International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-5, March 2019

A Framework for Logically Reconfigurable Cache Memory for High Performance and Low Power Consumption in Modern Processors

Mayuri Chawla, Sanjay, M. Asutkar, Vijay S. Chourasia

Abstract: ***From last few decades computer system becomes an essential part of everyone's life and day by day its necessity is keep on rising. Computation and processing time is main source & plays a key role to achieve a better performance with very low power & energy consumption. In every electronic devices used in data processing & thus fast data processing requires frequent transfer of data from main memory to CPU. However the performance grid-back of different data-oriented applications is depends on various way of accessing the data cache. Thus, cache memory enhances the performance of processor by tens or hundreds of times much better. Hence the key technique is to diminish overall energy consumption is to vitally shut off the part of a processor's cache. Reconfigurable cache memory is vital to enhance the store execution and lessens the vitality utilization. In this paper, an audit for past papers related with reconfigurable store memory were given and analyzed it our work .This paper presents our proposed technique that reduces overall power consumption and improved performance in computer system if implemented with the help of some already existing architecture. Paper has introductory section followed by literature review, proposed work, then flow of execution, simulation flow and then results & analysis section which contains observation taken on by doing various simulations. Detailed block diagram describes the working of proposed technique, followed by execution flow diagram that depicts the flow of execution of our proposed technique then simulation flow which depicts the simulation process of technique , then result and analysis sections depicts the partial result of proposed technique ,after that there is a conclusion section followed by future work and limitation of out proposed technique.***

Keywords: Cache, computer system, pipeline, memory architecture, data processing.

I. INTRODUCTION

Now a days in day-to-day life even enhanced performance based general-purpose processors are utilized for an assortment of utilization spaces, including scientific research, building, exchange handling, and choice help. All the more as of late, media handling runtime application have received significant consideration. Media handling runtime application required testing & computational necessities and could utilize orders-of-size higher execution than accessible today.

Manuscript published on 30 March 2019.
*****Correspondence Author(s)
Ms. Mayuri Chawla, Department of Electronics and Telecommunication Engineering, Jhulelal Institute of Technology, Nagpur (Maharashtra), India.
Dr. Sanjay M. Asutkar, Associate Professor, Department of Electronics and Communication Engineering, Manoharbhai Patel Institute of Engineering and Technology, Gondia (Maharashtra), India.
Dr. Vijay S. Chourasia, Department of Electronics and Communication Engineering, Manoharbhai Patel Institute of Engineering and Technology, Gondia (Maharashtra), India.

Earlier, they were kept running on specific DSP processors or on the other hand ASICs. Be that as it may, the benefits of broadly useful processors regarding simpler programmability, upgradability, furthermore, higher execution development bends contend for the expanded utilization of such frameworks for media preparing.

A few quantitative portrayals have demonstrated that applications from different areas show different attributes. An impede for broadly useful frameworks is that they should perform all around ok over every single such trademark. As these frameworks are utilized for an inexorably wide assortment of utilizations, a "one-measures-all" plan logic will be deficient. Current framework structures frequently incorporate uncommon highlights focused to certain key applications; in any case, regularly, these highlights are actualized in an inflexible way and their assets are squandered for applications that can't straightforwardly use them. For instance, the utilization of expansive stores is a typical pattern crosswise over broadly useful frameworks, once in a while devouring up to 80% of the complete transistor spending plan also, up to half of the pass on territory [1]. An elective structure reasoning is to fabricate a few flexibility in the framework with the goal that includes that utilization a substantial no. of assets can be utilized in various courses by many of the applications. We apply this rationality to the structure of reserves and propose another reserve association that we call reconfigurable stores. The possibility of reconfigurable structures itself isn't new; in any case, regular reconfigurable structures commonly include substantial changes in both equipment and programming. Interestingly, our reconfigurable store configuration keeps on utilizing a customary reserve structure with minor equipment and delicate Our society depends all the more intensely on PCs, cell phones and chip as time passes for both of entertainment, communication or for every day working routines..These gadgets are in incredible interest and having multi-functionalities highlights due to its preparing abilities. An enormous measure of information has been produced and prepared in these sorts of gadgets, because of the handling velocity and capacity limit is influenced by the information handling. Information preparing and capacity assumes a vital job in separating profitable observations from these expansive amounts of information. to accomplish versatility, to enhance adaptation to non-critical failure, and to rationally increase the I/O throughput by using simultaneous & parallel access to different capacity cache memory is required.

