Block-Enabled Memory Macros: Design Space Exploration and Application-Specific Tuning

Luca Benini[#] Alessandro Ivaldi[*] Alberto Macii[*] Enrico Macii[*]

[*] Politecnico di Torino
Torino, ITALY

[#] Università di Bologna
Bologna, ITALY

Abstract

In this paper, we propose a combined solution that allows us to customize the architecture of internally partitioned SRAM macros according to the given application to be executed. Energy savings with respect to monolithic memory configurations are above 40%, without access time violation.

1 Introduction

One of the most successful techniques for memory energy optimization is *partitioning*, which is based on the idea of accessing only a small number of memory cells at a time, in an effort to reduce switching activity in the heavily loaded bit lines and word lines of large SRAM macros.

In the past, research has clustered around two approaches. In the first (called henceforth *coarse-grain partitioning*), many smaller memory macros are instantiated instead of a single large memory and additional control circuitry is synthesized to activate one macro at a time [1, 2]. In the second approach (*block-enabled* memories), the internal circuitry of memory macros is modified to enable mutually exclusive activation of sub-blocks in the cell array [3, 4]. The second approach has the obvious advantages of minimizing duplication of decoding and output multiplexing logic and reducing control overhead. It has been proposed as a general-purpose technique for implementing energy efficient memories, without any specific consideration on memory access patterns. On the other hand, coarse-grain partitioning techniques have demonstrated the effectiveness of tailoring the number and size of the partitions to a known access trace.

In this work, we combine the advantages of both approaches. We propose a block-enabled memory organization which can be optimally tuned to a known access pattern. We develop analytical energy consumption and performance models for these memories. Based on these models, we formulate and solve the problem of optimally choosing partition granularity and boundaries for a given address trace, when the optimization objective is minimum energy under access time constraints. Results demonstrate how application-specific block-enabled memories outperform their general-purpose counterparts and how this solution is superior to coarse-grain partitioning thanks to its drastically reduced overhead.

2 Application-Specific Block-Enabled Memories

In this section, we introduce a new memory architecture that combines the advantages of coarse-grain partitioning to those of [3] and [4], called in the sequel DBL and DWL respectively, to achieve high energy efficiency at no performance (i.e., access time) penalty: The Application-Specific Block-Enabled (ASBE) memory.

The distinguishing feature of the ASBE memory architecture is that its organization and synthesis are driven by the knowledge of the memory access trace specific to the application that is supposed to run on the system.

The reference architecture we assume for building ASBE memories is shown in Figure 1.

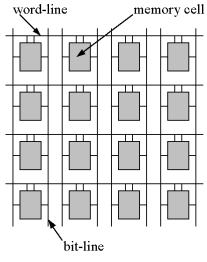

Figure 1: Basic Memory Architecture.

Given this memory configuration, and without changing the aspect ratio of the memory (i.e., the number of bit-lines and word-lines is fixed), we split horizontally and vertically the memory in different, possibly non-uniformly sized blocks, using the DBL and DWL techniques.

The splitting process is driven by the knowledge of the memory trace in such a way that the memory sub-blocks are fit around the addresses that are mostly required, following the principle described in [2]. In other words, the most common addresses are mapped onto memory sub-blocks that are small and with small bit-line and word-line capacitances; therefore, accessing those addresses is advantageous from the energy point of view.

The partitioning algorithm requires accurate models for the evaluation of the access time and of the energy consumption of the ASBE memory architecture.

The models are built assuming that the boundary logic of the memory array (e.g., sense amplifiers, pre-charge logic, multiplexers, bit-lines and word-lines selection logic, ...) is untouched after the partitioning process is performed.

Moving from the models used in Cacti [5], the delay for the bit-line can be computed as:

$$t_{bl} = (r_1 \cdot c_1 + (r_1 + r_2) \cdot c_2) \cdot log(\frac{V_{bitpre}}{V_{bitpre} - V_{bitsense}})$$

where V_{bitpre} is the pre-charge voltage and $V_{bitsense}$ is the sense voltage. The values for r_1, r_2, c_1 and c_2 are given by: $r_1 = R_{pass} + K \cdot R_{bitline}$, $r_2 = R_{pass} + WL \cdot R_{bitline}$,

1530-1591/04 $20.00 (c) 2004 IEEE

$c_1 = K \cdot (C_{bl} + C_{drain})$, $c_2 = WL \cdot C_{bl} + N_{part} \cdot C_{drain}$, where R_{pass} and C_{drain} are the on-resistance and the drain capacitance of a pass transistor that divides the main bit-line from the local one, respectively; K is the number of word-lines that compose the partition we are accessing; N_{part} is the number of partitions; $R_{bitline}$ and C_{bl} are the resistance and the capacitance of a portion of the bit-line equal to the height of a single memory cell, respectively; finally, WL is the number of word-lines. Delay t_{bl} may be different for each partition; therefore, we have to consider the maximum value for the whole memory.

The energy consumption during a memory access is due to the charging of the bit-lines and of the word-lines. The total capacitance of the bit-line is:

$$C_{tot-bl} = C_{bl} \cdot WL + N_{part} \cdot C_{drain} + (C_{bl} + C_{drain})K \quad (1)$$

Notice that, increasing the size of the partitions makes the capacitance and the energy increasing.

