Ultra-low Power Encryption Engine for Wireless Implantable Medical Devices

Saied Hosseini-Khayat
Parvin Bahmanyar
Ehsan Rahiminezhad
Digital System Design Laboratory
Electrical Engineering Department
Ferdowsi University of Mashhad, Mashhad, Iran
skhayat@um.ac.ir

Mohamad Sawan, *Fellow*, *IEEE*
Polystim Neurotechnologies Laboratory
Department of Electrical Engineering
Polytechnique Montréal
Montréal, QC H3C 3A7, Canada

***Abstract*—Wireless implantable medical devices are expected to perform cryptographic processing at an absolutely low level of power consumption. This paper presents the design of an ultra-low power ASIC core implementing the PRESENT encryption algorithm. To minimize power consumption, subthreshold CMOS logic is adopted. To implement robust combinational logic (S-Boxes) in PRESENT at subthreshold, a multiplexor-tree architecture based on CMOS transmission gates is proposed. Our post-layout simulations show that our PRESENT core consumes around 50 nW at 0.35V supply voltage at 25 kHz clock frequency, proving the feasibility of ultra-low power encryption.**

I. INTRODUCTION

Wireless implantable medical devices (WIMDs) employ radio transmission technology to enable remote patient monitoring and treatment. However, recent research [1,2] has brought into attention the potential security hazards associated with these devices. To design secure WIMDs, we make the following important observations:

- *Power*: Most WIMDs are battery-operated throughout their extended lifetime which may reach up to 10 years. (Note that replacing an implant normally requires surgery, which is risky, costly and inconvenient.) A back-of-the-envelope calculation suggests that for a modern small ion-lithium battery storing about 3000 Joules of energy to last about 10 years, the average power consumption of the entire WIMD must be less than 10 µW. Given that a WIMD must perform a fair amount of digital signal processing and radio transmission, the amount of power left for crypto-processing is severely limited. We aim at designing a crypto-engine that consumes less than 100 nW.
- *Speed*: Since vital signals do not vary too fast, most WIMDs do not require high processing speeds. Among the fastest signals in body are neural action potentials which can be safely sampled and processed at round 20 kS/s. Therefore our crypto-engine clock frequency is set to 25 kHz.

We have chosen to implement the light-weight block cipher PRESENT [3] as this algorithm provides an adequate level of security at minimal chip area and circuit complexity.

II. PRESENT ARCHITECTURE DESCRIPTION

The PRESENT block cipher is a light-weight encryption algorithm [3] for resource-constrained applications. PRESENT consists of 31 processing rounds and uses a substitution-permutation network. The data block size is 64 bits. We implement PRESENT-80, a version of PRESENT that uses a key size of 80 bits. The details of the PRESENT algorithm are fully described in [3]. Fig. 1 shows a single round of PRESENT, which consist of (a) bit-wise XOR of key and data, (b) 4-bit substitution box (the S-box), and (c) 64-bit bit shuffling (the P-box).

The hardware architecture of PRESENT as implemented in our ASIC core is shown in Fig 2. It is an iterative architecture to reduce chip area. There are two distinct processing loops shown in the figure: (a) key expansion loop (shown on the right-hand side of figure), (b) data processing loop (shown on the left-hand side of figure). The key schedule loop expands the main secret key into 32 sub-keys on-the-fly. The input to this loop is the 80-bit main key. The data processing loop perform one encryption round of PRESENT. An 80-bit key is set up at the key input (almost permanently). A 64-bit plaintext is set up at the plaintext input. When the reset signal goes low, at the rising edge of the clock signal, the key register and data register are both loaded from data and key input ports through their corresponding multiplexors.

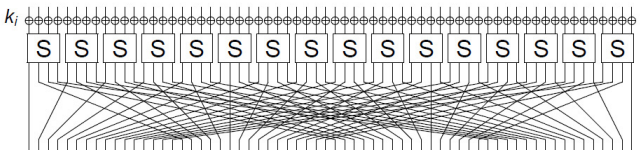

Figure 1. A single round of PRESENT [3]

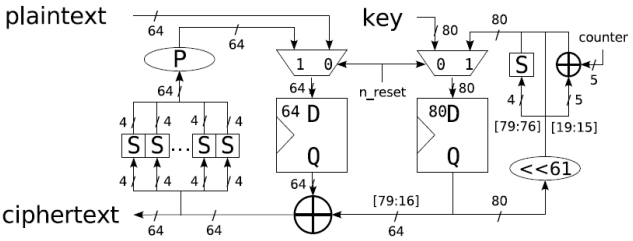

Figure 2. Architecture of PRESENT-80 [3]

After the reset signal goes high, for the next 31 rising edges of the clock, the plaintext is processed to produce a 64-bit ciphertext.

