

다양한 유전 연산자를 이용한 저전력 오류 정정 코드 설계

Design of Low Power Error Correcting Code Using Various Genetic Operators

이희성 · 홍성준 · 안성제 · 김은태*

Heesung Lee, Sungjun Hong, Sungje An, and Euntai Kim*

연세대학교 전기전자공학부

요 약

저전력 환경에서의 메모리 집적도가 증가함에 따라 메모리는 soft error에 매우 민감해졌다. 오류 정정 코드는 일반적으로 양산 이후 메모리의 soft error를 수정하기 위해서 사용된다. 본 논문에서는 새로운 저전력 오류 정정 코드의 설계방법을 제안한다. 오류 정정 코드의 전력소비는 parity check 행렬의 선택을 통해 최소화 될 수 있다. 따라서 오류 정정 코드의 설계는 비선형 최적화 문제로 포함되는데 우리는 다양한 유전 연산자를 포함하는 유전자 알고리즘을 이용하여 이 문제를 해결한다. 제안하는 방법을 Hamming code와 Hsiao code에 적용하여 그 효율성을 입증하였다

키워드 : 오류 정정 코드, 저전력, 유전자 알고리즘, Hamming code, Hsiao code

Abstract

The memory is very sensitive to the soft error because the integration of the memory increases under low power environment. Error correcting codes (ECCs) are commonly used to protect against the soft errors. This paper proposes a new genetic ECC design method which reduces power consumption. Power is minimized using the degrees of freedom in selecting the parity check matrix of the ECCs. Therefore, the genetic algorithm which has the novel genetic operators tailored for this formulation is employed to solve the non-linear power optimization problem. Experiments are performed with Hamming code and Hsiao code to illustrate the performance of the proposed method.

Key Words : Error correcting code, low power, genetic algorithm, Hamming code, Hsiao code

1. 서 론

저전압 환경에서의 메모리 집적도가 증가함에 따라 메모리는 soft error에 매우 민감해졌다[1]. 따라서 양산 이후 메모리 오류를 수정하는 방법들이 활발하게 연구 중이다. 이중 오류 정정 코드(Error Correcting Code, ECC)를 이용한 방법은 fault self-detecting/repairing 방법으로 최근 가장 각광받는 방법 중 하나이다. 메모리의 soft error는 양산 이전의 메모리 테스트 과정에서 검출 및 수정이 어렵기 때문에 메모리의 안정적인 동작을 보장하기 위해 ECC 기술이 필수적이다. 따라서 이러한 ECC를 통해 메모리에서 시스템의 안정도와 데이터의 보존성의 향상을 기대할 수 있다. 하지만 메모리의 읽고 쓰는 과정에서 ECC는 매번 동작하여야 하기 때문에 ECC의 전력소비가 크면 메모리에 큰 부하를 줄 수 있다. 또한 전력 소비가 회로 설계의 중요한 요소로 부각함에 따라 전력 소비가 적은 ECC를 개발하기 위하여 많은 연구가 있었다[2, 3]. 하지만, 이러한 연구들은 ECC만이 갖는 특성을 간과하고 일반적인 회로에서의 전력 감소

방법을 ECC에 적용하였다[4]. 단지 몇 가지 연구가 ECC의 특성을 반영하기 위하여 유전자 알고리즘을 이용한 설계 방법을 제안하였지만 이러한 연구들도 유전자 알고리즘의 장점과 유전 연산자들을 충분히 이용하지 못하였기 때문에 성능의 큰 향상을 보이지 못하였다[5]. 우리의 이전 연구에서는 이러한 문제점을 해결하기 위하여 ECC의 좋은 형질들을 잘 유지하고 전달할 수 있는 유전자 연산자들을 이용하여 저전력 ECC를 설계 하였다[5, 6]. 본 논문에서는 ECC의 설계에 적합한 여러 가지 유전 연산자들을 우리의 이전 연구에 추가하여 ECC의 전력 소비를 기존 연구들 보다 크게 줄인다. 이것을 통해 우리는 저전압 환경에서의 메모리의 생존성과 효율성을 높일 수 있다.

