

Full Length Research Paper

A novel design of nanometric reversible cache memory

Maryam Ashkar¹ and Majid Haghparast^{2*}

¹Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran.

²Department of Computer Engineering, Shahre-Rey Branch, Islamic Azad University, Tehran, Iran.

Accepted July 7, 2011

In this paper we proposed a reversible cache memory for the first time. Four formulas for computing quantum cost, number of garbage outputs, number of constant inputs and number of gates of the proposed reversible circuit are suggested. These formulas can be used for any type of cache memory with any number of address bits and cache lines or cache blocks. Moreover we proposed a fault tolerant reversible cache memory and then formulas for computing the parameters of this circuit are also suggested.

Key words: Reversible logic, cache memory, quantum cost, faults tolerant.

INTRODUCTION

One of the significant factors in VLSI circuit design is the control of energy dissipation. Actually the information loss for each bit dissipates an amount of $KT \ln 2$ joules of energy, where $K = 1.3806505 \times 10^{-23} \text{JK}^{-1}$ is the Boltzmann's constant and T is the absolute temperature at which computation is performed (Landauer, 1961). The number of bits removed during an operation is proportionate to the amount of energy dissipated in a system. Bennett demonstrated that for refraining $KT \ln 2$ energy dissipation in a logic circuit, the circuit must be constructed with reversible logic gates. In the reversible circuits, internal power dissipation is theoretically zero because the information is not lost. In this type of circuits, there is a one to one correspondence between input and output vectors and the number of inputs and outputs are equal. Furthermore, reversible circuits can be used in optical computing, quantum computing and nanotechnology. It is important to take into consideration that quantum circuits are constructed with reversible logic gates. One of the distinguishing characteristics of these types of circuits is the least number of garbage outputs, constant inputs and gates. The output that is not used for computation is called garbage output and the input that is added to a function to make it reversible is called constant input (Vasudevan et al., 2004; Bennett, 1973).

This is the first effort to design a reversible cache

memory and to introduce formulas for computing the quantum cost, number of garbage outputs, constant inputs and gates and to convert this reversible cache memory to a fault tolerant reversible circuit.

BASIC CONCEPTS

There are several reversible logic gates for designing reversible logic circuits such as Feynman gate (FG) (Feynman, 1985), F2G gate (Toffoli, 1980), Fredkin gate (FRG) (Fredkin and Toffoli, 1982), Peres gate (PG) (Peres, 1985), NFT gate (Haghparast and Navi, 2008) and HNFG gate (Haghparast and Navi, 2008a). A 2×2 Feynman gate also known as controlled NOT is depicted in Figure 1. It has two outputs: $P = A$ and $Q = A \oplus B$. If control input "B" is '1, first output (P) becomes "A" and second output (Q) becomes NOT of A, and if control input (B) is '0, first output (P) becomes (A) and (Q) becomes (A) (Haghparast and Navi, 2008b). The F2G gate is a Feynman gate with an extra input and output. It has two controlled NOT operations. A 3×3 F2G is a fault tolerant reversible gate which is shown in Figure 2 (Haghparast et al., 2009; Parhami, 2006). Fredkin gate has 3 inputs and 3 outputs. Input 'A is sent as the first output. It controls second and third inputs 'B and 'C. If $A = 1$, the second output (Q) becomes the "C" and the third output (R) becomes the "B". Actually B and C inputs are swapped in outputs. If $A = 0$, Q and R outputs become "B" and "C". Fredkin gate is shown in Figure 3

*Corresponding author. E-mail: haghparast@iausr.ac.ir.

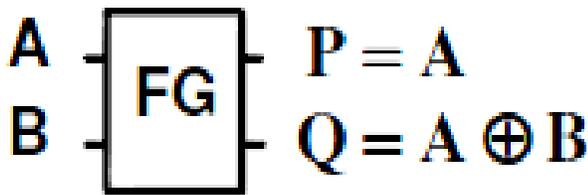


Figure 1. Feynman gate.

