An Energy-Efficient Programmable Manycore Accelerator for Personalized Biomedical Applications

Adwaya Kulkarni, Adam Page, Nasrin Attaran, Ali Jafari, Maria Malik, Houman Homayoun, and Tinoosh Mohsenin

***Abstract*— Wearable personalized health monitoring systems can offer a cost-effective solution for human health care. These systems must constantly monitor patients' physiological signals and provide highly accurate, and quick processing and delivery of the vast amount of data within a limited power and area footprint. These personalized biomedical applications require sampling and processing multiple streams of physiological signals with a varying number of channels and sampling rates. The processing typically consists of feature extraction, data fusion, and classification stages that require a large number of digital signal processing (DSP) and machine learning (ML) kernels. In response to these requirements, in this paper, a tiny, energy-efficient, and domain-specific manycore accelerator referred to as power-efficient nanoclusters (PENC) is proposed to map and execute the kernels of these applications. Simulation results show that the PENC is able to reduce energy consumption by up to 80% and 25% for DSP and ML kernels, respectively, when optimally parallelized. In addition, we fully implemented three compute-intensive personalized biomedical applications, namely, multichannel seizure detection, multiphysiological stress detection, and standalone tongue drive system (sTDS), to evaluate the proposed manycore performance relative to commodity embedded CPU, graphical processing unit (GPU), and field-programmable gate array (FPGA)-based implementations. For these three case studies, the energy consumption and the performance of the proposed PENC manycore, when acting as an accelerator along with an Intel Atom processor as a host, are compared with the existing commercial off-the-shelf general-purpose, customizable, and programmable embedded platforms, including Intel Atom, Xilinx Artix-7 FPGA, and NVIDIA TK1 advanced RISC machine -A15 and K1 GPU system on a chip. For these applications, the PENC manycore is able to significantly improve throughput and energy efficiency by up to 1872× and 276×, respectively. For the most computational intensive application of seizure detection, the PENC manycore is able to achieve a throughput of 15.22 giga-operations-per-second (GOPs), which is a 14× improvement in throughput over custom FPGA solution. For stress detection, the PENC achieves a throughput of 21.36 GOPs and an energy efficiency of 4.23 GOP/J, which is 14.87× and 2.28× better over FPGA implementation, respectively. For the sTDS application, the PENC improves a throughput by 5.45× and an energy efficiency by 2.37× over FPGA implementation.**

***Index Terms*— Low-power manycore accelerator, personalized biomedical applications, seizure detection, stress detection, tongue drive system (TDS).**

I. Introduction

RECENT innovations in the semiconductor industry made it possible to integrate various sensors and computing components in an embedded system-on-a-chip (SoC) processing platform. Wearable mobile platforms use embedded SoCs to process sophisticated and computationally intensive applications. With the rapid advances in small, low-cost wearable computing technologies, including smartphones and smartwatches, there is a tremendous opportunity to develop the ubiquitous personalized biomedical embedded systems capable of continuous vigilant monitoring of physiological signals. These systems have the potential to reduce the morbidity, mortality, and economic cost associated with many chronic diseases by enabling early intervention and preventing costly hospitalizations. In addition, recent advances in noninvasive sensor technologies enable the possibility that these systems can potentially monitor and analyze several modalities, including acceleration, pressure, temperature, electrocardiography (ECG), electromyography (EMG), electroencephalography (EEG), ultrasound, audio, and image signal streams. Embedded biomedical applications primarily consist of three basic stages: 1) a sensor front-end to capture and digitize physiological signals; 2) a processing stage to analyze, classify, and potentially store the sensors data; and 3) an RF module stage to transmit the data, classification, and/or diagnostics to the user or medical personnel [1]–[5]. There has been an incredible amount of innovation and improvement in sensor design that has dramatically reduced power while maintaining high accuracy. This is the result of technologies, such as microelectromechanical systems sensors and specialized analog-front-end (AFE) products targeted for physiological signals, such as Texas Instruments medical AFEs, namely, ADS129x and AFE44xx. There has also been a tremendous amount of work done on wireless RF modules ranging from specialized research modules to commercial modules, such as Bluetooth Smart (17.9-mA receiver, 18.2-mA transmitted, and 1-μA sleep). Still, the relatively high amount of power required to transmit raw or even compressed data makes

Manuscript received March 30, 2017; revised July 24, 2017; accepted August 30, 2017. Date of publication October 9, 2017; date of current version December 27, 2017. This work was supported by the National Science Foundation under Grant 1527151 and Grant 1329829. *(Corresponding author: Adwaya Kulkarni.)*

A. Kulkarni, A. Page, N. Attaran, A. Jafari, and T. Mohsenin are with the Department of Computer Science and Electrical Engineering, University of Maryland at Baltimore, Baltimore, MA 21250 USA (e-mail: adwayak1@umbc.edu).

M. Malik and H. Homayoun are with the Electrical and Computer Engineering Department, George Mason University, Fairfax, VA 22030 USA.

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

Digital Object Identifier 10.1109/TVLSI.2017.2754272

it essential to perform local onboard processing [6]. The enterprise of this paper is on the computing platform to address the unique challenges and the characteristics of biomedical applications. Realizing a low-power processor for biomedical computing in real time allows wearable biomedical devices to be capable of tracking the health and well-being of individuals with chronic disease using a holistic approach by integrating and interpreting multiple sensory inputs.

Current processor design, based on commodity general-purpose homogeneous processors, is not the most efficient in terms of performance/watt to process compute intensive applications [7]–[12]. To address the energy-efficiency challenge, heterogeneous architectures have emerged as the promising solutions in high performance as well as embedded platforms to significantly improve the energy efficiency by allowing applications to run on a computing core that matches the resource needs more closely than a single one-size-fits-all general-purpose core. A heterogeneous chip architecture integrates cores with various microarchitectures (in-order or out-of-order) or instruction set architectures (Thumb and $\times 86$) with on-chip graphical processing unit (GPU) or field programmable gate array (FPGA) accelerators to provide more opportunities for efficient workload mapping, so that the application can find a better match among various components to improve power efficiency. Examples of heterogeneous architectures in embedded domains are Xilinx ZYNQ (CPU + FPGA), NVIDIA Tegra TK1 and TX1 (Quad-core advanced RISC machine + CUDA embedded GPU), Qualcomm Snapdragon [CPU + digital signal processing (DSP) + GPU], and Samsung Exynos (Big + Little CPU + GPU). While conventional general-purpose heterogeneous architectures in wearable computing platforms promise to enhance energy efficiency significantly, they are not designed to handle the large diversity and the computational complexity of biomedical signals.

In fact, the state-of-the-art commodity general-purpose embedded platforms are not optimized to process this class of applications efficiently as they provide restricted choices with tradeoff between power, performance, and energy efficiency. Although integration with GPUs has provided opportunities to enhance the performance, it comes with significant power cost. In addition, to address the programmability challenge for diverse range of applications, these platforms are designed to provide general-purpose computing environments relying on enormous redundancy at various levels, deep and sophisticated memory hierarchy, and complex communication coherency network, which increase their inefficiency. Most recent works in developing a biomedical processor have focused on creating an SoC with specialized accelerator cores targeted for particular biomedical applications [13]–[17]. These approaches are not scalable to cover all kernels or applications, are often very expensive, and require long development time to develop specialized chips. Besides the major restrictions on power and area, the processor must be able to efficiently process several physiological signal streams with different characteristics. Table I provides some example common sensors with typical number of channels and sampling frequencies that are used by personalized biomedical applications. Processing these data streams often includes feature extraction, data fusion, and classification stages that consist of both DSP and machine learning (ML) kernels that exhibit task-level and data-level parallelism [18]. In response to all computing challenges of personalized biomedical applications discussed earlier, in this paper, we propose a programmable energy-efficient, domain-specific accelerator named power-efficient nanoclusters (PENC) to address the needs of biomedical signals to push the energy-efficiency boundaries to the next level. This paper, through an empirical setup on the state-of-the-art commodity embedded computing platforms and real measurements, makes the following major contributions.

TABLE I
EXAMPLE SENSORS FOR BIOMEDICAL APPLICATIONS WITH TYPICAL NUMBER OF CHANNELS AND SAMPLING FREQUENCIES. DEMONSTRATES THE VARIABLE SAMPLING FREQUENCIES AND MULTIPLE CHANNELS REQUIRED

Sensor	# Channels	Sampling Freq.	Description
HR	1 - 2	100 Hz	Heart Rate
SpO2	1 - 2	0.2 - 0.5 kHz	Blood Oxygen
ECG	3 - 12	0.2 - 1.0 kHz	Heart Elec Act.
EEG	6 - 64	0.1 - 1.0 kHz	Scalp Elec. Act.
GSR	1 - 4	50 - 100 Hz	Skin Conductance
EMG	4 >	1 - 2 kHz	Muscle Elec. Act.
RESP	1 - 3	50 - 100 Hz	Respiration

1) Propose PENC, an energy-efficient, domain-specific tiny programmable manycore accelerator to efficiently map and execute common kernels of personalized biomedical applications.
2) Develop the mappings of several DSP and ML kernels on PENC accelerator for energy-efficiency analysis.
3) Provide analysis in terms of performance and resource utilization of the DSP and ML kernels on FPGA, microcontroller, and multicore CPU- and GPU-based state-of-the-art embedded computing platforms along with comparison to the proposed PENC.
4) Perform thorough case study on three emerging compute-intensive biomedical signal processing applications, namely, multichannel seizure detection, multiphysiological stress detection, and stand-alone tongue drive system (sTDS), to fully evaluate PENC energy-efficiency advantage over commodity embedded solutions.

