480-GMACS/mW Resonant Adiabatic Mixed-Signal Processor Array for Charge-Based Pattern Recognition

Rafal Karakiewicz, *Student Member, IEEE,* Roman Genov, *Member, IEEE,*
and Gert Cauwenberghs, *Senior Member, IEEE*

***Abstract*— A resonant adiabatic mixed-signal VLSI array delivers 480 GMACS (10^9 multiply-and-accumulates per second) throughput for every mW of power, a 25-fold improvement over the energy efficiency obtained when resonant clock generator and line drivers are replaced with static CMOS drivers. Losses in resonant clock generation are minimized by activating switches between LC tank and DC supply with a periodic pulse signal, and by minimizing the variability of the capacitive load to maintain resonance. We show that minimum energy is attained for relatively wide pulse width, and that typical load distribution in template-based charge-mode computation implies almost constant capacitive load. The resonantly driven 256×512 array of 3-T charge-conserving multiply-accumulate cells is embedded in a template matching processor for image classification and validated on a face detection task.**

***Index Terms*— Adiabatic low-power techniques, resonant clock supply, computational memory, pattern recognition.**

I. Introduction

LOW power dissipation is a critical objective in the design of portable and implantable microsystems supporting the use of a miniature battery power supply, wireless power harvesting, or other low-energy power sources. Typical power budgets are in the low milliwatts for wearable devices and low microwatts for implantable systems. Despite the shrinking power budgets, there is ever more a need for high throughput computing and embedded signal processing. Future generations of wearable and implantable devices call for the integration of complex signal extraction and coding functions along with sensing and communication. Portable real-time pattern recognition systems, such as wearable face detection and recognition systems for the blind, are examples of such applications. The energy efficiency, defined as computational throughput per unit power (or, equivalently, the reciprocal of the energy per unit computation), thus has to be maximized. In this work the adiabatic charge-recycling principle is applied to mixed-signal charge-based computing to decrease power dissipation beyond $fCVdd^2$, while high computational throughout is maintained by employing an array-based parallel computing architecture.

When a CMOS inverter, in Fig. 1(a), charges a load capacitance C to voltage Vdd, the total energy taken from the voltage supply source is $CVdd^2$. Half of it is used to charge C, and the other half is dissipated in the pull-up network. When the output is driven low, the pull-down network discharges the energy stored in C, $\frac{1}{2}CVdd^2$, to ground. The resistances of the pull-up and pull-down networks affect the minimum charging and discharging times, but not the dynamic energy dissipated. The dynamic energy dissipation can be lowered by reducing supply voltage, load capacitance, or both.

Dynamic energy dissipation has a quadratic dependence on the supply voltage. This makes the reduction of supply voltages the most effective way to reduce dynamic energy dissipation. Dynamic voltage scaling has become a standard approach for reducing power dissipation when performance requirements vary in time. In modern processors the voltage and frequency are controlled in a feedback loop to maintain operation within a target power and temperature budget [1]. Local voltage dithering which toggles the supply between a small number of voltage levels to locally optimize energy consumption based on the workload of each circuit block has been reported [2]. Subthreshold circuits operate with the supply voltage below the threshold voltage of devices to further

R. Karakiewicz was with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada. He is now with SNOWBUSH Microelectronics, Toronto, ON M5G 1Y8, Canada (e-mail: raf@snowbush.com).

R. Genov is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: roman@eecg.utoronto.ca).

G. Cauwenberghs is with the University of California San Diego, La Jolla, CA 92093-0357, USA (e-mail: gert@ucsd.edu).

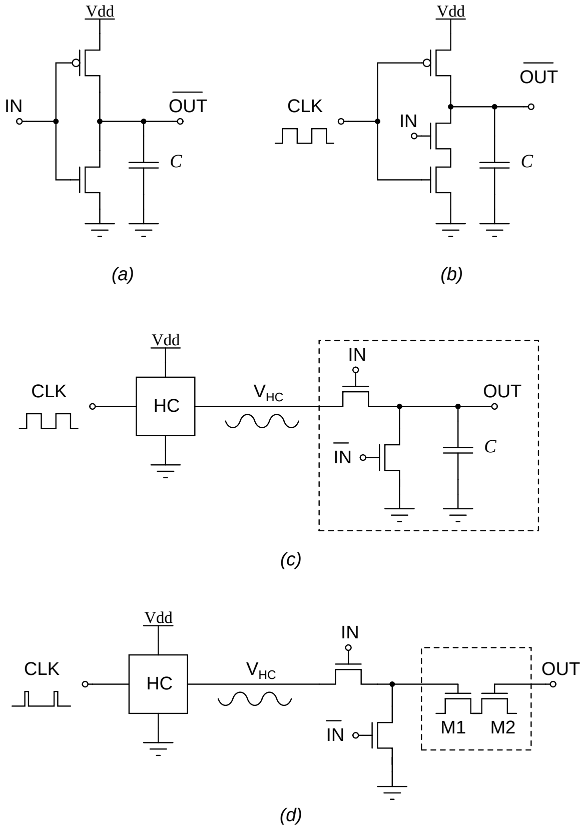


Fig. 1. Dynamic dissipation and resonant adiabatic energy recovery. *(a)* CMOS logic modeled as inverter driving a capacitive load; *(b)* CMOS dynamic logic equivalent; *(c)* Adiabatic logic modeled as transmission gate driving a capacitive load from a 'hot clock' V_{HC}; and *(d)* Adiabatic mixed-signal multiply-accumulation (MAC). A single cell in the MAC array is shown, with the charge-coupled MOS pair comprising a variable capacitive load.

reduce dynamic energy dissipation. A subthreshold static random access memory (SRAM) [3] and fast Fourier transform (FFT) processor [4] have recently been reported with optimal supply voltages of 300 mV and 350 mV respectively. By applying forward body bias the threshold voltage can be shifted lower to allow further voltage scaling and thus energy reduction [5].

The dynamic energy dissipation is reduced linearly with the load capacitance. If the speed is not critical, minimum device sizing reduces the capacitance at the cost of non-optimal propagation delay times. Dynamic logic, in Fig. 1(b), can be used to eliminate most of the PMOS capacitance. Finally, capacitance can be lowered by migrating to a new technology process with smaller minimum feature size at the cost of increased static power dissipation due to transistor leakage.

As opposed to static or dynamic CMOS logic drivers, adiabatic drivers slowly ramp the supply voltage from 0 V during the pull-up phase to reduce the voltage drop across the pull-up network. The voltage drop is made arbitrarily small by keeping the ramp period sufficiently longer than the time constant of the driver [6]. Generation of the ramp signal implies that power dissipation is reduced at the system level, not only the gate level. For long ramp periods, the voltage across C is approximately equal to the supply ramp voltage and the energy taken from the voltage source is $\frac{1}{2}CVdd^2$, the minimum required to charge C to Vdd. In general, a linear increase in the ramp charging time results in a linear decrease in the voltage drop across the pull-up network, and thus a linear decrease in dynamic energy dissipation. In the pull-down phase the energy stored on C is slowly discharged back into the supply voltage source by slowly ramping Vdd back to 0 V, again keeping resistive losses at a minimum. A number of adiabatic logic families utilizing voltage ramps have been developed such as adiabatic dynamic logic (ADL) [7], efficient charge recovery logic (ECRL) (2N2P) logic [8], pass-transistor adiabatic logic (PAL) [9], clocked adiabatic logic (CAL) [10], and true single-phase energy-recovery logic (TSEL) [11].

Generating ideal linear voltage ramps to provide constant charging and discharging currents incurs power dissipation in the supply generator, defeating the savings by adiabatic energy recovery. An oscillatory waveform, or hot clock (HC), from a resonator is typically used instead [6]–[8], [12], [13]. The increased energy dissipation in the pull-up network, due to the non-optimal sinusoidal shape [6], is offset by the low energy dissipation and simplicity of resonant hot clock generation. Resonant adiabatic computing, in Fig. 1(c), recycles energy in an oscillating LC tank where the total on-chip load capacitance, C, is utilized as the tank capacitor. The inductor can be implemented externally or can be distributed over the chip [14]. In each period, the charge stored on C is shifted back into the inductor and is reused in subsequent computations, decreasing the dynamic energy dissipated well below $CVdd^2$. In principle, dynamic energy consumption per unit computation in adiabatic circuits approaches zero with increasing oscillation period. In practice, the energy gain is limited by resistive losses in the tank and variability of load capacitance, which depends on signal activity. Resonant adiabatic arithmetic units [13], [15], [16] and line drivers [17], [18] have been reported with up to seven-fold energy efficiency gains over their non-adiabatic mode.

