

AMBA 기반 플랫폼을 이용한 디블록킹 필터 회로의 설계 및 검증

박강필, 이선영, 조경순
한국의국어대학교 전자정보공학부

Design and Verification of Deblocking Filter Circuit Using AMBA-Based Platform

Kangpil Park, Seonyoung Lee and Kyeongsoon Cho
Department of Electronics and Information Engineering
Hankuk University of Foreign Studies

E-mail : earman1128@hufs.ac.kr, leesy@ect.co.kr, kscho@hufs.ac.kr

Abstract

This paper presents an AMBA-based IP that can perform the deblocking filtering operations required in the H.264 video compression. The deblocking filter circuit was optimized for area and performance. The AHB wrapper was added to the circuit to interface with the AMBA-based platform. The AMBA-compliant operation of the proposed IP was verified on the platform board with Xilinx Virtex2 XC2V600 FPGA and ARM9 processor.

I. 서론

휴대폰을 비롯한 각종 디지털 정보기기들이 그 기능을 원활하게 수행하려면 마이크로프로세서, 메모리 등 다양한 부품을 필요로 한다. 이와 같은 여러 가지 부품들을 하나의 반도체 칩에 집적하여 시스템 기능을 수행할 수 있도록 구현한 것이 SoC (System on Chip) 이다. SoC 기능이 복잡하고 다양해짐에 따라 설계 기간을 단축하기 위하여 IP (Intellectual Property)의 체계적 재사용을 목적으로 제시된 설계 방법론이 PBD (Platform Based Design) 이다. H.264 디블록킹 필터를 선정하여 회로 설계를 수행하였으며, AMBA (Advanced Microcontroller Bus Architecture)[1] 기반 플랫폼을

이용하여 검증하였다. 여기에서 사용된 플랫폼은 AMBA 버스 구조를 기반으로 설계 되었으며, arbiter, decoder, multiplexer 등의 기본 IP 들을 포함하고 있다.

II. 디블록킹 필터 회로 구조

H.264/AVC[2]에 적용된 동영상 압축 방법들 중 디블록킹 필터[3]는 인코더와 디코더 모두에 적용되는 알고리즘이다. 역 양자화와 역 변환을 거쳐 복원된 데이터는 매크로 블록 경계선에서의 화질 열화를 갖게 되는데, 이때 디블록킹 필터를 적용하면 영상의 질을 향상 시킬 수 있다. 이와 같은 디블록킹 필터 연산을 효율적으로 수행하기 위하여 파이프라인 구조를 채택하였으며 외부 메모리와의 자료교환을 최소화 하기 위하여 필터 연산순서를 최적화 하였다.[4] 수직방향과 수평방향의 디블록킹 연산을 동일한 구조의 회로를 사용함으로써 회로 설계의 복잡도를 줄이고 구현되는 회로의 크기를 줄일 수 있었다. 또한 한 개의 매크로 블록을 처리하는데 소요되는 시간이 작기 때문에 고속 처리가 요구되는 장치에 적용이 가능하다.

복원된 데이터들은 디블록킹 필터 연산을 위해 그림 1 과 같이 외부 메모리에 저장된다. 외부 메모리에 저장된 데이터들은 내부 SRAM (Static Random Access

Memory)으로 전달된다. 내부 SRAM 에 저장된 데이터들은 더블록킹 필터 연산을 위해 데이터 버퍼로 전달되고 필터 연산부에 입력되어 수직방향과 수평방향으로 연산을 수행하는데 사용된다. 수직방향의 연산 후 필터링된 결과는 수평방향의 연산을 위해 트랜스포즈 버퍼를 거쳐 내부 SRAM 에 저장된다.

