

H.264 율제어 알고리즘의 하드웨어 설계

서기범^{1*}

¹우송대학교 철도전기정보통신학부

A hardware design of Rate control algorithm for H.264

Ki-Bum Suh^{1*}

¹Division of Railway electrical & information Communication Engineering,
Woosong University

요약 본 논문에서는 H.264 방식의 full HD실시간영상압축을 위한 율제어 모듈의 새로운 구조를 제안한다. 제안된 구조에서는 각 매크로블록 라인(full HD의 경우 120 매크로 블록, CIF의 경우 22 매크로 블록)을 따라 율제어 알고리즘을 사용함으로써 QP가 변경된다. JM의 H.264 율제어 알고리즘에는 복잡한 산술연산과 부동 소숫점 연산을 가지고 있기 때문에, 정수형 산술 CPU를 통한 율제어 알고리즘의 구현은 불가능하다. 따라서 우리는 부동 소숫점 연산 유닛을 채용하고, 이 부동소숫점 연산 유닛을 이용하여, 율제어 알고리즘을 구현하였으며, 이 하드웨어를 통하여 실시간에 동작할 수 있음을 확인하였다.

Abstract In this paper, we propose a novel hardware architecture for Rate control module for real time full HD video compression. In the proposed architecture, QP is updated by using the rate control algorithm to every the macroblock line(120MB for Full HD, 20MB for CIF image). Since there are many complex arithmetic and floating point arithmetic in rate control algorithm of JM for H.264, it is impossible to process the rate control algorithm using the integer arithmetic CPU core. So we adopted floating point arithmetic unit in our architecture, and implemented the rate control algorithm using the floating unit. With this implemented hardware, the implemented hardware is verified to be operated in real time.

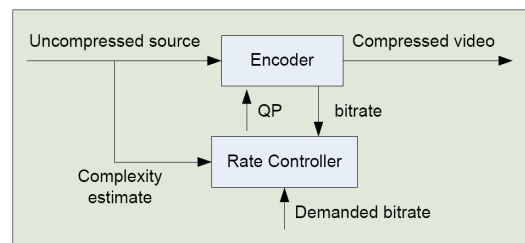
Key Words : H.264, Rate control, Floating point unit

1. 서론

H.264의 표준은 실시간 비디오 커뮤니케이션을 목표 디자인 되었다. 현대에는 네트워크 속도가 발전하여 고품질의 IPTV가 등장하고 있다. H.264는 옵션으로 Rate Control를 지원하고 있다.[1] IPTV에서 Rate Control를 사용하면 매우 유리하다. 네트워크 상황에 따라 화질이 조절되면서 네트워크 상황에 맞게 스트림이 발생되며 이용자에게 더 좋은 화질과 쾌적한 영상시청이 가능해진다. H.264의 율제어방식은 실시간으로 출력되는 스트림량을 파악하고 양자화(QP)값을 변화하면서 출력되는 스트림량을 균형 있게 조절하는 역할을 한다. 만약 복잡한 영상일 경우 비트스트림의 양이 증가한다. 그 경우 H.264의 Rate control은 양자화 값을 올려 출력되는 스트림을 균일하게 생성하게 할 것이다. 이와는 반대로 영상이 단순했다면 출력 스트림이 작게 나온다. H.264의 Rate

control은 QP값을 내려 그 부분에선 고품질의 영상을 제공하면서 출력스트림을 균일하도록 유지해야 한다.

RC(Rate control)모듈은 그림 1과 같이 효율적인 비트 할당을 통한 목표 비트 수에 근접한 비트를 발생하고, 또한 주어진 목표 비트 안에서 전체화면의 최적의 화질을 유지하기 위한 모듈이다. RC모듈은 할당된 목표 비트에 현재의 이전 영상의 복잡도를 파악하여 QP값을 예측하여 출력되는 스트림의 비트수를 조절하는 방식이다.



