論文97-34C-10-1

효율적인 실시간 영상처리용 2-D 컨볼루션 필터 칩

(An Efficient 2-D Convolver Chip for Real-Time Image Processing)

殷 世 英 *, 鮮 于 明 勳 *

(Se Young Eun and Myung Hoon Sunwoo)

요 약

본 논문은 곱셈기가 없는 새로운 구조의 영상처리용 디지탈 필터를 제안한다. 2차원 컨볼루션 필터링은 화질개선의 방법으로 가장 많이 사용된다. 현재 상용화된 2차원 컨볼루션 필터칩은 영상 데이타의 실시간 처리를 만족하기 위해서 다수의 곱셈기를 사용한다. 따라서 칩의 크기가 크며 처리속도 대 크기의 비율이 낮은 단점이 있다. 본 논문에서 제안하는 실시간 영상처리용 컨볼루션 필터칩은 곱셈기를 사용하지 않는 대신 하나의 shift-and-accumulator만을 사용함으로써 기존의 상용 칩에 비해 크기를 70% 이상 줄일 수 있다. 또한 2차원 영상 신호 뿐만 아니라 1차원 신호도 처리할 수 있도록 설계하였다. 제안된 필터 칩은 VHDL로 모델링한 후 SYNOPSYS™ CAD tool로 논리합성하여 구현하였다. 논리합성은 삼성 SOG 셀 라이브러리를 사용하였으며 타이밍 시뮬레이션을 수행하였다. 구현된 필터는 3,893개의 게이트로 구성되었으며 최대 동작 주파수는 125 MHz이고 MPEG-2 영상 규격인 720×480 크기 영상의 실시간 처리 조건(10.4 Mpixel/second)을 만족한다.

Abstract

This paper proposes a new real-time 2-D convolver filter architecture without using any multiplier. To meet the massive amount of computations for real-time image processing, several commercial 2-D convolver chips have many multipliers occupying large VLSI area. The proposed architecture using only one shift-and-accumulator can reduce the chip size by more than 70 % of commercial 2-D convolver filter chips and can meet the real-time image processing requirement, i.e., the standard of CCIR601. In addition, the proposed chip can be used for not only 2-D image processing but also 1-D signal processing and has good scalability for higher speed applications. We have simulated the architecture by using VHDL models and have performed logic synthesis. We used the Samsung SOG cell library(KG60K) and verified completely function and timing simulations. The implemented filter chip consists of only 3,893 gates, operates at 125 MHz and can meet the real-time image processing requirement, that is, 720×480 pixels per frame and 30 frames per second(10.4 Mpixels/second).

* 正會員, 亞州大學校 電機電子工學府

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

※ 이 연구는 1995년도 한국과학재단 연구비 지원에 의한 결과임.(과제번호 : 951-0915-124-2)

接受日字:1997年4月8日, 수정완료일:1997年9月23日

I. 서 론

영상 신호의 필터링은 화질 개선(image enhancement)의 가장 기본적인 수단으로서 널리 사용되고 있다[1-3]. 그러나 영상 신호는 데이타 속도와 연산량이 방대하여 실시간 필터링을 위해서는 고속의 필터가 필수적이다. 상용의 컨볼루션 필터의 경우 필터링 마

스크의 크기에 비례한 다수의 곱셈기가 사용되며[4] 이러한 곱셈기는 필터의 VLSI 구현시 칩(chip)의 면적을 많이 차지하므로 하드웨어 비용이 증가된다[4-6], 따라서 필터의 응용분야에서 요구되는 처리속도를 감안하여 하드웨어 비용을 줄일 수 있는 구조로서 곱셈기를 사용하지 않는 필터의 설계가 제안되어왔다[7-11]. 이러한 아키텍쳐는 일반적으로 power-of-two 계수 알고리즘을 바탕으로 하고 있으며 특히 특성 계수가 결정된 특별용도의 필터의 경우에는 별도의 SA(shift-and-accumulator)를 필요하지 않으므로 쉽게 처리속도 대 하드웨어 비용이 우수한 필터를 설계할 수 있다[7-9].

특성 계수를 제어할 수 있는 프로그래머블 필터를 설계시 SA는 필수적이며 SA를 이용한 효과적인 필터 아키텍쳐가 제안되어 왔다[10-11]. 그러나 제안된 방식의 아키텍쳐는 곱셈기 대신 SA를 모든 필터링 마스크나 탭에 사용하므로 하드웨어 비용 감소를 크게 기대할 수 없다.

