

논문 2017-54-6-7

Multi-Access Memory System(MAMS)의

속도 향상을 위한 아키텍처 설계

(Architecture design for speeding up Multi-Access Memory System(MAMS))

고 경 식*, 김 재 희*, 이스라엘**, 박 종 원**

(Kyung-sik Ko, Jae Hee Kim, S-Ra-El Lee, and Jong Won Park[©])

요 약

대용량 고화질의 영상 응용분야에서는 많은 양의 데이터를 고속으로 처리하는 기술이 필요하며, 이를 위해 고속화된 병렬처리 시스템이 요구된다. 2004년 park^[2]은 병렬처리 메모리의 충돌 없이 여러 처리기에 데이터를 접속할 수 있는 방법을 제안하였다. 제안된 MAMS(Multi-Access Memory System)는 이후 MAMS-PP16 및 MAMS-PP64 등으로 추가적인 연구가 이루어졌다. MAMS는 병렬처리를 위한 메모리 아키텍처로써 One-chip으로 구성되어야 하기 때문에 기존 MAMS와 동일한 기능을 수행하면서 아키텍처의 최소화 하는 방법의 연구가 필요하다. 주소 계산 (ACR: Address Calculation and Routing) circuit과 MMS(Memory Module Selection)circuit의 아키텍처는 메모리에 있는 데이터를 병렬처리기(Crossing Elements)들에게 전달한다. 본 논문에서는 MMS circuit을 사용하지 않고 ACR circuit 내부에 1개의 쉬프트와 메모리 모듈의 개수만큼의 조건문으로 구성하는 방법을 통해 아키텍처를 최소화 하는 방법을 제안한다. 구현한 아키텍처의 검증을 위해 Image correlation 실험을 하였다. 실험을 통하여 제안된 MAMS-PP64의 처리시간을 측정 하였으며, 그 결과 Ratio가 평균 1.05향상 된 결과를 확인 할 수 있었다.

Abstract

High-capacity, high-definition image applications need to process considerable amounts of data at high speed. Accordingly, users of these applications demand a high-speed parallel execution system. To increase the speed of a parallel execution system, Park (2004) proposed a technique, called MAMS (Multi-Access Memory System), to access data in several execution units without the conflict of parallel processing memories. Since then, many studies on MAMS have been conducted, furthering the technique to MAMS-PP16 and MAMS-PP64, among others. As a memory architecture for parallel processing, MAMS must be constructed in one chip; therefore, a method to achieve the identical functionality as the existing MAMS while minimizing the architecture needs to be studied. This study proposes a method of miniaturizing the MAMS architecture in which the architectures of the ACR (Address Calculation and Routing) circuit and MMS (Memory Module Selection) circuit, which deliver data in memories to parallel execution units (PEs), do not use the MMS circuit, but are constructed as one shift and conditional statements whose number is the same as that of memory modules inside the ACR circuit. To verify the performance of the realized architecture, the study conducted the processing time of the proposed MAMS-PP64 through an image correlation test, the results of which demonstrated that the ratio of the image correlation from the proposed architecture was improved by 1.05 on average.

Keywords : Architecture Minimize, Parallel Processing, Multi-Access Memory System, Image Processing, SIMD Computer

* 학생회원, ** 정회원, 충남대학교 정보통신공학과
(Department of Information Communications Engineering,
ChungNam National University)

© Corresponding Author(E-mail : ksjobmanager@naver.com)

※ 이 연구는 충남대학교 학술연구비에 의해 지원되었음

Received ; February 15, 2017 Revised ; May 15, 2017

Accepted ; May 24, 2017

I. 서 론

최근 대용량 고화질의 영상 응용분야가 확산됨에 따라 컴퓨터 아키텍처의 최소화 연구가 필요해 지고 있다. 다양하고 복합적인 멀티미디어 정보가 쏟아지면서 영상의 크기에 따른 메모리 공간의 한정성 극복이 대두되는

현실이다. 이러한 방대한 영상의 고속 처리를 위해 별도의 기술 및 장비 개발이 시급하며, 이를 극복 할 수 있는 다중접근기억장치와 멀티미디어 처리를 위한 병렬 처리기반의 아키텍처를 소개한다. 영상처리와 같이 반복적인 처리를 하는 분야에서는 SIMD 구조의 시스템이 적합하다. 영상에서 임의의 간격을 가진 여러 형태로 영상자료를 동시에 접근할 수 있는 다중접근기억장치인 MAMS (Multi-Access Memory System)^[1-2]는 $p \times q$ 개의 PEs (Processing Elements)를 가진 병렬처리기로 구현된다. 우리는 16개의 PE를 포함하는 MAMS-PP16^[3]을 2012년부터 연구하였으며, 현재 64개의 PE를 포함하는 MAMS-PP64^[4]는 2014년에 개발 및 구현되어 범용 컴퓨터를 대상으로 속도 향상을 확인하였다.

본 논문에서는 병렬처리기(PEs)들에게 메모리에 있는 데이터를 전달하기 위한 주소 계산 (ACR: Address Calculation and Routing)circuit과 MMS(Memory Module Selection)circuit의 아키텍처를 MMS circuit을 사용하지 않고 ACR circuit 내부에 1개의 쉬프트와 메모리 모듈의 개수만큼의 조건문으로 구성하여 아키텍처의 최소화하는 방법을 제안한다. 구현한 아키텍처의 검증을 위해 Image correlation 실험을 하였다. 제안한 MAMS-PP64의 성능을 평가한 결과 Ratio가 평균 1.05향상 된 결과를 확인 할 수 있었다.

II. 본 론

1. MAMS - PPS

많은 양의 Image Data를 처리하기 위해서는 제어 유닛과 많은 처리기를 가지는 SIMD 컴퓨터가 효율적이다.

SIMD 컴퓨터의 제어유닛은 Image 내부 데이터 요소들을 연속적으로 동작 명령어를 생성 시키며, 처리기들(PEs)을 효율적으로 활용하기 위해서는 p 와 q 를 사용하여 효율적인 메모리시스템을 구현해야 한다. 효율적인 메모리 시스템을 구현하기 위해서는 5개의 요소를 가져야 되며, 이 다섯가지 요소는 M, N, p, q, s 로 나타낼 수 있는데 다음과 같이 설명할 수 있다.

M 과 N 은 이미지의 가로 세로크기를 나타내며, p, q 는 MAMS에 대한 설계 파라미터로 p, q, s 가 결정 되면 p, q 는 8개의 서브어레이를 구할 수 있고, s 는 주소 할당함수를 구할 수 있다.

