ERROR CORRECTION CODING FOR VLSI MEMORIES

Conf. Proc. IEEE, London, Sept. 1980

R.M.F. Goodman

Department of Electronic Engineering, University of Hull.

Summary

This paper reviews the coding techniques used at present to protect data stored in semiconductor RAMs and introduces some new techniques being developed at Hull.

1. Introduction

The advent of cheap mass storage in terms of large semiconductor RAMs has great implications in the field of data communications. This is particularly so for such applications as intelligent terminals, hand-held terminals, personal data systems and data-bases, and personal computer networks. If large memory systems such as these are to be increasingly used in this distributed way, it is essential that they should be reliable, and not require frequent servicing. Although an individual RAM chip may have a reliability figure of better than 10^{-6} failures/hour; when large numbers of these are combined to form a total system the reliability of the system becomes exponentially worse. In these cases it is essential that some form of error correction coding (ECC) be used to protect against data loss.

LSI dynamic RAMs generally have a low failure rate of the order of 10^{-7} to 10^{-6} failures/hour. In addition, there is a 'learning-curve' for devices as they appear on the market, which means that the larger RAMs have a lower reliability than the smaller devices. The current industry standard 16K by 1 bit dynamic RAM has a failure rate of approximately 3×10^{-7} failures/hour[1], while the 64K RAMs just appearing will initially have a failure rate much worse than this. Thus although system reliability is increased by using larger RAMs, the need for ECC remains. The ECC systems used in memories are simple[2] when compared with the coding used on data transmission links such as space probes, but there are good reasons for this. Firstly, any ECC system will increase the access time or the cycle time, or both, of the memory. The simpler the system - the smaller this delay. Secondly, errors do not occur, and then pass, as in a data link. The predominant failure mode within a RAM chip is a 'stuck-at' fault. In this mode either, an individual cell, or a whole row or column within the X,Y memory array, appears to be stuck at a particularly value, 0 or 1, on read. Alternatively, the chip can catastrophically fail, and every location appears 'stuck'. Therefore errors are stationary in time, and this can greatly simplify the ECC system needed.

2. The need for ECC

Consider a 4 megaword semiconductor RAM system operating with a 16 bit microcomputer. Assuming that the memory is built out of industry standard 4116 type 16K by 1 bit dynamic RAMs, then the system takes the form of an array with 16 columns, C = 4096/16 = 256 chip rows, and D = 4096 devices, as in fig. 1. Given a device failure rate of $\lambda = 3 \times 10^{-7}$, the system mean time between failure is MTBF = $1/\lambda D$ = 813 hours or approximately one month. Expressed another way, we may define the reliability (R) of a single device as the probability of a single device operating correctly at time t. Thus $R = e^{-\lambda t}$, assuming constant failure rates,[3] and the probability of device failure is (1-R). The probability of an uncoded memory system operating correctly is the probability that all D devices operate correctly, that is, R^D. The probability of system failure at time t is then $(1-R^D)$. Applying this to the previous example we find that the probability of system failure at the 48 hour point is 6%. This is clearly unacceptable for many applications.

3. Reliability improvement by using Hamming Codes

Figure 2 shows a typical Hamming coded memory. The Hamming codes are a class of single error correcting codes of block length n bits, made up of k data bits, and (n-k) check bits. The Hamming distance of these codes is 3 and therefore a single error in any chip row of the memory can be corrected. The memory is organised in a C(rows) by n (columns) array. The data is stored in the C by k array of chips whilst the remaining C by (n-k) array holds the code parity checks. The redundancy is (n-k)/n, and $2^{(n-k)}-1 \geqslant n$. The system operates as follows. The k data bits are read into the memory from the computer bus and simultaneously into the parity check generator which is a simple logic circuit. The (n-k) checks are stored in the memory array. When the memory is read, the read data is again fed into a parity check generator which recalculates the checks. The bit by bit exclusive OR modulo-2 sum of the stored check bits and the recalculated check bits is known as the syndrome. The syndrome has the property that there is a one-to-one relation between syndromes and correctable error patterns. The syndrome is converted to a fault location word which has a '1' in the position in which the error is estimated to be. Exclusive OR-ing this with the read data will therefore correct the error. The system is essentially transparent to the user and only a small (10's of ns) delay is introduced by the checking logic. The reliability improvement can be estimated as follows. The probability of system failure is the probability of two or more errors occuring in any one chip row. This is simply:

