

SIMD 어레이 프로세서의 기술동향

아주대학교 장현만* · 응수환* · 선우명훈**

● 목	차 ●
1. 서 론	2.4 데이터 임출력
2. 아키텍처 구조	2.5 자치권(Autonomy)
2.1 상호연결망	2.6 Bit-serial 대 bit-parallel 방식
2.2 근접통신	3. 분석적 모델과 알고리즘 복잡성
2.3 비근접통신	4. 결 론

1. 서 론

SIMD(single instruction stream-multiple data streams) 어레이 프로세서(array processor)들은 많은 프로세서들을 연결하여 같은 명령어를 수행함으로써 본질적으로 수천 또는 수만 개의 프로세서를 사용 대규모 병렬성(massive parallelism)을 얻을 수 있다. 그러므로 실시간 신호 및 영상처리, 고속의 그래픽스 처리 등과 같이 막대한 양의 데이터를 취급하는 응용 분야에서 중요한 역할을 한다[1]. Bell Lab의 Unger가 이차원적 구조를 갖는 데이터의 처리를 위하여 격자연결 구조(mesh topology)에 기초를 둔 컴퓨터를 최초로 제안한 이후[2], 신호와 영상 데이터의[1-3] 구조에 적합한 많은 어레이 프로세서 병렬 컴퓨터들이 제안되어졌고 실제 공학 및 과학 분야에서 광범위하게 사용되고 있다[4, 5]. 본 논문에서는 SIMD 어레이 프로세서 구조에 대해 최근의 기술동향을 조사 분석한다.

어레이 프로세서들을 이루는 한 개의 PE(processing element)는 간단한 연산만을 수행

할 수 있는 ALU(arithmetic logic unit), 레지스터, 멀티플렉서 등으로 작고 효율적으로 설계하여 그 크기가 범용 프로세서보다 작다. 따라서 다수의 PE들을 한 개의 VLSI 칩안에 구현시킬 수 있다. 이 VLSI 칩 설계시 확장성을 두기 때문에 많은 양의 칩들을 연결하여 병렬 어레이 프로세서 컴퓨터를 구현할 수 있다. 특히 최근 VLSI 기술의 발전으로 PE를 수백개 이상 하나의 칩안에 집적시킬 수 있어 대폭적인 성능 향상을 이루게 되었다. 따라서 SIMD 어레이 프로세서들은 MIMD(multiple instruction streams-multiple data streams) 병렬처리 컴퓨터들보다 프로세서 개수가 월등히 많다. 통상적으로 어레이 프로세서들은 수천 또는 수만 개의 프로세서들로 구성되어지고 MIMD 병렬처리 컴퓨터는 수십 또는 수천 개로 구성된다. 더욱이 프로그래밍 환경 즉, 프로그래밍 언어, 컴파일러, 시뮬레이션 도구들의 개발은 MIMD 병렬처리 컴퓨터에서 보다 훨씬 용이하다. 이러한 장점 때문에 어레이 프로세서는 데이터 양이 막대한 실시간 레이다 신호처리, 영상신호의 움직임 추정, 부품검색, 컴퓨터 시각인식, 공장자동화, 가상현실(virtual reality), 애니메이션(animation), 정찰비행영상, 위성영

*미 회 위
**중신회원

상처리 등과 같은 분야에서 사용이 확대되고 있다.

어레이 프로세서들에서 사용되어지는 PE들끼리의 상호연결망(interconnection network)은 격자(mesh), 선형연결(linear array), 하이퍼큐브(hypercube), 트리(tree), 버스(bus), 피라미드(pyramid) 등이 있다. 격자연결 구조를 사용한 SIMD 어레이 프로세서들은 SOLOMON[6], ILLIAC IV[7], MPP[8, 9], CLIP[10, 11], DAP[12], GAPP[13], GRID[14], NTT[15], BAP[16], CAAPP[17], YUPPIE[18], 재구성 가능한(reconfigurable) 버스를 가진 MESH[19], CM 1[20], MasPar[21], BLITZEN[22], MGAP-2[23], SliM[24-26] 등이 있다. 그리고 선형연결 구조의 SIMD 어레이 프로세서는 CLIP7A[11], IMAP[27, 28], VIP[29], AIS[30], SLAP[31] 등이 있다.

이러한 어레이 프로세서들이 SISD(single instruction stream - single data stream)나 MIMD 컴퓨터와 비교하여 신호 및 영상처리 알고리즘에 대하여 성능 면에서 훨씬 좋지만, 몇 가지의 단점들을 가지고 있다. 이웃한 PE끼리의 데이터 교환을 위한 근접통신 시간부담(inter-PE communication overhead), 각 PE들이 서로 다른 독립적인 명령어를 수행할 수 있는 자치권(autonomy) 부재 등은 효율적인 신호 및 영상처리 알고리즘들의 수행에 있어 제한을 받는다. 특히 계산 중에 이웃 PE들간에 있어 정보 교환을 위한 많은 양의 근접통신이 발생한다. 이러한 근접통신 시간부담은 기존의 격자연결 구조의 SIMD 컴퓨터들의 심각한 단점이다[3, 41-44]. 근접통신 시간부담의 단점을 극복하고 성능을 향상시키기 위하여 새로운 격자연결 구조의 SliM 어레이 프로세서가 개발되었다[24-26]. 또한 최근 개발된 어레이 프로세서들은 SIMD 컴퓨터들이 마치 MIMD 컴퓨터들 같이 각각의 PE들이 서로 다른 연산들을 수행하는 자치권을 가진다. 이 자치권에 의해 SIMD 컴퓨터의 각 PE는 서로 다른 명령어를 수행하여 복잡한 계산을 처리할 수 있다. 따라서 어레이 프로세서는 최근 응용 범위의 폭이 점차 넓어지고 있는데 예를 들면, 데이터

베이스, 반도체 회로분석, 유체역학, 가상현실, 컴퓨터 애니메이션, 거래처리 등의 분야에서 사용되며 활용 분야도 더욱 확대될 것이다[20, 21].