Some existing adiabatic digital circuits rely on reversible logic [19] to minimize non-adiabatic energy losses [20]. Fully adiabatic circuits [19] require a backward path, where computations are reversed, to recover the energy. The need to reverse information flow places great constraints on what can be computed. For example, an AND gate requires an auxiliary output in order to make the architecture reversible.

Instead of implementing digital adiabatic logic, we perform reversible computing adiabatically in the mixed-signal domain [18]. Reversibility is inherent to the reversal of charge flow between two coupled MOS transistors, shown in Fig. 1(d). Transistors M1 and M2 comprise a charge-injection device (CID) which performs a one-bit multiply-and-accumulate (MAC) operation as detailed in Section II. To maintain high computational throughput, the mixed-signal adiabatic computing is performed on a charge-mode array [21], [22]. This work demonstrates that simple adiabatic techniques such as a resonant single-phase clock generator can be utilized effectively in parallel signal processing applications where array power dissipation often dominates.

There are a number of benefits in the choice of a parallel mixed-signal architecture. While parallel digital processors offer high throughput and energy efficiency with high accuracy [23], parallel analog processors often allow for further increases in integration density, computational throughput, and energy efficiency at the expense of reduced accuracy [24]–[26]. High integration density is achieved by compact analog circuits such as those operating in charge domain. Computational throughput is enhanced by larger dimensions of computing arrays with compact cells and by the low-cost nature of some of analog operations such as zero-latency addition in charge domain. Energy efficiency is increased as clocking is reduced or performed adiabatically as in the case of the presented architecture. Lower accuracy of computation is a result of non-idealities of analog components such as inherent non-linearity and mismatches and is typically only weakly dependent on the dissipated power for a given implementation. A detailed quantitative analysis of the analog-versus-digital trade-off is given in [27]. In targeted applications such as pattern recognition and data classification a modest accuracy of under 8 bits is often sufficient.

The charge-mode computing array presented here is embedded in a processor which performs general purpose vector-matrix multiplication (VMM), the computational core of any template-matching linear transform. The combination of resonant power generation and mixed-signal adiabatic computing on a massively parallel charge-mode array yields a 25-fold gain in energy efficiency relative to the same array operated with static CMOS logic line drivers.

The paper is organized as follows. Section II describes the architecture and circuit implementation of the charge-mode template-matching array. In Section III, a resonant adiabatic clock generator is introduced in order to achieve high energy efficiency of the array-based computation. Limitations of the resonant clock generator are formulated and analyzed. Section IV describes the circuits and VLSI implementation of the resonant adiabatic charge-mode array processor overcoming these limitations. Section V presents experimental results from the adiabatic array processor prototyped in 0.35-$\mu\mathrm{m}$ CMOS technology, and Section VI concludes with final remarks.

II. Charge-Mode Template-Matching Array

The charge-mode array supports general analog multiplication of a digital matrix by a digital vector, by using reversible charge flow between coupled transistors [21], [28], [29]. As shown in Figure 1(d), each cell in the array performs a multiply-accumulation (MAC) operation by selectively transferring charge between two charge-coupled transistors M1 and M2, where the gate of the first transistor M1 connects to the input line, and the gate of the second transistor M2 connects to the output line. Hence M1 implements multiplication by selectively performing or not performing the charge transfer, and M2 implements the accumulation by capacitive coupling onto the output line. The charge transfer is non-destructive, and therefore the computation performed is intrinsically reversible, returning the transferred charge after deactivation of the input. The adiabatic mixed-signal principle outlined here exploits the lossless nature of reversible charge flow in an array of MAC cells, with inputs supplied by adiabatic line drivers from a hot clock supply. The multiplication and accumulation are performed in parallel in a single cycle of the resonant clock, with the energy recycled upon recovery of the charge at the end of the cycle [22].

The resonant generator is critical in achieving high energetic efficiency, and is described in Section III.

A. Array Architecture and Circuit Implementation

The array performs general-purpose vector-matrix multiplication (VMM), the computational core of a variety of linear transform based algorithms in signal processing and pattern recognition. The VMM operation is defined as:

$$Y_m = \mathbf{W}_m \cdot \mathbf{X} = \sum_{n=0}^{N-1} W_{mn} X_n \qquad (1)$$

with N-dimensional input vector $\mathbf{X}$, M-dimensional output vector $\mathbf{Y}$, and $M \times N$ matrix elements $\mathbf{W}$.

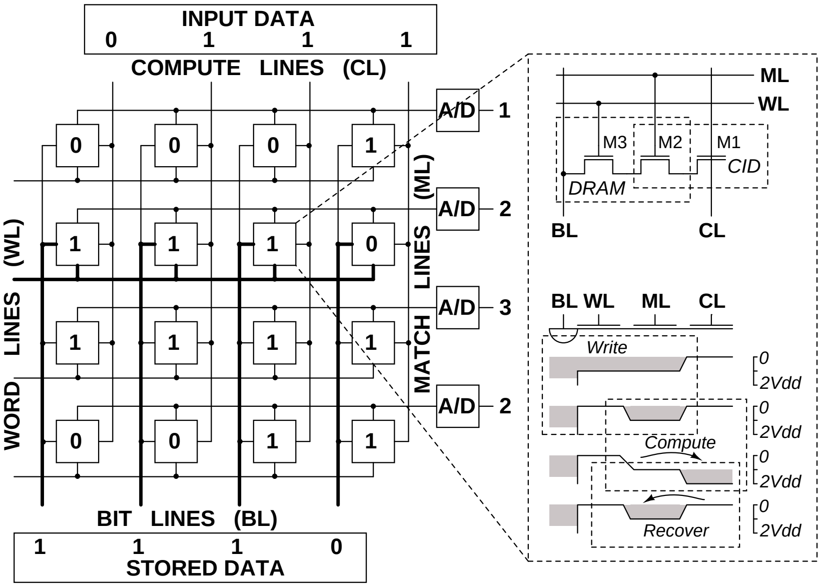


Fig. 2. Array processor architecture (left), circuit diagram of CID computational cell with integrated DRAM storage (right, top), and charge transfer diagram for active write and compute operations (right, bottom). A 1-bit binary data example is shown.

Fig. 2(left) depicts a simplified architecture of the array processor for one-bit binary input vector and matrix coefficients, and matrix dimensions of $M = N = 4$ [22]. The analog array is interfaced with a bank of on-chip row-parallel analog-to-digital converters (ADCs) to provide convenient digital outputs as needed in some applications as well as in the array experimental testing and demonstration.

The unit cell in the analog array shown in Fig. 2(top, right) combines a charge injection device (CID) computational element [28], [29] with a DRAM storage element [21]. During the *write* operation the data to be stored is broadcast on the vertical bit-lines (BLs), which extend across the array. A row to be written to is selected by activating its word-line (WL) turning transistor M3 on (*e.g.*, the second row in Fig. 2). The output match-line (ML) is held at *Vdd* during the *write* phase creating a potential well under the gate of transistor M2. This potential well is filled with electrons or emptied depending on whether the BL is logic-one or logic-zero respectively. Logic-one on BLs corresponds to 0V, while logic-zero corresponds to *Vdd*. During the *compute* operation, the input data is broadcast on the compute-lines (CLs) while MLs, previously precharged to *Vdd*, are now left floating. Logic-one CL bit corresponds to voltage *2Vdd*, while logic-zero corresponds to 0V. Each cell performs a one-quadrant binary-binary multiplication of its stored logic value and its CL logic value. An active charge transfer from M2 to M1 can occur only if there is a non-zero charge stored (*e.g.*, first, second, and third cells in the second row in Fig. 2), and if the potential on the gate of M1 rises above that of M2, to *2Vdd* (*e.g.*, second, third, and fourth columns in Fig. 2). In this case, the high-impedance gate of M2 couples to its channel and raises above *Vdd* by a fixed voltage depending on the charge and capacitance of M2 and the number of active cells in that row (*e.g.*, second and third cells in the second row in Fig. 2). The output of a row is a discrete analog quantity reflecting the number of active cells coupling into the ML of that row (*e.g.*, two cells, second and third, corresponding to the output of the second row equal to two in Fig. 2). In the numerical example given in Fig. 2, the correlation of the binary vector “1110” stored in the second row of the array with the binary input vector “0111” computed by the method described above yields the correct output equal to two.

