On The Yield of Compiler-based eSRAMs

X. Wang[1], M. Ottavi[2], F. Meyer[3] and F. Lombardi[2]

[1] IBM Corp, Essex Junction (VT) USA.
[2] ECE Department, Northeastern University Boston (MA) USA.
[3] ECE Department, Wichita State University Wichita (KS) USA.
E-mails: {lombardi, mottavi, xiawang}@ece.neu.edu; fred.meyer@wichita.edu

Abstract

This paper presents an extensive evaluation of the manufacturing yield of embedded SRAMs (eSRAM) which are designed using a memory compiler. The yield is evaluated by considering the different design constructs (generally referred to as kernels) that are used in defining the memory architecture through a compiler. Architectural considerations such as array size and line (word and bit) organization are analyzed. Compiler-based features of different kernels (such as required for decoding) are also treated in detail. An extensive evaluation of the provided redundancy (row, column and combined) is pursued to characterize its impact on the memory yield. Industrial data is used in the evaluation and an industrial ASIC chip (made of multiple eSRAMs) is also considered as design case.

1: Introduction

Today's Integrated Circuits (ICs) rely on efficient design techniques which allow manufacturing of complex digital systems. For cost-effectiveness the yield (i.e. the percentage of working or fault free chips in a batch) is commonly used as figure of merit for IC manufacturing. While dynamic array configurations are possible, the more common memory type remains the static RAM (SRAM). SRAMs are typically embedded in SoC and ASIC chips in large numbers; today, it is not difficult to find large ASIC chips or SoCs with 30 embedded memory arrays (which occupy more than 60% of the chip area). For ASIC or SoC, a compiler [11] is commonly employed in the design and organization of the embedded memory module(s); this tool provides flexibility and versatility in design options at both array and chip levels such that a high yield can be retained under different defect distributions. Repair facilities are usually provided to enhance the yield in the presence of defects and faults. The addition of redundancy to replace faulty resources has been proven to be effective in improving the yield of integrated circuits [1]. [2] has presented a detailed analysis of the yield of embedded static random access memories (eSRAM) which are generated using a compiler. A compiler-based memory array is usually made of so-called kernels. Kernels are pre-designed modules (inclusive of layout) which can be integrated onto a chip such as a SoC. Manufacturing of compiler-based memories is a complicated process because it requires to consider the several levels (layers) of the chip. The method of [2] analyzes the critical area of each kernel of the eSRAMs; from the critical area of the kernels and the compiler option, it is then possible to obtain the total critical area. This together with the calculated defect density allows to find the number of different fault types of a configuration of the eSRAM. Defect and fault analysis inclusive of industrial data have been presented [2] for these chips by taking into account the design constructs (referred to as kernels) and the physical properties of the layout. A new tool referred to as CAYA (Compiler-based Array Yield Analysis) has been proposed; CAYA is based on a characterization of the design process which accounts for fault types and the relation between functional and structural faults.

The objective of this paper is to provide a detailed assessment of the yield of compiler-based eSRAMs by utilizing the design framework of CAYA [2]. The memory yield is evaluated by considering the different design constructs (generally referred to as kernels) that are used in defining the desired architecture through a compiler. Architectural considerations such as array size and line (word and bit) organization are analyzed. Compiler based features for different kernels (such as required for decoding) are also treated in detail. Moreover, as an eSRAM is generated by using kernels, then different kernels have different impact on the yield. For example, the SRAM with a smaller decoder option will have less yield due to the additional number of input/output circuitry required. An extensive evaluation of the provided redundancy (row, column and combined) is pursued to characterize its impact on the overall memory yield. It will be shown that yield loss is more pronounced for bit lines than word lines (column redundancy requires a more complex implementation than word line redundancy because multiplexers and a sense amplifier along the bit line are needed to the output). Throughout this paper, industrial data is used in the evaluation and an industrial ASIC chip (made of multiple eSRAMs) is also considered. Also it will be shown that yield loss increases with the size of the eSRAM arrays, i.e. bigger the size of the eSRAM array, the most improvement in yield is accomplished by using redundancy. For the ASIC chip, this has been accomplished by a greedy assignment of the redundancy to multiple eSRAM arrays, i.e. redundancy is assigned first to the array of largest size.

