論文2001-38SD-6-7

Op Amp 회로를 이용한, 모듈로 (2^n-1) 병렬 승산기의 설계 및 그 기술의 응용

(Designing Modulo (2^n-1) Parallel Multipliers and its Technological Application Using Op Amp Circuits)

李 勳 圭 *, 金 鐵 **

(Hun-Giu Lee and Chul Kim)

요 약

본 논문은, Op Amp 회로를 이용한, 모듈로(modulo) (2^n-1) 병렬처리(parallel-processing) 잉여(residue) 승산기(multipliers)의 설계 및 이진(binary) 승산기 설계에 대한 그 기술의 응용(應用) 방법에 관(關)한 것이다. 전산처리(電算處理)에 있어서 승산속도(乘算速度)의 제약은 집적회로(VLSI) 기술의 발전에 많은 지장(支障)을 초래(招來)한다. 본 연구는, 이러한 문제를 해결키 위해 (Op Amp 회로를 이용) 모듈로 (2^n-1) 상에서, 시간복잡도(time complexity)가 $O(\log_2(\log_2(\log_2 n)))$보다 우수한, 일종(一種)의 모듈로 병렬 승산기를 구현함과 동시에, 그 기술의 이진 승산기 설계에 대한 응용방법을 모색한다. 이러한 병렬 승산기는 기존의 병렬 승산기들에 비(比)해 에어리어복잡도(area complexity) 및 시간복잡도(time complexity)에 있어 매우 우수한 성질들을 갖게 되며, 같은 효율(刻率)을 갖는 이진 승산기의 제작에 쉽게 응용할 수 있어 그 학술적 이용 가치가 높다.

Abstract

In this paper, we introduce modulo (2^n-1) parallel-processing residue multipliers, using Op Amp circuits, and their technological application to designing binary multipliers. The limit of multiplying speed in computational processing is a serious barrier in the advances of VLSI technology. To solve this problem, we implement a class of modulo (2^n-1) parallel multipliers having superior time complexity to $O(\log_2(\log_2(\log_2 n)))$ by applying Op Amp circuits, while investigating their technological application to binary multipliers. Since they have excellent time & area complexity compared with previous parallel multipliers, and are applicable to designing binary multipliers of the same efficiency, such parallel multipliers possess high academic value. Indexing Terms Modular Multipliers, Binary Multipliers, Parallel Processing, Operational Amplifiers, Mersenne Numbers.

* 正會員, 光云大學校 ISRC
(Kwangwoon University ISRC)
** 正會員, 光云大學校 數學科 ISRC
(Kwangwoon University ISRC)
接受日字:2000年8月10日, 수정완료일:2001年5月29日

Ⅰ. 승산기에 대한 현재의 연구와 그 문제점

1. 승산기에 대한 현재의 연구 동향(動向)

임의의 자연수 n에 대하여, $M_n=2^n-1$꼴로 표현되는 자연수들 M_n을 머젠수(Mersenne numbers) 라고 하며, 모듈로 머젠수(2^n-1) 잉여(residue) 시스템은,

RNS(Residue Number System)에서 그 운용(manipulating)과 스케일링(scaling)의 편리성으로 말미암아 심도 깊게 연구되고 있는, 특수한 모듈 리(moduli) 셑(set); $\{2^n-1, 2^n, 2^n+1\}$ 중(中) 하나이다. 본 논문에서 우리는, 모듈로 (2^n-1) 상(上)에서 구현된 (Op Amp 회로 응용의) 고성능 병렬 승산기 설계 방법 및 그 기술의 이진(二進) 승 산기 설계에 대한 응용에 관해 논한다.

전산처리(電算處理)에 있어서 시간 및 에어리어 복잡도(time and area complexity)가 우수한 승산기를 고안하는 일은 컴퓨터 학자들에게 있어 흥미 있고도, 도전적인 연구 과제이다. 왜냐하면 고도로 발달하고 있는 현대의 과학/기술이 보다 많은 데이터들을, 좀더 빠른 시간 내에 계산할 것을 요구하고 있기 때문이다. 최근들어 성능이 우수한 승산기들이 다양하게 고안되고 있는데[3,5,7,8,15]; 이러한 승산기들은 크게 모듈로(residue) 승산기[4,6~8,10,14,15], 디지털 이진 승산기[3,5], 아날로그 이진 승산기등[9]으로 나뉘고, 대부분 병렬처리 효과를 이용하여 속도 향상을 꾀하고 있다. 이러한 승산기들은 워드렝쓰(word lengths); 입력 비트 n이 큰 경우에 대해 「중복(redundant) 이진(binary) 가산(addition) 트리(tree)」[3]의 원리와 「프리캐리(precarry) 가산(addition) 컬럼(column) 압축(compression) 트리(tree)」[5] 의 원리를 써서, 가능한한 최소한의 에어리어복잡도와 최상의 시간복잡도를 유지하고 있다[7]. 그리하여 현재 고속 병렬 승산기들의 시간복잡도는 $O(\log_2 n)$에, 에어리어복잡도는 $O(n^2 \log_2 n)$에 머물고 있다.

2. 승산(乘算) 부울함수(Boolean functions)의 엔피하드(NP-hard) 복잡도

컴퓨터의 계산량/계산속도 문제들을 집적회로(VLSI)에서 해결해 가는 길은, 승산기와 같은 기초 연산회로의 시간복잡도를 질적으로 개선하는 것이다. 하지만 순수한 디지털 소자들은 본질적으로 승산(multiplying)의 부울함수가 갖게 되는 엔피하드의 복잡도[1]를 피해 가기 어려우므로 이것들만을 가지고 (양호한 에어리어복잡도를 유지하면서) 승산기의 시간복잡도를 질적으로 개선하는 데는 한계가 있다. 따라서 본 논문에서는 - OP Amp 회로를 이용 - 에어리어복잡도의 양호함이 유지됨과 동시에 시간복잡도가 $O(K)$; $O(K) < O(\log_2(\log_2(\log_2 n)))$인 모듈로 (2^n-1) 승산기의 설계 방법을 설명하며, 더 나아가서는 그 기술의 장점을 이진 승산기 설계에 응용하는 방법을 제시코자 한다.

