

Processor Performance Enhancement Using Self-Adaptive Clock Frequency

Sasikala D
P.G Scholar
Dept.of CSE,
Sri Venkateswara College of
Engineering,
Sriperumbudur, India

M.Ravichandran M.E.
Senior Lecturer
Dept.of CSE,
Sri Venkateswara College of
Engineering,
Sriperumbudur, India

Dr.C.S.Ravichandran
Principal
Dept.of EEE,
SSK College of Engineering &
Technology,
Coimbatore, India

ABSTRACT

Traditional design methodologies of a digital system assumes the worst case operating conditions to tolerate the physical and environmental variations, which ensure the system operates correctly and conservatively. The clock frequency of the processor is generally set to operate below the maximum permissible operating frequency which achieves less than the maximum performance gains. However, in-fact higher performance gain can be achieved by dynamic *overclocking* mechanism, which tunes the clock rate beyond the worst case assumptions. Dynamic overclocking mechanism may eventually lead to rise in temperature if it is adopted for a longer duration of time hence this problem is circumvented by using *throttling* mechanism at appropriate instances. Further, the throttling technique can be used to conserve *energy* when the system is with low work load. The aim of the work is to exploit the techniques of overclocking and throttling to enhance the performance, also to achieve optimal utilization of processor resources and at the same time maintaining the system reliability. The goal is hence achieved by employing an *Adaptive Neuro Fuzzy Inference System* to self-tune its frequency based on work load characteristics and also adapting itself to environmental variations.

General Terms

Adaptive Neuro Fuzzy Inference System.

Keywords

Overclocking, Throttling, Energy.

1. INTRODUCTION AND MOTIVATION

Innovations and improvements have long been made in computer and system architectures to essentially increase the performance. Improvements in semiconductor technology make it possible to incorporate millions of transistors on a very small die and to clock them at very high speeds. Achieving high performance is a primary goal of processor microarchitecture design. In spite of the complexities of new manufacturing technologies and increasingly complicated architectures, designers have been able to steadily increase the performance of high-end microprocessors. This improvement is achieved through optimizations at the architecture level and at the circuit level. The digital system which is a conglomerate group of components is expected to work correctly and stable at all conditions, is contributing to the necessity of extra conservatism in all layers of the system design.

The performance of processors has traditionally been characterized by their operating frequency. The operating frequency, at which a processor or any digital system is marketed, is the frequency at which it is tested to operate reliably under adverse operating conditions. In order to satisfy timing criteria, error free operation, designers are forced to assume worst case conditions while deciding the clock frequency. The worst case delay will be observed only if the longest path is exercised by the inputs. Such worst case timing delays occur rarely. As the clock period is fixed at much higher value than what is typically required, significant performance improvements can be achieved through overclocking.

Overclocking is a technique where the processor's performance is increased, by ramping the processor's frequency beyond the design specification. Overclocking seeks to exploit the performance gap left by worst case design parameters. Over the last decade, overclocking as a means to improve processor performance is gaining popularity. Chipset manufactures are introducing technologies that support overclocking. AMD's Overdrive, Advance Clock Calibration technologies and Intel's Turbo Boost Technology (Nehalem processor) which enables the higher performance through the availability of the increased core frequency are one among them.

Although overclocking mechanisms facilitate in improving performance, they adversely impact on-chip temperatures, leading to hotspots if exercised for longer periods. High temperature might cause the unexpected functional errors or permanent damage of the system, especially in high performance microprocessors. Thus, it is important to control the temperature as well as the energy consumption during overclocking. In fact, hotspots may move over time, depending on which on-chip functional blocks are most heavily used. Uneven activity from one functional block to another, results in localized hotspots that may move over time. So, we need to know the actual temperature of the functional block that needs to be controlled if it hits the threshold temperature during overclocking. The proposed system employs throttling as a mechanism to control the temperature.

Dynamic Frequency Scaling or CPU Throttling is a technique where a processor is run at less than maximum frequency. It is a widely used energy conserving technique. Reducing energy consumption has been one of the most interesting research topics in the computer architecture field. As technology trends leads to packing transistors ever more tightly, power densities are increasing rapidly. Demand for low power consumption in

battery-powered computer systems has risen sharply. This is because extending the service lifetime of these systems by reducing their power requirements is a key customer/user requirement. More recently, low power design has become a critical design consideration even in high-end computer systems, due to expensive cooling and packaging costs and lower reliability often associated with high levels of on-chip power dissipation. Dynamic Voltage and Frequency Scaling (DVFS) technique is one of the methods followed for achieving low power consumption while meeting the performance requirements. Intel's "Speed step" and AMD's "Cool'n'Quiet", "PowerNow" technologies are one among them.

