論文 94-19-10-19

통신 부담을 감소시킨 영상처리를 위한 병렬처리 방식 ASIC 구조 설계

正會員 安 炳 德* 正會員 鄭 智 元* 正會員 鮮 于 明 勳*1)

Design of an Image Processing ASIC Architecture using Parallel Approach with Zero or Little Communication Overhead

Byung Dug Ahn* Ji Won Jung* Myung Hoon Sunwoo*

要 約

본 논문에서는 근접한 Processing Element (PE)들간의 통신 부담을 경감시켜 영상신호를 실시간 처리할 수 있는 새로운 병렬처리 방식 ASIC 구조를 설계한다. 하나의 Sliding Memory Plane (SliM) Image Processor chip을 병렬처리 방식을 사용 3×3 PE를 격자 형태로 연결한다. 제안하는 Image Processor들을 다수 격자 형태로 연결하면 새로운 방식의 SIMD 병렬 컴퓨터인 SliM Array Processor를 구현할 수 있다. Sliding 개념은 별도의 보조 프로세서나 DMA를 사용치 않고 또한 PE들을 interrupt 걸지 않고 모든 화소가 이웃 PE로 이동됨을 의미한다. 따라서 근접 통신과 계산이 동시에 일어나 기존의 격자 연결 병렬 컴퓨터의 결정적 단점인 근접 통신 부담을 경감시킬 수 있다. 또한 하나의 PE에 두개의 입출력용 레지스터 plane을 사용, buffer를 제공하여 입출력 부담을 감소시킨다. SliM Image Processor에서는 단지 4개의 통신 link만으로 8가지 방향의 통신경로를 제공하는 by-passing path에 의해 통신 부담없이 대각선 통신을 수행할 수 있다. 제안하는 유일한 특성들로 인해 영상신호 처리시 성능을 향상시킬 수 있다. 영상신호 처리를 위한 알고리즘들을 효율적으로 수행키 위한 PE, Image Processor 구조 및 명령어를 설계한다.

Abstract

This paper proposes a new parallel ASIC architecture for real-time image processing to reduce inter-processing element (inter-PE) communication overhead, called a Sliding Memory Plane (SliM) Image Processor. The SliM Image Processor consists of 3×3 processing elements (PEs) connected by a mesh topology. With easy scalability due to the topology, a set of SliM Image Processors can form a mesh-connected SIMD parallel architecture, called the SliM Array Processor. The idea of sliding means that all pixels are slided into all neighboring PEs without interrupting PEs and without using a coprocessor or a DMA controller. Since the inter-PE communication and computation occur simultaneously, the inter-PE communication overhead, significant disadvantage of existing machines greatly diminishes. Two I/O planes provide a buffering capability and reduce the data I/O overhead. In addition, using the by-passing path

1* 亞洲大學校 電子工學科
Dept. of Electronics Engineering, Ajou University.
論文番號 : 94205
接受日字 : 1994년 7월 29일

provides eight-way connectivity even with four links. With these salient features, SliM shows a significant performance improvement. This paper presents architectures of a PE and the SliM Image Processor, and describes the design of an instruction set.

I. 서 론

Bell Lab의 Unger가 이차원적 구조를 갖는 데이타의 고속처리를 위하여 격자연결방식(mesh topology)의 병렬 컴퓨터를 최초로 제안한 이후[1], 실시간 신호 및 영상처리와 같이 막대한 양의 데이타를 처리하기에 적합한 많은 SIMD(single instruction multiple date stream) 병렬 컴퓨터들이 제안되어졌고[1][2][3] 실제 공학 및 과학 분야에서 광범위하게 사용되고 있다[4][5][6]. 그러나 기존의 SIMD 병렬 컴퓨터들은 몇 가지 단점들을 가지고 있다. 이웃한 PE끼리의 통신은 근접 통신(inter-processing element, inter-PE 또는 local communication) 부담, 데이타 입출력(I/O) 부담, 각 PE들이 서로 다른 독립적인 기능을 수행할 수 있는 자치권(autonomy) 부재등은 신호 및 영상처리 알고리즘들의 수행에 있어 제한을 받는다. 특히 계산 중에 이웃 PE들간에 있어 데이타 교환을 위한 많은 양의 근접 통신이 요구되는데 이 근접 통신 부담은 기존의 격자 연결 방식 SIMD 컴퓨터들의 심각한 단점이다[3][6][7][8][9]. 또한 SIMD 병렬 컴퓨터의 일부는 한 PE당 6개 또는 8개의 통신 link를 갖고 있어 상대적으로 복잡한 신호 연결망(interconnection network)들을 갖는다. 이와 같은 단점들을 극복하고 성능 향상을 위해 SliM Array Processor의 개념이 제안되었다[10][11][12]. 본 논문에서는 이의 구현을 위해 영상신호 처리 알고리즘들을 효율적으로 수행키 위한 새로운 PE 구조 및 명령어를 설계하고, 3×3 PE를 격자형태로 연결한 SliM Image Processor의 ASIC 구조를 제시한다.

