INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 6)
Available online at: www.ijariit.com

Review on memory divisions in computer architecture

Mudit Jain
mudit.jain2017@vitbhopal.ac.in
VIT Bhopal University, Bhopal, Madhya Pradesh

Devansh Patil
devansh.patil2017@vitbhopal.ac.in
VIT Bhopal University, Bhopal, Madhya Pradesh

Tanay Parikh
tanay.parikh2017@vitbhopal.ac.in
VIT Bhopal University, Bhopal, Madhya Pradesh

Ayush Naidu
ayush.naidu2017@vitbhopal.ac.in
VIT Bhopal University, Bhopal, Madhya Pradesh

P Sanjeevi
sanjeevi.p@vitbhopal.ac.in
VIT Bhopal University, Bhopal, Madhya Pradesh

ABSTRACT

This Research Paper is totally concentrated to define different memory systems that are present in the market, and what is their importance in today's generation. In this paper, we review the different hierarchies of the memory systems. It talks about cache-memory based systems and its various levels. Cache memories along with the virtual memories and processor registers form a field of memory hierarchies that depends on the principle of locality of reference. Most applications show the temporal and spatial zones among order and data. Then it describes about RAM (Random Access Memory) and its types which include DRAM (Dynamic Random-Access Memory) and SRAM (Static Random-Access Memory), it also describes the flash memory and its importance because of its small size and large memory containing abilities Memory hierarchies are intended to keep most likely referenced items in the fastest devices.

Keywords— *DRAM, SRAM, Cache memory, Memory hierarchy*

1. INTRODUCTION

In computing, memory refers to the computer hardware devices which are used to store the data and information for instantaneous use in the computers. It is synonymous with the term primary storage. The memory hierarchy is the key block in modern computer systems as the gap between the speed of the processor and the memory tend to increase. Computer memory operates at a higher speed, for sample Random Access Memory as a distinction from storage that provides slow to access program and data storage but offers higher capacities. If needed contents of the computer memory which can be transferred to the secondary storing through a memory administration technique which is called virtual memory. Though it is not detailed in this paper, Virtual Memory is cited because of inclusiveness and to introduce the TLB cache. An archaic synonym for memory is a store. The term meaning primary storage or main memory is often associated with addressable semiconductor memory that is Integrated Circuits (IC) containing Silicon-Based Transistors [1,3,4].

There are basically two types of semiconductors:

(a) Volatile
(b) Non-volatile

Flash memory is an example of non-volatile memory which is used as a secondary memory and ROM (Read Only Memory), P-ROM (Programmable-Read Only Memory), EEP-ROM (Electrically Erasable-Read Only Memory) and EP-ROM (Erasable Programmable- Read Only Memory) which are used for storing firmware such as BIOS (Basis Input/ Output Systems). The examples of volatile memory include a primary storage which is Dynamic-Random Access Memory and fast CPU cache memory which is typically Static-Random Access Memory which is fast but consumes a lot more energy as compared to the other memories. It offers lower memory areal density than DRAM. Most semiconductor memory is organized into memory cells or bits able flip-flops that stores either 0 or 1. On the other hand, flash memory organization includes both one bit per memory cell and multiple bits per cell which are called MLC (Multiple Level Cell). The memory cells are assembled into words of static word length. Here each word can be accessed by a binary address of n bit, making it possible to store 2 raised by n words in the memory. This implies processor registers normally are not considered as a memory since they only store one word and do not include an addressing mechanism.

It is amazing how many different types of electronic memory we see and use in daily life. Many of them have become an integral part of our life: RAM, ROM, Cache, Dynamic RAM, Static RAM, Flash memory, Memory Sticks, Virtual memory, Video memory, BIOS [1].

As the gap between memory speed and the processor speed grows, the program which works to improve the performance of the memory system has become gradually important. To understand and enhance memory performance researchers and practitioners in presentation analysis and compiler design require a detailed understanding of the memory pecking order of the target computer system. It's not easy to get perfect information about memory hierarchy. Seller microprocessor documentation is often incomplete, vague or worse in its description of important on-chip memory parameters. Also, today's computer system of government contains complex, multilevel memory systems where the processor is but one constituent of the memory systems. Because of the gap between processor performance and memory performance remains to grow, so, presentation analysis and compiler optimisations are increasingly focused on memory hierarchy of the target computers to optimise memory system we need to know for any level of TLB (Translation Lookaside Buffer) or cache, the size, line size, associativity, write allotment and replacement policies and whether each cache or TLB is split or unified [2].