The motivation behind this work is to enhance the performance by exploiting the performance gap left over by the worst case design parameters, and also to optimally utilize the processor resources. The proposed work achieves this by employing a novel software mechanism. It integrates the technique of overclocking and throttling along with the Adaptive Neuro Fuzzy Inference System (ANFIS) for tuning the processor frequency based on the work load characteristics. The fuzzy rule based system or Fuzzy Inference System (FIS) provides the knowledge to calculate the required frequency for an application and an ANFIS learns from the FIS knowledge base and adapts the frequency accordingly.

The rest of this paper is organized as follows. Section 2 discusses the related work, Section 3 explains the proposed work, experimental methodology, tools used, experimental results and Section 4 concludes the paper and also discusses the future direction of the work.

2. RELATED WORKS

The detailed literature survey gives an insight into the large number of related works and several techniques that have been proposed by the researchers with the goal of improving the performance and also to conserve energy. As Architectural advancements bring several improvements in system performance and new challenge to the designers, the research in this area is vast and it is impossible to provide an exhaustive overview of the contributions in this field, a very few works related to ours has been discussed here. Related works which discusses different techniques to addresses this challenge are either at the hardware or software level. Techniques such as Dynamic Thermal Management (DTM) schemes monitor system workload and adapt the system's behavior to save energy. These techniques are dynamic, run-time schemes operating at different levels of a computer system. It includes Dynamic Voltage and Frequency Scaling (DVFS) schemes that adjust the supply voltage and operating frequency of a processor to save power when it is idle.

The field of temperature aware design has recently emerged to maximize system performance under lifetime constraints. Consider the system lifetime as a resource that is consumed over time as a function of temperature. As the system is designed to operate for the worst case conditions, if it is forced to operate beyond its conservative limit it adversely affects the on-chip temperature leading to hotspots. The above said problem has been analyzed in paper [16], discusses the impact of the on-chip

temperature and the life-time of the chip during Overclocking and how much percentage the system can be reliably overclocked within the availability of increased core frequency as possible within the thermal limits.

The systems which operate at higher speed produce more heat than with the lower speed. So during overclocking temperature has become a major concern. Higher temperatures not only degrade the system performance, but also create hotspots and system failure. For accurate temperature measurement, on-chip sensors were adopted in modern processors. In fact, hotspots may move overtime, depending on which on-chip structures are most heavily used. Sung Woo Chung and Kevin Skadron [12] proposed a technique which uses on-chip counters instead of on-chip sensors to collect the activity data and by regression analysis finds the relation between activity data and temperature. As Present day microprocessors come with performance counter for debugging and performance characterization, these software counters help to calculate temperature of the on-chip functional units. Counter-based sensing provides localized temperature sensing with low hardware and execution-time costs.

Among the many works, Augustus K. Uht [1] introduces the TEATIME technique which adapts the clock frequency dynamically to enhance the performance and adapt to the system's operating conditions. It ignores the input dependence of the observed delay thus stabilizing on a frequency that is too conservative. Arif Merchant [2] examined the idea of using a variable speed clock to improve the performance by taking advantage of naturally occurring idle periods in the workload. The studies show that the optimal bang-bang control policy for the stochastic model of the Variable Speed Processor (VSP) is significantly better than the baseline Single Speed Processor (SSP). Many schemes have been proposed to trade off, Performance for Power/Energy reductions, extending lifetime of the system and so on [8, 9, 10, and 11].

A wealth of contributions by the architectural researchers with their proposed techniques and solutions in this area helps us to frame our proposed system with clear dimensionality in choosing the parameters and modeling the system. But none of the related works has encompasses all the parameters which we have taken for analysis.

3. PROPOSED WORK

The key idea behind the proposed system is to dynamically scale the frequency level of the processor based on its work load characteristics to enhance the performance and to optimally utilize the resources. The need for a particular resources such as caches, issue queue, functional unit etc., can vary significantly from an application to application and even within the different phases of a given application. The CPU structure utilization characteristics show that there is a correlation exists among these parameters. So while deciding the parameters for the proposed work we have included Cache Miss rate, Queue Occupancy rate, Functional Unit Occupancy rate along with the Temperature and Frequency.

