An Energy/Security Scalable Encryption Processor Using an Embedded Variable Voltage DC/DC Converter

James Goodman, Abram P. Dancy, and Anantha P. Chandrakasan, *Member, IEEE*

***Abstract*—Security concerns for battery-operated wireless systems require the development of energy-efficient data-encryption techniques that can adapt to the time-varying data rates and quality-of-service requirements inherent in a wireless application. This work describes the design and implementation of a configurable encryption processor that allows the security provided to be traded off with respect to the energy that is dissipated to encrypt a bit. The processor features an embedded high-efficiency variable-output DC/DC converter that allows the supply voltage to be dynamically varied to match the time-varying throughput and quality requirements of the data stream being encrypted. The resulting processor consumes 134 mW at 2.5 V when encrypting data at a rate of 1 Mb/s using a maximum bit width of 512 bits. The converter efficiency is 96% at the peak load of 134 mW. A comparison of our processor to a software implementation running on a low-power programmable processor shows that our implementation is two to three orders of magnitude more energy efficient.**

***Index Terms*—CMOS digital integrated circuits, cryptography, DC/DC conversion, low power, modular multiplication, reconfigurable architectures, variable power supply.**

I. Introduction

A MAJOR trend in computing hardware today is the development of battery-operated wireless systems such as cellular telephones and hand-held multimedia terminals (e.g., InfoPad [1]). Unfortunately, wireless systems are notorious for their inherent lack of security against unauthorized usage and eavesdropping—a problem that is costing wireless service providers hundreds of millions of dollars per year. In the cellular-phone industry alone, the losses attributed to fraud in 1997 were estimated by the Federal Communications Commission to be on the order of $400 million.

To address these privacy and fraud concerns, wireless systems designers need to exploit various cryptographic techniques such as authentication and encryption. Authentication is a mechanism by which users in a wireless system can verify their identity and status as a valid system user to the wireless system provider and/or other users. A variety of protocols have been developed for authentication in wireless networks (e.g., [2] and [3]). Data encryption, on the other hand, is used to ensure that users' identities and data are not exposed to unauthorized eavesdroppers. While there is a wealth of previous work regarding various encryption algorithms, care must be taken to select an algorithm that is best suited to the particular characteristics and demands of wireless systems. Previously, we have discussed various design constraints that influenced algorithm selection and compared various data-encryption techniques [4]. We concluded that a scheme that encrypted a data stream by XORing it with a pseudorandom key stream was best suited to low-power wireless applications under high bit error rates and limited bandwidth conditions. In this paper, we describe the design and implementation of an energy-efficient data-encryption processor that generates the pseudorandom sequence required for such an encryption scheme.

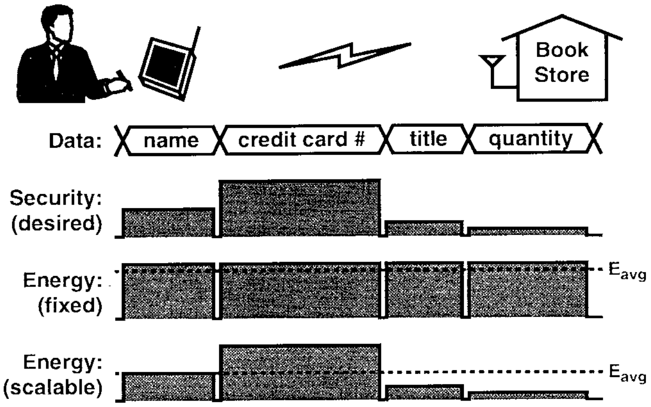


Fig. 1. Energy/security scalability example.

We focus on two main design constraints for portable wireless operation. First, the use of a battery source implies the need for an energy-efficient design methodology in order to maximize the battery lifetime. While encryption is often implemented in software in current mobile systems, several orders of magnitude reduction in energy is possible by using a dedicated hardware solution. The supply voltage must be optimized and computational structures must be constructed to minimize the number of transitions required to implement the given encryption function.

The second major constraint is that wireless systems typically exhibit time-varying data rates and quality requirements. Transmitted data streams often have an inherent structure consisting of both high- and low-priority information, which require different levels of security. For example (Fig. 1), in a

Manuscript received April 6, 1998; revised June 1, 1998. This work was supported by the Defense Advanced Research Project Agency under Contract DAAL01-96-2-0001.

The authors are with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

Publisher Item Identifier S 0018-9200(98)07376-4.

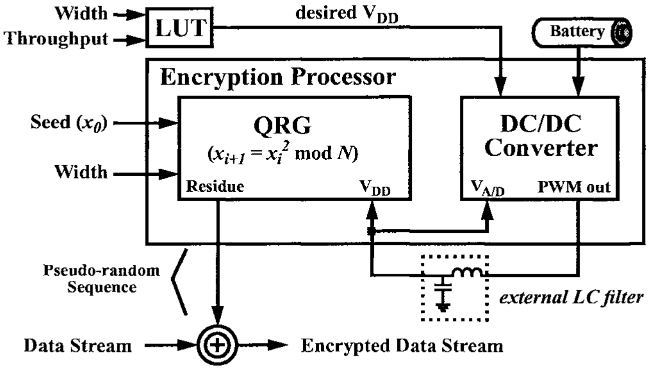


Fig. 2. System architecture of the scalable encryption processor.

retail transaction, certain pieces of information require a lot of security (e.g., credit card information), while other portions of the data stream may require less security (e.g., the title of the book ordered). For such systems, it is desirable to design a reconfigurable processor where the level of security and energy to encrypt a bit can be dynamically traded off. In a conventional system, the user can only decide whether or not encryption is to be used; the amount will be determined by the maximum security requirement of any one piece of information being transmitted. As a result, the average energy dissipation can be significantly higher than that in a scalable system.

Fig. 2 shows the overall system architecture of the encryption processor. The processor consists of two main functional blocks: a variable-security encryption engine and an embedded variable-output DC/DC converter. The encryption engine utilizes an algorithm known as the quadratic residue generator (QRG) to generate a cryptographically secure pseudorandom sequence that is serialized and then XORed with a serial data stream to form an encrypted data stream. The embedded DC/DC converter allows us to utilize variable power-supply techniques [5] in order to minimize the energy consumption by dynamically optimizing the supply voltage as the security requirements or data rate vary. The two blocks are coupled through the use of an external lookup table (LUT) that maps the security and throughput requirements into a digital word representing the desired supply voltage of the processor assuming worst case operating conditions. The digital word is then translated into the required supply voltage by the high-efficiency DC/DC converter.

II. Quadratic Residue Generator

The QRG is a stream cipher construction based on Blum *et al.*'s pseudorandom bit generator [6]. The QRG operates by performing repeated modular squarings of an initial seed value x_0

$$x_{i+1} = x_i^2 \bmod N \tag{1}$$

where the modulus N is the product of two distinct prime values p and q with the property that $p \equiv q \equiv 3 \bmod 4$. The least significant *log log* N bits[1] of each result are then extracted and serialized to form a cryptographically secure pseudorandom key-stream sequence [7]. This key stream has the added property that given the initial seed x_0, a user can access any result of the sequence (e.g., x_j) by performing the modular exponentiation

$$x_j = x_0^{2^j \bmod (p-1)(q-1)} \bmod N. \tag{2}$$

This indexing ability enables the QRG to recover from synchronization errors by allowing the algorithm to be reset to a known state. Hence, if several data bits are lost and the pseudorandom key-stream sequence becomes misaligned with the data stream, the user can wait for the next synchronization marker and use it to reset itself. In addition, bits received after the error and before the next marker can be saved and then decrypted after the fact by generating the required portion of the key stream using (2).

