CRT와 중첩다중비트 주사기법을 접목한 승산기
(Multiplier Using CRT and Overlapped Multiple-bit Scanning Method)

김 우 완 † 장 상 동 ††
(Wu Woan Kim) (Sang-Dong Jang)

요 약 최근 레지듀 수체계를 기반으로 하는 컴퓨터 영상처리, 음성출력 등의 디지털 신호처리 하드웨어에 관한 연구가 고속저가의 하드웨어 구현에 크게 기여하고 있다. 본 논문에서는 모듈라이(2^k-1, 2^k, 2^k+1)를 사용하여 RNS에서 WNS로 WNS에서 RNS로 변환하는 방법을 통해 승산기를 설계 및 구현한다. 이는 CRT 변환을 중첩다중비트 주사기법을 접목한 시뮬레이션을 통해, 기존의 방법보다 속도가 빠르다는 것을 알 수 있고, 이는 RNS의 병렬처리와 캐리부재의 연산특성 때문임을 알 수 있다.

키워드 : CRT, 승산기, 중첩 다중 비트 주사 기법

Abstract Digital signal processing hardware based in RNS is currently considered as an important method for high speed and low cost hardware realization. This research designs and implements the method for conversion from a specific residue number system with moduli of the form (2^k-1, 2^k, 2^k+1) to a weighted number system. Then, it simulates the implementation using a overlapped multiple-bit scanning method in the process of CRT conversion. In conclusion, the simulation shows that the CRT method which is adopted in this research, performs arithmetic operations faster than the traditional approaches, due to advantages of parallel processing and carry-free arithmetic operation.

Key words : CRT, Multiplier, Overlapped Multiple-bit Scanning Method

1. 서 론

최근 RNS(Residue Number System)를 기반으로 하는 컴퓨터 영상 처리, 음성 출력 등의 디지털 신호 처리 하드웨어에 관한 연구가 고속 저가의 하드웨어 구현에 크게 기여하고 있다[1-3]. 그러나 범용 계산 컴퓨팅에 있어서는 RNS는 나눗셈, 크기비교, 부호나 오버플로우 검출 등의 어려움으로 광범위하게 사용되지 않았으나, 이러한 문제해결을 위해 여러 방법들이 제시되면서 다시 산술 연산기 구현에 있어 중요한 수체계로 인식되고 있다[4-6].

RNS에서 연산의 큰 이점으로는 RNS의 각 자리수가 서로 독립적이어서 병렬 수행과 캐리 부재(Carry-Free) 산술 연산이 가능하다는 것이다[1]. RNS에서 WNS (Weighted Number System)로 변환하기 위한 대표적인 두 가지 방법으로 Chinese Remainder Theorem (CRT)을 이용한 접근 방법과 Mixed Radix Conversion(MRC) 알고리즘을 사용하는 접근 방법이 있으며[8-10], 이 중 CRT가 중요 알고리즘으로 대두되고 있으며, 여기서 대표적으로 사용되는 모듈라이 집합은 2^k-1, 2^k, 2^k+1이다.

본 논문에서 제안하는 승산기는 모듈라이 집합 2^k-1, 2^k, 2^k+1를 사용하며, WNS에서 RNS로 디코딩하여 레지듀 곱셈을 수행한 후, CRT를 이용하여 RNS에서 WNS로 변환하도록 하였으며, 이때 CRT의 변환속도를 줄이고자 CRT식을 중첩다중비트 주사 기법을 접목할 수 있도록 유도하여 승산기의 구현에 적용한다. 이 모듈라이의 특성을 응용하여 기존의 CRT방법과 중첩다중비트 주사(Overlapped Multiple-Bit Scanning Method)기법을 접목함으로써 CSA(Carry Save Adder)의 단계를 줄일 수 있고, 뿐만 아니라 간단한 룩업 테이블과 캐리 전달 지연을 줄일 수 있는 회로를 구성함으로써 계산속도를 증가시킬 수 있다.

2장에서는 관련연구에 대해 간단히 기술하고, 3장에는

† 종신회원 : 경남대학교 정보통신공학부 교수
wukim@zeus.kyungnam.ac.kr
†† 비 회 원 : 경남대학교 컴퓨터공학과
angong@hawk.com.kyungnam.ac.kr
논문접수 : 2002년 3월 11일
심사완료 : 2003년 8월 29일

본 논문에서 제시한 방법에 사용될 CRT식을 유도해 보이고, 8비트 승산기를 예로 설명할 것이다. 마지막 4장에서는 전통적인 방법의 승산기와 본 논문에서 구현한 승산기를 게이트 수준의 면적과 속도를 중심으로 비교해 보일 것이다.

