



(19)대한민국특허청(KR)  
(12) 등록특허공보(B1)

(51) 。 Int. Cl. G06F 13/38 (2006.01)	(45) 공고일자 (11) 등록번호 (24) 등록일자	2006년12월14일 10-0657411 2006년12월07일
---	-------------------------------------	--

(21) 출원번호 (22) 출원일자 심사청구일자	10-2005-0084265 2005년09월09일 2005년09월09일	(65) 공개번호 (43) 공개일자
----------------------------------	---	------------------------

(73) 특허권자                    한국과학기술원  
                                      대전 유성구 구성동 373-1

(72) 발명자                        이정협  
                                      대구 중구 동인동1가 33-1번지, 시티타운 상가 15호

                                      신영수  
                                      대전 유성구 도룡동 383-2번지, 과기원교수APT 2동 501호

(74) 대리인                        이종일

(56) 선행기술조사문헌 JP2002108805 A KR1020000013044 A US5062072 A * 심사관에 의하여 인용된 문헌	JP2002152052 A KR1020010026377 A
--	-------------------------------------

심사관 : 권오성

전체 청구항 수 : 총 6 항

(54) 천이 수를 유지하는 내로우 버스 인코딩/디코딩 장치

(57) 요약

본 발명은 천이 수를 유지하는 내로우 버스 인코딩/디코딩 장치에 관한 것으로, 특히 와이드(wide) 비트 데이터를 직렬화(serialize)하여 내로우(narrow)버스로 보낼 때 원 데이터의 천이 수를 보존하며 성능 감소가 없는 버스 인코딩/디코딩 장치에 관한 것이다.

본 발명에 따른 천이 수를 유지하는 내로우 버스 인코딩 장치는 짝수 값을 가지는 n-비트 원 데이터를 n/2-비트 데이터로 직렬화하는 시리얼라이저 수단과, 상기 시리얼라이저 수단에서 출력된 데이터를 저장하는 n/2 개의 저장 수단1, 상기 시리얼라이저 수단에서 출력된 데이터와 상기 저장 수단에서 출력된 데이터의 배타논리합을 구하는 n/2 개의 배타논리합연산 수단1을 포함하여 구성되는 것을 특징으로 한다.

대표도

도 1

## 특허청구의 범위

### 청구항 1.

짝수 값을 가지는  $n$ -비트 원 데이터를  $n/2$ -비트 데이터로 직렬화하는 시리얼라이저 수단과,

상기 시리얼라이저 수단에서 출력된 데이터를 저장하는  $n/2$  개의 저장 수단1과,

상기 시리얼라이저 수단에서 출력된 데이터와 상기 저장 수단에서 출력된 데이터의 배타논리합을 구하는  $n/2$  개의 배타논리합연산 수단1을 포함하여 구성되는 것을 특징으로 하는 천이 수를 유지하는 내로우 버스 인코딩 장치.

### 청구항 2.

청구항 1항에 있어서,

상기 시리얼라이저 수단은  $n/2$  개의 MUX와 MUX의 입력을 생성하기 위한 Counter로 구성된 것을 특징으로 하는 천이 수를 유지하는 내로우 버스 인코딩 장치.

### 청구항 3.

청구항 2항에 있어서,

2의 지수승 형태로 표현된  $n$ -비트 원 데이터를  $n/4, n/8, \dots, n/n$ -비트 내로우 버스로 전송하는 것을 특징으로 하는 천이 수를 유지하는 내로우 버스 인코딩 장치.

### 청구항 4.

$n/2$ -비트 내로우 버스로 전송되어 온 데이터와 저장 수단2에서 피드백된 이전 배타논리합 결과 데이터의 배타논리합을 구하는 배타논리합연산 수단2와,

상기 배타논리합연산 수단2에서 출력된 결과를 저장하여 이전 배타논리합 결과 데이터를 생성하는 저장 수단2와,

상기 저장 수단2에서 출력된  $n/2$ -비트 데이터를 원  $n$ -비트 데이터로 복원하는 디시리얼라이저 수단을 포함하여 구성되는 것을 특징으로 하는 천이 수를 유지하는 내로우 버스 디코딩 장치.

