

A hierarchical and parallel SoC architecture for vision processor

Kuizhi Mei^{a)}, Bin Zhang, and Chenyang Ge

*Institute of Artificial Intelligence and Robotics, University of Xi'an Jiaotong
No.28, Xianning West Road, Xi'an, Shaanxi, 710049, P.R. China*

a) meikuizhi@mail.xjtu.edu.cn

Abstract: This paper presents a hierarchical and parallel SoC (System on Chip) architecture for vision processor. The vision computing is divided into 3 task level parallel computing modules, which are vision decision, feature reorganization (or pattern generation), feature extraction. In the proposed architecture, there are two separately buses to integrate the 3 computing modules, and also the new interrupt for RISC processor to implement the synchronization between the hardware modules and software. The human-face detecting and tracking application demo has been mapped on the proposed architecture and verified on the FPGA. Architecture performance is also analyzed to show the proposed is more suitable for vision applications with higher image resolution.

Keywords: vision processor, SoC architecture, parallel, FPGA

Classification: Integrated circuits

References

- [1] F. Paillet and D. Mercier, "Design Solutions and Techniques for Vision System On a Chip," *IEEE ASIC/SOC 2000*. [online] <http://www.grosmi.net/Asic/ASIC2000.2up.pdf>
- [2] R. Etienne-Cummings, P. Pouliquen, and MA. Lewis, "A vision chip for color segmentation and pattern matching," *EURASIP Journal on Applied Signal Processing*, vol. 7, pp. 703–712, 2003.
- [3] Y. Watanabe, T. Komuro, and S. Kagami, *et al.*, "Multi-target tracking using a vision chip and its applications to real-time visual measurement," *J. Robotics and Mechatronics*, vol. 17, no. 2, pp. 121–129, 2005.
- [4] J. Schlessman, *et al.*, "Heterogeneous Mpsoc Architectures for Embedded Computer Vision," *ICME (IEEE International Conference on Multimedia & Expo)*, 2007.
- [5] C.-C. Cheng and C.-H. Lin, *et al.*, "iVisual: An Intelligent Visual Sensor SoC With 2790 fps CMOS Image Sensor and 205 GOPS/W Vision Processor," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 127–135, Jan. 2009.

1 Introduction

Hardware implementation for vision processing was originally some simple image binarization (1993, NCP Retina), gray level image processing (1998, PVLSAR2.2) and simple object tracking (2000, VAMPIRE) [1]. In 2003, a vision processor for colorfulness diagram separating and pattern matching was designed in John Hopkins University, it presented a continuous vision task from feature extraction to decision making [2]. And the research of vision processor is also starting from 1997 in Tokyo University, a chip for multi-object detecting and tracking was designed in 2003 [3]. For real-time computing is one of the tough problem to solve, a bi-bus SoC architecture for vision processor is proposed in this paper. Task level parallelism among feature extraction, feature reorganization (or pattern generation) and decision making generates the synchronous problem, and also an interrupt scheme and storing format for feature graph are proposed to solve the synchronization among the three modules. The paper is organized as following: in section 2, the bi-bus vision processor SoC architecture is proposed and explained; in section 3, the interrupt scheme and storing format for feature graph is described; in section 4, human face detecting and tracking application mapped on the architecture is described; in the conclusion, we give the architecture performance compared with state of the art of the design of *iVisual* in [5].

2 The bi-bus SoC architecture for the vision processor

Some typical SoC architectures for vision processors are already proposed. The heterogeneous MPSoC architecture based on a single on-chip AMBA bus in [4] presented a general vision application for the whole input-image processing, some dedicated IP modules are integrated in the chip such as *Optical Flow Detecting* and *Background Subtraction*. While the pipeline processing mode in [5] presented a tight coupling between the image sensor and the processing modules, some feature extraction circuit was designed according to the dedicated application, it was mainly used for some specific application such as the visual environment analysis.

Though the architecture in [4] is more general, the massive data transferring in the vision application may easily cause collision for the processing images data needed to transfer through the bus to the *Optical Flow* or the *Background Subtraction* modules. The architecture in [5] is more dedicated: the tight coupling mode between the *GP* and *FP* module is not adapted to new pattern generation from the primary features such as luminance, chrominance, binary, edge and etc.

So the proposed vision processor architecture is presented as Fig. 1 shows. To correspondence to Marr's 3 level vision processing, the architecture is composed of 3 main modules: RISC processor implemented by Leon2 processor, *Feature Extraction* (FE) and *Feature Reorganization* (FR). The sensing images from camera is firstly processed by the *FE* module and the sampled luminance image and the primary features of the original input images are got, then these feature images are stored in *SDRAM-2*. *FR* processes the

primary features in *SDRAM-2* to get some high-level features such as the matching points between neighboring images, connected areas.

