비동기 시스템용 고성능 16비트 승산기 설계

김학윤, 이유진*, 장미숙, 최호용
충북대학교 반도체공학과

Design of High Performance 16bit Multiplier for Asynchronous Systems

Hak-Yuń Kim, Eugene Lee, Mi-Sook Jang, Ho-Yong Choi
Dept. of Semiconductor Eng., Chungbuk Nat'l Univ.

Abstract - A high performance 16bit multiplier for asynchronous systems has been designed using asynchronous design methodology. The 4-radix modified Booth algorithm, TSPC (true single phase clocking) registers, and modified 4-2 counters using DPTL (differential pass transistor logic) have been used in our multiplier. It is implemented in 0.65㎛ double-poly/double-metal CMOS technology by using 6616 transistors with core size of 1.4×1.1㎟. And our design results in a computation rate exceeding 60㎒ at a supply voltage of 3.3V.

1. 서 론

신호처리 및 컴퓨터 응용기술의 발전과 함께 디지털 시스템의 구현 기법과 전자기기들의 소형화 추세에 따라서 고성능, 저전력으로 동작할 수 있는 구조들이 연구되고 있다. 비동기식 설계 기법은 전역클럭을 제거함으로써 불필요한 신호천이에 의한 전력소모 감소 및 저수준의 EMI설계가 용이하고, handshake protocol에 의해 평균 지연 성능을 가짐으로써 현재의 동기식 회로 설계기법의 대안으로 주목받기 시작하고 있다[1].

본 논문에서는, 향후 멀티미디어·초고속 정보통신을 위한 고성능, 저전력 시스템 설계를 위해 가장 많이 사용되고 있는 서브시스템 중의 하나인 승산기를 비동기식 설계기법으로 설계한다. [2]와는 달리 본 설계에서는 별도의 ALU를 사용하지 않고 전용 연산을 하는 승산기로 설계한다.

※ 본 연구는 반도체설계교육센터의 부분적인 지원을 받아 이뤄졌음.

본 승산기의 알고리즘은 속도향상을 위해 modified Booth 알고리즘[3]을 이용한다. 또한 비동기식 제어를 위한 제어부와 곱셈을 수행하는 데이터패스(datapath) 부로 구성된다. 제어부는 외부와의 비동기식 제어를 위해 4상 프로토콜(4-phase protocol)[1][4]을 이용하고, 데이터패스부는 속도를 향상시키기 위해 3단의 pipeline 구조[5][6]를 채택한다. 또한 partial product compression 속도를 높이기 위한 4-2 counter의 사용 및 TSPC 레지스터[7]를 사용하여 트랜지스터의 개수를 줄임으로써 저전력 및 속도의 향상을 추구한다.

2. 비동기 회로의 데이터 전송방식

비동기 회로는 전역클럭을 사용하지 않기 때문에 데이터가 언제 유효한 지를 알리는 타이밍 정보가 포함되어야 한다. 이런 타이밍 정보를 포함하는 데이터 표현 방식으로 bundled data를 사용하고, 승산기 외부와의 데이터 전송방식으로 4상 프로토콜을 이용한다[1][4].

(1) Data bundling

Data bundling 전송방식은 제어신호(Req, Ack)와 데이터 다발(bundled data)을 사용하여 데이터를 전송한다.

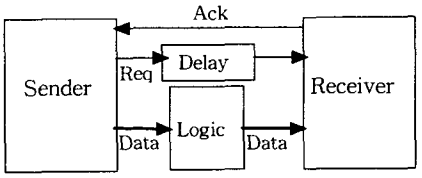

그림 1. 데이터 전송방식

그림 1과 같이 sender로부터 제어신호의 천이(Req ↑)가 발생하면, 조합회로의 예측된 연산시간보다 약간 긴 시간이 경과(delay)한 후에 receiver(제어신호와 데이터 다발을 받아들이는 쪽)는 안정된 데이터를 받고, 제어신호를 sender에게 발생(Ack↑)시켜 데이터의 전송완료 및 다음 데이터 수신준비를 알린다[1].

(2) 4상 프로토콜

승산기 외부 회로와의 데이터를 전송하기 위해 4상 프로토콜을 이용한다.

