International Journal of Engineering Science Invention (IJESI)
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 7 Issue 2Ver III || February 2018 || PP. 39-51

Exploration Tool for Instruction Memory Organisations Based On Accurate Cycle-Level Energy Modelling

Prof.Dr.G.Manoj Someswar[1], A.Suman Kumar Reddy[2]
1. Research Supervisor, Dr.APJ Abdul Kalam Technical University, Lucknow, U.P., India
2. Research Scholar, Dr.APJ Abdul Kalam Technical University, Lucknow, U.P., India
Corresponding Author: Prof.Dr.G.Manoj Someswar

ABSTRACT: The plan of current implanted frameworks is obliged by the necessities of present day installed applications. A considerable lot of these applications require not just supported operation for drawn out stretches of time, yet additionally to be executed on battery controlled frameworks. Under the imperative of not being mains-associated, the nonappearance of wires to supply a steady wellspring of vitality causes that the utilization of a vitality gathering source or an incorporated vitality provider restrains the operation time of these electronic gadgets. Direction memory associations are called attention to as one of the real wellsprings of vitality utilization in installed frameworks. As these frameworks are portrayed by prohibitive assets and a low-vitality spending plan, any improvement that is presented in the direction memory association permits to diminish the vitality utilization, as well as to have a superior circulation of the vitality spending plan all through the installed framework. This Ph.D. theory concentrates on the examination, investigation, proposition, usage, and assessment of low-vitality streamlining procedures that can be utilized as a part of the guideline memory associations of implanted frameworks. Genuine installed uses of the specific sub-domain of remote sensor hubs are utilized as benchmarks to appear, examine, and authenticate the benefits and disservices of every last one of the ideas in which this Ph.D. proposal depends on.

KEYWORDS: *ITRS (International Technology Roadmap for Semiconductors), DMH (Data Memory Hierarchy), IMO (Instruction Memory Organization), LB (Loop Buffer), CELB (Central Loop Buffer Architecture for Single Processor Organization)*

Date of Submission: 09-02-2018 Date of acceptance: 24-02-2018

The first key commitment is the efficient investigation of existing low-vitality improvement systems that are utilized as a part of guideline memory associations, delineating their similar points of interest, disadvantages, and exchange o s. Over that, the exploratory assessment that is introduced in this Ph.D. proposition utilizes a precise strategy with a specific end goal to have an exact estimation of parasitic and exchanging movement. Because of this reality, this assessment guides implanted frameworks architects to settle on the right choice in the exchange o s that exist between vitality spending plan, required execution, and region cost of the inserted framework. The second key commitment is the improvement of an abnormal state vitality estimation device that, for a given application and compiler, permits the investigation of building and compiler configurations, as well as of code changes that are identified with the guideline memory association. The third key commitment is the proposition and examination of a few promising usage of vitality efficient direction memory associations for a specific set of utilization codes and installed designs. In view of the past commitments, the work that is exhibited in this Ph.D. proposition demonstrates why additionally upgrading direction memory associations from the vitality utilization perspective will remain a critical pattern later on.

IMOSIM: Exploration Tool for Instruction Memory Organisations based on Accurate Cycle-Level Energy Modelling

Technology is nothing. What's important is that you have a faith in people, that they're basically good and smart, and if you give them tools, they'll do wonderful things with them. Due to the fact that the design space of the enhancements for reducing the energy consumption of the instruction memory organisation is huge, this Chapter proposes a high-level energy estimation tool that, for a given application and compiler, allows the exploration not only of architectural and compiler configurations, but also of code transformations that are related to the instruction memory organisation. The proposed tool, with a mean error of ^{3}:^{95} %, achieves reductions in time and effort to explore the design space of the instruction memory organisation.

Introduction and Related Work

Installed frameworks request different equipment structures to run applications that range from sight and sound shopper gadgets to industry control frameworks. In any case, not at all like broadly useful PC frameworks, implanted frameworks need to give high-calculation capacity, unwavering quality, consistency, and meet continuous imperatives with restricted assets, as well as a low-vitality spending plan. The mix of past

prerequisites and imperatives influences the decrease of aggregate vitality utilization to end up noticeably a major test for inserted frameworks architects.

As appeared in an ordinary inserted framework is made out of a processor design, a DMH (Data Memory Hierarchy), an IMO (Instruction Memory Organization), and a between centre correspondence arrange. For more subtle elements see Figure 1 Research works like showed that both memory gadgets take significant bits of chip range, as well as now represent up to 40 %60 % of the aggregate vitality spending plan of an implanted guideline set processor stage (see Figure 1). The vitality utilization of the IMO isn't insignificant and should be streamlined keeping in mind the end goal to diminish the general vitality utilization of the installed framework. With a specific end goal to decrease the vitality utilization of the IMO, implanted frameworks planners adjust or potentially parcel the IMO. From one viewpoint, circle buffering is a decent case of effective plan for the modification of the progressive system that exists in the IMO. J. Kinfolk et al. demonstrated that putting away little program portions in littler memory (e.g., as a LB (Loop Buffer)), the dynamic vitality utilization of the framework was diminished significantly. Then again, saving money is a decent case of effective technique for the apportioning of the IMO. Aside from the likelihood of utilizing various low-control working modes, the utilization of banks diminishes the effective capacitance when contrasted with a solitary solid memory, which prompts promote vitality decreases. Area talks about widely these equipment enhancements of the IMO. The ITRS (International Technology Roadmap for Semiconductors) predicts that the vitality utilization of installed frameworks will proceed with its quick development for the following decade due to their regularly expanding multifaceted nature and size. Aside from the vitality utilization, unwavering quality and consistency are likewise getting to be plainly basic, in light of the fact that them two are specifically identified with the vitality utilization and its dispersion over the framework. Because of these issues, an abnormal state vitality estimation apparatus is required amid the outline procedure of an inserted framework to build the re-enactment speed and the vitality funds. Exact vitality models must be made for each occurrence that forms the IMO, keeping in mind the end goal to appraise at abnormal state the aggregate vitality utilization of the IMO. The vitality estimation is performed in light of recreations that consider the joining of the vitality models of each occasion that makes the IMO. Past works have demonstrated refined vitality displaying strategies to correctly evaluate the vitality utilization at framework level. Notwithstanding, these techniques do not have a plan space investigation of the different IMO architecture options from the energy consumption point of view. In some cases, previous works use relatively basic IMOs (e.g., based on L1 instruction caches) avoiding the more complex IMO architecture options that are potentially much more energy-efficient. In other cases, the presence of the IMO in the system is completely ignored, because its energy consumption is counted as a part of the energy consumption that is assigned to the operations that are executed on the processor architecture.

