International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 4 (2015) pp. 8941-8953

http://www.ripublication.com

High Performance Computing on Heterogeneous/ Multiprocessors System Energy-Aware Design

K.Suresh [a], Elizabeth Isaac[a], Dr.M.Rajasekhara Babu
[a]*Research Scholars, School of Computing Science and Engineering,* VIT University, Vellore, TN, India
[b]*Faculty of Engineering, School of Computing Science and Engineering,* VIT University, Vellore, TN, India
Corresponding author Email id: suresh.k2012@vit.ac.in

Abstract

Software enabled energy proficiency need redevelop to keep up the vitality effective points of interest of different centers or adding centers it prompts fittings advancement by including centers included significant adaptability and enhancing the execution Most discriminating and real in becoming innovation for High Performance of interest of force & vitality and a critical issue in fueling innovations. Vitality enhancement is an empowering force Management. Decreasing the Energy Consumption of framework not digressing the general execution of the framework. The compiler improvement will help to decrease power diminishment at programming level. Power administration programming level need and better streamlined strategy requirement for measuring the challenges heterogeneous processing frameworks at general we get beneficial of examine streamlining criteria to minimization of general vitality utilization.

Most basic and significant in becoming innovation for High Performance of interest of force & vitality and a dire issue in controlling advances. Vitality advancement is an empowering force Management. The Consumption of Energy Should be ascertainable not just to Gate Level or Register Transfer (RT) Level additionally to the System Level. Lessening the Energy Consumption of framework not digressing the general execution of the framework. The compiler streamlining will help to diminish power diminishment at programming level. Power administration programming level technique is the code enhancement by measuring the challenges at where can get beneficial of research improvement criteria to minimization of general vitality utilization. The Energy utilization and run time registered for different compiler methods on Xscale Architecture utilizing XEEMU instrument. The advanced code chose and code is tuned progressively by shifting voltage-recurrence. The improved codes are tuned rapidly.

Keywords: Compiler Optimization, Performance Evaluation, Voltage-Frequency Scaling, Xscale Architecture.

Introduction

In recent years more development of technology in computer system consisting of multiprocessing capabilities and heterogeneous processors but more powerful processor and different combination processors energy is valuable because all aspects of our system are related to energy consumption. Energy has become an important aspect of life as the factors that generate power are on the edge of extinction. So it has become very important for us to conserve energy for future in any form like computing systems, Which can be either by battery driven or driven by ac power supply by using effective heterogeneous system the consumption of energy can be reduced. This can be applicable in compiling programmes on system and by using compatible machine codes. Power aware compilation is technique by which we make every developer or user to know the amount of energy used by their codes, if it is reasonable our system reduces the consumption of energy.

In present day world each joule of vitality is profitable on the grounds that all parts of our framework are identified with vitality utilization. Vitality has turned into a critical part of life as the variables that produce force are on the edge of eradication. So it has ended up imperative for us to monitor vitality for future in any structure like figuring frameworks, which can be either by battery determined or determined by air conditioning power supply. By utilizing powerful working framework the utilization of vitality can be diminished. This can be pertinent in ordering projects on framework and by utilizing perfect machine codes. Power mindful assemblage is procedure by which we make each engineer or client to know the measure of vitality utilized by their codes. In the event that it is sensible our framework decreases the utilization of vitality.

As indicated in figure1 the investigation strategy incorporates three principle steps the triangle venture of this procedure comprises in looking the working point that fulfils the most extreme number of framework prerequisites of the energy aware computing can be classified .

Performance is always plays major role in Computer Science Every Joule is precious in today's world every aspect of system is bound by energy consumption. Energy is an essential asset because the factors that generate it are mainly depleting resources. Hence it becomes an implicit requirement to conserve energy, be it in any form i.e. Computing systems, which may be either battery driven or driven by AC power supply. Power Consumption can be reduced by having efficient operating systems that consume lesser power. The same can be applied while compiling programs on systems where we can produce energy efficient machine codes. The propose a technique called power aware compilation .Using this technique, each and every developer or user could know the amount of energy consumed by their code further, if feasible our system optimizes the energy consumption.

The increasing importance of Energy consumption and power reduction are the major problems for computer systems. From computer to smart phones, in order to

run these devices all we need is power. LPD is important will be system design consideration because system with cost based and power is concerns.

