An Energy Efficient Programmable Manycore Accelerator for Personalized Biomedical Applications

Adam Page[1], Adwaya Kulkarni[1], Nasrin Attaran[1], Ali Jafari[1], Maria Malik[2], Houman Homayoun[2], and Tinoosh Mohsenin[1]

[1]Department of Computer Science & Electrical Engineering, University of Maryland, Baltimore County
[2]Electrical and Computer Engineering Department, George Mason University

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 and machine learning kernels. In response to these requirements, in this paper, a tiny, energy-efficient and domain-specific manycore accelerator referred to as Power Efficient Nano Clusters (PENC) is proposed to map and execute the kernels of these applications. Simulation results show that PENC is able to reduce energy consumption by up to 80% and 25% for DSP and machine learning kernels, respectively when optimally parallelized. In addition, we fully implemented three compute-intensive personalized biomedical applications, namely multi-channel seizure detection, multi-physiological stress detection and stand-alone tongue drive system (sTDS), to evaluate the proposed manycore performance relative to commodity embedded CPU, GPU and FPGA based implementations. For these three case studies, the energy consumption and performance of the proposed PENC manycore when acting as an accelerator along with an Intel Atom processor as a host, is compared with existing commercial off-the-shelf general purpose, customizable, and programmable embedded platforms including Intel Atom, Xilinx Artix-7 FPGA, and NVIDIA TK1 ARM-A15 and K1 GPU SoC. For these applications, the PENC manycore is able to significantly improve throughput and energy efficiency by up to 1,872x and 276x, respectively. For the most computational intensive application of seizure detection, the PENC manycore is able to achieve a throughput of 15.22 GOPs which is a 14x improvement in throughput over custom FPGA solution. For stress detection, PENC achieves a throughput of 21.36 GOPs and energy efficiency of 4.23 GOP/J which is 14.87x and 2.28x better over FPGA implementation respectively. For sTDS application, the PENC improves a throughput by 5.45x and energy efficiency by 2.37x over FPGA implementation.

***Index Terms*—Personalized biomedical applications, low power manycore accelerator, stress detection, seizure detection, tongue drive system**

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 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], [2], [3], [4]. 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 (MEMS) sensors and specialized Analog-Front-End (AFEs) products targeted for physiological signals, such as Texas Instruments medical AFEs like 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 Blue-tooth Smart (17.9mA RX, 18.2mA TX, 1uA sleep). Still, the relatively high amount of power required to transmit raw or even compressed data makes it essential to perform local onboard processing [5]. The enterprise of this work is on the computing platform to address the unique challenges and 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, are not the most efficient in terms of performance/watt to process compute intensive applications [6], [7], [8], [9], [10], [11]. To address the energy-efficiency challenge, heterogeneous architectures have emerged as a 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 micro-architectures (in-order or out-of-order) or instruction set architectures (Thumb and x86) with on-chip GPU or 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 ARM+CUDA embedded GPU), Qualcomm Snapdragon (CPU+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 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 trade-off between power, performance, and energy-efficiency. Although integration with GPUs have 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 [12], [13], [14], [15]. 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 digital signal processing (DSP) and machine learning (ML) kernels that exhibit task-level and data-level parallelism [16].** In response to all computing challenges of personalized biomedical applications discussed above, in this paper we propose a programmable energy-efficient, domain-specific accelerator named Power Efficient Nano Clusters (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 state-of-the-art commodity embedded computing platforms and real measurements makes the following major contributions:

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

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

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

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 [17], [10], [8], [9], [18], [18]. Integrating heterogeneous core in a multicore (such as ARM+MIPS), CPU+GPU, or heterogeneous CPU+GPU+FPGA, has been investigated in various studies. In more complex heterogeneous architecture, multi-core, GPU and even FPGA have been integrated to solve the ILP and TLP challenges. An example for FPGA+CPU+GPU is the Axel system [19] and Nvidia Tegra K1 and X1, that combines the benefits of the specialization of FPGA, the parallelism of GPU and the scalability of a multi-core architecture. These examples show that 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 [20]. Thus, these systems are impractical for resource constrained embedded portable environments. An example is the Nvidia Tegra that can reach up to 10W of power consumption that is not tolerable in resource-constrained biomedical embedded systems [21], [22]. A recent work has shown that each of multicore CPU and GPU based architectures offers a different power and performance trade-off for various biomedical applications [15]. Although easy to program, these processors have limited flexibility and parallelism. Therefore, a field programmable gate array (FPGA) is also explored which provides high flexibility but requires writing low-level logic. In [14], [23], [22] a high-level synthesis (HLS) tool was used to generate an accelerator for machine learning 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 work, 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 work have been carried on 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 [24], where 167 programmable processors having 16KB 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 FFT and video encoding. There has also been a recent work on using simpler cores for high performance computing applications in [25], they propose an ultra-low power platform built using tightly-coupled processing cores called PULP. This many-core platform consists of clusters of simpler 4 OpenRISC cores, having 64KB of L2 memory and 24 KB of TCDM (data memory) in 28 nm technology. This architecture is dedicated for computer vision applications such as smart surveillance cameras and autonomous micro-UAVs. A recent work has shown that KiloCore [16], 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 to cloud data centres 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 [26], [27], [28], [29]. This is mainly motivated by the inherent parallelism exist in biomedical applications with multi-channel signal analysis requirements, where multi-core architectures can bring significant energy-efficiency compared to 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 [30], [31], [32], [15]. Cope et al, have compared the implementation performance of image convolution on GPU, FPGA and CPU [33]; Fykse has compared image convolution processing on GPU and FPGA [34]. Asano et al have investigated the performance comparison of two dimensional filter on FPGA, GPU and CPU [35]; however none of this work has studied the trade-off between power and performance on state-of-the-art embedded heterogeneous platforms. In the context of customized processor design, there has been a number of research endeavors exploring a single core or a multi-core architecture design targeting specific biomedical applications. A massively parallel stream processor was introduced by Krimer et al. [36] which achieves 1 GOPS/W. An ultra low energy processor with low voltage operations was presented by Hanson et al. [37] 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 Yu et al. [38] for low energy mobile image processing using architecture level parallelism. In [39], Rosen et al. described a solution to implement predictable real-time applications on multiprocessors that uses a bus scheduling policy based on TDMA (Time Division Multiple Access). In their solution, processors are assigned time-slots to access the bus with static scheduling. Their proposed multicore architecture [40] is used for real-time biomedical monitoring and analysis system. Alemzadeh et al. [41] proposed a reconfigurable architecture for real-time assessment of individuals health status based on development of a patient-specific health index and online analysis of multi-parameter physiological signals. Bouwens et al. [42] proposed a Dual-Core system solution for wearable health monitors ECG R-peak detection application which consumes 65.38W. 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 multi-channel data streaming at various sampling rates. The analysis of these multiple streams requires a mix of data-level and task-level parallel computation [43], [44], [45], [46]. In addition, these applications often require a large number of digital signal processing (DSP) and machine learning (ML) techniques. DSP is often used to extract useful representations of the input data while machine learning is needed to perform automated classification for diagnostic and detection purposes. In this paper, Fig 1 shows block diagram of the seizure detection application [1], [47]. 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 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 (FIR), windowing, Fourier transform (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 multi-channel seizure detection, multi-physiological stress detection and Stand-alone tongue drive system (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.

