Ultralow-power Data Compression for Implantable Bladder Pressure Monitor: Algorithm and Hardware Implementation

Robert Karam[1,2], Steve Majerus[2], Dennis Bourbeau[3], Margot S. Damaser[2,4,5], and Swarup Bhunia[1]
Email: robkaram@ufl.edu, sjm18@case.edu, dbourbeau@fescenter.org, damasem@ccf.org, swarup@ece.ufl.edu
[1] Dept. of Electrical and Comp. Eng., University of Florida, Gainesville, FL 32608
[2] Advanced Platform Technology Center, L. Stokes VA Hospital, Cleveland, OH 44106
[3] Functional Electrical Stimulation Center, L. Stokes VA Hospital, Cleveland, OH 44106
[4] Dept. of Biomedical Eng., Lerner Research Institute, Cleveland, OH 44106
[5] Glickman Urology and Kidney Inst., Cleveland Clinic Foundation, Cleveland, OH 44106

***Abstract*—Urinary incontinence, overactive bladder, and other dysfunctions of the lower urinary tract are conditions which affect millions worldwide, imposing a high financial burden and greatly affecting quality of life. Diagnosis of these conditions can be facilitated by monitoring bladder activity over time. Recent work has demonstrated the feasibility of device implantation for chronic bladder pressure monitoring, which can help to improve existing diagnostic techniques. For wireless implants, a significant portion of the implant power consumption results not from pressure sensing, but rather from data transmission. In this paper, we present a novel algorithm which is designed to perform efficient, on-chip compression of bladder pressure data with tunable quality, resulting in significant overall power savings. We validate our approach by applying the algorithm to prerecorded bladder pressure data from 14 human subjects, and demonstrate high average compression ratios (∼5x), leading to similar reductions in transmission power draw, with low reconstruction error (RMSE ≈ 1.09). An ultralow-power hardware implementation of the proposed algorithm is obtained by synthesizing the design with TSMC $0.18\,\mu m$ technology, yielding an area of $1.34\,mm^2$ and average power of $2.3\,nW$ using low power design techniques. To our knowledge, this is the first example of dedicated on-chip compression for bladder pressure data designed for an ultralow-power biomedical implant.**

I. Introduction

Lower urinary tract dysfunction (LUTD) encompasses a number of debilitating conditions, including overactive bladder and urinary incontinence, which can significantly reduce the quality of life and impose a large financial burden for millions worldwide. For individuals with more complex pathophysiologies, diagnosis of LUTD can be accomplished through the use of urodynamic testing, an acute procedure which involves filling the bladder with saline, and monitoring bladder pressure with time. In some cases, the faster-than-physiological infusion rate of saline into the bladder can produce results that may not be representative of bladder function during natural filling, while the catheters used for infusion and pressure measurements can affect bladder activity, further confounding results.

We have developed a wireless bladder pressure monitor capable of 100 Hz sampling of bladder pressure and real-time telemetry [1] (Fig. 1(a)). To maintain a small implant size, device power consumption must be minimized. Because data transmission from the device is the dominant power draw, new strategies for reducing the data rate without sacrificing sensor accuracy are critical. In this paper, we present a novel algorithmic approach to produce highly-compressed representations of human bladder pressure data. In particular, we make the following novel contributions: (1) We present a tunable, lossy compression algorithm which leverages the signal properties of human bladder pressure data to perform effective data compression; (2) We validate the approach using prerecorded bladder pressure data from 14 human subjects, and show that the algorithm can achieve high compression ratios with low reconstruction error; and (3) We provide hardware implementation details with optimizations for ultralow power consumption, as well as area and power results from synthesis and layout in TSMC $0.18\,\mu m$ technology.

