



Micron's Automata Processor Architecture: Reconfigurable and Massively Parallel Automata Processing

Harold B Noyes
Senior Architect – Automata Processor Development

©2014 Micron Technology, Inc. All rights reserved. Products are warranted only to meet Micron's production data sheet specifications. Information, products, and/or specifications are subject to change without notice. All information is provided on an "AS IS" basis without warranties of any kind. Dates are estimates only. Drawings are not to scale. Micron and the Micron logo are trademarks of Micron Technology, Inc. All other trademarks are the property of their respective owners.

Outline

Introduction to Micron Technology

Five Technology Trends

Micron's Automata Processor Architecture

Scalability of the Automata Processor

Development Systems and Design Tools

Questions and Answers

Key Questions to Ask Yourself

Who is Micron Technology ?

What is an Automata Processor?

Why is a new architecture necessary?

NFA versus DFA – Why should you care?

What does the Automata Processor offer to systems?

What applications fit Automata Processor systems?

How do I design Automata Processor systems?



Micron at a Glance

Founded: October 1978, Boise, Idaho

FY2013 Net Sales: \$9.0 billion

NASDAQ Symbol: MU

Employees: ~30,000 worldwide

Products: We offer one of the world's broadest memory portfolios, including: DRAM components and modules, SSDs, NAND, and NOR, as well as other innovative memory technologies, packaging solutions, and semiconductor systems

Markets We Serve: Micron's products are designed to meet the diverse needs of computing, networking, server, consumer, mobile, automotive, and industrial applications

Patents: ~26,000 (~19,000 in force)

Expansive Product Offering

DRAM Technologies

SDRAM
DDR
DDR2
DDR3
DDR4
RLDRAM®
Mobile LPDRAM
PSRAM/
CellularRAM
Hybrid Memory Cube

DRAM Modules

FBDIMM
RDIMM
VLP RDIMM
VLP UDIMM
UDIMM
SODIMM
SORDIMM
Mini-DIMM
VLP Mini-DIMM
LRDIMM
NVDIMM

Bare Die

Multiple
Technologies

NAND Flash

TLC, MLC, SLC
Serial NAND
Enterprise NAND

Solid State Drives

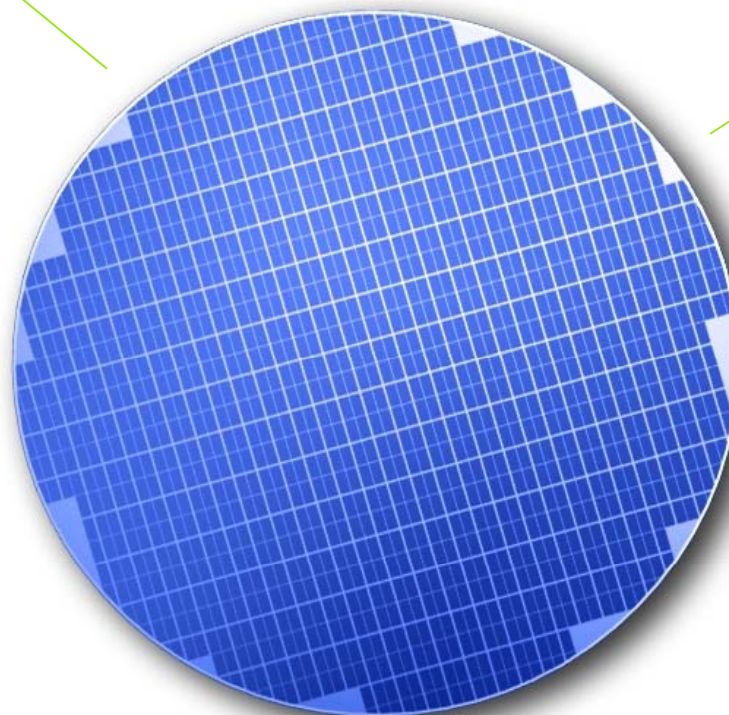
Client SSD
Enterprise SATA
Enterprise SAS
Enterprise PCIe

Managed NAND

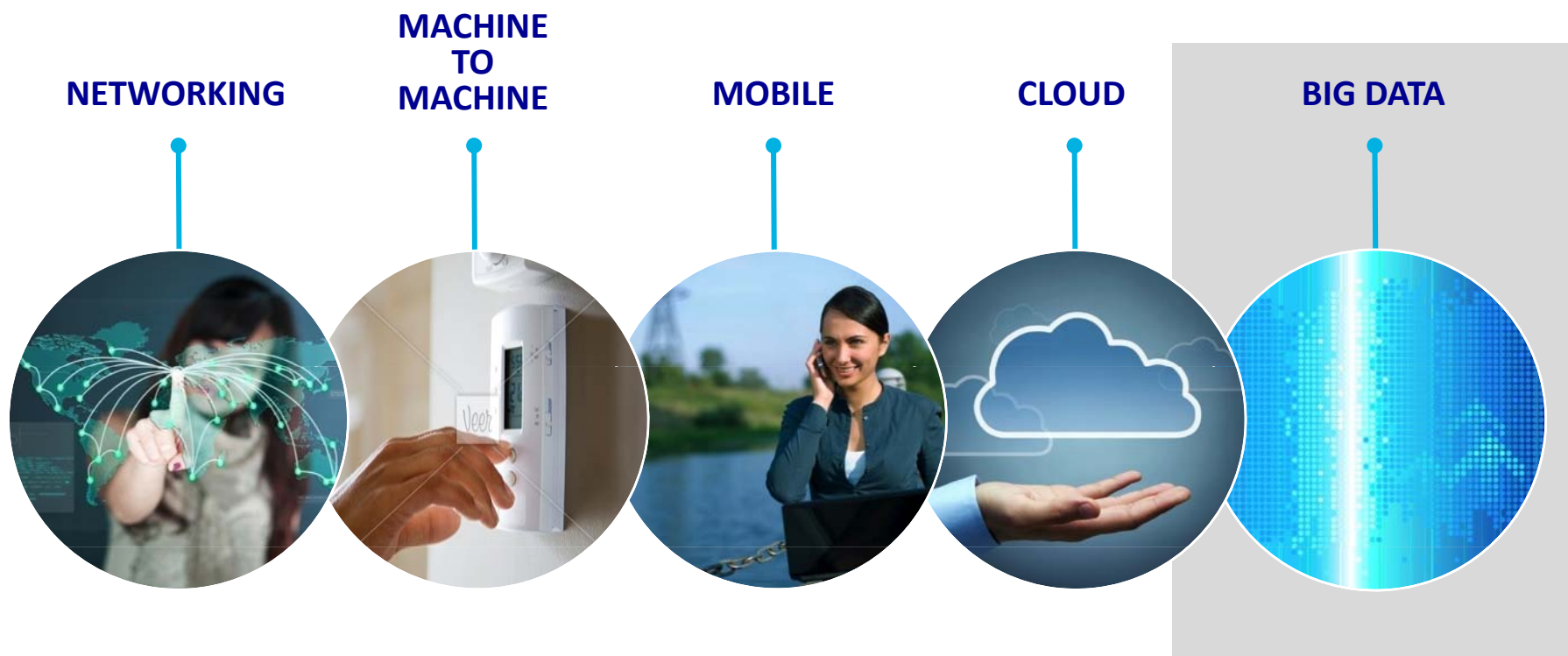
MCP
eMMC
ClearNAND® Flash
Embedded USB

NOR Flash

Parallel NOR
Serial NOR



Five Technology Trends



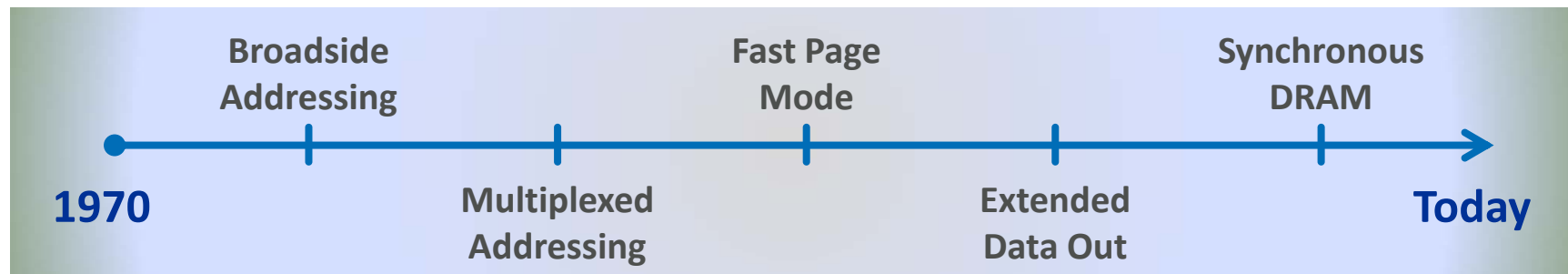
Big data presents unique challenges for memory systems

- Customers demand high performance for analytics
- Increasing levels of parallelism drive complexity in system architectures
- Massive scale requires aggressive power targets

A Repetitive Cycle

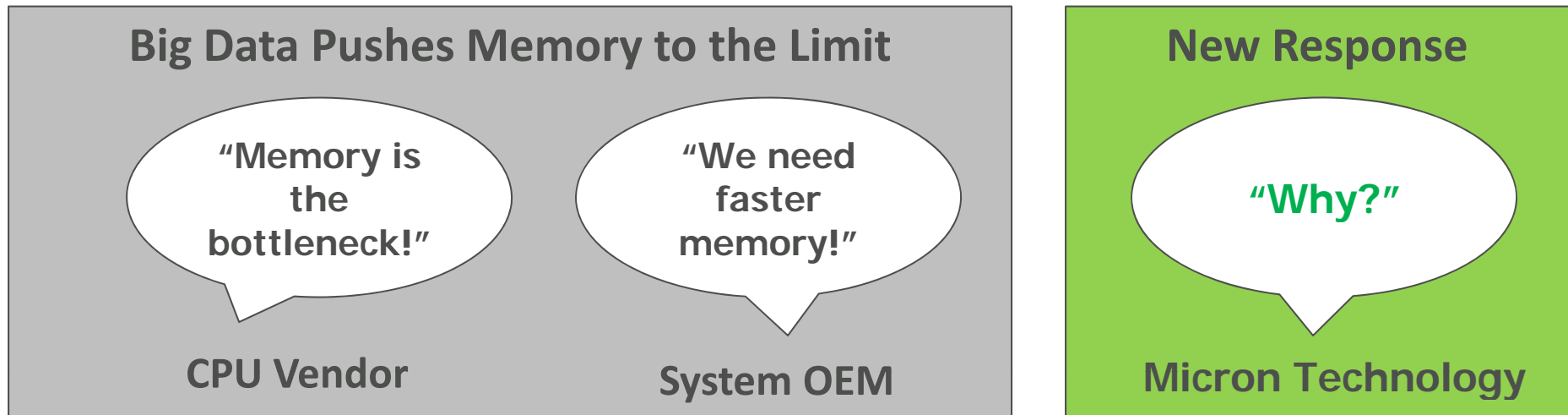


Innovations in memory interfaces...



... have been critical to improving performance.

Breaking the Cycle



- The modern relationship between processor and memory was conceived to avoid complications associated with the physical reconfiguration of ENIAC.
- Since the mid 1940s, 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.

Hybrid Memory Cube: A New Level of Performance

Revolutionary Approach to Break Through the “Memory Wall”

- Evolutionary DRAM roadmaps hit limitations of bandwidth and power efficiency
- Micron introduces a new class of memory: Hybrid Memory Cube
- Unique combination of DRAM on logic smashes through the memory wall

Unparalleled Performance

- Provides 15X the bandwidth of a DDR3 module
- Uses 70% less energy per bit than existing memory technologies
- Reduces the memory footprint by nearly 90% compared to today's RDIMMs



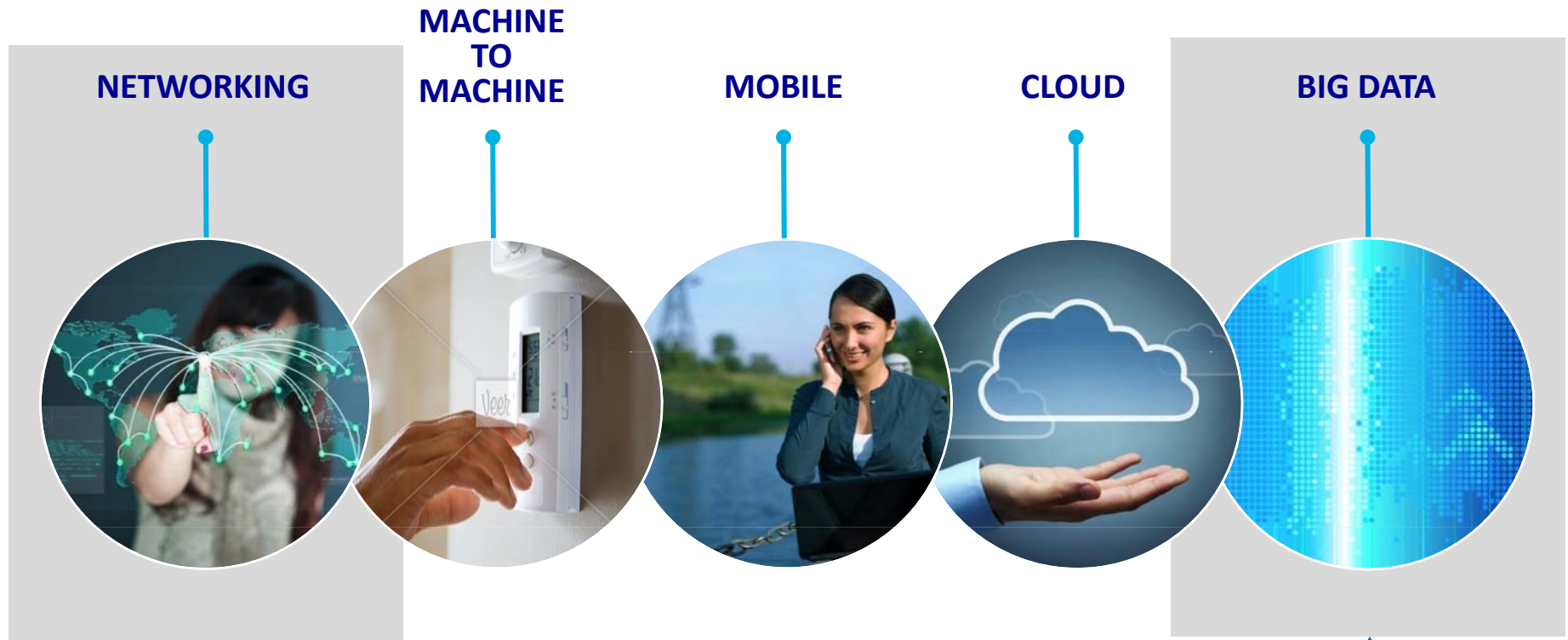
Key Applications

- Data packet processing, data packet buffering, and storage applications
- Enterprise and computing applications

How Did We Do It?

