파이프라인 방식 Baugh-Wooley 승산기의 구현과 성능 분석

한강룡[0], 최정필, 송호정, 황인재*, 송기용
충북대학교 컴퓨터공학과, 컴퓨터교육과*
{hankang, welcomerain, hjsong}@archi.chungbuk.ac.kr, {ihwang*, gysong }@chungbuk.ac.kr

Implementation and Performance Analysis of the Pipelined Baugh-Wooley Multiplier

Kang-Ryong Han, Jeong-Pil Choi, Ho-Jeong Song, In-Jae Hwang*, Gi-Yong Song
Dept. of Computer {Engineering, Education*}, Chungbuk National Univ.

요 약

본 논문에서는 Baugh-Wooley 승산 알고리즘을 '8x8-bit 15 stage 파이프라인 배열 승산기', '8x8-bit 2 stage 파이프라인 배열 승산기', '순수 조합 논리 배열 승산기'의 방식으로 FPGA상에서 구현하였으며, 각 구현방식의 성능을 비교 분석하였다.

1.서론

필터링, 변조, 비디오 프로세싱 등에서의 승산은 대부분 DSP 프로세서를 사용하여 하드웨어적으로 처리되며, 하드웨어 승산 기능을 포함하는 DSP 프로세서는 반도체 제작사로부터 공급되어 대부분의 하드웨어 엔지니어가 사용하고 있다. 그러나 몇몇 디자인에서는 특정 목적의 DSP 프로세서를 구하기가 쉽지 않은 경우도 있다. 이 경우 하드웨어 설계 엔지니어는 DSP 프로세서 대신에 FPGA를 대안으로 채택하여 여러 가지 이점을 동시에 얻을 수 있다. 예를 들면, DSP 프로세서의 주변회로 구성에 필요한 추가 비용을 절약할 수 있으며, DSP 프로세서가 자체적으로 보유하는 명령어들 때문에 소비되는 전력소모를 줄일 수 있으며 FPGA를 사용함으로써 완벽한 유연성과 높은 성능, 이미 시판된 DSP 프로세서의 사용에 비해 저렴한 비용으로 기능을 수행 할 수 있는 이점이 있다.

본 논문에서는 FPGA상에 구현한 고성능 승산기로 Baugh-Wooley 승산 알고리즘을 기반으로 하는 '8x8-bit 15 Stage 파이프라인 배열 승산기', '8x8-bit 2 Stage 파이프라인 배열 승산기', '순수 조합 논리 배열 승산기'를 설계, 구현하고 이들의 성능을 분석하였다.

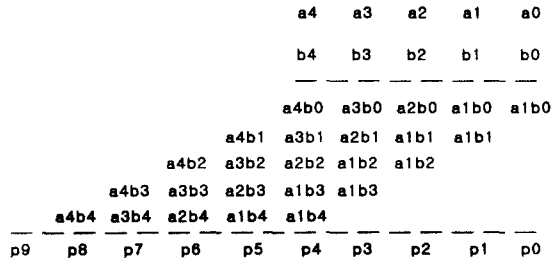
2. Baugh-Wooley 승산 알고리즘

Baugh-Wooley 승산 알고리즘은 unsigned number 승산기로부터 유도되었기 때문에 unsigned number 승산기로부터 접근할 필요가 있다. 피승수는 A로 승수는 B로 나타내었을 때, 이를 수식으로 표현하면 식(1)과 식(2)로 표현할 수 있다.

$$A=\sum_{i=0}^{N-1} a_i\,2^i \quad B=\sum_{j=0}^{N-1} b_j\,2^j \quad a_i\,b_j\in 0,1 \text{ --식(1)}$$

$$P=A\times B=\sum_{k=0}^{2N-1} p_k\,2^k=\sum_{i=0}^{N-1}\Big(\sum_{j=0}^{N-1} a_i\,b_j\,2^{i+j}\Big)\text{--식(2)}$$

위 식(2)에 N=5인 트리를 구성하면 <그림1>과 같은 트리를 얻을 수 있다.[2]



<그림 1> Unsigned Number Multiplication

식(3)과 식(4)는 signed binary number를 수식으로 표현한 것이다. 식(3)과 식(4)를 곱하여 식(5)와 같은 결과를 얻을 수 있다. 식(5)를 N=5인 트리로 구성하면 <그림2>와 같다. 식(3)과 식(4)에서 알 수 있듯이 signed number의 MSB는 크기와 부호를 함께 표현하므로 <그림2>에서 P7 = (-a4b3) +(-a3b4)와 같은 음의 부분곱이 나타나게 된다.[2]