본 논문에서는 특히 최근에 개발된 CLIP7A, DAP, CAAPP, MasPar, BLITZEN, MGAP-2, SliM, IMAP, VIP를 중심으로 보다 자세한 비교를 논의한다. 2절에서는 SIMD 어레이 프로세서들의 아키텍처 구조와 여러 SIMD 프로세서들을 비교 분석하고 3절에서는 SIMD 어레이 프로세서의 분석적 모델과 영상처리 알고리즘 복잡성을 이용 어레이 프로세서들을 비교 분석하며 마지막으로 4절에서 결론을 맺는다.

2. 아키텍처 구조

그림 1은 일반적인 SIMD 구조를 도시하며 이는 여러 개의 서로 다른 데이터에 대해서 동일한 명령어를 수행하므로 하나의 제어 유닛이 명령어를 N개의 서로 다른 PE들에게 전달하여 각각의 PE에 있는 데이터에 대해서 같은 명령어를 수행한다.

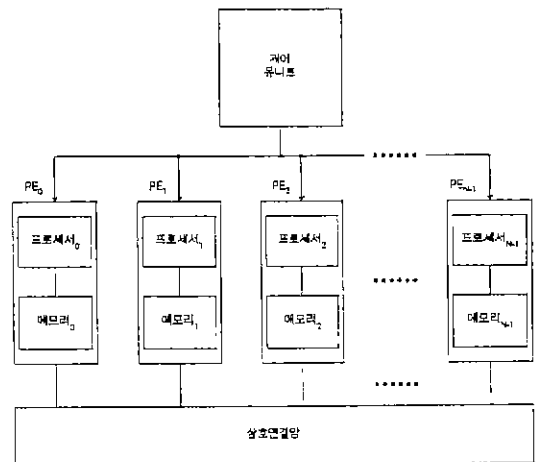


그림 1 일반적인 SIMD 구조

이 절에서는 아키텍처 구조들중 특히 성능에 직접적인 관련이 있는 상호연결망(inter connection network), 근접통신(inter-PE communication), 비근접통신(non-local communication), 데이터 입출력, 자치권(autonomy),

표 1 어레이 프로세서 아키텍처 구조 비교

	BLITZEN	MasPar	CAAPP	CLIP7A	DAP	MGAP-2	IMAP	VIP	SLiM
상호 연결망	X-Net	X-Net, 다단 크로스바	격자, Coteric	8 방향 격자, 선형	격자, X-Y 버스	격자	선형	선형	격자
근접 통신	X-Net	X-Net	격자	8 링크 & edge storage	격자	재구성 격자	선형	선형	S 평면
비근접 통신	-	다단 크로스바	Coteric	-	X-Y 버스	-	-	-	by-passing 경로
통신 경로	1 비트	1 비트	1 비트	1 비트	1 비트	1 비트	8 비트	16비트	1 비트
데이터 경로	1 비트	4 비트	1 비트	16 비트	1 비트 프로세서, 8 비트 보조 프로세서	1 비트	8 비트	1 비트	8 비트
연산 자치권	G & K 레지스터	E-비트 플래그	A 레지스터	condition 레지스터	A 레지스터	enable 레지스터	마스크 레지스터	-	C 레지스터
주소 자치권	쉬프트 레지스터	PMEM 유닛	CAM	CLIP7 보조 프로세서	-	-	어드레스 레지스터	-	A & C 레지스터
연결 자치권	P 레지스터	X-Net	MR 레지스터, Coteric	condition 레지스터	-	구성 레지스터	-	-	C 레지스터, SW
데이터 입출력	4 비트 버스	채널 스타일 구조	HCSM	D 레지스터	D 평면 & 커플러	링크 레지스터	쉬프트 레지스터	32 비트 입출력 레지스터	D & D' 평면

- : 지원안됨

데이터경로 폭(datapath width)에 초점을 맞추어 여러 SIMD 어레이 프로세서들의 아키텍처 구조들을 자세히 비교한다. 또한 각 프로세서에 유일하게 존재하는 독특한 아키텍처 구조들에 대해서도 간략히 논의한다. 표 1은 본 논문에서 조사한 아키텍처들의 구조들을 비교한 것이다.

2.1 상호연결망

어레이 프로세서에서 사용되는 여러 가지 상호연결망(interconnection network) 중에서 격자(mesh) 및 선형(linear) 구조를 살펴본다. 격자연결 구조에 기초한 SIMD 어레이 프로세

서에는 SOLOMON, ILLIAC IV, MPP, CLIP, DAP, GAPP, GRID, NTT, BAP, CAAPP, YUPPIE, 재구성(reconfigurable) 버스를 갖는 Meshes, CM I, MasPar, BLITZEN, MGAP-2 등이 있으며 선형연결 구조에 기초한 것은 CLIP7A, IMAP, VIP, AIS, SLAP 등이 있다.

CLIP4, CLIP7, BAP와 NTT가 대각선 통신의 시간부담을 줄이기 위해 PE당 6개나 8개의 통신 링크를 가지고 있는 반면 대부분의 격자연결 아키텍처들은 4개의 링크를 가지고 있다. BLITZEN과 MasPar에서 사용된 tri-state 연결망인 X-Net은 4개의 링크만으로 8가지 방향의 연결성을 가진다. 그림 2에서 보

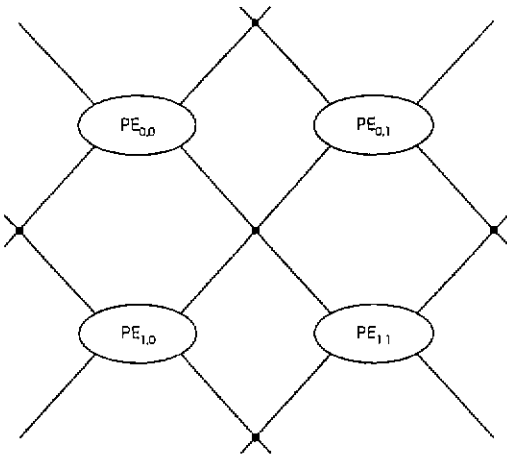


그림 2 X-Net 상호연결망

여주는 X-Net은 모든 PE를 주위 8개의 PE와 연결시킨다. 모든 PE는 대각선 방향으로 4개의 링크가 있으며 링크끼리 만나는 각 X 교점에는 tri-state 노드가 있어 주위 8개 PE와의 통신을 가능하게 한다. MasPar는 비근접통신을 위해 다단 크로스바(multistage crossbar) 상호연결망을 가지고 있다. 다단 크로스바 상호연결망은 비트 단위의 동기식 회로 스위칭(circuit switching)의 통신 방식을 사용하며, 64×64 의 크로스바 스위치인 라우터 칩이 목적지(destination) 주소를 디코딩하여 입력력을 연결한다.

