31st Annual International Conference of the IEEE EMBS
Minneapolis, Minnesota, USA, September 2-6, 2009

A Preferential Design Approach for Energy-Efficient and Robust Implantable Neural Signal Processing Hardware

Seetharam Narasimhan, Hillel J. Chiel and Swarup Bhunia

Abstract— **For implantable neural interface applications, it is important to compress data and analyze spike patterns across multiple channels in real time. Such a computational task for online neural data processing requires an innovative circuit-architecture level design approach for low-power, robust and area-efficient hardware implementation. Conventional microprocessor or Digital Signal Processing (DSP) chips would dissipate too much power and are too large in size for an implantable system. In this paper, we propose a novel hardware design approach, referred to as "Preferential Design" that exploits the nature of the neural signal processing algorithm to achieve a low-voltage, robust and area-efficient implementation using nanoscale process technology. The basic idea is to isolate the critical components with respect to system performance and design them more conservatively compared to the noncritical ones. This allows aggressive voltage scaling for low power operation while ensuring robustness and area efficiency. We have applied the proposed approach to a neural signal processing algorithm using the Discrete Wavelet Transform (DWT) and observed significant improvement in power and robustness over conventional design.**

I. INTRODUCTION

Miniaturized implantable systems provide an important interface with the central nervous system for interpreting and engineering its activity in terms of its communications with body parts [1]. Fig. 1(a) shows a typical interface of the signal processing system with a micro-electrode array and analog signal conditioning and transceiver electronics. A recently proposed neural signal analysis algorithm [2] is explained with the flow diagram in Fig. 1(b). Input to the algorithm is the digitized recorded neural signal, broken into fixed-size overlapping windows. The output is compressed neural data encoded as packets containing information about the detected spikes. We use multi-resolution wavelet analysis [3] of recorded signal to de-noise the data, detect and sort spikes and recognize behaviorally meaningful burst pattern across multiple channels from *in vivo* neural and muscular recordings. For hardware implementation of the digital signal processing block, nanoelectronics offers great potential due to its tera-scale integration density, low switching power and high performance. However, it also brings a number of design challenges, such as exponential increase in leakage power and lack of robustness due to device parameter variations [4].

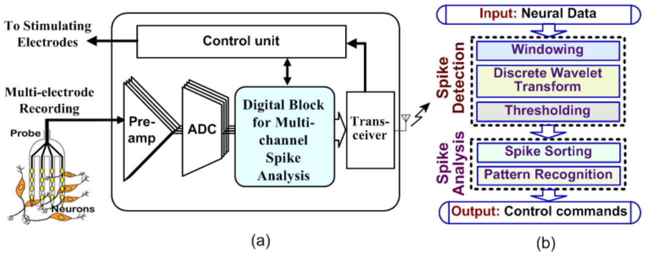


Fig. 1. (a) Block diagram of the overall system for neural recording and stimulation. (b) The flowchart of the neural signal processing algorithm.

In this paper, we propose a novel hardware design style for neural signal processing algorithms to achieve low-power, area-efficient and robust implementation using nanoscale devices. Due to the quadratic dependence of power on supply voltage, voltage scaling [5] has emerged as a popular low-power design approach. However, at scaled voltage, digital circuits can suffer functional failure. Reduced robustness of operation at scaled voltage is accentuated by increased variations in device parameters at nanometer technology nodes. The proposed design approach exploits the nature of the neural signal processing algorithm to achieve low-power operation while maintaining robustness. In this scheme, critical components of the circuit, in terms of system performance (such as output signal quality), are designed with a relaxed timing margin as compared to non-critical ones. With aggressive voltage scaling for low power and increased process variations, possible functional failures are confined to non-critical components of the system, thus allowing graceful degradation in performance. We have applied the proposed design approach in implementing the wavelet-based spike detection module. The concept can be applied hierarchically at bit-level by assigning more design margin to the most significant bits compared to the least significant ones.

II. RELATED WORK

Many researchers have previously addressed the issues of designing the analog front-end circuitry as well as algorithms for offline signal analysis. Some recent investigations, which have addressed the design of signal processing algorithms and hardware for real-time spike detection and analysis are described here. Harrison proposed a simple thresholding scheme for on-chip spike detection using analog comparators [6]. Olsson *et al* [7] proposed an on-chip data compression circuit which detects spikes using a simple adaptive thresholding scheme and transmits their amplitudes. Guillory *et al* [8] designed a 100-channel wired system for real-time

Dr. Chiel was supported by the NIH Grant no. NS047073.

