B. F. Fitzgerald
E. P. Thoma

Circuit Implementation of Fusible Redundant Addresses on RAMs for Productivity Enhancement

This paper describes the circuit schemes used to substitute redundant storage locations for defective ones found during testing. Word or bit lines are added along with appropriate bit steering circuitry to allow the replacement of a defective word or bit line. On-chip storage elements are "set" by the tester and used to store the binary addresses of the failing word or bit lines, which are then compared to the incoming addresses by the redundancy circuitry. This circuitry then activates the replacement word or bit lines and, by various means described, steers out the defective ones. A variation is described briefly which includes a word redundant circuit scheme that provides no penalty in memory access time by using separate sense amplifiers for the redundant lines.

Introduction

During the processing and manufacturing of memory chips, process-induced defects occur which can cause a small number of the cell storage locations to become inoperative without affecting the peripheral circuitry. The density of these defects becomes more critical to the overall yield as physical chip dimensions are increased and minimum line spacings are reduced. In the case of high density RAMs, increased yield and enhanced productivity can be obtained by adding redundant cell storage locations capable of replacing those affected by process-induced defects.

The use of redundancy for yield enhancement is not new; several schemes for redundancy implementation on core memories were published by Sakalay, Fletcher, and Kril [1-4]. A general redundancy scheme and its potential yield benefits were shown by Schuster [5], while Arzubi [6] and later Fitzgerald and Kemerer [7] devised methods for implementation on integrated circuit memory chips. Recently, applications of redundancy were presented by DeSimone *et al.* [8] for a family of RAMs, and Cenker *et al.* [9] also showed a practical application. Extensions of redundancy to higher levels of packaging are also feasible, as pointed out by Egawa *et al.* [10].

For the highest efficiency, the defect density of the process line and critical susceptible chip area must be matched with the amount of redundant locations added. Chip productivity then becomes a function of the defect densities, critical chip areas, and redundancy circuit efficiency.

Circuit implementation from a schematic and physical layout standpoint has a large influence on the overall effectiveness of redundancy. Considerations such as speed, transparency, bit replacement scheme, overall chip size, etc., must be weighed. This paper addresses two approaches for word and two for bit redundant circuit schemes, emphasizing the objectives and design tradeoffs involved. One redundant word line and one redundant bit line scheme are variants of known straightforward schemes, and more sophisticated approaches are described for both redundant word and bit line approaches.

Redundancy optimization

As redundant word and bit lines and the appropriate steering circuitry are added to a RAM, the chip size and the chip area susceptible to defects increase. Redundancy optimization then requires weighing the improvement in yield (and consequently in productivity) due to the redundancy against the reduction in productivity resulting from a larger die size and the additional defect-sensitive circuitry. As more redundancy is added, both the yield and

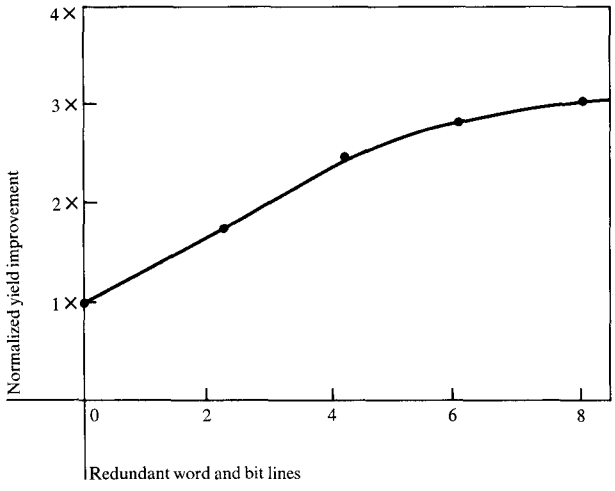

Figure 1 Normalized chip yield *versus* number of added redundant lines (based on the yield model).

