POWER AUDIT OF A SPACE-CERTIFIED MICROPROCESSOR

Guillermo González de Rivera, Javier Garrido, and Eduardo Boemo

School of Computer Engineering, Universidad Autónoma de Madrid
Ctra. de Colmenar Km. 15, 28049 Madrid, Spain.
http://www.ii.uam.es

Abstract.- *In this paper, an on-board μP system is characterized and the efficacy of some straightforward LPD techniques is quantified. As results, some system-level design rules to save power, even in a highly optimized space-certified microprocessor are obtained. The MA31750 chip has been selected as technological framework for the experiments.*

1. INTRODUCTION

Low-power design (LPD) results mandatory in spacecraft electronics. It decreases the energy requirements and improves reliability by lowering the temperature of the chips. However, the exhaustive process of certification of hard radiation devices [1] makes difficult the silicon manufacturers a fast incorporation of state-of-the-art techniques. This limitation does not exist in the next design step: several LPD ideas can be applied during the construction of on-board electronic systems. In this paper, the space-certified MA31750 [19] microprocessor has been selected as technological framework to test some published ideas.

The construction of low-power microprocessors, initially restricted to electronic watches, is today an active research line, impelled by the forces of cellular phone and portable computer markets. In this area, two complementary approaches can be identified: methodologies applied to the design of custom microprocessors and cores [2]-[5]; and the development of strategies to improve the power figure of existing architectures [6]-[9]. Additionally, considering that microprocessors are large and heterogeneous circuits, these research lines are benefited from almost all earlier LPD studies. An annotated list of main techniques can be found in [10]-[12].

A first step to develop a LPD plan on a particular microprocessor is to determine the power figure of the principal blocks that compose the board. This budgeting allows the designers to forecast the efficacy of a given LPD idea, or to identify bottlenecks in terms of power. Different approaches to analyze μP power have been published. In [7] a study on a semi-custom R3000 is presented. In [13] and

[14], the results of two contemporary microprocessors of Intel and Digital are described. Finally, in [15] the power parts of a portable communication processor are summarized.

In this paper, some of the above ideas are adapted to analyze and reduce power in a space-certified MA31750 microprocessor system. In the next section, the main hardware and software strategies developed to determine the power components are described. In section 3, the main results and the design rules obtained are summarized.

2. POWER BUDGET IN A μP-BASED SYSTEM

The analysis of a microprocessor board requires a set of actions in both hardware and software to increase the controllability and observability of the system. In the hardware side, all test points and different circuit configurations must carefully planned before starting the PCB design. Late modifications like the attachment of adapters, sockets, extra wires, components, etc., diminish the reliability of the board, make difficult the debugging, and produce distortions in the power measurements. In this work, the main features added to prototype board were:

- Five power independent supply lines for: a) the microprocessor, b) the 8KB ROM (CY7C261–25) [17], c) the 8KB RAM (CY7C185–20) [17], d) the chips in charge of address bus coding, and finally e) the miscellaneous logic (clock generator, address decoding, pull-ups, LEDs, etc.). Different voltages can be applied to each block.
- Test points required to measure the input current and the power supply voltage in each block.
- Micro switches to modify the circuit configuration.
- A set of sockets inserted in the address bus (between the μP and the memory), to test bus-coding techniques.
- An array of connectors bonded to each bus wire. The goal is to add extra capacitance for the emulation of different working conditions.
- Circuitry to externally control the μP and load different RAM memory contents (MA31750Console Mode).
- General testing points to connect the logic analyzer probes.

Complementary, a set of benchmark routines was developed. Some of them do not perform any specific action, while others execute the same computation in different ways. The routines embrace miscellaneous situations like the use of particular registers, random and sequential R/W operation over the RAM, different combination of instructions, etc. All routines run in an infinite loop to make possible to measure the average chip power [16]. Their main characteristics are summarized in Table I. Other reference on benchmark routines is [14].

Name	Goals	Description
base	Determination of the power consumption for typical operation. The routine activates the main microprocessor blocks.	Loop that performs R/W of internal registers, logic/arithmetic operations between registers, and access to ROM, RAM and E/S positions.
tipregx	Determination of the power consumption associated to particular registers. The x prefix corresponds to the register number.	Loop that executes a swap between the low and high byte of the x-numbered register.
summemx	Generation of activity in the external buses. Power consumption associated to RAM memory.	Loop that adds to a register a 16-bit datum stored in RAM. The x suffix corresponds to the register number.
sumregx	Generation of internal activity in the μP. Power consumption associated to different registers.	Loop that adds to a register a 16-bit datum stored in other register.
move	Comparison between routines that perform identical computation. This program is equivalent to the *load/store* one.	Loop that moves a memory block of 1000h positions stored in RAM to another part of the memory using the *move* instruction.
load/store	Complementary to *move.*	The *move* routine but using *load/store* instructions.