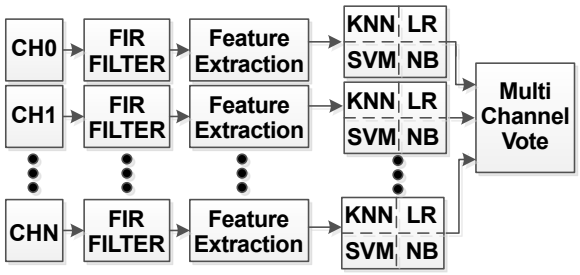


Fig. 1. Block diagram of a multi-channel seizure detection application containing feature extraction, ML classifier, multi-channel vote, and IO interface. The application highlights heavy use of DSP and ML kernels in addition to data-level and task-level parallelism.

III. Power Efficient Nano Clusters (PENC) ManyCore

A. PENC Manycore Overview and Key Features

PENC manycore accelerator is a homogeneous multiple instruction, multiple data (MIMD) architecture that consists of in-order tiny processors with a 6 stage pipeline, a RISC-like DSP instruction set and a Harvard Architecture model [46], [45], [44], [48], [49]. The core operates on a 16-bit datapath 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 intra-cluster communication directly via a bus and inter-cluster 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 was synthesized and fully placed and routed in a 65 nm CMOS technology using Cadence SoC Encounter and 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. PENC manycore architecture has 3 light-weight processing cores and a shared memory in a single cluster. **Our initial manycore architecture design had 4 processing cores and a hierarchical router within a cluster, which was ideal for DSP kernels for minimal data storage and localized processing [50]. 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 4 core implementation with 3 cores and a shared SRAM memory of 3K words and low latency bus-based architecture for inter-cluster communications, while maintaining the efficiency of low area and power consumption. Our initial results showed that 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 is packetized and sent to its output FIFO through a bus. When data is 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 intra-cluster communication. This includes a round-robin arbiter which chooses the next node to grant access based on 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

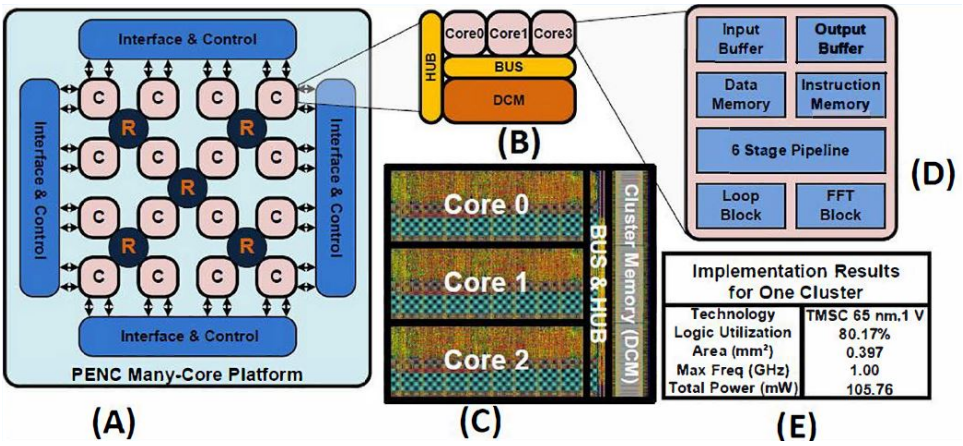


Fig. 2. (A) Power Efficient Nano Clusters (PENC), Manycore Architecture (B) Bus-based Cluster Architecture (C) Post-layout view of bus-based cluster implemented in 65nm, 1V TSMC CMOS technology (D) Block Diagram of core architecture (E) Post Layout implementation results of optimized bus-based cluster (consisting of 3 cores + bus + cluster Memory)

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 suite for machine learning 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 post layout 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 5 branching instructions removed, as they were redundant. This optimization managed to a 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 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 machine learning 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 PENC Manycore architecture. Fig 4 shows the post-layout implementation breakdown analysis of optimized PENC manycore comprising of 192 Processing cores, bus cluster, shared memory and router with Fig. 4A showing the area breakdown and Fig. 4B 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 post-layout report, the power results are obtained from the Encounter power analysis with careful

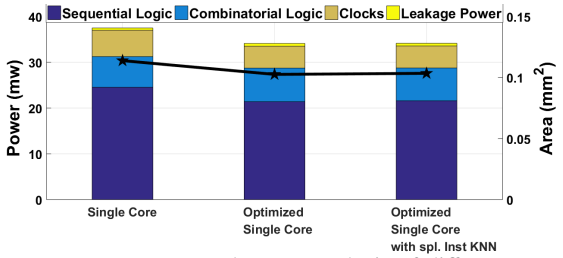


Fig. 3. Post-Layout area and power analysis of different customization of single processing core in PENC manycore architecture. Power is reported for 1GHz clk

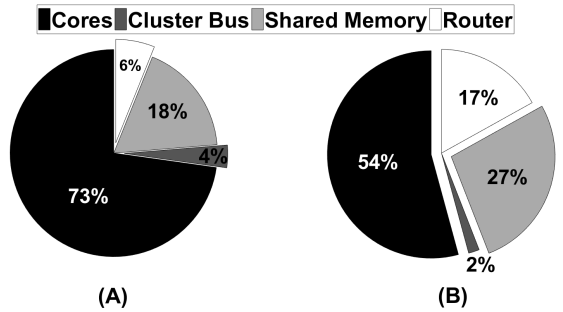


Fig. 4. Post layout implementation breakdown analysis of PENC manymore comprising of 192 processing cores, cluster bus, shared memory and router (A) Area Breakdown (B) Power Breakdown

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 [46], [45], ML kernels often require larger amounts of memory for their model data. This is addressed with the distributed cluster-level shared memory (DCM), that is interfaced to the bus. The shared memory within a cluster consists of 3 instances of SRAM cells of memory size 1024x16 bits 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-bits 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. **PENC architecture is ideally suited for personalized biomedical applications which require to compute a variety of multi-physiological signals in real-time within limited power budget. As previously shown in Table 1 and Figure 1, these biomedical applications process many physiological signals at different sampling rates. The processing of these parallel signals requires both digital signal processing (DSP) and machine learning (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, MAC, 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. Next section 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 post-layout 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 post-layout 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 on to the hardware design of the manycore platform which is in Verilog and simulated using Cadence ®NC-Verilog [51] as shown in Fig 5. The activity factor is then derived and is used by the Cadence ® [51] Encounter tool for accurate power estimation of application running on the post-layout VLSI hardware of the manycore. The manycore simulator reports statistics such as the number of cycles required for Arithmetic Logic Unit (ALU), branch, and communication instructions which are used for the throughput and energy analysis of the PENC manycore architecture.