디지털 영상은 영상 점 $I(i, *)$ 들의 $M \times N$ 행렬로 나타낼 수 있는데, 이 때 각 화소 $I(i, j)$ 는 해당 부분의 색과 밝기를 나타내며 다음 식 (1)과 같이 표기 된다.

$$I(i, j) : 2\text{-차원 } M \times N\text{의 데이터 배열 좌표} \quad (1)$$

$$(0 \leq i < M, 0 \leq j < N)$$

임의의 정수 p, q 를 설계 상수로 하였을 때 $p \times q$ 개의 화소들의 빠른 영상처리를 위해서는^[1-2], $M \times N$ 크기 배열($I(i, *)$)의 데이터 안에서 일정한 간격(r)을 가지고 그림 1과 같이 구성 된다^[2].

여기서 s 는 다음 조건을 만족하는 임의의 정수로써 식 (2)와 같이 정의 되며, Van Voorhis와 Morrin^[5]에 의해서도 고려되었다.

$$s \geq \lceil N/q \rceil \quad (2)$$

($\lceil x \rceil$ 는 x 보다 크거나 같은 정수 중에서 가장 작은 수를 나타낸다.)

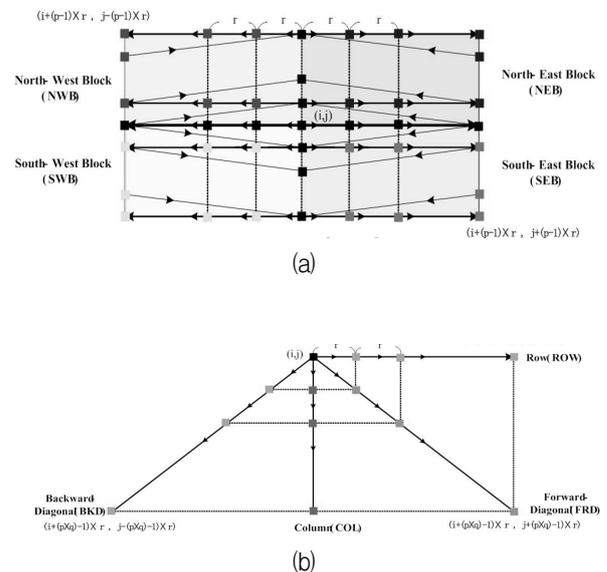


그림 1. (a) 4방향(SEB, SWB, NWB, NEB)의 블록 서브어레이
(b) 4방향(ROW, COL, FRD, BKD)의 라인 서브어레이

Fig. 1. (a) 4-directional block (SEB, SWB, NWB, NEB) subarrays.
(b) 4-directional line (ROW, COL, FRD, BKD) subarrays.

$p \times q$ 개의 영상을 블록, 행, 열의 형태로 동시에 접근할 수 있는 다중접근기억장치를 구성한다. 이때 주소 생성회로는 메모리 모듈을 배치하는 모듈 할당 함수와 배치된 모듈 내의 주소를 결정하는 주소 할당 함수를 이용한다. MAMS는 5 개의 요소를 충족시키면서 메모리 접근 시간이 일정한 메모리시스템이다. 그림 2는 2004년 ASIC 으로 구현 하였던 MAMS-PP4의 블록도이며, 여기서 $p=q=2, M \times N=128 \times 128, s=64, m=5$ 이다.

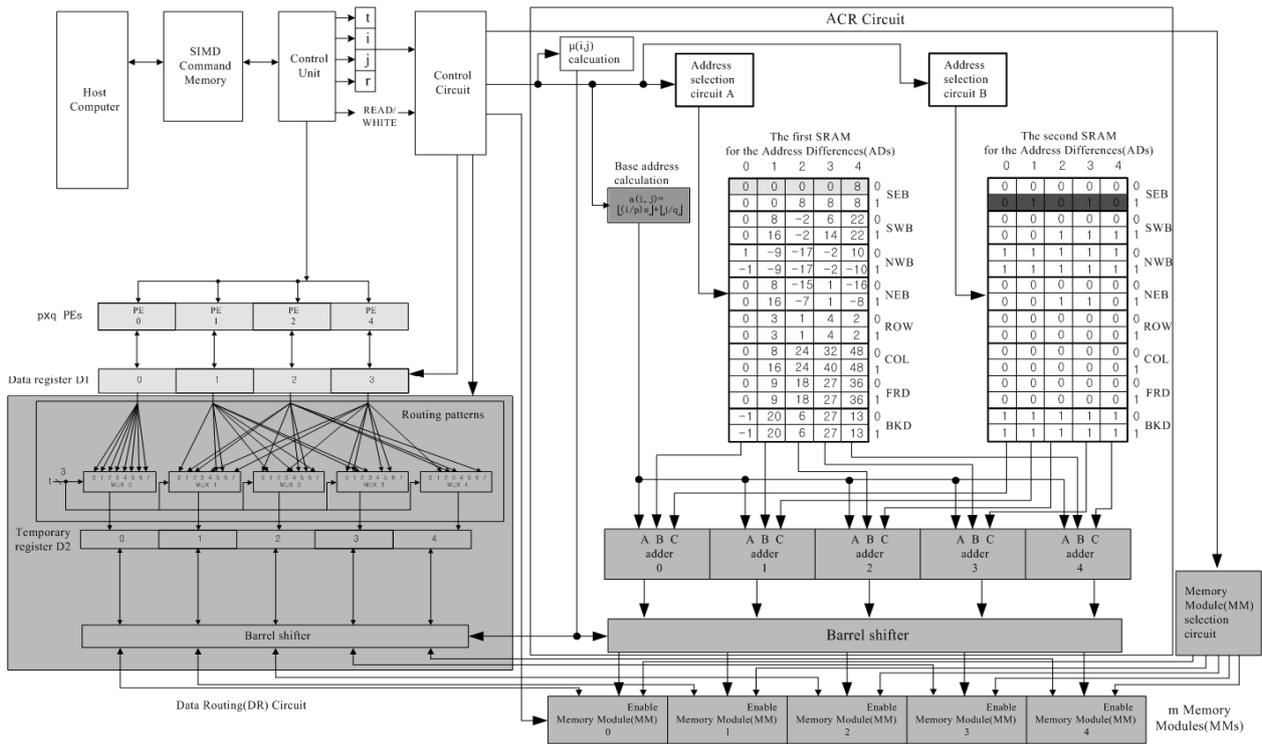


그림 2. 병렬처리 블록 다이어그램
Fig. 2. Function block of parallel processing system.

MAMS - PP4는 Address Calculation and Routing circuit
과 Memory Module Sselection circuit 으로 구성 된다.