4상 프로토콜의 동작은 그림 2와 같이 데이터버스에 새로운 데이터가 실리면(①) Req는 "1"로 천이(Req ↑, ②)하며, receiver가 데이터 수신의 준비가 되어 있을 때 데이터를 받아들이고 데이터의 전송완료에 해당하는 Ack 신호를 "1"로 천이(Ack↑,③)시킴으로써 응답한다. 이 receiver의 응답은 다시 sender를 초기상태로 천이(Req↓, ④)시키고, 이에 대한 응답으로 receiver도 초기상태로 천이(Ack↓, ⑤)한다. 4상 프로토콜은 4개의 시간영역(두개의 실선은 sender의 action, 두 개의 점선은 receiver의 action)으로 구분되며, 모든 제어신호(Req, Ack)가 다음 사이클 전에 초기화된다.

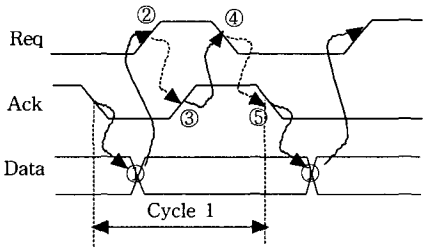

그림 2. 4상 프로토콜

3. Modified Booth 알고리즘

Modified Booth 알고리즘에서는 승수(multiplier)를 비트의 그룹들로 고려하여 처리한다. 본 논문에서는 승수의 비트들을 3비트의 그룹(4-radix)들로 나누어 처리함으로써, 덧셈의 회수를 1/2로 감소시켜 곱셈의 속도를 향상시킨다. 3비트 그룹 입력에 대해 표1과 같이 연산처리(action)를 하고 N, S, Z로 엔코딩하며[2], Booth mux의 제어신호의 역할을 한다. N(negative)은 -부호를, S(shift)는 ×2를, Z(zero)는 "0"의 flag를 나타낸다. 이때 신호 Z의 값이 "1"이면 pipe 레지스터를 리셋시켜 모든 비트의 출력을 "0"값으로 만들고, S가 "1"이 되면 pipe 레지스터는 1비트 left shift를 한다. N이 "1"이 되면 Booth mux에서는 1's complement된 값을 발생하고 이 값은 pipe 레지스터에 저장되는데, 이때 N 값은 2's complement를 만들기 위해 Modified 4-2 counter의 LSB에 더해진다.

표 1. Encoding method

Bit position			Action	Encoding		
i+1	i	i-1		N	S	Z
0	0	0	0	0	0	1
0	0	1	+M	0	0	0
0	1	0	+M	0	0	0
0	1	1	+2M	0	1	0
1	0	0	-2M	1	1	0
1	0	1	-M	1	0	0
1	1	0	-M	1	0	0
1	1	1	0	1	0	1

4. 승산기의 구조 및 회로 설계

(1) 구조 및 동작

그림 3과 같이 승산기는 크게 데이터패스부와 제어부로 구분된다.

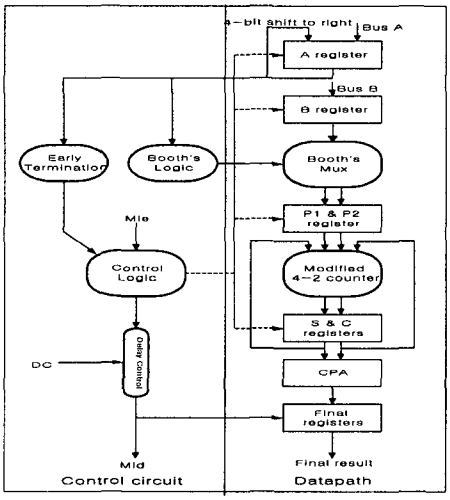

그림 3. 곱셈기의 블록 다이어그램

데이터패스부는 곱셈연산을 수행하며, 제어부는 데이터패스에 제어신호를 발생시키고 외부 모듈과 신호를 송수신한다. 곱셈연산의 과정은 다음의 3단계로 수행된다.

① 버스 A, B에 연산수들이 입력되면, 곱셈 요청신

호(Mle)에 의해 P1, P2, C, S 그리고 final 레지스터가 리셋되고, 두 연산수들은 A와 B 레지스터로 저장된다.

② A레지스터에 모두 "0"이 검출될 때까지 LSB (least significant bit) 방향으로 4비트씩 shift 시키면서 승수의 하위 5비트에 대해 부분곱셈을 반복한다.

③ Termination 비트가 검출되면 곱셈 완료신호(terminate)가 발생한다. Delay 단에 의해 지연된 terminate신호(Mld)가 CPA단의 덧셈결과를 최종 레지스터(final register)에 안정되게 래치 시킴으로써 곱셈이 완료된다.

