International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 1 (2018) pp. 375-384
© Research India Publications. http://www.ripublication.com

The Era of Expeditious NanoRAM-Based Computers

Enhancement of Operating System Performance in Nanotechnology Environment

Mona Nabil ElGohary
PH.D Student, Computer Science Department
Faculty of Computers and Information, Helwan University, Cairo, Egypt.

[1]ORCID: 0000-0002-1996-4673

Dr. Wessam ElBehaidy
Assistant Professor, Computer Science Department,
Faculty of Computers and Information, Helwan University, Cairo, Egypt.

Ass. Prof. Hala Abdel-Galil
Associative Professor, Computer Science Department
Faculty of Computers and Information, Helwan University, Cairo, Egypt.

Prof. Dr. Mostafa-Sami M. Mostafa
Professor of Computer Science
Faculty of Computers and Information, Helwan University, Cairo, Egypt.

Abstract

The availability of a new generation of memory that is 1000 times faster than traditional DDRAM which can deliver terabytes of storage capacity, and consumes very little power, has the potential to change the future of the computer's operating system. This paper studies the different changes that will arise on the operating system functions; memory management and job scheduling (especially context switch) when integrating NanoRAM into the computer system. It is also looking forward to evaluating the possible enhancements of computer's performance with NanoRAM.

Keywords: Nanotechnology, NanoRAM, Response Time, Nanoscale Transistor, Virtual Memory, Storage Hierarchy, Disk Swapping, Memory Allocation, Memory Fragmentation, Address Space, Process State Diagram, Context Switch, Context Switch Time

INTRODUCTION

Carbon Nanotube (CNT) is an excellent example of true nanotechnology: it has a very small diameter (less than 100 nanometers and can be as thin as 1 or 2 nm). It is a material that can be manipulated chemically and physically in very useful ways. It has important properties, such as extraordinary heat conductivity, electrical conductivity, and mechanical properties, and high length-to-diameter ratio. These properties make carbon nanotubes very useful in a wide range of applications such as chemical processing, biology, electronics, energy management, and many other fields to be developed [1].

Nantero, in collaboration with Fujitsu, uses carbon nanotubes manufactured as an alternative to silicon chips which is used in the traditional DDRAM to design a NanoRAM .This NanoRAM will achieve a 1,000x performance increase [2]. They announced that by 2018 will produce the first NanoRAM.

This new NanoRam has many excellent properties that would make an excellent replacement for the current DDRAM: being non-volatile, its large capacity, high speed read / write cycles. All the properties are introduced in the next section.

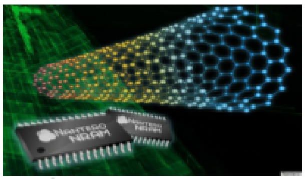
By replacing this NanoRAM instead of DDRAM in the CPU, this will affect the functionality of the operating system; such as main memory management, virtual memory, job scheduling, secondary storage management; and thus the efficiency of the computer system

The aim of this paper is to present the opportunities offered by NanoRAM technology to design more powerful and efficient operating system. In the following sections, we first discuss the properties of NanoRAM and then explain some components of the operating system and functions that can be reconstructed based on the use of NanoRAM.

NanoRAM

Carbon nanotubes have some unique structural properties. Such as massive storage capacity and transfer speed, and electrical properties and its needs for very low power to operate. This makes them ideal for building a new generation of computer storage. It also has the best ability to conduct electrical signals than any other material known to scientists now [3], which leads for a very fast speed of data transfer.

Nantero's NanoRAM has many remarkable advantages; low cost, very fast read speed (10^{15} cycles/sec) and write speed (10^{12} cycles/sec) [2], high reliability which makes it retains memory for more than 1000 years at 85^{o} C or 10+ years at 300^{o} C degrees. Furthermore, it is compatible with CMOS components, which means that no types of new equipments are required. It needs zero power consumption in the standby mode and scalability limitless because its dimensions may reach <5 nm in the future, and also it has huge storage capacity up to 1 terabytes [4] [5]. Figure (1) summarizes NanoRAM advantages.



Fast Read & Write

10^{12} Write / Sec, 10^{15} Read / Sec

Low Cost

High Reliability

Retains memory for > 1K years at 85° C or 10+ years at 300° C degrees

Works in CMOS fabs

No new equipments needed

Low Power

Zero in standby mode

Limitless Scalability

< 5 nm in future

Huge Capacity

Up to 1TeraByte

Figure 1. NanoRAM Advantages

MEMORY MANAGEMENT WITH NANORAM

In this section, some changes to the operating system components (memory management, secondary storage) are introduced when the NanoRAM is integrated into the computer.

A. Architecture

Define As figure (2) illustrates, The CPU must fetch programs from the RAM memory. Being volatile memory is the problem with RAM. Thus the hard disk, CD's and DVD's are used as a non-volatile media to store programs and data [6].

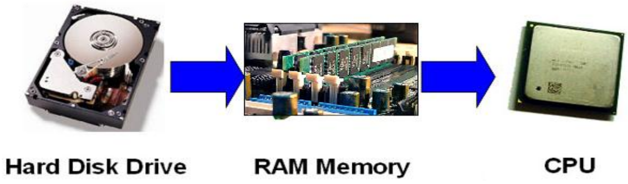


Figure 2. Traditional Computer Architecture

With NanoRAM the problem can be solved, the NanoRAM has a huge capacity (may reach 1 Terabyte as mentioned in section 2) and it is nonvolatile. So, The most direct and straightforward option is to simply replace disks with NanoRAM. The optical disks (CD, DVD) may be needed for long term backup or moving data between computers. The NanoRAM- based computer architecture is shown in figure (3).