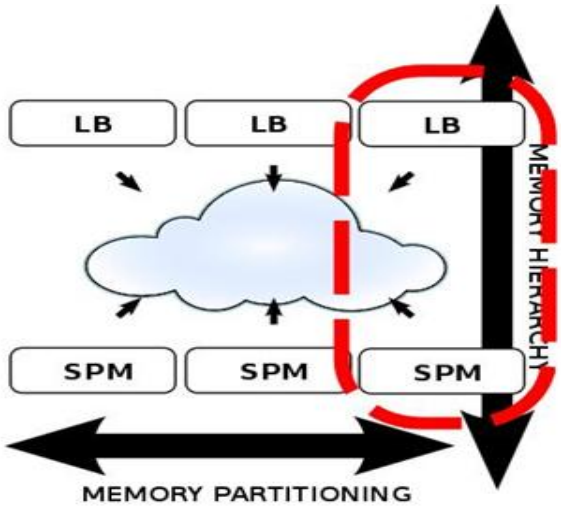


Figure 1: Memory hierarchy in the instruction memory organisation

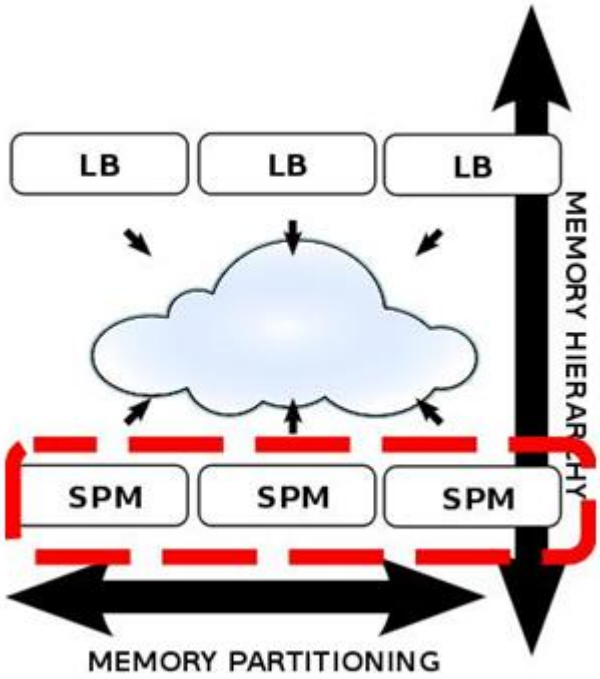


Figure 2: Memory partitioning in the instruction memory organisation

This Chapter proposes an abnormal state vitality estimation and investigation device that investigates the different design and compiler configurations that can execute the IMO. Figure 2 demonstrates the piece graph of the proposed abnormal state vitality estimation and investigation device. As can be found in this Figure, this apparatus consequently forms a given application in view of its qualities, a given processor engineering in view of its necessities, and a given power utilization library of different memory occasions, so as to help implanted frameworks architects to find the enhanced configuration of the IMO for the aggregate vitality utilization of the inserted framework. The proposed apparatus, with a mean mistake of 3:95 %, accomplishes significant diminishments in time and effort to investigate the outline space of the IMO.

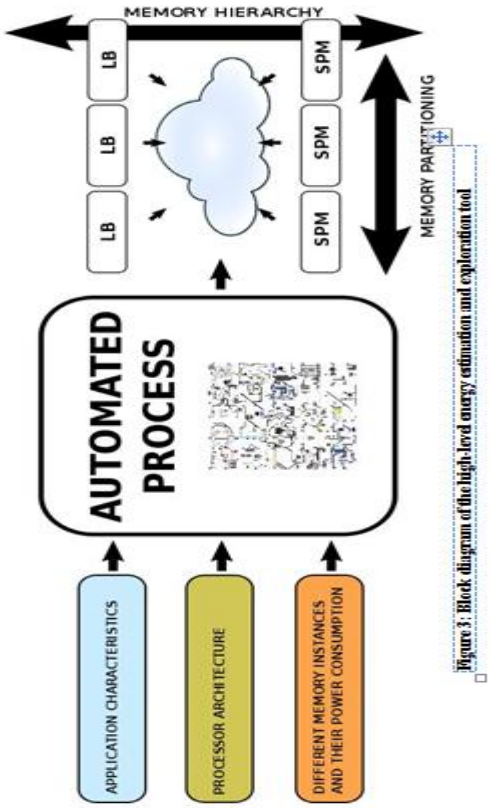
Design Space of the Instruction Memory Organisation

The work that is exhibited in this Chapter depends on a compositional classification that contains three noteworthy situations where vitality efficient circle arranged applications and stages would be able to in. In the accompanying passages, the agent engineering of the IMO of every situation is clarified in detail.

The CELB (Central Loop Buffer Architecture for Single Processor Organization) speaks to the most customary utilization of the circle buffer idea. For more subtle elements see in Figure 3 demonstrates the bland design of this IMO, which neither has apportioning on the up and up buffer engineering nor in the PM (Program Memory), and its associations rely upon a solitary incorporated segment. Thusly, parallelism in the execution of an application can't be accomplished in this design.

