

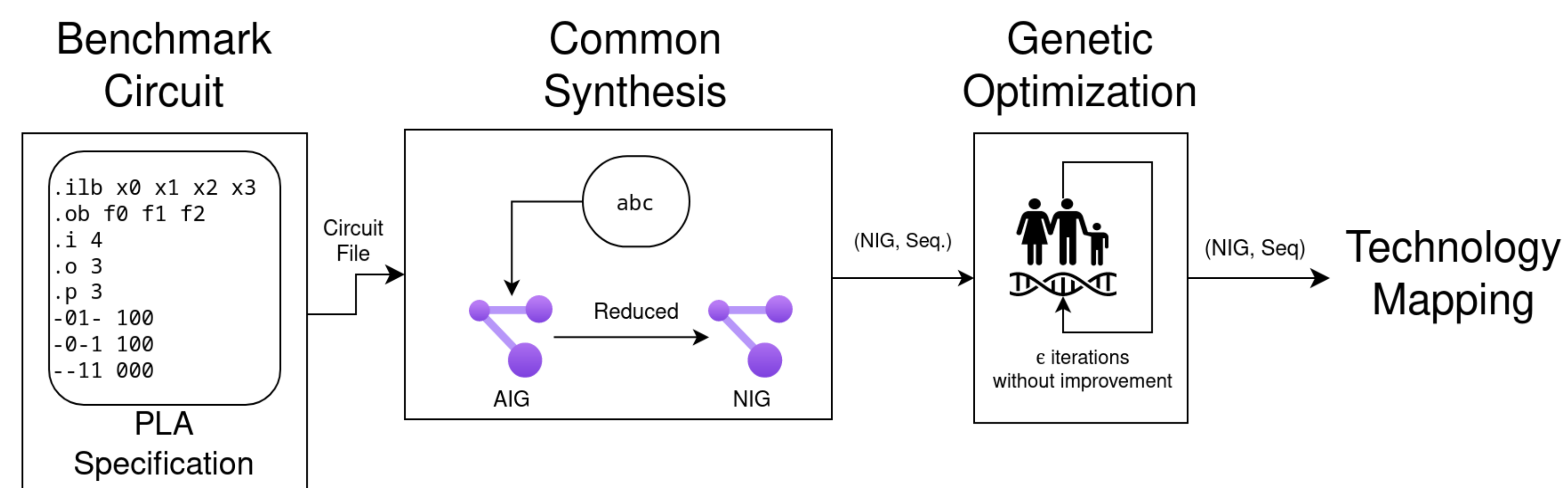
Synthesis Augmentation with Genetic Algorithms

Abstract

The von-Neumann architecture has a bottleneck which limits the speed at which data can be made available for computation. To combat this problem, novel paradigms for computing are being developed. One such paradigm, known as in-memory computing, interleaves computation with the storage of data within the same circuits. MAGIC, or Memristor Aided Logic, is an approach which uses memory circuits which physically perform computation through write operations to memory. Sequencing these operations is a computationally difficult problem which is directly correlated with the cost of solutions using MAGIC based in-memory computation. SAGA models the execution sequences as a topological sorting problem which makes the optimization well-suited for genetic algorithms. We then detail the formation and implementation of these genetic algorithms and evaluate them over a number of open circuit implementations. The memory-footprint needed for evaluating each of these circuits is decreased by up to 52% from existing, greedy-algorithm-based optimization solutions. Over the benchmarks evaluated, these modifications lead to an overall improvement in the efficiency of in-memory circuit evaluation of 128% in the best case and 27.5% on average.

Methods

This work makes use of genetic algorithms (GAs) which are configured to use single-point, order-based crossover. Populations of 2000 genotypic individuals were initialized and evaluated with their size as detailed in the problem model figure. The cost for each sequence is then used as the fitness, allowing us to optimize circuits in the EDA workflow as show in the Figure below. The workflow can iterate on solutions for as long as the user describes, allowing for lighter optimizations during the iteration phase and more performant optimizations at the end when more time can be dedicated to synthesis.



