Design and Implementation of a Scalable Encryption Processor with Embedded Variable DC/DC Converter

James Goodman
Department of EECS
Massachusetts Institute of Technology
Cambridge, MA 02139
jimg@mtl.mit.edu

Anantha Chandrakasan
Department of EECS
Massachusetts Institute of Technology
Cambridge, MA 02139
anantha@mtl.mit.edu

Abram P. Dancy
SynQor
188 Central Street
Hudson, MA 01749
abester@alum.mit.edu

1. ABSTRACT

This work describes the design and implementation of an energy-efficient, scalable encryption processor that utilizes variable voltage supply techniques and a high-efficiency embedded variable output DC/DC converter. The resulting implementation dissipates 134nJ/bit @ V_{DD} = 2.5V, when encrypting at its maximum rate of 1Mb/s using a maximum datapath width of 512 bits. The embedded converter achieves an efficiency of 96% at this peak load. The processor is 2-3 orders of magnitude more energy efficient than optimized assembly code running on a low-power processor such as the StrongARM.

2. INTRODUCTION

The proliferation of portable wireless communication devices has facilitated the need for using data encryption techniques in order to protect users and system providers. Studies have shown that fraudulent usage of wireless systems is costing both users and network providers several hundreds of millions of dollars a year.

The design described in this paper was motivated by two main design constraints imposed by portable wireless operation. The first constraint is that the portable nature of the application implies a battery-powered implementation. This requires the use of a low power design methodology in order to both maximize the battery lifetime, and to minimize the battery volume. This strict energy budget motivated us to utilize a dedicated hardware solution as it can be several orders of magnitude more energy efficient than a software solution.

The second major constraint is the time-varying data rates and quality requirements that are typically seen in wireless systems. As a result its desirable to design a reconfigurable encryption processor that is capable of dynamically trading off the level of encryption used, and the amount of energy expended to match the current quality requirements of the system. This scalability attempts to exploit the fact that transmitted data streams often have an inherent structure consisting of both high and low priority information that require varying levels of security.

DAC 99, New Orleans, Louisiana

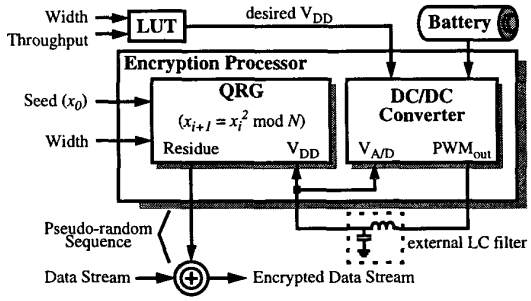


Figure 1: Encryption processor system view.

3. SYSTEM ARCHITECTURE

The overall architecture of the scalable encryption processor is shown in Figure 1. The processor consists of two main functional blocks: a variable security encryption engine, and a variable output DC/DC converter.

The encryption engine utilizes an algorithm known as the Quadratic Residue Generator (QRG [1]) to generate a cryptographically-secure pseudorandom keystream sequence that is then XORed with a serial data stream to form the encrypted data stream. The variable output DC/DC converter allows us to utilize variable supply techniques which dynamically adjust the supply voltage as the amount of computation varies in order to minimize the energy dissipation [2]. The two blocks are coupled through the use of an external look up table (LUT) that translates the current throughput and security requirements (as specified by the Width input) into a digital word representing the desired supply voltage. The embedded DC/DC converter then translates this digital word into a pulse-width modulated (PWM) signal that is filtered through an external LC filter to create the QRG's supply voltage. The voltage is also sampled by the converter in order to perform closed-loop voltage regulation.

3.1 Quadratic Residue Generator

The QRG generates the pseudorandom keystream sequence by performing repeated modular squarings of an initial seed value x_0:

$$x_{i+1} = x_i^2 \mod N \quad (1)$$

where N is the product of two distinct primes p and q. The least significant *loglog* N bits of each result are then extracted to form the keystream. The modular multiplication was performed using a modified version of Takagi's iterated radix-4 modular multiplication algorithm [3].