Ⅱ. 모듈로 (2^n-1) 승산기의 설계(設計)

1. 4 비트 모듈로 (2^n-1) 승산기의 설계

그림 1은 본 논문이 제안하고 있는 모듈로 (2^n-1) 승산기의 설계 예로서, 4 비트 승산기의 회로도이다. 블럭 G1의 n^2 부분곱(products) 생성부는 (A; A4, A3, A2, A1)와 (B; B4, B3, B2, B1)의 8 비트 신호를 입력받아 16 개의 부분곱을 형성하게 된다(여기서 A는 지그재그로 연결되고 B는 수직으로만 연결된다). 그림 2에 설명된 바와 같이, 동그라미(9)는 「AND 게이트(gate) 및 플립플롭(flip- flop)」의 소자(素子)를 나타내며, 사각형(10)은 플립플롭만의 소자를 나타낸다. 그리하여, 4 비트 승산기의 경우, 블럭 G1, G2, G3에 각각 16, 12, 8 개의 소자들이 배치된다. 한편, ($ADD1$, …, $ADD8$)은 (그림 3과 같은) 아날로그 가산기(加算機)들을 나타내고, ($FAD1$, …, $FAD8$)은 (그림 4와 같은) 플래쉬(flash) 아날로그디지털변환기(Analog Digital converters)들을 나타낸다. 종단(終壇)에 있는 'CLA 1'은 모듈로 (2^n-1) 캐리룩어헤드(Carry Look Ahead) 생성기(generator)[11]로서, A와 B의 신호를 모듈로 (2^n-1) 상(上)에서 승산한 최종 결과를 (C; C1, C2, C3, C4)으로 빠르게 출력하는 역할을 한다.

개략적인 동작원리는 다음과 같다. (A4, A3, A2, A1)은 차례대로 $(2^3, 2^2, 2^1, 2^0)$ 자리의 계수(係數)들(각각 0 또는 1)을 나타내고 (B4, B3, B2, B1)에 대해서도 마찬가지의 원리가 적용된다. 블럭 G1을 제 1 단계의 신호행렬(信號行列)로 본다면 그 (1,3) 원소는 A의 2^2 계수와 B의 2^1 계수의 곱이므로 2^3의 계수가 된다. 마찬가지로 하여, 신호행렬의 제 1 행은 모두 2^3의 계수가 된다. 한편, (3,2) 원소는 A의 2^3 계수와 B의 2^2 계수의 곱이므로 $2^3 * 2^2 = 2^1 \pmod{2^4-1}$의 계수가 된다. 이런 식으로, 1 단계 신호행렬의 (1, 2, 3, 4) 행에 있는 모든 원소들이 각각, $(2^3, 2^2, 2^1, 2^0)$의 계수들이 된다. 블럭 G1(부분곱 생성부)에서 AND 연산을 거친 신호들은 (일시적으로 플립플롭에 저장되었다가) 타이밍 클럭에 맞추어 일제히 블럭 G2의 플립플롭 군(群);

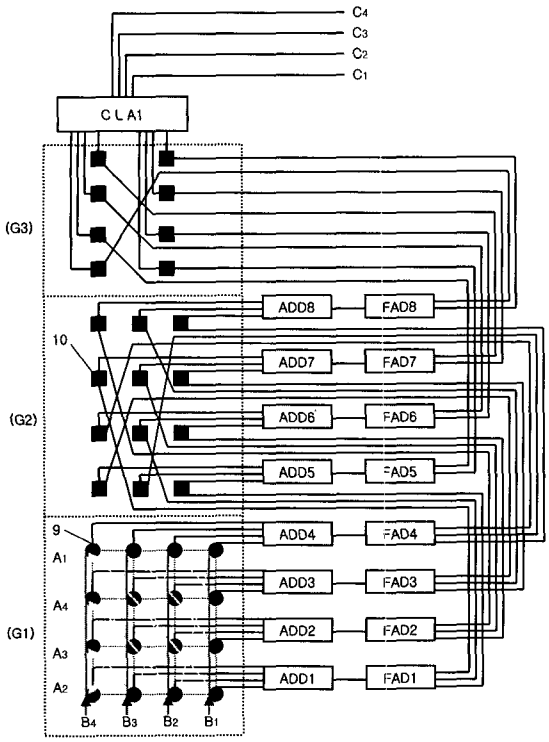

그림 1. 4 비트 모듈로 (2^n-1) 승산기의 설계
Fig. 1. A design of a 4-bit modulo (2^n-1) multiplier.

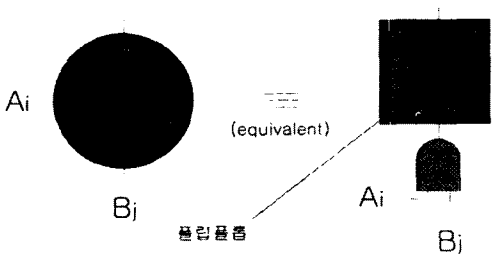

그림 2. 'AND게이트 및 플립플롭'의 소자
Fig. 2. An AND gate & flipflop device.

'제 2 단계 신호행렬'에로 자리를 옮기게 되는데, 그 도중에 가산기(ADD1, ADD2, ADD3, ADD4)와 플래쉬 A/D변환기(FAD1, FAD2, FAD3, FAD4)를 통과한다. 신호행렬 G1의 제 3 행 신호들은 ADD2에서 아날로그 값(0 - 4)로 환산(換算)되고, 이것이 FAD2를 거치면서 3 비트의 이진 값을 갖게 된다. 따라서 행렬 G2의 (1, 2, 3, 4) 행에 있는 모든 원소들이 각각 $(2^3,2^2,2^1,2^0)$의 계수들임을 전제로 하고, 행렬 G1의 3 행 원소들이 2^1의 계수들임을 고려하면, 그 최상위 비트 신호가 2

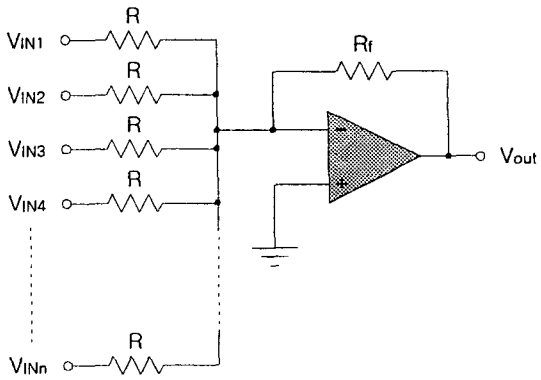

그림 3. Op Amp를 이용한 n 입력 가산기
Fig 3. An n-input adder using an Op Amp.