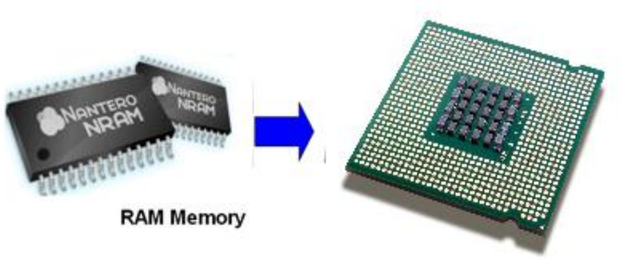


Figure 3. NanoRAM-based Computer Architecture

B. Storage Hierarchy

As shown in figure (4), there are many levels of storage in the operating system. They are classified based on the speed and the size.

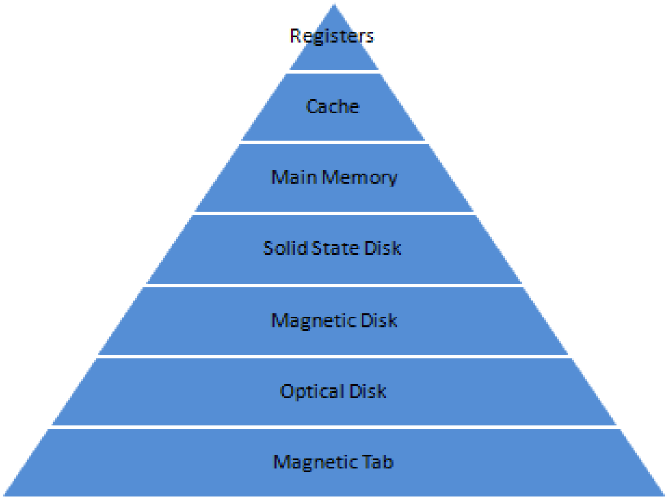


Figure 4. Traditional Storage Hierarchy

The next table [7][8] illustrates the access time of the different storage types in a computer with **Intel Pentium i7 6700 Processor** with 4GHZ speed [15] and **Seagate Backup Plus hard disk** with 220 MB/ Sec speed.

Table 1: Devices Access Time

Storage Type	Access Time
Processor Registers	0.25 ns
Cache	7 – 20 ns
DDRAM3	1 ns
Hard Disk	4540000 ns

The change in the operating system architecture and the super speed of the NanoRAM (access time = 0.001 ns) will cause some changes in the storage levels. Figure (5) shows the changes.

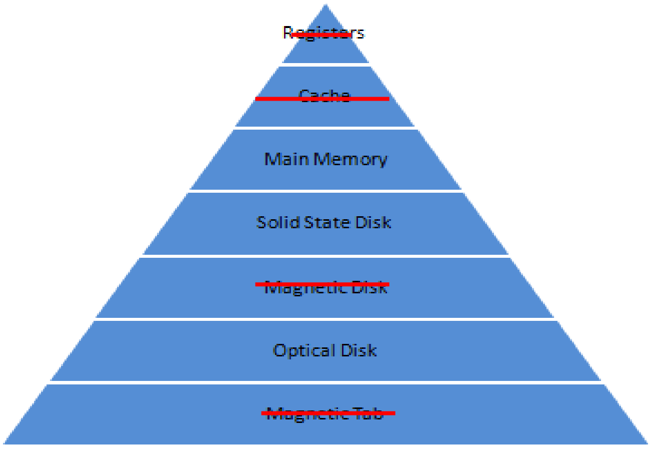


Figure 5. Storage Hierarchy with NanoRAM

C. Virtual Memory

The Today's computers have 2 levels of storage: slow, massive capability, sturdy storage device, and fast, smaller capability, volatile primary memory. Figure (6) shows that a computer can address more memory than the amount physically installed on the system. This extra memory is actually called **virtual memory** and it is a section of a hard disk that's set up to emulate the computer's RAM [9].

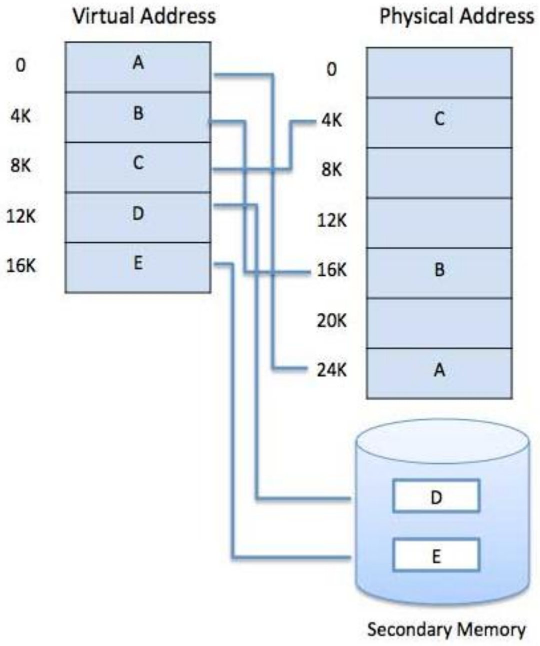


Figure 6. *Virtual Memory in Traditional Computer*

According to [10], virtual memory can be defined as "A storage allocation schema in which secondary memory can be addressed as though it were part of main memory".

The existence of large-scale available persistent memory (NanoRAM) could also be a lot of helpful than traditional memory systems. As a result of there's only 1 level of storage (no hard disk), virtual memory may be canceled, figure (7), for computers with one level of sturdy, quick storage.

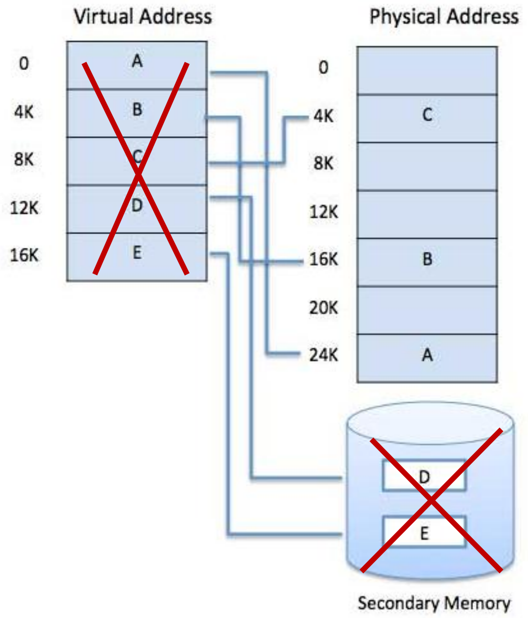


Figure 7. NanoRAM-based Computer System.

