

# Energy-Efficient Hybrid Analog/Digital Approximate Computation in Continuous Time

Ning Guo, *Student Member, IEEE*, Yipeng Huang, *Student Member, IEEE*, Tao Mai, *Member, IEEE*, Sharvil Patil, *Student Member, IEEE*, Chi Cao, Mingoo Seok, *Member, IEEE*, Simha Sethumadhavan, *Member, IEEE*, and Yannis Tsividis, *Life Fellow, IEEE*

**Abstract**—We present a unit that performs continuous-time hybrid approximate computation, in which both analog and digital signals are functions of continuous time. Our 65 nm CMOS prototype system is capable of solving nonlinear differential equations up to 4th order, and is scalable to higher orders. Nonlinear functions are generated by a programmable, clockless, continuous-time 8-bit hybrid architecture (ADC + SRAM + DAC). Digitally assisted calibration is used in all analog/mixed-signal blocks. Compared to the prior art, our chip makes possible arbitrary nonlinearities and achieves 16× lower power dissipation, thanks to technology scaling and extensive use of class-AB analog blocks. Typically, the unit achieves a computational accuracy of about 0.5% to 5% RMS, solution times from a fraction of 1  $\mu$ s to several hundred  $\mu$ s, and total computational energy from a fraction of 1 nJ to hundreds of nJ, depending on equation details. Very significant advantages are observed in computational speed and energy (over two orders of magnitude and over one order of magnitude, respectively) compared to those obtained with a modern microcontroller for the same RMS error.

**Index Terms**—Analog computation, continuous-time computation, continuous-time digital, energy-efficient computation, hybrid computation, low-energy computation, nonlinear function generation.

## I. INTRODUCTION

**A**NALOG and hybrid computers were dominant in the 1960s [1]–[4]. They were powerful tools for solving ordinary and partial differential equations (ODEs and PDEs), which are widely used to model and interact with physical systems. For example, in the Apollo space program, analog computers played an important role in simulating spacecraft dynamics and guiding control systems design [4]. All operations were carried out simultaneously, often with computation time independent of the problem size, and with no convergence issues as no time-discretization was used. However, analog and hybrid computers were abandoned in the 1970s—before the dominance of integrated circuits—while still limited by the

Manuscript received November 29, 2015; revised February 04, 2016; accepted March 08, 2016. Date of publication April 29, 2016; date of current version June 22, 2016. This paper was approved by Guest Editor Andrea Mazzanti. This work was supported by the National Science Foundation (NSF) under grant CNS 1239134.

N. Guo, S. Patil, M. Seok, and Y. Tsividis are with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA.

Y. Huang and S. Sethumadhavan are with the Department of Computer Science, Columbia University, New York, NY 10027 USA.

T. Mai is with Apple, Cupertino, CA 95014 USA.

C. Cao is with Broadcom Corporation, Irvine, CA 92606 USA.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSSC.2016.2543729

technology of the time, e.g., discrete-component computing modules, patch boards, etc. Since then, the technical community has hardly considered how modern technology could impact analog computing techniques.

Recently, analog computers were re-visited in the context of modern VLSI technology [5]; it was shown that, in this context, analog computers can be attractive for low-power, self-contained approximate computation and for speed-up of digital computation through co-processing (acceleration). However, in that fully-analog work, the types of mathematical problems that could be solved were limited, because the nonlinear functions available were hard-wired to only a few specific types, and most analog blocks were not calibrated.

In this work, we present a hybrid (mixed analog/digital) computing unit that mitigates the above issues. A programmable, clockless, continuous-time (CT) 8-bit hybrid architecture (ADC + SRAM + DAC) implements arbitrary nonlinear functions in a table-lookup manner. Extensive digitally-assisted calibration corrects analog imperfections in all analog/mixed-signal blocks, improving accuracy. We thus extend analog computing techniques to a new paradigm: continuous-time hybrid computation, where both analog and digital circuits operate in continuous time. In our approach, the time intervals in the digital signals contain important information, in contrast to the case with traditional synchronous or asynchronous digital signals.

Our objective is to demonstrate that approximate computing, a subject of research in the digital computing field [6], can be done efficiently by analog and hybrid circuits in VLSI. For illustration, we concentrate on the solution of differential equations. The analog solution of other types of problems has been addressed elsewhere [7]–[9].

A preliminary report of our work was presented at a conference [10]. The present paper expands on that presentation considerably, by discussing general considerations and estimates for computational energy and time, figures of merit, workflow and architectural details, detailed circuit design, and more measurement results and equation solution examples. An overview of our system is given in Section II. Section III discusses energy dissipation, solution time, and figures of merit. Section IV discusses the architecture of the chip and design choices. Section V describes the integrator design, Section VI the nonlinear function generator, and Section VII the digital calibration. Section VIII gives the measurement results, and Section IX gives equation solution examples. Section X offers a discussion and conclusions.

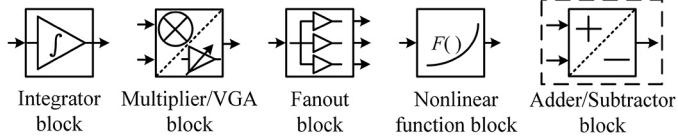


Fig. 1. Basic mathematical operation blocks on the hybrid computing unit.

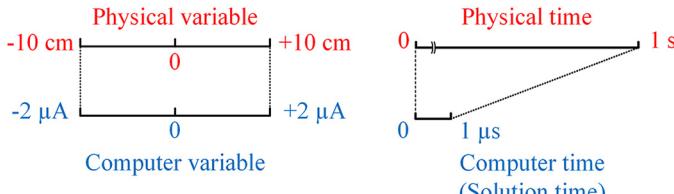


Fig. 2. Amplitude scaling and time scaling examples. The physical variable is assumed to be displacement in this example.

## II. OVERVIEW

The basic building blocks used on our chip are shown in Fig. 1. As we chose differential currents for signal representation (see Section IV for details), fanout blocks are used to make copies of the current signals that need to be distributed to several destinations. Addition and subtraction are done by just sending currents to a common node; thus, a separate adder/subtractor block is not needed on our chip. When used to solve ODEs / PDEs, the various blocks are connected in such a way that the resulting system is characterized by the same equations as that describing the physical system under investigation [1]–[3]; examples will be seen in Section IX. Each integrator’s output represents a system state, and the input to that integrator represents the derivative of that state. Following the imposition of initial conditions, the transient response of the circuits represents the solution of the equations.

Amplitude scaling and time scaling techniques [1]–[3] are necessary on our chip, so that the electrical variables and time are within desired ranges. An illustration example is given in Fig. 2. After the solution, both amplitude and time are unscaled to the original problem variables by a digital computer, which operates in synergy with our chip.

## III. COMPUTATIONAL ENERGY, COMPUTATIONAL TIME, AND FIGURES OF MERIT

The main purpose of the computational unit presented in this paper is the solution of ODEs, including nonlinear ones. The solution time,  $T_{\text{solution}}$ , is highly problem-dependent, and is defined as the time needed for a certain goal to be reached, e.g., a response dies out within a certain margin, or the desired numbers of cycles in a response is reached. Within this solution time, we want to accomplish the task with a solution computational energy,  $E_{\text{solution}}$ , which is as low as possible. This quantity is given by  $E_{\text{solution}} = T_{\text{solution}} P_{\text{equation}}$ , where  $P_{\text{equation}}$  is the total power consumption of the blocks used to map the differential equation on our chip. The latter approximately scales with the order of the equation to be solved,  $n$ , so we will express it as  $P_{\text{equation}} = n P_0$ , where  $P_0$  is the typical power dissipation per order of the differential equation.

