# Mobile Phone Camera의 이미지 프레임 단위 처리를 위한 소형화된 Serial-Divider의 하드웨어 구현

김경린* · 이성진* · 김현수* · 김강주** · 강봉순*

*동아대학교 전자공학과

**삼성전기 주식회사

# Hardware Implementation of Minimized Serial-Divider for Image Frame-Unit Processing in Mobile Phone Camera.

Kyung-rin Kim* · Sung-jin Lee* · Hyun-Soo Kim* · Kang-joo Kim** · Bong-soon Kang*

* Department of Electronic Engineering, Dong-A University

** SAMSUNG Electro-Mechanics Co. Ltd.

E-mail : kyrini@didec.donga..ac.kr

## 요 약

본 논문에서는 모바일 폰 카메라의 프레임 단위 영상 신호 처리 과정에서 필요한 나눗셈 연산을 위한 나눗셈기 설계 방법을 제안한다. 나눗셈기의 내부 데이터 처리 방법에는 직렬 방식과 병렬 방식이 있다. 직렬방식은 실시간 연산이 가능한 반면에 많은 비교기와 Buffer Memory의 사용으로 인해 하드웨어 사이즈가 크다. 병렬방식은 실시간 연산을 할 수 없지만 하나의 비교기를 공유해서 연산함으로 직렬방식에 비해 하드웨어 크기를 줄일 수 있다. 이미지 처리를 위한 프레임 단위 연산은 실시간 연산을 필요로 하지 않으므로 하드웨어 자원의 효율성을 위해 직렬방식 나눗셈기를 적용한다. 입출력 조건을 동일하게 해서 병렬방식과 직렬방식의 나눗셈을 구현하여 하드웨어 크기를 비교했을 때 동일한 동작 주파수에서 직렬방식의 나눗셈기가 병렬방식의 나눗셈기의 대략 1/8 정도의 하드웨어 크기를 가지는 것을 확인하였다.

## ABSTRACT

In this paper, we propose the method of hardware-design for the division operation of image frame-unit processing in mobile phone camera. Generally, there are two types of the data processing, which are the parallel and serial type. The parallel type makes it possible to process in realtime, but it needs significant hardware size due to many comparators and buffer memories. Compare the serial type with the parallel type, the hardware size of the serial type is smaller than the other because it uses only one comparator, but serial type is not able to process in realtime. To use the hardware resources efficiently, we employ the serial divider since frame-unit operation for image processing does not need realtime process. When compared with both in the same bit size and operating frequency, the hardware size of the serial divider is approximately in the ratio of 13 percentage compared with the parallel divider.

## 키워드

Serial divider, Parallel divider, Comparator, Minimization

## Ⅰ. INTRODUCTION

As the requirement of consumer is augmenting, mobile phone is getting smaller, nevertheless functions that mobile phone includes are getting increased. In this reason, how we use small hardware resources for many functions of mobile phone is important issue.

In this paper, we propose hardware implementation of a minimized serial divider that is applied to an Image Signal Processing (ISP) in mobile phone camera.

There are many processes for ISP such as Auto Exposure (AE), Auto White Balance (AWB), etc. And each process needs division

operation. However, hardware implementation of divider is one of the most complicated functions. There are two types of divider which are the parallel divider and the serial divider. The parallel divider can process in realtime, but it has large hardware size since it needs comparators as many as quotient bits and buffer memories for synchronization of each signal[1]. The serial divider ought to hold the input signal during data is outputted because of serial operation. Therefore it can not process in realtime. But it has smaller hardware size than the parallel divider because it does not need many buffer memories, but needs one comparator as well.

