

논문 2010-47SD-5-1

H.264 부호화기를 위한 Intra-prediction & DCTQ Hardware 구조

(The Architecture of Intra-prediction & DCTQ Hardware for H.264 Encoder)

서 기 범*

(Kibum Suh)

요 약

이 논문에서는, 풀 HD 영상을 실시간에 처리가능한 새로운 화면 내 예측 및 DCTQ 하드웨어구조를 제안한다. 화면내 예측, 4x4 을 처리하기 위한 예측과 변환, 양자화, 역양자화, 역변환및 복원의 전체 cycle 을 줄일 수 있는 방법을 제안한다. 4x4 예측 부호화 cycle을 줄이기 위해, 양자화과정을 예측 사이클에서 적용할 수 있도록 하였으며, 회로의 크기를 줄이기 위하여 9 가지 모드 중 2개의 모드를 먼저 선택하는 알고리즘을 사용하였다. 또한 16x16 예측과 8x8 예측 과정을 하나의 코어를 이용하여 설계하므로 크기를 줄였다. 제안된 구조는 108Mhz 클럭에서 full HD영상을 30frame/sec에서 동작하며, 한 매크로블록의 처리 cycle 은 425 cycle이다.

Abstract

In this paper, the novel architecture of Intra-prediction & DCTQ hardware, which can process for the Full HD image(1920x1088@30fps) in realtime, is proposed. The cycle optimization method for the overall cycle of prediction, transform, scaling, descaling, and reconstruction is proposed. To reduce the cycle in the 4x4 prediction, the quantization process is performed during the prediction cycle and pre-selection of 2 modes among the 9 modes is performed to reduce the hardware area. To reduce the hardware of 16x16 and 8x8 prediction, the sharing logic between 2 prediction is utilized. The proposed architecture can process the 30frame/sec of full HD image in 108 MHz clock and operate 425 cycle for one macroblock.

Keywords : H.264/AVC, Intra prediction

I. 서 론

H.264/AVC는 ITU-T와 ISO/IEC가 함께 표준화 작업을 진행시켜 얻어낸 새로운 비디오 압축 표준 방식으로 기존의 동영상 압축 방식과는 달리 높은 압축 효율과 다양한 전송 환경에서도 에러에 강한 특성을 지닌 최신의 영상 압출 표준화 방식이다.^[1] H.264/AVC의 가

장 큰 장점은 압축 효율이 높다는 것인데, MPEG-2나 MPEG-4와에 비해 압축률이 2배 이상 향상되었다.

이렇게 향상된 데에는 다양한 크기의 움직임 보상 및 1/4 화소 단위의 움직임 예측, 화면 내 예측, 정수 기반 DCT 변환, 디블록킹 필터와 엔트로피 부호화 같은 기술을 적용했기 때문이다. 이러한 새로운 기술을 채택하거나 기존 기술을 향상시킴으로 처리해야 할 연산량이 매우 증가하게 되었다. 따라서 실시간에 Software 적으로 HD급영상을 처리하기에는 많은 연산량 때문에 이를 하드웨어로 처리하는 부호화기의 설계하기 위해서는 하드웨어 기반의 구조 설계가 요구되고 있다.^[2-4]

* 정회원, 우송대학교 철도전기시스템학과
(Department of Railroad Electrical System Engineering, Woosong University)
접수일자: 2009년11월29일, 수정완료일: 2010년4월15일

본 논문에서는 H.264에서 도입한 화면 내 예측(intra prediction)기술에 대해서 HD급을 지원하기 위한 하드웨어 구조를 제안한다. H.264/AVC의 화면 내 예측 부호화는 현재 화면 정보를 가지고 압축을 하는 기술로서 4x4, 8x8, 16x16으로 여러 가지 예측 방법을 가지고 있다. 이중에 가장 연산량이 많은 4x4 예측은 9가지 예측 모드를 사용하여 예측하며, 각 모드에 대해서 에러값을 비교한 후 가장 작은 에러값을 갖는 모드에 대하여 DCTQ/IQIDCT를 수행하여 복원과정을 통하여 복원 값을 얻어야 다음 4x4블록의 이웃 화소 값의 예측 값 계산이 가능하다. 한 매크로블록은 16개의 작은 4x4블록으로 나누어지는데, 4x4 예측일 경우 한 블록마다 DCTQ/IQIDCT를 수행해야한다. 그에 비해 16x16, 8x8 예측은 4가지 예측모드를 가지고 각각 모드에 Hadamard 변환을 사용하여 에러 값을 비교한 후 가장 좋은 모드를 결정하면 된다. 따라서, 많은 연산량을 가지는 4x4 화면 내 예측 부호화를 HD급을 지원하기 위해 기존 구조(예측 후 변환)를 변경하지 않으면 긴 수행시간을 갖는다는 것을 알 수 있다.

따라서 본 논문에서는 긴 수행시간을 갖는 4x4 예측 사이클을 줄이기 위한 구조를 제안하였다.

본 논문의 구성은 다음과 같다. II장에서는 화면 내 예측 개요 및 기존 하드웨어 구조를 소개하고 새로운 하드웨어 구조의 필요성을 언급한다. III장에서는 적용된 알고리즘과 제안하는 하드웨어 구조에 대해서 이야기하고 IV장에서는 기존에 설계된 구조와 비교 분석을 하며, 마지막으로 V장에서는 결론을 맺는다.