Let the time interval of interest in a physical problem considered be  $T_{\text{physical}}$ . We want to achieve a solution time  $T_{\text{solution}}$  which should be as small as possible, to avoid wasting energy due to quiescent and leakage currents involved in  $P_0$ . We thus need to scale the time,  $T_{\text{solution}} = T_{\text{physical}}/\alpha$ , where the time scaling factor  $\alpha$  should be as large as possible; this time scaling is accomplished by choosing the parameter values in the circuits appropriately, usually the overall gain factor of the integrator block [1]–[3]. The upper limit on  $\alpha$  is set by the maximum computing speed, which is represented by the maximum frequency the computer can handle with acceptable error,  $f_{\text{max,computer}}$ . If we want to simulate a physical problem in which we expect the highest frequency of interest to be  $f_{\text{max,physical}}$ , we will have  $\alpha = f_{\text{max,computer}}/f_{\text{max,physical}}$ . Thus, the total energy consumption for a given problem solution will be  $E_{\text{solution}} = T_{\text{solution}} P_{\text{equation}} = T_{\text{physical}} f_{\text{max,physical}}/f_{\text{max,computer}} n P_0$ , or:

$$E_{\text{solution}} = n T_{\text{physical}} f_{\text{max,physical}} \text{FOM}_{\text{computer}} \quad (1)$$

where the quantity

$$\text{FOM}_{\text{computer}} = \frac{P_0}{f_{\text{max,computer}}} \quad (2)$$

is a problem-independent figure of merit (FOM), which can be interpreted as the typical energy dissipation of the computer, per problem order, over one period corresponding to  $f_{\text{max,computer}}$ ; the lower this FOM, the better.

In general, for a given task, we want both  $E_{\text{solution}}$  and  $T_{\text{solution}}$  to be small. This goal is captured by the following FOM:

$$\text{FOM}_{\text{task}} = E_{\text{solution}} T_{\text{solution}} \quad (3)$$

This is a highly problem-dependent FOM; it allows one to evaluate different approaches for solving the same equation, with the end of computation defined in the same manner, and with the same solution accuracy.

*Power scaling:* Basic design tradeoffs [11] show that we can speed up the analog blocks of a computer by increasing  $P_0$ , resulting in a proportional increase of  $f_{\text{max,computer}}$ ; however, as seen from (2), this leaves  $\text{FOM}_{\text{computer}}$  substantially unaffected, which suggests that this figure of merit is characteristic of a given technology and a specific architecture. Eq. (1) then shows that power scaling does not affect  $E_{\text{solution}}$  either. However, the speed-up decreases  $T_{\text{solution}}$  in proportion, and thus decreases  $\text{FOM}_{\text{task}}$  in (3). The above equations and observations help guide design.

## IV. CHIP ARCHITECTURE AND DESIGN CHOICES

The architecture of the chip is shown in Fig. 3. As the unit is meant to be scalable, this test chip was designed to only include a sufficient number of blocks to thoroughly test their function and interaction; it can solve nonlinear ODEs up to fourth order. To keep interference low, the system is organized from top to bottom as rows of analog, mixed-signal and digital blocks, with each individual block placed in separate deep n-wells used for substrate noise isolation. Each block can be connected to any other block through a network of full-crossbar topology, similar

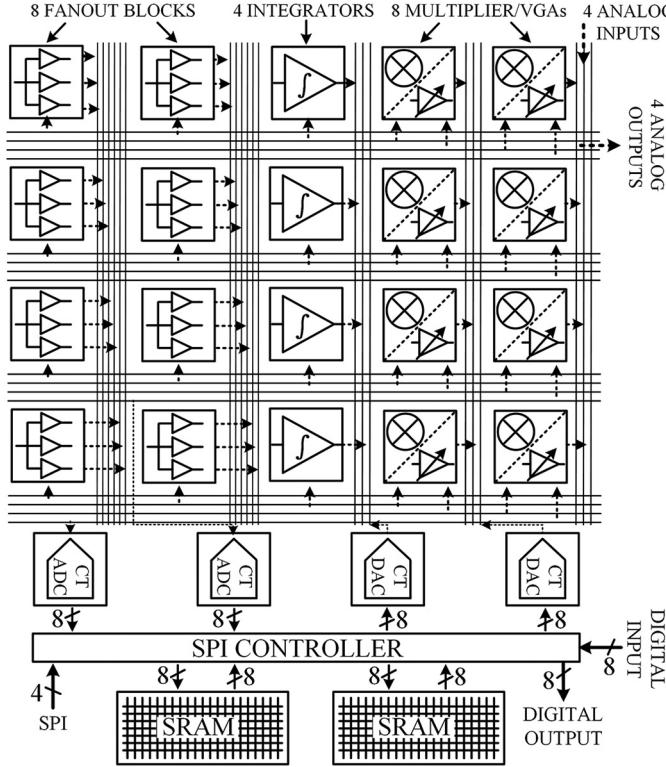


Fig. 3. Hybrid scalable computing unit architecture.

to those used in FPAs and in [5]. Each analog block's input is connected to a global horizontal wire, and its output to a global vertical wire; at the intersections, transmission gates connect any input to any output. The analog inputs/outputs of the chip, marked at the top right corner of Fig. 3, are in current mode; V-I/I-V converters are implemented off-chip for measurement purposes.

The chip is programmed by an external control board through a Serial Peripheral Interface (SPI). The SPI controller switches the system between different operating modes through external instructions, such as signal-path configuration, block parameter setting, computation, etc. The workflow of the computing unit is shown in Fig. 4. The currents corresponding to the state variables in the ODEs can be directly routed off-chip, or can be digitized by on-chip ADCs and sent back to the control board through SPI or parallel digital outputs. The architecture layers from the programming language down to the computing blocks are shown in Fig. 5. Symbolic math expressions of ODEs can be translated into 18-bit instruction words through several compiler layers and downloaded into the hybrid computer. The first 10 bits of the instruction indicate the address of the registers to be modified inside the targeted block, and the last 8 bits carry the content information.

We use differential currents for signal representation for all analog blocks (with dc-coupled, Class-AB, current-mode interfaces), because of several advantages over voltage-mode. First, addition and subtraction are easily done by merging currents with the proper polarity. Second, current-mode multiplication is easily implemented with trans-linear circuits. Lastly, for current-mode interfaces, the voltage swing is kept small, thus

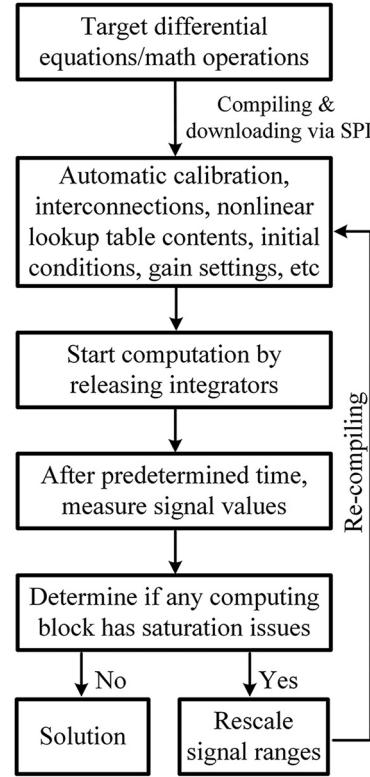


Fig. 4. Hybrid computing unit workflow.

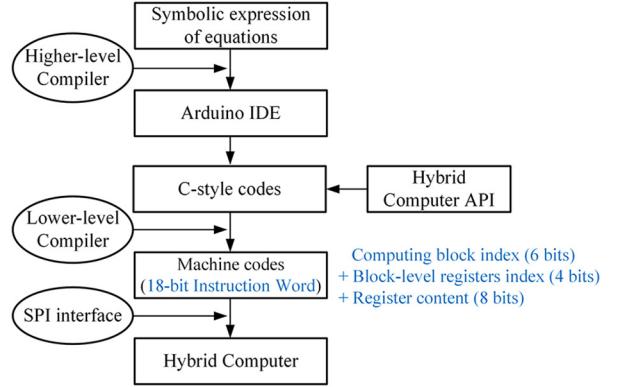


Fig. 5. Hybrid computing unit architecture layers.

mitigating the effects of capacitive coupling to critical signals, e.g., bias voltages, across the chip. Eight-bit accuracy (0.4%) was targeted for each block after calibration. The analog signal bandwidth involved in computation was defined from dc to 20 kHz, in order to limit the power dissipation and reduce parasitic effects (phase shift) to negligible levels.