To capitalize the performance, full advantage of limited resources is required to be taken to direct for the specific submission of the memory hierarchy. However, the old-fashioned custom memory hierarchy design organizations are ordered in phases. They separate the application optimization from the memory hierarchy architecture design, which inclines to outcome in locally optimal solutions. In Hardware-Software co-design organizations, most of the work focuses on using the reconfigurable logic to partition the calculation. However, the utilizing configurable logic to achieve the memory hierarchy design is not addressed generally [3].

1.1 Memory Hierarchy

Memory Hierarchy is an idea which is important for the CPU to be able to manipulate the data. Memory Hierarchy is a term Computer Scientists use to explain different ways a Computer handles information, whether the data is temporary or permanent. Computer memory is defined in the below hierarchy (figure 1).

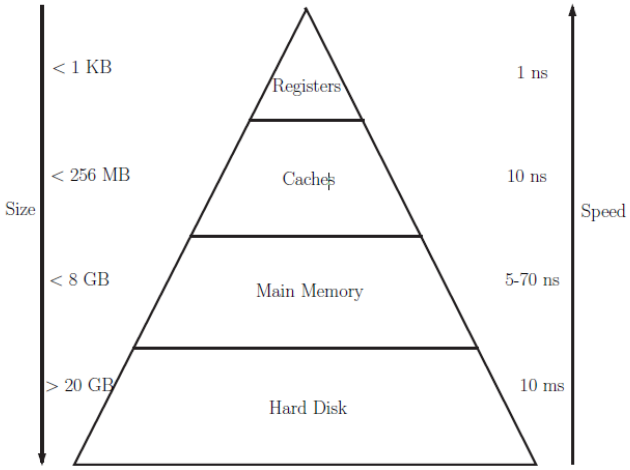

Fig. 1: Memory Hierarchy [6]

2. REGISTER FILES

Register files are being used by current processors to store the data, it also distributes transitional results of computations in Memory Hierarchy. Register Files is the extreme level in several different ways. Register Files is decided and ruled by contradictory requirements. In spherulitic path register read happens and it is achieved in a single cycle and desirable, frequency processor has a Register Files latency needs to be a pipeline which has advantages like:

(a) 1 Rises pipeline length.
(b) 2 Penalties of branch misprediction.
(c) 3 Impacting intricacy and performance.

Through large instruction windows capacity of the Register Files is immense to be effectively exploited. Operands should not be accommodated in Register Files which has advanced latency should have high bandwidth and a large number of sports so that it permits issuing and write back orders in each and every cycle. In multiple thread processors pressure on Register Files is high so a big number of thread context is being billeted Register Files should be protected from easy errors and it is substantial also, because of that any soft error quickly propagates to another system. Register Files have slight energy consumption to meet the power and thermal budget [16].

3. CACHE-BASED SYSTEMS

Even with the precisions in the technology, microprocessors are still considered quicker than the Main Memory. Memory access time is gradually the blockage in the overall performance of the application. As an outcome, an application might spend more amount of time waiting for the data. This not only damagingly impacts on the overall performance, but the application cannot be encouraged much from a processor clock-speed upgrade either. One other way too overwhelmed this problem is the addition of a small high-speed buffer memory between the supercomputer and the main memory. Such a buffer is generally mentioned to as cache memory or cache in short [4].

3.1 Mainly cache memory is divided into two levels:

3.1.1 Level 1

A level 1 cache (L1 cache) is a memory cache that is straight built into the microprocessor, which is used for storing the microprocessor's freshly accessed info, thus it is also called the primary cache. It is also knowing as the internal cache or system cache. L1 is one of the fastest cache memories presents in CPU. It has zero wait state boundary already built within the chip. And because of that, it becomes one of the most well-appointed cache memories in CPU. It is used for storing the data which is freshly used by the processor. Such as critical files that need to be executed directly, when the processor is executing a computer instruction, it is the first cache which is accessed and processed. It has a drawback as it has a limited size [20].

