

논문 2008-45SD-11-12

H.264 복호기를 위한 스케일러블 인트라 예측기 구조 설계

(Design of Scalable Intra-prediction Architecture for H.264 Decoders)

이 찬 호*

(Chanho Lee)

요 약

H.264는 ITU-T와 ISO/IEC의 최신 동영상 압축 코덱 규격으로 MPEG-2보다 2배 이상의 압축률과 고화질로 최근 그 적용 영역을 급격히 넓혀 가고 있다. H.264는 QVGA급의 작은 영상부터 HD 크기의 넓은 영상까지 다양하게 적용되므로 응용 분야에 따라 그 구조가 달라진다. 본 논문에서는 H.264 복호기의 인트라 예측기에 대해 응용 분야에 따라 구조를 쉽게 확장할 수 있는 스케일러블(scalable) 구조를 제안하고 이 구조에 따라 인트라 예측기를 설계하여 동작과 성능을 검증하였다. 제안된 구조는 인트라 예측기 내부의 연산기 수를 1~4개까지 변화시키면서 성능을 4배까지 향상시킬 수 있다. 또한 효율적인 버퍼 관리를 통해 메모리 접근을 최소화 하여 전력 소모를 줄였다. 제안된 인트라 예측기는 Verilog-HDL을 이용하여 설계하고 FPGA 상에서 동작을 검증하였다. 이를 바탕으로 연산기 수에 따른 인트라 예측기의 성능을 분석하였다.

Abstract

H.264 is a video coding standard of ITU-T and ISO/IEC, and widely spreads its application due to its high compression ratio more than twice that of MPEG-2 and high image quality. It has different architecture depending on demands since it is applied from small image of QVGA to large size of HD. In this paper, We propose a scalable architecture for intra-prediction of H.264 decoders. The proposed scheme has a scalable architecture that can accommodate up to 4 processing elements depending on performance demands and can reduce the number of access to memory using efficient memory management so as to be energy-efficient. We design the intra-prediction unit using Verilog-HDL and verify it by prototyping using an FPGA. The performance is analyzed using the results of design.

Keywords : H.264, MPEG-4/AVC, intra-prediction, scalable architecture, 메모리 관리

I. 서 론

디지털 신호처리, 저장매체 및 전송방식의 발전은 음성 정보에 국한된 서비스를 멀티미디어 전송 서비스로 진화시켰으며 최대한 많은 데이터를 저장하고 전송하기 위해서 효율적이고, 유연성이 있으며, bit와 packet 에러에 강한 여러 가지 압축 기술이 연구되었다^[1]. H.264는 ITU-T와 ISO/IEC가 공동으로 개발한 고성능 동영상

압축 표준으로 MPEG-4/AVC part 10으로도 불린다. H.264는 4x4의 작은 블록 크기, 1/4-pel 화소 보간, 다중 참조 프레임, 내부 블록제거 필터 등의 기술을 이용하여 MPEG-4보다 50% 이상, MPEG-2보다 2배 정도의 높은 압축률과 우수한 화질을 제공하여 그 응용 영역을 급속히 넓혀가면서 기존의 MPEG-2와 MPEG-4를 대체해 가고 있다. 높은 압축률과 우수한 화질을 제공하기 위한 새로운 알고리즘의 도입으로 H.264 압축 규격은 복잡한 연산처리를 요구하고 이에 따라 연산기 구조도 매우 복잡하다^[1]. 이러한 이유로 H.264 인코더와 디코더는 대체로 하드웨어로 설계되고 있다.

H.264 압축 규격의 화면 내 예측(intra-prediction)은 프레임 내에서 공간 중복성을 줄이는 과정이며, 변환(transform)과 양자화(quantization)이전 단계에서 화면

* 정희원, 숭실대학교 정보통신전자공학부
(School of Electronic Engr., Soongsil University)

※ 본 논문은 지식경제부가 지원하는 국가 반도체 연구개발사업인 “시스템집적반도체기반기술개발사업(시스템 IC 2010)”과 숭실대학교 교내연구비 지원을 통해 개발된 결과임을 밝힙니다. 또한 연구에 이용된 EDA 툴은 IDEC으로부터 지원받았습니다.