All non-analog signals and circuits involved in hybrid computation are continuous-time (CT) digital ones, previously demonstrated in signal processing applications [12]. Such circuits involve binary signals that are functions of continuous time, with their time details being an integral part of the signal representation, thus carrying more information than conventional digital signals including asynchronous ones, and avoiding aliasing [12]. To our knowledge, this is the first time that CT

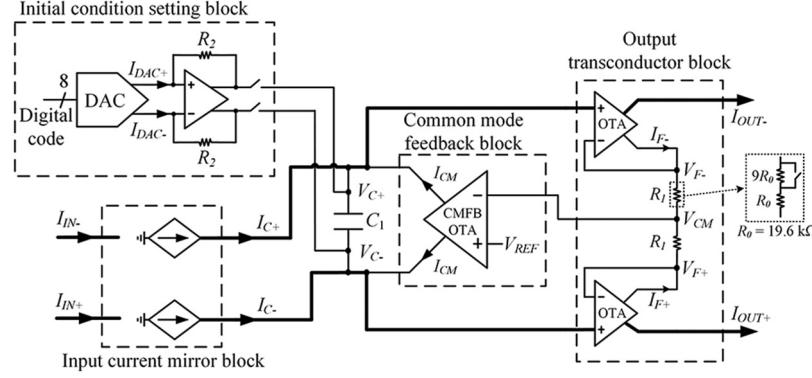


Fig. 6. Integrator architecture. The main signal paths for integration operation are in bold.

digital signals are used in hybrid computation, as will be seen later in Section VI.

The per-order power dissipation (Section III) for our computer is  $P_0 = P_{\text{integrator}} + kP_{\text{multiplier/VGA}} + lP_{\text{fanout}} + mP_{\text{nonlinear}}$ , where  $k$ ,  $l$  and  $m$  are the numbers of multiplier/VGA, fanout, and nonlinear function blocks used, which depend on equation details; the adder/subtractor blocks are not included, as they do not dissipate power, being simply nodes at which wires are joined together. For the equation benchmarks we have considered in this study (Section IX), we obtain the average values  $k = 2.3$ ,  $l = 2.0$  and  $m = 0.4$ ; these values will be used in this work. Throughout circuit design, a great effort was expended on minimizing power dissipation for a chosen bandwidth, thus helping to keep  $P_0$  and FOM<sub>computer</sub> (Section III) low.

The fanout blocks and multiplier/VGA blocks used in our system are based on those in [5] and will not be discussed here.

## V. INTEGRATOR DESIGN

The integrator architecture used in the hybrid computing unit is shown in Fig. 6. Compared to the integrator in [5], it has a dc gain that is much less sensitive to device mismatches. In Fig. 6, the main signal paths are highlighted in bold. Class-AB current mirrors inject a scaled copy of the input differential current to the integration capacitor. The capacitor voltage, which is an integral of the input current, is then converted to a differential output current ( $I_{OUT+} - I_{OUT-}$ ) by the output transconductor block, allowing interfacing with other current-mode analog blocks.

A simplified schematic for the input current mirror is shown in Fig. 7. We chose a Class-AB topology in order to keep the quiescent current low. The mirroring ratio  $k_1$ , can be configured as 1 or 0.1 by adjusting the width of the mirror devices. Gain-boosting amplifiers are added to the cascode devices [13] to increase the low-frequency output impedance of the current mirror, which reduces the loss in the integration operation.

A simplified schematic for the current-copying OTA in the output transconductor block, based on the design in [14], is shown in Fig. 8. It has two matching current outputs: one is the integrator output ( $I_{OUT}$ ), and the other ( $I_F$ ) drives the load resistor  $R_1$  in the common-mode feedback path, as shown in Fig. 6. The negative feedback of the OTA in Fig. 6 forces

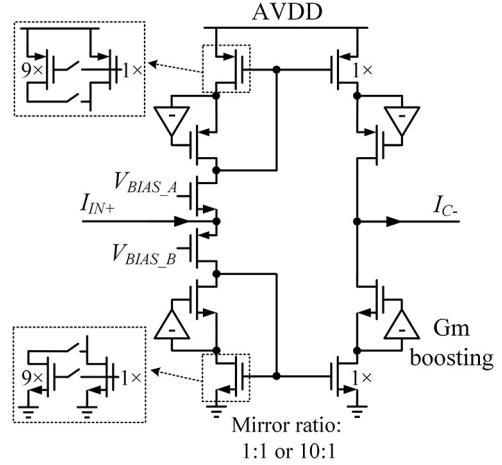


Fig. 7. Simplified schematic of the input current mirror of the integrator, with  $g_m$ -boosted cascodes and two configurable gains.

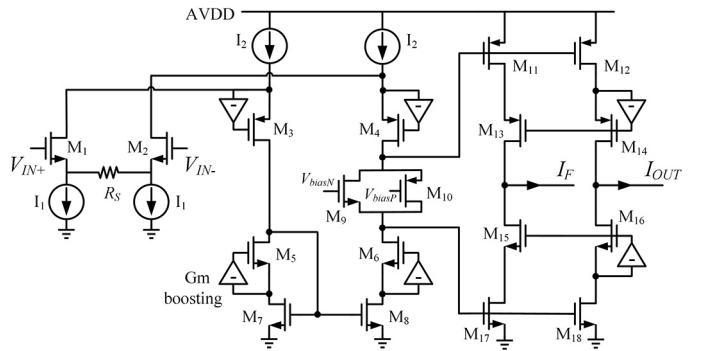


Fig. 8. Schematic of the current-copying OTA at the output stage of the integrator block.

the differential voltage ( $V_{F+} - V_{F-}$ ) across the resistors to follow the voltage across the capacitor ( $V_{C+} - V_{C-}$ ). The differential current through the resistors is therefore  $I_{F+} - I_{F-} = (V_{C+} - V_{C-})/R_1$ . Thus, the output current ( $I_{OUT+} - I_{OUT-}$ ), being a copy of the current ( $I_{F+} - I_{F-}$ ), is proportional to the voltage across the integration capacitor ( $V_{C+} - V_{C-}$ ), i.e.,  $I_{OUT+} - I_{OUT-} = (V_{C+} - V_{C-})/R_1$ . The gain of the output transconductor block is settable by changing the value of the load resistor  $R_1$ .

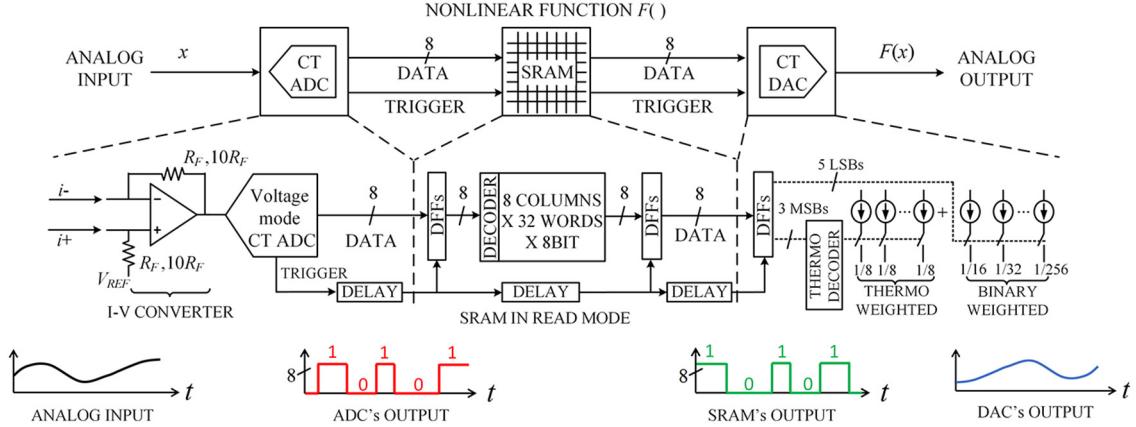


Fig. 9. Continuous-time programmable nonlinear function generator.

The CMFB block in Fig. 6 is an OTA similar to the one used in the output transconductor block (Fig. 8); it has two matching output currents ( $I_{CM}$ ). The CMFB block maintains the capacitor's common-mode voltage with respect to ground by injecting equal currents to both terminals of the capacitor. The common-mode component of the input current of the integrator is therefore absorbed by the OTA in the CMFB block.

