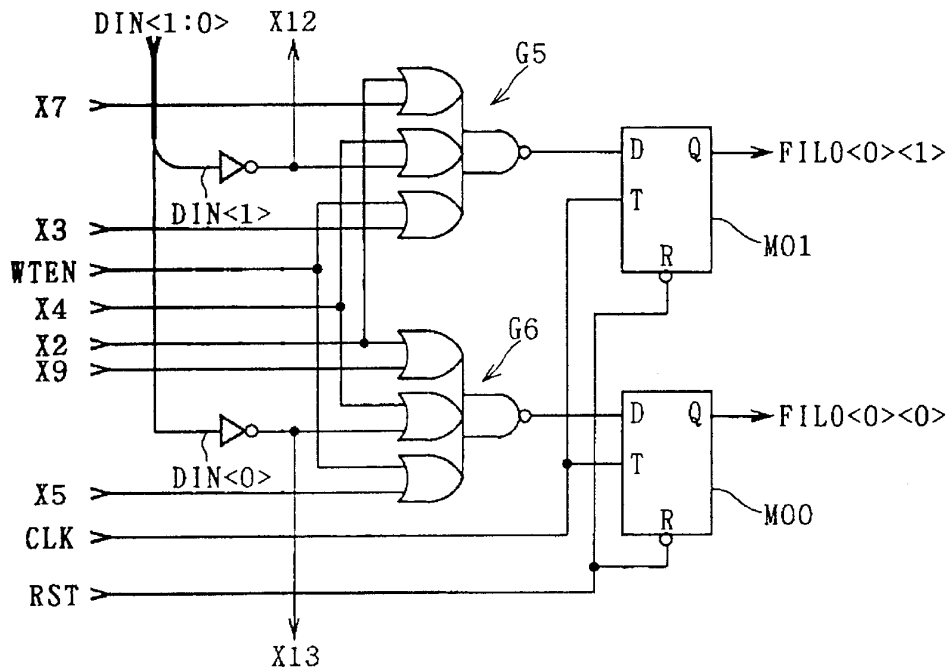
도면19



도면20

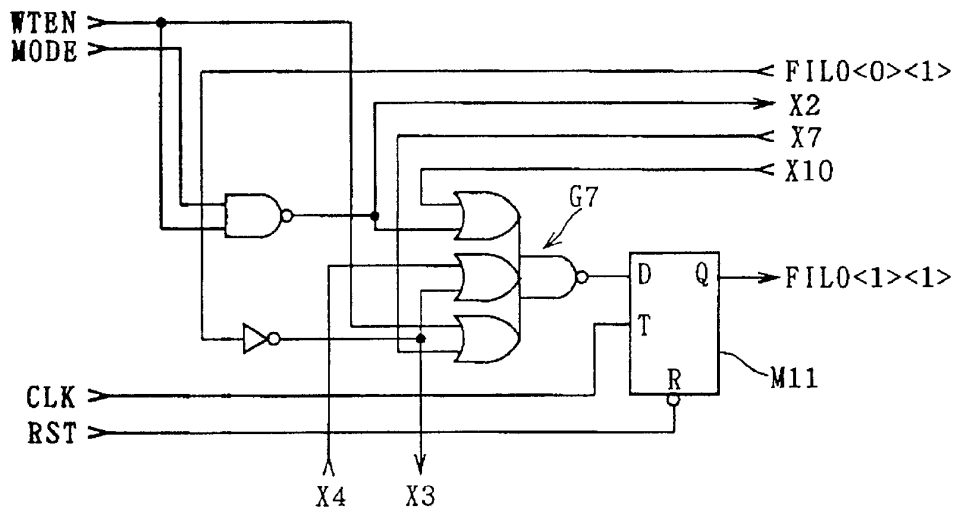


도면21

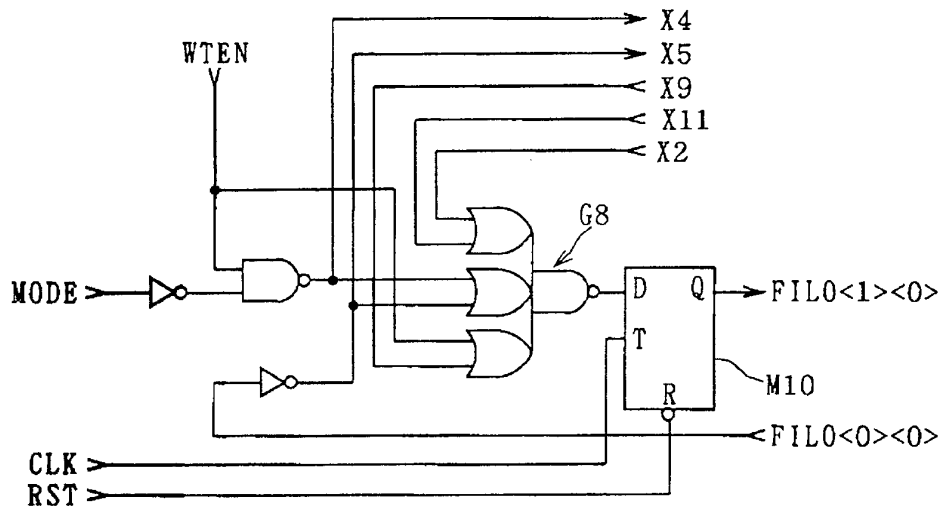




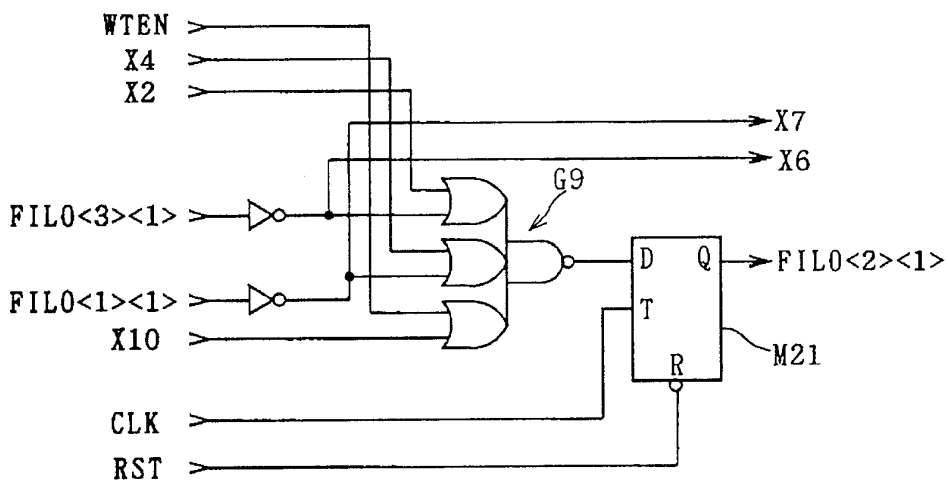
도면22