This paper is organized as follows; Section 2 introduces the design of eSRAMs by a compiler and the tool CAYA [2] which has been developed to describe and characterize the yield environment for these chips. Section 3 presents a detailed array yield analysis by considering different features such as array size, word and bit lines and a compiler kernel (i.e. the decoder option). Section 4 investigates and evaluates the impact of redundancy (and the arrangements possible for the spare rows and columns) on the yield of eSRAMs. Section 5 outlines the eSRAM yield for a case study involving an ASIC chip with multiple memories.

2: Compiler-based Memory Design and CAYA

In this paper, two types of eSRAMs made of different memory cells are considered; these cells (denoted as SRAM1PN and SRAM1PR) are one-port compiler based embedded SRAM (0.13u technology for ASIC). The structures of SRAM1PN and SRAM1PR have many similarities; SRAM1PR has 4 independent redundant (word) lines while SRAM1PN has no redundancy.

For eSRAMs, a compiler-based array consists of several functional modules. The functional modules of the SRAM1PN and SRAM1PR arrays are as follows: memory cells, DIO (data IO), AIO (ABIST IO), local word line driver, global word line driver, timing and decoding control circuits, BIST. ABIST is the normal BIST circuit with the addition of a a controller which is used to completely control the configuration of the BIST hardware (as required) and the test sessions.

Unlike a stand-alone SRAM [3], compiler-based memories have a large number of configurations (for example, SRAM1PN has as many as 14,000 configurations to account for different words, word width and decode options). The memory complier generates the layout of each configuration of the array [12]. The compiler option refers to the capability of specifying a memory configuration. The compiler option for the two cells considered in this paper is denoted as SRAM1PNwXbDdSsM1 (SRAM1PRwXbDdSsM1). Its detailed description is given in Table 1.

Each functional module is built using at least one type of placeable kernels. A placeable kernels defines a pre-designed layout of a circuit for the compiler-based arrays. The memory compiler places the kernels to form the layout of the arrays for a specific configuration. Different configurations of compiler-based arrays have different types and numbers of placeable kernels. Table 2 shows the placeable kernels.

The methodology of [2] is based on the following design process as applicable to compiler-based eSRAMs in today's industry:

1. Specify the desired memory array using the placeable kernels of the modules using the compiler.

Table 1. Compiler option of SRAM1PN and SRAM1PR

SRAM1PN	standard one-port SRAM with no redundancy.
SRAM1PR	standard one-port SRAM with redundancy.
w	5 digits specify the number of words.
b	3 digits, specify the data width of a word in bits.
d	2 digits specify the decoding arrangement.
s	1 digit specifies the number of subarrays
M1	array-clocked timing mode.

Table 2. Kernels for SRAM1PN and SRAM1PR arrays

Kernel	Placeable kernel	Number of kernels
CELL	CELL16_P	#cells=wb #CELL16_P=$wb/16$
DIO	DIO_L_P and DIO_R_P	#DIO=$bd/4$ #DIO_L_P=$bd/4/2$ #DIO_R_P=$bd/4/2$
LWLD	LWLDVR_P	#LWLDVR_P=$(bd/64)(w/d/4)$
GWLD	WLDVR_P	#WLDVR_P=$(w/d/4)$

2. From the compiler option, design and assemble the configuration of the array. Analyze the critical areas of each placeable kernel and the whole array from the layout of the eSRAM.
3. From the layout, determine the numbers and types of possible defects and faults.
4. If the array has no redundancy, use the negative binomial yield model.
5. If redundancy is provided, repair the memory array and calculate the yield according to a new proposed yield model (which accounts for redundancy).

