ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 2, Issue 2, Ver. 3 (April - June 2014)
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014)

FPGA Implementation of Energy Efficient Cache Architecture with Modified LFSR and Shared LUT Architecture

[I]J.Iswarya, [II]B.Muthupandian
[I]Final Year, M.E. – VLSI DESIGN, [II]Asst. Professor, ECE Department
[I,II]Sethu Institute of Technology, Kariapatti, India

Abstract
A tag to each way in the L2 cache is attached in this project, which is called as way tagged cache architecture. This way tag is sent to the way-tag arrays in the L1 cache when the data is loaded from the L2 cache to the L1 cache. Utilizing the way tags stored in the way-tag arrays, the L2 cache can be accessed as a direct-mapping cache during the subsequent write hits, thereby reducing cache energy consumption. Shared LUT and Modified LFSR Architecture which efficiently performs for data array management is proposed. The proposed system is designed using Verilog HDL, simulated using Modelsim software and synthesized using Xilinx Project Navigator.

Keywords
Cache, Low Power, Data Array management, way-tagged cache, way-tagged buffer.

I. Introduction

Multi-level on-chip cache systems have been widely adopted in high-performance microprocessors [1]–[3]. To keep data consistence throughout the memory hierarchy, write-through and write-back policies are commonly employed. Under the write-back policy, a modified cache block is copied back to its corresponding lower level cache only when the block is about to be replaced. While under the write-through policy, all copies of a cache block are updated immediately after the cache block is modified at the current cache, even though the block might not be evicted. As a result, the write-through policy maintains identical data copies at all levels of the cache hierarchy throughout most of their life time of execution. This feature is important as CMOS technology is scaled into the nanometer range, where soft errors have emerged as a major reliability issue in on-chip cache systems. It has been reported that single-event multi-bit upsets are getting worse in on-chip memories [7–9]. Currently, this problem has been addressed at different levels of the design abstraction. At the architecture level, an effective solution is to keep data consistent among different levels of the memory hierarchy to prevent the system from collapse due to soft errors [10–12]. Benefited from immediate update, cache write-through policy is inherently tolerant to soft errors because the data at all related levels of the cache hierarchy are always kept consistent. Due to this feature, many high-performance microprocessor designs have adopted the write-through policy [13–15]. While enabling better tolerance to soft errors, the write-through policy also incurs large energy overhead. This is because under the write-through policy, caches at the lower level experience more accesses during write operations. Consider a two-level (i.e., Level-1 and Level-2) cache system for example. If the L1 data cache implements the write-back policy, a write hit in the L1 cache does not need to access the L2 cache. In contrast, if the L1 cache is write-through, then both L1 and L2 caches need to be accessed for every write operation. Obviously, the write-through policy incurs more write accesses in the L2 cache, which in turn increases the energy consumption of the cache system. Power dissipation is now considered as one of the critical issues in cache design. Studies have shown that on-chip caches can consume about 50% of the total power in high-performance microprocessors [4]–[6].In this paper, we propose a new cache architecture, referred to as way-tagged cache, to improve the energy efficiency of write-through cache systems with minimal area overhead and no performance degradation. Consider a two-level cache hierarchy, where the L1 data cache is write-through and the L2 cache is inclusive for high performance. It is observed that all the data residing in the L1 cache will have copies in the L2 cache. In addition, the locations of these copies in the L2 cache will not change until they are evicted from the L2 cache. Thus, we can attach a tag to each way in the L2 cache and send this tag information to the L1 cache when the data is loaded to the L1 cache. By doing so, for all the data in the L1 cache, we will know exactly the locations (i.e., ways) of their copies in the L2 cache. During the subsequent accesses when there is a write hit in the L1 cache (which also initiates a write access to the L2 cache under the write-through policy), we can access the L2 cache in an equivalent direct-mapping manner because the way tag of the data copy in the L2 cache is available. As this operation accounts for the majority of L2 cache accesses in most applications, the energy consumption of L2 cache can be reduced significantly.

II. Proposed Maethod

The two level cache architecture is illustrated in Fig.1. Only the L1 data cache and L2 unified cache are shown as the L1 instruction cache only reads from the L2 cache. Under the write through policy, the L2 cache always maintains the most recent copy of the data. Thus, whenever a data is updated in the L1 cache, the L2 cache is updated with the same data as well. This results in an increase in the write accesses to the L2 cache and consequently more energy consumption.

