

# MUCCRA CHIPS: CONFIGURABLE DYNAMICALLY-RECONFIGURABLE PROCESSORS

*H.Amano Y.Hasegawa S.Tsutsumi T.Nakamura T.Nishimura V.Tanbunheng A.Parimala T.Sano M.Kato*

Department of Information & Computer Science, Keio University  
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223-8522 Japan  
E-Mail: [muccra@am.ics.keio.ac.jp](mailto:muccra@am.ics.keio.ac.jp)

## ABSTRACT

Coarse grained dynamically reconfigurable processor arrays (DRPAs) have been received an attention as a flexible and efficient off-loading engine in System-On-Chips (SoCs). Evaluation results in recent researches revealed that the parameters of optimal processor array structure: granularity, functions, array size, context size and interconnection flexibility, are completely different for each application. That is, DRPAs should be configurable for target SoCs and applications. MuCCRA is a project for developing a DRPA generator which can generate RTL model, testing environment and programming environment for various types of DRPAs just by selecting the specific parameters. Here, two prototype chips MuCCRA-1 and MuCCRA-2 developed in the project are introduced and evaluated. MuCCRA-1 was implemented with Rohm's 0.18 $\mu$ m CMOS process mainly for multi-media applications, while MuCCRA-2 with ASPLA's 90nm CMOS process was designed focusing on area optimization used as a cost-effective IP in multi-core SoCs.

## 1. INTRODUCTION

Coarse grained dynamically reconfigurable processor arrays (DRPAs) have been received an attention as a flexible and efficient off-loading engine for various types of System-on-Chips (SoCs). Some devices are commercially available [1, 2, 3, 4, 5], and some of them have been integrated into digital appliances.

In order to achieve better area- and power-efficiency compared with traditional field-programmable devices such as FPGAs, they incorporate the following properties: (1) a simple coarse grained processor consisting of an ALU, a data manipulator, a register file and other functional modules is used as a primitive processing element (PE) of an array, and (2) dynamic reconfiguration of a PE array which enables time-multiplexed execution is introduced.

Unlike common FPGAs, in which the island-style structure using Look-Up-Tables (LUTs) with 4 or 5 inputs are commonly used, there exist wide design choices in DRPAs, such as the PE granularity, the number of hardware contexts which can be switched dynamically, the total amount

of wiring resource, and the size of PE array itself. Our performance evaluation results revealed that the optimal PE array size considering the area and power consumption is different for each application [6]. Thus, there is no all-around architecture in DRPAs, and the structure should be configurable or customizable for its main target application. Of course, the PE array is dynamically reconfigurable, but its granularity, size, interconnection and function should be configurable for each target application. Since DRPAs are assumed to be embedded into an SoC, customization of architectures is done at the design time like a configurable processor.

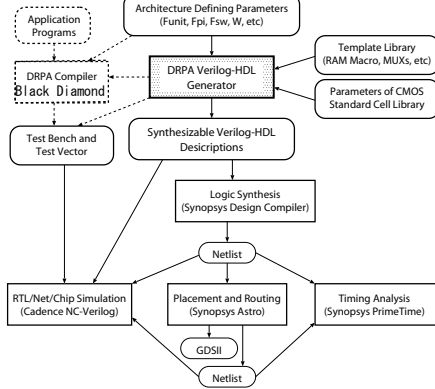
The object of Multi-Core Configurable Reconfigurable Architecture (MuCCRA) project is to develop a design methodology and framework which generate highly configurable DRPAs for various target applications. Here, we reported two prototype MuCCRA chips for establishing a flexible architecture generator.

## 2. CONFIGURABLE ARCHITECTURES MUCCRA

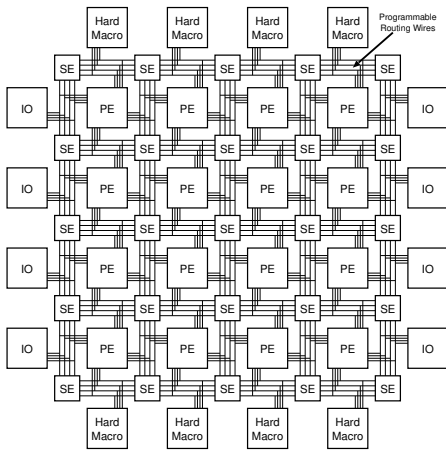
### 2.1. MuCCRA Design Environment

As shown in Figure 1, our final goal is generating both the chip layout of the DRPA and its programming environment based on the designer's demand. The fundamental DRPA architecture template is fixed, and the designers can generate their desired DRPAs by controlling parameters described in the parameter file. The target DRPA architecture model is called Multi-Core Configurable Reconfigurable Array (MuCCRA). Now, the generator can only generate a single core DRPA which can be used as an element of multi-core systems.

