

## 픽셀-병렬 영상처리에 있어서 포맷 컨버터 설계에 관한 연구

\*김현기, \*\*김현호, \*\*\*하기중, \*\*\*\*최영규, \*\*\*\*\*이천희  
극동정보대학, 충북과학대학, 영동전문대학, 충주대학교, 청주대학교  
전화 : 043-229-8448 / 핸드폰 : 019-415-8448

### A Study on the Design of Format Converter for Pixel-Parallel Image Processing

Hyun-Gi Kim, Hyeon-Ho Kim, Ki-Jong Ha, Young-Gyu, Choi, Gee-Han, Yoo, Cheon-Hee, Yi  
Dept. of Electronic, Chongju University  
E-mail : yicheon@chongju.ac.kr

#### Abstract

In this paper we proposed the format converter design and implementation for real time image processing. This design method is based on realized the large processor-per-pixel array by integrated circuit technology in which this two types of integrated structure is can be classify associative parallel processor and parallel process with DRAM cell. Layout pitch of one-bit-wide logic is identical memory cell pitch to array high density PEs in integrate structure. This format converter design has control path implementation efficiently, and can be utilized the high technology without complicated controller hardware. Sequence of array instruction are generated by host computer before process start, and instructions are saved on unit controller. Host computer is executed the pixel-parallel operation starting at saved instructions after processing start

#### I. 서론

현대의 VLSI 기술은 고밀도로 배열된 처리 소자들 (PE: processing elements)에서 메모리들과 프로세서들의 집적화를 추구하며, 이들 PE 어레이들은 픽셀-병렬 영상 처리 시스템을 하기 위한 기본 형태가 된다.

각각의 PE는 하나의 영상을 갖는 한 픽셀(pixel)을 저장하고 처리한다. 그림 1은 픽셀-병렬 영상 처리 시스템의 구성도를 보여주는데, 이것은 한 개의 처리 소자 배열, 컨트롤러, 주 컴퓨터와 두 개의 포맷 컨버터로 구성되어 있다. 카메라로부터 나오는 아날로그 신호는 디지털 신호로 변환되는데, PE에서 처리하기 위한 신호로 포맷이 되어 PE 어레이에 옮겨진다. 주 컴퓨터로부터 나온 명령들은 컨트롤러를 경유해서 모든 PE에 전달되는데 이 처리된 데이터들은 다음의 처리를 위해 재포맷 된다.

PE 어레이에 필요한 두 개의 집적 회로 구조를 고찰하면 내용을 어드레스가능 메모리 셀들을 사용하는 연상 병렬 프로세서로 구성된 그 첫 번째 구조는 이미 검증되었다. Mask-write and match-and-conditional-write 동작의 구현을 위해 필요한 이 구조에서 각각의 PE에 대해서는 로직이 거의 필요치 않으며 이 동작들은 좀 더 복잡한 작업을 수행하기 위하여 결합될 수 있다. 현재 개발 중에 있는 두 번째 구조는 종래의 DRAM cell 들을 사용하는 미세하게-결정(結晶)된 병렬 구조이다. 이 구조에서 각각의 PE에 필요한 로직은 동일한 칩의 DRAM과 결합됨으로서, 행 디코더에 필요한 것을 없애게 된다. 이런 식으로, 고밀도와 소규모 메모리 셀 크기를 가진 구조가 성취될 수 있다.

이 시스템은 주 컴퓨터로부터 나오는 명령을 구현할 수 있으나 실시간 영상 데이터가 시스템에 흘러 들어가고 나오는 패스가 결여되어 있다. 만일 A/D 컨버

터로부터 나오는 영상 데이터 출력이 pixel-by-pixel을 기초로 한 PE 어레이에 직접 전송된다면, 단지 하나의 PE만이 전송이 가능할 것이다. 따라서 좀 더 효율적인 PE 어레이를 이용하기 위해서는 데이터가 병렬 배열로 전송될 수 있도록 영상 데이터의 포맷변환이 필요하다. 본 논문의 핵심은 실시간 영상 처리를 실현하기 위하여 그림1에서와 같이 비디오 카메라등의 아날로그 영상을 디지털 영상으로 바꾸어 PE 어레이에서 처리하기 위한 입·출력 포맷 변환장치의 설계를 하였다. 시스템 환경은 UNIX OS 상에서 VHDL을 사용하여 Synopsis tool을 이용하여 합성하였고, Xilinx FPGA Chip으로 시뮬레이션하였다.

