An Energy-Aware Active Smart Card

Russell Tessier, *Member, IEEE*, David Jasinski, Atul Maheshwari,
Aiyappan Natarajan, Weifeng Xu, and Wayne Burleson, *Senior Member, IEEE*

***Abstract*—**
Despite recent advances in smart card technology, most modern smart cards continue to rely on card readers for power and clocking, creating a potential security gap. In this paper we present an energy-aware smart card architecture that operates using an embedded battery and crystal. This low-power VLSI system is continually active and provides enhanced security through periodic internal update when the card is detached from a reader. Our architecture achieves reduced power consumption by deactivating the majority of its circuitry, including an embedded microcontroller, for the vast majority of the card's lifetime. A proof-of-concept prototype implementation of the architecture has been developed including register transfer level and gate-level designs which have been synthesized to silicon. To permit extended operation for up to 18 months, critical design logic has been implemented using ultra-low power (adiabatic) circuit techniques.

I. INTRODUCTION

The standardization of smart cards has led to growing worldwide acceptance. Contemporary smart cards are used for a variety of applications including electronic commerce, identification and access control. While smart card applications vary broadly, current utility and security is generally limited by technology available in card readers. To date most smart card systems have relied on readers to provide vital system-level support for internal card power, clocking, and time awareness. A broad range of new applications, such as those requiring security updates via new encryption keys, necessitate periodic card activation.

For many smart card applications, the computational requirements of on-board digital circuitry may limit the lifespan of an embedded battery required for continuous operation. As a result, the development of a continually-active smart card requires special attention to card architecture and energy-saving design techniques to permit periodic activation. The size of contemporary batteries limits the continuous operation of the smart card since battery depletion remains a major issue. Consequently, the smart card must use little energy when detached from the reader.

To address this application-driven need, we have developed a flexible smart card architecture that combines selective deactivation techniques with ultra-low power circuit design to preserve battery lifespan. The digital circuitry of the smart card is logically split into two parts, one which is continually active and another which is active when the card is interfaced to a reader and for short bursts of time when it is away from the reader (periodically active). The periodically-active subsystem consists of an off-the-shelf microcontroller and security and I/O modules which allow for communication with an ISO 7816 standard [1] reader interface. The continually-active subsystem contains two parts, a counter and a content-addressable memory (CAM), which form a state machine and act as a timer for periodic card events.

To validate our system-level approach, a complete proof-of-concept digital prototype has been designed and simulated at register-transfer, gate, and transistor levels. The development of this prototype necessitated the investigation of several supporting VLSI technologies. A key requirement of our architecture is extremely low power consumption of the continually-active subsystem. To extend battery life, we have developed a new CAM architecture which uses ultra-low power adiabatic design techniques and implemented a fault tolerant counter using logic gates and flip flops from an existing adiabatic logic family [2]. The adiabatic CAM uses charge recycling via low-swing signalling on the match line to save energy. This result and associated row-based CAM fault tolerance allows extended, reliable operation. The fault tolerant counter is constructed from a library of adiabatic gates which use a single-phase sinusoidal power source. The interface between adiabatic and CMOS circuitry is isolated by registers facilitating integration with remaining circuitry. Circuit design techniques have been verified via layout and extensive simulation.

A layout of the resulting transistor level design had been created to allow for detailed power and performance analyses of critical portions of the design. To support a complete system, software mapping tools, including a commercial C compiler have been updated to target the architecture. Two security-based applications which benefit from continually-active behavior, a digital signature generator and an encryption key update routine, have been successfully implemented and tested. Our prototype implementation, implemented in 0.25 μm technology, operates continuously with an embedded battery lifetime of nearly 18 months. The use of adiabatic logic provides a 33% reduction in power for the CAM and a 93% power reduction for the embedded counter allowing for extended card usage.

The rest of the paper is organized as follows. In Section II we motivate our new smart card architecture and describe its potential application space. In Section III, relevant smart card technology is reviewed including cards which currently contain embedded batteries. The digital architecture of our energy-aware smart card is described in Section IV. In Section V we describe the design and implementation of our smart card prototype. Ex-

This work was supported in by part by the National Science Foundation under grants CCR-0081405 and CCR-9988238 and by a grant from Sharp Corporation. The authors are with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, 01003 USA. E-mail: tessier@ecs.umass.edu.

perimental results are presented in Section VI. Finally, Section VII concludes the paper and offers an assessment of results.

II. Motivation

Contemporary smart cards are used in increasingly secure environments that are vulnerable to potential adversaries. Current smart card application domains include financial transactions, physical access control, and transportation, among others [3] [4]. To protect sensitive information, these environments often require the use of data encryption to allow for secure communication and data storage. A limitation of these environments is the security of the private key which is used to encrypt critical information stored on the card and possibly transmitted to an external reader. If this private key is compromised, all information stored in and communicated from the card since the most-recent key change becomes vulnerable to decryption. Additionally, the system may become susceptible to the creation and encryption of forged information that is subsequently used to corrupt a card reader. As technology advances, the portability of smart cards may also make them vulnerable to other types of physical attacks [5], providing further motivation for frequent key updates.

To address the issue of time-dependent key vulnerabilities, *forward* encryption techniques are currently under development. These techniques use a series of private keys to encrypt or digitally sign data. Each private key is used for a period of time, replaced, and then destroyed. A single key (public or private) and timestamp are used at the destination to decrypt information [6] [7] created with any of the source keys. By varying the source private key over time and then destroying the old source keys after they expire, previously-encrypted data cannot be replicated or modified even if card security is compromised. Since both the smart card and associated reader systems are aware of key update frequency, fraudulently modified or created data can be quickly identified.

These types of security techniques address a number of the goals of smart card security. Even if the current private key stored on the smart card is discovered via exhaustive methods or other techniques, it is not possible to create or modify data associated with earlier time periods of card use, securing stored information (e.g. a financial balance, access information, etc). Periodic key update affords protection to secret information that is previously encrypted and stored on the card. Although these types of security protocols have primarily been applied to digital signature applications [6], more general encryption algorithms are now being considered [7].

The need for periodic key update motivates a new smart card architecture. A continually-active smart card architecture can periodically update one or more encryption keys, even when the card is not attached to a card reader. A key aspect of the new architecture is its time-aware nature; an accurate measure of time is kept on the card even if it is disconnected from the reader. This approach provides enhanced security since it separates the card's dependence on obtaining accurate time or key update information from a possibly compromised card reader. Additionally, due to periodic key update, all data processed and stored before the update is protected from duplication or modification by a card attacker. Periodic key updates also make it more of a challenge to exhaustively determine keys since a different analysis is needed after each time period. Although not studied in this work, continually-active smart cards may also allow for analog sensing and other low bandwidth and low performance operations that take place at periodic intervals.