2. 다양한 유전 연산자를 이용한 저전력 오류 정정 코드 설계

2.1 염색체

본 논문에서는 Hamming code와 Hsiao code를 이용하여 ECC를 설계하였다. 이 두 code는 가장 보편적이고 많이 사용되는 ECC이다. 본 논문에서 Hamming code에서는 다음과 같은 구조를 갖는 염색체를 사용한다.

접수일자 : 2008년 9월 1일

완료일자 : 2009년 2월 5일

* 교신 저자

This work was supported by the Ministry of Commerce, Industry and Energy of Korea.

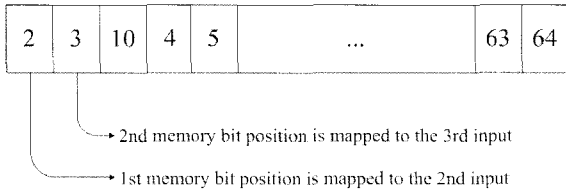


그림 1. Hamming code를 위해 사용된 염색체의 구조.
Fig. 1. Structure of chromosome for Hamming code.

염색체는 parity check generator 행렬의 column들의 순열을 표현하며 메모리의 입력 bit의 위치로 인코딩된다. 예를 들어 64bit 메모리에서 그림 1과 같은 “2, 3, 10, 4, 5, ..., 63, 64”는 염색체의 첫 번째 유전자는 메모리의 첫 번째 입력 bit에는 ECC 회로의 두 번째 bit가 연결됨을 두 번째 유전자는 메모리의 두 번째 입력 bit에는 ECC 회로의 세 번째 bit가 연결됨을 표현하고 염색체의 나머지 유전자들도 이와 같은 방식으로 메모리와 ECC 회로의 연결됨을 표현한다.

Hsiao code는 모든 low-order odd weight column과 몇 개의 higher-order odd weight column으로 구성되기 때문에, Hsiao code를 위한 염색체는 Hamming code와 같은 parity check generator 행렬의 column들의 순열 뿐만 아니라 higher-order odd weight column들을 선택하는 부분도 포함하여야 한다. 64bit 메모리에서 Hsiao code는 8개의 weight-1, 56개의 weight-3 column들을 포함하고 있으며 56개의 weight-5 column중에서 선택된 8개의 weight-5 column도 포함하고 있다. Hsiao code를 위한 염색체는 다음과 같다.

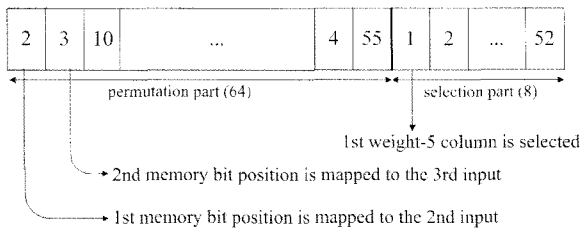


그림 2. Hsiao code를 위해 사용된 염색체의 구조.
Fig. 2. Structure of chromosome for Hsiao code.

그림 2와 같이 Hsiao code의 염색체의 왼쪽의 순열 파트(permutation part)는 56개의 weight-3 column들과 선택된 8개의 weight-5 column들의 배열을 표현한다. 오른쪽의 선택 파트(selection part)는 56개의 weight-5 column중에 선택된 8개의 column들을 나타낸다. 본 논문의 알고리즘을 다른 사이즈의 메모리에서 적용하기 위해서는 위에서 설명한 바와 같이 순열 파트는 odd weight들의 배열을 결정하고 선택 파트는 사용될 higher-odd wight들을 선정하며 그 크기만을 조정함으로써 적용이 가능하다[7].

