

서브어레이 접근을 위한 충돌회피 기억장치

박 춘자⁰, 박 종원

충남대학교 공과대학 정보통신공학과
(whtdeer⁰, jwpark)⁰@crow.cnu.ac.

Conflict-Free Memory System for Subarray Access

Chun-Ja Park⁰ Jong-Won Park

Dept. of Information Communications Engineering Chung-nam National University

요 약

이 논문에서는 pq개의 PE(Processing Element)를 가진 SIMD처리기에서 기억 장치 접근시간을 감소시키기 위한 충돌회피 기억장치를 제안했다. 이 기억장치는 MxN 배열내 자료들의 임의의 위치에서 일정 간격인 블록형태와 8방향 선형태인 pq개의 자료들의 동시 접근을 지원한다. 기억모듈 수는 pq보다 큰 소수이고, 간격은 기억모듈 수의 배수가 아닌 양수이다. 간단하고 빠른 주소계산회로와 이동회로를 위해, 요구된 자료들에서 첫번째 자료의 기준 주소와 pq개의 주소간의 차들로 구분한 후, 주소간의 차들은 첫번째 자료의 기준주소 더해진 후, 첫 번째 요소의 기억모듈 번호에 의한 오른쪽 회전이 간격을 가진 9가지 서브어레이 모두에게 요구된다. 9가지 자료 이동 형태를 멀티플렉싱과 회전에 의해 1가지로 감소시킨 효율적인 자료 이동 회로를 제안하였다. 제안된 충돌회피 기억 장치는 이전 기억 장치와 비교하여 자료 접근형태, 간격, 자료 배열의 크기에 제한, 하드웨어 비용, 속도, 복잡도면에서 개선하였다

1. 서 론

다중접근 기억장치에 의해 제공되는 pq개의 PE(Processing Element)들에서 사용하게 될 서브어레이 내의 자료의 동일 연산속도를 향상시키는 SIMD 컴퓨터를 적용할 여러 응용분야들이 있다. 여기서 서브어레이는 전체 MxN 자료 배열내의 pq개의 요소를 갖는 작은 자료 배열을 의미한다.

자료배열내 임의의 위치에서 다양한 형태 서브어레이 내의 pq개의 자료를 접근하는 병렬 기억 장치가 연구 중이다[1-7]. Lawrie 와 Vora 에 의해 제안된 기억 장치[5]는 일정간격을 갖는 행, 열, 대각선, 역대각선형태 서브어레이의 pq개의 자료들을 동시에 접근하도록 제공한다.

간격 1인 블록, 행, 열, 대각선, 역대각선 형태의 서브어레이 자료의 동시접근을 제공하는 Park[7]에 의해 제안된 주소계산 회로와 이동회로는 이전 기억 장치들[3-6]의 회로보다 하드웨어 비용, 회로제어의 복잡도, 속도 면에서 개선되었다.

이 논문에서 제안된 병렬기억장치는 이전 기억장치들[3-7]보다 주소계산과 이동 방법에 사용될 서브어레이 형태들과 간격들을 확장시킨 블록형태와 8방향 선형태 서브어레이 내의 pq개의 자료들의 동시접근을 제공한다.

소수(prime number) 개의 기억모듈을 가진 다중 접근 기억 장치는 2절과 3절에서 각각 이전의 기억장치와 비교하고 개선하였다.

2. 다중접근 기억장치의 일반적 설계

이 절에서는 서브어레이 형태를 지원하는 다중접근 기억장치의 일반적인 설계에 대해 고찰해 본다.

압축, 텍스처 분석, 또는 움직임 분석에 유용한 가우시안 피라미드 [8-10]의 빠른 생성이나 계층적 이산 상관 윈도우 함수가 서브어레이 접근에 대한 SIMD 연산의 예이다.