The generator reads the architectural parameter file, and generates the Verilog-HDL descriptions of DRPAs. A simple test bench is also generated for simulating the target architecture immediately. Since the generated Verilog-HDL descriptions are synthesizable, they can be logically and physically synthesized without any modifications. At the same time, architectural description files for a re-targetable compiler called Black Diamond which generates configuration data from C-like description are also created.



**Fig. 1.** MuCCRA Design Environment



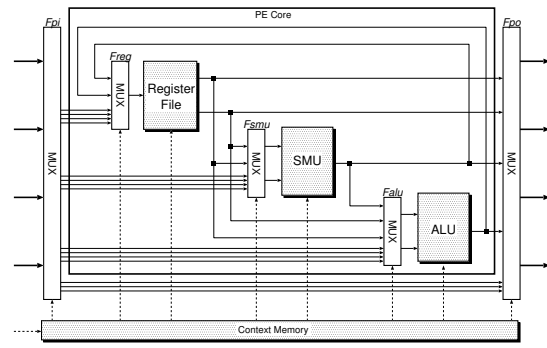
**Fig. 2.** MuCCRA Architecture with  $4 \times 4$  PEs

## 2.2. MuCCRA Architecture

MuCCRA architecture is consisting of a configurable part and fixed part. The PE array structure is parametrized, that is, the size of PE array, granularity of a PE, the number of hardware contexts, functions of each functional unit, intra-PE flexibilities, and inter-PE connections can be flexibly defined. On the contrary, the context control mechanism, configuration data management mechanism and data input/output are fixed. Thus, the interface between a MuCCRA core and other IPs is not changed.

### 2.2.1. Configurable part: Array Structure

An example of PE array structure of MuCCRA is shown in Figure 2. Here, an island-style interconnection structure like traditional FPGAs is adopted. That is, 2-dimensional interconnection which forms multiple routing channels is



**Fig. 3.** A Target PE Architecture and Intra-PE Flexibilities

provided, and each PE is surrounded by programmable routing wire segments. Connection blocks are provided between PEs and global routing channels for sending or receiving to or from PEs. On the intersection of a vertical and a horizontal channel, a Switching Element (SE) is placed. The SE is a set of simple programmable switches in which an entering link is connected to the other SEs. The number of channels in the global routing resources is parametrized as  $W$ , and each SE provides  $W$  independent switching modules. For each switching module, an entering link can be connected to  $F_{sw}$  other links. That is,  $F_{sw}$  means the SE flexibility or the flexibility of inter-PE global routing. The number of wires for a link is set to be the same as the granularity of PE ( $G$ ) shown later.

The structure of PE is shown in Figure 3. Each PE has a programmable PE Core, connection blocks, and a context memory. In the PE Core, like a lot of existing DRPA devices, a data manipulator called Shift & Mask Unit (SMU), an Arithmetic Logic Unit (ALU), and a register file are provided. Here, we use a single structure for every PE in the array. When PEs with special functions are required, they are allocated at the edge of PE array as a special hard macro.

The most fundamental parameter of MuCCRA is granularity of PE given by  $G$ .  $G$  specifies the data width treated in a PE and interconnection.  $G$  is set from 4 to 32 in the most cases.

The flexibility of interconnection of PE Core can be defined with the number of selectors provided on inputs and outputs of functional units such as PE. Each functional unit of PE Core has an input selector, and the number of input channels which can be selected by the unit is an important parameter. As shown in Figure 3, the input channel number for SMU, ALU, and register file are represented by  $F_{smu}$ ,  $F_{alu}$ , and  $F_{reg}$  respectively. These parameters are corresponding to the flexibility of intra-PE local routing.

Each PE is connected with global routing wires via connection blocks. The connection blocks pick up the data in global routing wires, and distribute to all functional units of a PE Core. We define the number of inputs and outputs that

can be connected to the connection blocks as  $F_{pi}$  and  $F_{po}$ . The operation of each function unit and local intra-PE connection are statically defined by configuration data called a context.

