ALGORITHMIC PARTIAL ANALOG-TO-DIGITAL CONVERSION IN MIXED-SIGNAL ARRAY PROCESSORS

Roman Genov[1,2] *and Gert Cauwenberghs*[2]

[1]Department of Electrical and Computer Engineering
University of Toronto, Toronto, ON M5S 3G4, Canada
[2]Department of Electrical and Computer Engineering
The Johns Hopkins University, Baltimore, MD 21218-2686
E-mail: *roman@eecg.toronto.edu, gert@jhu.edu*

ABSTRACT

We present an algorithmic analog-to-digital converter (ADC) architecture for large-scale parallel quantization of internally analog variables in externally digital array processors. The converter quantizes and accumulates a binary weighted sequence of partial binary-binary matrix-vector products computed on the analog array, under presentation of bit-serial inputs in descending binary order. The architecture combines algorithmic conversion of the residue, as in a standard algorithmic ADC, with synchronous accumulation of the partial products from the array. In conjunction with row-parallel digital storage of matrix elements in the array, two pipelined architectures are presented to accumulate partial products with common binary weight across rows: row-parallel ADC with digital post-accumulation, and row-cumulative ADC with analog pre-accumulation. Simulation results are presented to quantify the trade-off in precision and area for full-parallel flash, and row-parallel and row-cumulative partial algorithmic, analog-to-digital conversion on the array.

1. INTRODUCTION

An internally analog, externally digital architecture for parallel matrix-vector multiplication (MVM) was reported in [1]. The VLSI mixed-signal processing outperforms purely digital approaches by several orders of magnitude in throughput, density and energy efficiency, owing to massive parallelism in the architecture with bit-level distributed memory and processing elements. This template matching architecture consists of an analog computational array and a bank of row-parallel flash quantizers. A three-transistor unit cell combines a single-bit dynamic random-access memory (DRAM) and a charge injection device (CID) binary multiplier and analog accumulator. Digital multiplication of variable resolution is obtained with bit-serial inputs and bit-parallel storage of matrix elements

$$X_n = \sum_{j=0}^{J-1} 2^{-j-1} x_n^{(j)} \tag{1}$$

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

This work was supported by ONR N00014-99-1-0612.

by combining quantized outputs from multiple rows of cells (index i) over time (index j)

$$Y_m = \sum_{n=0}^{N-1} W_{mn} X_n = \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} 2^{-i-j-2} Y_m^{(i,j)} \tag{3}$$

with binary-binary MVM partials computed in a single array row

$$Y_m^{(i,j)} = \sum_{n=0}^{N-1} w_{mn}^{(i)} x_n^{(j)} . \tag{4}$$

The data flow over bit planes in the input and stored templates is illustrated in Figure 1. The benefits of using row-parallel flash ADCs to quantize partial products (4) are high conversion speed and additional gains (1.6 bits) in resolution achieved by averaging weighted quantized outputs across multiple rows and over several computational cycles [1]. The drawback of such implementation is exponential dependence of integration area and power dissipation on resolution, making high-resolution implementations expensive and impractical for use on portable platforms (*e.g.* miniature mobile artificial vision systems).

A second generation *Kerneltron* processor was reported in [2]. Its application to real-time object detection in streaming video in the framework of support vector machine learning paradigm [3] was demonstrated in [4]. *Kerneltron* II performs row-parallel delta-sigma analog-to-digital conversion combined with oversampled (unary) encoding of inputs achieving 8-bit output resolution and yielding significant data throughput for low-resolution inputs. The architecture employs algorithmic delta-sigma analog-to-digital converters [5] combining properties of algorithmic and oversampling ADCs. As shown in [5] an algorithmic delta-sigma ADC can be configured to allow for linear dependence of throughput on the number of bits in the bit-serial input representation by alternating sampling of a row output with residue resampling. Such a configuration yields a level of computational throughput comparable to that of the flash ADC architecture (for the same analog array computational cycle time) but with some temporal overhead due to several cycles needed to implement the radix in algorithmic conversion [5].

