

논문 2010-47SD-5-4

H.264/AVC 복호기를 위한 효율적인 인트라 예측기 하드웨어 구조

(The Hardware Architecture of Efficient Intra Predictor for H.264/AVC Decoder)

김 옥*, 류 광 기**

(Ok Kim and Kwangki Ryo)

요 약

본 논문에서는 H.264/AVC에서 압축 성능을 높이기 위해 사용된 기법중 하나인 인트라 예측에 대해 기술하고 인트라 예측 모드 연산을 효율적으로 수행하기 위한 인트라 예측기의 구조를 제안한다. 제안하는 인트라 예측기는 공통 연산기, 전처리 연산기, 인트라 예측 컨트롤러, 내부 메모리, 레지스터 컨트롤러로 구성된다. 공통 연산기와 전처리 연산기를 사용하여 연산량을 줄이고, 내부 메모리와 레지스터를 사용하여 외부 메모리와의 접근을 최소화 하였다. 제안한 인트라 예측기는 Verilog-HDL을 이용하여 설계하였으며, YUV 파일을 부호화 한 테스트 벡터를 이용하여 검증하였다. 설계된 인트라 예측기는 주로 핸드폰과 같은 휴대용 단말기에 사용하는 베이스라인 프로파일의 속하며 영상크기는 176x144 픽셀이다. 제안한 인트라 예측기의 예측 수행 사이클의 비교 분석 결과 기존에 비해 평균 약 60%의 향상된 결과를 얻었다.

Abstract

In this paper, we described intra prediction which is the one of techniques to be used for higher compression performance in H.264/AVC and proposed the design of intra predictor for efficient intra prediction mode processing. The proposed system is consist of processing elements, precomputation processing elements, an intra prediction controller, an internal memory and a register controller. The proposed system needs the reduced the computation cycles by using processing elements and precomputation processing element and also needs the reduced the number of access time to external memory by using internal memory and registers architecture. We designed the proposed system with Verilog-HDL and verified with suitable test vectors which are encoded YUV files. The proposed architecture belongs to the baseline profile of H.264/AVC decoder and is suitable for portable devices such as cellular phone with the size of 176x144. As a result of experiment, the performance of the proposed intra predictor is about 60% higher than that of the previous one.

Keywords : H.264/AVC, H.264, 화면 내 예측, Intra Prediction, Intra Predictor

I. 서 론

H.264/AVC 기술은 ISO/IEC와 ITU-T가 공동으로 설립한 JVT(Joint Video Team)에 의해 공동 개발 및

발표한 표준으로 지금까지의 비디오 압축 기술 중 압축 성능이 가장 우수한 비디오 부호화 표준이다. H.264/AVC 기술은 MPEG-1, MPEG-2, MPEG-4 비디오로 이어지는 기본 압축 흐름은 유사하게 사용하였지만 세부 알고리즘은 크게 변경 되었다^[1]. 새롭게 채택된 기술 덕분에 다른 부호화 표준보다 동일 화질을 기준으로 높은 압축률을 이루었으나 다른 비디오 부호화 표준과 비교하여 부호화의 복잡도가 증가하는 단점이 있다. 따라서 H.264/AVC의 세부 알고리즘을 효율적으로 구

* 학생회원, **정회원, 한밭대학교 정보통신공학과
(Department of Information and Communication Engineering, Hanbat National University)

※ 본 연구는 IDEC의 CAD Tool 지원사업 및 중소기업청의 산학협력실 지원사업의 연구결과임.

접수일자: 2009년11월20일, 수정완료일: 2010년3월1일

현하는 기술이 매우 중요하게 되었다. 본 논문에서는 H.264/AVC 복호기의 세부 알고리즘 중 하나인 인트라 예측에 대해서 기술하고, 인트라 예측연산을 효율적으로 수행하기 위한 인트라 예측기의 하드웨어 구조에 대해 제안한다. 제안한 인트라 예측기는 공통 연산기와 전처리 연산기를 구현하여 예측 연산 수행 사이클 수를 감소 시켰으며, 레지스터 컨트롤러를 구현하여 외부 메모리와의 접근을 최소화 하였다.

