Quantum Memory Management Systems

Caglar Koca
Next-generation and Wireless Communications Laboratory
Koc University
Istanbul, Turkey
cagkoca@ku.edu.tr

Ozgur B. Akan
Next-generation and Wireless Communications Laboratory
Koc University
Istanbul, Turkey
akan@ku.edu.tr

ABSTRACT

Quantum computation is a developing area in science. Owing to the new developments in nanotechnology, some practical quantum computing applications are emerging. Even though quantum computation possesses many inherent superiorities, it suffer from a tight bottleneck: Quantum Storage. In this paper, we identify properties of a good quantum storage unit and propose unique quantum memory structures and discuss them in terms of circuit complexity, access delay complexity, maximum connectivity and ancillary qubits required. We reveal that quantum teleportation provide advantage only in maximum connectivity among the properties we analyzed.

Categories and Subject Descriptors

B.10.10 [**Quantum Technologies**]: Quantum Computation; C.1.4 [**Other Architectures**]: Quantum Computing—*complexity measures, performance measures*

General Terms

Quantum Information

Keywords

Quantum communication, quantum computation, quantum storage, teleportation, swap

1. INTRODUCTION

Ever since Feynman hinted the quantum computing in 1982, a long way has been covered to turn this dream into reality. A number of quantum algorithms have revolutionized many fields of computing, especially in Searching [2] and Factoring [7]. Some quantum processor units [6, 1] and quantum arithmetic logic unit structures have been developed [9, 10] and some of them are engineered [10]. Some memory structures are also theorized [3], and thanks to the recent developments in nanotechnology, small sized quantum memory systems has been built [8]. However, to the best of our knowledge, no quantum memory management system feasible for large memory contents have been suggested.

Classical memory systems use finite number of buses to access a memory location. The desired memory location can be activated by various methods using encoders or multiplexers, while the other memory cells are kept at a high impedance state. The desired content may reach any place from the bus as it is not affected by other memory cells.

Due to the lack of high impedance gates for quantum states, quantum memory cells cannot be accessed directly by their addresses. In fact, there are two mechanisms which may transport qubits from one location to another: Swapping and teleportation [4].

For a memory consisting of N qubits, complexity of reaching the desired content is obviously $O(N)$ which renders unorganized swapping impractical. Teleportation itself cannot be a solution as well, there has to be another search at the target location of the qubit. Moreover, the EPR pairs are also qubits and they somehow have to be distributed to the teleportation zones. Note that sending EPR pairs are no different than retriving qubits. In both cases, a qubit is transferred from a target to a destination.

The rest of the paper is organized as follows. In the section II, we introduce the features of a quantum memory management system. In Section III, we explain and analyze possible one-staged memory management systems using only swapping. In Section IV, we deepen our analysis to multilayered memory management architectures. In Section V, we employ teleportation as well as swapping for memory management. We compare the suggested systems in Section VI. Section VII is conclusion.

2. QUANTUM MEMORY MANAGEMENT SYSTEMS

We suggest that a quantum memory management system has four main properties:

1. Access Delay Complexity
2. Circuit Complexity
3. Maximum Connectivity
4. Ancillary Qubits

Access delay complexity, ADC, is how access time increases as the memory size increases. ADC affects the overall speed of the system. Circuit complexity, CC, is how the number of gates required to access a memory location increases as the memory size increases. CC is related to the

NANOCOM'15, September 21 - 22, 2015, Boston, MA, USA
©2015 ACM 978-1-4503-3674-1/15/09 ...$15.00.
http://dx.doi.org/10.1145/2800795.2800810.

circuit size, cost of the system and feasibility. Maximum Connectivity, MC, is the most number of gates connected to a qubit. MC is a concept close to fan-out in classical circuits. Ancillary Qubits, AQ, are the number of qubits required to reach a memory location. These qubits are used in qubit teleportation, as we need an EPR pair to achieve teleportation. Note that MC and AQ give us direct values rather than complexities.

It is clear that all of these quantities should be small for a memory management system to be feasible. Below, we will explain and analyze the possible memory architectures with their access delay complexities, circuit complexities, maximum connectivity and ancillary qubit numbers.