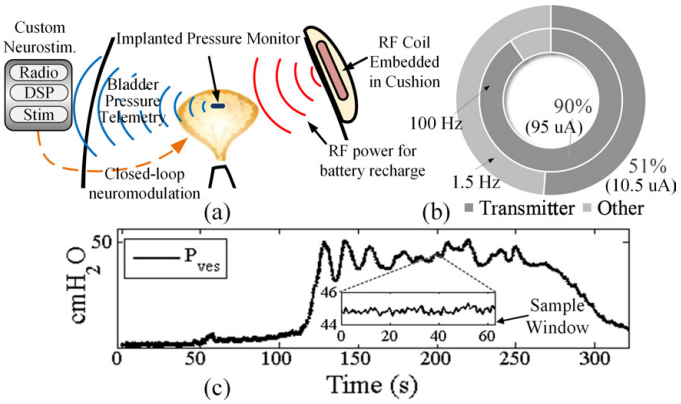

Fig. 1. (a) A wireless bladder pressure monitor, for diagnosis and treatment of lower urinary tract dysfunction. (b) Breakdown of current draw for wireless transmission at 100 Hz ($95\,\mu A$) vs 1.5 Hz ($10.5\,\mu A$); other circuitry independently draws $10\,\mu A$ [1]. (c) Measured vesical pressure, showing a contraction starting at time $t \approx 125$. The insert shows that, on small scales, the signal changes very slowly, and is therefore highly compressible.

II. Background and Motivation

Physiologic bladder pressures change incredibly slowly relative to modern electronic speed (Fig. 1(c)). Even when filled supraphysiologically at a rate of 100 mL/min in urodynamics studies, static bladder pressure typically rises slower than 4 cm H_2O/min. Bladder contractions are fast relative to storage pressure changes, occurring with a maximum pressure rate-of-change of 10 cm H_2O/s and typically lasting 1 to 30 seconds [2]. Motion artifacts superimposed on bladder pressures have higher frequency content, leading to recommended sample rates of 100 Hz to avoid aliasing effects. The primary goal in wireless pressure monitoring is to provide feedback data on the bladder state with sufficient accuracy and low latency such that contractions may be distinguished and acted upon separately from motion artifacts [3].

Data transmission from implantable medical devices consumes significant energy and is often the dominant power draw even in single-channel systems (Fig. 1(b)). Our bladder pressure monitor, for example, consumes 10x more energy when transmitting samples at 100 Hz then at 1.5 Hz [1]. Prior strategies for minimizing data transmission in pressure monitors have implemented very low, sub-Hz sample rates or burst transmissions of pressure history; these strategies have long latency which prevents real-time closed-loop bladder control. We demonstrated that a sparse approach can be effective, wherein pressure data is adaptively transmitted to match the activity level in the pressure signal. By only transmitting "important" samples real-time, low-latency telemetry was maintained, and average transmission power was reduced by 96% with 1.5% average error [1].

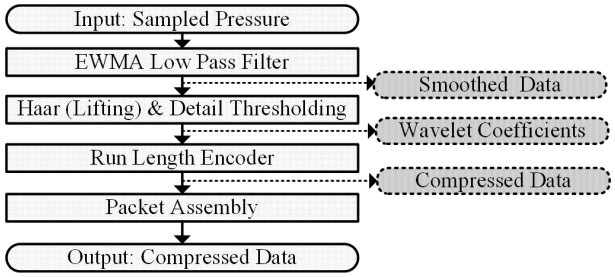

Fig. 2. Algorithm flow for efficient compression of bladder pressure data. Outputs of each stage are shown to the right.

The major drawback of the sparse transmission approach is that signal information is inherently lost when samples are not transmitted. This fundamentally prevents a faithful reconstruction, which may contain important diagnostic symbols for some patients. Pre-transmission compression can mitigate reconstruction errors, but requires silicon area and computation power draw on the implanted device. Modern VLSI technology combined with new circuit topologies, however, can manage this tradeoff by performing data compression in an efficient and lightweight manner suitable for implementation in an ultralow-power implant device.