Table I: Benchmark routines for power analysis.

3. EXPERIMENTAL RESULTS

In this section, the main experiments are presented. They have been separated in accordance with the strategy utilized. All values corresponds to a fixed clock frequency of 10 MHz. Power has been calculated indirectly by measuring the average input current that enters to each block.

3.1 ROM vs. RAM execution

In embedded applications, is usual to situate a fixed program in a ROM device. However, this practice is highly negative in terms of consumption. In our case, the total average power resulted 530 mW for a program stored in ROM, meanwhile this value was reduced to 238 mW (a power saving of the 45 %) for a RAM operation. In each measurement, the memory that was not being utilized was disabled.

In Fig.1 are depicted the µP, RAM, and ROM power components, when the benchmark program *base* is stored in ROM (light gray) or RAM (white). In both cases, the µP power consumption is similar, due to the bus capacitance is the same. The ROM chip utilized in the board [17] exhibits a static power of near 39 mW when disabled. As will be illustrated below, this relatively large value influences on the efficacy of others LPD techniques.

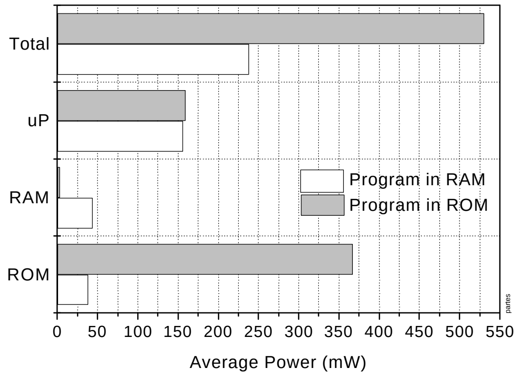

Fig.1: µP, RAM and ROM power components, when the program is executed from ROM (light gray) or RAM (white). Benchmark routine *Base*.

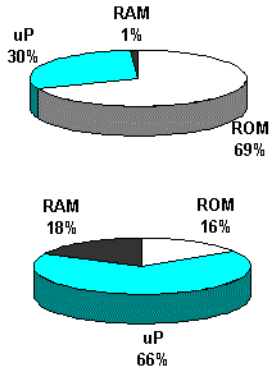

Fig.2: weight of the microprocessor power for program stored on ROM (above) or RAM (below). Benchmark routine *Base*.

Three principal options are available at system level to minimize the ROM power: a) to minimize memory size and usage, which should be enabled just to store a listener or bootstrap routine (to start a serial communication to receive the complete program); b) if an original copy of program must be stored in ROM, to transfer it to RAM (that would act in a shadow-RAM fashion) and then, disconnect the ROM power supply line; c) to distribute the ROM contents in a set of smaller chips and enable their power supply line by sections.

If the ROM-based execution is discarded, the weight of the microprocessor power respect to the RAM became important: it is near the 65 % of the overall consumption (Fig.2). This fact indicates that CPU-oriented LPD techniques will be effective in the MA31750.

3.2. Characterization of the internal registers

The MA31750 has 16 internal general-purpose registers. In order to characterize them, the set of identical benchmark routines, named *TIPREGx* (x corresponds to the number of register) were used. They perform the addition of a set of data using the x-numered register. Although they are general-purpose registers, two of them resulted singular in terms of power. For example, the simple utilization of R2 instead of R3 can produce a microprocessor power reduction of the 13% (Fig.3). To the best of our knowledge, this information has not been reported in the manufacturer technical literature. The compiler would easily improve the μP power figure by exploiting this feature.

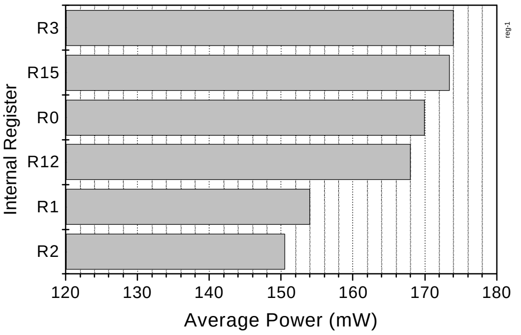

Fig.3: Microprocessor power consumption running benchmark routines *TIPREGx*.