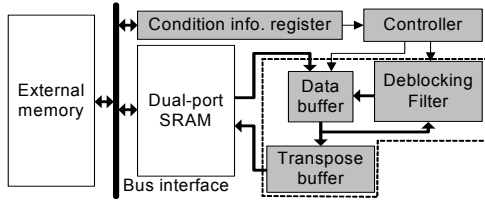


그림 1. 제안한 더블록킹 회로의 시스템 구성

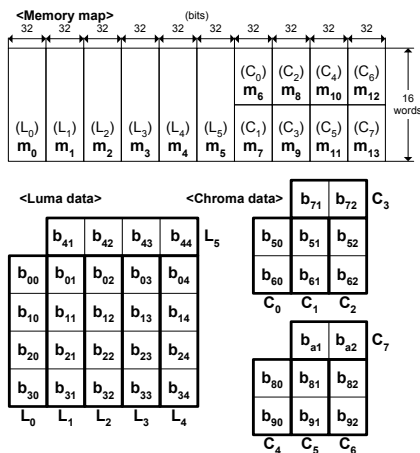


그림 2. 더블록킹 필터 연산의 메모리 맵

그림 2 는 더블록킹 필터 연산에 사용되는 내부 SRAM 의 메모리 맵과 이에 대응되는 데이터 영역을 나타낸 것이다. $L_0 \sim L_5$ 의 luminance 데이터 영역은 16 개의 32 비트 데이터들로 구성되고 $C_0 \sim C_7$ 의 chrominance 데이터 영역은 8 개의 32 비트 데이터들로 구성된다. Luminance 데이터와 chrominance 데이터 저장에 사용되는 내부 SRAM 의 크기는 160×32 비트, 총 5,120 비트이다. $L_0, C_0, C_4, L_5, C_3, C_7$ 는 (N-1)번째 더블록킹 필터 연산의 결과 중 N 번째 더블록킹 필터 연산에서 다시 연산이 이루어져야 할 부분이다. 이 영역들의 데이터를 내부 메모리에 저장함으로써 외부 메모리와 자료교환을 최소화 할 수 있다. 이는 매 매크로 블록마다 2,048 비트 (읽기, 쓰기 각각 1,024 비트)의 데이터 액세스를 줄여서 약 20%의 버스 점유율 감소효과를 가져온다.

그림 3 은 제안한 더블록킹 필터의 타이밍 다이어그램을 나타내고 있다. 더블록킹 필터 연산은

스테이지로 구분되어 있고 한 스테이지는 4 사이클이 소요된다. 각 스테이지에서는 현재 블록의 16 픽셀과 수직 혹은 수평방향에 인접한 블록의 16 픽셀들에 대해 4 번의 필터 연산이 이루어진다. 스테이지 0 에서 더블록킹 필터 연산의 왼쪽 4×4 SMB (sub-macroblock)의 데이터인 b_{00} 를 입력한다. 스테이지 1 에서 b_{01} 의 데이터들이 입력되면서 b_{00} 의 데이터들과 더블록킹 필터 연산이 이루어진다. 연산이 끝난 b_{00} 의 데이터들은 트랜스포즈 버퍼에 잠시 저장되었다가 내부 SRAM 에 저장되고, b_{01} 의 데이터들은 다음 사이클에서 b_{02} SMB 과 필터링 연산을 하게 된다.

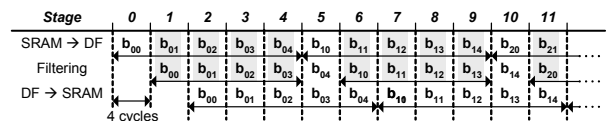


그림 3. 더블록킹 필터 연산의 타이밍 다이어그램

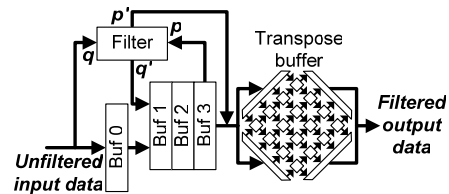


그림 4. 제안한 더블록킹 필터의 구조

파이프라인 구조를 구현하기 위해 그림 4 와 같은 더블록킹 필터 연산 블록을 사용하였다. 우선 필터링 되지 않은 데이터들을 4 픽셀 단위로 입력 받아 Buf0 ~ Buf3 의 레지스터들에 FIFO (first in first out) 방법으로 저장한다. 이전 블록의 데이터 p , 현재 블록의 데이터 q 를 입력 받아서 필터링된 p', q' 결과를 얻게 된다. 데이터 p 는 Buf3 레지스터에 저장되어 있는 값이고, 데이터 q 는 현재 입력되는 픽셀들이다.

표 1. 더블록킹 필터 회로의 크기와 성능 비교

Architecture	Gate count (K)	DF only	Total	Performance (fps)	SRAM (Kbits)	
		(cycles/MB)				
[5] (TPSRAM)	N.A.	>500	N.A.	N.A.	1 frame	
[6]	Basic (SPSRAM)	18.9	558	878	31.6	10.24
	Advanced (DPSRAM)	18.9	494	814	34.1	5.12
	Basic (TPSRAM)	18.9	462	782	35.5	10.24
	Dual array (TPSRAM)	20.7	294	614	45.2	10.24
Proposed (DPSRAM)	15.4	268	524	53.0	5.12	

(SPSRAM: single-port, DPSRAM: dual-port, TPSRAM: two-port)

표 1 은 이미 연구된 다른 접근방식과 성능 비교를 한 결과이다. 이들 실험결과는 모두 0.25 μ m 표준 셀을 사용하였고 100 MHz 주파수에서 1,280 x 720 영상에 대한 성능을 측정 한 것이다. 이전 연구 결과와 비교했을 때, 제안한 회로 크기는 약 3,000 게이트 이상 더 작고 성능은 훨씬 뛰어나다는 것을 알 수 있다.

