PROCEEDINGS

SBCCI2001

14th SYMPOSIUM ON INTEGRATED CIRCUITS AND SYSTEMS DESIGN

Pirenópolis, GO, BRAZIL
10-15 September 2001

SBC Brazilian Computer Society

IFIP WG10.5

acm

IEEE COMPUTER SOCIETY

Proceedings

14th Symposium on

Integrated Circuits and Systems Design

10-15 September 2001 – Pirenopolis, Brazil

Edited by

Ricardo Jacobi, Antonio Ferrari and Luigi Carro

Sponsored by

SBC – Brazilian Computer Society

Co-Sponsored by

IFIP WG10.5 - International Federation for Information Processing

SBMicro - Brazilian Microelectronics Society

Supported by

Capes, CNPq

In Cooperation with

ACM Sigda

Organized by

University of Brasilia (UnB)

IEEE COMPUTER SOCIETY

Los Alamitos, California

Washington • Brussels • Tokyo

IEEE Computer Society Order Number PR01333
ISBN 0-7695-1333-6
ISBN 0-7695-1333-4(case)
ISBN 0-7695-1335-2(microfiche)
Library of Congress 2001093975

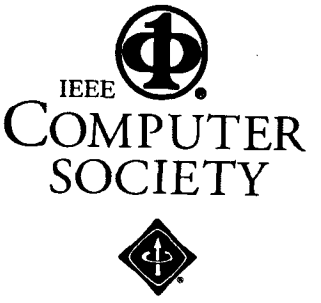
Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1 714 821 8380
Fax: + 1 714 821 4641
http://computer.org/
csbooks@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
http://shop.ieee.org/store/
customer-service@ieee.org

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

Editorial production by Frances M. Titsworth
Cover art design by Ricardo Reis
Cover art production by Joseph Daigle/Studio Productions
Printed in the United States of America by The Printing House



Foreword

On behalf of the SBCCI 2001 Organizing Committee and Program Committee, we wish you a warm welcome to Pirenópolis for the 14th Symposium on Integrated Circuits and Systems Design. For the first time it is held jointly with SBMicro 2001, the International Conference on Microelectronics and Packaging and the SBAC 2001, the Brazilian Symposium on Computer Architecture. We wish you a productive and agreeable interaction with participants of those events. SBCCI is a well established event that is now 18 years old. The first event was held in 1983 in Porto Alegre and since then traveled around the country, being located in Porto Alegre, Gramado, Rio de Janeiro, Ouro Preto, Jaguariúna, Recife, Buzios, Natal, and Manaus. Next year the event will be back to Porto Alegre. The Program Committee relied on several colleagues from many different countries and continents, which contributes to the high quality of the program and consolidates its international character. We had this year 37 papers accepted for presentation in 10 sessions: Embedded Systems, Rapid Prototyping, Formal Methods, Codesign, Cad & Test, Digital Design I, Digital Design II, Analog Design, Low-Power and Low-Voltage, and Physical Design. Besides the technical sessions there will be international tutorials, panels and, for the first time, a User Forum with special emphasis on student participation. Moreover, this year, for the first time, SBCCI occurs in cooperation with ACM Sigda. We would like to thank our sponsor, the Brazilian Computer Society, and the cosponsors, the International Federation for Information Processing – IFIP WG 10.5 – and the Brazilian Microelectronics Society, as well as all colleagues that contributed to the success of this initiative. This year SBCCI takes place in the Brazilian Cerrado, well known for its beautiful landscapes and nice weather. We wish you an excellent and fruitful week in Pirenópolis.

Ricardo Pezzuol Jacobi
General Chair

António Ferrari and Luigi Carro
Program Committee Chairs

Conference Organizers

General Chair

Ricardo Jacobi, *UnB, Brazil*

Program Chairs

António Ferrari, *IEETA, Portugal*

Luigi Carro, *UFRGS, Brazil*

Publicity Chair

Ricardo Reis, *UFRGS, Brazil*

Organizing Committee

José Camargo, *UnB*

Alba Cristina Melo, *UnB*

Anésio Lelles, *UnB*

Maria Aparecida, *UnB*

Carlos Llanos Quinteiro, *IESB*

Maria Emília Walter, *UnB*

Marie Honda, *UnB*

Steering Committee

Marcelo Lubaszewski, *UFRGS* (**chair**)