가. Address Calculation and Routing(ACR)

(1) Memory Module Assignment Circuit

$M \times N$ 영상에 대해서 블록($p \times q$), 행($1 \times (p \times q)$), 열($(p \times q) \times 1$)에 대해 동시에 접근할 경우 블록, 행, 열에 포함된 각 원소들은 서로 다른 메모리 모듈에 저장 되어야 하고, 메모리 모듈의 수를 m 이라하면, 영상의 각 픽셀 $I(i, j)$ 이 위치할 메모리 모듈을 결정하는 모듈할당함수(Module Assignment Function: μ)를 구할 수 있고 식 (3)^[2]과 같이 표현된다.

$$\mu(i, j) = ((i \times q) + j) \% m \quad (m : MM의 개수) \quad (3)$$

함수 μ 는 임의의 기준 위치 (i, j) 에 의하여 결정되며, j 좌표의 증가에 관련된 설계 파라미터 q, p 와 MAMS의 Memory Module의 개수 m ($m > p \times q$, m 은 소수)를 이용하여 구한다.

(2) Address Assignment Circuit

동시에 접근되는 데이터의 집합 s 의 원소들에 대응하는 메모리 모듈은 함수 μ 에 의해서 결정되지만, 실제로

각 메모리 모듈에 Read/Write하기 위해서는 데이터가 위치하는 메모리 모듈 내의 주소도 동시에 같이 정해져야 한다. 그러므로 s 의 각 원소에 대하여, 함수 μ 가 가리키는 메모리 모듈 내의 주소를 결정하는 함수 α 가 필요하며, 이를 주소할당함수(Address Assignment Function : α) 라 하고 식(4)와 같이 정의한다.^[1, 6]

$$\alpha(i, j) = \lfloor i/p \rfloor \times s + \lfloor j/q \rfloor \quad (4)$$

($\lfloor x \rfloor$ 는 x 보다 작거나 같은 정수 중에서 가장 큰 수를 나타낸다.)

함수 α 를 살펴보면, 집합 s 의 데이터들에 대한 서로 다른 메모리 모듈 내의 주소들은 함수 μ 와 상관없이 임의의 기준 위치 (i, j) 에 의하여 결정되는 것을 알 수 있다. 즉, 함수 α 와 함수 μ 는 서로 독립적이며, 두 함수의 조합으로 접근되는 데이터의 위치 지정을 위한 메모리 모듈과 주소를 유일하게 지정할 수 있다.

(3) Address selection Circuit

Address selection circuit은 2개의 SRAM을 위해 존재한다. Remaining SRAM의 주소를 계산하기 위해서는 Remaining Address selection circuit을 사용하고, Binary SRAM의 주소를 계산하기 위해서는 Binary Address selection circuit을 사용한다.

1) Remaining Address Selection Circuit(RASC)

Remaining Address selection circuit은 다음과 같은 수식으로 주소를 구할 수 있다.

$$RASC(i) = i \% p \quad (5)$$

MAMS - PP4 일때는 $p=2$ 이므로 주소는 0, 1의 값을 가지고, MAMS - PP64 일때는 $p=8$ 이므로 주소는 0~7 값을 가진다.

2) Binary Address Selection Circuit(BASC)

Binary Address selection circuit은 다음과 같은 수식으로 주소를 구할 수 있다.

$$BASC(j) = j \% q \quad (6)$$

MAMS - PP4 일때는 $q=2$ 이므로 주소는 0, 1의 값을 가지고, MAMS-PP64 일때는 $q=8$ 이므로 주소는 0~7 값을 가진다.

(4) SRAM for the Address Differences(ADs)

MAMS는 라우팅 주소에서 어드레스 계산을 분리하여 느린 나눗셈 문제를 해결하였지만, 어드레스 연산 회로의 복잡도는 여전히 문제점으로 남았다. 그래서 Park^[2]은 8개의 Subarray type ((South-West Block (SWB), South-East Block (SEB), North-West Block (NWB), North-East-Block (NEW), Forward-diagonal (FRD), Column (COL), Row (ROW) and Backward-diagonal (BKD))을 SRAM에 address Differences(ADs)값을 저장하여 회로의 복잡도를 해결 하였다. 하드웨어 시스템의 비용절감을 위해 MAMS-PP4^[7]는 영상처리에서 일반적으로 많이 사용되는 3개의 Subarray type(SEB, ROW, COL)을 SRAM에 저장 하였다.

1) Remaining SRAM

Remaining SRAM에 저장 되어 있는 Address Differences (ADs) 값은 메모리 모듈의 Index number를 계산하여 차이 값을 수동으로 저장하며, 3개의 Subarray type 수식은 다음과 같고, μ_2 는 각각의 메모리 모듈의 번호에 대하여 값이 구해지고 설정된 interval r 에 대한 각 Subarray type에 따라 r' 을 구할 수 있다.^[2]

$$SEB: RADs(i,r) = s((i\%p + (((\mu_2 \times r')\%m)/q)r)p) + (((\mu_2 \times r')\%m)\%r)/q, \quad (7) \\ 0 \leq \mu_2 \leq m, (r \times r')\%m = 1$$

$$ROW: RADs(i,r) = ((\mu_2 \times r')\%m)r/q, \quad (8) \\ 0 \leq \mu_2 \leq m, (r \times r')\%m = 1$$

$$COL: RADs(i,r) = s \times ((i\%p + ((\mu_2 \times (q \times r')\%m)r)/p), \quad (9) \\ 0 \leq \mu_2 \leq m, ((q \times r) \times (q \times r'))\%m = 1$$

2) Binary SRAM

Binary SRAM에 저장 되어 있는 Address Differences (ADs) 값은 각각의 메모리 모듈의 주소를 계산하여 차이 값을 수동으로 저장하며, 3개의 Subarray type 수식은 다음과 같다.^[2]

$$SEB: BADs(j,r) = (j\%q + (((\mu_2 \times r')\%q)r)/q) - (((((\mu_2 \times r')\%m)\%q)r)/q), \quad (10) \\ 0 \leq \mu_2 \leq m, (r \times r') = 1$$

$$ROW: BADs(j,r) = (j\%q + (((\mu_2 \times r')\%m)r)/q) - (((((\mu_2 \times r')\%m)\%q)r)/q), \quad (11) \\ 0 \leq \mu_2 \leq m, (r \times r')\%m = 1$$

$$COL: BADs(j,r) = 0 \quad (12)$$

(5) Adders and Barrel shifter

Adders은 기준 좌표에서 메모리 모듈의 Index number 차이 값과 메모리 모듈의 주소 차이 값을 더하여 메모리 모듈의 주소를 구한다.

$$Adder_T: D_T = A_T + B_T + C_T, \quad 0 \leq T < m \quad (13)$$

A_T : 식 (4)에 의해 구해진 기준 좌표 값($\alpha(i,j)$)을 0부터 $m-1$ 까지 동일한 값을 입력

$$(A_0, A_1, \dots, A_{m-2}, A_{m-1})$$

B_T : 식(7-9)에 의해 구해진 Index number 차이 값

$$(B_0, B_1, \dots, B_{m-2}, B_{m-1})$$

C_T : 식(10-12)에 의해 구해진 메모리 모듈의 주소 차이값 ($C_0, C_1, \dots, C_{m-2}, C_{m-1}$)

D_T : 가산기에 의해 구해진 결과 값

$$(D_0, D_1, \dots, D_{m-2}, D_{m-1})$$

메모리 모듈의 주소 값은 Subarray type에서 정의된 PEs의 순서로 구해진다. PEs의 순서와 메모리 모듈의 인덱스 번호를 맞추기 위해서는 식 (3)에서 계산된 $\mu(i,j)$ 배 만큼 우측으로 회전 한다.