이전 레벨의 모든 다른 노드와 모든 2ⁿ 번째 노드의 값은 재귀적으로 또는 직접적으로 레벨 K를 계산하기 위하여 SIMD 컴퓨터 내의 각 PE 에 할당되어야 한다. 그러므로 가우시안 피라미드[6]의 생성에 임의의 위치에서 간격 g (g>0) 을 갖는 블록이나 행 형태내의 영상 점들에 동시 접근을 지원하는 다중 접근 기억 장치가 기억장치 접근시간의 전체적인 감소를 위해 요구되어진다. 이와같이 SIMD 연산들의 기억장치 접근시간의 감소를 위해서 다중접근 기억장치는 기준좌표(i,j)와 간격 r 을 갖는 다음 9가지 서브어레이형태(블록(BL)과 8방향 선형: 동쪽선형(EL), 남동쪽 선형(SEL), 남쪽선형(SL), 남서쪽 선형(SWL), 서쪽 선형(WL), 북서쪽 선형(NWL), 북쪽 선형(NL), 북동쪽 선형(NEL))인 자료들의 동시 접근이 필요하다.

$$BL(i,j,r) = \{(i+ar, j+br) | 0 \leq a < p, 0 \leq b < q, 0 \leq i \leq M-rp, 0 \leq j \leq N-rq\} \quad (1)$$

$$EL(i,j,r) = \{(i+j-ar) | 0 \leq a < pq, 0 \leq i \leq M, 0 \leq j \leq N-rpq\} \quad (2)$$

$$SEL(i,j,r) = \{(i+ar, j+ar) | 0 \leq a < pq, 0 \leq i \leq M-rpq, 0 \leq j \leq N-rpq\} \quad (3)$$

$$SL(i,j,r) = \{(i+ar, j) | 0 \leq a < pq, 0 \leq i \leq M-rpq, 0 \leq j \leq N\} \quad (4)$$

$$SWL(i,j,r) = \{(i+ar, j-ar) | 0 \leq a < pq, 0 \leq i \leq M-rpq, rpq \leq j \leq N\} \quad (5)$$

$$WL(i,j,r) = \{(i, j-ar) | 0 \leq a < pq, 0 \leq i \leq M, rpq \leq j \leq N\} \quad (6)$$

$$NWL(i,j,r) = \{(i-ar, j-ar) | 0 \leq a < pq, rpq \leq i \leq M, rpq \leq j \leq N\} \quad (7)$$

$$NL(i,j,r) = \{(i-ar, j) | 0 \leq a < pq, rpq \leq i \leq M, 0 \leq j \leq N\} \quad (8)$$

$$NEL(i,j,r) = \{(i-ar, j+ar) | 0 \leq a < pq, rpq \leq i \leq M, 0 \leq j \leq N-rpq\} \quad (9)$$

위의 식에 나타난 간격 r인 식(1)-(9)와 기준좌표 (i,j)인 9가지 서브어레이 형태들이 그림 1에 나타나 있다

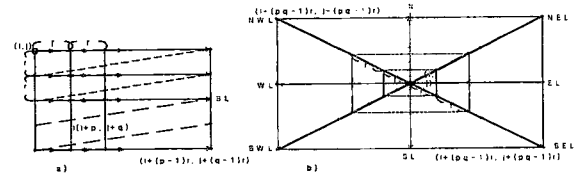


그림 1. 서브어레이 형태들, 간격들, 그리고 기준좌표들(i,j).

- a) 블록형태 서브어레이 (BL).
- b) 8방향 선형 서브어레이(EL,SEL,SL,SWL,WL,NWL,NL,NEL).

여기서 서브어레이 형태들의 사용은 C 프로그램과 그에 상응하는 프로그램으로 표현할 수 있다. 다중접근 기억장치내의 pq 개의 자료가 각각 다른 기억모듈에 동시에 요구되고, 위치에 대한 제한이 없어야 한다. 그림2은 부착되어지는 SIMD 컴퓨터 내의 다중접근 기억장치의 일반적인 설계의 블록도를 보여준다

서브어레이가 저장될 때 기억장치의 구성요소들은 다음의 연산들을 제어회로의 제어에 의해 순서대로 수행된다.