S. Narasimhan and S. Bhunia are with the Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH 44106, USA {sxn124, skb21}@case.edu

H.J. Chiel is with the Departments of Biology, Neuroscience and Biomedical Engineering, Case Western Reserve University, Cleveland, OH 44106, USA hjc@case.edu

978-1-4244-3296-7/09/$25.00 ©2009 IEEE

spike detection. However, they used a commercial DSP for performing most of their signal processing, which can prove to be too power hungry and too large for bio-implantable systems. Wavelet-based spike detection [9] and hardware implementation of wavelet transform for implantable neural interface applications [10] have been investigated before in the context of compression of multi-channel data. Most of these hardware implementations use conventional circuit design styles. In this paper, we explore a novel architecture/circuit co-design paradigm for implementing real-time neural signal analysis hardware.

III. Choice of Appropriate Architecture

First, we need to investigate appropriate architecture for low-power DWT implementation. Based on our design objectives of power and area minimization, a sequential implementation of the integer arithmetic lifting approach, as described in [10], seems most viable. Equation 1 contains the five basic steps for computing an approximation 'a' and a detail 'd' coefficient from the even and odd input data samples, represented as f_0 and g_0, using one level of DWT decomposition and the '$sym4$' wavelet basis function.

$$\begin{aligned}
&step\ 1: g_1(i) &&= g_0(i) &&+ C_0 * \ f_0(i) && \\
&step\ 2: f_1(i) &&= f_0(i) &&+ C_1 * \ g_1(i+1) &&+ C_2 * \ g_1(i) \\
&step\ 3: g_2(i) &&= g_1(i) &&+ C_3 * \ f_1(i) &&+ C_4 * \ f_1(i-1) \\
&step\ 4: a(i) &&= f_1(i) &&+ C_5 * \ g_2(i) &&+ C_6 * \ g_2(i-1) \\
&step\ 5: d(i) &&= g_2(i) &&+ C_7 * \ a(i+1) && \qquad (1)
\end{aligned}$$

Since each step requires similar computational hardware, we can identify a "processing element" (PE) as the basic computational block for the DWT module. It consists of two signed multipliers and two signed adders, as shown in Fig. 2. Since we are dealing with quantized integer representation of numbers, we need to make adjustments for the quantization of the filter coefficients at each step ("$TRUNC$").

For the overall DWT architecture, we can choose a parallel implementation, as shown in Fig. 3, where the windowing module buffers enough samples before the lifting operations take place. We have implemented an overlapping window scheme (8 samples overlap in a 72 sample window) in order to avoid missing spikes on the window edges. It takes 5 clock cycles for the five steps to be computed. The remaining cycles are used for other processing steps like thresholding and for processing data from multiple channels. The second scheme [10] involves a sequential computation of the five lifting steps by folding them back onto the same PE. This requires less registers and the latency is also reduced. The final structure of the control path, after register and multiplexer minimization, is shown in Fig. 4. It is worth noting that the PE needs to be operated at five times the clock speed in order to get one set of approximation and detail coefficients at the end of each cycle, resulting in higher clock speed and hence, higher power. For most biological signal processing, the frequency range of interest is in the order of a few (~ 10) KHz. If we need to compute wavelet coefficients for 100 channels, the maximum clock frequency of operation can be estimated to be $(100 * 5 * 10\ KHz) = 5\ MHz$. The two architectures described above represent a trade-off between area and power dissipation. The second architecture, although area-optimal, consumes higher power because of increased operating frequency, while the parallel architecture allows scaling of voltage and frequency to achieve quadratic and linear reduction in power, respectively. In the following section, we investigate circuit-level optimization techniques to enable low-power and robust operation.

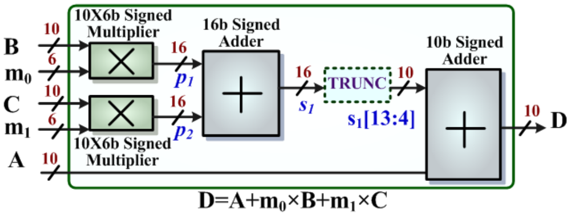


Fig. 2. Architecture of the main processing element for each step of the lifting wavelet transform.