3. ONE LAYERED ARCHITECTURES WITH ONLY SWAPPING

In this chapter, we discuss the possible architectures for a one layered memory management system. These architectures may not be feasible for large memories but they may constitute building blocks for larger memory structures. Note that we use Fredkin Gates [4] which have been physically realized and can be used to construct the suggested schemes. [11].

3.1 Simple Swapping

Simple Swapping is basically to swap each qubit until the target qubit reachs its destination. A simple swapping circuit is depicted on Figure 1. It is obvious that the circuit complexity of Swapping is $O(N)$, as it requires N Fredkin Gates. The access delay complexity is also N, due to the fact that, on average, we need to swap the memory content $\frac{N}{2}$ times. Its maximum connectivity is 2, as each qubit is connected to two gates. Due to the large amount of delay, we can say that simple swapping is not a feasible suggestion for fast access for large blocks. Note that output can be taken via any quantum wire.

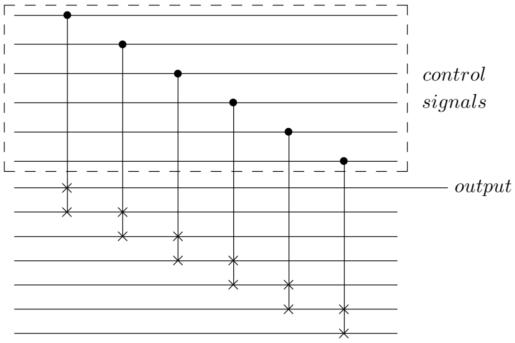

Figure 1: Simple Swapping Circuit.

3.2 Connected Swapping

We suggest that in order to enhance the simple swapping, we can simply connect all the memory cells to the target location. Therefore, one swap is sufficient to transfer the desired content to the target. The Connected Swapping is shown on Figure 2. Even though the circuit complexity remains as $O(N)$, the access delay complexity is now reduced to $O(1)$. However, the maximum connectivity becomes N. Note that, connecting many Fredkin Gates to a single memory cell may be impractical.

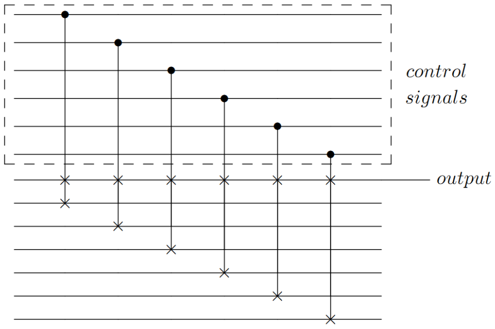

Figure 2: Connected Swapping Circuit.

3.3 Logarithmic Swapping

We can further idealize swapping to make Connected Swapping more practical, by trading the maximum connectivity with access delay complexity. In Logarithmic Swapping, some memory cells have increased connectivity than others so that the desired content hops to more connected memory locations to reach its target. Figure 3 displays our suggested Logarithmic Swapping scheme.

Note that in Logarithmic Swapping $\frac{N}{2}$ memory cells one Fredkin Gate connected to them. $\frac{N}{4}$ of them have two, $\frac{N}{8}$ of them have three Fredkin Gates and so on. Due to this architecture we name it as logarithmic. The most connected memory cell will have $\log_2(N)$ Fredkin Gates. The total number Fredkin Gates needed to access all memory locations can be calculated as, $\sum_i^{\log_2(N)} i\frac{N}{2^i}$ which is also $O(N)$.

The architecture of the circuit is designed to make sure that no cell is connected to more than $\log_2(N)$ other cells. This is due to the fact that each cell hops to a twice as many connected cell to reach the output.

It is obvious that the worst case access delay time is proportional to $\log_2(N)$. To calculate ADC, we assume that all memory locations are simultaneously accessed. $\frac{N}{2}$ least connected cells jump to the second level with one move. After this jump, there exist $\frac{N}{2} + \frac{N}{4}$ cells in the second level. Summing them all yields:

$$\sum_{i=1}^{log_2(N)} \frac{N}{2^i} = N\log(N) - N \qquad (1)$$

We can find ADC for a single memory cell by dividing (1) with N, hence ADC is $O(\log_2(N))$. Although the circuit complexity is not improved, the access time is much better compared to Simple Swapping and and much more practical compared to Connected Swapping.

Note that in the Logarithmic Swapping circuit, there appears two cells are connected to three Fredkin Gates, instead of one cell. This is due to the fact that the size of the memory is only eight qubits. For larger circuits, $\frac{N}{8}$ of the cells having three connectivity holds.

