UNIVERSITY OF ADELAIDE LIBRARY
4.8.89

Fault Tolerance in VLSI

Paul D. Franzon
B. Sc., B. E. (Hons.)

A thesis submitted to the
Department of Electrical and Electronic Engineering, the
University of Adelaide, to meet the requirements for award of the degree of
Doctor of Philosphy.

December 1988.

Abstract

A primary goal in microelectronic systems progress is the achievement of yet higher levels of functional integration. Today this is being addressed from two different perspectives: Firstly, introducing more circuitry onto the chips themselves and; secondly, packaging the chips in higher performance media.

Wafer-Scale Integration (WSI), whereby the chip assumes the size of a wafer, is one goal that combines elements from both of these perspectives. A purely advanced packaging approach is Hybrid Wafer-Scale Integration (HWSI), or silicon on silicon thin film hybrids. Both of these approaches offer many potential advantages, in terms of speed, reliability, power consumption, packaging density and cost. The potential advantages, as well as the disadvantages, are discussed in detail before a review of current WSI and HWSI projects is presented.

Currently one factor that limits chip size growth are the defects incurred in the production of any integrated circuit. Defect tolerance provides the means to overcome this limitation and is particularly important for the achievement of WSI.

A critical point in evaluating approaches to defect tolerance for VLSI, WSI and Ultra Large Scale Integration (ULSI) is the yield model used. A correct yield model allows the type and amount of the optimal level of defect tolerance to be determined. A yield model is presented here that takes account of both clustering and the influence of the reconfigurable interconnect. Two different approaches are presented which would be used for different modeling applications: yield, and expected number of connected processors. The latter form has a number of advantages. The model is applied to a VLSI signal processing chip, and to a generalized chip, to determine the kind of chip structures that can best benefit from defect tolerance. It is found that in order to benefit from defect tolerance regular structures covering more than 20% to 30% of the chip are required. The yield model is also applied in a consideration of granularity effects on wafer-scale arrays. As a result of this discussion on granularity a new metric is suggested for evaluating array element architectures.

Using this model as a basis, a number of alternative approaches to WSI are presented and evaluated. After a review of existing approaches, during which a suitable classification system is introduced, a new approach, called the "frame" approach is introduced. The frame scheme is aimed at the WSI implementation of 2D arrays, containing reasonably large elements. The design and implementation of WSI and HWSI examples of the "frame" scheme are presented. Practical lessons learnt about implementing WSI and HWSI designs are also discussed.

Finally a detailed comparison of different approaches to implementing 2D arrays in WSI is undertaken. The relative merits of the frame scheme are affirmed in this section. Examples are presented that demonstrate the relative advantages and disadvantages of the various approaches, indicating the important points to be considered when designing wafer scale arrays.

Contents

1 Introduction 1

1.1 Research Outline . . . 2

1.2 Advanced Packaging Technologies . . . 4

1.3 Potential Benefits of WSI . . . 8

1.3.1 Size and Weight . . . 8

1.3.2 Reliability . . . 9

1.3.3 Reduced Power Requirements . . . 10

1.3.4 Performance . . . 10

1.3.5 Production Costs . . . 10

1.4 Difficulties to be Overcome to achieve WSI . . . 11

1.5 Hybrid WSI . . . 13

1.6 Three Dimensional Packaging . . . 14

2 World Status of WSI 15

2.1 History of WSI . . . 15

2.2 Major WSI Projects . . . 16

2.2.1 ESPRIT 824 Program . . . 20

2.3 Major Hybrid WSI Projects . . . 24

2.4 Three Dimensional Packaging . . . 28

3 Yield Modeling 34

3.1 Faults in Silicon . . . 34

3.2 Non Fault Tolerant Yield Models . . . 36

3.3 Fault Tolerant Yield Modeling . . . 44

3.4 Fault Tolerant Chip Memories . . . 45

3.5 Yield Modeling for Processor Arrays . . . 46

3.5.1 Yield Formulation . . . 48

3.5.2 Expected Number of PEs . . . 51

3.5.3 Effects of Clustering 52
3.5.4 Comparison with other fault tolerant yield formula 53
3.6 Critical Area Determination 54
3.6.1 Purpose of Critical Area Determination 54
3.6.2 General Methodologies and Related Work 54
3.6.3 Critical Area Determination for the Transform and Filter Brick 57
3.7 Determination of Optimal Redundancy for TFB 57
3.7.1 Fault Tolerant Control Store 60
3.8 Granularity and Defect Tolerant Chips 61
3.9 Granularity in Wafer Scale Arrays 63
3.10 Modeling Interconnect Yield in Reconfigurable Circuits 72
3.11 Place of Yield Simulation in Defect Tolerant Design 80

4 Redundancy Techniques for Wafer Scale Arrays 82
4.1 Redundancy through Reconfiguration 82
4.2 Reconfiguration Technologies 84
4.3 Architectures for WSI 92
4.4 Tradeoffs in Redundancy Schemes 93
4.5 Linear Arrays 94
4.6 Mesh Arrays 98
4.7 Tree Structures 98
4.8 Maximizing the Speed of a Wafer-Scale Array 103
4.9 Conclusions 104

5 'Chip Frame' Scheme for Reconfigurable Mesh-Connected Arrays 106
5.1 Introduction 106
5.2 Reconfiguration Scheme 108
5.2.1 Approaches to Fault Tolerance 108
5.2.1.1 Detailed Scheme Description 115
5.3 Global Reconfiguration in Response to Faults 118
5.3.1 Global Reconfiguration in Response to Faults for 1D Arrays . . 124
5.3.2 Global Reconfiguration in Response to Faults - Mesh Arrays . 125
5.3.2.1 Scheme Utilization 125
5.4 Throughput Rate Optimization 126
5.4.1 RC Model 129
5.4.2 WSI Driver Proposals 131

5.4.2.1 ADVICE Results 133
5.4.3 Discussion of Modeling Results 133
5.4.3.1 Choice of Approach and Area-Time Optimality 134
5.4.3.2 Power Dissipation 135
5.4.4 Conclusions 137
5.5 Fault Detection and Isolation 137
5.6 Implementation 138
5.7 Summary 139

6 Comparing Redundancy Approaches for Mesh Arrays 152
6.1 Comparison Metrics 152
6.2 Classification of Mesh Arrays 154
6.3 Comparison of Mesh Redundancy Approaches 166
6.3.1 Monte Carlo Simulation 166
6.3.2 Utilization Comparisons 170
6.3.2.1 Effect of array size 178
6.3.2.2 Effect of Clustering 180
6.3.3 Speed Comparisons 187
6.4 Overall Comparison of Different Scheme Samples 196
6.5 Improved Schemes 211

7 Conclusions 214
7.1 Defect Tolerance and Advanced Packaging 214
7.2 Yield Modeling for Fault Tolerant VLSI and WSI 215
7.3 Making WSI and HWSI Parts Using the Frame Scheme 216
7.4 Comparison Studies of Fault Tolerant Mesh Arrays 217
7.5 Future Work 220

A Delay Models 221

B Published Papers 224

List of Figures

1.1 Substrate efficiency 5

1.2 Cost-density relationships for different interconnect media. 6

1.3 Broad brush comparison of ($US) costs for a WSI based product compared with the equivalent discrete die and board based product. . . . 12

2.1 Cross section of Mosaic's silicon circuit board 24

2.2 Microbridge connector structure developed by Hughes. 29

2.3 Basic structure of Hughes 3D computer 30

2.4 Through the wafer free space optical interconnect 31

2.5 Integrating technologies to form a wafer stack 32

2.6 A future mesh-connected multicomputer 33

3.1 Reliability failure rate bathtub curve. 37

3.2 Example of defect clustering on a wafer 39

3.3 Examples of photolithographic defects 55

3.4 Defect Size Distribution 56

3.5 Floorplan for the Transform and Filter Brick 58

3.6 Figures of Merit (FM) vs. the number of spare columns and rows in the control store, for the whole chip with parameters: $D_0 = 0.10$ defects/mm^2. $\alpha = 2$. $A_{cell} = 0.55A_{mem}$, $A_{col} = 0.19A_{mem}$, $A_{row} = 0.25A_{mem}$. 62

3.7 Figure of Merit FM vs. replicated element size, and the % of the chip area covered by the array containing this element. ($D_0 = 0.02$ defects/mm^2). 64

3.8 Figure of Merit FM vs. replicated element size, and the % of the chip area covered by the array containing this element. ($D_0 = 0.02$ defects/mm^2 and larger element size range). 65

3.9 Figure of Merit FM vs. replicated element size, and the % of the chip area covered by the array containing this element. (D_0 = 0.05 defects/mm^2). 66
3.10 Figure of Merit FM vs. replicated element size, and the % of the chip area covered by the array containing this element. (D_0 = 0.05 defects/mm^2 and larger element size range). 67
3.11 Figure of Merit FM vs. replicated element size, and the % of the chip area covered by the array containing this element. (D_0 = 0.10 defects/mm^2). 68
3.12 Figure of Merit FM vs. replicated element size, and the % of the chip area covered by the array containing this element. (D_0 = 0.10 defects/mm^2 and larger element size range). 69
3.13 Actual area required to achieve optimal Figures of Merit for D_0 = 0.05 defects/mm^2 case. 70
3.14 Effect of partitioning on the FM: Expanding area, $A_{PE} = 50$ mm^2 . . 73
3.15 Effect of partitioning on the FM: Constant area, $A_{PE} = 50$ mm^2 . . . 73
3.16 Effect of partitioning on the FM: Expanding area, $A_{PE} = 10$ mm^2 . . 74
3.17 Effect of partitioning on the FM: Constant area, $A_{PE} = 10$ mm^2 . . . 74
3.18 Mesh array reconfiguration scheme showing (a) all the wiring, and (b) an example with faults. 78
3.19 Contribution of wiring areas to yield areas. 79

4.1 TMR approach to implementing a fault tolerant array. 83
4.2 A temporal redundancy approach to implementing a fault tolerant array. 85
4.3 Example of a global redundancy approach to implementing a fault tolerant array. 86
4.4 Comparison of utilization for different redundancy approaches. 87
4.5 Two approaches to using fuse/join techniques. 89
4.6 Comparison of fuse only and fuse/join techniques for linear arrays. . . 90
4.7 Range of architecture topologies and their suitability for WSI 92
4.8 Taxonomy of methods used to form fault tolerant linear arrays. 96
4.9 Examples of LI chain schemes. 97
4.10 Chip usage of LI linear schemes. 97
4.11 Taxonomy of approaches to interconnect reconfiguration of mesh arrays. 99
4.12 First level redundancy approach to implementing fault tolerant trees. . 100
4.13 Level redundancy on one (PE) level only. 101

4.14 SOFT approach to fault tolerant trees 102
4.15 Bussed sparing approach to tree fault-tolerance. 102
4.16 Pipelining the interconnect can be used to maintain throughput. . . . 104

5.1 FIR Filter Section with Fixed, Preloaded Coefficients for Testbed Array 107
5.2 Row-oriented reconfiguration (with pass-transistor based switches) . . 109
5.3 Basic Outline of the Reconfiguration Scheme. 111
5.4 Mesh-Connected Cell Array with reconfiguration Frame per Cell . . . 112
5.5 Examples of some reconfigured frames. 113
5.6 Fault Tolerant Power Distribution. 114
5.7 Sea-of-switches reconfiguration 114
5.8 Programmable Control of Reconfiguration State. 116
5.9 Programming the Wafer-Scale Array. 117
5.10 Allowed vertical bus configurations (Figure 1 of 4). 119
5.11 Allowed vertical bus configurations (Figure 2 of 4). 120
5.12 Allowed vertical bus configurations (Figure 3 of 4). 121
5.13 Allowed vertical bus configurations (Figure 4 of 4). 122
5.14 Internal control signals. 123
5.15 Example of a configured linear array. 124
5.16 Example of a configured mesh connected array. 126
5.17 Utilization vs. Cell Fault Probability for Frame and Other Schemes Above. 127
5.18 Unbuffered and Various Buffered Frame Designs. 128
5.19 A length of reconfigurable interconnect and its equivalent RC model. . 130
5.20 Modified circuit model for a gate with a rising input. 131
5.21 Derivation of worst case delay (Case (a)) for 1 PE bypass. 132
5.22 [Next Page] Mask-Level Schematic of Monolithic WSI Array Site. . . . 140
5.23 [Page after next] Microphotograph of fabricated wafers showing WSI and HWSI halves. The empty space and additional chip sites seen are explained in the text. 140
5.24 Speed vs. driver and transmission gate size. Case (1) Non buffered. Simple RC model. 141
5.25 Speed vs. driver and transmission gate size. Case (1) Non buffered. Complex RC model. 142
5.26 Speed vs. driver and transmission gate size. Case (2) Bypass Drivers. Simple RC model. 143

5.27 Speed vs. driver and transmission gate size. Case (2) Bypass drivers. Complex RC model. 144
5.28 Speed vs. driver and transmission gate size. Case (3) Inter-PE buffers. Simple RC model. 145
5.29 Speed vs. driver and transmission gate size. Case (3) Inter-PE buffers. Complex RC model. 146
5.30 Speed vs. driver and transmission gate size. Case (4) Fully buffered. Simple RC model. 147
5.31 Speed vs. driver and transmission gate size. Case (4) Fully buffered. Complex RC model. 148
5.32 Speed vs. driver and transmission gate size. Case (1) Non-buffered. ADVICE results. 149
5.33 Speed vs. driver and transmission gate size. Case (2) Bypass drivers. ADVICE results. 150
5.34 Speed vs. driver and transmission gate size. Case (3) Inter-PE buffers. ADVICE results. 151

6.1 Taxonomy of approaches to interconnect reconfiguration of mesh arrays. 155
6.2 Example of a reconfiguration scheme using row bypass. 156
6.3 Achieving bypass using connecting elements. 157
6.4 Approaches to bypassing within rows, combined with column steering. 159
6.5 An example of a 2-D perturbation scheme based on LI and CE switches. 160
6.6 Interstitial redundancy scheme using LI connections only. 161
6.7 Schematic Layout of direct-reconfiguration structure. (reproduced from [Sami and Stefanelli, 1986] (c)IEEE) 161
6.8 Possible logical connections to a cell, showing employment of multiplexor structures. 162
6.9 Neighbour connections allowed using switched LI connections 164
6.10 Channel approach to fault tolerant mesh arrays. 165
6.11 Example of a reconfigured array using the channel approach (single channel width.) 166
6.12 Example of a reconfigured array using the channel approach (double channel width.) 167
6.13 Frame approach to all-GI connected array reconfiguration. 168
6.14 Example of a reconfigured frame approach array. 169
6.15 Comparison of the in-line row schemes and a combined CE/LI scheme. 171

6.16 Utilization of Interstitial Redundancy. R is the level of redundancy. . 172
6.17 Utilization comparison of different all-GI connected approaches. 174
6.18 Utilization vs. % of PEs faulty for a number of variations on Hedlund's hierarchical scheme. The labels refer to the number of PEs that are considered spare in each block of 12 PEs. 175
6.19 Utilization achieved by combining the hierarchical approach with the GI-column/in-line row reconfiguration scheme for the blocks. 176
6.20 Utilization of hierarchical scheme using varied width columns. 177
6.21 Effect of array size on utilization of in-line rows/LI columns scheme. . 179
6.22 Utilization of the CE & LI connections only scheme under assumptions of clustering and no clustering. 181
6.23 Utilization of the in-line rows/LI columns scheme under assumptions of clustering and no clustering. 182
6.24 Utilization of the in-line rows/GI columns scheme under assumptions of clustering and no clustering. 183
6.25 Utilization of Hedlund's scheme under assumptions of clustering and no clustering. 184
6.26 Utilization of the dynamically varied block size hierarchical scheme under assumptions of clustering and no clustering. 185
6.27 Utilization of all-GI schemes under assumptions of clustering and no clustering. 186
6.28 Relative frequencies of maximum switch lengths, in terms of number of switches in that path for the in-line rows/GI columns scheme. The raw array consisted of 100 PEs. 188
6.29 Number of switches in in-line rows/GI columns scheme for a number of array sizes. 190
6.30 Number of switches in longest path for all-GI schemes. 191
6.31 Number of switches in longest path for all-GI schemes – clustered faults. 192
6.32 A length of reconfigurable interconnect and its equivalent RC model. . 194
6.33 Optimum communications speed for different switch numbers and inter-switch wiring distances. 195
6.34 Communications speed for different switch numbers and inter-switch wiring distances with the switch size set to $S = 10$. 197
6.35 Approximate speed comparison of different redundancy approaches. . 198
6.36 Improving in-line row schemes' effectiveness by making the row reconfiguration wiring itself defect tolerant. 204

6.37 GI columns/in-line rows scheme modification to improve speed. 212
6.38 Vertical connections for an increased speed all-GI scheme. The horizontal connections are identical in nature. 213

List of Tables

1.1 Comparison of Packages Figures of Merit 7

2.1 TRW's superchip set 18

3.1 Yield formula for different defect density distributions 41

5.1 ADVICE Results: Driver and transmission gate size vs. frequency (MHz) 133

5.2 Area × Delay products for the first three cases 136

6.1 Comparison of Utilization and optimum communications speed at different failure rates. 199

6.2 Area effects of reconfiguration wiring and switching overhead. 201

6.3 $E(P)$ of 100 mm square arrays with different reconfiguration schemes. $D_0 = 0.03$ defects/mm^2. *Switch size is 5.* 203

6.4 $E(P)$ of 100 mm square arrays with different reconfiguration schemes. $D_0 = 0.03$ defects/mm^2. *Switch size is 1.* 204

6.5 Comparison of different schemes with switch size=5. 206

6.6 Comparison of different schemes with switch size= 1. 207

6.7 Comparison of different schemes with switch size=1. 208

Acknowledgements

As no man is an island, no PhD thesis is completed in isolation, and I wish to thank all of those who contributed to the environment in which this thesis was built. As many people contributed in one way or another to my finishing of this task, I am sure that you all are not covered. Apologies to those I have missed.

First of all thanks to my supervisor, Kamran Eshraghian, whose boundless enthusiasm and energy, even in the face of adversity, has always impressed me. His paving of the road to the end of this task is much appreciated. Thanks also to Pat Clarke, of the Australian Defence Science and Technology Organisation, a true scientist, with a deep curiousity for all things mathematical. I am very grateful to Pat for his suggestions. Now to the last, but not least, of the "Big Three", Stu Tewksbury of AT&T Bell Laboratories. Thanks for providing me the opportunity to work at AT&T Bell Laboratories, and thanks also for your ever ready optimism and help, both inside and outside my thesis.

Now to the team efforts. First of all back to AT&T Bell Laboratories in Holmdel, New Jersey. There a multitude of people gave me help, both technically, and also with the day to day problems of living in a foreign land. First, to Bryan Ackland, and his department. Their help was essential for the production of the "frame" wafers and I also thank them for their general stimulus. In particular I wish to thank Bob Clarke, Jo Othmer, Mehdi Hatamian, Glenn Cash, Jay O'Neill and Adriaan Ligtenberg. Others, at Bell Labs, that deserve to be thanked for their many useful discussions include Larry Hornak, Trevor Little, Tom Wik, Dave Beecham and Vijay Kumar. On the personal side I found many Americans helpful in the often difficult experience of living in the United States. As well as Bryan (who is actually Australian) and Stu I wish to particularly thank Miles Murdocca, Teri Lindstrom and Ruth Dossica (not of AT&T) for their help in this area.

Next to my colleagues here at the University of Adelaide. I wish to thank Micheal Liebelt for his invaluable help with the computing facilities here. I would like to also express my gratitude to Warren Marwood, and the TFB team (in particular John Rockliff, Alex Dickinson, Micheal Pope, Greg Zyner and Syd Fensom) for their stimulus and assistance. Additional thanks to the members of the FES for helping me to survive the PhD process.

Funding for this research was provided by AT&T Bell Laboratories and the Australian Defence Science and Technology Organisation and to them I am grateful.

Finally I would like to express gratitude to my family and to Sandra Whittle for

their critical personal support at various stages of this work.

Preface

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma in any University and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of this thesis. I also consent to this thesis being made available for photocopying and loan.

P.D. Franzon
23 December, 1988.

Chapter 1

Introduction

The introduction of fault tolerance into VLSI gives a number of potential advantages:

1. The economic advantage of yield enhancement for large VLSI (or ULSI[1]) chips.

2. The system advantage of being able to fabricate chips larger than currently economically possible. The ultimate goal here is making chips the size of wafers, or *Wafer Scale Integration (WSI).*

3. The lifetime advantage of being able to make systems that do not require physical repair. (Physical repair being defined here as repair requiring physical replacement of components.)

The first two of these are achieved by introducing a capacity for tolerating the failures that arise from manufacturing defects. This category of fault tolerance techniques are better referred to as *Defect Tolerance.* The requirements for defect tolerance in WSI are somewhat more stringent than those for smaller chips as the wafer yield would be zero without these techniques. Defect tolerance can be defined as [Saucier, 1986]:

> A defect tolerant circuit is a circuit which meets its user's requirements (functional and parametric specifications) despite the presence of end of manufacturing defects.

The achievement of an increase in system life expectancy (the third possible advantage above) implies either a capability to tolerate field failures as they occur, or to electronically, preferably automatically, effect a repair. The first capability results in a ***highly reliable*** system that has a long Mean Time Between Failure (MTBF) or

[1]Ultra Large Scale Integration [Meindl, 1984]

a high probability of surviving a mission. The second capability results in a ***highly available*** system which is down for repair the minimum amount of time possible. All of these fall under the scope of the definition of fault tolerant systems, depending on the system's requirements [Avizienis & Laprie, 1986]:

> Fault Tolerance: How to provide, by redundancy, service complying with the specification in spite of faults having occurred or occurring.

Achieving ultrahigh reliability is generally a more difficult task than achieving high availability or defect tolerance. High reliability implies that errors are detected and corrected as they occur whilst the other two capabilities imply a test and repair function only. Furthermore, achieving high availability requires a capacity for field test and repair or reconfiguration. Field reconfiguration is not a requirement for defect tolerance.

The results of research reported in this thesis is mainly concerned with techniques applicable to Defect Tolerance. Particular reference is given to Wafer-Scale Integration, though the details of defect tolerance techniques for VLSI and ULSI chips are also discussed extensively.

Fault tolerance for ultra-high reliability will not be discussed in this thesis. Ultrahigh reliability can be achieved using techniques that are extensions of the approaches to defect tolerance about to be discussed. Though these techniques are extensions of the defect tolerance approaches to be discussed (and these approaches can often be used to achieve ultrahigh reliability) they would only be used in special situations. The main reasons are that (1) other sources of low reliability are usually more significant, and (2) few situations justify the considerable effort and expense required.

Although defect tolerance is critical to its achievement, it is important to realize that WSI is as much a statement in packaging as in increasing chip integration. Mainstream VLSI packaging has essentially remained static to the point where the communications bandwidth on-chip has become more than an order of magnitude better than that can be achieved between chips [Katevenis, 1983]. WSI can be seen as one packaging alternative that overcomes this problem relatively economically.

1.1 Research Outline

In order to design for defect tolerance one needs both suitable models, to describe the fault situation, and suitable techniques, to 'repair' the faults. The main object of the research work reported on in this thesis was to produce such models and apply

them to two classes of defect tolerance problems to determine the relative tradeoffs in the applicable techniques. Particular consideration was given to the determination and comparison of approaches to mesh arrays. The defect tolerant design of a VLSI chip has also been the subject of some attention.

In this thesis the following results from this research program are presented:

- An accurate yield model is presented for modeling defect tolerant arrays (Section 3.6.) This model offers three major contributions:

  1. It uses an accurate function to model the distribution of defects amongst processors.
  2. The model comes in two forms:– yield; and expected number of processors. These forms are useful in different applications, as discussed in Section 3.6.
  3. A method is presented for accounting for wiring faults in reconfigurable arrays (Section 3.10.) Previously most workers either ignored the possibility of wiring faults or exaggerated their effect by assuming they all were fatal.

  The model was also extended to take account of faults clustering in processors.

- The yield model was applied to the design of a defect tolerant chip to determine its optimal redundancy (Section 3.7.)

- The yield model was also applied to some general cases of defect tolerant chips and wafers (Sections 3.8 and 3.9.) These studies demonstrated when defect tolerance can be usefully applied and indicated some of the trade-offs involved.

- A yield simulator was written and used to determine critical areas of chips. Its design and usefulness is discussed in Sections 3.6 and 3.11.

- Taxonomies are suggested for the different techniques that can be applied to the problem of reconfiguring linear and mesh arrays (Chapter 4.) These taxonomies were found useful when comparing the different reconfiguration methods.

- Particular attention is given to maximizing the speed at which WSI mesh arrays operate. A list of suitable techniques is given in Section 4.8. The application of some of these techniques is discussed in Chapters 5 and 6.

- Appropriate metrics for comparing redundancy schemes were developed (Section 4.4 and Chapter 6.)

- A new scheme for reconfiguring mesh arrays was designed and fabricated. The details of the scheme, its design, and lessons learnt during its production are presented in Chapter 5. Particular attention was paid to optimizing the scheme's speed. Models were developed to determine the speed of a reconfigurable interconnect.

- The different redundancy approaches for mesh connected arrays were compared on the basis of utilization, speed and yield for a number of case studies. This work is described in Chapter 6. The case studies in particular indicated the relative desirable properties of the various approaches. As a result of these studies a number of new approaches to reconfiguring mesh arrays were formulated. A number of new hierarchical approaches are suggested and compared in Section 6.3. Two new approaches, aimed at maximizing speed, are suggested in Section 6.5.

Before proceeding to report the results of this research, as background material, the relative merits of WSI will be discussed in this chapter, and the current world status of WSI projects will be treated in the next. The subject of yield modeling will be reviewed at the beginning of Chapter 3. A summary of existing applicable techniques for the wafer scale integration of mesh, and other, arrays will be presented in part of Chapter 4 and at the start of Chapter 6.

1.2 Advanced Packaging Technologies

The are many factors driving the search for advanced packaging and interconnect technologies:

- The "substrate gap" [Knausenberger & Schaper, 1984] in interconnect capability between Silicon and Printed Circuit Boards (PCBs).

- The cost savings that can be incurred by reducing the power required by on-chip drivers to drive off-chip interconnect. This has been exacerbated by the movement of mainstream IC technology to CMOS, with its lower driving capability.

- The growing mismatch between in-chip wiring density/performance and inter-chip wiring density/performance. In achieving VLSI most of the system is actually packaged in the IC medium whereas previously, with LSI and MSI,

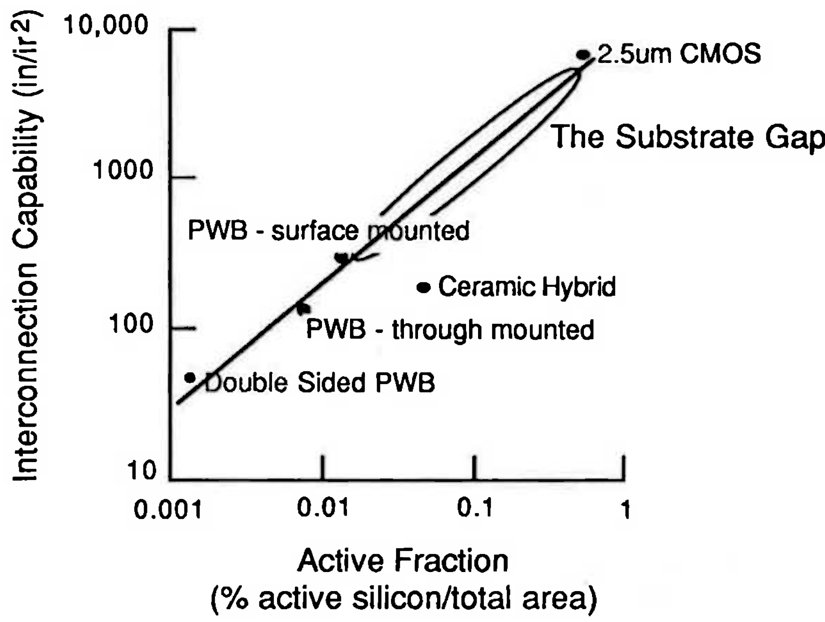


Figure 1.1: Substrate efficiency

most of the system was packaged in the PCB medium. With the majority of the system packaged in the high performance medium the mismatch becomes more noticeable.

W.H. Knausenberger and L.W. Schaper of AT&T coined the term "substrate gap" when they considered the *substrate efficiency* of various interconnect substrates [Knausenberger & Schaper, 1984]. "Substrate efficiency" refers to how much active silicon can be placed on a given substrate. They defined a substrate efficiency figure of merit F:

$$F = \frac{\text{active silicon area}}{\text{total substrate area}} \tag{1.1}$$

They then plotted the interconnection capability, defined as the length of wire that can be placed in an area of substrate against this percentage. This plot is duplicated in Figure 1.1 and there the wiring density vs. active circuit density gap is self apparent.

Taking this and other data, Messner has produced results that confirm this concept of a "substrate gap" [Messner, 1987]. As well as following interconnection density vs. active density arguments he determined the cost density relationships for various interconnection technologies. In constructing this model he described a cost per unit length of interconnect "price bulge", as illustrated in Figure 1.2.

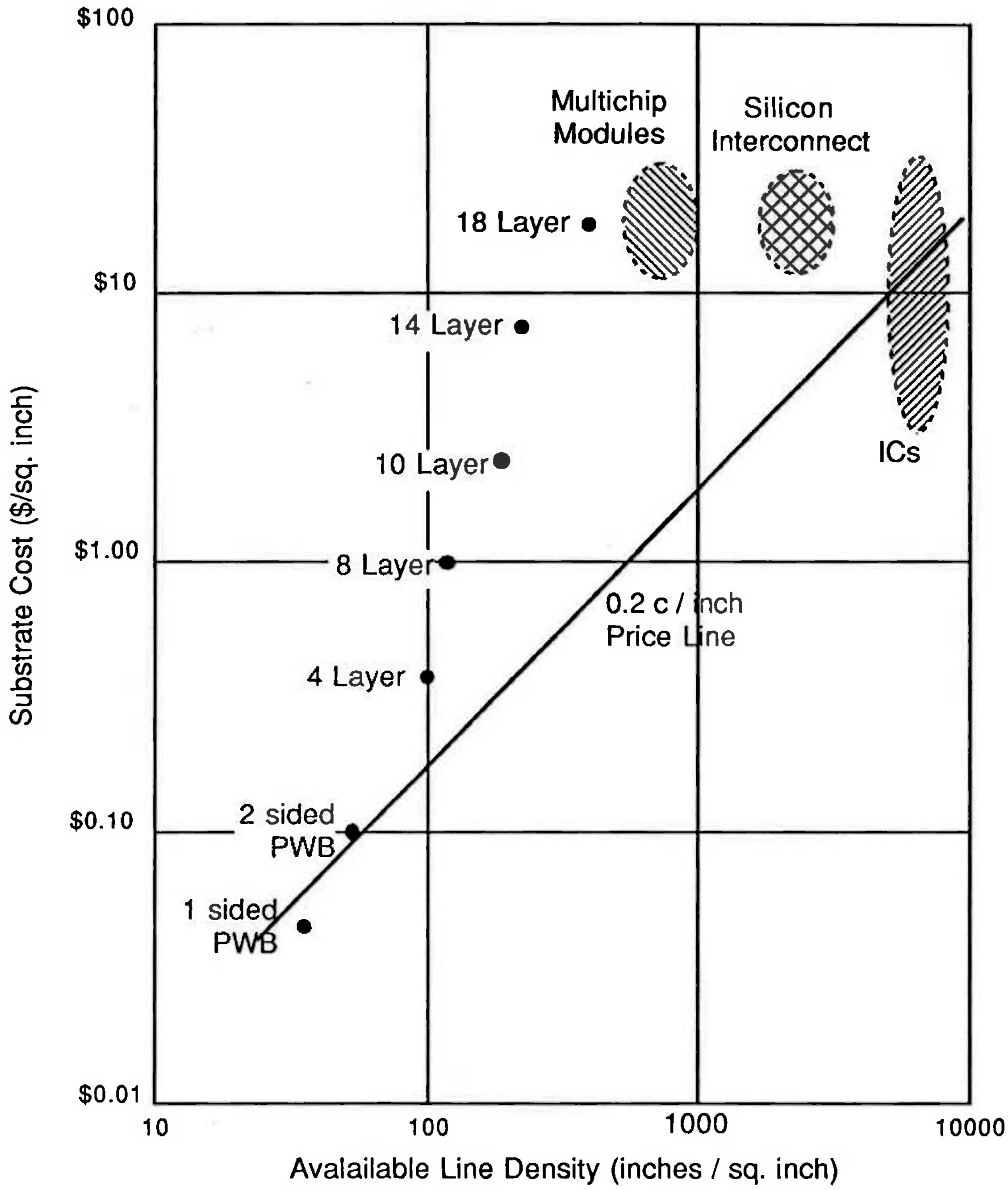


Figure 1.2: Cost-density relationships for different interconnect media.

Packaging Approach	*Power* × Delay	*Size/ Weight*	*Cost*	*Overall Figure of Merit*
Printed Wiring Board	1.00	1.00	1.00	1.00
Thick Film Multi-Layer on Ceramic	1.08	0.42	1.02	0.46
Ceramic Multi-Layer Hybrid	0.34	0.20	0.65	0.044
Thin Film Single-Sided Multi Layer Hybrid	0.19	0.14	0.60	0.016
Wafer Scale Int.	0.10	0.09	0.46	0.0041
Thin Film Double-Sided Multi Layer Hybrid	0.08	0.07	0.44	0.0025

Table 1.1: Comparison of Packages Figures of Merit

The "price bulge" occurs above the constant price per interconnect line of 0.2 cents/inch. Messner suggested that it was only because ICs and double sided PCBs were on this line that they were the primary media used in mass market consumer products. From this graph it can be seen that developing new interconnect media using silicon or ceramic hybrids are likely to return more cost benefit than the continual development of ever more expensive multilayer PCBs.

Neugebauer has compared a number of different packaging choices for a system with an operating speed of 100MHz [Neugebauer, 1984]. Doing this he determined some rough figures of merit for various important factors, which are repeated in Table 1.1. The main factor in determining the figures for Power x Delay and Cost were the extra driver power requirements for maintaining 100MHz. The cost figure was derived from the assumption that the main packaging cost of a system was determined by the cooling and power supply requirements.

Scaling worsens these problems. It is well known that on-chip interconnect speeds do not scale proportionally to on-chip feature sizes [Meindl, 1984]. Local interconnection speeds remain constant with scaling whilst long distance interconnections get worse as die sizes increase and line widths decrease. This same effect will impact inter

chip communications as well. Thus the power costs of driving inter chip connections will get proportionally worse with scaling unless lower capacitance connections are introduced.

Viewed as a transmission line, conventional PCB interconnects perform quite well with a bandwidth of around 250MHz [Ambekar et al., 1987,Rainal, 1979] and propagation delays of better than 190 ps/inch [Deierling, 1987]. In fact, as a transmission line, it performs better than silicon or ceramic based interconnect due to its lower substrate dielectric constant [Chong & McEnroe, 1985]. The substrate gap arises on density, cost and power considerations, rather than pure speed.

Until recently high speed system packages have generally been produced using expensive combinations of low gate count LSI chips and ultracompact packaging [Blodgett & Barbour, 1982] [Watari, 1987]. However as the functionality and speed of CMOS increases there will be a growing need for high speed packaging for these VLSI technologies. As can be seen in Table 1.1 the thin film interconnect technologies of WSI and thin film hybrids have more merit than the more traditional thick film technologies for high speed VLSI and ULSI.

1.3 Potential Benefits of WSI

The potential benefits of WSI, particularly when compared with conventional systems approaches, are listed below in decreasing order of importance. Where appropriate, comparison will also be made with other thin film approaches.

1.3.1 Size and Weight

The integration of several million transistors onto a single wafer leads to an obvious size and weight advantage. For example in the ESPRIT 824 project [Ivey, 1987] a SIMD WSI array has been developed that achieves a scale of integration in which a single wafer replaces ten boards of conventional chips. The reduced cooling and power supply requirements of a WSI system [Neugebauer, 1984] contributes towards smaller size and lower weight. The aerospace industry can, in particular, benefit from products with these advantages. Other thin film packaging approaches can be similarly placed for size and weight.

1.3.2 Reliability

The reliability advantages of WSI are clear and substantial also. A study of Naval Avionics systems failures [Robinson & Sauve, 1977] indicated that:

1. IC components were involved in less than *two percent* of all avionics failures.

2. Approximately *sixty percent* of all avionics failures were attributable to connector and cabling failure.

3. Twenty-Five percent of the failures were caused by maintenance and test procedures.

4. Thirteen percent were due to overstress and abuse of system components.

Although avionics is a physically harsh environment the conclusion can still be drawn that the most significant contribution new packaging techniques can make to improved systems reliability is to reduce the number of physical connectors required [Chaturvedi, 1988]. WSI achieves this to a significant degree indeed.

This conclusion can also be drawn from other sources using data gathered from telecommunications applications. There it has been found that the IC failure rate is roughly independent of the number of gates per chip [Hoover, Jr. et al., 1987]. A typical rate may be 50 FIT (failures in 10^9 hours) On the other hand the failure rate of external interconnections is directly proportional to their number at approximately 0.05 FIT per interconnection. In [Hoover, Jr. et al., 1987] it was concluded that a 10 times increase in the number of gates per IC results in a roughly similar increase in reliability for a constant sized system. (The FIT rates quoted are approximate only as they are influenced heavily by environment. For example, IC failure rates increase by a factor of two for every 10°C [Mahalingam, 1985]. The failure rate for package connections depends largely on the number of thermal cycles the connections go through. For example, a 40 pin leadless chip carrier can withstand between 2,000 and 5,000 thermal cycles between failures for 1 watt dissipation [Deierling, 1987].)

It would be expected that WSI would maintain this advantage over hybrid technologies. No figures are as yet available on the reliability of solder bump and other advanced hybrid connection technologies. IBM's thermal conduction modules, which employ solder bump techniques on LSI chips, are reputedly very reliable however.

WSI may also result in a higher availability than other packaging approaches. Many of the techniques used to implement defect tolerance are capable of being repeated later to effect a field repair if so desired. This may not seem important

given the two percent figure in point 1 above but does have an impact when one considers burn-in failures. Typically more ICs fail during burn-in than during their first five years of operation [Little, 1986]. This means that any re-repair capacity results in some economic advantage.

Not only may the reconfigurability of WSI lead to higher burn-in survivability and higher reliability but, depending on the actual implementation, it may also result in reduced maintenance costs. As the repair process is highly automated, repair turn-around would be shorter and higher availability would be the result.

1.3.3 Reduced Power Requirements

As mentioned in section 1.2 substantially reduced power would be required by WSI drivers. However the power density of such systems could be higher thus creating a different heat dissipation problem possibly requiring advanced cooling techniques. Other thin film packaging techniques would be similarly placed.

1.3.4 Performance

There is significant potential for performance enhancement due to the lower capacitance and higher density of connections available. This potential may be reduced however when the overheads required to achieve the necessary defect tolerance are included. It would be expected that thin film hybrids would enjoy a similar advantage.

1.3.5 Production Costs

Reductions in production costs stem from reduced packaging, power supply and cooling needs. Any possible reduction may vary on a project by project basis however. For example, extra costs may be introduced in handling higher cooling density requirements. Costs may also be incurred in unusual areas such as in carrying out the reconfiguration steps required.

An additional cost factor is the development cost. It may be usual that a WSI product would have a smaller market volume than many VLSI products and thus the already (probably higher) development cost must be amortized over a smaller production run.

The processing costs of thin film hybrids would be expected to be higher than for other packaging approaches.

A rough cost comparison of populating a board with VLSI components compared with producing a WSI product with a similar capability is given in Figure 1.3. The costs of manufacturing and packaging the VLSI components are taken from [Dicken, 1988]. The manufacturing and Quality Assurance costs are based on an Australian manufacturer's experience[2]. The wafer scale costs for testing and reconfiguration are estimates. In the example taken from [Dicken, 1988] each wafer yielded 56 working dies on average (calculated as the wafer cost divided by the die cost.) It is assumed for the wafer scale case that each wafer yields a number of processors with a similar functionality. The basis for this assumption is the reasonable expectation that the increase in functionality resulting from the extra area gained from the loss of the pads and inter-die kerfs is precisely matched by the area lost to the reconfiguration wiring and those processors that can't be included in the array due to limitations in the reconfiguration wiring's flexibility. It is further assumed, pessimistically, that the packaging and assembly costs are together twenty times worse than that of a single die and the testing and screening cost is the same as the wafer cut into die. These costs depend a lot on the details of the function being implemented. It is also assumed that the reconfiguration step costs the same as the testing step. It would be expected that a similar level of equipment and time would be involved in both of these steps. The bottom line is that the WSI based product ships at about one-third of the cost of the equivalent conventionally packaged product. This is not too dissimilar to the cost comparison calculated for Table 1.1. Neugebauer's method for determining cost was largely based on the assumption that supplying power and dissipating the resultant heat was the major cost component. In Figure 1.3 actual production costs are considered and simple cooling requirements are assumed.

1.4 Difficulties to be Overcome to achieve WSI

A typical wafer would be expected to contain 100 or so defects manifesting themselves as faults. The main problem to be solved in achieving WSI is to introduce sufficient defect tolerance to handle this number of faults.

Architecturally, this is a sizable constraint. Usually, the accepted solution is to fabricate an array of similar elements with a large number of spares. The object is to *reconfigure* these spares into and out of the system to get a working wafer. Some form of *Reconfiguration Scheme* is required to do this and this introduces a silicon

[2]Transponder Australia

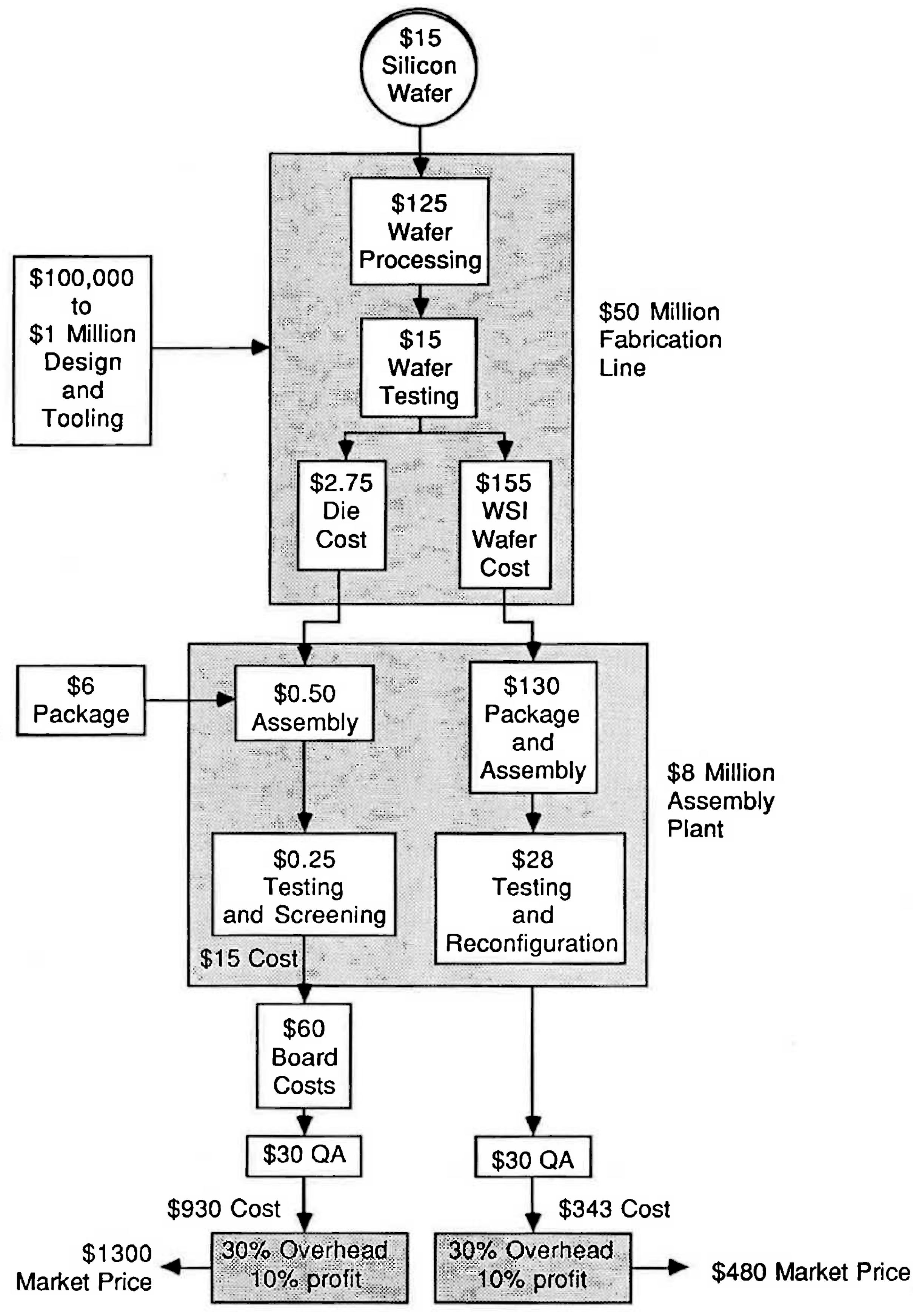


Figure 1.3: Broad brush comparison of ($US) costs for a WSI based product compared with the equivalent discrete die and board based product.

area and performance overhead. Judging the effects of these overheads and trading them off against the efficiencies of reconfiguration schemes will be examined for mesh arrays in Chapter 6.

Other problems that may create difficulty for a WSI product include:

- Possibly higher heat dissipation density requiring special packaging techniques;
- Possibly very high pin out count, again requiring special packaging;
- Increased design and development time;
- Smaller production runs;
- Difficult on-wafer power distribution and electromigration;
- Increased signal noise and crosstalk.

1.5 Hybrid WSI

As indicated in Table 1.1 thin film hybrids offer many of the advantages of WSI. Furthermore thin film hybrids do not suffer from the architectural limitations outlined in Section 1.4. One class of hybrid that has become of increasing interest in recent times are hybrids with silicon as their substrate. Given their closeness to WSI the techniques used to produce these hybrids are often referred to as Hybrid Wafer Scale Integration (HWSI), though the "H" may sometimes be dropped.

Silicon hybrids offer a number of advantages over ceramic hybrids or advanced PC boards [Lyman, 1987]. Using well established IC processing techniques, the silicon substrate can easily accommodate 10 to 25μm conductors compared with the 75μm conductors of most advanced PC boards or ceramic hybrids. This could reduce the interconnect area to less than 1/10 that of standard boards or hybrids.

With shorter wiring runs parasitic inductance and capacitance would be reduced. Also passive elements such as bypass capacitors and terminating resistors can be fabricated on the substrate.

Silicon hybrids also have the important advantage that the thermal coefficients of expansion of the substrate and mounted chips are the same. This results in significantly reduced mechanical thermal stresses. This is particularly advantageous for operating at liquid nitrogen temperatures. Here the performance of CMOS parts exceeds that of bipolar parts at room temperature [Johnson, 1984].

However higher IC densities can lead to higher thermal dissipation densities. Fortunately silicon is a good conductor of heat and it is easy to glue metal or silicon carbide heat sinks to the substrate, thus alleviating this potential problem.

1.6 Three Dimensional Packaging

The concept of placing subsystems on silicon wafers leads naturally to the idea of forming complete systems in this medium. Wafers could be connected through the edges only. However if the wafers could be stacked and inter–wafer communication made possible through the stack significant advantages may be gained. The concept of, and current projects in, three dimensional packaging and three dimensional communications will be discussed in Section 2.4.

Chapter 2

World Status of WSI

The concept of wafer scale integration is an attractive one and has been around for almost as long as ICs have. However it wasn't until the late 1970s that the practical limitations of WSI were appreciated and appropriate methodologies developed. This chapter presents a brief history of WSI followed by a summary of major WSI projects. Only projects resulting in, or intending to result in, actual wafers or large area chips will be described.

Current Hybrid WSI projects and current investigations into three dimensional packaging will also be presented.

2.1 History of WSI

The history of WSI can be divided into three periods:

1964–1970 In these early days of WSI (even before the name had been coined!) the intention was to carry out the transition from SSI to MSI and possibly even LSI in a single step. The technological means for doing this were largely developed at Texas Instruments [Lathrop et al., 1967]. They developed the technique of discretionary wiring (where a unique final wiring pattern was produced for each wafer) as the means for reconfiguring the wafer after probe testing. The cost however of making a unique mask for each wafer precluded commercialization of the concept. Little work was carried out in this area until 1977.

1977–1984 The renaissance of WSI started in 1977/78 with three important papers that re-awoke interest in the area [Finilla & Love, 1977] [Manning, 1977] [Aubusson & Catt, 1978]. In his paper, Aubusson coined the phrase "Wafer

Scale Integration". In the early 1980s there were several attempts at developing wafer scale products, the most notable of which was Trilogy's attempt to put an IBM 3081 processor on an ECL wafer [Peltzer, 1983]. Although unsuccessful, they provided much insight into the understanding of some of the problems associated with this technology. There were also a lot of academic research on the subject of WSI during this period.

1984– This period, which is continuing today, marks the maturation of WSI technology. In particular several projects have been initiated which cope realistically with the limitations of WSI. Several researchers have also pursued investigations into the possibility of Hybrid WSI (silicon on silicon hybrids) as a means of achieving some of WSI's advantages whilst avoiding the constraints required for full WSI.

2.2 Major WSI Projects

Listed below are the major WSI projects, past and present, that have reached my attention. Projects that have resulted in little published information will of course be missing from this list. For example, IBM has been reported several times as having some form of WSI effort [Cole, 1985][ICC, 1987] but have never announced any details. Only projects that resulted in the fabrication of some representative device will be described.

Discretionary Wiring Techniques

The first attempt at WSI was developed largely at Texas Instruments and eventually produced wafers [Lathrop et al., 1967]. The discretionary wiring technique involves making a unique mask for each wafer that can be used to connect together those circuits that work. Unfortunately, though many wafers worked, the cost of making a unique mask for almost every wafer hindered the concept from becoming a commercial reality. A methodology was developed to reduce the number of unique masks required [Calhoun, 1969] [Calhoun & MacNamee, 1972] but this approach was still insufficient to achieve commercial practicability.

Memory and Large Chip Efforts: NTT

Nippon Telephone and Telegraph has had a long interest in defect tolerance techniques. Three memory WSI wafers were fabricated in the late 1970s and the early 1980s [Kitano et al., 1980][Egawa et al., 1980] [Ueko et al., 1984]. More recently they have designed and fabricated a 37mm by 21mm sorting chip [Tsuda & Satoh, 1987].

WSI Processor: Trilogy

In the early 1980s Trilogy attempted to put an IBM 3081 CPU onto an ECL wafer [Peltzer, 1983] [Amdahl, 1984]. Unfortunately, the effort met with failure primarily as a result of a combination of technical difficulties. Trilogy was trying out a new design methodology while bringing up a new process line at the same time. In attempting to do all of this at once they found the combination of achieving an appropriate balance between high defect counts, maintaining sufficient redundancy for an economic wafer yield, and achieving high speeds beyond the technical resources of the company [Jones, 1987] [Fischetti, 1984].

Nevertheless Trilogy still believes that they would have succeeded in the end though the extra effort and delay would most likely have made their product unprofitable [Fischetti, 1984].

VHSIC Phase 2 and TRW

The U.S. Department of Defence's Very High Speed Integrated Circuit (VHSIC) program has entered its second phase where the aim is to obtain a 20 fold performance improvement over the Phase–1 microcircuits [Klass, 1986]. It is intended that this performance improvement be obtained by decreasing the minimum feature size to 0.5μm, increasing the speed to 100MHz and increasing the circuit density to to over 100,000 gates for 9mm square chips.

Two of the VHSIC Phase–2 contractors, IBM and Honeywell, aim to achieve these objectives by fabricating 9mm square CMOS or TTL chips and packaging the chips on small (100mm or less) ceramic hybrid boards. (Honeywell also has a separate silicon hybrid program.) TRW, on the other hand, is taking a more aggressive approach and fabricating 36mm square "superchips" in TTL and CMOS to achieve their goals [Ele, 1986]. A list of the superchips is given in Table 2.1. Each superchip is either one quarter or one third of a complete wafer.

