Compressed Bit Fail Maps for Memory Fail Pattern Classification

Jorg Vollrath, Ulf Lederer, Thomas Hladschik
White Oak Semiconductor, Infineon Technologies
vollratj@whiteoaksemi.com, ledereru@infineon.com, hladscht@whiteoaksemi.com

Abstract

This paper presents a new approach to configure compressed bit fail maps to allow fail pattern recognition. Construction of the special compression scheme will be shown. This takes typical memory array fail patterns into account. Examples for different failure types are given. This scheme allows minimizing the necessary catch memory size for fail classification. A 64Mbit fail map can be compressed to 2k allowing classification of 13 fail types. Since catch ram requirements are small, this scheme can be implemented in a manufacturing environment for all processed hardware. Compressed bit fail maps can be used to generate wafer and lot maps for diagnosis.

Introduction

Memory arrays are used for semiconductor technology development. The regularity of the memory array helps to localize process problems. During test each cell is tested for functionality and pass or fail information is stored. This can be done in a vector memory or a bit fail map.

A vector memory stores the address and the failing DQ for each failing read access. If one address is failing a read operation a couple of times, each fail address will be stored as a separate entry. A vector memory can overflow with many fails or long patterns with many read accesses.

A bit fail map is as big as the device under test and stores a failing cell at a certain location determined by the fail address. The bit fail map is a picture of the physical location of the memory cells on the chip /1,2/. A huge array with the size of the memory is generated having '0's for passing cells and '1's for failing cells. This fail memory array is often called catch ram.

The fail pattern of the memory cells points to possible problems in a given process. Pattern classification tools can be used to generate fail paretos of a wafer or a lot for a given process. Looking at bit fail maps gives a much more detailed information of process problems than yields.

As 64Mbit SDRAMs are in mass production and 256Mbit and 1 Gbit RAMs are developed, storing bit fail maps of a single chip, a wafer or a lot becomes more and more difficult, due to storage capacity and processing power issues. A wafer can have up to 1000 chips, a lot has 25 wafers. A 64Mbit product needs 200Gbyte memory requirements for storing bit fail maps. Compression schemes like zip tools can reduce the data size, but data analysis still has to crunch through the original data. A printout of bit fail maps or wafer maps will compress the bit fail map and reduces the visible resolution automatically without user control. Important bit fail map features may be lost. Using conventional methods, spotting characteristic fail signatures gets difficult.

To handle huge bit fail maps a special database and sophisticated software is needed. Other methods to reduce the amount of data are utilizing a vector memory or row and column counters. They try to maintain the necessary information for fail classification /3/. This paper presents a new scheme of bit fail map compression to reduce the data size, with controlled loss of information, and faster data handling and processing. Compression factors up to 100k can be achieved generating still very useful information. Conventional image processing and storage tools can be used for handling these maps, since the size of the bit maps is greatly reduced.

This paper will present first some typical patterns of fail bit maps of an 64M chip and reference these to process problems. Than the new lossy compression scheme is presented. A step by step procedure how to construct the compression scheme is outlined. A list of known fail types is the starting point for developing the new scheme. The number of fail types and their signature lead to the setup for the compression scheme. Some examples will be used to discuss parameters and limitations for the compression. It will be shown how this can be easily implemented in software and hardware.

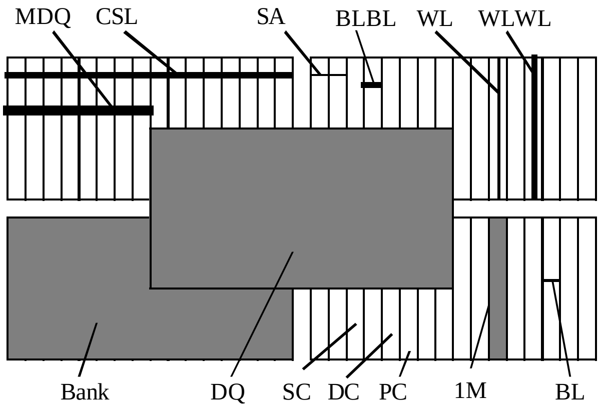

Figure 1: 64MDRAM schematic bit fail map

Bit fail maps and fail patterns

A schematic display bit fail map of a 64M device is shown in Figure 1 as a starting point for the compression scheme. A printed picture is an example for a simple bit fail map compression. Since a printer resolution is limited not all 64 million pixels for a 64M DRAM can be printed for a given picture size. A number of adjacent cells are compressed to one cell depending on the printer resolution.

