

Hierarchical N-Port Memory Architecture based on 1-Port Memory Cells

Hans Jürgen Mattausch

*Research Center for Nanodevices and Systems, Hiroshima University,
Higashi-Hiroshima 739, Japan
Tel.: +81824246268, Fax.: +81824227185
E-Mail: hjm@sxsys.hiroshima-u.ac.jp*

Abstract

The new hierarchical N-port memory architecture features parallel read/write access with low access conflict probability from all ports, although only 1-port memory cells are used. A simple, effective circuit is proposed for conflict handling and monitoring. In comparison with conventional implementation of all N ports in each memory cell, substantial memory area reductions between 28% (2 ports) and 68% (16 ports) can be realized, while access times are nearly equivalent. The architecture is a generalization of the previous state of the art and is applicable for all types of dynamic, static and non-volatile memory.

1. Introduction

Many electronic systems, like multiprocessors, artificial intelligence or multimedia systems, are containing several data processing subsystems. The subsystems work together, on complex computation tasks or to provide complex users services. Development goes towards increasing numbers of such subsystems and towards monolithic integration [1]. On-chip shared memory, which ideally allows independent, parallel and high bandwidth access to a common bases of data and programs from all subsystems, is needed for these systems.



*Fig. 1:
Structure of
the
switching
network
solution
to
the
problem*

Previous approaches to shared memories in monolithically integrated systems have applied techniques, known from large computer systems [e.g. 2]. The basic structure builds on 1-port memories and a separate switching network for 1- to N-port transition (Fig. 1). This separation has the disadvantage of extensive signal routing, which leads to increased access time, area as well as power dissipation. Moreover, practical implementations show a restriction to small numbers of 1-port memories $M(N)$ on the data storage side [2]. High probabilities for undesired access conflicts, when memory blocks are simultaneously accessed from more than one port, result. Conflicts have to be resolved by rejecting/delaying all but one access requirement.

A second straight forward approach implements the required port number in each memory cell [e.g. 3]. The increasing cell sizes, as well as the decoding and data/address routing overhead, lead to a substantial area increase of the memory for each additional port. Especially important are, however, practical layout difficulties, resulting from the problem of fitting a multitude of N final decoder stages into the width of a row (or column) of memory cells. This restricts the reasonable port number to rather small values (N_4) in practice. An exception

are small size register files for processors. Here higher port numbers are implemented, because processor performance, and not the unproportionally large area consumption, is the main issue.

2. New hierarchical N-port memory architecture

This paper proposes a hierarchical concept for achieving the goal of large port numbers N simultaneously with small area increase for additional ports, short access time, and small conflict probability.

The principle for achieving N -port access with 1-port memory cells is explained in Fig. 2. Two levels of hierarchy are exploited. In the general case there are: a) N ports; b) $M=2^m$ memory cells(=bits); c) $M_1=2^{m_1}$, $M_2=2^{m_2}$ elements on the two hierarchy levels ($M=M_1 \cdot M_2$, $m=m_1+m_2$); d) rectangular block arrangement, with 2^{i_1} , 2^{i_2} rows and $2^{m_1-i_1}$, $2^{m_2-i_2}$ columns, respectively. Data ports D_{ij} may be chosen to have any wordlength w_{-1} bit. For simplicity, however, $w=1$ bit is assumed here. To focus on the new features, we skip conventional aspects, like the general operating mode (synchronous /asynchronous), read/write handling or specialities of the different memory types (e.g. ROM, SRAM, DRAM).



Fig. 2: Hierarchical principle for achieving N-port memory access with 1-port cells. A_{ij} , D_{ij} are addresses and input/output data for port i on hierarchy level j , respectively. The important new circuits are highlighted by thicker border lines.

1-port memory cells are grouped on hierarchy level 1 into blocks with conventional word- /bitline decoding (2^{m_1} cells, m_1 address bits). Transition from 1 to N ports is realized with an “Active-Address-Select Circuit” for switching the m_1 address bits A_{i1} of the access requiring port i to the 1-port decoder, and an “Active-Port-Select Buffer” for switching data to/from the access requiring port i . These two circuits are of the multiplexer /demultiplexer type, respectively, and are realized in a straight forward, conventional way.