We are trying to reduce the consumption of power on Chip-Level [3] ,Gate-Level[4] ,Operating System Level[5].

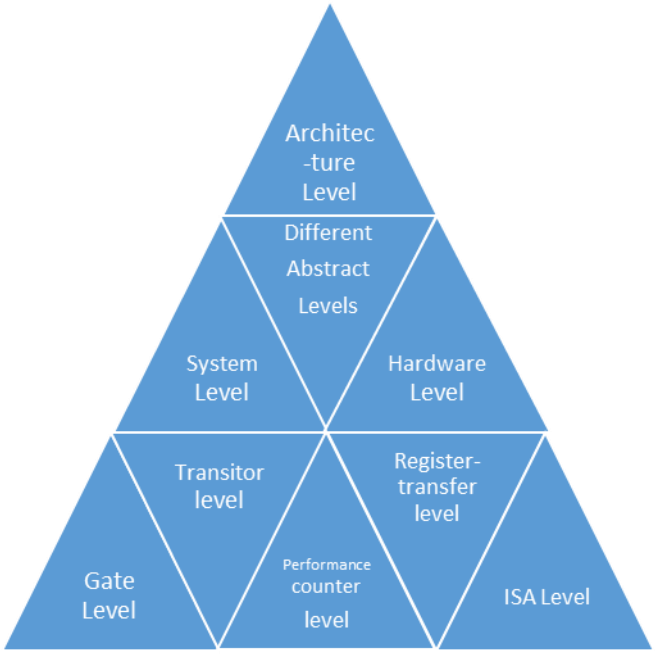

Figure 1: Classification of model

Processors and Compiler Level [6], but we are reducing the power at compiler level. When it comes to computer scientists a steady progress has been achieved basically in the form of Dynamic power management (DPM) and Dynamic voltage scaling (DVS) [7].

As per the survey of compiler, optimization is one of the most feasible ways for the developer to minimize the power consumption and improves the performance in multicores and different frequencies and voltages by selecting optimal using the Dynamic voltage and frequency scaling is the best optimization process to reduce the power consumption[21].

Related Work

The most effective power reduction technique is Dynamic voltage scaling. This result reducing the power supply voltage that can notably reduce power dissipation. It could be appropriate for eliminating idle times at low workload hours.

So power is not wasted by an idle processor. CPU consumes much power in convex fashion with frequency that can be reduced by using dynamic voltage scaling which makes CPU lower dynamic energy consumption.

Power-reduction can be done in two ways static and dynamic. Static techniques are applied at the time of design, such as compilation. Dynamic techniques are applied at the time of run time based on the workloads. Dynamic power *management* (DPM). When high performance is required, DPM allows hardware to consume more power; otherwise, the hardware enters a lower-power state. DPM techniques include dynamic voltage/frequency scaling (DVS/DFS) and clock gating. DVS/DFS finds the program section where voltage and frequency can be tuned on CPU with minimum loss in performance. To maintain the both energy and performance is vital role in DVS was introduced, this will help to apply different voltages for different executions of frequencies. (DVS) will allows the devices with change in voltage, increasing energy levels and efficiency of their operation in progress. DVS is used to reduce power by varying the voltages according to the load on the processor. Basically processors obtaining a power in two ways. One is through a compiler, second is an Assembly code manipulation or by another non-compiler method. Dynamic voltage scaling is a non-compiler method. On-compiler method checks the load on the processors and dynamically increases or decreases the processor frequency. DVS is one of the feasible and effective solutions to power reduction techniques. As a result lowering the supply voltage can reduce significantly lowering the power dissipation. It is suitable for eliminating idle times during low workload periods it leads no power wasted by an idle processor usually. Since the System processor power consumption increases in convex fashion, but DVS will helps to considerably reduce the system energy consumption. (DVS) is a mechanism dynamically adjust CPU voltage and frequency. DVS in embedded devices variation in processor utilization, lowering the frequency when the processor in less load, and running at maximum frequency when the processor is very largely loaded. DVFS will reduce energy systems. Because the frequencies are proportional to voltages.

