휴대단말용 RISC 프로세서의 32비트 MAC 구조

정갑천*, 박성모**
*전남대학교 전자공학과 **전남대학교 컴퓨터공학과
gcjung@ciscom.chonnam.ac.kr

(32-bit MAC Architecture of a RISC Processor for Portable Terminals)

Gab-Cheon Jung*, Seong-Mo Park**
*Dept. of Electronics Eng., Chonnam National Univ,
**Dept. of Computer Eng., Chonnam National Univ.

Abstract

In this paper, we designed 32-Bit MAC architecture of a RISC processor for portable terminals such as cellular telephones, personal digital assistants, notebooks, etc. In order to have minimum area with best performance, the MAC performs 32 by 8 multiplication per cycle, with early termination circuit that enables multiply cycles depend on the value of multiplier. It uses the sign bit of a partial product and two extra bits for sign extension,

The MAC is modeled and simulated in RTL using VHDL. The MAC is synthesized using IDEC C-631 Cell library based on 0.6㎛ CMOS 1-poly 3-metal CMOS technology.

Ⅰ. 서 론

셀룰러 폰, 휴대전용 단말기, 노트북 등과 같은 휴대 단말 시스템에서는 데이터 I/O 동작과 사용자 인터페이스 외에도 음성, 오디오, 비디오의 압축/복원 등과 같은 멀티미디어 응용들이 요구된다. 이와 같은 멀티미디어 응용들은 필터링, FFT(Fast Fourier Transform), DCT(Discrete Cosine Transform), 모션 보상 등의 신호처리들을 포함하며, 곱셈(multiply)과 누산(multiply-accumulate) 연산들이 많이 사용되어진다. 휴대단말기에 사용되어지는 RISC 프로세서나 DSP 프로세서는 이와 같은 신호처리 연산들을 실시간에 처리하기 위해서는 명령어 처리속도 향상뿐만 아니라 고속의 MAC이 필요하다.

본 논문은 전남대학교 학술연구비 지원에 의하여 연구되었음

이전에는 MAC 기능이 외부적인 DSP 프로세서나 LSI에서 구현되어 왔으나 휴대형의 내장형 시스템에서는 마이크로 프로세서, DSP 칩 모두를 사용하는 것은 시스템의 크기를 증가시키게 될 뿐 아니라, 전력 소모를 증가하게 되고, 특히 가격면에서 부담이 될 수 있어 마이크로프로세서에서 MAC 기능을 수행할 수 있도록 하는 것이 바람직하다. 그러나 MAC 회로가 내장형의 마이크로프로세서 내에 구현되기 위해서는 높은 성능 외에도 회로의 크기도 작도록 설계되어야 한다.

본 논문에서는 휴대 단말기용 32비트 RISC 프로세서에서 사용될 수 있는 signed/unsigned MAC (multiply accumulator) 구조를 설계하였다. 설계된 MAC은 3단 파이프라인으로 동작하며, 승수 값 및 곱셈 연산에 따라 3-7 싸이클이 소요된다.

Ⅱ. 곱셈 및 누적 연산

1. 수정된 부스 알고리즘

일반적으로 16 비트 이상의 곱셈기에서는 쉬프트와 덧셈연산의 반복연산이나, 어레이 구조, 수정된 부스 알고리즘이 사용되어진다. 쉬프트와 덧셈연산의 반복연산을 사용하는 경우 하드웨어의 크기가 적고, 전력을 적게 소비하는 장점이 있으나 속도가 너무 느리고, 어레이 구조의 경우 정규적인 구조를 지니고, 배선이 쉽다는 장점을 지니나 전달지연이 오퍼랜드 크기에 따라 선형적으로 증가하여 32비트 이상의 경우는 적합하지 못하다[1].