A tool named CAYA (compiler-based array yield analysis) has been developed [2] to facilitate the yield analysis procedure. CAYA performs two functionalities: the first functionality consists of calculating the number of each type of faults in an array for a specified configuration; the second functionality calculates the yield of this array. To perform the first functionality, CAYA is supplied as inputs the critical area of each kernel of the compiler-based array, the defect density and the compiler option for the specified configuration. CAYA generates the number of faults of each type. The critical area of each kernel is already pre-calculated by the critical area extraction tool (i.e. either INCA or CAA which are critical area analysis tools, the first one based on shape expansion and the second on monte-carlo simulation [10]) The defect density is obtained from the manufacturing line, while the compiler option defines the configuration of the array.

To perform the second functionality, CAYA utilizes [2] the number of faults of different type and the redundancy as inputs. The output is the yield of the array with the specified configuration. Redundancy is also supplied through the compiler option. The procedure to calculate the yield of compiler-based arrays as applicable using CAYA utilizes the following steps:

- *Step 1:* Design data is obtained from the layout of the kernels. INCA (or CAA) is executed to find the critical areas. Using the defect density (obtained from the manufacturing line and compiler option for the eSRAM configuration) the number of faults in the eSRAM is established. If the eSRAM has no redundancy, use existing yield techniques (such as the negative binomial model).
- *Step 2:* If the eSRAM has redundancy, then repair of the eSRAM can take place. Also, the number of faults left unrepaired is calculated. Fault types are also taken into account.
- *Step 3:* CAYA is executed to calculate the yield of the eSRAM based on a new model using a linear curve fitting by regression.

The most significant step in CAYA following the analysis of the critical areas and the defect density is to calculate the yield of the configuration of the eSRAM as generated by the compiler option. This is substantially different from traditional methods [4] [5] and [6] to [9]. For compiler-based memories, a more practical (less computational intensive) approach is required due to the large number of configurations possible as well as the inclusion and interface of CAYA with other design tools. In [2], an empirical model based on curve fitting by linear regression is utilized.

3: Array Yield Analysis

In this section, different parameters for the configurations generated by the memory compiler are analyzed in more detail. These are array-level parameters which are directly related to the ability to improve the yield, while changing the design of the eSRAM. The yield of a 1M bit SRAM1PN32768X032D32S2M1 is used as a normalized value of 100.

3.1: Array Size

There can be as many as hundred of eSRAM arrays in an ASIC chip; so redundancy allocation must be carefully allocated to reduce penalties involved in additional area and increased test time.

The addition of redundancy significantly influences memory design. For example, SRAM1PR has 4 redundant word lines per subarray, while SRAM2DR has 2 redundant word lines per subarray. The normalized array yields versus the size of SRAM1PN and SRAM1PR arrays are plotted in Figures 1 (a) and (b).

Not surprisingly by analyzing these Figures, yield loss increases with array size and number of subarrays. The yield difference between arrays (as direct benefit of the provided redundancy) increases also with array size. It has been found that a simple greedy algorithm can be utilized to assign the redundancy to the eSRAM arrays of an ASIC chip, i.e. within the limitation of chip size, assign first the redundancy to the eSRAM of largest size. The detailed description of this algorithm is well beyond the scope of this paper.

