Issues in Embedded DRAM Development and Applications

D. Keitel-Schulz

Siemens AG
Semiconductor Group
Balanstraße 73
D-81617 München, Germany
e-mail: doris.keitel-schulz@siemens-scg.com

N. Wehn

University of Kaiserslautern
Institute of Microelectronic Systems
Erwin-Schrödinger-Straße
D-67663 Kaiserslautern, Germany
e-mail: wehn@e-technik.uni-kl.de

Abstract

After being niche markets for several years, application markets for one-chip integration of large DRAMs and logic circuits are growing very rapidly as the transition to 0.25 µm technologies will offer customers up to 128 Mbit of embedded DRAM and 500 Kgates logic. However, embedded DRAM implies many technical challenges to be solved. In this paper we will address some of these technical issues in more detail.

1 Introduction

The transition to 0.25 μm or even smaller technologies allows the integration of up to 128 Mbit of embedded DRAM and 500 Kgates logic on the same piece of silicon. Figure 1 shows memory and logic complexities for various die sizes (excluding pads) in an advanced 0.25 μm embedded DRAM process. This possibility makes embedded DRAM[1] technology (eDRAM) very attractive for real "system-on-silicon" implementations [1, 2, 3, 4, 5]. Hence the market for eDRAM, estimated at 100–200 M in 1997, is projected to reach more than 4 billion in 2000. Additionaly eDRAM offers DRAM vendors the possibility to escape the actual DRAM prize desaster and to set up eDRAM IP.

The possibility to integrate large memory and logic on the same die has a large impact on system integration and performance, memory sizes, on-chip memory interfaces and memory structures. Main advantages of embedded DRAMs are higher memory bandwidth, lower power consumption, customized memory sizes and higher system integration. But embedded DRAM burdens disadvantages and/or challenges on technology and fabrication, testing and design methodologies.

With embedded DRAM, the system designer faces a new design parameter which enlarges tremendeously the architectural design space. He/she is no more restricted to the use of commodity DRAMs which imply standard sizes, interfaces and access protocols. The capacity of DRAMs increases by a factor of four for every new generation. As the growth of bandwidth requirements has kept pace with those of the memory, the interface width of DRAMs should thus have been growing as fast as the size of single DRAM devices. This has not happened for packaging reasons. As a consequence, in many applications the use of commodity DRAMs lacks of memory granularity and/or bandwidth.

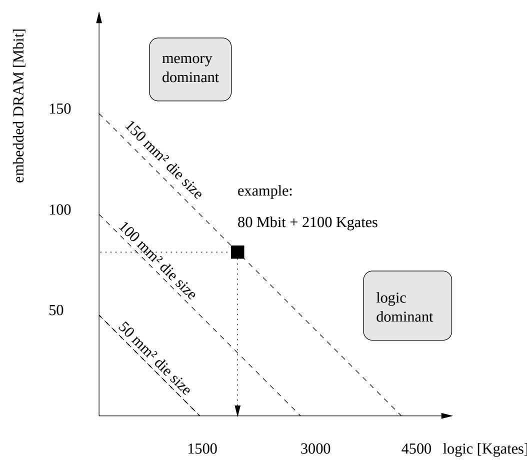

Figure 1: Trade-off logic and DRAM complexity for various die sizes

2 Comparison of embedded DRAM versus external DRAM

Most important with embedded DRAM is that the designer can adjust the bandwidth and memory size to its application. Let us consider a system which needs a total amount of C_{system} memory bits and T_{system} memory bandwidth. C_{device} and T_{device} denote the memory size and mem-

[1] We speak of "embedded DRAM" or "embedded logic" depending on whether the master process is a logic or a memory process. Note that some authors use the terms in exactly the opposite way.

ory bandwidth of a single commodity DRAM, respectively. There are two cases in which memory is wasted when the memory system is composed of commodity devices:

Case 1: the granularity of the memory devices forces more memory. This is the case if

$$\left\lceil \frac{C_{system}}{C_{device}} \right\rceil \times C_{device} > C_{system}.$$

E. g. the size of PC memory systems has grown by only half the rate of single DRAM devices (DRAM growth: 60%/year, Windows NT software growth: 33%/year).

