

논문 2005-42SD-4-5

# 고속 플래시 AD 변환기를 위한 Successive Selection Encoder의 Logical Effort에 의한 설계

(Design of the Successive Selection Encoder by the Logical Effort for High Flash Speed ADC's)

이 기 준\*, Kyusun Choi\*\*, 김 병 수\*\*\*

(Kijun Lee, Kyusun Choi, and Byung-soo Kim)

## 요 약

고속 flash ADC를 위하여, Successive Selection Encoder (SSE)라고 명명된 새로운 형태의 TC-to-BC encoder를 제안한다. 기존의 fat tree encoder가 OR 논리에 의하여 동작되는데 반하여, 제안된 SSE는 MUX 논리에 의하여 입력 TC 신호들 중에서 직접 출력 BC 신호를 선택한다. 제안한 SSE의 구현을 위하여, Logical Effort 방법과 Hynix 0.25um 제조 공정에 의한 실험을 바탕으로 효율적인 SSE의 구현 구조를 정하였다. 이론적 모델과 실험 결과를 보면, SSE가 fat tree encoder에 비하여 (1) one-out-of-n 신호를 발생시킬 필요가 없고, (2) 사용되는 게이트 수는 약 1/3로 감소하며, (3) 동작속도는 2배 이상 빨라진다. 제안된 SSE는 고속 ADC에 적합한 TC-to-BC encoder로 사용될 수 있다.

## Abstract

In this paper, a new type of the TC-to-BC encoder for high speed flash ADC's, called the Successive Selection Encoder (SSE), is proposed. In contrast to the conventional fat tree encoder based on OR operations, the BC outputs, in the new design, are obtained directly from TC inputs through simple MUX operations. The detailed structure of the SSE has been determined systematically by the method of the logical effort and the simulation on Hynix 0.25um process. The theoretical and experimental results show that (1) it is not required to generate one-out-of-n signals, (2) the number of gates is reduced by the factor of 1/3, and (3) the speed is improved more than 2-times, compared to the fat tree encoder. It is speculated that the SSE proposed in this study is an effective solution for bottleneck problems in high speed ADCs.

**Keywords:** Flash ADC, TC-to-BC encoder, Fat tree encoder, Successive selection encoder

## I. 서 론

능동회로와 논리회로를 함께 하나의 집적회로로 설계하는 혼합회로 (mixed circuit)와 SOC (System on Chip)의 설계와 제작에 있어, 고속 ADC (Analog-to-Digital Converter)는 기본 부품으로 매우 중요하다.

오늘날 널리 사용되고 있는 여러 종류의 ADC 구조들 중에서 flash ADC는 동작 속도의 관점에서 제일 우수한 구조로 알려져 있다. 그림 1은 3-bit flash ADC의 기본 구조를 보여준다. Flash ADC는 크게 나누어 analog comparator와 digital encoder의 두 부분으로 구성된다. 일반적으로 N-bit flash ADC의 경우, 외부에서 analog 입력이 인가되면 analog comparator 부분은 ( $2^N - 1$ ) 개의 comparator들에 의하여 TC (Thermometer Code) 신호를 발생하며, digital encoder 부분은 TC 신호를 BC (Binary Code) 신호로 변환한다. 기존의 TC-to-BC encoder 방식으로는 ROM/PLA encoder<sup>[10]</sup>, Wallace tree encoder<sup>[5]</sup>, XOR encoder<sup>[6]</sup>, pipeline

\* 정회원, 충남대학교 전기정보통신공학부  
(Division of Electrical and Computer Engineering,  
Chungnam Nat'l University)

\*\* Dept. of Computer Science and Engineering,  
Pennsylvania State University

\*\*\* 정회원, 삼성전자  
(Samsung Electronics)

접수일자: 2004년9월2일, 수정완료일: 2005년3월29일

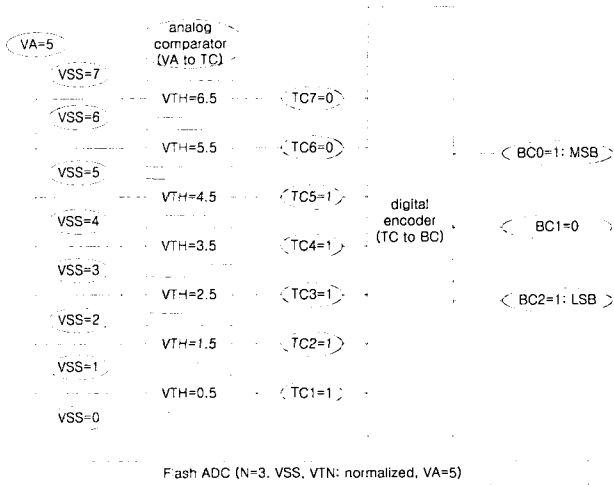


그림 1. 3-bit flash ADC (비교기의 문턱전압 VTH와 양자화 전압 수준 VSS는 정규화 되었고, analog 입력 VA=5가 인가 된 경우)  
Fig. 1. 3-bit flash ADC.

encoder<sup>[7]</sup>, fat tree encoder<sup>[11]</sup> 등이 있다. 이 중에서 fat tree encoder가 동작 속도와 전력 소모의 관점에서 제일 좋은 encoding 방식으로 알려져 있다.

