A Dynamically Reconfigurable Processor for the H.264/AVC Image Prediction

Y. Hayakawa and A. Kanasugi

Graduate School of Engineering, Tokyo Denki University
2-2 Kanda Nishiki-cho, Chiyoda-ku, Tokyo, 101-8457 Japan
(Tel: 03-5280-3705; Fax: 03-5280-3565)
(Email address: 09kme40@ms.dendai.ac.jp, kanasugi@eee.dendai.ac.jp)

Abstract: H.264/AVC provides high video quality at substantially low bit rates. It is useful for a save and transfer of video images by robot cameras. However, the computational complexity of H.264/AVC is very high. A high-speed general-purpose processor is necessary to process H.264/AVC. However, it is difficult to use such a processor for a portable device. Therefore, an application specific processor is necessary. Dynamic reconfiguration can expand virtually circuit area in limited chip area. Thus, this paper proposes a dynamically reconfigurable processor for the H.264/AVC image prediction. H.264/AVC contains intra and inter prediction process. Intra and inter prediction processes are not used at the same time. The proposed dynamically reconfigurable processor reconfigures those circuits. Proposed processor was designed and synthesized. As a result, LUTs (look up tables) were reduced to 93%, flip flops were reduced to 94%, and maximum delay was about the same.

Keywords: H.264/AVC, dynamically reconfiguration, inter prediction, intra prediction

I. INTRODUCTION

H.264/AVC is the latest video compression standard[1]. H.264/AVC provides high video quality at substantially low bit rates. It is useful for a save and transfer of video images by robot cameras. However, the computational complexity of H.264/AVC is very high[2]. The video resolution is proportional to the frame rate of application. Furthermore, video resolution increases every year. A high-speed general-purpose processor is necessary to process H.264/AVC. However, it is difficult to use such a processor for a portable device. Therefore an application specific processor is necessary.

H.264/AVC contains intra and inter prediction process, deblocking filter process, quantization process, integer discrete cosine transform process, encoding process, decoding process, inverse quantization process, and inverse integer discrete cosine transform process. Intra and inter prediction processes are not used at the same time. Intra and inter prediction process circuits are implemented independently by the general decoder.

Dynamic reconfiguration can expand virtually circuit area in limited chip area. Although reconfiguration requires temporary stop of circuit in a few milliseconds, dynamic reconfiguration changes circuit construction during operation without stopping circuit. Therefore, we can design circuit of many functions in small circuit[3-4].

Therefore, this paper proposes a dynamically reconfigurable processor for the H.264/AVC main profile image prediction.

II. H.264/AVC MAIN PROFILE INTRA AND INTER PREDICTION

H.264/AVC contains intra and inter prediction process, deblocking filter process, quantization process, integer discrete cosine transform process, encoding process, decoding process, inverse quantization process, and inverse integer discrete cosine transform process. Intra and inter prediction processes are not used at the same time by the general decoder. Intra prediction process uses neighboring samples for *N* by *N* block (for example 4 by 4, 16 by 16, etc). Inter prediction process uses reference picture (namely, before and after current picture). Intra and inter prediction process circuits are implemented independently.

1. Inter prediction process

Most inter prediction process is sample interpolation process. Luminance (luma) sample interpolation process and chrominance (chroma) sample interpolation process are different. Although luma sample interpolation process calculates quarter samples using 6-tap filter, chroma sample interpolation process calculates 1/8 samples. Luma sample interpolation process needs 448 times addition for a 4 by 4 block. Chroma sample interpolation process needs 96 times addition for a 2 by

2 block. Figure 1 shows luma sample interpolation circuit.

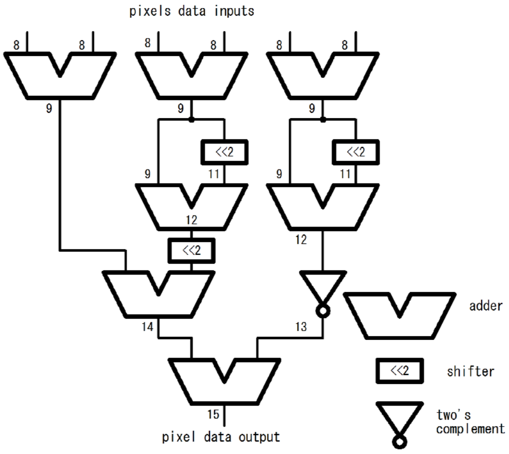

Fig. 1 Luma sample interpolation circuit

The 6-tap filter for luma sample interpolation process calculates as follows.

