RECONFIGURABLE PLATFORM DESIGN FOR WIRELESS PROTOCOL PROCESSORS

Tim Tuan, Suet-Fei Li, Jan Rabaey

Berkeley Wireless Research Center
University of California at Berkeley
{timt, suetfei, jan}@eecs.berkeley.edu

ABSTRACT

Low-energy protocol processing is a crucial issue in next-generation wireless systems. In modern wireless system design, this problem is tightly coupled with the signal processing needs. Fierce market competition and inventive wireless applications are imposing stricter design requirements in energy consumption, cost, size, and flexibility. To deal with these unique constraints, we incorporate the platform-based design methodology to deal with these constraints by advocating reusability. This paper presents this methodology, and its application on PicoRadio, a cutting-edge wireless system. In particular, we describe the design of a reconfigurable architecture optimized for protocol processing.

1. INTRODUCTION

Communication protocol processing is an increasingly important problem in the design of next-generation, wireless systems. The real-time nature of the protocol stack, and its tight coupling to the base-band processing, make this problem very relevant and applicable to digital communication signal processing. This paper focuses on reconfigurable protocol processor design for low-power wireless embedded systems.

Due to application requirements and severe market demand, the protocol processor has become a bottleneck in the design of today's communication systems. In recent years, a number of wireless standards, such as Bluetooth [1], 802.11 [2], and HomeRF [3], have emerged. These standards will soon become prevalent in embedded systems for numerous applications, such as cellular phones, home automation, and digital audio players. A common problem shared by all the standards is the need for low-energy, high-throughput protocol processing. Similar problems are seen in the network processor community, where the demand for high-speed Internet calls for the design of complex protocol processing engines [4].

At UC Berkeley, we are building an ultra-low energy, sensor-based, ad hoc network called PicoRadio [5]. The targeted power consumption for each node in the network is under 100μW to enable self-powering via energy scavenging [6]. The node must be smaller than 100mm^3, weigh less than 100g, and cost less than one dollar. Consequently, the protocol processor must be extremely energy efficient, small, and low-cost. Furthermore, the processor should provide the flexibility to accommodate different protocol designs, which is necessary for testing purposes as well as adapting to changing network traffic.

Using traditional design methodologies, meeting such tight constraints would require a very lengthy design process, which would damage the product's marketability. Consequently, a new design methodology is needed to design highly optimized communication protocol processors in a short period of time. We believe that platform-based design methodology [7] is a promising solution.

Section 2 will describe the concept of platform-based design in the context of protocol processor design in detail. The central part of the paper presents our application of platform-based design methodology in the design of a low-power reconfigurable architecture for processing the lower layers of PicoRadio protocol stack. In Section 3, we will present profiling of the protocol stack to identify its key functionalities and properties. In Section 4, we present the architecture design in the scope of platform-based design.

2. PLATFORM-BASED DESIGN

The need for shorter design time and greater design complexity has made it necessary to look to new design methodologies that support design reuse. Platform-based design supports design reuse by abstracting hardware to a higher level, or the system platform. Its three-phase flow is illustrated in Figure 1. The first step is the identification of a **system platform**. A system platform consists of a family of parameterizable architectural modules for computations and interconnects. The system platform is visible to the application software, allowing application programming to be done in software. The second step is platform instantiation. Given a system platform, to derive a system that supports the application, the designer would instantiate a particular platform by choosing a subset of system platform that best

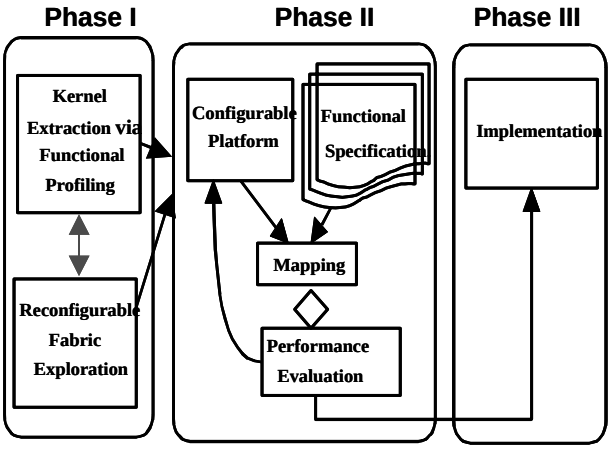

Figure 1: **Three-phase Design Methodology.**

supports the application. The final s tep is implementation of the system, which integrates and programs these architectural modules to perform the desired functions.

