논문 2008-45SD-9-15

공통연산부를 공유하는 H.264 디코더용 인트라 예측 회로 설계

(Design of Intra Prediction Circuit for H.264 Decoder Sharing Common Operations Unit)

심 재 오*, 이 선 영*, 조 경 순**

(Jaeoh Shim, Seonyoung Lee, and Kyeongsoon Cho)

요 약

본 논문은 H.264 디코더용 인트라 예측 회로 구조와 설계를 제시한다. H.264의 인트라 예측에는 총 17개의 예측 모드, 즉 루마 4x4 블록을 위한 9개의 예측 모드, 루마 16x16 블록을 위한 4개의 예측 모드, 크로마 8x8 블록을 위한 4개의 예측 모드가 있다. 모든 예측 모드에서 공통된 연산들을 추출하여 이들을 수행하기 위한 공통연산부를 정의하였다. 모든 예측 모드에서 이 연산부를 공유하는 제안된 회로 구조는 설계 측면에서 체계적이고 회로 크기 측면에서 효율적이다.

Abstract

This paper presents the architecture and design of intra prediction circuit for H.264 decoder. There are a total of 17 operational modes in the intra prediction of H.264 - nine modes for a luma 4x4 block, four modes for a luma 16x16 block and four modes for a chroma 8x8 block. We extracted common operations included in all prediction modes and defined the common operations unit to perform those operations. The proposed circuit architecture sharing this unit in all prediction modes is systematic from the design point of view and efficient in terms of circuit size.

Keywords: video compression, H.264, intra prediction, common operations unit

I. Introduction

The Joint Video Team of ISO/IEC MPEG and ITU-T VCEG proposed a video compression standard known as H.264[1]. In order to compress video data efficiently, H.264 uses many new techniques and one of them is the intra prediction[2]. Intra prediction makes use of similarities between neighboring macroblocks. It has nine modes for a luma 4x4 block, four modes for a luma 16x16 block and four modes for a chroma 8x8 block. Each prediction mode includes various combinations of operations such as addition and multiplication, and many of them require a large amount of computational efforts[3]. Hence the circuit architecture for the intra prediction should be not only efficient but also structured[4]. In order to achieve this goal, we investigated all prediction modes and found the operations common to each mode. We defined the common operations unit to compute those common operations. The proposed circuit based on this unit has four adders, one multiplier, one shifter and four 15-bit buffers, and supports all of the prediction modes defined in the H.264 standard. The number of gates in the synthesized circuit using 130nm standard cell library is 9,464 and its maximum operating frequency is 100.9MHz. This paper consists of five sections. In section II, the common operations unit, which is the

* 학생회원, ** 평생회원, 한국외국어대학교 전자정보공학부
(Department of Electronics and Information Engineering, Hankuk University of Foreign Studies)
※ 이 연구는 2008학년도 한국외국어대학교 교내학술연구비의 지원에 의하여 이루어진 것임.
접수일자: 2007년10월12일, 수정완료일: 2008년8월28일

core of the proposed architecture, is described. Section Ⅲ explains the architecture of our intra prediction circuit. Section Ⅳ shows the implementation results, and finally section Ⅴ concludes the paper.

Ⅱ. Common Operations Unit

As shown in Figure 1, the intra prediction of H.264 decoder accepts as inputs the prediction values uF'n obtained from the previously reconstructed blocks and the residual values D'n computed by the IDCT (Inverse Discrete Cosine Transform) module. H.264 defines three kinds of intra predictions for a luma 4x4 block, for a luma 16x16 block and for a chroma 8x8 block. Nine prediction modes for a luma 4x4 block are shown in Figure 2. Note that mode 2, which is the DC mode, uses the average of predictor values and does not use any directional prediction.

There are four prediction modes for a luma 16x16 block and also four prediction modes for a chroma 8x8 block. They are the DC mode, plane mode, vertical mode and horizontal mode. Among them the DC and plane modes do not use any directional prediction.

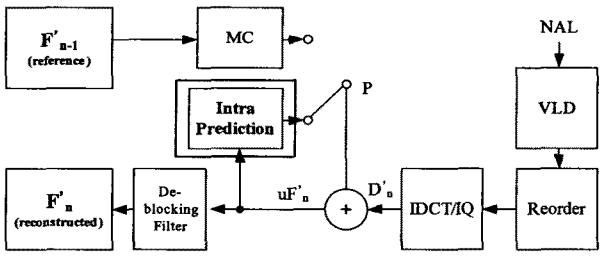

그림 1. H.264 디코더의 블록도
Fig. 1. Block diagram of H.264 decoder.