III. Compression of Bladder Pressure Data

Based on the properties of human bladder pressure data detailed in Section II, we note that the approximation of expected values within a limited sample window can serve as sufficient representations of bladder state for the majority of the recording period. Meanwhile, events like non-voiding or voiding contractions, or spikes in vesical pressure due to the contribution of the abdominal muscles, present as *unexpected* changes in bladder pressure, and therefore must be handled with greater attention to detail.

A. Algorithm Flow

In this context, we present an algorithm which leverages these properties to perform efficient compression of bladder pressure data (Fig 2). Samples are first processed with a lightweight exponentially-weighted moving average [4]. Following this stage, we utilize a lifting scheme version of the Haar wavelet transform, which intrinsically meets the requirements for discriminating between periods of bladder filling and periods of bladder activity. Specifically, we employ a series of lifting stages, separating low frequency signal approximations from high frequency details [5]. Sample data are split into even and odd elements, on which certain operations, namely the *predict* and *update* steps, are performed. These result in computation of successive even and odd elements in the transform space, as defined by Eqns. 1 and 2.

$$odd_{j+1,i} = odd_{j,i} - even_{j,i} \tag{1}$$

$$even_{j+1,i} = even_{j,i} + \frac{odd_{j+1,i}}{2} \tag{2}$$

A total of n-bytes, where n is a power of two, are processed in a single lifting stage, which generates $n/2$ approximations, and $n/2$ details. Subsequent stages similarly reduce the time resolution by half. Furthermore, at each stage, a tunable threshold T_d, applied coarsely to the odd (detail) coefficients of Eqn. 1, determines which components are unexpected, and therefore should be retained in the final reconstruction. Coefficients which are below the threshold are stored as 0. This process tends to produce long runs of 0s (0-runs) in the resulting wavelet domain. Therefore, Run Length Encoding (RLE) is applied in the final processing stage. The encoder output is passed to the packet assembly, which first outputs the header data, containing the run lengths, followed by the corresponding values.

B. Determination of T_d Range

Further analysis of the compression approach uncovers interesting properties for the average output size. Typically, thresholds below 3 do not produce 0-runs of sufficient length, which typically results in compressed representations that require more space than the original. We define the data compression ratio (DCR) as the ratio between compressed and uncompressed data sizes. For example, a 64 byte packet represented in 22 bytes yields a ratio of 22/64, or 34%. In the worst case, if no runs are found, the total size can be expressed as $2n + 1$: 1 byte for header length, the n-byte header array, and the n-byte values array, leading to a DCR of about 200%. Because the signal complexity for a given window is nondeterministic, success of the compression strategy is dependent on finding an appropriate range of values for T_d which provide a good balance between data size reduction and reconstruction error in the average case.

We determine this range empirically using a human cystometry dataset, which includes 64 recordings of vesical pressure from 14 human subjects sampled at 100 Hz and 8 bits per sample, with each recording lasting just over 8 minutes. An example recording is shown in Fig. 1(c). Subjects had confirmed Neurogenic Detrusor Overactivity due to Spinal Cord Injury. Data were acquired at the L. Stokes Cleveland VA Medical Center using clinical urodynamics procedures as approved by their Institutional Review Board.

Taking the window size $n = 64$, this yields roughly 773 non-overlapping sample windows per recording, or approximately 50000 total samples windows in the dataset. Each of these sample windows was compressed with thresholds starting at $T_d = 0$, and compression ended once the percent decrease between subsequent output sizes fell below 0.1%. We observed that a minimum threshold of ± 4 ADC codes, up to a maximum of ± 14 ADC codes (with an 8 bit ADC), provided an appropriate thresholding range for the cystometry dataset. Below 4, resulting data were generally larger than the original, while above 14, the average output size stayed relatively constant. These values correspond to approximately $\pm 1.6\%$ to $\pm 5.5\%$ of the 8-bit ADC range.

