Methodology for propagating technology trade-offs over memory modules to the application level

A. Papanikolaou M. Miranda F. Catthoor† H. Corporaal‡ H. De Man†
D. De Roest M. Stucchi and Karen Maex†
IMEC, Kapeldreef 75, Leuven, Belgium
†Also Professor at the Katholieke Universiteit Leuven, Belgium
‡Also Professor at the Technical University of Eindhoven, The Netherlands
{papaniko,miranda,catthoor,heco,deman,deroest,stucchi,maex}@imec.be

ABSTRACT

In this paper we show how to exploit energy-delay trade-offs that exist due to the different options for the technology parameters for the implementation of interconnect wires. We also evaluate how these trade-offs can be propagated to the memory module level, so we can minimise the power consumption of the entire memory organisation. Our approach is that at future technology nodes the delay problem can be handled at the application level, so given any delay slack obtained there, we can exploit it to make the interconnect wires slower and thus less energy consuming. We have shown that for real-life applications the power consumption can be reduced by about 34%, when compared to the option provided by the ITRS roadmap, while meeting the real-time constraints.

1. INTRODUCTION

As feature sizes scale down the wires become ever more dominant in delay and power compared to the logic [1, 2]. The reason is that they become smaller, (increased resistance, hence delay), and closer to each other (increased capacitance).

The delay problem can be handled on the application level so that, given any delay slack we can create, we can exploit it to make the switching on the interconnect wires slower and thus, as we will show, less energy consuming.

In this work we evaluate how technology trade-offs between delay and energy that can be offered at the level of technology parameters can be propagated and exploited at the level of application design. We use a three-step approach to investigate how the technology parameter trade-offs can affect the power consumption of a complete design. The first step (see Section 2) is to describe the effects that create these trade-offs at the physical level. The second step (Section 3) is to examine how they can affect the behaviour of a component at the level of IP block. In the last step (Section 4) we show how the trade-offs obtained at the module level can be used to further minimise the power consumption of the entire memory organisation for a specific application.

Many multimedia systems nowadays are heavily data-dominated [3, 4] and the on-chip memory organisation of these systems is becoming the bottleneck in power consumption. For this reason we have decided to use an SRAM memory as an IP block case study, where we investigate if the energy-delay trade-off of the technology parameters can be propagated to give a trade-off between memory delay and memory energy consumption per access. To accomplish this, we have built a wire-based model for small and medium sized embedded SRAMs.

Using a data transfer and storage optimised version of a Digital Audio Broadcast (DAB) channel decoder as a driver, we show significant gains in the memory organisation power consumption only by optimising the dimensions of the interconnect wires inside the memories. The slack in delay created by optimally mapping the application to the target architecture has allowed to propagate the full range available for energy reduction from the interconnect up to the system level. Hence, the power consumption of this application has been reduced by about 34%, if instead of the very fast wires, that are currently proposed by the ITRS roadmap, we can selectively use slower and more energy-efficient interconnect wires.

To alleviate the forementioned problem at the system/ application level, several groups are currently working on techniques to reduce the energy consumption in the memory subsystem. In groups cooperating in Torino/Bologna [7] and also at Univ. Irvine in California [8] effort is being invested on memory design issues and compilation techniques both at a relatively low abstraction level. Our Data Transfer and Storage Exploration methodology [4] is clearly a forerunner in this field in terms of orthogonality, abstraction level, and portability of the approach which results in much wider gains at the application level. However, there also most of the possible solutions have been fully exploited. So on the longer term, the bottleneck should be broken also by other means with emphasis this time on the interconnect contribution and its exploration when combined with system or application level optimisations techniques, as we show in this paper.

