포맷 변환기를 이용한 화소-병렬 영상처리에 관한 연구

°김현기*, 이용희**, 이천희***
극동정보대학 전자통신과*, 하이닉스 반도체**, 청주대학교 전자공학과***

A Study on the Pixel-Parallel Image Processing Using the Format Converter.

Kim Hyun-Gi*, Lee Young-Hee**, Yi Cheon-Hee***
Keukdong College*, Hynix Semiconductor**,Chongju University***

요 약

본 논문에서는 포맷 변환기를 사용하여 여러 가지 영상처리 필터링을 구현하였다. 이러한 설계 기법은 집적회로를 이용한 대규모 화소처리배열을 근거로 하여 실현하였다. 집적구조의 두가지 형태는 연산병렬프로세서와 병렬 프로세스 DRAM(또는 SRAM) 셀로 분류할 수 있다. 이러한 포맷 변환기 설계는 효율적인 제어 경로 수행을 능력을 가지고 있으며 하드웨어를 복잡하게 할 필요 없이 고급 기술로 사용 될 수 있다. 실험 결과 1)단순한 평활화는 더 높은 공간의 주파수를 억제하면서 잡음을 감소시킬 뿐 아니라 에지를 흐리게 할 수 있으며, 2) 평활화와 분할 과정은 날카로운 에지를 보존하면서 잡음을 감소시키고, 3) 평활화와 분할과 같은 메디안 필터링기법은 영상 잡음을 줄이기 위해 적용될 수 있고 날카로운 에지는 유지하면서 스파이크 성분을 제거하고 화소 값에서 단조로운 변화를 유지 할 수 있었다.

1. 서론

현대의 VLSI 기술은 고밀도로 배열된 PE(Processing Elements)에서 메모리와 프로세서들로 집적화되어 있으며 이들 PE 배열들은 화소-병렬 영상 처리 시스템을 위한 기본 형태가 된다. 각각의 PE는 하나의 영상을 갖는 한 화소(pixel)를 저장하고 처리한다.[1] 그림 1은 화소-병렬 영상 처리 시스템의 구성도를 보여주고 있는데 이것은 한 개의 PE 배열, 컨트롤러, 주 컴퓨터와 두 개의 포맷 변환기로 구성되어 있다. 카메라로부터 출력되는 아날로그 신호는 ADC(Analog-to-Digital Converter)를 거쳐 디지털 신호로 변환되며 이때 PE에서 처리하기 위한 신호로 포맷이 되어 PE 배열에 옮겨진다. 주 컴퓨터로부터 나온 명령들은 컨트롤러를 경유해서 모든 PE에 전달되며 처리된 데이터들은 다음의 처리를 위해 재포맷 된다.

PE 배열에 필요한 두 개의 집적 회로 구조는 내용을 어드레스 가능 메모리 셀에 사용하는 연산 병렬 프로세서와 종래의 DRAM 셀들을 사용하는 미세하게-결정(結晶)된 병렬 구조가 있다. 이 구조에서 각각의 PE에 필요한 논리회로는 동일한 칩의 DRAM과 결합됨으로서 행 디코더에 필요한 것을 제거해준다. 이런 식으로 고밀도와 소형의 메모리 셀 크기를 가진 구조를 구현 할 수 있다.

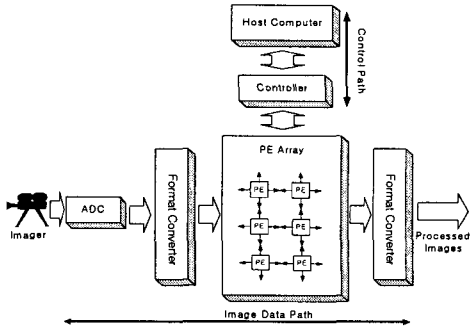


그림 1. PE 배열을 사용한 영상 처리 시스템

이 시스템은 주 컴퓨터로부터 나오는 명령을 구현할 수 있으나 실시간 영상 데이터가 시스템에 들어가고 나오는 경로는 결여되어 있다. 만일 ADC로부터 나오는 영상 데이터 출력이 화소를 기초로 한 PE 배열에 직접 전송되는 경우 단지 하나의 PE만이 전송이 가능할 것이다. 따라서 좀 더 효율적인 PE 배열을 이용하기 위해서는 데이터가 병렬 배열로 전송될 수 있도록 영상 데이터의 포맷변환이 필요하다. 본 논문엣는 그림1에서 나타낸것과 같이 비디오 카메라등의 아날로그 영상을 디지털 영상으로 바꾸어 PE 배열에서 처리하는 입·출력 포맷 변환장치를 설계 하여 여러 가지 필터링을 통하여 영상을 처리하는 과정을 제시하였다. 시스템 환경은 SUN UNIX OS 상에서 VHDL을 사용하여 Synopsis 툴을 이용하여 합성하였고, Xilnix FPGA 칩으로 시뮬레이션 하였다.

