eISP: a Programmable Processing Architecture for Smart Phone Image Enhancement

Thevenin Mathieu, Letellier Laurent, Paindavoine Michel, Schmit Renaud, Heyrman Barthelemy

eISP: a Programmable Processing Architecture for Smart Phone Image Enhancement

Mathieu Thevenin, Laurent Letellier
and Renaud Schmit
CEA, LIST Embedded Computing Lab
Bd 528 MB 94
F-91191 Gif sur Yvette – France
Email: FirstName.LastName@cea.fr

Barthelemy Heyrman
and Michel Paindavoine
UMR CNRS 5158
Rue Alain Savary
F-21000 Dijon – France
Email: FirstName.LastName@u-bourgogne.fr

***Abstract*— Today's smart phones, with their embedded high-resolution video sensors, require computing capacities that are too high to easily meet stringent silicon area and power consumption requirements (some one and a half square millimeters and half a watt) especially when programmable components are used. To develop such capacities, integrators still rely on dedicated low resolution video processing components, whose drawback is low flexibility. With this in mind, our paper presents eISP – a new, fully programmable Embedded Image Signal Processor architecture, now validated in TSMC 65nm technology to achieve a capacity of 16.8 GOPs at 233 MHz, for 1.5 mm^2 of silicon area and a power consumption of 250 mW. Its resulting efficiency (67 MOPs/mW), has made eISP the leading programmable architecture for signal processing, especially for HD 1080p video processing on embedded devices such as smart phone.**

I. Introduction

Video sensors, particularly as used in smart phones, have become a familiar part of everyday life. This market is placing drastic constraints on the power consumption and silicon areas of video image processors. Sensors are thus necessarily associated with signal processors, not only for color reconstruction purposes, but also for intrinsic image enhancement. The allowable power consumption of embedded image processors is currently a few hundred milliwatts, while smart phone computing capacities exceed many billions of operations per second (GOPs). To afford such capacities, today's phone integrators rely on dedicated video processing components that offer limited flexibility. With the programmable computing architectures now on the market, High Definition (HD) video processing, which requires several tens of GOPs, cannot be built into mobile devices. Because integrators prefer to embed their own image enhancement functions, it is crucial to make the computing resources behind the currently available sensors as programmable and as flexible as possible. Section two of this paper presents some common color reconstruction and image enhancement methods, followed, in section three, by an estimate of the computing resources necessary to implement them. Section four then describes the proposed Embedded Image Signal Processor (eISP) architecture - a fully programmable architecture designed to handle HD 1080p video streams (*i*.e. 1920×1080 resolution at 25 frames per second), thereby anticipating the capacities needed for next-generation smart phone video sensors. This discussion of eISP architecture, which is synthesized and routed for TSMC 65 nm technology, likewise details the silicon area, power consumption and computational characteristics of our new concept.

II. Video Pipeline

A set of embeddable processes is necessary for the capture and enhancement of video images and photographs produced by Complementary Metal Oxide Semiconductor (CMOS) sensors. Various processing steps and algorithms can be incorporated into the image reconstruction chain, also known as the "video pipeline".

Because evaluation of sensor exposure parameters is difficult, captured images do not usually cover the complete sensor dynamic range. Histogram normalization [1] or local adaptive methods [2] can, however, be applied to make optimum use of the available dynamic range for pixel value coding.

G	B	G	B	G	B
R	G	R	G	R	G
G	B	G	B	G	B
R	G	R	G	R	G
G	B	G	B	G	B
R	G	R	G	R	G

Fig. 1. Standard Bayer color array filter.

Noise reduction is an essential processing step. Image signal-to-noise ratio diminishes with a reduction in pixel size, which is also the trend observed in mobile consumer products. Fixed-pattern noise (FPN) is essentially linked to disparities in substrate properties and can be successfully characterized and removed. Electronic noise caused by thermal excitation is an additive white Gaussian noise. Its effect can be limited by applying a Gaussian blur or local adaptive filter. Finally, so-called "salt and pepper" noise occurs when pixel values are corrupted with respect to their neighborhoods. This type of noise is particularly visible to the human eye. Median filters or techniques enabling removal of extreme pixel values from a given group are best suited to this type of denoising [3].

