임베이드 컴퓨터시스템에 기초한 고효율 연산시스템 구성

°이택근*, 박춘명**
* 충주대학교 대학원 컴퓨터공학과
** 충주대학교 전기·전자 및 정보공학부 컴퓨터공학전공

A Construction of High Efficiency Arithmetic System based on Embedded Computer System

°Taek-Keun Lee* and Chun-Myoung Park**
* Dept. of Computer Engineering, Graduate School, Chungju Nat'l University
** Division of Electrical Electronic & Information Engineering, Computer Engineering Major, Chungju Nat'l University
E-mail : cmpark@gukwon.chungju.ac.kr

요 약

일반적으로 멀티미디어 하드웨어는 지금까지의 각종 데이터 처리보다는 훨씬 방대한 데이터양, 최적의 데이터 압축 및 복원, 초고속 전송, 암호화 및 복호화 등의 복합적이고 고기능의 기술이 요구되고 있다. 따라서 본 논문에서는 이러한 점들을 수행 할 수 있는 방법으로 최근에 그 활용도가 높아지고 있는 임베이드 컴퓨터시스템에 기반을 둔 고효율 연산시스템 구성의 한 가지 방법을 제안하였다.

1. 서론

최근에 멀티미디어 H/W와 S/W에 기반을 둔 여러 분야가 매우 급속도로 발전되고 있으며 21C에는 더욱 더 활용 및 적용이 요구될 것이다. 특히 멀티미디어 H/W는 지금까지의 각종 데이터 처리보다는 훨씬 방대한 데이터량, 최적의 데이터 압축 및 복원, 초고속 전송, 암호화 및 복호화 등의 복합적이고 고기능의 기술이 요구되고 있다.[1-5]

따라서 본 논문에서는 이를 해결할 수 있는 방법으로 최근에 그 활용도가 높아지고 있는 임베이드 컴퓨터시스템(Embedded Computer System)[6-8]에 기초하여 각종 멀티미디어 H/W 시스템에 기본적으로 사용되는 연산을 효율적으로 수행할 수 있는 고효율 연산시스템의 한 가지 방법을 제안하였다.

본 논문의 서술과정은 다음과 같다. 2장에서는 본 논문에서 사용되는 수학적 성질을 논의하고, 3장에서는 사칙연산에 대한 각각의 연산 알고리즘의 도출에 대해 논의하였다. 그리고 4장에서는 3장의 각각의 연산을 선택하기 위한 선택기의 구성에 대해 논의하였으며, 5장에서는 3장과 4장의 내용을 토대로 최종 임베이드 컴퓨터시스템에 기반을 둔 고효율 연산시스템 구성에 대해 기술하였다. 그리고 마지막 6장의 결론에서는 본 논문에서 제안한 고효율 연산시스템의 구성과 특징을 요약하였으며 앞으로의 전망을 기술하였다.

2. 수학적 배경

본 장에서는 본 논문을 전개하는데 필요한 유한체상의 수학적 성질을 논의한다.

[성질1] 다음 식(2-1)을 인수분해하여 m차 기약다항식을 구한 후 이를 0으로 하는 한근을 ξ라 할 때 식(2-2)와 같은 원시기약다항식을 얻을 수 있다.

$$X^{\mu}-X=X(X-1)(X^{\mu-2}+X^{\mu-3}+ \ldots\ldots +X+1)=0 \qquad (2\text{-}1)$$

where, $\mu=P^m$, P is prime number, m is integer.

$$F(\xi)=\sum_{i=0}^{m-1}\delta_i\xi^i$$
$$=\delta_0+\delta_1\xi^1+\ldots\ldots+\delta_{m-2}\xi^{m-2}+\delta_{m-1}\xi^{m-1} \qquad (2\text{-}2)$$

한편, 식(2-2)를 벡터공간(Vector Space)으로 표시하면 다음 식(2-3)과 같으며 특히 P=2인 경우를 비트벡터공간(Bit Vector Space)이라 한다.

$$\underline{F(\xi)}=[\delta_0\delta_1\ldots\ldots\delta_{m-2}\delta_{m-1}] \qquad (2\text{-}3)$$

여기서 $\delta_i(i=0,1,\ \ldots\ldots\ ,m-1)$는 ξ^i의 계수이다.

