

JPEG이 내장된 ISP를 위한 전력 효율적인 스캔 순서 변환

박현상^{1*}

¹공주대학교 전기전자제어공학부

Power Efficient Scan Order Conversion for JPEG-Embedded ISP

Hyun-Sang Park^{1*}

¹Division of Electrical, Electronics, and Control Engineering Kongju National University

요 약 ISP와 JPEG 인코더 사이에는 라스터 스캔 순서의 데이터를 8x8 블록 스캔 순서로 변환하는 스캔 순서 변환기가 위치한다. 최근에 단일 라인 메모리를 사용함으로써, 하드웨어 규모를 감축한 스캔 순서 변환기가 제안되었으나 매 사이클마다 기입과 독출 동작을 수행함에 따라서 전체 전력 예산의 대부분을 SRAM이 소모하는 문제점을 야기했다. 본 논문에서는 SRAM에 대한 액세스 빈도를 줄이기 위하여 데이터 packer와 unpacker를 스캔 순서 변환 과정에 삽입함으로써, SRAM에 대한 액세스 빈도를 1/8로 줄이는 구조를 제안한다. 실험결과, 제안한 구조를 적용할 경우 SXGA 해상도에서의 SRAM 전력소모량은 16% 이하로 줄어든다.

Abstract A scan order converter has to be placed before the JPEG encoder to provide 8x8 blocks from the pixels in raster scan order. Recently a hardware architecture has been proposed to implement a scan converter based on the single line memory. Since both read and write accesses happen at each cycle, however, the largest part of the entire power budget is occupied by the SRAM itself. In this paper, the data packing and unpacking procedure is inserted in the processing chain, such that the access frequency to the SRAM is reduced to 1/8 by adopting a packed larger data unit. The simulation results show that the resultant power consumption is reduced down to 16% for the SXGA resolution.

Key Words : JPEG, Compression, ISP, Scan conversion

1. 서론

JPEG (Joint Photographic Expert Group)은 정지영상을 압축하기 위해 제정된 국제표준[1,2]이다. JPEG은 영상을 겹치지 않는 8x8 블록 단위로 처리하기 때문에 하드웨어 구현이 용이하고 압축과정에서 요구하는 메모리량이 적지만, 높은 압축률에서는 블록화 현상(blocking artifact)이 부각되는 단점이 있다. 이를 제거하기 위해서 JPEG2000[3]에서는 작은 블록 단위로 압축을 수행하지 않고, 8x8 보다 훨씬 큰 블록 단위 혹은 전체 영상에 대한 Wavelet 변환을 수행한 후에 압축하는 방식을 채택하고 있다. JPEG2000에서는 64x64 이상의 블록을 사용하고 있는데, 8x8 블록 단위로 동작하는 JPEG에 비해서 수직해상도가 8배 이상으로 늘어남으로써, 영상 데이터를 저장하기 위한 메모리 공간 역시 같은 비율로 늘어나게

된다. 이런 비용을 토대로 JPEG2000은 높은 압축률에서도 JPEG보다 뛰어난 주관적인 화질을 제공할 수 있게 된다. 그러나 낮은 압축률에서는 JPEG2000과 JPEG의 화질 차이는 주관적으로 차이가 없는데다, 압축된 영상 데이터를 저장하기 위한 비휘발성 메모리의 가격이 매년 지속적으로 하락함에 따라서 배터리 전원을 둔 모바일 시스템에서는 JPEG2000 표준이 실질적으로 사장된 상태이다.

최근 카메라가 장착된 휴대전화가 대중화되고, 특히 개발도상국 등에서의 초저가 카메라 장착 휴대전화에 대한 시장의 요구가 증대함에 따라서, 원가절감 차원에서 VGA나 SXGA급 낮은 해상도를 가지는 카메라 모듈에 이미지 센서, ISP, JPEG 인코더 등을 통합해야 하기 때문에, 이미지 센서, ISP, JPEG등이 내장한 SoC 형태의 이미지 센서 개발로 이어지고 있다.

*교신저자 : 박현상(vandammm@kongju.ac.kr)

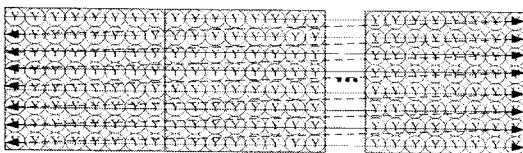
ISP는 이미지 센서로부터 입력되는 Bayer 패턴[4]의 영상으로부터 R, G, B 영상을 복원하여 이를 Y, Cb, Cr 영역으로 색변환하는 것이 가장 기본적인 기능이다. ISP는 ITU-R 601 규격[5]에 의거하여 색변환을 수행하고, 이를 chroma 4:2:2 형식의 부표본화[6]하여 Y, Cb, Cr 데이터를 상호 배치하여 출력한다[7]. 따라서 ISP는 라스터 스캔 순서로 영상 데이터가 출력되나, JPEG 인코더에는 8x8 블록 단위로 영상 데이터가 입력되어야 하므로, 라스터 스캔 순서의 데이터를 8x8 블록 스캔 순서로 변환해주는 장치가 ISP와 JPEG 인코더 사이에 위치해야 한다.

