

Power Estimation for Architectural Exploration of HW/SW Communication on System-Level Buses

William Fornaciari

Politecnico di Milano, DEI

P.zza L. Da Vinci, 32

20133 Milano (Italy)

Ph: +39-02-2399-3504

Fax: +39-02-2399-3411

fornacia@elet.polimi.it

Donatella Sciuto

Politecnico di Milano, DEI

P.zza L. Da Vinci, 32

20133 Milano (Italy)

Ph: +39-02-2399-3662

Fax: +39-02-2399-3411

sciuto@elet.polimi.it

Cristina Silvano

CEFRIEL

Via Fucini, 2

20133 Milano (Italy)

Ph: +39-02-23954-325

Fax: +39-02-23954-254

silvano@cefriel.it

ABSTRACT

The power consumption due to the HW/SW communication on system-level buses represents one of the major contributions to the overall power budget. A model to estimate the switching activity of the on-chip and off-chip buses at the system-level has been defined to evaluate the power dissipation and to compare the effectiveness of power optimization techniques. The paper aims at providing a framework for architectural exploration of a system design, focusing on the power consumption estimation of memory communication. Experimental results, conducted on bus streams generated by a real microprocessor and a stream generator, show how the variation of cache parameters and the introduction of bus encoding at the different levels on the memory hierarchy can affect the system power dissipation. Therefore, the proposed model can be effectively adopted to appropriately configure the memory hierarchy and the system bus architecture from the power standpoint.

1. INTRODUCTION

The increasing performance requirements of VLSI-based systems have pushed system architects to follow two main directions: increasing data bandwidth and adopting memory hierarchy. The first one consists of enlarging the data bandwidth to fill the gap between the speed of current microprocessors and that of the system interfaces. Consequently, both data and address buses have become very wide. However, due to the large intrinsic capacitances of the bus lines, a significant power is dissipated over the system buses. So, considerable power savings can be obtained by adopting encoding techniques to reduce the bus switching activity [1]. The second direction consists of hierarchically organizing the memory to trade-off speed and cost of memory accesses. The power cost of an external memory access is at least one order of magnitude higher than that of an on-processor access, due to capacitance overhead of I/O pads, on-board traces and input loads of DRAM components.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CODES '99 Rome Italy

Copyright ACM 1999 1-58113-132-1/99/05...\$5.00

In general, a multi-level cache implies a strong impact on the overall power budget due to the communication on heavily loaded buses. However, up to now, most cache studies are related to performance evaluation in terms of cache access time and miss rate with respect to the variation of cache parameters such as cache size, block size and associativity. Only recently, cache models have focused on power evaluation [11], [10]. Nevertheless, in our best knowledge, none of them considers the impact of caches and main memory on the power consumption due to the processor-memory interfaces in VLSI-based systems.

Aim of this paper is to evaluate the simultaneous effects of both bus encoding and memory hierarchy on the power dissipation of the system-level buses. In our model, the data and address buses are considered as the core of the communication between the processor and the memory and I/O sub-systems. In particular, they are used, on-processor, to access the higher level caches, as well as, off-processor, to access the external lower level caches, the main memory (possibly through a memory controller) and the I/O sub-systems to support the direct memory accesses from the I/O controllers.

The model can be helpful to define the most suitable cache configuration of each level in the hierarchy in terms of capacity, associativity and block size. To preserve the generality of the approach, we do not move down to the details of the internal architecture and the implementation technology of the cache memory. Thus, we do not include power figures related to the contribution of the memory arrays. However, the model can be easily combined with power data related to memory arrays [10].

The proposed power model consists of three main cooperating sub-models: the memory hierarchy, the bus encoder and the address/data stream generator. First, the model of the *memory hierarchy* enables us to consider multi-level unified or split caches offering several configurations in terms of cache size, block size, associativity, write strategy and block replacement policy. Second, the *bus encoding* model implements the most widely adopted power-oriented encoding schemes for both data and address buses. The model enables to insert the encoding schemes at the different levels of the memory hierarchy, and to evaluate their benefits. Third, the *stream generator* models typical address and data generated from the processor to the memory sub-system, taking into account spatial and temporal locality.