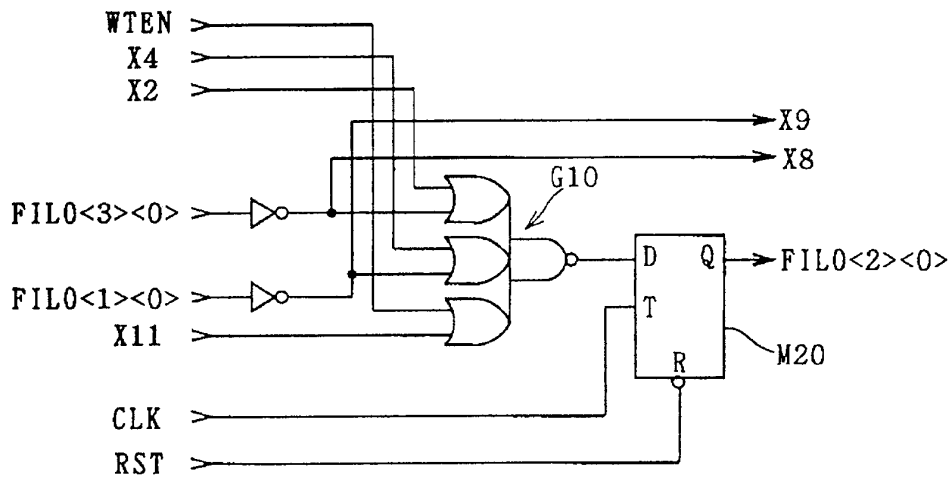
도면23



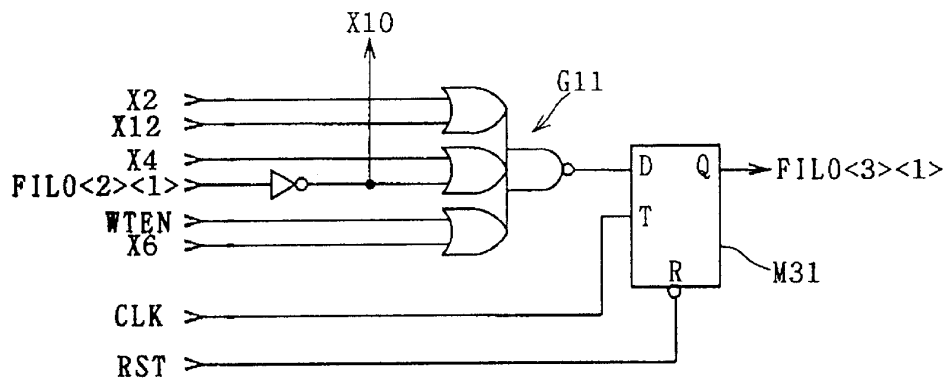
도면24



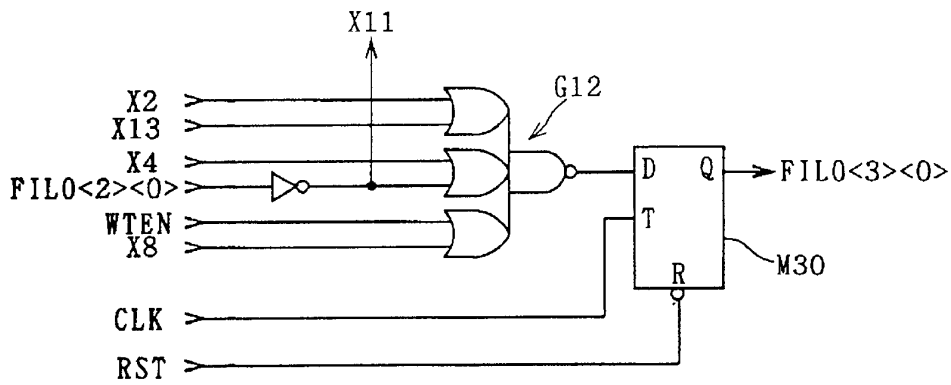
도면25



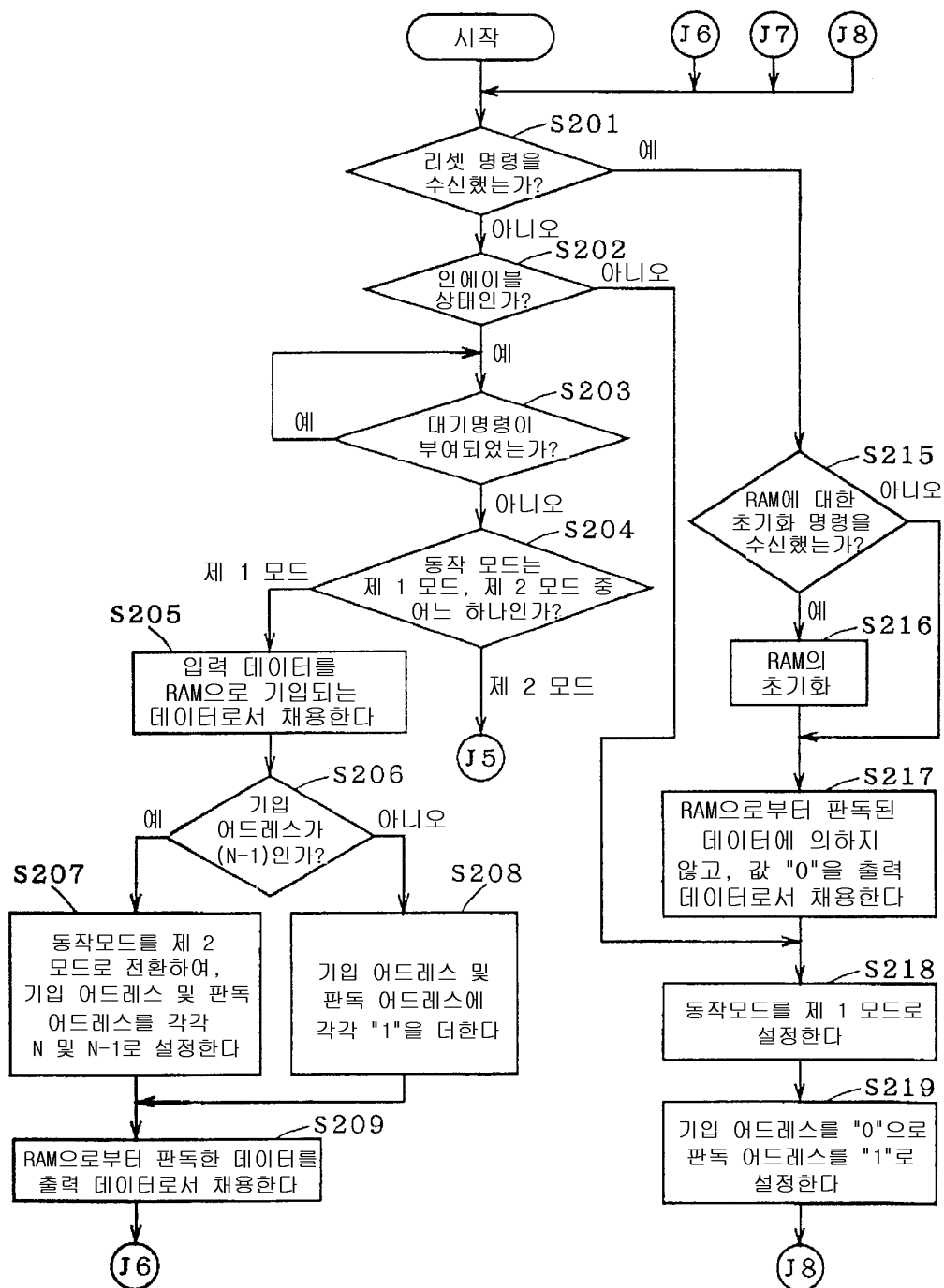
도면26