1. 레지스터 t, i, j, r에 지시에 따른 값이 넣어진다. 서브어레이 형태에 따라 각각 1~9 중 한 값을 t에, 기준주소는 i와 j에, 요구되어진 서브어레이 식(1)-(9)의 간격은 r에, 서브어레이 자체는 자료 레지스터에 위치하게 된다.
2. 주소계산과 이동회로는 서브어레이내의 자료 m개의 주소를 계산하고 계산된 주소를 m개의 기억 모듈로 이동한다.
3. 기억 모듈 선택회로는 m개의 기억 모듈 중에서 접근되어지는 pq개의 기억 모듈들을 활성화시킨다.
4. 자료이동회로는 자료레지스터내의 서브어레이의 자료들을 m개의 기억 모듈로 이동시킨다.
5. WRITE 신호에 의해 활성화된 pq개의 기억 모듈에 서브어레이내의 pq개의 자료가 저장된다.

여기서 2, 3, 4 번의 연산이 병렬로 수행된다.

마찬가지로 기억장치로부터 서브어레이의 자료를 가져올 때도 마찬가지로의 과정을 거쳐서, READ신호에 의해 읽어온다. READ시에는 2와 3 번의 연산이 병렬로 수행된다.

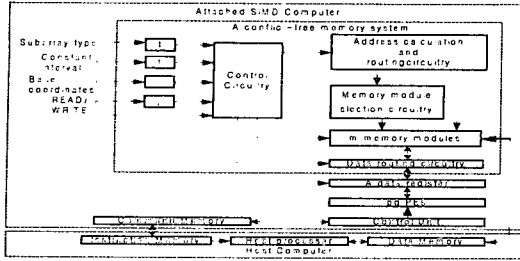


그림2. 다중접근 기억장치의 일반적 설계

2.1 기억 모듈 할당 함수

자료가 위치한 기억 모듈 번호를 결정하는 기억모듈 할당 함수는 다음과 같다.

$$\mu(i,j) = (iq+j)\%m. \quad (10)$$

기억 모듈 할당 함수(식10)는 $m=pq+1$, $2pq$, $pq2$ 인 경우에 대해서 Van voorhis와 Morrin[3]에 의해 연구되었고, $m=pq+1$ 인 경우에 대해 park[4], Park과 Harper[5,6]가 연구하였으며 m 이 pq 보다 큰 소수인 경우에 대해 park[7]에 의해 연구되었다.

정리 1은 간격 식(1)-(9)를 갖는 9가지 서브어레이에 대한 위의 식(10)의 접근성을 보여 준다[7].

정리1 : 기억모듈 할당함수 $\mu(i,j)$ ($m > pq$, m 은 소수)는 $r\%m \neq 0$ 때 간격 r 을 갖는 BL, EL, SEL, SL, SWL, WL, NWL, NL, NEL의 서브어레이내 pq 개의 자료를 다른 기억모듈들에 위치시킨다.

2.2 주소할당함수

기억모듈 내에 자료의 주소를 결정하는 주소할당함수는 다음과 같다.

$$a(i,j) = (i/p)s + j/q \quad (11)$$

여기서 s 는 $\lfloor N/q \rfloor$ 와 같거나 큰 임의의 정수이다.

이 함수는 van voorhis와 Morrin[3], Park[4,7], 그리고 Park과 Harper[6]에 의해 연구 되었다.

정의 : 다중접근기억장치가 요구된 서브어레이내의 모든 자료들에 동시접근을 시작하여 지연시간 없이 동시접근이 요구되어진 서브어레이내의 모든 자료들을 동시에 완벽하게 접근할 수 있을 때 충돌회피(Conflict-Free:CF)라고 한다.

그래서 다중접근 기억장치가 서브어레이에 대해 CF이면 기억 모듈 할당함수는 동시에 접근되는 서브어레이내의 자료들을 다른 기억 모듈내에 위치 시켜야하고 주소할당함수는 같은 기억 모듈내에 할당되는 서브어레이내의 자료들은 다른 주소를 할당 해야 한다. 정리2는 기억모듈 할당함수 $\mu(i,j)$ 와 주소할당함수 $a(i,j)$ 사이의 관계를 보여준다[7].