A simulation methodology has been set up to estimate the switching activity on system-level buses and to consider a wide range of processor-to-memory configurations. However, in this paper, we discuss the results conducted on two case studies: an embedded system based on the 32-bit low-power ARM7TDMI

processor and a high-end system including the 64-bit *PowerPC604e* processor. The analysis has been carried by using address and data streams generated by either a real processor or the stream generator. The latter case allows us to annotate the power effects of different spatio-temporal correlations in the bus streams. The simulation results show how the use of multi-level caches with variable parameters can be combined with bus encoding techniques to modify miss rates and power figures at the system-level.

The rest of this paper is structured as follows. Section 2 reviews the most recent works related to power-oriented cache modeling and bus encoding, while Section 3 describes the proposed system-level power model. In Section 4, we first describe the simulation methodology used to profile the behavior of system buses, then we report and discuss the simulation results obtained for the two case studies. Finally, Section 5 outlines the most significant contributions of the paper.

2. RELATED WORKS

Previous literature has been concentrated either on power estimation models for systems including memory hierarchies or on bus encodings for low-power consumption.

Su and Despain [10] proposed a model to evaluate the power/performance trade-offs in cache design and the effectiveness of novel cache design techniques targeted for low power. Among those techniques, vertical cache partitioning, and horizontal cache partitioning (i.e., cache sub-banking) have been investigated as well as Gray code addressing. The main limitation of this model consists of considering a single level cache whose architecture and implementation technology are fixed, so the reported results are strongly technology-dependent. Another model of energy consumption for the memory hierarchy has been provided in [14]. The adopted model is the same analytical model for *on-chip* caches proposed in [10], which has been generalized to include the main memory, so as to consider the amount of power required by cache misses.

In [11], the *Avalanche* framework is presented, to simultaneously evaluate the energy-performance trade-offs for *SW*, memory and *HW* for embedded systems. The energy estimation model is based on a *system-on-a-chip* architecture, and the power model includes the analytical cache model at the transistor-level, the DRAM main memory model and the *SW* model at the instruction level including the effects of caching.

Encoding paradigms for reducing the switching activity on the bus lines have been recently investigated ([4] [5], [6], [7], [9]). Most of them rely on the well-known *spatial locality* principle [12]. A comparative analysis of existing low-power bus encoding techniques, such as the *T0*, the bus-invert and mixed encoding is proposed in [7]. An additional contribution for special-purpose systems is provided in [8], while other encoding techniques at the system-level have been reviewed in [1] to discuss the introduction of redundancy in *space* (number of bus lines), *time* (number of cycles) and *voltage* (number of distinct voltage levels). Other approaches, such as [3], consist of directly changing the mapping of the information in memory, to decrease transition activity.

In summary, although some power models of caches and memories have been reported in literature, no one of them can be considered as a general model of the memory sub-system to estimate the power associated with the processor-to-memory communication in the system-level context. Our approach can be viewed as an attempt to fill such a gap. In particular, the original

contributions introduced by our model with respect to the previous ones are:

- It includes different levels in the memory hierarchy, such as on- and off-processor caches as well as the main memory;
- At each level in the hierarchy, it is quite general in terms of cache parameters, such as cache size, block size, associativity, etc.;
- It is independent of both the internal cache or memory organization and implementation technology;
- It considers the main low-power bus encoding techniques for both data and address buses;
- The bus encoder/decoder can be placed between the different levels of the memory hierarchy;
- Real bus tracings, derived from different application programs and executed by different processors, can be analyzed;
- Dedicated bus tracings can be generated to analyze the effects of different spatio-temporal correlations of the bus streams;
- The bus parameters can be varied in terms of width, frequency, and capacitive load.

3. THE SYSTEM-LEVEL POWER MODEL

The proposed model is composed of *three* main cooperating sub-modules: the *memory hierarchy*, the *bus encoder* and the *address/data stream generator*, which have been integrated in an object-oriented *SW* analysis tool written in *C++*. The models can be employed as basic blocks of different types of system architecture, ranging from dedicated system to general-purpose computer systems.