2. 관련연구

2.1 RNS의 표기

레지듀는 정수로된 N-튜플 { m_1, m_2, ···, m_N}로 구성된다. 이렇게 N-튜플로 나누어진 각각의 멤버를 모듈러스(modulus), 복수형은 모듈라이(moduli)라고 한다.

주어진 기수에 대해서, 정수 x의 레지듀 표현은 서로 다른 N-튜플{ r_1, r_2, ···, r_N }로 나타낸다. 여기서 정수 ri는 N개의 집합에 대한 수식으로 정의된다.

$$x = q_i m_i + r_i \quad i = 1,2,3,\cdots,N \tag{1}$$

그리고 q_i는 $0 \le r_i < m_i$를 선택하기 위한 정수이다. q_i는 $\frac{x}{m_i}$의 몫으로 정수이며 이것은 $\left[\frac{x}{m_i}\right]$라고 표기한다. r_i는 x를 m_i로 나눈 나머지로 양수가 된다. 그리고 이를 x mod m_i 또는 $|x|_{m_i}$로 표기하거나 x mod (m_i)로 표현하기도 한다. 정수 r_i는 x의 i번째 레지듀 자리수라고 부른다. 이에 대한 일반식은

$$x = m_i\left[\frac{x}{m_i}\right] + |x|_{m_i} \tag{2}$$

이다. x는 양수가 아닌 정수값을 가진다. x가 음수라고 하면 $\left[\frac{x}{m_i}\right]$ 또한 음수이다. 여기서 $|x|_{m_i}$는 반드시 양수값을 가져야 한다.

2.2 모듈라이의 선택과 부호 표현

레지듀 수를 표현하는 레지듀 자리수의 집합은 추가적으로 고려해야할 몇 가지 문제가 있다. 모듈라이의 선택, 부호를 표현하는 방법, 그리고 모듈라이와 부호 선택은 해결되어져야할 전형적인 문제이다.

레지듀 수를 나타내는 모듈라이의 선택은 레지듀 컴퓨터의 설계에서 매우 중요한 항목이다. 왜냐하면 계산속도와 하드웨어의 복잡도 때문이다. 이 두 가지 측면의 Trade-Off는 모듈라이의 선택을 어렵게 한다. 왜냐하면 MRC나 CRT의 경우는 변환시 속도를 줄일 수 있기 때문에 가능한 작은 갯수의 모듈라이를 선택해야 한다. 큰 모듈라이의 선택은 디코딩시에 하드웨어의 면적을 급격하게 증가시킨다. 그러므로 범위 M에 대해 보다 작은 값의 모듈라이를 선택하는 것이 면적을 작게 한다. 하지만, 본 논문에서는 면적보다는 속도를 빨리할 수 있는 모듈라이를 선택했다. 여기서 선택된 모듈라이는 상대적으로 쌍을 이루는 소수이다. 중첩 비트 주사 기법의 적용을 위해 모듈라이 2^k-1, 2^k, 2^k+1를 사용한다. 두 수를 곱한 결과를 N이라고 하면, N은 $3n$비트의 2진수로 표현 가능하고 그 범위는 $0 \le N < (2^n-1)2^n(2^n+1)$이다[11].

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

2.3 CRT와 CRT 적용 컨버터들의 구조

주어진 x의 레지듀 표현 $\{r_1, r_2, \cdots, r_N\}$에 대해, RNS에서 WNS로 변환하는 방법 중의 하나인 CRT는 $|x|_m$의 결정을 가능케 하고, 쌍을 이루는 모듈라이의 최대공약수는 1이다. 이러한 모듈라이는 상대적으로 쌍을 이루는 소수(Pairwise Relatively Prime)로 구성되며 식 (3)과 같다.

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

(단, $\hat{m}_j = \frac{M}{m_j}$, $M = \prod_{j=1}^{N} m_j$이고 $j \neq k$에 대해서 $(m_j, m_k) = 1$이다).