The I/O relationship of the integrator, assuming zero initial conditions, is  $I_{OUT+} - I_{OUT-} = 2\pi f_1 \int_0^t (I_{IN+} - I_{IN-}) dt'$ , where  $f_1 = k_1/(2\pi R_1 C_1)$  is the unity gain frequency of the complete integrator, including the input scaling factor  $k_1$ . The unity-gain frequency  $f_1$  is usually taken as  $f_{max,computer}$  (Section III) and thus sets the time scaling factor,  $\alpha$ , [1]–[3]. We use  $C_1 = 40.6 \text{ pF}$ ;  $R_1$  is selectable as either  $19.6 \text{ k}\Omega$  or  $196 \text{ k}\Omega$ ; and  $k_1$  is selectable as either 0.1 or 1. This allows selecting  $f_1$  as 2 kHz, 20 kHz, or 200 kHz.

The initial condition of the integrator is set by imposing a voltage across the integration capacitor using a transimpedance amplifier that is driven by an 8-bit current DAC, as shown in the upper left in Fig. 6. Input and output dc offset currents of the integrator are calibrated using 6-bit current DACs.

## VI. PROGRAMMABLE CT NONLINEAR FUNCTION GENERATOR

The programmable nonlinear function generators greatly expand the range of mathematical problems that can be solved, compared to the earlier effort in [5]. They are implemented in a programmable, CT hybrid architecture, shown in Fig. 9, consisting of a CT ADC, a CT SRAM and a CT DAC. We chose a clockless, CT architecture because it offers several important advantages. First, it is fast, as there is no clock and thus no clock period latency. Second, it has event-driven, activity-dependent power dissipation; the digital circuits switch only when the input analog signal changes. Third, since the circuit does not need a clock, no power is dissipated in distributing a clock signal. Finally, because this scheme operates in CT, it inherently avoids aliasing, which could otherwise affect computation accuracy in certain problems.

The nonlinear function generator works as follows. The input current in Fig. 9 is fed to a transimpedance amplifier, the output of which is converted by a voltage-mode, CT level-crossing

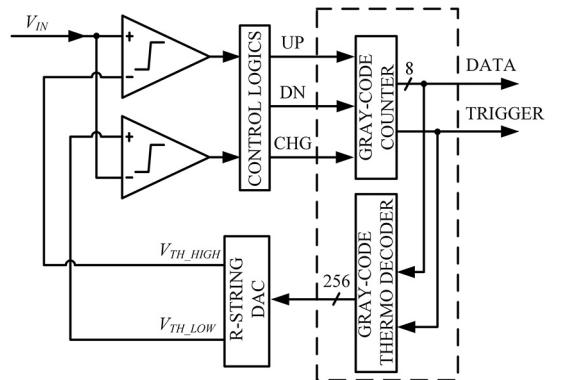


Fig. 10. Voltage-mode level-crossing ADC architecture.

ADC similar to that in [12], into an 8-bit CT digital signal, plus a trigger signal that indicates a level-crossing action in the analog input. The data from the ADC is used as the address at the input of the SRAM block, to fetch the nonlinear function values stored in it. The trigger signal is then passed through a delay line, to give enough time for the SRAM block to finish the read operation; after the data have been read out from the SRAM and have settled, the trigger signal triggers the SRAM's output DFFs and allows the nonlinear function data to be sent to the next stage. The following 8-bit CT DAC converts the data back to current signals, sent to other blocks. Since the conversion of the input and output analog signals is done in CT, this scheme works in real time. A detailed description of the various blocks is given next.

### A. Continuous-Time ADC

The 8-bit CT ADC can convert full-scale signals up to 20 kHz in two selectable signal ranges ( $\pm 2 \mu\text{A}$ ,  $20 \mu\text{A}$ ). The full-scale input voltage of the level-crossing ADC is 0.6 V. The architecture of the ADC is shown in Fig. 10. It has a feedback R-string DAC that adjusts the comparison voltages fed to the comparators in such a way that the input is contained between two successive comparison levels. The design is based on that in [12], except that, in contrast to that work, we use a Gray-code counter and a Gray-code thermometer decoder instead of the original shift-register array in the feedback path, as shown in

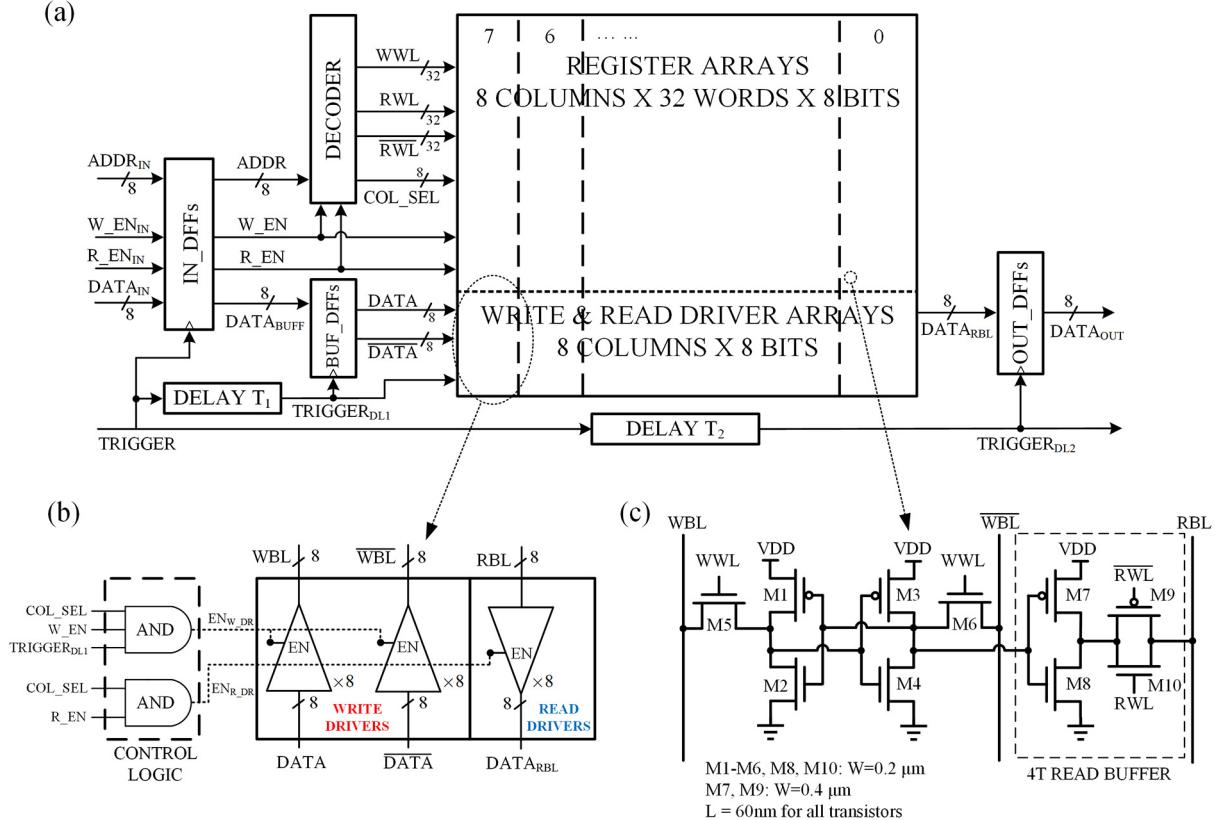


Fig. 11. (a) CT SRAM architecture. (b) Write and read drivers. (c) 10 T SRAM cell.

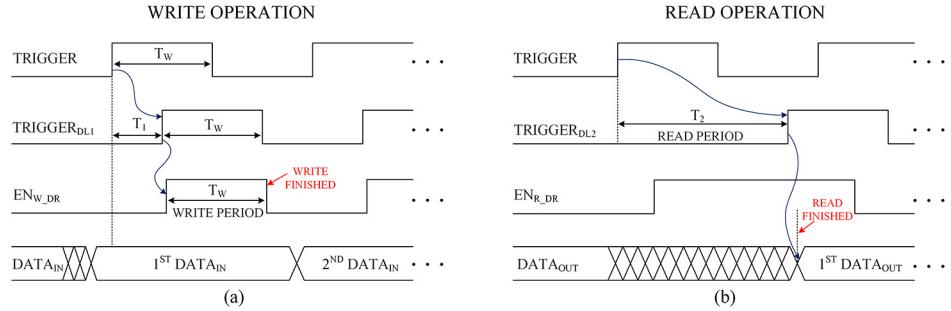


Fig. 12. Timing diagram of critical signals of CT SRAM in (a) write mode and (b) read mode.

the dashed box in Fig. 10, to greatly reduce the peak digital switching current.