3.1 The Memory Hierarchy Model

The model consists of a multi-level memory hierarchy of on-processor and off-processor caches and the main memory. A generic cache level can be organized either as single *unified* cache or *split* cache for instructions and data. The cache model enables the designer to consider several configurations in terms of cache size, block size, associativity, write strategy and replacement policy [12]. The write strategy can be *write-through* or *write-back* and, in case of a write miss, either the *write-allocate* or the *no-write-allocate* option can be chosen. For set or fully associative caches, the block replacement policy can be *random* or *LRU* (*Least Recently Used*).

3.2 The Bus Encoder Model

To evaluate the bus encoding effects on power consumption, the bus encoder model can be inserted either on the interface from the processor to the first level of the memory hierarchy or between any adjacent levels of the hierarchy. The model implements the most diffused power-oriented bus encoding techniques, namely Gray, Bus-Invert, *T0*, *T0-BI*, *Dual_T0* and *Dual_T0-BI*. The encoding schemes can be applied to both data and address buses.

3.3 The Address and Data Stream Generator

Our intent is to simulate the system-level HW/SW communication by using address and data streams derived either by tracing a real processor (by using programs such as *pixie* for the *MIPS* processor) or by using a dedicated stream generator, which simulates the execution of a generic program on a microprocessor.

Obviously, the generator outputs are tightly dependent on the processor architecture. The current version of the generator models a generic load/store *RISC* architecture. In particular, to derive the first set of experimental results we refer to the instruction set of the *ARM7TDMI*, a 32-bit low-power processor

supplied by the *Advanced RISC Machines Ltd.* For our analysis, we considered a sub-set of the whole instruction set, which is composed of *three* basic classes of instructions: Conditional Branch Instructions (*B*); Arithmetic-Logic or Data Processing Instructions (*DP*); Load/Store or Data Transfer Instructions (*DT*).

In our model, we assume the memory address spaces for data and instructions to be *separated*. Basically, the sequence of memory addresses is generated by assigning the percentage of the different classes of instructions with respect to the total number of generated addresses. The address sequence is generated from the processor by varying: the format and the execution frequency for each instruction class; the possible addressing modes for each instruction and the related execution frequency; the frequency to satisfy a conditional branch. All these parameters contribute to modify the level of the spatial and temporal locality of memory references. The address bus from the processor to the memory sub-system contains a memory address corresponding to a *datum* or an *instruction*. The address stream characteristics can be assigned depending on the desired level of the spatial and temporal locality.

The bi-directional data bus can carry *two* different types of information: the *instructions* loaded from the memory to the processor and the *data* loaded from the memory or stored in memory. The type of instruction contained at a given memory address depends on the parameters set for the address bus model. Meanwhile, the datum contained in the memory address can be generated either probabilistically or pseudo-randomly. In the first case, the model is based on a medium average model of the first order, *MA(1)*, to take in consideration the correlation between two consecutive data words, responsible for the switching activity on the system-bus.

4. SIMULATION METHODOLOGY AND EXPERIMENTAL RESULTS

The models enable us to simulate many processor-to-memory configurations, from embedded systems to high-end general-purpose computer systems. In this section, we describe the simulation methodology used to estimate the power consumption of two case studies: an embedded system (including the 32-bit low-power processor *ARM7TDMI*, the L1 cache and the main memory) and a high-end computer system using the 64-bit *PowerPC604e* processor and the memory sub-system (including the L1 and L2 caches and the main memory). The case studies also differ for other system parameters, such as bus frequency, bus capacitive loads, and power supplies. The behavior of the different architecture configurations have been compared with the behavior of the reference architecture composed of the corresponding processor interfacing the main memory.

For our purposes, the estimated power figures reported in this chapter are intended as the average power dissipated due to the communication on the system-level buses, thus we disregarded the power contributions of hardware resources to the overall power budget. Previously published power models can be adopted to evaluate the power associated with the memory arrays and the processor, while for the encoding logic we can use the data reported in ([7], [13]).

The average power consumption on the system bus is given by:

$$P_{ave} = \frac{1}{2} C_{load} V_{dd}^2 n_{trans} f$$

where C_{load} is the load capacitance, V_{dd} is the power supply voltage, n_{trans} is the average number of transitions on the bus lines,

and f is the operating frequency of the system bus. Let us assume $B^{(t)}$ be the value of the encoded word on the system bus at time t , and L be the total length of the generated stream, the average number of transitions on the bus lines, n_{trans} can be given by:

$$n_{trans} = \frac{\sum_{t=0}^{L-1} H(B^{(t)}, B^{(t+1)})}{L-1}$$

where $H(B^{(t)}, B^{(t+1)})$ is the Hamming distance between the encoded word on the bus at time t and $(t+1)$. In the following, we report the simulation results related to the embedded systems, whereas the results derived for the high-end system are reported in [13].

