Minimal Activity Mixed-Signal VLSI Architecture for Real-Time Linear Transforms in Video

Rafal Karakiewicz
Department of Electrical and Computer Engineering
University of Toronto
Toronto, ON M5S 3G4, Canada
Email: rafal@eecg.toronto.edu

Roman Genov
Department of Electrical and Computer Engineering
University of Toronto
Toronto, ON M5S 3G4, Canada
Email: roman@eecg.toronto.edu

***Abstract*— The mixed-signal processor performs digital vector-matrix multiplication using internally analog fine-grain parallel computing. The three-transistor CID/DRAM unit cell combines single-bit dynamic storage, binary multiplication, and zero-latency analog accumulation. Matrix coefficients are stored in a bit-parallel form. Delta-sigma analog-to-digital conversion of the analog array outputs is combined with oversampled unary coding of the digital inputs. Sorting of unary inputs results in at most a single input line transition for arbitrary multi-bit inputs. This amounts to a linear gain in energy efficiency of the computational array in the number of bits of the input vector. The 256 $\times$ 128 CID/DRAM processor with integrated 128 delta-sigma ADCs measures 3 mm $\times$ 3 mm in 0.5 μm CMOS and delivers 6.5 GMACS dissipating 5.9 mW of power. CID/DRAM array dynamic power dissipation is reduced by a factor of four through sorting 8-bit inputs.**

I. Introduction

Real-time computing of linear transforms on a battery-powered mobile platform imposes great demands on computational throughput and power consumption. The computational core of linear transforms in image and video processing applications, such as artificial vision and human-computer interfaces, is that of vector-matrix multiplication (VMM) in high dimensions:

$$Y_m = \sum_{n=0}^{N-1} W_{mn} X_n \tag{1}$$

with N-dimensional input vector X_n, M-dimensional output vector Y_m, and $M \times N$ matrix elements W_{mn} (templates).

The presented mixed-signal VMM processor contains a fine-grain parallel computational array, achieving a computational throughput of 1.1 GMACS for every mW of power. In what follows we concentrate on massively parallel VMM computation on a mixed-signal VLSI architecture with minimal activity inputs.

II. Mixed-Signal Computation

A. Internally Analog, Externally Digital Computation

The approach combines the computational efficiency of analog array processing with the precision of digital processing and the convenience of a programmable and reconfigurable digital interface.

The digital representation is embedded in the analog array architecture, with matrix elements stored locally in bit-parallel form

$$W_{mn} = \sum_{i=0}^{I-1} 2^{-i-1} w_{mn}^{(i)} \tag{2}$$

and inputs presented in bit-serial fashion

$$X_n = \sum_{j=0}^{J-1} \gamma_j x_n^{(j)} \tag{3}$$

where the coefficients γ_j are assumed in radix two, depending on the form of input encoding used. The VMM task (1) then decomposes into

$$Y_m = \sum_{n=0}^{N-1} W_{mn} X_n = \sum_{i=0}^{I-1} 2^{-i-1} Y_m^{(i)} \tag{4}$$

with VMM partials

$$Y_m{}^{(i)} = \sum_{j=0}^{J-1} \gamma_j Y_m^{(i,j)}, \tag{5}$$

$$Y_m{}^{(i,j)} = \sum_{n=0}^{N-1} w_{mn}^{(i)} x_n^{(j)} \ . \tag{6}$$

The binary-binary partial products (6) are conveniently computed and accumulated, with zero latency, using an analog VMM array [1]-[2]. In principle, the VMM partials (6) can be quantized by a bank of flash analog-to-digital converters (ADCs), and the results accumulated in the digital domain according to (5) and (4) to yield a digital output resolution exceeding the analog precision of the array and the quantizers [3]. In the present work, an oversampling ADC accumulates the sum (5) in the analog domain, with inputs encoded in unary format ($\gamma_i = 1$). This avoids the need for high-resolution flash ADCs, which are replaced with single-bit quantizers in the delta-sigma loop.