D. Memory-Disk Swapping

Although the process must be in memory to be executed, it can be temporarily swapped out of main memory (RAM) to a secondary storage (hard disk) and then swap in again into memory for continuing execution as shown in figure (8). The main benefit of swapping is to increase the degree of multiprogramming in a system by making the total physical address space of all processes exceeds the real physical memory of the system [11].

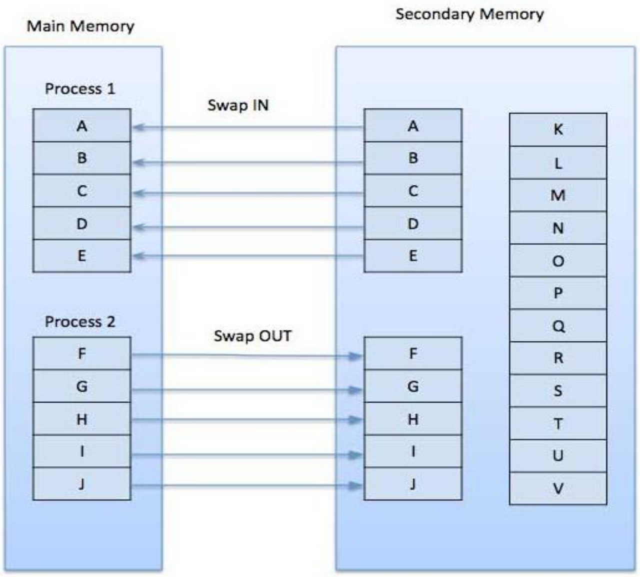


Figure 8. Memory-Disk Swapping [9]

Having a single storage (NanoRAM) on the computer makes the term "memory - disk swap" meaningless.

E. Multiple Address Spaces

Traditional processors use a virtual addressing as its addressing method [12]. That means there are two types of addresses; logical address / virtual address which is the address created by the operating system, and a physical address which is seen by the memory unit and loaded into the memory address register of the memory.

The logical address space is the group of all logical addresses generated by a CPU while the group of all physical addresses corresponding to these logical addresses is the physical address space. In the execution-time address-binding scheme, the logical and physical address spaces are different. As figure (9) shows the run-time mapping from logical to physical addresses is done by a hardware device called the memory-management unit (MMU) [11].

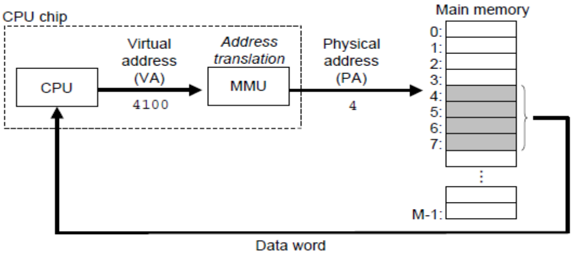


Figure 9. Memory Management Unit[12]

Today's computer systems provide different address spaces for each process whereas, with NanoRAM, the computer has a single, very fast, huge capacity storage that can be viewed as a single-address-space for all processes. But a protection system might need to be reconsidered.

F. Memory Allocation

The operating system usually allocates many user processes to be in main memory simultaneously. Therefore it needs to consider how to assign the available memory to the processes that are in the input queue waiting to be brought into memory [11] [13].

There are two methods for memory allocation; contiguous allocation and non- contiguous allocation.

Contiguous memory allocation when each process is located in a single portion of memory (fixed sized or variable sized) as a single chunk that is contiguous using first fit or best fit or worst fit option [11], While **non- contiguous allocation** allows the process to be non-continuously (paging, segmentation) allocated in physical memory whenever that memory is available and according to the program execution needs [14].

NanoRAM based system will treat its main memory as a single, fast and huge storage So, no longer needs to non-contiguous data allocation. NanoRAM might allocate the processes contiguously in the NanoRAM with the best fit option.

G. Memory Fragmentation

Memory fragmentation can be internal as well as external; the memory allocated to a process may be slightly larger than the requested memory as figure 10 shows. The difference between these two numbers is **internal fragmentation** (figure 10(A)) which is the unused memory that is internal to a partition. **External fragmentation** (figure 10(B)) exists when there is enough total memory space to satisfy a request but the available spaces are not contiguous: storage is fragmented into a large number of small holes [10] [11].

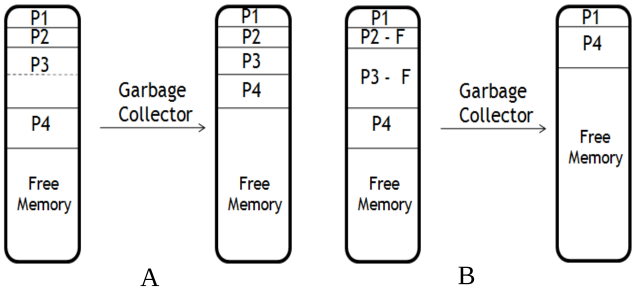


Figure 10. A) Internal Memory Fragmentation B) 0 Externa Memory Fragmentation

Because NanoRAM may implement the contiguous allocation architecture fragmentations may be needed. Fragmentation over time Internal for fixed sized and external for variable sized

SCHEDULING WITH NANORAM

This section introduces the main changes in process scheduling in NanoRAM- based computer system.

A. Process State Diagram

The As shown in the state diagram figure (11), the applications reside the hard disk and when it is launched, the long term scheduling moves the application to the hard disk job pool with the new state. After t this step, it moves as an active process to the RAM ready queue with the ready state. After the short term scheduling chooses the processes to be run by the CPU, the process moves to the running queue with running state. The medium term scheduling may be used in case of interrupt or disk swapping, long interrupts swap out the process to the hard disk job pool with the new state. While short interrupts move the process to the RAM ready queue with the ready state [12].