[그림 1] Rate control 구조

*교신저자 : 서기범(kbsuh@wsu.ac.kr)

이를 위한 알고리즘은 JVT 포럼을 통하여 제안 되어 왔고[2], 많은 연구자를 통하여 알고리즘이 연구되어 왔다.[3-5] 이러한 알고리즘 중 [2]가 참조 소프트웨어인 JM13.2[6]에는 포함되어 왔다. 지금까지 많은 알고리즘이 제안되었지만 현재까지의 최신 software인 JM15.6도 [2]의 방법을 표준화 소프트웨어에서는 채택을 하고 있다. 하지만 JM의 software codec을 하드웨어로 구현하는 문제는 H.264의 알고리즘은 정수값 연산을 통하여 구현될 수 있는 것에 비하여, 윌제어 알고리즘은 부동소수점의 처리 및 제곱근(root), 나눗셈 연산 등의 하드웨어로 구현하기 어려운 문제가 있다. 따라서 본 논문에서는 JM을 기반을 한 윌제어 알고리즘을 full Hardwired 설계를 위한 구조를 제안하고 구현하였다.

2. JM의 윌제어 알고리즘

H.264의 JM에 porting되어 있는 알고리즘은 다음과 같은 5가지 step을 통하여 구현된다.

Step 1. 유동 트래픽 모델을 이용하여 현재 화면의 목표 비트 할당

Step 2. 남아있는 비트를 부호화할 전체 부호화 단위에 할당

Step 3. 현재 프레임의 basic unit의 MAD추정

이것은 이전 프레임의 같은 위치의 기본유닛(collocated basic unit)의 MAD를 linear model을 사용하여 추정한다.

$$MAD_{cb} = a_1 \times MAD_{pb} + a_2 \quad (1)$$

Step 4. 하나의 basic unit의 QP 결정
quadratic rate model을 사용하며, 이 모델은 식(2)와 같다.

$$T(n_{i,j}) = X_1 \frac{MAD}{Q_{step}(N_{i,j})} + X_2 \frac{MAD}{Q_{step}^2(N_{i,j})} \quad (2)$$

Step 5. Step 4를 통해 얻어진 양자화 변수 값을 이용하여 비트-왜곡 최적화(RDO)를 수행한 후 부호화 하고, 모든 모델 변수 값을 갱신 한다.

위의 기술된 step4 와 step 5의 수행은 부동소수점을 이용한 나눗셈연산, 곱셈연산, 제곱근연산을 포함하는 복잡한 연산을 사용하기 때문에 실제 구현은 Software적으로 사용하는 ffmpeg을 사용할 경우, 상당히 지연된 부호화시간을 보인다. 특히 하드웨어로만 이루어져 있는 인코더의 경우에, 부동소수점의 연산을 갖는 CPU core를 쓰는 경우, 매크로블록당 처리data를 전달하기 위한 BUS의 점유와 모듈간의 인터페이스의 처리문제가 심각하다. 따

라서 현재 DVR 시스템에서는 윌제어를 하지 못하고, frame을 skip하는 방법을 사용하며, event가 발생 시 다른 wavelet이나 MJPEG으로 코딩하여, 이것을 보상하는 방법을 채택하고 있다.[6,7]

또한 software적인 방법을 사용하는 경우 영상 크기가 작은 경우 비스트림이 적어 예측이 용이 하지만 Full HD 일 경우 한 화면의 크기가 CIF의 20장에 달하는 크기이다. 그러므로 한 화면 당 새로운 QP를 연산하는 횟수 역시 증가해야 더 정확한 Rate control이 될 것이다. 새로운 QP를 예측하는 횟수가 증가 할수록 부동소수점 연산 역시 대폭 증가 할 것이다. 실시간 Full HD영상에서의 프로세서를 이용한 software적인 연산으로는 Rate control을 수행 할 수 없다. 따라서 본 논문에서는 Rate control의 효율적인 연산과 실시간 Full HD영상을 위한 하드웨어적인 Rate control의 구조를 제안한다. 제안된 하드웨어 구조는 현재 표준 reference C 코드에서 제공되는 JM 알고리즘과 호환 가능한 모델을 사용함으로써, 어떠한 조건에서도 최적의 성능을 갖도록 한다.