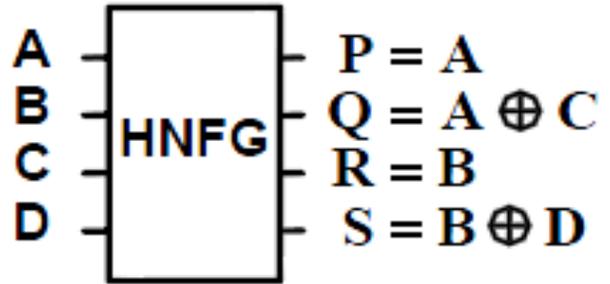


Figure 6. HNFG gate.

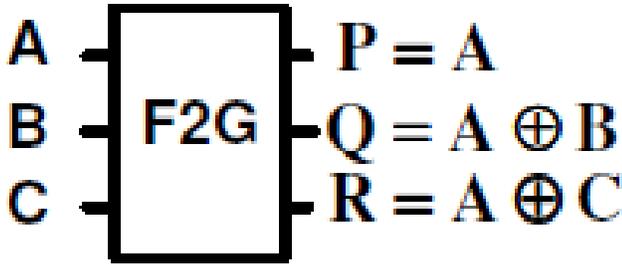


Figure 2. F2G gate.

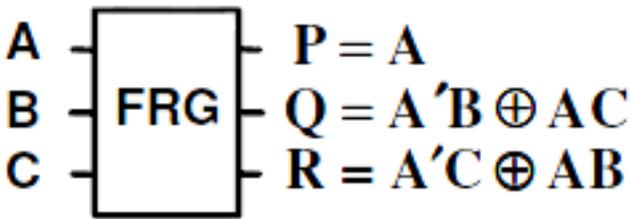


Figure 3. Fredkin gate.

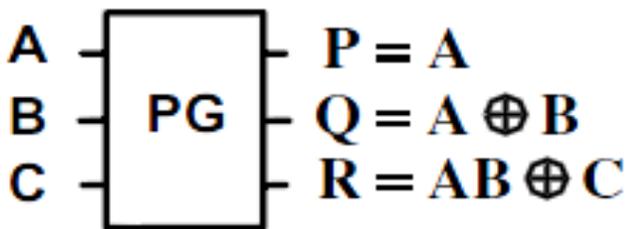


Figure 4. Peres gate.

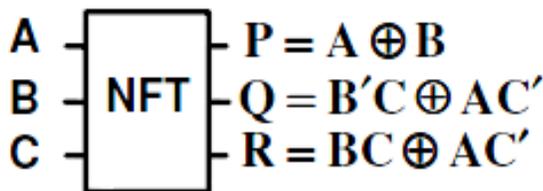


Figure 5. NFT gate.

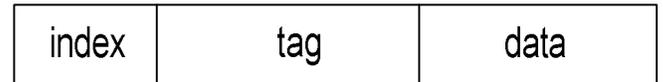


Figure 7. The structure of a cache block.

(Haghparast and Navi, 2007; Shams et al., 2008; Mohammadi et al., 2009). As it is shown in Figure 4, Peres gate (PG) has 3 inputs (A, B, C) and 3 outputs ($P = A$, $Q = A \oplus B$, $R = AB \oplus C$). Since the 3*3 NFT gate shown in Figure 5 is a fault tolerant reversible gate, its input parity and output parity are equal.

HNFG gate is illustrated in Figure 6. Each HNFG gate can be used instead of two 2*2 Feynman gates without the garbage outputs (Mohammadi et al., 2008; Mohammadi et al., 2008a).

CACHE MEMORY

Cache memory is smaller in size and higher in speed compared with other types of memories. This memory stores copies of data that are used in the main memory. The access rate of this type of memory is relatively high due to its lower volume. When the processor needs to read or write from/to a location in the main memory, first it checks whether a copy of that data is in the cache or not. So the processor immediately reads from or writes to the cache. This process is comparatively much faster than reading from or writing to the main memory. Cache memory is also used in the CPU (central processing unit). Cache row or cache block has the following structure and is shown in Figure 7. Index is a unique number for each cache block. Tag is used to refer to the address in the cache memory. Figure 8 shows two memories: cache memory and main memory. In both memories data (cache line) exists but the size ranges from 8 to 512 bytes in different designs. The size of the cache line is normally larger than the usual size requested by CPU instruction; the size ranges from 1 to 16 bytes (the largest addresses and data handled by current 32 and 64 bits architectures being 128 bits long). A processor when trying to read/write from/to a memory checks whether a