In this work we present two methods of quantizing analog array outputs achieving higher computational throughput, and offering advantages in conversion resolution, power dissipation (for the same area) and complexity of implementation. Section 2 intro-

0-7803-7761-3/03/$17.00 ©2003 IEEE

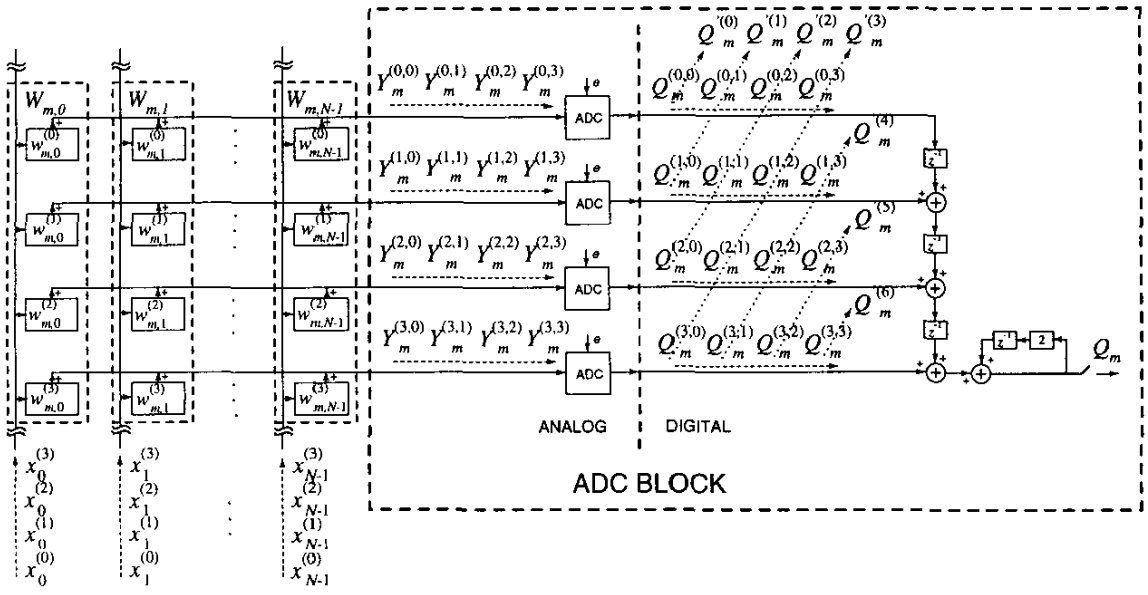

Figure 1: Block diagram of one row of the matrix in a matrix-vector multiplication mixed-signal VLSI architecture with flash analog-to-digital converters as the choice of quantizers. Matrix elements W_{mn} are binary encoded ($w_{mn}{}^{(i)}$ are their binary coefficients), and shown for a single m and $I = 4$ bits. Data flow of bit-serial (LSB-first) inputs $x_n{}^{(j)}$, partial outputs $Y_m{}^{(i,j)}$, quantized partial outputs $Q_m{}^{(i,j)}$, and the constructed quantized inner-product Q_m ($Q_m{}'^{(k)}$ are its binary coefficients, k=0,...I+J-2) with $J = 4$ bits.

duces the architecture of a row-parallel algorithmic partial analog-to-digital converter. It uses an additional residue modulator to extend its input dynamic range such that every cycle a sum of the previous iteration residue and a newly computed partial product output from the array can be partially quantized and accumulated. Algorithmic analog-to-digital conversion iterations and accumulation of partial products are thus interleaved. For every computational cycle on the matrix-vector multiplying array, the ADC performs one algorithmic iteration. This is in contrast to the full-parallel ADC configuration of Figure 1, where a flash converter performs a full ADC for every computational cycle, with digital outputs accumulated in the digital domain. Section 3 demonstrates how outputs of several rows, storing matrix elements in bit-parallel fashion, can be accumulated in analog domain while being partially quantized by using a row-cumulative algorithmic partial analog-to-digital converter. Conclusions are given in section 4.

2. ROW-PARALLEL ALGORITHMIC PARTIAL ANALOG-TO-DIGITAL CONVERSION

Instead of full quantization of analog array outputs (4) after every computation as in the case of flash architecture in Figure 1, algorithmic partial analog-to-digital converter accumulates array row outputs (the inner sum in (3)) while performing their partial quantization by resampling the residue at the same time. The principle of operation combines properties of pipelined and iterative algorithmic quantizers in that the bit-serial input signal is interleaved with the previous cycle of residue modulation in algorithmic conversion.