수정된 부스 알고리즘은 더해질 부분 곱들의 수를 감소시키기 위해 2의 보수의 recoding에 기반을 두는데 이는 하드웨어 면적을 적게 차지하게되고 곱셈기의 속도를 향상시킬 수 있어 본 논문에서는 부분 곱들의

수를 1/2으로 줄일 수 있는 Radix-2 수정된 부스 알고리즘을 사용하였다[2]. 표1은 Radix-2 수정된 알고리즘의 부분곱 선택을 나타낸다.

표 1. 부분곱 선택

Multiplier Bits	Selection
000	0
001	+Multiplicand
010	+Multiplicand
011	+2 × Multiplicand
100	-2 × Multiplicand
101	-1 × Multiplicand
110	-1 × Multiplicand
111	0

2. 부호 확장

부스 알고리즘을 적용한 곱셈기의 VLSI 구현의 경우 어레이를 규칙적으로 하기 위해 부호 확장기술을 사용하는데, 32비트의 경우 완전 부호확장을 수행하게 되면 하드웨어 면적이 많아지게 되고, 전달 지연도 증가하게된다. 특히 본 구조가 32 비트용인 것을 고려하여 부분곱 생성시 완전 부호확장 대신 다음과 같이 부분곱의 부호 비트(PPn)외에 2비트만을 추가하였다[3].

- 첫번째 PP(Partical Product):

$PP_{n+2} = PP_{n+1} = PP_n$ -------(1)

- 두번째 이후의 PP : -------(2)

i) 이전의 PP가 양수 : (1) 과 같음

ii) $PP_n = 0 : PP_{n+2} = PP_{n+1} = 1$

$PP_n = 1 : PP_{n+2} = PP'_{n+1} = 1$

$\cdot F_{j+1} = F_j + PP_{n,j}$ ($PP_{n,j}$: j번째 부분곱의 부호비트)-(3)

위의 식(1),(2)는 이전의 부분곱에서 음수 부호가 전달될 것인지를 지시하는 부가적인 플래그 비트 F를 사용하여 구현될 수 있으며 식 (3)과 같이 나타낼 수 있다. 여기서 $PP_{n,j}$는 j번째 부분곱의 부호비트를 나타낸다.

3. 곱셈 및 누적 연산 동작

수정된 부스 알고리즘을 32비트 곱셈에 적용하였을 때 부분곱의 수가 16개가 되고, 이를 모두 병렬 처리하도록 설계하면 속도면에서는 향상을 보이나, 하드웨어 크기가 너무 커지게 되고 전력소모도 많아진다. 본 논문에서는 32비트 연산을 1싸이클에 32비트×8비트 연산을 수행하도록 하였으며, 곱셈기가 병렬 처리할 부분곱의 수도 4개로 감소하므로 하드웨어 면적을 감소한다.

지원되는 곱셈 연산 기능들은 표2에 나타내었으며 Rm은 피승수, Rs는 승수, Rn은 MUL 또는 MLA 때 누산될 수를, Rd는 결과를 저장하는 레지스터를 나타낸다.

표 2. 설계된 MAC에서 지원되는 연산 기능

명령어	동 작	기 능	결과
MUL	multiply	Rd := Rm * Rs	하위 32비트
MLA	multiply and accumulate	Rd := Rm*Rs+Rn	하위 32비트
UMULL	unsigned multiply long	RdHi,RdLo := Rm*Rs	64비트
UMLAL	unsigned multiply and accumulate	RdHi,RdLo := Rm*Rs+RdHi,RdLo	64비트
SMULL	signed multiply long	RdHi,RdLo := Rm*Rs	64비트
SMLAL	signed multiply and accumulate	RdHi,RdLo :=Rm*Rs+RdHi,RdLo	64비트

Ⅲ. 설계된 MAC 구조

설계된 MAC은 그림 1의 블록도와 같이 부분곱 발생기, CSA 트리, 최종 부분합들의 결과를 위한 ALU 등의 세 부분으로 구성되어지고, 3단 파이프라인으로 동작한다.

