Eidgenössische Technische Hochschule
Swiss Federal Institute of Technology
Zürich

Electronics Laboratory

Reconfigurable Processor Board

Rolf Enzler and Didier Cottet

Rolf Enzler
ETH Zurich
Electronics Laboratory
Gloriastrasse 35
CH-8092 Zurich

Phone: +41-1-632 50 68
Fax: +41-1-632 12 10
E-mail: enzler@ife.ee.ethz.ch

IEEE Member No: 40075619

Didier Cottet
ETH Zurich
Electronics Laboratory
Gloriastrasse 35
CH-8092 Zurich

Phone: +41-1-632 51 27
Fax: +41-1-632 12 10
E-mail: enco@ife.ee.ethz.ch

Reconfigurable Processor Board

Contents

Abstract

The concept of dynamic reconfigurability combines advantages of hardware and software. The goal is to make use of the structural advantages of hardware without losing the flexibility of software. In this paper we present a reconfigurable processor board which is based on the concept of dynamic reconfigurability. We emphasize on the dynamic reconfigurability of the system and the multi-chip module (MCM) that is used as processing element.

To optimize the dynamic reconfigurability of the system, the architecture of the processor board contains dedicated hardware structures. Thus, reconfiguration overhead, that limits performance enourmously, has been reduced to a minimum. As processing element the so called FPGA-MCM is used. This MCM combines a field programmable gate array (FPGA) with plenty of memory capacity, which makes it suitable for use in reconfigurable systems.

1 Introduction

With the technological progress of programmable devices and programming methodologies, reconfigurable devices become more and more interesting and important. The concept of dynamic reconfigurability combines advantages of hardware and software with the intention of developing fast but flexible circuits.

An example is the use of reconfigurable devices as co-processors for digital signal processors (DSPs). Although DSPs are optimized for applications in digital signal processing, many tasks cannot be implemented efficiently on a DSP architecture. The reason is the sequential order in which DSPs execute their programs. DSPs cannot make use of parallel structures (e.g. of a fast fourier transform). Often, parallel data paths can be implemented

more efficiently on a programmable device. Therefore, reconfigurable co-processors will become an important alternative to DSPs.

The processor board that is described in this paper is a prototyping system to gain experience in using reconfigurable devices as co-processors. A main feature of the design is to optimize the system for dynamic reconfiguration and thus to minimize reconfiguration overhead. To have an idea which architecture is ideal for a reconfigurable system, several algorithms were evaluated. A major characteristic of a reconfigurable system is that it cannot be optimized for a single algorithm but must be suitable for a class of different algorithms.

The second main feature of the design is the use of the FPGA-MCM as processing element and thus to gain experience with multi-chip modules. In order to improve yield in production, the FPGA-MCM has a redundant memory structure. The support of the memory redundancy was a major requirement of the board design.

2 FPGA-MCM

2.1 Architecture

The FPGA-MCM is a multi-chip module (MCM) designed for use in form factor constrained solutions [1]. It consists of a field programmable gate array (FPGA) XC4013 by Xilinx and three MByte SRAM (figure 1). The FPGA is dynamically reconfigurable, i.e. a new hardware circuit can be downloaded at any time during operation. The memory is organized into two blocks, which can be addressed independently by the FPGA. Decoupling capacities are integrated on the MCM for the FPGA and the six SRAM dies. Table 1 summarizes the technological parameters.

