A FLEXIBLE SMART CAMERA SYSTEM BASED ON A PARTIALLY RECONFIGURABLE DYNAMIC FPGA-SOC

Daniel Ziener, Stefan Wildermann, Andreas Oetken, Andreas Weichslgartner, and Jürgen Teich

Hardware/Software Co-Design, Department of Computer Science
University of Erlangen-Nuremberg, Germany
email: {*daniel.ziener, stefan.wildermann, andreas.weichslgartner, juergen.teich*}*@cs.fau.de*

ABSTRACT

In this paper, we present an FPGA-based smart camera system with support for dynamic run-time reconfiguration. The underlying architecture consists of a static SoC which can be extended by dynamic modules. These modules are responsible for the stream-based image processing and can be loaded and unloaded at run-time. Furthermore, even the position of these modules in the processing chain can be exchanged.

1. INTRODUCTION

FPGA-based embedded systems are of increasing importance especially in the signal and image processing domain. For instance, intelligent embedded systems for image processing, such as smart cameras, rely on FPGA-based architectures [1]. Many FPGA devices support dynamic partial reconfiguration. With this advantage, we can envisage new designs with new and improved possibilities and properties, like an adaptive design, which can adapt itself to a new operation environment. Moreover, the ability to replace modules at run-time decreases the area overhead which further leads to smaller FPGAs. This reduces also the overall power consumption. Partially reconfigurable hardware provides also advantages in the design flow. With well defined interfaces, reconfigurable modules can be developed and implemented without the knowledge of the rest of the system. Even if the system is deployed, new modules can be developed and integrated into the system. Having a library with many reconfigurable modules, the system is able to adapt new environments or to meet new requirements which make such a system very interesting in the research field of self-organizing computing, for example, for smart cameras.

We will demonstrate such a smart camera system based on a partially reconfigurable SoC, described in details in [2], at the workshop. The static system provides a CPU, the SoC infrastructure and the interfaces for the camera system. Most of the image processing algorithms, e.g., filtering, color transformation and detection, or marker modules are implemented as partially reconfigurable modules which can be dynamically loaded and unloaded at run-time.

2. ARCHITECTURE

The system is implemented on the Xilinx Virtex-II Pro XUP board and consists of an embedded CPU sub-system including the external DDR-memory and the reconfigurable part

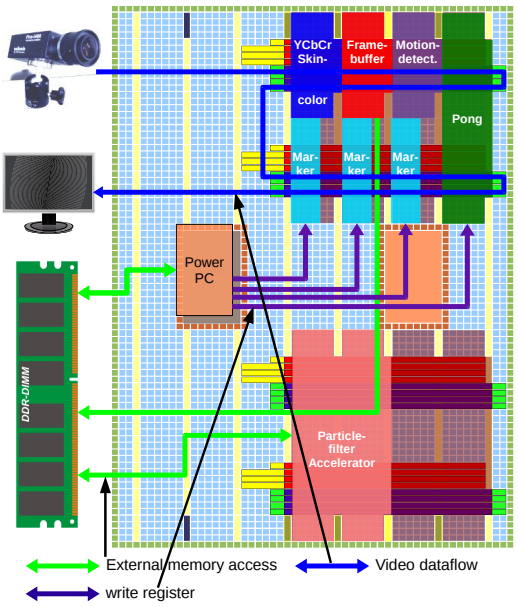


Fig. 1. System overview of the heterogeneous FPGA-based smart camera SoC platform consisting of CPU sub-system and reconfigurable area. Reconfigurable modules can vary in size and be freely placed, allowing a very good exploitation of the FPGA space.

(see Figure 1). In the following, these components and the communication interfaces between them are presented.

2.1. Embedded CPU Sub-system

The main purpose of the software part on the embedded CPU is to control and manage the overall system. It contains high-performance peripherals, interfaces, and other IP cores. These are, e.g., a memory controller to provide access to an external RAM, a serial port interface for user commands, and a module for accessing the integrated reconfiguration interface of the FPGA. All components of the embedded CPU sub-system are connected by the main on-chip system bus, the *processor local bus* (PLB).

2.2. Reconfigurable Area

The FPGA area is divided into a static and a dynamic part (see Fig. 1). The two dark-red areas on the right top and

bottom compose the dynamic part of the system. Reconfiguration is only possible in the dynamic part which contains a reconfigurable on-chip bus (*ReCoBus*) and *I/O bar* communication primitives to provide a communication infrastructure for dynamically loaded hardware modules. Both communication primitives are provided by the framework ReCoBus-Builder [3].