Edna Barros, *UFPE*

Ivan Saraiva Silva, *UFRN*

Ricardo Jacobi, *UnB*

Ricardo Reis, *UFRGS*

Vladirmir Castro Alves, *UFRJ*

Wilhelmus Van Noije, *USP*

Program Committee

Alex Orailoglu, *UC San Diego, USA*
Antonio Ferrari, *Univ. Aveiro, PT*
Arlindo Oliveira, *IST, PT*
Armando Gomes, *Motorola, BR*
Bernard Courtois, *TIMA, FR*
Carlos Llanos, *UPIS, BR*
Carlos Montoro, *UFSC, BR*
Daniel Gajski, *UC Irvine, USA*
David Deharbe, *UFRN, BR*
Dominique Borrione, *IMAG, FR*
Edna Barros, *UFPE, BR*
Erik Dagless, *Univ. Bristol, UK*
Erik Jan Marinissen, Philips, NL
Ernani Randon, Univ. S. Bolivar, VE
Eugenio Villar, Univ. Cantabria, ES
Fabian Vargas, PUCRS, BR
Fernando Moraes, PUCRS, BR
Fernando Silveira, IIE, UY
Franz Rammig, Univ. Paderborn, DE
Ivan Saraiva, UFRN, BR
Jose C. Monteiro, IST, PT
José da Mata, UFMG, BR
Jose Luis Huertas, CNM, ES
Juan Moreno, Univ. Catalunya, ES
Juergen Becker, TH Darmstadt, DE
Luigi Carro, UFRGS, BR
Luis Miguel Silveira, IST, PT
Luis Toledo, *UCC, AR*
Manfred Glesner, *TH Darmstadt, DE*
Manoel Lois, *UFRJ, BR*
Marcel Jacomet, *MicroLab I3S, CH*
Marcelo Lubaszewski, *UFRGS, BR*
Marius Strum, *USP, BR*
Martin Bolton, *STM, UK*
Masahiro Fujita, *Fujitsu, USA*
Meryem Marzouki, *Univ. Paris VI, FR*
Michel Renovell, *LIRMM, FR*
Oscar Calvo, *UBI, ES*
Peter Cheung, *Imperial College, UK*
Rajesh Gupta, *UC Irvine, USA*
Raoul Velazco, *IMAG, FR*
Raul Camposano, *Synopsis, USA*
Reinaldo Bergamaschi, *IBM, USA*
Reiner Hartenstein, *Univ. Kaiserslauten, DE*
Ricardo Ferreira, *UFV, BR*
Ricardo Jacobi, *UnB, BR*
Ricardo Reis, *UFRGS, BR*
Roman Hermida, *UCM, ES*
Sergio Bampi, *UFRGS, BR*
Syed Kamrul Islam, *Univ. Tennessee, USA*
Valentino Liberali, *Univ. Pavia, Italy*
Victor Champac, *INAOEP, Mexico*
Vladimir Castro Alves, *UFRJ, BR*
Yervant Zorian, *LogicVision, USA*

Reviewers

Alex Orailoglu
Alfredo Olmos
Altamiro Suzim
Ana Freitas
Andre Inacio Reis
Antonio Ferrari
Arlindo Oliveira
Armando Gomes
Bernard Courtois
Bernardo Kastrup
Carlos Beltran
Carlos Llanos
Carlos Montoro
Daniel Gajski
Dariusz Caban
David Deharbe
Dominique Borrione
Edna Barros
Erik Dagless
Erik Jan Marinissen
Ernani Randon
Eugenio Villar
Fabian Vargas
Fernando Chavez
Fernando Moraes
Fernando Silveira
Franz Rammig
Gilson Inacio Wirth
Heinz-Dieter Hummer
Henrique Santos
Horacio Neto
Isabel Teixeira
Ivan Saraiva Silva
Joao Paiva Cardoso
John Hatfield
José Augusto Lima
José C. Monteiro
José Carlos Pedro
José da Mata
José Luis Cura
José Luis Huertas
José Machado da Silva
José Martins Ferreira
Juan Carlos Lopez
Juan Moreno
Juergen Becker
Katzalin Olcoz
Luigi Carro
Luis Miguel Silveira
Luis Toledo
Manfred Glesner
Manoel Eusébio Lima
Manoel Lois
Manuel Medeiros Silva
Marcel Jacomet
Marcelo Lubaszewski
Marius Strum
Martin Bolton
Masahiro Fujita
Meryem Marzouki
Michel Renovell
Milagros Fernandez
Oscar Calvo
Pablo Petrashin
Peter Cheung
Rajesh Gupta
Raoul Velazco
Raul Camposano
Reinaldo Bergamaschi
Reiner Hartenstein
Ricardo Ferreira
Ricardo Jacobi
Ricardo Reis
Roman Hermida
Rui Aguiar
Rui Escadas Martins
Rui Silva Martins
Sergio Bampi
Sergio Cavalcante
Syed Kamrul Islam
Valentino Liberali
Valery Sklyarov
Victor Champac
Vladimir Castro Alves
Walter Lancioni
Wang Jiang Chau
Yervant Zorian
Zein Juneidi

