論文96-33A-10-23

SliM 이미지 프로세서 칩 설계 및 구현

(Design and Implementation of the SliM Image Processor Chip)

邑 壽 煥 *, 鮮 于 明 勳 *

(Soohwan Ong and Myung Hoon Sunwoo)

요 약

격자 연결 SIMD(Single Instruction Stream - Multiple Data Streams) 구조를 갖는 SliM (sliding memory plane) 어레이 프로세서(array processor)는 기존의 SIMD 어레이 프로세서가 갖는 근접통신 시간부담(inter-processing element(PE) communication overhead), 데이타 입출력 시간부담(data I/O overhead), 복잡한 상호 연결망 등 단점들을 극복하고 성능 향상을 위해 이미 제안된 바 있다. 본 논문은 이 SliM 어레이 프로세서를 구현하기 위해 5 x 5 PE들이 격자연결로 구성된 SliM 이미지 프로세서 ASIC (application specific integrated circuit) 칩의 설계 및 구현에 관하여 기술한다. 설계된 칩은 성능 향상 및 실제 구현상의 문제로 기존에 발표된 아키텍쳐를 수정 구현하였다. 설계시 Top-down IC 설계 방식에 따라 VHDL(VHSIC Hardware Description Language) 모델을 정립하였고 논리합성 및 Back-end 설계를 수행하였다. 셀 라이브러리는 0.8 μm VTI(TM) 표준 셀 라이브러리(v8r4.4)를 사용하여 ASIC 칩을 구현하였으며 전체 게이트 수는 25개의 128 x 9 bit SRAM을 제외하고 55,255개이다. 제작된 칩은 die 크기가 326.71 x 313.24 mil^2인 144핀 MQFP로 구현되었으며 동작 주파수는 25 MHz로 한 칩당 625 MIPS의 성능을 제공한다. 제작된 칩의 성능평가를 위해 병렬 알고리즘을 개발하였으며 기존의 외국 이미지 프로세서들과 비교하여 성능이 우수함을 입증하였다.

Abstract

The SliM(Sliding Memory Plane) array processor has been proposed to alleviate disadvantages of existing mesh-connected SIMD(Single Instruction Stream - Multiple Data Streams) array processors, such as the inter-PE(Processing Element) communication overhead, the data I/O overhead and complicated interconnections. This paper presents the design and implementation of SliM image processor ASIC(Application Specific Integrated Circuit) chip consisting of mesh connected 5 x 5 PE. The PE architecture implemented here is quite different from the originally proposed PE. We have performed the front-end design, such as VHDL(VHSIC Hardware Description Language) modeling, logic synthesis and simulation, and have done the back-end design procedure. The SliM ASIC chip used the VTI 0.8 μm standard cell library (v8r4.4) has 55,255 gates and twenty-five 128 x 9 bit SRAM modules. The chip has the 326.71 x 313.24 mil^2 die size and is packed using the 144 pin MQFP. The chip operates perfectly at 25 MHz and gives 625 MIPS. For performance evaluation, we developed parallel algorithms and the performance results showed improvement compared with existing image processors.

I. 서 론

병렬 SIMD 어레이 프로세서 컴퓨터들은 많은 프로세서들을 연결하여 같은 명령어를 수행함으로써 본질적으로 수천 또는 수만 개의 프로세서를 사용 massive

* 正會員, 亞洲大學校 電機電子工學部

(School of Electronical and Elec. Eng., Ajou Univ.)

接受日字: 1996年7月31日, 수정완료일: 1996年10月2日

parallelism을 성취할 수 있기 때문에 실시간 신호 및 영상처리등과 같이 막대한 양의 데이타를 취급하는 응용 분야에서 중요한 역할을 한다 [1]. Bell Lab의 Unger가 이차원적 구조를 갖는 데이타의 처리를 위하여 격자연결 방식에 기초를 둔 컴퓨터를 최초로 제안한 이후 [2], 신호와 영상 데이타의 구조에 적합한 많은 어레이 프로세서 컴퓨터들이 제안되어졌고 실제 공학 및 과학 분야에서 광범위하게 사용되고 있다 [3,4]. 어레이 프로세서들에서 PE들끼리의 상호 연결망(interconnection network)은 격자(mesh), 선형연결(linear array), 하이퍼큐브(hypercube), 트리(tree), 버스(bus), 피라미드(pyramid) 등이 있다. 격자연결 방식을 사용한 SIMD 어레이 프로세서들에는 SOLOMON [5], ILLIAC IV [6], MPP [7-9], CLIP [10-13], DAP [14], GAPP [15,16], CAAPP [17], YUPPIE [18], CM I [19], MasPar [20], BLITZEN [21], MGAP-2 [22] 등이 있으며 선형연결 방식을 사용한 SIMD 어레이 프로세서로는 IMAP [23], VIP [24] 등이 있다.