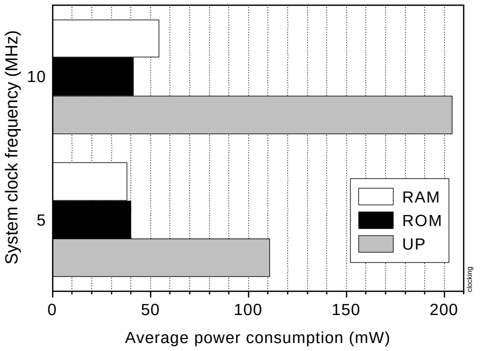

Fig.4: Power components (uP, ROM and RAM modules) measured at 5 and 10 MHz. Program stored on RAM. Benchmark routine *base.*

3.3 Clock frequency, Power and Energy

Even when lowering the clock frequency reduces the power consumption, this strategy does not result useful in terms of energy [15]. In a synchronous system, if the frequency is diminished by a factor of two, the time required to finish the computation will be exactly two times greater. But if a static power component exists (like occurs in our board), its contribution to the total energy will be also duplicated.

This effect is illustrated in Fig.4, where the μP, ROM and RAM power components have been measured at 10 and 5MHz. For the μP (the most dynamic block), a power saving of the 46% (from 204 to 111 mW) exists. However, the RAM memory just reduces their consumption from 54 to 38 mW (30%). Finally, the ROM (that is disabled) maintains a constant consumption of 40 mW. The combined power reduction in the three elements is 37% (from 298 to 189 mW). But considering that in both cases the same computation is being performed (the *base* routine) the energy consumption involved to operate at 5 MHz result 1.3 times higher than the corresponding value to 10 MHz. That is, the peak power (and the microprocessor junction temperature) has been reduced, but the batteries will be discharged earlier.

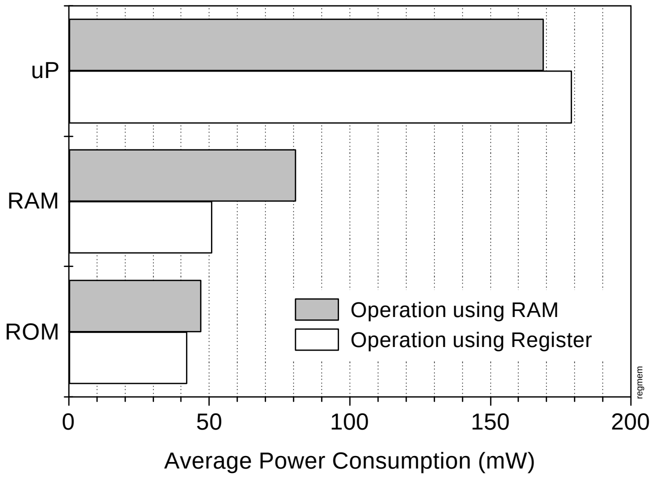

Fig.5: Power overhead of the usage of RAM to store intermediate results. Benchmark routines *SUMMEM* and *SUMREG.*

3.4 Avoiding memory operands

The extra power caused by the use of RAM as auxiliary registers has been also quantified. This problem was previously addressed in [18]. The benchmark routines *summemx* and *sumregx* (particularized to R1 and R2), were utilized to measure the µP, ROM, and RAM power components. In Fig.5 are depicted the main results. The µP power part result a 5,9 % lower (from 179 to 169 mW) when its internal registers are not utilized, but this reduction do not compensate the extra power in the RAM chips: from 50,1 to 80,7 mW. In addition, the increment of activity in the buses also contributes to a ROM power increment: from 42 to 47 mW. These numbers can also be analyzed in terms of energy, considering that external addressing increases the number of clock cycles required to perform the same computation. In this case, the information provided by the manufacturer [19] indicates that the operation in RAM requires 3 times more clock cycles than the equivalent computation using R1 and R2. As consequence, the corresponding energy required results 3.3 times greater.

3.5 Reducing energy by optimizing execution time

Previous discussion about the effect of a significant static power component suggests that, given a particular computation to be performed, the combination of instructions that reduce the execution time (the fastest program) will be a good candidate to require less energy (the frugal program). In addition, a system that finishes the computation task early can be set to power down mode earlier, a fact that has been pointed in [4]. An interesting consequence of these ideas is that, in absence of tools to minimize energy during the design cycle of a given electronic circuit, the technologists can reduce it indirectly, by increasing the bandwidth of the system [20], [21], taking advantage of the abundant tools available in any EDA suite.

In order to illustrate this strategy of optimization, the same computation was performed twice using different MA31750 instructions. The task consisted in moving a memory block of 1000 positions stored in RAM to another part of the memory. The results are depicted in Fig.6. The *move* routine exhibited a power consumption of 339 mW in µP and memories, meanwhile the power consumption for the *load/store* routine was 318 mW. Considering that the operations take 8202 and 40969 clock cycles respectively, the energy required for the fastest routine is near 5 times lower than the corresponding to the other routine.