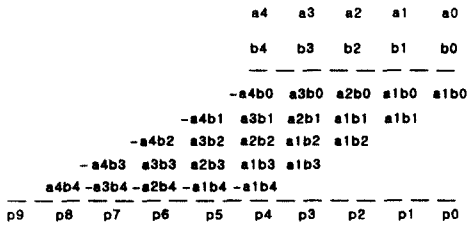
$$A=-\,a_{N-1}\,2^{N-1}+\sum_{i=0}^{N-2} a_i\,2^i \quad a_i\in 0,1\text{------식(3)}$$

$$B=-\,b_{N-1}\,2^{N-1}+\sum_{j=0}^{N-2} b_j\,2^j \quad b_i\in 0,1\text{ -------식(4)}$$

$$P=A\times B$$

$$=a_{N-1}\,b_{N-1}\,2^{2N-2}+\sum_{i=0}^{N-2}\sum_{j=0}^{N-2} a_i\,b_j\,2^{i+j}$$

$$-\Big(a_{N-1}\sum_{i=0}^{N-2} b_i\,2^{N+i-1}+b_{N-1}\sum_{i=0}^{N-2} a_i\,2^{N+i-1}\Big)\text{-식(5)}$$



<그림 2> Signed Number Multiplication

이처럼 signed number 승산은 음의 부분곱을 처리하기 위하여 부호 변환을 위한 특별한 처리 과정이 요구되기 때문에 이것을 하드웨어로 구현한다면 그만큼의 하드웨어 면적이 증가하며 처리속도가 지연 될 것이다.[1][2][5]

Baugh-Wooley는 음의 부분곱을 제거하기 위하여 식(5)의 음의 부분을 식(7)을 근거로 하여 식(6)과 같이 변형하였다.[1]

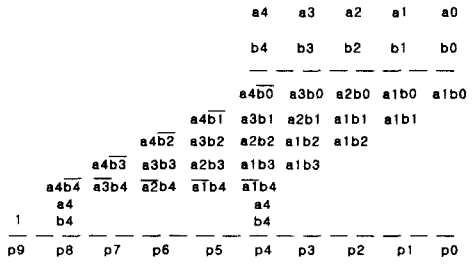
$$-a_{N-1}\sum_{i=0}^{N-2} b_i\,2^{N+i-1}$$

$$=a_{N-1}\left(-2^{2N-2}+2^{N-1}+\sum_{i=0}^{N-2}\overline{b_i}\,2^{N+i-1}\right)\quad\text{--식(6)}$$

$$-a_4b_0=a_4(1-b_0)-a_4=a_4\overline{b_0}-a_4\text{----------식(7)}$$

$$P=-2^{2N-1}+(a_{N-1}+b_{N-1}+a_{N-1}\,b_{N-1})\,2^{2N-2}$$
$$+\sum_{i=0}^{N-2}\sum_{j=0}^{N-2}a_i\,b_j\,2^{i+j}+(a_{N-1}+b_{N-1})\,2^{N-1}$$
$$+\sum_{i=0}^{N-2}b_{N-1}\,\overline{a_i}\,2^{N+i-1}+\sum_{i=0}^{N-2}a_{N-1}\,\overline{b_i}\,2^{N+i-1}\text{- 식(8)}$$

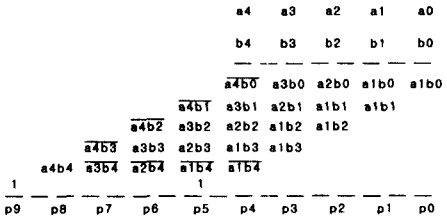
식(6)을 식(5)에 대입하면 식(8)과 같이 음의 부분곱이 제거된 식을 얻을 수 있으며, 이것을 N=5인 트리로 구성하면 <그림 3>과 같은 결과를 얻을 수 있다.[2]



<그림 3> Baugh-Wooley Multiplication

여기서 <그림 2>와 <그림 3>을 비교해보면, <그림 2> 트리의 최대 높이는 5이며, <그림 3> 트리의 최대 높이는 7이다. <그림 3>의 최대 높이가 <그림 2>보다 2가 큰 것을 알 수 있다. 이것을 하드웨어로 구현한다면, <그림 3>이 <그림 2>보다 처리시간이 지연될 수 있다. 이 문제점을 해결하기 위하여 Baugh-Wooley는 식(9) Modified Baugh-Wooley 승산 알고리즘을 제안하였다.[1] 식 (9)를 N=5인 트리로 구성하면 <그림 4>와 같다.

