Performance and Power Consumption Trade-offs for a VLIW DSP

Mostafa E. A. Ibrahim†, ‡, Markus Rupp†, and S. E.-D. Habib‡*
†Institute of Communications and RF Engineering-Vienna University of Technology, Austria
‡ Electronics and Communication Department Faculty of Engineering-Cairo University, Egypt
Email: {mhalas,mrupp}@nt.tuwien.ac.at, seraged@ieee.org

***Abstract*— Since performance and power consumption optimizations are crucial issues in embedded systems, it is necessary to find a trade-off between these optimization goals. This paper explores the performance and power trade-offs of VLIW processor, specifically the Texas Instruments TMS320C6416T DSP. We evaluate the effect of the global performance optimizations as well as a specific architecture feature on the power consumption of the targeted processor while running typical digital signal and image processing algorithms. We assess the specific C64x+ architecture feature, Software Pipelined Loop (SPLOOP), effect on the power consumption and the performance as well. The binaries used in this study were generated using the Texas Instrument `C/C++` Compiler, which allows control over the whole set of optimizations.**

I. Introduction

Since the 1990's, as the portable system market grows rapidly and reliability problems due to high power dissipation are becoming an issue for systems operating in high clock frequency, low power design is becoming more and more important and is now one of the major concerns in system design.

Digital signal processors are frequently used in embedded systems to permit application specifications in software. Since performance and power consumption optimizations are crucial issues in embedded systems, it is necessary to find a trade-off between these optimization goals.

Compilers traditionally are not exposed to the energy details of the processor. Current compiler optimizations are tuned primarily for performance and code size. Hence, it is important to evaluate how these optimization options influence power and energy consumption within the processor while running a software algorithm.

In this paper, we present a quantitative study wherein we examine the effect of the global performance optimization levels -o0 to -o3 of the compiler on the energy and power consumption of the TMS320C6416T (for the rest of the paper it is referred to as C6416T for brevity) DSP from Texas Instruments. In our experiments we use the Code Composer Studio (CCS), the Texas Instruments C/C++ compiler Ver.6.0.1, to produce the code binaries. First, we evaluate the effect of the global performance compiler optimizations on the energy and the power usage of the targeted processor as well as other performance measures such as the memory references, the cache miss rate and so on. Second, We assess the special C64x+ architecture feature SPLOOP on the performance as well as the power consumption.

The rest of the paper is organized as follows: Section II describes prior research efforts related to this work. Section III presents a general overview of the target architecture along with the experimental platform. Section IV examines the effect of invoking various global compiler performance optimizations and the specific architecture feature SPLOOP on the power consumption and performance measures. Finally, Section V summarizes the main contributions of this paper.

II. Related work

Recently some attempts to understand the scope of compiler optimizations, from the perspective of power dissipation and energy consumption, of programmable processors have been introduced. Valluri et al. [1] provided an evaluation of some general and specific optimizations in terms of the power/energy consumption of the Alpha processor while running some SpecInt95 and SpecFp95 benchmarks.

Kandemir et al. [2] studied the influence of five high-level compiler optimizations ,such as loop unrolling and loop fusion, on energy consumption with the help of a cycle-accurate energy simulator (SimplePower), a source-to-source code translator, and a number of benchmark codes. Seng et al. [3] revised the effect of the Intel compiler general and specific optimizations, for energy and power consumption, for a Pentium 4 processor running some benchmarks extracted from Spec2000.

Zafar et al. [4] examined the effect of loop unrolling factor, grafting depth and blocking factor on the energy and performance for the Philips Nexperia media processor PNX1302. As they interchangeably use the term energy and power for the same meaning, the achieved improvement in energy was directly related to the performance enhancement. Finally, Casas et al. [5] studied the effect of various compiler optimizations on the energy and power usage of the low power C55 DSP from Texas Instruments. Their work was based on a physical measurement platform for measuring the current drawn by the DSP core. Their work did not consider the effect of the compiler optimizations on many performance measures that significantly affect the power and energy usage, such as the memory referencing and the instructions per cycle.