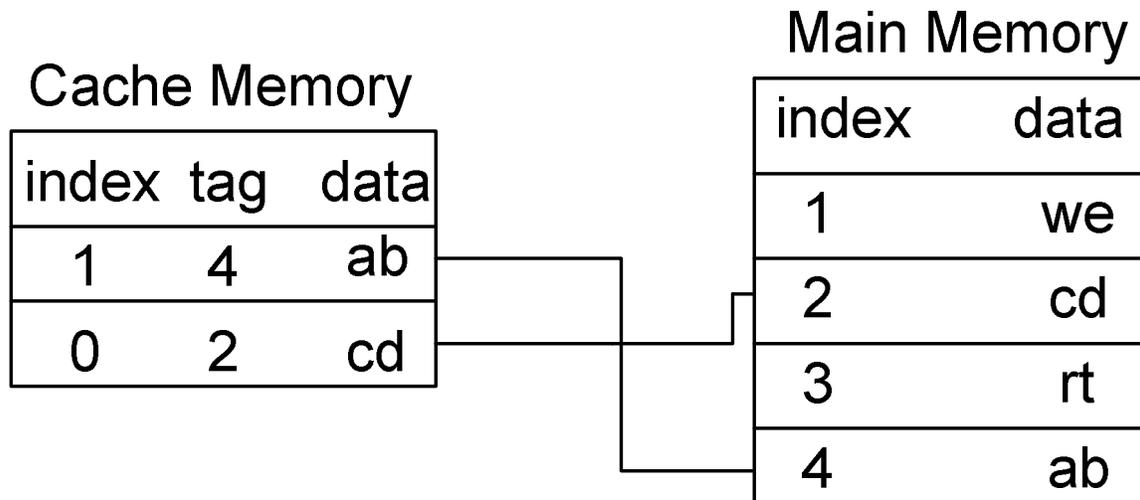


Figure 8. Diagram of a CPU.

copy of the new data exists in the cache memory or not. First, new coming address is compared with the existent addresses in the cache memory that is every bit of coming address is checked with every bit of the existent addresses. To do this comparison, when a new address enters the cache memory and sets on the address bus, the first bit of it becomes XNOR with the first bits of existent addresses in all the cache blocks. The same process is repeated for the next bits of the new imported address. These processes work parallel. If the result of the XNOR logic operator becomes 1, it means that the first bit of the new imported address and the first bit of the existent address of the first cache block are equal. So each XNOR operator shows whether or not the new coming bit exists in the cache memory.

The result of all XNOR operators together becomes AND operator. If the result of one AND logic operator becomes 1, it means that a copy of data exists in the cache memory. Consequently, the result of all AND operators becomes OR operator, output of which shows data control. Data control (output of OR operator) = 1 can mean the following:

- 1) Hit.
- 2) The data of new imported address exists in the cache memory and it is not necessary to trace the main memory for it.
- 3) One of the AND operators had become 1 before. Since the result of each AND operator is connected to actuator signal of each related cache block, actuator signal connected to this AND operator will become 1 and data can be sent or received to/from data bus.

If output of OR operator (data control) becomes 0 commonly known as Miss and it means that none of the AND logic operators becomes 1, so the imported address

is not equal to an existent address and consequently a copy of data does not exist in the cache memory.

OUR PROPOSED REVERSIBLE CACHE MEMORY

In this paper, two reversible cache memories with the different number of bits and cache blocks are proposed and depicted in Figures 9 and 10. The reversible cache memory functions similar to the binary logic cache memory but uses reversible gates. In order to design our proposed reversible circuit, we made use of HNFG, PG and NFT gates. The HNFG gates are used for comparing the bits of the new coming address with the bits of the existing addresses separately to check whether the correspondent bits are similar or not. But we used PG reversible gates to check all bits simultaneously. In other words the PG reversible gates check the cache for the copy of data. If the result of PG gates is 1 the data exists in the cache memory. Finally, NFT gate in Figure 9 shows Hit and Miss. The reversible cache memory depends on the number of bits and cache blocks (Figures 9 and 10). The first proposed reversible cache memory (Figure 9) has 4 bits and 2 cache blocks, so the number of PG, HNFG, NFT gates and quantum cost of this circuit are 6, 4, 1 and 37 respectively. The second proposed reversible cache memory (Figure 10) has 2 bits and 4 cache blocks so the number of PG, HNFG, NFT gates and quantum cost of this circuit are 4, 4, 3 and 39 respectively; certainly for another number of bits and cache blocks, quantum cost, number of garbage outputs, constant inputs and gates of the reversible circuits will change but the basic structure of figures and type of gates will remain stable.