As said, the cell performs non-destructive computation since the transferred charge is sensed capacitively on the MLs. Once computation is performed, the charge is shifted back from M1 into the DRAM storage transistor M2. Capacitive coupling of all cells in a single row into a single ML implements zero-latency analog accumulation along each row. An array of cells thus performs analog multiplication of a binary matrix with a binary vector. The architecture easily extends to multi-bit data [21].

B. Accuracy and Power Considerations

Sizing of transistors in the cell is of importance. The switch transistor M3 is of minimum size as needed to lower its parasitic capacitance and charge injection. Transistor M2 is 30 times larger than M3 in order to avoid DRAM soft errors, as dictated by the DRAM BL capacitance and by subthreshold leakage in the storage cell. Transistor M1 is sized such that the output dynamic range of the array is large yielding sufficient noise margins. It can be shown that the voltage on MLs is a monotonically increasing saturating function of the area of transistor M1. The area of M1 is chosen to be 50 percent of that of M2. Increasing the area of M1 beyond this value does not yield a substantial increase in the dynamic range but reduces the density of the array and the resonant frequency of the LC tank.

When the computational array is integrated with high-speed digital CMOS circuits on the same chip, excessive interference due to crosstalk may affect the operation of charge-mode cells. The resulting noise may be correlated for many cells and thus may not be averaged out during row-wise accumulation. One way to remove the effect of interference is by utilizing one row in the array as a reference row. This dedicated row has all logic-zero bits stored in it and has the same inputs as all the other rows. The output of the reference row is subtracted from outputs of all rows in a differential fashion in digital domain rejecting any common-mode signals.

Most of the power in the computational array is dissipated on driving CLs. If CLs are driven by conventional CMOS inverters, the power dissipated in the array is proportional to the frequency, array capacitance and the square of the supply voltage. As described in Section I, this power is lost and can not be recovered. To reduce the energy dissipated in the array, instead of being driven by CMOS inverters, all CLs are selectively coupled to an off-chip inductor such that the energy needed for computing can be adiabatically recycled by means of resonance, as described next.

III. Resonant Power Generation

The array capacitance together with an external inductor form an *LC* resonator, driven by an external clock CLK at resonance frequency to generate the hot clock power supply waveform $V_{HC}(t)$ in Fig. 1(d). Resistive losses in the adiabatic line drivers of the charge-mode array are minimized by keeping hot clock oscillation frequency sufficiently low. High computational throughput is nevertheless maintained by a fine-grain parallel architecture of the processor. The massive parallelism also allows to maintain the on-chip load capacitance at or near its mean value, tuned at resonance where the energy dissipation in the tank is lowest.

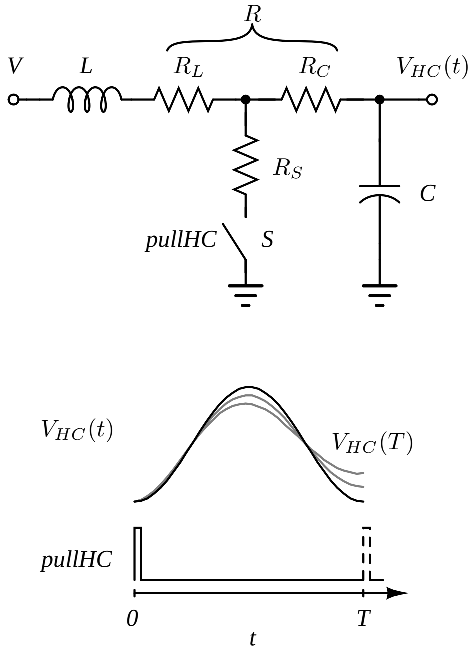


Fig. 3. Lossy *LC* oscillator and switch with fixed load capacitance C.

The efficiency of resonant power generation is thus limited by resistive losses in the tank and variability of on-chip load capacitance. Each of these limitations is analyzed next.

A. Tank Resistive Losses

A simple model of a constant-capacitance *LC* oscillator used to generate the hot clock voltage, $V_{HC}(t)$, is shown as an *RLC* circuit in Fig. 3, where C is the load capacitance implied by the charge-mode array, L is the tank inductor, and R represents parasitic resistive losses in the tank. The tank resistance $R = R_L + R_C$ decomposes into two contributions: parasitic resistance in the inductor R_L due its finite quality factor $Q_L = \omega L/R_L$; and parasitic resistance in the capacitor R_C accounting for non-zero on-resistance of the adiabatic line drivers represented by the IN switch in Fig. 1(d). The parasitic shunt resistance R_S accounts for non-zero on-resistance of the switch S, when it is activated. The switch S is used to initiate and maintain oscillations by periodically discharging C to ground. It is activated by a narrow pulse *pullHC*. The step response of the *RLC* circuit with small damping factor (in the limit for small R) is of the form:

$$V_{HC}(t) = V\left[1 - e^{-\frac{R}{2L}t}\cos\left(\frac{1}{\sqrt{LC}}t\right)\right]. \quad (2)$$

Switch S dissipates energy if the voltage across the capacitor is non-zero when it goes active. This energy

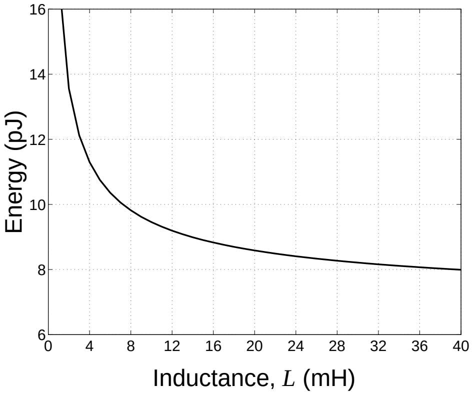


Fig. 4. Energy dissipation asymptotically approaching a finite non-zero value determined by the quality of inductor.

is minimized by pulsing *pullHC* at the minima of the *LC* tank voltage, and thus at the resonant frequency $f = (2\pi\sqrt{LC})^{-1}$, as shown in Fig. 3 for $T = 1/f$.

Resistive losses in R cause minima of the voltage $V_{HC}(T)$ at the next *pullHC* pulse to be non-zero as described by the exponential envelope:

$$V_{HC}(T) = V\left(1 - e^{-\pi R\sqrt{\frac{C}{L}}}\right).$$

Assuming a constant value of C, the dynamic energy, $\frac{1}{2}CV_{HC}^2(T)$, dissipated in each computation cycle is thus given by:

$$E|_{C=const} = \frac{1}{2}C\left[V\left(1 - e^{-\pi R\sqrt{\frac{C}{L}}}\right)\right]^2. \quad (3)$$

The choice of a minimum capacitance value is obvious. As for inductance, in theory, for a given load capacitance C, the dynamic energy dissipation can be made arbitrarily small by increasing L as is evident from (3). In practice, the dynamic energy dissipation asymptotically approaches a finite value, determined by the quality of the inductor, as the parasitic resistance of a wire-wound inductor $R_L \propto \sqrt{L}$ dominates the total resistance R for large L. [1] Thus increasing L beyond a certain level may not be justifiable as it yields diminishing reduction in energy dissipation as shown in Fig. 4 but results in a lower oscillation frequency and thus lower throughput.

B. Switch Resistance and Pulse Width

The shunt resistance of the switch R_S, to first order, does not contribute losses and does not affect the efficiency of the hot clock supply generator, provided that R_S is sufficiently small. During the duration of the *pullHC* pulse, the load capacitance C is discharged through the series resistor combination $R_C + R_S$. Incomplete settling in this RC network implies incomplete compensation of the exponential decay in the sinusoidal hot clock waveform, leading to a reduced amplitude hot clock and further resistive losses. A sufficiently small value of $R_C + R_S$, and sufficiently large pulse duration *pullHC* ensures that V_{HC} settles close to zero.