*This work has been funded by the Christian Doppler Laboratory for Design Methodology of Signal Processing Algorithms as well as the comet funded the K-project: Embedded Computer Vision

III. Experimental Setup

All power measurements are carried out on the C6416T DSP Starter Kit (DSK) manufactured by Spectrum Digital. The operating frequency, in our setup, is 1000MHz and the DSP core voltage is $1.2V$. The current drawn by the DSP core, while running a software algorithm, is captured using the Agilent 34410A $6\frac{1}{2}$ digit Digital Multi-Meter(DMM). Typical signal and image processing benchmarks from Texas Instruments libraries are used for demonstration purpose. The input data for all used benchmarks are located in the internal data memory. The C/C++ compiler embedded in the Code Composer Studio (CCS3.1) from Texas Instruments is used for getting the binaries. This compiler has many levels of optimization that are mainly tuned for optimizing speed and code size.

IV. Results and Analysis

First, we evaluate the effect of the global compiler performance optimizations (-o0 to -o3) on the energy and power consumption of the targeted processor. Second we analyze the effect on other performance measures such as the memory references, the cache miss rate and so on. Third, We assesses the special C64x+ architecture feature Software Pipelined Loop (SPLOOP) on the performance as well as the power consumption.

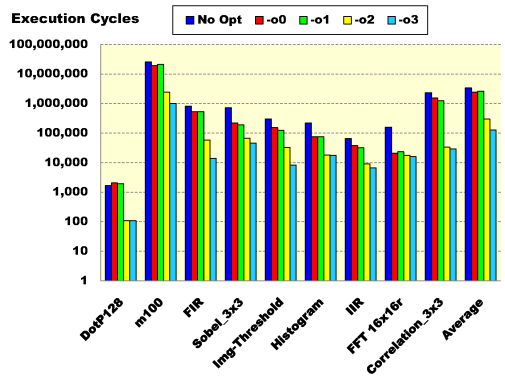

Fig. 1. Various optimizations versus execution cycles.

A. Performance and Power Consumption Trade-off

Figure 1 presents the execution cycles versus the four optimization levels (-o0 to -o3) for various benchmarks. The most aggressive optimization option -o3 reduces the execution time, on average, by 96.20%[§]. Although Fig. 1 demonstrates that these optimizations significantly decrease the execution time, Fig. 2 shows that these optimization options increase the power consumption, on average, by 24.4%. This percentage reaches 45% for some individual benchmarks, such as the correlation_3 $\times$ 3. On average, invoking -o2 or -o3 leads to much power consumption than invoking -o0 or -o1. The software pipelining loop feature is enabled with -o2 and -o3 allowing better instruction parallelization, and as we will explain later this have a significant impact on the power consumption.

[§]All the saving or increase percentages are obtained comparing to the case when no optimization option is invoked.

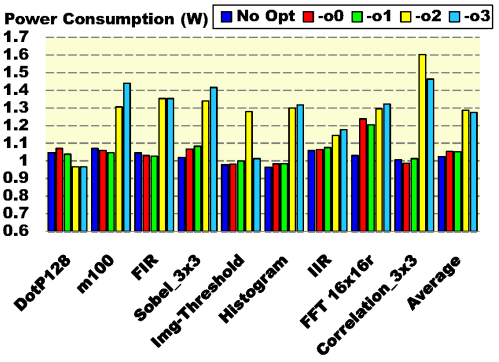

Fig. 2. Various optimizations versus consumed power.

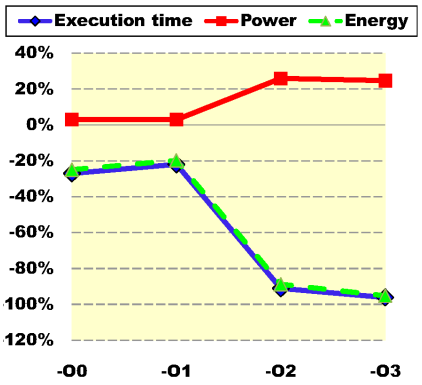

Fig. 3. Power, execution time and energy rate of change versus different optimization options.

Figure 3 demonstrates that there is a perfect correlation between execution time and energy consumption. The most aggressive performance optimization level -o3 reduces the execution time on average by 96.20%, while it reduces the energy on average by 95.40%. It is obvious that invoking -o2 or -o3 provides substantial energy saving than when invoking -o0 or -o1. This can be explicated by the fact that, at -o2 and -o3 the software pipelining loop is enabled, consequently leading to considerably higher reduction in the execution time and hence in the energy usage.