According to the two figures it can clearly be illustrated that the quantum cost, number of garbage outputs,

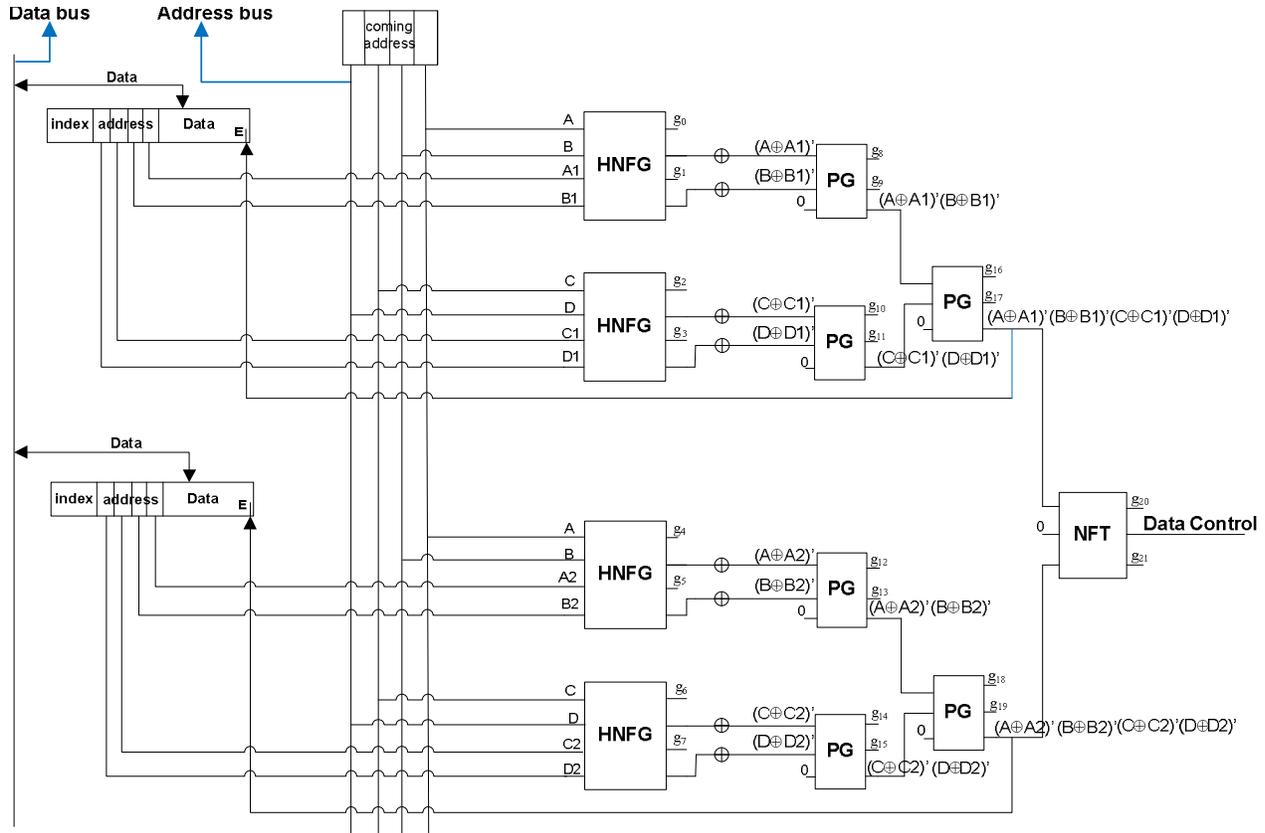


Figure 9. The first proposed reversible cache memory with 4 address bits and 2 cache blocks.

constant inputs and gates in the reversible cache memory are related to the number of bits and cache blocks.