The 64M bits are represented by a certain number of horizontal cells in x direction, with x display addresses (DX), and vertical cells in y direction, with y display addresses (DY). For a 64M device this can be 16k by 4k cells. To address 16k columns horizontally 14 addresses are necessary DX0..DX13. The 4k rows vertically are addressed by 12 addresses DY0..DY11.

In figure 1 some examples of fails are shown. The basic feature of a certain fail class is the number of failing cells (size) in x and y direction of the fail pattern and the repeat step size. Table 1 gives the fail types and step requirements for a typical DRAM.

The fail types are related to a certain DRAM design. The 64M SDRAM consists of 4 banks with 16 1Mbit blocks each with 512 word lines (WL) and 2048 bit lines (BL). Sense amplifiers (SA) are shared between adjacent 1Mbit blocks. Column select lines (CSL) run over one bank and activate 4 SAs in a 1Mbit block. Master data lines (MDQ) transfer data from the SAs to the outside. Data input and outputs (DQ) are associated with certain memory blocks.

In a simple fail model all these lines can be stuck at a certain value or shorted to the neighboring line. WLs can be open or be shorted with the adjacent WL (WLWL short). Full bit lines (BL) can fail, or can be shorted to the adjacent bit line (BLBL short). Single cells (SC), paired cells (PC), where two cells are shorted and double cells (DC) sharing the same contact can fail.

Table 1: Fail types, size and repeats

	Description	Size		Repeat	
		x	y	x	y
MDQ	Master data line	4k	128	4k	128
CSL	Column select line	8k	4	8k	4
SA	Sense amplifier	1k	1	512	1
WL	Word line	1	2k	1	2k
WLWL	WL/WL short	2	2k	1	2k
Bank	Bank	8k	2k	8k	2k
DQ	Data block	8k	2k	-	-
BL	Bit line	512	1	512	1
BLBL	BLBL short	512	2	512	1
1M	1Mbit block	512	2k	512	2k
SC	single cell	1	1	1	1
DC	Double cell	2	1	1	1
PC	Paired cell	1	2	1	1

A single failing cell is one cell wide in x and y direction. A bit line has 512 cells in x direction and one cell in y direction.
Since every fail type has a process root cause associated with it, it is easy to estimate process problems by simply looking at the number of fails per die or wafer for a given lot. This allows easy process monitoring and improvement. Table 2 shows a typical relation ship between fail type and process problem.

Table 2: Fail types and typical process root causes

	Description	Process problem
MDQ	Master data line	Metal 1 shorts
CSL	Column select line	Metal 2 open/short
SA	Sense amplifier	Contact open
WL	Word line	Metal 1 open
WLWL	WL/WL short	Metal 1 short
	Bank	Control circuit problem
DQ	Data block	Pad problem
BL	Bit line	Contact open
BLBL	BLBL short	Metal 0 short
1M	1Mbit block	Control circuit problem
SC	single cell	Dielectric breakdown
DC	Double cell	Contact problem
PC	Paired cell	Capacitance short

The position of the failure in the bit fail map gives sufficient information for physical failure analysis to deprocess the device and find the defect. Fail patterns and their relationship to the process give the opportunity for analyzing split lots and process changes looking at signature analysis of bit fail maps.

Comparing the fail types of table 1 with the schematic bit map of figure 1 already allows classification of certain fail types. A failing bank can be seen. SA fails can be distinguished from BL or WL fails. Other fail types as BLBL shorts and a BL, WLWL shorts and a WL and a SC, DC and PC are not possible to distinguish. In the next section this issue is addressed. A closer look at compression schemes to maintain certain features of the fail bit map will be taken.

The list of fail types is the basis to develop a new compression scheme.

General compression scheme

The new compression scheme is derived from the compression during the print out of a bit fail map. Looking at Figure 2, 16k pixels horizontally and 4k pixels vertically are compressed into much less pixels. For this example lets assume it is 1k by 256 pixels.

This compression can be achieved by looking at an area of 16 by 16 pixels and determining whether to set one pixel of the compressed map. If any of the 16x16 pixels is set the pixel in the compressed map will be set. This can also be expressed by looking at an address equation between the compressed addresses (CX, CY) and the display addresses (DX, DY) (Figure 2). Since compression takes place, less addresses for the compressed map are needed.