Sponsoring Societies

Sponsor: Brazilian Computer Society

Board of Directors

President: Flávio Rech Wagner, *UFRGS*

Vice-President: Luiz Fernando Gomes Soares, *PUC-Rio*

Director of Education: Sergio Schneider, *UFU*

Director of Events and Special Committees: Dilma Menezes da Silva, *USP*

Director of Publications: Ricardo Anido, *Unicamp*

Director of Regional Divisions: Robert C. Burnett, *PUC-PR*

Director of Marketing: Geber Ramalho, *UFPE*

Director of Special Programs: Claudionor J. Coelho Jr., *UFMG*

Director of Administration and Finances: Taisy Weber, *UFRGS*

Co-Sponsor: International Federation of Information Processing-IFIP WG 10.5

Board of Directors

President: P. Bollerslev, *DK*

Vice-President (Publications): R. Aiken, *US*

Vice-President (Finances): J. Granado, *PT*

Vice-President (Marketing): W. Grafendorfer, *AT*

Treasurer: D. Khakhar, *SE*

Secretary: R. Johnson, *IND.M*

Co-Sponsor: Brazilian Microelectronics Society – SBMicro

Board of Directors

President: Renato Perez Ribas, *UFRGS*

Vice-Presidente: Galdenoro Botura Jr., *UNESP*

Treasurer: Patrick Verdonck, *USP*

Secretary: Luis Otavio Saraiva Ferreira, *LNLS*

Table of Contents

14th Symposium on Integrated Circuits and Systems Design

Session 4 – Codesign

Session 5 – CAD & Test

Session 6 – Analog Design

Session 7 – Digital Design I

Session 8 – Physical Design

Session 1

Embedded Systems

Adaptive Systems-on-Chip: Architectures, Technologies and Applications

Jürgen Becker, Thilo Pionteck, Manfred Glesner

Darmstadt University of Technology
Institute of Microelectronic Systems
Karlstr. 15, D-64283 Darmstadt, Germany
e-mail: {becker, pionteck, glesner}@mes.tu-darmstadt.de

Abstract

The fast technological development in Very Large Scale Integration (VLSI) has enabled chip-designers to integrate complete electronic systems, formerly built of several separate chips, onto one single piece of silicon. These Systems-on-Chip (SoCs) introduce a set of various challenges for their interdisciplinary microelectronic implementation, from system theory (application) level over efficient CAD methods to suitable technologies. Important aspects for the industry are the flexibility and adaptivity of SoCs, which can be realized by integrating reconfigurable hardware parts on different granularities into Configurable Systems-on-Chip (CSoCs). The paper describes the major challenges and first approaches in architecture, design and application of application-specific adaptive SoCs, e.g. in digital baseband processing for future mobile radio devices.

1 Introduction

In the last years, the fast technological development in VLSI possibilities has brought the notion to single *System-on-Chip* (SoC) solutions. Trends in microelectronic systems design point to higher integration levels, smaller form factor, lower power consumption and cost-effective implementations. The achievement of this goal has to be efficiently supported by the concurrent development of new design methods including such aspects as flexibility, mixed-signal system-level exploration, re-usability and top-down SoC design. ICE defines a SoC to be a single chip that contains processing elements, application specific *Intellectual Property* (IP) and storage elements to define the overall function of the end product it supports [1]. But it seems that this definition is not quite sufficient. For example, this definition would also apply to the first electronic calculator, which contains a single IC, that is the entire calculator system. But these ICs only contained about 5000 gates and, certainly, nobody would call such an IC a SoC. In fact, such ICs would preferably be called Application Specific Integrated Circuit (ASIC) and SoCs can be described as an extension of the ASIC technology where the functionality that previously required a printed circuit board is merged onto a single silicon chip. The first SoCs appeared in the early 1990s and consisted almost exclusively of digital logic constructions. Today SoCs are often mixed-technology designs, including such diverse combinations as embedded DRAM, high-performance or low-power logic, analog, RF, and even more unusual technologies like Micro-ElectroMechanical Systems (MEMS) and optical input/output. But this development also raises its problems, e. g. it takes an enormous amount of time and effort to design a chip. With the predicted shrinking in semiconductor process geometries these problems will increase. The gap between of what can be built (silicon capacity) and of what can be designed is widening. So new design methodologies are needed to improve the design process to keep up with the technology improvements. The cornerstone of this required change in design methodologies will be the augmented use of parts from previous designs. The concept of using already existing parts from previous designs can be extended by making use of parts designed by third parties, which is called IP- or Core-based design [9], [10]. An overview on the status and perspectives of SoCs and IP-based EDA can be found in [2].