II. BACKGROUND

A. Related Work

1) Heterogeneous Processors: Heterogeneous architecture platforms have shown to provide significant advantages in enabling energy-efficient or area-efficient computing [9]–[11], [19], [20]. Integrating heterogeneous core in a multicore, such as advanced RISC machine (ARM) + million instructions per second (MIPS), CPU + GPU, or heterogeneous CPU + GPU + FPGA, has been investigated in various studies. In more complex heterogeneous architecture, multicore, GPU, and even FPGA have been integrated to solve the instruction level parallelism and task level parallelism challenges. An example for FPGA + CPU + GPU is the Axel system [21] and

NVIDIA Tegra K1 and X1, that combines the benefits of the specialization of FPGA, the parallelism of GPU, and the scalability of a multicore architecture. These examples show that the heterogeneous architecture can offer significant improvement for high computing demand applications. In general, in these systems, the overall performance can be improved by smart scheduling, allowing various heterogeneous computing components to work collaboratively on different parts of the program. In spite of all the performance benefit of integrating heterogeneous architectures, the challenge of high power consumption and high operating temperature remains an obstacle for deploying these designs in an embedded, wearable, and power constrained environment, including mobile devices. Particularly for many-cluster DSP and GPU platforms, while its been shown that these architectures are capable of providing the performance requirements of many computing intensive applications, they still suffer from high power consumptions and high operating temperatures [22]. Thus, these systems are impractical for resource constrained embedded portable environments. An example is the NVIDIA Tegra that can reach up to 10 W of power consumption that is not tolerable in resource-constrained biomedical embedded systems [23], [24]. A recent work has shown that each of multicore CPU- and GPU-based architectures offers a different power and performance tradeoff for various biomedical applications [16]. Although easy to program, these processors have limited flexibility and parallelism. Therefore, an FPGA is also explored, which provides high flexibility but requires writing low-level logic. In [15] and [24]–[26], a high-level synthesis (HLS) tool was used to generate an accelerator for ML kernels deployed in neural network and biomedical image processing and show significant performance and energy-efficiency benefit. However, as HLS is automated, it does not leverage all potentials of hardware acceleration. In this paper, in response, we use a custom, programmable manycore accelerator to leverage the enormous parallelism exists in biomedical applications to improve energy efficiency and benefit FPGA flexibility.

2) Domain-Specific Accelerator Processors: In the domain-specific platforms, several research works have been carried on the implementation of simpler cores for optimization rather than having application-specific processors. There has been work on simple programmable processors used for application-specific mapping. One such paper is [27], where 167 programmable processors having 16-kB shared memory implemented on 65-nm technology having an area of 0.17 mm^2, operating at 1.07 GHz consuming 47.5 mW when 100% active. This platform is dedicated for efficient computation of DSP, embedded, and multimedia applications, such as fast Fourier transform (FFT) and video encoding. There has also been a recent work on using simpler cores for high-performance computing applications in [28], and they propose an ultralow power platform built using tightly coupled processing cores called PULP. This manycore platform consists of clusters of simpler four OpenRISC cores, having 64 kB of L2 memory and 24 kB of tightly coupled data memory in 28-nm technology. This architecture is dedicated for computer vision applications, such as smart surveillance cameras and autonomous micro-unmanned aerial vehicles. A recent work has shown that KiloCore [18], a 32-nm 1000-processor computational platform, occupies 0.055 mm^2 area at a frequency of 1.78 GHz at 1.1 V. This chip consists of 1000 simple RISC types programmable processors and 12 independent memory modules. This platform is developed to address the concerns of extensive complex data computation, such as embedded Internet of Things (IOT) to cloud data centers for high-performance and energy-efficient computing. The proposed PENC manycore accelerator is different from other available platforms as it is a customized programmable architecture, targeting specifically personalized biomedical applications, with different characteristics than other studied domains.

3) Biomedical Processors: In the domain of general-purpose platforms for biomedical applications, recent work has shown how multicore architectures offer significant efficiency advantage over single core architecture when running various biomedical applications [29]–[32]. This is mainly motivated by the inherent parallelism existing in biomedical applications with multichannel signal analysis requirements, where multicore architectures can bring significant energy efficiency compared with a single core. Several research works have reported the performance results of parallel implementation of various computer vision-based biomedical applications on CPU and compared it with the accelerator implementations [16], [33], [34]. Cope *et al.* [35] and Kulkarni and Mohsenin [36] have compared the implementation performance of image convolution on GPU, FPGA, and CPU; Fykse [37] has compared image convolution processing on GPU and FPGA. Asano *et al.* [38] have investigated the performance comparison of 2-D filter on FPGA, GPU, and CPU; however, none of this paper has studied the tradeoff between power and performance on state-of-the-art embedded heterogeneous platforms. In the context of customized processor design, there have been a number of research endeavors exploring a single core or a multicore architecture design targeting specific biomedical applications. A massively parallel stream processor was introduced by Krimer *et al.* [39], which achieves 1 giga-operations-per-second (GOPs)/W. An ultralow energy processor with low voltage operations was presented by Hanson *et al.* [40] for wireless monitoring systems. The power consumption of the processor is optimized using a new low leakage memory, memory size and instruction set adjustments, and power gating. In another study, a sub/near threshold accelerator was proposed by Pu *et al.* [41] for low-energy mobile image processing using architecture-level parallelism. Rosen *et al.* [42] described a solution to implement predictable real-time applications on multiprocessors that uses a bus scheduling policy based on the time division multiple access. In their solution, processors are assigned time slots to access the bus with static scheduling. Their proposed multicore architecture [43] is used for the real-time biomedical monitoring and analysis system. Alemzadeh *et al.* [44] proposed a reconfigurable architecture for real-time assessment of individual's health status based on the development of a patient-specific health index and online analysis of multiparameter physiological signals. Bouwens *et al.* [45] proposed a dual-core system solution for wearable health monitors ECG R-peak detection

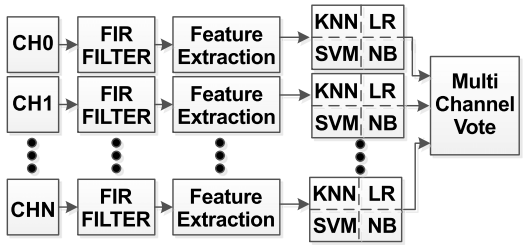


Fig. 1. Block diagram of a multichannel seizure detection application containing feature extraction, ML classifier, multichannel vote, and IO interface. The application highlights heavy use of DSP and ML kernels in addition to data-level and task-level parallelism.

application, which consumes 65.38 W. As discussed, most previous works have focused on creating an SoC by adding accelerator cores for a particular biomedical application. However, these modifications do not target the fundamental characteristics in common with a majority of biomedical applications. Besides the major restrictions on power and area, the processor must be able to efficiently process several physiological signal streams at often differing sampling frequencies.

B. Characteristics of Personalized Biomedical Applications

Among the many commonalities shared between personal biomedical applications, the need to process parallel streams of data in real time is a dominating feature. Table I showed that these applications require multichannel data streaming at various sampling rates. The analysis of these multiple streams requires a mix of data-level and task-level parallel computation [33], [34], [46], [47]. In addition, these applications often require a large number of DSP and ML techniques. DSP is often used to extract useful representations of the input data while the ML is needed to perform automated classification for diagnostic and detection purposes. In this paper, Fig. 1 shows the block diagram of the seizure detection application [1], [48]. This case study is an ideal example of a biomedical application that exhibits multiple streams (up to 24 EEG channels) of real-time data that must be processed with the DSP and ML kernels. In addition, the multiple streams allow for intuitive parallel processing. In order to demonstrate these dominant commonalities, we investigated various common DSP and ML kernels. The examined DSP kernels include filtering [finite impulse response (FIR)], windowing, FFT, orthogonal matching pursuit (OMP), and convolutional neural network (CNN) while the examined ML kernels include logistic regression (LR), naive Bayes (NB), support vector machine (SVM), and *k*-nearest neighbor (KNN).

In addition to exploring these various DSP and ML kernels, three case studies, including multichannel seizure detection, multiphysiological stress detection, and sTDS, are implemented on a number of general-purpose embedded hardware platforms. The platforms include Intel Atom processor, ARM Cortex-A15 processor, mobile TK1 GPU SoC, a Xilinx Artix-7 FPGA, and our proposed PENC manycore platform customized for personalized biomedical applications.

III. Power-Efficient Nano Clusters ManyCore

A. PENC Manycore Overview and Key Features

The PENC manycore accelerator is a homogeneous multiple instruction, multiple data (MIMD) architecture that consists of in-order tiny processors with a six-stage pipeline, an RISC-like DSP instruction set, and a Harvard Architecture model [33], [34], [47], [49], [50]. The core operates on a 16-bit data path with minimal instruction and data memory suitable for task-level and data-level parallelism. Furthermore, these cores have a low-complexity, minimal instruction set to further reduce area and power footprint. The lightweight cores also help to ensure that all used cores execute an application without an idle state, which can further reduce overall energy consumption. These light cores have simplified data memory, instruction memory, and instruction set architecture ensuring full utilization of their resources when used. The processor can support up to 128 instructions, 128 data memory, and provides 16 quick-access registers. In the network topology, a cluster consists of three cores that can perform intracluster communication directly via a bus and intercluster communication through a hierarchical routing architecture. Each cluster also contains a shared memory. Fig. 2 shows the block diagram of a 16 cluster version of the design, highlighting the processing cores in a bus-based cluster. Each core, bus, shared memory, and router were synthesized and fully placed and routed in a 65-nm CMOS technology using the Cadence SoC Encounter, and the results for one cluster are summarized in Fig. 2(e). The processing core contains additional buffering on the input in the form of a 32-element content-addressable memory (CAM). It is used to store packets from the bus and allow a finite-state machine (FSM) to find a word where the source core field corresponds to that in the IN instruction itself, where the IN instruction is used to communicate between the cores. For example, if the core is executing `IN 3`, the FSM searches through the CAM to find the first word whose source core is equal to three. This word is then presented to the processing core and processing continues. The PENC manycore architecture has three lightweight processing cores and a shared memory in a single cluster. Our initial manycore architecture design had four processing cores and a hierarchical router within a cluster, which was ideal for DSP kernels for minimal data storage and localized processing [51]. Since personalized biomedical applications use ML kernels, which often require large amount of memory for their model data, the previous architecture resulted in memory access time bottleneck. Hence, the proposed PENC manycore architecture replaces the four core implementation with three cores and a shared SRAM memory of 3K words and low latency bus-based architecture for intercluster communications, while maintaining the efficiency of low area and power consumption. Our initial results showed that the performance benefit of bringing additional cores within the cluster diminishes given the increase in total area, power consumption, and network congestion. Below are the key characteristics of the PENC manycore platform.