III. 플랫폼 기반 설계 검증 환경

ARM 사의 AMBA 버스는 단일 칩 통신의 기준이 되는 버스로서 AHB (Advanced High-performance Bus), ASB (Advanced System bus), APB (Advance Peripheral bus) 세 가지의 버스 구조를 제시하고 있다. 이 중 높은 클럭 주파수를 지원하고 버스트 데이터 전송, 분할 데이터 전송이 가능하고 또한 멀티레이어 버스 구조를 가지고 있어서 고성능 회로 설계가 가능한 버스 구조가 바로 AHB 구조이다.

3.1 AMBA 기반 플랫폼의 구조

본 논문에서 제안한 플랫폼에서는 AMBA 의 AHB 버스 구조를 사용하고 있다. 그림 5 는 제안된 플랫폼의 블록도 이다. 플랫폼은 arbiter, decoder, multiplexer, master 그리고 slave 를 포함하고 있다. Arbiter 는 master 들이 버스의 사용권을 요청할 때 우선순위 알고리즘을 이용하여 우선순위가 가장 높은 master 에게 버스의 사용권을 부여한다. Decoder 는 정해진 스케줄에 따라서 어떤 IP 가 다음 수행을 하게 될지 결정을 해주는 역할을 한다. Multiplexer 는 arbiter 와 decoder 에 의해 결정된 master 와 slave 의 신호들을 버스에 실어주는 역할을 한다. Processor2AHB Wrapper 는 ARM 과의 인터페이스를 위해 사용되었고 SDRAM 은 데이터를 저장할 메모리이다. ARM 은 오직 master 로만, SDRAM 은 slave 로만 사용된다. 검증하기 위해 삽입된 디블록킹 필터 (DF)모듈은 AHB2H.264 wrapper 를 통하여 동작하게 된다.

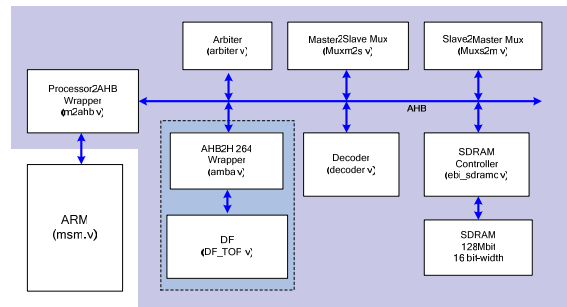


그림 5. 제안한 플랫폼의 블록도

3.2 플랫폼 보드상의 검증

플랫폼은 ARM 보드와 FPGA 및 SDRAM 이 포함된 보드로 이루어져 있다. 그림 6 은 PC 와 보드간의 데이터 통신을 보여주고 있다. 합성된 모듈은 시리얼 케이블을 통해서 FPGA 보드에 다운로드 되며 테스트 될 이미지는 USB 케이블을 이용하여 SDRAM 에 다운로드 된다.

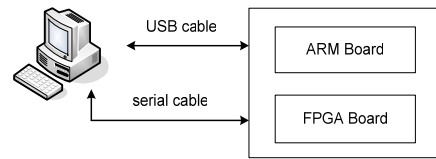


그림 6. PC 와 보드간 데이터 통신

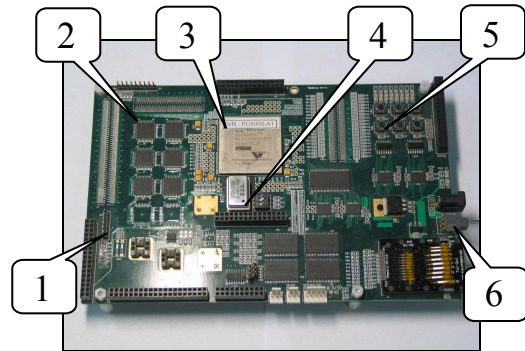


그림 7. FPGA 보드

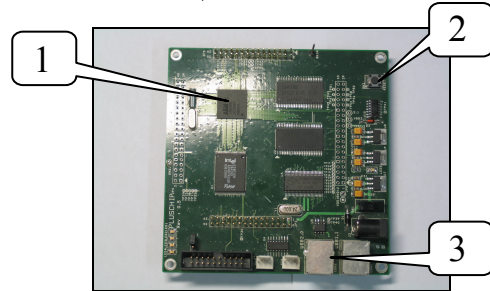


그림 8. ARM 보드

플랫폼 보드는 그림 7의 FPGA 보드와 그림 8의 ARM 보드로 구성 된다. 이 두 보드는 서로 60개의 핀을 통해 연결되어 통신하게 된다. 먼저 FPGA 보드의 구성을 살펴보면 그림 7의 ①은 합성된 회로가 전송될 시리얼 포트이며 ②는 FPGA에 다운로드 될 파일들이 저장되는 PROM (Programmable Read Only Memory)으로 총 6개가 존재한다. ③은 FPGA 칩으로 Xilinx Virtex2 XC2V6000을 사용하였다. ④는 오실레이터로 20MHz를 사용하였고 ⑤는 리셋 버튼 그리고 ⑥은 전원이다. 그림 8의 ①은 ARM9 프로세서 (ARM9S3C2419)이다. ②는 리셋 버튼이고 ③은 이미지 데이터를 받는 USB 포트이다.

