A New Reconfigurable Modem Architecture for 3G Multi-Standard Wireless Communication Systems

Jina Kim and Dong Sam Ha
VTVT (Virginia Tech VLSI for Telecommunications) Lab
Department of Electrical and Computer Engineering
Virginia Tech, Blacksburg, VA 24061
E-mail: {jnkim, ha}@vt.edu

Jeffrey H. Reed
MPRG (Mobile and Portable Radio Research Group)
Department of Electrical and Computer Engineering
Virginia Tech, Blacksburg, VA 24061
E-mail: reedjh@vt.edu

Abstract – **The trend in communication systems is towards more rapidly changing specifications with shorter time intervals between updates of existing standards. This results in a coexistence of many diverse standards. To support this diversity, communication systems should have a flexible architecture for easy and rapid updates. An ASIC design approach is not suitable for this purpose. Reconfigurable computing machines (RCMs) provide a promising solution for multi-standard communication system design. The flexibility and scalability of RCMs can support multiple standards and easily adapt to rapid changes of specifications. In this paper, we present a reconfigurable modem (RM) architecture targeting 3G multi-standard wireless communication systems. The proposed RM achieves scalability and low circuit complexity, while obtaining the performance close to an ASIC.**

I. INTRODUCTION

Implementing a system with an ASIC offers the most optimized solution for a specific application. However, the trend in communication systems is towards more rapidly changing specifications with a shorter time interval between updates of existing standards. Resultantly, communication systems should have a flexible architecture for easy and quick updates of many diverse standards. The ASIC design approach is not suitable for this purpose. Programmable devices such as Field Programmable Gate Arrays (FPGAs), General-Purpose Processors (GPPs) and Digital Signal Processors (DSPs) address flexibility problems. However, such programmable devices often fail to meet the speed requirements of some communication standards. Further, high power consumption and possibly the high cost of these devices make them even less attractive.

Reconfigurable computing machines (RCMs) provide a promising solution for multi-standard communication systems. The flexibility and scalability of RCMs can support multiple standards and easily adapt to rapid changes of specifications. There have been several efforts to develop reconfigurable architectures for communication systems in both industry and academia. A reconfigurable platform to implement a base station modem chip for third-generation (3G) communication systems was presented in [1]. The architecture targets word level operations, and thus it is inefficient for bit level operations such as code generation and Viterbi decoding operations. A fractal architecture with four types of clusters was investigated in [2]. Each cluster performs different operations. Since the architecture provides a unified interface for different node types, it supports most physical layer algorithms of communication systems and is highly scalable. A reconfigurable computing block targeting handheld devices for 3G communication systems was suggested in [3]. It can handle both bit-level and word-level operations using functional units (FUs), but the resource utilization is low for a mixture of bit-level and word-level operations. A reconfigurable architecture for WCDMA mobile applications was addressed in [4]. It is well suited for data path-oriented operations, but homogeneous processing elements (PEs) degrade the resource utilization and complicate the routing and the control.

In this paper, we propose a reconfigurable modem (RM) architecture targeting 3G multi-standard wireless communication systems. Our target applications are handheld receivers for third generation wireless communication systems, specifically WCDMA and CDMA2000. Application specific customization in conjunction with reconfiguration achieves performance comparable to that of an ASIC.

The paper is organized as follows. Section 2 reviews the design procedure as it relates to the target applications. Section 3 describes the structures of PEs and Section 4 provides the overall architecture of the proposed RM. Section 5 presents the performance comparison, and Section 6 concludes the paper.

II. DESIGN PROCEDURE

Our RM architecture targets two 3G wireless communication standards, WCDMA and CDMA2000, and the design objectives are scalability, low power dissipation, and low circuit complexity. The following explains our design philosophy in implementing the major three modem blocks for the two standards: cell searchers, rake receivers and Viterbi decoders.

0-7803-8834-8/05/$20.00 ©2005 IEEE.