Dependent on the application areas and constraints, important aspects for the microelectronic SoC candidate architectures and technologies are:

- *time-to-market* constraints have to be fulfilled,
- SoC implementation *flexibility*, e.g. *risk minimization* in the case of late specification changes,
- long product life cycles, due to multi-standard / multi-product implementation perspectives, and
- due to multi-purpose usage, high volumes of the same SoC to be fabricated (-> *cost decrease per chip*).

Microelectronic system designers now have 2 major alternatives for SoC integration:

- ASIC-based SoCs, consisting mainly of processor-, memory-, and ASIC-cores, or

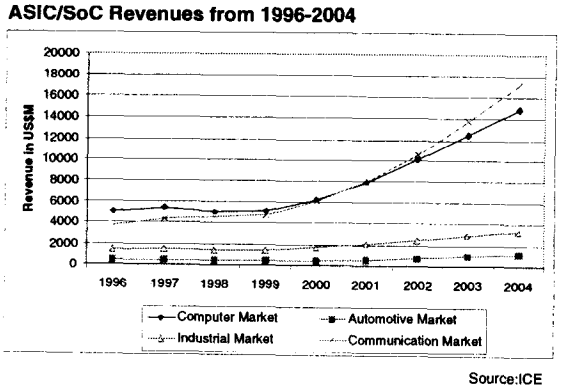


Figure 1: ASIC/SoC revenues in various areas

0-7695-1333-6/01 $10.00 © 2001 IEEE

- *Configurable SoCs* (CSoCs), consisting of processor-, memory-, probably ASIC-cores, and on-chip reconfigurable hardware parts for customization to a particular application.

CSoCs combine the advantages of both: ASIC-SoCs and multichip development using standard components, e. g. they require only minimal NRE costs, because they don't need expensice ASIC-tools and mask sets. In the following, recent developments and trends, as well as actual architectures, technologies and applications are discussed.

2 CSoCs: Developments & Trends

As stated above, most of microelectronic SoC solutions are a combination of ASICs, microcontrollers, and Digital Signal Processors (DSP) devices, whereas the percentage of ASIC-based SoCs within total ASIC market is steadily increasing (see figure 2). Appropriate final implementation technologies have to be selected, whereas different implementation trade-off alternatives concerning flexibility, cost, low power, and performance have to be considered, dependent on the application and situation. Thus, new emerging technologies like reconfigurable hardware architectures have also to be considered as alternatives for DSP- and/or ASIC-technologies, dependent on the required implementation trade-off. Reconfigurable hardware architectures have been proven in different application areas [3] [4] [5] [11] [12] to produce at least one order of magnitude in power reduction and increase in performance.

In the last years the ASIC/SoC markets for computer and communication applications had explosive revenue increases, compared to the industrial and autommotive areas (see figure 1). Especially, future markets for mobile communication systems promise huge revenues and a lot of challenges for the necessary microelectronic SoC solutions. 2nd generation (2G) mobile communication systems, i.e. GSM and IS-95 standards, had been rigorously defined and optimized to provide mainly operation for voice transmissions. On the other hand, 3G systems, i.e. based on the UMTS standard, will be defined to provide a transmission scheme which is highly flexible and adaptable to new services. This vision adds a new dimension to the challenges within the digital baseband design, since the final microelectronic systems must be able to support such a flexibility and adaptivity to mobile terminal to accommodate new services and