### B. Continuous-Time SRAM

The CT SRAM has 8-bit address/word length and its architecture is shown in Fig. 11(a). It has a CT digital data path and its operation is controlled by trigger signals, instead of a clock. The write and read drivers are shown in Fig. 11(b). The SRAM cell used in our work is based on the fully-static 10 T design in [15]; the sizing is shown in Fig. 11(c).

In write mode, the write enable signal  $W\_EN_{IN}$  is set HIGH and the read enable signal  $R\_EN_{IN}$  is set LOW at the SRAM input. The timing diagram of critical signals in write mode is shown in Fig. 12(a). After  $TRIGGER$  triggers the input flip-flops  $IN\_DFFs$ , the 8-bit address  $ADDR_{IN}$  is loaded

to the decoder. The 8-bit content  $DATA_{BUFF}$  is buffered by  $BUF\_DFFs$ . After a delay of  $T_1$  (550 ps is used in our design), which guarantees that the intended WWL (write word line) and  $COL\_SEL$  (column select) have settled,  $TRIGGER_{DL1}$  triggers  $BUF\_DFFs$  and sends  $DATA$  and its complement  $\overline{DATA}$  to the write drivers, which are high-enabled tri-state buffers, shown in Fig. 11(b). Since  $COL\_SEL$ ,  $W\_EN$  and  $TRIGGER_{DL1}$  are all HIGH now,  $EN_{W\_DR}$  is set HIGH and the write drivers are turned on. The differential digital signals  $DATA$  and  $\overline{DATA}$  are then driven onto the bit lines  $WBL$  and  $\overline{WBL}$ , written into the targeted SRAM cells. The write operation lasts for a duration of  $T_W$ , which is set by the external control board. After an interval  $T_W$ ,  $TRIGGER_{DL1}$  goes LOW and  $EN_{W\_DR}$  is set LOW, which disables the write drivers.

In read mode,  $W\_EN_{IN}$  is set LOW and  $R\_EN_{IN}$  is set HIGH at the input. The timing diagram of critical signals in this mode

TABLE I  
ANALOG OFFSETS MINIMIZED BY CALIBRATION

Block type	Output offsets* before calibration (nA)	Output offsets* after calibration (nA)
Fanout	109	4
Multiplier	57	6
Integrator	42	4

\*RMS values of all same type of blocks over one chip for  $\pm 2 \mu\text{A}$  range.

TABLE II  
SOLUTION ACCURACY IMPROVED BY CALIBRATION

ODE's physical background	Nonlinearity involved	RMS error* (uncalibrated)	RMS error* (calibrated)
Large angle motion of pendulum	Trigonometric function (sine)	7.3%	1.5%
Mass-spring dampers with Coulomb friction	Sign function	18.0%	0.5%

\*Relative to full scale.

is shown in Fig. 12(b). After the TRIGGER goes from LOW to HIGH,  $\text{ADDR}_{\text{IN}}$  are again loaded into the DECODER block, which generates  $\text{COL\_SEL}$ , read word line signal  $\text{RWL}$  and its complement  $\overline{\text{RWL}}$ . After  $\text{COL\_SEL}$  and  $\text{RWL}/\overline{\text{RWL}}$  are settled, the intended SRAM word contents are read out by the 4T read buffer inside the 10 T cell, shown in Fig. 11(c). At the same time, the read drivers (high-enabled tri-state buffers) in Fig. 11(b) are also turned on. After a certain amount of delay  $T_2$  (1 ns used in our design)  $\text{TRIGGER}_{\text{DL2}}$  goes HIGH, triggering  $\text{OUT\_DFFs}$  and  $\text{DATA}_{\text{OUT}}$  are sent to the following block, e.g., the CT DAC.

### C. Continuous-Time DAC

As shown in Fig. 9, the 8-bit CT DAC uses a conventional segmented current-steering architecture with two configurable ranges ( $\pm 2 \mu\text{A}$ ,  $20 \mu\text{A}$ ); thermometer coding is used for the three MSBs to ensure monotonicity. The high-frequency glitches generated by the DAC are filtered out by the follow-up bandwidth-limited computing blocks, thus having negligible effects on the overall solution. The CT DAC outputs a differential class-AB current. The encoding scheme for the DAC's digital input is unsigned binary, with the code shifted by  $-128$ .

## VII. DIGITAL CALIBRATION

Input/output offset calibration is done automatically by 6-bit thermometer-code current-steering DACs for all analog and mixed-signal blocks upon startup. The DACs inject a differential current into the class-AB input/output interfaces of analog blocks to minimize dc offsets. The calibration is done automatically using a microcontroller, kept off-chip for testing purposes. When in calibration mode, each analog block is connected, one after the other, to the chip's output to be measured by the microcontroller's ADCs. The microcontroller uses a binary search

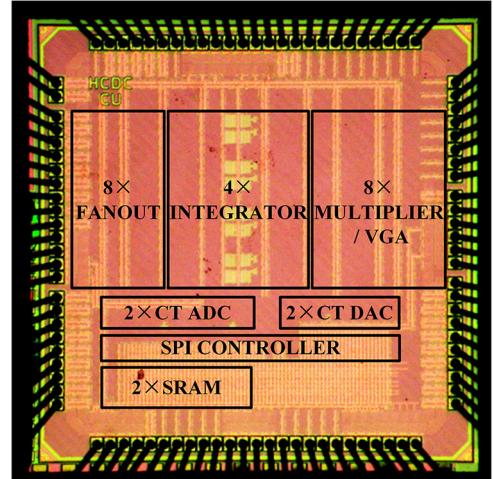


Fig. 13. Die photo.

TABLE III  
HYBRID COMPUTING UNIT PERFORMANCE (27 °C)

Supply voltage	1.2V	Block name	Power
Technology	TSMC 65nm LP	Fanout <sup>4</sup>	37 $\mu\text{W}$
Die area / active area	3.8 mm <sup>2</sup> / 2.0 mm <sup>2</sup>	Integrator <sup>4</sup>	28 $\mu\text{W}$
Number of integrators	4	Multiplier <sup>4</sup>	61 $\mu\text{W}$
Number of multipliers/VGA	8	VGA <sup>4</sup>	49 $\mu\text{W}$
Number of fanout blocks	8	CT ADC <sup>5</sup>	54 $\mu\text{W}$ / 82 $\mu\text{W}$
Number of SRAM	2	CT DAC <sup>5</sup>	4.6 $\mu\text{W}$ / 15 $\mu\text{W}$
Number of analog inputs/outputs	4/4	SRAM <sup>6</sup>	20 $\mu\text{W}$
Digital input/output word length	8 bits	Analog circuits leakage	6.7 $\mu\text{W}$
Programming interface	SPI	Digital circuits leakage (estimate)	85 $\mu\text{W}$
Integrator nonlinearity <sup>1</sup>	0.44%		
Fanout nonlinearity <sup>2</sup>	0.13%		
VGA/Multiplier nonlinearity <sup>3</sup>	0.15%		
ADC+DAC SNDR 1kHz/20kHz	46.3dB/53dB		
DAC DNL/INL	0.73LSB/0.67LSB		
$f_{\text{max},\text{computer}}$	20 kHz		
FOM <sub>computer</sub>	14.1 nJ		

<sup>1</sup>2  $\mu\text{A}$  range, full-scale 20 kHz sine input.

<sup>2</sup>RMS deviation from unity gain over  $\pm 85\%$  full scale.

<sup>3</sup>RMS deviation from unity gain over  $\pm 85\%$  full scale in VGA mode.

<sup>4</sup>2  $\mu\text{A}$  range, 20 kHz full-scale sine input.

<sup>5</sup>2  $\mu\text{A}$  range, 1 kHz/20 kHz full-scale sine input.

<sup>6</sup>20 kHz full-scale sine digital input from ADC; SRAM programmed as a linear lookup table.

algorithm to find the 6-bit calibration code that minimizes the offsets. Table I shows output offsets of analog blocks measured before and after calibration. The accuracy of solving differential equations is greatly improved by the calibrations, as shown in Table II for two examples. After calibration, we have a solution error smaller than 2%.

