(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl.[7]
G06F 13/14

(45) 공고일자 2004년06월09일
(11) 등록번호 10-0435215
(24) 등록일자 2004년05월31일

(21) 출원번호 10-1999-0066030
(22) 출원일자 1999년12월30일

(65) 공개번호 10-2001-0058674
(43) 공개일자 2001년07월06일

(73) 특허권자 삼성전자주식회사
경기도 수원시 영통구 매탄동 416

최기영
서울 서초구 방배3동 신동아아파트 1동 103호

(72) 발명자 신영수
서울특별시 관악구 신림2동 현대아파트 107동 911호

최기영
서울특별시 관악구 신림동 산56-1번지

민병호
서울특별시 성동구 옥수동 극동아파트 4동 1007호

장영훈
경기도 수원시 팔달구 영통동 살구골 현대아파트 726동 503호

(74) 대리인 이영필

심사관 : 송대종

(54) 버스 인코딩/디코딩 장치 및 그 방법

요약

본 발명은 내로우 데이터 버스를 사용하는 저전력 디지털 신호 프로세서 시스템을 위한 버스 인코딩/디코딩 장치 및 그 방법에 관한 것이다.
본 발명에 의한 데이터 버스의 n비트 데이터를 인코딩하는 장치는 n비트 데이터의 최상위 비트의 값이 1이면, 하위 (n-1)비트 데이터를 비트단위로 인버팅하는 조건부인버팅수단; 버스로 출력된 마지막 n비트 데이터를 저장하는 저장수단; 및 조건부인버팅수단에서 조건부로 인버팅된 하위 (n-1)비트 데이터를 저장수단에 저장된 하위 (n-1)비트 데이터와 비트 단위로 배타논리합하는 제1 배타논리합연산수단을 포함하고, n비트 데이터의 최상위 비트와 제1 배타논리합연산수단에 의해 배타논리합된 (n-1)비트의 데이터를 출력하는 것을 특징으로 한다.
본 발명에 의하면, 기존의 기술에 의한 BI 코딩 방법과 BITS 코딩 방법에서 사용하던 여분의 라인과 데이터의 인버팅 여부를 결정하는 회로를 제거함으로써 코아(core)와의 인터페이스 문제와 추가회로에 대한 오버헤드를 줄여 칩 면적을 감소시키고 소비전력를 절감시킬 수 있다.

대표도

도 3a

명세서

도면의 간단한 설명

도 1a는 종래의 기술에 의한 BI 인코더의 회로도이고, 도 1b는 종래의 기술에 의한 BI 디코더의 회로도이다.
도 2a는 종래의 기술에 의한 BITS 인코더의 회로도이고, 도 2b는 종래의 기술에 의한 BITS 디코더의 회로도이다.
도 3a는 본 발명에 의한 HIHR 인코더의 회로도이고, 도 3b는 종래의 기술에 의한 HIHR 디코더의 회로도이다.
도 4a는 음성 신호의 데이터 패턴별 빈도수를 표시한 그래프이고, 도 4b는 음악 신호의 데이터 패턴별 빈도수를 표시한 그래프이다.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 버스 인코딩/디코딩 장치에 관한 것으로서, 특히 내로우 데이터 버스를 사용하는 저전력 디지털 신호 프로세서(Digital Signal Processor:DSP) 시스템을 위한 버스 인코딩/디코딩 장치 및 그 방법에 관한 것이다.
SOC(System On a Chip) 기술의 급속한 발전에 따라 한 칩 안에 여러 가지 기 능 블락을 탑재할 수 있게 되고, 그에 따라 그러한 칩은 외부와의 인터페이스(interface) 시에 더 많은 수의 핀(pin)들을 필요로 하게 된다. 칩의 핀수를 늘리는 것은 칩의 가격과 소비 전력에 큰 부담을 주기 때문에 한정된 수의 핀들을 효율적으로 사용하는 방법을 고려해야 한다.
내로우 데이터 버스(Narrow Data buses) 방식에 의하면, 대기 사이클(wait cycles)에 의한 어느 정도의 성능(performance)의 감소가 허용되는 경우에 칩의 핀수 부족에 관한 문제를 줄일 수 있게 된다. 여기서, 내로우 데이터 버스라 함은 데이터 버스의 비트수가 처리되는 데이터의 비트수보다 적은 데이터 버스를 일컫는다. 나아가, 소비전력을 줄이기 위해서는 핀의 커패시턴스에서 소비되는 소비전력을 최소화하는 버스 코딩(bus coding) 방법이 사용된다. 이를 위해서는 버스로 데이터를 전송할 때, 이전 신호와 현재 신호간의 트랜지션(transition)이 적게 발생하도록 하여 전송하는 방법이 고려될 수 있다.
종래의 기술에 의한 내로우 버스(Narrow Bus)에 대한 코딩 방법에는 BI(Bus-Inverting) 코딩 방법과 BITS(Bus Inverting with Transition signaling) 코딩 방법이 있다. 도 1a 및 도 1b는 각각 BI 코딩 방법에 의한 인코더 및 디코더를 도시한 것이고, 도 2a 및 도 2b는 각각 BITS 코딩 방법에 의한 인코더 및 디코더를 도시한 것이다.
먼저, BI 코딩 방법은 무작위의 데이터 패턴(data pattern)에 대해서는 소비전력 감소 측면에서 잘 동작하지만, 음성(Speech)이나 음악(Music) 같은 데이터 패턴(data pattern)에 대해서는 효과적이지 못하다. 그 이유는 이러한 데이터들이 2 의 보수 형태를 취하므로, 예를 들어 16비트 데이터인 경우 상위 8비트에 0이나 1이 편중되기 때문이다.
음성(Speech)이나 음악(Music) 같은 데이터 패턴에 맞게 개발된 것이 BITS 코딩 방법인데, 이 방법은 BI 코딩방법에 비하여 효과적이다. 하지만, BITS 코딩 방법은 BI 코딩 방법과 마찬가지로 여분의 라인(line)과 데이터(data)를 인버팅(inverting)해야 할 것인가를 결정하는 여분의 회로가 포함되어야 하기 때문에 이에 대한 오버헤드(overhead)가 커지게 된다. 결국, 종래의 기술에 의한 내로우 버스(Narrow Bus)에 대한 코딩 방법들은 여분의 라인(line)에 필요한 핀(pin)을 더 추가해야 하기 때문에, 이미 만들어진 코아(core)를 사용할 때 제약을 준다.