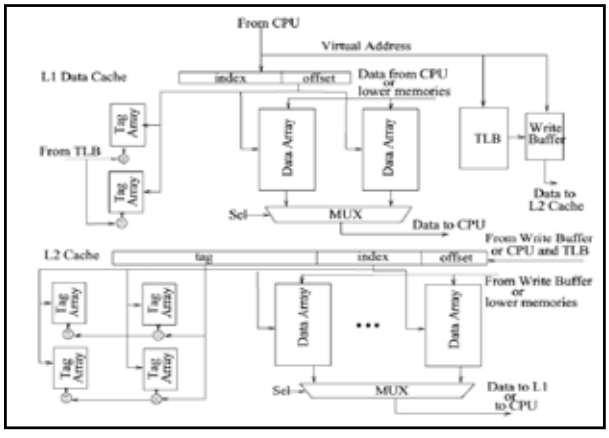


Fig. 1: Two level cache architecture

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

A. Way Decoder

The function of the way decoder is to decode way tags and activate only the desired ways in the L2 cache. As the binary code is employed, the line size of way-tag arrays is n=log N bits, where N is the number of ways in the L2 cache. This minimizes the energy overhead from the additional wires and the impact on chip area is negligible. For a L2 write access caused by a write hit in the L1 cache, the way decoder works as a decoder that selects just one way-enable signal.

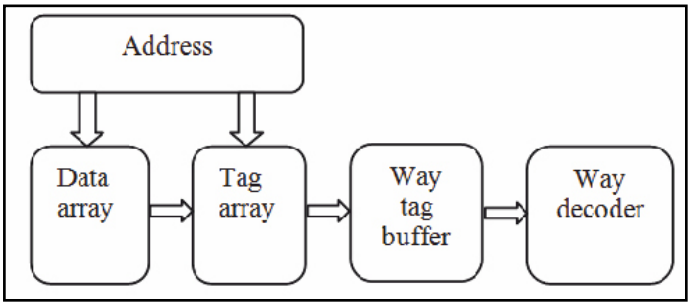


Fig. 2: Interfacing way tag array and buffer

The way decoder operates simultaneously with the decoders of the tag and data arrays in the L2 cache. For a write miss or a read miss in the L1 cache, we need to assert all way-enable signals so that all ways in the L2 cache are activated. To achieve this, the way decoder can be implemented by the circuit shown in Fig. 5.

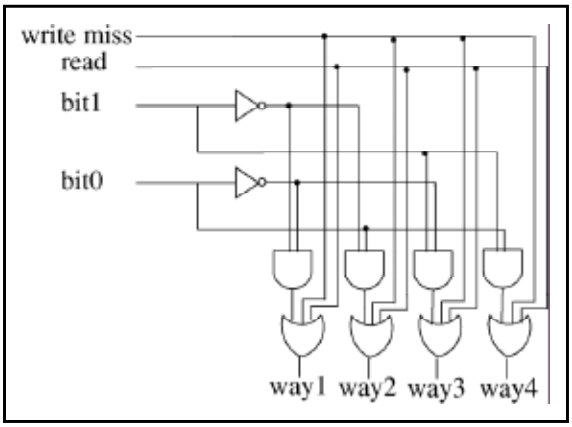


Fig. 3: Way decoder

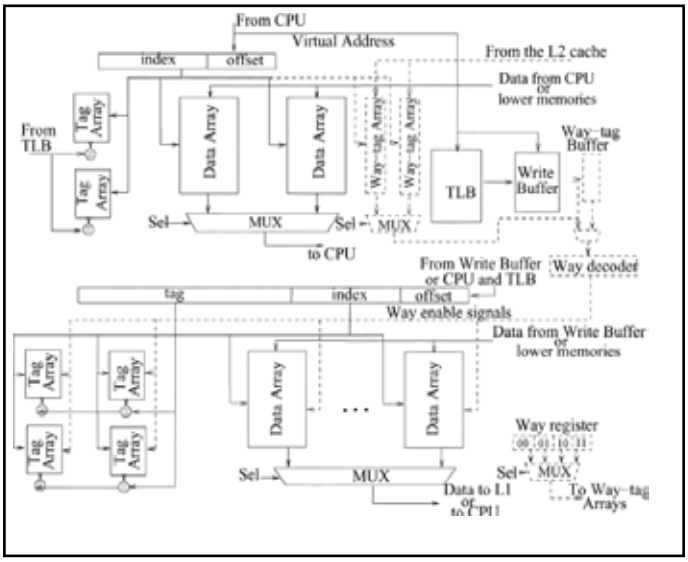


Fig. 4 : Way tagged cache architecture

Two signals read and write miss, determine the operation mode of the way decoder. Signal read will be "1" when a read access is sent to the L2 cache. Signal write miss will be "1" if the write operation accessing the L2 cache is caused by a write miss in the L1 cache.

