

Pipelined A/D 변환기 용 Charge-Shared Switching MDAC의 설계

박 만 규, 이 중 훈, 김 상 호, 김 상 민, 손 영 철, 김 대 정, 김 동 명
국민대학교
전화 : 02-910-4704 / 핸드폰 : 016-484-6594

Design of the Charge-Shared Switching MDAC for a Pipelined A/D Converter

Man-Gyu Park, Jong-Hoon Lee, Sang-Ho Kim, Sang-Min Kim,
Young-Chul Sohn, Daejeong Kim, Dong Myong Kim
Kookmin University
E-mail : mg486@hanmail.net

Abstract

This paper proposed a new charge-shared switching MDAC for a pipelined A/D converter. The proposed architecture accomplishes the same function of a conventional multiplying-digital-to-analog converter (MDAC). By adopting the proposed scheme, about 40% of the total capacitances could be reduced and the speed of the MDAC increases. The performance of the charge-shared switching MDAC has been proved by HSPICE simulations.

환기 중 빠른 동작속도를 얻으면서 면적과 전력 소모를 최적화한 구조가 pipelined A/D 변환기이다 [1]. 이러한 pipelined A/D 변환기에서는 입력 전압과 내부 A/D 변환기의 출력 전압의 차이인 잔류 전압을 증폭하는 MDAC의 역할이 매우 크다. 일반적으로 MDAC은 고정밀 switched capacitor 회로가 사용되며, 따라서 capacitor의 사용을 줄일 수 있는 구조가 요구된다.

본 논문에서는 pipelined A/D 변환기에 사용하는 기존의 MDAC에 비해 capacitor array를 약 40%정도 감소하여 동작속도는 증가하고 전력소모를 감소한 charge-shared switching MDAC을 제안한다.

I. 서 론

오늘날 대부분의 전자 시스템에서는 analog 회로의 부담을 줄이고 더 높은 해상도를 구현하기 위해 digital signal processing (DSP) 기법을 사용한다. 이러한 기법을 사용하기 위해서는 analog 신호를 digital 신호로 바꾸어 주는 A/D 변환기가 반드시 필요하다. 이러한 A/D 변

II. 기존의 MDAC 구조

Pipelined A/D 변환기에 사용하는 기존의 N-bit MDAC의 구조는 그림 1에서와 같이 증폭기, capacitor array, 그리고 switch들로써 구성한다 [2]. MDAC의 샘플링 모드와 증폭 모드는 두 개의 중첩되지 않는 클럭인 Q_s 와 Q_a

를 통해 각각 이루어진다. 샘플링 모드 ($Q_s = \text{'high'}$) 시 capacitor array의 bottom-plate에 analog 입력 전압이 샘플링되고 top-plate에는 증폭기의 입력 옴셋 전압이 나타난다. 증폭 모드 ($Q_a = \text{'high'}$)시 2^N 번째 capacitor를 제외한 나머지 capacitor들의 bottom-plate는 flash ADC의 thermometer code 값에 따라 접지 또는 기준전압 V_{ref} 에 연결하고 2^N 번째 capacitor의 bottom-plate는 출력단에 연결한다. 이에 따라 출력 전압 V_o 는 analog 입력 전압과 flash ADC의 출력 비트의 양자화 된 analog 전압의 차이인 잔류 전압 V_r 은 2^{N-1} 배 증폭되어

$$V_o = 2^{N-1} V_r \quad (2.1)$$

로써 출력된다.

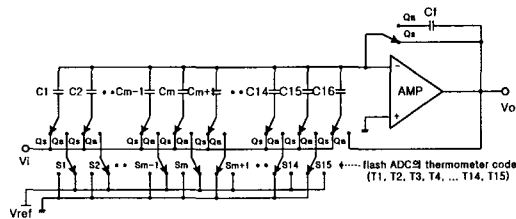


그림 1. 전형적인 N-bit MDAC의 구조 (N=4)

III. 제안하는 Charge-Shared Switching MDAC의 구조

3.1 Charge-Shared Switching MDAC의 개념

제안하는 charge-shared switching N-bit MDAC의 구조를 그림 2에 나타내었다.