3. Rate control 하드웨어 설계

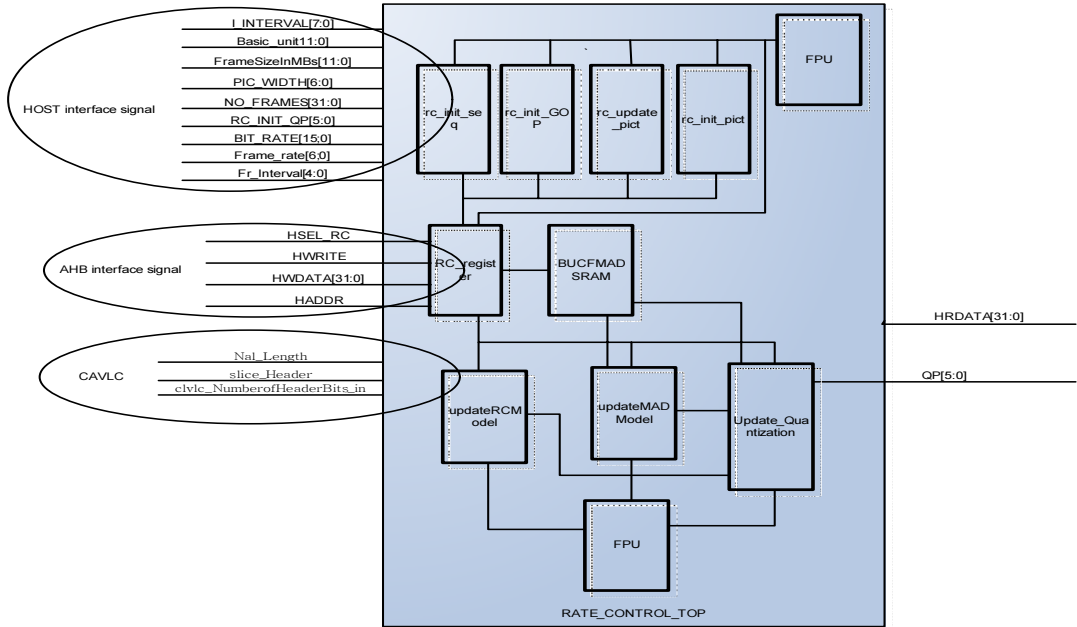
구현 하고자 하는 윌제어 하드웨어의 특징은 다음과 같다.

- 목표 비트율 : 256kbps ~ 14Mbps
- 영상 포맷 : CIF, VGA, D1, HD, Full HD
- Buffer Size : bit rate * 2.56
- 지연 시간 : HDR 규격
- 화질: 위 조건들을 만족시키는 동시에 가능한 최적의 화질 유지함.

Rate control은 영상 인코딩 시 크게 GOP단위 Picture 단위, Basic Unit(여러 개의 매크로블록의 모임으로 QP값이 바뀔 수 있는 기본단위)단위로 비트율을 제어 할 수 있으며, 각 구성 모듈 및 주변 모듈과의 관계를 나타낸 윌제어시스템 블록도는 그림 2와 같다.

Rate control의 입력 신호부분은 크게 3부분으로 나누어져있다.

- HOST 인터페이스
JM13.2 소프트웨어의 환경파일인 encoder.cfg 파일과 유사한 호스트데이터의 입력부
- AMBA AHB 인터페이스
매크로블록 단위의 데이터를 입력으로 받으며 MAD 값을 계산하기 위한 오차값을 입력.



[그림 2] Rate control 구조

● CAVLC 인터페이스

Rate control의 예측을 위한 픽처의 생성비트수와 slice 헤더의 비트수와 매크로블록 헤더의 비트수를 CAVLC에서 입력으로 받는다.

출력으로는 AHB버스를 통해 RC모듈의 상태를 출력하고 변경된 QP를 직접 출력한다.

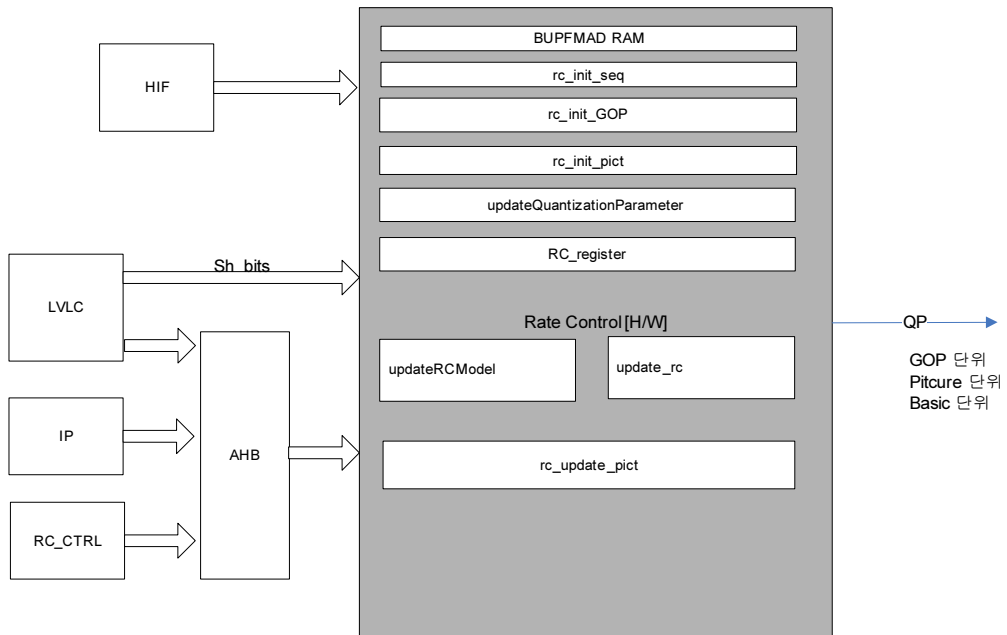
각 내부에는 두 개의 floating point unit을 이용하여, 내부적인 부동 소숫점 연산을 처리하도록 하였다. 부동 소숫점은 IEEE754(32bit)규격을 사용하였다.[8]