The challenges in making effective use of platform - based design methodology involve exploration of the architectures to see which best support the applications. This process involves two façades: First, we need to identify the key functions (kernels) in the target application set. Using profiling techniques, we can extract a set of kernel functions that represent the most costly - either computationally intensive or energy consuming - portion of the applications. A good platform must effectively support the key functions of the application; therefore, understand ing these functions is crucial to designing a good system using platform-based design.

The second façade is the exploration of architectural modules to gain insight on which modules best suit which functions. In phase I, we identify a set of possible architectures for the target applications. This may include both existing architectures as well as possibly new architectures. The architectures should be parameterizable to provide some degrees of freedom when system integration takes place. In phase II, we ex plore how effectively the kernel functions are supported by these architectures. To assist in performance evaluation, each architecture should have estimation models that provide first-order performance numbers for a given function. To allow final implementation of the system, characterization of the architectures should include means of programming.

3. FUNCTIONAL KERNELS OF WIRELESS PROTOCOL STACKS

Functional profiling te chniques [8] are applied to obtain the key operations in protocol processing (see Figure 2). The operations are loosely classified as either control or data processing dominated operati ons according to the

Layers	*Control Dominated*	*Data Dominated*
Application Transport	Localization Algorithms	Encryption; Decryption; Compression; Decompression
Network	Topology management; Routing & forwarding table lookup; Classification (Pattern matching; Parsing; Modification)	
MAC	Queue management; Channel assignment; Scheduling; Table lookup	
Physical	Synchronization; Timers; Segmentation; Reassembly	CRC/verification;

Figure 2: **Control/data kernels in wireless protocol stacks.**

nature of the algorithm. It should be emphasized this classification is implementation oriented and is quite different from the more traditional classification. In the latter case, “data plane” operations are directly on the feedforwar d path through the communication pipeline and have demanding real-time performance constraints, while control operations are not on this direct path and hence have looser timing constraints. Our classification is more relevant in the context of this paper since we are mostly concerned with implementation perspective of the protocols.

Based on our classification, an operation is data dominated if its complexity mostly comes from data processing. An appropriate model of computation for data processing is Dataflow [9]. On the other hand, an operation is control dominated if its complexity mostly comes from control structures. An appropriate model of computation in this case is finite state machine (FSM). It is apparent from Figure 2 that protocol processing is heavily control dominated and hence is described in some FSM based model. PicoRadio protocol stacks are described in the extended FSMs (EFSM), which are large networks of interacting FSMs with datapat h elements [10].

The classification of the functional kernels into control and data processing operations is very important for the realization of an efficient implementation. The different nature of control and data processing operations intuitively leads to different “optimal” implementation structures. An efficient architecture for protocol implementation should contain a mixture of these different implementation structures. The exact proportion of the different structures depends on the ratio between control and data in the application. In the following section, we will introduce a “hybrid architecture” that is based on this concept.

4. RECONFIGURABLE ARCHITECTURE

Before looking at specific architectures, we must first identify the key requirements of the applications. For PicoRadio, they are ultra -low energy consumption, small in area, low cost, and flexibility to design changes. Traditional computing architectures range from microprocessors to ASICs, but most fail to meet the energy requirement or the flexibility requirement. There also exist configurable processors [11] [12], but too often they are designed to specialize for applications other than communication protoc ol processing. This leaves us with the family of reconfigurable logic architectures, which are sufficiently low -level to allow low -energy circuit techniques, while providing some degree of flexibility through reconfiguration. Also, they are not limited to certain application sets.

4.1. Traditional Reconfigurable Architectures

Traditional reconfigurable architectures come in two flavors: field -programmable gate array (FPGA) and programmable logic device (PLD) [13]. FPGA and PLD architectures differ significantly in granularity. Using look-up table (LUT) technology, FPGAs can efficiently implement any arbitrary logic with few inputs. Since the LUTs are easily chained together to implement multilevel logic, this architecture is well suited for complex operations such as arithmetic and signal processing. On the other hand, the PLDs use programmable array logic (PAL) blocks that can each implement sum -of-product logic of many inputs but limited output. Thus, PLD structures are suitable for control FSMs. We performed experiments mapping benchmarks from the Two -Chip Intercom project [14] to commercial FPGA and PLD chips, and measured their utilization based on equivalent gate count. The results, shown in Figure 3, are consistent with the theoretical claims.