캐리의 부재, 결함의 격리, RNS 처리의 모듈라이 특성은 여러 논문에서 폭넓게 연구된바 있다. 디지털 신호처리(DSP)를 근간으로하는 장치의 이진수 표현에서 레지듀 표현으로의 변환은 수월하게 해낼 수 있으나, 변환처리의 오버헤드 때문에 레지듀에서 이진수로 변환하는 기술은 보다 주의를 요한다. 대표적인 방법으로 CRT와 MRC가 있으며 이중 CRT를 이용한 방법들에 대해 살펴보도록 한다.

논문[10]의 CRT를 기초로한 레지듀에서 이진수로 변환하는 구조는 모듈라이 집합 Sk를 사용하며, 다음과 같다.

$$S^k = \{2^m-1, 2^{2^0 m}+1, 2^{2^1 m}+1, 2^{2^2 m}+1, \ldots, 2^{2^k m}+1\}$$

모듈라이 집합 $S^0 = \{2^m-1, 2^m+1\}$와 $S^1 = \{2^m-1, 2^m+1, 2^{2m}+1\}$를 근거로 하는 레지듀에서 2진수로 변환하는 구조가 향상되어 졌으며, 변환 절차는 크게 다음과 같은 세 단계로 구성된다.

i) 레지듀에서 SD로 변환
ii) Signed-digit에서 2진수로 변환
iii) 엔드-어라운드 캐리 덧셈/뺄셈(end-around carry addition/subtraction)

첫 번째 절차는 CRT식의 SD표현이 리딘던트 덧셈기/뺄셈기 블럭에 사용에 의해 구현된다. 여기서, 내장된 곱셈은 간단한 시프트레프트 연산에 의해 재배치되고 캐리의 전달은 완전히 제거된다. 두 번째 절차는 CRT의 리딘던트 표현이 직접적으로 이진수 형태로 변환된

다. 마지막으로, 결과를 얻기 위해 엔드-어라운드 캐리(EAC) 덧셈/뺄셈을 수행한다.

논문[12]의 디코더는 세 모듈라이 집합 $\{2^n-1,\ 2^n,\ 2^n+1\}$를 기초로 한다. 이 디코더는 Unsigned-Magnitude 또는 2의 보수형태를 선택할 수 있다. 이 두 구조를 분석하면 첫 번째는 CSA를 이용하고 반면에 다른 구조는 모듈로 덧셈기(MA)를 이용한다. 이 레지듀 디코더에 대한 전체 블럭도는 그림 1과 같다.

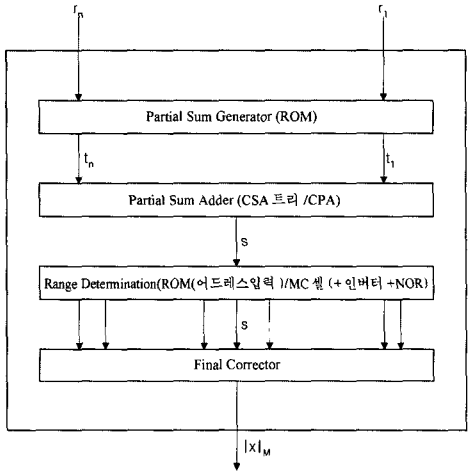

그림 1 레지듀 디코더 블럭 다이어그램

첫 번째 단계인 부분합 생성기(Partial Sum Generator)에서 ROM 주소 입력에 의해 부분합이 ROM에서 나온다.

$$t_i = \left| \hat{m}_i \left| \frac{r_i}{\hat{m}_i} \right|_{m_i} \right|_M$$

두 번째 단계인 부분합 덧셈기(Patial Sum Adder)가 부분합의 모듈로 M에 대한 전체합을 구한다. 이 단계의 계산을 위해 N-2번의 CSA트리 단계와 CPA단계가 필요하다. 세 번째 단계인 범위 결정기(Range Determinator)에서는 세 레벨이 필요하다. ROM과 크기 비교기(MC), 비트 교정자(BC)가 그것이다. 여기서 CRT식의 마지막 모듈라이 M을 적용한다. 그리고 마지막 교정자(Final Corrector)에서는 멀티플렉서를 이용하여 원하는 값을 선택하고 CLA에서 상수 지연을 가진다.