Traditionally, smart cards have been passive systems that remain inactive when not in use. In contrast, our new active architecture uses an embedded crystal and battery to maintain an accurate measure of time. This specific type of security could facilitate secure applications such as financial transactions (tolls, e-cash) and access control (passcard, immigration).

III. Previous Work

Although there are several commercial contactless smart cards containing embedded batteries [8] and others are under consideration [9], smart card battery use is limited. For existing battery-based systems, the function of the embedded battery is primarily to drive contactless communication from the card to the card reader [8]. Previous limitations on embedded battery support were in part due to the lack of appropriate available batteries. Smart card batteries must fit within the 85.6 mm x 54 mm x 0.76 mm (3.5 cm^3) dimensions required to meet constraints of the ISO smart card standard. A number of batteries currently exist which meet the smart card form factor and can provide the necessary mA peak current required by an embedded microcontroller and reader interface. Batteries such as the Cymbet PowerFab [10], Flexion [11], Varta 6804 [12], and Infinite Power LiteStar [13] are commercially available.

The problem of energy preservation in smart card systems has previously been studied in several projects. In Sedlak and Reiner [14], the power consumption of the communication portion of a contactless smart card is regulated to save energy. Recent commercial microcontroller architectures [15] [16] have been specifically optimized for low-power smart card use by lowering the supply voltage and providing associated architectural support. To date, no work has been reported regarding the development of low-power, fault-tolerant circuitry for smart cards requiring continuous operation. Smart cards often operate at one of two clock rates set by the ISO 7816 standard, 3.5712 MHz and 4.1952 MHz [4]. The former clock rate is utilized in our system.

IV. System Architecture

Using existing battery technology and microcontrollers, standard smart card batteries could not power a continuously functioning microcontroller-based card for an extended time period. As we will show in Section VI, a fully-functional microcontroller-based card running at 3.5712 MHz would completely consume a standard-sized battery's energy in about 4 hours, if all parts of the card were continually-active. Our approach to this design issue is to deactivate parts of the card's digital circuitry during most of the card's lifetime. By limiting microcontroller operation to the completion of a few periodic tasks, the card can have the ability to perform computation when it is away from the reader without a significant energy dissipation penalty.

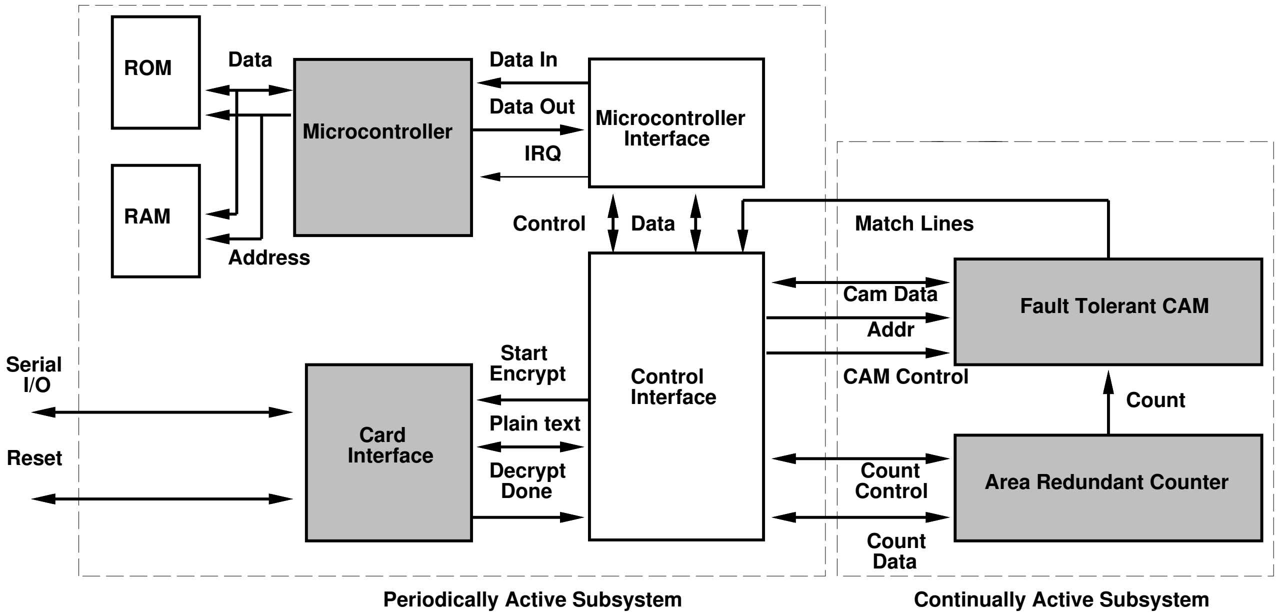


Fig. 1. The periodically-active and continually-active subsystems of the energy-aware smart card

As shown in Fig. 1, the energy-aware architecture is composed of a continually-active subsystem (CAS) and a periodically-active subsystem (PAS). The periodically-active subsystem consists of a microcontroller, a microcontroller interface, an encrypted card reader interface, and a control interface. The continually-active subsystem consists of a CAM and a counter that operate in concert to stimulate the microcontroller to perform periodic computations such as encryption key update. A 3.5712 MHz crystal and an embedded battery provide smart card power and timing support.

The overall architectural structure of the energy-aware architecture is similar to passive (battery-less) smart cards which contain microcontrollers [4], except for the addition of the continually-active circuitry. As shown in Section VI, the CAS module is a small fraction (about 13%) of the area of the entire circuit. When interfaced to a card reader, the internal battery is disabled and the external reader provides the necessary clock and power to the system. When removed from the reader, the crystal and embedded battery provide power and timing. Serial interface signals conforming to the ISO 7816 standard are communicated between the smart card and the card reader to provide data transfer. The continually-active subsystem is isolated from the card reader by the periodically-active subsystem.

The energy consumption of the energy-aware smart card (E_{card}) is a function of system static and dynamic power scaled by the operation time of the periodically-active subsystem (PAS) and continually-active subsystem (CAS). Static power for the PAS (Ps_{PAS}) and static and dynamic power for the CAS (Ps_{CAS} and Pd_{CAS}) is continuously consumed throughout card operation (T_{op}). Dynamic power for the PAS (Pd_{PAS}) is only consumed when the PAS is active (T_{PAS}) As a result, card energy consumption can be expressed as:

$$E_{card} = (Ps_{PAS} + Ps_{CAS} + Pd_{CAS}) \times T_{op} + Pd_{PAS} \times T_{PAS} \quad (1)$$

The goal of the energy-aware architecture is to reduce E_{card} and preserve the finite energy contained in the embedded battery. Even though the microcontroller dissipates considerable energy when it is active, the continually-active circuitry (CAM and counter) dissipates the most energy over the lifetime of the card because it is always active ($T_{op} >> T_{CAS}$). As shown in Section VI, Pd_{CAS} is significantly larger than $Ps_{CAS} + Ps_{PAS}$. Therefore, Pd_{CAS} is the most critical factor with respect to battery lifetime.