그러나 기존의 SIMD 어레이 프로세서들은 근접통신 시간부담, 데이타 입출력 시간부담 등의 심각한 단점을 가지고 있다 [25-29]. 이러한 단점들을 감소시키기 위해 'Sliding' 개념을 도입하여 이미 발표된 바 있는 SliM 어레이 프로세서는 근접통신 및 데이타 입출력 시간부담을 계산과 중첩시킴으로써 시간부담이 중첩되어 사라지는 효과를 거둘 수 있다 [30-33]. SliM 어레이 프로세서는 SIMD 병렬 컴퓨터로서 수천, 수만개의 PE들이 연결될 수 있으며 본 논문에서는 SliM 어레이 프로세서를 구현하기 위한 SliM 이미지 프로세서 칩의 설계와 구현에 관해 기술한다.

영상처리 알고리즘을 분석하여 효율적인 명령어 집합을 설계하여 PE의 ALU는 간단하고 함축적으로 설계하였다. 본 논문에서 구현한 SliM 이미지 프로세서는 기존에 발표된 SliM 이미지 프로세서[30-33]와는 달리 성능 향상 및 실제 구현상 이점을 위해 대폭 수정하였으며 그 차이점은 다음과 같다. 1) 버스의 복잡도를 줄이기 위해 accumulator-based 구조를 채택하였으며 2) accumulator와 주소 레지스터(*AR*: Address Register)를 확장된 accumulator 또는 확장된 variable length 쉬프터로 사용한다. 3) ALU 연산과 쉬프트 연산이 동시에 수행되도록 구현하여 곱셈기나 나눗셈기를 사용하지 않고 곱셈 및 나눗셈을 효율적으로 연산할 수 있으며 4) 모든 명령어는 조건적으로 수행할 수 있도록 하였다. 5) 2의 보수 연산을 위해 8 비트 대신 9 비트의 datapath를 채택하여 성능을 향상시켰다.

과거의 어레이 프로세서들은 자치권을 지원하지 않았으나 최근 개발된 어레이 프로세서들은 SIMD 컴퓨터들이 마치 MIMD(Multiple Instruction Streams - Multiple Data Streams) 컴퓨터들 같이 각각의 PE들이 서로 다른 연산(operation)들을 수행하는 연산 자치권(operation autonomy), 서로 다른 주소(address)를 갖는 주소 자치권(addressing autonomy), 서로 다른 연결(connection)을 갖는 연결 자치권(connection autonomy) 등을 갖는다. 이 자치권에 의해 SIMD 컴퓨터의 각 PE는 서로 다른 명령어를 수행하여 복잡한 계산을 처리할 수 있는 MIMD 컴퓨터의 기능을 갖고 있어 응용 범위의 폭이 점차 넓어지고 있다. SliM에서는 각 PE마다 독립적인 동작을 수행할 수 있는 세가지 자치권, 즉 연산 자치권, 주소 자치권, 연결 자치권을 제공하여 마치 MIMD 병렬 컴퓨터와 같이 독립적인 동작을 가능하게 한다. 이러한 특성들로 인해 SliM은 기존의 다른 영상처리를 위한 SIMD 병렬 컴퓨터보다 성능을 월등히 향상시켰다.

칩 구현시 Top-down IC 설계 방식에 따라 VHDL을 이용 구현하였다. VHDL 모델의 기능 확인을 위해 VANTAGE(TM) CAD 툴(tool)을 이용하여 모든 명령어와 다수의 알고리즘에 대해 시뮬레이션을 수행하였다. 논리합성은 COMPASS ASIC Synthesizer를 이용하였으며, COMPASS Chip Compiler를 이용 Back-end 설계를 수행하였다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 SliM의 구조 및 명령어 설계에 관해 기술하고 3장에서는 설계 및 구현에 대해 기술한다. 4장에서는 칩의 성능평가를 하며 끝으로 5장에서 결론을 맺는다.

Ⅱ. SliM의 구조 및 명령어 집합 설계

이 장에서는 하나의 PE 아키텍처 및 사양, SliM 이미지 프로세서의 구조, 명령어 설계 등에 대해서 기술한다.