3.1 율제어 하드웨어의 동작 과정

율제어 하드웨어의 동작과정은 그림3의 시스템블록도에서 JM알고리즘과 compatible한 다음과 같은 8 가지 과정을 통하여 동작한다. 이해를 돕기위하여 각 동작에서 함수는 JM의 함수이름과 같은 이름을 사용하였다.

- step 1. 율제어동작에 필요한 HIF 파라미터입력, 초기화 단계[초기 1회 동작]
- step 2. GOP 단위 초기화, 픽처 단위 초기화, 율제어결과인 QP를 전달한다.[GOP 단위,픽처 단위 동작]
- step 3. CAVLC, CABAC 동작 결과 Slice 헤더의 인코딩 결과(bit 수)를 입력 받는다. [Picture 단위 동작]

- step 4. 각 basic unit 인코딩 전에 quadratic R-D 모델 파라미터 갱신하고 [P Picture일 경우 동작], 계산된 QP를 전달한다.[basic unit단위 동작]
- step 5. 각 MB 인코딩 과정의 IP(intra prediction 및 DCTQ)동작 후 MAD_int를 생산한다.[P Picture의 MB 단위 동작]
- step 6. 각 MB 인코딩 과정의 CAVLC 동작 후Number of Header Bits 및 Number of Texture Bits를 입력 받는다.[MB 단위]
- step 7. step5의 MAD_int를 받아 MAD of MB를 갱신하고, IP(intra prediction)동작 경과인 MB_CBP, MB_TYPE를 입력으로 PREV_CBP, DQP, QP값을 갱신한다(외부에서 동작함).[P Picture의 basic unit 단위 동작]
- step 8. basic unit 동작이 완료된 후 생성된 slice 총 비트 스트림을 바이트 정렬하고 그 결과로 rate control 파라미터를 갱신한다.[Picture 단위 동작](단 UpdateRCmodel은 P Picture의 경우 수행)



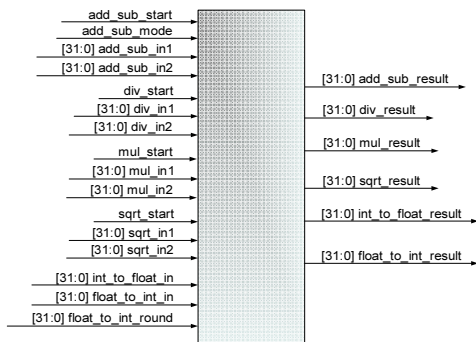
[그림 3] Rate control의 시스템 블록도

3.2 하드웨어 Rate control의 부동소수점 연산

Rate control의 거의 모든 연산은 부동 소수점 연산으로 이루어져있다. 하드웨어에서 부동소수점 연산을 수행하기 위해 제한한 구조에서는 2개의 FPU를 사용하여 최적화된 연산 속도를 구현 하였다. 1개의 FPU는 quadratic R-D 모델 업데이트와 QP를 구하는 모듈에서 사용하고 다른 하나의 FPU는 그 외 모든 모듈이 공유 하여 사용하고 있다.

150MHz에서 동작하며, 약 4만 gate 정도의 크기이다.

