Switching-Off Partially the Register File to Reduce Power Using Zero Detection Lookup Table Pointers

Moisés A. Zarate Segura, Luis A. Villa Vargas, Marco A. Ramírez, Oscar Camacho

Center for Computing Research of the National Polytechnic Institute, México

e-mails: {mzarate, lvilla, mramirez, oscarc}@cic.ipn.mx
Phone: +52-55-57-29-6000, Ext.56519

Abstract

Register file of modern microprocessor is a critical component of high instructions-issue out of order processors. Conventional large monolithic and multi-ports register file consumes considerable power, access time and die area as this structures increase in size. Data compression, like zero detection, and interleaved Multi-banked register file are techniques proposed to reduce energy. However, both adds control structures which would impact and limit the access time to registers. This paper presents a novel scheme which as well, exploits the prevalence of zero bits stored in the register file. We propose a circuit level architecture that switching-off bit-lines precharge and sense-amp circuits, using a small lookup table which provides information about the register data bits distribution. We evaluated several configurations employing zero detection with different bit-string sizes. Simulations result show that we can reduce total register file energy for SPECint95 benchmarks by around 14%. The CMOS design and SPICE power evaluation for a multiported register file are presented using 90nm technology process.

1. Introduction

Register file (RF) is one of the key components that play an important roll in wide-issue processors. This structure is supported with heavily-ported physical register files to increase the instruction throughput required in this dynamically scheduled microprocessors. To having a large register file with many ports poses longer access time delay and accounts for a significant fraction of overall processor energy budget in modern microprocessors [1, 2].

A lot of work has been done in order to find the best access time for on-chip caches. Moreover, dynamic energy consumption has been the attractor in most of the ideas presented in those works. However, in this paper we approach our work in power consumption, taking into account the two main ways of energy dissipation, static and dynamic using 90nm technology with 1.2V supply.

In CMOS circuits two types of dissipated energy can be distinguish: static (leakage energy) and dynamic energy. The energy dissipation is directly proportional to the transistors count and capacitance switching activity [3]. Power is dissipated in each register file access. Read access starts precharging the bit-lines to half-Vdd. Bits stored into the cells are transmitted to bit-lines only when wordline signal is asserted, and one of the previously precharged bit-lines is pulled down switching one of the previous bit-line precharged, no whatever the bit value stored into de cell. After that, the voltage difference is detected by the differential sense-amplifier in order to amplify and regenerate as fast as possible the bit stored in the cell. Writing to the register file is also usually performed differentially but require a full voltage swing on one of each pair of bit-lines. This implies that, as in the caches, power is dissipated for each bit accessed (read and write) in the register file.

In our initial examination we found that over 75% on average of the bits that pass to or from the register file are zeros. This behaviour have been analysed and exploited in a cache context to reduce energy [6, 7, 9, and 11]. Processors like MIPS R10000 include a bit-flag inside the register file to indicate if the correspondent registers value is zero [4]. However, this bits asymmetry has not been studied with the same depth for register file.

In this work, we propose a technique to reduce bit-lines switching activity in order of reduce register file energy consumption. We introduce a novel technique for register file energy reduction, *zero detection lookup table pointers* (ZDTP), which exploits the prevalence of zero-bits stored in the register file. When data value is written into register file, an entry in ZDTP is indexed with the register ID and actualized with the information generated by the zero-detector circuit, *zero pointer bit* (ZPB). If the same register is accessed later, each

ZDTP entry point out which bits most be disable to prevent bit-line discharge during the register file access.

The rest of this paper is structured as follows. Section 2 presents simulation results for the bits distribution in our baseline register file. Section 3 presents an overview of the structure of a conventional register file –baseline- array and describes the necessary circuitry changes to implement the ZDTP technique. It also analyzes the energy consumption in the conventional register file. Section 4 shows the energy savings from using ZDTP on the register file. It also presents the breakdown energy consumption of the lookup table used by the ZDTP. Section 5 concludes the paper.

