Advanced Science and Technology Letters
Vol.28 (CIA 2013), pp.77-81
http://dx.doi.org/10.14257/astl.2013.28.15

An FPGA Implementation of Shape Recognition Algorithms for Two-Leg Walking Robots

Young Chun Kwon[1], Yeon-woo Kim[2],
Sunhan Jeong[2], and Nakhoon Baek[1,*]

[1] School of Computer Sci. and Eng., Kyungpook National Univ., Daegu 702-701, Korea
[2] System LSI Division, Samsung Electronics, Suwon, Korea
kown100@nate.com, sanaeida@nate.com,
sunahn86@nate.com, oceancru@gmail.com

Abstract. We present an FPGA implementation of image processing algorithms, which is attached to a special purpose embedded board. Though typical embedded boards are weak in their computing powers, we strengthen them with FPGA-based parallel processing chips. Our implementation finally shows remarkable speed-ups. We present our overall architecture and implementation details. Our system shows a reasonable solution to enhance the computing power of relatively low-power devices.

Keywords: FPGA, parallel processing, two-leg walking robot, image processing, embedded system.

1 Introduction

When an digital image is processed with pixel-wise masking operations, most of its processing time are used for repeatedly applying the same operation to each pixel [1]. These kinds of repeated operations are one of the most suitable ones to be accelerated by parallel processing techniques. Thus, we have plenty of parallel-processing methods for these image processing operations.

In various fields including automatic control of mechanical robots, there are lots of hardware devices based on these kinds of simple digital image processing techniques [2]. These devices generally have plenty of hardware specifications, while their purposes are usually restricted to the pre-defined areas. In this case, it is most important to find the most optimal way of implementing the specified functionality with the restricted hardware devices.

In this paper, we aim to apply image-processing techniques for the motion control of a two-legs walking robot, as shown in Figure 1. As its main computing power, this robot is equipped with an embedded board, which is actually hard to achieve image

* corresponding author: Nakhoon Baek, oceancru@gmail.com

† This research is supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program (Immersive Game Contents CT Co-Research Center).

ISSN: 2287-1233 ASTL

processing operations in real-time. Thus, we accelerated the given hardware with parallelized special hardware: an FPGA chip.

We analyzed the required image processing operations, and designed overall architecture. We use HDL (hardware description language) [3] to design our special-purpose FPGA chips. This FPGA implementation of parallel image processing algorithms enables us to achieve the real-time motion control of the walking robot, which actually removes the pre-defined shaped objects. Our implementation result shows the way of accelerating low-tier embedded systems with FPGA-based parallel processing.

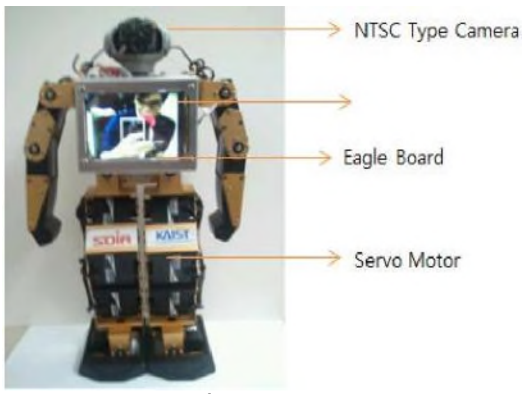

Fig. 1. Our final implementation:

Section 2 represents our design of the parallelized image processing system, and also our FPGA implementation. Implementation results are in Section 3, and finally conclusions are followed in Section 4.

2 Design and Implementation

The target embedded device is an ARM-based development board, with *uCLinux* Operating System. More specifically, it is equipped with the Eagle-series main CPU, 64MB SDRAM, 64MB NAND Flash memory, and an FPGA connection with 16bit address and data buses [4].

Our overall system architecture is represented in Figure 2. The simulator part has actually three different versions of image processing algorithms: a PC-based software implementation, another target embedded device-based software implementation, and our FPGA-based parallelized implementation. We can execute a specific image processing algorithm for all three implementation, and compare the results each other. The two-legs walking robot act as an I/O device. Its internal camera gets the digital images and transfers those images to the embedded board. The robot motion control algorithm works on this embedded board, with the computing support of the FPGA chip. Finally, our robot detects some geometric objects as obstacles in its walking path.

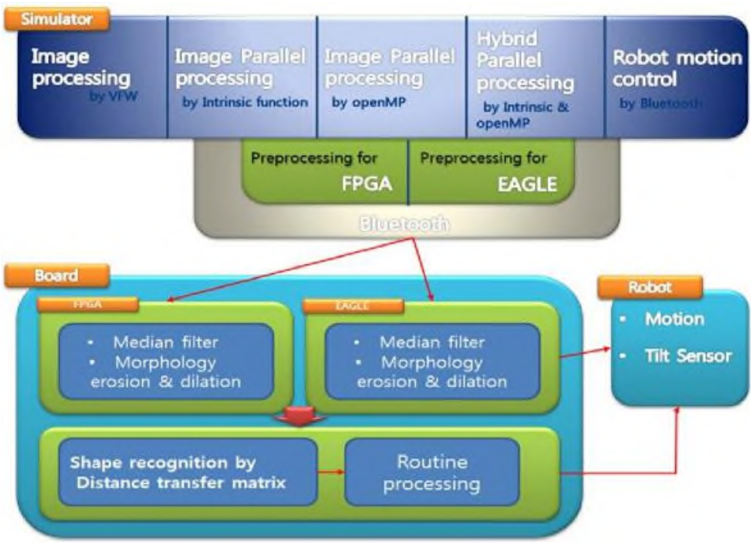
Simulator
Image processing by VFW
Image Parallel processing by Intrinsic function
Image Parallel processing by openMP
Hybrid Parallel processing by Intrinsic & openMP
Robot motion control by Bluetooth
Preprocessing for FPGA
Preprocessing for EAGLE
Bluetooth
Board
FPGA
EAGLE
Median filter
Morphology erosion & dilation
Median filter
Morphology erosion & dilation
Shape recognition by Distance transfer matrix
Routine processing
Robot
Motion
Tilt Sensor