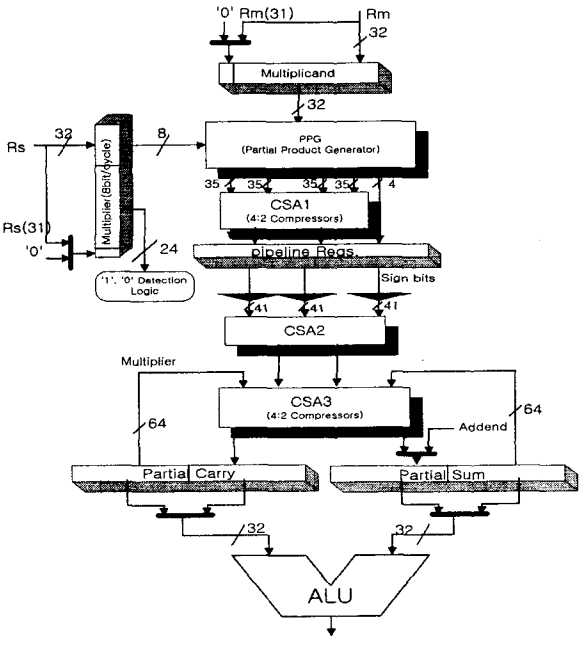

그림 1. 설계된 MAC 블럭도

1. 부분곱 발생기

부분곱 발생기는 1싸이클에 4개의 부분곱을 발생시키는 블록으로써 위에서 기술된 수정된 부스 알고리즘 및 부분곱의 축소된 부호 비트 사용에 기반을 두어 설계하였으며, 부분곱 발생기에서는 음수의 경우는 1의 보수 연산을 수행하고, 부분합들의 트리내에서 '1'을 더하도록 하여 2의 보수 연산을 수행하도록 하였다. 그림 2는 부분곱 발생기중 하나의 부분곱 선택기 로직을 나타낸다.

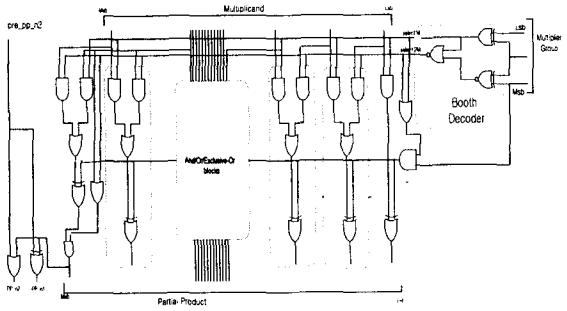

그림 2. 부분곱 선택기 로직

2. CSA 트리

부분곱들의 합을 위한 CSA 트리는 부분곱들의 효율적인 덧셈을 위해 4-2 compressor들로 구성된 2개의 CSA와 1개의 CSA(Carry Save Adder) 들로 구성되었다. 여기서 4-2 compressor는 두 개의 직렬 연결된 전가산기로 구성되며, 4개의 수와 하나의 캐리를 입력으로 하여 2개의 수와 하나의 캐리를 출력으로 내보낸다.

첫 번째 CSA 블록은 35 비트로 구성되어지며 부분곱 발생기에서 나온 4개의 부분곱들을 더하는 연산을 수행하고, 2번째 CSA는 41 비트로 구성되어지고, 보수 연산이 필요한 경우에 부분곱 발생기에서 나온 부분곱들에 '1'을 더하는 역할을 수행한다. 또한 unsigned의 경우 피승수의 값을 더하는 역할을 수행한다. 마지막 CSA 블록은 이전 부분 곱들의 합과 현재 부분 곱들을 누산한다. MAC 연산이 누산 연산의 경우에는 첫 번째 싸이클 때 누산 오퍼랜드 값을 Partial Sum에 저장함으로써 따로 누산할 때 부가적인 싸이클이 요구되지 않도록 하였다. unsigned의 경우는 승수의 값을 더하는 역할도 수행한다.

