중첩 다중비트 주사기법을 사용하여 레지듀에서 이진수로 변환하는 컨버터

o
장 상 동, 김 우 완
경남대학교 컴퓨터공학과

RNS to Binary Converter Using Overlapped multiple-bit scanning method

Sang-Dong Jang, WuWoan Kim

Dept. of Computer Engineering, Kyungnam University

요 약

최근의 보편적인 컴퓨터 응용분야인 컴퓨터 그래픽, 패턴인식, 음성 출력 등과 같은 제분야에서는 대용량의 데이터를 실시간으로 처리하는 것이 필수적이다. RNS는 캐리부재, 병렬처리 등의 특징을 가지므로 대용량 데이터의 실시간 처리를 지원하는 장치의 개발에 큰 이점이 있다.

본 논문에서는 RNS에서 웨이티드 수체계로 변환하는 방법을 유도하고 구현한다. 이 방법은 연산의 비트수가 증가하더라도 고정된 연산의 단계를 거치게 되고, 여기에서 이 방법의 효율성이 커진다. 이는 중첩 비트 주사기법을 CRT 변환시에 적용하는 새로운 방법이다. 그리고, 변환식의 유도와 실제 시뮬레이션의 결과를 타 시스템과 비교하여 본 논문의 방법이 타당함을 보여준다. 그 결과, 기존의 승산기보다 많은 하드웨어를 요구하지만, 이는 최근의 반도체 집적기술의 발전으로 인하여 큰 문제가 되지 않고, 반면에 병렬 수행과 캐리 부재의 특성으로 인해 기존의 방법보다 속도를 향상시킬 수 있다.

1. 서 론

최근의 보편적인 컴퓨터 응용분야인 컴퓨터 그래픽, 패턴인식, 음성출력 등의 제분야에서는 디지털 신호 처리 하드웨어에 관한 연구가 고속 저가의 하드웨어 구현에 크게 기여하고 있으며, 대용량의 데이터를 실시간으로 처리하는 것이 필수적이다. RNS는 캐리부재, 병렬처리 등의 특징을 가지므로 대용량 데이터의 실시간 처리를 지원하는 장치의 개발에 큰 이점이 있다[1][2].

RNS에서 연산의 큰 이점으로는 RNS의 각 자리수가 서로 독립적이어서 병렬 수행과 캐리 부재(Carry-Free) 산술 연산이 가능하다는 것이다[3]. RNS에서 이진수로 변환하기 위한 기본적인 방법은 Chinese Remainder Theorem(CRT)을 이용한 접근 방법과 Mixed Radix Conversion(MRC) 알고리즘을 사용하는 접근 방법 두가지이다[4][5][6].

본 논문에서 사용하는 레지듀 수체계에서 이진수로 변환하는 변환기 구조는 CRT를 기반으로 한다. 이것은 모듈라이 집합(2^k-1, 2^k, 2^k+1)을 사용하며, 이 모듈라이의 특성을 응용하여 중첩 다중비트 주사(Overlapped Multiple-Bit Scanning)방법으로 CRT를 구현함으로써 계산 단계 및 속도를 줄일 수 있다. 이를 위해 간단한 룩업 테이블과 캐리 전달 지연을 줄일 수 있는 회로를 추가 구성한다.

2. 레지듀 수 체계

RNS에서 어떤 정수 x 의 레지듀 표현은 서로 다른 N-튜플로 된 레지듀 자리수 $x=(r_1,r_2,\Lambda,r_N)$로 나타낸다. 여기서 r_i는 N개의 집합에 대한 수식으로 정의된다.

$$x = q_i m_i + r_i \qquad i = 1,2,3,\Lambda,N \qquad (1)$$

그리고 q_i는 $0 \le x_i((m_i-1)$의 범위 내에 있도록 하는 가장 큰 정수이고, x/m_i의 몫으로 정수이며, 이것은 $[x/m_i]$라고 표기한다. r_i는 x를 m_i로 나눈 나머지로 양수가 된다. 그리고 이를 $x \bmod m_i$ 또는 $|x|_{mi}$로 표기한다.

