Implementation of a Large Data Processing Method for Embedded System and CMOS SNR Application

Chien-Hung Chen, Tai-Shan Liao, and Chi-Hung Hwang

Instrument Technology Research Center, National Applied Research Laboratories, Hsinchu, Taiwan
{cchung,tsliao,chhwang}@itrc.narl.org.tw

Abstract. The embedded system is the future trends of instrument and the larger memory capacity of embedded system is favor to different variety of applications. The commercial embedded systems are always restricted by their embedded memory capacity and required to be upgraded, especially in image application. This article reports a new design and development of an embedded system built in a CMOS image SNR measurement instrument. The new developed approach using the mix technique of large data processing (MLDP) method for CMOS SNR calculation is described. The MLDP method uses an external memory device as auxiliary memory in the regular embedded system to break the memory capacity limitation. The experimental results show the new method is applied successfully in CMOS SNR measurement and the calculated speed is increased almost 200 times compared to that of the traditional method even thought the processing data size is over the embedded system memory.

Keywords: Large data processing, Embedded system, External memory, CMOS SNR.

1 Introduction

Embedded system is a kind of application centric computing system designed for special purposes. Traditionally, an embedded system is implemented on a single microprocessor board with the programs stored in ROM. These kinds of embedded systems are used in various applications such as power plant, automobile control, house systems and information appliances etc.

The basic function of an industrial embedded system is to acquire and accumulate data about the status of the objects, and to control its operation. Besides, the images captured by most of the commercial image cameras are analyzed by embedded system [1-5]. The functions of image camera such as image data readout, image data compression, or image data storage, are easily implemented by embedded system. Furth more, some special functions such as dark current analysis [5,6], pattern noise (FPN) correction [7,8], or image signal-to-ratio (SNR) measurement are also integrated in embedded system applications of the cameras.

However, this kind of application usually needs large memory capacity to store the temporary data, which will be calculated later during sequent processing. For example, when the designer needs to know the SNR data of image sensor, the image data

R.-S. Chang, T.-h. Kim, and S.-L. Peng (Eds.): SUComS 2011, CCIS 223, pp. 28–36, 2011.

are measured several times and the SNR results are calculated by the developed algorithm. In the Personal Computer (PC), this problem is easily solved because the designer can add the high memory capacity hard driver to PC memory. But in the embedded system, the system memory was fixed when it was produced. Therefore, memory issues are very important and often impact significantly the embedded system's performance.

The designer can choose the commercial embedded systems with largest memory capacity when the designer buys them and use them to analyze the large data. However, the data capacity is always not enough to feed the need when we are in the memory hungry era. For example, the CMOS image sensor size increases from 30,000 pixels to 100,000 pixels. The image data size of each figure will also improve almost 3.3 times. The other proposed methods are data and memory optimization [9-11]. There are many optimization algorithms in limited memory space, such as maximum a posteriori (MAP) algorithm [12], blocking [13] or tiling [14]. Those algorithms, originally described in the late 1980's, were generally used. However, those algorithms are seldom used in embedded systems. The reason is that the process speeds of embedded systems are not as fast as PC. The memory optimization algorithms are heavy loading for embedded system.

After paper surveying, we present a variety of optimization techniques for data and memory used in embedded systems. The main idea in this paper is to extend well-known techniques implemented in the paralleled processing and memory optimization techniques. We combine hardware and software to develop an algorithm and we call it the mix technique of large data processing (MLDP) method. The idea is that USB flash disk is used to extend memory capacity of embedded systems. Then the parallel processing method is used to read data and calculate the parameters of image.

In this article, we also propose a novel design for the NI CompactRIO embedded system built in the CMOS SNR measurement instrument. The embedded system incorporates industrial 400 MHz MPC5200 processor in order to support a real-time system. We also integrated field programmable gate array (FPGA) device to implement the digital logic control circuit. The FPGA can replace many digital logic parts, implement I/O control, as well as realize the large data processing algorithm that we propose in this paper. Finally, the CMOS SNR measurement instrument is well established and the SNR data is obtained successful by large data processing algorithm of this paper.