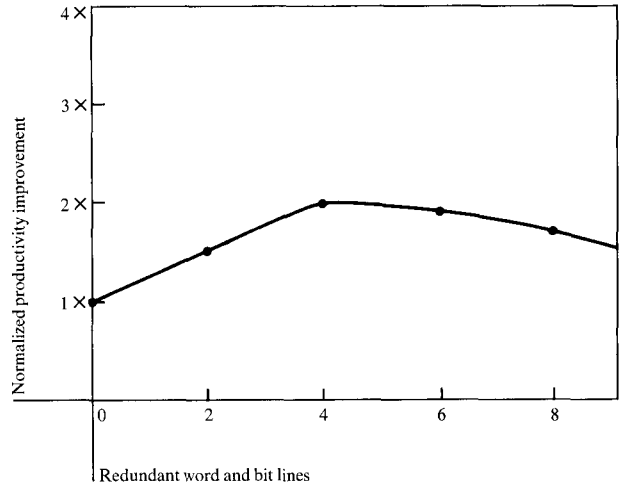

Figure 2 Normalized productivity *versus* number of added redundant lines (based on the yield model).

productivity of the RAM increase; however, a point is reached where optimum productivity is obtained. Additional redundancy only slightly increases productivity due to the trade-off between the added redundant circuit area and the fixability improvement in the array (see Figs. 1 and 2).

Yield calculations for memory chips with redundancy have been documented by numerous authors. Chen [11] calculated yields using multiple word and bit line redundancy, while Tammaru and Angell [12] incorporated redundancy for the computation of yields for memory and logic chips. The yield equations developed by these authors and others can be incorporated into models describing the perfect chip, fixable chip, and redundant circuit yields. The model we used for redundancy optimization was developed by Stapper, McLaren, and Dreckmann [13]. It relates process line defect densities and average defect sizes to the photolithographic ground rules and the defect-sensitive chip areas. Circuit design inputs consist of fault types fixed and not fixed by redundancy, along with the areas sensitive to defects in both the original and redundant circuitry. Once models such as this one are developed, they then can be used in an iterative fashion to determine optimum productivity.

The effectiveness of word redundancy *versus* bit redundancy may also vary. Word redundancy corrects faults on a different process level than bit redundancy, and, therefore, the effectiveness of each may be different based upon the defect densities and minimum ground rules used at each level (both can correct single cell faults). Also, bit redundancy can correct faults in the sense latches, while both word and bit redundancy can repair failures in their respective decoders. The implementation algorithm used after test for steering the replacement lines can also affect the overall word or bit redundancy efficiency.

To optimize redundancy, not only must the yield models be used to determine the correct number of replacement addresses, but other factors must be considered to achieve the breakdown between word and bit dimensions. Also, since the optimization point greatly depends on process line defect densities, which may be quite variable in the early period of manufacturing, a later chip redesign reducing the amount of redundancy may be in order.

Redundancy circuit objectives

Once the redundancy optimization is completed, the exact number of redundant word and bit lines to be added to a particular RAM is known. However, more specific circuit objectives must be developed that are intimately related to the chip operation.

Most methods of redundancy implementation have similar functional objectives, which are:

1. Transparency to the user once the redundant lines have been selected.
2. Maximum flexibility as allowed by the chip organization.
3. Little or no impact on chip performance.
4. Smallest size and lowest power requirements.
5. Minimum impact on chip reliability.

The choice of the method of storing the information identifying the locations of the defective lines may de-

pend largely on chip processing steps and chip system application. There are many ways in which to store this information, such as in latches, by laser personalization, through the use of fusible links, or with electronically programmable storage elements (EPROMs). Latches on the chip are volatile and therefore require a backup at the system level, whereas fusible (programmable) links and programmable storage elements do not; however, these elements must be added at some convenient processing level on the chip to be personalized after test. Field alterability of the redundancy information offered by various techniques could also be an advantage based on the overall computer reliability and organization. We chose fusible links for a family of RAMs as a medium for storing the redundancy information, primarily because they are nonvolatile. The links were constructed on a second layer of metal and "set" or blown using an electrical discharge.

