Interconnected Memory Design using GALS Concept

Shahina Parveen, Himanshu Joshi

***Abstract*—Memories are the essential and the vital part of any electronic system device. To have a memory on network or on system is an great asset to a system, however the need of memory size keeps on upgrading itself as the sytem keeps on working and upgrades itself during its lifetime so to have a memory with dynamic possibilities will be a great asset. Here in this paper we have presented a SOC which has a memory design which can be upgraded any time with any size as per the system requirements, to avoid the timing delays and to reduce the power consumption. Two memory architectures has been compiled using the GALS Concept (Globally Asynchronous and Locally Synchronous), to avoid the power consumption by switching the components. Memories designed have the capabilty of reading and writing the data at the same time, to enable the design in such a way finite state machines are used.**

***Index Terms*— Memory, GALS, SOC, finite state machines.**

I. Introduction

In computing, memory refers to the devices used to store information for use in a computer. The term primary memory is used for storage systems which function at high-speed (i.e. RAM), as a distinction from secondary memory, which pro-vides program and data storage that is slow to access but offer higher memory capacity. If needed, primary memory can be stored in secondary memory, through a memory management technique called "virtual memory". An archaic synonym for memory is store.

GALS architecture suggests to design a system in such a way so that if a system consists of individual modules which may have their individual clocks as well will get individual clock port but their functionality will depend on each other’s current state, i.e. the after the input arrived at the first block after the first block complets its process cycle then only the next block will get activated and the first block will gets back to the inactive state.

A finite-state machine (FSM) or finite-state automaton (plural: automata), or simply a state machine, is a mathematical model of computation used to design both computer programs and sequential logic circuits. It is conceived as an abstract machine that can be in one of a finite number of states.

II. II. PREVIOUS WORK

From the rigorous review of related work and published literature, it is observed that many researchers have designed MOS content addressable memory by applying different techniques. Researchers have undertaken different systems, processes or phenomena with regard to design and analyse MOS content addressable memory and attempted to find the un-known parameters. Since in the real world today VLSI/CMOS is in very much in demand, from the careful study of re-ported work it is observed that very few researchers have taken a work for designing MOS content addressable memory with CMOS/VLSI technology.[1] Data processing/storing that involves many search operations performed by software consumes enormous time. This is a hindrance to high-speed data processing. A fully parallel content addressable memory (CAM) compares search data with storage data in a parallel fashion, and is extremely suitable for high-speed data searching. A search operation carried out by a fully parallel CAM is several hundred times faster than that performed by software. Therefore, there have been many studies that have addressed the applicability of CAM’s to artificial intelligence (AI) machines, data-base systems, and so on in which search operations are conducted frequently. Conventional CAM’s use a static CAM cell, which keeps storage data in a latch, to accomplish a stable operation it was discussed by T. Nikaido, T. Ogura, S. Hamagrrchi, and S. Muramoto, in their publish paper in 2013 Static CAM cells,[2] However, the speed of a CAM comes at the cost of increased silicon area and power consumption, two design parameters that designers strive to reduce. As CAM applications grow, demanding larger CAM sizes, the power problem is further exacerbated. Reducing power consumption, without sacrificing Speed or area, is the main thread of recent research in large-capacity CAM. Several techniques have been proposed to reduce the power consumption in CAMs. In Jun. 2007 C. Zukowski and S. Wang, published a paper in which they used a selective precharge CAM which reduces the power by precharging the match lines only if there is match in the selected bits of the words.[3] The number of selected bits is optimized for minimum average energy the power consumption is reduced by shutting down the power for redundant comparisons but there was problem of voltage swing of match line occurred. Later on H. Miyatake, M. Tanaka and Y. Mori published a paper in which voltage swing of ML is reduced by charging it through an NMOS transistor and discharging it by PMOS comparison logic. A precomputation-based CAM was recently proposed by C. S. Lin, J.C. Chang and B. D. Liu which reduces the power by extracting a smaller size parameter from each word.[4] Hence, the power reduction is achieved at the expense of slower search speed. Moreover, increased complexity due to the additional logic makes it difficult to cascade multiple CAMs. Recently, a current-race sensing scheme is proposed that reduces the power. Latter on Nitin Mohan and Manoj Sachdev published a paper in which they presented a dual ML TCAM scheme they showed that by theoretical analysis and simulation results that the dual ML TCAM results in significant power reduction (up to 43%) at the expense of small trade-off in speed (4%). Verma and chandrakasan demonstrate that at very low supply voltages the static noise margin for SRAM will disappear due to process variation to address the low SNM for subthreshold supply voltage. This means, there is a need for significant increase in silicon area to have reduced failure when the supply voltage has been scaled down.[5]