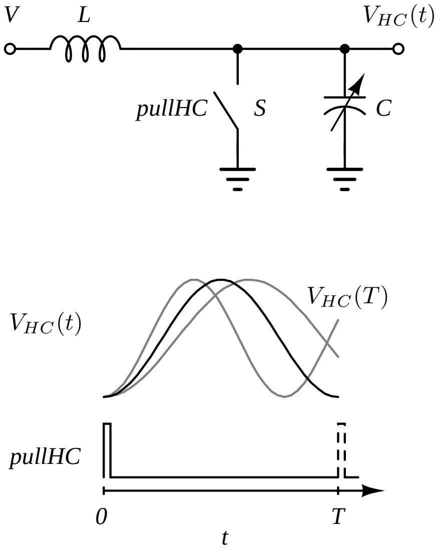


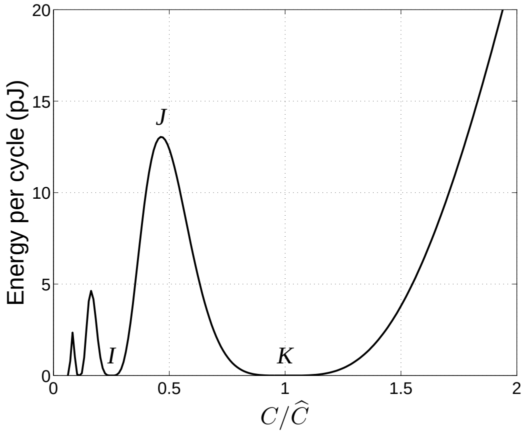
Fig. 5. Lossless *LC* oscillator and switch with variable load capacitance *C*.

As the sine wave of $V_{HC}(t)$ is approximately quadratic around its minima, the energy in (3) is insensitive to the pulse width variation of *pullHC* near its minima. This allows for a relatively large pulse width resulting in small energy losses. For larger pulse widths, the current through the inductor significantly affects the resonant clock waveform which extends outside the *2Vdd* interval and exerts extra energy losses [30].

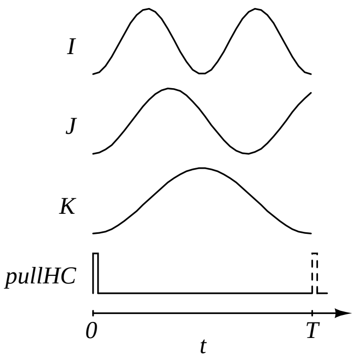
C. Load Capacitance Variability

A simple model of a variable-capacitance *LC* oscillator is shown in Fig. 5, where C has a mean value of $\widehat{C}$, and resistive losses as modeled in Section III-A are here assumed zero for simplicity. Signal *pullHC* is pulsed at the *LC* tank mean-capacitance resonant frequency $f = (2\pi\sqrt{L\widehat{C}})^{-1}$. However the instantaneous *LC* tank resonant frequency, $(2\pi\sqrt{LC})^{-1}$, depends on the load capacitance C, causing the pulse *pullHC* activating S to miss the minima of the oscillations when C deviates from $\widehat{C}$ as shown in Fig. 5. Substituting $t = T = 2\pi\sqrt{L\widehat{C}}$ into (2) and ignoring resistive losses yields the instantaneous voltage on C just before it is

[1] For an integrated, spiral-wound inductor, the dynamic energy dissipation increases for large L.



(a)



(b)

Fig. 6. (a) Dynamic energy dissipation in switch *S* of a lossless varying-capacitance *LC* oscillator, and (b) corresponding $V_{HC}(t)$ waveforms.

discharged to ground by S:

$$V_{HC}(T) = V\left[1 - \cos\left(2\pi\sqrt{\frac{\widehat{C}}{C}}\right)\right].$$

The dynamic energy $\frac{1}{2}CV_{HC}^2(T)$ dissipated each computation cycle,

$$E|_{R=0} = \frac{1}{2}C\left(V\left[1 - \cos\left(2\pi\sqrt{\frac{\widehat{C}}{C}}\right)\right]\right)^2, \quad (4)$$

is plotted in Fig. 6(a), along with the corresponding hot clock waveforms in Fig. 6(b). When $C = \widehat{C}$ (case *K*) or $C = \widehat{C}/4$ (case *I*), $V_{HC}(t)$ completes one or two full oscillation(s), respectively, before *pullHC* is pulsed so no energy is dissipated in *S*. At the minimum point with the widest concavity region, the dynamic energy

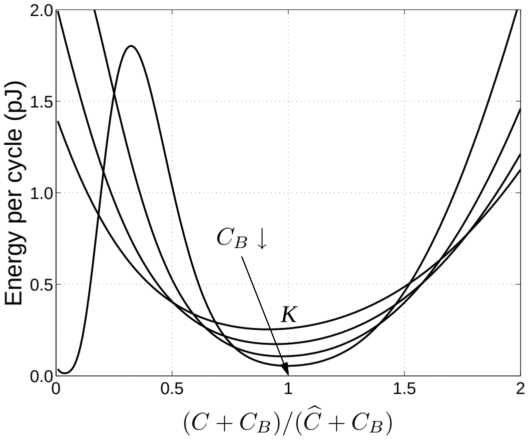


Fig. 7. Increasing the bypass capacitor C_B desensitizes dynamic energy dissipation to *C* variation.

dissipation approaches zero as the load capacitance *C* approaches its mean $\widehat{C}$ (point *K* in Fig. 6).

Adding an external bypass capacitor, C_B, in parallel with C increases the total load capacitance to $C + C_B$. The addition of sufficiently large C_B attenuates the effect of capacitive variations in the array on oscillation frequency and hence energy dissipation. In theory, without resistive losses, such oscillator would always operate at its ideal point, point *K* in Fig. 6, and dissipate zero energy. In practice, the energy dissipation due to *both* resistive losses and *C* variation must be considered:

$$E = \frac{1}{2}C\left(V\left[1 - e^{-\pi R\sqrt{\frac{\widehat{C}}{L}}}\cos\left(2\pi\sqrt{\frac{\widehat{C}}{C}}\right)\right]\right)^2. \quad (5)$$

As shown in Fig. 7, adding C_B desensitizes the dynamic energy dissipation to C variation at the cost of increasing the resistive energy dissipation. Thus an external capacitor was not utilized in this design.

D. Parallel Architecture

As shown in Section III-A and in Fig. 4, the resistive losses can be reduced by increasing inductance, which also reduces oscillation frequency. In order to maintain high computational throughput, a parallel array-based architecture is needed to perform large numbers of operations each clock cycle. Furthermore, as shown in Section V, data-dependent load statistics over large numbers of inputs in the array allow to maintain the array load capacitance at or near a constant value (at point *K* in Fig. 6) with approximately half of all cells active at any time. This minimizes dynamic losses not only in the array, but also in the resonant clock generator as shown in Section III-C. Next, we present

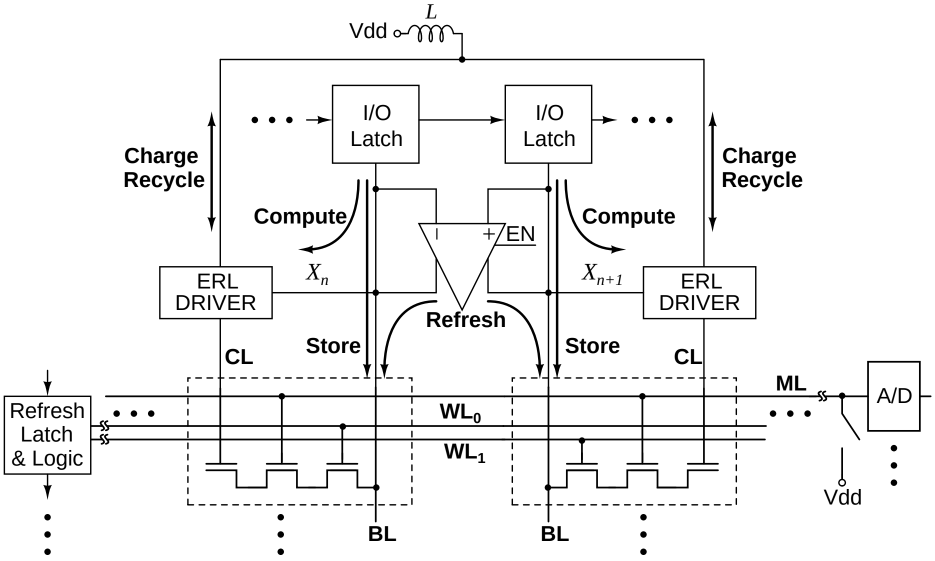


Fig. 8. Circuit diagram of functions peripheral to the cell including *store*, *refresh* and *charge-recycling adiabatic compute*.

a massively-parallel array first introduced in Section II that implements mixed-signal resonant adiabatic computing over large numbers of charge-coupled transistor pairs as shown in Figure 1(d).