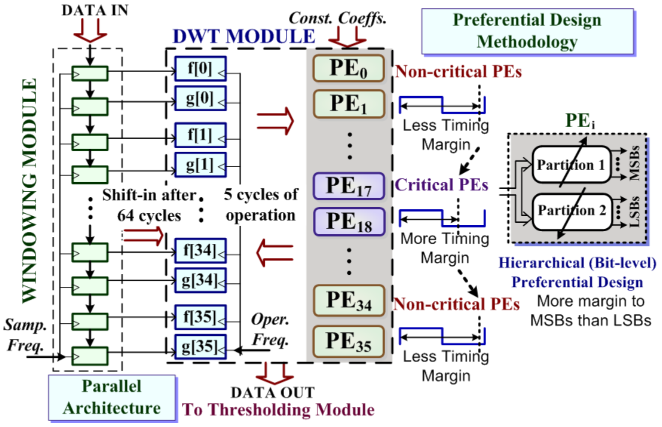
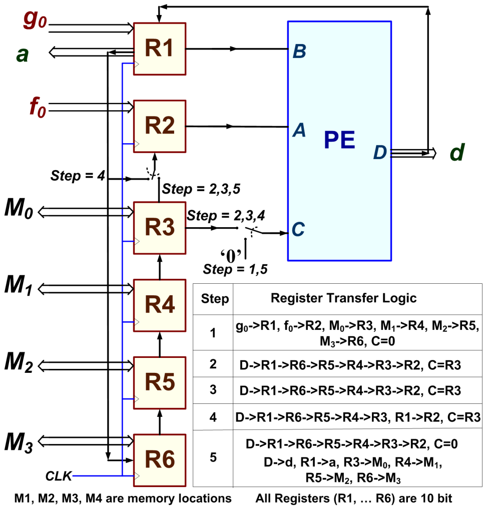


Fig. 3. Parallel architecture for DWT module. The proposed hierarchical preferential design approach is also illustrated.



Step	Register Transfer Logic
1	g_0->R1, f_0->R2, M_0->R3, M_1->R4, M_2->R5, M_3->R6, C=0
2	D->R1->R6->R5->R4->R3->R2, C=R3
3	D->R1->R6->R5->R4->R3->R2, C=R3
4	D->R1->R6->R5->R4->R3, R1->R2, C=R3
5	D->R1->R6->R5->R4->R3->R2, C=0 D->d, R1->a, R3->M_0, R4->M_1, R5->M_2, R6->M_3

Fig. 4. Register transfer logic for the sequential design.

IV. Circuit-Level Optimization

Low-power and robustness of operation typically impose contradictory design requirements. In logic circuits, the principal failure mechanism under device parameter variations at nanoscale technologies is *delay failure*, which occurs when the max path delay of a circuit exceeds the clock period. Low power design using voltage scaling accentuates delay failure probability under variations. To avoid these delay failures, conventionally, one needs to follow a worst-case design approach. However, such a design approach considerably compromises power dissipation and die area. On the other hand, in the case of a nominal design, any variation-induced failure may cause drastic changes in the outputs which are more critical in terms of signal quality. Hence, we propose a "*Preferential Design*" methodology in order to minimize area and power while allowing graceful degradation in signal quality under extreme parameter variations.

In this scheme (see Fig. 3), we first identify the critical processing elements (PEs) which contribute more to the output signal quality than others. These critical PEs need to be designed conservatively to provide them with a large delay margin, thus ensuring their robust operation. On the other hand, the non-critical PEs can be designed to reduce area and power. By ensuring that the failure is always confined to non-critical PEs, we achieve graceful degradation in output quality. Note that conservative design of the critical PEs and aggressive design of the non-critical ones allows significant area reductions over the worst-case design approach while achieving better output signal quality than area-optimal or nominal design. The skewing of delay margins across critical and non-critical components can be realized by using a constraint-driven logic synthesis or gate sizing approach. Such a preferential design approach can be easily integrated into the existing automatic design synthesis flow.

The proposed design paradigm can be applied hierarchically to different levels of design abstraction. Along with application of different timing margins to different PEs, one can assign different margins to different output bits inside a PE (as shown in Fig. 3). Since the most significant bits (MSBs) of a PE contribute more towards the output quality compared to the least significant bits (LSBs), we can assign higher margin to the MSBs and confine the delay failures to the least significant bits of the least significant components, thus ensuring minimal impact on output with voltage scaling or parameter variations. It is to be noted that the proposed preferential design approach can be used for other signal processing blocks in Fig. 1 such as spike sorting and pattern recognition, where we can isolate the critical computing blocks from non-critical ones.

TABLE I

Area and power values for different design methodologies.