### 청구항 5.

청구항 4항에 있어서,

상기 디시리얼라이저 수단은  $n/2$  개의 DEMUX와 DEMUX의 입력을 생성하기 위한 counter로 구성된 것을 특징으로 하는 천이 수를 유지하는 내로우 버스 디코딩 장치.

### 청구항 6.

청구항 5항에 있어서,

2의 지수승 형태로 표현된  $n$  값에 대해,  $n/4, n/8, \dots, n/n$ -비트 내로우 버스로 전송되어온 데이터를 원 데이터로 복원하는 것을 특징으로 하는 천이 수를 유지하는 내로우 버스 디코딩 장치.

## 명세서

### 발명의 상세한 설명

#### 발명의 목적

##### 발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 천이 수를 유지하는 내로우 버스 인코딩/디코딩 장치에 관한 것으로, 특히 와이드(wide) 비트 데이터를 직렬화(serialize)하여 내로우(narrow)버스로 보낼 때 원 데이터의 천이 수를 보존하며 성능 감소가 없는 버스 인코딩/디코딩 장치에 관한 것이다.

SoC (System-on-a-Chip) 디바이스들은 빠른 연산과 많은 데이터 전달을 요구하는 복잡한 어플리케이션(application) 수행을 위한 장치이다. SoC에서는 온 칩(on-chip) 모듈들 및 모듈들을 연결하는 버스 라인(bus line)의 수가 대단히 많은데, 이러한 많은 수의 버스선은 스큐(skew), 누화(crosstalk), 와이어링(wiring)의 어려움, 넓은 면적 차지등과 같은 많은 문제점을 발생시킨다. 따라서 버스 라인을 줄일 수 있는 장치가 필요하게 된다. 그러나 버스 라인을 줄이는 것만으로 모든 문제가 해결 되는 것은 아니다. 앞서 예기하였듯 SoC 디바이스에서는 많은 데이터를 빠른 시간 내에 수행 처리해야 하기 때문에 성능이 감소되어서는 안 되며 휴대용 기기에 적용하기 위해 전력 소모가 많아져서도 안 된다. 결국 버스 라인의 수는 감소시키며, 성능 및 전력 소모의 손해는 없는 간단한 장치가 필요하게 된다.

##### 발명이 이루고자 하는 기술적 과제

본 발명은 상기한 바와 같은 종래 기술의 불편함을 해결하기 위하여 안출된 것으로서,

본 발명의 목적은 원 데이터 비트 수보다 적은 내로우 버스 라인을 사용하면서 성능 및 천이 수는 유지시켜주는 간단한 버스 인코딩/디코딩 장치를 제공하는데 그 목적이 있다.

#### 발명의 구성

상기 목적을 달성하기 위하여 본 발명에 따른 천이 수를 유지하는 내로우 버스 인코딩 장치는 짝수 값을 가지는  $n$ -비트 원 데이터를  $n/2$ -비트 데이터로 직렬화하는 시리얼라이저 수단과, 상기 시리얼라이저 수단에서 출력된 데이터를 저장하는  $n/2$  개의 저장 수단1, 상기 시리얼라이저 수단에서 출력된 데이터와 상기 저장 수단에서 출력된 데이터의 배타논리합을 구하는  $n/2$  개의 배타논리합연산 수단1을 포함하여 구성되는 것을 특징으로 한다.

본 발명에 따른 천이 수를 유지하는 내로우 버스 인코딩 장치의 상기 시리얼라이저 수단은  $n/2$  개의 MUX와 MUX의 입력을 생성하기 위한 Counter로 구성된 것을 특징으로 한다.

본 발명에 따른 천이 수를 유지하는 내로우 버스 인코딩 장치는 2의 지수승 형태로 표현된  $n$ -비트 원 데이터를  $n/4, n/8, \dots, n/n$ -비트 내로우 버스로 전송하는 것을 특징으로 한다.