SliM에서 제안한 "sliding" 착상은 PE들을 interrupt 걸지 않고 PE간의 근접 통신과 계산을 중첩시킬 수 있어 근접 통신 부담이 마치 사라지게되는 개념이다. 이를 위해 기존 컴퓨터에서 사용되는 chip내의 DMA (Direct Memory Access) 구조나 보조 프로세서를 필요로 하지 않고 윈도우 연산자의 크기와 형태에 상관없이 "sliding"개념은 근접 통신을 중첩시킬 수 있는데 이는 근접 통신 부담을 상당부분 줄일 수 있다. 또한 PE를 interrupt 걸지않고 입출력을 계산과 중첩시키기 위하여 두개의 입출력 레지스터 plane을 사용한다. Fang은 그의 논문에서[9] 일반적인 격자연결 SIMD 어레이 프로세서 상에서 2차원 콘볼루션 알고리즘을 수행하기 위해 요구되어지는 근접 통신의 수는 $O(W^2)$의 통신 복잡성(communication complexity)을 갖는다고 주장하였다. 여기에서 윈도우 연산자(window operator)의 크기는 W×W이다. 즉, 윈도우 연산자의 크기가 커질수록 제곱에 비례한 통신 복잡성을 갖는다. 또한 실시간 응용에 있어 데이타 입출력 시간은 전체적인 성능에 병목현상을 초래할 수 있기 때문에 실시간 시스템 설계에 있어 필히 고려해야 할 중요한 문제이다.

또한 SliM에서는 명령어 사이클 부담없이 8가지 방향의 연결을 위해서 독특한 by-passing 개념을 제안함으로써 8개의 통신 link가 아닌 4개의 link만을 사용하여 대각선 통신도 부담없이 수행될 수 있다. 이로써 대각선 통신 시간이 크게 감소하며 실질적인 대각선 link가 필요 없어져 상대적으로 간단한 상호 연결망을 갖는다. 대각선 통신을 위한 또 하나의 방법은 sliding memory plane을 두번 shift시켜서 성취할 수 있는데 이 또한 PE의 연산 동작과 중첩될 수 있다. SliM에서는 3가지 형태의 자치권, 즉 operation, addressing 및 connection autonomy를 모두 제공하며 각 PE들은 이들을 사용 부분적으로 MIMD(multiple instruction multiple data stream) 병렬 컴퓨터와 같이 독립적으로 동작할 수 있다. SliM은 bit-serial 통신과 bit parallel 계산방식을 채택하여 VLSI 면적 및 입출력 핀수를 줄이고 속도를 증가시킬 수 있다.

SliM Image Processor의 명령어 설계시 영상처리 알고리즘들에[13] 필요한 연산들을 분석하여 이들이 효율적으로 수행될 수 있도록 하였고 또한 기존의

MPP[14],[15], GAPP[16] 및 CLIP7A[17]에서 사용하는 명령어들의 대부분을 채택하며 SliM Image Processor가 영상처리용 병렬 컴퓨터에 효과적으로 사용될 수 있다. SliM Array Processor의 시뮬레이션 모델은 미 국방연구기관(DARPA)에서 정한 low-level 영상 이해 benchmark[18]들에 대해서 다른 병렬처리 컴퓨터들보다 우수한 성능을 입증하였다[10],[11].

본 논문은 다음과 같이 구성된다. Ⅱ절에서는 SliM Image Processor의 새로운 PE구조를 제시하고, Ⅲ절에서는 격자연결 SliM Image Processor 구조를 설명한다. 영상처리 알고리즘들의 연산에 효율적인 명령어 설계를 Ⅳ절에서 설명하고, Ⅴ절에서 SliM에서의 영상처리 알고리즘에 대해 설명한다. 마지막으로 Ⅵ절에서는 결론을 기술한다.

Ⅱ SliM PE의 Architecture

전체적인 SliM Array Processor는 N×N개의 PE들로 구성된 프로세서 plane P, N×N개의 shift register들이 격자 형태로 연결된 sliding memory plane S, 입출력 전용으로 이용되는 D와 D′ plane, 데이터의 입출력을 제어하는 입출력 전용 프로세서(IOP) 및 프로세서 plane과 S plane을 제어하는 Control Unit(CU)으로 구성되어 있다. 전체적인 SliM 병렬 컴퓨터의 구조는 다른 논문에서[10],[11],[12] 상세히 기술하였고 이 논문에서는 새로이 설계한 SliM Image Processor ASIC에 대해 기술한다.

