ASIP Architecture Implementation of Channel Equalization Algorithms for MIMO Systems in WCDMA Downlink

Predrag Radosavljevic, Joseph R. Cavallaro, and Alexandre de Baynast Department of Electrical and Computer Engineering Rice University, MS-366 - 6100 Main Street Houston, Texas 77005 Email: {rpredrag, cavallar, debaynas}@rice.edu

Abstract-This paper presents a customized and flexible hardware implementation of linear iterative channel equalization algorithms for WCDMA downlink transmission in 3G wireless system with multiple transmit and receive antennas (MIMO system). Optimized (in terms of area and execution time) and power efficient Application Specific Instruction set Processors (ASIPs) based on Transport Triggered Architecture (TTA) are designed that can operate efficiently in slow and fast fading high scattering environments. The instruction set of TTA processors is extended with several user-defined operations specific for channel equalization algorithms that dramatically optimize the architecture solution for the physical layer of the mobile handset. The final results of presented design-space exploration method are the ASIP processors with low cost/performance ratio. Automatic software-hardware co-design flow for conversion of C application code into gate-level hardware design of ASIP architectures is also described. Implemented ASIP solutions achieve real time requirements for 3GPP wireless standard (1xEV-DV standard, in particular) with reasonable clock speed and power dissipation.

I. INTRODUCTION

Efficiency and flexibility are crucial features of processors in the next generation of wireless cellular systems. Processors need to be efficient in order to achieve real-time requirements with low power consumption for computationally very demanding algorithms in new emerging wireless standards (3GPP, 4G, 802.11x, DVB-S2, DAB, just to name a few). Flexibility, on the other hand, allows design modifications in order to respond to: different channel environments, changes of user requirements depending of the quality of service (QoS), different workloads, different kinds of data, etc. Often, efficiency and flexibility goals are conflicting.

In this work we propose both flexible and application specific (customized) hardware solutions for implementation of channel equalization algorithms at the physical layer of the 3G mobile handset. Processors for mobile handsets in cellular systems that support the 3GPP standard ([1] and [2]) require at the same time both high speed and low power dissipation. In addition, computationally very demanding algorithms are needed to remove high levels of multiuser interference especially in the presence of multiple antennas on the basestation and mobile handset (MIMO wireless system). Traditional architecture solutions are ASIC and DSP processors.

While computationally efficient and low power solution, ASIC processors are not flexible enough to support the necessary variations of implemented wireless applications. On the other hand, DSP processors, although fully programmable, cannot achieve high performance with low power dissipation in highly parallel 3G applications. The drawback of DSP architecture solutions is limited level of instruction and data parallelism that is necessary for future 3G/4G wireless applications. These are the reasons for the recent interest in new reconfigurable architectures with some level of programmability [3] and, at the same time, possibility for customization that is targeted to a class of wireless applications with high levels of data and instruction parallelism. These architectures are called Application Specific Instruction set Processors (ASIPs) and can replace multiple chip designs implemented as an ASIC architecture [4].

In this work we propose the implementation of iterative chip level channel equalization algorithms on ASIP processors based on Transport Triggered Architecture (TTA) [5] in WCDMA MIMO downlink transmission. Channel equalization restores the orthogonality of the spreading waveforms destroyed by channel multipaths and suppresses strong Multiple Access Interference (MAI) and Inter-Symbol Interference (ISI). We show that the application specific processor design based on TTA is programmable and configurable enough to handle different variations of channel equalization. The designed ASIP architecture operates efficiently in a broad range of channel environments defined by 3GPP standard (from twopaths Pedestrian A to five-paths Vehicular A channels, speed of mobile subscriber varies from 3km/h to 120 km/h) [1]. In order to achieve real-time requirements in high data rate downlink applications, parallel architectures are developed for low power dissipation with the clock frequency limited to approximately 150 MHz.

This paper is organized as follows. The principles of channel equalization at the receiver side as well as the equalizationspecific operations are introduced in section II. Customized ASIP architectures for channel equalization algorithm in MIMO downlink are proposed in section III. Simulation estimates (clock speed, power dissipation, and area) obtained with the TTA software tools are presented in section IV. Detailed description of exploration for optimal ASIP architectures can be found in section V. The automatic software-hardware design flow and gate-level synthesis are presented in section VI. The conclusions are stated in section VII.

