

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) Int. Cl. ⁶ H03M 7/40	(45) 공고일자 2001년06월01일	(11) 등록번호 10-0292050
(21) 출원번호 10-1997-0058867	(24) 등록일자 2001년03월19일	(65) 공개번호 특1999-0038968
(22) 출원일자 1997년11월08일	(43) 공개일자 1999년06월05일	

(73) 특허권자	현대반도체주식회사 김영환
(72) 발명자	충북 청주시 흥덕구 향정동 1번지 류승철 서울특별시 서초구 양재2동 204-15 김용환
(74) 대리인	서울특별시 서초구 양재2동 204-15 양순석, 한윤근

심사관 : 박형식

(54) 가변장복호기의 데이터 가변장치

요약

본 발명은 출력 데이터의 비트 수를 줄여서 칩 내의 데이터 버스가 차지하는 면적을 감소시킴과 동시에, 출력 데이터의 비트 수가 감소여 발생하는 동작 속도의 문제를 해결하기 위하여 멀티플렉서와 레지스터를 이용하여 미리 데이터를 액세스하도록 하여 효율적인 데이터의 입출력 동작을 구현하기 위한 가변장 복호기의 데이터 가변 장치에 관한 것으로, 출력시키하고자 하는 데이터의 워드 길이가 입력되고, 데이터 액세스 제어신호와 다중화 제어신호, 쉬프트 제어신호를 발생시키는 제어신호 발생수단과, 상기 데이터 액세스 제어신호가 활성화될 때마다 기 저장되어 있는 데이터의 쉬프트가 이루어지고, 또 상기 쉬프트에 따라 발생한 빈 저장 영역에 새로운 데이터의 입력이 이루어지는 제1데이터 저장수단과, 상기 다중화 신호에 따라 상기 데이터 저장수단에 저장되어 있는 데이터 가운데 일부를 선택적으로 취하여 출력하는 다중화 수단과, 상기 다중화 수단에서 출력되는 데이터가 입력되고, 상기 쉬프트 제어신호의 값에 따라 쉬프트 포인터가 가변되는 쉬프트를 포함하는 데이터 가변장치를 포함하여 이루어진다.

대표도

도2

명세서

도면의 간단한 설명

- 제1도는 종래의 가변장 복호기를 나타낸 블록도.
제2도는 본 발명에 따른 데이터 가변장치의 블록도.
제3도는 본 발명에 따른 어큐뮬레이터 블록의 블록도.
제4도는 본 발명에 따른 인코더의 구성을 나타낸 회로도.

* 도면의 주요부분에 대한 부호의 설명

10 : 입력버퍼	20, 30, 80~110 : 래치
40, 150 : 배럴 쉬프트	50, 160 : 어큐뮬레이터
60 : 프로그래머블 로직 어레이	70 : 출력버퍼
120, 130 : 멀티플렉서	170 : 인코더
174 : 배타적 오어게이트	171~173 : 디 플립플롭
176, 178 : 오어 게이트	175, 177 : 앤드 게이트
C : 자리올림	S : 쉬프트 포인터
L : 래치 인에이블 신호	M : 멀티플렉서 제어신호

발명의 상세한 설명

발명의 목적

발명이 속하는 기술분야 및 그 분야의 종래기술

본 발명은 가변장 복호기의 데이터 가변장치에 관한 것으로, 특히 출력 데이터의 크기를 감소시켜서 칩의 크기를 줄임과 동시에 효율적인 데이터의 입출력이 이루어지도록 하는 가변장 복호기의 데이터 가변장치에 관한 것이다.

영상 압축 기술의 국제 표준인 JPEG(Joint Picture Experts Group) 또는 MPEG(Moving Picture Experts Group)에서는 이산여현변환(DCT : Discrete Cosine Transform) 방법을 사용하여 영상 신호를 부호화하고, 이 부호화 과정에서 얻어지는 부호의 발생 확률을 이용하여 정보 압축을 구현한다.

일반적인 영상 신호에 있어서 각 화소 블록의 영상 정보, 즉 휘도(luminance)나 색상(chrominance) 따위의 값은 화면 전체에 걸쳐서 변화의 폭이 작다. 그러나 이 영상 신호를 이산여현변환하면 계수(DCT coefficients)의 발생 확률에는 심한 편중이 일어난다. 그 이유는 이산여현변환 전에 화면에 불규칙하게 분포하던 영상 정보값이 변환 후에는 공간 주파수(spatial frequency)의 저주파항 쪽으로 집중(decorrelate)되기 때문이다. 따라서 영상 정보를 복원하는데 필요한 정보의 대부분을 포함하는 저주파항을 취하고 고주파 항은 버림으로써 정보의 압축이 이루어지는 것이며, 이것이 제이펙 또는 엠팩에서 이산여현변환 방법을 사용하는 가장 큰 이유이기도 하다.

이와 같이 편중된 변환 계수에 대하여 발생 확률이 높은 값에는 길이가 짧은 부호를 할당하고, 발생 확률이 낮은 값에는 길이가 긴 부호를 할당하도록 부호기와 복호기 사이에 새로운 부호체계를 세우면 평균 부호길이를 줄일 수 있다. 이를 구현하기 위한 수단이 가변장 부호기(Variable Length Encoder)이며, 이를 복원하는 수단이 가변장 복호기(Variable Length Decoder)이다. 따라서 가변장 부호기 또는 복호기에서 처리되는 각각의 단위 데이터 블록의 길이 또한 가변적일 수밖에 없다. 이와 같은 종래의 가변장 복호기의 블록도를 제1도에 나타내었다.