본 발명에 따른 천이 수를 유지하는 내로우 버스 디코딩 장치는  $n/2$ -비트 내로우 버스로 전송되어 온 데이터와 저장 수단2에서 피드백된 이전 배타논리합 결과 데이터의 배타논리합을 구하는 배타논리합연산 수단2와, 상기 배타논리합2에서 출력된 결과를 저장하여 이전 배타논리합 결과 데이터를 생성하는 저장 수단2와, 상기 저장 수단2에서 출력된  $n/2$ -비트 데이터를 원  $n$ -비트 데이터로 복원하는 디시리얼라이저 수단을 포함하여 구성되는 것을 특징으로 한다.

본 발명에 따른 천이 수를 유지하는 내로우 버스 디코딩 장치의 상기 디시리얼라이저 수단은  $n/2$  개의 DEMUX와 DEMUX의 입력을 생성하기 위한 Counter로 구성된 것을 특징으로 한다.

본 발명에 따른 친이 수를 유지하는 내로우 버스 디코딩 장치는 2의 지수승 형태로 표현된 n 값에 대해, n/4, n/8, ..., n/n-비트 내로우 버스로 전송되어온 데이터를 원 데이터로 복원하는 것을 특징으로 한다.

이하, 바람직한 실시예를 통하여 본 발명에 따른 친이 수를 유지하는 내로우 버스 인코딩/디코딩 장치를 보다 구체적으로 살펴본다.

도1은 본 발명의 실시예에 따른 인코딩 장치의 회로도이다.

본 발명의 실시예에 따른 입력 데이터를 인코딩하는 장치는 도1에 도시된바와 같이 시리얼라이저 수단(10), 저장 수단(12), 배타논리합연산 수단(14)으로 구성된다.

시리얼라이저 수단(10)에서는 4개의 2:1MUX (multiplexor)가 사용되며 상기 MUX들은 1개의 Counter에 의해 동기 된

다. 시리얼라이저 수단(10) 상기 MUX들을 통하여 8비트 입력데이터( $X_7 X_6 \dots X_1 X_0$ )를 2개의 4비트 데이터( $X_0$ :  $X_7 X_5 X_3 X_1$  과  $X_1$ :  $X_6 X_4 X_2 X_0$ )로 직렬화한다.

상기 Counter는 상기 8비트 입력 데이터( $X_7 X_6 \dots X_1 X_0$ )가 f의 주파수에 동기 되어 입력된다면  $2 \cdot f$ 의 주파수로 동기 되어 0과1을 연속적으로 출력한다. 따라서 상기 Counter의 출력이 0일 때 상기 MUX를 통해  $X_0$  데이터가 우선 출력되고  $1/(2 \cdot f)$ 시간 후 상기 Counter의 출력이 1로 변하면  $X_1$  데이터가 출력된다. 이와 같은 동작을 상기 8비트 입력데이터( $X_7 X_6 \dots X_1 X_0$ )가 동기 될 때 마다( $1/f$  주기로) 반복한다.

실시예에서는 직렬화 방법을 한가지로 예시하였으나, n! (n은 원 데이터의 비트 폭)가지의 직렬화 방법이 가능하다.

저장 수단(12)은 4개의 플립플롭(flip-flop)으로 구성되어 있으며 상기 시리얼라이저 수단(10)에서 출력된 4비트 데이터 ( $X_0$  또는  $X_1$ )를 순차적으로 저장한다.

저장 수단(12) 역시 상기 시리얼라이저 수단(10)과 마찬가지로  $2 \cdot f$ 의 주파수로 동기 된다.

배타논리합연산 수단(14)은 4개의 배타논리합 게이트들로 이루어져 있으며 상기 시리얼라이저 수단(10)의 출력 데이터와 저장 수단(12)의 출력데이터를 배타논리합 한다. 따라서 시리얼라이저 수단(10)의 출력 데이터가  $X_0$ :  $X_7^i X_5^i X_3^i$

$X_1^i$  이면 저장 수단(12)의 출력 데이터는  $X_1$ :  $X_6^{i-1} X_4^{i-1} X_2^{i-1} X_0^{i-1}$  가 되어 이 두 데이터의 배타논리연산 결과가

내로우 버스로 출력( $Z_3 Z_2 Z_1 Z_0$ )된다. 여기서 사용된 i 는 상기 8비트 입력데이터의 상대 시간을 나타낸다.