여기서 Mle신호는 외부회로에서 승산기로의 Req신호 역할을 하고 Mld신호는 승산기에 이어지는 외부회로에서의 Reg신호의 역할을 한다.

표 2는 승산기의 내부 클럭의 사이클에 따른 각 블록별 데이터 처리상황을 보여준다. 본 곱셈기의 데이터패스부분은 최대 4개의 클럭 사이클을 필요로 한다. Booth logic은 승수를 Booth 엔코딩하는 부분이며, Booth logic의 숫자는 승수가 내부클럭의 각 사이클마다 4비트씩 처리되고 있음을 보여준다. 우선 클럭이 발생하기 전에 16비트 승수의 LSB부터의 4비트는 미리 엔코딩되고 이어지는 내부클럭에 의해 차례대로 4비트씩 엔코딩되어 그에 해당하는 N, S, Z값이 발생한다. Booth mux는 Booth logic에서 발생한 N, S, Z 값에 따라 각 싸이클마다 같은 피승수(16비트)를 처리한다. 설계된 승산기는 매 클럭마다 두 번의 Booth 엔코딩을 함으로써 두 개의 partial product를 발생하고, 이 partial product들은 modified 4-2 counter에서 덧셈이 수행된다. 표 2에서 P1~P8은 partial product를 표시한다. 표에서 보듯이 각 싸이클마다 2개의 partial product가 처리되고 있음을 알 수 있다. CPA항의 "O" 표시는 최대 5번째 내부클럭이 발생하면 final 레지스터로 CPA의 안정된 결과 값이 저장됨을 나타낸다.

표 2. pipe timing에 따른 데이터 처리

Cycle / Action	0	1	2	3	4	5
Booth's logic (bit)	0-3	4-7	8-11	12-15		
Booth's mux (bit)	0-15	0-15	0-15	0-15		
Modified 4-2 counter (partial product)		P1,P2	P3,P4	P5,P6	P7,P8	
CPA						O

(2) 회로 설계

본 승산기의 제어부의 설계는 그림 4와 같이 2개의 C게이트를 사용하여 4 phase 프로토콜로 설계한다. 곱셈요청 신호(Mle)가 입력되면, reset, 데이터 loading 제어신호인 Lt와 nLt, 그리고 내부클럭이 순차적으로 발생한다. 내부클럭은 5개의 인버터체인에 의해 1.9ns의 일정한 주기를 갖는다. 외부에서 terminate신호가 입력되면 곱셈완료신호(Mld)가 발생한다. 또한 연산속도를 향상시키기 위한 early termination 단은 전력소모를 줄이기 위해 정적 CMOS회로로 설계한다.

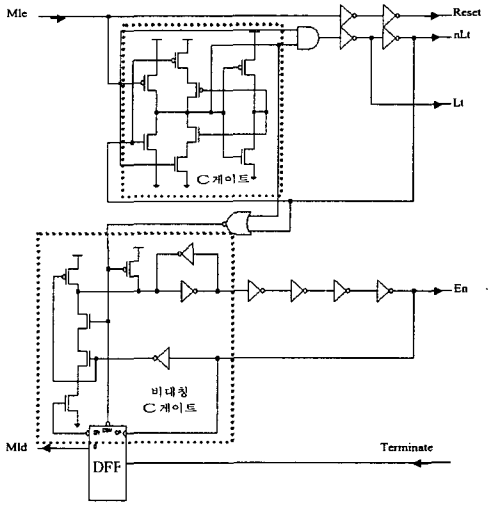

그림 4. 제어부 회로

데이타패스부는 TSPC 방식을 이용한 레지스터, DPTL[2]을 이용한 modified 4-2 counter 회로를 이용하여 고성능, 저전력의 16비트 비동기 승산기를 설계한다. 그림 5와 같이 TSPC 레지스터는 하나의 phase를 가지기 때문에 clock skew문제를 피할 수 있고, dynamic 레지스터이기 때문에 미리 충전된 전하에 의해 고속으로 동작할 수 있다. 또한 TSPC 레지스터는 최소 사이즈와 최소 개수의 트랜지스터를 사용하고, 하나의 enabling signal을 가짐으로써 전력 소비를 줄일 수 있다[2][7].

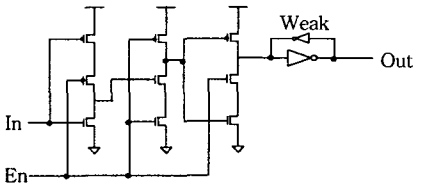

그림 5. Positive-edge triggered TSPC 레지스터

Modified 4-2 counter의 설계시는 낮은 전압전원에서도 충분한 noise margin을 얻고, 대칭적 구조로 쉽

게 layout할 수 있는 DPTL 방식을 사용한다.