In Fig. 1, the *FE* and *FR* modules are all connected to the same on-chip AHB bus. Decision making and other control tasks are implemented by Leon2 CPU which is connected with peripherals and external *SDRAM_1* by *AHB Bus_1*. *SDRAM_1* is used as the main memory of the RISC processor. And the feature images storing in *SDRAM_2* can only be read by Leon2 RISC through *Bridge2* connecting AHB Bus_1 and *AHB Bus_2*.

The *FE* module is now designed only including some primary operations, such as edge detecting, binarization of image, luminance, mainly using convolution product. Except for the luminance image, other feature images are all **binary values**. The various feature operating modules are all used as the master of the AHB bus and the feature images are stored through the *AHB Bus_2* by FIFOs. And for binarization of the features, data transferring volume is greatly reduced. From the architecture we could see that the vision algorithm can be distributed in three modules which the task parallelism is implemented, but the synchronization problem is generated for the distribute computing. And the synchronizing scheme is proposed in the following.

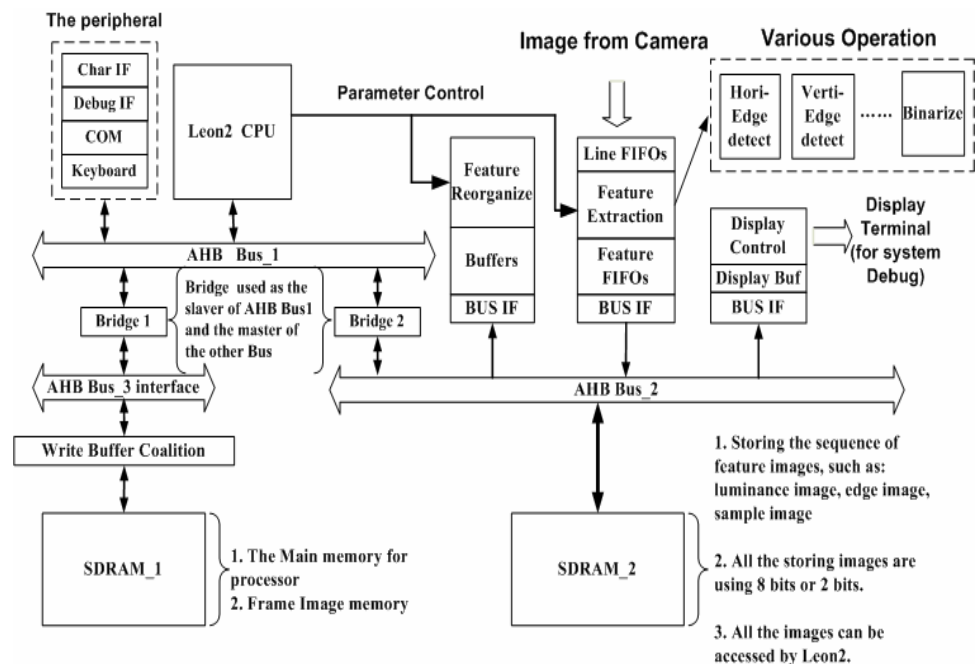


Fig. 1. The SoC architecture for the proposed vision processor

3 Synchronization scheme to solve distribute computing

The key problem to solve the synchronization in distribute computing is how every module could know which frame in-image they are processing. Given that the *FE* module is extracting the features of *n*th frame in-image, the decision making algorithm in RISC processor may process the features of (*n*-

k th frame in-image. The synchronization is solved by the interrupt scheme of Leon2 processor and the added properties of storing feature images.

There are 16 interrupt traps in Leon2 RISC for hardware interrupt using. To signify the operation of the sequence frame feature image storing, three interrupts are added also with some registers value set, they are listed in the following and shown in Fig. 2:

3.1 Frame Synchronization Interrupt (INT 14) :

When a new frame image is input with the Frame Sync pulse, the interrupt is generated in the positive edge of the pulse to notify the RISC to configure some parameters needed to storing the feature images in *SDRAM_2*, such as the starting address, the enable bit in the register *Image_en* to signify whether or not storing some feature images.

3.2 Frame storing Finishing Interrupt (INT 13) :

When the feature images are finished storing, an interrupt is generated to signify and also some bit in the *Image_over* register are set for inquiring. For the RISC processor easily knowing the feature images stored, some frame image information is also stored, such as: the sequence number, *Image_over* register value and etc.