OUR PROPOSED FAULT TOLERANT REVERSIBLE CACHE MEMORY

Here, two fault reversible cache memories with the different number of bits and cache blocks are proposed and depicted in Figures 11 and 12. An important step in converting the reversible cache memory to the fault tolerant reversible cache memory is the use of fault tolerant reversible gates such as NFT and F2G (Parhami, 2006; Azad Khan, 2002; Thapliyal et al., 2006).

EVALUATION OF OUR PROPOSED REVERSIBLE CACHE MEMORY

Until now quantum cost, number of garbage outputs, constant inputs and gates of the reversible cache memory has been calculated by drawing complicated and time consuming figures, but since it is a demanding task, we have proposed our own formulas. Our proposed formulas have the following benefits:

- 1) Time benefit when it comes to computing the quantum cost, number of garbage outputs, constant inputs and gates of the reversible cache memory.
- 2) It is not necessary to draw complicated and time consuming figures.
- 3) It is the simplest route for computing the quantum cost, number of garbage outputs, constant inputs and gates of the reversible cache memory.
- 4) For every bits.
- 5) For every cache blocks.

Our proposed formulas are as follows:

QC = quantum cost, m = number of bits, n = number of cache blocks:

- 1) Number of gates = $(nm/2)$ HNFG+ $(nm-n)$ PG+ $(n-1)$ NFT.
- 2) Number of garbage outputs = $(3 nm)-2$.
- 3) Number of constant inputs = $(nm)-1$.
- 4) $QC = (5 nm) + n-5$.

Evaluation of Figure 9 (m = 4, n = 2):

- 1) Number of gates = $(nm/2)$ HNFG+ $(nm-n)$ PG+ $(n-1)$ NFT = $4HNFG + 6PG + 1NFT$.