The brought together assets and worldwide correspondence of single-strung designs make CELB models less vitality efficient, when strategies of circle changes are connected to abuse parallelism inside circles.

The CLLB (Clustered Loop Buffer Architecture with Shared Loop-Nest Organization) mitigates these bottlenecks. For more points of interest see in Figure 3 demonstrates the CLLB engineering, which internal associations are controlled by one single part. For this situation, the controller is more unpredictable, in light of the fact that it controls the allotments that exist on top of it buffer design and program memory.



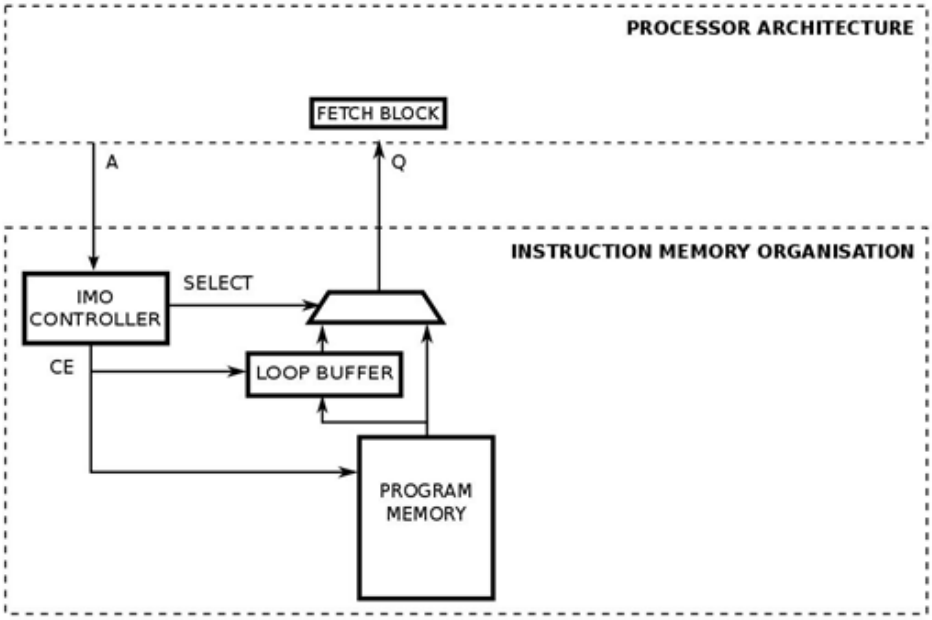


Figure 4: Instruction memory organisation with a central loop buffer architecture for single processor organisation

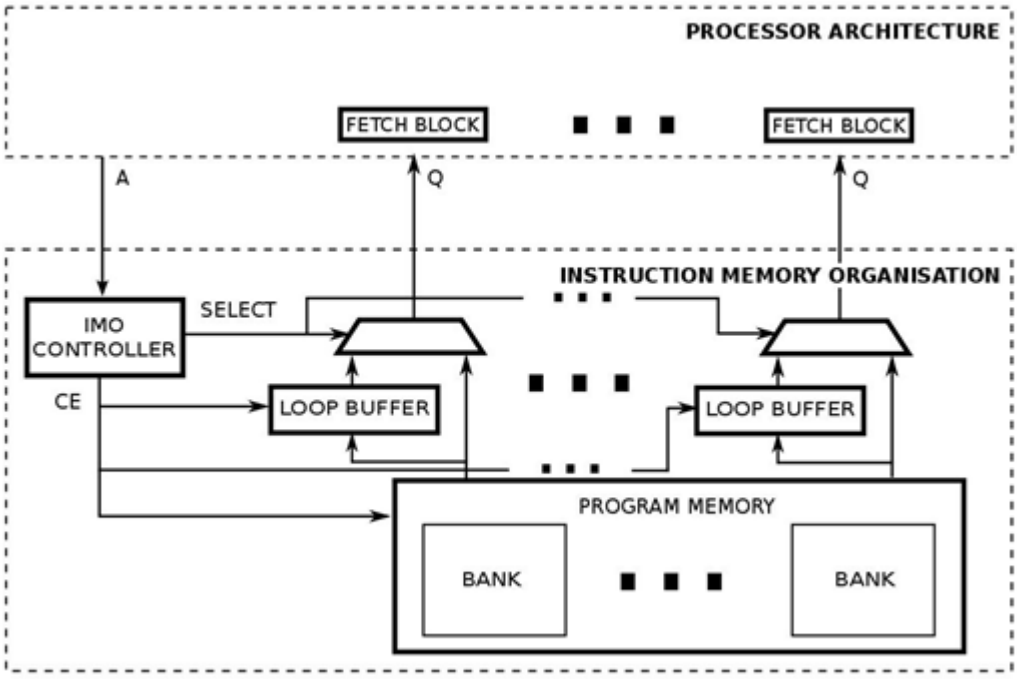


Figure 5: Instruction memory organisation with a clustered loop buffer architecture with shared loop-nest organisation