Single-sensor processing systems operate by capturing a luminance image. To reconstruct color data, sensor pixels are covered by a filter that alternatively allows each pixel to record a primary color. The best known type of color filter is the Bayer filter depicted in Figure 1. The white balancing step adjusts gain for color components, to make final image color look natural. The existing literature proposes many methods for illuminant color estimation [4], ranging from a simple "grey world" assumption to the "Retinex" [5] approach. The demosaicing step generates three plans (red, green and blue or luminance/chrominances) for each of the colors obtained from the raw image [6]. Finally, it is possible to improve visual image quality using contrast enhancement and edge sharpening techniques.

Following these steps, it is usually necessary to apply a high-pass filter for sharpening purposes; but local adaptive filters are also used to sharpen edges. Contrast enhancement is likewise often needed for better visual perception of image displays. This step involves processes ranging from common gamma correction to more sophisticated "tone mapping" techniques [7] .

III. Computing Resources

By studying a set of representative algorithms for the processing steps described above, it was possible to estimate the computing resources needed for HD 720p and HD 1080p video reconstruction. The algorithms selected for this study were as follows:

- fixed pattern noise correction;
- noise reduction by median filtering;
- bilateral filtering (to represent local adaptive filters);
- application of convolution-based 5×5 filters;
- estimation and correction of white balance ("grey world" and "white patch" techniques);
- bilinear and constant-hue demosaicing.

The number of processor operations dedicated to computing filter results was also studied. Results of this study demonstrated that more than half of processor cycles are devoted to address computation and program control [8]. The computing capacity required for a video pipeline to process HD 1080p, 1920×1080 streams at 25 frames per second, using the previously listed algorithms, is 44.5 GOPs. As shown in table I, most of these resources are employed for edge sharpening (51%), median filtering (22%) and demosaicing of the raw image (21%). Such analysis also serves to determine the basic operations required to run the algorithms.

TABLE I

Computing resources required to reconstruct HD 720p and HD 1080p video stream.

Type of Process	HD 720p 20MPx/sec (GOPs)	HD 1080p 52MPx/sec (GOPs)	Process Represen--tativeness
FPN removal	0.4	1.0	2%
White Balancing	0.7	1.9	4%
Demosaicing	3.5	9.2	21%
Median denoising	3.7	9.8	22%
Sharpening	8.5	22.6	51%
Total:	**16.7 GOPs**	**44.5 GOPs**	**100%**

IV. eISP Architecture

This section describes the eISP architecture, which can be programmed to process video streams with different resolutions, namely HD 720p and HD 1080p. This programmability enables support of a wide range of algorithms, in particular those requiring access to pixel neighborhoods.

A. State of the Art

Most of the now available mobile consumer products employ system-on-chip (SOC) architectures that are built around general-purpose embedded processing cores coupled with dedicated coprocessors. Such solutions lack flexibility, since their processing parameters are "frozen" for as long as a non-reconfigurable SOC, which may be used in more than one generation of products, remains in place. Strong market demand for more flexible real-time video processing components has thus generated a flurry of research activity, both in industrial laboratories and academic institutions (*e*.g. University of Stanford or MIT). One offshoot of this is the family of chips proposed by Stream Processors Inc., for real-time video stream processing [9]. Another is the Coarse-Grained Reconfigurable Image Stream Processor (ISP) [10] that can handle HD videos. While the latter is not fully programmable, it offers greater flexibility than dedicated processing components. The image memory that must be associated with it is nevertheless "costly" in terms of silicon area. Among yet other developments, SiliconHive markets a customizable architecture, known as HiveFlex, which accommodates 4 to 128 programmable processors [11]; and Xetal proposes a programmable, massively-parallel processor integrating 320 computing elements [12]. SIMPil architecture [13] has 4096 parallelized processors for computation of as many pixel blocks. With their high power consumption (higher than a half Watt) and large silicon areas (higher than 2 mm^2), none of these components are directly embeddable in smart phones. Many of them also need an image memory. Storage of an HD plan in fact requires several mm^2 of silicon area in 65 nm technology.

