

논문-06-11-3-06

## 내장형 JPEG 압축을 위한 단일 메모리 기반의 스캔 순서 변환기

박 현 상<sup>a)†</sup>

### Single memory based scan converter for embedded JPEG encoder

Hyun Sang Park<sup>a)†</sup>

#### 요 약

JPEG 표준에 의거한 정지 영상 압축을 위하여, 영상은 겹치지 8x8 블록 단위로 분할된다. 카메라 모듈로부터는 라스터 스캔 순서로 화소 데이터를 입력되기 때문에 JPEG 인코더와의 사이에 화소 순서 정렬을 위한 스캔 순서 변환기가 위치한다. 일반적으로 카메라와 인코더 동작을 병렬로 수행하기 위하여 8 라인에 해당하는 화소 데이터를 저장하기 위한 로컬 메모리를 2개 필요하다. 그러나 이와 같이 2개의 메모리를 사용하는 구조는 영상의 수평 해상도에 정비례하여 메모리 용량이 증가하는 문제 때문에 고해상도 영상 압축에는 적합하지 않게 된다. 본 논문에서는 하나의 메모리를 사용하여 스캔 순서 변환을 수행할 수 있는 룩 메모리 주소 생성 알고리즘을 제안한다. 이 알고리즘은 가산기와 감산기와 같은 기본 연산 소자로 구현되기 때문에, JPEG 인코더를 내장한 시스템을 효과적으로 구성할 수 있게 한다.

#### Abstract

An image is partitioned into non-overlapping 8x8 blocks for JPEG compression. A scan order converter is placed before the JPEG encoder to provide 8x8 blocks from the pixels in raster scan order. In general, its architecture requires two line memories for storing eight lines separately to allow the concurrent memory access by both the camera and JPEG processors. Although such architecture is simple to be implemented, it can be inefficient due to too excessive memory requirement as the image resolution increases. However, no deterministic addressing equation has been developed for scan conversion. In this paper, an effective memory addressing algorithm is proposed that can be devised only by adders and subtracters to implement a scan converter based on the single line memory.

Key Words : JPEG, Compression, Raster scan, Block scan, ISP, SoC

## I. 서 론

최근 카메라가 장착된 휴대전화가 대중화되면서 고부가

가치 기능을 가진 카메라 모듈에 대한 산업계의 요구가 증가하고 있다. 카메라 모듈<sup>[1]</sup>은 광학신호를 전기신호를 변환하는 이미지 센서와 이미지 센서에서 출력되는 R, G, B 데이터를 영상 처리하여 Y, Cb, Cr과 같은 색차 신호로 변환하는 ISP (Image Signal Processor)<sup>[2]</sup>를 최소 구성요소로 가진다. 이미지 센서는 CCD(Charge Coupled Device)<sup>[3]</sup>나

a) 국립공주대학교 전기전자공학부

Division of Electrical and Electronics Engineering, Kongju National University

† 교신저자 : 박현상(vandamm@kongju.ac.kr)

CIS(CMOS Image Sensor)<sup>[4]</sup>가 사용되며, 저전력 소모가 강조되는 제품 군에서는 CIS가 주도적으로 채택되고 있다. 이미지 센서에서 출력되는 RGB 영상 데이터는 임의의 화소 위치에서 한 가지 색성분 만을 가지는 Bayer 패턴<sup>[3]</sup>을 가지므로, 모든 화소 위치에서 3가지 색성분을 재현할 필요가 있다<sup>[5]</sup>. ISP는 각 화소에서의 색 성분을 재현하는 기능 외에도 자연스럽게 선명한 화질을 구현하기 위해서 AE (Auto Exposure), AWB (Auto White Balance), AF (Auto Focus) 등<sup>[6]</sup>의 기능을 부가하여 Y, Cb, Cr 색차 신호의 형태로 가공된 데이터를 출력한다. 반도체 제조공정의 미세화는 카메라 모듈의 소형화를 촉진하여 이미지 센서와 ISP를 하나의 칩<sup>[7,8]</sup>으로 제작하는 수준에 이르렀으며, 카메라 모듈의 가장 기본적인 기능인 정지 영상 촬영 기능까지도 내장하는 수준의 제품도 개발되어 있는 상황이다<sup>[9]</sup>.