Hierarchy level 2 contains $M_2=2^{m_2}$ such memory blocks, which appear here to have N -port capability. However, a given memory block is not accessible from more than one port simultaneously. An “Access-Conflict-Resolve Circuit” detects such conflict situations by comparing level 2 port-addresses A_{i2} , and resolves them by an algorithm for port prioritisation, e. g. a ranking of port importance. The “Row- and Column-Select-Signal Generators” on hierarchy level 2 generate activation signals RS_{i1} and CS_{i1} from port address parts A_{i2} . For each port i , these signals activate and control just one block of memory cells, with the “Active-Address-Select Circuit” and the “Active-Port-Select Buffer” serving for correct switching of level 1 address parts A_{i1} and data D_{i1} .

In particular the hierarchical architecture needs only 1-port decoding on hierarchy level 1, so that the layout problem of fitting the final stages of N word-/bitline decoders into the height/width of just one memory cell is relieved. Now the dimensions of a memory block, whose height/width can be adjusted to layout needs, are available for the final stages of the N “Row- and Column-Select-Signal Generators”.

Hierarchy has previously been used, with the focus on shortening access times in 1-port memories (mainly DRAMS). Examples include on-chip cache and banking [4], pipelining [5], and partitioning techniques [6]. A bitline hierarchy has been exploited for achieving 2-port access in fast cache memories [7]. The proposed architecture may appear, to have some similarities with the banking technique. However, banking employs memory subdivision and interleaved access operations to the memory blocks, to increase the bandwidth of data via a single port. Interleaved data is transported to/from the port by a fast bus. If the bus would be used to serve several ports, the obtained architecture corresponds to a switching network solution (Fig. 1).

3. Access conflict problem

All multiport memory architectures, which implement parallel access in real time, suffer from the problem of possible access conflicts (at least for write access). Thus, the discussion in this section applies also to port implementation in each memory cell and to switching network solutions.

3.1 Handling of occurring conflicts

For conflict handling, a conflict resolve algorithm has to be provided, which would ideally lead to equal access rejection probabilities from each port (fair algorithm). However, a fair algorithm normally requires large gate numbers for implementation. We therefore concentrate here on the “Port-Importance-Hierarchy” (PIH) algorithm, where the highest ranking port gets priority in a conflict situation.

Fig. 3 shows an “Access-Conflict-Resolve Circuit”, which implements the PIH algorithm for the case of 4 ports and $M_2=2^{m_2}$ blocks. Level 2 address parts A_{i2} of the 4 ports (m_2 bit each) are compared by EXNOR logic. The circuit generates “Port-Blocking Signals” PB_i , which are “0” in the case of no conflict and which become “1” for the less important ports in the case of a single or a multiple conflict. The “Access-Conflict-Resolve Circuit” works in parallel to the “Row- and Column-Select-Signal Generators” and therefore doesn’t increase memory access time. Signals PB_i are used, to block the final stages of the “Row- and Column-Select-Signal Generator” of port i in case of access rejection and can in addition monitor the port status



Fig. 3 : “Access-Conflict-Resolve Circuit” implementing port importance hierarchy (PIH) for 4 ports with EXNOR and OR logic. A_{i2} are the m_2 level 2 address bits for port i . PB_i are the Port-Blocking Signals for port i .

(access OK / rejected) for the system. A rejected access would normally be repeated in the next access cycle. Generalization of the circuit in Fig. 3 is straight forward. Each new port i requires an additional row of $(i-1)$ EXNOR functions for address comparison and one additional OR with $(i-1)$ inputs. For a given port number N , the required number of logic functions is thus given by $G_{PIH} = N(N+1)/2 - 2$.

Fair access needs much higher gate numbers. An algorithm, which e.g. groups the M_2 blocks on level 2 into N equal sized subgroups and changes port importance hierarchy in these subgroups in a cyclic

way, needs 5 times increased gate numbers.

3.2 Keeping conflict probability small

Large conflict probability is prohibitive for practical employment. Clearly conflict probability will decrease with increasing block number M_2 . But more precise information is needed, on how many blocks M_2 are required for a given port number N . Here the relationships for a statistical access distribution are given.