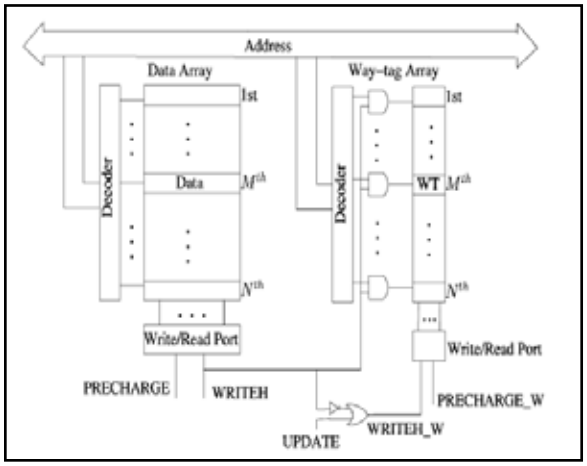


Fig. 5 : Way tag array design

In the proposed way-tagged cache, each cache line in the L1 cache keeps its L2 way tag information in the corresponding entry of the way-tag arrays, as shown in Fig. 4.When a data is loaded from the L2 cache to the L1 cache, the way tag of the data is written into the way-tag array. At a later time when updating this data in the L1 data cache, the corresponding copy in the L2 cache needs to be updated as well under the write-through policy. The way tag stored in the way-tag array is read out and forwarded to the way-tag buffer. Note that the data arrays in the L1 data cache and the way-tag arrays share the same address as the mapping between the two is exclusive. The write/read signal of way-tag arrays, WRITEH_W, is generated from the write/read signal of the data arrays in the L1 data cache as shown in Fig.4. A control signal referred to as UPDATE is obtained from the cache controller.

B. Way tag Buffer

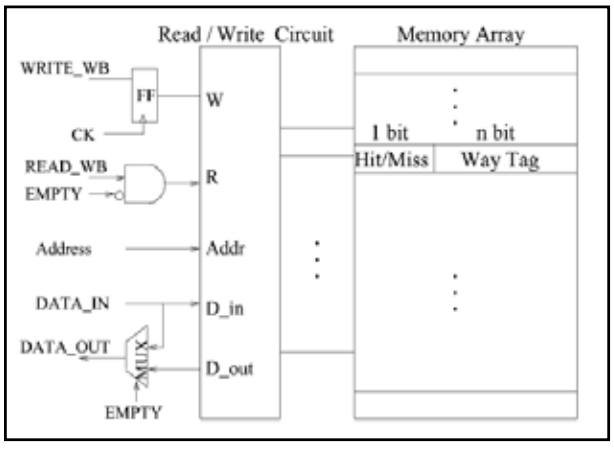


Fig. 6: Way tag buffer

Way tag buffer receives the data from way tag array and store in to memory elements. The output from way tag buffer is sent to way decoder. Way-tag buffer temporarily stores the way tags read from the way-tag arrays. The way-tag buffer has separate write and read logic in order to support parallel write and read operations. The write operations in the way-tag buffer always occur one clock cycle later than the corresponding write operations in the write

buffer. This is because the write buffer, L1 cache, and way-tag arrays are all updated at the same clock cycle when a STORE instruction accesses the L1 data cache (see Fig. 6). Since the way tag to be sent to the way-tag buffer comes from the way-tag arrays, this tag will be written into the way-tag buffer one clock cycle later. The EMPTY signal of the way-tag buffer is employed as the enable signal for read operations; i.e., when the way-tag buffer is empty, a read operation is not allowed. During normal operations, the write operation and the way tag will be written into the write buffer and way-tag buffer, respectively. Thus, when this write operation is ready to be sent to the L2 cache, the corresponding way tag is also available in the way-tag buffer, both of which can be sent together.