정리2 : 주소할당함수 $a(i,j)$ 와 기억모듈 할당함수 $\mu(i,j)$ 에 의해 같은 기억 모듈에 할당되는 자료들에 서로 다른 주소를 할당한다.

그래서 CF는 $r\%m \neq 0$ 일 때 간격 r 인 BL, EL, SEL, SL, SWL, WL, NWL, NL, NEL형태 서브어레이내의 주소할당함수에 의해 할당된 주소와 기억모듈할당함수에 의해 할당된 기억 모듈 번호를 갖는 pq 개의 자료들에 접근할 수 있는 다중 접근 기억 장치이다.

2.3 주소계산과 이동

이 절에서는 간격1을 갖는 BL(1), EL(2), SEL(3), SL(4), SWL(5)의 서브어레이에 대한 주소간의 차들을 계산하는 주소계산과 이동방법을 간격 (1)-(9)를 갖는 9가지 접근형태로 확장하였다. 주소계산과 이동방식의 불터도란 $p=q=2$, $M \times N=24 \times 32$, $s=16$, $m=5$ 인 경우에 대해 그림3에서 나타내었다. 5개의 덧셈기는 A,B,C를 통해 각각 입력되는 기준주소와 이전주소차들과 나머지주소차들을 가지고 5개의 주소를 계산한다. 5개의 주소들은 5개의 기억 모듈로 전송되기전에 배럴 시프터에서 기억모듈 할당함수 $\mu(i,j)$ 에 의해 오른쪽 회전을 한다.

2.3.1 모듈번호의 차들

정리1로부터 기억모듈 할당함수에 의해 할당되고 연우선순위로 정렬된 pq 개의 번호들에서 기준주소들이 (i,j) 인 자료요소들의 번호인 기준번호를 뺀 연어된 모듈번호 차들인 $\mu'(r,k)$ 는 $r\%m \neq 0$ 인 간격 (1)-(9)인 9가지 서브어레이에 따라 아래와 같이 표현된다.

$$BL: \mu'(r,k) = (k/q)r + (k\%q)r\%m - (kr)\%m, 0 \leq k < pq. \quad (12)$$

$$EL: \mu'(r,k) = (kr)\%m, 0 \leq k < pq. \quad (13)$$

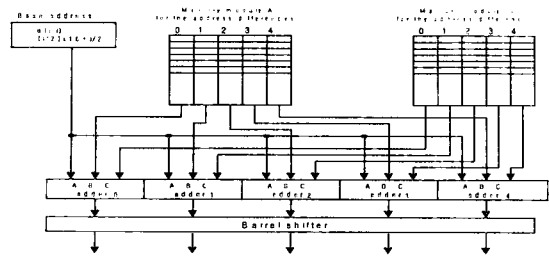


그림3. $p=q=2$, $M \times N=24 \times 32$, $s=16$, $m=5$ 인 경우의 주소계산과 이동회로

$$SEL: \mu'(r,k) = ((q+1)kr)\%m, 0 \leq k < pq. \quad (14)$$

$$SL: \mu'(r,k) = (qkr)\%m, 0 \leq k < pq. \quad (15)$$

$$SWL: \mu'(r,k) = ((q-1)kr)\%m, 0 \leq k < pq. \quad (16)$$

$$WL: \mu'(r,k) = (-kr)\%m, 0 \leq k < pq. \quad (17)$$

$$NWL: \mu'(r,k) = (-(q+1)kr)\%m, 0 \leq k < pq. \quad (18)$$

$$NL: \mu'(r,k) = (-qkr)\%m, 0 \leq k < pq. \quad (19)$$

$$NEL: \mu'(r,k) = (-(q-1)kr)\%m, 0 \leq k < pq. \quad (20)$$

2.3.2 주소간의 차들

간격 식(1)-(9)인 9가지 서브어레이에 대한 기준주소 (i,j) 로부터의 주소간의 차들은 주소할당함수에 의해 할당되어지고 행우선 순위로 정렬되어진 pq 개의 주소들에서 기준주소를 빼서 아래처럼 얻을 수 있다.