2.4 중첩다중비트 주사기법

중첩다중비트 주사기법(Overlapped Multiple-Bit Scanning Method)은 승수의 제로 스트링을 시프트하면 실행시간을 감소시킬 수 있다는 개념을 바탕으로 한다. 그러므로, 승수에서의 제로의 수는 승산이 필요로 하는 시간 단축과 비례하게 된다. 다음과 같이 승수에서 1이 k회 연속적으로 존재하는 스트링을 고려해 보자.

칼럼위치:	$\cdots$	$i+k$	$i+k-1$	$i+k-2$	$\cdots$	i	$i-1$	$\cdots$
비트값:	$\cdots$	0	1	1	$\cdots$	1	0	$\cdots$

i번째 비트부터 $i+k-1$번째 비트까지 1이 연속이다. 다음의 스트링 특성에 의해서

$$2^{i+k}-2^i = 2^{i+k-1}+2^{i+k-2}+\cdots+2^{i+1}+2^i$$

가 된다. 연속적인 1에 대해서 다음과 같이 재배치할 수 있다.

칼럼위치:	$\cdots$	$i+k$	$i+k-1$	$i+k-2$	$\cdots$	i	$i-1$	$\cdots$
비트값:	$\cdots$	1	0	0	$\cdots$	$\bar{1}$	0	$\cdots$

상위 비트 대열에 존재하는 1은 가산인 $+a$를, 하위 비트 대열에 존재하는 $\bar{1}$은 수행하여야 하는 감산인 $a-1$을 의미한다. 이러한 승수의 재코딩을 동원하면 k회의 연속적인 가산을 스트링 시작부에서의 한번의 가산과 종결부에서의 한번의 감산으로 대체할 수 있어, 특히 k의 값이 큰 경우에 필요한 가산 시간을 상당 부분 감소시킬 수 있게 된다. 이러한 스트링의 속성이 중첩다중비트 주사기법의 장점이다.

3. CRT와 중첩다중비트 주사기법을 접목한 승산기의 설계 및 구현

이장에서는 먼저 전체적인 승산의 단계별 수행순서와 승산기의 구조를 먼저 제시한 후, 본 논문의 핵심이 되는 CRT변환의 하드웨어적인 적용을 위해, 기존의 CRT식을 새로이 변형된 형태의 식으로 유도하는 과정을 보이고 난 뒤, 각 단계별로는 세부 설계 및 구현내용을 함께 설명하도록 한다. 먼저 본 논문의 승산기는 일반 PC에서 OR-CAD를 사용하여 게이트레벨로 구현하였으며, PC의 세부환경은 표. 1과 같다.

표 1 승산기의 구현환경

PC 환경	내용
CPU	333MHz
메모리	64M
운영체제	Windows98
시뮬레이션툴	OR-Cad 7.1

본 승산기는 모듈라이 집합 $2^k-1,\ 2^k,\ 2^k+1$를 사용하며 다음과 같은 단계별 처리에 의해 승산이 수행된다.

- 1단계 : 입력된 두 수를 레지듀로 디코딩.
- 2단계 : 각 모듈라이의 레지듀의 곱의 계산.
- 3단계 : CRT를 적용해 레지듀에서 이진수로 변환.
- 4단계 : 2의 보수로 변환하여 결과출력.

위의 단계 중 세 번째 단계에서 사용되는 CRT의 적

용방법이 본 논문에서 제시한 하드웨어 구조의 핵심이다. 이 식의 유도는 3.1절에서 설명하도록 하며, CRT식의 유도는 8비트, 16비트, 32비트 등의 2의 멱급승으로 이루어진 비트수 만큼의 승산기 구현이 가능하겠으며, 본 논문에서는 8비트 승산기를 예로 설명하도록 한다. 그림 2는 8비트 승산기의 구조이다.