$$P_e = 1 - [R^n + nR^{(n-1)}(1-R)]^C$$

For the example used earlier this gives a 5% probability of memory fail at the 813 hour point. The mean time between failure can be estimated by finding the mission time at which a 50% probability of system failure exists. This is 25,000 hours which is a considerable improvement over the uncoded memory.

The power of the Hamming codes can be extended by the addition of an extra overall parity check. This would give a total of 6 parity checks for the 16 bit data system. This additional check increases the minimum distance of the code to 4 and allows double errors to be detected but not corrected. Thus, when a double error occurs the fault location logic causes an interrupt to the processor which then enters an error handling routine. The double error can then be corrected by error analysis.

4. More powerful coding schemes

In order to correct more than one error per chip row it is necessary to increase the power of the coding scheme. This can be done in several ways. Firstly, higher power multiple error correcting codes such as B.C.H. codes can be used. The penalty, however, is more redundancy and more complex (and hence less reliable) error location circuitry. In the case of a 16 bit data width a double error correcting B.C.H. code would require a 26-bit wide memory array, i.e. 38% redundancy.

The second method involves the use of erasure decoding.[4] An erasure is a potentially erroneous symbol with known location. Because most memory failures are 'stuck-at' failures, the location of an error is known and stationary once it has occurred. This information can be used to correct further errors. A system based on the d = 4 Hamming code would operate as follows. When the first error occurs in a chip row, it is located and corrected in the normal manner for a Hamming code. The position of this error is stored in a register for future use. When a second error occurs in the chip row the syndrome indicates that a double error has occurred and the fault location logic uses both the syndrome and the stored error position to compute the position of the new error. The scheme cannot deal with a double error that occurs within the same memory cycle. This however, is a very low probability event and in practice errors occur sequentially with a very large number of memory cycles between successive failures. The erasure method can be extended to multiple error correction because the correction power of a code is bounded by $2t + r < d$, where t is the number of errors and r is the number of erasures. Thus, in principle, a d = 5 code could correct all triple errors using single error correction and double erasure detection.

Another method of dealing with multiple faults is the use of spares switching.[5] In this method a double error causes an interrupt which initiates a spare switching program. The spare then replaces a faulty bit plane and is updated by reconstructing the data. Switching logic then ensures that data is directed to the spare and not to the faulty bit plane.

At Hull the coding research team is working on a self-repairing system which incorporates both erasure decoding and spares switching.[6] However, significant advantages over these methods are achieved, as the spares switching is completely transparent to the user. The system is based on the d = 4 Hamming code, and operates as follows. When the first error occurs it is corrected and its location is stored. Also the automatic bit switching logic is initiated, and transfer of data takes place transparently to the user. Provided no second fault occurs before transfer takes place, this will restore the memory to its original condition apart from the use of one of the spares chips. If a second fault occurs before transfer is complete, then the erasure decoder corrects the double error and transfer can be completed. At the end of this transfer another spare is switched in to take care of the second error, and so on. Thus, the memory always tries to revert to a state in which the correction logic has no errors to correct.

5. Conclusions

ECC is essential in large semiconductor memory systems. The ultimate aim in trying to improve system reliability must be to incorporate on-chip ECC in the new large RAMs. The double benefits of fewer devices in the total memory system, and much higher individual device reliabilities could then be achieved. Also, ECC can be used to improve chip yields by reducing the reject rate on 'slightly' defective RAMs. If cheap mass storage can be achieved in VLSI by these means, the implications for data communications are significant.

6. References

1. Euzent, B.: 'Intel 2116 W-Channel Silicon Gate 16k Dynamic RAM', Reliability Report RR-16, Intel Corporation, 1977.

2. Altnether, J.: 'Error Detecting and Correcting Codes Part 1', Applications Note AP146, Intel Corporation, 1979.

3. Levine, L. and Meyers, W.: 'Semiconductor Memory Reliability with Error Detecting and Correcting Codes', Computer, October 1976, pp. 43-50.