$P_i(N, M_2) = Z_i(N, M_2) / Z(N, M_2)$ defines the probability for an unsuccessful access from port i . $Z_i(N, M_2)$ is the number of possible access configurations from the N ports to the M_2 memory blocks, which

$$P_i(N, M_2) = \frac{\sum_{k=1}^{N-1} Z_i(N, M_2, k)}{\sum_{k=0}^{N-1} Z(N, M_2, k)} \quad (1)$$

lead to the rejection of the access requirement from port i . $Z(N, M_2)$ is the total number of possible access configurations. It is advantageous, to group these configurations according to the number k ($0 \leq k \leq N-1$) of access rejections they are containing (eq. 1). Since the configuration number decreases rapidly with increasing multiplicity k of the conflict, only the first terms in (1) have to be considered. Determination of the number of

configurations for a given k is a combinatorial problem. The possible number of mappings of the N ports onto the M_2 memory blocks, which lead to k rejections, has to be calculated. Without any conflicts ($k=0$), the number of these mappings is e.g. given by the variation of N out of M_2 elements $Z(N, M_2, k=0) = M_2! / (M_2 - N)!$. For the case of 4 ports the above described mathematical treatment leads to the following exact results:

$$P_F(4, M_2) = \frac{3}{2M_2} \left(1 - \frac{2}{3M_2} + \frac{1}{6M_2^2} \right) \quad (2)$$

$$P_{PIH4}(4, M_2) = \frac{3}{M_2} \left(1 - \frac{1}{M_2} + \frac{1}{3M_2^2} \right) \quad (3)$$

$P_F(4, M_2)$ is the rejection probability with a fair conflict resolve algorithm and $P_{PIH4}(4, M_2)$ is the rejection probability for the least important port with the PIH conflict resolve algorithm.

$$P_F(N, M_2, k \leq 1) = \frac{\frac{N-1}{2(M_2 - N + 1)}}{1 + \frac{N(N-1)}{2(M_2 - N + 1)}} \quad (4)$$

$$P_{PIHN}(N, M_2, k \leq 1) = 2P_F(N, M_2, k \leq 1) \quad (5)$$

If multiple conflicts (i.e. $k > 1$) are neglected, which is a good approximation for large $M_2 - N^2$, the general relations (4), (5) for the rejection probabilities are obtained. Two properties of equations (2)-(5) should be noted: a) The most important term is always of the order M_2^{-1} . b) The least important port of the PIH algorithm has

just a factor 2 higher rejection probability, than a fair algorithm would offer, if large enough block numbers $M_2 - N^2$ are chosen. Property b) simply represents the fact, that the number of higher priority ports is always a factor 2 larger for the least important port, than for a port of median importance.

The PIH conflict resolve algorithm is therefore sufficient for the new hierarchical N -port memory architecture. Since conflict probability decreases inversely proportional to the block number M_2 , its deficit of a factor 2 in comparison to a fair algorithm can be easily compensated by choosing a factor 2 larger block number M_2 . Numerical evaluation of equations (3), (5) shows, that the choice $M_2 = N^2$ is sufficient to keep rejection probability below 0.2 (below 0.1 with $N > 6$) for the least important port.

4. Area and access time

Design data [8] of 1- and 2-port SRAM's (equal design rules, storage capacity, wordlength etc.) is linearly extrapolated, to perform the comparison between conventional implementation of all N ports in each

TABLE 1

Normalized area, access time estimates for conventional (port implementation in memory cell) and hierarchical N-port SRAM's, based on 1- and 2-port SRAM design data [8].

kind of SRAM data	area	area of cells	area of rest	access time
design data [8]	1	~0.75	~0.25	1
1-port	~1.6	~1.2	~0.4	~1.15
2-ports	$1+0.6 \cdot (N-1)$	$0.75+0.45 \cdot (N-1)$	$0.25+0.15 \cdot (N-1)$	$1+0.15 \cdot (N-1)$
N-ports estimate	$1+0.15 \cdot (N-1)$	0.75	$0.25+0.15 \cdot (N-1)$	$1+0.15 \cdot (N-1)$
conventional				
hierarchical				

memory cell and the new hierarchical concept. TABLE 1 shows the design data and the extrapolations (same design rules, capacity etc.), all normalized to the respective values of

the 1-port SRAM.

The step from 1 to 2 ports means an area increase of about 60% (45% cells, 15% decoding and routing), if ports are implemented in each memory cell. Each new port in a conventional design is assumed, to require the same additional area, with unchanged proportion between cell and decoding/routing area. Area increase for additional ports, comes only from additional decoding/routing, if the new hierarchical architecture is applied. It is estimated, to have the same magnitude, as the area increase from decoding/routing in a conventional design. With the resulting equations of TABLE 1, area savings between 28% (2 ports) and 68% (16 ports) are calculated.