회로 검증에 위해 Synplicity사의 Synplify 합성기를 사용하여 합성을 하였고, Xilinx사의 Project Navigator를 사용하여 배치 배선을 하였다. 이 회로는 시리얼 케이블을 통해서 FPGA 보드로 다운로드 되며 테스트 이미지는 USB 케이블을 이용하여 SDRAM의 0x14000000 ~ 0x14FFFFFF 주소영역에 다운로드 된다. 더블록킹 필터를 수행한 결과 데이터 또한 SDRAM의 0x14000000 ~ 0x14FFFFFF 주소영역에 저장된다. 이 데이터는 그림 10의 GUI 환경을 이용하여 PC로 업로드 된다.

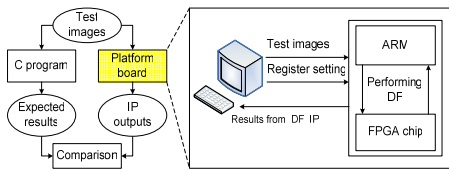


그림 9. 검증의 전체 흐름도

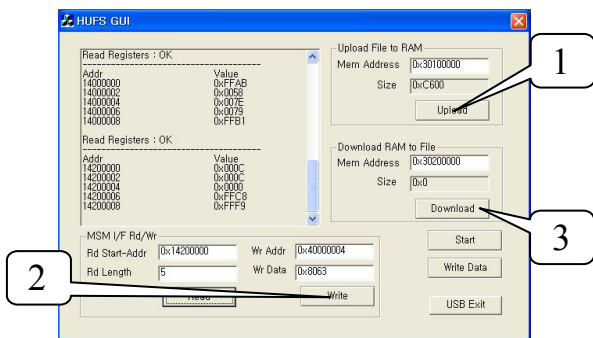


그림 10. 테스트를 위한 GUI 환경

그림 9는 검증의 전체 흐름도를 보여준다. C 프로그램상에서 추출한 데이터와 이미지 데이터를 보드상에 다운로드 시킨 다음, 연산결과 데이터를 서로 비교하여 동일 여부를 확인 함으로써 검증이

이루어진다. 그림 10은 PC와 보드간에 이미지를 업로드 및 다운로드 하는 GUI (Graphic User Interface) 환경을 보여준다. ①은 이미지 데이터를 FPGA의 SDRAM 영역에 업로드 할 때 사용하는 버튼이고 ②는 더블록킹 필터회로가 동작하도록 레지스터 값을 지정해주는 버튼이다. 그리고 ③은 SDRAM에 저장되어 있는 데이터를 PC에 다운로드 할 때 사용하는 버튼이다.

IV. 결론

본 논문에서는 효율적인 SoC 설계 방법론으로 알려져 있는 PBD 설계방법을 적용하여 H.264 더블록킹 필터 IP를 설계하고 검증하는 과정을 기술하였다. 플랫폼은 AMBA AHB 버스구조를 기반으로 설계하였고 arbiter, decoder, multiplexer 등의 기본 IP들을 포함하고 있다. Verilog HDL (Hardware Description Language)을 사용하여 RTL (Register Transfer Level) 회로를 설계하였고 시뮬레이션을 통하여 검증하였다. 또한 Xilinx FPGA와 ARM9 프로세서를 탑재한 보드를 제작하여 더블록킹 필터 IP가 AMBA 기반 플랫폼 상에서 동작하는 것을 확인하였다.

참고문헌

- [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] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. on Circuit Syst. Video Technol.*, Vol. 13, no. 7, pp. 614-619, Jul. 2003.
- [4] S. Lee and K. Cho, "An efficient architecture of high-performance deblocking filter for H.264/AVC," *Proc. of ITC-CSCC*, vol.2, pp.99-100, July. 2005.
- [5] M. Sima, Y. Zhou, and W. Zhang, "An efficient architecture for adaptive deblocking filter of H.264/AVC video coding," *IEEE Trans. on Consumer Electronics*, Vol. 50, no. 1, pp. 292-296, Feb. 2004.
- [6] Y. Huang, T. Chen, B. Hsieh, T. Wang, T. Chang, and L. Chen, "Architecture design for deblocking filter in H.264/JVT/AVC," *Proc. of Int. Conf. on Multimedia and Expo*, Vol. 1, pp. 693-696, Jul. 2003.