Design and Implementation of an Ultralow-Energy FFT ASIC for Processing ECG in Cardiac Pacemakers

Safwat Mostafa Noor, Eugene John, and Manoj Panday

Abstract—**In embedded biomedical applications, spectrum analysis algorithms such as fast Fourier transform (FFT) are crucial for pattern detection and have been the focus of continued research. In deeply embedded systems such as cardiac pacemakers, FFT-based signal processing is typically computed by application-specific integrated circuit (ASIC) to achieve low-power operation. This brief proposes a data-driven design approach for an FFT ASIC solution, which exploits the limited range of data encountered by these embedded systems. The optimizations proposed in this brief use the simple concept of hashing and lookup table to effectively reduce the number of arithmetic operations required to perform the FFT of an electrocardiogram (ECG) signal. By reducing the dynamic power consumption and overall energy footprint of FFT computation, the proposed design aims to achieve longer battery life for a cardiac pacemaker. The design is synthesized using a 90-nm standard cell library, and gate level switching activity is simulated to obtain accurate power consumption results. The proposed optimizations achieved a low energy consumption of 27.72 nJ per FFT, which is 14.22% lower than a standard 128-point radix-2 FFT when tested with actual ECG data collected from PhysioNet.**

Index Terms—**Block floating point (BFP), cardiac pacemaker, electrocardiogram (ECG), fast Fourier transform (FFT), hash, lookup table (LUT), low-power application specific integrated circuit (ASIC).**

I. Introduction

Fourier transform is undoubtedly one of the most influential signal processing algorithms, particularly so for biomedical applications [1], [2]. The advancements of digital signal sampling and the emergence of discrete Fourier transform (DFT) enable biomedical researchers to analyze the frequency components of physiological signals and detect concealed properties to identify physical anomalies. Such methods have been particularly effective in analyzing physiological signals such as the electrocardiogram (ECG) [3]–[5]. A practical application of the above-mentioned signal analysis can be found inside an implantable cardiac pacemaker, which is a device tasked with continuous monitoring of ECG signals to detect heart rhythm abnormalities [6]. In the event of a slow intrinsic rhythm or lack of normal impulse conduction, the pacemaker generates electrical stimulations to pace the heart muscle and maintain an adequate heart rate. A pacemaker has to operate in a highly power constrained environment and may last a very long period (~7–10 years) on a single battery [7]. Therefore, the embedded system inside a cardiac pacemaker needs to process the ECG signal with maximum energy efficiency to ensure a sufficiently long battery life. A typical system architecture for a cardiac pacemaker is shown in Fig. 1.

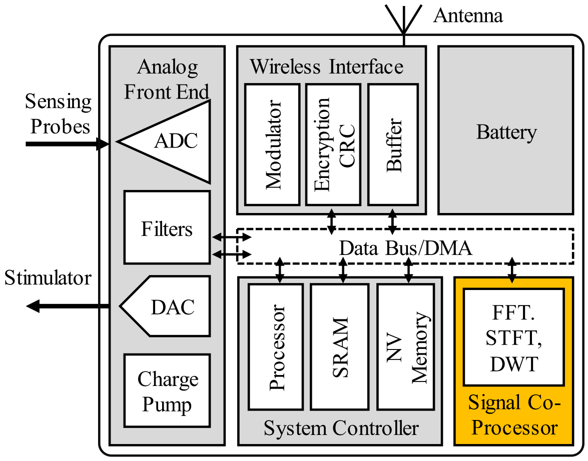


Fig. 1. Diagram of the embedded system inside a cardiac pacemaker. The signal co-processor highlighted in orange is responsible for continuous processing of sampled ECG data and is the target of this brief.

A central processor supervises the overall operation, whereas a dedicated co-processor executes the signal processing algorithm. The signal co-processor (highlighted in orange in Fig. 1) is responsible for computing fast Fourier transform (FFT) or similar spectrum analysis algorithms [8], [9]. This motivates the effort behind optimizing the design of an FFT co-processor. This brief presents a power-efficient FFT computation method and its VLSI implementation that requires minimal additional hardware and chip area but achieves significantly reduced circuit switching activity and dynamic power consumption.