[성질2] $GF(P^m)$상에서 원소생성은 식(2-2)를 0으로 하는 한 근 ξ의 멱승으로 구해지며, 원소의 개수는 P^m이다.

이를 식으로 표시하면 다음 식(2-4)와 같다.

$$GF(P^m)=\{0,\xi^1,\xi^2,\ldots\ldots,\xi^{\mu-2}=\xi^{-1},\xi^{\mu-1}=1\} \qquad (2\text{-}4)$$

where, $\mu=P^m$

[성질3] 역원의 존재

(1) $\theta+(-\theta)=0$인 가법에 대한 역원 $-\theta$가 존재한다. $(\forall\theta\in GF(P^m))$

(2) $\theta\cdot(\theta^{-1})=1$인 승법에 대한 역원 θ^{-1}이 존재한다.($(\forall\theta\in GF(P^m))$

[성질4] $(\theta+\psi)^{\mu}=\theta^{\mu}+\psi^{\mu}=\theta+\psi \quad (\forall\theta,\psi\in GF(P^m))$

[성질5] $\theta^i\theta^j=\theta^{i+j(\mathrm{mod}\ \mu-1)} \quad (\mu=P^m)$

여기서 $r=i+j$이라 하면 $i+j(\mathrm{mod}\ \mu-1)$은 $r(\mathrm{mod}\ \mu-1)$이며 $0\leq r\leq P^m-1$이다.

이상의 수학적 성질과 그 외의 본 논문을 전개하는데 필요한 수학적 성질은 참고문헌[9-11]을 참고하였다.

3. 연산 알고리즘

본 논문에서는 유한체 $GF(2^m)$상에서 m=1인 경우를 사용하며 이는 현존의 디지털시스템의 근간이 되며 수학적 개념으론 부울대수(Boolean Algebra)가 도입된다.

3-1. 가산 연산 알고리즘

피가산원소를 e_i, 가산원소를 e_j, 가산후원소를 e_a라 하고 이들을 비트벡터공간으로 표현한 것을 각각 $\underline{e_i}(av)$, $\underline{e_j}(bv)$, $\underline{e_a}(Av)$라 하면 두 원소 e_i와 e_j의 가산은 다음 식(3-1)과 같다.

$$e_i\oplus e_j=\underline{e_i}(av)\oplus\underline{e_j}(bv)=\underline{e_a}(Av) \qquad (3\text{-}1)$$

여기서 $i,j=0,1,\ldots\ldots,2^m-2,2^m-1$이고, $av,bv,Av\in GF(2)$ $(V=0,1,\ldots\ldots,m-2,m-1)$이고, $\oplus$는 mod2 합이다.

위 식(3-1)에서 살펴 본 바와 같이 $Av=av\oplus bv$ 이다.

따라서 bv를 가산연산시의 제어입력으로 사용하면 bv 값에 따라 av 값을 그대로 유지하거나 2의 보수를 취한 값이 되고, 이는 mod2 합의 수학적 성질과 같다. 이 내용을 식으로 표시하면 식(3-2)와 같다.

$$Av= av \text{ iff } bv=0$$
$$av' \text{ iff } bv\neq0 \qquad (3\text{-}2)$$

여기서, $av,\ bv,\ Av\in GF(2)\ (V=0,1,\ldots\ldots,m-2,m-1)$이다.

위 내용과 식(3-1) 및 (3-2)를 토대로 가산알고리즘을 도출하면 다음과 같다.

[STEP1] 피가산원소 e_i와 가산원소 e_j를 각각 비트벡터공간으로 표현한 $\underline{e_i}(av)$와 $\underline{e_j}(bv)$로 표시한다.

[STEP2] 가산원소 $\underline{e_j}(bv)$를 제어입력으로 사용하여 bv의 값이 "0"이면 피가산원소 $\underline{e_i}(av)$의 해당 비트값을 그대로 유지하고 "1"이면 2의 보수를 취한다.