C. Common or Shared LUT architecture

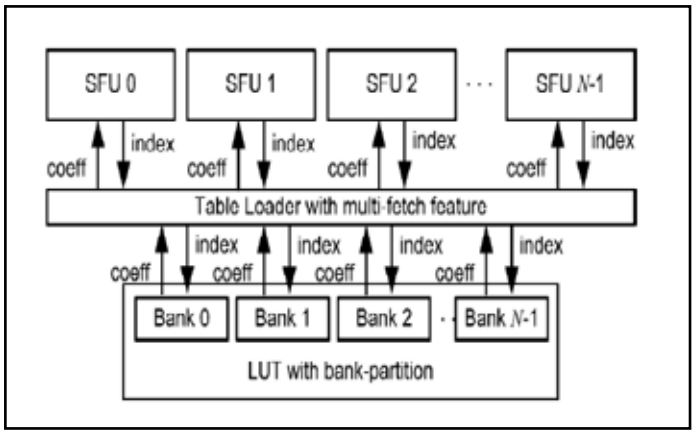


Fig. 7: Common or Shared LUT architecture

A shared or common LUT architecture is proposed to be applied in data array management of this cache architecture.

Since data array in cache architecture is associated with multiplexer selection based processor for data accessing, we are introducing an shared LUT in which all data information is loaded with table loader according to is index and coefficients for data finding and matching allocation during cache operations. Hence data array can be replaced by shared LUT architecture with effectively acts and reduces the total power consumption of overall way tag array cache architecture. From the fig .7. the shared LUT architecture is partitioned in to four banks with respective address associated with it. If a processor needs to access data from bank 3, it will directly access that information via its coefficient bit address by matching with table loader indexes .Hence a long searching process is limited to direct accessing technique through shared LUT architecture .Apart from banks it also has SFU-Special Functional Units in it. It is connected to table loader. These SFU's can access all the banks by having simple indexes like "000" the first zero represents the number of SFU i.e SFU 0. Hence the rest two zero's represents the bank coefficient. By bit matching, SFU easily connects with bank 0 which contain relevant data access in cache operations .If SFU0 and SFU 1 having values like "000" and "100" then confusion is cleared by higher priority portal. The Higher priority is nothing but one which comes first is allowed to access the data first too. The rest request signals accessed in parallel after that. To generate bits we include modified LFSR-Linear Feedback Shift Register in it.

D. Modified LFSR

There are three flip-flops in LFSR having 2^n-1 combinations in it. Default values are loaded in these flip-flops with one condition that not all the flip-flops are loaded with zeros simultaneously. If all the flip-flops are zeros at default means, it creates the loop and access the same memory location . Hence this condition is implemented while processing. These flip-flops having names given as Q-A, Q-B, Q-C is loaded with default values as "100" .For every clock signal the flip-flops are XORed and generate the output sequence in LFSR in an PN-sequence order up to 2^n-1 combinations. Now two conditions has to be satisfied. The PN sequence is shown in table I.

First is that the number of one's should be greater than number of zeros.

Secondly the difference between number of one's and zero's should be one.

When the above two conditions are satisfied then that sequence is selected as address generator for LUT architecture. This is Modified linear feedback shift register

Table 1 : PN Sequence Generator

Clk	QA^QC	QA	QB	QC	PN-SEQ
t	1	1	0	0	0
1	1	1	1	0	0
2	0	1	1	1	1
3	1	0	1	1	1
4	0	1	0	1	1
5	0	0	1	0	0
6	1	0	0	1	1

III. Simulation Results

The proposed architecture is designed using verilog HDL, simulated using modelsim software and synthesized using Xilinx project navigator. The IC view is illustrated in fig.8 ,the signal window for way tagged cache with LUT architecture is illustrated fig .9 and the simulation results for way tagged cache access with LUT architecture is shown in fig.10. The power summary is shown in fig .11.

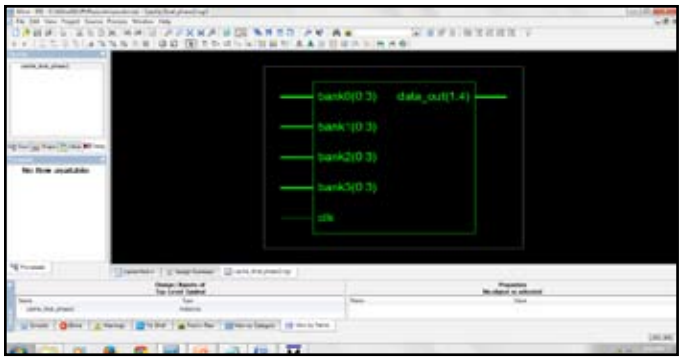


Fig. 8: Simulation results for cache IC View

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Fig. 9: Signal window of way tagged cache with LUT architecture

