Automaton Processor

Dennis Struhs

Arbeitsbereich Wissenschaftliches Rechnen
Fachbereich Informatik
Fakultät für Mathematik, Informatik und Naturwissenschaften
Universität Hamburg

17.11.2015

Topics Overview

Introduction

- Why would we need Automaton Processors?
  - Couldn't a CPU do the same job?
  - Maybe a GPU would be a good choice as well?
- Why would anybody want to even do research in this field?
  - Isn't the Von-Neumann architecture the final word of wisdom?
  - Shouldn't traditional systems be able to handle every problem there is with ease?

Traditional Computing Appliances

- Programs are usually working on rather small datasets
  - Data is most likely structured in a certain way / preprocessed
- GPU assists the CPU in specialized tasks
  - Large scale of applied matrix multiplikation
  - Billions of transistors solely for vector and matrix calculations
- Traditional Parallel Computing on von Neumann Systems
  - Use many CPUs and parallelized programs to raise performance

New Computing Challanges: Data Mining

"The Challenge of Complex, Unstructured Data"

- How to deal with large scale data input?
- How about real-time data analysis and evaluation?
- What's with unstructured data?
- How can this be performantly done using parallelisation?

Example of Unstructed Data

Figure: DNA Sequence

Structuring the Data

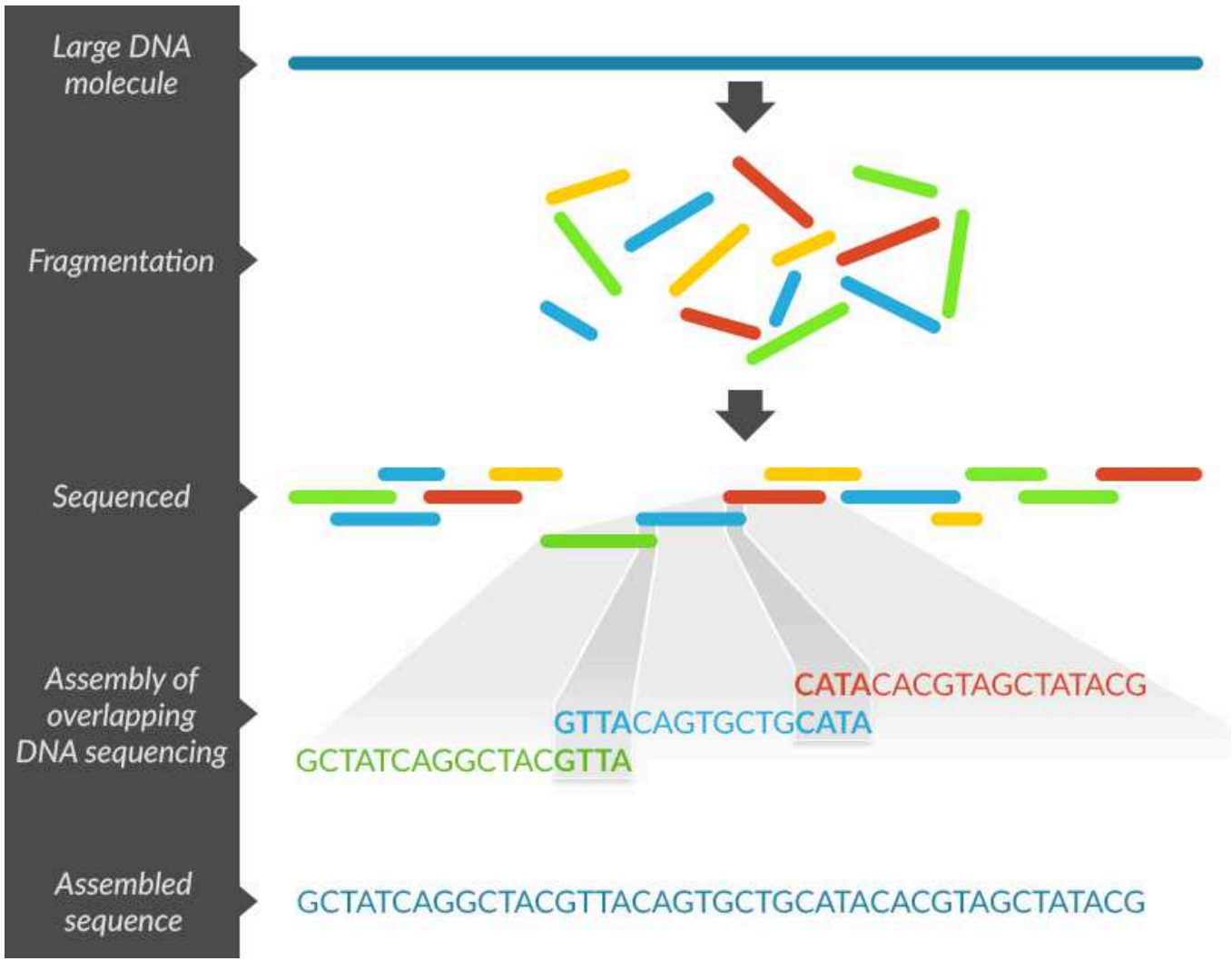
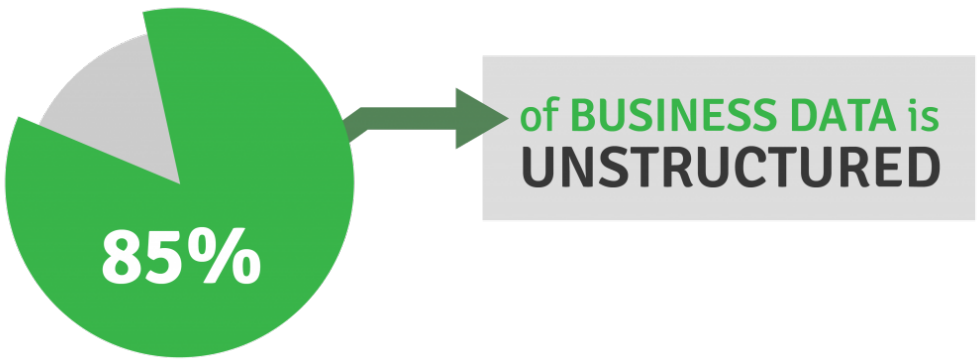