최종적인 출력 값은 32비트 RISC의 ALU를 사용하는데, ALU는 32 비트이기 때문에 요구되는 연산 출력이 64비트일 경우 처음에 하위 32비트를 내보내고, 다음 상위 32비트를 내보도록 하였다. 그림 3은 32 비트 곱셈을 위한 CSA 트리의 각 싸이클별 연산 수행을 나타낸다.

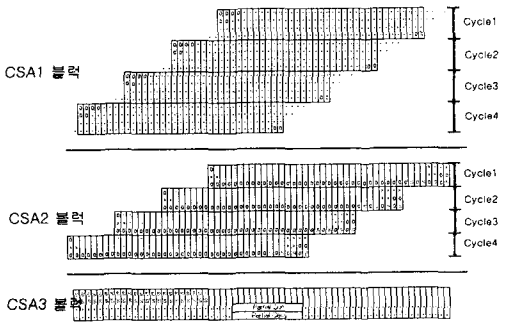

그림 3. CSA 트리의 각 싸이클별 연산

3. 조기 종결 회로

MAC 싸이클의 수는 승수 레지스터에 저장되어진 값에 달라질 수 있는데 승수 레지스터는 쉬프트 레지스터로서 싸이클당 8비트 쉬프트 하게되어 32×8 비트 연산을 수행하게 되며, 승수레지스터 7비트에서 32비트 값이 모두 '0'이거나 '1'이면 이후의 부스 인코딩 값은 '0'이 되므로 조기 종결하도록 하였다. 그러므로 RISC에서 곱셈 연산자중 값이 적은 수를 승수 레지스터로 지정하면 싸이클의 수를 감소시킬 수 있다. 표3은 승수레지스터의 값에 따른 싸이클 수를 나타낸다.

표3. 승수 값에 따른 곱셈 싸이클

승수값 / Mnemonic	2^8 이하	$2^8 \sim 2^{16}$	$2^{16} \sim 2^{24}$	$2^{24} \sim 2^{32}$
MUL/MLA	3	4	5	6
UMULL/ UMLAL	4	5	6	7
SMULL/ SMLAL	4	5	6	7

4. 제어 로직

3단 파이프라인으로 동작하는 MAC 제어를 위해서는 RISC 주 제어회로와는 별도로 제어로직을 설계하였으며, 동작 상태도는 그림 4와 같다.

그림 4의 상태도에서 mcycle0는 부분곱 발생기와 CSA1 블록을 제어하기 위한 싸이클 상태이고, mcycle1은 mcycle0의 파이프라인된 싸이클 상태로서 CSA3의 입력부의 mux들을 컨트롤하게 된다. et는 조기 종결회로에서의 출력값으로 '1'이면 곱셈 수행을 종결하고, 초기 "00" 상태로 귀환한다. 그림 5는 multiply long 명령어(UMULL, SMULL)들을 처리하기 위한 상태도로 partial sum, partial carry의 출력부의 mux들을 제어하기 위한 상태도이다.