II. 인트라 예측 알고리즘

인트라 예측은 슬라이스의 유사성을 제거함으로써 프레임 내의 공간 중복성을 줄이는 과정으로 변환과 양자화 이전 단계에서 인트라 예측을 거치게 하여 I 슬라이스의 압축효율을 높이는 방법이다. 이는 인접한 매크로 블록들이 유사한 특성을 가지고 있는 것에 기초를 두고 있으며 현재 슬라이스의 공간적 중복성을 제거하기 위하여 부호화된 이전 매크로 블록의 픽셀 값을 참조하여 가장 유사한 매크로 블록을 생성한다.

H.264/AVC의 인트라 예측은 MPEG-4의 인트라 예측과 많은 차이점이 있다. MPEG-4에서는 왼쪽 블록, 위쪽 블록, 왼쪽 상단 블록의 세 가지 블록을 참조 하지만 H.264/AVC에서는 왼쪽 블록, 위쪽 블록, 오른쪽 상단 블록, 왼쪽 상단 블록 네 가지를 참조한다. 또한 MPEG-4 예측의 경우 휘도 예측 방향이 결정되면 색차 예측은 휘도 예측 방향을 따라가게 되어 있으나, H.264/AVC의 경우 색차 정보와 휘도 정보는 각각 독립적으로 예측이 수행된다. H.264/AVC의 휘도 예측 모드는 경우에 따라 16x16 블록에 대한 예측 방법과 4x4 블록에 대한 예측 방법으로 나뉘어 적용되며, 색차 예측 모드는 8x8 블록 단위로 예측된다. 복잡한 영상은 4x4 블록과 같은 작은 블록 단위로 예측하고 평탄한 영상은 16x16 블록과 같은 큰 매크로 블록 단위로 예측함으로써 8x8 블록 단위로만 예측하는 MPEG-4와 비교할 때 복잡한 영상과 평탄한 영상 모두 높은 효율로 예측이 가능하다.

1. 4x4 휘도 예측

4x4 블록의 휘도 예측은 그림 1과 같이 매크로 블록에 대해 4x4 블록 단위로 주어진 순서에 따라 예측 연산이 수행되며, 예측하고자 하는 블록의 왼쪽 블록, 왼쪽 상단 블록, 위쪽 블록, 오른쪽 상단 블록을 참조하여

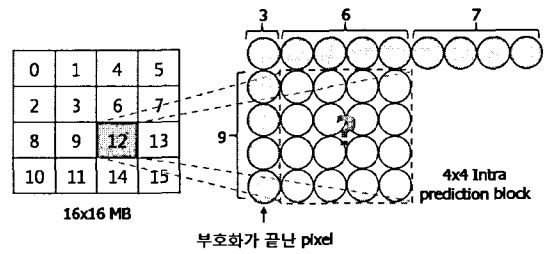


그림 1. 4x4 휘도 예측 순서와 참조 픽셀
Fig. 1. 4x4 luma prediction sequence and reference pixel.

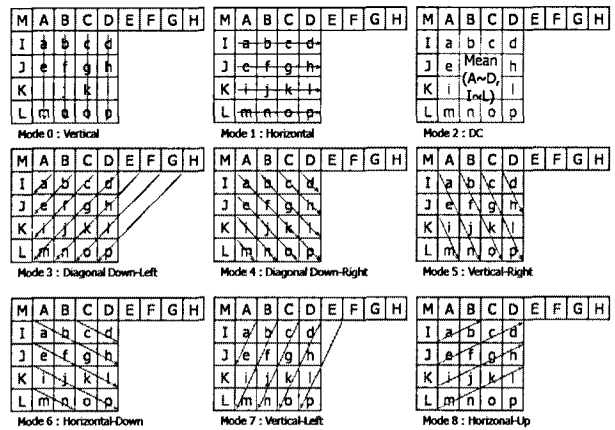


그림 2. 4x4 휘도 예측 모드
Fig. 2. 4x4 luma prediction mode.

예측을 수행한다. 참조되는 블록은 현재 블록 이전에 부호화되고 복원된 값이다.

4x4 블록 휘도 예측에는 9가지 예측 모드가 사용된다. 주변 블록 참조 픽셀의 평균값으로 대상 블록의 픽셀 값을 예측하는 DC 모드를 제외한 나머지 8가지 모드는 각각에 해당하는 예측 방향성을 가지고 있다. 4x4 블록 휘도 예측의 9가지 모드는 그림 2와 같다. DC 모드는 이웃 블록 픽셀의 참조 가능 여부에 따라 예측 연산이 달라지며, 다른 8가지 모드들은 각각 필요로 하는 모든 이웃 블록의 픽셀이 참조 가능한 경우에만 사용될 수 있다.