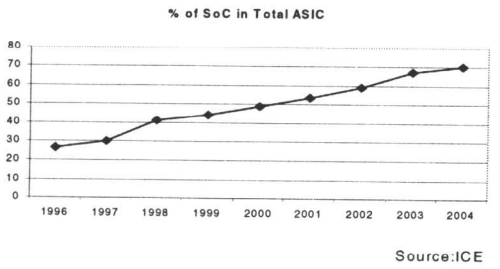


Figure 2: ASIC-based SoCs in total ASIC

situations easily and quickly. In figure 3 the DSP software performance requirements for the major signal processing tasks in next generation's UMTS receiver is given, according to [17]. Relative to GSM, UMTS and IS-95 will require intensive layer 1 related operations, which cannot be performed on today's processors [18]. Thus, an optimized HW/SW partitioning of these computation-intensive tasks is necessary,whereas the flexibility to adapt to changing standards and different operation modes has to be considered. Since today's low power DSPs cannot afford such a performance, the DSP load has to be reduced to release it for added value applications and to save power. Therefore, selected computation-intensive signal processing tasks have to be migrated from software to hardware implementation, e. g. to ASIC or reconfigurable hardware SoC parts (see section 3). Based on this situation and future market demands, now many industrial and academic CSoC products and approaches arise [11] [12] [13] [14] [15] [16] [19] [20] [21] [22] [23]. Especially the strong industry efforts, also of major players like Hitachi [16], indicate impressively the perspectives of CSoCs. In the following section two selected academic CSoC approaches are described and compared.

3 CSoCs: Architecture & Technologies

Future target SoC architecture may be composed of different cores such as DSPs, microcontrollers and memories, as well as of reconfigurable hardware and/or various ASIC support parts, whereas the final structures result from a detailed application and performance analysis while considering VLSI-oriented implementation issues. For example, the design of mobile baseband systems involves several heterogeneous areas, covering various aspects in communica-

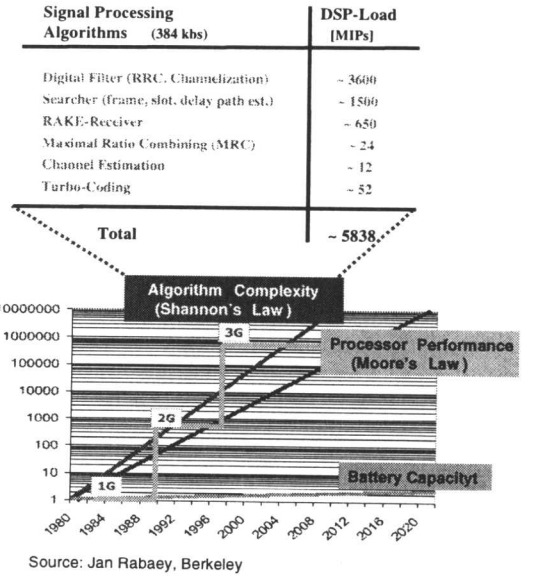


Figure 3: Baseband algorithm complexity [17] [18]

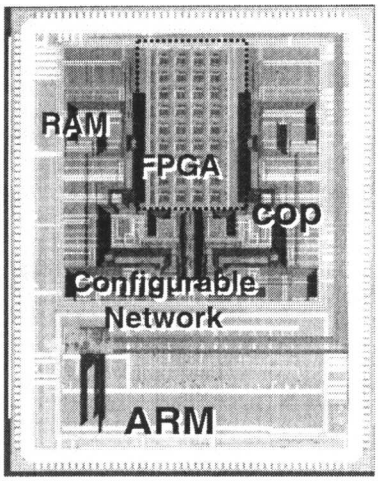


Figure 4: Major components of MAIA CSoC [20]

tion system application, in efficient CAD tool support, as well as in microelectronic architectures and technology issues. A good understanding of all relevant points related to this interdisciplinary area is essential for the success of the final product [18].