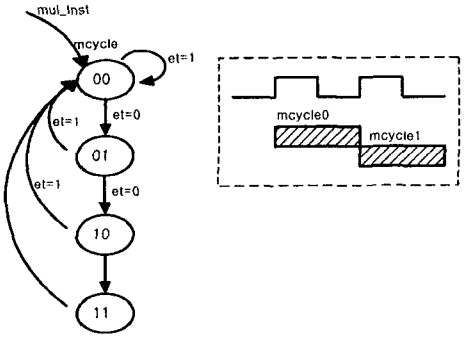

그림 4. MAC 싸이클 상태도

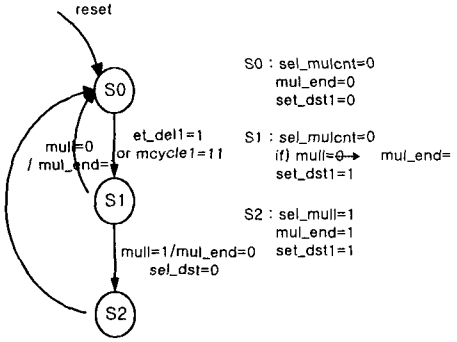

그림 5. UMULL, SMULL명령어를 처리하기 위한 상태도

Ⅳ. 시뮬레이션 및 구현

32비트 MAC은 VHDL을 사용하여 RTL 수준에서 모델링되었으며, 기능 검증 후 RISC 코어 VHDL모델에 추가되어 시뮬레이션 되었다. 그림 6은 MLA 명령어의 시뮬레이션을 결과를 보여주는데, 승산기의 값이 607327 즉 2^{16}~2^{24} 사이의 값이므로 오퍼랜드들이 enable된 후 결과값 "FFFE131FE8D45744"이 나오기까지 총 5 싸이클이 걸림을 알 수 있다.

RISC 코어 내부에서 동작이 검증이 완료된 MAC은 0.6㎛ CMOS 1-poly 3-metal 공정 기반의 IDEC C-631 셀라이브러리를 사용하여 Synopsys 툴에서 논리 합성 및 검증되었으며, 합성된 회로는 약 8400 게이트이다. 합성된 회로는 타이밍 검증 후 Mentor 툴에서 Auto P&R되었다.

Ⅴ. 결 론

본 논문에서는 셀룰러 폰, 휴대전용 단말기, 노트북

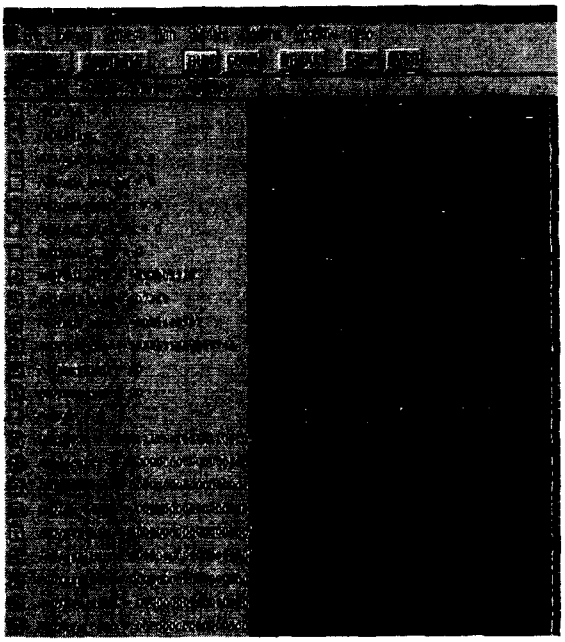

그림 6. MLA 명령어 시뮬레이션 결과

등과 같은 휴대 단말기용 32비트 RISC 프로세서에서 사용될 수 있는 signed/unsigned MAC 구조를 설계하였다. 설계된 MAC은 3단 파이프라인으로 동작하고, 부분곱 생성시 완전 부호확장 대신 3비트 만을 사용하며, 32×8비트 하드웨어 면적을 지닌다. 또한 조기 종결 기능을 갖도록 하여 승수 값에 따라 MAC 기능이 3-7 싸이클 소요된다. 설계된 MAC은 음성, 오디오, 비디오의 압축/복원 등과 같은 고속의 MAC 동작이 필요한 응용에 내장 모듈로서 효율적으로 사용될 수 있을 것이다.

참고 문헌

[1] H. Al-Twaijry, M. Flynn ,"Performance/Area Tradeoffs in Booth Multipliers", Technical Report: CSL-TR-95-684, Stanford Univ., November 1995.

[2] P.E. Madrid et. al., "Modified Booth Algorithm for High Radix Fixed-Point Multiplication", IEEE Trans. on VLSI Systems, Vol.1, No.2, pp.164-167, June 1993.

[3] I. S. Abu-khater, A. Bellaouar, M. I. Elmasry, "Circuit Techniques for CMOS Low-Power *High-Performance Multipliers*", *IEEE Journal of* Solid-State Circuits, Vol.31, No.10, pp. 1535-1546, October 1996.