2. 화소-병렬 영상처리 시스템

2.1 PE(Processing Element) 배열

낮은 비용으로 대규모 PE 배열을 만들기 위해서는 고밀도의 PE 구현이 이루어져야 한다. 따라서 적합한 집적회로 구조를 개발하기 위해 먼저 필요한 것이 순차주소 메모리 셀을 이용한 새로운 연산 병렬 프로세서 장치의 개발과 DRAM 셀을 사용하는 최적화된 병렬 구조를 사용하는 것이다. 전자는 각각의 PE에 필요한 논리의 양을 최소화하는 방법이며 후자는 메모리 셀의 크기를 최소화하는 것이다. 이 두 설계방법에서 1비트 크기의 PE 논리회로의 레이아웃 피치는 메모리 셀의 피치와 동일하게 하고 메모리와 논리회로사이의 대역폭은 최대화하고 PE의 범위는 최소화한다. 연산 병렬 프로세서 장치는 이중 폴리실리콘 (Double- polysilicon) CCD-CMOS 기술로 만들어져 왔으며 매우 편리한 것으로 입증되고 있으나 DRAM 셀을 이용한 구조의 구현은 아직 완전하지 않다.[2]

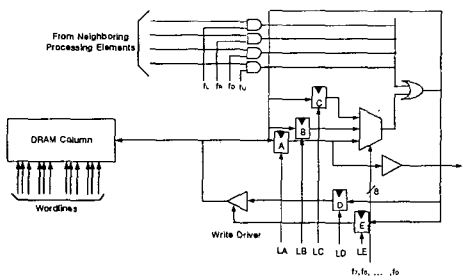


그림 2. DRAM 셀을 사용한 처리소자

그림 2는 두번째 집적회로 구조를 위한 PE의 설계방법을 보여주고 있다. PE의 논리는 DRAM 열 디코더 논리 대신에 128b DRAM 열로 집적화되어 있다. 세 개의 레지스터인 A, B, C는 함수발생기에 입력이 된다. 제어신호 f_7, f_6,....., f_0는 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_D, f_U에 의해서 함수발생기 결과와 합쳐진다. 최종결과는 레지스터 B, C, D, AND/OR E로 적재된다. 레지스터 E에 저장된 값들은 기록 구동기를 제어한다. 기록 구동기가 인에이블이면 PE는 *액티브(active)*하다고 말하며 기록 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_7, f_6,....., f_0), 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 기록값은 레지스터 A로 옮겨지며 그렇게 되면 지정된 '불'논리연산과 네트워크 기능에 의해 확인된 논리 동작이 실행된다. 그리고 이 논리 동작의 결과는 레지스터 B, C, D, AND/OR E로 적재된다.

2.2 포맷 변환기 설계

MDA(Multi-Dimensional Access) 메모리 저장 형식은 화소 비트들이 메모리 버퍼에 저장되기 전에 그것들을 재배열하고 처리된 데이터가 PE 배열에 전송되기 전에 그것들을 재배열할 수 있는 데이터 혼합기(Data Shuffler)들의 구현을 필요로 한다. 그것은 각 각의 메모리 칩에 대하여 여러 가지 다른 메모리 어드레스 레지스터(MARs)를 필요로 한다. 그림 3은 데이터를 PE 배열에 보낼 수 있는 포맷 변환기의 모듈을 보여주고 있다. 디지털화된 영상 데이터를 입력으로 하고 배열에 대한 전송을 위해서 적절한 단위로 묶여져 포맷된 데이터를 생성한다. 이 포맷 변환기를 위해 혼합기와 MAR 두 가지를 설계하여야 한다. 또한 다른 포맷 변환기에서 처리된 데이터를 디스플레이에 보내는데 이용되는 역행 처리과정을 이해하여야 한다.

그림 4의 저장 패턴이 보여주는 것처럼 데이터는 메모리 칩에 순서적으로 저장되지 않는다. 각각의 칩에 대해서는 각기 다른 연속성이 필요하다. 영상 데이터가 화소로 연달아서 도달되는 것처럼 단지 칩 0만이 위치 0(location 0)으로부터 위치 7(location 7)에 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. "즉" 해당 어드레스는 '점프'하며 두 개의 메모리 위치마다 순서가 바뀐다. 칩 2의 경우 바뀐 두 개의 연속적인 메모리 위치로 이루어진 두 세트의 순서가 된다

메모리 버퍼로부터 나온 데이터를 처리하는 8×1 영상의 경우에 동일한 비트-평면에 속하는 모든 비트들은 동일한 메모리 위치로부터 처리된다. 특히 비트-평면 0에 있는 모든 비트들은 위치 0에 있고 비트-평면 1은 위치 1