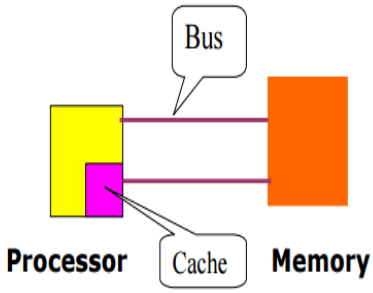

Figure 1: Cache Memory in Computer System

During study we find that Primary memory is almost moderate or low in operation where as cache memory has relatively high speed storage memory that creates figment high speed memory to the user of the system. Generally Cache memory stores information from repeated used memory location of primary memory but for some limited period of time which results data exchange at most extreme speed. Processor loads information from Main memory and duplicates into cache memory which leads litle bit delay in reading or writing of data called as miss penalty.

Hit ratio = % of memory accesses satisfied by the cache.

Miss ratio = 1 - hit ratio

Cache Bus - Cache is divided into different buses (additionally called fixed size blocks). Each bus has 4-64 bytes in it. During information exchange, an entire bus (line) is composed. Each line has a label that demonstrates the location in M from which the line has been replicated. Presently the high transistor thickness on current chip powers PC draftsmen to think about both power utilization and execution. Closing down parts of the microchip fills in as the least demanding, best instrument to ration control. Many broadly useful processors use this strategy [1, 2]. Since the storing structures on microchips utilize an expansive rate (up to 80%) of the transistors, closing down parts of the reserve would spare a lot of intensity [3]. Be that as it may, the span of the reserve incredibly influences execution, or the duration of period expected to execute programs. The ideal elite, less power cache memory will limit vitality utilization, or the result of intensity and processing time. This exploration will structure and assess another reserving method that powerfully closes down piece of a processor's store so as to decrease by and large vitality utilization. We likewise focus on a store controller that is powerfully reconfigurable as indicated by the running application and changes itself to give most elevated conceivable execution while utilizing least reserve. For as far back as of 37 years, Moore's law has precisely anticipated that the quantity of transistors on a solitary IC wills twofold at regular intervals [4]. Expanded transistor thickness has expanded working velocities at a similar rate, yet in addition caused much power utilization [5, 6]. This expanded power utilization creates undesired warmth, which possibly debases execution or decimates the IC. Generally, PC modelers have planned processors either for elite or for low power contingent upon the application. For instance, a wireless needs low power utilization with the goal that it won't consume the client's hand; in any case, a gaming console needs most extreme. Advances in media transmission and PC frameworks have pushed the improvement of new and complex inserted frameworks that go from modems and switches to PDAs, tablets and net-books. For instance, the advanced cell industry alone is required to move more than 500 million units amid 2012. To consider the fast improvement of implanted processors new registering designs are being produced. It ought to be seen that, regardless of some combination between universally useful processors (e.g., the Intel Core i7) and installed processors (e.g., the ARM Cortex or the MIPS32 processor family), these two kinds of processors are characteristically unique. The previous targets broadly useful applications with less power confinements. The last is proposed for explicit applications regularly with strict power and spending confinements. A standout amongst the most vital parts of any proficient computational framework is the memory chain of command subsystem. Nonetheless, since the fundamental essential memory is normally put far from the processor and works at an alternate recurrence, the entrance times for perusing or composing information into the primary memory are frequently high (e.g., many processor cycles). To alleviate this reality, reserve recollections are normally added to the processor to store a subset of the data in the fundamental memory. Since reserve recollections are set near the processor and can work at the processor recurrence, the entrance times to information in the store are short (commonly under 10 clock cycles). Along these lines, store recollections permit expanding the processor execution by tens or many occasions. This is particularly essential since the execution bottleneck of numerous information concentrated applications is really getting to the information. As referenced before, reserve recollections incredibly increment the processor execution however they are likewise in charge of a high bit of the processor's capacity utilization and chip zone (which influences processor cost). Along these lines, while expanding store sizes can prompt expanded processor execution, it additionally prompts a progressively costly and control hungry processor. To conquer this issue, reconfigurable store designs ought to be produced that can actualize dynamic calculations amid program execution. These calculations can likewise incorporate the shutting down of segments of the reserve when they are not being utilized. This would prompt huge enhancements in processor execution while diminishing the all out power utilization of the processor. Diminishing the vitality utilization of store in a framework on processor is one of the main important concerns of circuit originators in light of the fact that the complete intensity of store is commanded by cache. In spite of the fact that the working voltage reserve memory accessible in processor can be brought down to lessen the vitality utilization, the yield is altogether debased in light of the fact that store memory bit cells are regularly made out of least estimated transistors for high thickness, which makes store very powerless against process variety under a low working voltage.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Specifically, the deferral and vitality utilization in the bit line command the all out deferral and vitality utilization of reserve, separately, which implies that the postponement and vitality of store can be successfully enhanced by lessening the bit line swing amid the read activity. The move from scaling recurrence to scaling the quantity of centers proceeds with the pattern of worrying off-chip memory data transfer capacity and dependence on-chip reserves. Anyway the expenses of bigger reserves are noteworthy and developing. They normally involve 40– 60% of the chip zone and with spillage control surpassing exchanging power at sub-micron advancements; they are predominant buyers of vitality. Besides, examination of benchmarks have demonstrated that store usage is ordinarily low - underneath 20% for a larger part of benchmarks, with execution productivity averaging 4.7% and vitality effectiveness averaging 0.17%. This is in huge part because of the way that most of the reserve (particularly L2 and L3) is inactive more often than not yet contributes essentially to spillage control. The condition of the training in making reserves more vitality proficient has been to shut down store parts, for example, store lines, sets or ways - turn them off or keep up them in a low voltage state. Techniques center on when to kill which segments. Poor choices lead to costly misses and catalyst occasions and thusly procedures will in general be traditional.