1. PE Architecture

그림 1에서 보인 PE는 8-bit ALU, 레지스터(registers), 멀티플렉서(MUXs), 디멀티플렉서(DMUX) 그리고 4×1 스위치 소자(SW)들로 구성된다. shift 레지스터 s는 Sliding memory plane S의 한 소자이고 d와 d′은 입출력 plane D와 D′의 각 소자들이다. s 레지스터는 스위치 소자 SW를 통하여 4개의 이웃 PE들에 있는 레지스터들과 연결되어 있으므로 이를 통하여 근접 PE들이 연결된다. d와 d′ 레지스터는 좌우측 PE들의 d와 d′ 레지스터에만 연결되어 있는데 이러한 방법은 PE들을 interrupt걸지 않고 s 레지스터를 통한 근접 PE간의 통신과 데이타의 입출력을 제공할 수 있는데 그 설명은 다음과 같다.

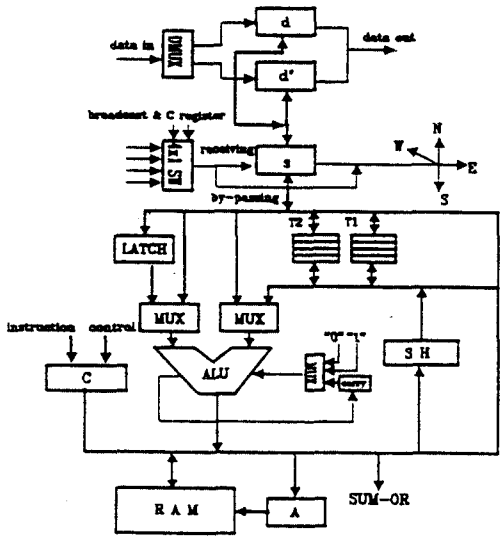


그림 1. 한 PE 구조
Fig. 1. The Architecture of a PE

ALU가 s 레지스터에 있는 화소를 처리하는 동안 그 화소는 이웃 s 레지스터로 이동되고 동시에 새로운 이웃 화소가 s 레지스터로 들어온다. ALU가 s 레지스터로 이동된 새로운 화소를 처리하는 동안 그 화소는 또 다른 이웃 레지스터로 이동되고 동시에 이 s 레지스터에는 또 다른 이웃 화소가 들어온다. 모든 PE들이 이러한 동작을 동시에 실행할 수 있기 때문에 모든 화소들은 동시에 동일한 방향의 이웃 s 레지스터로 이동된다. 각각의 동작들은 프로세서 plane 및 Sliding memory plane을 제어하는 CU와 입출력을 제어하는 IOP에 의해 독립적으로 제어될 수 있기 때문에 계산, 근접 통신 그리고 입출력이 동시에 일어날 수 있고 근접 PE간의 통신과 입출력 부담은 계산시간과 중첩되어 크게 줄어든다.

PE를 이루고 있는 다른 주요한 소자들은 가변길이 shift register(SH), condition register(C), 각 PE내의 기억소자 주소를 독립적으로 제공하기 위한 address register(A), 8 개의 범용 8-bit 레지스터(Ts) 그리고 PE의 내부에 위치해 데이타를 저장할 수 있는 적은 양의 기억소자(RAM)가 있다. SliM의 ALU에서 제공하는 함수들은 SliM의 독특한 특성에 따라 영상처리 알고리

즘들을 수행할 수 있도록 효율적으로 설계된 명령어들을 모두 지원할 수 있으며 다른 SIMD 병렬 컴퓨터들의 ALU보다 간단하고 함축적이다. 이는 VLSI 면적을 줄여 다수의 PE가 하나의 chip에 집적될 수 있도록 하기 위함이다. 또한 모든 데이타들은 이진보수를 사용한다. Table 1은 ALU에서 제공되는 함수들을 도시한 표이다.

표 1. ALU 함수
Table 1. Functions of ALU

Selection	M=H:Logic operation	M=L:Arithmetic operation	
S1 S0		Cn=L(no carry)	Cn=H(with carry)
0 0	F=$\bar{A}$	F=A	F=A PLUS 1
0 1	F=A·B	F=A MINUS 1	F=A
1 0	F=A+B	F=A PLUS B	F=A PLUS B PLUS 1
1 1	F=A⊕B	F=A MINUS B MINUS 1	F=A MINUS B

4×1 스위치 소자 SW는 이웃하는 PE(동서남북)들 사이에 bit-serial 통신을 위한 상호 연결망을 제공한다. SW의 출력은 이웃화소 중의 하나가 s 레지스터로 입력되는 receiving mode, s 레지스터로 입력되지 않고 by-passing path를 거쳐 또다른 이웃 PE 중의 하나로 전송되는 by-passing mode, s 레지스터로 입력되는 동시에 다른 이웃 PE들로 전송되는 receiving/by-passing mode를 동시에 수행할 수 있다[10,11,12]. SW의 선택 단자는 보통 CU에서 broadcast되는 명령어에 의해 제어되나 각 PE들에서 독립적인 연결을 필요로 할 경우에는 C 레지스터에서 제어를 할 수 있도록 설계하여 재구성(reconfigurability)이 가능하도록 하였고 따라서 4×1 SW는 각 PE마다 상호연결 상태가 다른 connection autonomy를 가능케 한다.