Parameters taken for Analysis:

Primary parameters:

- Temperature
- Queue Occupancy rate.
- Functional Unit Occupancy rate.
- Cache Miss rate.

Secondary parameter:

- Frequency.

The proposed architecture for the system is depicted in Figure 1.

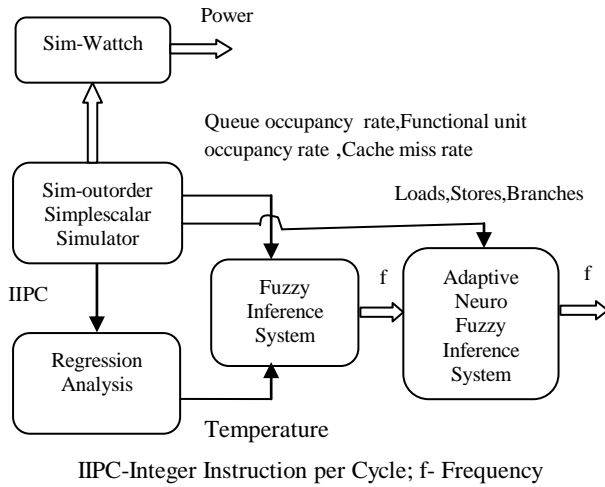


Figure 1: Proposed System Architecture

The correlation of the queue occupancy rate, cache miss rate, functional unit occupancy rate, metrics are indicative of the rate at which the instructions are flowing through the processing core. If the occupancy rate of these system parameter increases, instructions/data are not flowing through the processing units fast enough. The correlation between the primary parameters aids to determine the appropriate clock frequency for the system.

3.1 Experimental tools

Before delving into experimental procedure, we describe our simulation technique, benchmarks, and the different types of tools we studied, and used to model the system.

3.1.1 Simulation Tool: Simple-scalar & Sim-wattch

The microarchitectural model used in the proposed system is an extended version of the Simple-scalar [7] tool set, version 2.0. The base processor is PISA superscalar architecture, which is an out-of order 64-bit processor derived from Simple-scalar tool set. It is a cycle accurate simulator. We have made minor extensions to sim-outorder to account for Functional unit occupancy rate. The baseline configuration used for the simulation is given in Table 1.

Table 1. Simulated Processor Baseline Configuration

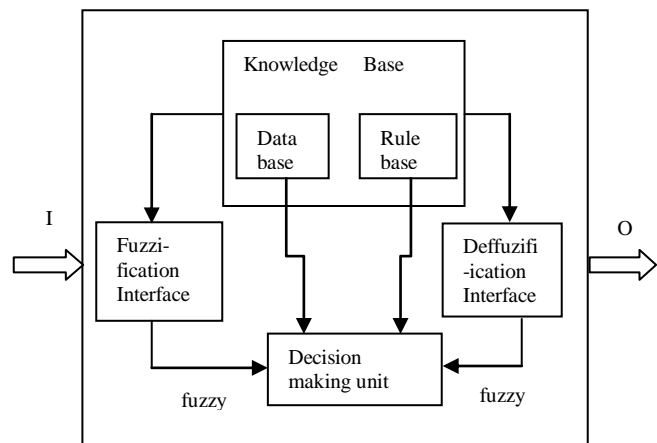
Parameter	Value	Comments
Processor		
Fetch width	4 instructions	Out-of-order
Decode width	4 instructions	Out-of-order
Issue width	4 instructions	Out-of-order

Commit width	4 instructions	In-order
Functional units	4 Integer ALU 1 Integer Multiplier/Divider 4 Floating point ALU 4 Floating point Multiplier/Divider	
RUU size	16	Instruction window
LSQ size	8	Enforces load store ordering

For power analysis we used sim-wattch [5] which is an extended version of Simple-scalar simulator. Wattch augments the Simple-scalar [7] cycle-accurate simulator (sim-outorder) with cycle-by-cycle tracking of power dissipation by estimating unit capacitances and activity factors. We evaluate the programs using benchmarks from the SPEC int95 suite. We have chosen a random subset of eight integer benchmarks: go, mgrid, jpeg, applu, apsi, li95, wave5, and vortex. These were chosen from the original 18 integer benchmarks to reduce the overall simulation times.