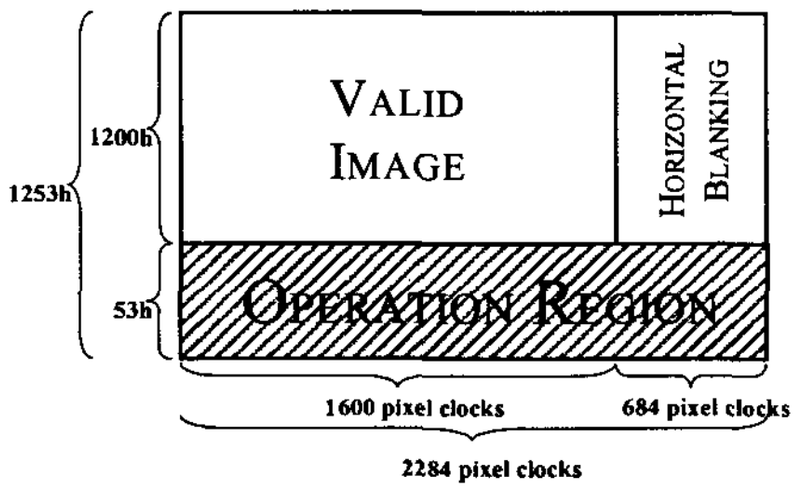


Fig. 1 Operation region of image sensor.

Figure 1 shows the operation region for frame-unit processing of image sensor [2]. The image sensor is composed of active region and nonactive region. Image is displayed when active region and then ISP is operated when nonactive region for a next frame because there are so many clocks between frames, which human can not perceive. For example, in the case of the 2-mega image sensor, operation region has about 120,000 clocks as shown in figure 1. ISP does not need realtime operation and input value is held during nonactive region. In this reason, we would apply the serial divider to ISP.

The paper is organized as follows. Section 2 presents the instructions of the proposed system, and section 3 depicts the results of hardware design and the performance comparison. Section 4 gives conclusions.

## II. SYSTEM MODEL

In this section, we explain the difference of operation methods between the parallel divider and the serial divider.

### 2.1. Parallel Divider

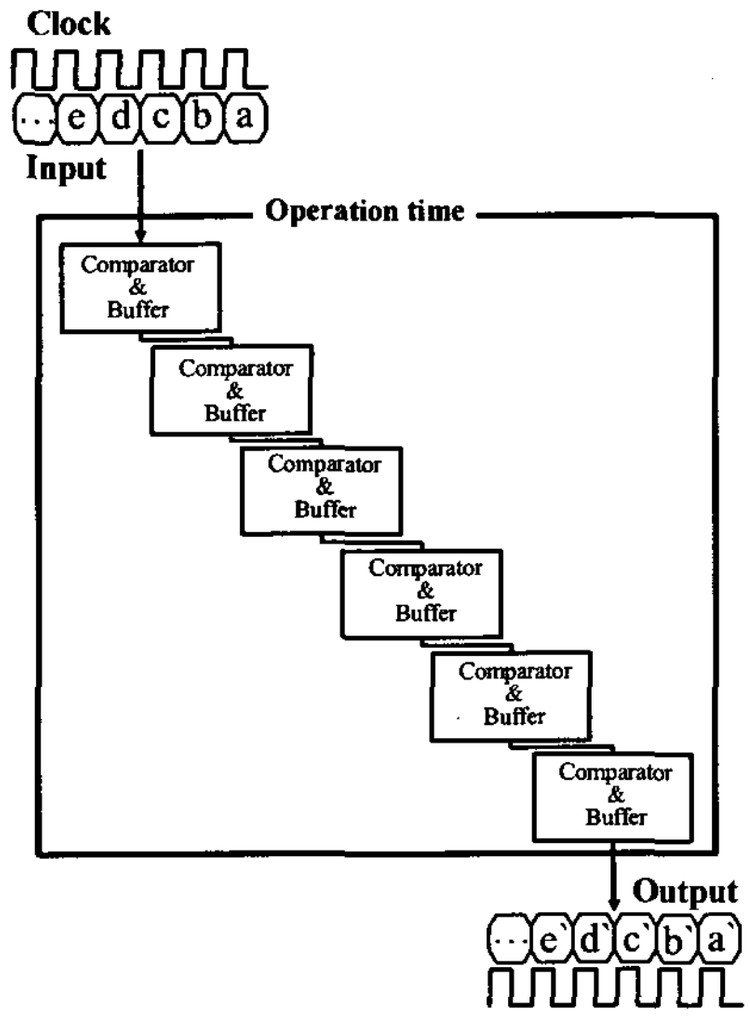


Fig. 2 Processing steps of parallel divider

Figure 2 shows processing steps of the parallel divider. Because the parallel divider operates as pipeline structure, it is able to output data at each clock cycle. In this reason, the parallel divider can process in realtime.