Chip Type	*Process*	*total devices (M)*	*usable devices (M)*	*chip size (in.)*	*yield %*	*power (W)*	*status*
FFT	bipolar	3.9	2.2	1.4 by 1.4	38	7.8–16.7	optional
2 Mb four port SRAM	CMOS	27.9	18.8	1.4 by 1.4	50	4	funded
Signal Processor	CMOS	27.9	9.8	1.4 by 1.4	29	7.9	funded
Data Processor	CMOS	34.7	21.7	1.4 by 1.8	20	8.5	optional
Associative Processor	CMOS	10.1	7.1	1.4 by 1.4	45	9.5	optional
Convolver Correlator	bipolar	10	6.2	1.4 by 1.8	42	14.5	funded

Table 2.1: TRW's superchip set

Unfortunately processing problems have so far prevented TRW from accomplishing their initial superchip aims. It was originally intended that E-beam techniques would be used to provide trench isolation in their 0.5μm CMOS process. However they were never able to do this repeatably so their aims had to be scaled down [Waller, 1988]. Instead of fabricating a 27.9 million transistor CMOS signal processor a 4 million transistor processor has instead been fabricated on a 1.3 in. by 1 in. die. The two designs are essentially the same except that the latter has no on-board RAM.

TRW has not released details of the defect tolerance techniques employed but it appears that the nonreplicated modules are using Triple Modular Redundancy (TMR) and the replicated modules have spares that can be configured in via 'soft' reprogrammable switches. The use of soft switches for reconfiguration means that the system can also be reconfigured in the field. These techniques bring the anticipated

yields up from an estimated 10% (with an expected 200 to 250 defects per square centimeter) to the figures given in Table 2.1.

The elements of the superchip set are all self testing with the test circuits occupying from 18% to 25% of the total chip area. The remainder of the difference between "usable devices" and "total devices" is probably used for spares.

Even with the heat densities indicated in Table 2.1 no packaging problems are anticipated.

WSI Disk

Formerly known as Sinclair Research [Inf, 1984], Anamartic's plans are to concentrate on what they perceive to be a market for WSI memory components.

It has long been recognized that between primary memory and disk memory there exist large gaps in terms of access time and cost per bit. Bubble memories were developed in an attempt to fill this gap. Anamartic perceive WSI based memories as succeeding here [Wilkinson, 1987].

Their studies have shown that a WSI based disk replacement unit would be cheaper than one based on conventional DRAM chip/PCB mount technology and would be more compact than one based on high density hybrid packaging techniques. Thus they see their WSI memories as providing a very high density memory unit that is more compact than primary memory whilst being faster and more reliable than disk memory. For compatibility they have made the interface to the WSI system look like a disk interface.

So far the company has fabricated a number of prototype wafers [Jesshope & Moore, 1986], which they will be enhancing with additional funding [Ele, 1987].

The company sees another possible market in applications where highly robust secondary memories are required.

RVLSI Program

A technologically orientated WSI program has been on-going at MIT's Lincoln Laboratories for several years with U.S. Department of Defence funding. The program has been orientated at bringing into service techniques for restructuring wafers using laser fuses and antifuses.

In the Restructurable VLSI (RVLSI) program reconfiguration is performed by laser programming of an interconnection network laid out between processing el-

ements. Initially the processing elements are not connected. After testing, fuses are blown and antifuse joins created in order to obtain a desired final connection pattern [Raffel, 1986] [Anderson, 1986]. The original fuses and antifuses required extra processing steps above those in normal fabrication lines [Wyatt et al., 1984] [Chapman, 1986] and have proven to be highly reliable. However the program now has designed a lower performance laser antifuse that does not require additional processing steps [Chapman et al., 1987].

The program has produced a number of wafers for signal processing applications which were mainly intended as technology demonstrators [Rhodes, 1986] [Gaverick & Pierce, 1983] [Raffel et al., 1985] [Mann & Rhodes, 1986]. More recently the group has branched into the fabrication of neural networks.

2.2.1 ESPRIT 824 Program

This program is an European Economic Community funded venture into Wafer Scale Integration. Funded to the level of approximately $US4.4M, with the industrial affiliates providing half of their own funding, this project spans four countries, six institutions and involves about 30 workers [Trilhe & Saucier, 1987].

In all of the proposed wafers, connections are to be made between working elements using CMOS transmission gates. Various technologies for programming these gates which are under investigation include:

1. *Floating Gate FETs.* E-beam programmed floating gate FETs are considered an attractive technology because of their high density and reprogrammability [Girard et al., 1987].

2. *Laser fuses.* The cutting of aluminium and polysilicon lines is a technique that leads to highly reliable open switches.

3. *Serially programmed latches.* The switches are programmed by serially fed in commands.

On many modern fabrication lines wafers are produced by stepping a scaled mask (usually containing a 10× magnified pattern) across the wafer. This creates a potential problem for a wafer scale part as it is desirable that some areas of the wafer have different masks than others, in particular around the periphery where the pads are. In the ESPRIT project this is solved by placing a set of unconnected pad drivers around the edge of the repeated 1:10 mask. A final 1:1 lower density wafer-wide

aluminium masking step is used to place the required peripheral pads and connect the required pad drivers only.

Another serious problem facing a wafer scale designer is that of distributing the required power over thin film aluminium without significant voltage drops and metal migration. One possible solution being investigated is the formation of a low cost, low resistance copper pattern on top of the passivation layer of the wafer [Barrett et al., 1986] [Barrett, 1987].

Three WSI designs are being planned for this project, namely a static RAM, a SIMD array and a defect tolerant microprocessor [Trilhe & Saucier, 1987].

4.5 Mbit static RAM

This RAM will be implemented in a four inch 1.25μm CMOS process like the other two projects. Unlike the Anamartic disk replacement it is logically organized as a conventional memory with 16 address lines.

SIMD Array

Intended mainly for image processing applications the aim of this project is to produce a 128 by 128 array of bit serial 30MHz Processing Elements (PEs) on a single wafer [Ivey et al., 1987]. This wafer has the capacity to replace 10 boards of conventional chips. An interesting facet of this scheme is that it will use a electron beam machine to test as well as to program the floating gates. Electron beam testing is normally considered uneconomic for VLSI design because of the excessive testing time required. However in this case the testing time is substantially reduced by using signature analysis, with all of the PE's signatures being produced in parallel, and then using the E-beam tester to read these otherwise difficult to access signatures. The E-beam tester, which is based on a scanning electron microscope, is then used to test the connections in parallel.

Each PE of the array consists of a serial adder/subtractor, two 64 bit RAMs and steering logic.

Large chip Microprocessor Based System

The motivation for this project is to develop a means for building large systems on a single large chip by (1) providing a set of building blocks, the primary block being a defect tolerant microprocessor [Genestier et al., 1986], and (2) providing a flexible interconnect in the form of a sea of gates. From these a microprocessor system that

can be dynamically reconfigured to meet changing requirements is produced. By providing different mixes of predefined building blocks a broad range of microprocessor based ASICs can be quickly developed.

WSI Associative String Processor: Brunel University

The WSI Associative String Processor (WASP) is intended to be a wafer scale example of a homogeneous, reconfigurable and programmable, fine-grain parallel processing architecture [Lea, 1987][Lea, 1986] based on associative memories.

The basic architecture of the WASP is a set of Associative String Processor substrings that can be flexibly interconnected into full string processors. Each ASP substring can be programmed separately so that the whole machine can operate either in SIMD, SISIMD or MISIMD mode. Each substring consists of a set of Associative Processor Elements (APE) which operate in SIMD mode and can also be flexibly interconnected. Each APE consists of registers, a comparator, and control, communication and processing logic.

MNOS Transistors: McDonald Douglas

McDonald Douglas has had an interest in WSI since 1971 [Hsia et al., 1979] and though nothing has been published since 1979 some continued attention appears to have been focused on military applications. Hsia's technological approach to WSI was to use the nonvolatile MNOS (metal-nitride-oxide) transistor for reconfiguration. The MNOS transistor is similar to the MOS transistor except that the gate oxide layer is replaced by a silicon nitride layer over a thinner silicon dioxide layer. By applying a moderately high voltage to the gate (about 25 volts) the silicon dioxide layer becomes permanently conductive through the trapping of charge carriers.

ALVEY WSI Program

This British project funded with about $US2.2M of government funds and matching industry funds is mainly intended as a means to fund various connected lines of research. The primary participants in this project are [Moore, 1987]:

- GEC,
- Plessey,
- ICL,

- Brunel University,

with Oxford and Southampton Universities being minor participants.

The main emphasis of this program is on Systolic Processing, with the industrial affiliates reportably having relatively little interest in full WSI. GEC has interests in the automated synthesis of Systolic Arrays. Plessey is developing a super Digital Signal Processor. ICL is experimenting with silicon hybrids. Brunel is using the project to continue funding on WASP.

Oxford University has several small research projects in process and is collaborating with GEC. Southampton University has brought up its own I^2L line and has fabricated a wafer scale memory on it [Bentley & Jesshope, 1986].

Other WSI projects

Micron Technology has announced a project to produce a 4Mb CMOS DRAM by clustering four 1Mb parts onto large single chips, and using Hamming codes to handle defective bits [Ele, 1985b].

In 1985 Inova Microelectronics Corp. announced that it intended to build 256-K static RAMs onto single pieces of silicon by using 8 interconnected 64-K SRAMs [Bennett, 1985]. Their redundancy technique allowed half or fully working 64-K "cells" to be interconnected to form the 256-K parts. The parts were packaged in regular 28 pin DIPs and it was predicted that they would have an access time of 55 ns. This will gave them four times the density of the then conventional 64-K RAMs and twice the speed of more recent 256-K SRAMs.

ITT Corp is another company working on what it calls "Large Area Integration" [Ele, 1985a]. However instead of aiming at the memory market a group in Shelton, Connecticut, is building a large chip cellular array processor based on bit-slice technology. Similar in architecture to Goodyear's MPP supercomputer or ICL's plc, the cellular array processor uses bit-parallel rather than bit-serial elements. These elements can be configured into separate, variable word-length blocks. Defect tolerance is handled by the switching of signals around defective elements.

GTE laboratories have investigated implementing multiple pipeline serial processors in WSI [Fried, 1984].

AT&T Bell Laboratories have an interest in Wafer Scale Integration as part of a larger novel packaging project [Franzon et al., 1987] [Franzon & Tewksbury, 1987].

Purdue University has carried out research aimed at exploring the possibility of placing an array of configurable highly parallel processors (CHiPs) [Snyder, 1982] on a

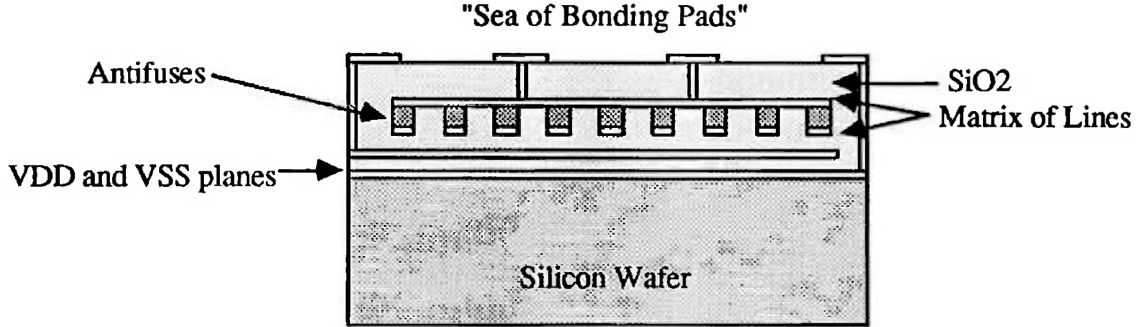


Figure 2.1: Cross section of Mosaic's silicon circuit board

single wafer [Hedlund, 1982] [Hedlund & Snyder, 1984]. Reconfiguration was carried out by programmable switches which could be used both for defect tolerance and for achieving different interconnect patterns. ICs containing the CHiP processor and some switches were fabricated but the study is essentially complete [Hedlund, 1987].

2.3 Major Hybrid WSI Projects

As discussed in Chapter 1, Hybrid WSI, or silicon on silicon hybrids, retain many of the benefits of pure WSI whilst avoiding some of its problems.

Mosaic's Hybrid Wafers

Currently Mosaic Systems Inc. has in manufacture a hybrid silicon product that is essentially an electrically programmed silicon circuit board [Bogdan, 1987]. Versions of these boards are currently commercially available.

A cross section of the hybrid board is illustrated in Figure 2.1. It consists of four metal layers beneath a "sea of bonding pads". The first two metal layers form the VDD and VSS planes. The next two layers form a matrix of orthogonal, parallel lines separated by amorphous silicon antifuses. These antifuses can be short circuited by high (20V) voltages but remain open circuited at normal switching voltages.

Electrically any formed line acts as a lossy transmission line with impressive delay characteristics. With their Series 2000 lines, the circuit does not require any special placement considerations if delays of no less than 2ns are acceptable. With their Series 3000 lines delays down to 350ps are possible with no special placements and

smaller delays can be achieved by considering placement carefully. As the lines are lossy they require no terminating impedances.

Currently the Universally Programmable Silicon Circuit Boards (Unipro SCBs), as they are known, are fabricated in one inch square sizes for the mass market. Three inch square wafer boards are also available. On the one inch boards 2758 0.12mm square signal pads are provided at a 0.3mm pitch. This allows high pinout ICs to be accommodated easily.

One prototype product already announced is a 1.1 Mbit static RAM being developed by Westinghouse Electric Co. for testing in a military project [EPP, 1987]. Consisting of seventeen 64Kbit SRAMs and seven buffer ICs, this module is built on two one inch SCBs and is packaged in a quad width flat pack.

Mosaic's SCBs have the particular advantage that high performance prototypes can be quickly turned around, the board configuration process taking only two hours.

Advanced VLSI Packaging Technology: AT&T Bell Laboratories

Standing for Advanced VLSI Packaging, AT&T's AVP Technology is based around a highly doped silicon wafer on which layers of conductors and dielectrics are customized. The AVP wafers consist of copper power and ground planes, three copper signal layers and a top layer of surface mount pads. Polyamide dielectrics separate the signal layers to reduce capacitance.

Signal line widths and spacings of $10\mu m$ are used giving low noise and low capacitance interconnections [Giffels et al., 1987]. These interconnections are designed to achieve a 50Ω characteristic impedance [Lyman, 1987]. It has been estimated that the AVP wafers will be able to operate at frequencies above 50 MHz synchronously [Bartlett et al., 1987] and up to 100 MHz asynchronously. Speeds of 45MHz have been demonstrated [Colbry et al., 1987].

The IC dies are flip mounted onto the wafer surface. Solder bumps or gold contacts are used to form the connections between the chip and the substrate. Flip mounting reduces the capacitance and inductance in comparison with leaded connections.

Two example subsystems have been built using this technology. The first was a 1.0 inch by 3.0 inch substrate for the three chip set WE32100 32 bit microprocessor. The advanced packaging increased the operating frequency by a factor of three and decreased the required board space by a factor of seven over conventional packaging [Lyman, 1987].

The second is a 256 input by 256 output nonblocking crosspoint switch [Colbry et al., 1987]. The switch consists of twenty four 6mm by 3.3mm 1μm CMOS ICs flip bonded onto an AVP wafer. Each IC contained 166 I/O pads. The switch dissipates 15 Watts and runs at 45 MHz. It was calculated that the conventionally packaged equivalent system would have dissipated 192 Watts due to the larger drivers thus required. Providing full size pads would have resulted in a 30% area growth for the chips involved. Additionally routing the huge amount of wiring required would have been very difficult.

Wafer Transmission Module

Developed at Rensselaer Polytechnic Institute (RPI), the Wafer Transmission Module (WTM) is a hybrid wafer approach based on high speed thick film transmission lines [Donlan et al., 1986]. The structure contains two interconnection and two power layers onto which IC dies are mounted by wire bond, flip mount or TAB bonding techniques. The WTMs experience a high yield due to the use of thick films and conservative design rules. In addition it is possible to repair many wafer defects using electron or focused ion beam techniques [Lin et al., 1986].

The WTMs can be made as three or four inch square wafers. The three inch wafer can mount 16 chips and has 120 I/O pins. The four inch wafer can mount 37 ICs and has 276 pins. Wiring channels are provided between the chip sites. Wiring delays may vary between sub-ns and 3ns depending on the connection length.

In cooperation with Tektronix, an example high performance system has been built using the WTM approach [Greub et al., 1987]. Referred to as a Fast Reduced Instruction Set Computer (FRISC), it utilizes advanced silicon bipolar differential logic ICs with 66ps gate delays. Containing 14,500 gates, the FRISC architecture is realized with 11 dies mounted on a 3.4 cm square hybrid and dissipates 40W. An operating frequency of better than 125MHz and a pipelined instruction rate in excess of 125 MIPS are expected with worst case wafer interconnect delays being 70 ps for 12mm connections.

High Density Hybrid Wafer Connections

Based in Grenoble, France, LETI has developed a hybrid wafer module with a number of interesting differences [Nicolas, 1987]. Technologically the project has three interesting aspects:

1. A dry process for creating high density microbumps for flip chip bonding. LETIS's microbumps can achieve densities of 400 bumps/cm^2 compared with densities of 50 bumps/cm^2 achievable with traditional technologies. Their microbump technology also requires a smaller capital investment than other approaches.

2. Through-wafer vias formed by laser drilling. This allows through the wafer connections for power and signal lines.

3. A high density multicontact edge connector has been developed for connection to these wafers. The contact pitch can be fixed between 0.3 and 1.27 mm with up to 600 contacts in one connector.

LETI's philosophy on in-wafer interconnect is to use chip type fine geometry lines for short connections and thicker, faster lines for long connections over low dielectric insulators.

Active substrate hybrid wafers

The concept of the active substrate hybrid wafer is to fabricate the wafer as a programmable interconnection system on to which ICs are bonded [Franzon et al., 1987] [Wooley et al., 1987]. An active substrate would allow the construction of a reprogrammable system that could respond to different user needs as well as possibly to failures. The active substrate could also be used to provide a high speed implementation of the system "glue".

To achieve high speeds bipolar devices would be preferred for the active substrate but examples have been built using CMOS substrates for the purpose of this thesis.

Other HWSI Projects

Hitachi have used a similar approach to the AVP approach to form a 7 ns, 128 Kbit ECL RAM in a 14.2mm by 25mm module [Lyman, 1987]. Eight RAMs and one logic chip are flip mounted onto a double layer metal silicon substrate which is then attached to a silicon carbide ceramic heatsink. The substrate is then mounted in a 96 lead flat pack.

A group at the Alabama Microelectronics Science and Technology Center of Auburn University are developing a silicon hybrid technique that involves placing the ICs in etched holes in the wafer [Lyman, 1986]. The ICs are dropped in whilst

the wafer, with IC, is placed against an optically flat surface. The IC is then glued in place by a polyamide binder. The end result yields steps on the wafer–binder–IC surface of less than 2μm. A second layer metal process is then used to connect the ICs with the first layer metal interconnect already on the wafer.

Honeywell have developed silicon substrate techniques based on two layer aluminium interconnect fabrication technology and a flip chip solder bump surface connection technology [Huang et al., 1983].

2.4 Three Dimensional Packaging

Theoretically, pure three dimensional VLSI systems have significant advantages over two dimensional systems [Rosenberg, 1983b].

In a practical sense too the prospect of three dimensional packaging and three dimensional communications through the package would be expected to lead to a number of advantages. Many WSI and HWSI boards will contain more functionality than is usually present on a conventional PCB. Restricting communications to the edges only may overly constrain system connectivity. Furthermore many algorithms and architectures would map more efficiently onto systems with three connection dimensions rather than two.

Communicating in the third dimension is not without problems however. The surface area and power requirements of the connections will be at least an order of magnitude larger than the requirements of the connections placed in the first two dimensions.

Microbridge Connections

Hughes Research Laboratories have developed a 3D wafer structure based on a microbridge connector as illustrated in Figure 2.2 [Grinberg et al., 1984].

Physically the microbridge consists of an aluminium bridge, raised 2.5 μm above the surface of the wafer, connected to an aluminium feedthrough that passes through the wafer. The area required is about 0.5 mm square but the surface under and around the bridge can be used for circuitry. In the proposed architecture each cell will fit within the area provided by the square defined by the diagonal formed by the bridge.

The compliant spring type structure created by using a raised bridge allows contact to be maintained over the expected range of wafer warping. This separation

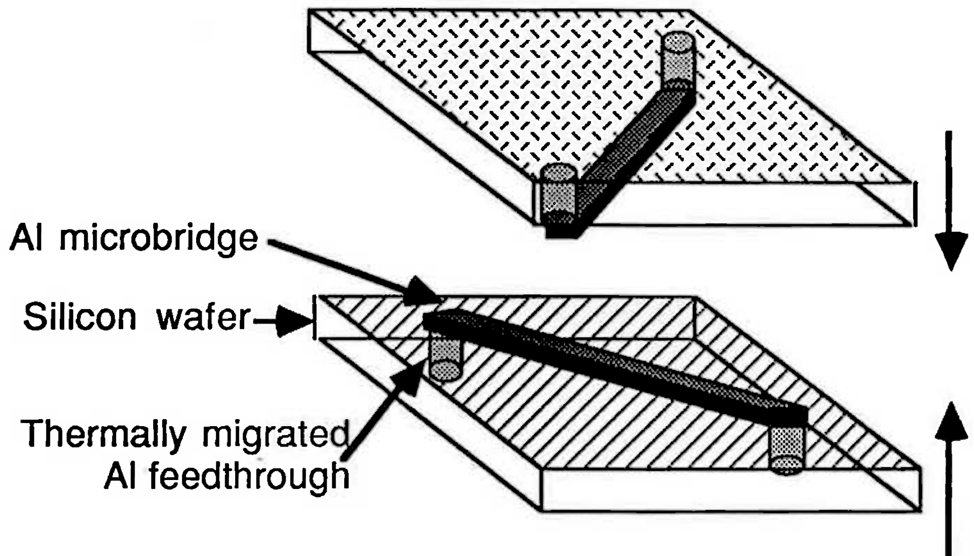


Figure 2.2: Microbridge connector structure developed by Hughes.

from the array surface also results in a low capacitance. No studies have been carried out on the reliability of the connection.

Hughes is building a 3-D signal and image processor based on this technology using the architecture given in Figure 2.3. The processor is intended for handling two dimensional problems. It consists of a 128 by 128 two dimensional array of bit serial processing elements, each element being formed by vertical connection of sub-components through the wafer stack. Each wafer in the stack contains one type of component only, which when combined with the components on the other wafers, forms the PE. A data bus runs vertically through the stack connecting the PE components over the microbridge connections.

Processors are built out of five components types: memory, accumulators, replicator planes (for the transmission of global variables), counters and comparators. Each is fabricated as a WSI array of identical components. Only the memory and replicator wafers have in wafer connections between neighbouring elements.

Redundancy is of course required for reasonable wafer yields. This has been achieved by duplicating the PE at each microbridge site and connecting one of the survivors (or one of the neighbour's survivors, depending on the particular scheme) to each vertical bus line.

It is intended that the 3-D processor will operate at a speed of 10MHz. This is

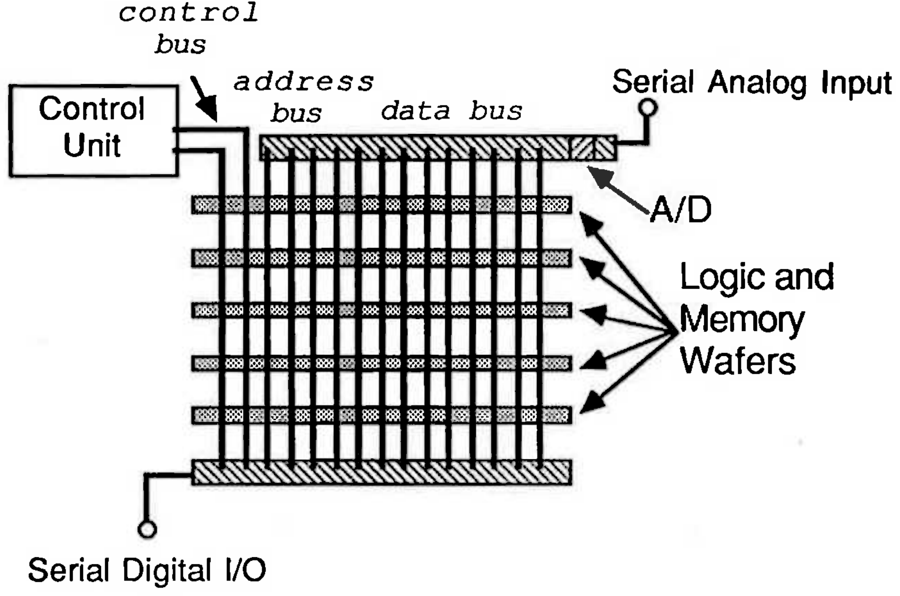


Figure 2.3: Basic structure of Hughes 3D computer

fast enough to enable a 256 × 256 matrix multiply to be carried out in 12.0 ms at an effective instruction rate of 10^4 mops.

Power dissipation was seen as a potential problem. It was reduced by by using CMOS exclusively, operating at a frequency of only 10MHz and having an architecture that uses only 2.5 functional planes per bus cycle. As a result no special cooling techniques are required.

Through the Wafer Optical Interconnect

As silicon is transparent at wavelengths above 1.1μm, AT&T Bell Laboratories have developed another solution to the 3D interconnect problem that uses free space optics [Hornak et al., 1986] [Hornak & Tewksbury, 1987].

Optics provides quite a different medium for communications than electronics. When compared to electronics, optical communications are characterized by very high speeds (up to 200MHz), larger driver and receiver sizes, a higher power consumption, and possibly a longer latency, depending on the received power. Size, power consumption and latency have tended to prevent optics from being used for planar on-chip interconnect. However when compared with electrical connections between wafers they become attractive, particularly when reliability, repairability, data

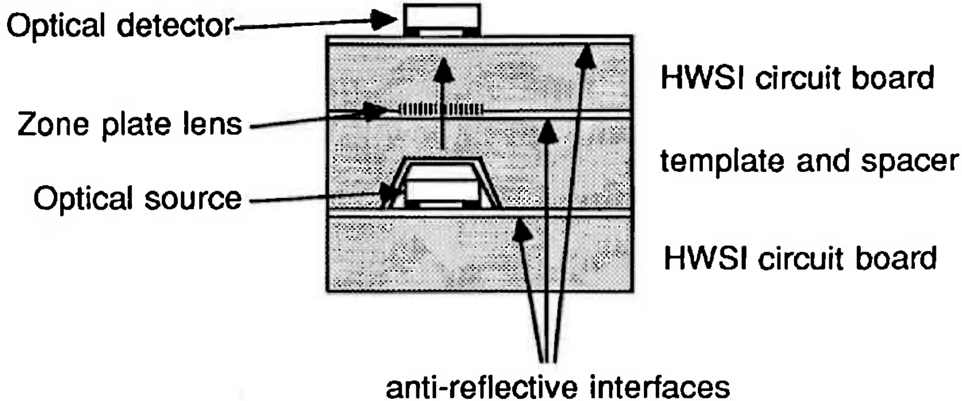


Figure 2.4: Through the wafer free space optical interconnect

rate and size are considered.

The concept being experimentally tested at AT&T Bell Laboratories is illustrated in Figure 2.4. In this arrangement, the optical sources are formed from an array of InGaAs/P 1×12 LEDs and the receiver is a matching p-i-n array. The linear array spacing is 250μm, which is about the same as regular bonding pads. To improve the optical coupling and reduce the crosstalk between the non-directionally radiating LEDs a "lens" array of Fresnel zone plates is placed between the emitters and sources.

Because the source and receiver are physically close (about 1mm) the optical signal at the receiver is strong enough that a low latency can be achieved, when compared with fibre-optic systems. High throughputs can also be obtained, though at the expense of high power requirements with 100mA of current being required to operate each source at top speed.

There is considerable scope however to reduce the power consumption by using a split and modulated central light source rather than multiple point sources. In the short term higher power density can be handled by circulating an appropriate coolant through the spacer. However it is anticipated that in early applications interconnect density will largely be limited by heat dissipation limitations.

Optical communications forms the core of the stacked wafer module as illustrated in Figure 2.5. The wafer stacks themselves can be joined together with 100 line/inch microcables connected through silicon micro-groove etched connectors.

A mesh-connected multicomputer architecture has been proposed for this technology as illustrated in Figure 2.6.

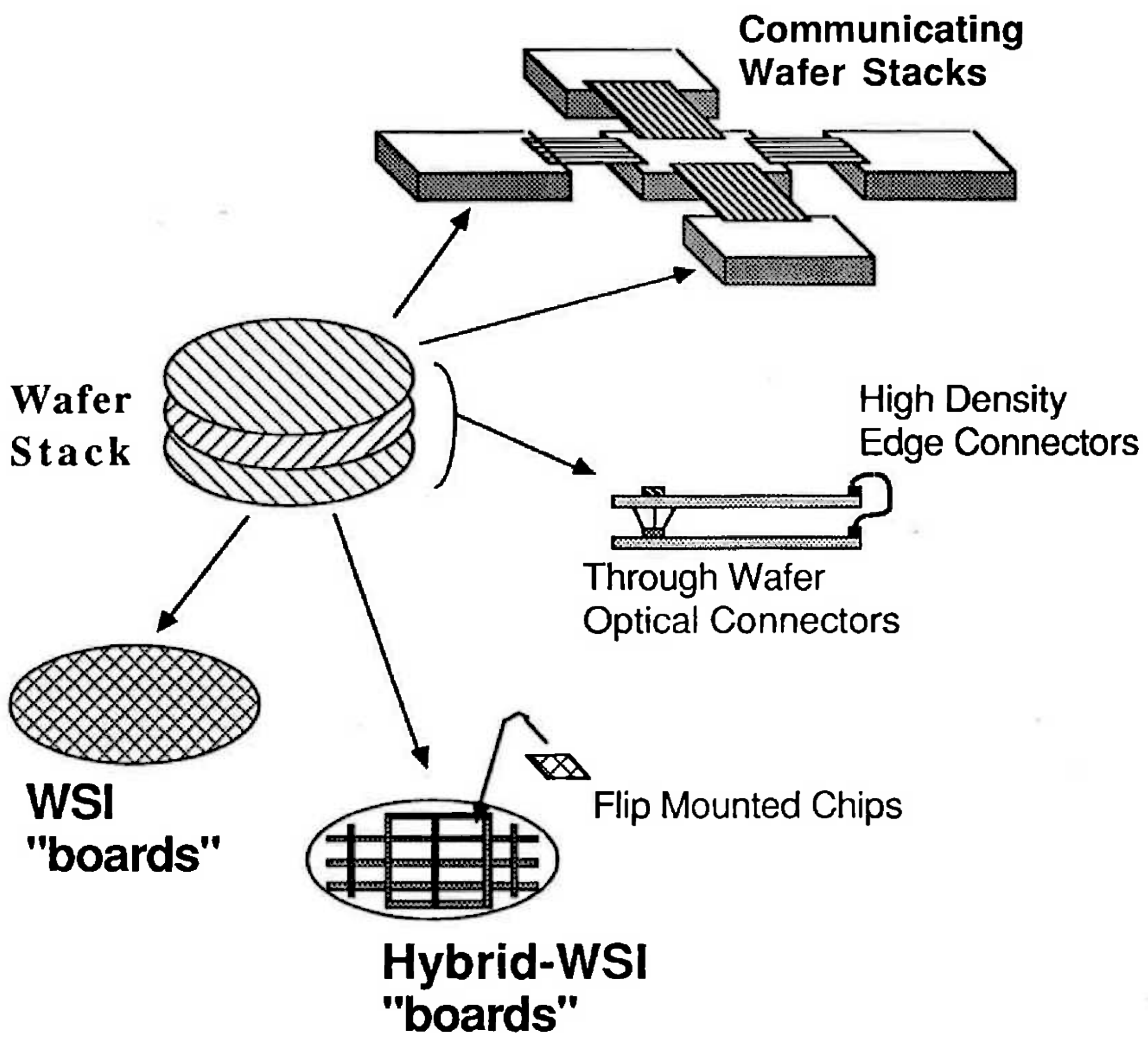


Figure 2.5: Integrating technologies to form a wafer stack

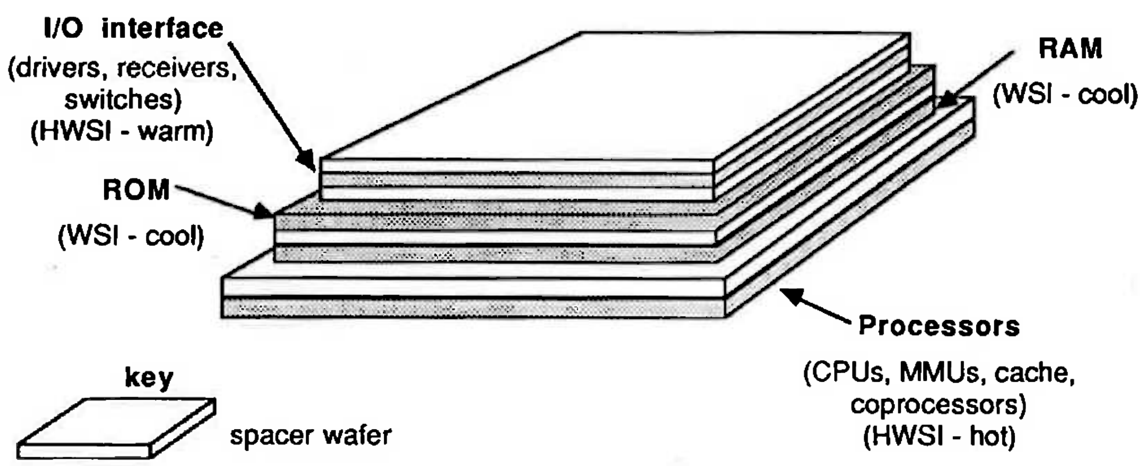


Figure 2.6: A future mesh-connected multicomputer

Chapter 3

Yield Modeling

3.1 Faults in Silicon

Physical failures in integrated circuits can be broadly divided into three classes [Abraham & Agarwal, 1986,Mangir, 1984]:

1. Failures induced during chip die fabrication;

2. Failures induced during packaging;

3. Failures that become apparent during field operation, due mainly to wear out mechanisms.

These classes are interrelated to some extent. In particular, defects introduced during fabrication or packaging may not manifest themselves as faults until some period of field operation has passed.

For chips most failures are manufacturing related. For example one large NMOS product experienced the following failure rates [Little, 1986]:

- 40% of the dies failed initial probe test after manufacturing;

- 0.5% – 3% of the packaged parts failed during burn-in;

- 0.01% of parts failed during shipping;

- 1% of parts failed during a 10 year product life.

The percentage of parts that pass the initial probe test of the unpackaged and unseparated dies is referred to as the *Yield*. Yield related failures manifest themselves in one of two forms [Mangir, 1984]:

- structural failures,
- performance failures.

Structural failures manifest themselves as a failure to meet functional specifications. Performance failures manifest themselves as a failure to meet speed or power requirements or as complete failure.

Defects come as either point defects, area defects or line defects. Not all defects result in failures, depending on where they fall. Approximately 60% of the yield loss on most lines is due to point defects. Approximately 30% of wafer defects come from the chemicals and the water used in the process [Dicken, 1988]. Point defects tend to result in structural failures and vary in size from sub-micron to over 100μm. Area defects cover large areas of the wafer and tend to be associated with performance failures. Line defects usually result from poor handling and are generally avoidable. Defects can arise from a number of sources:

1. *Photolithographic Defects.*

   Photolithographic defects result from extra or missing patterns introduced when the lithographic masks are used to generate the layout on the silicon. The main cause of photolithographic defects are air-borne or chemical-borne particles either adhering to the masks or to the wafers. Photolithographic defects are usually point defects and result in faults such as missing or extra metal patterns (leading to opens and shorts respectively.) Oxide pinholes are also caused by photolithographic defects. The main remedy for reducing photolithographic defect counts is a cleaner room. Automation can help in achieving this.

2. *Process Control Errors.*

   Defects can also result from process steps apart from photolithographic steps. During crystal growth, crystal defects may result in structural failures while chemical inclusions may act as recombination and generation centres and degrade performance. Contamination results in performance failures as do improper doping profiles and thin film errors.

Packaging related failures include lack of a hermetic seal, bonding failures, thermal coefficient mismatches, and lack of thermal dissipation paths resulting in mechanical stresses due to thermal expansion.

Field operation or *reliability*, as opposed to yield, failures exhibit failure rates that follow the well known reliability "bathtub curve". A typical failure rate bathtub curve

showing dominant reliability concerns and remedies for these concerns is reproduced from [Woods, 1986] in Figure 3.1. In general, yields and infant mortality rates tend to track each other as both are determined primarily by defect levels. Thus any effort to improve yield tends also to improve infant mortality rates. The relationship between yield and reliability is not quite as strong. For example, features that improve the yield of the metalization step may result in an earlier onset of electromigration failures [Woods, 1986].

Temperature considerations are also important if one is to consider reliability trends. As chips become denser, and the number of pins remains almost constant, chip power density will increase without a corresponding decrease in the thermal resistance to a heat sink. As reliability is seriously degraded by higher temperatures, more complex cooling techniques, such as direct die attachment and liquid or gaseous coolants, will become more important. Voltage scaling can of course be used to alleviate this.

Given the complexity of physical failure mechanisms in silicon circuits a very important issue is how to develop fault models that are both easy to use for the purpose of generating test vectors and also reflect the impact of the vast majority of physical failure mechanisms. A discussion of this problem is beyond the scope of this thesis.

3.2 Non Fault Tolerant Yield Models

Yield modeling concerns itself most with what the process engineer is trying to avoid – defects! This makes the formulation of satisfactory yield models difficult. However yield models play an important role in the semiconductor industry in three important ways:

1. *Wafer Probe Analysis.*

   The aim of Wafer Probe Analysis is to provide a tool to the process engineer to enable yield improvement. By quantitatively modeling actual device yield the major yield detractors can be more readily identified [Stapper et al., 1982].

2. *Device Yield Prediction.*

   Accurate device yield prediction enables most efficient use of a production line, particularly one producing a range of products. Yield prediction also enables one to determine product costings in advance, perhaps so as to make prod-

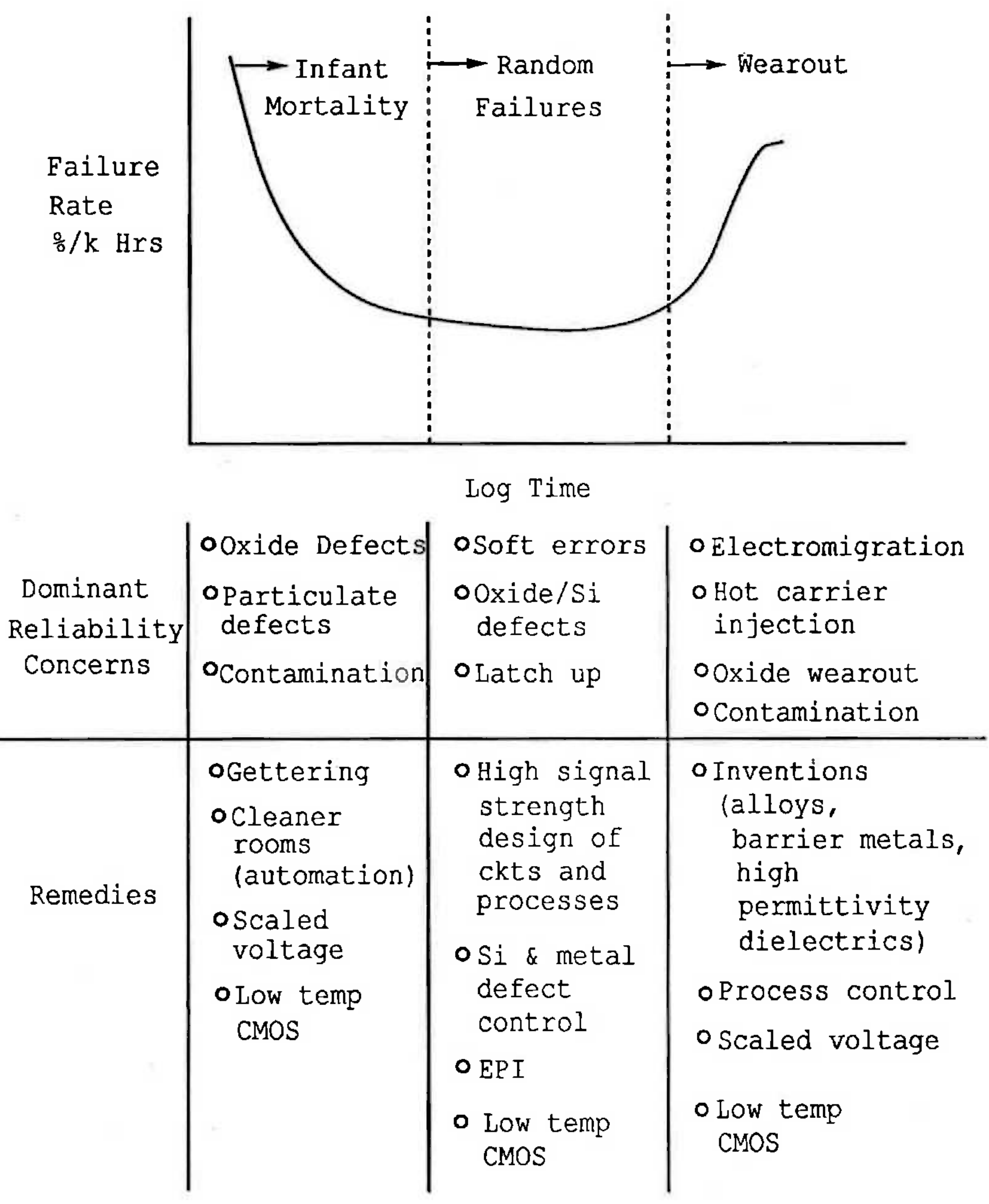


Figure 3.1: Reliability failure rate bathtub curve.

uct range choices and to determine ultimate capabilities, in particular largest economic chip size [Moore, 1970,Murphy, 1964].

3. *Technology Yield Prediction.*

   The object of technology yield prediction is to forecast the yield effect of making changes to a process line. Examples include design rule optimization and adding additional steps [Ipri & Sarace, 1977] [Turley & Herman, 1974] [Ipri, 1980].

Defect maps show two distinct defect types: *area* defects and *point* defects. As these two types are very noticeably different it has traditionally been accepted, and found useful, to model yield by an equation of the form [Stapper et al., 1983] [Paz & Lawson, 1977] [Mallory et al., 1983] [Ham, 1978]:

$$Y = Y_0 f(D_0, A, \alpha), \tag{3.1}$$

where

$$\begin{aligned} Y_0 &= \text{(1 - fraction of wafer suffering from area defects only)} \\ &\quad \text{or } \textit{Gross} \text{ yield}, \\ f(D_0, A, \alpha) &= \text{fraction of chips on wafer lost due to random defects}, \\ D_0 &= \text{average random defect density}, \\ A &= \text{chip area susceptible to defects (or critical area)}, \\ \alpha &= \text{measure of spread of defect density, } D, \\ &\quad \text{often fixed by the yield equation}, \end{aligned}$$

If random defects were purely (*ie.* homogeneously) random then the random defect yield would be a Poisson distribution and the probability of there being no defects in an area A would be:

$$Y = Y_0 e^{-D_0 A}. \tag{3.2}$$

However so called random defects have definite nonhomogeneous densities. Defect densities vary substantially between wafers and between different areas on a wafer. It has generally been observed that yields are a lot higher in the central half of the wafer than in the outer half of the wafer. A slight decrease in yield just in the centre of the wafer has also been observed. The yield ratio between edge and non-edge chip sites may even be as high as three or four to one [Yanagawa, 1969]. Furthermore it has been found that defects have a tendency to cluster in both small (*ie.* $O(100\mu)$) and

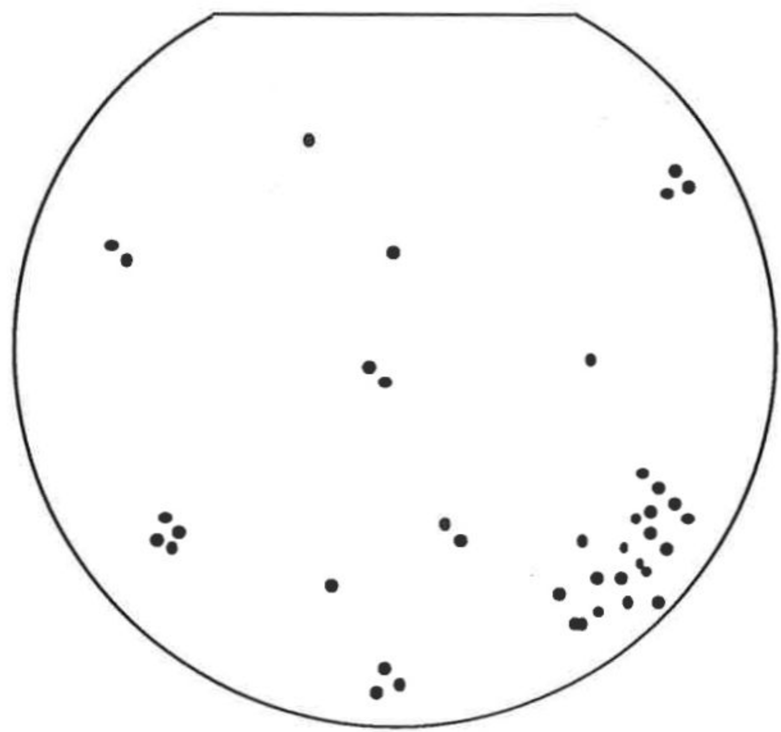


Figure 3.2: Example of defect clustering on a wafer

large (*ie.* over several chip sites) clusters. An example of clustering on a wafer is illustrated in Figure 3.2. With defects being distributed non-randomly, Poisson statistics becomes inapplicable. Even in the early days of LSI it was found that predictions made using Equation 3.2 proved to be overly pessimistic [Murphy, 1964,Moore, 1970]. It was hoped that in the VLSI era defects would become purely random, due to improved process control, but this has not occurred [Stapper, 1986b].

It has also been noted that observed fault distributions do not follow Poisson statistics. Poisson statistics would require that the distribution of the number of defects, k, in a given area, A, follow [Stapper, 1973]:

$$P(X = k) = e^{-D_0 A} \frac{(D_0 A)^k}{k!}, \tag{3.3}$$

with a variance equal to the mean. However it has been found that actual distributions measured on product lines have variance to mean ratios of anywhere between 4 and several 100 [Stapper, 1986b].

Murphy was the first to suggest that this deficiency in Poisson statistics could be handled by using compound Poisson statistics [Murphy, 1964]. He suggested compounding Poisson statistics with a distribution for D, $f(D)$:

$$Y = \int_0^\infty e^{-DA} f(D) dD. \tag{3.4}$$

He evaluated this for two forms of $f(D)$: the rectangular and triangular (which was intended to approximate Gaussian) distributions. These are given, with results obtained by using other distributions, in Table 3.1. Murphy's law, based on the triangular distribution, remained in use at Bell Laboratories and Western Electric, as well as on other process lines, until at least well into the 1970's.

The use of an exponential form for $f(D)$ was first suggested by Seeds [Seeds, 1967]. The resultant yield formula was later derived by Price using a different set of assumptions [Price, 1970].

Stapper proposed using a Gamma distribution for $f(D)$ [Stapper, 1973] based on a suggestion first published in [Okabe et al., 1972]. Stapper successfully fitted the resultant distribution function for the number of faults per wafer to a series of test wafers. The resultant yield equation,

$$Y = Y_0 \frac{1}{(1 + D_0 A/\alpha)^\alpha}, \tag{3.5}$$

has been in use at several IBM facilities, as well as other fabrication lines, since its introduction. Successful fit of this equation to experimental data has been reported

Distribution of D	$Y/Y_0 =$
0 D_0 D	e^{-D_0A} (Poisson)
0 D_0 $2D_0$ D	$\left(\frac{1-e^{-D_0A}}{D_0A}\right)^2$ (Murphy)
0 $2D_0$ D	$\frac{1-e^{-2D_0A}}{2D_0A}$
Exponential 0 D_0 D	$\frac{1}{1+D_0A}$
Gaussian sd 0 D_0 $\frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{1}{2}(\frac{D-D_0}{\sigma})^2}$ sd = σ = standard deviation	$\frac{1}{2}e^{(-D_0A+\frac{\sigma^2 D_0A}{2})}\times$ $\{1+\mathrm{erf}[\frac{1}{\sqrt{2}}(\frac{1}{\sigma}-\sigma D_0A)]\}$
sd^2 Gamma D_0 $\frac{\alpha}{\sqrt{\alpha}}(\alpha\frac{D}{D_0})^{\alpha-1}e^{-\alpha\frac{D}{D_0}}$ $sd^2=\sigma^2=1/\alpha$	$\frac{1}{(1+D_0A/\alpha)^{\alpha}}$

Table 3.1: Yield formula for different defect density distributions

several times [Okabe et al., 1972] [Turley & Herman, 1974] [Paz & Lawson, 1977] [Dingwall, 1968], which is significant considering how little data is published in this usually proprietary area.

Some authors have objected to Murphy's conjecture that a continuous Probability Distribution Function (PDF) can be assigned to the defect density. In particular Hu has argued quite vehemently that the defect density distribution is best described as non-continuous. He points at clustering as being a major reason for this [Hu, 1979,Hu, 1984]. Hu claims that using compound Poisson statistics results in optimistic yield predictions, particularly for VLSI chips. Thus, Hu suggests, yield can only be determined by dividing each wafer into regions, determining the average defect density for each region, applying Poisson statistics to each region and summating the results. Other authors to suggest that this, or a similar, approach is preferred include Warner [Warner, 1974,Warner, 1981] and Hemmert [Hemmert, 1981]. Hu fails to suggest a method for securing the necessary data for this very detailed yield model. This would require considerable effort.

Other spatial approaches to yield modeling have also been discussed. Gupta has suggested that variations in defect density be modeled by radial and angular distributions (thus ignoring wafer to wafer variations) [Gupta & Lathrop, 1972,Gupta et al., 1974]. Warner and Hemmert suggested a "window method" of prediction, where yield predictions are made on the assumption that chip failure rates for smaller chips can by mapped, on an area basis, onto larger chips by treating these larger chips as groups, or (paned) windows, of the smaller ones. The obvious difficulty of this approach is that small chips will generally have a larger percentage area, particularly in the pads, that is relatively empty and thus less susceptible to defects. This will result in underestimates of yield. Further objections to these approaches can be found in [Stapper, 1975,Stapper, 1981].

In practice however Stapper does indicate that several fabrication plants do model yield by dividing wafers into two, three or five regions, applying Poisson statistics to each region [Stapper, 1985]. However representing $f(D)$ by a continuous function, namely a Gamma function, can be shown to be consistent with this regionalizing [Stapper, 1976] [Paz & Lawson, 1977] [Stapper, 1985]. These references show that by using a Gamma distribution for $f(D)$, regional and wafer to wafer variations in defect density are properly accounted for. In [Stapper, 1986b] it was also shown that the effects of clustering are also correctly handled by using a Gamma distribution. In fact the clustering of defects and the variation in clusters between regions and wafers, particularly the wafer to wafer variations, are the main cause of the inadequacy of

Poisson statistics.

Stapper [Stapper, 1986b] has suggested that Equation 3.5:

$$Y = Y_0 \frac{1}{(1 + AD_0/\alpha)^\alpha} \tag{3.6}$$

should be replaced by:

$$Y = \frac{1}{(1 + AD_0/\alpha)^\alpha}, \tag{3.7}$$

with α being a function of area A, for redundancy calculations. Y_0 can be removed from the yield equation because the largest contribution to large area defects can be treated as, or often are, large clusters. Thus Y_0 can be removed by appropriately setting D_0 and α. α becomes a function of A so that larger areas are correctly handled. This makes defect tolerant yield modeling more straightforward. When modeling for redundancy, in fine grain circuits, (Stapper's main fault tolerant interests are in high density DRAMs) it does not directly matter whether a defect is part of a cluster or not as each defect will affect one circuit only and require the introduction of just one spare. This may not be true however for coarse grain circuits as small clusters of several defects could result in one faulty circuit only. It may be the case that as the grain size of redundant circuits approaches that of ICs, declustering through the use of a Y_0 factor may again become appropriate. Fault tolerant yield modeling will be discussed in detailed in the next section.

Typical values for the parameters D_0 and α are difficult to obtain due to their proprietary nature. In [Moore & Day, 1984] it was stated that for a $2\mu m$ line D_0 should not exceed 0.1 defects/mm^2. Typical values for α lie in the range from 0.25 to 4 depending on the defect mechanism [Stapper et al., 1980,Stapper, 1986a], with $\alpha = 1$ being a good approximation for for most defect types at IBM factories [Stapper, 1985]. As a product line matures it is found that the yield improves, D_0 decreases and α decreases indicating fewer, more highly clustered defects.

Despite all this work on analytic yield models many foundries still find empirical models more satisfactory. Equation 3.5 was used by Dingwall [Dingwall, 1968] and TRW [Warner, 1981], with $\alpha = 3$ before Stapper placed it on an analytic footing. Other foundries have used equation 3.5 simply because it tends to provide a good fit on an empirical basis.