II. 화면 내 예측 부호화

1. 화면 내 예측 부호화 개요

H.264/AVC에서 사용하는 화면 내 예측 부호화 방식은 휘도성분에 대해 4x4, 16x16 예측모드와 색차성분에 대해 8x8 예측모드로 나누어진다. 휘도성분에 대한 예측방법 중에 휘도 4x4 화면 내 예측 부호화 경우 그림 1의 왼쪽과 같이 16x16화소를 4x4화소 단위로 자른 16개의 블록을 0~15 순서로 처리한다. 예를 들면, 그림 1의 오른쪽 그림은 12번째 블록을 예측하기 위해 필요한 주변 블록을 보여주고 있다. A(좌측)는 9번 블록에서 4화소, B(상단)는 6번 블록에서 4화소, C(우측 상단)는 7번 블록에서 4화소, D(좌측 상단)는 3번 블록에서 1화소를 사용하여 블록내의 4x4화소의 값을 예측 부호화한

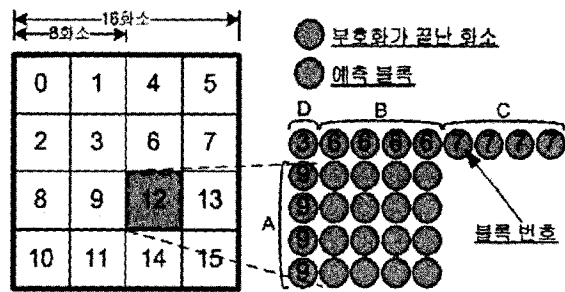


그림 1. 휘도 4x4 예측 부호화 및 부호화 순서
Fig. 1. The 4x4 prediction coding and its coding order.

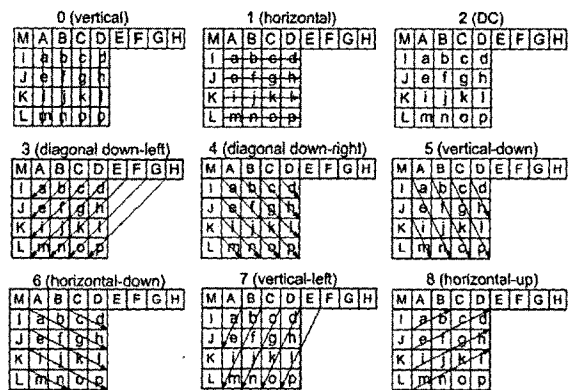


그림 2. 휘도 4x4 예측 부호화의 9가지 모드
Fig. 2. The 9 mode for the 4x4 prediction coding.

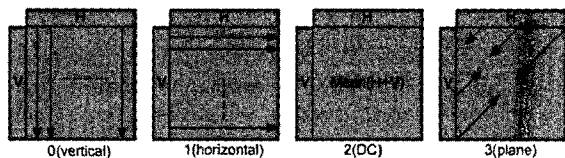


그림 3. 휘도 16x16 예측 부호화의 4가지 모드
Fig. 3. The 4 mode for the luma 16x16 prediction.

표 1. 색차 8x8 예측모드
Table 1. The prediction mode for the chroma 8x8 block.

모드 0	DC
모드 1	horizontal
모드 2	vertical
모드 3	plane

다. 예측방향은 그림 2와 같이 9가지 예측방향을 갖는데 가장 적절한 예측방향을 4x4화소 블록마다 1개씩 선택하고, 선택된 예측방향을 4x4화소의 블록단위로 부호화한다.

휘도 16x16 예측 부호화에서는 상단 매크로블록의 16화소와 좌측 매크로블록의 16화소로부터 그림 3과 같이 4가지 모드로 예측 부호화한다. 16x16화소를 한 번에 처리하고 4x4화소 단위로 DC 성분만 따로

hadamard 변환을 한다.

색차신호의 화면 내 예측은 휘도 16x16 예측과 예측 모드의 번호가 다른 것을 제외하고 처리하는 방법이 유사하다. 16x16 과 다른 점은 화소가 1/2인 8x8화소를 처리한다는 것과 두 개의(Cb and Cr)의 색차 성분이 항상 같은 예측 모드를 갖는다는 것이다.

2. 기존 화면 내 예측 부호화 구조

그림 4는 기존 설계된 하드웨어 구조이다^[6].

설계된 구조를 보면 휘도 성분의 경우 4x4예측과 16x16예측이 동시에 계산하는 것 알 수 있다. 4x4예측에서는 한 매크로블록 당 16개의 세부 블록으로 나누어 TQ/IQIT를 수행해야 예측이 가능하다는데 있다. 이전 세부 블록이 부호화되지 않았다면 다음 세부 블록은 예측할 수가 없다. 기존에 설계된 4x4예측 사이클을 그림 5에 나타내었다. 한 블록을 처리하는데 34사이클이 소요되는데 총 16 블록을 처리하므로 544 사이클이 소요되는 것을 알 수 있다. 그림 5.(b)와 (c)는 각각 휘도 16x16예측과 색차 8x8예측에 소요되는 사이클을 나타낸다. 소요되는 사이클을 보면 휘도 16x16예측에는 총

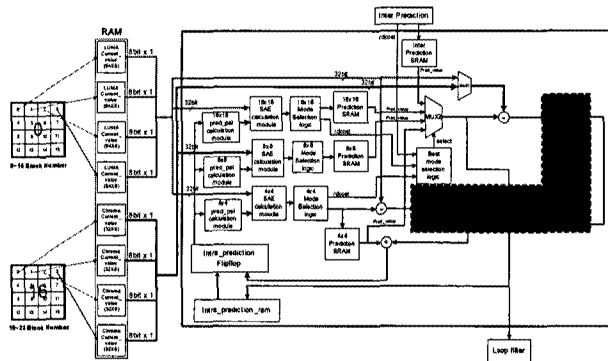


그림 4. 기존 H.264 화면내 예측 하드웨어 구조
Fig. 4. The previous architecture of H.264 intra_prediction hardware.