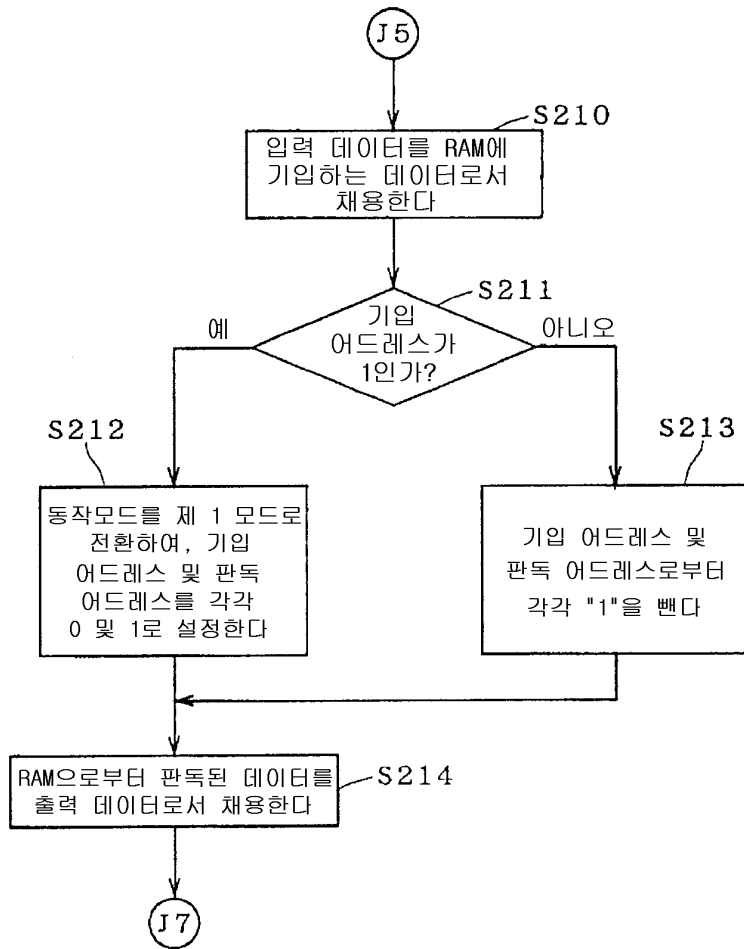
도면27



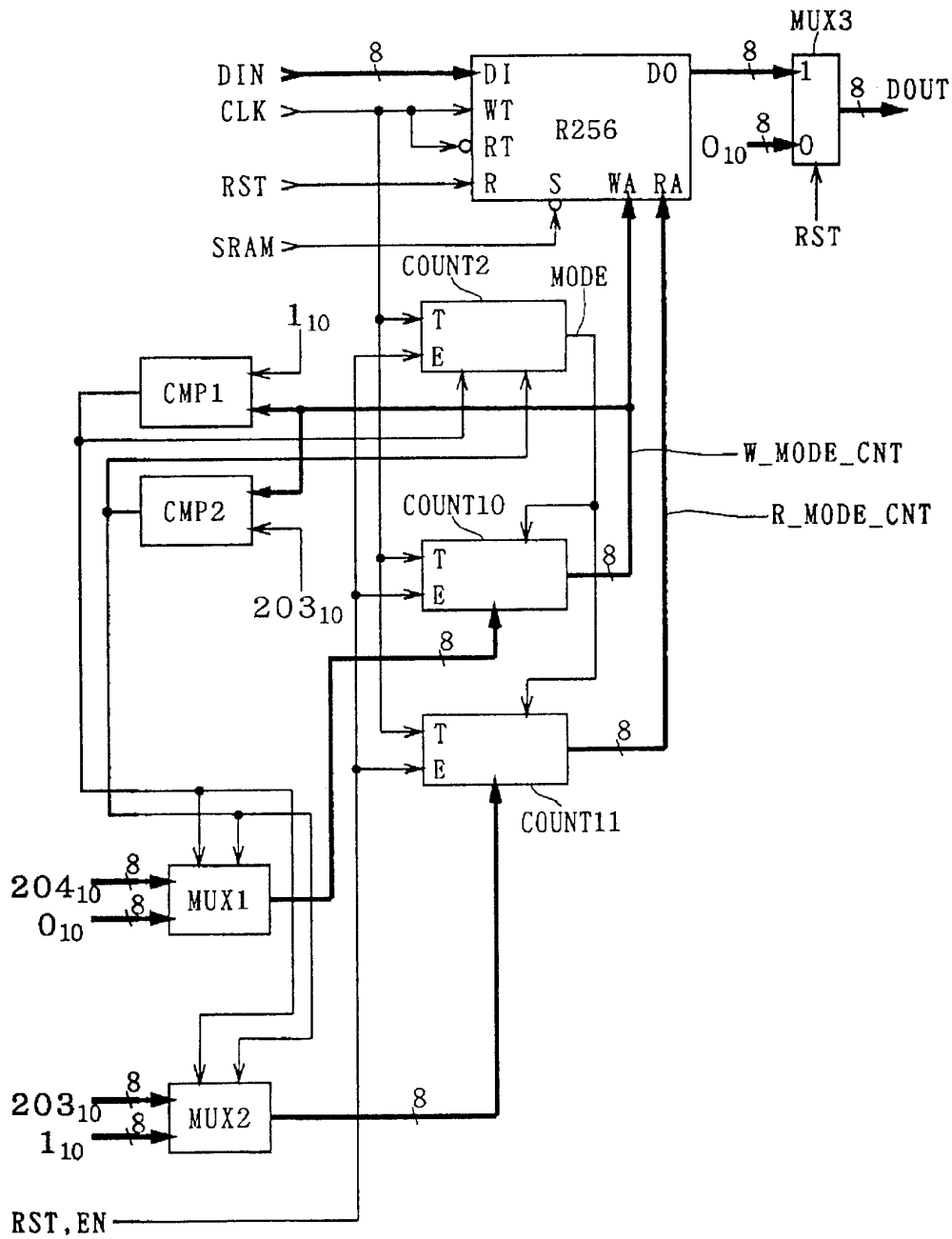
도면28



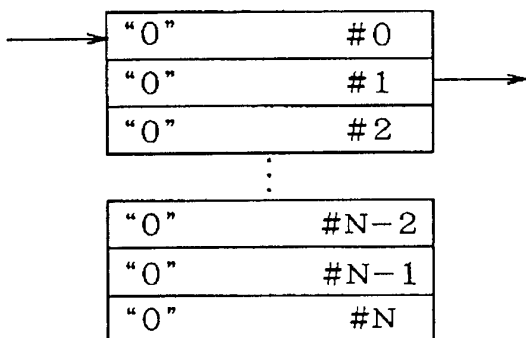
도면29



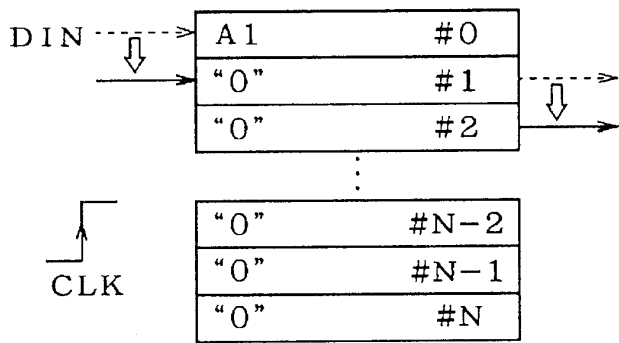
도면30