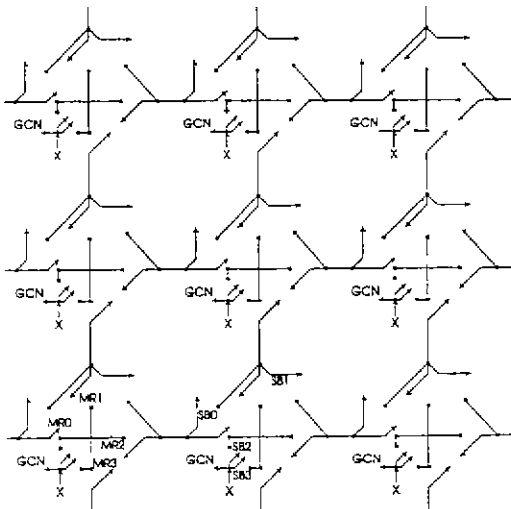


그림 3 Coterie 상호연결망

CAAPP는 4개의 이웃 PE들과 격자연결되어 연결된 영역 레이블링(connected component labeling) 알고리즘 및 비근접통신에 효과적인 Coterie 연결망을 가지고 있다. 그림 3에 도시된 3×3 Coterie 상호연결망은 MR-SB 레지스터가 연결을 제어한다. 모든 PE에 데이터를 전달할 때 GCN과 X를 이용하여 수평, 수직방향의 경로를 설정한다. DAP는 근접통신을 위한 4개의 링크와 수평, 수직방향 비근접통신을 위한 X-Y 버스를 가지고 있으며, MGAP-2는 4개의 링크를 가지고 초기화 과정에 따라 연결 방식의 재구성이 가능하다. ShiM은 명령어 사이클 시간부담 없이 8가지 방향의 연결을 위해서 by-passing 경로를 이용하여 8개의 링크 대신 4개의 링크만으로 대각선 통신을 효율적으로 수행한다. IMAP과 VIP는 선형연결 아키텍처로 좌우 PE와 연결된 2개의 링크를 가지고 있어서, 근접통신과 비근접통신에 비효율적이다.

2.2 근접통신

같은 링크로 연결된 이웃 PE들끼리 서로 데이터를 교환하는 근접통신은 데이터 처리 과정 동안 많이 일어난다. 이러한 근접통신의 시간부담은 기존의 격자연결 방식의 SIMD 아키텍처의 심각한 단점이다[32-35]. Fang[35]은 일반적인 격자연결 SIMD 아키텍처 상에서 $W \times W$ 윈도우 연산자에 대한 2차원 콘벌루션 알고리즘을 수행하기 위해 요구되는 근접통신 수는 $O(W^2)$ 의 통신 복잡성을 갖는다고 논문에서 밝혔다. 이러한 근접통신은 성능 저하를 유발할 수 있으므로 어레이 프로세서에 있어서 중요한 문제이다.

새로운 어레이 프로세서를 기술하기 전에 우선 1980년대 초의 어레이 프로세서인 MPP와 LIPP를 살펴본다. MPP는 근접통신을 위한 하드웨어 유니트가 없어 통신을 연산과 같은 시간에 중첩하여 수행할 수 없으며 1비트의 근접통신을 위해 하나의 명령어 사이클이 소요된다[8, 9, 32]. 근접통신 시간부담을 줄이기 위해 개발된 LIPP[33]의 성능은 1980년대 초의 어레이 프로세서들의 성능과 비교하여 상당히 개선되었다. 그러나 LIPP는 복잡한 게이트 논

리회로와 멀티플렉싱(multiplexing)을 위한 제어가 복잡하며[2] 메모리 모듈에서 읽힌 데이터는 최종 목적지에 도달하기 전에 다수의 멀티플렉서를 통과하게 되고[33] 심각한 전달 지연을 일으킨다. 그러므로, 윈도우 연산자의 크기가 3×3 보다 클 경우 LIPP도 근접통신 시간부담을 줄일 수 없다.

MPP를 향상시켜 최근에 상용화된 BLITZEN과 MasPar는 4개의 링크로 8가지 방향의 연결성을 갖는 X-Net을 가지고 있으며, 이는 대각선 통신을 위해 고안된 특별한 형태의 상호 연결망이다. 그러나 이 X-Net은 대각선 통신에 효율적이다. 근접통신을 위한 전용의 하드웨어 유니트가 없어 근접통신을 연산과 중첩시킬 수 없다[21].

따라서 BLITZEN과 MasPar는 MPP와 같이 1비트의 근접통신을 위해 하나의 명령어 사이클을 요구한다. MasPar는 이웃 PE에 8비트를 보내기 위해서 X-Net의 tri-state 초기화를 위한 3개의 명령어 사이클과 8비트를 위한 8개의 명령어 사이클이 소요되어 총 11개의 명령어 사이클이 요구된다.

CLIP7A PE는 두 개의 CLIP7 칩으로 구성되어 있는데 한 개는 프로세서로, 다른 한 개는 보조 프로세서로 사용된다. CLIP7은 초기에 8가지 방향의 격자연결 구조로 설계되었지만, CLIP7A에서 두 칩은 선형연결 구조로 구성되어 있다. 프로세서가 주로 데이터를 처리하는 반면에 보조 프로세서는 앞으로 설명할 주소 자치권(address autonomy)을 위한 독립적인 주소 생성을 담당한다. 보조 프로세서는 이웃하는 3×3 데이터를 효율적으로 처리하는 방식을 제공하지만 단지 이웃하는 3×3 데이터에만 제공되며, 다양한 데이터 버스를 분리하거나 연결하기 위해서는 보조 프로세서와 칩 외부에 보조 논리 회로(래치, 트랜시버, 버퍼 등)가 필요하다.