2. Bits Distribution in Register File Access

In this section, we present the distribution of bits values in register file accesses and show how to take advantage of this asymmetry distribution in order to maximize the energy savings. The performance of the proposed register file architecture has been evaluated using a modified version of SimpleScalar 3.0 [5]. We extended the Alpha 21264 model, to gather statistics on the presence of zero values in register file access. Eleven selected programs from the SPEC95 suit were used as benchmarks [5]. For our simulation we used the reference set inputs using the Alpha binaries.

We evaluated different bit-string granularities applying the zero detection technique in each configuration. Figure 1 and 2 (read and write respectively) shows the zero-bits detection in each register file access. For each case results are showed for 32-bits (*word*), 16-bits (*half word*), 8-bits (*byte*), and 4-bits (*half byte*) groups. We see that for all configurations the reduction was more than 40% on average. At this point we can make some observation: a) our 32-bits configuration reported over 30% of zero-detection. That is, if a processor avoid register access only when all 32-bits are zeros, as MIPS R10000 processor [4], only detect half of our best configuration; b) the more short the size is, the more zeros are detected; and c) our results show that *half byte* gives the greatest zero-detection overall. The *half byte* detection varies from 83% for apsi to close 49% for turb3d, with an average of 71% for reading and 60% for writing.

It is important to note that these numbers are our ideal baseline because we do not include additional bits (ZPB's) that can indicate whether the configuration contains all bits-zero. This is the upper-limit bit-lines swing reduction that zero-detection can prevent for our evaluated configurations.

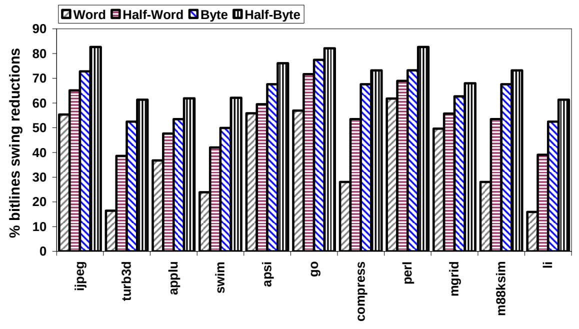

Figure 1: Reduction in the number of bit-line swings for read accesses when applying zero-detection to various sized bit fields.

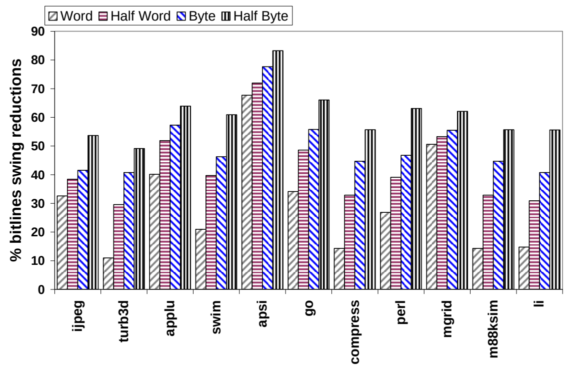

Figure 2: Reduction in the number of bit-line swings for write accesses when applying zero-detection to various sized bit fields.

3. ZDTP Circuit Design

Taking into account the enormous numbers of zeros detected during the register file accesses, the main conclusion from the previous section is that we can take advantage of this behavior to reduce energy in the register file. In this section we will describe the circuit design used to evaluate this hypothesis.

3.1 Reference Register File Approach

Figure 3 shows the baseline register file structured as eight-read and six-write ports, organized in four sub-banks of 32 entries of 32 bits each. We evaluated each read/write access to the register file in order of acquire power consumption information. We modelled the register file using the HSpice simulation tool for 90nm CMOS process with a nominal 1.2V supply.

