Self-Scaling Evolution of Analog Computation Circuits with Digital Accuracy Refinement

Steven D. Pyle, Vignesh Thangavel, Stephen M. Williams, and Ronald F. DeMara
University of Central Florida – Division of Electrical and Computer Engineering

15-18 June 2015
Agenda

Analog Computation Challenges
- difficult to design, low accuracy and precision, device range

SCALER
- two-pronged strategy utilizing digital resources to address analog challenges
  - Self-Scaling Genetic Algorithm & Differential Digital Correction

Computational Circuit Evolution
- demonstrate SCALER for four computational circuits
CMOS scaling limits
- technology scales → improve performance at present technology node
- utilize analog devices to perform continuous time computations

Could provide a 20-year leap in performance versus digital
- “Gene’s Law” indicates a 1000x to 10,000x improvement in efficiency of analog signal processing over DSP

Low-Power Analog DFT for OFDM
- Suh et al. 4-dB reduction power only 2-dB performance degradation

<table>
<thead>
<tr>
<th>Chipset</th>
<th>Virtex2Pro FPGA</th>
<th>Virtex FPGA</th>
<th>RASP2.9 FPAA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power consumption @ Processing delay</td>
<td>247 mW @ 49.6 ns</td>
<td>219 mW @ 101.6 ns</td>
<td>13.4 mW @ 4 μs</td>
</tr>
</tbody>
</table>

[Suh 2011]

[Hasler 2002]
Design Complexity

- analog circuits require design expertise
- **genetic algorithms (GAs)** combined with a **reconfigurable analog fabric** can automate analog circuit design

Device Limitations

- analog computations outside of device’s operating range
- **Self-Scaling Genetic Algorithm (SSGA)** adapts analog outputs to computationally tractable range

Accuracy and Precision Limitations

- analog circuit evolution may converge to close-but-not-accurate solutions
- high-precision DACs and ADCs are energy and resource intensive
- **Differential Digital Correction (DDC)** improves analog solutions post ADC with minimal overhead
Genetic Algorithms

Overview

- first, a population of configurations known as individuals is randomly generated
- for each individual, test inputs are applied, and outputs determine each individual’s fitness
- individuals are selected based on their fitness for crossover and mutation to populate the next generation
- perform fitness evaluation, crossover, and mutation until the desired fitness is achieved

Selection

- individuals chosen for crossover by randomly choosing two and then selecting the one with the best fitness
- called a tournament selection of two

Elitism

- the best and second best-fit individuals are retained into the next generation
- elitism of degree 2
Overview of Particle Swarm Optimization (PSO)

- A set of particles is randomly distributed within a range \([p_{\text{min}}, p_{\text{max}}]\) in \(n\)-dimensional space.
- Each dimension corresponds to a parameter to be optimized.
- Particles are tested by substituting their location into optimization problem and determining quality of output—similar to fitness in GAs.
- Each particle’s previous best location (\(p_{\text{Best}}\)) as well as global best location (\(g_{\text{Best}}\)) are saved.
- Each particle’s velocity and position are updated depending on \(p_{\text{Best}}\) and \(g_{\text{Best}}\) according to:

\[
\begin{align*}
    v_{n+1}^{i} &= v_{n}^{i} + c_{1} \text{rand}() (p_{\text{Best}}^{i} - x_{n}^{i}) + c_{2} \text{rand}() (g_{\text{Best}} - x_{n}^{i}) \\
    x_{n+1}^{i} &= x_{n}^{i} + v_{n+1}^{i}
\end{align*}
\]