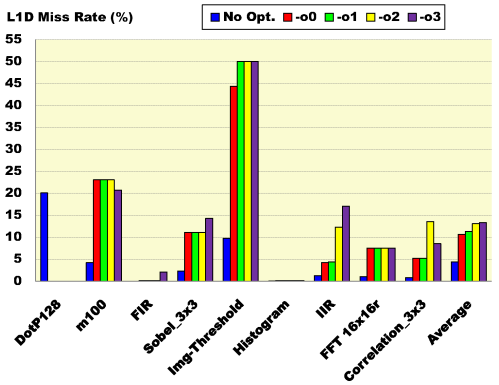

Fig. 4. Impact of optimizations on the L1D miss rate.

B. Optimizations Effect on Performance Measures

In order to analyze the previous results we find that it is worth to study the effect of the compiler optimizations on

four important performance measures: cache misses, memory references, instructions per cycles, and CPU stall cycles.

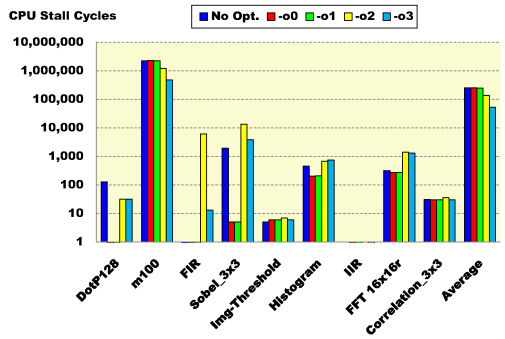

Fig. 5. CPU stall cycles versus different optimization options.

Figure 4 exposes the effect of different optimization levels on the level 1 data (L1D) cache miss rate. The L1D cache miss rate increases, on average, by almost 200% when -o3 is invoked. The L1D cache misses consequently require L2 cache/SRAM access which in turn provide additional power consumption. Knowing that one data cache miss causes six stall cycles, we may expect the CPU stall cycles to increase as the miss rate increases. But, as shown in Fig. 5 the CPU stall cycles are decreased by 78% when also -o3 is invoked.

The first reason that causes the previous result is that, the L1D cache of this DSP pipelines read misses. Hence, a single L1D read miss takes six cycles when serviced from L2 SRAM, and eight cycles when serviced from L2 cache. Miss pipelining can hide much of the miss overhead (CPU stall cycles) by overlapping the processing of several cache misses. The miss overhead can be expressed as $(4 + (2 \times M))$ when serviced from L2 SRAM or as $(6 + (2 \times M))$ when serviced from L2 cache where M is the number of misses [6]. Therefore, the cache misses pipelining provides significant reduction in the execution time but it still has no effect on the power consumption.

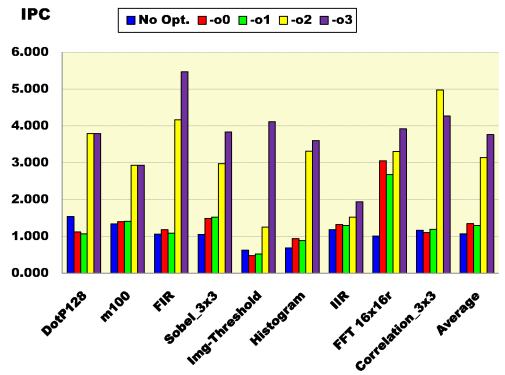

Fig. 6. Effect of various optimization options on the instructions per cycle.

The second reason is that the write cache miss does not directly stall the CPU because of the use of a L1D Write buffer [6]. This also, affects the execution time but has no effect on the power consumption especially when the write buffer is not full.

Figure 6 illustrates that the instructions per cycles (IPC) increased by 250% when -o3 is invoked. This surly decreases the execution time and consequently the energy usage, as more overlapping in the instructions execution per cycle will be achieved. But, this results in higher parallelization degree which in turn increases the power consumption.

