

# Distributed Processor Network on a Single Chip

Wjatscheslaw Galjan, Abdulrahman Hanoun, Wolfgang H. Krautschneider, and Friedrich Mayer-Lindenberg

**Abstract**—The Parallel-CPU-On-Chip project aims at implementing a distributed computing architectures with scalable performance. As a proof of concept, we integrated two 16-bit sequential microprocessors called CPU-III on a chip. The CPU-III supports many vital features for controlling including multi-threading, two-stage execution pipelining and an enhanced coprocessor interface which also supports multi-threading and coprocessor pipelining.

**Keywords**— distributed digital processing systems, multi-threading, scaleable digital systems, distributed processors

## I. INTRODUCTION

In multi sensor applications, there is much need for scalable computation power and controlling a high number of acquisition devices. Using dedicated application-specific integrated circuits (ASIC) may be less attractive, because any change of the application causes a redesign of the entire microchip. Using separate sequential controllers and Digital Signal Coprocessors would generalize the control and data paths so that the design becomes flexible and can be adjusted to a variety of applications.

Scalability is another matter of interest; embedded controllers have to support a degree of scalability which allows systems to grow beyond current dimensions. Increasing the system integration level by scaled-down submicron CMOS technology should be accompanied by scalability of the entire system.

Manuscript received September 29, 2006.

Wjatscheslaw Galjan and Wolfgang H. Krautschneider are with the Institute of Nanoelectronics, Hamburg University of Technology, Eissendorfer Str. 38, D-21073 Hamburg, Germany (Phone: +49-40-48278-3991; Fax: +49-40-48278-2877; E-mail: w.galjan@tu-harburg.de; krautschneider@tu-harburg.de).

Abdulrahman Hanoun and Friedrich Mayer-Lindenberg are with the Institute of Computer Technology, Hamburg University of Technology, Schwarzenbergstr. 95, D-21073 Hamburg, Germany (E-mail: a.hanoun@tu-harburg.de; mayer-lindenberg@tu-harburg.de).

The idea of this project is to realize a scaleable and easily extensible structure for low power and low cost multi sensor applications. To evaluate the scalable network-on-chip concept we have integrated two identical CPU-III cores [1] on a single chip. Both CPUs are able to execute completely different tasks in parallel and can communicate with each other and with an external coprocessor. To keep the system size small, the architecture of the CPU does not have DSP capabilities such as executing MAC operations in one cycle. For this purpose, the coprocessor with an appropriate interface has been taken into account. One or several coprocessors can be used to meet the required computation power. The CPU-III supports zero-overhead context switching for serving four coprocessor contexts sequentially.

The proposed application is depicted in Fig. 1. Several sensors are connected to one CPU, which is able to prepare and to pre-compute the data for the fast computation by the coprocessor. An advantage of this setup is the possibility to scale the whole system and to reuse the software for all CPUs. Optimizing the coprocessor according to the required functions would allow minimizing the power consumption of the whole system.

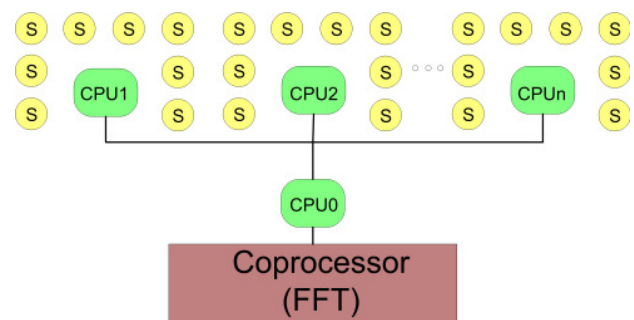


Figure 1 Multi sensor application

This paper is organized as follows: Section 2 introduces the CPU-III and lists its features, section 3 gives a brief description of the Distributed Processor Chip (DPC), section 4 describes prototyping of the chip,

section 5 addresses the verification environment, and finally section 6 gives a conclusion.