Thus, the major general goal for the development of such application-tailored architectures is to realize flexibility vs. power/performance trade-offs by releasing the DSP for other tasks, or by migrating functionality from ASICs to multi-granularity reconfigurable hardware. The following, two academic CSoC approaches will be sketched:

- the MAIA CSoC using a universal fine-grain reconfigurable hardware part [19] [20], and
- an application-specific CSoC using a coarse-grain dynamically reconfigurable DReAM architecture [6] [7]

The MAIA CSoC incorporates fine-grain FPGAs in its structure. The base architecture consists of one control processor and other satellite units (can be processors, FPGAs or other units such as MAC). During computation and reconfiguration sequential threads are instantiated on the control processor, which configures the satellite processors and the on-chip reconfigurable communication network and manages the overall control flow of applications, either in a static compiled order, or through a dynamic real-time kernel. Thus, the architecture is reconfigurable in two respects - inter-satellite communication confgurations and the fine-grain FPGA hardware part (see figure 4) . The MAIA processor consists of a microprocessor core (ARM8) and 21 satellite processors: two MACs, two ALUs, eight address generators, eight embedded memories(4 512x16bit, 4 1kx16bit) and an embedded low-energy FPGA. Connections between satellites are accomplished through 2-level hierarchical mesh-structured reconfigurable interconnect network. The ARM8 uses an interface control unit to configure and communicate data with satellites. The address

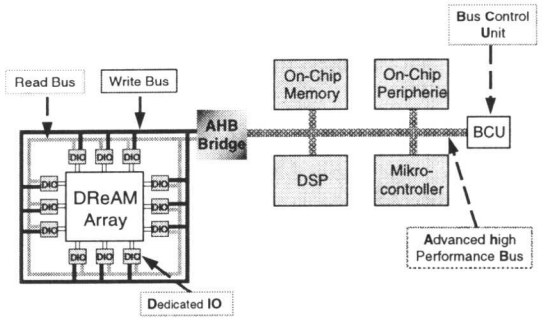


Figure 5: Integration of DReAM within CSoCs [6]

generators and embedded memories are distributed to supply multiple parallel data streams to the computational elements. The MAIA chip was implemented using 0.25U 6-level metal CMOS process with a supply voltage of 1V and additional voltages of 0.4V and 1.5V, The die size of the implementation was 5.2mmx6.7mm with 1.2 million transistors at 40 MHz with an average power dissipation of 1.5-2 mW. The Maia CSoC is optimized for selected mobile communication application parts, e. g. a full-rate VSELP voice coder algorithm was implemented at 30 MHz with 5.7 GOPS/Watt [19].

An example for a CSoC with integrated coarse-grain dynamically reconfigurable hardware is the DReAM (Dynamically Reconfigurable Architecture for future Mobile Communication Systems), which is currently synthesized at Darmstadt University of Technology. The DReAM architecture is designed for the requirements of future mobile communications systems, e.g. third generation (3G) systems [6]. Especially in the application area of mobile communication, standards are often changed or extended, which requires an adaptable SoC solution. The total system view

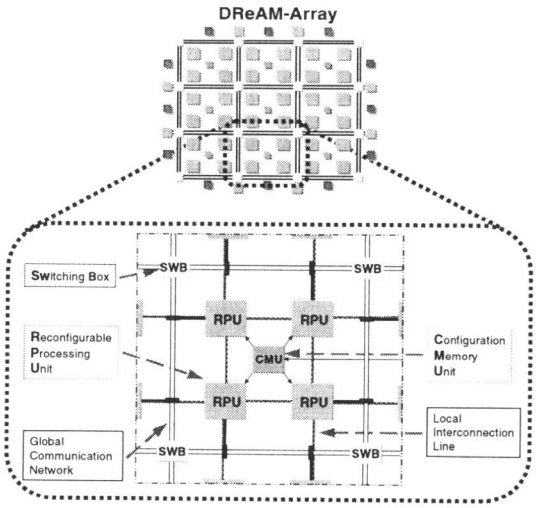
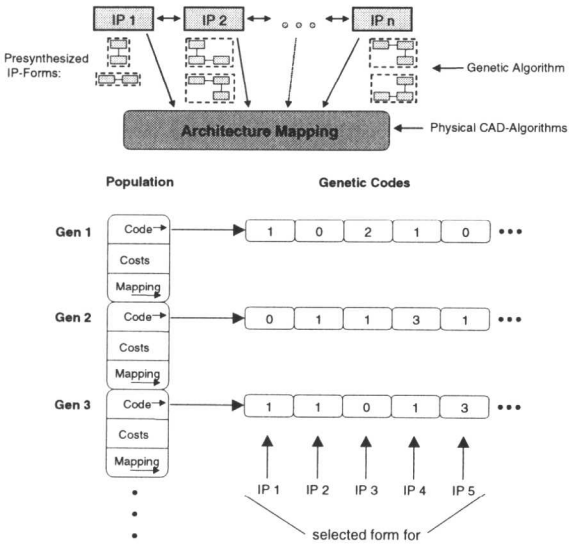


