18.5 A Configurable 12-to-237KS/s 12.8mW Sparse-Approximation Engine for Mobile ExG Data Aggregation

Fengbo Ren, Dejan Marković

University of California, Los Angeles, CA

Compressive sensing (CS) is a promising solution for low-power on-body sensors for 24/7 wireless health monitoring [1]. In such an application, a mobile data aggregator performing real-time signal reconstruction is desired for timely prediction and proactive prevention. However, CS reconstruction requires solving a sparse approximation (SA) problem. Its high computational complexity makes software solvers, consuming 2–50W on CPUs, very energy inefficient for real-time processing. This paper presents a configurable SA engine in a 40nm CMOS technology for energy-efficient mobile data aggregation from compressively sampled biomedical signals. Using configurable architecture, a 100% utilization of computing resources is achieved. An efficient data-shuffling scheme is implemented to reduce memory leakage by 40%. At the minimum-energy point (MEP), the SA engine achieves a real-time throughput for reconstructing 61-to-237 channels of biomedical signals simultaneously with <1% of a mobile device's 2W power budget, which is 76–350× more energy-efficient than prior hardware designs.

Electrocardiography (ECG), electroencephalography (EEG), and electromyography (EMG) signals, collectively referred to as ExG, contain critical information about the human body status. ExG signals can span 3 orders of magnitude in amplitude (10μV–10mV) and frequency (0.1Hz–500Hz), and can have a large difference in sparsity depending on the subject's activity. For instance, a low and high activity can produce a signal sparsity (k/n) of <2% and >15%, respectively [2], where k is the signal sparsity level and n is the signal dimension (# of samples). As a result, prior chip implementations of SA solvers [3-4] that are optimized to handle limited dynamic range and fixed problem size are not suitable for the intended application. We designed a flexible SA architecture (Fig. 18.5.1), which uses a single-precision floating-point data format to achieve a large dynamic range. It can be configured to handle different problem settings on-the-fly, including signal and measurement dimensions (n and m), signal sparsity level (k), reconstruction basis (A) and error tolerance ($\in$).

The human body is expected to have a low activity on average, where ExG signals feature a high sparsity, especially when thresholding schemes are used [2]. This is where orthogonal matching pursuit (OMP) has a better complexity-accuracy trade-off than other SA algorithms [4]. For efficient mapping, 3 algorithm reformulation techniques have been introduced to further break down the OMP algorithm into 6 basic linear algebra (BLA) operations at each iteration [5], which are grouped into 3 tasks: atom searching (AS), least-squares solving (LS), and estimation update (EU). Taking advantage of the algorithm-architecture co-design, most hardware resources in the SA architecture are shared across different tasks.

The computing resources in the SA architecture include the vector and scalar processing cores (VC and SC), Fig. 18.5.2. To support flexible problem sizes with high energy efficiency, 128 processing elements (PEs) in the VC are coordinated through an interconnect block (IB). When the IB is enabled, PEs are configured to perform inner product. Otherwise, PEs can be configured to support element-wise addition, multiplication, multiply-accumulation, and vector-scalar product. A shift-register logic (SRL) unit is used in the feedback path of each PE to enable folded processing of long vectors. The SC supports scalar comparison, addition, accumulation, and division. Depending on the top-level data-path configuration, the SC can either post-process a selective result from the VC or process independent data in parallel. For efficient local memory access, a 1.2KB and 1.5KB cache is dedicated to each PE and the SC, respectively (Fig. 18.5.1). To facilitate data communication between the VC and SC in long delay lines, a 768B cache is shared between all the PEs and the SC. In addition, a core-level SRL unit is used to connect the SC with all the PEs. This feedback path effectively reduces the loop latency between VC and SC to 4–8 cycles, thereby greatly accelerating the iterative BLA operations such as forward and backward substitution (FS and BS). The SA engine uses FIFO interfaces to handle the flow control at the data inputs and outputs. A fixed-to-floating-point conversion interface is also available at the input to allow the processing of different signal representations. When executing the OMP algorithm, the VC is dynamically configured in a SIMD fashion. Similarly, the SC and data-paths are also dynamically configured to perform the BLA operations in different tasks. As a result, a 100% utilization of the computing resources is achieved, maximizing area efficiency (Fig. 18.5.2).