3.1.2 Fuzzy Inference System

Fuzzy Inference System or fuzzy rule based system employs fuzzy *if-then* rules to model the qualitative aspects of human knowledge and reasoning processes without employing precise quantitative analyses. Basically a fuzzy inference system is composed of five functional blocks as shown in Figure 2.



I – Input; O - Output

Figure 2. Fuzzy Inference System

- Rule base containing a number of fuzzy if-then rules.

- **Database** which defines the membership functions of the fuzzy sets used in the fuzzy rules.
- **Decision-making unit** which performs the inference operations on the rules.
- **Fuzzification interface** which transforms the crisp inputs into degrees of match with linguistic values.
- **Defuzzification interface** which transform the fuzzy results of the inference into a crisp output.

Usually, the rule base and the database are jointly referred to as the **knowledge base**. Steps for **fuzzy reasoning** performed by fuzzy inference systems are:

- Compare the input variables with the membership functions on the premise part to obtain the membership values of each linguistic label. (**Fuzzification**).
- Combine the membership values on the premise part to get **firing strength (weight)** of each rule.
- Generate the qualified consequent (either fuzzy or crisp) of each rule depending on the firing strength.
- Aggregate the qualified consequents to produce a crisp output. (**Defuzzification**).

The proposed system fuzzifies the crisp input parametric values of the queue occupancy rate, functional unit occupancy rate, cache miss rate, temperature, and according to the fuzzy rules which is presented in Table 2; it determines the necessary frequency to run the application from the FIS and presents the defuzzified output. We used Mamdani model FIS for our work.

3.1.3 Adaptive Neuro Fuzzy Inference System (ANFIS)

Inspired by the idea of basing the fuzzy logic inference procedure on a feed-forward network structure, Jang proposed a fuzzy neural model - the *Adaptive Neural Fuzzy Inference System* or semantically equivalently, *Adaptive Network-based Fuzzy Inference System* or semantically equivalently, *Adaptive Network-based Fuzzy Inference System* (ANFIS), whose architecture is shown in Figure 3. It is a hybrid neuro-fuzzy technique that brings learning capabilities of neural networks to Fuzzy Inference Systems. The learning algorithm tunes the membership functions of a *Sugeno-type* Fuzzy Inference System using the training input-output data.

In the proposed system the measured values of load, store and branches from the simulator is then mapped onto the frequency represented by the FIS, and the set of values {load, store, branches, frequency} are given as training input – output data to the ANFIS model for learning. It adapts the frequency for an application based on these training values.