[STEP3] STEP2를 행한 후의 결과가 최종 가산후의 원소 $\underline{e_a}(Av)$가 된다.

3-2. 감산 연산 알고리즘

Mod2의 수학적 성질에 의해 감산은 가산과 같다. 따라서 $GF(2^m)$상에서 두 원소간의 감산 알고리즘은 가산알고리즘과 같다.

3-3. 승산 연산 알고리즘

피승산원소를 e_i, 승산원소를 e_j, 승산후원소를 e_M이라 하고 이들을 비트벡터공간으로 표현한 것을 각각 $\underline{e_i}(av)$, $\underline{e_j}(bv)$, $\underline{e_M}(Mv)$라 하면 두 원소 e_i와 e_j의 승산은 다음 식(3-3)과 같다.

$$e_i\otimes e_j=\underline{e_i}(av)\otimes\underline{e_j}(bv)=\underline{e_m}(Mv) \qquad (3\text{-}3)$$

한편, 피승산원소, 승산원소, 승산후원소의 기약다항식을 각각 $F(\xi)=\sum_{i=0}^{m-1}a_i\xi^i$, $G(\xi)=\sum_{j=0}^{m-1}b_j\xi^j$와 $H(\xi)=\sum_{k=0}^{m-1}M_k\xi^k$라 하면 식(3-3)은 다음 식(3-4)와 같이 표현 할 수 있다.

$$F(\xi)\otimes G(\xi)=\left(\sum_{i=0}^{m-1}a_i\xi^i\right)\otimes\left(\sum_{j=0}^{m-1}b_j\xi^j\right)$$
$$=\sum_{i=0}^{m-1}\left(\sum_{j=0}^{m-1}b_j\right)a_i\xi^{i+j}$$
$$=\sum_{i+j=0}^{2m-2}a_ib_j\xi^{i+j} \qquad (3\text{-}4)$$

여기서 $a_i,\ b_j\in GF(2)$이고 $i,j=0,1,\ldots\ldots,m-2,m-1$이다.

한편, 여기서 $r=i+j$라 하면 식(3-4)는 식(3-5)와 같고

이는 H(ξ)와 같아야 한다.

$$F(\xi)\otimes G(\xi)=\sum_{r=0}^{2m-2} a_i b_j \xi^r$$

$$= H(\xi)=\sum_{k=0}^{m-1} M_k \xi^k \qquad (3\text{-}5)$$

따라서 ξ^r의 r은 $m\le r1\le 2m-2$ 부분과 $0\le r2\le m-1$ 부분으로 분할 할 수 있으며 ξ^{r1}항을 수학적 성질로부터 ξ^{r2}항으로 표현하여 ξ^k항과 일치시킬 수가 있다.

또한, 이들 ξ^{r2}항들이 승산기 모듈 중 제어입력생성 모듈의 입력이 되고 이 제어입력 C_L에 의해 최종 승산후원소 $e_m(M_V)$를 얻는다.

이제 위 식(3-3)과 식(3-4)를 토대로 승산 알고리즘을 도출하면 다음과 같다.

[STEP1] 피승산원소 e_i와 승산원소 e_j를 각각 비트벡터공간으로 표현한 $e_i(av)$와 $e_j(bv)$로 표시한다.

[STEP2] 식(3-4)와 같이 ξ^{r1}항과 ξ^{r2}항을 각각 구한다.

[STEP3] ξ^{r1}항을 수학적 성질로부터 ξ^{r2}항으로 표현하여 제어입력 C_L을 구한다.

[STEP4] STEP3에서 구한 C_L의 값이 "0"이면 해당 ξ^{r2}항의 비트값을 그대로 유지하고 "1"이면 2의 보수를 취한다.

[STEP5] STEP4를 행한 후의 결과가 최종 승산후원소 $e_m(M_V)$가 된다.

한편, 제어입력 $C_L(L=0,1,\ldots\ldots,m-2,m-1)$은 식(3-5)의 ξ^{r1}항으로부터 구할 수 있다.

