MNRL and MNCaRT

An Open-Source, Multi-Architecture State Machine Research and Execution Ecosystem

Kevin Angstadt[†] Jack Wadden[‡] Vinh Dang[‡] Ted Xie[‡] Dan Kramp[‡]
Westley Weimer[†] Mircea Stan[‡] Kevin Skadron[‡]

[†]University of Michigan
{angstadt, weimerw}@umich.edu

[‡]University of Virginia
{wadden, vqd8a, ted.xie, dankramp, mircea, skadron}@virginia.edu

COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY OF MICHIGAN

UNIVERSITY of VIRGINIA

What is Automata Processing?

- Finite automata are state machines that match **patterns of data** within an input data stream
- Theoretical model used to model and process **regular expressions**
- Able to **accelerate a broad range of applications**: natural language processing, network security, graph analytics, high-energy physics, bioinformatics, pseudo-random number generation and simulation, data-mining, and machine learning.

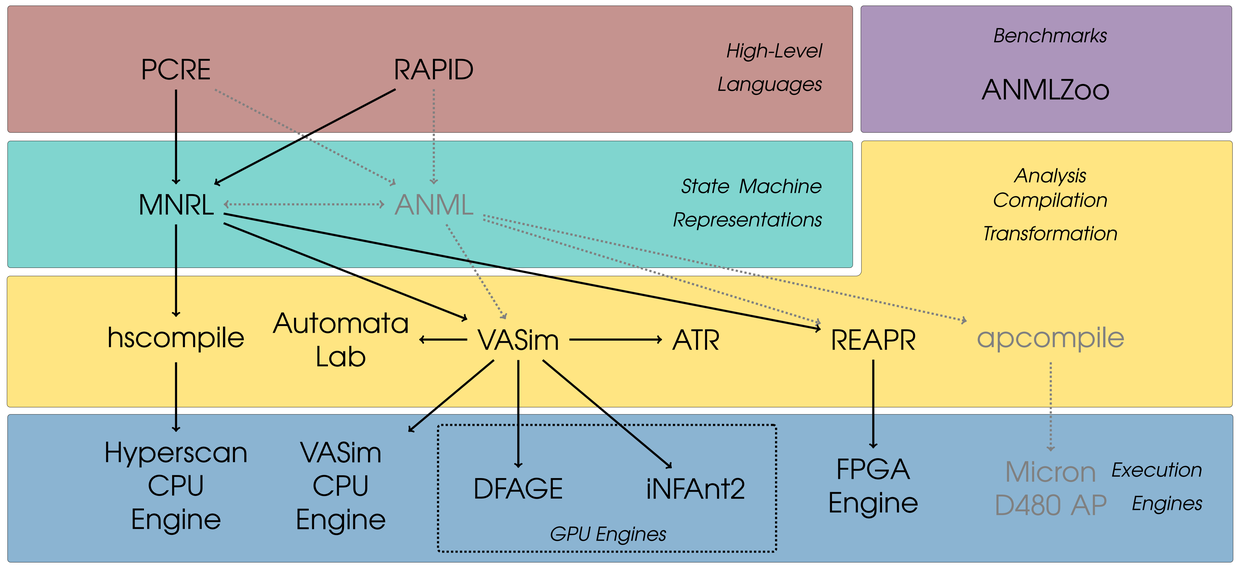
The MNCaRT Philosophy

- Previous research and tools were fractured by different file formats and restrictive licenses
- The **M**NRL **N**etwork **C**omputation **a**nd **R**esearch **T**estbed aims to open-source the end-to-end toolchain
- All tools operate on **common file format**
- Languages, Compilers, and Runtimes
- Support for all major architectures

MNRL: A Unified Representation

- **M**NRL **N**etwork **R**epresentation **L**anguage
- Flexible, JSON-based language for representing state machines
- MNRL file defines a collection of state machines (**network**) consisting of one or more **nodes**
- Direct support for validation of files
- APIs for both C++ and Python
- Easy to extend for experimental automata node types

MNCaRT Ecosystem



CPU Engines

- Hyperscan: state-of-the-art regular expression processing engine, extended to support MNRL networks
- VASim: supports execution of custom automata designs with experimental node types and collection of runtime statistics and profiles

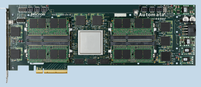
GPU Engines

- Support for both DFA (DFAGE) and NFA (iNFAnt2) execution on Cuda-enabled graphics cards
- Supports the study of trade-offs between the two execution models

REAPR FPGA Engine

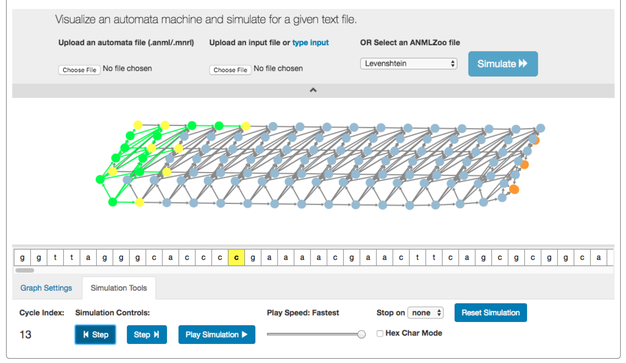
- MNRL networks are compiled to a custom module for execution on FPGAs (logic-based fabrics of LUTs and memory)
- Input/Output handled by Xilinx's SDAccel Framework

Micron D480 AP



- Prototype architecture for execution of automata
- Memory arrays used as computational medium
- Supported by converting MNRL to ANML representation

Automata Lab



Languages and Benchmarks

- Perl Compatible Regular Expressions
- RAPID: High-level language for defining inexact pattern matches
- ANMLZoo: Benchmark suite of diverse automata and regular expressions with associated input workloads

VASim & Automata-to-Routing

- Supports transformation, optimization, and execution of finite automata
- Prototype and test new automata designs and computational styles
- Experiment with place and route algorithms for custom spatial-reconfigurable architectures

Acknowledgements

This work was supported in part by grants from the NSF (CCF-1116673, CCF-1629450, CCF-1619123, CNS-1619098), AFRL (FA8750-15-2-0075), Jefferson Scholars Foundation, Achievement Rewards for College Scientists (ARCS) Foundation, a grant from Xilinx, and support from C-FAR, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of AFRL.

https://bit.ly/CAP-MNCaRT