본 논문에서는 곱셈기를 사용하지 않으며 하나의 SA를 사용하므로써 하드웨어 비용을 최소화할 수 있는 컨볼루션 필터 칩의 아키텍쳐를 제안하고 구현하였다. 구현된 필터 칩은 MPEG-2 표준안(CCIR601)인 초당 30 프레임, 720× 480의 영상 데이타(약 10.4 Mpixel/sec)[12] 를 실시간 필터링 할 수 있다. 제안된 필터는 VHDL(VHSIC Hardware Description Language)로 모델링하여 논리 합성하였으며 삼성 SOG(Sea-of-Gate) 셀을 사용하여 그 기능과 타이밍 시뮬레이션을 확인하였다. 최고 동작 주파수는 125 MHz로서 15.625 Mpixel/sec의 전송율을 가지는 영상 데이타를 실시간 처리할 수 있다. 전체 게이트 수는 3,893개이며 현재 상용화된 HARRIS semiconductor사의 3× 3 마스크 영상 필터 칩인 HSP 48901[4] 과 비교했을 때 약 70% 이상 게이트 수가 감소하였다. 보다 효과적인 비교로서 처리속도 대 크기의 비율(speed/size ratio)은 약 2배 증가하였다.

본 논문은 다음과 같이 구성되어있다. Ⅱ절에서는 설계된 필터의 수식적인 알고리즘과 아키텍쳐를 설명하고, Ⅲ절에서 제안한 필터의 설계에 대해 설명하며, Ⅳ절에서 곱셈기가 있는 필터 및 기존의 SA를 사용한 필터와의 비교 및 성능 평가를 서술하고 마지막으로 Ⅴ절에서는 연구에서 얻은 성과 및 개선방안을 서술한다.

Ⅱ. 설계된 필터의 알고리즘과 아키텍쳐

윈도우 마스크의 크기가 NxM인 영상 필터의 입력과 출력의 수열이 각각 $F(x,y)$, $G(x,y)$이고 마스크의 계수(mask coefficient)가 $H(x,y)$이면 컨볼루션 필터의 입력과 출력의 관계는 식 (1)로 주어진다. 결국 필터를 하드웨어로 구현하면 그 연산부는 각각 NxM개의 데이타와 계수의 곱의 합(sum of products)을 구하는 구조가 됨을 알 수 있다. 식 (1)에서 x, y는 영상의 좌표를 표시하는 정수이다.

$$G(x,y)=\sum_{n=0}^{N}\sum_{m=0}^{M}H(n,m)F(x-n,y-m) \qquad (1)$$

본 논문에서 제안하는 필터를 설명하기 위해 8 비트 데이타와 계수에 대한 3x3 마스크를 갖는 필터로 가정하면 식 (1)은 식 (2)와 같다. 식 (2)를 직접 구현하면 그림 1과 같은 구조를 가지며 8 비트 데이타 레지스터(F) 9개와 8 비트 계수 레지스터(H) 9개의 출력을 각각 쌍으로 9개의 곱셈기에서 곱을 구하고 곱셈기의 16 비트 결과를 8개의 16 비트 가산기로 구성된 16 비트 트리형 가산기에서 모두 더함으로써 하나의 필터링 출력 샘플을 구한다. 이와 같은 구조는 9개의 곱셈기가 이미 언급한 바와 같이 VLSI 구현시 연산 구조의 대부분의 면적을 차지하는 구조로서 데이타처리 속도대 게이트 수 면에서 효과적이지 못하다.