나. Memory Module(MM) Selection

Park에 의해 제안 되었던 Memory Module Selection circuit은 메모리 모듈의 사용 여부를 제어하며, 그림 2의 ACR circuit 외부에 별도의 circuit으로 구성되며, SEB와 ROW는 식 (14)와 같이 동일한 수식을 사용하고, COL수식은 식 (15)와 같다.^[2]

$$MMs(i,j) = ((i \times q + j)\%m + (k \times r)\%m)\%m, \quad (14) \\ 0 \leq k \leq q \times p$$

$$MMs(i, j) = ((i \times q + j) \% m + (q \times k \times r) \% m) \% m, \quad (15)$$

$$0 \leq k < q \times p$$

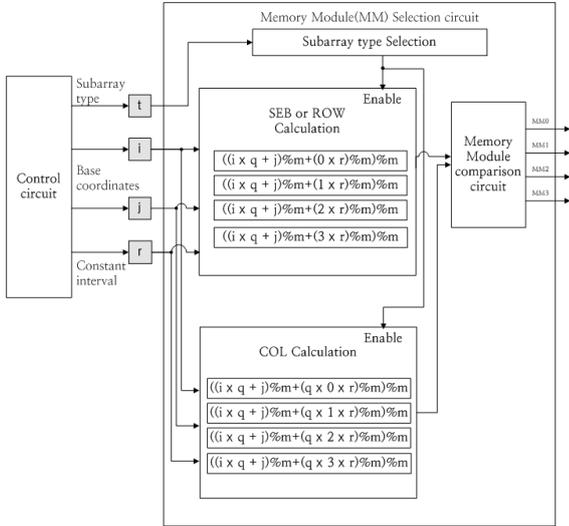


그림 3. MAMS-PP4의 MM selection circuit구성도
Fig. 3. Block diagram of MM selection circuit by MAMS-PP4.

식 (14)~(15)에서 발생하는 값은 사용 가능한 메모리 모듈의 값이며 이외의 값은 사용 되지 않는 모듈이다. 그림 3은 MAMS - PP4의 MM Selection circuit의 구성도이다. Control circuit을 통해 Subarray type, Base coordinates 그리고 Constant interval를 입력 받고, Subarray type Selection은 Subarray type에 맞는 Calculation을 선택한다. 선택된 Calculation은 식 (14)~(15)를 이용하여 메모리 모듈의 번호를 계산 한다.

이때 처리시간이 최소한 ACR circuit 처리시간과 같아야 된다. 그래서 Calculation은 k가 가질 수 있는 경우의 수만큼 함수로 구성해야 된다. MAMS-PP4에서는 $0 \leq k < 4$ 이므로 SEB or ROW Calculation 과 COL Calculation 들은 각각 4개의 함수로 구성 되고, 총 8개 함수로 구성 된다.

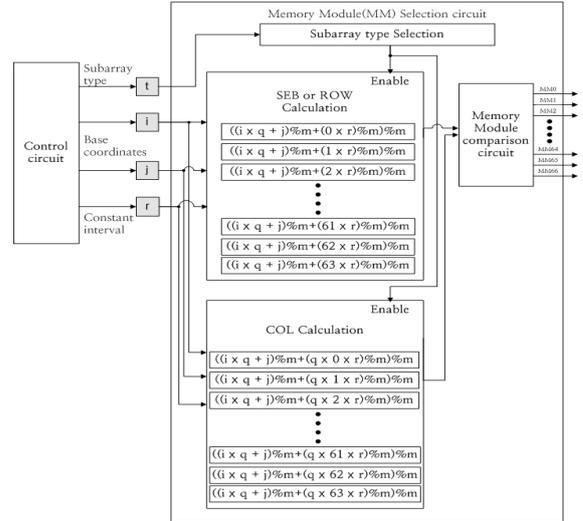


그림 4. MAMS-PP64의 MM selection circuit구성도
Fig. 4. Block diagram of MM selection circuit by MAMS-PP64.

Memory Module comparison은 Calculation으로부터 사용 가능한 메모리 모듈의 번호들을 비교하여 각각의 Memory Module에게 사용 여부를 전달한다.

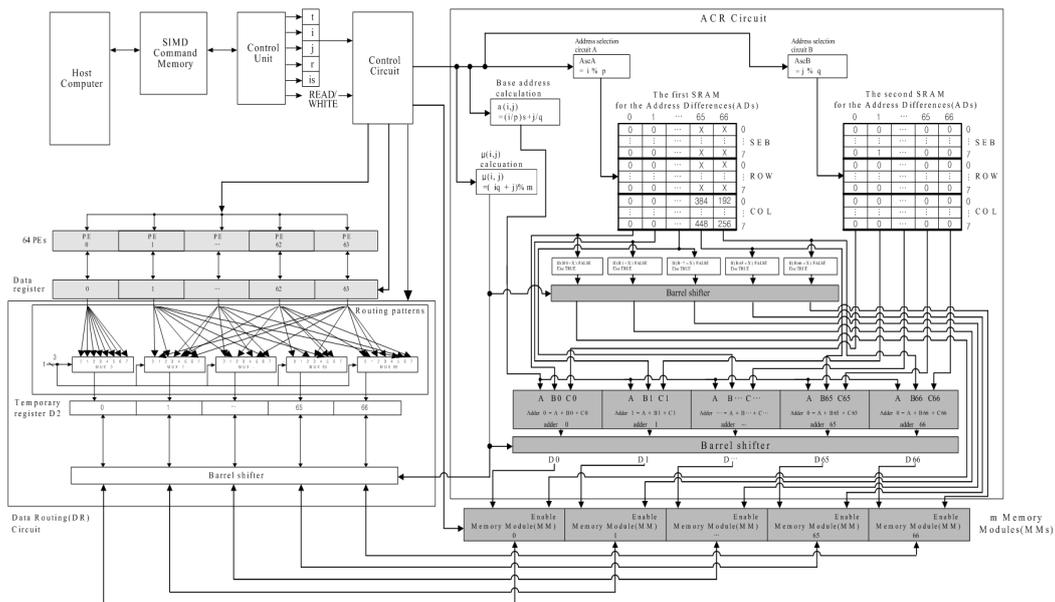


그림 5. 제안한 MAMS - PP64 구성도
Fig. 5. Block diagram of MAMS-PP64.