네개의 이웃 PE들은 s 레지스터를 통하여 bit serial로 연결되어 있다. d 또는 d'레지스터와 latch로의 연결은 bit-parallel로 연결하여 한 clock 사이클 안에 데이타 이동을 할 수 있도록 설계하였다. receiving mode로 설정되어 있는 PE들은 이웃 PE에서 sliding되는 데이타를 s 레지스터에 저장하며 이웃 PE들에서 다시 sliding되는 데이타와의 충돌을 방지하기 위해 latch로 데이타를 복사한다. s 레지스터의 출력은 이웃한 4개의 PE들로 sliding하여 근접통신을 수행한다.

d와 d' 레지스터는 실질적으로 D와 D' plane을 구성하는 하나의 소자들이다. DMUX에서 선택된 데이타가 d(또는 d') 레지스터로 bit-serial 방식으로 이동된다. 데이타 입출력 동작은 row-parallel 방식으로 수행되며 만일 화소가 8 bit이라면 8번의 shifting을 거쳐서 동일한 행상에 있는 d 레지스터를 채우게 된다. 다음번 화소의 LSB가 입력되면 이미 이 레지스터에 저장되어 있던 화소의 LSB는 이웃 PE의 d 또는 d'레지스터로 이동된다. 이러한 과정을 N×8번 반복하면 N×N shift 레지스터들로 이루어진 D 또는 D' plane은 하나의 영상 frame을 갖게 된다. 전단계에서 처리된 화소는 s 레지스터에서 d'(또는 d) 레지스터로 bit parallel 방식으로 shift된다. 이 데이타의 출력은 새로운 영상 frame의 화소들이 d'(또는 d) 레지스터로 입력됨과 동시에 이루어 진다. 이와 같이 두 개의 d와 d' 레지스터들은 buffering 능력을 제공하여 입출력 시간을 계산시간과 중첩시켜 데이타 입출력 부담을 줄일 수 있다.

T 레지스터는 8개의 8 bit 범용 레지스터로서 ALU에서 계산시 요구하는 데이타 및 계산후의 중간 결과값을 저장한다. 8 bit 곱셈 연산 수행시 SH 레지스터의 값을 받아서 일시 저장하며 s 레지스터로 입력되는 데이타를 저장하기도 한다. 이 레지스터에 저장된 데이타 값들은 s 레지스터 또는 ALU의 입력으로 인가된다. 32 bit를 갖는 SH 레지스터는 필요에 따라 가변 길이로 shift를 수행하며 논리 shift, 연산 shift, 부동 소수점의 계산, 곱셈 및 나눗셈 연산에서 효율적으로 사용된다.

CU로부터 broadcast되는 명령어에 따라 각 PE에 있는 condition 레지스터 C의 상태가 동시에 변화되어 모든 PE가 같은 동작을 수행할 수 있는 중앙 제어(centralized control) 방식이 가능하며 또한 각 PE가 데이타와 알고리즘에 따라 C 레지스터의 상태를 부분적으로 변화시켜 각 PE가 C 레지스터에서 나오는 제어 신호에 따라 독립적인 동작을 수행할 수 있는 분산 제어(distributed control) 방식도 수행할 수 있어 operation autonomy를 구현할 수 있는데 이에 대한 기술은 다음절에서 자세히 기술하기로 한다.

이미 언급한 바와 같이 데이타의 입출력 및 sliding 동작을 위한 통신 link는 bit-serial 방식을 채택하여 VLSI 면적 및 입출력 핀 수를 감소시키고 PE 내부의

계산에서는 데이타의 이동 및 연산을 bit-parallel 방식으로 보다 빠른 계산을 수행할 수 있게 한다. 그림 1에서 굵은 선은 bit-parallel datapath이며 얇은 선은 bit-serial datapath를 나타낸다.

2. 자치권(Autonomies)

기존의 SIMD 병렬 컴퓨터들과는[3] 달리 SliM에서는 3가지 종류의 자치권, 즉 operation autonomy, addressing autonomy, connection autonomy를 모두 제공함으로써 SIMD 병렬 컴퓨터의 단점인 자치권 부재를 해결할 수 있는데 이는 MIMD 컴퓨터와 같이 PE들이 부분적으로 독립적인 기능을 수행할 수 있어 성능을 향상시킬 수 있으며 그 내용은 다음과 같다.

1) Operation autonomy : C 레지스터에 activity bit flag를 두어 각 PE가 이 flag에 따라서 서로 다른 동작을 수행할 수 있게 한다. activity bit는 CU로부터 broadcast된 명령어에서 제어할 수 있고 또한 데이타의 결과 값이나 알고리즘에 따라 이 flag를 set/reset할 수 있다. activity bit flat가 set/reset된 상태에 따라 각 PE들이 서로 다른 연산을 수행한다. 이는 고급 언어에서의 if (condition) then (statements) else (statements) 문을 구현할 수 있다.