A major challenges in DVS are utilizing the application are need to reduce the power .Voltage scaling is a common technique to reduce power by simply adjusting the supply voltage either at design time or at run time to maximize energy efficiency. The developer can implement different optimization techniques and can choose the one which gives the best result in terms of energy (Joule) and run-time (Sec). The code can be tuned dynamically by varying frequency and voltage across the blocks or the regions in the code. In such a way that minimization in the energy consumption can also be obtained dynamically

Analysis

The less power consume by the CMOS Technology. A Power Consumption of CMOS Formula:

$$p = c\, v^2\, f \qquad (1)$$

Where p= power in watts, c = switch capacitance, v = supply voltage, and f is the clock frequency in hertz [15] this Suggests that there are essentially three ways to reduce power:

DVFS technique proposed to achieve low power consumption for the CPU. Describe the relationship between CPU clock frequency, power and energy using the equations provided in the Intel optimization documentation. We let V_{dd} represent the supply voltage and f.

Power α fV^2_{dd} (2)

Delay $=1/f\alpha 1/V_{dd}$

Energy α V^2dd (3)

Traditional (DVS) will not fit address scaling on system power consumption as the leakage power increases.

The various power analysis tools are JouleTrack [16], WATTCH [17], SimpleScalar [18], XTREM [19], U [20], Simics, Cache Access and Cycle Time Information: CACTI, Simple Power, General Execution-driven Multiprocessor Simulator (GEMS), and WARTS - Wisconsin Architectural Research Tool Set. Joule Track is MIT research lab product and a very efficient web based tool for software profiling. WATTCH is CPU power estimation tool. It analyses and optimizes power dissipation at micro architectural level, where as Simple Scalar is the complete tool set. XTREM and XEEMU is Xscale architecture specific tool. SIMICS is full system simulator. CACTI is the tool for measuring performance based on cache sizes and organization**.** GEMS simulator based on SIMICS. WARTS performs profiling and tracing of the programs. Among all XTREM and XEEMU is Intel(c) Xscale(c) architecture specific tool. XEEMU developed to simulate the runtime and power consumption of the Intel(c) Xscale(c) core. With the experimental results it showed XEEMU is faster and efficient than XTREM.

Ideal vitality utilization of k number, undertaking can be characterized as discovering the best mix of accessible voltages and frequencies to perform a predefined task with k clock ticks inside a predefined time T.

$$E^{(k)} = \sum_{i=1}^{N} t_i^{(k)} P_d(f_i, v_i) + P_T(T^{(k)} - \sum_{i=1}^{N} t_i^{(k)}) \quad (4)$$

s.t

$$\sum_{i=1}^{N} t_i^{(k)} f_i = K^{(k)}$$

$$\sum_{i=1}^{N} t_i\,(K) \leq I^{(k)}$$

$$t_i^{(k)} \geq 0\,; for\ i = 1,2,\ldots,N$$

Minimize E= $t_1 p_d(f_1, v_1) + t_2 p_2(f_2, v_2) + t_3 p_3(f_3, v_3)$ (5)

s.t

1. $t_1 f_1 + t_2 f_2 + t_3 f_3 = K$
2. $t_1 + t_2 + t_3 = T$
3. $t_1 \geq 0; for\ i = 1,2,3$

$t_2 = T - t_1 - t_3$ (5)

$$t_1 f_1 + [T - t_1 - t_3] f_2 + t_3 f_3 = K$$
$$t_1 f_1 + T f_2 - f_1 f_2 - t_3 f_2 + t_3 f_3 = K$$
$$[f_3 - f_2] t_3 = K - t_1 f_1 - T f_2 + t_1 f_2 = K - t_1 f_1 + (t_1 - T) f_2$$
$$T_3 = \frac{(K - T f_2) - t_1 (f_1 - f_2)}{f_3 - f_2} \quad (6)$$
$$f_2 = T - t_1 - t_3$$
$$= T - t_1 - \frac{(K - T f_2) - t_1 (f_1 - f_2)}{f_3 - f_2}$$
$$= \frac{T(f_3 - f_2) - t_1 (f_3 - f_2) - (K - T f_2) + t_1 (f_1 - f_2)}{f_3 - f_2}$$