제1도에 나타낸 가변장 복호기는 입력 버퍼(10)와 출력 버퍼(70), 데이터를 디코딩하는 프로그래머블 로직 어레이(60), 입력 버퍼(10)에서 프로그래머블 로직 어레이(60)로 전송되는 데이터의 길이를 가변시키는 가변 블록으로 구성된다.

제1도에서 가변 블록은 두 개의 래치(20)(30)와 32비트 배럴 쉬프터(barrel shifter; 40), 어큐물레이터(50)으로 구성된다.

입력 버퍼(10)에는 전단의 가변장 인코더로부터 출력되는 이코딩된 데이터가 입력된다. 입력 버퍼(10)에 저장된 데이터는 래치 인에이블 신호(L)에 의해 16비트 단위로 래치(20)에 액세스된다. 또 한번의 래치 인에이블 신호(L)가 발생하면 래치(20)에 저장되어 있는 데이터가 또 다른 래치(30)로 이동하며, 래치(20)에는 데이터 입력 버퍼(10)로부터 새로운 16비트의 데이터가 액세스되어 두 개의 래치(20)(30)는 모두 32비트의 데이터로 채워진다. 따라서 두 개의 래치(20)(30)를 통하여 16비트 단위의 쉬프트가 이루어지는 것이다.

배럴 쉬프터(40)는 32비트 단위로 데이터가 입력되며, 4비트의 쉬프트 포인터, 16비트 단위의 출력을 갖는다. 입력 데이터는 위에 설명한 두 개의 래치(20)(30)에 채워진 32비트의 데이터이며, 래치(20)의 데이터가 상위 비트로 되고 또 다른 래치(30)의 데이터가 하위 비트로 된다. 쉬프트 포인터(S)는 배럴 쉬프터(40)에 입력되어 있는 데이터 가운데 유효 데이터의 시작 비트를 지시하는 것으로서, 배럴 쉬프터(40)로부터 프로그래머블 로직 어레이(60)로 제공되는 데이터의 최대 비트 수가 16비트이므로 쉬프트 포인터의 길이는 4비트이면 충분하다. 이와 같은 쉬프트 포인터(S)는 어큐물레이터(50)로부터 제공된다.

어큐물레이터(50)는 이진수 0000의 쉬프트 포인터를 초기값으로 갖고, 프로그래머블 로직 어레이(60)에서 발생하는 4비트의 워드 길이(W)가 입력된다. 따라서 기 저장되어 있는 쉬프트 포인터(S)에 새로 입력되는 워드 길이(W)가 누산되며, 누산 결과는 배럴 쉬프터(40)의 쉬프트 포인터(S)로서 출력되고, 만약 자리올림(C)이 발생하면(즉, C=1) 입력 버퍼(10)의 데이터를 래치(20)(30)로 액세스하거나 래치(20)(30) 사이의 데이터 쉬프트에 필요한 래치 인에이블 신호(L)로서 사용된다.

프로그래머블 로직 어레이(60)는 가변장 부호기에 의하여 인코딩된 데이터를 복원(즉, 디코딩)하는 부분으로서, 복원된 데이터는 출력 버퍼(70)로 전송되어 저장된다.

이와 같은 종래의 가변장 복호기의 데이터 가변장치는 배럴 쉬프터(40)를 이용하여 두 개의 래치(20)(30)에 채워진 32비트의 데이터 가운데 필요한 길이(최대 16비트)의 데이터를 취하여 프로그래머블 로직 어레이(60)로 전송한다.

이와 같은 종래의 가변장 복호기의 가변 블록은 래치(20)(30)에 데이터를 채워넣기 위하여 어큐물레이터(50)에서 논리값 1의 자리올림이 발생할 때마다 데이터의 액세스가 이루어지기 때문에 액세스하고자 하는 데이터의 길이가 24비트라고 가정할 때 최대 세번의 논리값 1의 자리올림이 발생해야만 한다.

또 배럴 쉬프터(40)에서 프로그래머블 로직 어레이로 출력되는 데이터의 길이가 최대 16비트이므로, 프로그래머블 로직 어레이(60)에서는 최대 16비트의 데이터를 처리할 수 있는 회로 구성을 만족해야하기 때문에 프로그래머블 로직 어레이(60)의 크기가 매우 커질 수밖에 없다.

발명이 이루고자 하는 기술적 과제

이와 같은 본 발명은 출력 데이터의 비트 수를 줄여서 칩 내의 데이터 버스가 차지하는 면적을 감소시키고 동시에, 출력 데이터의 비트 수가 감소여 발생하는 동작 속도의 문제를 해결하기 위하여 멀티플렉서와

레지스터를 이용하여 미리 데이터를 액세스하도록 하여 효율적인 데이터의 입출력 동작을 구현하는데 목적이 있다.