1. PE 아키텍처

그림 1은 하나의 PE 아키텍처를 나타낸다. 256

gray-level을 갖는 8 비트 화소 계산시 발생하는 음의 정수 값을 처리하기 위해 SliM PE는 9 비트를 기본으로 한다. 하나의 PE는 *ALU*, 레지스터(registers), 기억소자(*SRAM*), 멀티플렉서(*MUXs*) 그리고 *4* x *1* 스위치 소자(*SW*)들로 구성되어 있다. *4* x *1 SW*는 이웃하는 PE들(동서남북) 사이에 비트-serial 통신을 위한 상호연결망을 제어한다. 그리고 *s* 레지스터는 *4* x *1 SW*의 입력에 대해 세가지 통신모드를 지원한다. 이웃 PE들에서 입력되는 데이타를 선택해서 *s* 레지스터로 데이타를 입력받는 receiving 모드 또는 통과시켜서 다른 이웃 PE들(동서남북)로 전송시키는 bypassing 모드, 그리고 데이타를 입력받으면서 통과시키는 receiving/bypassing 모드가 지원된다. 이러한 세가지 통신모드의 조합은 근접통신, 비근접 통신(nonlocal communication) 및 broadcast에 유용하다. 또한 대각선 통신 링크를 지원하여 4개의 링크로 8 방향의 통신을 제공한다. *d*와 *d'* 레지스터는 좌우측 PE들의 *d*와 *d'* 레지스터에만 연결되어 데이타 입출력을 담당한다.

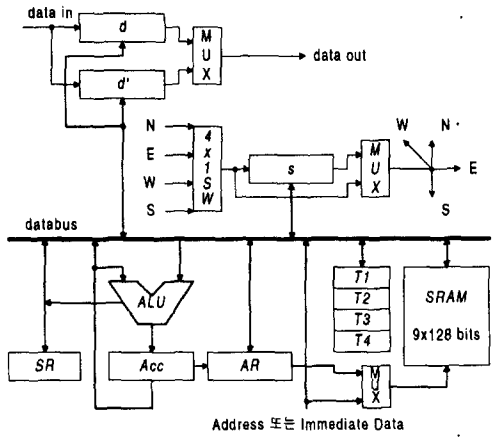

그림 1. SliM PE 아키텍처
Fig. 1. The Architecture of a SliM PE.

각 *ALU*가 화소에 대해 연산하는 동안 이웃 화소가 *s* 레지스터를 통해 들어오므로 계산과 근접통신이 동시에 일어날 수 있다. 또한 *d* 또는 *d'* 레지스터를 통한 입출력 동작도 근접통신 및 계산 과정과 별개로 동시에 일어날 수 있다. 이러한 기능은 근접통신, 데이타 입출력 및 계산을 독립적으로 제어하기 때문에 가능하며 근접 PE간의 통신과 입출력 시간부담을 계산 시간과 중첩시켜 이들 시간부담을 크게 줄일 수 있다.

PE를 이루고 있는 다른 주요 소자들은 accumulator (*Acc*), 상태 레지스터(*SR*: Status Register), 각 PE 내의 기억소자 주소를 독립적으로 제공하기 위한 주소 레지스터(*AR*), 4개의 범용 9-비트 레지스터(*Ts*) 그리고 PE의 내부에 위치해 데이타를 저장할 수 있는 128 x 9 비트 기억소자(SRAM) 등이 있다. 산술과 논리적인 쉬프트를 수행하는 *Acc*와 *AR*의 조합은 곱셈, 덧셈 및 부동 소수점 계산에 유용하게 쓰일 수 있다. *T* 레지스터는 중간 결과값을 저장하기 위해 이용되며 이는 기억소자 읽기/쓰기 시간부담을 줄일 수 있고, 특히 곱셈 및 나눗셈 연산에 효과적으로 쓰인다.

9 비트 *ALU*는 4 가지의 논리 함수와 8 가지의 산술 연산을 제공하며 carry lookahead 가산기(adder)를 사용하여 이진보수(two's complement) 연산을 수행한다. SliM PE는 accumulator-based 구조이므로 *ALU*의 입력중 하나는 *Acc*값이 된다. *ALU*에서 제공하는 함수들은 영상처리 알고리즘들을 수행할 수 있도록 설계된 명령어들을 효율적으로 지원한다. 또한 *ALU*의 기능 외에 뺄셈을 이용하여 2의 보수를 취하는 동작이 한 명령어 사이클에 이루어지도록 설계하였다.

기존의 SIMD 컴퓨터들과는 달리 SliM에서는 각 PE마다 독립적인 동작을 수행할 수 있는 3가지 종류의 자치권, 즉 연산 자치권, 주소 자치권, 상호연결 자치권을 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 있는데 그 내용은 다음과 같다.

1) 연산 자치권 : *SR*의 flag에 따라 각 PE가 서로 다른 동작을 수행할 수 있게 한다. 연산 자치권은 IF (condition) THEN (statements) ELSE (statements)와 같은 조건적 연산 수행에 효과적으로 사용된다.
2) 주소 자치권 : 기억소자 주소 레지스터 *AR*을 두어 기억소자 주소를 각 PE마다 독립적으로 제공할 수 있어 서로 다른 데이타에 대한 연산을 수행할 수 있다.
3) 연결 자치권 : 각 PE들은 이웃 PE와 독립적인 연결을 수행하여 edge를 따라 통신을 수행하는 curvature 계산과 같은 알고리즘에 매우 효율적이다.