## II. CPU-III

CPU cores are an essential component of any System-on-Chip. Embedded CPU cores perform sequential control and sometimes digital signal processing. The basic idea behind the CPU-III design is to realize a strict and clear separation between control path and a complex computation data path. Sequential control is performed by the 16-bit CPU-III while data path computations take place in stepwise pipelined coprocessors. This clear separation results in a simple yet effective CPU core which consequently results in reducing the overall costs of the SoC. However, the intended use of the CPU for controlling application-specific coprocessors dictates some special features including multithreading coprocessor interfacing, DMA, zero-overhead jump and call, and a simple parallel communication mechanism.

### A. Structure of the CPU-III

CPU-III has a special architecture half way between Harvard and von-Neumann. It has one dual port RAM for both data and instruction. This supports the FPGA environment very well (which the CPU-III is originally made for [5]) because it uses the existing memory resources very efficiently. This is not the only advantage of such architecture; using DP-RAM allows double instruction fetching when data is not to be fetched. Additionally, the call instruction can optionally be executed simultaneously with a computation instruction. Fig. 2 illustrates the CPU-III architecture. Each CPU can support up to four coprocessor threads with no context-change overhead, because of the implemented register banks [6].

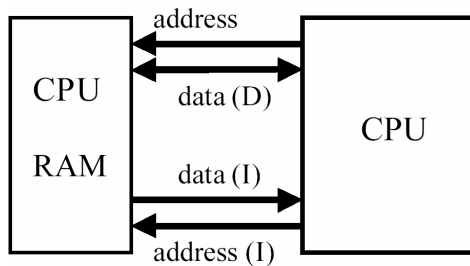


Figure 2 CPU-III Memory Interactions

### B. Coprocessor Interface and Parallel Port

The DMA ability helps in transferring blocks of data through a 16-bit parallel communication bus. IO block transfers can be used to load CPU data and instructions and also to directly transfer coprocessor data.

Coprocessors have their own data memories as in a pure Harvard structure. These memories can be wider than the CPU memories if necessary. The coprocessor interface is specially designed to support stepwise operating of the pipelined coprocessors. Through the coprocessor interface, the CPU controls the pipeline advancing of the coprocessor as well as the coprocessor memory transactions. The CPU acquires the computation results through exchange registers in the coprocessor. Fig. 3 illustrates the special coprocessor/CPU relation [2].

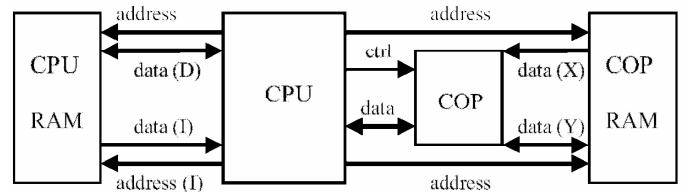


Figure 3 Interaction between a CPU III and the coprocessor

### C. Network capability

Networking can accumulate the necessary processing power from several data-path coprocessors. Therefore, it is one of the priorities for the CPU-III to have an IO bus system which is able to connect several CPU cores and other modules. Several CPU-III cores can form a Network-on-Chip using a Token-Ring communication bus. The 16-bit bus system uses simple handshaking and token passing mechanisms and can connect up to 8 CPU cores and/or other modules. Fig. 4 illustrates an example of a dual CPU core and an SPI RAM controller connected together by the 16-bit communication bus [3].

Point-to-point Handshaking signals address the communicating members, while the arbitration signals manage the token passing and hence controlling bus acquisition and granting.

