32nd Annual International Conference of the IEEE EMBS
Buenos Aires, Argentina, August 31 - September 4, 2010

Implantable Electronics: Emerging Design Issues and An Ultra Light-Weight Security Solution

Seetharam Narasimhan, Xinmu Wang and Swarup Bhunia

Abstract— **Implantable systems that monitor biological signals require increasingly complex digital signal processing (DSP) electronics for real-time *in-situ* analysis and compression of the recorded signals. While it is well-known that such signal processing hardware needs to be implemented under tight area and power constraints, new design requirements emerge with their increasing complexity. Use of nanoscale technology shows tremendous benefits in implementing these advanced circuits due to dramatic improvement in integration density and power dissipation per operation. However, it also brings in new challenges such as reliability and large idle power (due to higher leakage current). Besides, programmability of the device as well as security of the recorded information are rapidly becoming major design considerations of such systems. In this paper, we analyze the emerging issues associated with the design of the DSP unit in an implantable system. Next, we propose a novel ultra light-weight solution to address the information security issue. Unlike the conventional information security approaches like data encryption, which come at large area and power overhead and hence are not amenable for resource-constrained implantable systems, we propose a multi-level key-based scrambling algorithm, which exploits the nature of the biological signal to effectively obfuscate it. Analysis of the proposed algorithm in the context of neural signal processing and its hardware implementation shows that we can achieve high level of security with ~ 13X lower power and ~ 5X lower area overhead than conventional cryptographic solutions.**

I. Introduction

Implantable systems that record and manipulate biological activity can largely benefit from *in-situ* real-time signal processing. Such signal processing typically serves two major purposes: 1) compression of recorded data; and 2) real-time online signal analysis in order to recognize meaningful patterns from recorded data. Data compression is an important requirement for systems which wirelessly transmit recorded signals to the external world, possibly from multiple channels, in order to drastically reduce transmission bandwidth and power requirements. On the other hand, pattern recognition from recorded data is important in closed-loop control systems for manipulation of body activity or drug delivery. Such implantable systems [2] are used in diverse contexts, ranging from sensing abnormal activity in the central nervous system to stimulating sensory or motor neurons in the central or peripheral nervous system. With increasing requirement of the implantable systems to perform advanced digital signal processing (DSP) on large volume of recorded data, there is a growing need to develop efficient signal processing algorithms along with their low-power, low-area hardware implementation. It has been shown in [1] that the system power is dominated by the wireless transmission and that can be reduced drastically by using complex signal processing on-chip to reduce the volume of transmitted data. Several researchers have studied the feasibility [3] of implementing signal processing circuits of varying complexity inside the implant unit.

We have previously proposed the use of a hierarchical vocabulary-based neural pattern recognition algorithm [4] which can be efficiently implemented on-chip within the implant while satisfying the power and area bounds [5]. It has been shown that nanotechnology shows tremendous potential in implementing such systems due to large benefits in terms of integration density and switching power. However, it also brings in new design challenges such as reliability of operation due to the reduced robustness of nanoscale switches and increased leakage power of these switches. Furthermore, programmability of the implantable device in order to dynamically adapt to the changing functional requirements is becoming an important design parameter. Finally, security of the recorded information, when the signal is wirelessly transmitted outside the body, is emerging as an important requirement.

We note that these issues can be solved in an efficient manner by exploiting the nature of the data and/or the signal processing algorithm under consideration, instead of using conventional design techniques or hardware components such as micro-controller or DSP processor. In this paper, we focus on the emerging issue of information security in implantable systems. With pervasive use of implantable monitoring systems and increasing need for wireless transmission of the sensed data, these systems are being increasingly vulnerable to snooping attack by potential adversaries. However, data protection needs to come at extremely low power and area overhead, which is difficult to achieve by using conventional data encryption solutions. We propose a novel low-cost multi-level key-based obfuscation technique that exploits the nature of the recorded biological signal to protect it from possible attacks. The proposed approach scrambles the data at multiple levels using separate secret keys at minimal hardware overhead, compared to a conventional cryptography technique, namely advanced encryption standard (AES).