발명의 구성 및 작용

이와 같은 목적의 본 발명은 출력시키고자 하는 데이터의 워드 길이가 입력되고, 데이터 액세스 제어신호와 다중화 제어신호, 쉬프트 제어신호를 발생시키는 제어신호 발생수단과; 상기 데이터 액세스 제어신호가 활성화될 때마다 기 저장되어 있는 데이터의 쉬프트가 이루어지고, 또 상기 쉬프트에 따라 발생한 빈 저장 영역에 새로운 데이터의 입력이 이루어지는 제1데이터 저장수단과; 상기 다중화 신호에 따라 상기 데이터 저장수단에 저장되어 있는 데이터 가운데 일부를 선택적으로 취하여 출력하는 다중화 수단과; 상기 다중화 수단에서 출력되는 데이터가 입력되고, 상기 쉬프트 제어신호의 값에 따라 쉬프트 포인터가 가변되는 쉬프트를 포함하는 데이터 가변장치를 포함하여 이루어진다.

이와 같이 이루어진 본 발명의 바람직한 실시예를 제2도 내지 제4도를 참조하여 상세히 설명하면 다음과 같다. 제2도는 본 발명에 따른 데이터 가변장치의 블록도이다. 제2도에서 어큐뮬레이터 블록(140)은 제어신호 발생수단으로서, 입력으로는 3비트의 워드 길이(W)가 있고, 출력으로는 1비트의 래치 인에이블 신호(L)와 2비트의 멀티플렉서 제어신호(M), 3비트의 쉬프트 포인터(S)가 있다. 입력인 워드 길이(W)는, 본 발명의 데이터 가변장치로부터 가변 길이의 데이터를 입력받아 이를 처리하는 프로그래머블 로직 어레이에서 요구하는 유효 데이터의 길이를 의미한다.

래치 인에이블 신호(L)는 입력단을 형성하는 네 개의 래치(80~110)를 동시에 인에이블 시켜서 각각의 래치에서 데이터의 입력과 출력이 이루어지도록 하는 신호이며, 이때 각각의 래치(80~110)를 통하여 입출력되는 데이터의 단위는 8비트이다. 먼저 두 개의 래치(100)(110)에 16비트 입력 데이터의 상위 8비트와 하위 8비트가 각각 입력된다. 또한 이 두 개의 래치(100)(110)에서 출력되는 데이터는 또 다른 두 개의 래치(80)(90)에 입력되도록 연결되어 있다. 따라서 래치 인에이블 신호(L)가 활성화될 때마다, 두 개의 래치(100)(110)에 저장되어 있던 데이터가 또 다른 두 개의 래치(80)(90)로 이동하게 되고, 비워진 두 개의 래치(100)(110)는 각각 새로운 16비트 입력 데이터의 상위 8비트와 하위 8비트로 채워진다. 결과적으로 네 개의 래치(80~110)에서는 래치 인에이블 신호(L)가 활성화될 때마다 16비트 단위의 상향 쉬프트가 이루어지는 것이다.

멀티플렉서 제어신호(M)는 두 개의 멀티플렉서(120)(130)에 입력되는 선택신호이다. 멀티플렉서(120)는 위에 설명한 네 개의 래치(80~110)에 저장되어 있는 총 32비트의 데이터 가운데 8비트의 데이터만을 선택하여 출력한다. 또 다른 멀티플렉서(130)는 아직 래치에 입력되지 않고 입력 버퍼 등에서 대기중인 16비트 데이터의 상위 8비트와 세 개의 래치(90~110)에서 출력되는 24비트 데이터의 총 32비트 데이터 가운데 8비트의 데이터만을 선택하여 출력한다. 이 두 개의 멀티플렉서(120)(130)는 동일한 값의 멀티플렉서 제어신호(M)에 의해 출력 데이터가 결정되지만, 실제로 멀티플렉서(120)가 선택할 수 있는 데이터의 범위는 다른 멀티플렉서(130)가 선택할 수 있는 데이터의 범위보다 8비트만큼 상위 데이터이기 때문에, 실제로 두 개의 멀티플렉서(120)(130)에서 동시에 출력되는 데이터는 각각 상위 8비트와 하위 8비트의 연속된 16비트 데이터이다.

쉬프트 포인터(S)는 16비트 배럴 쉬프트(150)에 입력되는 쉬프트 변수이다. 배럴 쉬프트(150)의 출력단에는 프로그래머블 로직 어레이가 연결되어 있어, 배럴 쉬프트(150)로부터 8비트 단위로 데이터를 제공받아 일련의 데이터 처리과정을 수행한다. 이 프로그래머블 로직 어레이에서 처리되는 데이터는 MPEG의 특성을 설명할 때 언급한 바와 같이 그 길이가 매우 가변적이기 때문에, 배럴 쉬프트(150)로부터 8비트 단위로 제공되는 데이터가 모두 처리되지 않고 일부만 처리되기도 한다. 따라서 아직 처리되지 않는 데이터의 비트 수에 대한 정보를 쉬프트 포인터(S)의 형태로 배럴 쉬프트(150)에 전달하여 현재 배럴 쉬프트(150)에 입력되어 있는 데이터 가운데 유효한 부분의 시작 비트를 확인할 수 있도록 하는 것이다.