즉, ξ^{r1}을 ξ^{r2}항으로 표현해 이들을 mod2 합함으로써 쉽게 구할 수 있으며 이를 식으로 표현하면 다음 식(3-6)과 같고 제어입력 C_L의 개수는 m개이다.

$$\sum_{r1=m}^{2m-2} R_{r1}\xi^{r1}=\sum_{r1=m}^{2m-2} R_{r1}\left(\sum_{L=0}^{m-1}\xi^{L}\right) \qquad (3\text{-}6)$$

따라서 승산 연산은 ξ^r생성 부분과 제어입력C_L생성 부분을 합성하여 구할 수 있다.

즉, C_L의 값이 "0"이면 식(3-5)의 M_k 값은 R_{r2} 값을 유지하고 "1"이면 R_{r2} 값에 2의 보수를 취한 값이된다.

이를 식으로 표현하면 다음 식(3-7)과 같다.

$$M_k = \begin{cases} R_{r2} & \text{iff } C_L=0 \\ R_{r2}' & \text{iff } C_L=1 \end{cases} \qquad (3\text{-}7)$$

여기서 M_k, C_L, R_{r2}, $R_{r2}' \in GF(2)$이고 k, L, $r2=0,1,\ldots\ldots,m-2,m-1$이다.

그런데 식(3-7)은 가산 연산을 표현한 식(3-2)와 수학적으로 내용이 동일하다.

따라서 이 부분은 가산 연산을 그대로 사용할 수 있다.

3-4. 제산 연산 알고리즘

피제산원소, 제산원소 및 제산후원소를 각각 $e_i(av)$, $e_j(bv)$, $e_d(D_V)$라 하면 두 원소 e_i와 e_j의 제산은 다음 식(3-8)과 같다.

$$e_i\ominus e_j=e_i(av)\ominus e_j(bv)=e_i\otimes e_j^{-1}$$

$$=e_i(av)\otimes=e_d(D_V) \qquad (3\text{-}8)$$

여기서 e_j^{-1}은 e_j의 역원이며 bv^*는 bv에 대한 역원비트벡터공간이다.

한편, 피제산원소, 제산원소 및 제산후원소의 원시기약다항식을 각각 F(ξ), G(ξ)와 H(ξ)라 하면 식(3-8)은 다음 식(3-9)와 같다.

$$F(\xi)\ominus G(\xi)=F(\xi)\otimes[G(\xi)]^{-1}=H(\xi) \qquad (3\text{-}9)$$

여기서 $[G(\xi)]^{-1}$은 G(ξ)에 대한 승법역원생성다항식(multiplicative inverse element generation polynomial)이다.

또한, 2장의 성질2로 부터 $[G(\xi)]^{-1}$는 식(3-10)과 같다.

$$[G(\xi)]^{-1}=\left(\sum_{j=0}^{m-1} b_j\xi^j\right)^{K-2}(\text{where, } K=2^m) \qquad (3\text{-}10)$$

한편, 식(3-10)에서 $Q=2^m-2$라 하면 다음 식(3-11)과 같이 자승(Squaring) 형태로 분할 할 수 있다.

$$Q=2^m-2=2^1+2^2+2^3+\ \ldots\ldots\ +2^{m-1} \qquad (3\text{-}11)$$

위의 내용을 토대로 제산 알고리즘을 도출하면 다음과 같다.

[STEP1] 피제산원소 e_i와 제산원소 e_j를 각각 비트벡터공간으로 표현한 $e_i(av)$와 $e_j(bv)$로 표시한다.

[STEP2] 제산원소의 원시기약다항식 G(ξ)에 대한 승법역원생성다항식 $[G(\xi)]^{-1}$과 역원비트벡터공간 bv^*로 표시된 $e_j(bv^*)$를 구한다.

[STEP3] STEP2에서 구한 $e_j(bv^*)$를 승산 알고리즘의 승산원소로 한다.

[STEP4] 이후 부터는 승산 알고리즘의 STEP2이후와 동일하다.

4. 고효율 연산시스템 구성

본 장에서는 앞에서 논의한 연산을 통합하고 선택하기 위해서 분배기 모듈의 구성에 대해 논의한다.

4-1. 분배기 모듈 D_1

피연산원소인 $a_i(av)$는 가산일때는 가산기 모듈의 입력으로 승산일때는 ξ^r 생성 모듈의 입력으로 사용된다.

