

논문 2006-43SD-10-17

H.264 변환 및 양자화 기능을 갖는 AMBA 기반 IP 구현

(Implementation of an AMBA-Based IP for H.264 Transform and Quantization)

이 선 영*, 조 경 순**

(Seonyoung Lee and Kyeongssoon Cho)

요 약

본 논문은 H.264 비디오 압축 표준에서 필요로 하는 순방향 및 역방향 변환과 양자화를 수행할 수 있는 AMBA 기반 IP에 대해서 기술하고 있다. 변환과 양자화 회로는 면적과 성능 측면에서 최적화되어 있으며, AMBA를 기반으로 동작하기 위해서 AHB 래퍼 회로가 추가되었다. IP가 버스를 점유하는 시간과 비디오 데이터를 외부 메모리의 어느 위치에 저장할 것인지를 사용자가 지정할 수 있도록 설계하였다. Xilinx FPGA와 ARM9 프로세서를 장착한 플랫폼 보드를 사용하여 제안된 IP가 AMBA 표준에 근거하여 동작하는지를 검증하였다. 0.25 μ m 표준 셀을 사용하여 이 회로를 MPW 칩으로 제작하고 동작을 확인하였다.

Abstract

This paper describes an AMBA-based IP to perform forward and inverse transform and quantization required in the H.264 video compression standard. The transform and quantization circuit was optimized for area and performance. The AHB wrapper was added to the circuit for the AMBA-based operation. The user of the IP can specify how long the bus may be occupied by the IP and also where the video data are stored in the external memory. The function of the proposed IP based on AMBA Specification was verified on the platform board with Xilinx FPGA and ARM9 processor. We fabricated an MPW chip using 0.25 μ m standard cells and observed its correct operations on silicon.

Keywords : transform, quantization, H.264, video CODEC, AMBA, IP

I. Introduction

Digital convergence is one of the conspicuous trends these days. A cell phone with a digital camera and an electronic dictionary with an MP3 CODEC (coder and decoder) are the examples of digital convergence. These digital products require a complex SoC (system on chip) with a variety of functions. A fast time to market is, of course, mandatory. The platform-based design methodology

is one of the solutions to this problem. If we want to use this methodology successfully, two factors should be kept in mind. One is to use the IP's (intellectual properties) with good quality. The other is to use the IP's with standard interface. In this paper we present the design and verification results of the IP conforming to a standard bus interface protocol: the AMBA (advanced microcontroller bus architecture) Specification^[1]. The proposed IP can perform the transform and quantization required in H.264^[2]. We implemented the transform circuit using an area-optimized architecture^[3]. It can perform all the transform operations defined in H.264 including forward and inverse, residual and DC transforms using small number of resource blocks such as

* 학생회원, ** 정회원, 한국외국어대학교 전자정보공학부 (Department of Electronics and Information Engineering, Hankuk University of Foreign Studies)
 ※ 이 연구는 2006학년도 한국외국어대학교 교내학술 연구비의 지원에 의하여 이루어진 것임.
 접수일자: 2006년6월27일, 수정완료일: 2006년9월7일

adders and buffers. Note that all three transforms can be implemented using only additions and shifts without any multiplications^[4]. The quantization circuit works closely in association with the transform circuit. The platform on which the IP operates includes arbiter, decoder, multiplexer, master and slave IP's. We described the whole platform including the proposed IP at RTL (register transfer level) in Verilog HDL (hardware description language) and verified the AMBA-compliant transform and quantization operations through the RTL simulations and also on the platform board. We synthesized the proposed module into the gate-level circuit with 0.25 μ m standard cells using Design Compiler, designed the chip layout using Astro and implemented an MPW (multi-project wafer) chip. We tested the fabricated chip and observed its correct operations compliant to the AMBA Specification.

