Continuous-Time Hybrid Computation with Programmable Nonlinearities

Ning Guo, Yipeng Huang, Tao Mai, Sharvil Patil, Chi Cao, Mingoo Seok, Simha Sethumadhavan, and Yannis Tsividis

Columbia University
New York, NY USA
This talk presents a new principle:
Continuous-time hybrid computation

Both analog and digital signals are functions of continuous-time

Synergy with a digital computer through a common interface

Time intervals: Important info, unlike the case in async digital.
Outline

1. Background and motivation
2. Math operations
3. System architecture
4. Circuit design
5. Measurement results
6. Conclusions
Background and motivation

• Analog computers were dominant in the 1960s; they helped send Man to the moon!
  • Solving ordinary/partial differential equations
  • Parallel computation
  • No convergence issues

https://en.wikipedia.org/wiki/Analog_computer
Background and motivation

• Analog computers were abandoned in the 1960s and 1970s, while they were still using the technology you saw on the previous slide.

• Their potential in modern VLSI technology was not considered until recently [1].

• It was shown that VLSI analog computers are suitable for:
  • Low-power, self-contained approximate computation
  • Speed-up of digital computation through co-processing.

[1] G. Cowan et al., ISSCC 2005
Background and motivation

Analog computation example

Physical system

Math equation:
\[ \ddot{x} = -0.2 \dot{x} - 0.5x + 1; \]

Initial conditions:
\[
\begin{align*}
x(0) &= 9; \\
\dot{x}(0) &= -7
\end{align*}
\]
Background and motivation

Analog computation example

Physical system

Math equation:
\[ \ddot{x} = -0.2 \dot{x} - 0.5x + 1; \]
Initial conditions:
\[ x(0) = 9; \]
\[ \dot{x}(0) = -7. \]

Electrical system
Background and motivation

• We present a new principle: \textit{continuous-time hybrid computation}
  • Continuous-time analog computation
  • \textit{Continuous-time digital} computation 

• Compared to the fully analog approach, the new approach results in:
  • Better accuracy
  • Higher programmability and generality
Basic arithmetic operations:

- **Addition**
  \[ z(t) = x(t) + y(t) \]

- **Subtraction**
  \[ z(t) = x(t) - y(t) \]

- **Multiplication**
  \[ z(t) = x(t) y(t) \]

- **Integration**
  \[ y(t) = \int_0^t x(\tau) \, d\tau \]

- **Nonlinear function**
  \[ y(t) = F(x(t)) \]
Math operations

Amplitude scaling:

Problem variable
→
Machine variable

Example:

- 10 cm

- 2 μA

Machine variable

Time scaling:

Problem time
→
Machine time

Example:

0

1 s

0

1 μs

Machine time
Accuracy: 8-bit

Signal representation:
Differential current

Analog block interface:
DC coupled, Class-AB

Offsets:
< 1 LSB after calibration

Analog signal bandwidth:
DC - 20 KHz
Circuit design: Integrator block

**Input stage**
- Digital code
- DAC
- Input current mirrors
- Initial condition setting block

**Integration stage**
- Common mode feedback block
- Output transconductor

**Output stage**
- Output transconductor

Variables:
- $I_{IN+}$, $I_{IN-}$
- $I_C^+$, $I_C^-$
- $I_{DAC+}$, $I_{DAC-}$
- $Gm$
- $VC^+$, $VC^-$
- $V_{REF}$
- $V_{CM}$
- $V_{OUT+}$, $V_{OUT-}$
Circuit design

Circuit techniques used in other blocks:

Fanout block

Multiplier block (principle)

\[ I_{IN+} \rightarrow V_A \rightarrow I_{OUT1-} \rightarrow V_A \rightarrow I_{OUT2-} \rightarrow I_{OUT3-} \rightarrow V_D \]

M_{1,2,3,4} in weak inversion

\[ V_{GS1} + V_{GS3} = V_{GS2} + V_{GS4} \Rightarrow I_1 I_3 = I_2 I_4 \]

Analog signal routing
System architecture

Nonlinear function:

\[ y(t) = F(x(t)) \]
Circuit design: Nonlinear function block

Our approach: A **continuous-time** programmable lookup table

Advantages over discrete-time counterpart:

- **Activity-dependent** power dissipation
- **Faster response** to input changes
- **No aliasing**
Circuit design: Nonlinear function block

Implementation:

[2] B. Schell et al., ISSCC 2008
Circuit design: Nonlinear function block

Implementation:

\[
i - V_{REF} R_F, 10R_F \\
R_F, 10R_F \\
\text{1-V CONVERTER} \\
\]

\[
\text{Voltage mode CT ADC} \\
\text{CT ADC} \\
\]

\[
\text{SRAM IN READ MODE} \\
\text{DECODER} \\
\text{8 COLUMNS} \\
\text{X 32 WORDS} \\
\text{X 8BIT} \\
\]

\[
\text{CT ADC} \\
\text{CT DAC} \\
\]

\[
\text{ANALOG INPUT} \\
\text{x} \\
\]

\[
\text{AD C’s output} \\
\text{SRAM’s output} \\
\]

\[
\text{NO CLOCK} \\
\]

[2] B. Schell et al., ISSCC 2008
Implementation:

Circuit design: Nonlinear function block

[2] B. Schell et al., ISSCC 2008
Measurement results

Die photo

Key performance summary*

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply voltage</td>
<td>1.2V</td>
</tr>
<tr>
<td>Technology</td>
<td>TSMC 65nm LP</td>
</tr>
<tr>
<td>Die area/active area</td>
<td>3.8 mm² / 2.0 mm²</td>
</tr>
<tr>
<td>Integrator nonlinearity</td>
<td>0.44%</td>
</tr>
<tr>
<td>Fanout nonlinearity</td>
<td>0.13%</td>
</tr>
<tr>
<td>Multiplier/VGA nonlinearity</td>
<td>0.15%</td>
</tr>
<tr>
<td>ADC+DAC SNDR @20KHz</td>
<td>53dB</td>
</tr>
<tr>
<td>DAC DNL / INL</td>
<td>0.73 LSB / 0.67 LSB</td>
</tr>
</tbody>
</table>

*Measurement conditions listed in Digest paper
Measurement results

Power dissipation* (μW)

<table>
<thead>
<tr>
<th>Component</th>
<th>0</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fanout</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiplier</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VGA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT ADC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT DAC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Measurement conditions listed in Digest paper

Calibration helps improve accuracy

<table>
<thead>
<tr>
<th>Nonlinear differential equation</th>
<th>RMS error (uncalibrated)</th>
<th>RMS error (calibrated)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Van der Pol oscillator</td>
<td>17.7%</td>
<td>1.9%</td>
</tr>
<tr>
<td>Large angle motion of pendulum</td>
<td>7.3%</td>
<td>1.5%</td>
</tr>
<tr>
<td>Mass-spring dampers with Coulomb friction</td>
<td>18.0%</td>
<td>1.5%</td>
</tr>
</tbody>
</table>

*Measurement results listed in Digest paper
Measurement results

Nonlinear function generation:

\[ F(X) = \sin(X) \]

Activity-dependent power dissipation:

\[ F(X) = \sin(X) \text{ lookup} \]

\[ F(Y) = \text{sigmoid}(Y) \]

Relative Error

\[ X \text{ (rad)} \]

\[ Y \]

Power Dissipation (μW)

Lookup Rate (kHz)
Application demonstration

A two-wheel drive robot with model predictive control

Continuous-time system dynamics

\[
\begin{align*}
\dot{x} &= \cos \\
\dot{y} &= \sin \\
\end{align*}
\]

\(\omega\): angular velocity
\(v\): linear velocity
Application demonstration

A two-wheel drive robot with model predictive control

Possible futures states in 0.1s

Continuous-time system dynamics

\[
\begin{align*}
\dot{x} &= \cos \\
\dot{y} &= \sin \\
\omega: & \text{ angular velocity} \\
v: & \text{ linear velocity}
\end{align*}
\]

Current state

\[
\begin{align*}
x(t) & \quad \text{DATA} \\
\theta(t) & \quad \text{CT ADC} \\
\omega & \quad \text{DAC} \\
y(t) & \quad \text{CT ADC} \\
v & \quad \text{DAC}
\end{align*}
\]

\[
\begin{align*}
\text{DATA} & \quad \text{SRAM} \\
\text{TRIGGER SIGNAL} & \quad \text{SRAM} \\
\text{COS( )} & \quad \text{SRAM} \\
\text{SIN( )} & \quad \text{SRAM}
\end{align*}
\]

\[
\begin{align*}
T &= 0.84 \mu s, \quad \text{Energy} = 0.48 \text{ nJ} \\
\text{RMS error} &= 0.6\%
\end{align*}
\]
## Comparison to prior art

<table>
<thead>
<tr>
<th></th>
<th>One macro in [1]</th>
<th>Our chip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply voltage</td>
<td>2.5 V</td>
<td>1.2V</td>
</tr>
<tr>
<td>Technology</td>
<td>250nm CMOS</td>
<td>65nm CMOS</td>
</tr>
<tr>
<td>Active area (estimate)</td>
<td>6.3 mm²</td>
<td>2.0 mm²</td>
</tr>
<tr>
<td>Number of function blocks</td>
<td>25</td>
<td>26</td>
</tr>
<tr>
<td>Power with all blocks on (estimate)</td>
<td>18.8 mW</td>
<td>1.2 mW</td>
</tr>
<tr>
<td>Calibration</td>
<td>Integrators only</td>
<td>All blocks</td>
</tr>
<tr>
<td>Computation types</td>
<td>CT analog only</td>
<td>CT analog / CT hybrid</td>
</tr>
<tr>
<td>Nonlinearities available for computation</td>
<td>Specific types: exp(), log(), absolute, saturation, etc.</td>
<td>Arbitrary</td>
</tr>
<tr>
<td>On-chip ADC, SRAM, DAC</td>
<td>N/A</td>
<td>Available</td>
</tr>
</tbody>
</table>

[1] G. Cowan et al., ISSCC 2005
Conclusions

• We have presented the first continuous-time hybrid computing unit.

• Arbitrary nonlinear functions are implemented by a continuous-time hybrid architecture (ADC+SRAM+DAC).

• We have used the chip to successfully solve several benchmark equations and demonstrated the use of the chip in a robotic application.

• We expect this technique to find applications in low-power approximate computation and in acceleration of digital computation.
Acknowledgement

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

This work has been supported by National Science Foundation grant CNS 1239134.
References


Thank you!

Questions?
## Comparison details

<table>
<thead>
<tr>
<th></th>
<th>Time step size</th>
<th>Total clock cycles</th>
<th>Time needed for one solution</th>
<th>Energy consumption for one solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our hybrid chip</td>
<td>N/A</td>
<td>N/A</td>
<td>0.84µs</td>
<td>0.48nJ</td>
</tr>
<tr>
<td>0.4V RISC microprocessor [8]</td>
<td>0.1s</td>
<td>734</td>
<td>29µs</td>
<td>5.14nJ</td>
</tr>
</tbody>
</table>

35X better  11X better