JPEG (Joint Photographic Expert Group)은 정지영상을 압축하기 위해 제정된 국제표준<sup>[10]</sup>이며, 영상을 8x8 블록 단위로 분할하여 각 블록을 이산여현변환과 가변장부호화를 통해서 압축하는 기법이다.

카메라 모듈에는 ISP 내장 외에도 정지 영상 압축을 수행하는 기능이 요구되고 있다. 이 경우 전력 소모 감축을 위한 가장 보편적인 접근방식은 스캔 순서 변환을 위한 메모리를 칩 내부에 내장하고, 이의 용량을 최소로 줄이는 것이다.

스캔 순서 변환은 ISP로부터의 기입과 JPEG 압축을 위한 독출을 동시에 수행하기 위해서, 일반적으로 8 라인에 대응하는 화소 데이터를 저장할 수 있는 라인 메모리를 2 개 사용하여 구현한다. 이 구조는 설계가 용이하나 영상의 해상도가 증가함에 따라 메모리가 차지하는 면적이 지나치게 높아진다는 문제가 있다. 본 논문에서는 이 문제를 해결하기 위해서 단일메모리 기반의 스캔 순서변환기를 제안하고 이를 가능하게 하는 효과적인 메모리 제어 알고리즘을 제안한다.

## II. 본론

### 1. 스캔 순서 변환

그림 1에서 왼은 한 화소를 나타낸다. ISP는 그림 1(a)와

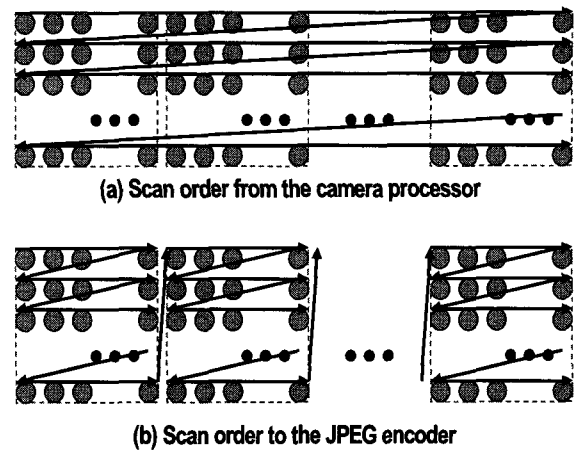


그림1. JPEG 압축을 위한 스캔 순서.

Fig. 1. Scan order conversion for JPEG compression.

같이 라스터 스캔 순서로 화소를 출력한다. JPEG은 8x8 블록 단위로 압축을 수행하기 때문에 그림 1(b)와 같이 블록 스캔 순서로 화소의 처리 순서를 변환해야 한다.

### 2. 스캔 순서 변환의 구현

그림 2와 같은 SoC구조에서는 마이크로 컨트롤러 (MCU: Micro-Controller Unit)를 비롯한 다수의 버스 마스터가 외부 메모리를 공유한다. ISP와 JPEG 압축기는 외부메모리에 독출이나 기입을 교대로 수행할 수 있기 때문에, 스캔 순서 변환이 순조롭게 이루어진다. 또한 대용량의 외부메모리 장착이 가능하기 때문에 해상도가 높은 영상을 처리하는 것이 용이해진다.

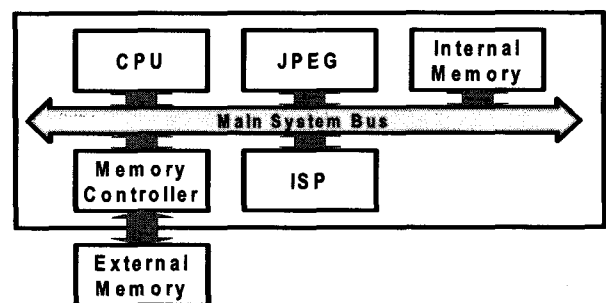


그림2. JPEG 내장 SoC 블록도.