2) Addressing autonomy : 주소 레지스터 A를 두어 기억소자 주소를 각 PE마다 독립적으로 제공할 수 있다. C 레지스터에 할당된 1-bit flag에 따라서 다음에 수행되는 명령어에 대한 기억소자 주소가 A 레지스터에 있는지 broadcast된 명령어에 있는지를 판별해서 addressing autonomy를 구현한다.

3) Connection autonomy : C 레지스터에 할당된 2-bit flag에 따라서 각 PE에 있는 4×1 SW의 입력이 결정되면 각 PE들이 이웃 PE들과 독립적인 연결이 성립된다. 또한 by-passing path를 따라 어떠한 PE와도 연결이 가능하므로 상호통신망 구조를 재구성(reconfigurability)할 수 있는 능력을 제공한다.

III. SliM Image Processor 구조

그림 2는 한 chip에서 3×3 PE들로 연결된 SliM Image Processor의 병렬 VLSI구조를 나타낸다. 9개의 PE들은 확장성을 갖는 격자 형태로 연결되어 있다. 즉 격자 형태의 연결은 수천, 수만 개의 PE들을 연결시킬 수 있어 SliM Array Processor의 구현이 매우 용이하다. 앞에서 언급한 바와 같이, 4개의 근접 PE들간의 연결은 각 PE내에 있는 s 레지스터에 의해 bit-serial 방식으로 이루어진다. 그러므로 chip들 사이의 근접 통신에 필요한 핀 수는 12개 뿐이며 데이타 입출력을 위해 좌우로 연결된 d 또는 d' 레지스터의 연결을 위한 핀 수는 6개만 필요하다. 전체적으로 9개의 PE들에 대해 근접 통신과 데이타 입출력을 위한 핀 수는 18개로서 chip 상에서 입출력 핀 수 및 chip 면적을 감소시키는 효과를 얻을 수 있다.

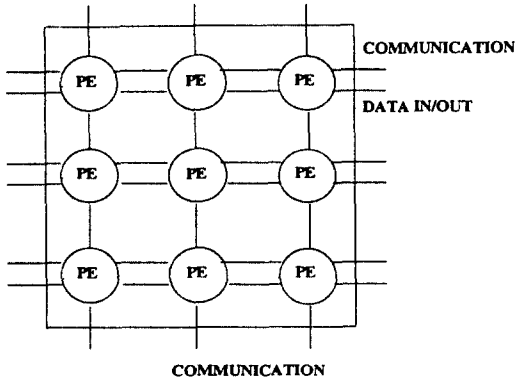


그림 2. SliM Image processor의 병렬 VLSI 구조
Fig. 2. A Parallel VLSI Architecture of the SliM Image Processor

앞에서 언급한 세가지 연결 방식을 이용하여 그림 3에서와 같이 대각선 방향의 이웃 PE들간의 가상적인 통신 link를 구현할 수 있다. 예를 들어, 중앙 PE에 대해 서쪽 PE가 by-passing mode로 설정되고, 서쪽 PE에 대해 남서쪽 PE가 receiving mode로 설정된다면 중앙에서 남서쪽 방향으로 가상 link가 구현될 수 있다. 이와 같은 방법으로 다른 모든 대각선 link를 구현할 수 있으나, 동일한 행 또는 열에 있는 2개의 이웃 PE들은 통신 link의 충돌 때문에 동시에 대각선 방향으로 데이타를 보낼 수 없다. 그러나 모든 PE들이 동시에 대각선 통신을 수행하는 알고리즘에서는 sliding memory plane을 전체적으로 두번 shift시켜서 구현할 수 있고 이는 계산시간과 중첩될 수 있기 때문에 통신 부담을 감소시킬 수 있다. 대각선 통신을 위한 가상 link인 by-passing path를 이용 단지 4개의 통신 link만으로 명령어 부담 없이 8가지의 방향의 상호연결 및

통신을 수행할 수 있다. 특히 가상 대각선 link들은 경계화소를 따라 계산하는 영상처리 알고리즘에 매우 유용하게 사용된다.