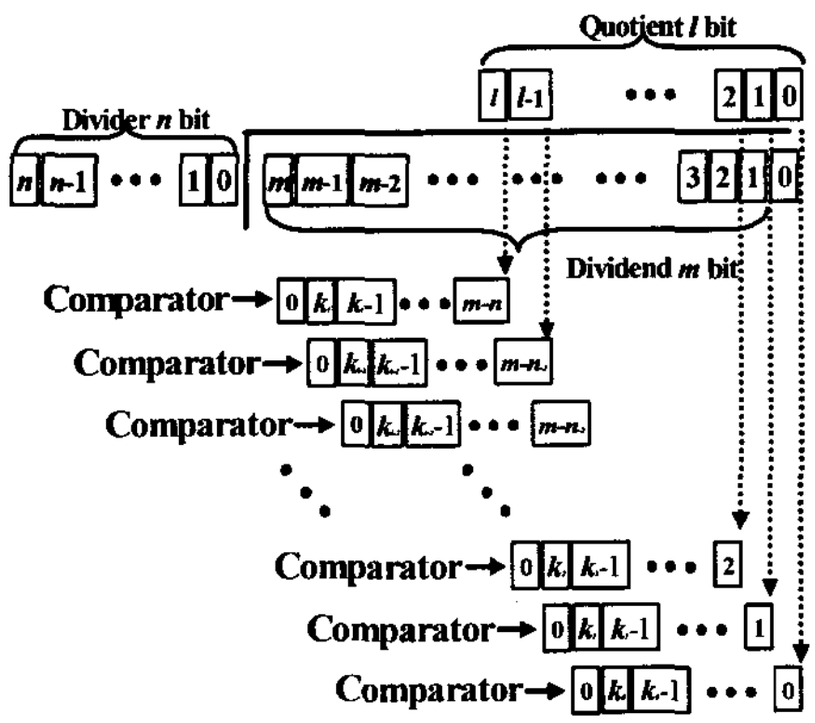


Fig. 3 Structure of Parallel divider

Figure 3 shows a structure of the parallel divider. It shows that how many comparators in the parallel divider are needed to make a quotient [1]. It needs a comparator for each divider and dividend. In this reason, parallel divider needs comparators as many as quotient bits.

### 2.2. Serial Divider

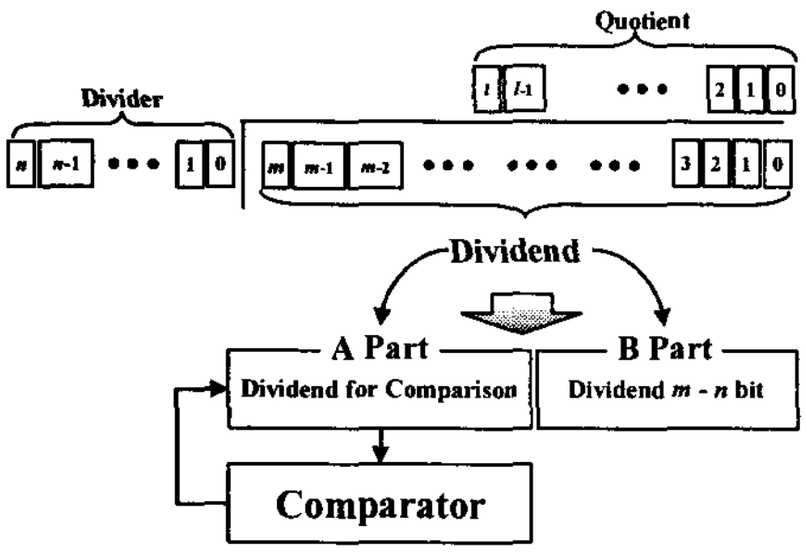


Fig. 4 Structure of Serial Divider

Figure 4 shows a structure of the serial divider [3]. There is only one comparator to compare a dividend with a divider. The division processing reiterates comparison between divider and dividend by sharing one comparator to derive the quotient.