2.2 유전 연산자

유전 연산자는 현재 세대의 훌륭한 성질을 가지고 있는 염색체들을 엮어서 다음 세대에 전달하는 기능을 갖는다. 일반적으로 유전 연산자로는 교차와 돌연변이가 사용된다. 교차는 부모 세대의 두 개체간의 염색체를 부분적으로 서로 바꿈으로써 자식 세대의 새로운 개체를 생성하는 것이다. 이때 부모의 특성이 자손에게 적절히 계승되어야 한다. 하지만 기존의 방법 [4]에서는 단순한 평균과 정렬을 이용한

교차를 사용하였다. 예를 들어 부모 염색체 “1, 3, 2, 4, 9, 7, 6, 8, 5”와 “4, 1, 2, 8, 7, 6, 9, 3, 5”를 교차하면 평균은 [2.5, 2.0, 2.0, 6.0, 8.0, 6.5, 7.5, 5.5, 5.0]이 될 것이고 이것을 정렬한 “3, 1, 2, 6, 9, 7, 8, 5, 4”를 자식 염색체로 사용한다. 하지만 예제에서와 같이 이 교차 연산자로는 부모 염색체들의 가장 중요한 정보인 parity check generator 행렬의 column들의 위치를 자식 염색체에 잘 전달하지 못한다. 또한 우리의 기존 연구[5, 6]에서는 이 문제를 해결하기 위하여 순환 교차만을 사용하였지만, 본 논문에서는 부모 염색체들의 위치 정보를 효과적으로 전달하기 위해 순환 교차에 부분 사상 교차를 추가하여 사용한다.

순환 교차는 각각의 부모 염색체들의 위치를 이용하여 자식 염색체를 생성한다. 예를 들어 다음과 같은 두 개의 부모 염색체들이 있다면

$$P_1 = (1, 3, 2, 4, 9, 7, 6, 8, 5) \\ P_2 = (4, 1, 2, 8, 7, 6, 9, 3, 5) \quad (1)$$

순환 교차는 P_1 의 첫 번째 유전자를 이용하여 자식 염색체를 만들기 시작한다.

$$O_1 = (1, X, X, X, X, X, X, X, X) \quad (2)$$

P_2 의 ‘4’는 처음으로 선택된 ‘1’바로 아래에 있기 때문에 자식 염색체의 두 번째로 고려될 유전자는 ‘4’이다.

$$O_1 = (1, X, X, 4, X, X, X, X, X) \quad (3)$$

같은 방식으로 P_2 의 ‘8’은 선택된 ‘4’ 바로 아래에 있기 때문에 다음으로 고려된다.

$$O_1 = (1, X, X, 4, X, X, 8, X, X) \quad (4)$$

같은 방식으로 ‘3’이 선택된다.

$$O_1 = (1, 3, X, 4, X, X, 8, X, X) \quad (5)$$

P_2 의 ‘1’은 ‘3’아래 있지만 이미 선택을 하였기 때문에 나머지 유전자 자리는 P_2 의 값들을 적용한다.

$$O_1 = (1, 3, 2, 4, 7, 6, 9, 8, 5) \quad (6)$$

다른 자식 염색체 O_2 는 P_2 를 먼저 이용하여 O_1 과 같은 방법으로 생성한다.

$$O_2 = (4, 1, 2, 8, 9, 7, 6, 3, 5) \quad (7)$$

부모 염색체의 유전자들의 위치는 parity generator 행렬의 column을 나타내며 위의 예제를 통해 순환 교차는 이것을 잘 보존함을 볼 수 있다. 이런 이유로 우리는 순환 교차를 이용하여 기존의 방법보다 부모 염색체들의 좋은 형질들을 다음 세대까지 유지 할 수 있다.