4. Walker, W.K.S., Sundberg, C.W. and Black, C.J.: 'A Reliable Spaceborne Memory with a Single Error and Erasure Correction Scheme', IEEE Transaction on Computers, Vol. C-28, No. 7, July 1979, pp. 493-500.

5. Carter, W.C. and McCarthy, C.E.: 'Implementation of an Experimental Fault-Tolerant Memory System', IEEE Transactions on Computers, Vol. C-25, No. 6, June 1976, pp. 557-567.

6. Goodman, R.M.F.: 'Error Correction Coding for Large Semiconductor Random Access Memories', SRC Grant: GR/B 17785, July 1979.

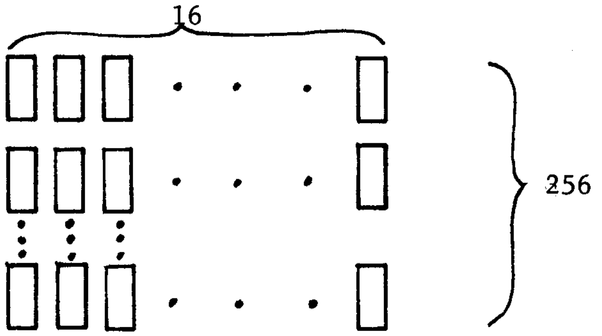

Fig.1.

RAM Organisation

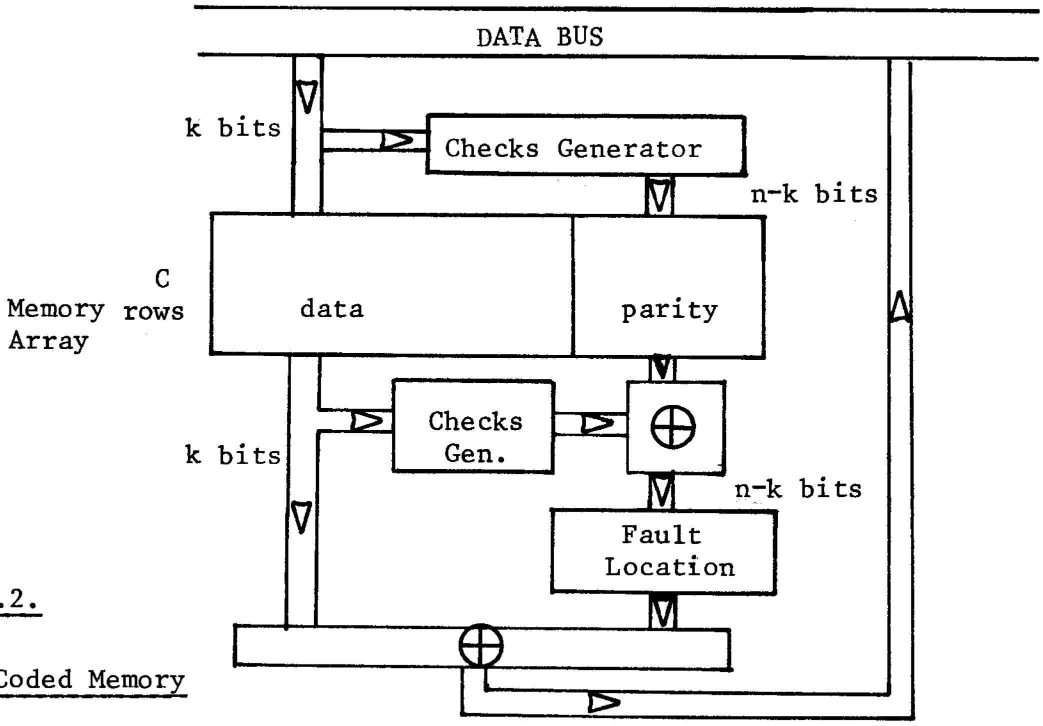

Fig.2.

Hamming Coded Memory