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. 이런 문제를 해결하기 위하여 라인 메모리를 하나만 사용하는 방법[8]이 제안되어서, 실리콘 면적 면에서 효과적인 스캔 순서 변환기의 구현이 가능하게 되었다. 그러나 이 방법에서는 단일 메모리를 한 개 사용하면서, 매 클럭 사이클마다 SRAM을 기입하거나 독출하는 동작을 수행하기 때문에 SRAM이 지속적으로 동작하는 상태가 되어, 전체 전력 예산의 대부분을 SRAM이 차지하는 결과를 낳았다. 본 논문에서는 SRAM의 전력 소모량을 감축하기 위하여 기존의 스캔 순서 변환 알고리즘을 변경하여, 실리콘 면적의 증가 없이 SRAM 동작에 소요되는 전력소모량을 SXGA 해상도의 경우 16% 이하로 감축하는 스캔 순서 변환기를 제안한다.

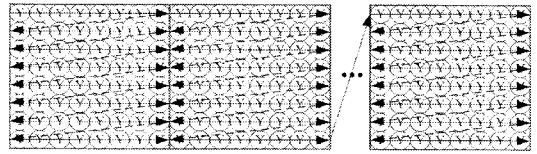
2. 스캔 순서 변환 시스템

2.1 스캔 순서 변환의 개요

그림 1에서 원 안의 Y는 휘도 화소를 나타낸다. ISP는 그림 1(a)와 같이 라스터 스캔 순서로 데이터를 출력한다. JPEG은 8x8 블록 단위로 압축을 수행하기 때문에 그림 1(b)와 같이 블록 스캔 순서로 데이터의 처리 순서를 변환해야 한다.



(a) ISP로부터의 라인 단위 데이터 출력 순서



(b) JPEG 압축을 위한 블록 단위 스캔 순서
[그림 1] JPEG 압축을 위한 스캔 순서 변환

2.2 단일 메모리 기반 스캔 순서 변환 알고리즘[8]

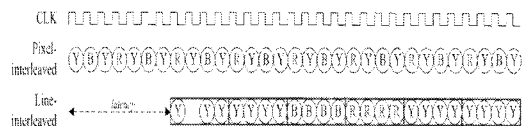
2.2.1 FIFO를 이용한 데이터 정렬

ISP에서는 ITU-R 656규격[7]에 근거하여 Y, Cb, Y, Cr과 같은 순서로 데이터가 상호 배치되어(interleaved) 출력되는데, 같은 속성을 가지는 Y 데이터는 16개가 연속적으로, Cb, Cr 데이터들은 각각 8개씩 연속적으로 출력되도록, FIFO를 이용해서 데이터의 순서를 정렬한다. 이렇게 정렬된 데이터가 라인 메모리로 입력될 경우 스캔 순서 변환 알고리즘은 휘도 영상만을 다루는 경우와, 4:2:2 색차형식의 컬러 영상을 다루는 경우에 있어서 차이점이 없어지게 된다.

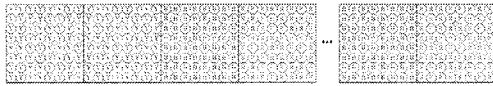
그림 2(a)는 연속적인 데이터를 만들기 위한 하드웨어 구성을 나타내며, Y성분을 위한 FIFO-Y와 Cb성분을 위한 FIFO-Cb의 최소 크기는 각각 8 바이트이며, Cr성분을 위한 FIFO-Cr의 최소 크기는 12 바이트이다. 그림 2(b)는 FIFO에 의해서 데이터가 정렬되는 형태를 도식적으로 나타내는데 이해를 돕기 위하여 4개의 데이터 단위로 정렬하는 경우를 나타냈다. 그림에서 원 안의 Y는 휘도 성분, B는 Cb 성분, R은 Cr 성분을 각각 나타낸다. 이와 같이 정렬을 하게 되면 출력되는 데이터의 형식은 그림 2(c)와 같게 되어, 좌측으로부터 8x8 블록 단위로 순차적으로 JPEG 압축을 수행하면, 2개의 8x8 Y 블록, 한 개의 8x8 Cb 블록, 한 개의 8x8 Cr 블록을 순차적으로 처리하는 결과가 되어, JPEG 인코더가 chroma 4:2:2 형식으로 부표본화된 데이터를 처리하는 표준 방식을 따르게 된다.