Figure 6: Hardware Structure of DReAM [6]



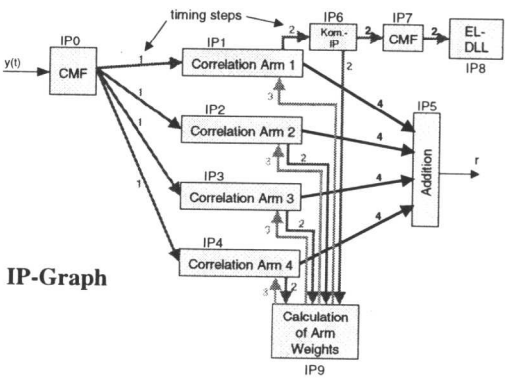

```
Basic Algorithm of Genetic Optimization
    clique based clustering of the IP-graph
    creating first population of size X
    WHILE counter < 15 DO
        FOR i = 0 TO i < X
            floorplanning
            routing
            saving best floorplan
        EndFor
        IF average_cost(population) <
            best_average cost THEN
            counter = 0
        ELSE
            counter = counter + 1;
        EndIf
        new_population = selection(population),
                         cross-over(population),
                         mutation(population)
        population = new_population
    EndWhile
End.
```

Figure 7: Genetic Optimization for IP-Mapping

of such a SoC is shown in figure 5. On the right side of the figure the typical SoC components like memory, DSP, and a microcontroler are shown. They are interconnected with an AHB bus, which is part the AMBA bus specification. Further details about the integration of DReAM with other SoC components can be found in [6]. The DReAM array hardware structure itself can be seen in figure 6. Reconfigurable Processing Units (RPUs), which can execute necessary arithmetic data manipulation for data-flow oriented protocol parts and also can execute FSM-type operations for control-flow oriented parts. The RPUs operate in parallel and have a 16-bit fast direct local connection to their direct neighbours and a 16-bit connection to the programable global communication network. In each RPU there is a small local memory in order to reduce the access of the main memory. Based on DReAM datapaths synthesized with an 0.35 µm CMOS process promising performance results are obtained for computation-intensive tasks of future CDMA-based communication systems, e. g. for implementing flexible RAKE-receiver architectures with 1.5 Mb/s data throughput [6]. In addition, efficient IP-based mapping techniques for DReAM including dynamic reconfiguration features are implemented and described in the following section, as well as an efficient rapid prototyping approach for such application-tailored CSoC solutions [8].

4 IP-based Mapping & Application

It is difficult to map applications onto coarse-grain dynamically reconfigurable architectures using today´s available programming and CAD tools. For developing new and promising IP-based methods the corresponding CAD tools also have to operate on the higher abstraction levels. Since automatized hardware synthesis from the behavioural and system level is still not sufficiently possible for ASICs and FPGAs, e.g. actual universal HDLs and their correpsonding synthesis environments are not suitable to support efficiently the application mapping of complex algorithms onto dynamically reconfigurable hardware arrays like DReAM. Therefore, we developed and implemented new IP-based CAD techniques for such kind of coarse-grained regular hardware architectures. Here, for each IP-core used within the possible complex application scenarios several alternatives of pre-synthesized IP-shapes are available in a characterized library, similar to standard cell synthesis. Such IP-



Running Times and Cost Values for Different Population Sizes

Population Size	Total Running Time (min)	Time Floor-planning (min)	Time Routing (min)	Time Genetic Opt. (min)	Analyzed Solutions	Best Solution (Costs)
20	43,4	14,0	24,8	1,4	962	255
40	45,5	15,7	29,3	1,3	1446	245
60	89,5	29,9	53,2	5,0	2358	240
80	73,1	26,1	40,1	4,5	1992	216

Best Solution without Clustering: 260

Figure 8: Mapping analysis of RAKE-Receiver

shapes consist of a variable number of RPUs and are realized by considering the special hardware attributes and toplogy of the DReAM array. Thus, the hardware designer has flexibilities how to realize a certain IP-funcion and the corresponding sophisticated and optimized pre-synthesis steps are done in advance for each IP to be used later efficiently by the architecture mapping phase (see figure 7). In addition, it is necessary to include additional information about data dependencies and data rates.