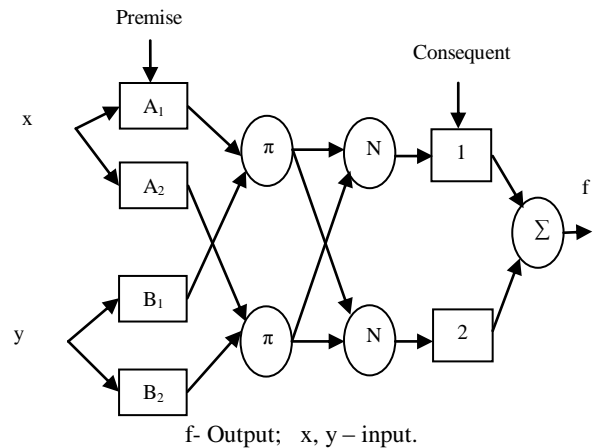


Figure 3. Adaptive Neuro Fuzzy Inference System Structure

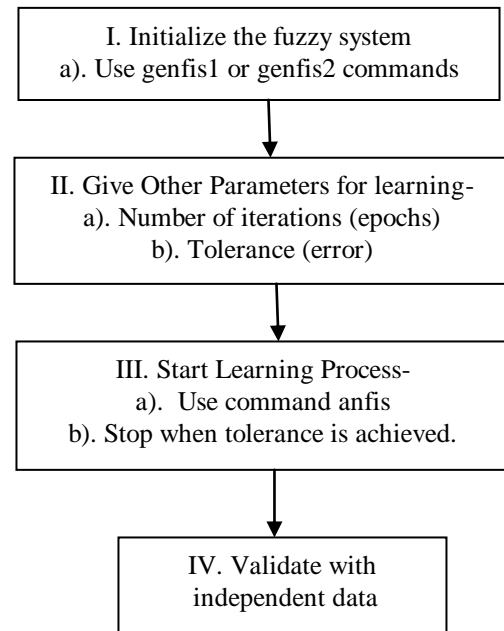


Figure 4. ANFIS Procedure

3.2 Experimental methodology

The first step of the process is profiling from the SimpleScalar simulator. For a sampling count of every 10,000 instructions the Cache miss rate, Queue occupancy rate and Functional unit occupancy rate, number of loads, stores and branches are measured. The simulation has been performed on a set of benchmarks from the SPEC95 (INT) benchmark suite. By using the regression analysis formula $Y = 14.92 * X + 50.39$ from [12], the value of the temperature is obtained where X is the Integer Instruction per Cycle (IIPC), which is obtained from the SimpleScalar simulator. The parameterized values are given as input for designing the membership function of the fuzzy inference system. The measured values of the primary variables are fed to the fuzzy inference system as input to trigger the

necessary action i.e. to determine the clock frequency based on the fuzzy rule. Fuzzy rules framed for the proposed work are governed by the number of Input membership function for each variable. Temperature range has been categorized as High, Low, and Threshold (3 levels). The membership function is depicted in Figure 6. The range for Cache miss rate, Functional unit occupancy rate, Queue occupancy rate has been defined as Low, High. Each of these parameters has 2 levels. As shown in Figure 5, the input membership functions are designed in such a way that both “Low” and “High” are triangular shaped with Low covering any value between 0 and 0.9 and High covering any value between 0.1 and 1. Hence by nCr total rules are 24. The levels for scaling the frequency (i.e. output) both throttle and overlocking have been divided into 3, Level1 (L1), Level2 (L2), and Level3 (L3). Hence, inclusive of rated frequency 7 levels of frequency scaling are possible. The output membership functions are depicted in Figure 7. Unless otherwise stated rated frequency is manufacture’s specified frequency.

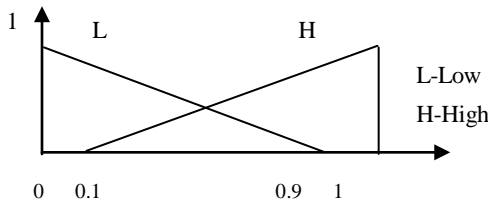


Figure 5. Input Membership Function(MF) for Queue occupancy rate, Functional unit occupancy rate, Cache miss rate

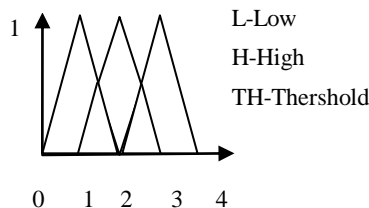


Figure 6. Input MF's for Temperature

The Input MF divides the range [0-1] into equal 3 segments.

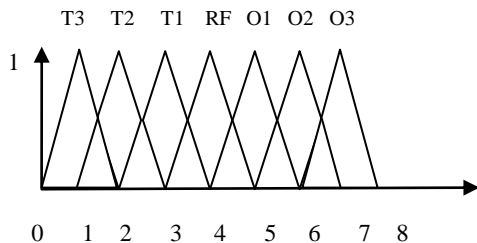


Figure 7. Output MF's

The Output MF divides the range [0-1] into equal 7 segments.

- T1-Level1 Throttle; T2-Level2 Throttle; T3-Level3 Throttle;
- RF- Rated Frequency; O1-Level1 Overclock;
- O2-Level2 Overclock; O3-Level3 Overclock

