A 5.9mW 6.5GMACS CID/DRAM Array Processor

Roman Genov, Gert Cauwenberghs, Grant Mulliken, Farhan Adil
Department of Electrical and Computer Engineering, Johns Hopkins University
3400 N. Charles St. Baltimore, MD 21218, USA
{roman,gert}@jhu.edu

Abstract

The pattern recognition 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. Delta-sigma analog-to-digital conversion of the analog array outputs is combined with oversampled unary coding of the digital inputs. The 256 × 128 CID/DRAM processor with integrated 128 delta-sigma ADCs measures 3 mm × 3 mm in 0.5 μm CMOS and delivers 1.1 GMACS/mW.

1. Introduction

Real-time video pattern recognition [1] on a mobile platform imposes great demands on computational throughput and power consumption. The presented mixed-signal processor contains a fine-grain parallel computational array, achieving a computational throughput of 1.1 GMACS for every mW of power. The internally analog processor interfaces externally in digital format.

The computational core of template matching operations in image processing and pattern recognition 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). In what follows we concentrate on massively parallel VMM computation in an oversampled mixed-signal architecture.

2. Mixed-Signal Computation

2.1. 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 [2]-[4]. 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 [5]. 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.

2.2. CID/DRAM Cell and Array

The unit cell in the analog array combines a CID (charge injection device [6]) computational element [3, 4] with a DRAM storage element. The cell stores one bit of a matrix element $w_{mn}{}^{(i)}$, performs a one-quadrant binary-unary (or binary-binary) multiplication of $w_{mn}{}^{(i)}$

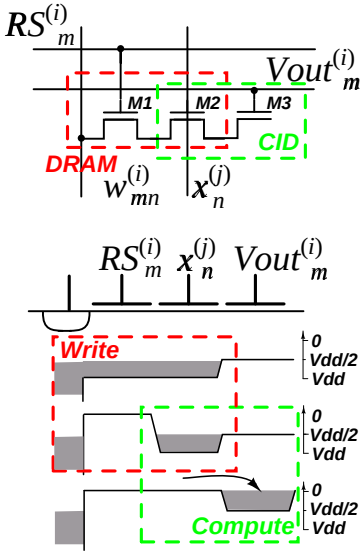

Figure 1. CID (charge injection device) multiply-and-accumulate cell with integrated DRAM storage (*top*). Charge transfer diagram for active write and compute operations (*bottom*).

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 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.

To improve linearity and to reduce sensitivity to clock feedthrough, we use differential encoding of input and stored bits in the CID/DRAM architecture using twice the number of columns and unit cells as shown in Fig. 2. This amounts to exclusive-OR (XOR), rather than AND, multiplication on the analog array, using signed, rather than unsigned, binary values for inputs and weights, $x_n{}^{(j)} = \pm 1$ and $w_{mn}{}^{(i)} = \pm 1$.

3. 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).

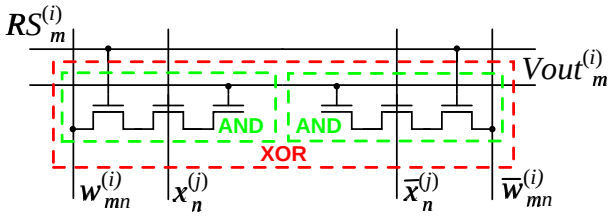

Figure 2. Two charge-mode AND cells configured as an exclusive-OR (XOR) multiply-and-accumulate gate.

3.1. Array Architecture

Fig. 3 depicts one row of matrix elements $W^{(m,n)}$ 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 $x_j{}^{(n)}$ 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.

3.2. Row-parallel $\Delta\Sigma$ Algorithmic ADC

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 shown in Fig. 4 that resamples the residue of the integrator after initial conversion. 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. The principle is analogous to extended counting [8] but avoids additional hardware by reusing the same $\Delta\Sigma$ modulator to quantize the residue.

Similar to residue resampling in an algorithmic (or cyclic) ADC, for each resampling the scale of conversion subranges to the LSB level of the previous conversion. For a first-order incremental $\Delta\Sigma$ ADC [7], resampling of the residue scales the range by a factor L, where L is the number of modulation cycles. If L is of radix two, *i.e.*, $L = 2^\ell$, then the subranging is conveniently accomplished in the architecture of Fig. 4 by shifting the bits in the decimating counter by ℓ positions for every resampling of the residue.