$$P=a_{N-1}\,b_{N-1}\,2^{2N-2}+\sum_{i=0}^{N-2}\sum_{j=0}^{N-2}a_i\,b_j\,2^{i+j}$$
$$-2^{2N-1}+2^N+\sum_{i=0}^{N-2}\overline{a_{N-1}\,b_i}\,2^{N+i-1}$$
$$+\sum_{i=0}^{N-2}\overline{a_i\,b_{N-1}}\,2^{N+i-1}\text{----------식(9)}$$



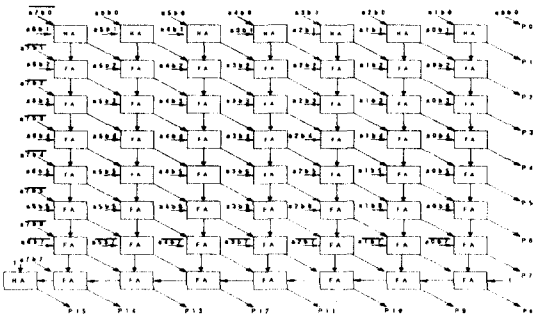
<그림 4> Modified Baugh-Wooley Multiplication

<그림 4> 트리의 최대 높이가 5로 <그림 3>과 비교했을 때 2가 감소하였고, <그림 2>와는 최대 높이가 같은 것을 알 수 있다. <그림 2>에서 음의 부분곱 문제와 <그림 3>에서 트리의 최대 높이가 커지는 것을 제거한 <그림 4>Modified Baugh-Wooley 승산 알고리즘을 하드웨어로 구현한다면 좀 더 성능이 좋은 승산기를 구현 할 수 있다.

3.파이프라인 방식 Baugh-Wooley 배열 승산기의 구조

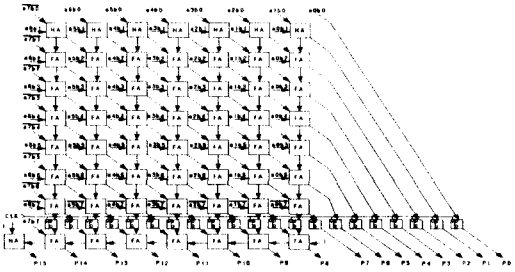
VLSI 또는 FPGA측면에서 볼 때 배열 승산기의 구조는 매우 높은 규칙성을 가지고 있기 때문에 전가산기들 사이의 수평,수직,대각선 연결들을 아주 짧게 구현할 수 있는 장점이 있으며, 간단하고 효율적인 레이아웃을 구성할 수 있다. 또한 전가산기들 사이에 래치를 추가하여 파이프라인을 구성함으로써 성능을 향상시킬 수 있다. 이와 같은 장점으로 인해 배열 승산기 구조는 자주 사용되고 있다.[2]

<그림 5>는 Modified Baugh-Wooley 승산 알고리즘을 기반으로 순수 조합 회로만 사용하여 배열 승산기를 구성하였다. 입력은 8x8-bit signed binary number이며 출력은 16-bit의 signed binary number이다. 최상단의 배열 블록들은 반가산기와 두 개의 AND 게이트로 구성되어 있으며, 이 외의 배열 블록은 전가산기 1개와 AND 게이트 1개로 구성된다. 대각선 방향을 통해 가산기의 sum이 전파되며, 수직선 방향으로 가산기의 carry가 전파된다.[2][3][4] 가장 낮은 하단의 배열 블록들은 Ripple Carry Adder구조를 가지고 있다.



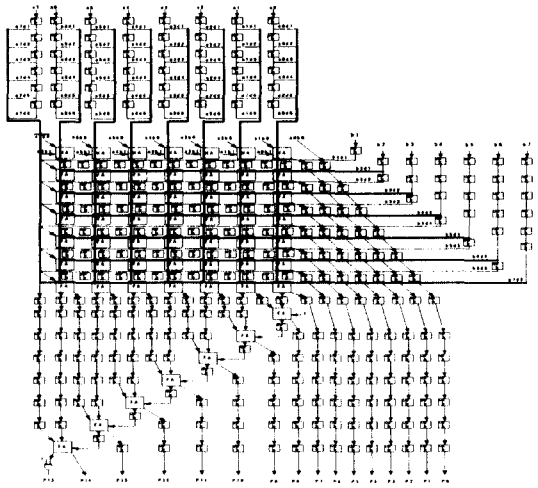
<그림 5> 조합회로로 구성한 8x8-bit B-W Array Multiplier

<그림 6>은 <그림 5>에 2 Stage 파이프라인 구조를 추가하였다. <그림 6>에서 21개의 D플립플롭을 추가하였으며, <그림 5>에 비해 약간의 처리속도 향상을 얻을 수 있다.