Fig. 2. Block diagram for JPEG-embedded SoC.

그러나 외부메모리를 사용할 경우 전력소모가 심하기 때문에 저전력을 위해서는 메모리를 내장해서 스캔 순서 변환을 수행해야 한다. ISP가 YCbCr4:2:2 색차형식<sup>[3]</sup>으로 화소를 출력할 때 휘도 성분 Y와 색차성분 Cb, Cr을 분리해서 4개의 라인 메모리를 기반으로 한 스캔 순서 변환기를 그림 3에 도시한다.

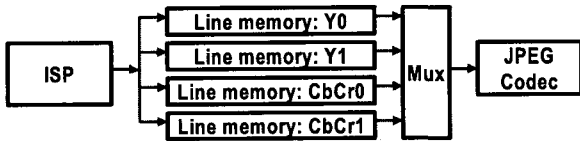


그림 3. 일반적인 스캔 순서 변환기 구조.  
Fig. 3. Conventional architecture for scan-order conversion.

표 1은 0.18m SRAM 합성기<sup>[4]</sup>를 토대로 영상의 해상도에 따른 메모리 요구량과 등가게이트 수를 정리한 것이다. JPEG 압축기의 게이트 수는 8만개 이하이므로 JPEG 압축기와 ISP를 병합할 경우 스캔 순서 변환기는 병합된 시스템의 규모와 전력량을 결정짓는다.

표 1. 해상도에 따른 메모리 요구량.

Table 1. Memory requirement according to the image resolution.

Standard	Resolution	Memory(byte)	Gate count
QCIF	176x144	5632	91764
CIF	352x288	11264	118448
VGA	640x480	20480	158340
XGA	1024x768	32768	240232

### 3. FIFO를 이용한 화소 순서 정렬

모든 메모리 코어에는 데이터를 저장하기 위한 메모리 셀 이외에 주소 해독기를 비롯한 부가회로가 내장되어 있다. 따라서 시간적으로 겹치지 않는다면 하나의 통합메모리를 사용하는 것이 효율적이다.

본 논문에서는 그림 4와 같이 FIFO를 사용해서 ISP에서 발생하는 화소 순서를 1차적으로 정렬한 뒤, 통합 메모리에 저장하는 구조를 제안한다. ISP에서는 Y, Cb, Y,

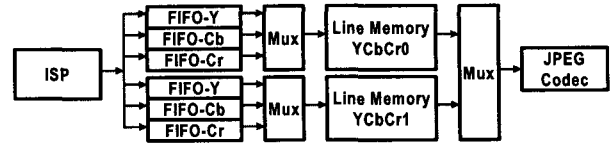


그림 4. FIFO기반의 화소순서 정렬.  
Fig. 4. FIFO-based pixel arrangement.

Cr과 같은 순서로 화소가 반복적으로 출력되고, FIFO에서는 16개의 Y화소에 이어서, 8개의 Cb화소, 8개의 Cr화소가 반복적으로 출력한다. 이렇게 정렬되어 라인메모리로 입력될 경우 라인메모리를 제어하기 위한 주소발생회로가 휘도 영상만을 다루는 경우와, 4:2:2 색차형식의 컬러영상을 다루는 경우에 기본적으로는 동일한 회로를 사용하는 것이 가능하다. 그림 4에서 Y성분을 위한 FIFO-Y와 Cb성분을 위한 FIFO-Cb의 최소 크기는 각각 8 바이트이며, Cr성분을 위한 FIFO-Cr의 최소 크기는 12 바이트이다.

표 2는 휘도 성분과 색차 신호에 대해서 별도의 메모리를 사용하지 않고, FIFO를 사용해서 휘도 성분과 색차 성분을 하나의 메모리에 저장했을 경우에 대해서 메모리 구현에 필요한 등가 게이트 수를 나타낸다.