An empirical model of the form

$$Y = e^{-\sqrt{D_0 A}} \tag{3.8}$$

has been used by Moore and others at Intel [Moore, 1970,Warner, 1981]. IBM has also used similar models with different fractional powers [Stapper et al., 1983]. Other

foundries have also reported its use. NCR has used a yield model consisting of a weighted average of Murphy's and Seed's models [Gulett, 1981].

Another empirical model was given in [Hurst, 1985]:

$$Y = \frac{1 - e^{-(A/A_0)}}{A/A_0} \times 100\% \tag{3.9}$$

where A_0 is the wafer area per defect which may be taken as (1985 figures) $1 \times 10^7 \mu m^2$ for bipolar technology, and $2 \times 10^7 \mu m^2$ for MOS technology.

In practice it is certainly the case that companies will use the yield model that correlates best with their actual experience. However a model with a purely empirical basis is of no use to the fault tolerant yield modeler as a fault distribution is required. Discussion will continue using analytic models only.

3.3 Fault Tolerant Yield Modeling

Fault tolerant yield modeling allows one to determine the level of spares required in a product. The most elemental approach would be to use Binomial statistics. *ie.* If, for a chip containing M circuits each of yield Y_1, the non redundant yield Y is,

$$Y = Y_1^M, \tag{3.10}$$

then the addition of R redundant circuits or spares would increase the yield to [Chen, 1969,Tammaru & Angell, 1967]

$$Y = \sum_{n=0}^{R} \frac{(M+R)!}{(M+R-n)!n!} Y_1^{(M+R-n)} (1 - Y_1)^n. \tag{3.11}$$

This approach assumes that the average number of defects is the same in each circuit and thus will be highly over-optimistic in practice.

In general, the correct expression for determining fault tolerant yield is derived from a form like

$$Y = \sum_{x=0}^{\infty} Pr[x \text{ defects}] Pr[x \text{ defects being fixable}], \tag{3.12}$$

where the first probability $Pr[x \text{ defects}]$ is calculated from the defect statistics and the second probability $Pr[x \text{ defects being fixable}]$ is determined by the the properties of the scheme used to implement fault tolerance.

If the device is made up of more than one module type then the yield would be expressed as (for the two module example)

$$\begin{aligned} Y &= \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} Pr[i \text{ defects in module type 1}] Pr[j \text{ defects in module type 2}] \\ &\quad \times Pr[\text{fixing device with defect distribution } (i,j)] \end{aligned} \tag{3.13}$$

In both of the above equations the last probability, Pr[fixable], is a function of two other probability distributions, the expected number of failed modules as a function of the number of defects, and the probability of being able to fix the device with this number of failed modules. Expressions for these will be given later.

The first probability, $Pr[x$ defects], is a property of the yield model used. For a Poisson distribution the probability distribution function (PDF) of getting x defects is

$$P(X = x) = \frac{D_0 A}{x!} e^{-D_0 A}. \tag{3.14}$$

Using this in Equation 3.12, with appropriate assumptions would result in equation 3.11.

For the situation where the Poisson distribution is compounded by a Gamma function the probability of getting x defects in a chip of area A is

$$P(X = x) = \frac{\Gamma(x + \alpha)(D_0 A/\alpha)^x}{x!\Gamma(\alpha)(1 + D_0 A/\alpha)^{\alpha + x}} \tag{3.15}$$

where $\Gamma(x)$ is the Gamma function.

A review of how this model can be applied to fault tolerant chip memories will be discussed in the next section.

3.4 Fault Tolerant Chip Memories

In fault tolerant chip memories redundancy is provided through spare rows and spare columns in the memory array. As the individual cells are so small it is reasonable to assume that each defect falls in a different cell. A detailed description of how to model fault tolerant yields in DRAMs is given in [Stapper et al., 1980].

As a simpler case, and as a prelude to Section 3.7, the case for static RAMs will be described here. It would be expected that each fault would manifest itself in one of four ways:

1. Single cell failure;
2. Row failure (row address line or address decoder);
3. Column failure (bit line or sense amplifier or column decoder if present);
4. Chip kill failure (fault is not repairable by provided redundancy. eg. clocks, power distribution, pads, etc.)

Due to the small size of the cells in higher density circuits, such as DRAMs, multiple cell and multiple column and row failures would also be common in these circuits. In current SRAMs yield simulations have shown these to be relatively unlikely due to the larger size of the cells. Each of these four failures 1–4 above would have a critical area associated with them: $A_{cell}, A_{row}, A_{col}$ and A_{kill} respectively. The total critical area would be $A_{tot} = A_{cell} + A_{row} + A_{col} + A_{kill}$. The probability of getting i single cell failures, j row failures and k column failures would be given by

$$\begin{aligned} P(i,j,k) &= \frac{\Gamma(i+j+k+\alpha)}{\Gamma(\alpha)} \\ &\times \frac{(1/\alpha)^{i+j+k}}{(1+D_0A/\alpha)^{i+j+k+\alpha}} \\ &\times \frac{(D_0A_{cell})^i(D_0A_{row})^j(D_0A_{col})^k}{i!j!k!} \end{aligned} \tag{3.16}$$

The yield can then be determined by adding up the respective $P(i,j,k)$s over fixable combinations of failures. For example if the memory had one spare row and one spare column the yield would be

$$Y = P(000)+P(100)+P(200)+P(010)+P(110)+P(011)+P(001)+P(101) \tag{3.17}$$

Stapper [Stapper, 1982] noted that it wasn't until one reached the scale of 256k DRAMs that it was necessary to consider dividing the defects into their component types (metallization, diffusion, pinhole etc.) and using a yield equation based on the form (ignoring redundancy)

$$Y = Y_0 \prod_{m=1}^{M} (1 + A_m D_{0m}/\alpha_m)e^{-\alpha_m} \tag{3.18}$$

where A_m is the critical area sensitive to defects of type m, D_{0m} is the mean defect density of defect type m, and α_m is its clustering parameter.

The concepts used for modeling fault tolerant memory yield can be extended to yield modeling for processor arrays.

3.5 Yield Modeling for Processor Arrays

Yield modeling for fault tolerant memories only had the aim of determining the appropriate level of redundancy, the method of introducing that redundancy generally being via spare rows and columns (with perhaps the use of spare blocks.) For processor arrays however the elements are larger and organized differently so more complex

schemes for providing redundancy are required. These schemes usually involve a reconfiguration step using extra wiring provided in the array. Thus here the aims are a little broader in that the yield model may be used to decide upon a redundancy approach as well as the amount of redundancy required.

Thus the model should consider three additional effects:

1. The replicated cells are larger so there is a non zero probability of two or more defects occurring in the same cell;
2. Often all the spares cannot be used to replace a faulty unit. *ie.* The array cannot always be *reconfigured* to make effective use of spares;
3. The wiring used to enable reconfiguration is itself prone to defects and thus needs to be considered in calculating the yield.

Typically only one approach, based on yield, is discussed. In the research being reported on here it has been found that there are two approaches that can be useful when investigating different aspects of expected fabrication line output for processor arrays, particularly wafer scale arrays:

1. *Yield.*

   The first approach is to determine the expected yield of an array requiring P working elements with an additional R spares provided, making a total of N elements fabricated. This would be the most common approach to redundant array modeling. Yield modeling would be used to determine one of three objectives:

   (a) The number of spares required to optimize the yield of a product requiring P processors. This would thus determine $N = P + R$ and the required die size.

   (b) When a full wafer is used the total number of processors N is fixed. The nature of the product may require that P be known. For example the application software may require this. In this case the yield model is required so as to determine for what value of P the maximum yield is obtained.

   (c) The wafer to wafer spread in the number of defects per wafer is usually higher than the spread within any one wafer, so the resultant spread in the distribution of the number of processors that survive on a wafer may

be quite high. With this situation it would be more reasonable to aim at producing a product with a range of P, possibly a number of discrete values. A yield model would be used to determine the most suitable values that could be expected. In memory products these are often referred to as "partial products" [Stapper et al., 1980]. Considering this spread, it may be best to take approach 2 below at least for evaluation purposes.

2. *Expected number of PEs.*

   At least for the purposes of comparing architectures and fault tolerance schemes a better approach than 1(c) above would be to calculate the average expected number of Processing Elements (PEs) in the final arrays. It may be desirable to exclude low yielding wafers from this calculation as they would most likely be uneconomic to package. The primary difference between this approach and 1(c) above, for a practical product, is that a continuum of PE numbers is acceptable here whereas in the above approach only certain numbers of PEs per wafer are acceptable. This form of calculation also better reflects the actual situation. In many applications a wide range of array sizes are acceptable. As there is a significant spread in the number of defects per wafer, a product range that can handle this spread will be preferable.

3.5.1 Yield Formulation

The expressions for yield will be developed first, the expression for the expected number of PEs being an extension of that for yield.

From equation 3.12, using equation 3.15 for the defect distribution, the yield of an array of N identical PEs, R of which are spare would be expressed as [Koren & Breuer, 1984] [Franzon, 1986b]

$$Y = \sum_{k=0}^{\infty} \frac{\Gamma(k+\alpha)}{\Gamma(\alpha)} \frac{(1/\alpha)^k}{(1+D_0A/\alpha)^{k+\alpha}} \frac{(D_0A_{mod})^k}{k!} P_{kNR} \tag{3.19}$$

where

A = total critical area susceptible to defects

A_{mod} = critical area of replaceable modules only

P_{kNR} = Probability of successful repair with k defects

P_{kNR} can be expressed as

$$P_{kNR} = \sum_{j=1}^{min(k,R)} Q_{kjN} C_{jNR} \tag{3.20}$$

where

$$
\begin{aligned}
C_{jNR} &= \text{Probability of fixing the array with } j \text{ faulty PEs and } R \text{ spares} \\
Q_{kjN} &= \text{Probability of } k \text{ defects being distributed in } j \text{ faulty modules}
\end{aligned}
$$

The C_{jNR} term represents the ability of the reconfiguration scheme to be able to repair the array with $j \leq R$ faulty PEs present. It is either determined analytically for simple arrays or estimated through Monte Carlo simulation. It should also normally account for the less than perfect fault coverage of the testing scheme used to determine which PEs are faulty.

The original formulation of Q_{kjN} [Mangir, 1984] had some inaccuracy so a new form will be presented here. Under the assumption that a defect is equally likely to fall anywhere on the array (clustering will be considered in Section 3.5.3) the Q_{kjN} term can be expressed in the form [Franzon, 1985,Franzon, 1986b]

$$Q_{kjN} = \theta_{kj} \prod_{i=0}^{j-1} (\frac{N-i}{N}) \frac{1}{N^{k-j}} \tag{3.21}$$

where θ_{kj} is a parameter obtained recursively by

$$
\begin{aligned}
\theta_{kj} &= j\theta_{k-1,j} + \theta_{k-1,j-1}, \text{ with starting conditions} \qquad (3.22) \\
\theta_{11} &= 1, \text{ and} \\
\theta_{kj} &= 0 \text{ for } k < 1, j < 1 \text{ or } j > k
\end{aligned}
$$

A result of these expressions is that $\theta_{kj} = 1$ for $k = j$. Additionally it is observed that

$$Q_{kjN} = 0 \text{ for } j > N \text{ and } j < 1 \tag{3.23}$$

The derivation of this form of Q_{kjN} proceeds as follows [Franzon, 1985] (dropping the N subscript for convenience):

With N modules present if 1 defect falls within these modules (producing a fault) then the probability of producing 1 faulty module is obviously

$$Q_{11} = 1. \tag{3.24}$$

If a second defect falls then

$$
\begin{aligned}
Q_{21} &= Pr[\text{2nd defect falls in same module as first}] \\
&= \frac{1}{N} \qquad (3.25) \\
Q_{22} &= Pr[\text{2nd defect falls in a different module}] \\
&= \frac{N-1}{N}. \qquad (3.26)
\end{aligned}
$$

For a third defect one obtains

$$\begin{aligned}
Q_{31} &= Q_{21} \times \\
&\quad Pr[\text{3rd defect falling in same module again}] \\
&= (\frac{1}{N})^2 && (3.27) \\
Q_{32} &= Q_{22} \times \\
&\quad Pr[\text{3rd defect falling into one of two already defective modules}] \\
&\quad +Q_{21} \times \\
&\quad Pr[\text{3rd defect falling into a non faulty module}] \\
&= (\frac{2}{N})(\frac{N-1}{N}) + (\frac{N-1}{N})(\frac{1}{N}) \\
&= (\frac{3}{N})(\frac{N-1}{N}) && (3.28) \\
Q_{33} &= Q_{22} \times \\
&\quad Pr[\text{3rd defect falling in a fault free module}] \\
&= (\frac{N-1}{N})(\frac{N-2}{N}) && (3.29) \\
& && (3.30)
\end{aligned}$$

continuing in a similar fashion

$$\begin{aligned}
Q_{41} &= (\frac{1}{N})Q_{31} \\
&= (\frac{1}{N})^3 && (3.31) \\
Q_{42} &= (\frac{2}{N})Q_{32} + (\frac{N-1}{N})Q_{31} \\
&= (\frac{7}{N^2})(\frac{N-1}{N}) && (3.32) \\
Q_{43} &= (\frac{3}{N})Q_{33} + (\frac{N-2}{N})Q_{32} \\
&= (\frac{6}{N})(\frac{N-2}{N})(\frac{N-1}{N}) && (3.33) \\
Q_{44} &= (\frac{N-3}{N})Q_{33} \\
&= (\frac{N-3}{N})(\frac{N-2}{N})(\frac{N-1}{N}) && (3.34) \\
& && (3.35)
\end{aligned}$$

Higher order expressions for Q_{kjN} can be obtained similarly.

The product part of Equation 3.21 arises naturally from the terms involving N that appear in obtaining these expressions. The coefficient θ_{kj} arises from the ob-

served recurrence

$$Q_{k,j} = jQ_{k-1,j} + Q_{k-1,j-1} \tag{3.36}$$

with appropriate boundary conditions.

3.5.2 Expected Number of PEs

The expression for calculating the expected number of surviving PEs $E(P)$ in an array is given by:

$$E(P) = \sum_{i=1}^{N} iP_c(i) \tag{3.37}$$

where N is the number of PEs in the array, and $P_c(i)$ is the probability of i PEs being connected into the array:

$$P_c(i) = \sum_{j=i}^{N} C_{ij}T(j). \tag{3.38}$$

C_{ij} is the conditional probability of a maximum of i PEs being connectable into an array where j of the PEs are good, and $T(j)$ is the probability of j PEs being good. C_{ij} can be determined by Monte Carlo simulation of the particular fault tolerance scheme. $T(j)$ is given by

$$T(j) = \sum_{k=0}^{\infty} Q_{k,N-j,N} Pr[k \text{ faults}] \tag{3.39}$$

where $Q_{k,N-j,N}$ is the probability that k faults produces $(N-j)$ faulty PEs as given by Equation 3.21, and $Pr[k$ faults$]$ is the probability of k faults and is given in Equation 3.15. Substitution of Equation 3.39 into Equation 3.38 and Equation 3.37 gives

$$E(P) = \sum_{i=1}^{N} i \sum_{j=i}^{N} C_{ij} \sum_{k=0}^{\infty} Q_{k,N-j,N} Pr[k \text{ faults}] \tag{3.40}$$

Since $C_{ij} = 0$ for $j < i$ equation 3.38 can also be written as

$$P_c(i) = \sum_{j=1}^{N} C_{ij}T(j) \tag{3.41}$$

Furthermore an important parameter that is related to C_{ij} is the *Utilization* U which is defined as *that percentage of good PEs that can be effectively used by the fault tolerant array,* or

$$U(j) = \frac{E(i)}{j} = \frac{\sum_{i=j}^{N} iC_{ij}}{j} \tag{3.42}$$

Equation 3.40 can then be written as

$$\begin{aligned} E(P) &= \sum_{i=1}^{N} i \sum_{j=1}^{N} C_{ij} T(j) \\ &= \sum_{j=1}^{N} \sum_{i=1}^{N} i C_{ij} T(j) \\ &= \sum_{j=1}^{N} j U(j) T(j) \end{aligned} \qquad (3.43)$$

which is easier to calculate as accurately determining $U(j)$ requires fewer Monte Carlo simulations than determining C_{ij} does. For ease of summation on a computer $E(P)$ is best expressed as

$$E(P) = \sum_{k=0}^{\infty} Pr[k \text{ faults}] \sum_{j=1}^{N} Q_{k,N-j,N} j U(j) \qquad (3.44)$$

and the summation over k can be terminated at $2N$ in practice without loss of accuracy due to the small contribution of higher terms.

3.5.3 Effects of Clustering

The expression developed for Q_{kjN} in Equation 3.21 assumes that defects do not cluster. *ie.* that the probability of an additional defect falling within an area, already containing say x defects, is proportional to the area S only, or

$$P[x+1 \text{ defects}|x \text{ defects}] = cS \qquad (3.45)$$

where c is a constant and S is the area under consideration. It has been shown [Stapper, 1984b] that the yield Equation 3.5 could be derived from an assumption introducing a form of clustering. The assumption was that the probability of finding a defect in an incremental area ΔS is given by

$$Pr[1, \Delta S|x, S] = (c + bx)\Delta S \qquad (3.46)$$

where x is the number of defects in the neighbouring area S and b is a parameter reflecting the level of clustering. This produced the required yield formula with

$$\alpha = c/b \qquad (3.47)$$

verifying α's role as a clustering parameter. An adaptation of Equation 3.46 is used here to provide a method of including the effects of clustering in Q_{kjN}.

Now it is very difficult to apply a clustering parameter of this form to obtain new expressions for Q_{kjN} as the derivation of these expressions did not keep track of the number of defects x already in each PE.

Instead the effects of clustering can be introduced into Q_{kjN} by keeping track of the average number of clustered faults in each PE, k/j. Thus the probability of an additional fault clustering, with k faults already clustered into j cells, is enhanced by an additional bx/c factor:

$$(\text{clustered})Q_{k+1,j,N} = \left(1 + \frac{b\,k}{c\,j}\right)(\text{unclustered})Q_{k+1,j,N} \tag{3.48}$$

ie. Q gets replaced when $k > j$ with:

$$(\text{clustered})Q_{k,j,N} = \left(1 + \frac{(k-1)/j}{\alpha}\right) \times (\text{unclustered})Q_{k,j,N} \times (\text{Normalizing Factor}) \tag{3.49}$$

the normalizing factor being required to satisfy the requirement for Q_{kjN} that

$$\sum_{j=1}^{N} Q_{kjN} = 1 \tag{3.50}$$

As the average level of clustering over all clustered and non-clustered faults is used this would turn out to be an underestimate. A more accurate expression for the influence of clustering is probably not worth developing as the effect of using equation 3.49 to gain the required results, instead of Equation 3.21, was found to be typically less than 3%.

3.5.4 Comparison with other fault tolerant yield formula

Two previous authors [Mangir, 1984,Koren & Pradhan, 1985] have attempted to obtain a yield expression for fault tolerant arrays of PEs. Both however produced expressions for Q_{kjN} that did not satisfy Equation 3.50. Since then however Koren has obtained a new correct form for Q_{kjN} [Koren & Pradhan, 1986]:

$$Q_{kjN} = \sum_{l=0}^{j}(-1)^l \binom{N}{l,\, j-l,\, N-j} \left[\frac{j-l}{N}\right]^k \tag{3.51}$$

where $\binom{N}{l,\, j-l,\, N-j} = \frac{N!}{l!(j-l)!(N-j)!}$ is the multinomial coefficient.

3.6 Critical Area Determination

3.6.1 Purpose of Critical Area Determination

Any yield expression requires two inputs: defect statistics, in the form of D_0, α, and Y_0; and the *Critical Area A* subject to defects of the device being modeled. The former can be obtained from defect monitors [Lukaszek et al., 1986,Buehler, 1985] or (more commonly) from product data. The area is often taken as the die area which of course may be inaccurate as different products have different layout densities. To account for this the notion of critical area or susceptible area was suggested, *ie.* the chip area that is critically affected by or is susceptible to defects.

The simplest approximate method for calculating critical area is to calculate the die area and subtract the area consumed by the pads (which can be assumed to be relatively defect free) and the white space area (extensive regions with no circuitry or wiring.) Another, slightly more accurate method would be to calculate the total mask area, *ie.* the total area covered by metalization, diffusion or polysilicon.

These simplified methods are suitable for most yield calculations. However there are some situations where more accurate critical area figures are required. In particular any yield monitoring work, especially for process design, requires accurate critical area determination (though care should be taken to ensure that the resultant defect statistics can be used with other critical area calculation approaches that are employed in the products.) Some fault tolerant designs require accurate critical area calculation, especially high density DRAMs [Stapper et al., 1980]. The detailed prediction of product yields may also require more detailed critical area calculations.

3.6.2 General Methodologies and Related Work

Different defect types have varied impacts on the critical area of a chip. Photolithographic defects arise when dust and other particles produce either missing or extra patterns, resulting in opens or shorts respectively. Some examples are given in Figure 3.3. Critical area determination for photolithographic defects require conductor spacing data as well as conductor width data. Pinhole defects, on the other hand, can only occur in the areas of the relevant oxides that separate two conductive layers. Thus the critical area for pinhole defects is simply the relevant oxide areas.

Critical area determination for photolithographic defects is approached by first assuming that defects are circular in shape with a determined size distribution. The size distribution is normally assumed to take a $1/\chi^3$ form for defect diameter χ above

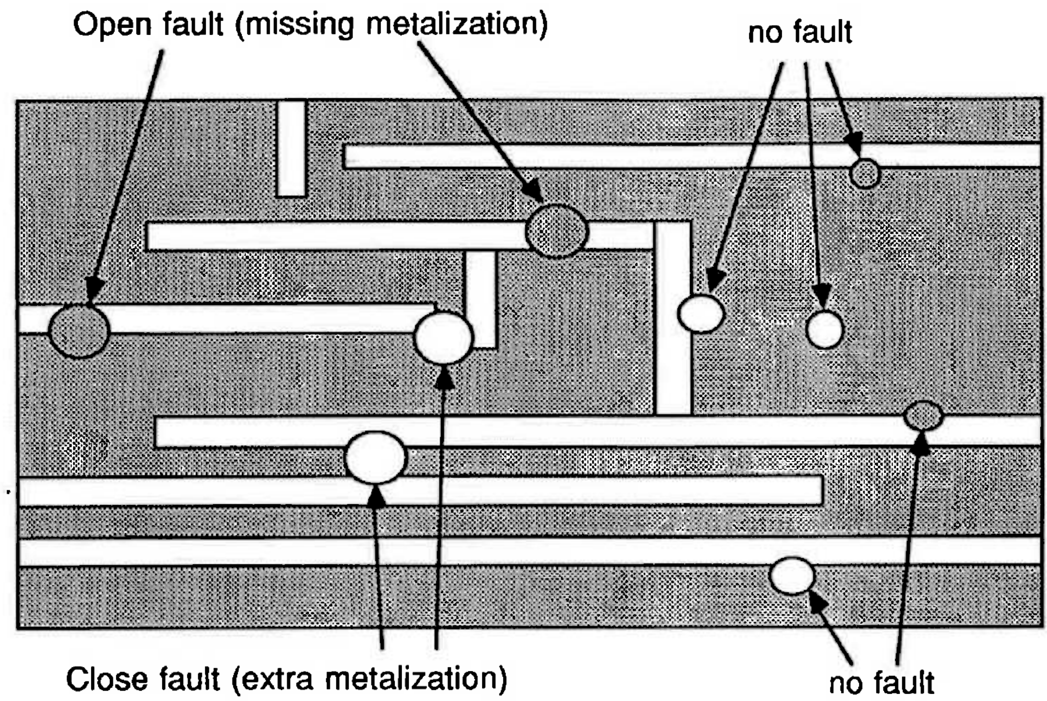


Figure 3.3: Examples of photolithographic defects

a certain threshold as illustrated in Figure 3.4. This distribution was determined from a long series of observations at IBM plants [Stapper, 1984a]. However there is no guarantee that this distribution will remain the same as devices are scaled to sub-micron limits [Ferris-Prabhu, 1988].

The first step is to obtain the critical area as a function $A(\chi)$ of defect size χ, and then integrate this with the defect size distribution. For simple structures $A(\chi)$ can be determined analytically. For more complex structures Monte Carlo simulation or an estimation technique is required. With the Monte Carlo simulation technique defects are placed randomly on a layout and the percentage of defects of each size that result in faults are determined. This percentage is multiplied by the area to give $A(\chi)$. This approach has been explored by this author and independently by Walker [Walker, 1986] [Walker & Director, 1986]. The same method has also been used by Stapper to determine memory yields [Stapper et al., 1980].

The analytic approach can only be used exactly for simple and regular structures such as parallel lines. For example the critical area for a long conductor of width w and length L is given by [Stapper, 1983]

$$A(\chi) = 0 \qquad \text{for } 0 <= \chi <= w \tag{3.52}$$

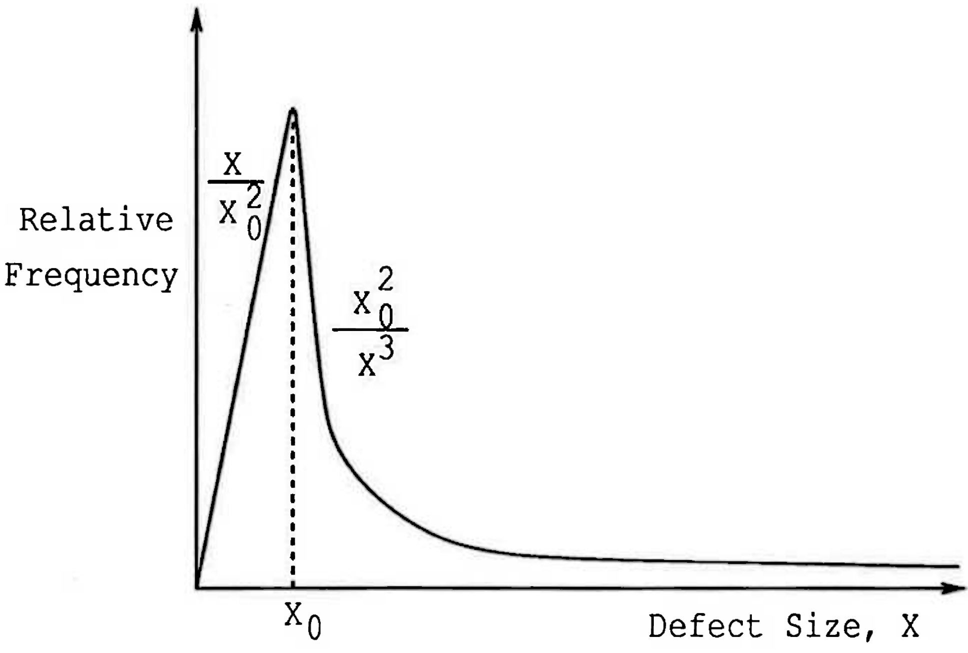


Figure 3.4: Defect Size Distribution

$$A(\chi) \;=\; L(\chi - w) \qquad \text{for } w <= \chi <= \infty. \tag{3.53}$$

Then, referring to the defect distribution in Figure 3.4, if $\chi_0 < w$ the critical area A_{crit} is given by

$$A_{crit} \;=\; \int_w^\infty A(\chi)P(\chi)d\chi \tag{3.54}$$

$$= \int_w^\infty L(\chi - w)\frac{\chi_0^2}{\chi^3}d\chi. \tag{3.55}$$

where $P(\chi) = \frac{\chi_0^2}{\chi^3}$ is the probability of the defect taking size χ. Critical areas for other simple structures, and further discussion on this approach, can be found in [Stapper, 1984a] [Ferris-Prabhu, 1985b] [Ferris-Prabhu, 1985a].

Maly attempted to extend the analytic approach to more complex structures by approximating each leaf cell by a set of parallel lines of different widths and spacings referred to as a "Virtual Artwork" [Maly, 1985].

In [Chen & Strojwas, 1987] another estimation technique to determine the critical area for more complex structures is described. This technique involves scanning the cells to determine defect effects and, like most estimation techniques, is faster than Monte-Carlo simulation.

It should be noted that in order to use detailed critical area results in a design the defect density of each layer must be known separately.

The Monte Carlo simulation approach for determining A_{crit} is often referred to as yield simulation [Walker, 1986]. Section 3.7 will discuss the defect tolerant design of a 200,000 transistor chip. Before presenting the details of that design, the yield simulator that was developed for determining its critical areas will be briefly discussed. Section 3.11 will discuss results concerning the general applicability of yield simulation techniques to defect tolerant chip design.

3.6.3 Critical Area Determination for the Transform and Filter Brick

A yield simulator was developed primarily with the aim of determining the optimal redundancy for the Transform and Filter Brick (TFB), a 200,000 transistor signal processing device designed at the University of Adelaide [Eshraghian et al., 1985]. TFB's floorplan is given in Figure 3.5.

The yield simulator used the Monte Carlo simulation approach as described above. The 2μm design was parsed into memory on a 0.5μm grid and then circular defects of various sizes were placed on this grid. The area covered by the defect was then tested for faults. The percentage of defects that resulted in faults, for a particular defect size then determines the critical area $A(\chi)$, as the same percentage of total area.

3.7 Determination of Optimal Redundancy for TFB

The Transform and Filter Brick (TFB) is a 200,000 transistor, CMOS, program controlled, parallel arithmetic processor for signal processing applications [Eshraghian et al., 1985]. It is designed to perform a target set of real time signal processing tasks such as correlation, convolution, adaptive and nonadaptive filtering, and Fast Fourier Transforms, the latter at a predicted speed of 800,000 butterflies per second. With such a large number of transistors the expected yield would be low. Thus any yield increase arising from redundancy would be most welcome! From a consideration of the floorplan the following redundancy schemes become possible candidates for implementation:

1. The provision of two spare ALUs on the ring bus.

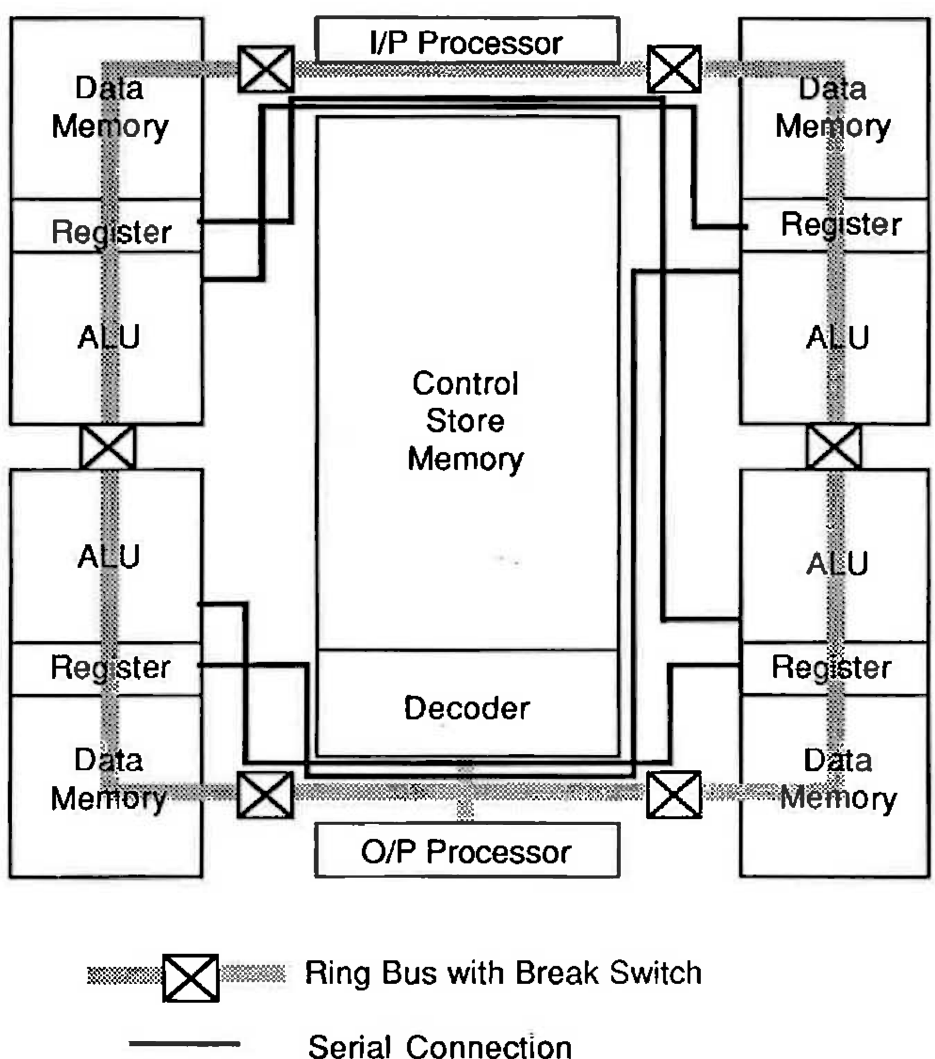


Figure 3.5: Floorplan for the Transform and Filter Brick

2. Each ALU is actually made up of identical replicated cells. A spare cell could be provided to each ALU for redundancy.

3. Each data memory consists of an array of 32×16 static memory cells. Redundancy could be achieved through the provision of spare rows and/or columns.

4. The control store is also a static RAM consisting of 128 words, each containing 63 bits. Spare rows and columns could also be used here to increase yield.

Before calculating the yield improvement it is also important to consider the possible detrimental side effects of providing redundancy:

1. Additional test procedures may be required to isolate the fault, rather than just present a go/nogo answer.

2. Some means must be provided to switch in the spares as required. The traditional DRAM techniques involving the use of laser or other fuses were unavailable to the designers so this was achieved through the use of latches and CMOS transmission gates. These circuits will introduce additional delays into the data paths.

3. Additional control circuitry is required to program in the redundancy.

4. Additional design time is required to implement all of these.

5. For redundancy schemes such as 1 above the resultant floorplan efficiency must also be considered. For example, the provision of only one spare ALU is clearly not a viable option as it results in an asymmetrical floorplan.

A figure of merit FM, accounting for both the improved yield and increased silicon area, was calculated for all of the above redundancy approaches [Mangir, 1984]:

$$FM = \frac{Y_{redundant}}{A_{redundant}} \frac{A_{non-redundant}}{Y_{non-redundant}} \tag{3.56}$$

Evaluation of the first redundancy option, provision of spare ALUs, showed no advantage (*ie.* $FM < 1$) due to the large amount of extra silicon area required to handle perhaps only one fault.

Each ALU consisted of 6 identical cells, each of approximately 0.5 mm^2 in area. The provision of a single spare cell resulted in a total FM to the whole chip of less than 1.1 and would have introduced considerable design complexity, and thus was not used.

The data memory, with a total area of about 1.3 mm^2 is a 32 row by 16 bit SRAM. Provision of a spare row gave resulted in a total FM for the chip of 1.1. Although implementation of this approach was straightforward, the method was not proceeded with as the observed benefits were marginal.

3.7.1 Fault Tolerant Control Store

Consideration was given to providing spare rows and columns for the control store. If an attempt is made to address a bad row a spare row decoder would disable normal addressing and cause the spare row to be accessed instead. Column redundancy would be implemented by a column switching network below the base of the RAM. From this description it is evident that the delay and area penalties associated with the provision of sparing would grow faster for the provision of extra spare columns than for rows.

The yield formula that best describes this situation is given by [Stapper et al., 1980]

$$Y = \sum_{\text{fixable combinations}} P(ijk), \tag{3.57}$$

where $P(ijk)$ is the probability of having i failed single cells, j failed complete columns and k failed complete rows:

$$P(ijk) = \frac{\Gamma(i+j+k+\alpha)}{\Gamma(\alpha)} \frac{(1/\alpha)^{i+j+k}}{(1+(D_0 A_{total})/\alpha)^{i+j+k+\alpha}} \frac{(D_0 A_{cell})^i (D_0 A_{col})^j (D_0 A_{row})^k}{i!j!k!}. \tag{3.58}$$

A_{total} is the critical area of the whole chip. As the yield simulator would have taken far too long to run on the whole chip this was taken as the total non-empty silicon area. A_{col} is the critical area of the columns including the sense amplifiers and the critical areas in the cells associated with whole column failures, *ie.* the bit lines. A_{row} is the critical area of the rows including address decoders and memory word lines. A_{cell} is the critical area of the memory cells minus bit lines and word lines. The summation variable, "fixable combinations", in Equation 3.57 refers to the combinations of i failed single cells, j failed columns, and k failed rows that can be tolerated given the number of spares present. For example with one spare row and one spare column, the fixable combinations of (i, j, k) are $(2,0,0)$, $(1,1,0)$, $(1,0,1)$, $(0,1,1)$, $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$. Any fault falling outside the rows and columns of the memory results in a complete chip failure.

The critical areas of the word and bit lines in a cell are not simply the mask areas of those lines. Instead they were determined by running the yield simulator on a

single cell under three different conditions. First with no changes, second with only the bit lines removed and, third with only the word lines removed. It was found that about 20% of the critical area was due to the bit lines and the effect of the word lines was negligible.

The yield simulator was only used to determine these percentages. The actual areas were calculated from the raw silicon area consumed, thus avoiding the requirement of using the yield simulator on the larger cells.

The required areas expressed as a percentage of total memory area A_{mem} were determined as:

$$A_{cell} = 0.55A_{mem} \quad (3.59)$$

$$A_{col} = 0.19A_{mem} \quad (3.60)$$

$$A_{row} = 0.25A_{mem} \quad (3.61)$$

where A_{mem} was calculated to be $11.8mm^2$ and the total chip area A_{tot} was calculated as $38.34mm^2$.

Figures of merit for various redundancy schemes were determined for a variety of conditions: α, the clustering parameter, was varied between 0.5 and 4; D_0, the average defect density, was varied between 0.05 $defects/mm^2$ and 0.5 $defects/mm^2$; and the percentage areas for A_{cell}, A_{col} and A_{row} were varied by up to a factor of 2. A sample set of results can be found in Figure 3.6.

The final choice on level of redundancy was largely independent of the variations in parameters described above (though the actual figures of merit were quite dependent on α and D_0.) Generally that the provision of one spare row and one spare column substantially improved the yield, with any further sparing providing small incremental improvements only. As the complexity of introducing more than one spare column was comparatively quite high it was decided to introduce only one spare column and four spare rows, an area overhead for the memory of 5%.

3.8 Granularity and Defect Tolerant Chips

The introduction of redundancy into this non-regular chip did not have a high return, at least in terms of an improved yield. To get the most out of any redundancy, ideally, one should have a small replicated cell which constitutes the bulk of the device. In the case of TFB, 8064 cells covered a third of the chip area and yet the best yield improvement factor was only 1.5. For structures like the ALU where there were only an order of 10 cells, covering about one tenth of the chip's critical area, a Figure of

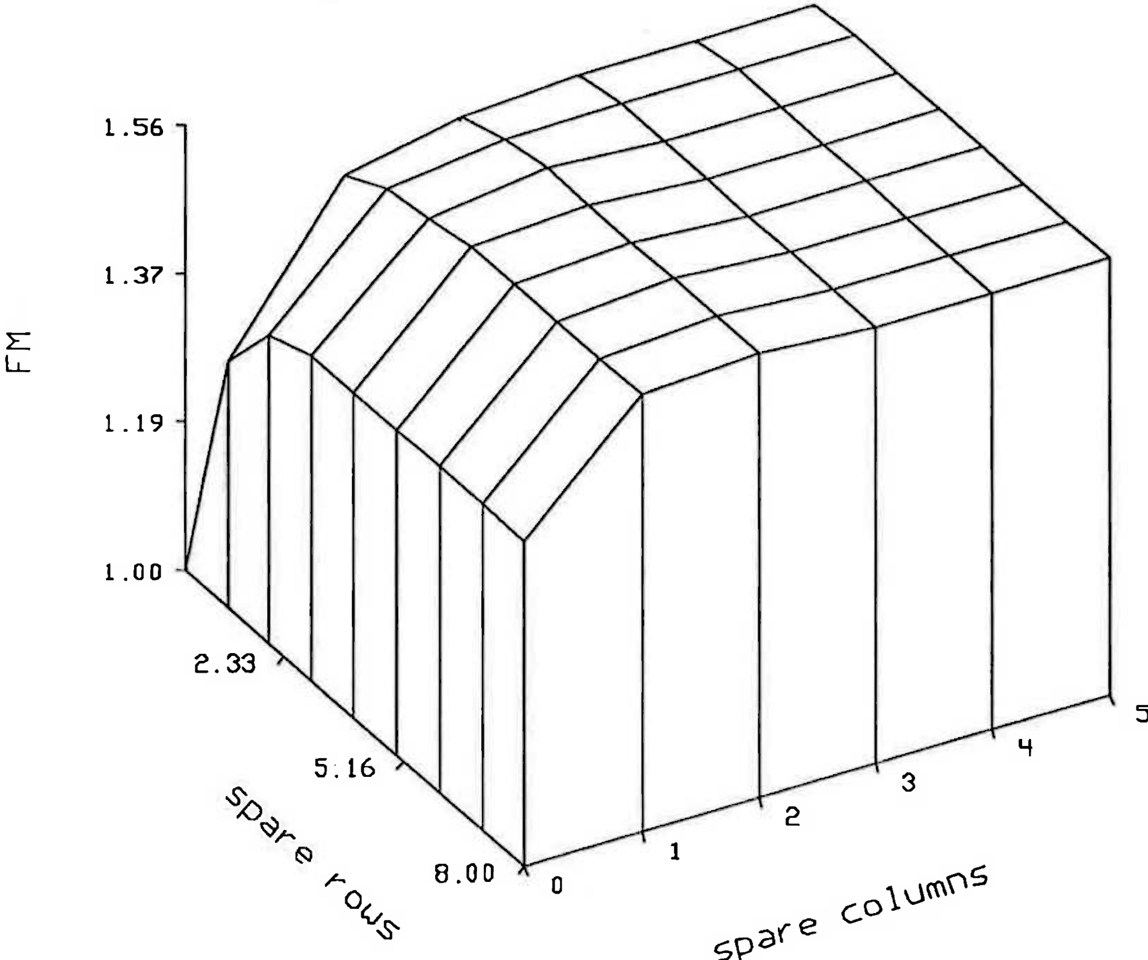


Figure 3.6: Figures of Merit (FM) vs. the number of spare columns and rows in the control store, for the whole chip with parameters: $D_0 = 0.10$ defects/mm^2. $\alpha = 2$. $A_{cell} = 0.55A_{mem}$, $A_{col} = 0.19A_{mem}$, $A_{row} = 0.25A_{mem}$.

Merit of only about 1.1 could be achieved. Given the considerable design effort and delay penalty that would be required to achieve the yield improvement it was decided not to provide any spares to such structures. Duplicating whole structures, such as ALUs, was clearly detrimental to improving yields.

To verify the above discussion the effects of defect tolerance on a more general chip was studied. The general chip had an area of 100 mm^2 and contained one array which consumed somewhere between 10% and 100% of the chip and contained repeated identical elements of size somewhere between 0.1 mm^2 and 1 mm^2. The optimal Figure of Merit FM,

$$FM = \frac{Y_{redundant}}{A_{redundant}} \frac{A_{non-redundant}}{Y_{non-redundant}}, \tag{3.62}$$

was determined for a number of cases within the bounds just specified. These results are plotted in Figures 3.7 to 3.12 for three different defect densities. Equation 3.19 was used to determine these results with the assumption that all of the provided spares could be used if required *ie.* $C_{kjN} = 1$.

From these plots it is evident that the main factor affecting the Figure of Merit (FM) is the percentage of chip area that is devoted to an array. In comparison the FM was extremely insensitive to the grain (element) size of the array. Given the combined performance and "extra effort" penalty of say 10% for introducing defect tolerance then a FM of 1.1 or better is required before fault tolerance can be considered. This requires that 20% or more of the chip be devoted to an array structure of some form in order for defect tolerance to be considered useful. The results are consistent across the full range of defect densities considered. As varying defect density is equivalent to varying area in the opposite fashion (it is the product of area and density AD_0 that is used in the yield equations) then this result also applies to a corresponding range of chip areas also.

The total chip area, including spares, is plotted for the $D_0 = 0.05$ defects/mm^2 case in Figure 3.13. Except for the 100% array cases with large grained elements the area overhead for spares required to achieve an optimal FM was generally small. Similar results were obtained for the other cases of D_0.

3.9 Granularity in Wafer Scale Arrays

The effect of granularity on redundancy decisions is just as important in wafer scale arrays as it is in chips. Generally the higher the granularity the higher the fault

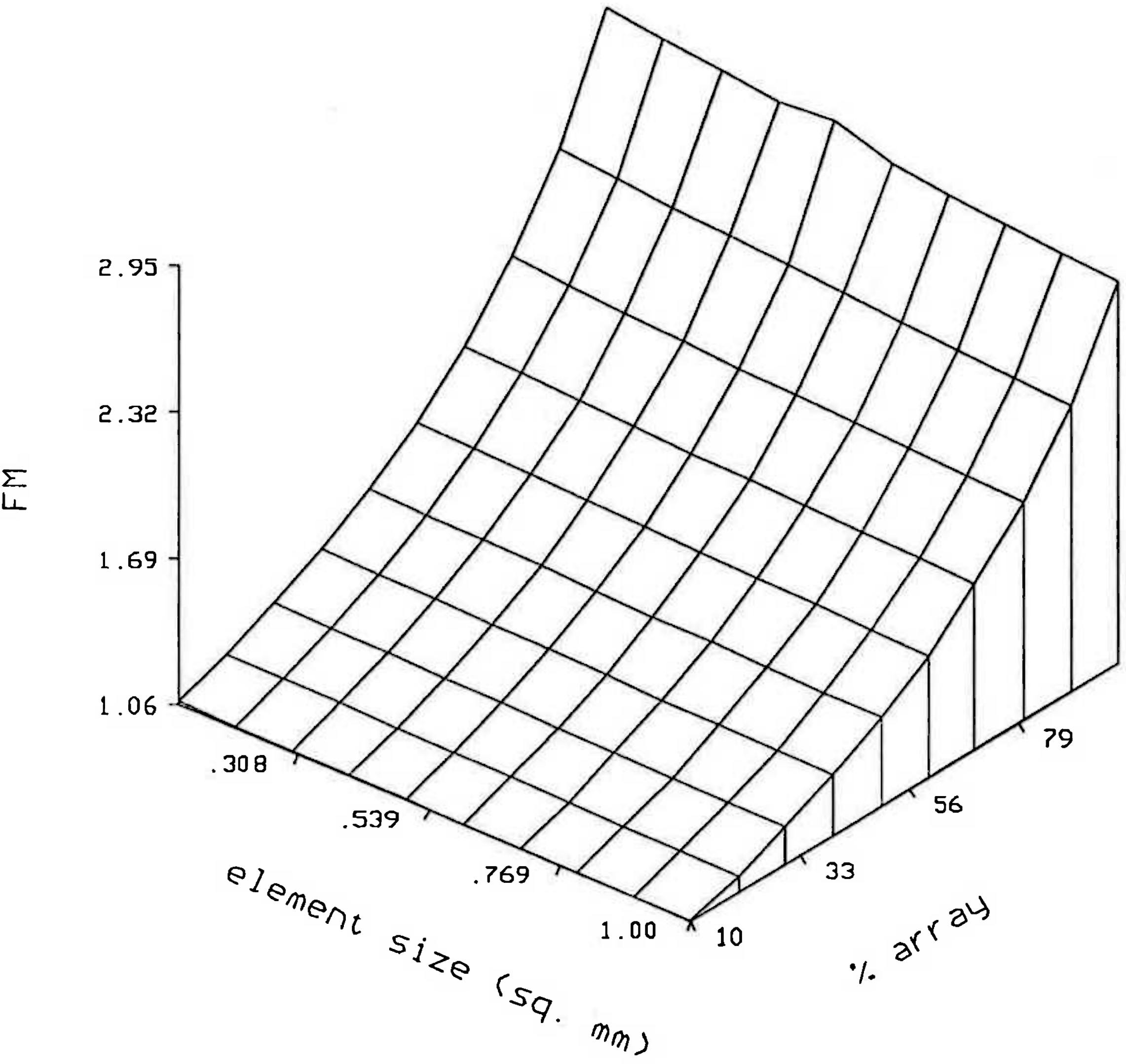


Figure 3.7: Figure of Merit FM vs. replicated element size, and the % of the chip area covered by the array containing this element. ($D_0 = 0.02$ defects/mm^2).

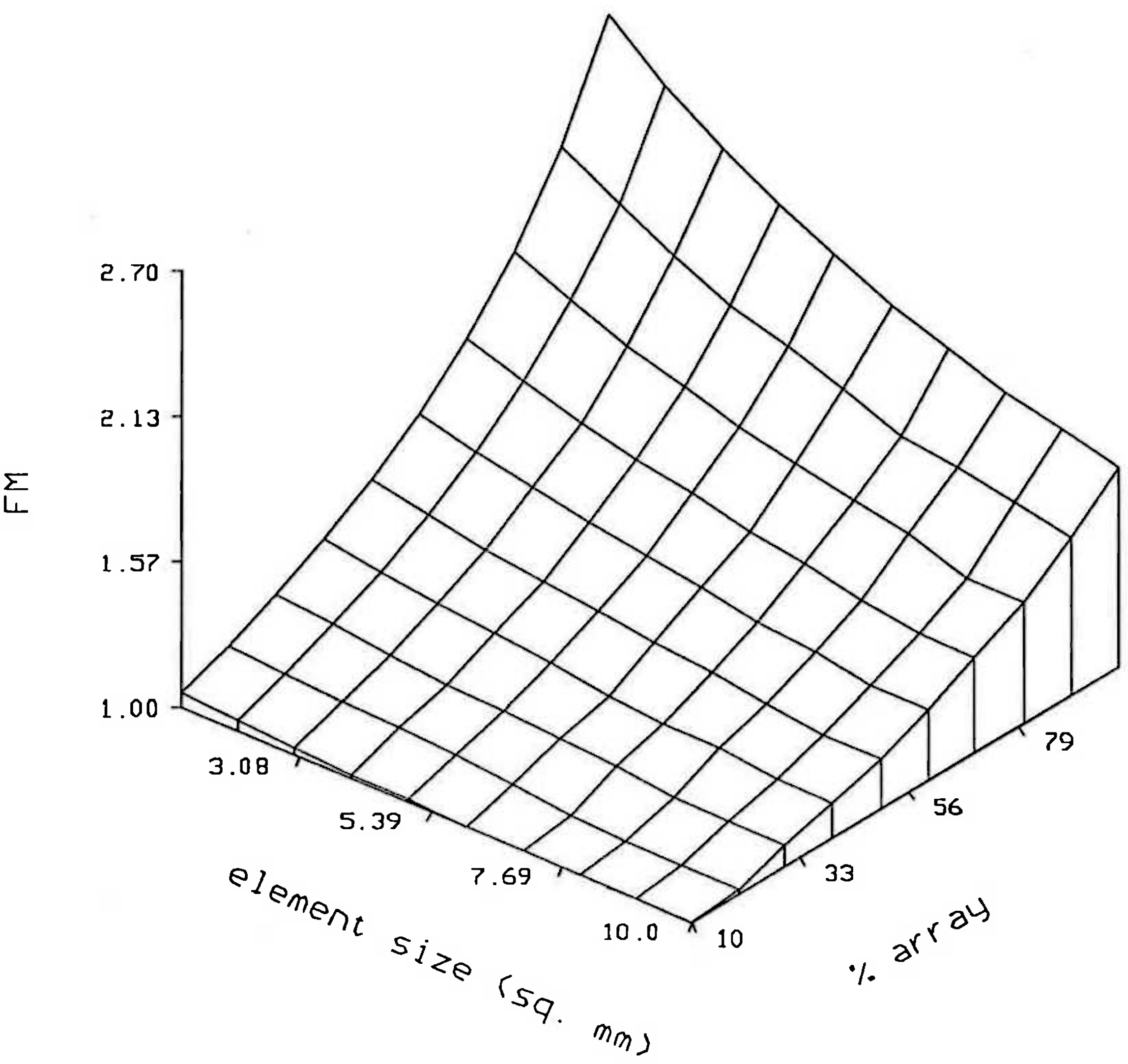


Figure 3.8: Figure of Merit FM vs. replicated element size, and the % of the chip area covered by the array containing this element. ($D_0 = 0.02$ defects/mm^2 and larger element size range).