(a) 라인 단위로 상호 배치하는 하드웨어의 블록도



(b) 화소단위로 상호 배치된 데이터 입력과 라인 단위로 상호 배치된 데이터 출력의 타이밍도



(c) FIFO를 통해서 정렬된 데이터를 8x8 블록 단위로 블록화

[그림 2] FIFO를 이용한 유사 성분 단위의 데이터 정렬

2.2.2 스캔 순서 변환을 위한 SRAM 주소 발생 알고리즘

일반적인 경우 2 개의 라인 메모리를 사용하여 각 라인 메모리 전체에 대해서 독출과 기입을 반복적으로 수행함으로써 데이터 스캔 순서를 변환한다. 만약 임의의 주소에 기입과 독출이 동시에 가능한 라인 메모리가 있다면, 2 개의 라인 메모리를 활용해서 기입과 독출을 병행하지 않고 하나의 라인 메모리만을 활용하는 것이 이론적으로 가능하다. 이는 라인 메모리에 주소가 인가되는 동안 독출버스는 해당 주소의 데이터가 비동기적으로 출력되고 라인 메모리를 구동하는 클럭에 동기 되어서 해당 주소에 기입버스의 데이터를 입력하는 구조를 채택함으로써 구현할 수 있다. 현재 기입되는 8 라인에 해당하는 데이터는 다음 8 라인을 저장하는 단계에서 독출된다. 이런 경우에 독출과 기입이 동시에 가능한 주소 생성을 위한 알고리즘은 그림 3과 같다.

```

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

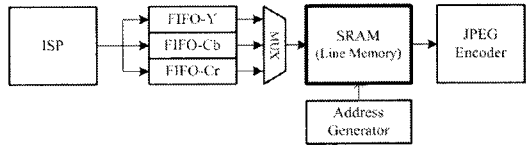
block_offset = 1;
line_offset = no_hor_bk;

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++){
                addr = anchor<<3 + h;
                line_number += line_offset;
                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 3] 순서 변환을 위한 라인 메모리 주소 발생 알고리즘

*no_hor_bk*는 통합 라인 메모리에 저장할 8x8 블록의 수를 나타낸다. 그림에서 수평방향의 해상도는 640이라고 가정한다. 그림에서 *i*, *v*, *h*는 각각 블록, 수직방향 화소, 수평방향 화소에 대한 인덱스를 나타낸다. *anchor*는 블록 내의 한 row의 각 화소가 공통적으로 가지는 주소 성분에 대응하는 변수이고, *addr*은 SRAM을 액세스하기

위한 주소를 나타내는 변수이다. 그림에서 *finished*라는 변수는 하나의 영상 프레임의 모든 화소가 독출 되었는지를 나타낸다. *block_offset*은 인접한 두 블록 anchor값의 차이를 나타내는 변수이며 초기값은 1이다. *line_offset*은 한 블록 내에서 수직방향으로 인접한 두 row anchor값의 차이를 나타내는 변수이며 초기값은 영상 프레임의 구성하는 수평방향 블록의 수와 같다. 또한 현재 8 라인을 기입할 때 사용했던 *line_offset*은 다음 8 라인을 독출할 때 사용할 *block_offset*에 해당하게 된다. 위에서 *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.



[그림 4] 화소 순서 정렬 시스템

그림 4는 상기 알고리즘에 기반을 둔 스캔 변환 시스템에 대한 블록도를 나타낸다. ISP의 출력단에 위치한 FIFO를 통하여 같은 성분의 화소 데이터들은 8의 배수로 묶여서, 단일 라인 메모리에 저장되도록 구성되어 있다.

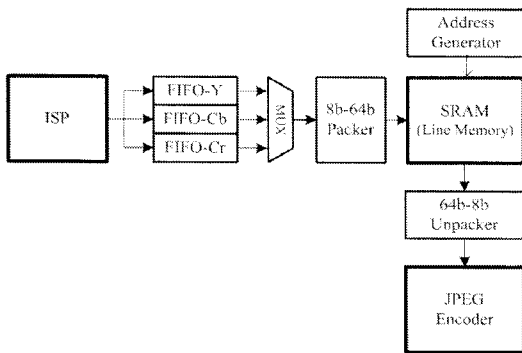
2.2.3 제한한 전력 효율적인 스캔 순서 변환 시스템

정적인 상태에서의 SRAM은 전력소모는 매우 낮으며, 기입이나 독출을 수행할 경우에만 주로 전력이 소모된다. 기입은 독출보다 10% 정도 전력 소모량이 높지만 큰 차이는 없으며, 메모리의 전체 용량이 커질수록 전력소모량은 증가하는 경향을 가진다. 또한 같은 메모리 용량이면 기입이나 독출되는 데이터 폭이 증가할수록 전력소모량도 증가한다 [9].

동적인 전력소모를 줄이기 위하여 SRAM을 액세스하는 과정에 대해서 계층구조를 구성하는 것이 바람직하다. 즉, 자주 사용되는 데이터는 작은 용량의 SRAM이나 레지스터에 저장하고, 드물게 사용되는 데이터는 고용량의 SRAM에 저장함으로써 고용량의 SRAM을 지속적으로 사용함으로써 야기되는 불필요한 전력소모를 감축하는 것이 가능하다.

스캔 순서 변환은 8-비트 데이터 단위로 수행되지만, 이보다 넓은 데이터 단위로 본다면 8x8 블록을 구성하는 8개의 8-비트 데이터 단위로 기입이나 독출이 진행된다.

따라서 라인 메모리로의 기입과 독출을 8-비트 데이터 단위로 수행하지 않고, 8개의 8-비트 데이터 단위로 수행할 경우 대용량 라인 메모리에 대한 액세스 빈도를 1/8로 감축할 수 있게 된다. 이를 구현하기 위하여 연속적인 8-비트 데이터 8개를 64-비트 단위로 묶어서 라인 메모리에 기입하고, 라인 메모리로부터는 출력되는 64-비트 데이터로부터 8-비트 데이터를 8 사이클에 걸쳐서 순차적으로 전달해주는 수단이 필요하다.



[그림 5] 제안한 화소 순서 정렬 시스템