For serial division operation, a dividend is separated into A-part that has n+1 bits and the B-part that has m-n+1 bits, which m dedicates MSB of the dividend and n dedicates MSB of the divider in figure 4. The A-part is compared with the divider and then MSB of the B-part would be alternated with LSB of A-part for next comparison.

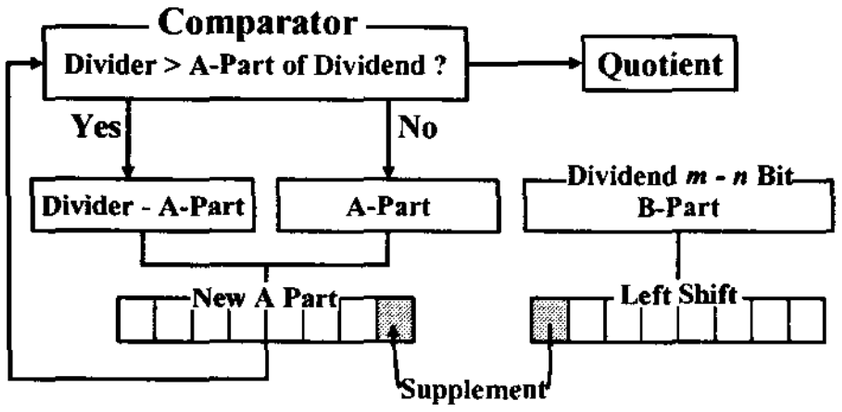


Fig. 5 Serial Divider Block Diagram

Figure 5 shows a block diagram of the serial divider [4]. If the A-part's value is larger than the divider's one, a novel value of the A-part becomes the A-part minus the divider. In addition, its LSB is supplemented with MSB of the B-part. If the divider's value is larger than the A-part's one, a novel value of the A-part becomes the values from the second bit to the last bit of the A-part's value and then its LSB is supplemented with MSB of the B-part.

The B-part is left shifted at each process. LSB of the novel A-part is alternated with MSB of the left shifted B-part for next comparison process.

If the divider's value is larger than the A-part's one, the quotient is saved as '0', but if the contrary condition, the quotient is saved as '1'.

To consider the remainder of this operation, LSB of the novel A-part ought to be transferred to '0'. If the remainder value is '0', the quotient is outputted. If the remainder value is '1', the quotient is added with '1' and then it is outputted. However, if every bits of the quotient are '1', it is not added with '1' to prevent an overflow [5].

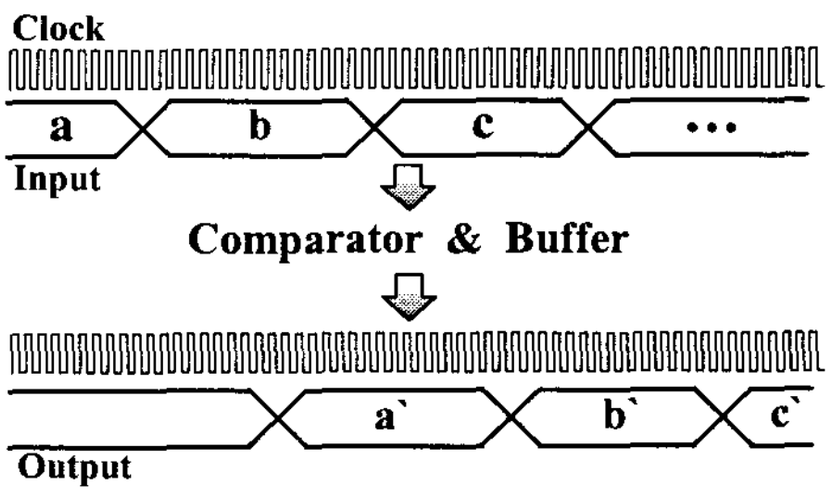


Fig. 6 Operation results of serial divider

Figure 6 shows operation results of the serial divider. Because of serial operation, input values are held until data is outputted as aforementioned. Hence, it can not be realtime process.