접수일자: 2008년7월29일, 수정완료일: 2008년10월28일

내 예측 과정을 거치게 하여 I-Slice의 압축효율을 높일 수 있도록 되어있다. 이는 인접한 매크로 블록들이 유사한 특성을 가지고 있는 것에 기초를 두고 있다^[2].

다른 연산기들과 마찬가지로 인트라 예측기도 영상의 크기가 클수록 연산량이 많아진다. 부드러운 영상의 재생을 위해서는 초당 프레임 수를 일정하게 유지해야 하므로 연산량이 늘어나도 이를 처리해야 하는 시간은 일정하다. 따라서 영상의 크기가 증가하면 연산기의 수가 증가하거나 동작 주파수가 증가해야 한다. H.264 영상 압축 규격은 휴대용 기기의 QVGA(320 x 240) 크기의 작은 영상에서 광미디어의 HD(1920 x 1080) 크기까지 다양한 영상 크기를 소화해야 한다. 동작 주파수를 증가시키는 방식은 저전력 설계를 어렵게 만들고 주어진 시스템에서 필요한 주파수를 공급받지 못할 수도 있다. 반면에 연산기수를 늘리는 것은 하드웨어 설계를 다시 해야 하는 문제가 있다. 따라서 용도별로 새로운 구조를 설계해야 하므로 효율적이지 못하다.

본 논문에서는 이러한 문제를 해결하기 위해 스케일러블(scalable) 구조의 인트라 예측기 구조를 제안한다. 제안한 인트라 예측기는 하나의 예측된 화소값을 하나의 연산기(PE: Processing Element)에서 처리하고 이 연산기 수를 늘려 연산 처리 속도를 증가시킬 수 있는 스케일러블 구조를 가지고 있다. 인트라 예측기는 처음부터 연산기를 추가할 수 있도록 설계하여 연산기를 추가할 필요가 있을 때 연산기 수만을 늘려 성능을 증가시킬 수 있다. 또한, 효율적인 메모리 관리를 통해 메모리 접근을 최소화하였다.

II. 스케일러블 인트라 예측기

1. H.264 표준의 인트라 예측 알고리즘

H.264 영상 압축 표준의 인트라 예측 연산은 그림 1에 나타난 바와 같이 원본 이미지에서 인트라 예측 이미지를 뺀 나머지(residual)와 인트라 예측 이미지에 대한 파라미터를 전송하고 이로부터 이미지를 복구하는 작업이다. 인트라 예측 이미지는 4 x 4 서브 블록 휘도 성분에 대해 9 가지의 예측 방식, 16 x 16 매크로 블록 휘도 성분에 대해 4 가지의 예측 방식, 그리고 8 x 8 매크로 블록 색차 성분에 대해 4 가지의 예측 방식으로 만들어진다^[2]. 이를 복구하기 위해서는 이미 복구된 영상의 데이터가 필요한데 4x4 휘도 블록인 경우 그림 2(a)의 대문자로 표시된 위치의 참조 픽셀 값이, 16x16 휘도 블록과 8x8 색차 블록은 그림 2(b)의 H와 V에 해

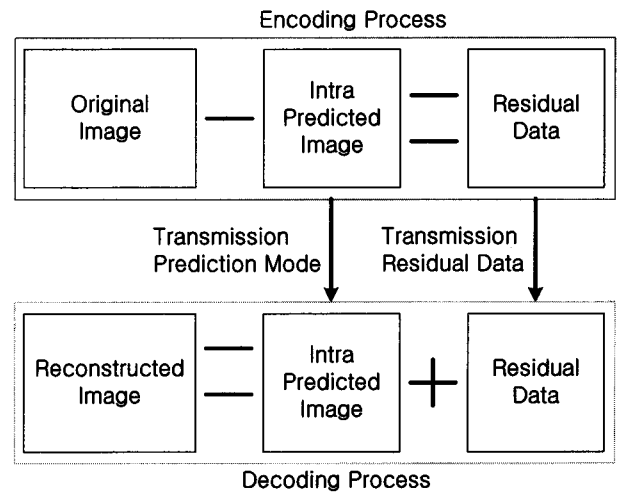


그림 1. 인트라 예측 연산 과정
Fig. 1. Encoding and decoding processes of intra prediction.