이 논문은 Successive Selection Encoder (SSE)라고 명명된 새로운 방식의 TC-to-BC encoding algorithm을 제안한다. 기존의 fat tree encoder는 OR 논리에 의하여 TC-to-BC encoding을 수행하는데 반하여, 제안한 SSE 방식은 MUX 논리에 의하여 TC 신호들 중에서 적합한 출력 BC 신호를 직접 선택한다. II장에서는 SSE의 encoding algorithm을 설명한다. Forward SSE를 제안한 참고문헌 [13]과 비교하여 볼 때, 이 논문에서 제안하는 backward SSE는 TC 신호를 직접 제어 신호로 사용하며 제어 신호에 의한 MUX 부하를 줄임으로써 동작 속도의 향상을 기대한다. III장에서는 logical effort에 의한 SSE의 구조 설계 및 상세 설계를 언급하며, IV장에서는 시뮬레이션 결과를 보여주고, 끝으로 V장에서 결론을 맺는다.

## II. Successive Selection Encoder

앞에서 설명하였듯이, N-bit Flash ADC의 comparator 부분에서는  $(2^N - 1)$  개의 TC 신호를 발생한다. 그림 1에서 정하는 바와 같이, 가장 적은 문턱 전압에 의한 TC 신호를 1번으로 하여 차례대로 번호를 정하고 T(k)를 k 번째 TC 신호라 하자. TC 신호의 중요한 특징 중에 하나는 만약  $T(k) = "1"$  (논리 1)이면, k 보다 적은 번호의 모든 TC 신호들도 논리 1의 신호 값을 가진다. 즉,

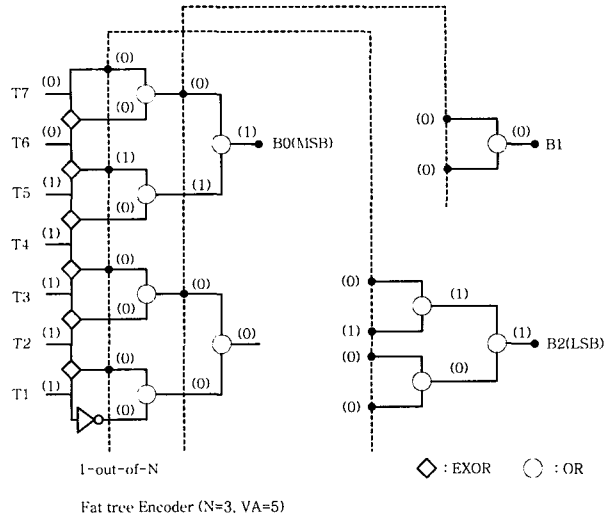


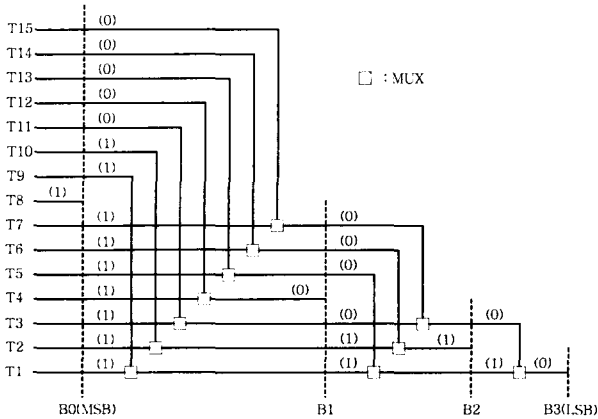
그림 2. 3-bit fat tree encoder  
Fig. 2. 3-bit fat tree encoder.

TC 신호는 온도계의 수온이 온도를 표시하는 것과 같은 방법으로 analog 입력 신호의 크기를 표시한다. 다음의 설명에서 N 개의 출력 BC 신호를  $\{B(0), B(1), \dots, B(N-1)\}$ 로 표시하며, B(0)는 MSB를, B(N-1)은 LSB를 의미한다.

### 1. Fat Tree Encoder

그림 2는 3-bit fat tree encoder의 동작 원리를 설명한다. 입력으로 TC 신호가 주어지면, 우선 EXOR 논리에 의하여  $2^N$  개의 one-out-of-n 신호들을 발생한다. One-out-of-n 신호란 총  $2^N$  개의 신호들 중에서 단 하나만 논리 1의 값을 가지며 나머지  $(2^N - 1)$  개의 신호들은 논리 0의 값을 가지는 것을 의미한다. 이 때, 논리 1의 값을 가지는 신호는 TC 신호가 논리 1에서 논리 0으로 바뀌는 위치를 표시하며, 이는 analog 입력 신호의 크기에 해당된다. 따라서 논리 신호  $(0, 0, \dots, 0)$ 에 해당되는 맨 아래 one-out-of-n 신호를 0번으로 하여 차례로 번호를 정하면,  $\{1, 3, 5, \dots\}$  번째의 one-out-of-n 신호들 중에서 논리 신호 1이 존재하면 B(N-1)의 신호 값은 논리 1이 되고, 아니면 논리 0가 된다. 마찬가지로,  $\{(2,3), (6,7), (10,11), \dots\}$  번째의 one-out-of-n 신호들을 조사하면 B(N-2)의 논리 신호 값을,  $\{(4,5,6,7), (12,13,14,15), \dots\}$  번째의 one-out-of-n 신호들을 조사하면 B(N-3)의 논리 신호 값을 알 수 있다. 이상의 방법으로 N 개의 BC 신호 값들을 모두 정할 수 있다.