Memory Module(MM) Selection circuit은 Memory Module의 개수가 작을 때는 큰 문제점이 발생하지 않는다. 하지만 MAMS-PP64는 그림 4와 같이 Memory Module의 개수가 많아지면 Calculation과 Memory Module comparison의 함수는 공간복잡도가 높아진다. 그리고 Memory Module(MM) Selection circuit은 ACR circuit 보다 시간복잡도가 높기 때문에 모듈의 동기화가 발생하여 처리 시간이 늦어진다.

우리는 Remaining Address Differences(ADs) 와 ACR circuit 를 새로이 설계하여 두 가지 문제점을 해결 하였고, Proposed of MAMS-PP64에서 설명한다.

2. Proposed of MAMS - PP64

우리의 목표는 메모리 시스템인 MAMS^[2]를 사용하여 시스템 성능이 향상 되는 MAMS-PP64를 구현하는 것이다. MAMS-PP64의 구조는 MAMS^[2]를 기초로 설계 되었으므로 Memory Module Assignment Circuit, Address Assignment Circuit 그리고 Address selection Circuit들의 구조는 동일하다. 그러나 앞에서 발생한 두 가지 문제점 때문에 MAMS - PP64 는 Memory Module Selection circuit을 제거 하고 Remaining Address Differences (ADs) 와 Adders and Barrel shifter를 새로이 설계하였다.

MAMS - PP64 는 Address Calculation and Routing circuit만 존재하며, 그림 5와 같은 구조로 구성된다.

제안한 MAMS-PP64는 MM Selection circuit을 대신하여 67개 논리케이트와 1개의 Barrel shifter를 추가하였고, MAMS-PPs^[2]의 MM Selection circuit 과 동일하게 메모리 모듈의 Enable/Disable을 동작하도록 구성하였다. (p=q=8, M x N=512 x 512, s=64, m=67).

가. Proposed of Address Calculation and Routing

(1) Remaining SRAM for MAMS-PP64

MAMS-PP64는 MAMS-PP4^[7]와 같이 회로의 복잡도와 하드웨어 시스템의 비용절감을 위해 영상처리에서 많이 사용되는 3개의 Subarray type(SEB, ROW, COL)을 SRAM 에 저장 한다.

Remaining Address Differences(ADs) 값은 메모리 모듈의 Index number의 차이값 이다. 그래서 우리는 식(7-9)를 이용하여 차이값을 수동으로 저장하고, 사용되지 않는 메모리 모듈의 Index number를 구하여 Remaining SRAM 에 저장한다.

MAMS의 Memory Module의 개수(m)는 p×q보다 큰 소수를 가지므로 사용 되지 않는 메모리 모듈의 개수($\mu 3$)를 식 (16)과 같이 구한다.

$$\mu 3 = m - p \times q \quad (16)$$

사용되지 않는 메모리 모듈의 개수($\mu 3$)를 이용하여 사용되지 않는 메모리 모듈의 Index number를 구하기 위하여 SEB와 ROW는 식 (17)을 사용하고 COL수식은 식 (18)을 사용하여 그 위치에 'x'를 표시한다.

$$NRADs = m - (\mu 3 \times r) + (t \times r), 0 \leq t < \mu 3 \quad (17)$$

$$NRADs = m - ((q \times (t + 1) \times r) \% m), 0 \leq t < \mu 3 \quad (18)$$

(2) MAMS-PP64 의 Adders 와 Barrel shifter

MAMS - PP64를 위한 Adders는 식 (13)을 이용하지만, 식 (18-20)에서 구해진 BT에 'X'가 입력되면 더하지 않고 'X'를 출력하도록 식 (19)와 같으며, D_T 는 주소 값 과 'X'를 가진다.

$$Adder_T: D_T = \begin{cases} \text{if}(B_T \neq 'X') A_T + B_T + C_T, \\ \text{else}' X', 0 \leq T < m \end{cases} \quad (19)$$

1장 (5) 의 Barrel shifter 와 같이 $\mu(i, j)$ 배 만큼 우측으로 회전 하면, PEs 와 메모리 모듈의 Index number가 맞추어지고 'X' 값을 가지는 D_T 는 disable이 된다. 우리가 제안한 방법은 Memory module(MM) selection circuit을 사용하지 않는다.

나. 제안한 Address Calculation and Routing 검증

우리는 MAMS-PPs^[2]에서 사용한 그림 6을 이용하여 제안한 ACR circuit을 검증 한다. 그림 6 a)~b)는 MAMS-PPs^[2]에서 사용한 메모리 모듈 인덱스 번호 와 주소를 보여 주고 있으며, c)는 Remaining SRAM 과 Binary SRAM에 입력되는 주소의 차이 중 3개의 subarray type(SEB, ROW, COL)을 보여 준다.

그림 6에서 SEB(6,7,1), ROW(2,3,3), COL(5,12,2)을 기존방법과 제안한 방법을 검증한다.

SEB(6,7,1)는 (6,7)에 있는 SEB subarray type 이며, 1의 간격으로 I(6,7), I(6,8), I(7,7), I(7,8) 순서대로 구성 되어야 된다. MAMS-PPs^[2]에서 제안한 ACR방법을 이용한 절차는 다음과 같다:

SEB(6,7,1)에 RASC(i) = 0, BASC(j) = 1, r = 1, 가산기 입력 AT의 값은 $\alpha(i, j) = 27$ 이므로 (27,27,27,27)이다. 그림 6 c)에서 얻은 Ads에서 가산기 입력 BT의 값은 (0,0,0,8) 이고, 가산기입력 CT의 값은 (0,1,0,1,0)