The energy-aware smart card can be programmed to perform computation when periodically triggered by the CAS. The CAS counter provides a time reference for periodic activation to perform operations such as key updates. This value is compared against a number of time stamps stored in the CAM. When the count matches a stored key, an interrupt is generated to the microcontroller awaking it from sleep mode and triggering a sequence of operations. The use of a CAM allows for simultaneous matching against multiple keys. Numerous time-sensitive events, such as multiple encryption key updates, card deactivation, or sampling could occur at time points specified by the CAM. Although the basic approach of partitioned active/inactive architectures has been used to reduce power consumption in other types of systems [17] [18], this is the first application of the approach to smart cards.

A. *Continually-Active Subsystem*

The fault tolerant counter and CAM which make up the continually-active subsystem are shown in Fig. 2. The area-redundant, fault tolerant counter includes two standard Gray counters and an associated error correction circuit. The redundant counters operate in parallel and the results are checked for correctness. If an error is detected, the output of the redundant counter is used. A segmented Gray counter architecture is used to minimize the possibility of soft error propagation. Parity prediction logic is used to check the Gray counter for errors. The

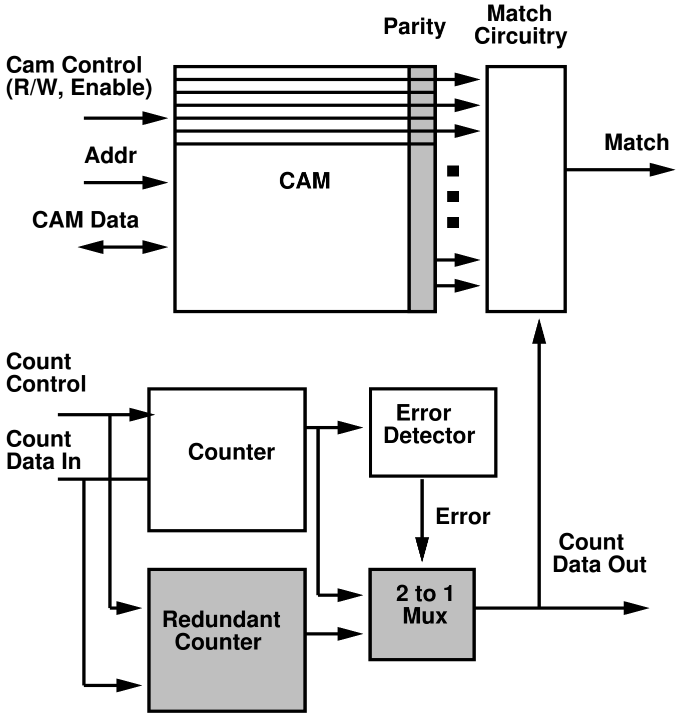


Fig. 2. Components of the continually-active subsystem

parity prediction is based on the observation that the parity of the Gray counter output toggles each cycle. Error detection using parity prediction allows for detection and correction of all single bit errors using only one redundant counter. An exclusive or (XOR) circuit and a toggle flip-flop are used to implement the parity prediction logic. Counter values can be read and written by the PAS when the microcontroller and secure interface are active. In addition to providing simplified fault tolerance, Gray counters have been shown to reduce power consumption by up to 20% versus standard ripple counters [19].

The CAM contains multiple 32-bit words, each augmented with a parity bit. The CAM compares the counter output with the data stored in the CAM and outputs match signals to indicate which CAM cells have been selected. The activation of a match signal sends an interrupt request to the microcontroller. The values stored in the CAM cells operate as timing keys for the interrupts. Periodic operations are activated according to matches with the keys stored in the CAM cells. Key values can be modified by the microcontroller or by the external smart card reader through memory writes.

As shown in Fig. 2, our CAM provides fault tolerance by including parity checking hardware in each row. Given the sensitive nature of our timing key information, the row-based parity approach offers continuous fault protection. Unlike typical RAMs which provide parity checking at the memory output, our row-based parity prevents false matches. Hardware-level fault tolerance is an important aspect of reliable smart card operation. A previous smart card system with an embedded battery [20] uses hardware redundancy (extra microcontroller, static random access memory (SRAM)) to ensure recovery from individual faults. An embedded error detector circuit determines when microcontroller results disagree. This fault tolerance approach is complementary to our technique of providing fault tolerance in the time-keeping portion of the digital circuitry rather than in the microcontroller portion.

The combined counter/CAM architecture of the CAS provides the flexibility needed to maintain fault tolerant, constant smart card operation. The fault tolerance of CAS state provides protection against single-event upset (SEU) data failures. The availability of multiple timing keys provides for different time periods which could be used to update multiple encryption keys. Although not studied in this work, additional possible uses of multiple interrupt periods include the periodic evaluation of system battery power and the potential evaluation of other environmental factors such as voltage and temperature. In total, these applications motivate the use of a CAM which holds multiple timing keys rather than a single register which could only hold a single timing key.

B. Periodically-Active Subsystem

The PAS is made up of a microcontroller and associated interfaces. The microcontroller receives instructions from the CAS or the reader interface via the control interface. As shown in Fig. 1, the microcontroller interface controls the CAM and counter via control, address and data signals. The interface contains a command decoder which interprets commands provided by the card reader and the microcontroller. The circuitry provides a state machine interface between the microcontroller and the continually-active circuitry. The microcontroller interface induces sleep mode by cutting off the on-board clock to the periodically-active subsystem under processor control.

Our energy-smart smart card follows previous card architectures in supporting a card reader interface. As displayed in Fig. 3, the reader interface consists of six blocks: a universal asynchronous receiver/transmitter (UART), data shifter, controller, key register, and encrypt and decrypt blocks. Previous smart cards have used encryption approaches such as RSA [21] [22], DES [22], and AES [23], although the basic structure of the interface has remained roughly constant. As seen in Fig. 3, a key, stored in the key register, is used for both encryption and decryption.

V. Architectural Prototype Implementation

To evaluate our new architecture, a functional prototype, including both hardware and software, was developed. This prototype is intended to demonstrate the design issues and relative power consumption associated with the energy-aware smart card architecture. To this end, we have used representative implementation technology and carefully analyzed component power consumption. The core of our implementation prototype system is an 8-bit 8051 microcontroller, although a number of different microcontrollers could have been chosen. The 8051 microcontroller is widely used in smart card architectures, including several passive smart card architectures [24] [25]. The 8051 interfaces to the control interface via an 11 bit bus.