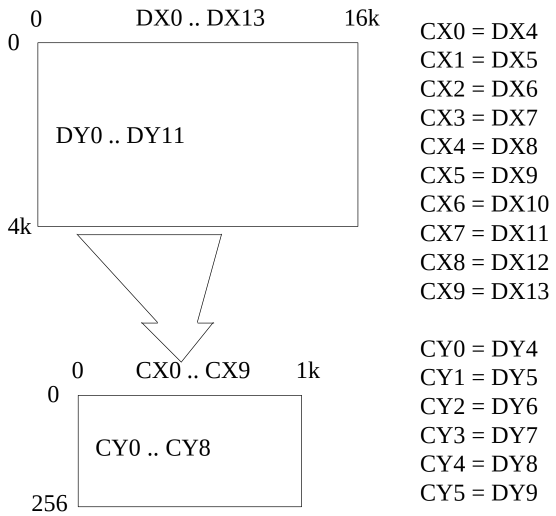

Figure 2: General compression scheme

To transfer the original map to the compressed map each address for the original map is generated. If a fail is encountered the address is transformed to the compressed address with the equations of figure 2. A fail bit is then set in the compressed map. This will set one fail in the compressed map if any of 16 x 16 = 256 bits is failing in the original map. Another way to describe the compression scheme is to drop the low order addresses.

To get a compressed bit map for fail signature analysis the equations for CX0..9 and CY0..5 have to be changed according to the required recognized fail type. They have to be mapped to different DX, DY addresses. Depending on the number of fail types and the compression ratio this can be quite tricky. The following section describes an approach to construct a minimum compressed bit fail map for the above mentioned fail types doing a bottom up approach.

Development of a minimum compressed bit fail map

Construction of the compressed bit fail map can start with any fail type. For this example DQ fails and bank fails are considered first. The central shaded area of figure 1 denotes the physical region where one DQ is failing. Figure 3 shows the 4 regions of the four DQs and a basic compressed map. Assigning each area one pixel generates a 4 by 4 pixel compressed bit map. The equations for the compression are:

CX0=DX12; CX1=DX13; CY0=DY8; CY1=DY9.

A failing DQ is setting 4 pixels of one DQ in the compressed map. A bank would fail if 2 by 2 pixels are set.

Putting in this map a failing DQ and a failing

DQ2	DQ1	DQ1	DQ2
DQ3	DQ0	DQ0	DQ3
DQ3	DQ0	DQ0	DQ3
DQ2	DQ1	DQ1	DQ2

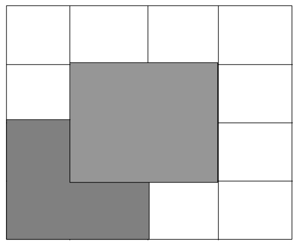

Figure 3: DQ regions

bank the overlap of the fails can be seen. One pixel failing the DQ is also in the failing bank. This shows some basic limitations of a compressed map, which will be discussed later.

The compressed bit map configuration can be checked for the impact of other fail types. A Failing CSL sets two horizontal adjacent pixels in the map. A 1M, WL or WLWL short will set two vertical pixels. BLs, BLBL shorts, MDQs, SCs, PCs and DCs will set one single pixel each. Summarizing the basic map allows classifying 4 different fail types: DQ, Bank, CSL and other. To distinguish the other fail categories this map has to be refined.

Considering a 1M segment fail, the WL and the WLWL short each pixel has to be expanded into 4 horizontal pixels resulting in a 16 by 4 map. Since a 1M segment is 512 wide and there are 64 1M segments DX9, DX10, DX11, DX12, DX13 and DY9 are deciding which segment is addressed. A lower x address (DX0..DX8), where all pixels are failing if a 1M segment fail occurs is needed for the compressed map. A 1M segment fail shows then up as 4 by 2 pixels. Choosing DX0 as one of the addresses gives 1 by 2 pixels for a failing WL and 2 by 2 pixels for a WLWL short (Figure 4). The equations for this compression are:

CX0=DX0; CX1=DX1; CX2=DX12; CX3=DX13; CY0=DY8; CY1=DY9.

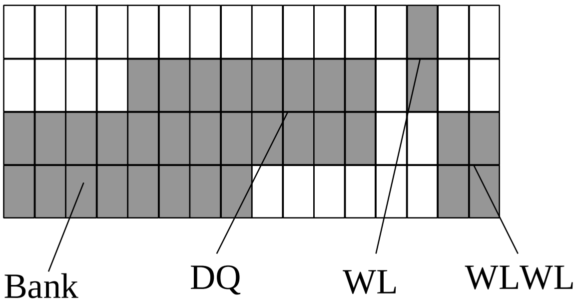

Figure 4: A 16 by 4 compressed map with a bank fail, a DQ fail, a WL fail and a WLWL fail.

Now the bank fail can also be interpreted as two 1M segment fails or a couple of WL fails. Signature analysis is more difficult for this case. On the other hand one failing WL represented by 1 by 2 pixels in the compressed map is in the original map 1k by 1k pixels big and can hide other WLs, BLs and SCs.

