A Low-power Microprocessor for Data-driven Analysis of Analytically-intractable Physiological Signals in Advanced Medical Sensors

Kyong Ho Lee and Naveen Verma *
Princeton University, Princeton, NJ 08544, USA
{kyonglee, nverma}@princeton.edu

Abstract

Data-driven methods based on machine learning enable powerful frameworks for analyzing complex physiological signals in medical-sensor applications; however, these methods are not well supported by traditional DSPs. A general-purpose microprocessor is presented in 130nm CMOS that integrates configurable accelerators, enabling low-energy hardware to support the broadest range of machine-learning frameworks reported to date. In addition to computational energy, memory limitations due to the high-order data-driven models are overcome by an embedded compression/decompression accelerator, which reduces the memory footprint by 4× with overhead <8%. Using six medical applications with real clinical data, overall energy savings of 3.1-497× are demonstrated with the accelerator-based architecture.

Introduction

There is great clinical interest in creating very-low-power implantable/wearable devices that can provide closed-loop modes of therapy, reactive disease management, and high-value decision support [1]. A critical need is the ability to analyze physiologically-complex signals available through low-power sensors (LFPs, EEGs, ECGs, etc.). The challenge is that no tractable analytical models generally exist for these signals or the underlying processes. However, powerful data-driven methods have emerged from the domain of machine learning, wherein physical sensor data is used to construct models and analyze signals. Unfortunately, these frameworks are not well supported by low-power processors, making their energy orders of magnitude higher than traditional DSP [2]. To overcome this, processors have recently integrated specialized hardware for specific frameworks (e.g., support-vector machines [2, 3]); however, the range of applications possible depends strongly on the modeling frameworks. We thus present a general-purpose processor for micro-power sensors that integrates configurable hardware accelerators for a broad range of machine-learning frameworks. This is achieved by decomposing the framework computations into structured hardware kernels which can be configurably combined. Beyond energy, memory requirements also pose strict limitations since the embedded data-driven models generally have no compact parametric representations. A specialized memory interface is thus integrated to provide low-overhead on-line compression/decompression. We demonstrate six medical-sensor applications [4]-[8] using real clinical data.

Architecture

Fig. 1 shows the hardware architecture to address the framework computations, memory requirements, and power management:

Framework computations. A 16b CPU provides flexible signal-feature extraction and top-level configuration control across applications. A configurable machine-leaning accelerator (MLA) enables low-energy support for data-driven modeling frameworks, including Gaussian-mixture models (GMM), hidden Markov models (HMM), support-vector-machine (SVM) regression/classification, active learning (AL), Kalman filtering, and principal component analysis (PCA); it also enables support for key feature-extraction kernels, such as FIR filtering, discrete wavelet transform, and discrete cosine transform. As illustrated, this is achieved through framework FSMs, for high-level control; kernel controllers, for configuring linear/non-linear arithmetic with canonical vector and matrix operands; and arithmetic engines, for enabling the underlying computations via flexible data-path hardware.

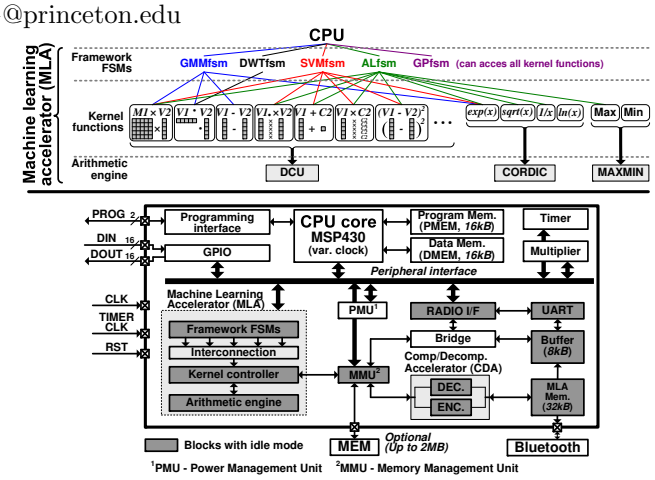


Fig. 1. Processor architecture maps machine-learning computations to structured hardware (a general-purpose FSM (GPfsm) enables direct CPU access to kernel functions to enhance programmability).