2. 16x16 휘도 예측

16x16 휘도 예측은 매크로 블록 전체에 대해 하나의 예측 모드를 사용하는 것으로 이미지의 단조로운 영상 부분에서 가장 좋은 결과를 나타낸다. 16x16 휘도 예측에는 4가지 예측 모드가 사용된다. 16x16 휘도 예측 모드가 그림 3에 나타나 있다. 16x16 휘도 예측은 현재 블록 이전에 부호화 되고 복원된 위쪽 블록과 왼쪽 블록의 픽셀 값을 참조하여 예측을 수행한다. 16x16 휘도

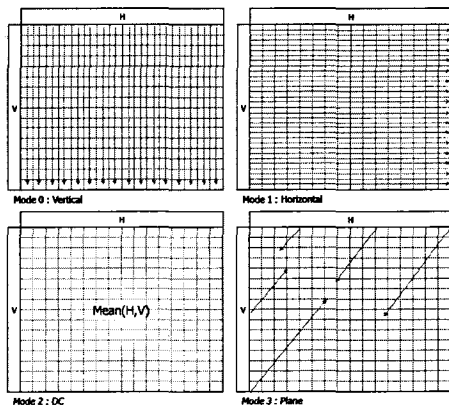


그림 3. 16x16 휘도 예측 모드
Fig. 3. 16x16 luma prediction mode.

예측은 4x4 휘도 예측 모드와 유사한 계산 방법을 사용하며, 일반적으로 4x4 휘도 예측에 비해 예측 효율이 떨어지지만 연산 데이터가 적으므로 단순한 블록의 경우 압축효율 면에서 4x4 휘도 예측보다 우수하다.

3. 8x8 색차 예측

색차 블록은 8x8 단위로 예측 된다. 색차 예측에는 4가지 모드가 지원된다. 블록 크기, 모드 순서의 차이만 있을 뿐 16x16 휘도 예측 모드와 동일하다. 색차 블록의 예측 모드가 그림 4에 나타나 있다.

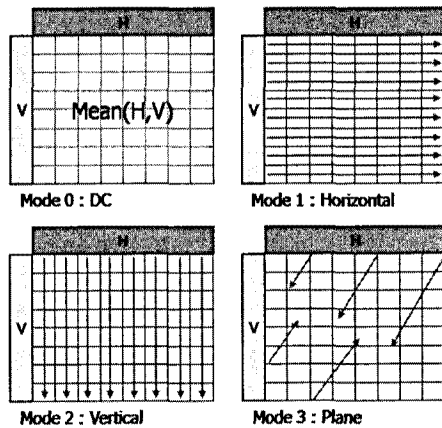


그림 4. 8x8 색차 예측 모드
Fig. 4. 8x8 chroma prediction mode.

III. 제안하는 인트라 예측기

1. 하드웨어 구조

본 논문에서 제안하는 인트라 예측기의 하드웨어 구조는 그림 5와 같다. 인트라 예측기는 전체 인트라 예

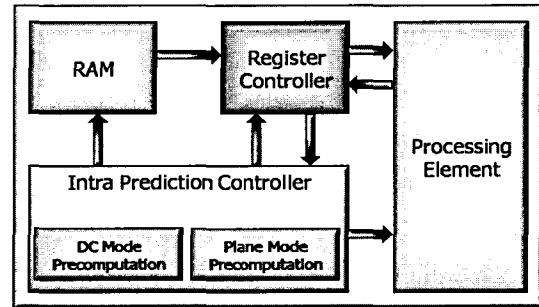


그림 5. 인트라 예측기의 하드웨어 구조
Fig. 5. Hardware architecture of intra predictor.

측기의 동작을 제어하고, 복잡한 예측 연산을 위한 전처리 연산기를 포함하는 인트라 예측 컨트롤러 부분과 실제 예측연산을 수행하는 공통 연산기 부분, 인트라 예측만을 위한 내부 메모리, 내부 메모리의 값을 예측 연산에 맞게 컨트롤 해주는 레지스터 컨트롤러 부분으로 구성된다.