II. Circuit Design for Transform and Quantization

1. Transform Circuit

The 2D (two-dimensional) transforms defined in H.264/AVC can be carried out by applying 1D (one-dimensional) transform equations to the input data, transposing the intermediate results and applying the same equations to the transposed data. The 1D transform equations of 4x4 forward residual transform are given by:

$$\begin{bmatrix} Y0 \\ Y1 \\ Y2 \\ Y3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} X0 \\ X1 \\ X2 \\ X3 \end{bmatrix} \quad (1)$$

The above equations can be decomposed into two parts^[5].

$$\begin{bmatrix} Y0 \\ Y2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} X0+X3 \\ X1+X2 \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} Y1 \\ Y3 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} X0-X3 \\ X1-X2 \end{bmatrix}$$

The 1D transform equations of 4x4 inverse residual

transform are decomposed in a similar way. We decomposed the 1D transform equations of 4x4 luma DC transform and the 2D transform equations of 2x2 chroma DC transform. All the decomposed equations share the same 2x2 coefficient matrices.

The 1D residual and luma DC transforms and the 2D chroma DC transform can be implemented using two adders (one for addition and the other for subtraction) and shifters. A transpose buffer is necessary to implement the 2D residual and luma DC transforms. Fig. 1 illustrates the proposed architecture of the transform circuit. It includes a 14x16-bit transpose buffer, an adder, a subtractor, four registers (R₀, R₁, R₂ and R_m), multiplexers and a control block. The bit width of the components in the circuit should be 16 to cover all the transforms^[2]. The input signal 'mode' provides the information to determine the kind of transform such as forward or inverse, luma or chroma, etc. Based on this signal, the control block controls the data flow within the transform circuit.

Fig. 2 and Fig. 3 show the parts of the circuit to perform the forward and inverse residual transforms among the operations of the circuit in Fig. 1. The 2D transform is performed by applying the 1D transform twice. The inputs of the first 1D transform are supplied to the input port 'din' at a rate of one pixel per clock. These pixels are stored in the registers temporarily and provided to the adder and subtractor at the properly scheduled time. The results are stored in the transpose buffer and then used as the inputs of the second 1D transform. The final outputs are available at the output ports 'Y_e' and 'Y_o'.

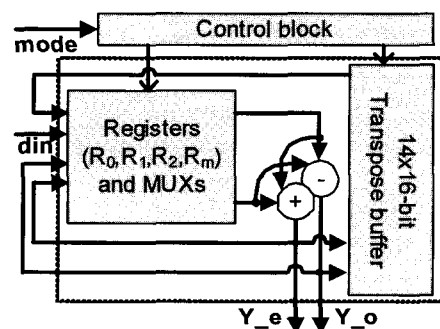


그림 1. 제안된 변환 회로의 구조

Fig. 1. Proposed architecture of the transform circuit.

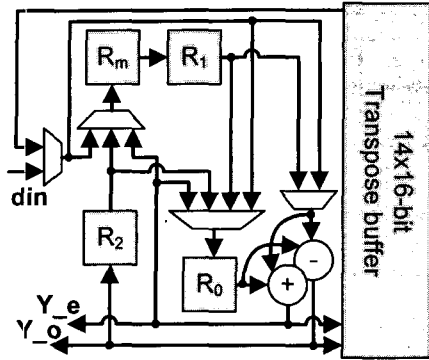


그림 2. 순방향 레지듀얼 변환 회로의 블록도
 Fig. 2. Block diagram of the forward residual transform circuit.

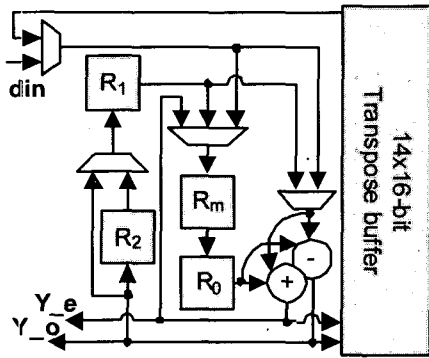


그림 3. 역방향 레지듀얼 변환 회로의 블록도
 Fig. 3. Block diagram of the inverse residual transform circuit.

