
실시간 단일 메모리 동시 입출력을 이용한 효율적인 영상 데이터 처리

이건중* · 한금희* · 류광기*

*한밭대학교 정보통신전문대학원

Efficient Image Data Processing using a Real Time Concurrent Single Memory Input/Output Access

Gunjoong Lee* · Geumhee Han* · Kwangki Ryoo*

*Graduate School of Information and Communication, Hanbat National University

E-mail : lgikmj@gmail.com, rnagml@nate.com, kkryoo@hanbat.ac.kr

요 약

실시간 영상신호압축에서 일정 분량의 신호를 저장한 후 다른 순서로 읽어내는 과정은 간단하지만 JPEG, MPEG1/2/4, H.264, HEVC 등의 거의 모든 표준에서 필수적으로 사용하고 있는 중요한 과정이다. 실시간 처리가 중요하기 때문에 지금까지는 필요한 영상 블록 크기의 메모리를 두 개 이용하여 동시에 번갈아 가며 읽고 쓰는 이중 버퍼링 방법을 사용하였다. 예외적으로 2D DCT에서의 전치버퍼의 경우는 입출력 순서가 단순하기 때문에 단일 버퍼링을 이용하여 입출력 순서의 변환이 가능하다. 본 논문에서는 불규칙한 임의의 입출력 순서에서도 유한한 횟수 안에 규칙적 형태의 입출력 순서열이 반복됨을 보이고 그것을 이용하여 단일 메모리를 사용하는 효율적인 실시간 메모리 입출력 기법을 구현하였다.

ABSTRACT

A memory access method that data are read with different sequences with writing order is a simple but important procedure in many image compression standards, such as JPEG, MPEG1/2/4, H.264, and HEVC. For real time processing, double buffering is widely used using two block sized buffers, that accesses buffers concurrently with alternative way to read and write. In some cases like a transpose memory in 2D DCT with a simple and regular access order, a single buffering which requires only single block sized buffer can be used. This paper shows that even in complex access orders there is a regularity among updating orders within a finite turns, and suggested an effective implementation method using a single block sized buffer to process concurrent read/write operation with different access orders.

키워드

Doubling Buffering, Scan Order, Image Compression, Zigzag Scan, Diagonal Scan

I. 서 론

영상신호는 일반적인 신호처리와는 달리 처리해야 할 데이터의 양이 방대하며 동영상의 경우는 실시간 처리가 중요하다는 특징을 가지고 있다. 많은 양의 공간적 신호를 효율적으로 처리하

기 위해서 일정크기의 블록 단위로 처리하는데 영상압축코덱이 그 대표적인 예이다. 블록단위로 신호처리를 할 경우 JPEG, MPEG의 Zigzag Scan이나 HEVC의 Diagonal Scan처럼 처리순서를 변경해야 할 경우가 생긴다[1-3]. 특히 실시간 처리를 위해서는 입력 데이터를 처리하고자 하는 블

록 크기의 메모리에 저장하는 동시에 그 이전에 저장한 데이터를 새로운 처리순서로 출력시켜야 한다[4].

그림 1과 그림 2는 입출력 데이터의 처리 순서를 바꾸기 위해 일반적으로 쓰이는 두 가지 방법을 보여주고 있다. 그림 1은 처리하고자 하는 블록 크기의 메모리 두 개를 이용하는 방법으로 각각의 메모리가 입출력 동작을 블록단위로 교대함으로써 실시간 데이터 처리를 수행한다. 범용으로 사용할 수 있는 방법이지만 처리하고자 하는 데이터의 양보다 두 배의 메모리 공간이 필요하다. 단점이 있다.

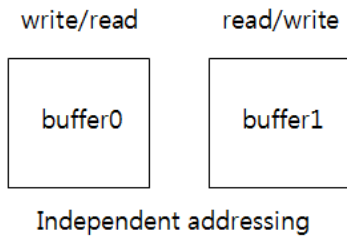


그림 1. 이중버퍼링을 사용한 입출력 순서변환

한편 그림 2는 처리하는 블록의 크기만큼만 메모리를 사용하는 방법으로 데이터를 출력하는 동시에 같은 주소에 데이터를 입력하는 방법을 사용한다[4].

입출력 순서가 단순하고 규칙성을 갖는 경우 사용할 수 있는 방법으로 입출력 순서가 복잡할 경우 메모리 접근주소가 매번 불규칙적으로 변하기 때문에 사용하기가 어렵다.

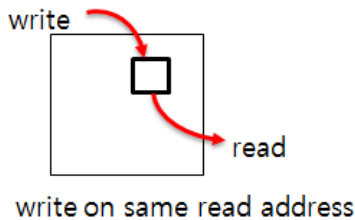


그림 2. 단일버퍼링을 사용한 입출력 순서변환

본 논문에서는 임의의 불규칙한 순서의 입출력 데이터 처리에 있어서도 유한한 횟수 안에 규칙적인 입출력의 반복 패턴이 나타남을 보였다.