제3도는 어큐뮬레이터 블록(140)의 구성을 나타낸 블록도이다. 제3도에 나타낸 바와 같이 본 발명에 따른 어큐뮬레이터 블록(140)은 어큐뮬레이터(160)와 인코더(170)로 구성된다. 어큐뮬레이터(160)에서는 새로 입력된 워드 길이(W)와 기 저장되어 있는 값의 누산이 이루어진다. 자리올림(C)을 제외한 누산 결과는 쉬프트 포인터(S)로서 출력되며, 자리올림(C)은 인코더(170)로 출력된다. 인코더(170)에서는 자리올림(C)을 이용하여 래치 인에이블 신호(L)와 멀티플렉서 제어신호(M)를 발생시킨다.

이와 같은 인코더(170)의 상세한 구성을 제4도에 나타내었다. 인코더(170)에서는 동일한 클럭 신호(CLK)에 동기되어 데이터의 입출력이 이루어지는 세 개의 디 플립플롭(171~173)을 통하여 멀티플렉서 제어신호(M)의 하위 비트(M0) 및 상위 비트(M1), 래치 인에이블 신호(L)를 각각 발생시킨다.

먼저, 멀티플렉서 제어신호(M)의 하위 비트(M0)를 발생시키는 디 플립플롭(171)에는 배타적 오어 게이트(174)의 출력이 데이터(D)로서 입력된다. 이 배타적 오어 게이트(174)에는 어큐뮬레이터(160)에서 출력되는 자리올림(C)이 입력되며, 디 플립플롭(171)에서 출력되는 하위 비트(M0)는 배타적 오어 게이트(174)의 또 다른 입력으로 피드백된다. 즉, (n+1)번째 클럭에서 디 플립플롭(171)으로부터 출력되는 하위 비트(M0_{n+1})의 논리값은, (n)번째 클럭에서 출력된 하위 비트(M0_n)와 (n)번째 자리올림(C_n)의 배타적 논리합 연산을 통해 얻는다.

멀티플렉서 제어신호(M)의 상위 비트(M1)를 발생시키는 디 플립플롭(172)에는 앤드 게이트(175)의 출력이 데이터(D)로서 입력된다. 이 앤드 게이트(175)에는 오어 게이트(176)의 출력이 또 다른 입력으로 되는데, 이 오어 게이트(176)에는 멀티플렉서 제어신호(M)의 하위 비트(M0)와 함께, 디 플립플롭(172)에서 출력되는 멀티플렉서 제어신호(M)의 상위 비트(M1)가 피드백되어 입력된다. 즉, (n+1)번째 클럭에서 디 플립플롭(172)으로부터 출력되는 상위 비트(M1_{n+1})의 논리값은, (n)번째 하위 비트(M0_n)와 (n)번째 상위 비트(M1_n)의 논리합 연산 결과를 (n)번째 자리올림(C_n)과 논리곱 연산함으로써 얻는다. 이를 다음의 표 2에 나타내었다.

래치 인에이블 신호(L)를 발생시키는 디 플립플롭(173)에는 앤드 게이트(177)의 출력이 데이터(D)로서 입

력된다. 이 앤드 게이트(177)에는 오어 게이트(178)의 출력이 또 다른 입력으로 되는데, 이 오어 게이트(178)에는 멀티플렉서 제어신호의 하위 비트(M0)와 함께, 반전된 자리올림(/C)이 입력된다. 즉, 디 플립 플롭(173)에서 출력되는 래치 인에이블 신호(L)는 이전에 출력된 하위 비트와 새로 입력되는 반전된 자리 올림의 논리합 연산결과를 이전에 출력된 상위 비트와 논리곱 연산한 결과이다. 따라서, (n+1)번째 클럭에서 출력되는 래치 인에이블 신호(L_{n+1})의 논리값은, (n)번째 하위 비트(M0_n)와 반전된 (n)번째 자리올림 (/C_n)의 논리합 연산 결과를 (n)번째 상위 비트(M1_n)와 논리곱 연산함으로써 얻는다.

이상의 설명에서 알 수 있듯이, 멀티플렉서 제어신호(M)의 값이 00인 상태에서 자리올림(C)이 발생하지 않으면(즉, C=0) 멀티플렉서 제어신호(M)의 값은 그대로 00을 유지하지만, 만약 자리올림(C)이 발생하면(즉, C=1) 멀티플렉서 제어신호(M)는 01로 된다. 이때의 래치 인에이블 신호(L)는 위의 두 경우 모두 0을 유지한다.

멀티플렉서 제어신호(M)의 값이 01인 상태에서 자리올림(C)이 발생하지 않으면 멀티플렉서 제어신호(M)는 계속 01을 유지하고, 만약 자리올림(C)이 발생하면 멀티플렉서 제어신호(M)는 10으로 된다. 이때에도 역시 래치 인에이블 신호(L)는 두 경우 모두 0을 유지한다.

멀티플렉서 제어신호(M)의 값이 10인 상태에서 자리올림(C)이 발생하지 않으면 멀티플렉서 제어신호(M)는 계속 00의 값을 갖게되며, 만약 자리올림(C)이 발생하면 멀티플렉서 제어신호(M)는 11로 된다. 이때 래치 인에이블 신호(L)는 자리올림(C)이 발생하지 않는 경우에만 1의 값을 갖게되고, 자리올림(C)이 발생하는 경우에는 0을 유지한다.