The data flow of the 1D forward residual transform is shown in Fig. 4. X_i is the input pixel supplied to the 1D transform circuit at clock i . The first input pixel X_0 is stored in R_m at clock 1 and the second input pixel X_1 is stored in R_0 at clock 2. The inputs of the first addition (denoted by '+') and subtraction (denoted by '-') are X_1 stored in R_0 and the third input pixel X_2 . The inputs of the adder and subtractor are highlighted by shading in Fig. 4. T_0 (X_1+X_2) and T_1 (X_1-X_2) are stored in R_m and R_2 at clock 3. X_0 is transferred from R_m to R_0 via R_1 and used as the input of the second addition and subtraction. T_2 (X_0+X_3) and T_3 (X_0-X_3) are stored in R_0 and R_2 at clock 4. Y_0 (T_2+T_0) and Y_2 (T_2-T_0) are the first outputs from the 1D transform circuit. T_3 and T_1 are shifted left by one bit, added and subtracted to generate the second outputs Y_1 (T_3*+T_1) and Y_3 (T_3-T_1*), where $T_3* = T_3 \ll 1$ and $T_1* = T_1 \ll 1$. These operations are repeated for the

Clock #	0	1	2	3	4	5	6	7	...
Input	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	...
R_0			X_1	X_0	T_2	T_3	X_5	X_4	...
R_1			X_0		T_0	T_1	X_4		...
R_m		X_0		T_0	T_1	X_4		T_0	...
R_2				T_1	T_3			T_1	...
Output						Y_0	Y_1		...
						Y_2	Y_3		...
+			X_1	X_0	T_2	T_3*	X_5	X_4	...
			X_2	X_3	T_0	T_1	X_6	X_7	...
-			X_1	X_0	T_2	T_3	X_5	X_4	...
			X_2	X_3	T_0	T_1*	X_6	X_7	...

$(T_3* = T_3 \ll 1, T_1* = T_1 \ll 1)$

그림 4. 1차원 순방향 레지듀얼 변환의 데이터 흐름
 Fig. 4. Data flow of the 1D forward residual transform.

rest of the input pixels. The operations of other transforms such as the inverse residual and the DC transforms are scheduled in a similar way.

All the data (Y_0, \dots, Y_{15}) generated from the 1D transform circuit should be stored in the transpose buffer. To perform the 2D transform the transposed data are fed back to the 1D transform circuit in the order of Y_0, Y_4, Y_8 and Y_{12} . The transpose buffer for the 4x4 transform usually consists of 16 registers.^[4,5] As illustrated in Fig. 5, we reduced the number of registers into 14 by carefully scheduling the transpose operations. As shown in Fig. 5 (a), our transpose buffer stores the first 12 input data (Y_0, \dots, Y_{11}) in the registers (R_0, \dots, R_{11}). The two buffer registers R_{12} and R_{13} are necessary because our 1D transform circuit generates a pair of data at a time. In order to accept the next input Y_{12} , the data that have been already stored in the register group 'loop 0' of Fig. 5 (b), i.e., Y_8, Y_4 , and Y_0 are shifted right by one. Then, Y_{12} is stored in the register R_8 . The data in 'loop 1', 'loop 2' and 'loop 3' do not move at this moment. After Y_1, Y_5 and Y_9 stored in 'loop 1', and Y_4, Y_8 and Y_{12} stored in 'loop 0' are shifted right by one, the next input Y_{13} is stored in the register R_9 . The registers of 'loop 2' and 'loop 3' are holding the previous data. Similarly, the last two inputs (Y_{14} and Y_{15}) are stored in the registers R_{10} and R_{11} . In this way we can guarantee the proper transpose

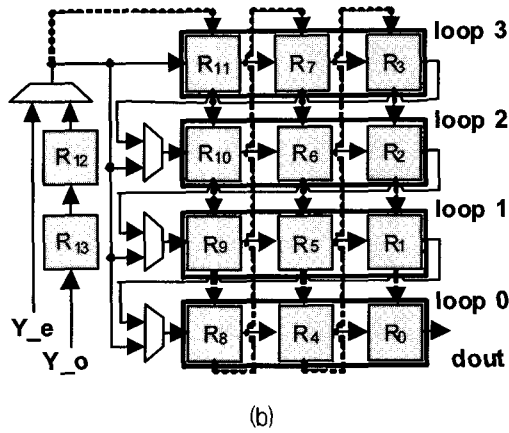
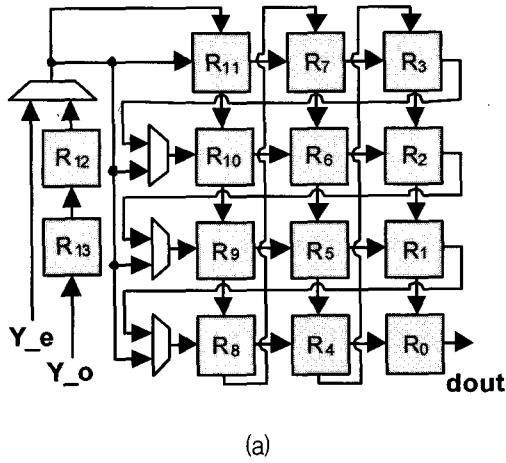


그림 5. 전치버퍼의 구조
Fig. 5. Structure of the transpose buffer.

operations with only 14 registers.