Memory requirements. To store framework models (which are based on data distributions) the MLA has a 32kB memory, which can be partitioned for multiple models and ensemble algorithms (e.g., adaboost, classifier trees). Since medical applications typically require high-order models to adequately analyze physiological signals [2], the compression/decompression accelerator (CDA) reduces the memory footprint by 4×. Its energy is minimized via simple encoding (ADPCM); though lossy, we show that the impact is negligible thanks to the resilience exhibited by probabilistic framework algorithms against noise in low-order bits [9].

Power management. Since the leakage power is low for logic and most applications focus on continuous medical signals, active energy is the primary concern. Active energy is managed by 16 clock domains. Idle-mode clock gating under software control is enabled by the power-management unit (PMU). The CPU clock is always on, but is adjustable via a clock divider to support variable-throughput requirements across applications. To minimize the overhead of network communication, a radio-interface provides background control of an off-chip Bluetooth module that interfaces with a gateway device for accessing wide-area networks.

Machine-Learning Accelerator (MLA)

Fig. 2 shows the MLA microarchitecture. In addition to specific framework FSMs, which configure and control the kernel controllers, a general-purpose FSM (GPfsm), configurable via a memory-mapped register, enables direct CPU access to the kernel controllers. This makes highly-programmable vector and matrix computations available to synthesis feature-extraction and modeling computations. The kernel controllers control the arithmetic engines to realize various linear-algebra and non-linear functions that occur frequently in the machine-learning frameworks. The arithmetic engines consist of a data-computation unit (DCU), a CORDIC, and a MAXMIN module. The DCU (described below) has a configurable data path to support the various kernels. The CORDIC supports the following non-linear transformations: exp(x), ln(x), 1/x, and sqrt(x). The MAXMIN supports both index search as well as min/max-value search in an array, and can be configured as a single 32b module or two parallel 16b modules.

*Partial support is provided by SRC and Qualcomm Innovation Fellowship; IC fabrication is provided by MOSIS.

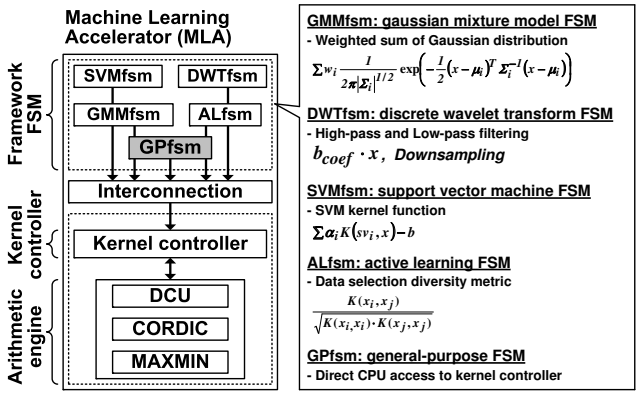


Fig. 2. MLA enables various machine-learning kernel functions.

Data Computation Unit (DCU)

Fig. 3 shows the DCU. Multiplications/additions/subtractions are structured in stages with selectable operands, enabling specialized data paths to be configured for the various kernels. The two-stage pipeline mitigates active energy due to propagating glitches. In order to maximize hardware usage, the operands from memory are provided through an interface (MMU) capable of 2× clock-boosting. From application profiling, a key challenge identified is the ability to optimize the fixed-point precision across the framework configurations. Two approaches overcome this. First, like the MAXMIN, the DPU can be configured as one 32b module or two parallel 16b modules wherein each input (in1 and in2) can be configured into two operands; configurability is achieved with low-overhead via the 33b adder shown, where the PAD bit allows separation/merging of two 16b paths. Second, a configurable barrel shifter enables in-line truncation to support any Q-format. For 16b mode, a reduce stage is used to combine the parallel outputs into a final result (e.g., for vector dot products, the outputs from sub-divided vectors are added together, as shown).