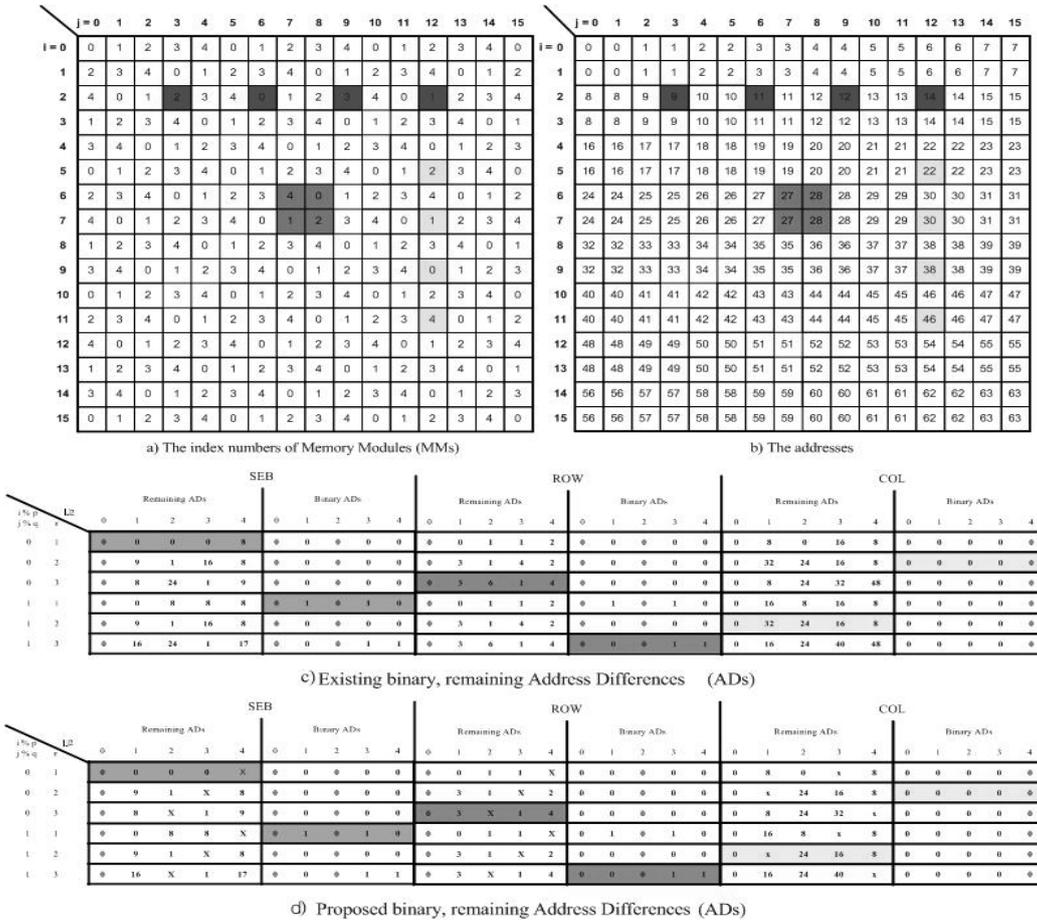


그림 6. 메모리 모듈 인덱스 번호, 메모리 모듈 주소, 기존 방식 바이너리와 나머지 주소의 차이, 제안한 바이너리와 나머지 주소의 차이

Fig. 6. The index numbers of Memory Modules(MMs), The addresses, Existing binary, remainig Address Differences (ADs), Propossed binary, remainig Address Differences(ADs).

이다. 이를 식 (13)에 의해 DT를 아래와 같이 구할 수 있다.

$$Adder_T : D_T = A(27,27,27,27,27) + B(0,0,0,0,8) + C(0,1,0,1,0) = (27,28,27,28,35)$$

Barrel shifter를 사용하여 $\mu(i,j)(= 4)$ 배로 우측으로 회전하면 (28, 27, 28, 35, 27)이고, 그림 2의 MMSc를 이용하여 사용할 수 있는 메모리 모듈을 구하면 (○, ○, ○, X, ○) 이와 같고(메모리 모듈 사용 여부 Enable:○/ Disable:X), 이를 이용하여 결과는 (28, 27, 28, X, 27) 이다.

본 논문에서 제안한 방법을 위해 그림 6 c) Remaining ADs 대신해 제안한 방법으로 새로이 계산된 값을 SRAM에 저장하기 위하여 사용되지 않는 Memory Module의 개수는 식 (16)에 의해 $\mu_3(=1)$ 이고 사용되지 않는 메모리 모듈의 index number는 $m=5, r=1, t=0$ 이므로 식 (18)에 의해 4가 된다. 여기서 구해진 메모리모듈의 index

number는 그림 6 d)와 같이 'X'를 표시하고 Remaining SRAM에 저장한다.

우리가 제안한 ACR방법을 이용한 절차는 다음과 같다:

(step 1) SEB(6,7,1)이 입력되면, RASC(i)=0, BASC(j)=1, $r=1, r'=1$ 을 구한다.

(step 2) 식 (4)에 의해 $\alpha(i,j) = 27$ 를 구하고, AT의 행렬 값 (27,27,27,27,27)을 구한다.

(step 3) 논문에서 제안한 방법으로 구한 그림 6 d) ADs에 의해 가산기 입력 BT의 값은 (0, 0, 0, 0, X)이고, CT의 값은 (0, 1, 0, 1, 0) 이다.

(step 4) 가산기 식 (13)에 의해 결과는 (27, 28, 27, 28, X)를 구한다.

(step 5) Barrel shifter를 사용하여 $\mu(i,j)(= 4)$ 배로 우측으로 회전하면 (28, 27, 28, X, 27) 이다.

제안한 방법과 기존 방법의 결과가 같음을 확인 할 수 있었고, 이를 통해 공간복잡도가 낮아짐을 확인 할 수 있다.

표 1은 SEB(6,7,1), ROW(2,3,3), COL(5,12,2)을 기존방법과 제안한 방법의 결과를 보여 준다.

표 1. 기존방법과 제안한 방법 결과
Table1. Result of existing and proposed method.

Classification	SEB(6,7,1)	ROW(2,3,3)	COL(5,12,2)	
RASC(i)	0	0	1	
BASC(j)	1	1	0	
r	1	3	2	
r'	1	2	3	
$\mu(i,j)$	4	2	2	
A_T	(27,27,27,27,27)	(9, 9, 9, 9, 9)	(22,22,22,22,22)	
Existing method	B_T	(0, 0, 0, 0, 8)	(0, 3, 6, 1, 4)	(0,32,24,16, 8)
	C_T	(0, 1, 0, 1, 0)	(0, 0, 0, 1, 1)	(0, 0, 0, 0, 0)
	D_T	(27,28,27,28,35)	(9,12,15,11,14)	(22,54,46,38,30)
	Barrel Shifter	(28,27,28,35,27)	(11,14,9,12,15)	(38,30,22,54,46)
	MMSc	(○,○,○,○,○,○)	(○,○,○,○,○,○)	(○,○,○,○,○,○)
	Result	(28,27,28, X,27)	(11,14,9,12, X)	(38,30,22, X,46)
The proposed method	B_T	(0, 0, 0, 0, X)	(0, 3, X, 1, 4)	(0, X,24,16, 8)
	C_T	(0, 1, 0, 1, 0)	(0, 0, 0, 1, 1)	(0, 0, 0, 0, 0)
	D_T	(27,28,27,28, X)	(9,12, X,11,14)	(22, X,46,38,30)
	Barrel Shifter	(28,27,28, X,27)	(11,14,9,12, X)	(38,30,22, X,46)
	Result	(28,27,28, X,27)	(11,14,9,12, X)	(38,30,22, X,46)

