Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

POWER-AWARE SYSTEM DESIGN FOR MULTIPROCESSORS AND VOLTAGE SCALING/FREQUENCY

K.SURESH, M.RAJASEKHARABABU

School of Computing Science and Engineering, VIT University, Vellore, TN, India

ABSTRACT

Most critical and major in growing technology for High Performance of demand of power & energy and an urgent problem in powering technologies. Energy optimization is an enabling power Management. The Consumption of Energy Should be ascertainable not only to Gate Level or Register Transfer (RT) Level but also to the System Level. Reducing the Energy Consumption of system not deviating the overall performance of the system. The compiler optimization will help to reduce power reduction at software level. Power management software level strategy is the code optimization by measuring the difficulties at where we can get profitable of investigate optimization criteria to minimization of overall energy consumption . The Energy consumption and run time computed for various compiler techniques on XScale Architecture using XEEMU tool. The optimized code picked out and code is tuned dynamically by varying voltage-frequency. The optimized codes are tuned dynamically.

Keyword:- *Compiler Optimization, Performance Evaluation, Voltage-Frequency Scaling, XScale Architecture.*

1. INTRODUCTION

In present day world every joule of 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 operating 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.

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. we 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 power efficiency of the system is important issue.considering the state of the art are in complexity and very important high performance computing processors traditional methods of operation static mode o methods are power is fixed operation of voltage and frequeny ,but not suitable for proportionately dynamic power system management is becoming an important issue

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 beacuase 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], Table I: Comparison of Static Power Management (SPM) and Dynamic Power Management (DPM) techniques

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SPM (off-line optimization)			
System/ Component Under Test (SUT/CUT)	Level of Detail	Evaluation Methodology	Description
CPU	Cycle level or RTL	Cycle-level Simulation	*Power Timer* , *Wattch* and *SimplePower* energy models
	Instruction level	Instruction-level simulation	Power Profiles for *Intel 486DX2*, *Fujitsu ,SPARClite '934* and *PowerPC*
System	Hardware component level (e.g. hardware state: CPU sleep/ doze/busy, LCD on/off etc.)	Functional simulation (Parameters via measurements)	*POSE* (Palm OS Emulator)
	Software component level (procedure/process/task)	Measurements (with monitoring tools)	Time driven sampling, *PowerScope* and Energy driven sampling
	Hardware & Software component level	Complete system simulation (CPU, Disc, Memory, OS, Application)	*SoftWatt* built upon *SimOS* system simulator
DPM (on-line optimization)			
(SUT/CUT)	Implementation level	Methodology	Description
CPU	CPU and System Software	DVS (Dynamic Voltage Scaling)	Interval-based scheduler and Real-time schedulers (Inter-task, Intra-task)
System	Components hardware (Disks, network interfaces, displays, I/O devices, etc.) and system software	Low power mode of operation	Shutdown/low- power unused devices
Cluster System	Multiple systems coordination (server clusters…)	CVS (Coordinated Voltage Scaling)	Coordinated DVS between multiple nodes

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. Dynamic voltage and frequency scaling is the best optimization process to reduce the power consumption.

Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. 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 bot h 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 porgess . 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 wer 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

3. ANALYSIS

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

$$p = c\, v^2 f$$

where p= power in watts, c = switch capacitance, v = supply voltage , and f is the clock frequency in hertz [15] this

Fig.2: If-conditional structure with Loop Inversion

suggests that there are essentially three ways to reduce power:

DVFS technique proposed to achieving low power consumption for the CPU.We 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 .

$$\text{Power} \alpha\ fV^2_{dd}$$

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

$$\text{Energy} \alpha\ V^2 dd$$

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

The various power analysis tools are Joule Track[16], WATTCH[17], Simple Scalar[18], XTREM[19], XEEMU[20], Simics, Cache Access and Cycle Time Information: CACTI, Simple Power, General Execution-driven Multiprocessor Simulator (GEMS), 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

Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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

4. 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.

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

Fig 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
```

Fig.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
```

Fig.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
```

Fig.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 5. The maximum energy difference is around 0.027 Joule at 5000 thousand iterations and runtime is around 0.90 Sec.

Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table II 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

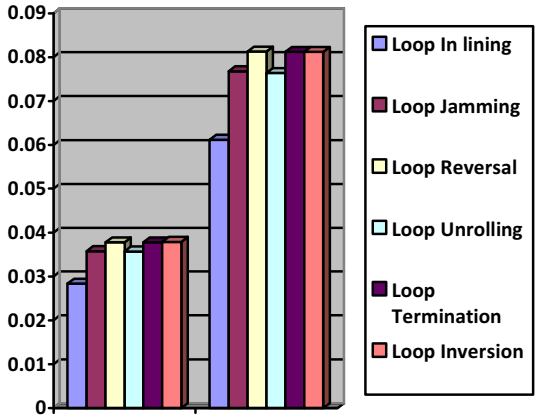

Fig.5 Energy Before And After Applying Loop Techniques

5. 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.

6. FUTURE ENHANCEMENT

In this paper compiler transformation techniques implemented on small program like factorial and matrix multiplication. In the same way these optimization techniques can be implemented on very large code by the programmer. Even for dynamic tuning of voltage and frequency inside the code helped in achieving the performance. Another algorithmic technique or can say hybrid algorithm can be implemented with this basic concept of DVFS .The entire techniques can also be taken to the parallel environment to get better result.

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

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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.
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).

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

[19] 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.

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

[21] 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.