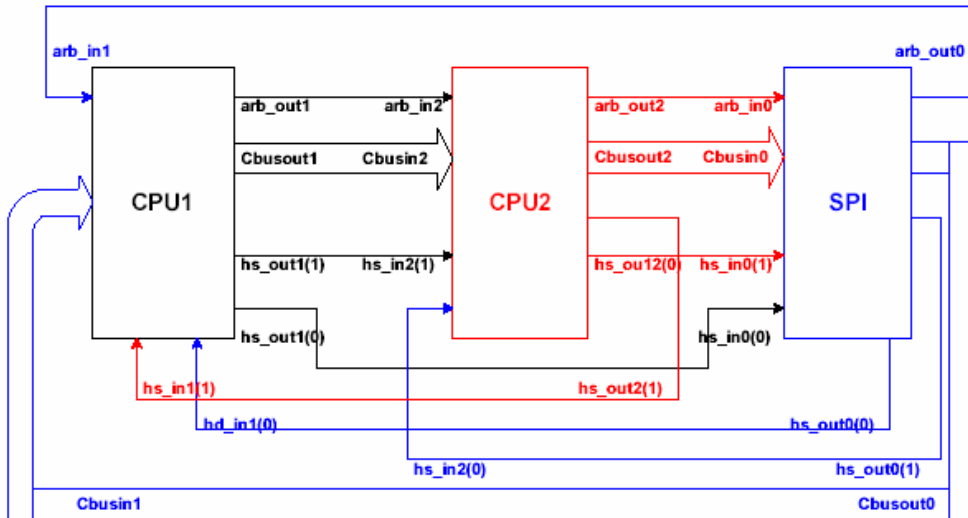
### D. Debug Interface

CPU-III has an internal 32-bit Debug Shift Register (DSR) which can be serially accessed by a JTAG compatible interface.

## III. ARCHITECTURE OF THE DPC

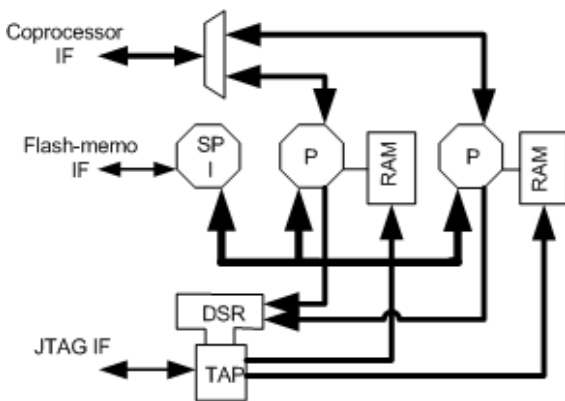
It has to be mentioned that before implementing the DPC on 0.35 $\mu$ m technology, it was programmed and tested on a XC3S1500 Spartan-III<sup>®</sup> FPGA.

The DPC has two identical CPU-III cores sequentially booted from an external serial flash memory using an SPI boot-sequencer. The DPC has a single external coprocessor interface shared by both CPU cores. Up to 4



**Figure 4** Token-ring communication bus system connects two CPU-III cores together with an SPI memory controller

different coprocessors can be connected to this interface. Fig. 5 shows the block diagram of the DPC.



**Figure 5** Block-diagram of Distributed Processor Chip

#### A. Booting of the DPC

After the reset signal goes high and the booting is enabled, one of the CPUs that acts as master device sends a data request to the serial flash module. The first 16-bit word transmitted to the requesting CPU contains the number of 16-bit words to be copied to the internal RAM of the CPU. After the master CPU has finished the booting procedure, it starts to execute the first instruction located at the address 0. The second CPU starts sending a memory request to the same serial flash memory. After booting is finished every CPU can request new data from the serial flash module. This results in the possibility to have two completely different programs running on both CPUs by using only one serial flash on the system board.

#### B. JTAG Interface

A TAP compatible controller is implemented to access the two debugging DSRs in both CPU cores. The JTAG interface is intended for testing purposes. It allows to load into the internal memories, to single step the CPUs, and also to read out the content of the internal double port memory banks. It makes it possible to execute every instruction and to send back the result of the computation for debugging purpose. Therefore, a possible malfunction of the system can be easily detected and corrected for the next design.

#### IV. INTEGRATED PROTOTYPE

A first integrated prototype of the Distributed Processor Chip was manufactured in a 0.35 $\mu$ m 3.3V CMOS technology from Austria-Microsystems. Fig. 6 shows a photograph of the die. The die size is 3700 $\mu$ m x 4200 $\mu$ m. Without the implemented test structures the area of the Distributed Processor Chip would be about 12 mm<sup>2</sup>.

The key-numbers of the implemented design are collected in Table 1. Using double port memories results in higher current consumption and larger area, but allows to double the execution speed of the program, running on the CPUs, and to keep the partitioning of the available memory between data and program universally. The total number of gates used on this implementation, excluding the memory blocks, is about 20,000.