발명이 이루고자 하는 기술적 과제

본 발명의 상기의 문제점을 해결하기 위하여 창작된 것으로서, 기존의 기술에 의한 BI 코딩 방법과 BITS 코딩 방법에서 사용하던 여분의 라인과 머조러티 보터(majority voter)라는 데이터의 인버팅(inverting) 여부를 결정하는 회로를 제거함으로써 코아(core)와의 인터페이스 문제와 추가회로에 대한 오버헤드를 줄여 칩 면적을 감소시키고 소비전력를 절감시키는 내로우 버스 코딩/디코딩 장치 및 그 방법을 제공함을 목적으로 한다.

발명의 구성 및 작용

상기의 목적을 달성하기 위하여, 본 발명에 의한 데이터 버스의 n비트 데이터를 인코딩하는 장치의 일실시예는 상기 n비트 데이터의 최상위 비트의 값이 1이면, 하위 (n-1)비트 데이터를 비트단위로 인버팅하는 조건부인버팅수단; 버스로 출 력된 마지막 n비트 데이터를 저장하는 저장수단; 및 상기 조건부인버팅수단에서 조건부로 인버팅된 하위 (n-1)비트 데이터를 상기 저장수단에 저장된 하위 (n-1)비트 데이터와 비트 단위로 배타논리합하는 제1 배타논리합연산수단을 포함하고, 상기 n비트 데이터의 최상위 비트와 상기 제1 배타논리합연산수단에 의해 배타논리합된 (n-1)비트의 데이터를 출력하는 것을 특징으로 한다.

본 발명에 의한 데이터 버스의 n비트 데이터를 인코딩하는 장치의 일실시예에서 상기 조건부인버팅 수단은 상기 n비트 데이터 중 하위 (n-1)비트 데이터를 각 비트별로 최상위 비트와 배타논리합하는 제2 배타논리합연산수단임을 특징으로 한다.

상기의 다른 목적을 달성하기 위하여, 본 발명에 의한 데이터 버스의 인코딩된 n비트 데이터를 디코딩하는 장치의 일실시예는 상기 버스로부터 입력된 인코딩된 이전 n비트 데이터 중 하위 (n-1)비트 데이터를 저장하는 저장수단; 상기 버스로부터 상기 이전 n비트 데이터 다음에 입력된 인코딩된 현재 n비트의 데이터 중 하위 (n-1)비트 데이터와 상기 저장수단에 저장된 (n-1)비트 데이터를 비트 단위로 배타논리합하는 제1 배타논리합연산수단; 및 상기 버스로부터 입력된 현재 n비트의 데이터 중 최상위 비트의 값이 1이면 상기 배타논리합연산수단에 의해 배타논리합된 (n-1)비트 데이터를 비트단위로 인버팅하는 조건부인버팅수단을 포함하고, 상기 버스로부터 입력된 현재 n비트의 데이터 중 최상위 비트와 상기 조건부인버팅수단에 의해 조건부로 인버팅된 (n-1)비트 데이터를 출력하는 것을 특징으로 한다.

본 발명에 의한 데이터 버스의 인코딩된 n비트 데이터를 디코딩하는 장치의 일실시예에서 상기 조건부인버팅수단은 배타논리합된 (n-1)비트 데이터를 각 비트별로 상기 최상위 비트와 배타논리합하는 제2 배타논리합연산수단임을 특징으로 한다.

상기의 또 다른 목적을 달성하기 위하여, 본 발명에 의한 데이터 버스의 n비트 데이터를 인코딩하는 방법의 일실시예는 (a) 상기 n비트 데이터의 최상위 비트의 값이 1이면, 하위 (n-1)비트 데이터를 비트단위로 인버팅하는 단계; (b) 상기 (a) 단계에서 조건부로 인버팅된 데이터의 하위 (n-1)비트를 마지막으로 인코딩된 데이터의 하위 (n-1)비트와 비트 단위로 배타논리합하는 단계; 및 (c) 상기 n비트 데이터의 최상위 비트와 배타논리합된 (n-1)비트의 데이터를 상기 버스로 출력하는 단계를 포함함을 특징으로 한다.

상기의 또 다른 목적을 달성하기 위하여, 본 발명에 의한 데이터 버스의 인코딩된 n비트 데이터를 디코딩하는 방법의 일실시예는 (a) 상기 버스로부터 입력된 인코딩된 이전 n비트 데이터 중 하위 (n-1)비트 데이터를 저장하는 단계; (b) 상기 버스로부터 이전 n비트 데이터 다음에 입력된 인코딩된 현재 n비트의 데이터 중 하위 (n-1)비트 데이터와 상기 (a) 단계에서 저장된 (n-1)비트 데이터를 비트 단위로 배타논리합하는 단계; (c) 상기 버스로부터 입력된 현재 n비트의 데이터 중 최상위 비트의 값이 1이면 배타논리합된 (n-1)비트 데이터를 비트단위로 인버팅하는 단계; 및 (d) 상기 버스로부터 입력된 현재 n비트의 데이터 중 최상위 비트와 상기 (c) 단계에서 조건부로 인버팅된 (n-1)비트 데이터를 출력하는 단계를 포함함을 특징으로 한다.

