논문 2006-43CI-3-9

합성체를 이용한 유한체의 역원 계산 알고리즘 구현

(An Implementation on the Computing Algorithm for Inverse Finite Field using Composite Field)

노 진 수*, 이 강 현**

(Jin Soo NOH and Kang Hyeon RHEE)

요 약

최근 멀티미디어 통신 시스템에서 유한체는 암호화 알고리즘에 적용되어지고 있으며. 특히 타원곡선 알고리즘 및 리드 솔로몬 등의 에러정정 코드는 유한체 상에서 정의 되어진다. 또한 많은 응용분야에서 유한체 연산의 실시간 처리를 요하므로 유한체 연산을 위한 전용 하드웨어 설계가 필요하게 되었고 이에 대한 많은 연구가 수행되어지고 있다. 본 논문에서는 합성체(Composite Field)를 이용하여 $GF(2^8)$의 유한체의 역원을 계산할 수 있는 알고리즘을 제시하고 이를 하드웨어로 구현하여 현재 사용되어 있는 'Itoh and Tsujii' 하드웨어 구조와 면적 및 계산 속도의 성능을 비교 하였다. 또한 AES의 SubBytes 블록에 이를 삽입하여 FPGA 에뮬레이터 보드 상에서 구현하여 성능평가를 통하여 제시된 알고리즘의 우수성을 확인하였다.

Abstract

Recently, Finite field is applied the cryptography in the modern multimedia communication. Especially, block codes such as Elliptic Curve Cryptosystem and Reed-Solomon code among the error correcting codes are defined with finite field. Also, finite field algorithm is conducting the research actively because many kind of application parts need the real time operating ability therefore the exclusive hardware have been implementing. In this paper, we proposed the inverse finite field algorithm over $GF(2^8)$ using finite composite field and implemented in a hardware, and then compare this hardware with the currently used 'Itoh and Tsujii' hardware in respect to structure, area and computation time. Furthermore, this hardware was inserted into the AES SubBytes block and implemented on FPGA emulator board to confirm that the superiority of the proposed algorithm through the performance evaluation.

Keywords: Finite Field, Composite Field, Cryptography, Subbytes Block, AES, FPGA

I. 서 론

최근 유한체(Finite field or Galois field)는 스위칭 이론, 디지털 신호처리 및 화상처리, 디지털 통신의 암호화 및 해독화를 요하는 보안 통신 등에서 많이 응용되고 있다.

특히 오류정정부호 중 BCH 부호나 Reed-Solomon 부호와 같은 블록부호는 유한체 상에서 정의되며, CD(Compact Disc), DAT(Digital Audio Tape), 타원곡선 알고리즘[1] 등의 부호화 및 복호화에 $GF(2^N)$의 산술연산이 적용되어진다. 그리고 현재 많은 응용분야에서 유한체 연산의 실시간 처리 능력을 필요로 하고 있으며, 이에 따라 유한체 연산을 위한 전용 하드웨어 설계에 대한 연구가 많은 부분에서 진행 되어지고 있다[2,3].

유한체는 사칙연산이 정의되는 유한개의 원소를 갖는 필드이며 모든 소수 P와 양수 N에 대해서 P^N개의 원소를 갖는 하나의 필드만이 존재한다. 이 필드를 유한체 필드라고 부르며 $GF(P^N)$으로 나타낸다. 유한체 상에서의 연산은 가·감산과 승산, 그리고 제산이 있으나 디지털 시스템은 유한체의 개수가 2의 승수인 $GF(2^N)$에서 이루어진다. 유한체의 역원(inverse finite field)의 계산은 크게 유한체 제산기를 이용하는 방법과 승산기를 이용하

* 학생회원, ** 평생회원, 조선대학교 전자공학과
(Dept. of Electronic Engineering, Chosun University)

접수일자: 2006년3월24일, 수정완료일: 2006년4월28일

는 방법으로 나누어진다. 제산기를 이용하는 방법은 빠른 동작 속도를 가지나 하드웨어의 면적이 커지며, 승산기를 이용한 방법은 하드웨어의 면적은 작아지나 계산에 많은 시간이 소모된다.

본 논문에서는 합성체(Composite Field)를 이용하여 $GF(2^8)$의 유한체의 역원을 계산할 수 있는 알고리즘을 제시하고 이를 하드웨어로 구현하여 현재 사용되어지는 'Itoh and Tsujii'[4] 하드웨어 구조와의 면적 및 계산 속도의 성능을 비교 하였다. 또한 AES의 SubBytes 블록[5,6]에 이를 삽입하여 Altera FLEX10K FPGA 에뮬레이터 보드에 구현하여 제시된 알고리즘의 회로가 정상적으로 동작함을 확인하였다.