예로,  $X_7^i X_6^i \dots X_1^i X_0^i$  데이터는  $X_7^{i-1} X_6^{i-1} \dots X_1^{i-1} X_0^{i-1}$  데이터 바로 다음에 입력되는 값이다. 반

대로 시리얼라이저 수단(10)의 출력 데이터가  $X_1$ :  $X_6^i X_4^i X_2^i X_0^i$  이면 저장 수단(12)의 출력 데이터는  $X_0$ :

$X_7^i X_5^i X_3^i X_1^i$  가 되어 이 두 데이터의 배타논리연산 결과가 내로우 버스로 출력( $Z_3 Z_2 Z_1 Z_0$ ) 된다.

실시예에서는 입력 데이터의 비트 수를 8비트로 사용하였으나 상기 입력 데이터 비트 수가 2의 배수이면 제약 없이 사용이 가능하다.

도2는 본 발명의 실시예에 따른 디코딩 장치의 회로도이다.

본 발명의 실시예에 따른 인코딩된 4비트 데이터를 디코딩하는 장치는 도 2에 도시된바와 같이 배타논리합연산 수단(20), 저장 수단(22) 및 디시리얼라이저 수단(24)로 구성된다.

배타논리합연산 수단(20)은 4개의 배타적논리합 게이트들로 이루어져 있으며, 인코딩된 4비트 데이터( $Z_3 Z_2 Z_1 Z_0$ )와 저장 수단(22)에서 피드백(feedback)된 값, 즉 이전 배타논리합 결과 데이터를 배타논리합 한다.

저장 수단(22)은 4개의 플립플롭으로 이루어져 있으며, 배타논리합연산 수단(20)의 출력데이터를 2\*f 주파수에 동기 되어 순차적으로 저장한다.

디시리얼라이저 수단(24)은 4개의 1:2DEMUX(demultiplexor)가 사용되며 상기 DEMUX들은 1개의 Counter에 의해 동기 된다. 상기 Counter는 2\*f의 주파수로 동기 되어 0과1을 연속적으로 출력한다. 따라서 상기 Counter의 출력이 0일 때

상기 DEMUX를 통해 저장 수단(22)의 출력 값들이 전달되어  $X_7 X_5 X_3 X_1$  데이터가 우선 출력되고 1/(2\*f)시

간 후 상기 Counter의 출력이 1로 변화면 이때의 저장 수단(22)의 출력 값들이 다시 전달되어  $X_6 X_4 X_2 X_0$  데이터가 출력된다.

디시리얼라이저 수단(24)은 2\*f의 주파수로 동기된 Counter의 출력에 따라 DEMUX들을 통하여 2개의 4비트 저장 수단

(22) 출력데이터를 8비트의 원 데이터( $X_7 X_6 \dots X_1 X_0$ )로 복원한다.

이하에서, 다음과 같은 데이터 패턴을 예로 들어 본 발명의 동작을 설명한다.

0	00000101	2
2	10101111	4
4	10010100	5
6	01110101	4
8	00001010	7
	총 22 번의 천이	

상기 첫 번째 열은 시간을 두 번째 열은 8비트 입력데이터를 나타낸 것으로 f=0.5임을 알 수 있다. 마지막으로 세 번째 열은 천이 수를 나타낸다. 이 때 초기데이터는 00000000로 되어있다고 가정하고 구한 값이다.

상기 예를 본 발명에 의한 인코딩 방법에 적용하면 다음과 같이 인코딩 된다.

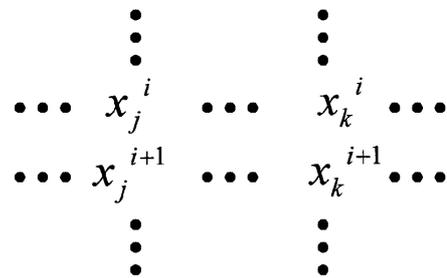
0	X0: 0000	0000	0
1	X1: 0011	0011	2
2	X0: 1111	1100	4
3	X1: 0011	1100	0
4	X0: 1000	1011	3
5	X1: 0110	1110	2
6	X0: 0100	0010	2
7	X1: 1111	1011	2
8	X0: 0011	1100	3
9	X1: 0000	0011	4
	총 22 번의 천이		