II. Design Issues

The major design challenges for signal processing electronics in an implantable system are highlighted in Fig. 1. The first two parameters, die-area and power, are important

S. Narasimhan, X. Wang and S. Bhunia are with the Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, Ohio, USA {sxn124, xxw58, skb21}@case.edu

978-1-4244-4124-2/10/$25.00 ©2010 IEEE

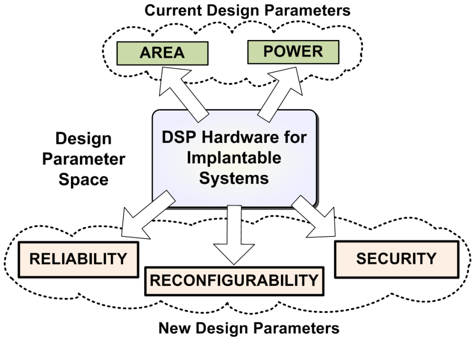


Fig. 1. Design parameters for implantable systems. Reliability, reconfigurability and security are emerging as important design parameters.

and have been widely addressed by existing work. The die-area is important since the packaged system should have a small form factor. On the other hand, ultra-low power operation is required to enable long-term operation using the embedded battery as well as to avoid tissue damage due to power-density induced temperature rise. The other emerging issues in bio-implantable systems are reliability, reconfigurability and security.

1) Reliability: While nanotechnology (sub-100nm feature size) is an attractive option for the design of DSP unit in implantable systems, it suffers from reduced reliability and yield due to lower noise margin and variations in process parameters. Furthermore, temporal parameter variations (due to environmental variations and device aging effects) also affect robustness of operation. Low power design techniques such as voltage scaling, gate sizing and power gating accentuate the reliability issues. Conventionally one needs to follow a worst-case design approach to avoid failures due to process and temporal variations, which are intrinsic to nanometer technologies. However, such a pessimistic design approach considerably compromises power dissipation and die area. The reliability issues can be handled at low area and power overhead by exploiting the nature of the signal processing algorithm. One effective solution to address the reliability and yield issues is to use variation-resilient design techniques such as *Preferential Design* [5]. In this approach, the critical components in terms of performance are designed with more relaxed timing margin than non-critical ones. Possible variation-induced failures are confined to non-critical components of the system, thus allowing graceful degradation in performance under variations. Further, due to higher timing margin in critical components, the system becomes suitable for application of low power design techniques.

2) Reconfigurability: The next important design issue for implementing signal processing algorithms in implantable systems is *reconfigurability* for tuning functionality during deployment. Variations in the nature of biological signals from patient-to-patient and temporal variations in signal and noise characteristics for the same patient requires calibration and tuning of the various parameters or changing functionality. To achieve this, one can use software reconfigurability, where the algorithm is coded in an embedded microprocessor [2] or use hardware reconfigurable platforms such as Field Programmable Gate Array (FPGA). But both options are not possible under the area and power budget allowable by implantable systems. This necessitates the investigation of alternative reconfigurable architectures such as Memory Based Computing (MBC) [6], which can be used to implement a particular algorithm within the area, power and performance bound, but keeps the option of reconfigurability in order to suit patient-to-patient and temporal variability. Such reconfigurable computing framework uses a dense memory array as underlying computing element, leading to over 2X power reduction at iso-performance compared to state-of-the-art FPGA [6]. In order to satisfy the area and power constraints, one can use a judicious mix of reconfigurable memory-based computing and custom logic blocks which can give us the required adaptability.

3) Security: Security is another emerging issue in implantable system design. In the rest of the paper, we will describe a lightweight solution for protecting the system from malicious attacks by an adversary.

III. Light-Weight Security Solution

Recently, researchers have articulated security concerns about implantable medical devices which use wireless communication protocols [7]. The lack of authentication and integrity mechanisms put patients at risk from attack by anyone with a transceiver. Various security threats considering different aspects of the implantable device usage have been described in [8]. For example, the security of data communicated over a wireless interface can be compromised. An attacker can eavesdrop on the wireless communication channel in order to get access to sensed data. For implantable stimulation/drug-delivery systems, the attacker can initiate malicious control signals leading to malfunction, injury or even death. For the first scenario, it is useful to encrypt the data based on a secure key to prevent it from being understood by snoopers. For the second scenario, one can use an authentication key to ensure that the input commands are coming from a safe and valid source.