We have implemented the Tiny Encryption Algorithm (TEA) [26] in hardware for our prototype system to provide for encrypt/decrypt in the secure interface, shown in Fig. 3. The

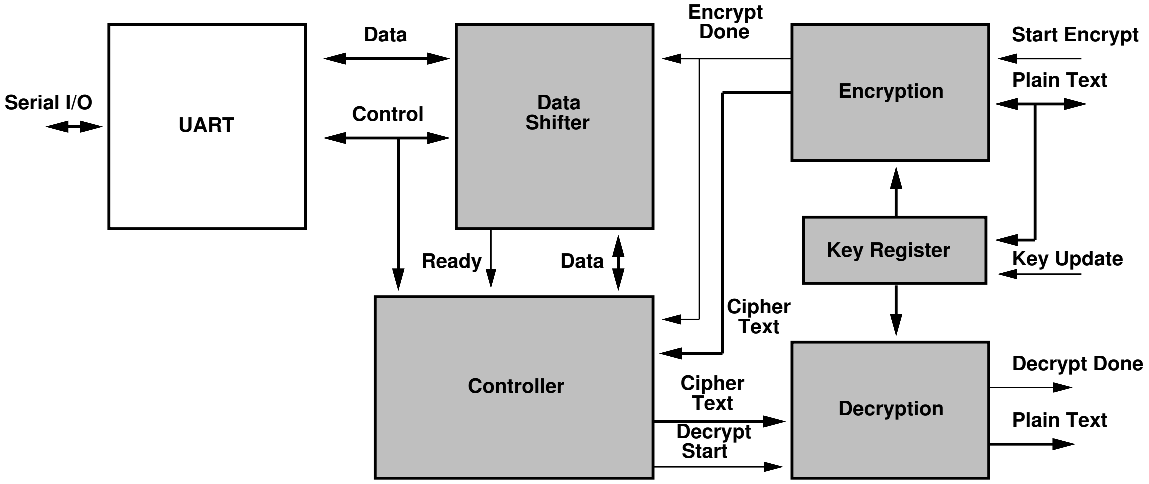


Fig. 3. The card reader interface of the periodically-active subsystem

TEA encrypt and decrypt operations outlined in [26] are implemented via a state machine controller in the secure interface. This algorithm protects instruction decoding for authentication, encryption, and decryption with a private key that can be periodically updated. As will be shown in Section VI, the dynamic power dominance of the CAS on card lifetime mitigates the effect of our specific choice of microcontroller and encryption interface for the prototype.

The implemented CAM and counter each operate on 32 bit data values. Up to 32 timing keys can be implemented in the prototype CAM. Although this key count likely exceeds what would be required for forward security protocols, the implemented CAM provides a forward-looking data point for power consumption that potentially could be reduced for commercial implementation.

A. Adiabatic Circuit Design

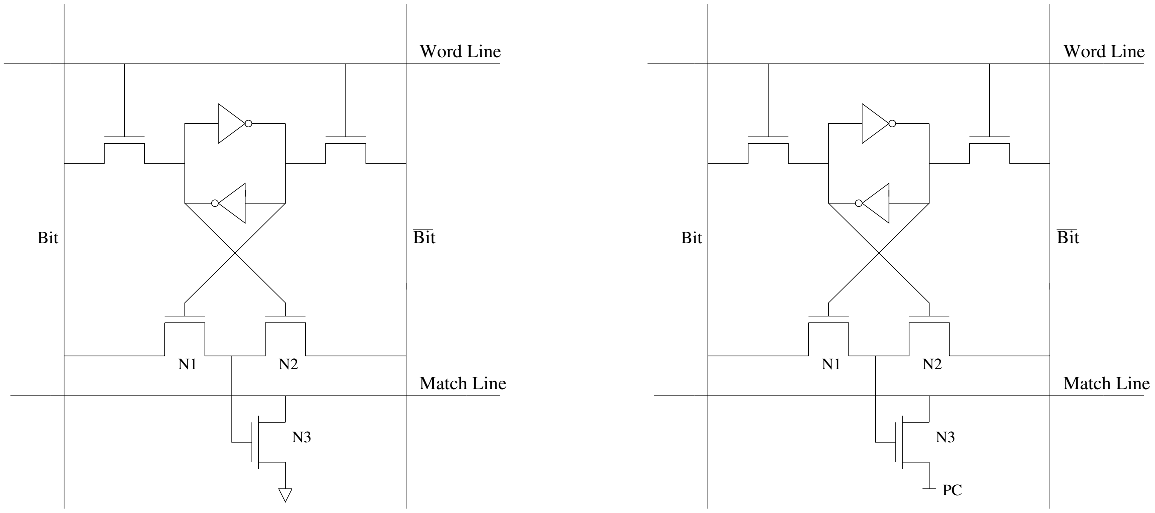
To allow for continuous operation with an embedded battery, it is desirable for the continually-active subsystem to dissipate power in the μW range. In addition to traditional methods of reducing power, such as clock gating, circuit design using *adiabatic* logic allows our smart card to operate more effectively in a low-power environment. Adiabatic circuit structures have been shown to consume 20% of the power [2] of their CMOS counterparts. Several adiabatic logic architectures have been proposed for low-power VLSI design [27] [28] [29] [30]. These architectures achieve low energy consumption by maintaining small potential differences across devices while conducting and allowing the energy stored in circuit capacitors to be recycled.

Our smart card architecture prototype extends previous work in adiabatic circuit design in two specific areas. To allow for low-power CAM operation, the match circuitry in a standard CAM has been been modified to operate in the adiabatic power range. Additionally, a set of logic gates from an existing adiabatic logic family has been used to implement a fault-tolerant adiabatic counter.

1) Adiabatic CAM Implementation: To provide periodic event matching with ultra-low power consumption, we have designed and implemented an adiabatic CAM. As shown in Fig. 4(a), a basic CAM cell is a standard SRAM cell augmented with three labeled compare circuitry transistors. In general, the match line is precharged before every evaluation phase, leading to a substantial source of power consumption. Several previous techniques to achieve CAM power reduction have included selective precharging of the match line, shutting off power to unwanted blocks, and alternating between match line active high and active low [31] [32] [33]. These approaches are designed to reduce power in high performance applications. For low performance applications, significant power is saved by increasing the switching time.

The structure of the adiabatic CAM cell, shown in Fig. 4(b), is the same as the basic CAM cell except that transistor N3 is connected to a clocked power supply (PC) instead of ground. The power clock causes low-swing transitions in the match line, saving considerable energy. The waveform for the adiabatic CAM under the mismatch condition is shown in Fig. 5. The bit lines are pre-discharged and the search data is loaded onto the bit lines. When a mismatch occurs, transistor N3 is ON and the match line follows the power clock; thereby maintaining a very low potential drop across the match line capacitance. The swing in the match line is maintained to a value of one threshold voltage less than the full rail to decrease the charge loss that would arise across N3. The charging and discharging paths are the same for the match line so charge is recovered in the same clock cycle. The match line is therefore held low after the evaluation phase because the energy is recovered by ramping down the power clock. When a match occurs, transistor N3 is OFF, causing no energy dissipation.