그림 2에서 보듯이, fat tree encoder에서는 one-out-of-n 신호들의 적합한 부분 집합에서 논리 신호 1이



Forward Successive Selection Encoder (N=4, V<sub>A</sub>=10)

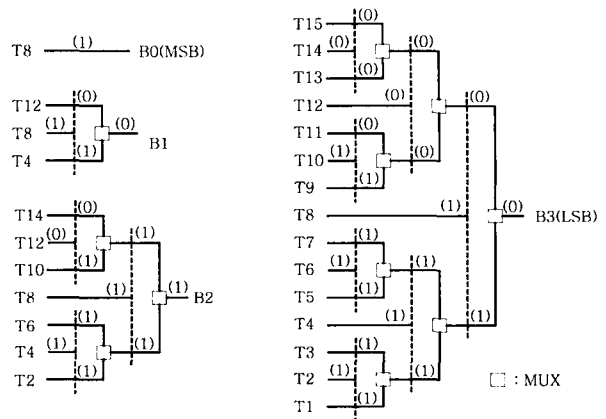
그림 3. 4-bit forward successive selection encoder  
Fig. 3. 4-bit forward successive selection encoder.

있는지 없는지를 판단하기 위하여 binary tree 구조의 연속적인 (N-1) 단계의 two-input OR 논리를 수행한다. N 개의 BC신호를 추출하기 위하여서는 N 개의 OR tree가 필요하다. 결국, Fat tree encoder는 one-out-of-n 신호 발생 단계를 포함하여 총 N 단계가 필요하며,  $(2^N-2)$  개의 EXOR 소자와  $(2^{N+1}-N-3)$  개의 two-input OR 소자가 필요하다.

### 2. Forward Successive Selection Encoder

TC 신호 자체도 analog 입력 신호의 크기 정보를 가지고 있다. 이를 이용하면 TC 신호에서 직접 출력 BC 신호를 구할 수 있다. 예로서, N-bit Flash ADC에서  $T(2^{N-1})$ 는 바로 B(0) (즉, MSB) 신호를 의미한다. 만약 B(0)="1" (논리 1)이면  $T(2^{N-1}+2^{N-2})$ 가, B(0)="0" (논리 0)이면  $T(2^{N-1}-2^{N-2})$ 가 B(1) 신호를 의미한다. 따라서, B(1) 신호는  $T(2^{N-1}+2^{N-2})$ 와  $T(2^{N-1}-2^{N-2})$  중에서 B(0) 신호 값에 따라 선택된다. 이상의 선택 과정을 계속 하면 모든 출력 BC 신호를 입력 TC 신호에서 직접 구할 수 있다.

설명 편의를 위하여 4-bit flash ADC의 경우를 생각하자. 그림 3은 4-bit forward SSE의 동작을 보여준다. 첫 번째 단계는 총 15 개의 TC 신호들 중에서 가운데에 해당되는 T(8)이 바로 B(0)이고, B(0) 값에 따라 위 부분 {T(9),...,T(15)}가 선택되거나 아래 부분 {T(1),...,T(7)}이 선택된다. 두 번째 단계에서는 선택된 7 개의 신호들 중에서 가운데에 해당되는 4 번째 신호가 B(1)에 해당되고, B(1)에 의하여 위 부분 또는 아래 부분의 3 개의 신호가 선택된다. 세 번째 단계에서는 선택된 3 개의 신호들 중에서 가운데 신호가



Backward Successive Selection Encoder (N=4, V<sub>A</sub>=10)

그림 4. 4-bit backward successive selection encoder  
Fig. 4. 4-bit backward successive selection encoder.

B(2)로 되고 B(2)에 의하여 위 또는 아래의 신호가 B(3)으로 선택된다. 이상의 forward SSE 방법은 쉽게 일반화 할 수 있다.

이 방법은 Successive Approximation ADC의 방식과 유사하며, 선택을 위하여 MUX 논리가 사용된다.

Forward SSE는 one-out-of-n 신호의 발생이 필요하지 않으며, 총  $(2^N-N-1)$ 개의 two-input MUX가 사용되고, (N-1) 단계를 거친다. 이는 fat tree encoder에 비하여, 계산 단계가 하나 줄어들었으며 게이트 수가 약 1/3로 감소한다. 그러나, Forward SSE는 다음의 두 가지 단점을 가지고 있다. (1) B(k) 신호 값을 결정하기 위하여 {B(0),..., B(k-1)}의 신호 값들이 미리 결정되어야 한다. (2) MUX의 제어 신호들은 과도한 MUX 부하를 가진다. 예로서, B(0)에 해당되는  $T(2^{N-1})$ 는  $(2^{N-1}-1)$  개의 MUX를 제어하여야 한다. 일반적으로, B(k) 신호에 해당되는 TC 신호는  $(2^{N-k-1}-1)$  개의 MUX를 제어한다. 이러한 단점들로 인하여 forward SSE의 동작 속도 향상을 기대할 수 없다.