그림 1에서 데이타 버스는 9 비트 데이타 경로를 갖으며 d, d' 및 s 레지스터 부분의 데이타 경로는 1 비트 데이타 경로를 갖는다. 각 레지스터들은 한 개의 화소를 포함하며 ALU는 9 비트 parallel 방식으로 연산을 수행한다. 따라서 SliM은 1 비트 serial 통신과 9 비트 parallel 계산을 수행한다. 영상처리에 대한 데이타들은 2진 데이타 형식이라기 보다는 그레이 준위를 갖는 정수형이기 때문에 비트 parallel의 계산이 더 효율적이다. 그러나 PE들간의 9 비트 parallel 링크들은 VLSI 면적의 많은 부분을 차지하며 칩상에서 입출력 핀 수를 증가시키는 원인이 되므로 SliM에서는 이웃 PE들간의 통신 링크 및 데이타 입출력 링크는 1 비트 serial 방식을 채택 VLSI 면적을 줄이고 입출력 핀 수를 감소시켰다.

2. 명령어 집합(Instruction Set) 설계

실시간 영상처리를 위해 주로 사용되는 알고리즘들, 즉 중간값 검출, 평균값 검출, 2차원 콘볼루션, 히스토그램, Sobel 연산 등을 위한 연산들과 기존의 격자연결 SIMD 병렬 컴퓨터들의 명령어들을 분석하여 SliM의 명령어를 설계하였다. 표 1은 SliM의 명령어 집합을 나타낸다. 하나의 칩에 다수의 PE들을 집적시켜야 하므로 한 PE의 면적을 줄이기 위해 반드시 필요한 명령어를 간단하고 효율적으로 설계하였다. 설계된 명령어의 마이크로코드(microcode)는 28-비트 width로서 수직 마이크로 명령(vertical microinstruction) 형식이며 sliding 동작과 데이타 입출력 동작은 각 명령어들과 동시에 수행할 수 있다. 명령어는 크게 데이타 움직임, 연산 및 논리 형태로 구분된다. 곱셈, 나눗셈 및 부동 소수점 연산은 한 칩내에 보다 많은 PE를 집적시키기 위해 전용 곱셈기, 나눗셈기 및 부동 소수점 연산기를 사용치 않아 하드웨어 부담을 감소시켰으나 효율적인 알고리즘 수행을 위해 쉬프트와 연산이 한 클럭 사이클에 동시에 수행될 수 있게하여 실행 명령어 수를 반 이상 줄였다. 곱셈과 나눗셈을 위한 명령어 MUL과 DIV를 제외한 모든 명령어는 한 사이클 내에 실행된다. 곱셈 연산은 이진보수 곱셈 알고리즘을 사용하여 11 명령어 사이클, 이진보수 restoring 나눗셈 알고리즘을 사용한 나눗셈은 13 명령어 사이클에 수행 완료된다. 부동 소수점 연산 또한 9-비트 bias-256 지수와 18-비트 이진보수 가수 형식으로 313 클럭 사이클에 수행된다.

표 1. 명령어 집합
Table 1. Instruction Set.

명령어 구분	Mnemonic	기능 설명
데이타 움직임	MOVE	Move Data
	MOVEcc	C 레지스터를 Update하는 Move
	SHIFT	Data shift (10 가지)
연산	ADD	덧셈
	SUB	뺄셈
	ADC	캐리를 포함한 덧셈
	SBC	캐리를 포함한 뺄셈
	MUL	곱셈
	DIV	나눗셈
	INC	'1'만큼 증가
	DEC	'1'만큼 감소
	COMP	2의 보수화
논리	AND, OR, XOR, NOT	일반 논리연산

3. SliM 이미지 프로세서

그림 2는 한 칩에서 5 x 5 PE들로 연결된 SliM 이미지 프로세서의 병렬 VLSI 구조를 나타낸다. 25개의 PE들은 확장성을 갖는 격자 형태로 연결되어 있다. 즉 격자 연결은 수천, 수만 개의 PE들을 연결시킨 어레이 프로세서의 구현이 용이하다. 앞에서 언급했듯이, 4개의 근접 PE들간의 연결은 각 PE내에 있는 s 레지스터에 의해 1 비트 serial 방식으로 이루어진다. 가장자리에 위치한 PE의 개수는 16개이며 각 PE는 한 비트의 NEWS 출력을 가지고 있다. 또한 칩의 4방향에서 각각 5개씩의 N, E, W, S 입력이 들어오므로 칩들 사이의 근접통신에 필요한 핀 수는 36개이며 데이타 입출력을 위해 좌우로 연결된 d 또는 d' 레지스터의 연결을 위한 핀 수는 10개만 필요하다. 전체적으로 25개의 PE들에 대해 근접통신과 데이타 입출력을 위한 핀 수는 46개로서 칩 상에서 입출력 핀 수 및 칩 면적을 감소시키는 효과를 얻을 수 있다.