[Wang 2010]
## Related Works

<table>
<thead>
<tr>
<th>Research Work</th>
<th>Analog Circuits Evolved</th>
<th>EA Type</th>
<th>Platform</th>
<th>Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Keymeulen 2000]</td>
<td>Multiplier</td>
<td>Intrinsic</td>
<td>Custom FPTA with 48 switchable transistor terminals in two 0.5um chips</td>
<td>Population-based and Fitness-based fault tolerance.</td>
</tr>
<tr>
<td>[Streeter 2002]</td>
<td>Cube</td>
<td>Extrinsic</td>
<td>Weakly-constrained virtual fabric under progressive voltage conditions</td>
<td>Average error 7-fold less than human design.</td>
</tr>
<tr>
<td>[Cornforth 2014]</td>
<td>Random Black Box Non-linear circuits</td>
<td>Extrinsic</td>
<td>NG-SPICE</td>
<td>Demonstrated that an age-fitness incremental algorithm is better for non-linear analog circuit fitness evaluation.</td>
</tr>
<tr>
<td>This work</td>
<td>Square root, cube root, square, cube</td>
<td>Intrinsic</td>
<td>Cypress PSoC-5LP</td>
<td>Digital resources enhance accuracy of self-scaling GA with PSO.</td>
</tr>
</tbody>
</table>
1) Self-Scaling Genetic Algorithm (SSGA)
   - **Extension to GAs** uses PSO to adapt analog outputs to a more computationally tractable range

2) Differential Digital Correction (DDC)
   - **Uses four PLDs and 1KB of memory** improves accuracy and precision of analog
Self-Scaling (SS) parameters

- scaling (A) and translation (B) parameters are optimized with a 2-dimensional PSO algorithm during GA-driven evolution

\[ SS\{f(x)\} = Af(x) + B \]

- removes intrinsic device range limitations
- allows the GA to dynamically explore and exploit potentially more computationally-tractable ranges of the device
- each individual's fitness is determined by

\[ \text{fitness} = \sum_{x=0}^{M} |SS\{f(x)\} - oracle_x| \]

where \( M \) is the number of test inputs and \( oracle_x \) is a precomputed ideal output for input \( x \)
Computational Analog Elements (CAEs)

- Could be high-level analog blocks – integrators, filters, adders, etc.
  - or, low-level analog devices such as transistors or capacitors
- Each gene of the SSGA contains all information to fully develop a CAE
  - Function
  - Parameters
  - Routing
- Each individual's genome contains
  - Enough genes to configure all CAEs for the target application
  - SS parameters

Hypermutation

- Reinitializes all SS parameters while saving gBest and each pBest
- Reinitializes half of the population for new genes to utilize new SS parameters
- Utilized to break SSGA out of local minimums
Overview

- Once SSGA is completed and best-fit individual determined, a 256-valued *Normalized Error Array (NEA)* is generated, which contains fractions of maximum analog error.
- DDC uses GA to evolve a small amount of digital fabric that when presented the same 8-bit input, $x$, as the analog fabric to output $D(x)$ such that the fitness is minimized.

$$fitness = \sum_{x=0}^{M} |oracle_x - SS\{f(x)\} - NEA[D(x)]|$$

- After each individual’s fitness is evaluated, they are selected based on their fitness for *crossover* and *mutation* to populate the next generation.
- Desired fitness is achieved or the maximum generation is reached.

Chromosome

- Chromosome for each PLD has: AL (active lines), AND array and OR array parameters representing configuration bits.
- Input lines that are active are determined at boot-time and marked as active lines.

Differential Digital Correction (DDC)
Population

- **individual**: represents 4 PLDs evolving *simultaneously*
- **population**: 80 individuals is randomly initialized while copying *active lines from seed configuration*

Selection and Crossover

- **tournament selection**: is done with a tournament size of two and constitute 40 of the total 80 individuals per generation
- **single-point crossover**: is performed between the fitter 40 and another individual randomly chosen from the whole population
- **new offspring**: replace the lower fraction of the population and fitter individuals in tournament selection fill the upper fraction
- **replacement**: number of individuals replaced is a variable parameter
Adaptive Hardware and Systems 2015

Analog Computational Circuit Evolution

First demonstration of analog CC evolution on commercial device

- PSoC 5LP from Cypress
- Reconfigurable analog fabric primarily consisting of Switched Capacitor op-amp blocks (SC blocks)
- Reconfigurable digital fabric consisting of Universal Digital Blocks (UDBs)
- ARM Cortex M3 core

Experimental Configuration

- SC blocks used as high-level CAEs with 8 fixed topologies with a variety of parameters
  - naked op-amp
  - trans-impedance amplifier
  - continuous-time mixer
  - discrete-time mixer
  - unity gain buffer
  - first order modulator
  - programmable gain amplifier
  - track and hold amplifier

