Real-Time Implementation Of Fuzzy Logic Based DVFS For Leon3 Architecture

Shaheryar Najam, Jameel Ahmed
Riphah International University, I-14, Islamabad, Pakistan
shaheryar.najam@riphah.edu.pk

***Abstract*—In the field of embedded system, there is a trade-off between power/performance optimization. Furthermore, demand for High Performance Computing (HPC) is increasing as well [1], hence many heuristics techniques were presented at various development levels such as hardware software co-design, schedulers and optimal code compilation. However, there is a critical issue of power and heat dissipation with these HPC. This paper presents a fuzzy logic based Dynamic Voltage and Frequency Scaling (DVFS) techniques to reduce the power without compromising the performance. For real time implementation, Leon3 architecture was implemented on Vertex 5 FPGA board. Results were tested using both series and parallel benchmarks and 19 % reduction in power consumption is observed.**

***Index Terms*—DVFS, HPC , FPGA, Leon3, Ubuntu.**

I. Introduction

There has been a rapid growth in demand for HPC. Due to which, designers are focusing on multi-core architectures. The prime purpose of shifting toward multi-core architecture is to exploit the Data Level Parallelism (DLP) and Instruction Level Parallelism (ILP). In this way, performance of the system can be increased to next level. But there exit a trade off between performance and power consumption. Due to these enhancements and progressions, power consumption and heat dissipation of the system has been increasing rapidly as well. High power consumption and heat dissipation has been a crucial issue for designers as it results in high cost of cooling units.

On the other hand, large amount of parallelism is often go under utilized and become liability rather than beneficial [2]. This is due the the reason that work load available in computing system is dynamic so, in those scenarios, where workload is either less available or less parallel. Hence, there is a requirement for smart processor that is capable of efficiently utilizing the resources to optimize the power consumption of the system. However, there lies various challenges in reconfiguring the multi-core architectures as discussed by J.ahmed et al. [3]

Various techniques has been implemented in order to optimize the power consumption [4]–[6]. Targeted technique in this paper is Dynamic Voltage and Frequency Scaling (DVFS). Frequency scaling works on the principal of workload, when less workload is available for processor, running the processor on maximum frequency is not optimal rather, scaling down the frequency would be a batter option for saving power without disturbing the overall performance. However, scaling down the frequency in such scenarios would not be beneficial all the time, as due to the memory latency, minimum frequency would not result in saving power rather, it would lengthen the task to such limit that over all power would be increased [7].

Scaling frequecny should be smartly operated. There is requirement of controller to efficiently select frequency. A smart fuzzy based controller [8], proposed in this paper, has ability to efficiently utilize the resources. Fuzzy Logic Controller (FLC) work on the feedback principal and takes throughput and power consumption as input and depending on these input, decides frequency of system. Next section explains you related research of DFVS algorithms.

II. Related Work

Various algorithm have been presented by numerous groups of researchers. Manish Bansal et al. presented a DVFS algorithm [9]. Operating frequency selection rely upon the utilization of CPU. Frequency of the processor is scaled up/down using Linux kernel-level governor. For battery powered embedded systems, we need best method to get less power consumption with frequency scaling. There is an issue while doing frequency scaling, performance may get effected. Basically, frequency is scaled down when less workload is available for the processor. Proposed scheme uses a feedback mechanism as algorithm keeps track of CPU utilization.

Feedback based DVFS controller for reducing the energy has been proposed by Manousakis et al. [10]. Due to the dynamic attributes of the workload, there are few points where workload is Input/Output (I/O) intensive. Such application requires more memory access as compared to computation. Therefore, running these application with high frequency might not be efficient in terms of performance to power consumption ratio. Proposed scheme using feedback based controller for Linux, tries to locate the I/O intensive workload and adjusts the processor's operational frequency accordingly. Results have shown significant reduction in power consumption and heat dissipation.