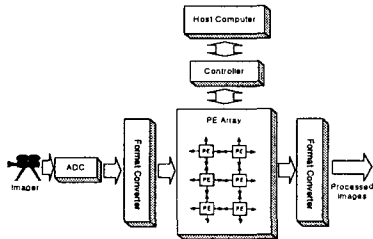


그림 1. PE 배열을 사용한 영상 처리 시스템

## II. 픽셀-병렬 영상처리 시스템

### 2.1 PE(Processing Element) 배열

저비용으로 큰 PE 배열을 만들기 위해서는 고밀도의 PE 구현이 이루어져야 한다. 따라서 적합한 집적회로 구조를 개발하기 위해 먼저 순차주소 메모리 셀을 이용한 새로운 연관 병렬 프로세서 장치의 개발이다. 다음은 DRAM 셀을 사용하는 최적화된 병렬 구조를 들 수 있다. 진자는 각각의 PE에 필요한 논리의 양을 최소화하며, 후자는 메모리 셀의 크기를 최소화한다. 이 두 디자인에서 1비트 크기의 PE 로직의 레이아웃 피치는 메모리 셀의 피치와 동일하게 하고 메모리와 로직사이의 대역폭은 최대화하고 PE의 범위는 최소화한다. 연관 병렬 프로세서 장치는 이중폴리실리콘(Double-polysilicon) CCD-CMOS 기술로 만들어져 왔으며 매우 편리한 것으로 입증되고 있으나 DRAM 셀을 이용한 구조의 구현은 아직 완전하지 않다.

그림 2는 연관 병렬 프로세서에 사용되는 PE 구조를 보여주고 있는데, PE는 3개의 3진수, 즉 0, 1, 그리고 X로 구성된 3진수 값에 의해 동작을 실행한다. 메모리 워드에는 3 상태 부호로 된 64개의 메모리 셀이 구성되며 이 셀은 여러 필드(A, B, C...처럼)로 나누어진다. 이러한 일치동작은 X(don't care)로 표시되어 사용되지 않는 필드와 제시된 패턴에 일치되는 PE를 찾아낸다. 예를 들어 일치동작 (3, X, 5)는 A=3이며 C=5인 모든 PE와 동일시하고, 감지증폭기출력을 1에 맞춘다. 이러한 일치 결과는 함수발생기로 전달되고 함수

발생기는 PE를 차례로 제어하여 기록 구동을 하게 된다. 이러한 방법으로 일치결과 이후의 기록 동작에 대한 조건을 설정할 수 있는 것이다. 만일 기록 구동기를 사용할 수 없다면 PE는 차단되고 그 메모리는 그대로 남아있게 된다. 각 PE의 활동레지스터는 입력된 일치 결과의 논리 조합을 허용하면서 1비트의 상태를 유지한다. 한 단어의 필드들도 역시 차단될 수 있으며, 그렇게 되면 여러 필드들이 변경되는 동안 다른 필드들은 보존된다. 기록(-, 6, 8) 동작은 A 필드의 내용은 바꾸지 않은 채 남겨두고 B=6, C=8에 맞춘다. 이러한 일치와 조건기록동작은 가장 기본적인 데이터 단위를 갖게 되는데, 이것들은 직렬 비트 산술을 포함한 더 복잡한 작업을 하는데 결합될 수 있다.

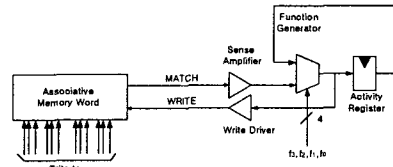


그림 2. 연관 병렬 프로세서에 사용되는 PE 구조

연관 메모리 단어에 속해 있는 특별한 셀을 이용하여 각각의 PE들은 바로 근접해 있는 4개의 인접 PE(좌, 우, 상, 하)에 데이터를 직접 전달할 수 있다. 데이터 이동 동작은 연관동작과 함께 실행될 수도 있다.