To design a row-parallel algorithmic partial ADC we use residue modulators [6] comprising a quantizer and an adder. Its diagram and transfer function are shown in Figure 2 *(a)*. The digital code is generated as:

$$D_{out1} := (V_{in1} > V_{ref}), \tag{5}$$

with the signal range decreasing by a factor of two:

$$V_{out1} = V_{in1} - V_{ref} D_{out1}, \tag{6}$$

as shown in the figure.

A radix-2 iterative algorithmic ADC consisting of a residue modulator, multiply-times-two element, and unit delay element is shown in Figure 2 *(b)*. The output code is generated serially:

$$D_{out} := (2V'_{in} > V_{ref}) \tag{7}$$

At initialization the input signal is sampled:

$$V'^{(0)}_{in} = V_{in}, \tag{8}$$

producing MSB value in the same clock cycle. For standard algorithmic analog-to-digital conversion, the remaining bits are obtained by iterative conversion of the radix-2 residue of the previous conversion cycle:

$$V'^{(i+1)}_{in} = V^{(i)}_{out} \tag{9}$$

where the output voltage (in the radix-2 case) is defined as:

$$V_{out} = 2V'_{in} - D_{out} V_{ref}, \tag{10}$$

and where V_{ref} equals the (effective) input signal range. The range of the residue signal is the same as the input range as shown on the transfer characteristic on the right of Figure 2 *(b)*.

The radix-2 residue modulation scheme for the algorithmic ADC can be extended to accumulate and quantize partial results from the computational array in synchrony with the algorithmic

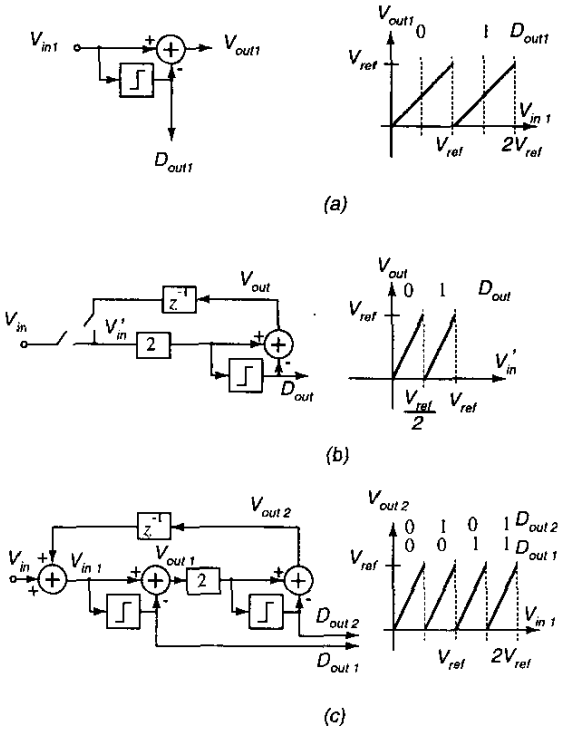

Figure 2: *(a)* Block diagram of a residue modulator; *(b)* a standard algorithmic ADC; *(c)* an algorithmic partial ADC with cumulative input; and their respective transfer characteristics.

iteration. The inherent property of the analog array architecture is bit-serial representation of the row outputs, $Y_m{}^{(i,j)}$. As the input vectors binary coefficients, $x_n{}^{(j)}$, are presented MSB-first over time, the outputs from the array are binary weighted in descending order (MSB-first) as well. We use this property in a design of a residue-input additive algorithmic partial ADC presented in Figure 2 *(c)*.

In this scheme, the input to the rightmost quantizer is not only the residue of the previous cycle computation, but also combines the row output produced in the given cycle, which carries the same binary weight because partial analog outputs from the array arrive in descending order of binary weights. The addition of the residue and the additional input from the array doubles the signal range making it $2V_{ref}$. This calls for an additional block to perform extra 1-bit quantization and halve the range. For this purpose we insert the residue modulator shown in Figure 2 *(a)* in the conversion loop, with generated digital code described by (5).