Our experiments show a match line energy savings of about a factor of 10, ($5e - 10^{-12}$ J versus $5e - 10^{-11}$ J), for a 32 $times$ 32 CAM operating at 13.95 KHz. As described in Section VI, about 33% of the power of a standard 32 × 32 CAM is consumed by match circuitry. Previously-reported low-power CAM results have been targeted at designs which operate at speeds on the order of 100 MHz. Even at this speed, our adiabatic match line approach compares favorably with previous low-power CAM techniques which demonstrate match-line optimization. In contrast to our 10× match line power improvement, Zukowski and Wang [31] reported an 85% improvement



(a) Basic CMOS CAM Cell

(b) Adiabatic CAM Cell

Fig. 4. CMOS and adiabatic CAM cells. The adiabatic cells are used in the CAM located in the continually-active subsystem

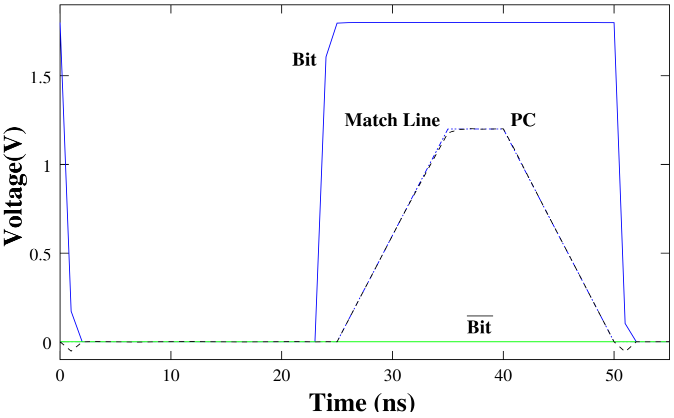


Fig. 5. Bit, match line, and power clock (PC) waveforms for the adiabatic CAM. Note that the match line follows the power clock in this example.

using match line selective precharge. Our overall CAM power reduction results are similar to those reported by Thirugnanam, et al. [32] for a CAM whose match line polarity toggles every access cycle for tests performed at 100 MHz. Since, unlike these previous CAMs, our CAM is optimized to operate at low speed (KHz), these comparisons would be expected to be more favorable for the adiabatic CAM as clock rates are reduced.

2) Adiabatic Static Logic (ASL) Counter: To promote implementation simplicity, we designed an area-redundant read/write counter using static logic gates from the adiabatic static logic (ASL) family [2]. The counter has been designed and built using ASL primitives such as XOR, AND, and OR gates and T and D flip flops. The characteristics of these primitives were previously described in [2]. ASL utilizes a single-phase sinusoidal power source to recycle charge from capacitive load to source through diodes, eliminating a DC path to ground. A 32-bit T-flop counter was integrated with our smart card architecture.

Previous ASL work [2] focused primarily on the construction of individual gate and flip flop primitives. To combine these primitives into a functioning counter we determined that an increase in gate output capacitance was needed. The use of adiabatic logic can induce an increase in leakage current. For low frequency operation, leakage current can potentially lead to unwanted power dissipation and require current refresh. Our ASL logic circuits were tested for leakage current and it was found that this current could be controlled by an increase in output capacitance. The increased capacitance reduces leakage current along ASL source-drain paths during the long rise and fall times associated with adiabatic circuitry by keeping charge constrained to the larger capacitor.

Fifteen flip flops allow for a counter period of up to one year at 13.95 KHz. The counter clock rate is derived from the card primary clock (3.5712 MHz) via division by eight flip flops. The same derived clock is also used to drive the CAM. Register banks are used to interface between the adiabatic logic of the CAS and the standard logic of the PAS. Data transfer occurs when the power clock is at its peak. To generate the single-phase power source for the adiabatic circuitry, a state-machine based resonator is used. This approach uses a bank of switched capacitors and resistors to generate and recycle energy [2] and requires no inductors. The approach is well-suited to our smart card architecture since it can be easily implemented in CMOS and requires minimal space.

B. Design Integration and Testing

The development of the energy-aware smart card prototype has required the design and layout of the architecture, the implementation of design mapping tools, and the mapping of two security-related applications to the target architecture. The implementation has been developed to meet ISO 7816 specifications regarding clock and data rates, data transfer protocols, and size. Initially, Verilog register transfer level (RTL) models of the smart card, including interfaces, the TEA module, the 8051 microcontroller (MCU), the counter and the CAM, were designed. All RTL modules except for the counter and CAM were subsequently synthesized using Synopsys Design Compiler. The functionality of the smart card was verified at the gate level using the Synopsys VCS simulator. Virginia Tech's 0.25 μm standard cell library [34] was used to implement the design at the device level. Power data was generated by simulating design Verilog files with the Synopsys Power Compiler. The design was then floorplanned, placed, routed, and verified with Cadence Silicon Ensemble. Design parameters were then extracted with Cadence Virtuoso and custom designed layouts were used to replace the CAM and counter.

To verify the functionality of the architecture, a number of simulation steps were performed. An integration of all component modules, including the secure interface, 8051, counter, and CAM were simulated together at both the RTL and gate levels using Synopsys VCS. The RTL simulation of the architecture requires about 2 hours while gate-level evaluation requires four days. A testbench was used that implements a sample set of reader commands and readerless key updates. To verify adiabatic behavior, the counter and CAM were verified at the device level using HSPICE simulation. RTL code for the UART was obtained from CMOSexod [35] and the 8051 RTL code was obtained from Dalton [36].

A mapping flow, based on the Keil compiler and assembler was used to map C applications to the byte level. The resulting read-only memory (ROM) programming data was used in conjunction with 8051 simulation. The entire system was then simulated using the Synopsys VCS tool. An interrupt subroutine interface allows CAS-generated event interrupts to awake the 8051 from sleep mode.

C. Sample Applications

To test the functionality of our system and verify the ability to update encryption keys without the use of a reader, we evaluated two key based routines that rely on periodic update. The first application automatically updates the TEA private key without the aid of a smart card reader. As shown in Fig. 2, the CAM generates a match signal indicating when an application should be triggered. This action causes an interrupt request (IRQ) signal that activates the 8051 MCU. The second application generates RSA digital signatures and also supports periodic private key update.