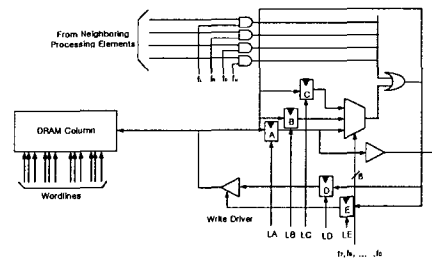


그림 3. DRAM 셀을 사용한 처리소자

그림 3은 두번째 집적회로 구조를 위한 PE의 디자인을 보여주고 있다. PE의 논리는 DRAM 열 디코더 논리 대신에 128 b DRAM 열로 집적화되어 있다. 세 개의 레지스터인 A, B, C는 함수발생기에 입력이 된다. 제어신호  $f_1, f_2, \dots, f_6$ 는 256의 3입력 '불'논리연산 기능을 지정하며 레지스터 A 또한 이웃한 PE들에게 입력이 된다. PE들로부터 좌, 우, 상, 하로 전달된 값들은 통제신호  $f_L, f_R, f_U, f_D$ 에 있어서 함수발생기 결과와 합쳐진다. 최종결과는 레지스터 B, C, D, AND/OR E로 적재된다.(함수발생기, AND gate, 그리고 OR gate는 논리함수를 보여주기 위해 별도의 소자들로서 그려져 있으며 실제 구현은 이러한 소자들을 단일 동적 논리 회로로 통합하는 것이다.) 레지스터 E에 저장된 값들은 기록 구동기를 제어한다. 기록 구동기가 인에이블이면 PE는 액티브(active)하다고 말하며, 기록

픽셀-병렬 영상처리에 있어서 포맷 컨버터 설계에 관한 연구

동작에 의해 값들은 레지스터 D에서 메모리로 저장된다. 만일 기록 구동기를 사용할 수 없게 되면 PE 메모리의 내용들은 전 상태를 유지한다.

PE 배열을 위한 명령들은 메모리 동작, '불'논리연산기능( $f_1, f_2, \dots, f_6$ ), 네트워크 기능( $f_L, f_R, f_D, f_U$ ), 그리고 레지스터 적재 신호(LA, LB, LC, LD, LF)을 지정하게 된다. 명령의 실행은 메모리 동작의 시작과 함께 시작된다. 어느 정도 메모리 동작을 통해 판독 값과 기록 값은 레지스터 A로 옮겨지며 그렇게 되면 지정된 '불'논리연산과 네트워크 기능에 의해 확인된 로직 동작이 실행된다. 그리고 이 로직 동작의 결과는 레지스터 B, C, D, AND/OR E로 적재된다.

2.2 포맷 컨버터 디자인

MDA(multidimensional access) 메모리 저장 패턴은 픽셀 비트들이 메모리 버퍼에 저장되기 전에 그것들을 재배열하고 처리된 데이터가 PE 어레이에 전송되기 전에 그것들을 재배열할 수 있는 데이터 혼합기(data shuffler)들의 구현을 필요로 한다. 그것은 각각의 메모리 칩에 대하여 여러 가지 다른 메모리 어드레스 레지스터(MARs)를 필요로 한다. 그림 6은 데이터를 PE 어레이에 보낼 수 있는 포맷 컨버터의 모듈을 설명한다. 디지털화된 영상 데이터를 입력으로 하고, 어레이에 대한 전송을 위해 적절한 단위로 묶여져 포맷된 데이터를 생성한다. 이 포맷 컨버터를 위해 혼합기와 MAR 두 가지의 디자인을 한다. 또 다른 포맷 컨버터에서 처리된 데이터를 디스플레이에 보내는데 이용되는 역행 처리과정을 설명한다.

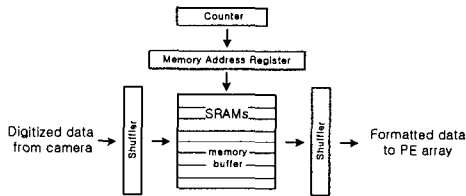


그림 6. 포맷 컨버터의 모듈

그림 7의 저장 패턴이 보여주는 것처럼 데이터는 메모리 칩에 순서적으로 저장되지 않는다. 각각의 칩에 대해서는 각기 다른 연속성을 가진다. 영상 데이터가 픽셀로 전달아서 도달되는 것처럼 단지 칩 0만이 위치 0으로부터 위치 7에 까지 그 메모리 위치들의 연속적인 어드레스를 유지한다. 칩 1의 경우 메모리 위치 1이 우선 어드레스되고 위치 0이 그 뒤를 따르며 그리고 연속적으로 위치 3과 위치 2가 뒤따른다. 다시 말해서, 그 어드레스는 '점프'한다. 두 개의 메모리 위치마다의 순서가 바뀐다. 칩 2의 경우 그것은 바뀐 두 개의 연속적인 메모리 위치로 이루어진 두 세트의 순서를 가진다.

