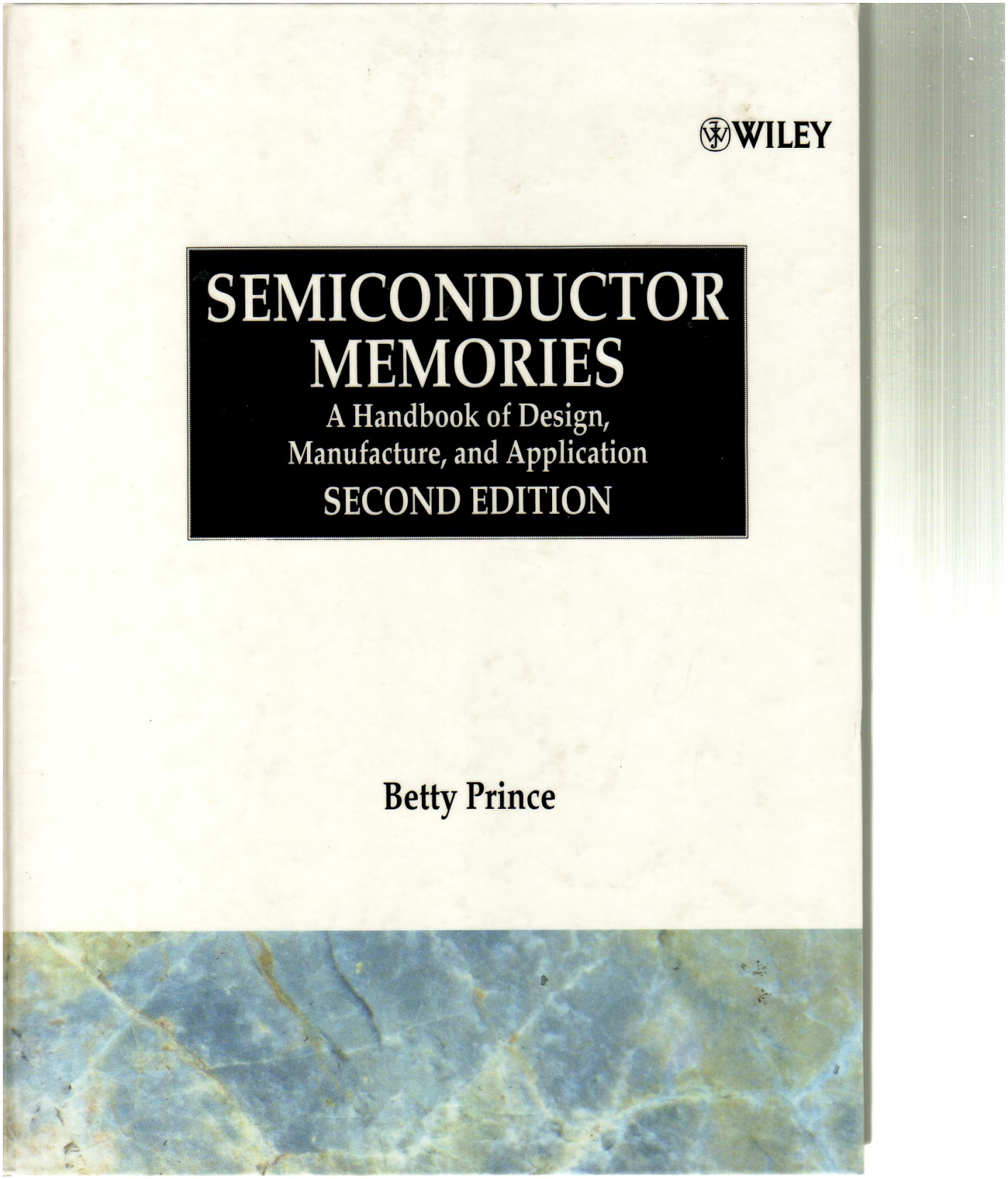


Semiconductor Memories

A Handbook of Design, Manufacture and Application

Second Edition

Betty Prince
Texas Instruments, USA

JOHN WILEY & SONS
Chichester • New York • Brisbane • Toronto • Singapore

Second edition of the book *Semiconductor Memories* by B. Prince and G. Due-Gundersen

Reprinted with corrections February 1995
Reprinted January 1996
Reprinted June 1996
Reprinted June 1997

Other Wiley Editorial Offices

John Wiley & Sons, Inc., 605 Third Avenue,
New York, NY 10158-0012, USA

Jacaranda Wiley Ltd, G.P.O. Box 859, Brisbane,
Queensland 4001, Australia

John Wiley & Sons (Canada) Ltd, 22 Worcester Road,
Rexdale, Ontario M9W 1L1, Canada