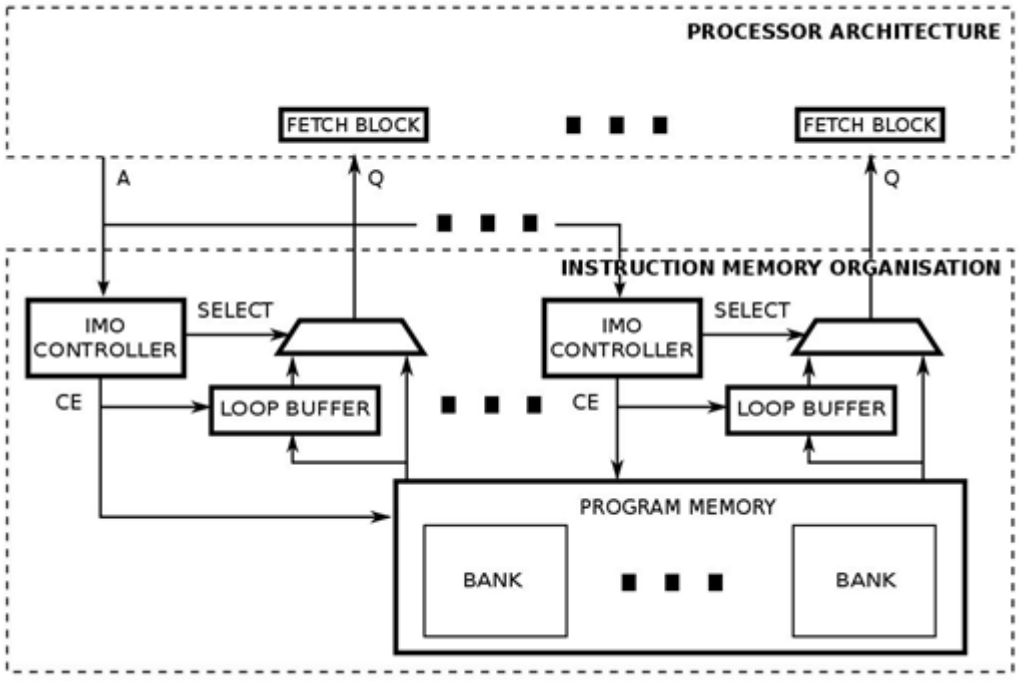


Figure 6: Instruction memory organisation with a distributed loop buffer architecture with incompatible loop-nest organisation

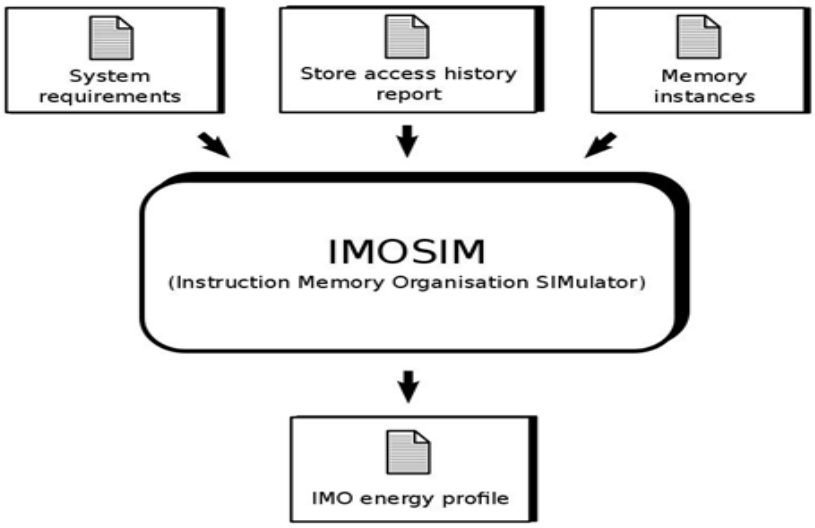
Proficient parallelism isn't accomplished with a CLLB engineering, because of the way that circles with different strings of control must be converged into a solitary circle with a solitary string of control. The required code change is performed utilizing systems like circle changes (e.g., circle combination). In any case, not all circles of an application can be efficiently abused in this way. On account of incongruent circles, parallelism can't be efficiently abused, on the grounds that they require different circle controllers, which brings about loss of vitality and execution. In this manner a need exists for a multi-strung stage, that could bolster execution of various contradictory circles with insignificant equipment overhead. This issue is understood by a DLB (Distributed Circle Buffer Architecture with Incompatible Loop-Nest Organization). In this circle buffer design, not just the circle buffer recollections are conveyed over the IMO, yet in addition the circle buffer controllers that oversee them. Hence, in this unique arrangement of circle buffer designs, each circle buffer memory has its own particular neighbourhood circle buffer controller. Because of this reality, DLB models can work like multi-strung stages permitting the execution of incongruent circles in parallel with negligible equipment overhead. For more points of interest see in Figure 6 demonstrates the non specific engineering of this current delegate circle buffer design. As appeared in this Figure, the inward associations of this IMO are overseen by a rationale of controllers that is disseminated over the design. In this IMO, segment exists in both the circle buffer engineering and the program memory. The controller of this IMO is much more mind boggling than the controllers of the past ones, since it likewise controls the execution of each circle in the comparing circle buffer design to permit the parallel execution of circles with different iterates. The vitality estimation and investigation instrument that is proposed in this Chapter is the just a single, known by the creator, that permits the reproduction of this novel IMO from the vitality utilization perspective.

IMOSIM (Instruction Memory Organization SIMulator)

IMOSIM (Instruction Memory Organization SIMulator) is an abnormal state vitality estimation device that, for a given application and compiler, investigates different designs and configurations that can make the IMO. This investigation causes implanted frameworks planners to find the design and compiler configuration that is ideal from the perspective of the aggregate vitality utilization of the installed framework. So as to play out an entire plan space investigation of the IMO, the delegate models that are appeared are utilized to emulate each circle buffer engineering that is as of now distributed in writing. As appeared in Figure 7, IMOSIM requires three contributions to content le organization to know the necessities of the implanted framework engineering and the inserted application under re-enactment: the store get to history report of the application, the prerequisites of the installed framework, and the cycle-level vitality models of every single one of the segments that form the agent IMOs that are depicted. The client has as yield not just the content les that contain the vitality profiles of the segments that shape the agent IMOs, yet in addition the diagrams that speak to these information. Figure 8 is a decent case of chart that implanted frameworks planners can get from IMOSIM.