$$BL: a'(i,j,r,k) = s((i\%p+(k/q)r)/p) + (j\%q+(k\%q)r)/q, 0 \leq k < pq. \quad (21)$$

$$EL: a'(i,j,r,k) = (j\%q+kr)/q, 0 \leq k < pq. \quad (22)$$

$$SEL: a'(i,j,r,k) = s((i\%p+kr)/p) + (j\%q+kr)/q, 0 \leq k < pq. \quad (23)$$

$$SL: a'(i,j,r,k) = s((i\%p+kr)/p), 0 \leq k < pq. \quad (24)$$

$$SWL: a'(i,j,r,k) = s((i\%p+kr)/p) + (j\%q-kr)/q, 0 \leq k < pq. \quad (25)$$

$$WL: a'(i,j,r,k) = (j\%q - kr)/q, 0 \leq k < pq. \quad (26)$$

$$NWL: a'(i,j,r,k) = s((i\%p-kr)/p) + (j\%q-kr)/q, 0 \leq k < pq. \quad (27)$$

$$NL: a'(i,j,r,k) = s((i\%p-kr)/p), 0 \leq k < pq. \quad (28)$$

$$NEL: a'(i,j,r,k) = s((i\%p-kr)/p) + (j\%q+kr)/q, 0 \leq k < pq. \quad (29)$$

$j\%q$ 에 영향을 받는 주소간의 차들은 이전 항목이기 주소간의 차들의 크기를 감소 시킨다. 식(21) BL형태의 서브어레이의 주소간의 차들은 아래처럼 나뉘어 진다.

$$BL: a'(i,j,r,k) = s((i\%p+(k/q)r)/p) - ((k\%q)r)/q, 0 \leq k < pq \text{ (for input B of each adder)} + (j\%q+(k\%q)r)/q - ((k\%q)r)/q, 0 \leq k < pq \text{ (for input C of each adder)}. \quad (30)$$

BL형태와 같이 나머지 8가지 형태의 주소간의 차들을 구할 수 있다

2.3.2 주소계산과 이동

기억 모듈로부터 기억 모듈번호를 오름차순한 주소간의 차들은 버퍼 기억장치[7]의 처리과정에 따라 얻을 수 있다

버퍼기억장치 시스템의 처리과정 A[7]

1. 주소차들과 행우선순위로 서브어레이 형태의 번호차들을 계산한다
2. 기억모듈 0에서부터 기억모듈들 μ_2 의 번호들에 오름차순 정렬된 주소간의 차들을 계산한다.
3. μ_2 대신 $(\mu_2 + \mu(i,j))\%m$ 를 대치하여 기억모듈 $\mu(i,j)$ 로부터 기억모듈들의 오름차순 정렬된 주소간의 차들을 계산한다.
4. 주소간의 차들을 기준주소에 더한 후 기억모듈 0에서부터 오름차순 정렬된 기억모듈번호들에 주소간의 차들을 $\mu(i,j)$ 번 회전시킨다. BL(12)형태서브어레이 번호차들은 $0 \leq k < pq$ 에 대해 다음처럼 변형된다.

$$k = ((\mu_1(i,j,r,k) - \mu(i,j))r)\%m, 0 \leq k < pq, r \cdot r = 1\%m \quad (31)$$

그래서 $\mu_1(i,j,r,k)$ 번째 모듈의 BL형태 서브어레이의 주소간의 차 $a_1'(i,j,r,\mu_1(i,j,r,k))$ 은 식30에 대신에 식31의 $((\mu_1(i,j,r,k) - \mu(i,j))r)\%m$ 로 대치하고 k 대신에 $\mu_1(i,j,r,k)$ 인 기억 모듈 번호에 따라 주소간의 차들을 배열로 얻을 수 있다.