2 The CMOS SNR Measurement Algorithm

We propose a CMOS SNR measurement algorithm that is used to obtain the performance of CMOS images from the repeated image data measurements. The flowchart of CMOS SNR processing is shown in fig 1. At the beginning, the system parameters such as image exposure time, readout speed, as well as row data output is set. Then, the luminous of light source is controlled under the image pixel saturation state. The CMOS image data is measured and stored repeatedly. Those image data will be used to calculate the SNR of image data.

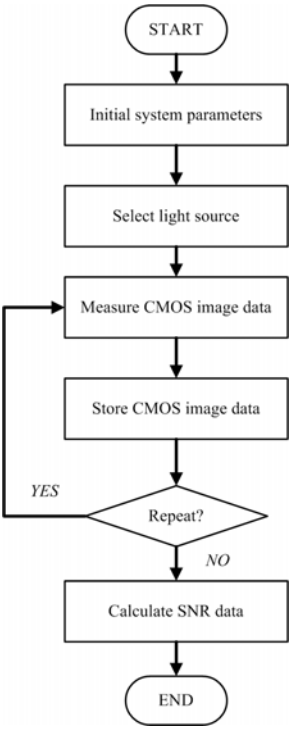

Fig. 1. Flowchart of SNR measurement

3 The MLDP Method Description

The MLDP method will be described in this paragraph. We divided the MLDP method into three steps. They are extended embedded system memory, processing data storing, and retrieving parallel data readout.

First, we use additional and external USB flash to expand the embedded system memory issue. Most of the commercial embedded systems have USB flash protocol and their kernel. We use this feature to add USB flash for extending memory of embedded system, wherein the additional memory space can store the temporary calculated data of CMOS SNR results.

Second, we store the temporary data in USB flash by hexadecimal numbering system. In order to save the extended memory space, we change the decimal numbering system to hexadecimal numbering system and store the information as Windows file system. This method can save one character space for each pixel. The CMOS image sensor which has 512 x 256 pixel size and 12 bits analog-to-digital converter resolution is demonstrated in this research. There are 131,072 total pixel numbers. In this normal state, the memory needs four character spaces to store the data, for example 1025 or 1513. If we use the hexadecimal numbering system, the row data will be shorter and $1025_{(10)}$ becomes $401_{(16)}$, $1513_{(10)}$ becomes $5E9_{(16)}$. Larger image pixel numbers will be saved more additional memory spaces.

Third, we need to read the data out and use those row data to calculate the SNR of CMOS image. The binary data readout technique was used in this part. After experimental test, the binary readout method is 3-times faster than the normal method. Then, those data will be processed by parallel method.

In memory structure, we create two-dimensional array to store the image row data individually and calculate the SNR parameter. The data status in the two-dimensional array is shown in fig 2. Each array stores the same pixel data between different frames. After image processing, we only use 256 arrays of system memory for data temporary storing, that also can predict the total system used and it also can significantly reduce the loading of embedded system memory in large data processing.