IV. Adiabatic Array Processor

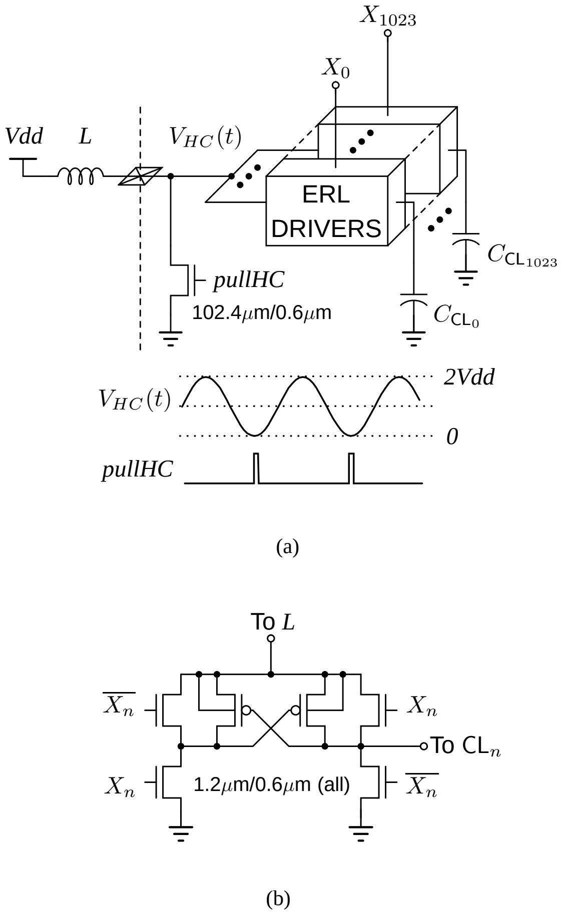
The hot clock supply generator sees the array of MAC cells as a variable load capacitance C as in Fig. 5, where the variation in C is implied by variations in input. As demonstrated further below, these variations are kept at a minimum by virtue of the parallel nature of the computation. The architecture, circuits and implementation of the processor are described next.

A. Circuits

Fig. 8 shows the block diagram of the array peripheral functions with signal paths for *store*, *refresh*, *compute* and *charge recycle* functions marked [22]. Two columns, $n - th$ and $(n+1) - th$, of the first row are shown. Matrix coefficients are loaded into the dynamic random access memory (DRAM) from a shift register in the *store* phase. The CID/DRAM cells on folded bit-lines (BLs) are periodically refreshed after several compute cycles, alternating between even and odd columns with separate word-lines (WLs). In the *compute* cycle the input data, X_n, $n = 0, .., N-1$, enable adiabatic energy recovery logic (ERL) drivers [13]. They conditionally connect the off-chip off-the-shelf inductor to the on-chip capacitance of active compute-lines (CLs) to enable *charge recycling* through resonance.

The capacitance of all active CLs is utilized to perform adiabatic computing on the full array as schematically shown in Fig. 9(a). The *LC* tank is replenished with external energy from the DC voltage source *Vdd* by pulsing *pullHC* at the minima of voltage waveform $V_{HC}(t)$. A doubled dynamic range of *2Vdd* is thus obtained. Signal *pullHC* also serves to synchronize the hot clock waveform to other circuits in the processor.

The choice of the frequency of signal *pullHC* is important. As discussed in Section III-C, variations in the total CLs capacitance cause the frequency of tank oscillation to deviate from that of signal *pullHC* $f = 1/T$ resulting in additional energy losses. One solution to this problem is to use differential coding of data, with complementary inputs and complementary stored coefficients. This ensures that exactly half of all CLs are connected to the inductor. The capacitance of each CID/DRAM cell is approximately identical, regardless of whether charge is stored as determined by the binary matrix element value. This invariance owes to the fact that transistor M1 operates either in strong inversion or accumulation mode, with approximately same gate capacitance. Thus, by ensuring that always half of all CLs are active, the array capacitance is kept constant. This approach, however, requires twice the number of cells and thus doubles the silicon area. Instead, we observe that in typical data, such as images, the probability of a binary coefficient being zero or one is approximately half for most of the coefficients. This implies that the number of logic-one bits in the input vector is typically approximately half. In the Central Limit, the number of logic-one bits in



(a)

(b)

Fig. 9. (a) Resonant clock generator for adiabatic power supply, and (b) input-enabled energy recovery logic (ERL) driver.

an N-dimensional binary vector follows a binomial distribution approximated by a Gaussian distribution with mean $N/2$ and standard deviation $\sqrt{N}$. Hence the relative width of the distribution tends to zero for large N. This property of typical data is exploited here in order to minimize energy losses in the *LC* tank due to array capacitance variability as validated in Section V. For applications where many binary coefficients are non-Bernoulli, we have developed a simple stochastic data modulation scheme to pseudo-randomize input data with any statistics at the expense of a small modulation and demodulation overhead [31].

The circuit diagram of a modified ERL driver is shown in Fig. 9(b). When the input vector component bit, X_n, is logic-one, the corresponding compute-line, CL$_n$, is connected to the inductor through a pass gate. A pass gate is utilized in order to realize an energy-efficient fully-adiabatic driver. As the maximum voltage on the inductor is *2Vdd*, while the logic-one level of X_n is *Vdd*, a cross-coupled PMOS transistor pair ensures that the pass gate is turned off completely when X_n is low. High-voltage devices are used to accommodate the doubled dynamic range. The signal *pullHC* is synchronized with the clocks for all peripheral circuits, generated from the same master clock. Tuning of the resonance condition is achieved either by tuning of the master clock frequency, or by adjusting the value of the external inductance. Integrating adaptive mechanisms for tuning may further reduce power dissipation, especially for highly variable data or for off-the-shelf inductors with a large spread of values, but with a potentially significant overhead.

The transistor sizes shown in Figs. 9(a) and 9(b) are determined as follows. Referring to Fig. 3, R_S corresponds to the resistance of the nMOS switch driven by signal $pullHC$ in Fig. 9(a), and R_C corresponds to the parallel combination of the ERL drivers on-resistance in Figs. 9(a) and 9(b). The value of R_S is chosen such that the circuit operates at an optimum point where the resistance of the switch driven by $pullHC$ is small enough to keep resistive losses in it small, and the capacitance of its gate is small enough to keep the energy needed to drive it small. In this design, $R_S = 10\Omega$ and the gate capacitance of the switch is $270fF$. To minimize resistive losses, R_C has to be small, but there is little benefit in making it much smaller than R_S. The sizing shown in Fig. 9(b) yields the average resistance of the pass gate in an ERL driver of less than $5k\Omega$. With approximately half of the ERL drivers active for typical inputs (see below), the corresponding value for R_C is less than 10Ω as needed to balance losses in R_C and R_S under silicon area constraints.

B. Implementation

The integrated prototype of the mixed-signal adiabatic vector-matrix multiplication (VMM) processor depicted in Fig. 10 occupies 4×4 mm^2 in 0.35-μm CMOS. The processor consists of four self-contained cores. Each core contains 128×256 CID/DRAM computational storage elements, a row-parallel bank of 128 8-bit $\Delta\Sigma$ algorithmic analog-to-digital converters (ADCs) [21], pipelined input shift registers, sense amplifiers, refresh logic, and scan-out logic. All of the supporting digital clocks and control signals are generated on the chip. The modular architecture allows the four cores to operate in 1×4, 2×2, and 4×1 configurations to compute 128×1024, 256×512, and 512×256 dimensional binary vector-matrix products respectively. This flexibility is necessary in implementing linear transforms with various input and output dimensions.

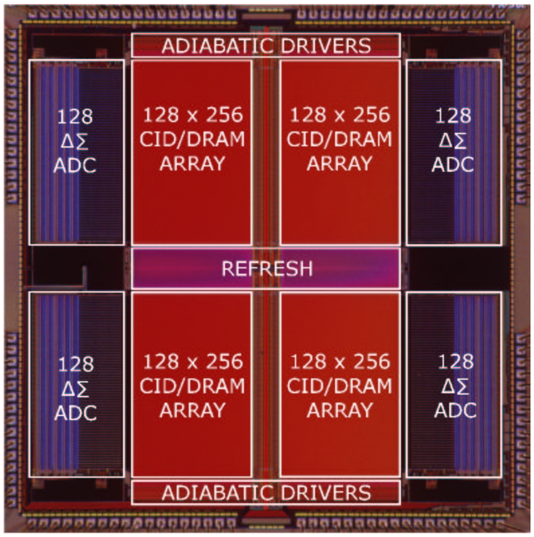


Fig. 10. Adiabatic VMM processor micrograph and floorplan. Fabricated in a standard 0.35-μm CMOS process, the processor occupies 4×4 mm^2.