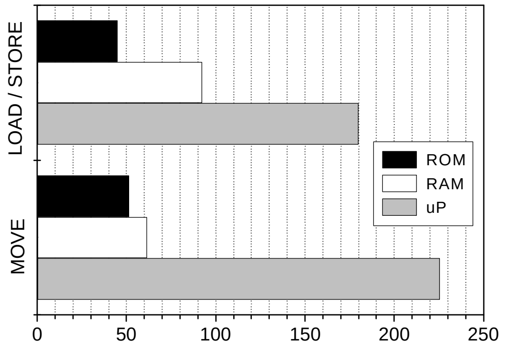

Fig.6: Power consumption of identical computations using two different programs. Routines: *move* and *load/store*. Program stored on RAM.

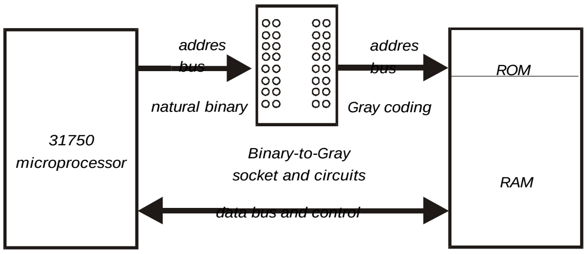

Fig.7: Circuit configuration to test the effect of bus coding.

3.6 Gray coding

Under the assumption of an address bus counts sequentially during large part of the program execution, the use of Gray coding to reduce off-chip activity has been pointed by several researchers [22], [23]. This technique is suitable to be incorporated during the implementation of a custom µP core, but its applicability to existing microprocessors is limited. An external Gray converter would be required, constituting a new source of dissipation. However, there is a chance of success if a positive balance occurs between: a) the memory power reduction (due to its lower activity at its inputs) and b) the extra power consumed by the Gray coder. For a given µP system, arriving to an analytical response to this question is difficult. However, the answer can be found, at reasonable cost, in the experimental field. The use of external coding to operate in Gray addressing was evaluated for our board. Several test points to insert the extra circuitry were included. In addition, a strategy to reorder the compiled instructions at the memory in a Gray fashion was developed.

In Fig.7 is depicted a scheme of the circuit. First, the standard circuit configuration was characterized, with the µP address bus connected directly to the memory. Then, all measurements were repeated with a Binary-to-Gray coder inserted between them. A bank of extra capacitance of 1 nF was optionally connected to the bus, to simulated a higher fanout produced by large RAM banks. Three hardware alternatives were tested for Gray coding: an EPROM-based look-up table, PALs devices, and finally discrete 74HC86 (quadruple 2-input XORs). The speed-grade selected for the devices was the maximum value that still satisfied the bus timing. The best result in terms of power corresponded to discrete gates. PALs (the fastest technological option) and EPROMs must be discarded for this application due to their intrinsic high consumption. The main characteristics of each alternative are summarized in Table 2.

	Static Power	Total Power @ 10MHz	Propagation delay
EPROMs 27HC256	322,5 mW	322 mW	55 ns
PALCE16V8H	357 mW	360 mW	15 ns
74HC86	1,9 mW	4.7 mW	24 ns

Table 2: Main characteristics of different Gray coding blocks. Benchmark routine *base*.

Main results were disappointing: just a 5% of reduction in the whole board for the case of heavily loaded bus (1nF per line), meanwhile no advantage was measured for standard bus loading conditions. In this case, the main drawback of Gray coding is that the input blocks of the memory (address decoding) can reduce its power due to its lower activity. However, at the output pads of the memory, the data are outputted in the same sequence in spite of natural of Gray addressing, because of the program has been previously rearranged before load it in memory. As result, no advantage can be expected of coding, if the following two facts occurs: the off-chip power of the memory is the main component of the chip, and the buses are slightly loaded.

4. CONCLUSIONS

In this paper LPD techniques have been tested in a MIL-STD-1750A specified microprocessor. Considering that this type of devices have a severe process of qualification, the main idea has been to demonstrate that some of the current techniques published in technical journals can be immediately adapted by the end-users. Several rule-of-thumb can be stated: a) devices that includes hundred of internal pull-ups (PALs, EPROMs) must be avoided; b) if possible, a copy of the program must be transferred to the RAM, and being executed from this devices; c) the overhead of using RAM as intermediate operands is high in terms of energy; d) a previous characterization of the µP registers can lead to an straightforward power reduction; e) in absence of tools and models to calculate energy in µP-based systems, the programmer (or the compiler) must minimize execution time.