John Wiley & Sons (SEA) Pte Ltd, 37 Jalan Pemimpin #05-04,
Block B, Union Industrial Building, Singapore 2057

Transferred to digital printing

British Library Cataloguing-in-Publication Data

A catalogue record for the book is
available from the British Library

ISBN 0 471 92465 2 ; 0 471 94295 2 (pbk)

Typeset by Techset Composition Limited, Salisbury, Wiltshire
Printed and bound by Antony Rowe Ltd, Chippenham, Wiltshire

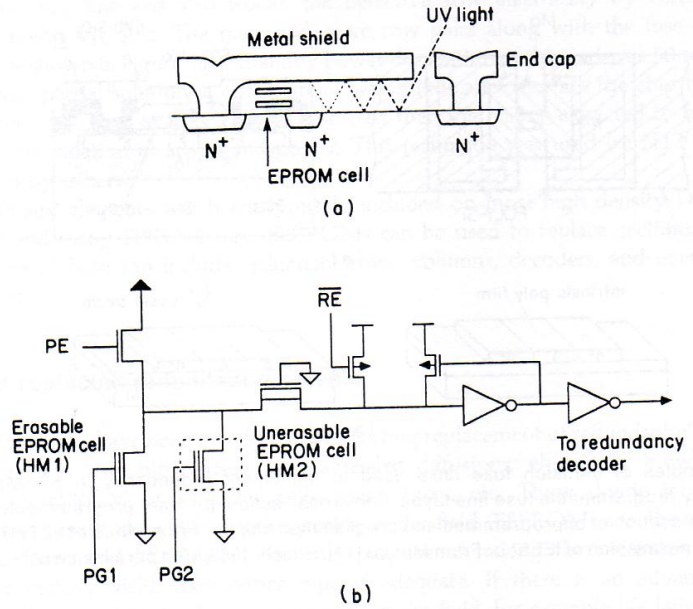


Figure 15.8
(a) Schematic cross-section of UPROM cell showing basic EPROM storage transistor with metal shield to prevent UV light from erasing the programmed cell. (b) Redundancy address circuit using UPROM cell. (a: From Gaw *et al.* [10], Intel 1985, with permission of IEEE; b: From Higuchi *et al.* [12], NEC 1990, with permission of IEEE.)

NEC [12] on a 16Mb EPROM in 1990 also used the UPROM but in conjunction with an EPROM on the redundant elements. For in process test before and after assembly the EPROM was programmed. Then at final test the UPROM can be programmed. This technique permits complete testing at each stage of the manufacturing process. Figure 15.8(b) shows the parallel EPROM and UPROM cells on the redundancy address circuit.

Redundancy on a EEPROM has been included by Intel [3] in 1983 with the use of an EEPROM cell and a CAM. In this case each redundant row has one CAM associated for each row select address. In addition it has a CAM which can be programmed to select that row for use at the address encoded in the CAMs. During testing the number of defective rows and their addresses are remembered. If this is equal or less than the number of redundant rows available, then the redundant rows are tested and, if good, are programmed in. Lattice [4], in 1985, also described a fast EEPROM using EEPROM programmable fuse elements.

15.4 ERROR CORRECTION

Redundancy is used primarily for correcting 'hard' errors in the memory circuits to improve the production yield. With error correction it is possible to detect and correct