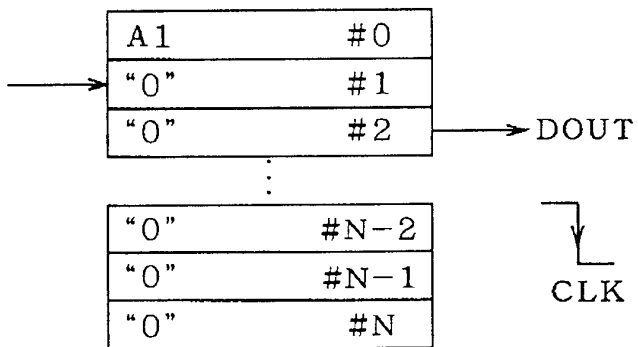
도면31



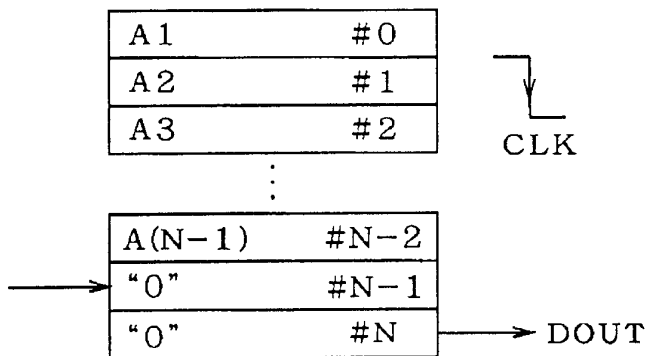
도면32



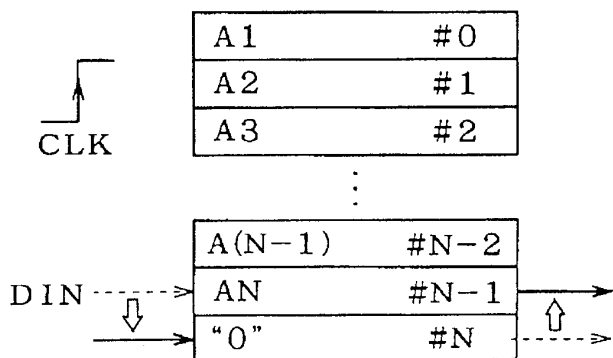
도면33



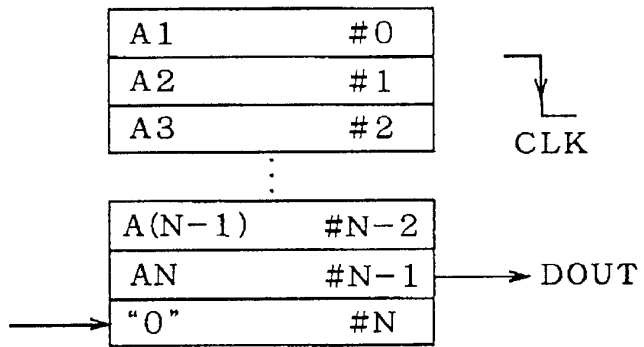
도면34