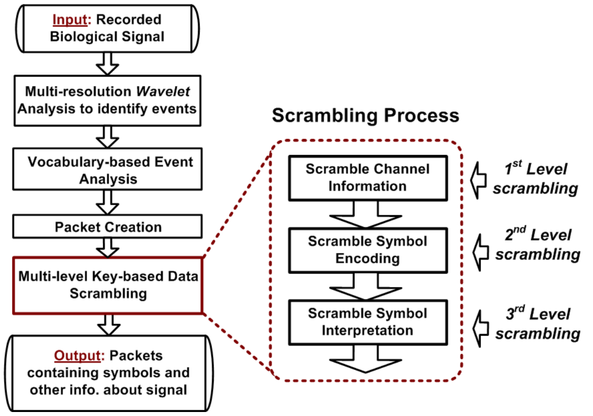

Input: Recorded Biological Signal
Multi-resolution Wavelet Analysis to identify events
Vocabulary-based Event Analysis
Packet Creation
Multi-level Key-based Data Scrambling
Output: Packets containing symbols and other info. about signal
Scrambling Process
Scramble Channel Information
Scramble Symbol Encoding
Scramble Symbol Interpretation
1st Level scrambling
2nd Level scrambling
3rd Level scrambling

Fig. 2. Main steps for biological signal processing algorithm. The inclusion of security as a part of the main flow enables us to use low-overhead design approaches to achieve security against possible attacks. The key-based scrambling for data obfuscation can be performed at three levels to achieve higher protection against snooping attacks.

Next, we describe a light-weight obfuscation scheme for countering the first attack scenario by by exploiting the nature of the algorithm, as an alternative to conventional resource-hungry encryption. The main steps for the proposed hierarchical key-based data security algorithm are shown in Fig. 2. The inclusion of security as a main step of the algorithm helps us achieve a low-overhead solution. By using multiple levels of scrambling to obfuscate the neural data using several keys, we also achieve high levels of information security. For the neural spike detection and vocabulary-based analysis which we consider as an example implantable system, the following properties are particularly useful. The neural signal consists of low frequency signal, which is characterized by intermittent bursts of sharp events called spikes (which allows us to use wavelet transform for de-noising) and the repetitive nature of these patterns helps us develop a vocabulary-based algorithm [4] for neural data compression and efficient neural data analysis. We can obfuscate the data in an output packet by scrambling the channel number, the alphabet symbol used for vocabulary encoding and the wavelet coefficients used to describe a particular symbol. We use different encodings for different parts of a packet and use a key-based approach for scrambling the data in each part so that only a transceiver with the correct key can accurately decode the signal.

Permutation is a traditional technique to scramble various types of signals for providing security during transmission. For example, a scrambling algorithm based on permutation of Fast Fourier Transform (FFT) coefficients was proposed for speech encryption in [10]. However, a logic implementation of the permutation algorithm consumes too much area and power to be suitable for implantable circuits. On the other hand, if we consider all possible permutations of a particular set of data, and use a pseudo-random key of sufficient length to denote the order of permutation to be used for encoding each packet, it allows us to explode the search space for an attacker using brute-force techniques to break the scheme. Here, we use two levels of encoding, by using one key to denote the particular order of scrambling and a second level of key to change the order for each packet. Also, the hardware overhead can be minimized by storing only the required permutations for a particular value of the key. For example if the key has 16 characters, we need to consider storage of 16 different permutations of the symbols. The storage is done in a 16-row memory, which is actually a lookup table. Each row in the lookup table is a particular permutation of all the neural signal symbols.

1) Security Analysis: For analyzing the number of trials taken to break the encoding by brute-force approaches, we consider one level of key-based obfuscation used for encoding the alphabet symbols of the vocabulary. Initially, the vocabulary symbols correspond to the binary encoding of the location of each symbol. If we consider the vocabulary size to be 32, we need 5 bits to encode the symbol. We can arrange 32 values in $32! = 2.63 \times 10^{35}$ possible permutations. If we have to choose 16 out of these possible permutations, we have $(32!)^{16} = 5.3 \times 10^{566}$ possible key values if we allow repetitions. The number of trials for identifying the correct key by brute force is impossibly large compared to $2^{128} = 3.4 \times 10^{38}$, which is the number of trials required to break a 128-bit key used in an AES cipher. With increasing key length, the complexity of breaking the key by brute force increases exponentially. Next, we show that the area overhead increases only linearly with the key length.