Every resampling improves the output resolution by a factor L, or ℓ bits, limited by noise and mismatch in the implementation. The effect of capacitance mismatch is minimized by using a ratio-insensitive scheme for resampling the residue, shown in Fig. 5. The presented scheme is equivalent to algorithmic A/D conversion, but avoids in-

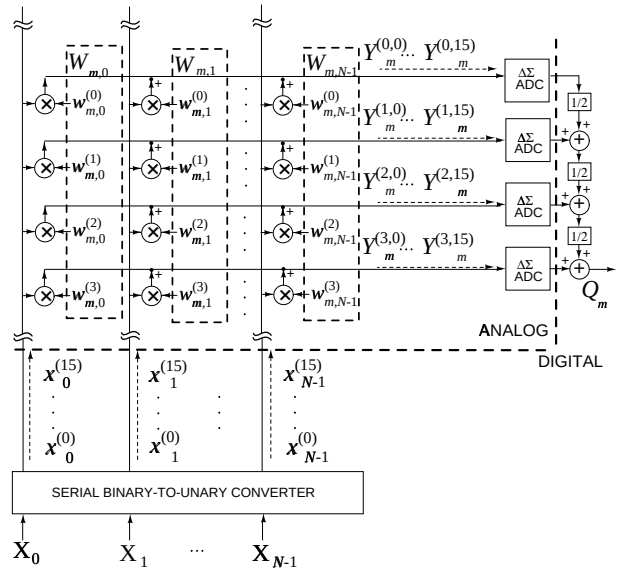

Figure 3. Block diagram of one row in the matrix with binary encoded elements $w^{(i)}{}_{mn}$, for a single m and unary encoded inputs. Data flow of bit-serial inputs $x^{(j)}{}_n$ and corresponding partial product outputs $Y^{(i,j)}{}_m$, with $J = 16$ bits. The full product for a single row $Y^{(i)}{}_m$ is accumulated and quantized by a delta-sigma ADC. The final product is constructed in the digital domain according to (4).

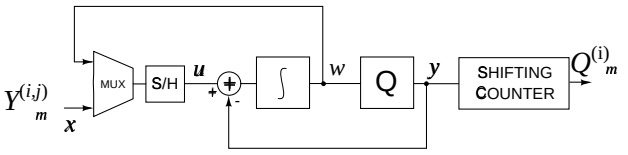

Figure 4. Block diagram of $\Delta\Sigma$ algorithmic ADC with residue resampling, including decimator. One ADC is provided for each row in the array architecture.

terstage gain errors without the need for precisely ratio-ed analog components.

The resampling of the residue in the oversampled ADC can be combined with correspondingly rescaling the coefficients γ_j in the input encoding. In principle, higher resolution digital inputs can be presented by unary encoding bits in groups of ℓ, each covering L modulation cycles of the subranging oversampled ADC. In the example of Fig. 3, only the first 4 bits are unary encoded and presented in the first algorithmic cycle, with $L = 16$. With a single resampling of the residue, the $\Delta\Sigma$ modulator obtains $2\ell = 8$ bit effective resolution in $2L = 32$ cycles.

4. Experimental Results

A pattern recognition processor prototype integrated on a $3 \times 3\,\mathrm{mm}^2$ die was fabricated in 0.5 μm CMOS technology. The chip contains an array of 256×128 CID/DRAM

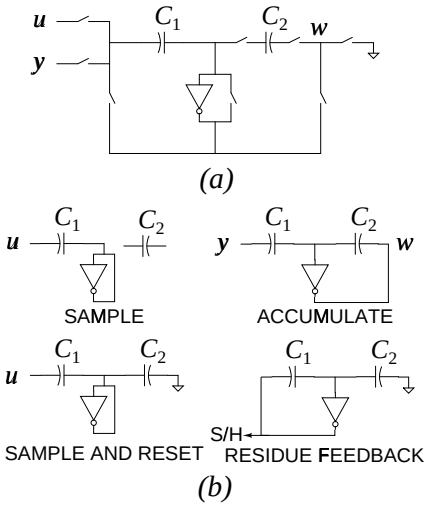

Figure 5. Invertible, resetable analog accumulator. *(a)* Circuit diagram. *(b)* Operational modes.

cells, and a row-parallel bank of 128 $\Delta\Sigma$ algorithmic ADCs. Fig. 6 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. 7 shows the measured linearity of the computational array, configured differentially for signed (XOR) multiplication. The case shown is where all binary weight storage elements are actively charged ('1' bit sequence), and an '1' sequence of bits is shifted through the input register, initialized to '0' bit values. For every shift in the input register, a computation is performed and the result is observed on the output sense line.