$$\begin{cases} p_{t1}=A-5B+20C+20D-5E+F & (1) \\ \quad A\sim E\text{: 8 bit integer samples} & \\ p_1=\text{Clip}((p_{t1}+16)/32) & (2) \end{cases}$$

$$\begin{cases} p_{t2}=G-5H+20I+20J-5K+L & (3) \\ \quad G\sim L\text{: 15 bit filtered samples } (p_{t1}) & \\ p_2=\text{Clip}((p_{t2}+512)/1024) & (4) \end{cases}$$

$$\text{Clip}(p)=\begin{cases} 0, & p<0 \\ p, & 0\le p\le 255 \\ 255, & p>255 \end{cases} \qquad (5)$$

Chroma sample interpolation process for one sample calculates as follows.

$$\begin{cases} p_{t3}=O\cdot(8-x)(8-y)+P\cdot(x)(8-y)+Q\cdot(8-x)(y)+R\cdot(x)(y) & (6) \\ \quad O\sim R\text{: 8 bit integer samples} & \\ p_3=\text{Clip}((p_{t3}+32)/64) & (7) \end{cases}$$

2. Intra prediction process

Intra prediction process uses top and left neighbor samples. Intra prediction process consists of luma intra prediction process for a 4 by 4 block, luma intra prediction process for a 16 by 16 block, and chroma intra prediction process for an 8 by 8 block. Luma intra prediction process for a 4 by 4 block contains three calculations as indicated by the following.

16 times sum of 2 values.

16 times sum of 4 values.

A sum of 8 values.

Luma intra prediction process for a 16 by 16 block contains two calculations as indicated by the following.

A sum of 32 values.

$$\begin{cases} p[x,y]= & \\ \quad \text{Clip}((a+b\cdot(x-7)+c\cdot(y-7)+16)>>5), & \\ \quad \text{with } x,y=0,1,\ldots,15 & (8) \\ a=16\,(p[-1,15]+p[15,-1]) & (9) \\ b=(5H+32)>>6 & (10) \\ c=(5V+32)>>6 & (11) \\ H=\sum_{x'=0}^{7}(x'+1)(p[8+x',-1]-p[6+x',-1]) & (12) \\ V=\sum_{y'=0}^{7}(y'+1)(p[-1,8+y']-p[-1,6+y']) & (13) \end{cases}$$

Chroma intra prediction process contains two calculations as indicated by the following.

A sum of 8 values.

$$\begin{cases} p[x,y]= & \\ \quad \text{Clip}((a+b\cdot(x-3)+c\cdot(y-3)+16)>>5), & \\ \quad \text{with } x,y=0,1,\ldots,7 & (14) \\ a=16\,(p[-1,7]+p[7,-1]) & (15) \\ b=(34H+32)>>6 & (16) \\ c=(34V+32)>>6 & (17) \\ H=\sum_{x'=0}^{3}(x'+1)(p[4+x',-1]-p[2+x',-1]) & (18) \\ V=\sum_{y'=0}^{3}(y'+1)(p[-1,4+y']-p[-1,2+y']) & (19) \end{cases}$$

III. PROPOSED PROCESSOR

Intra and inter prediction process circuits are implemented independently by general decoder. In this paper, circuit area was reduced by dynamically reconfiguration for these circuits. The proposed circuit is based on 13 luma sample interpolation process, because this process is the largest. This process consists of 91 adders. The connections of adders are reconfigured by the multiplexers. Some circuits did not incorporate, because circuit areas increase. These 70 adders were reduced by reconfiguration.

The proposed dynamically reconfigurable processor reconfigures luma sample interpolation process, chroma sample interpolation process, luma intra prediction process, and chroma intra prediction process. Luma intra prediction process has 13 modes. Chroma intra prediction process has 4 modes. This processer calculates luma sample interpolation process for a 4 by 4 block in 19 clock cycles. This processer calculates chroma sample interpolation process for a 2 by 2 block in 2 clock cycles. This processer calculates luma intra prediction process for a 4 by 4 block in 4 clock cycles at the maximum. This processer calculates luma intra prediction process for a 16 by 16 block in 66 clock cycles at the maximum. This processer calculates chroma intra prediction process for an 8 by 8 block in 18 clock cycles at the maximum.