II. Theory and Motivation

The FFT is an efficient algorithm for calculating the DFT and is preferred in practical applications since it has a complexity of only $O(N\log_2 N)$ [10]. A pair of complex-valued signal samples is computed using a butterfly structure in the FFT algorithm. In the underlying hardware, each butterfly performs four multiplications and six additions in the chosen number representation format. The real and imaginary coefficients of the complex variables can be represented in floating point, fixed point, or block floating point (BFP) representation. The BFP numbering concept is preferred, where computation accuracy and high dynamic range are required without the added complexity of full floating point implementations [11]. BFP representation is attractive for digital signal processor (DSP) applications as it can provide the precision and dynamic range required by algorithms such as FFT using fixed point arithmetic [12].

In contrast to IEEE-754 floating point, the BFP scheme assigns a single exponent to a set of numbers. In this arrangement, fixed point hardware can be utilized to perform arithmetic operations such as multiplication and addition on the mantissa block. After every operation, the common exponent associated with the block needs to be scaled to fit the largest number in the set of mantissas.

Manuscript received June 30, 2018; revised October 10, 2018; accepted November 12, 2018. Date of publication December 14, 2018; date of current version March 20, 2019. The work of E. John was supported by the National Institute of General Medical Sciences of the National Institutes of Health under Award 1SC3GM096937-01A1. *(Corresponding author: Safwat Mostafa Noor.)*

S. M. Noor is with the Apple Lonestar Design Center, Austin, TX 78746 USA (e-mail: safwatmostafa.noor@utsa.edu).

E. John is with the Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: eugene.john@utsa.edu).

M. Panday is with the Division of Cardiology, Department of Medicine, University of Texas Health Science Center at San Antonio, San Antonio, TX 78229 USA (e-mail: panday@uthscsa.edu).

Color versions of one or more of the figures in this brief paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2018.2883642

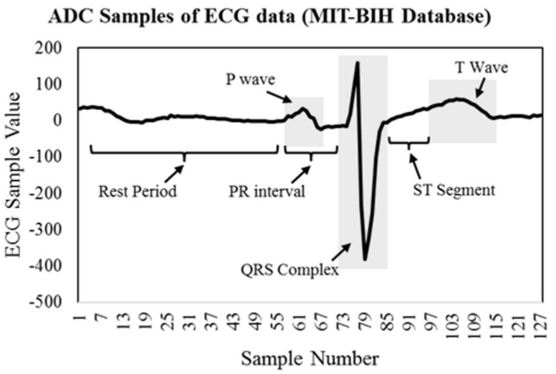


Fig. 2. Plot of 11-bit digital ECG samples from MIT-BIH long-term ECG database [16]. The markers indicate various segments of an ECG signal. Excluding the QRS complex, the range of signal values is limited within 0–60.

A. ECG Data Patterns

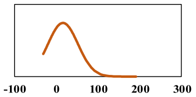
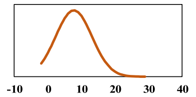
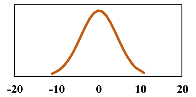
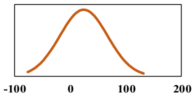
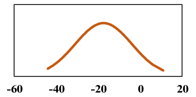
Detecting and classifying abnormalities in the ECG are key to identifying various types of abnormal heartbeats known as Arrhythmia. The process of classification requires certain spectral features to be extracted from the ECG [17], [18]. The process of FFT-based feature extraction and classification continues throughout the operating life cycle of the pacemaker, consuming a significant portion of the available energy. Therefore, power optimization in the FFT algorithm is highly desired.