The number of redundant bit and word lines to be added determines the minimum number of fuses needed. If M is the number of redundant word lines and 2^N is the number of regular word lines, then the minimum number of fuses is $M(N + 1)$. With this scheme for calculating the minimum number of fuses, each fuse is used to hold one bit of the binary address of the word or bit line to be replaced by a specific redundant line. In addition, each redundant line has an enable fuse that must be blown to activate that line.

The chip description dictates many of the objectives for the redundancy circuitry. For high speed RAMs, performance penalties, if any, must be minimized. Multiple output bit organization requires bit or word steering circuitry to achieve an any-for-any replacement over the entire word or bit dimension. Read-modify-write or partial store functions may require expanded control over the redundant lines due to the different requirements these functions have on the address selection.

The most important circuit objectives for the redundancy circuitry, however, come from the internal chip timings and the overall amount of space that can be allocated for redundancy. The chip timings dictate the speed at which the redundancy path must operate to match the normal chip access path. This, along with the area constraints, determines circuit design type, power requirements, and redundant circuit replacement method.

Although the functional requirements of redundancy are generally universal, the actual circuit requirements of the circuit function are unique to every RAM design. The following set of assumptions apply to the circuit approaches we used for the implementation of redundancy:

1. More than one word or bit line,
2. Fusible link programming,
3. Minimum or no performance impact,
4. Any-for-any replacement.
5. Multiple output bit (more than one output chip pad).

Word redundancy

During normal operation, the word line on a RAM is timed to be activated as soon as the word decoding is completed; the addition of word redundancy is the most difficult to incorporate with little or no penalty in access time. The straightforward scheme we used on the 64K-bit RAM (described first) requires the deselection of the normal word decoder associated with the defective word line to occur after the decision has been made to activate redundancy. A more sophisticated approach used on the IBM 32K-bit RAM shares the same basic circuitry; however, due to its organization, the need to deselect the normal word decoder is not present. The latter approach can be incorporated with no access penalty, whereas the first approach described may contribute one. Both are now described in detail.

• *Straightforward approach*

For the simpler word redundancy scheme, the incoming binary address for the word dimension is buffered and inverted in the regular manner. Each binary bit of this address decodes the word system such that only one word line driver is active. In the meantime, the binary bits are individually compared to the state of the fuses. If more than one redundant word line is present, this comparison occurs in parallel for each redundant line. When the incoming address matches the state of the fuses, the redundant word line is selected. In addition, the deselection of the normally active word line is done using a deselect generator, completing the replacement procedure (see Fig. 3).

Investigation of the access path shows that the time until activation of the word line for normal operation is determined only by the speed of the word decoder after the formation of the true and complement addresses. However, the access time for the redundant line is determined by additional circuit blocks, the compare circuitry and the redundant decoder, including the deselect driver. Only after the time delay incurred by these redundant circuits can the normal decoder be deselected. This time delay is considered the access time adder for this type of redundancy.

The access time adder can be reduced by increasing the speed of the compare circuit, redundant decoder, and deselect generator. It is generally quite simple to make the time delay through the redundant circuits equal to or

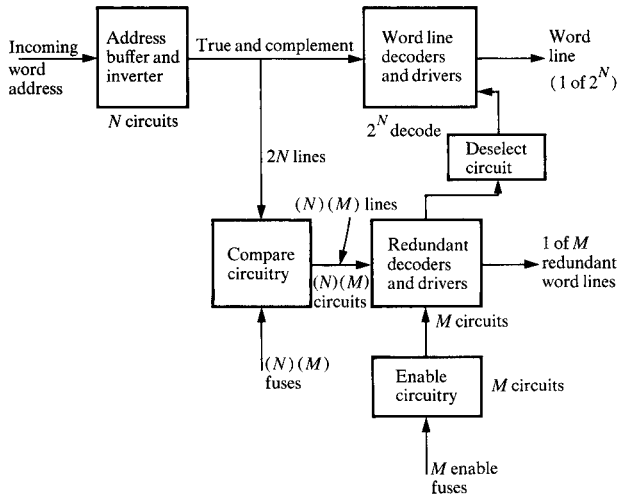

Figure 3 Block diagram of conventional word redundancy scheme assuming M redundant word lines and N address bits.