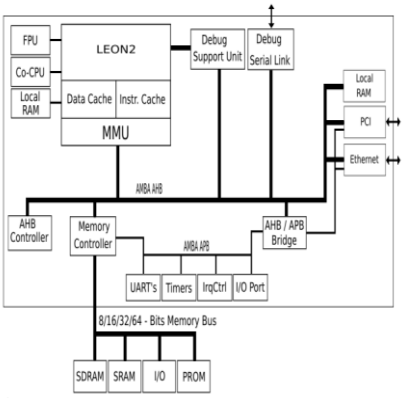

Figure 2: LEON System overview

In each framework reserve memory plays a vital and builds the processor execution yet in addition requires substantial sum processor's capacity and chip region .Hence, expanding store estimate drives better processor execution, yet it likewise prompts increasingly costly and control devouring processor. To take care of this issue, our proposed methods can be utilized which has reserve controller calculation excessively to powerfully stop the parts of some store unit to diminish processor control utilization dynamic calculations amid program execution. This calculation can likewise incorporate the shutting down of segments of the reserve when they are not being utilized. This would prompt critical enhancements in processor execution while diminishing the absolute power utilization of the processor. Figure.1 demonstrates a LEON framework that will be utilized to execute our proposed reserving method.

A. LEON System Specifications

Architecture	**32 Bit Risc**
Inst Set Architecture	SPARC V8
Pipeline Stages	5
FPU Support	YES
MMU	YES
Registers Available	40 to 520

Table 1: LEON System Specifications

B. Possible Cache Configurations

Number of Sets	**1-4**
Set Size	1 – 64 KB
Cache Size	1 – 256 KB
Replacement Policies	LRU
Write Policy	Write Through

Table 2: Possible Cache Configurations

II. LITERATURE REVIEW

In a numerous substantial information handling framework ,the storage memory add to an enhancement in the preparing execution. In, it was demonstrated that the capacity throughput and I/O reaction decide nearly the whole computational time. During a a normal cache memory access, generally few storage cache mechanism circuit is normally gets ON and other non-working parts does disperse some amount of power. There are so many circuit strategies has been developed to diminish leakage energy of unusable cache storage parts. Some techniques keeps the not working parts of cache storage in rest mode without concerning that whether the information should be held or not [6], while some of them characterize the less active mode for not-chosen parts and can save the condition of transistors of each storage cells [7]. In Non -active mode methods, the cell inclination voltage and power is decreased and it is braced at the dimension where the memory cell information can securely remain. Recently various techniques for sleepy mode has proposed from that one technique makes use of one sleep/rest transistor to set voltage limit to a higher than zero out of a sleep mode. Past ones is the known as reserve control minimization which is accomplished by determining and remunerating any adjustments in leakage current. The significant downside of this method the Opamp used in these techniques requires more consumption of DC power & current which must be imitated along every datum bank to give the required granularity. The dense speed - transistor permits SRAM circuitry to have adequate headroom for exchanging off transistor space, rapidly changing execution and constant drainage of current and power to accomplish dense thickness and enhanced execution/control. In paper[8] author proposes a Reconfigurable storage cache configuration that empowers the reserve SRAM clusters to be powerfully isolated into numerous blocks which is used by processor to perform various activities.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