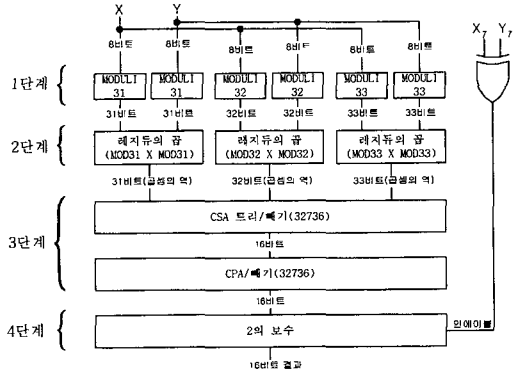

그림 2 승산기 전체 구조

3.1 제안된 CRT 변환식

$$|N|_M = \left| \sum_{j=1}^{N} \hat{m}_j \left| \frac{r_j}{\hat{m}_j} \right|_{m_j} \right|_M \tag{4}$$

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

RNS에서 WNS로의 변환은 CRT를 이용한다. 이때의 CRT는 식 (4)와 같으며, 풀어서 쓰면 식 (5)와 같이 나타낼 수 있다.

$$|N|_M = \left| \hat{m}_1 \cdot \left| \frac{1}{\hat{m}_1} \cdot x \right|_{m1} + \hat{m}_2 \cdot \left| \frac{1}{\hat{m}_2} \cdot y \right|_{m2} + \hat{m}_3 \cdot \left| \frac{1}{\hat{m}_3} \cdot z \right|_{m3} \right|_M \tag{5}$$

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

여기서 M에 대해 세 모듈라이 $m_1 = 2^k - 1$, $m_2 = 2^k$, $m_3 = 2^k + 1$ 적용하면 식 (6)과 같이 된다.

$$M = \prod_{i=1}^{3} m_i = (2^k - 1) \times 2^k \times (2^k + 1) = 2^{3k} - 2^k \tag{6}$$

$k \geq 2$라는 조건하에서 각 $|\hat{m}_1|$, $|\hat{m}_2|$, $|\hat{m}_3|$의 값을 다음과 같이 유도할 수 있다.

$$\begin{aligned}
|\hat{m}_1|_{m_1} &= |2^k(2^k+1)|_{2^k-1} \\
&= |2^k\{(2^k-1)+2\}|_{2^k-1} \\
&= |2^k(2^k-1)+(2^k)2|_{2^k-1} \\
&= |2^k(2^k-1)|_{2^k-1} + |(2^k)2|_{2^k-1} \\
&= 0 + |(2^k-1+1)2|_{2^k-1} \\
&= |(2^k-1)2+2|_{2^k-1} \\
&= |2|_{2^k-1} \\
&= 2(k \geq 2)
\end{aligned}$$

$$\begin{aligned}
|\hat{m}_2|_{m_2} &= |(2^k-1)(2^k+1)|_{2^k} \\
&= |2^k 2^k - 1|_{2^k} \\
&= |2^k 2^k|_{2^k} + |-1|_{2^k} \\
&= 0 + |2^k - 1|_{2^k} \\
&= 2^k - 1
\end{aligned}$$

$$\begin{aligned}
|\hat{m}_3|_{m_3} &= |(2^k-1)2^k|_{2^k+1} \\
&= |(2^k+1-2)(2^k+1-1)|_{2^k+1} \\
&= |(2^k+1)^2 + 3(2^k+1) + 2|_{2^k+1} \\
&= 0 + 0 + |2|_{2^k+1} \\
&= 2(k \geq 2)
\end{aligned}$$

위의 유도식에 의해서

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

로 나타낼 수 있고, 이 각각에 대한 곱셈의 역을 구하면 식 (7)과 같은 결과를 얻을 수 있을 것이다.

$$\left| \frac{1}{\hat{m}_1} \right|_{2^k-1} = \left| \frac{1}{2} \right|_{2^k-1} = 2^{k-1}, \left| \frac{1}{\hat{m}_2} \right|_{2^k} = \left| \frac{1}{2^k-1} \right|_{2^k} = 2^k - 1$$

$$\left| \frac{1}{\hat{m}_3} \right|_{2^k+1} = \left| \frac{1}{2} \right|_{2^k+1} = 2^{k-1} + 1 \tag{7}$$

식 (7)의 결과를 식 (5)에 대입하고 식을 변형하여 보도록 한다.

$$\begin{aligned}
|N|_M &= \left| \hat{m}_1 \cdot \left| \frac{1}{\hat{m}_1} \cdot x \right|_{m1} + \hat{m}_2 \cdot \left| \frac{1}{\hat{m}_2} \cdot y \right|_{m2} + \hat{m}_3 \cdot \left| \frac{1}{\hat{m}_3} \cdot z \right|_{m3} \right|_M \\
&= \left| 2^k(2^k+1) \cdot |2^{k-1} \cdot x|_{2^k-1} \right. \\
&\quad + (2^k-1)(2^k+1) \cdot |(2^k-1) \cdot y|_{2^k} \\
&\quad \left. + (2^k-1)(2^k) \cdot \left| (2^{k-1}+1) \cdot z \right|_{2^{k+1}} \right|_{2^{3k}-2^k}
\end{aligned} \tag{8}$$

여기서, A,B,C를 식 (9)와 같다고 하면 식 (8)은 식 (10)과 같이 변형할 수 있을 것이다.