Note that the value of T_d is inversely proportional to the resulting compression ratio and directly proportional to reconstruction error. A high value of T_d will result in longer 0-runs at the cost of quality, where as a low value has the inverse effect. Therefore, we define a single compression quality metric Q as

$$Q = \lfloor T_{max} \times 2^R \rfloor - T_d \tag{3}$$

where $T_{max} \approx 0.055$, R is the ADC resolution in bits, and T_d is the threshold value used to obtain a compressed signal with quality Q. This enables a user to specify a desired quality from 0 to 10, where 0 corresponds to low quality, high compression, and 10 corresponds to high quality, low compression. In practice, a Q value of 0 will produce more highly compressed results and reconstructions that tend to omit abdominal-pressure (motion) induced artifacts in the vesical pressure measurement, whereas a Q of 10 will produce high-fidelity reconstructions including motion artifacts. This is demonstrated in Fig. 3(b). The particular choice therefore depends on the clinical application.

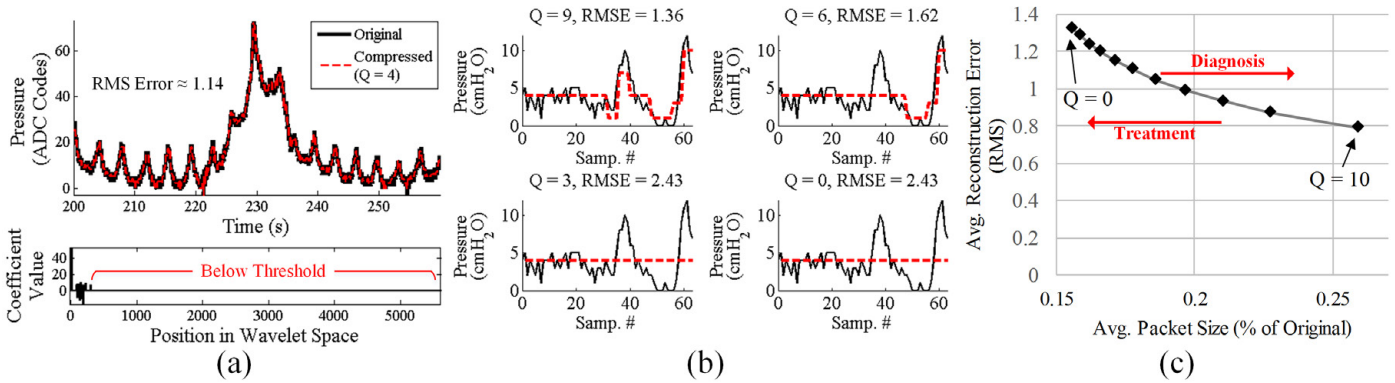

Fig. 3. (a) (Top) Sample algorithm output ($Q = 4$) showing original and reconstructed pressure signals for 60 seconds of data. The compressed signal required an average 8.8 Bytes for each 64 Byte window, with an average RMS error of 1.14; (Bottom) Post-threshold detail coefficients from the lifting scheme. The extended 0-runs enable more efficient compression by Run Length Encoding. (b) Sample algorithm outputs for a single 64 byte (0.64 second) window showing variation in output for various values of Q; note that, in this case, a $Q \leq 3$ disregards transient pressure spikes, and the entire window is represented as the signal average. (c) Quality of algorithm output, comparing dataset-wide average compression ratios with corresponding reconstruction error for all values of Q.

TABLE I. AVERAGE COMPRESSION AND RECONSTRUCTION ERROR FOR SUBJECTS 1-14, FOR $Q = \{0, 5, 10\}$ (STANDARD DEV. SHOWN).

