

복합구조 승산기의 설계

송호정, 이재진, 송기용
 충북대학교 컴퓨터공학과

hjsong@dce3.chungbuk.ac.kr, ceicarus@just.chungbuk.ac.kr, gysong@chungbuk.ac.kr

Design of a Multiplier with Complex Structure

Ho-Jeong Song, Jae-Jin Lee, and Gi-Yong Song
 Dept. of Computer Engineering, Chungbuk National University

Abstract

This paper proposes a 16-bit × 16-bit multiplier for 2 twos-complement binary numbers with complex structure and implements it on a FPGA.

I. 서론

승산은 다양한 디지털 시스템에서의 기본적인 연산 중의 하나이다. 잘 알려져 있는 조합 승산기의 하나인 배열 승산기[1]는 소자들의 효율적인 반복성을 지니고 있으므로 VLSI 구현에 적합하다. 중복(redundant) 2진 연산[2-4]을 기반 하는 승산기에서 가산은 일정한 시간에 수행되지만 중복 2진 표현을 처리하기 위하여 과도의 게이트가 요구된다. 이 논문에서, 복합 구조를 가지는 16-bit 승산기를 제안하고, 그것을 FPGA 상에 구현하였다.

II. 배열 승산기

Booth 배열 승산기는 2의 보수 2진수에 대해 승산을 수행하는 전형적인 조합 승산기이다. Booth 승산기는 소자의 효율적인 반복성을 지니

고 있으며, 전체적인 계산 시간을 감소시키기 위해 연산을 동시에 수행한다. 간소화한 데이터 흐름의 4-bit Booth 승산기의 셀 배열은 그림 1에서 보였다.

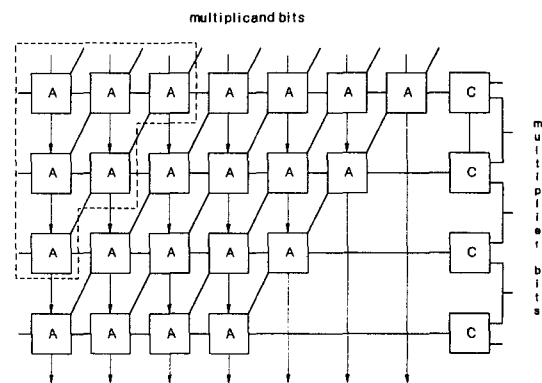


그림 1. 4-bit Booth 배열 승산기

셀 C는 Booth 코드, 셀 A는 가산과 감산을 위한 셀이다. 점선에 의해 둘러싸인 영역에 있는 셀 들은 음수 계산에 의해 요구된 부호-확장을 위해 사용된다. 또한 n-bit 승수를 위한 행의 수는 n개이다.

Booth 승산기의 각 bit를 위한 셀 들의 행은 가산이나 감산의 잠재력으로 표현되기 때문에 Booth 코드로부터의 가산과 감산의 수의 감소는 이 배열에 도움이 되지 않는다. 배열의 높이 n을

감소시키기 위해, canonical 부호 숫자 코드 (canonical signed digit code)나 radix-4 코드가 고려되어야 한다. 예를 들어 canonical 부호 숫자 표현은 먼저 승수의 분리된 1과 0들을 확인함으로써 얻을 수 있다. 단지 한번의 가산이나 감산은 고립된 1이나 0들에 의해 수행될 필요가 있다. canonical 부호 숫자 형식을 위한 규칙은 표 1에 보였다.

표 1. canonical 부호 숫자 형태를 위한 규칙

multiplier bit-pair	mode -in	Booth bit	mode -out
0 0	0	0	0
0 1	0	1	0
1 0	0	0	0
1 1	0	1̄	1
0 0	1	1	0
0 1	1	0	1
1 0	1	1̄	1
1 1	1	0	1

mode는 승수의 인접한 비트 쌍과 mode-in이 전가산기의 입력으로 들어갔을 때 승수의 1과 0이 전가산기의 carry-out과 정확히 같다는 검출에 의하여 1과 0들이 분리된 것을 확인한다. 그러므로 carry-lookahead 회로를 이용하여 1 bit를 한번에 승산기를 통하는 대신 1 스텝에 승산기를 canonical 부호 숫자 표현으로 동시에 코드화 할 수 있다.