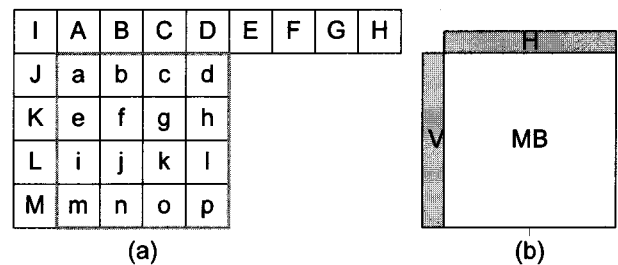


그림 2. 인트라 예측을 위한 참조 픽셀값의 위치 (a) 4x4 휘도 블록 (b) 16x16 휘도 블록과 색차 블록
Fig. 2. Positions of reference pixels for intra-prediction. (a) 4x4 luminance block (b) 16x16 luminance block and chroma block.

당하는 참조 픽셀 값이 필요하다. 따라서 복구된 영상의 픽셀 값의 일부를 다시 이용해야 한다.

또한, 디코더의 인트라 예측기는 17 가지의 예측 방식을 모두 처리할 수 있는 연산기를 포함하여야 한다. 16 가지의 연산은 참조 픽셀 값을 복사하거나 가중 평균 값을 구하는 비교적 간단한 연산이지만, 16x16 휘도 블록과 8x8 색차 블록에 대한 플레인(plane) 모드는 아래의 식과 같은 매우 복잡한 연산을 요구한다^[2].

내부 메모리에 저장되는 참조픽셀은 그림 3에서 보는 바와 같이 이미 예측 연산이 끝난 위와 왼쪽 매크로 블록의 아랫줄 픽셀과 오른쪽 줄 픽셀들이다. 해당 매크로블록은 인트라 예측이 끝나면 디블록킹 필터로 보내져 필터링이 진행되므로 필요한 참조 픽셀을 별도로 저장하여야 한다. 참조 픽셀 저장에 필요한 메모리 크기는 전체 이미지 크기가 아닌 가로로 배치된 매크로블록의 수에 의해 결정된다. 즉, 그림3의 가로 진하게 칠

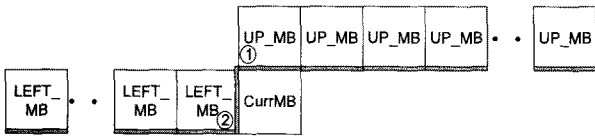


그림 3. 인트라 예측 연산을 위한 참조 픽셀
Fig. 3. Reference pixels for intra-prediction.

해진 부분이 인트라 예측기가 필요한 영상 이미지의 하나의 픽셀 라인이다.

2. 스케일러블 인트라 예측기 구조

복호기의 인트라 예측기는 그림 1의 복호과정에서 인트라 예측 이미지를 생성시키는 역할을 한다. 전송된 예측 이미지 모드 파라미터와 참조 픽셀 값을 이용하여 표준에서 정해진 방식에 따라 예측 픽셀 값을 계산하는데 플레인 모드를 제외하면 모두 값의 복사나 가중 평균값 계산이므로 하나의 연산기로 대부분의 모드에 대해 예측 픽셀 값을 얻을 수 있다. 인트라 예측기는 플레인 모드를 제외하면 4x4 서브블록 예측이 16x16 블록 예측과 8x8 블록 예측보다 연산량이 많다^[3]. 따라서 4x4 블록에 대한 연산기를 이용하여 16x16 블록이나 색차 블록에 대한 연산이 가능하다. 4x4 블록 예측에 필요한 연산은 덧셈기와 쉬프트 연산으로 구현이 가능하다. 그림 4는 하나의 예측픽셀을 연산하기 위한 연산기(PE: Processing Element) 구조이다. 총 4개의 덧셈기와 한 개의 쉬프트로 구성되며, 평균값 예측 모드에