Case 2: the memory bandwidth forces parallel access to several memory devices. Unnecessary memory is induced if

$$\left\lceil \frac{T_{system}}{T_{device}} \right\rceil \times C_{device} > C_{system} \Rightarrow \frac{T_{system}}{C_{system}} > \frac{T_{device}}{C_{device}}$$

Thus the wasted memory for a given system C_{system}, T_{system} which has to be composed of memory devices C_{device}, T_{device} is:

$$max\{\left\lceil \frac{C_{system}}{C_{device}} \right\rceil, \left\lceil \frac{T_{system}}{T_{device}} \right\rceil\} \times C_{device} - C_{system}$$

The ratio $\frac{T_{system}}{C_{system}}$ characterizes an application requirement. A low ratio means that the application demands relatively small bandwidth compared to its large memory sizes (e. g. workstation applications), a high ratio means that a large bandwidth must be provided by a small amount of memory (e. g. 3D graphics applications). This ratio is called the *fill frequency* [6] which gives the number of times per second a given memory can be completely filled with new data. It is important to notice that in the past $\frac{T_{system}}{C_{system}}$ over the different applications has been roughly constant or has increased, while the DRAM device fill frequencies $\frac{T_{device}}{C_{device}}$ have declined steadily [6, 7]. The consequence (see case 2 above) is unwanted memory, especially in applications where $\frac{T_{system}}{C_{system}}$ increases (e. g. graphic applications).

Let's have a more detailed look on the memory bandwidth which can be calculated as $T_{device} = IO_{width} \times f_{IO}$. IO_{width} is the width of the memory device and f_{IO} the data IO frequency. Due to the page-miss penalty of DRAMs f_{IO} is not a constant value. The access time to a dataword in another page differs by one order of magnitude compared to the access time to a dataword in the same page. Thus f_{IO} can vary by one order of magnitude and the sustainable f_{IO} is very application dependent. To maximize this value on the memory level, eDRAM offers the possibility to adapt the page size to the application, to integrate cache lines directly into the eDRAM macro, to apply multibank structures [8, 9] or to use an access-sequence control scheme as proposed in [10].

A second factor which influences f_{IO} is the load capacitance which has to be driven by the memory buffers. Obviously lowering this load increases f_{IO}. Typically there is a difference of a factor of 10–50 between on- and off-chip driver loads. In addition, inductivity caused by the package and the board lines is eliminated if the DRAM/logic connection is done on-chip, thus system noise immunity is enhanced. However, the most important factor which influences the memory bandwidth is IO_{width}. In commodity devices the IO width is limited to 16–32 pins due to packaging reasons. Embedded DRAM provides buswidths up to 512 bit or even more. Since the memory interface is on-chip, the total pin count of the chip is reduced and pad-limited designs may be transformed into non-pad limited ones.

Obviously eDRAM can offer a finer granularity in memory sizes (steps of 256 Kbit or 1 Mbit) and a much higher bandwidth range than commodity DRAMs. Thus the fill frequency of an embedded DRAM module can be tuned towards the fill frequency of the application. Figure 2 illustrates this advantage. Peak bandwidths and fill frequencies of commodity DRAMs (EDO, SDRAM, SGRAM, DDR, Rambus) and embedded DRAM cores are depicted in a logarithmic scale for 16 Mb and 64 Mb, respectively.