Optimize the solution

$$t_2 = T f_3 - T f_2 - t_1 f_3 + t_1 f_2 - K + T f_2 + t_1 f_1 - t_1 f_2)$$
$$t_2 = \frac{(t f_3 - K) + t_1 (f_1 - f_3)}{f_3 - f_2}$$
$$t_1 \geq 0 \quad ,$$
$$t_2 \geq 0$$
$$t_3 \geq 0$$
$$t_2 \geq 0$$
$$(T f_3 - K) + t_1 (f_1 - f_3) \geq 0$$
$$(T f_3 - K) \geq -t_1 (f_1 - f_3)$$
$$(T f_3 - K) \geq t_1 (f_3 - f_1)$$
$$t_1 \leq \frac{T f_3 - K}{f_3 - f_1}$$
$$t_1 \leq \frac{K - T f_3}{f_3 - f_1}$$
$$t_3 \geq 0$$
$$(K - T f_3) - t_1 (f_1 - f_2) \geq 0$$
$$(K - T f_3) \geq t_1 (f_1 - f_2)$$
$$\frac{K - T f_2}{f_1 - f_2} \geq t_1$$
$$t_1 \leq \frac{K - T f_2}{f_1 - f_2}$$
$$t_1 \geq 0$$

Result

$$0 \le t_1 \le \frac{K - Tf_2}{f_1 - f_2}$$

$$E = t_1 P_1 + t_2 P_2 + t_3 P_3 \quad (7)$$

$$E = t_1 P_1 + \left\{\frac{(Tf_3 - K) + t_1(f_1 - f_3)}{f_3 - f_2}\right\} P_2 + \left\{\frac{(K - Tf_3) - t_1(f_1 - f_2)}{f_3 - f_2}\right\} P_3$$

$$E = \frac{t_1 P_1 (f_3 - f_2) + (Tf_3 - K) P_2 + t_1 (f_1 - f_3) P_2 + (K - Tf_3) P_3 - t_1 (f_1 - f_3) P_3}{f_3 - f_2}$$

$$= t_1 \left[\frac{P_1 (f_3 - f_2) + (f_1 - f_3) P_2 - (f_1 - f_2) P_3}{f_3 - f_2}\right] + \left[\frac{P_2 (t_3 f_3 - K) + (K - Tf_2) P_3}{f_3 - f_2}\right]$$

$$= t_1 \alpha + \beta$$

$$if\ \alpha \le 0$$

$$t_1 = \frac{K - Tf_3}{f_1 - f_2}$$

$$t_{3=0}$$

$$t_2 = T - t_1 - t_3$$

$$t_2 = T - \frac{K - Tf_2}{f_1 - f_2}$$

$$= \frac{Tf_1 - Tf_2 - K + Tf_2}{f_1 - f_2}$$

$$= \frac{Tf_2 - K}{f_1 - f_2}$$

Based on proposed optimal energy consumption computing for kth task of different voltage frequencies DVFS based the closest energy optimized calculated from equation 1 to 7.

Method

Energy

The energy represented as E and measured in the Joules,the consumption energy in T seconds and power measured in Watts (W). The goal of the proposed scheduling will reduce the clock speed that work on the processor and reduce voltage to the minimum needed of system frequency.

There are various optimization techniques we have already mentioned. Among all optimization techniques compiler loop optimization techniques plays a major role. Here compiler loop transformation techniques are taken into consideration. Among loop transformations Loop Inlining, Loop Jamming, Loop Reversal, Loop Termination, Loop Unrolling and Loop Inversion are implemented. Whereas among function preserving transformation Recursion removal and register variable

techniques are implemented. These techniques are implemented for minimizing the run time and consumption of energy [23].

```
1. Read x, n, sum, i
2. sum<-0
3. If-conditional i<=n
   Begin
   sum=sum+nextTerm(x,i)
   End
4.Return sum
```

Figure 1: Minimizing the run time and consumption of energy

```
1.Read x, n, sum , fact, mult, i, j
   2.    sum fl0
3For i=1 to n instep of 1
Begin
        For j=1 to i instep of 1
     Begin
fact = fact*j
mult = mult*x
      End
       4.sum = sum+ mult/fact
  End
    5.Return sum
```