- *Targeting specific applications*: WCDMA and CDMA2000 employ QPSK modulation, so a suitable processing element (PE) structure should be able to execute both real and imaginary computations simultaneously. Another requirement is that the code generation, despreading, and add-compare-select (ACS) operations make up a large proportion of the target applications. Code generation and despreading operate at the chipping rate and are performed most frequently, so they require the highest clock frequency. The Viterbi decoder performs many ACS operations. Hence, the PE structures are tuned to efficiently implement the code generation, despreading, and ACS operations. Finally, data flows from the chipping rate blocks to symbol rate blocks. Thus, PEs are grouped into chipping rate blocks and symbol rate blocks. Since the characteristics of the target application affect the entire RM architecture as well as the structures of the PEs, our RM architecture and the internal PE structures are carefully tuned for the target application. Note that the other existing designs such as [3], [1] and [4] are not as application specific.
- *Scalability*: We adopted several levels of hierarchy and an identical bit-width for all PE interfaces to enhance scalability. The hierarchical structure simplifies controller operations, and the identical bit-width provides a uniform interface between PEs.
- *Low power dissipation and low circuit complexity*: The multiply-and-accumulate (MAC) operation results in complex hardware and large power dissipation. We allocated only one PE for MAC operations, and its performance is sufficient to handle all MAC operations. Allocation of only one PE, which is shared between different blocks in need of MAC operations, saves power and area. It is another distinction from some existing designs such as [3] and [4].

Based on the above design approach, we designed FUs for the three function blocks: cell searcher, rake receiver, and Viterbi decoder. The following list includes the required FUs for each function block.

- *Cell searcher*: Registers, adders, comparators, and multipliers
- *Rake receiver*: Registers, counters, logic gates, LFSRs, adders (subtractors), accumulators, multipliers, shifters, and comparators
- *Viterbi decoder*: Registers, adders, and comparators

Based on the above analysis, we designed four different types of PEs which consist of multiple FUs. Each PE is dedicated for one of four basic operations, such as Bit Manipulation (BM), one-bit correlation (OBC), MAC, or ACS. At the next level in the hierarchy, PEs are grouped into four different types of PE modules. A PE module provides temporal and spatial data locality due to short reconfigurable data paths between consecutive operations. Next, PE blocks consist of multiple PE modules and local memories. Finally, we designed the top level architecture of the RM.

III. PROCESSING ELEMENT STRUCTURES

The four basic operations for our RM are *BM, OBC, MAC,* and *ACS*, and necessary FUs are assembled to form a PE dedicated for a basic operation. The interconnections between FUs are reconfigured to optimize the required operations temporally and spatially. As noted above, the bit width of all data paths in PEs are identical for improved scalability. The following describes the design of four PEs.

A. Type PE_A: Bit Manipulation

A PE_A supports BM operations such as code generation and logic operation. Code generation is a core operation of a cell searcher and a rake receiver. As depicted in Figure 1, a PE_A generates both the real and imaginary portions of the code simultaneously. Each portion has a register, a counter, two configurable LFSRs (CLFSRs) [4] and five 2-input logic gates. PE_A is capable of generating both the channelization code and the scrambling code with one reconfiguration, and this consequently minimizes the reconfiguration overhead in the code generation.

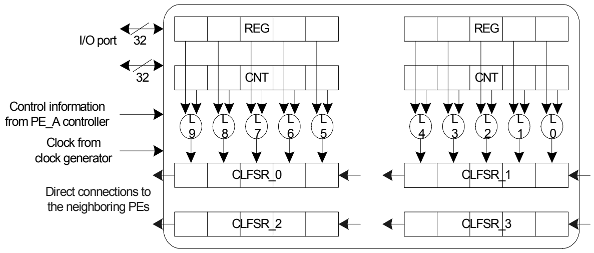
I/O port
32
32
Control information from PE_A controller
Clock from clock generator
Direct connections to the neighboring PEs
REG
CNT
L 9
L 8
L 7
L 6
L 5
L 4
L 3
L 2
L 1
L 0
CLFSR_0
CLFSR_1
CLFSR_2
CLFSR_3

Figure 1. PE_A structure

B. Type PE_B: One Bit Correlation (OBC)

A PE_B targets the despreading operation, which is the most frequent operation in a cell searcher and a rake receiver. Correlation of the input data with the one bit code sequence accomplishes despreading. Similar to a PE_A, a PE_B consists of two internal parts, which generate correlation of real and imaginary parts simultaneously. As shown in Figure 2, each part is composed of two 3-input XOR gates, an adder (or subtractor), a register, and a shifter.