Multi-core architecture with heterogeneous cores are another approach of efficient utilization of resources. Different workload is scheduled at these cores. Hence, per core DVFS apprach would be a beneficial technique to reduce energy and heat dissipation. Hence, on-chip, per core voltage and frequency regulator targeting the portable embedded systems

S. Najam is with the Department of Electrical Engineering (FEAS), Riphah, International, University of Sciences and Technology, I-14, Islamabad, Pakistan, 47040 secondary e-mail: (shaheryar.najam@riphah.edu.pk).

TABLE I
SYSTEM CONFIGURATION FOR LEONE3 PLATFORM

CPU Frequency	Reconfigurable.
Processor Cores	4 (SMP)
I-Cache	8KB, 32bytes/line
D-Cache	4KB, 32bytes/line
Integer Unit	Target Technology
Simulation Environment	ModelSim
GRLIB Release	GPL 1.4.1-b4156

was proposed by Kim et al. [11]. Unlike the conventional off chip DVFS scheme, proposed scheme uses on-chip voltage regulator such that the voltage regulators provide benefit of both per core voltage control and nanosecond voltage switching. On-chip voltage regulator improves the effectiveness of DVFS and have significant impact on power saving. However, such design have several challenges including regulator efficiency and output voltage transient characteristics, which are significantly impacted by the system-level application of regulator. Various DVFS algorithms have been proposed and implemented having significant results in power optimization [4], [12]–[14].

III. METHODOLOGY

FLC proposed in the paper, have been designed to work on the feedback principal. It takes *throughput* and *power consumption* of the system as feedback parameters and depending on rule base, fuzzy decide the operating frequency. Targeted platform for real time implementation was *Vertex 5 FPGA* board and soft core of *Leon3* architecture was ported to FPGA board along with *Ubuntu* operating system. Implementation details are as follows.

A. Hardware Customization

GRLIB distribution of LEON3 is an open source IP freely available from Gaisler website but this IP does not support two key features i.e dynamic voltage frequency scaling and performance monitoring unit in hardware design of LEON3. Systems performance has been considered as an important metric to decide optimal configuration of reconfigurable parameters such as operating frequency. Hence, one of the important contributions of this article is to include support for both DVFS and performance monitoring unit (PMU) in hardware design of LEON3 processor. As LEON3 is a soft core processor so the hardware design can be customized to include these features. System configuration of Leon3 platform is given in Table I

1) DVFS Feature in Hardware Design: The Leon3 architecture has been extended to include frequency reconfigurable feature using DCM_ADV an IP Core provided by Xilinx which includes full access to all the features of original DCM as well as support for dynamic reconfiguration circuit to dynamically synthesize a new frequency adjusted clock. This functionality has been implemented by dynamically changing the multiply and divide attributes of DCM $CLKFX\_MUL$ and $CLKFX\_DIV$ using dynamic reconfigurable ports (DI and DADDR) introduced in DCM_ADV IP Core. The DCM induction in clock generation unit is intact and DCM_ADV primitive has been introduced, the system clock is switching between the clock generated by DCM and DCM_ADV. The clock generated by DCM is static whereas the clock generation using DCM_ADV is dynamic, moreover when generating a new frequency adjusted clock, LEON3 processor clock switches to static clock coming from DCM and when the new frequency adjusted clock is locked, processor clock switches back to clock coming from DCM_ADV. This is useful to ensure that processor clock never stops as stopping the processor clock may halt the pipeline until it awakes back. DVFS modification is shown in Fig. 1