4. MULTI-LAYERED ARCHITECTURES WITH ONLY SWAPPING

In the previous section, we discussed the one layered architectures using only swapping. Here, we use these architectures and organize multi-layered memory management systems. Note that there exist many possibilities, however, we

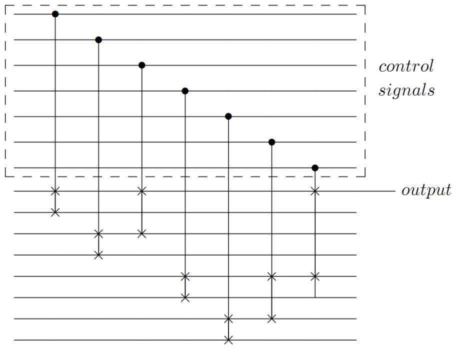

Figure 3: Logarithmic Swapping Circuit.

focus only on three feasible circuitry.

4.1 Mixed Swapping

Using Simple Swapping and Logarithmic Swapping, it is possible to build more realistic architectures. By building fixed size memory blocks using Logarithmic Swapping, and connecting them using Simple Swapping, it is possible to reach any memory location with relatively faster than Simple Swapping. It is also more realistic as the number of gates connected to a single memory cell is limited, unlike Connected Swapping and Logarithmic Swapping. A Mixed Swapping Circuit Scheme is provided on Figure 4.

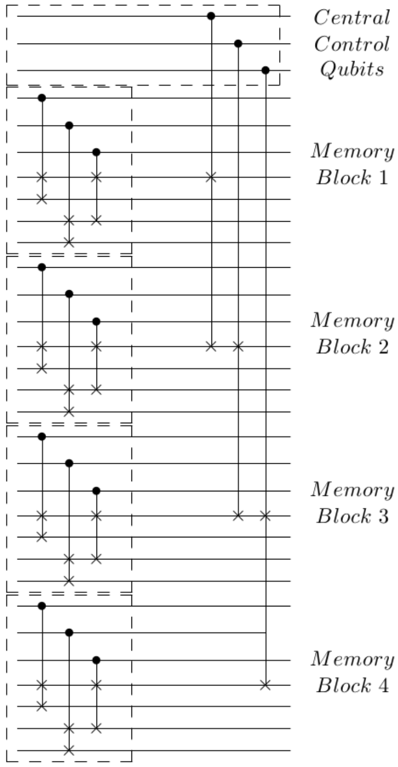

Figure 4: Mixed Swapping Circuit.

We can calculate the circuit complexity as follows. If the memory block size is n, there will be $\frac{N}{n}$ blocks. Each block has $O(n)$ Fredkin Gates and as a result, CC will be $O(N)$.

Access delay complexity depends both on the memory block size and number of total memory cells. Inside a memory block, ADC is $O(\log_2(n))$ swapping is required and there will be $\frac{N}{n}$ memory blocks. Therefore, the ADC becomes $O(\frac{N}{n}log_2(n))$. For a fixed memory size, i.e., for constant N, minimizing this result gives us $n = \mathrm{e}$. Even though a memory block cannot be constructed with such small number of memory cells, to achieve lower ADC we need to keep the memory block sizes small.

The maximum connectivity for such a system depends only on the memory block size. Therefore, the maximum connectivity is $\log_2(n)$

4.2 Concatenated Connected Swapping

We can use memory blocks organized with Connected Swapping and connect them using Connected Swapping in the second layer as well. We can even increase the number of layers in the circuitry. A Concatenated Connected Swapping Circuit is displayed in Figure 5.

The advantage of such an architecture is obviously in the maximum connectivity. MC for each block is n and for a memory size of N, MC of the overall circuitry will be $max(n+1, \frac{N}{n})$. The best performance of this architecture is at $n = \sqrt{N}$, which is $\sqrt{N}+1$. A better performance might be achieved using multiple layers. If we use m layers of Connected Swapping, MC will be $\sqrt[m]{N}+1$.

Access delay complexity performance of this architecture is also favorable. At each memory block, ADC is simply $O(1)$ and for m layers it is $O(m)$.

