An Energy-Efficient Reconfigurable Baseband Processor for Flexible Radios

Ada S. Y. Poon
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Email: poon@uiuc.edu

***Abstract*— Most existing techniques for reconfigurable processors focus on the computation model. This paper focuses on increasing the granularity of configurable units without compromising flexibility. This is carried out by matching the granularity to the degree-of-freedom (DOF) processing in most wireless systems. A design flow that accelerates the exploration of trade-offs among various micro-architecture for the configurable unit is discussed. A prototype processor is implemented using the Intel 0.13 μm CMOS low-power standard cell library. The estimated energy efficiency is on the same order of dedicated hardware implementations.**

I. Introduction

The popularity of wireless communication leads to the proliferation of many air interface standards such as the IEEE 802 families and the 3G families. This evolution of standardization continues in an accelerated manner. Flexible radio architecture that can support not only multiple standards but also upcoming ones, becomes an important research area. For the digital part of the radio, domain-specific reconfigurable processors offer the advantage of flexibility as general-purpose processors, and low-power by exploiting parallelism in the baseband algorithms and providing a direct spatial mapping from algorithms to architecture, and hence reducing the memory and control overhead associated with general-purpose processors. Most existing techniques focus on the computation model of the reconfigurable processor, for example, ADRES [1], RaPiD [2], MorphoSys [3], RAW [4], MATRX [5], and PADDI [6]. They, in general, compose of an array of heterogeneous coarse-grained configurable units controlled by either RISC (reduced instruction set computer), VLIW (very long instruction word), or both instruction sets. The granularity of configurable units is usually a word-level operation such as a multiplier, an ALU, or a register.

As the granularity of configurable units directly impacts the energy efficiency of the hardware, this paper focuses on increasing the granularity of the configurable units without compromise flexibility. This is carried out by matching the granularity to the degree-of-freedom (DOF) processing in wireless systems. A wireless channel is built upon multiple signal dimensions: time, frequency, and space (antenna array) [7]. The sophistication of a transceiver is measured by its resolvability along these dimensions. For example, a system with bandwidth of W, transmission inteval of T, and number of antennas N has a resolution of $2WTN$ degrees of freedom. Baseband algorithms are collectively performing channel estimation over these degrees of freedom, and modulation (demodulation) of data symbols onto (from) these degrees of freedom such as DS-CDMA, OFDM, and space-time coding schemes. We abstract operations that are typically performing as per degree-of-freedom and put them into a single dominant configurable unit. In our design, this unit consists of 4 multipliers, 5 adders, 2 accumulators, 2 shifters, 8 two's complement operations, and 2 multiplexers. We believed that it is the largest grain in the literature. To ease programming, all pairs of inputs and outputs of the unit has at most one clock cycle of latency. Meeting these specifications as well as achieving the throughput requirement require a design flow that accelerates the exploration of trade-offs among various micro-architecture for the configurable unit. We use the Chip-in-a-Day design flow developed at U. C. Berkeley [8].

The reconfiguration mechanism is similar to RaPiD [2]. Control signals are divided into *hard* control and *soft* control. The hard control signals are similar to those in an FPGA and change infrequently. The soft control signals are similar to those in microprocessor and change almost every clock cycle. The hard control signals have fixed-length instructions, while the soft control signals have variable-length instructions as only a small portion of them is used in any computation.

Finally, a prototype of the processor is implemented using the Intel 0.13 μm CMOS low-power standard cell library. It consists of an array of 9 configurable units. Among them, four of them are the dominant DOF units, two of them are interconnect units, and the remaining three are accelerators for coordinate transformation, maximum-likelihood detection, and miscellaneous arithmetic operations. The interconnect units are specifically designed to support pipeline and stream processing so as to further enhance the energy efficiency. It utilizes a time-multiplexed cross-bar architecture to substantially reduce the amount of wires. Therefore, the entire baseband processor is a multi-rate system. The interconnect and the memory are running at 200 MHz while most of the configurable units are running at 50 MHz. The total gate count is 569e3 and the estimated power consumption is 63.4 mW. The energy efficiency is 95 MOP/mW which is on the same order of dedicated hardware implementations.