- Micron-designed logic controller
- High-speed link to CPU
- Massively parallel through-silicon via connection to DRAM

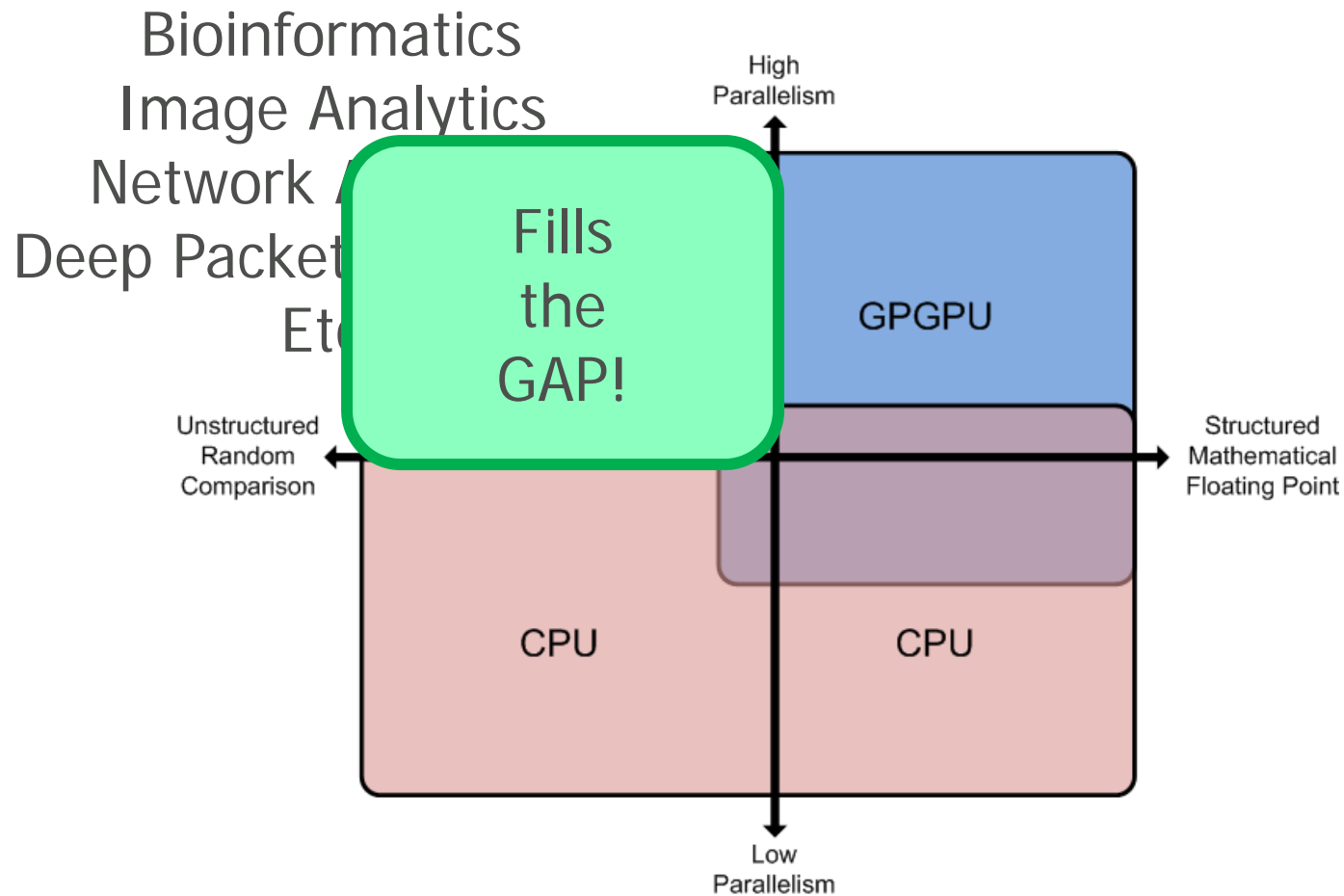
Five Technology Trends



Networking and big data present unique challenges for computing systems

- Inherently parallel and nondeterministic problems
- Massive data analyzed in thousands of ways

Automata Processor Technology ...



“An Efficient and Scalable Semiconductor Architecture for Parallel Automata Processing”

Dlugosch, P.; Brown, D.; Glendenning, P.; Leventhal, M.; Noyes, H., "An Efficient and Scalable Semiconductor Architecture for Parallel Automata Processing," *Parallel and Distributed Systems, IEEE Transactions on* ,
doi: 10.1109/TPDS.2014.8

- <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6719386&isnumber=4359390>

Example: Network Traffic Analytics

Network security

- Regular expression signature traffic inspection

Service provider network analytics

- Deep packet inspection to monetize traffic

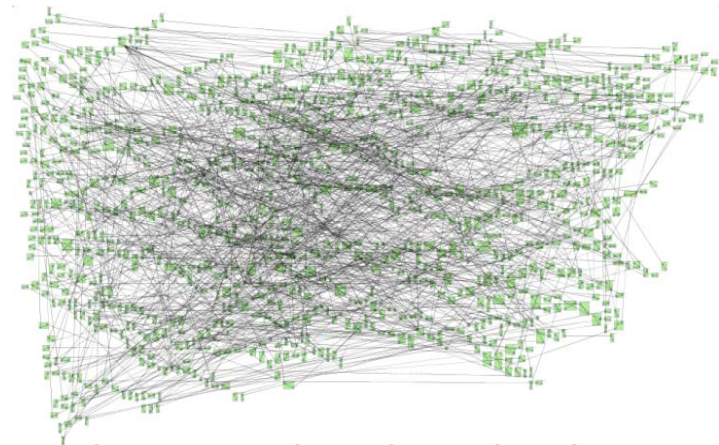


*Dr. Michela Becchi,
a leading authority
on performance
analysis of pattern
matching engines*

	Intel Xeon 2.4 GHz*	Automata Processor**
Performance	192 Mb/s	1 Gb/s
Power	80W	4W
Cost	1X	0.1X

*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



***UoM network security benchmark rule-set
compiled on Micron Automata Processor***

Example: Bioinformatics

Massively parallel problem space

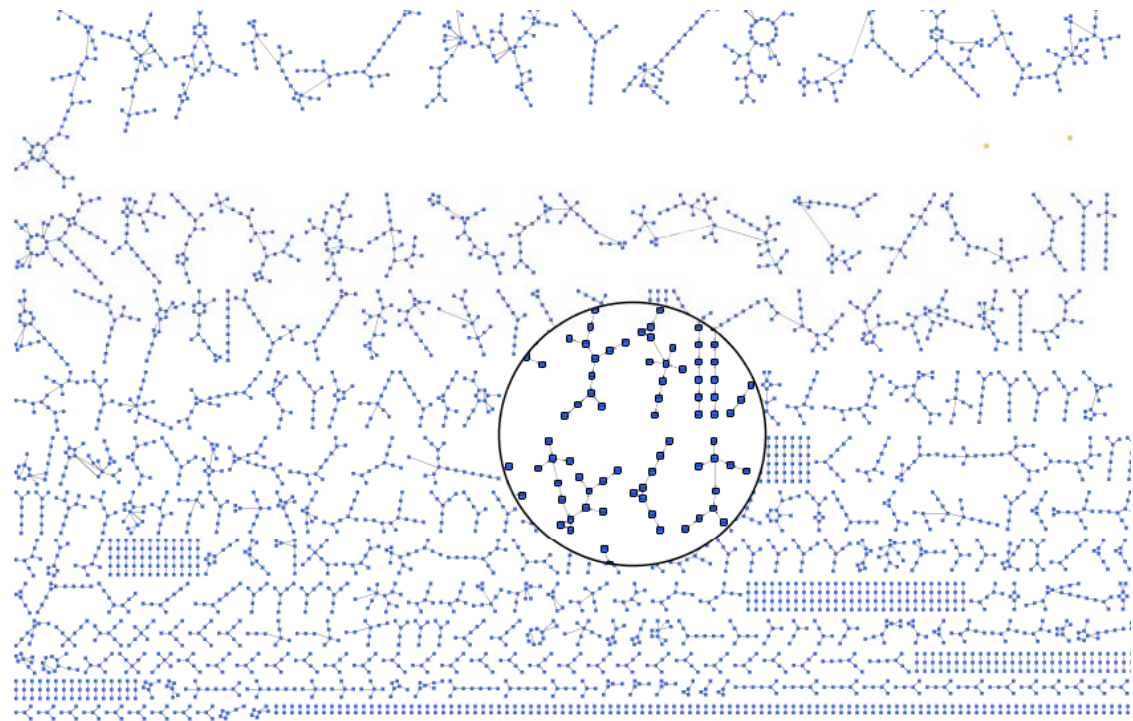
- Human genome mapping of ~10–100 base pair reads to a 3.2 billion base pair reference genome, as one example



**Georgia
Tech**



Professor Srinivas Aluru is a leading researcher for Automata Processors in bioinformatics applications



***Prosite protein sequence patterns mapped to
Micron's Automata Processor***

Breakthrough Performance

Planted Motif Search Problem	Automata Processor	UConn - BECAT Hornet Cluster
Processors	48 (PCIe Board)+CPU	48 CPU (Cluster/OpenMPI)
Power	245W-315W ¹	>2,000W ¹
Cost	TBD	~\$20,000 ¹
Performance (25,10)	12.26 minutes ²	20.5 minutes
Performance (26,11)	13.96 minutes ²	46.9 hours
Performance (36,16)	36.22 minutes ²	Unsolved

- PlantedMotif Search – A leading NP-hard problem in bioinformatics
- Solutions involving high match lengths (l) and substitution counts (s) are often presented to HPC clusters for processing [Performance (l,s)]
- Independent research predicts Micron's 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)

Serial vs. Parallel

Given these numbers:

10358

20013

9752

15171

Calculate the following:

Sum

Average

Standard Deviation

Multiply

Given this picture:



Answer the following:

Is it a dog or a cat?

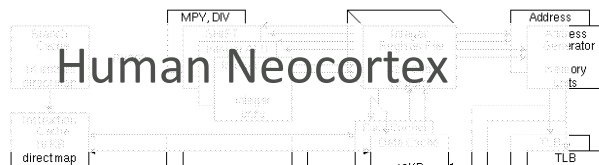
Is it young or old?

What color is it?

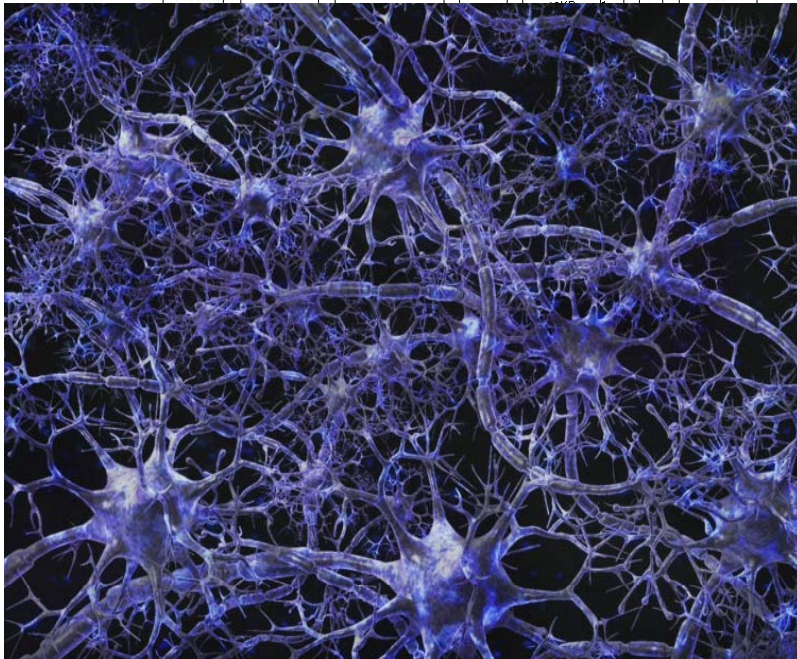
What else is in the picture?