The reference architecture for the embedded system consists of the 32-bit low-power processor *ARM7TDMI* running at 33 MHz and supplied at 3.3 V. The main memory interfaces the processor through 32-bit address and data buses operating at 66 MHz and loaded by 60 pF. During the simulation, we generated a *1K* instructions stream composed of 15% *B* instructions (20% *NSB*), 40% *DP* instructions and 45% *DT* instructions. The memory hierarchy includes a L1 *off*-processor cache with write through, no write allocate and random block substitution policies. The power profiling is based on the variation of cache features in terms of the cache size (4KB, 8KB, 16KB and 32KB), the block size (32-bit, 64-bit, 128-bit and 256-bit) and the associativity (one-two-four-eight way). Based on the reference architecture, four different case studies have been considered

CASE A aims at studying the effects of bus encoding on a system composed of the processor, the *on*-chip encoder and the main memory (Figure 1). The global bus power is due to both the processor-to-encoder and the encoder-to-memory buses. The power figures derived by varying the encoding schemes are reported in Table 1 for both addresses and data buses.

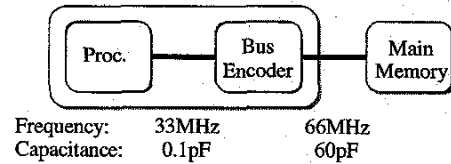


Figure 1. Architecture to evaluate the bus encoding effects (CASE A).

| Encoding | Address Bus Power [mW] | %Saved | Data Bus Power [mW] | %Saved |
|------------|------------------------|--------|---------------------|--------|
| Binary | 197.64 | Ref. | 330.53 | Ref. |
| BI | 190.88 | 3.42 | 293.37 | 11.24 |
| T0 | 185.95 | 5.91 | 330.8 | 0.08 |
| T0 BI | 180.96 | 8.44 | 296.11 | 10.41 |
| Gray | 102.38 | 48.20 | 339.37 | -2.67 |
| Dual T0 | 35.12 | 82.23 | 330.8 | -0.08 |
| Dual T0 BI | 25.32 | 88.20 | 323.74 | 2.05 |

Table 1. Address and data bus power for different encoding techniques. The binary encoding power corresponds to the power of the reference architecture.

For address buses, all the proposed techniques imply some power savings, while for the data side some advantages are provided by the *BI*-related methods. For the address bus, better results are obtained by those codes exploiting the spatial locality of addresses, such as the *Dual_T0* and *Dual_T0_BI* which imply 82.23% and 88.20% savings, respectively, since they can identify addresses related to data and instructions, whose memory spaces are separated in our model. The data bus transfers both instructions and loaded/stored data, so it does not ensure the same

level of locality as the address bus. For the data bus, the *BI* code, which limits to $N/2$ the number of transitions of N -bit buses, provides the highest saving (11.24%). A trade-off analysis taking into account also the encoder power consumption has been reported in [7].

CASE B aims at studying the effects of the off-chip first level cache, whose parameters vary from 4KB to 32KB for the cache size, 32-bit to 256-bit for the block size and associativity of one-two-four-eight way (see Figure 2).

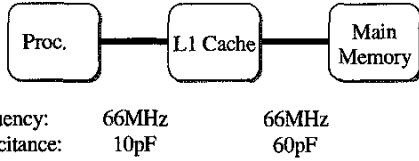


Figure 2. Architecture to evaluate the cache effects (CASE B).

First of all, we analyzed the miss rate trends by varying the cache size, the block size and the associativity. As expected, two basic behaviors have been derived by assuming the associativity as fixed. First, given the block size, the miss rate decreases as the cache size increases. Second, given the cache size, the miss rate decreases as the block size increases, in fact larger blocks take advantage of spatial locality [12].

Similarly, two basic trends have been derived, given a fixed block size. First, for the same cache size, the miss rate is reduced when passing from direct mapped to two- and four-way, while it is constant or slightly increases from four-way to eight-way. Second, given the associativity, the miss rate is reduced when the cache capacity grows.