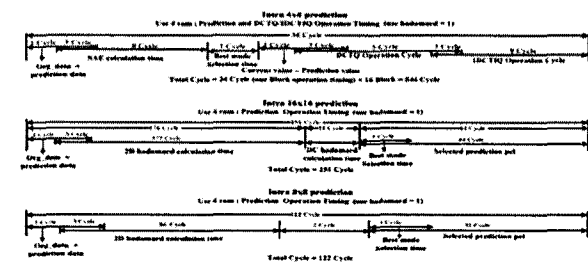


그림 5. 기존구조의 화면내 예측 처리 사이클
Fig. 5. The processing cycle of the prediction for the previous architecture.

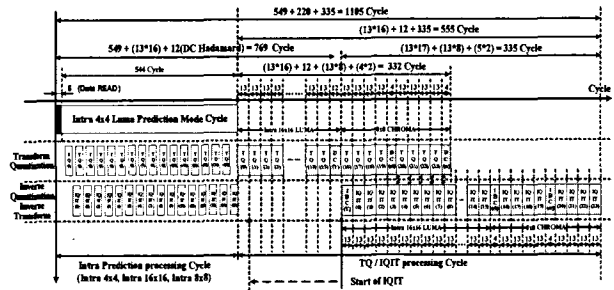


그림 6. 화면내 예측및 TQ/IQIT 의 기존 사이클
Fig. 6. The previous cycle of intra prediction and TQ/IQIT.

251 사이클, 색차 8x8예측에는 122 사이클이 소요된다. 휘도 4x4 예측이 끝나기 전까지 남은 기다린 후 TQ/IQIT를 수행하는데 그림 6과 같이 555 사이클이 소요된다. 한 매크로블록을 처리하는데 총 사이클은 1104 = 5(data READ) + 544(예측) + 555(변환) 소요된다.

3. 새로운 화면 내 예측 부호화 구조의 필요성

기존에 설계된 하드웨어 구조는 54MHz에서 D1급까지는 처리가 가능했지만 HD급을 지원하기에는 부족하다. 하드웨어 구조변경 없이 기존 사이클로 HD급을 처리하기 위해서는 280MHz에서 계산하면 가능하지만 처리 속도를 높이는 것은 한계가 있다. 그림 7를 보면 108MHz에서 HD급을 지원하기 위해서는 화면 내 예측 부호화와 TQ/IQIT를 합친 사이클이 440 사이클 안에 계산을 끝내야 가능함을 알 수 있다. 기존에 제안된 구조에서는 휘도 4x4예측에 소요되는 시간만으로도 100 사이클이 초과된다. TQ/IQIT까지 계산하는 사이클 수까지 합친다면 600사이클 이상 초과된다. 처리속도를

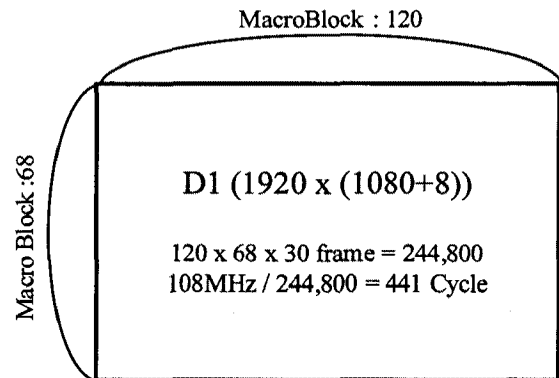


그림 7. Full HD급 영상을 처리하기 위한 요구되는 사이클
Fig. 7. The required cycle to process full HD resolution image.

떠나서 총 1104사이클이 소요되는 기존 하드웨어에서 440 사이클 내에 처리하는 하드웨어로 변경을 하기위해서는 구조적 변경이 불가피하다.

III. 적용된 알고리즘과 제안하는 하드웨어 구조

그림 8은 본 논문에서 제안하는 화면 내 예측 부호화기의 전체구조를 보여준다.

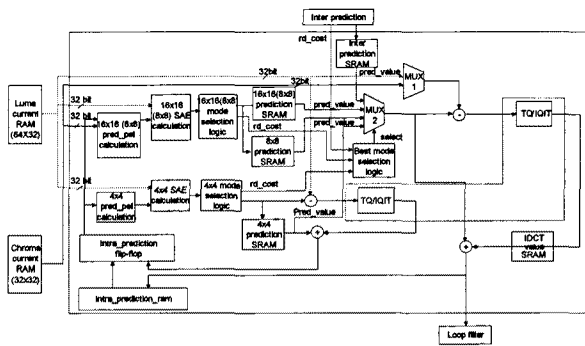


그림 8. 제안하는 화면 내 예측 부호화 구조
Fig. 8. The proposed intra prediction coding architecture.