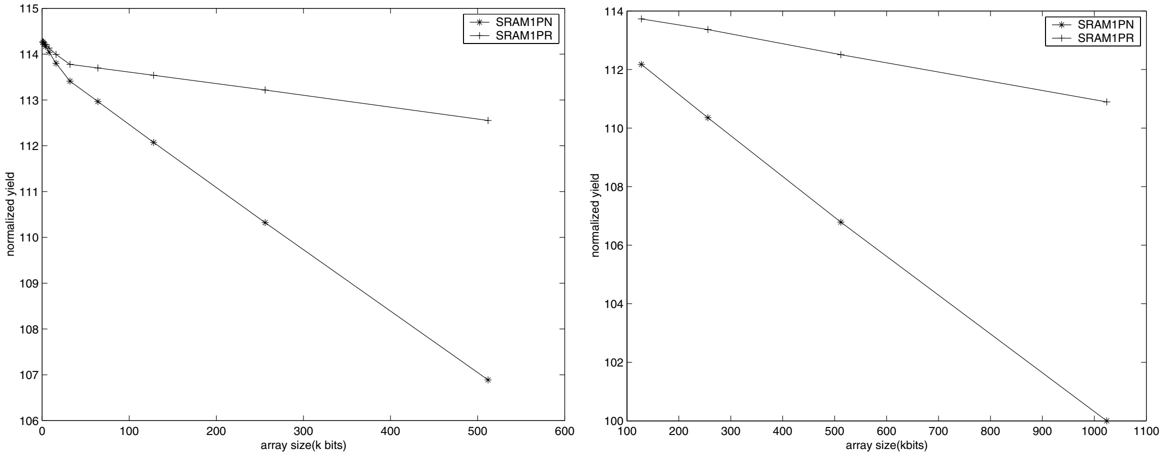

Figure 1. Normalized yield of SRAM1PN and SRAM1PR:(a) 1 subarray (b) 2 subarrays

3.2: Decoder Option

In the compiler, the word width defines the number of output bits of a memory. The decoder option defines the number of bit lines to be decoded into one bit of the output word. For example in

Table 3. 512k bit SRAM decoder option yield impact

Configuration	#words	word width	decoder (d)	#word lines	# bit lines	Norm. yield
Non redundant	16384	32	32	512	1024	106.89
Non redundant	8192	64	16	512	1024	106.93
Non redundant	4096	128	8	512	1024	106.96
4 spare rows	16384	32	32	512	1024	112.58
4 spare rows	8192	64	16	512	1024	112.60
4 spare rows	4096	128	8	512	1024	112.62

SRAM1PN16384X032D16S1M1, the word width is 32 and the decoder option is 16, so 16 bit lines will be decoded into one bit (the total number of bit lines is 16 × 32).

The decoder option in the compiler also affects the yield; Table 3 shows the impact of this option on SRAM1PN and SRAM1PR arrays of fixed size (i.e. 512k bits). An increase in the decoder option (i.e. d) results in a decrease of yield, i.e. if two eSRAM arrays have the same number of word and bit lines, then the SRAM array with a larger decoder has also a larger number of AIOs. So if two SRAM arrays have same critical areas for the common kernels, then the eSRAM array with a larger decoder will account for more critical areas due to the additional AIOs and therefore, a smaller yield will be accomplished.

3.3: Word and Bit Lines

The numbers of word and bit lines have different impact on the yield of eSRAMs. To quantify the yield loss due to bit lines, the number of word lines was fixed to 512 and the decoder option was fixed to 32; the relationship for the yield by increasing the number of bit lines is shown in Figure 2 (a). The slope of this line is S_{bl} while for eSRAMs with a fixed number of bit lines (i.e. 512) and decoder (fixed to 32), Figure 2 (b) shows the plot of the yield as a function of increasing the number of word lines. In this case the slope of the line is S_{wl}.

From Figure 2 (a) and (b) can be seen that the value of S_{bl} is smaller than S_{wl}, i.e. the yield loss due to bit lines is larger than the yield loss due to word lines. This occurs because the critical areas along bit lines are larger than the critical areas along word lines. Since most of the faults are single cell faults, they can be fixed by either using a redundant wordline or column. However, the implementation of column redundancy is harder due to the extra reading and writing circuits such as sense amplifiers and data out latches. Therefore, in the layout, word lines provide a better source of redundancy.