또한 본 연구에서는 부분 사상 교차를 순환 교차와 함께 사용하여 보다 전력 소비가 적은 ECC를 설계하였다. 부분 사상 교차도 순환 교차와 마찬가지로 부모 세대의 염색체의 위치를 보존하는 특성을 갖는다. 부분 사상 교차를 설명하기 위해 다음과 같은 부모 세대 염색체가 있다면

$$P_1 = (1, 2, 3|4, 5, 6, 7|8, 9) \\ P_2 = (4, 5, 2|1, 8, 7, 6|9, 3) \quad (8)$$

두 개의 cut point를 랜덤하게 결정한 다음 이 부분의 부모 염색체의 유전자들을 다음과 같이 맞바꾼다.

$$\begin{aligned} O_1 &= (X, X, X|1, 8, 7, 6|X, X) \\ O_2 &= (X, X, X|4, 5, 6, 7|X, X) \end{aligned} \quad (9)$$

그리고 cut point의 외곽 부분들로부터 다음과 같은 사상(mapping)을 정의한다.

$$\begin{bmatrix} 1 \leftrightarrow 4 \\ 8 \leftrightarrow 5 \\ 7 \leftrightarrow 6 \\ 6 \leftrightarrow 7 \end{bmatrix} \quad (10)$$

부모 염색체의 유전자들 중에서 사상이 없는 부분으로 자식 염색체의 유전자들을 우선 채운다.

$$\begin{aligned} O_1 &= (X, 2, 3|1, 8, 7, 6|X, 9) \\ O_2 &= (X, X, 2|4, 5, 6, 7|9, 3) \end{aligned} \quad (11)$$

마지막으로 O_1 의 첫 번째 유전자는 사상 $1 \leftrightarrow 4$ 때문에 '4'가 되고 O_2 의 첫 번째 유전자는 '1'이 된다. 같은 방법들을 이용하여 자식 염색체의 나머지 유전자들을 사상들을 이용해 채운다.

$$\begin{aligned} O_1 &= (4, 2, 3|1, 8, 7, 6|5, 9) \\ O_2 &= (1, 8, 2|4, 5, 6, 7|9, 3) \end{aligned} \quad (12)$$

위의 예제를 통해 부분 사상 교차도 순환 교차와 마찬가지로 부모 염색체의 유전자들의 위치인 parity generator 행렬의 column들을 잘 보존함을 볼 수 있다. 이런 이유로 우리는 부분 사상 교차를 순환 교차와 함께 이용하여 기존의 방법보다 부모 염색체들의 좋은 형질들을 다음 세대까지 유지할 수 있다.

Hsiao code에서는 염색체가 두 개의 파트를 가지고 있으므로 순환 교차와 부분 사상 교차를 순열 파트에 산술 교차는 선택 파트에 적용하는 혼성 구조의 교차 연산자를 이용하였다. 산술 교차는 부모 염색체들의 임의의 선행 결합을 자식 염색체로 사용한다. P_1 과 P_2 를 Hsiao code의 부모 염색체라고 한다면,

$$\begin{aligned} P_1 &= (A_1 A_2 \dots A_{64} a_1 a_2 \dots a_8) \\ P_2 &= (B_1 B_2 \dots B_{64} b_1 b_2 \dots b_8) \end{aligned} \quad (13)$$

여기서 대문자는 순열 파트, 소문자는 선택 파트를 나타낸다. 본 논문에서 사용한 혼성 구조의 교차 연산자는 우선 0에서 1사이의 랜덤 변수 r 을 발생시킨 후 다음과 같은 식을 이용하여 자식 염색체 O_1 과 O_2 를 생성한다.

$$\begin{aligned} O_1 &= (C_1 C_2 \dots C_{64} c_1 c_2 \dots c_8) \\ O_2 &= (D_1 D_2 \dots D_{64} d_1 d_2 \dots d_8) \end{aligned} \quad (14)$$

여기서,

$$\begin{aligned} (C_1 C_2 \dots C_{64}) &= (A_1 A_2 \dots A_{64}) * (B_1 B_2 \dots B_{64}) \\ (D_1 D_2 \dots D_{64}) &= (B_1 B_2 \dots B_{64}) * (A_1 A_2 \dots A_{64}) \end{aligned} \quad (15)$$

이고

$$\begin{aligned} c_i &= [ra_i + (1-r)b_i] \\ d_i &= [rb_i + (1-r)a_i] \text{ for } i = 1, 2, \dots, 8 \end{aligned} \quad (16)$$

이다. 위의 식에서 *는 순환 교차와 부분 사상 교차를 나타내며 $[\cdot]$ 는 가우스 기호를 나타낸다.