### 3. Backward Successive Selection Encoder

Forward SSE인 그림 3에서 B(2)를 선택하는 과정을 다시 살펴보자. 설명 편의를 위하여  $mux(U,L,C)$ 를 제어 신호 C가 논리 1이면 U가 선택되고, 논리 0이면 L이 선택되는 MUX 논리라 하자. 그러면,  $B(0)=T(8)$ ,  $B(1)=mux(T(12),T(4),B(0))$  이고,

$$B(2)=mux(mux(T(14),T(6),B(0)), mux(T(10),T(2),B(0)), B(1)) \quad (1)$$

이 된다. 만약,  $B(0)=1$  이면,  $B(1)=T(12)$ ,  $B(2)=F=mux(T(14),T(10),T(12))$ 가 되며,  $B(0)=0$  이면,  $B(1)=T(4)$ ,  $B(2)=G=mux(T(6),T(2),T(4))$ 가 된다. 여기서, F와 G는  $B(2)$ 가 될 수 있는 후보 신호 들이다. 선택 순서를 거꾸로 하여, 후보 신호 F, G 들을 미리 결정하고  $B(0)$  값에 따라  $B(2)$ 를 결정하는 방법으로 정리하면 그 수식은 다음과 같다.

$$B(2)=mux(mux(T(14),T(10),TC(12)), mux(T(6),T(2),T(4)), T(8)) \quad (2)$$

$B(2)$ 를 선택하는 두 개의 수식을 비교하면, forward SSE인 수식 (1)에서는  $B(0)=T(8)$ 이 두 개의 MUX를 제어하며  $B(1)$ 이 결정되어야만 최종 선택 결과,  $B(2)$ 를 얻을 수 있다. 그러나, backward SSE인 수식 (2)에서는  $B(0)=T(8)$ 은 한 개의 MUX 만을 제어하며, 모든 제어 신호 들을 TC 신호에서 직접 구할 수 있다. 이상의 방법에 의하여, 그림 3의 forward SSE 구조는 그림 4의 backward SSE 구조로 바뀌어 진다. Backward SSE를 일반화하면, N-bit flash ADC에서 출력 BC 신호들을 선택하는 과정은 다음과 같다.

Alg. Backward Successive Selection.

```

begin
  stage 0:  $G(0,r)=T(r)$  for  $r=1,\dots,(2^N-1)$ ;
  for  $k=1,\dots,(N-1)$  {
    for  $s=1,\dots,k$ , {  $p=N-1-k+s$ ;  $q=2^{k-s}-1$ ;
      for  $r=0,\dots,q$ , {
         $c=2^p+r*2^{p+1}$ ;  $upp=c+2^{p+1}$ ;  $low=c-2^{p+1}$ ;
         $G(s,c)=mux(G(s-1,upp),G(s-1,low),T(c))$ ;
      }
    }
  }
  output BC:  $B(k)=G(k,2^{N-k})$  for  $k=0,\dots,(N-1)$ ;
end

```

Backward SSE는 forward SSE와 같은 계산 단계와 같은 MUX의 수가 사용되지만, (1) MUX의 제어 신호는 TC신호를 직접 사용하며, (2) 한 TC 신호가 제어하는 MUX의 수는 최대 (N-1)이다.

일반적으로,  $B(k)$ 와 관련되는 TC 신호들의 모임을  $A(k)=\{T(j): j=2^{N-k-1}+r*2^{N-k}, r=0,1,2,\dots,(2^k-1)\}$ 라 하면,  $A(k)$ 의 TC 신호 들이 각각 제어 하는 MUX의 개수는 (N-k-1)이고, k 단계의 two-input MUX 선택 작업을 거쳐  $A(k)$ 의 TC 신호 들 중에서  $B(k)$ 가 선택된다.

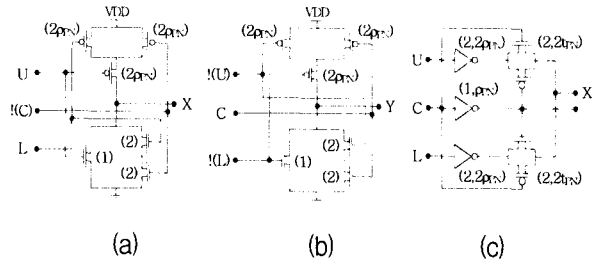


그림 5. MUX의 구현 (a) TCMUX1 (b) TCMUX2 (c) Transmission gate를 이용한 MUX (괄호 안의 숫자는 트랜지스터의 상대적인 채널 폭 크기.  $p_{PN}$  : PN ratio of the inverter,  $t_{PN}$  : PN ratio of the transmission gate)

Fig. 5. Implementation of the MUX. (a) TCMUX1. (b) TCMUX2. (c) Transmission gate-MUX.