In the LS task, column and row access of a triangular matrix (L) is required for performing FS and BS, respectively, for realizing Cholesky factorization (CF). As accessing a row of L is equivalent to accessing a column of L^T, a straightforward memory-mapping scheme for PE caches is to store both L and L^T as a regular square matrix (Fig. 18.5.3). We refer to this scheme as a mirror mode, where columns of L and L^T can be accessed at the same address of each PE cache in an ascending and descending order, respectively. The mirror mode allows the square matrix to fold into 128×128 blocks to perform a larger-size (k>128) CF at the cost of doubled memory size. As memory leakage dominates the total power, a data-shuffling scheme that is more efficient in utilizing memory space is realized. In the shuffle mode, each row of L is stored in a shuffled order across adjacent PE caches. As a result, the rows and columns of L can be accessed at the same and at a different address of each PE-cache, respectively. To recover the data order correctly, a shuffler performing circular position-shift is implemented. Compared to the mirror-mode implementation, a 2× memory size reduction results in a 40% lower total power.

Reconstruction signal-to-noise ratios (RSNR) from 1-minute recordings of real ExG signals downloaded from the PhysioBank database are measured on the SA engine (Fig. 18.5.4). CS samples of ExG signals are encoded by random Bernoulli matrices, with a 5% overlapping window applied. In order to observe the raw signal sparsity, no thesholding scheme is applied. The best orthogonal basis for reconstructing ECG, EEG, and EMG are found to be the Haar discrete wavelet transform (DWT), the discrete cosine transform (DCT), and a DWT-DCT joint basis, respectively. It is also found that the RSNR performance is sensitive to $\in$. Dynamically configuring $\in$ to 3–5% of the energy of each CS sample results in the best RSNR performance. In general, using higher n for reconstruction improves RSNR at the cost of lower throughput and higher energy. To achieve RSNR>15dB with maximum throughput, the preferred n for reconstructing the chosen ECG, EMG, and EEG recordings is found to be 256, 128, and 512, respectively.

The power and operating frequency of the SA engine are measured at different supply voltages (Fig. 18.5.5). The MEP is found at V_{DD}=0.7V, which corresponds to a 12.8mW power and a 12.2MHz operating frequency. At the MEP, the SA engine achieves a throughput of 237, 123, and 66KS/s and an energy efficiency of 54, 104, and 194nJ/sample for reconstructing ECG, EMG, and EEG signals, respectively, at RSNR>15dB. This throughput corresponds to the simultaneous reconstruction of 237, 61, and 132 channels of ECG, EMG, and EEG data, respectively. At V_{DD}=1V, the maximum operating frequency of the SA engine is 25.3MHz. Compared to the MEP, a 2× higher throughput can be achieved at the cost of 3× lower energy efficiency.

The SA engine is compared to an Intel Core i7-4700MQ processor and prior chip implementations [3-4] of generic SA solvers (Fig. 18.5.6). Overall, the SA engine achieves a 2× higher throughput with up to 14,100× better energy efficiency for ExG signal reconstruction than the software solver running on the CPU. For high-sparsity signal reconstruction, the SA engine is 76–350× more energy efficient than prior hardware designs. With a <1% power budget of mobile devices, the 5.13mm^2 SA engine in 40nm CMOS (Fig. 18.5.7) can provide a 2–3× energy saving at CS-based sensor nodes, while providing timely feedback and bringing signal intelligence closer to the user.

Acknowledgements:
We thank Broadcom Foundation for funding support and TSMC University Program for chip fabrication.

References:
[1] F. Chen, *et al.*, "Design and Analysis of a Hardware-Efficient Compressed Sensing Architecture for Data Compression in Wireless Sensors," *IEEE J. Solid-State Circuits*, vol. 47, no. 3, pp. 744-756, 2012.
[2] A. Dixon, *et al.*, "Compressed Sensing System Considerations for ECG and EMG Wireless Biosensors," *IEEE Trans. on Biomedical Circuis and Syst.*, vol. 6, no. 2, pp. 156-166, 2012.
[3] P. Maechler, *et al.*, "VLSI Design of Approximate Message Passing for Signal Restoration and Compressive Sensing," *IEEE J. on Emerging and Selected Topics in Circuits and Syst.*, vol. 2, no. 3, pp. 579-590, 2012.
[4] P. Maechler, *et al.*, "Implementation of Greedy Algorithms for LTE Sparse Channel Estimation," *Asian Solid-State Circuits Conf*, pp. 400-405, 2010.
[5] F. Ren, *et al.*, "A Scalable and Parameterized VLSI Architecture for Compressive Sensing Sparse Approximation," *Electronics Letters*, vol. 49, no. 23, pp. 1440-1441, 2013.