r_i는 정수 x의 i번째 레지듀 자리수라고 부른다. 유일한 레지듀 표현을 위해서 모듈라이는 반드시 상대적으로 쌍을 이루는 소수이어야 한다. 즉,

$$GCD(m_i, m_j) = 1, \quad for \quad i \neq j \qquad (2)$$

이것은 범위 $0 \le x < (\prod_{i=1}^{N} m_i = M$ (N은 모듈라이의 수)내에서 유일한 표현이 존재한다는 것을 의미한다. 그리고 CRT는 이러한 레지듀 자리수 표현을 이진수로 변환하는 방법중의 하나이며 식 (3)과 같다.

$$|x|_M = \left| \sum_{j=1}^{N} \hat{m}_j \left| \frac{r_j}{\hat{m}_j} \right|_{m_j} \right|_M \qquad (3)$$

$$(\hat{m}_j = \frac{M}{m_j},\quad M = \prod_{j=1}^{N} m_j,\quad and \quad (m_j, m_k) = 1 \quad (for \quad j \neq k))$$

본 논문은 중첩 다중 비트 주사 기법의 적용하기 위해 모듈라이 집합(2^k-1, 2^k, 2^k+1)를 사용하고, 이때 두 수를 곱한 결과를 N이라고 하면, N은 3n 비트의 이진수로 표현 가능하고 그 범위는 식(4)와 같다.

$$0 \le N < (2^n - 1)2^n(2^n + 1) \qquad (4)$$

부호는 암시적 방법을 사용한다. 최초의 입력으로 2의 보수형태를 사용하고 CSA 트리와 CPA를 거쳐 나온 값에 대해 다시 2의 보수를 취한다.

3. 유도식

두 정수 A와 B의 각 레지듀 표현에 대한 곱은 식 (5)와 같이 나타낸다.

$$|A \cdot B|_M = \left(\|A|_{m1} \cdot |B|_{m1}|_{m1}, \|A|_{m2} \cdot |B|_{m2}|_{m2}, \|A|_{m3} \cdot |B|_{m3}|_{m3} \right) \qquad (5)$$

여기서 연산 ·은 곱셈으로 정의한다. 큰 수의 산술연산보다 작은 레지듀 자리수로 변환하여 연산하는 것이 바람직하며, 이때의 각 레지듀 자리수 연산은 서로 병렬적으로 수행되고, 레지듀 자리수 사이의 캐리는 존재하지 않는다.

식(5)의 각 레지듀를 이진수로 변환하기 위해 CRT 변환을 사용한다. 이때 본 논문에서 제안하는 중첩 다중 비트 주사기법의 적용을 수식으로 유도 및 적용하여 이의 타당함을 밝힌다. 식 (3)의 CRT 변환 공식에 사용할 각 모듈라이를 식 (6)과 같이 표현한다.

$$m_1 = 2^k - 1 \qquad m_2 = 2^k \qquad m_3 = 2^k + 1$$
$$M = \prod_{i=1}^{3} m_i = (2^k - 1)2^k(2^k + 1) = 2^{3k} - 2^k \qquad (6)$$

CRT에 적용하기 위해서 먼저 다음을 구한다.

$$\begin{aligned} |\hat{m}_1|_{m_1} &= |2^k \times (2^k + 1)|_{2^k - 1} \\ &= |2^k \times \{(2^k - 1) + 2\}|_{2^k - 1} \\ &= |2^k(2^k - 1) + 2(2^k)|_{2^k - 1} \\ &= |2(2^k - 1) + 2|_{2^k - 1} \\ &= |2|_{2^k - 1} \\ &= 2 (k \ge 2) \end{aligned}$$

$$\begin{aligned} |\hat{m}_2|_{m_2} &= |(2^k - 1) \times (2^k + 1)|_{2^k} \\ &= |2^k(2^k) - 1|_{2^k} \\ &= |-1|_{2^k} \\ &= |2^k - 1|_{2^k} \\ &= 2^k - 1 (2^k - 1 (2^k) \end{aligned}$$

$$\begin{aligned} |\hat{m}_3|_{m3} &= |(2^k - 1) \times (2^k)|_{2^k + 1} \\ &= |\{(2^k + 1) - 2\} \times \{(2^k + 1) - 1\}|_{2^k + 1} \\ &= |(2^k + 1)^2 - 3(2^k + 1) + 2|_{2^k + 1} \\ &= |2|_{2^k + 1} \\ &= 2 (k \ge 2) \end{aligned}$$

$$|\hat{m}_1|_{m1} = 2, \quad |\hat{m}_2|_{m2} = 2^k - 1, \quad |\hat{m}_3|_{m3} = 2 \qquad (7)$$

식(7)을 사용해서 곱셈의 역을 구할 수 있다. 곱셈의 역은 순환하는 특성이 있어 단지 몇 비트에 해당하는 환형 시프트로 계산된다. 식(7)의 값을 근거로 곱셈의 역을 구하면 표 1과 같이 유일한 값을 가지는 해가 나온다.