가. 공통 연산기

인트라 예측에는 서로 상이한 17가지 모드가 존재하며 동일한 예측 모드일 경우에도 픽셀마다 서로 다른 연산을 통해 예측되기 때문에 연산기의 구조가 매우 복잡해진다. 다양한 예측 연산을 효율적으로 수행하기 위하여 모든 예측 연산을 수행 할 수 있는 공통 연산기(Processing Element)를 그림 6과 같이 설계 하였다.

공통 연산기는 3개의 픽셀 입력과 라운드/시프트 연산을 위한 입력 신호, 플레인 모드를 위한 컨트롤 신호, 바이패스 경로를 위한 컨트롤 신호를 가진다. 공통 연산기의 연산은 4x4 블록 단위로 이루어지며, 한 번에 하나의 픽셀 연산이 가능하다. 제안하는 구조에서는 4

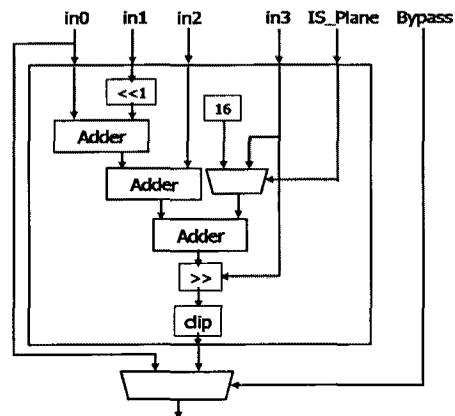


그림 6. 공통 연산기의 구조
Fig. 6. Architecture of processing element.

개의 공통 연산기를 사용하여 한 클록에 1x4 픽셀의 연산을 수행하여 4x4 블록을 계산하는데 최대 네 사이클이 소요된다. 공통 연산기는 덧셈과 시프트 연산만으로 구현하였으며, 블록 내의 공통적이고 반복적인 부분은 공통 연산기의 결과를 임시 저장하여 재사용할 수 있게 하여 연산량을 감소시켰다. 또한 수직 모드, 수평 모드, DC 모드와 같이 동일한 값으로 예측이 수행되는 경우는 바이패스 경로를 이용하여 추가적인 연산 없이 예측을 수행하도록 설계하였다. 바이패스 경로를 통해 4x4 블록 연산을 한 사이클에 수행 가능케 함으로써 전체적인 연산량과 수행 사이클 수를 감소시켰다.

나. 전처리 연산기

인트라 예측 모드 중 많은 이웃블록의 값을 필요로 하거나 복잡한 연산을 필요로 하는 DC 모드와 플레인 모드를 위하여 전처리 연산기를 설계하였다. 복잡한 예측 연산을 공통 연산기를 통한 예측 연산이 수행되기 전에 전처리 연산 과정을 거치도록 함으로써 인트라 예측 연산의 복잡도와 공통 연산기의 복잡도를 줄였다. 전처리 연산기는 곱셈 연산을 사용하지 않고 덧셈과 시프트 연산만을 사용하여 구현하였으며, 전처리 과정을 거쳐서 연산된 값은 다음 픽셀 예측 시 재사용할 수 있게 함으로써 동일한 연산의 반복 수행을 줄여 예측 연산 사이클 수를 감소시켰다. 또한, 전처리 과정 후 동일 매크로 블록 내의 예측 연산은 공통 연산기만을 통해서 간단하게 예측 가능하다.

(1) DC 모드의 전처리 과정

DC 모드는 현재 예측하고자 하는 블록의 위쪽 블록과 왼쪽 블록 픽셀의 평균값으로 예측이 이루어지며, 인트라 예측 모드 중 많은 이웃블록의 픽셀 값을 참조하는 방법이다. 16x16 휘도 예측의 DC 모드는 최대 32개의 이웃블록의 픽셀 값을 필요로 하기 때문에 많은 연산량이 요구된다. 따라서 DC 모드 예측을 위해 이웃블록 픽셀들의 평균값을 전처리 과정을 통해 미리 연산을 수행한다. DC 모드의 전처리 연산 과정을 한번 수행한 후 동일 블록의 픽셀 값을 예측할 때 별도의 연산을 수행하지 않고 전처리 연산 결과 값을 재사용할 수 있도록 하여 반복적인 예측 연산의 수행을 감소시켰다. DC 모드의 전처리 수행 사이클 수가 표 1에 나타나 있다. 16x16 휘도 블록을 제외한 나머지 블록은 모두 하나의 사이클 내에 DC 모드의 전처리 연산이 가능하다.

