Automata Processing

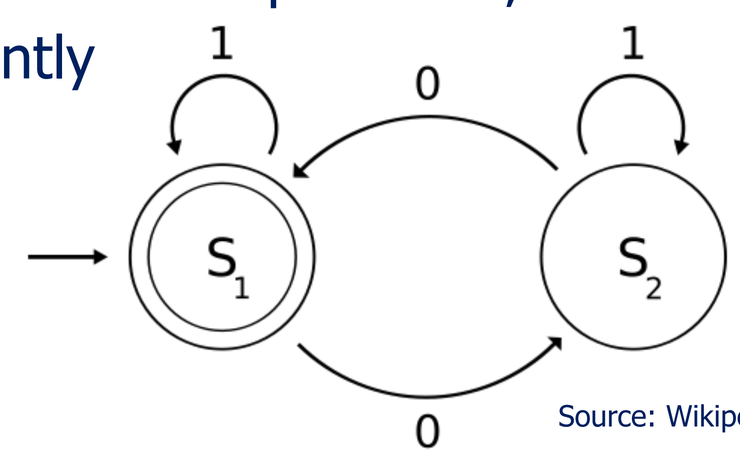
An introduction – From NFA theory to architecture

Ke Wang[1] (Research Scientist of CAP)
Matt Tanner[2]
Kevin Skadron[1] (Center Director of CAP)
Mircea Stan[1] (Assoc. Director of CAP)
[1] University of Virginia
[2] Micron Technology Inc.

16. September 2015

Finite Automaton

- *A finite automaton* is a set of states and transition rules that respond to input
- Recognizes *regular languages* – which can be very complex patterns, e.g. (1*01*01*)*
  - Many applications, e.g. fuzzy string search
- *Non-determinism* (NFA) allows multiple concurrent paths through the automaton
  - This is very powerful, handles combinatorial problems, checks many possibilities concurrently
  - Avoids exponential cost of DFA (deterministic finite automaton)
- Micron's AP adds counters, Boolean elements

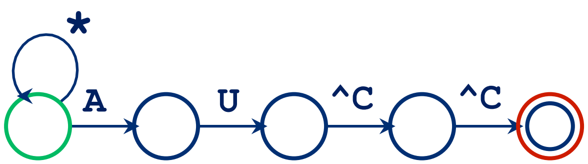


Source: Wikipedia

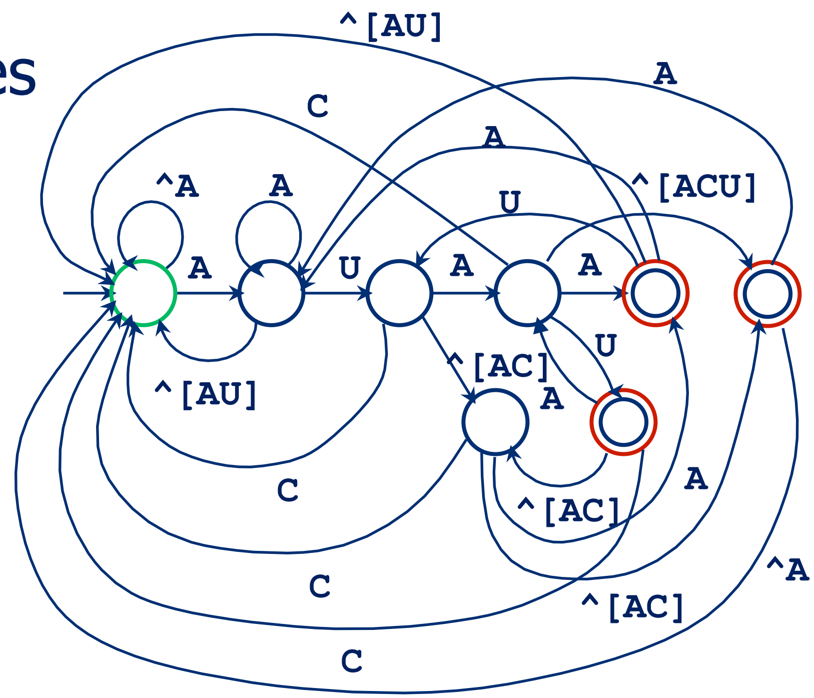
DFA accepting strings with even number of 0's

Automata Equivalence

- Any nondeterministic machine can be modeled as deterministic
  at the expense of exponential growth in the state count
  - Today's supercomputers model NFA as DFA, traversing every edge to find a solution. This leads to state explosion.
- SNORT example: 100 NFA nodes replace 10,000 DFA nodes



Nondeterministic Finite Automaton (NFA)



Deterministic Finite Automaton (DFA)