3.1.2 Level 2

An L2 cache memory is located outside and isolated from the microprocessor chip core but it is found on same processor chip package. In past times L2 cache initiatives place them on the motherboard and which made them quite slow. In modern CPUs, the L2 cache memory is normally planned in microprocessor units, though they are not fast as L1 cache, as it is present outside of the core, its capacity can be increased and it is still fast than the main memory. L2 cache is also known as secondary cache or external cache.

3.1.3 Level 3

A specialized cache that is used by the CPU is known as LEVEL 3 cache. In some certain superior processors, it is built within the CPU module but usually, it is built onto the motherboard. It avoids bottlenecks created by the fetch and execute cycle which takes too long. By working together with the L1 and L2 cache to improve computer performance. The L2 cache gets information feed from L3, which then onwards information to the L1 cache. It is quicker than the main memory (RAM), but its memory performance is slower compared to L2 cache The L3 cache is usually built onto the motherboard between the main memory (RAM) and the L2 and L1 caches of the processor module. In order to prevent bottlenecks resulting from the fetching of these data from the main memory, it serves as another bridge to park information like processor commands and frequently used data. In short, the L3 cache and L2 behavior are same, before L2 got built-in within the processor module itself [20].

3.2 Cache mapping is of three types

3.2.1 Direct mapping: In this mapping data is kept in both RAM and cache. Index part and tag part are is two division of address space.

3.2.2. Associative mapping: In this type of mapping, content, and address both of memory world is kept in associative memory. In this, it randomly seats words in the cache memory.

3.2.3. Set associative mapping: It is the grouping of both direct and associative mapping. Lines are gathered in the form of sets. This type is an improved form of direct mapping by removing the drawbacks of direct mapping [9].

First data needed by CPU is taken into the cache from memory after that CPU registers take cache. The transfer of data from the CPU to cache is fast and from cache to memory is slow. Storage of results is in the contradictory direction. Data is copied into the cache by the system. Depending on the cache architecture details, the data is then instantly copied back to memory, or deferred. If an application needs the same data again, data access time is reduced significantly if the data is still in the cache [4].

Finally, the advantages of cache so it is very much operative in system performance, in the system it is the fastest memory, the motherboard's system bus is not used by CPU for data transfer and because of that CPU can process data much faster by sidestepping the bottleneck created by a system bus. But nothing is perfect in this world so it also has disadvantages as it is costlier than RAM [12].

4. RANDOM ACCESS MEMORY

Random Access Memory is the location (hardware) in a computer where the operating system, data, and application programs in current use are kept so that they can be rapidly retrieved by the computer's processor for real picture which is used inside the systems you can prefer fig 2. From another kind of storage in a computer (hard disk, CD -ROM and floppy disk) RAM is much quicker than any of this thing in reading and writing. When the computer is turned off or shunt down RAM misplaces its data. And when you turn on the computer again files and operating system are once again loaded into RAM usually from the hard disk. If we compare RAM and hard disk with a person then we can say that a person's short-term memory is RAM and long-term memory is a hard disk. The short-term memory focuses on work at hand but can keep so many facts in view at a time. If short-term memory fills up, the person's brain occasionally is able to refresh it from facts stored in long-term memory. A computer also works this way. If RAM fills up, the processor needs to frequently access to the hard disk to overlay old data in RAM with new, it results in slowing down the computer's operation [13].

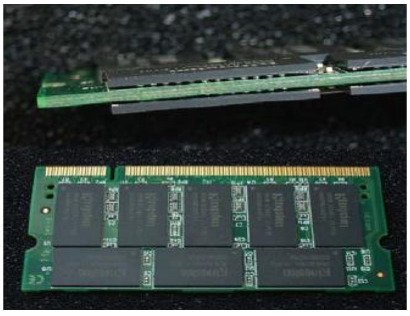

Fig. 2: RAM [8]

4.1 Static Random-Access Memory

Static Random-Access Memory is one of the variations of RAM. It is designed to fill two needs first one is to deliver a direct interface to CPUs at speeds unachievable by DRAMs and the second one is to replace DRAMs in systems that require very low power consumption. It performs very well in low power applications due to the nature of the device. SRAM cells are comprised of six MOSFETs. Bit information is stored in four transistors which act as across attached inverters, while the residual transistors control access to data during reading/write operations is going on. It uses low power competence that's why it is used in portable equipment's and also it does not require a refresh cycle due to the absence of capacitors in its design. We can use SRAM as it is still volatile but this type of applications is not used because of its price [14].