Figure: Genetic Sequencing Workflow

Large Scale Pattern Matching is Required

- Many scientific appliances for handling big data
- Economic interested in big data is growing steadily
- Quick way of handling this amount of data is required
- Data can deprecate over time losing its significance



The Automata Processor - A Parallel Computing Solution

- A memory based, massively parallelisable silicon device which is able to process data streams in real time

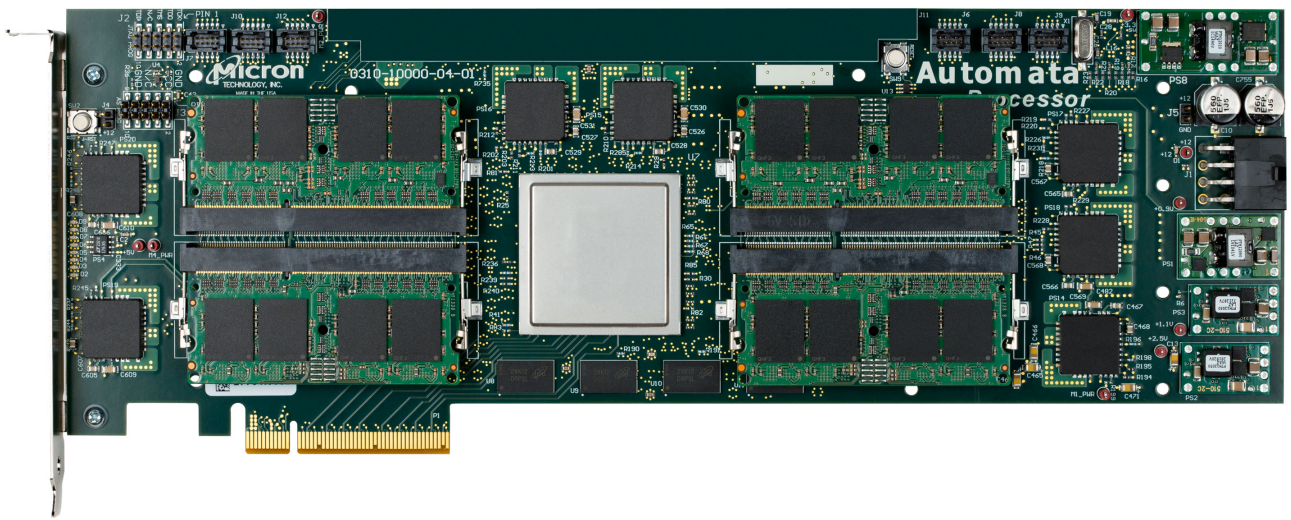


Figure: Micron Automata Processor

The Automaton Processor - A Parallel Computing Solution

- How does it integrate into commonly used Systems?
  - Needs an existing system that operates it
  - Is (so far) not a standalone Processing Unit
- It is meant to be used along a traditional Computing Setup
  - Provides a non-Von-Neuman processing architecture
- Pursues the implementation of a NFA based design