멀티플렉서 제어신호(M)의 값이 11인 상태에서 자리올림(C)이 발생하지 않으면 멀티플렉서 제어신호(M)는 01로 되고, 만약 자리올림(C)이 발생하면 멀티플렉서 제어신호(M)는 10으로 된다. 이때 래치 인에이블 신호는 멀티플렉서 제어신호(M)의 값이 11이기 때문에 자리올림(C)의 발생에 관계없이 두 경우 모두 1의 값을 갖는다.

이와 같이 구성된 본 발명에 따른 데이터 가변장치의 동작을 임의의 워드길이(W)가 입력되었을때를 예로 들어 설명하면 다음과 같다.

먼저, 회로동작의 초기상태에서는 어큐뮬레이터 블록(140)은 이진수 000을 초기값으로 가지며, 자리올림 역시 0의 초기값을 갖는다. 따라서, 래치 인에이블 신호(L) 역시 0의 값을 갖고, 멀티플렉서 제어신호(M)의 하위 비트(M0)와 상위 비트(M1) 역시 각각 0의 초기값을 가지며, 쉬프트 포인터(S) 또한 000의 초기값을 갖는다.

이때 두 개의 래치(100)(110)에는 각각 상위 8비트와 하위 8비트의 총 16비트 데이터가 입력되어 있으나, 나머지 두 개의 래치(80)(90)에는 아무런 유효 데이터도 입력되어 있지 않다. 멀티플렉서 제어신호(M) 역시 00이기 때문에 두 개의 래치(80)(90)에 저장되어 있는 데이터 이외에는 선택할 수 없는 상태여서 실제로 멀티플렉서(120)(130)을 통하여 출력할 수 있는 유효 데이터는 없다. 이때의 쉬프트 포인터(S)는 000의 초기값을 갖는다.

이 상태에서 네 개의 래치(80~110)를 모두 데이터로 채우기 위하여 의도적으로 래치 인에이블 신호(L)를 활성화시키면, 두 개의 래치(100)(110)에 저장되어 있는 상위 8비트와 하위 8비트의 데이터는 각각 또 다른 두 개의 래치(80)(90)로 이동하게 되고, 래치(100)(110)에는 새로운 상위 8비트와 하위 8비트의 16비트 데이터가 채워진다. 이때 멀티플렉서 제어신호(M)는 00이기 때문에, 멀티플렉서(120)(130)는 각각 래치(80)(90)의 데이터를 액세스하여 배럴 쉬프트(150)로 출력한 상태이다.

이 상태에서, 이진수 011의 워드 길이(W)가 어큐뮬레이터 블록(140)에 입력되면, 어큐뮬레이터(160)에서는 초기값 000에 새로 입력된 워드 길이(W) 011이 누산되어 011의 새로운 쉬프트 포인터(S)를 발생시킨다. 이 누산 과정에서 자리올림은 발생하지 않았기 때문에 자리올림(C)은 여전히 0의 논리값을 갖는다.

인코더에 입력된 논리값 0의 자리올림(C)은 제4도의 배타적 오어 게이트(174)의 출력을 0으로 만들어 디 플립플롭(171)에는 0의 데이터가 보관된다(이 값은 이후 클럭 CLK가 활성화되면 하위 비트 M0으로서 출력 된다). 또 논리값 0의 자리올림(C)은 앤드 게이트(175)의 출력을 0으로 만들어 디 플립플롭(172)에도 0의 데이터가 보관된다. 상위 비트(M1)의 논리값 0은 앤드 게이트(177)의 출력을 0으로 만들어 디 플립플롭(173)에도 역시 0의 데이터가 보관된다. 이 상태에서 클럭(CLK)이 활성화되면 각각의 디 플립플롭(171~173)에 보관되어 있는 데이터가 출력되어, 멀티플렉서 제어신호(M)의 값이 00이 되고, 래치 인에이블 신호(L) 역시 0이 된다.

위의 동작에서 어큐뮬레이터 블록(140)에 입력된 워드 길이(W)의 값이 011인 것은 현재 배럴 쉬프트(150)를 통하여 프로그래머블 로직 어레이에 제공되어 있는 8비트의 데이터 가운데 상위 3비트만을 사용하였다는 것을 의미하기 때문에, 3비트를 제외한 나머지 5비트가 유효한 데이터임을 의미한다. 따라서 프로그래머블 로직 어레이에서는 새로운 데이터의 액세스가 필요하지 않기 때문에 멀티플렉서 제어신호(M)뿐만 아니라 래치 인에이블 신호(L)의 값도 모두 초기 상태인 00과 0의 값을 유지하도록 하여, 새로운 데이터의 액세스를 실시하지 않는 것이다.

다음으로, 이진수 111의 새로운 워드 길이(W)가 입력되면, 어큐뮬레이터(160)에서는 현재값 011에 새로운 워드길이(W) 111이 누산되어, 1의 자리올림(C)과 010의 새로운 쉬프트 포인터(S)가 발생한다.

멀티플렉서 제어신호(M)의 하위 비트(M0)와 상위 비트(M1)가 모두 0인 상태에서 논리값 1의 자리올림(C)이 발생하였으므로, 제4도의 인코더의 배타적 오어 게이트(174)의 출력은 10이되어 디 플립플롭(171)에 1의 데이터가 보관된다. 그러나 이전 상태의 멀티플렉서 제어신호(M)가 00이었기 때문에 자리올림(C)의 발생과 무관하게 앤드 게이트(175)(177)의 출력이 모두 0이 되어 디 플립플롭(172)(173)에는 각각 0의 데이터가 보관된다. 이 상태에서 클럭(CLK)이 활성화되면 각각의 디 플립플롭(171~173)에 보관되어 있던 데이터가 출력되어, 멀티플렉서 제어신호(M)의 값이 01이 되고, 래치 인에이블 신호(L)는 0이 된다.