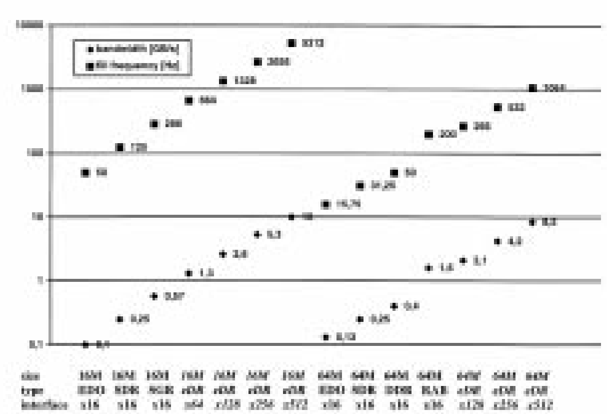

Figure 2: Bandwidth and fill frequency

Low power is another important issue which can be positively influenced by eDRAM. A DRAM core is always optimized for low power. Hence power can be mainly optimized by minimizing IO power or deactivating idle memory banks. In [11] it is reported that the power consumption of off-chip data transfers, compared to a typical 16-bit arithmetic operation like an addition, is about 33-fold. Especially in multimedia applications which are excellent candidates for eDRAM applications memory transfer operations occur very frequently. The IO power consumption of a DRAM is given by $V_{DD} \times V_{swing} \times C_{load} \times f_{IO}^{av}$. f_{IO}^{av} is the average switching frequency of the IO memory bus which has to be charged/discharged to V_{swing}. As stated in a preceding paragraph, an internal load is 10–50 times smaller than an external load. Hence, assuming the same interface (e. g. LVTTL), IO power consumption decreases with the same factor. Reducing the power supply of the logic is the most efficient way to save power in the logic part of the chip [12]. However, reducing V_{dd} degrades the frequency and with it the memory bandwidth. To compensate for this degradation, the logic has to be parallelized implying a widening of the memory interface to meet T_{system}. A wide interface is one of the inherent advantages of eDRAM. Moreover V_{swing} is reduced, yielding a further power reduction. An excellent overview on techniques to minimize power on the system level with respect to memory and data transfers is given in [11].

In [13] it is reported that merging a microprocessor with DRAM can reduce the latency by a factor of 5–10, increase the bandwidth by a factor of 50 to 100 and improve the energy efficiency by a factor of 2 to 4. Other examples of embedded processors with embedded DRAM are given in [14, 15, 16].

But as explained later embedded DRAM comes not for free. Big challenges are imposed on technology, fabrication, testing and design methodology.

3 Applications of embedded DRAM

The current move to eDRAM is mainly driven by two factors: first portable devices which demand for low power, second the processor-performance and fill frequency gap. Due to the complexity of the advantages, disadvantages and challenges of eDRAM, it is not possible to give a simple formula for the advisability of eDRAM in a specific project. However, some rules of thumb can be given:

- The product volume and product lifetime are usually high.
- Either the memory content is high enough to justify the higher DRAM process costs, or eDRAM is required for bandwidth, low power consumption or other reasons.
- Other things being equal, eDRAM will find its way first into portable applications.

Embedded DRAM has already occupied a large part of the market for 3D graphics accelerator chips for laptops; in this segment, the advantages of lower power consumption and higher performance cannot be ignored. Embedded DRAM is also slated to conquer a large part of the desktop PC and games market for graphics chips in the next few years. Memory sizes of 32–64 Mbit are likely to be required, mainly for frame storage.

Other main markets for eDRAM are networking and embedded CPU applications (hard-disk drives, printers, etc.). Network switching is the high-end market for eDRAM: memory sizes of up to 128 Mbit and interface widths up to 512 are required for reading and writing data packets out of large buffers. As switches are not consumer products, the volume is relatively small, but the prize premium is high. Embedded CPU-applications are driven mainly by system cost; the products contain embedded processors, and the memory is used for storage of programs as well as data. Memory requirements are more modest than for graphics controllers, both in terms of size and bandwidth.

Several other markets are possible for eDRAM, including mobile phones, personal digital assistants (PDAs), etc. However, it is unlikely that eDRAM will capture the PC market for main memory, as the need for flexibility and an upgrade path is too strong.

A final aspect is that several business models are common in the eDRAM sector, from foundry business to ASIC-like business.

4 Technological Challenges

One of the biggest challenges in eDRAM is the increased process complexity. The memory density based on a 1-T cell in a DRAM technology is about one order of magnitude denser than a 3-T dynamic cell in an advanced logic process. This density advantage requires dedicated technology steps not found in logic technologies.

An optimum process should offer the most advanced DRAM cell arrays and high performance MOSFETs with high density multi-level interconnect at reasonable costs. However, requirements on a high performance logic process (large I_{dsat}, low V_{th}, salicidation, borderless contacts, n+/p+ gate doping, large number of interconnect layers etc.) contradict to the requirements on a high density, cost efficient DRAM process.