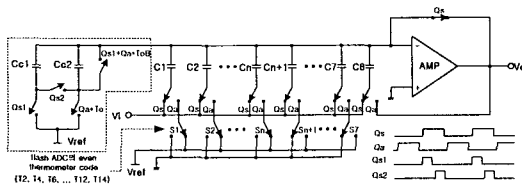


그림 2. 제안하는 charge-shared switching N-bit MDAC의 구조(N=4)

샘플링 모드 동안 capacitor array를 충전하는 전하량 Q_s 는

$$\begin{aligned} Q_s &= 2^{N-1} C V_i \\ &= 2^{N-1} C \left(m \frac{V_{ref}}{2^N} + V_r \right) \end{aligned} \quad (3.1)$$

이고, 증폭 모드 동안 capacitor array를 충전하는 전하량 Q_a 는

$$Q_a = n C V_{ref} + C V_o \quad (3.2)$$

이다. 따라서 전하량 보존의 법칙에 의해 출력단에 나타나는 출력 전압 V_o 는

$$V_o = 2^{N-1} V_r + \left(\frac{m}{2} - n \right) V_{ref} \quad (3.3)$$

이다.

식(2.1)과 식(3.3)에서, 제안하는 charge-shared switching N-bit MDAC이 정상적인 출력 $2^{N-1} V_r$ 이 출력단에 나타나기 위한 조건은 $m = 2n$ 이다. 여기에서 m 은 'high'가 되는 flash ADC의 thermometer code의 개수이고 n 은 charge-shared switching N-bit MDAC에서 기준 전압 V_{ref} 에 연결되는 unit capacitor의 개수이다. 따라서 표 1에서 보는 바와 같이 제안하는 구조에서 정상적인 동작을 위해서는 m 이 홀수일 때 $\frac{V_{ref}}{2}$ 의 기준전압을 필요로 한다.

표 1. 제안하는 4-bit MDAC에서의 m과 n의 관계

m	n	m	n
1	1/2	9	4 + 1/2
2	1	10	5
3	1 + 1/2	11	5 + 1/2
4	2	12	6
5	2 + 1/2	13	6 + 1/2
6	3	14	7
7	3 + 1/2	15	7 + 1/2
8	4		

3.2 Charge-Shared Switching 단

제안하는 charge-shared switching N-bit MDAC 구조에서의 charge-shared switching 단은 그림 2에서 점선 안에 나타내었다. Q_a 이 'high'일 때 charge-shared capacitor C_d 의 bottom-plate에 기준 전압 V_{ref} 를 샘플링하고 Q_b 가 'high'가 되면 C_d 에 충전되어 있는 charge가 C_d 과 C_e 에 똑같이 sharing되어 C_e 에는 $\frac{V_{ref}}{2}$ 를 샘플링한다. Q_a 가 'high'일 때 charge-shared capacitor C_e 를 flash ADC의 odd thermometer code의 logic 신호 T_o 에 따라 선택적으로 V_{ref} 에 switching한다. 이러한 charge sharing 개념을 이용하면 증폭 모드 동안 선택적으로 capacitor array의 bottom-plate에 $\frac{V_{ref}}{2}$ 를 switching 한 효과를 나타낼 수 있다. 한편, 클럭이 반복될 때마다 C_e 에 charge가 계속 충전되어 있는 현상을 방지하기 위해 Q_s , Q_a 와 T_o 의 logic signal에 의한 방전경로를 둔다. 또한 Q_b 에 의한 스위치의 feedthrough error를 제거하기 위해 클럭 Q_b 의 falling time은 샘플링 클럭 Q_s 의 falling time과 증폭 클럭 Q_a 의 rising time의 사이에 위치한다.

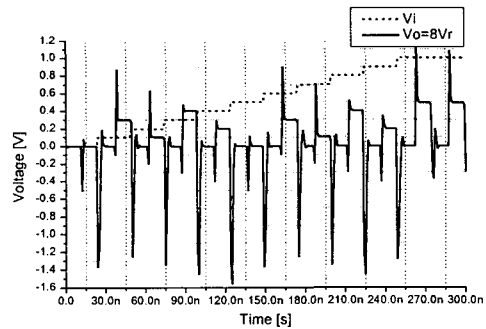
3.3 Charge-Shared Switching MDAC의 동작 원리와 특징

표 1에서 보는 바와 같이 m 이 짝수인 경우 n 은 정수이고 n 개의 unit capacitor의 bottom plate를 기준 전압 V_{ref} 에 연결한다. 그리고 m 이 홀수인 경우에 n 은 '정수+1/2' 이므로 그 정수만큼 unit capacitor를 기준전압 V_{ref} 에 연결하고 1/2에 대해서는 charge-shared switching단을 이용하여 $\frac{V_{ref}}{2}$ 에 연결한 효과를 얻는다.