The architecture of QRG was designed to be energy scalable in the sense that it allows the energy dissipation per input sample to be varied with respect to quality. In this application the quality refers

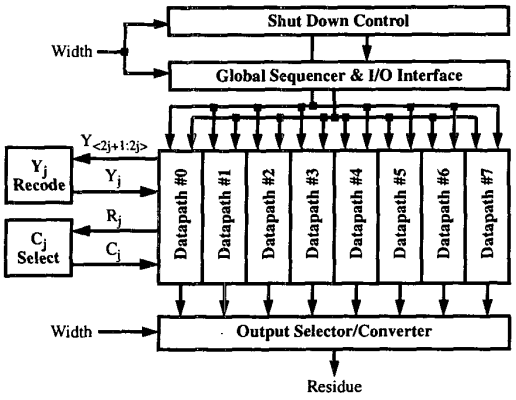


Figure 2: Scalable architecture of the QRG

to the level of security, which is a function of the size of the modulus N:

$$Security \sim O\left(e^{1.701\,(\ln N)^{1/3}(\ln\ln N)^{2/3}}\right) \quad (2)$$

As a result the architecture (Figure 2) allows the multiplier architecture to be reconfigured on the fly to range from 64 to 512 bits in 64 bit increments. The scalable nature of the architecture has the added advantage that it makes it a simple matter to extend the processor to even larger widths in future implementations, allowing it to keep pace with increasing security demands.

The datapath itself was partitioned into eight 64 bit blocks controlled by a global sequencer that also serves as the I/O interface for the processor. The outputs of the datapaths are fed to an output selector/converter that is responsible for determining which bits are required by the output, and then converting those bits from an internal redundant representation into an externally usable non-redundant binary form. Energy dissipation is minimized by the use of a shutdown controller that makes extensive use of clock gating to disable unused portions of the processor in order to minimize the switched capacitance.

3.2 Embedded DC/DC Converter

The embedded DC/DC converter utilizes a PWM-based switching regulator architecture that features a very low power controller that dissipates on the order of 100's of μW. Very low control power enables the converter to maintain high levels of efficiency at very low load power level conditions on the order of 10's of milliwatts.

The top-level architecture of the DC/DC converter is shown in Figure 3(a). The current output voltage (V_{out}) is sensed using a 7-bit charge-redistribution A/D whose output is then scaled and compared to a digital value representing the desired output voltage. The resulting error term is then used to correct the duty cycle of the output power switches via the PWM generator.

The PWM generator (Figure 3(b)) utilizes a power and area-efficient hybrid delay-line/counter-based architecture. The generator consists of a 32 stage delay-line configured as a ring oscillator that is phase-locked to a reference clock. A programmable divider allows the ring oscillator frequency to be set from 2 to 32 times that of the reference. This variation in frequency, combined with the 32-to-1 multiplexor, allows the reference clock period to be divided into between 64 and 1024 equal increments.

Previous proposals for PWM generators in power supply design focused either on pure delay-line based approaches [4], or fast-clock counters[5], which required large amounts of area, and power respectively. Our hybrid approach achieves a 9x reduction in area relative to an equivalent pure delay-line implementation, and a 32x reduction in power relative to an equivalent fast-clocked counter implementation.

Static power dissipation within the A/D was eliminated by both our choice of a standard charge-redistribution A/D architecture (which requires no amplifiers), and the use of a dynamic comparator. Static power dissipation within the PLL was minimized using very small current references (100 nA) within the charge pump through the use of a modified MOS Widlar current reference that operates in the subthreshold regime.

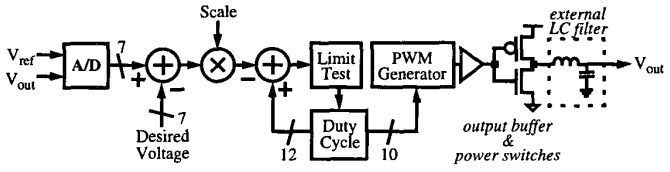
4. Low Power Methodologies

A variety of low power design methodologies were utilized in the design of the processor in order to maximize its energy efficiency.