본 논문의 구성은 Ⅱ장에서 유한체의 역원계산 알고리즘에 대하여 알아보고, Ⅲ장에서는 본 논문에서 제안한 합성체를 이용한 유한체의 역원계산 알고리즘을 구현하였다. Ⅳ장에서는 회로구현 및 동작특성과 성능을 비교하였으며, Ⅴ장에서 결론을 맺는다.

Ⅱ. 유한체의 역원계산 알고리즘

유한체의 역수를 구하는 대표적인 방법으로는 Fermat theorem[7], 확장 유클리드 알고리즘[5]을 이용하는 방법 그리고 유한체 승산[8]을 이용하는 방법, 그리고 본 논문에서 제안하는 합성체를 이용하는 방법이 있다.

1. Fermat 알고리즘

Fermat theorem은 임의의 수 a와 p가 서로소(relatively prime)의 관계를 가질 때 식 (1)과 같이 표현된다.

$$\begin{cases} a^p = a \bmod p, a \in GF(p) \\ a^{-1} = a^{p-2} = a^{2^N-2} \bmod 2^N \end{cases}, p = 2^N \quad (1)$$

Fermat 알고리즘으로 역수를 구할 경우에 $GF(2^N)$상에서 $N-1$번의 유한체 승산과 $N-1$번의 유한체 제곱의 연산이 필요하므로 계산량이 많아진다는 단점이 있다.

2. 확장 유클리드 알고리즘

두 다항식 $a(x)$와 $b(x)$, $\{\deg(a(x)) < \deg(b(x))\}$의 GCD(Greatest Common Divisor)를 유클리드 알고리즘을 이용하여 계산하는 과정은 식 (2)와 같다.

$$\begin{aligned} r_{-1}(x) &= b(x), \quad r_0(x) = a(x) \\ r_{-1}(x) &= q_0(x)\, r_0(x) + r_1(x) \\ r_0(x) &= q_1(x)\, r_1(x) + r_2(x) \\ &\vdots \quad \vdots \\ r_{k-1}(x) &= q_k(x)\, r_k(x) + r_{k+1}(x) \end{aligned} \quad (2)$$

식 (2)에서의 GCD를 $g(x)$라 하면, 확장 유클리드 알고리즘에 의해서 식 (3)과 같은 관계가 성립한다.

$$\begin{aligned} &s(x)b(x) + t(x)a(x) = g(x), \\ &g(x) = GCD(a(x), b(x)) \end{aligned} \quad (3)$$

아울러 식 (3)의 $s(x)$와 $t(x)$는 식 (4)와 같은 반복적인 계산으로 구할 수 있다.

$$\begin{aligned} &s_{-1}(x) = 1, \quad s_0(x) = 0 \\ &t_{-1}(x) = 0, \quad t_0(x) = 1 \\ &s_i(x) = q_{i-1}(x)\, s_{i-1}(x) + s_{i-2}(x) \\ &\qquad (-1 \le i \le k+1) \\ &t_i(x) = q_{i-1}(x)\, t_{i-1}(x) + t_{i-2}(x) \end{aligned}$$

식 (4)에서 $q_i(x)$는 식 (2)에서 계산되어지며, 식 (2)의 $r_{k+1}=0$일 때까지 $t(x)$를 반복적으로 계산한 $t_k(x)$는 $b(x)$가 되고, $t_{k-1}(x)$는 $a^{-1}(x)$가 된다.
여기서 식 (3)을 다시 쓰면 식(5)가 된다.

$$t(x)a(x) = g(x) \bmod b(x) \quad (5)$$

이때, $b(x)$가 기약 다항식(irreducible polynomial)이므로 항상 $g(x)=1$이다. 따라서 $t(x)a(x)=1 \bmod b(x)$ 이므로 식 (6)을 통해 역원이 계산된다.

$$t(x) = \frac{1}{a(x)} \bmod b(x) \quad (6)$$

3. $GF(2^8)$ 유한체 승산기를 사용한 역원 계산

$GF(2^8)$ 필드에 유한체의 원소인 y가 입력되어 질 때 y의 역원은 식(7)과 같이 유도 되어진다.

$$\begin{aligned} &y^{2^8-1} = y^{255} = 1 \\ &y^{-1} = y^{-1} \cdot y^{255} = y^{254} \end{aligned} \quad (7)$$