B. CID/DRAM Cell and Array

The unit cell in the analog array combines a CID (charge injection device [4]) computational element [2] with a DRAM storage element. The cell stores one bit of a matrix element

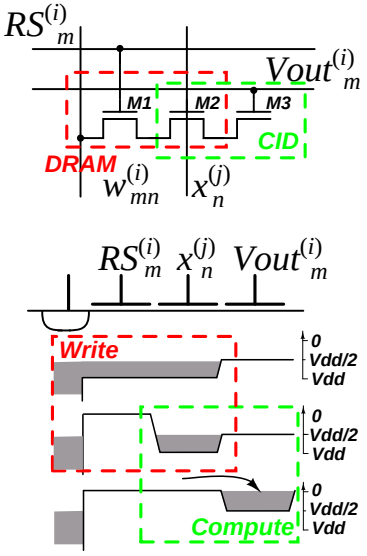

Fig. 1. CID (charge injection device) multiply-and-accumulate cell with integrated DRAM storage (*top*). Charge transfer diagram for active write and compute operations (*bottom*).

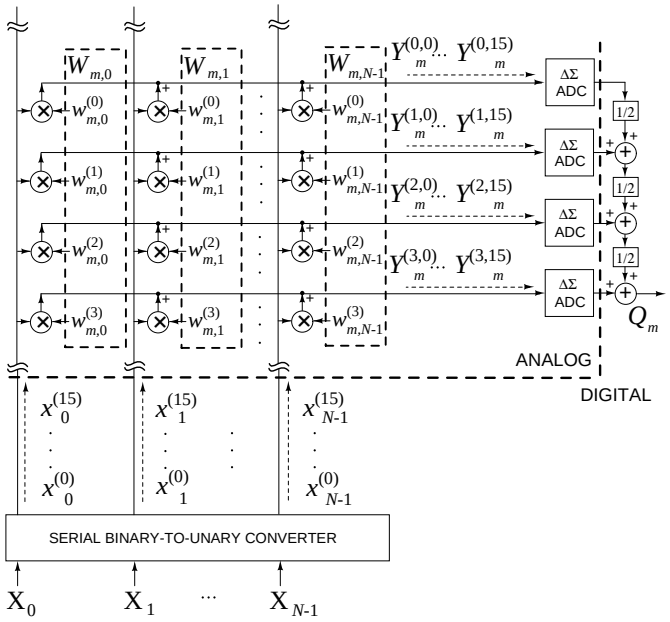

Fig. 2. Block diagram of one row in the matrix with binary encoded elements ${w^{(i)}}_{mn}$, for a single m and unary encoded inputs. Data flow of bit-serial inputs ${x^{(j)}}_n$ and corresponding partial product outputs ${Y^{(i,j)}}_m$, with $J = 16$ bits. The full product for a single row ${Y^{(i)}}_m$ is accumulated and quantized by a delta-sigma ADC. The final product is constructed in the digital domain according to (4).

${w_{mn}}^{(i)}$, performs a one-quadrant binary-unary (or binary-binary) multiplication of ${w_{mn}}^{(i)}$ and ${x_n}^{(j)}$ in (6), and accumulates the result across cells with common m and i indices. The circuit diagram and operation of the cell are given in Fig. 1. It performs a non-destructive computation since the transferred charge, Q, is sensed capacitively at the output. An array of cells thus performs (unsigned) binary-unary multiplication (6) of matrix ${w_{mn}}^{(i)}$ and vector ${x_n}^{(j)}$ yielding ${Y_m}^{(i,j)}$, for values of i in parallel across the array, and values of j in sequence over time.

C. Oversampling Mixed-Signal Array Processing

The conventional delta-sigma ($\Delta\Sigma$) ADC design paradigm allows to reduce requirements on precision of analog circuits to attain high resolution of conversion, at the expense of bandwidth. In the presented architecture a high conversion rate is maintained by combining delta-sigma analog-to-digital conversion with oversampled encoding of the digital inputs, where the delta-sigma modulator integrates the partial multiply-and-accumulate outputs (6) from the analog array according to (5).

Fig. 2 depicts one row of matrix elements W_{mn} in the $\Delta\Sigma$ oversampling architecture, encoded in $I = 4$ bit-parallel rows of CID/DRAM cells. One bit of a unary-coded input vector is presented each clock cycle, taking J clock cycles to complete a full computational cycle (1). The data flow is illustrated for a digital input series ${x_n}^{(j)}$ of $J = 16$ unary bits.