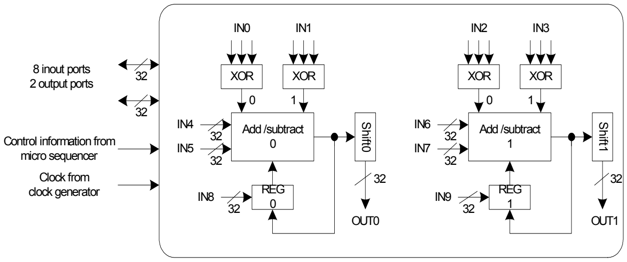

Figure 2. PE_B structure

C. Type PE_C: Multiplication and Accumulation

Figure 3 shows the FUs of a PE_C, which supports the MAC operations necessary for a cell searcher. A PE_C also supports compensation and power measurement for a rake receiver. In addition to the FUs, four internal registers, a status register, and a one's counter are inserted into a PE_C. Internal registers store intermediate computational values, the status register keeps the status of branch conditions, and the one's counter counts number of 1's in a word. These extra FUs are provided for functions beyond MAC operations.

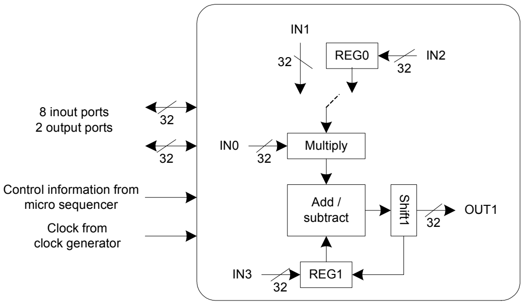

Figure 3. PE_C structure – Core FUs

D. Type PE_D: Add, Compare, and Select

Figure 4 illustrates the structure of a PE_D, which performs ACS operations, a core operation of a Viterbi decoder. The time tracker of a rake receiver and the peak detector of a cell searcher also employ a PE_D. Although we optimized the PE_D structure for the Viterbi decoder in our RM, an ACS operation can also be used for other operations, such as additions and subtractions. Like the PE_C structure, a PE_D includes adders, comparators, and multiplexers for ACS operations. Some additional FUs cover other operations, if necessary. Four internal registers store intermediate values, a status register processes conditional statements, and a 1's counter performs distance calculation for a branch metric.

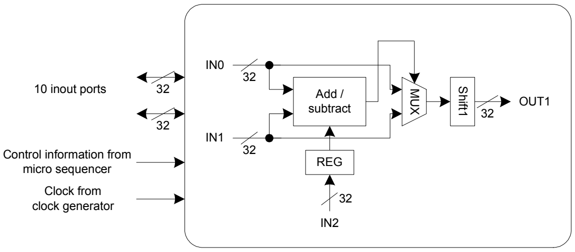
10 inout ports
32
IN0
IN1
Add / subtract
MUX
Shift1
OUT1
REG
IN2
Control information from micro sequencer
Clock from clock generator

Figure 4. PE_D structure – Core FUs

IV. OVERALL ARCHITECTURE

The four types of PEs in the previous section form the bottom level of our RM hierarchy. Four identical PEs are grouped into a PE module, which is the second level in the RM hierarchy. PEs inside a PE module communicate with each other using reconfigurable bidirectional data paths.