3.3 Interrupt generated by FR module (INT 7)

Taking the frame differencing operation in the *FR* module as an example, when differencing operation is finished, a new feature reorganization of frame-differ image is stored in *SDRAM_2*, and also an interrupt is generated.

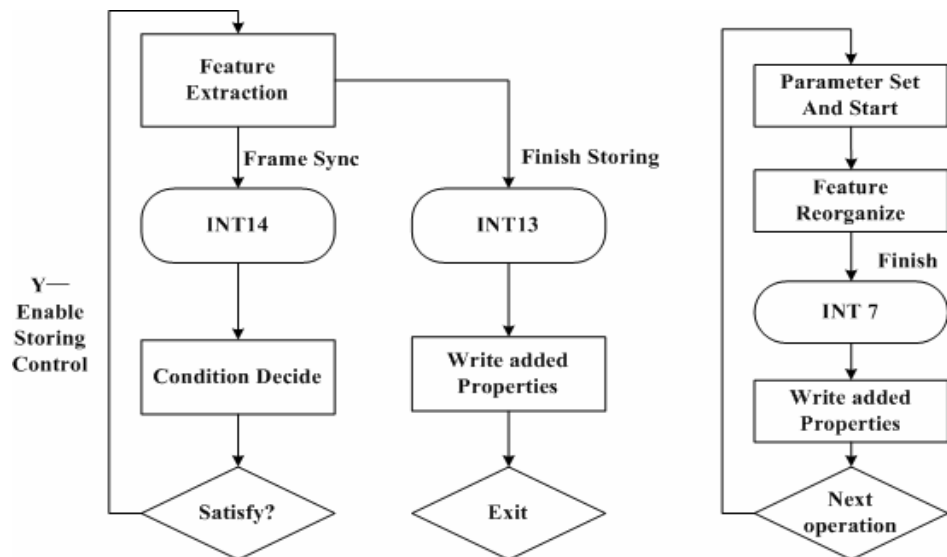


Fig. 2. Interrupt scheme for FE and FR

4 Effectiveness of the proposed processor

Human Face detecting is one popular problem in computer vision. There are many methods for face detecting: using knowledge, pattern matching, statistical learning and features. Among those, we select the detecting method based on skin_color feature for it remains unchangeable under the variance of luminance, rotating and sheltering. The program flow of the detecting and tracking algorithm mapping on the proposed processor is shown in Fig. 3. By the operation of image filtering, segmentation, clustering and decision based on prior-knowledge, the position of the human face is detected from the input frame image.