In principle, there are two completely different approaches to tackle these issues: starting with a DRAM process as the basic process, enhance as much as possible the MOSFET transistors, improve routing pitches and add additional metal layers. Alternatively, start with a logic process as the basic process and add DRAM capabilities. Both approaches have their specific application advantages and disadvantages (see table 1).

High dense DRAM cells can be implemented either by trench or stacked based concepts. Trench based cells (see figure 3) are preferable for embedded DRAM technologies since:

- There is no real mix of relevant process steps. The ca-

	DRAM based	Logic based
Performance	10–25% less than optimized logic performance	close to optimized logic performance
DRAM density	comparable to commodity DRAM devices	up to a factor of 2 less
DRAM performance	commodity DRAM performance	minor DRAM performance
Logic densities	lower than ASIC technology	comparable to ASIC technology
Manufacturing	yield learning from commodity product	limited yield learning from older DRAM generation or with dedicated mass product
Evolution	at least one shrink/year with 30% size reduction	about 2–3 years lifetime until next generation available
Wafer cost	DRAM process + MOSFET impr. + interconnect	significant adder depending on DRAM concept

Table 1: Advantages/disadvantages of different technology approaches

pacitance is completed before gate oxydation. Thus the process is a low temperature process after gate oxydation.

- The storage cell has a larger capacitance than a stacked based cell.
- A trench process provides a flat topology which simplifies the augmentation of additional interconnect layers.

The technology selection should be done according to the following rules of thumb. A DRAM based technology should be prefered as base technology if:

- a large portion of DRAM and medium to high performance in the logic part,
- additional analog features, low power operations,
- short product cycles and high volume.

are requested by the product. A logic based process should be prefered if small DRAM sizes, highest performance in the logic part and typical ASIC product cycles are requested by the product.

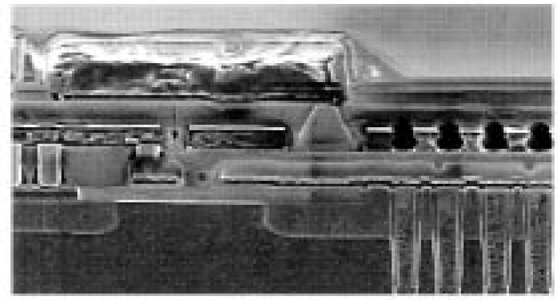

Figure 3: Trench memory cell

Another important aspect is the used design methodology and the turn-around time in the development cycle of the product. DRAM design methodology is usually transistor and thus bottom up oriented. A DRAM is a handcrafted, highly yield and area optimized IC with less flexibility. The fabs are tuned for yield and throughput, but logic fabs are tuned for short processing time. Customized logic is developed with synthesis based top down design methodologies. Complex cells like DSP cores or memories are included as macrocells which have to satisfy rigorous design guidelines. Thus when developing an eDRAM process, the logic library impact has to be minimized. Merging both design methodologies and fab philosophies is a further challenge for the successful application of eDRAM.

5 Testing Methodology

A test methodology is essential for the successful use of eDRAM macros. Testing DRAMs is very different from testing logic. In commodity DRAMs testing can account for up to 50% of the vendor's production costs. DRAM tests not only determine the speed and check the correct functionality, but also ensure the correct storage under worst case conditions. Wafer and package burn-in procedures are necessary to discover partially damaged devices. Since DRAM includes redundancy, two wafer-level tests are necessary. Thus DRAM test is a complex process consisting of several steps based on highly specialized test equipment and algorithmic test patterns. A DRAM test is 10 to 100 times longer than a conventional logic test. To keep the test time as low as possible DRAMs are tested in parallel.

It is desirable to reuse as much as possible from commodity DRAM testing for an eDRAM test. However:

- eDRAM core structures (size, interface width, number of memory banks, pagelength etc.) vary from application to application.
- The memory interface is not directly accessible at the

IO pins. The eDRAM has to be tested as embedded "macro".

- Target quality of eDRAM is application dependent. E. g. in graphics application, "soft" problems are much more acceptable than if eDRAM is used for program data storage.