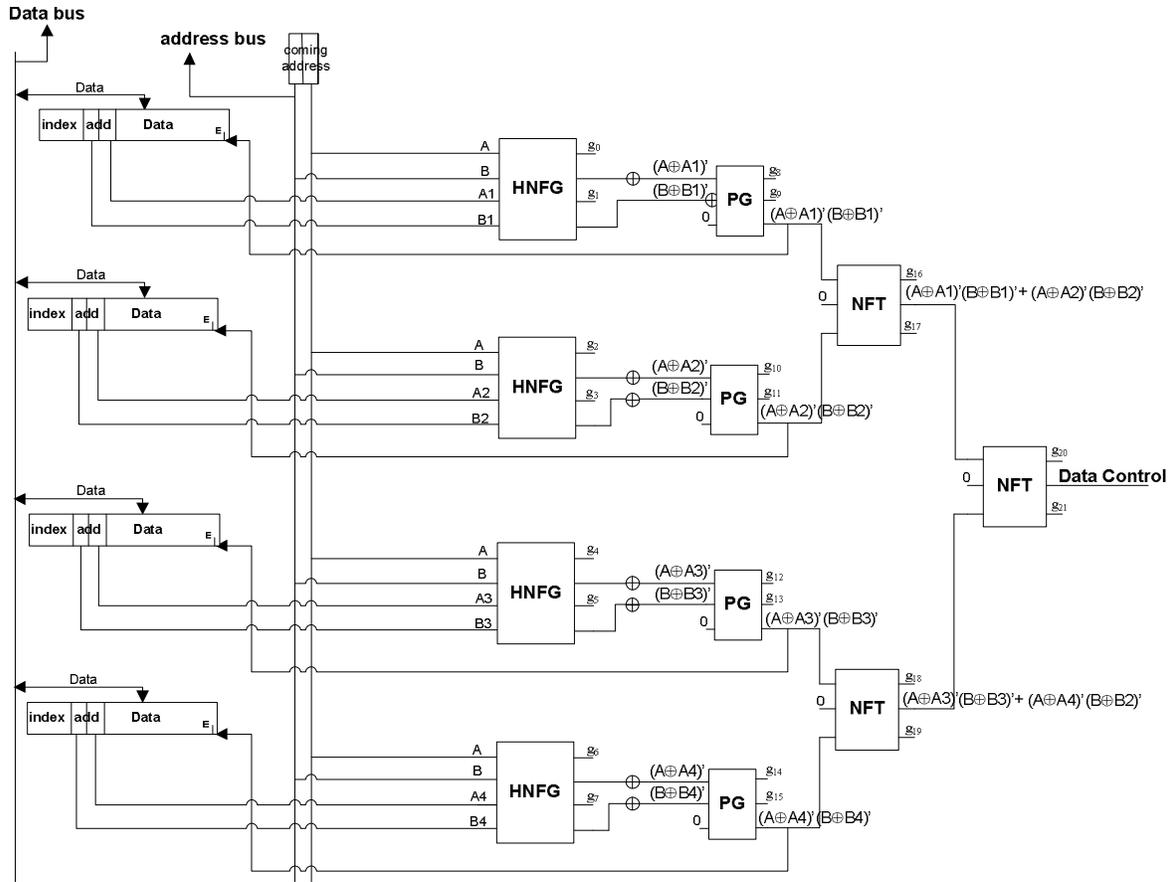


Figure 10. The second proposed reversible cache memory with 2 address bits and 4 cache blocks.

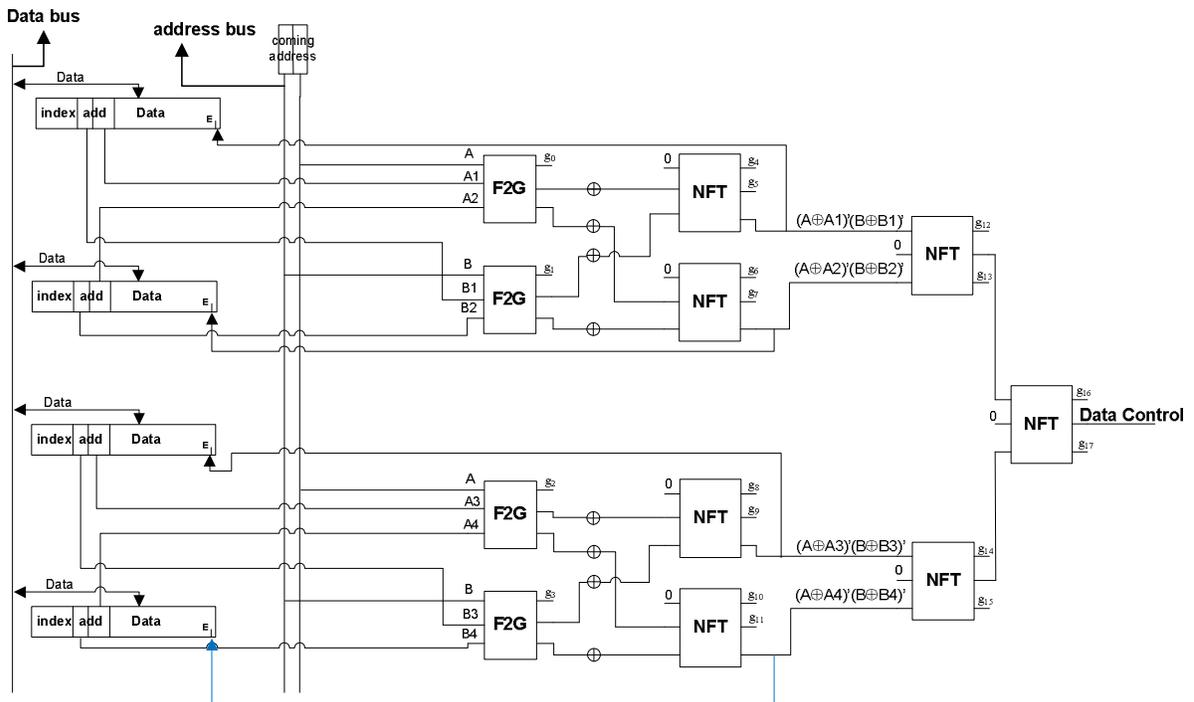


Figure 11. The first proposed fault tolerant reversible cache memory with 2 address bits and 4 cache blocks.