An ECG is a digital record of electric impulses of the heart captured using sensing electrodes. Fig. 2 shows an 11-bit digital ECG record plotted against time. The data are nearly periodic with every heartbeat and exhibit the well-known P-wave, PR interval, QRS complex, ST segment, T-wave, and the rest period [15]. Statistical analysis shows that samples outside the QRS complex have a narrow range of values (0–60) and a low standard deviation (18.3). Therefore, two randomly selected samples outside the QRS complex will hold a high probability of having the same value. This holds true for all cycles of the ECG signal under normal conditions.

For this brief, alongside a normal ECG sample, four types of abnormal ECG records were collected from the MIT-BIH Arrhythmia database [20], [26], covering the most common cases of Arrhythmia. The records include Sinus Bradycardia (SBR), A-V block (AVB), Atrial Fibrillation (AFIB), and Atrial Flutter (AFL). The statistical footprint of these 11-bit nonideal ECG records is summarized in Table I. SBR and AVB exhibit low standard deviation since the heart fails to generate its natural pacing impulses in these conditions. Values from these conditions indicate that in most common cases of pacemaker usage, the sample values encountered by the system will hold a high probability of being repeated over the course of time. For the remaining two cases of AFIB and AFL, the rate of change is somewhat higher due to the nature of the conditions. Calculated values for standard deviation were 42.9 and 13.5 for AFIB and AFL, respectively. Although not commonly encountered, analyzing the performance of the proposed FFT during the processing of AFIB and AFL adds to the robustness and scalability of the approach presented in this brief.

Due to the observed repeated values, when an input sequence from any of these ECG records is processed through an FFT algorithm, a significant number of redundant arithmetic operations take place. To illustrate the redundancy, samples from all ECG records were used as an input to a 128-point BFP FFT simulator utilizing Q15 data format. In each stage of the FFT, the operands to multiplication step within the butterfly computation were logged. The occurrence of unique operand combinations was counted, and only the top five repeated combinations were taken into consideration. Fig. 3 shows the percentage of repeated multiplications with respect to the total multiplications in each stage. The top five repeated inputs in the first four stages contribute to 38.3% of the total multiplication on average. By identifying the top five occurring multiplications of each stage and storing their result for future lookup will avoid a significant number of multiplication operation and circuit switching. Using relatively small lookup table (LUT) of five entries, this optimization concept can be implemented to reduce dynamic power consumption for a 128-point FFT. The concept of result lookup in FFT has been possible in software implementation [18] based on IEEE-754 single precision floating point arithmetic. However, utilizing this technique for BFP requires novel hashing technique and broadens the effectiveness to larger sample set FFT algorithms.

TABLE I
Statistics of the Normal and Abnormal ECG Records

Parameters	Mean	Standard Deviation	Absolute Range	Probability Density Function
Normal	16.26	35.88	161	
AV Block	7.73	5.41	27	
Sinus Bradycardia	0.21	4.33	22	
Atrial Fibrilation	24.63	42.91	56	
Atrial Flutter	-17.43	13.50	33	

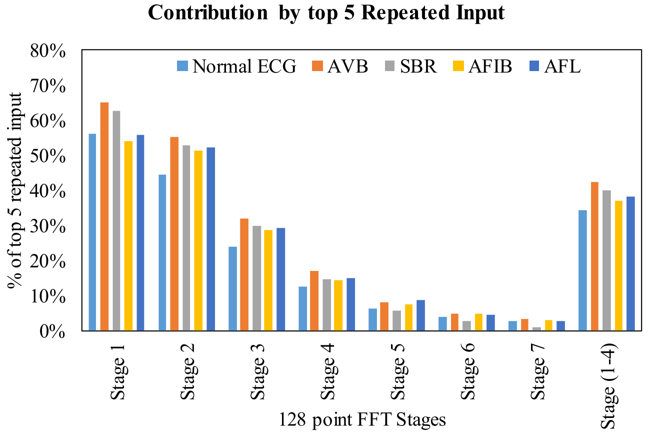


Fig. 3. Bar plot showing the percentage of repeated multiplications of the seven stages in the 128-point FFT coming from the top five repeated input.