이하에서 첨부된 도면을 참조하여 본 발명을 상세히 설명한다.

도 3a에 의하면, 본 발명에 의한 데이터 버스의 8비트 데이터를 인코딩하는 장치의 일실시예는 조건부인버팅수단(10), 저장수단(14) 및 제1 배타적 논리합연산수단(12)를 포함한다. 도 3a에서 데이터 버스의 비트수는 8비트로 예시되어 있으나, 본 발명에서는 데이터 버스의 비트수에 대한 제약을 두지 않는다.

조건부인버팅수단(12)는 8비트 데이터($x_i^0,\cdots,x_i^6,x_i^7$)의 최상위 비트(x_i^7)의 값이 1이면, 하위 7비트 데이터($x_i^0,\cdots,x_i^5,x_i^6$)를 비트단위로 인버팅한다. 도 3a에 의하면, 본 발명에 의한 인코딩 장치의 일실시예에서 조건부인버팅수단(12)은 하위 7비트 데이터($x_i^0,\cdots,x_i^5,x_i^6$)를 각 비트별로 최상위 비트(x_i^7)와 배타논리합하는 7개의 배타논리합 게이트로 이루어짐이 바람직하다.

저장수단(14)은 버스로 출력된 마지막 8비트 데이터($x_{i-1}^0,\cdots,x_{i-1}^6,x_{i-1}^7$)를 저장하며, 도 3a에 의하면 8개의 래치(latch)로 이루어짐이 바람직하다.

제1 배타적논리합연산수단(12)은 인버팅 수단(10)에서 조건부로 인버팅된 하위 (n-1)비트 데이터를 저장수단에 저장된 하위 (n-1)비트 데이터($x_{i-1}^0,\cdots,x_{i-1}^5,x_{i-1}^6$)와 비트 단위로 배타논리합하며, 도 3a에 의하면 7개의 배타논리합 게이트로 이루어짐이 바람직하다.

n비트 데이터의 최상위 비트와 상기 제1 배타논리합연산수단에 의해 배타논리합된 (n-1)비트의 데이터는 저장수단(14)을 거쳐 버스로 출력된다.

도 3b에 의하면, 본 발명에 의한 데이터 버스의 인코딩된 n비트 데이터를 디코딩하는 장치의 일실시예는 저장수단(20), 제1 배타논리합연산수단(22) 및 조건부인버팅수단(24)를 포함한다. 도 3b에서 데이터 버스의 비트수는 8비트로 예시되어 있으나, 본 발명에서는 데이터 버스의 비트수에 대한 제약을 두지 않는다.

저장수단(20)은 버스로부터 입력된 인코딩된 이전 8비트 데이터($z_{i-1}^0,\cdots,z_{i-1}^6,z_{i-1}^7$) 중 하위 7비트 데이터($z_{i-1}^0,\cdots,z_{i-1}^5,z_{i-1}^6$)를 저장하며, 도 3b에 의하면 7개의 래치로 이루어짐이 바람직하다.

제1 배타논리합연산수단(22)은 버스로부터 이전 8비트 데이터($z_{i-1}^0,\cdots,z_{i-1}^6,z_{i-1}^7$) 다음에 입력된 인코딩된 현재 8비트의 데이터($z_i^0,\cdots,z_i^6,z_i^7$) 중 하위 7비트 데이터($z_i^0,\cdots,z_i^5,z_i^6$)와 상기 저장수단에 저장된 7비트 데이터($z_{i-1}^0,\cdots,z_{i-1}^5,z_{i-1}^6$)를 비트 단위로 배타논리합하며, 도 3b에 의하면 7개의 배타논리합 게이트로 이루어짐이 바람직하다.

조건부인버팅수단(24)는 상기 버스로부터 입력된 인코딩된 현재 8비트의 데이터 중 최상위 비트(z_i^7)의 값이 1이면 상기 배타논리합연산수단에 의해 배타논리 합된 7비트 데이터를 비트단위로 인버팅한다. 도 3b에 의하면, 본 발명에 의한 디코딩 장치의 일실시예에서 조건부인버팅수단(24)은 배타논리합된 (n-1)비트 데이터를 각 비트별로 각 비트별로 최상위 비트(z_i^7)와 배타논리합하는 7개의 배타논리합 게이트로 이루어짐이 바람직하다.

상기 버스로부터 입력된 인코딩된 8비트의 데이터 중 최상위 비트와 상기 조건부인버팅수단에 의해 조건부로 인버팅된 7비트 데이터가 디코딩된 데이터로서 출력된다.

이후에는, 본 발명에서 제시된 코딩 방법을 종래의 기술에 의한 BI(Bus-Inverting) 코딩 방법 및 BITS(Bus Inverting with Transition signaling) 코딩 방법과 비교하여 HIHR(Half Identity Half Reverse) 코딩방법이라 한다.

이하에서, 다음과 같은 데이터 패턴을 갖는 인간 음성(human voice)를 예로 들어 본 발명의 동작을 설명한다.