In general, the advantage of increasing the associativity is that it decreases the miss rate. The improvement in miss rate comes from reducing misses that compete for the same location. Our miss rate trends are confirmed by the ones reported in [12]. The largest gains are obtained in going from direct-mapped to two-way set-associative, while the benefits of further associativity are not as big. Smaller caches obtain a significantly larger absolute benefit from associativity (especially for larger block sizes) because the base miss rate of a small cache is larger. As cache sizes grow, the relative improvement from associativity is constant or increases slightly; since the overall miss rate of a large cache is lower, however, the opportunity for improving the miss rate decreases and the absolute improvement in the miss rate from associativity shrinks significantly. The potential disadvantages of associativity are, in general, increased costs and slower access time.

The power trend behaves similarly to the miss rate because, by increasing the number of memory requests directly satisfied by the cache, the number of references to the main memory decreases. Consequently, a considerable reduction of the traffic occurs on the cache-to-memory bus, which has to switch larger capacitance (60 pF) than the processor-to-cache bus (10 pF). The corresponding power figures show a similar trend. Due to space limitation, only the diagrams of two-way set-associative caches are reported in the following (Figure 3 and Figure 4).

Note that a power reduction occurs for both address and data buses for larger cache sizes. A power reduction for larger block sizes can be observed only for the address bus, whilst for the data bus the power is almost invariant for any block size. As a matter of fact, for larger block sizes, the number of consecutive addresses loaded in cache increases and thus the average number of transitions (i.e. the power) of the address bus decreases for larger

block sizes. The data bus behavior is quite different, since the data value of consecutive memory locations are distributed randomly. So the power is almost the same for larger block sizes.

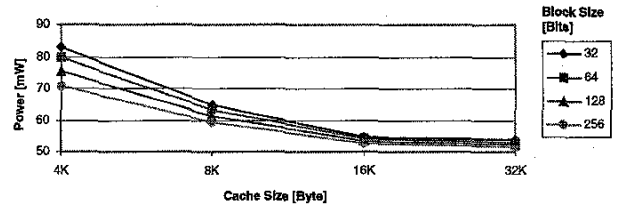


Figure 3. Power for address bus vs cache size for a two-ways set associative cache and four different block sizes.

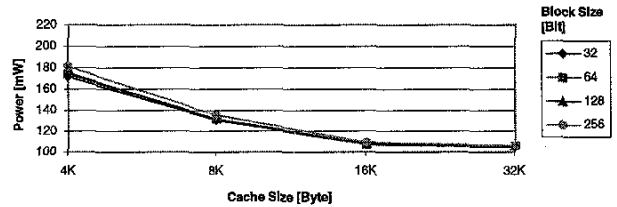


Figure 4. Power for data bus vs cache size for a two-ways set associative cache and four different block sizes.

We analyzed also the bus power consumption versus the associativity for four different cache and block sizes. As expected from the miss rate behavior, fixed the block and cache size, the largest energy savings are obtained from direct-mapped to two-way set associative caches, while further associativity can slightly improve energy consumption. As noted for the miss rate, this behavior is less noticeable for higher capacity caches. Given the block size and the associativity, the power decreases as the cache capacity increases, as expected by the miss rate behavior.

A comparison with the bus power dissipated by the reference architecture (197.64 mW and 330.53 mW for address and data bus respectively) has shown how the memory hierarchy implies performance advantages but also power savings. The average percentage reduction is approximately 68% for the address bus and 57% for the data bus. The reduction increases for larger cache sizes. Note that these results do not consider the internal power dissipation of the cache array, thus the effective reduction could be traded-off by the cache internal power.

For **CASE C** (see Figure 5), the bus encoder is implemented on-processor, whilst an off-processor L1 cache is provided, whose cache size varies from 4KB to 32KB, the associativity is one-two-four-eight way and the block size is 64-bit.

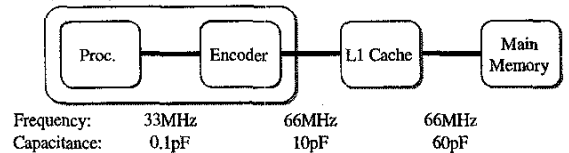


Figure 5. Architecture to evaluate the bus encoding and caching effects (CASE C).