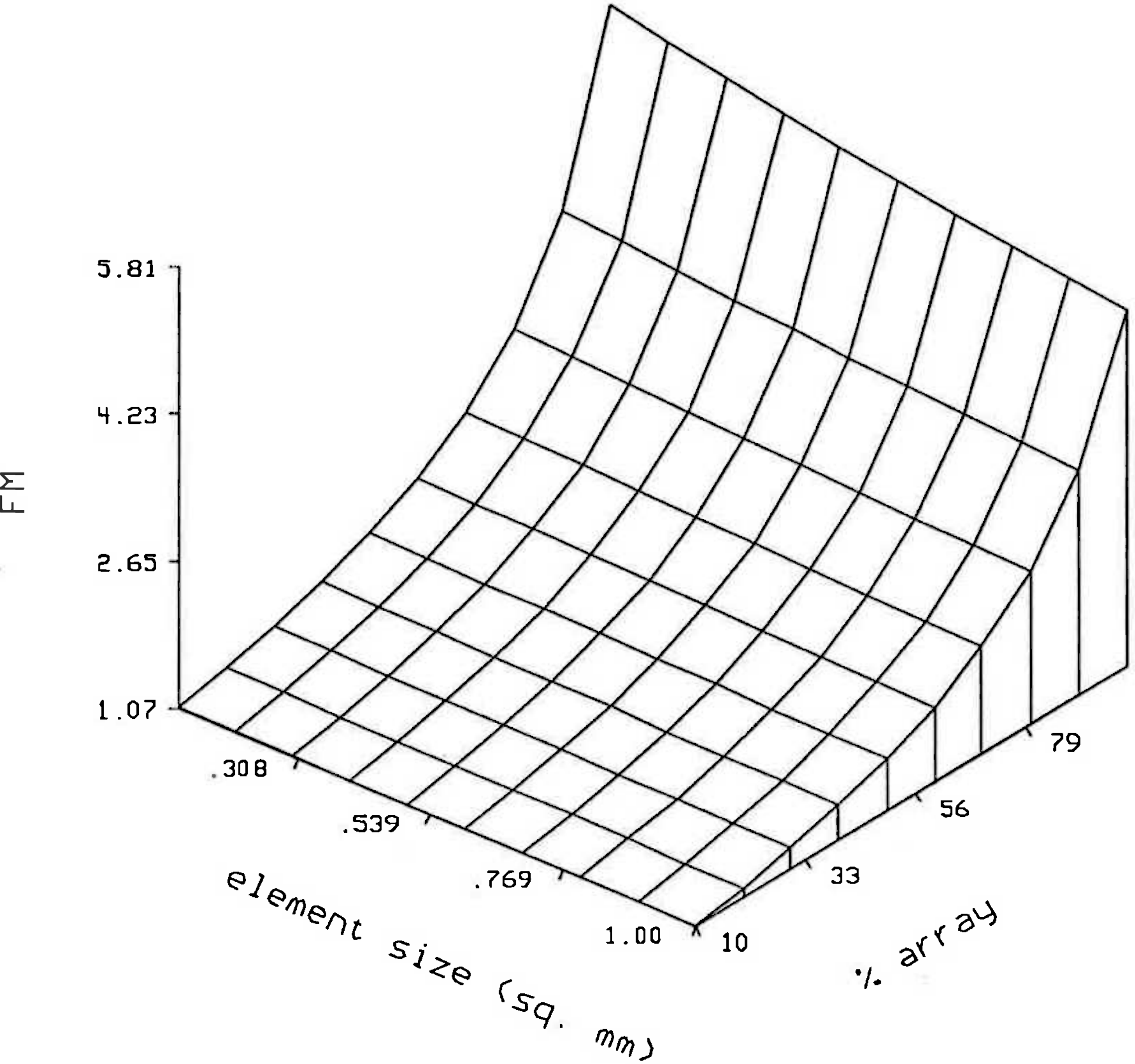


Figure 3.9: Figure of Merit FM vs. replicated element size, and the % of the chip area covered by the array containing this element. ($D_0 = 0.05$ defects/mm^2).

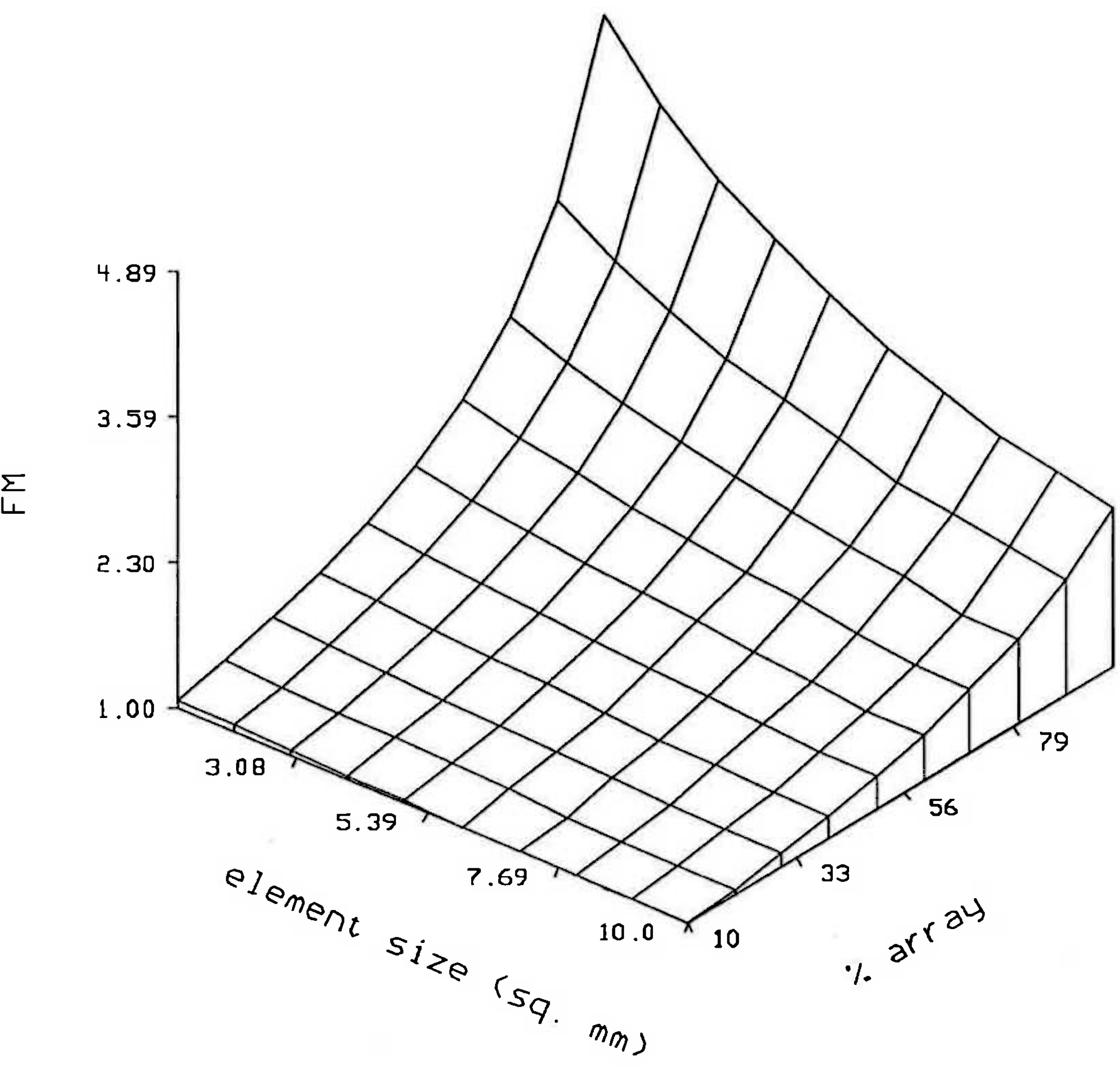


Figure 3.10: Figure of Merit FM vs. replicated element size, and the % of the chip area covered by the array containing this element. ($D_0 = 0.05$ defects/mm^2 and larger element size range).

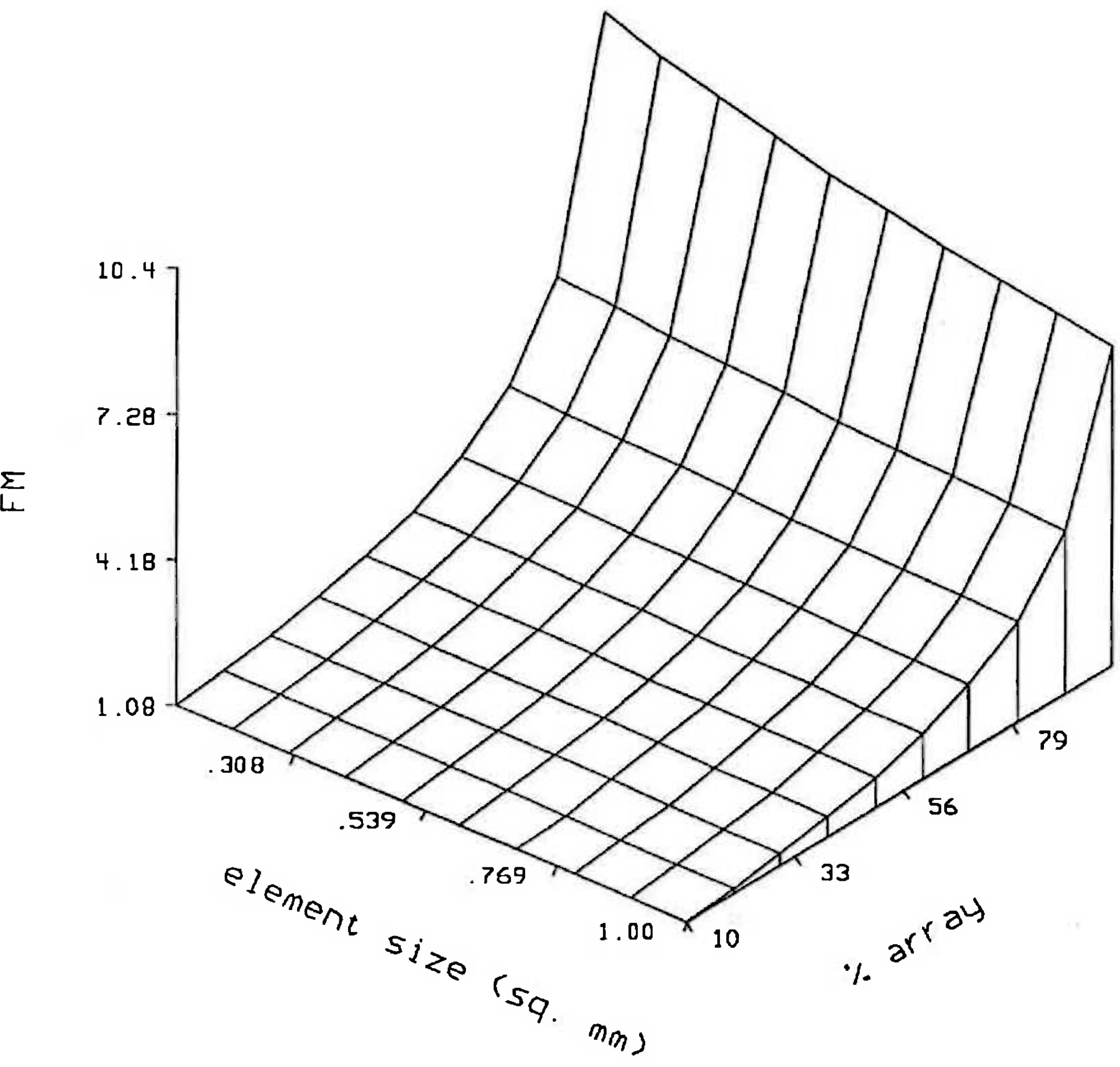


Figure 3.11: Figure of Merit FM vs. replicated element size, and the % of the chip area covered by the array containing this element. ($D_0 = 0.10$ defects/mm^2).

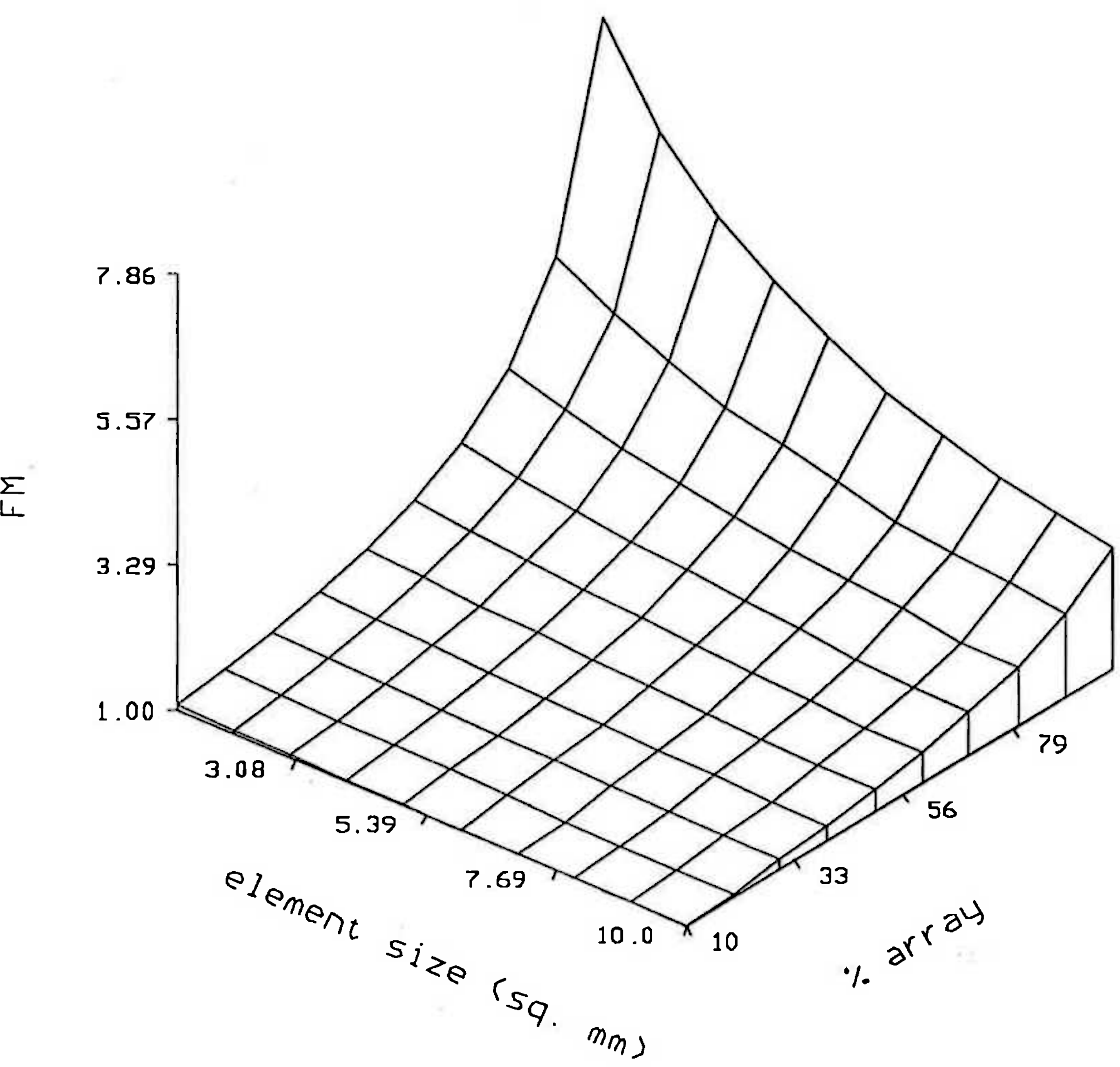


Figure 3.12: Figure of Merit FM vs. replicated element size, and the % of the chip area covered by the array containing this element. ($D_0 = 0.10$ defects/mm^2 and larger element size range).

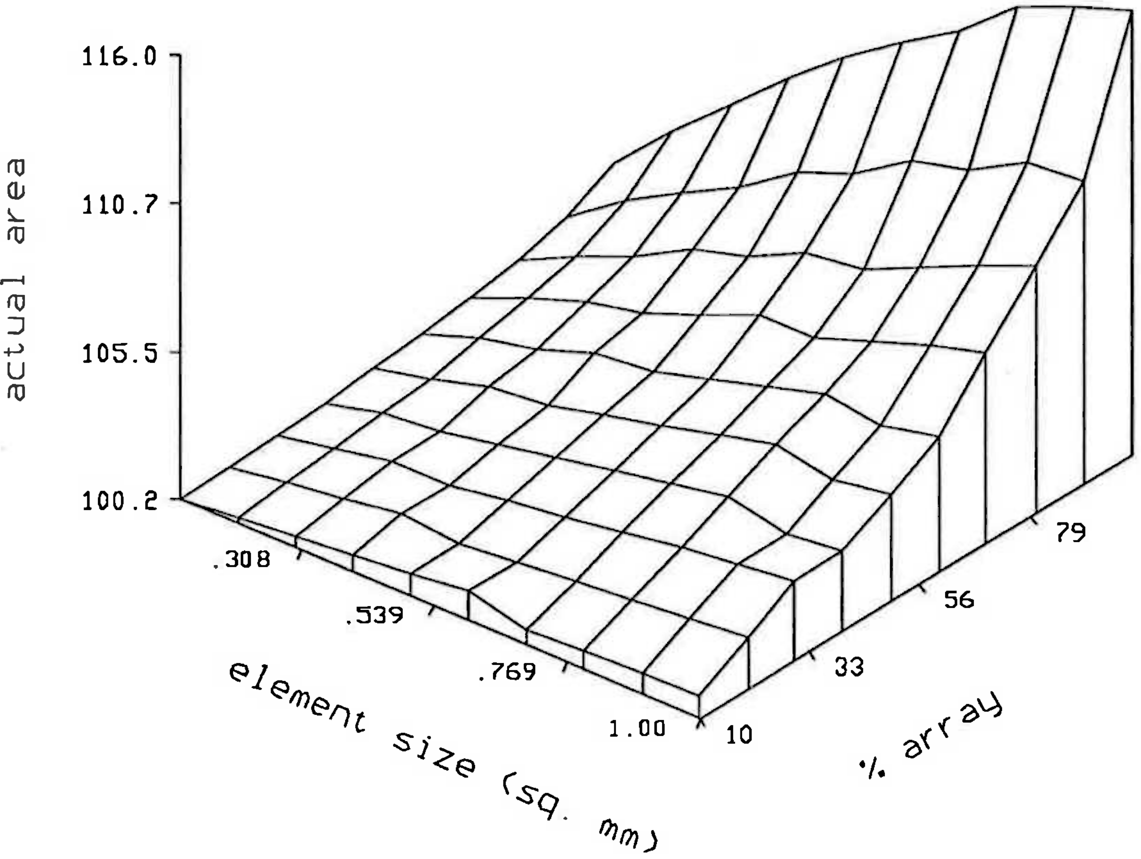


Figure 3.13: Actual area required to achieve optimal Figures of Merit for $D_0 = 0.05$ defects/mm^2 case.

tolerant yield that can be expected. By using the smallest PE size possible both the raw PE yield is maximized and, as any fault takes out only the area of a single cell, silicon utilization is maximized. (This result is tempered somewhat by the effect of the overhead required for reconfiguration. This overhead may dictate that cells should be grouped into super cells for the purpose of reconfiguration [Moore & Day, 1984].)

Bit-serial architectures tend naturally to result in small PEs and in fact have a lot of advantages for VLSI implementation [Denyer & Renshaw, 1985]. Most bit-serial architectures however are hard-wired for a particular class of operations. To achieve programmability the PE will usually have to be larger. For example even though the PE in Lyon's MSSP is nominally bit-serial it still occupies a 5 mm square chip [Lyon, 1984]. In order to achieve the required multi-functioned programmable flexibility the PE was organized as three bit-serial functional units grouped around a 64 word memory.

Concentrated programmable control generally requires less silicon than distributed control. This is because many programmable functions, such as flow control, affect all functional units, but are only implemented in one control unit in a centralized approach. Thus for programmable SIMD machines, and even more so for MIMD machines, the natural tendency would be to have large PEs so that control can be centralized. Matching the size of the control unit, as far as the floor plan is concerned, may often result in using bit-parallel rather than bit-serial arithmetic units. An example is the TFB floorplan. Going to four bit serial ALUs would result in the control store consuming over 90% of the silicon area. If more than four bit-serial units were used then design of efficient control distribution and data flow would be much more difficult.

Speed requirements may also dictate the use bit-parallel processors. Bit-parallel arithmetic units have lower latencies, an often desirable property.

Thus there is a minor conflict here. A move to WSI requires small PEs for yield purposes, whilst architectural requirements may call for large PEs. The conflict is readily resolved however by realizing that the architecture that may result in the most efficient utilization of silicon as far as a single PE in an array processor is concerned may not be the architecture that results in the most efficient silicon utilization of a whole wafer device.

One concept that provides some measure of this tradeoff is that of area utilization, defined as that fraction of the silicon area that is used in the final reconfigured array [Koren & Breuer, 1984]. This measure will be used when reconfiguration schemes are compared in Chapter 6. However for the discussion here it is not quite the measure

wanted. It measures total area, not total functionality. As an illustration of the yield effects of granularity on total functionality consider the concept of *partitioning.*

Imagine the (unrealistic) situation where TFB was being considered for WSI. As diagonal microcode is used in TFB, with separate microcode fields for each ALU and Data Memory, it would be possible to consider replacing each TFB unit with four sub-units, or quadrants. Certain control functions would have to be duplicated in each sub-unit and more wiring would to be required thus introducing an area overhead,

$$\text{Area Overhead} = \frac{W \times A_{sub-PE} - A_{PE}}{A_{PE}} \tag{3.63}$$

where W is the number of sub-PEs formed by partitioning, A_{sub-PE} is the area of a sub-PE and A_{PE} is the original PE area. In other situations where there is no replication within the PE it may be possible to split the PE into two or more different element types, individually reconfigurable, or consider another smaller PE architecture that may not be as efficient at the required task as the original PE. In this latter case the area overhead would be replaced by a performance overhead per surviving PE.

Two different situations arise:

1. Expanding Area. The total array area may be adjusted so as to maximize the FM for each individual case. This would be applicable to arrays that are built as large chips rather than complete wafers.

2. Constant Area. This would apply to complete wafer scale arrays.

Figures of Merit for both of these approaches are given in Figures 3.14 to 3.17 for the following base cases (indicated with respective optimal redundancies):

1. An array of 45 PEs of area 50 mm^2, 25 of which are redundant.

2. An array of 33 PEs of area 10 mm^2, 13 of which are redundant.

The parameters for the yield equation were: $D_0 = 0.05$ defects/mm^2 and $\alpha = 1$.

3.10 Modeling Interconnect Yield in Reconfigurable Circuits

A fault in the inter-PE wiring or in the control circuits that govern reconfiguration can have an effect ranging from none at all to complete array failure depending on

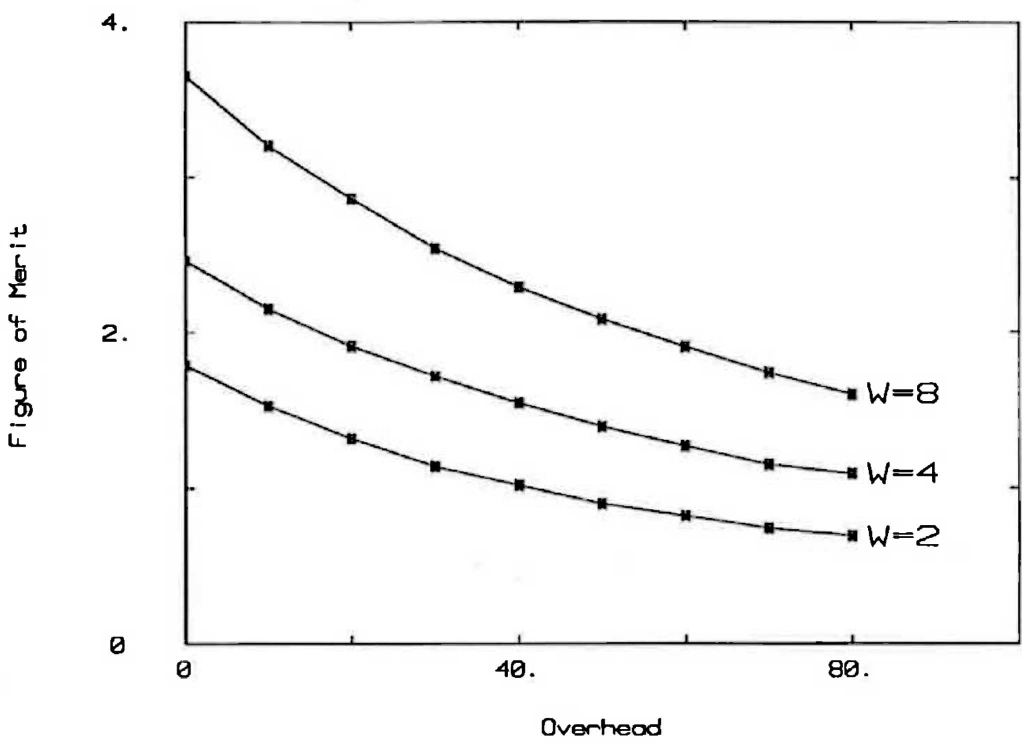


Figure 3.14: Effect of partitioning on the FM: Expanding area, $A_{PE} = 50$ mm^2

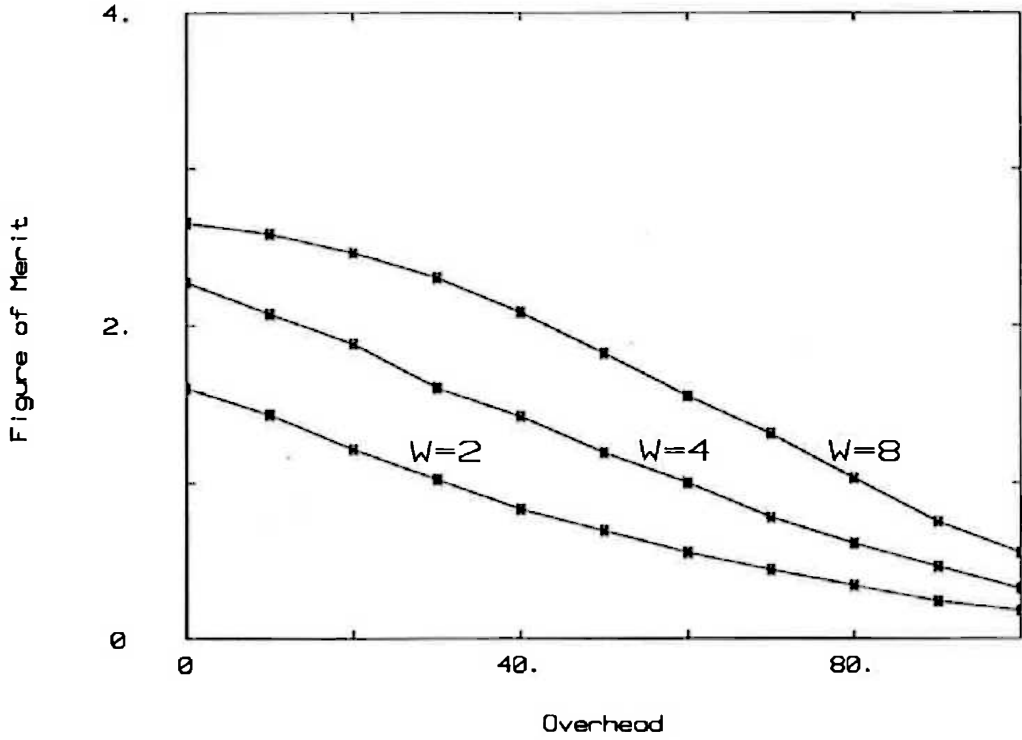


Figure 3.15: Effect of partitioning on the FM: Constant area, $A_{PE} = 50$ mm^2

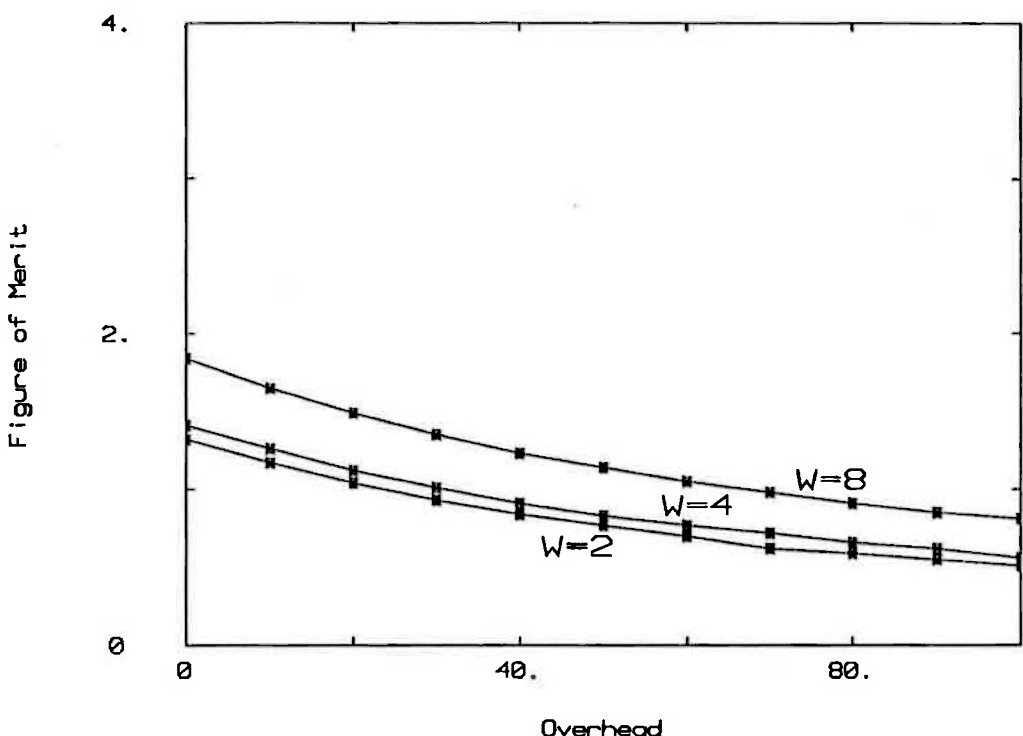


Figure 3.16: Effect of partitioning on the FM: Expanding area, $A_{PE} = 10$ mm^2

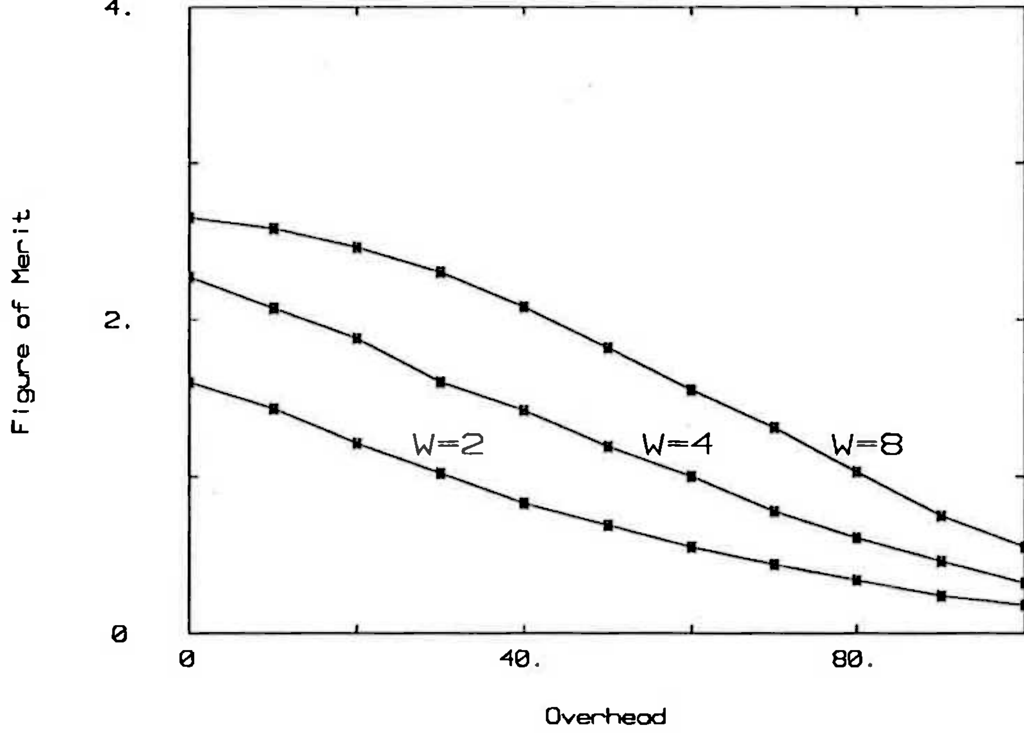


Figure 3.17: Effect of partitioning on the FM: Constant area, $A_{PE} = 10$ mm^2

its location. Some previous studies have assumed that any wiring fault will lead to complete array failure [Mangir, 1984]. This overstates the impact of the area devoted to reconfigurable wiring on yield. More commonly it is assumed that the wiring contains no faults at all. In this section the effect of wiring faults will be introduced into the above model by assuming that wiring faults only fall into one of the four following cases [Franzon, 1986b]:

1. The wiring fault can be treated as a fault in one associated PE;
2. The wiring fault leads to complete array failure ("array kill");
3. The wiring fault produces no effect at all;
4. The wiring fault can be treated as a fault affecting a predetermined number of PEs.

The first effect can be modeled by adding the wiring area to the PE area; the second by adding area to the array area; and the third by adding no area at all, as far as the yield model is concerned. The fourth effect has to be handled by adding parameters to the model. The justifications for choosing these categories are as follows:

1. Many faults, particularly for near neighbour only connections, result in limiting access to one PE only. If the fault is in the connecting wire then access is limited from one direction only. If the fault is in the reconfiguration controller then all access to the PE is most likely denied. Some such controller faults may limit access to several PEs. Additionally a fault resulting in the removal of one PE may result in the removal of several PEs from the array. This would happen, for example, when no further replacement PEs are available for that row and a whole column has to be removed. This last effect is handled automatically by using the Utilization term U or C_{ij} in the yield equation, and thus does not need to be included here.
2. Many wires, such as power and clock lines, are distributed over the whole array and a failure in these will usually result in an array kill. In addition, in many reconfiguration schemes a fault in some of the interconnect may result in an array kill. Generally it is desirable to minimize the array kill area. Another possible contribution to the array kill area is the extra hardware added to control the reconfiguration step. This is particularly the case for self-configuring

arrays. (Evans and McWhirter [Evans & McWhirter, 1986] have suggested a means of reducing this effect for Evan's reconfiguration scheme.)

3. Except for the simplest reconfiguration schemes, most of the inter-PE wires provided are not used because they are included to enable redundancy and thus many wiring faults result in no effect.

4. The final case may arise from situations where a row or column is affected by a single failure. For example, in many arrays an interdigited comb pattern is used for distributing power and clocks. A failure in one of the comb fingers will mean that on average half of the associated row or column is lost. Note that this is not necessarily the case in all arrays. In [Franzon & Tewksbury, 1987] the power and clock distribution can withstand one failure in each column. Some interconnect failures can also result in the complete loss of a row or column.

The fourth case cannot be adequately handled by adjusting A_{mod} but is best covered by introducing A_{row} and/or A_{col} as was done for the memory case. For example, in the case where row failures are possible the yield expression becomes:

$$Y = \sum_{k=0}^{\infty}\sum_{l=0}^{\infty} \frac{\Gamma(k+l+\alpha)}{\Gamma(\alpha)} \frac{(1/\alpha)^{k+l}}{(1+D_0A/\alpha)^{k+l+\alpha}} \frac{(D_0A_{mod})^k (D_0A_{row})^l}{k!l!} P_{klNR} \quad (3.64)$$

where

$$P_{klNR} = \sum_{j=1}^{min(k,R)} \sum_{m=1}^{l} Q_{kjN} Q_{lmN_R} C_{jmNR} \quad (3.65)$$

and

A	=	total critical area
A_{mod}	=	critical area of modules only
A_{row}	=	critical area that causes complete row failure
D_0	=	average defect density
k	=	number of defects in modules A_{mod}
l	=	number of defects in A_{row}
P_{klNR}	=	Probability of successful repair with (k,l) failures
j	=	number of module failures
m	=	number of complete row failures
Q_{kjN}	=	Probability of k defects being distributed in j faulty modules

$Q_{lmN_{row}}$ = Probability of l defects being distributed in m faulty rows

C_{jmNR} = Probability of fixing the array with j faulty modules and m faulty rows

To calculate the expected number of PEs the easiest method is to subtract the expected number lost due to row/column failures from the expression calculated in Section 3.5.2. The expected loss for the situation where there is a row failure mechanism would be

$$L(P) = \sum_{l=1}^{\infty} Pr[l \text{ row defects}] \sum_{m=1}^{N_{row}} Q_{l,m,N_{row}} m V_r(m) \tag{3.66}$$

where l is the number of row faults, $Pr[lrowdefects]$ is the probability of l row defects, m is the number of faulty rows and $V_r(m)$ is that percentage of total PEs lost due to the faulty rows. N_{row} is the number of rows. The expectation number obtained by subtracting $L(P)$ from $E(P)$ calculated in Section 3.5.2 will be a slight underestimate due to a small amount of duplicity between PEs lost due to failed rows and PEs lost due to failed individual PEs.

For a power failure in an interdigitated, non-fault tolerant power distribution scheme, if the loss of part of a row can be effectively contained to that row, $V_r(m) = N_{col}/2$, where N_{col} is the number of columns and thus the number of PEs in a row. *ie.* half of the PEs in a row will be lost on average. For the scheme that will be analyzed below power and clocks would not be fed along a row because loss of power in part of one row means that approximately the same portion of all the other rows will be unusable. Partial loss of column would most likely mean the utilization loss of most of a column and thus $V_r(m)$ would be larger than $N_{col}/2$ in this case, the exact figure being determinable through Monte Carlo simulation.

As an example of how to determine figures for categories 1–3 above consider the mesh array reconfiguration scheme given in Figure 3.18. In this scheme a mesh is mapped onto a faulty array by bypassing faulty PEs within each row and by steering the columns around these faulty PEs. The contributions of wiring area to the different areas required for yield calculations are determined as follows:

The array kill area is

$$A_{kill} = A_2 + A_4(1 - UY) \tag{3.67}$$

where the wiring areas A_i are determined by reference to Figure 3.19, Y is the yield of the PEs, and U is the utilization of the PEs. The contribution of A_4 is determined by

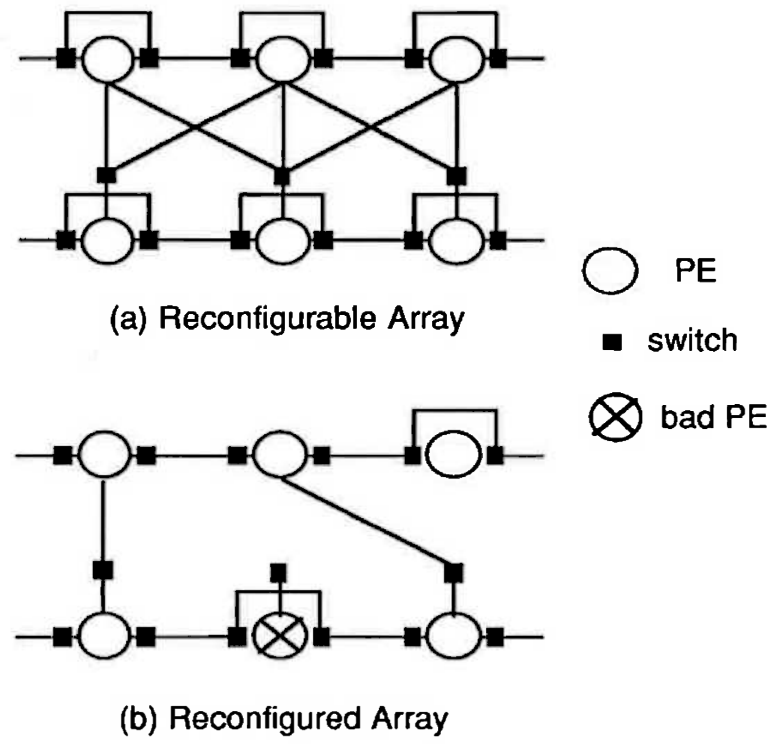
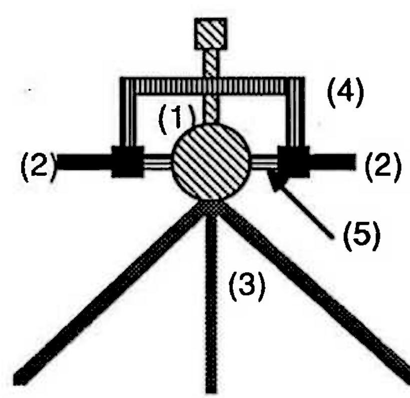


Figure 3.18: Mesh array reconfiguration scheme showing (a) all the wiring, and (b) an example with faults.



contributes to PE area (1)

contributes to fatal area (2)

part of this area contributes to PE area
remainder has no effect (3)

part of this area contributes to fatal area
remainder has no effect (4)

part of this area contributes to fatal area
remainder has no effect (5)

Figure 3.19: Contribution of wiring areas to yield areas.

consideration of what percentage of these areas are actually required in a reconfigured array. The areas A_i of course refers to wiring area over the whole array, not just the wiring area around each PE. The PE area is

$$A_{PE} = A_1 + A_5UY + \frac{1}{3}A_3UY. \tag{3.68}$$

In this case only $\frac{1}{3}UY$ of the wiring area A_3 is used (note UYN PEs are used, where N is the number of PEs in the array.)

As you can see above A_{kill} and A_{PE} are adjusted simply by adding that wiring area actually required.

Using these terms for area in a yield calculation results in a self referential equation (the individual PE yield Y is required to determine the utilization U and thus the array yield or $E(P)$.) This is not a major problem however as the wiring areas involved are relatively small and approximate values of Y and U can be determined fairly readily. If more accuracy is required then the calculation can be iterated.

Examples of the effect of wiring yield on considering different approaches to WSI will be discussed in a later chapter.

3.11 Place of Yield Simulation in Defect Tolerant Design

Spares were provided for the control memory of TFB in the form of extra rows and columns. The mechanism used to switch in these spares was essentially to bypass those rows and columns containing faults. This is quite appropriate for a memory as complete row or column failure is quite likely. For other structures using larger cells, such as computational units, complete row and column failures are less likely and thus other reconfiguration mechanisms are more suitable [Franzon, 1986a] [Moore & Day, 1984]. Also a slightly different yield equation, reflecting the different repair mechanism, is more suitable, as discussed in the previous section.

In the control memory each cell contributed to all three of the defect tolerant areas, A_{cell}, A_{row} and A_{col}, required by the yield model. It was for this reason that the use of a yield simulator was considered. As it turns out the memory used in this example is not of sufficient density to require this level of detail of calculation for the determination of optimal redundancy. From the literature [Stapper et al., 1980] it appears that it isn't until RAM sizes of 64k bits are reached that detailed critical area calculations are required.

Yield simulation is also most likely not required for redundancy determination for other types of reconfigurable arrays (examples of reconfigurable linear arrays are presented in [Moore & Day, 1984], whilst examples of reconfigurable mesh connected arrays are presented in [Franzon, 1986a]. With usually only one type of cell present, determining the exact critical area of this cell for defect tolerant applications is most likely unnecessary.

Communications and the replacement of faulty cells by spare cells in this type of array is carried out by some form of steerable interconnect. Multiple connection paths need to be provided to each cell so that replacement can be carried out in a flexible manner.

A yield simulator may sometimes be profitably employed to determine the relative critical areas of the interconnect that cause each of these class of failures so that their effect can be employed in the yield model [Franzon, 1986b]. The area of the interconnect that causes whole chip failures is a very crucial area as overall yield is very sensitive to it. In fact all effort should be applied to ensure this area is kept as small as possible by either making it less vulnerable to defects through improved layout, or using defect tolerance techniques on the interconnect. Koren uses yield simulation techniques to show that the latter is generally preferable [Koren et al., 1988].

We will now go on to look at the various approaches available for making large chip or full wafer arrays of processing elements, before proceeding to analyze these approaches using the models just presented.

Chapter 4

Redundancy Techniques for Wafer Scale Arrays

4.1 Redundancy through Reconfiguration

Due to the large number of defects present, without redundancy any WSI product would have zero yield. To achieve a useful yield a large degree of yield enhancement is required. This is best achieved by covering most of the wafer with an array structure as a large number of spares are required. More irregular architectures can only benefit from Triple Modular Redundancy (TMR) or N Module Redundancy (NMR) approaches, as discussed below, and these, are far from ideal defect tolerant techniques for WSI. There are many approaches to implementing defect tolerant arrays and taking cues from fault tolerant computing, these include:

1. *Triple Modular Redundancy (TMR).*

   With TMR each Processing Element (PE) is replaced by three PEs and a voter (Figure 4.1.) The majority vote on the three PEs' outputs forms the accepted result. No testing step is required. This way one faulty PE can be handled automatically. With NMR a majority vote is carried out on N PEs.

2. *Local Sparing.*

   Each PE is provided with local spares. The resulting structure looks similar to TMR except that the Voter V is replaced by a Chooser C that tests the PEs and chooses which will be connected. Spares are not shared between PEs. Testing must be carried out to ascertain as to which PE will be used. If a failure is detected in this PE then one of its spares is employed in its place.

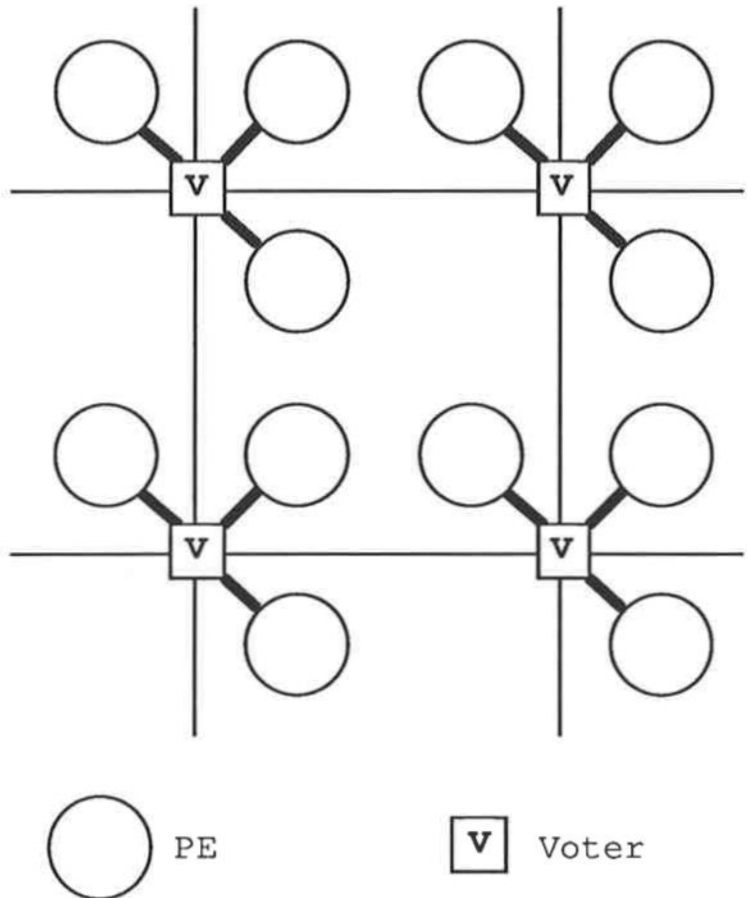


Figure 4.1: TMR approach to implementing a fault tolerant array.

3. *Temporal Redundancy.*

   In temporal redundancy a faulty PE is replaced by spare cycles on a neighbouring PE. Examples can be found in [Kuhn, 1983] [Negrini et al., 1985] and [Sami & Stefanelli, 1984]. Figure 4.2 shows an example where the normal duty cycle of each PE is 50%. This type of redundancy is only useful in cases where neighbouring PEs do have spare cycles, as in many systolic arrays.

4. *Global Sparing and Redundancy through Reconfiguration.*

   The whole array is provided with a set of spare PEs (see Figure 4.3.) These PEs are switched in as required to replace faulty units. Some *Reconfiguration Scheme* must be provided to allow this switching to take place. Formulating this reconfiguration scheme involves satisfying and considering a number of trade-offs. The "original" and "spare" PEs are often not explicitly labeled, every PE qualifying as both.

A basic comparison of the last three approaches above is given as a plot of utilization versus the percentage of PEs that are faulty in Figure 4.4. Since TMR would have a lower utilization than local sparing it is not included. These results were obtained either by carrying out a Monte Carlo simulation of the different redundancy approaches or by taking the relevant author's results. It is immediately evident that the last approach, global sparing and reconfiguration will usually be the only approach worth considering for achieving defect tolerance. In calculating the utilization of the temporal redundancy approach the raw results were multiplied by 50% to account for the loss of having to run the PEs at a 50% duty cycle. However, as mentioned, often this may be inherent in the array design. In these cases temporal redundancy may be as useful as global sparing.

4.2 Reconfiguration Technologies

In order to use globally provided spares a global reconfiguration step must take place. This entails choosing and switching of alternate connection paths. A variety of technologies have been proposed for this process:

- *Discretionary Wiring.* A final custom layer (or layers) of metal are added to the wafer after testing the PEs.

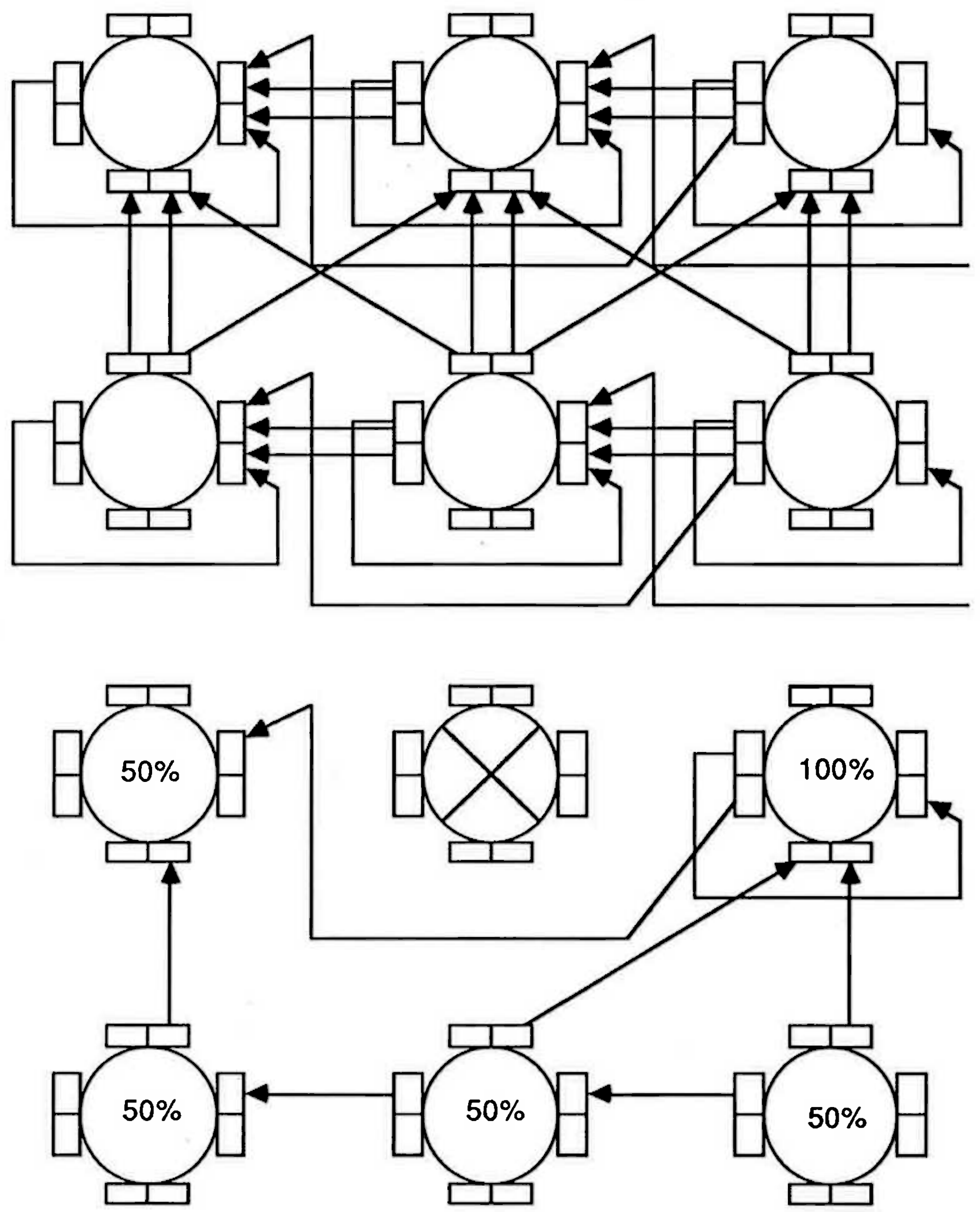


Figure 4.2: A temporal redundancy approach to implementing a fault tolerant array.