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

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

식 (10)에 있는 각 A, B, C에 곱해지는 값들을 보면 스트링 특성을 가졌음을 알 수 있으며, 이 식을 승산기에 적용한다.

3.2 두 입력의 레지듀 변환 및 승산

그림 2의 첫 번째 단계에서 두 이진수 입력 X,Y에 대해 모듈라이 2^k-1, 2^k, 2^k+1에 해당하는 디코딩 회로를 구성하여 RNS로 바꾸고 두 번째 단계에서 두 수를 곱한다. 두 수 X,Y를 곱한 결과를 N이라고 하면, N은 $3n$비트의 2진수로 표현 가능하고 이 수의 범위는 $0\le N<(2^n-1)(2^n)(2^n+1)$이다[11].

8비트 승산기의 경우 결과 값의 그 범위가 $-2^8\times(2^8-1)\sim-2^8\times-2^8$이고, 이는 15비트로 표현이 가능하며. $15=3n$, $n=5$가 되므로 모듈라이는 $\{2^k-1,\ 2^k,\ 2^k+1\}=\{31,32,33\}$이 된다. 2의 보수 입력의 양수, 음수 변환은 이 디코딩 회로를 통해 제거되며, 최초입력의 최상위 부호비트는 곱셈의 결과로 양수 또는 음수인지 결정하여 그림 2의 네 번째 단계에서 2의 보수를 취하기 위한 선택선으로 사용된다.

3.3 제안 방법의 적용

곱셈의 역은 다음과 같이 구할 수 있다. $m_1=31$, $m_2=32$, $m_3=33$에 대해

$$\begin{aligned}M&=\prod_{i=1}^{3}m_i=31\times32\times33=32736\\\left|\hat{m}_1\right|_{m_1}&=\left|32\times33\right|_{31}=\left|1056\right|_{31}=2,\\\left|\hat{m}_2\right|_{m_2}&=\left|31\times33\right|_{32}=\left|1023\right|_{32}=31,\\\left|\hat{m}_3\right|_{m_3}&=\left|31\times32\right|_{33}=\left|992\right|_{33}=2\end{aligned} \tag{11}$$

로 나타낼 수 있고 이 각각의 모듈라이에 대한 곱셈의 역은 다음과 같이 16, 31, 17이다.

$$\left|\frac{1}{\hat{m}_1}\right|_{31}=\left|\frac{1}{2}\right|_{31}=16,\quad\left|\frac{1}{\hat{m}_2}\right|_{32}=\left|\frac{1}{31}\right|_{32}=31,\quad\left|\frac{1}{\hat{m}_3}\right|_{33}=\left|\frac{1}{2}\right|_{33}=17$$

곱셈의 역이 상수로 정의됨에 따라 실제 구현에 있어서는 몇 비트정도 시프트만 시키면 된다. 그림 2의 2단계에서 3단계로 넘어가는 과정에서의 회선의 연결로 해결된다.

위 (11)에서 구한 값을 식 (10)에 대입하면 (12)와 같이 된다.

$$\left|N\right|_{32736}=\left|(1056\times\left|16\times x\right|_{31})+(1023\times\left|31\times y\right|_{32})+(992\times\left|17\times z\right|_{33})\right|_{32736} \tag{12}$$

식 (12)의 CRT식은 식 (10)에서 확인할 수 있듯이 $\left|16\times x\right|_{31}$, $\left|31\times y\right|_{32}$, $\left|17\times z\right|_{33}$를 각각 A, B, C라고 두었을 때, 1056, 1023, 992의 2진수 표현의 중첩다중비트 주사기법을 적용한 값은 표 2와 같고, 이를 적용한 것이 그림 3이다.