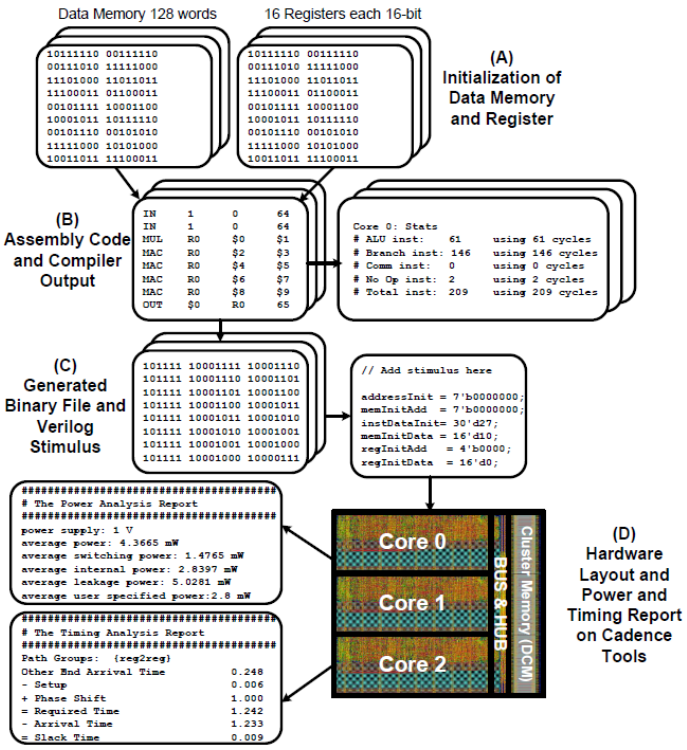


Fig. 5. Mapping of PENC manycore simulator and compiler flow high-level diagram and mapping the PENC hardware design using compiler.

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 digital signal processing kernels were mapped onto the PENC manycore. The DSP kernels include fast Fourier transform (FFT), finite impulse response filter (FIR), orthogonal matching pursuit (OMP), dot-product operation (DOT), and Convolutional Neural Network (CNN). In our previous work, we have designed specialized hardware for these kernels [1], [48], [49], [52]. 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

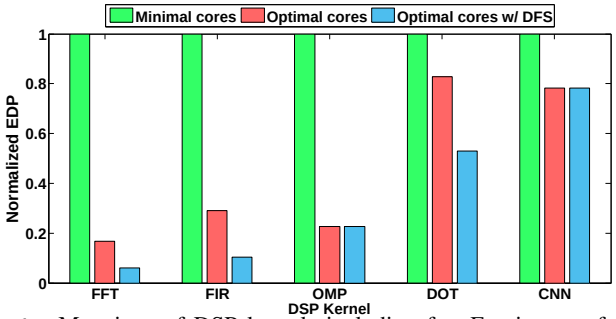


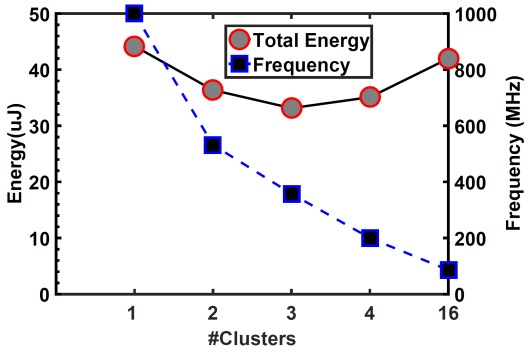
Fig. 6. Mappings of DSP kernels including fast Fourier transform (FFT), FIR filter (FIR), Orthogonal Matching Pursuit (OMP), dot-product (DOT), and Convolutional Neural Network (CNN). First mapping uses the minimum cores needed. The second mapping utilizes optimal cores to leverage parallelism.The third is the same as second but with dynamic frequency scaling (DFS).

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 mapping for the five DSP kernels with their corresponding energy-delay product (EDP). The plot shows that the manycore can efficiently parallelize all of the kernels and is able to achieve an energy delay product reduction of up to 10x. 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 same as the mapping when optimal amount of cores are used. OMP maps [53] sketching of 384x384 size image and CNN maps the convolution layers of LENET-5 [54].

2) ML Kernel Mapping: Many digital signal processing kernels require very little static and dynamic memory. For example, an 128-point FFT requires around 512 words of memory assuming twiddle factors are pre-computed and the input is complex. On the other hand, many machine learning 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 17000 data). This is accommodated for by having cluster-level shared memory accessible through the cluster's bus. The mapping of a ML kernel onto the manycore is performed similar to the mappings of the DSP kernels. The KNN algorithm with 512 model data is mapped using between 1 and 16 clusters. The results are depicted in Fig. 7. This figure shows different mappings of feature 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 3 clusters which was able to reduce energy by 25% and execution time by 63% compared to 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 are shown in Fig 8 as well.

IV. Case Studies

In this paper, we explore three applications namely Stress Detection, Seizure Detection and Tongue Drive System to address the requirements for personalized biomedical applications that compute a variety of multi-physiological signals in real-time within limited power budget: Seizure detection (Fig 1) exploits multi-channel parallel signal processing for 22 to 64 EEG channels. Stress detection in Fig 12 exploits multi-physiological signal processing for heart rate, accelerometer, respiration, and galvanic skin conductance. Tongue Drive System (TDS) in Fig 15 exploits 3D magnetic sensor data through tongue movement for 12 channels. For these three applications, processing of parallel data streams which The processing of these signals requires both digital signal processing (DSP) and machine learning (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 behaviour 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 standalone accelerator similar to the implementation for Artix FPGA and microcontroller as will be discussed in Table III and Figure 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.