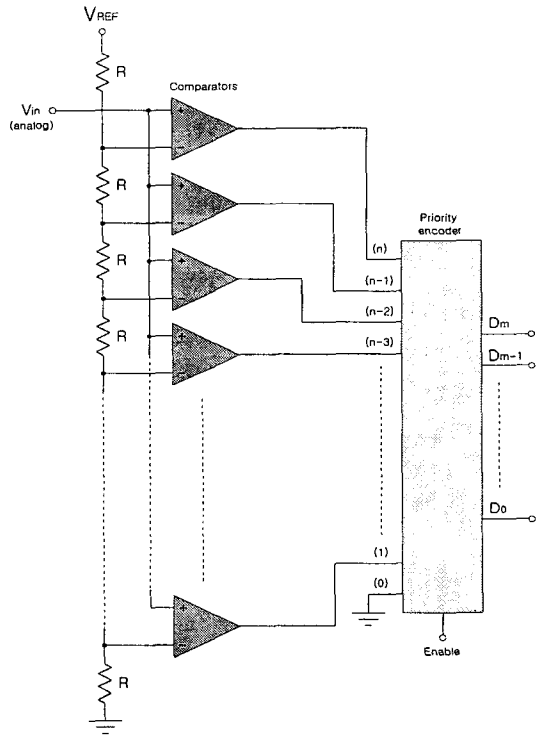

그림 4. Op Amp를 이용한 m 비트 플래쉬 A/D변환기
Fig. 4. An m-bit Flash A/D converter using Op Amps.

행 앞서 가게 되고(행렬 G2의 (1,1) 원소가 되고), 그 다음 비트 신호는 1 행 앞서 가게 되며 ((2,2) 원소가 되며), 최하위 비트 신호는 같은 3 행으로 간다. 그런데 모듈로 2^n-1 잉여 수 시스템은 그 최상위 비트와 최하위 비트를 이어서 원형(圓形)으로 감아 줄 경우 회전대칭성(回轉對稱性)이 성립하므로($2^{n+s}=2^s$

(mod 2^n-1)), 행렬 G1의 (1, 2, 4) 행에 대해서도 마찬가지의 원리를 적용해 줄 수 있다. 그리하여 제 1 행의 경우에는 FAD4의 최상위 비트 신호가 행렬 G2의 3 행($2^3*2^2=2^1$(mod 2^4-1))에 있게 되며, FAD4의 그 다음 비트 신호 및 최하위 비트 신호들도 각각 제 4 행 및 1 행에 있게 된다. 여기서 본래의 행에 남게 되는 신호들과, 1 행 앞서가는 신호들 및 2 행 앞서가는 신호들을 각각 3, 2, 1 열에 위치시킴으로써, 모든 신호들이 일시에 병렬적으로 처리되도록 할 수 있다. 이런 식으로, 신호행렬 G3에 압축된 신호들이 저장되게 된다. 이때, 블럭 G1, G2, G3의 플립플롭 군들이 하나의 클럭 신호에 대해서 일괄적으로 동기화(synchronized)되므로 3 회의 싸이클에 의해 CLA1의 입력을 얻게 되며, CLA1은 블럭 G3에 위치한 두 이진 벡터의 플립플롭 신호들을 (빠른 속도로) C의 출력으로 바꾸어 주는 역할(役割)을 한다.

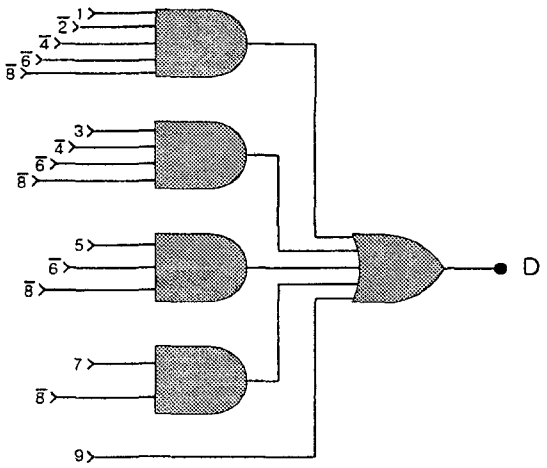

그림 5. 프라이오러티엔코더의 대표적인 한 유니트[13]
Fig. 5. A representative unit of a Priority Encoder.

2. n^2 부분곱 형성부(形成部)에 대한 두 입력 신호들의 일반적 배선(配線)

그림 6를 통해서, n^2 부분곱 생성부의 일반적 배선 상태를 다음과 같이 설명할 수 있다. 앞에서와 같이 $n\times n$의 제 1 단계 신호행렬을 설정하고, 그림과 같이 배선해주면 그 $(1,2,\cdots,n)$ 행의 모든 원소들이 각각 $(2^{n-1},\cdots,2^1,2^0)$ 자리의 계수들이 된다. 예를 들자면, 그 (1,1) 원소는 2^0(A1)의 계수와 2^{n-1}(Bn)의 계수의 곱이므로 2^{n-1} 자리의 계수가 되고, 그 (2,1) 원소는 2^{n-1}(An)의 계수와 2^{n-1}(Bn)의 계수의 곱이므로 $2^{2n-2}=2^{n-2}$(mod 2^n-1) 자리의 계수가 된다. 이와 같이, 제 1 단계 신호행렬의 제 1 열 B가 신호(Bn)으로 고정되어 있음을 고려하면, $(i,1)$ 원소가 2^{n-i} 자리의 계수임을 알 수 있다. 또한, 그림으로부터 열(j)가 증가함에 따라 B 신호의 자리는 j 만큼 자리 수가 감소하고, A 신호의 자리는 j 만큼 자리 수가 증가하므로 그 (i,j) 원소는 2^{n-i} 자리; $2^{n-i}*2^{-j}*2^{+j}=2^{n-i}$ (mod 2^n-1)의 계수가 됨을 알게 된다. 그러므로 $n\times n$의 일반 입력에 대해서도, 그림 1의 제 1 단계 행렬(G1)과 같은 형태의 부분곱 생성부를 설계할 수 있다.

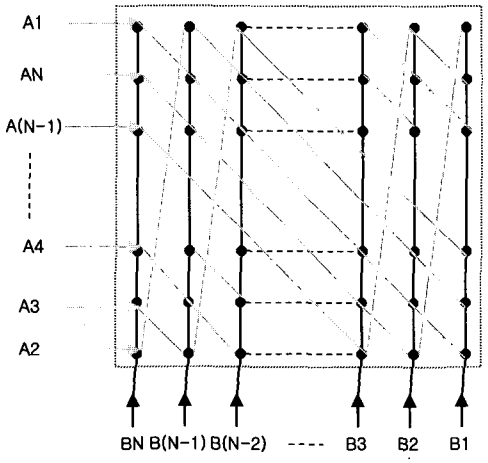

그림 6. n^2 부분곱(products) 형성부(形成部)에 대한 두 입력 신호들의 일반적 배선(配線)
Fig. 6. The general wiring of two input signals about the formation part of n^2 products.