To keep the test development time and the test time itself as short as possible and to retain maximum reusability, the implications on an eDRAM test concept are:

- Apply standard tests of commodity DRAMs as widely as possible. Adjust the tests to the target quality.
- Apply BIST techniques as widely as possible.
- Multiplex IO pins for eDRAM testing and provide some type of standard pinout for the tester.
- Apply parallel tests of several eDRAM modules on chip level. Compress data which have to be read out from the chip.
- Provide the possibility to test the DRAM with a logic tester.
- Use standard scan techniques to isolate the macro.

Apparently such a testing methodology trades-off test development time and test time with additional logic on-chip. However an eDRAM chip contains always custom logic. Hence the area penalty of this additional logic (some Kgates) is tolerable.

6 DRAM Core Concepts

Flexible memory concepts are a need to allow quick and first right implementations of customized eDRAM macros [17]. A generator for DRAM macros is usually a tilling machine which assembles the predefined selfcontained memory blocks and generates the respective views for the CAD-environment in use. Selfcontained means, that all functions of the memory are already available and that the redundancy is already included. In the case of logic based technologies some first steps to develop real DRAM compilers like for SRAM and ROM applications are on the way. The probablity to succeed with this approach is quite high, because the DRAM structures are less critical than the structures of commodity products. Usually they are 1–2 generations behind the commodity memory. For today's state of the art DRAM-based technologies, a similar approach is not available. In this case the yield normally is too sensitive to even small design and topology changes. Thus the technology choice can restrict the flexibility and automation of a "DRAM generator". For this reason we prefer the name "core concept" instead of "generator".

General requirements on an eDRAM core concept are as follows:

- Building-block architecture with a fine granularity of memory sizes. Reuse of high-volume DRAM subarrays to provide small area penalty.
- Large range of interface width to yield a large fill frequency spectrum and high-speed synchronous interface.
- Multibank architecture and variable page sizes to minimize page-misses.
- Flexible redundancy concept to tailor the redundancy to the core size and quality requirements. eDRAM yield similiar to commodity DRAM yield.
- eDRAM macros must strictly conform to ASIC core methodologies.
- The core concept must provide a test methodology.

A concept which fulfills all these requirements is the SIEMENS embedded DRAM core concept which uses a $0.24\,\mu m$ technology based on its 64/256 Mbit SDRAM process [18]. Figure 4 shows the internal structure of a core composed of 2 banks, with 256 bit interface width and 40 Mbit size.

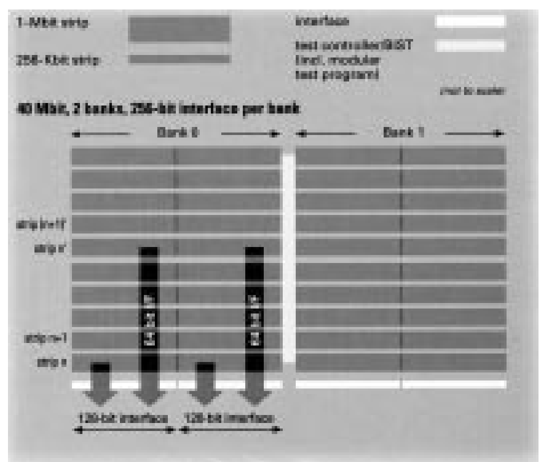

Figure 4: Internal structure of a 40 Mbit core

The concept provides cores with up to 128 Mbit size and interface width up to 512 bit. Table 2 presents key features of some example configurations. The core concept includes an efficient test concept which is based on a small customized test controller with about 5 Kgates. The controller is provided as synthesizable VHDL code and can be controlled as well from a memory tester as from a logic tester. Although the memory bus can be up to 512 bits wide, the test controller only requires access to 30–40 pins which can be switched between normal and test mode.

	2 Mbit	16 Mbit	64 Mbit
die area $[mm^2]$	3.5	23	65
power $[mW/MHz]$	0.9	6.2	6.2
interface width	64	512	512
bandwidth	1.24	9.89	8.94

Table 2: Example configurations

This concept is actually applied to different applications like TV scan-rate converters, TV picture-in-picture chips, modems, speech-processing chips, hard-disk drive controllers, graphics controllers, and networking switches. These applications cover the full range of memory sizes (from a few Mbits to 128 Mbits), interface widths (from 32 to 512 bits), and clock frequencies (from 50 to 150 MHz), which demonstrates the versatility of the concept.