- UDBs contain 2 PLDs each
- each DDC individual utilizes the PLDs of 2 UDBs

[Cypress 2013]
Test Cases

Four computational circuits evolved

\[ x^2, \sqrt{x}, x^3, \text{ and } \sqrt[3]{x} \]

- without SSGA, only \( \sqrt{x} \) and \( \sqrt[3]{x} \) fit within the native range of the PSoC-5LP
- \( p_{max} \) is the maximum allowed value for SS parameters
  - need bounds or else PSO can quickly converge to impractical values
- \( \text{penalty case} \) is the difference between the output and the oracle for each test case to which a 10-point penalty is added

Experimental Setup

- each CC is evolved five times with different seeds for both a standard GA and the SSGA
- \( \text{max generation} = 500 \)
- \( \text{hypermutation condition} = 100 \)
- in order to compare to previous works, component complexity is defined as the number of operations each component can perform.
  - resistor = 1
  - BJT = 4
  - SC Blocks = 8
- since we only have four SC blocks, our circuit complexity is fixed to 32

<table>
<thead>
<tr>
<th></th>
<th>Square</th>
<th>Square-root</th>
<th>Cube</th>
<th>Cube-root</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native input range</td>
<td>0V-2.02V</td>
<td>0V-4.08V</td>
<td>0V-1.60V</td>
<td>0V-4.08V</td>
</tr>
<tr>
<td>w/ simple GA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Effective input</td>
<td>0V-4.08V</td>
<td>0V-4.08V</td>
<td>0V-4.08V</td>
<td>0V-4.08V</td>
</tr>
<tr>
<td>Range with SSGA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Native Output Range</td>
<td>0V-4.08V</td>
<td>0V-2.02V</td>
<td>0V-4.08V</td>
<td>0V-1.60V</td>
</tr>
<tr>
<td>Effective Output</td>
<td>0V-16.65V</td>
<td>0V-2.02V</td>
<td>0V-67.92V</td>
<td>0V-1.60V</td>
</tr>
<tr>
<td>range with SSGA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p_{max} )</td>
<td>20</td>
<td>2</td>
<td>68</td>
<td>2</td>
</tr>
<tr>
<td>( \text{penalty case} )</td>
<td>0.5V</td>
<td>0.1V</td>
<td>0.5V</td>
<td>0.1V</td>
</tr>
</tbody>
</table>

Adaptive Hardware and Systems 2015
Fitness Scores, Error, and Magnitude of Improvement

- **cube circuit**: worst relative fitness, but best SS demonstration by increasing range 17x
- **square circuit**: most improvement as range unobtainable with standard GA, but requires less scaling
- **square-root & cube-root**: able to evolve well w/ standard GA, but improved w/ SSGA
Results

Observed results

- Greatest reduction in average error was seen for the cube circuit where a reduction from 1160mV to 732mV yielded a 36.89 percent reduction in error on average.

- DDC improved accuracy by reducing average error in square, square root and cube-root CCs by 28.57, 10.67 and 16.3 percent respectively, on average.
Summary of Approach

- SCALER scales, translates, and refines evolved analog computational circuits using evolved digital resources.
- The creative hybrid analog-digital design evolved, leverages the relative advantages of both circuit domains.
- PSO with an Island-like GA was incorporated to realize a 12.9-fold fitness improvement of the best-fitness analog circuit.
- Following evolution of analog computational circuits, DDC evolves a precise digital error compensation circuit to compensate for analog aberrations.
Extensions

- SSGA could be applied to frequency domain analysis via adjustment of FFT coefficients
- Precision of DDC can further be improved by using values that can satisfy a 16-bit mapping instead of the 8-bit mapping used here
- SCALER overall could benefit from more exploration in the search space of PSO parameters and seeds used for unrefined GA

Future Work

- larger FPAA platforms with additional computational analog blocks to determine range of accurate computation achievable
- intrinsically generate faster and energy-efficient hybrid analog-digital computational circuits for repetitive scientific and embedded applications
References