III. Proposed Design

The design of the FFT application specific integrated circuit (ASIC) is structured around a butterfly computation unit. A 16-bit signed adder and a 16-bit signed multiplier for BFP calculation are the only arithmetic blocks in the butterfly computation unit, controlled by a finite state machine. The diagram of the butterfly module is shown in Fig. 4. As stated earlier, the multiplier in the butterfly unit has access to a five-entry lookup memory, which is designed to provide the multiplication results for the most frequent operands. Performing a

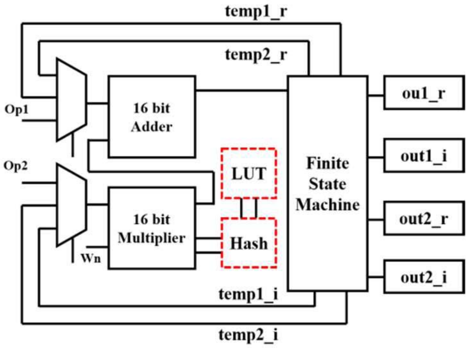


Fig. 4. Block diagram of the butterfly unit. A standard BFP arithmetic butterfly unit.

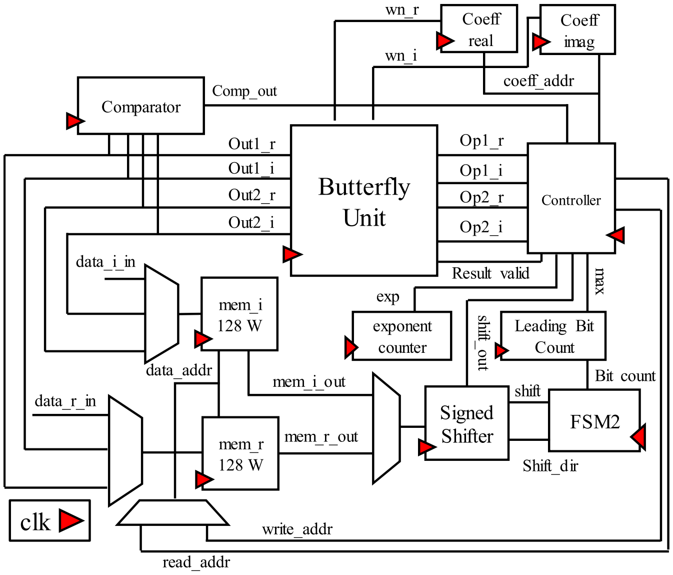


Fig. 5. Simplified block diagram of the entire FFT co-processor. Based around a single butterfly unit, this circuit iterates through the butterfly operation of every stage sequentially.

lookup on the table requires an efficient hash algorithm for generating a lookup index. In this implementation, a folded XOR hash function was utilized to generate an index to the LUT. The XOR hash function is popular for its simplicity and excellent nonuniform hash value distribution to minimize potential conflicts [19]. Each input to the multiplier was passed to the hash function to generate an index to the LUT. The XOR hash function computed a 4-bit hash value by XOR-ing the four nibbles (4 bits) that exist in a 16-bit number. The area required to synthesize the lookup enabled multiplier and hash function was 5523 μm^2, merely 9.7% larger than the standard booth multiplier utilized in a standard FFT circuit.