The security of the generator is derived from the difficulty of determining whether or not a number is a square-root modulo-N (i.e., determining quadratic residuosity). This problem has been proven to be equivalent to that of factoring the modulus N into its constituent prime factors p and q [6], the same problem upon which the much more widely known RSA algorithm [8] is based. Factoring is known to be an NP-complete problem that requires massive amounts of time and computation. The amount of computation required to factor a given n-bit modulus ($n = \lceil \log N \rceil$) depends on the factoring algorithm that is used. The running times of modern factoring algorithms all have the same general form [9]

$$L(N, \nu, a + o(1)) = e^{(a+o(1))(\ln N)^{\nu}(\ln \ln N)^{1-\nu}} \tag{3}$$

where a and ν are algorithm-specific constants and N is the n-bit integer to be factored.

Using the best known algorithm for factoring large integers (the general number field sieve [10]), it is estimated that it will require $\sim 3 \times 10^4$ MIPS-years[2] to factor a 512-bit number.

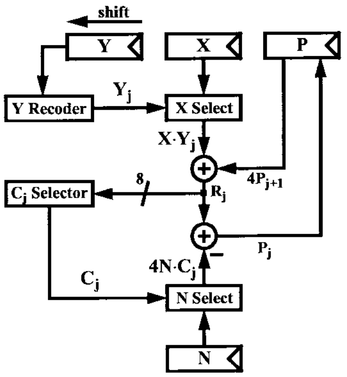
III. An Energy/Security Scalable QRG Architecture

The security guarantees and strong pseudorandomness properties of the QRG come at the cost of the complexity of the modular squaring operation required during each iteration. Hence, the performance of the QRG depends entirely on the ability to perform modular multiplication operations quickly and efficiently.

There are two primary ways to perform modular multiplications: sequentially and concurrently. In a sequential approach, an $n \times n$-bit multiplication is first performed, followed by a $2n \times n$-bit division, where n is on the order of several hundred bits. Unfortunately, the sequential approach has numerous inefficiencies, such as the fact that the intermediate result requires a $2n$-bit register (more if a redundant representation is used), and generating the intermediate result requires a time-consuming $2n$-bit carry propagate addition (CPA). As a result, the sequential approach leads to a slow and inefficient

[1] All logarithms are to the base two unless otherwise specified.

[2] A MIPS-year is the number of computations performed by a 1-MIPS computer operating nonstop for one year ($\sim 3 \times 10^{13}$ computations).



```
INPUTS:
  • $N$: $n$-bit binary modulus
  • $X$: $n$-digit redundant multiplicand
  • $Y$: $n$-digit redundant multiplier
OUTPUTS:
  • $P$: $n$-digit redundant product
        ($P = X \cdot Y \bmod N$)
ALGORITHM:
  1. $P_{n/2+1} = 0$
  2. for $j$ = floor($n/2$) downto -1 do
     • recode $Y<2j+1:2j>$ into $Y_j$
     • $R_j = 4P_{j+1} + X \cdot Y_j$
     • compute $C_j$ using 8 MSD of $R_j$ and $N$
     • $P_j = R_j - 4N \cdot C_j$
  3. $P = P_{-1}/4$
```

Fig. 3. Modular multiplication algorithm.

implementation. A much more efficient approach is to perform the multiplication and division concurrently by performing a partial modular reduction during each step of the multiplication algorithm. As a result, intermediate results require only a few additional digits (e.g., two additional digits [11]) and the results can be kept in a redundant form for both operations so there is no need for a time-consuming CPA. This leads to a much more efficient implementation, a fact that is reflected in the predominant use of concurrent algorithms for performing high-speed modular multiplication [11]–[13]. Note that performance optimizations used in conventional modular multipliers for RSA-based encryption schemes are not applicable to the QRG as the high overhead costs associated with common techniques such as Montgomery multiplication [14] cannot be amortized efficiently in the QRG.

Given the iterative nature of concurrent modular multiplication algorithms (operand sizes on the order of 512 bits preclude the use of array implementations), the multiplier's performance is dictated by two factors: the number of iterations and the cycle time of each iteration. The number of iterations required to perform an n-bit modular multiplication is $n/\log r$, where r is the radix of the multiplication. Hence, the multiplication can be sped up by using a higher radix algorithm. However, for radices above four, multiples of the modulus must be precomputed and stored. The resulting overhead and additional circuit complexity can offset any benefits of utilizing the higher radix. The cycle time of the multiplier can be significantly reduced by the use of a redundant representation that eliminates carry propagation chains. However, the cost of using a redundant representation is that at some point, the result must be converted into a nonredundant binary representation, which will require a CPA. To maintain high performance, this CPA must be performed in such a way as to remove it from the critical path of the multiplier.

A modular multiplication algorithm based on Takagi's iterated radix-4 modular multiplication algorithm [11] was used in the processor. The algorithm (Fig. 3) performs the operation $X \cdot Y \bmod N$ using a conventional iterated approach. During each iteration, two digits of the Y operand are recoded into a radix-4 digit Y_j, which is then used to select ± 2, ± 1, or 0 times the X operand and add it to the previous result (P_{j+1}) to generate the intermediate result R_j. The eight most significant digits of R_j are used to approximate its value and generate a quotient estimate C_j that is used to modularly reduce the intermediate result by selectively adding/subtracting multiples of the modulus N to R_j and forming the new result P_j.

This algorithm is particularly well suited for use in the QRG as its inputs and outputs utilize compatible redundant number formats so that each result can be fed directly back into the multiplier without requiring a time-consuming transformation. In addition, the algorithm maps well to an efficient bit-sliced implementation that reduces global interconnect by distributing control functions and memory locally within the bit slice. A by-product of using both a redundant representation and a bit-sliced implementation is that the critical path of the multiplier is independent of the multiplier's width. Hence, only the number of iterations performed needs to be varied as the multiplier width is changed.

A. Reconfigurable Architecture

Energy scalable computing requires the development of architectures that can be dynamically reconfigured in order to allow the energy consumption per input sample to be varied with respect to the quality. For the QRG, quality refers to the cipher's security, which is equivalent to the amount of time required to factor the n-bit modulus

$$\text{Security} \sim O\left(e^{1.701 n^{1/3}[\ln(n/\log e)]^{2/3}}\right). \tag{4}$$

Hence, security is a subexponential function of modulus width. The energy consumption of the QRG varies with the number of iterations that must be performed, the width of the multiplier, and the operating supply voltage. Assuming the supply voltage is optimized for the multiplier width n, and using a simple first-order delay model where delays scale inversely with supply voltage, the energy scales according to the relationship

$$\text{Energy} \sim O\left(\frac{n^4}{\lfloor \log n \rfloor^3}\right) \sim O(n^4) \tag{5}$$

which is a polynomial function of the modulus width.

Providing this energy/security scalability requires the development of a scalable architecture that can dynamically reconfigure the width of the QRG to vary from 64 to 512 bits in 64-bit increments. The scalable nature of the architecture can be exploited in future implementations to extend the processor to larger widths with a minimal amount of effort, making it particularly well suited to increasing security demands.