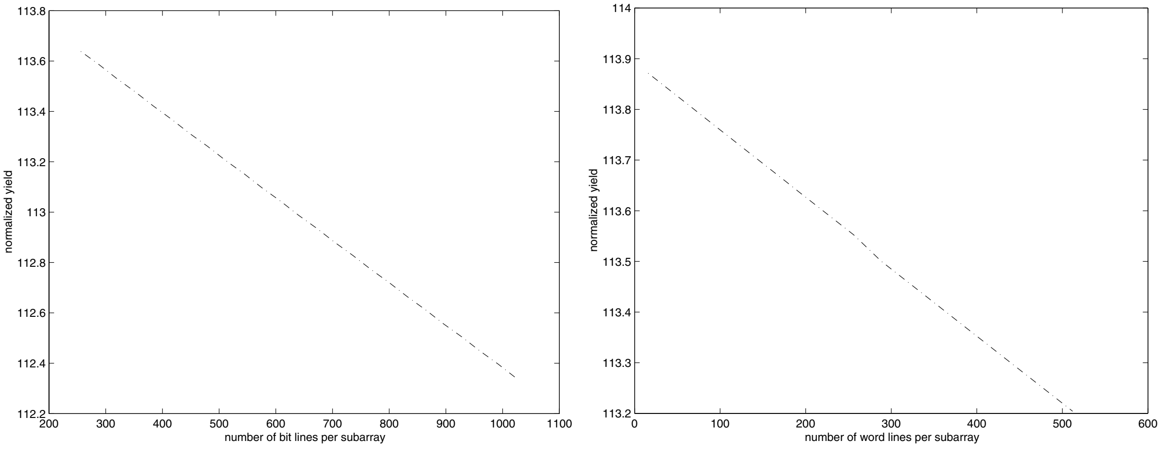

Figure 2. Normalized Yield of SRAM1PN:(a) vs bit lines (b) vs word lines

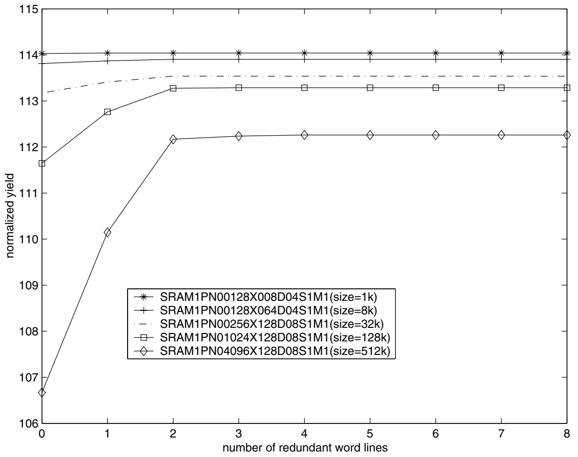

Figure 3. Yield of eSRAMs versus number of redundant word lines

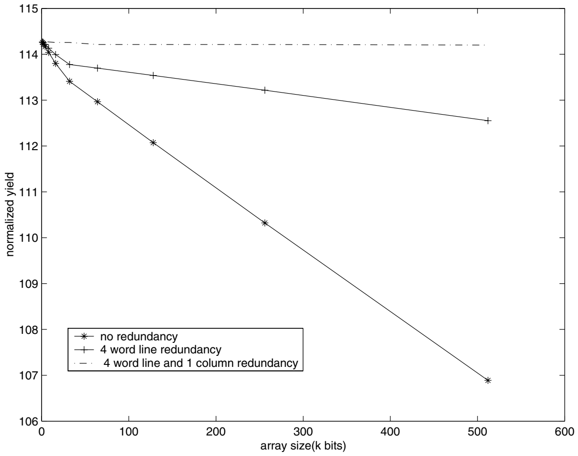

Figure 4. Normalized yield of SRAM1PN with different redundancy schemes

4: Redundancy Analysis

This section provides further results for a better understanding of redundancy resources on the yield of eSRAMs.