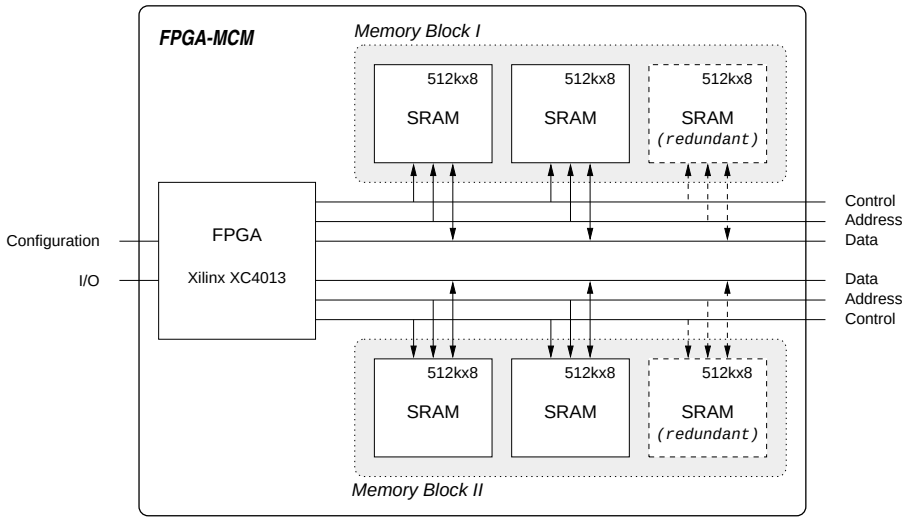

Figure 1: Architecture of the FPGA-MCM

2.2 Redundant Memory Structure

To improve yield in MCM production, the memory has a redundant structure [2]. One of the three memory devices in each block is redundant, i.e. the MCM can still operate when one die per block is damaged. Thus, the usable memory capacity decreases to two MByte. However, the support of the redundant memory structure is no prerequisite for using the MCM. Fully functional MCMs can be used in environments that do not support the memory redundancy, but make use of the whole capacity of three MByte.

2.3 Comparison with a PCB Solution

Compared with a conventional PCB system the MCM has advantages concerning size, weight, speed and power dissipation. As a result, the MCM is likely to be used in critical environments.

The locality of FPGA and memory devices permits fast access of local data. Thus, efficiency is improved for computational intensive algorithms.

Xilinx FPGA XC4013		
Parameter	*Value*	*Note*
Process	CMOS SRAM	
Number of gates	13.000	
CLB matrix	24×24	CLB: configurable logic block
Number of CLBs	576	
Number of IOBs	192	IOB: input/output block
Number of flip-flops	1.536	2 flip-flops per CLB,
		2 flip-flops per IOB
Max. RAM bits	18.432	$2 \times 16 \times 1$ bit each CLB

Samsung SRAM KM684002		
Parameter	*Value*	*Note*
Type	SRAM	Static RAM
Organisation	$512k \times 8$	
Access time	$17ns$	measured
Process	CMOS	TTL-compatible in/out

FPGA-MCM		
Parameter	*Value*	*Note*
Process	MCM-L	
Laminate	FR4	6 layers
Package	ball grid array	299 balls, $1.27mm$ pitch
Size	$45 \times 45mm^2$	
Power dissipation	< 10 watt	

Table 1: Technological parameters of the FPGA-MCM

3 Processor Board

3.1 Concept

The processor board was designed for use as a reconfigurable system. A key feature is the implementation of a fast and independent reconfiguration of the processing element. Another key feature is to support the memory redundancy of the FPGA-MCM.

3.2 Architecture

The processor board is designed for the 5Volt-, 33MHz-, 32Bit-PCI-Standard (figure 2). The PCI9060 interface device by PLX handles the PCI protocol. On the local side a 32 bit data bus is supported, that is clocked at 25 MHz. The calculated maximum throughput in burst mode is 50 MB/s.

As communication controller the FLEX EPF8820 by Altera is used. This programmable device controls all data transfers on the processor board. It contains an arbiter which controls the access requests for the local bus, it is responsible for reconfiguring the FPGA and supports the FPGA-MCM's memory redundancy.

The MCM-SRAMs can be accessed in a very flexible way. The memory is organized into two memory blocks which contain two usable memory devices each (one device is redundant). Because of the flexible addressing scheme of the four memory devices, data words of 8, 16, 24 or 32 bit can be processed.

For image processing applications, the processor board contains a 16 bit video interface. The video signals are differential and RS-422 compatible.