Fig. 4 shows the architecture of the QRG. The main portion of the processor is the multiplier data path, which is composed of eight 64-bit blocks that can be enabled/disabled depending on the current width requirements of the algorithm. The control for the QRG is done through the use of a global sequencer that partitions the control logic in such a way as to facilitate dynamic reconfigurability by minimizing the overhead incurred. The sequencer also serves as the I/O interface to the processor.

The use of a redundant representation typically requires a time-consuming conversion operation between the internal redundant representation and the external nonredundant binary representation. However, only the least significant $\log \log N$ bits of each result are required. The output converter circuit

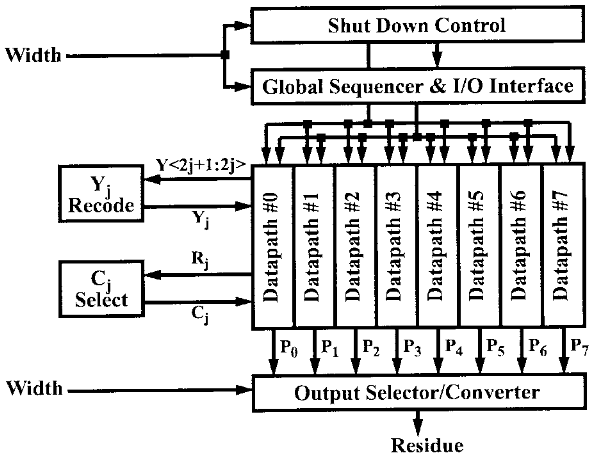


Fig. 4. Architecture of the QRG.

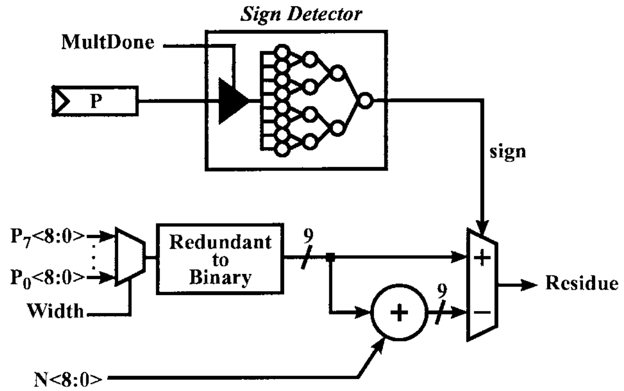


Fig. 5. Block diagram of QRG output converter.

(Fig. 5) uses the width input to determine which digits of the result are required and uses multiplexors to gate these digits into the output converter. These digits are converted into a binary format using a 9-bit carry select adder. The result of this conversion can be in the range $-N/2 < P < N/2$, and a sign-detection circuit is required to determine if a correction factor (i.e., the nine least significant bits (LSB's) of the modulus N) is required. The sign detection circuit utilizes a tree-based topology that features matched delay paths in order to minimize spurious transitions and delays. The converter is only enabled when a result is waiting to be converted in order to further minimize spurious transitions.

IV. Energy-Reduction Techniques

Several architectural and circuit techniques were used in order to minimize the energy consumption of the processor.

A. Concurrency-Driven Voltage Scaling

Reducing the supply voltage reduces the energy consumption quadratically [15]. Unfortunately, propagation delays increase as supply voltages are reduced, leading to a degradation in overall performance. However, by reducing the critical path of the multiplier, the supply voltage can be lowered while still maintaining the initial clock rate, and hence performance.

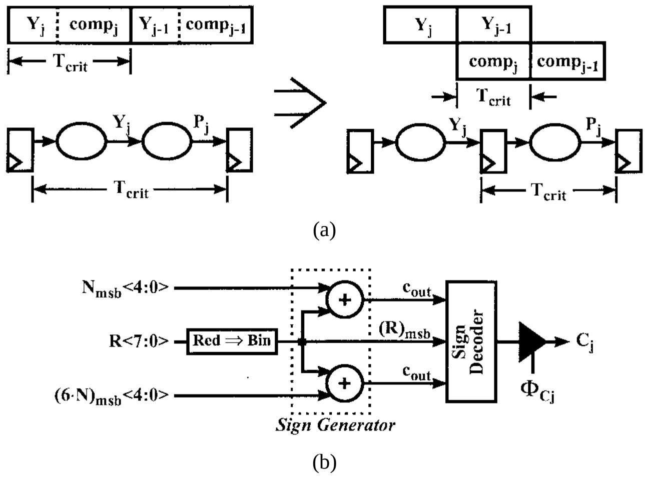


Fig. 6. Critical path reduction techniques. (a) Pipelining the Y-recoder. (b) Parallelizing the quotient estimate.

One way to reduce the critical path of the multiplier is to exploit any parallelism in the algorithm to overlap portions of the computation through the use of pipelining. In the modular multiplication algorithm used, the recoding of the next iteration's radix-4 Y digit (Y_{j-1}) can be overlapped with the current iteration by pipelining the Y recoding circuitry [Fig 6(a)].

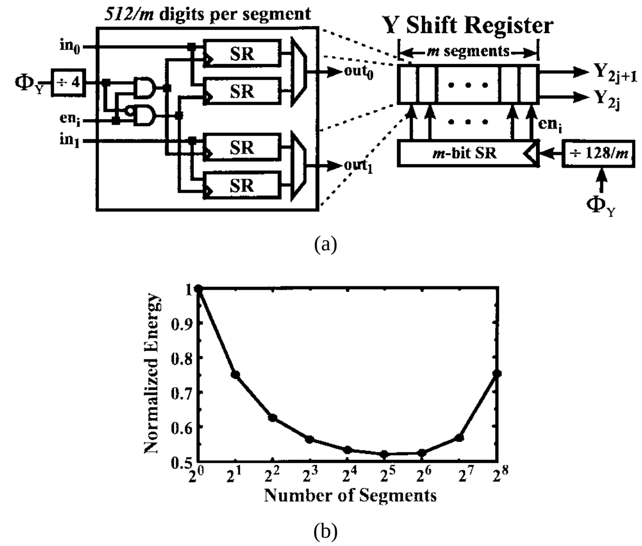
The critical path can also be reduced by accelerating the determination of the quotient estimate C_j. A naive approach to compute C_j requires three time-consuming carry-propagate additions. A much more time-efficient approach takes advantage of the fact that only the signs of these intermediate results are required. Hence, a fast carry lookahead-based sign generator circuit can be used to generate these sign bits in parallel [Fig. 6(b)].

Using these techniques, the critical path of the multiplier was reduced by 27%, allowing the supply voltage to be reduced from 2.9 to 2.5 V, for a total energy reduction of 23%.

B. Clock Gating and Shutdown

Clock gating is used extensively within the processor to disable unused portions of the circuitry in order to minimize the switched capacitance. The enabling/disabling of unused data paths occurs during the multiplier setup phase as the width of the QRG is varied. In addition, the power control block also disables portions of the circuitry as the multiplication is being performed. This intramultiplication power control occurs in the parallelization and systematic shutdown of the Y operand shift register that is distributed throughout the data path and used in the recoding of the Y operand.