Figure 2: For-loop structure with loop Inlining transformation

Already self-tuned in terms of optimization level so compiler methods are highlighted more over in comparison with the optimization level. The optimization techniques are implemented on simple programs like factorial and matrix multiplications. In loop Inlining the execution of the calling sequence gets eliminated.

```
1.Read x, n, l, m, i
2.  l ←1
3. m ←1
4.For i=1 to n instep of 1
   Begin
    m = m*i
    l = l*x
   End
5.Return l/m
```

Figure 3: For-loop structure with loop jamming transformation

```
1. Read x, n, i, sum
 2. sum fl0
 3. For i=n to 1 instep of 1
          Begin
4.  sum = sum+ nextTerm(x,i)
          End
5. Return sum
```

Figure 4: For-loop structure with loop reversal transformation

The average performance percentage improvement in terms of energy is 25.03 % and runtime is 24.78 %. After applying Loop Jamming transformation the reduction in energy consumption is obtained. The energy before and after optimization taken and bar graph is plotted against X-axis Fig 17. The X-axis holds iterations these iterations are from 500 to 500x10 similarly bar graph is plotted for runtime also Fig 6. The maximum energy difference is around 0.027 Joule at 5000 thousand iterations and runtime is around 0.90 Sec.

As indicated in figure 5 the investigation strategy incorporates three principle steps the first venture of this procedure comprises in looking the working point that fulfils the most extreme number of framework prerequisites. When the working point is checked and approved, the outline model can be looked into and upgraded. Anticipated and settled past the past two levels. We note there are three classes of obligations to be fulfilled relying upon the investigation level. The five star incorporates asset utilization requirements in term of accessible execution assets and their vitality utilization. The second stipulations class concerns the force plan, parallelism and schedulability prerequisites.

The framework plan investigation system tells about the diverse vitality levels and distinctive planning errands, the fitting recurrence levels of the processor regarding battery force of gadget and mapping programming errand ideally getting attainable arrangement

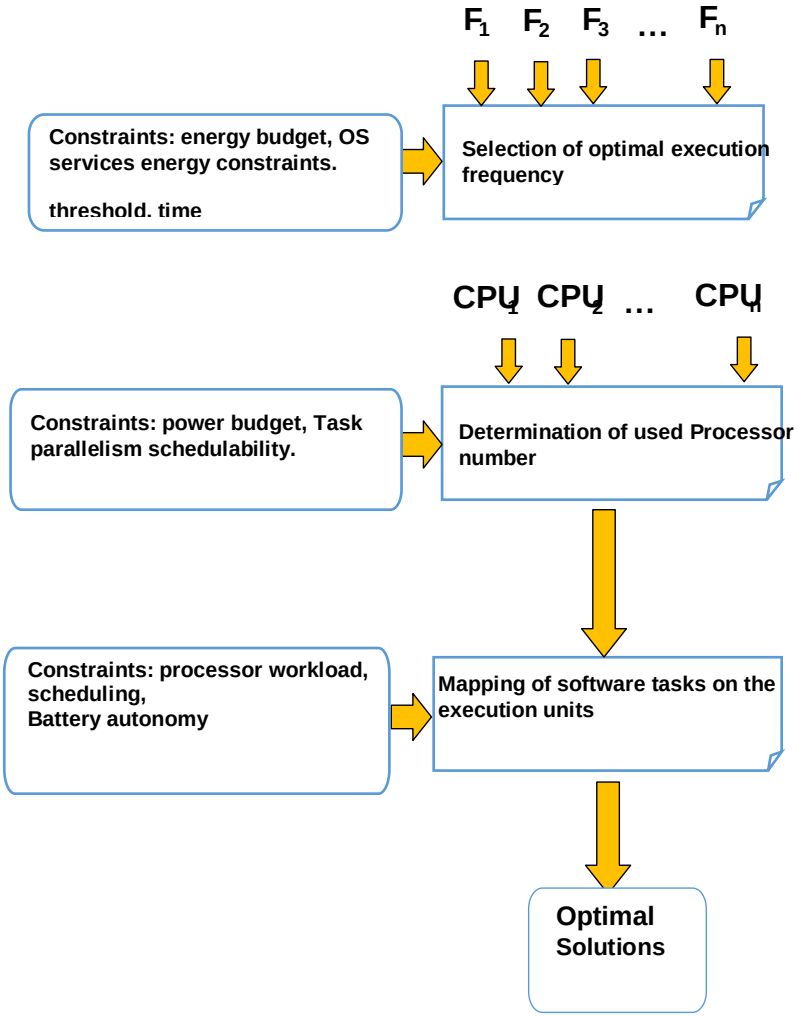

Figure 5: System design exploration methodology

Table 2: Average Percentage Performance of Energy and Runtime (Values taken from 500 to 500x10 Iterations)