상기 첫 번째 열은 시간을 두 번째 열은 시리얼라이저 수단(10)의 출력을 나타낸다. 2\*f 주파수로 변화됨을 알 수 있다. 세 번째 열은 베타논리합연산 수단(14)의 출력 값으로 인코딩 장치의 최종 출력 값이기도 하다. 주목할 것은 네 번째 열로 총 천이 수가 상기 원 데이터의 총 천이 수와 동일하다. 이는 인접한 버스 와이어(wire)간의 커플링 커패시턴스(coupling capacitance)의 영향을 무시한다면, 다시 말해, 면적 커패시턴스(area capacitance)의 영향만 고려한다고 가정하면, 상기 원 8비트 데이터를 원 8 비트 버스 라인을 이용하여 전달할 때 버스에서 소모되는 전력(에너지)은 본 발명에 의한 인코딩 방법을 이용하여 4 비트 버스 라인을 이용하여 전달할 때 버스에서 소모되는 전력(에너지)과 같다는 것을 의미한다. 따라서 본 발명에 의한 인코딩 방법을 적용하면 성능 및 전력의 손해 없이 적은 버스 라인의 자원으로도 동일한 구현이 가능하다.

본 발명에 의한 디코딩 과정은 인코딩 과정의 역이다.

이하에서, 본 발명의 인코딩 방법을 이용할 때 상기 총 천이 수가 유지됨을 증명을 통해 보인다.

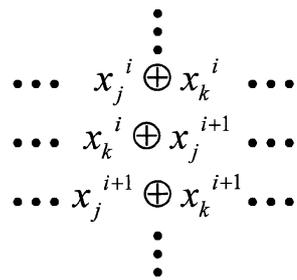
다음의 데이터는 원 데이터를 일반화하여 나타낸 것으로 i 는 상대시간을, j와 k는 임의의 비트 열을 나타낸다.



j와 k 열에 대해 연속된 비트의 천이 수를 계산하면 수학적 식 1과 같다.

$$\text{수학적 식 1} \\
 \text{천이 수} = (x_j^i \oplus x_j^{i+1}) + (x_k^i \oplus x_k^{i+1})$$

j와 k 열을 본 발명의 방법에 적용하여 일반화하여 나타내면 다음과 같다.



이에 대한 천이 수를 구하면 수학적 식 2가 되고 이는 수학적 식 1과 동일함을 알 수 있다.

$$\text{수학적 식 2} \\
 \text{천이 수} = \{(x_j^i \oplus x_k^i) \oplus (x_k^i \oplus x_j^{i+1})\} + \{(x_k^i \oplus x_j^{i+1}) \oplus (x_j^{i+1} \oplus x_k^{i+1})\} \\
 = (x_j^i \oplus x_j^{i+1}) + (x_k^i \oplus x_k^{i+1})$$

수학적 식 2를 살펴보면 처음 항에서 j열의 천이 수를 얻을 수 있는데 k열의 비트 데이터가 다리(bridge)역할을 함을 알 수 있다. 또한 XOR의 성질로 상기 k열의 비트 데이터가 사라짐을 알 수 있다. 두 번째 항도 이와 마찬가지로 k 열의 천이 수를 구하는데 j열의 비트 데이터가 다리역할을 하고 사라짐을 알 수 있다. 이는 두 열을 본 발명의 인코딩 방법을 이용하여 하

나의 열로 직렬화하면 각각의 열의 천이 수를 합친 것이 그 열의 천이 수가 됨을 의미한다. 또한 j와 k가 임의의 열이라 하였으므로 n 비트의 입력 데이터에 대해 천이 수는 동일하지만 서로 다른 형태의 n/2 비트 인코딩 데이터를 n!개 만들 수 있음을 의미한다.