비록 canonical 부호 숫자 표현이 평균 2n/3개의 0들을 포함한다해도, canonical 부호 숫자 형식에서 각 비트 쌍은 가산과 감산의 잠재력을 표현하고 이 결과는 수정 배열(modified array)에서 n/2개의 행을 갖게된다. 다시말하면, radix-4 코드는 승수의 인접한 3 비트를 조사하고 표 2에 따라 위치에 상응하는 코드값을 생성한다.

표 2. radix-4 코드를 위한 규칙

multiplier adjacent bits	bit-recoding at corresponding position
0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	1
1 0 0	1̄
1 0 1	1̄
1 1 0	1̄
1 1 1	0

radix-4 코드는 평균 5n/8개의 0들을 포함한다. 그러나 승산기의 각 비트 쌍을 위한 하나의 동작인 가산이나 감산의 잠재력의 수는 n/2개이다. 그래서 radix-4 코드의 상태는 코드기 회로(recoder circuit)의 차이를 제외하고는 canonical 부호 숫자 코드와 같다.

그림 2a에서 보인, canonical 부호 숫자 표현을 위한 코드기는 carry-lookahead 회로를 사용하는 mode-out을 결정하고, 승수 비트와 carry look-ahead 회로로부터 입력을 받은 변환기로부터 코드를 생성시킨다. 또한 그림 2b에서 보인, radix-4 표현을 위한 코드기는 승수의 인접한 3개의 비트로부터 코드를 생성한다.

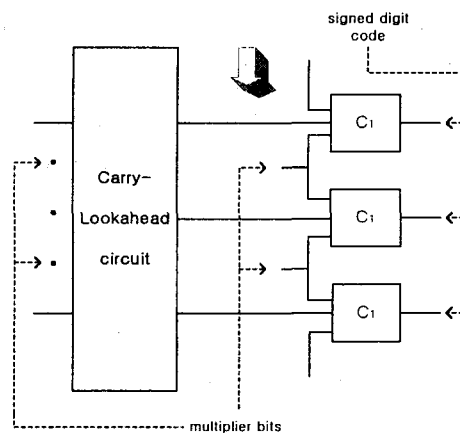


그림 2a. canonical 부호 숫자 형태를 위한 변환기

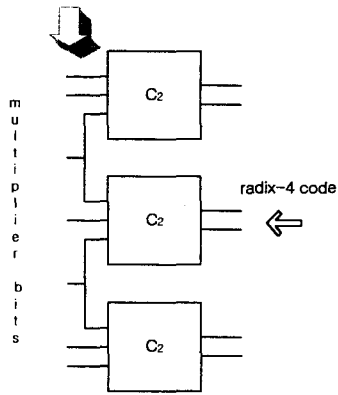


그림 2b. radix-4 코드를 위한 변환기

그림 2의 C1, C2는 각 경우에서의 변환기이다. 그림 2a에서 n-bit의 승수를 계산하는 승산기에서 radix-4 코드를 위한 코드기가 n/2개의 변환기 셀을 필요로 하는데 비해서 canonical 부호 숫자 형태를 위한 코드기는 n개의 변환기 셀을 필요로 한다.

radix-4 코드기의 복잡도는 canonical 부호 숫자 코드기의 복잡도 보다 낮다. 그러므로 승산기를 위한 코드는 radix-4 코드로 선택하였다.

간소화된 데이터 흐름의 radix-4 코드를 기반으로 한 8-bit 수정 배열 승산기의 셀 배열은 그림 3에서 보였다.

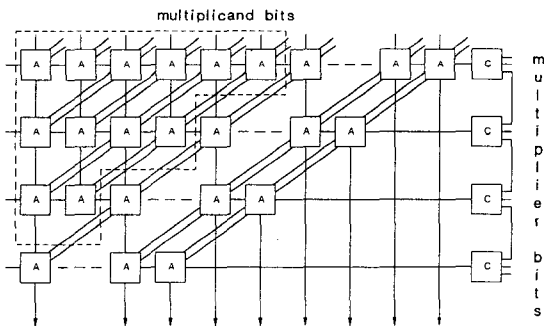


그림 3. 수정 배열 승산기

그 배열의 행의 수는 n/2개까지 감소되었지만

부호 확장을 위한 소자는 여전히 음수 계산을 수용하기 위해 수용해야 한다.