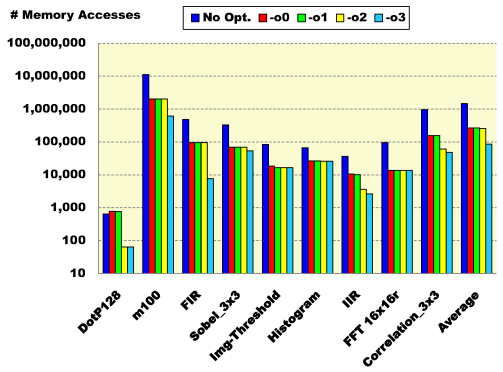

Fig. 7. Effect of different optimization options on the memory references.

Although, Fig. 7 points out that the memory references are decreased by 94% which is expected to save the consumed power, we find that the power is increased. This emphasizes our results in [7], [8] that the Instruction Management Unit (IMU), the unit which is responsible for fetching and dispatching instructions, contribution to the total power consumption dominates the memory referencing contribution.

C. Specific architectural optimization effect on power consumption

SPLOOP is a type of instruction scheduling that exploits Instruction Level Parallelism (ILP) across loop iterations. It is a specific architectural optimization feature of the C64x+ CPU, the C64x CPU does not support the SPLOOP.

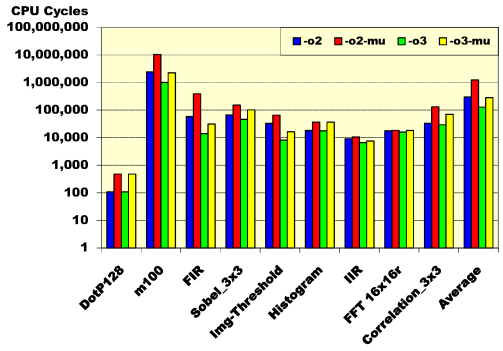

Fig. 8. various optimizations versus execution cycles.

This feature allows the CPU to store a single iteration of loop in a specialized buffer and contains hardware that will selectively overlay copies of the single iteration in a software pipeline manner to construct an optimized execution of the loop [9]. The SPLOOP feature is implicitly enabled with the global optimization options -o2 and -o3. However, we override this by invoking the -mu option which disables only the SPLOOP feature. To distinguish between the case when the software pipelining is enabled or disabled, we use the term -o2-mu and -o3-mu to indicate that the SPLOOP feature is disabled. In this section we evaluate the impact of the SPLOOP

on the power and energy consumption as well as the previous four performance measures.

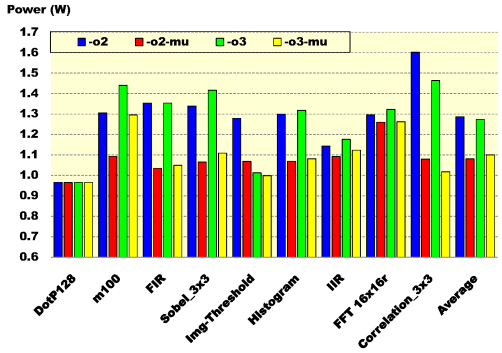

Fig. 9. Impact of software pipelining on the consumed power.

First,we examine the effect of the SPLOOP on power and energy consumption. Figure 8 clearly illustrates the strong impact of this feature on the execution cycles. The execution time increases by 317.5% and 120.9% when the -o2-mu and -o3-mu are invoked, respectively. This increase is relative to the case when -o2 and -o3 are invoked. It is noticeable that the impact of disabling the SPLOOP on the execution cycles is higher when invoking -o2 than when invoking -o3. Since -o3 includes all the individual optimizations that exists in -o2 as well as more performance oriented optimizations that aim to reduce the execution cycles.

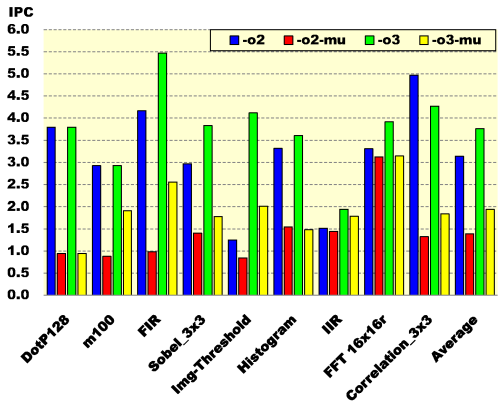

Fig. 10. Software pipelining effect on IPC.