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.

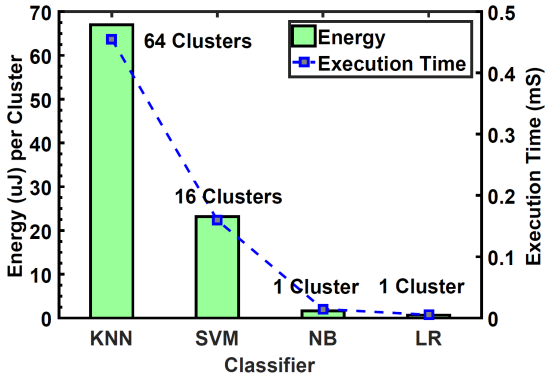


Fig. 8. Energy per cluster and execution time with mappings of ML kernels including KNN-3 (17,500 traning samples), linear SVM (4937 support vectors), NB and LR on PENC manycore platform.

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 4th most common

neurological disorder in the US 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, psycho-social, 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 (SUDEP).

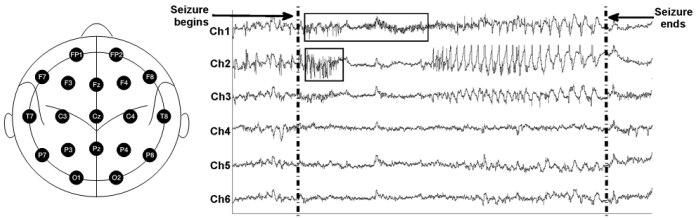


Fig. 9. Seizure detection overview showing EEG electrode placement on scalp with waveform from 6 channels highlighting seizure onset.

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], [47], [5]. 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 4 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 5 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 multi-channel voting scheme to determine the final classification.

For our study, the windows consist of 256 samples (1 second) 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. 10. The mapping highlights the parallelism that exists both between the EEG channels and within the KNN classifier kernel. For different machine learning classifier, the task graph for seizure detection system is depicted in Fig 11.

B. Multi-physiological Stress Detection Application

Stress is a physiological response to the mental, emotional, and physical challenges that everyone encounters in their daily life [55]. There are strong links between stress and overall health, concentration and ability to perform tasks. Predicting levels of stress using multi-modal physiological sensors has been an active research topic in recent years [55], [56], [57], [58]. These sensors usually include electrocardiogram (ECG), electromyogram (EMG), Galvanic Skin Response (GSR), respiration (Resp) and accelerometer.

In our previous work, a multi-modal stress detection hardware system was implemented to detect the level of stress by analyzing multiple physiological signals including heart rate and accelerometer [59]. The processing consists of 3 main

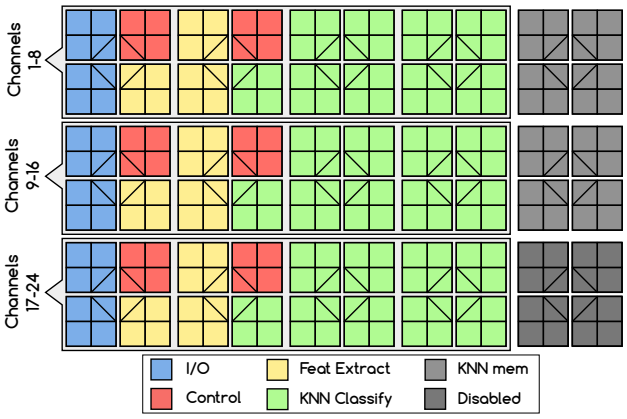


Fig. 10. Mapping of KNN-based seizure detection application onto PENC manycore.

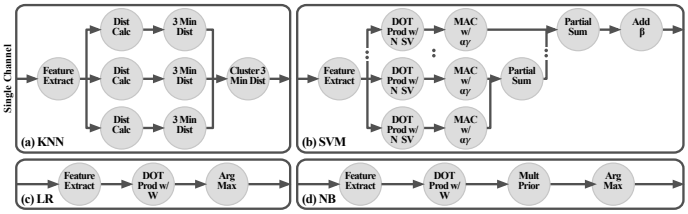


Fig. 11. Task graphs of each variation of the seizure detection application, when using KNN, SVM, LR and NB ML algorithms. The graphs highlight the task-level parallelism and interconnect between features extraction and classification stage for each channel.

stages as depicted in Fig. 12. The physiological sensor data is 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 4 temporal features (one feature for heart rate and 3 features for accelerometer). Each feature sample is then classified using 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 [60]. Our explicit goal is to determine an algorithmic model from which the level of stress could be determined using multi physiological signals. For our study, the windows consist of 6-second samples containing both HR and accelerometer signals with 50% overlapping windows. Fig 13 illustrates the simulation environment from which data has 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. Stand-alone Tongue Drive System

The Stand-alone Tongue Drive System (sTDS) developed at GTBIONICS Lab in Georgia Institute of Technology and

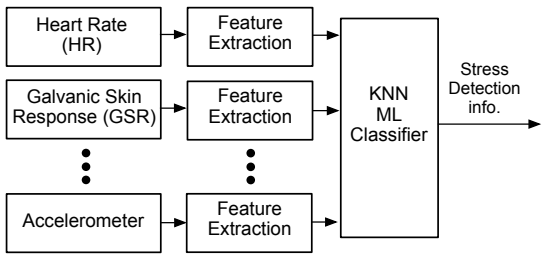


Fig. 12. Block diagram of a multi-physiological stress detection system containing data acquisition by sensors, feature extraction, and machine learning classifier to generate result.

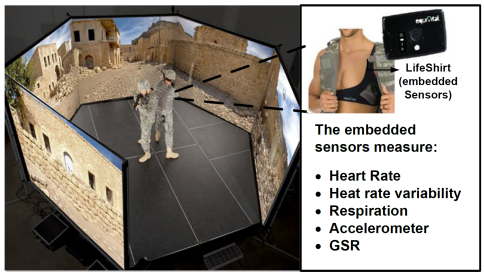


Fig. 13. 300 degree simulator to collect the multi-physiological data during different levels of stress using the embedded sensors in wearable lifeshirt [60].

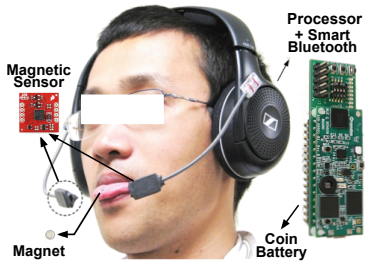


Fig. 14. sTDS prototype placed on a headset which includes a custom low power processor, four magnetic sensors, a low energy bluetooth and a battery.