A. System Architecture and Performance

The architecture of the FFT ASIC is designed around the optimized butterfly module. The main supporting components required for BFP exponent scaling at every stage include a signed shifter, an exponent counter, and a leading bit counter. The butterfly computation module is designed to output two complex results $X_i[n]$ and $X_i[n+1]$ and the max 16-bit variable within $X_i[n]$ and $X_i[n+1]$. The max output of every butterfly is stored and compared with the previous values to find the maximum magnitude generated in every FFT stage. Before the starting of the next stage, this maximum magnitude is used by the leading bit counter to find the free space that can be used to shift and scale the input variables to the next stage. Scaling improves the precision and allows a higher dynamic range in the overall process. The scaling value is used to set the shifter to shift every data read from the memory banks. With every shift, the value of the exponent counter is updated according to the scaling value at the end of each stage. The implementation adheres to the "in-place" nature of the FFT algorithm, meaning that the result of each stage will be written back to the

TABLE II
HIERARCHICAL BREAKDOWN OF POWER

Module	Description	Dynamic Power (µW)	Leakage Power (µW)	Total Power (mW)
fft128	Top Module	2448.44	992.63	3.441
bfly	Butterfly Unit	353.51	77.50	0.431
fsm	FSM Butterfly	30.51	1.42	0.032
add16	16-bit adder	63.78	3.40	0.067
lut_mult16	16-bit LUT mult	145.85	34.50	0.180
misc_logic	muxes etc.	89.59	6.03	0.096
misc_reg	I/O, temp reg	6.98	15.82	0.023
scale_logic	BFP scaling logic	0.53	5.47	0.006
coeff_i_mem	Coefficient storage	0.00	8.50	0.009
coeff_r_mem	Coefficient storage	6.07	11.00	0.017
memory_i	Sample storage	623.88	381.00	1.105
memory_r	Sample storage	832.33	381.00	1.213
fsm	Top FSM	89.66	23.50	0.113
misc_logic	muxes etc.	115.45	2.85	0.168
misc_reg	I/O, temp reg etc	90.29	40.64	0.131

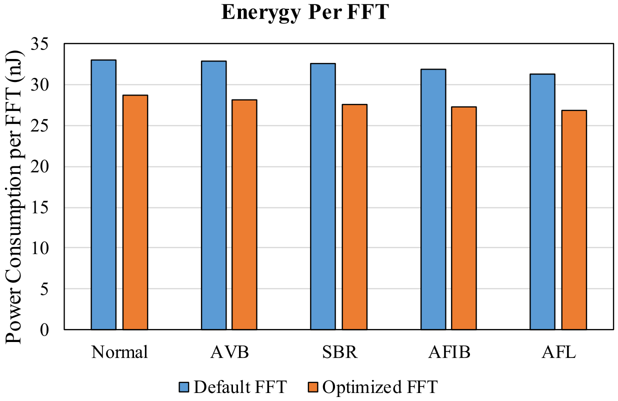


Fig. 6. Energy per FFT. The optimized FFT circuit provides a more energy-efficient solution for processing various ECG signals.

TABLE III
COMPARISON WITH THE STATE-OF-THE-ART FFT IMPLEMENTATIONS

Parameters	This Work	Ref. [21]	Ref. [22]	Ref. [23]
Technology (T)	90 nm	130 nm	45 nm	110 nm
Area	0.19 mm^2	2.29 mm^2	Not Given	0.42 mm^2
Voltage (V)	1.2 V	1.2 V	1.1 V	1.2 V
Clock Rate	100 MHz	100 MHz	100 MHz	50 MHz
Data Type	BFP	Fixed	Fixed	Fixed
Data Length (L)	16 Bits	11 Bits	32 Bits	12 Bits
FFT Points (N)	128	128	1024	2048
Active Power	3.44 mW	67.9 mW	102.2 mW	3.64 mW
Time/FFT	8.06 µS	4.48 µS	51.2 µS	139.5 µS
Normalized Energy [1]	27.7 nJ	330.2 nJ	474.4 nJ	23.3 nJ

[1] $Normalized\ Energy = \frac{Power \times Execution\ Time}{(\frac{L}{16})^{1.2} \times (\frac{T}{90}) \times (\frac{N log_2 N}{128 log_2 128}) \times (\frac{V}{1.2})^2}$