3. r 단계 신호행렬에서 $r+1$ 단계 신호행렬에로의 캐리/합 전파(傳播)

그림 7은 제 r 단계 신호행렬(블럭 G(r))에서 제 $r+1$ 단계 신호행렬(블럭 G(r+1))에로의, 캐리/합의 전파를 나타낸다. (r 단계의 각 연산세포(가산기 + 플래쉬 A/D변환기)들이, m_r 비트의 입력 신호들을 ($\lfloor \log_2 m_r \rfloor +1$) 비트의 출력 신호들로 바꾸어 주므로) r 단계 신호행렬이 $n\times m_r$의 크기를 갖는다면, $r+1$ 단계 신호행렬은 $n\times(\lfloor \log_2 m_r \rfloor +1)$의 크기를 갖게 된다.

r 단계 신호행렬의 제 1 행 원소들은 (2016 ~

2017)으로 나타내고, 제 n 행 원소들은 (2001 ~ 2020)으로 나타내기로 한다. 또한, 제 r 단계 신호행렬의 i 행이 (2002 ~ 2019) 또는 (2003 ~ 2018)으로 나타내질 수 있으며, $r+1$ 단계 신호행렬의 i 행도 (2007 ~ 2014) 또는 (2009 ~ 2013)으로 나타내질 수 있다고 가정하자. 그런 다음에, 제 r 단계 신호행렬의 i 행 출력을 받는 FAD의 [최상위 비트 신호를 제 $r+1$ 단계 신호행렬의 $(i-\lfloor \log_2 m_r \rfloor, 1)$ 원소에 대응되는, 그 다음 비트 신호를 $(i-\lfloor \log_2 m_r \rfloor+1, 2)$ 원소에 대응되는, …, 최하위 비트 신호를 $(i, \lfloor \log_2 m_r \rfloor+1)$ 원소에 대응되는] 각각의 플립플롭들에 입력시키면 (i가 조건식; $\lfloor \log_2 m_r \rfloor < i \leq n$을 만족시킬 경우) 그 입/출력 신호들이, (2002 ~ 2019)의 분포에서, (2008 ($(i-\lfloor \log_2 m_r \rfloor, 1)$ 원소에 대응되는 플립플롭의 입력 신호) ~ 2014 ($(i, \lfloor \log_2 m_r \rfloor+1)$ 원소에 대응되는 플립플롭의 입력 신호)) 꼴의 분포로 배치되고 : 그렇지 않으면(조건식; $1 \leq i \leq \lfloor \log_2 m_r \rfloor$을 만족시키면) 그 입/출력 신호들이, (2003 ~ 2018)의 분포에서 (2006 ($(n+i-\lfloor \log_2 m_r \rfloor, 1)$ 원소에 대응되는 플립플롭의 입력 신호) ~ 2004 ($(n, \lfloor \log_2 m_r \rfloor - i+1)$ 원소에 대응되는 플립플롭의 입력 신호) ~ 2011 ($(1, \lfloor \log_2 m_r \rfloor - i+2)$ 원소에 대응되는 플립플롭의 입력 신호) ~ 2013 ($(i, \lfloor \log_2 m_r \rfloor+1)$ 원소에 대응되는 플립플롭의 입력 신호)) 꼴의 분포로 배치된다; 왜냐하면, $s \geq i$일 때 $2^{n-i+s} = 2^{s-i} \pmod{2^n-1}$의 관계식이 성립하므로, 행의 위치가 s 회(回) 상위(上位) 자리로 쉬프트(shift)되는 것들의 이진 자리는 2^{s-i}이고, 따라서 1 행을 지나친 신호들은 $n-(s-i)$ 행의 위치로 회전되기 때문이다.

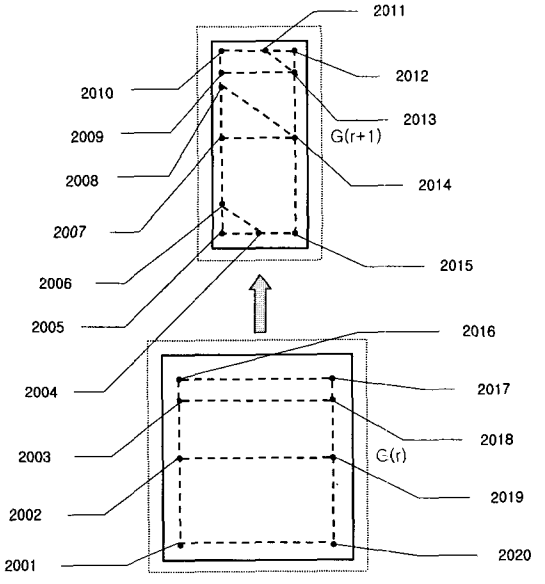

그림 7. r 단계 신호행렬에서 $r+1$ 단계 신호행렬에로의, 캐리/합의 전파(傳播)

Fig. 7. The propagation of carries & sums from the rth signal matrix to the $r+1th$.

4. 캐리/합의, 일반적 신호 흐름 및 배선

그림 8은 (캐리/합의 일반적인 신호 흐름/배선을 나타내며) 그림 6, 7의 분석들을 하나의 도면에 종합한 것이다. 블럭 G1은 부분곱 생성부(제 1 단계의 $n \times n$ 신호행렬)을 지시하고 있다. 따라서 그 행렬의 제 1 열은 (3001 - 3003)으로 나타내지고, 제 i 행은 (3002 - 3009)로 나타내진다. 모듈로 (2^n-1) 잉여 수 시스템은 신호행렬들의 행에 대해, 회전대칭성(回轉對稱性)이 성립하므로:

$$2^{n-i+\Delta} = 2^{\langle \frac{n-i+\Delta}{n} \rangle} \pmod{2^n-1},$$
$$0 \leq n-i \leq n-1 \text{ 그리고 } \Delta \in \{0,1,2,\cdots\},$$