A Fundamental Breakthrough

CPU Block Diagram



Human Neocortex

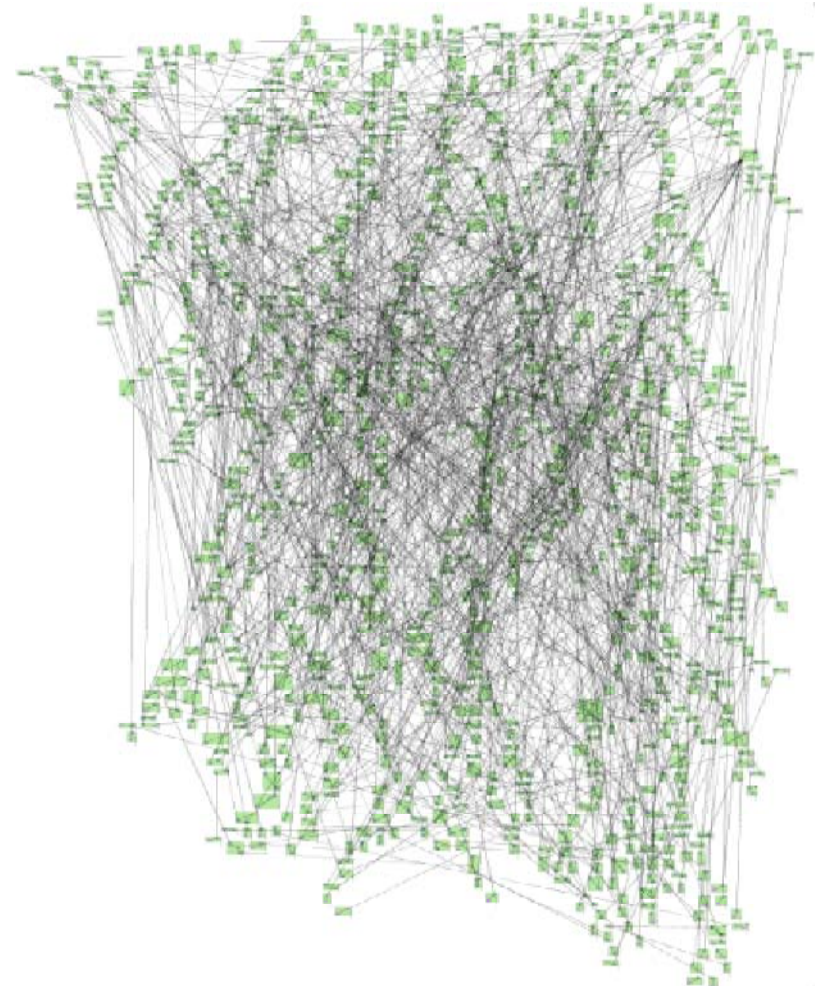


```
HERE: SOMP HERE ; stay in this loop  
END ; end of asm source file
```

Traditional PC

Single- or multi-core CPU

Automata Processor Program



The Automata Processor Architecture

The Parallelism of Memory Architectures

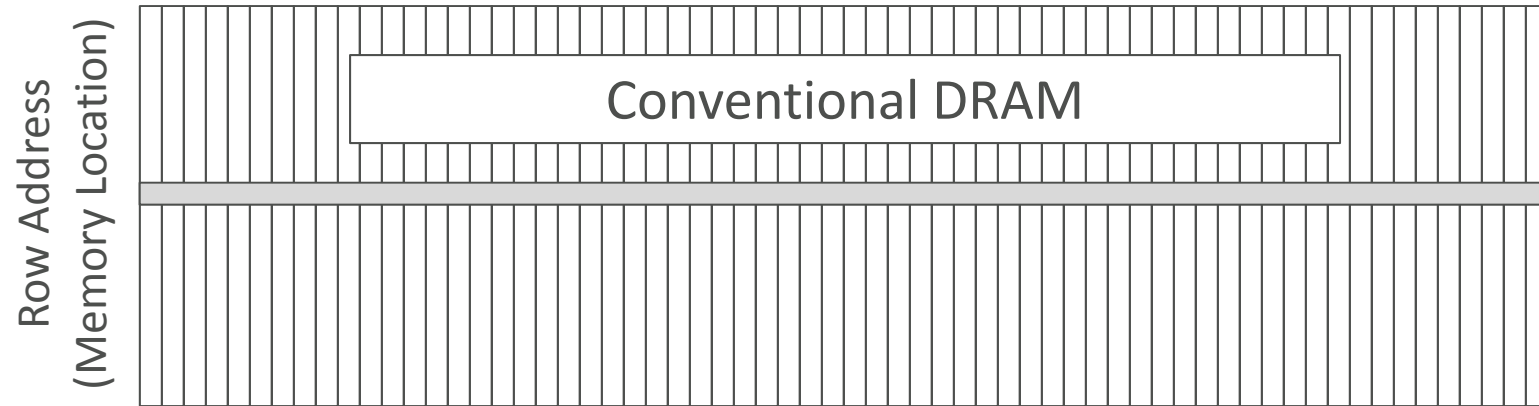
The Automata Processor Elements

The Automata Routing Matrix

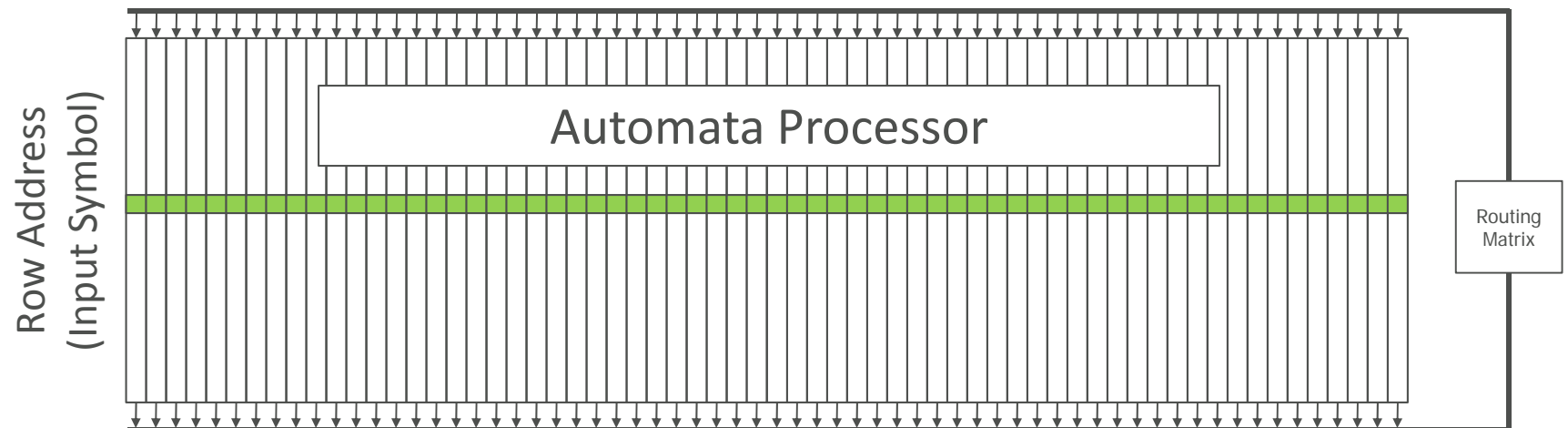
Scalability

System Development

Basic Operation Comparison: DRAM vs. Automata Processor



Each row access results in one word being retrieved from memory.

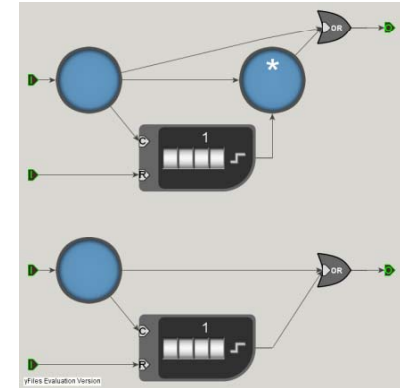


Each row access results in 49,152 match and route operations.

Automata Processor Architecture

Three active element classes:

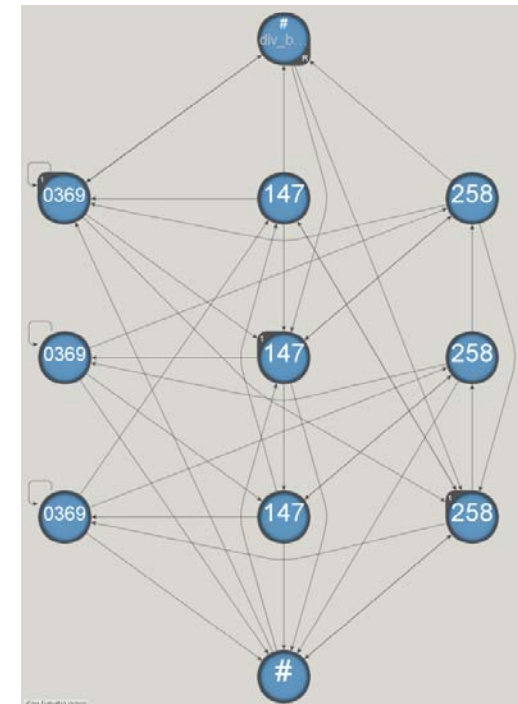
- State Transition Element (STE)
- Boolean Element (BOOL)
- Counter Element (CNTR)



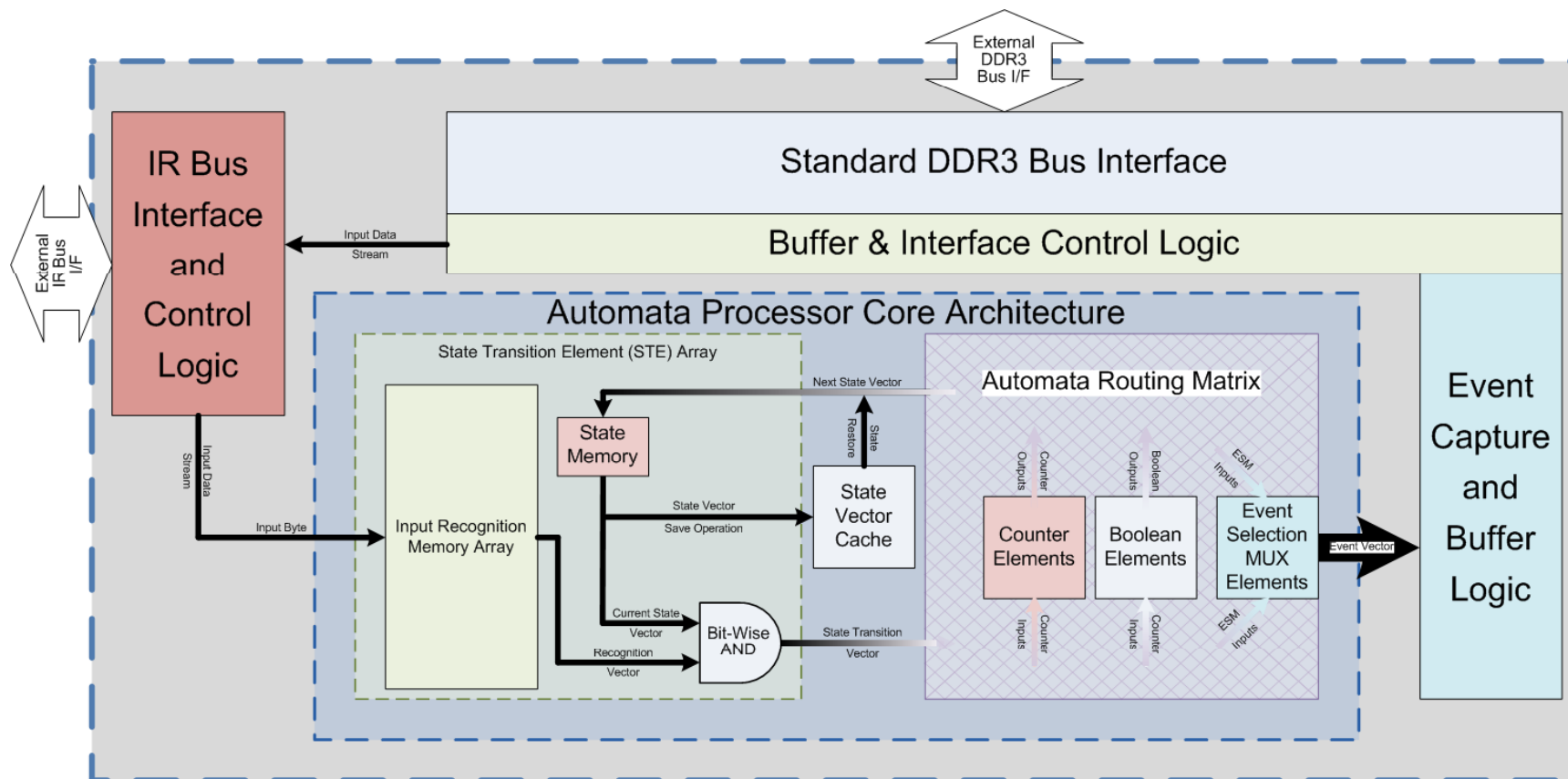
AP “computing machines” are built by connecting many of these active elements together, via the Automata Routing Matrix