The flow for the TEA key update application is shown in Fig. 6. The sequence of events commences when the timing key CAM entry for TEA *key update* (the timing_key register) matches the value of the CAS time counter. This action causes the assertion of a *Match* signal interrupt to the 8051. Since multiple CAM entries can stimulate the *Match* signal, the interrupt subroutine initially polls CAS counter and CAM locations to determine which application is triggered and then calls the corresponding subroutine. After the TEA update interrupt has been verified, the 8051 generates a new 128-bit key based on the old key. The new key is subsequently forwarded to the reader interface. To moderate key updates, a second 32-bit CAM register, the Energy Monitor, is used to indicate approximately how much energy remains in the battery. After updating the key, the interrupt subroutine sets the corresponding trigger for the next key update based on the energy status. If a low energy state is detected (via Energy Monitor value E) that is less than a preset limit (E_limit), the application update frequency can be adjusted to save power. For example, the key could be updated weekly instead of daily. Energy Monitor values are periodically updated from values in the CAS embedded counter. The TEA key update subroutine was written in C and compiled to byte level.

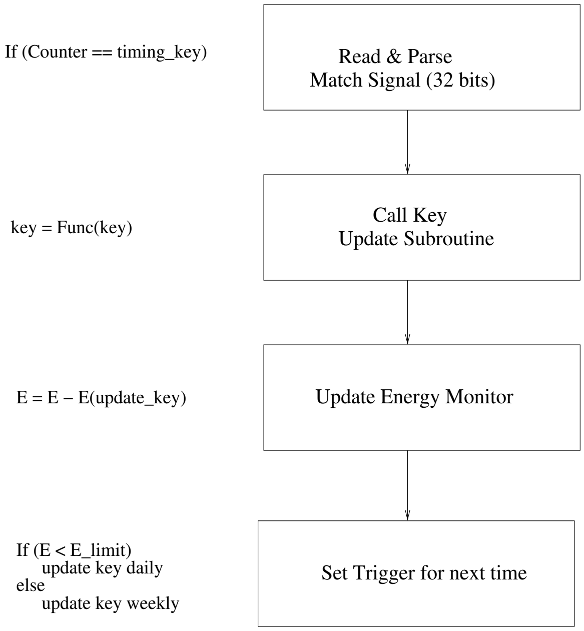


Fig. 6. Actions required during interrupt subroutine for TEA key update

The TEA algorithm is insufficient for providing digital signatures. One requirement of a digital signature is that one party can create the signature and a second party can verify but not create the signature. Such functionality can only be provided by a private-key public-key asymmetric system [37]. To evaluate the types of activities likely to be required for smart card forward security, a digital signature procedure based on RSA encryption was developed. In this procedure, the reader sends an *Authenticate* command to the card. A series of values read before this authentication step are loaded into the 8051 and converted into a single value using a simple addition hash function based on XORs. Then the card loads the three values read by the reader into the 8051: the initial counter value, the 31st word of the CAM, and the first 32-bit word of the key. These values are hashed into an 8-bit value using the XOR functions. When the *Authenticate* command arrives, the 8-bit hash value is signed using the private RSA key. After the computation is complete, the signed hash function (i.e. the signature) is combined with the 16-bit public key, which is stored in RAM, and transmitted to the reader for authentication. The reader has knowledge of the hash function, the data that entered the hash function, and the public key to authenticate the data. However, only the card can sign the data, because only the card has the private key. In the current prototype application, four different private key - public key pairs are loaded onto the card, and are changed periodically using the same interrupt process as the TEA key update scheme. Although these applications illustrate the flexibility of our approach, more sophisticated appli-

Modules	8051	Interfaces	CAM	Counter	RAM	ROM	Total
Area (μm^2)	490,000	1,760,000	72,777	220,000	80,000	45,000	3,178,180
Active Power	14.9 mW		4.1 μW	11.7 nW	65 μW	186 μW	15.1 mW
Sleep Power	765 nW		4.1 μW	11.7 nW	-	-	4.9 μW
Transistors	39,442	81,250	10,826	9,248	8,796	18,514	168,076

TABLE I

THE AREA, POWER CONSUMPTION, AND TRANSISTOR COUNT OF PROTOTYPE ENERGY-AWARE SMART CARD COMPONENTS

cations and key update protocols could be implemented with an increased ROM size and better mapping tools. Since the ROM is inactive most of the time, an increased size likely would not significantly affect the power consumption of the overall circuitry.

VI. RESULTS

To verify the functionality and benefits of our energy-aware smart card architecture, all components were simulated at the functional and transistor levels. A layout of the completed design was created to verify area constraints. Transistor counts, area and power values of all components are listed in Table I. CAM and counter values were generated from 0.25 μm full-custom adiabatic designs, as discussed in Section V. All other values are based on 0.25 μm standard cells. The following characteristics are used to determine constraints:

- In active mode, the PAS circuitry (8051 and interfaces), RAM, and ROM operate at 3.5712 MHz, the standard operating frequency of smart card circuitry.
- In sleep mode, the PAS circuitry does not receive a clock signal and dissipates leakage power. The RAM and ROM receive neither a clock signal nor power and dissipate no static or dynamic power.
- The CAS circuitry (CAM and counter) operates continuously at 13.95 KHz (3.5712 MHz / 256) regardless of operating mode.

A dramatic difference can be seen between the power dissipated and current drawn in active and sleep modes. The smart card in active mode dissipates a total of 15.1 mW (14.9 mW by the PAS), while in sleep mode the card dissipates 4.9 μW (765 nW leakage by the PAS). The digital circuitry draws 6.0 mA of current in active mode and 2.0 μA of current in passive mode.

The TEA key update and RSA digital signature applications, described in Section V-C, were mapped to the card to evaluate battery longevity. The TEA application requires 13,095 microcontroller instructions (0.044 s) per key update and the RSA application requires 12,499 instructions (0.042 s) per signature. By using the calculated power dissipation rates from Table I, the key update frequencies shown in Table II, and Eq. 1, it was possible to determine the length of service for a card employing TEA with periodic key updates. For example, for daily update it is assumed that T_{PAS} is 0.044 s, while T_{op} spans the entire day. The static and dynamic power values are scaled by time to determine energy consumption per day. The energy stored in the battery is then divided by this value to determine smart card lifespan in days. Table II illustrates duration of service values for a variety of smart card batteries and a selection of TEA key update schedules. Total battery energy is listed in the third row based on values taken from manufacturer data sheets [11] [12] [13]. As shown in the left side of Table II, the use of adiabatic circuitry allows for card usage for almost 18 months. All batteries fit the ISO 7816 form factor.

The utility of the adiabatic circuitry can be assessed by re-examining power consumption and duration of service values if the adiabatic CAM and counter are replaced with CMOS 0.25 μm versions. To assess the differences, CMOS versions of the components were created and analyzed. Although the layout size and transistor counts of the CMOS components are similar to their adiabatic counterparts, the CAM power (6.0 μW versus 4.1 μW) and counter power (165.0 nW versus 11.7 nW) are substantially higher. As shown in Table II, the use of CMOS rather than adiabatic circuitry reduces lifetime by about one-third, primarily due to increased CAM power. If the entire design was kept continually-active at all times (continuously consuming 15.1 mW) the battery would expire in about 4.2 hours.

