Micron Automata Processor
A Brief Introduction
## 1946-2006: The Consistent Message

<table>
<thead>
<tr>
<th>CPU Vendor</th>
<th>System OEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Memory is the bottleneck!”</td>
<td>“We need faster memory!”</td>
</tr>
</tbody>
</table>

### The Response

<table>
<thead>
<tr>
<th>Memory Industry</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Sure, we can do that!”</td>
</tr>
</tbody>
</table>

## 2007: Big Data Problems Begin…

<table>
<thead>
<tr>
<th>CPU Vendor</th>
<th>System OEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Memory is the bottleneck!”</td>
<td>“We need faster memory!”</td>
</tr>
</tbody>
</table>

### The Response

<table>
<thead>
<tr>
<th>Micron Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Why? Let’s dig into this a little more…”</td>
</tr>
</tbody>
</table>
Asking ‘Why?’ Took Us Back to the Beginning

- The modern relationship between processor and memory was conceived to avoid complications associated with **physical reconfiguration** of ENIAC.
- Since the mid 1940’s, most computer systems have been built on this basic architectural concept. The **role of memory** in systems was firmly cast.
- Micron concluded that important advancements can be made if we **challenge** this deeply rooted historical concept.
Challenge Everything

- Can memory be used for purposes other than storage of data and CPU instructions?
- Is it conceivable that the ‘Memory Wall’ is not actually a memory problem at all? What causes the Wall?
- Are traditional multi-core concepts the best way to achieve massive parallelism?
- Is software comprised of sequential machine instructions the best solution? Is this software even necessary?
- Can user-designed machines be more powerful than commercially developed CPU’s?
Introduction to Automata Processing

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

- Hardware implementation of non-deterministic finite automata (plus some extra features)
- A scalable 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 Micron’s semiconductor devices
- Addresses complex computational problems with unprecedented parallelism and performance
How Does it Work?

- Enables the FULL capacity of a ROW activation to be used on every memory cycle (~50,000 bit word size)
- Each memory element is an independent pattern-matching engine
- Provides the basis for massively parallel comparison operations and parallel random access memory cycle
- Implements high speed spatial & temporal pattern matching functionality
- Allows for complex automata to be implemented in parallel with self-synchronized operation
Automata Processor Hardware Building Blocks

State Transition Element (STE)  49,152
Counter Element  768
Boolean Logic Element  2,304
Nine Programmable Functions
Parallel event output “virtual pins”  6,144
Run-length Encoding

• Pattern-matching based compression technique
• Sequences (patterns) in which the same data value occurs in many consecutive data elements are stored as a single data value and count

• Setup a dictionary including all combinations of symbol and count as words
• When one word matched, report that word
  How to differentiate a5 and a6?

Only report after the consecutive symbols end
Run-Length Encoding

aabbcccccccccaabb

a2b2c4c4c1a2b2
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
- Unstructured data

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

**Data Analytics:**
- Highly parallel operation
- Real-time results
- Unstructured data
Network Traffic Analytics

- **Network Security**
  - Regular Expression signature traffic inspection
- **Service Provider Network Analytics**
  - Deep packet inspection to monetize traffic

<table>
<thead>
<tr>
<th></th>
<th>4-core CPU 2.4GHz*</th>
<th>Automata Processor**</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>192Mbps</td>
<td>1Gbps</td>
</tr>
<tr>
<td>Power</td>
<td>80W</td>
<td>4W</td>
</tr>
<tr>
<td>Cost</td>
<td>1x</td>
<td>0.1x</td>
</tr>
</tbody>
</table>

*Test benchmark: 534 complex PCRE rules with 35% edge traversal from Snort NIDS. Publication: “Evaluating Regular Expression Matching Engines on Network and General Purpose Processors” Becchi et al.

** Micron rule set compilation and performance estimate

Dr. Michela Becchi, a leading authority on performance analysis of pattern matching engines

UoM network security benchmark rule-set compiled on Micron Automata Processor
Cyber Security Application using PCRE

- Regular Expressions (REGEX’s) are used to concisely specify a set of symbol sequences, aka patterns
- REGEX’s use common operators to define these sets…
  - Boolean ‘OR’
  - Grouping
  - Quantification
  - Wildcards
  - Assertions
  - Escapes
  - Anchors
  - Modifiers
  - Classes

Infinite Number of Patterns

/ PICS-version\s+(\d{5,8}|\d(\x2e\d){10,})\s*\+\s+/
Patterns – A Graphical & Netlist View

Compiler converts expression to automata. Memory & Switch Programming Instructions

Netlist is created representing graph

Fully Programmed Device

Compiler converts expression to automata.
Natural Parallelization of Automatons

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

- Massively parallel problem space
  - Human genome mapping ~100 base pair reads to 3.2 billion base pair reference genome
  - Comparisons across genomes

Professor Srinivas Aluru is leading research on Automata Processors in bioinformatics applications

Prosite protein sequence patterns mapped to Micron Automata Processor
## Breakthrough Performance