위의 동작에서 어큐뮬레이터 블록(140)에 입력된 워드 길이(W)의 값이 111인 것은 곧 현재 배럴 쉬프트

(150)를 통하여 프로그래머블 로직 어레이에 입력되어 있는 8비트의 데이터 가운데 현재 쉬프트 포인터가 지시하고 있는 상위 3비트 다음의 7비트를 사용하였다는 것을 의미한다. 그러나 현재 배럴 쉬프트(150)를 통하여 제공되어 있는 8비트의 데이터에는 5비트의 유효 데이터만이 포함되어 있기 때문에, 2비트의 데이터를 추가로 제공받기 위하여 다음 8비트의 데이터를 액세스하도록 해야한다.

즉, 배럴 쉬프트(150)에 입력되어 있는 16비트의 데이터 가운데 상위 8비트는 더 이상 유효하지 않은 데이터이므로, 멀티플렉서 제어신호(M)를 01로 전환시켜서 두 개의 래치(90)(100)에 저장되어 있는 데이터를 선택하도록 한다. 즉, 멀티플렉서(120)(130)를 통하여 8비트 단위의 상향 쉬프트를 구현함으로써 새로운 8비트의 데이터를 추가로 액세스한 것이다.

이 상태에서 또 한번의 이진수 111의 워드 길이(W)가 입력되면, 어큐뮬레이터 블록(140)에서는 현재값 010에 새로 입력된 111의 워드길이(W)가 누산되어 논리값 1의 자리올림(C)과 함께 001의 새로운 쉬프트 포인터(S)가 발생한다.

멀티플렉서 제어신호(M)의 상위 비트(M1)와 하위 비트(M0)가 각각 0과 1인 상태에서 논리값 1의 자리올림(C)이 발생하였으므로, 제4도의 인코더의 배타적 오어 게이트(174)의 출력은 0이되어 디 플립플롭(171)에 0의 데이터가 보관되며, 앤드 게이트(175)의 출력은 1이되어 디 플립플롭(172)에는 1의 데이터가 보관된다. 그러나 현재의 멀티플렉서 제어신호(M)의 상위 비트(M1)가 0이기 때문에 앤드 게이트(177)의 출력은 0이되어 디 플립플롭(173)에 0의 데이터가 보관된다. 즉, 멀티플렉서 제어신호(M)는 10이고, 래치 인에이블 신호(L)는 0이다.

이 상태에서 클럭(CLK)이 활성화되면 각각의 디 플립플롭(171~173)에 보관되어 있던 데이터가 출력되어, 멀티플렉서 제어신호(M)의 값이 10이 되고, 래치 인에이블 신호(L)는 0이 된다. 이 경우에도 멀티플렉서 제어신호(M)가 01에서 10으로 전환되어 멀티플렉서(120)(130)를 통하여 8비트의 상향 쉬프트가 이루어졌다. 즉, 현재 배럴 쉬프트(150)를 통하여 제공되어 있는 8비트의 데이터에는 6비트의 유효 데이터만이 포함되어 있기 때문에, 1비트의 데이터를 추가로 제공받기 위하여 다음 8비트의 데이터를 액세스 해야한다. 따라서, 배럴 쉬프트(150)에 입력되어 있는 16비트의 데이터 가운데 상위 8비트는 더 이상 유효하지 않은 데이터이므로, 멀티플렉서 제어신호(M)를 01로 전환시켜서 두 개의 래치(90)(100)에 저장되어 있는 데이터를 선택하도록 한다. 즉, 멀티플렉서(120)(130)를 통하여 또 한번의 8비트 단위 상향 쉬프트를 구현함으로써 새로운 8비트 데이터를 추가로 액세스한 것이다.

이 상태에서 이진수 110의 새로운 워드 길이(W)가 입력되면, 기존의 001의 쉬프트 포인터(S)에 새로 입력된 110의 워드길이(W)가 누산되어 111의 새로운 쉬프트 포인터(S)가 발생하며, 이때 자리올림(C)은 발생하지 않는다.

멀티플렉서 제어신호(M)의 상위 비트(M1)와 하위 비트(M0)가 각각 1과 0인 상태에서 자리올림(C)이 발생하지 않았으므로, 제4도의 인코더의 배타적 오어 게이트(174)의 출력은 0이되어 디 플립플롭(171)에는 0의 데이터가 보관되며, 논리값 0의 자리올림(C)으로 인하여 앤드 게이트(175)의 출력 역시 0이되어 디 플립플롭(172)에도 0의 데이터가 보관된다. 그러나 현재의 멀티플렉서 제어신호(M)의 상위 비트(M1)가 1이고, 반전된 자리올림(/C) 역시 1이기 때문에 앤드 게이트(177)의 출력은 1이되어 디 플립플롭(173)에는 1의 데이터가 보관된다. 즉, 멀티플렉서 제어신호(M)는 00이 되고, 래치 인에이블 신호(L)는 1이된다.