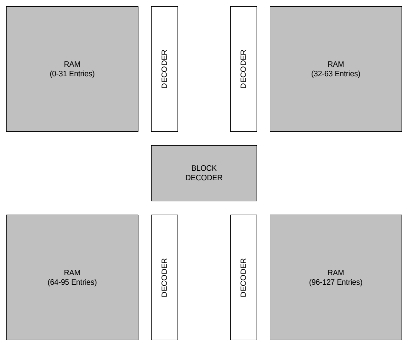

Figure 3: Architecture of 8-r and 6-w ports, organized in four sub-banks of 32x32-bit

On every register file access, only the corresponding sub-bank is enabled. However, the power consumption of sub-banks that are not used is important as we can see below. Bit-lines are precharged to %50 of full rail in read access to limit the voltage swing. On the other hand, the precharge-circuit used for write access keeps the bit-lines always precharged to full rail. It is important because bit-lines only swing when write-enable is active.

Table 1. Breakdown of power consumption for baseline register file sub-bank design.

#ent X bits	Write(mw)	Read (mw)	Idle (mw)
32x32	150	300	130

The power consumption for read and write access are showed in table 1. Moreover, we can see the static consumption (column: *idle*) for the disabled sub-banks that are not used in each register file access. To calculate the total power in the baseline register file, we proceeded as follow: the power consumption (read or write) in the accessed sub-bank plus the power consumption in the three idle sub-banks (one for each disabled sub-bank). From this table is important to note that even when the register file is not enabled, the power consumption is quite high, almost the same consumption than a write access.

3.1 ZDTP Description

In this section we present the architecture for our ZDTP register file scheme showed in the figure 4.

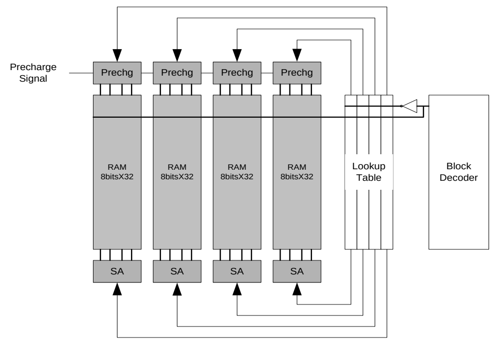

Figure 4: Structure of the register file sub-bank.

The primary modification to the baseline register file architecture is the inclusion of the *Lookup-Table* (LT), as the Figure 4 shows. This augmented structure is much the same to a small register file, with the same number of entries and ports. The size of each entry depends of the granularity used: {1, 2, 4, 8}-bits for *word*, *half-word*, *byte* and *half-byte* respectively. For writes to the register file (taking the *byte* configuration as example), we need to check when the data bits are all zero for each *byte*, writing “1” (the ZPB) in the LT for each zero-*byte* detected or “0” for bytes different to zero. Zero-bytes detected are not written in the Register File, only the bytes different to zero. For read access both the lookup-table and the register file are accessed using the registers ID. As the figure 5 shows, we used the CLK and the row signal, generated by the decoder, to activate both the lookup table and the register file wordlines.

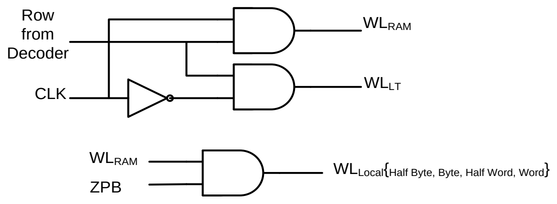

Figure 5: Circuitry generator of Lookup-Table wordline (WL_{LT}), RF global wordline (WL_{RAM}) and RF local wordline (WL_{Local}{ *half-byte*, *byte*, *half-word, word*}).