Redundant Word Lines

Because the actual redundancy of SRAM1PR are word lines, initially this type of redundant resource will be analyzed. Figure 3 shows the impact of different numbers of redundant word lines on the yield of eSRAMs. It is obvious that redundant word lines have more impact on the yield of large size eSRAMs; two redundant word lines account for the more significant increase in yield due to repair. Redundancy of 3 or more lines seems to be an excessive amount as saturation in yield occurs for all eSRAMs.

Column Redundancy

As discussed in previously, many faults (such as the faults affecting a whole column [2]) can not be repaired by word line redundancy. The addition of column redundancy is more difficult than word line redundancy because a column includes bit lines as well as DIO. The normalized yield of three arrangements is shown in Figure 4 versus array size.

As calculated by CAYA, different types of faults require different redundancy: for a 1024k bits SRAM1PN array 76% of the structural faults can be fixed by word line redundancy; an additional

Table 4. Area overhead of redundancy schemes for a SRAM1PN array

RWL	RCOL	Area overhead (percentage)
4	0	1
3	1	1.3
2	2	1.5
1	3	1.8
0	4	2

Table 5. Characteristics of the eSRAM arrays

Array size (k bits)	Option configuration	#word lines	#bit lines
1	SRAM1PN00128X008D04S1M1	32	32
8	SRAM1PN00128X064D04S1M1	32	256
32	SRAM1PN00256X128D08S1M1	32	1024
128	SRAM1PN01024X128D08S1M1	128	1024
256	SRAM1PN02048X128D08S1M1	256	1024
512	SRAM1PN04096X128D08S1M1	512	1024

21% of the structural faults that affect columns can be fixed by column redundancy. 2% of the structural faults occur in the support circuits and they are not repairable. Assume equal density and distribution in the faults and kernels; under this assumption a 4 to 1 assignment in the number of row/column (word and bit lines) lines provides the best results.

Combined (Row and Column) Redundancy

The impact of the combination of row and column redundancy is now evaluated with respect to the yield by considering the fault density and its relationship to area and kernels. Array yield by employing the redundant schemes of Table 4 is computed. The area overhead has been calculated for SRAM1PN04096X128D08S1M1, i.e. a memory array (single subarray) of 512k bits made of SRAM1PN cells. Note RWL (RCOL) denotes the number of redundant word (column) lines. The characteristics of the eSRAM arrays whose yield is evaluated in Figure 5, are given in Table 5. Figure 5 shows that 3 redundant word lines and 1 redundant column (3RWL+1RCOL) or 2 redundant word lines and 2 redundant columns (2RWL+2RCOL) are the arrangements that result in the highest yield. However, column redundancy occupies more area than word line redundancy and is more difficult to implement on a chip; hence, the best arrangement once area complexity is also taken into account is (3RWL+1RCOL). This is a function of array size too. From Figure 5 , for eSRAMs of small size, the redundancy scheme made of 4RCOL results in a higher yield than the 4RWL redundancy scheme, however for eSRAMs of large size, 4RCOL results in less yield than 4RWL. The reason for this result is that by increasing the size of the eSRAM, the number of word lines (as shown in Table 5 and the percentage of faults occurring on the word lines are also increased. Hence, the provision of word line redundancy greatly affects the yield in a more significant manner than column redundancy.

5: Case Study: an Industrial ASIC Chip

In today's electronic systems an ASIC may integrate more than 100 eSRAM arrays; if there is no redundancy, then a single bit fault in an eSRAM array will cause the whole chip to fail. The addition of redundancy will increase the area of the chip; CAYA can be used to guide a designer in allocating redundancy at chip level. Acp is an industrial ASIC chip. It has 68 eSRAMs of different configurations as listed in Table 6. For example, SRAM2D denotes a two-port eSRAM with no redundancy while SRAM2DR is a two-port eSRAM with two redundant word lines. All arrays in leftmost column of Table 6 have no redundancy.