Let us explain the theoretical aspects behind the fuzzy rules that have been modeled for our system. The parameters represent the linguistic variables for the FIS. Let’s say with an example, if the queue occupancy is high, cache miss rate is low, functional unit occupancy rate is low and the temperature is low, Overclocking aids in improving the overall throughput. However, if throttling is initiated at this instant it might lead to memory overflow. Temperature is a key factor during overlocking. When the processor temperature is high if throttling is initiated, it will bring down the temperature and provides the safe operation. Throttling clamps the on-chip temperature by maintaining the temperature to a predefined value. Throttling techniques may be used to reduce the energy consumption of an executed task while ensuring that the task meets its deadline. For low-cost cooling solutions, however, reducing total power is more important. It also turns out that thermal optimization necessitates reductions in frequency that reduces power enough so that power-delivery limits are also met. Therefore, balancing the overlocking technique within thermal limits helps us to achieve performance gains as well as energy consumption. Safe zone acceleration is achieved by integrating the technique of overlocking and throttling. Based on this conclusive knowledge, the fuzzy rules (*if-then rules*) are framed and it has been presented in the form of Table (Table 2). The metric values presented in the Table 2 should be read as H- High, L-Low, TH-Throttle, OC-Overclock, and RF-Rated Frequency. The tabulated values should be read from left to right, the parameters (**Temperature, Q.O.R, F.O.R, C.M.R**) form *if part (premise)* of the rule and the last parameter (**Frequency**) forms the *then part (consequent)* of the rule for FIS. The first column of Table 2 represents rule number of FIS and the values of the subsequent rows take the numerical order i.e., rule number 1, 2,.. 24. Lets explain in detail how to interpret the values in the table as *if-then* fuzzy rule. Rule 1: If temperature is High and Queue occupancy rate is Low and Functional unit occupancy rate is Low and Cache miss rate is Low Then Frequency is throttled to Level3 of rated frequency. As temperature hits at a higher value to provide a safer operation the frequency has been brought down. Rule 18: If Temperature is Low and Queue occupancy rate is Low and Functional unit occupancy rate is Low and Cache miss rate is High Then Frequency is overlocked to Level2 of rated frequency. Aggressive overlocking is inhibited by continuously monitoring the system parameters for a predefined optimal value. However, if it reaches the threshold value, the clock speed is reduced to next lower level by applying throttling technique. The leverage has been performed in accordance to the rules. All the other rules in the Table 2 can be interpreted similar to the explanation given for Rule 1 and Rule 18. The real values taken for the experiment has been provided in squared brackets and the units in circled brackets in the explanation given for parameters below the Table 2.

For a sampling count of every 10,000 instructions the number of loads, stores and branches executed during the simulated period is mapped onto the frequency value produced by the FIS. The data set representing load, store, branch as input and frequency as output is given to the Adaptive Neuro Fuzzy Inference System (ANFIS) for training and it has been validated by the checking data set. The proposed system adapts the frequency for an application which had been previously trained by the data set.

The integrated action of overclocking and throttling by the adaptive neuro fuzzy control mechanism provides the system reliable.

Table 2. Fuzzy Rules for the Proposed System

Rule	Temperature	Q.O.R	F.O.R	C.M.R	Frequency
1	H	L	L	L	L3-TH
2	H	L	L	H	L2-TH
3	H	H	L	L	L2-TH
4	H	H	L	H	L1-TH
5	H	L	H	L	L3-TH
6	H	L	H	H	L2-TH
7	H	H	H	L	L2-TH
8	H	H	H	H	L1-TH
9	T	L	L	L	L3-TH
10	T	L	L	H	RF
11	T	H	L	L	L3-TH
12	T	H	L	H	RF
13	T	L	H	L	L2-TH
14	T	L	H	H	L1-TH
15	T	H	H	L	L1-TH
16	T	H	H	H	L1-TH
17	L	L	L	L	L2-TH
18	L	L	L	H	L2-OC
19	L	H	L	L	L1-OC
20	L	H	L	H	L3-OC
21	L	L	H	L	RF
22	L	L	H	H	L2-OC
23	L	H	H	L	L1-OC
24	L	H	H	H	L1-TH

Temperature (Celsius): H-High [61-80], T-Threshold [44-75], L-Low [39-59].

Q.O.R: Queue Occupancy Rate (cycles): L-Low [0-65], H-High [10-75].

F.O.R: Functional Unit Occupancy Rate (cycles): L-Low [0-180], H-High [20-200].

C.M.R: Cache Miss Rate (cycles): L-Low [0-65], H-High [10-75].

TH-Throttle; OC-Overclock; RF-Rated Frequency.

Watch:

The power model watch which is a modified extended version of SimpleScalar simulator is used in the proposed system to calculate the average power consumption of an application.

Results:

The following metrics are used to evaluate and understand the results.

- **Average Power:** Power consumption averaged on a per-cycle basis.
- **Performance:** We use the common metric of CPU Execution time.

