2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

ApproxBioWear: Approximating Additions for Efficient Biomedical Wearable Computing at the Edge

Alish Kanani[1], Rajat Bhattacharjya[2], and Dip Sankar Banerjee[3]

***Abstract*— Wearables in the biomedical domain have been of extensive use in the current era. Given the importance of wearable computing, it has become necessary to innovate on enhancing hardware efficiency. The domain of approximate computing offers a conclusive method to lower area, power and delay in hardware in addition to a marginal loss in accuracy. In this paper, we investigate ApproxBioWear, a technique which enables the use of approximate computing for efficient biomedical wearable computing at the edge. The methodology involves approximating additions during the functional stages of an error-resilient biomedical signal processing algorithm and determining the application accuracy. Upon evaluating the Pan-Tompkins algorithm, which is used to detect QRS peaks in ECG signals, it is observed that the ApproxBioWear approach reduces the power consumption and chip area by 19.27% and 19.71% respectively on an average with a marginal loss in accuracy.**

I. Introduction

Given the rise of COVID-19, it has become ever important to ramp up efficiency in wearables, as they help fetch preliminary results necessary for further diagnosis. Edge devices such as FitBit [1], Apple Watch [2] help gather information about various physiological signals such as heart rate, electrocardiogram (ECG), respiration and pulse rate. Hence, power efficiency is of utmost importance in any wearable system as they are battery operated and need to have a high lifespan.

Algorithms involving biomedical signal processing are error-resilient in nature [3] as these algorithms involve operations such as the Fast Fourier Transform, Wavelet Transform, Hadamard Transform etc. which due to their compute pattern show error resilience. The error resilience possessed by biomedical signal processing algorithms can be attributed to factors like noise and redundancy in real-world data and the application's capability of mitigating accuracy loss as a result of efficient compute patterns. In addition to that, there lie noise sources such as electronic noise or noise of measurements, structured noise etc. during any sort of biomedical signal acquisition, but we still can study the signal's effects which implies the fact of error resilience possessed by biomedical signal processing algorithms. Also, given the need for high battery life in wearables alongside lesser delay in computation, the algorithms involved in biomedical signal processing come up to be a popular choice

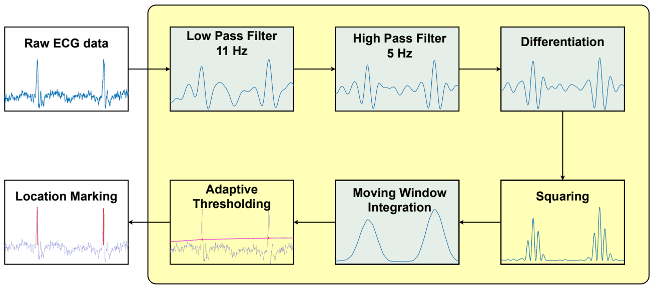


Fig. 1: Pan-Tompkins Algorithm and Error Resilient Blocks

for approximations. The domain of approximate computing offers the ability to lower area, delay and power parameters for hardware efficiency in exchange for a marginal loss in accuracy [4]. Hence, for error-resilient algorithms, approximate computing has come up extensively over the last decade. Approximate adders [5], [6] and multipliers [4] are the basic circuit elements in any approximate system.

Computing in any form involves the use of operations such as additions and multiplications. Accurate digital arithmetic circuits consume a good amount of on-chip area and power. In addition to that, there is a significant latency involved for output generation. In order to address that issue, additions and multiplications have been done approximately [4]–[6] keeping in mind the nature of error-resilience possessed by applications that require such computations extensively. In the domain of biomedical signal processing, the use of digital filters is extensive in nature. We target approximations in the computations involved in digital filtering and obtain substantial results which validate our approach. In an N-point FIR Filter, if we need to generate one output, then we need to perform N additions (refer Eq. 1). Since this clearly shows high computational complexity, the choice for approximating additions in a digital filter makes complete sense.

II. Methodology

To explain the complete methodology, we have taken the famous Pan-Tompkins QRS peak detection Algorithm [7] which is discussed below. The results shown in the upcoming section are also for the Pan-Tompkins Algorithm but the methodology can be used for any biomedical signal processing algorithm that is error-resilient in nature. A block diagram of our methodology is given in Fig. 2.