표 2. YCbCr 통합 라인 메모리의 등가 게이트 수.

Table 2. Equivalent gate count for YCbCr-unified memory

Standard	YCbCr-separated	YCbCr-unified
QCIF	91764	59224
CIF	118448	85906
VGA	158340	128352
XGA	240232	194730

### 4. 메모리 제어 알고리즘

그림 5는 그림 4의 구조에서 독출주소를 발생하기 위한 알고리즘이다. no\_hor\_bk는 통합 라인 메모리에 저장할 8x8 블록의 수를 나타낸다. 그림에서 수평방향의 해상도는 640이라고 가정한다. 그림에서 i, v, h는 각각블록, 수직방향 화소, 수평방향 화소에 대한 인덱스를 나타낸다. anchor는 블록 내의 한 row의 각 화소가 공통적으로 가지는 주소

성분에 해당한다.

```

no_hor_bk = 640>>3;

for (i=0; i<no_hor_bk; i++){
    for (v=0; v<8; v++){
        anchor = v*no_hor_bk + i;
        for (h=0; h<8; h++){
            addr = anchor<<3 + h;
        }
    }
}
    
```

그림 5. 일반적인 메모리 독출 주소 발생알고리즘.  
 Fig. 5. General algorithm for read-address generation.

상기 알고리즘은 2 개의 라인 메모리가 있을 때 각 라인 메모리 전체를 독출 하기 위한 주소를 교대로 발생해서 연속적으로 독출이 가능해진다. 만약 임의의 주소에 기입과 독출이 동시에 가능한 라인 메모리가 있다면, 2 개의 라인 메모리를 활용해서 기입과 독출을 병행하지 않고 하나의 라인 메모리 만을 활용하는 것이 가능하다. 이는 라인 메모리에 주소가 인가되는 동안 독출버스는 해당 주소의 데이터가 비동기적으로 출력되고 라인 메모리를 구동하는 클럭에 동기되어서 해당 주소에 기입버스의 데이터를 입력하는 구조를 채택함으로써 구현할 수 있다.

현재 기입되는 8 라인에 해당하는 화소는 다음 8 라인을 저장하는 단계에서 독출된다. 본 논문에서는 이런 경우에 독출과 기입이 동시에 가능한 주소생성을 위한 알고리즘을 그림 6과 같이 제안한다. 그림에서 finished라는 변수는 하나의 영상 프레임의 모든 화소가 독출 되었는지를 나타내는 변수이다. block\_offset은 인접한 두 블록 anchor값의 차이를 나타내며 초기값은 1이다. line\_offset은 한 블록 내에서 수직방향으로 인접한 두 row anchor값의 차이를 나타내며 초기값은 영상 프레임을 구성하는 수평방향 블록의 수와 같다. 또한 현재 8 라인을 기입할 때 사용했던 line\_offset은 다음 8 라인을 독출할 때 사용할 block\_offset에 해당하게 된다.

그림 6의 알고리즘에서는 3 개의 승산기와 2 개의 나머지 연산자를 필요로 한다. 그러나 anchor를 생성할 때 사용되는 i, v는 순차적으로 증가하므로 다음과 같이 표현하는 것이 가능하다.

```

no_hor_bk = 640>>3;
no_size = no_hor_bk<<3 - 1;

block_offset = 1;
line_offset = no_hor_bk;

while (!finished){
    for (i=0; i<no_hor_bk; i++){
        for (v=0; v<8; v++){
            anchor = (block_offset*i + line_offset*v) % no_size;
            for (h=0; h<8; h++){
                addr1 = anchor<<3 + h;
            }
        }
    }
    block_offset = line_offset;
    line_offset = (line_offset*no_hor_bk) % no_size;
}
    
```