The organization of the paper is as follows. Section II evaluates various baseband algorithms and presents the granularity of configurable units. Section III summarizes the Chip-in-a-Day design flow. Section IV describes the micro-architecture

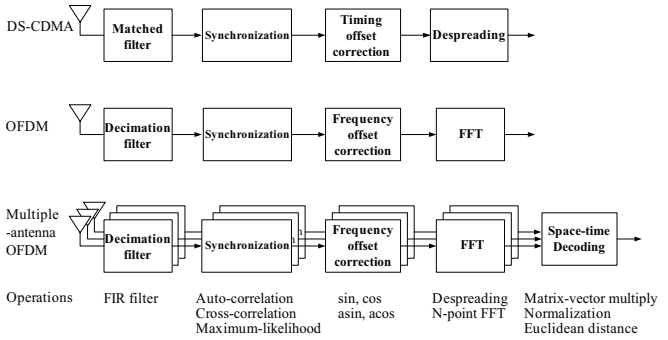

Fig. 1. Block diagrams of DS-CDMA, OFDM, and MIMO OFDM receivers.

of each configurable unit. Section V gives an overview of the programming model. Finally, we will conclude this paper in Section VI.

II. Choice of Granularity

Fig. 1 shows the degree-of-freedom processing (also known as symbol processing) of three popular receiving systems: DS-CDMA, OFDM, and multiple-antenna (MIMO) OFDM. DS-CDMA is part of the 3G celluar standard and the IEEE 802.11b wireless LAN standard. OFDM is part of the IEEE 802.11a/g wireless LAN standard and the IEEE802.16e wireless MAN standard. Multiple-antenna OFDM is part of the IEEE 802.11n high throughput wireless LAN standard.

In DS-CDMA systems [9], the incoming signal first goes through a matched filter and then correlates with either its delayed replica or a training sequence during synchronization. Once a signal is detected, the estimated timing offset will be compensated. Afterwards, the signal de-spreads with the PN sequence to obtain modulated data symbols. Similar to DS-CDMA systems, the incoming signal of OFDM systems [10] first goes through a decimation filter and then followed by either auto-correlation or cross-correlation for synchronization. Upon detection of signal, the estimated frequency offset is corrected and then data symbols are demodulated by the FFT operation. Finally, the multiple-antenna OFDM system is composed of a parallel of several OFDM systems. The demodulated symbols after the FFT operation is combined in accord with the space-time coding schemes used such as the V-BLAST [11] and the SVD-based algorithms [12].

The bottom of Fig. 1 summarizes the operations involved. To support all the operations by an array of homogeneous configurable units would make the configurable unit too bulky. Instead, we group the most frequent and similar operations together to be supported by an array of dominant units. Each unit is called the DOF configurable unit. All remaining operations will be supported by three other configurable units: a CORDIC (coordinate rotation digital computer), a maximum-likelihood (ML) accelerator, and a dual-core ALU. Table I summarizes the classification of operations. As a whole, Fig. 2 shows the architecture of the reconfigurable baseband processor. In the prototype implementation at 0.13 μm CMOS, the number of DOF units is 4.

TABLE I

Granularity of configurable units and operations supported.

Configurable units	Operations supported
DOF (dominant)	FIR filter Auto-correlation Cross-correlation Matrix-vector multiply De-spreading Euclidean distance calculation N-point FFT
CORDIC	Sine, Cosine Asin, Acosine Normalization
ML	Maximum likelihood
ALU	Miscellaneous mathematic operations Asynchronous control

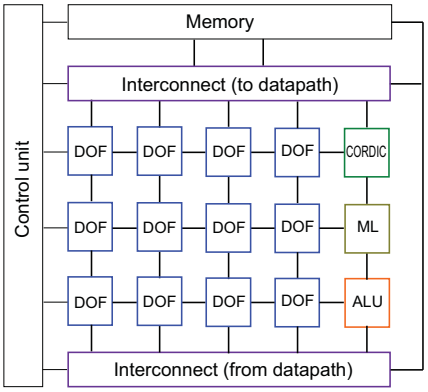

Fig. 2. Macro-architecture of the reconfigurable baseband processor.

The receiver chains shown in Fig. 1 also reveals the stream-based processing of most baseband algorithms in wireless communication. The better we could preserve this property in the implementation, the more power efficient the processor will be. The power efficiency is derived from maintaining data locality. In our interconnect configurable units, the outputs from the array of datapath configurable units can be configured to either feeding back to the memory or being inputs to the array. This is illustrated in the right part of Fig. 2 where the interconnect to datapath has inputs from memory or from the outputs of datapath. To get an idea, the interconnect units are designed such that they can be configured into a pipelined datapath illustrated in Fig. 3.