4. 합성체

이차의 합성체를 이용하여 유한체의 역원을 구하는 알고리즘은 Cristof Paar[9]에 의해 증명되었으며 2개의 유

한체 ζ 와 δ가 $\delta=\zeta^{-1}$의 관계를 가질 때 식 (8)에 의해서 유한체의 역원을 구할 수 있다.

$$\zeta = Z_1\gamma + Z_0$$
$$\delta = D_1\gamma + D_0$$

$$D_0 = (Z_1A + Z_0)F^{-1} \quad (8)$$
$$D_1 = Z_1F^{-1}$$

$$F = Z_1^2B + Z_1Z_0A + Z_0^2$$

본 논문에서는 식 (8)을 기본식으로 하여 분산산술(Distributed Arithmetic)에 의하여 유한체의 역원 계산 알고리즘 하드웨어 구조를 제안하였다. 이에 대한 설명은 3장에서 자세히 다루겠다.

Ⅲ. 합성체를 이용한 역원 알고리즘

유한체 *GF(256)*은 각각 $GF(2^8)$, $GF(4^4)$ 또는 $GF(16^2)$로 나타낼 수 있다[10,11]. 본 논문에서는 $GF(16^2)$을 $GF(P^{NM})$의 합성체로 사용하여 분산산술을 적용시켜 유한체 역원 계산 알고리즘을 설계한다.

1. 유한체 역원 유도과정

$\zeta \in GF((2^4)^2)$일 때 식 (9)와 같이 나타 낼 수 있다.

$$\zeta = Z_1\gamma + Z_0 \quad (9)$$

이때, Y는 $GF(16^2)$ 의 기약다항식 $P(x)$의 근이며 $Z_1, Z_2 \in GF(2^N)$ 을 만족한다. 식 (10)은 기약다항식 $P(x)$이다.

$$P(x) = x^2 + Ax + B \quad (10)$$

식 (9), (10)으로부터 ζ^{-1}은 식 (11)과 같이 유도된다.

$$\zeta^{-1} = \delta = D_1\gamma + D_0$$
$$\zeta \cdot \delta = 1$$
$$= D_1Z_1\gamma^2 + (D_1Z_0 + D_0Z_1)\gamma + D_0Z_0 \quad (11)$$
$$P(\gamma) = 0,$$
$$\gamma^2 = A\gamma + B$$

식 (12), (13)은 유한체의 역원인 D_1과 D_0를 구하는 과정으로 식 (11)로부터 유도된다.

$$(D_1Z_0 + D_0Z_1 + AD_1Z_1)\gamma + (D_0Z_0 + BD_1Z_1) = 0\gamma + 1 \quad (12)$$

$$D_1Z_0 + D_0Z_1 + AD_1Z_1 = 0, D_0Z_0 + BD_1Z_1 = 1$$
$$D_1 = D_0Z_1(Z_0 + AZ_1)^{-1}, D_0 = D_1Z_1^{-1}(Z_0 + AZ_1)$$
$$D_0(Z_0 + BZ_1^2(Z_0 + AZ_1)^{-1}) = 1 \quad (13)$$
$$D_1Z_1^{-1}(Z_0^2 + AZ_1Z_0 + BZ_1^2) = 1$$

결과적으로 합성체를 이용한 유한체의 역원 계산과정은 식 (14)와 같이 유도된다.

$$D_1 = Z_1 \cdot F^{-1}, D_0 = (Z_0 + AZ_1) \cdot F^{-1}$$
$$F = Z_0^2 + AZ_1Z_0 + BZ_1^2 \quad (14)$$

2. 제안된 역원 알고리즘의 블록도

식 (14)로부터 그림 1과 같은 유한체의 역원 계산 알고리즘을 구현하였다. 여기에서 사용된 합성체의 기약다항식은 $P(x) = x^2 + x + \beta^{14}$을 사용하였으며, β^{14}는 $\beta^3 + 1 = y^3 + 1$ 와같이 계산된다.

그림 1에서 S 블록은 입력으로 들어가는 확장체(Extension field)를 합성체로 변화시켜 주는 배열 블록이며, R 블록은 합성체를 확장체로 변화시켜주는 배열 블록으로 $R = S^{-1}$의 관계를 가진다. 그림 2는 S와 R블록의 값을 나타낸다[1].