	Compression (% Orig.)			**Error (RMS)**		
#	$Q = 0$	$Q = 5$	$Q = 10$	$Q = 0$	$Q = 5$	$Q = 10$
1	0.23	0.29	0.58	2.53	2.10	1.33
2	0.17	0.20	0.29	1.54	1.20	0.81
3	0.16	0.16	0.20	0.95	0.85	0.69
4	0.15	0.18	0.27	1.31	1.05	0.73
5	0.15	0.18	0.26	1.48	1.20	0.83
6	0.16	0.18	0.29	1.43	1.23	0.86
7	0.14	0.17	0.24	1.21	0.98	0.68
8	0.13	0.14	0.19	1.08	0.82	0.59
9	0.15	0.17	0.23	1.32	1.14	0.84
10	0.16	0.17	0.20	1.09	0.98	0.82
11	0.17	0.18	0.27	1.36	1.22	0.94
12	0.17	0.20	0.28	1.40	1.17	0.84
13	0.14	0.15	0.16	0.68	0.61	0.53
14	0.16	0.18	0.26	1.41	1.22	0.89
	$0.16 \pm$ 0.02	$0.18 \pm$ 0.03	$0.27 \pm$ 0.10	$1.34 \pm$ 0.40	$1.13 \pm$ 0.32	$0.81 \pm$ 0.18

C. Quality of Reconstruction

Fig. 3(a) shows a sample output for a 60 second compression window with $Q = 4$. On average, the compressed data required 8.8 Bytes for each 64 Byte window (14%), with an average RMS error of 1.14. This is slightly better than the expected compressed size of 17%, and comparable to the expected RMS error of 1.15, as shown in Fig. 3(c). Table I shows results for compression and RMS error for all subjects, at selected quality factors of 0, 5, and 10. In the worst case quality ($Q = 0$), a DCR of 0.16 ± 0.02 was achieved, with an RMS error of 1.34 ± 0.40, while in the best case ($Q = 10$), a DCR of 0.27 ± 10 was achieved, with an RMS error of 0.81 ± 0.18. Fig. 3(b) visually demonstrates how the output can vary for different levels of Q; note that, for the lower levels of quality (0 and 3), both rises in pressure are ignored, where as at higher levels of quality (6 and 9), one or both events are represented. DCRs for these windows range from 5% ($q = 0, 3$) to 17% ($Q = 6$) and 23% ($Q = 9$).

IV. IMPLEMENTATION RESULTS

A. Hardware Architecture

The algorithm was implemented in hardware using Verilog HDL, synthesized to gates using Synopsys Design Compiler, and Cadence Encounter was used for layout. We perform synthesis and layout at the $0.18\,\mu m$ and $0.5\,\mu m$ technology nodes to observe the effects of scaling on the design. The design was simulated using Synopsys VCS using as input samples from the human cystometry dataset, described in Section III. The overall hardware architecture is shown in Fig. 4(a).

Initial filtering (EWMAF) is performed with $\alpha = 0.5$, enabling the use of efficient bit-shifting instead of more costly division. The Haar lifting scheme (HAAR_XFM) is implemented sequentially in single module, which enables efficient resource utilization. In addition, following the operations defined in Eqns. 1 and 2, the implementation operates on data in-place, such that only a single $n \times m-$bit set of registers is required for all stages, as shown in Fig. 4(b), A total of 4 stages are used to process the 64 byte packets, leading to $64 >> 4 = 4$ bytes of approximation coefficients, which are generally non-zero, and 60 bytes of detail coefficients. In practice, the majority of the details in Level 1 (32 bytes) and Level 2 (16 bytes) are below T_d and are therefore omitted from the packet. Any detail coefficients above the threshold are considered to be important or significant events in the signal, and are therefore retained in the packet. Once coefficients are computed, the packet is transferred to the run length encoder, which encodes data and assembles it into the final packet format (RLE_PKT) for transmission.

B. Area, Power, and Performance

Layouts with TSMC $0.18\,\mu m$ and AMI $0.5\,\mu m$ technology [6] result in total areas of $1.34mm^2$ and $11.00mm^2$, as shown in Figs 4(c) and (d), respectively. Roughly 40% of the overall area is due to the memory elements, implemented as flip-flops. Using different memory technologies such as SRAM, or even emerging nanoscale memory technologies, can result in significant area reduction.