i 행($i > \lfloor \log_2 n \rfloor$)에 대한 캐리/합의 신호 흐름을 기술(記述)해 줌으로서 전체 신호들의 병렬적 처리 과정을 설명할 수 있다. 그러므로 i 행 신호들을 축출하여, 그 $(i,1)$, $(i,2)$, …, $(i,n-1)$, (i,n)의 원소들을 (3004, 3005, …, 3006, 3007)로써 표시할 때, 이 신호들이 ADD101과 FAD101을 거치게 되면 ($\lfloor \log_2 n \rfloor+1$) 비트의 이진 출력 신호(3011, 3012, …, 3017, 3018)가 되고, 이것들은, 또한 동시에 각각, 제 2 단계 신호행렬의 $(i-\lfloor \log_2 n \rfloor, 1)$, $(i-\lfloor \log_2 n \rfloor+1, 2)$, …, $(i-1, \lfloor \log_2 n \rfloor)$, $(i, \lfloor \log_2 n \rfloor+1)$ 위치에 있는 플립플롭들의 입력 신호들을 나타내게 된다(여기서 블럭 G2는 제 2 단계 신호행렬/플립플롭 군을 지시한다). 그 다음 2 단계 신호행렬의 i 행 신호들($(i,1)$, $(i,2)$, …, $(i, \lfloor \log_2 n \rfloor)$, $(i, \lfloor \log_2 n \rfloor+1)$)을 축출하여, 이것들을 각각, (3013, 3014, … , 3015, 3016)으로 나타낼 수 있다. 이 신호들이 다시 ADD102와 FAD102를 거치면 ($\lfloor \log_2 (\lfloor \log_2 n \rfloor+1) \rfloor+1$) 비트의 이진 출력 신호가 되고, 이 출력 신호(3019, 3020, …, 3024, 3025)들은 각각, 제 3 단계 신호행렬의 [$(i-\lfloor \log_2 (\lfloor \log_2 n \rfloor+1) \rfloor, 1)$, $(i-\lfloor \log_2(\lfloor \log_2 n \rfloor$

$+1)\rfloor+1, 2), \cdots, (i-1, \lfloor \log_2(\lfloor \log_2 n \rfloor+1)\rfloor), (i, \lfloor \log_2(\lfloor \log_2 n \rfloor+1)\rfloor+1)]$ 원소 위치에 있는 플립플롭들의 입력 신호들을 나타내게 된다. 블럭 G3는 바로 이와 같은 절차의 병렬 연산들이 되풀이되는 과정을 나타내고 있다. K 번의 클럭 주기(週期)가 지나간 후에, 제 1 단계(블럭 G1)의 입력 신호들은, K 단계 신호행렬의 i 행 신호((3021), (3022), (3023))를 거쳐서 K+1 단계 신호행렬(블럭 G4)의 i 행 신호((3027), (3028))에 도달한다. 그런데, CLA1[11]의 입력으로 주어진 K+1 단계 신호행렬의 i 행 신호((3027), (3028))은 (사실상) 임의적 행의 신호들을 대표하므로, CLA1의 입력 신호는 모듈로 (2^n-1) 상에서의 두 수인(각각, n 비트인) 입력 신호들이 되고, 이 신호들이 (CLA1에 의해서) 시간복잡도 $O(1)$ 이내에 n 비트의 출력(3030)으로 바뀌게 되는 것이다.

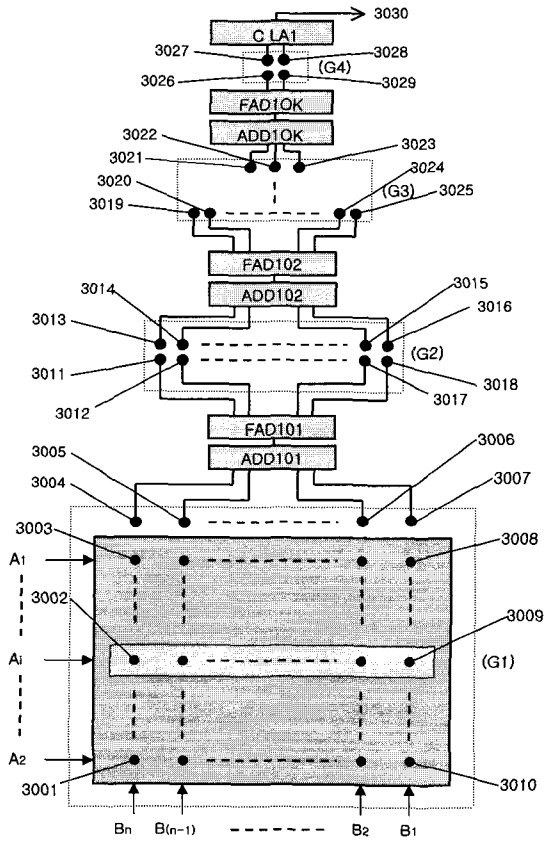

그림 8. 캐리/합의 - 일반적 - 신호 흐름 및 배선
Fig. 8. The general signal propagation and wiring of carries & sums.

Ⅲ. 이진 승산기의 설계

다음으로 앞에서 설명한 모듈로 (2^n-1) 병렬 승산기의 설계 기술을, 일반적인 n 비트 이진 승산기에 적용하는 방법을 설명코자 한다. r 단계 신호행렬은 이제 $2n\times m_r$의 행렬이 되고, 이 행렬들의 (1, …, 2n-1, 2n) 행들은 각각, $(2^{2n-1}, \cdots, 2^1, 2^0)$의 자리들을 나타낸다. 여기서, m_r은 r 단계 신호행렬의 열 수로서 모듈로 (2^n-1) 승산기에서의 그것과 일치되는 값이다(따라서, n 비트의 모듈로 (2^n-1) 승산기가 (K+1) 개의 신호행렬을 갖게된다면, 본 논문이 제시하고 있는 n 비트의 이진 승산기도 (K+1) 개의 신호행렬을 갖게 된다). 행의 수가 두 배이므로, 각 단계에 적용되는 아날로그연산세포 개수가 약 두 배로 늘어나고, 신호행렬의 원소들에 대응되는 소자들의 개수도 약 두 배로 늘어난다. n^2 부분곱 생성부의 배선 방법은 전형적(典型的)인 이진 승산기의 그것과 같다. 다만 신호행렬의 정의(定義)을 고려하여 상위(上位) 비트들이 1 행 쪽에, 하위(下位) 비트들이 $2n$ 행 쪽에 오도록 한다. 캐리/합의 전파에 따른 - r 단계에서 $r+1$ 단계로의 - i 행 $(1\le i\le 2n)$ 신호들 배치 방법은 앞에서와 같이 [i행 FAD의 최상위 비트 신호를 $(i-\lfloor \log_2 m_r \rfloor, 1)$에, 그 다음 비트 신호를 $(i-\lfloor \log_2 m_r \rfloor+1, 2)$에, …, 최하위 비트 신호를 $(i, \lfloor \log_2 m_r \rfloor+1)$ 위치에] 있는, $r+1$ 단계 신호행렬의 신호가 되도록 한다. (주의할 것은, $s\ge i$에 대해 $2^{n-i+s}=2^{s-i}(\mathrm{mod}\, 2^n-1)$와 같은 캐리/합 전파의 회전대칭성이 더이상 성립치 않는다는 것이다; 따라서, i행의 이진 자리에 대해 $2^0\le 2^{2n-i}\le 2^{2n-1}$의 관계식이 항상 성립하고, 신호들의 캐리는 - 선형적(線形的)으로 - 상위 비트 자리로만 이동하여, $2n$ 비트의 스팬(span)에 걸쳐 분포하게 된다. 그러므로, 제 1 행(2^{2n-1} 자리)의 상위 비트로 전송되는 캐리/합 신호들은 무시(無視)되며, 이러한 신호들에 대해서는 신호 전송을 위한 배선이 불필요하다.) 신호행렬 원소들 사이에서의 신호 천이(遷移) 방식에 맞추어 그것에 대응되는 소자들을 배선하여 주면 되는데, 이진 승산기의 신호행렬은 그 원소들의 값이 0(zero)인 영역들이, 상위 비트 및 하위 비트 영역에 일정한 규칙을 따라 군집(群集)을 형성(形成)하여 분포하므로, 이것