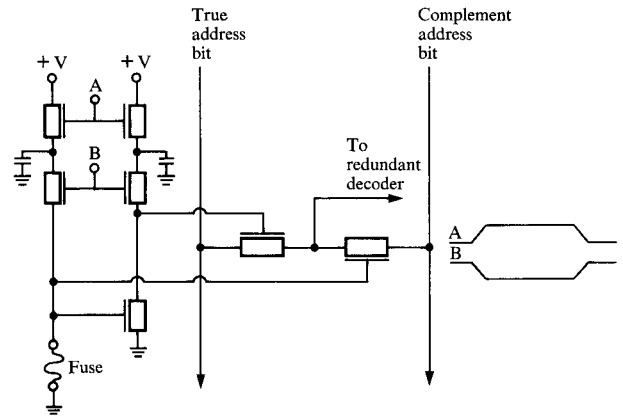

Figure 4 Compare circuitry.

less than the decode time of the regular decoder, so that the adder is less than the normal decode time. In certain cases, this adder can be "hidden" in the access path by revising the regular decoders so that the speed of deselection of this circuit when initiated by the redundant deselect generator is faster than when initiated by the normal buffered addresses. For example, if the normal decode deselect time minus the decode deselect time by the deselect generator is equal to or greater than the time delay through the redundant circuits, then this scheme would have no impact on performance.

The heart of the redundancy circuitry is the compare circuit. Since many of these circuits must be added, simplicity and small size are virtues. An efficient compare circuit is shown in Fig. 4. The basic operation consists of allowing either the true or complement of any address to pass to the redundant decoder based on the state of the fuse. A binary address matches the state of the fuses when the compare circuits all allow the "zero-volt" true or complement address bits to pass. The compare function can actually be done with only three devices; however, the additional devices shown are used to eliminate dc power dissipation through the fuse. Bootstrapping can also be added to the pass devices to enhance performance.

An additional requirement for this type of word redundancy involves the dummy or half-cell word line. If the sensing scheme used employs a dummy cell technique, additional control may be required beyond the dummy word line decoder. When the replacement redundant line is located on the same side of the sense latches as the line to be replaced, no additional control is necessary. However, if the replacement line is on the opposite side, then the selection of the dummy word lines must be inverted. This requires a modification to the dummy decoder/driver such that activation and deactivation is a function of both normal decoding and replacement word line location.

Sophisticated approach

Another system of implementation includes word redundancy circuitry that incurs no penalty in memory access time by using separate sense amplifiers for the redundant word lines and selectively utilizing "data read" from the redundant lines, as shown in Fig. 5. The general concept of this type of redundancy is: if a match occurs between the incoming address and the fuses, the redundant word line is activated; however, due to the use of auxiliary sense amplifiers, it is not necessary to "go back" and deselect the regular word line. The elimination of this deselect step avoids an access penalty. The decision of which read/write buffer to use occurs in parallel with the sensing operation and, therefore, is "hidden" in the access time. In effect, since there is no interaction between the regular sense latches and the auxiliary ones, the redundant system can be activated, allowing the normal system to continue to operate. Through the operation of the data output direct circuit, data are taken from the auxiliary sense latches and inhibited from the normal latches.

During a store operation, data are driven through both read/write buffers into the selected array and the redundant storage locations. Thus, the data direct control is not needed for the store operation.