V. Experimental Results

The processor functionality was validated in a template-based face detection application. Real-time detection of objects such as faces on a low-power wearable platform allows to implement miniature visual aids for the blind. Template-based pattern recognition is computationally expensive as it requires matching of each input with a set of characteristic templates. The parallel processing architecture lends itself naturally to such an application. A pattern recognition engine was trained off-line on a face recognition data set distributed by the Center for Biological and Computational Learning (CBCL) at MIT. [2] The classifier was then programmed on the processor with visual templates stored in the CID/DRAM array. Inner-product based similarities between each input and all templates were computed on the array. Both inputs and templates are 11×11 pixel image segments. Experimentally, we validated that the processor produces classification results on an out-of-class test set that are identical to those obtained by emulation in software, testifying to the robustness of the architecture and circuit implementation. For this task, perfect classification was obtained. A few examples of the correct classifications of faces and non-faces by the processor are given in Fig. 11.

Fig. 12 shows typical statistics of images from the MIT CBCL face data set. Most of natural scene images have binary coefficients which are equally probable (Bernoulli distributed, $P(0) = P(1) = 0.5$). This implies that the sum of logic-one bits in a fragment of a typical image (the same as the number of logic-one bits) follows a normal (Gaussian) distribution with low variance. Over 95 percent of the data in Fig. 12 fall within less than 18 percent of the entire input range, within two standard deviations of the mean. Points labeled μ, -2σ, and $+2\sigma$ are fitted parameters based on an ideal normal distribution and mark the mean and two standard deviations spread, evaluated over the face data set. The corresponding experimentally measured hot clock waveforms are shown on the top of Fig. 12. The hot clock oscillates at a frequency determined by the number of compute-lines (CLs) connected to the external inductor plus all the parasitic capacitance in the hot clock path. The *pullHC* signal frequency and its duty-cycle are tuned to coincide with the minima of the hot clock oscillations when half of the inputs, X_n, are active. As input data deviates from this mean, *pullHC* misses the minimum voltage point in discharging the tank capacitor, increasing the dynamic energy dissipation.

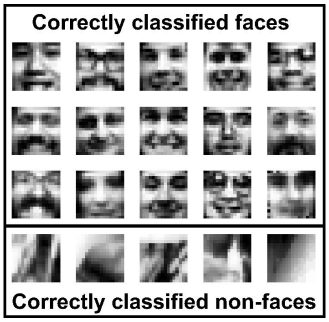


Fig. 11. Examples of faces and non-faces correctly classified by the prototyped VMM processor from a face detection experiment.

Fig. 13(a) shows the experimental setup utilized for measuring power consumption of the array. The array is configured to operate in one of the two modes, adiabatic and static, for comparative purposes. DC current delivered by the DC power supply is measured in each case. In the adiabatic mode, each active CL is driven by the hot clock through the pass gate of an energy recovery logic (ERL) driver. The product of measured average current through the DC supply and its voltage Vdd represents the total measured power which includes the losses in the resonant tank supply generator, implemented using an external wire-wound inductor, as well as in the ERL drivers and CID/DRAM array. Power dissipated to generate and drive the signal $pullHC$ in the adiabatic mode is small compared to the power dissipated in the clock generator and the array [30].

In the static mode, the CLs are driven by an external

[2] http://cbcl.mit.edu/software-datasets/index.html

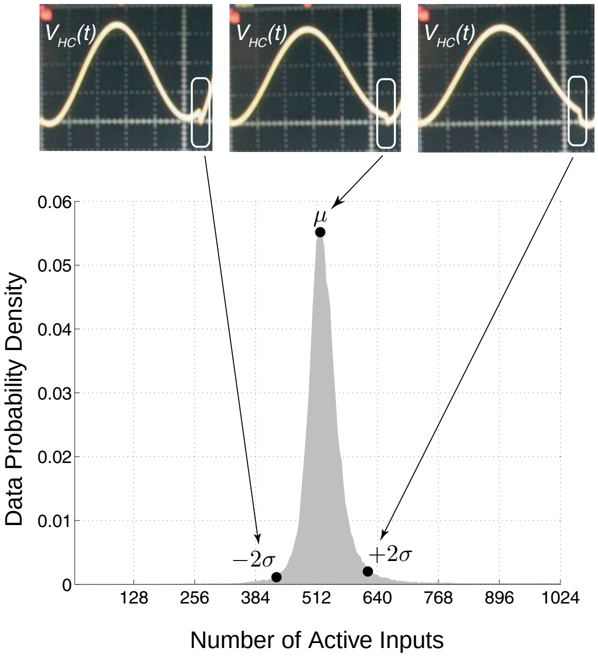


Fig. 12. Probability density of the number of active inputs for the MIT CBCL face data (bottom) and corresponding experimentally measured hot clock waveforms (top). The nominal hot clock frequency is 13.7kHz. The peak-to-peak voltage amplitude is 3.3V with 1.65V power supply.

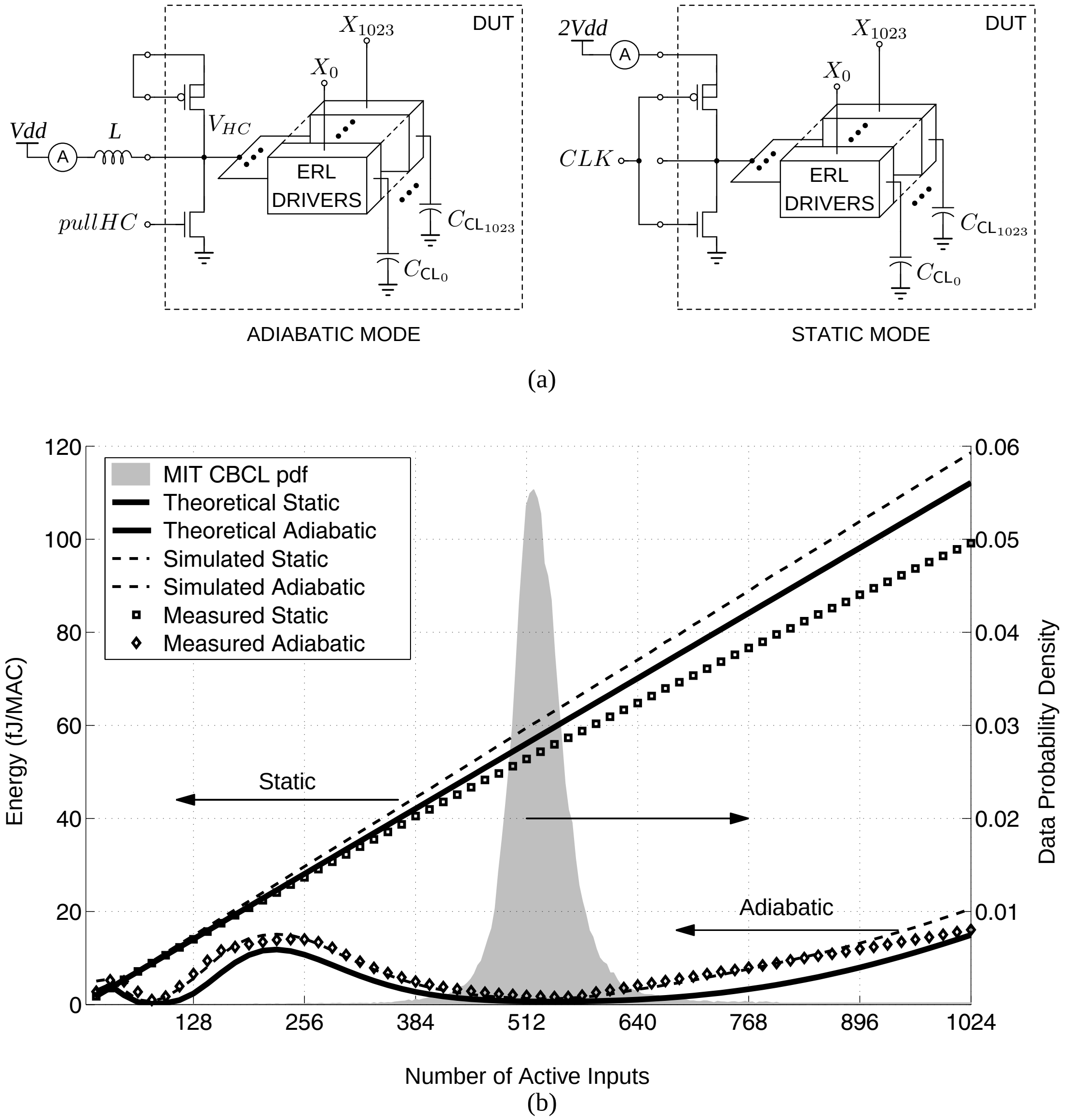
digital signal CLK through CMOS inverters (only one inverter is shown for simplicity), with ERL drivers functioning as static CMOS buffers, as shown in Fig. 13(a). In this mode the inductor is shorted and the value of the supply voltage is increased to $2Vdd$ to yield the same voltage swing. The power in the static mode is measured as the product of the average current through the CMOS inverters as shown, multiplied by the DC voltage $2Vdd$ supplying this current. Power dissipated to generate and drive the signal CLK in the static mode is similar to that for the signal $pullHC$ in the adiabatic mode. Both are small and thus are omitted from the comparative analysis.