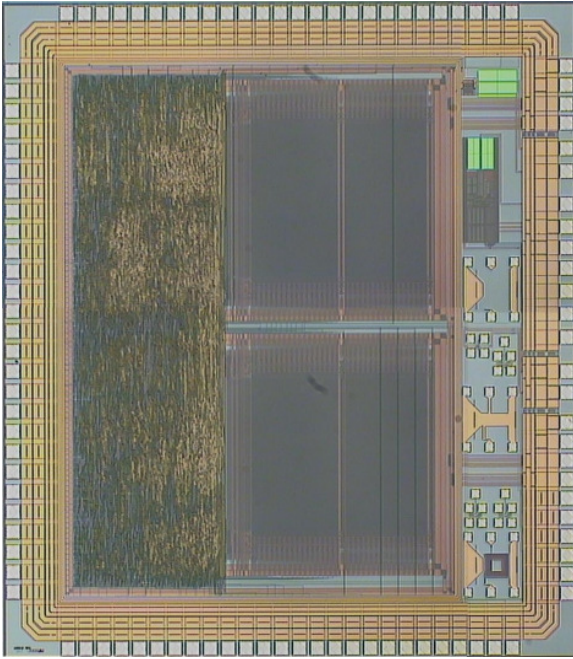


Figure 6 Photograph of the distributed Processors Chip.

Table 1 Key-numbers of the DPC

Components	Power Consumption	Area
Memory	2 x 1.4 mW/MHz	2 x 2.2 mm <sup>2</sup>
CPU1+CPU2 +Mem.Contr	1.35 mW/MHz	3.5 mm <sup>2</sup>
Whole System	4.15 mW/MHz	12 mm <sup>2</sup>

#### V. VERIFICATION BOARD

For verification purposes a test setup has been designed and manufactured at the Institute of Computer Technology of Hamburg University of Technology. The board has a xc3s1500 Spartan-III FPGA [6], USB interface using different standards interfacing chip FT2232C, two DPCs and a standard 16-bit SRAM memory device. The FPGA chip plays the role of a universal programmable interfacing environment for the two on-board DPCs. It also makes it possible to implement and to test different coprocessors. The SRAM module can be directly accessed by the FPGA as general purpose memory and for partially reconfiguring the FPGA with configuration data stored therein.

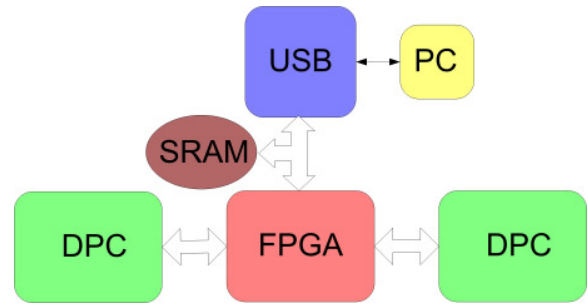


Figure 5 Block description of the test setup

#### VI. CONCLUSIONS

In this work we have presented a design and implementation of a distributed processor chip. The chip has two CPU-III processors. These are simple 16-bit controllers yet have many features to support scalable complex computing. We implemented our chip using 0.35 $\mu$ m CMOS technology and developed a suitable verification environment.

Along with the chip verification activities, coprocessors for different multi sensors applications can be designed.

#### REFERENCES

- [1] F. Mayer-Lindenberg, Dedicated Digital Processors, Wiley&Sons 2004.
- [2] F. Mayer-Lindenberg, Design and application of a scalable embedded systems' architecture with an FPGA based operating infrastructure, Croatia, DSD'2006
- [3] F. Mayer-Lindenberg, A heterogeneous parallel system employing a configurable interconnection network, PDCS Washington 1997.
- [4] [http://www.tu-harburg.de/ti6/ddp/Rn\\_cpu2.pdf](http://www.tu-harburg.de/ti6/ddp/Rn_cpu2.pdf)
- [5] <http://www.tuhh.de/ti6/mitarbeiter/ah/Research/CPU3.html>
- [6] [www.xilinx.com](http://www.xilinx.com)