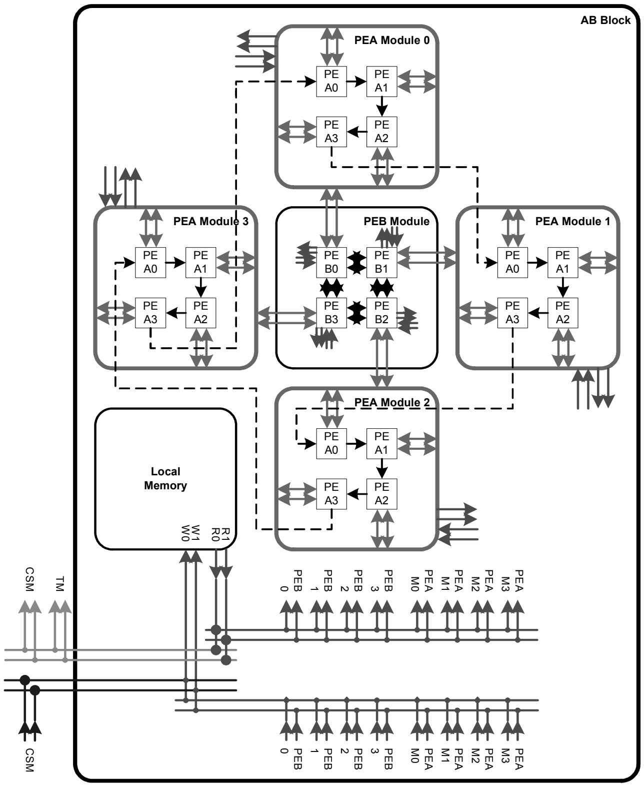

Figure 5. Structure of AB block

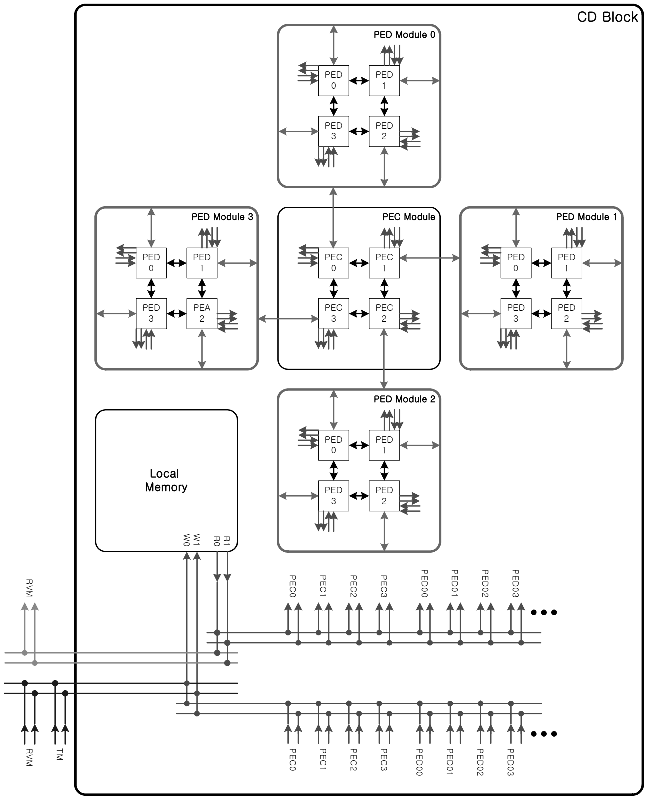

Figure 6. Structure of CD block

PE blocks form the third level in the hierarchy. One PE_A module, four PE_B modules and a local memory comprise an AB block. A CD block is comprised of four PE_C modules, a PE_D module, and a local memory. The major operation of AB blocks is despreading, and that of CD blocks is the Viterbi decoding operation. Figures 5 and 6 show an AB block and a CD block, respectively. The reconfigurable data paths between PEs, between PE modules, and between PE modules and a local memory are depicted. Directions of the data paths can be set individually independent of other data paths. A data path can also be in idle mode. The structure of the AB block corresponds to that of the CD block. The major difference between an AB block and a CD block is the existence of direct data paths between PE_As in different PE_A modules for an AB block. These paths are intended to configure a longer LFSR for long code generation.

The overall RM architecture, as shown in Figure 7, consists of PE blocks, an input buffer, a front-end memory, a transfer memory, and a back-end memory. AB blocks perform chipping rate operations and CD blocks conduct symbol rate operations. Our RM relies on a memory, instead of buses, to reduce the power dissipation of data transfers from AB blocks to CD blocks. The front-end memory and the back-end memory store the intermediate computation results.