First, the shift register is parallelized four ways in order to reduce its clock rate to $f_{\text{mult}}/2$ from $2 \cdot f_{\text{mult}}$, which reduces the switched capacitance and thus the power by a factor of four [16]. The shift register is then partitioned into m segments. When the least significant digit of the Y operand shifts out of a segment, the segment no longer contains useful information and may be disabled by gating the clock to each of the segment's registers. Hence, the shift register is systematically



(a)

(b)

Fig. 7. Switched capacitance reduction of the Y operand shift register.

shut down as the multiplication progresses [Fig. 7(a)]. Ideally, each segment should contain only a single digit so that the minimum number of registers are clocked on any given cycle. However, the overhead of the enable signal generation and distribution grows quadratically with the number of signals, offsetting the benefits of having a large number of segments. We can formulate an expression for the total number of register bits clocked during a given n-bit multiplication by using m-way segmentation

$$\#\text{bits} = 2\sum_{i=0}^{m-1}\left(n-\frac{n}{m}i\right)\cdot\left(\frac{n}{2m}\right)+m^2 = \frac{n^2}{2}\cdot\frac{m+1}{m}+m^2 \tag{6}$$

where the factor of two accounts for the fact that each redundant digit requires 2 bits of storage. Simulations have determined that the optimum number of segments is 32, which approximately halved the average switched capacitance of the Y shift register [Fig. 7(b)]. The net effect of these techniques reduced the switched capacitance of the Y operand shift register by a factor of eight.

C. Self-Timed Gating

A major source of unnecessary switched capacitance in arithmetic circuits is due to spurious transitions that occur because of glitch propagation within the data path. With a data-path width of up to 512 bits, these spurious transitions can add up to a significant amount of wasted energy. One way we have reduced these spurious transitions is by eliminating glitch-generating carry propagation chains altogether through the use of redundant representations. Another approach used in our processor utilizes a self-timed gating approach similar to that used in memory design [17] to partition each multiplier iteration into three distinct phases: R_j computation, C_j computation, and P_j computation (Fig. 8). Tristate buffers are inserted between each of these phases to serve as barriers that prevent glitches from propagating into the next phase of the computation. The buffer enable signals are generated by passing the clock through a delay line that models the critical path of the multiplier and tapping it at the points corresponding to the generation of the R_j and C_j values. This technique succeeded in reducing the switched capacitance of the multiplier by 20% (including the overhead of the delay chain, gating signal distribution, and tristate buffers) as measured from extracted layout using the Powermill power-estimation tool [18].

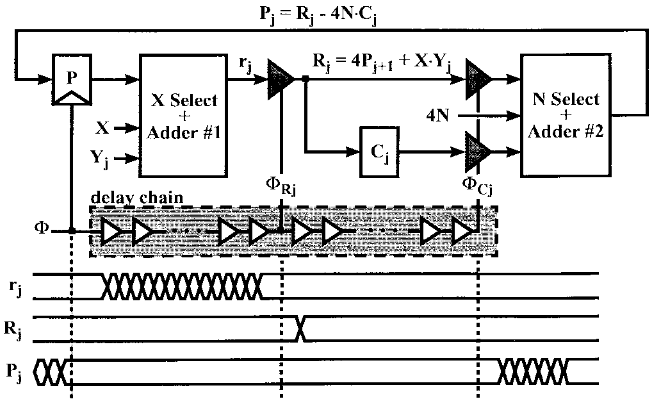


Fig. 8. Block diagram of self-timed gating of the data path.

D. Variable Supply Techniques

Further energy reduction can be achieved using aggressive voltage-supply scaling techniques that take advantage of the time-varying data rates and quality requirements found in wireless applications. In a conventional fixed-supply system, when the computation workload (i.e., the normalized amount of computation per multiplication) is low, due to a reduced data rate or quality of service, the processor will compute as fast as possible and then idle for some fraction of the sample period. Hence, the energy consumption is a linear function of the normalized workload per input sample as the number of operations being performed is reduced. In a variable-supply system, the reduced workload allows the supply voltage and clock frequency to be reduced while still maintaining the required computational throughput. Thus, the energy is reduced relative to a fixed supply system as both the number of operations and the voltage at which they are being performed are reduced.

V. Variable-Output DC/DC Converter

The converter operates in a closed-loop voltage-regulated configuration (Fig. 2), with the converter and QRG coupled through an external LUT that translates the time-varying security and throughput requirements into a digital word representing the required supply voltage. Fig. 9 shows the block diagram of the converter. The current output voltage (V_{out}) is sensed using an internal 7-bit analog-to-digital (A/D) converter, and the resulting digital word is compared to the value programmed to reflect the desired supply voltage. The resulting error term is then scaled in an array multiplier stage and subtracted from the previous iteration's duty-cycle command to produce the next duty-cycle command. The internal representation of the duty cycle is 12 bits, and the ten most significant bits (MSB's) are passed to the pulse-width

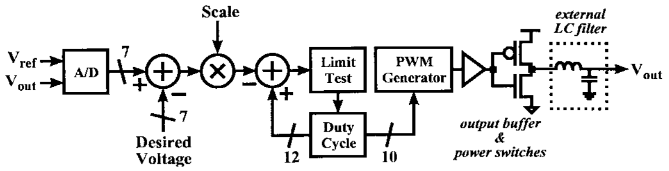
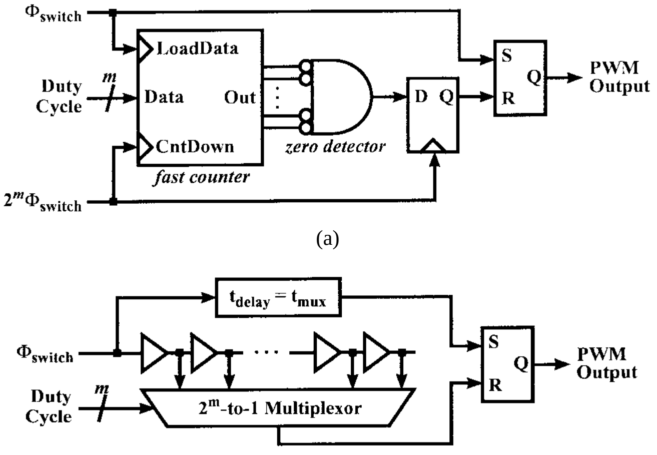


Fig. 9. DC/DC converter top-level architecture.



(a)

(b)

Fig. 10. PWM generator architectures. (a) Fast-clocked counter approach. (b) Pure delay-line approach.

modulation (PWM) stage to create the output. This compensation network forms a variable-gain integral controller. The sample rate of this controller is fully programmable but ultimately limited by the conversion rate of the A/D, which was designed to be 100 ksamples/s.

The output stage is that of a down converter with synchronous rectification. Wide lateral NMOS and PMOS devices are used for the power switches.

A. PWM Generator

The variable DC/DC converter utilizes a very power- and area-efficient hybrid delay-line/counter-based approach to generate the required PWM signal for the output power switches. Previous proposals for generating a PWM signal from a digital command word used either fast-clocked counters [19] or a pure delay-line-based approach [20].

Fast-clocked counters partition the switching interval into 2^m subintervals using an m-bit down counter and zero detector [Fig. 10(a)]. At the beginning of each switching interval, the output flip-flop is set and the counter is loaded with the duty-cycle command word. The counter counts down to zero, at which point the flip-flop is reset. Unfortunately, the counter clock frequency is 2^m times the switching frequency, which implies high power dissipation (on the order of several milliwatts) and thus low efficiencies under low output power conditions.