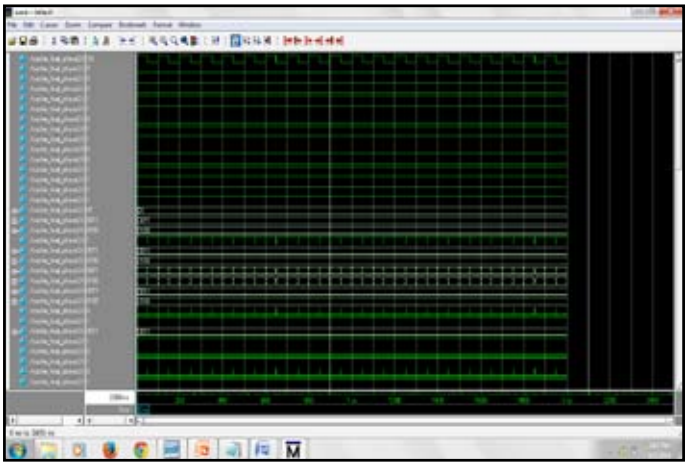


Fig.10: Simulation results for Way Tag cache access with shared LUT

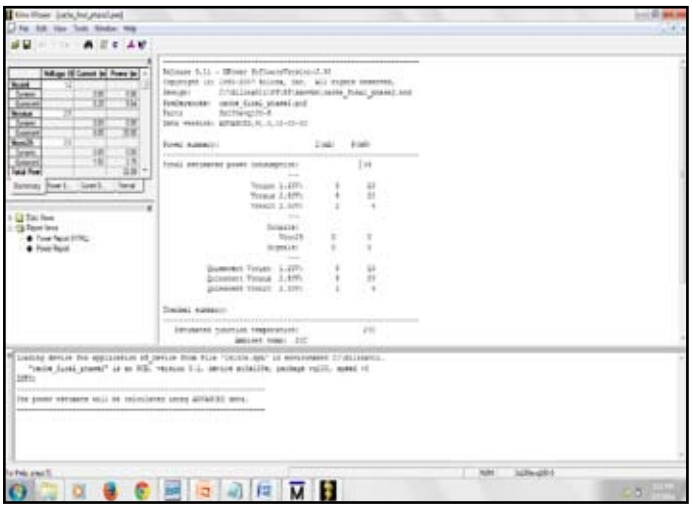
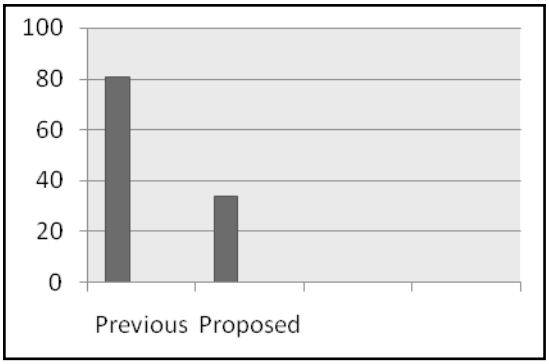


Fig.11: Simulation results for power consumption

Table 2 : Performance Analysis

Parameter	Previous	Proposed
Power Consumption	81mw	34mw

Graph I : Performance Analysis



When compared with the way-tagged cache with two existing low power cache design techniques named Phased access cache technique and Two level cache architecture, way tagged cache produce efficient energy reduction and reliably fast accessing of caches at L2 level. But after the implementation of shared LUT architecture the power estimation is further reduced up to one third of the phase one power summary results. From the performance analysis of way tagged cache, it produces the reduced number of slices up to 40 and LUT's up to 70 in number, as for as IOB is considered they are limited to 145. On the whole, efficient and eminent energy reduction up to 65.4% is achieved through this technique without performance degradation with low latency, less area overhead and low hardware utility.

IV. Conclusions

This paper presents a new energy-efficient cache technique for high-performance microprocessors employing the write-through policy. The proposed technique attaches a tag to each way in the L2 cache. This way tag is sent to the way-tag arrays in the L1 cache when the data is loaded from the L2 cache to the L1 cache. Utilizing the way tags stored in the way-tag arrays, the L2 cache can be accessed as a direct-mapping cache during the subsequent write hits, thereby reducing cache energy consumption. The shared LUT architecture replaced in the place of data array in which all the data information is loaded with table loader according to is index and coefficients for data finding and matching allocation during cache operations. the shared LUT architecture is partitioned in to four banks with respective address associated with it. If a processor needs to access data from any bank number, it will directly access that information via its coefficient bit address by matching with table loader indexes. Thus a long searching process is limited to direct accessing technique through shared LUT architecture. Simulation results demonstrate significantly reduction in cache energy consumption with minimal area overhead and no performance degradation. Furthermore, the idea of way tagging with shared LUT architecture can be applied to many existing low-power cache techniques such as the phased access cache to further reduce cache energy consumption.