Cases	Delay Cons. (ns)	Area (mm^2)	Area Ovhd (%)	Power (μW)
A (Area-optimal)	4.7	0.195	-8.64	311.4
B (Intermediate I)	4.5	0.202	-5.61	311.8
C (Nominal)	4.4	0.213	0.00	312.9
D (Intermediate II)	4.3	0.222	4.22	313.5
E (Quality-optimal)	4.0	0.230	7.88	314.4
F (Pref_des)	4.0 - 4.7	0.214	0.19	312.9

V. Simulation Results

We designed the wavelet engine following the parallel architectural scheme described in Section III. Register-Transfer-Level (RTL) Verilog code was written and functionally verified. The circuits were synthesized using *Synopsis Design Compiler* using a LEDA standard cell library at TSMC 250nm technology node. Power and area values are obtained by scaling the netlist to 70nm [12] technology node.

To investigate the effect of supply voltage scaling and process variations (modeled as V_{th} variations) on the probability of delay failure, we considered nine different cases for three supply voltages ($1.0V$, $0.9Vand$ $0.8V$) and three increasing degrees of process variation (0%, 10% and 20%), where $(1V, 0\%)$ is considered as the nominal operating condition for design. To achieve the Preferential Design, the thirty-six instances of the PE were chosen with different timing margins (*Delay Cons.*), with the more critical blocks (the ones in the middle are more critical compared to the ones at the edges) assigned with maximum timing margin. The area and power values for the entire spike detection engine for different cases are presented in Table I. The area overhead values are computed as a percentage of the area in the nominal case (*Case C*). It can be observed that *Case A* has the least area, while *Case E*, the worst-case design, has huge area overhead. We have kept two intermediate cases (*Case B* and *D*) to show the trend. The Preferential Design (*Pref_des*) *Case F* has similar area and power as the nominal case.

To investigate the effect on the signal quality, we used a five sec train of noise-free extracellular neural spikes (Hodgkin Huxley model) sampled at $10KHz$ with an amplitude range of $+75\mu V$ to $-42\mu V$ (using 8-bit signed quantization) and simulated the algorithm in MATLAB [11]. The impact of voltage scaling and process variations is simulated by introducing uniform random noise at each step

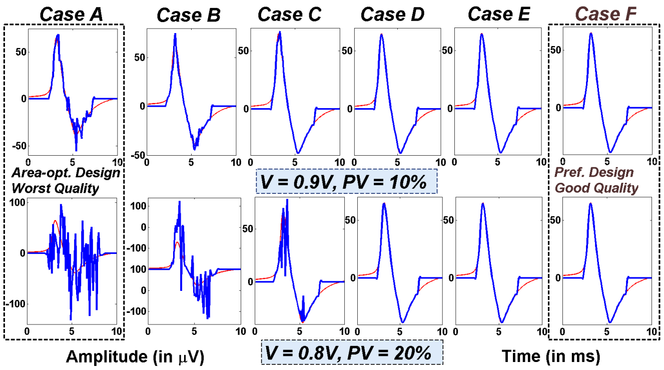


Fig. 5. Spike reconstruction quality for different design methodologies. Case F (*Pref_des*) gives much better quality under voltage scaling and process variations compared to Case C (nominal design) or Case A.

TABLE II

Qual VALUES IN *dB* FOR DIFFERENT CASES UNDER DIFFERENT VOLTAGE AND PROCESS VARIATION CONDITIONS.

Supply Voltage	V $= 1.0V$			V $= 0.9V$			V $= 0.8V$		
Proc. Var.	0%	10%	20%	0%	10%	20%	0%	10%	20%
A (Area-optimal)	∞	20.45	8.51	14.72	13.95	7.51	-4.20	-4.29	-4.51
B (Intermediate I)	∞	24.25	12.37	18.44	17.48	11.56	0.13	-0.03	-0.10
C (Nominal)	∞	31.93	18.93	24.95	22.51	17.91	6.38	6.16	5.94
D (Intermediate II)	∞	67.05	66.08	70.06	56.95	50.77	39.32	37.95	35.28
E (Quality-optimal)	∞	∞	∞	∞	∞	∞	∞	∞	∞
F (Pref_des)	∞	**59.27**	**49.09**	**59.05**	**57.76**	**51.67**	**46.71**	**42.97**	**32.19**

of computation through the PE. The signal reconstructed after spike-detection in the nominal case is considered as the reference signal s, and in all other cases as signal + noise ($sn = s + n$). The quality degradation is computed as:

$$Qual = 10 * \log_{10} \frac{\sum_{i=1}^{T} s(i)^2}{\sum_{i=1}^{T} (sn(i) - s(i))^2}, \qquad (2)$$

where T is the total number of samples in the signal. We present *Qual* values for the nine different cases, considering five different designs along with preferential design, in Table II. Under nominal operating conditions all designs have no noise, hence the corresponding *Qual* values are ∞. This also holds for the worst-case design (*Case E*). The degradation in signal quality can also be observed in Fig. 5, where one randomly chosen reconstructed spike (superimposed over the original signal in red) is shown for the six different design styles for two operating conditions. It can be inferred from these results that *Case A* is poor in terms of signal quality, *Case E* design is poor in terms of area, while preferential design (*Case F*) has better signal quality compared to the nominal design (*Case C*) at almost iso-area and iso-power.