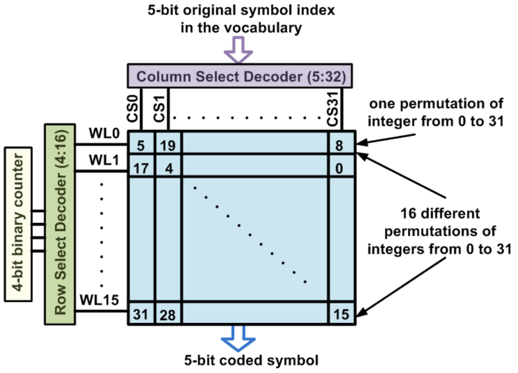
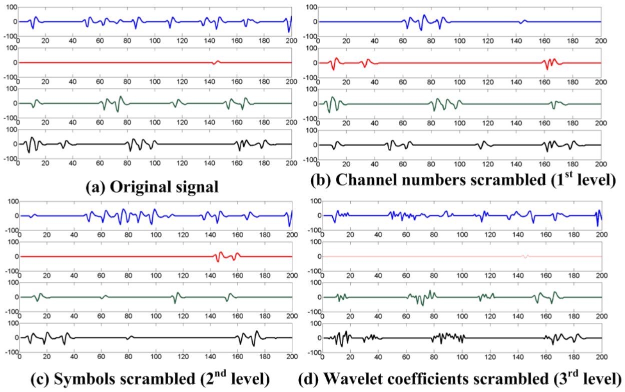

5-bit original symbol index in the vocabulary
Column Select Decoder (5:32)
CS0
CS1
CS31
one permutation of integer from 0 to 31
WL0
WL1
WL15
5
19
8
17
4
0
31
28
15
16 different permutations of integers from 0 to 31
4-bit binary counter
Row Select Decoder (4:16)
5-bit coded symbol

Fig. 3. Lookup table structure to implement one level of data scrambling.

2) Hardware Overhead: If we have M neural signal symbols in the vocabulary, we have $M!$ different permutations in the lookup table (LUT). Generally, the value of M is nontrivial, making $M!$ too large for implementation. However, we can choose some of the $M!$ permutations to build the LUT, which will not affect the security level too much, given that the number of permutations is not less than the size of the key. Here we choose $M = 32$, which, from our experience, is a reasonable value for encoding symbols in neural signals. We set our key length to be 16 characters, where each character denotes one of the 32! permutations in total. If we store all the permutations in the lookup table while taking into consideration the fact that each symbol is represented with 5 bits, the total size of the lookup table would be $32! \times 32 \times 5$ bit, which is 5.14×10^{33} kB, which is impossible to implement. However, we note that since the key is one-time configurable for each device, we need to store only 16 out of all the 32! permutations in the lookup table. Now, the size of the lookup table becomes $16 \times 32 \times 5$ bits ($= 2560$ bits), which is a feasible memory size for implantable circuits. By custom design, we can reduce the overhead for the peripheral circuitry needed for accessing the different values stored in the memory. In fact, the peripheral hardware we need is merely a 4 bit counter

TABLE I
Comparison of hardware overhead for our scheme with AES.

Circuit / Sub-circuits		Area (μm^2)	Power (μW)
Conventional Scheme	AES (1 V, 100 MHz)	76502.95	20530
	AES (0.7 V, 2 MHz)	76502.95	410
Proposed Data Scrambling Scheme	5:32 decoder	189.23	0.03
	6:64 decoder	325.23	0.04
	7:128 decoder	570.29	0.05
	4-bit binary counter	120.96	0.04
	4:16 decoder	103.39	0.02
	SRAM (All 3 LUTs)	13655.90	30.87
	Total	**14965.01**	**31.05**
Percentage Reduction		80.44%	92.4%



(a) Original signal

(b) Channel numbers scrambled (1st level)

(c) Symbols scrambled (2nd level)

(d) Wavelet coefficients scrambled (3rd level)

Fig. 4. The effect of scrambling on four channels of neural signal. After each level of scrambling, the output signal gets obfuscated, where the information content is completely distorted.

with 4-to-16 bit decoder for row selection, since we need to access the memory serially one row at a time. The random access of each symbol from a row is managed by using a 5-to-32 bit column select decoder, which also allows us to share the sense amplifiers between the same bits of each symbol, since only one symbol needs to be read at a time. The hardware implementation scheme for symbol scrambling is shown in Fig. 3. The look-up tables for encoding different parts of the packet using different keys can be combined into a single memory array with shared serial-row access circuitry. We present the area and power results in the next section.