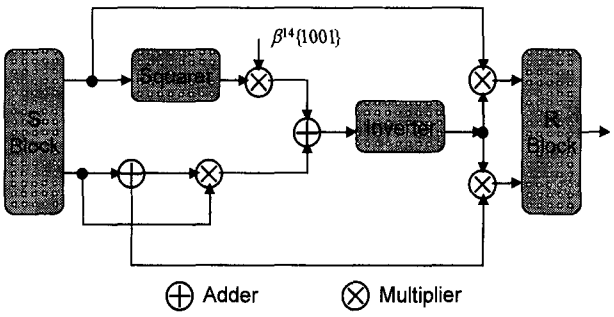


그림 1. 합성체를 이용한 역원계산 알고리즘 블록도
Fig. 1. Block diagram of Inverse computing algorithm using composite field.

$$S = \begin{pmatrix} 0&1&0&0&0&1&0&1 \\ 0&0&1&1&0&0&0&0 \\ 0&1&0&0&1&0&1&1 \\ 0&0&0&0&0&1&0&1 \\ 1&0&0&1&1&0&1&1 \\ 0&0&1&0&0&0&1&0 \\ 0&1&1&0&0&1&0&1 \\ 0&1&1&0&1&1&0&1 \end{pmatrix} \quad R = \begin{pmatrix} 0&1&1&1&1&0&1&0 \\ 1&0&0&1&0&0&0&0 \\ 1&0&0&0&0&0&1&0 \\ 1&1&0&0&0&0&1&0 \\ 0&0&0&0&0&0&1&1 \\ 0&0&1&0&0&1&0&1 \\ 1&0&0&0&0&1&1&0 \\ 0&0&1&1&0&1&0&1 \end{pmatrix}$$

그림 2. S와 R 블록 계수
Fig. 2. Block coefficient of S and R.

Ⅳ. FPGA 구현 및 실행 결과

제안된 유한체의 역원 알고리즘이 정상적으로 동작하는지 확인하기 위하여 Altera FLEX10K 상에 구현하였다. 구현 알고리즘의 동작 확인은 AES의 SubBytes Transform 회로를 설계하여 역원 계산 부분을 롬 테이블이 아닌 제안된 역원 계산 회로로 대치하여 구현하였으며 블록도는 그림 3과 같다.

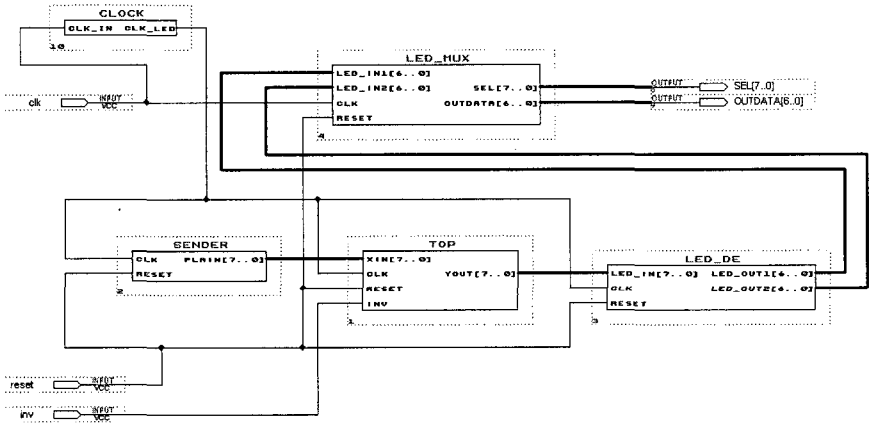
3장에서 유도된 식 (14)와 제안된 회로의 성능을 측정하기 위하여 VHDL을 사용하여 구현하였으며, 시뮬레이션 결과 값이 정상적으로 출력됨을 확인 할 수 있었다. 결과 값의 비교는 AES의 S-box의 역원 표와 본 논문에서 설계한 유한체 역원 계산 알고리즘을 사용하였다. 그림 4는 기능 시뮬레이션 결과 값으로 한 클럭 지연 후 'YOUT' 핀에서 출력되는 값이 AES의 S-box의 역원 표와 동일한 값임을 확인할 수 있으며 S-box의 값을 입력했을 때 'output' 핀에서 역원 값이 정확히 출력됨을 확인할 수 있었다.

그림 5는 FPGA 에뮬레이터 보드 상에 구현된 그림으로 'OUTPUT DATA' 부분에서 출력 데이터가 세븐 세그먼트로 출력되어지고 'RESET KEY'를 통하여서 회로가 초기화 된다. 'INVERSE KEY'는 회로의 역변환을 제어하는 부분이다.