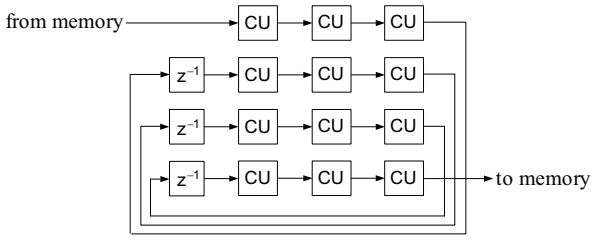

Fig. 3. Illustrate stream processing supported by the interconnect configurable units. In the diagram, the CU blocks refer to the datapath configurable units.

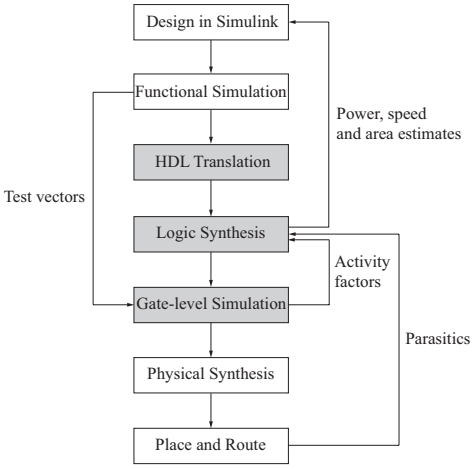

Fig. 4. Illustrates the ASIC design flow.

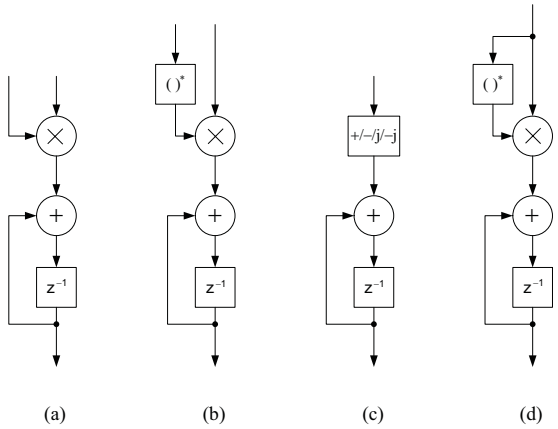

Fig. 5. Illustrates the architecture of (a) FIR filter, (b) auto-correlation and cross-correlation, (c) despreading, and (d) Euclidean distance calculation.

In summary, there are five different configurable units: DOF, CORDIC, ML, ALU, and interconnect unit. Their micro-architecture will be discussed in Section IV. These micro-architectures are the result of many iterations of architectural trade-offs. To accelerate the exploration of these trade-offs, we use a CAD (computer-aided design) flow detailed next.

III. Design Flow for Architectural Trade-off Analysis

Fig. 4 summarizes all the levels in the design flow. The design entry is the block-based input description of MathWorks Simulink [13]. Fixed-point data types are used which yields a cycle-by-cycle and bit-accurate simulation for functional verification as well as generation of test vectors for later gate-level verification. The fixed-point blocks (adders, multipliers, registers, and multiplexers) in the Simulink model is then mapped to the corresponding modules of Synopsys Module Compiler [14]. The netlist described by the Simulink model together with the mapped modules are translated into a HDL (hardware description language) description that is compatible with Synopsys Design Compiler [15]. After that, logic synthesis is performed on the HDL description given the technology library. A gate-level netlist is generated. Based on test vectors produced from the functional simulation, gate-level simulation is performed which outputs the activity factors. The logic synthesis is repeated taking into account these activity factors. Estimation on the power, speed, and area are obtained. These estimates provide feedback for design comparison and analysis which allow us to explore alternative architectures quickly and choose a more optimal architecture given the design constraints. After many iterations, the desired design is then handoff to the physical synthesis and followed by the place-and-route.

IV. Architecture of Configurable Units

In this section, we will describe the detail architecture of each configurable unit and the trade-off involved in arriving at the current design.