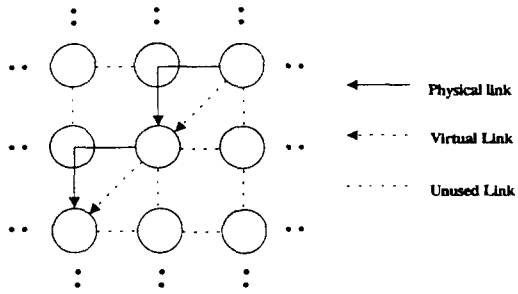


그림 3. 가상 대각선 Link
Fig. 3. Virtual Diagonal Links

IV. 명령어(Instruction Set) 설계

영상처리 알고리즘들을[7] 분석하여 필요한 연산들과 기존의 격자연결 SIMD 병렬컴퓨터들인 MPP[8,9] GAPP[10] 및 CLIP7A[11]의 명령어들을 분석하여 SliM의 명령어를 설계하였다. 하나의 chip에 다수의 PE들을 집적시켜야 하므로 한 PE의 게이트 수를 줄이기 위해 명령어를 간단하고 효율적으로 설계하였다. 설계된 명령어의 microcode는 32 bit width로서 수직 마이크로 명령(vertical microinstruction) 형식이며 sliding 동작은 각 명령어들과 동시에 수행할 수 있고 기억소자, 연산 및 논리 형태로 구분된다. 표 2에 보시한 명령어들은 MUL과 DIV를 제외하고 한 명령어 cycle에 수행될 수 있으며 operation autonomy를 위해 모든 명령어가 C 레지스터의 특정 영역이 'true'이면 동작을 수행하고 'false'이면 주어진 명령어 동작을 수행하지 않는 조건적인 동작이 가능하다. 예를 들면, ADD T1 T2 명령어 수행시 C 레지스터의 조건이 참이면 덧셈을 수행한 결과가 T1 레지스터에 저장되고 거짓이면 T1 레지스터는 현재 값을 유지한다. 곱셈, 나눗셈 및 부동 소수점 연산은 한 chip내에 보다 많은 PE를 집적시키기 위해 곱셈기, 나눗셈기 및 부동 소수점 연산기를 사용치 않고 shift와 연산이 한 clock 사이클에 동시에 수행될 수 있게하여 hardware 부담을 감소시켰으며 이진보수 알고리즘을 분석하여 설계된 명령어 만을 사용하여 구현하였다. 곱셈 연산은 이진보수 곱셈 알고리즘을 사용하여 9 clock 사이클, 이진보수 restoring 나눗셈 알고리즘을 사용한 나눗셈은 40 clock 사이클에 수행 완료된다. 부동 소수점 연산 또한 8 bit bias 128 지수와 16 bit 이진보수 가수 형식으로 294 clock 사이클에 수행된다. 이 명령어들을 본 논문에서 제시한 PE architecture 상에서 다양한 데이타 경로(datapath)를 따라 수행할 수 있다. Table 2는 SliM의 instruction set을 보인 것이다.

표 2. SliM의 명령어
Table 2. The Instruction Set of SliM

Instruction Types	Examples	Description
Memory	MOVE	Move Data
Arithmetic	ADD	Addition
	SUB	Subtraction
	ADC	Add with carry
	SBC	Subtract with carry
	MUL	Multiply
	DIV	Divide
	INC	Increment
	DEC	Decrement
	ASR	Arithmetic Shift Right
	ASL	Arithmetic Shift Left
Logical	AND, OR, XOR	
	NOT	Invert
	LSR	Logical Shift Right
	LSL	Logical Shift Left

다음은 설계된 명령어들이 SliM PE내에서 연산 및 operand에 대한 데이타 이동 경로의 몇가지 예를 나타낸다.

기억소자 명령어(Memory Type Instruction)
MOVE
RAM <-> T (T1, T2), SH, s, A 레지스터
s <-> T (T1, T2), d 또는 d' 레지스터
산술 연산 명령어(Arithmetic Type Instruction)
ADD, ADC, SUB, SBC, MUL, DIV
operand 1 : latch
T (T1, T2) 레지스터

RAM
operand 2 : RAM
SH 레지스터
T (T1, T2) 레지스터
논리 명령어(Logical Type Instruction)
AND, OR, XOR
operand 1 : latch
T (T1, T2) 레지스터
RAM
operand 2 : RAM
SH 레지스터
T (T1, T2) 레지스터

V. SliM 병렬 영상처리 알고리즘

이 절에서는 실시간 영상처리에 주로 사용되는 2-D 콘볼루션 및 Sobel 연산자를 위한 병렬 알고리즘을 SliM에 적합하게 개발하고 이들 알고리즘의 구현시 계산과 통신이 완전히 또는 대부분 중첩됨을 설명한다.

1. 2차원 콘볼루션 알고리즘

그림 4는 2차원 콘볼루션 알고리즘을 RTL(Register Transfer Level) 레벨의 알고리즘을 사용하여 ALGOL과 유사한 언어로[10] 나타낸 것이다. 여기서 s는 Sliding Memory Plane S에 있는 shift 레지스터를 나타내고, Latch는 통신시 데이타의 충돌을 막고 통신부분과 계산부분의 동기 신호를 맞추기 위한 레지스터이며, T는 각 PE에서 중간 결과 값을 저장하기 위한 레지스터이다. 그림 4에서 볼 수 있듯이 W^2개의 곱셈과 $2(W^2-1)$개의 덧셈이 필요하며, 근접통신은 완전히 중첩된다. W는 윈도우 크기를 나타내며 SliM에서는 W에 관계없이 계산과 근접 통신이 완전히 중첩된다. 그림 4에서 각 statement들은 한 줄에 두가지 명령을 기술하였는데, 이는 계산을 수행하는 semi-colon 앞 부분과 Sliding을 담당하는 semi-colon 뒷 부분이 동시에 수행되어 계산과 통신이 완전히 중첩됨을 의미한다. 따라서 이 알고리즘은 $O(W^2)$의 계산 복잡성(computation complexity)과 zero 통신 복잡성을 갖는다.

```
T ← Latch*w00; s ← a neighboring pixel;
                                    /* Sliding */
for i ← 1 until W - 1 do
                                    for j ← 0 until W   1 do
    T ← T + Latch*wij;  s ← a neighboring pixel;
                                    /* Sliding */
```