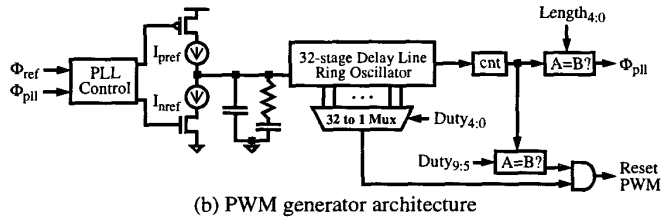
4.1 Glitch Reduction

A major source of unnecessary switched capacitance in arithmetic structures is due to spurious transitions caused by glitch propagation within the datapath due to carry-propagation chains within adder structures. However, by utilizing a redundant internal representation these carry-propagation chains were eliminated completely from our design.

Another primary source of spurious transitions arises from unbalanced delay paths through the combinational logic that makes up each bitslice. The resulting glitching was minimized by applying self-timed gating techniques such as those used for control signal



(a) Top-level architecture



(b) PWM generator architecture

Figure 3: Variable output DC/DC converter architecture

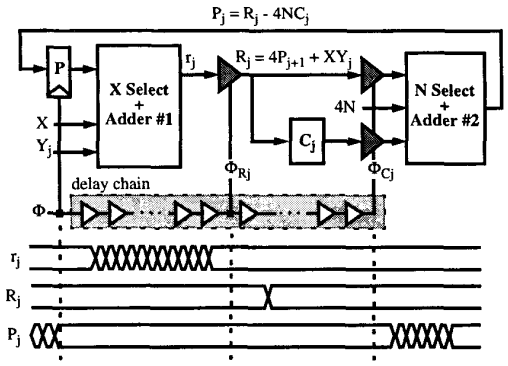


Figure 4: Self-timed gating of the datapath.

timing in memories, to partition the computation into three distinct phases. Tri-state buffers are inserted between each phase to serve as glitch barriers that are only enabled when their inputs are known to be valid and stable. The enable signals for these buffers are generated by passing the clock through a delay line that matches the critical path of the circuit, and then tapping at the points corresponding to the location of the buffers (Figure 4). This technique succeeded in reducing the overall switched capacitance of the processor by 20% as measured using Powermill (including the overhead of the buffers, enable signal generation and distribution).

4.2 Concurrency Driven Voltage Scaling

The energy consumption of static CMOS integrated circuits can be reduced quadratically by scaling down the supply voltage at which they're operating. However, reducing the supply voltage has the unfortunate side effect of increasing propagation delays, leading to a reduction in circuit performance. As a result, any reduction in supply voltage must be accompanied by a corresponding reduction in the critical path of the circuitry. One way of reducing the critical path of the circuit is to exploit any parallelism within the algorithm to allow portions of the computation to be overlapped via pipelining. In the modular multiplication algorithm used, it's possible to overlap the radix conversion/recoding computation of the next iteration with the current one (Figure 5).

In addition, the algorithm requires a magnitude estimation of the current partial product for performing a partial modular reduction.

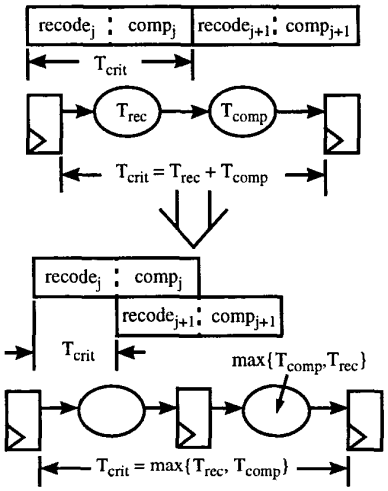


Figure 5: Pipelining the radix conversion.

Typically this estimate requires three time consuming carry propagate additions. However, it can be optimized by realizing that only the signs of the results are required, and these can be computed using fast carry-lookahead circuitry to compute them in parallel.

As a result of these optimizations the critical path of the QRG was reduced by 27%, leading to a supply voltage reduction from 2.9V to 2.5V, and an energy reduction of 26%.

4.3 Clock Gating and Shutdown

Clock gating is used extensively within the QRG to reduce the switched capacitance of the processor by shutting down unused portions of the datapath. The enabling/disabling of unused datapaths occurs as the width of the QRG is varied during the setup phase.