B. Parallelism

As an example, for a processor running at 233 MHz, only three processor cycles per pixel would be available for processing an HD 1080p image stream. This is far from enough for the several dozen cycles needed to perform even a simple convolution. As already seen above, more than 50% of processor cycles are used for address computation and program control. Different kinds of parallelism must be implemented to increase the processing time available per pixel. Instruction-level parallelism (ILP) is supported by use of Very Long Instruction Word (VLIW) processors. Spatial parallelism is ensured by parallel use of several processors,

each assigned to a different pixel. Temporal parallelism is supported by pipelining processes and performing control, address computation and data access as background tasks.

C. Computing Tiles

eISP architecture is made up of programmable computing tiles that integrate two-way VLIW processors specifically designed to limit power consumption and silicon area usage. Unlike a conventional VLIW processor, the two-way design calls for all operators and registers to be shared by the two data paths. Processor data path width is 24 bits, and its core logic comprises 5200 two-port equivalent gates. A scratchpad memory option (up to two 2^{24} addressable 24 bit words, typically 256×24 bits) is likewise proposed. By running two instructions per clock cycle, the processor core performs a total of 400 MOPs at 200 MHz. The VLIW processors operate in Single Instruction Multiple Data (SIMD) mode and are all associated with the same program memory. This is the part of the tile that is dedicated to computation. It collaborates with a neighborhood controller designed to limit power consumption and silicon area. Its function is to convert the incoming pixel stream to a form (*i*.e. pixel values and relevant neighborhood data) that can be directly accessed by the processors. In this way, data access instructions are performed as background tasks, and address computation is completely eliminated at processor level. The neighborhood controller is designed to limit power consumption and silicon area while allowing direct processor access to all neighborhood data. Each computing tile contains an input/output interface that enables integration of pixel values from the incoming stream and serves to reconstruct the outgoing stream from tile-computed data. Such a tile typically contains 4 to 16 processors. Processor operating frequency is defined when the program is compiled, thus affording optimum control of system power consumption. Complex processing operations require chaining of tiles to build a complete eISP processing architecture. Pixel transfers between tiles take place via a Time Delay Multiplexed Access (TDMA) bus to enable reconfiguring of the order in which the tiles transmit data. This bus also provides a clock signal that keeps the whole architecture synchronized for tile chaining purposes.

V. Implementation and Results

This section describes how a video pipeline is ported to the eISP architecture. It first presents the algorithms and their arrangement on the different computing tiles. It then provides hardware implementation results (power consumption and silicon area data).

A. Algorithms

A video pipeline was ported to the two-way VLIW processor to validate the eISP architecture. This chain included the following operations:

- fixed pattern noise correction;
- histogram normalization;
- noise reduction by defective pixel deletion;

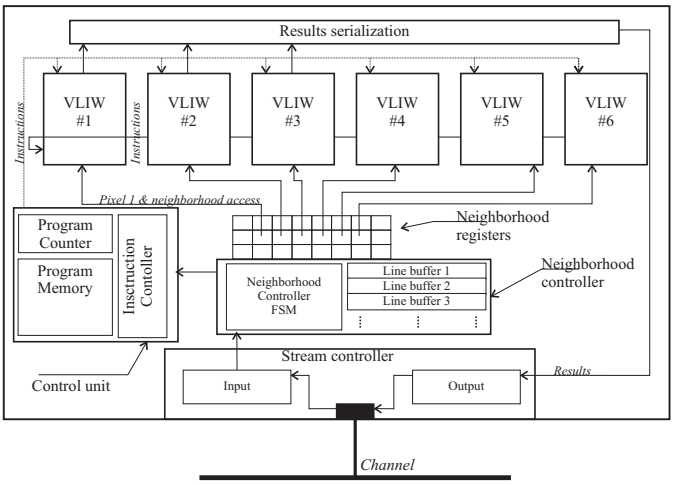

Fig. 2. A 6 processors computing tile.