### 2.2.2. Common fixed part: Context Switching Mechanism and I/O

Each PE and SE in MuCCRA equip their context memory in which the configuration data for a particular operation is held. The central controller broadcasts a context pointer to all of the reconfigurable elements including PEs and SEs. The configuration data for a context is read out from the context memory according to the context pointer, and they are reconfigured in parallel. This type of dynamic reconfiguration is called a multicontext scheme, and a lot of current devices support it. In the multicontext devices, the dynamic reconfiguration can be done in only one clock cycle by distributing the context memory into each reconfigurable module.

The number of contexts which can be stored in the context memory is limited by the context memory size  $C$ . Thus, the configuration data which cannot be stored in the context memory is stored in the central configuration memory, and distributed to unused area of each context memory during the execution. This mechanism, called the virtual hardware, has been proposed and researched long time but rarely implemented in real chips. However, all MuCCRA chips provide this mechanism, and application which requires more contexts number than  $C$  can be executed. For high speed configuration data distribution, a multicast mechanism called RoMultiC[7] is adopted. The context control and configuration data distribution mechanism are common in all MuCCRA chips, and cannot be changed except the size of context memory  $C$  which influences the area of PE and SE.

I/O mechanism is also a fixed part of generated MuCCRA architecture. Like most of the released DRPAs, a certain number of distributed memory modules can be provided as hard circuit macros allocated at edges of PE array. The double buffering mechanism is adopted for each distributed shared memory[8], that is, two modules are provided for each distributed shared memory which can be switched each other after a task executed on PE array is finished. By using the mechanism, during computation, the results of the previous task and the streaming data to be processed in the next task are transferred through the I/O modules.

## 3. MUCCRA PROTOTYPE CHIPS

In order to develop and evaluate the DRPAs generation, we implemented two MuCCRA prototype chips: MuCCRA-1 and MuCCRA-2.

The parameters and processes used in two prototype chips are shown in Table 1. Here,  $F_{alu}$ ,  $F_{smu}$  and  $F_{reg}$  are set to be

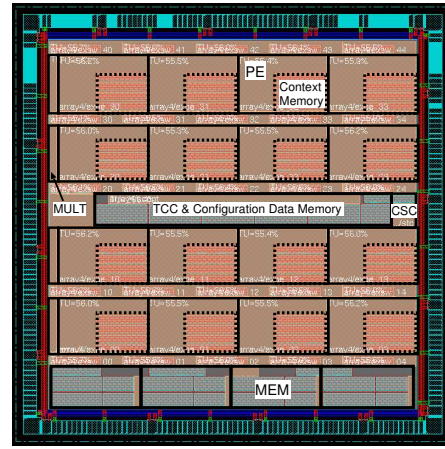


Fig. 4. Layout of MuCCRA-1 Chip

the same, and referred as  $F_{unit}$ .  $F_{po}$  is set the same as  $F_{pi}$ .

Table 1. Specifications of MuCCRA-1 and MuCCRA-2

	MuCCRA-1	MuCCRA-2
Granularity( $G$ )	24bits	16bits
Array Size	4×4	4×4
Contexts Size ( $C$ )	64	16
Connect. Flex. ( $F_{unit}$ , $F_{pi}$ )	4	4
Interconnect. Flex. ( $F_{sw}$ )	2	3
IP modules	Multipliers Memory	Memory
Process	Rohm's 0.18um	ASPLA's 90nm

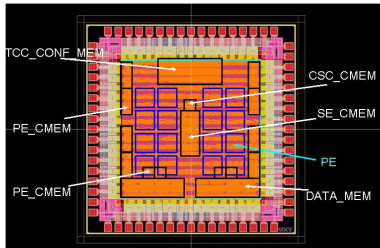
### 3.1. MuCCRA-1

The first prototype, MuCCRA-1, was designed with Rohm's 0.18um process, and implemented on 5-mm square die with 189 I/O pads. Since it is designed for multi-media processing, the granularity ( $G$ ) is set to be 24bits. Multiplier modules are provided at the edge of the PE array consisting of 4×4 normal PEs without a multiply operation. That is, it uses a heterogeneous structure. As shown in the layout (Figure 4), a large part of PE and SE are occupied by the context memory which can hold 64 contexts.