```
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);
            addr = anchor;
            line_number += line_offset;
            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 6] 제안한 라인 메모리 주소 발생 알고리즘

그림 5는 제안한 화소 스캔 순서 변환기와 ISP, JPEG 인코더 등을 포함한 전체 시스템의 블록도를 나타낸다. FIFO를 통하여 Y, Cb, Cr 성분 별로 정렬된 데이터들은 8b-64b packer에 의해서 8개씩 묶여 64-비트 데이터로 구성된다. 이 데이터는 8 사이클마다 1번씩 라인 메모리에 기입된다. 또한 블록 스캔 순서를 고려하여 8 사이클마다 1번씩 라인 메모리로부터 64-비트 데이터가 독출된다. JPEG 인코더는 8-비트 데이터를 입력으로 받아들이므로, 64b-8b unpacker는 이 데이터를 8 사이클에 걸쳐서 시분

할하여 JPEG 인코더로 순차적으로 전달한다.

그림 6은 64-비트 단위의 데이터 단위로 라인 메모리를 액세스하기 위한 주소를 발생하는 알고리즘이다. 사용된 변수는 그림 3에 나타난 것과 동일하다. *addr*은 SRAM을 액세스하기 위한 주소를 나타내는 변수인데, 변수 *anchor*와 같은 값을 가진다. 그림 3에 나타났던 인덱스 *h*에 대한 루프는 64b-8b unpacker에 의해서 등가의 하드웨어로 구현된다.

2.2.4 구현결과 분석

표 1은 VGA(640x480) 해상도에 대해서 제안한 주소 발생 알고리즘, FIFO, 8b-64b packer, 64b-8b unpacker 등으로 구성된 전체 스캔 순서 변환 시스템의 등가 게이트 수를 나타낸 것이다. 0.13mm 공정 셀 라이브러리[9]와 저전력 SRAM 컴파일러를 사용해서 합성한 결과이다. 0.3 ns의 클럭 스쿠를 적용했으며, DFT(Design For Testability)를 고려하지 않은 수치이다. 동작주파수는 166 MHz 이상으로 합성가능하나, SXGA급 이하의 저해상도 ISP의 동작주파수는 60 MHz를 넘지 않기 때문에 동작속도에서의 제한은 없다. Packer와 unpacker를 추가하면서 전체 등가 게이트 수는 4.9% 증가했으나, 순수한 논리 회로가 차지하는 면적은 SRAM을 포함한 전체 면적 대비 10.2%에 불과하기 때문에 전력소모량은 전적으로 메모리에 의해서 결정된다.