1) Bus-Based Cluster: Cores use the `IN` and `OUT` instructions to communicate with each other. When a core executes an `OUT` instruction, the data and relevant addressing information

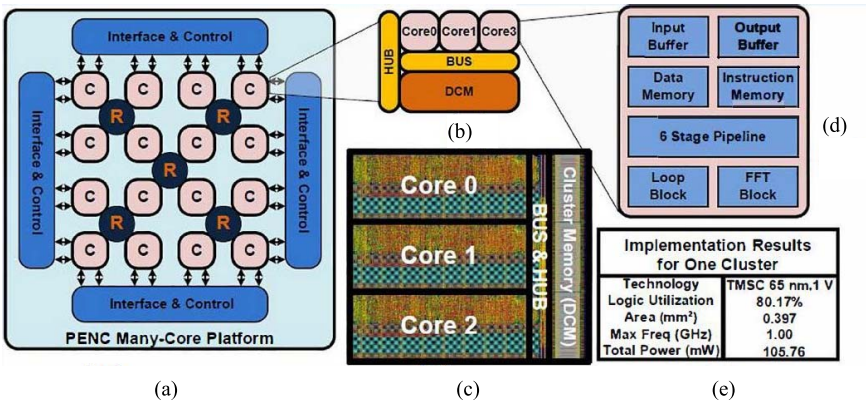


Fig. 2. (a) PENC manycore architecture. (b) Bus-based cluster architecture. (c) Postlayout view of bus-based cluster implemented in 65-nm, 1-V TSMC CMOS technology. (d) Block diagram of core architecture. (e) Postlayout implementation results of optimized bus-based cluster (consisting of three cores + bus + cluster memory).

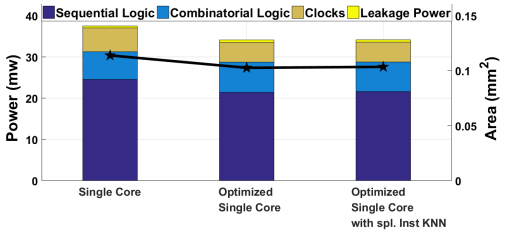


Fig. 3. Postlayout area and power analysis of different customizations of single processing core in PENC manycore architecture. Power is reported for 1-GHz clk.

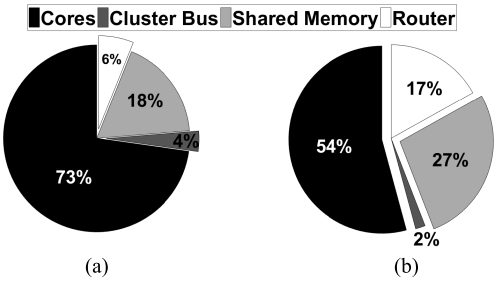


Fig. 4. Postlayout implementation breakdown analysis of PENC manymore comprising of 192 processing cores, cluster bus, shared memory, and router. (a) Area breakdown. (b) Power breakdown.

are packetized and sent to its output first-in-first-out (FIFO) through a bus. When data are present in a core's output FIFO, it requests to use the cluster bus. The bus then arbitrates between requests, only granting those whose transactions can be completed. The bus treats each transmission of data as a single transaction, since it behaves with a simple push or data-driven protocol. The bus is used for intracluster communication. This includes a round-robin arbiter, which chooses the next node to grant access based on the round-robin scheme. Once the node gets access, it wraps the processing core pipeline with layers of buffering and is the main level in the PENC architecture that interacts with the bus. The destination core is used by the bus to forward the packet to the appropriate location, and the source core is used by the requesting node to satisfy its corresponding `IN` instruction. Based on the destination address and the data fields, the recipient core stores the address of the data.

2) Domain-Specific Customization of Instruction Sets: Customizing a processor's instruction set for a particular computing domain is an efficient way of improving the processors performance. Designing an application-specific hardware for each given application is expensive; hence, a customized instruction set in the manycore can have a remarkable effect on power and area. The PENC architecture is optimized to best suited for ML kernels. There are lightweight processing cores containing a limited instruction set for efficiency with a handful of specialized instructions, such as absolute distance calculation and sorting. Fig. 3 shows the postlayout power and area results of single processing core with various optimizations (single core, optimized single core, and optimized single core with special instruction KNN) in terms of area and power. The optimized single processing core has five branching instructions removed, as they were redundant. This optimization managed to get reductions of 9.90% in area and 9.01% in power for a single processing core, and 7.40% in area and 6.86% in power for the PENC manycore architecture (192 cores, cluster bus, shared memory, and route). The optimized processing core with special KNN instruction is comprised of optimized single processing core with an added instruction for absolute distance calculation for KNN ML kernel. From the bar graph, it can be observed that this optimization has a reduction of 9.13% in area and 8.93% in power for a single processing core, and 6.83% in area and 6.80% in power for the PENC manycore architecture. Fig. 4 shows the postlayout implementation breakdown analysis of optimized PENC manycore comprising of 192 Processing cores, bus cluster, shared memory, and router with Fig. 4(a) showing the area breakdown and Fig. 4(b) showing the power breakdown. These results are obtained after Place and Route using Cadence Encounter for 65-nm technology. The area results come from the postlayout report, and the power results are obtained from the Encounter power analysis with careful consideration of activity factor, capacitance, IR drop, and rail analysis. These results are used to compare with the off-the-shelf processors.

3) Efficient Cluster Memory Access Architecture: While the lightweight cores are ideal for DSP kernels that require minimal static data [34], [47], ML kernels often require larger amounts of memory for their model data. This is addressed with the distributed cluster-level shared memory that is

interfaced to the bus. The shared memory within a cluster consists of three instances of SRAM cells of memory size 1024×16 bit making up a total of 3072 words and can be accessed within the cluster using the bus and from other clusters through the router. To access the memory, cores use two memory instructions: `LD` and `ST`. The maximum depth of the cluster memory is 2^{16} words, since registers and data memory are both 16-bit wide and can, therefore, supply a 16-bit memory address. Using data memory as operands for instructions is still beneficial to using `LD` and `ST` from an efficiency standpoint because of the one-cycle read/write capability. Referencing data from the cluster memory has latency and requires a separate instruction, which reduces the overall instructions per cycle that the pipeline can complete. However, the `LD` and `ST` instructions enable the use of a much larger addressable space, which allows the PENC to support many applications. The PENC architecture is ideally suited for personalized biomedical applications, which require to compute a variety of multiphysiological signals in real time within limited power budget. As previously shown in Table I and Fig. 1, these biomedical applications process many physiological signals at different sampling rates. The processing of these parallel signals requires both DSP and ML kernels that exhibit task-level and data-level parallelism. For PENC, each signal can be processed in parallel in different designated clusters. The proposed PENC features, including lightweight processing cores, domain-specific customization of instructions (i.e., sort, distance calculation, FFT, multiply and accumulate, as well as low latency memory and IO access instructions), and enhanced bus-based cluster architecture for low latency shared memory access make this MIMD platform address the needs of this class of applications. Section III-B provides empirical results showing how these manycore-specific features are well suited for personalized biomedical applications.

B. PENC Platform Evaluation Setup

For the PENC manycore, we developed stand-alone simulator and compiler that take user's code and postlayout hardware results as seen in Fig. 5. The simulator provides cycle accurate results, including completion time, instructions, and memory usage per core that directly come from the postlayout VLSI hardware of processors. It also serves as a reference implementation of the architecture to make testing, refining, and enhancing the architecture easier. Each task of algorithm is first implemented in assembly language on every processing core using manycore simulator. Assembly is a very low level language, which is equivalent to the actual instructions running on the processor. The simulator reads the assembly codes per core, compiles to binary, and puts them in the instruction memory to program the cores. It also initializes the register file and data memory in each core. It models the functionality of the processor and calculates the final state of register files and data memories. For execution time and energy consumption analysis of the algorithm, binaries obtained from the compiler are mapped onto the hardware design of the manycore platform, which is in Verilog and simulated using

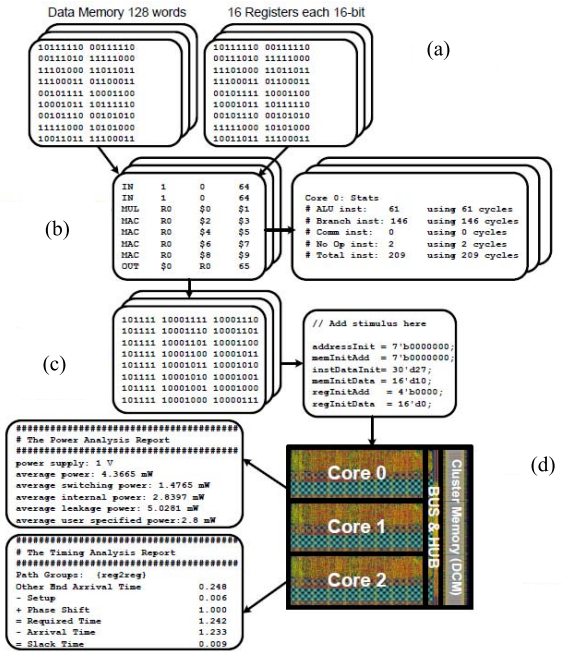


Fig. 5. Mapping of PENC manycore simulator and compiler flow high-level diagram and mapping the PENC hardware design using compiler. (a) Initialization of data memory and register. (b) Assembly code and compiler output. (c) Generated binary file and Verilog stimulus. (d) Hardware layout and power and timing report on Cadence tools.

Cadence NC-Verilog [52], as shown in Fig. 5. The activity factor is then derived and is used by the Cadence [52] Encounter tool for accurate power estimation of application running on the postlayout VLSI hardware of the manycore. The manycore simulator reports statistics, such as the number of cycles required for arithmetic logic unit, branch, and communication instructions, which are used for the throughput and energy analysis of the PENC manycore architecture.

C. PENC Evaluation on DSP and ML Kernels

In order to demonstrate the proposed PENC manycore's effectiveness at targeting personalized biomedical applications, experiments were performed that highlight the unique characteristics of these applications. Specifically, the experiments map various DSP and ML kernels with performance measured in energy, execution time, and memory demands.