IV. Simulation Results

We created SPICE netlists corresponding to different circuits and performed power simulations using 70nm Predictive Technology Model (PTM) [12]. For comparison purposes, we take a low-power implementation of an AES circuit as an example. Assume we have 64 neural signal channels to record, and the sampling frequency is 25 kHz. We have 2.5×10^6 samples to process per second. Equivalently, we need to process 25 samples in 10^{-5} seconds. If each sample is an 8 bit wavelet coefficient, we have $25 \times 8 = 200$ bits to process in every 10^{-5} seconds. Since 10-stage pipelined AES can process 128 bits in 10 cycles, for 200 bits, around 20 operating cycles are needed. Therefore the AES working frequency should be $20/(10^{-5}) = 2$ MHz. We used voltage scaling and threshold voltage scaling to reduce dynamic and leakage power. For the proposed approach, we need three column select decoders of different sizes corresponding to the three different regions of the packet. Also, the LUT size was taken to be $16 \times ((32 \times 5) + (64 \times 6) + (128 \times 7)) = 23040$ bits $= 2.8$ kB. Compared to the huge power consumption of *low-power AES*, we obtain only 31μW for an SRAM-based implementation of the look-up table. The area and power values for conventional AES, low-power AES and the different sub-circuits used for our scheme are shown in Table I. The results of data scrambling at multiple levels for a 4-channel neural signal are shown in Fig. 4. The signal is recorded from four nerves in the buccal ganglion of a sea-slug (*Aplysia californica*). It can be observed that the obfuscated data still contains spike-like waveforms but the information is completely mis-represented, causing the attacker to gain only wrong information. However, the transceiver with the correct key can accurately de-obfuscate the data and get the correct information.

V. Conclusion

In this paper we have analyzed the emerging design issues such as reliability, configurability and security in realizing the digital signal processing unit in implantable systems. With increasing need to incorporate advanced signal processing capability in implantable devices, these parameters would be increasingly important to consider along with area and power criteria. We have noted that instead of using conventional design approaches, one can exploit the nature of signal processing algorithms and the recorded signals to address the emerging design requirements under tight power/area constraints. We have extended this design philosophy to implement an ultra light-weight data protection algorithm for wireless implantable systems. The proposed algorithm exploits the bursty nature of the biological data with repeating symbols to achieve effective obfuscation of recorded data during the encoding and packet creation process. We have shown that, compared to existing encryption algorithms, such an obfuscation process can significantly reduce the hardware overhead. Future investigation will include application of the security solution to other implantable system applications and hardware validation of the algorithms through test chip fabrication and measurement.

References

[1] R.J. Chandler, S. Gibson, V. Karkare, S. Farshchi, D. Markovic and J.W. Judy, "A system-level view of optimizing high-channel-count wireless biosignal telemetry," 31^{st} *Annual International Conference of the IEEE EMBS*, 2009.

[2] R. Allan, "Medtronic sets the pace with implantable electronics," *Electronic Design*, Oct. 2003.

[3] Z.S. Zumsteg *et al*, "Power feasibility of implantable digital spike sorting circuits for neural prosthetic systems," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 3, 2005.

[4] S. Narasimhan, M. Cullins, H.J. Chiel, and S. Bhunia, "Wavelet-based neural pattern analyzer for behaviorally significant burst pattern recognition," 30^{th} *Annual International Conference of the IEEE EMBS*, 2008.

[5] S. Narasimhan, H.J. Chiel and S. Bhunia, "A preferential design approach for energy-efficient and robust implantable neural signal processing hardware," 31^{st} *Annual International Conference of the IEEE EMBS*, 2009.

[6] S. Paul and S. Bhunia, "Computing with nanoscale memory: model and architecture," *IEEE/ACM International Symposium on Nanoscale Architecture (NANOARCH)*, 2009.

[7] V. Stanford, "Pervasive health care applications face tough security challenges," *IEEE Pervasive Computing*, 2002.

[8] D. Halperin, T.S. Heydt-Benjamin, K. Fu, T. Kohno and W.H. Maisel, "Security and privacy for implantable medical devices," *IEEE Pervasive Computing, Special Issue on Implantable Electronics*, 2008.

[9] R.S. Chakraborty, S. Narasimhan and S. Bhunia, "Hardware Trojan: Threats and emerging solutions", *HLDVT*, 2009.

[10] S.E. Borujeni, "Speech encryption based on fast Fourier transform permutation," *IEEE International Conference on Elcetronics, Circuits and Systems*, 2000.

[11] F. Delgosha and F. Fekri, "Stream cipher using finite-field wavelets," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005.

[12] Predictive Technology Model, [Online] http://www.eas.asu.edu/∼ptm/