AMT DAP 510과 610은 하나의 PE가 두 개의 프로세서를 갖는다. 1비트 프로세서는 통신과 계산을 담당하며 8비트 보조 프로세서는 계산만을 담당한다. 근접통신은 보조 프로세서가 계산하는 동안 1비트 프로세서를 통하여 이루어진다. 논리 연산과 정수 덧셈 같은 간단한

계산들은 1비트 프로세서에서 수행되며[12], 이와 같은 간단한 계산 동안에 근접통신은 수행될 수 없다. 이러한 이유로 1비트 PE는 계산과 부분적인 근접통신의 중첩을 위한 8비트 보조 프로세서를 필요로 한다.

CAAPP는 동서남북 4방향으로 이웃한 PE들의 메모리에 접근할 수 있는 격자연결 구조를 가지고 있어 부분적으로 근접통신을 중첩시킬 수 있다. 가장 근접한 이웃 PE들 사이의 Coterie 연결망은 연결된 영역 레이블링 알고리즘에는 유용하지만 다른 알고리즘에는 효율적이지 못하다. CM I은 근접통신을 위한 격자연결과 비근접통신을 위한 하이퍼큐브 경로를 가지고 있지만 근접통신 중첩은 시킬 수 없다. CM II는 격자연결을 가지고 있지 않고 단지 하이퍼큐브 경로만을 가지고 있어 근접통신에 있어 비효율적이다.

MGAP-2는 구성(configuration) 레지스터를 갖고 있어서 연결 구조를 다양하게 재구성할 수 있다. 이러한 연결 구조는 여러 비트로 이루어진 워드 단위의 데이터를 계산할 때 PE들을 워드 셀(word cell) 단위로 구성하게 한다. 수평, 수직, S자 형태의 스네이크(snake) 등의 다양한 연결을 위해 초기화를 해야 하며 워드 길이가 다른 데이터를 처리하기 위해 한번 고정된 연결 구조를 수정하기 위해서는 다시 초기화해야 하는 단점이 있다. IMAP과 VIP는 선형연결 구조로서 두 개의 링크를 가지고 있다. IMAP은 근접통신이 계산과 중첩되지 않으나 8bit-parallel로 근접통신을 수행하여 근접통신 시간부담을 줄인다. VIP는 근접통신 명령을 계산 명령과 동시에 수행할 수 있는 16비트의 쉬프트 레지스터가 존재하여 근접통신과 계산을 중첩시킬 뿐만 아니라 16bit-parallel로 근접통신을 수행하여 근접통신 시간부담을 줄인다.

SIM은 PE간의 근접통신을 중첩시키기 위한 S 레지스터 평면(plane)이 있고 이는 PE들과 독립적으로 동작하며 보조 프로세서나 칩 내에 DMA(direct memory access)를 필요로 하지 않는다. 따라서 윈도우 크기 및 형태에 상관없는 PE들간의 근접통신 중첩은 근접통신 시간부담을 줄일 수 있다.

2.3 비근접통신

다른 링크로 연결된 이웃하지 않는 PE끼리 데이터를 교환하는 것을 비근접통신이라 하며, 하나 이상의 링크를 거쳐 수행된다. 일부 격자 연결 SIMD 어레이 아키텍처는 전형적인 격자 연결 구조의 단점인 비근접통신 시간부담을 줄이기 위한 특별한 하드웨어 구조를 가지고 있다. 만약 이와 같은 특별한 구조가 없다면 좌측 상단에 있는 PE가 우측 하단에 있는 PE로 데이터를 전송할 때 $N \times N$ 의 격자연결 아키텍처에서 $O(N^2)$ 스텝이 걸린다. 여기서 하나의 스텝이란 서로 근접한 PE간에 하나의 기본 데이터를 옮기기 위한 시간을 의미한다. DAP, YUPPIE 및 LIPP는 비근접통신을 직접 지원해 줄 수 있는 X-Y 버스를 가지고 있으나, 버스는 VLSI 칩에서 많은 면적을 차지하게 되고 버스 충돌이 생기면 병목현상이 생기는 단점이 있다. MasPar는 비근접통신을 위해 다단 크로스바 상호연결망을 가지고 있으며 Slim의 by-passing 경로는 효과적으로 비근접통신을 제공할 수 있다. CAAPP의 Coterie 연결망은 비근접통신을 제공하며 CM I 과 CM II는 비근접통신에 적합한 하이퍼큐브로 연결되는 경로를 가지고 있다. 반면 CLIP7, MPP, BLITZEN, MGAP-2, IMAP 및 VIP는 직접적으로 비근접통신을 수행할 능력을 가지고 있지 않다.

2.4 데이터 입출력

데이터 입출력은 전반적인 성능에 대한 병목현상의 원인이 되어 결과적으로 전체적인 성능을 저하시키게 된다. 어레이 프로세서에서 입출력과 계산이 동시에 일어난다면 입출력 시간부담이 계산과 중첩될 수 있어 입출력 시간부담을 감소시킬 수 있다. MPP와 BLITZEN은 고속의 입출력 전용 시스템이 있지만, 데이터 입출력을 계산과 중첩시키지는 못한다. MasPar는 채널 형태의 입출력 아키텍처를 가지고 있으며, 입출력과 계산을 중첩시킬 수 있는 라우터(router)라 불리는 다단 크로스바 상호연결망을 가지고 있다. CLIP7 칩은 독립적인 입출력 레지스터를 가지고 있으며, CLIP7A에서는 프로세서와 보조 프로세서가 독립적으로 계산

과 데이터 입출력을 제어할 수 있어 입출력과 계산이 중첩될 수 있다. DAP는 입출력을 중첩시키는 DMA 콘트롤러와 같은 역할을 하는 D 레지스터 평면과 커플러(coupler)를 가지고 있다. IMAP과 VIP는 데이터 입출력 레지스터를 계산과 독립적으로 제어함으로써 입출력을 중첩시키며, Slim은 입출력 중첩을 제공하는 두 개의 입출력 레지스터 평면(D와 D')을 가지고 있다. 그러나, CAAPP는 HCSM(host-CAAPP shared memory)을 지원하나 입출력 시간 중첩을 지원하지 않는다.