Fig. 13(b) shows energy consumption per computation of the CID/DRAM array in the static mode and in the adiabatic mode as a function of the number of active inputs (number of logic-one bits in the input vector). Theoretical, simulated and experimentally measured results are plotted. The experimental data were measured utilizing the testing setup depicted in Fig. 13(a). In the static mode, as expected the energy consumption per computation is a linear function of the total capacitance of active CLs. In the adiabatic mode, the energy consumption per computation is a non-monotonic function of the the total capacitance of active CLs matching that described by Eqn. (5) and shown in Figs. 6(a) and 7.

The probability density distribution of the number of active inputs for the MIT CBCL face data set is also shown in Fig. 13(b). For the MIT CBCL face data set the adiabatic processor yields experimentally measured computational energy efficiency of 480 GMACS/mW. This number is obtained by multiplying the measured energy efficiency of the array by the corresponding MIT CBCL face data probability density function for each number of active inputs and adding the results together. For the same data, the processor yields energy efficiency of 19 GMACS/mW when configured in the static mode. This corresponds to a 25-fold improvement in energy efficiency. The processor performs 128×256 binary multiply-and-accumulate operations on each of the four arrays corresponding to 1.8 GMACS computational throughput at 13.7 kHz hot clock frequency.

Contributions of subthreshold leakage, junction leakage or gate tunneling to overall power dissipated in the array are insignificant. Scaling the design to deep submicron technologies may require additional design considerations such as negative voltage gate biasing and low-voltage junction biasing. In general, compared to high-speed digital designs, low power dissipation of the array maintains lower temperature of the die and thus lower leakage currents.

Not included in the MAC array and supply generator power is the power dissipated in the ADCs, and other peripheral functions such as shift registers which can be efficiently implemented using conventional digital adiabatic design techniques. The bank of 512 ADCs [21] including non-adiabatic clock generators measures 6.3mW of power dissipation from a 3.3V supply, at 15kHz parallel sample rate. Even though this ADC design yields adequate energy efficiency of 3.2 pJ per sample per quantization level, this power level is orders of magnitude larger than that of the adiabatic array and resonant supply. In the present prototype the ADCs were included for convenience of characterization. For applications requiring quantized outputs, the challenge is to extend the mixed-signal adiabatic VMM principle to implement adiabatic analog-to-digital conversion. Possible directions for adiabatic ADC design are charge-redistribution ADCs [32] or charge-based folding ADCs [33]. Other applications in pattern classification, such as vector quantization or nearest neighbor classification, call for winner-take-all (WTA) or rank-ordered selection of best template matches. WTA selection is efficiently implemented using a cascade of comparators, and potentially adiabatically implemented in the charge domain [34].



(a)

(b)

Fig. 13. (a) Experimental setup for measuring supply current, and corresponding power consumption of the array in adiabatic and static modes. (b) Theoretical, simulated and experimentally measured energy consumption per computation cycle of the array as a function of input data statistics in the adiabatic mode and in the static mode. The MIT CBCL face data set statistics are shown in gray.

The measured adiabatic VMM processor characteristics are summarized in Table I.

VI. Conclusion

We have shown that an array of simple multiply-and-accumulate cells, consisting of charge-coupled transistor pairs, constitutes a virtually lossless capacitive load to a resonant hot clock generator, leading to significant (25-fold) savings in energy efficiency over a lossy driven system where adiabatic line drivers are replaced with CMOS logic drivers. The 4mm×4mm, 512×256-cell array in 0.35-μm CMOS delivers 480 GMACS (4.8×10^{11} multiply-and-accumulates per second) for every mW of power.

Minimum energy dissipation requires low-resistance line drivers, but does not require low-resistance switching in the resonant supply for a reasonably shaped, low duty cycle clock signal. Minimum energy also requires minimum variability in the capacitive load, which is ensured owing to the statistics of inputs controlling charge transfer in a large array of MAC cells.

The adiabatic array and resonant supply generator was embedded in a vector-matrix multiplication processor and demonstrated on a face detection task,

TABLE I
MEASURED CHARACTERISTICS

Technology	0.35 μm CMOS
Supply Voltage	1.65V
Die Area	4×4 mm^2
Array Area	2.7×1.8 mm^2
CID/DRAM Cell Area	9.9×3.6 μm^2
CID/DRAM Cell Count	131,072
Throughput	1.8 GMACS at 13.7 kHz
Array Energy Efficiency	19 GMACS/mW Static mode
	480 GMACS/mW Adiabatic mode
Output Resolution	8 bits
Column mismatch	± 1 LSB for 97% of columns

with stored coefficients obtained by off-line training over example data. Further research is directed towards implementing ADC quantization or WTA selection in the adiabatic domain [33], [34] for a complete adiabatic mixed-signal system-on-chip. Applications include pattern recognition [22], data compression [23] and CDMA matched filters [26].

REFERENCES

[1] R. McGowen, C. A. Poirier, C. Bostak, J. Ignowski, M. Millican, W. H. Parks, and S. Naffziger, "Power and temperature control on a 90nm Itanium family processor," *IEEE J. Solid-State Circuits*, vol. 41, no. 1, pp. 229–237, Jan. 2006.

[2] B. H. Calhoun and A. P. Chandrakasan, "Ultra-dynamic voltage scaling (UDVS) using sub-threshold operation and local voltage dithering," *IEEE J. Solid-State Circuits*, vol. 41, no. 1, pp. 238–245, Jan. 2006.

[3] ——, "A 256kb sub-threshold SRAM in 65nm CMOS," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2006, pp. 628–629.

[4] A. Wang and A. P. Chandrakasan, "A 180mV subthreshold FFT processor using a minimum energy design methodology," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, Jan. 2005.

[5] J. Kao, M. Miyazaki, and A. P. Chandrakasan, "A 175mv multiply-accumulate unit using an adaptive supply voltage and body bias architecture," *IEEE J. Solid-State Circuits*, vol. 37, no. 11, pp. 1545–1554, Nov. 2002.

[6] W. C. Athas, J. G. Koller, and L. J. Svensson, "An energy-efficient CMOS line driver using adiabatic switching," in *Proc. IEEE Fourth Great Lakes Symposium on Design Automation of High Performance VLSI Systems*, Mar. 1994, pp. 196–199.

[7] A. G. Dickinson and J. S. Denker, "Adiabatic dynamic logic," *IEEE J. Solid-State Circuits*, vol. 30, no. 3, pp. 311–315, Mar. 1995.

[8] Y. Moon and D.-K. Jeong, "An efficient charge recovery logic circuit," *IEEE J. Solid-State Circuits*, vol. 31, no. 4, pp. 514–522, Apr. 1996.

[9] V. G. Oklobdzija, D. Maksimovic, and L. Fengcheng, "Pass-transistor adiabatic logic using single power-clock supply," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 10, pp. 842 – 846, Oct. 1997.

[10] D. Maksimovic, V. G. Oklobdzija, B. Nikolic, and K. Current, "Clocked CMOS adiabatic logic with integrated single-phase power-clock supply," *IEEE Trans. VLSI Systems*, vol. 8, no. 4, pp. 460–463, Aug. 2000.

[11] K. Suhwan and M. C. Papaefthymiou, "True single-phase adiabatic circuitry," *IEEE Trans. VLSI Systems*, vol. 9, no. 1, pp. 52–63, Feb. 2001.