2. TECHNOLOGY LEVEL TRADE-OFFS

The scaling down of the interconnect technology, as dictated by the ITRS roadmap [9], implies a shrinking of the local interconnect lines in all three dimensions, i.e. length, width and spacing (or pitch) and height by a factor $s < 1$ [9]. The shrinking increases the total capacitance C of a line and keeps the RC product of the same line almost constant. However, by varying some geometric parameter of the interconnect lines inside a fixed technology node, a possibility exists to obtain a trade-off between RC delay and capacitance C. In one instance, this is done by keeping the pitch and the height of the wires constant and changing only the

width of the lines. In this case, it can be shown that the linear dependence between RC and C is removed: both R and C change, because if the line gets smaller in cross-section the spacing increases. This case has been explored on a typical worst-case interconnect 2D cross-section model for the 45nm technology node. Values of resistance and capacitance per unit length have been extracted from the model by using line resistivity values and dielectric constant values dictated by the ITRS roadmap. In particular, the capacitance has been extracted by the commercial static solver Raphael.

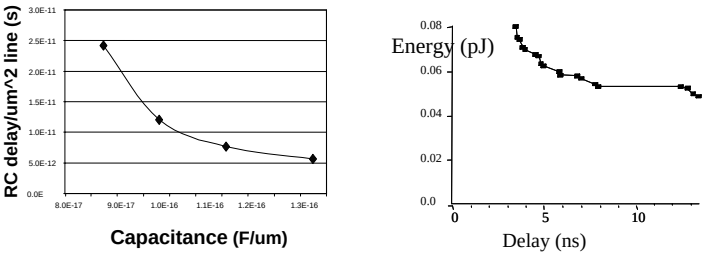

Figure 1: Left: Delay vs. capacitance trade-off in local interconnect wires, Right: Pareto exploration of a 64kbit memory instance, partitioning and interconnect parameters range

The trade-offs are shown in Figure 1(left) for the local interconnect. It is clear that making the same changes in the other types of interconnect (intermediate and global) will yield similar trade-offs for those types. This behaviour should be exploited to obtain really good designs, as will be shown. Furthermore, the ranges in delay and capacitance are good. For capacitance, which corresponds directly to energy consumption, the range is about 34%, which means that system level energy consumption can be affected to a satisfactory degree. The subsequent range in delay, however, is almost a factor 4. So, roughly, an approximately 34% extra gain in system energy consumption can be achieved by relaxing the timing contastraints on the wires by a factor 4. We will show that such a delay slack can be created at the application level, by meeting the throughput constraints of the system via data-parallelissation techniques [10]. Provided that the range in capacitance can propagate to the application level design, we can really affect performance and energy consumption at the system level.

3. PROPAGATION OF TECHNOLOGY TRADE-OFFS TO IP BLOCK LEVEL

To be able to propagate these reported ranges to the application level, we need, first, to build an IP block level model and evaluate the impact of the technology trade-offs at this level. In order to choose what kind of module to model, we have to take into account the application domain we are focusing on, which is that of embedded multi-media and communication applications, in general data-dominated ones. The main characteristic of these applications is the large amount of data that has to be stored and processed. As a result, we have chosen to use a model of an embedded SRAM as a representative IP block for our purposes. Furthermore, memories include a large number of long interconnect wires, such as bit-lines and word-lines, making the impact of technology trade-offs more visible than on functional units. The main advantage, however, is that memories are regular structures with controllable floor-plans and they do not present any 'random' place & route or timing closure problems. This results in fewer modelling difficulties and better accuracy, compared to functional units.

3.1 SRAM model

To model an embedded SRAM we have started from the CACTI model [5]. This is a complete energy/delay/area model for embedded caches.

The main advantage of CACTI is the fact that it is scalable to different technology nodes. To achieve this scalability the results (delay and energy consumption) are scaled linearly with feature size. This scaling method is not very accurate, especially for delay, but gives an indication for the energy and delay trends of the memory for smaller feature sizes.

The main shortcomings of this model include the outdated circuits and the very old technology parameters. It was built considering the 0.8 um technology node. The result is that to accurately model a future embedded SRAM apart from the projected technology parameters, one should also change the templates of the circuits. The circuit structures that were used for this node, will not be usable for the deep submicron technology nodes, due to the different challenges that have come up, i.e. leakage and static power, importance of interconnect wires etc.. But, lacking a better memory model we have used this one for our experiments.