그러므로 이를 수행하기 위한 제어입력 T_1과 패스 트랜지스터 $G_{D1i}(i=0,1)$로 모듈 D_1의 기본 셀을 구성할 수 있다. 이를 식으로 표현하면 다음 식(4-1)과 같고 이를 토대로 D_1 모듈을 구성할 수 있으며 이에 대한 진리치표는 표4-1과 같다.

a_i= y_{0i} if T1=0 ⇨ 가산기 모듈
y_{1i} if T1=1 ⇨ 승산기 모듈 (4-1)
여기서 i=0,1,......,m-2,m-1이다.

표 4-1. 모듈 D1의 기본셀(D1-cell)에 대한 진리치표.
Table 4-1. Truth table for basic cell(D1-cell) of module D1.

a_i	T1	G_{D10}	Y_{0i}	G_{D11}	Y_{1i}
a_i	0	ON	a_i	OFF	×
a_i	1	OFF	×	ON	a_i

where, × means nonpass

4-2. 분배기 모듈 D_2

연산원소인 $a_i(bv)$는 가산과 감산일때는 가산기 모듈의 입력으로, 승산일때는 ξ^r 생성 모듈의 입력으로, 제산일때는 승법역원생성 모듈의 입력으로 사용된다.

그러므로 이를 수행하기 위한 제어입력 T_1과 T_0 및 패스 트랜지스터 $G_{D2i}(i=0,1,2,3)$로 모듈 D_2의 기본 셀을 구성할 수 있다. 이를 식으로 표현하면 식(4-2)와 같으며 이에 대한 진리치표는 표4-2와 같다.

b_j = Y_{0j} if T1T0=00 ⇨ 가산기 모듈
Y_{1j} if T1T0=01 ⇨ 감산기 모듈(생략가능)
Y_{2j} if T1T0=10 ⇨ ξ^r생성 모듈
Y_{3j} if T1T0=10 ⇨ 승법역원생성 모듈
(4-2)

4-3. 고속 연산기 구성

앞의 4-1절과 4-2절의 내용을 토대로 최종 임베이드 컴퓨터시스템에 기반을 둔 고효율 연산시스템을 구성하면 다음 그림4-1과 같다.

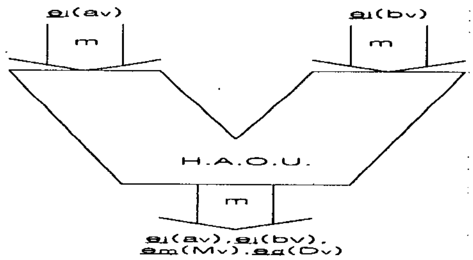
표 4-2. 모듈 D_2의 기본 셀(D_2-cell)에 대한 진리치표
Table 4-2. Truth table for basic cell(D_2-cell) of module D2.

b_i	T_1 T_0	G_{D20}	G_{D21}	Y_{0i}	G_{D22}	G_{D23}	Y_{2i}
b_j	0 0	ON	ON	b_j	ON	OFF	×
b_j	0 1	ON	OFF	×	ON	ON	b_j
b_j	1 0	OFF	ON	×	OFF	OFF	×
b_j	1 1	OFF	OFF	×	OFF	ON	×

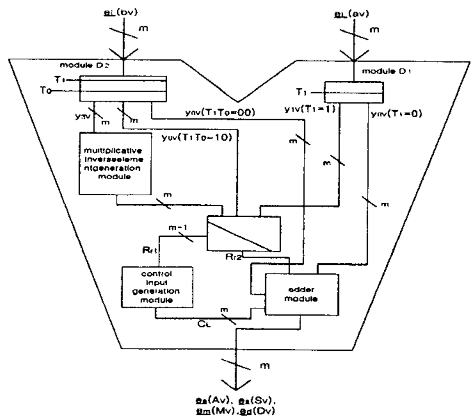
continued

b_i	T_1 T_0	G_{D24}	G_{D25}	Y_{0i}	G_{D26}	G_{D27}	Y_{2i}
b_j	0 0	OFF	ON	×	OFF	OFF	×
b_j	0 1	OFF	OFF	×	OFF	ON	×
b_j	1 0	ON	ON	×	ON	OFF	×
b_j	1 1	ON	OFF	×	ON	ON	×

where, × means nonpass.