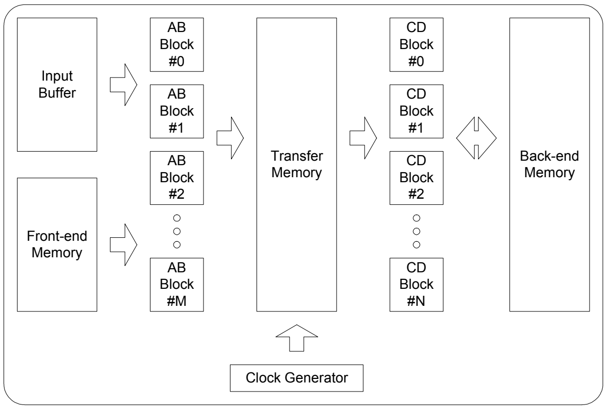

Figure 7. Overall architecture

V. PERFORMANCE COMPARISON

To estimate the performance of our RM and compare it with an ASIC approach, we implemented a rake receiver on the proposed RM architecture and in an ASIC environment. The performance metrics are circuit complexity and critical path delay. To estimate the circuit complexity and the critical path delay, we used Synopsys Design Compiler with Artisan's TSMC 0.18 μm CMOS library.

The rake receiver has four simultaneous physical channels, and each channel has six fingers. Each finger consists of despreaders, a channel estimator, a channel compensator, and a time tracker. In the RM implementation, the rake receiver utilizes 12 AB blocks and seven CD blocks. The local memory size is 64×32-bit, and the transfer memory size is 128×32-bit.

Table I summarizes the synthesis results. The estimated area for our RM is 6.9 M equivalent NAND2 gates, while that for ASIC version is 5.5 M NAND2 gates. This is mainly due to the use the identical bit width (which is set to the maximum) for our RM design, whereas the ASIC version optimizes the bit width for every functional block. If we optimize individual bit widths, the circuit complexity of our RM will be lowered significantly at the cost of scalability. The proposed RM architecture has a critical path delay of 13.0 nsec and that for the ASIC implementation is 7.5 nsec. The longer critical path delay of our RM is due to long data paths inside PEs. However, our RM design meets the required specifications of the target applications, which is 16.3 nsec. Thus, it should not be considered as a disadvantage for our RM. Although we did not complete simulations for power estimation, preliminary results indicate that the difference in power dissipation for the two designs is insignificant.

TABLE I. PERFORMANCE COMPARISON OF AN ASIC AND OUR RM

	Achieved performance	
	Circuit complexity (# NAND2 gates)	***Critical path delay (nsec)***
ASIC	5.5 M	7.53
RM	6.9 M	13.02

VI. CONCLUSION

In this paper, we presented a new RM architecture for 3G wireless communication systems, specifically WCDMA and CDMA2000. Our design approach intends to address drawbacks of other existing RCMs targeting wireless communication systems. To obtain comparable performance to an ASIC, we tailored our RM architecture to accommodate the specific target applications. We identified four key operations, BM, OBC, MAC, and ACS, of the modem, and our RM architecture adopts a hierarchical design and a uniform data bandwidth to improve scalability. We implemented a rake receiver with our RM and in an ASIC environment. Our RM increases the area by only 25 percent, and both designs meet the speed requirement. The area increase for our RM is well justified considering the advantages of our RM such as flexibility and scalability. It should be noted that the area of our RM can be reduced easily through optimization of the data path widths.

REFERENCE

[1] V. Baumgarte, F. May, A. Niickel, M. Vorbach, and M. Weinhardt, “PACT XPP – A self-reconfigurable data processing architecture”, Proceedings of ERSA'03, pp. 242-251, June 2003.

[2] www.quicksilvertech.com, Quicksilver technology.

[3] P.J.M. Havinga, L.T. Smit, G.J.M. Smit, M. Bos, P.M. Heysters, “Energy management for dynamically reconfigurable heterogeneous mobile systems”, Heterogeneous Computing Workshop, April 2001.

[4] Ahmad M. Alsolain, “Dynamically reconfigurable architecture for third generation mobile systems”, Ph.D. dissertation Ohio University, August 2002.