5.683954 0000010110101111 X0:00000101,X1:10101111

10.578125 0000101010010100 X2:00001010,X3:10010100
-5.625000 1111101001100000 X4:11111010,X5:01100000
1.019531 0000000100000100 X6:00000001,X7:00000100
3.484375 0000001101111100 X8:00000011,X9:01111100
상기 첫 번째 열은 음성 데이터 패턴을 10진수로 표현한 것이고, 두 번째 열은 16비트의 2의 보수 고정소숫점 표현이고, 세 번째 열은 8비트 데이터 버스에 전 송되기 위한 형태로 표현한 것이다.
상기 두 번째 열의 16비트 데이터를 8비트 데이터 버스에 실어 보내는 경우를 살펴보면 다음과 같다. 보내는 데이터의 순서는 16비트 데이터를 상기 세 번째 열과 같이 분리하여, X0,X1,…의 순으로 보내게 된다.

수학식 1

$$Y_i = \begin{cases} x_i^{n-1} | X_i(n-1), & \text{if } x_i^{n-1}=0 \\ x_i^{n-1} | \overline{X_i(n-1)}, & otherwise \end{cases}$$

$$Z_i = y_i^{n-1} | TS(Y_i(n-1), Z_{i-1}(n-1))$$

8비트 데이터를 보낼 때는 수학식 1에 제시된 것과 규칙으로 코딩하여 보내게 된다. 즉, X0=00000101이므로, 먼저 $Y_{ ¡}$ 식을 적용하면 최상위 비트가 0이므로 인버팅을 수행하지 않아 Y1=00000101이고, $Z_{ ¡}$ 를 적용하면 Y1의 최상위 비트를 제외한 Y1의 나머지 7비트와 바로 이전에 버스에 실렸던 데이터 Z0을 배타논리합(exclusive OR) 연산을 하여 Z1은 00000101을 유지하게 된다. 이때, Z0은 00000000으로 초기화된 것으로 본다.
따라서, Z1이 버스에 실리게 된다. 다음 값인 X1=10101111의 경우에는 최상위 비트가 1이므로 Y2=11010000으로 인버팅되고, Z2는 Y2의 최상위 비트를 자신의 최상위 비트로 유지하고 Y2의 하위 7비트와 Z1의 하위 7비트를 배타논리합(exclusive OR)하게 된다. 그러면, 1|TR(Y2,Z1) = 1|TR(1010000,0000101)은 결국 1과 1010101이 되어, Z2는 11010101이 된다. 이후, Z2가 버스에 실리게 된다.
상기 예시된 사람 음성의 데이터 패턴을 본 발명에 의한 HIHR 코딩방법에 적용하면, 다음과 같이 인코딩된다.
X Y Z
X0: 00000101 ----->00000101----->00000101
X1: 10101111 ----->11010000----->11010101
X2: 00001010 ----->00001010----->01011111
X3: 10010100 ----->11101011----->10110100
X4: 11111010 ----->10000101----->10110001
X5: 01100000 ----->01100000----->01010001
X6: 00000001 ----->00000001----->01010000
X7: 00000100 ----->00000100----->01010100
X8: 00000011 ----->00000011----->01010111
X9: 01111100 ----->01111100----->00101011
본 발명에 의한 HIHR 코딩방법에서 디코딩 과정은 인코딩 과정의 역 과정으로서, 수학식 2로 표현된다.

수학식 2

$$Y_i = z_i^{n-1} | TS(Z_i(n-1), Z_{i-1}(n-1))$$

$$X_i = \begin{cases} y_i^{n-1} | Y_i(n-1), & \text{if } z_i^{n-1}=0 \\ y_i^{n-1} | \overline{Y_i(n-1)}, & otherwise \end{cases}$$

종래의 기술에 의한 BI 코딩 방법과 BITS 코딩 방법, 그리고 본 발명에 의한 HIHR 코딩 방법의 세가지에 대하여 트랜지션 수를 비교해 보면 다음과 같다.
먼저, BI 코딩 방법에 의한 트랜지션 수를 표시한다.
X Z 트랜지션 수
X0: 00000101 ----->000001010 0
X1: 10101111 ----->101011110 4
X2: 00001010 ----->000010100 4
X3: 10010100 ----->011010111 4
X4: 11111010 ----->111110100 4
X5: 01100000 ----->011000000 4
X6: 00000001 ----->000000010 3
X7: 00000100 ----->000001000 2
X8: 00000011 ----->000000110 3
X9: 01111100 ----->100000111 2 = 30번의 트랜지션 발생
BI 코딩 방법에서는 원래의 신호와 이전 신호의 트랜지션 수가 버스 폭(bus width)의 반이 넘을 때 원 신호를 인버트(