Von Neumann Architecture

- Realises all components of a deterministic Turing Machine
  - A concept for implementation of universal machines.
- A Turing Machine consist of an input band and an Automaton
- The Automata operates the data stored on the input band
  - It has a writer/reader that touches one item of the input band
  - The Automaton controls the movement and the manipulation of the cells content

Example of a Turing Machine

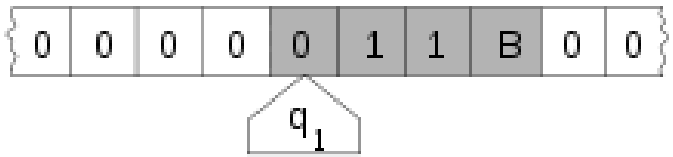


Figure: Input Band with Read/Write Head

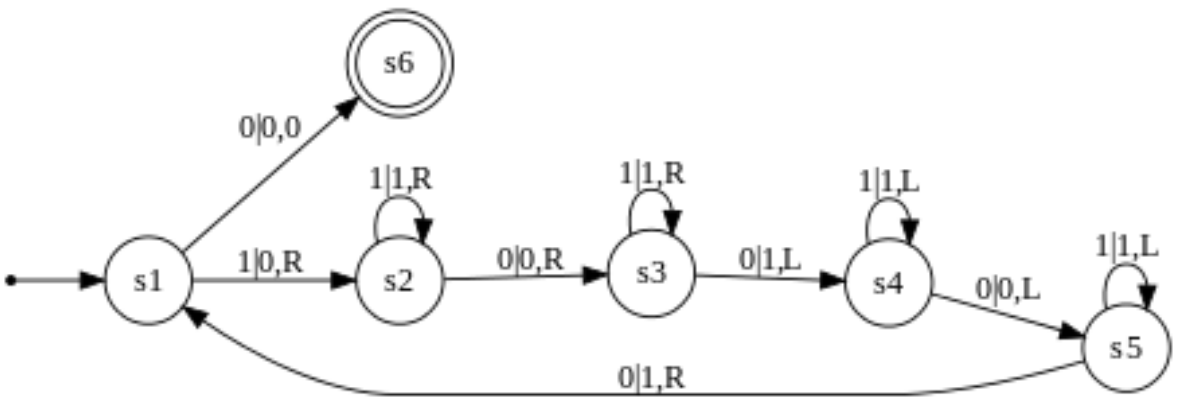
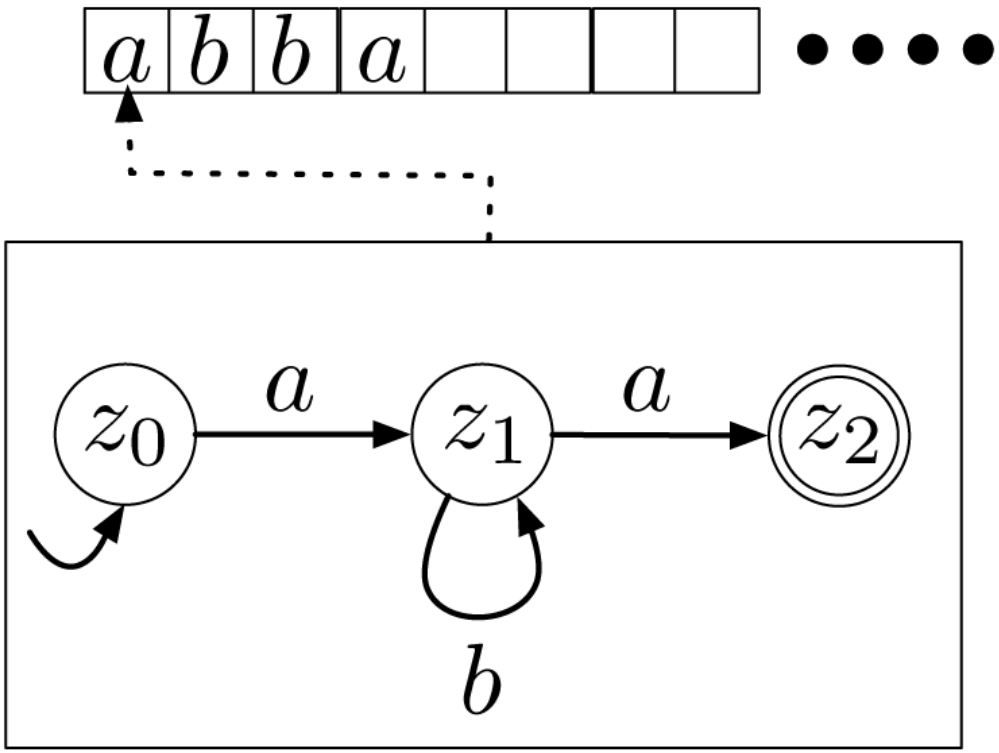