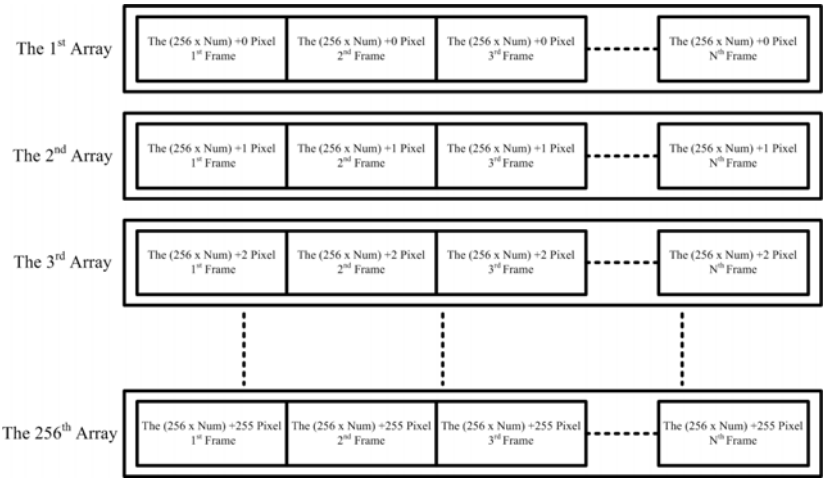

Fig. 2. The pixel data status in different array

4 System Implementation

The CMOS image SNR data is obtained by repeatedly measuring the image data difference between the different luminous of light sources. Generally, the SNR data of CMOS image need to be obtained in 3 status of luminous of light source. There are 25%, 50%, and 75% luminous. This is because that the traditionally SNR measurement methods are really complicated and waste the human resources.

A block diagram of the CMOS image SNR measurement is shown in fig 4. The hardware is composed of an NI CompactRIO embedded system, a 3M gate reconfigurable I/O FPGA, a CPLD component, CMOS image readout circuits, an integrating sphere, a light source controller, and a fluxmeter. The operating frequency of the embedded system is 400 MHz, 128 Mbyte system memory, and the input-output (I/O) interface is via a FPGA background board for the port expansion. The processing result can be transmitted to personal computer and display it later.

Initially, the integrating sphere functions a stable light source (shown as A in Fig. 3) to CMOS image sensor. The pixel numbers of CMOS image sensor are 512 x 256 (shown as B in Fig. 3). The image sensor, which includes 12-bits analog-to-digital

(A/D) converter, can be set some parameters, such as exposure time, or data readout mode. The time control and parameter setting of image sensor are via the CPLD (C) component, and then pass through the 12-bits image row data to the buffer (D). The digital image data is transmitted to embedded system via the high speed I/O ports (E). The maximum speed of I/O ports is 10 Mbyte/s. The FPGA (F) is used to control the sampling data in I/O ports and transmit the image row date to embedded system. The fluxmeter (I) can measure the luminous data of integrating sphere and transmit the luminous data to the embedded system. After all, the embedded system can modify the luminous of light source automatically or manually by the light source controller (H).

The embedded system (G) is the main core in this system. The embedded system function controls all the processes in real time and operates without any interruption. The NI CompactRIO real time embedded system was used in this research, and it can offer stand-alone embedded execution for deterministic real time applications. The NI CompactRIO is implemented by LabVIEW[15,16] graphical development software.

We also use the FPGA (F) to read the luminous data from fluxmeter (I), transmit the luminous data to embedded system, and sent the light source changed commands to light source controller. The development software was also LabVIEW [15,16] graphical program.

