

# 저전력 입출력을 위한 반복적인 버스반전 부호화

정덕기<sup>o</sup>, 손윤식, 정정화  
한양대학교.

서울시 성동구 행당동 산 17번지  
dolman@hymail.hanyang.ac.kr

## Recursive Bus-Invert Coding for Low-Power I/O

Duk-Gi Jung<sup>o</sup>, Yoon-Sik Son, Jong-Wha Chong  
Hanyang University  
HaengDang-Dong, SeongDong-Gu, Seoul, Korea  
dolman@hymail.hanyang.ac.kr

### Abstract

In this paper, we propose the bus coding technique for low power consumption. For CMOS circuit most power is dissipated as dynamic power for charging and discharging node capacitances. Though the I/O and bus are likely to have the very large capacitances associated with them and dissipate much of the power dissipated by an IC, they have little been the special target for power reduction. The conventional Bus-Invert coding method can't decrease the peak power dissipation by 50% because the additional invert signal line can invoke a transition at the time when Bus-Invert coding isn't used to code original bus data. The proposed technique always constraints the Hamming distance between data transferred sequentially to be below the half of the bus width, and thus decrease the I/O peak power dissipation and the I/O average power dissipation.

### I. 서론

최근 들어서 이동전화나 노트북 컴퓨터와 같은 휴대용 제품이 보편화되면서 시스템이 안정되게 동작하는

지속시간은, 제품의 무게 및 크기와 더불어 시장에서의 성패를 좌우하는 중요한 요인이 되었다. 그러나 대부분의 시스템에서 사용하는 충전식 배터리의 성능은 지난 30년 동안 2배에서 4배정도의 향상밖에 보이지 못하고 있다[9]. 따라서 회로의 기능은 그대로 유지하면서 전력소모는 줄이기 위한 저전력 설계(low-power design)가 지난 10여 년 동안 활발히 연구 되었다.

CMOS회로는 TTL, ECL 등의 회로에 비해 전력소모가 상당히 적다. 이러한 이유로 CMOS공정은 저전력 회로를 구현하기 위한 주요한 공정기술이 되었다. CMOS회로에서 전력소모는 정적전력소모와 동적전력소모로 나눌 수 있다[1]. CMOS회로에서 정적전력소모는 누설전류에 의한 전력소모를 의미하며 수나노와트( $10^{-9}W$ )/소자 정도이다. 동적전력소모는 두 가지 요인으로 나눌 수 있다. 첫 번째로 '0'에서 '1'로 또는 '1'에서 '0'으로 논리적 동작을 하는 동안 pull-up과 pull-down이 동시에 동작하여 짧은 시간동안  $V_{dd}$ 에서  $GND$ 까지 경로가 형성되므로 흐르게 되는 단락회로전류에 의한 전력소모요인이다. 두 번째로 각 게이트의 출력단의 부하 커패시턴스에 전하의 충전과 방전이 반복되면서 소모되는 전력소모요인이다. 후자에 전력소모가 회로 전체 전력소모의 약 80~90%를 차지한다.

CMOS회로의 동적전력소모는

$$P_{dynamic\ total} = \sum_{all\ gates} C \cdot V_{dd}^2 \cdot f \cdot P_t \quad (1)$$