Based on the proposed architecture, we described the RTL circuit in Verilog HDL and synthesized the gate-level circuit with 3,299 logic gates using standard cell technology. The maximum operating frequency is 106MHz. A macro block in a 4:2:0 color image consists of 24 4x4 residual blocks, one 4x4 luma DC block and two 2x2 chroma DC blocks. Since 810 (32x24 + 32x1 + 2x4 + 2) clocks are required to process one macro block including two-clock latency, our circuit can process 331 frames per second for an image with 352 x 288 pixels. Table 1 compares the circuit size with other approaches. Our circuit can perform all the transforms defined in the H.264 with much smaller number of adders. The architecture proposed in [5] does not support any DC transforms even though it uses more logic gates than ours. The approach adopted in [7] shows an extreme case of a

표 1. 회로 크기 비교

Table 1. Comparison of Circuit Size.

		[5]	[6]	[7]	Ours
Forward Transform	Residual	O	O	O	O
	Luma DC	X	O	X	O
	Chroma DC	X	O	X	O
Inverse Transform	Residual	O	O	X	O
	Luma DC	X	O	X	O
	Chroma DC	X	O	X	O
Number of adders		8	16	64	2
Transpose buffer (bits)		16x16	16x16	None	14x16
Gate count		3,524	6,538	N.A.	3,299

(O: supported, X: unsupported)

parallel architecture and is not concerned with area-efficiency. Since the H.264 standard is announced, there have been many approaches to optimize each part of H.264 CODEC circuit. As far as the authors know, our circuit is the smallest among the transform circuits that have been reported.

2. Quantization Circuit

The H.264 assumes a scalar quantizer. The basic forward quantizer operation is

$$Z_{ij} = \text{round}(Y_{ij} / Qstep) \tag{3}$$

where Y_{ij} is a coefficient of the transform. $Qstep$ is a quantizer step size and Z_{ij} is a quantized coefficient. A total of 52 values of $Qstep$ are supported by the H.264 standard and indexed by a QP (quantization parameter). The simplified forward operation is described as follows.

$$\begin{aligned} |Z_{ij}| &= (|W_{ij}| \cdot MF + f) \gg qbits \\ \text{sign}(Z_{ij}) &= \text{sign}(W_{ij}) \\ qbits &= 15 + \text{floor}(QP / 6) \end{aligned} \tag{4}$$

where W_{ij} is an unscaled coefficient of the transform. The values of MF are defined in the standard. f is $2^{qbits} / 3$ for intra blocks or $2^{qbits} / 6$ for inter blocks.

Fig. 6 illustrates the architecture of the circuit that can perform forward and inverse quantization operations. The input data are supplied to the input ports 'q_in0' and 'q_in1'. A quantization parameter is

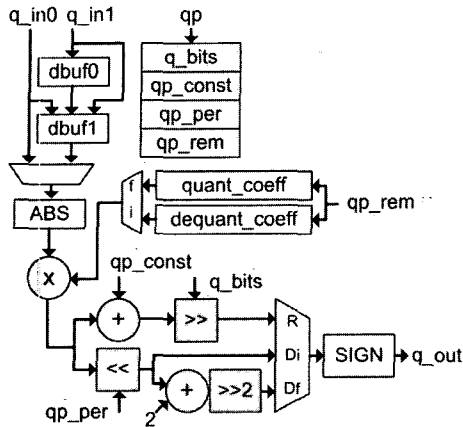


그림 6. 양자화 회로 구조
Fig. 6. Architecture of the quantization circuit.

provided to the input port 'qp' and is used to generate four parameters ('q_bits', 'qp_const', 'qp_per' and 'qp_rem'). Those parameters are internally used in the quantization circuit. The letters 'f', 'i', 'R' and 'D' used in the multiplexers in Fig. 6 denote forward, inverse, residual and DC, respectively. Since a pair of coefficients is generated per clock from the transform circuit, two registers 'dbuf0' and 'dbuf1' buffer them. The final quantization output is available at the output port 'q_out' at a rate of one coefficient per clock. We described the quantization circuit at RTL and synthesized the gate-level circuit with 3,901 logic gates using standard cell technology.