A pure delay-line-based approach [Fig. 10(b)] partitions the switching interval into 2^m subintervals using a tapped-delay line containing 2^m variable delay elements (e.g., current-starved inverters). The total delay of the line is made equal to the switching interval of the supply through the use of a delay-locked loop (DLL) so that the output of the kth delay element occurs $k/2^m$th of the way through the switching interval. The delay-matching network is used to offset the propagation delay of the multiplexor. The disadvantage of this approach is that it requires a 2^m-to-1 multiplexor in order to gate the required delay-line tap to the reset input of the output flip-flop, which can require substantial area as m increases.

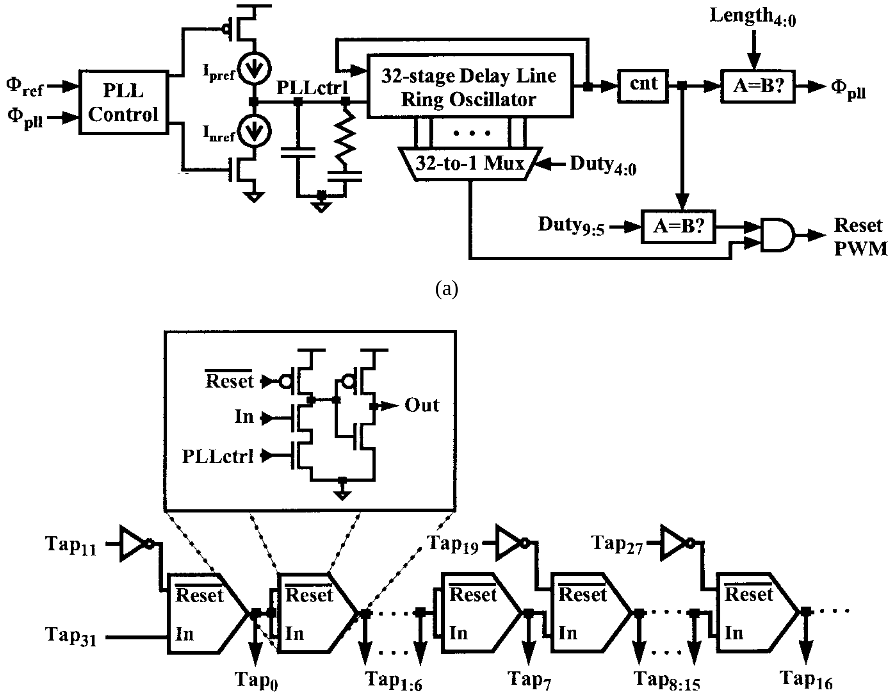
The approach used in this design is a hybrid of the delay-line and counter-based approaches. The PWM generator [Fig. 11(a)] consists of a 32 stage delay line configured as a ring oscillator that is phase locked to a reference clock (Φ_{REF}). A programmable divider allows the ring oscillator frequency to be set two to 32 times faster than the reference frequency. The taps of the delay line then divide the input clock period into between 64 and 1024 equal increments using a 32-to-1 multiplexor. The PWM output is generated by setting the output flip-flop on the rising edge of Φ_{REF} and then resetting the flip-flop when the ring oscillator pulse arrives at the kth tap of the delay line selected by the multiplexor for the nth time, where k and n are specified using the five LSB's and five MSB's of the duty-cycle command word, respectively.

The delay line contains 32 variable delay elements, each consisting of a current-starved buffer, divided into four eight-buffer segments [Fig. 11(b)]. Postcharge logic [21] is used to match leading-edge and falling-edge propagation times and allows a ring oscillator to be created with an even number of stages. The delay of the delay line is controlled by adjusting the gate signals on starvation-type NMOS devices, which controls the speed of the positive going edge at the output of each buffer. The control-node voltage is controlled through a phase-locked loop (PLL) using a charge pump. The biasing for the charge pump is generated on-chip with a low voltage modified 100-nA MOS Widlar current source that uses MOS devices operating in subthreshold (Fig. 12). The compensation network of the PLL is also implemented on-chip using poly-poly capacitors and a poly resistor.

The hybrid approach used in our processor provides considerable advantages over both of the aforementioned approaches. By using a delay line, the circuit can be clocked at a much lower rate than in the fast-clocked counter approach, resulting in a 32× reduction in power in this implementation. This enables us to achieve significantly higher efficiencies at low load power levels. The use of the counter enables the size of the delay line to be reduced so that the width requirements of the selection multiplexor can be reduced by a factor of eight relative to the pure delay-line implementation (assuming 256 taps). This yields a 9× reduction in area relative to the delay-line-based PWM circuit.

B. A/D Converter

The A/D converter is a 100-ksamples/s, 7-bit, charge-redistribution converter. The advantage of a charge redistribution converter for low-power applications is that it can be implemented without amplifiers, which would typically cause significant static currents to be dissipated. A dynamic comparator (Fig. 13) was utilized to compare the capacitor array voltage to an external analog reference at each stage



(a)

(b)

Fig. 11. (a) PWM generator block diagram. (b) PWM generator delay line.

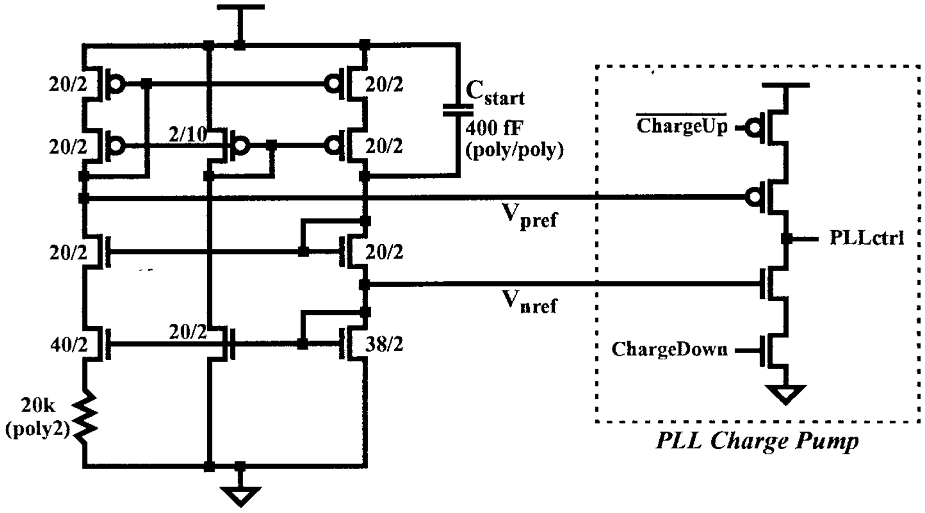


Fig. 12. Low-voltage modified Widlar current source and PLL charge pump (dimensions in micrometers).

of the conversion. The dynamic comparator dissipates power only during evaluations and requires no external biasing networks. The capacitor array utilizes common centroid layout to improve capacitor matching, and there are two rows and columns of dummy devices on the perimeter of the array to enhance matching further. Due to the relatively low resolution of the converter, unit capacitor sizing was rather aggressive; a 10 $\times$ 10 μm poly-poly capacitor giving 47 fF of capacitance.