표 1. Logical effort에 의한 두 종류 MUX의 비교  
Table 1. Comparison of MUX's by the logical effort.

MUX	#TR	size	g(signal)	g(control)	$p*(1+p_{PN})$
TC	6	$5+6p_{PN}$	2	2	$3+2p_{PN}$
TFER	8	$8+4p_{PN}+4t_{PN}$	2	$4(1+t_{PN})/(1+p_{PN})$	$4+4t_{PN}$

### III. SSE의 설계

Backward SSE (이하, SSE로 부름)를 설계하려면, 우선 기본 소자인  $mux(U,L,C)$ 를 구현하여야 한다. 이를 부울리안 대수로 표시하면,  $mux(U,L,C)=U*C+L*!(C)$ 로 표현된다. 여기서, !는 inverting 논리, +는 OR 논리, \*는 AND 논리를 의미한다. 그런데,  $T(k)=1$ 이면 k보다 적은 모든 r에 대하여  $T(r)=1$ 인 성질과 앞 장에서 설명한 backward SSE 알고리즘에서 U, L, C의 indexing 번호가 각각  $upp=(c+2^{p+1})$ ,  $low=(c-2^{p+1})$ , c인 점을 이용하면,  $TCMUX(U,L,C)=U*L*!(C)$ 로 간단히 할 수 있다. CMOS 논리는 기본적으로 inverting 논리이므로  $A=!(U)$ ,  $B=!(L)$ ,  $D=!(C)$ 라 할 때,  $TCMUX1(U,L,D)=!(U+L*D)$ 와  $TCMUX2(A,B,C)=!(B+A*C)$ 인 두 종류 static TCMUX를 구현하고 번갈아 사용한다 (그림 5(a)와 (b)).

MUX의 또 다른 구현으로 transmission gate를 이용하여 구현할 수 있는데, 이 경우 신호의 안정성을 위하여 입력 단 또는 출력 단에 buffer를 달아야한다 (그림 5(c)). 표 1은 TCMUX와 입력 단에 buffer로 인버터를 연결한 transmission gate-MUX의 성능을 logical effort 관점에서 비교한다. 이 비교표에서 올바른 비교를 위하여 제어 신호와 관련된 인버터는 포함하지 않는다. 표 1에 의하면 static TCMUX가 transmission gate-MUX에 비하여 더 효율적임을 알 수 있다.

두 종류의 TCMUX 들을 번갈아 배치함에 있어, 제 1 단에 TCMUX1을 배치하는 경우를 late mode control 이라 하고, 제 1 단에 TCMUX2를 배치하는 경우를 early mode control이라 하자 (그림 6).

SSE의 구현에 있어 두 번째로 고려하여야 할 사항은 MUX를 제어하는 제어 회로의 설계이다. 기본적으로 SSE에서는 TC신호를 직접 제어 신호로 사용할 수 있지만, 이 경우 TC 신호에 과도한 MUX 부하가 걸리고 동작 속도를 저해할 수 있다. 여러 가지 형태의 제어 회로를 생각할 수 있지만, 각각의 TCMUX마다 인버터를 병행하는 경우와 제어 신호로 사용되는 TC 신호마다 두 개의 인버터를 사용하는 경우를 실험한다.

그림 6은 TCMUX마다 인버터를 병행하는 경우의 3-bit SSE의 회로도이다. 입력 TC 신호들에서부터 출력 BC 신호까지의 경로는 MUX에 의하여 선택되는 신호 경로와 MUX의 제어 경로로 나누어진다. 또한, 출력 BC 신호들이 동일한 시간에 발생하도록 조절하기 위하여 출력 BC 신호가 선택된 이후에 추가로 지연회로로서 인버터 들을 직렬 연결한다. 그림 6(a)에서처럼 홀수의 수행 단계에 의하여 출력 신호가 반전되는 경우는 최종 단계 인버터 들을 연결한다. 결국, 각각의 신호 경로에서의 MUX와 지연회로를 합한 단계(stage) 수는 동일하다.

다음으로, 설계한 encoder의 내부 MOSFET 크기 (채널 폭)의 설계는 logical effort에 의한 크기 조절 방법을 사용한다<sup>[3]</sup>. 논리회로에서 하나의 경로(path) 안에 있는 모든 단위 소자들이 동일한 지연시간을 가질 때 그 경로가 동작 속도의 관점에서 최적화된다는 성질을 이용하여, 모든 단위 소자 들이 같은 logical effort delay를 가지도록 설계 한다 (equal delay sizing 방법). 즉, 모든 단위 소자 들이 동일한 logical effort delay,  $f$  를 가진다고 가정 한 후,  $f$ 를 변화시켜 가면서 전체 logical effort delay,  $F_{TOT}$ 가 가장 적게 되는  $f^*$ 를 찾은 다음,  $f^*$ 에 따라 transistor sizing을 한다. 이러한 방법으로 transistor sizing을 하였을 경우, 앞 단계의 소자 들은 최소 크기의 transistor로 설계되며, 뒤 단계에서 출력 부하를 고려한 transistor sizing이 이루어짐을 알 수 있다. 끝으로, 각 출력 BC 신호마다 동일한 동작 속도를 가지도록 추가로 지연 회로의 크기를 부분적으로 재 조정한다.