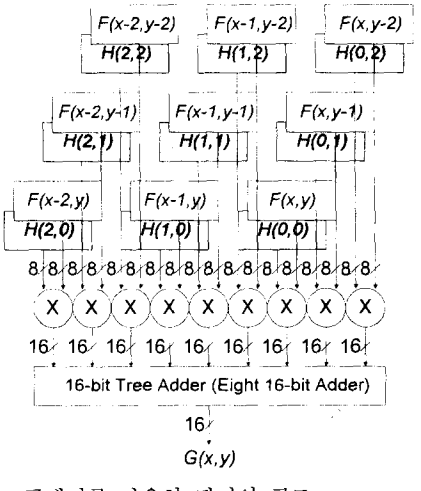

그림 1. 곱셈기를 사용한 필터의 구조
Fig. 1. The filter architecture using multipliers.

$$G(x,y)=\sum_{n=0}^{2}\sum_{m=0}^{2}H(n,m)F(x-n,y-m) \qquad (2)$$

곱셈기대신 SA를 이용한 기존의 필터의 구조[10-11

]를 본 논문에서 제안하는 필터와 비교하기 위해서 식 (2)를 이진수 체계로 표현하면 식 (3)과 같고 식 (3)에서 $H(n,m)=h_k(n,m)2^k$이며 $h_k(n,m)$는 0또는 1인 이진수이다. 식 (3)을 하드웨어로 구현하면 그림 2와 같다.그림 2에서 계수 $H(n,m)$를 저장하는 레지스터는 쉬프트 로테이트 레지스터로서 하나의 클럭 사이클에 한 비트씩 쉬프트하면서 계수의 각 비트를 MSB부터 순차적으로 출력함으로써 논리곱 게이트가 데이타와 계수의 모든 부분곱을 출력 하도록 한다. 하나의 SA는 16비트 레지스터와 16비트 가산기로 구성되며 한 쌍의 데이타와 계수의 부분곱들을 쉬프트와 누적을 반복하여 8클럭 사이클 동안에 곱셈을 완성한다. 이와 같은 9개의 SA의 출력을 하단의 16 비트 트리형 가산기가 모두 더함으로써 하나의 필터링 출력 샘플을 구하게 된다. 이와 같은 구조는 그림 1과 같이 곱셈기를 사용하는 구조에 비하여 속도면에서는 단점이 있으나 하드웨어 비용이 적게 드는 장점이 있다. 그러나 9개의 8x8 곱셈기를 72개의 논리곱 게이트, 9개의 16비트 가산기, 9개의 accumulate 레지스터로 대치한는 것은 실제로 하드웨어 비용의 큰 감소를 가져오기 힘들다.

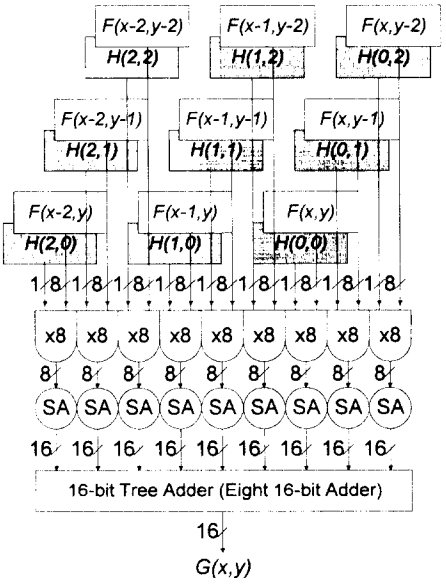

그림 2. 기존의 SA를 이용한 필터의 구조
Fig. 2. The filter architecture using SAs.

$$G(x,y) = \sum_{n=0}^{2}\sum_{m=0}^{2}\sum_{k=0}^{7} F(x-n,y-m)\ h_k(n,m)2^k \qquad (3)$$

본 논문에서 제안하는 필터의 아키텍쳐는 식 (4)를 구현한 것으로 식 (4)는 식 (3)을 분배법칙과 결합법칙을 수행하여 구할 수 있으며 식을 적용한 구조는 그림 3과 같다.

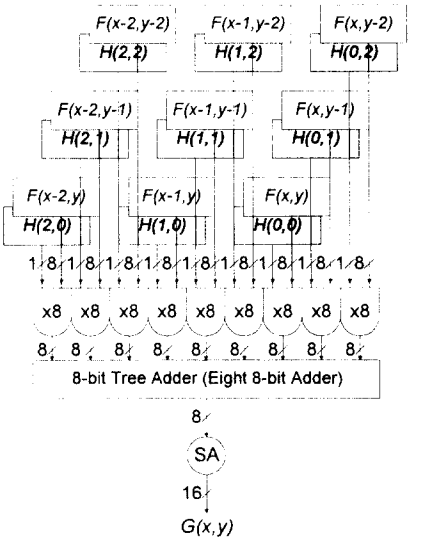

그림 3. 제안하는 필터의 구조
Fig. 3. The proposed filter architecture.