Access time estimation for the hierarchical architecture has to consider, that: a) Memory subdivision reduces word-/bitline capacitances, as well as RC-delay for gate level wordlines. Consequently signal switching in these most critical access path portions will be fast. b) Smaller memory area will reduce routing capacitances and thus switching times in all access path portions. c) Transition from 1- to N-port capability, will require 2 additional gate delays in the access path. Aspects a), b) are expected, to be dominating in most designs, so that an access time advantage may be achieved. In TABLE 1 the more conservative estimate of no access time penalty is made.

Comparison to the solution with a separate switching network is more difficult, because comparable design data (same design rules etc.) is not available. However, separation of port number transition and memory part leads to extensive signal routing. Hence, increased area and power consumption as well as longer access times result. The penalty is expected to be at least about 20%. Moreover it's difficult, to achieve the large memory block numbers, necessary for low conflict probability, with a separate switching network.

5. Discussion and conclusion

The proposed hierarchical N-port memory architecture is a generalization of previous concepts. It capitalizes on good points, while avoiding the bad points: (i) Port implementation in each memory cell can be regarded as the special case, where the number of memory cells in each block (hierarchy level 1) is just one cell. The "Active-Address-Select Circuit" and the "Active-Port-Select Buffer" thus reduce to single transistors or transistor pairs, while block internal word-/bitline decoders are not necessary. (ii) Transition from 1 to N-ports is contained in the hierarchical architecture and

TABLE 2

Ranking of architecture concepts with respect to important design criteria for multiport memories

criteria	switching network	N-port memory cell	hierarchical architecture
area	2	3	1
access time	3	1	1
power loss	2	3	1
conflict prob.	3	1	2
large N	2	3	1

not separated from the memory part. Thus a modular and regular concept is achieved.

TABLE 2 summarizes our results by a ranking of the possible architecture concepts with respect to the important design criteria for multiport memories. The hierarchical architecture alone offers solutions, satisfying simultaneously all of these criteria.

In conclusion, the new hierarchical N-port memory architecture has large flexibility, to construct optimum solutions for shared memories in future integrated systems. It can be implemented with similar advantages for all known types of dynamic, static and non-volatile memory. Fast access (e.g. SRAM technology) as well as large storage (e.g. DRAM technology) can be realized. Because of the modular and regular concept, large numbers of memory blocks, to keep access conflict probability small, are easily implemented. Besides exploitation for system integration, derivation of multiport memory chips is of course possible. Already 2- or 3-port memories with real time parallel access may be of interest for practical realization.

Acknowledgment

The author thanks A. Iwata, S. Yokoyama, T. Ae, M. Nagata, K. Shibahara and M. Hirose, all with Hiroshima University, for very helpful discussions.

References

- [1] H. Sasaki, "Multimedia Complex on a chip", ISSCC Dig. of Tech. Papers, pp. 16-19, 1996
- [2] K. Gutttag, R. J. Gove, and J. R. Van Aken, "A Single-Chip Multiprocessor for Multimedia: The MVP", IEEE

Computer Graphics & App., vol. 12, pp. 53-64, 1992

- [3] K. Tran, "Demonstration of 5T SRAM and 6T Dual-Port RAM Cell Arrays", Sym. on VLSI Circuits, pp. 68-69, 1996
- [4] Y. Nitta et al, "A 1.6 GB/s Data-Rate 1Gb Synchronous DRAM with Hierarchical Square-Shaped Memory Block and Distributed Bank Architecture", ISSCC Dig. of Tech. Papers, pp. 376-377, 1996
- [5] H.-J. Yoo et al, "A 150Mhz 8-Banks 256M Synchronous DRAM with Wave Pipelining Methods", ISSCC Digest of Tech. Papers, pp. 250-251, 1995
- [6] M. Nakamura et al, "A 29ns 64Mb DRAM with Hierarchical Array Architecture", ISSCC Digest of Tech. Papers, pp. 246-247, 1995
- [7] K. Osada, H. Higuchi, K. Ishibashi, N. Hashimoto, and K. Shiozawa "A 2ns Access, 285Mhz, Two-Port Cache Macro using Double Global Bit-Line Pairs", ISSCC Dig. of Tech. Papers, pp. 402-403, 1997
- [8] 1- and 2-port SRAM designs of the author at Siemens AG, Germany, unpublished