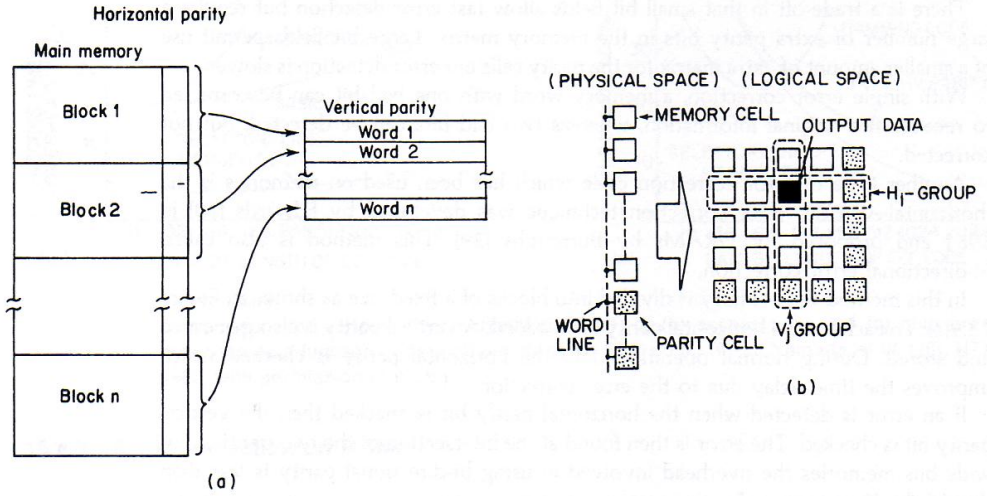


Figure 15.9
Basic bidirectional parity code technique showing: (a) memory divided into blocks of fixed size with a horizontal parity bit added to each byte and a vertical parity bit stored for each block. (b) memory cell with parity cells shown in physical space and horizontal–vertical (*H–V*) groups in logical space. (a: From Osman [29], Unisys 1987, with permission of IEEE; b: From Yamada *et al.* [28], NTT 1987, with permission of IEEE.)

both soft and hard errors occurring either in manufacturing or in the field and so improve the reliability of the memory device in the system as well as the yield. The two methods can be used effectively in combination to give improvement in long term in-system yield.

15.4.1 Error correction theory

Error correction involves two steps. One is detecting the error and the other is correcting it. The simplest method of error correction is a simple complete duplication of every bit in the array. The drawback is, of course, the increase in the chip size. It is possible to use various coding techniques which take less area on the chip.

Hamming codes [1], which were first developed by Hamming in 1950, are frequently used for this purpose. With a Hamming code, error correction can be performed on many different bit field sizes. This entails using additional parity bits in the array. To determine the number of parity bits needed to correct a given number of data bits, the following inequality must be satisfied

$$2k \geqslant m + k + 1$$

where 'm' is the number of data bits to be corrected and 'k' is the number of parity bits needed for correction.

There is a trade-off in that small bit fields allow fast error detection but require a large number of extra parity bits in the memory matrix. Large bit fields permit use of a smaller amount of extra matrix for the parity cells but error detection is slower.

With single error correction, a memory word with one bad bit can be corrected to recover the original information, whereas two bad bits can be detected, but not corrected.

Another type of error correction code which has been used on memories is the 'horizontal–vertical' error correction technique was developed by Edwards [30] in 1981 and proposed for DRAMs by Burroughs [29]. This method is also called bi-directional error correction.

In this method the memory is divided into blocks of a fixed size as shown in Figure 15.9(a). To each byte, a horizontal parity bit is added. A vertical parity is also generated and stored. During normal operation only the horizontal parity is checked which improves the time delay due to the error correction.

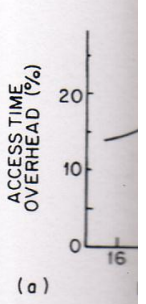
If an error is detected when the horizontal parity bit is checked then the vertical parity bit is checked. The error is then found at the intersection of the two parities. For wide bus memories the overhead involved in using bi-directional parity is less than that of the Hamming codes.

The basic concept is shown generalized in Figure 15.9(b). Memory cells and parity cells associated with each word line are arranged into a two-dimensional logical space which forms horizontal and vertical groups. Parity information for each data group is written in the appropriate parity cell. Error correction is done by checking the parity of the associated pair of horizontal and vertical parity bits. This technique allows simple ECC operation with smaller error correction circuitry than does Hamming code.

In order to compose a bidirectional parity code $k + m + 1$ parity cells are added to $k \times m$ memory cells on each word-line. The same number of bit-lines are also added to read these parity cell data. A pair of horizontal and vertical data selectors are arranged close to the cell array to select a pair of H and V groups including the output data.

The error detection circuit consists of only a '$k + 1$' bit, a horizontal parity checker, a '$m + 1$' bit vertical parity checker, and a two input AND gate. Two kinds of parity checking are carried out simultaneously and then the two input AND gate operates as a decoder which detects a bit error from the results of the parity checking and supplies the error correcting signal. Using this signal the data is corrected through the XOR gate. The improvement in access time and chip area overhead as a function of data bit length of the bidirectional (H–V) error correction code are shown in Figures 15.10(a) and (b). The figure also shows the additional improvement of a variation of the bidirectional error correction code called selector line merged ECC which reassigns the H–V logical space diagonally along word-lines to permit short selector lines for both the horizontal and vertical selectors.

Error correction is useful in systems not only to correct soft errors due to noise or alpha particle hits, but also for hard error correction. A soft error can be corrected by reprogramming the bit. A hard error is a 'stuck at' failure. Since only one error in a word can be corrected, a system failure with error correction present occurs only if two errors occur in the same word. This occurrence can be lessened by clearing the soft errors more frequently and by regularly replacing parts with hard errors.



15.4.2 Erro