TABLE IV

ESTIMATED BATTERY LIFE OF A PACEMAKER USING FFT-BASED DETECTION SCHEME

Component	**Percentage of Total Energy**	**Energy (nJ) (Default FFT)**	**Energy (nJ) (Optimized FFT)**	**Ref. [21]**	**Ref. [22]**	**Ref. [23]**
Detection	67%	27.72	22.26	330.15	474.4	23.34
Pacing	33%	13.65	13.65	13.65	13.65	13.65
Total	100%	41.37	37.43	343.80	488.05	36.99
Pacemaker Battery Energy (J) [7]		12096				
Possible Number of Detect + Pacing		2.92E+11	3.23E+11	3.52E+10	2.48E+10	3.27E+11
Interval Between Detect + Pacing (mS)		1.00				
Total Runtime (S)		2.92E+08	3.23E+08	3.52E+07	2.48E+07	3.27E+08

memory to overwrite the original data. The twiddle factors needed for the calculations are precomputed and stored in two read-only memory modules. The overall system architecture is given in Fig. 5.

Each butterfly operation required 12 clock cycles to complete. Besides the butterfly operation, the reading of input data, scaling, and writing back to memory required another six clock cycles, resulting in 18 clock cycles altogether. For a 128-point FFT, a total of $64 \times 7 = 448$ butterfly operations take place, resulting in $448 \times 18 = 8064$ cycles per FFT. Choosing a maximum clock frequency of 100 MHz allows the computation to be completed within 8.064 μs, which is negligible within the sample accumulation window.

IV. RESULTS AND ANALYSIS

Two separate designs, with and without the proposed optimization, were implemented for comparison. Synthesis and layouts of the designs were realized using the Synopsys 90-nm Generic Library [20]. Low-power techniques such as clock gating and use of mixed-vt libraries were enabled. Postroute simulation and design rule check verification were performed to ensure correct functionality before simulating the postroute netlist for estimating power consumption. Switching activity during simulation was captured in switching activity interchange format, which was used along with parasitic information used in PrimeTime PX (PTPX) to determine activity-based power consumption at an operating voltage of 1.2 V and a clock speed of 100.0 MHz. The hierarchical breakdown of the power is shown in Table II. Implementation of the optimized design was possible within a chip area of 198404 μm^2 (0.19 mm^2).

To represent a practical workload, real-world ECG data were used to simulate its operation. Five different sets of ideal and nonideal ECG data sets were used as inputs. Average power analysis report was reported by PTPX. The reported dynamic power consumption was 14.22% lower than the unoptimized FFT design. The time required to finish each computation was 8.06 μs, yielding an average energy consumption of 27.72 nJ for a single FFT. The energy consumption per FFT for different input sets is shown in Fig. 6.

A. Comparison With Existing Work

The FFT power reduction approach presented in this brief is compared with present day state-of-the-art FFT solutions from different disciplines and is shown in Table III. The energy for each design is normalized to remove the effect of different design parameters. Linear scaling using data bit length, smallest feature size, a total number of butterfly operations, and an exponential scaling of the voltage was used to scale the energy per FFT of each design. The normalized results clearly highlight the beneficial effects of the optimizations against other methods of FFT computation. Although an eDRAM-based FFT is able to achieve slightly better results, it does so with higher area utilization and risk of memory retention failure [23]. Both of these properties are not desirable in a life-critical and compact pacemaker application.