As shown in Fig. 7, a full layout of the architecture in 0.25 μm using the Virginia Tech standard cell library was performed and integrated with crystal and battery terminals. The critical path of the card runs from the secure interface to the 8051 and ends at the CAM data bus. The CAS circuitry (CAM and counter) takes up about 12.9% of the circuit area. Our smart card design was created in a more advanced technology than previously-reported smart cards. Typically, smart cards are designed in older technologies, such as 1.02 μm or 0.62 μm [38]. Smart cards typically fit within a 5 mm $\times$ 5 mm square. The smart card discussed in this work is approximately 1.9 mm $\times$ 1.65 mm. Using quadratic scaling, it is estimated that the smart card designed in this work would be approximately 4.39 mm $\times$ 4.39 mm if it were designed in a 0.62 μm technology. Our smart card circuitry is therefore comparable in scaled area to traditional smart cards.

VII. CONCLUSION

In this paper we have described a novel energy-aware smart card architecture that is continually active. This architecture is motivated by applications, such as forward security techniques, that require periodic update, even when the smart card is away from a reader. Our smart card architecture increases the lifetime of a standard smart card battery by disabling the majority of the smart card circuitry during general operation. To demonstrate the practicality of our architecture, a complete prototype system, including layout, mapping tools, and applications has been developed. The active lifetime of the prototype is enhanced by the use of adiabatic circuit design and energy-aware

	Adiabatic CAS			CMOS CAS		
Battery Name	Varta 6804	Flexion F-402903V0001	Infinite Power LiteStar	Varta 6804	Flexion F-402903V0001	Infinite Power LiteStar
Energy	25 mAh	15 mAh	9.2 mAh	25 mAh	12 mAh	20 mAh
$Key\ update\ freq.$						
Never	1.43	0.86	0.53	1.03	0.62	0.38
Hourly	1.33	0.80	0.49	0.99	0.59	0.36
Daily	1.42	0.85	0.52	1.03	0.62	0.38
Weekly	1.43	0.86	0.53	1.03	0.62	0.38

TABLE II

CALCULATED DURATION OF SERVICE IN YEARS FOR THE ENERGY-AWARE SMART CARD FOR TYPICAL BATTERIES AND KEY UPDATE FREQUENCIES

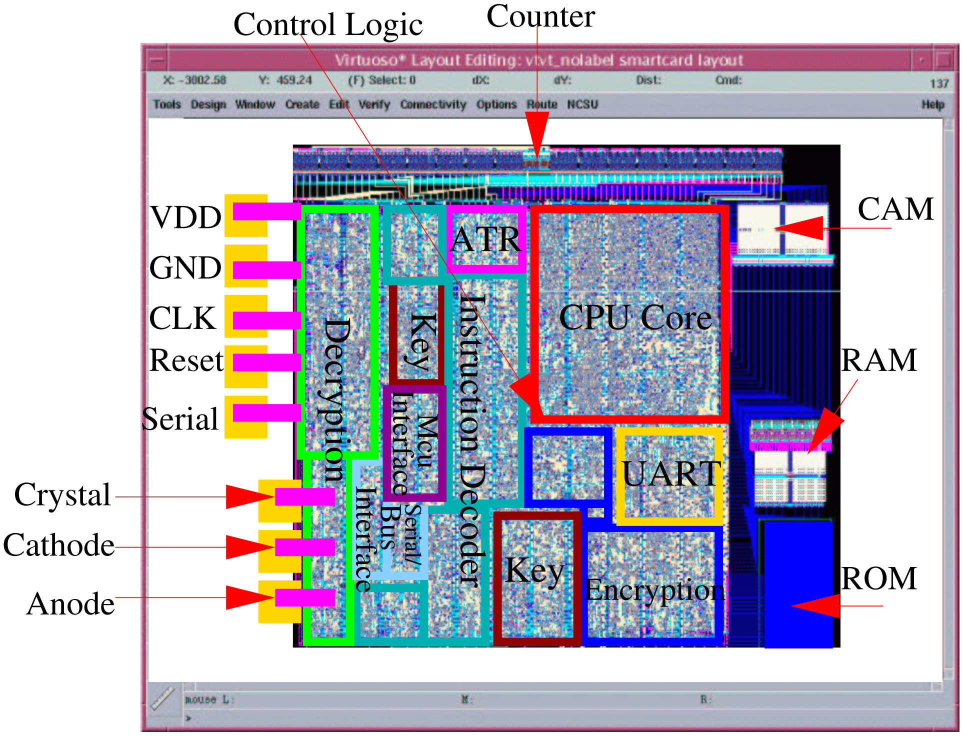


Fig. 7. The integrated energy-aware smart card system

architectural techniques. Our smart card prototype interfaces to a standard smart card reader and provides dynamic encryption key update to enhance card security. The architecture has been shown to be compatible with existing battery technology.

REFERENCES

[1] International Standardization Organization, *Identification cards - Integrated circuit(s) cards with contacts*, 1988.

[2] J. Marjonen and M. Aberg, “A single clocked adiabatic logic - a proposal for digital low power applications,” *Journal of VLSI Signal Processing*, vol. 27, pp. 253–268, Mar. 2001.

[3] K. Vedder and F. Weikmann, “Smart cards - requirements, properties, and applications,” *Springer-Verlag Lecture Notes in Computer Science*, vol. 1528, pp. 307–331, June 1997.

[4] W. Rankl and W. Effing, *Smart Card Handbook*, John Wiley and Sons, Inc., New York, NY, 2000.

[5] M. J. Wiener, “Efficient DES Key Search,” *School of Computer Science, Carleton University, Technical Report TR-244*, May 1994.

[6] M. Bellare and S. Miner, “A forward-secure digital signature scheme,” in *Proceedings: Crypto ’99*, Aug. 1999, pp. 431–448.

[7] M. Bellare and B. Yee, “Forward security in private key cryptography,” in *Proceedings: Topics in Cryptography - CT-RSA*, April 2003, pp. 1–18.

[8] J. Ferrari, R. Mackinnon, S. Poh, and L. Yatawara, “Smart Cards: A Case Study,” *IBM International Technical Support Organization Report*, Oct. 1998.

[9] H. Rouault, E. Crochon, F. Vacherand, S. Martinet, and R. Salot, “Power microsources for smart cards,” in *Proceedings: e-Smart Conference*, Sept. 2003.

[10] Cymbet Corporation, *Power Fab Data Sheet*, 2001.

[11] Silicore, Inc., *F-402903V001 Engineering Data Sheet*, 2003.

[12] Varta Consumer Batteries, *Varta 6804 Data Sheet*, 2003.