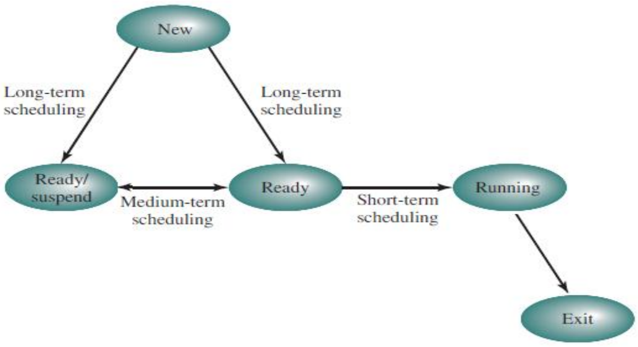


Figure 11. Traditional Process State Diagram

With NanoRAM, every application that is resident on a local computer (on NanoRAM) could be perpetually “active” (ready to run) , The concept of launch (Long Term Scheduling) would no longer need ,and instead applications are directly scheduled by short term scheduling whenever some form of input (user, device, or other events) is directed towards them. In addition, the long and short term interrupts will change the running state to ready state (no new state). The NanoRAM –based computer’s process state diagram is shown in figure 12.

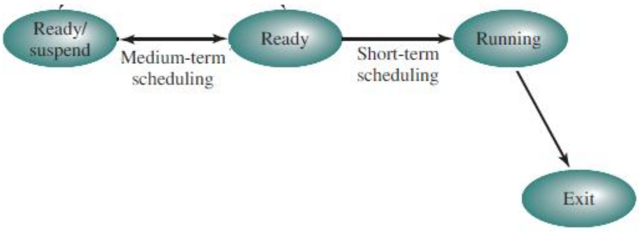


Figure 12. Process State Diagram with NanoRAM

B. Process Context Switch

The context switching process would seem to be straightforward. At some time a running process is interrupted and the operating system assigns another process to the running state and turns control over that process.

When an interrupt occurs, the system needs to save the current **context** of the process running on the CPU so that it can restore that context when its processing is done, essentially suspending the process and then resuming it. The context is represented in the PCB of the process. It includes the value of the CPU registers, the process state, and memory-management information. Generically, we perform a state save of the current state of the CPU, be it in kernel or user mode, and then a state restores to resume operations [10].

Switching the CPU to another process requires performing some steps, these steps are shown in figure13.

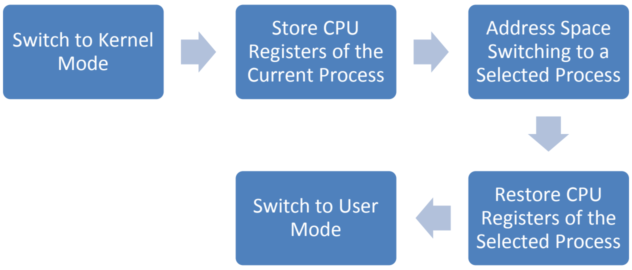

Switch to Kernel Mode
Store CPU Registers of the Current Process
Address Space Switching to a Selected Process
Restore CPU Registers of the Selected Process
Switch to User Mode

Figure 13. Context Switch Steps

METHODS

Considering these three computer systems with different processors and RAMs:

1. **The first system (S1)** which has a traditional processor with traditional RAM (TPTR); Intel i7-6700 processor with 4GHZ speed and 12 nm size [15] integrated with a DDRAM3. Also, it has a Seagate Backup Plus hard disk with data transfer speeds up to 220 MB per second [7].
2. **The second system (S2)** which has the same previous traditional processor with NanoRAM (TPNR). NanoRAM has access speed of 1012cycle / second and access time equals 0.001 ns
3. **The third system (S3)** which has a Nano-processor with NanoRAM (NPNR).

For these three computer systems, we formulate an equation to calculate the context switch time for each and the total response time.

a) Context Switch Time

According to figure 13, the context switch time (M) is based on 4 parameters; switch to kernel mode, storing/restoring CPU registers, switching address space, and switch to user mode.

So, the formulated equation for the context switch time for the first system (S1) will be:

Context Switch Time 1 (M1) = SwKrnl + RegSR + SwAddr + SwUsr (Eq. 1)

Where:

- **M1 :** context switch time for S1 system
- **SwKrnl :** switch to kernel mode
- **RegSR :** storing / restoring CPU registers
- **SwAddr :** Switching address space
- **SwUsr :** switch to user mode

Table 2 illustrates the time of context switch steps using S1. By replacing the DDR with NanoRAM, CPU registers’s values can be moved to NanoRAM instead of processor registers (0.001 ns for NanoRAM and 67 ns for processor registers). Also, the address space switching will be just a direct access to the first memory location of the selected process or to the next address to which the process was interrupted (0.001 ns instead of 184 ns).

Table 2. Context Switch Steps Access Time

Task	Time
Switch to Kernel Mode (SwKrnl)	481 ns
Store / Restore CPU Registers(RegSR)	67 ns
Switching Address Space (SwAddr)	184 ns
Switch to User Mode (SwUsr)	330 ns
Access Time for NanoRAM (NTAccess)	0.001 ns
Hard Disk Access Time (DAT)	4540000 ns

So, the formulated equation for the context switch time for the second system (S2) will be:

Context Switch Time 2 (M2) = SwKrnl + 2*NTAccess + NTAccess + SwUsr (Eq. 2)

Where:

- **NTAccess:** access time for NanoRAM.

If there is a Nano- based processor in the computer system with 1000x performance enhancement [14] integrated with a NanoRAM, the formulated equation for context switch time (M3) equation for the third system will be:

Context Switch Time 3 (M3) = SwKrnl/1000 + 2*NTAccess + NTAccess + SwUsr / 1000 (Eq.3)

b) Total Response Time

The total response time is affected by the context switch time and the disk – memory swapping time (which involves moving some or all processes from main memory to hard disk).