II. EQUALIZATION ALGORITHM

Powerful low complexity chip-level channel equalization based on iterative Conjugate Gradient (CG) algorithm [6] for inversion of covariance matrix and computation of filter coefficients in various scattering environments (slow and fast fading channels) has been proposed in [7] and [8]. The main feature of CG equalization is fast convergence speed to the LMMSE solution (direct inversion of covariance matrix). To be robust to the channel variations several adaptations of basic CG equalization have been applied [9] while keeping the same architecture. In this work, channel equalization and adaptive variations for fast fading environments are all mapped to the same optimized and flexible ASIP architecture.

Equalization algorithm is composed of two main steps: i) Channel estimation/covariance matrix computation/filter update part mainly consists of sign-test operations followed by complex addition/subtraction operations, as well as complex multiplications and real multiplication with shifting capability. ii) Filtering+despreading/descrambling (user detection) is a uniform algorithm that consists mainly of complex multiplications and accumulations. Indeed, the equalization algorithm consists of some non-standard operations therefore several specific user-defined operations can be designed for efficient ASIP implementation. The block diagram of channel equalizer in MIMO downlink based on iterative CG algorithm is shown in Figure 1.



Fig. 1. Block diagram of CG equalizer at the mobile handset

III. IMPLEMENTATION OF ASIP ARCHITECTURE

A. TTA architecture

The implemented ASIP architecture is based on the Transport Triggered Architecture (TTA) [5] that is a superclass of VLIW architecture. TTA exploits both instruction and data level parallelism. The architecture is flexible and new functional units (FUs), buses, and registers can be added without any restrictions. In addition, application specific support is provided by implementation of user-defined function units customized for a given application. TTA is programmed by directly specifying data transports which then trigger the operations as the side effect (implicit data control). The advantages of the TTA processors are, also, short cycle time, and fast and application specific processor design. The MOVE software toolset [10] enables an exhaustive design-space exploration.

B. Our ASIP Architecture Solution

We propose a 32-bit wide ASIP architecture (buses and ports of FUs are 32 bits wide) with 16-bit fixed-point arithmetic [9] for real-time equalization that is suitable to operate in low/high scattering and slow/fast fading environments. Real-time requirements are related to the 3GPP chip rate of 1.2288 Mcps [1] that provides available time of approximately 810 ns for processing one data sample.



Fig. 2. Optimized architecture: co-processor for CG filter update

The design of ASIP equalizer architecture is optimized by implementing several Special Functional Units (SFUs). Complex multiplier (CXMUL in Figures 2, and 3) with shifting (normalization) is one of the designed SFUs. Packing of 16-bit data (I and Q part of received samples) into 32-bit numbers enables that SFU for complex arithmetic have two, instead of four, input ports (the operands are two complex 32-bit numbers). Data are unpacked inside the SFU (unpacking of real and imaginary parts of two complex input operands), complex arithmetic on four 16-bit numbers is performed and two results (real and imaginary part) are packed in 32-bit value. In this way both the instruction and data parallelism are achieved within the CXMUL SFU and the amount of data transports across the buses is reduced. Consequently, the number of buses is significantly decreased. Comparison between traditional solution with real multipliers and real adders and the alternative architecture solution with complex multiplier is shown on Figure 3.

An SFU for arithmetic operations with sub-word parallelism is also implemented (sub-word add/subtract operation between two 32-bit numbers represents two parallel add/subtract operations on packed 16-bit operands, for example). The main feature of this Special Arithmetic Logic Unit (SALU in



Fig. 3. Real multipliers/adders and corresponding complex FU

Figure 2) is sign-test of pilot samples and then, depending on operand sign, the appropriate sub-word operation between I and Q values of received data is performed. This two-stage operation is a very frequent operation in the channel estimation algorithm. The third kind of SFU that is mainly utilized in updating of filter coefficients using iterative CG algorithm is real multiplication with right shifting by a varying number of bits. Since the fixed-point implementation of CG algorithm requires some arithmetic precision adjustments, the implementation of this SFU helps to achieve better convergence to the LMMSE solution.

By implementing all of these special function units we are able to significantly reduce the bus traffic and connections between buses and FUs, and to optimize (in terms of area, execution time and power dissipation) the overall architecture design. The cost database is updated with the power and area estimates of the implemented SFUs. Our design still contains standard FUs such as arithmetic FU (add/subtract, shifting) and load/store units as an interface with data memory. The optimized architecture configuration for CG filter update (full equalization algorithm presented in Figure 1 except filtering and user detection) with SFUs is presented in Figure 2.