Figure: Visualized Transition Function

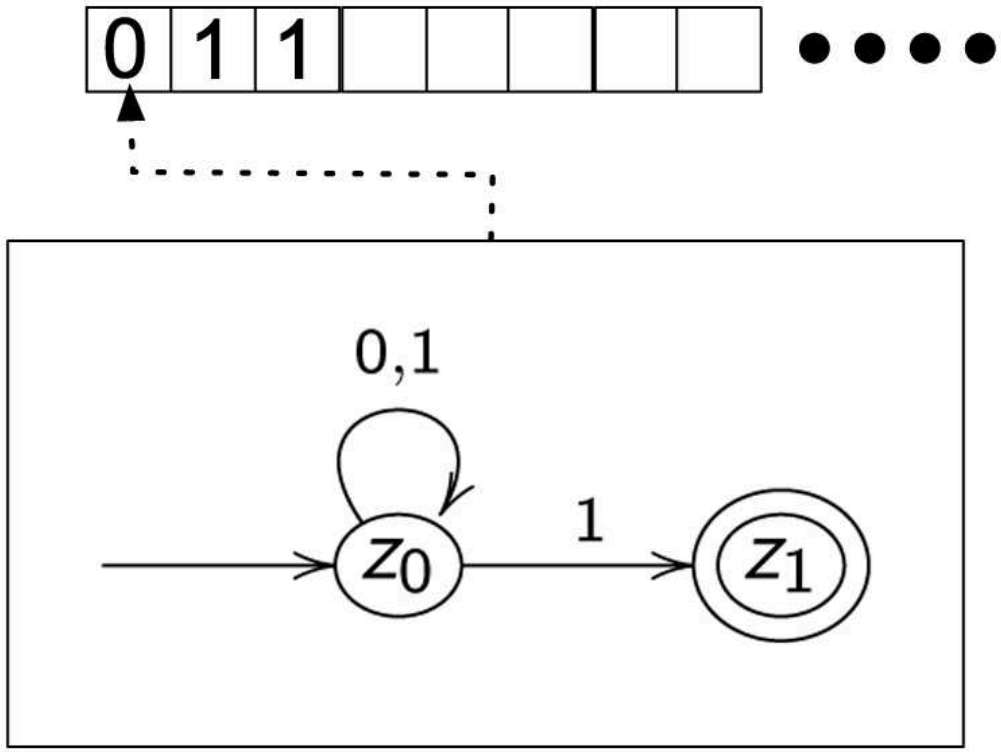
NFA versus DFA

- **D**eterministic **F**inite **A**utomaton



NFA versus DFA

- **N**ondeterministic **F**inite **A**utomaton



All possible paths on the Word 011:

- $(z_0, 011) \vdash (z_0, 11) \vdash (z_1, 1)$ – blocked
- $(z_0, 011) \vdash (z_0, 11) \vdash (z_0, 1) \vdash (z_0, \lambda)$ – rejected
- $(z_0, 011) \vdash (z_0, 11) \vdash (z_0, 1) \vdash (z_1, \lambda)$ – accepted!

As long as there is at least one Path that succeeds, the Automaton will accept the input word.

The Automata Processor: Technical Details

- How does the Automata Processor implement the NFA?
  - It achieves the NFA implementation by using many DFAs runned simultanously to emulate an NFA
  - Each DFA-Cell will complete Symbol-Delivery, Comparison and Activation in a single clock cycle every time
  - There are no race conditions or timing loops to take care of since the DFA-Cells are not dependant on each other