The architecture of PRESENT-80 (Fig. 2) is relatively simple. In terms of the number of hardware resources, it requires 80+64 D-flipflops, 80+64 2-to-1 multiplexors, 64+5 XORs, and 16+1 S-boxes (labeled with S). The P-box (labeled with P), and the bit-rotation block (labeled with "≪61") are simply bit permutations and do not need logic gates (although they consume relatively a large chip area). The S-boxes are all identical and each is a 4-input 4-output combinational logic circuit implementing the Boolean function in Table I.

TABLE I. S-box (4-input, 4-output Boolean function)

x	0	1	2	3	4	5	6	7
$S(x)$	C	5	6	B	9	0	A	D
x	8	9	A	B	C	D	E	F
$S(x)$	3	E	F	8	4	7	1	2

III. SUBTHRESHOLD LOGIC DESIGN

To minimize power consumption, we adopt subthreshold logic design (V_{DD}<0.7V) [4]. Subthreshold logic has been previously employed in biomedical logic circuits with low to modest speed requirements [5]. The subthreshold current is exponentially related to the gate voltage leading to an exponential reduction in power consumption. But it also causes an exponential increase in delay. It has been shown [4, 5] that the reduction in power outweighs the increase in delay, thus producing an overall reduction in energy consumption. The main challenge in subthreshold circuit design is to achieve robust operation in the face of process and temperature variations. We adopt a 0.18 µm process in which variations can still be under control. The problem becomes increasingly worse at finer technologies.

While mainstream ASIC designers almost always employ logic synthesis tools with vendor-designed fully-tested standard logic cells, this luxury is not available in subthreshold logic design. The vendor-designed standard cells have not been optimized for subthreshold operation [6] and cannot be employed as they are. We designed the cells required in our core and tested them for optimal operation at subthreshold supply voltage at all process and temperature corners.

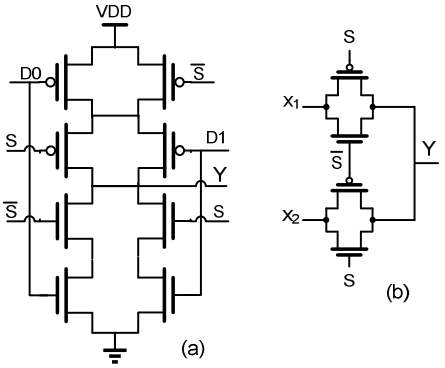

Figure 3. (a) Standard and (b) T-gate-based MUX21

Finally we manually laid-out the entire PRESENT circuit in a TSMC 0.18 µm process. The cells required in our core are as follows: (a) single bit 2-to-1 Mux, (b) 2-input XOR gate, (c) D flipflop, and (d) S-box. In all of those, we use CMOS transmission gates (T-gates) as building block. We describe each of the cells in the next sections.

A. 2-to-1 Mux cell (MUX21)

Fig. 3(a) shows a MUX21 using a standard CMOS configuration, with *S* as the select input, *D* as data input, and *Y* as data output. This circuit does not perform robustly at subthreshold voltage, especially at process corners. We use the T-gate-based MUX21 shown in Fig. 3(b), with *S* as the select input, *X* as data input, and *Y* as data output. This circuit performs well at subthreshold supply voltages at all process corners. In 0.18 µm TSMC technology, at V_{DD}=0.35V, *f*=25 kHz, the standard MUX21 consumes 37 pW whereas the T-gate-based MUX21 consumes only 13.5 pW. The considerable power saving can be attributed to the following facts: (1) the T-gate-based MUX21 has fewer number of transistors, (2) T-gates do not provide direct path from V_{DD} to Ground, reducing total leakage current, and (3) when driving the T-gates at low frequencies, there is no need for large buffers and one can use weak inverters to produce only a minimum drive current.