표 1. 곱셈의 역

	$\left\|\frac{1}{\hat{m}_1}\right\|_{2^k-1}$	$\left\|\frac{1}{\hat{m}_2}\right\|_{2^k}$	$\left\|\frac{1}{\hat{m}_3}\right\|_{2^k+1}$
k=2	1	3	3
k=3	4	7	5
k=4	8	15	9
k=5	16	31	17
k=6	32	63	33
M	M	M	M
k에 대한 곱셈의역	2^{k-1}	2^k-1	2^{k-1}+1

식(5)에 의해서 나온 각 레지듀 자리수를 각각 x, y, z라고 하고 식(3)에 적용하면 식(8)을 얻을 수 있다.

$$\begin{aligned} |N|_{2^{3k}-2^k} = &\left|2^k(2^k+1) \cdot \left|2^{k-1} \cdot x\right|_{2^k-1}\right. \\ &+ (2^k - 1)(2^k + 1) \cdot \left|(2^k - 1) \cdot y\right|_{2^k} \\ &+ (2^k - 1)(2^k) \cdot \left|(2^{k-1} + 1) \cdot z\right|_{2^{k+1}}\Big|_{2^{3k}-2^k} \end{aligned} \qquad (8)$$

$$\left|2^{k-1} \cdot x\right|_{2^k-1} = A, \quad \left|(2^k - 1) \cdot y\right|_{2^k} = B, \quad \left|(2^{k-1} + 1) \cdot z\right|_{2^{k+1}} = C \qquad (9)$$

식(9)를 식(8)에 적용하여 중첩 다중비트 주사기법에 적용할 수 있도록 스트링 속성(property)를 이용하여 식(10)과 같이 재배치한다.

$$\begin{aligned} &\left|2^k(2^k+1) \cdot A + (2^k - 1)(2^k + 1) \cdot B + (2^k - 1)(2^k) \cdot C\right|_{2^{3k}-2^k} \\ &= \left|(2^{2k} + 2^k) \cdot A + (2^{2k} - 1) \cdot B + (2^{2k} - 2^k) \cdot C\right|_{2^{3k}-2^k} \\ &= \left|(2^{2k+1} - 2^{2k} + 2^k) \cdot A + (2^{2k} - 1) \cdot B + (2^{2k} - 2^k) \cdot C\right|_{2^{3k}-2^k} \end{aligned} \qquad (10)$$

식(10)의 A, B, C에 필요한 비트수는 표 2와 같고, 이는 CSA 트리를 사용한 덧셈의 인접비트에 서로 영향을

끼치지 않으므로 병렬수행이 가능하다.

표 2. A, B, C의 소요비트수

근 거	소요비트수
$0 \le A = \left\|(2^{k-1}) \cdot x\right\|_{2^k-1} \le 2^k - 1$	k-1 비트
$0 \le B = \left\|(2^{k-1}-1) \cdot y\right\|_{2^k} \le 2^k - 1$	k-1 비트
$0 \le C = \left\|(2^{k-1}+1) \cdot z\right\|_{2^k+1} < 2^k$	k 비트

식(11)을 그림 1과 같이 적용할 수 있다.

$$|x|_{2^{3k}-2^k} = \left| \begin{array}{l} 2^{2k} \cdot B + (-2^k \cdot C) - B + 2^{2k} \cdot C \\ \qquad + 2^{2k+1} \cdot A + 2^k \cdot A \\ \qquad\qquad + (-2^{2k} \cdot A) \end{array} \right|_{2^{3k}-2^k} \qquad (11)$$

전체 x에 대한 모듈라이 2^{3k}-2^k를 적용하기 다음과 같은 간단한 알고리즘을 적용한다. 먼저 식(11)에서 x는 범위 $0 \le x < 3(2^{3k}-2^k)$내에 있으므로 다음과 같은 조건이 성립한다.

if $K = 2^{2k} \cdot B + (-2^k \cdot C) - B + 2^{2k} \cdot C \ge 2^{3k} - 2^k$

then $K = K - \underline{(2^{3k} - 2^k : CHK, CHK')}$ *endif*

if $K + 2^{2k+1} \cdot A + 2^k \cdot A + (-2^{2k} \cdot A) \ge 2^{3k} - 2^k$

then $K = K) + 2^{2k+1} \cdot A + 2^k \cdot A + (-2^{2k} \cdot A)$

$\underline{-(2^{3k} - 2^k : SUB)}$ *endif*

이렇게 CSA 트리는 그림 1과 같이 구성한다.