Transforming CACTI into a model for an embedded memory is a straightforward task, by only taking into account the data part of the cache and not the tag part. But, for the purposes of this work we need a model for an embedded SRAM at the 90, 65 and 45nm technology nodes. Down to the 45nm node we have good models for the interconnect wires from in-house simulations, but no good circuit and transistor models exist yet below 90nm.

However, by experimenting with CACTI for old technology nodes and from the results reported by recent publications [6, 11, 12] we see that the contributions of the main components in energy and delay are balanced. We expect this trend to continue, memory designs should continue being well balanced. Following this trend and taking into account the increasing importance of wire delay and energy consumption we have assumed that logic components contribute a constant percentage of total delay and energy consumption.

3.2 Memory module level exploration

On top of this model and the exploration of possible partitioning schemes that is included in CACTI we have added an exploration of the dimensions of the interconnect wires. The exploration of partitioning options heavily reduces both memory delay and energy per access, when compared to a monolithic cell array memory. This exploration can, however, provide a small energy-delay trade-off.

Coupling the trade-off at the technology level to the memory model we can see its large influence on the entire memory at the level of IP block. In combination with the limited energy-delay trade-off because of memory partitioning, we can now have very good ranges in the energy-delay Pareto optimal trade-off curve of the entire memory, see Figure 1(right).

The final output of this model is now an energy-delay-area Pareto optimal trade-off curve which shows all the optimal

feasible operating points of the particular memory, see Figure 1(right) (only energy-delay Pareto points are shown). Thus the designer has the freedom to choose the memory which just satisfies the application delay constraints and has the least possible energy consumption per access.

It is important to note here that not all these Pareto points are necessary in order to get system level gains. What is more important is the available range rather than the number of points.

4. PROPAGATION OF IP BLOCK LEVEL TRADE-OFFS TO APPLICATION LEVEL

So far we have been dealing with the wires that exist inside the memories of the memory organization. Apart form these, wires are also used for the implementation of the buses. But, the contribution of these wires can be kept low by using two optimization techniques. The first is an activity-aware power optimal floorplaning. The idea is to place the heavily active memories close to the datapaths, so that they have short conenctions. Less active memories can be placed farther away. The second technique is bus segmentation, which partitions the bus into several segments and only the necessary segments are activated for each memory transfer. Combining these two techniques makes sure that the inter-memory interconnect power consumption stays relatively small and we can focus only in the intra-memory power consumption. Experiments show that the power consumed on the buses can be kept under 20% of the total power of the memory organization (memories and buses), if the application has been already optimized for data transfer and storage, using e.g. DTSE.

In order to assess the impact of the energy vs. delay trade-offs on the application level we have to apply the IP block level trade-offs on an actual application. The driver application we have used is a Digital Audio Broadcast (DAB) channel decoder which has been optimised for data transfer and storage management using the DTSE methodology [4]. After optimisations, the clock frequency required to implement the application while meeting the real-time constraints is only 43 MHz.

These optimisations help to relax the timing constraints on the individual memories significantly, thus creating an opportunity to trade off delay slack for minimum energy consumption. Thus, global system-wide trade-offs can be made which allow that the optimal memory organisation is affected by the use of the trade-off space in the memory selection process.

In order to see how the energy-delay trade-offs propagate to the application level we have performed an experiment using only the ITRS, thus fastest, option for the interconnect wires both inside the memories.

The goal of the experiment is to find the optimal number of memories that should be used in the memory organisation to minimise the total memory organisation power consumption. We take into account the number of memories, the power consumption for each memory and the access frequency of each memory. The result is the optimal memory organization and the total power dissipated on it.

A second experiment has involved using the slowest possible, energy optimal, interconnect wires which meet the real-time constraint of the application. This means that the wires inside the memories will be customised to the delay requirements of each memory. Memories on the critical path will have faster wires than the ones that have relaxed timing constraints. The results are shown in Figure 2.

By comparing the results of these two experiments we can see that for the best memory allocation and assignment case (10 1st layer memories), the selective use of slow, but power-efficient, wires can save up to 34% in power consumption of the memory organization compared to a design based on the ITRS roadmap proposed points.

