Quantum Dot Cellular Automata Memories

Diwakar Agrawal
Department of electrical Engineering
Indian Institute of Technology Kanpur
Kanpur, India

Bahniman Ghosh
Department of electrical Engineering
Indian Institute of Technology Kanpur
Kanpur, India

ABSTRACT

In this paper we have discussed various memory architectures for Quantum Dot Cellular automata. New architectures have been proposed and their comparison has been done on the basis of area and latency. A protocol for using the serial memory has been outlined. A hybrid memory has been proposed. It is shown that the hybrid memory can be used as a tradeoff between area advantage of serial memory and latency advantage of parallel memories to get an optimized result.

General Terms

Memories, Quantum dot Cellular Automata

Keywords

QCA, Memories, Quantum Dot Cellular Auromata,

1. INTRODUCTION

Since the beginning of the CMOS era the device sizes are continuously shrinking following the Moore's law and the clock speeds and computing performance are increasing. But since the last decade the performance of the devices are improving mostly because of the exploitation of the parallelism in the processors and software. With the advancement of the technology the device sizes are reaching their physical limit [1]. Hence there is a lot of research in the alternative technologies like Quantum Dot Cellular Automata [2, 3, 4, 5]. Quantum Dot cellular automata uses the position of the confined electrons in the quantum dots to represent logic states and unlike conventional electronics does not rely on electron flow. This columbic force is responsible for all the logic operations and transfer of the states from one location to another.

A lot of research has been done in the QCA devices because of its ability to go past the physical size limits of the CMOS devices [6]. This includes work at circuit level as well as at device level. Various works have been done in the memory architectures based on QCA [7, 8, 9, 10, 11]. The objective of this paper is to provide a comprehensive discussion of the QCA memory architectures. The paper presents a hybrid architecture which is a tradeoff between qualities of series and parallel memories. We have also discussed the protocol for using the serial memory. Circuits presented in this work have been simulated using QCADesigner version 2.0.3 [12, 13].

2. QCA Memories

QCA literature proposes two types of memories i.e. parallel and serial memories. Both types of memories have their merits as well as demerits. Parallel memories have multiple 1-bit memory loops. Hence all the bits of a word can be accessed simultaneously which results into low latency. But the disadvantage of using the parallel memory is that there is a lot of circuitry which is repeated for bits in a word. Hence parallel memories are not preferred where area is the more important commodity than latency of the memory. The serial memories, on the other hand, require less area but have high latency. Since multiple bits are stored in a serial memory loop all the bits will reach the output one by one and hence have much more latency than the parallel memories. But since there is no repetition of the circuitry for the different bits, the area consumed is less. We shall now discuss various memory architectures available in detail.

2.1 Parallel Memory

Figure 1 shows a one bit memory loop used for implementing the parallel memory. The memory unit contains a multiplexer whose output is fed back into one of its inputs. During the read operation, Rd/Wt signal low, the multiplexer is in feedback mode thus functioning as a memory loop.

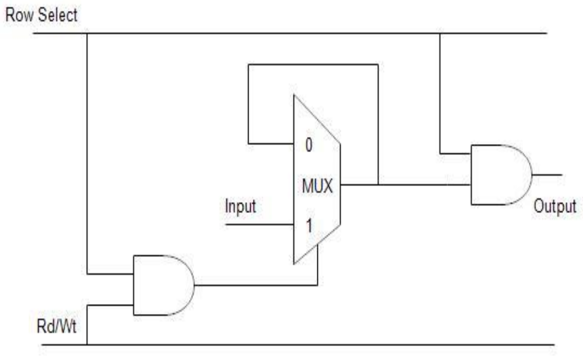

Figure 1: Schematic of parallel memory

When the Rd/Wt signal is high, new data is written into the memory unit. Figure 2 shows the circuit schematic and the layout of the memory unit.

