

IoT Devices Authentication Using In-Memory Computation

MICRON AUTOMATA PRCESSOR

July 28 ,2017

High Performance Computing Architecture

Spring 2017

Student# 201692544

Abstract

Since IoT is an information technology buzzword today, and In-Memory Computation is a highly parallel processing architecture of comprehensive search and analysis. There is no standard architecture for IoT, exploring the architectural aspects of authentication of things connected to others via the Internet and small networks. This process is compulsory before joining the internet either it is Web of Things, Machine to Machine, or Human to Machine frame of communication. For connected moveable devices or objects when they switch from one network to another, their connectivity to the network must be ensured, and on the other side when intrusion encountered from outside the IoT environment it must be handled by the security policy implemented on servicing computing machine. This search is as simple to implement with hash tables, but hashing becomes inefficient when entries become more significant and not scalable. Devices profiles and firewalls for authentication and intrusion detection/prevention must be maintained after they are authenticated. When thousands of devices need to get connected with IoT environment then managing their authentication and intrusion blocking policies must be performed quickly and giving an impression of dedicated service response from the server machine. These high-performance servers are implemented with MIMD machines. Accelerating service of requests from IoT's low-end devices using In-Memory Computation is will be a new approach for comprehensive search and analysis. The technique of Automata Processing used for search of device identity speed up the processes of authentication if implemented on Micron Automata Processor, an In-Memory Computing Architecture. In this report with amalgamation of IoT and Non-Von Neuman Architecture with a new approach of authentication with their background and viability study.

Organization of Report

This report is organized into five chapters. The first chapter begins with introduction on IoT and In-Memory Computation with Micron Automata and how In-Memory Computation is deriving the IoT and end up with motivating problem statement. Chapter 2 contains literature review, and a motivating example of similar to our problem small scale example is discussed. Micron Automata is a new research and development under Micron. Its architecture is also explored in literature review chapter. Chapter 3 is devoted to theoretical findings on Similarity searches algorithms, and locality Hashing rest of the chapter is dedicated to IoT Hardware architecture, and processors range for IoT hardware with application devices of IoT. In Chapter 4 Micron Automata Programming paradigm and problem input format discussed with its internal building blocks and programming similarity with those building blocks are more clearly discussed. Chapter 5 gives methodology and quantitate analysis of authentication problems modeled as NFA on Micron Automata. The conclusion contains the achievement and shortcomings of this work. References sections contain complete sources strongly appreciated in this work.

List of Figures

Figure 1: Simplified Reference Architecture of IoT	13
Figure-2 Historical Cost of Computer Memory and Storage [4].	14
Figure 3: AP Chip Containing Numerous STEs, Programable Counters and Boolean Cells.....	17
Figure 4: Configuration of Interconnection among STEs and Counter with one Boolean Cell.....	17
Figure 5 : A concrete Implementation of Network Devices on Web of Things Interface	19
Figure 6: IoT Model for Research and Development of SSME.....	22
Figure 7 IoT Reference Architecture (Courtesy of FREMANTLE, 2015).....	25
Figure 8: Micron Automata Design (Micron).....	27
Figure 9 : Complete System Architecture of Micron AP for kNN Search Algorithm.	31
Figure 10 : Performance of AP with kNN variants	32
Figure 11: Energy Efficiency Comparison	32
Figure 12 : Building Blocks of IoT devices	34
Figure 13: Performance Comparison of MIPS M-Class Processor.....	36
Figure 14 : Complete SoC Solution of M Class IoT Processor and, Cloud Connectivity using Flow Cloud.....	37
Figure 15 : Finite Automata Types and Categories	38
Figure 16: Example of Protein Automata Mapping on AP	39
Figure 17: NFAs for Two Strings Matching "HELLO" and "WORLD"	42
Figure 18 : NFA for MAC Address Matching.....	42
Figure 19: NFA for MAC Address Matching eliminating the string separator	42

List of Tables

Table 1: World Internet Users with Populations States of 2017, Copyright © 2017, Miniwatts Marketing Group.....	12
Table 2: Brief History of Finite Automata Processor/Center for Automata Processing.....	15
Table 3: Micron Automata Processor Board Specifications	29
Table 4 : Execution Time of Finding Protein Patterns	33
Table 5: NFA Implementation computation and Storage Cost	41

Abbreviations

IoT	Internet of Things
CAP	Center of Automata Processing
UVI	University of Virginia
AP	Automata Processor
FA	Finite Automata
NFA	Nondeterministic Finite Automata
DRAM	Dynamic Random-Access Memory
ACM	Association of Computing Machinery
IEEE	Institute of Electrical and Electronic Engineering
SDK	Software Development Kit
STE	State Transition Element
ML	Machine Learning
k-NN	k-Nearest Neighbors
CAM	Content Addressable Memory
NLP	Natural Language Processing
IP	Internetworking Protocol
RFID	Radio Frequency Identification
EPC	Electronic Product Code
EPCDS	Electronic Product Code Discovery Service
EPCIS	Electronic Product Code Information Service
EPCSS	Electronic Product Code Security Service
ONS	Object Naming Service
Wi-fi	Wireless Fidelity
NFC	Near Field Communication
WSN	Wireless Sensor Network
AI	Artificial Intelligence
CE	Consumer Electronics
HTTP	Hyper Text Transfer
CoAP	Constrained Application Protocol

MQTT	Message Queue Telemetry Transport
QR Code	Quick Response Code
PRC	Placement and Routing Compilation
PIM	Processing In-Memory
LSH	Locality Sensitive Hashing
DAC	Data Acquisition Module

Notations

~	Approximation
x	Multiple
e	Exponent
#	Number Sign
-	String Separator

Table of Contents

Abstract	2
Organization of Report	3
List of Figures	4
List of Tables	5
Abbreviations	6
Notations	8
Table of Contents.....	9
1. Introduction.....	11
1.1 Internet of Things.....	11
1.2 In Memory Computing Deriving IoT.....	13
1.3 What is Micron Automata Processor	15
1.4 Problem Statement	17
2. Literature Review	19
2.1 Related Work	19
2.2 IoT Literature Review	21
2.3 IoT Reference Architecture	24
2.4 Micron Automata Processor Architecture	27
3. Theoretical Findings	30
3.1 Similarity Search	30
3.2 Applications on AP.....	30
3.3 IoT Hardware Architecture	33
3.4 IoT Processors	34
3.5 Reasons of MIPS as Ideal Processors for IoT	35
3.5 Examples of IoT Connected Devices	37
4. Building Blocks of NFA in Micron Auomata	38
4.1 NFA Construction	38
4.2 State Transition Elements	38
4.3 Counter Elements.....	38
4.4 Boolean Elements	39
4.5 Input Output Elements.....	39

4.6 Automata Network Markup Language.....	39
5. Methodology and Quantitative Analysis.....	41
5.1 Objective	41
5.2 Device Identifiers.....	41
5.3 Programming Challenge	41
5.4 Constructing NFA.....	42
5.5 Performance Analysis	44
5.6 Proposed NFA Design	44
Conclusion	48
References	49

Introduction

1.1 Internet of Things

The term Internet of Things(IoT) is a description of embedded devices connected through the internet interacting with each other exchanging information, and making decisions. Everything from your glasses to your cars and your home appliances to your office workplace sharing the information to make effects on others. The size of devices is different, but their communication is based on the relationship of their connectivity. Some devices have a bunch of resources whereas some are equipped with limited resources. So, implementing the same communication protocol which needs a standard set of resources is challenging.

Kevin Ashton proposed the term Internet of Things in 1999 [1] while he was presenting a new idea on RFID at Procter & Gamble. At that time, information was generated by only computers operated by humans that he supposed to say now things will generate information. This is because of the instant of the growth rate of internet users. As per statistics are given in Table-1 [2]. Although the Internet is overgrowing due to technological advancements, newer networking trends there is also a gap of internet users and populations of continents, so this gap is an opportunity to fill with.

IoT means that all talk of physical objects with each other. Some technologies which are driving the Internet of Things including, Nano Technology, RFID, Smart Grids, Computer network these all concepts are influenced by ubiquitous communication and computer networks.

Now, we are living in an era of ubiquity where humans are generating less information in comparison to machines and devices leading toward the Internet of Things. There are ~2000 articles about the Internet of things on IEEE Xplore and more than 1,004,926 Books, Articles, and surveys on ACM digital library specifically on Internet of Things (Accessed on 02 June 2017). Companies depending on IoT infrastructure are concerned and facing problems due to lack of skills their employees have, industry standards of IoT, lacking management knowledge and commitment.

Table 1: World Internet Users with Populations States of 2017, Copyright © 2017,
Miniwatts Marketing Group

WORLD INTERNET USAGE AND POPULATION STATISTICS						
MARCH 25, 2017 - Update						
World Regions	Population (2017 Est.)	Population % of World	Internet Users 31 Mar 2017	Penetration Rate (% Pop.)	Growth 2000- 2017	Users % Table
Africa	1,246,504,865	16.6 %	345,676,501	27.7 %	7,557.2%	9.3 %
Asia	4,148,177,672	55.2 %	1,873,856,654	45.2 %	1,539.4%	50.2 %
Europe	822,710,362	10.9 %	636,971,824	77.4 %	506.1%	17.1 %
Latin America / Caribbean	647,604,645	8.6 %	385,919,382	59.6 %	2,035.8%	10.3 %
Middle East	250,327,574	3.3 %	141,931,765	56.7 %	4,220.9%	3.8 %
North America	363,224,006	4.8 %	320,068,243	88.1 %	196.1%	8.6 %
Oceania / Australia	40,479,846	0.5 %	27,549,054	68.1 %	261.5%	0.7 %
WORLD TOTAL	7,519,028,970	100.0 %	3,731,973,423	49.6 %	933.8%	100.0 %

IoT will be fully realized when smaller networks of IoT will become larger, forming organizations and industries networks. IoT research includes various technologies in terms of hardware and software. Success if IoT is entirely dependent on the availability of internet whereas 50% of the world population does not have access to the internet, and there must be a customer interest before private companies invest in internet infrastructure. Internet availability is not only the issue for implementation of IoT, but as we keep exploring the area numerous challenges are ahead in which availability of Internet is although first but security, cheap sensor development, Energy consumption, scalability, ability to compute, fault tolerance and acceptability among the society. Scalability and security issues we are going to address in this project and further explore an exciting opportunity to leverage the infrastructure of servicing to IoT devices with less response time and energy efficiency because higher energy is consumed in communication. Big data and privacy issues are fundamentally concerned with IoT's future. There are other exciting opportunities available but beyond the scope of this project. However, these issues or not to be dealt with local development rather than a global phenomenon.

1.2 In Memory Computing Deriving IoT

In-memory computing is the storage of information in main memory rather than relational databases operating on disks. In-memory computing is prominent in

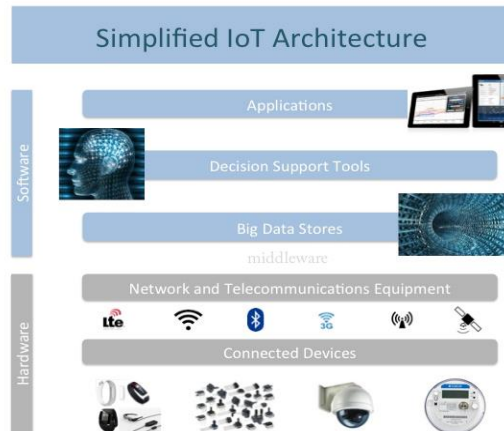


Figure 1: Simplified Reference Architecture of IoT

determining patterns quickly massively analyzing the bunch of data. Memory prices are being dropped which is a significant attraction towards the In-memory Computing which made the systems economical and efficient in terms of their specific application applicability [3]. Numerous companies used In-Memory Computing for analyzing data in seconds rather than spending long hours. Some reasons are below,

- Caching the countless data constantly
- Fast response times for big searches

Complex processing structures SAP is a German enterprise software group developed an In-Memory Computing technology called High-Speed Analytical Appliance, which is used as a sophisticated machine for implementing compression techniques. Storing information in RAM got 10,000 times faster than disks, which analyzed data in seconds rather than spending hours. Why we are discussing the In-Memory computing platform developed by SAP is showing the significance of implementing it, but how it is advantageous in IoT advancement and how it can be used for IoT challenges is a vibrant

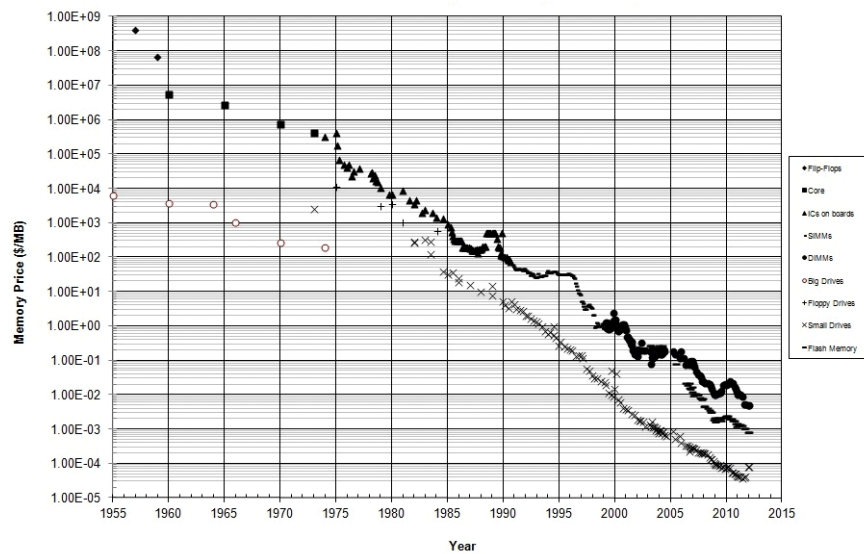


Figure-2 Historical Cost of Computer Memory and Storage [4].

area to be explored. Here we see In-Memory Computation in the context of IoT narrowing down the scope, and we will emphasize the architectural amendment in simple IoT reference architecture with In-Memory Computing solution, which will be illustrated in later sections of this project report. Since the price of DRAM is decreasing proportionally and this significant decrease in memory price is so become an exciting domain to use In-Memory Computing. Here is an interesting analysis of Jason Stamper, Analyst, 451 Research explored the significance of IoT with advancement how can it save the billions of \$ in terms of money and time. GE report says if machines will become autonomous taking decision just efficiency in that system of 1% could save 15 years of expenditures listed below,

- Jet Plane Fuel for 30 Billion Dollars in the Airline Industry
- 60 ~ Billion Dollars spending on health care will be eliminated with intelligent equipment at hospitals and optimized ways of treatments
- Fuel consumption 66 Billion Dollars for global gas-fired power plant fleet

1.3 What is Micron Automata Processor

Micron Automata is a reconfigurable processing architecture developed by Micron in 2013 and hereafter available in CAP (Center for Automata Processing, University of Virginia U.S) in 2016. Historical background of AP is given in Table 2 you can see there is a large gap between State Charts and AP innovation this is because in that era of improvement work was more concerned towards the efficiency, throughput and parallelism in MIMD machines and performance boost for Kernel, Synthetic and Peak performance benchmarks approach, reconfigurable computing also got more attention. AP is a first Non-Von-Neuman architecture in the 21st century. It was developed to address computations on graphs pattern matching and big data analysis. Some of the profound applications on Micron Automata are

Table 2: Brief History of Finite Automata Processor/Center for Automata Processing

Invention	Credits	Year
Turing Machine	Alan Turing	1936
Finite Automata	Warren McCulloch &	1943
Cellular Automata	John Von Neuman &	1948
Finite State Machine	George Mealy/Edward	1955/1956
Regular expression & finite	Stephen Kleene	1956
Nondeterministic finite	Michael Rabin & Dana	1959
Conway game of life	John Conway	1970
Statecharts	David Harel	1987
Micron Introduction to AP	Micron.Inc	2013
Center for Automata	UVA Established	2013
AP Hardware	CAP	2016

Bioinformatics, Network Security, Finance, Big Data, Data Mining, Natural Language Processing, High Energy Physics and Machine learning. Here we address symmetric application to network but security that is not addressed before. The detailed architecture of AP will be discussed in Chapter 2. One beauty of AP is that it is energy efficient. When we come to unstructured data processing Micron

Automata is the best choice because the inputs of the micron automata processor is an unstructured data stream, and this unstructured data processing is typical in big data problems. AP is a highly parallel design for pattern matching, and we will consider that pattern matching feature of AP is detailed in this project.

1.3.1 How Automata Different

Since complex problems for Von Neuman architecture must be decomposed into a smaller set of statements to be implemented on Hardware also, these statements come up with dependencies and data movement, which limit the problem solving of these applications. The speed of problem-solving system will then dependent on system capability to movement of data as well. AP is addressing these issues inherently due to its non-Von Neuman Architecture.

Conventional memory chip design is hugely parallel, and one memory read operation is done on with multiple data accesses in Von Neuman Architecture is of limited width(serialization) and only a limited number of bits can be sent at a time step i.e one clock cycle, resulting the limit of processor data access rate and becoming the memory bottleneck called memory wall. Several enhancements into Non-von Neuman Architecture, like introducing the multilevel cache hierarchy and TLBs to implement short look for page tables. However, these techniques are now facing challenges on real user significant data processing benchmarks where millions of comparisons and search operations need to be done in one-time step.

1.3.2 How Automata Works

Micron Technology has recently introduced a new processing architecture, the Automata Processor (AP), that is a native hardware implementation of a classic computational model, nondeterministic finite automaton (NFA). NFAs are primarily symbolic pattern-matching machines, which on the one hand limits their generality. However, true NFAs are extraordinarily powerful at this task, because allowing an arbitrary number of states to be active at the same time allows massive parallelism. Automata consist of thousands of State Transitions Elements, and these elements are configured via various interconnections ways

to model a specific search problem. After configuring the interconnection of STEs design come up with a bunch of state machines. We also have Boolean cells and programmable counters. Micron Automata are designed on PCIe board and contain an FPGA fabric which is used as a communication interface between the host system and Automata Chips. Roughly board provides 1.5 million processing elements i.e. STEs, which together have the capability of doing more than 200 trillion match operations at a time. Figures 3 and 4 gave the inside view of Micron Automata one chip.

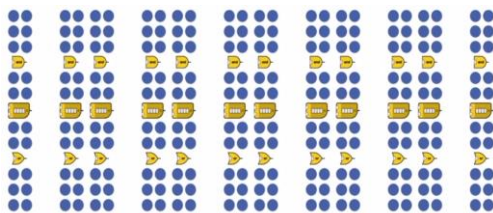


Figure 3: AP Chip Containing Numerous STEs, Programable Counters and Boolean Cells

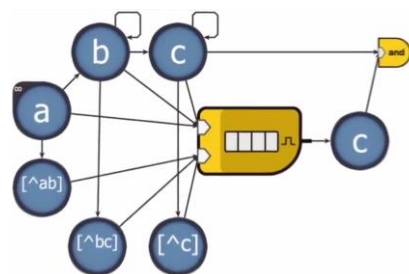


Figure 4: Configuration of Interconnection among STEs and Counter with one Boolean Cell

PCIe is used to support traditional CPU architecture, and it does not need any operating system. It is more like a GPU. The only challenge is to change the problem structure to NFA, and Micron automata can outperform. Finding a useful pattern from astronomical data is complicated for a traditional CPU. Typically loading the pattern and comparing it with all available candidates in data set.

Network Security that is a new implementation on AP was to check the specific threat profiles coming to network environment simultaneously and recognition of attack. We are going to use the same idea but in a more abstract way to put back hashing techniques for checking the IoT devices whether they can enter network system or not.

1.4 Problem Statement

As we collect more and more data about the world around us and digitize artifacts from the past data which is inherently required in every field of inquiry and analysis. The future success of industries is dependent on big data analysis success. However,

this big data is not only generated by humans. Now machines are significant sources of generating this information, and processing of big data is necessarily required for establishment of future business, research, and academia. This information generating devices architecturally came under the umbrella IoT. IoT challenges and significant parallel areas of health care, cybersecurity, big data, machine learning, etc. are today's interest in growing industry and world of technology. Security protocols for protecting information are becoming an exciting and challenging area for today's technology vendors and maintenance groups. Recognizing the "Things" on the internet is a challenging process for servers computing. With the advent of IPv6 3.40×10^{38} distinct machines will be addressable. However, addressing the machines will rise the challenge of authentication. IP is just Internetworking Protocol, which is used to route data traffic, but authentication of legitimate users ahead is a great challenge. Cyber force is becoming popular now. Discussions on how these devices are authenticated with other protocols are beyond the scope of this project. We are considering architectural specifications of machines running authentication protocol on In-Memory Computing Architecture. Running recognition and authentication protocol on this RISC processor in faster way need multiple data in-out ports and which is limited to some extent. There is a limited performance against recognition for the devices. Imagine the world of devices and IPv6 which contains the provision of 2128 binary combinations for each device. A newer approach called near associative memory search is a solution. Recognition of these devices using Micron Automata In-Memory Computing with a significant gain of performance improvement and the energy-efficient solution is proposed design in this project.

Literature Review

2.1 Related Work

Sensor nodes are data collection points in the IoT system. Machine learning approaches are typically used to model data that is coming from sensor nodes of IoT. These machine learning implementations are computation-intensive let them deal with a high-end server-side running ML implementations. However, IoT devices are provided in increasing their capability of processing and downward capability as well (Discussed in detail in Chapter 5). Similar work was done with appliance recognition [5], and their architecture relies on a web of things. However, there are many common factors that are equivalent to use in this kind of scenario and others related to IoT devices recognition and other problems implementations. There are also fixed capabilities of these designs to which is that devices are just getting data which is more precisely called data collection using sensor networks and react based on that data collection. The reaction of devices is entirely autonomous if they are the only ones to make a decision based on that collected data, but this concept is diminishing the role of IoT because in IoT things communicate to each other and that

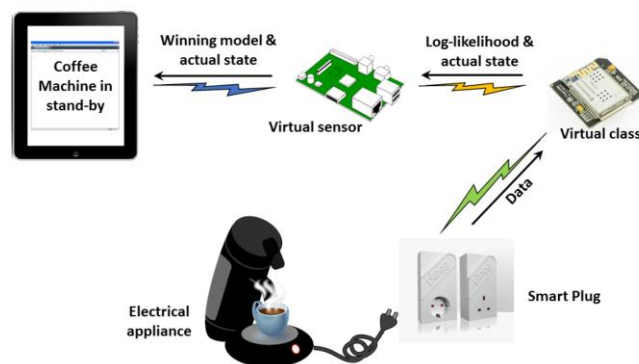


Figure 5 : A concrete Implementation of Network Devices on Web of Things Interface communication is necessary for deciding. For interacting devices together devices need to be connected. There are several networking protocols developed for network-based communication, and OSI reference model is also a reference for the design and troubleshooting the underlying every protocol hardware set. An implementation is

given below in **Figure 5** Where every class of device given an API which is accessible through HTTP protocol were from a client point of view a single GET request will return the status of devices where an intermediary virtual class running on the FlyportPRO Wi-Fi is used to interact with devices through source processor capable of classifying devices and web requests coming on HTTP baseline. However, these small devices correct identification rate is 90% for this small made architecture. Digging deeper into millions of devices it is challenging to get the devices interconnected but before they must get recognized. Since embedded devices and multimedia applications are generating an unprecedented volume of data. Processing this big data generated by thousands or millions of devices is a big challenge for today's computing industry. In-Memory Computing¹ has become a new domain to cope up with challenges coming in today's era of big data processing. In Von Neuman, Architecture data movement (in and out) has become a bottleneck limiting throughput and latency. Runtime energy-efficient solutions can be made through Near-Data Processing reducing the data movement.

Application of Near Memory computing. Lee and Kotalik codesign and evaluate kNN in which similarity search is implemented using highly parallel distance calculation and global k-top search [6]. They present new non-deterministic finite automata using temporal encoding; it can be used to evaluate kNN. Comparison of AP with state of the art CPU, GPU, and FPGA's with the current generation of AP processor achieve 52.6x speed up while maintaining competitive energy efficiency as heterogeneous computing substrate. They proposed several automata processing techniques that can be adopted to get the additional performance of 73.6x available in next-generation AP hardware.

CAMs are the first commercial substrates that were coming with near data processing limited features. Nevertheless, the accessibility of CAMs was limited which make it difficult to program access and evaluate. Random proposed all pair similarity search accelerator extension for L2-Cache to accelerate NLP applications. AP pattern mining and similarity search are well-explored application area previous work has

¹ Here we use In Memory Computation and Near Memory Computation Interchangeably.

shown that AP provides a significant improvement in energy efficiency in Biological Motif Search, Brill Tagging and Outside pattern mining. AP is also significant processing bench for graph processing (finding cliques and Hamiltonian). The use of DRAM Technology to implement NFA leads to high capacity and therefore provides abundant parallelism on pattern matching.

Nevertheless, Moore's Law is slowing in which power constraints (power wall) is a significant issue limiting the scalability of CMOS and on the other side, limiting the von Neumann architecture to support higher degrees of parallelism required in irregular application data sets. In data centers and IoT servicing, only hardware accelerators with broad applicability to general-purpose systems are likely to gain adoption in near future. Pattern matching, recognition, and searching are becoming increasingly common in data mining, data warehousing, cybersecurity and IoT. Tagging objects in IoT system fundamentally creating a virtual world of talking machines giving them identification similar approach of swarm machines in robotics, but implementations are different.

2.2 IoT literature overview

“Profound technologies are those became disappear” (Anonymous). They are distinguishable until they do not come into daily life routines. IoT is now becoming the most profound technology in today's world. Things become challenging as well. Ability to code the objects, track and redirect them become efficient industrial processing gains, reducing errors, preventing the theft incorporating the flexible organization's systems through IoT.

Objects in IoT are not only electronic devices that we assume to be the devices generating data and communicating incorporating but this is also applicable to the daily technology advanced devices products, instruments, gadgets, clothes, furniture's all persons, animals, chair, fridge, tube light, dishes, home appliances, things or objects in real worlds. The more in-depth look of IoT shows a global network that allows interacting machines to machines, human to machine, or human to human, and every object is provided with a unique identity.

Here are some aliases of IoT WoT, Technology Omnipotent, Internet of Objects, Embedded Intelligence Connected Devices, Omniscient, and Omnipresent which is

the integration of computation and physical processes. Requirements of building IoT systems are listed below which came into a general consideration of IoT system establishment.

- Dynamic Demand for Resources
- Application Availability
- Data Protection
- Efficient Power Consumption
- Near Application Execution
- Access to open and Interoperable Cloud system

IoT is an umbrella containing many terms as wireless networks, Internet Connected wearables connected to the Internet, embedded systems with low power consumption, Radio Frequency Identification Tracing, Mobile Phones, Smart Homes, V2V. Accessing the cloud system is the scope of this project. Whereas IoT

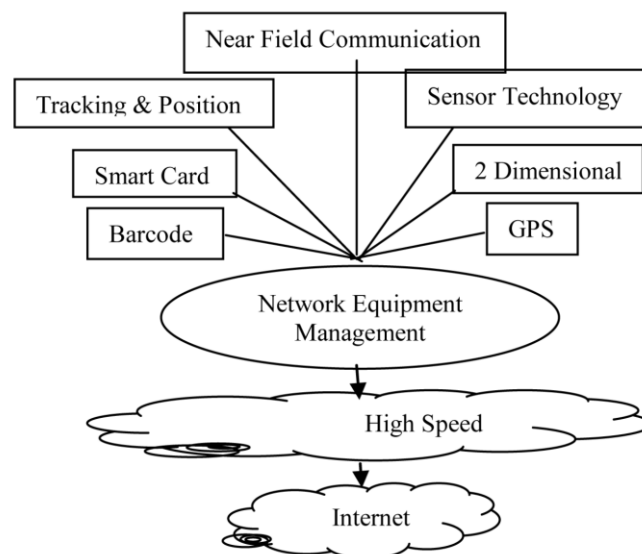


Figure 6: IoT Model for Research and Development of SSME

is generally divided into three layers of operations. a) Hardware b) Middleware c) Presentation discussed in detail in later sections of **IoT Hardware**. The initial Model of IoT research was as given below in Figure 6. Internet of Things was inspiration from RFID community concerned with identifying information of objects tagged by browsing an address or corresponding entry Radio Frequency identification using

Near- Field Communication. Some well-known technologies (Wikipedia) of IoT are given below in the following sections.

RFID: Radio Frequency Identification system is one in which objects are identified by objects or persons connected through a wireless medium with a frequency range of radio waves composing a serial number of the object. RFID of objects first time in history used during 2nd world war. One of the important issues of objects in the cost of an effective manner. Active RFID, Passive RFID and Semi-Passive RFID are three RFID tags used in numerous wireless applications.

IP: Internet Protocol (IP) is an Internetworking Protocol used on the Internet which is developed in 1970. Two versions of Internet Protocol are IPv4 later IPv6. Each version differently describes the address and range of support of distinct addresses are also different. IPv4 gives 4.3 billion addresses and IPv6 provides 85,000 trillion addresses.

EPC: Electronic Product Code is 64 or 98 bits code. It is embedded inside the RFID tag. EPC has unique Serial of the product and its specification and its manufacturer information. Components of EPC are ONS, EPCDS, EPC, and EPCSS.

Barcode: Barcode is a method of encoding letters and numbers into bars (vertical lines) with different widths. Recently QR code has become a more significant interest due to reliability and storage capacity.

Wi-fi: Wireless fidelity is a well-known network technology using which computers and devices get connected through a wireless connection. Integrated wifi into notebooks, handheld, Consumer Electronics nowadays entirely becoming wireless cities.

Bluetooth: Bluetooth is a wireless short distance technology over radio waves used for communication between closely placed static of mobility devices. Communication using Bluetooth between various devices, handheld PCs, PDAs, Cameras printers within the circa of 10 – 1000 meters.

ZigBee: ZigBee is a low-cost development of wireless technology medium used for short ranges and small data transfer. It is a scalable, reliable and flexible protocol design.

NFC: NFC is a small-range wireless technology with ~13.56 MHz within a distance of 4.0 cm. It does not require a line of sight communication and work in a dirty environment.

Actuators: Some instrument which converts electrical energy into mechanical energy. They have a direct impact on something that needs to be operated through actuators using automated decisions.

WSN: Wireless Sensor Network is the physical distribution of autonomous devices. These networks are used in the monitoring of collectively the physical states of the environment, including temperature, sound, vibrations, pressure, etc. WSN based IoT being receiving astonishing attention in healthcare, military, fire control, and flood detection, etc.

AI: Artificial Intelligence is sensitive and responsive to electronic environments. Embedded, Context-aware, adaptive and personalized systems are common characteristics of AI systems.

IoT is full of applications in medical, manufacturing, industrial, transportation, education, governance, and mining. Some flaws are limiting IoT growth are there is no standard architecture of IoT, Universal standardization is required, Technologies are not converged and need of standard protocols.

2.3 IoT Reference Architecture

IoT architecture has some requirements that contribute to the uniqueness of the IoT network. These requirements come from a usage perspective, technology, the way IoT devices are manufactured [7], also existing best practices of server-side internet

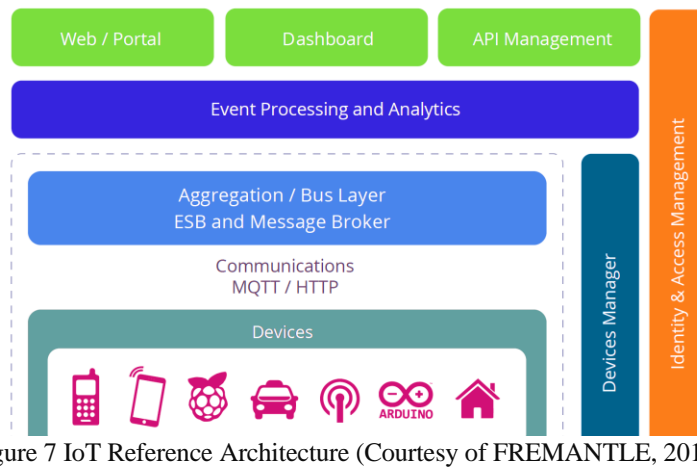


Figure 7 IoT Reference Architecture (Courtesy of FREMANTLE, 2015)

connectivity. These requirements are connectivity and communication, Security, Scalability, Integration, Predictive Analysis, Data Analysis, and actuation. We will not go into the details of these requirements. The reference architecture contains set of components. All the layers shown in **Figure 7** are realized from specific technology. In this scope, we work closely to Event Processing, Device Management, and Identity, access management layer of IoT

2.3.1 Device Layer

The lowest layer of IoT architecture is called the device layer. This layer must contain a direct or indirect connection to Internet. There are numerous devices and have an identity. Identity may be a unique identifier(UUID) burnt onto a device. UUID is burned in the address provided by the manufacturer of the device (Bluetooth Identity, MAC Address). Preferably these devices should have unchangeable ID.

2.3.2 Communication Layer

The communication layer is responsible for the connection of devices and the integrity of communication between devices. Well, known protocols for this communication between devices are HTTP/HTTPS, MQTT 3.1/3.1.1, and CoAP.

2.3.3 Aggregation Bus Layer

It is essential due to broker's communications. It can transfer HTTP server and MQTT broker. This layer maintains a combination of a communication protocol for different devices. It also can transform a bridge between different protocols. It must also be able to perform the PEP for policy-based authentication access. Requesting the identity device will access management layer to validate access requests. The bus layer responds through PDP to permit or deny resource access.

2.3.4 Event Processing and Analytics Layer

This layer takes the input of triggering events from the bus and provides processing ability upon these events. This layer stores data into a database where each device has initiated his event now stored in the database. The traditional approach is to write a server-side application that handles requests from clients and then acknowledge the requests allowing or disallowing the device. EPA uses the scalable map, reducing analysis of data from devices. Secondly, initiate real-time event handling. These approaches need to be highly scalable, more abundant data storage for storing events. Complex event processing in a fast manner using computing inside memory called in-memory computing system with real-time actions based on the data and activity of devices. This approach is design issue of this project and discussed further in coming sections of the report.

2.3.5 Client External Communication Layer

This reference architecture needs to provide ways to devices to communicate inside device oriented as well as outside the device-oriented network.

2.3.6 Device Management

The device management layer is composed of two parts. One is server-side application with established connection of devices and communication is done by many protocols. The device manager is the name of this server computing base. The device manager is responsible for maintaining device identities and be able to work with access

management and identity layer so that to manage access control over devices. It is profiling of devices that tells how much control the device has and how much control administrator has for that device. DM layer maintains the record for device's locking the devices which are compromised with the management of security and identifier control.

2.3.7 Identity and Access Management

This layer as the name implies, provides a token for issuing and validation. Identity services OpenID Connect support used for identifying inbound requests from the web layer. Directory or users with policy management for accessing controls.

2.4 Micron Automata Processor Architecture

Micron Technology recently has introduced a non-Von Neuman architecture based Micron Automata Processor. We have discussed an overview of the Micron Automata Processor in Introduction Section 1.3. Its architecture is based on Non-von Neuman machine which is an efficient computational model for Non-Deterministic Finite Automata. NFA is primarily symbolic pattern-matching machines. Non-Deterministic Finite Automata is extraordinarily fast because it allows several states to be active at the same time leading to massive parallelism.

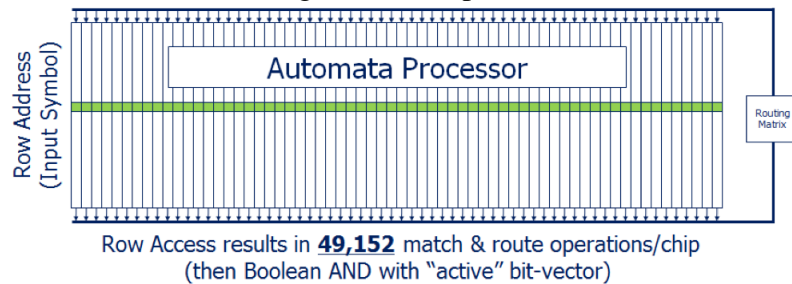


Figure 8: Micron Automata Design (Micron)

Since Non-von Neuman is efficient for NFA represents a transition table in memory, each entry represents a state following with successor state. An arbitrary number of states must be active at the same time. Von Neuman architecture cannot support a large number of random accesses leading to cache misses with higher miss penalties. GPUs are also not accommodating high bandwidth demand of NFAs. Regular

expressions and DFA need to be converted into NFA in which only one state can be active, which is a challenge in this project to do and discussed in detail in the Methodology Section Chapter 5. DFA miss rates are very high often 100% [8] and higher misses in rest of the levels of cache.

Implementation of NFAs is, if done on native hardware is a compelling solution allowing arbitrarily many numbers of states to be active at a time. NFA is allowed to explore as many permutations as possible. Figure 8 shows how Micron Automata uses memory. Each column in memory represents a single NFA state. The native symbol of automata processor is 8 bit, which leads to 256 different possible inputs can activate the corresponding row of the array and can read the result of NFA on that symbol. These are combined with bit vectors, which indicate which state is active. Several states can respond to each input symbol concurrently every clock cycle.

2.4.1 AP Boards Architectural Specifications

The current generation of AP comes with 32 chips per board. Sixteen chips automata boards are also available does not matter 16 or 32 chips are available onboard each chip is operating at the rate of 133 MHz. These boards are standard PCI Express boards. Current boards allow the AP chips partitioning among multiple concurrent data streams. This allows the full utilization of available interface bandwidth. FPGA is used as a routing fabric on AP boards. It contains DRAM memory controller as well for interfacing with AP ranks. FPGAs are used to develop acceleration pipelines and following the AP to perform pattern matching tasks. The program is launched on AP using classical offload model. The host Processor will perform initial configuration of NFA inside AP. Sending data to AP and then receive the results. Multiple passes are required in case the candidate of interest exceeds the board capacity but only the pattern, not macrostructure needs to be updated. Micron's AP SDK provides ANML language for creating designs of automata networks also Java, Python and C interfaces are available. Macro is like an ordinary subroutine in other programming languages. It involves two

phases configuration of AP board. Placement and Routing Compilation and loading configuration. AP compiler generates binary sequence of automata networks. Table 3 gives the detail of AP board features.

Table 3: Micron Automata Processor Board Specifications

Micron D480 AP Board	
Processing Chips	32/16
STEs Per Board	1536 K / 768 K
Ranks	4
Chips Per Rank	8
Communication Interconnection	Altera FPGA
Buffer Size	4 GB
FPGA Logic Elements	270,000
FPGA ALMs	101,620
On-Chip Memory	17 Mb
On Board Memory	DDR3
Programming Platform	Micron AP SDK
SDRAM	4 GB DDR3
OS Compatible	Centos 6 SUSE 11 Ubuntu 14 Windows 7
Configuration Time	50 Milli Sec
Cooling	Active/ Passive
Temperature	0o to 45o C
Power Consumption	300 watts
Interfacing	PCIe Compliant x8 land PCB Connector
Safety	RoHS Compliant

Theoretical Findings

3.1 Similarity Search

Similarity Search shares a principal of searching in an ample space where only correct matching comparator gives the identification of paired objects in storage space. In this era of an unprecedented volume of information generation, sources of existing similarity searches have become a pivotal idiom to solve problems in searching domains. Index matrices are used to obtain scalability in similarity searches.

3.1.1 Nearest Neighbor Search

Nearest Neighbor is a searching method of finding point against the given point, which is most near to given point. Near is a sense of similarity. It is like post office problem by assigning to the nearest post office before residence. kNN discussed in later sections is a direct generalization of Nearest Neighbor Search.

3.1.2 Metric Search

Metric Space is a set containing distance vector of all members if the set defined. Metric Search takes place inside metric space where distance is defined before. Metric Search is usually applied to relatively static data collections and use the properties of metric search space and allow similarity searches to be performed in an efficient manner.

3.1.3 Locality Sensitive Hashing

LSH is a well know similarity search approach in which items hashed with similar items into the same bucket, which is much smaller than universal data items.

3.2 Applications on AP

Due to the generation of an unprecedented volume of data from embedded and multimedia applications, processing of data is required first to be in the index and searchable. In 2010, 260 billion images were shared on Facebook (Beaver & Kumar, 2010) whereas 300 hours of YouTube per minute uploads (YouTube, \Statistics -

YouTube," , 2014.). In-memory computing is a better option to make these operations possible with expected rates of performance.

3.2.1 KNN Implementation on AP and Speedup Statistics

K-nearest neighbor is an approach commonly for similarity search consisting of many parallelizable distance calculations and a serial global top k sort. kNN is an ideal candidate for near data processing due to its general implications with simplicity and parallelism with a small set of results. Application of kNN on Micron Automata Processor in [6] has shown **52.6x** speedup over the multicore family processor and with evaluating potential impact lead to the **73.6x** performance speedup with their novel non-deterministic finite automata.

Micron Automata is a similar hardware design for non-deterministic finite automata. Complete system architecture for implementing near memory search

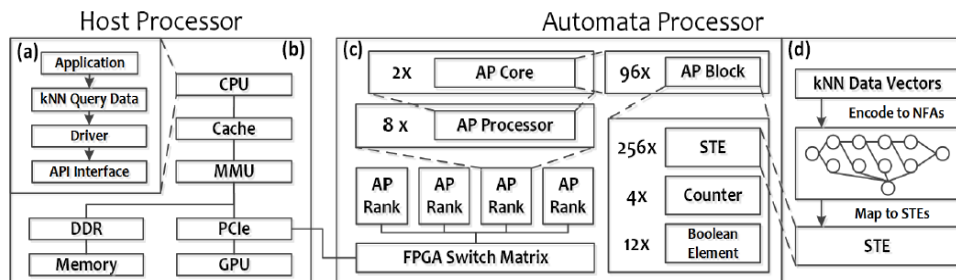


Figure 9 : Complete System Architecture of Micron AP for kNN implementation of Near Memory Similarity Search Design [6].

on micron automata is given in Figure 10. The software stack for AP NFA Application in Figure 10(a) gives the application design overview which is running on host processor Figure 10(b). FPGA and PCIe bridge the communication gap between the host processor and AP. Figure 10 (c) is the internal architecture of automata processor, and configuration overview of mapping kNN NFA on STE is given in Figure 10(b). Each core of AP is divided into two halves consisting of 96 AP blocks, and each block is composed of 265 STE giving 24576 STEs per half AP core and 1572864 STEs per board. Every state of NFA is mapped to one STE. Every AP block has 4 counters and 12 BEs. Counters are incremented activating down streams. Getting this abundant speedup is true, but AP has some constraints directly relevant to the application

and natural language to be modeled for that application. Performance analysis and energy efficiency of AP for kNN search algorithm are given in Figure 10 Below.

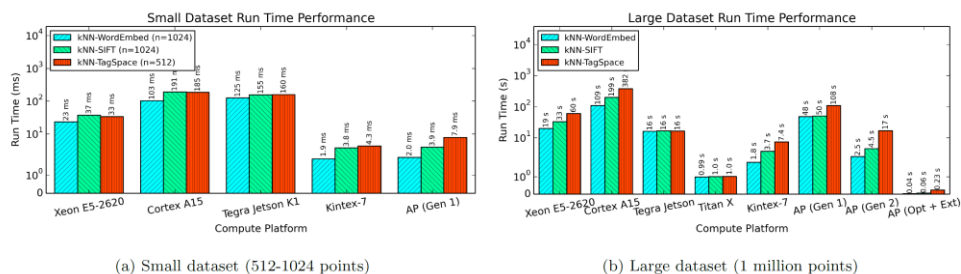


Figure 10 : Performance of AP with kNN variants (lower is better) (*T. Lee, Kotalik, & C., Near Memory Similarity Search on Automata Processors, 2016*)

3.2.2 Common Issues with AP

AP hardware is not provided with arithmetic units, whereas STEs can be modeled by taking them as part of the lookup table, but it is inefficient to do any arithmetic computation in AP hardware. AP hardware assumes that every single input is derived from host processor. Therefore, any feedback or output of states cannot be combined to dynamically form other nested NFAs. Symbol stream must be defined statically before being streams into AP. There is not a dynamic modification of symbol stream.

3.2.3 Energy consumption of AP

Micron AP Gen-1 provides 43x energy efficiency over the general-purpose core. Typically one chip of AP draws 4W of power. Keeping the reconfiguration time into consideration, FPGA and GPU comparison of energy with AP surprisingly gave better energy efficiency than both targets. Gen2 AP has architectural improvements that will considerably give higher energy efficiency than FPGA fabrics as well [6]. Energy consumption for kNN search algorithm shown in

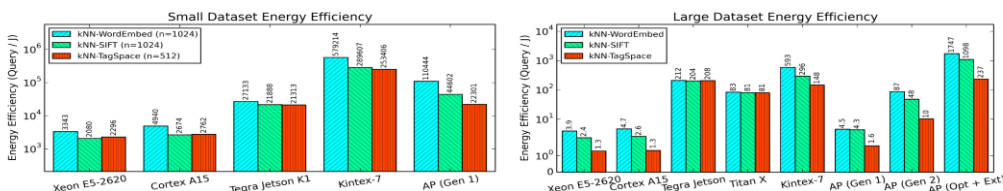


Figure 11: Energy Efficiency Comparison

comparison to CPU and GPU in Figure 11 below (a) contains smaller data and (b) Large datasheet. Some optimizations of AP and NFA lead towards higher performance and greater energy efficiency. Automata Optimizations are Vector Packing, Symbol Stream Multiplexing, and Statistical Activation Reduction. Whereas architectural Extensions are Counter Increment Extension, Counter Dynamic Threshold Extension and STE Decomposition Extension.

3.2.4 AES Encryption and PROTOMATA

PROTOMATA is finding patterns in the protein samples of 300 bytes each, and their execution time is given below in **Table 4**. Comparing with intel Pentium

Table 4 : Execution Time of Finding Protein Patterns

Proteins (300B) \ Patterns	1	10	100	1000
1	$5.1256 * 10^{-6}$	$5.3938 * 10^{-6}$	$8.0758 * 10^{-6}$	$3.4896 * 10^{-5}$
10	$4.5356 * 10^{-5}$	$4.5624 * 10^{-5}$	$4.8306 * 10^{-5}$	$7.5126 * 10^{-5}$
100	$4.4766 * 10^{-4}$	$4.4792 * 10^{-4}$	$4.5061 * 10^{-4}$	$4.7743 * 10^{-4}$
1000	$4.4707 * 10^{-3}$	$4.4709 * 10^{-3}$	$4.4736 * 10^{-3}$	$4.5004 * 10^{-3}$
10000	$4.4701 * 10^{-2}$	$4.4701 * 10^{-2}$	$4.4704 * 10^{-2}$	$4.4730 * 10^{-2}$
100000	$4.4700 * 10^{-1}$	$4.4700 * 10^{-1}$	$4.4700 * 10^{-1}$	$4.4703 * 10^{-1}$
1000000	4.4700	4.4700	4.4700	4.4700

IV gave same results of 1000 Patterns in 10 minutes. Whereas in AES Encryption AP outperforms from state of the art CPU but not the GPU due to fabrication technology difference and the NFA fragmentation given in Table-5 below

Table 5: AES Encryption Algorithm Performance

<i>Architectures</i>	<i>Processing Rate</i>	<i>Throughput</i>
AP	133 MHz	6 Gbps [2]
Intel AES-NI	3.47 GHz	0.5 - 0.7 Gbps [9]
Nvidia GPU	900 MHz	14.6 Gbps [10]

3.3 IoT Hardware Architecture

Hardware choice of IoT is depended on the cost of the product for problem solution, the experience of user capabilities of the application. IoT Project Managers after many surveys from hundreds of PM in IoT devices implemented solutions, only 27% of the application PM is aware of Hardware, and more than 70% are not aware of Hardware of IoT solution implemented in their management [9]. Companies are

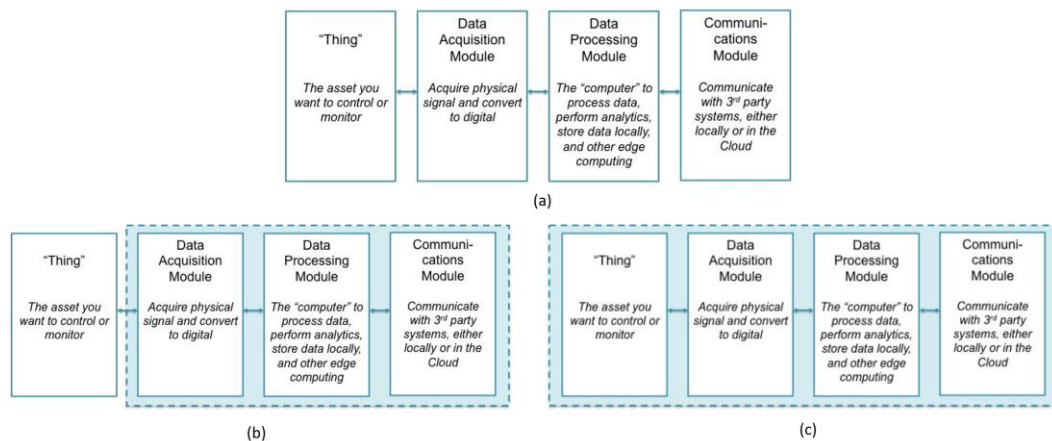


Figure 12 : Building Blocks of IoT devices (a) Commonalities in IoT Hardware (b) Thing is stand alone as a dumb device (c) Thing is fully integrated into smart Device. (*Elizalde, 2017*) making their existing assets smart to connect them to cloud. Either company build their new devices or change their older devices to new smart devices. An essential thing which is the first building block of IoT is shown in Figure 11 first block in all (a)(b)(c) variants of IoT devices. The second basic block is data acquisition module, which collects data from field and converts signals into digital for processing. DAC can also perform some small analytics on data but dependent upon nature of DAC. Processing power and what data capacity needs to be processed is the primary focus of Processing block which is a generically third block of IoT Hardware architecture. The fourth module is Communication Module; this circuitry is used for communication with the cloud platform. Contains ports USB, Serial (232/438), Profinet or Modbus, and used the technology may wireless or wired to connect to network cloud. Next sections contain a more detailed view of processing machines used in IoT.

3.4 IoT Processors

IoT includes everything wearables, smart framing energy, healthcare, and multimedia devices. Types of processors are dependent upon the sensing needs

and processing capacity as discussed in 3.1. Some broad processors categories are given below, [10].

Smart Sensor is microcontrollers connected via Wifi containing many interfaces of sense. The performance of CPU is ~100 MIPS with energy-efficient standards of connectivity.

Connected Audio Devices are equipped with processors capable of around 1000 DMIPS. I am starting from Bluetooth speakers to home cinema systems.

Connected Video Included IP cameras and need to have powerful GPU for the rendering of 3D graphics. Like connected audio devices except the video processing engine integrated inside due to this reason came into more exceptional performance delivering devices.

Multimedia Rich Devices Are vibrant processing nodes in IoT requiring processors, which are a full range of processing capabilities for complex workloads. Processing circuitry consists of Multicore CPU and powerful 3D rendering image processing engines. These devices require prior processing before sending data to the cloud. This requires big processing hardware, which is a Multi-Core GPU pipeline and processors capable of processing AI applications as well.

High-Density Computing Nodes These computing systems include large data centers, Storage Networks, Cloud Computing delivering high performance and energy-efficient. Complex algorithms are handled in Many-Core CPUs with GPU, DSP, and FPGAs with hardware multithreading allowing devices to scale much better performance. The heterogeneous system gave energy efficiency and minimal area solutions.

3.5 Reasons for MIPS as Ideal Processors for IoT

There are many factors for choosing a correct processor for IoT devices as given in 3.2 in IoT; these factors are power/area, performance, software support, and security. However, the hardware architecture is not the only way to get IoT processor chosen. There are different ways of choosing processor [11].

3.5.1 Virtualization Support of Hardware

MIPS M51xx class CPUs have five different processors with full hardware virtualization support giving seven guests VM on a single processor. Hardware virtualization gives some new functionalities of IoT system. For example, using hardware virtualization your processor kept running firmware update, application in isolation and if firmware updates become corrupted then functioning of the smart device on the application will not be affected.

3.5.2 Elegant Architecture

MIPS architectures provide unique features to strengthen IoT hardware. MIPS processors provide 32 GP registers double in strength to other CPUs exposing to the programmer. Individual shadow registers are also provided for faster context switching. MicroMIPS is a class of MIPS processor that is provided with low-cost solutions using less memory for a semiconductor device. Small instruction size efficient addressing scheme small code size and provide small money solutions with lower manufacturing costs. These features are also best suited for cryptographic processing. MIPS M class CPUs are 20-25% faster in MHz than home kit sessions.

Core	Cortex-M0	Cortex-M3	Cortex-M4F	microAptiv UP
Instruction set architecture	ARMv6-M	ARMv7-M	ARMv7E-M with FPv4-SP extension	MIPS32 with DSP enhancements
Clock frequency	16 MHz	48 MHz	64 MHz	200 MHz
Set up accessory – first phase with static setup code (with dynamic setup code: 3 times as long)	5.0 s	1.5 s	0.45 s	0.12 s
Set up accessory – second phase with static or dynamic setup code	14.9 s	4.3 s	1.33 s	0.35 s
Open session	0.94 s	0.26 s	0.06 s	0.01 s

Figure 13: Performance Comparison of MIPS M-Class Processor

3.5.3 Memory Management Unit and Cache Memories

Several CPU configurations of MIPS came with MMU and cache controllers. M-Class processor can run RTOS, including TCP/IP stack and Ethernet driver. This capability makes M class CPUs to support rich OS with lower power TCP/IP stack connectivity. This allows creating more sophisticated UIs.

3.5.4 DSP for Audio Processing

Voice is one of the best ways of communicating with IoT devices. M-Class MIPS CPUs are equipped with instruction set available in the industry and give 2x

speedup competing with CPUs. An entire voice processing system for IoT made by omgtec [12] SoC solution for voice processing.

3.5.5 System Integration and Cloud Connectivity

Next-generation of IoT devices is combined Esigma IP, Power VR, MIPS. This SoC solution will give a complete secure connected processor solution for Industrial IoT and small consumer market. Creator Ci40 consisting of 6LoWPAN

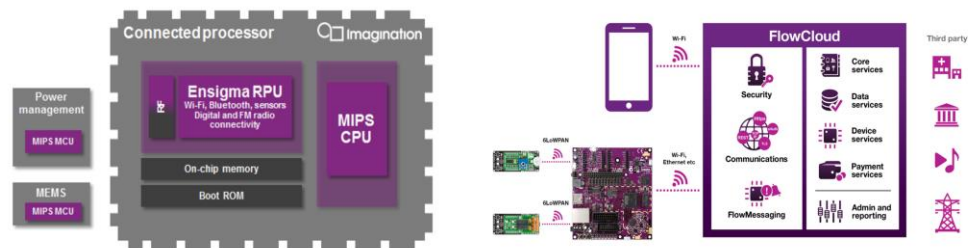


Figure 14 : Complete SoC Solution of M Class IoT Processor and, Cloud Connectivity using Flow Cloud low-power sensors energized with MIPS MCU is a high-performance point integrating an MIPS CPU and Enigma RPU. Developers are not locked to Intellectual Property. This design kit is wholly based on open-source software technology and stays connected to the cloud via Flow Cloud Figure 13.

3.6 Examples of IoT Connected Devices

There are some interesting examples of IoT devices [13]. In-Car Wi-fi gives the ability to connect old car without buying a new one using MIPS-Powered Sequans 4G chipset then the distribution of signals within the car using Wi-fi. Users can monitor the location of their car at any place and any time. It eliminates the use of smartphone hotspot inside the car. It turns on automatically when your car is started and does not need any battery and can connect the maximum ten devices. Sugar look HD camera dock is another example which is connected via high definition multimedia interface to TV and play footages from SD card USB interfaces. Arduino Tian a newer Arduino Open Source Board which is designed for many IoT-based projects implementations also in wearable electronics. ChipKit Wi-fi Dev board is a new power and compact designed board. It is used in IoT and many consumer-focused projects.

Building Blocks of NFA in Micron Automata

Finite automata are divided into two categories with output-oriented and without output. The figure gave pictorial representation of Finite Automata.

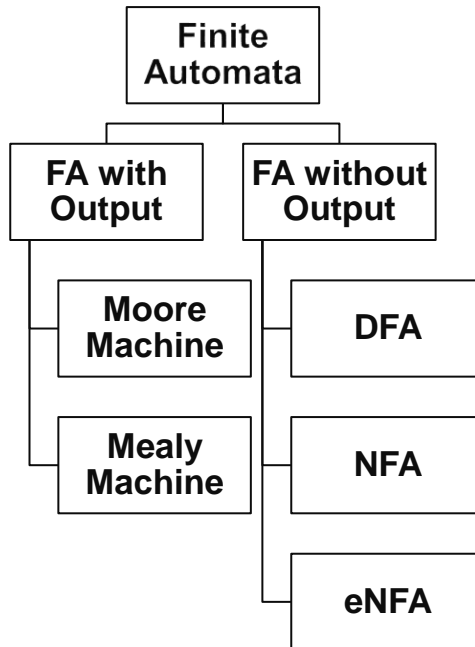


Figure 15 : Finite Automata Types and Categories

4.1 NFA Construction

NFA inside AP is constructed using elements and connections. Every state of NFA is modeled as an element of AP representing states of NFA. Directed connection routes between elements called connections.

4.2 State Transition Elements

The fundamental element inside AP is STE. Each STE defines the symbol set that provides the matching characteristics of input symbol. A single STE is direct model of NFA state.

4.3 Counter Elements

Counter provides special provisions for counting events based on STE behavior. These counters will be updated when specific state is achieved. For Example it may be used to count how many matches of the same string occurred in one against one single input symbol. Counter elements associated with STEs are processed in the same symbol cycle as of STE.

4.4 Boolean Elements

Boolean elements behave the same as counter elements in same symbol cycle output of Boolean Element depends upon input symbol. These elements are configured to see the values of input variables from incoming connections. Basic AND, OR, NOT and NAND with SOP and PoS expression simulation can be done using these logic elements.

4.5 Input-Output Elements

Input-output elements are start and final states of automata respectively. Start state also referred to as initial state is activated using input stream, and rest of the states will be dependent upon activation of their previous state, and final state is also activated when right back driving state is active. The final state when becomes active it is said to be reporting state.

4.6 Automata Network Markup Language

ANML is a comprehensive language for defining automata networks. ANML also provides macros construction for reusability of automata. Automata components must be described between matched automata network tag. All components have a unique ID at their opening tag. Each element is defined with a specific tag definition. If initial, reporting and latched behavior is not necessary all will use so-called default

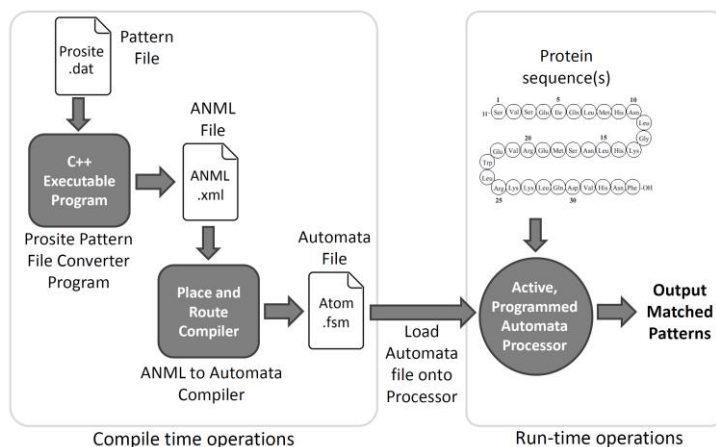


Figure 16: Example of Protein Automata Mapping on AP

Unspecified. State Transition Elements specifies the symbol set. Counters will define the target, and logical elements must specify the logical configurations. Ongoing elements connections will also be defined. The active-on-Match tag is used for this connection purpose. With this basic understanding NFA can be defined for AP.

Methodology and Quantitative Analysis

5.1 Objective

We will configure AP for finding a match of identities among requesting devices that need to be authenticated. We considered only 4 types of identifications which generally used for network devices, mobile phones, and any electronic device i.e. serial number.

5.2 Device Identifiers

MAC Address: Media Access Control Address consists of 48 bits and is burned in the address of devices (NIC) manufacturer. These are factory programmed 2 Kib to 256 Kib serial EEPROMs and write-protected for global uniqueness. Its applications are wireless, networking, Bluetooth and Ethernet, etc.

Unique ID Serial Number: These are preprogrammed and distinct 32-bit Microchip serial numbers and also scalable to the variants like 48,64 and 128 bit and subject to the requirement of ID length. These identifications are generally used with device identifications, healthcare, medical, application authentications, and security. EEPROM devices under Microchip Technology can be custom programmed, so identification is subject to the domain of application where systems are also working; it is subject to the strength of devices, machines need to be authenticated.

Open Device Identification Number: ODIN is an identifier derived from the MAC address and hashed to derive user security and privacy.

Unique Device Identification Number: is a 40- character string identifier of Apple devices, which includes iPhone, iPads and iPods.

5.3 Programming Challenge

AP is the first device in the semiconductor to directly emulate the execution of Nondeterministic finite automata. AP is a scalable solution to finite automata problems. ANML language is used to program Micron Automata Processor. Utilizing the Micron Automata Processor is to give problems in the format of NFA, which is a challenging task to convert the problem format into NFA.

5.3.1 NFA implementation on Sequential Processor

Sequential Processors are Von Neuman Processors which take the simple approach of modeling the NFA by defining the states and connections and maintain the directory to keep track of which state is active and which is an inactive giving token to the previous state to get an update on the newer state. Since sequential machine performs every operation in a sequence, and worst case for the elaborate graph of NFA will be a processing complexity of $O(n^2)$. Whereas for direct storage solution complexity will be $O(n)$, which is unrealistic in terms of larger n .

5.3.2 NFA implementation on Automata Processor

NFA implementation on Micron Automata Processors is physical due to the granularity of STEs in the automata processor (Described in 2.4.1). Each element can be dynamically connected to other STE elements in AP core space, which refers to configuration type AP. That is why automata is capable of modeling NFA with direct configuration mapping. NFA storage cost for automata processor is $O(n)$ see Table-4. Since AP is abundant in storage context so if n is larger it can be easily managed on AP hardware.

On the other hand, if n is too large to accommodate on AP it can be configured again with rest of configuration. Since configuration time of AP is much less (~ 50 ms). Processing cost is proportional to stages of automata. AP can provide all inputs simultaneously, and all the states will do their processing in single time step which gives the processing cost of $O(1)$.

Table 5: NFA Implementation computation and Storage Cost

Processing Machine For NFA	Storage Cost	Processing Cost
Von Neuman Sequential Machine	$O(n)$	$O(n^2)$
Automata Processor	$O(n)$	$O(1)$

5.4 Constructing the NFA

Creating NFA is a straightforward process in case of string matching as of our requirement, we must compare device identifier strings and go through the matching

criteria. Constructing the NFA is every character in the candidate string needs to create a state for that character. States will be connected in the order in which they appear in the string order. The final state of the automata will be activated if given string is matched and passed through all previous states. One character is processed in one-time step which is called the symbol cycle of given automata. The inspection of multiple active states in automata network gives an extent of parallelism. Further parallelism is derived from the parallel processing of multiple NFA in same time step. Refer to Figure for simultaneous string matching automata operations.

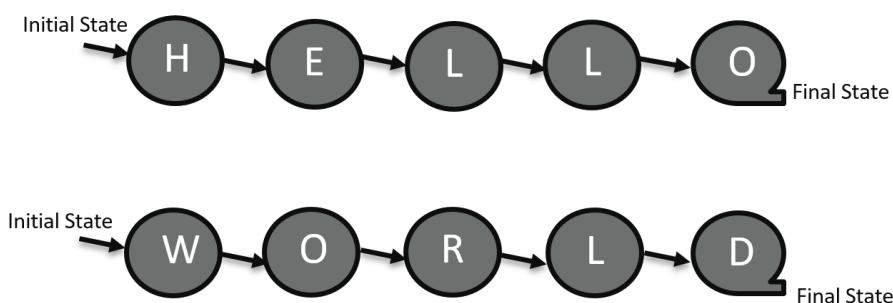


Figure 17: NFAs for Two Strings Matching "HELLO" and "WORLD"

Each AP core contains 24K STEs can be dynamically configured but no connection can be made between two different cores of AP. AP core process input data at the rate of 1 Gbps. Each character composed of 8 bits so calculating the Characters per second.

$$1 \text{ Gbps} = \frac{1,000,000,000}{8} \text{ Characters Per Second} \dots \dots \dots (5.1)$$

$$= 125 \text{ M Characters Per Seconds}$$

Calculating the symbol cycle of every state with processing complexity of $O(1)$.

$$\text{Symbol Cycle} = \frac{1}{125,000,000} = 8 \text{ ns} \dots \dots \dots (5.2)$$

AP core is associated with their ranks in 1,2,4,8 cores. Different groups of cores receive data from data stream in parallel effectively giving a throughput of 8 Gbps. Whereas all cores in one group give processing throughput of 1 Gbps. For effective throughput of NFA computation, AP outperform as compare to its competitors [14]. AP balances the capacity and throughput by when scaling across multiple chips. New

automata can be added² to the chip dynamically without compiling the existing automata again in contrast to FPGAs and GPUs.

5.5 Performance Analysis

AP contains six ranks in its one instance and eight chips per rank. The core cannot cooperate they give individual comparison operations. If the problem is perfectly divided into eight subsets then its execution speed-up becomes eight times that of one core. On the other hand, AP ranks can be connected to host system with single PCI Express port. If we consider throughput AP is a better solution than FPGA-based solutions. For an FPGA to give comparable data computation rate it must process 1-8 characters per cycle (Symbol cycle of AP), but the transition of automata needs to be at the same rate also it requires more considerable data control to process eight characters in one clock cycle of FPGA data path. Some FPGA implementations give 2.57 Gbps [15] throughput but require less capacity as compared to AP roughly half capacity of one AP rank. Another solution [16] of AP comparison with FPGA gives 10 Gbps result but this is consuming 8 characters per cycle and 3.5 Gbps with 2 characters per cycle. All solution gives less capacity than AP. GPU solution gives 14 Gbps [17] GPU solution gives 4 connections from each element whereas the AP allows 16 connections and it can support only one six of the elements of one AP rank. The multiplicity of GPUs is inefficient in terms of scalability possibility with AP. AP is superior in almost many aspects. However, AP does not have a higher throughput in every case.

5.6 Proposed NFA Design

Device IDs discussed in section 5.2 need to be considered for finding occurrence, which is like the functioning of string matching Non-Deterministic Finite Automata. Reporting will be done only when complete string is matched. Device IDs are also represented as strings. For example, 7C-B0-C2-B7-FE-FE is the machine address of device on which this writing is being done. ID is a string of 12 Characters and 5 Special Characters (- or White Space) then NFA for this machine address will be as given in Figure below,

² Used to add new identity of device into the automata network for future comparison of device ID and also for hand off situations.

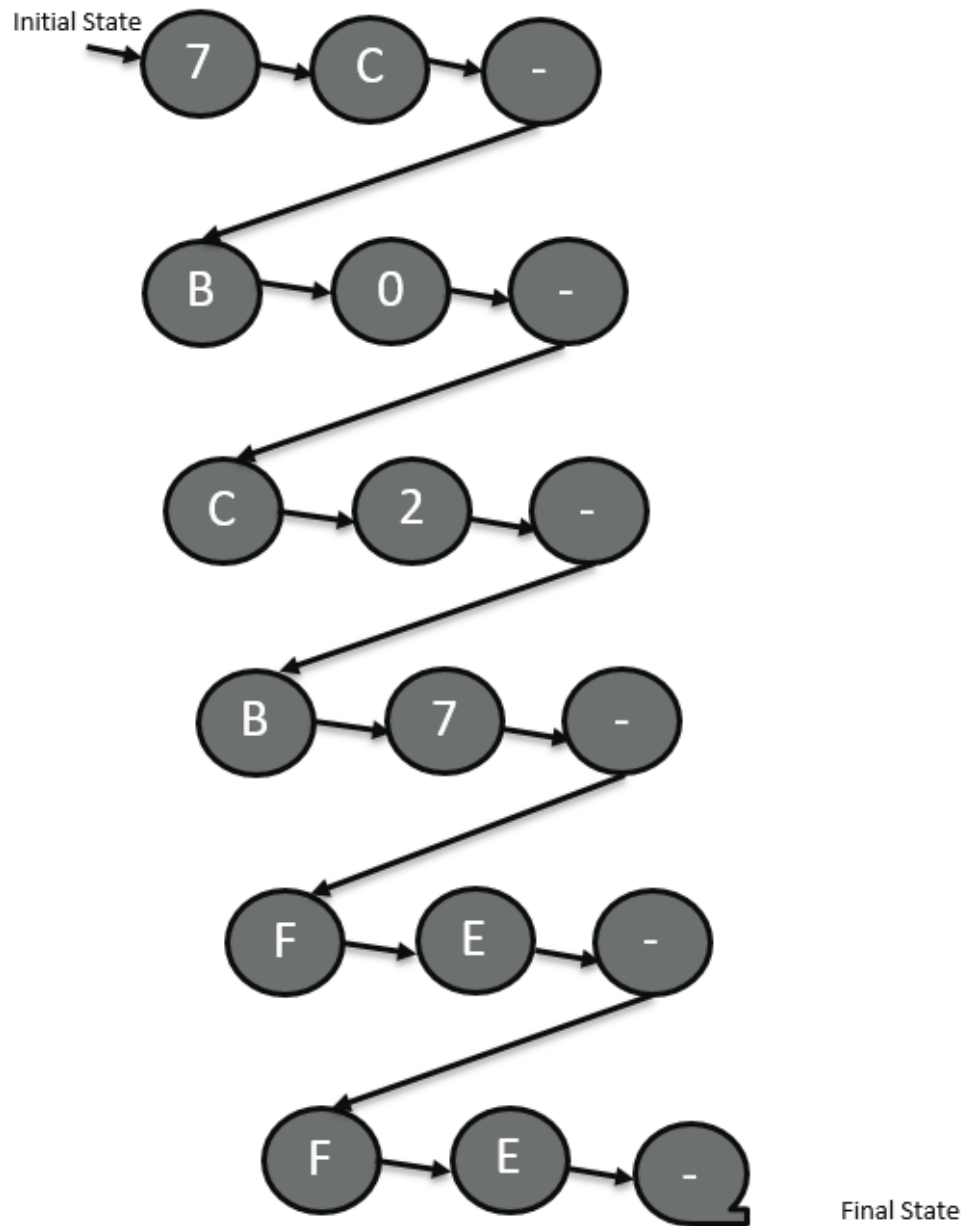


Figure 18 : NFA for MAC Address Matching

We have 48 core AP board. Each AP chip is divided into two halves. One half contains 96 blocks and each block contains 256 STEs.

For a 48 Core AP Board Number of STEs will be

$$= \#Ranks \times \#APs \text{ Per Rank} \times \# \text{ Cores per AP} \times \# \text{ Blocks Per Core} \times \# 256$$

$$\Rightarrow 256 \times 96 \times 2 \times 8 \times 6 = 2359296 \text{ STEs}$$

For one perfect String Matching, we have 18 STEs in one NFA.

$$\frac{2359296}{18} = 131072 \text{ NFAs} \dots \dots \dots (5.3)$$

Different groups of cores receive data from data stream concurrently effectively giving a throughput of 8 Gbps. Then the symbol cycle is

$$\frac{1}{8 \text{ Gbps}} = 0.125 \text{ nsec} \dots \dots \dots (5.4)$$

To get maximum throughput which is ideal case is that we can just recognize devices near 131072 in time given in equation 5.4 avoiding the wire delay.

$$0.125 \times 18 = 2.25 \text{ nsec} \dots \dots \dots (5.5)$$

Though some practical issues like configuration time and data streaming may cause this throughput to reduce, it still has ultra-significant speed. If we change the automata design Figure 17 to address matching omitting the white space of string separator then quantitative values will be changed. Variants of equations are given below.

$$\frac{2359296}{12} = 196608 \text{ NFAs}$$

$$\frac{1}{8 \text{ Gbps}} = 0.125 \text{ nsec}$$

$$0.125 \times 12 = 1.5 \text{ nsec}$$

Initial State

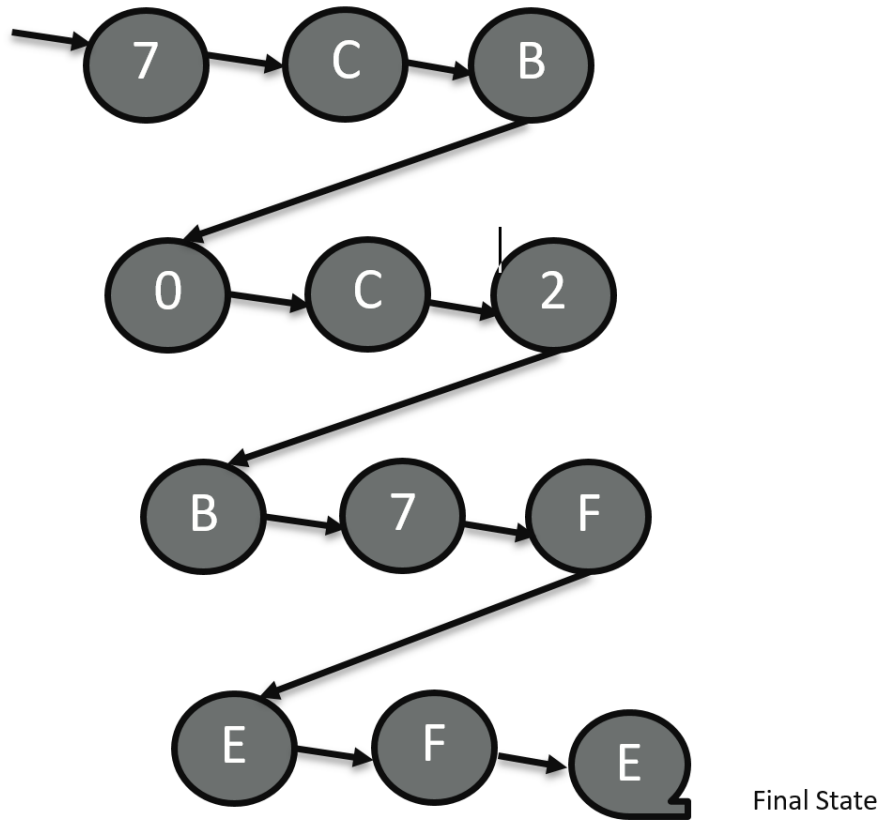


Figure 19: NFA for MAC Address Matching eliminating the string separator

Conclusion

We have seen IoT reference architecture and how In-Memory Computing is deriving IoT. Micron Automata is an In-Memory Computation Architecture which reverses the Von Neuman Architecture approach we have seen how micron automata outperform in specific search and pattern matching analysis from FPGA, GPU and CPU based solutions. Due to higher processing throughput, we have considered AP for IoT devices authentication as a small application and see its performance and architectural aspects. AP is also good processing architecture in terms of energy consumption based on findings. Several aspects are discussed and referenced from different researcher's work. AP Technology is not used in IoT development solutions yet. Authentication is just a small part of IoT can be implemented with this In-Memory Computing. Processing in Memory came into consideration from 1990s but not gained much attention due to attractions of reconfigurable computing solutions. Both IoT and In-Memory Computing are not established in terms of their standards several types of research are being happened on these fields and significant improvements can be made by interleaving phenomenon of both growing field for future solution to digital era of information handling. Micron is also providing SDKs for early adopters to find the solutions to their problems using AP NFA modeling techniques. C and Python interfaces for creation of automata network to be mapped on AP hardware. We were supposed to implement AP hardware, which was one section of the report but unfortunately due to unavailability of AP SDK it was not possible to test simulation results due to which we have to go through comprehensive theoretical findings of our proposed design and find supporting arguments on implication of proposed design. One thing that is challenging in performing computation on AP hardware needs the problem format to be represented in terms of NFA, so we need to convert every problem into NFA before doing any implantation on AP Hardware. AP configuration latencies are also challenging for mapping of NFA on AP hardware. In Future, we can implement this design on AP using Micron Automata SDK which is exciting work ahead.

References

- [1] K. Ashton, "The 'The Internet of Things' Thing", RFID Journal, 2009.
- [2] "Internet Usage Statistics", Internet World Stats, 2017. [Online]. Available: <https://www.internetworldstats.com/stats.htm>. [Accessed: 02- Jun- 2017].
- [3] P. Taylor, "Benefits of 'in-memory computing'", Financial Times, 2011. [Online]. Available: <https://www.ft.com/content/ee237d7a-8c6e-11e0-883f-00144feab49a>. [Accessed: 01- Jun- 2017].
- [4] C. Mc and J. C, "Freedom, Electronics and Tech", hblok.net, 2017. [Online]. Available: <https://hblok.net/blog/posts/2013/02/13/historical-cost-of-computer-memory-and-storage/>. [Accessed: 13- Feb- 2013].
- [5] Antonio and Bovet, "Appliance Recognition on Internet-of-Things Devices", 2017.
- [6] L. T, Vincent, Kotalik and Justin, "Near Memory Similarity Search on Automata Processors", 2016.
- [7] P. Fremantle, "A Reference Architecture for Internet of Things", 2015.
- [8] K. Wang, K. Angstadt and C. Bo, "An Overview of Micron's Automata Processor", in Charlottesville, VA, 22904 USA, 2016.
- [9] D. Elizalde, "Introduction and Explanation of IoT Hardware", IoT Hardware, 2017.
- [10] A. Voica, "A Guide to Internet of Things (IoT) Processors - Imagination Technologies", Imagination Technologies, 2017. [Online]. Available: <https://www.imgtec.com/blog/a-guide-to-iot-processors/>. [Accessed: 24- Jul- 2017].
- [11] A. Voica, "5 reasons why MIPS M-class CPUs are ideal for IoT - Imagination Technologies", Imagination Technologies, 2017. [Online]. Available: <https://www.imgtec.com/blog/5-reasons-why-mips-m-class-cpus-are-ideal-for-iot/>. [Accessed: 24- Jul- 2017].
- [12] A. Voica, "Imagination, TSMC collaborate on a wide range of IoT subsystems - Imagination Technologies", Imagination Technologies, 2017. [Online]. Available: <https://www.imgtec.com/blog/imagination-tsmc-collaborate-on-iot-subsystems/>. [Accessed: 24- Jul- 2017].
- [13] A. Voica, "Four MIPS-based connected devices for the IoT revolution - Imagination Technologies", Imagination Technologies, 2017. [Online]. Available: <https://www.imgtec.com/blog/four-mips-connected-devices-iot-revolution/>. [Accessed: 24- Jul- 2017].
- [14] C. Sabotta, "Advantages and challenges of programming the Micron Automata Processor", Masters, Iowa State University, 2017.

- [15] H. Wang and S. Pu, "A Counter-Based Algorithm for Regular Expression Matching", *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, pp. 92 - 103, 2013.
- [16] V. Prasanna and Y. Yang, "High-Performance and Compact Architecture for Regular Expression Matching on FPGA", *IEEE Transactions on Computers*, vol. 61, pp. 1013 - 1025, 2011.
- [17] Y. Zu and M. Yang, "GPU-based NFA implementation for memory efficient high speed regular expression matching", 2012.