성능 비교를 위하여 fat tree encoder도 설계하였다. One-out-of-n 신호의 생성을 위한 EXOR 논리는 TCMUX의 설계에서처럼  $EXOR=U*!(L)+!(U)*L$  대

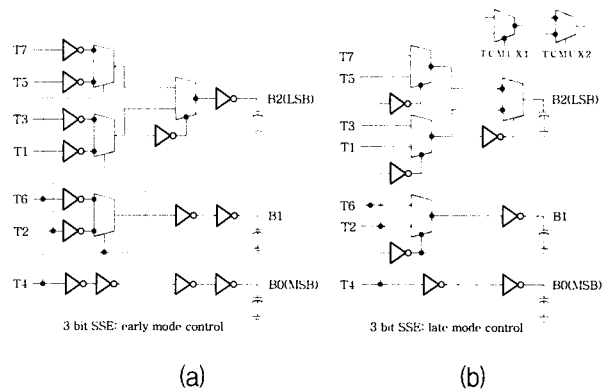


그림 6. 3-bit SSE (a) Early mode control (b) Late mode control (점선은 제어 경로)

Fig. 6. 3-bit SSE. (a) Early mode control. (b) Late mode control.

신에  $TCEXOR=!(U+!(L))$ 를 사용하여 간단히 인버터와 NOR 게이트로 구현하였다. OR논리의 구현은  $y=(A+B)+(C+D)=!(!(A+B))*!(C+D))$ 의 부울리안 수식에 의하여 2 단 OR 논리를 NOR-NAND 게이트로 구성하였다.

#### IV. 실험 결과

앞에서 언급한 여러 종류의 encoder 들을 설계하고 상대적인 비교 및 검증하기 위하여 SPICE input file을 자동 생성하는 컴퓨터 프로그램을 개발하였다. 개발된 컴퓨터 프로그램은 (1) VCVS와 gain booster를 포함한 comparator, (2) fat tree encoder, (3) SSE (early mode control과 late mode control, 그리고 TCMUX 마다 인버터를 병행하는 경우와 TC 신호마다 두 개의 인버터를 사용하는 경우의 제어 회로)의 net list를 자동 생성 하며, equal delay sizing 방법에 의한 자동 크기 조절이 가능하다.

설계 공정으로 Hynix 0.25um 공정을 사용하였다 (IDEC 제공). 채널 길이는 0.25um로 고정하였으며, DC bias는 2.5V를 인가하였다. 실제 layout은 하지 않았으나 source와 drain의 diffusion 부분에 적어도 하나의 contact을 넣을 수 있을 만큼 그 크기를 정하였다. Hynix 0.25um 공정의 설계 규칙에서 적어도 하나의 contact이 있기 위한 diffusion 부분의 최소 폭은  $W_{diff,min}=0.58um$ , 최소 길이는  $L_{diff,min}=0.64um$ 이고, 최소 크기 MOSFET의 채널 폭은  $W_{ch,min}=0.35um$ 이다 (제공 한 자료로는 0.32um이나 공정 마진을 고려). 따라서, MOSFET의 채널 폭이  $W$ 이면  $W_{diff}=\max(W, W_{diff,min})$ ,  $AS=AD=(L_{diff,min}*W_{diff})$ ,  $PS=PD=(2*L_{diff,min}+W_{diff})$ 로 계

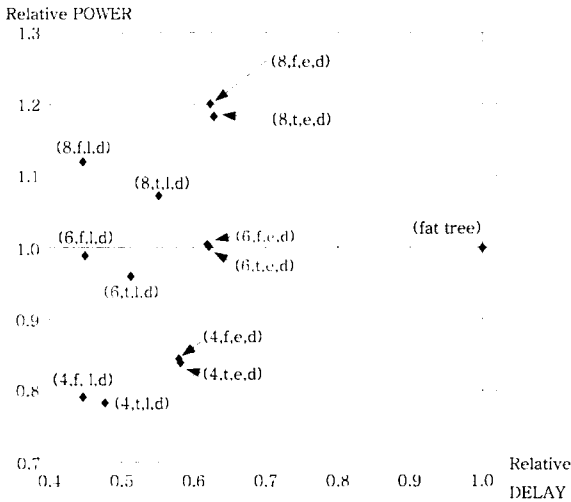


그림 7. Fat tree encoder에 대하여 지연시간과 전력 소모를 정규화한 실험 결과 (f: MUX마다 인버터를 병행하는 경우, t: TC 신호마다 두 개의 인버터를 사용하는 경우, l: late mode control, e: early mode control, d: equal delay sizing)

Fig. 7. Power-delay graph normalized with respect to the fat tree encoder.