Proposed circuit has 13 blocks. The connections of those blocks and adders are reconfigured by the multiplexers. Those 13 blocks were numbered. Six blocks are almost the same type (No.6~12). Figure 2 shows block diagram of those circuits. Figure 3 shows block diagram of three blocks (No.0, 1, and 2). Those blocks are used for luma sample interpolation process, luma intra prediction process, and chroma intra prediction process. For example, luma intra prediction process for a 16 by 16 block uses shaded adders. Those adders calculate H in (12). Figure 4 shows block diagram of three blocks (No.3, 4, and 5). Those blocks are almost the same as the previous description blocks (No.0, 1, and 2). However, those can calculate p in (8). Those blocks are used for luma sample interpolation process, luma intra prediction process, and chroma intra prediction process. For example, luma intra prediction process for a 16 by 16 block uses shaded adders. Those adders calculate p in (8). Nine blocks (No.0~8) have 6 by 8 bits inputs at the minimum. However, 4 blocks (No.9~12) have 6 by 15 bit inputs at the minimum. Because, processor have to calculate p_{t2} in (3). In addition, those blocks (No.9~12) calculate p_{t1} in (1).

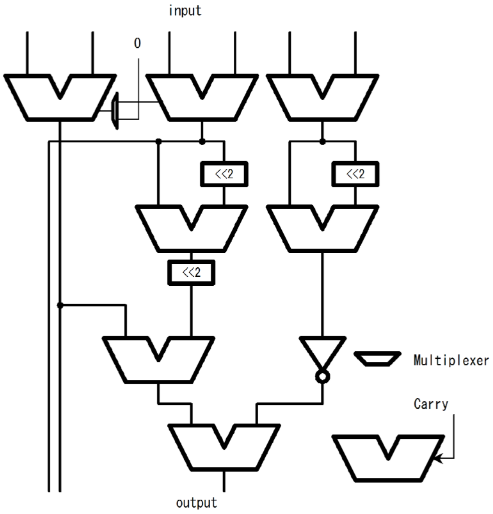

Fig. 2 The block diagram of one block (No.6 ~12)

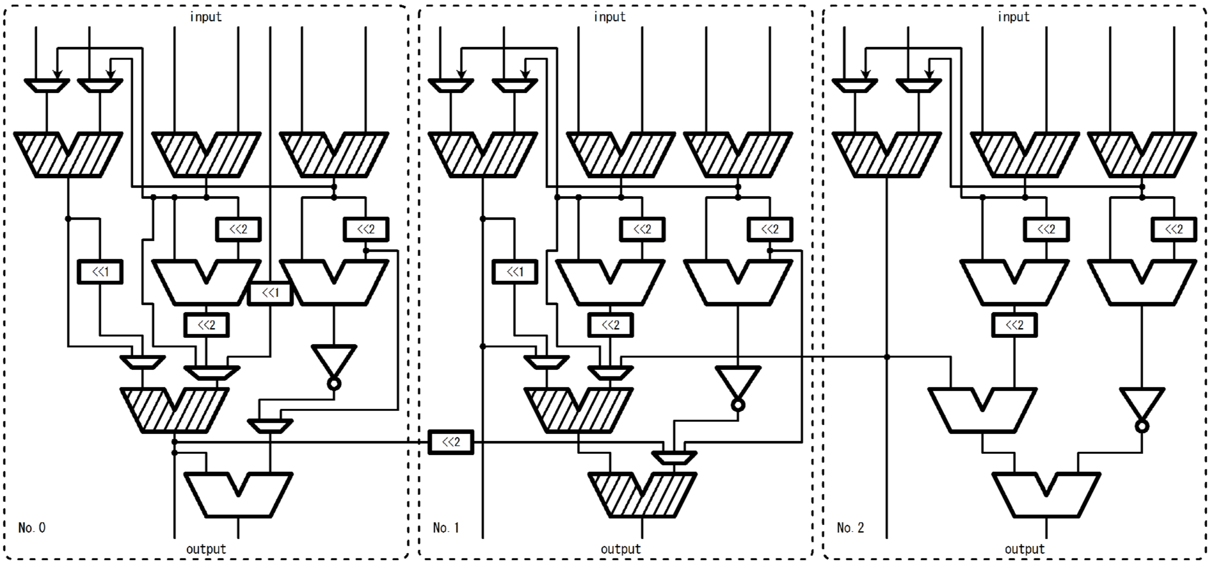

Fig. 3 The block diagram of three blocks (No.0, 1, and 2)