기존 구조(그림 4)와 제안하는 구조(그림 8)을 비교하여 보면 휘도 4x4 예측과정 전에 Fast Mode Decision 모듈이 있고 SAE calculation과정에서 DCT계산이 추가되었고 TQ/IQIT로 들어가는 신호선이 생겼음이 보인다. 다음으로 휘도 16x16예측과 색차 8x8예측의 구조가 통합된 것이 보일 것이다. 휘도 4x4예측과 휘도16x16예측, 8x8예측으로 나누어 제안된 알고리즘과 제안하는 구조에 대해서 설명한다.

1. 제안된 알고리즘과 구조

가. 제안된 알고리즘

휘도 4x4예측 부호화 사이클을 줄이기 위해 양자화 시점을 best mode가 결정되기 전으로 옮김으로 해서 9 가지 모드에 대해 각각의 양자화 모듈이 필요하게 된다. 한 개의 양자화 모듈의 gate크기는 29,247 gate이므로 9개의 양자화 모듈을 사용하는 것은 하드웨어 사이즈를 너무 크게 만들기 때문에 Fast Mode Decision Algorithm^[6] 적용하였다.

적용한 Fast Mode Decision Algorithm 알고리즘은 9 가지 모드를 계산하는 SAE calculation 이전에 예측할 4x4화소 값을 가지고 두 가지의 모드 값을 미리 결정하

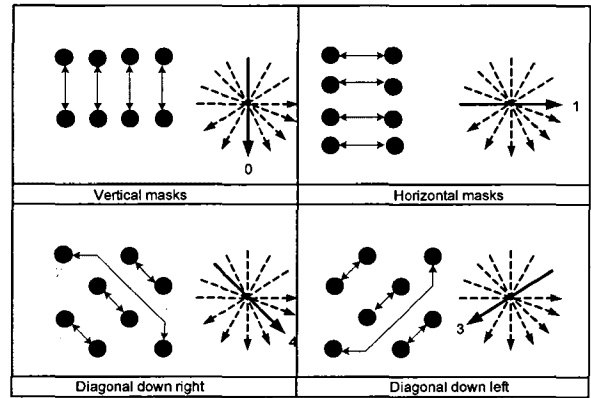


그림 9. Fast Mode Decision 알고리즘
Fig. 9. Fast Mode Decision Algorithm.

표 2. 4가지 모드의 계산식
Table 2. The calculation equation for 4 mode.

Num	Mode name	Equations for cost
0	Vertical	$cost = a-i + b-j + c-k + d-l $
1	Horizontal	$cost = a-c + e-g + i-k + m-o $
3	Diagonal down left	$cost = b-e + g-j + l-o + d-m $
4	Diagonal down right	$cost = c-h + f-k + i-n + a-p $

표 3. 휘도 4x4예측 후보모드 결정
Table 3. The determination of candidate mode.

Minimum cost modes	Second Minimum cost modes	Candidate modes
Modes 0	Modes 1	Modes 0, 7
Modes 0	Modes 3	Modes 0, 5
Modes 0	Modes 4	Modes 0, 7
Modes 1	Modes 0	Modes 1, 8
Modes 1	Modes 3	Modes 1, 6
Modes 1	Modes 4	Modes 1, 8
Modes 3	Modes 0	Modes 3, 7
Modes 3	Modes 1	Modes 3, 7
Modes 3	else	Modes 3, 2
Modes 4	Modes 0	Modes 4, 6
Modes 4	Modes 1	Modes 4, 5

는 알고리즘이다. 그림 9는 Fast Mode Decision를 계산하는 방법을 그림으로 보여준 것이고 표 2는 계산방법을 수식으로 적은 것이다.

표 2에서 Num 0, 1, 3, 4는 Vertical, Horizontal, Diagonal down left, Diagonal down right와 각각 대응

표 4. 경계구역을 고려한 2모드 선택
Table 4. 2mode selection in the boundary region.

Conditions	Model1	Model2
up_available == 0 && left_available==0	Modes = 1(invalid)	Modes = 2
up_available == 0	Modes = 1	Modes = 2
left_available ==0	if (Select mode = 0) Modes = 0 if (Select mode = 3) Modes = 3 if (Select mode = 7) Modes = 7 else Modes = 3	Modes = 2

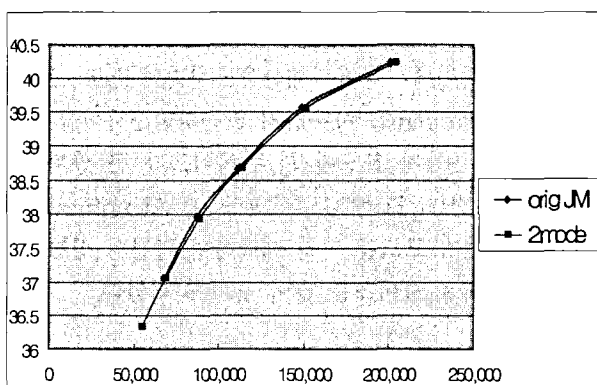


그림 10. Fast Mode Decision를 적용한 Rate Distortion Curve

Fig. 10. Rate distortion curve using the fast mode decision.

한다. 4가지 방법으로 계산하여 가장 작은 값을 Minimum cost modes라 하고 두 번째 작은 값을 Second Minimum cost modes라 한다. 표 3은 Minimum cost modes값과 Second Minimum cost modes값에 따라 후보모드가 결정되는 방법을 나타낸다.