표 1. DC 모드의 전처리 연산 사이클 수
Table 1. Number of precomputation cycle in DC mode.

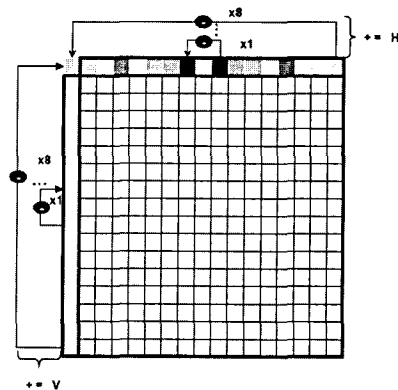
MB type	이용 가능한 이웃 블록	수행 사이클수
4x4	왼쪽 블록	1
	위쪽 블록	1
	왼쪽 블록, 위쪽 블록	1
16x16	왼쪽 블록	2
	위쪽 블록	2
	왼쪽 블록, 위쪽 블록	4
8x8	왼쪽 블록	1
	위쪽 블록	1
	왼쪽 블록, 위쪽 블록	1

전처리 연산 후 공통 연산기의 바이패스 경로를 사용하여 예측 연산을 수행하도록 하여 전체적인 예측 연산 수행 사이클 수를 감소시켰다.

(2) 플레인 모드의 전처리 과정

16x16 휘도 예측과 8x8 색차 예측의 플레인 모드는 선형 플레인 함수가 수평 및 수직 방향에 사용되는 예측 방법으로 인트라 예측 모드 중에서 가장 복잡한 연산과정을 필요로 하는 예측 방법이다. 16x16 휘도 예측의 플레인 모드가 그림 7에 나타나 있으며, 8x8 색차 예측의 플레인 모드는 그림 8에 나타나 있다.

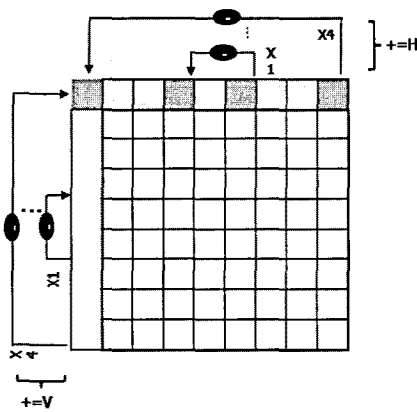
수식에 나타난 바와 같이 플레인 모드 예측은 복잡한 연산 과정을 통해 이루어지며, 또한 많은 연산량을 필



$$pred16x16i[x, y] = Clip((a + b \times (x - 7) + c \times (y - 7) + 16) \gg 5)$$

- $a = 16 \times (p[-1, 15] + p[15, -1])$
- $b = (5 \times H + 32) \gg 6$
- $c = (5 \times V + 32) \gg 6$
- $H = \sum_{x=0}^7 (x'+1) \times (p[8+x', -1] - p[6-x', -1])$
- $V = \sum_{y=0}^7 (y'+1) \times (p[-1, 8+y'] - p[-1, 6-y'])$

그림 7. 16x16 휘도 예측 - 플레인 모드
Fig. 7. 16x16 luma prediction - plane mode.



$$pred_{8x8c}[x, y] = Clip((a + b \times (x - 3) + c \times (y - 3) + 16) \gg 5)$$

- $a = 16 \times (p[-1, 7] + p[7, -1])$
- $b = (17 \times H + 16) \gg 5$
- $c = (17 \times V + 16) \gg 5$
- $H = \sum_{x'=0}^3 (x'+1) \times (p[4+x', -1] - p[2-x', -1])$
- $V = \sum_{y'=0}^3 (y'+1) \times (p[-1, 4+y'] - p[-1, 2-y'])$

그림 8. 8x8 색차 예측 - 플레인 모드
Fig. 8. 8x8 chroma prediction - plane mode.