III. IP Design and Verification

1. IP Design Based on AMBA Platform

The AMBA is defined on an on-chip communications standard for designing embedded microcontrollers. Within the specification three distinct buses are defined: AHB (advanced high-performance bus), ASB (advanced system bus) and APB (advanced peripheral bus). We built the design platform to verify our transform and quantization circuit, based on the AHB architecture. Fig. 7 shows the block diagram of the platform including arbiter, decoder, multiplexer, master and slave IP's. The arbiter allows the master the authority of using a bus depending on the priority

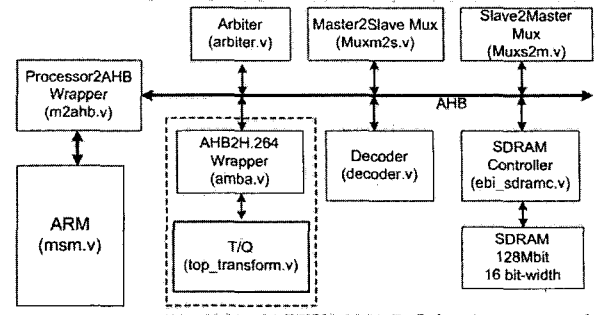


그림 7. 제안된 플랫폼의 블록도
Fig. 7. Block diagram of the proposed platform.

표 2. T/Q 설정 레지스터
Table 2. T/Q Configuration Register Map.

Bit	15	14~10	9~5	4~1	0
Data 1	QP		NoF	IS	FI
Data 2	RLS				
Data 3	SB	AM			

scheme^[8]. The decoder determines which IP will be selected. The two multiplexers 'Master2Slave' and 'Slave2Master' select the signals from the master and slave that are allowed by the arbiter and send them to the bus. The wrapper 'Processor2AHB' provides an interface between the AHB and ARM. The external memory to store the video data is the SDRAM (synchronous dynamic random access memory). The proposed transform and quantization IP is denoted as 'T/Q'. The wrapper 'AHB2H.264' provides an interface between the AHB and 'T/Q'.

Table 2 shows the configuration register map for the 'T/Q' IP. The register consists of three 16-bit segments: 'Data 1', 'Data 2' and 'Data 3'. The 'Data 1' segment includes 'QP', 'NoF', 'IS' and 'FI'. 'QP' is the quantization parameter ranging from 0 to 51. 'NoF' is the number of frames for which the transform and quantization will be performed consecutively. 'IS' represents the image size. In total 16 kinds of image size are supported: from the smallest 128x96 to the largest 2048x1536. 'FI' denotes the direction of the transform and quantization, i.e., forward or inverse. If the bus is granted to the 'T/Q' IP, the 'lock' signal of the AHB is set to the high state for a continuous data transfer with the SDRAM. Since a particular IP should not occupy the bus too long, a user's option, 'RLS' in the 'Data 2'

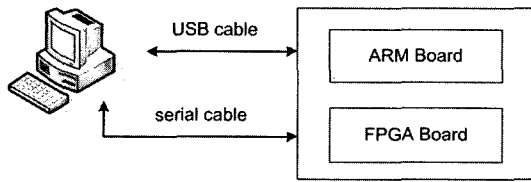


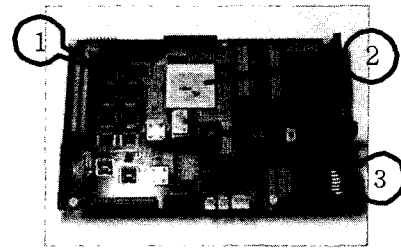
그림 8. 플랫폼 보드의 구성
Fig. 8. Organization of platform board.

segment, is added to determine when the 'lock' signal will be released. The 'RLS' specifies the number of macroblocks during which the 'lock' signal maintains the high state. The 'Data 3' segment consists of 'SB' and 'AM'. 'SB' is the start bit to initiate the 'T/Q' operation. 'AM' is to determine where the video data will be located in the SDRAM.