Circuit complexity of Concatenated Advanced Swapping can be calculated as follows. For each block CC is $O(n)$. In this configuration, there will be n^{m-1} blocks in the first layer and n^{m-2} blocks in the second layer etc. Summing all the layers, we get $n\frac{1-n^{m-1}}{1-n}$ blocks, and for large n, there exists approximately n^{m-1} blocks. Multiplying this result with $O(n)$, the overall complexity becomes $O(n^m)$, which is equal to $O(N)$.

4.3 Concatenated Logarithmic Swapping

We can organize memory blocks using Logarithmic Swapping and collect the outputs of these blocks in an intermediate location, where we can employ Logarithmic Swapping one more time. Furthermore, we can increase the number of layers in the memory architecture. A Concatenated Logarithmic Swapping Circuit is in Figure 6.

The CC for this architecture is also $O(N)$, as in the previous architectures. Note that if the block size is n, there will be $\frac{N}{n}$ for a memory size of N qubits. Each memory block has a CC of $O(n)$, hence the whole system has a CC of $O(N)$.

To calculate the access delay complexity, we need to consider the layered structure of the architecture. Inside a memory block the ADC is $O(\log_2(n))$, and in the second layer, it is $O(\log_2(\frac{N}{n}))$. Summing these values give us $O(\log_2(N))$. Note that this result is independent of the block size. Note that addition of other layers do not change this result, as $O(\log_2(\frac{N}{n_1}) + \sum_i \log_2(\frac{n_i}{n_{i+1}}))$, which is again $O(\log_2(N))$.

The maximum connectivity is the connectivity in each block. In the memory blocks, MC is $\log_2(n)$ and in the intermediate location it is $\log_2(\frac{N}{n})$. If the most connected qubits of the layers are connected to each other, MC becomes $\log_2(N)$, which gives us no advantage over the simple

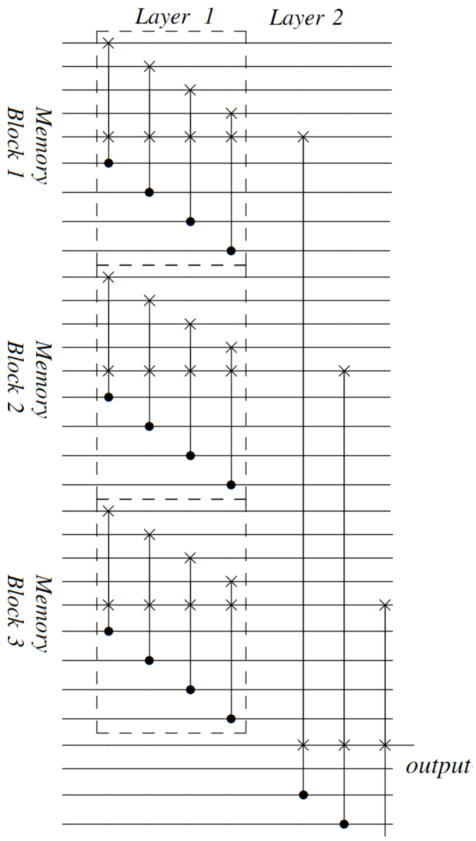

Figure 5: Concatenated Connected Swapping Circuit.

Logarithmic Swapping. Hence, we need to connect the least connected qubits in the first layer to the second layer. MC then becomes, $max(\log_2(\frac{N}{n}), \log_2(n))$. This result is minimized for $n = \sqrt{N}$. For m-layered architecture, MC becomes $log_2(\sqrt[m]{N})$. Note that this is the best result among memory management architectures using swapping.

Also note that no ancillary qubit is used in Swapping.

5. ARCHITECTURES WITH SWAPPING AND TELEPORTATION

To reduce the access delay complexity due to swapping, employing teleportation is also suggested [5]. This approach is based on small memory units connected to qubit refresh units and to a code teleporter. The contents of the memory cells are to reach the teleporter by Simple Swapping, and then they are to be teleported to their target location. Figure 7 is the memory structure suggested by [5].

In this architecture, the main delay is due to swapping, and the number of swappings required is proportional to the size of quantum memory units. However, these memory units cannot be made arbitrarily small, as after teleportation the qubit does not directly reach its desired destination. Qubits may only be teleported to an intermediate location where one of the EPR pairs used for teleportation is stationed. Hence, the small sized memory units require larger number of EPR pairs to be teleported and it increases the size of the intermediate location, where a second search needs to be done. Hence, as we will see, Teleportation only reduces the maximum connectivity and has no effect on any of the other discussed criteria. Here, we discuss the implementation of teleportation on different swapping techniques.