One key observation in the design is the relatively low clock frequency required for functional operation. Specifically, the ADC performs 100 Hz sampling, yielding one 64 Byte packet every 0.64 seconds. By comparison, the algorithm requires between 318 and

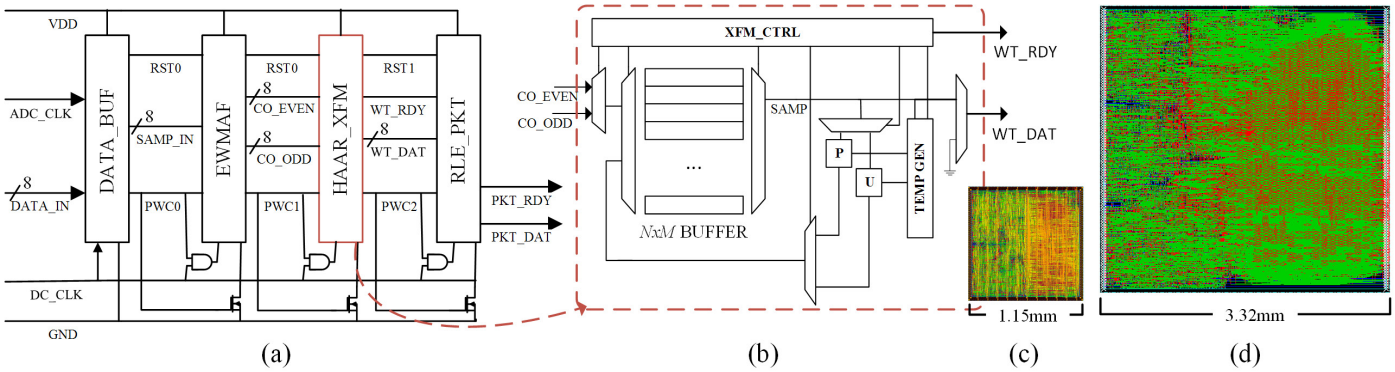

Fig. 4. (a) Hardware implementation of the data compression algorithm, optimized for ultra-low power consumption. Neighboring modules enable power and clock signals to subsequent processing stages. (b) Expanded view of lifting scheme implementation, featuring memory resource reutilization for area and power efficient processing. (c) Layout in $0.18\,\mu m$ and (d) $0.5\,\mu m$ technology.

TABLE II. SYNTHESIS RESULTS WITH AND WITHOUT LEAKAGE CONTROL FOR $0.18\,\mu m$ AND $0.5\,\mu m$ TECHNOLOGIES.

	w/ Leakage Ctrl		**w/o Leakage Ctrl**	
	$0.18\,\mu m$	$0.5\,\mu m$	$0.18\,\mu m$	$0.5\,\mu m$
DC_CLK (kHz)	1000	1000	1	1
Avg Pow (dyn) (nW)	0.81	11.9	98.7	1462
Avg Pow (sta) (nW)	1.53	2.27	976.6	1447
Avg Pow (tot) (nw)	2.34	14.1	1075	2910
Energy / Packet (nJ)	95.8	578.5	688.2	935.9

444 cycles – with a considerably shorter critical path – to produce a compressed packet; the variation in cycles arises from the packet output stage, since its duration depends on the resulting packet size. On average, the number of cycles will vary from 325 to 332 for different values of Q ranging from 0 to 10, respectively. This presents many opportunities for low power design, such as the use of high-V_T gates and coarse-grained supply gating of individual pipeline stages for static power reduction, or reducing V_{DD} for both static and dynamic power reduction.