$$BL: a_1(i,j,r,2) = s((i\%p + (((\mu_2 - \mu(i,j))r)\%m)/q)/r)/p) + (((((\mu_2 - \mu(i,j))r)\%m)/q)r)/q, 0 \leq \mu_2 < m \text{ (for input B of each adder)} + (j\%q + (((\mu_2 - \mu(i,j))r)\%m)/q)r/q - (((((\mu_2 - \mu(i,j))r)\%m)/q)r)/q, 0 \leq \mu_2 < m \text{ (for input C of each adder)}, r \cdot r = 1\%m \quad (32)$$

편의를 위해 기억모듈번호 $\mu_1(i,j,r,k)$ 를 μ_2 로 대치하였다 식(32)에 μ_2 대신 $(\mu_2 + \mu(i,j))$ 를 대치하여 다음의 $\mu(i,j)$ 로부터 오름차순

된 기억모듈번호들에서 BL형태 서버레이의 주소자들을 구했다.

$$BL: a1'(i,j,r,(\mu2+\mu(i,j))\%m) \quad s((i\%p+(((\mu2 \cdot r)\%m)/q)r)/p) \\ +(((\mu2 \cdot r')\%m)\%q)r/q, \quad 0 < \mu2 < m \text{ (for input B of each adder)} \\ + (j\%q+(((\mu2 \cdot r')\%m)\%q)r/q) - (((\mu2 \cdot r')\%m)\%q)r/q, \\ 0 < \mu2 < m \text{ (for input C of each adder)}, \quad r, r' = 1\%m \quad (33)$$

식(33)의 주소자들은 a(i,j)에 더해진 후, BL형태 서버레이의 a1(i,j,r, μ2)를 구해 μ(i,j)만큼 오른쪽 회전시켰다.

0에서부터 오른쪽순된 기억모듈번호들에서 나머지 8가지형태 서버레이의 주소들은 BL형태처럼 같은 처리과정을 이용하여 구할 수 있다. 9가지 주소간의 자들의 크기

BL, SEL, SWL, NWL, NEL형태 서버레이의 주소자들 크기는 주소자들을 위한 기억모듈A에 #r · 5 · p x mlog2(MN/pq)비트와 기억모듈 B에 #r · 5 · q x m비트이다. 여기서 #r은 간격들의 숫자로 나타난다. EL과 WL형태의 크기는 #r · 2 · q x mlog2(MN/pq)와 #r · 2 · q x m비트이다. SL과 NL형태는 주소간의 자가 없다. 그러므로 제안된 기억 장치들을 위한 주소자들의 전체크기는 #r · ((7p + 2q) x m log2(MN/pq) + 7q x m) 비트이다. 간격 (1) (9)인 9가지의 서버레이 전체주소자들 위한 기억모듈 A와 B의 크기는 저장하기에 너무크기 때문에 요구된 서버레이의 주소자들은 컴파일 시간에 계산되어 저장된다.

2.4 자료이동

WRITE연산을 위해, 자료 레지스터에서 m개의 기억 모듈들로 자료들을 이동시키는 자료이동회로 역할은 적절한 기억 모듈을 pq개의 자료들을 이동시키는 것이다. 다음의 수식에 따라 pq개의 자료들 정렬시킨 후 이 자료들의 기억 모듈번호를 가지고 기억 모듈 번호 0부터 시작되는 응답차순 정렬을 위해 오른쪽 회전이 μ(i,j)만큼 수행되어진다.

$$BL: D2((kr)\%m) \leftarrow D1(k), \quad 0 < k < pq, \quad (34)$$

여기서 D1과 D2는 자료 레지스터와 임시 레지스터들이다. 이동형태가 접근 형태와 간격에 영향을 받기 때문에 이동형태 9가지 식을 이용하는 자료이동회로의 제어는 복잡하다. 다음에 따라 제어의 복잡도를 감소시킬 수 있다. 식 (34)의 BL형태 서버레이 자료이동 형태에 대해 간격 r인 기억모듈 번호들인 (kr)%m은 (k · m + 1 + k · r1)%m이다. 여기서, r1 = r%am 이고, 11은 양수이다. 그래서 식(34)의 간격 r인 BL형태 서버레이의 자료이동 형태는 아래식에서 r1(r%am)인 (m-1)개의 서로 다른 이동경로가 선택되어진다.