invert)한다. 그리고, 비교할 때에는 트랜지션(transition)된 신호와 비교한다.
다음, BITS 코딩 방법에 의한 트랜지션 수를 표시한다.
1의 개수 Y|I Z|I 트랜지션 수
X0: 00000101 ----> 000001010----->000001010
X1: 10101111 ----> 010100001----->010101011 3
X2: 00001010 ----> 000010100----->010111110 3
X3: 10010100 ----> 100101000----->110010110 3
X4: 11111010 ----> 000001011----->110011101 3
X5: 01100000 ----> 011000000----->101011100 3
X6: 00000001 ----> 000000010----->101011110 1
X7: 00000100 ----> 000001000----->101010110 1
X8: 00000011 ----> 000000110----->101010000 2
X9: 01111100 ----> 100000111----->001010111 4 = 23번의 트랜지션
BITS 코딩 방법은 BI 코딩 방법과는 달리, 자체 신호의 1의 개수를 세어서 인버트(invert)할 것인가 여부를 결정한다. 1의 개수가 버스 폭(bus width)의 반보다 크면 트랜지션을 한다.
마지막으로, HIHR 코딩 방법에 의한 트랜지션 수를 표시한다.
X Y Z 트랜지션 수
X0: 00000101 ---->00000101---->00000101
X1: 10101111 ---->11010000---->11010101 3
X2: 00001010 ---->00001010---->01011111 3
X3: 10010100 ---->11101011---->10110100 6
X4: 11111010 ---->10000101---->10110001 2
X5: 01100000 ---->01100000---->01010001 3
X6: 00000001 ---->00000001---->01010000 1
X7: 00000100 ---->00000100---->01010100 1
X8: 00000011 ---->00000011---->01010111 2
X9: 01111100 ---->01111100---->00101011 5 = 26번의 트랜지션
본 발명에 의한 HIHR 코딩 방법은 원 신호의 최상위 비트(Most Significant Bit:MSB)를 이용하여 Y 데이터를 만든다. MSB가 1이면 MSB는 그대로 두고 나머지 7비트를 인버팅하고, MSB가 0이면 원 데이터를 그대로 유지한다. 다음으로, 새롭게 처리된 Y 데이타와 이전의 Z 데이타를 비트 단위로 배타논리합(exclusive OR) 연산한다. 배타논리합(Exclusive OR) 연산은 피연산자가 서로 다른 값일 경우만 논리 1이고, 다를 때에는 논리 0을 출력한다.
세 가지 경우를 비교하였을 때 BITS 코딩 방법이 가장 트랜지션이 적고, 그 다음이 HIHR 코딩 방법임을 알 수 있다. 그러나, BITS 코딩 방법은 여분의 라인(line)과 추가적인 회로가 필요하므로 칩 면적이나 가격을 비교한다면 HIHR 코딩 방법이 더 효과적임을 알 수 있다.

[표 1]
전체 트랜지션 수의 비교(음성 및 클래식 음악)

패턴 이름	패턴 수	Unencoded #trans.	BI 코딩 방법		BITS 코딩 방법		HIHR 코딩 방법	
			#trans.	%red.	#trans.	%red.	#trans.	%red.
음성출력	3276	12375	10423	15.8	6518	47.3	7348	40.6
클래식음악	3276	13204	10690	19.0	8719	34.0	9585	27.4

[표 2]
전체 트랜지션 수의 비교(FFT 연산결과)

패턴 이름	패턴 수	Unencoded #trans.	BI 코딩 방법		BITS 코딩 방법		HIHR 코딩 방법	
			#trans.	%red.	#trans.	%red.	#trans.	%red.
FFTR데이터	1566	7767	6375	17.9	3690	52.5	4337	44.2
FFTI데이터	1566	7714	6351	17.7	3684	52.2	4160	46.1

표 1과 표 2는 세 가지 코딩 방법에서의 트랜지션 수를 서로 비교한 것이다. 이때, 각 패턴 데이터는 16비트이지만, 이 데이터를 8비트로 나누고 나뉘어진 데이터에 세 가지 코딩 방법을 적용한 결과 데이터간의 트랜지션 수를 비교한 것

이다. 표 1은 음성(Speech)과 클래식 음악(Classic Music)에 대한 실험치이고, 표 2는 FFT 연산에 대한 결과이다.

[표 3]
인코더 및 디코더의 면적, 지연시간, 소비전력 비교

	인코더			디코더		
	BI	BITS	HIHR	BI	BITS	HIHR
면적(㎛ ^{2})	19076	18626	4659	2662	11392	9968
지연시간(ns)	3.29	3.87	0.38	0.15	0.38	0.38
소비전력(㎼)	2309	2409	411	120	2102	1618

표 3에는 세 가지 코딩 방법 각각의 경우에 사용되는 인코더(Encoder)와 디코더(Decoder)의 면적(area), 지연시간(delay) 그리고 소비전력(power)을 음성(Speech) 패턴 데이터를 적용시켜 서로 비교해 놓은 것이다. 표 3의 데이터에서 제시된 바와 같이, 본 발명에 의한 HIHR 코딩 방법이 인코더나 디코더에서 다른 두 가지 경우보다 면적, 지연시간 그리고 소비전력을 종합하여 판단할 때 가장 좋은 성능을 보임을 알 수 있다.

[표 4]
OFF 칩 구동 중 전체 전력 소모의 비교

패턴 명	$C_{offchip}$ (pF)	Unencoded P(㎽)	BI		BITS		HIHR	
			P(㎽)	Red.(%)	P(㎽)	Red.(%)	P(㎽)	Red.(%)
음성출력	10	27.9	28.2	-1.1	18.1	35.1	17.6	36.9
	15	38.2	36.9	3.4	23.5	38.5	23.7	38.0
	20	48.5	45.6	6.0	28.9	40.4	29.8	38.6
	25	58.9	54.3	7.8	34.3	41.8	35.9	39.0
	30	69.1	63.0	8.8	39.7	42.5	42.0	39.2
고전음악	10	29.7	28.1	5.4	23.2	21.9	22.5	24.2
	15	40.7	36.7	9.8	30.4	25.3	30.4	25.3
	20	51.7	45.3	12.4	37.7	27.1	38.4	25.7
	25	62.6	53.9	13.9	44.9	28.3	46.3	26.0
	30	73.6	62.5	15.1	52.2	29.1	54.3	26.2