4.2 Dynamic random-access memory

Dynamic random-access memory stores all its data in each bit of data in a distinct tiny capacitor within the integrated circuit. The capacitor can either be discharged or charged, these two states are taken to characterize the two values of a bit, conventionally called 0 and 1. The data stored in the capacitor is soon misplaced as an electric charge on the capacitors slowly leaks off. For protecting the loss of data DRAM requires an external memory refresh circuit which sporadically restores the data in the capacitors, restoring them to their original charge. Because of this refresh obligation, it gets different from the static random-access memory which does not necessitate data to be refreshed. DRAM is a volatile memory but it is not used because its data get lost quickly as power is detached. However, DRAM does show limited data remanence. DRAM has many demands in digital electronics where low-cost and high-capacity memory is required. One of the major applications for DRAM is the graphics cards (where the "main memory" is called the graphics memory) and main memory in modern computers. It is also used in numerous video games consoles and portable devices. Although SRAM is costly so we only use SRAM over DRAM where speed is of a superior concern than cost, such as the cache memorise in processors [15].

5. FLASH MEMORY

Flash memory follows Moore's law. Flash memory is low-priced than DRAM and quicker than disks. So, in this way research has been going on assimilating flash devices as an auxiliary for storage purposes. You can see in figure 3 the high-security USB flash drive architecture. The flash provides respectable performance and a price characteristic to back a virtual memory. Through asynchronous page write options longer written dormancy is hidden and it is relatively longer and allows faster page access. Flash virtual memory has been examined and its use had been deliberated. The system of their responsiveness improves flash disk as swap space. Due to its limited write durability swapping was done to avoid for supporting and avoiding flash. It has investigated the truth behind the issues and revisiting the VM hierarchy in light of flash memory [5].

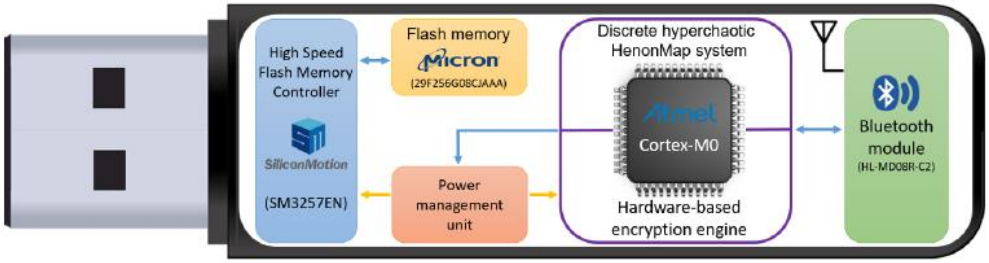

Fig. 3: High-security USB flash drive architecture [11]

6. HARD DISK

The hard disk is also called hard drive or fixed disk, is an electromechanical data storage device. It is used to store magnetic storage and repossess digital information. It uses one or more rotating disks which are coated with magnetic material. you can see a commonly used hard disk in figure 4. The platters are paired with magnetic heads which has actuator arm.it reads and writes to platter surfaces. The data is retrieved in a random-access manner. It's can be stored in any order but not sequentially [15].

Fig. 4: Hard Disk [18]

7. MAGNETIC TAPES

It improves the presentation and it also extends the utility of information which calls for access for even larger data, so the need for more data storage with more time opens doesn't matter how we make progress. The parameters are capacity, access time, data transfer rate, and the cost per bite for basic routine characteristics of storage devices. You can see a magnetic tape and a CD in figure 5(i)and 5(ii) which are used to store the data previously. Each storage has own characteristics and most applications of modern electronic computer era are now 30 years old. We have seen main-memory technology development from vacuum tubes to mercury delay lines to cathode-ray tubes to magnetic cores and now perceiving the change to semiconductor technology with LSI [17].