## III. Design of Performance

There are examples of the parallel divider and serial divider that have same input and output bit size. Four examples designed by using Verilog-HDL models are shown in Table 1, 2. After the verification, the models are synthesized by using the Synopsys synthesizer with the TSMC 0.25um library.

### 3.1. Parallel Divider

Table 1. The synthesize results of the parallel divider.

| Input | | Output | Gate Counts |
|---|---|---|---|
| Dividend [bit] | Divider [bit] | Quotient [bit] | |
| 33 | 24 | 13 | 8232 |
| 28 | 20 | 10 | 7968 |
| 24 | 16 | 8 | 6417 |
| 11 | 13 | 14 | 6440 |

Table 1 shows gate counts of the parallel divider in the different in/output bit size. The

average gate-count is 7265 as shown in Table 1.

### 3.2. Serial Divider

Table 2. The synthesize results of the serial divider.

| Input | | Output | Gate Counts |
|---|---|---|---|
| Dividend [bit] | Divider [bit] | Quotient [bit] | |
| 33 | 24 | 13 | 1058 |
| 28 | 20 | 10 | 990 |
| 24 | 16 | 8 | 838 |
| 11 | 13 | 14 | 884 |

Table 2 shows gate counts of the parallel divider in the different in/output bit size, nevertheless on a par with bit size of the parallel dividers in table 1. The average gate-count is 943 as shown in Table 2.

### 3.3. Comparison Hardware size

Table 3. Comparison of hardware size between the parallel divider and the serial divider

| | Parallel Divider | Serial Divider | Hardware size rate [%] |
|---|---|---|---|
| 1 | 8232 | 1058 | 12.85 |
| 2 | 7968 | 990 | 12.42 |
| 3 | 6417 | 838 | 13.06 |
| 4 | 6440 | 884 | 13.73 |

Table 3 shows comparison of hardware size between the parallel divider and the serial divider that have same in/output conditions. Each case, the hardware size of the serial divider is approximately in the ratio of 13 percentage compared with the parallel divider.

### 3.4. Operation time of dividers

Table 4. Comparison of operation time [clock]

| Output | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Parallel divider | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Serial divider | 16 | 32 | 48 | 64 | 70 | 86 | 102 |

Table 4 shows the operation time of this system. This example has 28 bits dividend, 20 bits divider and 10 bits quotient. The parallel divider generates the first quotient at the 14th clock and then other quotients are generated simultaneously with following clocks. However, the serial divider generates the first quotient at the 16th clock and then other quotients are generated each 16 clocks later.

## IV. Conclusion

In this paper, we propose the serial divider that has small hardware size in comparison with the parallel divider. The division operation is the requisite arithmetic-operation for ISP. However, it requires large hardware size compared with its capacities. Since the proposed serial-divider in this paper operates the division with significantly small hardware size as it employes one comparator, it is applied to any system. Hence, we are able to expect that the proposed system is more straightforward than parallel divider in ISP.


## ACKNOWLEDGMENT

The authors wish to thank the IDEC for its software assistance.


## REFERENCES


[1] Kyung-rin Kim, Jung-hwan Park, Kang-joo Kim, Bong-soon Kang, "Design of High Speed Divider for Multimedia System based on Mobile Application" The 16th Conference on KICS, IEEK, ICROS, pp. 63-66, Jun. 2007.

[2] MT9T013 1/4-Inch 3.1-Megapixel CMOS Digital Image Sensor, Micron Data Book, Jul. 2006.

[3] MILOŠ D. ERCEGOVAC, TOMÁS LANG, *DIVISION AND SQUARE ROOT Digit-Recurrence Algorithms and Implementations*, KLUWER ACADEMIC PUBLISHERS, 1994

[4] Douglas J. Smith, *HDL Chip Design, A Practical Guide for Designing, Synthesizing and Simulating ASICs and FPGAs using VHDL or Verilg*, Doone Pubns, Mar. 1998.

[5] M. R. Patel, K. H Bennett, "Analysis of speed of a binary divider using a variable number of shifts per cycle" Oxford Journals Mathematics & Physical Sciences Computer Journal, Vol, 21, No, 3, pp. 246-252, Mar. 1978