그림 6은 Synopsys 에서 합성한 게이트레벨 회로이며, Synopsys에서 제공하는 core_slow.db를 사용하여 합성하였을 때 전체 셀 면적이 4,593.87이었다.

표 1은 본 논문에서 제안된 역원계산 알고리즘과 'Itho and Tsujii'[4] 알고리즘과의 성능 비교 값이며 비교 대상은 셀 면적과 회로 동작 속도이다.

표 1에서 본 논문에서 제안된 알고리즘이 'Itoh and Tsujii' 알고리즘에 비해 셀 면적은 61.60% 감소하였으며, 24.97%의 동작 속도 향상을 보였다.



CLOCK : Clock division circuit(40:1) *Top : SubBytes(Inverse SubBytes) Transform circuit*
LED_MUX : Seven segment control switch *LED_DE : Seven segment signal decode* *SENDER : Input signal data*

그림 3. FPGA 구현 블럭도
Fig. 3. Block diagram of FPGA implementation.

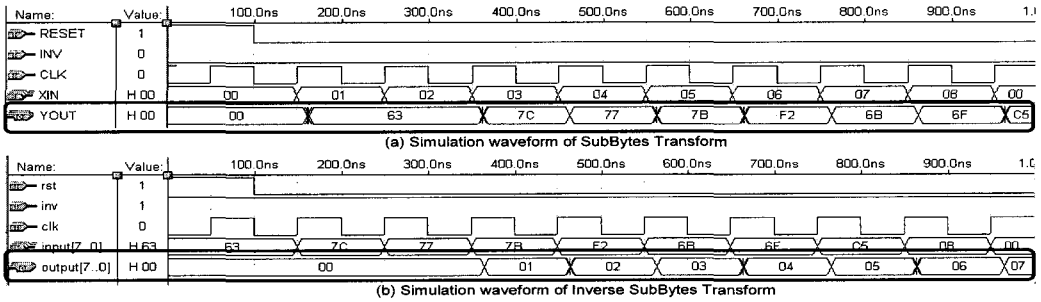


그림 4. 기능 시뮬레이션 결과 값
Fig. 4. The result waveform of the function simulation.

표 1. 제안된 회로와 [4]의 성능 비교
Table 1. Comparison results with proposed circuit and [4].

Design Architecture	Cell Area	Data Arrival Time
'Itoh and Tsujii'[4]	12,030.64	19.96[ns]
Proposed Design	4,593.87	15.16[ns]

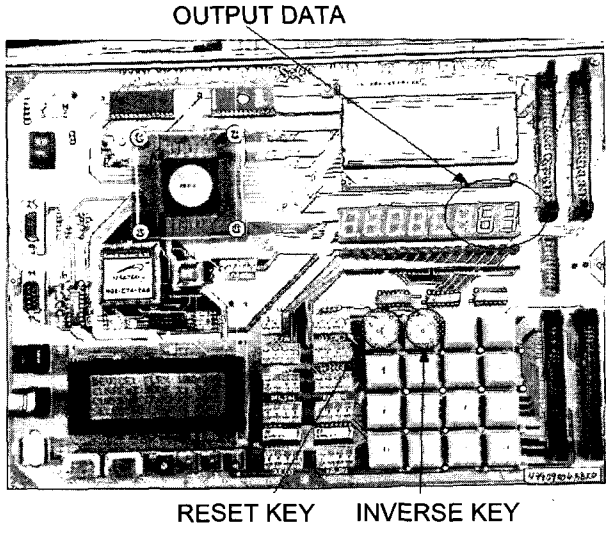


그림 5. FLEX10K FPGA 구현
Fig. 5. FLEX10K FPGA implementation.

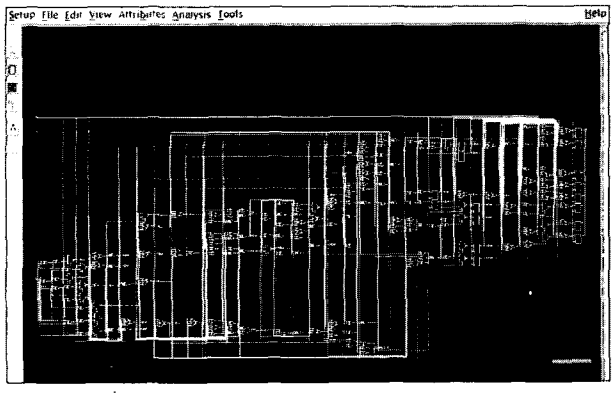


그림 6. 게이트레벨 합성 회로
Fig. 6. Gate level synthesis circuit.