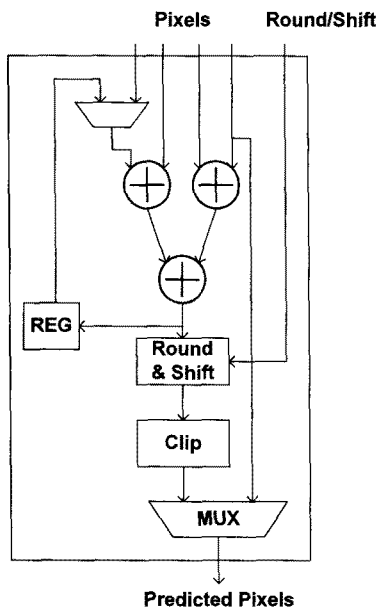


그림 4. 픽셀 연산을 위한 연산기(PE) 구조
Fig. 4. Schematic of a processing element(PE) for calculating predicted pixels.

필요한 레지스터와 수직과 수평 예측을 위한 바이패스 라인을 가지고 있다. 덧셈기중 하나는 clip 함수 유닛에 포함되어 있다. 이 연산기는 16x16 휘도 블록과 8x8 색차 블록의 플레인 모드를 제외한 모든 예측 연산에서 사용 가능한 구조로 설계하였다.

플레인 모드의 a, b, c, H, V는 인트라 예측기에서 가장 복잡한 연산으로 이를 효과적으로 계산하기 위해 Y. W. Huang 등이 제안한 알고리즘을 이용하였다^[4]. 이 알고리즘을 적용하면 곱셈기를 사용하지 않고 덧셈기와 쉬프트 연산만으로 동작할 수 있는 장점이 있다. 그림 5에 플레인 모드 연산을 위한 로직의 블록 다이어그램이 나타나 있다. A0, A1, A2, A3는 매크로블록에서 위의 4개의 4x4 서브 블록에서 가장 왼쪽 위에 위치하는 예측 픽셀 값이다. 이 값들이 구해지면 이를 seed로 해서 4x4 블록의 윗줄의 나머지 세 개의 예측 값은 PE를 이용하여 계산이 가능하다.

하나의 연산기는 한번에 하나의 예측 픽셀을 연산 할 수 있는데 출력값이 다음 픽셀 연산을 위한 입력값과 연관성이 없으므로 연산기의 수를 증가시키면 인트라 예측 연산 속도를 증가시킬 수 있다. 따라서 연산기를 하나 추가 할 때 마다 연산 속도는 대체로 그에 비례하

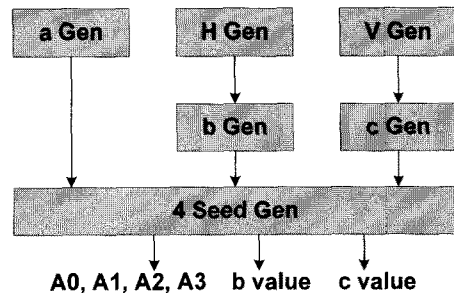


그림 5. 플레인 모드 연산을 위한 연산기 구조
Fig. 5. Block diagram of calculation unit for plane modes of 16x16 luminance and chroma blocks.

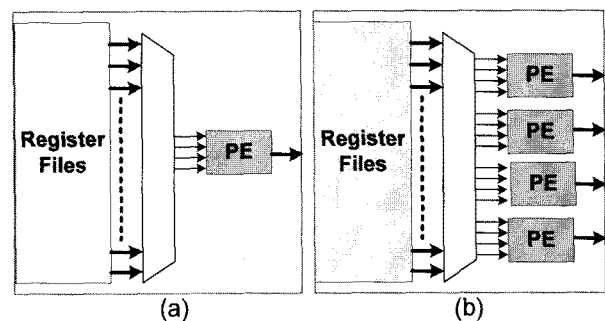


그림 6. 연산기수에 따른 예측 픽셀 연산 방법 (a) PE 1개 (b) PE 4개
Fig. 6. Calculation of predicted pixels using (a) 1 PE and (b) 4 PEs.