[13] Infinite Power Solutions, *LiteStar Technical Brief*, 2003.

[14] H. Sedlak and R. Reiner, *Clocked integrated semiconductor circuit and method for operating such a circuit*, US Patent and Trademark Office. US Patent 6,864,730, May 2005.

[15] Renasas Technology, Inc., *AE46C1 Data Sheet*, 2003.

[16] MIPS Technologies, Inc., *SmartMIPs Architecture, Smart Card Extensions Data Sheet*, 2001.

[17] V. Raghunathan, T. Pering, R. Want, A. Nguyen, and P. Jensen, “Experience with a low power wireless mobile computing platform,” in *Proceedings: International Symposium on Low Power Electronics and Design*, Aug. 2004, pp. 363–368.

[18] E. Shih, P. Bahl, and M. Sinclair, “Wake on wireless: An event driven power management strategy,” in *Proceedings: International Conference on Mobile Computing and Networking*, Sept. 2002, pp. 160–171.

[19] A. Maheshwari, W. Burleson, and R. Tessier, “Trading off transient

fault-tolerance and power consumption in deep submicron VLSI circuits," *IEEE Transactions on VLSI Systems*, vol. 12, pp. 299–311, Mar. 2004.

[20] W. Jackson, *Fault-Tolerant Smart Card*, US Patent and Trademark Office. US Patent 4,908,502, Mar. 1990.

[21] Axalto, A Division of Schlumberger Limited, *Cryptoflex Data Sheet*, 2002.

[22] Infineon Technologies, *Security and Chip Card ICs SLE 66CX320P Product Information*, 2001.

[23] F. Sano, M. Koike, S. Kawamura, and M. Shiba, "Performance evaluation of AES finalists on the high-end smart card," in *Proceedings: Third Advanced Encryption Standard Candidate Conference*, Jan. 2000, pp. 82–93.

[24] Philips Semiconductor, *SmartMX Platform Features: Secure Smart Card Controller Platform*, 2004.

[25] Atmel Corporation, *AT89SC: Secure Microcontrollers for Smart Cards*, 1998.

[26] D. Wheeler and R. Needham, "TEA, a tiny encryption algorithm," in *Proceedings: Fast Software Encryption Workshop*, Dec. 1994, pp. 363–366.

[27] A. Kramer, J. S. Denker, B. Flower, and J. Moroney, "2nd order adiabatic computation with 2N-2P and 2N-2N2P logic circuits," in *Proceedings: International Symposium on Low Power Design*, Apr. 1995, pp. 191–196.

[28] M. Knapp, P. Kindlmann, and M. C. Papaefthymiou, "Implementing and evaluating adiabatic arithmetic units," in *Proceedings: IEEE Custom Integrated Circuit Conference*, May 1996, pp. 115–118.

[29] D. Maksimovic and V. G. Oklobdzija, "Pass-transistor adiabatic logic using single power-clock supply," *IEEE Transactions on Circuits and Systems-II*, vol. 44, pp. 842–846, Oct. 1997.

[30] D. Maksimovic, V. G. Oklobdzija, B. Nikolic, and K. W. Current, "Clocked CMOS adiabatic logic with integrated single phase power-clock supply," *IEEE Transactions on VLSI Systems*, vol. 8, pp. 460–463, Aug. 2000.

[31] C. Zukowski and S. Wang, "Use of selective precharge for low-power CAMs," in *Proceedings: IEEE International Symposium on Circuits and Systems*, Nov. 1993, pp. 745–770.

[32] G. Thirugnanam, N. Vijaykrishnan, and M. J. Irwin, "A novel low-power CAM design," in *Proceedings: IEEE International ASIC/SOC Conference*, Sept. 2001, pp. 198–202.

[33] K. Lin and C. Wu, "A low-power CAM design for LZ data compression," *IEEE Transactions on Computers*, vol. 49, pp. 1139–1145, Oct. 2000.

[34] J. B. Sulistyo and D. S. Ha, "Developing Standard Cells for TSMC 0.25um Technology under MOSIS DEEP Rules," *Department of Electrical and Computer Engineering, Virginia Tech, Technical Report VISC-2002-01*, Jan. 2002.

[35] CMOSexod, *Micro-UART - Synthesizable Universal Receiver Transmitter Data Page*, 2003.

[36] University of California, Riverside, Department of Computer Science, *Dalton Project Data Page*, 2003.

[37] Jan Van Der Lubbe, *Basic Methods of Cryptography*, Cambridge University Press, Cambridge, United Kingdom, 1998.

[38] H. Handschuh and P. Paillier, "Smart card crypto-coprocessors for public-key cryptography," *Cryptobytes*, vol. 4, pp. 6–11, Jan. 1998.

PLACE
PHOTO
HERE

Russell Tessier is an associate professor of electrical and computer engineering at the University of Massachusetts, Amherst. He received the B.S. degree in computer engineering from Rensselaer Polytechnic Institute, Troy, N.Y. in 1989 and S.M. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA in 1992 and 1999, respectively. His research interests include computer architecture, field-programmable gate arrays and system verification.

PLACE
PHOTO
HERE

David Jasinski received the B.S. degree from Worcester Polytechnic Institute, Worcester, MA in 2002 and the M.S. degree in 2003 from the University of Massachusetts, Amherst. His current interests include sonar system development and deployment.

PLACE
PHOTO
HERE

Atul Maheshwari received the B.E. degree in electronics and communication engineering from the L.D. College of Engineering, Gujarat University, Ahmedabad, India in 1998. He received the M.S. and Ph.D degrees in electrical and computer engineering from the University of Massachusetts, Amherst, in 2001 and 2004, respectively. His current research interests include high-speed on-chip interconnects, high-performance and low-power digital circuits, and soft error rate mitigation in VLSI circuits.

PLACE
PHOTO
HERE

Aiyappan Natarajan received the B.E degree in electronics and communication engineering from the University of Madras, Chennai, India in 1999 and the M.S degree in electrical and computer engineering from the University of Massachusetts, Amherst in 2003. He is currently involved in the design of next generation microprocessors.

PLACE
PHOTO
HERE

Weifeng Xu received the B.S. and M.S. degrees in electrical engineering from Fudan University, Shanghai, China in 1997 and 2000, respectively. He is current pursuing a Ph.D. degree at the University of Massachusetts, Amherst. His research interests include reconfigurable computing and fault tolerant systems.

PLACE
PHOTO
HERE

Wayne Burleson is an associate professor of electrical and computer engineering at the University of Massachusetts, Amherst. He received B.S. and M.S. degrees in electrical engineering from MIT, Cambridge, MA and a Ph.D. in electrical engineering from the University of Colorado, Boulder. Dr. Burleson's research interests include VLSI design, reconfigurable computing, content-adaptive signal processing, embedded security and multimedia instructional technologies.