This redundancy technique lends itself nicely to the use of two arrays with common bit dimension addressing between them. If a defective word line is located in the left

array, then its redundant replacement would be located in the right array, and vice versa. The need for the auxiliary sense amplifiers is thereby eliminated. Without the two-array organization, the size impact of this type of redundancy is generally greater than the first type described. The major disadvantage with this approach is when a defect causes a short circuit from the defective line to the power supply. However, this problem can be virtually eliminated by using resistance isolation between the normal word lines and the word line generator so that the generator's operation is unaffected.

Testability of the redundancy before permanently steering it to the failed address is a desirable feature that can be offered with either type of redundancy. Since the enable circuit is already present and initiates the redundancy when the enable fuse is blown, additional control can be added for testability. For example, an input pad can be added that, when selected, activates an enable circuit that is assigned to a particular redundant line even though the enable fuse has yet to be blown. The replaced address is represented by the state of the unblown fuses. If there is more than one redundant word line present, the "unblown address" can be altered by inverting the sense of a few compare circuits so that each redundant line replaces a different normal line. With this testability feature, a simple test can be developed to guarantee redundancy functionality.

Bit redundancy

Bit redundancy is in many ways similar to word redundancy with respect to the basic building block circuitry involved, but there are two major differences. The first is that the high performance requirements necessary to minimize an access time adder are not as stringent for the bit dimension. The second difference is in its application to a multiple output bit memory. When a chip is organized so that it has one data latch (*i.e.*, one output bit per fetch cycle), the bit redundancy replacement is straightforward. However, when chip organization is such that the bit lines are grouped, with each group having a separate data latch (*i.e.*, multiple output bits per fetch cycle), the redundant bit lines must traverse the data latch dimension for maximum flexibility. Otherwise, each group of bit lines must have, in its group, a redundant bit line in order to employ the straightforward replacement scheme, which would use chip area inefficiently.

- *Straightforward scheme*

For the simpler scheme (one data latch) the data are written into both the defective and redundant bit lines during a store operation. For a dynamic memory chip, the store operation activates the bit switch early in the cycle, allowing the read/write head to write the data. By allowing writing into both the normal and redundant lines, it is not necessary to disconnect the data path to the defective line at this early point in the cycle. This technique eliminates any access time adder (assuming the store and fetch cycle times to be the same) that may have existed had deselection been necessary. It should be noted again that, by not deselecting the defective line for the store operation, certain defects causing shorting to the power supplies become nonfixable.

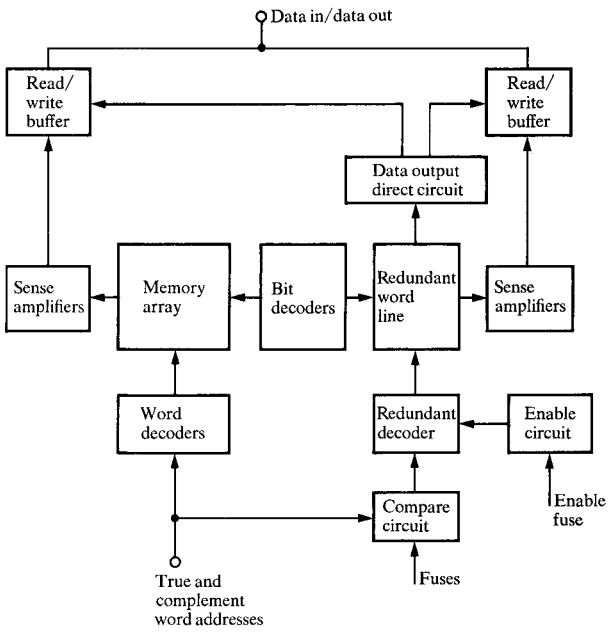

Figure 5 Word redundancy variation.