표 4는 음성(Speech)과 고전음악(Classic Music)에 세 가지 코딩 방법을 사용하였을 때의 인코더측만 측정하여 서로 비교해 놓은 것이다. 표 4에서 알 수 있듯이, HIHR의 소비전력 절감(power saving)은 $C_{offchip}$ = 10pF일 때의 36.9%로부터 $C_{offchip}$ =30pF일 때의 39.2%가 이루어짐을 볼 수 있다. 이는 BITS 코딩 방법과 거의 동등한 수준이지만, 칩 면적과 여분의 라인에 대한 가격(cost)을 감안할 때 HIHR 방법이 더 좋은 방법이라 할 수 있다.
디지털(Digital) 보청기에 HIHR 코딩 방법을 적용하여 소비전력을 측정해 보았다. 이 때, 외부 메모리(External Memory)와 디지털 신호 프로세서 코아(DSP core) 사이의 데이터 전송시에 HIHR 코딩 방법을 적용하였다. 외부 메모리(External Memory)를 구동할 때에는 10~30pF의 범위로 커패시턴스(capacitance)를 변화시키면서 시뮬레이션(Simulation)하였고, 내부 메모리(Internal Memory)를 구동할 때에는 3pF을 사용하였다. 전원은 3.3V, 클럭(Clock) 주파수는 10㎒를 적용하였다. 표 5는 16비트 패턴 데이터를 8비트 데이터 버스에 실어보내는 것을 예시한 것이다.

[표 5]
OFF 칩 구동 중 전체 전력 소모의 비교

$C_{offchip}$ (pF)	Uncoded P(㎽)	HIHR 코딩방법	
		P(㎽)	Red.(%)
10	1.60	1.00	37.5
15	2.16	1.34	38.0

20	2.73	1.64	39.9
25	3.30	1.98	40.0
30	3.87	2.29	40.8

표 5는 래치(Latch)와 출력 구동기(Output Driver)만을 포함하여 측정한 값이다. 디코더에서의 소모 전력은 153μW이다. 따라서, HIHR 코딩 방법은 uncoded 방식과 비교시 $C_{offchip}$ 이 10pF일 때 (1.6mW - 1mW - 0.153mW)/(1.6mW) = 27.9%, 30pF일 때 (3.87mW - 2.29mW - 0.153mW)/(3.87mW) = 36.9%의 전력을 절약할 수 있었다.
다음, 16비트 패턴 데이터를 16비트 데이터 버스에 실어보내는 경우, 칩의 내부 노드에서의 커패시턴스는 대략 3pF이므로 이를 이용하여 시뮬레이션하면 Unencoded의 경우 소비전력은 1.63㎽이고, HIHR 코딩 방식에서의 인코더 측의 소비전력은 1.05㎽이고, 디코더 측의 소비전력은 0.276㎽으로 측정되었다. 따라서, (1.63㎽ - 1.05㎽ -0.276㎽)/(1.63㎽) = 18.7%의 전력을 절약할 수 있다. 따라서, 비록 HIHR 코딩 방식은 내로우 데이터 버스(Narrow Data Bus)에 대하여 창작되었지만, 도 4a 및 도 4b에 도시된 음성(Speech)이나 음악(Music) 패턴 데이터와 같이 0 주위에 많은 데이터들이 몰려 있는 분포를 보이면 버스 폭(Bus Width)과 데이터 폭(Data Width)이 같아도 효과적임을 볼 수 있다.

발명의 효과

본 발명에 의하면, 기존의 기술에 의한 BI 코딩 방법과 BITS 코딩 방법에서 사용하던 여분의 라인과 머조러티 보터(majority voter)라는 데이터의 인버팅(inverting) 여부를 결정하는 회로를 제거함으로써 코아(core)와의 인터페이스 문제와 추가회로에 대한 오버헤드를 줄여 칩 면적을 감소시키고 소비전력를 절감시킬 수 있다.
본 발명은 내로우 데이터 버스(Narrow Data Bus) 뿐만 아니라 0 주위에 많은 데이터들이 몰려 있는 분포를 보이는 패턴 데이터에 대해서는 버스 폭(Bus Width)과 데이터 폭(Data Width)이 같아도 소비전력을 절감할 수 있다.

(57) 청구의 범위

청구항 1.
데이터 버스의 n비트 데이터를 인코딩하는 장치에 있어서,
상기 n비트 데이터의 최상위 비트의 값이 1이면, 하위 (n-1)비트 데이터를 비트단위로 인버팅하는 조건부인버팅수단;
버스로 출력된 마지막 n비트 데이터를 저장하는 저장수단; 및
상기 조건부인버팅수단에서 조건부로 인버팅된 하위 (n-1)비트 데이터를 상기 저장수단에 저장된 하위 (n-1)비트 데이터와 비트 단위로 배타논리합하는 제1 배타논리합연산수단을 포함하고,
상기 n비트 데이터의 최상위 비트와 상기 제1 배타논리합연산수단에 의해 배타논리합된 (n-1)비트의 데이터를 출력하는 것을 특징으로 하는 버스 인코딩 장치.

청구항 2.
제1항에 있어서,
상기 조건부인버팅 수단은 상기 n비트 데이터 중 하위 (n-1)비트 데이터를 각 비트별로 최상위 비트와 배타논리합하는 제2 배타논리합연산수단임을 특징으로 하는 버스 인코딩 장치.