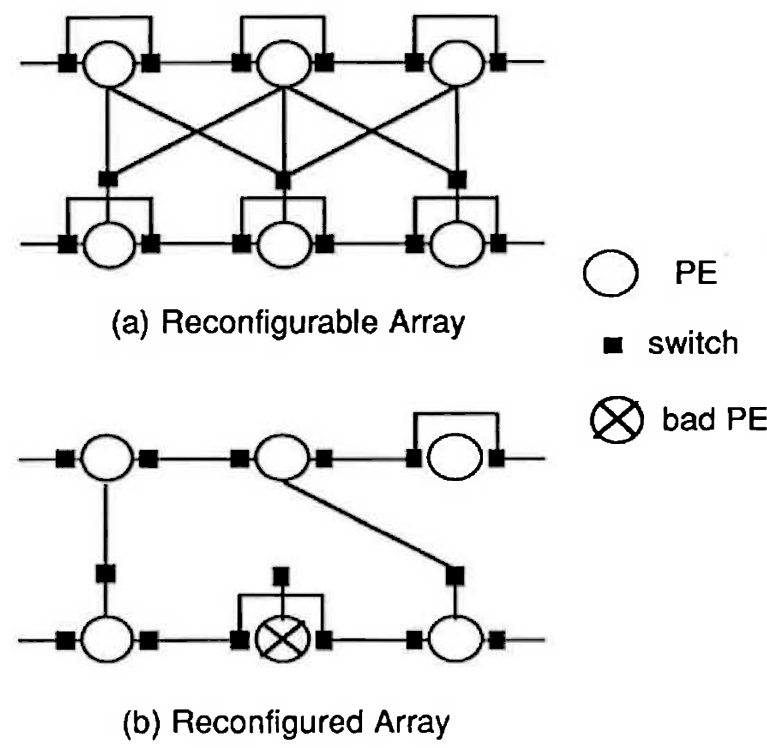


Figure 4.3: Example of a global redundancy approach to implementing a fault tolerant array.

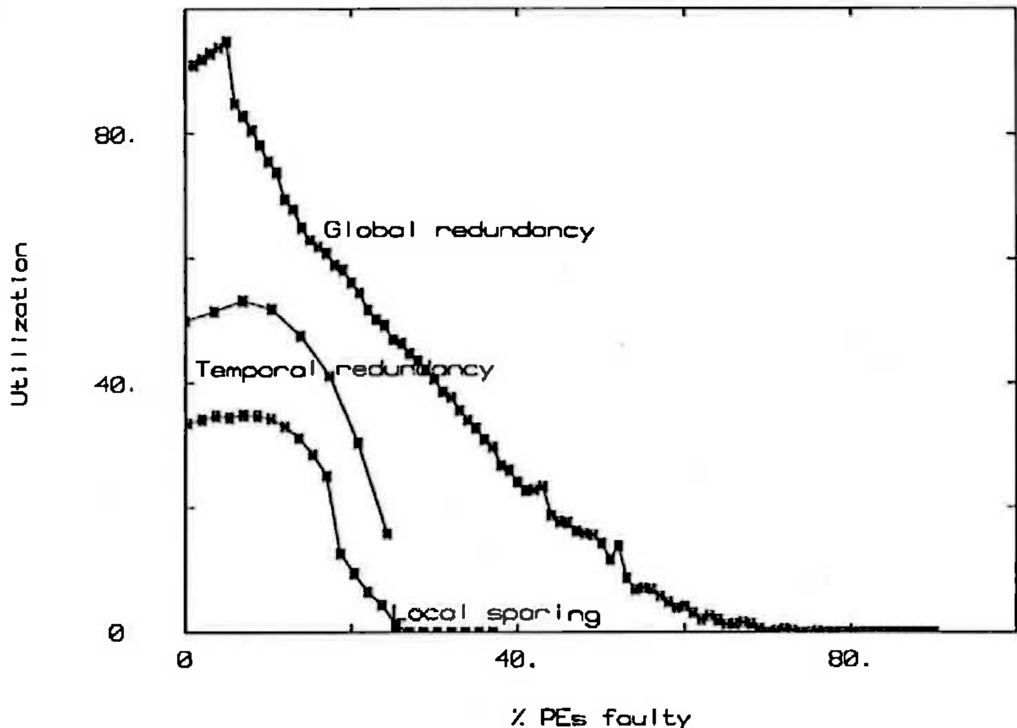


Figure 4.4: Comparison of utilization for different redundancy approaches.

- *'Hard' Fuse and Join Techniques.* Energy is applied to connections on the wafer after testing to either create new connections (join connections) or to break existing ones (fuses.)
- *'Soft' Configurable Switched Interconnect.* Normal circuit switches are used to steer the connections as required.

A form of discretionary wiring was actually the first approach ever considered for WSI [Lathrop et al., 1967]. In this approach a final custom metalization step is carried out for each wafer after probe testing. Unfortunately the expense of providing a custom mask for each wafer outweighed any potential cost benefit of WSI, thus preventing further exploitation of this technique. Part of this cost is in the wafers that must be rejected because of faults in the final custom metalization. An approach that reduced the number of masks required was tried but eventually not pursued [Calhoun, 1969] [Calhoun & MacNamee, 1972].

More recent work at Rensselaer Polytechnic Institute has resulted in another discretionary wiring approach that results in a high yield interconnect process [Donlan et al., 1986] [Donlan et al., 1985b] [Donlan et al., 1985a] [Taylor et al., 1985]. By using an E-beam machine to pattern the resist and a thick

film lift-off process to form microtransmission lines on the wafer they achieve reasonable throughputs (30 wafers per hour) and a high wiring yield. This yield has been further improved by the use of focused ion beam repair techniques [Lin et al., 1986]. By using LC mode microtransmission lines rather than regular metal lines with thinner oxides high inter-PE communication speeds can be achieved (of the order of 2ns for 5cm lines [Donlan et al., 1985a].) These techniques are however aimed more at hybrid wafer scale integration rather than at purely monolithic wafer scale integration.

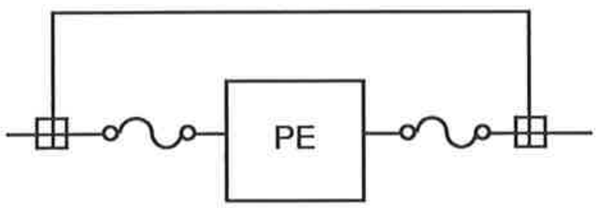
Another discretionary wiring approach that has been developed is that of laying copper tracks on top of the passivation layer to provide low resistance interconnects [Barrett, 1987][Barrett et al., 1986]. Currently wet etching techniques are used to produce 3μm thick, 10μm wide lines or 20μm thick bumps. Ion etching techniques can be used to reduce line widths down to 3μm, though at reduced yields. Laser photodeposition is also being investigated. The main applications are seen in the on-wafer distribution of power and the laying of long signal lines, such as clock lines. A limited form of discretionary wiring for reconfiguration is being considered, in the form creating small distance permanent shorts between aluminium signal lines.

Many fuse techniques have been developed by high density memory manufacturers [Moore, 1986]. Most commonly these techniques involve the electrical or laser activation of polysilicon fuses. Fuse/join techniques can be applied to defect tolerant applications by using one of two approaches (Figure 4.5):

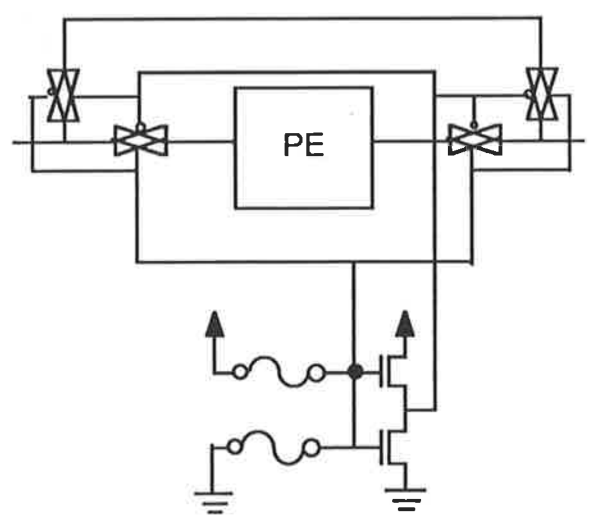
A. The signal paths are programmed directly by forcing joins or opens. (*ie.* 'Hard' configured.)

B. The signal paths are switched by normal MOS switches but these switches are programmed by circuits whose detailed function is determined by fuses and joins. (*ie.* 'Hard' programmed 'soft' configured.)

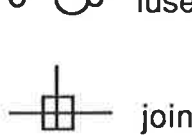
Approach B can be useful in a variety situations. If the unblown fuses or created joins have a higher resistance or lower reliability than normal MOS switches then approach B is useful. Approach B can also be used to reduce the number of programming steps. For example, if multiple paths are switched then with approach B only one programming step can program all these paths, whilst with approach A one step would be required for each path. The number of programming steps can be reduced further if the reconfiguration algorithm allows a small number of bits to be decoded for the programming of a large number of gates. The reconfiguration approach to be described in Chapter 5 serves as an example of this.



Approach A - Use fuses and joins to create connections

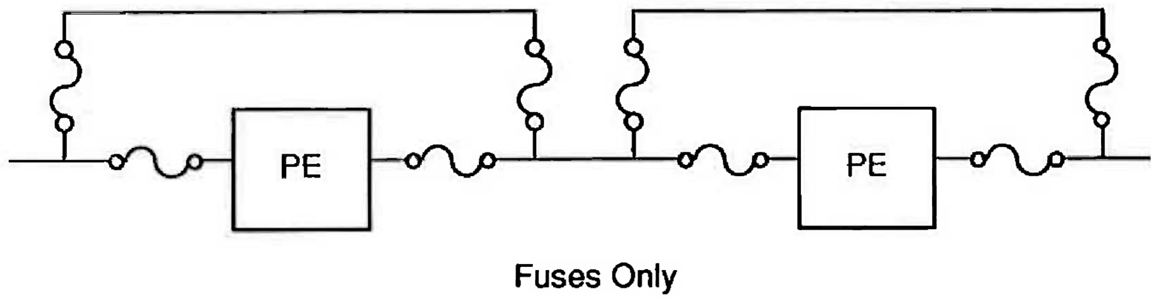


Approach B - Fuses used to program soft switch controller



CMOS transmission gate

Figure 4.5: Two approaches to using fuse/join techniques.



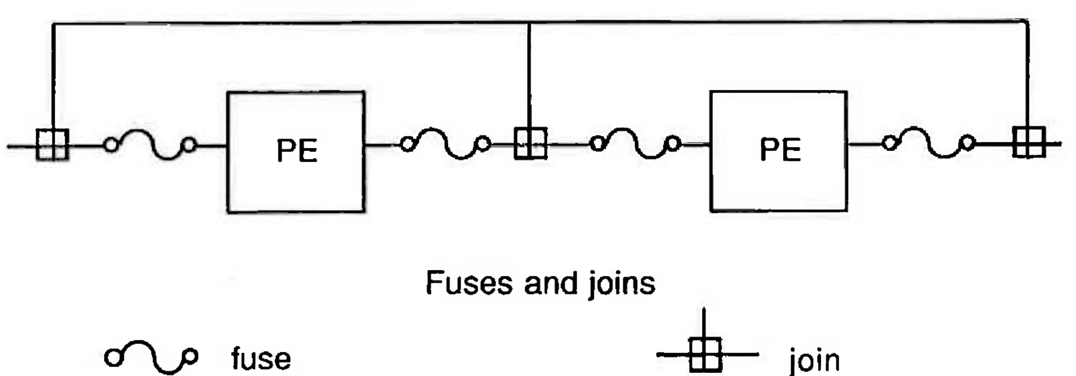


Figure 4.6: Comparison of fuse only and fuse/join techniques for linear arrays.

Laser fusing of metal lines is a straightforward and reliable fusing technique that can be used for either approach. If no join technique is available then approach B should be considered. An example of the relative number of steps that would be required to use approach A for a fuse & join technique compared with approach A for a fuse only technique is given in Figure 4.6. A fuse only technique requires four steps for each and every PE, whilst a fuse & join technique requires four steps only for every PE that has to be bypassed. This difference could save considerable processing time during the programming of a wafer scale array.

Laser fuse and join techniques have been developed in MIT's RVLSI program [Raffel, 1986]. Their fuse technique involves a simple lasing of a metal line. Two laser join techniques have been developed. In the first, a special process is used to create a sandwich consisting of an amorphous silicon layer between two metal layers, that upon laser heating forms a reliable, low impedance (2.5Ω) join [Wyatt et al., 1984] [Chapman, 1986]. Worst case delays times with these lines are about 5ns for 40mm of line. A fuse technique that is compatible with standard CMOS processing has also

been developed [Chapman et al., 1987] [J. Canter et al., 1986]. With this technique two implants, separated by 4μm, experience dopant migration when heated by a laser giving rise to a short in the diffusion layer. This results in a 16μm join with about 75Ω of resistance. Typically current architectures require thousands of processing steps and many laser processing stations would be required to keep up with the throughput of a typical fabrication line. Development is too early to estimate the cost impact of this.

Other join techniques involve the permanent turning-on of a field effect transistor. As this would result in a somewhat uncertain channel resistance these join techniques are best used for approach B. MNOS field oxide techniques [Hsia et al., 1979] or 5V EPROM techniques [Gupta et al., 1982] can be used to create permanently programmed switches. Floating gate FETs can be programmed as shorts by implanting a charge in the channel with an E-beam [Girard et al., 1987].

Both of these "hard-wiring" techniques introduce several new potential problems though:

1. A longer lead time is required in order to obtain and set up the required equipment.

2. Extra processing steps are required, introducing additional cost.

3. Faults discovered during burn-in or in-field use cannot be handled.

These limitations have lead to many researchers choosing a soft-programmed approach where the signals are steered via normal switches (typically CMOS transmission gates) and the programming is performed by reading from an off-wafer program store. This introduces new difficulties however:

1. Compared with approach A above, any connection is likely to have a higher resistance, with a 2.5μm CMOS transmission gate having a resistance of about 3.5kΩ typically. This substantially increases the RC product delay and thus potentially reduces the speed. Approach B also requires the introduction of high resistance switches in the path.

2. In both the soft configured approach and the hard wired approach B a switched network is likely to have more area overhead due to the programming switches.

3. Switch configurations have to be read in from off wafer during power up. This introduces an initialization delay and a further area overhead when compared with approach B.

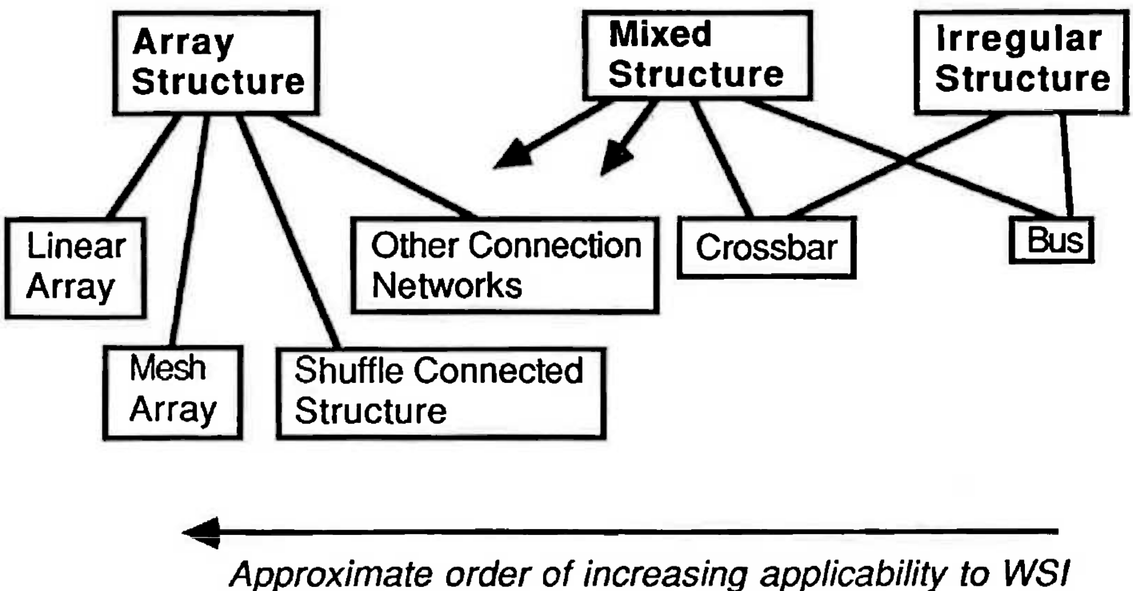


Figure 4.7: Range of architecture topologies and their suitability for WSI

4.3 Architectures for WSI

A possible general classification of the range of architecture topologies available is given in Figure 4.7. These topologies are arranged from left to right in an order indicating their potential application to WSI, those on the left being considered the most suitable. Furthermore these left most architectures would be most suited to soft configured WSI as they have the least intrinsic wiring and switching overhead. This suggested ordering arises because (i) generally arrays containing a single repeated element is an easier structure to provide spares for, and (ii) less richly interconnected structures are more amenable to reconfiguration, requiring less reconfiguration area overhead.

Architectures that have been seriously considered for WSI include one dimensional and two dimensional arrays, tree architectures and irregular architectures. It is the general consensus however that current defect levels are too high for reasonable redundancy schemes to be implemented for irregular architectures [Newman, 1985].

4.4 Tradeoffs in Redundancy Schemes

In designing any electronic piece of equipment the basic aim is to achieve the particular computational task at minimum cost. The cost comes from a variety of sources. In VLSI and WSI the three major cost contributors are the cost of producing the silicon part, the costs associated with testing it and the costs of packaging it. There is some element of tradeoff between these. Testability costs can be reduced by giving over silicon area to testability features. Packaging is mainly a function of final environment, pin count and heat dissipation requirements. These first two are relatively independent of silicon area for WSI. The last however is greatly dependent on clock speed, particularly for CMOS. It may be desirable to use additional silicon area, in the form of more processing elements, to reduce clock speed so that special heat dissipation requirements can be kept down.

For many array problems a linear speed-up in solution time is gained with the addition of extra processors. Thus any number of processors can be used to form the array, the total area being the main factor determining fabrication costs. Furthermore it is assumed that this cost increases with silicon area linearly. If this is the situation then the best processor design is the one that achieves the minimum product of area $\times$ time, where the area and time are those totals required to solve a particular problem.

For a wafer-scale array the area/time tradeoff is a little more complex. The final choice of PE and redundancy scheme is the one that maximizes

$$\frac{\text{Array yield} \times \text{Area}_{PE} \times \text{PE speed}}{\text{Area}_{total}} \qquad (4.1)$$

or

$$\frac{E(P) \times \text{Area}_{PE} \times \text{PE speed}}{\text{Area}_{total}} \qquad (4.2)$$

where Array yield and $E(P)$ are determined as appropriate using the expressions derived for them in Chapter 3, Area_{PE} is the area a PE, PE speed is the speed at which the PE can operate, and Area_{total} is the total array area. This is the same as minimizing area $\times$ time except the functional degradation of usable area is accounted for by the factor $\frac{E(P)\times\text{Area}_{PE}}{\text{Area}_{total}}$. This factor is the area utilization measure suggested by Koren [Koren & Breuer, 1984].

Factors which affect this tradeoff include:

- *PE area.*

  PE area directly impacts yield. The effect of this tradeoff was discussed in Chapter 3.

- *Area Overhead of Reconfiguration Scheme.*

  The area overhead subtracts directly from the space available for the processing elements. It also leads to some level of yield reduction as given by the equations in Chapter 3.

- *Utilization of the PEs by the Reconfiguration Scheme.*

  Improving the utilization achievable by the reconfiguration scheme results in a direct improvement of array yield. This utilization increase is generally only achievable at the expense of area overhead however.

- *Area Overhead of Other Array Constructs.*

  It is also required that some array area be made available to power and clock distribution, and testability and controllability features. Unless special effort is taken, the area of these may contribute to the array kill area for yield purposes. It is absolutely essential to minimize the array kill area in order to keep the yield up.

- *Delays in the Reconfigured Connections.*

  When switches are used to reconfigure the connections a communications delay is introduced. This delay may degrade the speed of the array. There are a number of approaches available for improving this delay, at a cost in area. These will be discussed briefly in Section 4.8.

As mentioned above, there are other cost factors beside area and speed. Good testability features are critical for successful wafer-scale arrays as reconfiguration is based solely on the outcome of these tests. A factor for fault coverage is included in the yield equations given in the previous chapter. The time and equipment required to reconfigure the array is another cost factor. The time is partially influenced by the complexity of the reconfiguration scheme. A more complex reconfiguration scheme is likely to have a higher utilization and thus a higher yield. The equipment requirement is largely determined by the reconfiguration technology used.

4.5 Linear Arrays

A proposed taxonomy of linear fault tolerant arrays is given in Figure 4.8. Formally the problem of producing a wafer-scale linear array becomes that of mapping a 1-D graph onto a 2-D graph containing faulty nodes and links. As an extra dimension

is present a large amount of flexibility is available to the array designer. Because of this linear arrays are generally considered the most suitable structure for wafer-scale integration.

The simplest approach is to ignore the second dimension and embed the chain in a linear structure. Point to point local interconnect (LI) connections or general switched bypass using global interconnections (GI) can be used.

By using point to point connections, LI schemes avoid the problem of additional delays through chains of switches. However for each new point to point connection an additional switch is required on the input and output of the PE. The simplest LI chain schemes requires two inputs and two outputs per PE (Figure 4.9) [Finnila & Love, 1978] [Moore & Day, 1984]. When embedded in a two dimensional mesh four inputs and four outputs are required per PE in the case of a square mesh [Manning, 1977] and six inputs and outputs are required in the case of a hexagonally connected mesh. Nearest neighbour 2-D schemes can result in a higher utilization of good PEs, though complex reconfiguration algorithms are required. Simpler, but lower utilization, algorithms have been developed for the problem of embedding a linear array in a mesh [Aubusson & Catt, 1978] [MacDonald & Neish, 1982] [Aubusson & Gledhill, 1978]. However using these algorithms results in a lower utilization. The array yields of some LI chain schemes and the chip usage (%good × utilization) of an efficient 2-D scheme is reproduced from [Moore & Day, 1984] in Figure 4.10. Nevertheless, because of the ease with which it can be programmed in a distributed fashion, the "Catt spiral" [Aubusson & Catt, 1978] is enjoying some commercial success [Jesshope & Moore, 1986]. The utilization of the Catt spiral can be improved upon if connections are allowed to double back over themselves [Fussel & Varman, 1982][Chamberlain, 1984].

100% utilization can be achieved with the linear GI scheme. The cost comes in the form of long bypass chains, of length $O(lgN)$, that can appear after reconfiguration [Rosenberg, 1983a] [Leighton & Leiserson, 1985] [Greene & Gamal, 1984]. These long chains, with one switch being introduced for each bypassed PE, may introduce considerable delays. These chains can be shortened, at the expense of utilization, by allowing short cuts between rows so that sections of rows can be bypassed immediately [Rosenberg, 1984].

In a 2D GI approach all of the PEs can be used with a longest connection distance of $O(\sqrt{lgN})$. Alternatively the longest connection distance can be reduced to $O(1)$ if only a fraction of the good PEs are required [Greene & Gamal, 1984] [Leighton & Leiserson, 1985].

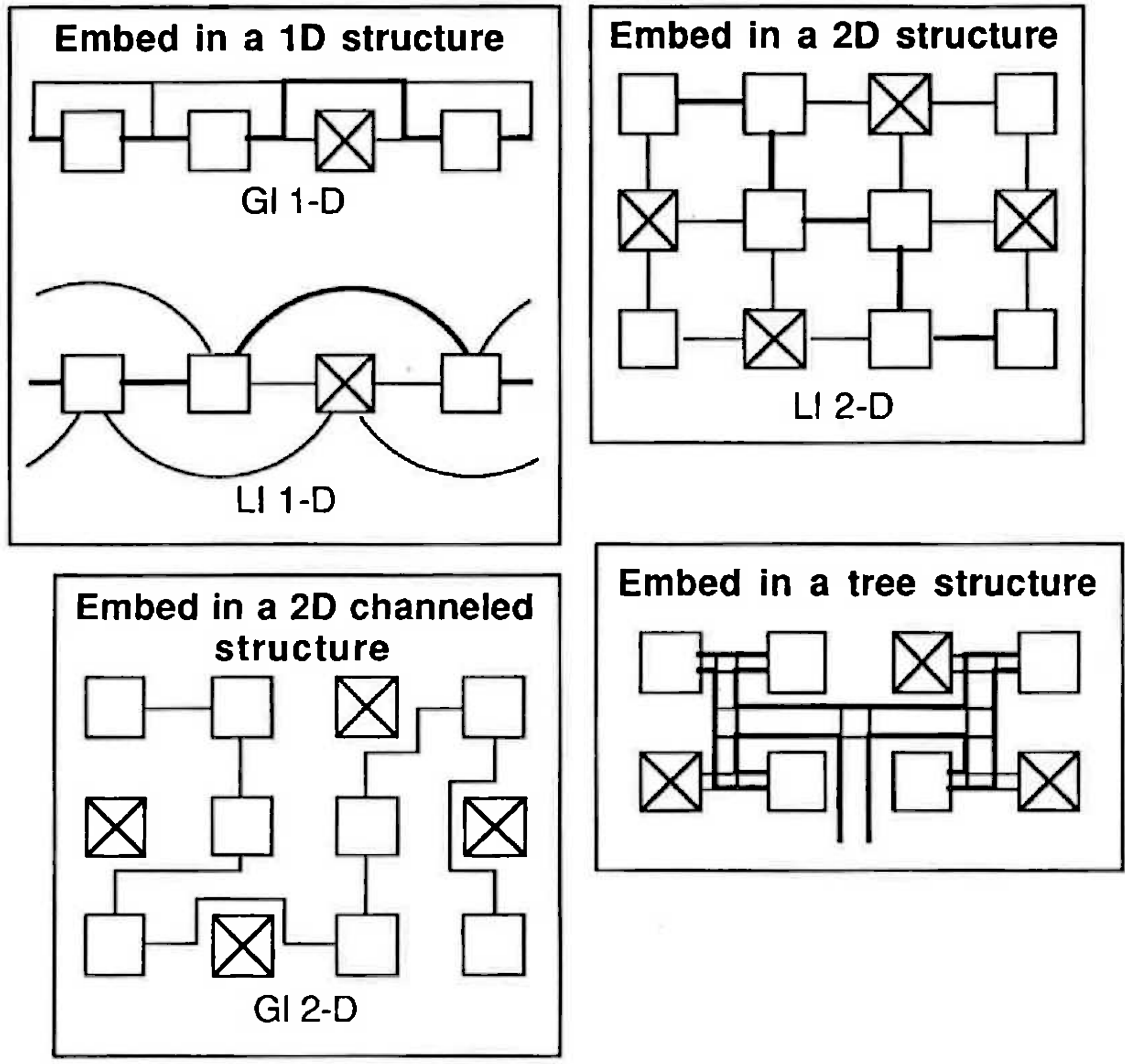


Figure 4.8: Taxonomy of methods used to form fault tolerant linear arrays.

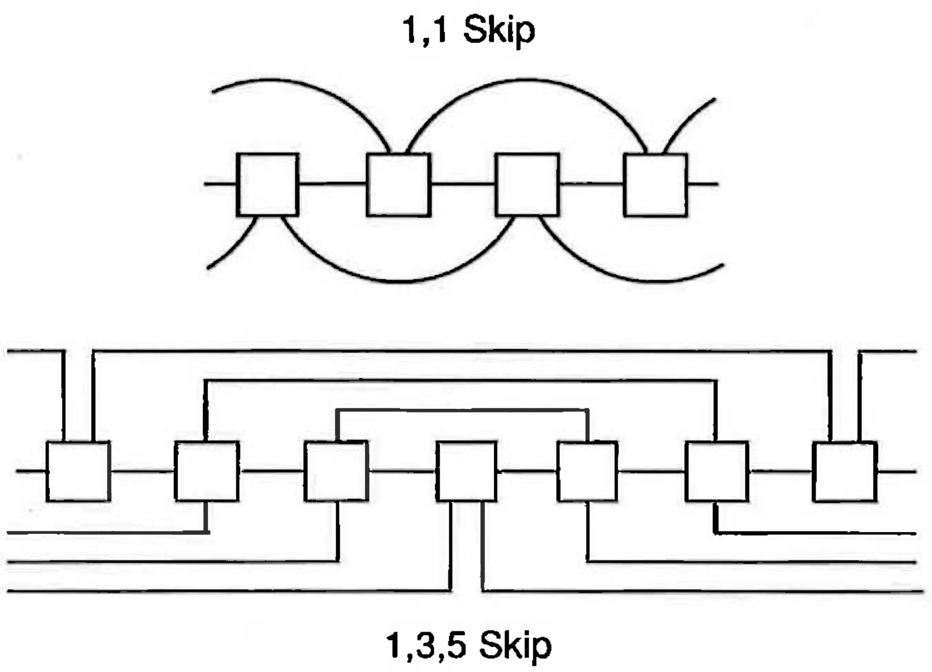


Figure 4.9: Examples of LI chain schemes.

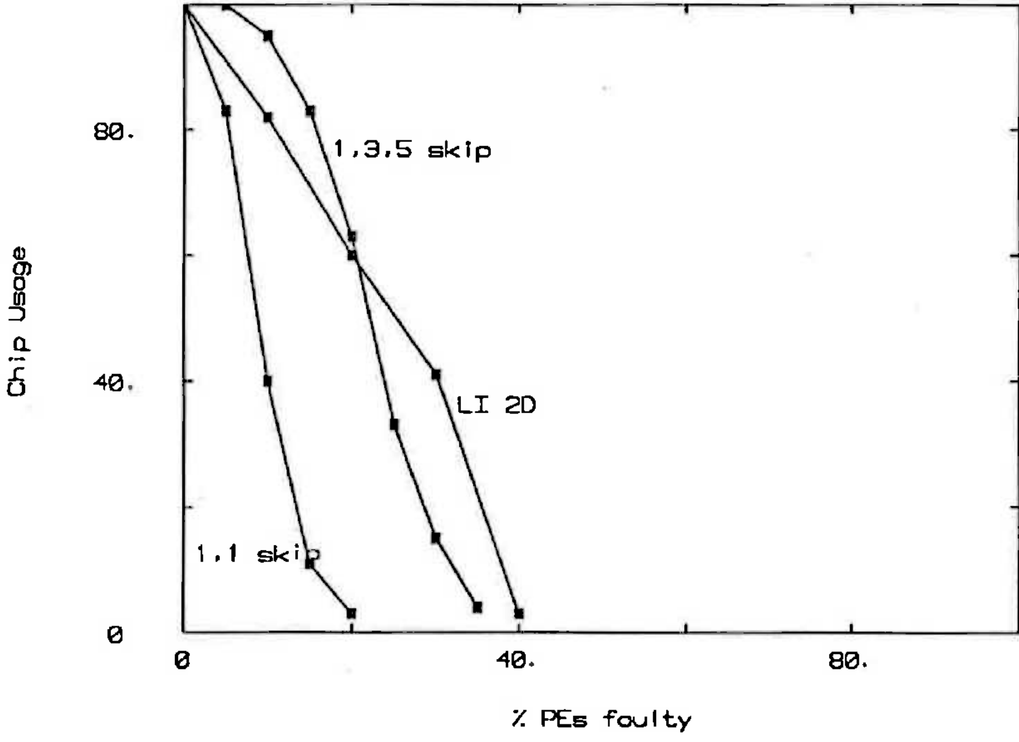


Figure 4.10: Chip usage of LI linear schemes.

An elegant solution, in which the linear array is embedded in a binary tree structure, was presented in [Bhatt & Leiserson, 1982]. Again however the longest connection is of length $O(lgN)$. This result is guaranteed for all arrays as the root must always be traversed at some stage.

As a general rule reconfiguration schemes for linear arrays can achieve close to 100% utilization without an inordinate impact on speed or area overhead, at least as when compared with mesh arrays.

4.6 Mesh Arrays

Two dimensional arrays are often the best means for mapping a problem onto silicon. For many applications they are a natural architecture, for example in image processing problems. Matrix operations are also best carried out on 2-D arrays, whether hexagonal or mesh, for the same reason. An equivalent linear array (or more likely set of arrays) with the same functionality and speed would often be a lot larger in area. Dynamic programming and sorting are also functions that can be carried out efficiently on mesh structures. It is expected that the next generation of multicomputers will be based on mesh networks [Athas & Seitz, 1988].

The drawback of WSI mesh arrays, compared with linear arrays, is that more complex reconfiguration approaches are needed to achieve similar levels of utilization. This results in lower yields for greater wiring overhead areas.

A taxonomy of interconnect reconfiguration approaches to mesh arrays is given in Figure 4.11. Examples of these approaches, with comparison results will be given in Chapter 6.

4.7 Tree Structures

Implementing a fault-tolerant tree structure requires even a greater number of reconfiguring links than are required for mesh arrays. For this reason comparatively little work has been done in this area. However trees, particularly binary trees, are an important computational structure lending themselves naturally to the efficient solution of problems such as sorting and searching. Because they have a shorter node to node communications latency than mesh arrays, many multiprocessor structures have been proposed that use tree structures. Thus it is important to investigate possibilities in this area. As will be seen however most fault tolerant tree structures with sufficient

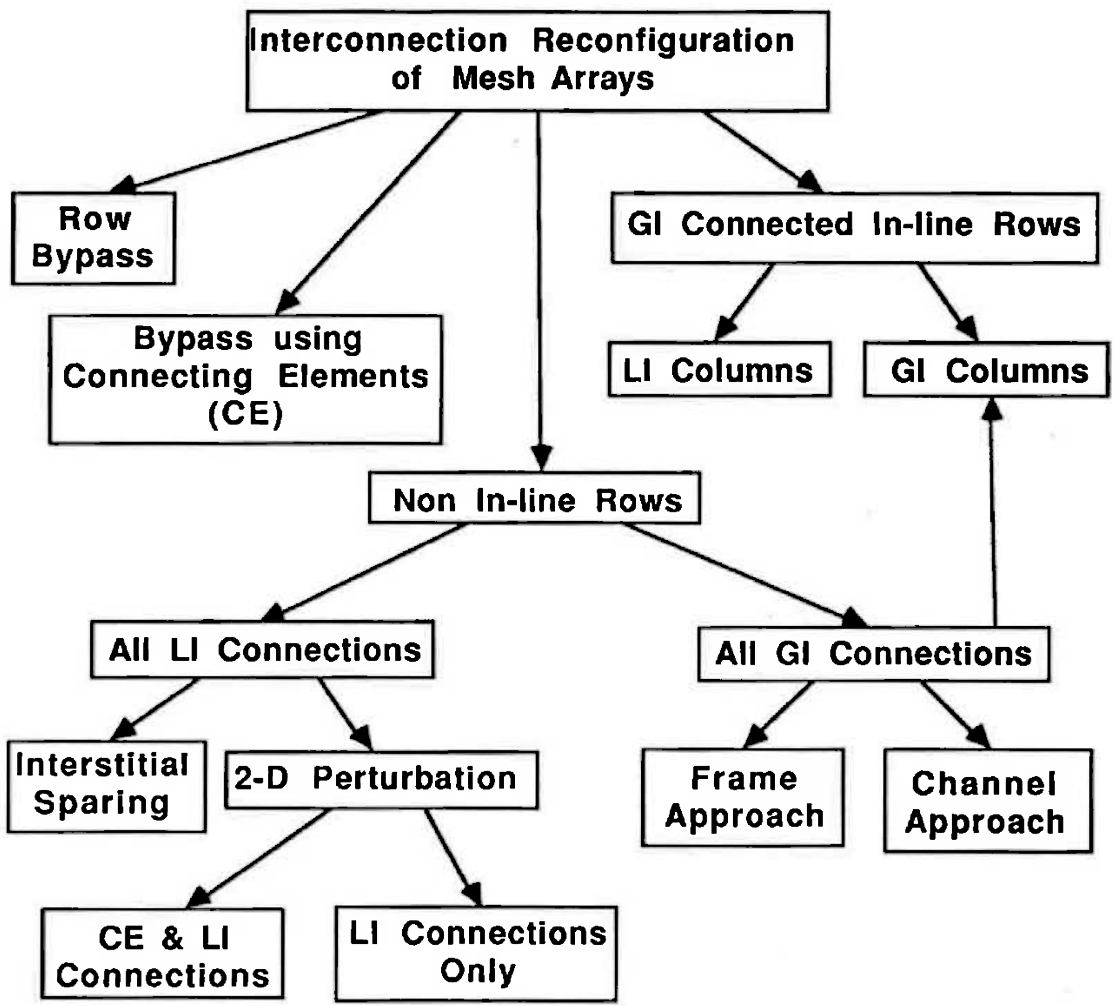


Figure 4.11: Taxonomy of approaches to interconnect reconfiguration of mesh arrays.

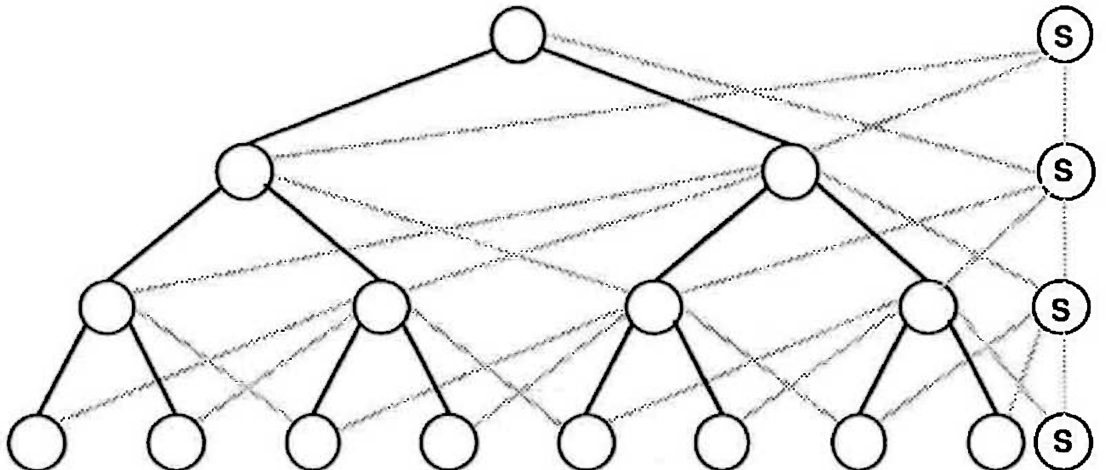


Figure 4.12: First level redundancy approach to implementing fault tolerant trees.

spares for consideration for WSI require many long additional communication paths for implementation. This introduces additional area and delay time costs.

Four basic approaches have been suggested for implementing fault tolerant tree structures:

1. *Level Redundancy.*

   If all nodes are PEs then spares can be provided at each level of the processor array [Raghavendra et al., 1983]. Alternatively, spares can be provided at just the lower level if all of the parent nodes are switches [Harden & Strader, 1988]. This of course makes the correct functioning of the parent switches critical to the correct functioning of the array. These two approaches are illustrated in Figures 4.12 and 4.13 respectively.

2. *Modular Redundancy.*

   A module is a two-level subtree with one spare node and switches that allow this spare to replace any of the other nodes in the tree [Hassan & Agarwal, 1986]. A fault free tree can then be built out of these modules.

3. *Subtree Orientated Fault Tolerance (SOFT) Approach.*

   In the SOFT approach spare nodes are distributed at the bottom of the tree and brother nodes are connected, as illustrated in Figure 4.14

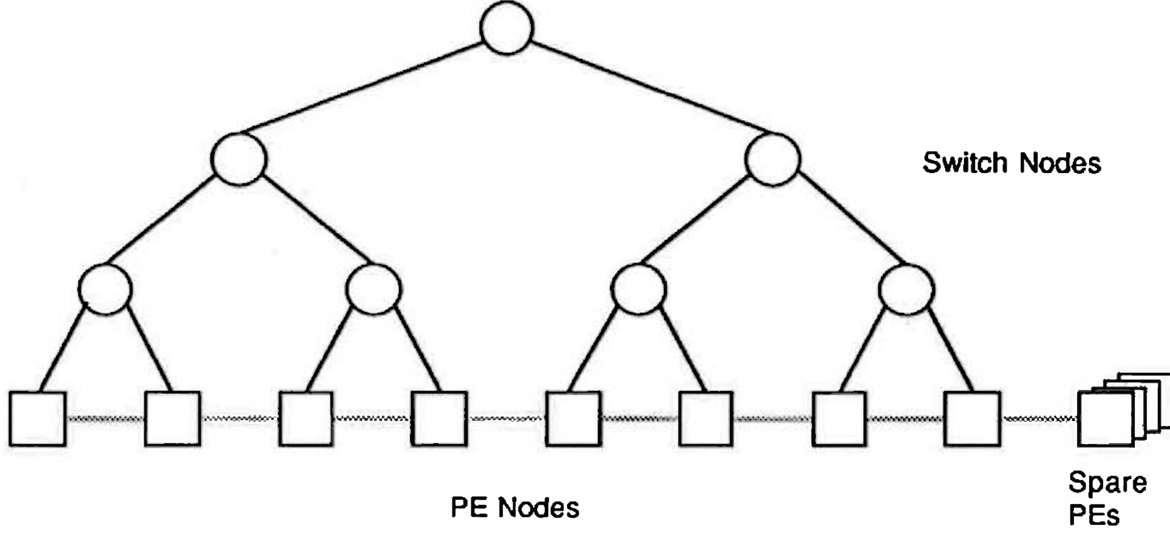


Figure 4.13: Level redundancy on one (PE) level only.

[Lowrie & Fuchs, 1987]. Up to 50% sparing is possible. However many of the parent nodes, particularly near the top of the tree, don't acquire any fault tolerance from this approach.

4. *Bussed Sparing.*

   A bus structure is added to the binary tree that allows replacement of any node with any spare. The spares are all connected to the main trunk of the bus as illustrated in Figure 4.15 [Howells & Agarwal, 1988].

Detailed comparison of these approaches won't be carried out here. For more details see the relevant papers, particularly [Howells & Agarwal, 1988]. The main common element to note with all of these approaches, when compared with fault tolerant mesh and linear structures, is that the introduction of fault tolerance to binary tree structures introduces the following negative attributes:

- Several structures have a large non-fault-tolerant core, usually at the root of the tree. This would decrease wafer yield.
- All structures introduce long links going through many switches. Compared with mesh arrays, this results in longer and slower switched paths and greater wiring overhead.

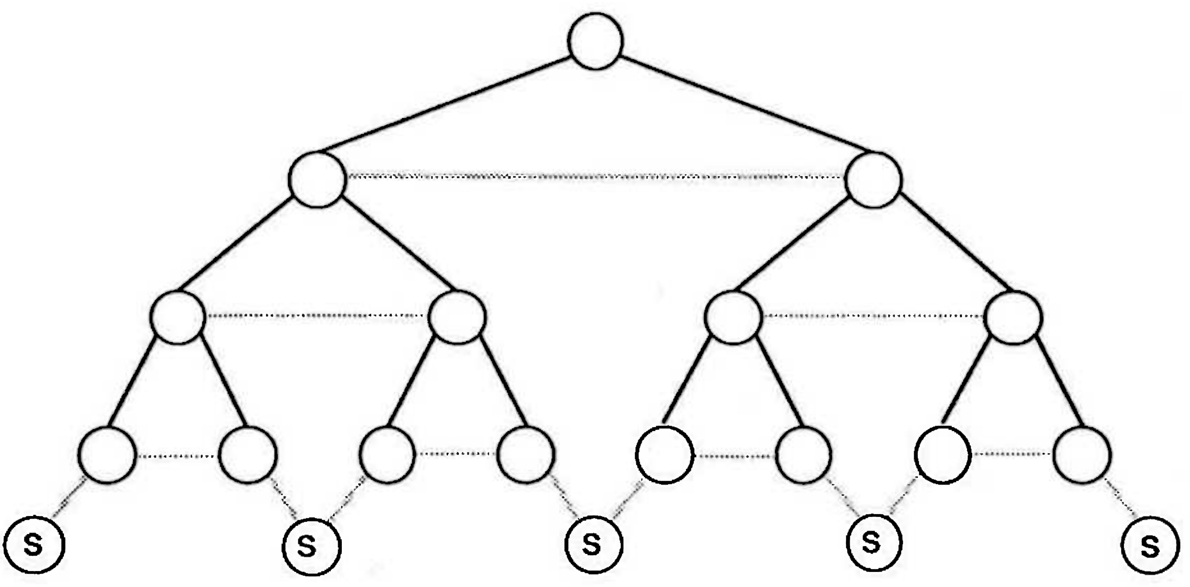


Figure 4.14: SOFT approach to fault tolerant trees

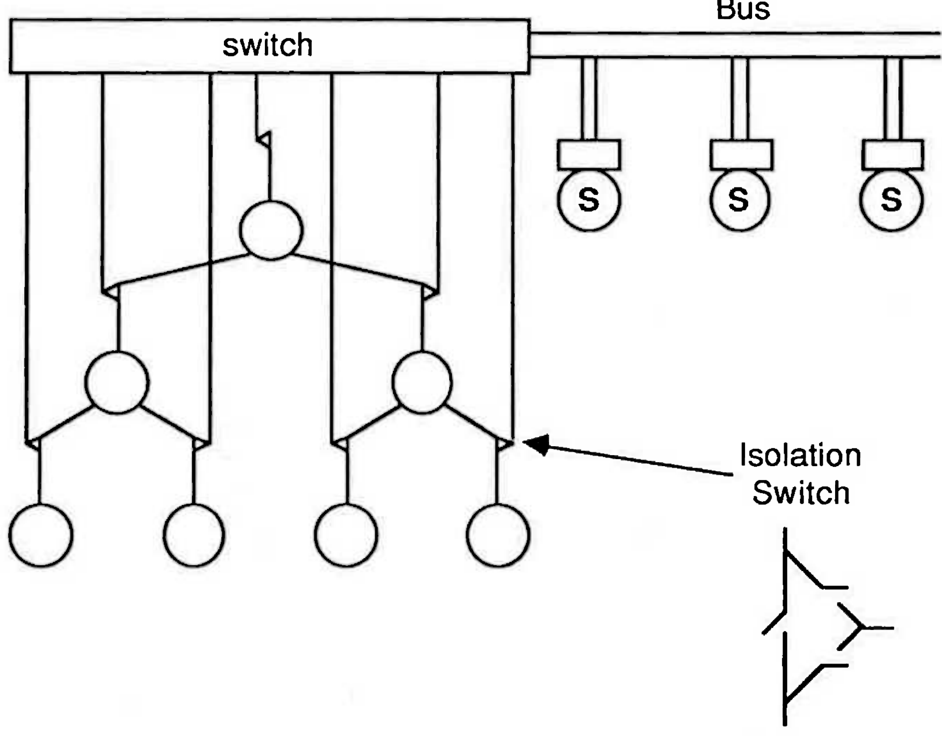


Figure 4.15: Bussed sparing approach to tree fault-tolerance.

4.8 Maximizing the Speed of a Wafer-Scale Array

Most reconfigured lines act as RC delay lines. If the reconfiguration is carried out using laser fuses, referred to as a "hard" reconfiguration technology, the switch resistance is normally of the order of 1-100Ω. On the other hand if normal CMOS transmission gate switches are used, as is the case with most WSI approaches, resistances of the order of 1kΩ or more are introduced. This "soft" configuration approach of using CMOS switches is preferred by many researchers because it requires a lower technology investment and allows for field reconfiguration. There are three basic approaches that can be used to minimize the inter-PE communications delay, or alternatively, minimize its effect:

1. *Employ fewer switches in the inter-PE path.*

   This can be done generally at the expense of either utilization and/or increased wiring area. LI schemes, as a rule, use fewer switches per path than GI schemes but at the expense of both of these extra costs. The optimal solution can be sought by evaluating the *yield* × *area* × *time* metric for each approach.

2. *Increase the sizes of switches and drivers.*

   Larger switches and drivers have lower resistances and thus result in decreased delays. Interspersing drivers in the switched paths may also result in decreased delays. An array using this approach is discussed in the next chapter.

3. *Communicate asynchronously.*

   If the PEs only communicate on say one clock cycle out of four, which can be relatively common in many array structures, then asynchronous communications, of up to this time delay, can be used between PEs. (*eg.* If each PE is doing one butterfly in an FFT, each butterfly would actually take many clock cycles to complete and by staggering computation times many cycle communications can take place without affecting throughput.)

4. *Pipelined Communications.*

   By placing a register at each switch the inter-PE communications can be pipelined so that the throughput can be maintained at the expense of latency and increased switch area [Kung & Lam, 1984]. This is relatively straightforward for linear arrays but is more difficult for mesh arrays where the inter-PE delay has to be the same for each path. An example is illustrated in Figure

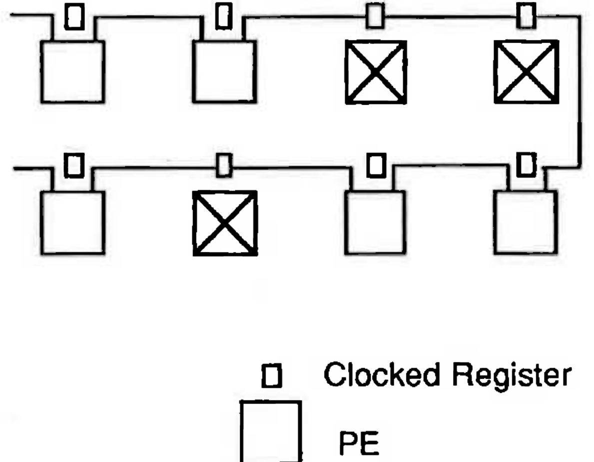


Figure 4.16: Pipelining the interconnect can be used to maintain throughput.

4.16. Katevenis and Blatt suggested that the capacitive lines themselves can be used as the storage elements in a pipelined scheme [Katevenis & Blatt, 1985]. They also suggested that if communications is not required on each clock cycle then connections can be multiplexed on the same line.

5. *Use of a BiCMOS Process.*

   This allows fast bipolar switches to be used for the interconnect and compact, low power CMOS switches to be used for the functional units.

4.9 Conclusions

Redundancy for WSI is best achieved by restricting the architecture to an array, and providing global spares that can be used by first reconfiguring the interconnect between the processing elements. A number of techniques can be used to provide a reconfigurable interconnect. "Hard" techniques, such as laser fusing and joining, require special equipment and can not be used to repair in service failures. "Soft" techniques, using transistor based switches, avoid both of these drawbacks but may result in low communication speeds. Methods are available to surmount this problem if necessary.

Linear arrays are the best architecture for WSI because they have the least interconnect. A taxonomy for linearly connected arrays is suggested above. Mesh

connected arrays are the next best architecture in general. The remainder of this thesis shall be primarily restricted to the discussion of defect tolerant mesh arrays.

Chapter 5

'Chip Frame' Scheme for Reconfigurable Mesh-Connected Arrays

5.1 Introduction

Reconfiguration schemes for the wafer-scale integration (WSI) of regular arrays of computation cells use additional data paths and switching circuitry to bypass faulty cells and connect functional cells into a working array. Both the granularity (e.g. area or number of logic gates) of individual cells and the width (i.e. number of data bits transferred in parallel) of data paths between cells strongly impact upon the area and circuitry overhead associated with array reconfiguration. Fine-grain cells (with higher individual cell yields) and serial data paths (minimizing area overhead and throughput rate degradation due to reconfiguration paths and switches) have so far been favored to obtain practical WSI arrays which approach the capabilities of ideal (fault-free and without reconfiguration) WSI arrays.

However, there are many applications requiring large-grain cells to obtain the highest achievable local computing rate and requiring wide data paths to achieve maximum communication rates between cells. The "chip frame" scheme discussed here was motivated by the problem of implementing a high-performance, mesh-connected data communications environment [Tewksbury & Hornak, 1987] for a massively parallel mesh-connected computing system.