 978-1-4799-6224-2/15/$31.00

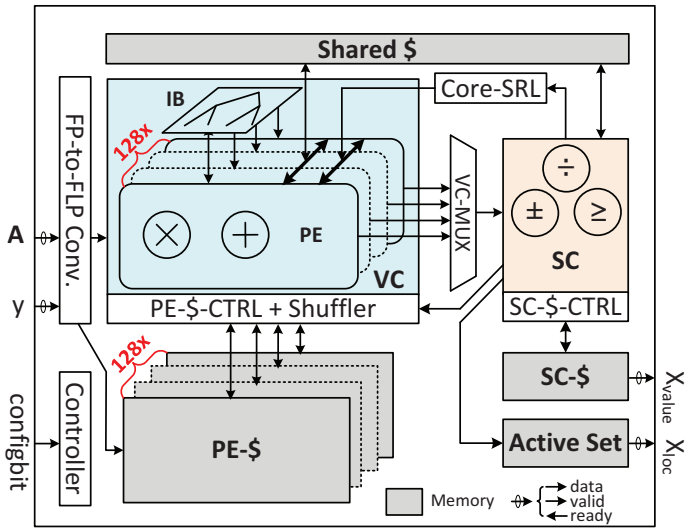
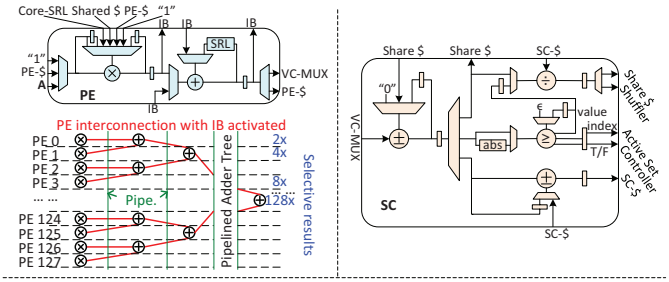


Figure 18.5.1: System architecture of the SA engine.



Task		PE	PE-SRL	IB	VC-MUX	Core-SRL	SC	PE-$	SC-$	Share $	Active Set
AS		Inner prod.	Off	On	Static	Off	ACC, CMP	R/ Residue	Off	Off	W/ Index
LS	Op 1	Inner prod.	Off	On	Static	Off	ACC	Off	Off	W/ IR	R/ Index
	Op 2	MAC	On	Off	Dyn.	On	ACC, DIV	R/W **L**	R/ **D**	R/W IR	Off
	Op 3	Mult	Off	On	Static	Off	ACC	R/ **L**	W/ **D**	R IR	Off
	Op 4	MAC	On	Off	Dyn.	On	ACC, DIV	R/ $\mathbf{L}^{\mathrm{T}}$	Off	W/ IR	Off
EU		MAC	On	Off	Off	Off	ACC, ADD	R/W residue	R/W x	R/ IR	R/ Index

IR = Intermediate Result, R/W = read/write, x = sparse coefficient

Figure 18.5.2: PE and SC block diagrams and the dynamic configuration scheme of the SA engine for executing the AS, LS, and EU tasks of the OMP algorithm.

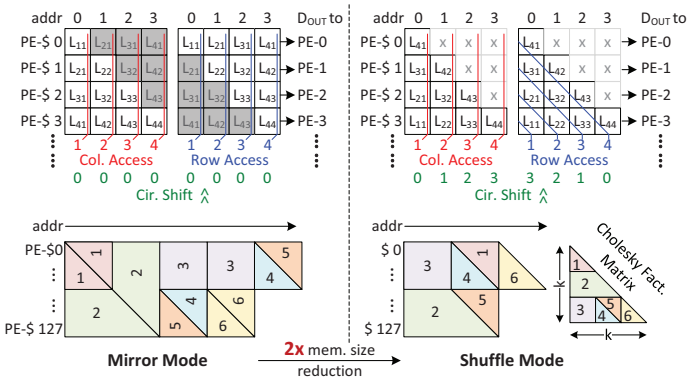


Figure 18.5.3: The memory access and folding scheme of PE caches for handling Cholesky factorization in mirror and shuffle mode. Column and row access is needed for solving FS and BS, respectively.

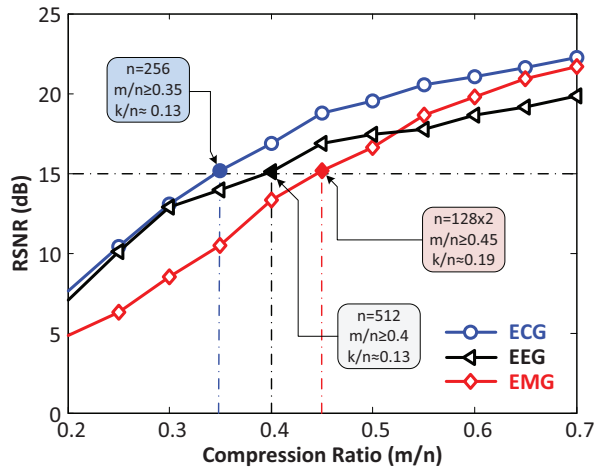
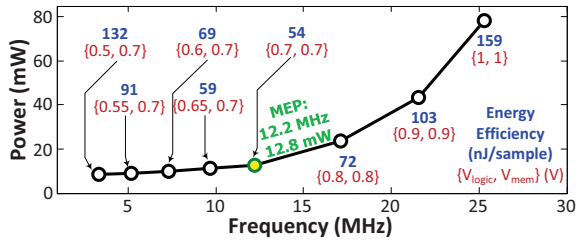