For the fetch operation, deselection of the data path to the defective bit line is necessary. However, this deselection can occur much later in the cycle. The deselection is typically done during the sensing time and thereby is hidden in the access path. The bit redundancy concept for the fetch parallels that of the first word redundancy scheme described, whereas the store operation is like the second approach. (See Fig. 6.) The circuitry used can be virtually the same as the word system redundancy with only a slight modification to the deselect generator for the store operation. Actual deselection of the normal bit line can be done at the bit switch, with redundant control incorporated into the bit decoders.

- *Sophisticated scheme*

Bit redundancy is much more complex if the bit substitution must traverse data paths or data latches. Not only must the redundancy be steered by bit address, but it must also be controlled by data pad assignment so that redundancy can be directed to any address and any data

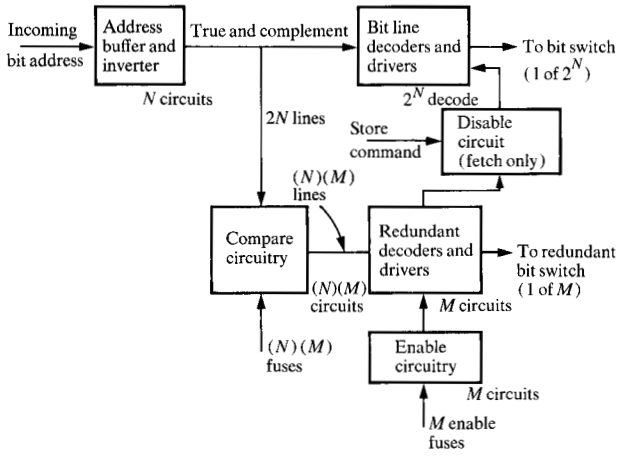

Figure 6 Bit redundancy assuming one chip data-out pad, M redundant bit lines, and N address bits.

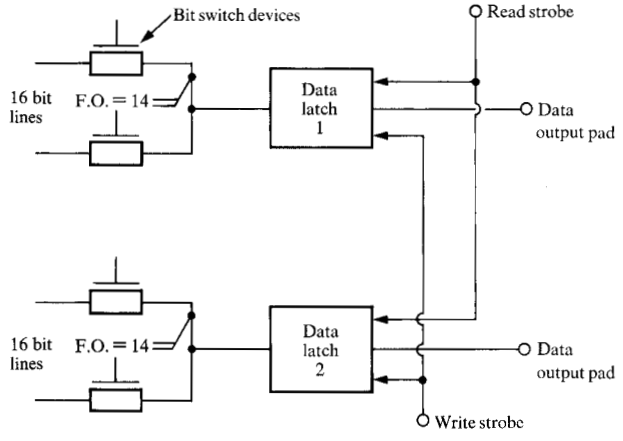

Figure 7 Two-bit chip.

pad. To show the steering necessary for this type of bit redundancy, we apply the methodology to a hypothetical two-bit output chip containing 32 regular bit lines. Figure 7 shows the block diagram for this two-bit chip containing two output pads each gated into 16 lines through data latches. Each data latch is controlled by a read and write strobe.

When two redundant lines are added so that each has an any-for-any replacement capability over the entire bit dimension, they must also have unique data latches, as diagrammed in Fig. 8. Otherwise, each redundant bit line would need multiple bit switches for steering capability to any data latch, causing increased bit line capacitance. As with the "one-bit" redundancy scheme, no steering is done for the store operation. Compare circuits are used in the same way as previously described, but the similarities stop here. The basic concept of this redundancy scheme is not to isolate the defective bit line by deselection of the bit switch, but rather to inhibit the data latch associated with this replaced bit line. The redundant line is then steered to this particular data path with the use of a bit steering circuit.

Not only is it necessary to have the address and enable fuses, but additional fuses called steering fuses must be supplied to each redundant bit line for steering to the proper data pad. When a match occurs between the incoming address and the state of the address fuses, the gate of the redundant bit switch is turned on, allowing for storing or fetching from a redundant bit line. The speed of the compare circuit and redundant decoder must be as fast as the normal decoding.