본 발명의 인코딩 방법을 재귀적(recursive)으로 적용시키면, 다시 말해 인코딩된 데이터에 다시 인코딩을 가하면 원 데이터의 천이 수를 유지하며 데이터 비트 폭(width)을  $n/(2^m)$ 으로 줄일 수 있다. 여기서 m은 재귀적으로 인코딩을 가한 횟수이다. 일실시예로 원 데이터의 비트 폭이 8비트인 데이터 열을 이용한다.  $8 = 2^3$  이므로 재귀적 인코딩이 3번까지 가능하다.

m = 0	m = 1	m = 2	m = 3
00000101 2	0000 0	00 0	0 0
10101111 4	0011 2	00 0	0 0
10010100 5	1100 4	01 1	0 0
총 11 번	1100 0	00 1	0 0
	1011 3	11 2	0 0
	1110 2	00 2	1 1
	총 11 번	00 0	1 0
		00 0	0 1
		01 1	1 1
		10 2	0 1
		10 0	1 1
		01 2	0 1
		총 11 번	0 0
			0 0
			0 0
			0 0
			0 0
			1 1
			0 1
			1 1
			1 0
			1 0
			0 1
			1 1
			총 11 번

인코딩 횟수를 1,2,3번 늘려도 천이 수는 변하지 않음을 보여준다.

이상으로 첨부된 도면을 참조하여 본 발명의 실시예를 설명하였으나, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 그 기술적 사상이나 필수적인 특징을 변경하지 않고 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 따라서 이상에서 기술한 실시예는 모든 면에서 예시적인 것이며 한정적이 아닌 것이다.

### 발명의 효과

상기한 바와 같은 본 발명에 따른 천이 수를 유지하는 내로우 버스 인코딩/디코딩 방법은 원 방법과 비교해 동일한 성능과 전력 소모를 유지하면서 버스 라인의 수를 줄여 자원(resource) 절감의 효과 및 스큐, 와이어링의 어려움, 누화, 넓은 면적 등의 문제점들을 해결해 주는 효과가 있다.

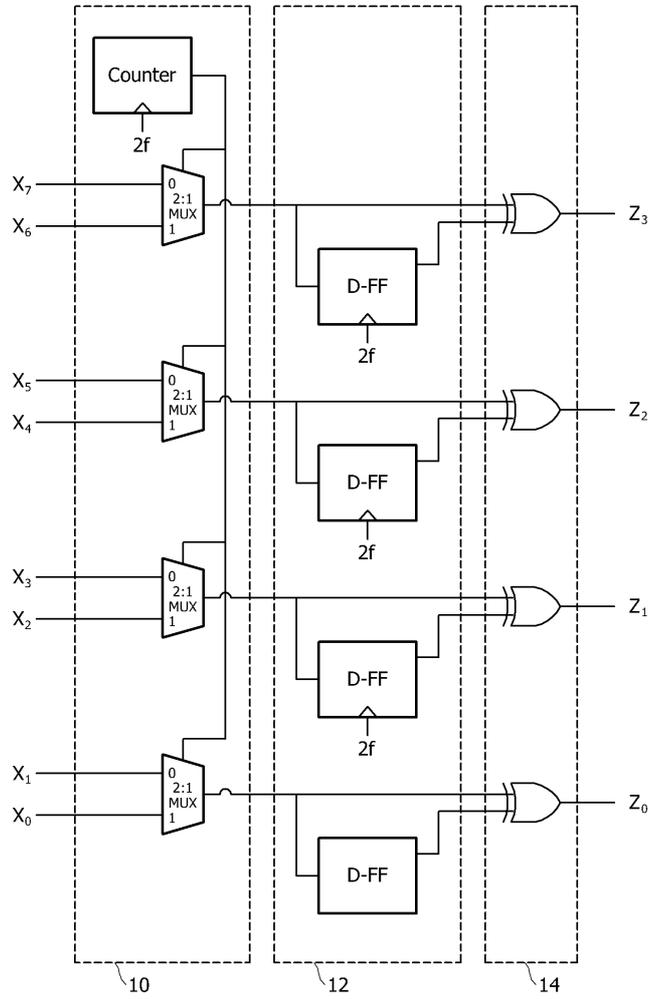
### 도면의 간단한 설명

도1은 본 발명의 실시예에 따른 인코딩 장치의 회로도이다.

도2는 본 발명의 실시예에 따른 디코딩 장치의 회로도이다.

도면

도면1



도면2