표 2. Fat tree encode와 SSE의 비교 (FOX=10, SSE: f, l, d)

Table 2. Comparison of the fat tree encoder and the SSE. (FOX=10, SSE: f, l, d).

bit	method	delay (ps.)	power (uW)	f	F <sub>TOT</sub>	Size (um)	#TR	inv	nand/nor
									mux
4	fat tree	528.58	13.723	2.48	21.014	538.37	266	55	8/31
	SSE	236.80	10.892	2.32	16.688	406.03	176	55	11
6	fat tree	733.23	47.634	2.31	25.211	1821.2	1130	203	44/137
	SSE	328.94	47.170	2.24	20.985	1356.3	750	204	57
8	fat tree	941.03	169.18	2.15	29.076	6758.1	4586	783	192/563
	SSE	420.11	189.70	2.19	25.327	4860.0	3052	785	247

산된다. 내부 연결선의 효과를 추정하기 위하여 MOSFET 마다 게이트 커패시턴스의 10%에 해당되는 선형 커패시턴스를 게이트 노드와 접지 노드 사이에 추가로 연결하였다. Transistor sizing 시, 입력 대비 출력의 크기 비율은 10으로 하였다. 즉, FOX=10 (10개의 표준 인버터가 출력 부하로 연결). 시뮬레이션 결과, PN ratio는 2.65, 표준 인버터의 NMOSFET 폭은  $W_{N, std}=0.6\mu m$ 로 정하였고, 지연 시간은 FOX=3의 조건에서 약 60psec로 측정된다.

Comparator 부분은 VCVS (SPICE의 E-element)를 사용하여 기능 구현 만을 하였으며, 2 단의 표준 인버터 들로 gain booster를 구현하였다. 즉, 외부 analog 입력 신호는 VCVS와 gain booster를 거쳐서 encoder의 입력으로 제공된다. VCVS의 전압 이득은 25,000으로

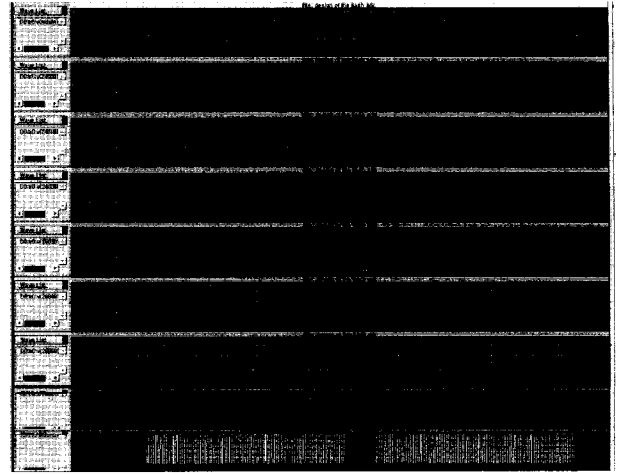


그림 8. 8-bit SSE의 실험 결과

Fig. 8. Simulation results of the 8-bit SSE.

하였고 analog 입력 전압의 상, 하한 전압은 각각 0.5V와 2.0V로 정하였다. 따라서, 양자화 전압은  $(2.0-0.5)/2^N$ 이 된다.

그림 8은 설계한 8-bit SSE의 동작 결과를 보여준다. 느리게 변화하는 삼각파 형태의 analog 입력 신호에 대하여 SSE가 올바른 출력 BC 신호를 발생하고 있다. 그림 7과 표 2는 시뮬레이션 결과를 보여준다. 모든 결과 자료는 공통으로 설계된 gain booster 부분과 encoder 부분을 합한 결과이다. 동작 속도 (worst case delay)와 전력 소모의 측정은 출력 BC 신호가 (0,0,...,0)에서 (1, 1,...,1)로 변갈아 변화하는 경우에서 측정한 결과이며, 이 때의 입력 기울기는 0.1nsec.로 하였으며, 전력 소모는 3개의 펄스 파에 대한 결과이다 (6번의 신호 변환).

그림 7에서 보면, 동작 속도의 관점에서 late mode control이 early mode control보다 우수하며, MUX 마다 인버터를 병행하는 경우가 TC 신호마다 2개의 인버터를 사용하는 경우보다 우수한 것으로 나타난다. 표 2를 보면, 제 II 장에서 예상한 바와 같이 SSE가 fat tree encoder에 비하여 크기의 관점에서 적은 게이트 수와 트랜지스터의 수를 사용하고 있으며, 설계 면적도 작다. 설계 면적은 트랜지스터들의 채널 폭을 전부 합한 결과인데 실제 레이아웃을 하였을 경우에도 비슷한 비율을 보일 것으로 예상된다. 실험 결과를 보면, SSE가 fat tree encoder에 비하여 50% 이상의 동작 속도 향상을 보여준다. 8-bit SSE의 경우, fat tree encoder에 비하여 전력 소모가 다소 증가 하였는데, 이는 제어신호를 발생하기 위하여 MUX마다 인버터를 병행하여 다는 방법을 사용하였기 때문이다. 한편, forward SSE와 backward SSE의 비교는 2장에서 언급한 바와 같이 이

론적으로 명확히 차이가 나므로 실험적으로 그 차이를 규명하지는 않는다.