Optimization Techniques	Eavg (Average Energy Performance Percentage)	Rtavg (Average Runtime Performance Percentage)
Loop In lining	0.0284	0.0612
Loop Jamming	0.0358	0.0768
Loop Reversal	0.0378	0.0813
Loop Unrolling	0.0357	0.0764
Loop Termination	0.0378	0.0812
Loop Inversion	0.0379	0.0812

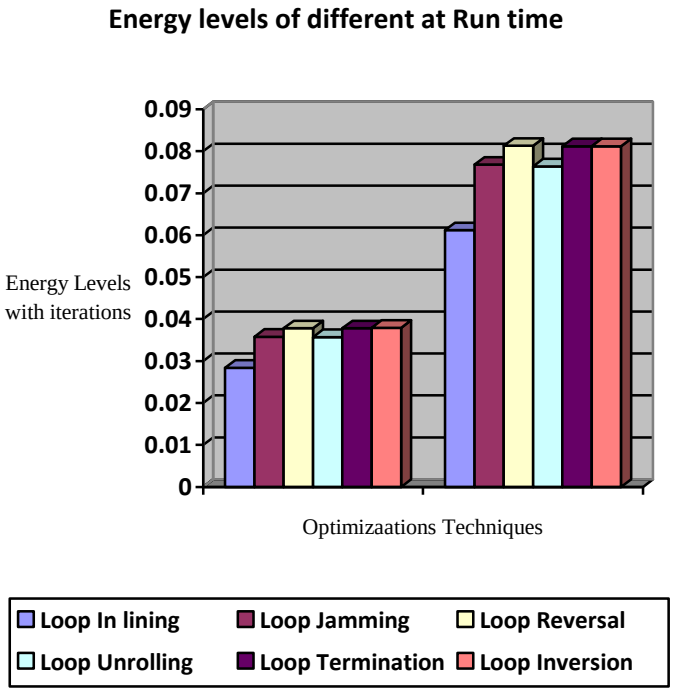

Figure 6: Different Energy Before and after applying Loop Techniques

Conclusion

There are many ways by which energy consumption will be reduced and reduced energy consumption and gives effective performance of system to reduce the energy usage and increase energy efficiency, operating systems need to be able to measure or estimate current power consumption, predict a tasks workload and control a series of power saving mechanisms. The component that decides which measures to activate in order to save power is called a power management policy. Due to the complexity involved in accurately estimating and predicting power consumption, today's approaches are heuristic. Some to the tools can capable to reduce static and dynamic voltages at different level in software point of view we can reduce the power at loop optimizations because loop are in order of the bench marks and closely we tested the a DVFS strategy that impacted the energy and time taken reduced in the result to minimizing energy usage during application execution.

Power consumption of embedded applications devices are important challenge. Future vision of energy related consumption will be important design concept because good design of system leads to the energy ware system.

References

[1] W. Kim, D. Shin, H. YUn, J. Kim, and S. Min . Performance comparision of dynamic voltage scaling algorithms for real-time systems. In proceedings of the symposium on Real-time and Embedded Technology and Applications, 2002

[2] L. Barroso and U. Holzle, "The case for energy-proportional computing," Computer, vol. 40, pp. 33–37, December 2007

[3] J. Tsao, Interpolation artifacts in multimodality image registration based on maximization of mutual information, IEEE Trans. Med. Imaging 22 (7) (2003) 854–864,doi:10.1109/TMI.2003.815077.

[4] Chih-Shun Ding, Chi-Ying Tsui, Member, IEEE, and Massoud Pedram, Member, IEEE "Gate- Level Power Estimation Using Tagged Probabilistic Simulation", IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems,Vol.17,No.11.(November 1998),Page No.54-66.