Thus, we can formulate the total response time (TmResponse) in general as:

$$\textbf{Total Response Time for } S_i \textbf{ (TmResponse)} = \sum \textbf{ProExecTm} + \textbf{SwNum} * M_i * \frac{(100-\text{ProPerc})}{100} + \textbf{SwNum} * (M + 2 * \textbf{TmDiskSw}) * \frac{\text{ProPerc}}{100} \quad \textbf{(Eq.4)}$$

Where:

- **ProExecTm:** process execution time.
- **SwNum:** number of switches.
- **M_i:** context switch time for system i.
- **ProPerc:** the percentage of the processes that need disk swapping
- **TmDiskSw:** disk swapping (disk access time)

EXPERIMENTA RESULTS

There are two variable parameters considered when the total response time is calculated:

i. The number of processes that the computer system runs. In this paper, five cases ($Case_i$: i=1, 2..., 5) are used, each one contains a different number of processes (ProNum=10, 20, 30, 40, and 50 respectively). Each case ($Case_i$) is being run with a different number of switches (SwNum=1, 2, 3, 4, and 5 respectively) on the three systems (S1, S2, and S3) with their different context switch values (M1, M2. And M3).

ii. The percentage of the processes that need disk swapping. In this paper, five different swapping percentages are used (ProPerc=10%, 20%, 50%, 80%, and 100% respectively).

For applying these previously formulated equations, consider the time taken for each task in Table (2)

a) The context switch time for our three systems , by applying (1), (2), and (3) respectively, is:

M1= 481 + 67 + 184 + 330 = **1062 ns**

M2= 481 + 3 * 0.001 + 330 = **811.003 ns**

M3=0.481 + 3 * 0.001 + 0.330 = **0.814 ns**

Table 3 presents the total execution time (**TET**) for each case measured in nanoseconds (∑**ProExecTm** in each case).

Table 3. Execution Time for input Cases

Case	TET(S1) ns	TET(S2) ns	TET (S3) ns
1	235000	235000	235
2	773000	773000	773
3	2152000	2125000	2125
4	2787000	2787000	2787
5	3198000	3198000	3198

S1 and S2 are measured in the microsecond, but S3 is measured in nanoseconds. So, S1 and S2 multiplied by 1000 to be in nanosecond scale.

The response time is calculated using (4) with different swapping percentages (P=10%, P=20 %, P=50%, P=80%, P=100%) and with a different number of switches (SwNum =1, 2, 3, 4, and 5) for the three systems. Then the average of all execution times of all cases (case1, case2, case3, case4, case5) with each number of switches (SwNum =1, 2… 5) is calculated using (5).

Average of Total Response Times (AvgTmResponse) = ∑ (TmResponse (Case $_i$)/ 5 (Eq.5)

- **Average of total response times with 10% disk swapping processes**

Using M1, M2, and M3, (4) with **ProPerc** = 10%, and (5), the results of response time **(AvgTmResponse)** for S1, S2, and S3 will be:

Table 4. Average Response Time with ProPerc=10%

SwNum	AvgTmResponse (M1=1062) ns	AvgTmResponse (M2=811.003) ns	AvgTmResponse (M3=0.814) ns
1	273,806,2	182,441,1	1,830
2	3647124	1825222	1825
3	4556186	1826033	1826
4	5465248	1826844	1827
5	6374310	1827655	1828

The following figure shows the average of total response times values with ProPerc=10%, M1, M2, and M3.

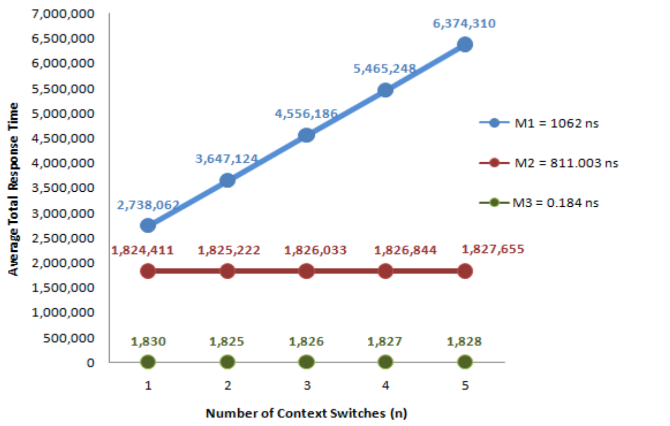


Figure 14. Average Response Time (ppoPerc=10%) for M1 , M2 , M3

The next figure shows the response time enhancement percentage between the three response times (with 10% disk swapping); the first column is the time enhancement percentage between the first system (S1) and the second system (S2).While the second column is the time enhancement percentage between the first system (S1) and the third system (S3).

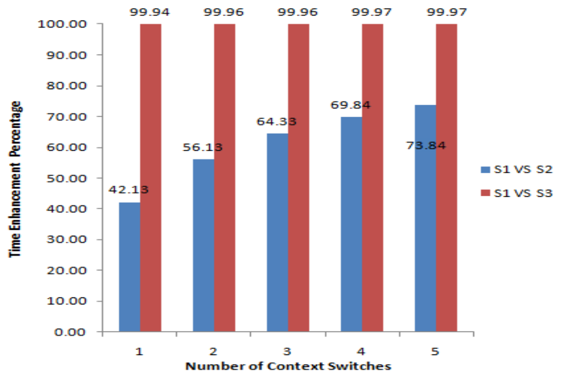


Figure 15. Time Reduction Enhancement for the three cases with ProPerc=10%

- **Average total response time with 20% disk swapping processes**

Using M1 , M2 , M3 , (4) with **ProPerc**= 20%, and (5) , the results of response time **(AvgTmResponse)** for S1 , S2 , and S3 will be :

Table 5. Averaage Response Time with ProPerc=20%

SwNum	AvgTmResponse (M1=1062) ns	AvgTmResponse (M2=811.003) ns	AvgTmResponse (M3=0.814) ns
1	3,646,062	1,824,411	1,830
2	5,463,124	1,825,222	1,825
3	7,280,186	1,826,033	1,826
4	9,097,248	1,826,844	1,827
5	10,914,310	1,827,655	1,828

The following figure shows the average of total response times values with ProPerc=20%, M1, M2, and M3.