The timings of register file access signals are shown in Fig. 6. Each access is pipelined and can be performed each cycle. The operation sequence in the timing diagram is as follow: while the row signal is active (not depicted in the Fig. 6), during the first half of the CLK cycle the LT is accessed using WL_{LT} to read the ZPB-bits. In the second half of the CLK cycle the ZPB-bits are used to: a) enable local wordlines (WL_{Local}\{ *half-byte, byte, half-word, word*\}); b) if ZPB= WL_{Local}=0 the register file is accessed normally; c) if ZPB= WL_{Local}=1 the precharge-circuit, the local word-line and the sense amplifier circuit are disabled to avoid switching the bit-line for each: *half-byte*, *byte*, *half-word* or *word* respectively.

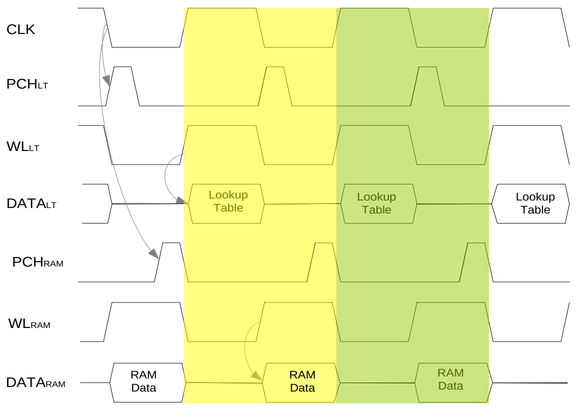

Figure 6: Timings in the ZDTP Register File access.

The lookup-table structure has an impact in both, the number of bits switching by access and global power consumption in the register file. The following sections explain the breakdown on power and bit-lines switching in the lookup-table.

4. Experimental Results

In this section we present the experimental results for the modeled ZDTP scheme: switching activity and power consumption.

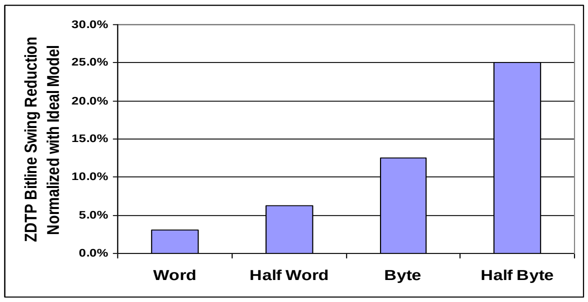

Figure 7: reduction of the global bit-lines switching results reported in section 2.

The Fig. 7 shows that the switching activity reduction reported, for read access, in figures 1 is affected due to the bits activity of the lookup-table. Write access have a similar pattern. That is, there is an important loss of the global bit-lines switching activity reduction reported in section 2. Our results shows that *half byte* gives the greatest loss (25%) followed by *byte* with 12.5%. However, the profit margin compared with a register file without zero detection is still high, over 30% on average for the eleven benchmarks evaluated.

The presence of the lookup table represents a high cost mainly for *half byte* configuration. In order to overlap the switching activity of the LT in each access (eight bits for *half byte*), at least two *half bytes*=0 most be detected. When any or only one *half byte*=0 is detected, it is like not avoid any bit-line switching due to the cost (switching activity) imposed by the lookup table. That is, when only one *half byte* is detected as zero, this *half byte* is not written in the register file, however eight bits most be written in the lookup table consuming the appropriate power consumption of the access. On the other hand, if at least two *half byte=0* are detected, the ZDTP writes eight bits to the look-up table and at the same time avoid writing eighteen bits in the register files. In this way, the eight bits written in the LT are overlapped or hidden with the eighteen bits that the ZDTP avoids to write in the register file. For *byte*, *half word* and *word*, detecting only one zero is enough to hide or overlap the bit-lines switching in the lookup table. This behaviour is showed in figure 8. The numbers of times that a maximum of two zeros was detected in each access for *half byte, byte, half word* or *word* are: (11.1%-88.8%), (23.2%-76.7%), (30.2%-69.7%), (56.5%-43.4%). Where (11.1%-88.8%) means: 11.1% on average were detected a maximum of two *half bytes=0* and 88.8% three or more *half bytes=0 were* detected. As we claimed before, the profit margin is still high to save energy in the register file.