그림 4. 2차원 병렬 콘볼루션 알고리즘
Fig. 4. The 2-D Convolution Algorithm

만일 윈도우의 크기가 3×3이고 8비트 화소라면 알고리즘의 총 수행시간은 $9n_m+16$ 사이클이 필요하다. 여기서 n_m은 두개의 8비트에 대한 이진 곱셈에 필요한 사이클이다. 기억소자 대신에 T 레지스터가 사용된다면 n_m은 8이 되고 총 88 사이클이 필요하다. 대부분의 SIMD 컴퓨터가 10 MHz 이상으로 동작함으로 SliM의 동작 속도가 10 MHz라고 가정하면 명령어 사이클은 100 ns이다. 이런 가정하에 총 소요시간은 8.8μs이다.

2. Sobel 연산자 알고리즘

Sobel 연산자 알고리즘은 영상 신호의 경계면을 찾기 위하여 사용된다. 그림 5는 Sobel 연산자 알고리즘에 사용되는 윈도우 계수(coefficient)를 나타낸 것이다.

-1	0	1
-2	0	2
-1	0	1

(a) X_magnitude

-1	-2	-1
0	0	0
1	2	1

(b) Y_magnitude

그림 5. Sobel 연산자의 윈도우 계수
Fig. 5. Window coefficients of the Sobel Operator

그림 6은 Sobel 연산자 알고리즘 중 X_magnitude의 계산 알고리즘을 구현한 것이다. 처음에 화소가 s 레지스터에 있다고 가정하자. X_magnitude에 대해 Sliding 동작이 진행될 때 모든 곳에서는 계산과 통신이 중첩되나 중앙과 북쪽 그리고 남쪽을 지날 때는, 계수가 영이므로 계산이 필요치 않다. 따라서 계산 외에 다른 필요한 동작들을 수행한다. 그림 6의 알고리즘중 여섯번째 문장에서는 통신과 계산이 중첩되지 않는다. 그러므로 이 알고리즘은 O(1)의 통신 복잡성과 O(1)의 계산 복잡성을 갖는다. 이와 같이 특별한 경우로서 Sobel 연산자 알고리즘과 같이 중간에 계산이 필요치 않은 알

고리즘들은 통신 부담없이 수행할 수 없다. 그러나 대부분의 알고리즘은 통신 부담없이 수행할 수 있다. 그림 6의 알고리즘을 그림 5의 (b) 윈도우 계수에 대하여 수행하면 Y_magnitude도 얻을 수 있다. 만일 8-bit의 데이타 화소라면 X_magnitude에 대해 10 사이클, Y_magnitude에 대해 10 사이클이 필요하다. 따라서 명령어 사이클이 100 ns라 가정하면 총 소요시간은 2 μ s이다.

```
own_pixel ← Latch;
        s ← South pixel;
            /*Store pixel in memory;
              Sliding to North*/

T ← own_pixel;
        s ← Southwest pixel; /*Sliding to East*/

T ← T - Latch;
        s ← West; /*Sliding to South*/

T ← T   (Latch≪Latch);s ← Northwest pixel;
         /*≪represent 1-bit left shigt*/

T ← T   Latch;
        s ← North; /*Sliding to West*/
        s ← Northeast; /*Sliding to North*/

T ← T + Latch;
        s ← East; /*Sliding to North*/

T ← T + (Latch≪Latch);s ← Southeast;
        /*Sliding to North*/

X_Mag ← (T+Latch) ≫2;
        /*two 1-bit right shifts*/
```