The I/O bar is used for streaming the data and establishing point-to-point connections between hardware modules which are located in the dynamic area. The basic principle is that each module is able to read the incoming data, modify it if necessary, and pass it further to the next module. Details are described in [4]. In the smart camera platform, the I/O bar is used to stream video data between the various reconfigurable processing modules. The modules can modify the video stream or generate additional output signals.

The ReCoBus [4] (RCB) is an on-chip bus that is suitable for dynamically integrating hardware modules into an FPGA by partial reconfiguration. To allow communication between the embedded CPU sub-system and the reconfigurable part, a PLB/RCB Bridge translates the ReCoBus protocol to the PLB protocol and vice versa. Using the ReCoBus and the bridge, the modules can be accessed from the CPU, e.g., to configure the module with memory-mapped registers. Furthermore, the modules have also direct access to the external memory (DMA). To allow high-speed data transfers between hardware masters and the memory controller, the bridge uses the *native port interface* of the memory controller.

3. RECONFIGURABLE MODULES

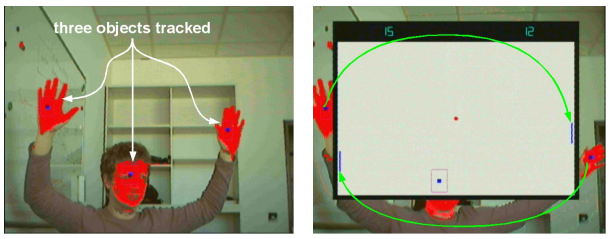
We implement several reconfigurable modules to tackle a wide spectrum of applications for our smart camera platform. In this section we present some of these modules.

The *skin color detection* is implemented as a hardware slave module that reads the color values from the I/O bar and marks them as *skin* or *non-skin* by comparing them with a color template. We have implemented modules for RGB and YCbCr color spaces. The classification is written as an additional signal onto the I/O bar together with the unmodified video stream.

The *filter module* is a sliding-window image processing filter. The current implementation supports a 3x3 filter matrix. To access different image lines, the module stores two lines in a BRAM-FIFO. Due to the flexible coefficient matrix which parameters are stored in CPU accessible registers, a module can be configured for different filter functions. For example, a sobel filter which can be used for edge detection.

The *framebuffer* hardware master module is implemented to store the current input image. This is done by double buffering the images in the on-chip memory via the ReCoBus using the NPI interface. We use 32 Bit for storing one pixel, with 24 Bit for the input RGB values and the remaining 8 Bit free for classification results.

The *particle filtering framework* is partitioned into software and hardware part. The software part performs the sampling and applies the motion model. The hardware part is used as a co-processor to perform the evaluation steps.



(a) The particle filter tracking three objects.

(b) The object tracker used to play a pong game.

Fig. 2. The smart camera tracks three image regions (a person's head and hands). The tracked hand positions are directly used to control the paddles of the video game.

The *motion detection module* compares the pixel values of two subsequent images to detection motion. Like the skin color detection module, the result (motion/no motion) is written as an additional signal onto the I/O bar.

The *pixel marker module* colors classified pixel or regions with a specified color. The classification of the pixel is signaled to the marker module with additional I/O bar signals. The color can be configured by a register interface.

An embedded design for tracking human motion is implemented as an example application to show the flexibility of the proposed platform. The idea is to detect and track skin-colored image regions, which is done by applying particle filtering. The current implementation makes it possible to track up to 3 image regions. One marker module is used per region tracker. A simple *tennis game* is implemented on top of this application, which can be directly controlled by the hands of a person, using the results of the tracker (see Fig. 2).

4. CONCLUSION

This paper presents a flexible FPGA-based smart camera platform which offers a reconfigurable area for dynamic hardware modules which can be loaded during run-time. One benefit of such an approach is that customizable and self-adaptive flexibility can easily be implemented.

5. REFERENCES

[1] J. Schlessman, C.-Y. Chen, W. Wolf, B. Ozer, K. Fujino, and K. Itoh, "Hardware/software co-design of an FPGA-based embedded tracking system," in *Proceedings of CVPRW*, 2006, p. 123.

[2] A. Oetken, S. Wildermann, J. Teich, and D. Koch, "A Bus-based SoC Architecture for Flexible Module Placement on Reconfigurable FPGAs," in *Proceedings of the International Conference on Field Programmable Logic and Applications (FPL)*, Milan, Italy, Aug. 2010, pp. 234–239.

[3] "Recobus-homepage," http://www.recobus.de/.

[4] D. Koch, C. Beckhoff, and J. Teich, "Recobus-builder a novel tool and technique to build statically and dynamically reconfigurable systems for fpgas," in *Proc. of FPL 2008*, Sept. 2008, pp. 119–124.