The system-bus power is the sum of three bus contributions. The power versus the cache size for two-way set associative cache and several bus encoding techniques is reported in Figure 6 and Figure 7 for address and data bus, respectively. Concerning the address bus, the power dissipation is considerably reduced by adopting the Gray, *Dual_T0* and *Dual_T0_B1* schemes. The average percentage of power saved by several encodings with

respect to the reference architecture are reported in Table 2 for four cache sizes.

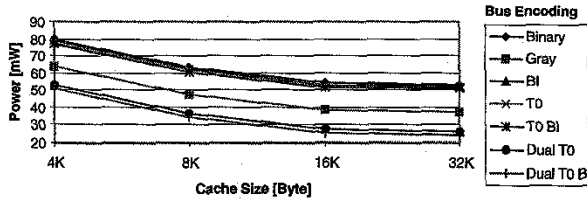


Figure 6. Power for address bus vs cache size for two-ways set associative cache and several bus encoding techniques.

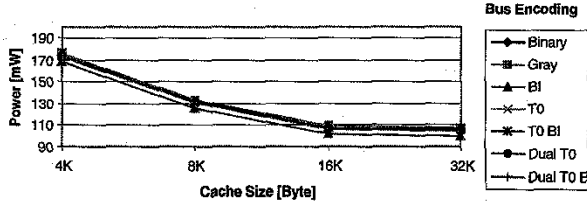


Figure 7. Power for data bus vs cache size for two-ways set associative cache and several bus encoding techniques.

| Cache Size | Binary | Gray | BI | T0 | T0_BI | Dual T0 | Dual T0_BI |
|------------|--------|-------|-------|-------|-------|---------|------------|
| 4KBytes | 58.78 | 66.74 | 59.28 | 59.70 | 60.12 | 72.42 | 73.41 |
| 8KBytes | 67.46 | 75.42 | 67.96 | 68.38 | 68.80 | 81.10 | 82.09 |
| 16KBytes | 72.17 | 80.13 | 72.67 | 73.08 | 73.50 | 85.80 | 86.80 |
| 32KBytes | 72.76 | 80.72 | 73.26 | 73.68 | 74.10 | 86.40 | 87.39 |

Table 2. Average percentage of power saved on address bus by using several bus encoding techniques with respect to the reference architecture for four cache sizes.

The Binary column corresponds to the architecture studied in the CASE B, where the block size is 64-bit. All the encodings, combined with the L1 cache, imply significant power benefits with respect to the reference architecture. For each encoding, larger savings can be obtained for larger caches. As before, for larger caches, the number of accesses to the main memory decreases and consequently the power related to the heavy loaded buses. Finally, Table 3 reports the average percentage of power saved by the adopted encodings with respect to the binary encoding (same as CASE B) for four cache sizes. Considerable advantages can be obtained by using both the *Dual_T0* and *Dual_T0_BI* codes.

| Cache Size | Gray | BI | T0 | T0_BI | Dual T0 | Dual T0_BI |
|------------|-------|------|------|-------|---------|------------|
| 4KBytes | 10.22 | 1.22 | 2.22 | 2.25 | 33.09 | 35.50 |
| 8KBytes | 24.48 | 1.54 | 2.82 | 4.11 | 41.92 | 44.98 |
| 16KBytes | 28.65 | 1.80 | 3.30 | 4.81 | 49.06 | 52.63 |
| 32KBytes | 29.25 | 1.84 | 3.36 | 4.91 | 50.08 | 53.74 |

Table 3. Average percentage of power saved on address bus by using several bus encoding techniques with respect to the binary encoding for four cache sizes.

Concerning the data bus, the power saving of the encoding schemes with respect to the binary code (CASE B) is very limited (it reaches up 4% for *BI*). As before, all the reported results do not consider the internal power of the cache and encoder blocks.

The analysis of CASE D, which aims at evaluating the effects of off-processor cache followed by off-processor bus encoder, is similar to those carried out for the CASE C. The address bus results confirm that encoding techniques exploiting the sequentiality can be effectively adopted for address buses, while

for data buses, due to the low correlation existing among data related to consecutive time steps, the binary encoding is preferred since the power savings of other encodings are not noticeable. Conversely from the address trends reported for the CASE C, the address trends show how the power savings due to the introduction of bus encoding techniques are very limited for large cache sizes. In fact, for large caches, the number of accesses on the cache-to-memory bus, where the encoder is placed, is very low.