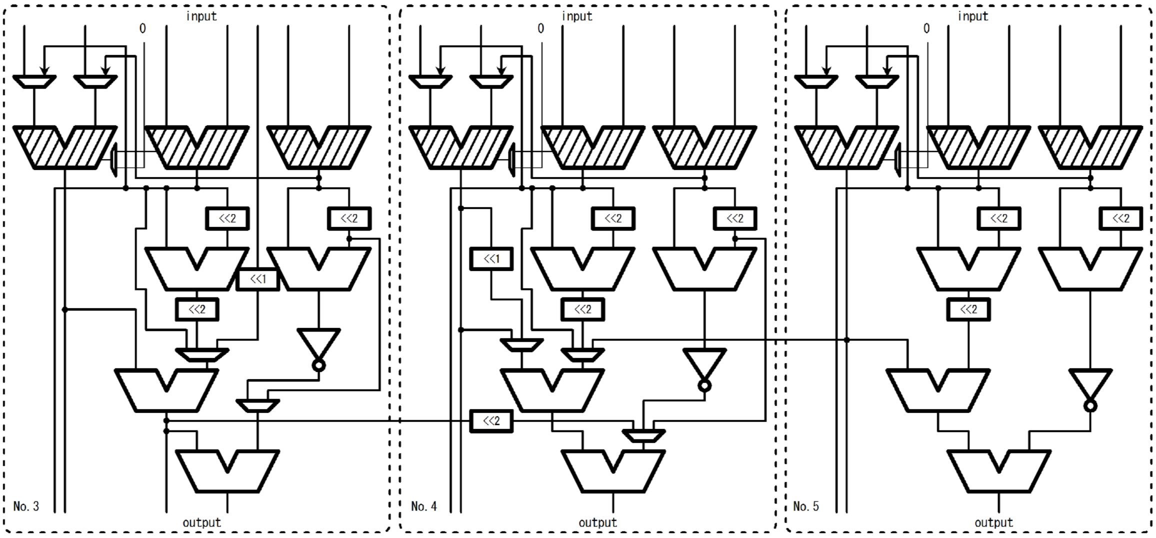

Fig. 4 The block diagram of three blocks (No.3, 4, and 5)

Reconfigurations of 13 blocks and connection of 13 blocks are controlled by control unit. The inputs of control unit are samples data, mode (luma intra prediction process, luma intra prediction process for a 16 by 16 block, and so on) select signals, and so on. Calculation results are output four pixels (32 bits) at a time.

IV. EVALUATION

The dynamically reconfigurable prediction circuit for H.264/AVC decoding was synthesized using Xilinx ISE 11.1 CAD software. The target FPGA (Field Programmable Gate Array) is Virtex-5 of Xilinx Corp. (XC5VLX50T). The result was compared with a circuit without dynamically reconfiguration. These circuits can calculate in the same clock cycles. Table 1 summarizes the logic synthesis results. As a result, LUTs (look up tables) were reduced to 93%, flip flops were reduced to 94%, and maximum delay was about the same.

Table 1. Processor synthesis results

	LUTs	Flip flops	Delay[ns]
Proposed	4181	1608	11.678
General	4508	1705	11.878
Rate	0.927	0.943	0.983

V. CONCLUSION

This paper proposed a dynamically reconfigurable processor for the H.264/AVC image prediction. The proposed processor contains 13 blocks. The proposed processor reconfigures luma sample interpolation process, chroma sample interpolation process, and intra prediction process. Seventy adders were reduced by reconfiguration. The proposed processor was designed and synthesized. The result was compared with a circuit without dynamically reconfiguration. As a result, LUTs (look up tables) were reduced to 93%, flip flops were reduced to 94%, and maximum delay was about the same.

ACKNOWLEDGEMENTS

This work was supported by Tokyo Denki University Science Promotion Fund (Q09J-01).

REFERENCES

[1] ITU-T Recommendation H.264 (2005), Advanced Video Coding for Generic Audiovisual Service

[2] S. Chien, Y. Huang, C. Chen, H. Chen, and L. Chen (2005), Hardware architecturedesign of video compression for multimedia communication systems, IEEE Commun. Mag., vol. 43, no. 8, pp. 123–132

[3] T. Sato, H. Watanabe and K. Shiba (2005), Implementation of dynamically reconfigurable processor DAPDNA-2, VLSI Design, Automation and Test, 2005 IEEE VLSI-TSA International Symposium, pp. 323-324

[4] T. Sugawara, K. Ide and T. Sato (2004), Dynamically Reconfigurable Processor Implemented with IP Flex's DAPDNA Technology, IEICE Trans. Inf. & Syst., D, E87 (8), pp. 1997-2003