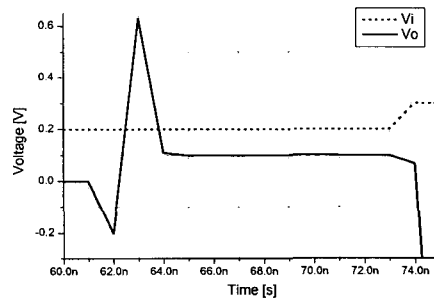
이러한 charge-shared switching MDAC을 사용할 경우 기존의 4-bit MDAC에서 17개의 unit capacitor들을 사용하지만 제안하는 MDAC에서는 10개의 unit capacitor들만으로 기존의 MDAC의 기능을 완벽하게 수행한다. 따라서 약 40%의 capacitor array의 감소로 칩 내에서의 면적과 전력소모는 감소하는 반면 동작 속도는 증가하여 기존의 MDAC에 비해 뛰어난 성능을 얻을 수 있다.

IV. Simulation 결과

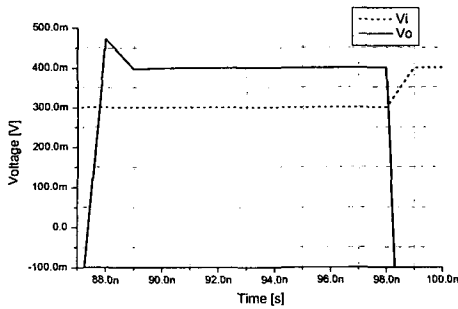
그림 3 (a)는 charge-shared switching 4-bit MDAC의 구조에서 40MHz 샘플링 속도로 입력 전압을 변화시켜 가면서 0~300ns까지 CMOS 0.35 μ m 표준 공정에 HSPICE로써 simulation한 입력과 출력의 결과 파형이다.



(a)



(b)



(c)

그림 3. (a) 제안하는 charge-shared switching 4-bit MDAC 구조의 simulation 결과 파형 (b) 입력 전압 $V_i = 300\text{mV}$ 일 때 출력전압의 확대 파형 (m=짝수) (c) 입력 전압 $V_i = 200\text{mV}$ 일 때 출력전압의 확대 파형 (m=홀수)

그림 3 (b), (c)는 입력 전압이 200mV (m=홀수)와 300mV (m=짝수)일 때 각각의 출력 전압을 확대하여 나타낸 파형들이다.

표 2는 제안하는 charge-shared switching 4-bit MDAC 구조의 이론치와 그림 3의 결과를 바탕으로 하여 서로 비교한 표이다.

표 2. 제안하는 charge-shared switching 4-bit MDAC 구조의 이론치와 simulation 결과 비교

입력전압 [mV]	m	n	잔류 전압(×8)	
			이론치[mV]	결과치[mV]
0	0	0	0	0
100	1	1/2	300	299.7
200	3	1+1/2	100	100.2
300	4	2	400	399.9
400	6	3	200	199.9
500	8	4	0	-0.1
600	9	4+1/2	300	300
700	11	5+1/2	100	100
800	12	6	400	399.9
900	14	7	200	199.8
999	15	7+1/2	192	192.1

위에서 보는 바와 같이 제안하는 charge-shared switching 4-bit MDAC 구조는 기존의 MDAC의 기능을 완벽하게 수행함을 검증하였다. 또한 같은 조건 하에서 기존의 MDAC의

capacitor array 전력 소모는 $1.5 \mu\text{W}$ 이었으나 제안하는 구조는 $0.76 \mu\text{W}$ 로써 전력소모가 절반으로 감소하였다.

V. 결론

본 논문에서는 pipelined A/D 변환기에 사용하는 charge-shared switching MDAC 구조를 제안하였다. 이 구조를 채택함에 따라 전체 unit capacitor의 수가 약 40% 줄어 칩 내에서의 면적은 감소하고 속도가 증가하며 capacitor array의 전력소모는 감소하도록 하였다. 기존의 MDAC 구조의 기능 수행과 동작 성능의 향상을 CMOS $0.35\mu\text{m}$ 표준 공정에 HSPICE로 simulation하여 검증하였다.

Acknowledgments

본 논문은 Brain Korea 21 프로젝트의 지원과 IDEC(IC design education center)의 디자인 소프트웨어 지원에 의한 것이며, 이에 깊이 감사드립니다.

참고 문헌

- [1] 이승훈, 김범섭, 송민규, 최중호, CMOS 아날로그/혼성모드 집적시스템 설계(下). 시그마프레스, 1999.
- [2] 최희철, 안길초, 이승훈, 강근순, 이성호, 최명준, "10-bit 20-MHz CMOS A/D 변환기" 전자공학회 논문지, 제33권, A편, 제4호, 1996. 4.