래치 인에이블 신호(L)가 활성화되었으므로, 두 개의 래치(100)(110)에 저장되어 있는 16비트의 데이터는 각각 래치(80)(90)로 이동하고, 비워진 래치(100)(110)에는 각각 상위 8비트와 하위 8비트의 새로운 16비트 데이터가 액세스된다.

멀티플렉서 제어신호(M)가 00으로 복귀함으로써 두 개의 멀티플렉서(120)(130)는 각각 래치(80)(90)의 데이터를 액세스할 수 있다. 그러나 실제로 래치(80)(90)에 저장되어 있는 16비트 데이터는 멀티플렉서 제어신호(M)가 00으로 복귀하기 이전에 래치(100)(110)에 저장되어 있던 데이터이므로, 실제로 멀티플렉서(120)(130)에서 액세스할 수 있는 데이터는 멀티플렉서 제어신호(M)가 00으로 복귀하기 이전과 동일한 데이터이다.

즉, 쉬프트 포인터가 701에서 실제로 배럴 쉬프트(150)에 입력되어 있는 상위 8비트와 하위 8비트의 데이터 가운데, 상위 8비트 데이터의 1비트가 아직 유효한 비트이므로, 멀티플렉서(120)(130)를 통한 8비트 단위 쉬프트(즉, 8비트의 새로운 데이터 액세스)는 필요치 않다. 그러나 래치(80~110)에 보관중인 32비트 데이터 가운데 상위 16비트의 데이터가 이미 멀티플렉서(120)(130)를 통하여 액세스 되었기 때문에, 래치(80~110)에 새로운 16비트 데이터의 액세스가 요구되므로, 멀티플렉서 제어신호(M)가 10에서 00으로 복귀할 때 래치 인에이블 신호(L)가 활성화되도록 하여 래치(80~110)를 통하여 새로운 16비트의 데이터 액세스를 구현한 것이다.

발명의 효과

이와 같은 본 발명은 출력 데이터의 비트 수를 줄여서 칩 내의 데이터 버스가 차지하는 면적을 감소시킴과 동시에, 출력 데이터의 비트 수가 감소여 발생하는 동작 속도의 문제를 해결하기 위하여 멀티플렉서와 레지스터를 이용하여 미리 데이터를 액세스하도록 하는 것으로, 이를 청구항 1의 발명을 통하여 구현하였다. 청구항 2 내지 청구항 6의 발명은 워드 길이의 누산을 통하여 발생한 다수의 제어 신호를 통하여 래치와 멀티플렉서를 제어함으로써 데이터의 효율적인 액세스와 쉬프트를 구현하도록 한 것이다.

(57) 청구의 범위

청구항 1

입력 데이터를 가변 길이로 분할 출력하기 위한 데이터 가변장치에 있어서, 기 저장되어 있는 쉬프트 제어신호의 값에 새로 입력된 상기 워드 길이를 누산하여 새로운 쉬프트 제어신호 값을 발생시키는 누산기와; 상기 누산기의 누산 결과에 따라 발생한 새로운 쉬프트 제어신호 값에 자리올림이 발생하는 경우, 상

기 자리올림을 입력받아 상기 데이터 액세스 제어신호와 상기 다중화 제어신호를 발생시키는 인코더로 구성되어, 출력시키고자 하는 데이터의 워드 길이가 입력되고, 데이터 액세스 제어신호와 다중화 제어신호, 쉬프트 제어신호를 발생시키는 제어신호 발생수단과; 입력 데이터 가운데 상기 워드 길이로 나타낼 수 있는 최대 길이 만큼의 데이터가 각각 저장되는 제 1 내지 제 4의 네 개의 래치로 이루어져서, 상기 제 1 래치에는 상기 입력 데이터 가운데 상기 워드 길이로 나타낼 수 있는 최대 길이 만큼의 하위 비트가 저장되고, 상기 제 2 래치에는 상기 입력 데이터의 상기 워드 길이로 나타낼 수 있는 최대 길이 만큼의 상위 비트가 저장되며, 상기 제 3 래치와 상기 제 4 래치에는 상기 데이터 액세스 제어신호가 활성화될 때마다 상기 제 1 래치와 상기 제 2 래치에서 출력되는 데이터가 각각 입력되고, 상기 제 1 래치 및 상기 제 2 래치에 저장되어 있는 데이터가 상기 제 3 래치와 상기 제 4 래치로 출력되면, 상기 데이터 액세스 제어신호에 의해 상기 제 1 래치 및 상기 제 2 래치에 새로운 데이터가 각각 입력되도록 구성되어 상기 데이터 액세스 제어신호가 활성화될 때마다 기 저장되어 있는 데이터의 쉬프트가 이루어지고, 또 상기 쉬프트에 따라 발생한 빈 저장 영역에 새로운 데이터의 입력이 이루어지는 제 1 데이터 저장수단과; 상기 입력 데이터 가운데 상기 워드 길이로 나타낼 수 있는 최대 길이 만큼의 상위 비트의 데이터, 또는 상기 제 1 내지 상기 제 3 래치에 저장되어 있는 데이터를 상기 다중화 제어신호에 따라 선택적으로 출력하는 제 1 멀티플렉서와; 상기 제 1 내지 상기 제 4 래치에 저장되어 있는 데이터를 상기 제 1 멀티플렉서와 동일한 값의 상기 다중화 제어신호에 따라 선택적으로 출력하는 제 2 멀티플렉서로 구성되어 상기 다중화 신호에 따라 상기 데이터 저장수단에 저장되어 있는 데이터 가운데 일부를 선택적으로 취하여 출력하는 다중화 수단과; 상기 다중화 수단에서 출력되는 데이터가 입력되고, 상기 쉬프트 제어신호의 값에 따라 쉬프트 포인터가 가변되는 쉬프트를 포함하는 데이터 가변장치.