표 1. 연산기 1개와 4개일 때의 4x4 휘도 블록에 대한 연산 사이클 비교

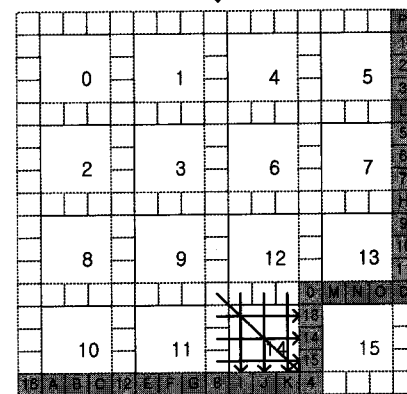
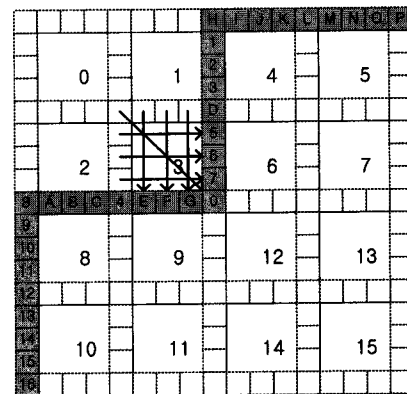
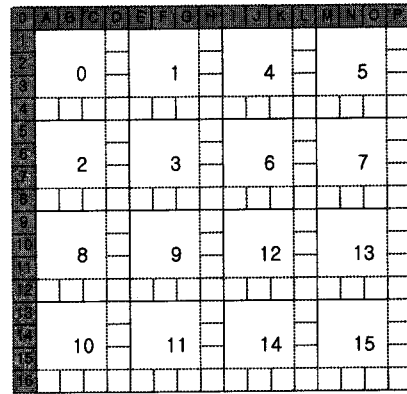
Table 1. Comparison of operation cycles using 1 PE and 4 PEs for 4x4 luminance blocks.

모드	0	1	2	3	4	5	6	7	8
PE1 (cycle)	16	16	19	16	16	16	16	16	16
PE4 (cycle)	4	4	7	4	4	4	4	4	4

여 증가한다. 예를 들어 1개의 연산기가 매크로블록 당 400사이클에 동작을 완료할 경우, 연산기를 4개로 증가시키면 약 100사이클로 연산 속도가 증가된다. 그림 6과 같이 PE에 입력값을 공급하는 로직만 변경시키면 PE의 수를 쉽게 증가시켜 인트라 예측기의 성능을 향상시킬 수 있다. 즉, 연산기의 추가는 동작 주파수의 증가 없이 연산 속도를 증가시킬 수 있다. 전체 연산기의 수는 1, 2, 4개일 때 가장 큰 효과를 볼 수 있고 3개 또는 5개 이상에서는 연산기 수만큼의 성능 향상을 기대할 수 없다.

표 1에 4x4 블록에 대한 9가지 연산 모드에서 연산기가 1개인 경우와 4개인 경우에 대해 동작 사이클 수를 비교한 결과가 나타나 있다. 모드 2를 제외하고는 연산기가 4배가 되면 동작 사이클이 1/4로 감소하는 것을 볼 수 있다. 모드 2의 경우에는 연산기가 동작하기 전에 준비 사이클이 필요한데 이는 연산기의 수와 관계없이 필요하므로 엄밀하게는 4배의 성능 향상을 보이지는 않는다. 그러나 모드 2가 연속해서 나오는 경우는 파이프라인 방식으로 동작하므로 초기 레이턴시에 의한 영향은 상대적으로 감소하고 모드 2가 차지하는 비율이 크지 않으므로 전체적으로는 4배 정도의 성능 향상을 기대할 수 있다.