돌연변이는 개체의 각 유전자에 대하여 돌연변이 확률을 적용하여 그 값을 바꾸는 것으로 국소적인 랜덤탐색의 일종

이다. 교차 연산이 집단의 개체들이 현재 가지고 있는 정보들을 이용해서 결합하고 탐색하는 것에 비해 돌연변이 과정은 현재 집단에 존재하지 않는 새로운 정보를 제공해 준다. 본 논문에서는 Hamming code와 Hsiao code의 순열 파트에서 돌연변이는 두 개의 유전자를 선택하여 맞바꾸는 방법을 이용하였고 Hsiao code의 선택 파트에서는 임의로 그 값을 바꾸었고 만약 그 값이 중복된다면 그것이 없어질 때까지 돌연변이를 반복하였다. 하지만 이러한 돌연변이 연산의 경우 자연계에서 돌연변이 확률이 낮은 것처럼 일반적으로 매우 낮은 확률로 발생시킨다.

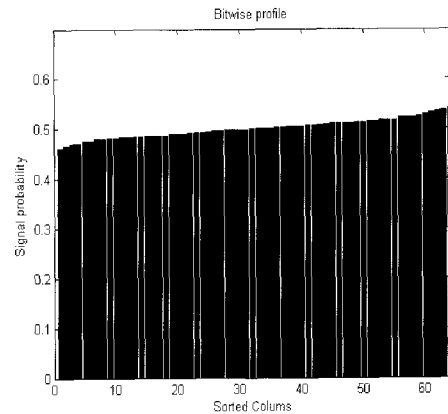
2.3 적합도 함수

본 연구의 목적은 전력 소비를 최소화 하는 ECC를 찾는 것이다. ECC check 회로는 XOR gate들의 인-아웃 상태가 변할 때 가장 많은 전력 소비를 한다. 따라서 염색체의 적합도는 gate들의 인-아웃 상태의 transition에 반비례 한다. XOR gate들의 인-아웃 상태의 transition 숫자를 t 이라 했을 때 유전자 염색체의 적합도 함수는 다음과 같이 정할 수 있다.

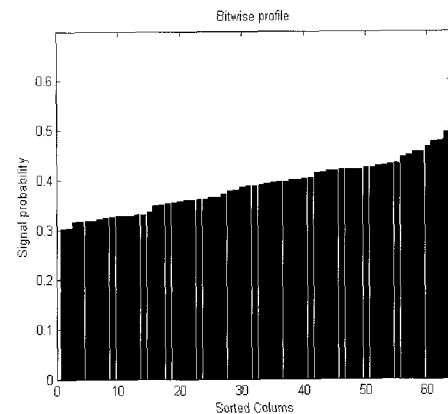
$$fitness = -1 \times t \quad (17)$$

3. 실험

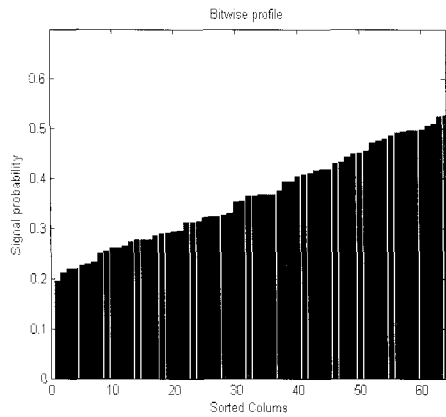
본 논문에서는 그림 3과 같은 특징을 갖는 64-bit 메모리 데이터를 이용하여 성능을 평가하였다.