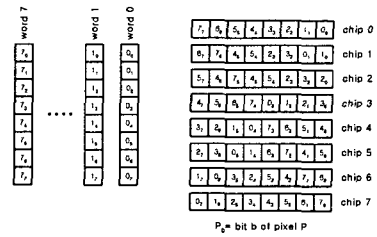


그림 7. 8x8 MDA 메모리 저장 패턴

이러한 '점프' 알고리즘은 계속되어서 픽셀 0에서 픽셀 7에 있는 비트들은 어드레스의 역행 순서로 칩 7에 저장된다. 바로 이런 어드레스 구성을 수행할 수 있는 한가지 간단한 방식은 카운터의 활용이다. 특히, 카운터 출력들은 각각의 MAR에 따라 바뀌며 그림 8과 같다.

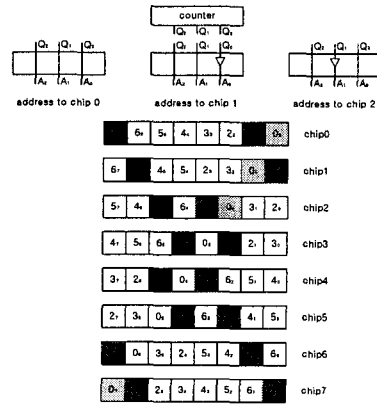


그림 8. 데이터 저장을 위한 MARs

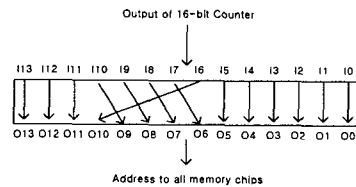


그림 9. 데이터를 액세스하는데 필요한 MAR

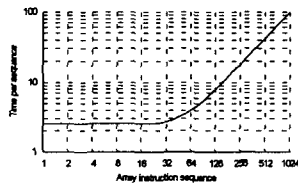
메모리 버퍼로부터 나온 데이터의 처리는 8x1 영상의 경우에, 동일한 비트-플레인 에 속하는 모든 비트들은 동일한 메모리 위치로부터 처리된다. 그러므로, 단지 하나의 어드레스 순서만이 모든 칩을 어드레스 할 필요가 있고, 어드레스 순서는 연속적이다. 특히, 비트-플레인 0에 있는 모든 비트들은 위치 0에 있고 비트-플레인 1은 위치 1에 있으며 나머지도 계속 이어진다. 마찬가지로 단지 하나의 어드레스만이 256x256 픽셀에 필요한 메모리 버퍼로부터 나온 데이터를 처리하는

데 필요하다. 그러나, 메모리 위치를 순차적으로 어드레스 할 수는 없다. 이것은 고-밀도 병렬 프로세서에 있는 SAMs의 디자인 때문이다. 이들 각자의 SAMs들은 128개의 레지스터를 포함함으로써 비트의 재배열은 그림 9와 같다.

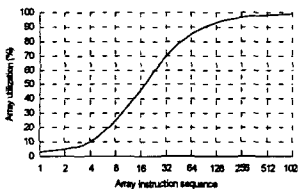
### III. 실험 및 고찰

Wire-Wrap VMEbus panel을 사용하는 기본 컨트롤러를 설계하였다. 이 기본 컨트롤러는 16 b 마이크로프로그램 순차, 4개의 메모리 모듈, 버스 인터페이스의 구성요소, 그리고 많은 PLD(programmable logic device)와 레지스터 칩을 이용한다. 이러한 실험을 함으로써 시스템 동작의 특징을 알고 이 디자인의 고밀도 패키지에 대한 유동성에 초점을 맞추었다. 프로그램이 가능한 논리 장치와 레지스터 칩을 게이트 배열로 대체함으로써 훨씬 더 밀도 있는 구현이 이루어졌다. 컨트롤러는 PE 장치의 10MHz에서 최적으로 기능을 수행하게 된다.