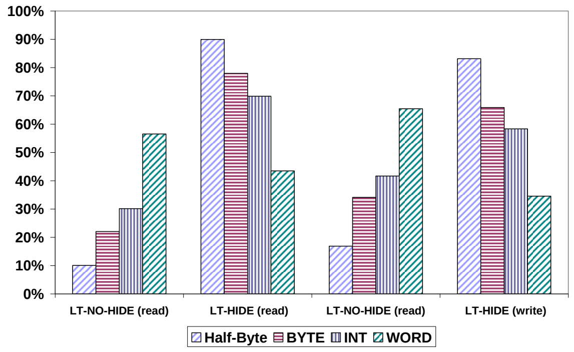

Figure 8: % of accesses to the look-up table that can or can not be overlapped with the zeros detected by access.

In order to quantify the power consumption for the lookup table, we modelled it using CMOS technology process of 90nm. Table 2 shows the breakdown of power consumption for entries of 8, 4, 2,1 bits lookup table design. Static consumption is again noteworthy. The power consumption when the LT is not accessed is quite similar to the power consumption of a write access. Moreover, it is approximately 1/6 of power consumption in write access in our baseline register file. The power consumption for write access of a *Byte* and *half byte* configuration is equivalent to an idle LT cycle. To calculate the total power consumption in the lookup table, we proceeded as follow: power consumption (read or write) in the active LT plus three times the idle power (one for each disabled LT). We conclude that the dynamic energy for write access is negligible in relation to the static consumption.

Table 2. Breakdown of energy consumption for LookUp Table Sub-bank.

Configuration	#ent X bits	Write (mw)	Read (mw)	Idle (mw)
word	32x1	4	10	3
half word	32x2	8	20	6
byte	32x4	15	37	15
half byte	32x8	35	72.5	35

4.1 ZDTP Power Consumption

Figure 7 presents the power savings for the register file using ZDTP schema. Even thought *half byte* and *byte* were configurations with more zeros detected, the lookup table power consumption is quite high, mainly the static consumption. Only *half word* and *word* were configurations where ZDTP save power, the 12% and 14% on average read/write accesses respectively, and 6.4% and 8.7% for *word* read/write access respectively.

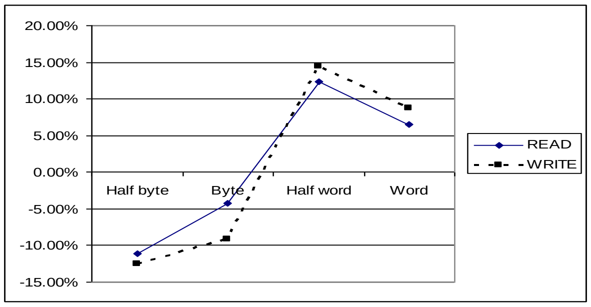

Figure 9: Register file energy reduction using the ZDTP scheme.

5. Related Work

Several other techniques have been proposed that also exploit the asymmetrical bits distribution to conduct studies to reduce energy consumption in cache memories and register files.

Villa, Zhang and Asanovic propose the dynamic zero compression to save cache energy. They avoid the whole read/write access to the cache, detecting dynamically zero-bytes during the cache access. They adds an additional bit indicator to each cache byte that indicates whether the byte contain all zero bits [6]. The same idea was tackled by Chang and Lai where they propose two transistor level dynamic zero-sensitive schemes to prevent the bit-line discharging in reading zeros from cache. In this technique they do not use a zero bits indicators [7]. In [8] J. Tseng and Asanovic showed how to exploit this bits asymmetry to reduce energy for memories with single-ended read bit-lines schema. Other related work about single-ended bit-lines for ROMs and small RAMs was proposed by Park and Chang [9], where the main idea is inverts stored words to reduce the total number of bit-lines discharges.