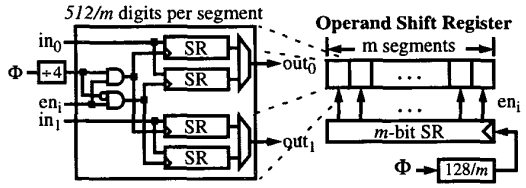
In addition, the power controller also disables portions of the datapath that are no longer required while the modular multiplication is being performed. This intra-multiplication power control occurs in the parallelization and systematic shutdown of long shift registers used for recoding operands that are distributed throughout the chip. The shutdown is performed by partitioning the shift register into *m*-digit segments and then systematically shutting down the segments as the least significant digit of the operand is shifted out of the segment (Figure 6(a)). Analysis has shown that the optimal value of *m* to be 32, resulting in a halving of the registers' switched capacitance (Figure 6(b)).

4.4 Variable Supply Techniques

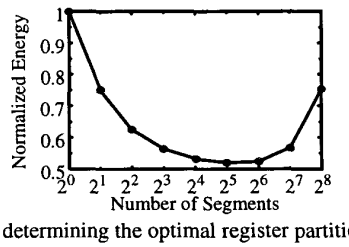
The time-varying data rates and quality requirements that are inherent in a wireless application make it an ideal application for the use of aggressive voltage supply scaling techniques. In a conventional fixed supply system the processor will perform the required amount of computation and then idle once it has finished. Hence, the energy consumption scales linearly with the amount of work that is being performed (i.e., the workload).

By using a variable supply voltage, a reduced workload allows the processor to operate at a reduced supply voltage and clock rate. Hence, the energy consumption is lower than that of the fixed supply system as both the number of operations, and the voltage at which they are being performed is reduced (Figure 7).

The resulting energy savings are what motivated us to develop, and embed, a variable output DC/DC converter within the proces-



(a) parallelization and shutdown of operand shift register



(b) determining the optimal register partitioning

Figure 6: Shift register parallelization/shutdown.

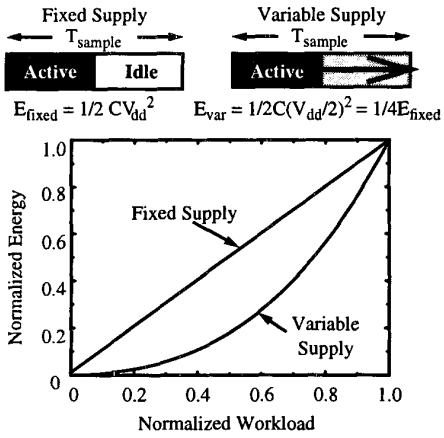


Figure 7: Variable vs. fixed supply operation.

sor.

5. DESIGN FLOW

Initially a baseline was established in software running on a low-power embedded processor. The architecture of the processor was then designed using a top-down approach. The goal of this architecture was to provide an energy-efficient implementation of the QRG algorithm that was capable of operating at a peak rate of 1Mbps. The design had to be dynamically reconfigurable, giving the user the ability to vary the width of the processor from 64b to 512b in 64b increments, and be several orders of magnitude more energy-efficient than the baseline software implementation.

5.1 Software Implementation on SA-1100

In a typical wireless application an algorithm such as the QRG would be implemented in software running on a low power embedded microprocessor. In order to establish a baseline for our design we implemented the QRG in software on the StrongARM SA-1100 -- the lowest powered, highest performing low power embedded 32-bit general purpose processor available at the time. This analysis enabled us to investigate various techniques for writing energy-efficient software for energy-constrained applications such as this.

When implemented in the C programming language, and assuming the processor performs no other function, the computational requirements of the algorithm result in a peak encryption rate of just 26.5kb/s (at a width of 512 bits and clock rate of 200MHz). This results in an energy consumption of 13.11μJ/bit[1]. By utilizing highly-optimized assembly language we were able to reduce the number of cycles per operation by a factor of five, leading to an encryption rate of 125.7kb/s. By reducing the number of cycles we were able to reduce the energy consumption to 2.68μJ/bit. In addition, the assembly language code minimizes the number of energy-intensive external memory references. A comparison of the energy dissipation in the C and assembly language implementations is shown in Figure 8.