A. DOF

The operations supported by the DOF configurable unit tabulated in Table I mostly have complex numbers as inputs. Therefore, we design the DOF unit as a complex datapath. This increases the granularity without compromise flexibility. The control overhead is then reduced by up to 75% as the same control signals are applied to up to four parallel operations. The number of memory access is reduced by 50% due to data locality.

Now let us look into possible architectures for each supporting operation. The FIR filter can be realized by the standard multiply-accumulate (MAC) unit as shown in Fig. 5(a). Both auto-correlation and cross-correlation are similar to the FIR filter except that an additional conjugate operator is needed at one of the inputs as shown in Fig. 5(b). The matrix-vector multiply operation is composed of a series of inner product operations which can be realized by either the architecture in Fig. 5(a) or (b). For the de-spreading operation, the multiplier can be replaced by a simpler operator that can rotate the phase of the input by $0°$, $90°$, $180°$, or $270°$ as shown in Fig. 5(c). Calculation of the Euclidean distance has the same architecture as the auto-correlation and cross-correlation except that now the two inputs are tight together shown in Fig. 5(d). All the architectures shown in Fig. 5 are very similar.

For the FFT operation, there are two computation methods: decimation-in-frequency (DIF) where additions are before multiplication and decimation-in-time (DIT) where multiplication is before additions [16] as illustrated in Fig. 6. Compared with the architecture in Fig. 5, the complex multiplier is in common at the least. We can cascade the DIF architecture with those in Fig. 5, and yield one of the two possible architecture for the DOF configurable unit shown in Fig. 7(a). The other possible architecture is by cascading the architecture in Fig. 5 with that of DIT as shown in Fig. 7(b) where both the complex multiplier and a complex adder are shared.

In Architecture I, seperate adders are used for the FFT operation and the accumulator, and only the multiplier is shared. Therefore, it has *simpler control* than Architecture II

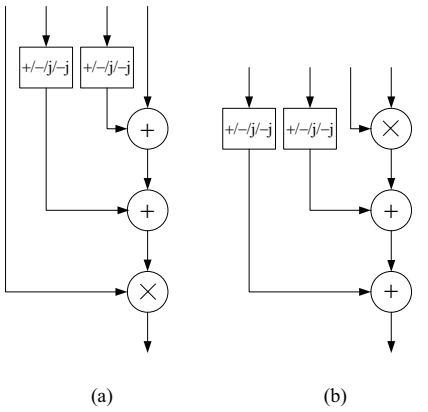

Fig. 6. Illustrates possible architecture for the FFT operation: (a) radix-4 decimation-in-frequency and (b) radix-4 decimation-in-time.

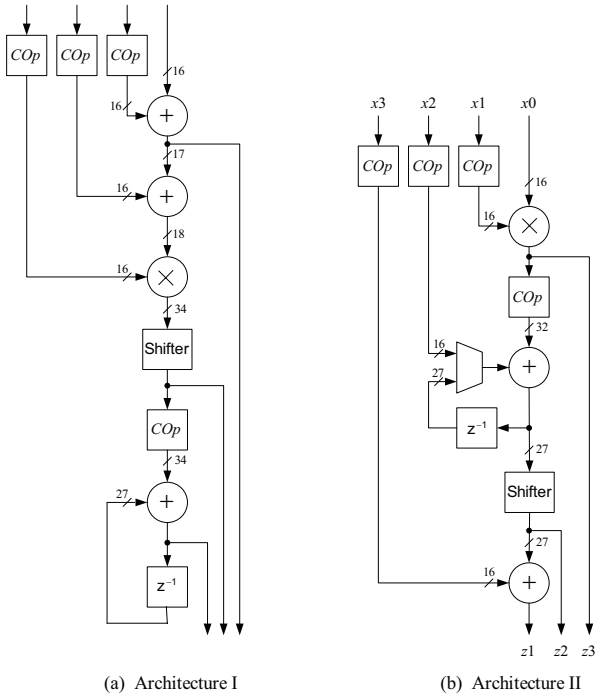

Fig. 7. Illustrates the two possible architecture for the DOF configurable unit. The operator COP is defined as: if x is the input to COP, the output can be selected as either x, $-x$, jx, $-jx$, jx^*, or $-jx^*$. The number shown is the bit width of the corresponding real and imaginary components. Inputs and outputs have bit-width of 16.

where both the multiplier and an adder are shared. To ease programming, all pairs of inputs and outputs are constrained to have at most one cycle of latency. This implies that the register in the accumulator is the only pipeline stage. Architecture I presumably has *longer delay* than Architecture II. To meet the throughput requirement, Arechitecture II apparently would yield a better trade-off. Even it has fewer components, it is composed of 4 multipliers, 5 adders, 2 accumulators, 2 shifters, 8 two's complement operations, and two multiplexers. Is it possible to run the entire configurable unit within one cycle of reasonable speed, say 50 MHz?