인트라 예측을 위해서는 참조픽셀이 필요하다. 휘도 4x4 서브 블록의 경우에는 그림 2(a)에 나타난 것처럼 13 바이트(A, B, C, ..., K, L, M), 휘도 16x16 블록은 그림 2(b)에서와 같이 32 바이트(좌측: 16바이트, 상측: 16 바이트), 색차 8x8 블록은 16 바이트(좌측: 8 바이트, 상측: 8 바이트)가 각각 필요하다. 주변 참조픽셀 값들은 이전에 복호화 되어 복원되고 디블록킹 필터(deblocking filter)를 적용하기 전의 픽셀 값을 의미한다. 즉, 인트라 예측기가 사용하는 참조픽셀은 필터링되기 전 픽셀 값이기 때문에 움직임 보상이 요구하는 참조 프레임 데이터와는 따로 저장이 필요하고, 연산의 최소기준은 4x4 서브블록 단위이다. 따라서 인트



Internal Memory

그림 7. 인트라 예측기의 참조 픽셀 저장을 위한 제안된 메모리 관리 기법

Fig. 7. Proposed memory management for reference pixels in intra-prediction unit.

라 예측을 하기위해 참조 영상 값을 저장할 메모리와 효율적인 메모리가 필요하고 이를 위한 관리 방법이 필요하다.

그림 7은 제안한 인트라 예측기의 메모리 관리 방법을 나타낸다. 그림에서 진하게 칠해진 부분이 참조 픽셀이 저장된 레지스터 파일을 나타낸다. 4x4 블록 연산의 경우 블록에 따라 참조 픽셀이 달라지므로 레지스터

파일의 참조 픽셀 값은 계속 바뀌게 된다. 이때 메모리 관리가 효과적으로 이루어져야 연산의 흐름을 방해하지 않고 파이프라인이 동작하여 연산 사이클에서 손실이 발생하지 않는다. 그림 7에서 서브 블록 0을 연산할 때 필요한 참조픽셀은 레지스터 파일의 0~4와 A~H까지이다. 서브블록 0에 대한 연산이 끝나면 해당 참조픽셀은 더 이상 사용되지 않기 때문에 레지스터 파일의 해당 위치에 1번과 2번 블록 연산에 필요한 참조픽셀을 저장할 수 있다. 그림 7의 두 번째 그림에서 3번 블록에 대한 연산이 끝나면 3번 블록의 오른쪽 네 개의 픽셀은 5, 6, 7, 0번 주소에, 아래의 네 개의 픽셀은 E, F, G에 저장된다. 네 번째 값은 오른쪽 줄의 마지막 픽셀과 동일하므로 공동으로 사용한다. 따라서 그림 7의 철해진 부분에 해당하는 크기의 메모리만 있으면 하나의 매크로블록에 대한 연산을 진행할 수 있다. 10, 11, 14, 15 서브 블록의 경우에 아래 네 픽셀은 나중에 다시 필요하기 때문에 내부 메모리에 저장한다. 그림 7의 세 번째 그림에서 14번 서브 블록의 경우 I, J, K, 4가 이에 해당한다. 사용된 내부 메모리는 32비트 데이터 폭을 가지며 메모리 접근(읽기/쓰기)은 매크로블록 당 8 사이클이 사용된다. 매크로블록의 세로줄 픽셀인 P, 1, 2, 3, L, 5, 6, 7, H, 9, 19, 11, D, 13, 14, 15, 0은 다음 매크로블록 연산에 다시 이용되므로 내부 메모리에 저장하지 않고 가로줄의 픽셀만 저장하고 다음 매크로블록에서는 다시 가로줄에 해당하는 참조 픽셀을 가져와서 채운다. 따라서 디블록킹 필터로 보내는 데이터를 다시 받아 레지스터 파일에 저장할 필요가 없다.