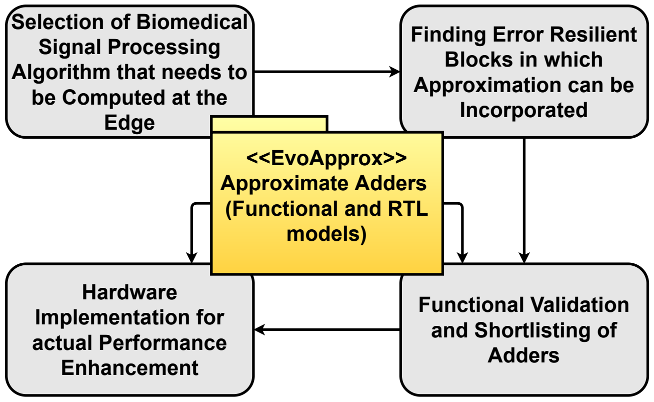
A. Pan-Tompkins Algorithm

The Pan-Tompkins Algorithm is a popular filtering technique that is used to detect QRS complexes in ECG signals [7]. The QRS complex is the combination of three of the graphical deflections seen on a typical electrocardiogram. It is the main spike seen on an ECG line. There are mainly 6

[1]Alish Kanani is a final year undergraduate student in the Electrical Engineering Department at Indian Institute of Technology Jodhpur, India. kanani.1@iitj.ac.in

[2]Rajat Bhattacharjya is a Software Engineer at Parallel Wireless, Inc. rbhattacharjya@parallelwireless.com

[3]Dip Sankar Banerjee is with the Computer Science and Engineering Department at Indian Institute of Technology Jodhpur, India. dipsankarb@iitj.ac.in



Selection of Biomedical Signal Processing Algorithm that needs to be Computed at the Edge
Finding Error Resilient Blocks in which Approximation can be Incorporated
<<EvoApprox>> Approximate Adders (Functional and RTL models)
Hardware Implementation for actual Performance Enhancement
Functional Validation and Shortlisting of Adders

Fig. 2: Block diagram of Methodology

stages involved in a typical Pan-Tompkins Algorithm which is shown in Fig. 1. Because of a simple yet efficient QRS detection algorithm, it is being used in a wide range of wearable ECG monitoring devices.

B. Finding Error Resilient Blocks

In most biomedical applications, filters are the important stage due to noisy data acquisition systems, especially in the wearables, as the sensors have to be energy efficient. While the end outcome should be accurate, so filters help in noise reduction as well as separation of the required information from noisy data, therefore they have to be error-resilient in nature. Any similar digital processing technique which deals with noisy data could be the targeted block in which approximation can be incorporated. In the Pan-Tompkins Algorithm, all the digital filters use the Finite Impulse Response (FIR) filtering method to filter the incoming data.

$$y[n] = \sum_{i=0}^{N} b_i \times x[n-i] \qquad (1)$$

As given in Eq. 1, the main operations to implement N point FIR filter are addition and multiplication (here b_is are filter coefficients and x is input data). We use EvoApprox [8] 16-bit sign-extended adders library as an add operation is involved in every cycle of filtering. Since four out of six blocks of the Pan-Tompkins Algorithm use the above filter implementation which is highlighted in Fig. 1, we have targeted these blocks for approximation. The same approach can be used for any biomedical signal processing algorithm.

C. Adder Selection by Functional Validation

As the output of any biomedical signal processing algorithm is used in diagnosis, the accuracy of the final output is really important and should not be decreased below a certain threshold. So, to choose an approximate adder, one or more error metrics of the signal processing algorithm needs to be taken into account. After incorporating approximation, if the algorithm clears some predefined cutoff above which the application works well, the adders which are responsible for the approximation can be selected for further processing. In order to achieve this, we replace the accurate adders with any of the functional approximate adders that are being considered, in the chosen error-resilient blocks. We give some random yet relevant input and run the algorithm to decide whether the selected approximate adder can be used further.

In the case of the Pan-Tompkins Algorithm, we have replaced the accurate adders in previously selected FIR filtering blocks highlighted in Fig. 1 with different Power vs Error parameters Pareto optimal approximate 16-bit sign-extended adders available in the EvoApprox library. Since the Pan-Tompkins Algorithm is used to detect QRS peak from the ECG, we have taken the output of Moving Window Integration and threshold value after Adaptive Thresholding to calculate different error metrics and chose 10 approximate adders for hardware implementation, which can be found in Table I.

D. Hardware Implementation

The actual benefit of incorporating approximate adders can be explored by hardware implementation of the algorithm. Based on the application of the selected algorithm, different hardware synthesis flows can be used. The first step is RTL implementation which is common for all synthesis flows. The quick way to compare the effect of different approximate adders is by implementing only the blocks in which the approximation is happening and getting power, area and timing results by synthesizing the HDL code using some ASIC Flow Tool.