To make matters worse, the performance numbers given above are seen to be overly-optimistic when one considers the fact that the

1. Reported energy dissipation is for the SA-1100 core only, it does NOT include the I/O interface or external DRAM/SRAM.

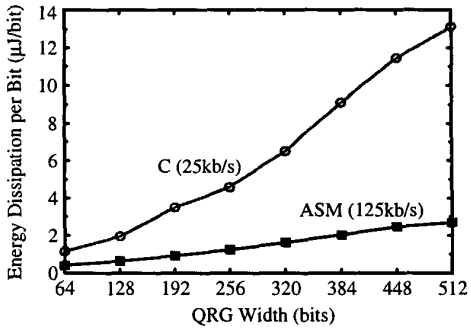


Figure 8: Energy dissipation of S/W implementation.

encryption function must share the limited computational resources of the processor with other processes that are executing concurrently.

5.2 Processor Design and Implementation

Initially a full algorithm simulator was first constructed in the C programming language. This simulator was used to debug the algorithm, as well as to generate test vectors for use at later stages of the design flow.

The algorithm was then implemented at an RTL level using behavioural verilog, from which a structural verilog model was constructed and simulated to enable further debugging.

The Cadence design environment was used for schematic capture, and both Hspice and the Synopsis EPIC design tool suite were used for performing schematic level simulation and power estimation. The circuit design of both the QRG and DC/DC converter utilized a static CMOS design style to ensure a robust implementation.

Cadence was also used for the final layout, LVS, and extraction. The datapath was designed using an area-efficient bit-sliced implementation that reduces global interconnect by distributing control functions and memory locally within the bitslices. The control logic, interface circuitry, and converter controller were implemented using standard cells. The converter layout required careful attention due to its analog circuitry and large power switches, both of which required extensive isolation through the use of guard rings. The A/D's capacitor array utilized a common centroid layout with additional dummy columns and rows to facilitate better matching. The relatively low resolution of the A/D allowed us to use a relatively aggressive unit capacitor sizing of 10μm x 10μm and 47fF.

Final verification was performed on a fully-extracted layout of the entire chip using the EPIC Timemill simulator to ensure functionality. This extensive simulation resulted in a first-pass, fully-functional implementation of the processor.

5.3 Power Estimation and Analysis

The use of EPIC's Powermill power estimation tool proved to be invaluable during the design cycle. The tool enables the designer to get reasonably accurate estimates of the power consumption at both the schematic entry level and extracted layout level of the design flow. This enabled us to investigate various optimizations and trade-offs at both the architectural and circuit levels of the design hierarchy that resulted in significant power/energy savings.

5.4 Fabrication and Testing

The processor was fabricated using a conservative 0.6μm process

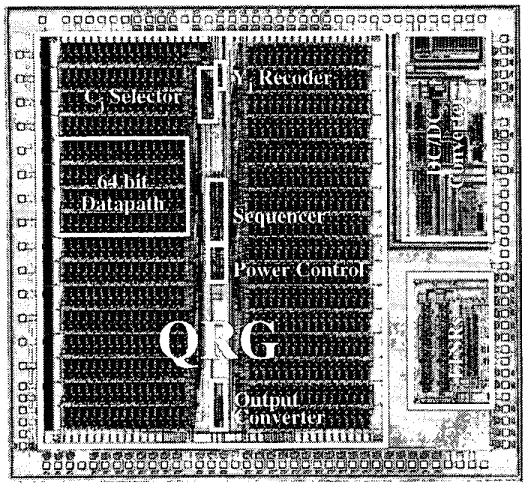


Figure 9: Die photograph of encryption processor.

with two metal layers and two polysilicon layers. Process details are given in Table 1, and an annotated die photograph of the processor is shown in Figure 9. Processor functionality was initially verified using an external power supply, logic analyzer/stimulus generator, and printed circuit board.

Dimensions	6.2mm x 7mm
Device Count (QRG)	260k
Device Count (DC/DC Converter)	8k
Process	0.6µm DPDM
PMOS Threshold Voltage	-0.88V
NMOS Threshold Voltage	0.75V

Table 1: Process summary.

Once both the QRG and DC/DC converter were tested independently and found to be fully functional, a system-level test was performed using the embedded converter as the QRG power supply and the testfixture shown in Figure 10.