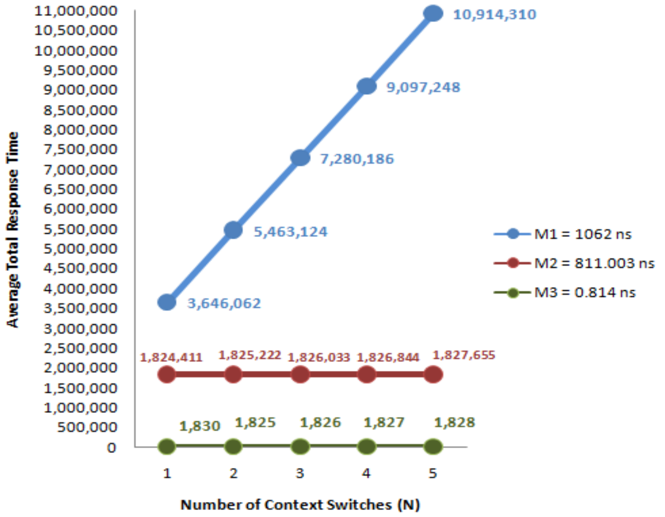


Figure 16. Average Response Time (ProPerc=20%) for M1 , M2 , M3

The next figure shows the response time enhancement percentage between the three response times (with 10% disk swapping); the first column is the time enhancement percentage between the first system (S1) and the second system (S2).While the second column is the time enhancement percentage between the first system (S1) and the third system (S3).

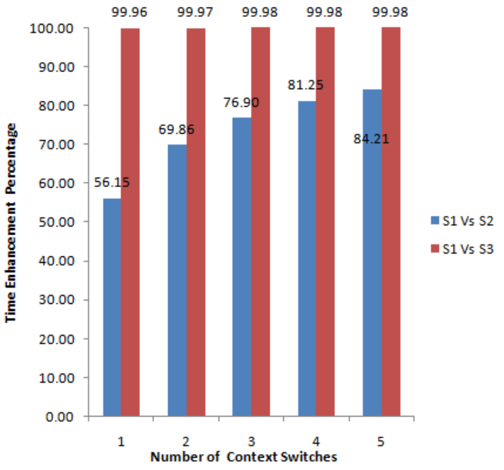


Figure 17. Time Reduction Enhancement for the three cases with ProPerc=20%

- **Average total response time with 50% disk swapping processes**

Using M1, M2, and M3, (4) with ProPerc = 10%, and (5), the results of response time (AvgTmResponse) for S1, S2, and S3 will be:

Table 6. Averaage Response Time with ProPerc=50%

SwNum	AvgTmResponse (M1=1062) ns	AvgTmResponse (M2=811.003) ns	AvgTmResponse (M3=0.814) ns
1	6,370,062	1,824,411	1,830
2	10,911,124	1,825,222	1,825
3	15,452,186	1,826,033	1,826
4	19,993,248	1,826,844	1,827
5	24,534,310	1,827,655	1,828

The following figure shows the average of total response times values with ProPerc=50%, M1, M2, and M3.

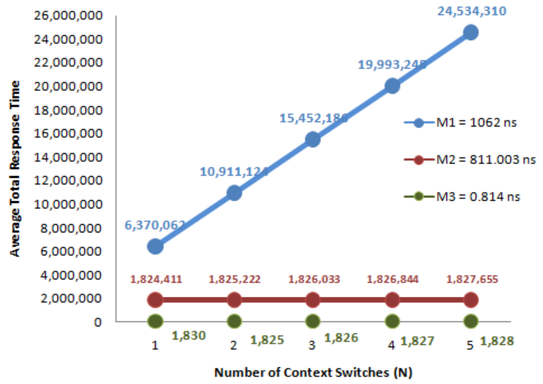


Figure 18. Average Response Time (ProPerc=50%) for M1 , M2 , M3

The next figure shows the response time enhancement percentage between the three response times (with 10% disk swapping); the first column is the time enhancement percentage between the first system (S1) and the second system (S2).While the second column is the time enhancement percentage between the first system (S1) and the third system (S3).

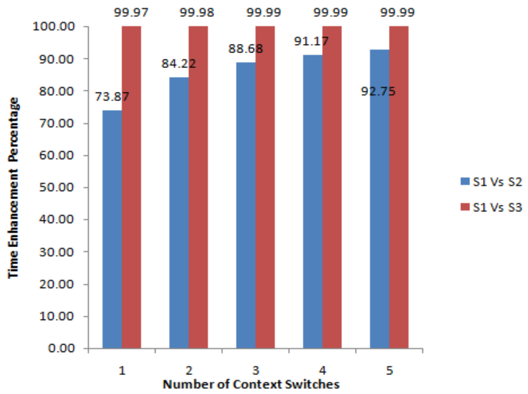


Figure 19. Time Reduction Enhancement for the three cases with ProPerc=50%

- **Average total response time with 80% disk swapping processes**

Using M1, M2, and M3, (4) with **ProPerc** = 80%, and (5), the results of response time **(AvgTmResponse)** for S1, S2, and S3 will be:

Table 7. Averaage Response Time with ProPerc=80%

SwNum	AvgTmResponse (M1=1062) ns	AvgTmResponse (M2=811.003) ns	AvgTmResponse (M3=0.814) ns
1	9,094,062	1,824,411	1,830
2	16,359,124	1,825,222	1,825
3	23,624,186	1,826,033	1,826
4	30,889,248	1,826,844	1,827
5	38,154,310	1,827,655	1,828

The following figure shows the average of total response times values with ProPerc=80%, M1, M2, and M3.