- 32 bit float 덧셈
- 32 bit float 뺄셈
- 32 bit float 나눗셈
- 32 bit float square root 연산
- 32 bit float 의 integer변환 연산
- 32 bit integer 의 float변환 연산

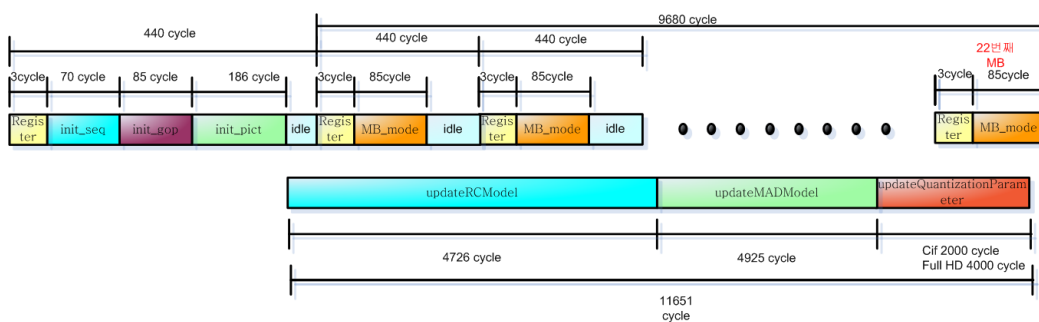


[그림 4] FPU의 블록도

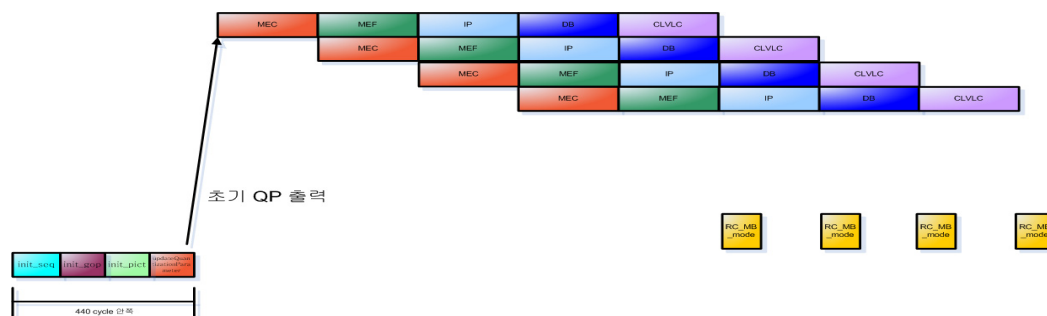
사용 FPU는 IEEE754형식의 부동소수점 연산을 처리 가능하다. 그림 4의 FPU모듈은 다음 연산을 담당한다. 제작된 FPU의 경우 TSMC 0.18um CMOS공정에서

3.3 하드웨어 Rate control 동작 cycle

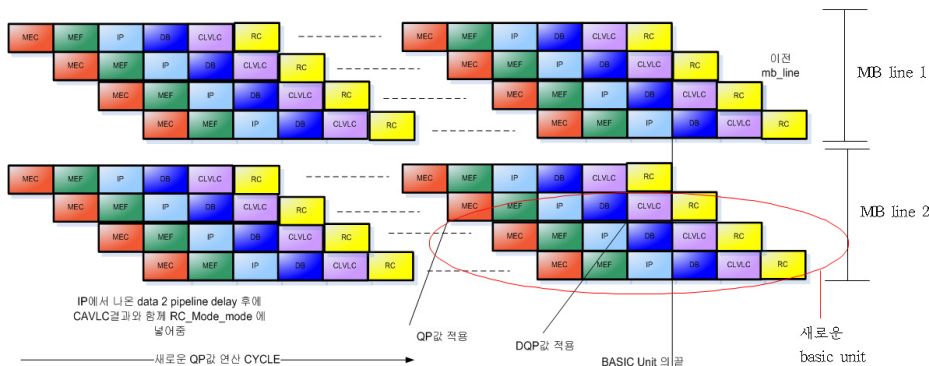
그림 5는 CIF영상에서의 rate control의 동작 cycle을 보여준다. 처음 과정을 보면 init_seq와 init_gop, init_pict 모듈이 동작 완료한 후 초기 QP를 연산 후 인코더가 실행 되게 된다. 그 후 rate control에서는 MB 단위 연산을 계속 수행하게 되고 basic unit이 끝나게 되면 update RC model과 update MAD model, update QP모듈을 수행하여 다음 basic unit의 QP를 출력하게 된다. 출력 되는 QP를 연산하기 위해서는 이전 basic unit의 모든 MB가 종료 한 후 연산을 수행 할 수 있다. 하드웨어 연산 cycle은 CIF영상에서 11651cycle, Full HD에서 13651 cycle이 소비된다. 그런데 full HD의 영상의 경우에 한 매크로블록이 수행되는 시간은 440 cycle이므로 바로 basic unit로 QP가 전달되지 못하게 된다. 따라서 최종 출력 QP는 1개의 basic unit이 딜레이 되어 사용하게 된다.