기존의 Fast Mode Decision Algorithm^[5]은 Frame 경계부분에서 발생할 수 있는 문제에 대해서는 처리하지 않았다. 경계영역을 처리하지 않음으로 발생하게 되는 원인은 내부화소만을 가지고 판단하기 때문이다. 어떤 문제점이 생기는지 예를 들면 Frame의 첫 번째 슬라이스일 경우 위쪽으로는 화소가 존재하지 않는다. 그렇지만 내부 화소만으로 모드를 결정하게 되면 Vertical 방향으로 모드가 결정될 수 있게 된다. 그래서 우리는 Fast Mode Decision Algorithm으로 구한 2가지 모드에 대해서 표 4와 같은 up_available과 left_available의 경계부분인지를 알려주는 신호로 조건이 맞으면 Fast Mode Decision Algorithm으로 구한 값을 강제적으로 바꾸도록 만들었다.

이렇게 구해진 2가지 모드는 각각 SATD(sum of absolute transformed difference)을 통하여 여러 값을 구한 후 작은 값을 최적 모드로 결정한다. 그림 10은 Fast Mode Decision Algorithm를 적용한 JM8.5에서 foreman_cif YUV 영상 파일을 가지고 Rate Distortion Curve를 그린 것이다. 그림을 보면 JM에 비해 약간 떨어짐을 확인 할 수 있다. 하드웨어 사이즈를 고려해서 이 정도의 화질 열화는 충분히 고려할 수 있다.

나. 적용된 구조

기존 하드웨어 transform(DCT/Hadamard)모듈에서 사용하는 transpose의 구조를 변경하였다. 그림 11은 기존 구조이다. 하나의 변환을 사용하여 두 번 계산을 수행하도록 만들었다. 예측 계산 시 4x4화서 크기로 예측하는데 두 번의 DCT/Hadamard를 통과해야만 변환이 된다. 이렇게 하나를 사용하면 4x4 블록을 계산하는 동안 다음 블록은 기다려야하게 된다. 휘도 4x4예측을 제외한 나머지 예측과 부호화 과정에서는 연속적으로 처리가 가능한데 기존 하드웨어에서 지원하지 못하였다.

제안하는 구조에서는 두 개의 변환과 transpose를 사

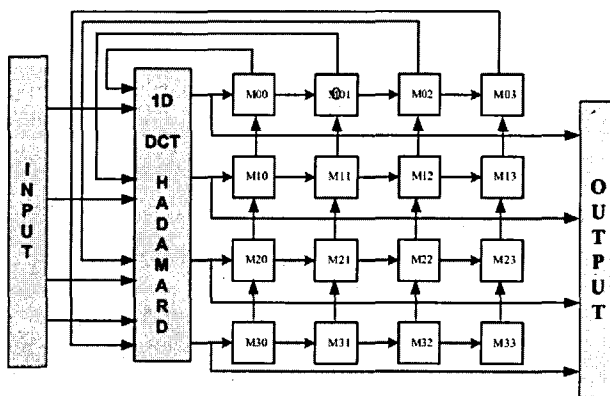


그림 11. 기존 하드웨어 transform구조
Fig. 11 The previous transformation architecture.

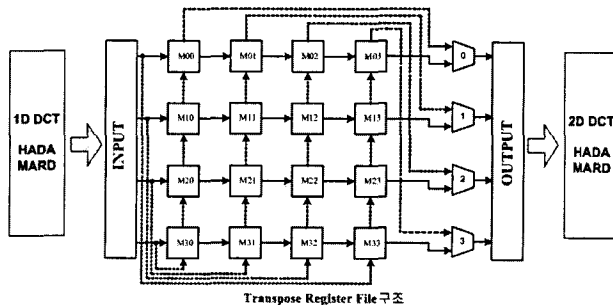


그림 12. 적용된 하드웨어 transform구조
Fig. 12. The transform architecture adopted for the proposed architecture.

용하여 연속적인 처리가 가능하도록 변경하였다.^[3] 그림 12는 변경된 transform 구조이다. 입력과 출력 단에 각각 변환기를 사용하였으며 transpose register 부분에서 M00, M01, M02, M03으로 출력하는 동안 입력을 M30, M31, M32, M33으로 받는다. 출력을 M03, M13, M23, M33으로 할 경우에는 입력을 M00, M10, M20, M30으로 받는다.

기존 transform 에서는 4x4블록을 처리하는데 입력 4사이클 출력 4사이클이 걸려 8 사이클이 소요 되었다. 적용된 transform에서는 첫 번째 데이터 처리에만 입력 사이클로 4사이클 소요되고 그 이후에는 Pipeline을 적용하여 처리하므로 출력 4사이클만이 걸리게 된다.

2. 변경된 휘도 4x4예측 구조

화면 내 부호화와 변환에 정해진 사이클은 440 이내 여야 한다. 예측 이후에 변환과정에서 140 사이클 정도 가 필요하다. 그럼 휘도 4x4예측을 300 사이클 이내에 계산되어야 한다. 300 사이클을 16블록으로 나누면 한 블록 마다 18 사이클에 계산을 되어야 한다는 결론이 나온다. 한 블록을 18 사이클에 계산하기 위해서 변경한 구조는 4가지이다.