Multiple independent automata can (and should!) coexist

All automata run in parallel



First-Generation Automata Processor: The D480



Basic Compute Cycle

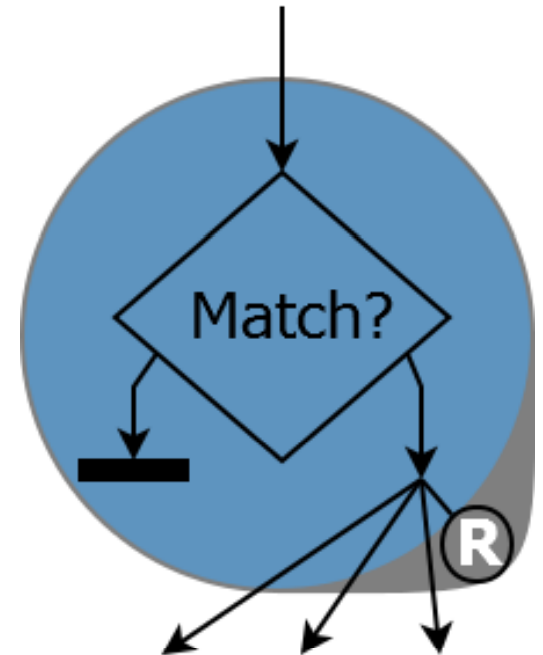
Initial state: One or more STEs (nodes) are active

Input symbol is presented to all STEs

Active STEs determine whether they match

Matching STEs

- Activate “downstream” STEs, via the ARM
- Inputs to CNTRs and BOOLs, via the ARM
- If it is a “reporting” STE, CNTR, or BOOL, an event vector is generated



Three Automata Processor Active Elements

State Transition Element (STE)

Counter Element (CNTR)

Boolean Element

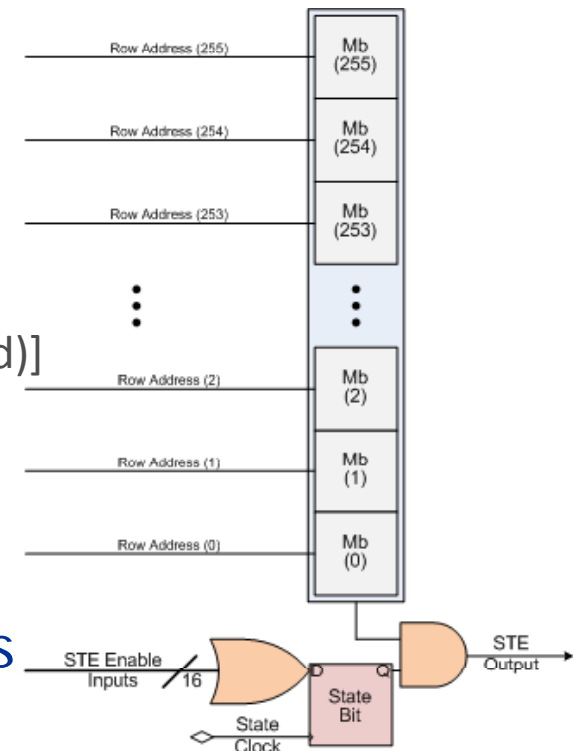
The State Transition Element (STE)

Processes the input symbol

Each STE contains:

- Input data value(s) to be **accepted**
[any and/or all of the possible 256 byte values]
- Current state memory (active/inactive)
- STE Output = (active state) AND (input accepted)]
- “Next state enable” input logic
[selects any of 16 possible inputs (logical OR)]

Each chip contains 49,152 (48K) STEs



The
AP Workbench
Icon
for an STE



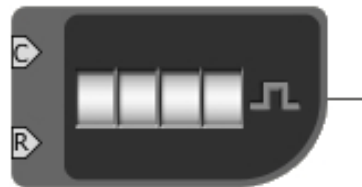
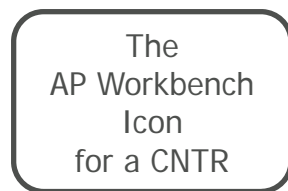
The Counter Element (CNTR)

Each CNTR contains:

- Target count value [12-bit value]
- Output mode selection [pulsed or latched output]
- Re-load mode selection [automatic or on-reset]
- “Count-enable” and “Reset” inputs
[selects any of 16 possible inputs for each (logical OR)]

CNTRs can be cascaded to realize count values up to 48 bits

One chip contains 768 CNTRs



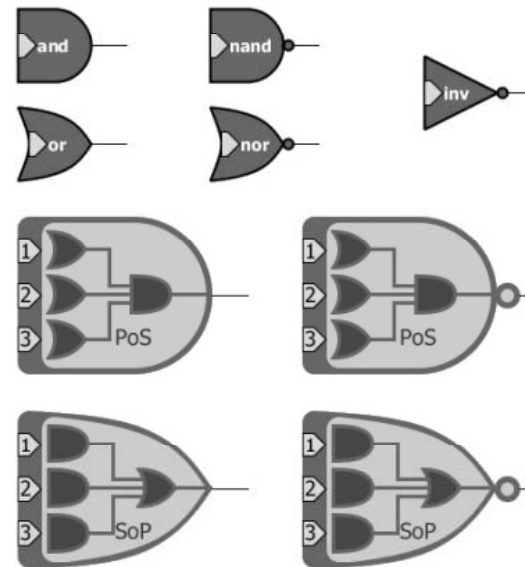
The Boolean Element (BOOL)

Each BOOL:

- Selects one of nine possible Boolean functions
- Selects any of 16 possible inputs
- Selectively implements an “end-of-data” AND of the output
- Is purely combinatorial logic; contains no state

One chip contains 2,304 BOOLs

The AP Workbench
Icons
for BOOL Functions



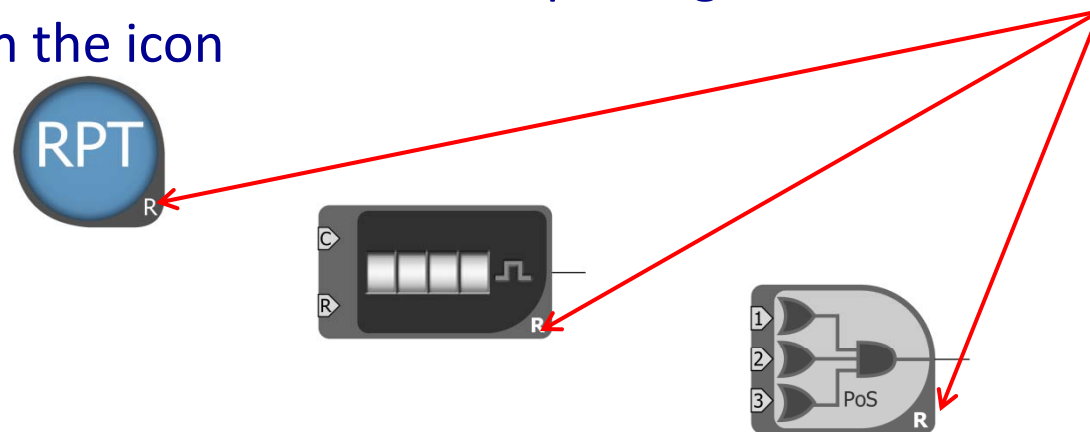
Event (Output) Selection

One chip can designate up to 6,144 signals as “events” to output

- Drives the selected signals as outputs of the Automata Processor
- Disables unused outputs
- Is purely combinatorial logic; contains no state

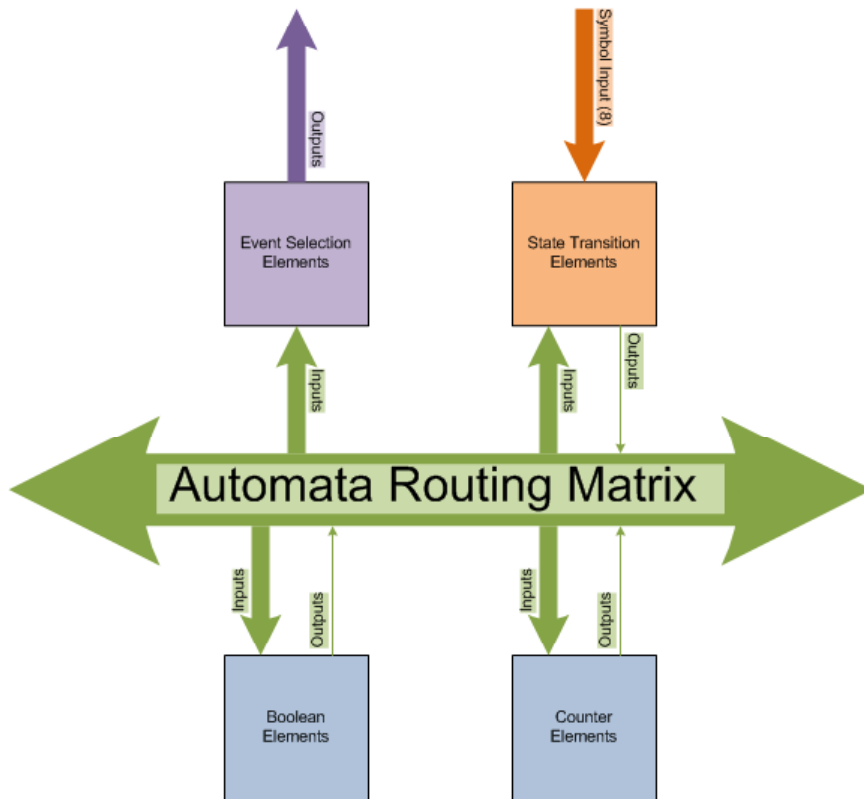
Fully programmable

AP Workbench identifies event reporting nodes with the “R” indicator in the icon



The Automata Routing Matrix (ARM)

A fully programmable “wiring connection” matrix



Connects the selected element outputs to the ARM signals (outputs from the STEs, CNTRs, and BOOLs)

Connects the selected ARM signals to the inputs of the elements (inputs to the STEs, CNTRs, and BOOLs)

Selects the event outputs

Automata Processor's ARM: Sufficient Connectivity Through Hierarchy

A proprietary matrix of buffers, switches, and routing lines implement the programmable Automata Routing Matrix (ARM)

The hierarchy:

- Elements are arranged in Rows

- Rows are arranged into Blocks

- Blocks are arranged into a grid

The Hierarchy: Rows of Elements

Each Row Contains:

16 STEs

1 BOOL *or* CNTR

2 Event outputs

BOOL Or CNTR	STE	STE	STE	STE	STE	STE	STE	STE	STE	STE	STE	STE	STE	STE	STE	STE
--------------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

The Hierarchy: Blocks of Rows

Each Block Contains:

16 Rows

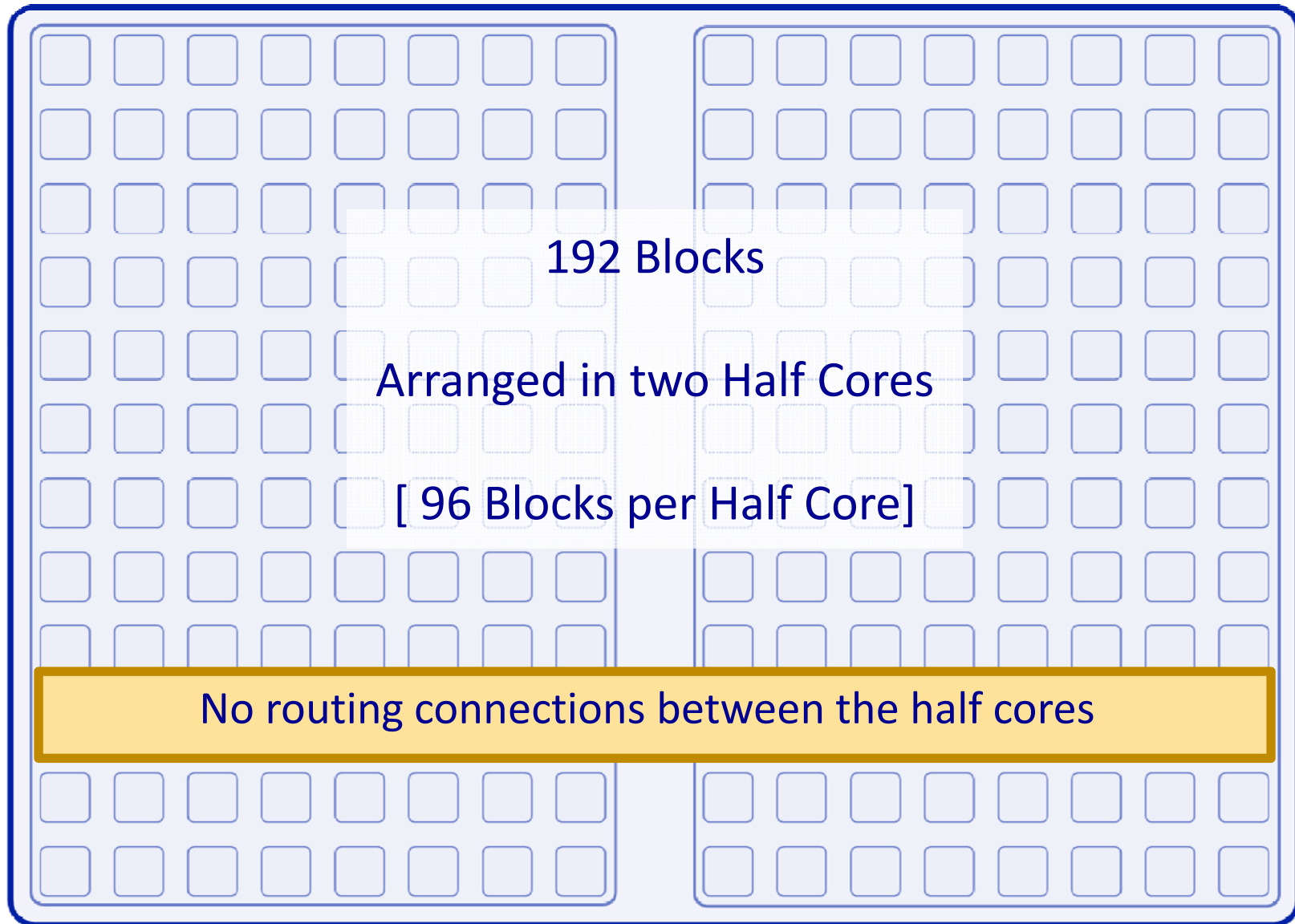
256 STEs

12 BOOLs

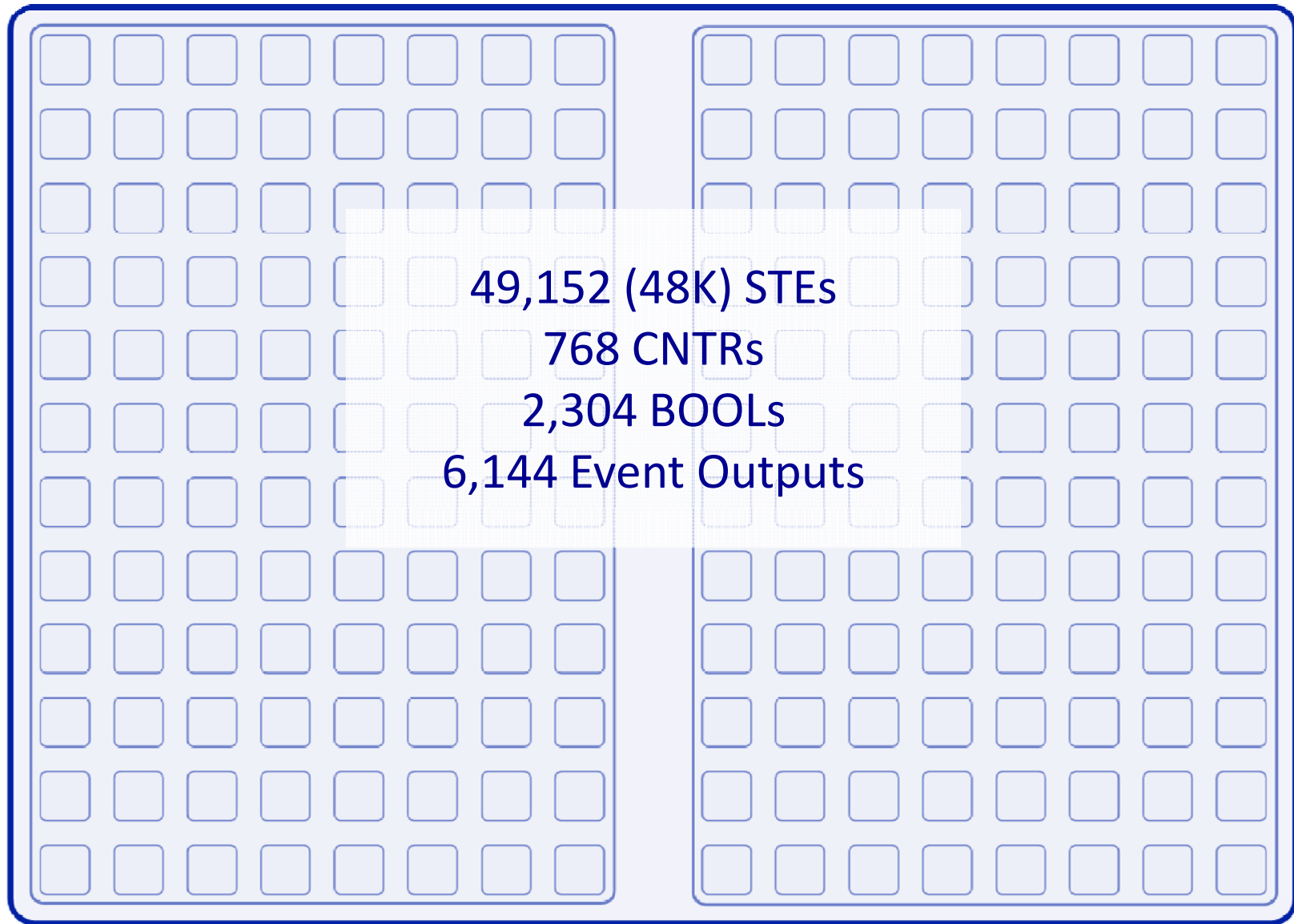
4 CNTRs

BOOL 11	STE 255	STE 254	STE 253	STE 252	STE 251	STE 250	STE 249	STE 248	STE 247	STE 246	STE 245	STE 244	STE 243	STE 242	STE 241	STE 240
BOOL 10	STE 239	STE 238	STE 237	STE 236	STE 235	STE 234	STE 233	STE 232	STE 231	STE 230	STE 229	STE 228	STE 227	STE 226	STE 225	STE 224
BOOL 9	STE 223	STE 222	STE 221	STE 220	STE 219	STE 218	STE 217	STE 216	STE 215	STE 214	STE 213	STE 212	STE 211	STE 210	STE 209	STE 208
BOOL 8	STE 207	STE 206	STE 205	STE 204	STE 203	STE 202	STE 201	STE 200	STE 199	STE 198	STE 197	STE 196	STE 195	STE 194	STE 193	STE 192
BOOL 7	STE 191	STE 190	STE 189	STE 188	STE 187	STE 186	STE 185	STE 184	STE 183	STE 182	STE 181	STE 180	STE 179	STE 178	STE 177	STE 176
BOOL 6	STE 175	STE 174	STE 173	STE 172	STE 171	STE 170	STE 169	STE 168	STE 167	STE 166	STE 165	STE 164	STE 163	STE 162	STE 161	STE 160
BOOL 5	STE 159	STE 158	STE 157	STE 156	STE 155	STE 154	STE 153	STE 152	STE 151	STE 150	STE 149	STE 148	STE 147	STE 146	STE 145	STE 144
BOOL 4	STE 143	STE 142	STE 141	STE 140	STE 139	STE 138	STE 137	STE 136	STE 135	STE 134	STE 133	STE 132	STE 131	STE 130	STE 129	STE 128
BOOL 3	STE 127	STE 126	STE 125	STE 124	STE 123	STE 122	STE 121	STE 120	STE 119	STE 118	STE 117	STE 116	STE 115	STE 114	STE 113	STE 112
BOOL 2	STE 111	STE 110	STE 109	STE 108	STE 107	STE 106	STE 105	STE 104	STE 103	STE 102	STE 101	STE 100	STE 99	STE 98	STE 97	STE 96
BOOL 1	STE 85	STE 84	STE 83	STE 82	STE 81	STE 80	STE 79	STE 78	STE 77	STE 76	STE 75	STE 74	STE 73	STE 72	STE 71	STE 70
BOOL 0	STE 69	STE 68	STE 67	STE 66	STE 65	STE 64	STE 63	STE 62	STE 61	STE 60	STE 59	STE 58	STE 57	STE 56	STE 55	STE 54
CNTR 3	STE 53	STE 52	STE 51	STE 50	STE 49	STE 48	STE 47	STE 46	STE 45	STE 44	STE 43	STE 42	STE 41	STE 40	STE 39	STE 38
CNTR 2	STE 37	STE 36	STE 35	STE 34	STE 33	STE 32	STE 31	STE 30	STE 29	STE 28	STE 27	STE 26	STE 25	STE 24	STE 23	STE 22
CNTR 1	STE 21	STE 20	STE 19	STE 18	STE 17	STE 16	STE 15	STE 14	STE 13	STE 12	STE 11	STE 10	STE 9	STE 8	STE 7	STE 6
CNTR 0	STE 5	STE 4	STE 3	STE 2	STE 1	STE 0										

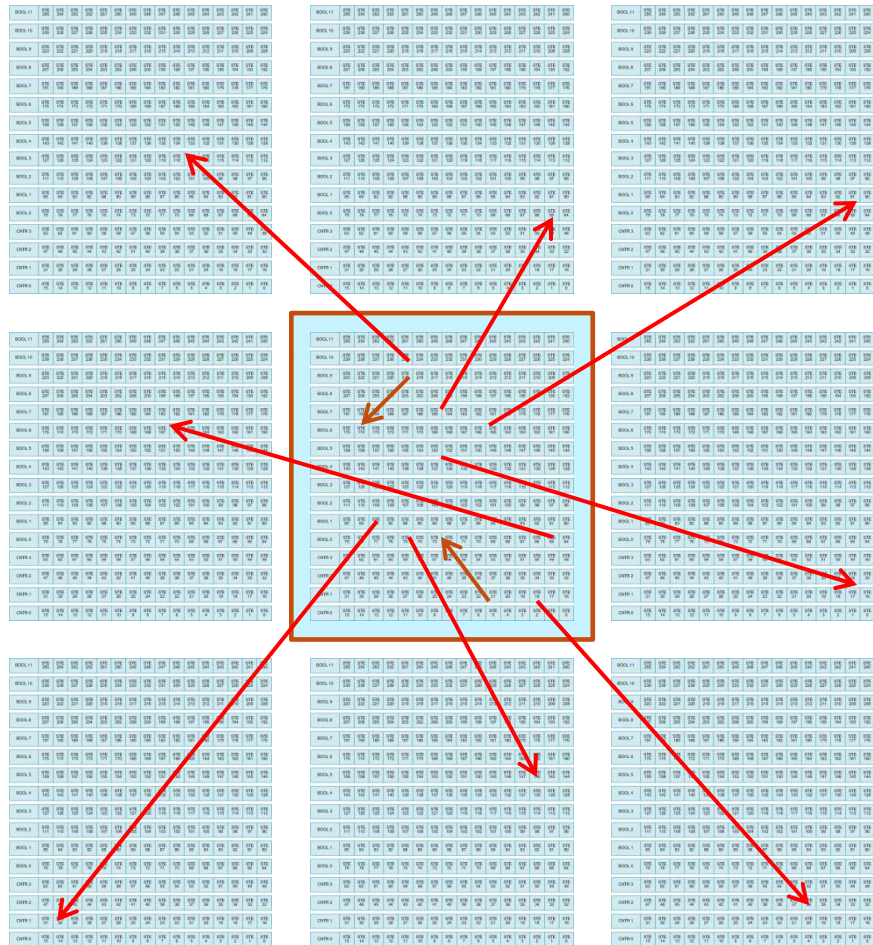
The Hierarchy: Grid of Blocks



The Automata Processor



Automata Routing Matrix Connectivity



- Elements (STEs, CNTRs, BOOLs) may route to *any* other element:
 - **Within** its block and/or
 - Any of the **8 adjacent** blocks
- Elements may route to multiple other elements, simultaneously
- Each element can potentially route to *any of 2,304 elements*, including itself
- Longer routes are possible, if willing to sacrifice performance

Automata Processor Scalability

Scaling for Automata Capacity

- Design multiple chips into ranks, similar to memory ranks
- Design with multiple ranks

Scaling for System Performance

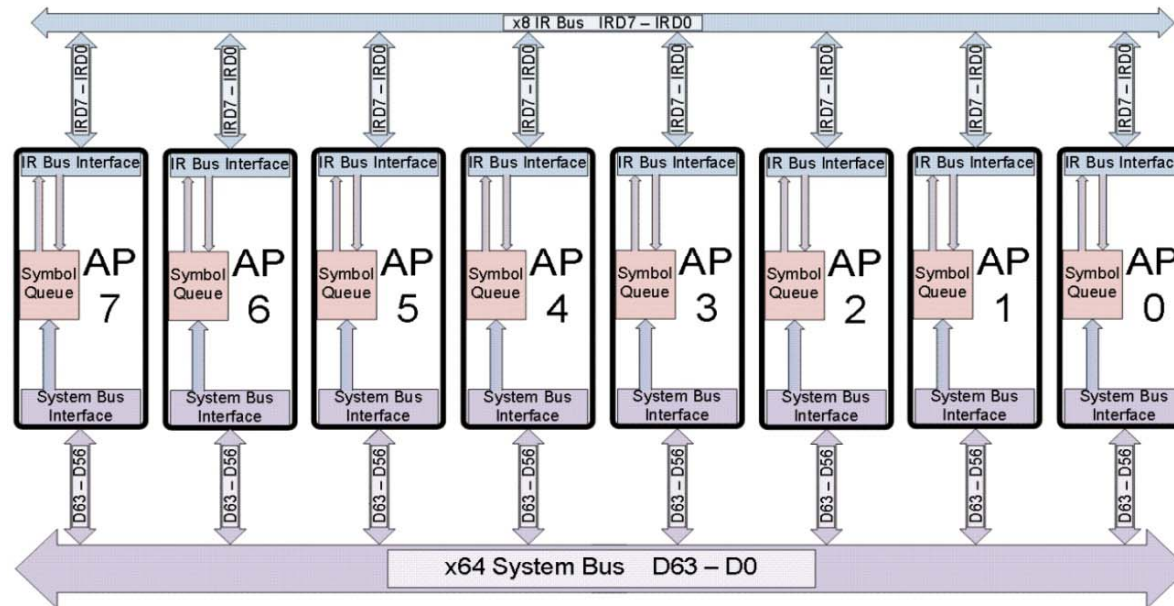
- Design for multiple logical Automata Processors
 - Each logical processor runs a separate data set
 - Logical processors can implement different automata
 - Each rank can be divided into as many as four logical processors