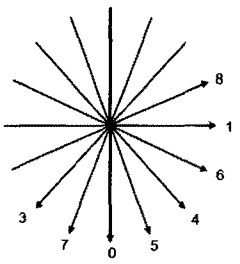

그림 2. 예측 모드와 방향성
Fig. 2. Prediction modes and corresponding directions.

We investigated the operations involved in all of 17 prediction modes and found that the following operation is common to every mode.

$$F(W,X,Y,Z,\alpha) = (W+X+Y+Z+2) >> \alpha \quad (1)$$

Figure 3 illustrates the structure of the COU (Common Operations Unit) to implement the above operation. It accepts four pixels as input, and performs four additions and one right-shift by α bits.

$$Pred_1 = (P_0 + P_1 + P_2 + P_3 + P_{16} + P_{17} + P_{18} + P_{19} + 4) >> 3 \quad (2)$$

$$Pred_2 = (P_i + P_j + P_k + P_l + 2) >> 2 \quad (3)$$

$$Pred_3 = (P_i + 2*P_j + P_k + 2) >> 2 \quad (4)$$

$$Pred_4 = (P_i + P_j + 1) >> 1 \quad (5)$$

The predictors (P_0~P_{32}) for a luma 4x4 block and for a luma 16x16 block are shown in Figure 4. Equations (2)~(5) express the prediction values of

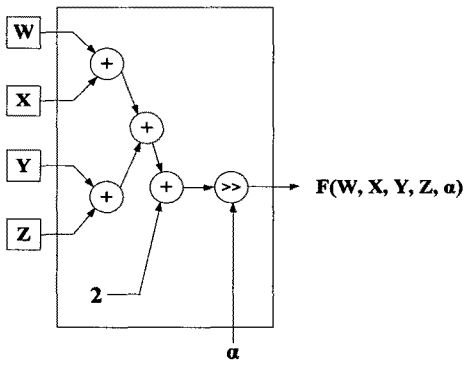

그림 3. 공통연산부의 구조
Fig. 3. Structure of common operations unit.

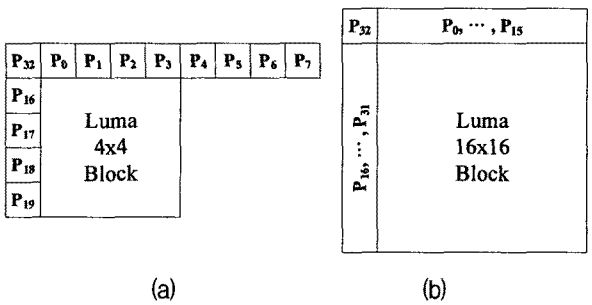

그림 4. 루마 블록의 예측값 위치
Fig. 4. Position of predictors for luma blocks.

seven modes out of nine modes (except horizontal and vertical modes) for a luma 4x4 block. The subscripts i, j, k and l represent the positions (0~32) of the predictors. These equations are rearranged into equations (6)~(9) by using the common operation F specified in equation (1).

$$\begin{aligned}\text{Pred}_1 = F\{&F(P_0,P_1,P_2,P_3,0),\\ &F(P_{16},P_{17},P_{18},P_{19},0),0,-2,3)\}\end{aligned} \tag{6}$$

$$\text{Pred}_2 = F(P_i,P_j,P_k,P_l,2) \tag{7}$$

$$\text{Pred}_3 = F(P_i,P_j,P_j,P_k,2) \tag{8}$$

$$\text{Pred}_4 = F(P_i,P_j,P_i,P_j,2) \tag{9}$$

The DC mode of a luma 16x16 block requires computing the average value of 32 pixels as shown in equation (10). This equation is also rearranged into equation (11) by repeating the common operation F eight times.

$$\text{Pred}_5 = \left(\sum_{i=0}^{31} P_i + 16\right) >> 5 \tag{10}$$

$$\begin{aligned}\text{Pred}_5 = F\Bigg(&\sum_{i=0}^{3} F(P_{4i},P_{4i+1},P_{4i+2},P_{4i+3},0),\\ &\sum_{i=4}^{7} F(P_{4i},P_{4i+1},P_{4i+2},P_{4i+3},0),0,14,5\Bigg)\end{aligned} \tag{11}$$

The plane mode of a luma 16x16 block is the most complicated one. The first stage is to compute H and V using equations (12). The second stage is to compute A, B and C using equations (13). Finally, the prediction values are obtained by equation (14). All computations are performed by applying the common operation F repeatedly. As a result, the plane mode of a luma 16x16 block requires one COU and one multiplier. All the modes for a chroma 8x8 block can be expressed in a similar way.