B. 2-input XOR gate (XOR2)

Fig. 4 shows three different 2-input XOR gates. Circuit (a) is a standard CMOS configuration. Circuit (b) is the Tiny-XOR [7], and circuit (c) is a T-gate-based XOR which is used in our design. Our HSpice simulations show that Tiny-XOR performs better than the standard circuit at process corners down to V_{DD}=0.35V at *f*=25kHz. The T-gate-based XOR also performs well at the above operating point. As shown in Table II, the T-gate-based XOR has the lowest power consumption of all three circuits in a 0.18 µm process. Therefore, we adopted this circuit to be used in our PRESENT core.

TABLE II. Power Consumption of XOR2 circuits (*f*=25 kHz)

Circuit	**Power (V_{DD}=0.35V)**	**Power (V_{DD}=0.4V)**
Standard	Unreliable at corners	200 pW
Tiny XOR	128 pW	165 pW
T-gate XOR	120 pW	157 pW

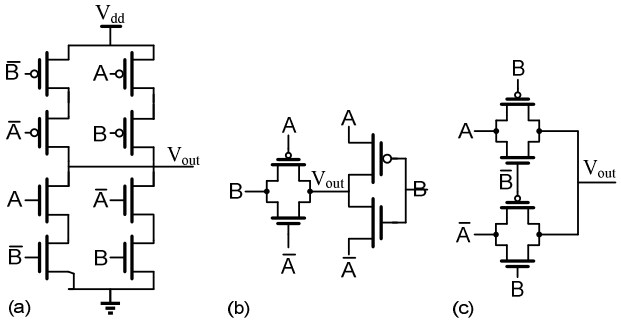

Figure 4. (a) Standard XOR, (b) Tiny XOR, (c) T-gate-based XOR

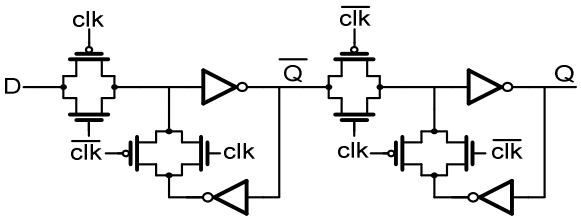

Figure 5. Master-slave T-gate based D-flipflop [8]

C. D-flipflop

Fig. 5 shows the circuit of well-known [8, p. 22] T-gate-based master-slave D-flipflop. Among the several different D-flipflop circuits we evaluated for operation at subthreshold supply voltage, the circuit shown in Fig. 5 performed most reliably and at the lowest power consumption.

In subthreshold design, the exponential dependence of current on threshold voltage V_T becomes more important than sizing. Therefore, cutting the feedback loop for writing into a latch leads to more robust operation at all corners. The flip flop in Fig. 5 uses T-gates to cut off feedback paths in the latches. Our HSpice simulations for this circuit produced an estimated power consumption of 78 pW at V_{DD}=0.35V and f=25 kHz.

D. S-box

There are 17 so-called S-boxes in the PRESENT cipher. An S-box is a special 4-input 4-output combinational Boolean function defined in Table I. This function is such that no logic minimization is possible. When expanded into its disjunctive normal form, this function consists of 4 Boolean expressions, each having at least 5 minterms [3]. Overall, implementing the entire function would require at least 18 3-input AND gates, 4 4-input AND gates, 3 5-input OR gates and 1 5-input OR gates. Given that a total of 17 S-boxes are needed in a PRESENT core, these requirements make S-boxes the most challenging part of designing an ultra-low power PRESENT circuit. A straightforward implementation of this logic in standard CMOS circuits would perform unacceptably at subthreshold voltages due to stacked transistors.