A,B,C는 부호없는 이진수로 변환하고, 이때의 값의

표 2 1056, 1023, 992에 대한 중첩다중비트 주사기법적용

계산값 / 실제값	스트링 특성	적용	적용식
1056A	2A, 0, 0 / 0 0 1 0 0 0 0 1 0 0 0 0 0 / -4A, 2A, 0	$2^{10}\times2A=2^{11}A$ $2^4\times2A=2^5A$ $2^8\times-4A=-2^{10}A$	2048A - 1024A + 32A
1023B	0, 0 / 0 1 1 1 1 1 1 1 1 1 1 / 4B, 0, 0, -B	$2^8\times4B=2^{10}B$ $-B$	1024B - B
992C	0, 0 / 0 1 1 1 1 1 0 0 0 0 0 / 4C, -2C, 0	$2^4\times-2C=-2^5C$ $2^8\times-4C=2^{10}C$	1024C - 32C

표 3 레지듀 값의 이진수 표현 비트수

A, B, C의 값의 범위	표현 비트의 수
$0 \le A = \left\|(2^{k-1}) \cdot x\right\|_{2^k-1} - 1 \le 2^k - 2$	k
$0 \le B = \left\|(2^{k-1}-1) \cdot y\right\|_{2^k} - 1 \le 2^k - 1$	k
$0 \le C = \left\|(2^{k-1}+1) \cdot z\right\|_{2^k+1} - 1 < 2^k$	k +1

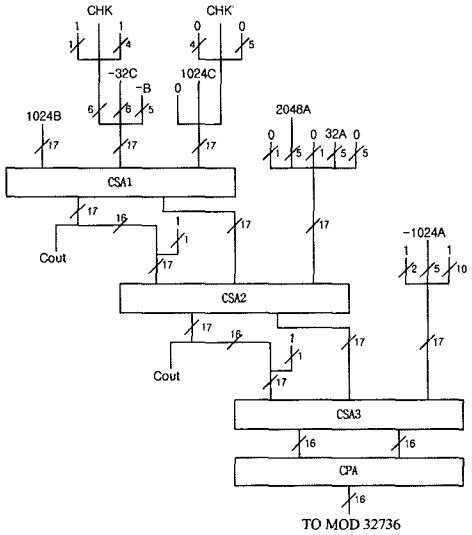

그림 3 CSA트리 및 CPA의 블럭도

범위 및 사용비트의 수는 표 3과 같다. 그러므로 8비트 승산기에서의 A, B, C는 각각 5, 5, 6비트로 표현한다.

위의 식 (10)의 $|N|_M$에서 마지막 나눗셈 연산의 피젯수가 되어지는 M이 $N<3M$의 조건을 만족하기 때문에 N이 M보다 2배를 넘을 경우가 있을 수 있다. 이때, 실제 나눗셈 연산을 행하는 대신에 최대로 두 번까지, M $(= 2^{3k} - 2^k)$의 크기만큼의 감산을 수행함으로써 값의 정확성을 유지할 수 있다. 여기서 감산의 수행은 두 가지 방법으로 처리되며, 그 중 한가지는 중첩다중비트 주사기법의 핵심인 스트링 특성을 이용하여 간단한 비트의 조작으로 이루어질 수 있다. 그림 3의 CHK와 CHK'연산이 그것이다. 또, 하나는 SUB블럭에서 감산을 수행하는 것이다. 이 둘은 아래와 같은 조건을 만족할 때 수행할 수 있도록 한다. 아래의 두 식에서 밑줄친 부분이 각각의 회로가 수행할 내용이다.

Look-Up 회로 :

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

SUB블럭 회로 :

$$if \quad K + 2^{2k+1} \cdot A + 2^k \cdot A + (-2^{2k} \cdot A) \ge 2^{3k} - 2^k$$
$$then \quad K = K + 2^{2k+1} \cdot A + 2^k \cdot A + (-2^{2k} \cdot A) - \underline{(2^{3k} - 2^k : SUB)} \quad endif$$

여기서 Look-Up은 중첩다중비트 주사기법의 스트링 특성을 이용하여 CSA트리의 입력단에 CHK의 신호를 사용함으로써 감산기 역할을 수행하며, SUB 블록은 CPA에서 나온 결과에 스트링 특성을 이용하여 또 한번의 감산을 선택적으로 할 수 있다. 식 $|N|_M$에서 N<3M의 조건을 만족하기 때문에 최대 두 번의 감산으로 값의 정확성을 유지할 수 있다.