Despite the increase in the execution cycles when SPLOOP is disabled, the power consumption decreases, on average, by 16% and 13.65% when -o2-mu and -o3-mu are invoked, respectively. Figure 9 shows the power consumption reduction for all the the benchmarks.

We find that disabling the SPLOOP has no effect on the memory references and the L1D cache miss rate. But it significantly affect the IPC as shown in Fig. 10. The IPC decreases, on average, by 55.75% and 48.5% when -o2-mu and -o3-mu are invoked, respectively, resulting in lower instruction parallelism rate and consequently resulting in the pre-mentioned decrease in the power consumption.

The power consumption increases, on average, by 7.41% when -o3-mu is invoked compared to the case when no optimization option is invoked. Hence, the SPLOOP contributes by 69.7% to the total power increase when -o3 is invoked. Therefore, more attention to the design of the specialized hardware for the SPLOOP should be paid to compromise the performance and power trade-offs for the C6416T.

V. Conclusion

The growing use of digital signal processors in embedded systems necessitates the use of optimizing compilers supporting special architecture features. In this paper, we explore the performance and power trade-offs of the commercial off-the-shelf VLIW processor C6416T from Texas Instruments. We evaluated the effect of the global performance optimization options -o0 to -o3 on the power consumption. The results show that the most aggressive performance optimization option -o3 reduces the execution time, on average, by 95%, while it increases the power consumption by 24.4%.

In order to analyze the main cause of this power increase we inspect the optimizations effect on some other performance measures, such as the memory references and the data cache miss rate. The expected power saving due to the memory references decrease is overridden by the power increase, due to the IPC increase. Moreover, we assess the C64x+ software pipelined loop feature effect on the power consumption and the performance as well. The results show that the software pipelined loop feature contributes, on average, by 69.7% to the total power consumption increase when -o3 is invoked. Hence, more attention to the design of the specialized hardware for the software pipelined loop should be paid to compromise the performance and power trade-offs for the C6416T.

References

[1] M. Valluri and L. John, "Is Compiling for Performance == Compiling for Power?" in *Proceedings of the 5th Annual Workshop on Interaction between Compilers and Computer Architectures*, Monterrey, Mexico, Januarry 2001.

[2] M. Kandemir, N. Vijaykrishnan, and M. J. Irwin, "Compiler Optimizations for Low Power Systems," *Chapter 10, Robert Graybill and Rami Melhem: Power aware computing*, pp. 191–210, 2002.

[3] J. S. Seng and D. M. Tullsen, "The Effect of Compiler Optimizations on Pentium 4 Power Consumption," in *Proceedings of the Seventh Workshop on Interaction between Compilers and Computer Architectures INTERACT'03*. Washington, DC, USA: IEEE Computer Society, February 2003, pp. 51–56.

[4] N. Z. Azeemi and M. Rupp, "Energy-Aware Source-to-Source Transformations for a VLIW DSP Processor," in *proceedings of the 17th ICM'05*, Islamabad, Pakistan, December 2005, pp. 133–138.

[5] C. B. J. G. Miguel Casas-Sanchez, Jose Rizo-Morente, "Effect of Compiler Optimizations on DSP Processor Power and Energy Consumption," in *Conference on Design of Circuits and Integrated Systems (DCIS'07)*, Seville, Spain, November 2007.

[6] Texas Instruments, *TMS320C6416T,DSP Two-Level Internal Memory Reference Guide*, February 2006, sPRU610C. [Online]. Available: www.ti.com

[7] M. E. A. Ibrahim, M. Rupp, and S. E.-D. Habib, "Power Consumption Model at Functional Level for VLIW Digital Signal Processors," in *Proceedings of the Conference on Design and Architectures for Signal and Image Processing (DASIP'08)*, Bruxelles, Belgium, November 2008, pp. 147–152.

[8] M. E. A. Ibrahim, M. Rupp, and H. A. H. Fahmy, "Power Estimation Methodology for VLIW Digital Signal Processor," in *Proceedings of the Conference on signals, systems and computers (SSC'08)*, Asilomar, CA, US, October 2008.

[9] Texas Instruments, *TMS320C64x/C64x+, DSP CPU and Instruction Set Reference Guide*, July 2007, sPRU732D. [Online]. Available: www.ti.com