V. 결 론

본 논문에서는 디지털 통신의 암호화 및 보안 통신 등에서 사용되고 있는 유한체의 역원을 계산 할 수 있는 알고리즘과 하드웨어 구조를 제안하였다. 기존의 유한체 역원 계산 알고리즘은 유한체 승산 과정을 반복하여 역원을 계산하는 방법으로 알고리즘 및 하드웨어 구조가 간단한 장점을 가졌으나 동작속도 및 회로의 면적이 증가되었다. 제안된 알고리즘은 유한체의 역원 계산과정을 합성체와 확장체로 나누워 계산 하는 방식으로 알고리즘의 복잡도는 증가하나 하드웨어 구현 시 동작속도 및 하드웨어 면적에서 기존의 알고리즘보다 성능이 향상됨을 알 수 있다. 하지만, 본 논문에서 제안된 알고리즘은 유한체 차수의 증가에 따라 회로 구현의 복잡도 또한 증가하는데 이를 해결하기 위하여 향후 과제로는 제안된 하드웨어 구조에서 S와 R 블록의 확장체 부분을 제거함으로 하드웨어의 복잡도를 감소시킬 수 있는 연구가 필요하다.

참 고 문 헌

[1] Michael Rosing, "Implementing Elliptic Curve Cryptography," Oreilly&Associates Inc, 1998.
[2] C. Paar. "Efficient VLSI Architectures for Bit-Parallel Computation in Galois Fields," Ph.D. thesis, Institute for Experimental Mathematics, University of Essen, 1994.
[3] Vincent Rijmen, "Efficient implementation of the Rijndael S-box," http://www.esat.kuleuven.ac.be/~rijmen/rijndael/sbox.pdf, 2000.
[4] LSI Design Contest 2004, http://www.ie.u-ryukyu .ac.jp/~wada/design04 /contest2004e .html
[5] Chin-Pin Su et. al, "A Highly Efficient AES Cipher Chip," ASP-DAC2003, pp.561-562, Jan 2003.
[6] Satoh A., Morioka, S., Takano, K. and Munetoh, S. "A Compact Rijndael Hardware Architecture with S-Box Optimization," Advances in Cryptology - ASIACRYPT, LNCS, Vol.2248, pp.239-254, 2001.
[7] "Announcing the ADVANCED ENCRYPTION STANDARD," Federal Information Processing Standards Publication 197, Nov. 26, 2001.
[8] T. Itoh and S. Tsujii, "A fast algorithm for computing multiplicative inverses in GF(2m) using normal basis," J. Society for Electronic Communication, pp. 31-36, 1986.
[9] C. Paar, P. Fleischmann and S. Pedro, "Fast Arithmetic for Public-Key Algorithms in Galois Fields with Composite Exponents," IEEE Trans. on Computers., vol. 48, no. 10, pp. 1025-1034, Oct. 1999.
[10] Cillian O'Driscoll, "Hardware implementation aspects of the Rijndael block cipher," Master's thesis, University Coll Cork, Europe, Oct. 2001.
[11] E. D Mastrovito, "VLSI Architecture for Computation in Galois Field," Ph.D. Thesis, Linkping Univ., Sweden, 1991.

저 자 소 개

노 진 수(학생회원)

2002년 조선대학교 전자공학과 학사졸업.

2004년 조선대학교 전자공학과 석사졸업.

2006년 조선대학교 전자공학과 박사과정.

<주관심분야 : UWB, 생체인식, 양자컴퓨팅>

이 강 현(평생회원)-교신저자

1979년, 1981년 조선대학교 전자공학과 공학사 및 석사

1991년 아주대학교 대학원 공학박사

1977년~현재 조선대학교 교수

1991년, 1994년 미 스탠포드대 CRC 협동연구원.

1996년 호주시드니대 SEDAL 객원교수

2000년~현재 한국 멀티미디어 기술사 협회 이사

2002년 영국 런던대 객원 교수

2002년 대한전자공학회 멀티미디어연구회전문위원장

2003년 한국 인터넷 방송/TV 학회 부회장

2003년~현재 대한전자공학회 홍보이사

2005년~현재 조선대학교 RIS 사업단장

<주관심분야 : 멀티미디어 시스템 설계, Ubiquitous convergence>