The portrayal of the framework is provided to IMOSIM through the data of the framework necessities and the store get to history report. The framework prerequisites data is a total depiction of the interface between the processor engineering and the IMO. This info gives the width and number of directions that the processor engineering requires keeping in mind the end goal to execute the given application, and in addition the recurrence that is utilized as a part of the framework. Along these lines, this info gives to IMOSIM the depiction of the between centre correspondence arrange that exists in the framework. The store get to history report gives to IMOSIM the profile data that is identified with the run-time conduct of the application. This info data, that is produced by the same ISS (Instruction Set-Simulator) that is utilized to certify the right usefulness of the framework, contains the memory tends to that are gotten to by the processor design in every execution cycle.

Figure 7: Flowchart of IMOSIM



The portrayal of the memory cases that can make the IMO is provided to IMOSIM through cycle-level vitality models. These models can be acquired in two ways. Either from the information sheets of the business recollections that the installed frameworks fashioner needs to utilize, or making the vitality models of the memory occurrences from post-design back commented on vitality recreations. In this last alternative, RTL (Register-Transfer Level) recreations must be performed to create exact vitality waveforms of the memory case under displaying with some client specified preparing VCD (Value Change Dump) les. Accordingly, the vitality utilizations that are identified with the different sort of gets to that the memory cases have in their various operation modes are known through pattern line conditions that are acquired from discrete estimations by polynomial bend fitting. The level of the polynomial condition is picked by the implanted frameworks originator in light of the coveted level of precision. With a specific end goal to cover the entire outline space of a memory case, varieties top to bottom, width, and innovation utilized as a part of its execution are finished. In spite of the fact that, the vitality models of the memory controllers contain the control rationale of the associations between different examples and the enactment of the different operation modes, these models are straightforwardly identified with the parameters that define the memory cluster. Subsequently, a similar strategy is taken after to determine the conditions that depict the vitality utilization of this some portion of the IMO. IMOSIM models the memory controllers of the circle buffer structures as takes after:

CELB designs. The rationale of the memory controller controls the associations with initiate the entire circle buffer design when the execution of a circle is recognized, and additionally the banks that can make the program memory.

CLLB designs. The memory controller of these agent structures expands its rationale to control likewise the enactment of the circle buffer recollections that are within the circle buffer design. DLB designs. Aside from the past capacities, the memory controller of this engineering can control each circle buffer memory of the circle buffer design isolate. This reality permits the parallel execution in this circle buffer design of circles with different iterations and body estimate.

IMOSIM forms its three sources of info in light of scientific conditions that consider the condition of the IMO in the specific execution cycle that is assessed. The circle buffer configuration is thought to be fixed amid the recreation. Because of this requirement, a constrained arrangement of parts can be decided for a specific circle. A split of the information circle is defined as a vector St <n, where (St)i is the quantity of guideline words that are put away in bank I at time t, and n is the quantity of banks that frame the circle buffer memory. Condition portrays formally the scope of sizes that a split can have. The measure of each split is thought to be ceaseless and ranges from zero to the estimation of the span of the greatest bank (Bmax).

0 St Bmax 8t (1)

In the energy models, the size of each bank that forms the loop buffer memory is defined as a vector Bi <, where Bi is the size of the bank i. The sizes of the banks are assumed to be continuous and range from zero to a maximum size value Bmax as it is described in Equation. The value Bmax as well as the size of each bank are selected to keep the addressing logic as simple as possible.

0 Bi Bmax 8i (2)

The accesses are defined as a vector At $<^n$, where (At)i is the number of accesses to the bank i at time t, and n is the number of banks that form the loop buffer memory. The relation between the accesses to a memory bank (At)i and the possible splits of the input loop (St)i is expressed by Equation 3. From this Equation, it is possible to derive that A S.

At = St $_{()}$ (St)i Bi 8i (3)

The aggregate vitality utilization of every last one of the memory examples that make the IMO is defined, as appeared in Equation 4, as a vector Ei <, which is displayed as summation of dynamic vitality utilization and static vitality utilization. From one viewpoint, the dynamic vitality utilization (Edynamic)i relies upon the sort of access (At)i that is performed in the memory case Bi. Then again, the spillage vitality utilization (Eleakage) I is made out of the vitality that is expended in the working modes that are initiated amid the execution of the cycle. Dynamic and spillage vitality utilization are depicted, where (At)i is a vector that demonstrates which sort of access is being performed in the memory occurrence, and Ct is a vector that shows the operation mode (dynamic, o, and maintenance) of the memory case for the specific execution cycle in which the entrance to the memory example is being performed. The yield of IMOSIM is the assessment of Equation 4, Equation 5, and Equation 6 for every execution cycle of the application under re-enactment.

Ei = (Edynamic)i + (Eleakage)i (4)
(Edynamic)i = (Ewrite)i(At)i + (Eread)i(At)i (5)
(Eleakage)i = ((Eact)i + (Eoff)i + (Eret)i)Ct (6)

For ultra-low vitality implanted frameworks, the temperature varieties that are caused by the running of these frameworks are irrelevant. Moreover, the most extreme temperature, that these frameworks can accomplish, is lower than the temperature in which unwavering quality disappointments can show up. Because of the way that the present acknowledgment of IMOSIM is centred around ultra-low vitality implanted frameworks, spillage based warm reliance and PVT (Process, Voltage, and Temperature) varieties don't should be considered yet by this adaptation of IMOSIM. Target Compiler Target Compiler Technologies

Experimental Result

In this exploratory system, the processor engineering is planned utilizing. For more data see Appendix B. Hence, keeping in mind the end goal to confirm the right usefulness of the framework as well as produce the store get to history report, the ISS of Advances is utilized. The assessment is performed utilizing TSMC (Taiwan Semiconductor Manufacturing Company) 90nm LP (Low Power) libraries and business recollections. A clock recurrence of 100MHz is chosen. Genuine installed applications in the ultra-low vitality space are utilized as benchmarks to assess the abnormal state vitality estimation and investigation instrument that is proposed in this Chapter. The chose benchmarks, which can be found in Table 1, are prime cases not just of all application areas that are circle commanded, display sufficient open door for information as well as guideline level parallelism, include signals with various word-lengths, and require a moderately set number of variable increases, yet in addition of more universally useful applications spaces that can be found in the range of remote base-band flag handling, sight and sound flag preparing, or different sorts of sensor flag handling. The benchmarks are chosen with the objective of demonstrating the considerable flexibility of IMOSIM.