제어 패스 디자인의 실행을 위해 10MHz에서의 1 명령에서 1024까지의 명령의 길이를 가진 순차에 대한 제어 패스의 실행을 시뮬레이션하였다.



(a) 순차의 길이와 순차 호출시간



(b) 순차의 길이와 어레이의 관계  
그림 20. 제어 패스 성능 평가

그림 20은 시뮬레이션 결과를 보여주고 있다. 그림 20의 (a)는 호스트 컴퓨터에서부터 컨트롤러까지 시작 주소를 전송하는데 필요한 시간의 양을 보여준다. 그림에서 곡선이 굽곡되는 부분을 보면 짧은 순차에서는 순차가 호스트 컴퓨터에 의해 호출할 때마다 약 2.5μs로 통과하지만 긴 순차에서는 각 순차를 이동시키는데 걸리는 시간이 시작 주소를 전송하는데 필요한 시간을 초과한다. 따라서 실행시간은 순차의 길이에 따라 달라짐을 알 수 있다.

(b)는 순차의 길이와 어레이의 활용과의 관계를 보여준다. 만약 순차가 짧으면 컨트롤러는 다음 시작 주

소를 받기 전에 각 명령들의 순차를 어레이로 전달하는 것을 끝낸다. 그 결과 어레이 이용율은 낮다. 만일 순차가 길면 컨트롤러는 각 순차의 전달이 끝나기 전에 다음 시작 주소를 받는다. 그렇기 때문에 어레이의 이용은 각 순차를 시작하기 위해 컨트롤러가 필요로 하는 외부의 클럭 주기에 의해서만 제한을 받는다. 따라서 50명령보다 긴 순차들이 효율적으로 실행된다.

### V. 결론

영상 데이터는 A/D 컨버터에 의해 한 픽셀씩 교대로 출력된다. 그러나 데이터를 PE 어레이에 신속히 전송하고, 그 어레이를 효율적으로 이용하기 위해서는 데이터를 재배열할 필요가 있다. 포맷 컨버터를 포함하는 하드웨어는 카메라로부터 얻은 실-시간 영상은 물론이고, 호스트 컴퓨터로부터 얻은 영상 데이터를 처리할 수 있을 만큼 편리함으로 보여준다.

PE 어레이가 각 라인마다 단지 256픽셀을 처리하기 때문에 만일 디지털 데이터를 출력하는데 있어서 픽셀당 200ns로 각기 다른 샘플링 비율을 지닌 다른 A/D 컨버터가 사용된다면 데이터 패스 보드는 20MHz(50ns)에서 동작한다. 따라서 각각의 메모리 버퍼를 구현하는데 필요한 SRAMs의 수는 절반으로 줄일 수 있다. 각 포맷 컨버터마다 2개의 메모리 버퍼가 필요한 것임에도 불구하고 보드상의 SRAM의 총 개수는 변함이 없다. 1개의 픽셀을 저장하는데 다양한 판독과 기록 주기가 요구되므로 훨씬 더 풍부한 단어를 지닌 SRAM이 메모리 버퍼를 구현하는데 사용된다. 이렇게 해서 하드웨어상의 SRAM 칩수를 줄일 수 있다.

본 논문에서는 실시간 영상 처리를 실현하기 위한 포맷 컨버터 설계와 구현 방법을 제시하였다. 데이터는 메모리 버퍼에 일시적으로 저장되고 수직 블랭킹 주기동안 PE 어레이에서 PE 어레이로 전송된다. 각 포맷 컨버터에 두 개의 메모리 버퍼를 구현하여 전체적인 디자인을 개선시켰다.

### 참고문헌(또는 Reference)

- [1] Jeffrey C. Gealow, Frederick P. Herrmann, Lawrence T. Hsu, Charles G. Sodini, "System Design for Pixel-Parallel Image Processing, IEEE Transactions on VLSI Systems, Vol. 4, No. 1, pp. 32-41, Mar. 1996.
- [2] Yoshiki Tsujihashi, Hisashi Matsumoto, Hidekatsu Nishimaki, Atsushi Miyamishi, Hiroomi Nakao, Osamu Kitada, Shuuhei Iwade, Shinpei Kayano, and Masayoshi Sakao, "A High-Density Data-Path Generator with Stretchable Cells" IEEE J. Solid-State Circuits, Vol. 29, No. 1, pp. 2-8, Jan. 1994.