6. Performance Measurements

The processor was tested at all possible widths, at a variety of rates ranging from 1kb/s to 1Mb/s and was found to be fully functional in all configurations.

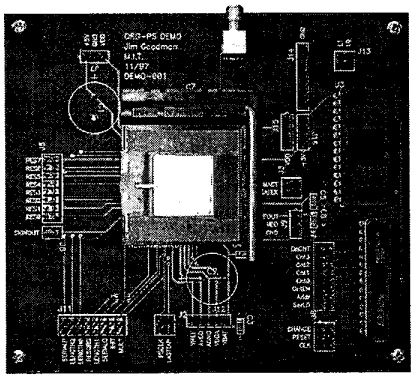


Figure 10: System-level test fixture.

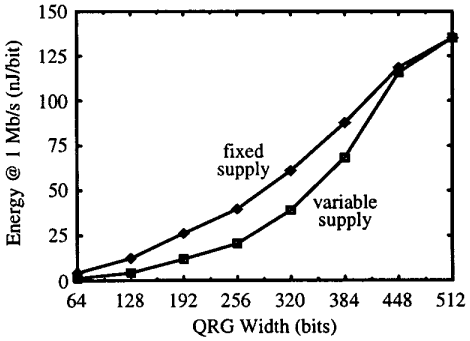


Figure 11: Energy consumption of QRG vs. width.

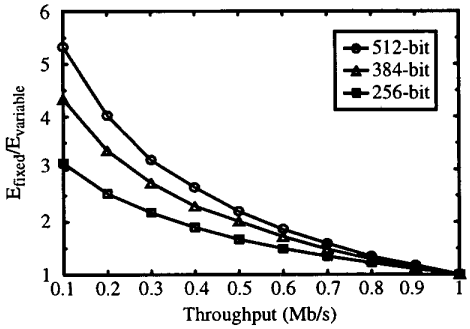


Figure 12: Effects of variable supply voltage.

6.1 Quadratic Residue Generator

The energy scalability of the QRG is seen in Figure 11 which shows the effects of disabling unused datapaths, and reducing the supply voltage to exploit a reduction in the amount of computation as the width is varied from 512 bits down to 64 bits, at a fixed keystream rate of 1 Mb/s. The unusual shape of the curve arises due to the non-linear relationship between clock frequency and width.

Figure 11 also demonstrates the energy savings that result by using a variable supply voltage. These savings are better illustrated in Figure 12 which shows the energy reductions for several QRG widths at a variety of throughputs. The variable supply technique provided energy reductions of up to 5.33x at throughputs of 100 kb/s.

Figure 13 compares the energy-efficiency of our hardware implementation to the aforementioned software implementations. The hardware implementation dissipates between 0.6 nJ/bit (@100 kb/s, 64b) and 134 nJ/bit (@1 Mb/s, 512b). At comparable data rates of 125 kb/s and 25 kb/s, the hardware implementation is 2-3 orders of magnitude more energy efficient than the software implementations. If the differences in process technology are taken into consideration then the processor would be 3-4 orders of magnitude more energy efficient.

6.2 Embedded DC/DC Converter

An annotated, close-up photograph of the embedded DC/DC converter is shown in Figure 14. The system performance of the converter is demonstrated in Figure 15 which shows how the converter responds to changing quality requirements (i.e., variations in the width of the QRG). The 90% settling time of the output supply voltage is approximately 100µs.

The efficiency of the converter is shown in Figure 16 for a variety

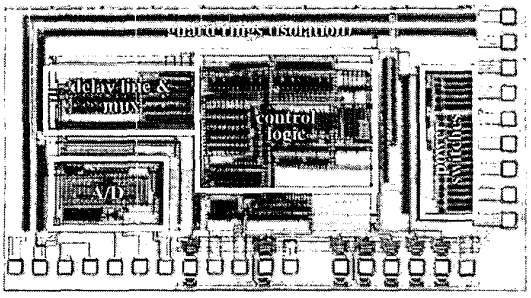


Figure 13: Close-up of DC/DC converter.

of loads. The converter achieves efficiencies of up to 96% at the peak load 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 100mW. However, the converter was designed to support multiple outputs and a second set of power switches could be included to provide high efficiencies at low loads. A separate stand-alone implementation of the converter that utilized this approach achieved efficiencies on the order of 90% at loads on the order of 100's of μW.