C. Performance

Table I summarizes the characteristics of the converter controller under two different configurations. The efficiency of the converter is shown in Fig. 14 for a variety of loads up to a maximum of 134 mW. The drop-off in efficiency at low loads is due to the fixed overhead of the switching losses in the output power switches, which were optimized for loads on the order of 100 mW. However, the converter was designed with the ability to operate multiple outputs using the same control circuitry. Hence, at light loads, the efficiency can be improved by using a second set of switches optimized for loads on the order of 1 mW. Efficiencies of 90% have been measured at loads on the order of hundreds of microwatts using a separate stand-alone implementation of the converter that utilizes this approach.

In comparison to a recently reported embedded converter [22], our implementation achieves higher efficiencies at all power loads of interest for our application (e.g., 95% versus ~80% @ 100 mW and 80% versus ~40% @10 mW).

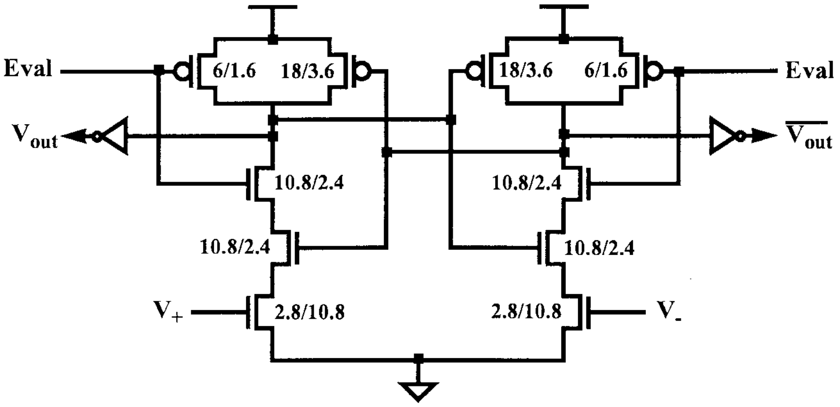


Fig. 13. Dynamic comparator (dimensions in micrometers).

TABLE I
SUMMARY OF EMBEDDED DC/DC CONVERTER PERFORMANCE

A/D INL	±0.5 LSB
A/D DNL	$+0.3, -0.4$ LSB
# of taps	1024
Switching Frequency	500kHz
Min. Supply Voltage	2.05V
PLL & Logic Current	199.3μA
Analog Circuits Current	1.5μA
# of taps	256
Switching Frequency	500kHz
Min. Supply Voltage	1.35V
PLL & Logic Current	42.8μA
Analog Circuits Current	1.5μA
Inductor Value	440μH
Capacitor Value	0.22μF
Output Ripple ($f_{switch} \geq 500kHz$)	$40mV$

TABLE II
PROCESS DETAILS FOR THE ENCRYPTION PROCESSOR

Dimensions	6.2mm × 7mm
Device Count (QRG)	260k
Device Count (DC/DC Converter)	8k
Process	0.6μm DPDM
PMOS Threshold Voltage	V_{tP} = -0.88V
NMOS Threshold Voltage	V_{tN} = 0.75V

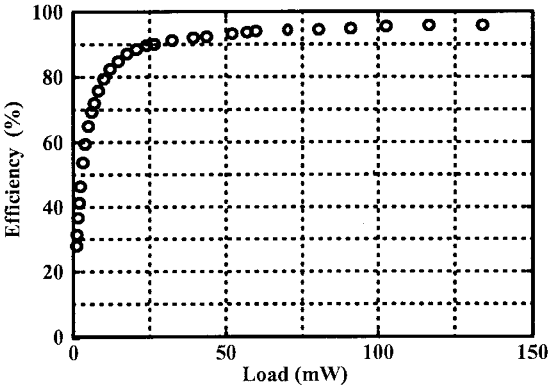


Fig. 14. Efficiency of embedded DC/DC converter.

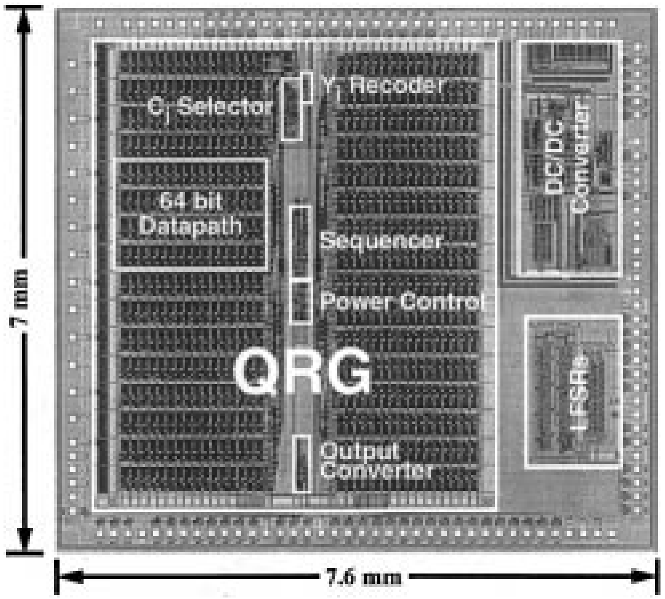


Fig. 15. Microphotograph of encryption processor.

VI. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The encryption processor was implemented using a standard static CMOS design style in a 0.6-μm double-poly double-metal process. Process details are given in Table II. Fig. 15 shows a microphotograph of the processor with several sections highlighted, and Fig. 16 shows a close-up view of the embedded converter. The size of the converter is somewhat misleading, as a large portion of its circuitry is dedicated to test structures used to characterize this prototype implementation that could be eliminated in future implementations.

The encryption processor has been tested at all possible widths, at a variety of rates and supply voltages, and has been found to be fully functional. At its maximum operating speed and width, the QRG circuit operates at a supply voltage of 2.5 V and dissipates 134 nJ per output bit at a rate of 1 Mb/s. This implies a maximum power consumption of 134 mW (140 mW if the power consumption of the DC/DC converter is included).

Energy scalability can be seen in Fig. 17, which shows the effects of both shutting down unused data paths (fixed supply)

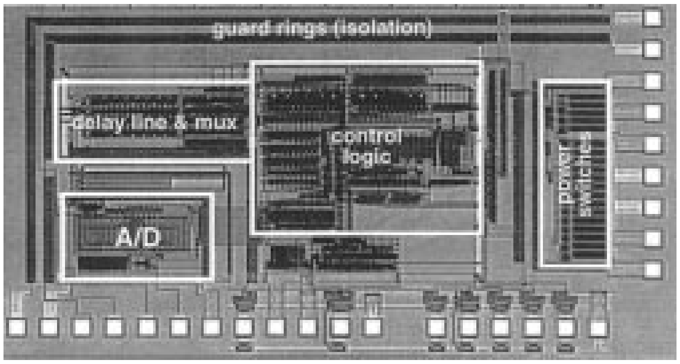


Fig. 16. Close-up of embedded DC/DC converter circuitry.

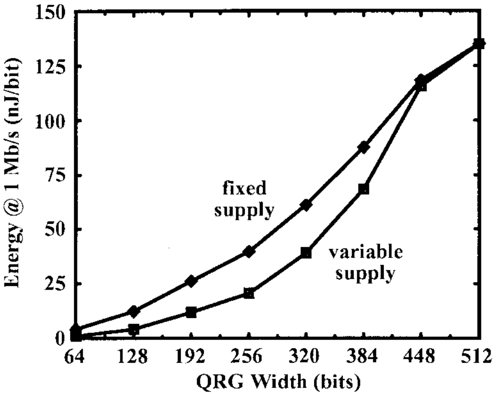


Fig. 17. Energy consumption versus QRG width.