Discussion

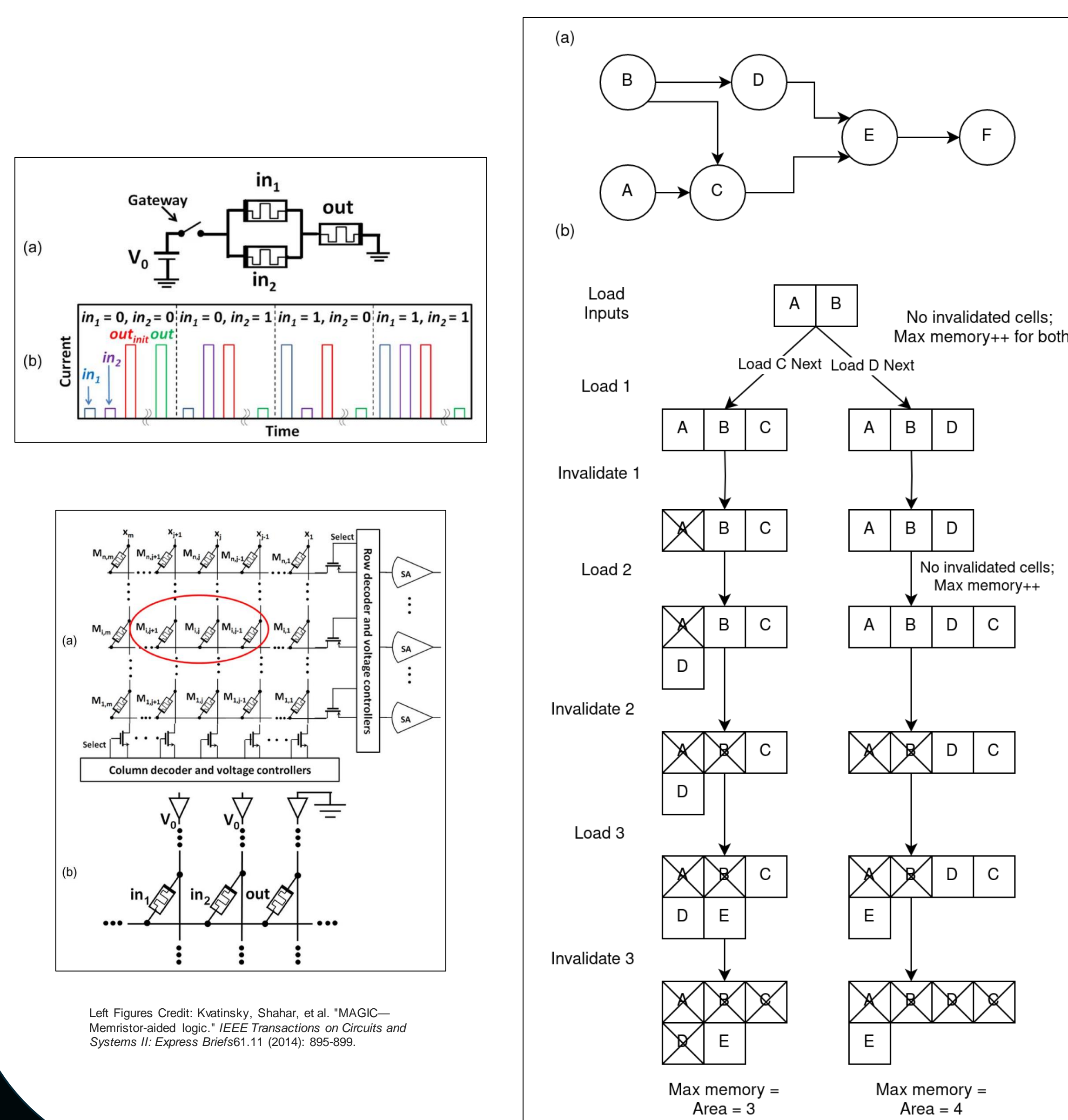
In contrast to one of the prior state-of-the-art works no additional work to identify operations which can be performed in parallel is done. Even without these additional optimizations the efficiency increased in the majority of cases and on average for the benchmark circuits. Potential applications of this technology could optimize circuits beyond what is feasible by human engineers by leveraging the search potential of GAs to more effectively explore circuit representations. Additionally, adding GAs to the EDA pipeline allows for their reuse in other contexts in the pipeline; novel applications of GAs for EDA can therefore be explored.

Future Work

Future work will explore:

1. Opportunities for integration of genetic algorithms and augmentation earlier and throughout the synthesis pipeline
2. Hyperparameter configurations conducive to quickly converging upon a suitably optimal solution
3. Optimizations which can reduce the required synthesis time of this process in order to improve the developer experience and make benefits of this model applicable in more spaces

Problem Model



Current Work & Results

Table 1: Performance comparison between this work and prior works. Improvements detailed are the percentage decrease in cycles and area and the percentage improvement to overall efficiency using the efficiency metric defined in [4]. Cycles are measured in the number of operations needed to fully compute the formula. Area is the number of memory cells needed. Efficiency is calculated according to the formula $Efficiency = 10^6 \cdot Area^{-1} \cdot Cycles^{-1}$. Efficiency metrics have been rounded to the nearest whole number for display, greater is better.

Benchmark	SIMPLER MAGIC [4]			SAGA			Improvement			
	Cycles	Area	Efficiency	ϵ	Cycles	Area	Efficiency	Cycles	Area	Efficiency
5xp1	119	39	215	5000	160	29	216	-34.5%	25.6%	0.47%
9symm1	218	57	80	5000	306	57	57	-40.4%	0.0%	-28.8%
clip	160	47	133	5000	233	40	107	-45.6%	14.9%	-19.5%
cm150a	67	39	383	50	52	22	874	22.4%	43.6%	128%
cm162a	64	35	446	50	87	20	575	-35.9%	42.9%	28.9%
cm163a	66	36	421	50	76	17	774	-15.2%	52.8%	83.8%
misex1	83	33	365	500	84	17	700	-1.20%	48.5%	91.8%
parity	81	35	353	500	104	20	481	-28.4%	42.9%	36.3%
sao2	128	53	147	5000	213	43	109	-66.4%	18.9%	-25.9%
x2	73	33	415	5000	80	16	781	-9.59%	51.5%	88.2%

Table 2: The average percentage change of each statistic for all 10 benchmark circuits when comparing SIMPLER to this work. Negative values are decreases in the metric while positive are improvements.

Statistic	Cycles	Area	Efficiency
Arithmetic Mean	-25.5%	33.6%	38.3%
Geometric Mean	-29.5%	32.3%	27.5%
Standard Deviation	25.3%	19.2%	56.7%
95% Confidence	(-41.1, -9.82)	(21.7, 45.5)	(3.17, 73.5)

Acknowledgements

This work was funded in part by the ORCGS fellowship at UCF. It extends prior work from a project in EEE 5336 on CAD of VLSI at the UCF in Fall of 2023.

ANDEY ROBINS

andey.robins@ucf.edu

Ph.D. Student



DRACO LAB | www.ece.ucf.edu/DRACO

Director: Dr. Mike Borowczak

Dept. Of Electrical & Computer Engineering
University of Central Florida

Student Scholars Symposium – March 26-27th 2024