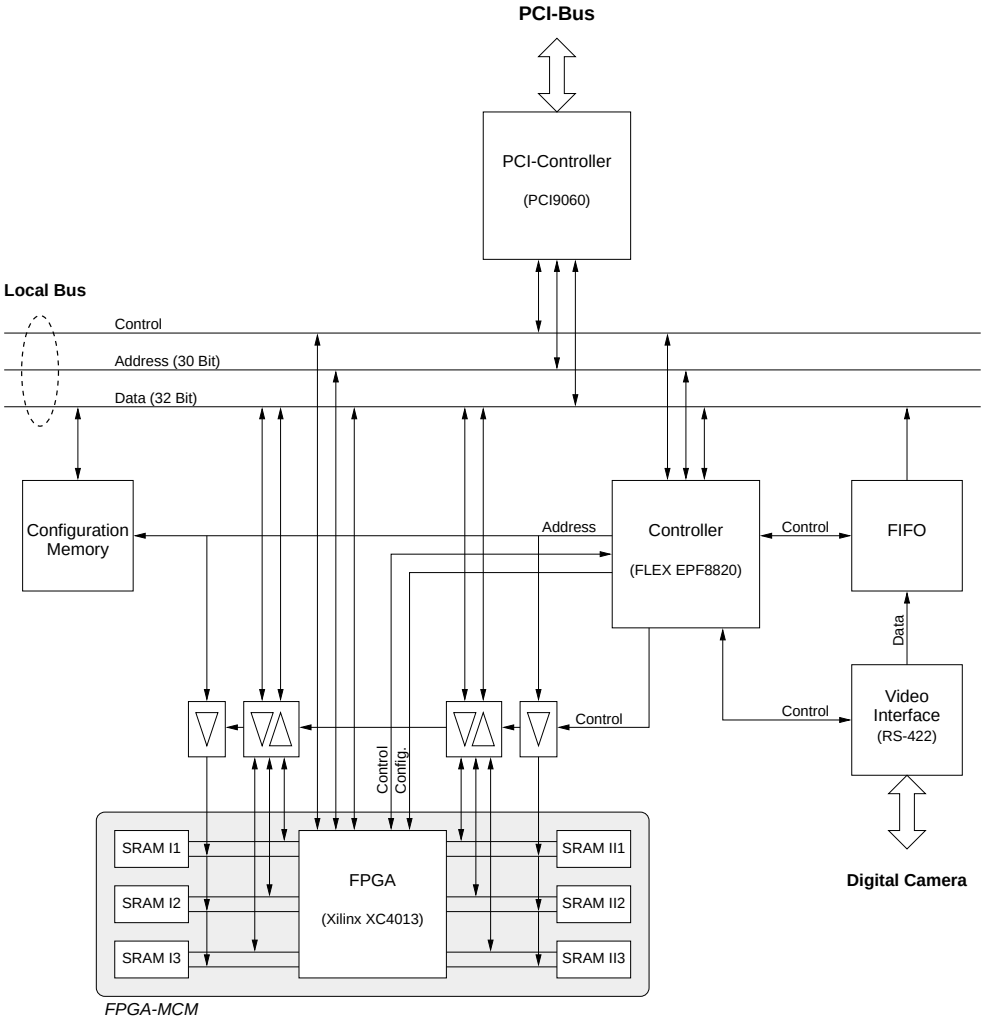

Figure 2: Architecture of the processor board

3.3 Dynamic Reconfiguration

The processor board contains a dedicated configuration memory, which can store up to 16 different FPGA configuration files. This concept has two major advantages: On one hand the FPGA can be reconfigured at the highest possible speed, on the other hand reconfiguration requires no resources of the host computer, since it runs completely autonomously. The measured reconfiguration time is 30 ms.

FPGA reconfiguration is controlled by the FLEX device and can be initiated locally or by the host computer. During reconfiguration, data communication on the local bus is guaranteed.

3.4 Memory Redundancy of the FPGA-MCM

To support the redundant memory structure, additional circuitry is needed. The FLEX is responsible for controlling the transceiver devices and therefore defines the data paths between local bus and memory devices (figure 2). Information about the working memory devices is sent from the host to the processor board during initialization. Routing of the data path is controlled completely by hardware, therefore software has not to care about memory redundancy. From the software's view, the memory consists of two memory blocks with two logic memory devices each.

4 Algorithms

In order to define the final architecture of the processor board, several algorithms were evaluated, which are explained in this section. Conclusions are given in section 5.

4.1 Traffic Sign Recognition

The algorithm for traffic sign recognition was developped at the Paul Scherrer Institute [3]. Goal is to locate traffic signs within images. Feature matching is based on the search of predefined templates which constitute parts of the objects to be detected.

At our institute an image processing system called Smart Camera was implemented [4]. Low level image processing (Sobel-filtering) is done close to the camera sensors, high level image processing (feature and template detection) on the SHARC signal processor by Analog Devices. The performance on the Smart Camera system for images of 256×256 pixel is 1.5 seconds.

Table 2 lists the estimated resource requirements for a slightly simplified implementation of the algorithm. To adapt the algorithm to the FPGA-MCM architecture, the order of the templates was changed and their size slightly reduced.

Listed are only the needed resources related to the functionality of the algorithm. Additional resources have to be considered for the SRAM controller and the process flow control. When the FPGA is reconfigured for every algorithmic step, no lack of resources arises.

To decrease the perfomance loss caused by reconfiguring the FPGA, process flow is optimized so that four images are processed at once. This means that in each configuration step four images are processed before the FPGA is reconfigured. Thus, configuration overhead is shared by four images, which increases performance accordingly. The estimated throughput for images of 256×256 pixel increases in that case to ten images a second.