- ① SATD를 계산하기 위한 Hadamard distortion을 DCT로 교체
- ② 예측 과정에서 예측과 변환을 동시 계산
- ③ 양자화 위치 변경
- ④ Fast Mode Decision Algorithm 적용^[2]

그림 13은 위에서 설명한 3가지를 적용한 휘도 4x4 예측 부호화의 변경된 구조이다. 입력 사이클로 7사이

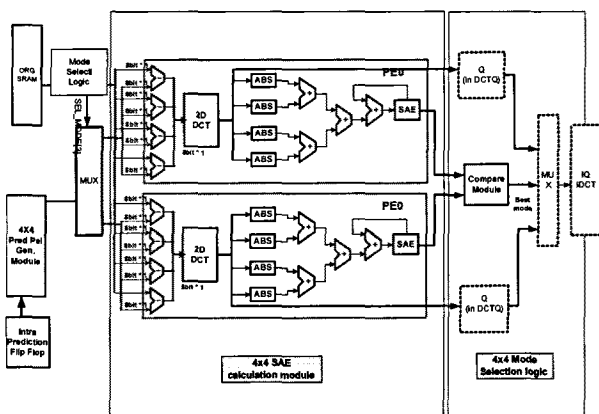


그림 13. 휘도 4x4예측 부호화의 구조
Fig. 13. The adopted 4x4 prediction architecture.

클이 걸리고 한 블록을 계산하는데 18사이클이 걸리므로 $7+18*16 = 295$ 사이클이 걸린다.

가. 양자화 위치 변경과 Fast Mode Decision Algorithm 적용

기존 구조에서는 모드가 결정되기 전에는 변환과 양자화를 수행하지 못한다. 16개의 화소를 4개 화소씩 계산하기 때문에 4 사이클이 소요된다. 모드가 결정하는데 1사이클이 소요되어 총 5 사이클의 지연이 발생하였다. 그림 14는 제안하는 구조이다. 제안하는 구조에서는 양자화 위치를 모드가 결정되기 전 Fast Mode Decision Algorithm를 적용하여 구한 두 가지 모드를 각각 양자화기에 넣어 계산을 한다. Fast Mode Decision Algorithm를 사용하지 않는다면 5사이클을 줄이기 위해서 9가지 모드에 회로 사이즈가 큰 양자화를 하나씩 써야한다. Fast Mode Decision Algorithm를 적용하여 두 가지 모드를 미리 예측하여 약간의 성능저하는 있지만 회로 사이즈를 많이 줄이게 되었다.

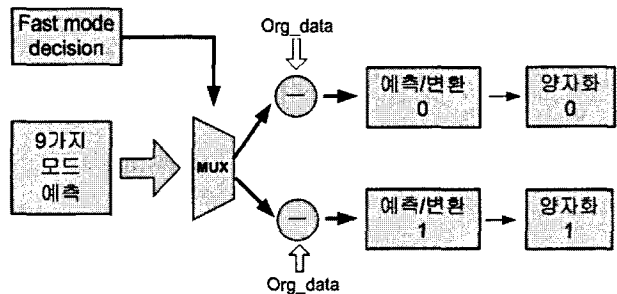


그림 14. 휘도 4x4예측모드의 선택 및 양자화
Fig. 14. The selection of 4x4 prediction mode and quantization.

나. 예측 과정에서 예측과 변환을 동시 계산

그림 15.(a)는 기존 휘도 4x4예측과정을 블록도로 그

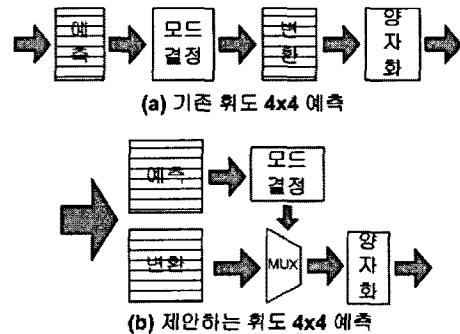


그림 15. 휘도 4x4예측의 예측모드 선택
Fig. 15. The prediction mode selection in 4x4 prediction.

린 것이다. 그림을 보면 예측과정을 거친 후 best mode 를 선택하여 한 개의 변화기를 사용하여 계산하였다. 예측에서 8 사이클, 변환에서 8 사이클이 소요 되었는데 그림 14(b)를 보면 예측과 변환을 같은 사이클에 계산하는 것을 볼 수 있다. 동시에 계산을 하므로 2개의 양자화기가 추가되어 하드웨어 사이즈 커졌지만 변환 과정에서 걸리던 8사이클을 줄일 수 있게 되었다.

3. 변경된 휘도 16x16 및 색차 8x8예측 구조

이번 절에서는 휘도 16x16예측과 색차 8x8예측의 변경된 구조에 대해서 설명한다. 휘도 4x4예측과정에서 한 블록마다 18사이클에 계산을 끝내기 위해서 변경된 구조는 하드웨어 사이즈를 증가 시켰다. 그림 16.(a)는 기존 하드웨어 구조인데 휘도 16x16과 색차 8x8예측을 따로 사용하였다. 그림 16.(b)의 제안하는 하드웨어에서는 휘도 16x16예측과 색차 8x8예측에서 계산 방법이 동일한 부분을 공동으로 사용하도록 만들었다. 이렇게 변경하게 된 이유는 휘도 16x16예측은 251 → 85 사이클에 계산이 끝나며 색차 8x8예측은 122 → 43 사이클에 계산이 끝나게 되었기 때문이다. 두 예측을 통합하여 4가지 예측모드(Vertical, Horizontal, DC, Plane)에 대한 4개의 Hadamard transform을 공동으로 사용하여 하드웨어 사이즈를 줄였다. 계산 순서는 먼저 휘도 16x16예측을 계산한 이후 색차 8x8예측을 계산한다. 총 걸리는 사이클 수를 계산하면 85(휘도) + 43(색차)= 128 사이클이 걸리지만 휘도예측에서 색차예측으로 전환 시에도 Pipeline이 가능하여 입력 사이클 4개를 줄일 수 있어 124사이클에 계산을 끝낸다.