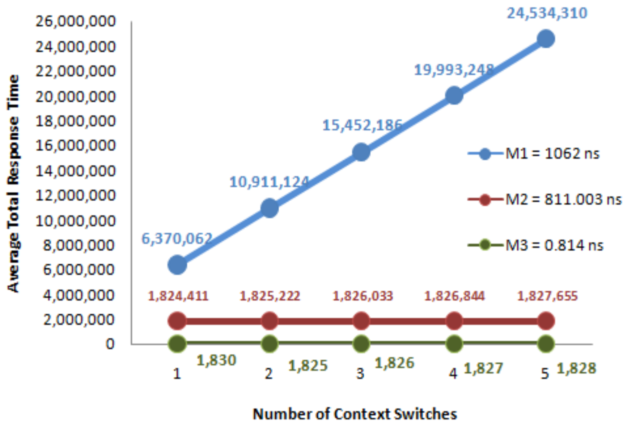


Figure 20. Average Response Time (ProPerc=80%) for M1 , M2 , M3

The next figure shows the response time enhancement percentage between the three response times (with 10% disk swapping); the first column is the time enhancement percentage between the first system (S1) and the second system (S2).While the second column is the time enhancement percentage between the first system (S1) and the third system (S3).

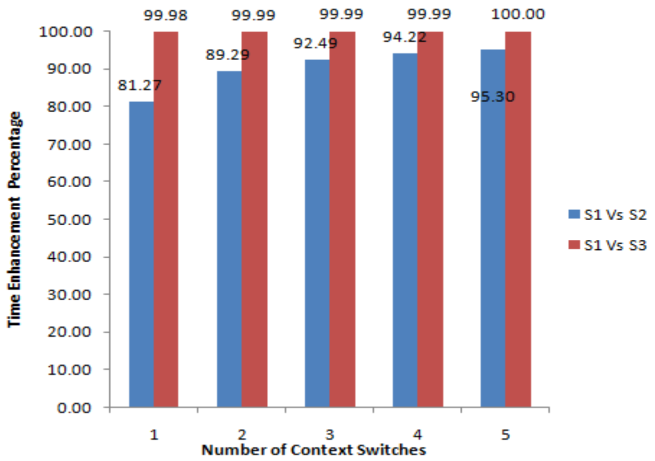


Figure 21. Time Reduction Enhancement for the three cases with ProPerc=80%

- **Average total response time with 100% disk swapping processes**

Using M1, M2, and M3, (4) with **ProPerc** = 100%, and (5), the results of response time **(AvgTmResponse)** for S1, S2, and S3 will be:

Table 8. Averaage Response Time with ProPerc=100%

SwNum	AvgTmResponse (M1=1062) ns	AvgTmResponse (M2=811.003) ns	AvgTmResponse (M3=0.814) ns
1	10,910,062	1,824,411	1,830
2	19,991,124	1,825,222	1,825
3	29,072,186	1,826,033	1,826
4	38,153,248	1,826,844	1,827
5	47,234,310	1,827,655	1,828

The following figure, figure 22, shows the average of total response times values with P=100%, M1, M2, and M3.

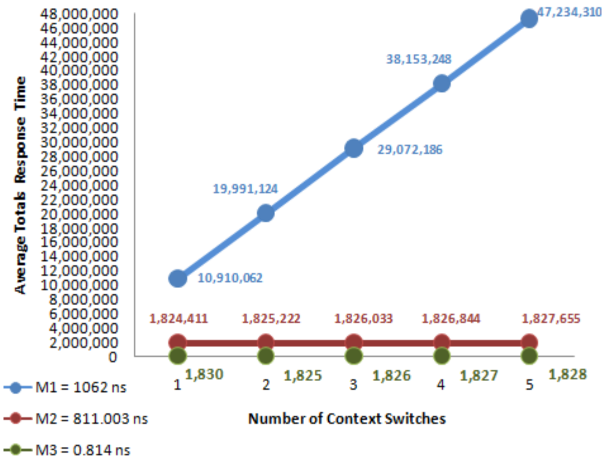


Figure 22. Average Response Time (ProPerc=100%) for M1, M2 , M3

The next figure shows the response time enhancement percentage between the three response times (with 10% disk swapping); the first column is the time enhancement percentage between the first system (S1) and the second system (S2).While the second column is the time enhancement percentage between the first system (S1) and the third system (S3).

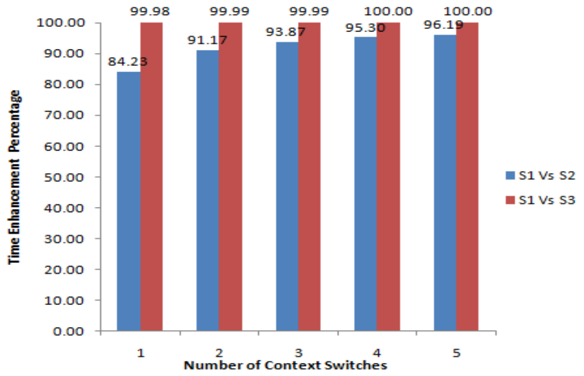


Figure 23. Time Reduction Enhancement for the three cases with ProPerc=100%

DISCUSSION

The researchers use a dataset of 50 processes which arranged in 5 cases ; case 1 , case2 , case 3 , case 4 , and case 5 with 10, 20 , 30 , 40 , and 50 processes respectively.

The context switch time was calculated for three different computer systems (S1, S2, and S3) resulting three values of context switch time (M1, M2, and M3).

The response time was also calculated with the three context switch time values (M1, M2, M3) with different number of context switches (SwNum=1 ,2,3,4,5) then the average of response times with each SwNum is calculated and the percentages of time reduction (enhancement) are figured out in figures 15 , 17, 19, 21 , 23.