$$BL: D2((kr1)\%m) \leftarrow D1(k), \quad 0 < k < pq, \quad (35)$$

식 (35)의 BL의 이동형태는 나머지 8가지 형태(EL, SEL, SL, SWL, WL, NWL, NL, NEL) 서버레이에 r1, ((q+1)r1)%m, (qr1)%m, ((q-1)r1)%m, (r1)%m, ((q+1)r1)%m, (qr1)%m, ((q-1)r1)%m 안에 r1을 바꿈으로 이동패턴을 사용할 수 있다. 9가지 접근 형태에 대한 전체 이동형태는 (m-1)개이다. 그래서 자료 이동회로는 간격 (1)-(9)인 9 가지 형태의 서버레이에 대해 식(35)의 BL형태 하나로 감소된다. 제안된 자료 이동 회로는 그림4에 나타나 있다.

그림에서 ROM, log2((m-1)x(m-1)) 디코더, 3상버퍼를 사용한 셀렉터를 이용하여 간격이 있는 9가지 형태의 서버레이는 (m-1)가지의 이동형태를 만든다.

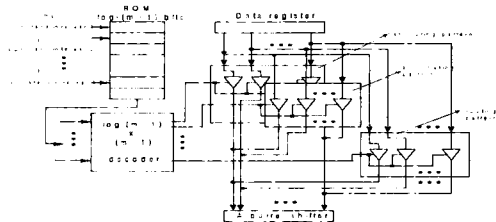


그림4. WRITE연산을 위한 자료이동회로

3. 이전의 기억장치와의 비교

이 절에서는 제안된 주소계산과 이동, 그리고 자료의 이동 방법을 서버레이의 형태, 간격, 자료배열의 크기의 제한에 대해 이전 기억장치 [3-7]와 비교하였다. 주소계산과 주소이동회로[7]는 이전의 것[3-6]에 비해 하드웨어의 비용, 회로 제어의 복잡성, 속도면에서 개선되었다.

Lawrie와 Vora[5]는 간격과 자료 배열의 크기와 기억 모듈의 수가 제약이 있다. 이전 기억 장치[3-7]와 이 논문에서 제안된 기억 장치는 자

료 배열의 크기에 대한 제약이 없다. 제안된 기억 장치의 제약은 이전의 기억장치[5]처럼 정리1로부터 간격이 기억모듈 개수의 배수가 아니어야 한다는 것이다. 이전 기억장치[3]에 있던 자료이동회로는 μ(i,j)에 의해 제어되는 가변적인 오른쪽회전이 요구되었다. Park[4,7], Park과 Harper[6]에 의해 제안된 기억장치의 자료이동은 멀티플렉서와 배럴시프터에 의해 수행된다. 이 기억 장치[5]에서는 이전 기억장치[3]의 자료 이동 회로내의 가변적인 오른쪽 회전 변환기를 복잡하고 비싸고 느린 crossbar네트워크에 의해 대체했다. 이 논문에서 멀티플렉서와 회전을 사용한 자료이동 방법이 제안된다. 여기서, 9가지 형태의 서버레이에 대해 양수의 간격을 가지는 9가지 자료이동 형태가 ROM과 log2(2(m-1)) x 2(m-1)디코더, 3상 버퍼들을 이용한 셀렉터, 그리고 배럴시프터에 의해 구현된 간격 r1(r%am)인 1개 이동 형태로 감소된다. 그러므로 주소계산, 주소이동 그리고 자료이동에 대해 이전의 기억장치 [3-7]보다 제안된 방법은 서버레이 형태들, 간격 그리고 자료배열의 크기에 대한 제한 면에서 개선되었다.