Step	Algorithm	Resources
1	Get four images (FIFO)	512kB SRAM (FIFO)
2	Sobel-filtering	375 kB SRAM (FIFO) 512 kB SRAM (LUT) 36 CLBs for 3×3 environment (8 bit) 4 CLBs each arithmetic step ($\sim \times 10$) 8 CLBs each pipeline register ($\sim \times 4$)
3	Feature matching	375 kB SRAM (FIFO) 120 CLBs for feature environment 4 CLBs each pipeline register ($\sim \times 12$)
4	Template matching	1..2 kB SRAM (output register) 32 CLBs for offset table (SRAM) 50 CLBs for control logic
	Total	1.5 MB SRAM (FIFO) 0.5 MB SRAM (LUT) 250-300 CLBs

Table 2: Hardware resources for traffic sign recognition

4.2 Workspace Monitoring System

To provide advanced robotic systems with the capability to share their workspace with humans, they must be equipped with an adequate security system. The Workspace Monitoring System monitors an invisible but clearly defined, dynamically adjustable safety zone around the robot for intruding objects [5][6]. If an object is detected in the separation surface, the robot is stopped to avoid a collision.

Table 3 lists the algorithmic steps as well as the required resources for a FPGA implementation [7]. The result shows that the FPGA-MCM cannot meet the requirements unless the algorithms are simplified (smaller images, reduction of computational precision). To simplify the algorithm accordingly would require a comprehensive test to guarantee stability. Therefore, a dedicated architecture is more suitable than using the FPGA-MCM.

4.3 Gauss and Canny Operators

Gauss and Canny operators are often used in image processing for edge and line detection. With a Gaussian filter for eliminating noise before the edge detection and a non-maximum-suppression afterwards very good results can be achieved. The environments for these operators have to be chosen quite large (11×11, 13×13) to gain advantage of the Gaussian filter. Moreover, the coefficients must not be rounded or scaled arbitrarily because of quantization noise and variance of the Gaussian function.

Since the algorithms can be implemented in several different ways, table 4 contains only the needed resources for operators and arithmetic functions. The estimations show that large environments cannot be fitted into the FPGA, because it lacks of resources. Thus, the use of the FPGA-MCM is only practical when small environments are required.

Step	Algorithm	Resources
1	Produce left and right image	One FIFO per picture
2	Transformation of the left image	8000 gates 512 kB SRAM 5 MB VRAM (LUT)
3	Lowpass	2000 gates $2k \times 8$ FIFO
4	Sobel-direction	3500 gates 256 kB SRAM (LUT) $2k \times 8$ FIFO
5	Correlation	2000 gates $2k \times 8$ FIFO
6	Texture analysis	2000 gates $2k \times 8$ FIFO
7	Segmentation	1500 gates ($2k \times 8$ FIFO)
8	Transition-coding	2200 gates 32 kB SRAM
	Total	19000 gates 800 kB SRAM 5 MB VRAM $10k \times 8$ FIFO

Table 3: Hardware resources for the Workspace Monitoring System

Algorithm	Resources		
Operator Register	3×3:	36	CLBs
	5×5:	100	CLBs
	9×9:	324	CLBs
	11×11:	484	CLBs
Arithmetic Operations (8 Bit)	Adder:	4	CLBs
	Multiplier:	32	CLBs

Table 4: Hardware resources for Gauss and Canny Operators

4.4 DNA Sequence Comparison

DNA sequences consist of only four different components, the nucleotides. Very long DNA sequences have to be scanned in order to detect similar segments within them.

Because of their wide word width, microprocessors and digital signal processors work inefficiently on these computations. A FPGA however can process computations of two bit word width very efficiently. Thus, the FPGA-MCM is suitable for this application. Moreover, the possibility of implementing an architecture with FPGA-MCMs in parallel increases performance nearly ideally. A drawback is that a multi FPGA-MCM architecture increases memory capacity to an amount where the memory is not anymore used efficiently. That implies too high costs compared to the gained advantages. As a result, the use of the FPGA-MCM is only suitable for small systems but not for high performance systems with several parallel FPGA-MCMs.

An example of an implemented DNA sequence comparison system can be found in [8].

5 Conclusions

5.1 Dynamic Reconfiguration

The use of a dedicated configuration memory for fast reconfiguration of the FPGA proved successfully. The FPGA can be reconfigured within 30 ms. However, a general problem of reconfigurable devices has been identified: Reconfiguration time cannot be ignored, because it is quite long with 30 ms.