(a) dataset1



(b) dataset2



(c) dataset3

그림 3. 메모리 데이터의 bitwise profile
Fig. 3. Bitwise profile of memory data.

dataset1은 0과 1의 나타나는 비율이 모든 bit에서 비슷하다. dataset3에서는 0이 1보다 상위 bit에서 많이 나타나며 하위 bit로 갈수록 나타나는 비율이 비슷해진다. dataset2는 dataset1과 dataset3의 중간 특징을 갖는다. 표 1은 사용한 유전 파라미터를 나타낸다.

표 1. 유전자 알고리즘의 파라미터들.

Table 1. Parameters for genetic algorithm.

Parameters	Value
Max. generation number	100
Crossover rate	0.6
Mutation rate	0.05
Population size	300

본 논문의 제안하는 알고리즘의 우수성을 보여주기 위하여 기존의 방법들[4, 5]로 설계된 Hamming code와 Hsiao code와의 소비 전력을 비교하였다. 실험의 신뢰도를 높이기 위해 우리는 초기 해 집단을 다르게 해서 10번의 실험을 반복하였고 그 결과 값들을 표 2, 3, 4에 나타내었다.

표 2. Dataset1의 전력 소비 비교.

Table 2. Comparison with power consumption in dataset1.

Hamming code				
Methods	best	worst	average	std
Gohsh [4]	127031	127463	127223.2	135.7
CX [5]	127010	127306	127167.1	99.4
PMX	127010	127366	127217.0	95.5
CX+PMX	126854	127303	127111.8	143.8

Hsiao code				
Methods	best	worst	average	std
Gohsh [4]	107131	107513	107315.9	111.6
CX [5]	107089	107313	107192.2	86.5
PMX	107030	107590	107294.5	169.1
CX+PMX	107017	107307	107189.1	99.9

표 3. Dataset2의 전력 소비 비교.

Table 3. Comparison with power consumption in dataset2.

Hamming code				
Methods	best	worst	average	std
Gohsh [4]	125979	126357	126163.5	150.2
CX [5]	125917	126271	126089.1	115.8
PMX	125713	126254	126019.3	142.2
CX+PMX	125439	126185	125975.4	236.5

Hsiao code				
Methods	best	worst	average	std
Gohsh [4]	103067	103579	103290.8	153.2
CX [5]	102975	103283	103128.0	144.9
PMX	103046	103427	103234.4	135.2
CX+PMX	102902	103258	103097.5	133.2

표 4. Dataset3의 전력 소비 비교.

Table 4. Comparison with power consumption in dataset3.

Hamming code				
Methods	best	worst	average	std
Gohsh [4]	123150	123575	123411.1	197.1
CX [5]	122917	123503	123243.6	194.5
PMX	123001	123558	123248.8	203.7
CX+PMX	122893	123435	123160.2	179.8

Hsiao code				
Methods	best	worst	average	std
Gohsh [4]	99325	99780	99643.4	136.6
CX [5]	99276	99666	99495.1	141.3
PMX	99459	100175	99684.4	206.5
CX+PMX	99263	99557	99445.5	123.3

표에서 “CX”는 순환 교차만을 사용하였을 경우 “PMX”는 부분 사상 교차만을 사용하였을 경우, “CX+PMX”는 동시에 순환 교차와 부분 사상 교차를 모두 사용할 때의 결과를 보여준다. 위의 표에서 보는 바와 같이, 순환 교차나 부분 사상 교차 한 가지만을 사용할 때와 비교해서 두개의 교차 방법을 동시에 사용함으로써 더욱 효과적으로 부모 세대의 훌륭한 형질을 자식 세대에 전달할 수 있기 때문에 Hamming code와 Hsiao code 모두에서 기존의 방법보다 좋은 성능을 보여준 것은 물론 더욱 성능이 향상됨을 알 수 있다.