에 있으며 나머지도 동일하다. 마찬가지로 단지 하나의 어드레스만이 256×256 화소에 필요한 메모리 버퍼로부터 나온 데이터를 처리하는데 필요하다. 하지만 메모리 위치를 순차적으로 어드레스 할 수는 없다. 이것은 고-밀도 병렬 프로세서에 있는 SAMs의 설계 및 구현 때문이다. 이들 각자의 SAMs들은 128개의 레지스터를 포함함으로써 각각의 영상 블록을 두 개의 세트로 나눈다. 그러므로 각각의 영상블록을 배열에 전송할 때 처음 16개의 열에서 모든 열들은 두 번째 세트의 상응하는 열보다 앞서야 하므로 이것은 어드레스 순서에서 '점프'를 필요로 한다. 비트의 재배열은 그림 5와 같다.

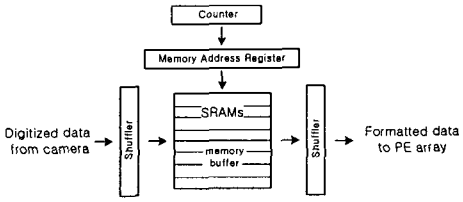


그림 3. 포맷 변환기의 기능적 모듈

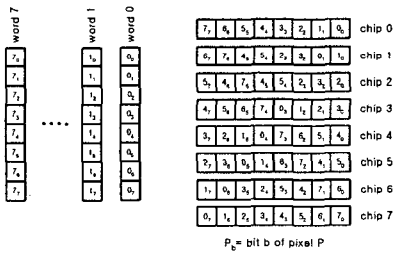


그림 4. 8×8 MDA 메모리 저장 패턴

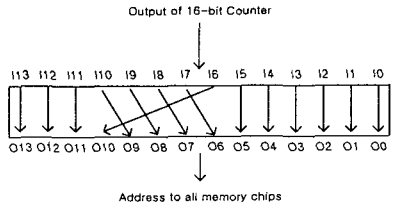
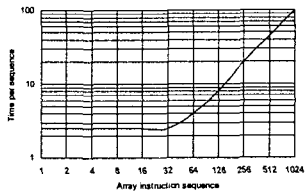


그림 5. 데이터를 액세스하는데 필요한 MAR의 기능 설명

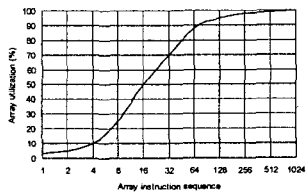
3. 실험 및 고찰

3.1 제어 패스 성능 평가

본 실험에서는 Wire-Wrap VMEbus panel을 사용하는 기본 컨트롤러를 구현하였다. 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 설계의 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 설계의 실행을 위해 10MHz에서의 1명령에서 1024까지의 명령의 길이를 가진 순차에 대한 제어 패스의 실행을 평가하였다.



(a)순차의 길이와 순차 호출시간



(b)순차의 길이와 배열의 관계

그림 6. 제어 패스 성능 평가

그림 6은 실험 결과를 보여주고 있다. 그림 19의 (a)는 호스트 컴퓨터에서부터 컨트롤러까지 시작 주소를 전송하는데 필요한 시간의 양을 보여준다. 그림에서 곡선이 굴곡되는 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 배열의 활용과의 관계를 보여준다. 만약 순차가 짧으면 컨트롤러는 다음 시작 주소를 받기 전에 각 명령들의 순차를 배열로 전달하는 것을 끝낸다. 그 결과 배열 이용율은 낮다. 만일 순차가 길면 컨트롤러는 각 순차의 전달이 끝나기 전에 다음 시작 주소를 받는다. 그렇기 때문에 배열의 이용은 각 순차를 시작하기 위해 컨트롤러가 필요로 하는 외부의 클록 주기에 의해서만 제한을 받는다. 따라서 30명령보다 긴 순차들이 효율적으로 실행된다.

3.2 화소-병렬 처리 결과

영상 획득시 생기는 잡음은 평활화 강도를 변화시킴으로써 감소시킬 수 있다.

3×3 커널 $\frac{1}{8}\begin{bmatrix}0&1&0\\1&4&1\\0&1&0\end{bmatrix}$을 지닌 영상을 반복적으로 발산하는 것은 가우스의 평활화 동작과 유사하다. 적용된 컨볼루션의 수는 가우스 필터의 편차를 결정한다.