[그림 5] Rate control 동작 cycle



[그림 6] AHB interface 입력(최초 I frame)



[그림 7] QP와 DQP의 전달

3.4 전체 인코더에서의 Rate control 동작

그림 6은 AHB interface를 이용하여 입력 하는 부분이다. 최초 인코더 시작시 init_seq의 동작 신호를 입력 하게 되면 자동으로 init_seq동작이 끝난 후 init GOP, init pict, update QP모듈이 동작하게 되고 RC_MB_mode는 CAVLC 모듈이 완료된 후에 AHB interface를 이용하여 모듈을 동작 시킨다. 그 외의 모듈들은 Basic unit의 동작 상태를 확인 하고 있으므로 자동으로 동작이 이루어진다.

그림 7은 전체 인코더에서의 QP와 DQP의 전달과정을 보여준다. 그림의 위는 이전 라인을 뜻하고 아래쪽은 다음 라인의 인코딩 과정을 보여준다. 위 라인에서 생성된 데이터를 이용하여 아래 라인의 인코딩 동작 중 새로운 QP를 연산하게 되고 그 QP는 3번째 라인에 적용 되게 된다. 2번째 라인의 데이터는 3번째 라인에서 연산 후 4번째 라인에 QP를 적용하게 된다. QP값은 MEC모듈에 처음 전달하게 되면 DQP는 CAVLC 모듈에 전달하게 된다. 그리고 DB모듈의 QP는 이전의 파이프라인의 IP동작

결과인 CBP를 MB type의 상태에 따라 매크로블록의 skip 가능성이 있다면 새로운 QP를 사용하지 않고 이전의 QP 값을 사용하게 된다. 그 이유는 디코더 입장에서 매크로블록의 skip이 발생 했다면 DQP(differential QP)값을 알 수 없기 때문에 QP의 변화를 파악 할 수 없기 때문이다. 그래서 DB모듈의 QP값은 IP모듈의 QP값과 차이를 보일 수 있다.

4. 실험 결과

제안된 하드웨어의 구조에서와 JM알고리즘과의 차이점은 실시간 처리를 위하여 1 basic unit의 QP값의 지연을 하드웨어에서 시키고 있다는 점이다. 따라서 하드웨어의 QP의 1 unit delay에 의한 영향이 얼마나 나오는지 확인하였다.

P값의 1개 basic unit딜레이로 인한 PSNR의 변화율은 표 1과 같다.

[표 1] One basic Unit delay로 인한 PSNR 변화율

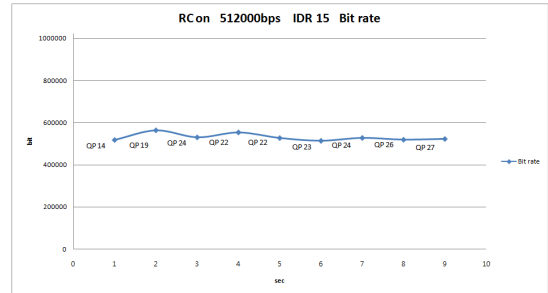
| 조건 | ReferenceC (JM 13.2) | One basic unit Delay 후 |
|--|----------------------------|----------------------------|
| foreman CIF 15IDR 256Kbps 150 frame | PSNR Y = 30.29(db) | PSNR Y = 30.25(db) |
| | PSNR U/V = 39.66/41.42(db) | PSNR U/V = 39.65/41.39(db) |
| | Total Bit = 1,285,644 | Total Bit = 1,285,808 |

PSNR Y는 30.29에서 30.25로 0.04dB 하락 하였다. 이 저하현상은 한 개의 basic unit delay 를 시킨 영향이지만, FPU 를 갖는 CPU를 사용하였을 경우에도 실시간으로 처리하기 위해서는 생길 수밖에 없는 현상이다.