In [9,10],author proposed an arrangement for reconfigurable storage with already set size, the store design can drawn in as prompt mapped save or as 2 way set familiar store, and can pick 1, 2, 4 or 8 words for each & every square for each different operational mode. The central point of the reconfiguration cache instrument is the Control Block. At the point when the working mode is picked, the reconfiguration methodology with no torpidity. By then, the save is set up to work in the new picked mode at the accompanying memory get to.

In paper [11] authors explained the amount of Energy Consumption in Reconfigurable cache in MPSoC Architecture: This is a approach consists of two-Level temporary cache Optimization used to determine the minimum estimation of energy consumed in Embedded system. Here the author has done researched to explore multilevel data storage temporary cache multiprocessor architecture by determining the minimum estimation of the energy consumption in while processing the data during system performance. In paper[10] authors presented a sort of creating Set and way management based data storage cache system for dynamic Run-Time reconfiguration of cache, which allows reconfiguration of cache storage with its size and associativity that results the energy-consumption delay of the smart data storage cache is on average of 16% much good than simple data storage cache reconfigurations.

In paper[12,13] all three authors has given introduction on new and unique concept of size-aware based storage cache management scheme which results the enhanced the performance of compressed data caches and given an average increase of 20.4% of effective data storing capacity while using Least-Recently Used (LRU) approach, and 25.8% over the Run-Time RE-Referred time-slot pre-assuming based scheme.

In paper [14] author described a storage structure based technique which will divide a subset of the routes in a set cooperative reserve amid the times of unobtrusive reserve action, still making the full storage cache memory circuits active & operational for different main operations of data processing. In paper[14] authors proposes a calculation for reconfiguration scheme termed as Adaptive Balanced Computing(ABC) techniques which does runtime data asset design based on interest of data from various application stored on cache memory and then does processing data assets. In any point if cache storage reserve request is less, then this data cache storage can be usable for other runtime system and user applications.

III. PROPOSED WORK

In this paper we introduces an active run-time re-sizable information/Data storage cache model which is normally uses the comparable thought of AMC, so it will have enhanced performance outcome than various static model and non-active cache model generally used IC chip and also try to overcome the weaknesses of AMC by using different technique. The significant parameters of our data cache configuration incorporate various block size, number of sets and their associativity. Progressively changing and updating these value would all be able to change the size of the cache storage and afterward lead to enhance the cache storage performance and lesser the consumption of energy utilization in cache. Yet, in for all intents and purposes, changing size of block or updating the number of set is genuinely a troublesome issue. As we know that changing or updating the set of number or size of block will influence the no. of required tag bits, for example expanding the size of blocks will decreased the no. of necessary tag bits requirement and its vice-versa. Hence the updating or changing the size of block or no. of set will add more difficulties and complexity of the tag comparison module because of the differing tag dimension. Hence we will not consider fluctuating size of blocks or no. of set in our practical experiments.

A. Cache Memory:

The cache memory system is a Harvard design with 1-64 Kbyte per path for both the guidance and the information store. Each store line can contain somewhere in the range of 4 and 8 sub-obstructs, each containing 4 byte. The stores can be arranged as it is possible that single direction coordinate mapped or 2-4 way set acquainted. In a multi-way design three substitution strategies can be utilized: least as of late utilized (LRU), last as of late supplanted (LRR) and pseudo-irregular. When utilizing LRR, just two way set associativity is accessible. For decreased reserve miss idleness the guidance store utilizes gushing amid line-refill, which implies that information is sent to the processor in the meantime as it is kept in touch with the store. On a store miss in the cache memory, just the asked for sub-square is brought. The information store utilizes compose through strategy. To limit pipeline slows down brought about by store directions, a twofold word compose cushion is utilized. The reserves have bolster for individual store line securing multi-way setups. To keep a particular memory deliver to be hindered from the reserve, the last line in each set cannot be bolted. So as to support reserve consistency when there are a few units on the AHB (AMBA High Speed Bus) transport fit for keeping in touch with the memory, the information store can perform transport snooping on the AHB transport. Transport snooping is just accessible when the MMU is handicapped, since the reserve is for all intents and purposes tended to.