Global combinatoric optimization methods (genetic algorithms [25]) are applied here to find an optimal combination of the different IP-forms which creates a mapping with minimized hardware size and communication resources. The genetic optimization examines efficiently the whole design space. The result is the selection and generation of several IP-topology combinations that are to be used in the actual optimization step. Every chosen combination of IP-forms is mapped onto DReAM using extended macrocell floorplanning and placement methods based on *shape functions* ([24], see [7]), which are adapted to coarse-grained reconfigurable array architecture topologies.

At each step by the genetic algorithm so called individuals are created, whereas each of them is representing a particular combination of IP-forms or IP-topologies. The overall genetic optimization process is divided into the following major steps (see basic genetic algorithm in figure 7):

1. First, the best individuals of a generated population will be selected using a so called *fitness function*. They will be saved as an interim population.

2. Afterwards a certain number of these individuals will be fused by a *cross-over* function.

3. Next, the *mutation* operation is performed by transforming the genetic codes of a certain number of individuals slightly.

4. Finally, for each of the newly created individuals the best mapping and the *fitness function* value will be calculated. The optimization starts with step 1 again until 15 times the average fitness of the population has not improved.

The *fitness function* represents the quality metric for the genetic optimization process, which takes into account the needed number of RPUs ($c_1 * cost_{area}$),an virtual exceed of the allowed number of RPUs ($c_2 * cost_{vio}$) , and the communication costs between IP-cores ($c_3 * cost_{com}$).

In addition, dynamic reconfiguration allows to use areas of a DReAM-Array at different times for different puposes, e.g. a mobile system works on both, GMS and UMTS in separate time steps and with different hardware allocations. The CAD methods sketched here are able to generate the corresponding dynamic configuration codes for the DReAM-Array, which are dynamically accessible in the on-chip memory of the mobile device, if needed. Thus, CSoCs with integrated DReAM hardware can be flexible adapted to various situations in future mobile communication systems, e.g. switching between different standards and protocols, as well as between different bandwidth and service requirements.

As an example of an complex application for future mobile communication systems we examined a Rake-receiver algorithm based on CDMA transmission technology. It consists of 10 communicationgs IP-cores (see IP-graph in figure 8). For each of them several alternative IP-topologies were created in a library to use them in the optimization. The data tranfers between the IP-cores can be devided into 4 separat non-overlapping time steps. Therefore, for this example it is possible to optimize the necessary routing resources using dynamic reconfiguration. In figure 9 the resulting floorplans for all 4 non-overlapping time steps are shown with dynamic reconfiguration of needed interconnect allocations. The table in figure 8 shows the runtime measurments for different population size. The use of the

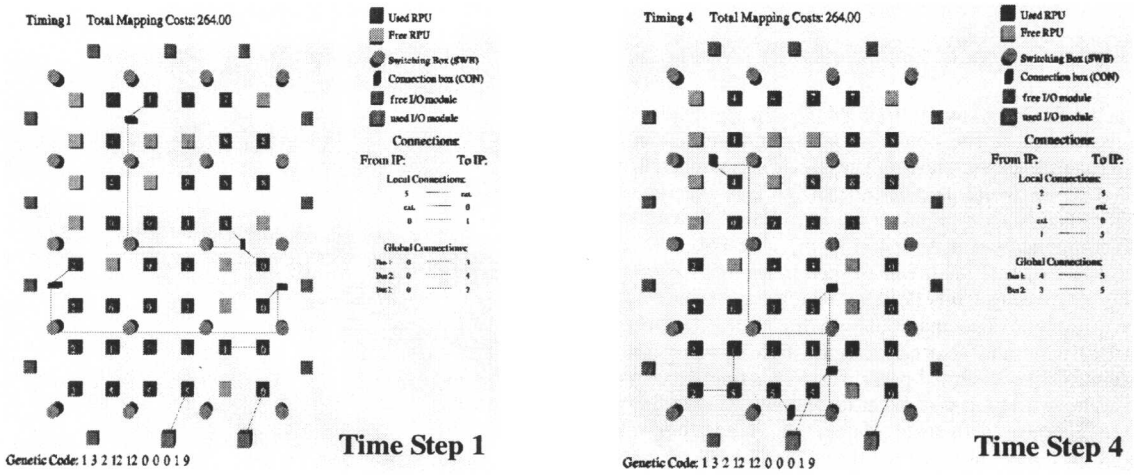


Figure 9: RAKE-Receiver Architecture Mapping Results for independent Time Steps 1 & 4