This redundant "compare" also activates the inhibit read circuitry whose operation is best described as a steering network, as shown in Fig. 9. Each redundant bit line must have its own set of switches.

Thus, either the redundant latch or the regular latch receives a read strobe (switch 1) based upon the state of the "compare" output. In addition, the steering fuses assign that read strobe to one of the two redundant data lines by permanently setting switch 2, which is selected by the state of the steering fuses.

The reason for this additional level of control over the inhibit circuit is to interface the steering fuses with the address fuses so that the read pulse matches the redundant bit line with the proper data pad.

The bit steering circuitry works in combination with the pass devices (shown in Fig. 8), which connect the selected redundant latch to the output data pad chosen by the steering fuse.

The enable operation and enable fuse are incorporated into the steering circuitry so that, when enabled, the circuitry activates one of the two pass devices connected to each redundant data latch. As mentioned, the steering fuses interface with the read strobe circuitry, selecting which redundancy "compare" controls the read operation.

In summary, the fuse information stored contains both the replacement address and the associated data pad number. The fuses that contain the data pad information assign the redundant bit line to a data pad through interaction with the inhibit read strobe circuit and the bit steering circuit. When the address fuses match the incoming

address, a pulse is generated to activate the redundant bit switch and steer the read strobe to the redundant data latch. For a store, both the selected redundant line and the defective line are active. For a fetch, data are transferred to the data pad from the regular data latch or the redundant data latch as a function of whether an address match has or has not occurred.

The total bit system operation depends on how the redundancy controls interface with the regular bit system. We used the two methods described here (the simple one-data pad scheme and the more complex multi-data pad scheme) in our RAM family because

1. Only a small part (the compare circuit) of the redundancy requires high speed operation.
2. The steering circuits can be controlled by dc voltages derived from the state of the fuses.
3. The implementation is easy to expand for any number of bit lines or data pads.
4. There is no impact on performance.
5. No modification to the regular bit system is required.

In general, the choice of a bit redundancy technique depends greatly on the chip organization and functional requirements. With many variations on these general approaches, each application becomes as unique as the chip design incorporating it.

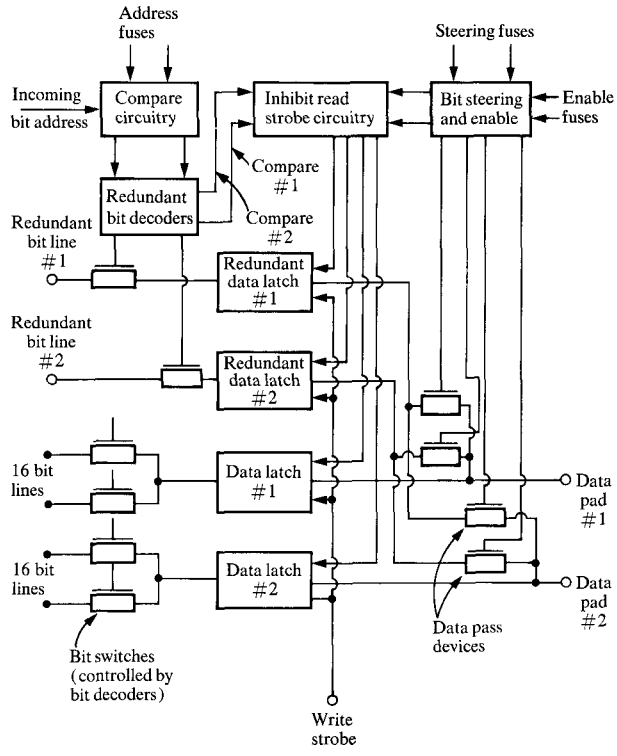

Figure 8 Two-bit chip with two redundant bit lines.