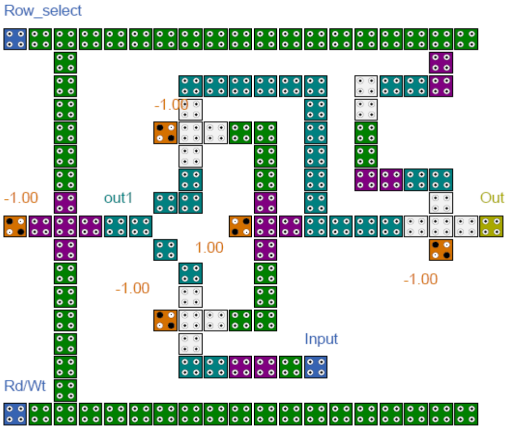

Figure 2: QCA layout of the Parallel memory

Design presented in [3] lacks in considering that all inputs to a MV should have same latency for it to work properly. The input in the AND gate just before the output does not have same delay in [3], which has been corrected in the presented design

2.2 Serial Memory

Serial memories offer an important advantage over parallel memories. Serial memories are more compact due to less duplication of circuitry. Many architectures have been proposed so far to emulate a serial memory [9, 10, 11]. But most of these works do not use the conventional clocking mechanism and deploy clocks which are complex in nature. It is obvious that presence of multiple clocking mechanisms in a circuit will be complex and difficult to implement rather that a circuit architecture which uses a conventional clocking and thus is similar in functioning to other circuits developed.

The present architecture uses the memory in motion paradigm for implementation of serial memory. In this the memory loop contains the same number of bits looping in it as is the word size of the memory. For this, the feedback loop has clock zones which are equal to 4 times the number of bits in a word (4 clock phases for 1bit). Figure 3 shows the circuit diagram for two bit and four bit serial memory structure.

The present memory does not have any complex clocking mechanism as presented in [9, 10 11], and is compatible with the general clocking scheme in used in QCA systems. For writing the Rd/Wt should be high for same number of periods as is number of bits in a word. Similarly for read operation it should be low for the same number of clock cycle as is number of bits in a word.

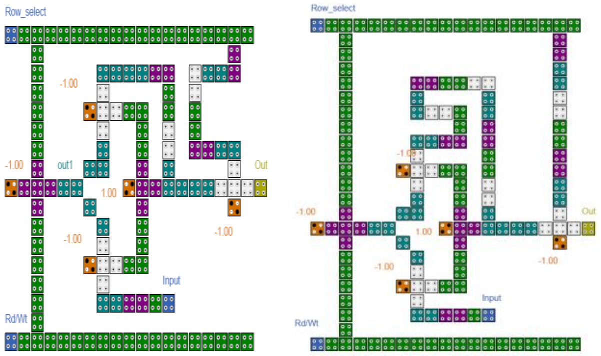

Figure 3: (a)Two bit serial memory unit (b) Four bit serial memory unit

3. HYBRID MEMORY

In previous section we saw that both types of memories have their advantages as well as disadvantages. Hence it is suggested that a combination of these two types of memories be used to get an optimal performance with respect to latency as well as area. For example if the word size required for the memory is 32 bits (say) then we can have a memory which have parallel memory units of 8 bits each to comprise a full word. Figure 4 shows an arrangement for the following effect.

A hybrid memory of 32 bit word size can be made in different ways i.e. different combination of parallel and series memory architecture. Table 1 shows the various possible sizes of the serial memory unit used in a 32 word memory unit.