UMBC 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 [61], [2]. Fig. 14 shows the sTDS which 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. 15 depicts functional block diagram of the sTDS. The sensory input consists of four 3D 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 is sent through an Earth Magnetic Interference (EMI) attenuation block which utilizes regression analysis to remove noise artifacts as well as Earth magnetic field. Once this stage is complete, the data is then all fed into a machine learning classifier stage that makes a final classification based on these samples.The use of temporal and spatial components helps to dramatically reduce error. Logistic Regression (LR) is implemented as the machine learning classifier and the detection accuracy of LR is 96.6%. As shown in the block diagram in Fig. 15, similar to the seizure application, task-level and datalevel parallelism exist. Task-level parallelism exists in the data acquisition, EMI and ML classifier modules. These analysis results show that LR is the best candidate for the proposed sTDS because not only could it achieve similar accuracy compared to another algorithm, but it also consumes lower energy consumption and needs smaller memory for saving the calibration coefficients. Hence, LR is chosen as ML classifier and it is implemented on different hardware platforms. As a case study, the sTDS was implemented on different platforms

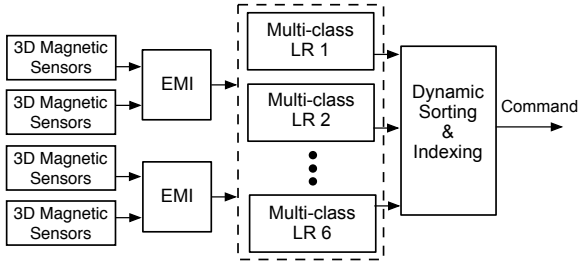


Fig. 15. Block diagram of the sTDS containing external magnetic interference (EMI) cancellation kernel and multi-class machine learning classifier where Logistic Regression (LR) is used.

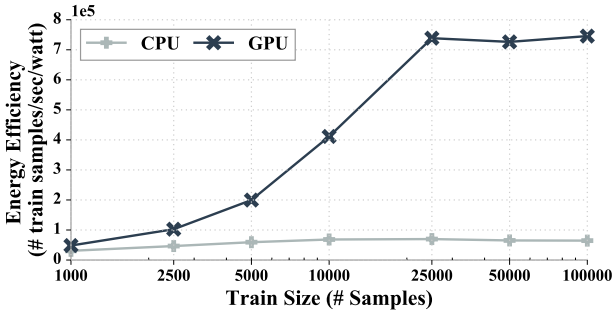


Fig. 16. Comparison of KNN kernel on Jetson TK1 when using quad-core ARM-A15 and embedded K1 GPU. Utilizing the GPU enables significantly improving energy efficiency by up to 11x.

including FPGA (Xilinx Artix-7 XA7A15T), Microcontroller (ARM Cortex-M4) and PENC many-core, and the results will be discussed in Section VI.

V. Off-the-shelf Platforms & Experimental Setup

To better gauge the performance of PENC many-core 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 x86-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 17. 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 details of the targeted platforms including the board capabilities, processors included, and application mappings.

Fig. 17. Experimental setup to obtain power and execution time measurements of NVIDIA Jetson TK1 (as well as Intel Edison) platforms using TI INA219 and Arduino.

A. NVIDIA Jetson TK1

NVIDIA's Jetson TK1 is a System-on-Chip (SoC) combining the Kepler graphics processing unit (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 and one low power processor. Each ARM A15 CPU has a 32KB L1 data and instruction cache supporting 128-bit NEON™ general-purpose single instruction and SIMD instructions. All processors configuration have shared access to a 2MB 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 the GPU, we are able to achieve several orders of magnitude energy-efficiency improvement over the ARM CPU counterpart. For example, Fig. 16 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 11x 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 Internet of Things (IOT). It contains an ultra low-power system on chip (SoC) with a dual-core Intel Atom processor (IA-32), 1 GB of DDR3, Wi-Fi, Bluetooth, and 4 GB eMMC memory running at a fixed clock of 500 MHz. The Intel Edison platform is used to obtain results by using Intel Atom processors standalone 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 x86 cores. This was done using SIMD optimizations performed both by the compiler and in the code. Further, parallelism was exploited by multi-threading 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 SSE 4.2. To enhance the SIMD further, Intel® Performance Primitives (IPP) 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.**

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

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

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 HDL. 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 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 standalone 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 x86-based CPU of Intel Edison. Table III provides results for all three applications including throughput, power, energy, and energy efficiency. In the table, results for all three applications are recorded when each platform is executing at its maximum clock frequency. However, for this class of personalized biomedical

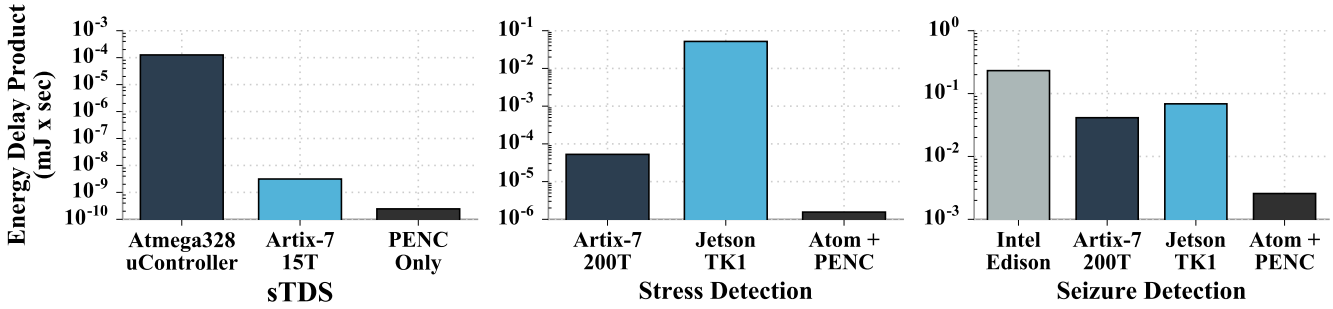


Fig. 18. **Comparison of energy-delay-product (EDP) for three case studies when implemented on several processor combinations including Atmega328 micro-controller, Artix-7 FPGA, Jetson TK1, and PENC manycore. The EDP is calculated as Energy/Throughput, where throughput is the number of decisions per second (i.e. inverse of the time taken to complete one classification).**

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**

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 MICRO-CONTROLLER, WHILE FOR STRESS DETECTION AND SEIZURE DETECTION APPLICATIONS, THE PENC ACCELERATOR RUNS WITH A HOST CPU (ATOM PROCESSOR) FOR DATA MARSHALING.

applications, the sampling frequency is relatively low in the range of 50 Hz to 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. 18 provides comparisons of manycore to COTS processor combinations in terms of energy-delay-product 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× sec.** 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 has the main disadvantage of long development time to design hardware-defined solution. Furthermore, the PENC manycore requires 13x, 34x, and 16x lower EDP compared to FPGA solution for sTDS, stress, and seizure detection, respectively.