1) DSP Kernel Mapping: In the first experiment, various DSP kernels were mapped onto the PENC manycore. The DSP kernels include FFT, FIR filter, OMP, dot-product operation (DOT), and CNN. In our previous work, we have designed specialized hardware for these kernels [1], [49], [50], [53]–[56]. For PENC manycore mapping, an initial mapping was performed that used the minimum number of cores to act as a baseline. A second mapping was then performed that used the optimal amount of cores. This was done by selecting the best from a number of implementations. The final mapping is equivalent to the second mapping but scales the frequency of each core to meet the execution time of the first mapping using dynamic frequency scaling (DFS). Fig. 6 shows all three of these mappings for the five DSP kernels with their corresponding energy-delay product (EDP). The plot shows that the manycore can

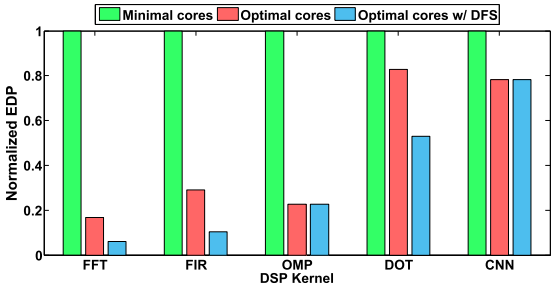
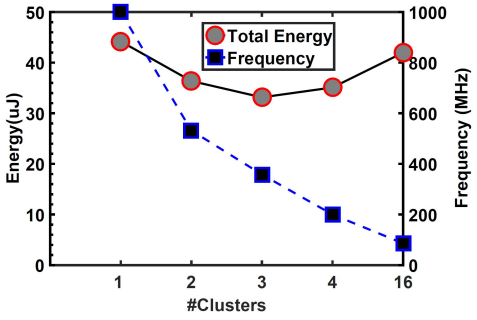


Fig. 6. Mappings of DSP kernels, including FFT, FIR filter, OMP, dot-product (DOT), and CNN. First mapping uses the minimum cores needed. The second mapping utilizes optimal cores to leverage parallelism. The third is the same as the second but with DFS.



Design	Cores	Mems	Routers
1 cluster	3	1	1
2 cluster	6	2	1
3 cluster	9	3	1
4 cluster	12	4	1
16 cluster	48	16	5

Fig. 7. Comparison of different mappings of feature extraction and KNN ML kernel with 512 training samples on the manycore with frequency scaling to meet deadline. Additional clusters can be utilized to exploit KNN parallel structure allowing to reduce frequency. Energy dissipation (with processing core, bus, shared memory, and router) and frequency values are shown in the plot. Table provides resources used for different mappings.

efficiently parallelize all of the kernels and is able to achieve an EDP reduction of up to 10×. It is important to note that kernels, such as CNN and OMP, do not use DFS, because these kernels are parallel and complete almost simultaneously. Therefore, their final mapping is the same as the mapping when optimal amount of cores are used. OMP maps [57] the sketching of 384 × 384 size image, and CNN maps the convolution layers of LENET-5 [58].

2) ML Kernel Mapping: Many DSP kernels require very little static and dynamic memory. For example, a 128-point FFT requires around 512 words of memory assuming twiddle factors are precomputed and the input is complex. On the other hand, many ML kernels can often require storing a large volume of model data. For example, KNN essentially requires storing all of the training data. This could correspond to thousands of values requiring to be stored (e.g., 17 000 data). This is accommodated for by having cluster-level shared memory accessible through the cluster's bus. The mapping of an ML kernel onto the manycore is performed similar to the mappings of the DSP kernels. The KNN algorithm with the 512 model data is mapped using between 1 and 16 clusters. The results are shown in Fig. 7. Fig. 7 shows different mappings of feature

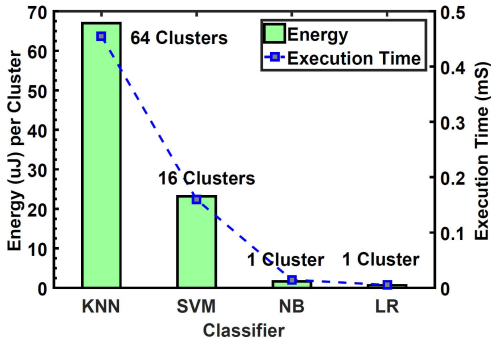


Fig. 8. Energy per cluster and execution time with the mappings of ML kernels, including KNN-3 (17 500 training samples), linear SVM (4937 support vectors), NB, and LR on PENC manycore platform.

extraction for KNN ML kernel using 512 training samples on the PENC with frequency scaling for each core. As can be seen, increasing the number of clusters to map KNN allows the operating clock frequency to be dramatically reduced. The optimal mapping is obtained using three clusters, which was able to reduce energy by 25% and execution time by 63% compared with single cluster. Fig. 8 shows energy per cluster and execution time of four ML (including KNN-3, linear SVM, LR, and NB) kernels mapping on PENC manycore. The required number of clusters to map the ML kernels on the manycore is shown in Fig. 8 as well.

IV. Case Studies

In this paper, we explore three applications, namely, stress detection, seizure detection, and TDS, to address the requirements for personalized biomedical applications that compute a variety of multiphysiological signals in real time within limited power budget. Seizure detection (Fig. 1) exploits multichannel parallel signal processing for 22 to 64 EEG channels. Stress detection in Fig. 12 exploits multiphysiological signal processing for heart rate (HR), accelerometer, respiration, and galvanic skin conductance. TDS in Fig. 15 exploits 3-D magnetic sensor data through tongue movement for 12 channels [59]. For the three applications that are implemented in this paper, processing of parallel data streams of the signals requires both DSP and ML kernels that exhibit task-level and data-level parallelism. These applications represent diversity both in terms of application type and variety number of sensors within biomedical signal processing domain as well as computational behavior and memory requirements. For example, TDS requires a very small training data, which can fit in PENC manycore data memory, and thus, PENC can be used as a stand-alone accelerator similar to the implementation for Artix FPGA and microcontroller as will be discussed in Table III and Fig. 18. For the Stress detection and seizure detection implementation (which require more data storage and transferring), the PENC accelerator runs with a host CPU (Intel Atom Edison Processor) for data marshaling. This would be similar to the operation of Jetson TK1 platform, which contain ARM processor interfaced with GPU. These applications are further discussed in this section.

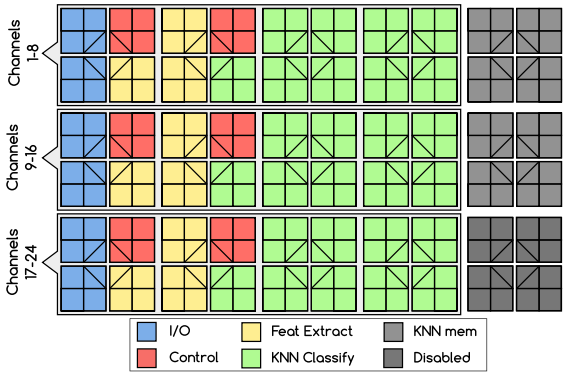


Fig. 9. Mapping of KNN-based seizure detection application onto PENC manycore.

A. Seizure Detection Application

Epilepsy is a leading neurological disease that affects approximately 2.2 million Americans. According to a recent Institute of Medicine report, epilepsy is the fourth most common neurological disorder in the United States with roughly 1 in 26 people being diagnosed with epilepsy in their lifetime [1]. The ability to monitor epileptic patients in an ambulatory setting is a crucial tool that has significant medical, psychosocial, cost, and safety advantages. For example, such a tool could be used to help determine minimal effective dosages or to alert medical personnel when a seizure is detected, which can help reduce the occurrences of sudden unexpected death in epilepsy.

In our previous work, a flexible seizure detection hardware system was implemented to detect the onset of a seizure by analyzing multiple channel, scalp-based EEG data in real time [1], [6], [48]. The developed system is capable of processing up to 24 channels of EEG electrodes that are digitized using specialized AFE ICs. Each stream of EEG sensor data is sampled at a rate of 256 Hz with 16-bit resolution. The processing consists of four main stages as previously shown in Fig. 1. Each EEG sensor is first passed through filters to remove high frequency and dc components. A feature extraction stage is then used to convert windows of time-series data into five temporal features per EEG channel. Each channel's features are then classified using one of four classifiers: KNN, SVM, NB, and LR. The last stage uses a multichannel voting scheme to determine the final classification.

For our study, the windows consist of 256 samples (1 s) with 50% overlapping windows. This means that a window will contain half-second of new data, which gives a 500-ms deadline to process each window. The mapping of the KNN version of the seizure detection application onto the PENC manycore can be seen in Fig. 9. The mapping highlights the parallelism that exists both between the EEG channels and within the KNN classifier kernel. For different ML classifiers, the task graph for seizure detection system is shown in Fig. 10.

B. Multiphysiological Stress Detection Application

Stress is a physiological response to the mental, emotional, and physical challenges that everyone encounters in their daily life [60]. There are strong links between stress and overall health, concentration, and ability to perform tasks. Predicting levels of stress using multimodal physiological sensors has been an active research topic in recent years [60]–[63]. These sensors usually include ECG, EMG, galvanic skin response, respiration (Resp), and accelerometer.

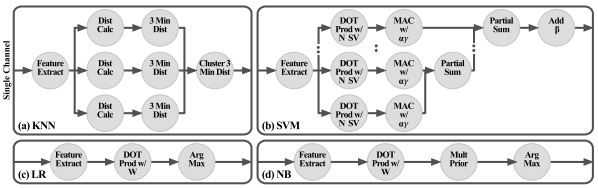


Fig. 10. Task graphs of each variation of the seizure detection application, when using KNN, SVM, LR, and NB ML algorithms. The graphs highlight the task-level parallelism and interconnect between features extraction and classification stage for each channel.

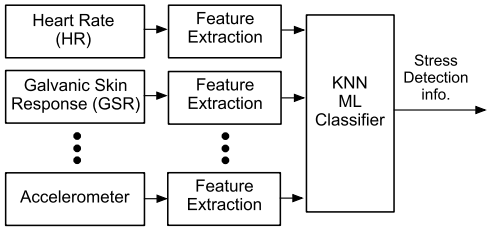


Fig. 11. Block diagram of a multiphysiological stress detection system containing data acquisition by sensors, feature extraction, and ML classifier to generate result.

In our previous work, a multimodal stress detection hardware system was implemented to detect the level of stress by analyzing multiple physiological signals, including HR and accelerometer [64]. The processing consists of three main stages, as shown in Fig. 11. The physiological sensor data are first passed through an initial filter stage to remove high frequency and dc components. A feature extraction stage is then used to convert windows of time-series data into four temporal features (one feature for HR and three features for accelerometer). Each feature sample is then classified using the KNN classifier. We used the data from a naturalistic shooting task in which stress was manipulated by incorporating different feedback modalities for making incorrect decisions [65]. Our explicit goal is to determine an algorithmic model from which the level of stress could be determined using multiphysiological signals. For our study, the windows consist of 6-s samples containing both HR and accelerometer signals with 50% overlapping windows. Fig. 12 shows the simulation environment from which data have been acquired. As a case study, the stress detection application was implemented on different platforms, including FPGA (Xilinx Artix-7 XC7A200T), TK1 GPU, and PENC manycore.