IV. EXECUTION FLOW

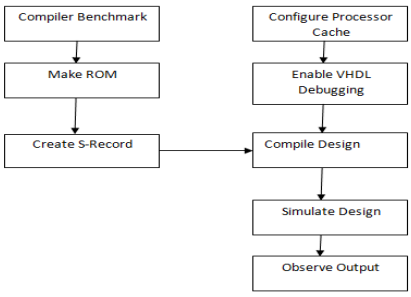

Figure 3: Flow of Execution

The above Figure, demonstrates the working process of our research work yield show. In this work we will initially consider distinctive compiler configuration to mention different objective facts .

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Our work stream in two sections at the same time where we will do different compiler structure and designing processor reserve. Store framework is Harvard engineering with 1-64 Kbyte per path for both the guidance and the information reserve. Each reserve line can contain somewhere in the range of 4 and 8 sub-obstructs, each containing 4 byte. The reserves can be designed as it is possible that single direction coordinate mapped or 2-4 way set cooperative. In a multi-way design three substitution approaches can be utilized: least as of late utilized (LRU), last as of late supplanted (LRR) and pseudo-irregular. When utilizing LRR, just two ways set associativity is accessible. For decreased store miss inactivity the guidance reserve utilizes spilling amid line-refill, which implies that information is sent to the processor in the meantime as it is kept in touch with the reserve. On a store miss in the information cache, just the asked for sub-block is brought. The information reserve utilizes compose through strategy. To limit pipeline slows down brought about by store guidelines, a twofold word compose support is utilized. The stores have bolster for individual reserve line securing multi-way arrangements. To keep a particular memory deliver to be hindered from the store, the last line in each set can't be bolted. So as to support reserve consistency when there are a few units on the AHB (AMBA High Speed Bus) transport fit for keeping in touch with the memory, the information store can perform transport snooping on the AHB transport. Transport snooping is just accessible when the MMU is debilitated, since the store is for all intents and purposes tended to.

A. CONFIGURABLE CACHE ARCHITECTURE

The configurable system cache scheme incorporates four different parameters: size of Cache, which can be designed as 2 KB, 4 KB, or 8 KB; their associativity or relativity, which can be defined as 1-way, 2-way, or 4-way (directly mapped); Size of line, which can be of 64 bytes, 32 bytes, or of 16 bytes; and prediction of way , which can be ON or turned OFF (however if the storage cache is designed and set as direct mapped then it will be constantly off).Our experimental model /Architecture is referred from the proposal given by the Spanberger. We will try to to create the comparatively similar model with various parameters used in data cache. As described in the Spanberger proposal, we will also make use of dynamic changeable associativity to update the size of cache. However our proposed study is little bit different in terms of storage cache processing because in Spanberger proposal they had studied on instruction cache but we will proceed our study on data cache in processor system which makes our idea of study unique. In our early research study we found that Instruction cache is significantly more linear than the data cache because linear property of instruction. So have to find out that whether a similar thought is useful for data cache or not. After finding this fact and answer of the question discussed above we will start our research on VHDL debugger for troubleshooting the code. After doing the thorough research study on compiler design, design issues and parameters and different types cache ,we will propose the design of our model and will to do simulation based on various parameters of designing the cache studied in literature review on simulator and will do analysis on results obtained .

V. SIMULATION FLOW

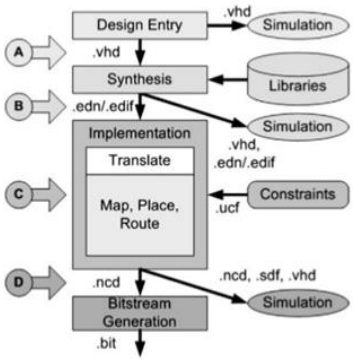

Figure 4: Simulation Process of Reconfigurable Cache design

Fig.4 shows the simulation execution steps for the model we will suppose to design and implement.