To answer this question, we implement Architecture II and pass the design to the CAD flow described in Section III where the design is synthesized using the 0.13 μm CMOS standard cell library. After a number of iterations, we arrive at the final design with all the bit-width defined as in Fig. 7(b). At the speed of 50 MHz, the estimated power consumption including clock tree is 6.55 mW and the estimated gate count is 16e3. The power estimate is equivalent to 180 MOP/mW. This includes the control logics for the 21 hard control bits and 4 soft control bits.

We were surprised that the entire DOF configurable unit can be implemented with only one cycle of latency and power efficiency on the same order of dedicated hardware implementations. It is because the output delay profile of an array multiplier matches with the input profile of a ripple adder, and the output delay profile of the ripple adder also matches with the input profile of another ripple adder. As a result, the difference in the delay between the multiply-add-add operation and the multiply-only operation is approximately equivalent to only 2-bit delay. Even array multipliers and ripple adders have longer delay than other implementations, the cascaded total delay is not increased by much. The power efficiency is then due to array multipliers and ripple adders being more power efficient than other implementations.

B. CORDIC

The CORDIC configurable unit implements the CORDIC algorithm [17]. It can be configured to perform either polar to rectangular transformation, or rectangular to polar transformation. The former transformation undertakes the various trigonometric operations while the later performs the normalization operation in Table I. The core building block is the CORDIC stage which is composed of adders and shifters. The output precision depends on the number of CORDIC stages used. Roughly, each additional stage gives an additional bit of precision. Now, there are three possible architectures:

- *Spatial multiplexing.* For an N-bit precision, we physically have N CORDIC stages connected one after another. The configurable unit then runs at the slowest allowable speed. This implementation has two advantages: (1) it is more power efficient as it runs slower; and (2) no local control and local memory are needed which simplify the design. However, this implementation occupies more area.
- *Time multiplexing.* There is only one CORDIC stage and is iterated N times. The configurable unit is needed to run at a much higher speed, and also it has control and memory overhead. Therefore, it is less power efficient. However, it gives an area efficient implementation.
- *Joint spatial-temporal multiplexing.* This is a hybrid of the first two approaches. We first factorize N into $N = N_1 N_2$. Then there are physically N_1 CORDIC stages connected one after another. These N_1 stages are iterated N_2 times to obtain the desired precision.

The hybrid approach apparently makes a better trade-off among power, speed, and area efficiency.

TABLE II

PARAMETERS OF THE 3 DIFFERENT CORDIC IMPLEMENTATION.

Architecture	N	N_1	Bit-width	Speed (MHz)
I	8	8	16	50
II	10	10	12	50
III	8	2	16	200

TABLE III

PERFORMANCE SUMMARY OF DIFFERENT CORDIC IMPLEMENTATION.

Architecture	Precision (%)	Area (gates)	Power (mW)
I	0.44	12316	5.2253
II	0.19	10755	4.8166
III	0.44	7950	5.6578

We evaluate three different implementations as tabulated in Table II where the bit-width refers to the input bit-width of adders and shifters inside the CORDIC stage. Architecture I and II belong to the spatial multiplexing approach while Architecture III uses the hybrid approach. Their functional performance and physical performances at 0.13 μm CMOS are summarized in Table III. As expected, Architecture III occupies the smallest area but is the least power efficient. While comparing Architecture I with Architecture II, we notice that the use of more CORDIC stages but each having less precise arithmetic units yields better output precision, area estimate, and power estimate. As our application domain is wireless communications, power efficiency is more critical, we therefore choose Architecture II for the CORDIC configurable unit.

C. Other Units

Any miscellaneous operation is supported by two general-purpose 16-bit ALUs running at higher speed. This unit should be very flexible and therefore we use the the MIMD (multiple instruction stream, multiple data streams) approach and shared memory. Fig. 8 shows a block diagram of the architecture. For better computation efficiency, we choose CISC (complex instruction set computers) as the instruction set architecture. Currently, there are 19 opcodes: left shift, right shift, absolute value, addition, subtraction, increment, decrement together with 6 compare operations and 6 logical operations. The instruction length for each ALU is 14 bits.