A Memory-Derived Architecture

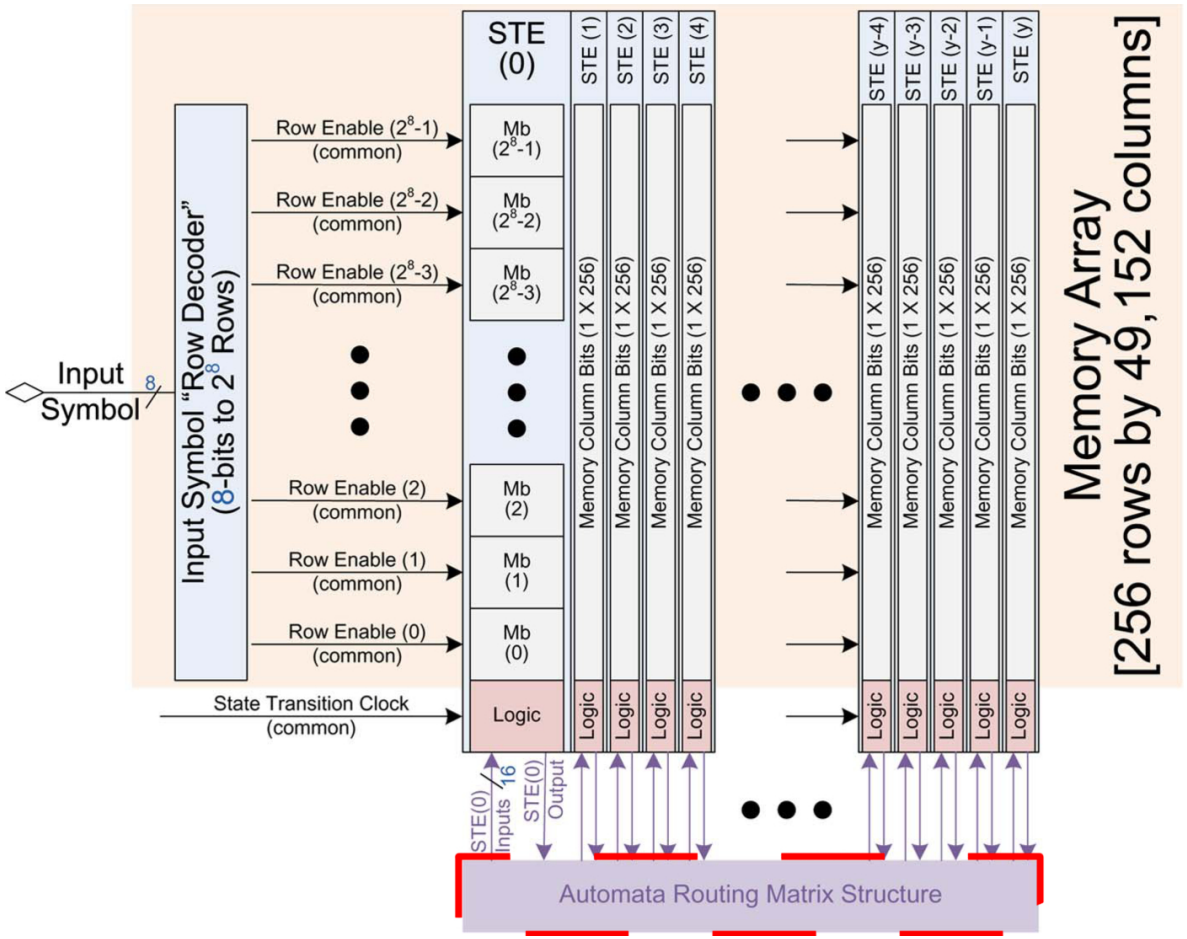


Figure: The Memory Array with 8-Bit DDR3 Bus interface

A Memory-Derived Architecture

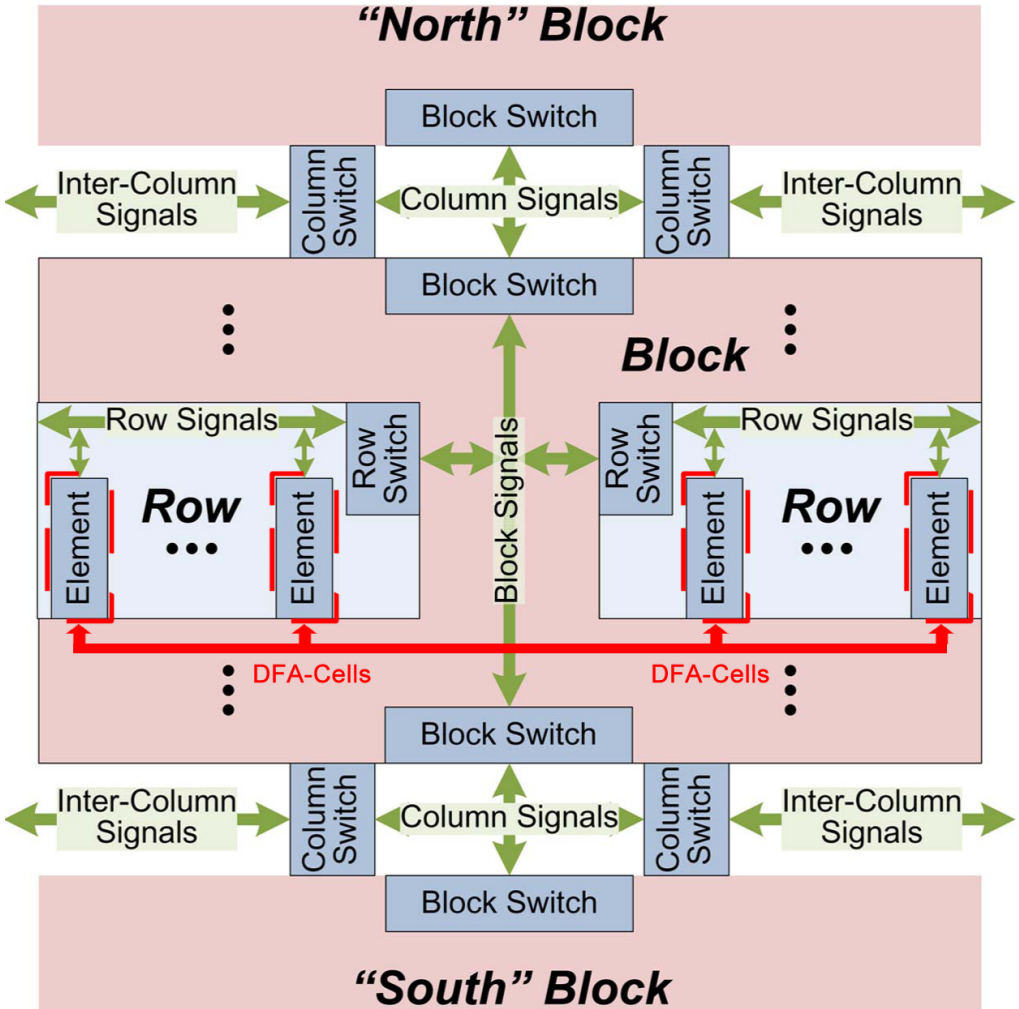


Figure: The Memory Routing Matrix with DFA-Cells Highlighted

The DFA-Cells Location