The code for the second output bit of the algorithmic partial ADC is:

$$D_{out2} := (2V_{out1} > V_{ref}) \tag{11}$$

Substituting equation (5) into (6) and then into (11), we get:

$$\begin{aligned} D_{out2} &:= [2(V_{in1} - D_{out1}V_{ref}) > V_{ref}] \\ &:= [2(V_{in1} - (V_{in1} > V_{ref})V_{ref}) > V_{ref}] \end{aligned} \tag{12}$$

The expression for the output voltage

$$V_{out2} = 2V_{out1} - D_{out2}V_{ref} \tag{13}$$

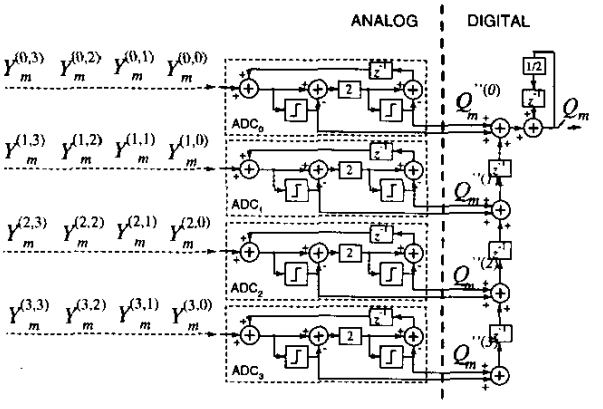

Figure 3: Block diagram of the ADC block implemented with row-parallel algorithmic partial ADCs, in common for m-th output vector component with I-bit matrix elements and J-bit input vector. The case shown is $I = J = 4$ with MSB-first bit-serial input representation.

can be expanded, using equation (6), as

$$V_{out2} = 2(V_{in1} - D_{out1}V_{ref}) - D_{out2}V_{ref}, \tag{14}$$

describing a four-segment, double-range, gain-2 transfer characteristic shown in Figure 2 *(c)*.

The described 2-bit-per-iteration algorithmic partial ADC is used in the row-parallel architecture shown in Figure 3, implementing an alternative to the flash-parallel ADC block of Figure 1. In digital accumulation of partial products $Q'_m{}^{(i)}$, in *Central Limit*, the quantization noise terms variances are additive. It can be shown that due to averaging the signal-to-noise ratio of the overall computation is enhanced by approximately 0.8 bits compared to that of each row-parallel quantizer [7] as shown in Figure 5.

When all of a row outputs have been fed into the algorithmic partial ADC, the conversion can be continued by operating on the signal residue (as in standard iterative algorithmic ADCs). In this case the input signal is equal to zero and the residue is of the range V_{ref}, so the output of the first quantizer is always zero. The resolution in this case is limited only by circuit implementation inaccuracies (*i.e.* gain error, second quantizer comparator offset).

3. ROW-CUMULATIVE ANALOG-TO-DIGITAL CONVERSION

The architecture described above can be extended to perform quantization of outputs across multiple rows in the analog array, storing digital matrix elements in bit-parallel fashion. Several rows thus produce partial products of the same output, which can be pooled and quantized using the same algorithmic partial ADC. Analog outputs common to m-th matrix row are combined using a pipelined accumulator. The accumulation is performed both in analog and digital domains by means of the cumulative residue modulation architecture shown in Figure 4 (in this example matrix elements are 4-bit binary encoded).

For MSB-first bit-serial input vector, X_n, elements $Y_m{}^{(i,j)}$ of the analog array output matrix have the same binary weight when they are sampled for common values of $i + j$ in time and space.

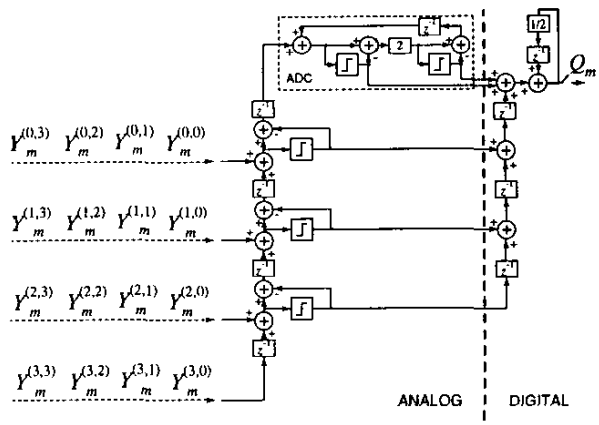

Figure 4: Row-cumulative quantizer architecture with a single algorithmic partial ADC for m-th output vector component with I-bit matrix elements and J-bit input vector. The case shown is $I = J = 4$ with MSB-first input vectors bit-serial representation.