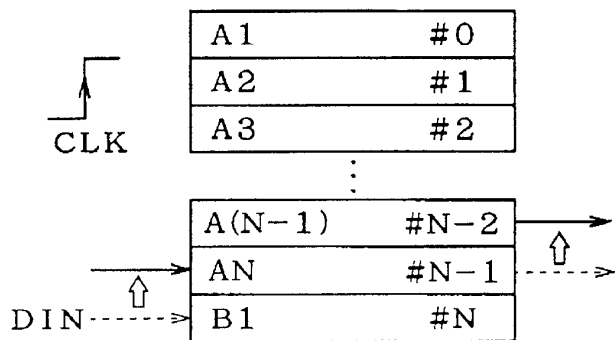
도면35



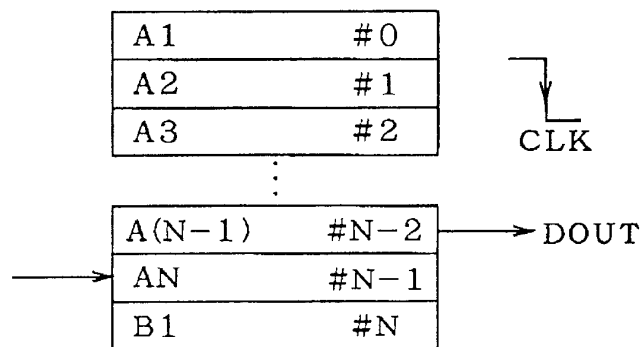
도면36



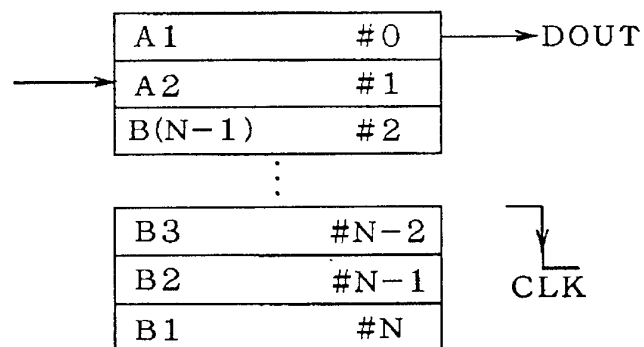
도면37



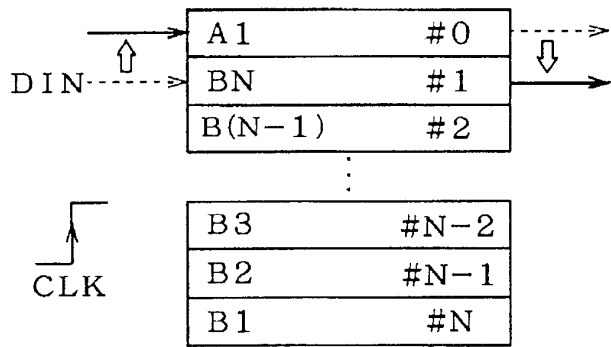
도면38



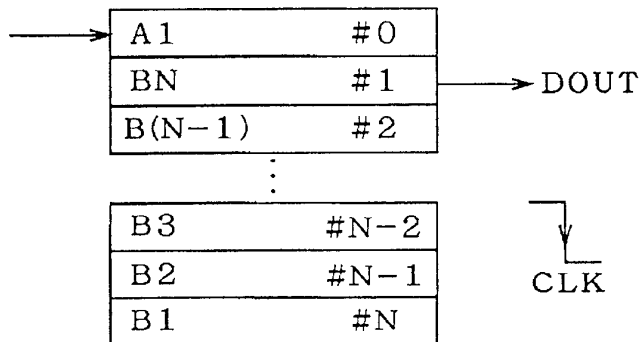