Micron

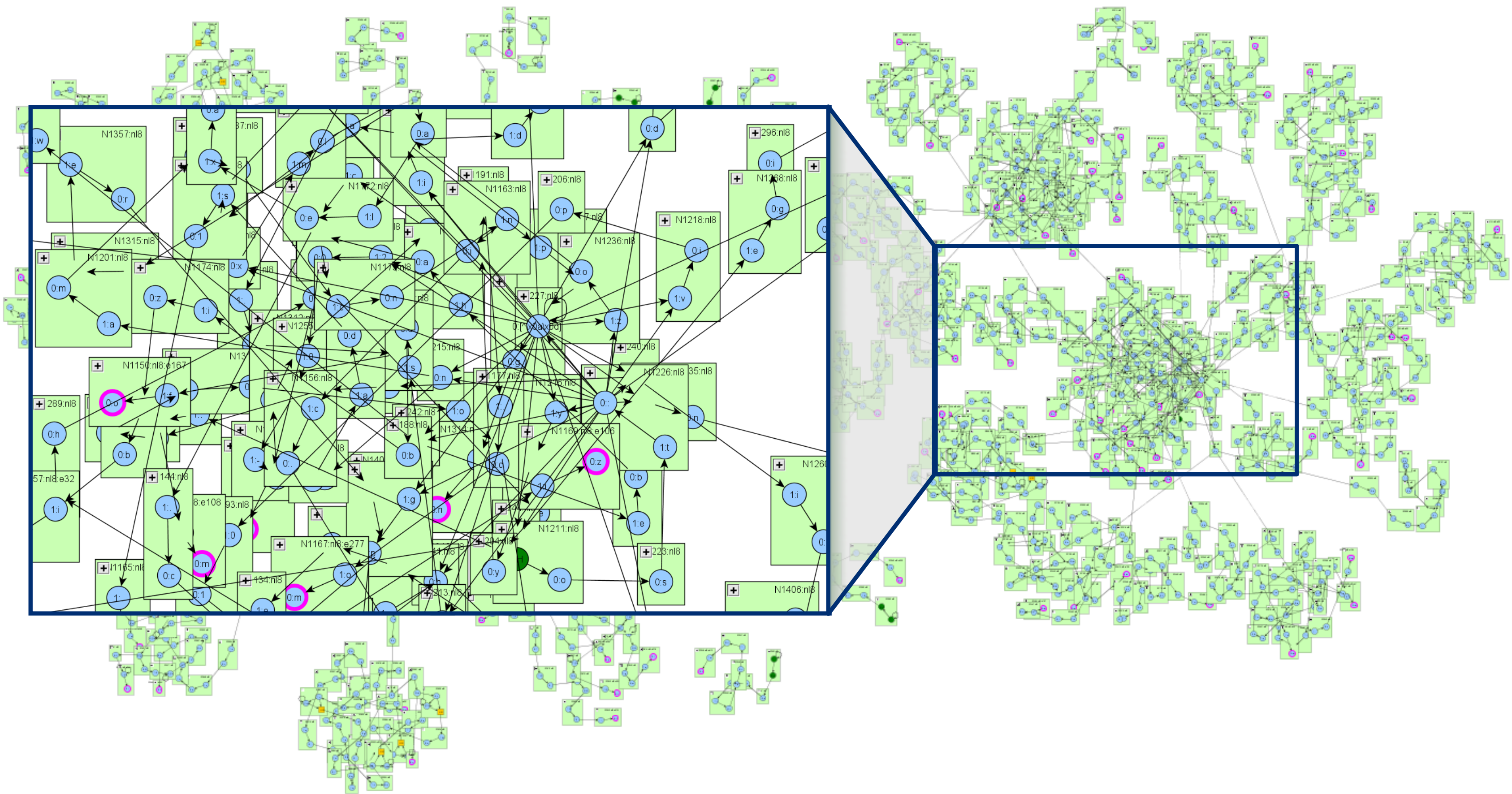
Introduction to Automata Processing

The Automata Processor (AP) is a programmable silicon device capable of performing very high-speed, symbolic pattern matching, allowing comprehensive search and analysis of complex, unstructured data streams.

- Hardware implementation of *non-deterministic finite automata* or *NFA* (plus some extra features)
- A massively parallel, scalable, reconfigurable, two dimensional fabric comprised of ~50,000 simple processing elements per chip, each programmed to perform a pattern matching and activation task each cycle
- Exploits the very high and natural level of parallelism found in DRAM
- On-board FPGA allows sophisticated processing pipelines