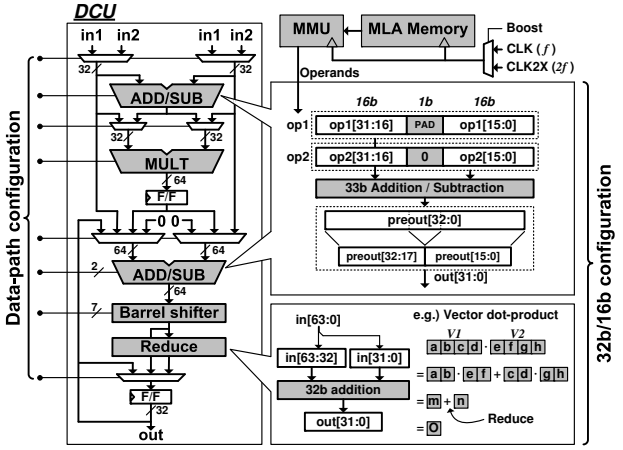


Fig. 3. DCU can be configured into data-paths specialized for the MLA kernels, and enables precision scaling via (1) configurability into 32/16b modules, and (2) in-line barrel shifting and truncation.

Compression Decompression Accelerator (CDA)

Fig. 4 shows the CDA. Various compression algorithms were investigated. ADPCM was selected due to its low computational complexity, enabling substantial compression (4×) with low energy overhead (<8%). Though ADPCM is lossy, various applications were profiled, validating negligible impact in performance; the results from a seizure detector are shown. The CDA consists of two parallel encoders and decoders to support the 32/16b data formats. The DCU can be configured for upto four operands, but the kernel computations involve up to three operands from memory; local buffers thus enable data from three vectors to be stored locally so that the iterations required for estimation in the ADPCM algorithm can proceed simultaneously. Compression then involves encoding deltas (with respect to the running estimates) using values/indices from steptable/indextable LUTs; decompression involves using the encoded bits in parallel to retrieve decodings from the LUTs.

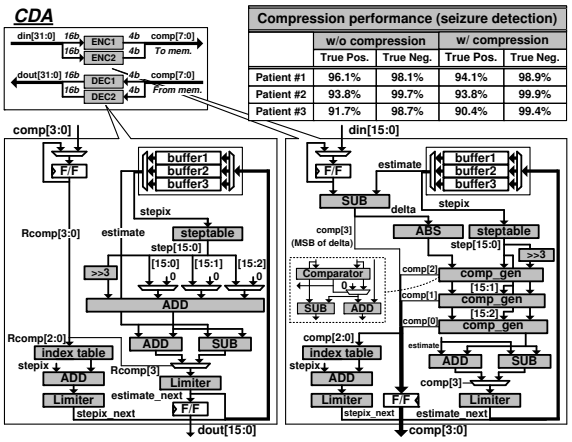
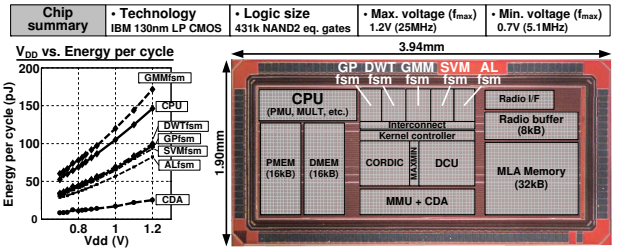


Fig. 4. CDA enables low-overhead on-line data compression (encoding) and decompression (decoding) using ADPCM; seizure-detector performance illustrates negligible impact of lossy ADPCM encoding.