Fig. 2. Overall architecture of our system.

The camera-captured images from the robot are 320x240 resolution with RGB colors. At the first stage of our algorithm, we apply a median filter to remove potential noises. And then, a morphology filter is used to additionally remove noises in the image.

At the next stage, the distance-transformation matrices are used to isolate the candidate regions for the geometric objects [5]. It also checks their RGB colors with color correction techniques. Additional location information are used to identify those geometric objects in the input images.

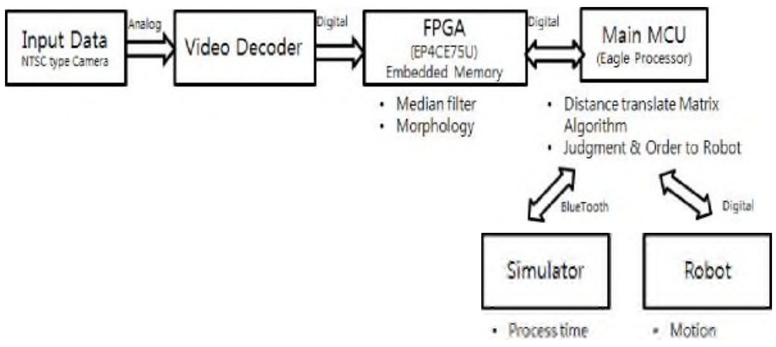

Fig. 3. Information flow for the embedded board and our FPGA chip

Figure 3 shows the information flow on the embedded board attached to the walking robot. From the NTSC-type video camera, the images are captured and converted into digital images. Then, our FPGA chip performs a median filter and another morphology filter on those images. The main CPU processes the distance-transformation matrix algorithm in its sequentially implemented execution. Then, the motion control algorithms for the walking robot are executed to make required motions of the robot.

3 Implementation Results

To make FPGA implementations, we carefully selected a set of mask-based image processing algorithms: median filter, morphology filter, mean filter, sharpening filter, and edge detection filter. For comparison purpose, we actually implemented all these algorithms on three different forms: a PC-based software implementation, an embedded board-based software implementation, and an FPGA implementation.

In the case of the embedded board-based software implementation, the target embedded board has relatively low computing power, and thus, shows much slow execution speeds. For example, the median filter algorithm works in 1,070 msec on the target embedded board, while the PC-version works in 44 msec, which is about 25 times faster. Our FPGA implementation fully utilizes the speed-up of parallelized implementation, and shows much faster execution speeds of 97 msec.

Table 1. Experimental results for geometric objects.

	target geometric object		
	circles	triangles	squares
Software implementation (on embedded boards)	25 success / 30 trials	24 success / 30 trials	27 success / 30 trials
our FPGA-based parallelized implementation (FPGA chips)	28 success / 30 trials	25 success / 30 trials	26 success / 30 trials

For the practical experiments, we have checked the success and failure of the 2-legs walking robot, as shown in Table 1. We provide three different types of objects: circles, triangles, and squares. We tested 30 times for each object, to both of embedded-board based software implementation and our FPGA-based parallelized implementation. As shown in Table 1, our FPGA implementation outperforms the software implementation for circles and triangles, while shows similar score for squares. Conclusively, our FPGA-based parallelized implementation outperforms the software-based implementation, with remarkable speed-ups.

4 Conclusion

In this paper, we realized specific image processing algorithms into their parallelized implementations, as FPGA chips, for relatively low-powered embedded devices. Our final analysis shows that these FPGA implementations mark final scores equal to or better than typical embedded device implementations. In contrast, from the processing time point of view, it works much faster than embedded device implementations, and even similar speed to the PC implementations. Our implementation results show that parallelized implementations of image processing algorithms on FPGA chips can supplement the low computing power of typical embedded devices.

Acknowledgments. This research is supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program (Immersive Game Contents CT Co-Research Center).

References

1. Petrou, M., Petrou, C.: Image Processing: The Fundamentals, Wiley, (2010)
2. Blake, G., Deslinski, R., Mudge, T.: A survey of multicore processors: A review of their common attributes, IEEE Signal Processing Magazine, vol. 26, no. 6, pp. 26–37 (2009)
3. Thomas, D., Moorby, P.: The Verilog Hardware Description Language, Springer, (2008)
4. ADC, Inc, Eagle board Architecture diagram, application: http://www.adc.co.kr/product/
5. Grevera, G.: Distance Transform Algorithms and Their Implementation and Evaluation, Deformable Models, pp. 33-60, Springer (2007)