4. DCT및 양자화및 역양자화 역 DCT 과정

예측모드의 결정이 종료된 후의 DCT및 양자화 역양자화 역DCT 과정(post processing)은 휘도성분 16x16 화소를 4x4화소 크기로 나누어 16개의 블록과 휘도 16x16예측을 고려하여 DC성분 한 블록을 계산하고 색차성분은 8x8화소 Cb, Cr를 4x4화소 크기로 나누어 8개의 블록과 DC성분2x2화소 크기 2개를 계산한다. 이때 그림 6과 같이 휘도 16블록, 색차 8블록의 한 블록을 계산하는데 13사이클이 소요되고 , 2개의 색차 DC계산하는데 8 사이클이 소요된다.

그러면 $13*(16+8)+8 = 320$ 사이클이 걸린다. 복호화에서는 휘도성분을 부호화가 끝나기를 기다린 후 휘도성분을 복호화하며 색차성분에 대해 부호화를 시작하였

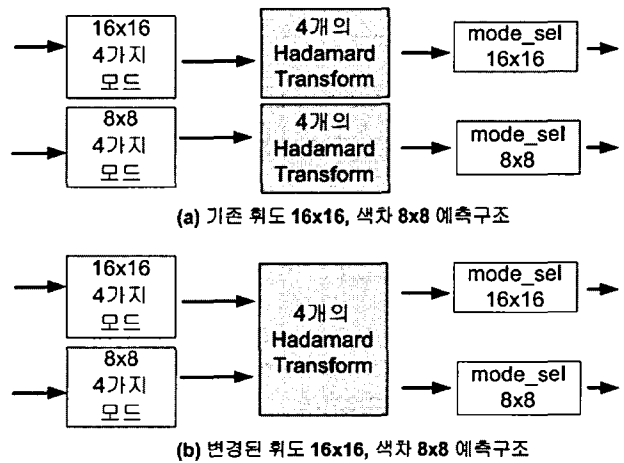


그림 16. 16x16예측과 8x8예측 구조의 공유
Fig. 16. The sharing between 16x16 prediction and 8x8 prediction.

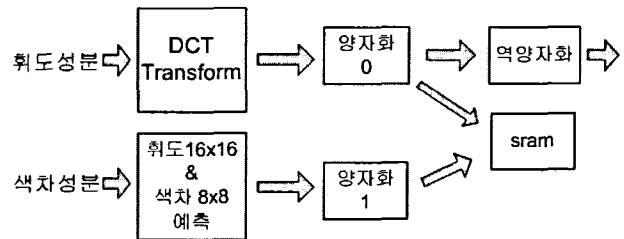


그림 17. 부호화시 사용하는 두 개의 양자화
Fig. 17. Quantization using the dual quantization unit.

다. HD급을 지원하기위해 요구하는 440사이클에서 예측 시 소요되는 295 사이클을 뺀 나머지 145 사이클에 처리하기에는 현재 구조로는 무리가 있다. 일단 예측구조에 적용한 transform을 적용하여 부호화시 Pipeline이 가능하도록 변경하였다. 변경한 결과 초기 입력 4 사이클, 양자화 계산 시 3사이클, 한 블록 처리 시 4 사이클이 소요되어 휘도 부호화 시 $7+(16*4) = 71$ 사이클, 색차 부호화 시 $7+(8*4) = 39$ 사이클이 소요 되었다. 제안한 하드웨어에서 휘도 4x4예측을 할때 양자화를 두 개 사용하였으므로, 부호화시에는 휘도성분에 양자화와 색차성분을 양자화를 동시에 수행하여 cycle 의 감소를 가져왔다. 휘도 16x16 및 색차 8x8 예측부분에서 사용되는 Hadamard transform 의 수식을 계산하고 역양자화는 나머지 하나를 사용하여 계산하였다. 색차를 계산하는 사이클을 숨김으로 부호화시 걸리는 사이클은 휘도 성분 계산 시 소요되는 71 사이클 전부이다. 그림 17은 부호화시 사용되는 두 개의 양자화를 그림으로 표현한 것이다.

부호화는 휘도와 색차 성분을 동시에 처리가 가능하지만 복호화에서 한 개의 역양자화 역변환을 사용하여

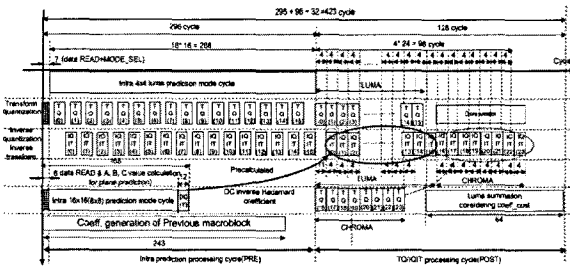


그림 18. 제안된 구조에 의한 매크로블록 처리 시간
Fig. 18. One macroblock processing time of the proposed architecture.