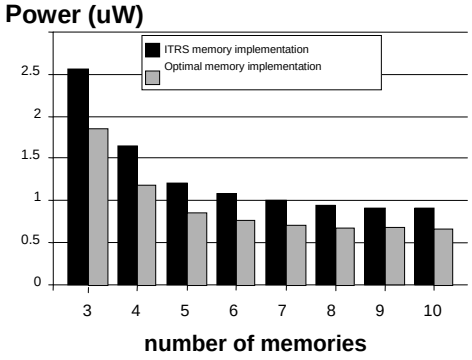

Figure 2: System power consumption using the ITRS and the power-optimal interconnect options

5. CONCLUSIONS

In this paper we show how to exploit energy-delay trade-offs that exist due to the variation of the technology parameters during the implementation of interconnect wires and how to propagate these to the system/application level via the memory module level. In this way, we have shown that for current data dominated applications, the power consumption at future technology nodes can be reduced by about 34%.

6. REFERENCES

[1] D. Sylvester, K. Keutzer, *Impact of small process geometries on microarchitectures in systems on a chip*, Proceedings of the IEEE, vol.89, no.4, p. 467, April 2001.

[2] J.A. Davis, R. Venkatesan, A. Kaloyeros, M. Beylansky, S.J. Shouri, K. Banerjee, K.C. Saraswat, A. Rahman, R. Reif, J.D. Meindl, *Interconnect limits on gigascale intergration (GSI) in the 21st century*, Proceedings of the IEEE, no.3, vol.89, pp. 305, March 2001.

[3] R.Gonzales, M.Horowitz, *Energy dissipation in general-purpose microprocessors* IEEE Journal of Solid-state Circ., Vol.SC-31, No.9, pp.1277-1283, Sep. 1996.

[4] F. Catthoor, S. Wuytack, E. De Greef, F. Balasa, L. Nachtergaele, A. Vandecappelle *Custom memory management methodology exploration of memory organization for embedded multimedia system design* Kluwer, June 1998.

[5] Steven J. E. Wilton and Norman P. Jouppi "CACTI: An Enhanced Cache Access and Cycle Time Model" *IEEE Journal of Solid-State Circuits*, Vol.32, No.5, pp.677-687, May 1996.

[6] Teruo Seki, Eisaki Itoh, Chiaki Furukawa, Isamu Maeno, Tadashi Ozawa, Hiroyuki Sano and Noriyuki Suzuki, *A 6-ns 1-Mb CMOS SRAM with Latched Sense Amplifier*, IEEE Journal of Solid-State Circuits, no.4, vol.28, pp. 478, April 1993.

[7] A.Macii, L.Benini, M.Poncino, "Memory Design Techniques for Low Energy Embedded Systems", ISBN 0-7923-7690-0, Kluwer Acad. Publ., Boston, 2002.

[8] P.R.Panda, N.D.Dutt, A.Nicolau, "Memory issues in embedded in systems-on-chip: optimization and exploration", Kluwer Acad. Publ., Boston, 1999.

[9] International Technology Roadmap for Semiconductors 2001, http://public.itrs.net.

[10] E.Brockmeyer, A.Vandecappelle, F.Catthoor, *Systematic cycle budget versus system power trade-off: a new perspective on system exploration of real-time data-dominated applications* Proc. IEEE Int. Symp. on Low Power Electronics and Design, pages 137-142, Rapallo, Italy, Aug. 2000.

[11] Motomu Ukita, Shuji Murakami, Tadato Yamagata, Hirotada Kuriyama, Yasumasa Nishimura and Kenji Anami, *A Single-Bit-Line Cross-Point Cell Activation (SCPA) Architecture for Ultra-Low-Power SRAM's*, IEEE Journal of Solid-State Circuits, no.11, vol.28, pp. 1114, November 1993.

[12] B. Amrutur, M.A. Horowitz, *Speed and power scaling of SRAM's* IEEE Transactions on Solid State Circuits, vol. 35, no. 2, p. 175, February 2000