수를 감소시켜 전체적인 연산량이 감소되었다. 플레인 모드의 전처리 수행 과정은 그림 9와 같다.

16x16 휘도 블록의 플레인 모드 전처리 과정은 H와 V 변수 연산에 각각 두 사이클을 소요하고, b 변수 연산에 한 사이클, a와 c 변수 연산에 한 사이클, 마지막으로 씨드 값 연산에 한 사이클을 소요하여 총 일곱 사이클을 사용한다. 8x8 색차 블록의 플레인 모드 전처리 과정은 H와 V 변수 연산에 각각 한 사이클을 소요하고, b 변수 연산에 한 사이클, a와 c 변수 연산에 한 사이클, 마지막으로 씨드 값 연산에 한 사이클을 소요하여 총 다섯 사이클을 사용한다.

다. 레지스터 컨트롤러

인트라 예측은 예측 연산 수행 시 이웃블록의 값을 참조하여 예측 연산을 수행한다. 만약 예측 연산마다 연산에 필요한 이웃블록의 값을 참조하기 위해 외부 메모리를 접근한다면 연산시간이 오래 걸릴 뿐만 아니라 전력 소비 면에서도 비효율적이다. 따라서 인트라 예측 연산만을 위한 내부 메모리를 사용하고, 레지스터를 효율적으로 사용함으로써 외부 메모리와의 접근을 최소화 하였다. 제안하는 레지스터의 구조가 그림 10에 나타나 있다. 레지스터의 구조는 크게 Neighboring 레지스터와 Window 레지스터로 구성된다.

Neighboring 레지스터는 내부 메모리에 저장된 이웃 블록 픽셀 값과 SUM 블록을 통해 복호가 완료된 픽셀 값을 전달받아 현재 예측하고자 하는 매크로 블록의 이웃블록 픽셀 값을 저장한다. Window 레지스터는 동일한 매크로 블록 안에서도 참조하는 이웃블록이 다른

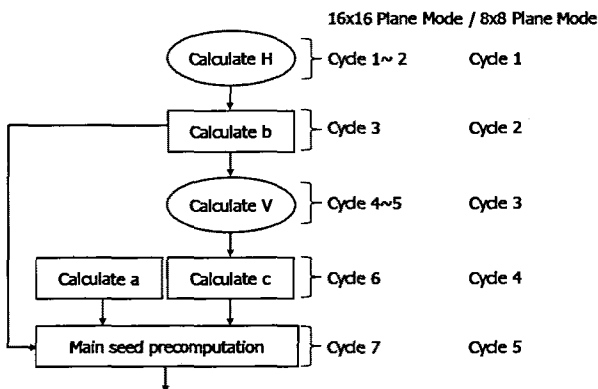


그림 9. 플레인 모드의 전처리 과정
Fig. 9. Precomputation process of plane mode.

요로 하는 예측 방법이다. 플레인 모드 예측 연산은 픽셀 값을 예측하기 위해 미리 연산되어야 하는 변수들이 존재하는데, 이 변수들을 공통 연산기를 통한 예측이 수행되기 전에 전처리 과정을 통하여 미리 연산한다.

플레인 모드의 전처리 연산 과정을 수행한 결과 값을 씨드 값으로 저장하여 이 값을 이용하여 나머지 픽셀의 예측을 수행할 수 있게 하였다^[2~3]. 16x16 휘도 예측일 경우나 8x8 색차 예측일 경우에도 플레인 모드 전처리 과정은 동일 블록에 대해 한 번만 수행하므로 복잡한 연산을 여러 번 수행하지 않아 동일한 연산의 반복 횟

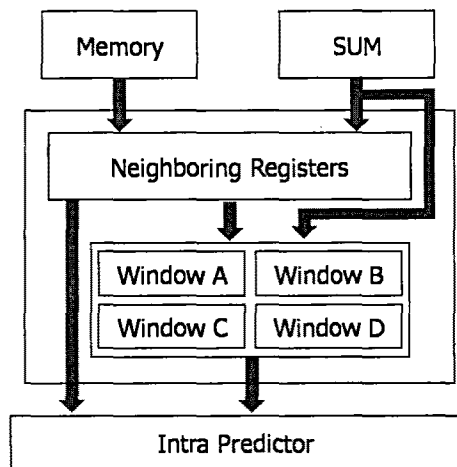


그림 10. 레지스터의 구조
Fig. 10. Architecture of registers.