B. Application in Cardiac Devices

With a smaller power profile, the new FFT ASIC can benefit implantable battery-powered biomedical devices to sustain a longer runtime. In a pacemaker, Fourier transform is one of the primary computations executed continuously in real time. Therefore, in such applications, the reduction in power is expected to extend the battery life significantly. The embedded system in a cardiac pacemaker typically hosts other circuit components such as an ADC, one or more analog filters, a charge pump for higher voltage generation, a pulse generator for generating heart stimulations, and a DSP for computing FFT and other signal processing tasks [6]. Based on the datasheet of a modern pacemaker [7], a pacemaker typically consumes 67% power for ECG analysis and 33% power for pacing during continuous single chamber pacing. Based on these power distributions and the FFT energy obtained in this brief, the overall energy breakdown and battery life of a pacemaker are calculated and presented in Table IV. The battery runtime is computed based on a 3.36-Wh battery capacity from [7] and a typical 1-kHz rate of detection and pacing operation [24]. The energy for each pacing operation is used to calculate the absolute possible number of operations till the battery reaches its end of service state. Assuming continuous detection and pacing occurring at every 1 ms (1-kHz rate), the battery life for the default FFT was calculated to be 9.9 years. Commercially available pacemakers have battery life in the range of 7–10 years, and it varies with the severity of the heart problem and the corresponding pacing requirement [7], [25]. Using the FFT ASIC presented in this brief, the power consumption of the detection algorithm can be improved by 14.22%, extending overall runtime. Since the detection logic and FFT operations are active throughout the lifetime of the pacemaker, the overall energy saving adds up to deliver significantly improved runtime.

V. CONCLUSION

Efficient processing of biological data such as ECG samples is a challenging task in resource-constrained environments. Ensuring algorithmic efficiency and reducing power consumption require application-specific optimizations of the hardware, along with low-power synthesis and implementation techniques. When domain knowledge of ECG data is utilized during architectural development of the system, greater insight and optimal design become possible. In this brief, a highly efficient FFT architecture was proposed and developed by statistically analyzing the nature of FFT for computing ECG signals. Based on the findings, modifications to the butterfly unit were proposed to develop an efficient FFT ASIC that effectively minimizes its energy footprint. Reusing intermediate arithmetic results by using LUTs and BFP operations reduced the overall switching activity and dynamic power consumption by 14.22% across various usage cases. The low computation energy of 27.72 nJ per 128 FFT directly will lead to longer battery life in the miniature battery-powered cardiac pacemakers [26] of the future. Although this novel approach for FFT computation was presented for ECG data, the concept of profiling a signal and populating appropriate LUTs can be applied to any data set that exhibits the probability of repeated samples. Such properties are observed in popular wearable applications such as heart rate monitors and fitness trackers. Other closely related signal processing algorithms such as short-time Fourier transform or discrete wavelet transform can also benefit from such dynamic computing-based optimization when processing signals with limited data range and recurring values.

Acknowledgment

The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

[1] A. P. Yoganathan, R. Gupta, and W. H. Corcoran, "Fast Fourier transform in the analysis of biomedical data," *Med. Biol. Eng.*, vol. 14, no. 2, pp. 239–245, Mar. 1976.

[2] M. Akay, *Time Frequency and Wavelets in Biomedical Signal Processing* (IEEE Press Series in Biomedical Engineering). 1998.

[3] P. de Carvalho, J. Henriques, R. Couceiro, M. Harris, M. Antunes, and J. Habetha, "Model-based atrial fibrillation detection," in *ECG Signal Processing, Classification and Interpretation: A Comprehensive Framework of Computational Intelligence*, A. Gacek and W. Pedrycz, Eds., London, U.K.: Springer, 2012, pp. 99–133.

[4] F. M. Vaneghi, M. Oladazimi, F. Shiman, A. Kordi, M. J. Safari, and F. Ibrahim, "A comparative approach to ECG feature extraction methods," in *Proc. 3rd Int. Conf. Intell. Syst. Modelling Simul.*, 2012, pp. 252–256.

[5] N. A. Abdul-Kadir, N. M. Safri, and M. A. Othman, "Effect of ECG episodes on parameters extraction for paroxysmal atrial fibrillation classification," in *Proc. IEEE Conf. Biomed. Eng. Sci.*, 2014, pp. 874–877.

[6] S. A. P. Haddad and W. A. Serdijn, "Ultra low-power biomedical system designs," in *Proc. Ultra Low-Power Biomed. Signal Process.*, Dordrecht, The Netherlands: Springer, 2009, pp. 131–172.