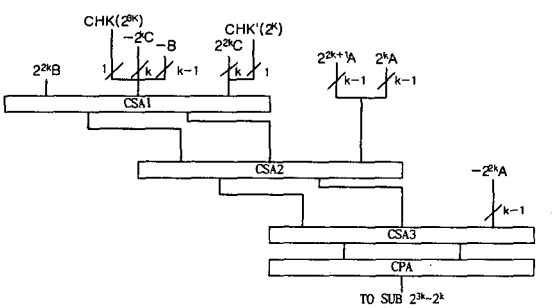

그림 1. CSA 트리 및 CPA

이 트리에서는 여러 값들이 인접 비트에 방해하지 않고 병렬적으로 수행될 수 있다는 이점을 가지게 된다. 여기서 CHK 룩업은 (2^{3k}-2^k)를 초과하는 경우에 0, 그렇지 않은 경우에는 1이 되도록 레지듀 비트연산을 수행한다. 또한 SUB 회로는 스트링 속성을 이용하여 최상위 비트가 1(2^{3k})일 경우, 이 값을 0으로하고 k+1 번째 비트가 1이 되도록 구성한다. 또 k+1~최상위에서 두번째 비트까지가 모두 1일 경우는 모두 0으로 클리어 시키고 나머지 다른 비트들은 관여하지 않는다. 그리고, 마지막에 2의 보수를 취한다.

4. 결 론

본 논문에서 제시하는 CRT 적용방법은 병렬수행과 캐리부재로 인해 빠른 연산을 수행한다. 이렇게 중첩 다중비트 주사기법을 적용하면 연산하고자 하는 비트수가 증가하더라도 단지 3번의 CSA 트리 단계를 넘어서지 않는다.

CRT를 이용하여 구현한 시스템의 속도는 빨라졌으나, 수천개이상의 디코딩회로 때문에 면적은 상당히 증가한다. 하지만, IC 집적기술의 발달로 이는 크게 문제시되지 않는다.

8비트 승산을 할 때를 예로 타승산기와 속도를 비교 하면, Braun 승산기의 속도는 $92\Delta_T$로 이미 알려져 있다[7].그리고, 중복 다중 비트 주사기법은 $76\Delta_T$가 소요된다. 본 논문의 방법을 적용하면 $61\Delta_T$로 현저히 빨라진다.

그러나, 마지막단에서 2의 보수에 의한 캐리 전달지연 시간은 본 논문에서도 여전히 존재한다. 그러므로 향후 연구과제로 이러한 캐리 지연시간을 줄이는 방법의 연구가 필요하다.

참 고 문 헌

[1] W. K. Jenkins, "Techniques for residue to analog conversion for residue encoded digital filters, " IEEE Trans, Circuits Syst., vol. CAS-25, pp 553-562, July 1978.

[2] A. A Sawchuk and T. C Strand, "Digital optical computing" Proc IEEE, Vol. 72, pp. 758-779, July 1982.

[3] H. M. Razavi, and J. Battelini., "Design of a residue arithmetic multiplier" IEE Proceedings-G, Vol. 139, No. 5, pp. 581-585, October 1992.

[4] Khalid M. Ibrahim., and Salam N. Saloum, "An efficient residue to binary converter design," IEEE Trans on Circuit and System, Vol., 35, No. 9, pp. 1156-1158, September, 1988.

[5] A. Hiasat "New designs for a sign detector and a residue to binary converter," IEE Proceeding, vol. 140, No. 4, pp. 247-252, August 1993.

[6] F. Pourbigharaz, Member, IEEE and H.M Yassine, "A Signed-Digit Architecture for Residue to Binary Transformation," Algorithmica, pp. 79-119, 3, 1998.

[7] Braun, E. L., Digital Computer Design, Academic Press, New York, 1963.