

# 8.3 Morphic Architecture Grp 56 & Grp 63

*by Chirag Modi*

---

**Submission date:** 14-Nov-2019 12:42PM (UTC+0530)

**Submission ID:** 1212259570

**File name:** NEUROMORPHIC\_ARCHITECTURES\_2.pdf (1.31M)

**Word count:** 5199

**Character count:** 26972

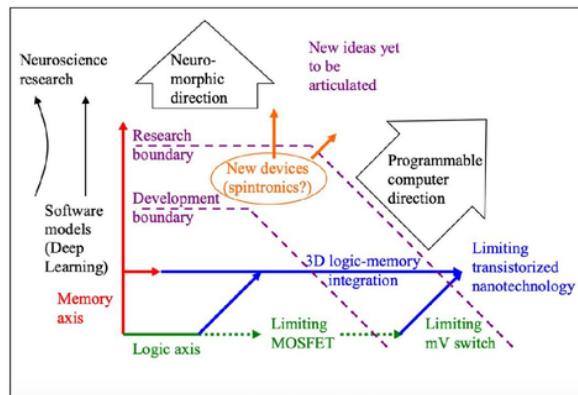
## **Group 63**

### **Topic 8.3 Morphic Architecture**

| <b>Name</b>     | <b>ID</b> | <b>Email</b>               | <b>Contact</b>  |
|-----------------|-----------|----------------------------|-----------------|
| Chirag N Modi   | 201601165 | mchirag1247@gmail.com      | +91 8320360696  |
| Bhargav Diwakar | 201601037 | bhargavdivakar10@gmail.com | +91 7203939741  |
| Vishal Dasa     | 201601168 | dasavishal111@gmail.com    | +91 97271 95890 |
| Ravi Tiwari     | 201501153 | 201501153@daiict.ac.in     | +91 8169080148  |
| Srikumar Sastry | 201601016 | 201601016@daiict.ac.in     | +91 9925268098  |
| Vedant Gupta    | 201601145 | 201601145@daiict.ac.in     | +91 9426478484  |

### **8.3.1 INTRODUCTION**

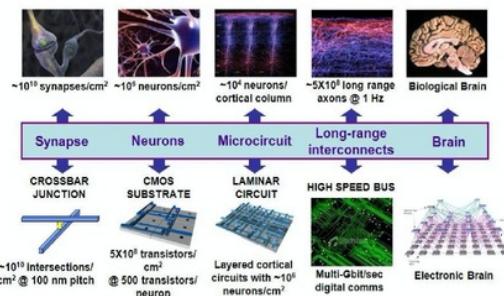
Moore's Law is the commentary made in 1965 with the aid of Gordon Moore, co-founder of Intel, that the two variety two of two transistors on a microchip doubles each two years, although the cost of computer systems is halved. But these days professionals are claiming that the trend is slowing down. It is becoming difficult to double the variety of transistors in the time body of two years while additionally being affordable. So to strength the next wave of electronics we want to seem at the different promising selections that work. Few of them include Quantum Computing, Neuromorphic Computing, Wave Computation, Spin interconnect, etc. In this chapter we talk about two morphic architectures and its future in electronics industry. Before moving any in addition we understand about morphic architectures and what are the different kinds of it.



**Fig. 8.3.1.1: The figure shows the roadmap of electronics after the end of Moore's Law.**

Morphic word is used in electronics to refer to circuits or devices which can adapt to a given problem and handle it efficiently. These architectures are inspired from biological structures or some scientific computational models. This was first introduced in The Emerging Research Architecture section of ITRS 2007, to refer to biologically stimulated architectures that embody a new sort of computation paradigm in which adaptation performs a key function to successfully tackle the particulars of problems. Morphic architectures include a broad type of mixed-signal structures that are centered on a specific application and that draw proposal for their structure from the application. In some cases, processing is carried out in the analog domain, providing orders of magnitude enhancement in overall performance and strength dissipation, albeit with reduced accuracy. As an example, biologically inspired inference networks for cognition may also yield to a partial analog implementation and grant good sized positive factors in performance relative to their digital counterparts.

In this chapter we will be further elaborating on the features of morphic architectures. We will talk about three different types of morphic architectures namely 1) Neuromorphic Architecture; 2) Cellular Automata; and 3) Cortical Architecture. We will look at the visualization and implementation of these architectures through different circuits. We will also learn about the current advancements of such technologies and lastly we will learn about where these architectures will be heading in the future.

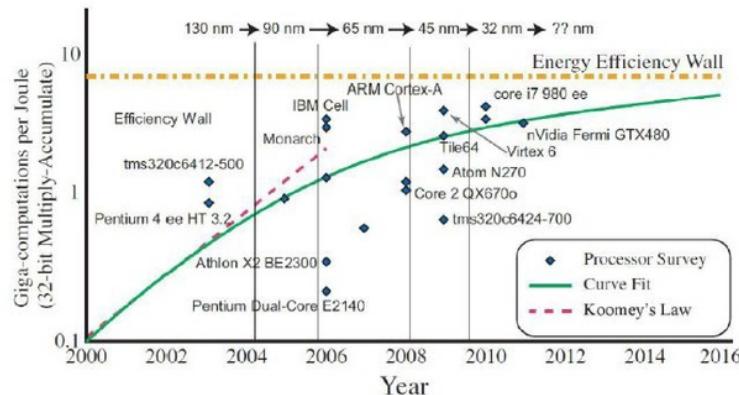


**Fig. 8.3.1.2: The figure shows the correspondence between biological architectures and hardware architectures.**

### 2 8.3.2 NEUROMORPHIC ARCHITECTURES

Neuromorphic Architectures was first introduced by Carver Mead in 1989. They are also known as neuromorphic processing or neuromorphic engineering. These hardware systems tend to use the basic neural computation function as their primary operation. Their objective is to use concepts just as weighted connections, long-term potentiation, non-linear activations and inhibitions from computational neuroscience to solve some of the real-world problems.

As compared to the existing solutions such as Von Neumann architectures, neuromorphic architectures exhibit low power operations and fault tolerant hardware solutions which are extremely useful in distributed and computation intensive tasks commonly seen in today's embedded hardware. Present architectures face the problem of scaling and power which is solved by neuromorphic computing.

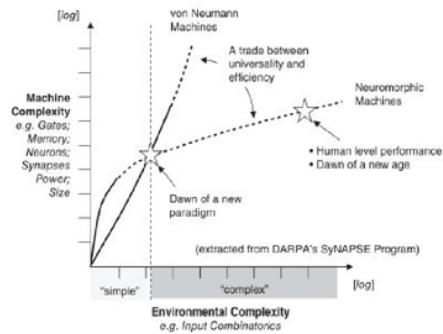


**Fig 8.3.2.1: Roadmap for achieving large neuromorphic hardware systems. [12]**

Neuromorphic designing is a combination of various disciplines such as physics, biology, mathematics and computer science to build artificial neural systems like object recognition system, autonomous rovers, audio processors which work on the principles of biological neural systems.

The features of neuromorphic computing include:

1. Exhibit human-like intelligence which can be specifically found in the neurons of humans.
2. Ability to handle anomalies, faults and detect noise to increase the performance of certain tasks.
3. They can be operated at a very low input power.



**Fig. 8.3.2.2 : Comparison between the design complexity of Von Neumann and neuromorphic architectures with increasing number of inputs. [1]**

The human brain is divided into areas (called cortex) which have specific function like visual processing, memory etc. Similarly, neuromorphic machines (LSI) are also application specific. Performance of existing Von Neumann can be increased by using these neuromorphic machines as supplemental CMOS. As a result of their variety of applications, they are referred to as more than Moore candidates. In real scenario, one of the main problems is to efficiently implement an artificial neuron. As the actual neuronal system in humans is too complex, the level of abstraction of a neuron is an important consideration in this regard. This implementation may be as simple as addition and subtraction of signals entering a neuron. There are existing technologies which can be used to represent a neuron like single electron transistor (SET), memristors, RTD etc. Depending on this technology, the potential and level of abstraction can be varied which open up new opportunities in the emerging nanoelectronics devices. Another problem is how signals are handled along the neuron and the synapses. In a biological neuron the computation occurs in a DIGITAL-ANALOG-DIGITAL fashion. Researchers have tried various techniques to develop analog synapses using flash memories but all suffer a limitation of designing the controller of electrons and limited rewriting. In contrast, memristors (resistive RAMs and atomic switches) offer a better solution for implementing such synapses. CMOS neurons are often used with memristive nano-junction and additional controllers to constitute CMOL architectures. CMOL (Cross Nets) was introduced in ITRS 2007 to refer to structures where nanogrids of single molecule is fabricated on traditional CMOS. These architectures pave the way for development of large scale neural networks which have many layers. [1]

A last significant issue is to incorporate the noise tolerance property of artificial neural networks into electronics. The most common technique to deal with noise in analog and digital systems is suppression. Neural networks on the other hand use this noise to improve their performance. Some noise sensitive devices may use this concept to design computation models.

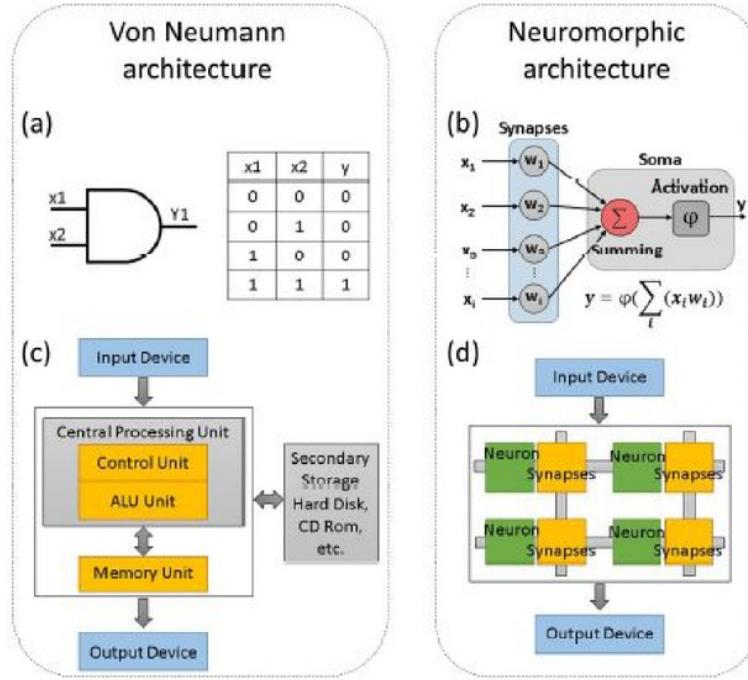
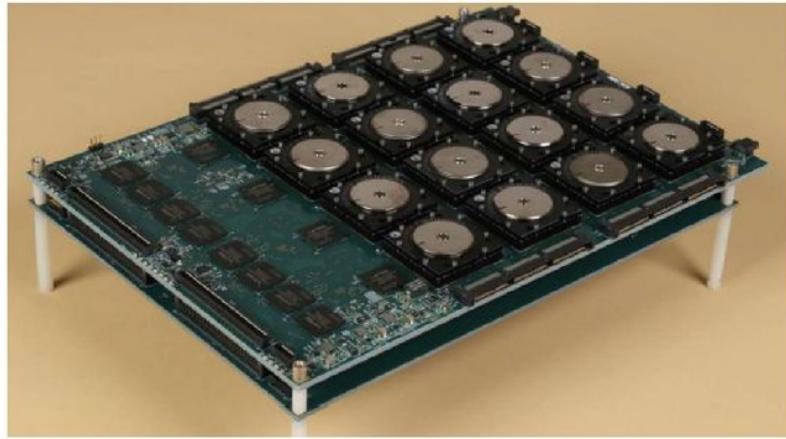


Fig 8.3.2.3 : This figure shows the contrast between Neuromorphic and Von Neumann architectures. (1) Traditional AND gate with its logic table. (2) Schematic of Von Neumann. (3) Basic computation model of an artificial neuron. (4) Schematic of neuromorphic architecture. [10]

### 8.3.2.2 EXAMPLES OF NEUROMORPHIC ARCHITECTURES

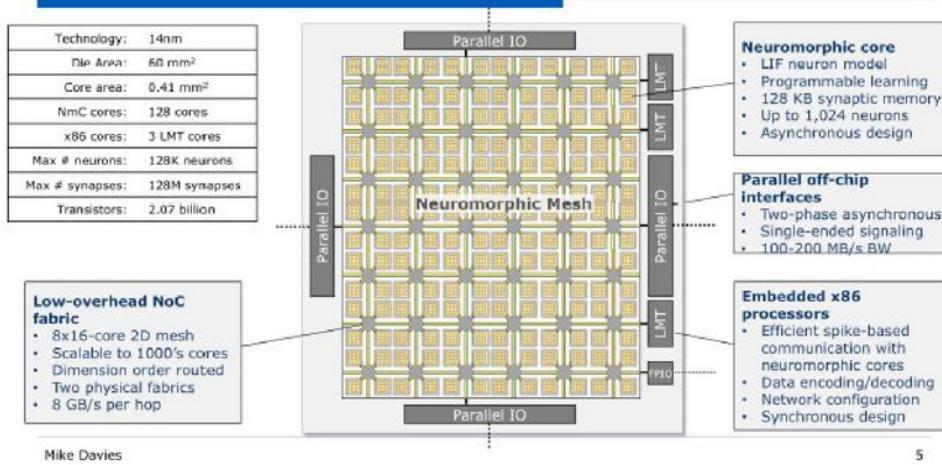
TrueNorth is a neuromorphic CMOS incorporated circuit created by IBM in 2014. It's a manycore processor prepared on a chip structure, with 4096 cores, each having 256 programmable derived neurons for associate degree completeness of somewhat quite 1,000,000 neurons. On these lines, every cell has 256 programmable "synapses" that pass away the sign between them. during this means, the arduous and quick variety of programmable synapses is somewhat 268 million. Its basic semiconductor count is five billion. Since memory, figuring, and correspondence area unit restricted in every of the 4096 neurosynaptic cores, TrueNorth goes round the von-Neumann-plan bottleneck and is extremely imperativeness capable, with IBM declaring an influence use of 70mW and an influence thickness that's 1/10,000th of typical microchips. The junction chip works at lower temperatures and power since it simply attracts management important for estimation.



**Fig. 8.3.2.4: DARPA SyNAPSE board with 16 TrueNorth chips. [11]**

Loihi is a neuromorphic chip planned by Intel Labs that uses associate degree unconventional spiking neural system (SNN) to actualize versatile self-changing occasion driven fine-grained parallel calculations accustomed execute learning and induction with high effectiveness. The chip is a 128-neuromorphic cores many-core IC factory-made on Intel's fourteen nm procedure and highlights a noteworthy programmable computer code learning motor for on-chip SNN making ready. The chip was formally exhibited at the 2018 Neuro galvanized machine components (NICE) workshop in American state. [2]

## Loihi Chip Architecture: Fine-Grained Mesh



Mike Davies

5

**Fig 8.3.2.5 : Architectural overview of Intel's "Loihi" neuromorphic computing chip.**  
[3]

### **8.3.3 Quantum Dot Cellular Automata Architecture**

8

In the 1940s, Stanislaw Ulam and John von Neumann introduced **Cellular** automation. Neumann worked on self-replicating system at that time, initially his model was based on the notion of robots building other robots but this idea failed as it was very difficult and expensive to build. To tackle this problem a suggestion regarding using a discrete system for creating a self-replicating model was made by Ulam. Ulam and Neumann in the 1950's worked together on the problem of calculating liquid motion. They used Ulam's suggestion of using discrete system for creating a self-reproducing model and build the first ever cellular automata model. The definition and concept behind cellular automata used by Neumann and Ulam are explained in the next paragraph.

A discrete model studied in complexity science, physics, mathematics, computer science and many other growing fields. Cells are the basic building blocks of a cellular automation, **regular grid** of cells is what that comprise a cellular automation. Each cell in the cellular automation is **in a state** selected from some finite set of predefined **states**. On and off are examples of such states. The dimension of grid is finite and the size can be in any finite number. In cellular automata an initial state is defined at  $t=0$  and these states are updated by advancing  $t$  by 1 and by using a mathematical function that works on the neighbourhood of the cell, which is established compared to itself

#### **8.3.3.1 Why Cellular Automata?**

The CMOS technology is failing many aspects and its use has been reduced because of the following reasons, firstly it has high power consumption, secondly it has low speed and density more than 10nm is not possible using CMOS, finally it has a lot of scaling and feature sizing issues. In response to these problems many researches were done and many solutions popped up. The most widely researched solutions are carbon nanotubes (CNT), quantum-dot cellular automata (QCA) and **single electron transistors** (SET). Out of all these most of the researchers believe that QCA is the most ideal option to replace CMOS. There are many advantages of QCA, it is transistor less, can achieve THZ operating speeds and device density of  $10^{12}$  devices/cm<sup>2</sup>. These advantages make QCA very useful for devices that require high performance or low power.

QCA was first proposed in 1994, it is different from other old computational techniques where information is transferred using electric current. In QCA the binary information is transferred from one place to another by using coulombic force of interaction between QCA cells. Today due to the requirements of the world circuits need to be fabricated of smaller and smaller size but due to the limitations of conventional technologies like CMOS VLSI these technologies become difficult to surmount and hence we think of using QCA for such circuits.

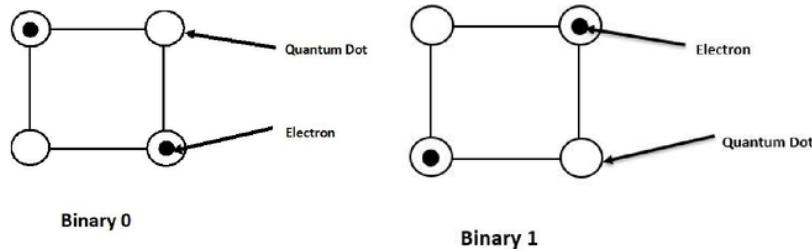
The functioning of QCA depends upon the charge configuration in the quantum dots of QCA. The binary information is encoded by this charge configuration and it is propagated in the circuit by coulombic forces of interaction between the QCA cells. In QCA's circuit one advantage is that no power supply is needed for individual cell and no current flows between the cells.

### 8.3.3.2 Basics of QCA Device

The arrangement of the quantum dots are such that they form a square. These quantum dots have a very small diameter which makes their charging energy greater than  $k_B T$ . In these cells two mobile electrons are put-in, these electrons can travel between those four quantum dots by electron tunnelling. This type of movement is represented through a path in the figure below. Due to coulombic repulsion the electrons can only take the opposing corners of the cells, thus there are only two types of orientation possible, with each orientation we associate a polarization as shown in the figure below. The figure shows the maximum distance that the two electrons can acquire between them without escaping the cell.

It is thought that electron tunnelling can be totally controlled by potential barriers placed underneath the cells. Capacitive plates can be used to control this potential barrier between QCA cells.

The figure of QCA cell is below:



**Fig 8.3.3.1: QCA cell polarization and representation of binary 1 and 0**

In QCA an isolated cell has two polarization state as explained above. These states are represented as  $P = +1$ , part (a) of the figure above and  $P = -1$ , part (b) of the above figure. These polarizations store the binary data. The polarization state  $P = +1$  can be characterized as binary 1 whereas the polarization  $P = -1$  can be characterized as binary 0. The polarization i.e.  $P$  of a QCA cell can be calculated using the formula:

$$P = \frac{(\rho_1 + \rho_3) - (\rho_2 + \rho_4)}{\rho_1 + \rho_2 + \rho_3 + \rho_4}$$

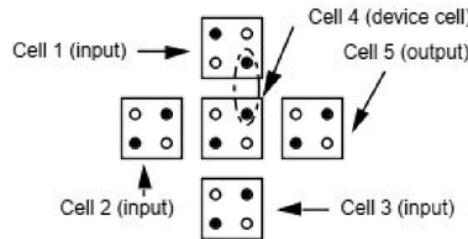
here  $\rho_1$  is the right top quantum dot and the numbering is done in clockwise order from that dot. Aside from these polarized states there is also an unpolarized state for a QCA cell. In this state the potential barrier between the quantum dots is lowered which causes the electrons to occupy a single quantum dot. In such a state the cell exhibits a small amount or no polarization at all.

### 8.3.3.3 Basic Logic Devices Using QCA

Next in our discussion in this topic is the designing of basic logic gates using QCA. The fundamental QCA logic is the wire, the majority gate and the inverter. We will in our discussion first discuss the fundamental building blocks and then some examples of logic gates built using QCA.

#### **8.3.3.4 The Majority Gate**

The majority gate is the basic building block in the QCA. It is nearly used in all the circuits designed by QCA. It would be beneficial for us to discuss this logic gate in detail while learning about QCA logic devices. This is a three input gate as shown in the figure below. The cell 4 in the figure is called the device or the driver cell. The majority gate works by taking the device cell to its lowest energy state. The lowest energy state is attained by the driver cell when its polarization is equal to the majority gate of three input cells polarization. The input cell is defined as a cell whose polarization changes based on a signal travelling towards the driver cell. The reason why the driver cell has minimum energy when it assumes the majority polarization is that when it assumes the majority polarization the electronic repulsion between the driver cell and the input cell is minimum.



**Fig 8.3.3.2. Majority Gate**

The above figure is an example where the centre cell attains minimum energy while having the majority polarity i.e.  $P=+1$ . To understand this let us look into the coulombic interaction between all the input cells with the driver cell. In the figure if we only see the driver cell and cell 1 then the driver cell should change its polarity to minimize repulsion but to minimize repulsion with cell 2 & 3 the driver cell should have the same polarity. Since, majority of the input cells have polarization  $P = +1$  in this case the driver cell will have the same polarity as these cells to minimize repulsion with majority of the cells.

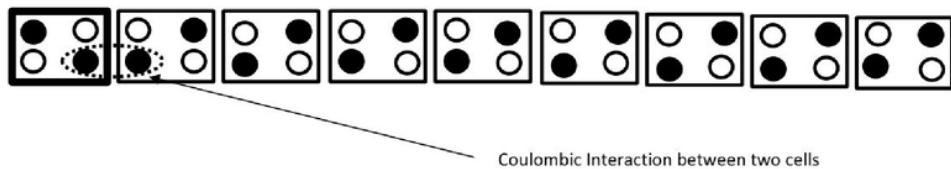
The output of this gate can be characterized by the equation:

$$\text{out} = a.b + b.c + c.a$$

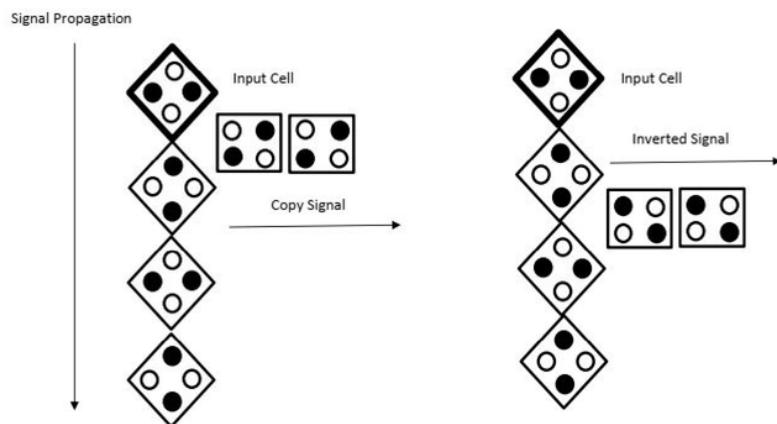
#### **8.3.3.5 A “90-Degree” QCA Wire**

The figure below is of a “90-degree” straight QCA wire and shows how data propagates in these wires. A wire in QCA terms is a horizontal sequence of QCA cells. In the figure binary information passed to the wire transmits from left to right without changing because of coulombic interactions

between the cells. In the first part of the figure the first cell has a polarity of -1 and the second has a polarization of +1. Here we assume that the charges in the cell 1 are fixed i.e. they are trapped in the quantum dot they are occupying whereas the charges in other cells are not fixed or trapped they are free to move between the quantum dots using tunnelling. This assumption saves us from the danger of charges traveling in the opposite direction from which they came. Here the  $P = -1$  travels across the wire because of coulombic interaction. In this example the cell 1 changes the polarization of cell 2 and from this the charges in cell 2 become trapped which changes the polarization of cell 3 and so on. This process continues down throughout the length of the wire.



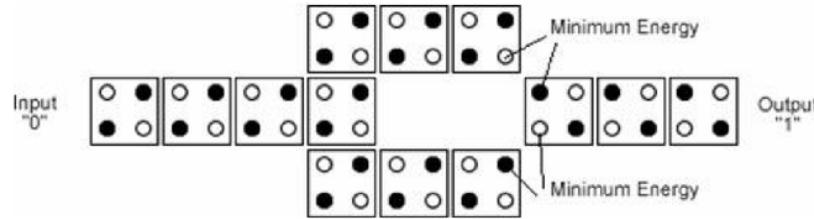
**Fig 8.3.3.3: QCA wire**



**Fig 8.3.3.4: Propagation of effect in QCA wire**

### 8.3.3.6 NOT GATE

Let us start our discussion of the inverter gate i.e. the NOT gate. There are various implementations through which this gate can be formed, we will discuss one such implementation here. NOT gate is used to invert the input voltage. NOT gate gives the output one on input zero and vice versa. The structure of NOT gate using QCA is shown below:



**Fig 8.3.3.5: NOT gate**

The Parallel wires contain the same polarization state as input but the structure of NOT gate invert the polarization state for output wire.

Now to discuss some examples, let us discuss the implementation of AND, OR using QCA. Firstly, let us look at the implementation of AND and OR gate. The logical function is represented for majority gate below:

$$Z = PQ + QR + PR \quad (2.2)$$

By setting the any of the input cell to the zero, majority gate works like AND gate.

$$\text{AND} = PQ + Q(0) + P(0) = PQ \quad (2.3)$$

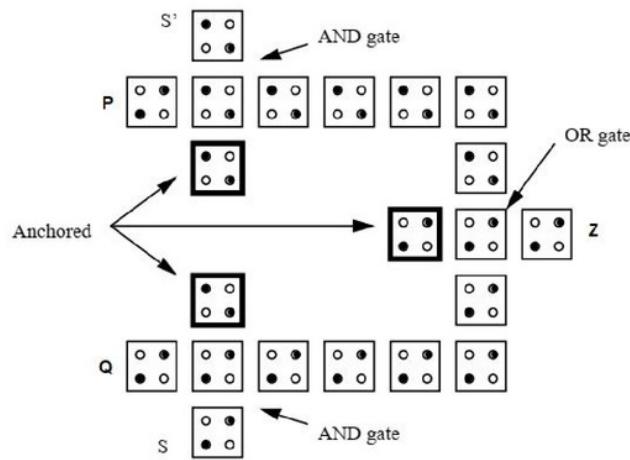
And OR function can be obtained by setting any input cell to one.

$$\text{OR} = PQ + Q(1) + P(1) = P + Q \quad (2.4)$$

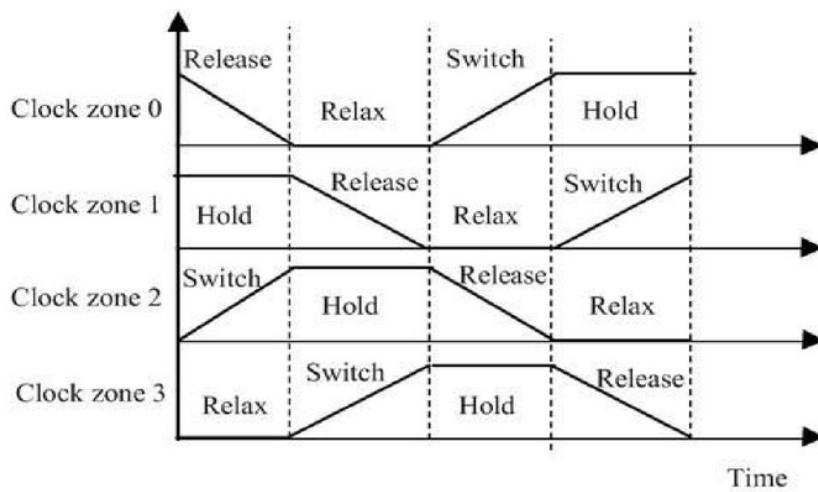
using the QCA cells AND, OR and NOT functionality can be implemented. So the QCA has complete functionality logic set. Using QCA cells any Boolean circuit can be implemented. To discuss one such circuit we have the implementation of 2x1 multiplexer in the figure 8.3.2.6.

### 8.3.3.7 QCA CLOCK

As the traditional circuits have two phases in the clock circuit built using the QCA has four phases. The clock can be given to each QCA cells but the QCA cells are divided into subarrays so that advantage of multi-phase and pipelining can be taken. In each QCA cell of same subarray hold the same clock phase that changes the polarization state. Clocking is very necessary in QCA to ensure that the information is transmitted correctly from the input cell to the output cell. In QCA the clock is divided into 4 different periodic phases as Switch, Release, Relax, and Hold as shown in the figure below.

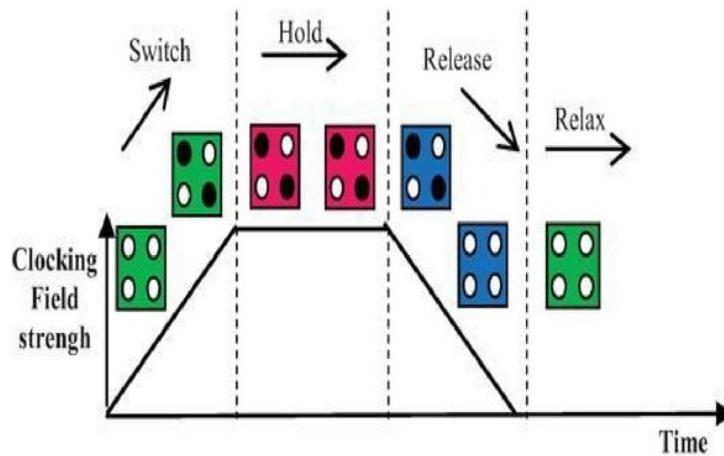


**Fig 8.3.3.6: 2x1 QCA multiplexor with logical equation  $Z = PS' + QS$**



**Fig 8.3.3.7: Clocking [4]**

4 In the switch phase of the clock, QCA cell is in non-polarization state and changes to polarization state. During this phase the potential barrier is raised to high. During the hold phase, polarization state doesn't change. It remains the same as previous phase and doesn't change. The potential barrier remains at high. In the release phase again, the transition happens. The potential barrier is lowered so the cells lose their polarization state and becomes nonpolarized. In the relax phase potential, Barrier doesn't change from the previous phase. So cells remain in nonpolarized states. Then again, the switch phase takes place. Figure 8.3.2.8 shows the polarization state as well as potential barriers of QCA cells during each phase.



**Fig 8.3.3.8: Stages Of Operation [4]**

### 8.3.3.8 QCA IMPLEMENTATION

There are couple of ways are proposed to implement QCA.

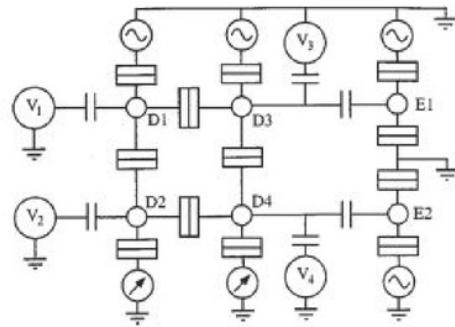
#### 1) Metal QCA

Metal QCA consists of four aluminium islands which are connected by aluminium oxide tunnel connections and <sup>3</sup>capacitors. The island capacitance can be determined by the area of the tunnel junctions. This device has been successfully built using the Electron Beam Lithography. Figure 8.3.2.9 shows a simple schematic of Metal QCA. Behavior of the Metal QCA is tested by the experiment that match to the QCA device. Sequential and Logic circuits are also built using the Metal QCA as a building block.

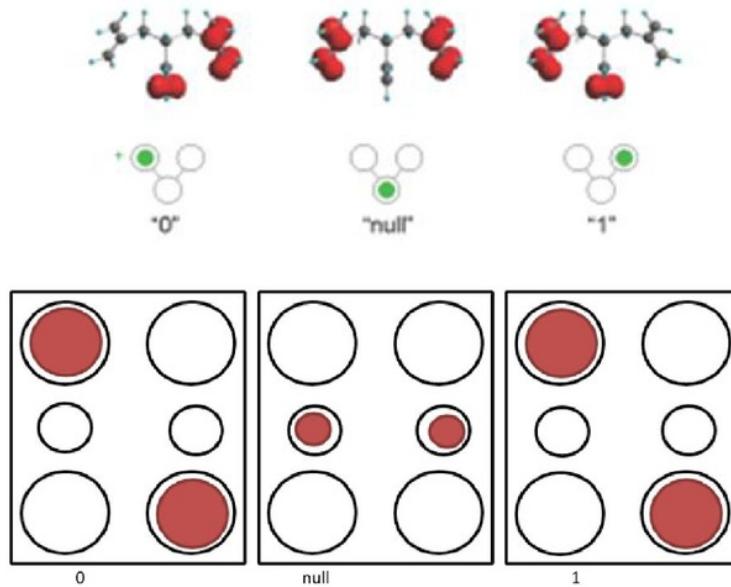
Here aluminum dots are D1-D4 and E1-E2 sense the output.

#### 2) Molecular QCA

<sup>3</sup>In Molecular QCA, each molecule acts as a QCA cell in which redox centers acts as dots and tunneling is provided by bridging ligands. Molecular QCA operates at higher frequency and at room temperature. Molecular QCA also have higher device density.



**Fig 8.3.3.9 : Schematic diagram of four dot metal QCA cell [5]**



**Fig 8.3.3.10: Schematic diagram of six dot QCA cell [6]**

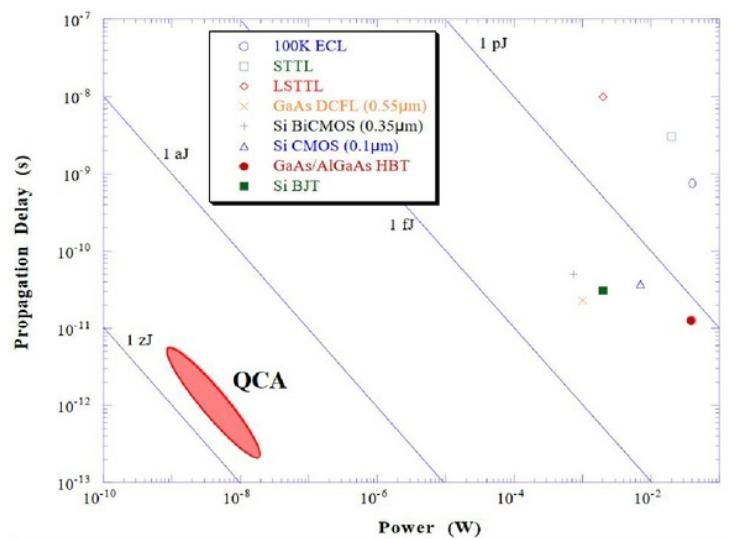
### 3) Magnetic QCA

Interaction between nanoparticles guarantee that structure is always bistable. So, information can be transferred through these nanoparticles. But this doesn't have much switching speed compared to the current computers.

### 8.3.3.9 Limitations Of QCA

**Table 8.3.3.1: Projection of QCA in various benchmarks**

| Benchmark              | 2006                     | 2012      |
|------------------------|--------------------------|-----------|
| Feature size           | 2 nm                     | 10 nm     |
| No. of devices         | 4                        | $10^6$    |
| Circuit speed          | 10 kHz                   | 100 GHz   |
| Events / chip / s      | $4 \cdot 10^4$           | $10^{14}$ |
| Power supply, $V_{dd}$ | 0.1 mV                   | 0.1 mV    |
| Power dissipation      | 1 pW (excluding cooling) | n/A       |
| Temperature            | 4 K                      | 4 K       |



**Fig 8.3.3.11: Power vs Delay graph for QCA and other technologies**

The fabrication of QCA is expensive. There are very few techniques available for lithography. Perfect alignment of cells is also required otherwise results in defects and this alignment is very difficult to achieve. Circuit to work at room temperature cell dimensions should be order of nanometer.

In the figure above shows that for devices other than QCA for low propagation delay we require large amount of power, but for QCA low propagation delay can be achieved by low power , which is one of the biggest advantages of QCA. This is one of the biggest reasons why QCA is loved by researchers and is the hottest technology to replace CMOS.

### 8.3.4 CORTICAL ARCHITECTURES

Cortical architecture based circuits are one approach through which modelling of brain has been attempted. In Cortical architecture the modelling of circuits is done in comparison to the Neocortex of

the human brain. Neocortex is responsible for higher order functions in humans such as cognition, generation of motor commands, language, sensory perception, etc.

Our brain uses electrochemical impulses, or spikes, between neurons to transfer messages within itself and the human body. These spikes hold information in them and complex computation relies on how spikes communicate in highly-connected networks. The researchers hope to be able to see the performance of more biological neurons in real time by modeling machines on the architecture of the brain. A full-scale supercomputer based on cortical architecture will help us understand even larger networks which were previously thought to be out of reach. This will help in understanding the healthy and unhealthy functioning of the brain.

Cortical Architecture is a very new field and hence a lot of research is going. Researchers as well as big companies have all attempted to make circuits or products that come close to the efficiency of the neocortex. They have applications in Digital Signal processing(DSP), Machine Learning, Computer Vision, etc. In this section we will be talking further on these examples and try to understand cortical architecture through few applications.

#### **8.3.4.1 APPLICATION: SpiNNaker**

SpiNNaker or the Spiking Neural Network Architecture is the largest supercomputer in the world which is designed to work in a manner similar to the human brain. It is a many core supercomputer designed at the School of Computer Science, University of Manchester. On October 14, 2018 the Human Brain Project(HBP) announced that the million core milestone had been achieved.



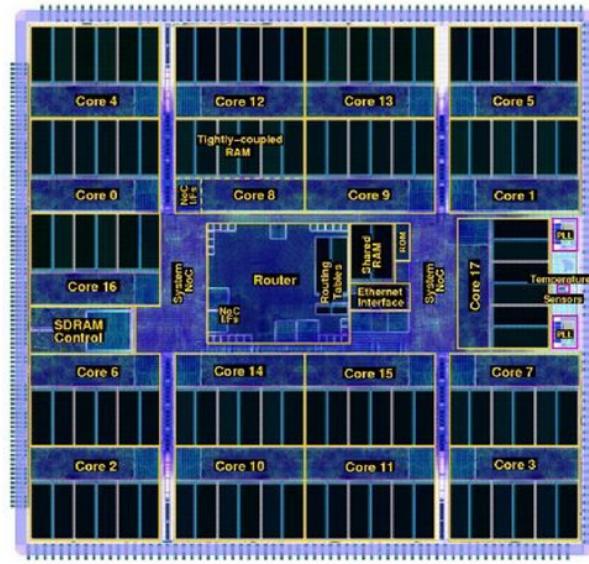
**Fig. 8.3.4.1 The SpiNNaker 1 million core machine assembled at the University of Manchester.**

[7]

Each of the chips used in SpiNNaker machine consists of 100 million transistors and can handle actions which are in order of millions of million per second. It replicates the human brain biological neuron. The network of the chips used inside this machine match the neocortex of the brain. This machine is better than any other machine and capable of modelling huge amount of biological neurons. Computer systems that are large scale such as the SpiNNaker contain electronic circuits which are used to mimic the electrochemical spikes which are a characteristic of Neuromorphic computing.

SpiNNaker is unique in many ways than the traditional computers because, unlike traditional computers, huge amounts of information is not sent from one point to another through a network which is standard. In Fact it shows similarity with architecture of the brain which supports parallel communication, which simultaneously sends small amounts of information which are in order of billions to thousands of different destinations.

Real-time high-level processing in a cluster of isolated brain networks has been simulated using SpiNNaker. One of it is an 80,000 neuro model of a sub-part of the cortex. The cortex receives and processes information from the senses. The human brain is 1,000 times bigger than the brain of a mouse and that itself consists of more than 100 million neurons. This given an idea about the scale of information it has to process. The fundamental use of the supercomputer is helping neuroscientists better understand the working of human brain.



**Fig. 8.3.3.2 The architecture of the 18-core SpiNNaker chips(Cortical Architecture) [8]**

There are various regions in which the SpiNNaker machine can be helpful and newer fields keep coming up. Applications of this machine include simulation of Basal Ganglia, a region of our brain which is an area vulnerable to Parkinson's disease. This shows that massive neurological breakthrough is possible in pharmaceutical testing and other sciences. SpiNNaker has also been recently used to control a robot called the SpOmnibot. The robot interprets visual information which is real-time and navigates towards certain objects while ignoring others using functionalities provided by SpiNNaker.

### **DARPA UPSIDE/Cortical processor study**

The Defence Advanced Research Projects Agency(DARPA) started building on Bio-Inspired Algorithms for Machine Learning. Their main application was on Digital Image Processing using Deep Learning principles. Their work consists of improvements on two parts, one on the software side

and the other on the hardware side. Through their approach, DARPA was successful in exploiting the physics of emerging devices to perform extremely fast and low power computation. Preprocessing, Segmentation, Tracking and Classification of images was achieved at 100 times better rate than conventional approaches available. The power efficiency was also much better. The hardware was 1000 times more power efficient than normal systems.

For sophisticated Department of Defense(DoD) related applications DARPA improved on Unconventional Processing Signals for Data Exploitation(UPSIDE). UPSIDE generally consists of three tasks. UPSIDE Task 1 consists of Image Processing Pipeline preparation and Inference Module development. UPSIDE Task 2 consists of implementation using mixed CMOS of computational model. UPSIDE Task 3 consists demonstrating Image processing using next-generation devices with proper computer modelling.

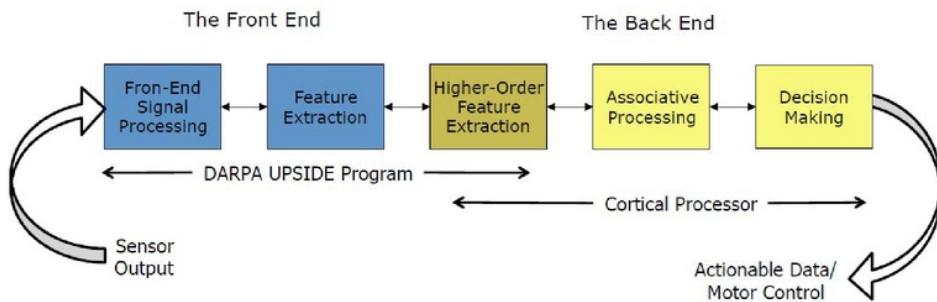


Fig 8.3.3.3. Block diagram for Image Processing using DARPA UPSIDE Program [9]

### DARPA Cortical Processor Concept

Since the Algorithms used by DARPA are Bio-inspired, the hardware has to be designed in a manner which closely resembles the human body. More specifically, the cortex of the human brain. It is difficult for us to mimic brain completely but we know enough to abstract biological principles. This helps us to modify ML algorithms according to our need. They are trying to create a computer architecture with suitable algorithms that mimic specific characteristics of human brain functions, including learning and pattern recognition and addressing data recognition control and its challenges. This will help them understand complexities for DoD systems. Today, the most common approach is to use algorithms which are hand-crafted and application specific. But they require long computing time and high precision which limits their ability to learn large datasets rapidly and do not work well for real-time applications.

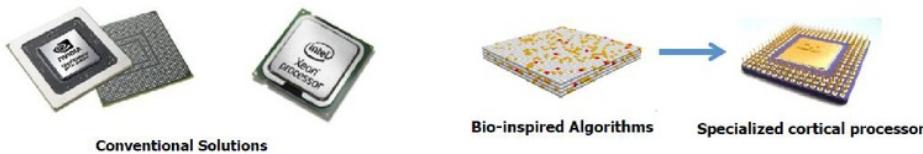


**Fig 8.3.3.4. DARPA Cortical Processor Concept [9]**

The new approach is to use neural algorithms inspired by brain that use a low-precision, temporal and hierarchical memory structure. These algorithms will evolve with changing data and requires lesser training times. But using only better algorithms is not enough, the hardware has to be improved as well to meet the requirements. As seen in Fig 8.3.3.4 that a cortical processor is required for continuous real time learning. High performance and lower power algorithms can be created using optimized silicon architectures for real time applications. Boost is seen in embedded system operations and large scale applications as well.

### Mapping Bio-Inspired Algorithms to Hardware

We know that Bio-inspired learning algorithms require matched hardware. Higher-connectivity, better local memory and parameter storage and simple and low precision computation are some of their advantages. They are configurable as well.



**Fig 8.3.3.5. a. Conventional processor vs b. Bio-inspired Cortical processor [9]**

Conventional processors are not good for cortical algorithms. They provide limited parallelism and have constrained processor/memory partition. They consist of complex instruction sets which make it difficult for us to build good programs on them. On the other hand custom architectures suit well for bio-inspired approach. For neuro architecture and computation they use optimized version of conventional CMOS fabrication. They also eliminate the need to use high risk components.

### Cortical Processor Hardware Possibilities



**Fig 8.3.3.6. Object Detection Experiment [9]**

DARPA testing shows that the efficiency of cortical architecture can be seen greatly in object detection. One such example is shown in Fig. 8.3.3.6 in which the object used for detection is motorcycle. Compared to the traditional deep learning approach, a bio-inspired algorithm when applied on cortical architecture is much better. As we know that Moore's law is nearing its end. It is

getting difficult to get more transistors with each geometry shrink. But what else has been observed is that Dennard scaling has stopped. This means that voltage decreases have stalled even as feature sizes shrink[8]. DARPA UPSIDE study has proven to be an exception.



**Fig 8.3.3.7. a.Deep Learning approach vs b.Cortical processor approach [9]**

The deep learning approach faces several challenges. Image 8.3.3.7.a shows that it does not train on real-time data, extensive offline training is required. For inserting a new object, we would have to re-train the entire network. Even for the best hardware mapping in GPU, in which all nodes are active, we can process only 2.3 images per second. And processing every image requires 212 Joules per image, which is quite high.

The cortical performs much better. It runs in real time and exhibits online on the go adaptation and learning. As the Fig. 8.3.3.7.b depicts, there is no problem while adding new objects, they can be added continuously. The classification result comes out to be fixed point single precision and all nodes do not need to be active all the time. Hence it saves a lot of power as well. We can process 1000 images per second and each image processing takes just 0.0004 Joules per image. This is significantly lower than deep learning approach. Thus in a way DARPA has overcome the challenges of Moore's law and Dennard scaling.

## 8.3 Morphic Architecture Grp 56 & Grp 63

### ORIGINALITY REPORT

|                  |                  |              |                |
|------------------|------------------|--------------|----------------|
| <b>5</b> %       | <b>1</b> %       | <b>4</b> %   | %              |
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

### PRIMARY SOURCES

- 1** [www.itrs.net](http://www.itrs.net) **1**%  
Internet Source
- 2** Asai, Tetsuya, and Ferdinand Peper. "Explorations in Morphic Architectures", Emerging Nanoelectronic Devices, 2014. **1**%  
Publication
- 3** Razieh Farazkish, Fatemeh Khodaparast. "Design and characterization of a new fault-tolerant full-adder for quantum-dot cellular automata", Microprocessors and Microsystems, 2015. **<1**%  
Publication
- 4** Radhouane Laajimi. "Chapter 3 Nanoarchitecture of Quantum-Dot Cellular Automata (QCA) Using Small Area for Digital Circuits", IntechOpen, 2018. **<1**%  
Publication
- 5** Dan Hammerstrom. "DARPA neurocomputing", 2015 IEEE International Electron Devices Meeting (IEDM), 2015. **<1**%  
Publication

---

6

Hemant Balijepalli, Mohammed Niamat. "Design of a nanoscale Quantum-dot Cellular Automata Configurable Logic Block for FPGAs", 2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS), 2012

<1 %

Publication

---

7

C.S. Lent. "An architecture for molecular computing using quantum-dot cellular automata", 2003 Third IEEE Conference on Nanotechnology 2003 IEEE-NANO 2003 NANO-03, 2003

<1 %

Publication

---

8

M. Sipper. "The emergence of cellular computing", Computer, 1999

<1 %

Publication

---

9

engineering.nd.edu

<1 %

Internet Source

---

10

Computational Complexity, 2012.

<1 %

Publication

---

11

Lu, Yuhui, and Craig Lent. "Self-doping of molecular quantum-dot cellular automata: mixed valence zwitterions", Physical Chemistry Chemical Physics, 2011.

<1 %

Publication

---

Exclude quotes

Off

Exclude matches

< 5 words

Exclude bibliography

On