<table>
<thead>
<tr>
<th>Planted Motif Search Problem</th>
<th>Automata Processor</th>
<th>UCONN - BECAT Hornet Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processors</td>
<td>48 (PCIe Board)+CPU</td>
<td>48 CPU (Cluster/OpenMPI)</td>
</tr>
<tr>
<td>Power</td>
<td>245W-315W(^1)</td>
<td>&gt;2,000W(^1)</td>
</tr>
<tr>
<td>Cost</td>
<td>TBD</td>
<td>~$20,000(^1)</td>
</tr>
<tr>
<td>Performance (25,10)</td>
<td>12.26 minutes(^2)</td>
<td>20.5 minutes</td>
</tr>
<tr>
<td>Performance (26,11)</td>
<td>13.96 minutes(^2)</td>
<td>46.9 hours</td>
</tr>
<tr>
<td>Performance (36,16)</td>
<td>36.22 minutes(^2)</td>
<td>Unsolved</td>
</tr>
</tbody>
</table>

- Planted Motif Search - a leading “NP Complete” problem in bioinformatics
- Solutions involving high match lengths and substitution counts are often presented to HPC clusters for processing
- Independent research predicts the Micron Automata Processor significantly outperforms a multi-core HPC cluster in speed, power and estimated cost

\(^1\) Micron Technology Estimates, Not including Memory of 4GB DRAM /Core
\(^2\) Research conducted by Georgia Tech (Roy/Aluru)
Automata Processor Hardware Concurrency

- Each column represents one STE
- Each row represents the response of every STE to a specific input symbol
- All operations for active elements performed concurrently
- Computes Next State Enable Vector for all state transitions and Match Vector for output on each cycle
- Each output can be routed to activate other STEs
- Has deterministic processing performance irrespective of the complexity of the automata
- Well suited for combinatorial problems

Next State Enable Vector

\[ E_t = f(M_{t-1}) \]
Automata Processor Multichip Scaling

Distributes the input data between all chips within a rank every cycle
Automata Processor Multichip Input Concurrency

With same number of chips: chose between multiple input stream concurrency or more automata capacity applied to fewer input streams.
Basic Specifications

- 1st Silicon has been fabricated in Micron’s 300mm manufacturing facility in Manassas, VA
- First generation device is produced on a trailing edge (50nm) commodity DRAM process

Basic Specs:
- 6.6T path decisions/second
- 4W Max TPD (estimated)
- 512 Entry State Vector Cache
- Inter-chip bus to facilitate scaling
- DDR3 physical interface
- < 150 mm² die size
Graph Connectivity Check, e.g. Social Networks

Check whether two nodes are connected to each other and how their distance from each other

Dijkstra’s algorithm with Fibonacci heap: $O(E + V \log V)$

AP implementation: $\min(E, V-1)$

Graph Connection

Node

Output

Routing

STE

Reporting element
Graph Connectivity Check

- Example checks the connection from b node to g node AP
- Can check many start-destination pairs in parallel
Automata Processor: Support & Tools

**PCIe Development Board**
- Industry Standard PCIe bus interface
- Capacity for up to 48 AP’s
- Large FPGA capacity
- DDR3 for local storage

**Software Development Kit**
- AP Optimization, loading & debugging tools

**Workbench Tool**
- Converts schematic automata to Micron ANML description language
Automata Processor Development Board
PCIe, 6 Ranks, 48 chips, 2.36M STEs

- Altera Stratix IV GX230 FPGA
- JTAG header
- AS header
- UDIMG socket M2
- Reset button
- 2x4 Power Connector
- SODIMM socket M4_0
- SODIMM socket M4_1
- PCIe Gen2 x8 Connector
- 2GB DDR3 M1
- SODIMM socket M3_0
- SODIMM socket M3_1
Automata Processor Software Integration

Compiler
- Hardware resource optimization
- Pre-compiled automata macros
- Automata either PCRE or ANML

Runtime
- Manages concurrent, independent input streams to multiple chips per rank and ranks per device
- Stops, swaps, and reloads input streams
- Dynamic loading of compiled automata
- Dynamic modification of automata characteristics
- Device Driver
Other Application Experience, Next Steps

- Have also implemented stack (i.e., memory) and cellular automata
  - Shows that AP is Turing-complete, much more powerful than NFA
- Encoding/decoding
- Graph algorithms
- Bioinformatics
- Frequent-set mining and other combinatorial problems
- Image feature detection looks promising
  - Treat object features as symbols in an alphabet
  - Treat variations as additional symbols
  - Look for related features as strings
  - Now we can treat this as a regular language
- New programming models, languages
Goals of the U.Va. Center

- Build critical mass of academic/industry collaborations
- Provide early access to cluster of Aps
- Training and programming support
- Develop foundational, open-source building blocks
- IP sharing
- Proposal teaming
- Clearing house for industry-academic funding
In Summary

• By reconsidering decades old architectural concepts, Micron has fully exploited the inherent parallelism found in memory
• This parallelism can be harnessed and focused on leading problems across multiple application domains
• The non-von-Neumann design of the Automata Processor delivers the rare “trifecta” of improved power consumption, performance and cost
• Micron will lead the industry through continued advancement and development of the AP architecture
• U.Va.’s CAC will be a nexus for AP research