MuCCRA-1 was taped out on the last November, now under fabrication, and will be available on this July. Table 2 shows the execution time of designed applications evaluated with post-layout simulation. "Discrete Cosine Transform (DCT)" is a task used in JPEG coder, " $\alpha$ -Blender" is a simple image processing, "SHA-1" is a hash algorithm used in an encryption, and "Viterbi" is a decoder for error correcting code used in communication. It works from 20MHz to 40MHz clock speed depending on the application design, and the execution speed is about twice of that of the Texas

**Table 2.** Execution Time for 4 Applications on MuCCRA-1

	ExecClocks	Delay [ns]	ExecTime [ $\mu$ s]	Power [mW]
DCT	195	40	7.8	85.1
$\alpha$ -Blender	644	24	15.5	103.3
SHA-1	418	50	20.9	50.6
Viterbi	600	42	25.2	45.9

**Fig. 5.** Layout of MuCCRA-2 Chip

Instrument's digital signal processor (TMS320C6713) which works 225MHz clock with extremely low power consumption.

### 3.2. MuCCRA-2

MuCCRA-2 was implemented on 2.5-mm square die with 51 I/O pads. ASPLA's 90nm process was used. As shown in the layout (Figure 5), since I/O pads and buffers occupy a large part of die area, the core of MuCCRA-2 is only 1.5-mm square. Since MuCCRA-1 requires considerable area as an IP embedded on a multi-core SoC, the main challenge of MuCCRA-2 is the reduction of the area without degrading its performance. For this purpose, we adopted smaller granularity and context size than those of MuCCRA-1. A context memory module is shared by two PEs and four SEs for reducing the number of memory modules. On the other hand, multiply operations are provided in all PEs, since it appeared that the number of multipliers often dominates performance in MuCCRA-1. As a result, the array becomes a homogeneous structure. The interconnection capacity is also enhanced so as to improve the utilization ratio of PEs.

**Table 3.** Applications on MuCCRA-2

Application	Contexts	Delay(nsec)	Exe. time (nsec)
$\alpha$ -Blender	5	11.1	5643
Contrast	11	13.1	5057

MuCCRA-2 was taped out on this April, is now under

fabrication, and will be available on this September. Table 3 also shows the execution time of designed applications ("Contrast" enhances the contrast of input image using histogram equalization.) evaluated with post-layout simulation. These application programs were developed with a re-targetable compiler Black Diamond. Although enough number of applications have not been implemented, the performance of MuCCRA-2 is about three times that of MuCCRA-1 if the computation data width is less than 16bits.

## 4. CONCLUSION

Here, a parametrized DRPA generator is introduced. By specifying architectural parameters such as PE granularity and several connection flexibilities, the generator can automatically generate the synthesizable Verilog-HDL description and verification environment. Two prototype chips MuCCRA-1 and MuCCRA-2 have been developed in the project. The evaluation results using applications demonstrated that the proposed tool can generate practical layouts and programming environment.

**Acknowledgement** This work is supported in part by Japan Science and Technology Agency (JST) and Japan Society for the Promotion of Science (JSPS). The authors thank to VLSI Design and Education Center (VDEC), and Prof. Kobayashi and his colleagues in Kyoto University for their design flow of ASPLA/STARC 90-nm CMOS process.

## 5. REFERENCES

- [1] F. Veredas, M. Scheppler, W. Moffat, and B. Mei, "Custom Implementation of the Coarse-Grained Reconfigurable ADRES Architecture for Multimedia Purposes," in *Proc. of FPL*, Aug. 2005, pp. 106–111.
- [2] M. Motomura, "A Dynamically Reconfigurable Processor Architecture," *Microprocessor Forum*, Oct. 2002.
- [3] T. Sugawara, K. Ide, and T. Sato, "Dynamically Reconfigurable Processor Implemented with IPFlex's DAPDNA Technology," *IEICE Trans. on Information & System*, vol. E87-D, no. 8, pp. 1997–2003, May 2004.
- [4] M. Petrov, et al., "The XPP Architecture and Its Co-simulation within the Simulink Environment," in *Proc. of FPL*, Aug. 2004, pp. 761–770.
- [5] Rapport, Inc., <http://www.rapportincorporated.com/>.
- [6] Y. Hasegawa, et al., "Performance and Power Analysis of Time-multiplexed Execution on Dynamically Reconfigurable Processor," in *Proc. of RAW*, Apr. 2006.
- [7] V. Tanbunheng, M. Suzuki, and H. Amano, "RoMultiC: Fast and Simple Configuration Data Multicasting Scheme for Coarse Grain Reconfigurable Devices," in *Proc. of FPT2005*, Dec. 2005, pp. 129–136.
- [8] H. Amano, et al., "An I/O mechanism on a Dynamically Reconfigurable Processor - Which should be moved: Data or Configuration," in *Proc. of FPL*, Sept. 2005, pp. 347–352.