Figure 18.5.4: Measured RSNR of ECG, EEG, and EMG signals reconstructed on the DWT, the DCT, and joint DWT-DCT basis, respectively, with different signal dimensions (n), compression ratios (m/n), and signal sparsity (k/n).



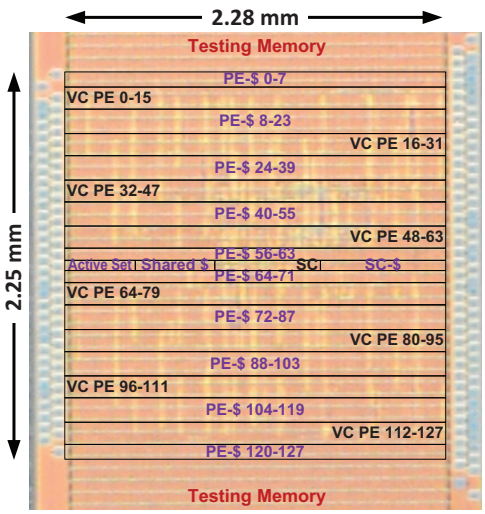
at MEP		Signal Dimension (n) 128	256	384	512	640	768	896	1024
Throughput (KS/s)	ECG	387	237	102	79	64	38	33	29
	EMG	213	123	54	41	24	20	13	12
	EEG	331	201	87	66	54	32	27	19
Energy Efficiency (nJ/sample)	ECG	33	54	125	163	200	337	390	444
	EMG	60	104	238	312	536	640	954	1087
	EEG	39	63	147	194	238	399	466	685

Figure 18.5.5: Power vs. frequency measured at different V_{DD}, measured throughput and energy efficiency for ExG reconstruction when operating at the MEP (the highlighted numbers are the best performance for achieving RSNR>15dB).

Design	Intel i7-4700MQ	[3]			[4]	This work
Technology	22nm	180nm			65nm	40nm
Target app	General	LTE channel estimation			Audio	Biomedical sensing
Algorithm	OMP/AMP	MP	GP	OMP	AMP	OMP, K-OMP
Signal dimension	Large	256			512[1]	up to 1024
Measurement dim.	Large	200			512	up to 512
Sparsity level	Large	50	18	10	-	up to 192
Core area (mm²)	174.4	0.73	1.21	2.42	0.629	5.13
Dynamic range	High (float-pt)	Low (fix-pt)			Low (fix-pt)	High (float-pt)
PE parallelism	SSE4	2	8		32	128
Local memory (KB)	7,424	-			5.76	147
Freq. (MHz)	2,394	140	128		333	27.4
Throughput (KS/s)	5 to 98[3]	2			397	12 to 237[2]
Power (mW)	47,000	88	209	200	177.5	8.6 to 78
Energy Efficiency* (nJ/sample)	451,322	2,444	5,778	11,222	-	32[4] (high sparsity)
	503,028	-			223	389[5] (low sparsity)

* Technology scaling to 40nm: Delay~1/S, Power~1/U², where S=L/40nm, U=V_{DD}/0.9V.
[1] The supported problem size is 1024x512, but only half of the sampling matrix is generic.
[2] ExG reconstruction throughput measured at MEP.
[3] ExG reconstruction throughput measured in MATLAB simulation.
[4,5] For fair comparison, the numbers are measured for the same problem size (m,n,k) as in [3,4].

Figure 18.5.6: Comparison with an Intel Core i7-4700MQ processor and prior chip implementations of the generic SA solver.



Technology	40nm 1P8M CMOS FO4 16.3ps (TT)
Transistor Flavor	SVT 0.11% HVT 99.89%
Transistor Count	61 M
I/Os	Digital 42/58 Power 156
Core V_{DD}	Logic: 0.5 to 1 V Mem: 0.7 to 1V
I/O V_{DD}	2.5 V
Core Size	2.28 x 2.25 mm

Figure 18.5.7: Die photo and chip summary.

978-1-4799-6224-2/15/$31.00