The intra-rank (IR) bus distributes a unique data set to each logical processor

All logical processors within the rank run in parallel

- Multiple ranks of automata processors **are** multiple logical processors, managed separately by the drivers, and run in parallel

Automata Processor: Multi-chip Scalability



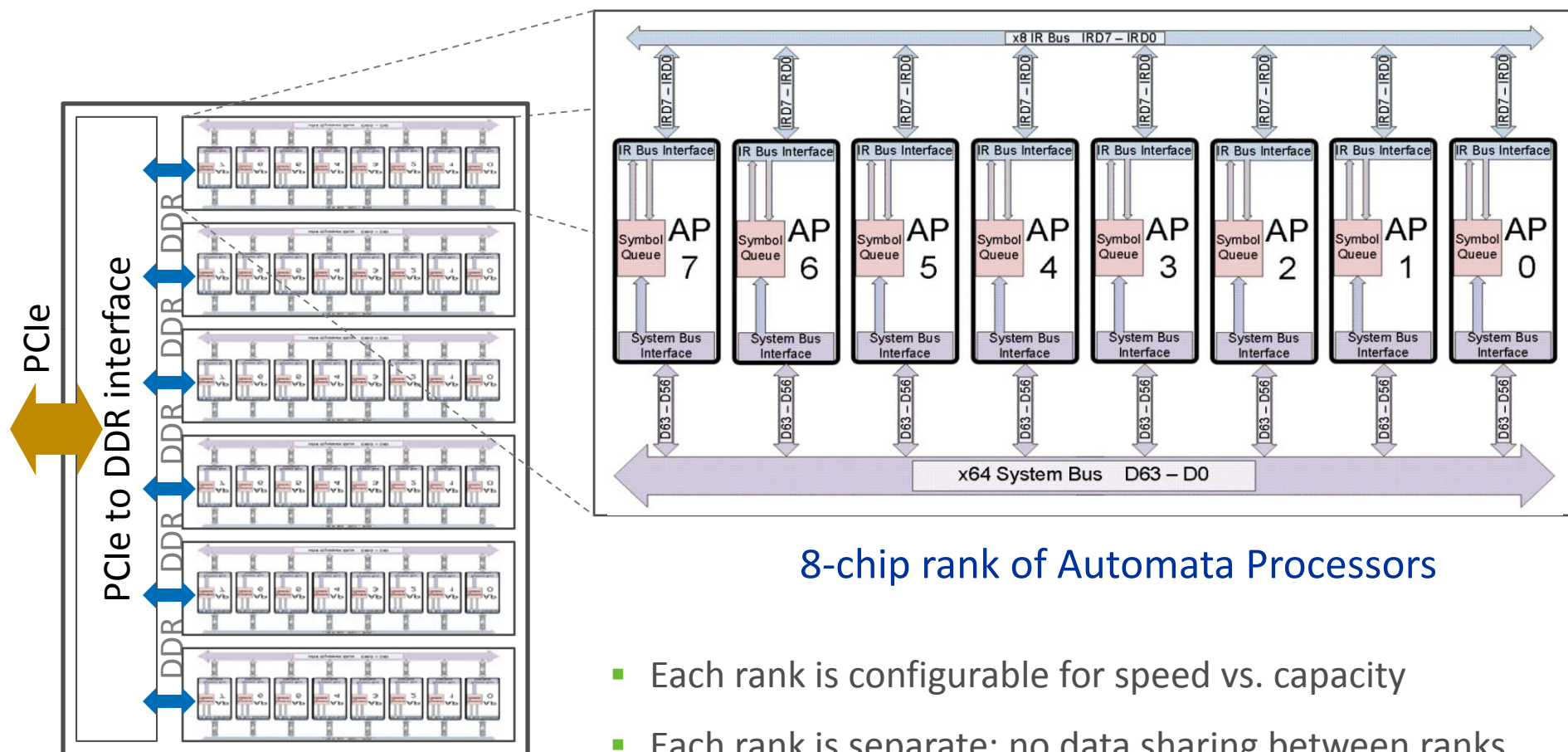
Example: One 64-bit rank of Automata Processor chips

Configurable for speed vs. capacity

- Operate as 1, 2, or 4 parallel machines (384K, 192K, or 96K STEs)
- Effective throughput up to 1.0 Gb/s, 2.0 Gb/s, or 4.0 Gb/s
- Utilizes the full 64-bit DDR3 bus
- Shares data among chips, through the intra-rank (IR) bus

Host controller must manage data streams

Automata Processor: Multi-Rank Scalability



8-chip rank of Automata Processors

- Each rank is configurable for speed vs. capacity
- Each rank is separate; no data sharing between ranks
- Host controller must manage streams

The Results of the Automata Processor's Scalability



MICROPROCESSOR *report*

Insightful Analysis of Processor Technology

"Micron's 48-chip evaluation board scales this bandwidth to a ridiculous 38 TB/s, which enables Automata to solve problems that traditional processors cannot."

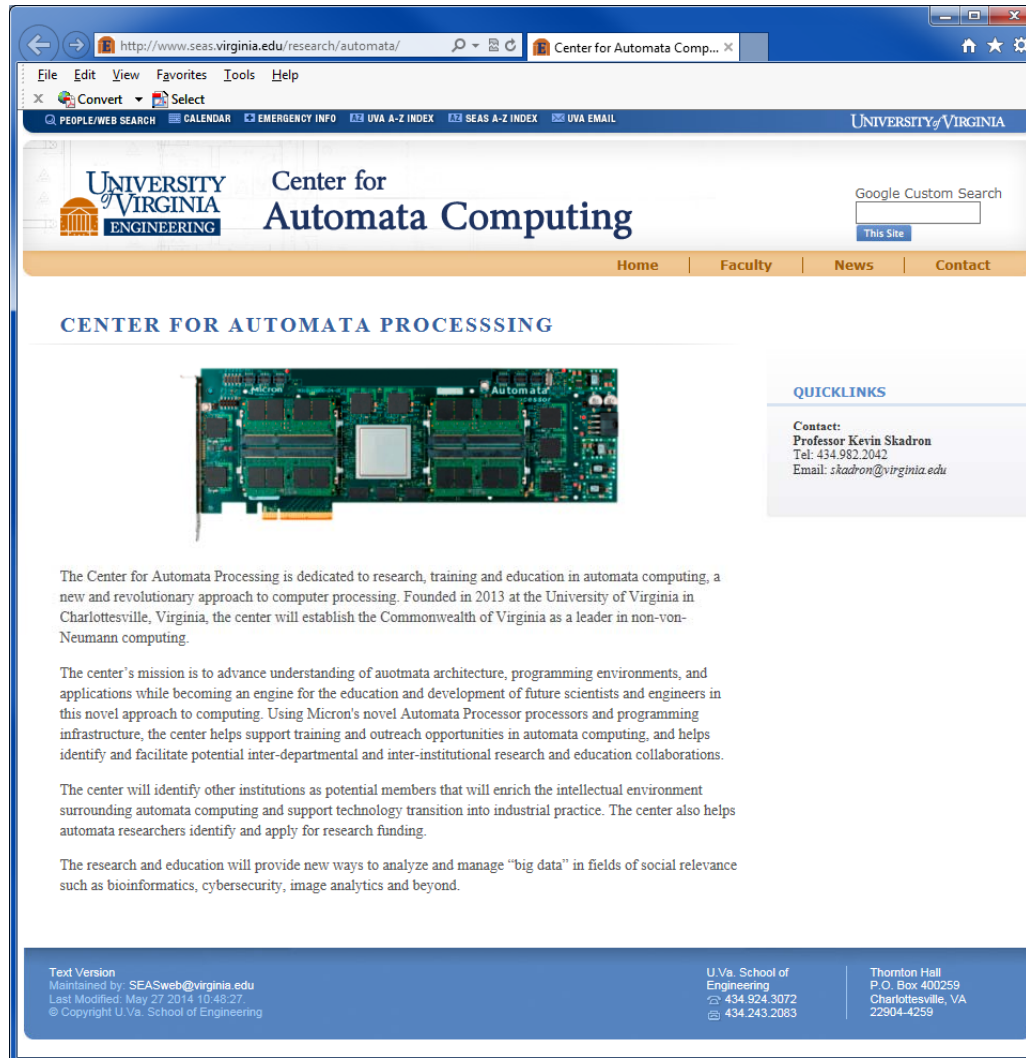


Online Resources



[Micron's Innovations Page: Automata Processor](#)

The University of Virginia's Center for Automata Computing



[Center for Automata Computing](#)

Development Cycle

Design solution and implement in PCRE or ANML

Simulate, debug, and improve design

Create hardware programming file

- Compiler performs placement of core structures
- Compiler routes connections through ARM
- Compiler creates hardware programming file

Load hardware programming file into the AP chip(s)

Runtime

Write input data to the AP chip

Handle event vectors (can be interrupt-driven or polled)

Software Development Kit (SDK)

Automata Processor Workbench (AP Workbench)

Command line tools

- Emulator
- RegExCompiler
- ANML Compiler

C language API

- Runtime
- Development
- Emulation

Python interface to C API functions

Linux device driver for PCIe dev board

Designing the Automata

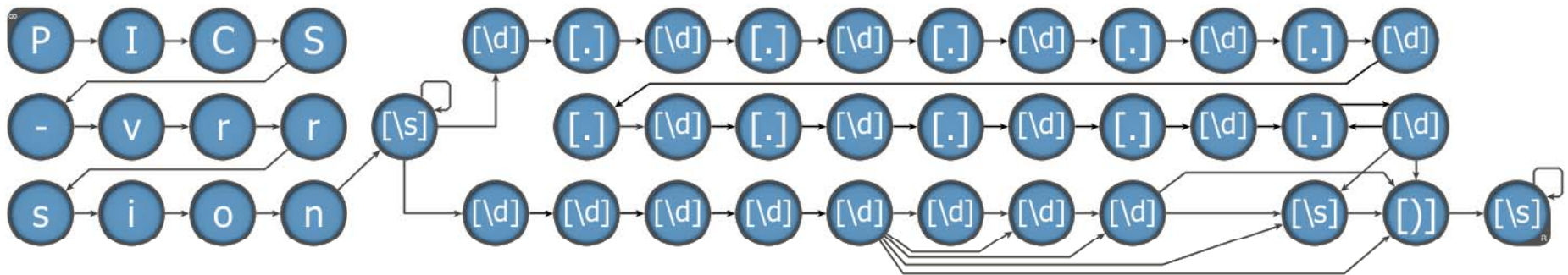
- ▶ **Two languages supported**

- ▶ Regular Expressions (PCRE)
- ▶ Automata Network Markup Language (ANML)

- ▶ **An example regular expression (taken from SNORT):**

`/PICS-version\s+(\d{5,8}|\d(\x2e\d){10,})\s*\x29\s+/\`

- ▶ **The compiled output is shown below:**



Automata Network Markup Language (ANML)