- bilinear filter demosaicing;
- edge sharpening.

Fixed pattern noise cancellation can be achieved with five cycles per pixel and does not require neighborhood access.

Histogram normalization requires eight cycles per pixel. Partial histograms are contained in the processor scratchpad. The interframe delay is then used to combine them into a complete histogram.

Noise reduction is implemented by replacing the pixel value by the average neighborhood value, if the latter differs significantly from the former (*i*.e. falls outside a tolerance range of 12.5% or $\frac{1}{8}^{th}$ of the initial value). With two-way VLIW processors, this operation requires fourteen cycles and access to a 5×5 neighborhood, which is less ”costly” than a median filter.

Gamma correction makes use of a lookup table (LUT) stored in the processor scratchpad and requires 4 cycles with the two-way processor.

Automatic white balance is estimated using the grey-world method. During the relevant cycle, a counter is incremented and the pixel value is multiplied by a predefined coefficient. The system takes advantage of interframe delay to compute these multiplication coefficients. Four processor cycles are required, but no an access to pixel neighborhood is necessary.

Bilinear demosaicing by the two-way VLIW processor requires ten processor cycles and access to a 3×3 neighborhood.

Finally, edge sharpening for the three color channels derived from demosaicing is performed by 3×3 convolution, using 12 processor cycles.

B. Adaptability to Algorithms

All the proposed video pipeline algorithms cannot be executed by a single computing tile. This would require 85 processor cycles, *i*.e. at least 32 processors at 233 MHz on a single tile. Data interdependency would further complicate such a configuration, particularly for access to pixel neighborhoods. A tile computing capacity (in cycles) can be obtained by the equation 1.

$$\left(\frac{Clock_{Proc}}{Clock_{Pixel}} - 1\right) \times ProcNb \tag{1}$$

The best way to simplify integration of the algorithms is therefore to distribute them over more than one tile. An ideal arrangement, *i*.e. six tiles containing six processors each, is shown in Figure 3. This architecture can perform 16.8 GOPs, or 2.8 GOPs per tile, at 233 MHz. It consumes 40 mW per tile, for an efficiency of 70 MOPs/mW; and its total silicon area is 1.5 mm^2 for 240 mW.

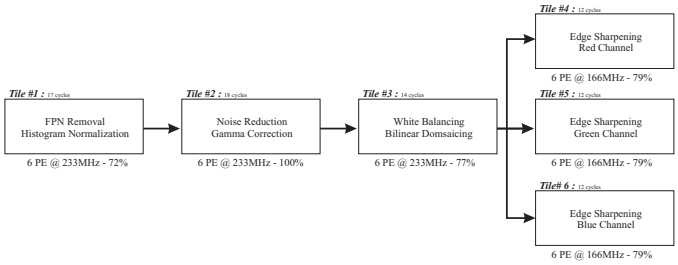

Fig. 3. Algorithms arrangement on the different computing tiles.

VI. Conclusion

For today's competitive embedded processor market, where mobile devices still rely essentially on dedicated processing components, this paper proposes a new solution: eISP, a fully programmable computing architecture for real-time video processing. Not only does eISP allow control of silicon area and power consumption (1.5 mm^2, 250 mW),it also proposes enough real computing capacity (16.8 GOPs) to accommodate the HD 720p and HD 1080 video streams that will become the standards for future smart phone imaging. Results given here were validated on an eISP model synthesized and routed in TSMC 65 nm technology for a frequency of up to 400 MHz – an example of a routed tile is shown in Figure 4. With this design it is possible to customize computing capacity to individual requirements, by increasing the number of processors per eISP computing tile, their operating frequencies or the overall number of tiles. Such an architecture also allows incorporation of cutting-edge algorithms. Further improvements, for example integration of dedicated coprocessors, are now being studied, with a view to extending eISP technology to telecommunication applications.