For comparison purposes, since all the approximate blocks in the Pan-Tompkins Algorithm are FIR filters, we had implemented the FIR filter only once. Here, only the adder is replaced with the previously selected approximate adders' circuit from the EvoApprox library keeping all the other circuits accurate. We have used standard cell design flow to generate the required performance results. The detailed analysis is described in Section III-B.

III. Results and Analysis

So as to evaluate the effect of approximate additions in QRS peak detection using the Pan-Tompkins Algorithm, we first perform the functional validation, i.e, the accuracy analysis after approximating additions at the software level. Then we move on to checking the hardware efficiency achieved by employing those approximate adders used in functional validation at the hardware level.

For functional validation, we have implemented the algorithm in MATLAB and replaced the FIR filtering blocks with approximated filters. We have used the MIT-BIH Arrhythmia Database [9] as the raw ECG input.

In the hardware evaluation, we have described a 100 point FIR filter with Verilog HDL which is implemented with a single cycle Multiply–Accumulate unit, where the exact adder that is present is replaced with selected approximate adders and few accurate adders for comparison purposes. The RTL model is then synthesized using the 45nm NanGate Open Cell Library in Synopsys Design Compiler.

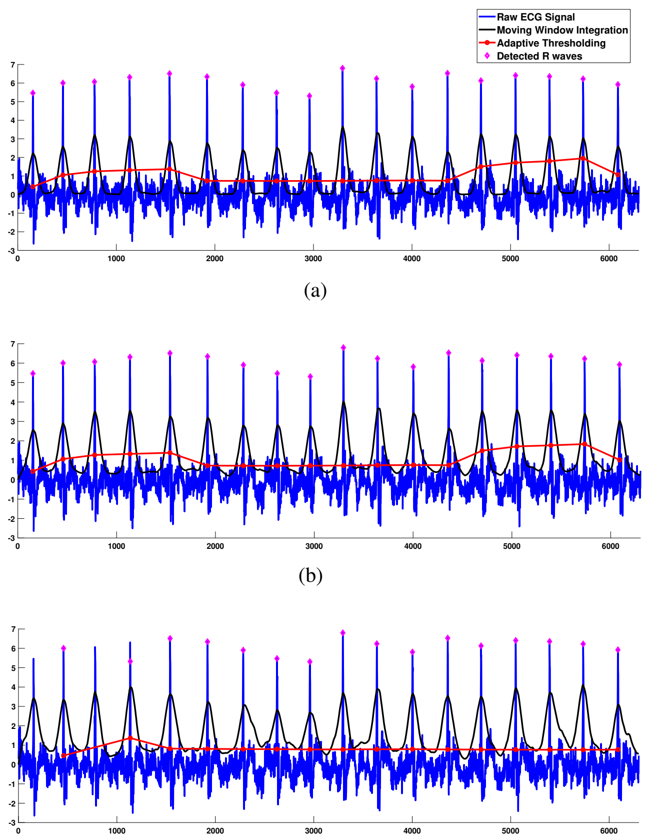


Fig. 3: Output from Moving Window Integration; Adaptive Thresholding and Detected R signals using (a) Accurate adder, (b) add16se_1Y7 adder and (c) add16se_29A adder

A. *Functional Validation*

For accuracy analysis and adder selection, we have calculated Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR) of Moving Window Integration; and Mean Square Error (MSE) of thresholds as error metrics. The accurate outputs of these two stages are used to calculate the same.

Fig. 3 shows the difference in output because of approximation. In Fig. 3(b) there is a small change in output of moving average compared to an accurate one, these outputs are calculated using add16se_1Y7 adder, similar results are also found with add16se_2H0, add16se_2JY, add16se_2LJ, add16se_20J, add16se_26Q and add16se_294 adders. While in Fig. 3(c), the difference is more evident as at the starting it fails to detect few peaks and the output of the threshold is also not following a similar trend compared to an accurate one. These outputs are calculated using add16se_29A adder while with add16se_2E1 and add16se_25S the outputs are similar. The reason for these differences can also be explained using the error matrix shown in Table I. For demonstration purposes we have shown the graphical output of only 2 adders, the detailed accuracy analysis can be seen in Table I.