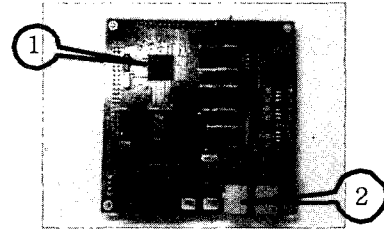
2. Verification and Implementation

As shown in Fig. 8, the platform is composed of two boards: FPGA board and ARM board in Fig. 9. In the FPGA board, ① is the PROM (programmable read only memory), ② is the Xilinx Virtex2 XC2V6000 FPGA (field programmable gate array) and ③ is the 20MHz-oscillator. In the ARM board, ① is the ARM9S3C2410 processor and ② is the USB (universal serial bus) port. Synplify v7.0 of Synplicity and Project Navigator of Xilinx are used for synthesis and placement/routing, respectively. The maximum frequency obtained for the target FPGA is 38MHz. The synthesized circuit is downloaded to the FPGA board through the serial cable. The test images are transferred from the host computer to the SDRAM (0x14000000 to 0x141FFFFFF) through the USB cable and supplied to the 'T/Q' circuit in the FPGA. The computation results from the 'T/Q' circuit are stored in the SDRAM (0x14200000 to 0x143FFFFFF), transferred to the host computer through the USB cable and then compared to the expected results. Fig. 10 shows the verification flow.

We developed the C program to provide the convenient verification environments. Fig. 11 shows the GUI (graphical user interface) of the program. ① is the button to transfer the test images from the host computer to the SDRAM. ② is the button to set



(a) FPGA board



(b) ARM board

그림 9. 플랫폼 보드 사진
Fig. 9. Photographs of platform board.

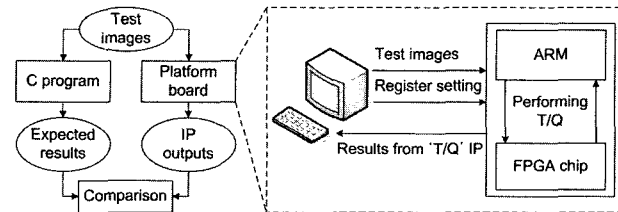


그림 10. 회로 검증 흐름도
Fig. 10. Flow diagram of circuit verification.

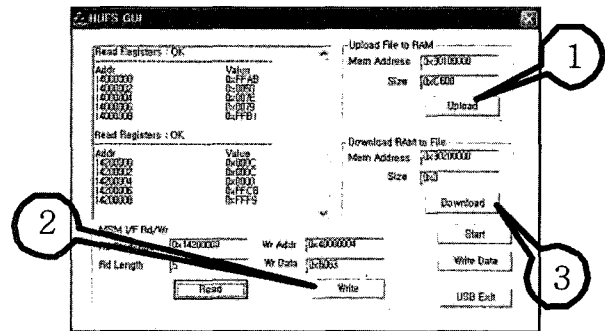


그림 11. GUI 검증 프로그램
Fig. 11. GUI of verification program.

the configuration register values to control the transform/quantization operations and the wrapper interface with the AHB. ③ is the button to transfer the computation results in the SDRAM to the host computer.

We synthesized the gate-level circuit with 0.25μm standard cells of MagnaChip Semiconductor using Design Compiler, and verified the circuit operations

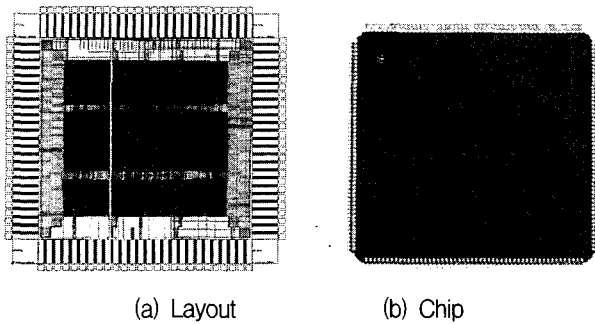


그림 12. 레이아웃과 제작된 칩
Fig. 12. Layout and fabricated chip.