Again this map can be checked for the other failure types:

MDQs , BL, BLBL, SA will each set 4 by 1 pixels. DC will set 2 by 1 pixels and SCs and PCs will set 1 pixel.

To distinguish SC and DC fails the map has to be expanded by the lowest y address DY0. Then BL and SA fails are different from the MDQ and BLBL fail. The x address DX9 for adjacent 1M Blocks has to be taken to distinguish between the BL and SA. Another y address (DY1..4) takes care of different shapes for MDQ and BLBL fails.

This gives the following equation for a 32 by 16 pixel compressed map:

CX0=DX0; CX1=DX1; CX2=DX9; CX3=DX12; CX4=DX13; CY0=DY0; CY1= DY4; CY2=DY8; CY3=DY9.

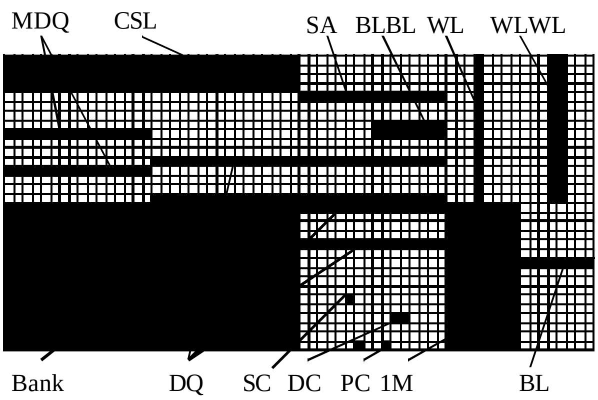

Figure 5: Compressed bit fail map 64x32 pixels

Figure 5 shows a 64 by 32 pixel representation of the bit fail map with the according fails and the following equation:

CX0=DX0; CX1=DX1; CX2=DX3; CX3=DX9; CX4=DX12; CX5=DX13; CY0=DY0; CY1=DY3; CY2= DY4; CY3=DY8; CY4=DY9.

The different fail types are marked as black areas. The grid shows each single pixel of the compressed map. The shape of the fail and the dimension makes fail classification possible. Since the compressed bit fail map is small a simple pattern recognition program can be written to classify the fail types. Table 3 gives the x and y dimension of each fail type.

Table 3: x and y dimension of each fail type in a compressed 64 by 32 bit map

Failtype	x	y
Bank	32	16
DQ	16	1
1MSeg	8	16
MDQ	16	1
CSL	32	4
WLWL	2	16
WL	1	16
SA	16	1
BLBL	8	2
BL	8	1
DC	2	1
PC	1	2
SC	1	1

There are a lot of benefits of a compressed map. 64Mbit can be compressed to 64*32 = 2kBit. This is a tremendous saving by a factor of 32k. A pattern recognition algorithm can be implemented a lot easier and runs a lot faster than on original 64Mbit maps. Storage and automation of the analysis of wafers and whole lots becomes feasible /4/.

An example using a compressed bit fail map

The above mentioned scheme was used to evaluate the impact of a 12 s high voltage stress on a 64Mbit SDRAM on wafer level. A compressed bit fail map of a functional test of 600 chips of one wafer was done first, then a 4V voltage stress in a march pattern was applied to the device for 12 seconds. Then a second compressed bit fail map was taken. The compressed bit fail map before stress was subtracted from the compressed bit fail map after stress to look at the stress induced fails. An automatic pattern recognition program extracted the mean of the number of fails per type and per die as shown in figure 6. The SC signature matched the electrical analysis showing an increase in the off current of the actual device resulting in a single cell fail. A correct classification of fails a single cells was possible with 1kbit compressed bit fail maps.

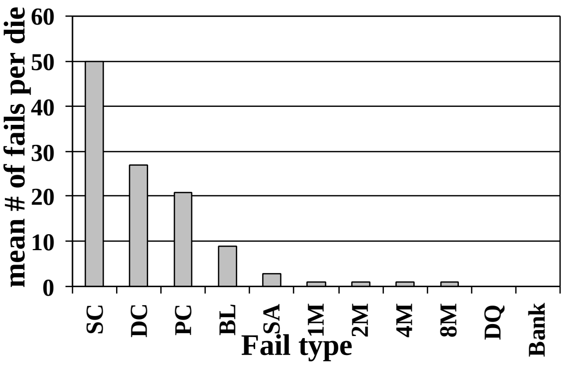

Figure 6: Stress induced fail analysis of compressed bit fail maps

The next section deals with features and limitations of the compression.