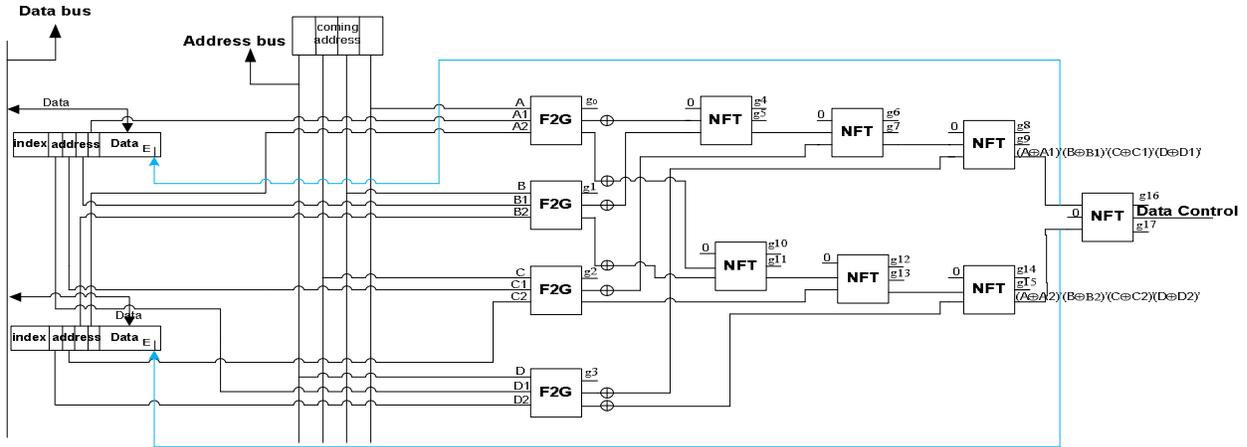


Figure 12. The second proposed fault tolerant reversible cache memory with 4 address bits and 2 cache blocks.

Table 1. Evaluation of our proposed reversible cache memory.

	No. of gates ((nm/2) HNFG+ (nm-n) PG+ (n-1) NFT)	No. of garbage outputs ((3nm)-2)	No. of constant Inputs ((nm)-1)	Quantum cost (5 nm + n-5)
Figure 9	4HNFG + 6PG + 1NFT = 11	22	7	37
Figure 10	4HNFG + 4PG + 3NFT = 11	22	7	39

Table 2. Evaluation of our proposed fault tolerant reversible cache memory.

	No. of gates (nm/2) F2G + (nm-1) NFT	No. of garbage outputs ((5/2)(nm))-2	No. of constant inputs (nm-1)	Quantum cost (6 nm -5)
Figure 11	4F2G + 7NFT	18	7	43
Figure 12	4F2G + 7NFT	18	7	43

- 2) Number of garbage outputs = $(3 \text{ nm}) - 2 = (3 \cdot 2 \cdot 4) - 2 = 22$.
- 3) Number of constant inputs = $(\text{nm}) - 1 = (2 \cdot 4) - 1 = 7$.
- 4) $\text{QC} = (5 \text{ nm}) + n - 5 = (5 \cdot 2 \cdot 4) + 2 - 5 = 37$.

Evaluation of Figure 10 (m = 2, n = 4):

- 1) Number of gates = $(\text{nm}/2) \text{ HNFG} + (\text{nm}-n) \text{ PG} + (n-1) \text{ NFT} = 4\text{HNFG} + 4\text{PG} + 3\text{NFT}$.
- 2) Number of garbage outputs = $(3 \text{ nm}) - 2 = (3 \cdot 4 \cdot 2) - 2 = 22$.
- 3) Number of constant inputs = $(\text{nm}) - 1 = (4 \cdot 2) - 1 = 7$.
- 4) $\text{QC} = (5 \text{ nm}) + n - 5 = (5 \cdot 2 \cdot 4) + 4 - 5 = 39$.

Summary parameters of Figures 9 and 10 are shown in Table 1.

EVALUATION OF OUR PROPOSED FAULT TOLERANT REVERSIBLE CACHE MEMORY

For computing the quantum cost, number of garbage

outputs, constant inputs and gates of the fault tolerant reversible cache memory, we have proposed four formulas. These formulas are not equal with the existent formulas in the previous section of this paper. Our proposed formulas are as follows:

- 1) Number of gates = $(\text{nm}/2) \text{ F2G} + (\text{nm}-1) \text{ NFT}$.
- 2) Number of garbage outputs = $((5/2)(\text{nm})) - 2$.
- 3) Number of constant inputs = $(\text{nm}) - 1$.
- 4) $\text{QC} = (6 \text{ nm}) - 5$.