한 명령어 사이클 내에 PE의 내부 메모리에 있는 데이타를 이용 명령어를 실행하기 위해서는 메모리 제어 신호의 타이밍이 정확해야 한다. 이를 위해 게이트로 구성된 지연소자를 이용할 경우 실제 칩 구현시 오류 동작할 우려가 많기 때문에 내부적으로 two phase non-overlapping 클럭 발생회로를 구현하여 타이밍을 조절하였다. 명령어를 명령어 레지스터로 fetch한 후 디코더를 거친 신호를 명령어 래치로 fetch한다. 디코

더는 하나로서 모든 PE들에 broadcast한다. 파이프라인 단계는 명령어 fetch, 디코드, 실행의 3단계이다.

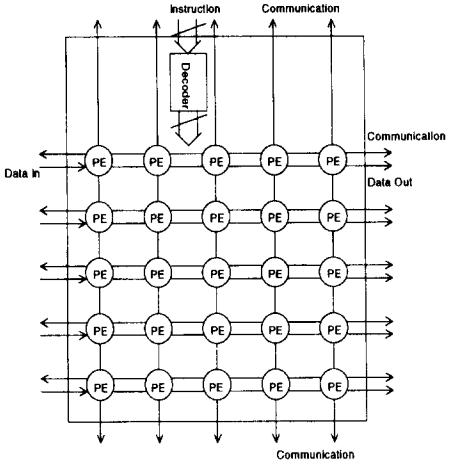

그림 2. SliM 이미지 프로세서 구조
Fig. 2. The Architecture of SliM Image Processor.

Ⅲ. 설계 및 구현

이 장에서는 SliM의 한 PE 구조에 대한 각 블럭들의 VHDL behavior, structure 모델, 논리합성, 시뮬레이션 및 Back-end 설계에 대해 기술한다.

1. VHDL 모델 및 논리합성

각 블럭들의 제어신호(control signal)를 정의하고, 데이타 입력값에 대해 시뮬레이션을 수행한 후 top-down 설계 방식에 따라 CAD 툴에서 논리합성이 가능토록 모델링 하였다. 구현된 동작(behavior) 모델은 각 블럭들에 대해 구조(structure) 모델로 포트 매핑 시킴으로써 하나의 PE에 대한 VHDL 모델을 구현하였고, 25개의 PE를 포트 매핑시켜 SliM 이미지 프로세서의 모델링을 수행하였다. 시뮬레이션 및 프로그램의 용이성을 위해 C 언어를 이용하여 어셈블러를 설계하였다. SliM PE에 대해 설계된 명령어들의 가능한 모든 조합에 대해 VANTAGE(TM)를 이용한 시뮬레이션을 통해 기능수행 여부를 확인하였다. 또한 자주 쓰이는 병렬처리 방식 영상처리 알고리즘들인 중간값 검출, 평균값 검출, 2차원 콘볼루션, 히스토그램, Sobel 연산, 최대/최소값 검출 등의 알고리즘에 대해서도 시뮬레이션을 수행하였다.

SliM의 한 PE에 대한 VHDL 모델을 바탕으로 COMPASS(TM)의 0.8 μm 표준 셀 library를 사용하여 논리합성을 수행하였다. *ALU* 및 디코더와 같은 조합 논리회로(combinational logic circuit)에 대해서는 최적화된 VHDL 모델을 이용하여 SYNOPSYS (TM)로 논리합성을 수행하였으며 래지스터와 같은 순차적 논리회로(sequential logic circuit) 및 MUX등은 COMPASS(TM)의 ASIC Synthesizer와 Datapath Compiler를 사용 논리합성을 수행하였다. 논리합성 후의 시뮬레이션 결과는 게이트 레벨의 모든 시간지연 및 타이밍 정보를 포함하고 있으므로 fanin/out, rising/falling time 등과 같은 실제 회로상의 특성들을 고려하여 VHDL 모델의 타이밍 시뮬레이션 결과와 동일함을 검증하였고 정확한 하드웨어 구조 및 명령어 기능을 확인하였다.