More details and diagrams for all the presented cases as well as for the high-end system can be found in [13].

5. CONCLUSIONS AND FUTURE WORKS

Up to now, the study of bus encoding techniques has disregarded the presence of memory hierarchy along the processor-to-memory communication path. Aim of this work has been to evaluate the effects on power of bus encoding schemes in the presence of multi-level cache memories. Current effort is devoted to analyze high-end general purpose systems targeted for PowerPC architecture, where the presence of Virtual Memory as well as a finer grain model of the memory arrays contribution have to be taken into account. As future perspective, this work offers the basis to define novel bus encoding methods taking into account the effects of memory hierarchy on global communication.

6. REFERENCES

- [1] M. R. Stan and W.P. Burleson, "Low-Power Encodings for Global Communication in CMOS VLSI," *IEEE Trans. on VLSI Systems*, Vol. 5, No. 4, Dec. 1997, pp. 444-455.
- [2] S. Ramprasad, N.R. Shanbhag, I.N. Hajj, "Signal Coding for Low Power: Fundamental limits and Practical Realizations," *ISCAS98: 1998 Int. Symp. on Circuits and Systems*, Monterey, CA 1998.
- [3] P. R. Panda, N. D. Dutt, "Reducing Address Bus Transitions for Low Power Memory Mapping," *EDTC-96: IEEE European Design and Test Conference*, pp. 63-67, Paris, France, March 1996.
- [4] M. R. Stan, W. P. Burleson, "Bus-Invert Coding for Low-Power I/O," *IEEE Trans. on VLSI Systems*, Vol. 3, No.1, pp.49-58, Mar. 1995.
- [5] C. L. Su, C. Y. Tsui, A. M. Despain, "Saving Power in the Control Path of Embedded Processors," *IEEE Design and Test of Computers*, Vol. 11, No. 4, pp. 24-30, Winter 1994.
- [6] H. Mehta, R. M. Owens, M. J. Irwin, "Some Issues in Gray Code Addressing," *GLS-VLSI-96: IEEE 6th Great Lakes Symposium on VLSI*, pp. 178-180, Ames, IA, Mar. 1996.
- [7] L. Benini, G. De Micheli, E. Macii, D. Sciuto, C. Silvano, "Address Bus Encoding Techniques for System-Level Power Optimization," *DATE 98: IEEE Design Automation and Test in Europe, Paris 1998*.
- [8] L. Benini, G. De Micheli, E. Macii, M. Poncino, S. Quer, "Power Optimization of Core-Based Systems by Address Bus Encoding," *IEEE Trans. on VLSI Systems*, Vol. 6, No. 4, Dec. 98, pp. 554-562.
- [9] E. Musoll, T. Lang, J. Cortadella, "Working-Zone Encoding for Reducing the Energy in Microprocessor Address Buses," *IEEE Trans. on VLSI Systems*, Vol. 6, No.4, Dec. 98, pp.568-572.
- [10] C.L. Su and A.M. Despain, "Cache Design Trade-offs for Power and Performance Optimization: A Case Study," *ISLPED95, Int. Symp. on Low Power Design*, Monterey, CA 1995.
- [11] Y. Li and J. Henkel, "A Framework for Estimating and Minimizing Energy Dissipation of Embedded HW/SW Systems," *1998 ACM/IEEE Design Automation Conference*, Jun. 98.
- [12] J. L. Hennessy and D.A. Patterson, *Computer Architecture: A Quantitative Approach*, 2nd Ed., Morgan Kaufmann, 1996
- [13] C.Silvano, "Power Estimation and Optimization Methodologies for Digital Circuits And Systems", Ph. D. Thesis, University of Brescia, Italy, Dec. 1998.
- [14] P. Hicks, M. Walnock, R.M. Owens, "Analysis of Power Consumption in Memory Hierarchies", *ISLPED-97: 1997 Int. Symp. on Low Power Elect. and Design*, Monterey, CA, Aug. 97, pp. 239-242.