[5] K. Flautner, S. Reinhardt, and T. Mudge. Automatic performance setting for dynamic voltage scaling .In proceedings of the 5th symposium on Operating systems Design and Implementation, December 2002.IS Department , Ghent University Sinit Pietersnieuwstraat 41,B-9000 Gent, Belgium 2010.

[6] kenneth Hoste, Lieven Eechkout at.al "COLE: Compiler Optimization Level Exploration" Kenneth Hoste Lieven Eeckhout E.

[7] D. Marculescu. On the use of microarchitecture-driven dynamic voltage scaling. In Workshop on Complexity-Effective Design, June 2000.

[8] Advanced microdevices, Inc. Mobile AMD athlon 4 processor model 6 CPGA data sheet. Publication 24319, November 2001.

[9] Intel corporation Intel 80200 Processor based on Intel Xscale Microarchitecture: Developer's Manual. Order Number: 273411-003 (March 2003)..

[10] D. Shin, J. Kim, and S.Lee. Low-Energy intra-task Voltage scaling using static timing analysis. In proceddings of Design Automatic Conference, pages 13-23,1994.

[11] Zili shao,meng wang, ying chen, chun Xue, Meikang Qui, Laurence T. Yang, and Edwin H. –M.Sha, "Real-Time Dyanamic Voltage Loop scheduling for multi-core Embedded systems"IEEE Transactions On circuits and systems-li:Express Briefs, vol. 54, No. 5,May 2007,page No.445.

[12] R.J. Rost, OpenGL Shading Language, 2nd edition, Addison-Wesley Professional, 2006.

[13] T. Burd and R.Brodersen, "Energy Efficient CMOS Microprocessor Design," proc. 28th Hawaii Int'l Conf.on system sciences, 1995.

[14] D. Blythe, The Direct3D 10 system, ACM Trans. Graph. 25 (3) (2006) 724–734, doi:http://doi.acm.org/10.1145/1141911.1141947.

[15] W.R. Mark, R.S. Glanville, K. Akeley, M.J. Kilgard, Cg: a system for programming graphics hardware in a C-like language, in: SIGGRAPH'03: ACM SIGGRAPH, ACM Press, New York, NY, USA, 2003, pp. 896–907,doi:http://doi.acm.org/10.1145/1201775.882362.

[16] I. Buck, T. Foley, D. Horn, J. Sugerman, K. Fatahalian, M. Houston, P. Hanrahan, Brook for GPUs: stream computing on graphics hardware, ACM Trans. Graph. 23 (3) (2004) 777–786,doi:http://doi.acm.org/10.1145/1015706.1015800.

[17] D. Brokks, V. Tiwari, and M. Martonosi, "Wattch: A framework for Architectural-Level power analysis and optimizations", in proc. ISCA, Jun.2000,pp.83-94.

[18] Contreras, G., Martonosi, M., Peng, J., Ju, R., Lueh, G.Y.: XTREM: a Power simulator for the Intel Xscale core. SIGPLAN Not. 39(7), 115-125(2004).

[19] R. Strzodka, M. Droske, M. Rumpf, Fast image registration in DX9 graphics hardware, J. Med. Inform. Technol. 6 (2003) 43–49.

[20] zolt' an Herezegl, Akos Kissl, Daniel Schmidit2, NorbertWehn2, and tabor Gyim' Othyl "XEEMU: An improved Xscale powewr simulator", PATMOS conference held in Gothenburg,Sweden in September 2007.

[21] N. Courty, P. Hellier, Accelerating 3D non-rigid registration using graphics hardware, Int. J. Image Graph. 8 (1) (2008) 1–18.

[22] P. Muyan-Özc¸elik, J.D. Owens, J. Xia, S.S. Samant, Fast deformable registration on the GPU: a CUDA implementation of demons, in: The 2008 International Conference on Computational Science and its Applications,ICCSA 2008, IEEE Computer Society, 2008, pp. 223–233.

[23] Áron Csendes "Survey of Dynamic Voltage Scaling Methods for Energy Efficient Embedded" 8th International Conference on Applied Informatics Eger, Hungary, January 27–30, 2010. Vol. 1. pp. 413–420.

[24] Y. Wang, K. Li, H. Chen, L. He, and K. Li, "Energy-aware data allocation and task scheduling on heterogeneous multiprocessor systems with time constraints," IEEE Transactions on Emerging Topics in Computing, 2014.