The network nodes in this application are (distributed) intelligent routing and control functions requiring areas comparable to VLSI IC's. However, design and

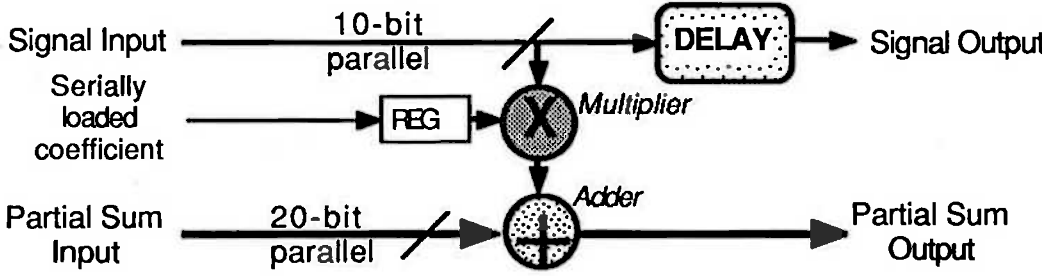


Figure 5.1: FIR Filter Section with Fixed, Preloaded Coefficients for Testbed Array

analysis of the functional behavior of such a distributed control communication function is a complex problem, extending beyond the basic issues of concern in WSI realization of arrays of large-grain cells. For experimental evaluation of such WSI issues, it was decided to initially implement a reconfigurable mesh-connected array whose cells are parallel data, pipelined FIR filter sections [Tewksbury et al., 1987] rather than mesh network routing cells. The FIR filter section, shown schematically in Figure 5.1, had already been designed and fabricated by Hatamian and Cash [Hatamian & Cash, 1987] in the 2.5μm CMOS technology targeted for the reconfiguration chip frame. Simple examination of the bit-level data paths in Figure 5.1 shows that, even though portions of an FIR filter section may be faulty, there are likely to be some bit-level paths which will function correctly. Since the individual bit lines of a parallel data path are treated the same by the reconfiguration scheme, the performance degradation due to reconfiguration can be evaluated using individual bit-level lines, even though some cells may have internal faults. Furthermore, the highly pipelined FIR filter section is internally dominated by short data paths between register stages, achieving high internal throughput rates of about 80 MHz, well beyond the throughput rate expected on the longer reconfiguration paths. The reconfigured paths then are the speed limiting paths, allowing clear evaluation of the speed degradation under various reconfiguration states. For such reasons, the FIR filter sections provide a convenient testbed array cell for the evaluations pursued here.

In general, ideal WSI realizations of large-grain cells will suffer from low cell yields, causing a considerable area overhead and significant speed degradation when augmented for reconfiguration. However, in the case of large-grain cells, a better reference for comparison is probably a conventional realization, with packaged cell functions interconnected on a printed wiring board. Such a realization itself introduces poor area

utilization (i.e. actual chip area is small relative to the printed wiring board area) and typically exhibits a speed degradation between cells relative to the achievable on-chip speeds. From this vantage point, WSI realization of large-grain cell arrays may be attractive despite the high area overhead and speed degradation relative to an ideal WSI reference. Should low cell yields impose too severe a restriction, a pseudo-WSI approach [Franzon et al., 1987][Tewksbury & Hornak, 1987] here called hybrid-WSI (HWSI) may be useful. HWSI mounts unpackaged VLSI IC's on a wafer-size silicon integrated substrate (containing, in addition to passive interconnections, line drivers, test circuitry, and other small-scale communication circuitry). The project described here implements a monolithic array of cells on one half of a silicon wafer and a HWSI circuit board on the other half of the wafer [Franzon & Tewksbury, 1987]. This reflects our interest in stacked, wafer-level modules (WSI or HWSI layers, depending on compatibility with WSI) [Franzon et al., 1987].

5.2 Reconfiguration Scheme

Mesh-connected arrays of identical large-grain cells with wide parallel data paths between cells impose at least two major constraints on the reconfiguration scheme:

1. Relatively low (<50%) cell yields and relatively small array sizes (e.g. less than about 15×15) favors high usage ("utilization") of functional cells in constructing a working array (*eg.* [Franzon, 1986b].)

2. Wide data paths favor a minimum number of extra switches and connections added for reconfiguration.

This section will describe how the design for the "Chip-Frame" approach arose out of a consideration of these, and other, factors.

5.2.1 Approaches to Fault Tolerance

A number of tradeoffs are made in choosing any fault tolerant scheme. As most schemes involve reconfiguration of the interconnect between processing elements the technology with which this is to be done must first be decided. Then it is necessary to choose the topology of interconnect that forms the basis of the fault tolerant array. This section briefly reviews various approaches to fault tolerance for WSI and indicates how this scheme arose from consideration of these options.

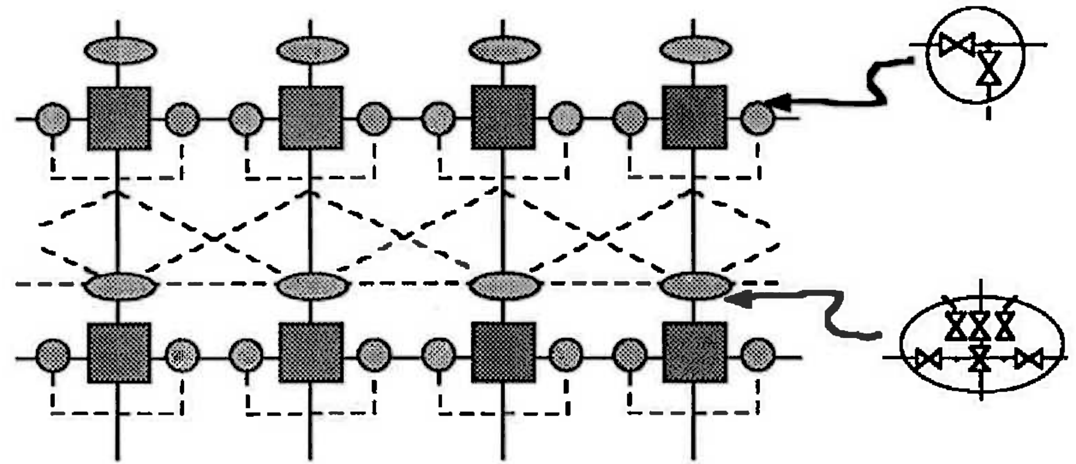


Figure 5.2: Row-oriented reconfiguration (with pass-transistor based switches)

It was decided to use conventional production technologies as much as possible. Thus a soft-switching reconfiguration scheme, using non-permanent switches was employed. Another reason soft switching is used, rather than make-link and break-link techniques, is because the latter do not save appreciable area, provide no in-field reliability enhancement potential and involve special processing techniques. The attendant disadvantage is however that the switches are volatile.

A comparison of different fault tolerant schemes for mesh architectures can be found in [Franzon, 1986a]. From a comparison of different fault tolerant schemes for mesh architectures [Franzon, 1986a] (see Chapter 6 also), it was shown that the use of external switches to reconfigure the data paths was generally preferable to the use of switches internal to the PEs. Internal switches limit the PE to connections with nearest neighbours only. An important figure of comparison, in demonstrating this, was that of "average PE utilization", which was defined as that percentage of functional PEs that can be used in an array, for a particular PE yield.

One topology employing external switches that was particularly successful, in terms of utilization, over a wide range of PE yields was first described in [Moore & Mahat, 1985] and is illustrated in Figure 5.2. Here it can be seen that the scheme employs two lines of bypass between each row and two additional lines between each column. The array is reconfigured by organizing the columns through the vertical, diagonal and horizontal column bypass lines and then organizing the rows along each physical row bypassing unused PEs as appropriate. The main limitation with this scheme, particularly in the yield area of interest (PE yields of about

50%), is that the utilization of good PEs is limited by the row with the least number of functioning PEs. That is because each column has to use one element from each row and thus the number of columns is potentially limited by the row with the least number of functioning PEs. Even adding extra horizontal column bypass lines does not improve the utilization of good PEs significantly.

Another problem is that the "hard core" of circuits that must work for the array to function is quite large. Because the logical rows map directly onto the physical rows all of the row lines and the row steering logic must work for the array to work. This sizable "hard core" can be a significant yield detractor [Franzon, 1986b]. Thus the basic improvement made here was the addition of lines to allow the steering of rows off the original rows. This necessitated the provision of vertical bypass for the column lines as well. This was done by providing two segmented ring buses for each PE, one bus for the horizontal or row lines, and another for the vertical or column lines. This, along with the switching topology, is shown in Figure 5.3. Note that the total number of lines placed between the columns and rows has now been effectively doubled over the row orientated scheme just discussed. In addition the number of switches required to steer each bit has been quadrupled from 7 to 28. A further important enhancement suggested is in the way that the diagonals are handled. It was pointed out that the crossing diagonal connections, are never used simultaneously and thus could be replaced by the simpler Manhattan connections seen in Figure 5.3. An overall view of how the scheme looks, laid out as a mesh, is given in Figure 5.4. An example of how reconfiguration takes place around the frames is shown in Figure 5.5.

Memory and control is provided at each PE site to determine which two of the eight external connections to each ring bus is used, how the ring bus is segmented and whether it is connected to the PE or not.

In the scheme presented here the "hard core" of circuits that are critical to correct functionality is reduced to a bare minimum. This was achieved by the following means:

- Allowing the logical rows of the connected array to be routed amongst different physical rows means that faulty bypass circuits can themselves be bypassed as well. As faults often cluster, a PE fault may also be accompanied by a nearby fault in the bypass so this "bypassing the bypass" is quite important.

- As the logic controlling the bypass is local to each PE, a fault here can be bypassed just as easily.

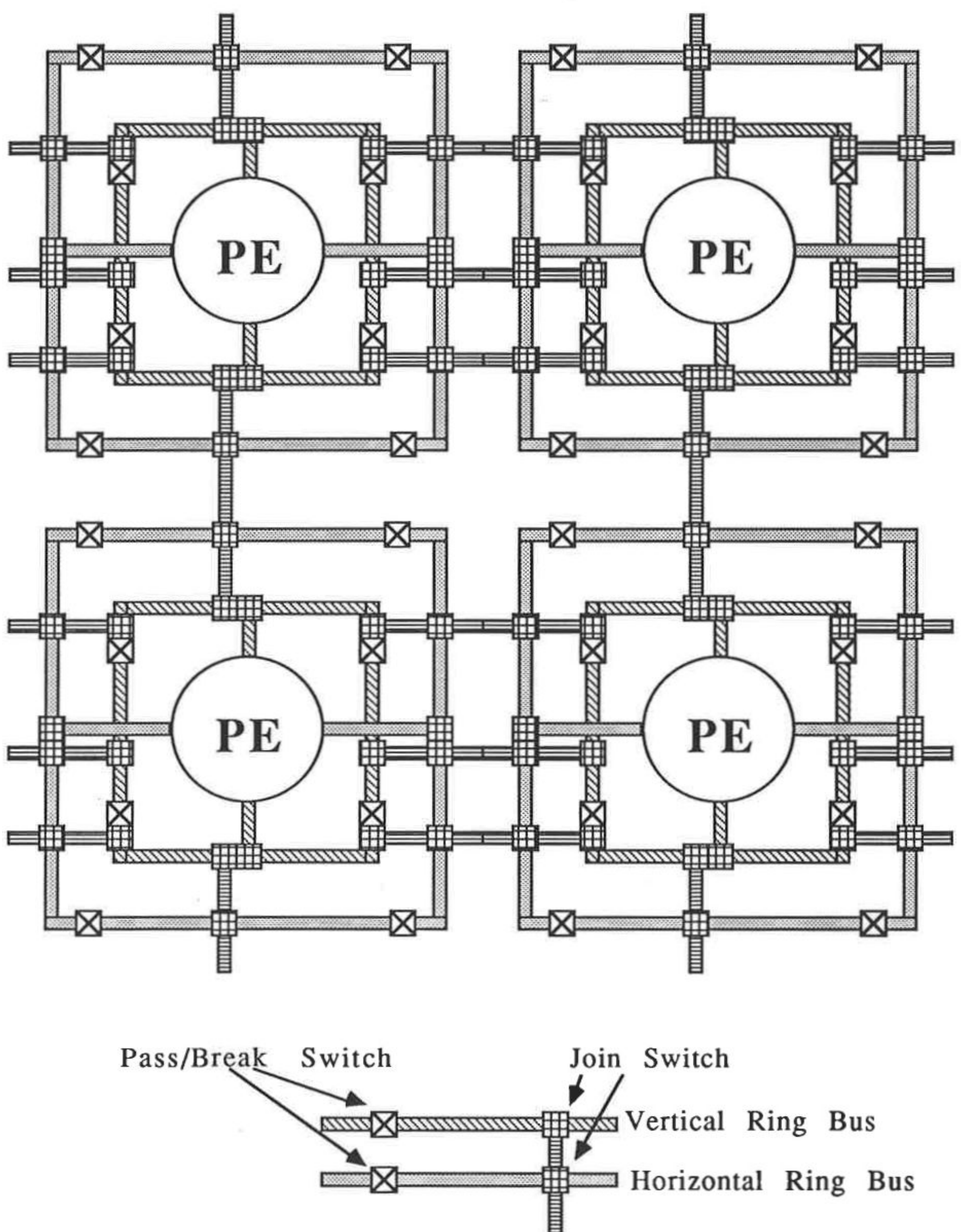


Figure 5.3: Basic Outline of the Reconfiguration Scheme.

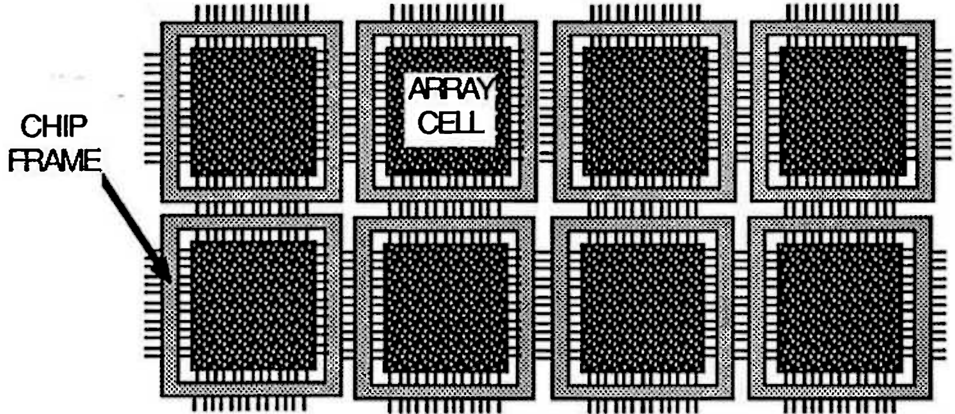


Figure 5.4: Mesh-Connected Cell Array with reconfiguration Frame per Cell

- The power and clock connections are a major contribution to the "hard core" of the wafer that must work. These have been made defect tolerant also. Basically, sufficient redundancy is provided in these lines so that they can physically be restructured, by laser cutting, once faults are detected. The layout of the fault tolerant power rails is illustrated in Figure 5.6.

Another scheme that has been proposed using switches that are entirely external to the PE is illustrated in Figure 5.7 [Hedlund, 1982]. With this approach the switches are not even loosely associated with the PEs, as has been suggested here, and this could have been expected to lead to greater area usage efficiencies. There are several factors that would indicate that this is not the case.

- At first sight it would appear that as this scheme requires only two lines per inter-PE channel the overall area overhead would be reduced. However the switches required are far more complex than the ones required for the frame scheme, with each switch having at least 7, if not 9 or more connection states. This sort of switch would consume considerable silicon area. In fact each 8 bit switch requires an area of 250 by 250 lambda [Hedlund, 1982].

- Unlike the frame scheme the control of the switches is not localized around each PE. This introduces considerable extra complexity in the control logic and the reconfiguration strategy, leading to several problems:

  - Each switch has to be fully controlled through states set in a memory. The amount of memory required works out to at least 32 bits per PE,

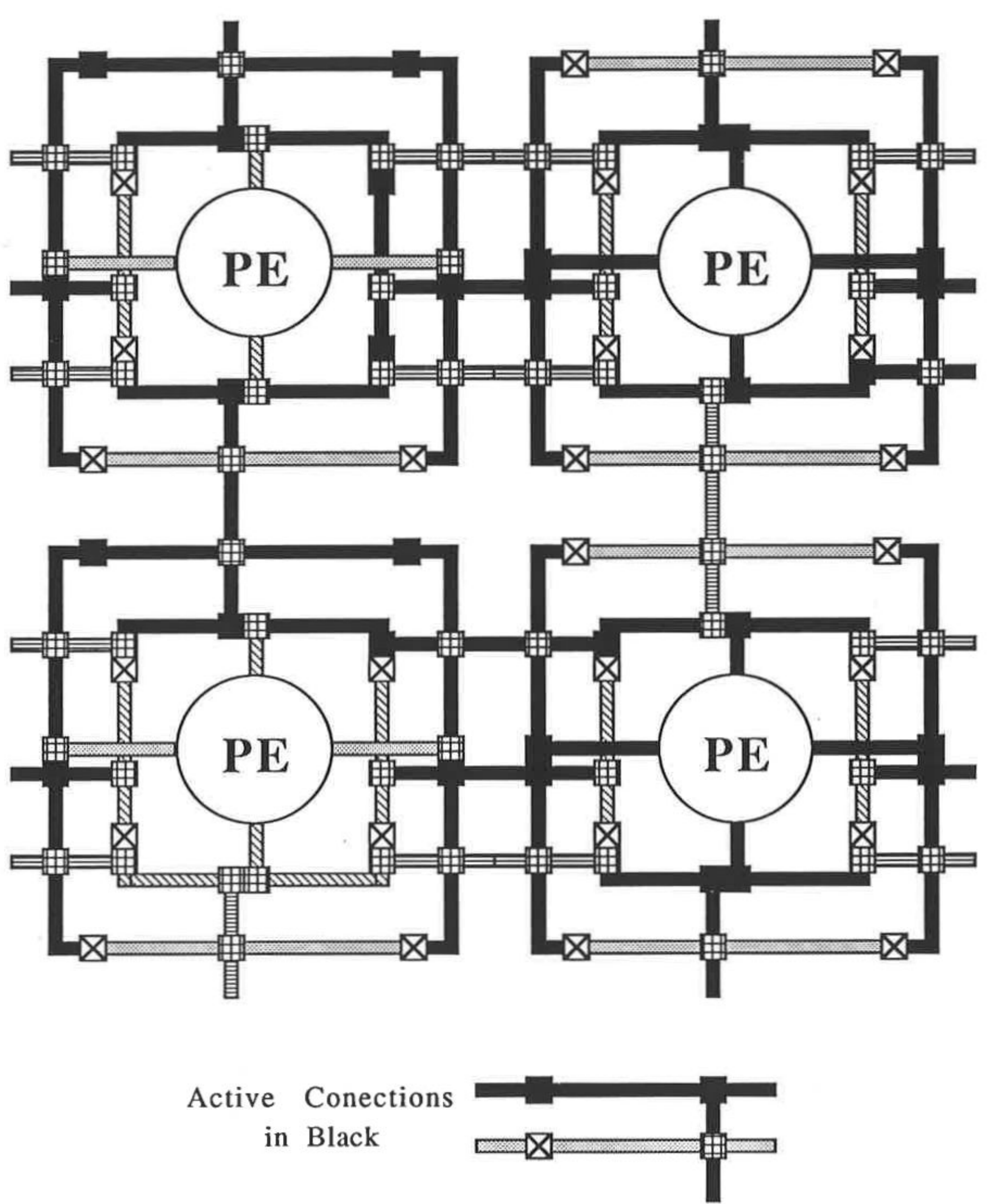


Figure 5.5: Examples of some reconfigured frames.

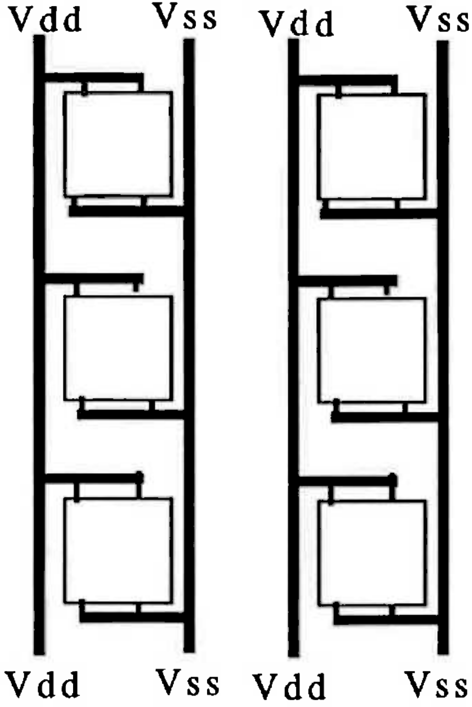


Figure 5.6: Fault Tolerant Power Distribution.

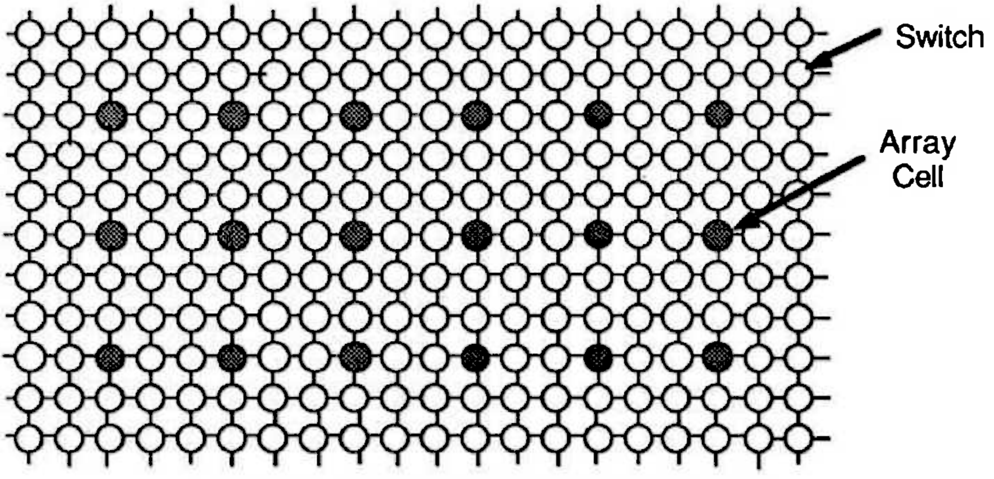


Figure 5.7: Sea-of-switches reconfiguration

compared with 13 bits required in the frame scheme. Feeding this amount of memory would thus be a proportionately larger problem and making this feed mechanism fault tolerant would become more necessary. This fault tolerance would consume more area again.

- As the data routing is completely separate to the PEs the formulation of the reconfiguration scheme becomes a more difficult problem. In [Hedlund, 1982] a simple hierarchical scheme is suggested where the array is divided into blocks of 12 PEs of which only 4 are required. By taking this simple approach efficient utilization of surviving PEs is obtained only for a narrow range of PE yields. More complex divide and conquer approaches, based on graph theoretic considerations, have been suggested that provide better utilizations for this type of structure [Greene & Gamal, 1984][Leighton & Leiserson, 1985], but the actual computation of array configurations would still be a difficult task as complex pattern recognition steps are required.
- The completely generalized switches may make array testing a more difficult problem.

A more detailed description of the frame scheme will now be provided.

5.2.1.1 Detailed Scheme Description

Each interconnection line and switch shown represents N lines and switches for the N-bit, parallel data path. The frame basically provides two data path rings around the cell. Segmentation switches isolate particular segments of the ring to connect data incident at one ring port to another ring port. Access switches provide either cell-to-frame or frame-to-frame connections at each external port of the frame. The connectivity of the frame is quite flexible, despite its apparent simplicity. Any of the 8 edge links can be used for the required up/down/left/right bypass connections, providing a total of 70 possible frame reconfiguration states, with about half of these states being useful. Although only 7 bits of control information are needed to specify one of 70 states, it is more efficient to directly specify the state of each side of the frame. This greatly reduces the complexity of the control state decoding logic at the modest cost of adding a few more flip-flops to store control state information. In Figure 5.8, three control bits are used per side and a single bit specifies whether the internal cell is functional, giving a total of 13 control bits per frame. These control

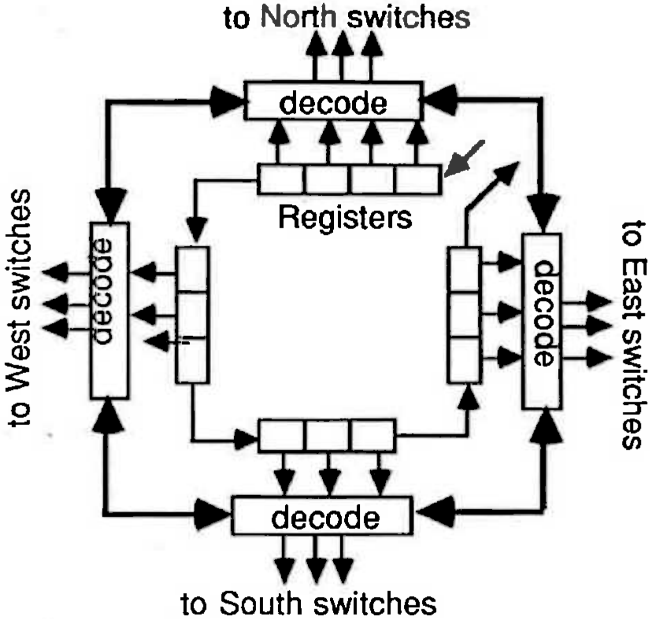


Figure 5.8: Programmable Control of Reconfiguration State.

bits are loaded serially, as illustrated in Figure 5.9 for the whole wafer, adding slightly to the "hard core" circuitry on the wafer.

Although the chip frame is designed to be consistent with the mesh-connected communications network function discussed earlier, the FIR filter sections actually used are connected in serial chains. The reconfiguration states allow reasonably good interconnection of such serial chains by connecting the FIR filter section's inputs to both the left and top frame edges and the section's outputs to both the right and bottom frame edges.

Ideally the data paths would be bidirectional. However the large-grain cells introduce relatively long bypass interconnections (even for a single cell bypass.) To maintain throughput rates, line drivers must be provided along the interconnection paths, establishing a line directionality.

The flexibility inherent in the scheme is best illustrated by looking at the set of bus configurations allowed. Figures 5.10 to 5.13 give the full set of bus configurations for the vertical ring bus, the horizontal ring bus having a similar set of (rotated) configurations. Control signals U0 to U7 and D0 to D7 indicate which connections are considered active for connecting the "Up" PE (*ie.* the logical vertically preceding PE) and the "Down" PE (or the logical vertically succeeding PE), as illustrated in Figure 5.14. The signal "Co" indicates whether the PE is connected (Co = 1) or

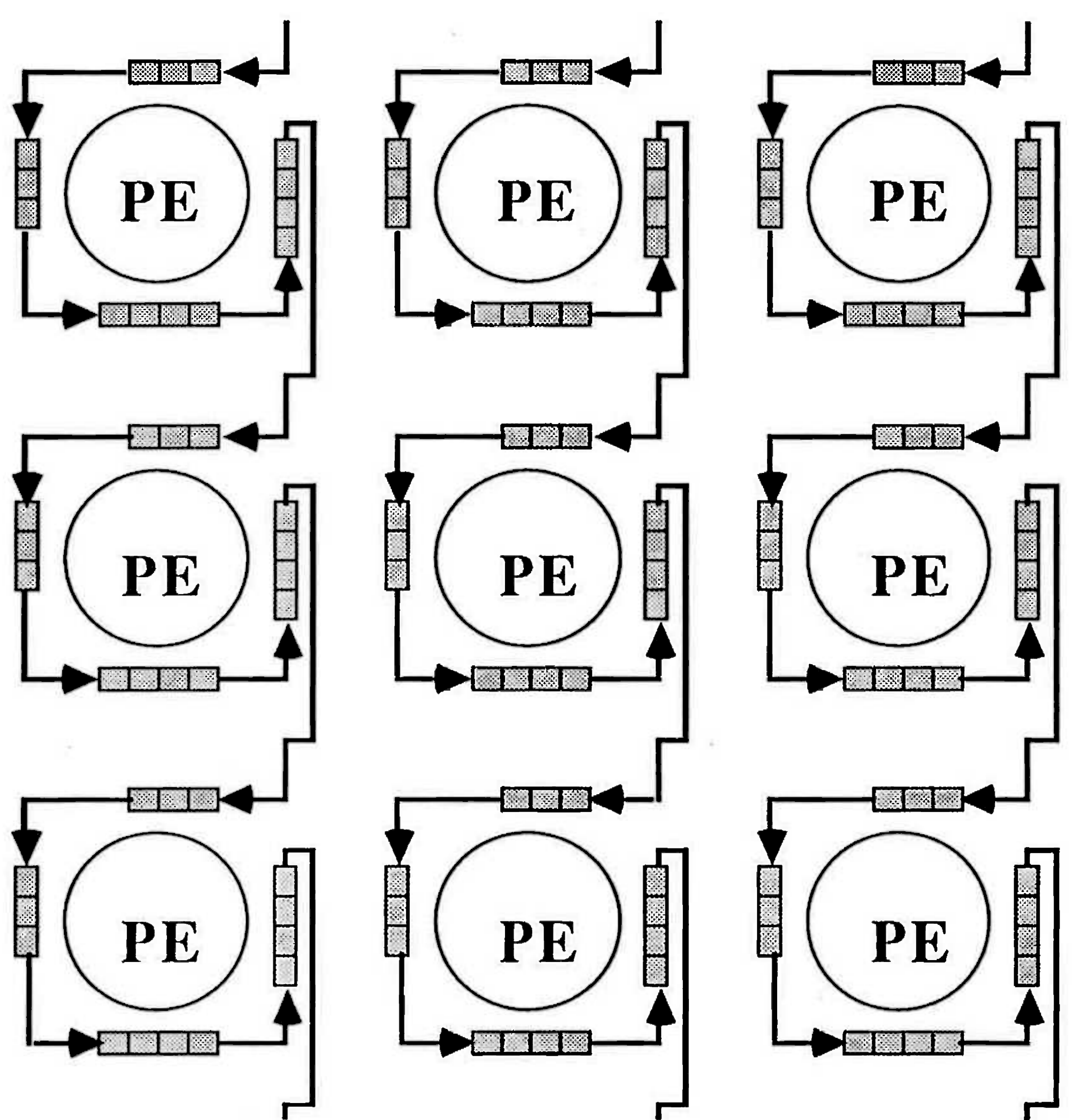


Figure 5.9: Programming the Wafer-Scale Array.

being bypassed (Co = 0.)

5.3 Global Reconfiguration in Response to Faults

Due to the flexibility provided, no simple algorithm can be given for global reconfiguration in response to faults.

Certainly there is a tradeoff between the complexity of the computation required to reconfigure the array and the utilization of the good processing elements. For example, the thrust of [Hedlund, 1982] was the choice of a relatively simple reconfiguration approach thus resulting in a lower than optimal good PE utilization, but handling the complexity problem.

On the other hand to do a full search of all the possible reconfiguration choices for an optimum would be prohibitive, even for a small array. Fortunately this search space can be significantly reduced, on average, by conducting a depth first search with the following considerations allowing for considerable pruning of the search:

- An important design factor for a WSI array is the speed with which it can operate. A speed limiter which has to be reduced as much as possible is the inter-PE delay through the wires and switches of the links. Every PE bypassed adds significantly to this delay. Thus, for performance reasons, the maximum number of PEs that can by bypassed by any inter PE link must be fixed. The bypass is kept to a distance of one for arrays with up to about 30% of PEs faulty and increased after that point is reached. This requirement also results in a reduction of the search space for possible solutions. However it also might mean that no solution is possible for certain fault patterns. In these cases speed can be traded off for utilization, if this is acceptable.

- If the optimum solution is desired then the whole search space must be covered. However if one is willing to accept a slightly less than optimal solution then such a solution can often be found amongst the first few possibilities searched. The criterion for accepting such a solution can be based on an acceptance level of the utilization achieved for good processors. This acceptance level would be determined by two factors:

    1. Experience gained from reconfigurations performed on arrays with similar fault levels and fault patterns.

    2. Results obtained from earlier full depth searches of this search space.

Vertical Bus Configurations	
Active Internal Control Bits	Bus Configuration
Co = 1 One of U7, U0 or U1	One of
Co = 1 One of D5, D4 or D3	One of
Co = 1 U6	
Co = 1 U2	

Figure 5.10: Allowed vertical bus configurations (Figure 1 of 4).

Vertical Bus Configurations	
Active Internal Controls	**Bus Configuration**
Co = 1 D6	
Co = 1 D2	
Co = 0 Any of these pairs: U7 & D3...D5 U0 & D3...D5 U1 & D3...D5 U5 & D7, D0, D1 U4 & D7, D0, D1 U3 & D7, D0, D1	Any one of Any one of

Figure 5.11: Allowed vertical bus configurations (Figure 2 of 4).

Vertical Bus Configurations	
Active Internal Controls	Bus Configuration
Co = 0 Any pair: D6 & U7, U0 or U1 U6 & D7, D0 or D1	Any one of
Co = 0 Any pair: U6 & D5, D4 or D3 D6 & U5, U4 or U3	Any one of
Co = 0 Any pair: D2 & U7, U0 or U1 U2 & D7, D0 or D1	Any one of

Figure 5.12: Allowed vertical bus configurations (Figure 3 of 4).

Vertical Bus Configurations	
Active Internal Controls	**Bus Configuration**
Co = 0 Any pair: U2 & D3, D4 or D5 D2 & U3, U4 or U5	any one of
Co = 0 Any pair: U6 & D2 U2 & D6	
Co = 0 Any pair: U7 & D0 or D1 U0 & D7 or D1 U1 & D7 or D0	Pair of
Co = 0 Any pair: U5 & D4 or D3 U4 & D5 or D3 U3 & D5 or D4	Pair of

Figure 5.13: Allowed vertical bus configurations (Figure 4 of 4).

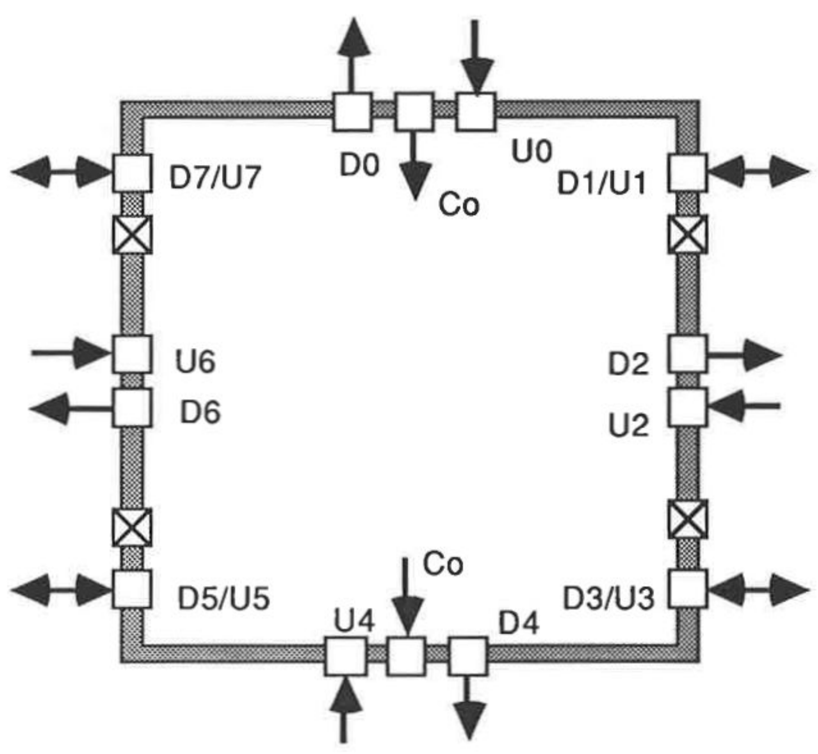


Figure 5.14: Internal control signals.

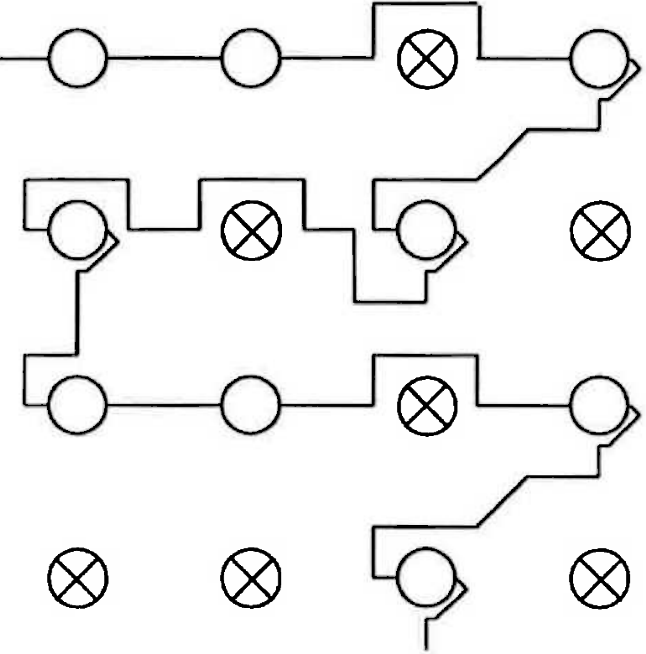


Figure 5.15: Example of a configured linear array.

Thus at the beginning of the search the acceptance level will be set from experience gained with similar fault patterns. This level will be adjusted as the search proceeds to reflect the ease or difficulty with which the rules are finding valid constructions of appropriate size.

With this pruned depth first search strategy it should be possible to find "good" faulty array solutions without resorting to excessive search times or simple artifices.

5.3.1 Global Reconfiguration in Response to Faults for 1D Arrays

The first architecture to be implemented takes the form of a linear array. An example of a reconfigured one dimensional array embedded in this redundancy scheme is given in Figure 5.15. It is not claimed that this scheme provides the best means for configuring one dimensional arrays however it does display some merit. A linear array was chosen for the initial implementation due to the immediate availability of a verified PE mask design.

With reference to Figure 5.15 the rules used for constraining the search space to find a reasonable solution can be readily visualized:

1. To minimize inter-PE delays and thus keep the speed to a reasonable level only one bypass is allowed between used PEs.

2. The order in which possible successive PEs are chosen are determined by the following preferences:

   - Use a left to right succession of PEs whenever possible.
   - Choose a PE closest to the origin PE within the above constraints.

Using these rules a prioritized search tree can be generated that is best searched in the depth first pattern described previously.

5.3.2 Global Reconfiguration in Response to Faults - Mesh Arrays

Reconfiguration of mesh connected arrays in the presence of faults incurs an extra constraint that makes the procedure more difficult. Because there are more ways of assigning PEs to logical positions in the mesh the unconstrained search space is much larger than it is for linear arrays.

The rules to be used to constrain the search space are similar in nature to those used for linear arrays. Figure 5.16 gives an example of an array reconfigured with the following rules to constrain the search:

1. The maximum bypass, both vertical and horizontal, is one PE for low failure rates, increasing for higher rates.

2. The order of choosing successive PEs to fill the search tree's branches is determined through a measure of minimal distance to the origin PE and the PE's parents, within the constraints of rule 1 above.

3. As the search tree is descended the array is grown from the starting point in a fashion so as to retain its two dimensional structure.

5.3.2.1 Scheme Utilization

Figure 5.17 shows the utilization (fraction of functional cells able to be configured into a mesh array) vs % PEs faulty, illustrating the good performance of the frame for low yield cells.

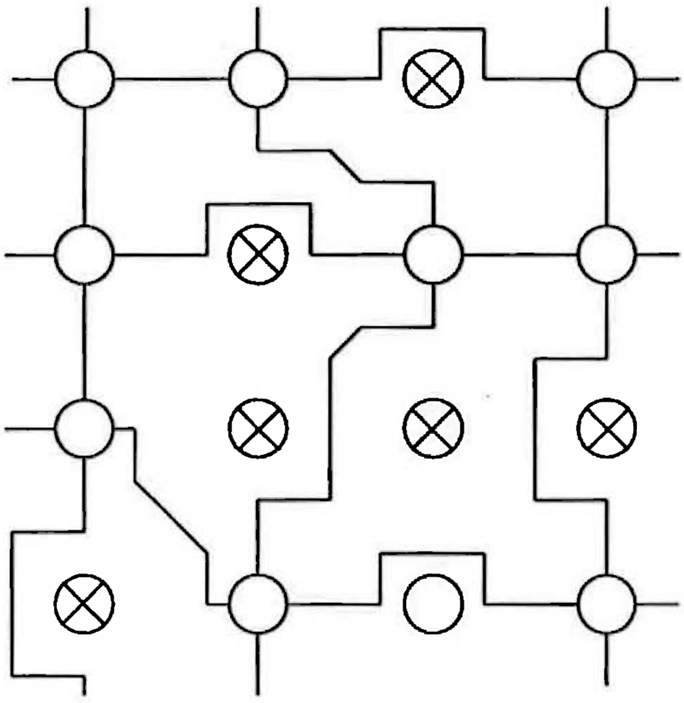


Figure 5.16: Example of a configured mesh connected array.

5.4 Throughput Rate Optimization

The segmentation and access switches of the frame are implemented as simple transmission gates. In the ON state, they insert a significant series resistance into the data paths. Although that resistance can be decreased using large width-to-length ratios, this increases the switch capacitance and increases the area required for each switch. Between switches, significant line capacitance (and, for narrow lines, significant line resistance) are inserted into the data paths.

Switch and line resistance and capacitance then introduce significant RC delays along data paths, whereas the ideal WSI reference would avoid reconfiguration paths and switches (introducing only the line resistance and capacitance).

To minimize such RC delays, the three schemes for addition of distributed drivers shown in Figure 5.18 were considered. The delay performance was evaluated using (1) ADVICE (an AT&T Bell Laboratories version of SPICE), (2) a simple lumped RC delay model using a switched resistance gate model [Carter & Guise, 1983] and (3) an "complex" RC model with a more accurate gate delay model [Horowitz, 1983].

Using the same model parameters, calculations using the RC models were within 10–20% of the ADVICE simulation results. The delay evaluations were performed

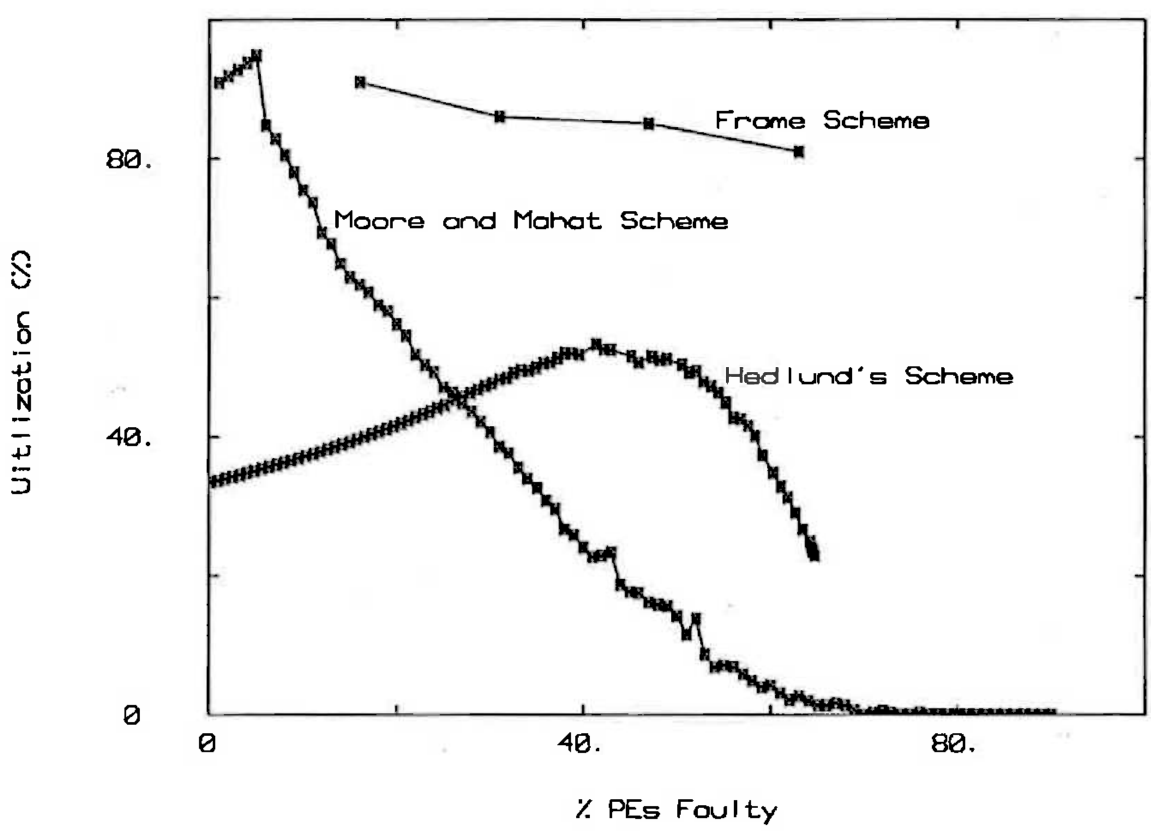
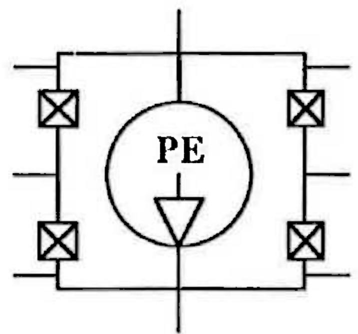
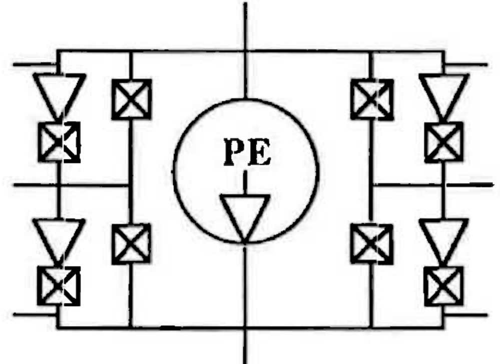


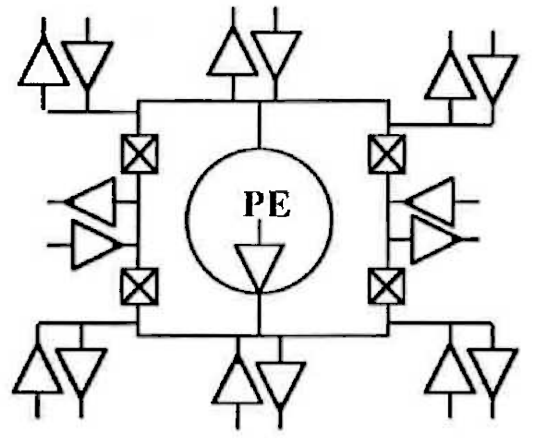
Figure 5.17: Utilization vs. Cell Fault Probability for Frame and Other Schemes Above.



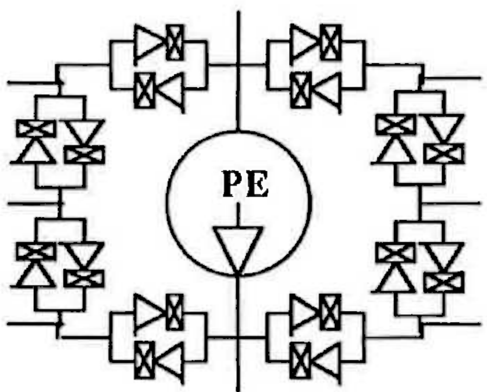
(a) Internal Buffers Only



(b) Buffers within Ring Bus



(c) Inter PE Buffers



(d) Fully Buffered Ring Bus

Figure 5.18: Unbuffered and Various Buffered Frame Designs.

while varying the driver sizes (channel width/length ratio) and the transmission gate sizes (again channel width/length ratio). These analyses predict that the driver distribution shown in Figure 5.18c, with drivers placed on paths between frames provides the minimum delays. For the 2.5μm CMOS technology and model parameters, minimum delay was obtained for driver width/length = 20 and pass transistor width/length = 15. This choice was also found to be area-delay optimal as well as having a reasonably low power dissipation.

With this choice of distribution of drivers and array, throughput rate should be about 40 MHz, about half the internal throughput rate of the FIR filter sections.

Details of this study will now be presented.

5.4.1 RC Model

An example of a length of reconfigured interconnect and its equivalent RC model is given in Figure 5.19. If metal lines are used exclusively then the metal line resistance, *Rmlm*, and capacitance, *Cmlm*, can be treated as lumped components instead of as distributed components, without much loss in accuracy. In fact the line resistance is so small, when compared with other resistances in the line, that it can be ignored altogether. The delay time, defined as the time required for the output voltage to reach 60% of its final value can be given by [Carter & Guise, 1983] (referring to Figure 5.19):

$$\tau_d = R_1(C_1 + C_2 + C_3) + \sum_{i=1}^{3} \sum_{j=i}^{3} R_i C_j \quad (5.1)$$

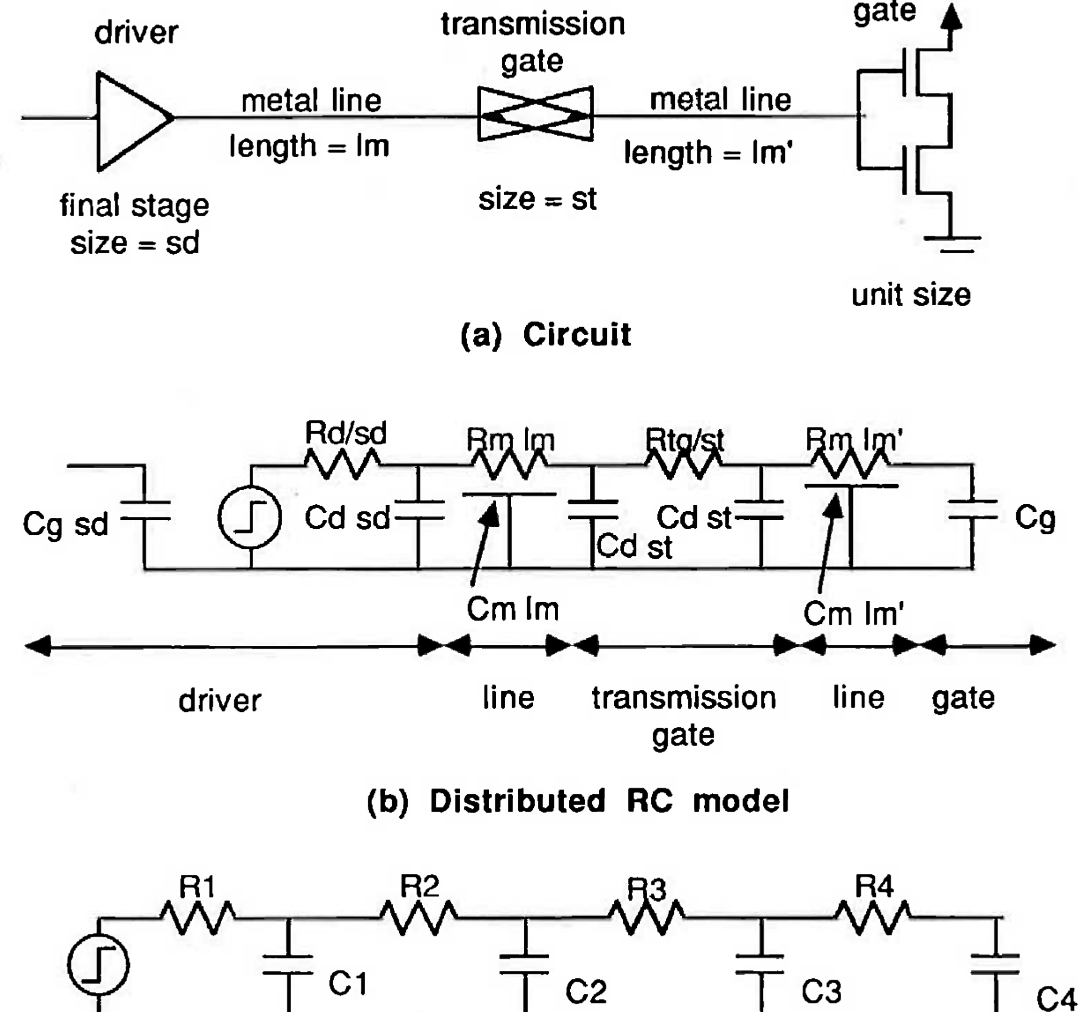
It has been suggested however that this simple RC model, though useful for modeling interconnect, is inadequate for active circuitry [Horowitz, 1983]. Horowitz suggested a more suitable model for describing the driving component delay:

$$\tau_d = \sqrt{(\tau_r ln V_s)^2 + 2(1 - V_s)\tau_{in}\tau_{gm}} \quad (5.2)$$

where (referring to Figure 5.20):

$$\tau_r = R_{gate} C_{load}$$
$$\tau_{gm} = G_{m_{gate}}^{-1} C_{load}$$
$$\tau_{in} = R_{prev} C_{prev}$$
$$V_s = normalized\ switching\ voltage\ (normally\ 0.5)$$

τ_d represents the time required the gate to switch from an input of V_s to an output of $1 - V_s$.



(c) Lumped RC approximation

(in practice Rm lm & Rm lm' are negligible)

Figure 5.19: A length of reconfigurable interconnect and its equivalent RC model.