A key idea in our ultra-low power design is the following: *Any combinational Boolean function can be implemented using a tree of 2-to-1 logic multiplexors. Since the T-gate-based MUX21 performs superbly in subthreshold regime, one can implement any Boolean function using a tree of T-gate-based MUX21's to attain ultra-low power consumption.*

If the four outputs of an S-box are called S_0, S_1, S_2, S_3. Fig. 6 shows the mux-tree implementation of the S_0. The other three outputs can also be implemented in a similar fashion. As depicted in the figure, the four inputs x_0, x_1, x_2, x_3 drive the select inputs of the MUX21's. The data inputs of the first layer of the mux-tree are set to constant data (1 or 0) corresponding to the different rows of the truth table of S_0. Each combination of binary values for x_0, x_1, x_2, x_3 selects one row of the truth table. Therefore for a 4-input Boolean function (as for S_0), the mux-tree will consist of 4 layers and a total of 15 MUX21's. Obviously, this approach does not scale well with the number of inputs to the function. However, for a small number of input (e.g. 4 in the case of an S-box), this approach is feasible. An advantage of this approach is that the same circuit works for all functions of the same number of inputs; only the fixed data to the first layer of the mux-tree needs to be changed to correspond to the truth table of the given function.

In some mux-trees, depending on the Boolean function implemented, some optimization may be possible. For example, in the circuit shown in Fig. 6, the top mux controlled by x_0 and the top mux controlled by x_1 can be omitted, reducing area and power.

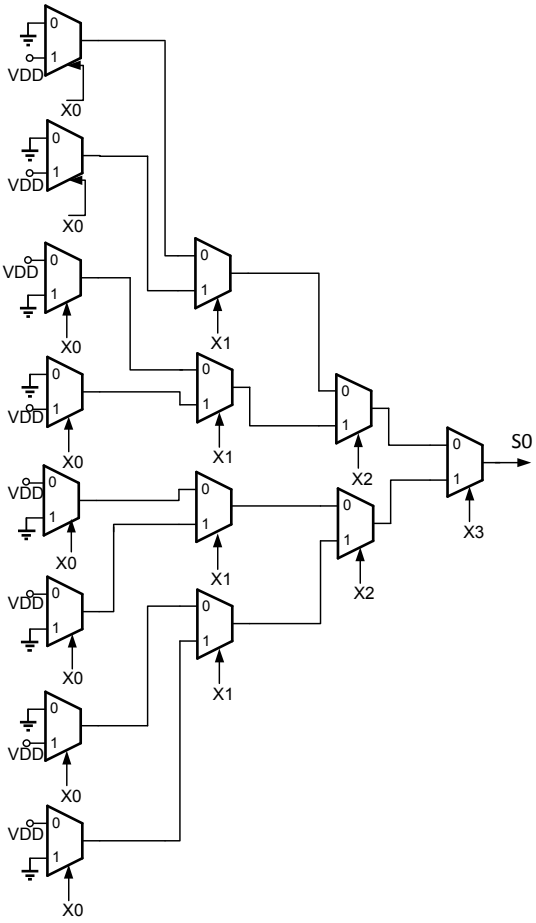

Figure 6. Mux-tree implementation of S0 output

Our HSpice simulations show that at V_{DD}=0.35V, f=25 kHz, an S-box implemented as a T-gate-based mux-tree consumes around 390 pW, whereas an S-box implemented in standard CMOS gates consumes about 487 pW. In addition to about 20% power saving, a mux-tree S-box works reliably at all process corners whereas a standard CMOS S-box performs poorly at some corners (in subthreshold regime).

E. Post-layout Simulation and Comparison

We laid-out our PRESENT-80 core manually (shown in Fig. 7) in a TSMC 0.18 µm CMOS process. The circuit uses the cells described in Section III as its building blocks. Our post-layout simulations of the PRESENT core reports a total of power consumption of 48 nW at V_{DD}=0.35V at 25 kHz clock frequency. This is well below our initial target of 100 nW by a considerable margin and certainly acceptable for WIMD applications. We expect that post-fabrication measurements will validate our simulated results.

Our PRESENT-80 core was shown (by post-layout simulation) to consume no more than 50 nW at V_{DD}=0.35V at 25 kHz clock frequency. The closest published result to this date [9] reports 210 nW for the standard AES block cipher at 30 kHz at V_{DD}=0.35V in a 0.18 CMOS process. Since AES has 128-bit blocks and 10 rounds while PRESENT has 64-bit blocks and 32 rounds, to make a fair comparison, the amount of energy per encrypted bit for the two ciphers are compared: At 25 kHz operating frequency, PRESENT consumes 0.96 pJ/bit of energy (this paper) while the AES engine in [9] consumes roughly 4.8 pJ/bit.