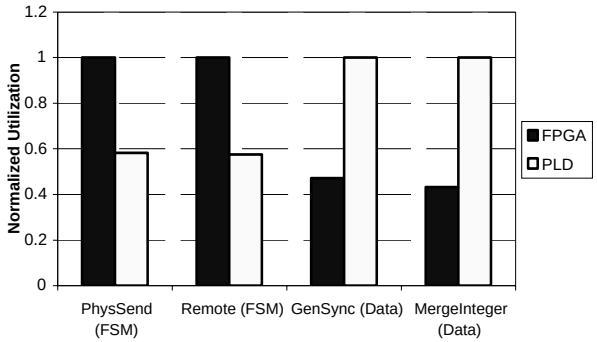

Figure 3: Implementation results of wireless protocol blocks.

4.2. Hybrid Architecture for Protocol Processing

As mentioned in Section 3, PicoRadio protocol stack takes the form of EFSMs. Consequently, we are constructing a reconfigurable architecture using both PAL and LU T blocks for control and datapath respectively. By utilizing each structure on functions that they are best suited for, we can achieve the best performance in the combined structure.

The architecture uses hybrid cells, each consisting of a small PAL block for control, and a small array of LUTs and flip-flops (FFs) for data processing. Figure 4 shows a block diagram of this architecture. Each cell in this structure comprises a PAL block and a small array of LUTs and FFs; thus, each cell corresponds to a sma ll FSM. Since protocols have many interacting FSMs, the architecture shall have an array of these hybrid cells.

4.2.1. Architecture Description

Figure 4 shows a detailed block diagram of a hybrid cell. Since the data processing elements are isolated in the LUT portion of the cell, the FSM must generate control signals that feed into the LUTs. The data inputs go directly to the LUTs, and the control inputs go directly to the PAL. Similarly, the data outputs come from the LUTs, and the control outputs come from the PAL.

Since control and data outputs, as well as any internal control signals may be used in the control plane, these signals must be fed back into the PAL block. Layout of the block should be considered carefully to minimize the lengths of these feedback signals.

4.2.2. Estimation Models

As mentioned in Section 2, to simplify performance evaluation of the architectures, an architecture should have estimation models that provide first -order performance

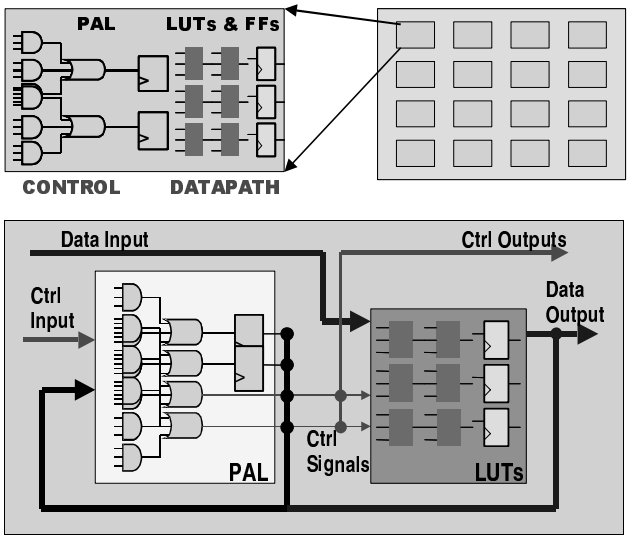

Figure 4: Basic block diagram of the hybrid architecture.

LUT Power Estimation

*LUT Power = (LUTs * 2.2uW) + (FF * 0.5uW)*

PAL Power Estimation

*PAL Power = (P-terms * Inputs * 0.05uW) + (Outputs * 0.7uW)*

Figure 5: Power estimation equations for LUT and PAL implementations.

numbers for a given function. Having estimation models greatly expedites the exploration process by preventing the need to physically map designs to the structures. For our hybrid architecture, we mapped basic designs to our LUT and PAL Spice models and monitored the amount of power used by each part of the archi tecture. Based on these results, we created first -order power models for LUT and PAL implementations. Although the accuracy of first-order estimations is limited, they are acceptable in the early stages of the design process.