Same as in previous Tables, the yield of SRAM1PN32768X032D32S2M1 (size of 1M bits) is used

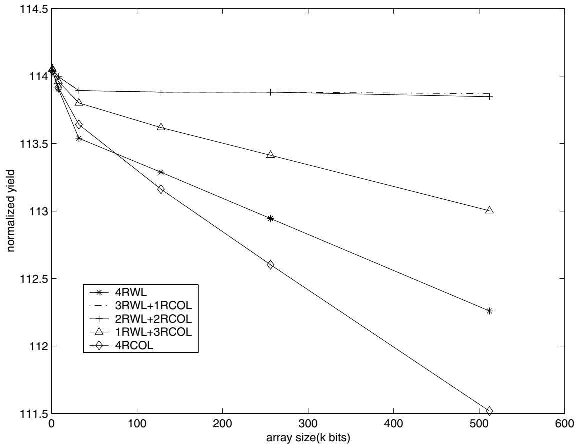

Figure 5. Redundancy schemes for a SRAM1PN array

Table 6. Types and numbers of eSRAM arrays in an ASIC chip and their redundant versions

Configuration	Number of Arrays	Redundant Configuration	Redundant Word Lines
SRAM1PN00128X128D04S1M1	20	SRAM1PR00128X128D04S1M1	4
SRAM1PN00192X107D04S1M1	8	SRAM1PR00192X107D04S1M1	4
SRAM1PN00256X015D04S1M1	2	SRAM1PR00256X015D04S1M1	4
SRAM1PN00512X009D04S1M1	4	SRAM1PR00512X009D04S1M1	4
SRAM1PN01024X009D04S1M1	2	SRAM1PR01024X009D04S1M1	4
SRAM1PN01664X128D04S1M1	1	SRAM1PR01664X128D04S1M1	4
SRAM1PN02048X064D04S1M1	2	SRAM1PR02048X064D04S1M1	4
SRAM1PN02048X071D04S1M1	4	SRAM1PR02048X071D04S1M1	4
SRAM1PN02048X072D04S1M1	2	SRAM1PR02048X072D04S1M1	4
SRAM1PN02048X128D04S1M1	2	SRAM1PR02048X128D04S1M1	4
SRAM1PN04096X008D08S1M1	1	SRAM1PR04096X008D08S1M1	4
SRAM1PN04096X064D16S1M1	6	SRAM1PR04096X064D16S1M1	4
SRAM1PN04096X128D08S1M1	2	SRAM1PR04096X128D08S1M1	4
SRAM2D0256X128D04S1M1	2	SRAM2DR0256X128D04S1M1	2
SRAM2D0384X064D04S1M1	1	SRAM2DR0384X064D04S1M1	2
SRAM2D0640X128D04S1M1	10	SRAM2DR0640X128D04S1M1	2
SRAM2D3840X008D16S1M1	1	SRAM2DR3840X008D16S1M1	2

at a normalized value of 100 %. The ASIC chip prior to introducing redundancy had a normalized yield of 47 % with an array layout area consisting of 16482093 cells (where cell is the basic unit for the technology). If row redundancy is used in all arrays (as shown in Table 6), the normalized yield is increased to 80.7 % while the total area is given by 17651363 cells. This corresponds to an increase of 72 % in normalized yield at a 7.1 % increase in area. As currently the technology does not permit in the compiler to have column redundancy, an estimate was derived for this case. The addition of a redundant column to the redundancy of Table 6, will result in a 219 % normalized yield and a 10 % increase in additional area (both values compared with the case of no redundancy). This shows that the provision of redundancy results in significant benefits in yield of ASIC chips at modest area overhead.

6 Discussion and Conclusion

The following features of redundancy and its implications on eSRAMs can be ascertained from the results of the previous sections.