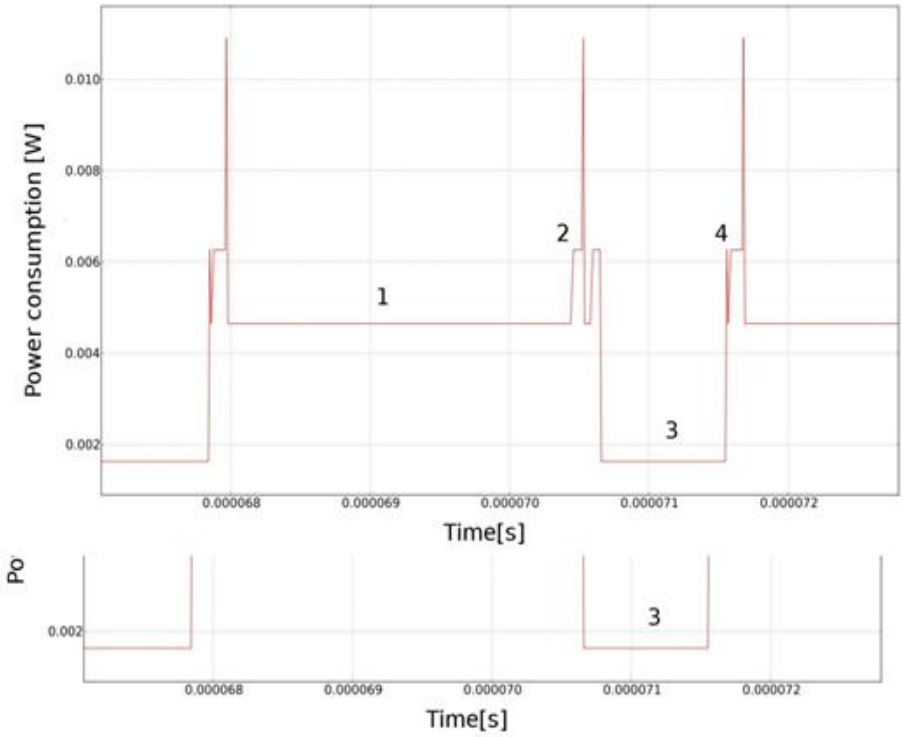


Figure 8: Run-time behaviour of IMOSIM showing the power consumption of an IMO based on a CELB architecture

The run-time output of IMOSIM with the benchmark AES (see Section C.3) is presented in Figure 8. This Figure shows an overview of how the power consumption of the IMO changes every cycle depending on the components that are active. In region 1 of this Figure, the instructions are only fetched from the program memory, because during the execution of non-loop parts of the application code, instructions are fetched directly from this memory. In region 2, the IMO detects that a loop is been executed, and therefore, the instructions are fetched from the program memory to both the loop buffer architecture and the processor architecture. In this loop iteration, the loop buffer architecture records the instructions of the loop. Once the loop is stored, the execution of the loop is in region 3, where the instructions are fetched from the loop buffer architecture instead of the program memory. In the last iteration of the loop, which can be seen in region 4, the connection between the processor architecture and the program memory is restored, such that subsequent instructions are only fetched from the program memory

Table 1: Profiling information of the benchmarks that are used in the experimental evaluation of IMOSIM

Benchmark [Reference]	Cycles	Issue Slots	Bits per Instruction	LB Size [Instructions]	Loop Code [%]	NOP Instructions [%]
AES [TSH+10] (See Section C.3)	3; 347	1	16	32	77.44	0.09
BIO-IMAGING [PFH+12] (See Section C.4)	334; 071	4	80	64	98.01	26.70
CWT NON-OPTIMISED [YKH+09] (See Section C.5)	274; 464	1	16	32	6.61	0.48
CWT OPTIMISED [YKH+09] (See Section C.6)	102; 827	1	20	64	6.36	2.50
DWT [DS98] (See Section C.7)	758; 216	2	16	518	65.70	25.19
MRFA [QJ04] (See Section C.9)	177; 170	2	32	64	19.13	17.01