제안한 연산 구조와 메모리 관리법을 적용한 스케일러블 인트라 예측기의 구조가 그림 8에 나타나 있다. 제안된 인트라 예측기는 PE를 쉽게 추가 할 수 있는 스케일러블 구조를 가지고 있다. 인트라 예측기는 일반

적으로 연산량이 상대적으로 적으므로 동작 주파수를 올리면 하나의 PE로 연산이 가능한 경우도 있으나 저 전력 설계의 경우 또는 동작 주파수를 올릴 수 없는 경우에는 PE를 추가함으로써 원하는 성능을 얻을 수 있다. 제안한 메모리 관리 기법을 이용하여 레지스터 파일을 관리하면 그림 7의 ①과 같이 디블록킹 필터로 보내는 값을 다시 읽어올 필요가 없어 연산 사이클과 불필요한 동작을 줄일 수 있다.

III. 설계 및 검증

제안한 구조에 따라 설계된 하나의 연산기를 가지고 있는 인트라 예측기를 Verilog-HDL을 이용하여 RTL 코드로 설계하였다. 설계한 인트라 예측기는 시뮬레이션과 FPGA 구현을 통해 동작을 검증하였다. 검증 결과 매크로블록 당 약 450사이클이 소모되어 SD(720x480)와 HD(1920x1080) 영상을 각각 20MHz와 104MHz의 주파수에서 동작이 가능하다. 제안된 인트라 예측기는 스케일러블한 구조를 가지고 있기 때문에 연산기를 추가하여 4개로 구성했을 때 동작 주파수는 5MHz, 26MHz로 줄어든다. 그림 9는 FPGA를 이용하여 검증 플랫폼에서 동작시켜 얻은 영상 이미지이다. QCIF 크기의 "foreman" 이미지가 정상적으로 복호되고 있음을 보여주고 있다.

Synopsys Design Compiler와 0.25um CMOS 셀 라이브러리를 이용한 인트라 예측기의 합성 결과 및 기존 결과와의 비교를 표 2에 정리 하였다. 설계된 인트라 예측기는 약 20,400 게이트 크기를 갖고, 최대 200MHz에서 동작 가능하다. 그러나 제안된 구조에서는 104MHz에서 HD 영상 크기를 예측할 수 있다. 구현된

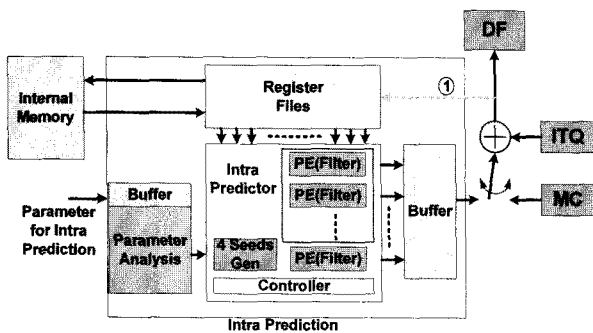


그림 8. 제안하는 스케일러블 인트라 예측기의 구조
Fig. 8. Proposed architecture of scalable intra-prediction unit.

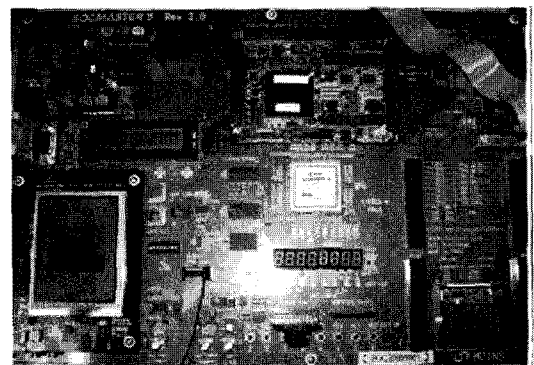


그림 9. FPGA 검증 시스템을 이용한 검증 결과
Fig. 9. Verification result using an FPGA prototyping board.

표 2. 기존의 구조와 제안한 구조의 구현 결과 비교
Table 2. Comparison of implementation results of the conventional and the proposed architecture.