a) symbol



b) internal structure

그림 4-1. 임베이드 컴퓨터시스템에 기반을 둔 고효율 연산시스템 블록다이아그램
Fig. 4-1. Block diagram of high efficiency arithmetic system based on embedded computer system

5. 결론

본 논문에서는 최근에 그 활용과 향 후 21C에 많이 적용되는 멀티미디어 H/W 시스템에 반드시 필요한 고효율 연산시스템을 임베이드 컴퓨터시스템에 기반을 두고 구성하는 한가지 방법을 제안하였으며 특징을 요약하면 다음과 같다.

본 논문에서 제안한 고효율 연산시스템은 다음과 같은 6개의 모듈로써 구성된다.

1. 가산기 모듈
2. ξ^r생성 모듈
3. 제어입력 C_L 생성 모듈
4. 승법역원생성 모듈
5. 분배기 모듈 D_1
6. 분배기 모듈 D_2

또한, 승산기 모듈과 제산기 모듈은 각각 다음과 같은 모듈을 합성함으로써 구성된다.

승산기 모듈=(ξ^r생성 모듈)+(제어입력 C_L 생성 모듈)+(가산기 모듈)

제산기 모듈=(승법역원생성 모듈)+(승산기 모듈)

위에서 본바와 같이 가산기 모듈은 가산, 감산, 승산, 제산의 어떤 연산을 하더라도 항상 사용된다.

그리고 가산기 모듈내의 cell-A의 개수는 m개, ξ^r생성 모듈내의 cell-M의 개수는 m^2개이고 분배기 모듈 D_1과 D_2내의 D_1-cell과 D_2-cell의 개수는 각각 m개이다.

또한, 제안한 고속 연산기는 모듈들의 합성으로 구성되므로 m의 확장에 따른 고속 연산기는 각 모듈을 m에 따라 확장만 하면 되며 최종 고속 연산기는 분배기 모듈로서 합성하여 용이하게 구성할 수 있다.

[참고문헌]

[1] I.F.Blake, *Algebraic Coding Theory:History and development*, Down, Hutchinson & Ross,Inc., Stroudburg,Pennsylvania,1973.

[2] R.E.Blahut, *Fast Algorithms for Digital Signal Processing*, Addison-Wesley Publishing Company, Inc.,1985.

[3] S.Y.Kung, *VLSI ARRAY PROCESSORS*, Prentice-hall,Inc.,1988.

[4] K,Bromley, Sun-Yuan Kang and E.Swartzlander, *International Confernece on SYSTOLIC Array*, Computers Society, Press, N.Y.,1985.

[5] M.D.Ercegovac and T.Lang, *Digital Systems and Hardware/Firmware Algorithms*, John Wiley & Sons, Inc., Canadea, 1985.

[6] Alfred K.W. Yeung and Jan. M. Rabaey,"A Recobfigurable Data-Driven Mutiprocessor Architecture for Rapid Prototyping of High Throughtput DSP Algorithms," HICSS-26 vol.1, IEEE Coputer Society Press, 1993.

[7] F. Balarin et al., Hardware-Software Co-design of Embedded Systems: The Polis Approach, Kluwer Academic Press, Boston, June 1997.

[8] R.K.Gupta, Co-Synthesis of Hardware and Software for Digital Embedded Systems, vol. 329, Kluwer Academic Publishers, Boston, Aug. 1995.

[9] R.Lidi and G.Pilz, *Applied Abstract Algebra*, Spring-Verlg,Inc.,N.Y.,1984.

[10] E.Artin,*Galois Theory*,NAPCO Graphic arts, Inc.,Wisconsin.1971.

[11] C.Y. Lee, E.H. Lu and J.Y. Lee,"Bit-parallel systolic multipliers for GF(2^m) fields defined by all-one and equally space polynomials,"vol. 50, no.5, pp.385-393, IEEE Trans. Compt., May 2001.