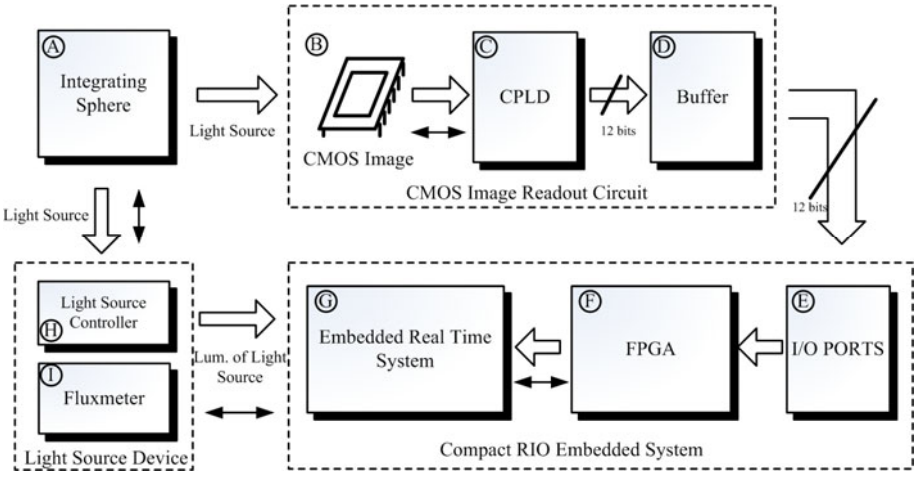

Fig. 3. The CMOS SNR measurement system block diagram

5 Results and Discussion

The accomplished CMOS SNR measurement system is pictured in fig 4. The complete system is composed of an embedded system & FPGA backboard, a fluxmeter, an integrating sphere, and image readout circuits with CMOS sensor. In fig 5, the screen clearly displays one array image row data. The X axis is pixel number and Y axis is gray level of pixel data. In the upper graph, there are 131072 (512 x 256) total pixels and 6 dead pixels are found. The dead pixels will affect the accuracy of SNR calculation. Therefore, the dead pixels need to be fixed before the processing. The bottom graph shows the pixel data, which the dead pixel has been fixed.

Table 1 shows the SNR results of 6 different frames measured using the same luminous light source. The SNR data designates the mean values and standard deviation (STD) of CMOS frames data during the different number frames. From table 1, we can observe that frame numbers of 250 or 300 are the most proper numbers for the CMOS row data measurement in this study. This experimental results show that the mean value and STD become small if the frame data are large enough. Basic on probability theory or statistics, the result is more significant for the applications if the data set or data population approach the normal distribution. A small standard deviation indicates that the data points tend to be close to the mean; therefore, the result will also be more precise. In addition, the high frame numbers will increase the processed data size and add the embedded system memory loading. Therefore, we purposed a new method which includes the extend USB memory for data storing.

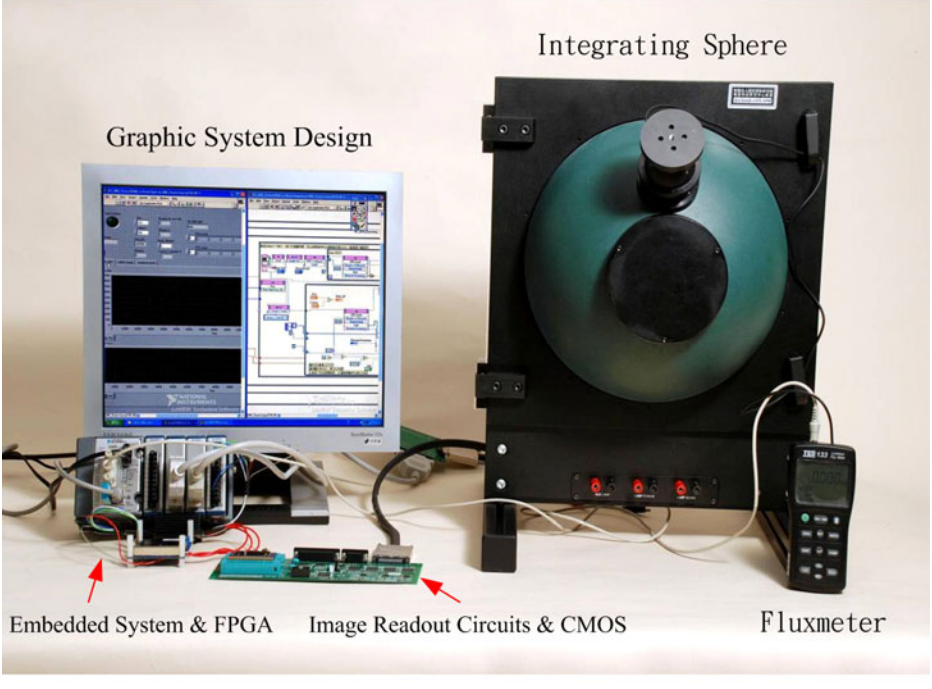

Fig. 4. The CMOS SNR measurement system

In data processing time experiment, we use 5 different frames to compare the normal method and MLPD method, and also test the processing speeds between inside memory and extended memory. The normal method of inside memory is that the image row data is stored into the memory of embedded system and calculated. In the normal method of extended memory, the image row data is stored to extended memory, then read them out and calculate pixel by pixel with regular methods. The MLDP is the large data processing method that we propose in this paper. The data sizes of