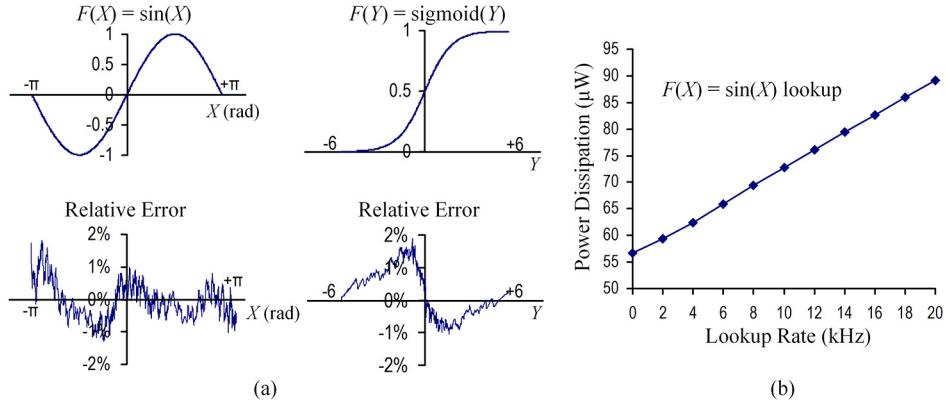


Fig. 14. (a) Nonlinear function lookup examples. (b) Power dissipation.

TABLE IV  
COMPARISON TO PREVIOUS WORK

	One macro in [5]	Our chip
Supply voltage	2.5V	1.2V
Technology	250nm CMOS	65nm CMOS
Active area (estimate)	6.3 mm <sup>2</sup>	2.0 mm <sup>2</sup>
Number of function blocks	25	26
Power with all blocks on (estimated)	18.8 mW	1.2 mW
Programming interface	Non-standard	SPI
Programming environment	Simulink	Arduino IDE
Calibration	Integrators only	All blocks, automatic
Computation types	CT analog only	CT analog / CT hybrid
Nonlinearities available for computation	Specific types	Arbitrary
On-chip ADC, SRAM, DAC	N/A	Available
On-chip digital controller	N/A	Available
Shut down of unused blocks	N/A	Available
$f_{\max, \text{computer}}$	25 kHz <sup>1</sup>	20 kHz
FOM <sub>computer</sub>	150.4 nJ <sup>1</sup>	14.1 nJ

<sup>1</sup>Estimated.

## VIII. MEASUREMENT RESULTS

The test chip has been fabricated in TSMC 65 nm LP CMOS technology. The die photo is shown in Fig. 13. The active area of the chip is 2.0 mm<sup>2</sup>, including circuits used for testing purposes and general programmability; the area would be considerably smaller if only special-purpose computation tasks were targeted. A performance summary is shown in Table III. The measured nonlinearities and noise are consistent with our intended 8-bit accuracy.

Two examples of nonlinear analog function generation and their errors compared to ideal values are shown in Fig. 14(a); the full cycle ( $-\pi$  to  $+\pi$ ) sine function and sigmoid function table lookups have normalized RMS errors of 0.56% and 0.76%, respectively. The total power dissipation of the nonlinear function generator is signal-dependent, decreasing as the table lookup activity decreases, as shown in Fig. 14(b).

To compare our work to the prior art as reported in [5], we use one macro block of that work, which contains a similar number of functional blocks as our hybrid computing chip, as shown in Table IV. The increased functionality is apparent, as is the

TABLE V  
ODES MAPPED ON OUR HYBRID COMPUTING CHIP AND THE BLOCKS NEEDED

No.	ODE's physical background	Order	Integrator	Multiplier/VGA	Fanout	Nonlinear function
1	Mass-spring damper	2	2	2	2	0
2	Large angle motion of pendulum	2	2	2	2	1
3	Mass-spring dampers with Coulomb friction	2	2	2	2	1
4	Van der Pol oscillator	2	2	2	3	0
5	Two-wheel differential-drive robot	3	3	2	2	2
6	Two coupled nonlinear oscillators	4	4	6	6	2
7	Inverted pendulum	4	4	8	7	2
8	1-D heat equation	4	4	4	8	0

lowering of the power dissipation and FOM<sub>computer</sub> by more than an order of magnitude.

## IX. EQUATION SOLUTION EXAMPLES

We have successfully tested the chip using a variety of equations, shown in Table V. We provide two examples below.

The first example is a coupled mass-spring system involving nonlinear springs, shown in Fig. 15(a). The nonlinear ODEs describing the system dynamics are shown in Fig. 15(b). We simulate the motion of the two masses, with displacements  $x_1$  and  $x_3$ , for a physical time of 40 s. Fig. 15(c) shows the block diagram that maps the equations, where the nonlinear functions  $\text{sign}(x_i)\sqrt{|x_i|}$  are implemented as lookup tables. When solving the equations shown in Fig. 15(c), the state variables  $x_1(t)$  and  $x_3(t)$  are continuously varying with time; they are converted by the CT-ADCs into CT digital signals, which are immediately fed into the following SRAMs in order to look up the nonlinear function values. The following CT-DACs convert the SRAMs' outputs back to analog signals, which are distributed to several destinations through fanout blocks. The signal flow in this computing technique is CT hybrid: it is CT digital inside the ADC + SRAM + DAC chain, and CT analog elsewhere.

Our hybrid chip solves the nonlinear ODEs in 320  $\mu\text{s}$  with energy consumption of 0.25  $\mu\text{J}$ , and with 4.7% RMS error relative to full scale. Fig. 15(d) shows the representative solution of  $x_1(t)$  from our chip, together with the ideal solution.

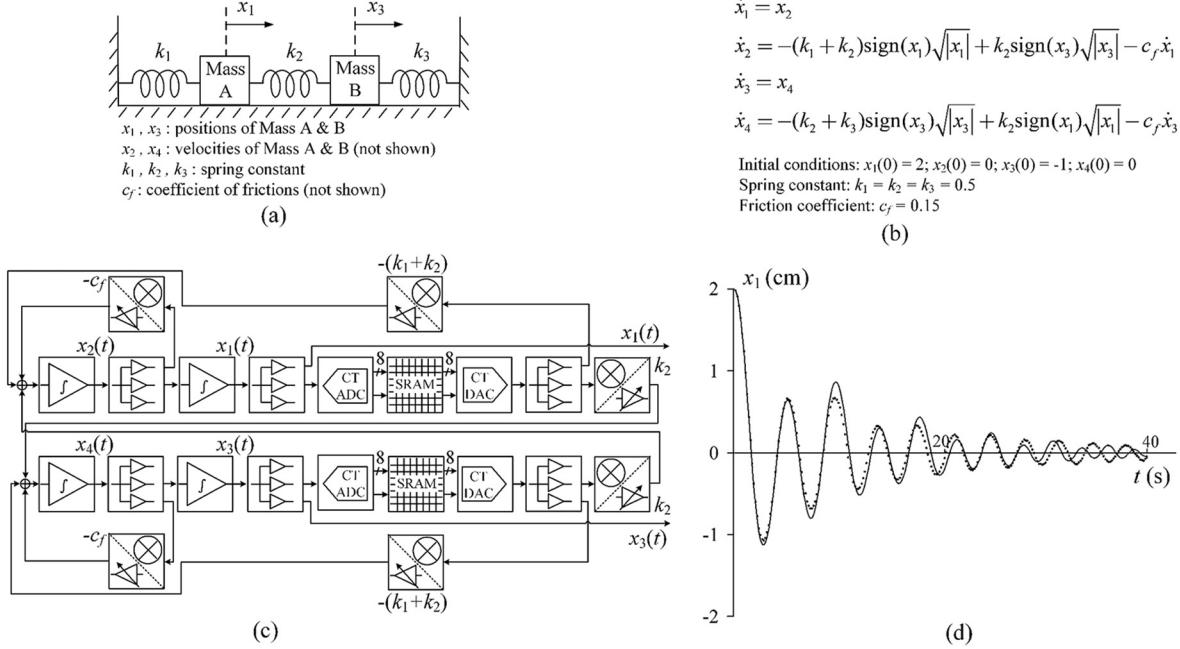


Fig. 15. (a) A 1-D mass-spring system example. (b) The nonlinear ODEs describing its dynamics. (c) The block diagram solving the equations in (b). (d) The solution of  $x_1$  from our hybrid computer (dots) and the ideal solution (solid line).