From simulation of benchmarks , we obtained cost estimates and deduced a set of prediction equations for the power costs of FPGA and PAL structure for our benchmarks. Figure 5 shows these equations. The estimates are based on 0.25 µm technology on a 1.0V supply. Estimation of the data p ortion is based on the energy consumed by LUTs and FFs. Estimation of the control processor is based on a dynamic logic PAL implementation, which has significantly lower energy consumption than a traditional sense amp based implementation. Note that interc onnect power is not included, which is a degree of error that we allow for the sake of simplicity.

4.2.3. Performance Analysis

Using the power models, we can obtain first -order performance results to see how well the architecture works for our applications. The power models allow us to do this at an early stage, before any detailed implementation has taken place.

Using the *mac_a* design from the TCI project as benchmark, we estimated its power consumption under three different scenarios: purely FPGA implementatio n, purely PAL implementation and the proposed hybrid approach. The results, shown in Figure 6, suggest that the hybrid architecture out -performs the other two scenarios. We can expect even greater gain when the power dissipation of the PAL reduces as our r esearch in low -energy PAL matures.

5. CONCLUSION

We have introduced a design methodology for wireless protocol processor design using PicoRadio as the driver

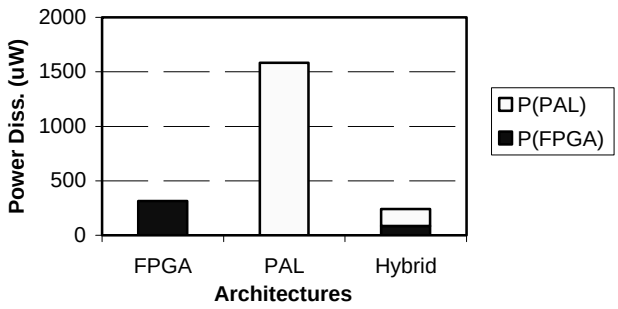

Figure 6: Power comparison of different architectures.

application. Our main focus is the exploration of a reconfigurable platform. A hybrid cell architecture is presented to implement the intertwinement of control and data processing in the protocol stacks. Further study is required to devise the optimal hybrid architecture based on the share of control and data processing in the application specification.

6. REFERENCES

[1] The Bluetooth Special Interest Group http://www.bluetooth.com/.

[2] IEEE 802.11 Working Group for WLAN http://www.manta.ieee.org/groups/802/11/

[3] The HomeRF Working Group http://www.homerf.org/.

[4] M. Hathaway, "Building Next Generation Network Processors," Proceedings of Gigabit Ethernet Conference, pp. 310-319, 2000

[5] J. Rabaey et al. "PicoRadio Supports Ad Hoc Ultra -Low Power Wireless Networking," IEEE Computer, vol. 33, no. 7, pp. 42 -48, July 2000.

[6] R. Amirtharajah, A. P. Chandrakasan, "Self -Powered Signal Processing Using Vibration -based Power Generation", IEEE Journal of Solid State Circuits, vol. 33, no. 5, pp. 687-95, 1998.

[7] A. Ferrari and A. Sangiovanni -Vincentelli, "System Design: Traditional Concepts and New Paradigms," Proceedings of the Int. Conf. on Computer Design, Austin, Oct. 1999.

[8] J. Rabaey et al. "Challenges and Opportunities in Broadband and Wireless Communication Designs", Proceedings of ICCAD, San Jose, Nov 2000.

[9] E. Lee and T. Parks, "Dataflow Process Networks", Proceedings of the IEEE, vol. 83, no. 5, pp. 773-801, May 1995.

[10] L. Lavagno, A. Sangiovanni -Vincentelli & E. Sentovich, "Models of Computation for Embedded System Design", NATO ASI Proceedings on System Synthesis, Il Ciocco, Italy, 1998.

[11] H. Zhang et al, "1V Heterogeneous Reconfigurable Processor IC for Baseband Wireless Applications," Proceedings of ISSCC, pp. 68-69, 2000.

[12] Darren C. Cronquist et al, "Architecture Design of Reconfigurable Pipelined Datapaths," Twentieth Anniversary Conference on Advanced Research in VLSI, 1999.

[13] M. Smith, *Application Specific Integrated Circuits: Chapter 5.4. Altera MAX*, Addison-Wesley, 1997.

[14] Two Chip Intercom Project, http://bwrc.eecs.berkeley.edu/Research/Two Chip Intercom/.