Kondo and Nakamura propose a Bit partitioned register file to reduce area, access time and save energy, based on the fact that many operands do not need the full register. Thus, they propose to use these use-less bits for others operands [10]. Reduce energy consumption in the register file and ALU is proposed by Kim [11]. When an operand for *add* instruction is zero, it is prevented to be executed on ALU because the result value is known in advance. A technique to reduce the static energy is proposed by Ubal, Sahuquillo, Petit and López [12]. Working at the cache word level, power supply is removed whether the store instruction's data is zero.

5. Conclusions

In this paper we proposed a zero detection lookup table pointers technique to reduce power consumption in the register file. This technique augments the traditional register file with a lookup table structure. The information of this lookup table is used to disable the precharge-circuit, the word-line and the sense amplifier circuit to avoid swinging the bit-lines for the configurations analyzed: *half-byte, byte, half-word or word*.

We have modelled the register file for a 90nm CMOS process. A simulations result shows that 16-bits size compression shows the higher power consumption reduction of 12% for read access and 14% for write on average.

We have observed that static consumption is a factor important in the whole energy consumption in our 90nm CMOS technology models. Our power consumption result shows that sub-banking like techniques are not enough to reduce energy consumption. This type of techniques most be complemented with others strategies in order to reduce static consumption.

At the present time we are working about the analysis of different CMOS models in order to find the balance between techniques like the proposed in this paper and the technology used to implement it.

10. References

[1] V. Zyuban and P. Kogge, "The Energy Complexity of Register Files", Proc. 1998 Int'l Symp. Low Power Electronics and Design (ISLPED), Aug. 1998, pp. 305-310,

[2] Farkas, Keith I. and Jouppi, Norman P. and Chow, Paul, "Register File Design Considerations in Dynamically Scheduled Processors", Technical Report 95/10, Digital Equipment Corporation Western Research Lab, November 1995.

[3] J. Rabaey, *Digital Integrated Circuits*, Prentice Hall, 1996.

[4] Kenneth C. Yeager, "The MIPS R10000 Superscalar Microprocessor", IEEE Micro Vol. 16, Issue 2, April 1996, pp. 28-40.

[5] T. Austin, E. Larson, and D. Ernst, "Simplescalar: An Infrastructure for Computer System Modeling", IEEE Computer, Vol. 35, No. 2, Feb. 2002, pp. 56-67.

[6] Luis Villa, Michael Zhang, Krste Asanovic, "Dynamic zero compression for cache energy reduction", MICRO, 2000, 214-220.

[7] Yen-Jen Chang, Feipei Lai, "Dynamic Zero-Sensitivity Scheme for Low-Power Cache Memories", IEEE Micro 25(4), 2005, pp. 20-32.

[8] J. Tseng and K. Asanovic, "Energy-efficient register access", In Proc. XIII Symposium on Integrated Circuits and Systems Design, Manaus, Brazil, September 2000.

[9] B.-I. Park Y.-S. Chang and C.-M. Kyung, "Conforming inverted data store for low power memory", In ISLPED, August 1999, pp. 91–93,

[10] Masaaki Kondo, Hiroshi Nakamura, "A Small, Fast and Low-Power Register File by Bit-Partitioning", IEEE Proceedings of the 11th International Symposium on High-Performance Computer Architecture, San Francisco USA, 2005, pp. 40-49.

[11] R. Ubal, J. Sahuquillo, S. Petit and P. López, "Applying the sero switch-off technique to reduce static energy in data cache", Proceedings of the 18th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD'06), Ouro Preto Brazil, October 2006, pp. 133-140.

[12] Kim, Soontae, "Reducing ALU and Register File Energy by Dynamic Zero Detection", IEEE International Performance, Computing, and Communications Conference, April 2007, New Orleans USA, pp. 1097-2641.