Over J clock cycles, the oversampling ADC integrates the partial products (6), producing a decimated output

$$Q_m^{(i)} \approx \sum_{j=0}^{J-1} \gamma_j Y_m^{(i,j)} \ , \qquad (7)$$

where $\gamma_j = 1$ for unary coding of inputs. Decimation for a first-order delta-sigma modulator is achieved using a binary counter.

Higher precision can be obtained in the same number of cycles J by using a higher-order delta-sigma modulator topology. However this drastically increases the implementation complexity. Instead, we use a modified topology that resamples the residue of the integrator after initial conversion [6]. A sample-and-hold resamples the residue voltage of the integrator and presents it to the modulator input for continued conversion at a finer scale. With a single resampling of the residue, the $\Delta\Sigma$ modulator obtains 8-bit effective resolution in 32 cycles.

D. VLSI Implementation

A mixed-signal VMM processor prototype integrated on a 3×3 mm^2 die was fabricated in 0.5 μm CMOS technology. The chip contains an array of 256×128 CID/DRAM cells, and a row-parallel bank of 128 $\Delta\Sigma$ algorithmic ADCs. Fig. 3 depicts the micrograph and system floorplan of the chip. The layout size of the CID/DRAM cell is $18\lambda \times 45\lambda$ with $\lambda = 0.3\mu m$.

The processor interfaces externally in digital format. Two separate shift registers load the templates along odd and even columns of the DRAM array. Integrated refresh circuitry periodically updates the charge stored in the array to compensate for leakage. Vertical bit lines extend across the array, with two rows of sense amplifiers at the top and bottom of the array. The refresh alternates between even and odd columns, with separate select lines.

Fig. 4 shows the measured linearity of the computational array. For every shift in the input register, a computation is performed and the result is observed on the output sense line.

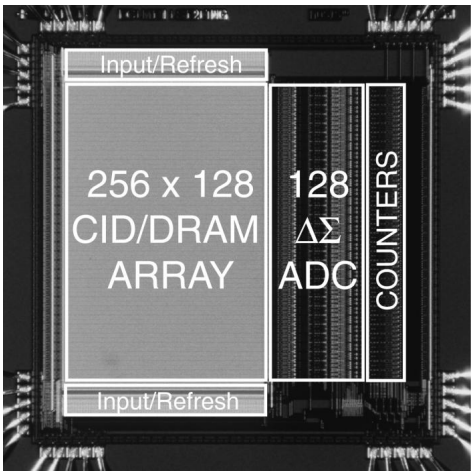

Fig. 3. Micrograph of the mixed-signal pattern recognition processor prototype, containing an array of 256×128 CID/DRAM cells, and a row-parallel bank of 128 $\Delta\Sigma$ algorithmic ADCs. Die size is $3\ \mathrm{mm} \times 3\ \mathrm{mm}$ in 0.5 μm CMOS technology.

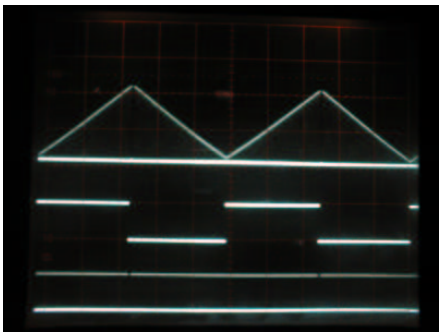

Fig. 4. Measured linearity of the computational array configured for signed multiplication on each cell (XOR configuration). Two cases are shown: binary weight storage elements are all actively charged, and all discharged. Waveforms shown are, *top to bottom*: the analog voltage output on the sense line; input data (in common for both input and weight shift register); and input shift register clock.

The chip contains 128 row-parallel $\Delta\Sigma$ algorithmic ADCs, *i.e.* one dedicated ADC for each m and i. In the present implementation, Y_m is obtained off-chip by combining the ADC quantized outputs ${Y^{(i)}}_m$ over i (rows) according to (4). The $\Delta\Sigma$ ADC yields 8-bit resolution over two subranging cycles of 4 bits each, for a total of 32 clock cycles [6].

Table I summarizes the measured performance. The CID/DRAM array dissipates 3.3 mW for a 10 μs computational cycle, and the bank of $\Delta\Sigma$ ADCs dissipates 2.6 mW yielding a combined conversion rate of 12.8 Msamples/s at 8-bit resolution.