7 Conclusion

Embedded DRAM opens new possibilities for system designers which go far beyond the simple integration of logic and commodity memory. Parameters of commodity DRAMs which designers have been forced to take for given, including size, interface width, and organization, are now available as design parameters. In addition, trade-offs between logic and memory are possible. Designers and chip architects must exploit these new degrees of freedom to develop new system solutions. Flexible memory concepts are necessary to allow fast implementation. New design-for-testability techniques are necessary which consider the special need of memory testing. Furthermore, the transistor-oriented memory design methodology must be merged with high-level based design methodologies.

Acknowledgments

Thanks to our colleagues from the embedded DRAM team of Siemens AG and the University of Kaiserslautern.

References

[1] O. Kimura (Moderator). DRAM + Logic Integration: Which Architecture and Fabrication Process. Evening Discussion Session at the 1997 International Solid State Circuits Conference, February 1997.

[2] J. Borel. Technologies for Multimedia Systems on a Chip. In *1997 International Solid State Circuits Conference, Digest of Technical Papers*, volume 40, pages 18–21, February 1997.

[3] H. De Man. Education for the Deep Submicron Age: Business as Usual? In *Proceedings of the 34th Design Automation Conference*, pages 307–312, June 1997.

[4] N. Dutt (organizer). How will Memory Issues Impact Synthesis for Embedded Systems-on-Silicon? Panel Discussion at the 10th International Symposium on System Synthesis, September 1997.

[5] N. Wehn and S. Hein. Embedded DRAM Architectural Trade-Offs. In *Design, Automation and Test in Europe*, pages 704–708, February 1998.

[6] S. A. Przybylski. *New DRAM Technologies: a comprehensive analysis of the new architectures*. Report, 1996.

[7] B. Prince. *High Performance Memories*. John Wiley & Sons, 1996.

[8] T. Watanabe et al. Modular Architecture for a 6.4-Gbyte/s, 8-Mbit DRAM-integrated media chip. IEEE Journal of Solid-State Circuits, Vol. 32, pp. 635-641, May 1997.

[9] T. Yabe et al. A Configurable DRAM Macro Design for 2112 Derivative Organizations to be synthesized Using a Memory Generator. In *1998 International Solid State Circuits Conference, Digest of Technical Papers*, pages 72–73, February 1998.

[10] K. Ayukawa, T. Watanabe, and S. Narita. An Access-Sequence Control Scheme to Enhance Random-Access Performance of Embedded DRAMs. IEEE Journal of Solid-State Circuits, Vol. 33, No. 5, pp.800-806, May 1998.

[11] F. Catthoor. Energy-delay efficient data storage and transfer architectures: circuit technology versus design methodology solutions. In *Design, Automation and Test in Europe*, pages 709–714, February 1998.

[12] A. Chandrakasan and R. Broderson. Low Power Digital CMOS Design. Kluwer Academic Publisher, 1995.

[13] D. Patterson et al. Intelligent RAM (IRAM): Chips that Remember and Compute. In *1997 International Solid State Circuits Conference, Digest of Technical Papers*, volume 40, pages 224–225, February 1997.

[14] T. Shimizu et al. A Multimedia 32b RISC Microprocessor with 16 Mb DRAM. In *1996 International Solid State Circuits Conference, Digest of Technical Papers*, volume 39, pages 216–217, February 1996.

[15] Y. Aiomoto et al. A 7.68-GIPS 3.84-GB/s 1-W Parallel Image-Processing RAM integrating a 16Mb DRAM and 128 processors. In *1996 International Solid State Circuits Conference, Digest of Technical Papers*, pages 372–373, February 1996.

[16] K. Murakami, S. Shirakawa, and H. Miyajima. Parallel Processing RAM Chip with 256Mb DRAM and Quad Processors. In *1997 International Solid State Circuits Conference, Digest of Technical Papers*, volume 40, pages 228–229, February 1997.

[17] T. Suanaga et al. DRAM Macros for ASIC chips. IEEE Journal of Solid-State Circuits, Vol. 30, pp. 1006-1014, September 1995.

[18] SIEMENS Semiconductor Homepage. `http://www.siemens.de/Semiconductor`, 1997.