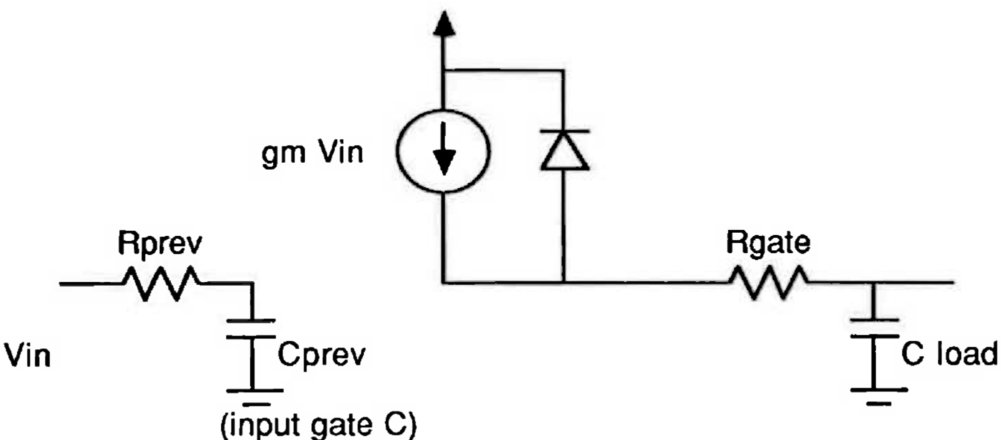


Figure 5.20: Modified circuit model for a gate with a rising input.

These two models - the "simple" RC model, and the "complex" RC model with modified driver delay will be used to investigate different driver distribution options.

5.4.2 WSI Driver Proposals

The different WSI driver proposals were given in Figure 5.18. In order to achieve a good balance between speed and fault tolerance it was decided that the maximum bypass allowed would be around one processing element only. Thus, by comparing different bypass cases, the worst case delay can be shown to be the case illustrated in Figure 5.21(a).

The details of the RC delay equations for each of the four driver arrangements are given in Appendix A.

These four arrangements were evaluated, using both RC models, for a number of values of driver size, *sd*, and transmission gate size, *st*, and the results are plotted in Figures 5.24 through to 5.31 given at the end of this chapter. Some care should be taken when comparing these figures as the scales differ on some. The frequencies were determined simply as $1/\tau_d$ and thus represent upper values only. A step at $sd = 32$ may be noticed in several of the plots. This arises from a decision that the driver size would increase from $stages = 2$ to $stages = 3$ when the final stage reached the size, $sd = 32$. The application of these models with this rule suggests that two stages should be retained for some $sd > 32$. The ratio between successive stages was fixed in all cases to 4 : 1 on the basis of the analysis in [Lewis, 1984].

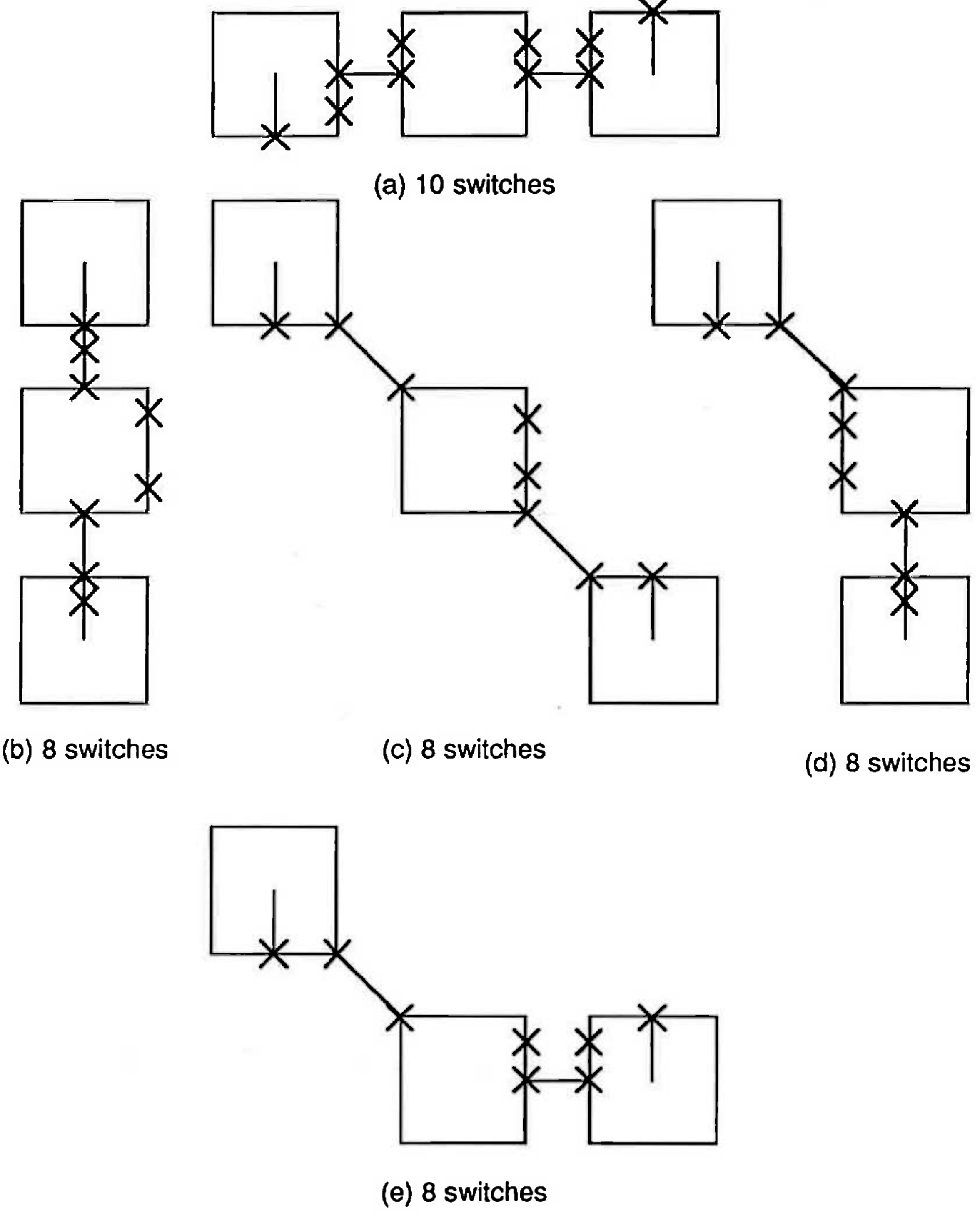


Figure 5.21: Derivation of worst case delay (Case (a)) for 1 PE bypass.

sd	st	freq: Case 1	freq: Case 2	freq: Case 3	freq: Case 4
20	10			39.9	26.20
20	15			41.6	
20	20	30.73	40.50	41.43	24.50
20	30				22.28
20	40	28.40	35.89	35.56	20.27
20	60	26.50	30.38	30.00	
20	80	23.30	25.99	25.70	
40	20	35.82	39.14	39.15	
40	40	37.72	38.64	39.00	
40	60	36.40	36.04	34.99	
40	80	34.36	33.11	32.38	
60	20	37.61	42.52	43.47	
60	40	41.06	43.86	44.00	
60	60	40.68	42.07	42.00	
60	80	39.37	39.86	38.97	
80	20	38.15	43.29	44.00	
80	40	42.32	45.76	44.88	
80	60	42.63	44.68	43.42	
80	80	41.77	42.86	41.72	

Table 5.1: ADVICE Results: Driver and transmission gate size vs. frequency (MHz)

5.4.2.1 ADVICE Results

A large number of points were also determined using ADVICE simulations. For consistency τ_d was defined as the delay between the input reaching 60% of its final value and the output reaching 60% of its final value. The results are plotted, along with interpolated points for easy viewing, in Figures 5.32 to 5.34 given at the end of this chapter. The determined values are also given in Table 5.1. Again the frequency was taken as $1/\tau_d$. The step previously observed at $sd = 32$ is evident in some of these plots though it is not as prominent, partially due to the fewer points actually determined. However it verifies the RC model result in a general manner.

5.4.3 Discussion of Modeling Results

Similar trends were discovered with the application of each of the three modeling approaches:

1. Speed increases monotonically with driver size.

2. For any particular driver size an optimum transmission gate size exists.
3. This optimum transmission gate size is about the same for each modeling approach.
4. A better choice could be made on the driver size at which to change from two to three stages. As a two stage driver was the final choice closer investigation of this was not necessary.
5. The general shapes of the RC model and ADVICE model surfaces obtained bear more than a superficial resemblance in each case.

The results predicted by the three models can disagree by up to 20%, with the results obtained from the complex RC model generally being slightly more consistent with the ADVICE results than those of the simple RC model.

One problem with the RC models was the difficulty experienced in determining appropriate resistance values. The capacitance values can be obtained simply from the ADVICE parameters but the resistance parameters were obtained by measuring the on-resistance of the transmission gates in an ADVICE simulation. This resistance value is of course an approximation as transistors are non-linear devices.

In order to obtain a better set of resistance parameters an attempt was made to fit the simple RC model to the ADVICE results. This attempt was abandoned when the resulting fitted resistances came out to be negative! Additional calculations were made using different resistance values again with no significant changes in the relative results.

Overall however, for broad brush calculations, it can still be hypothesised that an RC modeling approach is satisfactory as it indicated the trends fairly accurately in this case.

5.4.3.1 Choice of Approach and Area-Time Optimality

The conclusions that can be drawn from both the ADVICE simulations and the RC model calculations are fairly similar. Case 4, the fully buffered ring case, provides the worst performance of the four suggested driver distributions. For the same driver and transmission gate size, case 3 (inter-PE buffers) gives the best result but only by a small margin.

However each design has different numbers of drivers and transmission gates so a more suitable basis for choosing the best design may be to consider the area-delay product.

If the area of a size=20 driver or transmission gate is taken as 20 units then the area requirements of Cases 1 to 3 are:

Case 1: 2sd + 28st. One full size driver is placed at the two outputs of the PE and each transmission gate in the ring is full size.

Case 2: 10sd + 36st. There are eight full size drivers in the ring bus and two full size PE output drivers. All 36 transmission gates are full size.

Case 3: 8sd + 28st. Each PE has eight drivers around each edge of its own (the other eight belong to neighbouring PEs) and 28 full size transmission gates. The driver input transmission gates on each side are not counted as full size and neither are the output drivers of the PE itself.

Using these figures as the area and the ADVICE delay results, area-delay products were calculated and are given in Table 5.2. Case 3, $sd = 20$, $st = 15$ provides a good choice both on the basis of absolute speed and total area. However if the driver size, *sd*, was increased further in Case 1, the same results can be achieved with a big driver and small transmission gates. Nevertheless the Case 3 result was the one used in the actual design.

5.4.3.2 Power Dissipation

As well as area, another "cost factor" to be considered is the extra power required by these large drivers. This will be determined, for comparison purposes, by calculating the power needed to switch the total capacitance in any one line at the required speed. In doing this resistive losses are ignored. Though significant, the resistive losses are not as large as the switching power requirements and thus can be ignored for present purposes. Thus the power can be calculated from the product CV^2f where C is the total capacitance in a single bypass line from original source to final destination. Assuming $f = 40MHz$ and using the capacitance values for a choice of *sd* and *st* that would result in this speed (except for case 4 which cannot in actual fact reach 40MHz) we obtain:

Case 1: Power = 21.6 mW per line

Case 2: Power = 34.1 mW per line

Case 3: Power = 24.9 mW per line

Case 4: Power = 17.25 mW per line

Case 3 arises as a reasonable choice.

sd	st	AT: Case 1	AT: Case 2	AT: Case 3
20	10			11
20	15			14
20	20	20	22	17
20	40	41	46	36
20	60	65	78	61
20	80	98	118	93
40	20	18	28	22
40	40	32	48	37
40	60	48	71	57
40	80	67	99	79
60	20	18	31	24
60	40	30	46	36
60	60	44	66	51
60	80	60	87	70
80	20	16	35	27
80	40	80	49	39
80	60	43	66	53
80	80	57	69	69

Table 5.2: Area × Delay products for the first three cases

5.4.4 Conclusions

In conclusion:

1. Case 3 was the best choice as far as speed and area-delay and almost the best choice as far as minimizing power consumption.
2. The conclusions drawn from the RC model results were the same as those drawn with the ADVICE model. The RC model provided an adequate description for comparison purposes.
3. Adding complexity to the RC model did not significantly impact the accuracy of the RC model results.

Though Case 3 was the actual optimal result for speed and area-delay, Case 1 was good enough, with large enough source drivers, to warrant consideration. It is estimated that Case 1 might well be the optimal choice for smaller processing element sizes.

5.5 Fault Detection and Isolation

The problem of fault detection/isolation of reconfigurable WSI circuits is simplified by separately testing the array cells and the cell interconnection paths. Once the cell interconnections are verified, testing of the individual cells of the array is relatively straightforward. However, given the large number of configuration states of the chip frame, verification of the correct functioning of all states of all frames (with some states requiring that data pass through the enclosed cell) becomes quite difficult. Testing the array solely from a limited number of connections to the outer edge of the array will generally be difficult.

The most straightforward way to overcome this problem is to provide a means of observation internal to the array. In the chip frame a set of probe pads have been provided between the PE and the frame itself. A control state that disconnects the frame from the cell allows each cell to be tested. These pads can also be used to test inter-PE paths, either to other pads or to the array edge. Initial testing of the array would then involve the following general sequence of test steps:

1. Power-up the array under gradually increasing current limited power supply conditions to determine whether there are any power/ground short conditions (and to perhaps localize and repair such shorts).

2. Using a probe card and the cell test mode (i.e. cell disconnected from frame), test each array cell to locate faulty cells.

3. Given the set of faulty cells, determine and test the desired reconfiguration state of each frame.

   This can be done in one of two ways. The required paths can be explicitly tested by using the probe pads at each end. Alternatively, and perhaps more practically, the peripheral pads at the array edges can be probed to test connected paths through the internals of the array. As the PEs have been previously tested, and are arithmetic in nature, they can be set up to pass data through unchanged and thus test the interconnect.

4. If the desired state is found to be faulty, then select and test a new reconfiguration state.

The addition of internal test sites and a control state to isolate cell inputs from the frame provides a convenient approach to initial testing of the array.

5.6 Implementation

The monolithic WSI array using reconfiguration chip frames and the HWSI array of reconfiguration frames (onto which unpackaged FIR filter section IC's are flip-chip mounted) were designed using the MULGA VLSI CAD tools [Weste, 1981]. Both schemes are implemented on a common wafer (*ie.* in the style of a multi-project wafer), the monolithic WSI array occupying one half of the wafer and the HWSI array occupying the other half. The layout of one monolithic WSI array site is shown in Figure 5.22. Each reconfigurable site is relatively large (about 11mm by 11 mm) with considerable white space. The wasted white space originated from (1) the wide chip frame required to achieve 30-bit wide data paths along with associated drivers and transmission gates, and (2) limitations imposed by the symbolic CAD design tools. Use of the conventional VLSI IC tools in the WSI project described here uncovered a number of limitations, several of which were resolved with the help of the MULGA designers. Others, however, were more severe.

In the MULGA design environment cells are specified symbolically, compacted to form mask layout and then routed to the pads using an automatic channel router. The design presented here was too large to compact as one super module, so pieces of the frame had to compacted separately and then connected to each other and the

PE, or flip mount, using the channel router. These frames then had to be routed to each other and the PE, or flip mount, using a channel routing step. Compaction before connection prevented the layout from being constructed so that pieces could simply abut each other. The channel router was also limited because it was designed for the job of routing to a chip pad frame only. Together these limitations resulted in considerable wasted space in the final layout. Many iterations had to be performed on the design to attempt to reduce this wasted space as much as possible.

After construction of the half wafers there were further problems in the extraction and simulation of the design. The construction of the extractor meant that only part of the design could be extracted. Only switch level simulation could be performed on this extracted netlist. These problems apply as much to large chips as they do to WSI devices.

Though the ease of symbolic design was appreciated in the earlier stages, a mask level design tool, or a compaction tool that allowed the spacial specification of some mask features, would have been useful. As discovered (and quickly supplied) the computer being used to run the CAD tools should have as much primary memory and swap space as possible. Almost a Gigabyte of disk space was required to store the intermediate files required. Parallel simulation tools are essential for the verification of such large structures [Ackland et al., 1986]. Unfortunately the extractor available at the time prevented successful simulation on such a parallel simulator.

A microphotograph of the complete wafers as manufactured is given in Figure 5.23. As mentioned above, the empty space inside the pads arises due to limitations of the channel router used. The chips around the edge are test sites or the FIR filter chips required for flip mounting on the HWSI wafers. Leaving blank silicon space on a wafer can create fabrication problems and had to be avoided.

5.7 Summary

The "frame" scheme provides a different approach to the reconfiguration of mesh arrays with a significant number of faulty PEs. The design and implementation of the scheme has been described. In the next chapter it will be compared with a number of other approaches to the reconfiguration of mesh arrays. There it will be shown that the frame scheme performs well when compared with other schemes that achieve a similar level of utilization.

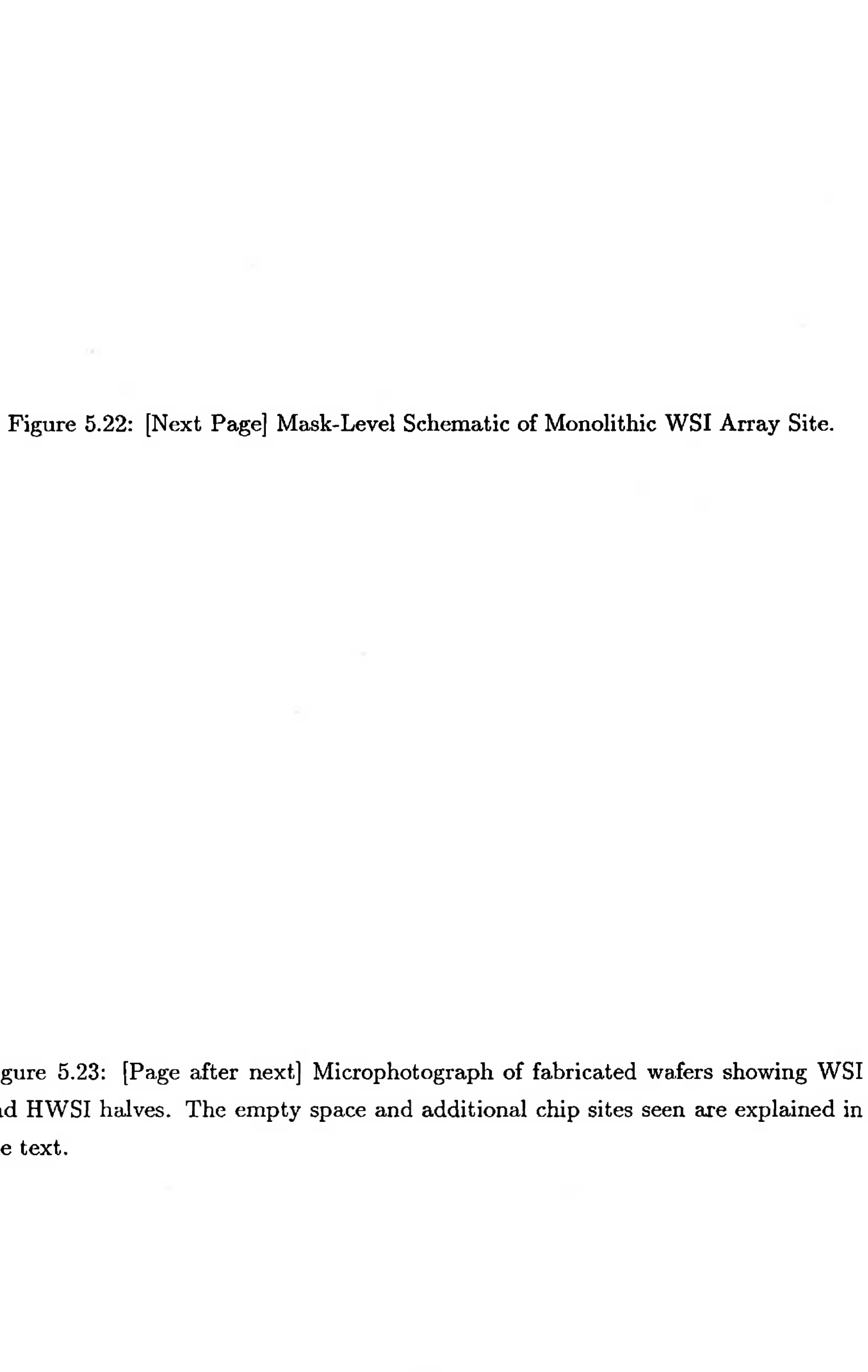
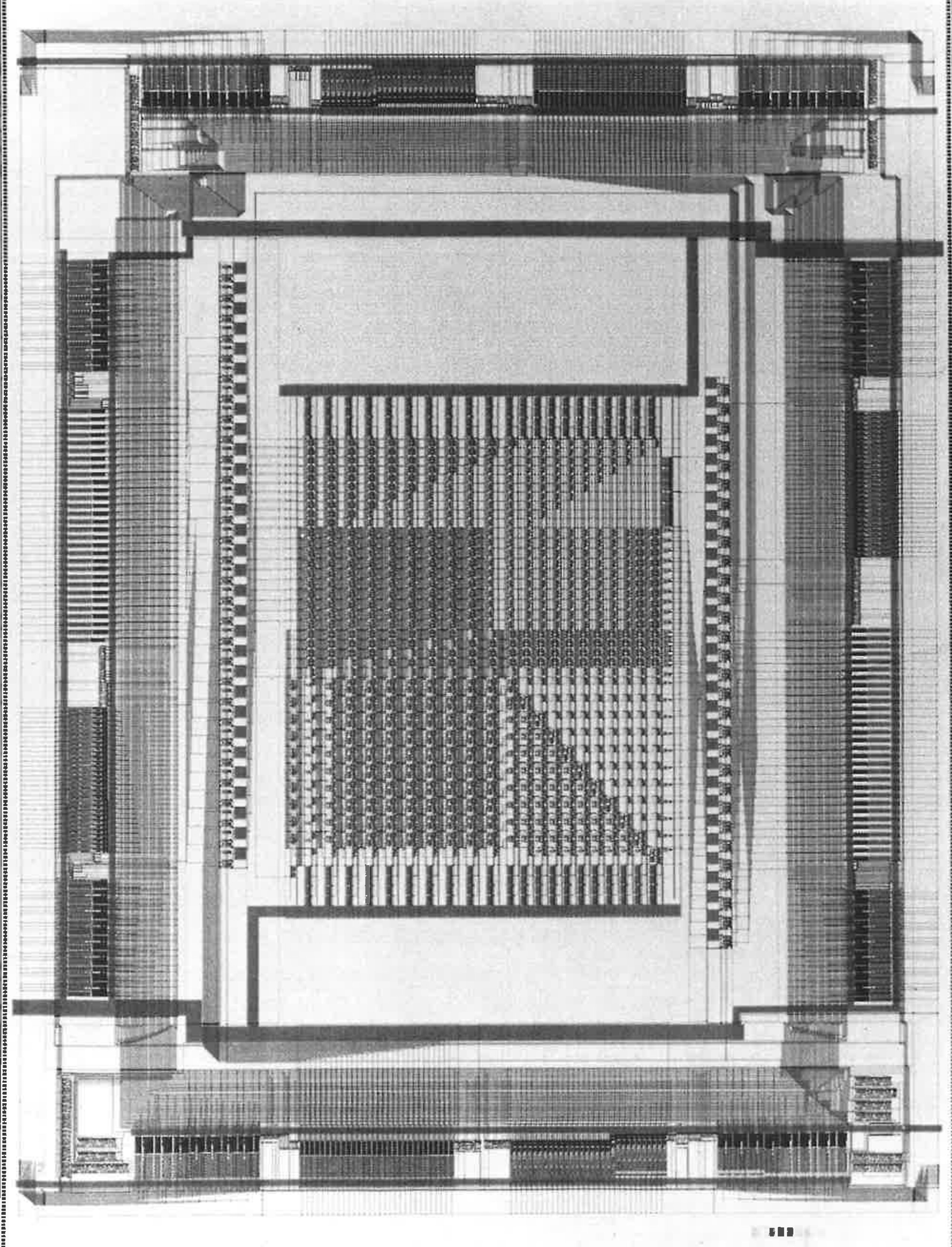
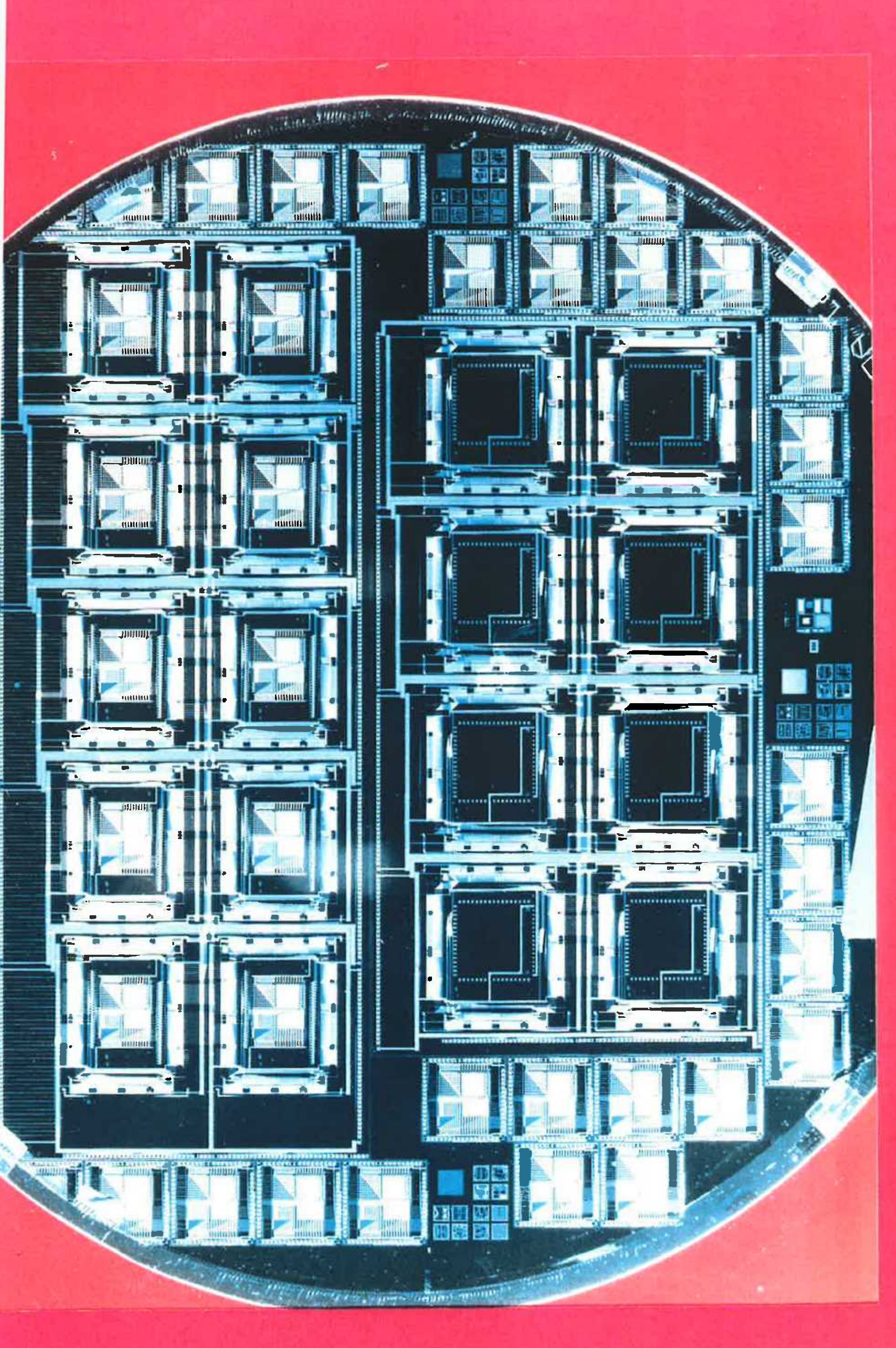


Figure 5.22: [Next Page] Mask-Level Schematic of Monolithic WSI Array Site.

Figure 5.23: [Page after next] Microphotograph of fabricated wafers showing WSI and HWSI halves. The empty space and additional chip sites seen are explained in the text.





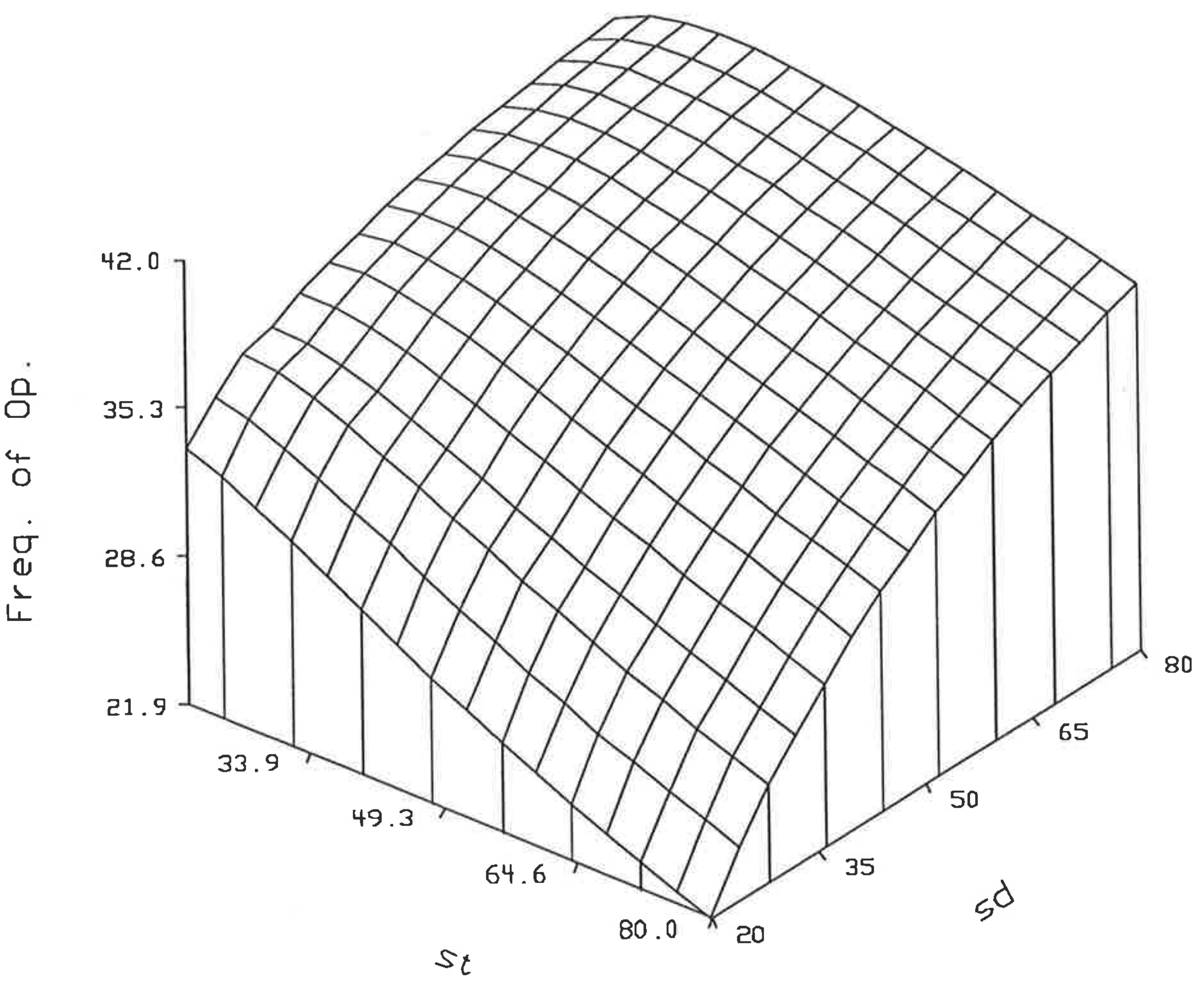


Figure 5.24: Speed vs. driver and transmission gate size. Case (1) Non buffered. Simple RC model.

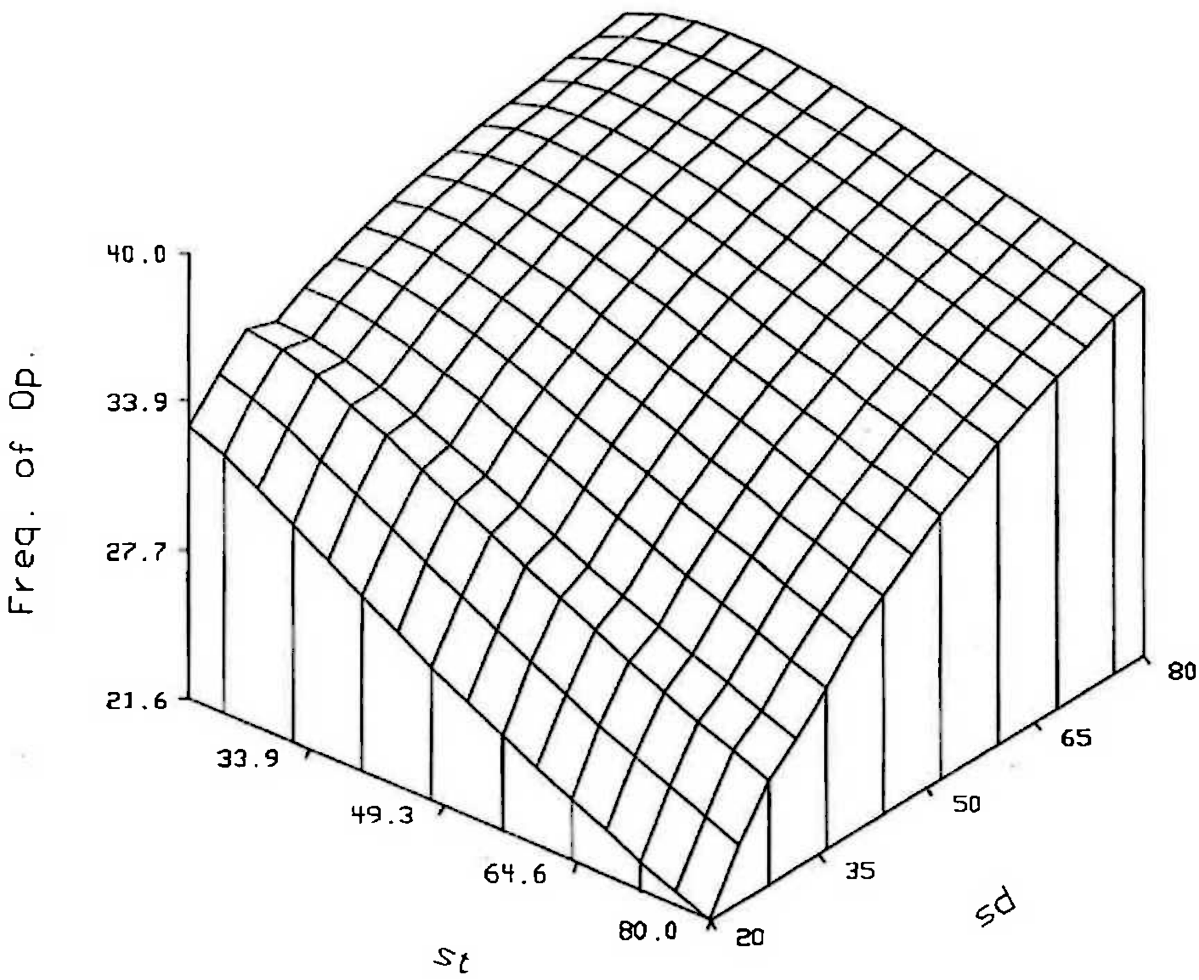


Figure 5.25: Speed vs. driver and transmission gate size. Case (1) Non buffered. Complex RC model.

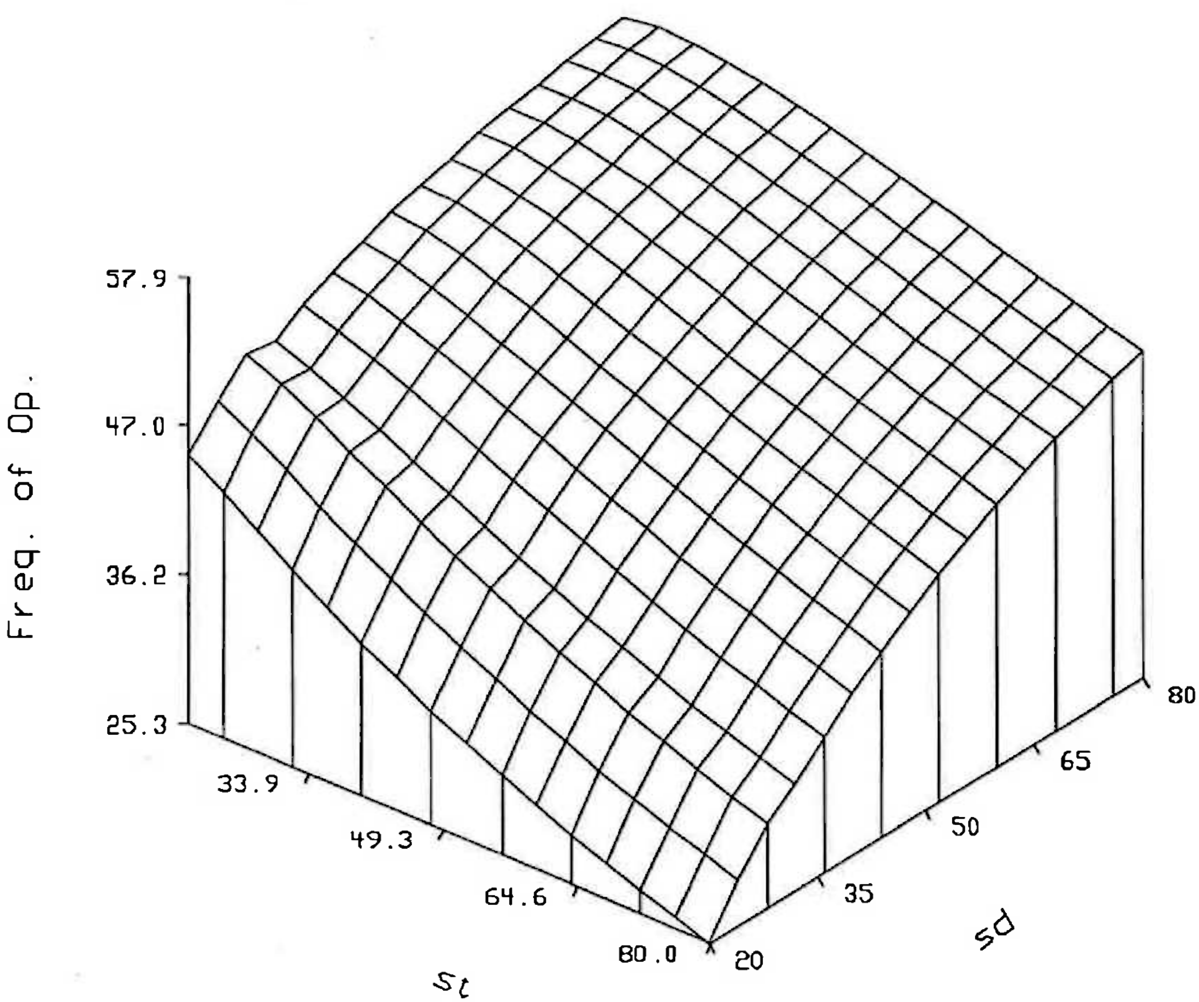


Figure 5.26: Speed vs. driver and transmission gate size. Case (2) Bypass Drivers. Simple RC model.

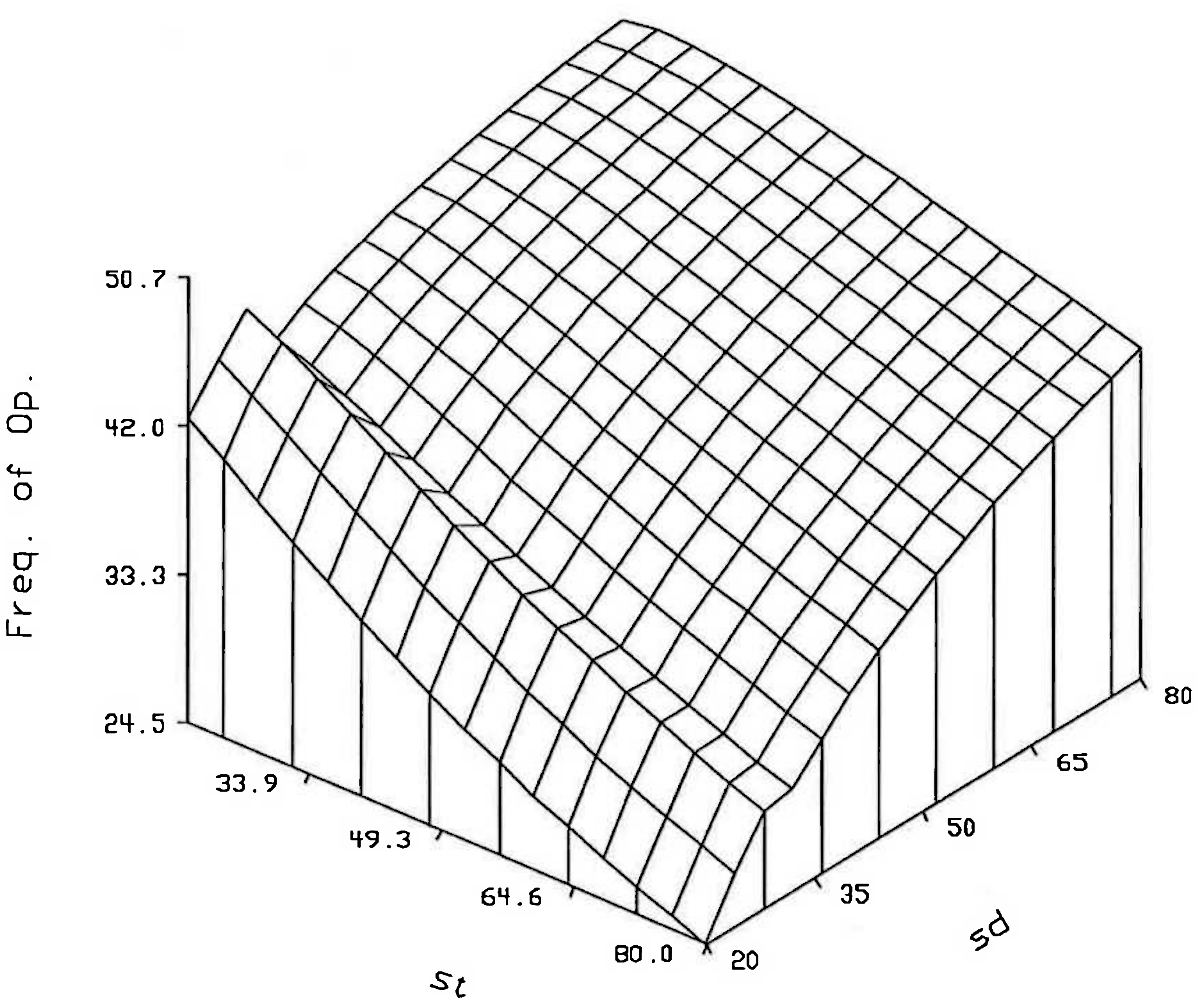


Figure 5.27: Speed vs. driver and transmission gate size. Case (2) Bypass drivers. Complex RC model.

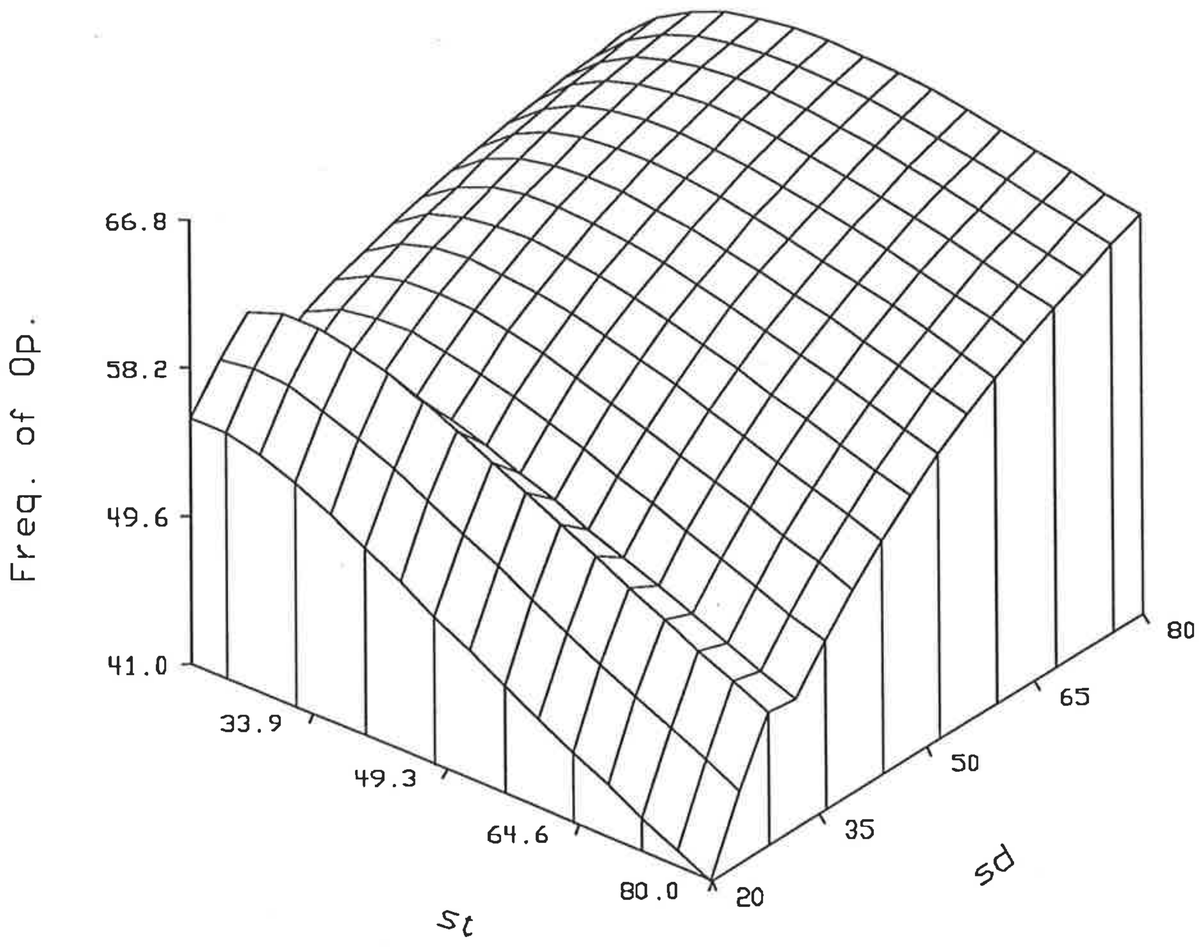


Figure 5.28: Speed vs. driver and transmission gate size. Case (3) Inter-PE buffers. Simple RC model.

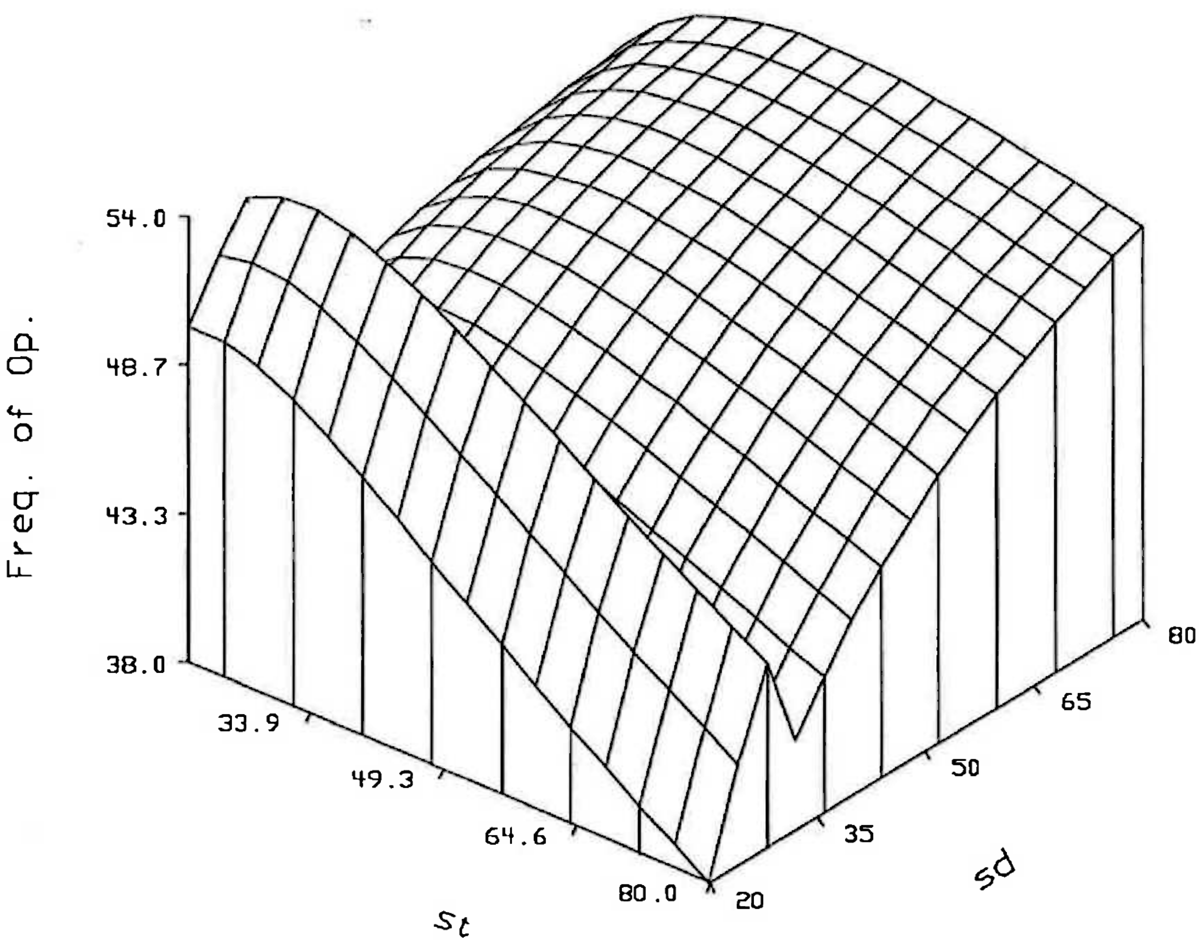


Figure 5.29: Speed vs. driver and transmission gate size. Case (3) Inter-PE buffers. Complex RC model.

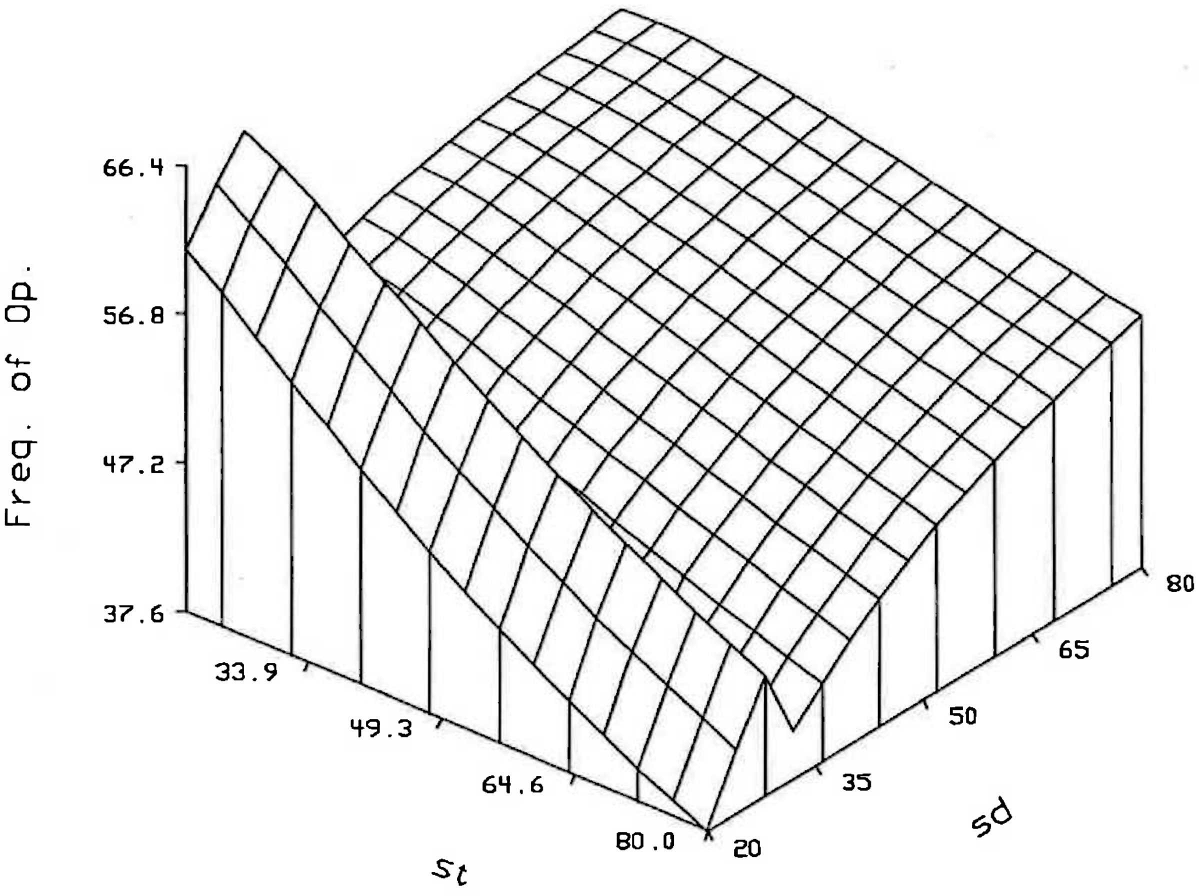


Figure 5.30: Speed vs. driver and transmission gate size. Case (4) Fully buffered. Simple RC model.