그림 6. 제안한 메모리 주소 발생 알고리즘.  
 Fig. 6. Proposed algorithm for address generation.

```

no_hor_bk = 640>>3;
no_size = no_hor_bk<<3 - 1;

block_offset = 1;
line_offset = no_hor_bk;

while (!finished){
    block_number = 0;
    for (i=0; i<no_hor_bk; i++){
        for (v=0; v<8; v++){
            line_number = 0;
            anchor = (block_number + line_number) % no_size;
            for (h=0; h<8; h++){
                addr1 = anchor<<3 + h;
                line_number += line_offset;
                line_number %= no_size;
            }
            block_number += block_offset;
            block_number %= no_size;
        }
    }
    block_offset = line_offset;
    line_offset = (line_offset*no_hor_bk) % no_size;
}
    
```

그림 7. 제안한 메모리 주소 발생 알고리즘 (2).  
 Fig. 7. Proposed algorithm for address generation (2).

나머지 연산자는 가산기와 비교기를 이용해서 구현할 수 있다. 그림 8은 주소 생성을 위한 최종적인 알고리즘이다.

위 그림에서 *line\_offset*을 계산하기 위해서 승산기가 필요하지만 영상의 해상도가 고정될 경우 *no\_hor\_bk*가 고정되기 때문에 승산기는 다수의 가산기로 대체된다. 나머지 연산자는  $64 \times no\_hor\_bk$  사이클 이내에 처리되면 되므로, 가산기와 FSM (Finite State Machine)을 이용해서 설계할 수 있다.

```
no_hor_bk = 640 >> 3;
no_size = no_hor_bk << 3 - 1;
```

```
block_offset = 1;
line_offset = no_hor_bk;
```

```
while (!finished){
    block_number = 0;
    for (i=0; i<no_hor_bk; i++){
        for (v=0; v<8; v++){
            anchor = (block_number + line_number);
            anchor -= no_size if (anchor >= no_size);
            for (h=0; h<8; h++){
                addr1 = anchor << 3 + h;
                line_number += line_number;
                line_number -= no_size if (line_number >= no_size);
            }
            block_number += block_offset;
            block_number -= no_size if (block_number >= no_size);
        }
        block_offset = line_offset;
        line_offset = (line_offset * no_hor_bk) % no_size;
    }
}
```

그림 8. 제안한 메모리 주소 발생 알고리즘 (3).  
Fig. 8. Proposed algorithm for address generation (3).

표 3은 제안한 알고리즘을 사용해서 주소를 발생할 때 각 해상도에 따라 필요한 라인 메모리의 증가 게이트 수를

표 3. 단일 통합 메모리의 증가 게이트 수.  
Table 3. Equivalent gate count for the single unified memory.

Standard	YCbCr-separated	Proposed
QCIF	91764	29612
CIF	118448	42953
VGA	158340	64176
XGA	240232	97365

나타낸다. 가장 대중적인 VGA의 경우 59%의 메모리 회로 감축이 가능해진다.

그림 9는 제안한 알고리즘에 기반을 둔 스캔 변환 시스템을 블록도를 나타낸다. ISP의 출력단에 위치한 FIFO를 통하여 같은 성분의 화소 데이터들은 8의 배수로 묶여서, 단일 라인 메모리에 저장되도록 구성되어 있다.

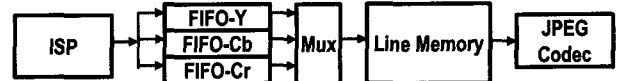


그림 9. 제안한 화소 순서 정렬 시스템.  
Fig. 9. Proposed scan conversion system.

### III. 결 론

표 4는 VGA(640x480) 해상도에 대해서 제안한 주소 발생 알고리즘과 FIFO를 이용한 데이터 정렬 기능을 가진 스캔 순서변환기의 증가 게이트 수를 나타낸 것이다. 0.18μm 공정 셀 라이브러리<sup>[5]</sup>와 저전력 SRAM 컴파일러를 사용해서 합성한 결과이다. 0.3ns의 클럭 스쿼를 적용했으며, DFT를 고려하지 않은 수치이다.