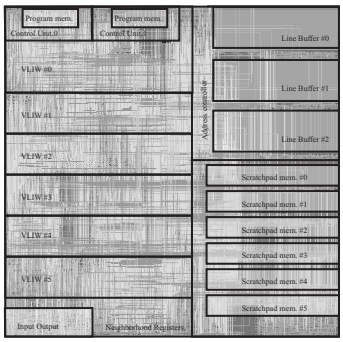

Fig. 4. Layout of a single computing tile containing 6 VLIW processors, scratchpads and a 3×3 neighborhood access – TSMC 65nm 0.13 mm^2 .

References

[1] **J. C. Russ, *The Image Processing Handbook - Fifth Edition*. 270 Madison Avenue - New York, NY 100: CRC Press, 1997.**

[2] **K. Goh, Y. Huang, and L. Hui, "Automatic video contrast enhancement," *Consumer Electronics, 2004 IEEE International Symposium on*, pp. 359–364, 1-3, 2004.**

[3] **M. Motwani, M. Gadiya, R. Motwani, and F. C. Harris, "Survey of image denoising techniques," in *Proceedings of GSP 2004*, Santa Clara, CA, September 2004, pp. 27–30.**

[4] **J. V. Autor, "Color constancy and image segmentation techniques for applications to mobile robotics," Ph.D. dissertation, Universitat Politecnica De Catalunya, 2005.**

[5] **L. Meylan and S. Süsstrunk, "High dynamic range image rendering using a Retinex-based Adaptive filter," *IEEE Trans. Image Process.*, vol. 15, no. 9, pp. 2820–2830, December 2006.**

[6] **B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicing: color filter array interpolation," in *IEEE Signal Processing Magazine*, 2005, pp. 44–54.**

[7] **K. Devlin, "A review of tone reproduction techniques," Department of Computer Science, University of Bristol, Tech. Rep. CSTR-02-005, November 2002.**

[8] **M. Thevenin, M. Paindavoine, L. Letellier, and B. Heyrman, "Processor extensions for CMOS sensor imaging in camera phone," in *Photonics Europe 2008 - Photonics in Multimedia*, vol. 7001, Apr 2008.**

[9] **B. Khailany, J. Williams, T. adn Lin, E. Long, M. Rygh, D. Tovey, and W. Dally, "A Programmable 512 GOPS Stream Processor for Signal, Image, and Video Processing," *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 202–213, Jan 2008.**

[10] **J. Chen and S.-Y. Chien, "Crisp: Coarse-grained reconfigurable image stream processor for digital still cameras and camcorders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 9, pp. 1223–1236, Sept 2008.**

[11] **C. A. Pinto, A. Beric, S. P. Singh, and S. Farfade, "Hiveflex-video vsp1: Video signal processing architecture for video coding and post-processing," *Multimedia, 2006. ISM'06. Eighth IEEE International Symposium on*, pp. 493–500, Dec. 2006.**

[12] **R. Kleihorst, A. Abbo, A. van der Avoird, M. Op de Beeck, L. Sevat, P. Wielage, R. van Veen, and H. van Herten, "Xetal: a low-power high-performance smart camera processor," *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, vol. 5, pp. 215–218 vol. 5, 2001.**

[13] **H. H. Cat, A. Gentile, J. C. Eble, M. Lee, O. Vendier, Y. J. Joo, D. S. Wills, M. Brooke, N. M. Jokerst, and A. S. Brown, "SIMPil: An OE integrated SIMD architecture for focal plane processing applications," in *Proc. 3rd International Conference on Massively Parallel Processing using Optical Interconnections, Maui*, 1996, pp. 44–52.**