In [3], PRESENT-80 with the same architecture as ours was synthesized for the Virtual Silicon (VST) standard cell library based on the UMC L180 0.18µm 1P6M Logic, and was shown (by simulation) to consume 5 µW at 100 kHz.

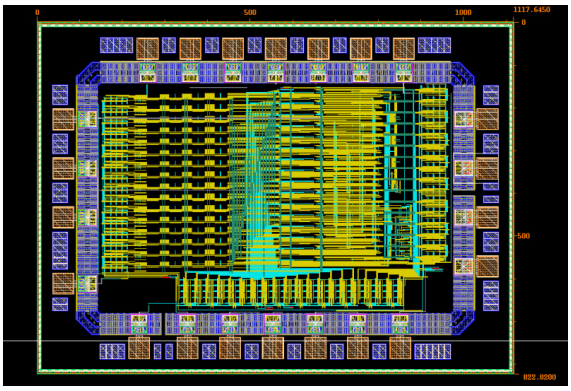

Figure 7. PRESENT chip layout

IV. CONCLUSION

The design of an ultra-low power encryption engine for the PRESENT block cipher aimed at application in wireless implantable medical devices was presented. The ASIC core is designed to operate at subthreshold voltages at 25 kHz in 0.18 µm CMOS process. This process node was chosen rather than the newer, finer technologies because subthreshold design at finer process nodes becomes more difficult due to large process variations. In addition, the finer process nodes provide higher densities and faster clock speeds, which are not required in our type of applications. We used T-gate-based 2-to-1 multiplexors as the building block for the different component required in our core. This resulted in robust operation at all process corners at subthreshold voltages at low frequencies. Thus, it was shown that extremely low power encryption is feasible in hardware. Our design also shows superiority of PRESENT over AES for ultra-low power applications, including implantable medical devices, RFIDs and sensor networks.

ACKNOWLEDGMENT

The authors express their appreciation to Mr. Farjad Farshid for his valuable assistance in IC layout of the PRESENT core.

REFERENCES

[1] Daniel Halperin, Tadayoshi Kohno, Thomas S. Heydt-Benjamin, Kevin Fu, and William H. Maisel, "Security and privacy for implantable medical devices," *IEEE Pervasive Computing*, Vol. 7, No. 1, January - March 2008.

[2] Daniel Halperin, Thomas S. Heydt-Benjamin, Benjamin Ransford, Shane S. Clark, Benessa Defend, Will Morgan, Kevin Fu, Tadayoshi Kohno, and William H. Maisel, "Pacemakers and implantable cardiac defibrillators: software radio attacks and zero-power defenses," *IEEE Symposium on Security and Privacy*, May 2008.

[3] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher," 9th International Workshop on Cryptographic Hardware and Embedded Systems, *CHES 2007*, Vienna, Austria, LNCS, Springer-Verlag, September 2007.

[4] H. Soeleman and K. Roy, "Ultra-low-power digital subthreshold logic circuits," *Proc. Int. Symp. Low-Power Electronics Design*, 1999.

[5] C. H. Kim, H. Soeleman, and K. Roy, "Ultra-Low-Power DLMS Adaptive Filter for Hearing Aid Applications," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, Vol. 11, No. 4, Aug. 2003.

[6] Paul Gerrish, Erik Herrmann, Larry Tyler, and Kevin Walsh, "Challenges and Constraints in Designing Implantable Medical ICs," *IEEE Trans. on Device and Materials Reliability*, Vol. 5, No. 3, 2005.

[7] A. Wang, B. H. Calhoun and A. Chandrakasan, *Sub-threshold design for ultra low-power systems*, Springer Publishers, 2005.

[8] Neil H. E. Weste, and Davis Harris, *CMOS VLSI Design*, 3rd Edition, Pearson Education, 2005.

[9] C'edric Hocquet, Dina Kamel, Francesco Regazzoni, Jean-Didier Legat, Denis Flandre, David Bol, Francois-Xavier Standaert, "Harvesting the potential of nano-CMOS for lightweight cryptography: an ultra-low-voltage 65 nm AES coprocessor for passive RFID tags," J. Cryptographic Engineering, 1(1), 2011.