C. Standalone Tongue Drive System

The sTDS developed at the GTBIONICS Lab in the Georgia Institute of Technology and the University of Maryland, Baltimore County is an assistive, unobtrusive tongue-operated device that allows for real-time tacking of the voluntary tongue motion in the oral space for communication, control, and navigation applications [66]. Fig. 13 shows the sTDS, which

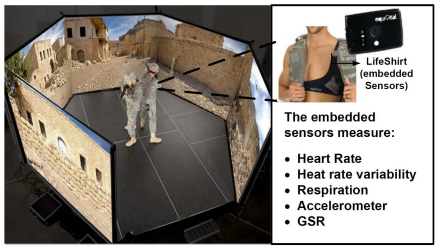


Fig. 12. 300° simulator to collect the multiphysiological data during different levels of stress using the embedded sensors in wearable life shirt [65].

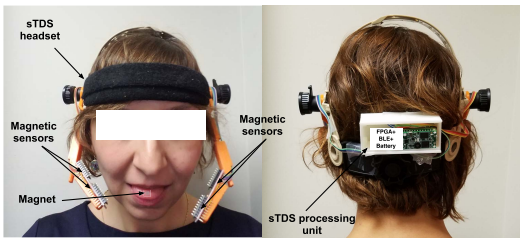


Fig. 13. sTDS prototype placed on a headset, which includes a low-power FPGA, four magnetic sensors, a Bluetooth low energy transceiver, a battery, and a magnetic tracer, which is glued to the users tongue.

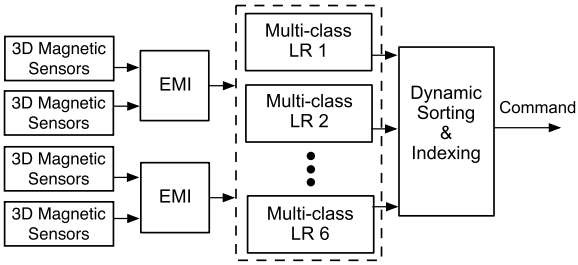


Fig. 14. Block diagram of the sTDS containing external magnetic interference (EMI) cancellation kernel and multiclass ML classifier where LR is used.

is placed on a headset. The sTDS device is a useful assistive technology that can substitute some of the hand functions with tongue motions. sTDS detects user's tongue movements through sensing the changes in the magnetic field generated by a small magnetic tracer, roughly the size of lentil, adhered to the tongue. The processing consists of converting these real-time magnetic field input streams into discretized commands to control environment. Fig. 14 shows the functional block diagram of the sTDS. The sensory input consists of four 3-D magnetic sensors that provide X-, Y-, and Z-axis magnetic field readings at a sampling rate of 50 Hz. In the first stage, the data are sent through an external magnetic interference (EMI) attenuation block, which utilizes regression analysis to remove noise artifacts as well as Earth magnetic field. Once this stage is complete, all the data are then fed into an ML classifier stage that makes a final classification based on these samples. The use of temporal and spatial components helps to dramatically reduce error. LR is implemented as the ML classifier, and the detection accuracy of LR is 96.6%. As shown in the block diagram in Fig. 14, similar to the seizure application, the task-level and the data-level parallelism exist. Task-level parallelism exists in the data acquisition, EMI, and ML classifier modules. These analysis results show that the LR is the best candidate for the proposed sTDS, because not only could it achieve similar accuracy compared with another algorithm, but it also consumes lower energy consumption and needs smaller memory for saving the calibration coefficients. Hence, LR is chosen as the ML classifier, and it is implemented on different hardware platforms. As a case study, the sTDS was implemented on different platforms, including FPGA (Xilinx Artix-7 XA7A15T), microcontroller (ARM Cortex-M4), and PENC manycore, and the results will be discussed in Section VI.

V. Off-the-Shelf Platforms and Experimental Setup

To better gauge the performance of the PENC manycore processor for personalized biomedical applications, we compared against several commercial off-the-shelf general-purpose and programmable processing platforms for all three case studies conducted in this paper. In order to do this, we targeted a number of platforms that contain low-power ARM-based CPUs, Intel embedded ×86-based CPUs, FPGAs, and embedded GPUs.

For each case study, we obtain the execution time and power consumption required to classify sample data across a variety of processor combinations. This is achieved by actively recording these metrics for a large number of samples and then averaging to derive the per classification performance. For power results, we measure the power consumption of both the processor and any external memory required. For power measurements, we used the in-house hardware simulator for PENC, while for Artix FPGA, we used Xilinx Xpower Analyzer, and for other hardware platforms, we did board measurements. While a few platforms, such as Intel Atom Edison and Jetson TK1, include built-in monitoring capabilities, we utilized an external TI INA219 voltage and power IC connected to each system's main power rails to ensure measurement consistency, which is shown in Fig. 16. For each platform, great care was taken to disconnect and power OFF all other peripherals, including HDMI, debug circuitry, and Wi-Fi/Bluetooth. The following discusses the details of the targeted platforms, including the board capabilities, processors included, and application mappings.

A. NVIDIA Jetson TK1

NVIDIA's Jetson TK1 is an SoC combining the Kepler GPU and a 4-plus-1 ARM processor arrangement. The 4-plus-1 processor configuration consists of five Cortex ARM-A15 processors, four high-performance processors, and one low power processor. Each ARM A15 CPU has a 32-kB L1 data and instruction cache supporting 128-bit NEON general-purpose single instruction and single instruction multiple data (SIMD) instructions. All processors configuration have shared access to a 2-MB L2 cache. For both the stress and seizure detection applications, we have experimented with using the embedded K1 GPU as an efficient accelerator; however, sTDS, which requires less processing, does not take advantage of using a GPU. Torch, a scientific computing framework, was used to efficiently implement both of these applications on the CPUs and embedded GPU. By exploiting

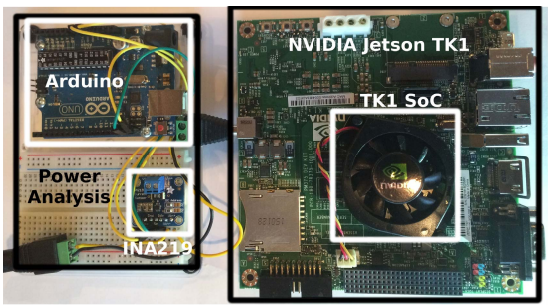


Fig. 15. Experimental setup to obtain power and execution time measurements of NVIDIA Jetson TK1 (as well as Intel Edison) platforms using TI INA219 and Arduino.

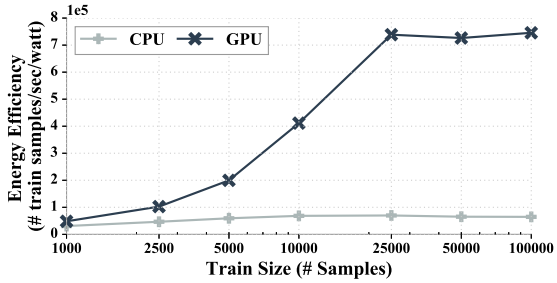


Fig. 16. Comparison of KNN kernel on Jetson TK1 when using quad-core ARM-A15 and embedded K1 GPU. Utilizing the GPU enables significantly improving energy efficiency by up to 11×.

the GPU, we are able to achieve several orders of magnitude energy-efficiency improvement over the ARM CPU counterpart. For example, Fig. 17 shows the improvement in energy efficiency of KNN when varying the model size with and without the GPU. For larger model sizes that exhibit higher parallelism, the GPU is able to improve efficiency by up to 11× over using quad-core CPU. The improvement tapers off once the GPU is maximally utilized.

B. Intel Edison

The Intel Edison is a low-power platform targeted for wearable devices and IOT. It contains an ultralow-power SoC with a dual-core Intel Atom processor (IA-32), 1 GB of double data rate type three (DDR3), Wi-Fi, Bluetooth, and 4-GB embedded multimedia controller memory running at a fixed clock of 500 MHz. The Intel Edison platform is used to obtain results by using Intel Atom processors stand alone as well as acting as a host for the PENC manycore accelerator. When using solely the Atom processors, great care was taken to efficiently utilize the low power ×86 cores. This was done using SIMD optimizations performed both by the compiler and in the code. Furthermore, parallelism was exploited by multithreading wherever possible, such as across EEG channels in the seizure detection application. The GCC/G++ compiler was passed using architecture's appropriate flags to increase the compilers effort on performance, such as −O3, mtune = native, and specification of the floating point unit of streaming SIMD extensions 4.2. To enhance the SIMD further, Intel Performance Primitives were used when the compiler could not vectorize or correctly map the functions to SIMD instructions. When the manycore is interfaced as an accelerator (for seizure and stress detection applications), the Intel Edison is used as the host to perform data marshaling. In this case, the system toggles between active mode to transfer windows data and sleep mode otherwise.

TABLE II
ARTIX-7 FPGA PERFORMANCE FOR DIFFERENT CASE STUDIES. BOTH DYNAMIC POWER AND TOTAL RESULTS ARE PRESENTED FOR FPGA CORE ONLY

Design	sTDS	Stress detection	Seizure detection (KNN)
FPGA package	XA7A15T	XC7A200T	XC7A200T
Slice count (#)	194	204	3,788
Memory (Kb)	0.8	1200	2,917
Operating freq. (MHz)	20	220	100
Latency (cycles)	132	57,603	644,622
Dynamic Power (mW)	2	42	274
Leakage Power (mW)	70	132	122
Total Energy	462 nJ	45.5 uJ	2.55 mJ

C. Xilinx Artix-7 Cmod-A7 and Nexys

As alternatives to traditional software-based CPU and GPU solutions, Cmod-A7 and Nexys platforms enable targeting Xilinx Artix-7 FPGA. FPGAs are highly flexible, allowing on-the-fly configuration to optimize bit resolution, clock frequency, parallelization, and pipelining for a given application. In addition, modern FPGAs provide accelerators to boost the performance for operations, such as multipliers, generic DSP cores, and embedded memories. The main disadvantages of FPGAs, however, are that they have substantially higher leakage power and require writing low level logic blocks in Hardware Description Language. For all three case studies, complete FPGA hardware solutions were developed in Verilog that utilized highly parallel, highly pipelined DSP and ML kernels. Both real-time and simulated projections using commercial tools were used to perform timing and power analysis when running test stimulus. For the sTDS application, the smallest Artix 7 FPGA, Artix-15T, is targeted on the Cmod-A7 platform. For stress and seizure detection applications, the Artix-200T FPGA is targeted on the Nexys platform. Table II summarizes the results of implementing each case study onto its respective Artix FPGA.