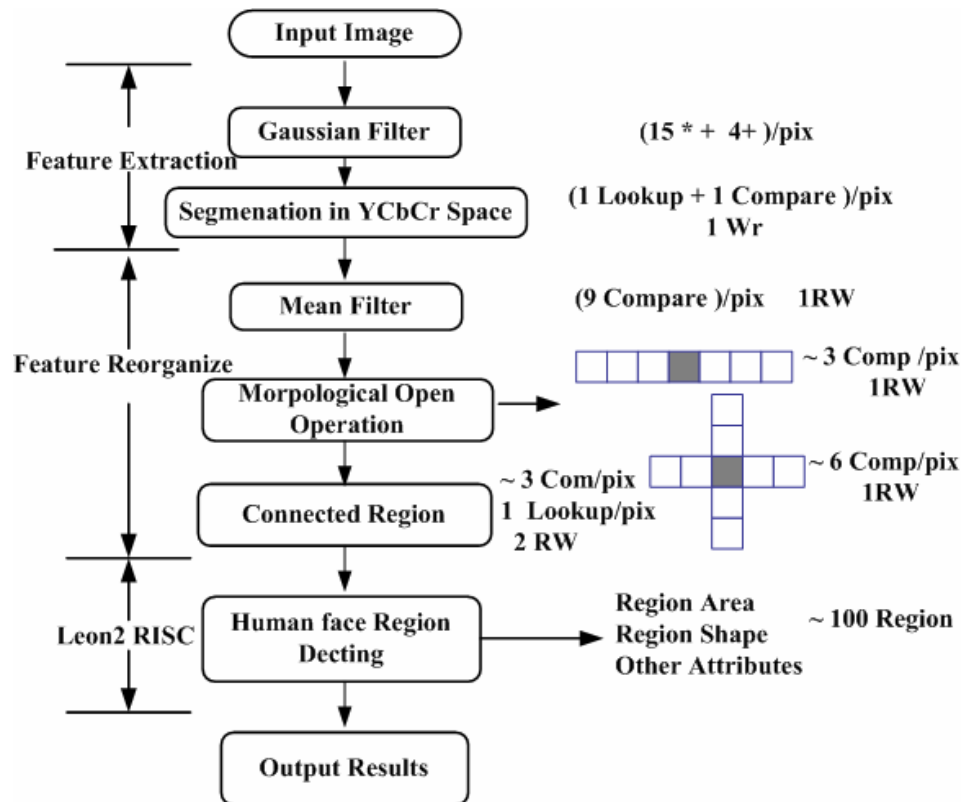


Fig. 3. Framework of face detecting on the processor and its complexity analysis

In Fig. 3, the features used are YCbCr space segmentation image. The feature are got by the *FE* and stored in *SDRAM_2*. Then the *FR* module can get the feature data from *SDRAM_2* and generate some new pattern which is the connected region. Executing the algorithm in Leon2 then computes the face position and output the command to control the camera by UART port. So the key of face tracking vision task is: firstly, detect the human face in the current frame image; then decide the human face according to the history information and some decision rules. The computing complexity of each pixel and its memory access in every step are also shown in Fig. 3. *Lookup* operation usually means very complex computing such as $\arctan(Cr/Cb)$ in *Segmenation in YCbCr Space*. For frame_in image with

800×480 at 100 frame/s, computing power needed is about 1651 MOPS in *FR* and *FE*. Some details in *FE* and *FR* in detecting are as following:

4.1 Skin color segmentation and computation

Luminance and chrominance are segmented in HIS space without the sensitiveness to light variance. The threshold value of the segmentation for skin_color is based on statistical analysis. Mean filter is used as a nonlinear smoothing filter while the image noise can be eliminated and image details can be remained. The filter mask selected is 3×3 .

4.2 Open operation of mathematical morphology in the FR module

It is mainly to get the connected region from the feature images using the morphologic computing. Firstly, the morphologic erosion is to eliminate the noise and minor regions in the feature image. And then dilation process is to remain the main connected region. In the experiment, the dilation select the cross mask in 3×3 , and the erosion select the horizontal mask in 1×7 as shown in Fig. 3.

4.3 Connected region generated in the FR module

Building the optional skin region is based on the rule of 4-neighboring clustering. Then these regions are taken as the optional human-face region to decide in Leon2 RISC according to the attributes of the face as shown in Fig. 3. The 4-neighbor connected region marking can be implemented in scanning the binary image only once, other features of the connected region can also be got simultaneously, such as the outside rectangle position, efficient pixels, and etc.

5 Conclusion

This paper presents a hierarchical and parallel SoC architecture for vision processing. The vision computing is divided into 3 tasks and realized a distributed parallel computing. The proposed architecture uses two separate buses to integrate the 3 computing modules, and also the interrupt mode to implement the synchronization among the hardware modules. The human face detecting and tracking application has been effectively and real-timely verified on the FPGA.

Supposing the access efficiency of SDRAM_2 is 72% in 50 MHz and 1 8 bit luminance-graph and 4 bi-value feature-graphs stored in SDRAM_2, *FR* module read or write SDRAM_2 8 times in total, *Leon2* read SDRAM_2 3 times (1 8 bit+2 feature-graphs). Needed Memory bandwidth for processing 1 frame is 360 K words: $800 \times 480 \{1 \times 8/32 + 4 \times 1/32 + 8 \times 1/32 + 1 \times 8/32 + 2 \times 1/32\}$, the maximum processing speed can attain up to 100 frame/s. Architecture performance comparison in Table 1 shows the proposed architecture more suitable for vision processing with higher resolution input images.

Table I. Different Architecture Parameters in ours and [5]

Architecture Parameters	[5]	Ours
Image Resolution	128×128	800×480 (maximal)
Feature Extraction	1 with high complex (FP)	Reconfigurable (4 simple)
On-chip Memory for Feature Processing	1Mbits (BM for 64 frames)	800Byte×2 (Using 3×3 convolution operator for feature extraction)
Processing speed (Memory Band Constrained)	The throughput is 360fps.	100frame/s (50M×72%/360K)
Programmability	Difficult programming in Global Processor (GP)	Hierarchical in HW-SW segmentation and easy in FR configuring

Acknowledgments

The authors thank the multi-processor SoC group members in the Institute of AIAR, they are Qing Guo, Yuhai Li and Hao Lei. Thanks also to the fund of High Technology Plan in China with the grant No. “2009AA01Z307”, “2009AA011709” and the Foundation of “SJ08ZT14-7” and “NSFC-60905007”.