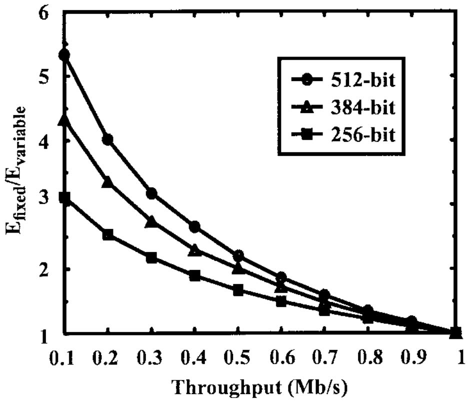


Fig. 18. Energy reduction for varying QRG width and throughput.

and varying the supply voltage to compensate for the variations in computation as the width of the QRG is varied from 512 down to 64 bits at a key-stream rate of 1 Mb/s. The somewhat unusual shape arises because the operating frequency of the QRG is a very nonlinear function of the QRG width

$$f_{\mathrm{QRG}} = \frac{\lceil \log N \rceil / 2 + 4}{\lfloor \log \log N \rfloor} \times f_{\mathrm{DATA}}. \tag{7}$$

Fig. 17 also demonstrates the benefits of using a variable supply voltage relative to a fixed supply—the energy reduction due to the variable supply varies between 1× at a width of 512 bits to 3.8× at a width of 64 bits. When variations in throughput are taken into consideration, the savings can increase up to a factor of 5.33× at a rate of 100 kb/s (Fig. 18).

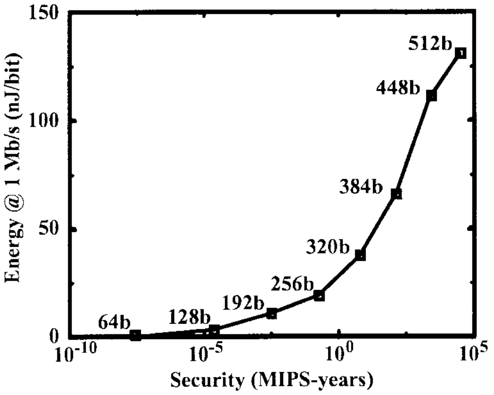


Fig. 19. Energy consumption versus security provided.

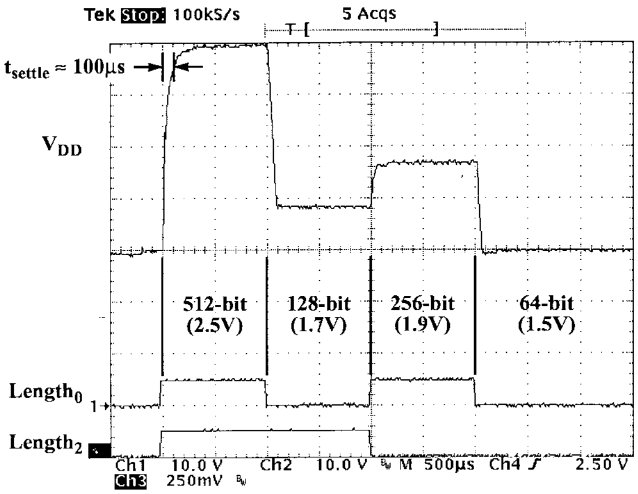


Fig. 20. Dynamic performance of embedded DC/DC converter.

Fig. 19 demonstrates how the energy to produce a key-stream bit of the processor varies as a function of the security that is being provided. As expected, the ratio of security to energy increases exponentially as the width of the processor is increased.

The system performance of the embedded DC/DC converter is shown in Fig. 20. The figure shows how the converter reacts to changing quality requirements (as indicated by the bottom two traces). The 90% settling time of the output is approximately 100 μs.

In comparison to a software implementation running on a low-power 32-bit embedded processor (e.g., StrongARM [23]), it is estimated that our hardware implementation is approximately 100 times more energy efficient. If differences in process technologies used to implement the StrongARM and the encryption processor are taken into account, the energy-efficiency ratio approaches 1000.

VII. Hybrid System

Despite the variety of energy-reduction techniques used during the course of the design, there may still be ultralow power applications (e.g., a wireless video sensor or communication device) for which the power requirements of the processor may be prohibitively high. In such an application, the allotted

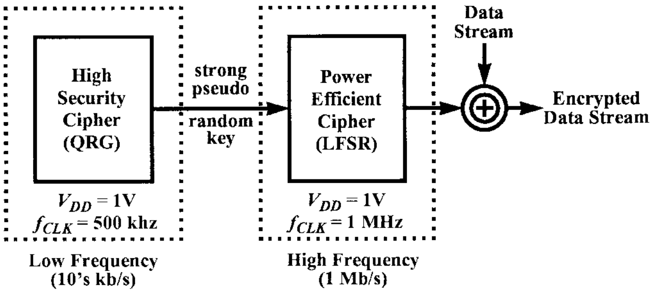


Fig. 21. Proposed hybrid encryption system block diagram.

power budget for encryption may be on the order of hundreds of microwatts at a data rate of 1 Mb/s, which is three orders of magnitude less than the QRG implementation shown here (134 mW).

To provide an adequate level of security while satisfying these strict power requirements, we propose the use of the hybrid system shown in Fig. 21. In this system, the strong pseudorandom output of the QRG is used to periodically reinitialize a much more power efficient, but less secure, cipher. Very power-efficient ciphers can be constructed using the well-developed theory of linear feedback shift registers (LFSR's) [24]–[26]. However, the power efficiency of these ciphers comes at the cost of a firm security guarantee—numerous proposed LFSR-based ciphers that were thought to be secure have been successfully attacked (e.g., [27] and [28]).

Reinitialization of the LFSR-based cipher with the sequence output by the QRG augments the security of the LFSR-based cipher as it has been shown that, without the ability to factor the modulus N, the pseudorandom output of the QRG is indistinguishable from a truly random source [6]. Hence, an attacker is forced continually to restart their attack at the beginning of each initialization period, and the amount of data exposed for any given successful attack is minimized. In addition, by partitioning the system in this way, only the power-efficient cipher needs to operate at the 1-Mb/s data rate. The QRG can operate at a greatly reduced rate on the order of several kilobits per second. For example, in a video sensor application, if the key were updated every frame (i.e., 30-Hz refresh rate), and each update required 100 bits of key information, then the power consumption of the QRG would be

$$P_{\text{QRG}} = (20\,\text{nJ/bit} \cdot 100\,\text{bits}) \cdot (1\text{V})^2 \cdot 30\,\text{Hz} = 60\,\mu\text{W}. \quad (8)$$

The power consumption of the LFSR-based cipher has been measured to be 15 μW at a data rate of 1 Mb/s and supply voltage of 1 V. Combining these results gives a total estimated hybrid system power consumption less than 100 μW, which is well within the allotted power budget.