그림 6. Sobel 연산자에 대한 병렬 알고리즘
Fig. 6. The Parallel Algorithm for Sobel Operator

VI. 결 론

본 연구에서는 영상처리를 위한 ASIC 개발을 위해 기존의 격자연결 SIMD 병렬 컴퓨터들이 가지고 있는 단점, 즉 이웃한 PE간의 근접 통신 부담, 데이타 입출력 부담, 자치권 부재 및 PE들간의 복잡한 상호 연결 방등의 단점들을 대폭 경감시킨 SliM의 한 PE 구조 및 9개의 PE를 격자 연결한 SliM Image Processor ASIC의 구조를 설계하였다. SliM에서 제안된 "sliding" 동작(근접 통신과 계산이 동시에 수행되어 근접 통신 부담을 대폭 감소시키는 개념), 데이타 입출력이 계산과 중첩, 가상 대각선 link, bit-serial 통신과 bit-parallel 계산등은 SliM의 성능을 향상시키는 주요 특징들이다. SliM Image Processor는 확장성이 뛰어난 격자 연결 방식으로 설계하였고 이는 곧 격자 연결 SIMD SliM Array Processor 설계를 용이케 한다. 설계된 명령어들은 영상처리 알고리즘들을 분석 필요한 연산을 효율적으로 수행하고 기존의 SIMD 컴퓨터인 MPP, GAPP 및 CLIP7A에서 사용하는 명령어들을 대부분 채택하여 실시간 영상처리 응용에 효과적으로 사용될 수 있게 하였다. 추후 설계된 PE, Image Processor 및 명령어에 대해 VHDL(VHSIC Hardware Description Language) 모델을 구현하여 기능 수행을 검증하고 논리합성을 수행한 후 여러개의 PE들을 연결 시킨 병렬처리 방식 VLSI chip 구현을 위한 연구와 특별한 목적의 architecture에 대해 sliding 개념의 적용성에 대해서도 연구를 계속할 것이다.

참 고 문 헌

[1] S. H. Unger, "A computer oriented toward spatial problems" Proc. IRE, vol. 46, pp. 1744-1750, Oct. 1958.

[2] Special Issue on Computer Architecture for Image Processing. IEEE Computer, Jan. 1983.

[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," Procedings of IEEE, vol. 76, Aug. 1988, pp. 970-981.

[6] T. Kushner, A. Y. Wu, and A. Rosenfeld, "Image processing on MPP," Pattern Recognition, vol. 15, pp. 121-130, 1982.

[7] 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.

[8] 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.

[9] 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.

[10] M. H. Sunwoo and J. K. Aggarwal, "A sliding memory plane array processor," IEEE Trans. Parallel and Distributed Systems, vol. 4, pp. 601-612, June 1993.

[11] M. H. Sunwoo and J. K. Aggarwal, "A vision tri-architecture (VisTA) for an integrated computer vision system," in Proc. DARPA Image Understanding Benchmark Workshop, Avon, CT, Oct. 1988.

[12] 선우 명훈, "A SIMD Architecture with Zero or Little Communication Overhead for Low Level Vision," 제5회 신호처리 합동학술대회, pp. 489-492, Sep. 1992.

[13] J. P. Strong, "Basic Image Processing Algorithms on the Massively Parallel Processor", Multicomputers and Image Processing Algorithms and Programs Kendall Preston, Jr. and Leonard Uhr, Ed., Academic Press, 1982, pp. 47-86.

[14] K. E. Batcher, "Design of a massively parallel processors," IEEE Trang. Comput., vol. C-29, pp. 836-840, Sep. 1980.

[15] K. E. Batcher, "Bit-serial parallel processing systems," IEEE Trans. Comput., vol. C-31, pp. 377-384, May 1982.

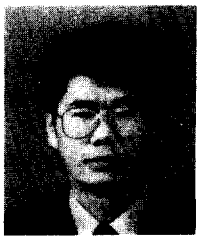
[16] R. Davis and D. Thomas, "Systolic array chip matches the pace of high-speed processing," Electronic Design, vol. 32, No. 22, pp. 207-218, Oct. 1984.

[17] 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.

[18] C. Weems, E. Riseman, A. Hanson, and A. Rosenfeld, "An integrated image understanding benchmark : recognition of a 2 1/2D mobile," in Proc. DARPA Image Understanding Workshop, Washington D.C., 1988, pp. 11-126.

[19] Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, The Design and Analysis of Computer Algorithms, Addison Wesley, 1974.

이 연구는 1993년도 한국과학재단 연구비지원에 의한 결과임. (과제번호 : 931-09-016-1)



安炳德 (Byung Dug Ahn) 정회원
1968년 4월 1일생
1993년 2월 : 아주대학교 전자공학과 졸업 (공학사)
1994년 3월 ~ 현재 : 아주대학교 전자공학과 석사과정
※주관심분야 : ASIC 설계 및 Parallel Architecture

鄭 智 元 (Ji Won Jung) 정회원
1968년 8월 16일생
1993년 2월 : 아주대학교 전자공학과 졸업(공학사)
1994년 3월 ~ 현재 : 아주대학교 전자공학과 석사과정
※주관심분야 : ASIC 설계 및 Parallel Architecture

鮮于明勳(Myung Hoon Sunwoo) 정회원
1980년 2월 : 서강대학교 전자공학과 졸업(공학사)
1982년 2월 : 한국과학기술원 전기 및 전자공학과 졸업 (공학석사)
1982년 3월 ~ 1985년 8월 : 한국전자통신연구소 연구원
1990년 8월 : Univ. of Texas at Austin(공학박사)
1990년 8월 ~ 1992년 8월 : Motorola DSP Chip Operation 연구원
1992년 8월 ~ 현재 : 아주대학교 전자공학과 조교수
※주관심분야 : 통신 및 신호처리 ASIC 설계, Parallel Architecture, VLSI 구조 및 설계, 디지탈 이동통신