그림 7에서 볼 수 있듯이 단순한 평활화는 더 높은 공간의 주파수를 억제하면서 잡음을 감소시킬 뿐 아니라 에

지를 무디게 한다. 평활화와 분할 과정은 날카로운 에지를 보존하면서 잡음을 감소시킨다. 각각의 컨볼루션 전에 각 픽셀의 값은 가장 가까이 이웃해 있는 4개의 픽셀의 값과 비교되어진다. 그 차이가 분할 임계치 때보다 더 크면 3×3 커널은 국부적으로 수정되어 강도의 변화를 보존하게 된다. 예를 들어, 어떤 화소의 값이 임계치 보다 더 많아짐으로써 이웃해 있는 화소의 값과 다르지만 다른 그 밖의 이웃 화소들과는 처음보다 적은 값을 갖게 되어 달라지는 경우,

수정된 3×3 평활화 커널 $\frac{1}{8}\begin{bmatrix}0&1&0\\1&5&0\\0&1&0\end{bmatrix}$이 적용될 것이다. 3×3 커널을 수정하고 적용하는 과정이 각각의 화소마다 실행되어야 하기 때문에 평활화와 분할작업은 화소-병렬 하드웨어에 자연스럽게 병합된다.

(a) 원래 영상

(b)평활화 영상 (c)평활화 및 분할 영상

(d)3×3 메디안 필터 (e)5×5 메디안 필터

그림 7. 필터링후의 여러 가지 영상처리결과

평활화와 분할과 같은 메디안 필터링(Median Filtering)은 영상 잡음을 줄이기 위해 적용된다. 각 출력 화소의 값은 그 출력 화소에 집중되는 부분의 모든 입력 값의 메디안이다. 메디안 필터링은 날카로운 에지는 유지하고 화소값의 단조로운 변화를 보존하는 반면 곡선의 스파이크는 제거한다. 그림 7(d),(e)는 메디안 필터링의 효과를 보여주고 있다. 3×3 메디안 필터에 있어서 각 출력 값은 3×3 픽셀 영역에 있는 9개의 입력 값의 메디안이고 5×5 메디안 필터에서의 각 출력 값은 25개의 입력 값의 메디안이다.

6. 결론

영상 데이터는 A/D 변환기에 의해 한 화소씩 교대로 출력된다. 그러나 데이터를 PE 배열에 신속히 전송하고 그 배열을 효율적으로 이용하기 위해서는 데이터를 재배열할 필요가 있다. 포맷 변환기를 포함하는 하드웨어는 카메라로부터 얻은 실-시간 영상은 물론이고 호스트 컴퓨터로부터 얻은 영상 데이터를 처리할 수 있을 만큼 편리함을 보여 줄 수 있다.

PE 배열이 각 라인마다 단지 256화소를 처리하기 때문에 만일 디지털 데이터를 출력하는데 있어서 화소당 200ns로 각기 다른 샘플링 비율을 지닌 다른 A/D 변환기가 사용된다면 데이터 패스 보드는 20MHz(50ns)에서 동작한다. 따라서 각각의 메모리 버퍼를 구현하는데 필요한 SRAMs의 수는 절반으로 줄일 수 있다. 각 포맷 변환기마다 2개의 메모리 버퍼가 필요한 것임에도 불구하고 보드상의 SRAM의 총 개수는 변함이 없다. 1개의 화소를 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 포맷 변화기를 이용하여 원래 영상을 평활화, 평활화 및 분할, 메디안 필터링을 하는 작업을 구현하였다. 단순한 평활화는 더 높은 공간의 주파수를 억제하면서 잡음을 감소시킬 뿐 아니라 에지를 무디게 할 수 있으며 평활화와 분할과 같은 메디안 필터링기법은 영상 잡음을 줄이기 위해 적용될 수 있고 날카로운 에지는 유지하면서 스파이크 성분을 제거하고 화소 값에서 단조로운 변화를 유지 할 수 있었다.

참고 문헌

[1] Jeffery C. Gealow, Frederick p. Herrmann, Lawrence T. Hsu, and Charles G. Sodini, "System Design for Pixel-parallel Image Processing.", IEEE Trans. on VLSI systems, pp. 32-41, Mar. 1996.

[2] F. P. Herrmann and C. G. Sodini, "A 256-element Associative Parallel Processing." in Symp, VLSI Circuits : Dig. Tech. Papers, pp. 99-100, June 1994.

[3] F. P. Herrmann and C. G. Sodini, "A Dynamic Associative Processor for Machine Vision Applications." IEEE Micro, vol.12, no.3, pp. 31-41, June 1992.

[4] 조화현, 최절호, 권병헌, 최명렬, "실시간 처리를 위한 콘트라스트 조정기법," 한국정보처리학회논문지 제7권 제6호, pp. 1988-1995, June. 2000.

[5] Jonathan Rose, Abbas El Gamal, and Alberto Sangiovanni-Vincentelli,"Architecture of Field-Programmable Gate Arrays," Proc. of the IEEE, vol. 81, no. 7, pp. 1013-1029, July 1993.