이다[1][2]. 여기서  $C$ 는 각 부하의 정전용량,  $V_{dd}$ 는 전원전압,  $f$ 는 클럭주파수이다. 그리고  $P_t$ 는 동작계수로 클럭당 데이터천이의 평균값이다. 본 논문에서는 전원전압은 최적화 되어 있다고 가정한다. 또한 각 부하의 정전용량은 회로의 설계 및 레이아웃에 의하여 영향을 미치게되며 또한 최적화 되어 있다고 가정한다. 전체 회로를 구성하는 회로블록들간에 데이터 폭이  $n$ 인 버스를 통한 연속적인 데이터 전송이 발생할 때 이전에 전송된 데이터와 현재 전송하려는 데이터의 Hamming 거리[8]의 최대 값은  $n$ 이다. 이것은 모든 버스라인에서 천이가 발생함을 의미하며 이때 소모되는 전력이 버스에서의 순간 최대 전력소모 값이다. 버스를 통하여 데이터를 전송할 때 단위 시간(클럭)당 천이개수가  $n/2$ 를 넘지 않도록 하기 위하여 버스반전 부호화방법이 제안되었다[3]. 그러나 버스반전 부호화방법을 적용하면 부가된 invert 신호선을 포함하여 단위 시간당 천이개수가  $n/2$ 를 넘어서는 경우가 발생한다. 본 논문은 단위 시간당 천이 개수가  $n/2$ 을 넘지 않는 반복적인 부호화 방법을 제안한다. 제안하는 방법은 기존의 버스반전 부호화 방법이 가지는 하드웨어 오버헤드와 동일한 오버헤드로 구현가능하며, 실험을 통하여 제안한 방법의 효용성을 입증하도록 한다.

## II. 저전력 입출력을 위한 버스반전 부호화

일반적으로 입출력부의 정전용량은 칩 내부회로의 정전용량의 수백배이다[3]. 수식(1)에서와 같이 동적전력소모는 정전용량의 크기에 비례하므로 입출력회로에서 몇 번의 천이 회수를 감소시킴으로서 내부회로의 수백 내지 수천 번의 천이에 의해 소모되는 만큼의 전력을 줄일 수 있다. 입출력 회로의 동적전력소모는 일반적으로 전체전력소모의 약 50%정도이다[4][5][6][7]. 버스반전 부호화를 적용하기 위하여 칩 내부에 부호화를 위한 회로가 부가되어 칩 내부의 천이회수가 증가( $O(n \log n)$ [3])하더라도 데이터의 부호화로 입출력부의 천이회수가 감소되므로 전체 전력소모는 줄어들게 된다. 다음의 예를 살펴해보도록 하자.

만일  $V_{dd}=1V$ ,  $f=1MHz$ ,  $C_{av I/O}=1000$ ,  $P_{t I/O}=10$ ,  $C_{av in}=10$ 이고  $P_{t in}=1000$ 이라고 가정하고 부호화이전의 동적전력소모량을 계산하면

$$P_{uncoded} = P_{dynamic in} + P_{dynamic I/O} = 20,000$$

이다. 부호화에 의하여 입출력부의 동작계수가 부호화이전의 절반인  $P_{t I/O}=5$ 로 감소한다면, 내부 회로부의 동작계수가  $P_{t in}$ 은  $35 (\approx 10 \log 10)$ 만큼 증가하여 1035가 된다. 부호화이후의 동적전력 소모량을 계산하면

$$P_{coded} = 1000 \cdot 5 + 10 \cdot 1035 = 15,350 \approx (3/4)P_{uncoded}$$

가 되어 약 25%의 소모전력 감소효과를 볼 수 있다. 저전력을 위한 입출력 버스의 부호화 방법인 버스반전 부호화는 데이터 폭이  $n$ 비트 클럭동기 데이터 버스와 부가의 1비트 신호인 'invert'를 사용한다.  $n$ 비트 버스를 위한 버스반전 부호화를 설명하기 위하여 사용되는 기호들을 다음과 같이 정의한다.

$O(i)$ :  $i$ 번째 원형 데이터( $n$ 비트)

$C(i)$ :  $i$ 번째 반전부호화된 데이터( $n$ 비트)

$R(i)$ :  $i$ 번째 반복적인 반전부호화된 데이터( $n$ 비트)

$C'(i)$ :  $i$ 번째 invert신호가 포함된 반전부호화된 데이터( $n+1$ 비트)

$R'(i)$ :  $i$ 번째 invert신호가 포함된 반복적인 반전부호화된 데이터( $n+1$ 비트)

정의1:  $H[s, t]$ 는  $n$ 비트 데이터인  $s$ 와  $t$ 사이의 Hamming거리이다.