All the parameters of our proposed fault tolerant reversible cache memory are depicted in Table 2.

CONCLUSION

In this paper as in addition to our pioneering attempt to design reversible cache memory, we have proposed four formulas for computing the quantum cost, number of garbage outputs, constant inputs and gates. With these formulas, there is needless to draw complicated and

time-consuming figures for computing the parameters of the reversible cache memory. Furthermore, by making use of reversible F2G and NFT gates, we converted reversible cache memory to fault tolerant reversible circuit and then we have suggested formulas for computing the parameters of this fault tolerant circuit.

REFERENCES

- Landauer R (1961). Irreversibility and heat generation in the computing process. *IBM J. Res. Dev.*, 5(3): 183-191.
- Vasudevan DP, Lala PK, Parkerson JP (2004). A novel approach for online testable reversible logic circuit design. *Proceedings of the 13th Asian Test Symposium*. pp. 325-330.
- Bennett CH (1973). Logical reversibility of computation. *IBM J. Res. Dev.*, 17: 525-532.
- Feynman R (1985). Quantum mechanical computers. *Optics News*, 11: 11-20.
- Toffoli T (1980). Reversible computing. *Tech Memo MIT/LCS/TM-151*. MIT Lab for Computer Science.
- Fredkin E, Toffoli T (1982). Conservative logic. *Intl. J. Theor. Phys.*, 21: 219-253.
- Peres A (1985). Reversible logic and quantum computers. *Phy. Rev.* 32: 3266-3276.
- Haghparast M, Navi K (2008). A Novel Fault Tolerant Reversible Gate for Nanotechnology Based Systems. *Am. J. Appl. Sci.*, 5(5): 519-523.
- Haghparast M, Navi K (2008a). A Novel Reversible BCD Adder for Nanotechnology Based Systems. *Am. J. Appl. Sci.*, 5(3): 282-288.
- Haghparast M, Navi K (2008b). Design of a Novel Fault Tolerant Reversible Full Adder for Nanotechnology Based Systems. *World Applied Sciences J.* 3(1): 114-118.
- Haghparast M, Navi K, Eshghi M (2009). Optimized Reversible Multiplier Circuit. *J. Circuits, Syst., Comput.*, 18(2): 311-323.
- Haghparast M, Navi K, Jafarali SJ, Hashemipour O (2008). Design of a Novel Reversible Multiplier Circuit Using HNG Gate in Nanotechnology. *World Appl. Sci. J.*, 3(6): 974-978.
- Haghparast M, Navi K (2007). A Novel Reversible Full Adder Circuit for Nanotechnology Based System. *J. Appl. Sci.*, 7(24): 3995-4000.
- Shams M, Haghparast M, Navi K (2008). Novel Reversible Multiplier Circuit in Nanotechnology. *World Appl. Sci. J.*, 3(5): 806-810.
- Mohammadi M, Haghparast M, Eshghi M, Navi K (2009). Minimization and Optimization of Reversible BCD-Full Adder/Subtractor Using Genetic Algorithm and Don't Care Concept. *Int. J. Quantum Inf.*, 7(5): 969-989.
- Mohammadi M, Eshghi M, Haghparast M (2008). On design of Multiple Valued Sequential reversible circuits for Nanotechnology Based Systems. *IEEE*. pp. 1-6.
- Mohammadi M, Eshghi M, Haghparast M, Bahrololoom A (2008a). Design and Optimization of Reversible BCD Adder/Subtractor Circuit for Quantum and Nanotechnology Based Systems. *World Appl. Sci. J.*, 4(6): 787-792.
- Parhami B (2006). Fault tolerant reversible circuits. *Proc. 40th Asilomar Conf. Signals, Systems, and Computers*. Pacific Grove, CA.
- Azad Khan MD (2002). Design of full adder with reversible gate. *International Conference on Computer and Information Technology*. Dhaka, Bangladesh. pp. 515-519.
- Thapliyal H, Kotiyal S, Srinivas MB (2006). Novel BCD adders and their reversible logic implementation for IEEE 754r format. *Proceedings of the 19th International Conference on VLSI Design*. pp. 3-7.