VI. IMPLEMENTATION RESULTS AND PLATFORM COMPARISON

For each case study, complete implementations are performed onto a subset of platform configurations best suited for the particular task. For sTDS application, which contains the least complexity, the processing platforms targeted include Atmega328 microcontroller, Artix-7 15T FPGA, and PENC manycore in stand-alone mode. For stress and seizure detection, we target the Artix-7 200T FPGA on Nexys, embedded K1 GPU on NVIDIA TK1, and PENC manycore with Intel Edison acting as host. In addition, seizure detection application also has implementation results using solely ×86-based CPU of Intel Edison. Table III provides results for all three applications, including throughput, power, energy, and energy efficiency. In Table III, results for all three applications are

TABLE III

BREAKDOWN OF HARDWARE RESULTS FROM RUNNING ALL THREE APPLICATIONS ON A VARIETY OF PROCESSING PLATFORMS. RESULTS INCLUDE THROUGHPUT, ENERGY, AND ENERGY EFFICIENCY. FOR EACH APPLICATION, RELATIVE IMPROVEMENT OF ENERGY EFFICIENCY OVER LOWEST PERFORMING PLATFORM IS PROVIDED. dec/sec CORRESPONDS TO CLASSIFICATION DECISION PER SECOND. NOTE THAT THE RESULTS FOR ALL THREE APPLICATIONS ARE RECORDED WHEN EACH PLATFORM IS EXECUTING AT ITS MAXIMUM CLOCK FREQUENCY. PENC ACCELERATOR OPERATES STAND ALONE FOR THE sTDS APPLICATION SIMILAR TO FPGA AND MICROCONTROLLER, WHILE FOR STRESS DETECTION AND SEIZURE DETECTION APPLICATIONS, THE PENC ACCELERATOR RUNS WITH A HOST CPU (ATOM PROCESSOR) FOR DATA MARSHALING

Case Study	Platform				Application Evaluation				
	Processor	Clock (MHz)	Power (mW)	Area (mm2)	Throughput (dec/sec)	Energy (mJ)	Energy Efficiency (dec/sec/watt)	Energy Efficiency (GOP/J)	Energy Efficiency (Rel. Improv)
sTDS	Atmega328 uController	8	24.4	25	440	5.54E-02	1.80E+04	0.0096	**1x**
	Artix-7 15T FPGA	20	72	225	151,520	4.75E-04	2.10E+06	1.11	**116x**
	PENC Manycore	1,000	166	0.32	823,723	2.01E-04	4.97E+06	2.64	**276x**
Stress Detection	Artix-7 200T FPGA	220	774	361	3,831	2.02E-01	4,950	1.86	**74x**
	Jetson TK1 GPU SoC	800	4,250	529	286	1.49E+01	67	0.025	**1x**
	PENC Manycore + Atom	1,000	5,050	175	56,961	8.87E-02	11,279	4.23	**168x**
Seizure Detection w/ KNN	Dual-core Atom	500	3,500	144	123	2.85E+01	35	0.24	**1x**
	Artix-7 200T FPGA	100	995	361	155	6.41E+00	156	1.08	**4.5x**
	Jetson TK1 GPU SoC	800	5,450	529	282	1.94E+01	52	0.36	**1.5x**
	PENC Manycore + Atom	1,000	12,414	175	2,196	5.65E+00	177	1.23	**5.1x**

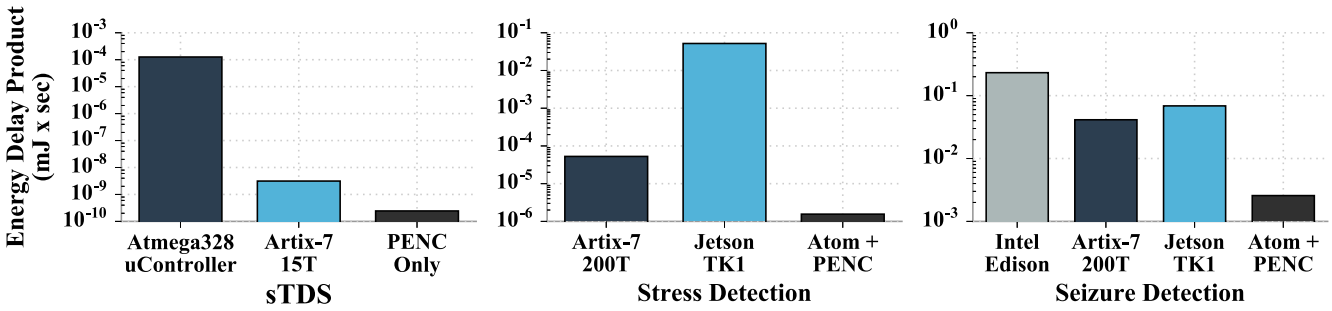


Fig. 17. Comparison of EDP for three case studies when implemented on several processor combinations, including Atmega328 microcontroller, Artix-7 FPGA, Jetson TK1, and PENC manycore. The EDP is calculated as energy/throughput, where throughput is the number of decisions per second (i.e., inverse of the time taken to complete one classification).

recorded when each platform is executing at its maximum clock frequency. However, for this class of personalized biomedical applications, the sampling frequency is relatively low in the range of 50 Hz–2 KHz as shown in Table I. Therefore, PENC and other platforms can run at much lower frequency to meet the application deadline, and thus significantly lower the power consumption. PENC has DFS feature built in each core, which allows each core to adjust its frequency according to the kernel/application deadline, and this was shown in Fig. 6. To better understand the benefit of PENC manycore, Fig. 17 provides comparisons of manycore to COTS processor combinations in terms of EDP for sTDS, stress detection, and seizure detection applications. In all scenarios, the PENC manycore has significantly lower EDP than all other studied processors. The EDP is calculated as energy/throughput, where throughput is the number of decisions per second (i.e., inverse of the time taken to complete one classification). For example, for seizure detection, the time to complete for PENC is 0.45 ms and energy is 5.65 mJ; thus, EDP is 0.002 mJ $\times$ s. Minimizing EDP is important for personalized biomedical applications as it is critical to both promptly making decisions and to do so with minimal energy. The custom FPGA solutions achieve the second best EDP for all three applications but have the main disadvantage of long development time to design hardware-defined solution. Furthermore, the PENC manycore requires 13$\times$, 34$\times$, and 16$\times$ lower EDP compared with FPGA solution for sTDS, stress, and seizure detection, respectively.

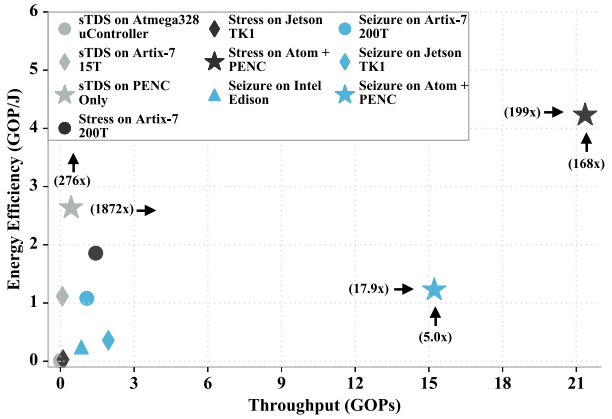


Fig. 18. Comparison of energy efficiency (GOP/J) versus throughput (GOPs) for all three case studies implemented on several processor combinations.

In Fig. 18, the processing combinations for all three applications are further evaluated in terms of energy efficiency versus throughput. We utilize GOPs to normalize based on computation complexity of each application when determining efficiency and throughput. As demonstrated in the plot, the PENC manycore is able to improve performance along both of these dimensions. For seizure application, which exhibits greatest complexity of approximately 7 million operations per classification, utilizing the PENC manycore in

concert with Intel Edison host is able to improve energy efficiency by 18× and throughput by 5× over just using the host processor. The high throughput is achieved due to significant level of parallelism that can be exploited across the EEG channels. On the other hand, for sTDS that contains far less levels of parallelism, the manycore is able to exploit pipelining similar to FPGA to significantly improve performance over single-core architecture.

VII. Conclusion

This paper explores the choice of embedded architectures for energy-efficient processing of personalized biomedical applications. Biomedical applications share strong commonalities requiring sampling from a number of physiological signals and processing that contains various DSP and ML kernels. The software, as well as hardware implementations of ML personalized biomedical applications, is compared. For the choice of software, the state-of-the-art commercial off-the-shelf embedded processing platforms, such as ARM and Atom CPUs along with K1 GPU, are compared with the hardware implementation of these kernels on embedded low-power FPGA. To further push the energy efficiency, a custom lightweight, symmetric manycore architecture is proposed that enables exploiting task-level and data-level parallelism within biomedical kernels, DFS, and specialized instructions and memory architecture to significantly reduce the energy usage. By using the optimal number of cores with DFS, we demonstrated the ability to reduce energy usage by up to 80% and 25% for DSP and ML tasks, respectively, relative to using the minimal number of cores. The PENC manycore requires 13×, 34×, and 16× lower EDP compared with FPGA solution for sTDS, stress, and seizure detection, respectively. The PENC manycore was further compared with other commercial off-the-shelf platforms for three compute-intensive personalized biomedical applications, including sTDS, stress detection, and seizure detection. For these end-to-end applications, the PENC manycore is able to significantly improve throughput and energy efficiency by up to 1872× and 276×, respectively. For the most computationally intensive application of seizure detection, the PENC manycore is able to achieve a throughput of 15.22 GOPs, which is a 14× improvement in throughput over custom FPGA solution. For stress detection, the PENC achieves a throughput of 21.36 GOPs and an energy efficiency of 4.23 GOP/J, which improves the throughput by 14.87× and the energy efficiency by 2.28× over FPGA implementation, respectively. For sTDS, the PENC improves the throughput by 5.45× and the energy efficiency by 2.37× over FPGA implementation.

Acknowledgment

The authors would like to thank A. Kulkarni, C. Shea, T. Abtahi, and A. Puranik for some preliminary results in this paper.

References

[1] A. Page, C. Sagedy, E. Smith, N. Attaran, T. Oates, and T. Mohsenin, "A flexible multichannel EEG feature extractor and classifier for seizure detection," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 2, pp. 109–113, Feb. 2015.