7. Conclusions

The migration to, and inherent lack of security of, portable wireless communication systems requires the development of energy efficient architectures for performing data encryption in energy constrained environments. In addition, the time-varying nature of wireless communication channels requires the development of dynamically reconfigurable architectures that can adapt to variations in data rate and quality (i.e., security). These variations can in turn be exploited for significant energy reduction through the use of variable voltage supply techniques. This requires the development of embedded, high-efficiency, variable output power converters that vary the system supply voltage to meet a given performance specification rather than a given voltage specification.

The culmination of our efforts resulted in the design of a dynamically reconfigurable encryption processor that was found to be 2-3 orders of magnitude more energy efficient, and capable of an order of magnitude higher performance than equivalent software implementations. The embedded power converter architecture strikes a balance between energy and area efficiency that yields a hybrid counter/delay-line based architecture that is an order of magnitude more area efficient than pure delay-line architecture, and over an order of magnitude more energy efficient than a fast-clocked counter based architecture. The resulting efficiency of the converter is 96% under full load conditions (i.e., 134mW).

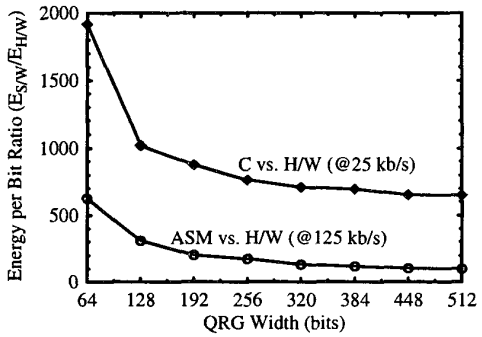


Figure 14: Comparison of energy dissipation.

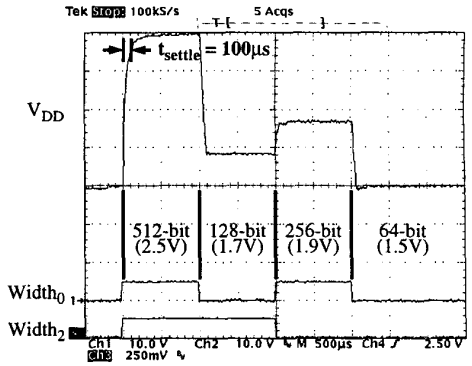


Figure 15: Performance of DC/DC converter.

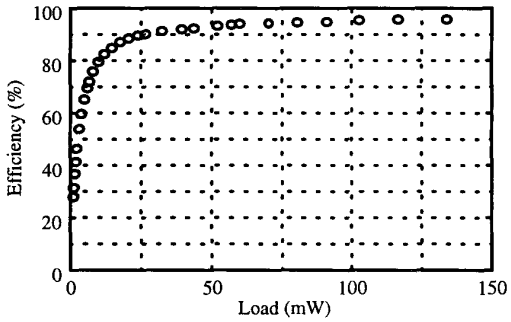


Figure 16: Embedded DC/DC converter efficiency.

8. Acknowledgements

This work was supported by DARPA.

9. References

[1] Blum, L., M. Blum, M. Shub, "A simple unpredictable pseudo-random number generator," *SIAM Journal on Computing*, vol. 15, no. 2, pp. 364-383, May 1986.

[2] Gutnik, V., A. P. Chandrakasan, "Embedded power supply for low power DSP," *IEEE Transactions on VLSI Systems*, vol. 5, no.4, pp. 425-435, December 1997.

[3] Takagi, N., "A radix-4 modular multiplication hardware algorithm for modular exponentiation," *IEEE Transactions on Computers*, vol. 41, no. 8, pp. 949-956, August 1992.

[4] Dancy, A. P., A. P. Chandrakasan, "Ultra low power control circuits for PWM converters," *IEEE Power Electronics Specialists Conference*, pp. 21-27, 1997.

[5] Wei, G-Y., M. Horowitz, "A low power switching power supply for self-clocked systems," *1996 International Symposium on Low Power Electronics and Design*, pp. 313-318, 1996.