VHDL을 이용하여 각 컴포넌트를 매핑한 후 SYNOPSYS(TM)를 이용하여 최상위 블럭을 설계하였다. 설계된 최상위 블럭에 대한 EDIF 파일을 COMPASS(TM)에서 읽어들여 논리합성된 컴포넌트로 매핑하여 하나의 PE를 완성하였다. 각각의 컴포넌트는 COMPASS(TM)와 SYNOPSYS (TM)를 이용하여 가장 최적화된 결과를 이용하였다.

하나의 PE에는 기억소자를 제외하고 2 입력 NAND 게이트를 기준으로 2102개이므로 트랜지스터의 수는 약 8408개 정도이다. 최종 SliM 이미지 프로세서칩은 5 x 5개의 격자연결된 PE와 하나의 디코더로 구성되어 있으며, 55,255개의 게이트와 각 PE가 포함하고 있는 128 x 9 비트 SRAM으로 이루어져 있다. 전체 핀 수는 데이타 입출력을 위한 핀 10개, 근접통신을 위한 핀 36개, 클럭과 reset이 각각 하나, 명령어를 위한 핀 28개를 합하여 모두 76개이다. SliM 이미지 프로세서의 동작 확인을 위한 시뮬레이션은 일반적인 병렬처리 방식 영상처리 알고리즘들과 모든 명령어의 조합에 대하여 수행하였다. 대부분의 알고리즘에서 계산과 통신이 거의 중첩되어 통신 부담이 없었다.

2. Back-end 설계

SliM 칩의 Back-end 설계는 COMPASS CAD 툴(TM)을 사용하였다. 각 블럭을 정의하고 위치를 설정하는 Floorplan을 마친 후 블럭에 cell을 배치하고 블럭 내의 배선과 블럭간의 배선을 수행하였다. 이를 마친 후 RC값이 포함된 지연 정보를 갖는 netlist를 추출하여 postroute 시뮬레이션을 수행하였다.

Floorplan은 모두 7 차례에 걸쳐 수행하여 그 중 가장 최적화된 Floorplan으로 최종 Back-end 설계를 마쳤다. 블럭들의 개수와 배치를 조절하며 7 차례 시도하여 최적의 결과를 얻어 최종적으로 die 크기는 326 x 313 mil^2 이다. 모두 25개의 내부 기억소자 블럭과 13개의 표준 셀 블럭으로 구성된다. 기억소자 블럭은 back-end 설계시 standard cell 영역에 포함되지 않고 독립적인 블럭으로 존재하기 때문에 25개의 내부 기억소자 블럭이 존재한다. 13개의 표준 셀 블럭에는 2개의 PE가 포함된 12개의 블럭과 하나의 PE와 디코더를 포함하는 중앙의 1개 블럭으로 구성된다.

최종 floorplan을 이용 표준 셀 영역과 각 블럭간의 배선을 실시하였다. 각 블럭간의 배선은 블럭간의 간격(routing channel), 모양과 위치 등의 영향으로 연결되지 않은 신호가 발생한다. 이들은 직접 수동 배선을 해주었다. 연결되지 않은 신호들을 모두 연결해 준 후 RC 값이 포함된 지연 정보를 가진 netlist를 추출하였다.

배치 및 배선을 마친 후 추출한 netlist와 front-end 설계의 netlist가 동일함을 확인하고 postroute 시뮬레이션을 실시하였다. worst, typical, best case에 대해 시뮬레이션을 수행하여 결과를 확인하였다. 최대 클럭 속도는 worst-case인 경우 18 MHz까지 수행하였다. 제조공정과정에서 시험을 위해 요구하는 클럭 속도 1 MHz의 testvector를 작성하였으며 이는 약 3 만 line 이다. 또한 공정 후 내부 회로의 결함 여부를 알 수 있도록 toggling rate은 100%로 올렸으며 수행 가능한 명령어는 모두 수행하였다. 그림 3은 SliM 이미지 프로세서 칩의 Micrograph이다.

SliM 칩은 144 핀 MQFP 패키지(package)로 구현하였다. 사용하고 있는 핀은 시험을 위한 핀 10개를 포함하여 87개의 신호 핀과 57개의 전력(power) 핀으로 구성된다. 소모되는 전력은 worst-case로 계산하였을 때 3 W정도였으며 typical인 경우 1.68 W이다. 전력 소모가 많으므로 전력 핀을 57개로 많이 두었다. 또한 시험을 위해 10개의 핀 중 9개는 내부 9개 PE의 s 레지스터의 NEWS 출력과 연결하였으며 한 핀은 시험 제어신호(test enable)로 사용한다. 패드(PAD)는 모두 CMOS 패드를 사용하였다. SliM 이미지 프로세서 칩을 이용하여 병렬 시스템을 구축할 시에 SliM 칩들끼리 연결되므로 특성이 좋은 CMOS 패드로 통일하였다. 출력은 fanout을 고려하지 않아도 되는 시험 핀만 2 mA CMOS 패드를 사용하였으며 나머지 출력 핀은 4 mA CMOS 패드를 사용하였다. 하나의 출력 핀은 최대 2개의 SliM 칩과 연결된다.