[12] W. C. Athas, L. J. Svensson, and N. Tzartzanis, "A resonant signal driver for two-phase, almost-non-overlapping clocks," in *Proc. IEEE Int. Symp. on Circuits and Systems*, vol. 4, May 1996, pp. 129–132.

[13] W. C. Athas, N. Tzartzanis, W. Mao, L. Peterson, R. Lal, K. Chong, J.-S. Moon, L. J. Svensson, and M. Bolotski, "The design and implementation of a low-power clock-powered microprocessor," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1561–1570, Nov. 2000.

[14] S. C. Chan, K. L. Shepard, and P. J. Restle, "Distributed differential oscillators for global clock networks," *IEEE J. Solid-State Circuits*, vol. 41, no. 9, pp. 2083–2094, 2006.

[15] E. Amirante, J. Fischer, M. Lang, A. Bargagli-Stoffi, J. Berthold, C. Heer, and D. Schmitt-Landsiedel, "An ultra low-power adiabatic adder embedded in a standard 0.13-um CMOS environment," in *Proc. IEEE European Solid-State Circuits Conf.*, Sept. 2003, pp. 599–602.

[16] K. Suhwan, C. H. Ziesler, and M. C. Papaefthymiou, "A true single-phase 8-bit adiabatic multiplier," in *Proc. IEEE Design Automation Conf.*, 2001, pp. 758–763.

[17] H. Yamauchi, H. Akamatsu, and T. Fujita, "An asymptotically zero power charge-recycling bus architecture for battery-operated ultra-high data rate ULSI's," *IEEE J. Solid-State Circuits*, vol. 30, no. 4, pp. 423–431, Apr. 1995.

[18] M. Amer, M. Bolotski, P. Alvelda, and T. Knight, "160x120 pixel liquid-crystal-on-silicon microdisplay with an adiabatic DAC," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 1999, pp. 212–213.

[19] C. H. Bennett and R. Landauer, "The fundamental physical limits of computation," *Scientific American*, vol. 253, no. 1, pp. 38–46, 1985.

[20] J. Lim, K. Kwon, and S.-I. Chae, "Reversible energy recovery logic circuit without non-adiabatic energy loss," *Electronics Letters*, vol. 34, no. 4, pp. 344–346, Feb. 1998.

[21] R. Genov, G. Cauwenberghs, G. Mulliken, and F. Adil, "A 5.9 mW 6.5 GMACs CID/DRAM array processor," in *Proc. IEEE European Solid-State Circuits Conf.*, Sept. 2002.

[22] R. Karakiewicz, R. Genov, A. Abbas, and G. Cauwenberghs, "175 GMACS/mW charge-mode adiabatic mixed-signal array processor," in *Proc. IEEE Symposium on VLSI Circuits*, June 2006, pp. 126–127.

[23] A. Nakada, T. Shibata, M. Konda, T. Morimoto, and T. Ohmi, "A fully parallel vector-quantization processor for real-time motion-picture compression," *IEEE J. Solid-State Circuits*, vol. 34, no. 6, pp. 822–830, 1999.

[24] A. Kramer, "Array-based analog computation," *IEEE Micro*, vol. 16, no. 5, pp. 40–49, Oct. 1996.

[25] T. Shibata, T. Nakai, N. M. Yu, Y. Yamashita, M. Konda, and T. Ohmi, "Advances in neuron-MOS applications," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 1996, pp. 304–305.

[26] T. Yamasaki, T. Nakayama, and T. Shibata, "A low-power and compact CDMA matched filter based on switched-current technology," *IEEE J. Solid-State Circuits*, vol. 40, no. 4, pp. 926–932, Apr. 2005.

[27] R. Sarpeshkar, "Analog versus digital: Extrapolating from electronics to neurobiology," *Neural Computation*, vol. 10, pp. 1601–1638, 1998.

[28] C. Neugebauer and A. Yariv, "A parallel analog CCD/CMOS neural network IC," in *Proc. IEEE Int. Joint Conference on Neural Networks*, vol. 1, Seattle, WA, 1991, pp. 447–451.

[29] V. Pedroni, A. Agranat, C. Neugebauer, and A. Yariv, "Pattern matching and parallel processing with CCD technology," in *Proc. IEEE Int. Joint Conference on Neural Networks*, vol. 3, June 1992, pp. 620–623.

[30] D. Maksimovic and V. G. Oklobdzija, "Integrated power clock generators for low energy logic," in *Proc. IEEE Power Electronics Specialists Conference*, 1995, pp. 61–67.

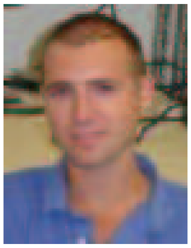
[31] R. Karakiewicz, R. Genov, and G. Cauwenberghs, "1.1 TMACS/mW load-balanced resonant charge-recycling array

processor," in *Proc. IEEE Custom Integrated Circuits Conference*, 2007.

[32] R. Suarez, P. Gray, and D. Hodges, "Charge redistribution analog-to-digital conversion techniques – Part II," *IEEE J. Solid-State Circuits*, vol. SC-10, no. 6, pp. 379–385, Dec. 1975.

[33] R. Genov and G. Cauwenberghs, "Dynamic MOS sigmoid array folding analog-to-digital conversion," *IEEE Trans. Circuits and Systems I,* vol. 51, no. 1, pp. 182–186, Jan. 2004.

[34] K. Kotani and T. Ohmi, "Feedback charge-transfer comparator with zero static power," in *IEEE Solid-State Circuits Conference*, Feb. 1999, pp. 328–329.



Rafal Karakiewicz (SM'03) received the B.A.Sc. and the M.A.Sc. degrees in Electrical Engineering from the University of Toronto, ON in 2003 and 2006 respectively. He is currently a design engineer at SNOWBUSH Microelectronics, Toronto, ON, Canada.

Roman Genov (SM'96-M'02) received the B.S. degree in Electrical Engineering from Rochester Institute of Technology, NY in 1996 and the M.S.E. and Ph.D. degrees in Electrical and Computer Engineering from Johns Hopkins University, Baltimore, MD in 1998 and 2003 respectively.

Dr. Genov held engineering positions at Atmel Corporation, Columbia, MD in 1995 and Xerox Corporation, Rochester, NY in 1996. He was a visiting researcher in the Laboratory of Intelligent Systems at Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland in 1998 and in the Center for Biological and Computational Learning at Massachusetts Institute of Technology, Cambridge, MA in 1999. He is presently an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Toronto, Canada.

Dr. Genov's research interests include analog and digital VLSI circuits, systems and algorithms for energy-efficient signal processing with applications to electrical, chemical and photonic sensory information acquisition, biosensor arrays, neural interfaces, parallel signal processing, adaptive computing for pattern recognition, and implantable and wearable biomedical electronics.

He received Canadian Institutes of Health Research (CIHR) Next Generation Award in 2005, and Dalsa Corporation Componentware Award in 2006. He served as a technical program co-chair of IEEE Conference on Biomedical Circuits and Systems in 2007. He serves on the Advisory Board of the Department of Electrical and Computer Engineering at Rochester Institute of Technology, Rochester, NY. He is an Associate Editor of IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS.

Gert Cauwenberghs (SM'89-M'94-S'04) received the M.Eng. degree in applied physics from University of Brussels, Belgium, in 1988, and the M.S. and Ph.D. degrees in electrical engineering from California Institute of Technology, Pasadena, in 1989 and 1994.

He is Professor of Biology at University of California San Diego, La Jolla, where he directs the Integrated Systems Neuroscience Laboratory. Previously, he held positions as Professor of Electrical and Computer Engineering at Johns Hopkins University, Baltimore Maryland, and as Visiting Professor of Brain and Cognitive Science at Massachusetts Institute of Technology, Cambridge.

Dr. Cauwenberghs' research aims at advancing silicon adaptive microsystems to understanding of biological neural systems, and to development of sensory and neural prostheses and brain-machine interfaces. His activities include design and development of micropower analog and mixed-signal systems-on-chips performing adaptive signal processing and pattern recognition.

He is a Francqui Fellow of the Belgian American Educational Foundation, and received the National Science Foundation Career Award in 1997, Office of Naval Research Young Investigator Award in 1999, and Presidential Early Career Award for Scientists and Engineers in 2000. He serves on the Technical Advisory Board of GTronix, Inc., Fremont CA. He was Distinguished Lecturer of the IEEE Circuits and Systems Society in 2003-2004, and chaired its Analog Signal Processing Technical Committee in 2001-2002. He currently serves as Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I, IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING, and IEEE SENSORS JOURNAL.