Discussion of features and limitations of compression

The parameter of the compressed bit fail map is the chosen compression factor. There is a minimum size for a given number of fail classes. In the above example it is 64 by 32 = 2048 bits. Another limitation is the fail density of a memory device. The number of fails depends on the defect density. If the defect density is to high all pixels will be set for the compressed map. No fail signature classification will be possible. Restrictions of the fail classification algorithm give the maximum number of defects allowed. A fail classification can require all adjacent cells to be pass. The defect density is then limited to $1/4^{th}$ of the compressed map for a single cell.

To deal with high failure densities two strategies are available. First the bit fail map can be expanded to a bigger size. This has the drawback of more storage space requirements. The second option is to use only a subset of the complete memory. Addresses, which are not used by the compression scheme, can be fixed at one value. In the above example fails can be counted only, if DX5 and DX6 are 1. A quarter of fails on a single chip are counted. This is additional information loss, but helps to classify the remaining failures.

During fail classification care must be taken. A huge fail can mask smaller fails. A bank fail will mask a quarter of the chip. All smaller sized fails like SCs, BLs and WLs will be masked by the failing bank. For every bank fail, the number of SCs, BLs and WLs failing has to be scaled up. This procedure results in more accurate fail classification.

A comparison of this scheme to a vector memory can be made. A vector memory stores address and data bits for every fail event. A 64Mbit RAM organized by 8 needs 8 bits for the data and 23 bits for the address. A single cell fail needs then 31 bits. A WL is 1k long and needs 31k. A CSL needs 1Mbits. A bit line is 512 bits long and needs 16kbits. All these memory requirements are bigger than the 2k used for the compressed map. Most of the fail types generate a lot of vector data. Vector memories can be quite large, but data analysis and fail classification can be quite painful. Fail classification is easy with compressed maps. A fail signature analysis tool can compare a given pattern with the compressed bit map. A single cell fail would use a pattern where one fail is surrounded by passing cells. This pattern is shifted in the above mentioned step size over the whole array.

Software and hardware implementation

A simple software implementation will generate sequentially all addresses for the original bit fail map and read out pass fail information. If a fail is read, the address will be converted with the above equations to the address of the compressed map. A fail will be set at this location.

A hardware implementation can be directly done at the memory tester. The memory tester will generate the addresses anyway. These addresses have to be converted to the compressed addresses and fails have to be stored. This can be done via a switch matrix or a scrambler. Additional address lines for the memory device will be dropped. Figure 7 shows a typical setup. Most memory testers have this set up already built in. Only the right scrambling configuration is needed.

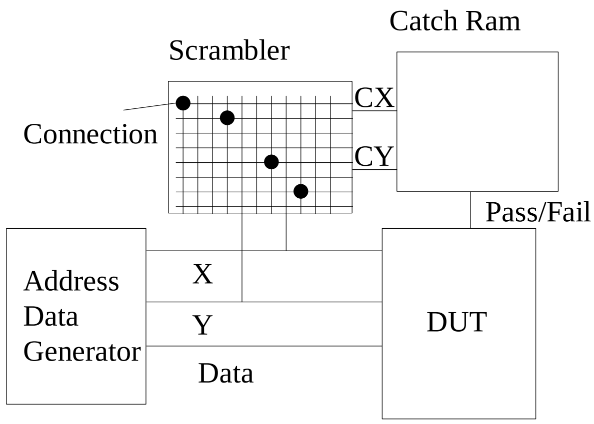

Figure 7: A compressed bit map set up

Conclusion

This paper presented a simple approach to generate hardware compressed bit fail map for signature analysis in a production environment. The procedure to obtain the mapping equations optimized according to fail types was outlined. This makes it possible to look for SC fails even in compressed bit fail maps. The minimum size of a compressed bit fail map was discussed. An example of a signature analysis was presented. Options to improve the compression scheme for high defect density cases were mentioned. This information can be used to construct a robust diagnosis tool for process monitoring and further product analysis.

References

/1/ Microelectronics Manufacturing Diagnostics Handbook, Abraham L. Landzberg, Van Nostrand Reinhold, New York, 1992

/2/ Integrated Circuit Manufacturability, J. P. de Gyvez, Dhiraj K. Pradhan, IEEE press,1998

/3/ Semiconductor Manufacturing Process Monitoring Using Built-In Self-Test for Embedded Memories, I. Schanstra et al., Proceedings of the International Test Conference, Washington DC, 1998, pp. 872-881

/4/ Tutorial: SDRAM Characterization, Joerg Vollrath, Proceedings of the MTDT 1999, San Jose, CA, pp. 62-69