

# Zero Detect-Based Low Power Registers File Access

Moisés A. Zárate S. \*, Oscar Camacho Nieto \*,  
Osvaldo Espinosa Sosa \*

\* Digital Systems Laboratory, Computer Science  
Research Center, National Polytechnic Institute,  
Mexico City, MEXICO

Luis A. Villa Vargas

Applied Mathematics and Computation, Program of  
Research, Mexican Institute of the Oil, Mexico City,  
MEXICO

**Abstract** - With the intention of reduce significantly the energy that wastes away when having a read or write access to the register file, since the technique Zero Detect diminishes the transfers of chains of bits for each one of the accesses already mentioned. In this paper is presented an improvement in the reduction of the energy consumption for the register file of a superscalar processor with the technique of Zero Detect, for which the registration is divided in chains of 32, 16, 8 y 4 bit's (word, integer, byte and nibble). The realized experiments show that it is possible to diminish the consumption of potency up to 50%.

**Keywords:** Zero Detect, Registers file, Low Power.

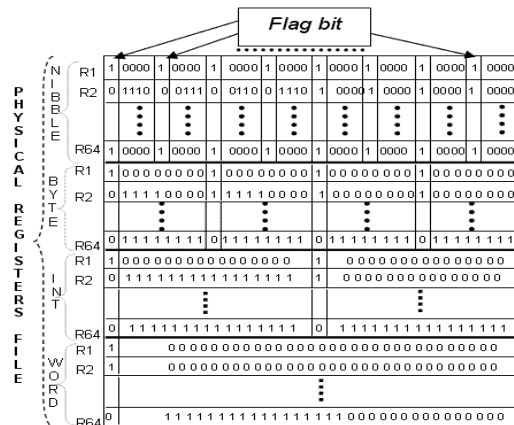
## 1 Introduction

As it is known, the register file is a fundamental component of the superscalar processors which consume a considerable amount of energy when processor needs to access to those files contents. For that reason, several techniques have been created in order to improve the performance increasing the hardware, but to cause of this, it has a higher energy consumption which is wanted to be diminished without affecting the [1]. One of the most important techniques on energy reduction is when having a line bit to indicate zero chains [2]. Others have sectioned those register files with multiple sub-banks in order to access every time only one bank which has an active bit line for each one of them [3]. Other techniques had a little transcendence such as separation of a register zero [2, 4, 5, y 6], which consists on detecting single registers that contain zero values within operations that are carried out inside the functional units, which is very significant, since when having an access to read the register, a lot of energy is wasted because bits are only read with zero values. It is also important to mention that others have proposed implementation of techniques that reduce multiple access that they are had when making a reading or writing to the register file [4, 7], achieving with it the energy reduction or the increment in performance. Techniques for detection of zeros in registers files are very important since thank to them good results have been obtained for the energy

reduction inside the superscalar processors and these techniques are susceptible of continuous improving.

## 2 Proposal

According to the previous needs, we created our own technique called *Zero Detect*, which is an extension of the techniques mentioned previously. It is important to mention that from the total of registers that are used for the operations inside the functional units; we can observe that 49% of them are similar to *zero*. As it is know, all registers that are used inside the architecture are physical, and they were born with zero value from renaming of the logical registers with zero register detections were made by each reading access and writing. They can meet groups of bits with zero value inside the registers with value different from zero, for this reason the fundamental part of this technique is the detection of chains of nibbles, bytes, integers and words equals to zero, for registers equal and different from zero. Once obtained these detections, a flag bit is placed by each one of them, see figure 1.



**Figure 1** It shows the *Zero Detect* technique applied to chains of nibbles, bytes, integers and words.

### 2.1 Methodology

To be able to know with detail the benefits that can be carried out inside a processor Alpha-21264 we use the Smplescalar V3.0 simulator. The modifications that were

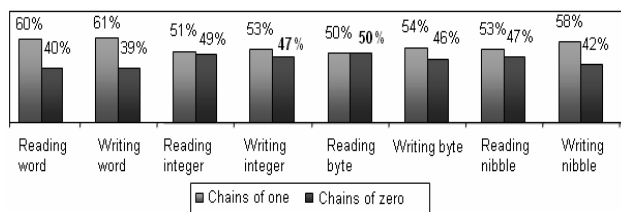
made inside the simulator to be able to know the stage where they used most of the registers, the detections of registers *zero*, the divisions of chains and the implementation of the technical *Zero Detect*, was proven by means of the work loads SPEC95.