4. 결론 및 향후 연구과제

본 논문에 제안하고 있는 방법은 CRT변환에 있어서 중첩다중비트 주사기법의 스트링 특성을 이용함으로써 CSA트리의 단계가 3으로 고정된다. 하지만 여전히 CPA의 상주 지연이나 2의 보수를 취하기 위한 캐리 전달 지연시간이 여전히 존재한다.

8비트 승산기를 기준으로 게이트의 면적과 속도를 기준으로 기존의 승산기와 비교하면 표 4와 같은 결과를 보이며, 면적의 경우는 디코딩 회로로 인해 상당한 증가를 보이나 이는 IC 집적기술의 발달로 크게 문제되지 않으며, 속도면에서는 전통적인 승산기보다 빠르다는 것을 확인할 수 있다. 여기서 Braun 승산기의 속도는 이미 알려져 있고[11], 중첩다중비트 주사기법의 승산기[13]도 첫단과 마지막단에서의 2의 보수로 인한 지연속도 CAS트리, CPA, 피승수에 대한 배수 0, ±2, ±4를 위한 회로, XOR 지연시간이 필요하다.

본 논문에서도 2의 보수에 의한 캐리 전달지연 시간이 여전히 존재한다. 향후 연구과제로 이러한 캐리 지연시간을 줄이는 방법의 연구가 필요하며, 캐리가 발생하

표 4 8비트 승산기를 기준으로 한 기존 승산기와의 비교

면적			속도		
구현 승산기	중복다중 비트주사[13]	Braun 승산기[11]	구현 승산기	중복다중 비트주사[13]	Braun 승산기[11]
4705 Ag	1185 Ag	626 Ag	61 Δ_T	76 Δ_T	92 Δ_T

지 않는 새로운 방법의 연구가 지속적으로 필요하다.

참 고 문 헌

[1] K. Elleithy and M. Bayoumi, "A Systolic Architecture for modulo Multiplication," *IEEE Trans Circuit and System-II: Analog and Digital Signal Processing*, vol. 42, pp. 725~729, Nov. 1995.

[2] 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.

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

[4] Ahmad A. Hiasat, "New Efficient Structure for a Modular Multiplier for RNS," *IEEE Trans Computers*, vol. 49, pp. 170~174, Feb. 2000.

[5] J. Bajard, L. Didier, and P. kornerup, "An RNS Montgomery Modular Multiplication Algorithm," *IEEE Trans Computers*, vol. 47, pp. 766~776, July. 1998.

[6] A. Wrzyszcz, D. Milford, and E. Dagless, "A New Approach to fixed-Coefficient Inner Product Computation over Finite Rings," *IEEE Trans Computers*, vol. 45, pp. no. 12, pp. 1,345~1,355, Dec. 1996.

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

[8] 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.

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

[10] F. Pourbigharaz, Memver, IEEE and H.M Yassine, "A Signed-Digit Architecture for Residue to Binary Transformation," *Algorithmica*, pp. 79-119, 3, 1988.

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

[12] Khaled M. Elleithy, *Member, IEEE*, and Magdy A. Bayoumi, Senior Member, *IEEE*, "Fast and Flexible Architectures for RNS Arithmetic Decoding," *IEEE Transaction on Circuits and Systems*, vol. 39, No. 4, pp. 226-235, April 1992.

[13] Kai Hwang, "Computer Arithmetic Principles, Architecture, and Design," School of Electrical Engineering Purdue University, 1979.

김 우 완

1982년 경북대학교 전자공학과 졸업(공학사). 1990년 Texas A&M 대학교 전기전자공학과 졸업(공학석사). 1995년 Texas A&M 대학교 전기전자공학과 졸업(공학박사). 1996년~현재 경남대학교 정보통신공학부 부교수. 관심분야는 고속컴퓨터구조, 컴퓨터 네트워크, 모바일 컴퓨팅, 멀티미디어통신

장 상 동

1997년 경남대학교 전산통계학과(이학사). 1999년 경남대학교 대학원 컴퓨터공학과(석사). 2001년~현재 경남대학교 컴퓨터공학과 박사 수료. 관심분야는 컴퓨터구조, 컴퓨터네트워크(모바일 컴퓨팅), 인터넷 프로토콜