4. 결론

다양한 영상과 신호 처리 그리고 행렬연산들은 호스트컴퓨터에 부착되어진 pq개의 PE로 구성된 SIMD컴퓨터에 의해 속도를 향상시킬 수 있다. 부착된 SIMD 컴퓨터는 자료의 위치, 간격, 자료배열의 크기에 제약이 없는 자료 배열내의 간격이 있는 서버레이의 다양한 형태들이 갖는 pq개의 자료요소들을 동시 접근할 수 있는 기억장치가 요구된다.

이 논문에서는 배열의 크기, 자료 요소의 위치에 제약이 없는 양수 간격을 가지는 하나의 블록 또는 8방향 선형(EL, SEL, SL, SWL, WL, NWL, NL, NEL) 서버레이의 pq개 자료요소에 동시접근을 지원하는 소수개의 기억 모듈을 가진 중첩회피 기억 장치를 설명했다.

이 논문에서는 기억 장치에서 효율적인 주소계산 회로와 주소 이동 회로, 그리고 자료이동회로가 간격 1인 클리, 행, 열, 대각선, 역대각선 서버레이에 대해 확장된 버전의 기억장치들 위해 제안되었다. 고속 기억 모듈에서 주소간의 차이를 미리 결정하고 저장하면 예전의 회로와 비교하여 주소의 계산회로와 이동 회로를 빠르고 간단하게 제어하게 된다. 더 나아가, 9가지의 자료 이동 형태가 1개의 이동형태로 감소된 효율적인 자료이동 회로를 제안하였다. 따라서, 제안된 중첩회피 기억 장치는 예전 기억 장치 이상으로 개선되었다. 하나의 자료 배열내에 임의의 위치에 있는 적어도 3가지 접근형태의 자료요소들의 동시 접근이 지원된다. 그리고 접근 형태, 간격, 자료배열 크기의 제약과 하드웨어 비용, 속도, 복잡성의 제한 내에서 기억 장치내의 주소계산과 주소 이동회로, 그리고 자료이동회로의 하드웨어 구현이 고려된다.

REFERENCES

[1] P. Budnik and D. J. Kuck, "The organization and use of parallel memories," *IEEE Trans Comput.*, vol. C-20, pp.1566-1569, Dec. 1971.
 [2] D. H. Lawrie, "Access and alignment of data in an array processor," *IEEE Trans. Comput.*, vol. C-24, no. 12, pp. 1145-1155, Dec. 1975.
 [3] D. C. Van Voorhis and T. H. Morrin, "Memory systems for image processing," *IEEE Trans. Comput.*, vol. C-27, pp.113-125, Feb. 1978.
 [4] J. W. Park, "An efficient memory system for image processing," *IEEE Trans. Comput.*, vol. C-35, pp.669-674, July 1986.
 [5] D. H. Lawrie and C. R. Vora, "The prime memory system for array access," *IEEE Trans. Comput.*, vol. C-31, pp.435-442, May 1982.
 [6] J. W. Park and D. T. Harper III, "An efficient memory system for the reconstruction of a Gaussian pyramid," *IEEE Trans. Parallel Distrib.Syst.*, vol. 7, No. 8, pp.855-860, Aug. 1996.
 [7] J. W. Park, "An efficient buffer memory system for subarray access," *IEEE Trans. Parallel Distrib.Syst.*, vol. 12, No. 3, pp.316-335, Mar. 2001.
 [8] A. Rosenfeld, *Multiresolution image processing and analysis*, Springer-Verlag, 1984.
 [9] P. J. Burt, "Fast filter transforms for image processing," *Comput.Vision, Graphics, Image processing* 16, pp.20-51, 1981.
 [10] P. J. Burt, "Fast algorithms for estimating local image properties" *Comput. Vision, Graphics, Image processing* 21, pp.368-382, 1983.

"이 연구는 BK21 충남대학교 정보통신인력양성 사업단의 지원을 받아서 수행하였습니다."