A possibility to decrease the reconfiguration overhead is presented with the traffic sign recognition algorithm (section 4.1). In every configuration step four images are processed (instead of only one) before reprogramming the FPGA. Thus the configuration overhead is shared among four images and the performance loss is decreased accordingly (figure 3).

New generation devices of Xilinx (XC6200 family) soften this problem. Reconfiguration time is reduced significantly with a new configuration technique and the possibility to reconfigure only parts of the FPGA.

5.2 Algorithms

As the evaluation of different algorithms in section 4 shows, the FPGA-MCM is not suitable for all kinds of algorithms. Main problems arise with the limited resources of the FPGA and the inefficient usage of the given memory capacity.

In general it can be stated that an algorithm should have the following characteristics to be ideally processed in several reconfiguration steps: It should be partitionable the way that all parts have more or less the same execution time, the partitions should ideally make use of the FPGA's resources and execution time should be much longer than reconfiguration time. The more this features can be met, the less the reconfiguration overhead effects performance.

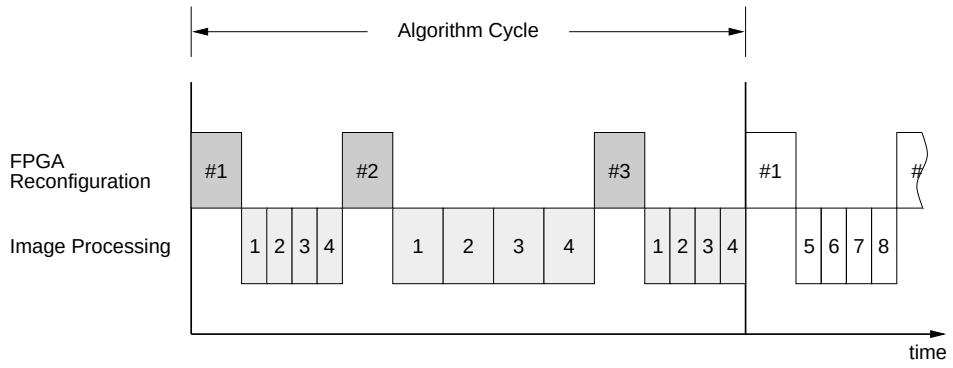

Figure 3: Configuration and process flow

5.3 Performance

To have an idea of the resulting performance, an estimation was done for the traffic sign recognition algorithm (section 4.1). With partitioning the algorithm into three steps and the above explained effort to reduce configuration overhead, a speed-up of 15 was estimated compared to the Smart Camera system. As a result, the concept of dynamic reconfiguration can be used successfully and the goal to combine advantages of hardware and software is met.

5.4 Memory Redundancy of the FPGA-MCM

The concept of a redundant memory structure proved a success from the functional point of view. The price that has to be paid therefore is an additional external circuitry. A possibility for further projects is to integrate this circuitry on the MCM. As a drawback the MCM cannot be used anymore in an environment that does not support the memory redundancy. A third of the available memory capacity could not be used, which is not satisfying from the point of costs. To decide on the direction to go in future projects, further experience with the production yield of MCMs can give evidence.

References

[1] J.-P. Wyss: *FPMCM (Field Programmable MCM) - Technical Brief.* Electronics Laboratory, ETH Zurich, August 1996.

[2] J.-P. Wyss: *An MCM for Neural Network and Similar Applications. Fault Tolerance and Reconfiguration Structures in Multi Chip Modules.* Electronics Laboratory, ETH Zurich.

[3] P. Seitz, G.K. Lang: *Using local orientation and hierarchical spatial feature matching for the robust recognition of objects.* Visual Communications and Image Processing '91: Image Processing, Volume 1606, 1991.

[4] R. Mudra, M. Thaler, G. Tröster: *Smart Camera System.* Electronics Laboratory, ETH Zurich.

[5] M. Rechsteiner, B. Schneuwly, G. Tröster: *Dynamic Workspace Monitoring.* Electronics Laboratory, ETH Zurich, September 1994.

[6] M. Rechsteiner, M. Thaler, G. Tröster: *Implementation Aspects of a Real Time Workspace Monitoring System.* Electronics Laboratory, ETH Zurich, July 1995.

[7] M. Rechsteiner: *Real Time Inverse Stereo System for Surveillance of Dynamic Safety Envelopes.* Dissertation ETH No. 12004, ETH Zurich, 1996.

[8] D.T. Hoang: *Searching Genetic Databases on Splash 2.* Proceedings 1993 IEEE Workshop on Field-Progammable Custom Computing Machines, pp. 185-192, IEEE, 1993.