[2] S. Viseh, M. Ghovanloo, and T. Mohsenin, "Toward an ultralow-power onboard processor for tongue drive system," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 2, pp. 174–178, Feb. 2015.

[3] A. Jafari *et al.*, "An EEG artifact identification embedded system using ICA and multi-instance learning," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017.

[4] J. Yoo, L. Yan, D. El-Damak, M. A. B. Altaf, A. H. Shoeb, and A. P. Chandrakasan, "An 8-channel scalable eeg acquisition SoC with patient-specific seizure classification and recording processor," *IEEE J. Solid-State Circuits*, vol. 48, no. 1, pp. 214–228, Jan. 2013.

[5] K. H. Lee and N. Verma, "A low-power processor with configurable embedded machine-learning accelerators for high-order and adaptive analysis of medical-sensor signals," *IEEE J. Solid-State Circuits*, vol. 48, no. 7, pp. 1625–1637, Jul. 2013.

[6] A. Jafari and T. Mohsenin, "A low power seizure detection processor based on direct use of compressively-sensed data and employing a deterministic random matrix," in *Proc. IEEE Biomed. Circuits Syst. (Biocas) Conf.*, Oct. 2015, pp. 1–4.

[7] M. Malik and H. Homayoun, "Big data on low power cores: Are low power embedded processors a good fit for the big data workloads?" in *Proc. 33rd IEEE Int. Conf. Comput. Design (ICCD)*, Oct. 2015, pp. 379–382.

[8] M. Malik, S. Rafatirah, A. Sasan, and H. Homayoun, "System and architecture level characterization of big data applications on big and little core server architectures," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Oct. 2015, pp. 85–94.

[9] M. K. Tavana, M. H. Hajkazemi, D. Pathak, I. Savidis, and H. Homayoun, "ElasticCore: Enabling dynamic heterogeneity with joint core and voltage/frequency scaling," in *Proc. 52nd Annu. Design Autom. Conf.*, 2015, p. 151.

[10] V. Kontorinis, M. K. Tavana, M. H. Hajkazemi, D. M. Tullsen, and H. Homayoun, "Enabling dynamic heterogeneity through core-on-core stacking," in *Proc. 51st ACM/EDAC/IEEE Annu. Design Autom. Conf.*, Jun. 2014, pp. 1–6.

[11] H. Homayoun, V. Kontorinis, A. Shayan, T.-W. Lin, and D. M. Tullsen, "Dynamically heterogeneous cores through 3d resource pooling," in *Proc. IEEE 18th Int. Symp. High-Perform. Comput. Archit.*, Feb. 2012, pp. 1–12.

[12] A. Lukefahr *et al.*, "Composite cores: Pushing heterogeneity into a core," in *Proc. 45th Annu. IEEE/ACM Int. Symp. Microarchitecture*, Dec. 2012, pp. 317–328.

[13] C. Kim, M. Chung, Y. Cho, M. Konijnenburg, S. Ryu, and J. Kim, "ULP-SRP: Ultra low power samsung reconfigurable processor for biomedical applications," in *Proc. Int. Conf. Field-Programm. Technol. (FPT)*, 2012, pp. 329–334.

[14] S.-Y. Hsu *et al.*, "A sub-100μw multi-functional cardiac signal processor for mobile healthcare applications," in *Proc. Symp. VLSI Circuits (VLSIC)*, 2012, pp. 156–157.

[15] K. Neshatpour *et al.*, "Big biomedical image processing hardware acceleration: A case study for K-means and image filtering," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 1134–1137.

[16] M. Malik *et al.*, "Architecture exploration for energy-efficient embedded vision applications: From general purpose processor to domain specific accelerator," in *Proc. EEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Pittsburgh, PA, USA, Jul. 2016, pp. 559–564.

[17] K. Neshatpour, M. Malik, M. A. Ghodrat, and H. Homayoun, "Accelerating big data analytics using FPGAs," in *Proc. IEEE 23rd Annu. Int. Symp. Field-Programm. Custom Comput. Mach. (FCCM)*, Washington, DC, USA, May 2015, p. 164.

[18] B. Bohnenstiehl *et al.*, "KiloCore: A 32-nm 1000-processor computational array," *IEEE J. Solid-State Circuits*, vol. 52, no. 4, pp. 891–902, Apr. 2017.

[19] R. Kumar, K. I. Farkas, N. P. Jouppi, P. Ranganathan, and D. M. Tullsen, "Single-ISA heterogeneous multi-core architectures: The potential for processor power reduction," in *Proc. 36th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Dec. 2003, pp. 81–92.

[20] K. Neshatpour, M. Malik, and H. Homayoun, "Accelerating machine learning kernel in hadoop using FPGAs," in *Proc. 15th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCGrid)*, May 2015, pp. 1151–1154.

[21] K. H. Tsoi and W. Luk, "Axel: A heterogeneous cluster with FPGAs and GPUs," in *Proc. 18th Annu. ACM/SIGDA Int. Symp. Field Program. Gate Arrays*, Feb. 2010, pp. 115–124.

[22] X. Mei, L. S. Yung, K. Zhao, and X. Chu, "A measurement study of GPU DVFS on energy conservation," in *Proc. Workshop Power-Aware Comput. Syst.*, 2013, Art. no. 10.

[23] A. Kulkarni, C. Shea, T. Abtahi, and T. Mohsenin, "Low overhead cs-based heterogeneous framework for big data acceleration," *ACM Trans. Embedded Comput. Syst.*, to be published.

[24] A. Page, A. Jafari, C. Shea, and T. Mohsenin, "SPARCNet: A hardware accelerator for efficient deployment of sparse convolutional networks," *J. Emerg. Technol. Comput. Syst.*, vol. 13, no. 3, pp. 31:1–31:32, May 2017. [Online]. Available: http://doi.acm.org/10.1145/3005448

[25] K. Neshatpour, M. Malik, M. A. Ghodrat, A. Sasan, and H. Homayoun, "Energy-efficient acceleration of big data analytics applications using FPGAs," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Nov. 2015, pp. 115–123.

[26] K. Neshatpour, A. Sasan, and H. Homayoun, "Big data analytics on heterogeneous accelerator architectures," in *Proc. Int. Conf. Hardw./Softw. Codesign Syst. Synthesis (CODES+ISSS)*, Oct. 2016, pp. 1–3.

[27] D. N. Truong *et al.*, "A 167-processor computational platform in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 44, no. 4, pp. 1130–1144, Apr. 2009.

[28] F. Conti, D. Rossi, A. Pullini, I. Loi, and L. Benini, "PULP: A ultra-low power parallel accelerator for energy-efficient and flexible embedded vision," *J. Signal Process. Syst.*, vol. 84, no. 3, pp. 339–354, Sep. 2016. [Online]. Available: http://dx.doi.org/10.1007/s11265-015-1070-9

[29] A. Y. Dogan *et al.*, "Power/performance exploration of single-core and multi-core processor approaches for biomedical signal processing," in *Proc. Int. Workshop Power Timing Modeling, Optim. Simulation*, 2011, pp. 102–111.

[30] R. G. Dreslinski, B. Zhai, T. Mudge, D. Blaauw, and D. Sylvester,, "An energy efficient parallel architecture using near threshold operation," in *Proc. 16th Int. Conf. Parallel Architecture Compil. Techn.*, Sep. 2007, pp. 175–188.

[31] A. M. Kulkarni, H. Homayoun, and T. Mohsenin, "A parallel and reconfigurable architecture for efficient OMP compressive sensing reconstruction," in *Proc. 24th Ed. Great Lakes Symp. VLSI (GLSVLSI)*. New York, NY, USA, 2014, pp. 299–304.

[32] A. Page, N. Attaran, C. Shea, H. Homayoun, and T. Mohsenin, "Low-power manycore accelerator for personalized biomedical applications," in *Proc. 26th Ed. Great Lakes Symp. VLSI (GLSVLSI)*. New York, NY, USA, 2016, pp. 63–68. [Online]. Available: http://doi.acm.org/10.1145/2902961.2902986

[33] M. K. Tavana, A. Kulkarni, A. Rahimi, T. Mohsenin, and H. Homayoun, "Energy-efficient mapping of biomedical applications on domain-specific accelerator under process variation," in *Proc. 2014 Int. Symp. Low Power Electron. Design (ISLPED)*, New York, NY, USA, 2014, pp. 275–278.

[34] J. Bisasky, H. Homayoun, F. Yazdani, and T. Mohsenin, "A 64-core platform for biomedical signal processing," in *Proc. 14th Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2013, pp. 368–372.

[35] B. Cope *et al.*, "Implementation of 2D convolution on FPGA, GPU and CPU," *Imperial College Report*, pp. 2–5, 2006.

[36] A. Kulkarni and T. Mohsenin, "Accelerating compressive sensing reconstruction OMP algorithm with CPU, GPU, FPGA and domain specific many-core," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 970–973.

[37] E. Fykse, "Performance comparison of GPU, DSP and FPGA implementations of image processing and computer vision algorithms in embedded systems," Ph.D. dissertation, Dept. Electron., Norwegian Univ. Sci. Technol., Trondheim, Norway, 2013.

[38] S. Asano, T. Maruyama, and Y. Yamaguchi, "Performance comparison of FPGA, GPU and CPU in image processing," in *Proc. Int. Conf. Field Program. Logic Appl.*, Aug. 2009, pp. 126–131.

[39] E. Krimer, R. Pawlowski, M. Erez, and P. Chiang, "Synctium: A near-threshold stream processor for energy-constrained parallel applications," *IEEE Comput. Archit. Lett.*, vol. 9, no. 1, pp. 21–24, Jan. 2010.

[40] S. Hanson *et al.*, "A low-voltage processor for sensing applications with picowatt standby mode," *IEEE J. Solid-State Circuits*, vol. 44, no. 4, pp. 1145–1155, Apr. 2009.

[41] Y. Pu, J. P. de Gyvez, H. Corporaal, and Y. Ha, "An ultra-low-energy multi-standard JPEG co-processor in 65 nm CMOS with sub/near threshold supply voltage," *IEEE J. Solid-State Circuits*, vol. 45, no. 3, pp. 668–680, Mar. 2010.