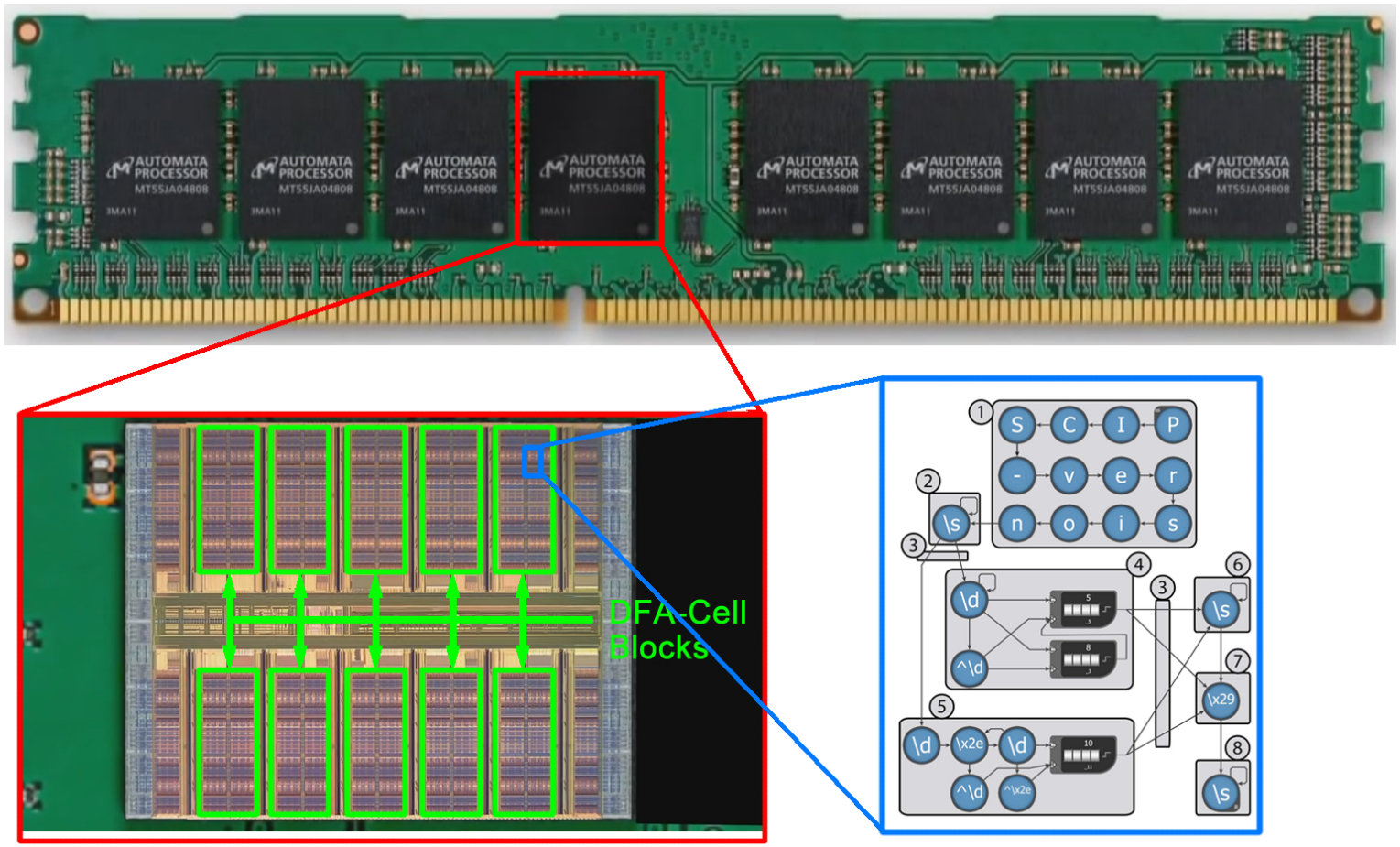


Figure: The Automata Processor Memory Modules

The DFA-Cells Explained

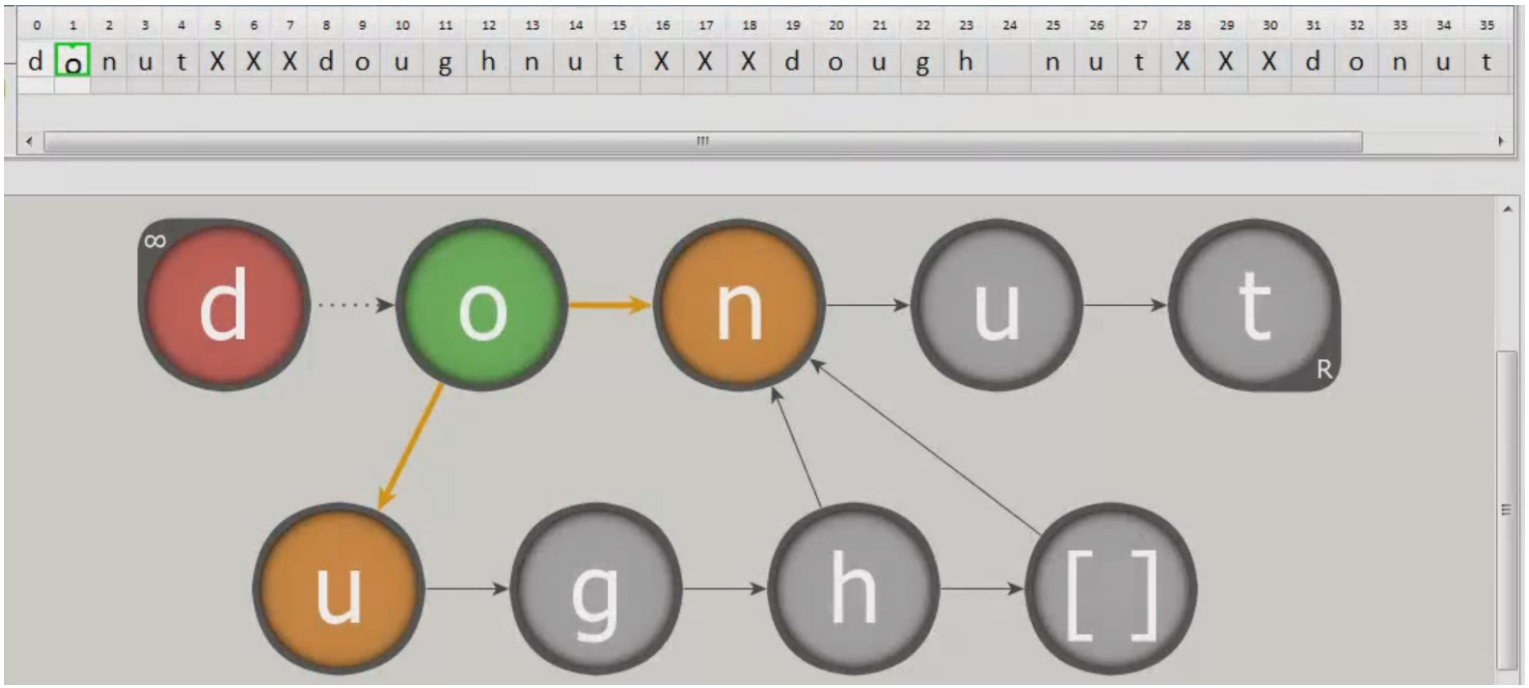


Figure: A simple example of a programmable pattern for a DFA-Cell

The Self Development Kit Features

- Features a fully interactive GUI-Based application
  - Drag & Drop your Elements onto your workspace
  - Connect your elements intuitiviely for full functionality
- Has debugging mechanisms to find deadlocks
- Program each DFA-Cell with a different Pattern
- Realtime-Check if the Automaton matches the desired pattern