불가능하다. 그래서 부호화가 끝난 시점에서 SRAM에 저장을 하였다. 휘도 성분은 SRAM에 저장하는 동시에 바로 역양자화를 계산한다. 휘도 복호화 시 68 사이클이 걸리고 색차 복호화 시 37이 소요된다. 휘도성분 64 사이클이 부호화 계산하고 같은 시간대에 계산한다. 실제 부호화에 걸리는 사이클은 $68+37-64 = 41$ 사이클이다. 부호화가 끝나고 CBP를 계산하는데 64 사이클이 소요된다.

그림 18은 제안된 구조에 의한 매크로블록 처리시간을 보이고 있다.

IV. 성능 분석

표 5는 기존 하드웨어와 제안하는 하드웨어의 타임 비교이다. 기존 하드웨어는 예측(549)과 부호화(555)를 더한 1104 사이클이 소요 되었다. 제안하는 하드웨어에서는 예측(295)과 부호화(140)를 더한 435 사이클이 걸리는데 여기서 휘도 4x4예측의 마지막 블록 계산에서 애러 값이 나오는 시점을 예측의 끝으로 잡으면 10 사

표 5. 사이클 비교
Table 5. Cycle comparison.

목 차		기존 하드웨어	제안한 하드웨어
예 측	휘도 4x4예측	549 cycles	295 cycles
	휘도 16x16예측	251 cycles	124 cycles
	색차 8x8예측	122 cycles	cycles
부호화		555 cycles	140 cycles
총 계		1104 cycles	425 cycles

표 6. 하드웨어 사이즈 감소
Table 6. Hardware size reduction.

모듈명	Gates	기존 하드웨어	제안한 하드웨어	증가 /감소
양자화	29,247	1	2	+29,247
4x4변환	8,319	19	10	-74,871
총 계				-45,624

* TSMC 0.18um, Clock = 6 ns의 경우

표 7. Cycle 비교
Table 7. Cycle comparison.

	[7]	[3]	[5]	제안
cycles/MB	490	450	1115	425

이클을 줄일 수 있다. 그래서 제안하는 하드웨어에서 걸리는 사이클은 425 사이클이다.

표 6은 구조 변경으로 하드웨어 사이즈의 변화를 나타낸 것이다. Fast Mode Decision Algorithm를 사용하여 휘도 4x4예측의 9가지 모드에 대한 4x4변환의 7개 수를 줄였고 변환을 같이 하므로 2개를 추가하여 5개를 줄였다. 휘도 16x16예측 & 색차 8x8예측에서 4개의 4x4변환을 줄였다. 총 9개의 4x4변환기를 줄였다.

전체 cycle을 비교하기 위하여, 회로의 MB 당 cycle을 표 7.에 보인다.

V. 결 론

본 논문에서는 HD급 영상을 실시간으로 처리할 수 있는 H.264/AVC를 위한 효율적인 화면 내 부호화 & transform 하드웨어설계 구조를 제안하였다. 변환과 양자화의 위치변경, Fast Mode Decision Algorithm 적용, transform 구조 변경으로 수행 시간을 줄였다. 제안된 구조를 적용하여 실험한 결과, 하나의 매크로블록을 처리하는 사이클 수는 425사이클로 기존 방식보다 38%가 줄었다. 제안된 하드웨어 구조는 Verilog HDL을 이용하여 설계 및 검증되었고, 1980x1088 영상을 30 frame/sec 영상을 108MHz에서 동작 시킬 수 있음을 확인하였다.

참 고 문 헌

[1] J. V. Team, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video

- Specification. ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC
- [2] 이찬호 “ H.264 복호기를 위한 스케일러블 인트라 예측기 구조 설계”, 2008년 11월 전자공학회 논문지 제 45 권 SD 편 제 11 호, pp1108-1113
 - [3] 허가열, 조효문, 조상복, “H.264/AVC의 인트라 예측 병렬파이프라인 실행알고리즘”, 2008년 9월 전자공학회 논문지 제 45 권 SP 편 제 5 호, pp 487-494
 - [4] 전주일, 김재민, 강현수, “개선된 H.264/AVC 인트라 예측 방법”, 대한전자공학회, 대한전자공학회 학술대회 대한전자공학회 2008년 하계종합학술대회, 2008. 6, pp. 993~994
 - [5] Kibum Suh, Seong-Mo Park, Hanjin Cho, “An Efficient Hardware Architecture of Intra Prediction and TQ/IQIT Module for H.264 Encoder”, *ETRI Journal*, vol.27, no.5, Oct. 14, 2005, pp 511-524
 - [6] abderrahmane Elyousfi, Ahmed Tamtaoui, and EL Houssine Bouyakhf “Fast Mode Decision Algorithm for Intra prediction in H.264/AVC Video Coding” *IJCSNS International Journal of Computer Science and Network Security*, vol.7 No.1, January 2007.
 - [7] 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 Circuit and Systems for video technology*, vol.15, no.3, march 2005.

 저 자 소 개



서 기 범(정회원)

1989년 한양대학교 전자공학과
학사 졸업.

1991년 한양대학교 전자공학과
석사 졸업.

2000년 한양대학교 전자공학과
박사 졸업.

2000년~2002년 한국전자통신연구원 선임연구원
2002년~현재 우송대학교 철도전기정보통신학부
교수

<주관심분야 : 집적회로설계, 영상신호통신>