Table II shows synthesis results at $0.18\,\mu m$ and $0.5\,\mu m$ technology nodes. Note that the static power for $0.5\,\mu m$ is about 1.5x higher than that at $0.18\,\mu m$, as shown in the standard cell library reference [6]. Synthesis results at $0.18\,\mu m$ also demonstrate that static power is about 10x greater than the dynamic power when operating at the minimum required frequency, roughly 7x the speed of the sample clock – in this case, 694 Hz – as dictated by the worst case number of processing cycles. Scaling below $0.18\,\mu m$ would exacerbate this disparity; it is crucial, therefore, to reduce the effect of leakage on the circuit. Supply gating has been effectively used in similar situations [7], and has been shown to effectively reduce leakage power by above 90% in some cases [8]. Therefore, we can expect to achieve significant power savings by operating at a higher frequency, leading to longer idle times. By varying the DC_CLK frequency between 1 kHz and 100 MHz, and assuming the worst case average runtime (332 cycles), we observed that the total (static + dynamic) average power consumption can be reduced from $1.1\,\mu W$ to as low as $2.3\,nW$ $(0.18\,\mu m)$, and from $2.9\,\mu W$ to $14.1\,nW$ $(0.5\,\mu m)$. This is achieved in both cases when operating DC_CLK at 1 MHz, leading to an effective duty cycle of $5.4 \times 10^{-4}\%$ compared to the $0.64\,s$ required to fill the 64 Byte sample buffer. This also reduces the post-packet acquisition latency, enabling more responsive conditional neurostimulation in treatment applications [3].

Supply gating transistors also requires additional power control signals. Due to the deterministic nature of the first three pipeline stages, each module can generate the supply enable signal to its neighbor ("PWC" in Fig. 4(a)). We adopt this approach to simplify the IC power network design and reduce area overhead.

V. CONCLUSION

We have presented a novel, tunable algorithm for wavelet transform based efficient compression of bladder pressure data. We have tested the algorithm using a large human cystometry dataset, and results show excellent compression with low average RMS error. We have implemented the algorithm in hardware, and results from synthesis and layout in $0.18\,\mu m$ technology demonstrate ultra-low average power of $2.3\,nW$ with an area of $1.32\,mm^2$. To the best of our knowledge, this is the first example of a compression algorithm which is tailor-made to human bladder pressure data, that is also amenable to ultralow-power hardware implementation. In this way, it can help support diagnosis of lower urinary tract dysfunction, and provides an important tool for biomedical research in this area.

REFERENCES

[1] S. Majerus *et al.*, "Wireless implantable pressure monitor for conditional bladder neuromodulation," in *BioCAS*. IEEE, 2015, pp. 1–4.

[2] A. Gammie *et al.*, "International continence society guidelines on urodynamic equipment performance," *Neurourology and Urodynamics*, vol. 33, no. 4, pp. 370–379, 2014.

[3] R. Karam *et al.*, "Real-time classification of bladder events for effective diagnosis and treatment of urinary incontinence," *IEEE TBME*, vol. 63, no. 4, pp. 721–729, April 2016.

[4] C. C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages," *International Journal of Forecasting*, vol. 20, no. 1, pp. 5 – 10, 2004.

[5] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM Journal on Mathematical Analysis*, vol. 29, no. 2, pp. 511–546, 1998.

[6] Oklahoma State University, "FreePDK: Unleashing VLSI to the Masses," Online, [Accessed June 22, 2016]. [Online]. Available: http://vlsiarch.ecen.okstate.edu/flows/MOSIS_SCMOS/osu_soc_v2.7/cadence/flow/

[7] S. Narasimhan, H. J. Chiel, and S. Bhunia, "Ultra-low-power and robust digital-signal-processing hardware for implantable neural interface microsystems," *IEEE TBioCAS*, vol. 5, no. 2, pp. 169–178, 2011.

[8] J. W. Tschanz *et al.*, "Dynamic sleep transistor and body bias for active leakage power control of microprocessors," *Solid-State Circuits, IEEE Journal of*, vol. 38, no. 11, pp. 1838–1845, 2003.