We have presented the results of an initial study of the effects of overclocking and throttling technique. The results obtained from the benchmark during the initial phase of the experiment had always been resulted at above the optimal temperature, hence the processor performed throttling and the performance metric of CPU Execution time for the benchmarks between the existing system and the proposed system is presented in Figure 8. The results we obtained at this juncture are very promising, opening up many different directions. In addition, we analyzed the power efficiency of these systems and are depicted in Figure 9 which shows that when the processor is provided with the required frequency level so as to provide “just-enough” speed to process the system workload while meeting the thermal limits and thereby reduces the power consumption. Figure 10 shows that on an average the dynamic frequency scaling of the proposed system save 9%-10% of power consumption, instead of running the processor with the static frequency.

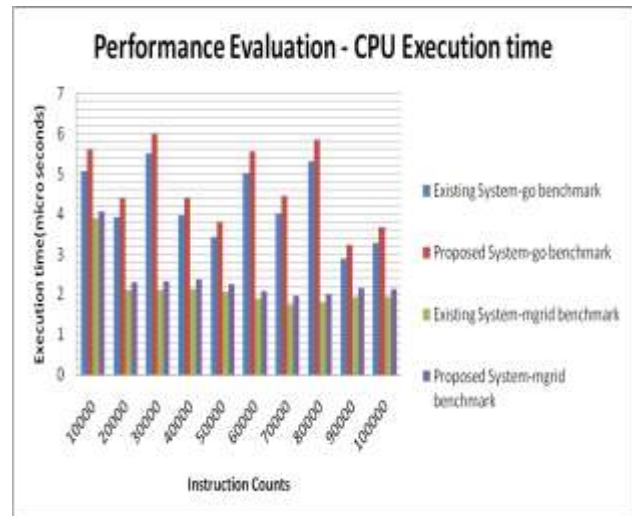


Figure 8. Performance Evaluation based on CPU Execution time between the Existing System and the Proposed System for SPEC95 Benchmark

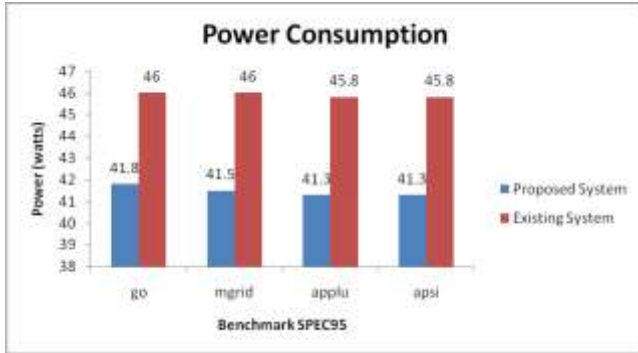


Figure 9. Average Power consumption between the Existing and the Proposed System for SPEC95 Benchmarks.

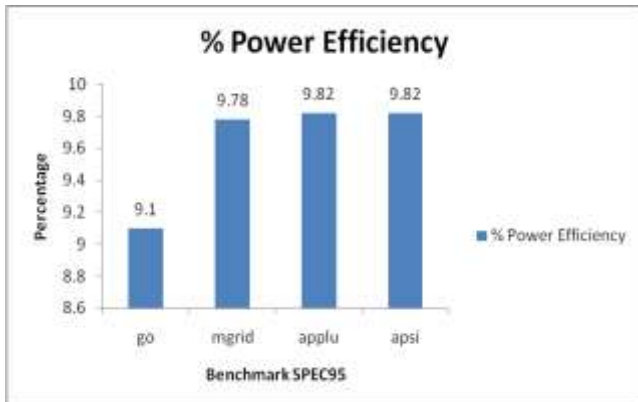


Figure 10. Percentage reduction in overall power consumption by the Proposed System compared to the Existing System.