- ▶ ANML is a language for describing automata, created by Micron
 - ▶ ANML is an XML language for describing the composition of automata networks
 - ▶ Syntax of ANML documents is rigorously defined by an XML Schema
- ▶ An example of the ANML representation, of an automaton, is:

```
<?xml version="1.0" encoding="UTF-8"?>
<automata-network name="SNORT w STEs" id="SNORT w STEs">

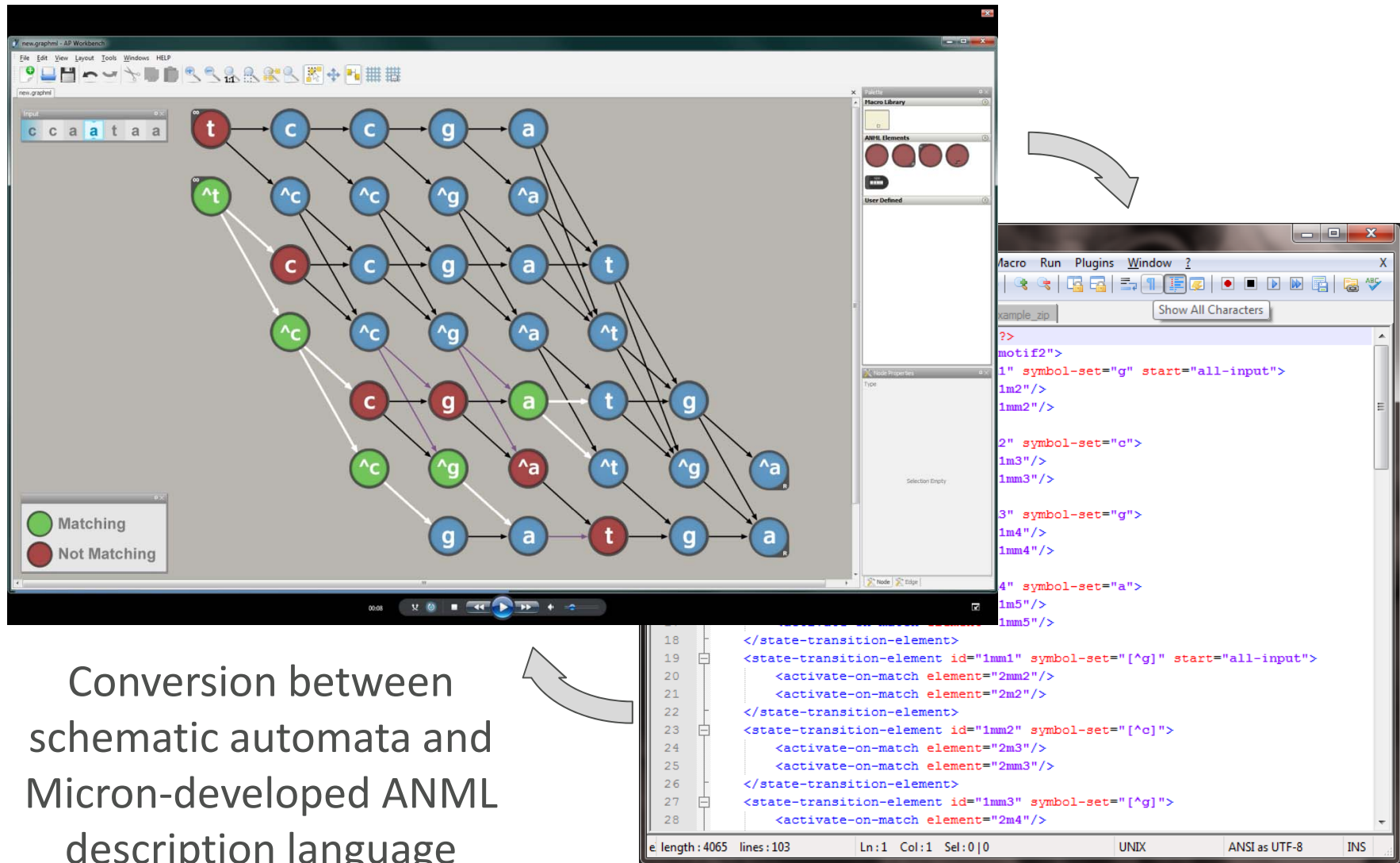
  <description>
    This implements the Regular Expression:
      /PICS-version\s+(\d{5,8}|\d(\x2e\d){10,})\s*\x29\s+ /
    in Micron's ANML format.
  </description>

  <state-transition-element id="_2" symbol-set="P" start="all-input">
    <activate-on-match element="_3"/>
  </state-transition-element>

  <state-transition-element id="__32__" symbol-set="[">
    <activate-on-match element="__33__"/>
  </state-transition-element>

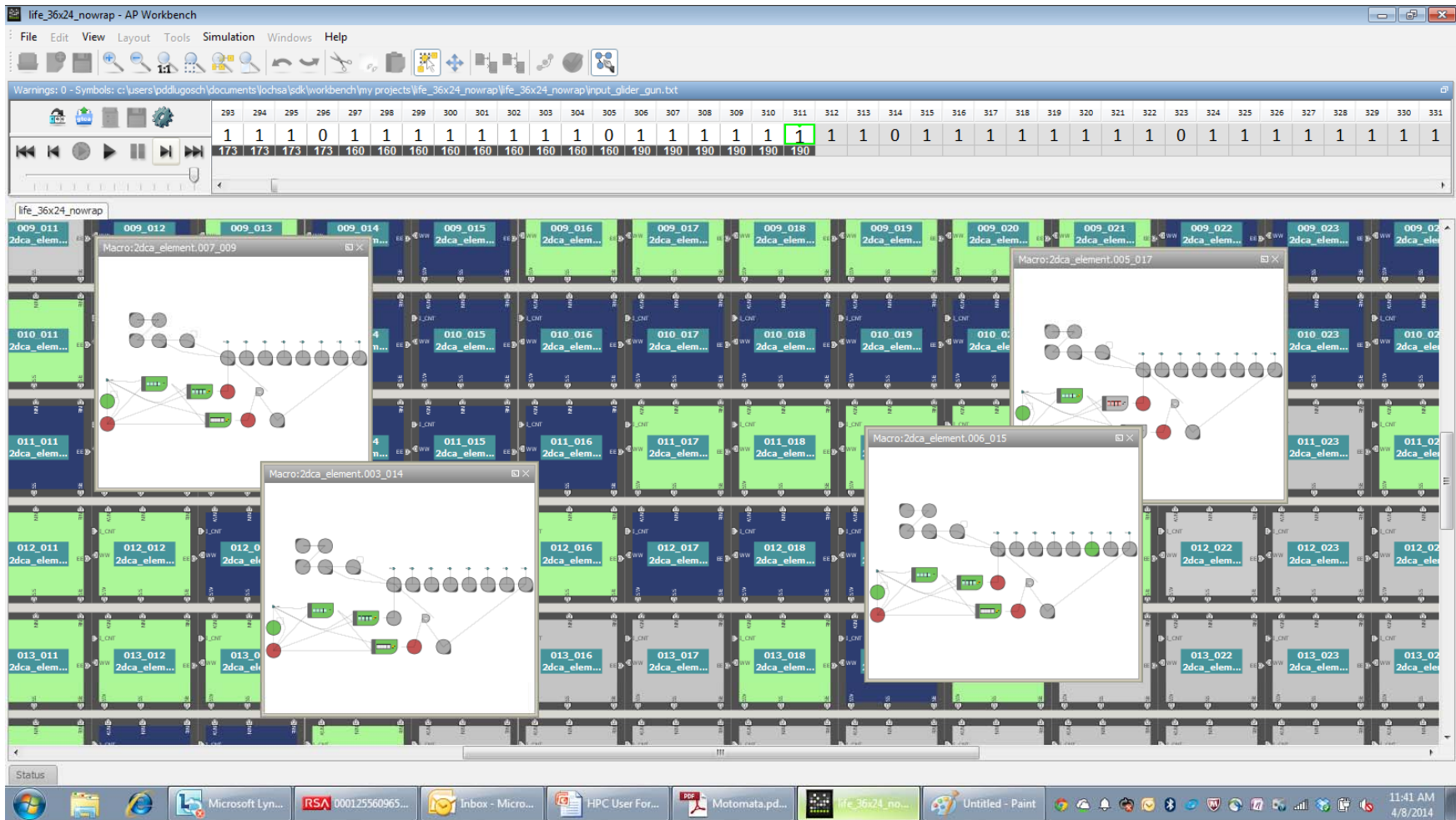
  <state-transition-element id="__46__" symbol-set="[\s]">
    <report-on-match/>
    <activate-on-match element="__46__"/>
  </state-transition-element>
```


Automata Processor Workbench



Conversion between
schematic automata and
Micron-developed ANML
description language

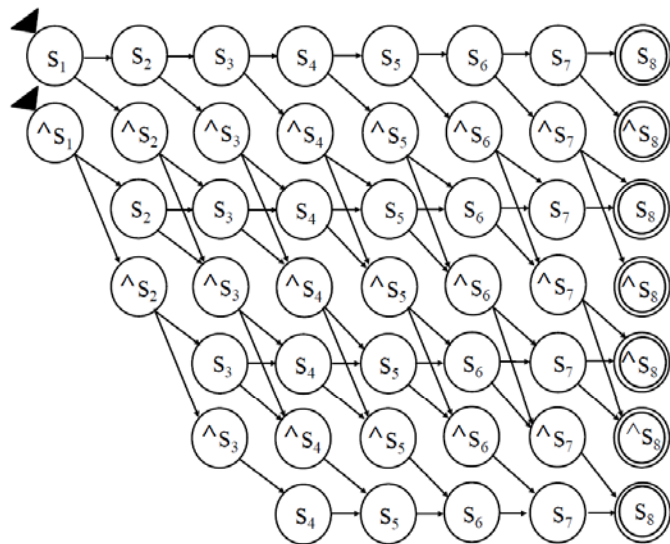
Automata Processor Workbench



GUI, Nested Macro Support, Design Simulation Support

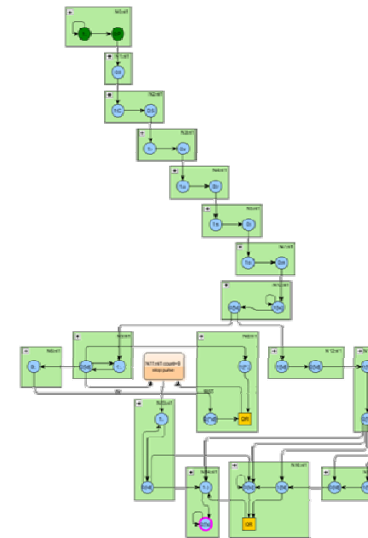
Design Methodologies: ANML, REGEX, Scripting, Full Custom

Application: Bioinformatics
Function: Fuzzy Matcher



Compiler Input: Scripted ANML

Application: Cyber Security
Function: DoS Attack Apache



Compiler Input: REGEX

Automata Processor SDK supports a variety of input methods

Converting Regular Expressions into Automata

Given the regular expression below:
/PICS-version\s+(\d{5,8}|\d(\x2e\d){10,})\s*\x29\s+/

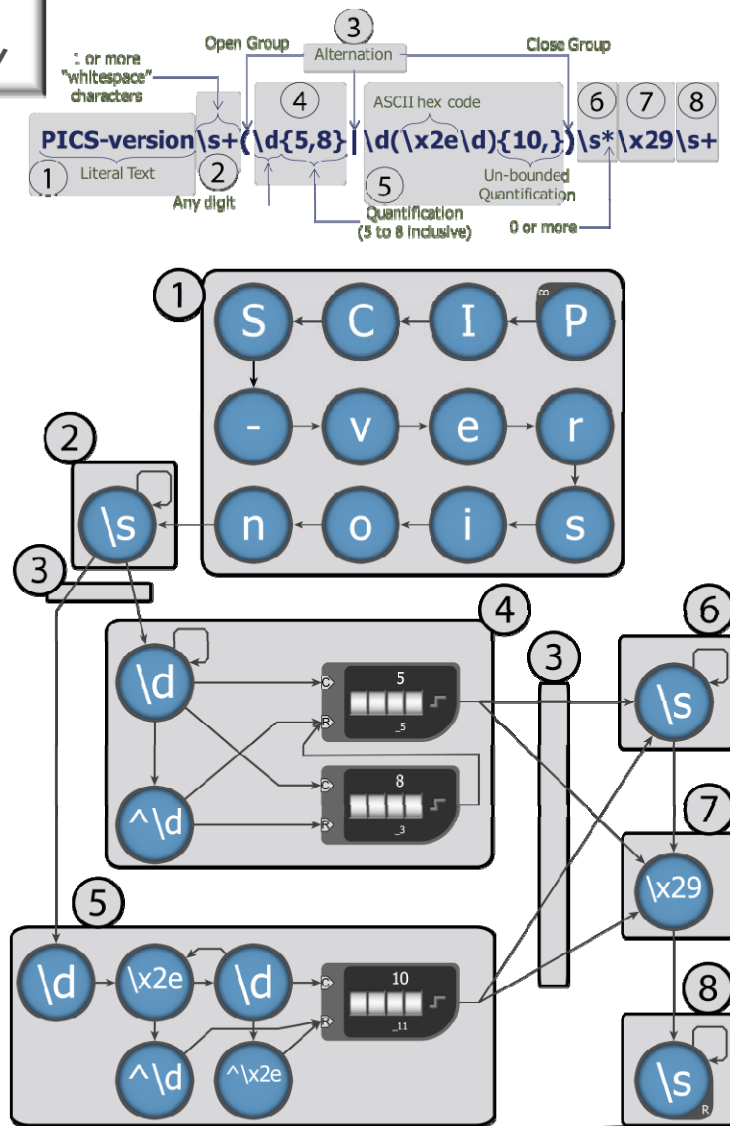
The compiler converts the RegEx into an NFA automata and does optimization and place & route

Compiled automata are programmed into the Automata Processor

Once programmed, the Automata Processor performs massively parallel processing in hardware

Chip Resource Utilization

STEs	23	(0.0468% of one chip)
CNTRs	3	(0.3906% of one chip)
BOOLs	0	(0% of one chip)
ESMs	1	(0.0163% of one chip)