	[5]	[6]	[7]	Proposed
Technology	0.18um	0.25um	0.18um	0.25um
Logic Gates	28,707	32,190	11,900	20,400
Operating Frequency (image size)	120MHz (HD)	54MHz (CIF)	107MHz (CIF)	104MHz (HD)
# of cycles/MB	490	4,545	8,885	450
Profile	Baseline	Baseline	Baseline	Baseline

인트라 예측기를 기존의 결과와 비교해 보면 동작 주파수는 104MHz에서 HD 영상 디코딩이 가능하여 매크로블록당 가장 적은 수의 사이클로 동작이 가능함을 알 수 있다. 합성 면적의 경우 참고 문헌 [5]와 [6]보다 작고 [7]보다는 큰 것을 알 수 있다. 그러나 참고문헌 [7]의 경우 매크로블록당 8,885 사이클이 필요하여 제안한 구조의 450 사이클보다 20배 가까이 크다. 이는 연산기를 줄여 면적을 반으로 줄이면서 성능을 1/20로 줄인 것과 같아 비교 대상이 될 수 없다. 참고 문헌 [6]의 매크로블록당 연산 사이클 수는 동작 주파수로부터 계산한 것으로 표 3의 결과보다는 더 작을 것으로 예상되나 문헌상에 인트라 예측기만의 성능 지표가 나타나 있지 않다. 그러나 동작 주파수와 면적을 고려할 때 제안한 구조보다는 훨씬 클 것으로 예상된다.

IV. 결 론

본 논문에서는 H.264/AVC 복호기를 위한 고성능 인트라 예측기 구조를 제안하였다. 제안된 인트라 예측기의 기본 연산기는 덧셈기와 쉬프트만으로 구성되어 있고 한번에 하나의 예측 픽셀을 연산 할 수 있다. 이 연산기를 필요에 따라 증가시키면 동시에 연산할 수 있는 픽셀 수가 증가하여 동작 성능을 향상시킬 수 있다. 제안한 인트라 예측기는 연산기를 1, 2, 4개로 구성할 수 있는 스케일러블 구조를 가지고 있다. 또한, 효율적인 메모리 관리를 통해 메모리 크기를 줄이면서 메모리 접근을 최소화하여 전력 소모를 줄이고 대기 시간을 줄여 연산 사이클을 단축하였다. 제안한 구조를 기반으로 인트라 예측기를 Verilog-HDL을 이용하여 설계하고 FPGA를 이용하여 검증하였다. 설계 결과물을 0.25um CMOS 공정과 Synopsys Design Compiler를 이용하여 합성한 결과 20,400의 게이트 수와 200MHz의 최대 동

작 주파수를 얻었고 100MHz에서 HD 영상 복호가 가능하여 기존 결과에 비해 우수함을 보였다.

참 고 문 헌

- [1] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, pp. 688 - 703, July 2003.
- [2] I. E. G. Richardson, "H.264 and MPEG-4 VIDEO COMPRESSION," John Wiley & Sons, pp.159-224, 2003.
- [3] W. Zouch, A. Samet, M. A. Benayed, F. Kossentini and N. Masmoudi, "Complexity analysis of Intra prediction in H.264/AVC," *IEEE International Conference on Microelectronics*, vol.1, pp.713-717, 2004.
- [4] Y. W. Huang, B. Y. Hsieh, T. C. Chen and L. G. Chen, "Analysis, fast algorithm, and VLSI architecture design for H.264/AVC intra frame coder," *IEEE Trans. on Circuits and Systems for Video Technology*, vol.15, pp.378-401, 2005.
- [5] T.-C. Chen, C. J. Lian, and L.-G. Chen, "Hardware architecture design of an H.264/AVC video codec," *Asia and South Pacific Design Automation Conference*, pp. 750-757, Jan. 2006.
- [6] S. M. Park and et.al, "VLSI Implementation of H.264 Video Decoder for Mobile Multimedia Application," *ETRI Journal*, vol.28, pp.525-528, 2006.
- [7] 윤철환, 정용진, "모바일 환경에서의 H.264/AVC를 위한 인트라 예측기의 구현 및 검증," *전자공학회 논문지-SD*, 44권 12호, pp. 93-101, 2007.12

저 자 소 개

이 찬 호(정회원)

대한전자공학회 논문지

제43권 SD편 제9호 참조

현재 숭실대학교 정보통신전자공학부 교수