[7] *Biotrnik Effecta Pacemaker Technical Manual*, Biotronik, Berlin, Germany, 2014.

[8] J. A. Crowe, N. M. Gibson, M. S. Woolfson, and M. G. Somekh, "Wavelet transform as a potential tool for ECG analysis and compression," *J. Biomed. Eng.*, vol. 14, no. 3, pp. 268–272, May 1992.

[9] B. S. Shaik, G. V. S. S. K. R. Naganjaneyulu, T. Chandrasheker, and A. V. Narasimhadhan, "A method for QRS delineation based on STFT using adaptive threshold," *Procedia Comput. Sci.*, vol. 54, pp. 646–653, 2015.

[10] G. D. Bergland, "A guided tour of the fast Fourier transform," *IEEE Spectr.*, vol. 6, no. 7, pp. 41–52, Jul. 1969.

[11] G. Govindu, L. Zhuo, S. Choi, and V. Prasanna, "Analysis of high-performance floating-point arithmetic on FPGAs," in *Proc. 18th Int. Parallel Distrib. Process. Symp.*, 2004, p. 149.

[12] D. Elam and C. Lovescu, "A block floating point implementation for an N-point FFT on the TMS320C55X DSP," Texas Instrum. Appl., Dallas, TX, USA, Tech. Rep. SPRA948, 2003.

[13] K. Minami, H. Nakajima, and T. Toyoshima, "Real-time discrimination of ventricular tachyarrhythmia with Fourier-transform neural network," *IEEE Trans. Biomed. Eng.*, vol. 46, no. 2, pp. 179–185, Feb. 1999.

[14] W. Jiang and S. G. Kong, "Block-based neural networks for personalized ECG signal classification," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1750–1761, Nov. 2007.

[15] J. D. Bronzino, *Biomedical Engineering Handbook*, vol. 2. Boca Raton, FL, USA: CRC Press, 1999.

[16] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[17] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, May/Jun. 2001.

[18] S. Mostafa and E. John, "Reducing power and cycle requirement for FFT of ECG signals through low level arithmetic optimizations for cardiac implantable devices," *J. Low Power Electron.*, vol. 12, no. 1, pp. 21–29, 2016.

[19] *On Designing Fast Nonuniformly Distributed IP Address Lookup Hashing Algorithms*. Accessed: Oct. 10, 2018. [Online]. Available: https://dl.acm.org/citation.cfm?id=1721728

[20] R. Goldman *et al.*, "Synopsys' open educational design kit: Capabilities, deployment and future," in *Proc. IEEE Int. Conf. Microelectron. Syst. Educ.*, Jul. 2009, pp. 20–24.

[21] Y. Bo, R. Dou, J. Han, and X. Zeng, "A hardware-efficient variable-length FFT processor for low-power applications," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2013, pp. 1–4.

[22] Z. Qian and M. Margala, "Low-power split-radix FFT processors using radix-2 butterfly units," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 9, pp. 3008–3012, Sep. 2016.

[23] G. Kang, W. Choi, and J. Park, "Embedded DRAM-based memory customization for low-cost FFT processor design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 12, pp. 3484–3494, Dec. 2017.

[24] L. Hejjel and E. Roth, "What is the adequate sampling interval of the ECG signal for heart rate variability analysis in the time domain?" *Physiol. Meas.*, vol. 25, no. 6, p. 1405, 2004.

[25] (2017). *Medtronic Adapta Specifications*. Accessed: Mar. 13, 2016. [Online]. Available: http://www.medtronic.com/us-en/healthcare-professionals/products/cardiac-rhythm/pacemakers/adapta.html

[26] *Micra Transcatheter Pacing System | Medtronic*. Accessed: Dec. 3, 2016. [Online]. Available: http://www.medtronic.com/us-en/healthcare-professionals/products/cardiac-rhythm/pacemakers/micra-pacing-system.html