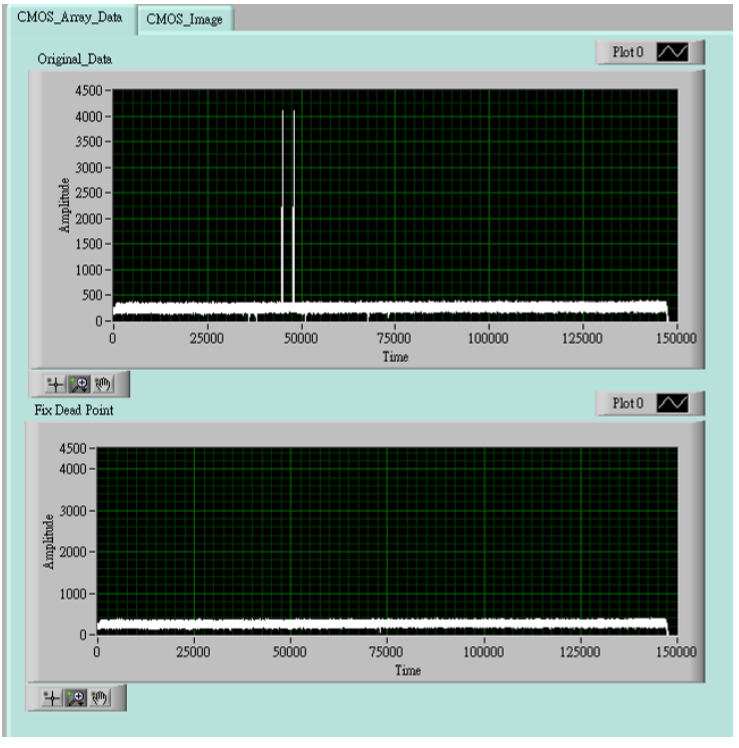

Fig. 5. The CMOS image row data

Table 1. The mean value and standard deviation between different measurement frames

FRAME NUMBER	50	100	150	200	250	300
MEAN	43.639	43.654	43.524	43.533	43.415	43.411
STANDARD DEVIATION	0.953	0.714	0.609	0.614	0.521	0.493

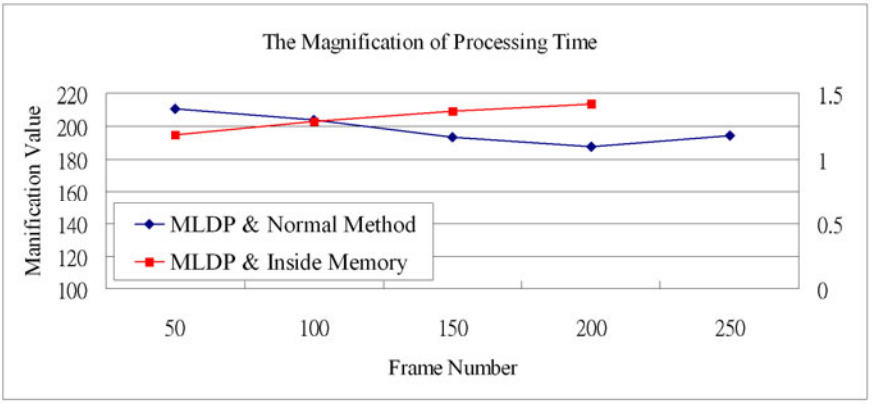

Fig. 6. The magnification of processing time

processing are 25 M(50 image frames), 50 M (100 image frames), 75 M(150 image frames), 100 M(200 image frames),, and 125 M(250 image frames), respectively. The 250 frames (125 MByte) data cannot process in inside memory because of overflow. The reason is that the system memory only carries 128 MByte, and the program of system and directive of program also need memory. We compare MLDP method with normal method of extended memory and with normal method of inside memory. The inside memory method uses the least processing time in the experimental results. But if the row data are lager than system memory, the embedded system cannot operate. The magnification values of processing time shows in fig 6. The X axis of fig 6 is frames number, the left side of Y axis is the magnification of MLDP and normal method, and right side of Y axis is the magnification of MLDP and inside memory. The processing time of MLDP method is almost 200 times faster than the normal method of extended memory. The performance of MLDP method still can have almost 80 % of the inside memory method. Those results verify that the MLDP method can process the large data in embedded system with considerable accuracy even thought the data size are higher than system memory.