Memory processing has been considered as the pace-setter for scaling a technology. A number of performance parameters including capacity (that relate to area utilization), cost, speed (both access time and bandwidth), retention time, and persistence, read/write endurance, active power dissipation, standby power, robustness such as reliability and temperature related issues. Memristor was originally envisioned in 1971 by circuit theorist Leon Chua, Memristor - the missing circuit element. IEEE Trans. Circuit Theory 18, 507–519.[6] He describe memristor as a missing non-linear passive two terminal electric component relating electric charge and magnetic flux linkage Leon Chua more recently argued that the definition should be generalized to cover all forms of 2-terminal non-volatile memory devices based resistance switching effects although some experimental evidence contradicts this claim. In 2008 team at HP labs announced the development of switching memristor based on a film of titanium dioxide these devices are intended in nanoelectric memory, and computer logic. Chua postulated that a new circuit element defined by the singlevalued relationship must exist, whereby current moving through the memristor is proportional to the flux of the mag-netic field that flows through the material. The design of CAM based on memristor using VLSI technology.[7] The memristor behaves as a switch, much like a transistor. However, unlike the transistor, it is a two-terminal rather than a three-terminal device and does not require power to retain either of its two states so due to this property of memristor there is further reduction in power consumption than that of conventional CAM. Memristor changes its resistance between two values and this is achieved via the movement of mobile ionic charge within an oxide layer, furthermore, these resistive states are non-volatile.[8] This behavior is an important property that influences the architecture of CAM systems, where the power supply of CAM blocks can be disabled without loss of stored data. Therefore, memristor-based CAM cells have the potential for significant saving in power dissipation.

III. METHODOLOGY

Globally asynchronous locally synchronous (GALS) de-signs are gaining importance due to the fact that the synchrony assumption is failing in large synchronous designs. This is because of the ever increasing clock frequencies, which causes the time taken for a signal to propagate between different components to be longer than the clock period. In a GALS design, the communication between the synchronous components occurs asynchronously. The synchrony assumption holds within each synchronous component. However, there are other challenges facing the design of GALS systems. There is a lack of tools and design methodologies to facilitate GALS designs. In most cases, GALS designs are constructed using ad hoc methods, where synchronous components are encapsulated with some wrapper logic and communication is handshake driven. Furthermore, these ad hoc approaches are not easily subject to formal reasoning about the correctness of a design. Hence, we need to identify the basic ingredients for a successful GALS design methodology.[9]

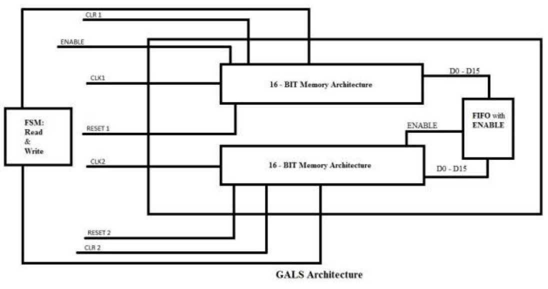
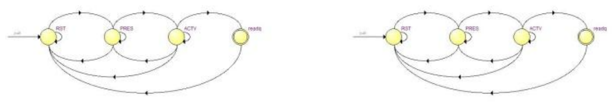


Figure 1. Proposed Architecture

Designed architecture (Figure 1), consists of the memories which are two separate 16 bit memories and can be extended upto n-bits are used to implement the suggested GALS architecture in the base paper of this thesis work. Two independent memory architectures works independently externally but they are connected through a FIFO which enables the system to switch between the memories according to the need of a user.

To enable the system to write and read over a single clock pulse two FSMs have been designed as presented in the figure 2.



(a) Write FSM (b) Read FSM

Figure 2. Finite State Machines

IV. RESULTS

The architecture proposed in figure 1 has been implemented using VHDL, on successful compilation over XILINX generated the RTL schematic presented in figure 3.

As the proposed model suggested memory architecture has been designed using a fifo and the concept of GALS has been incorporated to enhance the timing and to reduce the power consumption, also the designed architecture uses a fifo for a lossless data transmission between the two memory architecture which are incorporated into a single model design.