표 4. 제안한 스캔 순서 변환기의 게이트 수.  
Table 4. Equivalent gate count for the proposed scan-order converter.

Gate count	Conventional	Proposed
Memory cells	158350	64176
FIFO controller	-	2211
Address generator	558	2505
Total	158898	68892

제안한 구조는 4개의 라인 메모리를 사용한 기존구조에 비해서 56.6%까지 게이트 수를 감축함을 알 수 있다. 일반적으로 JPEG 압축기의 게이트 수는 8만 정도이므로, ISP에 JPEG 압축기와 스캔 순서 변환기를 내장할 경우 필요한 게이트 수는 VGA의 경우 기존 방법에 비해서 38%의 감축이 가능해진다. 따라서 제안한 스캔 순서 변환기는 저전력을 지향하는 JPEG 압축기 내장 ISP에 적합한 특성을 가진다.

참 고 문 헌

- [1] J. Hurwitz, S.G. Smith, A.A. Murray, P.B. Denyer, J.Thomson, S. Anderson, et al., "A miniature imaging module for mobile applications," ISSCC Digest of Technical Papers, pp. 90-91, Feb. 2001.
- [2] H.S. Kim, J.Y. Kim, S.H. Hwang, I.C. Park, and C.M. Kyung, "Digital signal processor with efficient interpolation and histogram accumulation," IEEE Trans. on Consumer Electronics, Vol. 44, No. 4, pp. 1389-1395, Nov. 1998.
- [3] 죽촌유부, 디지털 CCD 카메라기술, 미래컴, 2001년.
- [4] A. El Gamal, and H. Eltoukhy, "CMOS image sensors," IEEE Circuits and Devices Magazine, Vol. 21, No. 3, pp. 6-20, May-June 2005.
- [5] R. Ramanath, W.E. Snyder, and G.L. Bilbro, "Demosaicing methods for Bayer color arrays," J. Electronic Imaging, Vol. 11, pp. 306-315, July 2002.
- [6] J.S. Lee, Y.Y. Jung, B.S. Kim, and S.J. Ko, "An advanced video camera system with robust AF, AE, and AWB control," IEEE Trans. on Consumer Electronics, Vol. 47, No. 3, pp. 694-699, Aug. 2001.
- [7] S.H. Lim, and A. El Gamal, "Integration of image capture and processing - beyond single chip digital camera," Proceedings of SPIE Electronic Imaging Conference, Vol. 4306, pp. 219-216, Jan. 2001.
- [8] K.H. Yoon, C.K. Kim, B.H. Lee, "Single-chip CMOS image sensor for mobile applications," IEEE Trans. on Solid-State Circuits, Vol. 37, No. 12, pp. 1839-1845, Dec. 2002.
- [9] TransChip, Inc., A CMOS imager with integrated ISP and JPEG compression, Feb. 2003.
- [10] ISO/IEC, ISO/IEC 10918-1:1994, Information technology - Digital compression and coding of continuous-tone still images: Requirements and guidelines, 1994.
- [11] K. Jack, Video Demystified: A Handbook for the Digital Engineer, LLH Technology Publishing, 2001.
- [12] Samsung Electronics, ASIC Databook: 0.18um 1.8V CMOS Standard Cell Library for Pure Logic Products, June 2001.

저 자 소 개



박 현 상

- 1991년 2월: 한국과학기술원 전기및전자공학과 학사
- 1993년 3월: 한국과학기술원 전기및전자공학과 석사
- 1999년 8월: 한국과학기술원 전기및전자공학과 박사
- 1991년 3월 ~ 2005년 2월: 삼성전자 시스템LSI사업부 책임연구원
- 2005년 2월 ~ 현재: 국립공주대학교 전기전자공학부 조교수
- 2005년 12월 ~ 현재: 삼성종합기술원 자문교수
- 주관심분야: 멀티미디어 SoC, 디지털 비디오 신호 처리, 동영상 압축