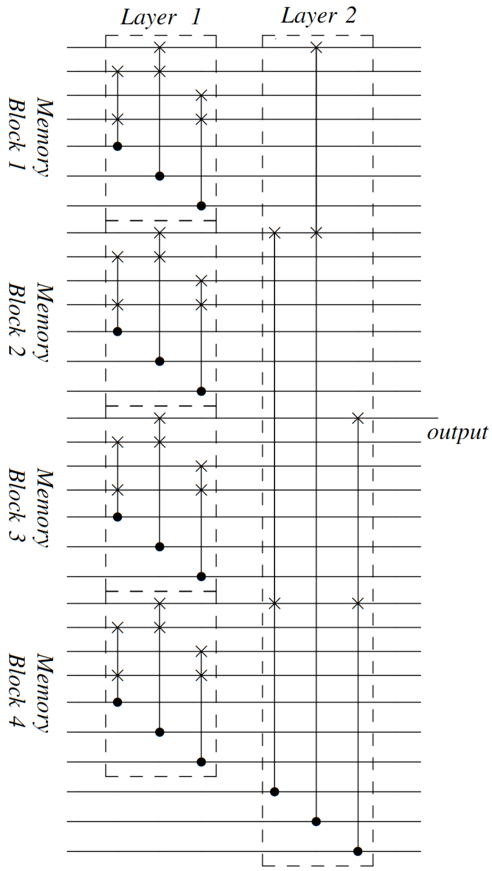

Figure 6: Concatenated Logarithmic Swapping Circuit.

5.1 Access Delay Complexity

We can find an optimum limit for the number of teleporters for the fastest access. Assuming the time delay for teleportation is $O(1)$, the ADC is due to either swappings from the original memory location to the teleporter, or swappings from the intermediate location to the destination. If we have N memory cells and n teleporters, there will be $\frac{N}{n}$ memory cells per memory block with a teleporter. Teleporters transport the target memory cell content to an intermediate location, which can loosely be considered as random access memory (RAM) for a quantum computer. The size of the intermediate location is equal to the number of teleporters, as one of the EPR pairs employed by the teleporters will be residing here.

5.1.1 Simple Swapping

In case of employing Simple Swapping in both memory blocks and intermediate locations, ADC then becomes $O(\frac{N}{n} + n)$, which is basically $O(max(\frac{N}{n}, n))$. This result is clearly minimized for $n = \sqrt{N}$. If we construct three intermediate locations with sizes n_1 and n_2, the complexity becomes

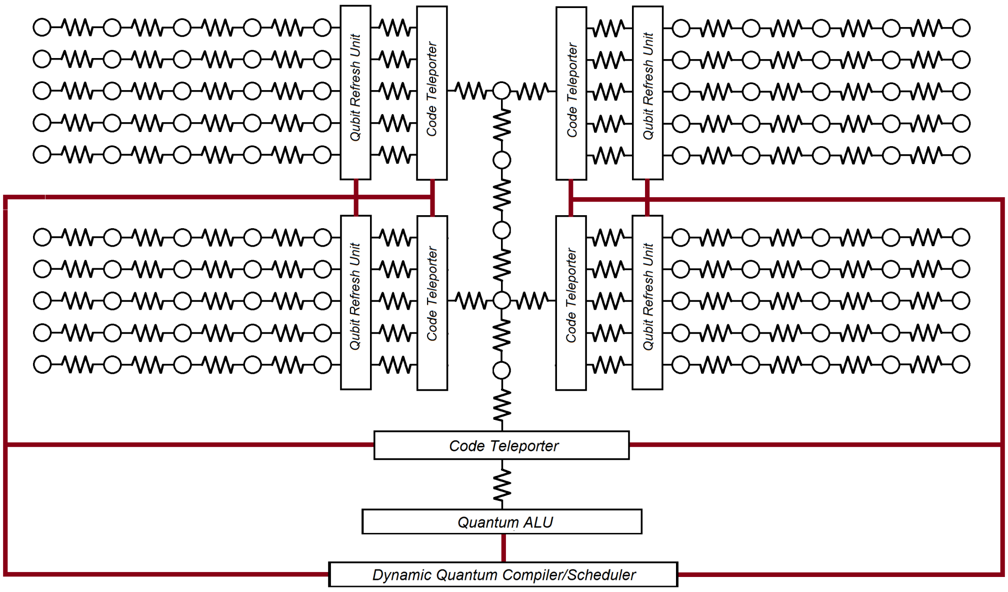

Figure 7: Quantum Memory Management with Swapping and Teleportation.