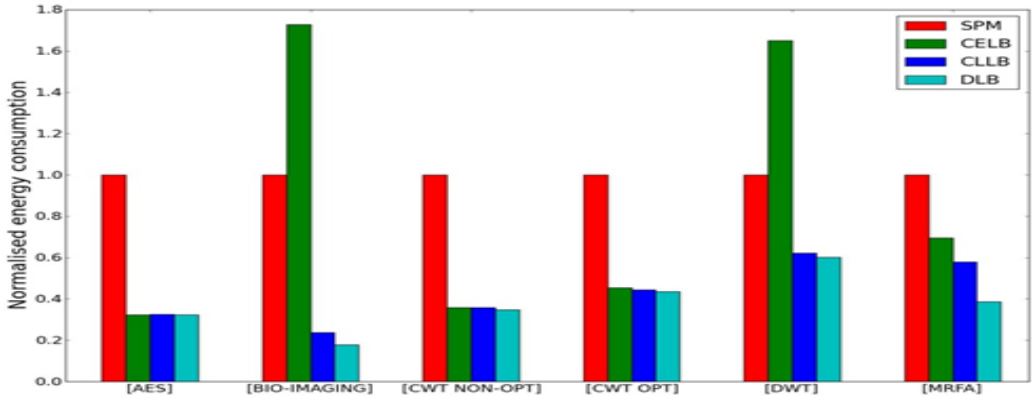
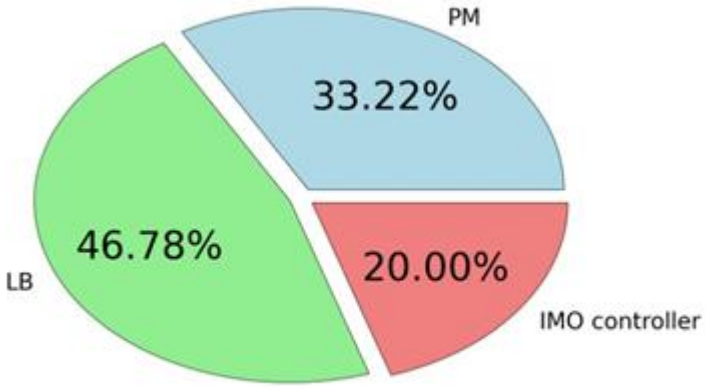


Figure 9: Normalised energy consumption in different IMOs running the selected benchmarks

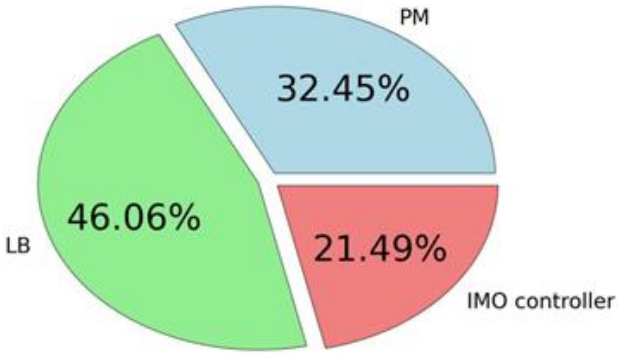
Figure 9 contains a bar plot where an examination is performed among the benchmarks that are displayed in Table 1. In this Figure, it is conceivable to perceive how the code changes and the choices of the compiler affect the vitality utilization of the IMO. The standard design is an IMO that is construct just with respect to a SPM (Scratchpad Memory). As appeared in Figure 9, no structural improvement that is presented in the IMO delivers a significant decrease of the vitality utilization of the framework. The reason of the varieties in the total estimation of the vitality diminishments depends on the percentage of the execution time that is related to loop code. If this percentage is low, the energy savings are smaller than the case where this percentage is high. On the one hand, the difference in energy consumptions between the CELB architecture and the CLLB architecture are related to the loop transformations that are applied in the application in order to parallelise its execution. An effective parallelism leads to higher difference between these two loop buffer architectures, where the CLLB architecture has less energy consumption. However, a poor
parallelism will lead to small difference between them, and even, in some cases the CLLB architecture could have more energy consumption than the CELB architecture. On the other hand, the difference in energy consumptions between the CLLB architecture and the DLB architecture is related to the behaviour of the compiler. If the compiler maps the application on the loop buffer architecture effectively, the difference between energy reductions will increase, where the DLB architecture will be the loop buffer architecture that has less energy consumption. These results show that an energy efficient embedded design has to take care not only about the architectural

configurations that are used in the embedded system, but also about the characteristics of the application that is running on it.
Figure 10 shows how for the benchmark AES the distribution of the energy consumption in the IMO changes from one representative loop buffer architecture to other. If the CELB and the CLLB architectures are compared, it is possible to see that the energy related to the loop buffer controller is increased

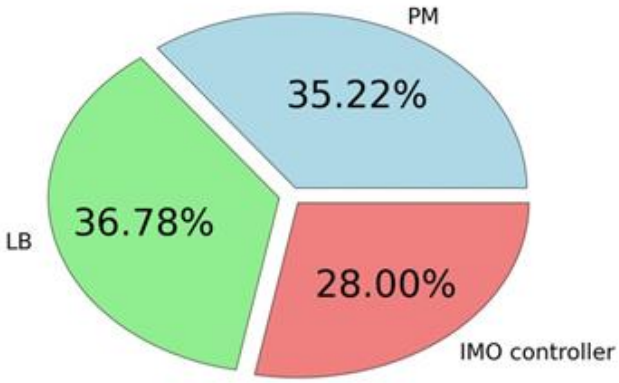
in the last loop buffer architecture, due to the fact that the controller have to control the activation or deactivation of more components. Also, it is possible to appreciate that the consumption of the loop buffer architecture is reduced. This fact is related to the reduction in the dynamic energy consumption that is caused by the use of smaller memory instances. If the DLB architecture is analysed, it is possible to appreciate that the percentage of the energy consumption that is related to the loop buffer controller of this loop buffer architecture is higher than the loop buffer controllers of the previous loop buffer architectures. This is due to the fact that the loop buffer controller of the DLB architecture is more complex than the previous loop buffer architectures. However, as it is possible to see from this Figure, the efficient management that is performed by the loop buffer controller leads to higher reductions in the loop buffer architecture and, as a consequence in the IMO. It should be noted that the absolute value of the energy consumption of the program memory is not constant in all these IMOs due to two facts. Firstly, the number of cycles that the application requires for its execution over these loop buffer architectures changes. Secondly, the size and width of the memory changes, as well as its composition when it is banked.



(a) CELB architecture



(b) CLLB architecture



(c) DLB architecture

Figure 10: Energy consumption breakdown for each one of the representative IMOs

In order to evaluate the accuracy of this high-level energy estimation and exploration tool, comparisons between IMOSIM and post-layout simulations were performed. The execution on the CELB architecture of the real-life embedded applications that are presented in Table 1 shows a mean error Of , as it can be seen from Table 2. This mean error is 3.59 compensated by the reductions in simulation time that are offered by IMOSIM, which are based in the advantage of not performing a synthesis of the system architecture every time that this is modified.