References

1. Cheng, A.M.K., Wang, Y.: A dynamic voltage scaling algorithm for dynamic workloads. Journal of Signal Processing Systems 52, 45–57 (2007)
2. Smith, M., Miller, J., Daeninck, S.: A test-oriented embedded system production methodology. Journal of Signal Processing Systems 56, 69–89 (2009)
3. Talavera, G., Jayapala, M., Carrabina, J., Catthoor, F.: Address generation optimization for embedded high-performance processors: a survey. Journal of Signal Processing Systems 53, 271–284 (2008)
4. Pardo, F., Dierickx, B., Scheffer, D.: Space-variant non-orthogonal structure CMOS image sensor design. IEEE Journal of Solid State Circuits 33, 842–849 (1998)
5. Fossum, E.R.: CMOS image sensors: electronic camera on a chip. IEEE Transactions on Electron Devices 44, 1689–1698 (1997)
6. Correia, J.H., de Graaf, G., Kong, S.H., Bartek, M., Wolffenbuttel, R.F.: Single-chip CMOS optical microspectrometer. Sensor and Actuators A 82, 191–197 (2000)
7. Mühlmann, U., Ribo, M., Lang, P., Pinz, A.: A new high speed CMOS camera for real-time tracking applications. In: ICRA 2004 (New Orleans), pp. 5195–5200 (2004)
8. Schwarz, M., Ewe, L., Hauschild, R., Hosticka, B.J., Huppertz, J., Kolnsberg, S., Mokwa, W., Trieu, H.K.: Single Chip CMOS Imagers and Flexible Microelectronic Stimulators for a Retina Implant System. Sensor and Actuators A 83, 40–46 (2000)
9. Panda, P.R., Catthoor, F., Dutt, N.D., Danckaert, K., Brockmeyer, E., Kulkarni, C., Vandercappelle, A., Kjeldsberg, P.G.: Data and memory optimization techniques for embedded systems. ACM Trans. on Design Auto. of Electronic Systems 6, 149–206 (2001)
10. Catthoor, F., Wuytack, S., DeE Greef, E., Balasa, F., Nnachtergaele, L., Vandecappelle, A.: Custom Memory Management Methodology. In: Exploration of Memory Organization for Embedded Multimedia System Design. Kluwer Academic, Dordrecht (1998)
11. Byrd, R.H., Lu, P., Nocedal, J.: A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific and Statistical Computing 16, 1190–1208 (1995)

12. Schurgers, C., Catthoor, F., Engels, M.: Memory Optimisation of MAP Turbo Decoder Algorithms. IEEE Transaction on VLSI Systems 9, 305–312 (2001)
13. Golub, G., Van Loan, C.F.: Matrix Computations. Johns Hopkins University Press, Baltimore (1989)
14. Wolfe, M.: More iteration space tiling. In: ACM/IEEE Conference on Supercomputing, pp. 655–664 (1989)
15. Aspey, R.A., McDermid, I.S., Leblanc, T., Howe, J.W., Walsh, T.D.: LABVIEW graphical user interface for precision multichannel alignment of Raman lidar at Jet Propulsion Laboratory, Table Mountain Facility. Review of Scientific Instruments 79, 094502 (2008)
16. Blacksell, M., Wach, J., Anderson, D., Howard, J., Collis, S.M., Blackwell, B.D., Andruczyk, D., James, B.W.: Imaging photomultiplier array with integrated amplifiers and high-speed USB interface. Review of Scientific Instruments 79, 10F506 (2008)