## V. 결 론

새로운 형태의 TC-to-BC encoder를 제안하고, Logical Effort 방법과 SPICE 시뮬레이션을 통하여 구조 설계 및 상세 설계를 수행하였다. 제안한 SSE는 MUX 논리에 의하여 동작되며, 그 구조로는 late mode control 방식과 MUX 마다 인버터를 병행하는 방법이 좋은 것으로 판단된다. SSE는 기존의 fat tree encoder에 비하여 계산 단계, 게이트 수, 전체 크기, 동작 속도의 관점에서 우수함을 이론적 모델과 실험 결과를 통하여 확인할 수 있었다. 특히, 기존의 fat tree encoder보다 제안된 SSE 방식이 50% 이상의 동작 속도 향상을 보이는 것을 실험을 통하여 확인할 수 있었다. 제안된 SSE는 고속 flash ADC의 설계에 적합한 것으로 사료된다.

## 참 고 문 헌

- [1] Daegy Lee, Jincheol Yoo, Kyusun Choi, and Jahan Ghanavi, "Fat Tree Encoder Design for Ultra-High Speed Flash A/D converters," from <http://www.cse.psu.edu/~MDL/paper/mwscas02.pdf>, 2002.
- [2] Jincheol Yoo, Kyusun Choi, and Jahan Ghaznavi, "Quantum Voltage Comparator for 0.07 um CMOS Flash A/D Converters," in ISVLSI, pp. 280-281, 2003.
- [3] Ivan Sutherland, Bob Sproull, and David Harris, "Logical Effort: Designing Fast CMOS circuits," Morgan Kaufmann Publishers, Academic Press, 1990.
- [4] 이승훈, 김범섭, 송민규, 최중호 공저, "CMOS 아날로그/혼성모드 집적시스템 설계," IDEC 교재개발시리즈 16, 시스마프레스, 1999.
- [5] R. Kanan, F. Kaess, and M. Declereq, "A 640mW High Accuracy 8-bit 1GHz Flash ADC Encoder," IEEE International Symposium on Circuits and Systems, 2: 420-423, 1999.
- [6] P. Xiao, K. Jenkins, M. Soyuer, H. Ainspan, J. Burghartz, H. Shin, M. Dolan, D. Harame, "A 4b 8G Sample/s AD Converter in SiGe Bipolar Technology," IEEE International Solid-State Circuits Conference, pp. 124-125, 1997.
- [7] H. Luong, D. Hebert, and T. Duzer, "Fully Parallel Super-conducting Analog-to-Digital Converter," IEEE Transactions on Applied Superconductivity, 3(1): 2633-2636, March, 1993.
- [8] M. Choi and A. Abidi, "A 6b 1.3G Sample/s AD converter in 0.35um CMOS," IEEE International Solid-State Circuits Conference, pp. 1276-1278, 2001.
- [9] B. Yu and W. Black, "A 900MS/s 6b Interleaved CMOS Flash ADC," IEEE Custom Integrated Circuits Conference, pp. 149-152, 2001.
- [10] S. Padoan, A. Boni, C. Morandi, and F. Venturi, "A Novel coding Schemes for the ROM of parallel ADCs featuring reduced conversion noise in the case of single bubbles in the thermometer code," IEEE International Conference on Circuits and Systems, 2: 271-274, 1998.
- [11] M. Sugawara, H. Yoshida, M. Mitsuishi, S. Nakamura, S. Nakaigawa, Y. Kunisaki, and H. Suzuki, "A 2.5V 100MS/s 8bit ADC using Pre-Linearization Input Buffer and Level Up DAC/Subtractor," Symposium on VLSI Circuits Digest of Technical Papers, pp. 170-173, 1998.
- [12] M. Ito, T. Miki, S. Hosotani, T. Kumamoto, Y. Yamashita, M. Kijima, T. Ockuda, and K. Okada, "A 10bit 20MS/s 3V Supply CMOS A/D Converter," IEEE J. Solid-State Circuits, 29(12): 1531-1536, Dec. 1994.
- [13] 나유삼, 송민규, "3.3V 8-bit 200MSPS CMOS Flooding/Interpolation ADC의 설계," 전자공학회논문지 38권 SD 3호, 44-50쪽, 2001년 3월.

저 자 소 개



이 기 준(정회원)

1978년 서울대학교 공업교육학과 전자전공 (공학사)

1981년 한국과학기술원 전기 및 전자공학과 (공학석사)

1986년 한국과학기술원 전기 및 전자공학과 (공학박사)

1986년~현재 충남대학교 전기정보통신공학부 교수

<주관심분야 : VLSI-CAD, 회로설계, 알고리즘>



Kyusun Choi

1985년 Electrical Engineering, Lehigh Univ., Bethlehem, PA, B.S.

1987년 Electrical Engineering, Pennsylvania State Univ., M.S.

1993년 Dept. of Computer Science and Engineering, Pennsylvania State Univ., Ph. D.

1993년~현재 Assistant Professor, Dept. of Computer Science and Engineering, Pennsylvania State Univ.

<주관심분야 : Mixed/RF/DSP 집적회로 설계>



김 병 수(정회원)

1999년 남서울대학교 전자공학과 (공학사)

2001년 한양대학교 전자공학과 (공학석사)

2002년~현재 충남대학교 전자공학과 박사과정 재학 중

1986년~현재 삼성전자

<주관심분야 : 반도체 제조공정/패키징>