청구항 3.
데이터 버스의 인코딩된 n비트 데이터를 디코딩하는 장치에 있어서,
상기 버스로부터 입력된 인코딩된 이전 n비트 데이터 중 하위 (n-1)비트 데이터를 저장하는 저장수단;
상기 버스로부터 이전 n비트 데이터 다음에 입력된 인코딩된 현재 n비트의 데이터 중 하위 (n-1)비트 데이터와 상기 저장수단에 저장된 (n-1)비트 데이터를 비트 단위로 배타논리합하는 제1 배타논리합연산수단; 및
상기 버스로부터 입력된 현재 n비트의 데이터 중 최상위 비트의 값이 1이면 상기 배타논리합연산수단에 의해 배타논리합된 (n-1)비트 데이터를 비트단위로 인버팅하는 조건부인버팅수단을 포함하고,
상기 버스로부터 입력된 현재 n비트의 데이터 중 최상위 비트와 상기 조건부인버팅수단에 의해 조건부로 인버팅된 (n-1)비트 데이터를 출력하는 것을 특징으로 하는 버스 디코딩 장치.

청구항 4.
제3항에 있어서,
상기 조건부인버팅 수단은 배타논리합된 (n-1)비트 데이터를 각 비트별로 상 기 최상위 비트와 배타논리합하는 제2 배타논리합연산수단임을 특징으로 하는 버스 인코딩 장치.

청구항 5.
데이터 버스의 n비트 데이터를 인코딩하는 방법에 있어서,
(a) 상기 n비트 데이터의 최상위 비트의 값이 1이면, 하위 (n-1)비트 데이터를 비트단위로 인버팅하는 단계;
(b) 상기 (a) 단계에서 조건부로 인버팅된 데이터의 하위 (n-1)비트를 마지막으로 인코딩된 데이터의 하위 (n-1)비트와 비트 단위로 배타논리합하는 단계; 및

(c) 상기 n비트 데이터의 최상위 비트와 배타논리합된 (n-1)비트의 데이터를 상기 버스로 출력하는 단계를 포함함을 특징으로 하는 버스 인코딩 방법.

청구항 6.

데이터 버스의 인코딩된 n비트 데이터를 디코딩하는 방법에 있어서,

(a) 상기 버스로부터 입력된 인코딩된 이전 n비트 데이터 중 하위 (n-1)비트 데이터를 저장하는 단계;

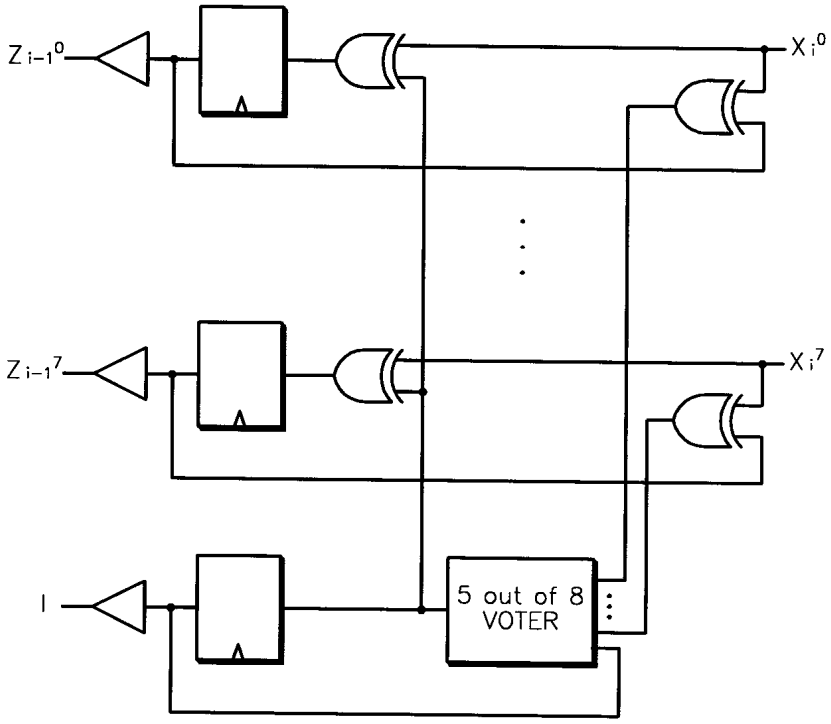
(b) 상기 버스로부터 이전 n비트 데이터 다음에 입력된 인코딩된 현재 n비트의 데이터 중 하위 (n-1)비트 데이터와 상기 (a) 단계에서 저장된 (n-1)비트 데이터를 비트 단위로 배타논리합하는 단계;

(c) 상기 버스로부터 입력된 현재 n비트의 데이터 중 최상위 비트의 값이 1이면 배타논리합된 (n-1)비트 데이터를 비트단위로 인버팅하는 단계; 및

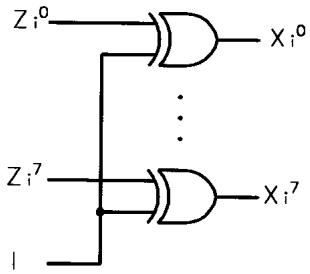
(d) 상기 버스로부터 입력된 현재 n비트의 데이터 중 최상위 비트와 상기 (c) 단계에서 조건부로 인버팅된 (n-1)비트 데이터를 출력하는 단계를 포함함을 특징으로 하는 버스 디코딩 방법.