After incorporating all the Pareto optimal approximate 16-

TABLE I: Performance Metrics for Functional Validation

Adder	SSIM	PSNR	MSE of threshold
add16se_1Y7	0.6147	7.3008	0.0011
add16se_2E1	0.4144	-1.5748	0.3699
add16se_2H0	0.973	34.0319	0.0001541
add16se_2JY	0.8193	18.3554	0.0016
add16se_2LJ	0.8934	24.3307	0.0002917
add16se_20J	0.9515	30.4783	7.80x10^{6}
add16se_25S	0.4743	0.9262	0.3752
add16se_26Q	0.9929	42.365	8.52x10^{6}
add16se_29A	0.4955	1.4894	0.277
add16se_294	0.6263	7.9271	0.061

TABLE II: Area, Worst-case delay and Power metrics of 100 point FIR filter with different approximate and accurate adders

Type of Adder	Adder	Area (μm^2)	Delay (ns)	Power (μW)
Approximate	**add16se_1Y7**	25639.20	2.24	15009
	add16se_2E1	24767.52	2.18	14309
	add16se_2H0	27691.92	2.16	16208
	add16se_2JY	26166.95	2.16	15209
	add16se_2LJ	27979.20	2.21	16348
	add16se_20J	27083.05	2.24	16072
	add16se_25S	25377.46	2.36	14619
	add16se_26Q	27163.92	2.16	15697
	add16se_29A	25688.42	2.3	15219
	add16se_294	26288.51	2.24	15622
Accurate	**Brent–Kung adder**	33272.08	2.23	18919
	Carry-Lookahead adder_2	29111.31	2.14	16274
	Carry-Lookahead adder_4	29111.31	2.15	16273
	Carry-Lookahead adder_8	30849.35	2.12	17850
	Carry-Select adder_2	38735.72	2.17	23593
	Carry-Select adder_4	39602.88	2.31	24171
	Carry-Select adder_8	36241.98	2.36	21428
	Carry-Skip adder_2	31905.37	2.6	18937
	Carry-Skip adder_4	32376.72	2.74	18916
	Carry-Skip adder_8	31498.65	2.61	17786
	Ripple-Carry adder	28768.96	2.37	16214

bit sign-extended adders available in the EvoApprox library, we have shortlisted 10 adders based on error metrics which are shown in Table I. Based on the accuracy requirement of the application, one can choose any adder from this list. The selected adders have a diverse range in terms of error metrics. The SSIM of moving window integration varies from 0.9929 (add16se_26Q) to 0.4144 (add16se_2E1), while the MSE of the threshold varies from 7.80×10^{6} (add16se_20J) to 0.3699 (add16se_2E1). The general error parameters for these adders like Error Rate, Mean Absolute Error, Mean Squared Error, etc. can be found in the EvoApprox library data [8].

B. *Hardware Evaluation*

As mentioned before, we have implemented a 100 point FIR filter in Verilog which is common for Low Pass, High Pass and Moving Window Integration filter blocks as shown in Fig. 1. The Differentiation block is implemented using [-(1/6),-(1/6),0,(1/6),(1/6)] filter coefficient. Now, since we have implemented the FIR filter using a single cycle Multiply-Accumulate unit, total N clocks are required for any data to appear at the output. Also, the squaring operation uses only multiply units, so the worst-case delay of squaring block is always less compared to an FIR filter.

To compare the performance result of approximate adders with accurate ones, we have used 16-bit Carry-Lookahead adder, Carry-Select adder and Carry-Skip adder with 2,4 and 8 block size (the naming convention of Carry-Lookahead adder_2 signifies that the block size is 2, it is similar for the other accurate adders being used in Table II. Block size implies the bit-width, i.e., if for a 16-bit number the block size is 2, then there will be 8 blocks of bit-width 2 bits each), we have also used the traditional Ripple Carry adder and additionally the Brent–Kung adder.

Worst-case delay, total area required for a 100 point FIR filter, and its power consumption is shown in Table II.

C. Analysis

The average SSIM considering the 7 good adders presented in Table I and Section III-A is 0.84, which is a quite high value. Considering all the adders presented in Table I, the average SSIM comes out to be 0.73. While the average MSE of 7 best approximate adders is $9.17 \times 10^{\ 3}$. Hence, it makes perfect sense to incorporate approximate adders for applications in biomedical signal processing.

We have observed that out of the 10 shortlisted approximate adders, 7 detect peaks accurately and these can be used in a real clinical setting. However, there is a slight variation in peak detection for 3 approximate adders, add16se_2E1, add16se_25S and add16se_29A. Still, in scenarios where accurate peak detection isn't required, these 3 approximate adders can be used.