4. 결 론

저전압 환경에서의 메모리 집적도가 증가함에 따라 메모리는 soft error에 매우 민감해졌다. 오류 정정 코드는 일반적으로 양산 이후 메모리의 soft error를 수정하기 위해서 사용된다. 본 논문에서는 새로운 저전력 오류 정정 코드의

설계방법을 제안하였다. 오류 정정 코드의 설계는 비선형 최적화 문제로 포함되는데 우리는 다양한 유전 연산자를 포함하는 유전자 알고리즘을 이용하여 이 문제를 해결하였다. 제안하는 방법을 Hamming code와 Hsiao code에 적용하여 그 효율성과 효과를 볼 수 있었다.

참 고 문 헌

- [1] 이희성, 김은태, "유전자 알고리즘을 이용하여 설계된 ECC회로의 uniform distribution memory data에서의 성능분석," *대한전자공학회 하계종합 학술대회 논문집*, vol. 30, no 1, pp. 667-668, 2007.
- [2] K. Mohanram and N. A. Touba, "Input ordering in concurrent checkers to reduce power consumption," in *Proc. of IEEE Symposium on Defecated Fault Tolerance*, pp. 87-95, 2005.
- [3] D. Rossi, V. Dijk, R. Kleihorst, A. K. Nieuwland, and C. Metra, "Power consumption of fault tolerant codes: the active elements," in *Proc. of Intentional On-Line Testing Symposium*, pp. 61-67, 2003.
- [4] S. Ghosh, S. Basu, and N. Touba, "Reducing power consumption in memory ECC checkers," *International Test Conference*, pp. 1322-1331, 2004.
- [5] H. Lee and E. Kim, "A new genetic design for Error Correcting Code for power minimization," *Journal of Circuits, Systems, and Computers*, vol. 17, no. 5, pp. 773-783, 2008.
- [6] H. Lee, J. Sung, and E. Kim, "Reducing power in error correcting code using genetic algorithm," in *Proc. Int. Conf. Computer Information and Systems Science and Engineering*, pp. 179-182, 2007.
- [7] H. Lee, S. Hong, and E. Kim, "Reducing power consumption in Error-Correcting Code using various genetic operators," in *Proc. Joint Int. Conf. Soft Computing and Intelligent Systems and International Symposium on Advanced Intelligent Systems*, pp. 2033-2036, 2008.

저 자 소 개



이희성(Heesung Lee)
 2003년 : 연세대학교 전기전자공학부 졸업 (공학사)
 2005년 : 연세대학교 전기전자공학부 석사과정 졸업(공학석사)
 2005년~현재 : 동 대학원 전기전자공학과 박사과정

관심분야 : Computational intelligence, 로봇 비전, 패턴 인식
 E-mail : 4u2u@yonsei.ac.kr



홍성준(Sungjun Hong)
 2005년 : 연세대학교 전기전자공학부 졸업 (공학사)
 2005년~현재 : 동 대학원 전기전자공학과 석박사통합과정

관심분야 : 기계 학습, 생체 인식
 E-mail : imjune@yonsei.ac.kr



안성제(Sungje An)
 2007년 : 연세대학교 전기전자공학부 졸업 (공학사)
 2007년~현재 : 동 대학원 전기전자공학과 석사과정

관심분야 : Mobile Robot SLAM, 패턴 인식
 E-mail : haga0801@yonsei.ac.kr



김은태(Euntai Kim)
 1992년 : 연세대학교 전자공학과 졸업 (공학사, 전체수석)
 1994년 : 연세대학교 전자공학과 석사과정 졸업(공학석사)
 1999년 : 연세대학교 전자공학과 박사과정 졸업(공학박사)

1999년 3월~2002년 2월 : 국립한경대학교 제어계측공학과 조교수
 2002년 3월~현재 : 연세대학교 전기전자공학부 부교수
 2003년 : University of Alberta, visiting researcher
 1998년~현재 : IEEE TFS, IEEE SMC, IEEE CAS, FSS 등에서 심의위원 활동 중
 2003년 : 대한 전자공학회 해동상 수상

관심분야 : Computational intelligence, 지능형 로봇
 Phone : +82-2-2123-2863
 E-mail : etkim@yonsei.ac.kr