[표 1] 합성 결과

Gate Count	[4]	Proposed
Memory cells	63896	64576
FIFO controller	2211	2211
Address Generator	2505	2236
Packer	0	2043
Unpacker	0	878
Total	68612	71944

표 2는 VGA 해상도에 대응하여 SRAM에서 사용되는 전력소모량을 비교하여 정리한 것이다. 기존 방법에서는 매 클럭 사이클마다 기입과 독출을 반복하기 때문에 평균 전력과 최대 전력은 동일하지만, 제안한 구조에서는 기입과 독출을 8 클럭 사이클마다 1번씩만 수행하기 때문에 평균 전력량은 최대 전력량의 1/8에 해당한다. 그러나 제안한 방법에서는 데이터 폭이 64-비트이기 때문에, 8-비트 데이터 폭을 사용하는 기존 방법에 비해서는 기입과 독출에 소요되는 최대 전력량은 2배 가량 증가하는 문제점이 있으나, 평균 전력량만을 비교하면 기존 구조보다 1/4 이하로 줄어든다.

[표 2] VGA 해상도에서의 메모리전력소모량

Power [uW/MHz]	10240x8	1280x64
Read power	25.94	45.34
Write power	29.44	63.54
Max power	55.38	108.88
Avg power	55.38	13.61

표 3은 SXGA 해상도에 대응하여 SRAM에서 사용되는 전력소모량을 비교하여 정리한 것이다. SXGA(=1280x1024) 해상도는 저가형 JPEG 내장 카메라 모듈에서 VGA 해상도 다음으로 보편적으로 사용되는 해상도이다. 기존 구조와 평균 전력량을 비교하면 기존 구조의 16% 이하로 줄어든다.

[표 3] SXGA 해상도에서의 메모리전력소모량

Power [uW/MHz]	20480x8	2560x64
Read power	51.88	58.52
Write power	58.88	78.57
Max power	110.76	137.09
Avg power	110.76	17.13

3. 결론

본 논문에서는 이미지 센서, ISP, JPEG 인코더 등을 통합하여 SoC 형태로 제작된 단일 칩 이미지 센서 구현이나, JPEG 인코더가 내장된 ISP를 위한 스캔 순서 변환기의 구조를 제안했다. 제안한 스캔 순서 변환기는 단일 라인 메모리를 사용하여, 휘도 영상뿐만 아니라, Y/Cb/Cr 컬러 영상에 대해서는 유연하게 처리할 수 있으며, 64-비트 단위로 SRAM을 액세스 함으로써 기존에 발표되었던 구조에 비해서는 SRAM에서 소모되는 전력량을 VGA 해상도에서는 25% 이하로, SXGA 해상도에서는 16% 이하로 감축할 수 있게 되었다. 따라서 제안한 스캔 순서 변환기는 저전력을 지향하는 SoC 형태의 이미지 센서에 적용하기에 적합한 특성을 가진다.

참고문헌

[1] ISO/IEC, *ISO/IEC10918-1: 1994, Information technology - Digital compression and coding of continuous-tone still images: Requirements and guidelines*, 1994.
 [2] G. K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30-44, Apr. 1991.

[3] ISO/IEC, *ISO/IEC15444-1: Information technology - JPEG2000 image coding system - Part1:Core coding system*, 2000.
 [4] J. Adams, K. Parulski, and K. Spaulding, "Color processing in digital cameras," *IEEE Micro*, vol. 18, no. 6, pp. 20-31, June, 1998.
 [5] Rec. ITU-R BT.601-5, *Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide screen 16:9 Aspect Ratios*, 1995.
 [6] K. Jack, *Video Demystified: A Handbook for the Digital Engineer*, LLH Technology Publishing, 2001.
 [7] Rec. ITU-R BT.656-4, *Interfaces for digital component video signals in 525-line and 625-line television systems operating at the 4:2:2 level of Recommendation ITU-RBT .601(PartA)*, 1998.
 [8] 박현상, "내장형 JPEG 압축을 위한 단일 메모리 기반의 스캔 순서 변환기", *한국방송공학회 논문지*, 제 11권, 제3호, pp. 320-325, 9월, 2006.
 [9] Samsung Electronics, *ASIC Databook: 0.13um 1.2V CMOS Standard Cell Library for Pure Logic Products*, Oct 2001.

박 현 상(Hyun-Sang Park)

[중심회원]



- 1993년 8월 : 한국과학기술원 전기및전자공학과 (공학석사)
- 1999년 8월 : 한국과학기술원 전기및전자공학과 (공학박사)
- 1998년 12월 ~ 2005년 2월 : 삼성전자 LSI사업부 책임연구원
- 2005년 3월 ~ 현재 : 공주대학교 전기전자제어공학부 부교수

<관심분야>
영상처리, 카메라 신호처리, 멀티미디어 SoC