$O(max(\frac{N}{n_1}, \frac{n_1}{n_2}, n_2))$, and this result is minimized for $n_1 = N^{2/3}$ and $n_2 = N^{1/3}$. Generalizing this result for m intermediate locations, the complexity then becomes $O(\log_2(N))$.

5.1.2 Connected Swapping

If we employ Connected Swapping, ADC is definitely $O(m)$, as in each layer, the ADC is $O(1)$, hence, the number of layers provides us ADC.

5.1.3 Logarithmic Swapping

Assuming we employ Logarithmic Swapping for both the memory blocks and the intermediate locations, the ADC is $O(\log_2(\frac{N}{n})) + \log_2(n))$, as both of the operations had to be performed in order. Hence, ADC becomes $O(log_2(N))$ which is the same result obtained for the memory management systems with only Logarithmic Swapping. Furthermore, using more than one layer of intermediate locations will not change the result as the complexity will then become $O(\log_2(\frac{N}{n_1}) + \log_2(\frac{n_1}{n_2}) + \log_2(n_2))) = O(log_2(N))$, where n_1 and n_2 are the sizes of the intermediate locations.

5.2 Circuit Complexity

In Swapping and Teleportation memory management system, the CC is mainly the number of Fredkin Gates and teleporters in the main memory. In the previous section, we calculated the CC of any of the swapping techniques as $O(N)$. CC of any memory block will be $O(n)$, and since there will be $\frac{N}{n}$ such units, the complexity will be $O(N)$. For multiple intermediate layer structures, this result does not change as the higher layers will serve less memory blocks and will have a smaller degree of complexity.

Regardless of the swapping method and number of intermediate locations, the CC for N cells is always $O(N)$.

5.3 Maximum Connectivity

MC of Swapping and Teleportation architecture depends only on the swapping mechanism as teleportation introduces no further connectivity to the system.

5.3.1 Simple Swapping

In case of Simple Swapping, MC is simply 2, as each memory cell is connected to two other cells.

5.3.2 Connected Swapping

If we use Connected Swapping for the whole architecture, MC is determined by the largest memory block size. Since MC is n for a block of size n, MC for the architecture is $max(\frac{N}{n}, n)$. As in the use of Logarithmic Swapping this result is optimized for $n = \sqrt{N}$ and for an m intermediate locationed structure it becomes $\sqrt[m+1]{N}$.

5.3.3 Logarithmic Swapping

If we use Logarithmic Swapping, the memory block size, i.e., memory cells per teleporter determines the connectivity of the architecture. If there are n teleporters, then MC becomes $max(\log_2(\frac{N}{n}), \log_2(n))$, which is then minimized for $n = \sqrt{N}$. For multi layered architectures, the connectivity is minimized for a block size of $\sqrt[m+1]{N}$ for m intermediate locations. Therefore, MC becomes $\log_2(\sqrt[m+1]{N})$.

5.4 Ancillary Qubits

In order to teleport one qubit, an EPR pair is required, hence for every teleportation two ancillary qubits are used. If there are m intermediate locations, obviously the number of ancillary qubits required will be $2m$.

6. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the discussed memory management systems. As suggested in section II, four criteria will be used here, which are ADC, CC, MC and AQ. A summary is presented on Table 1.

First of all, we can deduce from Table 1 that the circuit complexity values for all these circuits are the same. Al-

Table 1: Comparison of Quantum Memory Management Systems

Architecture	ADC	MC	AQ
Simple Swapping	$O(N)$	2	0
Connected Swapping	$O(1)$	N	0
Logarithmic Swapping	$O(log_2(N))$	$log_2(N)$	0
Mixed Swapping	$O(\frac{N}{n}log_2(n))$	$log_2(n)$	0
Concatenated Connected S.	$O(m)$	$\sqrt[m]{N}+1$	0
Concatenated Logarithmic S.	$O(log_2(N))$	$log_2(\sqrt[m]{N})$	0
Simple S. and Teleportation	$O(\log_2(N))$	2	2m
Connected S. and Teleportation	$O(m)$	$\sqrt[m+1]{N}$	2m
Logarithmic S. and Teleportation	$O(\log_2(N))$	$\log_2(\sqrt[m+1]{N}))$	2m

though we can decrease the circuit size in some structures, the growth will still be linear.