도면39



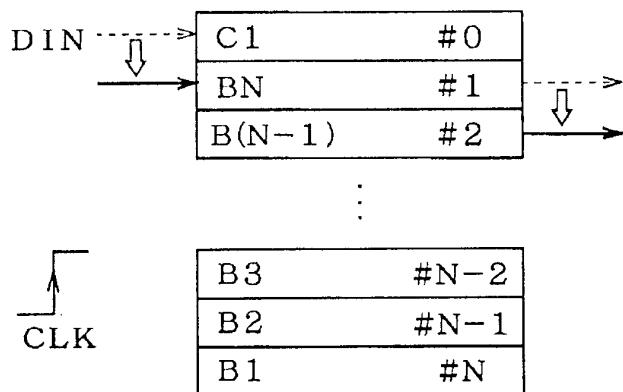
도면40



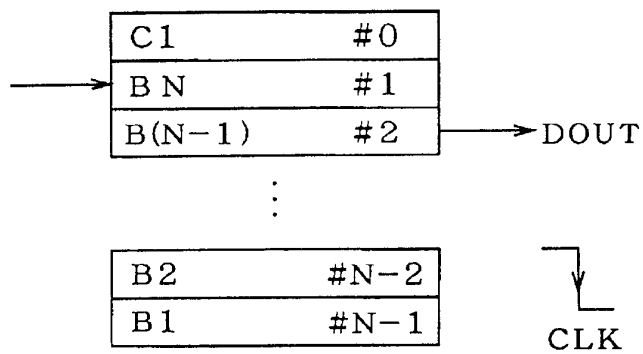
도면41



도면42

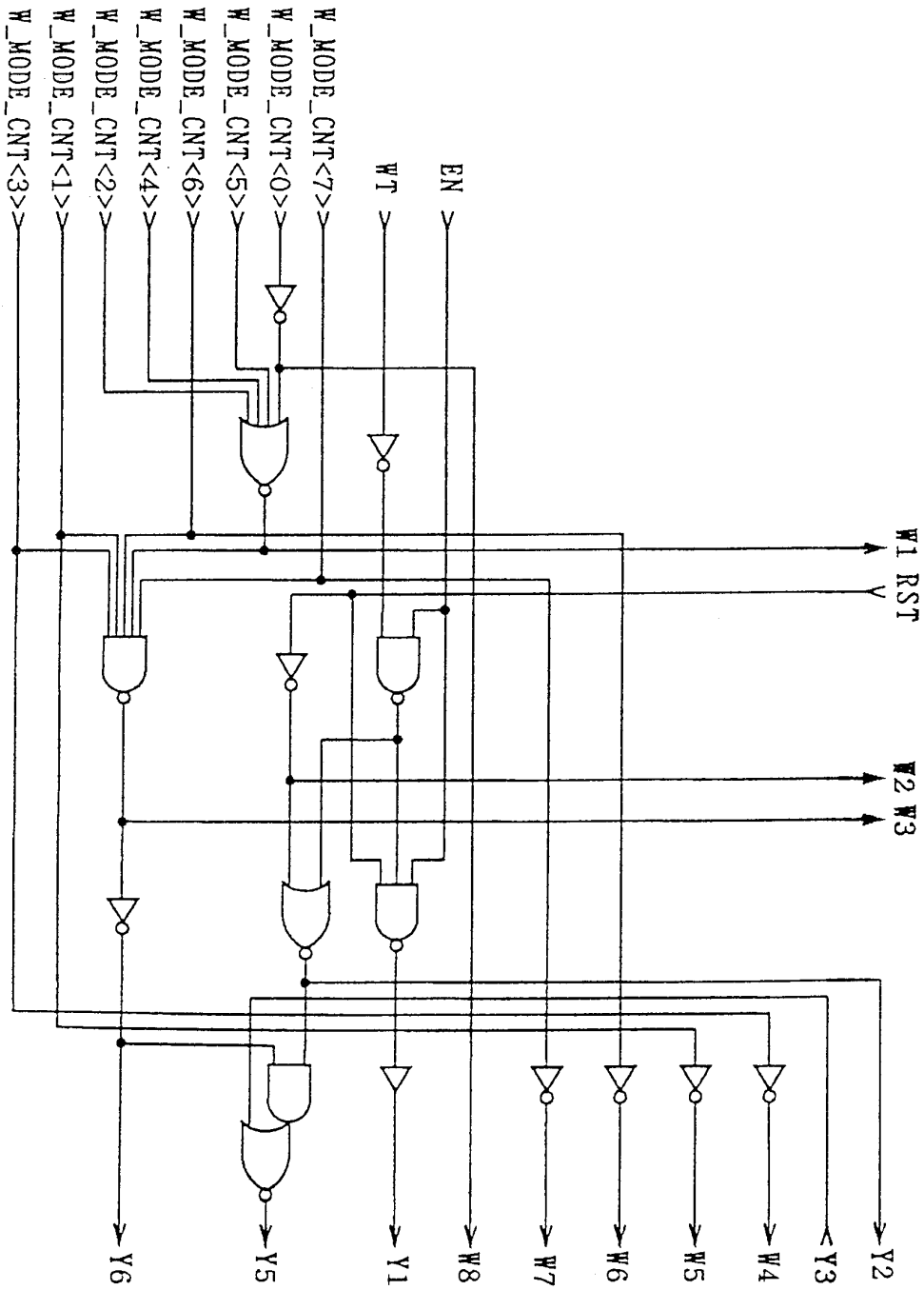


도면43

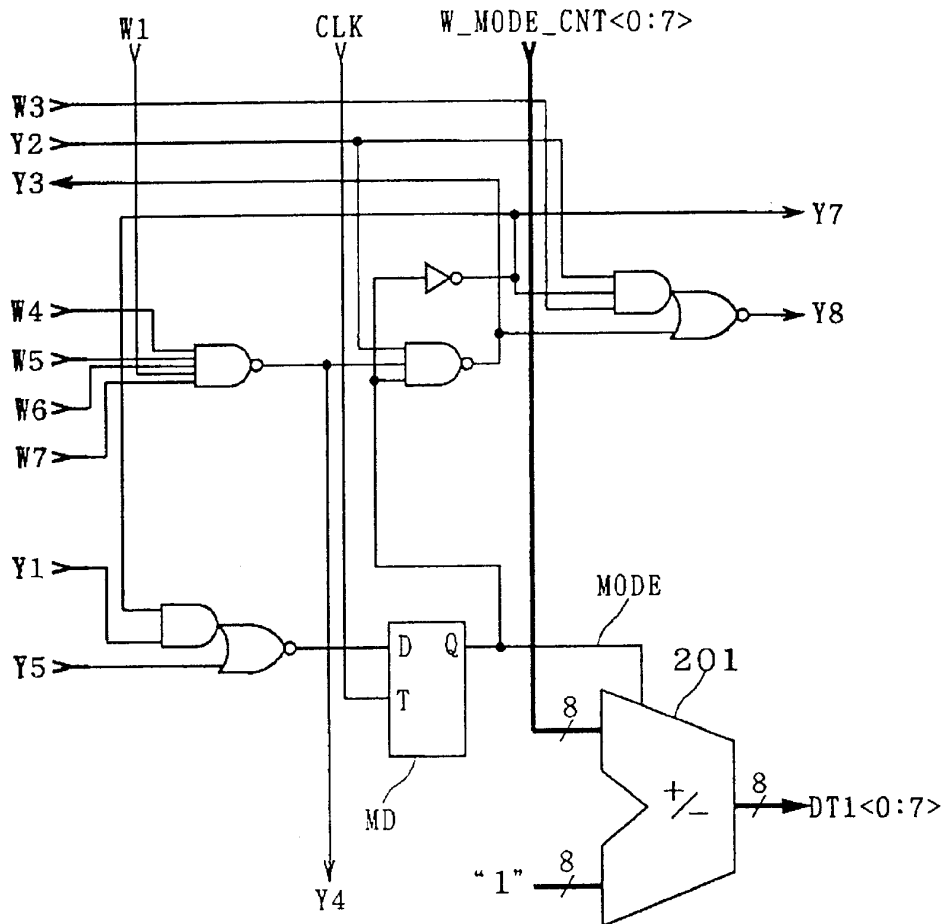