2.5 자치권(Autonomy)

앞에서 언급한 바와 같이 일반적인 SIMD 구조는 여러 개의 서로 다른 데이터에 대하여 동일한 명령어만 수행할 수 있으며, 상호연결망 중에서 격자연결 구조는 비근접통신에 있어서 비효율적이다. 이러한 단점을 극복하고 성능을 향상시키기 위해 Maresca[5]는 SIMD 어레이 프로세서의 세 가지 다른 자치권, 즉 연산 자치권(operation autonomy), 주소 자치권(address autonomy), 연결 자치권(connection autonomy)을 정의하였다. 이들 자치권에 따라 어레이 프로세서는 효율적인 계산 능력을 가지게 되며 더욱 복잡한 알고리즘에 대해서 이용될 수 있다. SIMD 어레이 프로세서는 같은 시간에 모든 PE들이 같은 연산을 수행하도록 제한된다. 그러나 새로운 어레이 프로세서는 자치권을 가지고 있어 각 PE는 같은 시간에 다른 연산을 수행할 수 있어 SIMD 구조를 MIMD 구조처럼 사용한다. 각 자치권에 대한 설명은 다음과 같다.

연산 자치권(operation autonomy) : 각 PE들은 데이터와 PE의 상태에 따라 서로 다른 연산을 수행할 수 있다. 조건적인 연산들을 위한 플래그(flag)를 사용하여 이를 조건으로 이용 IF (condition) THEN (action)문과 CASE (condition) (action)문과 같은 조건적인 제어 흐름을 실행할 수 있다.

주소 자치권(address autonomy) : 일반적인 SIMD에서 모든 PE는 똑같은 주소의 데이터를 페치(fetch)한다. 이러한 단점을 없애기 위해 각 PE에 있는 인덱스 또는 주소 레지스터

를 사용해서 주소 자치권을 제공한다. 따라서 각 PE는 서로 다른 메모리 위치의 데이터를 폐치할 수 있으며 다양한 종류의 데이터 구조를 효과적으로 구현한다.

연결 자치권(connection autonomy) : 연결 자치권은 효율적인 통신 경로와 재구성 연결 능력을 제공한다. 연결 자치권에 의해서 다양한 연결 방식이 제공될 수 있으며 다양한 통신이 이루어 질 수 있어 보다 복잡한 알고리즘들을 수행할 수 있다.

SIMD 어레이 프로세서들의 자치권 부재를 보완하기 위한 방편으로 새로운 어레이 프로세서들은 이들 자치권을 채택하고 있다. CLIP7A는 보조 프로세서를 이용하여 연산, 연결 및 주소 자치권을 제공한다. CAAPP는 A와 MR 레지스터, Coterie 연결망, 그리고 CAM(content addressable memory)을 이용함으로써 세 가지 형태의 모든 자치권을 가지고 있다. BLITZEN은 G와 K 레지스터를 가지고 연산 자치권을, P 레지스터로 연결 자치권을 가지며 가변 길이 쉬프트 레지스터를 이용하여 주소 자치권을 가진다. MasPar는 플래그에 있는 E 비트를 이용한 연산 자치권과 X-Net을 이용한 연결 자치권, 그리고 PMEM 어드레스 유니트를 이용한 주소 자치권을 가진다. DAP는 A 레지스터를 이용한 연산 자치권을 가지고 있으며, MGAP-2는 구성 레지스터로 연결 자치권을 제공하며 enable 레지스터로 연산 자치권을 부여한다. IMAP은 마스크(mask) 레지스터로 연산 자치권을, 어드레스 레지스터로 주소 자치권을 제공하는 반면 VIP는 자치권을 제공하지 않는다. SliM은 A와 C 레지스터 그리고 멀티플렉싱 역할을 하는 스위치(SW)를 사용하여 세 가지 자치권을 지원한다.

2.6 Bit-serial 대 bit-parallel 방식

Bit-serial과 bit-parallel 방식에 대해서 어느 방식이 성능 면이나 가격 면에서 이로운가에 대한 논쟁은 여전히 계속되고 있다. Bit-serial 방식은 경제적이고 집적성이 용이한 반면 bit-parallel 방식은 고속이고 프로그램을 용이하게 한다. MPP, CLIP, DAP, GAPP,

GRID, CAAPP, CM I, BLITZEN, MGAP-2 등은 bit-serial 통신과 계산에 기초를 둔 반면, SOLOMON, ILLIAC IV, CLIP6, IMAP 등은 bit-parallel 통신과 계산을 기초로 한다. MasPar, CLIP7, DAP 및 SliM은 고속과 경제성 모두를 제공하기 위하여 최근의 추세는 bit-serial 통신과 bit-parallel 계산을 채택하고 있다. 특이하게 VIP는 bit-parallel 통신과 bit-serial 계산을 채택하였다.

신호와 영상처리에서는 대부분의 연산이 이진 데이터보다 정수 형태로 수행되므로 bit-parallel 방식이 보다 효율적이다[10, 11]. 더욱이 VLSI 기술의 발전은 bit-parallel 방식으로도 보다 집적된 시스템의 개발을 가능하게 한다. 그러나 PE들 사이의 bit-parallel 링크들은 VLSI 면적의 많은 부분을 차지하고 VLSI 칩안에서 입출력 핀들의 수를 증가시키는 원인이 된다. 그러므로 이웃하는 PE들간의 링크가 1비트인 bit-serial 방식을 채택함으로써 VLSI 면적을 보다 작게 차지할 수 있고 입출력 핀수도 감소시킬 수 있다. 어레이 프로세서에서의 이러한 추세는 bit-serial 통신과 bit-parallel 계산을 도입하고 있는 MasPar, CLIP7, DAP 및 SliM에서 발견할 수 있다. DAP의 보조 프로세서는 8비트이지만 프로세서 자체는 bit-serial이다. MasPar는 4비트 ALU를 제공하여 부동 소수점 계산에 매우 효율적인 16 비트 지수(exponent)와 64비트 가수(mantissa) 레지스터를 가지고 있으며, CLIP7은 16 비트 ALU와 1비트 근접통신 링크를 제공하고 있다. BLITZEN, CAAPP, MGAP-2는 bit-serial 통신과 계산을 수행하며, 이중 BLITZEN은 입출력을 4비트 parallel로 수행한다. IMAP은 근접통신 시간부담을 줄이기 위해 bit-parallel 통신과 계산을 사용하며, VIP는 bit-parallel 통신과 bit-serial 계산을 사용한다.