그림 2의 구조와 비교하면 각 데이타와 계수의 부분곱을 쉬프트 누적하여 곱을 구하기 전에 다른 데이타와 계수의 부분곱들을 모두 더한 후 SA에서 그 결과를 쉬프트하면서 누적하는 구조이다. 이렇게 함으로써 그림 1의 구조와 비교하였을 때에는 9개의 8x8 곱셈기와 16 비트 트리형 가산기 대신 하나의 SA와 8 비트 트리형 가산기를 사용할 수 있으며 그림 2의 구조와 비교하였을 때에는 9개의 SA와 16 비트 트리형 가산기 대신 1개의 SA와 8 비트 트리형 가산기를 사용할 수 있다. 따라서 하드웨어 비용을 그림 1의 구조와 그림 2의 구조에 비교하여 크게 감소 시킬 수 있음을 알 수 있다. 또한 그림 2의 구조는 SA가 8 클럭 사이클동안 누적한 후 트리형 가산기에 유효한 데이타를 줄 수 있지만 그림 3의 구조에서는 부분곱을 논리곱 게이트에서 구한 후 바로 트리형 가산기에 유효한 데이타를 주므로 트리형 가산기에 파이프라인 아키텍처를 적용하기에 용이하다.

$$G(x,y) = \sum_{k=0}^{7}\sum_{n=0}^{2}\sum_{m=0}^{2} F(x-n,y-m)\ h_k(n,m)2^k \qquad (4)$$

Ⅲ. 제안한 필터의 설계

본 논문에서 제안하는 필터는3x3 마스크나 9개의

탭을 가지는 2차원 및 1차원 겸용 필터이다. 연산부에는 각 계수 레지스터의 MSB와 각 데이타들을 논리곱한 9개의 부분곱을 더하는 트리 가산기가 있으며 트리 가산기의 출력을 계수 레지스터가 로테이트하여 MSB가 본래 값의 LSB가 되는 8클럭 동안 레프트 쉬프트 시키면서 더해주는 기능을 하는 16비트 가산기와 누적 레지스터가 있다.

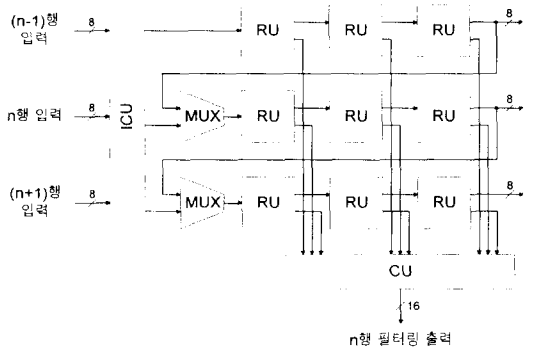

그림 4. 설계된 필터 칩의 전체 아키텍처
Fig. 4. The the designed filter chip.

그림 4는 제안된 필터를 구현하기 위해 구성한 아키텍처로서 9개의 레지스터 유니트(RU), 연산 유니트(CU), 입력 제어 유니트(ICU)의 부블럭으로 구성된다. 먹스(Mux)와 디먹스(Demux)를 통해 1차원 신호와 2차원 신호에 맞게 데이타와 계수의 흐름을 조정한다. 따라서 1차원 신호의 필터링을 할 때에는 9개의 레지스터 유니트가 직렬로 연결된 9 탭으로 사용되며 2차원 영상 신호일때에는 9개의 레지스터 유니트가 3×3의 계수 마스크와 데이타 윈도우로 사용된다. 레지스터 유니트들에서 출력된 9개의 부분곱을 입력 받은 연산 유니트는 모두 더한후 쉬프트 누적함으로써 여덟 클럭에 한번씩 필터링된 데이타를 출력한다.

레지스터 유니트의 구조는 그림 5와 같이 8비트의 데이타 레지스터와 계수 레지스터로 구성되어 있다. 각 레지스터 유니트의 출력은 데이타와 계수 한 비트의 부분곱이며 첫번째 클럭에 데이타와 계수 MSB의 부분곱을 출력하기 시작하여 마지막 여덟번째 클럭에 데이타와 계수 LSB의 부분곱을 출력한다. 계수 레지스터는 외부에서 인가되는 클럭(시스템 클럭)을 사용하며 데이타는 각 계수의 모든 비트와 한 번씩 부분곱셈을 해야하므로 시스템 클럭에서 주기가 8배로 분주된 클럭(데이타 클럭)에 동기되어 다음단의 레지스터 유니트로 이동된다.