또한 그러한 사실을 이용하여 실제 영상압축코덱에 쓰이는 불규칙한 형태의 실시간 입출력 변환 블록의 효율적인 구현방법을 제시하고 그 효율성을 이전 방법과 비교하여 입증하였다.

II. 본 론

A. 단일 메모리를 이용한 데이터 입출력 순서 변환의 기본 개념 및 문제점

단일 메모리를 이용한 데이터 입출력 순서변환은 동시에 이루어지는 입출력 데이터 간의 자리바꿈을 이용한다.

영상압축표준에서 자주 쓰이는 8x8 이차원 DCT는 먼저 1차원 변환을 수행한 후 그 출력의 가로와 세로를 바꾸어 다시 1차원 DCT를 수행한다. 이 때 입출력 데이터의 순서를 바꾸기 위해 전치버퍼가 필요한데 입출력 순서가 단순하여 간단히 동시 입출력을 위한 구조를 구현할 수 있다 [5].

반면 표 1은 불규칙한 출력 순서를 갖는 차세대 영상압축코덱에 쓰이는 4x4 Diagonal Scan의 경우를 나타낸다[3]. 이 경우 그림 2와 같은 단일 메모리를 이용한 동시 입출력 방법을 사용하면 표 1과 같이 입출력 순서열이 매번 불규칙하게 변하기 때문에 구현하기가 어렵다.

표 1. 4x4 Diagonal Scan에서의 입출력순서

입력	0	1	2	3	4	5	6	7
출력	0	4	1	8	5	2	12	9
입력	8	9	10	11	12	13	14	15
출력	6	3	13	10	7	14	11	15

B. 이전 연구에서의 처리방법 및 제약

앞서 언급한 문제점을 해결하기 위해 매번 갱신되는 주소열을 별도의 주소저장용 메모리에 저장하는 방법을 이용하는 방법이 제시되었다[4]. 매번 바뀌는 물리적 주소를 별도의 메모리에 저장한 후 다음 순서열은 실제 물리적 주소가 아닌 주소를 저장한 메모리의 주소를 참조하여 결정한다. 이와 같은 방법은 임의의 불규칙한 순서열에도 대응할 수 있는 장점이 있지만 처리해야 하는 블록의 크기가 커질수록 주소를 저장하는 공간의 크기도 커지게 되어 효율성이 떨어지는 단점이 있다.

C. 새롭게 갱신되는 입출력 순서열 패턴의 유한성 및 반복성

표 2는 앞서 언급한 4x4 블록의 Diagonal Scan에서 단일 메모리를 사용한 방법을 적용할 경우의 순서열이 갱신되는 것을 나타낸 표이다. 단일 메모리를 사용하는 방법을 적용할 경우 매번 블록을 처리할 때마다 접근하는 주소가 불규칙하게 바뀌는 것을 알 수 있다.

표 2. 4x4 Diagonal Scan에서의 순서열 갱신

초기값	순서열 갱신 횟수											
	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	0	0	0	0	0	0	0
1	4	5	2	1	4	5	2	1	4	5	2	1
2	1	4	5	2	1	4	5	2	1	4	5	2
3	8	6	12	7	9	3	8	6	12	7	9	3
4	5	2	1	4	5	2	1	4	5	2	1	4
5	2	1	4	5	2	1	4	5	2	1	4	5
6	12	7	9	3	8	6	12	7	9	3	8	6
7	9	3	8	6	12	7	9	3	8	6	12	7
8	6	12	7	9	3	8	6	12	7	9	3	8
9	3	8	6	12	7	9	3	8	6	12	7	9
10	13	14	11	10	13	14	11	10	13	14	11	10
11	10	13	14	11	10	13	14	11	10	13	14	11
12	7	9	3	8	6	12	7	9	3	8	6	12
13	14	11	10	13	14	11	10	13	14	11	10	13
14	11	10	13	14	11	10	13	14	11	10	13	14
15	15	15	15	15	15	15	15	15	15	15	15	15

이 때 입출력 순서열과 접근주소는 다음과 같은 규칙성을 가진다.

[규칙 1] 입력순서와 출력순서는 고정된 대응관계 유지한다.

[규칙 2] 1회의 블록처리에서 모든 주소는 한번만 그러나 빠짐없이 접근한다.

규칙 1에 의해 n번째 처리된 물리적 주소는 다음 차례에는 항상 고정된 m번째 순서에 처리가 된다. 또한 규칙 2에 의해서 임의의 물리적 주소가 가질 수 있는 주소는 전체 데이터의 수를 최대값으로 하는 유한하고 고정된 집합으로 제한된다.

표 2의 4번째(초기값 3) 처리되는 물리적 주소를 보면 3-8-6-12-7-9의 순서가 반복됨을 알 수 있다. 마찬가지로 집합에 속한 다른 숫자들도 같은 순서로 변화한다(초기값 12의 경우 12-7-9-3-8-6).

위의 규칙에 의해 데이터를 분석해 보면 실제 접근하는 물리적 주소는 세 가지 기본 순서열을 바탕으로 변화하고 나머지는 각 기본 순서열에 offset을 더한 관계를 가지고 있음을 알 수 있다 표 3은 기본 순서열과 종속되는 순서열과의 offset 관계를 나타내고 있다.