에 맞추어 회로를 설계하면 효율적이다. 이와 같이 하고, 종단(終壇)에 있는 모듈로 (2^n-1) 캐리룩어헤드생성기(CLA1)를 이진수에 대한 CLA(CLA2)로 바꾸어 줌으로서, 이진 승산기를 구현할 수 있게 된다. 다음의 그림 9는 이러한 일반 원리에 따라 설계한 4 비트 이진 승산기의 회로도이다; 모듈로 (2^n-1) 승산기에서와 같이, 그림 9에서 AF는 「AND 게이트 및 플립플롭」의 소자를 나타내며, FF는 플립플롭만의 소자를 나타낸다.

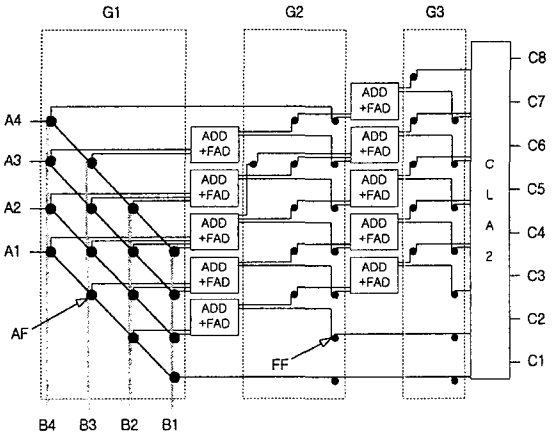

그림 9. 4 비트 이진 승산기의 효율적 설계
Fig. 9. An efficient design of a 4-bit binary multiplier.

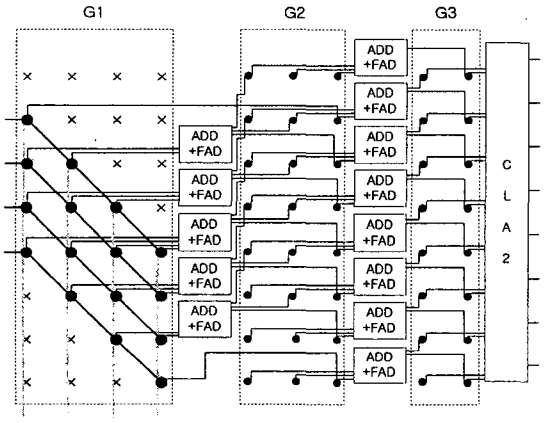

그림 10. 4 비트 이진 승산기의 설계(규칙 어레이)
Fig. 10. A design of a 4-bit binary multiplier (regular array).

한편, n이 클 경우에 대해서는 그림 10과 같은 규칙(regular) 어레이(array)를 적용한다; 제 2 단계 신호행렬부터는 모든 신호행렬 원소들에 대해 무조건적으로 플립플롭 소자들을 할당하며, 또한 배선하는 것이다. 이때, 전체 신호행렬들에 대한 소자들의 개수에 대해, 다음의 자명(自明)한 관계식이 성립한다:

$$\lim_{n\to\infty}\frac{1}{n^2}(n^2+2n(\lfloor\log_2 n\rfloor+1)+2n(\lfloor\log_2(\lfloor\log_2 n\rfloor+1)+\cdots+2n(2))=1$$

위의 등식으로부터, n이 커질 경우에는 전체 신호행렬들에 대한 소자들의 개수가 n^2에 정비례하여 증가함을 알 수 있고 (실재로, n이 512 이상인 경우에는 전체 소자들의 개수가 n^2의 107.5 %보다 작게된다.) 이것은 n이 큰 경우들에 대해서도 효율적인 회로설계가 가능함을 의미하는 것이다.

Ⅳ. 성능(性能) 분석(分析)

본 논문이 제안하고 있는 승산기들은, 시간복잡도(time complexity)와 에어리어복잡도(area complexity) 그리고 기술상의 파급효과(波及效果) 측면에서 그 나름의 특별한 가치를 낳는다.

우선, 제안된 모듈로 (2^n-1) 승산기의 시간복잡도를 계산해 보기로 한다. 디지털 신호가, 그림 3 및 4에 나타난 ADD와 FAD를 통과하는 데 걸리는 시간은 (제품에 따라 다를 것이나) n에 무관하게 약 10 n sec 내외라고 볼 수 있다; 이것은 디지털 신호들이 1 비트 전가산기(full-adder)를 한 번 통과하는데서 생기는 타임딜래이(time delay)와 동등한 수준이다. 그러므로 - 와이어(wire)에 의한 딜래이가 존재치 않는다는 가정 하에 - 각 아날로그연산세포에 사용되고 있는 ADD의 시간복잡도는 $O(1)$이 된다. 마찬가지로 FAD[12]의 시간복잡도의 값도 $O(1)$이되고(그림 5, Priority Encoder의 대표적인 한 유니트[13] 참조) 따라서, r 단계 신호행렬에서 $r+1$ 단계 신호행렬로 신호가 천이하는 데 소요되는 시간의 시간복잡도는 $O(1)$이 된다. 한편, 종단에 사용된 CLA의 시간복잡도의 값도 $O(1)$이 되므로, 전체 연산에 대한 시간복잡도는 $O(K)$: $O(K)<O(\log_2(\log_2(\log_2 n)))$이되며, 여기서 K는 n에 대한 의존변수(依存變數)로서 아래의 관계식:

$$\lceil f^K(n) = \lceil \log_2(\lceil \log_2(\cdots(\lceil \log_2 n \rceil + 1)\cdots),$$
$$f(x) = \lfloor \log_2 x \rfloor + 1 \rfloor$$

을 만족시킨다.