4x4 휘도 예측을 위하여 Neighboring 레지스터와 SUM 블록을 통하여 픽셀 값을 전달 받아 4x4 블록 단위에 맞는 이웃블록 픽셀 값을 저장한다. 예측 완료된 픽셀 값을 내부 레지스터에 저장하여 사용함으로써 메모리 접근을 줄이고, Neighboring 레지스터와 Window 레지스터를 통하여 예측 모드 연산에 따라 참조하는 이웃블록 픽셀 값을 적절하게 전달 받음으로써 각 모드의 예측이 효율적으로 이루어진다.

IV. 성능 평가

1. 테스트 벡터 생성

제안한 인트라 예측기를 검증하기 위한 테스트 벡터는 JM9.4 부호화 소스를 이용하여 생성하였다. JM9.4 부호화 소스를 이용하여 다양한 YUV 파일에 대해 부호화를 수행하였다. 테스트 벡터로 사용된 YUV 파일이 그림 11에 나타나 있다. YUV 파일들의 크기는 176X144이고 베이스라인 프로파일로 부호화 되었다. 이진 형식의 파일 포맷을 가진 부호화된 파일을 텍스트 포맷으로 변환하여 인트라 예측기의 테스트 입력으로 사용하여 복호화를 진행하였다. 복호화를 마친 출력 값을 텍스트 포맷으로 저장하고 이를 다시 이진 형식의



그림 11. 테스트 벡터로 사용된 YUV 파일
Fig. 11. YUV file which used for test vector.

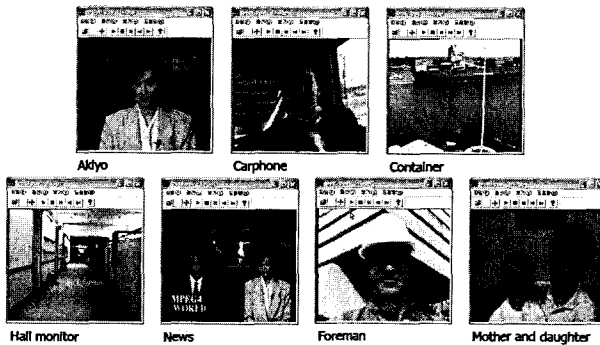


그림 12. 복호 후의 YUV 파일
Fig. 12. YUV file after decoding.

파일로 변환하여 복호화 결과를 확인하였다. YUV 파일이 재생 가능한 소프트웨어로 복호화 된 파일을 읽어 들인 결과 부호화 이전 영상과 복호 이후 영상이 동일하게 출력되는 것을 확인하였다. 복호 후의 영상이 그림 12에 나타나 있다.

2. 구현 결과

제안한 인트라 예측기는 Chartered 0.18um CMOS 공정을 이용해 합성한 결과 최대 100MHz에서 동작 가능하다. 제안한 인트라 예측기의 하드웨어 사이즈를 비교 분석한 결과를 표 2에 나타내었고, 예측 연산 수행 사이클 수를 비교 분석한 결과를 표 3에 나타내었다. 130nm 공정의 [4]와는 절대 비교가 어렵지만, [5]에 비해 비교적 적은 게이트 수를 가진다. 예측 연산 수행에

표 2. 인트라 예측기의 게이트 수 비교
Table 2. Comparison of intra predictor gate count.

	[4]	[5]	제안
공정	130nm	180nm	180nm
게이트 수	9,464	184,322	154,611
최대 동작주파수	100.9MHz	-	100MHz

표 3. 인트라 예측기의 예측 연산 사이클 비교
Table 3. Comparison of intra predictor computation cycle.

Mode	Clock Cycle			Reduction Ratio(%)	
	제안	[4]	[5]		
4x4	Vertical	1	4	11	82.95
	Horizontal	1	4	11	82.95
	DC	2	6	15	76.67
	Diagonal down left	4	4	11	31.82
	Diagonal down right	4	4	11	31.82
	Vertical right	4	4	11	31.82
	Horizontal down	4	4	11	31.82
	Vertical left	4	4	11	31.82
	Horizontal up	4	4	11	31.82
16x16	Vertical	16	64	91	78.71
	Horizontal	16	64	91	78.71
	DC	20	78	91	76.19
	Plane	71	279	94	49.51
	8x8	Vertical	4	16	48
Horizontal		4	16	48	83.33
DC		8	18	40	67.78
Plane		21	87	60	70.43
Average	11.1	38.8	39.2	60.09	

필요한 사이클 수는 표 3에 제시된 바와 같이 기존의 결과에 비해 평균적으로 60% 이상 감소하는 것으로 나타났다.