### 3 Results

We can observe that if the value of the bits flags were similar to 1, the saving of readings of 32 bits was obtained (*words*), 16 bits (*integers*), 8 bits (*bytes*) and 4 bits (*nibbles*), but when the value of the bit was 0 they were had to enable the chains to be able to carry out the readings of them in a second cycle, that which resulted in the reading of 33 bits (*words*), 17 bits (*integers*), 9 bits (*bytes*) and 5 bits (*nibbles*). These indicates a lost one in the saving of readings and in the time of execution due to the second cycle, but even so you speculate that the *Zero Detect* presented an energy saving. The explanation is that at level alone transistor is wasted away energy of the activation of the 6 transistors of the flag bit and it was avoided that the 192 (*words*), 96 (*integers*), 48 (*bytes*) and 24 (*nibbles*) remaining transistors were activated.

#### 3.1 Evaluation of *Zero Detect* technique with the different divisions

As it was mentioned previously the *Zero Detect* techniques consists on adding a flag bit to all the chains *words*, *integers*, *bytes* and *nibbles*, for which, the results they generate are shown in the figure 2, alone they are of the activation of the flags bit. In the figure 2, it is observed that 40% of readings and 39% of writings were *words* chains similar to zero. It is also shown that 49% of readings and 47% of writings were *integer's* chains similar to zero.



**Figure 2** It shows the behavior of *Zero Detect* for the chains of nibble, bytes, integer and words with same and different value to zero in readings and writings.

For the case of bytes the figure 2 shows that 50% readings and 46% of writings were bytes chains similar to zeros. The results showed that you almost speculated 50% of energy reduction, that which is better in comparative with *word* and *int*. Lastly in the figure 2, it is shown that 47% of readings and 42% of writings were *nibbles* chains similar to zero, with these values one had an optimisation of a little but of 45% of energy saving, that which was smaller than the *integer* detections and *byte*, but even so it was good.

## 4 Conclusions

We concluded that *Zero Detect* model created with the simulator was good since it helps to optimize the physical register file with almost 50% of energy saving, what indicates at transistor level that only it wasted away energy when being activated the 6 transistors of the flag bit in all the columns belonging to each chain, despite of the penalization of 5% that we obtained in the performance. With the percentage obtained due to optimization of energy consumption that *Zero Detect* technique presents, it opens the possibility of the creation of new models that allow similar consumption savings but also without having the penalization of performance. We could obtain also much precise data with like TANNER o PSPICE in order to implement it in a Programmable Logic Device (PLD) and simulate it in power simulators.

## 5 References

- [1] Teresa Monreal Arnal, "Technical Hardware to Optimize the Use of the Registrations in Superscalar Processors Superscalar," PhD. Thesis of the University of Zaragoza, Department of Computer science and Engineering of Systems, June of the 2003.
- [2] R. Balasubramonian, S. Dwarkadas, and D. H. Albonesi, "Reducing the complexity of the register file in dynamic superscalar processors," MICRO-34, December 2001.
- [3] J.-L. Cruz, A. Gonzalez, M. Valero, and N. P. Tophan, "Multiple-banked register file architectures," ISCA-27, pages 316-325, June 2000.
- [4] A. Sez nec, E. Toullec, and O. Rochecouste. "Register write specialization register read specialization: A path to complexity-effective wide-issue superscalar processors," MICRO-35, Istanbul Turkey, November 2002.
- [5] G. Kane and J. Heinrich, "MIPS RISC Architecture (R2000/R3000)," Prentice Hall, 1992.
- [6] Luis Villa, Michael Zhang, and K. Asanovic, "Dynamic zero compression for cache energy reduction," MICRO-33, Monterey CA., Mexico, December 2000.
- [7] J. Tseng and k. Asanovic, "Energy-efficient register access," Proc. XIII Symposium on Integrated Circuits and System Design, Manaus, Brasil, September 2000.