Step 1]. Shows the design entry of our proposed model. In second step for simulation we will use different available libraries to synthesis the design.

Step 2]. Here we will actually implement our model using translators, interpreters, routing, mapping .the output of this step will be the .ncd file. Here we will also consider some limitation of our model.

Step 3]. The .ncd bile generated in 3 steps will be given as input to further processing to generate the bit stream data which will decide how fast the transfer of bit stream is flow using our resizable cache model.

VI. RESULT AND DISCUSSION

In order to do experiment, generate result and perform analysis the multiple attributes (i.e, Associativity, Cache Line and Bytes per line) are consider for simulation using Dhrystone technique.

6.1 Result # 1 Dhrystone Result with 1024 cache size:

In simulation we have taken cache size 1024 bytes, which transfers 32 bit per lines with associativity as1 and generated Dhrystone result as 586.

Cache Size	1024 Bytes
Associativity	1
Cache Lines	32
Bytes per Line	32
Dhrystone Result	586

Table 3: Dhrystone Result with 1024 cache size

6.2Result # 2 Dhrystone Result with 512 cache size:

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

In simulation we have taken cache size 512 bytes, which transfers 16 bit per lines with associativity as 1 and generated Dhrystone result as 572.

Cache Size	512 Bytes
Associativity	1
Cache Lines	16
Bytes per Line	32
Dhrystone Result	572

Table 4: Dhrystone Result with 51 cach

VII. CONCLUSION

In our research work , we have compared different cache memory based on their size, based on which we can conclude that proposed technique if implemented will frequently increases the performance of computer system to some extent , although our proposed technique is yet to be tested in some other system with different specification too. In our future work we will take some more simulation and will observe and discuss more and different result of this technique. Having gone through the different techniques available to maximize the cache usage at a minimal power we observe that some techniques work on resizing of instruction cache ,some work on specific circuit control mechanism and some other work on the different algorithm and all of the scheme work on the criteria of improving the performance in terms of reducing the power requirement, the area required and the reduction of the processing time and area. After doing doing thorough study we come to a conclusion that many application consume excessive energy even if they are using the cache or not. Hence there is a need to present a technique that dynamically provides the cache and shuts off the processor cache if not in use to save the power and energy consumption.

FUTURE & SCOPE

Despite the fact that we have thought about various store estimate, still we have to check and looked at the quantity of sets ,set size ,line size of reserve and distinctive substitution strategies in future work which will assist us with deciding which reserve setup is best for which application in various PC framework. Limit of our proposed work is applicable to different microprocessor or micro controllers which are normally used to run some small computer applications. Since this configuration is application depended, the processor outcome can't be enhance simply by reducing the cache capacity While the processing. As we know cache always needs more power to be an active and if the cache which is not in active on not in use if get totally powered off will lead reduced overall energy consumption. Hence our proposed dynamic reconfigurable cache technique will enable the design to accomplish comparatively enhanced performance when application requires more cache and will lead lesser power dispersal when less data cache is required..

REFERENCES

1. S. P. Dandamudi, "Fundamentals of Computer Organization and Design", 2nd ed. New York, USA: Springer, 2012.
2. Albonesi D.H, "Selective cache ways: on demand cache resource allocation," in Proc. 32nd Inter-national Symposium on Microarchitecture, 1999.
3. Santana Gil, A.D., Benavides, Hernandez, Herruzo, "Reconfigurable Cache implemented on an FPGA" International Conference on Reconfigurable Computing and FPGAs, IEEE. 2010.
4. Jungwoo Park, Jongmin Lee and Soontae Kim, "A Way-Filtering-Based Dynamic Logical–Associative Cache Architecture for Low-Energy Consumption". In IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, 2017, pp.793-805.
5. Vincent Heuring, and Harry Jordan,"Computer Systems Design and Architecture", Massachusetts: Addison-Wesley, 1997.
6. Michael Powell, Se-Hyun Yang, Babak Falsafi, Kaushik Roy and T. N.
7. Vijaykumar. "Gated-Vdd: A Circuit Technique to Reduce Leakage in Deep- Submicron Cache Memories ." Proceedings of the

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)