We show that by exploiting the custom nature of ASIP architectures the internal structure is simplified especially the interconnection network between FUs that is a major concern for area occupation and power dissipation. Furthermore, the architecture design is more general than CG equalization - the majority of implemented user-defined operations can be utilized in other linear equalization schemes such as Least Mean Square (LMS). It can be shown that the identical ASIP architecture design can be programmed for LMS equalization [9].

IV. SIMULATION RESULTS

In this section we present two hardware architectures for CG equalization: i) single processor for full equalization referred to in Table I as '1', and ii) a second solution with two parallel co-processors: one for channel estimation/covariance matrix computation/filter update referred to as '2a', and the other co-processor for filtering+despreading/descrambling referred to as '2b'. Both solutions are evaluated with and without implementation of SFUs. The presented ASIP processors are obtained by using MOVE software tools [10] (compiler and processor explorer) that have been modified in order to be able

to utilize SFUs [9]. A cost-database of hardware components contains estimates for dynamic power dissipation and area based on the 0.13 μ CMOS technology. Frequency *f* is the minimum clock frequency necessary to achieve real-time requirements. In Table I, we consider a six-path Pedestrian B environment [11] (largest computational complexity) in the presence of two transmit and two receive antennas.

 TABLE I

 Comparative statistics of different equalizer architectures

 implemented with and without SFUs in Ped. B environment

| Туре | Buses | P[mW] | A[gates] | Real Mult | f[MHz] |
|--------------|-------|-------|----------|-----------|--------|
| 1. W/o SFUs | 32 | 190 | 262,063 | 8 | 203 |
| 2a. W/o SFUs | 32 | 119 | 190,554 | 8 | 102 |
| 2b. W/o SFUs | 32 | 128 | 230,009 | 8 | 102 |
| 1. W SFUs | 20 | 72 | 130,036 | 9 | 135 |
| 2a. W SFUs | 14 | 36 | 76,225 | 5 | 104 |
| 2b. W SFUs | 9 | 29 | 54,519 | 4 | 101 |

Implementation of SFUs causes less data traffic that leads to a significant reduction in the number of buses. The number of instructions (data transports - 'move operations') is significantly decreased (from 15,196 to 9,418, for full equalization of one block of 4096 received chips) by implementing SFUs for complex arithmetic and sub-word parallelism. The smaller interconnection network with fewer buses automatically reduces the instruction length - reduction from 736 bits (architecture 1. without SFUs from Table I) to 384 bits (architecture 1. with SFUs). As a consequence, the power dissipation and the area of the processor core are also dramatically decreased.

In the two co-processor architecture, filtering on the second co-processor is based on the updated filter coefficients that are sent from the first co-processor via the external RAM memory interface (shown in Figure 2). Two co-processors operate simultaneously in a pipelined fashion - filtering of the previous block of data samples is performed while the filter update for the current block is computed. The benefit of the single processor approach is the fact that there is no need for additional external interface for inter-processors. The main drawback is the larger workload that needs to be processed leading to a higher clock frequency and eventually to higher power dissipation (see simulation estimates from Table I).

TABLE II SIMULATION STATISTICS FOR SINGLE TTA PROCESSOR FOR FULL EQUALIZATION ALGORITHM IN DIFFERENT CHANNEL ENVIRONMENTS

| Environment | $T_{ex}[cycles]$ | P[mW] | f_{clk} [MHz] |
|-------------|------------------|-------|-----------------|
| PedA2x2 | 251,305 | 39 | 76 |
| PedB2x2 | 446,490 | 72 | 135 |
| VehA2x2 | 503,964 | 90 | 152 |

Simulation statistics (for processing one block of 4096 data samples) on the single TTA processor implementation with SFUs for full CG equalization algorithm in pedestrian and vehicular environments [11] is presented in Table II. After the design exploration phases (described in the next section for two-coprocessor solution), it is determined that the optimal architecture (in term of cost/performace ratio) consists of: 20 buses, 4 load/store units, 8 arithmetic FUs (ALUs), 3 SALUs, one real multiplier with shifting ability by various number of bits, and two CXMULs (total of nine real multipliers, see Table I). Data (received data samples and known training sequence) are stored in two dual-port RAM blocks. The proposed ASIP processor is a power efficient and flexible solution that can operate in different channel environments and achieve realtime requirements for 3GPP high data rate downlink standard with a reasonable clock frequency.