정의2:  $P(i) \downarrow$ 는 데이터 $P(i)$ 가 반전된 데이터이다.

버스반전 부호화 방법은 다음과 같이 기술될 수 있다.  $H[C(i), O(i+1)] \leq n/2$  이면  $C(i+1) = O(i+1)$ , invert = '0'이 되며,  $H[C(i), O(i+1)] > n/2$  이면  $C(i+1) = O(i+1) \downarrow$ , invert = '1'이 된다.

이 방법에 의하면 시간적으로 연속되는 두 데이터간의 Hamming거리는 항상  $(n/2)+1$ 이하가 된다.

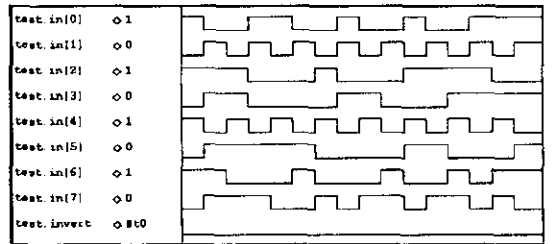


그림 1. 전형적인 8비트 데이터 버스

그림 1은 8비트 데이터 버스상의 16단위시간대 동안의 연속적으로 변하는 데이터로 천이 개수는 71개이다.

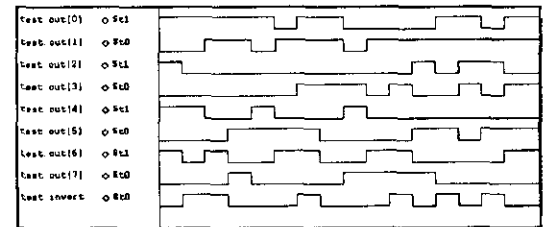


그림 2. 버스반전 부호화된 데이터 버스

그림 2는 그림 1의 데이터에 버스반전 부호화 방법을 적용하여 부호화된 데이터로 천이 개수는 invert신호를 포함하여 55개로 줄었다.

버스반전 부호화 방법에 의하여 한 단위시간대와 다음 단위시간대 사이에 Hamming거리는  $n/2$ 이하가 된다. 그러나 새로이 추가된 신호인 invert신호에 의하여 최대 Hamming거리는  $(n/2)+1$ 이하가 된다.

### III. 반복적인 버스반전 부호화

본 논문에서 제안하는 반복적인 버스반전 부호화는 기존의 방법인 버스반전 부호화에 부가의 반전신호버스를 사용하지 않고 반복적으로 버스반전 부호화를 적용하여 데이터간의 Hamming거리를  $n/2$ 이하로 줄이는 방법이다. 먼저 연속적인 버스반전 부호화마다 데이터의 반전을 나타내기 위한 각각의 invert신호를 갖는 반복적인 버스반전 부호화에 대하여 논의한다.

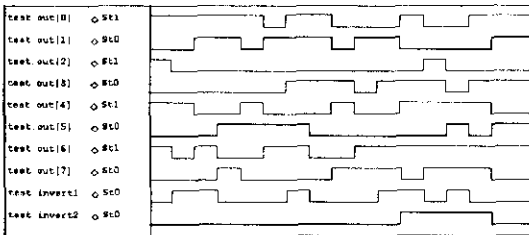


그림 3. 두 개의 반전신호가 있는 반복적인 버스반전 부호화된 데이터 버스

그림 3은 반복적인 버스반전 부호화 방법을 적용하여 두 번째의 반전신호인 invert2신호를 포함하고 있는 부호화된 데이터를 나타내고 있으며 데이터의 천이 개수는 55개로 버스반전 부호화에 의한 데이터 천이 개수와 같다. 이와 같이 두 번째 반전신호의 데이터 천이로 반전부호화에서 발생하는 데이터의 천이 개수보다 반복적인 반전부호화에서 발생하는 데이터의 천이 개수가 증가할 수 있다는 문제점이 있다.