각 순서마다 처리되는 물리적 주소가 서로 다른 크기의 집합을 이루고 순서가 반복되기 때문

에 각 집합의 크기의 최소공배수를 구하면 전체 순서열이 최소 몇 번의 순환을 거치면 최초의 순서에 이르게 되는지 알 수 있다. 4x4 Diagonal Scan의 경우 각 집합의 크기는 1,4,6이고 최소공배수는 12이므로 12번의 순환을 거치면 처음의 순서열로 돌아오게 됨을 알 수 있다.

표 3. 기본순서열과 offset

초기값	블록카운터 값						offset	크기
	0	1	2	3	4	5		
1	1	4	5	2			0	4
4	4	5	2	1			1	
5	5	2	1	4			2	
4	4	5	2	1			3	
3	3	8	6	12	7	9	0	6
8	8	6	12	7	9	3	1	
6	6	12	7	9	3	8	2	
12	12	7	9	3	8	2	3	
7	7	9	3	8	2	12	4	
9	9	3	8	2	12	7	5	
10	10	13	14	11			0	4
13	13	14	11	10			1	
14	14	11	10	13			2	
11	11	10	13	14			3	

D. 효율적인 주소발생기의 제안 및 구현방법

본 논문에서는 효율적인 주소발생기의 구현을 위하여 각 순서열이 갖는 반복성과 연관성을 이용하였다. 그림 3은 구현한 주소발생기의 구조를 보여주고 있다.

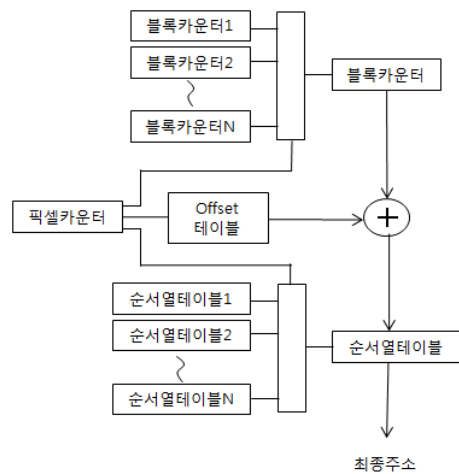


그림 3. 주소발생기의 구조

먼저 각 순서열은 같은 횟수안에 반복성을 보이는 것끼리 분류하여 동일한 블록카운터를 공유하도록 하였다. 그리고 같은 블록카운터 값을 갖는 순서열들은 고유의 offset값과 더하여 최종적인 자신의 블록카운터 값을 갖게 된다. 최종 블록카운터 값을 자신에 맞는 순서열테이블의 인덱스로 사용되어 최종 주소값을 결정하게 된다

III. 결 론

표 4는 입출력 순서열이 서로 다른 다양한 scan 블록을 본 논문에서 제안한 방법과 이중버퍼링을 이용한 이전의 방법으로 각각 구현하여 비교한 결과이다. 가장 큰 부분을 차지하는 메모리의 크기가 반으로 줄어들어 전체적으로 최소 40% 이상의 면적 감소효과가 있음을 알 수 있다

표 4. 다양한 scan 블록에서의 구현결과 비교
(TSMC 0.18um CMOS공정,166MHz)

Gate수	이전방법	제안한방법	감소율
dia4x4	4847	2634	45.7%
zz4x4	4838	2730	43.6%
zz8x8	21976	12931	41.2%

새로 제안되고 발표를 눈앞에 두고 있는 영상 표준인 HEVC에서 볼 수 있듯이 영상압축의 처리 블록은 점점 더 커지고 있는 추세이다[3]. 본 논문에서 제시된 방법은 처리 블록의 크기에 상관없이 적용될 수 있는 방법으로 블록의 크기가 커질수록 실질적 면적 감소효과는 더 커지는 효율적인 방법임을 알 수 있다

감사의 글

본 논문은 교육과학부와 한국연구재단의 지역 혁신인력양성사업 및 지식경제부 출연금으로 수행한 ETRI SW-SoC융합R&BD센터와의 공동 연구의 결과입니다.

참고문헌

[1] Pennebaker W. B. "JPEG Still Image Data Compression Standard", New York: Van Nostrand Reinhold, 1993.
 [2] Advanced Video Coding for Generic Audiovisual Services, ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4,AVC), ITU-T and ISO/IEC JTC 1, Version 13: Mar. 2010

[3] High efficiency video coding text specification draft 8, JCT-VC of ITU-T SG6 WP3 and ISO/IEC JTC1/SC29/WG11, July. 2012
 [4] 이건중, et. al., "간접 주소접근 방법을 이용한 효율적인 Zigzag Scanning 하드웨어 설계," 대한전자공학회 대전충남지부학술대회, 대전, 2011.
 [5] Jun Rim Choi, et. al., "A 400MPixel/s IDCT for HDTV by Multibit Coding and Group Symmetry," in Proc. ISSCC, San Francisco, CA, 1997.