5. REFERENCES

[1] J. Wall and A. Macdonald (Eds.), “The NASA ASIC Guide Assuring ASICS for Space”, Jet Propulsion Laboratory, 1993. An on-line version is available at *http://nppp.jpl.nasa.gov/asic/title.page.html*

[2] S. Gary, P. Ippolito, G. Gerosa, C. Dietz, J. Eno y H. Sanchez, "PowerPC 603, A Microprocessor for Portable Computers", *IEEE Design and Test of Computers*, pp.14-23. Winter, 1994.

[3] C. Su, C. Tsui y A. Despain, "Low Power Architecture Design and Compilation Techniques for High-Performance Processors", *Proc. IEEE 1994 Spring COMPCON*, pp.489-498. IEEE Press, 1994.

[4] C. Piguet, “Low-Power Microprocessors and Memories”, *Low-Power/Low-Voltage IC Design*, Santa Clara, April 1995.

[5] C. Piguet, et al, “Low-Power Design of 8-b Embedded CoolRisc Microcontroller Cores”, *IEEE J. of Solid-State Circuits*, Vol.31, Nº7, pp.1067-1078, July 1997.

[6] P. Drake y K. Burch, *"Portable Power, The Competitive Edge of the 68HC11"*, App. Note LP2/D, Motorola Inc. 1993.

[7] T. Burd and B. Peters, "A Power Analysis of a Microprocessor: A Study of an Implementation of the MIPS R3000 Architecture", ERL T. Report, UC Berkeley, 1994. Available at *http://infopad.eecs.berkeley.edu/~burd/gpp/gpp.html*

[8] V. Tiwari, S. Malik, A. Wolfe and M. Lee, "Instruction Level Power Analysis and Optimization of Software", *Journal of VLSI Signal Processing*, pp.1-18, Kluwer Academic Publishers, 1996.

[9] C. Turner, *"Calculation of TMS320C2XX Power Dissipation"*, Texas Instruments, 1996.

[10] A. Chandrakasan and R. Brodersen, "Minimizing Power Consumption in Digital CMOS Circuits", *Proc. of the IEEE*, Vol.83, N°4, pp.498-523. April 1995.

[11] S. Devadas y S. Malik, "A Survey of Optimization Technique Targeting Low Power VLSI Circuits*", Proc. 32nd DAC Conf*, pp.242-247, 1995. Available at: *www.ee.princeto.edu/~sharad/pub.html*

[12] M. Pedram, "Power minimization in IC design: principles and applications," *ACM Trans. on Design Automation of Electronic Systems*, Vol.1, No.1, pp.3-56, 1996. Available at *http://atrak.usc.edu/~massoud/*

[13] V. Tiwari et al., "Reducing Power in High-performance Microprocessors", *Proc. 35th DAC Conf.*, ACM 1998.

[14] M. Gowan, L. Biro and D. Jackson, "Power considerations in the Design of the Alpha 21264 Microprocessor", *Proc. 35th DAC Conf.*, ACM 1998.

[15] Burd and R. Brodersen, "Processor Design for Portable Systems", *Journal of VLSI Signal Processing*, 1996. Available at *http://infopad.eecs.berkeley.edu/infopad-ftp/*

[16] A. Abnous, K. Seno, Y. Ichikawa, M. Wan, and J. Rabaey, "Evaluation of a low-power reconfigurable DSP architecture", *Lecture Notes in Computer Science*, Vol.1388, pp.55-60. Springer, 1998.

[17] Cypress Semiconductors *"BiCMOS/CMOS Data Book"*, 1990.

[18] V. Tiwari, P. Ashar and S. Malik, "Compilation Techniques for Low Energy: An Overview", *Proc. Solid States Council Symposium on Low Power Electronics*, IEEE Press 1994.

[19] GEC Plessey, *"Hi-Rel IC and ASIC Handbook"*, 1995.

[20] E. Boemo, G. González de Rivera, S. López-Buedo and J. Meneses, "Some Notes on Power Management on FPGAs", *Lecture Notes in Computer Science*, Nº975, pp.149-157. Berlin: Springer 1995.

[21] P.Alfke, *"Personal Communication"*, 1995.

[22] C. Su, C. Tsui, A. Despain, "Saving Power in the Control Path of Embedded Processors", *IEEE Design & Test of Computers*, pp.24-29. Winter 1994.

[23] M. Stan, W. Burlesson, "Bus-Invert Coding for Low-Power I/O", *IEEE Trans. on VLSI Systems*, vol.3, nº1, pp.49-58. March 1995.