The ML configurable unit is designed to accelerate the maximum-likelihood detection. Its arechitecture is shown in Fig. 9. It is the simplest among all the configurable units.

The processor has two memory banks: data memory and coefficient memory. The data memory has simultaneously read and write ports, while the coefficient memory has a read/write port. To reduce the amount of wiring, outputs from the data memory, from the coefficient memory, from the external data port, and from the configurable units are individually time-multiplexed. The time-multiplexed signals are then demultiplexed at inputs of configurable units. We

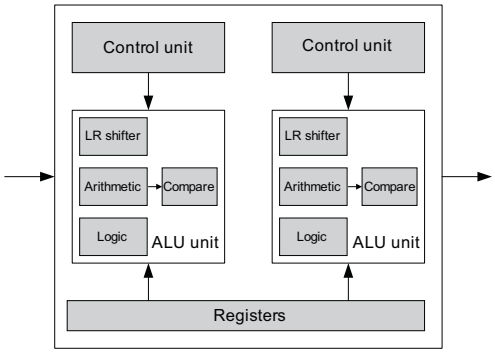

Fig. 8. Block diagram of the Dual-core ALU.

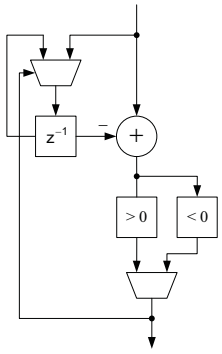

Fig. 9. Schematics of the ML accelerator.

coin this interconnect structure as a *time-multiplexed cross-bar interconnect*. It provides a compact interconnection, but it requires extra circuitry for the multiplexing and demultiplexing of signals, and also it incurs one cycle of latency.

Finally, Table IV summaries the performance of the prototype processor synthesized using the Intel 0.13 μm CMOS low-power standard cell library. The power estimates have included the clock tree. The processor has 4 DOF configurable units. Both data and coefficient memories are 8 KB. In addition, the control unit contains two 8 KB memories to store configuration bits and instructions for the dual-core ALU respectively. The energy efficiency is 95 MOP/mW which is on the same order of dedicated hardware implementations.

TABLE IV

PERFORMANCE SUMMARY OF THE PROTOTYPE PROCESSOR AT 0.13 μM CMOS.

Units	Speed (MHz)	Latency (cycles)	Area (gates)	Power (mW)
DOFs	50	1	4×15 843	4×6.55
CORDIC	50	1	10 755	4.82
ML	50	1	362	0.09
ALU	200	4	4 456	5.23
Interconnect	200	4	9 119	7.06
Memory & Control	200	4	482 282	19.40
Total			569 210	63.40

TABLE V
NUMBERS OF HARD AND SOFT CONTROL BITS.

Units	Hard Control	Soft Control
DOFs	4×21	4×1
CORDIC	1	0
ML	1	1
ALU	0	28
Interconnect	36	0
Memory	21	0
Total	183	33

V. COMPUTATION MODEL

The configuration bits are divided into hard control and soft control. The hard control bits change infrequently over time. For example, if the processor is configured to perform the function FFT, the hard control bits will not change during the entire execution of the funtion. The soft control bits, on the other hand, change frequently, for example, instructions of the dual-core ALU. At the beginning of every function, a fixed length instruction is issued to specify the hard control bits and also to specific which portion of soft control bits will be issued. During the execution of the function, that portion of soft control bits is updated every cycle by a variable length instruction. This separation of control signals therefore supports a more efficient representation of programs. In particular, the number of hard control bits are far more than the number of soft control bits. To get an idea of the difference, Table V summaries the control signals in all of the configurable units and the memory. For the prototype processor, there are a total of 216 control bits where 183 of them are hard control bits and 33 of them are soft control bits.

The baseband processor improves the energy efficiency by introducing parallelism in the spatial domain. But joint spatial-temporal programming is a hard problem. In particular, the sequential to parallel compiler problem has not been solved satisfactorily. Instead, we use a spreadsheet format to program the processor. The horizontal axis of the spreadsheet lists the configuration of all units in the processor while the vertical axis lists the time of execution. The spreadsheet is then compiled to instructions which are then downloaded to the memory inside the control unit for execution.