도면

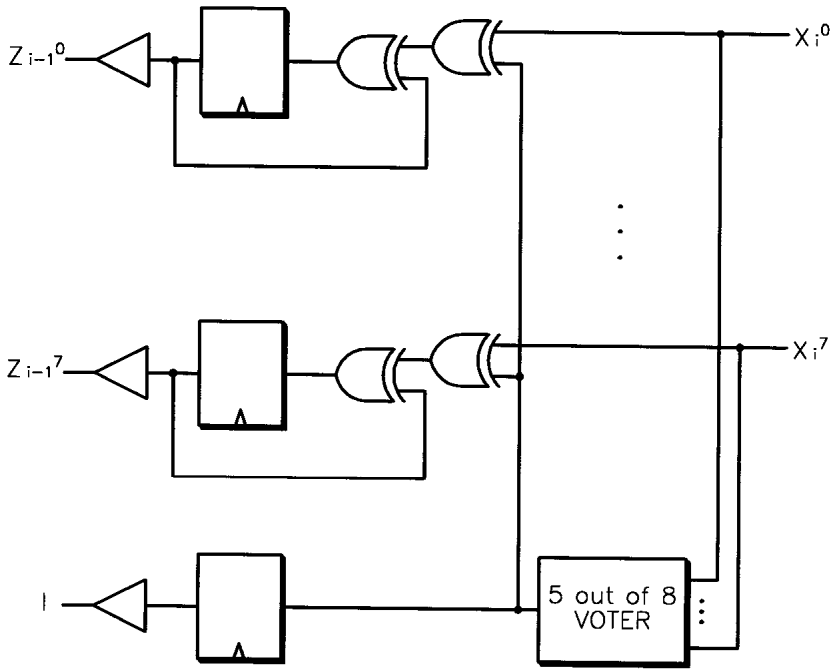
도면1a



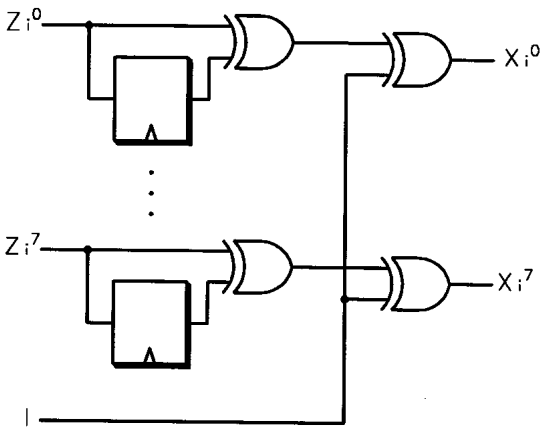
도면1b



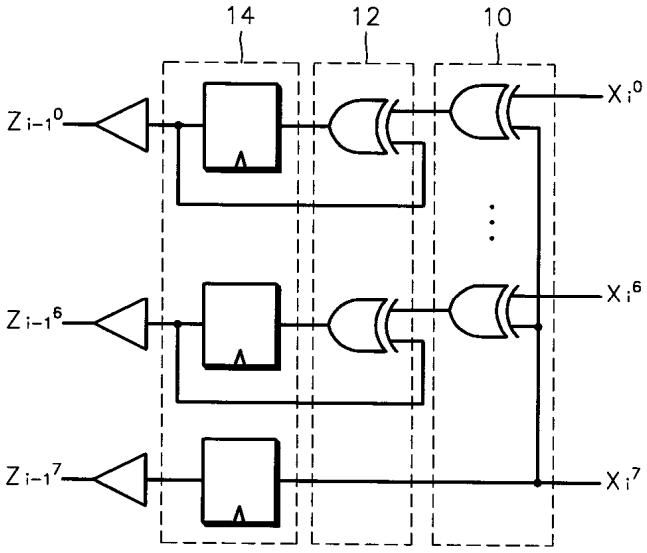
도면2a



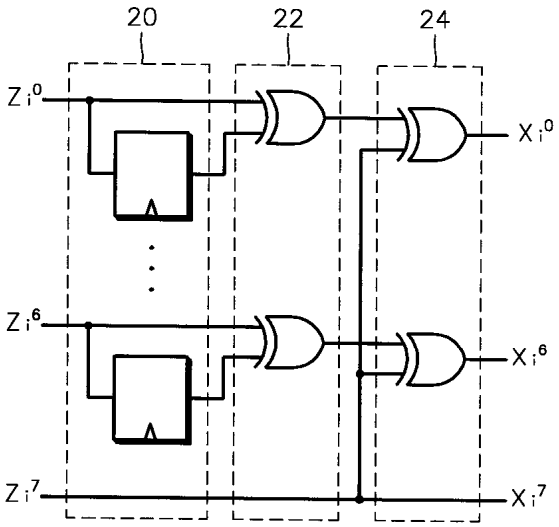
도면2b



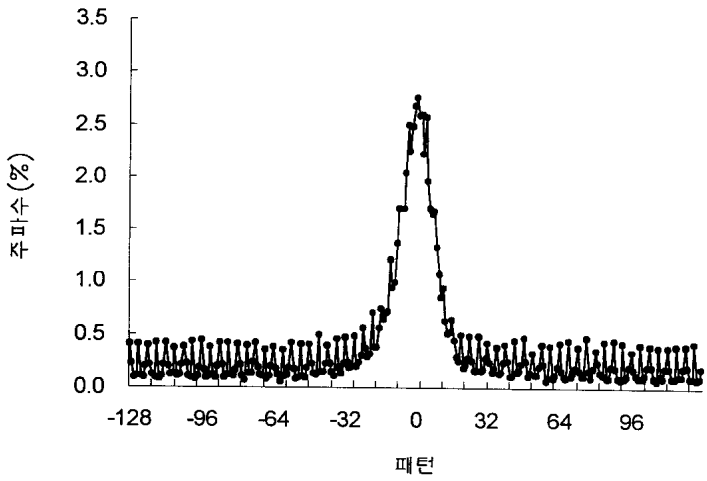
도면3a



도면3b



도면4a



도면4b