Table 2: Accuracy evaluation of IMOSIM

Benchmark [Reference]	IMOSIM simulation	Post-Layout simulation	Error [%]
AES [TSH+10] (See Section C.3)	$5{:}04\ 10^{05}$ [mJ]	$4{:}97\ 10^{05}$ [mJ]	1:34 %
BIO-IMAGING [PFH+12] (See Section C.4)	$2{:}71\ 10^{03}$ [mJ]	$2{:}89\ 10^{03}$ [mJ]	6:79 %
CWT NON-OPTIMISED [YKH+09] (See Section C.5)	$4{:}44\ 10^{03}$ [mJ]	$4{:}35\ 10^{03}$ [mJ]	2:02 %
CWT OPTIMISED [YKH+09] (See Section C.6)	$2{:}08\ 10^{03}$ [mJ]	$2{:}15\ 10^{03}$ [mJ]	3:56 %
DWT [DS98] (See Section C.7)	$2{:}07\ 10^{02}$ [mJ]	$2{:}15\ 10^{02}$ [mJ]	3:76 %
MRFA [QJ04] (See Section C.9)	$3{:}18\ 10^{03}$ [mJ]	$3{:}38\ 10^{03}$ [mJ]	6:23 %

I. Conclusion

Scientists have exhibited that an applicable bit of the aggregate vitality spending plan of an implanted guideline set processor stage is identified with the direction memory association. Past works indicated modern vitality demonstrating strategies to definitely gauge the vitality utilization of an installed framework. In any case, these techniques do not have a vitality investigation of the full design scope of the direction memory association. This Chapter proposes an abnormal state vitality estimation and investigation device, which finds the advanced configuration for the aggregate vitality utilization of the inserted framework, by investigating different configurations of the guideline memory association, for both given application and compiler. Aside from the lessening in time and effort to investigate the plan space, the proposed device permits the investigation of the effects

that are caused by code changes and compiler configurations in the aggregate vitality utilization of the guideline memory association.

References

[1]. Eugene Shih, Seonghwan Cho, Fred S. Lee, Benton H. Calhoun, and Anantha Chandrakasan. Design considerations for energy-efficient radios in wireless micro sensor networks. Journal of VLSI Signal Processing Systems , 37(1):7794, May 2004.

[2]. Kim, H., S. Kim, N. V. Helleputte, A. Artes, M. Konijnenburg, J. Huisken, C. V. Hoof, and R. F. Yazicioglu, A Configurable and Low-Power Mixed Signal SoC for Portable ECG Monitoring Applications, Journal of IEEE Transactions on Biomedical Circuits and Systems: IEEE Computer Society, 2013.

[3]. Artes, A., R. Fasthuber, J. L. Ayala, P. Raghavan, and F. Catthoor, Design Space Exploration of Loop Buer Schemes in Embedded Systems, Special Issue of Journal of Systems Architecture on Design Space Exploration of Embedded Systems: Elsevier Amsterdam, 2013.

[4]. A., R. Fasthuber, J. L. Ayala, P. Raghavan, and F. Catthoor, Design Space Exploration of Distributed Loop Buffer Architectures with Incompatible Loop-Nest Organisations in Embedded Systems, Journal of Signal Processing Systems: Springer New York, 2013.

[5]. A., J. L. Ayala, J. Huisken, and F. Catthoor, Survey of Low-Energy Techniques for Instruction Memory Organisations in Embedded Systems, Journal of Signal Processing Systems: Springer New York, 2012.

[6]. A., J. L. Ayala, and F. Catthoor, Power Impact of Loop Buffer Schemes for Biomedical Wireless Sensor Nodes, Journal of MDPI Sensors: MDPI AG, 2012. , and the following articles in international peer-reviewed conferences:

[7]. Komalan, M., M. Hartmann, J. I. Gomez, C. Tenllado, A. Artes, and F. Catthoor System Level Exploration of Resistive-RAM (ReRAM) based Hybrid Instruction Memory Organization, Memory Architecture and Organization Workshop (MeAOW), 2012.

[8]. A., J. L. Ayala, and F. Catthoor, IMOSIM: Exploration Tool for Instruction Memory Organisations based on Accurate Cycle-Level Energy Modelling, IEEE International Conference on Electronics, Circuits, and Systems (ICECS), 2012.

[9]. H., R. Firat, S. Kim, V. N. Helleputte, A. Artes, M. Konijnenburg, J. Huisken, J. Penders, and V. C. Hoof, A Configurable and Low-Power Mixed Signal SoC for Portable ECG Monitoring Applications, Symposium on VLSI Technology and Circuits, 2011.

[10]. A., J. L. Ayala, V. A. Sathanur, J. Huisken, and F. Catthoor, Run-time Self-tuning Banked Loop Buffer Architecture for Power Optimization of Dynamic Workload Applications, IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SOC), 2011.

[11]. A., Power Consumption on Loop Buffer based Instruction Memory Organizations, STW.ICT Conference on Research in Information and Communication Technology, 2010.

[12]. Artes, A., F. Duarte, M. Ashouei, J. Huisken, J. L. Ayala, D. Atienza, and F. Catthoor, Energy Efficiency using Loop Buffer based Instruction Memory Organization, International Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems (IWIA), 2009.

International Journal of Engineering Science Invention (IJESI) is UGC approved Journal with Sl. No. 3822, Journal no. 43302.

Prof.Dr.G.Manoj Someswar "Exploration Tool For Instruction Memory Organisations Based On Accurate Cycle-Level Energy Modelling" International Journal of Engineering Science Invention (IJESI), vol. 07, no. 02, 2018, pp. 39–51.