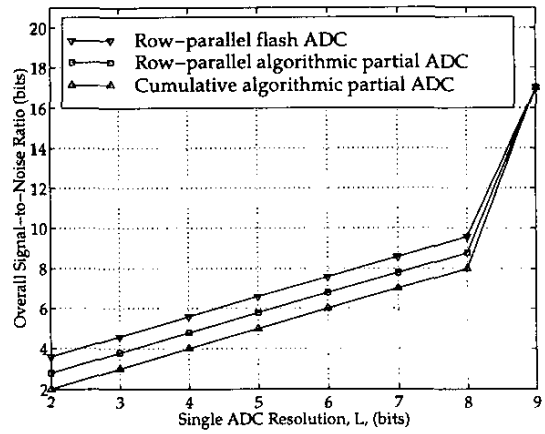

Figure 5: Comparison of resolution gains in three parallel analog-to-digital converters for $N = 511$ and $I = J = 4$. Due to quantization noise averaging in digital accumulation of quantized array outputs, flash ADC architecture enhances computation resolution by approximately 1.6 bits [1], algorithmic partial ADC architecture – by 0.8 bits. Row-cumulative architecture offers the native resolution of the converter as no digital averaging takes place. For $L = 9$ the resolution of ADCs exactly matches the dimensionality of the input vector yielding the ideal overall SNR.

This can be accomplished by delaying accumulated contributions along rows in pipelined fashion. In an analog accumulator, every addition of two equally weighted outputs increases the signal range by V_{ref}. To retain large dynamic range while avoiding saturation, a cascade of residue modulators [6], one per row, is used instead with their outputs accumulated by a digital delay line. Whenever an analog sum exceeds the range, a carry bit is generated and propagated in the digital domain. Effectively two goals are reached. The signal range is kept constant while the analog value is being partially quantized.

By the time an analog value is accumulated over I rows, $I - 1$ bits of its digital representation are already available. Cross-row partial sums are further quantized by an algorithmic partial ADC presented in the previous section. Here, they are combined with the previous cycle residue to produce remaining output bits. In digital domain all of the bits are accumulated to produce the full digital representation of the constructed inner-product, Q_m, for m-th matrix row. Further fine-scale conversion can be performed by operating solely on the signal residue.

The cumulative algorithmic partial analog-to-digital conversion scheme uses one ADC per one output vector component, Y_m. Addition in analog domain prior to conversion requires fewer ADCs per array. It also simplifies digital post-processing, eliminating most of the need for addition and multiplication. The system however is not as versatile or scalable, has higher latency and reduced throughput, and forfeits the gains in resolution that characterize row-parallel architectures as illustrated in Figure 5.

4. CONCLUSIONS

We presented a partial quantization technique, algorithmic partial analog-to-digital conversion, and its use in row-parallel and row-cumulative data conversion in mixed-signal array processors with bit-level binary data encoding. Algorithmic partial ADC simultaneously samples iterative residue and analog array binary-weighted serial output. By accumulating array row outputs in time while performing partial quantization, algorithmic partial ADC achieves computational throughput equivalent to that of a flash ADC architecture while offering advantages in area and power dissipation over their exponential dependence on resolution in flash ADCs. The row-parallel architecture allows to maximize the computational throughput, at a slight expense in resolution over the flash-parallel architecture since binary weighted accumulation is partially performed in the analog domain. The row-cumulative implementation further simplifies both analog and digital implementation, but offers the native resolution of the converter without the gains in resolution through digital averaging that the other two architectures offer.

5. REFERENCES

[1] R. Genov, G. Cauwenberghs, "Charge-Mode Parallel Architecture for Matrix-Vector Multiplication," *IEEE T. Circuits and Systems II*, vol. **48** (10), pp. 930-936, 2001.

[2] R. Genov, G. Cauwenberghs, G. Mulliken, F. Adil, "A 5.9mW 6.5GMACS CID/DRAM Array Processor," *Proc. European Solid-State Circuits Conference (ESSCIRC'2002)*, Florence Italy, Sept. 24-26, 2002.

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

[4] R. Genov, G. Cauwenberghs, "Kerneltron: Support Vector 'Machine' in Silicon," *Proc. SVM'2002*, Lecture Notes in Computer Science, Niagara Falls ON, Aug. 10, 2002.

[5] 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.

[6] 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.

[7] R. Genov, "Massively Parallel Mixed-Signal VLSI Kernel Machines," Ph.D. Thesis. Johns Hopkins University, Baltimore, MD 2002.