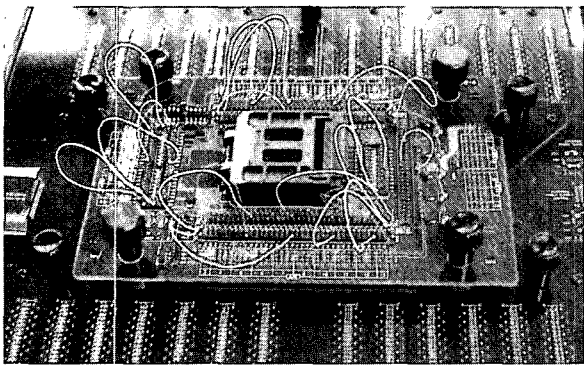


그림 13. IMS ATS2를 이용한 테스트 환경
Fig. 13. Testing environments using IMS ATS2.

for the best and worst cases by the pre-layout and post-layout simulations. We performed P&R (placement and routing), DRC (design rule check), LVS (layout versus schematic) using Astro and fabricated our MPW chip. Fig. 12 shows the final layout ($2,413\mu\text{m} \times 2,426\mu\text{m}$) and the photograph of the fabricated chip. The package of our chip is LQFP (low-profile quad flat package) and has a total of 208 pins. As shown in Fig. 13, we tested the fabricated chip using a prototype tester IMS ATS2 and observed correct operations compared with the expected outputs.

IV. Conclusions

We proposed an AMBA-based IP that can perform the H.264 transform and quantization operations for video compression and decompression. The core of the IP is the transform and quantization circuit optimized for area and performance. The AHB wrapper provides an interface conforming to the

AMBA Specification. In order to offer the flexibility, our IP allows the user to specify the options such as how long the bus may be occupied by the IP, where the video data are stored in the external memory, and so on. We described the whole platform including our IP at RTL and verified the AMBA-compliant operation through the RTL simulations. We also built the platform board with Xilinx FPGA and ARM9 processor and demonstrated the correct operations. An MPW chip with our IP was fabricated using $0.25\mu\text{m}$ standard cells to prove its AMBA-compliant operations on silicon. The proposed IP will be used as an essential block when we implement the H.264 CODEC using the platform-based design methodology. It is important to make sure that there is no problem in the bandwidth when the AMBA bus is shared by many IP's. As a future work, we are planning to address this problem when we implement the H.264 CODEC.

References

- [1] ARM Ltd, "AMBA specification rev 2.0," Document Number ARM IHI 0011A, 1999.
- [2] Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC), Mar. 2003.
- [3] S. Lee and K. Cho, "An efficient architecture of area-optimized transform circuit for H.264/AVC," *Proc. of ITC-CSCC*, pp. 749-750, 2005.
- [4] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Trans. on Circuit Syst. Video Technol.*, Vol. 13, pp. 598-603, Jul. 2003.
- [5] L. Liu, L. Qiu, M. Rong and L. Jiang, "A 2-D forward/inverse integer transform processor of H.264 based on highly-parallel architecture," *Proc. of IEEE International Workshop on System-on-Chip for Real-Time Applications*, pp. 158-161, 2004.
- [6] T. Wang, Y. Huang, H. Fang and L. Chen, "Parallel 4x4 2D transform and inverse transform architecture for MPEG-4 AVC/H.264," *Proc. of IEEE ISCAS*, pp. 800-803, 2003.
- [7] I. Amer, W. Badawy and G. Jullien, "Hardware

prototyping for the H.264 4x4 transformation,"
*Proc. of IEEE International Conference on
ASSP*, pp. 17-21, 2004.

- [8] M. Conti, M. Caldari, G. B. Vece, S. Orcioni and
C. Turchetti, "Performance analysis of different
arbitration algorithms of the AMBA AHB bus,"
Proc. of 41st DAC, pp. 618-621, 2004.

저 자 소 개

이 선 영(학생회원)

제 37권 SD편 1호 참조.
현재 한국외국어대학교 전자정보공학부 박사과정.

조 경 순(정회원)

제 37권 SD편 1호 참조.
현재 한국외국어대학교 전자정보공학부 교수.