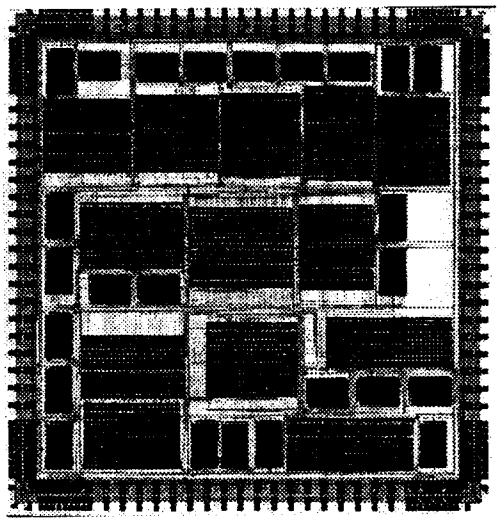

그림 3. SliM 칩의 Micrograph
Fig. 3. The Micrograph of SliM Chip.

Ⅳ. 성능평가

Simulation 상 18 MHz가 최대 클럭 주파수였으나 실제 칩은 testvector를 이용하여 25 MHz까지 동작 확인을 수행하였다. 현재 5 x 5개의 SliM 이미지 프로세서 칩을 연결한 prototype SliM 어레이 프로세서를 설계하여 시험중이다. 하나의 어레이 프로세서 보드에는 25 x 25개의 PE로 구성되어 있는데 표 2는 2장의 어레이 프로세서 보드를 사용하여 50 x 50개의 PE가 있는 경우와 512 x 512개의 PE가 있는 경우에 대해 비교 분석한 것이다. 2-D 콘볼루션의 경우 곱셈 연산이 많이 사용되는데 ALU 연산과 shift 연산이 동시에 처리되는 구조이기 때문에 11 사이클에 곱셈이 이루어진다. 따라서 전체 연산 수가 크게 줄어든다. 뿐만 아니라 근접통신이 계산과 완전히 중첩되기 때문에 기존의 어레이 프로세서들에 비해 그 성능이 매우 우수하다. 다른 알고리즘에 대해서도 비트-parallel 연산을 수행하며 근접통신 시간부담이 적기 때문에 성능이 월등하다.

표 3은 기존의 어레이 프로세서들과의 성능을 비교한 것으로 MGAP [34]에서 언급한 내용을 기초로 한다. 이 표에서 SliM이 국외의 어레이 프로세서들 보다

성능이 우수함을 알 수 있다.

표 2. SliM 칩을 이용한 성능평가
Table 2. Performance Evaluation using SliM Chip.

	명령어 수		연산 시간(μs)	
# of PEs	50 x 50	512 x 512	50 x 50	512 x 512
Sobel 연산	10,406	86	420	3.44
2-D 콘볼루션	19,965	165	800	6.60
중간값 검출	27,588	228	1,100	9.12
평균값 계산	21,538	178	860	7.12
히스토그램	124,553	6687	4,980	267.48

표 3. 성능비교
Table 3. Performance Comparison.

Array Processor	클럭 속도	8-bit 덧셈 [μs]	32-bit 덧셈 [μs]	8-bit 곱셈 [μs]
CLIP4 (96 x 96 PEs)	25μs	200	-	2000
DAP (64 x 64)	200ns	5.4	30	45
MPP (128 x 128)	100ns	2.5	9.7	7.0
GAPP (96 x 128)	100ns	2.5	9.7	25.2
MGAP (128 x 128)	40ns	1	1	5.24
SliM (50 x 50)	40ns	0.081	0.437	0.437

V. 결 론

격자연결 SIMD 컴퓨터들이 일반적으로 갖는 단점인 근접통신 시간부담, 데이타 입출력 시간부담을 대폭 감소시킬 수 있는 'Sliding' 이라는 개념을 도입한 SliM 이미지 프로세서 칩을 구현하였다. 또한 SliM 이미지 프로세서는 자치권 부재 및 PE들간의 복잡한 상호연결망 등의 단점을 경감시킬 수 있다. 다수의 영상처리 알고리즘을 분석하여 효율적인 명령어 집합을 설계하였으며 하나의 PE에 대한 아키텍쳐 정의 및 사양을 결정하였고, 특히 확장성을 두어 응용 분야에 따라 용이하게 확장할 수 있도록 설계하였다.