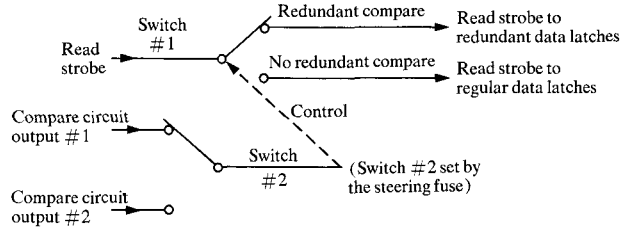

Figure 9 Steering network.

Conclusions

Described here has been a comprehensive look at the implementation of word and bit redundancy for dynamic VLSI memory chips. The addition of optimized redundancy is a powerful circuit technique for increasing the productivity of RAMs. However, redundancy does not have to be confined to memory chips; the circuit techniques employed for memory can be applied to other type designs as well, such as CCDs and logic with extensions for module and card uses. The design approaches presented here for word and bit redundancy are efficient applications covering a wide range of circuit objectives and are the building block concepts for developing an effective redundancy implementation.

Acknowledgment

The authors wish to recognize C. A. Kilmer and D. K. Tewarson for their work in the application of redundancy and for fruitful discussions of the redundancy approaches described in this paper.

References

1. F. E. Sakalay, "Correction of Bad Bits in a Memory Matrix," *IBM Tech. Disclosure Bull.* **6,** 1-2 (1964).
2. F. E. Sakalay, "Memory System for Using Storage Devices Containing Defective Bits," U.S. Patent 3,422,402, U.S. Cl. 340/172.5, January 1969.
3. R. P. Fletcher, "Storage System Using a Storage Device Having Defective Storage Locations," U.S. Patent 3,444,526, U.S. Cl. 340/172.5, May 1969.
4. R. S. Kril, "Memory System," U.S. Patent 3,689,891, U.S. Cl. 340/172.5, Sept. 1972.
5. S. E. Schuster, "Multiple Word/Bit Line Redundancy for Semiconductor Memories," *IEEE J. Solid-State Circuits* **SC-13,** 689-703 (1978).
6. L. M. Arzubi, "Memory System with Temporary or Permanent Substitution of Cells for Defective Cells," U.S. Patent 3,755,791, U.S. Cl. 340/173R, Aug. 1973.
7. B. F. Fitzgerald and D. W. Kemerer, "Memory System with High Performance Word Redundancy," *IBM Tech. Disclosure Bull.* **19,** 1638-1639 (1976).
8. R. R. DeSimone, N. M. Donofrio, B. L. Flur, R. H. Kruggel, H. H. Leung, and R. Schnadt, "FET RAMs," *1979 IEEE ISSCC Digest of Technical Papers* **22,** 154-155 (1979).

9. R. P. Cenker, D. G. Clemons, W. R. Huber, J. B. Petrizzi, F. J. Procyk, and G. M. Trout, "A Fault-Tolerant 64K Dynamic RAM," *1979 IEEE ISSCC Digest of Technical Papers* **22,** 150-151 (1979).
10. Y. Egawa, N. Tsuba, and K. Masuda, "A 1 Mb Full Wafer MOS RAM," *1979 IEEE ISSCC Digest of Technical Papers* **22,** 18-19 (1979).
11. A. Chen, "Redundancy in LSI Memory Array," *IEEE J. Solid-State Circuits* **SC-4,** 291-293 (1969).
12. E. Tammaru and J. B. Angell, "Redundancy for LSI Enhancement," *IEEE J. Solid-State Circuits* **SC-2,** 172-182 (1967).
13. C. H. Stapper, A. N. McLaren, and M. Dreckmann, "Yield Model for Productivity Optimization of VLSI Memory Chips with Redundancy and Partially Good Product," *IBM J. Res. Develop.* **24,** 398-409 (1980, this issue).

Received May 23, 1979; revised November 29, 1979

The authors are located at the IBM General Technology Division laboratory, Essex Junction, Vermont 05452.