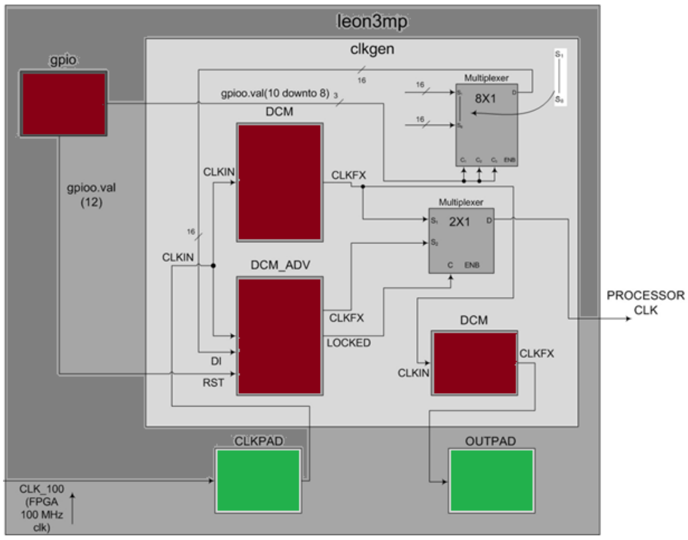

Fig. 1. Advance DCM modification of Leone3

2) Implications of DVFS on UART: Universal Asynchronous Receiver/Transmitter (UART) is one of the peripherals attached to the Advance Peripheral Bus (APB) and is mapped at 0x80000100 in the address space. The interface is provided for serial communications and this UART has a programmable 12-bit scalar to generate the desired baud-rate, the number of scalar bits can be increased with VHDL generic s-bits. system clock has a direct effect on scalar value so dynamically changing the system clock frequency without adjusting the scalar value accordingly can result in UART malfunctioning. In order to avoid this situation we have reconfigured the UART scalar register every time a new frequency adjusted clock is locked. The UART scalar register is mapped at 0x8000010C so it can be reconfigured from the application program.

B. Input Parameter

FLC works on the feedback principal and take two input parameter. Input information is crucial as rule based inference, designed to achieve the required result, highly depend on the membership value of the input variables. FLC used following two input parameter.

1) Throughput: Throughput is defines as number of task performed by a processor in a perpendicular time frame. Usually throughout is calculated to analyze the performance

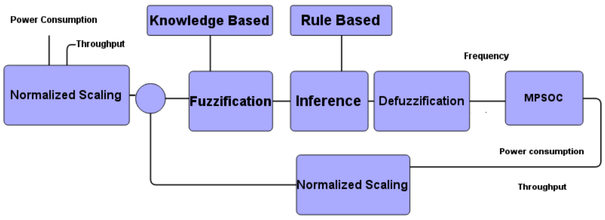

Fig. 2. Block diagram of feedback driven fuzzy logic system

of system. Throughput of a proposed architecture, for a particular benchmark, was calculated using *perf* tool of Ubuntu operating system

2) Power Consumption: As power optimization is the prime objective of FLC therefor, It is important to keep track of power consumption of the system. Power consumption of the system was analyzed and modeled using *powerstat* tool of Ubuntu operating system.

C. Real Time Implementation of Fuzzy Logic Controller

A FLC has been designed to optimized the reconfigurable parameters. FLC works on feedback system and updates the input parameters tries to optimize the power consumption and performance ratio. Block diagram of FLC is shwn in Fig. 2 . Initially, to fuzzify input and output parameter, three different fuzzy subset was defined and assigned to their membership function respectively i.e low, middle and high boundary and are denoted by μ_A, μ_B and μ_C respectively. For example, throughput of the processor varies from 0 to 1. Lower membership function μ_A is bounded between 0 and 0.35, μ_B membership means moderate throughput and it is bounded between 0.2 and 0.8, μ_C membership means high throughput and it is bounded between 0.65 and 1. Membership for other inputs and outputs are also designed in similar approach and details are given in Table II and III . Establishment of relationship between input and output parameters of system is critical, fuzzy logic rules were designed as shown in Table IV. The rules were deigned in order to balance performance and energy consumption ratio. FLC keeps track of overall cache miss rate, energy consumption and throughput for all cores and strive to optimize cache size, frequency.