Measurements and Demonstrations

The processor is implemented in 130nm CMOS from IBM. Fig. 5 shows a die photo, hardware measurements, and a comparison with the state of the art, illustrating the advances in hardware support achieved for machine-learning algorithms. The processor voltage scales from 1.2-0.7V for real-time operation of the applications tested. The applications, tested with real medical data, include the following: (1) ECG-morphology-based arrhythmia detection using SVM classification [4]; (2) ECG-wavelet-based arrhythmia detection using GMMs [5]; (3) patient-adaptive ECG-morphology-based arrhythmia detection using AL; (4) EEG-based seizure detection using SVM classification [6]; (5) EEG-based sleep-stage prediction using hybrid HMMs and GMMs [7]; and (6) spike/LFP-based motor-intention decoding for brain-machine interface using SVM regression [8]. From application profiling, energy savings of 3.1-497× are achieved compared to CPU-only implementations (the total power of the applications on the processor is denoted in the last column of the table).



Comparison with the state-of-the-art (Energy / clock cycle with V_{DD} = 0.7V†)

	This work	Lee [2]	Kwong [9]	Ashouei [10]	Sridhara [11]
CPU	52.3pJ	46.3pJ	46.0pJ	49.3pJ	65.2pJ
SVM	32.2pJ	53.8pJ	X	X	X
GMM	60.2pJ	X	X	X	X
DWT	34.3pJ	X	X	X	X
AL	28.7pJ	41.2pJ	X	X	X
DCU	51.8pJ	X	44.2pJ (FIR)	X	X
CORDIC	23.7pJ	X	19.6pJ	X	X
MAXMIN	22.4pJ	X	X	X	X
CDA	8.4pJ	X	X	X	X
Technology	130nm	130nm	130nm	90nm	130nm

†Min. allowed V_{DD} for real-time applications is 0.7V

Application profiling

	Energy†† (w/o acc.)	Energy†† (w/ acc.)	Energy savings	Power (w/ acc.)
1) Arrhythmia detection (ECG morphology base) [3]				
	12830µJ (@ 1.2V)	25.8µJ (@ 0.7V)	497.3x	25.8µW
2) Arrhythmia detection (ECG wavelet base) [4]				
	856.0µJ (@ 0.8V)	12.0µJ (@ 0.8V)	71.3x	12.0µW
3) ECG-based patient-adaptive arrhythmia detection				
	1283mJ (@ 1.2V)	3.06mJ (@ 0.7V)	419.3x	30.6µW
4) EEG-based seizure detection [5]				
	4053µJ (@ 1.2V)	93.6µJ (@ 0.8V)	43.3x	93.6µW
5) EEG-based sleep-stage classification [6]				
	2661mJ (@ 1.2V)	862mJ (1.2V)	3.1x	3.1mW
6) Spike/LFP-based motor decoding (BMI) [7]				
	580µJ (@ 0.8V)	6.7µJ (@ 0.79V)	86.6x	6.7µW

††Energy per output result

Fig. 5. Die photo, prototype measurements, and comparison.

References

[1] A. Csavoy, et al., *VLSI Symp.*, Jun. 2009, pp. 4-7.
[2] K. H. Lee and N. Verma, *ESSCIRC*, Sep. 2012, pp. 285-288.
[3] J. Park, et al., *JSSC*, Nov. 2012, pp. 2711-2723.
[4] P. de Chazal, et al., *IEEE Tran. Biomed. Eng.*, Jul. 2004.
[5] R. J. Martix, et al., *J. Med. Syst.*, 2012, pp. 677-688.
[6] A. Shoeb and J. Guttag, *ICML*, Jun. 2010.
[7] A. B. Rossow, et al., *BRC*, Jan. 2011, pp. 1-6.
[8] L. Shpigelman, et al., *NIPS*, 2008, pp. 1489-1496.
[9] V. K. Chippa, et al., *DAC*, Jun. 2010, pp. 555-560.
[10] J. Kwong and A. P. Chandrakasan, *ESSCIRC*, Sept. 2010, pp. 526-529.
[11] M. Ashouei, et al., *ISSCC*, Feb. 2011, pp. 332-334.
[12] S. R. Sridhara, et al., *VLSI Symp.*, Jun. 2010, pp. 15-16.