3. 분석적 모델과 알고리즘 복잡성

이 절에서는 SIMD 어레이 프로세서의 분석적 모델과 알고리즘 복잡성(algorithm complexity)에 대해 논의한다. 전체적인 처리시간 T_c 는 크게 데이터 입출력 시간 T_{io} , 계산시간

T_{CP} 및 근접통신 시간 T_{pp} 의 세 가지 요소로 구성된다. 분석을 간단하게 하기 위해서 격자 연결 어레이 프로세서의 크기가 입력되는 영상 데이터의 크기와 같거나 크다고 가정한다. 전체 영상을 처리하는 총 시간은 다음의 식 (1)과 같이 표현할 수 있다.

$$T_A = T_{IO} + T_{CP} + T_{pp} \quad (1)$$

데이터의 입출력이 동시에 일어난다면 입출력 시간은 식 (2)로 표시된다. n_o 는 영상 프레임의 열(column) 수를 나타내고 t_o 는 한 열에 대한 입출력 사이클 시간을 나타낸다. 일반적으로 데이터 입출력은 동시에 일어난다.

$$T_{IO} = n_o * t_o \quad (2)$$

계산에 필요한 시간은 식 (3)과 같이 표시된다. 여기서, n_i 는 지정된 알고리즘을 실행하기 위한 명령어의 개수이고, t_i 는 프로세서의 명령어 사이클 시간이다. Bit-parallel 시스템의 n_i 는 bit-serial 시스템의 n_i 보다 사용하는 데이터 경로 비트 수로 나눈 만큼 작다.

$$T_{CP} = n_i * t_i \quad (3)$$

근접통신 시간은 식 (4)로 표현된다. n_s 는 특정 알고리즘에서 이웃한 PE들로 전달되는 비트 수이며, t_s 는 이웃한 PE 사이에서 1비트 통신에 대한 소요 시간이다. Bit-parallel 시스템에서도 통신 링크는 1비트이므로 바이트의 개수 대신 전송될 비트 수를 고려해야 한다.

$$T_{pp} = n_s * t_s \quad (4)$$

그러므로 전체 영상에 대한 총 처리시간은 식 (5)와 같이 나타낼 수 있으며, 어레이 프로세서가 입출력에 대해 버퍼링(buffering) 능력을 가지고 있다면 입출력은 계산과 중첩될 수 있다. 이 경우에 총 처리시간은 식 (6)과 같이 줄어든다.

$$\begin{aligned} T_A &= n_o * t_o + n_i * t_i + n_s * t_s \\ &= T_{IO} + T_{CP} + T_{pp} \end{aligned} \quad (5)$$

$$\begin{aligned} T_A &= T_{CP} + T_{pp} \quad \text{if } T_{CP} + T_{pp} \geq T_{IO} \\ T_A &= T_{IO} \quad \text{otherwise} \end{aligned} \quad (6)$$

또한 시스템이 계산을 하는 동안 근접통신이 일어날 수 있다면 통신은 계산과 중첩될 수 있다. 그러나 어떤 알고리즘에 대한 통신은 완전히 중첩될 수 없다. 예를 들어 T_{pp} 가 T_{CP} 보다 크면 T_{pp} 는 일부는 여전히 존재한다. 이로 인해 총 처리시간은 더욱 감소한다. 총 처리시간 T_A 는 다음과 같이 표현된다.

$$\begin{aligned} T_A &= T_{CP} + \rho T_{pp}, \quad \text{if } T_{CP} + \rho T_{pp} > \\ &\quad (1 - \rho) T_{pp} \quad \& \quad T_{CP} + \rho T_{pp} \geq T_{IO} \\ T_A &= (1 - \rho) T_{pp}, \quad \text{if } (1 - \rho) T_{pp} > \\ &\quad T_{CP} + \rho T_{pp} \quad \& \quad (1 - \rho) T_{pp} \geq T_{IO} \\ T_A &= T_{IO}, \quad \text{otherwise} \end{aligned} \quad (7)$$

ρ 는 T_{CP} 와 T_{pp} 의 중첩되지 않은 부분이다. ρ 가 0 즉, T_{pp} 가 T_{CP} 보다 작아서 완전히 중첩될 경우 T_A 는 순수한 계산시간이 된다.

MPP와 BLITZEN과 같은 격자연결 SIMD 어레이 프로세서의 분석적 모델은 식 (5)로 나타낼 수 있으며, CAAPP는 T_{pp} 를 $(1 - \rho) T_{pp}$ 로 바꾼 식 (5)로 나타낼 수 있다. MasPar, CLIP7, DAP, IMAP, VIP, Slim과 같이 시스템이 데이터 입출력과 계산이 중첩된다면 시스템의 총 처리시간은 식 (6)으로 나타낼 수 있으며, CLIP7, DAP, VIP, Slim과 같이 시스템이 근접통신의 중첩 특성을 갖는다면 시스템의 분석적인 모델은 식 (7)로 나타낼 수 있다. 대다수 SIMD 구조의 계산시간은 입출력 시간이나 근접통신 시간보다 크기 때문에 총 처리시간은 순수한 계산시간만으로 구성된다.