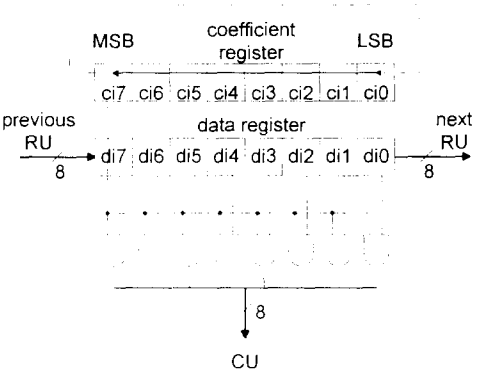

그림 5. 레지스터 유니트의 구조
Fig. 5. The Register Unit.

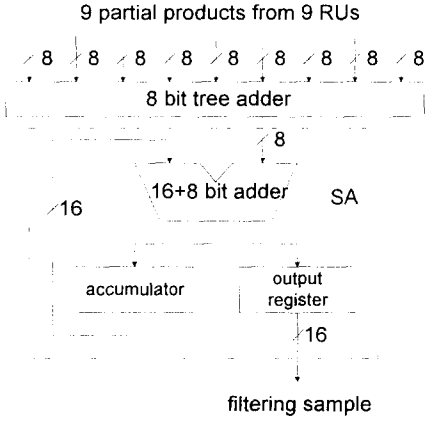

그림 6. 연산 유니트의 구조
Fig. 6. The Computation Unit.

연산 유니트의 구조는 그림 6과 같다. 8 비트 트리 가산기(tree adder)는 8개의 8비트 가산기가 3단으로 파이프라인되어 구성되며, 8비트 가산기는 2개의 4비트 CLA(carry look ahead) 가산기를 연결한 그룹 CLA이다. 16+8비트 가산기(16+8 bit adder)는 8비트 CLA 전 가산기(full adder)와 8비트 반 가산기(half adder)를 연결한 그룹 CLA이다. 따라서 전체 필터의 지연시간(delay time)은 비트수가 가장 큰 16+8비트 가산기에 의해 결정된다. 또 16+8비트 가산기의 출력은 누적 레지스터(acc register)를 거쳐 다시 16+8비트 가산기의 입력으로 궤환(feedback)되어 입력는데 누적 레지스터를 쉬프트 레지스터로 쓰는 대신 누적 레지스터의 출력을 가산기 입력에 직접 1비트씩 상위 비트로 쉬프트시켜 연결하여 곱셈 기능을 수행하며 이는 쉬프트에 소요되는 한 클럭을 단축시킨다. 따라서 16+8비트 가산기의 현재의 출력은 과거 출력의 2배(1비트 쉬프트)값에 현재 트리 가산기의 출력

값을 더한 값이 되며 이 과정은 계수 레지스터가 로테이트하여 다시 본래의 값을 갖는 8클럭을 주기로 반복 수행한다. 또한 한 주기의 마지막 클럭에서는 출력 레지스터(output register)는 가산기의 출력을 새로 저장하고 누적 레지스터는 다음 주기에 현 주기의 값을 넘겨 주지 않도록 초기화 된다.

컨볼루션 필터링에서는 한 행의 필터 출력 값을 구하기 위해 세개 행의 입력이 필요하며 따라서 두개의 과거행 데이타를 저장하는 행 버퍼가 필요하다. 이러한 행 버퍼는 칩의 크기와 확장성 문제로 하나의 칩에 집적하기에 어려우며 따라서 행 버퍼를 외부에 두거나 필터의 3개 행 입력을 메모리로 부터 모두 받는 방법을 사용할 수 있는데 메모리로 부터 3개 행의 데이타를 받는 경우 같은 영상 데이타를 3개의 메모리에 중복하여 저장하여야 한다. 본 논문에서 제안하는 필터는 입력 제어 유니트라는 간단한 FSM(Finite State Machine)을 두어 3개의 메모리에 각각 행의 수를 모듈로 3으로 구분하여 중복 없이 저장하여 사용할 수 있다. 즉 영상 데이타가 그림 7와 같이 (3k+1)행, (3k+2)행, 3k행 별로 구분되어 저장되어 있으면 각 행의 입력이 종료될 때 행전환 신호를 줌으로서 스위칭 상태(switching state)를 변화시켜줌으로써 그림 7의 우측 출력 열과 같이 영상 데이타를 필터링에 적합하도록 행 단위로 스위칭하여 출력하여 준다.