32비트 CPA단은 속도의 향상을 위해 CSA (carry skip adder)[8]를 비트 분할(상위 비트부터 6비트-7비트-6비트-5비트-4비트-4비트)하여 설계한다.

5. 결 과

각 회로의 논리검증을 위해 compass툴을 이용하였으며, Hspice를 이용하여 설계회로의 timing 분석을 하였다. 각 블록의 지연시간은 표 3과 같다.

표 3. 각 블럭의 지연시간

Block	Delay(ns)
Register	0.668
Booth's logic	0.485
Booth's mux	0.177
Modified 4-2 counter	0.797
CPA(carry skip adder)	9.11

데이터패스 부분은 pipeline구조를 사용하여 설계하였기 때문에 승산기의 내부에 사용되는 내부클럭의 주기는 지연시간이 가장 큰 순차회로 단(A register + Booth's logic + Booth's mux)과 제어회로에서 만들어 줄 수 있는 주기를 고려하여 1.9ns의 주기를 갖도록 구현되었다.

본 승산기의 최대 곱셈연산 시간은 4회의 내부클럭의 주기와 CPA단의 연산시간의 합인 16.71ns(60㎒)가 되었다.

본 승산기를 2중 metal, 2중 poly의 0.65㎛공정을 이용하여 구현한 결과, 6616개의 트랜지스터가 사용되고, 면적은 1.4×1.1㎟로 작게 구현되었다. 승산기의 전체 레이아웃 도면은 그림 7과 같다.

6. 결 론

본 논문에서는 modified Booth algorithm을 이용하여 비동기 설계기법으로 16비트 고성능 승산기를 설계하였다. 설계결과 1.4×1.1㎟의 작은 면적으로 구현되었으며 60㎒ 동작주파수를 얻었다.

본 승산기는 데이터패스부의 처리 시간에 알맞은 주기의 내부클럭을 발생시킬 수 있는 제어부의 설계와 CPA단의 성능을 향상시킨다면 보다 나은 성능의 승산기가 될 수 있을 것이다.

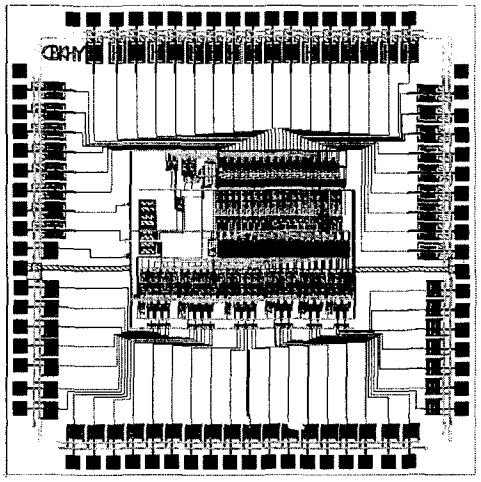

그림 7. 비동기 승산기 레이아웃

[참고문헌]

[1] Al Davis and Steven M. Nowick, "An introduction to asynchronous circuit design," UUCS-97-013, Sep. 1997.

[2] Jianwei Liu, "The design of an asynchronous multiplier," Master's Thesis, Dept. of Computer Science, Manchester University, England, Sep. 1995.

[3] Gensuke Goto and Atsuki Inoue, "A 4.1-ns compact 54×54-b multiplier utilizing sign-select Booth encoders," IEEE Journal of Solid-State Circuits, vol. 32, no. 11 pp. 1676-1682, Nov. 1997.

[4] 최병수, "A study on delay insensitive 16-bit asynchronous microprocessor design and *implementation*," Master's Thesis, Dept. of Infomation & Communications, K-JIST, 1997.

[5] M. Santoro, "SPIM: a pipelined 64×64bit iterative multiplier," IEEE Journal of Solid-State Circuits, vol. 24, no.2 pp. 487-493, Apr. 1989.

[6] Ivan E. Sutherand, "Micropipelines," Communications of the ACM, vol. 32, no. 6, pp. 720-738, June 1988.

[7] Jiren Yuan and Christer Svensson, "High-speed CMOS circuit technique," IEEE Journal of Solid-state Circuits, vol. 24, no. 1, pp. 62-70, Feb. 1989.

[8] Wayne Wolf, *Modern VLSI design*, Prentice Hall, 1998.