To defuzzify, *centroid* method is used which calculates the center of gravity (COG) of the particular membership function area. Averaging in centroid method dilutes the control action and makes controller less sensitive to minor variations. COG was calculated using following equation.

$$COG = \frac{\sum_{x=a}^{b} \mu_A(\chi) x}{\sum_{x=a}^{b} \mu_A(\chi)} \tag{1}$$

D. DFVS

FLC using the rule base system, proposed a frequency of the system. Scaling the frequency has significant impact on power and execution time. Scaling down the frequency would reduce the power consumption but on the other hand, it will lengthen the task and increase execution time of the process. This is evident in Fig. 3.Therefore, frequency must be selected to find a balance between power consumption and throughput of the system Benchmark used for this result was *coremark*. In this regard FLC tries to find the balance and force the system to run on proposed frequency by using the *DCM_ADV* module, discussed in section III.

TABLE IV
RULES FOR FUZZY LOGIC CONTROLLER

Energy	Throughput	Frequency
L	L	H
L	M	M
L	H	-
M	L	H
M	M	M
M	H	M
H	L	M
H	M	M
H	H	L

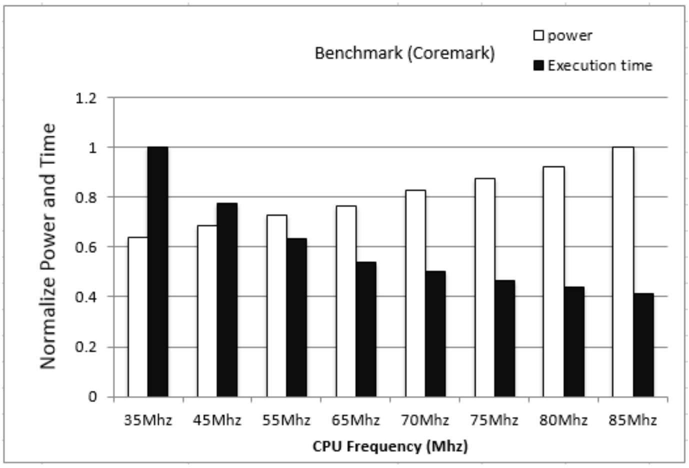

Fig. 3. Impact of frequency scaling on power and execution time

IV. RESULTS

Results for FLC, proposed in this paper, were tested using both series and parallel work. Impact of DVFS on series benchmarks were more significant as workload is distributed in series. Due to this fact, scaling frequency significantly change