In Fig 19, the processing combinations for all three applications are further evaluated in terms of energy efficiency versus

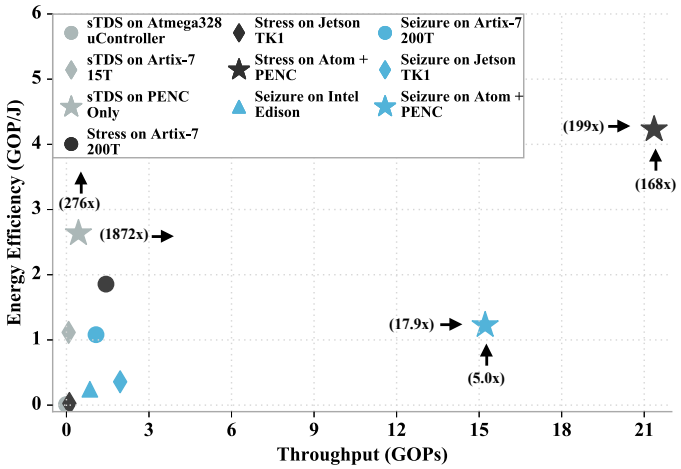


Fig. 19. Comparison of energy efficiency (GOP/J) versus throughput (GOPs) for all three case studies implemented on several processor combinations.

throughput. We utilize giga-operations-per-second (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 18x and throughput by 5x 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. Conclusions

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 digital signal processing and machine learning kernels. The software, as well as hardware implementations of machine learning personalized biomedical applications, are compared. For the choice of software, 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, dynamic frequency scaling, 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 13x, 34x, and 16x lower EDP compared to FPGA solution for sTDS, stress, and seizure detection, respectively. The PENC manycore was further compared to other commercial off-the-shelf platforms for three compute-intensive personalized biomedical applications, including stand-alone tongue drive system (TDS), 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 1,872x and 276x, 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 14x improvement in throughput over custom FPGA solution. For stress detection, the PENC achieves a throughput of 21.36 GOPs and energy efficiency of 4.23 GOP/J which improves the throughput by 14.87x and energy efficiency by 2.28x over FPGA implementation respectively. For sTDS the PENC improves the throughput by 5.45x and energy efficiency by 2.37x over FPGA implementation.

VIII. Acknowledgement

Authors would like to thank Amey Kulkarni, Colin Shea, Tahmid Abtahi and Abhilash Puranik for some preliminary results in this work. This research is based upon work supported by the National Science Foundation under Grant No. 1527151 and 1329829.

References

[1] A. Page, C. Sagedy *et al.*, "A flexible multichannel eeg feature extractor and classifier for seizure detection," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 62, no. 2, pp. 109–113, 2015.

[2] S. Viseh, M. Ghovanloo, and T. Mohsenin, "Towards an ultra low power on-board processor for tongue drive system," *Circuits and Systems II: IEEE Transactions on, accepted*, vol. 62, no. 2, pp. 174–178, Feb 2015.

[3] 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 journal of solid-state circuits*, vol. 48, no. 1, pp. 214–228, 2013.

[4] 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 Journal of Solid-State Circuits*, vol. 48, no. 7, pp. 1625–1637, 2013.

[5] 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 *IEEE Biomedical Circuits and Systems (Biocas) Conference*, Oct 2015.

[6] 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 *Computer Design (ICCD), 2015 33rd IEEE International Conference on*. IEEE, 2015, pp. 379–382.

[7] 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 *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 85–94.

[8] 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 *Proceedings of the 52nd Annual Design Automation Conference*. ACM, 2015, p. 151.

[9] V. Kontorinis, M. K. Tavana, M. H. Hajkazemi, D. M. Tullsen, and H. Homayoun, "Enabling dynamic heterogeneity through core-on-core stacking," in *Proceedings of the 51st Annual Design Automation Conference*. ACM, 2014, pp. 1–6.

[10] H. Homayoun, V. Kontorinis, A. Shayan, T.-W. Lin, and D. M. Tullsen, "Dynamically heterogeneous cores through 3d resource pooling," in *IEEE International Symposium on High-Performance Comp Architecture*. IEEE, 2012, pp. 1–12.

[11] A. Lukefahr, S. Padmanabha, R. Das, F. M. Sleiman, R. Dreslinski, T. F. Wenisch, and S. Mahlke, "Composite cores: Pushing heterogeneity into a core," in *Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE Computer Society, 2012, pp. 317–328.

[12] 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 *Field-Programmable Technology (FPT), 2012 International Conference on*. IEEE, 2012, pp. 329–334.

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

[14] K. Neshatpour, A. Koohi, F. Farahmand, R. Joshi, S. Rafatirad, A. Sasan, and H. Homayoun, "Big biomedical image processing hardware acceleration: A case study for k-means and image filtering," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2016.

[15] M. Malik, F. Farahmand, P. Otto, N. Akhlaghi, T. Mohsenin, S. Sikdar, and H. Homayoun, "Architecture exploration for energy-efficient embedded vision applications: From general purpose processor to domain specific accelerator," in *IEEE Computer Society Annual Symposium on VLSI, ISVLSI 2016, Pittsburgh, PA, USA, July 11-13, 2016*, 2016.

[16] B. Bohnenstiehl, A. Stillmaker, J. J. Pimentel, T. Andreas, B. Liu, A. T. Tran, E. Adeagbo, and B. M. Baas, "Kilocore: A 32-nm 1000-processor computational array," *IEEE Journal of Solid-State Circuits*, vol. PP, no. 99, pp. 1–12, 2017.

[17] 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 *Microarchitecture, 2003. MICRO-36. Proceedings. 36th Annual IEEE/ACM International Symposium on*. IEEE, 2003, pp. 81–92.

[18] K. Neshatpour, M. Malik, and H. Homayoun, "Accelerating machine learning kernel in hadoop using fpgas," in *Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on*. IEEE, 2015, pp. 1151–1154.

[19] K. H. Tsoi and W. Luk, "Axel: a heterogeneous cluster with fpgas and gpus," in *Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays*, 2010, pp. 115–124.