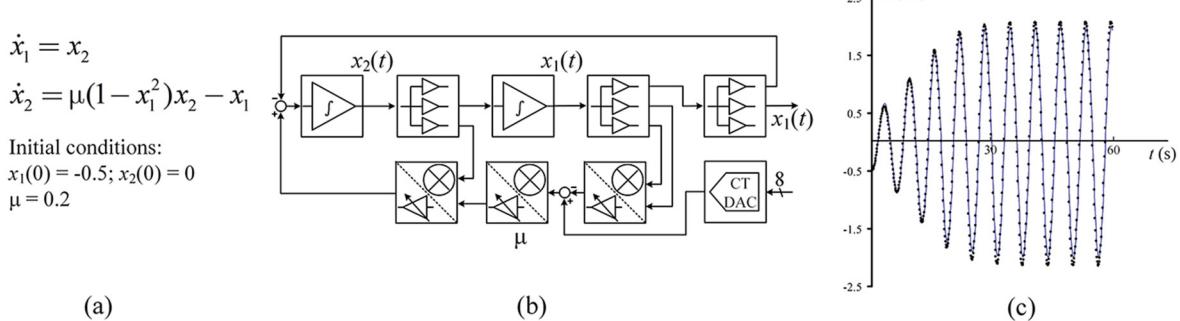


Fig. 16. (a) Van der Pol equation example. (b) The block diagram solving the equations in (a). (c) The solution of  $x_1$  from our hybrid computer (dots) and the ideal solution (solid line).

As a second example, we report on the solution of the second-order, nonlinear Van der Pol equations [16], as shown in Fig. 16(a). Fig. 16(b) shows the block diagram that maps the equations. We solve for a physical time of 60 s. The solution of  $x_1(t)$  is shown in Fig. 16(c), together with the ideal solution. Our hybrid chip solves this problem in 480  $\mu$ s with energy consumption of 0.14  $\mu$ J, and with 4.6% RMS error relative to full scale.

We now compare the performance of our chip to that of a digital microcontroller. Given that we are interested in approximate computation with low energy, we compare to a state-of-the-art MSP430 microcontroller of the same 65 nm technology node [17] (0.4 V, 25 MHz, 7  $\mu$ W/MHz), rather than a full-blown processor. We implemented two popular numerical algorithms for solving ODEs, Euler method and Runge-Kutta 4th order (RK4) method. As expected, the RK4 method gave far better results, so it was chosen for the comparison below. We compiled the numerical algorithms from C code to MSP430 assembly with GNU C compiler, with all optimizations enabled. The resulting assembly was checked against cycle counts published by Texas

Instruments in the MSP430 User's Guide, and the total cycle counts were used to estimate the time and energy consumption by the microcontroller. The time step sizes of RK4 method were chosen so as to obtain the same accuracy as that obtained by our unit, for a fair comparison. The results are shown in Table VI. Due to the discrete time nature of numerical integration, the RK4 method requires many iterations over a large number of clock cycles, which results in long computing time on the microcontroller. For the two equation examples, our chip achieves an advantage of about two orders of magnitude in solution time and one order of advantage in solution energy. This results in an advantage of about three orders of magnitude for our chip in FOM<sub>task</sub> (time-energy product, see Section III), where smaller numbers mean better performance for a given computing task, in solving nonlinear ODEs. While we do not want to push this comparison to a specific microcontroller, the large savings obtained in terms of solution time, computational energy, and time-energy product point to the promise of our technique for energy-efficient approximate computation.

TABLE VI  
COMPARISON TO A MICROCONTROLLER

	Coupled mass-springs, 4.7% RMS error	Van der Pol, 4.6% RMS error		
	<b>Our chip</b>	MSP430 <sup>1</sup> , RK4 method	<b>Our chip</b>	MSP430 <sup>1</sup> , RK4 method
Time step size (s)	N/A	0.85	N/A	0.23
No. of iterations	N/A	47	N/A	260
Clock cycles per iteration (est.)	N/A	21.6k	N/A	7.15k
Total clock cycles (est.)	N/A	1015k	N/A	1859k
Solution time ( $\mu$ s)	<b>320</b>	41k	<b>480</b>	74k
Solution energy ( $\mu$ J)	<b>0.25</b>	7	<b>0.14</b>	13
FOM <sub>task</sub> ( $\mu$ s* $\mu$ J) (Time-energy product)	<b>80</b>	287k	<b>67</b>	962k

<sup>1</sup>25 MHz, 7  $\mu$ W/MHz. N/A: Not Applicable.

## X. DISCUSSION AND CONCLUSIONS

We have introduced continuous-time hybrid approximate computation, and have implemented a prototype system in a scalable architecture in 65 nm CMOS technology. The system can do CT computation with arbitrary nonlinearities that are implemented by a CT ADC + SRAM + DAC architecture, demonstrating for the first time the use of CT digital signals in hybrid computation. CT digital signals are used to do table-lookup tasks in our case; nevertheless, they can also be considered for computation [18]. With CT digital signals involved, hybrid computing attains more versatility, while ensuring aliasing-free operation and adaptive power dissipation.

We have also successfully demonstrated the solution of nonlinear differential equations up to 4th order; the architecture is scalable to higher order. Extensive digitally-assisted calibration is used to improve analog computation accuracy, which is of the order of 0.5% to 5%, depending on the details of the equations. However, the tradeoffs involved in our approach are very different from those in digital computation; thus, in the latter, precision can be increased at will by adding bits, whereas such luxury does not exist in our case. For solving ODEs, digital computation can also use more energy (through smaller time steps) to achieve higher accuracy. On the other hand, in many applications, such as in cyber-physical systems, the overall system accuracy is limited anyway by that of the sensors and actuators involved, so extra bits in computation would not bring a significant advantage. Another difference involves chip area. In our case, it scales approximately in proportion to the problem order, whereas in digital computation a higher-order problem just results in longer computation times, leaving area unaffected.

We should note that analog and hybrid computers for solving differential equations have “taken a 40-year break”, and we are just beginning to research their possibilities in modern VLSI technology. It would thus be premature to draw final conclusions regarding comparisons with digital computers, which have a huge R&D effort behind them. Nevertheless, a limited

comparison for the specific cases reported shows that, compared to a conventional microcontroller in the same technology node, our hybrid computing unit is capable of giving much faster solution (by about two orders of magnitude) with large energy savings (one to two orders of magnitude), for the same error. Thus, one possible use of the techniques presented is in applications where approximate solutions are sought with low computational energy, as is often the case with cyber-physical systems.

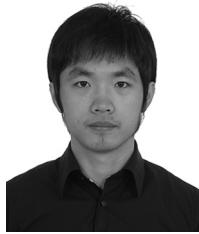
## ACKNOWLEDGMENT

The authors thank Chien-Tang Hu, Doyun Kim, Glenn Cowan, Jianxun Zhu, Teng Yang, Yang Xu, Yu Chen, and Zhe Cao for valuable discussions.