다음으로, 제안된 승산기의 에어리어복잡도를 생각해 보기로 한다. 각 단계의 신호행렬들이 갖게 되는 원소들의 개수는 그것에 대응하는 소자(素子)들의 개수: $\lceil n^2, n(\lfloor \log_2 n \rfloor + 1), n(\lfloor \log_2(\lfloor \log_2 n \rfloor + 1) \rfloor + 1), \cdots, n(2) \rfloor$와 같다. 그런데, 비교기(Comparator)의 수가 n인 하나의 FAD가 차지하는 실리콘에어리어를 n으로 나눈 값을 α라 하면, α는 ADD의 한 입력이 차지하는 실리콘에어리어 및 하나의 플립플롭이 차지하는 실리콘에어리어에 비해 동등이하의 크기를 갖는다고 말할 수 있다. 따라서, 하나의 아날로그연산세포가 차지하는 실리콘에어리어의 복잡도는 $O(n)$이 된다 (FAD는 출력 비트 수가 m일 때 2^m-1의 비교기 및 프라이오러티엔코더 논리게이트들이 요구될 수 있으나, 이 개수는 FAD에 출력을 보내는 ADD 입력 수(數)의 복잡도($O(n)$)을 넘지 않는다). 그러므로, CLA1의 에어리어복잡도($O(n^2)$)과, 제 1 단계 신호행렬의 디지털 소자들 및 아날로그연산세포들이 갖게되는 에어리어의 복잡도($O(n^2)$) 그리고, 신호행렬들이 형성하는 총 스테이지(stages)의 수(K)를 동시에 고려하여, 전체 회로의 에어리어복잡도가 $O(n^2K)$이 됨을 알 수 있다.

본 논문이 제안하고 있는 모듈로 (2^n-1) 병렬 승산기의 장점은 이진 승산기의 설계에 응용될 수 있으며, 이러한 이진 승산기의 시간복잡도는 모듈로 (2^n-1) 병렬 승산기의 시간복잡도; $O(K)$와 같게되고, 에어리어복잡도는 - 신호행렬에 대응되는 각 단에서의 디지털 소자들 개수와, 합산증폭기들 입력의 개수로부터 - '$O(2n*n*K) = O(n^2K)$'이 됨을 알 수 있다.

따라서 현재까지의 연구 결과와, 이상의 분석들을 종합하여 하나의 표로 정리하면 다음과 같다.

cf) $O(K) < O(\log_2(\log_2(\log_2 n)))$

'표 1'을 통해서, 우리는 승산의 부울함수가 갖게 되는 엔피하드 복잡도를 피해 가기 위한 노력들이, 기본적으로 두 종류의 승산기로 나타남을 볼 수 있다. 그 하나는 (n^2 개의 원소들로 이뤄진 부분곱 형성부를 기

표 1. 승산기들의 복잡도(complexity) 비교(比較)
Table 1. Comparison of the complexities of multipliers.

명칭 및 참고문헌	시간복잡도 (Time Complexity)	에어리어복잡도 (Area Complexity)	간행 (Year)
Wallace Tree Multipliers[2]	$O(\log_2 n)$	$O(n^2\log_2 n)$	1964
Redundant Binary Addition Tree Multipliers[3]	$O(\log_2 n)$	$O(n^2\log_2 n)$	1985
Column Compression Tree Multipliers[5]	$O(\log_2 n)$	$O(n^2\log_2 n)$	1989
Montgomery Multipliers (systolic array)[14]	$O(n)$	$O(n^2)$	1993
Montgomery Multipliers (linear systolic array)[15]	$O(n)$	$O(n)$	1995
RNS Modular Multipliers[7]	$O(\log_2 n)$	$O(n^2\log_2 n)$	2000
제안된 모듈로 (2^n-1) 승산기들	$O(K)$	$O(n^2K)$	
제안된 이진 승산기들			

저(基底)로 하는) 고속 병렬 승산기들로서 시간 및 에어리어복잡도가 각각, $O(\log_2 n)$ 및 $O(n^2 \log_2 n)$이 된다. 다른 하나는, n 개의 원소들로 이뤄진 부분곱 형성부를 기저로 하고, 1000 비트 이상의 입출력을 갖게 되는 몽고메리 승산기들이다. 몽고메리 승산기의 경우 에어리어복잡도가 $O(n)$이 될 수 있는 장점이 있으나, 시간복잡도가 $O(n)$보다 작아질 수 없어 속도 면에서 저성능이다($n > 1024$의 경우, 고속 병렬 승산기들에 비해 100 배 이상의 연산시간이 요구될 수 있다). 본 논문에서 제안한 승산기들은 전자(前者)에 속한 것으로, 시간 복잡도가 $O(K)$이 되어 n이 클수록 스피드 측면에서 유리한 반면, 에어리어복잡도가 $O(n^2)$보다 크기 때문에 기존의 몽고메리 승산기들을 대체하기 어렵다[7]. 그러므로, 제안된 승산기들의 참다운 가치는, 시간 및 에어리어복잡도 측면에서의 '고속 병렬 승산기들'에 대한 상대적 우수성에 있다.