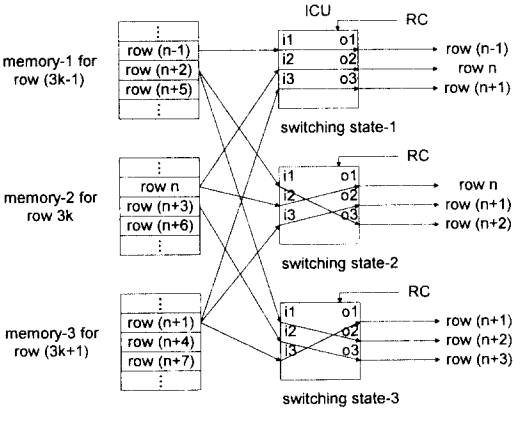

그림 7. 입력 제어 유니트의 구조
Fig. 7. The Input Control Unit.

Ⅳ. 기존의 필터와의 비교 및 성능평가

제안된 필터는 VHDL로 모델링 한후 SYNOPSYS™ CAD tool로 삼성 SOG 셀을 사용하여 논리합성과 시뮬레이션을 수행하였다. 타이밍 시뮬레이션을 통하여 최대 시스템 클럭이 125 MHz(8 nsec 주기), 데이타의 입출력 속도가 15.625 MHz(64 nsec 주기)가 됨을 검증하였다. 그림 8은 제작된 필터 칩의 사진으로 3,893개의 게이트로 구성되었으며 47개의 동작 핀과 17개의 전력 핀을 가지고 있다.

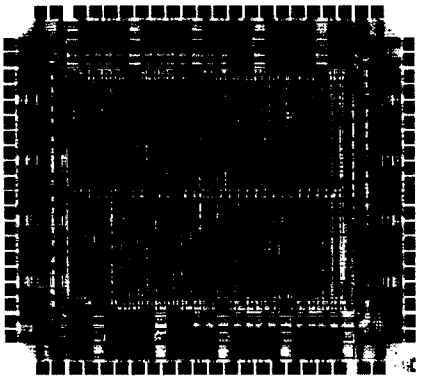

그림 8. 제작된 필터 칩의 사진
Fig. 8. The photograph of the implemented filter chip.

표 1의 곱셈기가 있는 HARRIS semicondutor사의 HSP48901 칩은 3×3 영상 필터로서[4] 본 논문에서 제안한 필터와 같은 기능을 갖는 필터이며 객관적으로 비교하기 위해 제시하였다. 크기의 단위는 객관성을 유지하기 위해 공정 과정에 따라 변화 가능성이 많은 실제 물리적인 크기를 사용하지 않고 게이트 수로 하였다. HSP48901의 규격은 HARRIS semicondutor사의 메뉴얼[4]을 참조했고 연산 블럭의 게이트 수는 공개되지 않은 관계로 메뉴얼에 도시된 구조를 바탕으로하여 4개의 트랜지스터(transistor)를 1개의 게이트로 계산하여 얻은 수치이며 곱셈기는 9개의 배열형 곱셈기(array multiplier)를 사용한 것으로 가정하였다.

표 1에서 보인바와 같이 HSP48901은 13,594개의 게이트로 구성되며 본 논문에서 제안된 필터의 게이트 수는 3,893개이므로 약 71%의 게이트 감소 효과를 보았다. 또한 본 논문에서 제안된 필터는 720×480 크기의 영상신호를 초당 30 프레임으로 실시간 처리할 수 있다. 데이타 속도 대 크기의 비율면에서 1.8배 효과적이며 실제 필터링을 실행하는 연산 블럭의 비율은 약 3.1배 우수하다.

표 1. 기존의 필터 칩과의 비교
Table 1. Comparison with other filter chips.