[20] X. Mei, L. S. Yung, K. Zhao, and X. Chu, "A measurement study of gpu dvfs on energy conservation," in *Proceedings of the Workshop on Power-Aware Computing and Systems*. ACM, 2013, p. 10.

[21] A. Kulkarni, C. Shea, T. Abtahi, and T. Mohsenin, "Low overhead cs-based heterogeneous framework for big data acceleration," *ACM Transactions on Embedded Computing Systems*, 2017.

[22] 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

[23] K. Neshatpour, M. Malik, M. A. Ghodrat, A. Sasan, and H. Homayoun, "Energy-efficient acceleration of big data analytics applications using fpgas," in *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 115–123.

[24] D. N. Truong, W. H. Cheng, T. Mohsenin, Z. Yu, A. T. Jacobson, G. Landge, M. J. Meeuwsen, C. Watnik, A. T. Tran, Z. Xiao *et al.*, "A 167-processor computational platform in 65 nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1130–1144, 2009.

[25] 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

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

[27] R. G. Dreslinkski *et al.*, "An energy efficient parallel architecture using near threshold operation," in *Proceedings of the 16th International Conference on Parallel Architecture and Compilation Techniques*. IEEE Computer Society, 2007, pp. 175–188.

[28] A. M. Kulkarni, H. Homayoun, and T. Mohsenin, "A parallel and reconfigurable architecture for efficient omp compressive sensing reconstruction," in *Proceedings of the 24th Edition of the Great Lakes Symposium on VLSI*, ser. GLSVLSI '14. New York, NY, USA: ACM, 2014, pp. 299–304.

[29] A. Page, N. Attaran, C. Shea, H. Homayoun, and T. Mohsenin, "Low-power manycore accelerator for personalized biomedical applications," in *Proceedings of the 26th Edition on Great Lakes Symposium on VLSI*, ser. GLSVLSI '16. New York, NY, USA: ACM, 2016, pp. 63–68. [Online]. Available: http://doi.acm.org/10.1145/2902961.2902986

[30] 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 *Low Power Electronics and Design (ISLPED), 2014 IEEE/ACM International Symposium on*. IEEE, 2014, pp. 275–278.

[31] A. Page, N. Attaran, C. Shea, H. Homayoun, and T. Mohsenin, "Low-power manycore accelerator for personalized biomedical applications," in *Proceedings of the 26th edition on Great Lakes Symposium on VLSI*. IEEE, 2016, pp. 275–278.

[32] J. Bisasky, H. Homayoun, F. Yazdani, and T. Mohsenin, "A 64-core platform for biomedical signal processing," in *Quality Electronic Design (ISQED), 2013 14th International Symposium on*. IEEE, 2013, pp. 368–372.

[33] B. Cope *et al.*, "Implementation of 2d convolution on fpga, gpu and cpu," *Imperial College Technical Report*, pp. 2–5, 2006.

[34] E. Fykse, "Performance comparison of gpu, dsp and fpga implementations of image processing and computer vision algorithms in embedded systems," 2013.

[35] S. Asano *et al.*, "Performance comparison of fpga, gpu and cpu in image processing," in *2009 international conference on field programmable logic and applications*. IEEE, 2009, pp. 126–131.

[36] E. Krimer, R. Pawlowski, M. Erez, and P. Chiang, "Synctium: a near-threshold stream processor for energy-constrained parallel applications," *IEEE Computer Architecture Letters*, vol. 9, no. 1, pp. 21–24, 2010.

[37] S. Hanson, M. Seok, Y.-S. Lin, Z. Foo, D. Kim, Y. Lee, N. Liu, D. Sylvester, and D. Blaauw, "A low-voltage processor for sensing applications with picowatt standby mode," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1145–1155, 2009.

[38] Y. Pu, P. de Gyvez, H. Corporaal, Y. Ha *et al.*, "An ultra-low-energy multi-standard jpeg co-processor in 65 nm cmos with sub/near threshold supply voltage," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 3, pp. 668–680, 2010.

[39] 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 *Real-Time Systems Symposium, 2007. RTSS 2007. 28th IEEE International*. IEEE, 2007, pp. 49–60.

[40] I. Al Khatib, F. Poletti, D. Bertozzi, L. Benini, M. Bechara, H. Khalifeh, A. Jantsch, and R. Nabiev, "A multiprocessor system-on-chip for real-time biomedical monitoring and analysis: architectural design space exploration," in *Proceedings of the 43rd annual Design Automation Conference*. ACM, 2006, pp. 125–130.

[41] H. Alemzadeh, M. U. Saleheen, Z. Jin, Z. Kalbarczyk, and R. K. Iyer, "Rmed: A reconfigurable architecture for embedded medical monitoring," in *Life Science Systems and Applications Workshop (LiSSA), 2011 IEEE/NIH*. IEEE, 2011, pp. 112–115.

[42] F. Bouwens, J. Huisken, H. De Groot, M. Bennebroek, A. Abbo, O. Santana, J. Van Meerbergen, and A. Fraboulet, "A dual-core system solution for wearable health monitors," in *Proceedings of the 21st edition of the great lakes symposium on Great lakes symposium on VLSI*. ACM, 2011, pp. 379–382.

[43] 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

[44] M. Khavari Tavana, A. Kulkarni *et al.*, "Energy-efficient mapping of biomedical applications on domain-specific accelerator under process variation," in *Proceedings of the 2014 International Symposium on Low Power Electronics and Design*, ser. ISLPED '14. New York, NY, USA: ACM, 2014, pp. 275–278.

[45] J. Bisasky, H. Homayoun *et al.*, "A 64-core platform for biomedical signal processing," in *Quality Electronic Design (ISQED), 2013 14th International Symposium on*, March 2013, pp. 368–372.

[46] J. Bisasky, D. Chandler, and T. Mohsenin, "A many-core platform implemented for multi-channel seizure detection," in *Circuits and Systems (ISCAS), IEEE International Symposium on*, May 2012, pp. 564–567.

[47] A. Page, S. Pramod *et al.*, "An ultra low power feature extraction and classification system for wearable seizure detection," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, Sept 2015.

[48] A. Kulkarni, Y. Pino, M. French, and T. Mohsenin, "Real-time anomaly detection framework for many-core router through machine-learning techniques," *Journal on Emerging Technologies in Computing (JETC)*, vol. 13, no. 1, pp. 10:1–10:22, Jun. 2016. [Online]. Available: http://doi.acm.org/10.1145/2827699

[49] A. Kulkarni, A. Jafari, C. Sagedy, and T. Mohsenin, "Sketching-based high-performance biomedical big data processing accelerator," in *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*, 2016, pp. 1138–1141.