V. Acknowledgement

References

[1] *T. N. Vijaykumar, —Reactive-associative caches,in Proc. Int. Conf. Parallel Arch. Compiler Tech., 2011, p. 4961.*

[2] *J. Dai and L. Wang, "Way-tagged cache: An energy-efficient L2 cache architecture under write-through policy," in Proc. ISLPED, 2009, pp. 159–164.*

[3] *F. X. Ruckerbauer and G. Georg akos, "Soft error rates in 65 nmSRAMs: Analysis of new phenomena," in Proc. IEEE Int. On-LineTest. Symp., 2007, pp. 203–204.*

[4] *G. H. Asadi, V. Sridharan, M.B. Tahoori, and D. Kaeli, "Balancing performance and reliability in the memory hierarchy," in Proc. Int. Symp.Perform. Anal. Syst. Softw. , 2005, pp. 269–279.*

[5] *X.Vera, J.Abella, A.Gonzalex and R.Ronen, "Reducing of error vulnerability of data caches," presented at the Workshop System Effects Logic Soft Errors, Austin, TX, 2007.*

[6] *P. Kongetira, K. Aingaran, and K. Olukotun, "Niagara: A 32-way multithreaded Sparc processor," IEEE Micro , vol. 25, no. 2, pp. 21–29,Mar. 2005.*

[7] *X. Vera, J. Abella, A. Gonzalez, and R. Ronen,"Reducingsofterror vulnerability of data caches," presented at the Workshop System Effects Logic Soft Errors, Austin, TX, 2007*

[8] *L. Li, V. Degalahal, N. Vijaykrishnan, M. Kandemir, and M.*

J. Irwin, "Soft error and energy consumption interactions: A data cache per-spective," in Proc. Int. Symp. Low Power Electron. Design, 2004, pp.132–137.

[9] K. Osada, K. Yamaguchi, and Y. Saitoh, "SRAM immunity to cosmic-ray-induced multi errors based on analysis of an induced parasitic bipolar effect,"IEEE J. Solid-State Circuits , pp. 827–833,2004.

[10] D. Wendell, J. Lin, P. Kaushik, S. Seshadri, A. Wang,V. Sundararaman,P.Wang,H.McIntyre,S.Kim,W.Hsu,H. Park,G.Levinsky,J. Lu, M. Chirania, R. Heald, and P. Lazar, "A 4 MB on-chip L2 cache for a 90 nm 1.6 GHz 64 bit SPARC microprocessor," in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papsers , 2004, pp. 66–67.

[11] J.Maiz, S.hareland, K.Zhang and P.Armstrong "Characterization of multi-bit soft error events in advanced SRAMs," in Proc. Int. Electron Devices Meeting , 2003, pp. 21.4.1–21.4.4.

[12] S. Rusu, J. Stinson, S. Tam, J. Leung, H. Muljono, and B. Cherkauer,"A 1.5-GHz 130-nm itanium 2 processor with 6-MB on-die L3 cache,"IEEE J. Solid-State Circuits,vol.38, no. 11, pp. 1887–1895, Nov. 2003.

[13] G. Konstadinidis, K. Normoyle, S. Wong, S. Bhutani, H. Stuimer, T.Johnson, A. Smith, D. Cheung, F. Romano, S. Yu, S. Oh, V. Melamed,S. Narayanan, D. Bunsey, C. Khieu, K. J. Wu, R. Schmitt, A. Dumlao,M. Sutera, J. Chau, and K.J. Lin, "Implementation of a third-generation1.1-GHz 64-bit microprocessor," IEEE J. Solid-State Circuits, vol. 37,no. 11,pp. 1461–1469, Nov. 2002.

Author's Profile

J.ISWARYA received her B.E. degree in Electronics and Communication Engineering from the Raja College of Engineering and Technology, Veerapanjan, Madurai, India, in 2011. Currently doing M.E - VLSI Design in Sethu Institute of Technology, Virudhunagar India. Her research interest includes: low Power VLSI and Image Processing

MR.B.MUTHUPANDIAN received his B.E., from College of Engineering in 19 and completed her M.E from College of Engineering and Technology in 2005.Presently working as an Associate Professor in the Department of ECE at Sethu Institute of Technology, Tamilnadu, India.