4. CONCLUSION AND FUTURE WORK

The main motto of the project is to enhance the processor performance's gain by dynamically varying the clock frequency based on the work load and to adapt to environmental conditions. The need for robust power-performance modeling and optimization at all system levels will continue to grow with work load and performance requirements for both low-end and high-end systems. The proposed system achieves this by self-tuning the clock frequency, using the technique of overclocking and throttling along with the adaptive neuro fuzzy inference system. To avoid the hotspots or rise in temperature during overclocking a software based performance counter is used to calculate the temperature. These soft sensors augment the hardware sensors to provide localized temperature sensing with low hardware and execution-time costs. The benchmark results show that the optimal frequency for an application reduces the power consumption. Interesting directions for the future work include performance based counter techniques which are well suited for temperature-aware job scheduling. The future direction of the project is to extend the idea of overclocking and throttling on to multi-core processors. The proposed technique can also be extended on to various kinds of application processors like mobile processors, desktop, server, network processors. The

idea would be to find which processor benefits more from the integrated overclocking and throttling technique.

5. REFERENCES

- [1] A.K. Uht, "Uniprocessor Performance Enhancement through Adaptive clock Frequency control", IEEE Transactions on computers, vol.54, no.2, February 2005.
- [2] A. Merchant, B.Melamed, E.Schenfeld and B.Sengupta, "Analysis of a Control Mechanism for a Variable Speed Processor", IEEE Transactions on Computers, vol.45, no.7, pp.968-976, July 1996.
- [3] B.Colwell, "The Zen of Overclocking", IEEE Computer, vol.37, no.3, pp.9-12, 2004.
- [4] D.A.Patterson and J.L.Hennessy, "Computer Architecture: A Quantitative Approach", First Edition, Morgan Kaufman Publishers, 1990.
- [5] D.Brooks, V.Tiwari, and M.Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations", in Proceedings of the 27th Annual International symposium on Computer Architecture, pp.83-94, 2000, <http://www.eecs.harvard.edu/~dbrooks/wattch-form.html>.
- [6] David.A.Patterson and John L.Hennessy, "Computer Organization and Design", Third Edition, Morgan Kaufman Publishers, 2005.
- [7] Doug Burger and Todd M.Austin, "The simplescalar tool set version 2.0", Technical Report 1342, Computer Sciences Department, University of Wisconsin, June 1997, <http://www.simplescalar.com/docs.html>.
- [8] Greg Semeraro, David H. Albonesi, Steven G. Dropsho, Grigorios Magklis, Sandhya Dwarkadas and Michael L. Scott, "Dynamic Frequency and Voltage Control for a Multiple Clock Domain Microarchitecture", ACM/IEEE International Symposium on Microarchitecture, pp.356-367, Nov 2002.
- [9] Intel Turbo Boost Technology in Intel "Core" Microarchitecture (Nehalem) Based Processor, White Paper, November 2008.
- [10] Michael Huang, Jose Renau and Josep Torrellas, "Profile-Based Energy Reduction for High-Performance Processors", 4th ACM workshop on Feedback Directed and Dynamic Optimization (FDDO-4), Dec 2001.
- [11] Q. Wu, P. Juang, M. Martonosi, and D. W.Clark, "Voltage and Frequency Control with Adaptive Reaction Time in Multiple-Clock-Domain Processors" in *Proceedings of the 11th Int'l Symposium on High-Performance Computer Architecture*, pp 178-189, February 2005.
- [12] Sung Woo Chung and Kevin Skadro, "Using On-chip Event Counters for High-Resolution, Real-Time Temperature Measurement", International Symposium on Parallel and Distributed Processing and Applications (ISPA), Springer-Verlag LNCS, pp 63-74, Dec 2006.

- [13] T.D. Burd, T.A. Pering, A.J. Stratakos, and R.W. Brodersen, "A Dynamic Voltage Scaled Microprocessor System," *IEEE J. Solid State Circuits*, vol. 35, no. 11, pp. 1571-1580, Nov. 2000.
- [14] Vikas Agarwal, M.S. Hrishikesh, Stephen W. Kecklar and Doug Burger, "Clock Rate versus IPC: The End of the Road for Conventional Microarchitectures", *Proceedings of 27th annual International Symposium on Computer Architecture*, pp 248-259, June 2000.
- [15] Viswanathan Subramanian, Mikel Bezdek, Naga D. Avirneni, Arun Somani, "Superscalar Processor Performance Enhancement through Reliable Dynamic Clock Frequency Tuning", pp.196-205, in *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07)*, June 2007.
- [16] V.Subramanian, P. K.Ramesh, and A.K. Somani, "Managing the Impact of On-Chip Temperature on the Lifetime Reliability of Reliably Overclocked Systems" pp. 156-161 in *Second International Conference on Dependability*, June 2009.