[42] J. Rosen, A. Andrei, P. Eles, and Z. Peng, "Bus access optimization for predictable implementation of real-time applications on multiprocessor systems-on-chip," in *Proc. 28th IEEE Int. Real-Time Syst. Symp. (RTSS)*, Dec. 2007, pp. 49–60.

[43] I. A. Khatib *et al.*, "A multiprocessor system-on-chip for real-time biomedical monitoring and analysis: Architectural design space exploration," in *Proc. 43rd Annu. Design Autom. Conf.*, 2006, pp. 125–130.

[44] H. Alemzadeh, M. U. Saleheen, Z. Jin, Z. Kalbarczyk, and R. K. Iyer, "RMED: A reconfigurable architecture for embedded medical monitoring," in *Proc. IEEE/NIH Life Sci. Syst. Appl. Workshop (LiSSA)*, Apr. 2011, pp. 112–115.

[45] F. Bouwens *et al.*, "A dual-core system solution for wearable health monitors," in *Proc. 21st Ed. Great Lakes Symp. Great Lakes Symp. (VLSI)*, 2011, pp. 379–382.

[46] H. Ghasemzadeh and R. Jafari, "Ultra low-power signal processing in wearable monitoring systems: A tiered screening architecture with optimal bit resolution," *ACM Trans. Embed. Comput. Syst.*, vol. 13, no. 1, pp. 9:1–9:23, Sep. 2013. [Online]. Available: http://doi.acm.org/10.1145/2501626.2501636

[47] J. Bisasky, D. Chandler, and T. Mohsenin, "A many-core platform implemented for multi-channel seizure detection," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2012, pp. 564–567.

[48] A. Page, D. Chandler, and T. Mohsenin, "An ultra low power feature extraction and classification system for wearable seizure detection," in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Sep. 2015, pp. 7111–7114.

[49] A. Kulkarni, Y. Pino, M. French, and T. Mohsenin, "Real-time anomaly detection framework for many-core router through machine-learning techniques," *J. Emerg. Technol. Comput.*, vol. 13, no. 1, pp. 10:1–10:22, Jun. 2016. [Online]. Available: http://doi.acm.org/10.1145/2827699

[50] A. Kulkarni, A. Jafari, C. Sagedy, and T. Mohsenin, "Sketching-based high-performance biomedical big data processing accelerator," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 1138–1141.

[51] J. James Darin Chandler and T. Mohsenin, "An efficient network on chip (NOC) for a parallel, low-power, low-area homogenous many-core DSP platform," M.S. thesis, Univ. Maryland, Baltimore County, Baltimore, MD, USA, 2012, p. 81.

[52] (Mar. 2017). *Cadence Design System*. [Online]. Available: http://www.cadence.com/

[53] A. Kulkarni, Y. Pino, and T. Mohsenin, "SVM-based real-time hardware trojan detection for many-core platform," in *Proc. 17th Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2016, pp. 362–367.

[54] A. Page and T. Mohsenin, "FPGA-based reduction techniques for efficient deep neural network deployment," in *Proc. IEEE 24th Annu. Int. Symp. Field-Programm. Custom Comput. Mach. (FCCM)*, May 2016, pp. 1–8.

[55] A. Kulkarni, Y. Pino, and T. Mohsenin, "Adaptive real-time trojan detection framework through machine learning," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, May 2016, pp. 120–123.

[56] A. Kulkarni, T. Abtahi, C. Shea, A. Kulkarni, and T. Mohsenin, "PACENet: Energy efficient acceleration for convolutional network on embedded platform," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017.

[57] A. Kulkarni, T. Abtahi, E. Smith, and T. Mohsenin, "Low energy sketching engines on many-core platform for big data acceleration," in *Proc. 26th Ed. Great Lakes Symp. VLSI (GLSVLSI)*, New York, NY, USA, 2016, pp. 57–62. [Online]. Available: http://doi.acm.org/10.1145/2902961.2902984

[58] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[59] A. Jafari, M. Ghovanloo, and T. Mohsenin, "An embedded FPGA accelerator for a stand-alone dual-mode assistive device," in *Proc. IEEE Biomed. Circuits Syst. (BIOCAS) Conf.*, Oct. 2017.

[60] F.-T. Sun, C. Kuo, H.-T. Cheng, S. Buthpitiya, P. Collins, and M. Griss, "Activity-aware mental stress detection using physiological sensors," in *Proc. Int. Conf. Mobile Comput., Appl., Serv.*, 2010, pp. 211–230.

[61] J. Choi, B. Ahmed, and R. Gutierrez-Osuna, "Development and evaluation of an ambulatory stress monitor based on wearable sensors," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 2, pp. 279–286, Mar. 2012.

[62] J. A. Healey and R. W. Picard, "Detecting stress during real-world driving tasks using physiological sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 2, pp. 156–166, Jun. 2005.

[63] Y. Deng, Z. Wu, C. H. Chu, and T. Yang, "Evaluating feature selection for stress identification," in *Proc. IEEE 13th Int. Conf. Inf. Reuse Integr. (IRI)*, Aug. 2012, pp. 584–591.

[64] N. Attaran, J. Brooks, and T. Mohsenin, "A low-power multi-physiological monitoring processor for stress detection," in *Proc. IEEE SENSORS*, Oct. 2016, pp. 1–3.

[65] D. Patton, "How good is real enough? 300 degree of virtual immersion," M.S. thesis, Dept. Psychol., Towson Univ., Towson, MD, USA, 2013.
[66] A. Jafari, N. Buswell, M. Ghovanloo, and T. Mohsenin, "A low power wearable stand-alone tongue drive system for people with severe disabilities," *IEEE Trans. Biomed. Circuits Syst.*, to be published.

Adwaya Kulkarni received the B.E. degree in electronics and communication from Visvesvaraya Technological University, Belgaum, India, in 2008, and the master's degree in system level integration from Heriot-Watt University, Edinburgh, U.K., in 2010. She is currently pursuing the master's degree in computer engineering with the University of Maryland at Baltimore, Baltimore, MA, USA.

She was a System on a Chip Design Verification and Validation Engineer with Tata Elxsi Pvt Ltd, Bangalore, India. She did an internship at Intel, San Jose, CA, USA, as a Product Development Engineer, and would be joining full time as a Product Development Engineer at Intel after finishing her masters. Her current research interests include implementing machine learning and convolutional neural network kernels on manycore architecture and designing domain-specific manycore accelerators for energy-efficient and real-time computing.

Adam Page received the B.S. degree in computer engineering and the B.A. degree in mathematics, and the Ph.D. degree in computer engineering from the University of Maryland at Baltimore, Baltimore, MA, USA, in 2012 and 2016, respectively.

He is currently a Senior Software Engineer with Samtec, Mechanicsburg, PA, USA. He is actively researching strategies to efficiently deploy deep learning algorithms and is also the Designer of SPARCNet, a field programmable gate array (FPGA)-based accelerator for efficient deployment of sparse convolutional neural networks. He has authored over ten papers in peer-reviewed conferences and journals including two invited papers and one best paper award. His current research interests include the advancement of intelligent systems in the low-power embedded space that leverages the state-of-the-art machine learning with efficient hardware optimization and implementation techniques, and targeting multiprocessor system-on-chips for embedded design that incorporates graphics processing unit and FPGA fabric.

Nasrin Attaran received the master's degree in computer engineering from the University of Maryland at Baltimore, Baltimore, MA, USA, in 2017.

Her current research interests include low-power wearable multisensor biomedical devices, and machine learning and digital signal processing algorithms to design and implement health monitoring applications.

Ali Jafari is currently pursuing the Ph.D. degree with the Computer Science and Electrical Engineering Department, University of Maryland at Baltimore, Baltimore, MA, USA.

His current research interests include low-power analog/mixed signal ASIC and field programmable gate array designs, hardware accelerators for deep neural networks and machine learning algorithms, electronic sensors design, hardware–software embedded systems design, and developing low-power wearable monitoring systems.

Maria Malik received the B.E. degree in computer engineering from the Center of Advanced Studies in Engineering, Islamabad, Pakistan, and the M.S. degree in computer engineering from George Washington University, Washington, DC, USA. She is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, George Mason University, Fairfax, VA, USA.

Her current research interests include the field of computer architecture with the focus of performance characterization and energy optimization of big data applications on the high-performance servers and low-power embedded servers, accelerating machine learning kernels, parallel programming languages, and parallel computing.

Houman Homayoun received the B.S. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2003, the M.S. degree in computer engineering from the University of Victoria, Victoria, BC, Canada, in 2005, and the Ph.D. degree from the Department of Computer Science, University of California at Irvine, Irvine, CA, USA, in 2010,

He was with the University of California at San Diego, La Jolla, CA, USA, as a National Science Foundation Computing Innovation Fellow awarded by the Computing Research Association and the Computing Community Consortium. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA, USA, where he holds a joint appointment with the Department of Computer Science. He is also the Director of the George Mason University Green Computing and Heterogeneous Architectures Laboratory. He is also leading a number of research projects, including the design of next generation heterogeneous multicore accelerator for big data processing, nonvolatile STT logic, heterogeneous accelerator platforms for wearable biomedical computing, and logical vanishable design to enhance hardware security, which are all funded by the National Science Foundation, General Motors Company, and Defense Advanced Research Projects Agency.

Tinoosh Mohsenin received the M.S. degree in electrical and computer engineering from Rice University, Houston, TX, USA, in 2004, and the Ph.D. degree in electrical and computer engineering from the University of California at Davis, Davis, CA, USA, in 2010.

She is currently an Assistant Professor with the Department of Computer Science and Electrical Engineering, University of Maryland at Baltimore, Baltimore, MA, USA, where she directs the Energy Efficient High Performance Computing Lab. She also leads a number of research projects, including the design of next generation wearable biomedical processors, hardware accelerators for deep learning and convolutional neural networks, real-time brain signal artifact removal, and processing for brain computing interface and assistive devices, which are all funded by the National Science Foundation, Army Research Lab, Boeing, and Xilinx. She has over 60 peer-reviewed journal and conference publications. Her current research interests include the development of highly accurate high-performance processors for machine learning, knowledge extraction, and data sparsification and recovery that consume as little energy as possible.

Dr. Mohsenin has served as a Technical Program Committee Member of the International Solid-State Circuits Student Research, the IEEE Biomedical Circuits and Systems, the IEEE Circuits and Systems, and the International Symposium on Quality Electronic Design. She serves as a Secretary of the IEEE P1890 WG on Error Correction Coding for Non-Volatile Memories. She has served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I. She serves as an Associate Editor of the IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS.