# A Novel Systolic Array Architecture for Matrix Multiplication Circuit Design using Carbon Nanotube Technology

Alireza Azimian
Department of Computer Engineering
Khatam University
Tehran, Iran

Ali Kargaran Dehkordi
Department of Computer Engineering
Khatam University
Tehran, Iran

Mohammad Tehrani
Department of Computer Engineering
Khatam University
Tehran, Iran

## ABSTRACT
recently, parallel computing has been considered increasingly and many researchers have focused on this topic in order to enhance their designs, especially speed parameter to reach lower delay in computational operations. Among the methods which use parallel computing, systolic arrays have attracted researcher's attention because of its unique characteristics. Systolic arrays are arrays of processors which are connected to a small number of nearest neighbors in a mesh-like topology. Processors perform a sequence of operations on data that flows between them. Generally the operations will be the same in each processor, with each processor performing an operation (or small number of operations) on a data item and them passing it on to its neighbor. Systolic arrays are often using for specific operations, such as "multiply and accumulate", to perform massively parallel integration, convolution, correlation, matrix multiplication or data sorting tasks. On the other hand, silicon limitations for transistors fabrication in future causes a need to substitute this technology by an appropriate ones that among them carbon nanotube (CNT) technology has the most probability. In this paper we conducted a survey on using systolic array in multiply and accumulate operations by a VLSI circuit based on CNT technology.

## Keywords
Systolic Array; Parallel, CNT, Matrix Multiplication, CMOS, Cell.

## 1. INTRODUCTION
Recently, enormous technological advances have been considered in the area of VLSI circuits. Such technological advances causes to appear a new way of computing, constituted of highly parallel computing systems for specific applications. Systolic array computing is an interesting approach, proposed originally by Kung and Leiserson [1], at the end of the seventies. A systolic array is a parallel computing device for a specific application that is conducted by a large number of simple processing elements called cells, interconnected with local communication in a usual way. In computer architecture, a systolic architecture is a pipelined network arrangement of Processing Elements (PEs). It is one of the forms of parallel computing, where cells compute the data which receive as input and store them in an independent way. A systolic architecture is an array constituted of matrix-like rows of cells. Here, the Processing Elements is similar to central processing units (CPUs).

Matrix multiplication plays an important role in numerical linear algebra. It is used for compute these products in several stages as well as in many technical problems such as digital signal processing, weather prediction, pattern recognition and etc. Therefore, develop an ideal algorithm for performing these computations is at the focus of many researchers. Matrix multiplication is a very common computation and parallel implementation is an appropriate way for implement them. Some structures, such as systolic arrays, are very suitable for matrix multiplication and are also possible to implement in VLSI circuits due to its simple and regular design.

To follow the Moore's law that states the number of transistors on integrated circuits (ICs) doubles almost every two years [2], the feature size of conventional silicon based MOSFETs must be scale down. CMOS technology has faced serious problems by scaling down of the feature size into the nanoranges, such as increase of current leakage, high power densities, decreased gate control, short-channel effects and high sensitivity to process variations [3]. For this reason some new technologies have been presented such as single-electron transistor (SET), quantum-dot cellular automata (QCA) [4], [5] and carbon nanotube field effect transistor (CNFET) [6-9]. To overcome mentioned problems, circuit designers have been focused on these technologies [9-14]. Among new technologies, CNFET is one of the most promising alternative technologies to the silicon-based technology due to its remarkable properties [15]..

## 2. CNFET'S SPECIFICATION AND MANUFACTURING
Silicon transistors are the main elements of integrated circuits. In Nanoelectronic technology. Carbon nanotubes are cylindrical carbon molecules with unique features that are used in applications such as Nanoelectronics, optoelectronics and other materials.

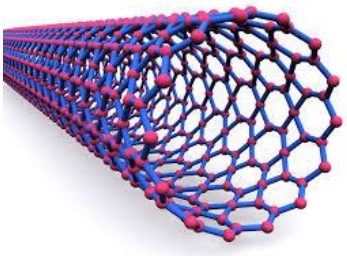

**Fig. 1. SWCNT**

Carbon nanotubes are discovered as a product of arc discharge test conducted to produce C60 in 1991. One of the nanotubes specification that effects their electric specifications is the number of walls. nanotubes' diameter is the main difference between single wall structure and multiple walls.

As shown in figure 2, carbon nanotube transistors are very similar to CMOS based transistors with a difference that in CNFET based technology, the channel's material is carbon nanotube instead of silicon in CMOS technology. Due to the SWCNT's band gap energy is approximately equal to semiconductor, a large number of CNFET based transistors are single-wall. Figure 2 presents a CNFET transistor based on top-gate and back-gate specifications.

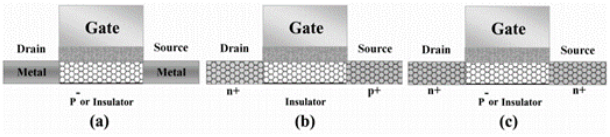


**Figure 2. Different types of CNFET device; (a) SB-CNFET, (b) TCNFET, (c) MOSFET-like CNFET.[15].**

Scalable threshold voltage is the major feature of the CNFETs that is proportional to the inverse of nanotube's diameter according to equation 1:

$$V_{th} = \frac{0.43}{D_{CNT}(nm)}(V) \qquad (1)$$

## 3. PROPOSED MODEL

Matrix multiplication plays a central role in numerical linear algebra, since one has to compute this product in several stages of almost all numerical algorithms, as well as in many technical problems, especially in the area of digital signal processing, pattern recognition, plasma physics, weather prediction, etc. Therefore, finding an efficient algorithm for performing these computations is at the focus of interest of many researchers. Matrix multiplication is a very regular computation and lends itself well to parallel implementation. Regular structures, such as systolic arrays (SAs), are well suited for matrix multiplication and are also amenable to VLSI implementation because of simple and regular design, and nearest-neighbor communications. A systolic system is a network of processing elements (PEs) that rhythmically compute and pass data through the system. Once a data item is brought from the memory, it can be used effectively in each PE as it passes while being "pumped" from cell to cell along the array.

In this section we are presented an algorithm for implementing matrix multiplication based on a VLSI circuit using carbon nanotube technology. In order to understand our provided method first take a look at the general block diagrams of a matrix multiplication by systolic array in figure 3.

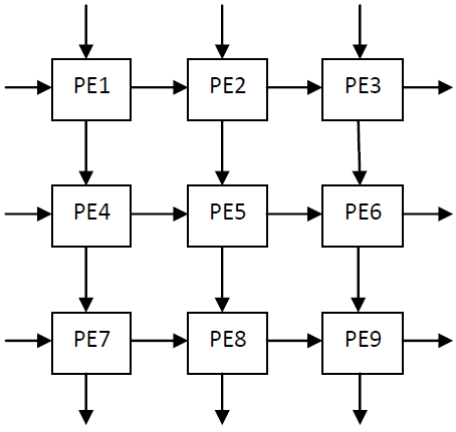


**Figure 3. Systolic Architecture for Matrix Multiplication**

As specified in figure each processing element (PEs) performs a specific multiplication operation. For example, in order to multiply first row and first column at first operation in two different matrices, one of the PEs does the task (PE1). For more understanding we present two square 3×3 matrices and we would perform the multiplication operation using these PEs. First of all, in order to multiply bit numbers we are required to design an appropriate VLSI circuit. As specified in figure 4 we have used this circuit to do the mentioned operation in a transistor level.

$$\begin{bmatrix} A_1 & A_3 & A_5 \\ B_1 & B_3 & B_5 \\ C_1 & C_3 & C_5 \end{bmatrix} \begin{bmatrix} A_2 & B_2 & C_2 \\ A_4 & B_4 & C_4 \\ A_6 & B_6 & C_6 \end{bmatrix}$$

For this purpose take a look at the following equation:

$$A_1A_2 + A_3A_4 + A_5A_6 = PE1\ Block's\ answer \qquad (2)$$

Finally, at the last step we have done all operations by all PEs in a parallel process. For instance, we have done a 2×2 matrix multiplication using systolic array architecture which explain step by step in the rest. First of all, take a look at the following 2×2 matrix:

$$\begin{bmatrix} A_{1(1,0)} & A_{2(1,0)} \\ A_{3(1,0)} & A_{4(1,0)} \end{bmatrix} \boxtimes \begin{bmatrix} B_{1(1,0)} & B_{2(1,0)} \\ B_{3(1,0)} & B_{4(1,0)} \end{bmatrix}$$

According to the previous description about systolic array and its performance to do matrix multiplication, we need for blocks as well as for clock pulse to implement all the operations. For more perception, we present the mentioned structure in figure 4.

Considering figure 4, 2×2 matrix multiplication has done as shown in figure 4.

Now, we want to provide a survey of each block's inner structure and explain that how they work to do their special task in every step. All of the blocks' structure and circuit are the same, for this reason, just the first block's structure is presented considering each input is a 2-bit binary number. Each block includes two serial registers (D flip-flop) for every input as well as the number of four registers for the total block's output, one 2-bit binary multiplier to do the multiply operation in each step, four serial Full-Adders to implement 4-bit adding operation and finally four and gate to implement a reset pulse in step 1 for each block. All of the circuits' clock are the same. Pay attention the total valid answer for the whole matrix multiplication is reached after 4 clock, but for the first block which has been done and simulated in this paper, answer is valid after two clock pulse as shown in figure 4. In order to understand more about our provided structure for each block look at the figure 5.

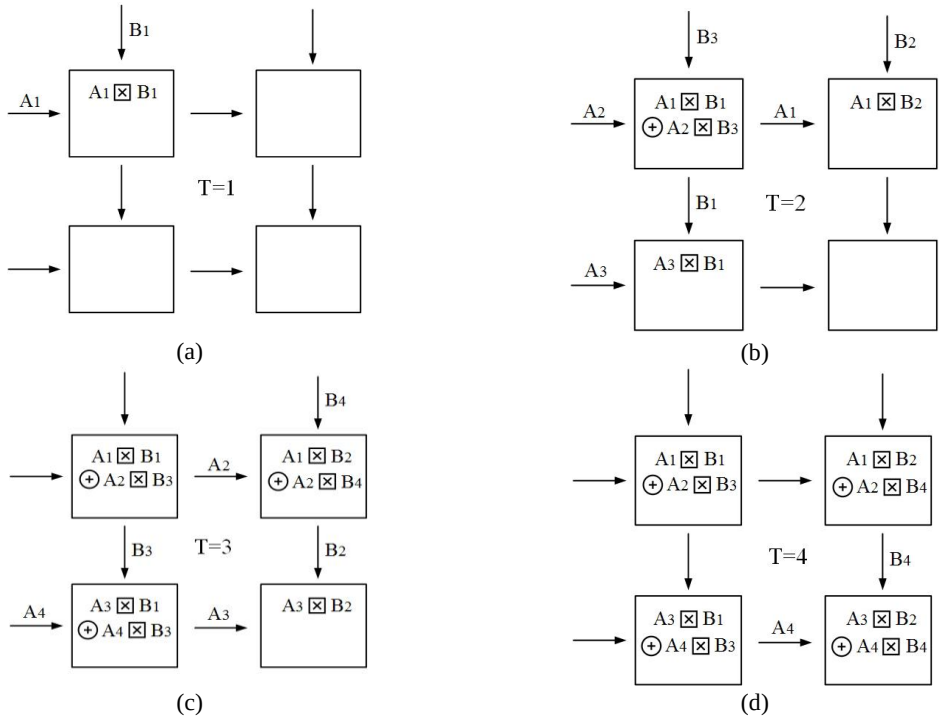


**Figure 4. Matrix multiplication steps using systolic array architecture. a) Step 1 in rising clock 1. b) Step 2 in rising clock 2. c) Step 3 in rising clock 3. d) Step 4 in rising clock 4.**

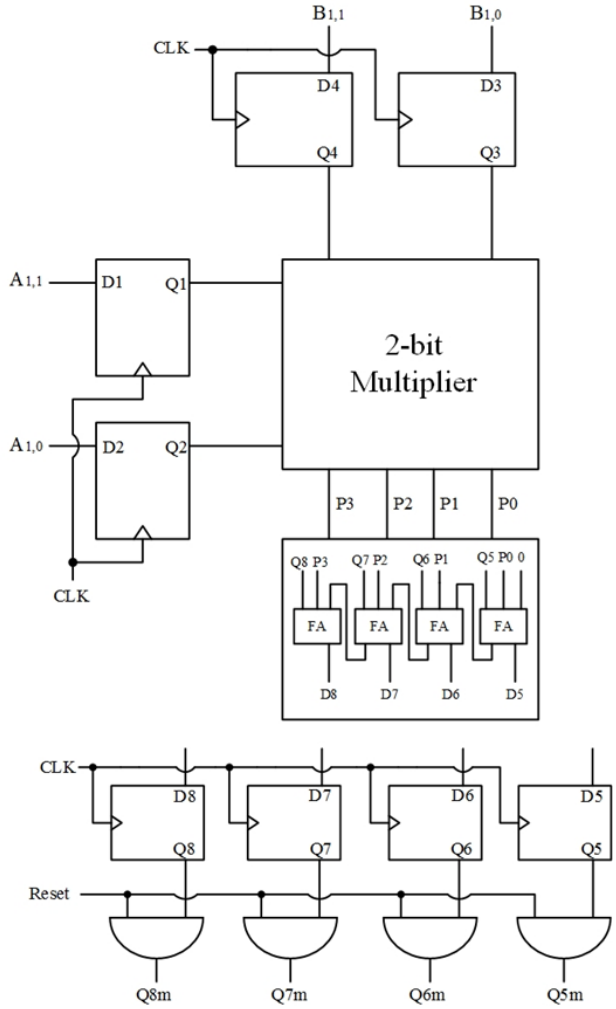


**Figure 5. Each block's structure to do multiply operation.**

Notice that all registers are rising edge triggered D flip-flops. In order to evaluate our provided design we have simulated the mentioned circuit by HSPICE simulator in 1V supply voltage and 25°C temperature. Simulated results is shown in table I.

**Table I. simulated results of the proposed circuit.**

| Circuit | Power (*e-05W) | Delay (*e-12S) | PDP (*e-16J) |
|---|---|---|---|
| Proposed CNT circuit | 8.1503 | 3.8792 | 3.1617 |

Transient analysis of the sample inputs is shown by figure 6. In every step we have applied a pair of 2-bit number to the proposed circuit in order to multiply them to reach the final output. In this example our inputs in the first block are shown in the following matrix:

$$\begin{bmatrix} 11 & 00 \\ A3 & A4 \end{bmatrix} \times \begin{bmatrix} 01 & B2 \\ 10 & B4 \end{bmatrix}$$

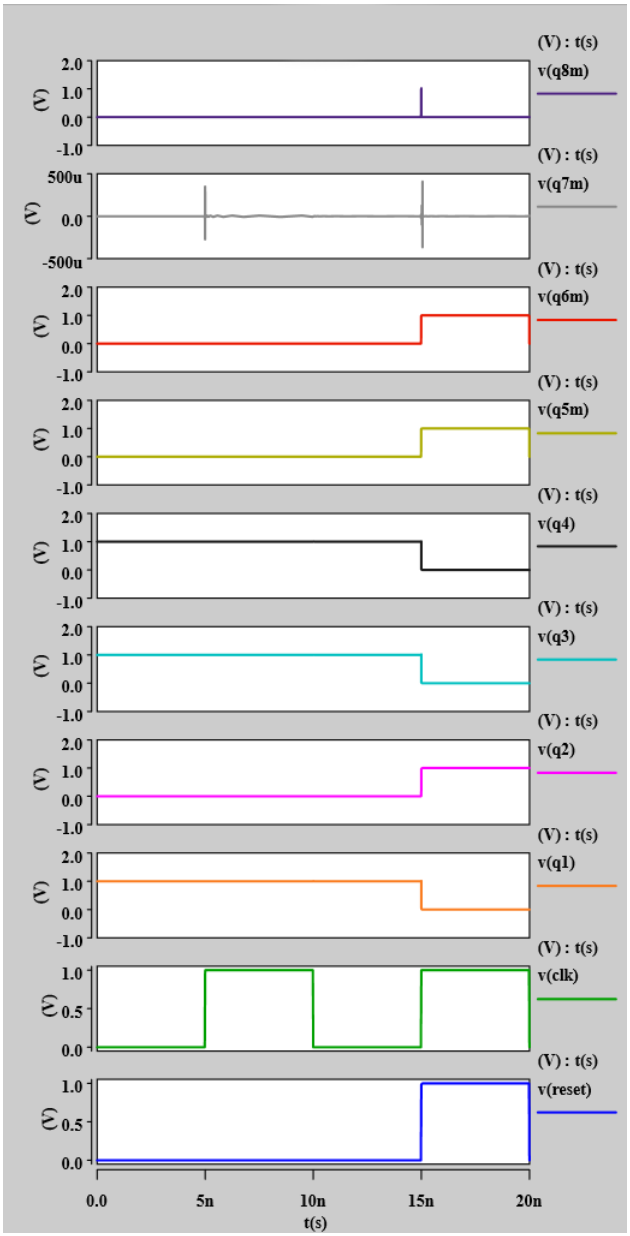


**Figure 6. Transient analysis of a sample 3×3 matrix.**

## 4. CONCLUTION

Systolic array is one of the best ways of perform massively parallel integration. By using this technology we can reach to the high speed and low cost parameters in order to implement many operations such as multiply and accumulate operations. On the other hand, using carbon nanotube technology in transistors channels instead of silicon materials would conclude to a high speed for VLSI circuits. Finally, combining these methods will result in an ultra-high speed parameter in order to use in applications which reach to a lower delay in computations have been considered.

## 5. REFERENCES

[1] Kung, H. T. and Leiserson, C. E. ”Systolic arrays for VLSI”, in:Introduction to VLSI Systems, C. A. Mead and L. A. Conway, Chapter 8.3, Addison-Wesley, 1980.

[2] Lin, S., Kim, Y.B., and Lombardi, F. (2009), ‘A Novel CNTFET based Ternary Logic Gate Design’, in Proceedings of the 52nd IEEE International Midwest Symposium on Circuits and Systems 2009, Cancun, Mexico, 2–5 August, pp. 435–438.

[3] Y.B. Kim, Challenges for nanoscale MOSFETs and emerging nanoelectronics, Trans. Electr. Electron. Mater. 11 (3) (2010) 93–105.

[4] Keikha, A., Dadkhah, C., Tehrani, M., & Navi, K. (2011). A novel design of a random generator circuit in QCA. International Journal of Computer Applications, 35(1).

[5] Navi, K., Tehrani, M. A., & Khatami, M. (2012). Well-polarized quantum-dot cellular automata inverters. International Journal of Computer Applications, 58(20).

[6] M.H. Moaiyeri, R. Faghih Mirzaee, A. Doostaregan, K. Navi, O. Hashemipour, A universal method for designing low-power carbon nanotube FET-based multiple-valued logic circuits, IET Comput. Digit. Tech. 7 (4) (2013) 167– 181.

[7] M.A. Tehrani, F. Safaei, M.H. Moaiyeri, K. Navi, Design and implementation of multi-stage interconnection networks using quantum-dot cellular automata, Microelectron. J. 42 (6) (2011) 913–922.

[8] M.H. Moaiyeri, A. Doostaregan, K. Navi, Design of energy-efficient and robust ternary circuits for nanotechnology, IET Circuits Devices Syst. 5 (4) (2011) 285–296.

[9] H. Cho, and E. E. Swartzlander, Adder designs and analyses for quantum-dot cellular automata, IEEE Trans. Nanotechnol. 6 (2013) 374–383.

[10] A. S. Shamsabadi, B. S. Ghahfarokhi, K. Zamanifar and N. Movahedinia, Applying inherent capabilities of quantum-dot cellular automata to design: D flip-flop case study, J. Syst. Architect. 55 (2009) 180–187.

[11] J. Lee, J. H. Lee, I. Y. Chung and C. J. Kim, Comparative study on energy-efficiencies of single-electron transistor-based binary full adders including nonideal effects, IEEE Trans. Nanotechnol. 10 (2011) 1180–1190.

[12] W. Wei, J. Han and F. Lombardi, A hybrid memory cell using single-electron transfer, IEEE Int. Symp. Nanoscale Architectures, San Diego, CA, 8–9 June 2011, pp. 17–23.

[13] V. Srinivasan, R. V. Venkatraman and K. K. Senthil Kumar, Schmitt trigger based SRAM cell for ultralow power operation-A CNFET based approach, Procedia Eng. 64 (2013) 115–124.

[14] C. Vudadha, P. S. Phaneendra, V. Sreehari and M. B. Srinivas, CNFET based ternary magnitude comparator, IEEE Int. Symp. Communications and Information Technologies (ISCIT), Gold Coast, Queensland, Australia, 2–5 October 2012, pp. 942–946.

[15] Safaei, F., Moaiyeri, M. H., & Tehrani, M. A. (2011, February). Design and evaluating carbon nanotube interconnects for a generic delta MIN. In Parallel, Distributed and Network-Based Processing (PDP), 2011 19th Euromicro International Conference on (pp. 488-492). IEEE.