V. 결 론

본 논문에서는 인트라 예측을 효율적으로 수행하기 위하여 H.264/AVC 복호기를 위한 인트라 예측기의 하드웨어 구조를 제안하고 설계 및 검증 하였다. 제안한 인트라 예측기는 크게 전체 인트라 예측기의 동작을 제어하며, 전처리 연산기를 포함하는 인트라 예측 컨트롤러 부분과 실제 예측연산을 수행하는 공통 연산기 부분, 인트라 예측만을 위한 내부 메모리, 내부 메모리의 값을 예측 연산에 맞게 컨트롤 해주는 레지스터 컨트롤러 부분으로 구성된다. 공통 연산기를 통해 블록내의 공통적이고 반복적인 부분을 재사용함으로써 예측 연산량을 감소시켰으며, 전처리 연산기를 통해 복잡한 예측 모드 연산의 반복 수행을 줄여 예측 연산 수행 사이클 수를 감소시켰다. 또한 레지스터를 적절하게 사용함으로써 외부 메모리와의 접근을 최소화하였다. 제안한 인트라 예측기는 Verilog-HDL을 이용하여 설계하였고 적절한 테스트 벡터를 통하여 검증하였다. 예측 연산 수행 사이클 비교 분석 결과 기존의 인트라 예측기보다 우수함을 확인하였다.

참 고 문 헌

- [1] 박기현, 이론과 실무의 조화 코덱의 세계로의 초대, 홍릉과학출판사, 2007년 2월.
- [2] Ke Xu and Chiu-Sing Choy, "A Power-Efficient and Self-Adaptive Prediction Engine for H.264/AVC Decoding", IEEE Transactions on Very Large Scale Integration Systems, Vol. 16, No. 3, pp. 302-313, March 2008.
- [3] Yu-Wen Huang, Bing-Yu Hsieh, Tung-Chien Chen and Liang-Gee Chen, "Analysis, Fast Algorithm, and VLSI Architecture Design for H.264/AVC Intra Frame Coder", IEEE Transactions on Circuits and Systems For Video Technology, Vol. 15, No. 3, pp. 378-401, March 2005.
- [4] 심재오, 이선영, 조경순, "공통연산부를 공유하는 H.264 디코더용 인트라 예측 회로 설계", 대한전자공학회는논문지, 제45권 SD편, 제9호, 103-109쪽, 2008년 9월
- [5] 박중식, 이성수, "H.264 복호기에서 움직임 보상기와 연계하여 메모리 접근면에서 효율적인 인트라 예측기 설계", 대한전자공학회는논문지, 제45권 SD편, 제6호, 37-42쪽, 2008년 6월
- [6] J. V. Team, Advanced Video coding for generic audiovisual services, ITU-T Recommendation H.264 and ISO/IEC 14496-10 AVC, May 2005.

저 자 소 개



김 옥(학생회원)
 2008년 한밭대학교 공과대학
 정보통신공학과 공학사
 2010년 한밭대학교 정보통신전문
 대학원 정보통신공학과
 공학석사
 2010년 현재 (주)실리콘웍스 개발
 본부 연구원

<주관심분야 : SoC 플랫폼 설계 및 검증, 영상신호처리>



류 광 기(정회원)
 1986년 한양대학교 공과대학
 전자공학과 공학사
 1988년 한양대학교 대학원
 전자공학과 공학석사
 2000년 한양대학교 대학원
 전자공학과 공학박사
 1991년~1994년 육군사관학교 교수부 전자공학과
 전임강사
 2000년~2002년 한국전자통신연구원 시스템
 IC 설계팀 선임연구원
 2003년~현재 한밭대학교 정보통신공학과 부교수
 <주관심분야 : SoC 플랫폼 설계 및 검증, 하드웨어/소프트웨어 통합설계 및 검증, 멀티미디어 코덱 설계>