III. Minimal Activity VMM Architecture

The oversampling architecture described in Section II-C maintains high throughput by combining all of the array computational cycles for a single bit-serial input within one delta-sigma modulated analog-to-digital conversion. In order to do so, the input is digitally oversampled by a binary-to-unary converter. The simplest K-bit binary-to-unary converter

TABLE I

Summary of measured performance

Technology	0.5 μm CMOS
Area	3mm $\times$ 3mm
Power	5.9 mW
Supply Voltage	5 V
Dimensions	256 inputs $\times$ 128 templates
Throughput	6.5 GMACS
Output Resolution	8-bit

is a bank of K latches (per input vector component) where the binary value stored in the k-th latch is presented to the output 2^k times, for $k = 0, ..., K-1$. Such conversion from the binary representation to the unary one preserves the number of bit-to-bit "0"-to-"1" and "1"-to-"0" logic level transitions in the bit-serial data stream. Dynamic power dissipated by the computational array is proportional to the number of such transitions as array input lines are driven to input vector coefficient $x_n^{(j)}$ values. The number of bit-to-bit transitions in each input vector component can range from 0 to K (counting the transition to the next input) depending on the input data statistics. Array dynamic power can therefore be minimized by minimizing input bit-to-bit transitions.

The purpose of this Section is two-fold. First, real image data statistics are presented demonstrating their non-minimal bit-to-bit transition activity. Second, a technique for minimizing the array dynamic power dissipation through unary input data sorting is presented and validated on real image data.

A. Real Image Data Statistics

In artificial vision systems and interactive human computer interfaces input data are real images. This Section investigates bit-to-bit transition statistics of real images on the example of *Lena*.

Fig. 5 depicts *Lena*'s bit transition statistics for different binary resolutions, K. For $K \geq 3$, LSB-to-(LSB-1) bit transitional probabilities (shown with the dashed line) are approximately 0.5. This bit transition Bernoulli probability distribution is due to the fact that real images have uncorrelated less significant bits, which are Bernoulli random variables themselves. Assuming independence, bit-to-bit "0"-to-"1" and "1"-to-"0" transitional probabilities are approximately equal to 0.25 each for those less significant bits. This yields approximately a 0.5 probability of a LSB-to-(LSB-1) transition for $K \geq 3$. The cumulative bit transitional probability (shown with the solid line in Fig. 5) is greater than 0.5 due to higher correlation of the more significant bits.

Dynamic power dissipation of the computational array is proportional to the input switching activity. The simple statistical study above demonstrates that when computation is performed on real image data, such as *Lena*, on average input transitions happen at least $K/2$ times for K-bit inputs.

B. Unary Input Sorting

The unary nature of the input allows to reduce the number of logic level transitions in its bit-serial sequence. Doing so

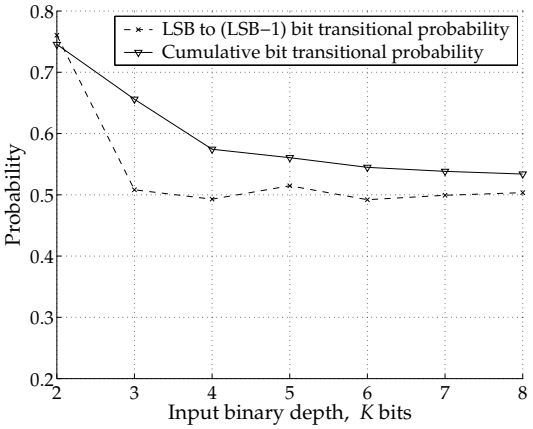

Fig. 5. *Lena* bit transitional probabilities.

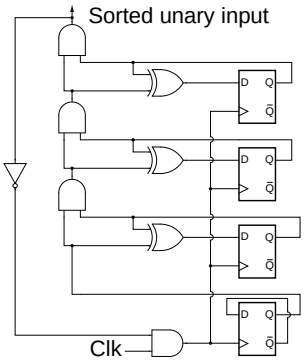

Fig. 6. An example of a 4-bit binary-to-sorted-unary converter implementation.

reduces the dynamic power dissipation of the computational array proportionally, without affecting the computation results. All unary coefficients have the same weight ($\gamma = 1$) and their temporal order in the bit-serial input sequence in the oversampling architecture in Fig. 2 is not important. The number of input transitions can be reduced from $K/2$ (for Bernoulli input data) to two per input component by simple bit sorting.