위에서 설명한 분석적 모델이 아닌 알고리즘 복잡성을 이용하여 어레이 프로세서의 성능을 비교하면 다음과 같다. 격자연결 아키텍처들의 알고리즘 복잡성은 통신과 계산의 복잡성을 포함한 전체 시간 복잡성에 근거를 둔다. 그러나, Fang[35]은 통신 복잡성과 계산 복잡성을 분리하여 다루었다. 단위시간내에 각 PE는 이웃한 PE로부터 1개의 워드 데이터나 연산 결과 또는 논리연산 결과를 송수신할 수 있다. 즉, 전체 알고리즘 복잡성 $O(n)$ 는 알고리즘이 근접통신을 위해 필요한 $C_1 * n$ 과 산술, 논리연산을 위해 필요한 $C_2 * n$ 으로 나타낼 수 있다. 이때 C_1 과 C_2 는 양의 상수이다.

위에서 살펴본 바와 같이, Fang[35]은 격자

연결된 아키텍처의 일반적인 2차원 콘벌루션 알고리즘은 $O(W^2)$ 의 통신 복잡성을 갖는다고 기술하였다. DAP, VIP, SLIM 등과 같이 시스템이 근접통신의 중첩 특성을 갖는다면 근접통신 시간부담이 완전히 상쇄되므로 2차원 콘벌루션 알고리즘은 $O(0)$ 의 통신 복잡성을 갖는다. 그리고, 많은 병렬 영상처리 알고리즘은 $O(0)$ 또는 $O(1)$ 의 통신 복잡성을 갖는다[24]. 일련의 영상처리 과정은 동일한 영상 프레임에 대해 순차적으로 수행되므로 입출력 시간은 전체 계산시간보다 작다고 가정할 수 있다. 시스템이 입출력 중첩 특성을 갖는다면 입출력 시간은 상쇄되므로, 입출력 복잡성은 $O(0)$ 이다.

4. 결 론

본 논문은 최근 SIMD 어레이 프로세서들의 아키텍처 구조에 대해 조사 분석하였고, SIMD 어레이 프로세서의 분석적 모델과 알고리즘 복잡성에 대해서 논의하였다. 아키텍처들의 새로운 특징들을 간략히 요약하면 다음과 같다. BLITZEN, MasPar는 X-Net를 가지고 있어 대각선 통신에 효율적이며, SLIM은 by-passing 경로를 이용하여 4개의 링크만으로 8방향의 링크를 구성할 수 있으며, 보조 프로세서 또는 칩 내의 DMA 컨트롤러 없이 근접통신을 중첩시킬 수 있는 S 평면을 가진다. MasPar는 지수와 가수를 위한 2개의 레지스터를 갖고 있어 4 비트 ALU와 함께 부동 소수점 연산을 수행하는데 효과적이며, CLIP7A는 주소 자치권을 부여하기 위한 보조 연산장치를 갖는다. DAP의 경우에는 계산만을 담당하는 8 비트 보조 프로세서와 통신과 계산을 담당하는 1 비트 프로세서로 구성된다. CAAPP는 연결된 영역 레이블링 알고리즘을 효율적으로 구현하기 위한 연결 자치권을 제공하는 Coterie 연결망을 갖는다. MGAP-2는 데이터 워드 길이에 따라 재구성할 수 있는 연결구조를 제공하며 VIP는 선형연결 구조로써 근접통신을 중첩시킨다.

새로운 어레이 프로세서들은 자치권을 제공하여 모든 PE가 같은 명령어를 수행하는 SIMD를 MIMD처럼 사용하여, 각각의 PE가

서로 다른 명령어를 수행할 수 있다. 또한 어레이 프로세서는 대규모 병렬처리를 수행할 수 있으며 소프트웨어와 하드웨어의 설계가 비교적 용이하며 시스템 가격이 상대적으로 저렴하다. 그리고 VLSI 기술의 발달로 한 개의 칩에 많은 수의 PE를 집적시키는 것이 가능하게 되었고, 가격 대비 성능이 우수하여 실시간 레이다 신호처리, 영상신호의 움직임 추정, 부품검색, 컴퓨터 시각인식, 공장자동화, 데이터 매이스, 반도체 회로분석, 유체역학, 가상현실, 컴퓨터 애니메이션, 거래처리, 정찰비행영상, 위성영상처리 등과 같이 어레이 프로세서의 응용 분야가 점차 확대되고 있다.

참고문헌

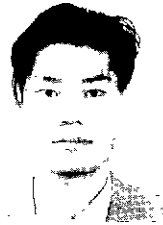
- [1] Special Issue on Computer Architecture for Image Processing, IEEE Computer, Jan. 1983.
- [2] S. H. Unger, "A computer oriented toward spatial problems," Proc. IRE, vol. 46, pp. 1744-1750. Oct. 1958.
- [3] F. A. Gerritsen, "A comparison of the CLIP4, DAP and MPP processor-array implementations," Computing Structures for Image Processing, M. J. B. Duff, Ed., Academic Press, 1983, pp. 15-30.
- [4] T. J. Fountain, "A survey of bit-serial array processor circuits," Computing Structures for Image Processing, M. J. B. Duff, Ed., Academic Press, 1983, pp. 1-13.
- [5] M. Maresca, M. A. Lavin, and H. Li, "Parallel architectures for vision," Proceedings of IEEE, vol. 76, Aug. 1988.
- [6] J. Gregory and R. McReynolds, "The SOL-OMON computer," IEEE Trans. Electronic Comput., vol. EC-12, pp. 774-780, Dec. 1963.
- [7] G. H. Barnes, R. M. Brown, M. Kato, D. J. Kuck, D. L. Slotnick, and R. A. Stokes, "The ILLIAC IV computer," IEEE Trans. Comput. vol. C-17, pp. 746-757, Aug. 1968.
- [8] K. E. Batchor, "Design of a massively par-