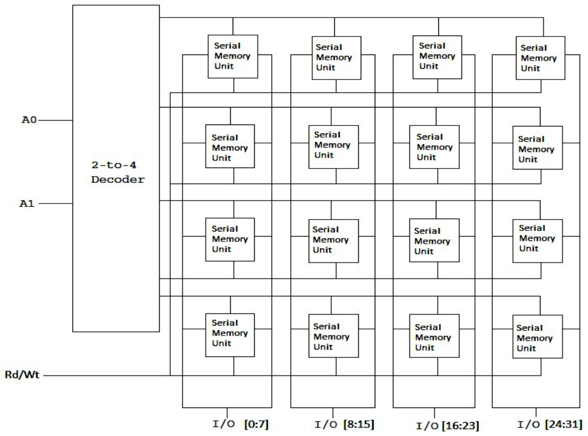

Figure 4: Layout of a Hybrid Memory (Serial memory unit size= 8 bits)

A hybrid memory of 32 bit word size can be made in different ways i.e. different combination of parallel and series memory architecture. Table 1 shows the various possible sizes of the serial memory unit used in a 32 word memory unit.

Table 1: Latency and number of cells of various 32 bit hybrid memory structures

Serial Memory unit size	Total Latency	Number if cells in one unit	Number of parallel units required	Total Number of cells
32	33	366+x	1	366+x +C
16	17	238+x	2	476+2x +C
8	9	174+x	4	696+4x+C
4	5	142+x	8	1136+8x+C
2	3	126+x	16	2016+16x+C
1	2	126+x	32	4032+32x+C

Table 1 shows the list of possible configurations of the 32 bit hybrid memory structure. ‘x’ in the table denotes the cells used for connection between the serial memory units. From [7] we see that it is about 30 cells. ‘C’ is a constant number of cells required to implement the decoder. Figure 5 shows the graph of following Function F.

$$F= (\text{Latency}) * (\text{Total number of cells required})$$

From the figure 5 we see that for a 32-bit memory the objective function attains minima for the serial memory size 4. The minimum value of the objective function in this case is 6880 as opposed to 7344 for the configuration with serial memory size 8. An objective function can be designed to give more weight to latency and vice-versa.

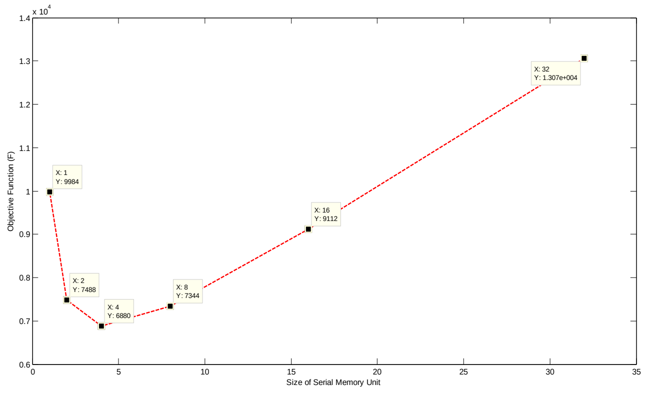

Figure 5: Various 32bit hybrid memory configurations and their Objective function values

4. PROTOCOL FOR USING SERIAL MEMORY

In this section we have described a set of rules that will be followed while using a serial memory. In a circuit most of the computation is on parallel data. Hence while using a serial memory we will have to convert the data from serial to parallel and vice versa. The serial memories are memory in motion in nature. The serial memory uses the memory in motion paradigm which means the bits are moving in a loop. Whenever the read signal arrives then it is not necessary that the first bit of a word is at the output. For example instead of order of bits 0123 the output order may be 2301. To remove this problem a counter shall be used.

Figure 1 shows only one Rd/Wt signal in circuit of a serial memory. Hence whenever the processor is not writing, the memory is supposed to be in the read mode. But this is not feasible as there is a requirement to barrel shift the output based on the location of the data bits at the instant when the data is required. Hence we propose following memory structure with separate Read and Write signals (figure 6).