V. DESIGN EXPLORATION METHODOLOGY

More detailed insight into strategies of TTA design exploration for presented architecture solution with two simultaneously operating co-processors is described in this section. Also, a comparison between the two co-processor architecture and the single processor solution will be presented. In Figure 4, power and area estimates for both co-processors are shown as a function of minimum clock frequency needed to achieve realtime requirements for the 1xEV-DV standard [2]. Proposed architectures are optimized for CG equalization in a Pedestrian B channel environment (the most computationally complex case) with two transmit and two receive antennas.

The same following strategy for design exploration is applied for both co-processor architectures. The starting coprocessor configurations have a large number of: buses (24 for co-processor for filter update, and 16 for co-processor for filtering and user detection), register file ports (32 read and write ports divided into 8 register files), and hardware units (ALUs, SALUs, CXMULs, LSUs, etc) in order to achieve real-time requirements with a clock frequency of 66MHz. The resource exploration is performed, and the number of registers and register file ports are reduced to the point where the execution time is still approximately the same. Now, both co-processors have the same number of buses and FUs as before, but the number of register file ports and registers is significantly reduced. These become new starting co-processors for the next stage of resource optimization - reduction of mostly buses and function units. The result of this two-stage resource optimization process is the optimized co-processors with minimum number of register file ports, registers, buses and function units. Four discrete values (66, 100, 133, and 200 MHz) for minimum clock frequency for achieving realtime requirements are chosen for accurate power and area estimation since the cost database contains power and area estimates only for discrete values of clock period. This set of co-processors represents initial architectures for the last phase of design exploration - the connectivity optimization where the unnecessary connections between buses and function units are removed. This optimization stage simplifies the interconnection network between FUs and significantly lowers the power dissipation and gate count (area). The final estimates for power dissipation and area of fully optimized co-processors for the range of clock frequency between 66 MHz and 200 MHz are shown in Figure 4.



Fig. 4. Design exploration of co-processor for CG filter update and coprocessor for filtering, both optimized for Pedestrian B channel

The most optimal range of clock frequency required to achieve real-time requirements is between 100 MHz and 133 MHz. For a clock frequency of only 66 MHz, larger area (more hardware units that operate in parallel) is needed for achieving real-time requirements. Although the clock frequency is relatively low, the power dissipation is high because of the larger number of functional units, buses, and interconnections. On the other hand, if the minimum clock frequency is set to be higher (about 200 MHz), the area of the corresponding optimal co-processors is only slightly reduced since small architecture reduction in this range of clock frequency can cause significant deterioration of performance. Because of the higher clock frequency the power dissipation starts to increase in this region.

As mentioned, the single processor architecture for implementation of full CG equalization (pipelined execution of CG filter update algorithm and user detection on the same processor) is an alternative solution. The most efficient processor (in terms of power, area, and execution time) is already presented in section III (see Table II). An identical design exploration strategy is applied. The area of approximately 130,000 gates and the power dissipation (for clock frequency of 133 MHz) of about 72 mW are somewhat larger than the cumulative estimates for two co-processors for the same clock speed (see Figure 4). The reasons for this are larger workload and non-perfect parallelism between different sub-parts of the equalization algorithm.

VI. GATE-LEVEL SYNTHESIS OF ASIP ARCHITECTURES

In this section, the principles of automatic hardware implementation of optimized ASIP processors based on TTA with special function units will be described. Synthesis result for gate-level design of ASIP processor for full channel equalization (including the user detection) from Table II is presented. We show that by using several software tools from different vendors it is possible, in a fast and efficient way, to produce hardware implementation of the target ASIP processor. The entire hardware-software co-design flow is presented on Figure 5.



Fig. 5. Design flow from HLL code to hardware implementation

In general, hardware/software co-design [12] is a well known strategy for fast and flexible ASIC hardware implementation of applications described by High Level Languages (HLLs), such as the C/C++ programming language. This strategy allows to efficiently avoid design errors and to decrease design costs and time-to-market. In our case, hardware design starts with a C language description of the equalization algorithm that needs to be implemented at the physical layer of the mobile handset in a MIMO wireless system. As mentioned, by using modified MOVE tools and the library of designed special function units we are able to generate descriptions of area and power efficient ASIP processors. After that, the processor description file is converted into a VHDL representation of the processor core by using our modified MOVEGen (processor generator) tool [13]. Automatically generated VHDL code for the processor core together with VHDL code for predesigned components (program and data memories and other peripherals) can be directly used by the Xilinx XST synthesis tool [14] for fast FPGA prototyping. The same VHDL design can be used by Mentor Graphics tools (Leonardo Spectrum [15] and IC Station [16]) in order to obtain gate-level representation and layout of the target processors. CMOS libraries have to be also included in Mentor Graphics design flow.