III. 실험

본 장에서는 Memory Module Selection circuit을 제거 하고 새로이 제안한 Address Calculation and Routing circuit이 기존 구조와 비교하기 위하여 그림 7과 같이

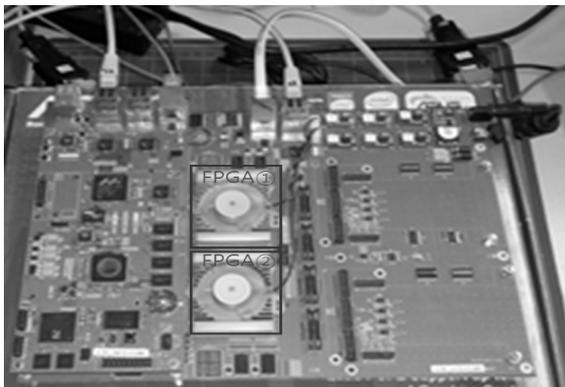


그림 7. FPGA 보드 구성
Fig. 7. Configuration of board by FPGA.

2개의 FPGA(Xilinx Virtex6@100MHz) 보드를 구성하였고, Verilog를 사용하여 FPGA①에는 기존 구조를 FPGA②에는 제안한 구조를 구성하였다.

표 2는 Serial Processor, 기존방식의 MAMS-PP64와 제안한 MAMS-PP64를 처리 기반을 보여주고, Serial Processor는 Intel Core2@ 2.66 GHz이고 기존 방식과 제안한 MAMS-PP64의 클럭 속도는 100 MHz이다.

표 2. Correlation 영상처리를 위한 알고리즘 처리기반
Table2. The processing base of correlation image algorithm.

Classification	Serial Processor	Existing MAMS-PP64	Proposed MAMS-PP64
Clock speed	Intel Core 2@ 2.66 GHz	FPGA VERTEX6 @ 100 MHz	FPGA VERTEX6 @ 100 MHz
Operating system	Window 7	Linux	Linux
Program	Visual Studio 2010	C++	C++

표 3은 기존 방식의 FPGA 합성 결과와 제안된 방식의 FPGA 결과를 보여 주고 있고, 제안된 방식의 Slice LUT의 사용 개수가 103898개로 기존 방식으로 구현한 모듈의 119073개 보다 약 13% 감소한 것을 확인 할 수 있었고, Slice Register와 LUT-FF는 약 1% 내외의 감소하였다. 이러한 결과는 기존의 방법보다 제안된 방법의 공간복잡도가 낮아짐을 보여 준다.

표 3. FPGA 합성 결과
Table3. Synthesis result of FPGA.

Logic Utilization	Existing MAMS-PP64			Proposed MAMS-PP64		
	Used	Available	Utilization	Used	Available	Utilization
Number of Slice Registers	262263	301440	87%	262254	301440	87%
Number of Slice LUTs	119073	150720	79%	103898	150720	77%
Number of fully used LUT-FF pairs	1132	4180	27%	1126	4180	26%
Number of bonded IOBs	323	600	53%	323	600	53%
Number of BUFG/BUFGC TRLs	1	32	3%	1	32	3%

우리는 시간복잡도를 알아보기 위하여, 이전연구^[4]에서 사용한 Image correlation을 이용한 방식으로 실험하여, 그림 7과 같이 Serial Processor와 MAMS-PP64를 하나의 프로그램으로 구성 하였다.

실험 방법은 마스크 64×64, 128×128, 256×256의 마스크를 이용하여 각각 10 번 실행하였다. 그 결과를 표 4~6에 보여준다. Serial Processor는 64×64 에서는 21~23(초), 128×128 에서는 73~76(초), 256×256 에서는 175~177(초)로 처리시간에 변화가 있다는 것을

확인 하였다. 그리고 기존 MAMS-PP64 처리 시간은 64×64 에서는 6(초), 128×128 에서는 19(초), 256×256 에서는 44(초)이며, 제안한 방식에서는 64×64 에서는 5.7(초), 128×128 에서는 17.9(초), 256×256 에서는 41.6(초)로 일정한 속도로 빠르게 처리됨을 확인 할 수 있었다.

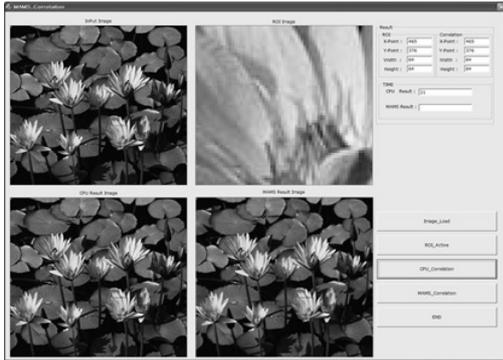


그림 8. Image Correlation 프로그램
Fig. 8. Image correlation application program.

표 4. 64×64 ROI를 사용하여 10 개의 테스트 결과
Table4. Results of ten tests using 64×64 Region Of Interest (ROI).

Mask Size	Order of Execution	Response Time						
		Existing MAMS-PP64(s) ①	Proposed MAMS-PP64(s) ②	Serial Processor(s)		Ratio of Serial Processor to MAMS-PP64		
				Value	Average ③	Existing PP64 (= ③/①)	Proposed PP64 (= ③/②)	Ratio of Existing PP64 to Proposed PP64 (= ①/②)
64×64	1	6	5.7	22	22.20	3.70	3.89	1.05
	2	6	5.7	23				
	3	6	5.7	21				
	4	6	5.7	22				
	5	6	5.7	22				
	6	6	5.7	23				
	7	6	5.7	23				
	8	6	5.7	23				
	9	6	5.7	21				
	10	6	5.7	22				

표 5. 128×128 ROI를 사용하여 10 개의 테스트 결과
Table5. Results of ten tests using 128×128 ROI.

Mask Size	Order of Execution	Response Time						
		Existing MAMS-PP64(s) ①	Proposed MAMS-PP64(s) ②	Serial Processor(s)		Ratio of Serial Processor to MAMS-PP64		
				Value	Average ③	Existing PP64 (= ③/①)	Proposed PP64 (= ③/②)	Ratio of Existing PP64 to Proposed PP64 (= ①/②)
128×128	1	19	17.9	73	74.80	3.94	4.17	1.06
	2	19	17.9	75				
	3	19	17.9	74				
	4	19	17.9	75				
	5	19	17.9	76				
	6	19	17.9	74				
	7	19	17.9	76				
	8	19	17.9	75				
	9	19	17.9	75				
	10	19	17.9	75				

표 6. 256×256 ROI를 사용하여 10 개의 테스트 결과
Table6. Results of ten tests using 256×256 ROI.