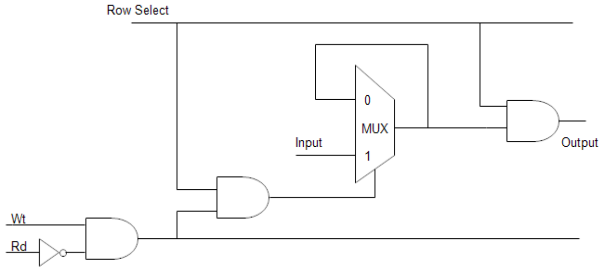

Figure 6: Modified serial memory layout

Table 2: Various possible memory states

Serial number	Wt	Rd	Function
1	0	0	--
2	0	1	Read
3	1	0	Write
4	1	1	Forbidden state

We need two separate read and write signals to know the difference between the two states when the processor wants to read the memory and when it is just not writing. State 1 and 2 will have the same effect on the memory structure, so one might wonder what the use of Read signal is. We shall later see how a separate Rd signal helps in management of the serial memory.

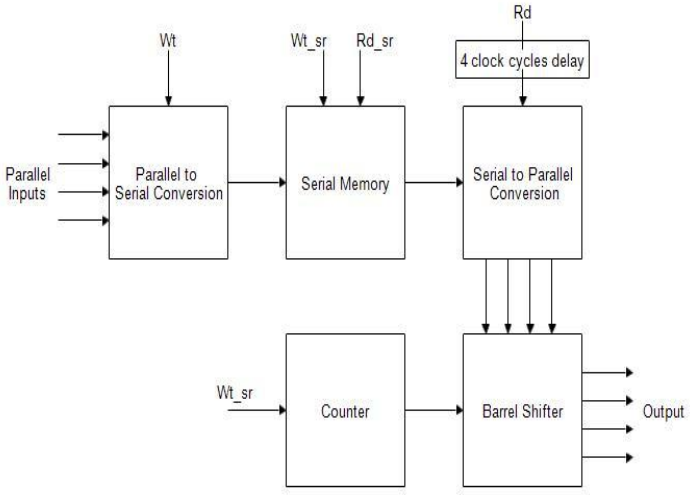

Figure 7: Serial Memory Protocol

Figure 7 shows the protocol for using a serial memory. The parallel inputs to the memories for writing are first converted into serial data with the help of a crossbar network as shown in figure 8 (taken from [14]).

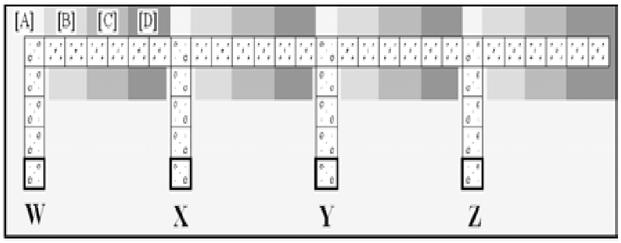

Figure 8: Crossbar network for serial parallel conversion [14]

Following are the rules for a 4 bit serial memory. The same can be extrapolated for serial memories of other sizes.

1. Processor will generate a read and write signal only for one cycle. They will be extrapolated for 4 clock cycles by the use of parallel to serial converter whenever the value of either is high. Such signal having the read/write signal serially for four cycles shall be referred henceforth as Wt_sr and Rd_sr.
2. Processor cannot generate a new read/write request within 4 cycles of generating a request since any read write operation will involve reading/writing of 4 bits. Hence one request must be completed before another can be started.

3. Parallel inputs will be converted to serial inputs whenever Wt bit is set.
4. At the falling edge of the Wt_sr signal the counter (divide by 4) will be reset. This counter [15] will be used to keep track of which bit of the word is currently at the output. Counter is reset at the falling edge because it shall be reset only when the latest data is written. In case of two back to back writes the counter will be reset only after the final write.
5. At the falling edge of the Rd signal (delayed by 4 clock cycles) the serial output of the serial memory will be forced to a parallel output with the help of the crossbar network.
6. This parallel output will be shifted using the barrel shifter [16] according to the value of the counter at the instant. This the parallel outputs bits will get align in the proper order.