III. 복합 구조 승산기의 설계

부호 확장을 위한 소자는 제거되고, 각각의 행의 쌍을 분리하는 것에 의해 높이(행의 수)는 다시 감소될 것이다. 그리고 복합 구조를 따라 각각의 쌍을 분산시킨다. 복합 구조를 가진 2의 보수 2진수를 위한 16-bit 승수의 배치는 그림 4에서 보였다.

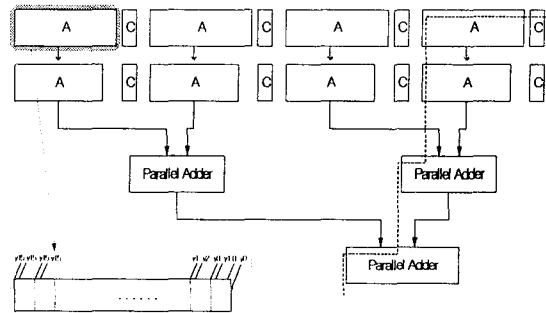


그림 4. 16-bit 복합 구조 승산기

제일 위의 두 개의 행은 코드화 된 승수 비트들의 각 쌍을 위한 부분 결과를 만들어내고, 그들을 동시에 가산한다. 그런 다음 부분 결과의 합을 아래의 병렬 가산기에 공급한다. 즉, 부분 결과의 축적된 합은 중간에 있는 병렬 가산기로부터 생성되고 아래 부분의 병렬 가산기로 입력되어 다시 가산된다. 점선은 승산 시간을 결정하는 전파 경로를 보인다.

그림 5는 Spartan XCS40-PQ240[5] FPGA 상에 합성된 16-bit × 16-bit 복합 구조 승산기가 구현된 것을 보인다.

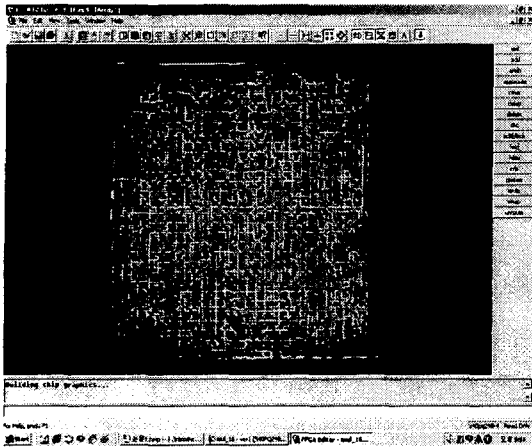


그림 5. 복합 구조 승산기의 구현

IV. 결론

이 논문에서 radix-4 코드를 기반으로 한 복합 구조의 16-bit×16-bit 2의 보수 2진수 승산기를 제안하였고 그것을 FPGA상에 구현하였다.

구현된 복합 구조 승산기는 기존의 Booth 배열 승산기나 수정 배열 승산기보다 낮은 높이(행의 수)로 설계되어 계산 속도가 더 빠름을 알 수 있다.

향후 연구 계획으로는, Booth 배열 승산기, 수정 배열 승산기, 복합 구조 승산기 각각에 대한 복잡성과 계산 속도의 관점에서 비교 분석이 요구된다.

참고문헌

- [1] Hayes, J.P., *Computer Architecture and Organization*, 3rd Ed. McGraw Hill
- [2] K.W.Shin, B.S.Song and K.Bacrania, "A 200-MHz Complex Number Multiplier Using Redundant Binary Arithmetic." *IEEE J. Solid-State Circuits*, vol.33, pp.904-909, June 1998.
- [3] S.M.Yen, C.S.Laih, C.H.Chen and J.Y.Lee, "An Efficient Redundant-Binary Number

to Binary Number Converter." *IEEE J. Solid-State Circuits*, vol.27, pp 109-112, Jan.1992.

- [4] Y.Harata, Y.Nakamura, H.Nagase, M.Takigawa and N.Takagi, "A High-Speed Multiplier Using a Redundant Binary Adder Tree." *IEEE J. Solid-State Circuits*, vol.SC-22, pp28-33, Feb.1987.
- [5] Xilinx, 1999 *Xilinx Data Book*.