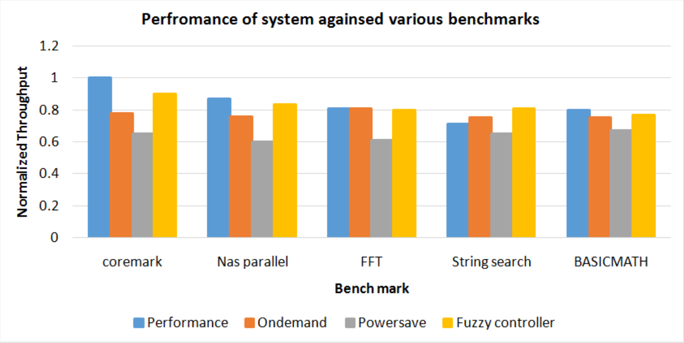

Fig. 4. Performance of system using FLC and governors of Ubuntu

TABLE II
FUZZY MEMBERSHIP FUNCTIONS FOR INPUT VARIABLES

$\mu_A = \begin{cases} 0 & if\ 0.35 \leq\ Throughput\ \leq 0 \\ \frac{0.35-x}{0.35} & if\ 0 \leq\ Throughput\ \leq 0.35 \end{cases}$ $\mu_B = \begin{cases} 0 & if\ 0.8 \leq\ Throughput\ \leq 0 \\ \frac{x-0.2}{0.3} & if\ 0.2 \leq\ Throughput\ \leq 0.5 \\ \frac{0.8-x}{0.3} & if\ 0.5 \leq\ Throughput\ \leq 0.8 \end{cases}$ $\mu_C = \begin{cases} 0 & if\ 1.0 \leq\ Throughput\ \leq 0.65 \\ \frac{x-0.65}{0.35} & if\ 0.65 \leq\ Throughput\ \leq 1.0 \end{cases}$
Normalized Throughput
$\mu_A = \begin{cases} 0 & if\ 0.35 \leq\ EnergyConsumption\ \leq 0 \\ \frac{0.35-x}{0.35} & if\ 0 \leq\ EnergyConsumption\ \leq 0.35 \end{cases}$ $\mu_B = \begin{cases} 0 & if\ 0.8 \leq EnergyConsumption\ \leq 0 \\ \frac{x-0.2}{0.3} & if\ 0.2 \leq\ EnergyConsumption\ \leq 0.5 \\ \frac{0.8-x}{0.3} & if\ 0.5 \leq\ EnergyConsumption\ \leq 0.8 \end{cases}$ $\mu_C = \begin{cases} 0 & if\ 1.0 \leq\ EnergyConsumption\ \leq 0.65 \\ \frac{x-0.65}{0.35} & if\ 0.65 \leq\ EnergyConsumption\ \leq 1.0 \end{cases}$
Normalized Energy Consumption

TABLE III
FUZZY MEMBERSHIP FUNCTIONS FOR OUTPUT VARIABLES

$\mu_A = \begin{cases} 0 & if\ Operatingfrequency\ \leq 16MHz\ or \geq 20MHz \\ 1 & if\ 16 \leq\ Operatingfrequency\ \leq 20MHz \end{cases}$ $\mu_B = \begin{cases} 0 & if\ Operatingfrequency\ \leq 20MHz\ or \geq 25MHz \\ 1 & if\ 20MHz \leq\ Operatingfrequency\ \leq 25MHz \end{cases}$ $\mu_C = \begin{cases} 0 & if\ Operatingfrequency\ \leq 25MHz\ or \geq 33MHz \\ 1 & if\ 25MHz \leq\ Operatingfrequency\ \leq 33MHz \end{cases}$
Operating frequency

the performance of the system. However, Power consumption is saved in both series and parallel benchmarks by scaling down the frequency. Scaling frequency to minimum level may save power but large amount of disturbance is expected in performance. It is important o analyze the workload and scale frequency proportionately. Power consumption of system using FLC is shown in Fig. 5. In saving power, it is important that throughput is not disturbed to greater extent. Results of throughput is shown in Fig.4 and it is evident that upto 19% of power is saved with just 3 % of compromised throughput.

It is important to note that power saving by FLC may vary across the various benchmarks. This is due to nature of benchmarks. If a high computational benchmark of series nature is executed, opportunity to save power would be less. In such cases, any attempt to save extra power will be costly in terms of execution time. Therefore, power saving in various benchmarks is different as FLC doesn't try to save only power, rather it tries to find the optimum point between power and performance.

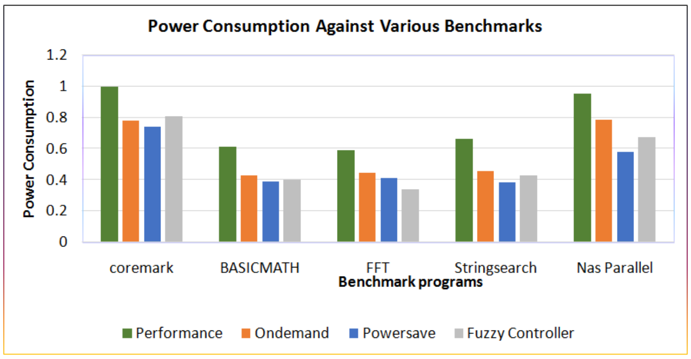

Fig. 5. Power consumption of system using FLC and governors of Ubuntu

V. CONCLUSION

Proposed scheme in the paper uses FLC to reconfigure the frequency of processor to reduce power consumption without compromising the throughput to greater extend. Results of implementation of proposed fuzzy logic based DVFS enabled controller shows that it can save power consumption upto 19 % with only 3 % of loss in throughput of system.