The gate level representation of the proposed ASIP processor for full channel equalization is obtained by using Mentor Graphics tools. The ASIP processor for CG/LMS equalization with special function units is synthesized by using ASIC library for 0.5μ CMOS technology. The synthesis estimate of the processor core area (the area of processor without peripherals) is 135,014 gates which approximately corresponds to the area given by the TTA software simulator (see Table I) although the gate count is obtained by using 0.5μ instead of 0.13μ CMOS libraries.

VII. CONCLUSION

We proposed an optimized and flexible ASIP architecture based on TTA for 3GPP channel equalization at the physical layer of the mobile handset in MIMO wireless systems that can operate efficiently in various channel environments including high scattering fast fading transmission channels. It is shown that the area and power consumption can be dramatically reduced by implementing application-specific functional units. At the same time, additional speedup in the execution time is achieved. Two different hardware solutions are presented: single processor for full equalization and two interfaced coprocessors, as well as the methodology for design-space exploration for the optimal ASIP architectures for CG equalization. It is shown that the solution with two parallel co-processors achieves real-time with a slower clock frequency but requires the use of an external interface.

We also presented an efficient design flow for hardware implementation of ASIP architectures. This design flow starts with fixed-point C code for the application and ends with gatelevel implementation of the target processor. Several software tools from different vendors are efficiently combined together in order to achieve automatic software-hardware co-design of ASIP processors based on TTA.

VIII. ACKNOWLEDGEMENTS

This work was supported in part by Nokia Corporation, Texas Instruments, Inc., and by NSF under grants EIA-0224458, and EIA-0321266.

REFERENCES

- "1xEV-DV Evaluation Methodology (Rev.26)." Third Generation Partnership Project Two (3GPP2), May 2001.
- [2] J. Fonollosa, R. Gaspa, X. Mestre, A. Pages, M. Heikkila, J. Kermoal, L. Schumacher, A. Pollard, and J. Ylitalo, "The IST METRA project," *IEEE Communications Magazine*, vol. 40, pp. 78–86, July 2002.
- [3] F. Sun, S. Ravi, A. Raghunathan, and N. K. Jha, "Custom-Instruction Synthesis for Extensible-Processor Platforms," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, pp. 216–228, February 2004.
- [4] K. Keutzer, S. Malik, and A. R. Newton, "From ASIC to ASIP: The Next Design Discontinuity," in *IEEE International Conference on Computer Design*, pp. 84–90, September 2002.
- [5] H. Corporaal, Microprocessor Architecture: From VLIW to TTA. John Wiley and Sons, 1997.
- [6] G. Golub and C. Van Loan, *Matrix Computations*. The Johns Hopkins University Press, 1991.
- [7] M. Heikkilla, K. Routsallainen, and J. Lilleberg, "Space-Time Equalization using Conjugate-Gradient Algorithm in WCDMA Downlink," in *Personal, Indoor and Mobile Radio Communications*, Sept. 2002.
- [8] A. de Baynast, P. Radosavljevic, and J. R. Cavallaro, "Chip Level LMMSE Equalization for Downlink MIMO CDMA in Fast Fading Environments," in *IEEE Globecom*, Dec. 2004.
- [9] P. Radosavljevic, "Channel Equalization Algorithms for MIMO Systems and ASIP Architectures," Master's thesis, Rice University, April 2004.
- [10] J. Heikkinen, J. Sertamo, T. Rautiainen, and J. Takala, "Design of Transport Triggered Architecture Processor for Discrete Cosine Transform," in 15th Annual IEEE International ASIC/SOC Conference, Sept. 2002.
- [11] L. Schumacher, J. Kermoal, and P. Mogensen, "I-METRA D2 Channel Characterisation," *3GPP I-METRA Project*, Oct. 2002.
- [12] W. Wolf, "A Decade of Hardware/Software Codesign," *IEEE Computer*, vol. 36, pp. 38–43, April 2003.
- [13] "MOVEGen User's Manual." Tampere University of Technology, January 2004.
- [14] "Xilinx ISE Foundation [Online]." Available: http://www.xilinx.com/ products.
- [15] "Leonardo Spectrum [Online]." Available: http://www.mentor.com/ leonardospectrum.
- [16] "IC Station [Online]." Available: http://www.mentor.com/cicd/ icstation.html.