V. 검토(檢討)

Op Amp를 이용한 회로에서는 분해능(resolution) 문제 및 고주파(1 G Hz 내외)에서의 클럭킹(clocking) 문제가 대두될 수 있다. 그럼에도 불구하고, 현재 20 비트 이상의 A/D변환기가 상용화(常用化)되고 있는 것을 보면 승산기의 입력 비트; n이 10^6 이하인 경우에 대해서는 분해능 문제가 용이(容易)하게 해결되는 것을 볼 수 있고, 고주파에서 발생하는, ADD와 FAD 신호들의 클럭킹 문제는 프로세서(processor)의 회전(cycle) 수(數)에 제약(制約)을 부여함으로서, ADD와 FAD의 차단주파수 (f_c)가 약 1 G Hz 이하이고 입력 비트(n)이 작을 경우에는 제안된 승산기들의 연산속도가 고속 병렬 승산기들의 그것에 비해 저성능이나 ($C_1 \cdot \log_2 n < C_2 \cdot K$, $\because C_1 < C_2$), f_c가 1 G Hz 이상이거나 또는, n이 클 경우에는 오히려 고성능이다 ($C_1 \cdot \log_2 n > C_2 \cdot \log_2(\log_2(\log_2 n)) > C_2 \cdot K$: 여기서 C_1은 디지털 신호가 1 비트 전가산기를 한 번 통과하는데 걸리는 시간($\triangle t$)에 의해 결정되는 상수로서 약 $3\triangle t$가 되고[5], C_2는 제안된 승산기의 제 r 단계 신호행렬(플릴플롭 군) 신호들이 $r+1$ 단계 신호행렬에로 이동하는데 걸리는 시간으로, f_c가 클수록 작아질 수 있는 딜래이 상수이다). 따라서, 제안된 승산기들의 이론상 복잡도가 실질적인 연산속도와 실리콘에어리어의 효율로서 발현되기 위해서는, 사용되는 ADD와 FAD의 차단주파수가 충분히 커야한다; 이때, ADD와 FAD에 대한 딜래이(pass delay)는 중(重)시하지 않았는데, 그 이유는 디지털과 아날로그 소자들 사이의 딜래이 차이가 미미하기 때문이다. 그런데, 최근 들어 아날로그 IC 기술의 중요성이 재고되면서 10 G Hz 내외의 고주파에서도 동작할 수 있는 ADD와 FAD의 개발이 어렵지 않을 것으로 예상된다. 이러한 소자들이 개발될 경우, 제안된 승산기들의 연산속도와 에어리어는 기존의 가장 빠른 디지털 승산기보다 더욱 빠르게 되며, 또한 적은 양의 실리콘에어리어를 요구하게 된다. 게다가 이러한 성능 차이는 n이 커질수록 더욱더 증가하게 되는 것이다. 그러므로, 제안된 승산기들은 n이 크면서도 가장 빠른 연산이 요구되는 특수 목적의 회로들에 유용하게 쓰일 수 있다. 그 응용들 중의 하나는 정보보호 분야[16]이다. 최근 그 활용이 증가하고 있는 정보보호 HW 구현 기술분야에서 제안된 승산기들이 유용하게 활용될 수 있다.

참 고 문 헌

[1] R. E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Transac- tions On Computers*, vol. 35, no. 8, pp. 677~691, Aug. 1986.

[2] C. S. Wallace, "A Suggestion for a Fast Multiplier," *IEEE Transactions On Electronic Computers*, vol. 13, pp. 14~17, Feb. 1964.

[3] N. Takagi, H. Yasuura, and S. Yajima, "High-Speed VLSI Multiplication Algorithm with a Redundant Binary Addition Tree," *IEEE Transactions On Computers*, vol. 34, no. 9, pp. 789~796, Sept. 1985.

[4] F. J. Taylor, "Large Moduli Multipliers for Signal Processing," *IEEE Transactions On Circuits and Systems*, vol. 28, no. 7, pp. 731~736, July 1981.

[5] B. P. Sinha, and P. K. Srimani, "Fast Parallel Algorithms for Binary multiplication and Their Implementation on Systolic Architectures,"

IEEE Transactions On Computers, vol. 38, no. 3, pp. 424~431, Mar. 1989.

[6] A. Hiasat, "New Memoryless, mod ($2^n \pm 1$) Residue Multiplier," *Electronics Letters*, vol. 28, no. 3, pp. 314~315, Jan. 1992.

[7] A. A. Hiasat, "New Efficient Structure for a Modular Multiplier for RNS," *IEEE Transactions On Computers*, vol. 49, no. 2, pp. 170~174, Febr. 2000.

[8] E. D. Di Claudio, F. Piazza, and G. Orlandi, "Fast Combinational Processors for DSP Applications," *IEEE Transactions On Computers*, vol. 44, no. 5, pp. 624~633, May 1995.

[9] A. Saed, M. Ahmadi, and G. A. Jullien, "Analog Digits: Bit Level Redundancy in a Binary Multiplier," *IEL Data Base*(0-7803-5148-7/98/$10.00ⓒ1998 IEEE), pp. 236~240, 1998.

[10] J. B. Shin, J. K. Kim, and H. L. Kwang, "Optimisation of Montgomery Modular Multiplication Algorithm for Systolic Arrays," *Electronics Letters*, vol. 34, no. 19, pp. 1830~1831, Sept. 1998.

[11] C. Efstathiou, D. Nikolos, and J. Kalamatianos, "Area-Time Efficient Modulo 2^n-1 Adder Design," *IEEE Transactions On Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 41, no. 7, pp. 463~467, July 1994.

[12] R. F. Coughlin and F. F. Driscoll, *Operational-Amplifiers & Linear Integrated Circuits*, Prentice Hall, pp. 434~436, 1998.

[13] T. L. Floyd, *Digital Logic Fundamentals*, C. E. Merrill Publishing Company, pp. 205~212, 1977.

[14] C. D. Walter, "Systolic Modular Multiplication," *IEEE Transactions On Computers*, vol. 42, no. 3, pp. 376~378, Mar. 1993.

[15] B. Arazi, "Digital Signature Device," *US Patent*, #5448639, 1995.

[16] 김 광조, 김 철 "정보보호 이론의 발전," 전자공학회지, 제 21권 5호, pp. 443~456, 1994년 5월.

저 자 소 개

李 勳 圭(正會員)

1968년 5월 11일생, 1997년 2월 광운대학교 전파공학 학사. 1999년 2월 광운대학교 전파공학석사. 1999년 1월~1999년 6월 : (주) 데이터시큐어 인턴연구원. 1999년 7월~1999년 8월 : (주) 새로택 신입연구원. 1999년 8월~1999년 12월 : 가톨릭대학교 의과학원 MRI 연구소 위촉연구원. 2000년 1월~현재 : 광운대학교 정보통신보안연구실 연구원. <주관심분야 : CAD 및 VLSI 설계, 스위칭이론, 암호화알고리즘>

金 鐵(正會員)

1959년 10월 13일생. 1984년 2월 연세대학교 수학과 학사. 1989년 12월 미국 NC 주립대 석사, 박사. 1988년 8월~1990년 6월 미국 Shaw Univ. 수학 및 정보학과 전임강사. 1990년 8월~1990년 2월 미국 University of South Dakota 수학과 조교수. 1991년 3월~현재 : 광운대학교 수학과 정보통신보안 연구실 교수. <주관심분야 : 정보통신보안이론과 응용, 암호화알고리즘, 암호해독론>