표 1. 반복적인 버스반전 부호화에서의 반전신호

반전신호		내 용	invert
invert1	invert2		
0	0	한 번도 반전이 일어나지 않음	0
0	1	두 번 모두 반전이 일어남	0
1	0	첫 번째 한 번만 반전이 일어남	1
1	1	두 번째 한 번만 반전이 일어남	1

표 1에서는 두 개의 반전신호인 invert1, invert2는 새로운 반전신호인 invert로 대체할 수 있으며, 이 invert신호는 두 번의 부호화를 수행한 invert1신호와 동일한 신호임을 알 수 있다. 따라서 첫 번째 반전부호화의 반전신호인 invert1을 포함한  $n+1$ 개의 데이터에 대하여 버스반전 부호화를 적용하여 반전신호의 증가로 발생하는 문제점이 해결될 수 있다.

[정리1]  $H[R'(i), R'(i+1)]$ 은 반복적인 버스반전 부호화 방법에 의해 항상  $n/2$ 이하로 억제된다.

증명 : 첫 번째 버스반전 부호화에 의하여 시간적으로 연속인  $n$ 비트 데이터의 최대 Hamming거리는  $n/2$ 이다. 여기에 1비트의 invert신호가 첨가되어  $n+1$ 비트 데이터의 최대 Hamming거리는  $(n/2)+1$ 이 된다. 두 번째 버스반전 부호화에 의하여 invert신호를 포함하는  $n+1$ 비트 데이터에 대하여 다시 버스반전 부호화를 적용한다. 반복적인 버스반전 부호화의 과정 중 두 번째 반전부호화에서 반전이 발생하는 경우는 다음의 두 경우이다. 첫 번째는  $H[C(i-1), O(i)] > n/2$ ,  $H[R(i-1), C(i)] \leq n/2$ ,  $H[C(i), O(i+1)] \leq n/2$ ,  $H[R(i), C(i+1)] = n/2$ 인 경우로  $H[R'(i), C'(i+1)] = (n/2)+1$ 이 되어 두 번째 반전이 발생한다. 두 번째는  $i$  단위시간대에 두 번째 반전부호화 과정에서 반전이 발생하여  $R'(i) = C'(i) \downarrow$ 이고  $H[C(i), O(i+1)] > n/2$ ,  $C(i+1) = O(i+1) \downarrow$ ,  $H[R(i), C(i+1)] > n/2$ 이 되어 반전이 발생한다. 따라서 두 번째 반전에 의해서 최대 Hamming거리는  $n/2$ 이하가 된다.

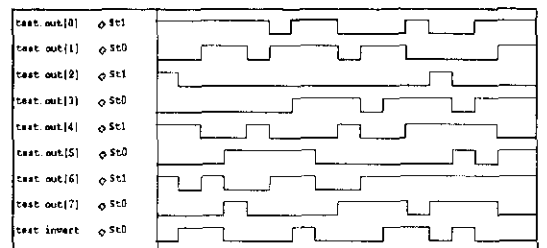


그림 4. 반복적인 버스반전 부호화된 데이터 버스

그림 4는 본 논문에서 제안한 방법인 하나의 반전신호를 이용하여 반복적으로 버스반전 부호화를 실행한 결과 부호화된 데이터 버스의 상태이다. 제안된 방법에 의한 부호화된 후의 데이터의 천이 개수는 invert신호를 포함하여 53개로 줄어든 것을 확인할 수 있다.

#### IV. 실험 및 고찰

제안된 반복적인 버스반전 부호화 방법 및 기존의 방법인 버스반전 부호화 방법을 적용하여 화상, 음성 및 무작위 데이터에 대한 버스부호화를 수행하였다.