VIII. Conclusion

Security must become an integral part of wireless systems if the technology is going to be trusted by the mainstream user. However, providing security in an energy-efficient manner requires the development of dynamically reconfigurable architectures that can adapt to the time-varying data rates and quality requirements inherent in wireless systems. These time-varying properties make wireless systems an ideal application for energy reduction using variable voltage supply techniques. This requires the development of embedded high-efficiency variable output power supplies that dynamically vary the system supply voltage to satisfy a given performance specification, as opposed to a given voltage value. The culmination of our efforts led to the design of a dynamically reconfigurable encryption processor that is two to three orders of magnitude more energy efficient and has an order of magnitude better performance than the lowest powered general-purpose processor executing an equivalent software program.

References

[1] R. W. Brodersen, "The network computer and its future," in *1997 IEEE Int. Solid State Circuits Dig. Tech. Papers*, 1997, pp. 32–36.

[2] D. Brown, "Techniques for privacy and authentication in personal communication systems," *IEEE Personal Commun. Mag.*, pp. 6–10, Aug. 1995.

[3] A. Aziz and W. Diffie, "Privacy and authentication for wireless local area networks," *IEEE Personal Commun. Mag.*, pp. 25–31, 1994.

[4] J. Goodman and A. P. Chandrakasan, "Low power scalable encryption for wireless systems," *ACM Wireless Networks*, pp. 55–70, Jan. 1998.

[5] V. Gutnik and A. P. Chandrakasan, "Embedded power supply for low power DSP," *IEEE Trans. VLSI Syst.*, vol. 5, pp. 425–435, Dec. 1997.

[6] L. Blum, M. Blum, and M. Shub, "A simple unpredictable pseudo-random number generator," *SIAM J. Comput.*, vol. 15, no. 2, pp. 364–383, May 1986.

[7] U. V. Vazirani and V. V. Vazirani, "Efficient and secure pseudo-random number generation," in *Advances in Cryptology—Proc. CRYPTO '84*, 1985, pp. 193–202.

[8] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1979.

[9] A. M. Odlyzko, "The future of integer factorization," *CryptoBytes*, RSA Laboratories, vol. 1, pp. 5–12, Summer 1995.

[10] A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, and J. M. Pollard, "The number field sieve," in *Proc. 22nd Ann. ACM Symp. Theory of Computing*, 1990, pp. 564–572.

[11] N. Takagi, "A radix-4 modular multiplication hardware algorithm for modular exponentiation," *IEEE Trans. Comput.*, vol. 41, pp. 949–956, Aug. 1992.

[12] H. Orup and P. Kornerup, "A high-radix hardware algorithm for calculating the exponential M^E Modulo N," in *Proc. 10th IEEE Symp. Computer Arithmetic*, 1991, pp. 51–57.

[13] H. Morita, "A fast modular-multiplication algorithm based on a higher radix," in *Advances in Cryptology—Proc. CRYPTO '89*, 1990, pp. 387–399.

[14] P. Montgomery, "Modular multiplication without trial division," *Math. Computation*, vol. 44, pp. 243–264, 1987.

[15] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, pp. 473–484, Apr. 1992.

[16] T. Barber, P. Carvey, and A. P. Chandrakasan, "Designing for wireless LAN communications," *IEEE Circuits Devices Mag.*, vol. 12, no. 4, pp. 29–33, 1996.

[17] A. P. Chandrakasan, A. Burstein, and R. W. Brodersen, "A low-power chipset for a portable multimedia I/O terminal," *IEEE J. Solid-State Circuits*, vol. 29, pp. 1415–1428, Dec. 1994.

[18] *Powermill User's Manual*, Synopsys Technologies, 1998.

[19] G.-Y. Wei and M. Horowitz, "A low power switching power supply for self-clocked systems," in *Proc. 1996 Int. Symp. Low Power Electronics and Design*, 1996, pp. 313–318.

[20] A. Dancy and A. P. Chandrakasan, "Ultra low power control circuits for PWM converters," in *Proc. IEEE Power Electronics Specialists Conf.*, 1997, pp. 21–27.

[21] R. J. Proebsting, "Speed enhancement techniques for CMOS circuits," U.S. Patent 4 985 643.

[22] T. Kuroda *et al.*, "Variable supply-voltage scheme for low-power high-speed CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 33, pp. 454–462, Mar. 1998.

[23] J. Montanaro *et al.*, "A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor," *IEEE J. Solid-State Circuits*, vol. 31, pp. 1703–1714, Nov. 1996.

[24] S. W. Golomb, *Shift Register Sequences.* San Francisco, CA: Holden-Day, 1967.

[25] D. Coppersmith, H. Krawczyk, and Y. Mansour, “The shrinking generator,” in *Advances in Cryptology—Proc. CRYPTO ‘93*, 1994, pp. 22–39.

[26] C. G. Gunther, “Alternating step sequences controlled by de Bruijn sequences,” in *Advances in Cryptology—Proc. EUROCRYPT ‘87,* 1988, pp. 5–14.

[27] T. Beth and F. C. Piper, “The stop-and-go generator,” in *Advances in Cryptology: Proc. EUROCRYPT ‘84,* pp. 88–92.

[28] P. R. Geffe, “How to protect data with ciphers that are really hard to break,” *Electronics,* vol. 46, pp. 99–101, Jan. 1973.

James Goodman received the B.A.Sc. degree in electrical engineering from the University of Waterloo, Canada, in 1994. He received the S.M. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1996, where he currently is pursuing the Ph.D. degree.

His current research interests are energy-efficient reconfigurable architectures for implementing cryptographic algorithms and protocols, as well as low-power asynchronous design. He has held a variety of industrial positions as both a student and a full-time Engineer with companies such as Bell-Northern Research, Ltd., CAE Electronics, Ltd., and DY-4 Electronics, Inc., working on a variety of projects ranging from virtual-reality hardware engines to real-time CASE tools.

Abram P. Dancy received the S.B. and M.Eng. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1996.

The focus of his research was the development of high-efficiency power supplies for ultralow-power applications. He joined Synqor, Hudson, MA, in 1997, where he currently is developing power supplies for a variety of applications.

Anantha P. Chandrakasan (S’87–M’95) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer science (EECS) from the University of California, Berkeley, in 1989, 1990, and 1994, respectively.

He is an Associate Professor of EECS at the Massachusetts Institute of Technology, Cambridge. He was an Assistant Professor of EECS there from 1994 to July 1998. He held the Analog Devices Career Development Chair from 1994 to 1997. His research interests include ultralow-power implementation of custom and programmable digital signal processors, wireless sensors and multimedia devices, emerging technologies, and CAD tools for VLSI. He is a coauthor of *Low Power Digital CMOS Design* (Norwell, MA: Kluwer Academic) and a coeditor of *Low Power CMOS Design* (New York: IEEE Press). He has been a member of the Technical Program Committee of various conferences, including ISSCC, VLSI Circuits Symposium, DAC, ISLPED, and ICCD. He was a Technical Program Cochair of the 1997 ISLPED and VLSI Design ’98. He is a General Cochair of the 1998 ISLPED and the Signal Processing Subcommittee Chair for ISSCC ’99.

Dr. Chandrakasan is a member of the Design and Implementation of Signal Processing Systems Technical Committee of the Signal Processing Society. He was a General Cochair of the 1998 IEEE Computer Society Annual Workshop. He is a Program Cochair for the 1998 IEEE Workshop on Signal Processing Systems.