## REFERENCES

- [1] A. S. Jackson, *Analog Computation*, New York, NY, USA: McGraw-Hill, 1960.
- [2] A. E. Rogers and T. W. Connolly, *Analog Computation in Engineering Design*, New York, NY, USA: McGraw-Hill, 1960.
- [3] G. A. Korn and T. M. Korn, *Electronic Analog and Hybrid Computers*, New York, NY, USA: McGraw-Hill, 1964.
- [4] J. A. Lawrence, “The role of JSC engineering simulation in the Apollo program,” *Simulation*, vol. 57, no. 1, pp. 9–16, 1991.
- [5] G. Cowan, R. Melville, and Y. Tsividis, “A VLSI analog computer/math co-processor for a digital computer,” *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2005, pp. 82–83.
- [6] J. Han and M. Orshansky, “Approximate computing: An emerging paradigm for energy-efficient design,” *Proc. 18th IEEE Eur. Test Symp.*, May 2013, pp. 1–6.
- [7] Y. Zhang, “Revisit the analog computer and gradient-based neural system for matrix inversion,” *Proc. IEEE Int. Symp. Intelligent Control*, Jun. 2005, pp. 1411–1416.
- [8] R. S. Amant, A. Yazdanbakhsh, J. Park, B. Thwaites, H. Esmaeilzadeh, A. Hassibi, L. Ceze, and D. Burger, “General-purpose code acceleration with limited-precision analog computation,” *Proc. 41st Int. Symp. Computer Architecture*, Jun. 2014, vol. 42, pp. 505–516.
- [9] C. R. Schlottmann and P. E. Hasler, “A highly dense, low power, programmable analog vector-matrix multiplier: The FPAA implementation,” *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 1, no. 3, pp. 403–411, Sep. 2011.
- [10] N. Guo, Y. Huang, T. Mai, S. Patil, C. Cao, M. Seok, S. Sethumadhavan, and Y. Tsividis, “Continuous-time hybrid computation with programmable nonlinearities,” *Proc. 41th Eur. Solid-State Circuits Conf.*, Sep. 2015, pp. 279–282.
- [11] E. Vittoz and Y. Tsividis, “Frequency–dynamic range–power,” *Tradeoffs in Analog Circuit Design*, C. Toumazou, G. Moschytz and B. Gilbert, Boston, MA, USA: Springer, 2002, pp. 283–313.
- [12] B. Schell and Y. Tsividis, “A clockless ADC/DSP/DAC system with activity-dependent power dissipation and no aliasing,” *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2008, pp. 550–551.
- [13] K. Bult and G. Geelen, “The CMOS gain-boosting technique,” *Analog Integr. Circuits Signal Process.*, vol. 1, pp. 119–135, 1991.
- [14] A. Putra, T. Hui Teo, and S. Rajinder, “Ultra low-power low-voltage integrated preamplifier using class-AB op-amp for biomedical sensor application,” *Proc. IEEE Int. Symp. Integrated Circuits*, Sep. 2007, pp. 216–219.
- [15] D. Kim, G. Chen, M. Fojtic, M. Seok, D. Blaauw, and D. Sylvester, “A 1.85 fW/bit ultra low leakage 10 T SRAM with speed compensation scheme,” *Proc. IEEE Int. Symp. Circuits and Systems*, May 2011, pp. 69–72.
- [16] D. Kaplan and L. Glass, “Two-dimensional differential equations,” *Understanding Nonlinear Dynamics*, New York, NY, USA: Springer, 1995, pp. 240–244.
- [17] D. Bol, J. De Vos, C. Hocquet, F. Botman, F. Durvaux, S. Boyd, D. Flandre, and J. Legat, “SleepWalker: A 25-MHz 0.4-V sub-mm<sup>2</sup> 7- $\mu$ W/MHz microcontroller in 65-nm LP/GP CMOS for low-carbon wireless sensor nodes,” *IEEE J. Solid-State Circuits*, vol. 48, no. 1, pp. 20–32, Jan. 2013.

- [18] Y. Tsividis, *Systems, apparatus, and methods for providing continuous-time signal differentiation and integration*, U.S. patent application US 14/082,945, Nov. 2013 [Online]. Available: <http://www.google.com/patents/US20140139280>



**Ning Guo** (S'14) received the B.S. degree in electrical engineering from Dalian University of Technology, Dalian, China, in 2010, and the M.S. and M.Phil. degrees from Columbia University, New York, NY, USA, in 2012 and 2015, respectively. He is currently working towards the Ph.D. degree in the Department of Electrical Engineering at Columbia University.

His research interests include continuous-time analog/hybrid computing, energy-efficient approximate computing, unconventional computing architecture, and field-programmable analog arrays.



**Yipeng Huang** (S'12) received the B.S. degree in computer engineering in 2011 and the M.S. and M.Phil. degrees in computer science in 2013 and 2015, respectively, all from Columbia University, New York, NY, USA. He is currently pursuing the Ph.D. degree in computer science at Columbia University.

His research interests include applications of analog computing, and performance and efficiency benchmarking of robotic systems.



**Tao Mai** (M'12) received the B.S. and M.S. degrees in electrical engineering from Columbia University, New York, NY, USA, in 2012 and 2013, respectively.

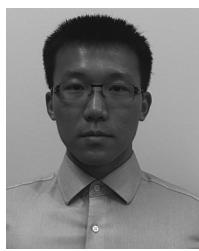
His research interest while at Columbia was in continuous-time signal processing. Since then, he has worked on clock generator IC designs at Silicon Labs, Sunnyvale, CA, USA, and he is currently an RF IC Design Engineer with Apple, Cupertino, CA, USA.



**Sharvil Patil** (S'12) received the B.E. (Hons.) degree in electrical and electronics engineering from Birla Institute of Technology and Science, Pilani, India, in 2009, and the M.S. degree in electrical engineering from Columbia University, New York, NY, USA, in 2012. He is currently pursuing a Ph.D. in electrical engineering at Columbia University.

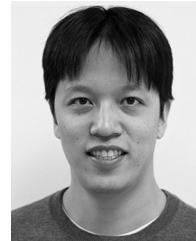
Between 2010–2011, he was with ST Microelectronics, India, as an analog design engineer, where he designed high-speed digital-to-analog converters. His research interests include data converters and signal processing.

Mr. Patil was the recipient of the Analog Devices Outstanding Student Designer Award at Columbia University in 2014.



**Chi Cao** received the B.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2012, and the M.S. degree in electrical engineering from Columbia University, New York, NY, USA, in 2014.

Since then, he has been with Broadcom Corporation, Irvine, CA, USA, where he works on high-speed serial links.



**Mingoo Seok** (M'10) received the B.S. degree (*summa cum laude*) in electrical engineering from Seoul National University, Seoul, South Korea, in 2005, and the M.S. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 2007 and 2011, respectively, all in electrical engineering.

He has been an Assistant Professor in the Department of Electrical Engineering at Columbia University, New York, NY, USA, since 2012. He was a member of technical staff at Texas Instruments, Dallas, TX, USA, in 2011. His research interests include variation/voltage/thermal/aging-adaptive circuits and architecture, ultra-low-power SoC design for emerging embedded systems, machine-learning computing systems, and non-conventional computing and control systems design.

Dr. Seok received the 1999 Distinguished Undergraduate Scholarship from the Korea Foundation for Advanced Studies, 2005 Doctoral Fellowship from the same organization, and 2008 Rackham Pre-Doctoral Fellowship from University of Michigan. He also won 2009 AMD/CICC Scholarship Award for picowatt voltage reference work and 2009 DAC/ISSCC Design Contest for the 35 pW sensor platform design. He recently won the 2015 NSF CAREER Award. He has been serving as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I since 2013, and IEEE TRANSACTIONS ON VLSI SYSTEMS since 2015.



**Simha Sethumadhavan** (M'10) received the Ph.D. degree from The University of Texas at Austin, TX, USA, in 2007.

He is an Associate Professor of Computer Science at Columbia University, New York, NY, USA. His research interests are in computer architecture and computer security.

Dr. Sethumadhavan is a recipient of Alfred P. Sloan fellowship, the NSF CAREER Award, and multiple “top picks” in computer architecture conference awards.



**Yannis Tsividis** (M'74–SM'81–F'86–LF'12) received the B.S. degree from the University of Minnesota, Minneapolis, MN, USA, and the M.S. and Ph.D. degrees from the University of California, Berkeley, CA, USA, in 1972, 1973, and 1976, respectively.

He is Edwin Howard Armstrong Professor of Electrical Engineering at Columbia University, New York, NY, USA. He has worked at Motorola Semiconductor and AT&T Bell Laboratories, and has taught at the University of California at Berkeley, the Massachusetts Institute of Technology, and the National Technical University of Athens, Greece.

Dr. Tsividis has received the 1984 IEEE W.R.G. Baker Award for the best IEEE publication, and is recipient or co-recipient of best paper awards from the European Solid-State Circuits Conference in 1986, the IEEE International Solid-State Circuits Conference in 2003, and the IEEE Circuits and Systems Society (Darlington Award, 1987; Guillemen-Cauer Award, 1998 and 2008). He has received Columbia's Presidential Award for Outstanding Teaching in 2003, the IEEE Undergraduate Teaching Award in 2005, and the IEEE Circuits and Systems Education Award in 2010. In 2012, he was elected Professor Honoris Causa at the University of Patras, Greece, and in 2013 he received the Outstanding Achievement Award of the University of Minnesota. He received the IEEE Gustav Robert Kirchhoff Award in 2007.