- allel processors," *IEEE Trans. Comput.* vol. C-29, pp. 836-840, Sep. 1980.
- [9] J. L. Potter, Ed., *The Massively Parallel Processor*, The MIT Press, 1985.
- [10] T. J. Fountain, "Plans for the CLIP 7 chip," *Integrated Technology for Parallel Image Processing*, S. Levialdi, Ed., Academic Press, 1985, pp. 199-214.
- [11] T. J. Fountain, K. N. Matthews, and M. J. B. Duff, "The CLIP7A image processor," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 310-319, May 1988.
- [12] DAP Series Technical Overview, Active Memory Technology Inc., 1989.
- [13] T. M. Silberberg, "The Hough transform on the geometric arithmetic parallel processor," in *Proc. Comput. Architect. for Pattern Anal. Machine Intell.*, 1985, pp. 387-393.
- [14] L. N. Robinson and W. R. Moore, "A parallel processor array architecture and its implementation in silicon," in *Proc. IEEE Custom Integrated Circuits Conf.*, Rochester, NY, May 1982, pp. 41-45.
- [15] T. Sudo and T. Nakashima, "An LSI adaptive array processor," in *Proc. IEEE Int. Solid-States Circuits Conf.*, San Francisco, Feb. 1982, pp. 122-123, 307.
- [16] A. P. Reeves, "A systematically designed binary array processor," *IEEE Trans. Comput.* vol. C-29, pp. 278-287, April 1980.
- [17] C. Weems, "Some sample algorithms for the image understanding architecture," in *Proc. DARPA Image Understanding Workshop*, Washington D.C., 1988, pp. 127-138.
- [18] H. Li and M. Maresca, "Polymorphictorus: a new architecture for vision computation," in *Proc. IEEECS Comput. Architect. for Pattern Anal. Machine Intell.*, Seattle, WA, Oct. 1987, pp. 176-183.
- [19] R. Miller, V. K. P. Kumar, D. Reisis, and Q. F. Stout, "Meshes with reconfigurable buses," in *Proc. MIT Conf. Advanced Research in VLSI*, 1988, pp. 163-178.
- [20] W. D. Hillis, *The Connection Machine*, MIT Press, 1985.
- [21] J. R. Nickolis, "The design of the MasPar MP-1: a cost effective massively parallel computer," in *Proc. IEEE Compeon Spring 90*, 1990, pp. 25-28.
- [22] E. W. Davis and J. H. Reif, "Architecture and operation of the BLITZEN processing element," in *Proc. 3rd Int. Conf. Supercomputing*, vol. III, 1988, pp. 128-137.
- [23] Fan Zhou et al., "Introducing MGAP-2," in *Proc. 5th Int. Symp. The Frontiers of Massively Parallel Computation*, Feb. 1995, pp. 281-288.
- [24] M. H. Sunwoo and J. K. Aggarwal, "A sliding memory plane array processor," *IEEE Trans. on Parallel and Distributed Systems*, vol. 4, pp. 601-612, June 1993.
- [25] M. H. Sunwoo, S. Ong, B. Ahn, and K. Lee, "Design and Implementation of a Parallel Image Processor Chip for a SIMD Array Processor," in *Proc. Int. Conf. Application Specific Array Processors*, Strasbourg, France, July 1995, pp. 66-75.
- [26] Hyun M. Chang, Myung H. Sunwoo and Tai-Hoon Cho, "Implementation of a ShM Array Processor," in *Proc. 10th International Parallel Processing Symposium*, Hawaii, April 1996, pp. 771-775.
- [27] Y. Fujita et al., "A Real-Time Vision System Using Integrated Memory Array Processor Prototype LSI," in *Proc. '92 IAPR Workshop on Machine Vision Applicat.*, 1992, pp. 47-50.
- [28] S. Okazaki et al., "A Compact Real-Time Vision System Using Integrated Memory Array Processor Architecture," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 5, no. 5, pp. 446-452, Oct. 1995.
- [29] Michael Hall, "A Study of Parallel Image Processing on the VIP Architecture," PhD thesis, LIU-TEK-LIC-1995:41, Depart-

ment of Electrical Engineering, Link ping, Sweden, October 1995.

- [30] L. A. Schmitt and S. S. Wilson, "The AIS-5000 Parallel Processor," IEEE Trans. PAMI, vol. 10, no. 3, pp. 320-330, May 1988.
- [31] A. L. Fisher and P. T. Highnam, "The SLAP Image Computer," Parallel Architectures and Algorithms for Image Understanding, V. K. P. Kumar, Ed., Academic Press, 1991, pp. 307-337.
- [32] T. Kushner, A. Y. Wu, and A. Rosenfeld, "Image processing on MPP," Pattern Recognition, vol. 15, pp. 121-130, 1982.
- [33] T. Ericsson and P-E. Danielsson, "LIPP - a SIMD multiprocessor architecture for image processing," in Proc. 10th Annual Int. Symp. Comput. Architect., 1983, pp. 395-400.
- [34] J. P. Strong, "The Fourier transform on mesh connected processing array such as the massively parallel processor," in Proc. IEEECS Comput. Architect. for Pattern Anal. Machine Intell., 1985, pp. 190-196.
- [35] Z. Fang, X. Li, and L. M. Ni, "On the communication complexity of generalized 2-D convolution on array processors," IEEE Trans. Comput., vol. 38, No. 2, pp. 184-194, Feb. 1989.

장 현 만



1995.2 이주대학교 전자공학 학사
 1995.3~현재 이주대학교 전기전자공학부 석사과정
 관심분야: Parallel Architecture 및 ASIC 설계

옹 수 환



1994.2 이주대학교 전자공학 학사
 1996.2 이주대학교 전자공학 석사
 1996.3~현재 이주대학교 전기전자공학부 박사과정
 관심분야: 통신 및 신호처리용 ASIC 설계

선우 명훈



1980.2 서강대학교 전자공학 학사
 1982.2 한국과학기술원 전자공학 석사
 1982.3~85.8 한국전자통신연구소(ETRI) 연구원
 1985.9~90.8 Univ. of Texas at Austin 전자공학 박사
 1990.8 ~ 92.8 Motorola. DSP Chip Division, Member of Technical Staff
 1992.8~현재 이주대학교 전기전자공학부 조교수
 관심분야: Parallel Architecture, VLSI Architectures, 통신 및 신호처리용 ASIC 설계

● APSEC '96 논문모집 ●

- 일 시 : 1996년 12월 4~7일
- 장 소 : 교육문화회관
- 주 최 : 소프트웨어공학연구회
- 문 의 처 : 포항공과대학교 강교철 교수
 T. 0562-279-2258
 F. 0562-279-2299
 E-mail: kck@wision.postech.ac.kr