We see that ADC is traded off with MC. Systems with larger maximum connectivity values offer less access delay complexity. However, these systems are harder to build for large memories. Nevertheless, they can be considered for small fast access memories, analogous to the cache memory of a classical microprocessor.

We also realize that, apart from techniques using Connected Swapping, the best ADC value achieved is $O(log_2(N))$. However depending on the number of intermediate locations, we can reduce ADC with a factor of m^{-1}. Investigating the MC values, the best value is $O(log_2(n))$ for Mixed Swapping.

Another important observation from Table I is that techniques with teleportation do not provide any additional advantage over Logarithmic, Mixed or Concatenated Logarithmic Swapping techniques. In addition, the ancillary qubits are required only in teleportation. The required AQ number makes multi-layered Swapping and Teleportation system unfeasible, as that many ancillary qubits will be required to write it in the memory as well as reading it. In these architecture schemes, EPR pairs for teleportation has to be constantly created and distributed to memory locations. This fact reduces the feasibility of these techniques even further.

We conclude that Logarithmic, Mixed and Concatenated Logarithmic Swapping techniques are all possible, as these architectures have both smaller ADC and MC.

7. CONCLUSION

Quantum Computers, due to the recent developments in nanotechnology, are almost in our grasp. In this work, we focused on the bottleneck of quantum computers, the memory. We have suggested novel quantum memory management structures and analyzed them in the light of four important factors, Access Delay Complexity, Circuit Complexity, Maximum Connectivity and Ancillary Qubit use.

Although these architectures are well suited to transfer qubits from one location to another, a total quantum memory management system would require classical memory management systems integrated to the quantum memory management systems as well. During accessing of any qubit, many will change their location hence a track of all these operations should be kept in classical memory cells. As a future work, these architectures should be refined and the classical memory management complexities included as well.

ACKNOWLEDGEMENT

This work was supported in part by the Turkish Scientific and Technical Research Council (TUBITAK) under grant #109E257.

8. REFERENCES

[1] L. DiCarlo, J. M. Chow, J. M. Gambetta, and et al. Demonstration of two-qubit algorithms with a superconducting quantum processor. *Nature*, 260:240–244, 2009.

[2] L. Grover. A fast quantum mechanical algorithm for database search. *Proc. 28th Ann. ACM Symp. Theory of Computation*, pages 212–219, 1996.

[3] H. Häffner, C. F. Roos, and R. Blatt. Quantum computing with trapped ions. *arXiv:quant.ph/0809.4368v1*, 2008.

[4] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2002.

[5] M. Oskin, F. T. Chong, and I. L. Chuang. A practical architecture for reliable quantum computers. *IEEE Computer*, 35:79–87, 2002.

[6] P. Rabl, D. DeMille, J. M. Doyle, and et al. Hybrid quantum processors: Molecular ensembles as quantum memory for solid state circuits. *Phys. Rev. Lett*, 97:1869–76, 2006.

[7] P. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. *Proc. 35th Ann. Symp. Foundations of Computer Science*, page 124, 1994.

[8] V. Singh, S. J. Bosman, and S. B. H. et al. Optomechanical coupling between a multilayer graphene mechanical resonator and a superconducting microwave cavity. *Nature Nanotechnology*, 9:820–824, 2014.

[9] M. K. Thomsen, R. Gluck, and H. B. Axelsen. Reversible arithmetic logic unit for quantum arithmetic. *Journal of Physics A: Mathematics and Theoretical*, 42:2002, 2010.

[10] A. Vetteth, K. Walus, G. A. Jullien, and et al. Quantum dot cellular automata carry-look-ahead adder and barrel shifter. *IEEE Emerging Telecommunications Technologies Conference, Pharmacoeconomics*, 24:1055–68, 2002.

[11] J. S. Wenzler, T. Dunn, and T. e. a. Toffoli. A nanomechanical fredkin gate. *Nano Lett.*, 14:89–93, 2014.