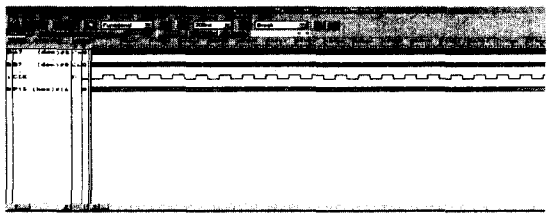
<그림 6> 8x8-bit 2 Stage Pipelined B-W Array Multiplier

<그림 7>은 모든 전가산기에 D플립플롭을 추가하여 15 Stage 파이프라인 구조를 추가하였다. <그림 5>에 비해 많은 크기의 하드웨어 면적이 소요되지만, 월등히 높은 처리 속도를 얻을 수 있다.[2][3]

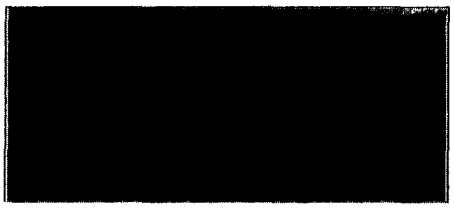


<그림 7> 8x8-bit 15 Stage Pipelined B-W Array Multiplier

<그림 8>은 <그림 7>을 Xilinx Foundation F2.1i상에서 VHDL을 이용하여 구현한 다음 시뮬레이션 한 결과이다. 13 CLOCK 지연 후 승산의 결과가 출력되고 있다. <그림 9>는 Xilinx S30PQ240 칩에서 합성한 결과이다.



<그림 8> 15 Stage Pipelined B-W Array Multiplier Simulation



<그림 9> 15 Stage B-W Array Multiplier in FPGA

4.성능 분석

<표 1>은 위 세 가지 승산기들에 대하여 회로구성(synthesis) 후 Xilnix Foundation 2.1i의 리포트를 비교 분석한 표이다. 2 Stage 8-bit signed multiplier는 조합 논리에 비해 CLB(Configuration of Logic Block)수가 감소하였다. CLB수가 감소한 이유는 CLB의 기능과 synthesis 알고리즘에 의한 차이점 때문이라고 볼 수 있다. 15 Stage 8-bit signed multiplier는 조합논리 8-bit signed multiplier에 비해 CLB수가 약 2배 증가하였다. 조합논리 8-bit signed multiplier에 비해 2 Stage 8-bit signed multiplier는 약 1.6배, 15 Stage 8-bit signed multiplier는 약 5배의 처리속도가 향상된 것을 볼 수 있다.

Multiplier	Area(CLB)	Minimum Period	Maximum Frequency
조합논리 8-bit Signed	86/576 14%	53 ns	19 Mhz
2 Stage 8-bit Signed	80/576 13%	32 ns	31 Mhz
15 Stage 8-bit Signed	163/576 28%	10.7 ns	93 Mhz

<표 1> 처리 속도와 면적의 비교

5. 결론

본 논문은 Baugh-Wooley 승산 알고리즘 기반의 배열 승산기를 '8x8-bit 15 Stage 파이프라인 배열 승산기', '8x8-bit 2 Stage 파이프라인 배열 승산기', '순수 조합 논리 배열 승산기' 방식으로 FPGA상에서 구현한 다음, 그 결과를 하드웨어 면적과 처리속도 측면에서 비교 분석하였다. 보다 높은 처리속도를 얻기 위하여 보다 많은 단계의 파이프라인 구조를 선택하였고, 이로 인해 2배정도의 하드웨어 면적이 증가하였지만 처리속도 측면에서 약 5배의 속도가 향상되는 것을 알 수 있었다. 이처럼 하드웨어 면적과 속도는 서로 trade-off 관계에 있으므로 회로 설계자는 회로 설계 시 요구사항에 따라 하드웨어 면적과 처리 속도를 적절히 절충하여 설계에 적용할 필요가 있다. 또한 이들 승산기를 바탕으로 더 좋은 성능의 고속 승산기의 연산 및 구현방식에 대한 연구가 필요하다 하겠다.

[참고문헌]

[1] Baugh, C. R and B. A. Wooley, "A Two's Complement Parallel Array Multiplication Algorithm," *IEEE Trans. Computers, Vol. 22*, pp. 1045-1047, December 1973.

[2] Behrooz Pasrhami , *"Computer Arithmetic Algorithms and Hardware Designs,"* Oxford University Press, 2000

[3] K.C. Chang, *"Digital Systems Design with VHDL and Synthesis,"* IEEE Computer Society, 1999

[4] Charles H. Roth, Jr., *"Digital Systems Design Using VHDL,"* PWS Publishing Company , 1998

[5] 신경욱 외 4명.*"디지탈 연산 알고리즘 및 회로 설계,"* IDEC 광반지역센-99-31, 1999