Acknowledgment

We would like to thank ICT Ministry of IT, Government of Pakistan as this research is part funded by National ICT RD fund under grant ICTRDF/TRD/2012/65. This research was also supported in part by Riphah International University and NUST We are also grateful to anonymous reviewers for their insightful suggestions.

References

[1] J. Ahmed, M. Y. Siyal, S. Najam, and Z. Najam, “Energy-delay product and throughput,” in *Fuzzy Logic Based Power-Efficient Real-Time Multi-Core System*. Springer, 2017, pp. 17–22.

[2] M. Y. Qadri, N. N. Qadri, and K. D. McDonald-Maier, “Fuzzy logic based energy and throughput aware design space exploration for mp-socs,” *Microprocessors and Microsystems*, vol. 40, pp. 113–123, 2016.

[3] J. Ahmed, M. Y. Siyal, S. Najam, and Z. Najam, “Challenges and issues in modern computer architectures,” in *Fuzzy Logic Based Power-Efficient Real-Time Multi-Core System*. Springer, 2017, pp. 23–29.

[4] D. You and K.-S. Chung, “Dynamic voltage and frequency scaling framework for low-power embedded gpus,” *Electronics letters*, vol. 48, no. 21, pp. 1333–1334, 2012.

[5] J. Leverich, M. Monchiero, V. Talwar, P. Ranganathan, and C. Kozyrakis, “Power management of datacenter workloads using per-core power gating,” *IEEE Computer Architecture Letters*, vol. 8, no. 2, pp. 48–51, 2009.

[6] K. Meng, R. Joseph, R. P. Dick, and L. Shang, “Multi-optimization power management for chip multiprocessors,” in *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*. ACM, 2008, pp. 177–186.

[7] W.-Y. Liang, P.-T. Lai, and C. W. Chiou, “An energy conservation dvfs algorithm for the android operating system,” *JoC*, vol. 1, no. 1, pp. 93–100, 2010.

[8] J. Ahmed, M. Y. Siyal, S. Najam, and Z. Najam, “Fuzzy logic theory,” in *Fuzzy Logic Based Power-Efficient Real-Time Multi-Core System*. Springer, 2017, pp. 47–62.

[9] M. Bansal, D. Pushpan, H. Medha, S. Jain, and S. Jain, “Cpu frequency scaling by utilization,” *Technocal Report, IIITB-TR-2012-003*, 2012.

[10] I. Manousakis, M. Marazakis, and A. Bilas, “Fdio: A feedback driven controller for minimizing energy in i/o-intensive applications.” in *HotStorage*, 2013.

[11] W. Kim, M. S. Gupta, G.-Y. Wei, and D. Brooks, “System level analysis of fast, per-core dvfs using on-chip switching regulators,” in *High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on*. IEEE, 2008, pp. 123–134.

[12] B. Dietrich, S. Nunna, D. Goswami, S. Chakraborty, and M. Gries, “Lms-based low-complexity game workload prediction for dvfs.” in *ICCD*, 2010, pp. 417–424.

[13] S.-Y. Peng, T.-C. Huang, Y.-H. Lee, C.-C. Chiu, K.-H. Chen, Y.-H. Lin, C.-C. Lee, T.-Y. Tsai, C.-C. Huang, L.-D. Chen *et al.*, “Instruction-cycle-based dynamic voltage scaling power management for low-power digital signal processor with 53% power savings,” *Solid-State Circuits, IEEE Journal of*, vol. 48, no. 11, pp. 2649–2661, 2013.

[14] X. Lin, Y. Wang, Q. Xie, and M. Pedram, “Task scheduling with dynamic voltage and frequency scaling for energy minimization in the mobile cloud computing environment,” *Services Computing, IEEE Transactions on*, vol. 8, no. 2, pp. 175–186, 2015.