Given a memory access trace, to compute the total energy we have to multiply every access by the corresponding energy cost. Grouping the access trace based on the partitions, we can calculate the energy as:

$$E_{total} = \sum_{i=1}^{N_{part}} E_i A_i$$

where E_i and A_i are the energy and the number of accesses for partition i, respectively.

The number of partitions impacts the energy demand and the data access time. This implies that, in order to effectively generate the ASBE architecture that best suits a given input trace, we have to determine the optimal number of partitions N_{part} required by the application of DBL. N_{part} is computed by solving the derivative of the bit-line capacitance over the number of partitions. More specifically, if the memory is partitioned into blocks of the same size, the number of word-lines combined together is given by: $K = \frac{WL}{N_{part}}$

Hence, Equation (1) becomes:

$$C_{bl} \cdot WL + N_{part} \cdot C_{drain} + (C_{bl} + C_{drain}) \frac{WL}{N_{part}}$$

and the corresponding derivative is:

$$C_{drain} - (C_{drain} + C_{bl}) \cdot \frac{WL}{N_{part}{}^2}$$

By zeroing the derivative, we can calculate the optimal value for the number of partitions as:

$$N_{part-opt} = \sqrt{WL \cdot (1 + \frac{C_{bl}}{C_{drain}})}$$

and then the number of word-line grouped together as:

$$K_{opt} = \frac{WL}{N_{part-opt}}$$

3 Experimental Results

To validate the applicability of the ASBE memory architecture, we have considered some C benchmarks taken from the Mediabench suite. We have assumed an ARM-based platform; thus, each program was fed to ARMulator that takes the C code and the memory specification parameters and generates the dynamic memory access trace, which is provided as input to the modified version of the MemArt partitioning engine of [2], which incorporates the energy and access time models described before.

Characterization of the models was done by extracting the parasitic parameters from memory macros referring to a 0.18μ CMOS technology provided by STMicroelectronics.

Different strategies of memory partitioning have been applied, in order to perform a comprehensive comparison of the ASBE approach to state-of-the-art solutions. In particular, we have considered the MemArt coarse-grain partitioned memories, the combined DBL+DWL block-enabled memories and the ASBE memories.

Figure 2 shows energy and access time savings for each architecture with respect to the monolithic implementation.

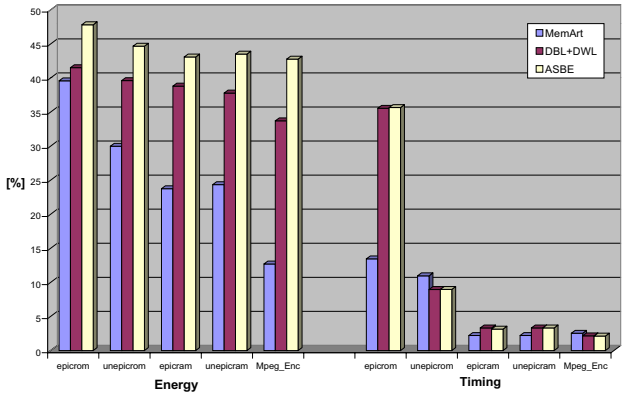

Figure 2: Results.

From these results, it is evident that the ASBE solution outperforms all the others from the energy point of view. Energy savings are, in all cases, higher than those provided by MemArt and DBL+DWL by a significant amount (e.g., almost 10% for the Mpeg_Enc benchmark).

We observe also that, for the ASBE memories, the improvement in energy is always paired with a reduction of the access time; such a reduction is very similar to that obtained using DBL+DWL, while in a few cases it is smaller than what can be achieved using the MemArt coarse-grain partitioned architecture.

4 Conclusions

In this paper, we have proposed a new approach to memory partitioning, that combines the concepts of internal partitioning (that is, divided bit-lines and word-lines) to that of external, profile-driven partitioning and that guarantees high energy savings at no performance penalties.

References

[1] S. L. Coumeri, D. E. Thomas, "An Environment for Exploring Low-Power Memory Configurations in System-Level Design," *ICCD-99*, pp. 348-353, Austin, TX, October 1999.

[2] L. Benini, L. Macchiarulo, A. Macii, M. Poncino, "Layout-Driven Memory Synthesis for Embedded Systems-on-Chip," *IEEE TVLSI*, Vol. 10, No. 2, pp. 96-105, April 2002.

[3] A. Karandikar, K. K. Parhii, "Low Power SRAM Design using Hierarchical Divided Bit-Line Approach," *ICCD-98*, pp. 82-88, Austin, TX, October 1998.

[4] M. Yoshimoto *et al.* "A Divided Word-Line Structure in the Static RAM and Its Application to a 64K Full CMOS RAM," *IEEE JSSC*, Vol. 18, No. 5, pp. 479-485, October 1983.

[5] P. Shivakumar, N. P. Jouppi, *CACTI 3.0: An Integrated Cache Timing, Power and Area Model*, WRL Research Report 2001/2, Compaq Western Research Labs, Palo Alto, CA, December 2001.