1) Yield loss increases with the size of the eSRAM arrays, i.e. bigger the size of the eSRAM array, the most improvement in yield is accomplished by using redundancy. For an ASIC chip, this has been accomplished by a greedy assignment of the redundancy to multiple eSRAM arrays, i.e.

redundancy is assigned first to the array of largest size.

2) The choice of a compiler option has also a significant impact on yield. The decoder option has been analyzed in detail. For two SRAM arrays with the same number of bit and word lines; it has been shown that the SRAM with a smaller decoder option will have less yield due to the additional number of AIOs required.

3) In the proposed design, the compiler based array is generated by the kernels. The yield model is a function of the critical areas as per the definition of the kernels. Therefore different kernels have different impact on the yield. For example consider again the decoder option; it has been shown in this paper for small eSRAM arrays, the decoder option is not significant. The decoder option becomes very important for large arrays because memory designs are significantly affected by this feature once the number of (bit and word) lines is increased.

4) It has been shown that yield loss is more pronounced for bit lines than word lines. This occurs because along bit lines, there are DIO and AIO of larger size and more critical area than for the GWLDVR and the LWLDVR.

5) Column redundancy requires a more complex implementation than word line redundancy because multiplexers and a sense amplifier along the bit line are needed to the output. However, column redundancy is necessary for some configurations of eSRAM arrays. For a 1M bits memory array SRAM1PN32768X032D32S2M1 (with 1024 word lines and 1024 bit lines), faults along bit lines account for more than 25% of the density; these faults can not be repaired by word line redundancy. As an extreme case, consider the memory array SRAM1PN00064X256D04S1M1 (with 16 word lines and 1024 bit lines): faults along bit lines account for 80% of the fault density, hence column redundancy is a necessary and also in this case, CAYA provides with excellent facilities to help designers to select an appropriate assignment of redundant resources.

References

[1] I. Koren and Z. Koren, Defect Tolerant VLSI Circuits: Techniques and Yield Analysis, Proceedings of the IEEE, Vol. 86, pp. 1817-1836, Sept. 1998.

[2] X. Wang, M. Ottavi and F. Lombardi. "Yield Analysis of Compiler-based Arrays of Embedded SRAMs" pp. 3-10, *Proc. of IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems*, 2003.

[3] 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.*, vol.24, no.3, pp.398-409, 1980.

[4] R. M. Warner, "Applying a Composite Model to the IC Yield Problem". *IEEE Journal of Solid State Circuits.*, vol.SC-9, no.3, pp.86-95, June 1974.

[5] C. H. Stapper, "On Murphy's Yield Integral". *IEEE Trans. Semiconductor Manufacturing*, vol.4, no.4, pp.294-297, Nov. 1991.

[6] C. H. Stapper, "On yield, fault distributions and clustering of particles." *IBM J. Res. Develop.*, vol.30, no.3, pp.326-338, May, 1986.

[7] C. H. Stapper, "Large-Area Fault Clusters and Fault Tolerance in VLSI Ciruits" *IBM J. Res. Develop.*, vol.33, no.2, pp.162-173, March 1989.

[8] C. H. Stapper, "Small-Area Fault Clusters and Fault Tolerance in VLSI Ciruits" *IBM J. Res. Develop.*, vol.33, no.2, pp.174-177, March 1989.

[9] C. H. Stapper, "Improved Yield Model for fault-Tolerant Memory Chips". *IEEE Tan. on Computers,*vol.42, no.7, pp.872-881, July 1993.

[10] GA Allen, "A Comparison of Efficient Dot Throwing and Shape Shifting Extra Material Critical Area Estimation," *Proc. of IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems* , 1998, pp. 4452.

[11] www-3.ibm.com/chips/products/ asics/products/ememory.html

[12] IBM, on-line document, ASIC Memory Compiler, 2002.

[13] IBM, on-line document of ASIC Dept, 2002.