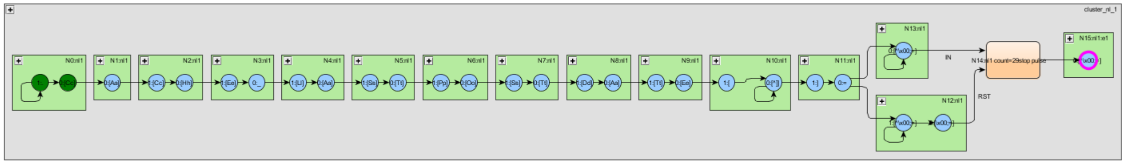
AUTOMATA PROCESSING

Unstructured Data – Unstructured Processor

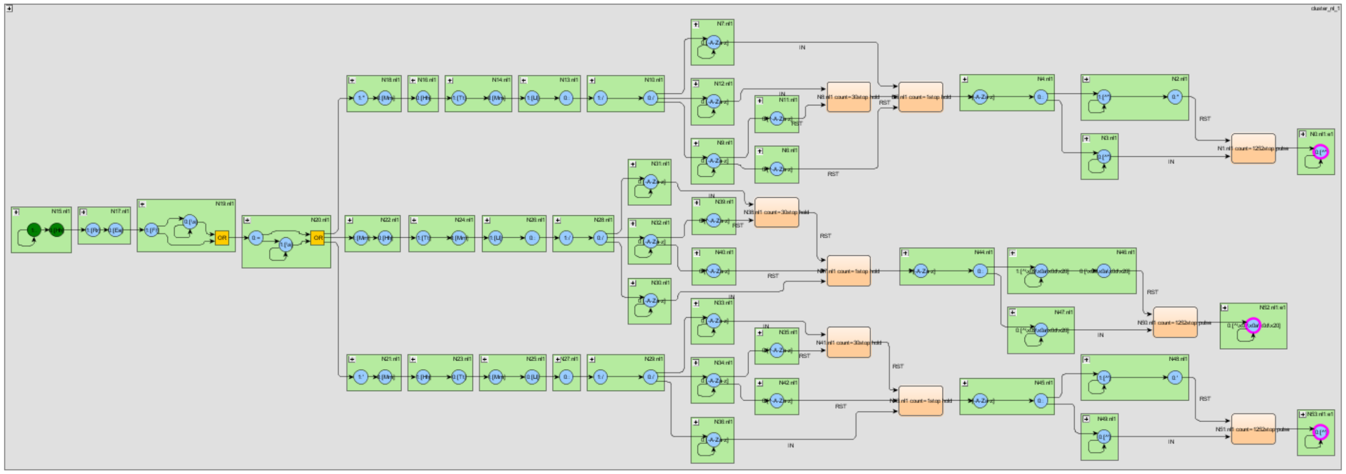


Parallel Automata

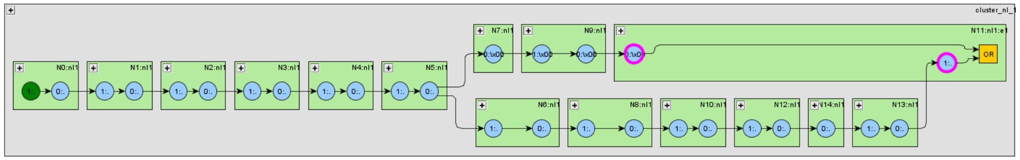
Pattern #1 →



Pattern #2 →

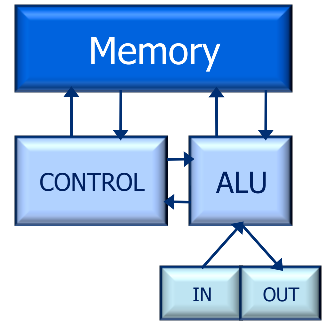


Pattern #3 →



Parallelization of automata requires no special consideration by the user. Each automaton operates independently upon the input data stream.

Non von Neumann Parallel Architecture



- AP avoids the von Neumann bottleneck of instruction fetch
  - Instead: hardware reconfiguration
- AP converts time complexity to space complexity
- AP allows massive parallelism
  - Every automaton node can inspect every input symbol
  - Can process a new input symbol every clock cycle
- Fills the unusual "MISD" role in Flynn's taxonomy

SISD	**SIMD**
MISD	**MIMD**

Many Layers of Parallelism

- Each STE: test many different symbol matches per cycle, per input symbol
  - Von Neumann (VN) architecture needs multiple instructions
- Across STEs: different matching rules for an input position
  - VN needs multiple instructions
- Multiple activations: branching—activate many potential successor paths
  - Non-determinism very difficult in VN; exp. growth in space complexity
- Multiple automata: independent rules
  - VN requires multiple threads, limited capacity
- Multiple streams
  - VN requires multiple threads

Problems Aligned with the Automata Processor

- AP strengths
  - Complex/fuzzy pattern matching
  - Combinatorial search space
  - Highly parallel set of analysis steps for each input item
  - Unstructured data, unstructured communication
    - Esp. with high fan-out/fan-in
  - These challenges are common in “big data” analytics!
- AP limitations
  - No arithmetic, only counting (but on-board FPGA can help)
  - Changing the “program” requires a reconfiguration step

Problems Aligned with the Automata Processor

Applications requiring **deep analysis** of **data streams** containing **spatial** and **temporal** information are often impacted by the **memory wall** and will benefit from the **processing efficiency** and **parallelism** of the Automata Processor

Network Security:
- Millions of patterns
- Real-time results
- Unstructured data

Bioinformatics:
- Large operands
- Complex patterns
- Many combinatorial problems
- Unstructured data

Video Analytics:
- Highly parallel operation
- Real-time operation
- Unstructured data

Data Analytics:
- Highly parallel operation
- Real-time operation
- Complex patterns
- Many combinatorial problems
- Unstructured data

So far: 10-100X+ speedups possible!

This tutorial

- Architecture of the AP
- Software ecosystem & AP Programming
- Applications