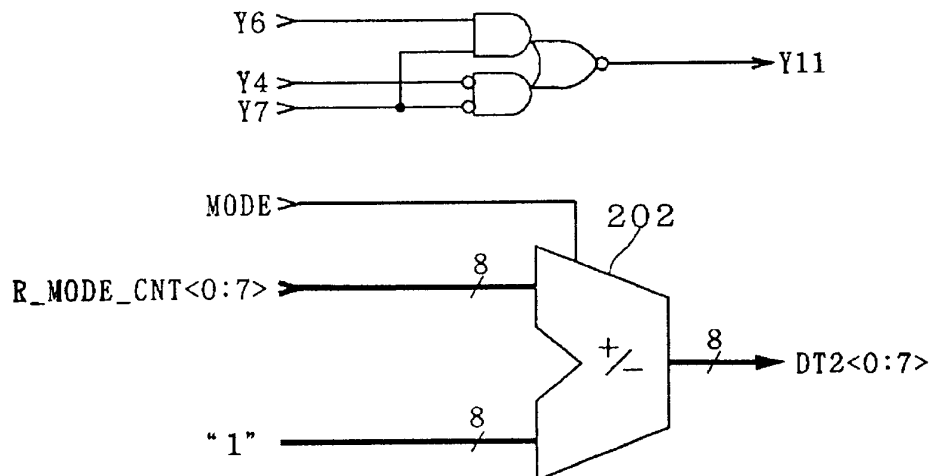
도면44



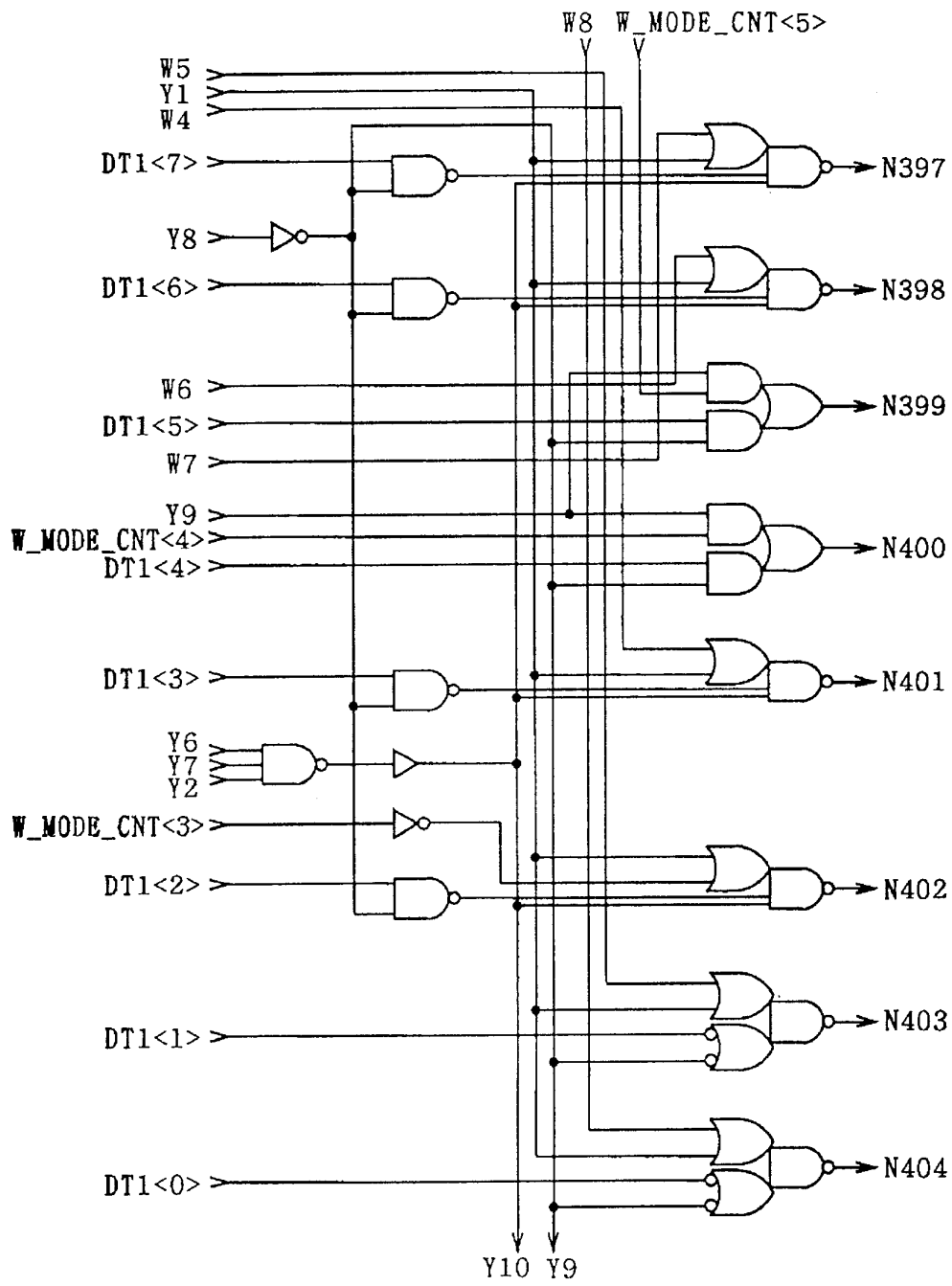
도면45



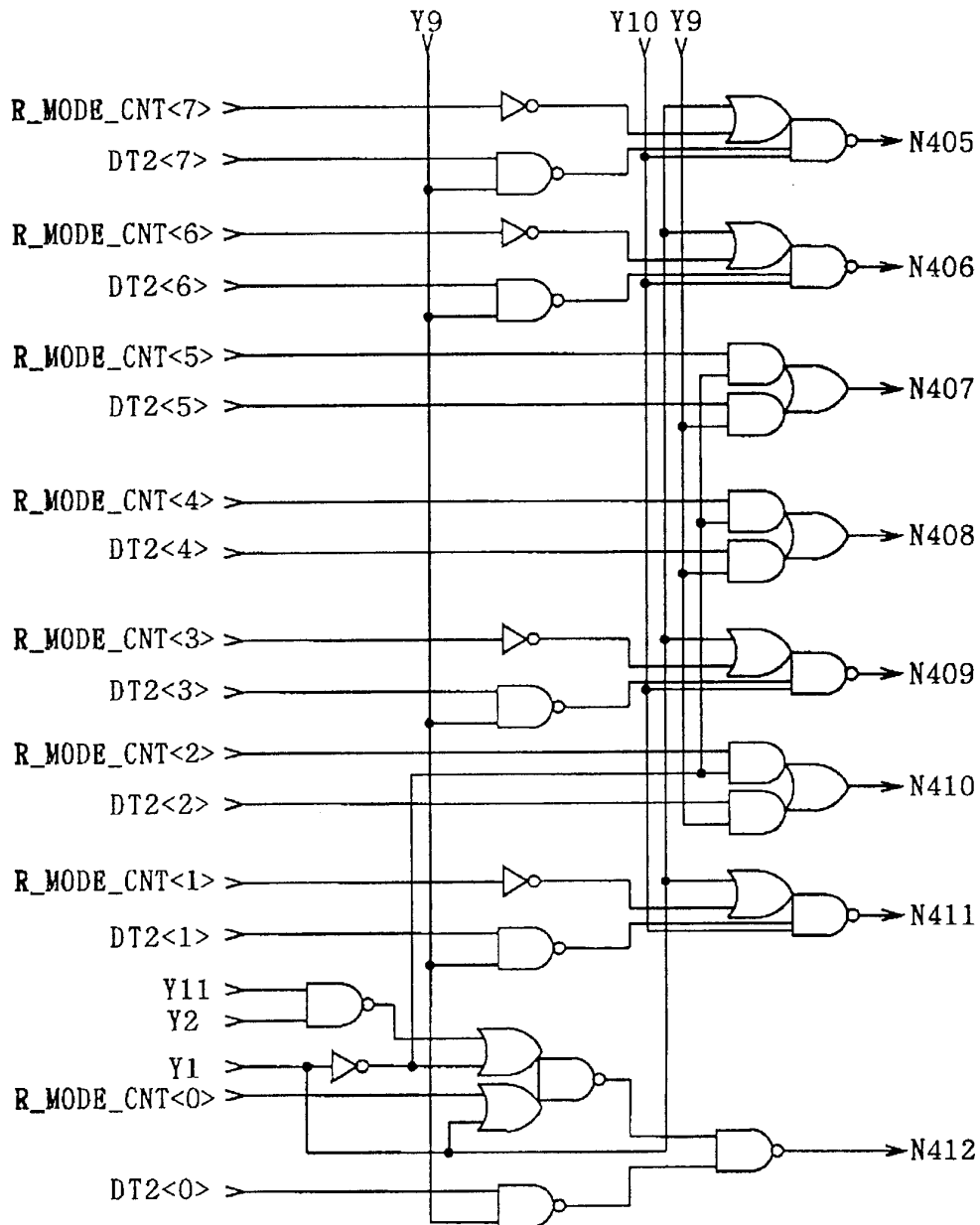
도면46



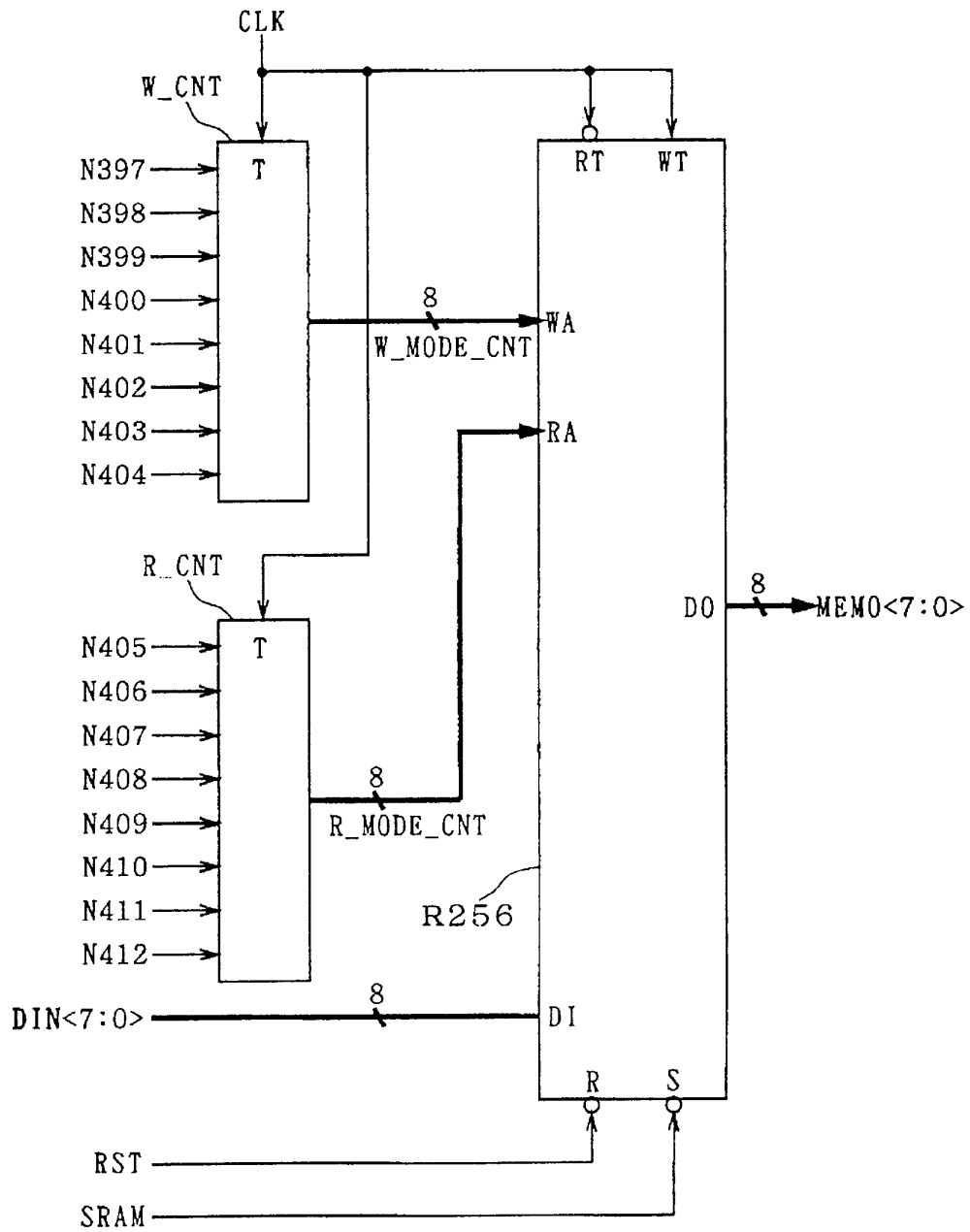