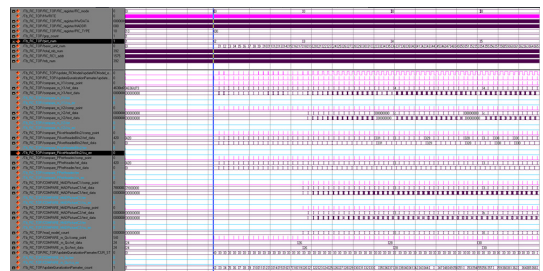
출력되는 비트스트림의 양을 계산해보면 초당 257,161.6 비트로 설정한 256k 비트와 유사하게 출력 되고 있으므로 CBR(constant bit rate)의 경우 정상적으로 비트율제어가 수행되고 있음을 보인다.

그림 8은 512kbps의 rate control의 QP의 변화량과 초당 비트 출력량을 나타낸다. QP가 변화 하면서 출력 되는 비트 양을 균형적으로 유지 하는 것을 볼 수 있다. 제안된 basic Unit의 delay를 갖는 하드웨어 구조는 verilog coding을 통하여 구현하였다. TSMC 0.18um CMOS 공정을 사용하여 합성 하였을 때, 24만 7550 gate의 크기를 가지며, 150MHz에서 동작함을 확인하였다. 그림 9의 시물

레이션 결과 파형은 각각의 모듈 출력의 REF_C와 비교 하였다. 각신호의 경우 error가 없음을 보이고 있다.



[그림 8] 512 kbps 비트레이트에서 율제어 결과



[그림 9] 시물레이션 결과 파형

5. 결론

본 논문에서는 Full HD급 영상을 실시간으로 처리 할 수 있는 하드웨어 기반의 율제어기를 설계 하였다. 일반적인 하드웨어 인코더가 율제어 알고리즘이 없이 부호화 되는 것에 비하여, 설계된 하드웨어는 JM 13.2의 율제어 알고리즘을 그대로 적용하였으며, 한 basic Unit의 delay 된 QP값을 사용하는 방법을 적용하였고, JM 13.2와 거의 같은 성능을 나타낸다. 부동 소숫점을 처리하기 위하여 FPU와 기본 모듈을 설계하였으며, TSMC 0.18um 공정에서 150Mhz에서 동작함을 확인하였다.

참고문헌

- [1] ISO/IEC 14496-10:2005 International Standard.
- [2] JVT-G012(Document from JVT(<http://wftp3.itu.int/av-arch/jvt-site/>))
- [3] Chang-hyun Lee, Seong-joo Lee, Yun-je Oh, Jaeseok Kim, "Cost-Effective Frame-Layer H.264 Rate Control

- for Low Bit Rate Video," icme, pp.697-700, 2006
IEEE International Conference on Multimedia and Expo, 2006
- [4] 손남래, 신윤정, 이귀상, "장면전환에 효율적인 H.264 /AVC 비트율제어 기법", 대한전자공회 논문지,제 44 권 SP편 제1호,pp 26~39. 2007.1
- [5] Loren Merritt and Rahul Vanam," Improved rate control and Motion Estimation for H.264 Encoder", ICIP 2007, pp V-309~312, 2007.6
- [6] H.264/AVC encoder reference software 13.2 (available <http://iphome.hhi.de/suehring/tml/>)
- [7] 이일주,임성준, 채현석,"H.264 코덱을 사용한 고성능 DVR 시스템 개발에 관한 연구", 한국산학기술학회 논문지, Vol 10. No.1, pp 110~116,2009.1
- [8] 정준모, "ARM-Excalibur를 이용한 H.264/AVC 디코더의 HW/SW 병행설계", 한국 산학기술학회 논문지, Vol.10, No.7, pp1480~1483,2009.10
- [9] IEEE Std 754-1985, IEEE Standard for Binary Floating-Point Arithmetic. 1985

서 기 범(Ki-Bum Suh)

[정회원]



- 1991년 2월 : 한양대학교 일반대학원 전자공학과 (전자공학석사)
- 2000년 8월 : 한양대학교 일반대학원 전자공학과 (전자공학박사)
- 2000년 3월 ~ 2002년 2월 : 한국전자통신연구원 선임연구원
- 2002년 3월 ~ 현재 : 우송대학교 철도전기정보통신학부 조교수

<관심분야>

영상압축, 정보통신, FPGA, Chip Design