표 2. 각종 데이터의 천이 개수

부호화방법 \ 데이터형식	화상	음성	무작위
원형	4,398	4,492	4,388
버스반전 부호화	4,110	3,812	3,634
제안된 부호화	4,076	3,705	3,526

버스반전 부호화 및 제안된 부호화에 의한 부호화 이후의 데이터의 천이개수는 표 2와 같다. 표 2에 의하면 화상데이터의 한 데이터와 다음 데이터간의 상관관계가 음성이나 무작위 데이터보다 크므로 두 데이터간의 Hamming 거리가 음성이나 무작위 데이터보다 일반적으로 가깝다. 따라서 코딩으로 인하여 천이의 개수 감소가 적다.

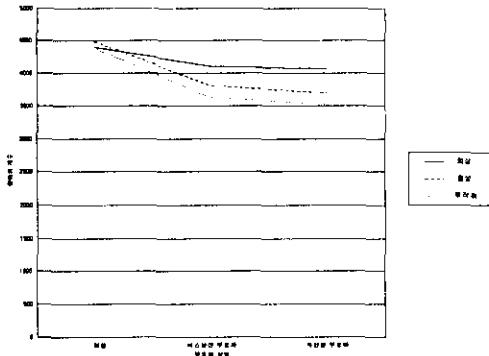


그림 5. 각종 데이터에 대한 부호화 실험결과

그림 5에 의하면 데이터간의 상관관계가 적을수록 부호화의 효율이 높다는 것을 알 수 있다.

#### V. 결론

본 논문에서는 저전력 시스템을 구현하기 위하여 입출력부에서의 동적전력 소모를 줄이는 방법중의 하나인 반복적인 버스코딩 부호화 방식을 제안하였다. 제안된 방식에 의한 버스코딩을 수행할 경우 데이터 버스의 시간적으로 연속되는 두 데이터간의 Hamming

거리는  $n/2$ 이하로 억제된다. 제안된 방법은 한 칩 내부의 모듈간, 칩, 패키지, PCB 수준 등의 대부분의 전자 회로에 적용이 가능하다. 따라서 낮은 하드웨어 오버헤드로 높은 효율의 저전력 시스템의 구현에 적합한 방식으로 앞으로 많은 연구가 수행되어 질 것으로 예상된다.

#### 참고문헌

- [1] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design, A Systems Perspective*. Reading, MA: Addison-Wesley Publishing Company, 1988.
- [2] A. P. Chandrakasan, S. Sheng and R. W. Broderson, "Low-Power CMOS Digital Design," *IEEE Journal of Solid-State Circuits*, pp. 473-484, April 1992.
- [3] M. R. Stan and W. P. Burleson, "Bus-Invert Coding for Low-Power I/O," *IEEE Trans. of VLSI Systems*, pp. 49-58, March 1995.
- [4] D. Dobberpuhl *et al.*, "A 200-MHz 64-bit dual-issue CMOS microprocessor," in *IEEE Custom Integrated Circ. Conf.*, 1990, pp. 19.7.1-19.7.6.
- [5] C. A. Neugebauer and R. O. Carlson, "Comparison of wafer scale intergration with VLSI packaging approaches," *IEEE Trans. Components, Hybrids, and Manufact. Technol.*, pp. 184-189, June 1987.
- [6] W. C. Tan and T. H. Y. Meng, "Low-power polygon renderer for computer graphics," *Int. Conf. on A.S.A.P.*, pp. 200-213, 1993.
- [7] R. Wilson, "Low power and paradox," *Electronic Engineering Times*, pp. 38, Nov. 1, 1993.
- [8] T. R. N. Rao and E. Fujiwara, *Error-Control Coding for Computer Systems*, Prentice Hall, 1989.
- [9] J. M. Rabaet and M. Pedram, *Low Power Design Methodologies*. Kluwer Academic Publishers, 1996.