$$\begin{aligned}H &= \sum_{i=0}^{7} (i+1) * F(P(8+i), \sim P(6-i),1,-2,0)\\ V &= \sum_{j=0}^{7} (j+1) * F(P(8+j), \sim P(6-j),1,-2,0)\end{aligned} \tag{12}$$

$$\begin{aligned}A &= 16 * F(P_{15},P_{31},0,-2,0)\\ B &= F(5 * H,0,0,30,6)\\ C &= F(5 * V,0,0,30,6)\end{aligned} \tag{13}$$

$$\begin{aligned}&\text{Pred}_6 = F(A, B*(x-7), C*(y-7),14,5)\\ &\text{for } 0 \le x, y \le 15\end{aligned} \tag{14}$$

As shown above, COU achieves the efficiency for intra prediction operations by sharing the resources. The straightforward implementation of the intra prediction circuit following the H.264 reference program[5] will require 88 adders, 18 multipliers and 11 shifters. Since COU consists of 4 adders, 1 multiplier and 1 shifter, it is much more efficient in terms of circuit size to implement the prediction circuit sharing one COU.

We need to store the intermediate computation data to use COU more efficiently. The proposed circuit uses a register file with the minimum number of buffers to store those data. The register file consists of four 15-bit buffers (R_0, R_1, R_2, R_3). The usage of the buffers is different for each prediction mode, as shown in Table 1.

표 1. 예측 모드 별 버퍼 사용현황
Table 1. Buffer usage for each prediction mode.

Mode			R_0	R_1	R_2	R_3
Luma 4x4	0	Horizontal	X	X	X	X
	1	Vertical	X	X	X	X
	2	DC	O	O	X	X
	3	Diagonal Down Left	O	O	O	X
	4	Diagonal Down Right	O	O	O	X
	5	Vertical Left	O	O	O	X
	6	Vertical Right	O	O	O	X
	7	Horizontal Up	O	O	O	X
	8	Horizontal Down	O	O	O	X
Luma 16x16	9	Horizontal	X	X	X	X
	10	Vertical	X	X	X	X
	11	DC	O	O	O	O
	12	Plane	O	O	O	O
Chroma 8x8	13	Horizontal	X	X	X	X
	14	Vertical	X	X	X	X
	15	DC	O	O	X	X
	16	Plane	O	O	O	O

(O: Used, X: Not used)

For all the modes of a luma 4x4 block except the horizontal, vertical and DC modes, the intermediate results are stored into R_0, R_1 and R_2 and then the final results are provided at the next clock cycle as outputs. Notice that the horizontal and vertical modes inherently do not require any computation. The computation procedure for the luma 4x4 DC mode in equation (6) is as follows:

1. Compute $F(P_0,P_1,P_2,P_3,0)$ and store the result into R_0.
2. Compute $F(P_{16},P_{17},P_{18},P_{19},0)$ and store the result into R_1.
3. Compute $F(R_0,R_1,0,-2,3)$ using the data in R_0 and R_1.

The computation procedure for the luma 16x16 DC mode in equation (11) is as follows:

1. Compute $F(P_0,P_1,P_2,P_3,0)$ and store the result into R_0.
2. Compute $F(P_4,P_5,P_6,P_7,0)$ and store the result into R_1.
3. Compute $F(R_0,R_1,0,-2,0)$ using the data in R_0 and R_1, and store the result into R_0.
4. Compute $F(P_8,P_9,P_{10},P_{11},0)$ and store the result into R_1.
5. Repeat steps 3 and 4 to obtain the sum of 32 predictors.

The computation procedure for the luma 16x16 plane mode in equations (12), (13) and (14) is as follows:

1. Perform two subtract operations for H in equation (12) and store the results into R_0 and R_1.
2. Add the data in R_0 and R_1 and store the result into R_0.
3. Perform the next subtract operation and store the result into R_1.
4. Repeat steps 2 and 3 to obtain H.
5. Compute B using H in R_0 and store the results into R_2.
6. Compute V in equation (12) in the same way as H.
7. Compute C using V in R_0 and store the results into R_3.
8. Compute A and store the results into R_0.
9. Compute F in equation (14) using A, B and C.