A quick look on the SDK

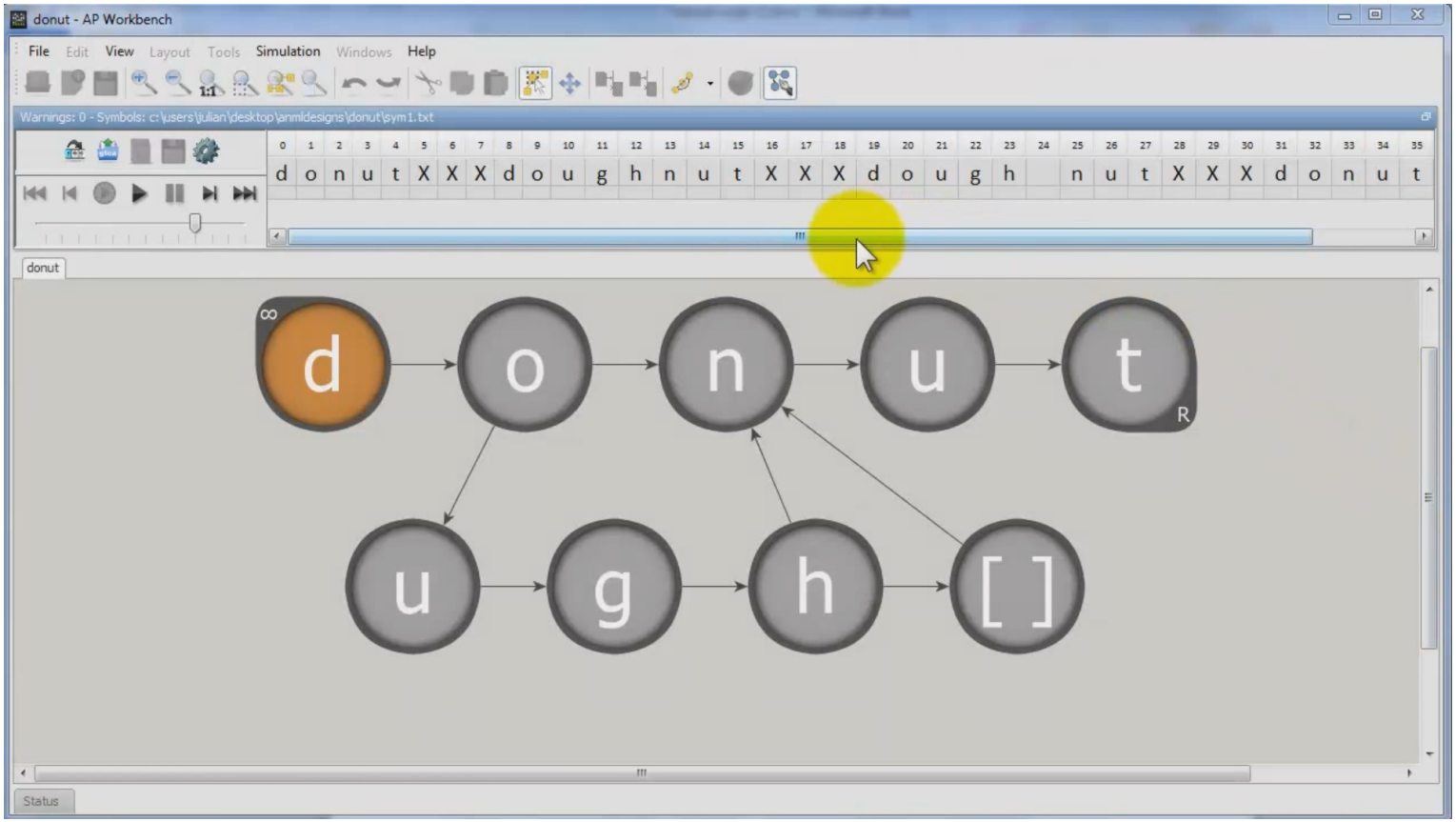


Figure: The Automata Processor SDKI

Zusammenfassung

- Traditional Approach
  - Not well suited for "big data"
  - Potentially underperforming even when parellelised
- Automaton Processor
  - Ideal for preprocessing big data in real time
  - Many computation elements allow for scalable parallelisation

Literature


Désiré Athow.
All you need to know about Automata, Micron's revolutionary processor.
Techradar.pro, (1):1, 3 2014.


Russell Fish.
Two Views of the Post PC World - Automata Processor and TOMI Celeste, Part 1-4.
EDN Network, (1):1–4, 3 2014.


Micron.
A Massively Parallel Computing Solution.
http://www.micron.com/about/innovations/automata-processing, (1):1, 11
2015.