(a) **(b)**

Fig. 5: (a) Magnetic Tapes, (b) CD [19]

8. IMPROVING MEMORY HIERARCHY PERFORMANCE

The gap between CPU speed and memory speed are growing day by day as innovative generations of computer systems are being introduced and used in the cloud [21-30]. Multi-level memory hierarchies are the standard architectural design which is used to address this memory access bottleneck. As the gap between CPU speed and memory speed upsurges, systems are being created with deeper hierarchies. Attaining high performance on such systems needs couture the reference behavior of applications to well match the characteristics of the machine's memory hierarchy. Techniques such as loop blocking and data prefetching have significantly improved memory hierarchy utilization for regular applications. A constraint of these methods is that they are not as effective for irregular applications. Improving routine for irregular applications is extremely important since large-scale scientific and engineering simulations are using adaptive irregular methods. Irregular applications are characterized by outlines of data and computation that are not known until runtime. In such applications, access to data often has poor spatial and sequential locality, which central to ineffective use of a memory hierarchy. Improving memory system performance for the irregular applications requires addressing problems of both the latency and bandwidth. Latency is a problem because of poor temporal and spatial reuse results in elevated cache and Translation Lookaside Buffer miss rates. Bandwidth is a problem because unintended references which are found in irregular applications tend to have poor spatial locality. Thus, when access cause blocks of data to be drawn into the various levels of the memory hierarchy, items within a block are either referenced only a few times or not at all before the block is evicted due to conflict or capacity misses, even though these items will be referenced later in the execution [10].

9. MEMORY HIERARCHY HARDWARE-SOFTWARE CO-DESIGN IN EMBEDDED SYSTEMS

Embedded system has many categories of diverse characteristics as related to general-purpose systems. Software and hardware both get combine so that they can run specific applications that range from multimedia into the industrial control system. These kinds of application differ very high in their characteristics. Maximize performance, minimize cost request for different hardware architects or it has a trade-off between the performance and the cost because of the expected objectives. On second general purpose system and embedded system are characterized by the restrictive resources and embedded system.in addition to the vigorous restriction, embedded systems have to deliver high computation capability and meet real-time constraints [7].

10. CONCLUSION

In this research paper, we have given an introduction to Memory Systems. We have tried to show the importance of Memory Systems in our day to day life. First, we have discussed the memory hierarchy. In the hierarchy we have discussed the register files, then we have discussed the cache-based systems and its different levels. Further moving down, the paper we have discussed RAM and its different types which include DRAM and SRAM. Some introductory information about Flash Memory has also been given. Also, information regarding Hard Disks and Magnetic Tapes is given. Techniques to improve the Memory Hierarchy performance has also been discussed. Memory Hierarchy is a very wide concept but we have tried to provide some of its basic information for the beginners.

11. REFERENCES

[1] Nikola Zlatan "Computer Memory, Applications and Management". p.30 2016
[2] Clark L. Coleman and Jack W. Davidson "Automatic Memory Hierarchy Characterization" Department of Computer Science, University of Virginia 2001
[3] Zhiguo Ge, H. B. Lim, W. F. "Wong Memory Hierarchy Hardware-Software Co-design in Embedded Systems" 2005
[4] Ruud van der Pas, High-Performance Computing "Memory Hierarchy in Cache-Based Systems" Sun Microsystems, Inc. Part No.817-0742-10 11/20/02, Revision an Edition: November 2002
[5] Mohit Saxena and Michael M. Swift "FlashVM: Revisiting the Virtual Memory Hierarchy" Department of Computer Sciences University of Wisconsin-Madison 2009