도면47



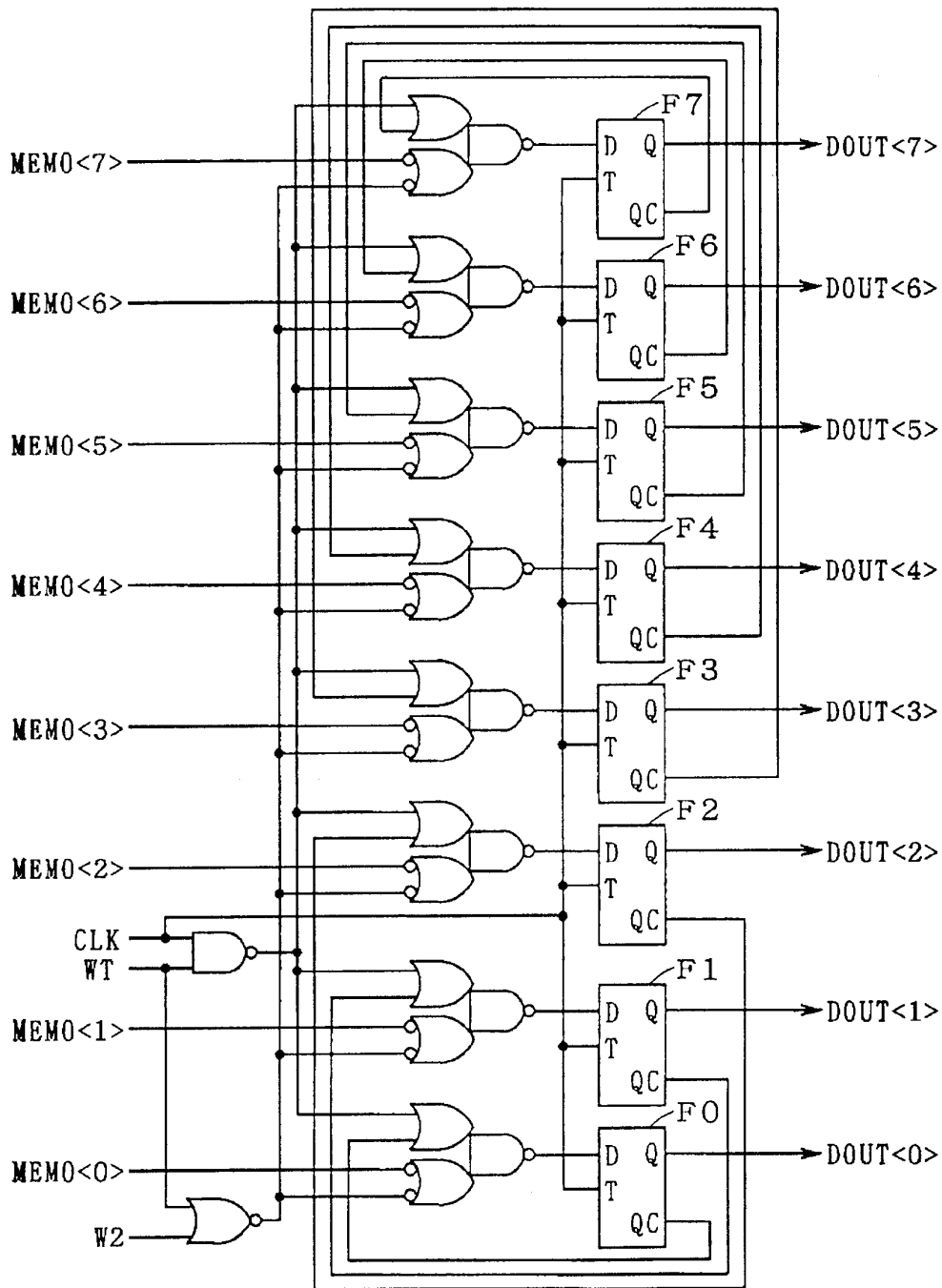
도면48



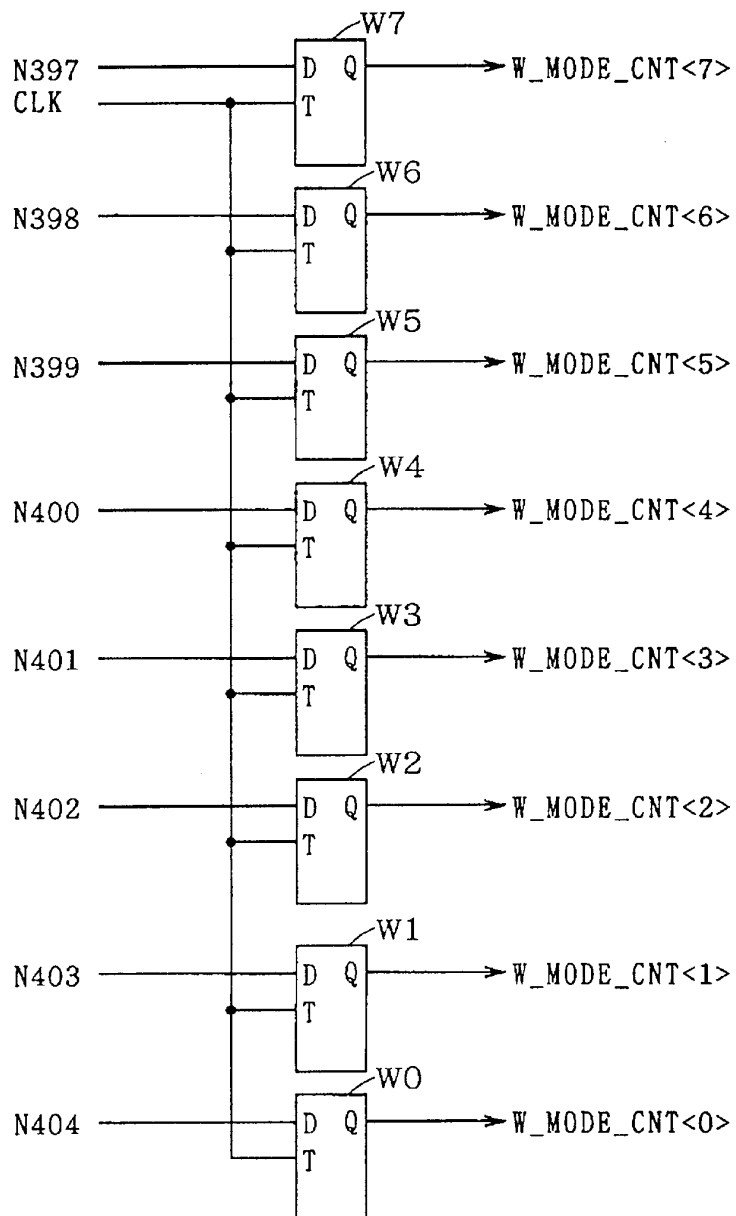
도면49



도면50

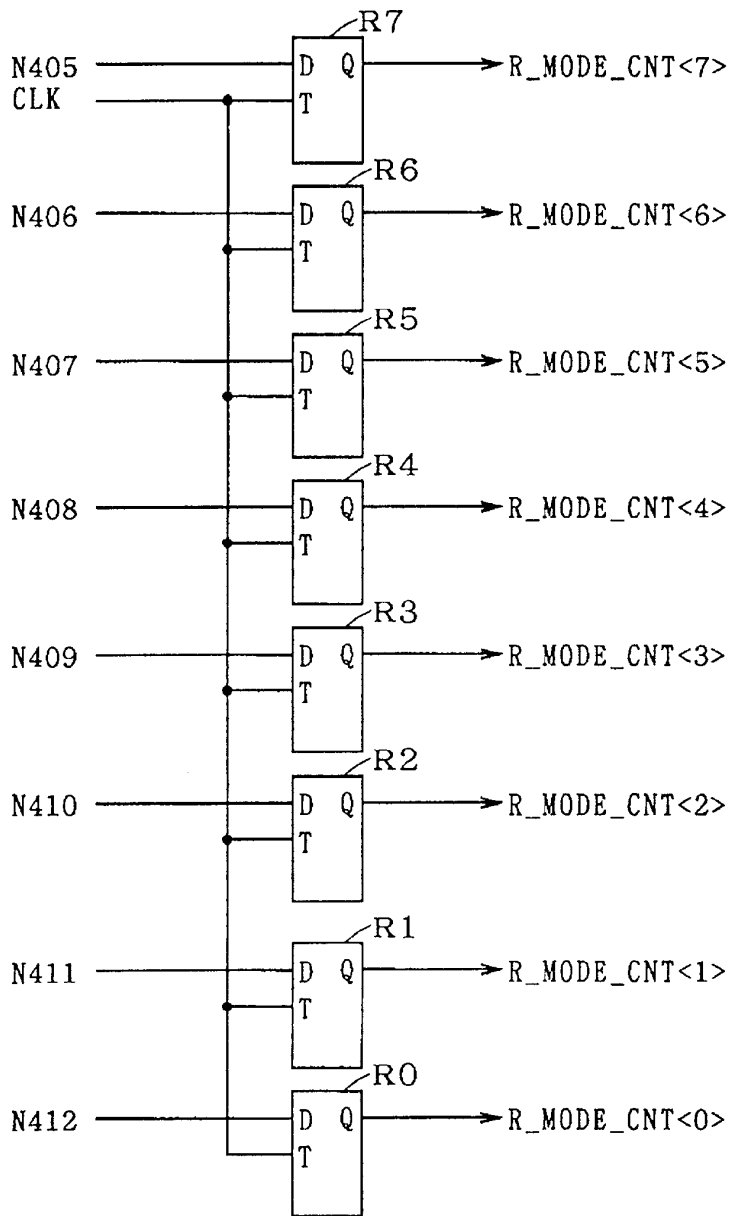


## 도면51

W\_CNT

도면52

R\_CNT



도면53