청구항 2

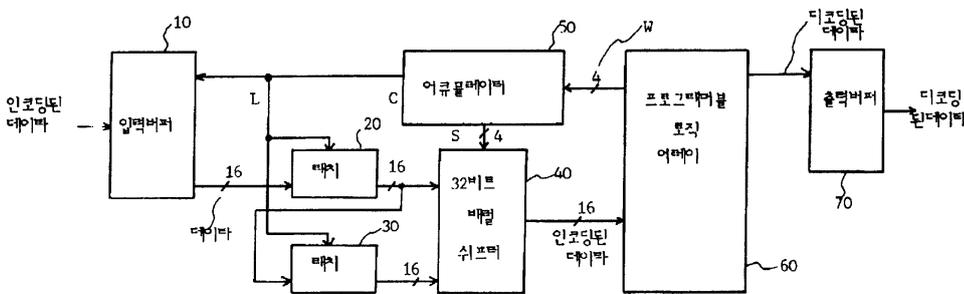
청구항 1에 있어서, 상기 인코더는, 배타적 오어 게이트에 상기 자리올림이 입력되고, 상기 배타적 오어 게이트의 출력이 상기 다중화 제어신호의 제 1 단위 비트로서 출력되며, 상기 제 1 단위 비트가 상기 배타적 오어 게이트의 또 다른 입력으로 피드백되는 제 2 데이터 저장수단과; 제 1 오어 게이트에 상기 제 1 단위 비트가 입력되고, 상기 제 1 오어 게이트의 출력 및 상기 자리올림이 제 1 앤드 게이트에 입력되며, 상기 제 1 앤드 게이트의 출력이 상기 다중화 제어신호의 제 2 단위 비트로서 출력되고, 상기 제 2 단위 비트가 상기 제 1 오어 게이트의 또 다른 입력으로 피드백되는 제 3 데이터 저장수단과; 제 2 오어 게이트에는 상기 제 1 단위 비트와 함께 상기 자리올림의 반전된 값이 입력되고, 제 2 앤드 게이트에는 상기 제 2 오어 게이트의 출력과 상기 제 2 단위 비트가 입력되며, 상기 제 2 앤드 게이트의 출력이 상기 데이터 액세스 제어신호로서 출력되는 제 4 데이터 저장수단을 포함하는 데이터 가변장치.

청구항 3

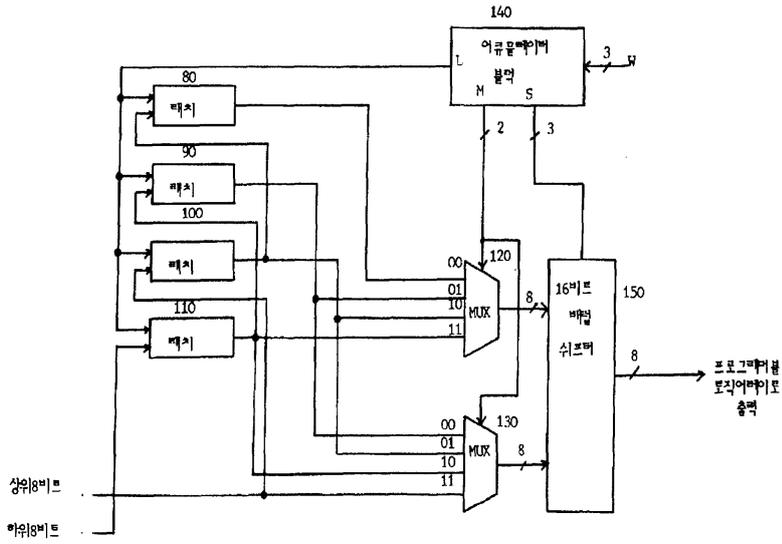
청구항 1에 있어서, 상기 제 1 데이터 저장수단에 입력되는 상기 입력 데이터가 상기 워드 길이로 나타낼 수 있는 최대 길이의 2배의 길이를 갖는 데이터 가변장치.

도면

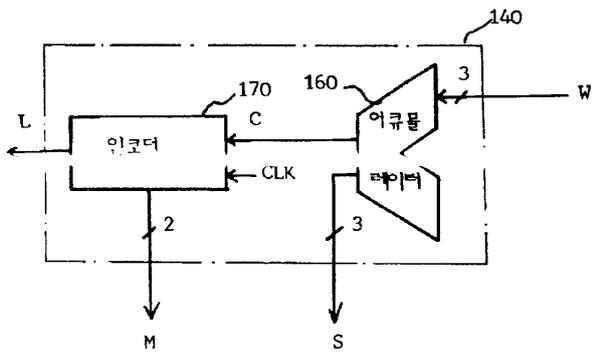
도면1



도면2



도면3



도면4