III. Architecture

The circuit architecture proposed in this paper is shown in Figure 5. The core parts of the architecture are the COU to compute the common operation F defined in equation (1) and the register file to store the intermediate computation results. Input data selector module selects the appropriate predictors from the image data stored in the internal SRAM (Static Random Access Memory). Three SRAM blocks are used for Y, Cb and Cr image data. The selected predictors are then transferred to the COU. The output data generator module provides the output data at a rate of four prediction values per one clock cycle. OUT0 ~ OUT3 from the output data generator module are added to IDCT_OUT0 ~ IDCT_OUT3 from the IDCT module. Some of the results are stored into the SRAM to be used as predictors for the intra prediction of next blocks. The prediction controller module generates the control signals for COU, register file and output data generator modules.

The proposed intra prediction circuit is interfaced with the IDCT circuit. We assume that the IDCT circuit generates four pixels (one row in a 4x4 block) per one clock cycle. In order to avoid using a buffer or an internal memory, we designed an intra prediction circuit according to the timing shown in

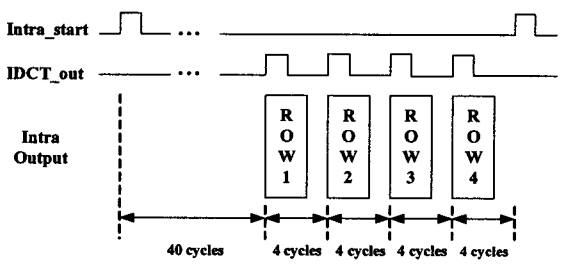

그림 6. IDCT 모듈과 인트라 예측 모듈의 인터페이스

Fig. 6. Interface between IDCT module and intra prediction module.

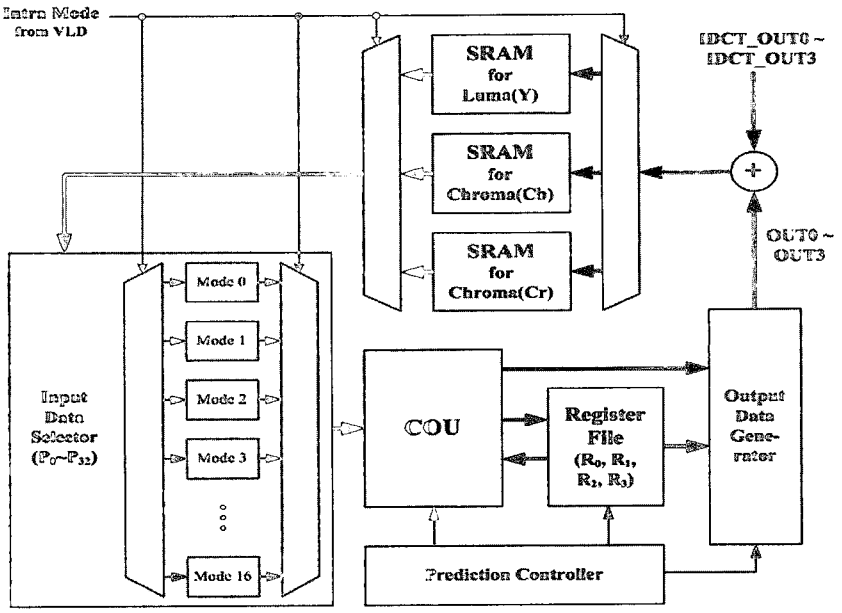

그림 5. 인트라 예측 회로의 제안된 구조
Fig. 5. Proposed architecture of intra prediction circuit.

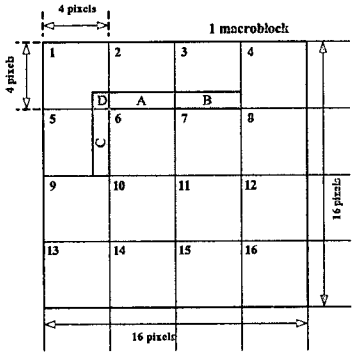

그림 7. 루마 4x4 블록의 예측값
Fig. 7. Predictors for a luma 4x4 block.

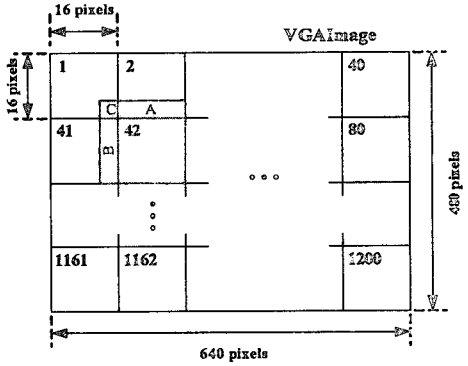

그림 8. 루마 16x16 블록의 예측값
Fig. 8. Predictors for a luma 16x16 block.