[6] Deepak Ajwani, Henning Meyerhenke Chapter 5. Realistic Computer Models January 2010
[7] Zhiguo Ge, H. B. Lim, W. F. "Memory Hierarchy Hardware-Software Co-design in Embedded Systems" Department of Computer Science, Singapore-MIT Alliance, National University of Singapore 2005
[8] Otto Andersen1, Idun Husabø Anderssen1, Johan Liu2, Teng Wang2, David Whalley3, Helge Kristiansen4, Tom Ove, Grønlund4, Krystyna Bukat5, Jianying Liu6, Xiuzhen Lu6, Zhaonian Cheng6, December 2006
[9] Dr. Vivek Chaplot "Cache Memory: An Analysis on Performance Issues" HEAD, Dept. of Computer Science Bhupal Nobles' University Volume 4, Issue 7, July 2016
[10] John Mellor-Crummey, David Whalley, Ken Kennedy "Improving Memory Hierarchy Performance for Irregular Applications Using Data and Computation Reordering's" 2001
[11] Teh-Lu Liao 1, Pei-Yen Wan 1, Pin-Cheng Chien 1, Yi-Chieh Liao 2, Liang-Kai Wang 3 and Jun-Juh Yan 4 "Design of High-Security USB Flash Drives Based on Chaos Authentication" 2018
[12] Akshay Kanwar, Aditi Khazanchi, Lovenish Saluja "Cache Memory Organization" International Journal of Engineering and Computer Science ISSN:2319-7242 Volume 2 Issue 10 October 2013 Page No. 2944-2950
[13] Peter Haugen, Ian Myers, Bret Sadler, John Whidden "A Basic Overview of Commonly Encountered types of Random-Access Memory", 2001
[14] Sourangsu Banerji "Architectural Design of a RAM Arbiter" Department of Electronics & Communication Engineering, RCC-Institute of Information Technology, Under West Bengal University of Technology, April 2014
[15] https://en.wikipedia.org/wiki/Dynamic_random-access_memory(accessed 04/10/2018)
[16] Sparsh Mittal A "Survey of Techniques for Designing and Managing CPU Register file" Concurrency and computation: practice and experience Concurrency Computat.: Pract. Exper. 2016; 00:1–23
[17] Albert S. Hoagland, "Magnetic Recording Storage" IEEE transactions on computers, vol. c-25, no. 12, December 1976
[18] Dr. engineer Salah alkhafaji "Fundamentals of Information Technology" January 2016
[19] Federica Bressan,1 Antonio Rodà,2 Sergio Canazza,2 Federico Fontana,3 Roberta Bertani4, "The Safeguard of Audio Collections: A Computer Science-Based Approach to Quality Control—The Case of the Sound Archive of the Arena di Verona, December 2013
[20] Techopedia https://www.techopedia.com/definition/8048/level-1-cache-l1-cache (accessed 04/10/2018)
[21] P. Sanjeev and P. Viswanathan, 'NUTS scheduling approach for cloud data centers to optimize energy consumption', *Computing (Springer)*, Vol. 99, No. 12, pp. 1179-1205, 2017.
[22] P. Sanjeevi and P. Viswanathan, 'Workload Consolidation Techniques to Optimize Energy in Cloud: Review', *Int. J. of Internet Protocol Technology*, Vol. 10, No. 2, pp. 115–125, 2017.
[23] P. Sanjeevi and P. Viswanathan, 'DTCF: Deadline Task Consolidation First for energy minimization in cloud data centers', *International Journal of Networking and Virtual Organizations*, Inderscience, Vol. 19, No. 3, pp. 209–233, 2017.
[24] P. Sanjeevi and P. Viswanathan, 'Employing Smart Homes IoT Techniques for Dynamic Provision of Cloud Benefactors', *Int. J. of Critical Computer-Based Systems*, Inderscience, Vol. 7, No. 3, pp. 209–224, 2017.
[25] P. Sanjeevi and P. Viswanathan, 'A survey on various problems and techniques for optimizing energy efficiency in cloud architecture', *Walailak Journal of Science and Technology*, Vol. 14, No. 10, 2017.
[26] P. Sanjeevi, P. Viswanathan, M. R. Babu, and P. V. Krishna, 'Study and Analysis of Energy Issues in Cloud Computing', *International Journal of Applied Engineering Research*, Vol. 10, No. 7, pp. 16961-16969, 2015.
[27] P. Sanjeevi, G. Balamurugan, and P. Viswanathan, 'The Improved DROP Security based on Hard AI Problem in Cloud', *Int. J. of Internet Protocol Technology*, Vol. 9, No. 4, pp. 207-217, 2017.
[28] P. Sanjeevi and P. Viswanathan, 'A green energy optimized scheduling algorithm for cloud data centers', *IEEE International Conference on Computing and Network Communications, Trivandrum*, pp. 941-945, 2015.
[29] P. Sanjeevi and P. Viswanathan, 'Towards energy-aware job consolidation scheduling in the cloud', *International Conference on Inventive Computation Technologies (ICICT 2016), IEEE Xplore*, pp. 361- 366, 2016.
[30] G. Kesavan, P. Sanjeevi and P. Viswanathan, 'A 24 hour IoT framework for monitoring and managing home automation', *International Conference on Inventive Computation Technologies (ICICT 2016), IEEE Xplore*, pp. 367-371, 2016.