A fifo is a section which acts as a mediator between the two memory architectures to control the rate of transfer of data. As the analysis of the memories suggests that the data transmits at the lower rata data transfer rate and the data receiving rate is slightly higher than the data transmit rate so as to control the data flow rate it becomes very important to have a fifo in between which will now act as buffer and will provide the balance for both the ends to avoid any data loss during the transmission

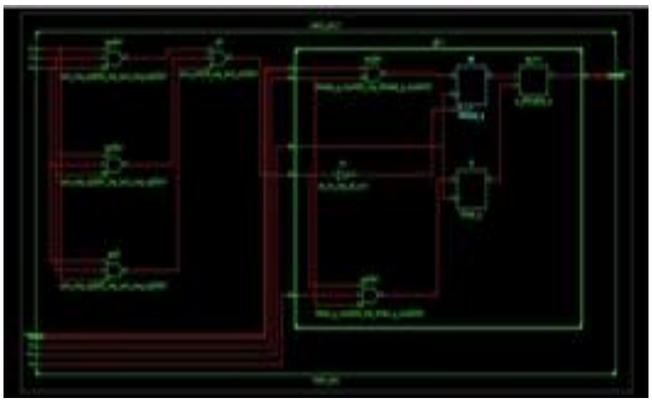


Figure 3. Implemented Architecture

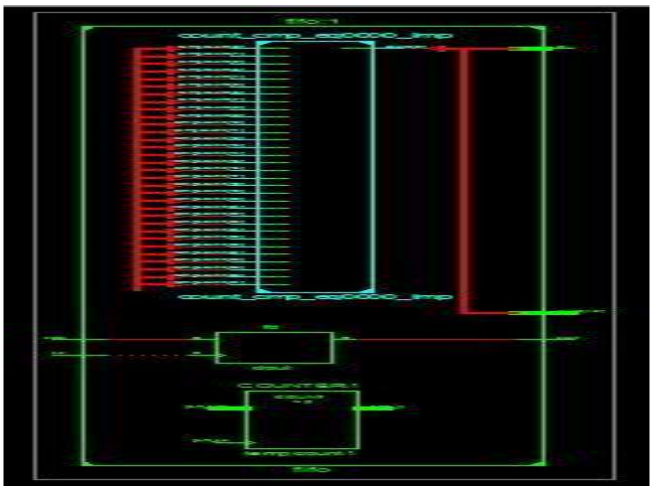


Figure 4. FIFO Architecture

Resulted time has been calculated by the Xilinx and have been mentioned in the design summary report, it can be noted that the complete processing time is 1.121ns.

V. CONCLUSIONS

On the successful complilation of the designed hardware over the Xilinx 14.2 and Altera Quartus 12.1, it can be postulated that the designed architecture can be very usefull for the memory architectures, as it has every essential module to save and secure the data and also the GLAS concepts helps to keep the the design under low power consumption category.

Also on the basis of the result obtained this work can further be taken to the next level by implementing the same architecture for the multiple processors cum memory architectures to enhance the speed of the system and reduce the timing delays without compromising the duability and the realibility of the system.

VI. REFERENCES

[1]L. O. Chua "Memristor"- The missing circuit element methodology. IEEE Transactions circuit theory, vol. 18, pp. 507-519, 2009
[2]L, O, Chua, S. M. Kang "Memristor devices and systems" Proc. IEEE, VOL. 64, PP. 209-223, 2013
[3]Kamran Eshraghian, Kyoung-Rok Cho, Omid Kavehei, Soon-Ku Kang ,Derek Aband Sung-Mo Steve Kang "Mem-ristor MOS Content Addressable Memory(MCAM): Hybrid Architecture for Future High Performance Search Engines" "IEEE Transaction on very large scale integration system", Vol 19, pp. 1-11, 2010
[4]K. Pagiamtzis and A. Sheikholeslami, "Content-Addressable Memory (CAM) Circuits and Archetictures: A Tutorial and Survey," IEEE Journal of Solid-State Circuits, vol. 41, pp. 712-727, 2009.
[5]Baker, R. Jacob (2010). CMOS: Circuit Design, Layout, and Simulation, Third Edition.
[6]http://en.wikipedia.org/wiki/Memristor
[7]Weste, Neil H. E. and Harris, David M. (2010). CMOS VLSI Design
[8]Krieger, J. H.; Spitzer, S. M. (2010), "Non-traditional, Nonvolatile Memory Based on Switching and Retention Phe-nomena in Polymeric Thin Films", Proceedings of the 2010 Non-Volatile Memory Technology Symposium, IEEE, p. 121.
[9]Chattopadhyay, A.; Rakosi, Zoltan (2014), Combi-national logic synthesis for material implication, "2014 IEEE/IFIP 19th International Conference on VLSI and System-on-Chip", VLSI-SoC: 200.
[10]The IEEE website. [Online]. Available: http://www.ieee.org/ [11] Valov, I.; Linn, E.; Tappertzhofen, S.; Schmelzer, S.; van den Hurk, J.; Lentz, F.; Waser, R. (2013). "Nanobatteries in redox-based resistive switches require extension of memristor theory". Nature Communications