Figure 6. Intra_start is the signal to notify the beginning of intra prediction process. IDCT_out informs the output timing of the IDCT module. Some of the pixel values obtained by adding the prediction values and residual values computed by the IDCT module will be used as the predictors for next blocks.

They are stored in the SRAM for this purpose and Figure 7 illustrates those predictors in case of a luma 4x4 block. The intra prediction for luma 4x4 blocks in a 16x16 macroblock is performed according to the order specified as index numbers from 1 to 16. In this figure the predictors for block 6 are A, B, C and D, which are included in blocks 1, 2, 3 and 5 and should be stored in the SRAM. Figure 8 shows the predictors of a luma 16x16 block in an image with 640x480 pixels. While computing the pixel values for the first row (from blocks 1 to 40), the 640 pixel values in the last line should be stored in the SRAM to be used as the predictors for the second row (from blocks 41 to 80).

IV. Implementation

We described the proposed intra prediction circuit in Verilog HDL (Hardware Description Language) at RTL (Register Transfer Level). The simulation results using Cadence NC-Verilog were verified by

표 2. 예측 모드 별 매크로블록 당 클럭 사이클 수
Table 2. Number of clock cycles per macroblock in each prediction mode.

Mode	Description		Cycles
0	Luma 4x4	Horizontal	64
1		Vertical	64
2		DC	96
3		Diagonal Down Left	64
4		Diagonal Down Right	64
5		Vertical Left	64
6		Vertical Right	64
7		Horizontal Up	64
8		Horizontal Down	64
9	Luma 16x16	Horizontal	64
10		Vertical	64
11		DC	78
12		Plane	279
13	Chroma 8x8	Horizontal	16
14		Vertical	16
15		DC	18
16		Plane	87

comparing with the results from the H.264 reference program[5]. The gate-level circuit was synthesized using Synopsys Design Compiler and 130nm standard cell library. The number of gates in the synthesized circuit is 9,464 and the maximum operating frequency is 100.9MHz. The number of clock cycles required per macroblock in each prediction mode is specified in Table 2. Therefore our circuit can process up to 61 frames of image with 1280x1024 pixels per second. We could not provide the comparison results with other approaches for the size and performance of the proposed circuit because of the lack of published data. As far as authors know, our circuit is comparative and suitable to be used as an IP (Intellectual Property) for low-cost and high-performance H.264 decoders.

V. Conclusion

We proposed an efficient architecture of intra prediction circuit for H.264 decoder. The proposed circuit is based on the COU defined to compute the common operations included in all prediction modes. By using the COU and sharing buffers, the proposed architecture is not only structured but also efficient. Since our circuit is comparative in terms of its size and shows a good performance, it can be used in the design of low-cost and high-performance H.264 decoders.

참 고 문 헌

[1] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14469-110 AVC), March, 2003.
[2] T. Wiegand G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," IEEE Trans. on Circuit System for Video Technology, Vol. 13, Issue 7, pp.560-576, July, 2003.
[3] I. G. Richard, H.264 and MPEG-4 Video Compression, Wiley, 2003.
[4] Esra Sahin and Ilker Hamzaoglu, "An Efficient Hardware Architecture for H.264 Intra Prediction Algorithm," Design, Automation & Test in Europe Conference & Exhibition, pp.1-6, April, 2007.
[5] JVT H.264 Reference Software Version JM11.

저 자 소 개

심 재 오(학생회원)

2007년 2월 한국외국어대학교 전자정보공학부 학사 졸업.

2007년~현재 한국외국어대학교 전자정보공학부 석사과정.

<주관심분야 : SoC 설계>

조 경 순(평생회원)

1982년 2월 서울대학교 전자공학과 학사 졸업.

1984년 2월 서울대학교 전자공학과 석사 졸업.

1988년 12월 미국 Carnegie Mellon University 전기 및 컴퓨터 공학과 박사 졸업.

1988년 11월~1994년 8월 삼성전자(주) 반도체 총괄 선임, 수석 연구원.

1994년 8월~현재 한국외국어대학교 전자정보공학부 조교수, 부교수, 정교수.

<주관심분야 : SoC 설계>

이 선 영(학생회원)

1998년 2월 한국외국어대학교 전자정보공학부 학사 졸업

2000년 2월 한국외국어대학교 전자정보공학부 석사 졸업.

2001년 2월~2006년 4월 (주)ECT 반도체설계 연구소 선임연구원.

2000년 2월~현재 한국외국어대학교 전자정보공학부 박사과정.

<주관심분야 : SoC 설계>