VI. CONCLUSIONS

In this paper, we introduce an energy-efficient reconfigurable baseband processor. The energy efficiency is obtained by utilizing coarse-grain configurable units. We have programmed the processor to run the radix-4 FFT, two streams of simultaneous radix-2 FFT (for systems with two antennas), the FIR filters, and the synchronization algorithm. In the synchronization algorithm, the DOF units are configured to perform the cross-correlation. Their outputs are passing to the ML accelerator. The output from the accelerator is then passing to the dual-core ALU. An exception will be issued by the ALU to the control unit when packet is detected. Thus, the dual-core ALU not only supports miscellaneous arithmetic operations but also provides a mechanism for asynchronous control to offload the control unit. We believe that such a coarse-grain architecture could be embeded in wireless devices to provide flexibility to an otherwise fully dedicated hardware solution, or to improve the overall energy efficiency of a purely general-purpose processor solution.

Through the architectural design process, we use an approach that utilizes immediate feedback from the underlying physical implementation. This is particularly critical to us as the dominant configurable unit (DOF) has a very aggressive specification. This approach not only help explore more optimal architecture but also accelerates the implementation of the design. For the prototype processor, it is a multi-rate system with more than half a million gates. We finished it in less than a year, from the architectural design to the tape-out of the prototype chip. We believed that such a design methodology is useful in designing energy-efficient wireless systems.

ACKNOWLEDGEMENT

The author would like to thank Professor Robert Brodersen for the discussion on various aspect of the processor architecture and the RSS team of the Intel research for supporting the tape-out of the prototype chip.

REFERENCES

[1] B. Mei, A. Lambrechts, and D. Verkest, "Architecture exploration for a reconfigurable architecture template," *IEEE Design & Test of Comput.*, vol. 22, no. 2, pp. 90–101, Mar. 2005.
[2] C. Ebeling, C. Fisher, G. Xing, M. Shen, and H. Liu, "Implementing an ofdm receiver on the RaPiD reconfigurable architecture," *IEEE Trans. on Comput.*, vol. 53, no. 11, pp. 1436–1448, Nov. 2004.
[3] H. S. et al., "MorphoSys: An integrated reconfigurable system for data parallel and computation-intensive applications," *IEEE Trans. on Comput.*, vol. 49, no. 5, pp. 465–481, May 2000.
[4] E. Waingold et al., "Baring it all to software: Raw machines," *IEEE Trans. on Comput.*, vol. 53, no. 11, pp. 1436–1448, Nov. 2004.
[5] E. Mirsky and A. DeHon, "MATRIX: a reconfigurable computing architecture with configurable instruction distribution and deployable resources," in *Proc. IEEE Symp. FPGAs for Custom Computing Machines*, Napa Valley, CA, Apr. 1996, pp. 157–166.
[6] D. C. Chen and J. M. Rabaey, "A reconfigurable multiprocessor ic for rapid prototyping of algorithmic-specific high-speed dsp data paths," *IEEE J. Solid-State Circuits*, vol. 27, no. 12, pp. 1895–1904, Dec. 1992.
[7] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.
[8] http://bwrc.eecs.berkeley.edu/Research/Insecta/default.htm.
[9] A. J. Viterbi, *CDMA Principles of Spread Spectrum Communication*. Addison-Wesley, 1995.
[10] J. Heiskala and J. Terry, *OFDM Wireless LANs: a Theoretical and Practical Guide*. Sams, 2002.
[11] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Tech. J.*, pp. 41–59, Autumn 1996.
[12] A. S. Y. Poon, D. N. C. Tse, and R. W. Brodersen, "An adaptive multi-antenna transceiver for slowly flat fading channels," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1820–1827, Nov. 2003.
[13] http://www.mathworks.com/products/simulink/.
[14] http://www.synopsys.com/products/datapath/datapath.html.
[15] http://www.synopsys.com/products/logic/design_compiler.html.
[16] A. V. Oppenheim, *Discrete-time Signal Processing*. Prentice hall, 1989.
[17] H. Dawid and H. Meyr, "CORDIC algorithms and architectures," in *Digital Signal Processing for Multimedia Systems*, K. K. Parhi and T. Nishitani, Eds. Marcel Dekker, 1999, pp. 623–655.