In the comparison of S1 and S2 (the blue columns in the figures 15, 17, 19, 21, 23) In the case of 10% disk swapping processes was the lowest improvement value equaled to 42.1% less time and the highest improvement percentage was 73.8, with ProPerc= 50%, the lowest improvement percentage was 73.87% and the highest time reduction percentage was 92.75% and with ProPerc=100 %, 84.23% was the minimum enhancement percentage and it reached 96.19%.

Comparing S1 and S3 (the red columns in the figures with P=10%, the highest value was 99.97% and the lowest was 99.94, turning to ProPerc=50%, the best percentage was 99.99% and the worst value was 92.75%, with ProPerc=100% the enhancement percentage reached 100%.

The researcher found that at the NanoRAM based computer system (with the i7 processor or with Nano processor), as the value of the percentage of disk swapping processes (ProPerc) increases, its rate of time improvement increases.

The number of switches (SwNum) affected also the percentage of improvement, In the comparison of S1 and S2 (the blue columns in the figures) In the case of the SwNum=1 , the lowest improvement value equaled to 42.1% less time (with ProPerc=10%) and the highest percentage was 84.23% (with ProPerc=100%) , with SwNum=5 the lowest improvement percentage was 73.8% (with ProPerc=10%) and the highest time reduction percentage was 96.19% (with ProPerc=100%).

Also when comparing S1 and S3 (the red columns in the figures 15, 17, 19, 21,23), with SwNum=1 , the highest value was 99.98%(with ProPerc=100%) and the lowest was 99.94%(with ProPerc=10%), turning to SwNum=5 , the best percentage was 100% (with ProPerc=100%) and the worst value was 99.97%(with ProPerc=10%)

The researcher found that at the NanoRAM based computer system (with the i7 processor or with Nano processor), as the value of the number of context switches (SwNum) increases, its rate of time improvement increases.

SUMMARY

To sum up, table (9) compares the general properties of the traditional DDR with NanoRAM, and table (10) compares the DDRAM-based computer system with the NanoRAM-based computer system in term of the basic functions of the operating system.

Table 9. Traditional RAM VS NanoRAM (GENERAL)

Criterion	Traditional RAM	NanoRAM
Size	32 GB allowed to 64 GB	256 GB allowed to 1 TB
Ram Rate (Bandwidth)	DDR4 3200 : 25.6 GB/Sec	0.8 TB/Sec= 800 GB/Sec
Heat Resistance	Low –Medium	High : 10+ years at 300 degree 1000 years at 85 degree
Read / Write Cycles	10^9 Write per second , 10^9 Read per second	10^{12} Write per second , 10^{15} Read per second
Program Erase Cycles	5000 – 8000 , Best 100,000 = 10^5 per second	100 billionth = 10^{11} Cycles per second

Table 10. NanoRAM-based computer system vs DDRAM-based computer system

Criterion	Traditional RAM	NanoRam
Storage Hierarchy	Registers , Cache , Main memory , Solid State Disk , Magnetic Disk , Optical Disk , Magnetic Tab.	Main memory , Solid State Disk , Optical Disk
Secondary Storage	Yes , Magnetic Hard Disks	No Need
Virtual Memory	Yes , With Hard Disk	No Need
Memory Management Unit	Relocation Register , Base – Limit TLB : Translation Lookaside buffer	Base – Limit TLB : Translation Lookaside buffer
Disk Swapping	Yes	No
Memory Allocation	Continuous , Non Continuous Variable Size , Fixed Size First Fit , Best Fit , Worst Fit	Continuous Variable Size Best Fit
Segmentation	Yes	No
Paging	Yes	No
Fragmentation	Internal , External	Internal , External

RFERENCES

[1] Mona Nabil ElGohary, Mostafa-Sami M. Mostafa, Hala AbdElgalil, Wessam ElBehaidy, Nov2017, "From Silicon Chips to Carbon Nanotubes ", International Journal of Computer Application

[2] Lucas Mearian, Aug2016, "Fujitsu making DRAM killer with 1,000 X performances boost". [Online].Available: https://www.computerworld.com/article/3113822/data-storage/fujitsu-touts-dram-killer-with-1000x-performance-boost.html

[3] Chris Mellor , Aug2016, "Deep inside Nantero's non-volatile carbon nanotube RAM tech". [Online].Available: https://www.theregister.co.uk/2016/08/31/nram_dev_nantero_signs_fujitsu/

[4] Nantero Team, 2017 "Technology Advancements in Nanotechnology". [Online]. Available: http://nantero .com/technology/

[5] Brian Wang , , August 2017 ,"Carbon nanotube nonvolatile NRAM memory 1000 times faster than Flash will be commercially released by the end of 2018 by Nantero and Fujitsu". [Online].Available: https://www.nextbigfuture.com/2016/08/carbon-nanotube-nonvolatile-nram-memory.html

[6] Gabriel Torres, Sept 2007, "How the Cache Memory Works". [Online]. Available: http://www. hardwaresecrets.com/how-the-cache-memory-works/

[7] Best External Hard Drives 2017, 2017. [Online].Available: http://www.allbestlist.com/best-external-hard-drives.html

[8] PC Guide Team, "Cache Characteristics". [Online].Available: http://www.pcguide.com/ref/mbsys /cache/charSpeed-c.html

[9] Tutorials Point Team , "Operating System - Virtual Memory". [Online] . Available: https://www.tutorialspoint.com/operating_system/os_virtual_memory.htm

[10] William Stalling, Operating Systems Internals and Design Principles, 2012, 7th Edition

[11] A.silbershatz and P.B.Galvin and G.Gagne, Operating System Concepts, 2013, 9th Edition

[12] Onur Mutlu, spring 2015.Lecture 20: Virtual Memory

[13] B. Ramamurthy, 2004, Lectures notes from the text supplement by Siberschatz and Galvin

[14] Jerry Breecher, Operating systems Memory Management

[15] Intel Group, "Intel Core i7- 6700 Processor " "https://www.intel.com/content/www/us/en/products/processors/core/core-vpro/i7-6700.html?wapkw=intel +i7+6700"