We can see that there is a 19.27% power saving on average for all of the approximated adders in Table II relative to the accurate adders for implemented FIR filter. As shown in Table I, the approximate adder add16se_2E1 performs worse in terms of MSE, but from Table II, it can be derived that it saves 25.18% power compared to accurate adders, which is highest among all the approximate adders. This shows the trade-off of accuracy with hardware efficiency.

Coming to comparing area requirements, we can see that the FIR filter with approximate adder add16se_2E1 provides an area-saving of 24.63% compared to the average area of all accurate adders presented in Table II. Meanwhile, on average we see that the savings on the area front using approximate adders is 19.71% compared to accurate adders.

Worst-case delay statistics show that the circuits with approximate adders on an average are 5.13% faster than the accurate adders. However, at the edge, power and area are of primary concern, so we put our main focus on those two parameters. The statistics presented above clearly show why approximate adders are a good choice for biomedical signal processing.

IV. Related Works

The discussion for power efficiency in wearables is not new. Nia et al. [10] quantified the energy and storage requirements of WBAN that uses eight biomedical sensors. Their analysis suggested that there lies a major gap between the energy requirements for long-term continuous monitoring and the capabilities of devices that are currently present. However, the angle of introducing approximate computing to wearables hasn't been explored in depth. Prabakaran et al. [3] proposed a two-step methodology for employing approximate computing in biomedical signal processing. Their method involved two stages of quality evaluation that would enable determining the approximation parameters. Our method is different from them in the sense that we incorporate approximation during the functional stages of biomedical signal processing. If after approximation, the algorithm works well in terms of error resilience, we go for hardware implementation of the same. In addition to that, the approximation we incorporate is only for the addition operations involved in those stages. Also, we have incorporated already available approximate adders in a library [8] instead of fixed approximate full adders as used in [3]. The main motivation of using a library is the future scope and scalability as any new approximate adder with better performance can directly be incorporated into ApproxBioWear.

V. Conclusion

In this paper, we presented ApproxBioWear, an approach that can be used to increase hardware efficiency in wearables. The core concept behind this approach is approximating the addition operations involved in filtering steps of a biomedical signal processing algorithm so as to mainly save power and area. Upon employing the ApproxBioWear approach, we see that the application accuracy stays almost the same after approximation as compared to accurate operations. On average the presented methodology provides an area-saving of 19.71% and power-saving of 19.27%.

This paper is in accordance with the principles outlined in the Helsinki Declaration of 1975, as revised in 2000 for experiments involving human subjects.

References

[1] K. Diaz, D. Krupka, M. Chang, J. Peacock, Y. Ma, J. Goldsmith, J. Schwartz, and K. Davidson, "Fitbit®: An accurate and reliable device for wireless physical activity tracking," *International journal of cardiology*, vol. 185, pp. 138–140, 03 2015.

[2] A. Khushhal, S. Nichols, W. Evans, D. Gleadall-Siddall, R. Page, A. O'Doherty, S. Carroll, L. Ingle, and G. Abt, "Validity and reliability of the apple watch for measuring heart rate during exercise," *Sports Medicine International Open*, vol. 1, pp. E206–E211, 06 2017.

[3] B. S. Prabakaran, S. Rehman, and M. Shafique, "Xbiosip: A methodology for approximate bio-signal processing at the edge," ser. DAC '19. New York, NY, USA: Association for Computing Machinery, 2019.

[4] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *2013 18th IEEE European Test Symposium (ETS)*, 2013, pp. 1–6.

[5] A. Kanani, J. Mehta, and N. Goel, "ACA-CSU: A Carry Selection Based Accuracy Configurable Approximate Adder Design," in *2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2020, pp. 434–439.

[6] R. Bhattacharjya, V. Mishra, S. Singh, K. Goswami, and D. S. Banerjee, "An Approximate Carry Estimating Simultaneous Adder with Rectification," in *Proceedings of the 2020 on Great Lakes Symposium on VLSI*, ser. GLSVLSI '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 139–144.

[7] J. Pan and W. J. Tompkins, "A real-time qrs detection algorithm," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 3, pp. 230–236, 1985.

[8] V. Mrazek, R. Hrbacek, Z. Vasicek, and L. Sekanina, "Evoapprox8b: Library of approximate adders and multipliers for circuit design and benchmarking of approximation methods," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017, pp. 258–261.

[9] G. Moody and R. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[10] A. M. Nia, M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, "Energy-efficient long-term continuous personal health monitoring," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 1, no. 2, pp. 85–98, 2015.