설계된 한 PE를 VHDL 모델로 구현 기능 검증을 수행하였다. 가능한 모든 명령어 조합을 포함하여 다수의 영상처리 알고리즘에 대해 시뮬레이션을 수행하였다. 논리합성을 완료하여 기능 검증 및 타이밍 검증을 수행하였고 ASIC 구조 설계를 완료하였으며 ASIC 칩을 제작하였다. 제작된 칩의 시험을 위해 시험보드를 제작하여 testvector에 대한 기능 검증을 완료하였다. 칩의 동작속도 및 칩의 정확한 특성을 추출하기 위해 하나의 칩 시험을 위한 PCB 보드를 제작하여 25 MHz에서 동작함을 확인하였으며 이에 대한 성능 평가를 수행하였다. 개발된 칩 25개를 이용하여 Prototype SliM 어레이 프로세서를 구현하였으며 실시간 영상처리를 위한 효율적 병렬처리 알고리즘을 개발하였다. 추후 보다 많은 PE를 집적시킨 SliM 이미지 프로세서 칩을 구현하여 더욱 성능을 향상시킬 것이다.

참 고 문 헌

[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] 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.

[4] M. Maresca, M. A. Lavin, and H. Li, "Parallel architectures for vision," *Proceedings of IEEE*, vol. 76, Aug. 1988.

[5] J. Gregory and R. McReynolds, "The SOLOMON computer," *IEEE Trans. Electronic Comput.*, vol. EC-12, pp. 774-780, Dec. 1963.

[6] 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.

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

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

[9] J. L. Potter, Ed., *The Massively Parallel*

Processor, The MIT Press, 1985.

[10] M. J. B. Duff, "Review of the CLIP image processing system," in *Proc. National Computer Conference*, 1978, pp. 1055-1060.

[11] T. J. Fountain, "Towards CLIP 6 - an extra dimension," in *Proc. IEEECS Comput. Architect.* for Pattern Anal. and Image Database Management, 1981, pp. 25-30.

[12] T. J. Fountain, "Plans for the CLIP 7 chip," *Integrated Technology for Parallel Image Processing*, S. Levialdi, Ed., Academic Press, 1985, pp. 199-214.

[13] 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.

[14] DAP Series *Technical Overview*, Active Memory Technology Inc., 1989.

[15] 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.

[16] 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.

[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, "Polymorphic-torus: a new architecture for vision computation," in *Proc. IEEECS Comput. Architect. for Pattern Anal. Machine Intell.*, Seattle, WA, Oct. 1987, pp. 176-183.

[19] W. D. Hillis, *The Connection Machine*, MIT Press, 1985.

[20] J. R. Nickolis, "The design of the MasPar MP-1: a cost effective massively parallel computer" in *Proc. IEEE Compcon* Spring 90, 1990, pp. 25-28.

[21] 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.

[22] Fan Zhou et al., "Introducing MGAP-2," in *Proc. 5th Int. Symp. The Frontiers of Massively Parallel Computation*, Feb. 1995, pp. 281-288.

[23] 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.

[24] Michael Hall, "A Study of Parallel Image Processing on the VIP Architecture," PhD thesis, LIU-TEK-LIC-1995:41, Department of Electrical Engineering, Linköping, Sweden, October 1995.

[25] 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.

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

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] M. H. Sunwoo and J. K. Aggarwal, "A sliding memory plane array processor for low level vision," in *Proc. Int. Conf. Pattern Recognition*, Atlantic City, June 1990, pp. 312-317.

[32] M. H. Sunwoo and J. K. Aggarwal, "VisTA - an image understanding architecture," *Parallel Architectures and Algorithms for Image Understanding* V. K. P. Kumar, Ed., Academic Press, 1991, pp. 121-154.

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

[34] R. S. Bajwa, P. Keltcher, C. Nagendra and R. M. Owens, "Cost Effective Image and Signal Processing Using Fine Grain Array Processors," *Proc. of the DSPx'94*, June, 1994.

저 자 소 개

邕 壽 煥(正會員)

1994년 2월 아주대학교 전자공학 학사. 1996년 2월 아주대학교 전자공학 석사. 1996년 3월 ~ 현재 아주대학교 전자공학 박사과정. 주관심분야는 영상, 통신 및 신호처리용 ASIC 설계 등임

鮮于 明 勳(正會員)

1980년 2월 서강대학교 전자공학 학사. 1982년 2월 한국과학기술원 전자공학 석사. 1982년 3월 ~ 1985년 8월 한국전자통신연구소 (ETRI) 연구원. 1985년 9월 ~ 1990년 8월 Univ. of Texas at Austin 전자공학 박사. 1990년 8월 ~ 1992년 8월 Motorola, DSP Chip Division, 미국. 1992년 8월 ~ 현재 아주대학교 전기전자공학부 조교수. 주관심분야는 VLSI 및 Parallel Architecture, 통신, 영상 및 신호처리용 ASIC 설계 등임