필터 / 규격	HSP48901	기존의 SA를 이용한 필터	제안된 필터
시스템 클럭 속도	30 MHz	175 MHz	125 MHz
데이타 처리 속도	30 Mpixel/sec	17.6 Mpixel/sec	15.6 Mpixel/sec
게이트 수	13,594 gates	약 7,500 gates	3,893 gates
처리 속도/게이트 수	2.20 Kpixel/gate	약 2.34 Kpixel/gate	4.02 Kpixel/gate

또한 기존의 SA를 이용한 필터와의 비교에서는 데이타 처리속도는 동일한 수준을 유지할 수 있으며 SA를 9개에서 1개로 트리형 가산기의 비트수를 16 비트에서 8비트로 감소시켜 역시 하드웨어 비용을 크게 줄일 수 있다.

V. 결 론

본 논문에서는 처리속도 대 크기의 비율을 향상시킬 수 있는 컨볼루션 필터 칩의 새로운 구조를 제안하고 설계하였다. 이는 곱셈기를 사용하지 않고 하나의 SA만을 사용하는 새로운 구조로 게이트 수를 기존의 곱셈기를 사용한 필터의 70% 이상 감소시키고도 720×480 크기의 영상 데이타를 초당 30 프레임으로 실시간 처리할 수 있는 필터를 구현하였으며 처리속도 대 크기면에서도 우수함을 확인하였다. 또한 기존의 SA를 사용하는 필터에 비하여 SA의 수를 9개에서 1개로 줄였으며 트리형 가산기의 비트수는 16 비트에서 8 비트로 감소시켰다. 제안된 필터는 삼성 SOG 셀 라이브러리를 사용하여 구현되었으며 3,893개의 게이트로 구성되며 최대 동작 주파수는 125 MHz이다.

참 고 문 헌

[1] Rafael C. Gonzalez and Richard E. woods, *Digital Image Processing*, Addison Wesley, June 1993, pp. 189-200.

[2] Milan Sonka, Vaclav Hlavac and Roger Boyle, *Image Processing, Analysis and Machine Vision*, Chapman and Hall, 1993, pp. 67-96.

[3] Edward R. Dougherty, *Digital Image Processing Methods*, Marcel Dekker, 1994, pp. 1-19.

[4] HARRIS semiconductor Inc., *Digital Signal Processing*, 1994.

[5] Vijai K. Madisetti, *VLSI Digital Signal Processors*, Butterworth-Heinemann, 1995, pp. 121-129.

[6] Neil H.E. Weste and Kamran Eshraghian, *Principles of CMOS VLSI Design*, Addson Wesley, 1993, pp. 317-325.

[7] R. Jain, R. T. Yang, T. Yoshino, FIR-GEN: A computer-aided design system for high performance FIR filter integrated circuit, *IEEE Trans. Signal process.*, vol. 39, pp. 1655-1668, July 1991.

[8] T. Yoshino, R. Jain, P. T. Yang, H. Davis, W. Gass, and A. H. Shah, A 100-MHz 64-tap FIR digital filter in 0.8-mm BiCMOS gate array, *IEEE J. Solid-State Circuit.*, vol. 25, pp. 1494-1501, Dec. 1990.

[9] M. Ishikawa *et al.*, Automatic layout synthesis for FIR filters using a silicon compiler, in *Proc., 1990 IEEE Int. Symp. Circuit Syst.*, May 1990, New Orleans, LA, pp. 2588-2591.

[10] Woo Jin Oh and Yong Hoon Lee, Implementation of Programmable Multiplierless FIR Filters with Power-of-two Coefficient, *IEEE Trans. Circuits Syst.*, vol. 42, pp. 553-555, August 1995.

[11] K. Khoo, A. Kwentus, and A. N. Willson, Jr., An efficient 175MHz Programmable FIR digital Filter, in *Proc.. 1993 IEEE Int. Conf. Circuits Syst.*, Chicago, IL, pp. 72-75, May 1993.

[12] ISO-IEC/JTC1/SC29/WG11, MPEG92/229 (revised), *Information on requirements for MPEG-2 Video*, Jul. 1992.

저 자 소 개

殷 世 英(正會員)
1996년 2월 아주대학교 전자공학 학사. 1996년 2월 ~ 현재 아주대학교 전자공학 석사과정. 관심분야는 통신 및 신호처리용 ASIC 설계

鮮于 明勳(正會員) 第 34卷 C編 第 8號 參照