To investigate the effectiveness of preferential design at the bit level, we considered the architecture in Fig. 4, which uses a single PE in a time-multiplexed fashion. If we synthesize the PE with nominal delay constraint to all the bits, the path delays increase consistently from LSB to MSB, as shown in Fig. 6. On the other hand, we can enforce an opposite trend in the delay distribution using a delay-constrained synthesis process. This helps in achieving higher signal quality under variations since the failures are now confined to the LSBs of the PE. Such a bit level preferential design can also be applied hierarchically to the first architecture by designing the MSBs of the less critical PEs more robust than the LSBs.

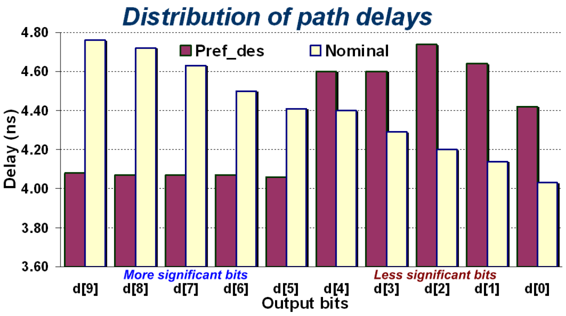


Fig. 6. Distribution of path delays for different output bits of a single PE in both nominal design and bit-level *Preferential Design*.

VI. CONCLUSION

We have presented a novel design methodology for low-power, robust and area-efficient implantable neural signal processing hardware. The proposed methodology exploits the nature of the signal processing algorithm to preferentially assign more delay margins to the critical components compared to the less critical ones. Such an approach allows aggressive voltage scaling while maintaining robustness of operation under process variations. Simulation results for a signal processing step show that compared to conventional design, the proposed approach can achieve significant savings in area and power, which are important design parameters for bio-implantable circuits. Future work will involve application of the approach to other neural signal processing steps.

REFERENCES

[1] M.A.L. Nicolelis, "Actions from thoughts", *Nature*, vol. 409, 2001.

[2] S. Narasimhan, M. Cullins, H.J. Chiel and S. Bhunia, "Wavelet-based neural pattern analyzer for behaviorally significant burst pattern recognition", *IEEE Engineering in Medicine and Biology Society Conference*, 2008.

[3] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley Cambridge Press, 1996.

[4] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A.Keshavarzi and V. De, "Parameter variations and impact on circuits and micro architecture", *Design Automation Conference*, 2003.

[5] R. Gonzalez, B.M. Gordon and M.A. Horowitz, "Supply and threshold voltage scaling for low power CMOS", *IEEE Journal of Solid-State Circuits*, pp. 1210-1216, 1997.

[6] R. Harrison, "The design of integrated circuits to observe brain activity", *Proc IEEE*, vol. 96, no. 7, pp. 1203-1216, July 2008.

[7] R.H. Olsson III and K.D. Wise, "A three-dimensional neural recording microsystem with implantable data compression circuitry", *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, Dec 2005.

[8] K.S. Guillory and R.A. Normann, "A 100-channel system for real time detection and storage of extracellular spike waveforms", *Journal of Neuroscience Methods*, vol. 91, no. 1-2, pp. 21-29, Sep 1999.

[9] V.J. Samar, A. Bopardikar, R. Rao and K. Swartz, "Wavelet analysis of neuroelectric waveforms: A conceptual tutorial", *Brain and Language*, vol. 66, no. 1, pp. 7-60, Jan 1999.

[10] K.G. Oweiss, A. Mason, Y. Suhail, A.M. Kamboh and K.E. Thomson, "A scalable wavelet transform VLSI architecture for real-time signal processing in high-density intra-cortical implants", *IEEE Trans. on Circuits and Systems*, pp. 1266-1278, Jun 2007.

[11] Matlab Wavelet Toolbox, Version 2.1. Mathworks, [Online] http://www.mathworks.com/products/wavelet

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