Runtime I/O

Input: Stream of 8-bit symbols

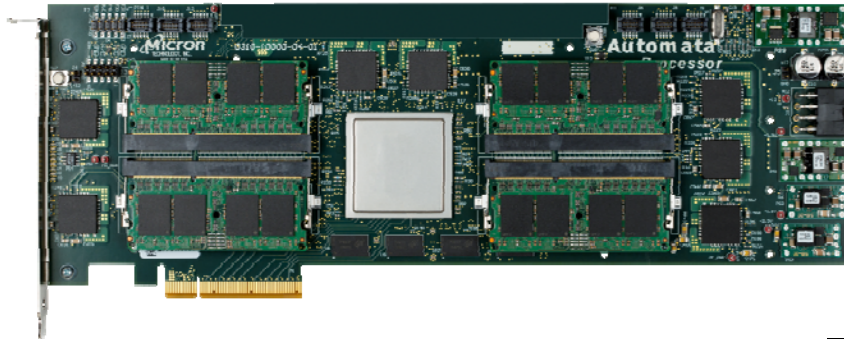
g t a c c c a g g t c a a g t g a c c
33 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651 1652



Event Vector	Node 1 Input symbol 4
Event Vector	Node 9 Input symbol 7
Event Vector	Node 51 Input symbol 10
Event Vector	Node 75 Input symbol 10
Event Vector	Node 75 Input symbol 11
Event Vector	Node 96 Input symbol 12
Event Vector	Node 7 Input symbol 15
Event Vector	Node 385 Input symbol 28
Event Vector	Node 89 Input symbol 33

Output: List of event vectors

Automata Processor: Support and Tools

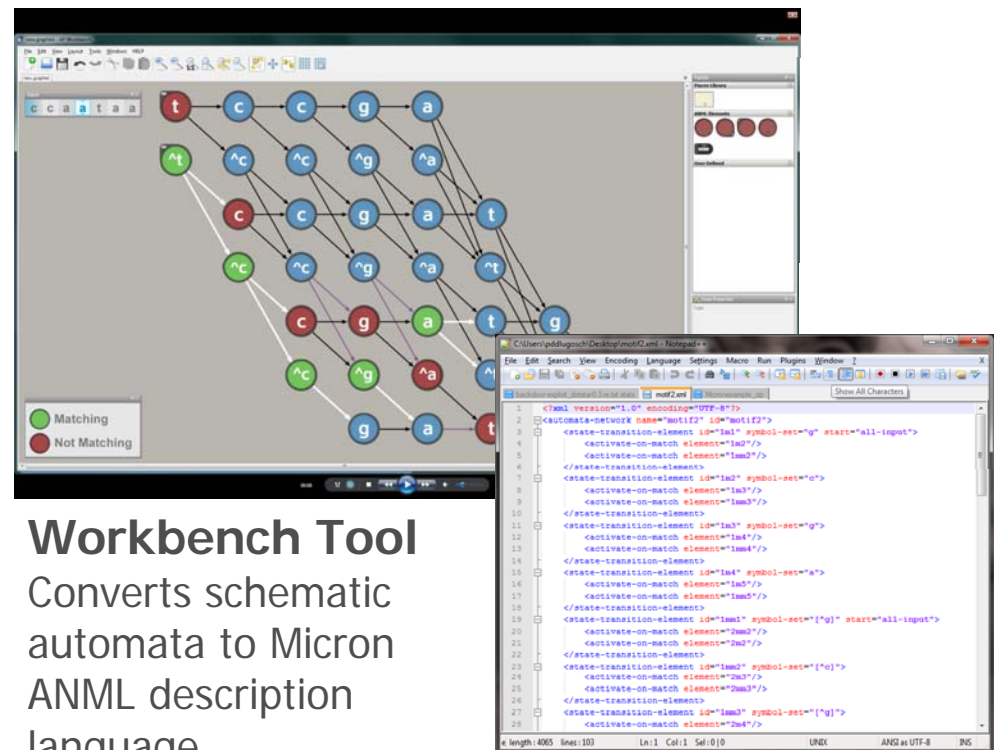


PCIe Development Board

- Industry-standard PCIe bus interface
- Capacity for up to 48 APs
- Large FPGA capacity
- DDR3 for local storage

Software Development Kit

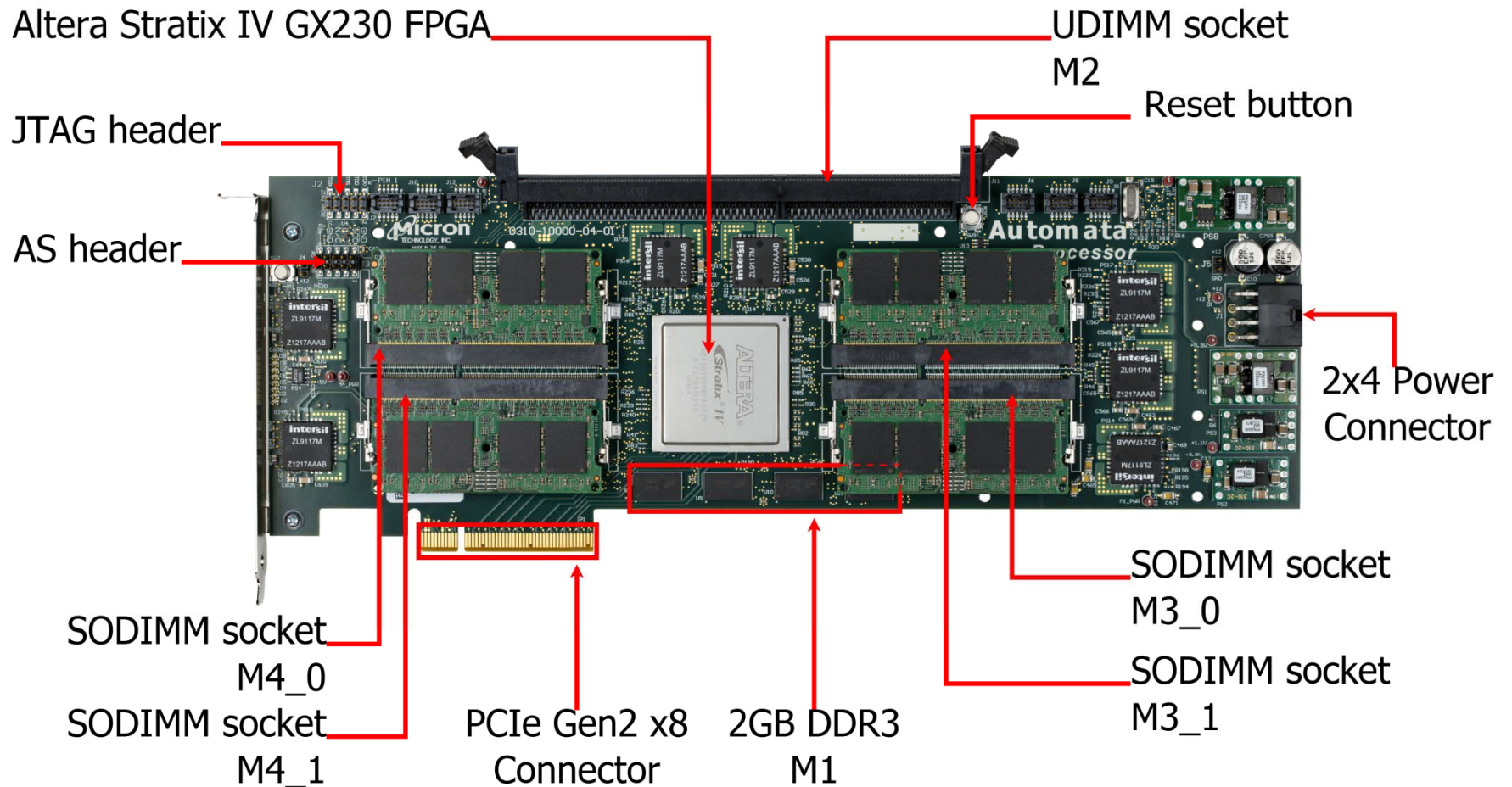
AP optimization, loading, and debugging tools and compiler



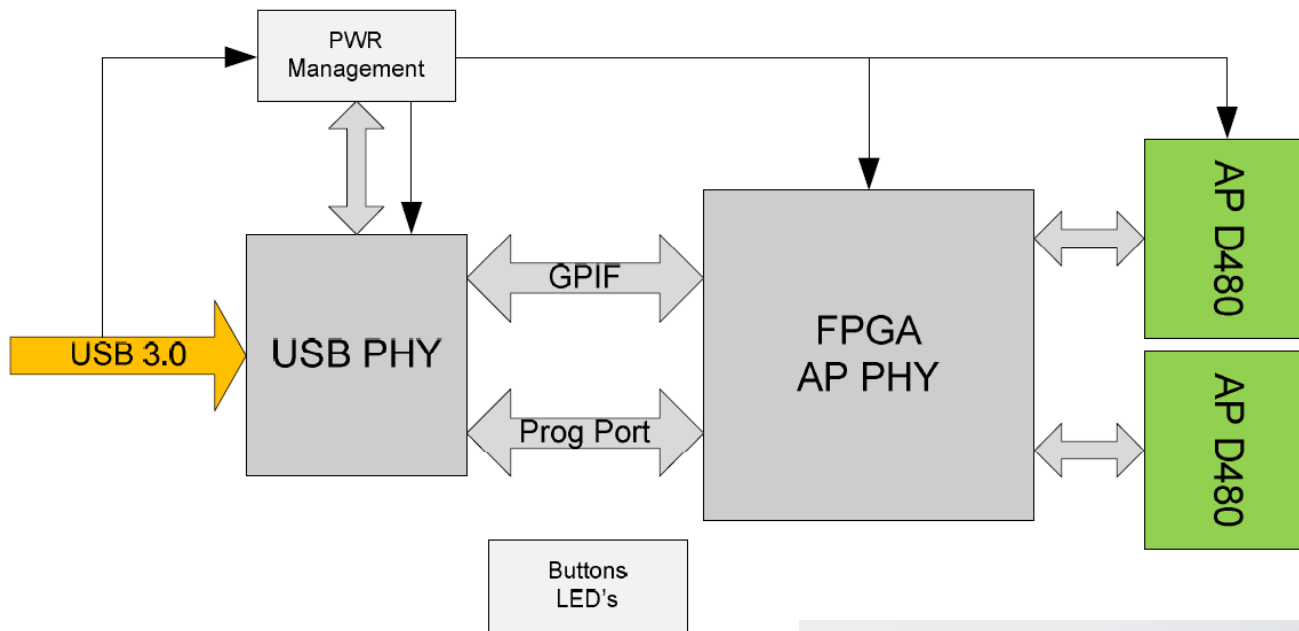
Workbench Tool

Converts schematic automata to Micron ANML description language

PCIe Dev Board

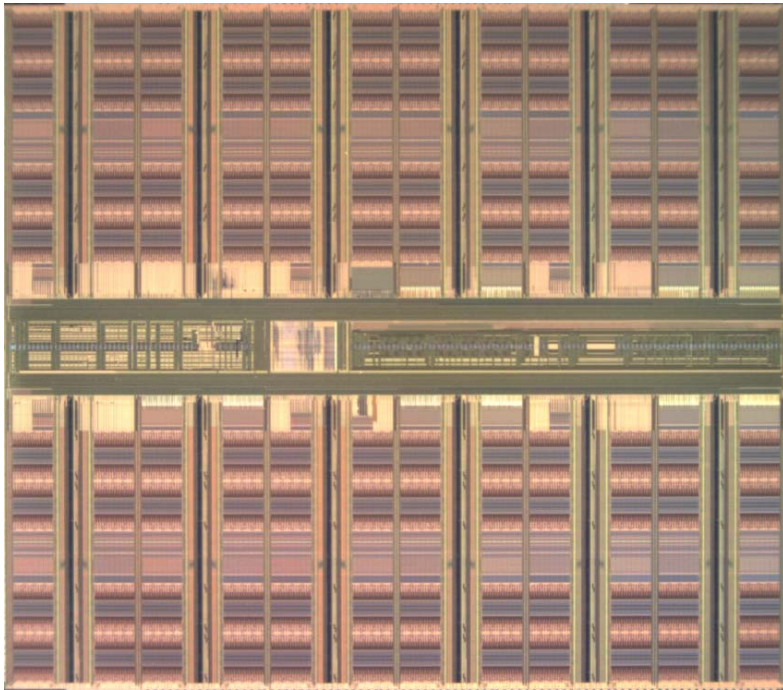


USB Dev Board



First Silicon Implementation of Micron's Automata Processor

The Micron D480 Automata Processor Die



Sample Availability: 4Q14
Production Availability: 2Q15

Device Elements

- State Transition Elements(STEs): 48K (49,152)
- Counter Elements (CNTRs): 768
- Boolean Elements (BOOLs): 2,304
- Automata Routing Matrix

Performance Characteristics

- 134M Symbols/Second (1.0 Gb/s)
- Max Automata Size: 24,576 STEs
- Max Event (Match) Capacity: 6,144 bits
 - 6 Independent Event Regions
 - Event Vector Division (2, 4, 8, 16)
- 512 Entry On-Chip State Vector Cache

Interface and Package

- DDR3 Bus Interface
- x8 Data Bus Width
- 144-ball FPBGA (15.4mm x 12.0mm)
- Total Power (MAX): ~4W

Automata Processor Summary

Micron's Automata Processor is a revolutionary architecture

Programmable and re-programmable to fit the applications

Offers major performance gains for parallel applications

Complete development environments

- SDK general availability: Q3, 2014
- PCIe development board available: Q4, 2014
- USB development board available: Q2, 2015

Engineering samples available: Q4, 2014

Production ramp: 2Q15