We see that the Rd signal is useful because it lets the serial to parallel conversion of the serial data possible. Hence even though the implementation of a separate Rd signal does not have any effect on the memory structure, it has an important role in the big picture.

5. CONCLUSION

In this paper we have discussed various memory configurations and debated their merits and demerits. The hybrid memory configuration was presented as a tradeoff between series and parallel memories. Example of the 32 bit memory shows that the product of latency and number of cells is minimum for the structure with serial memory size 4. A protocol for using the serial memory in an architecture which also has parallel data is presented. We can see that serial memory units, though having the benefit of low area, have a lot of extra circuitry requirement for their compatibility with the parallel data handling units.

6. REFERENCES

[1] Lent , C.S., Tougaw, P. D. and Parod W.,1993, Quantum cellular automata, Nanotechnology, vol. 4, no.1, pp.49-57, Jan. 1993

[2]Tougaw, P.D., Lent, C.S.,1993, Logical devices implemented using quantum cellular automata”, J. Appl. Phys., 75 (3) 1818.

[3] M. Macucci, 2006, Quantum Cellular Automata, Imperial college Press.

[4] Lombardi, F. et al., 2008 Design and Test of Digital Circuits by Quantum-Dot Cellular Automata, F. Lombardi and J. Huang, Eds. Artech House.

[5] International Technology Roadmap for Semiconductors, Executive summary, http://www.itrs.net/Links/2005ITRS/ExecSum2005.pdf, 2005 Edition

[6] Lent, C.S.and Isaksen, B.,2003,Clocked molecular quantum dot cellular automata, IEEE Trans. Electron Devices, vol. 50, pp. 1890-1896, Sept. 2003.

[7] Walus K., Vetteth, A., Jullien, G.A., Dimitrov, V.S.,” RAM Design Using Quantum-Dot Cellular Automata”, NanoTechnology Conference,vol 2, pp. 160-163, 2003.

[8] Berzon, D.; Fountain, T.J.; , "A memory design in QCAs using the SQUARES formalism," VLSI, 1999. Proceedings. Ninth Great Lakes Symposium on , vol., no., pp.166-169, 4-6 Mar 1999

[9] Vankamamidi, V.; Ottavi, M.; Lombardi, F.(2005). Tile-based design of a serial memory in QCA. In Proceedings of the 15th ACM Great Lakes symposium on VLSI (GLSVLSI '05)

[10] Vankamamidi, V.; Ottavi, M.; Lombardi, F. "A line-based parallel memory for QCA implementation," Nanotechnology, IEEE Transactions on , vol.4, no.6, pp. 690- 698, Nov. 2005

[11] Taskin, B.; Bo Hong; , "Dual-Phase Line-Based QCA Memory Design," Nanotechnology, 2006. IEEE-NANO 2006. Sixth IEEE Conference on , vol.1, no., pp. 302-305, 17-20 June 2006

[12] K. Walus, T. J. Dysart, G. A. Jullien, and R. A. Budiman. QCADesigner: a rapid design and simulation tool for quantum-dot cellular automata. Nanotechnology, IEEE Transactions on, 3(1):26–31, 2004.

[13] http://www.mina.ubc.ca/qcadesigner, QCADesigner documentation, consulted on 20 Nov. 2011

[14] Graunke, C.R.; Wheeler, D.I.; Tougaw, D.; Will, J.D.; , "Implementation of a crossbar network using quantum-dot cellular automata," Nanotechnology, IEEE Transactions on , vol.4, no.4, pp. 435- 440, July 2005

[15] Kun Kong; Yun Shang; Ruqian Lu; , "Counter designs in quantum-dot cellular automata," Nanotechnology (IEEE-NANO), 2010 10th IEEE Conference on , vol., no., pp.1130-1134, 17-20 Aug. 2010

[16]A. Vetteth et al., “Quantum dot cellular automata carry-look-ahead adder and barrel shifter,” presented at the IEEE Emerging Telecommunications Technologies Conf., 2002