[50] 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," *Master Thesis-University of Maryland, Baltimore County*, p. 81, 2012.

[51] "Cadence design system," *http://www.cadence.com/*, March 2017.

[52] A. Page and T. Mohsenin, "Fpga-based reduction techniques for efficient deep neural network deployment," in *Field-Programmable Custom Computing Machines (FCCM), 2016 IEEE 24th Annual International Symposium on*, 2016, pp. 1–8.

[53] A. Kulkarni, T. Abtahi, E. Smith, and T. Mohsenin, "Low energy sketching engines on many-core platform for big data acceleration," in *Proceedings of the 26th Edition on Great Lakes Symposium on VLSI*, ser. GLSVLSI '16. New York, NY, USA: ACM, 2016, pp. 57–62. [Online]. Available: http://doi.acm.org/10.1145/2902961.2902984

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

[55] F.-T. Sun, C. Kuo, H.-T. Cheng, S. Buthpitiya, P. Collins, and M. Griss, "Activity-aware mental stress detection using physiological sensors," in *International Conference on Mobile Computing, Applications, and Services*. Springer, 2010, pp. 211–230.

[56] J. Choi, B. Ahmed, and R. Gutierrez-Osuna, "Development and evaluation of an ambulatory stress monitor based on wearable sensors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 2, pp. 279–286, March 2012.

[57] J. A. Healey and R. W. Picard, "Detecting stress during real-world driving tasks using physiological sensors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 2, pp. 156–166, June 2005.

[58] Y. Deng, Z. Wu, C. H. Chu, and T. Yang, "Evaluating feature selection for stress identification," in *Information Reuse and Integration (IRI), 2012 IEEE 13th International Conference on*, Aug 2012, pp. 584–591.

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

[60] D. Patton, "How good is real enough? 300 degree of virtual immersion," *Masters Thesis, Towson University Department of Psychology*, 2013.

[61] M. Sahadat, A. Alreja, P. Srikrishnan, and M. Ghovanloo, "A multi-modal human computer interface combining head movement, speech and tongue motion for people with severe disabilities," in *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE*, 2015, pp. 1–4.

Adam Page is a PhD candidate in Computer Engineering at the University of Maryland, Baltimore County (UMBC). His main research focus is on the advancement of intelligent systems in the low-power embedded space that leverages state-of-the-art machine learning with efficient hardware optimization and implementation techniques. In particular, he is interested in targeting multiprocessor system-on-chips (MPSoC) for embedded design that incorporates GPU and FPGA fabric. He is actively researching strategies to efficiently deploy deep learning algorithms and is the designer of SPARCNet, an FPGA-based accelerator for efficient deployment of sparse convolutional neural networks. During his academic career, Adam has published over 10 papers in peer reviewed conferences and journals including two invited papers and one best paper award. He received his B.S. degree in Computer Engineering and B.A. degree in Mathematics from the University of Maryland, Baltimore County in 2012.

Adwaya Kulkarni did her Masters in System Level Integration from Heriot-Watt University, Edinburgh, Scotland (U.K) in 2010 and B.E in Electronics and Communication from Visvesvaraya Technological University India in 2008. She worked for 4years as SoC Design Verification and Validation Engineer with Tata Elxsi Pvt Ltd, India . She is currently working towards the Ph.D. degree in Computer Engineering at the University of Maryland, Baltimore County. Her research interest include implementing machine learning kernels on manycore architecture, designing domain-specific manycore accelerators for energy efficient and real-time computing.

Nasrin Attaran is currently working toward her Masters degree in Computer Science and Electrical Engineering department, at University of Maryland Baltimore County. Her research interests include low power wearable multi-sensor biomedical devices as well as machine learning and digital signal processing algorithms to design and implement health monitoring applications.

Ali Jafari is currently working toward his Ph.D. degree in Computer Science and Electrical Engineering department, at University of Maryland Baltimore County. His research interests include Low power Analog/Mixed signal ASIC and FPGA design of digital signal processing algorithms, electronic sensors design and low power hardware-software embedded system design.

Maria Mallik is currently working towards the Ph.D. degree in Electrical and Computer Engineering department, at George Mason University, VA. She has received the M.S. degree in Computer Engineering from the George Washington University, DC and B.E. degree in Computer Engineering from the Center of Advanced Studies in Engineering, Pakistan. Her research interests are in 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 is an Assistant Professor of the Department of Electrical and Computer Engineering at George Mason University. He also holds a joint appointment with the Department of Computer Science. He is the director of GMUs Green Computing and Heterogeneous Architectures (GOAL) Laboratory. Prior to joining George Mason University, he spent two years at the University of California, San Diego, as National Science Foundation Computing Innovation (CI) Fellow awarded by the Computing Research Association (CRA) and the Computing Community Consortium (CCC). Houman is currently leading a number of research projects, including the design of next generation heterogeneous multicore accelerator for big data processing, non-volatile STT logic, heterogeneous accelerator platforms for wearable biomedical computing, and logical vanishable design to enhance hardware security which are all funded by National Science Foundation (NSF), General Motors Company (GM) and Defense Advanced Research Projects Agency (DARPA). Houman received his PhD degree from the Department of Computer Science at the University of California, Irvine in 2010, an MS degree in computer engineering in 2005 from University of Victoria, Canada and his BS degree in electrical engineering in 2003 from Sharif University of technology.

Tinoosh Mohsenin is an Assistant Professor in the Department of Computer Science and Electrical Engineering at University of Maryland Baltimore County, where she directs Energy Efficient High Performance Computing (EEHPC) Lab. She received her PhD from University of California, Davis in 2010 and M.S. degree from Rice University in 2004, both in Electrical and Computer Engineering. Prof. Mohsenin's research focus is on 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. Prof. Mohsenin has over 60 peer-reviewed journal and conference publications. She currently 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 National Science Foundation (NSF), Army Research Lab (ARL), Boeing and Xilinx. She has served as associate editor in IEEE Transactions on Circuits and Systems-I (TCAS-I) and currently serves as an associate editor in the IEEE Transactions on Biomedical Circuits and Systems (TBioCAS). She has served as technical program committee member of the International Solid-State Circuits Student Research (ISSCC-SRP), IEEE Biomedical Circuits and Systems (BioCAS), IEEE Circuits and Systems (ISCAS) and International Symposium on Quality Electronic Design (ISQED). She also serves as secretary of IEEE P1890 WG on Error Correction Coding for Non-Volatile Memories.