Bit sorting is a computationally inexpensive operation requiring little overhead. One example of a 4-bit binary-to-sorted-unary converter implementation is depicted in Fig. 6. The 4-bit shift register is a part of the data pipeline and is reused as a counting bit sorter by adding a few extra gates and switches per input component. The overhead is insignificant as it scales linearly in the number of input dimensions.

The number of input transitions can be further reduced to one per input component by extending the bit sorter in Fig. 6 to alternate sorting up and down for subsequent input vectors. The negligent overhead in integration area and power dissipation of the binary-to-sorted-unary converter yields at least a factor of $K/2$ decrease in power dissipation. For 8-bit images, this corresponds to a four-fold gain in energy efficiency. The results are validated by simulating the gain in energy efficiency for both Bernoulli data and *Lena* as shown in Figure 7.

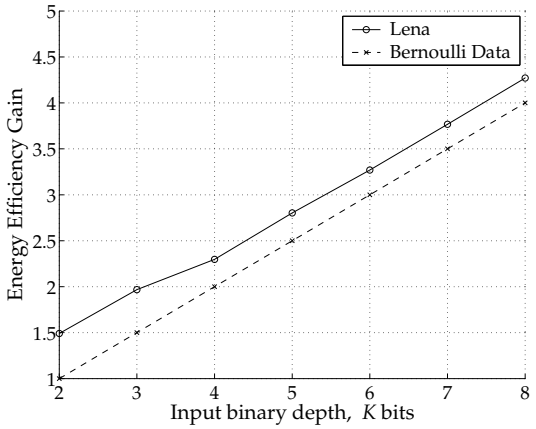

Fig. 7. Simulated improvement gain in energy efficiency of the CID/DRAM array when unary input sorting is implemented for random data with Bernoulli bit distribution and for real image data (*Lena*).

IV. Conclusions

A minimal switching activity oversampling charge-mode VLSI architecture for computing real-time linear transforms has been presented. An internally analog, externally digital architecture offers the best of both worlds: the density and energetic efficiency of an analog VLSI array, and the convenience and versatility of a digital interface. A $\Delta\Sigma$ oversampled algorithmic ADC architecture relaxes precision requirements in the quantization and allows for input bit sorting for minimum switching activity.

A 256 $\times$ 128 cell prototype was fabricated in 0.5 μm CMOS. The combination of analog array processing, oversampled input encoding, and $\Delta\Sigma$ algorithmic analog-to-digital conversion delivers a computational throughput of over 1 GMACS per mW of power, while maintaining 8-bit effective digital resolution. Input bit sorting reduces the CID/DRAM dynamic power dissipation by a factor of four for 8-bit inputs.

References

[1] A. Kramer, "Array-based analog computation," *IEEE Micro*, vol. **16** (5), pp. 40-49, 1996.

[2] V. Pedroni, A. Agranat, C. Neugebauer, A. Yariv, "Pattern matching and parallel processing with CCD technology," Proc. IEEE Int. Joint Conference on Neural Networks (IJCNN'92), vol. **3**, pp 620-623, 1992.

[3] R. Genov, G. Cauwenberghs "Charge-Mode Parallel Architecture for Matrix-Vector Multiplication," *IEEE T. Circuits and Systems II*, vol. **48** (10), 2001.

[4] M. Howes, D. Morgan, Eds., *Charge-Coupled Devices and Systems*, John Wiley & Sons, 1979.

[5] R. Genov, G. Cauwenberghs, "Stochastic Mixed-Signal VLSI Architecture for High-Dimensional Kernel Machines," *Advances in Neural Information Processing Systems*, Cambridge, MA: MIT Press, vol. 14, 2002.

[6] G. Mulliken, F. Adil, G. Cauwenberghs, R. Genov, "Delta-Sigma Algorithmic Analog-to-Digital Conversion," IEEE Int. Symp. on Circuits and Systems (ISCAS'02), Scottsdale, AZ, May 26-29, 2002. ISCAS'2002.