Mask Size	Order of Execution	Response Time						
		Existing MAMS-PP64(s) ①	Proposed MAMS-PP64(s) ②	Serial Processor(s)		Ratio of Serial Processor to MAMS-PP64		
				Value	Average ③	Existing PP64 (= ③/①)	Proposed PP64 (= ③/②)	Ratio of Existing PP64 to Proposed PP64 (= ①/②)
256×256	1	44	41.6	175	175.5	3.99	4.21	1.05
	2	44	41.6	176				
	3	44	41.6	177				
	4	44	41.6	175				
	5	44	41.6	175				
	6	44	41.6	176				
	7	44	41.6	176				
	8	44	41.6	174				
	9	44	41.6	175				
	10	44	41.6	176				

Serial Processor와 MAMS-PP64의 처리시간 비교는 Serial Processor의 일정하지 않는 처리시간 때문에 평균을 구하여 각각을 비교 하였고, 기존 MAMS-PP64와 제안한 MAMS-PP64의 처리속도를 비교 하였다. 처리 속도 비교방법은 Ratio를 이용하였으며, 제안한 방식의 처리속도 대 비교대상방식의 속도를 비교 하였다.

64×64 마스크의 경우 Ratio를 비교하면 제안한 방식과 기존방식은 처리 속도는 1.05배 향상됨을 확인할 수 있고, 128×128 마스크의 Ratio는 1.06배, 256×256 마스크의 Ratio는 각각 1.05배 향상된 결과 보여 주고 있다.

IV. 결 론

우리는 영상처리 고속화에서 병렬처리 메모리의 충돌 없이 여러 처리기에 데이터를 접속할 수 있는 구조를 가지는 MAMS를 One-chip으로 구성하기 위하여 공간 및 시간복잡도를 낮은 아키텍처를 연구하였다.

본 논문에서는 기존 MAMS에서 MMs circuit이 Memory Module의 개수가 증가에 따라 공간복잡도가 증가하고 ACR circuit보다 처리시간이 길다는 단점을 보완하기 위하여 Remaining Address Differences(ADs) 와 Adders and Barrel shifter를 새로이 제안하였다.

이와 같이 제안한 MAMS-PP64는 FPGA 합성결과를 통해 제안된 방식에서 Slice LUT가 약 13% 감소, Slice Register와 LUT-FF는 약 1% 감소되어 공간복잡도가 낮아짐을 확인 하였다.

Image correlation실험을 통해 제안한 MAMS-PP64는 기존 MAMS-PP64 Ratio가 평균 1.05향상 된 결과 확인됨으로써 시간복잡도를 낮출 수 있다.

본 논문에서 사용된 영상처리시스템 보드가 실험용 FPGA이기 때문에 100Mhz Clock을 사용하였으나, 향

후 영상처리시스템 보드를 실험에서 비교하였던 Intel Core 2@ 2.66 GHz와 같은 하드웨어 성능일 경우 제안한 MAMS-PP64는 약 28배의 성능향상을 기대할 수 있고, 향후 연구에서는 3D 그래픽스의 셰이더(Shader)를 위하여 MAMS를 이용한 시스템을 구성하여 기존 셰이더 처리기들 보다 향상된 성능의 시스템을 기대할 수 있을 것이다.

REFERENCES

- [1] J.W. Park, "An Efficient Memory system for Image Processing," IEEE Trans. Computers, Vol. C-35, No. 7, pp. 669-674, Jul. 1986.
- [2] J.W. Park, "Multiaccess Memory System for Attached SIMD Computer," IEEE Trans. on Computers, Vol. 53, No. 3, pp. 1439-452, Apr. 2004.
- [3] Y-J Lee, J.H. Kim, and J.W. Park, "Performance Analysis of Implementation on Image Processing Algorithm for Multi-Access Memory System Including 16 Processing Elements," Journal of the Institute of Electronics Engineers of Korea, Volume 49, Number 3, pp. 8-14, 2012.
- [4] J.H. Kim, K.S. Ko, C.S. Oh, and J.W. Park, "64 Processing Elements with Multiaccess Memory System to Speedup Image Correlation," Advanced Science Letters, Volume 22, Number 9, pp. 2376-2380(5), September 2016.
- [5] D. C. Van Voorhis and T. H. Morrin, "Memory systems for image processing," IEEE Trans. Comput., vol. C-27, pp. 113-125, Feb. 1978.
- [6] P. Budnik and D. J. Kuck, "The Organization and Use of Parallel Memories," IEEE Trans. Computers, vol.C-20, no. 12, pp. 1566-1569, Dec. 1971.
- [7] H. Lee, H. K. Cho, D.S. You and J. W. Park, "An MAMS-PP4: Multi-Access Memory System used to improve the processing speed of visual media applications in a parallel processing system," IEICE Trans. Fundamentals. vol. E87 A, no. 11 November, 2004.
- [8] J.H. Lim, S. M. Park, J. W. Park, "Design to Chip with Multi-Access Memory System and Parallel Processor for 16 Processing Elements of Image Processing Purpose," Journal of Korea Multimedia Society Vol. 14, No. 11, pp. 1401-1408 November. 2011.
- [9] J.S. Park, J.H. Kim, K.S. Ko, J.W. Park, "Feature Extraction System for High-Speed Fingerprint Recognition using the Multi-Access Memory System," Journal of Korea Multimedia Society Vol. 16, No. 8, pp. 914-926, August 2013.

저 자 소 개



고 경 식(학생회원)
1986년 동국대학교 전자계산학과 학사졸업.
2006년 충남대학교 산업대학원 전자정보통신공학과 석사졸업.
2006년 남대학교 정보통신공학과 박사과정.

<주관심분야: 병렬처리, 영상처리>



김 재 희(학생회원)
2005년 영산대학교 컴퓨터공학과 학사졸업.
2009년 울산대학교 교육대학원 전자계산학과 석사졸업.
2010년 충남대학교 정보통신공학과 박사과정.

<주관심분야: 병렬처리, 영상처리>



이스라엘(정희원)
2005년 영산대학교 컴퓨터공학과 학사졸업.
2011년 충남대학교 정보통신공학과 석사졸업.
2011년 충남대학교 정보통신공학과 박사과정.

<주관심분야: 병렬처리, 영상처리>



박 종 원(정회원)-교신저자
1979년 충남대학교 전자공학과 학사졸업.
1981년 한국과학기술원 전산학과 석사 졸업.
1991년 한국과학기술원 전산학과 박사 졸업.

1983~1993년 충남대학교 전산학과 부교수
1994년~현재 충남대학교 정보통신공학과 교수
<주관심분야: 영상처리, 병렬처리, 의학영상처리>