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-bit each, for a total of 32 clock cycles as shown in Fig. 8.

Table 1 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.

5. Conclusions

An oversampling charge-mode VLSI architecture for parallel vector-matrix multiplication has been presented. An internally analog, externally digital architecture offers

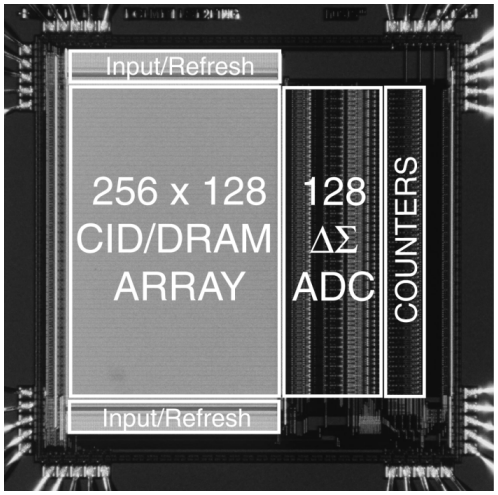

Figure 6. Micrograph of the mixed-signal pattern recognition processor prototype, containing an array of 256×128 CID/DRAM cells, and a row-parallel bank of 128 $\Delta\Sigma$ algorithmic ADCs. Die size is $3\ \mathrm{mm} \times 3\ \mathrm{mm}$ in 0.5 μm CMOS technology.

Table 1. Summary of measured performance

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

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.

A 256×128 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 1GMACS per mW of power, while maintaining 8-bit effective digital resolution.

Acknowledgments: This research was supported by ONR N00014-99-1-0612 and ONR/DARPA N00014-00-C-0315. The chip was fabricated through the MOSIS service.

[1] Papageorgiou, C.P, Oren, M. and Poggio, T., "A General Framework for Object Detection," in *Proceedings of International Conference on Computer Vision*, 1998.

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

[3] C. Neugebauer and A. Yariv, "A Parallel Analog CCD/CMOS Neural Network IC," Proc. IEEE Int. Joint Conference on Neural Networks (IJCNN'91), Seattle, WA, vol. **1**, pp 447-451, 1991.

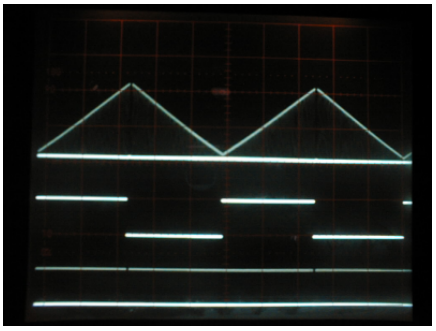

Figure 7. Measured linearity of the computational array configured for signed multiplication on each cell (XOR configuration). Two cases are shown: binary weight storage elements are all actively charged, and all discharged. Waveforms shown are, *top to bottom*: the analog voltage output on the sense line; input data (in common for both input and weight shift register); and input shift register clock.

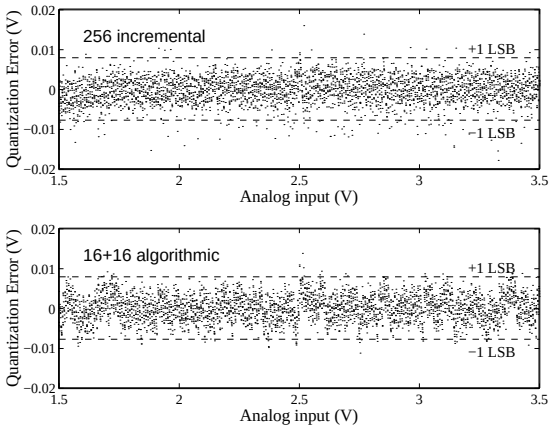

Figure 8. *Integral quantization residue, recorded from a single quantizer channel of the CID/DRAM processor in Figure 6, configured for 8-bit conversion.* Top: $\Delta\Sigma$ *incremental conversion* ($N = 256$). Bottom: $\Delta\Sigma$ *algorithmic A/D conversion* ($N = 16$, *2-step).*

[4] 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.

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

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

[7] O.J.A.P. Nys and E. Dijkstra, "On Configurable Oversampled A/D Converters," *IEEE J. Solid-State Circuits*, vol. **28** (7), pp 736-742, 1993.

[8] R. Harjani and T.A. Lee, "FRC: A Method for Extending the Resolution of Nyquist Rate Converters Using Oversampling," *IEEE T. Circuits and Systems II*, vol. **45** (4), pp 482-494, 1998.