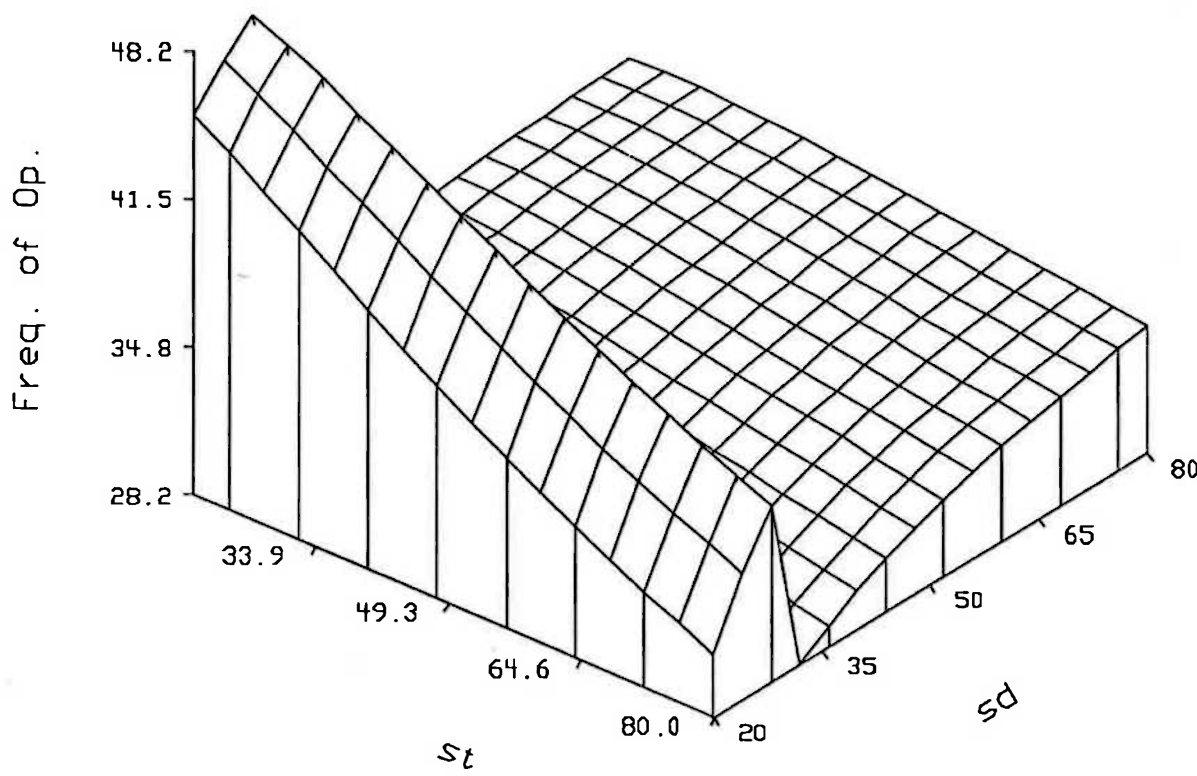


Figure 5.31: Speed vs. driver and transmission gate size. Case (4) Fully buffered. Complex RC model.

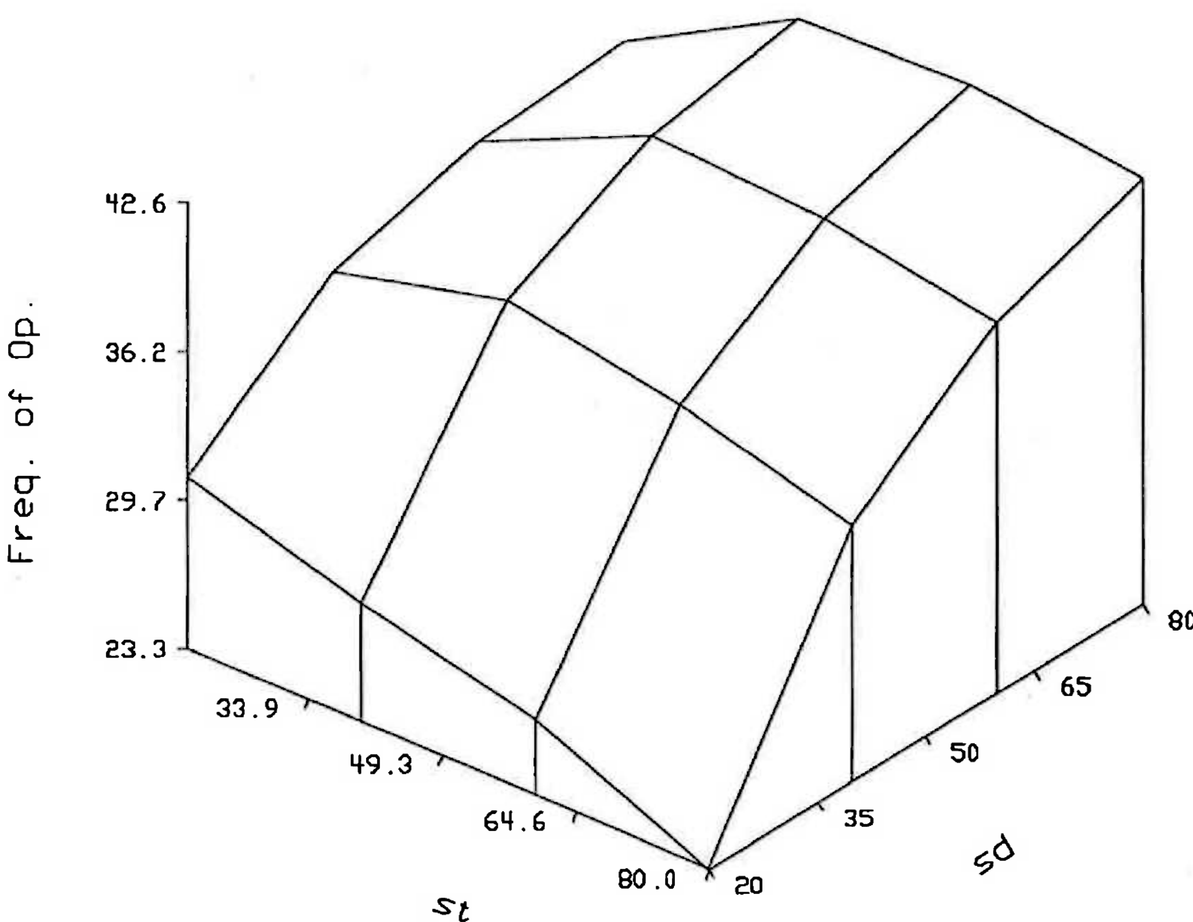


Figure 5.32: Speed vs. driver and transmission gate size. Case (1) Non-buffered. ADVICE results.

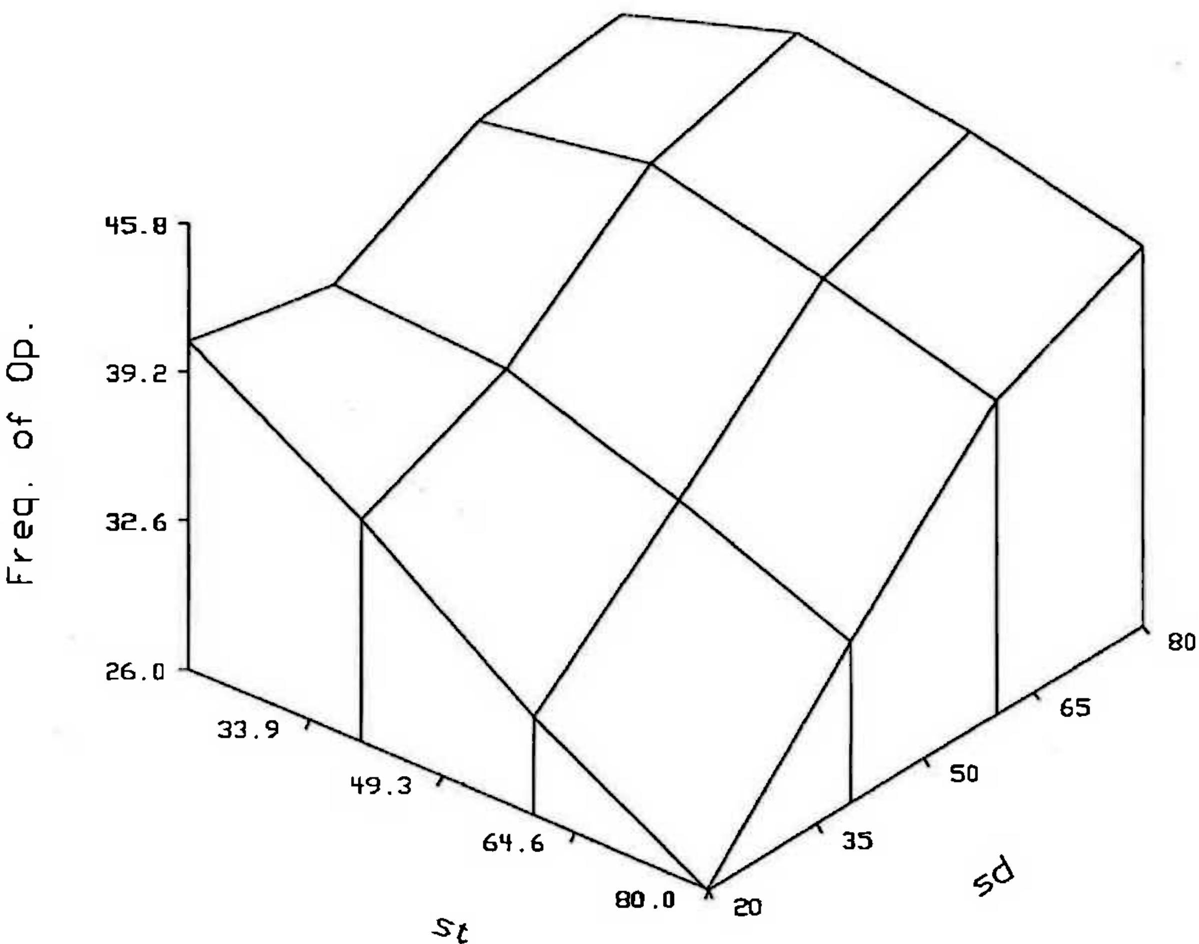


Figure 5.33: Speed vs. driver and transmission gate size. Case (2) Bypass drivers. ADVICE results.

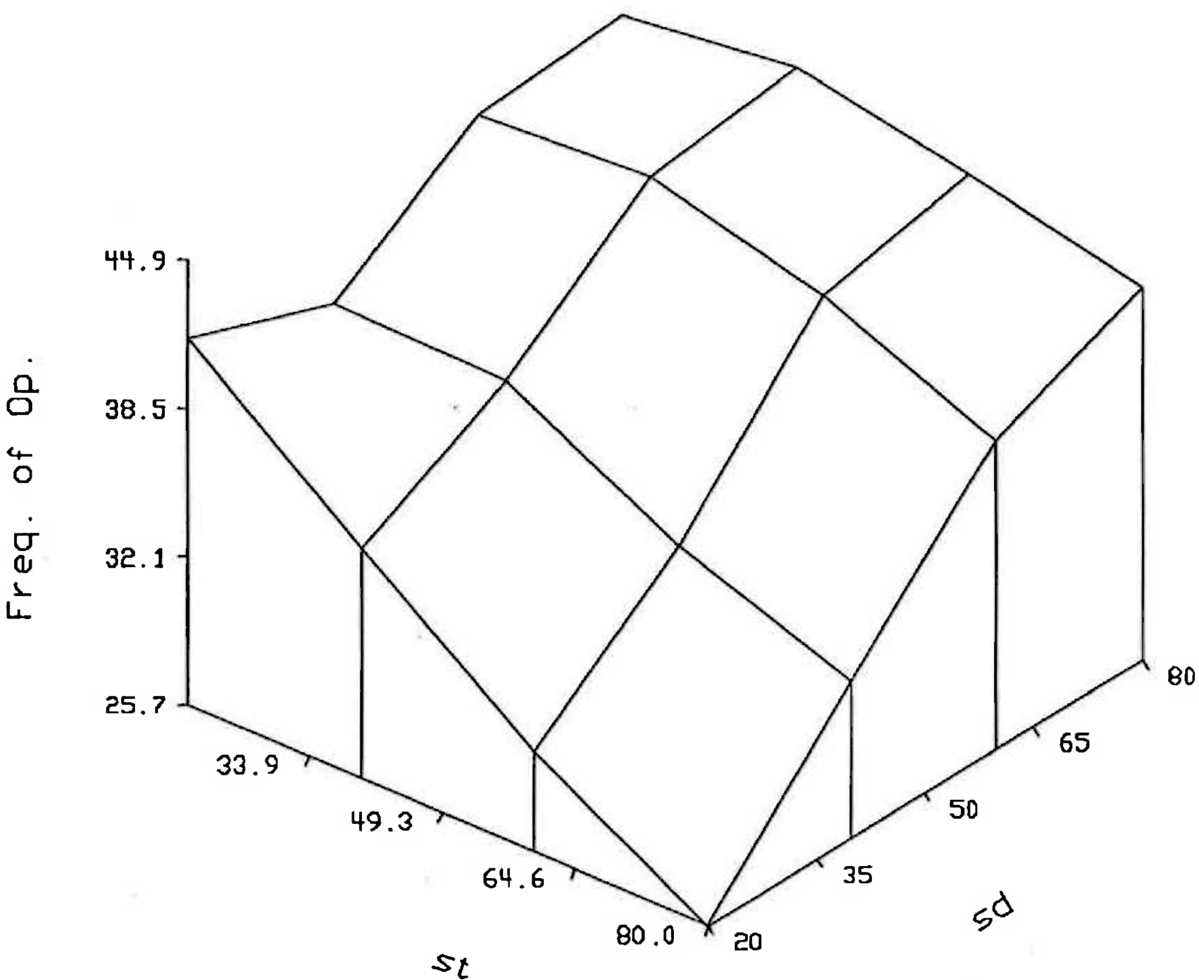


Figure 5.34: Speed vs. driver and transmission gate size. Case (3) Inter-PE buffers. ADVICE results.

Chapter 6

Comparing Redundancy Approaches for Mesh Arrays

6.1 Comparison Metrics

In Section 4.4 it was stated that when implementing an architecture requiring defect tolerance, the choice should be made so as to maximize either the

$$\frac{\text{Yield} \times \text{Area}_{PE} \times \text{Speed}}{\text{Area}_{total}} \tag{6.1}$$

product or the

$$\frac{E(P) \times \text{Area}_{PE} \times \text{Speed}}{\text{Area}_{total}} \tag{6.2}$$

product, depending on the application. (Area_{PE} is the functional area of each PE, Area_{total} is the total area of the array, including interconnect.) For applications where the product requires a fixed number of processing elements the Yield based expression is more appropriate. This would usually be the situation for large scale chips. For many WSI applications the $E(P)$ based expression would be more appropriate. It would often make economic sense for a wafer scale product to allow for the production of parts with different numbers of processing elements. This is simply because the wafers generally have very different numbers of defects.

As discussed in Chapter 3, to use defect tolerant techniques profitably for large chips or for WSI arrays, a substantial area of the device must be covered by an array containing a single type of processing element.

The PE size has a significant effect on the Yield × Area or $E(P)$ × Area component. The influence of PE size on either of these expressions is a tradeoff between the following factors:

1. The minimum PE size required for performing the set task.

   For some tasks this may be a single bit adder, for others a reasonable sized control unit may be required in addition to arithmetic type units and memory.

2. The degree to which functionality per unit area increases with PE size.

   If functionality per unit area increases quickly with PE size then larger PE sizes would be preferable. For products with a small control overhead (usually associated with limited programmability) functionality per unit area does not change much when one goes from single bit PEs to two bit PEs. For tasks where the PE requires a reasonable sized control unit increasing PE size and functionality may increasing the functionality per unit area as the control overhead fraction is decreased.

3. The area overhead required for the reconfiguration scheme.

   Silicon area spent on reconfiguration is area lost to functionality. The area overhead depends on the scheme used, the size of the interconnection switches employed and the number of bits that are required to be communicated. Faults in the wiring area may also lead to a reduced yield.

4. The percentage of the wafer area still usable after defects.

   Other factors aside, this suggests making the PEs as small as possible (so that a single defect requires the removal of the smallest amount of device area.) Consideration of factors 2 and 3 above work against this to suggest an increase in PE area. Factor 1 determines the minimum PE size.

5. The relationship between PE yield and PE utilization.

   $E(P)$ is directly proportional to PE utilization and array yield is also related. For PE yields below about 80% the impact of the reconfiguration scheme on the utilization and $E(P)$ or array yield is quite significant.

6. The effect of wiring on PE utilization.

   This is largely a function of the reconfiguration scheme used as well as the details of the implementation of the power and clock nets. A method for handling this was presented in Chapter 3.

Most of these factors depend to a large degree on the details of the function being implemented, and thus are generalizable only to a limited extent. Determining the

best point of trade-off between all these area factors can be determined by looking at the area utilization measure $\frac{E(P)\text{Area}_{PE}}{\text{Area}_{total}}$ [Koren & Breuer, 1984]. This measure is defined as that percentage of the array that is used for computation and thus reflects the interplay of the factors described above.

The speed of a defect tolerant processor array can be limited either by the speed of the processing element, or the speed of the communications allowed by the interconnect wiring. The last may depend solely on the amount of wiring and the number of switches between the connected PEs, or it may be determined by the nature of the communications implemented, as described in Section 4.8. (*eg.* Multi clock cycle messages or pipelined interconnect.) As the PE size increases it would usually be the case that the PE speed would decrease. Simultaneously to this, the percentage of PEs that are faulty would increase and the length and speed of the interconnect would decrease. The rate of decrease of interconnect speed with increasing PE size is likely to be faster than the rate of decrease of the PE speed. Aside from this little can be said about the speed components of the above metrics without knowing details about the function to be implemented.

6.2 Classification of Mesh Arrays

A taxonomy of interconnect reconfiguration approaches to mesh arrays is given in Figure 6.1. Row bypass schemes are the simplest approach (Figure 6.2) but result in a low utilization. Another simplistic approach is to use good PEs as purely bypass elements, bypassing an entire row and column for each failed PE (Figure 6.3.) This approach results in the lowest PE usage of any scheme and thus would not usually be used for yield enhancement of WSI arrays. Bypassing bad PEs within a row and then steering the columns around the bad PEs, as illustrated in Figure 6.4, achieves reasonable utilizations without too much overhead. Two approaches are illustrated. With LI (Local Interconnect – point to point connections only) connected columns each PE can only be vertically connected to the three PEs below it. With GI (Global Interconnect – switched connections) connected columns extra bypass is provided to enable connection to any PE in the row below. An example of such an approach using LI columns was presented by Evans [Evans, 1985]. Evans also describes a distributed algorithm that can be used to reconfigure the array. A similar approach has also been used independently by Marwood and Clarke to produce a field reconfigurable array [Marwood & Clarke, 1985] and at McGill University [Pateras & Rajski, 1988]. A hierarchical test strategy for this array was presented in [Evans & McWhirter, 1986]. The utilization can be increased

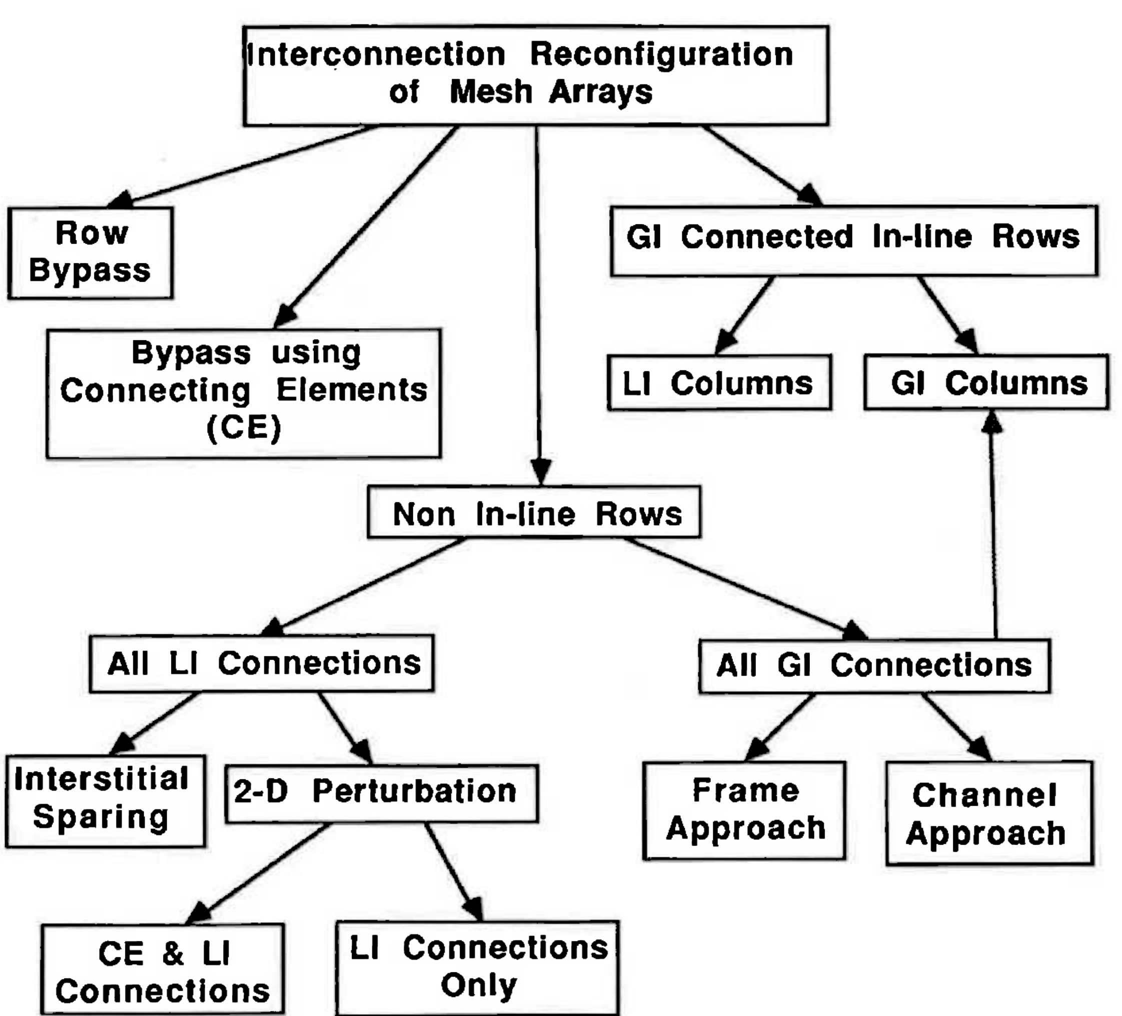


Figure 6.1: Taxonomy of approaches to interconnect reconfiguration of mesh arrays.

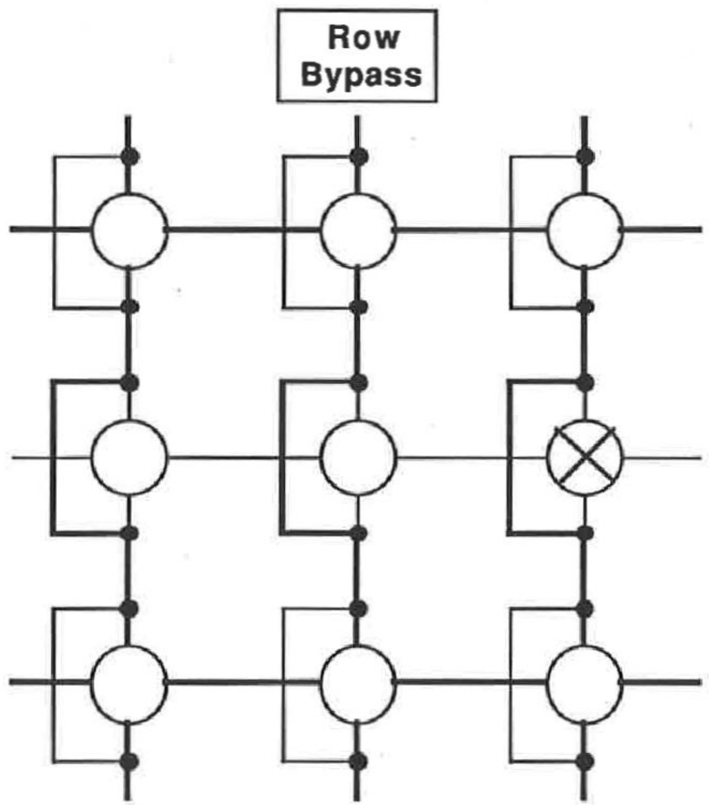


Figure 6.2: Example of a reconfiguration scheme using row bypass.

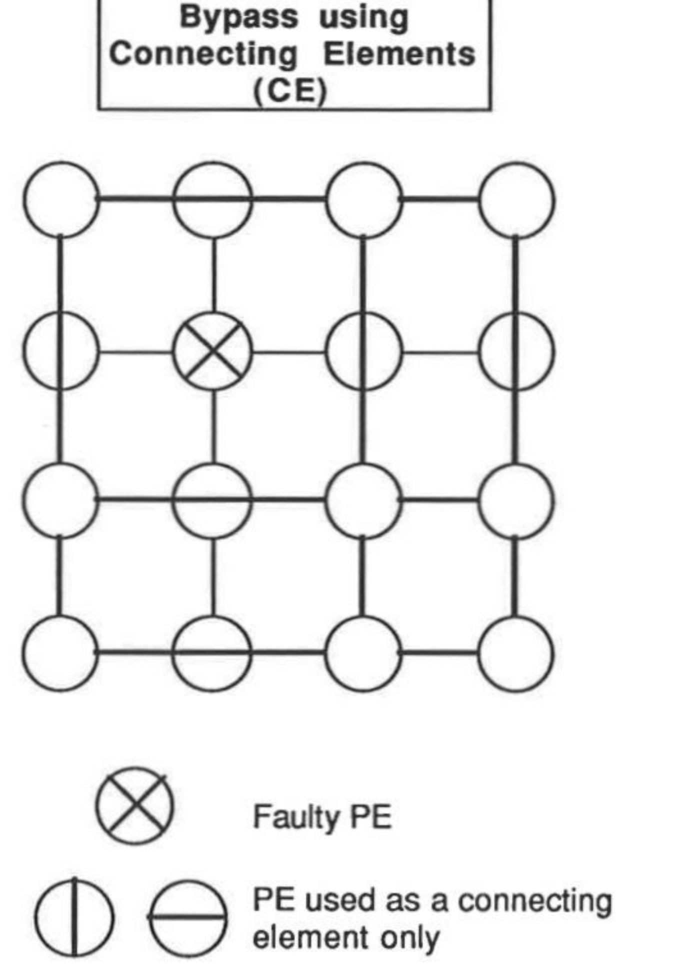


Figure 6.3: Achieving bypass using connecting elements.

by using a GI strategy to steer the columns [Moore & Mahat, 1985].

As a general rule, a reconfiguration scheme based on LI connections results in a lower utilization than schemes based on GI connections with similar levels of wiring overhead. On the other hand LI switched connections will be faster because they'll have fewer switches per connection. Additionally LI based schemes tend to have a higher tolerance to interconnection faults as the connections are not shared. A number of primarily LI connection based schemes have been suggested. What is perhaps the simplest example is illustrated in Figure 6.5 which also uses CE (Connecting Element – using working PEs as bypass) based bypasses as appropriate [Negrini et al., 1985]. The type of algorithm that is usually applied to LI and LI-like schemes is described here as a 2-D perturbation algorithm. This description is used because the final PE locations are perturbations on their fault-free locations, the possible number of perturbations being a small number only.

Another interesting approach using LI based fault tolerance is to place spares in the interstices between the normally used PEs, as shown in Figure 6.6. Each spare can then be used to replace any of its connected neighbours [Singh, 1985]. Singh also discusses interstitial options using fewer spares, that could be applied to higher yielding PEs. One complication with this scheme however is the wiring complexity required around each PE to allow the use of spares on all four sides of that PE.

A number of primarily LI based schemes have been proposed by workers at the Politecnico di Milano. They have proposed two basic interconnection structures and a number of distributed perturbation reconfiguration algorithms based on these structures. All of their reconfiguration strategies are aimed at reconfiguring an array with at most only one spare row and one spare column. The strategies are essentially LI in that only certain sets of PEs are allowed to be connected. Because of this, the maximum number of spares that can be usefully employed are 2N+1 spares for an N×N required array size. The relative richness of the interconnect provided allows for a high utilization of these spares. Due to the small number of spares allowed these schemes are best used for producing defect tolerant chips and not wafer-scale devices.

The first structure is based on the use of multiplexers [Sami & Stefanelli, 1983] [Sami & Stefanelli, 1986]. A schematic layout of the "direct reconfiguration layout", as it is called by the authors, is reproduced from [Sami & Stefanelli, 1986] in Figure 6.7. This structure allows for direct vertical connections, direct horizontal connections, and some limited horizontal bypass, as illustrated in Figure 6.8.

Higher spare utilizations can be achieved by using switches rather than multiplexers [Sami & Stefanelli, 1986] [Gentile et al., 1984]. Several distributed reconfiguration

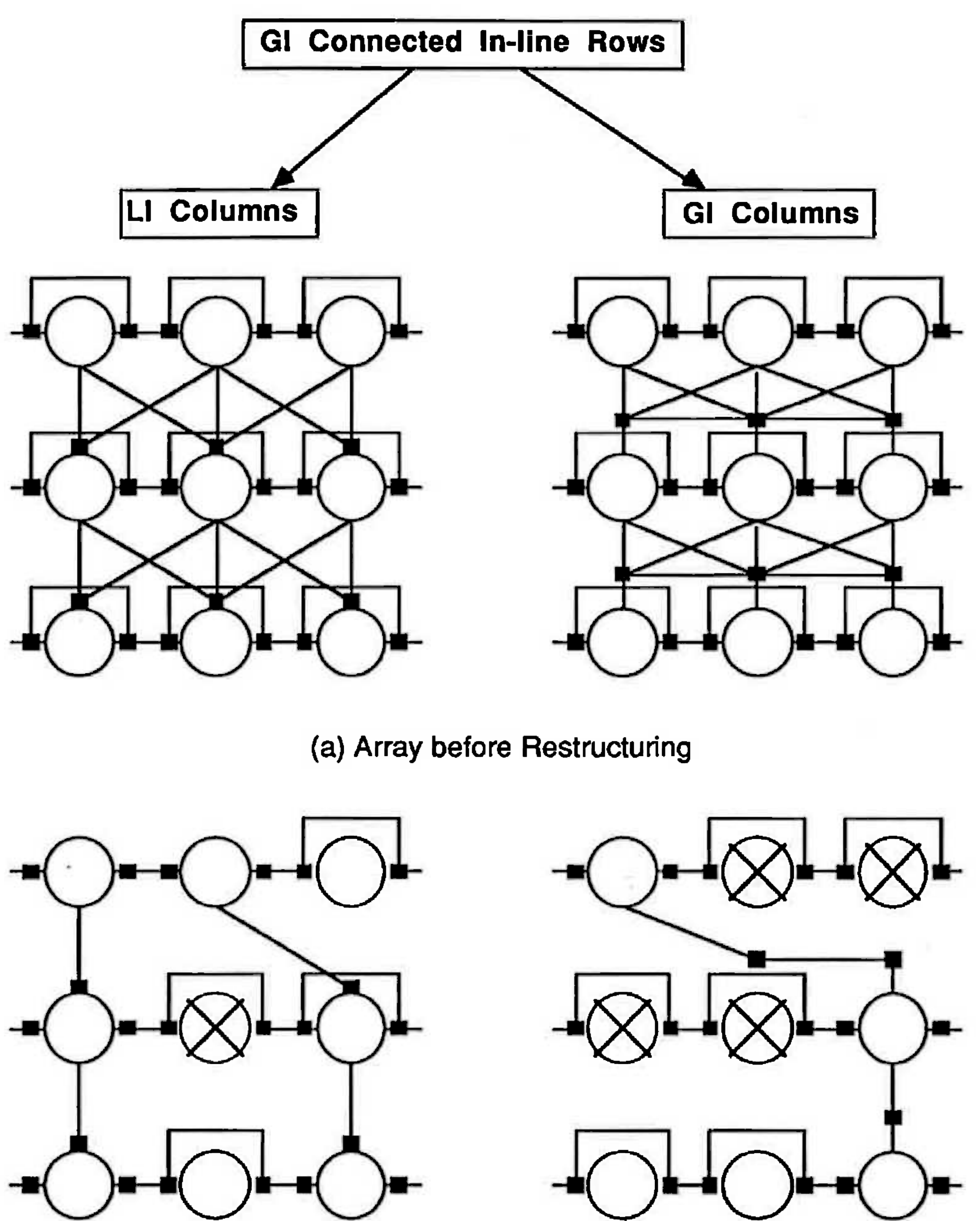
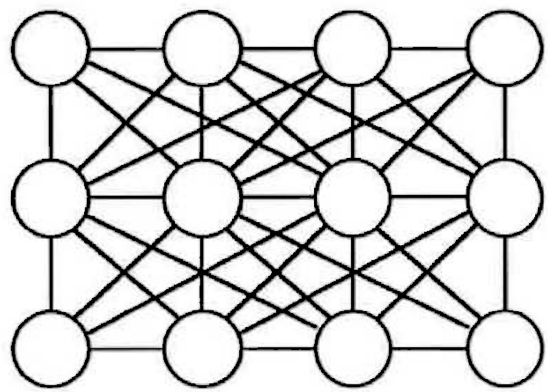
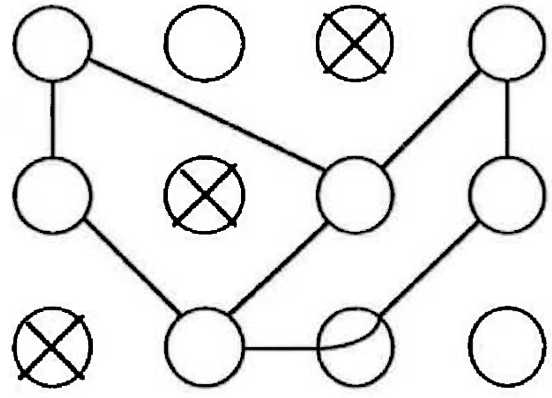


Figure 6.4: Approaches to bypassing within rows, combined with column steering.



(a) Reconfigurable array using LI and CE connections



(b) Example of a reconfigured 3 x 2 array

Figure 6.5: An example of a 2-D perturbation scheme based on LI and CE switches.

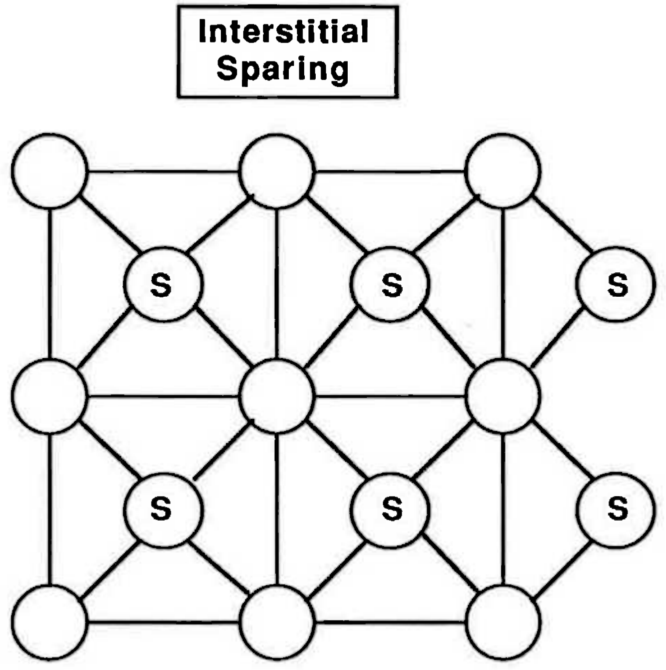


Figure 6.6: Interstitial redundancy scheme using LI connections only.

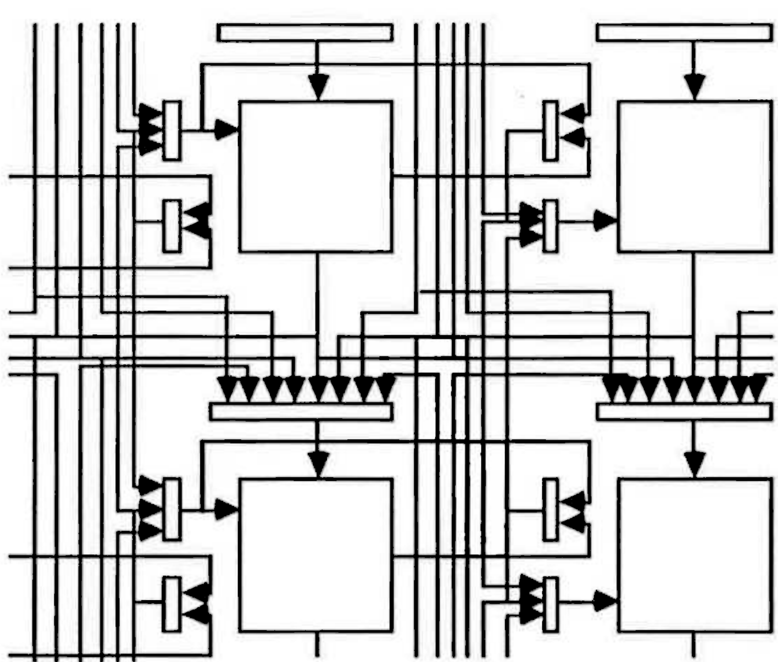


Figure 6.7: Schematic Layout of direct-reconfiguration structure. (reproduced from [Sami and Stefanelli, 1986] (c)IEEE)

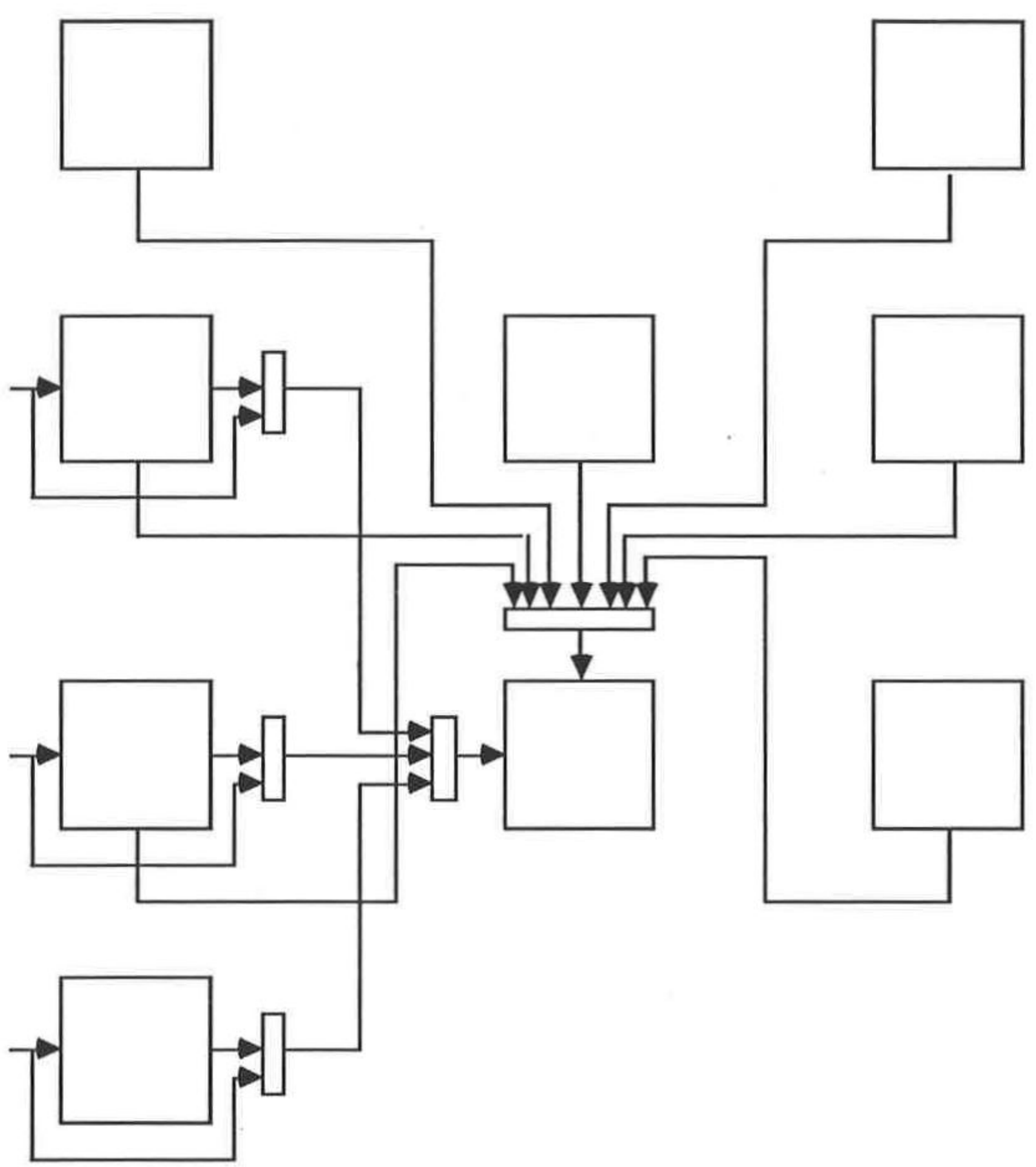


Figure 6.8: Possible logical connections to a cell, showing employment of multiplexor structures.

algorithms that make effective use of the 2N+1 spares provided are also presented in these references. This structure, showing allowable logical neighbours, is given in Figure 6.9. In [Gentile et al., 1984] a circuit layout developed for this approach is also described.

The most flexible approach to reconfiguring fault tolerant mesh arrays is to provide the array with a rich matrix of GI lines connected by general switches. This approach however does attract the penalties of a larger area overhead and the requirement for more complex reconfiguration strategies to ensure effective use of the extra flexibility.

Conceptually there are two ways to approach the organization of generally switched schemes. With the *channel approach* (Figure 6.10) the switches and GI connections form a number of channels between the rows and columns of the PEs [Hedlund, 1982] [Hedlund & Snyder, 1984]. Reconfiguration takes place by forming connections through those rows. Examples are given in Figures 6.11 and 6.12. The assignment of connections to channels and the setting of the switches is determined for the whole array, off the array in a centralized manner. The RVLSI program (see Sections 2.3 and 4.2) uses a channel approach to describe and reconfigure its arrays.

Reconfiguration strategies for the channel approach may vary. If so desired the full complexity of the problem can be avoided by using a hierarchical approach. With the hierarchical approach the PEs are organized into blocks. Reconfiguration consists of two steps then. First the connections within the blocks are reconfigured so as to form sub-arrays of the desired size. These sub-arrays are then connected to form the complete array [Hedlund, 1985] [Wang et al., 1987]. This approach does not result in an optimum utilization.

The complexity of optimally reconfiguring a channel based wafer can be approached analytically. In [Leighton & Leiserson, 1985] and [Greene & Gamal, 1984] the VLSI complexity (in terms of bounds on area, channel width and delay) of the reconfiguration problem is considered and bounds on utilization and connection length are determined. Generalized forms of divide and conquer algorithms that could be used to reconfigure arrays with 100% utilization (given a suitable channel width) are also presented.

In the frame approach, channels are configured slightly differently. Here the GI interconnects are specifically associated with particular PEs, rather than being treated as a completely separate entity (Figure 6.13.) An example of an array reconfigured using the frame approach is given in Figure 6.14. This association allows for easier programming of the switches, with less programming overhead, and better speed optimization as there are fewer switches used overall. On the other hand, the wiring area overhead is higher than with the channel approach. An example of a frame based

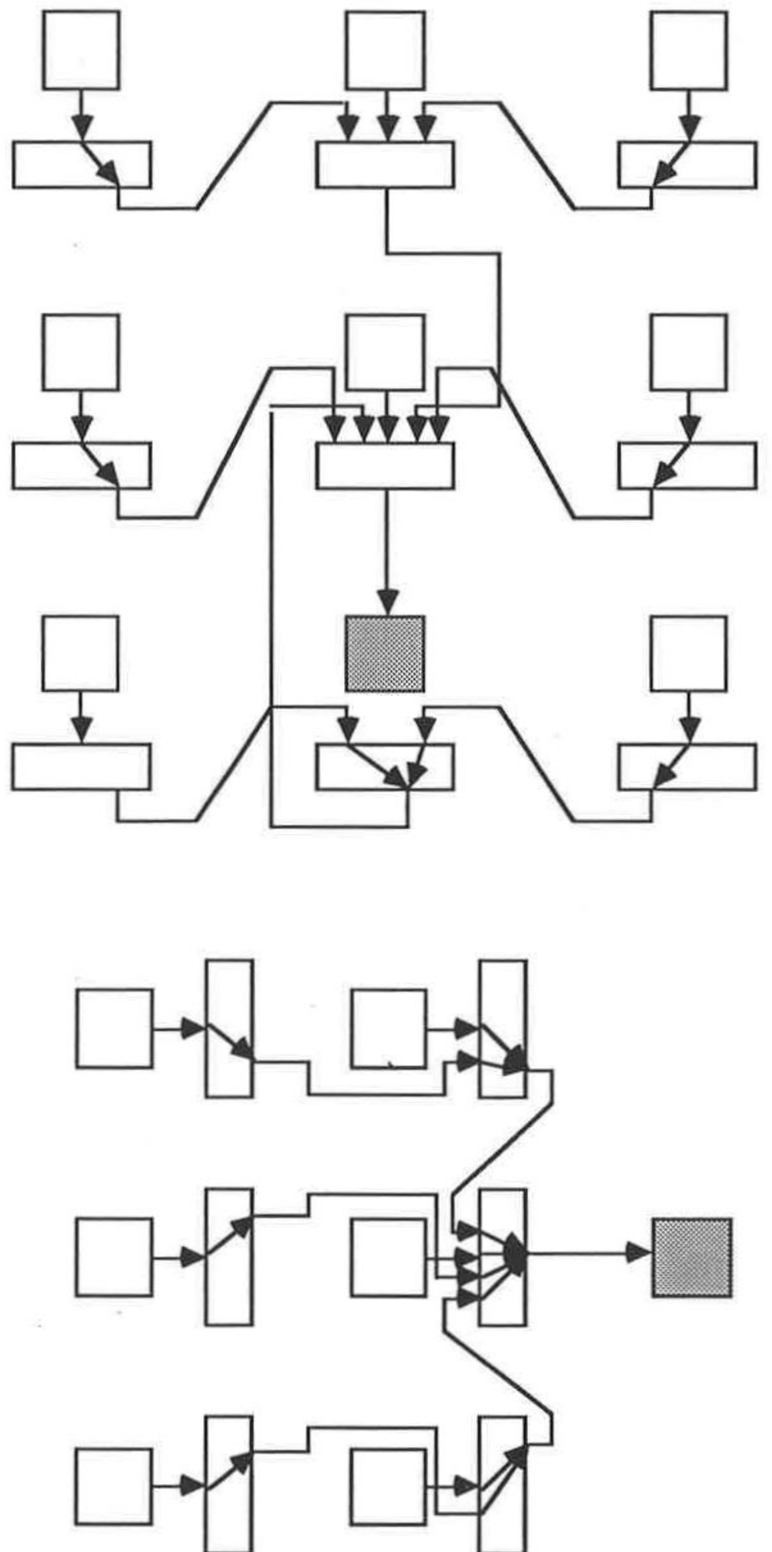


Figure 6.9: Neighbour connections allowed using switched LI connections

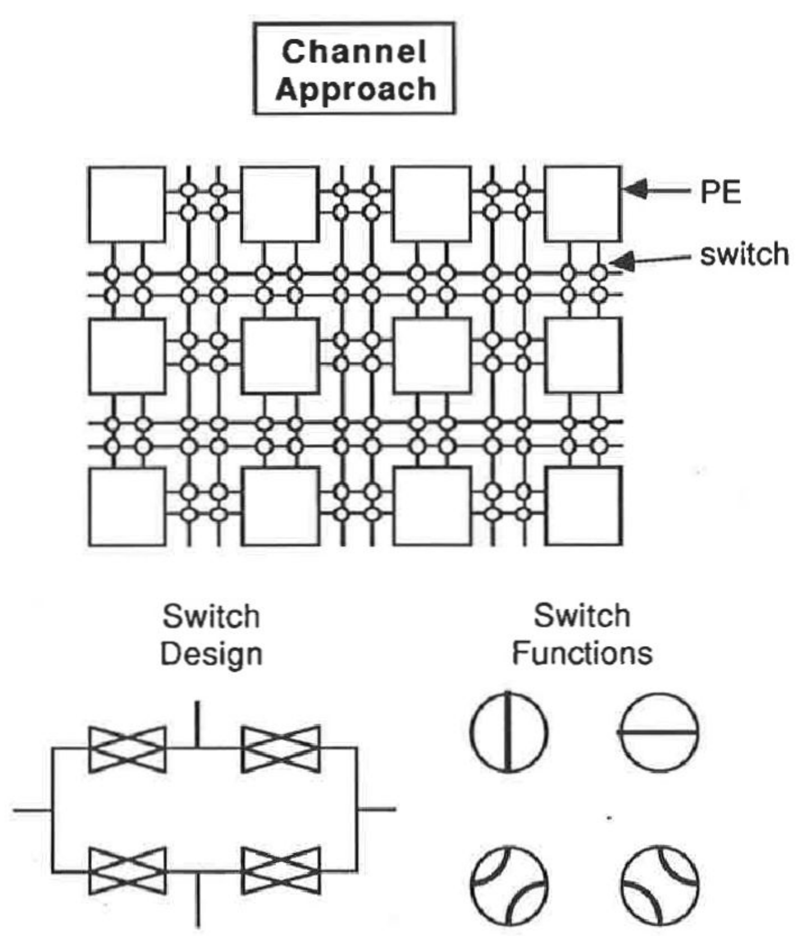


Figure 6.10: Channel approach to fault tolerant mesh arrays.

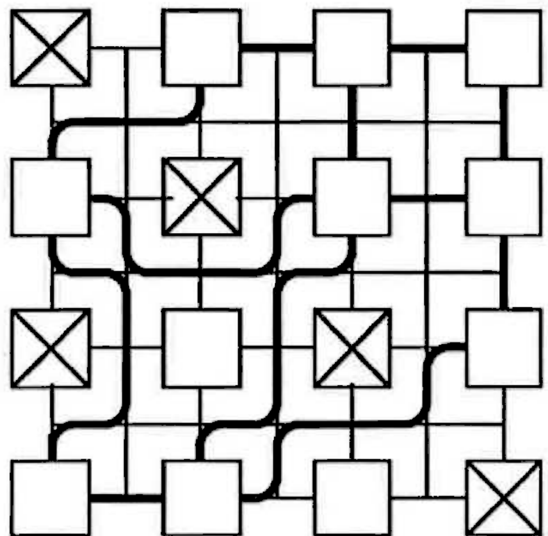


Figure 6.11: Example of a reconfigured array using the channel approach (single channel width.)

solution was given in the previous chapter.

6.3 Comparison of Mesh Redundancy Approaches

6.3.1 Monte Carlo Simulation

With the Monte Carlo method, the required quantities are determined by observing a suitably constructed random process [Buslenko et al., 1966]. By using a computer a large number of samples can be taken and a result determined to any required degree of accuracy.

For example, to determine the expectation $x = M\xi$ of a random variable ξ, N-fold samples of the value of ξ can be taken in a series of independent tests: $\xi_1, \xi_2, \ldots, \xi_N$. The mean value is then

$$\overline{\xi} = \frac{\xi_1 + \xi_2 + \cdots + \xi_N}{N} \tag{6.3}$$

According to the law of large numbers

$$\overline{\xi} \approx M\xi = x \tag{6.4}$$

with a probability as close to 1 as required for sufficiently large N.

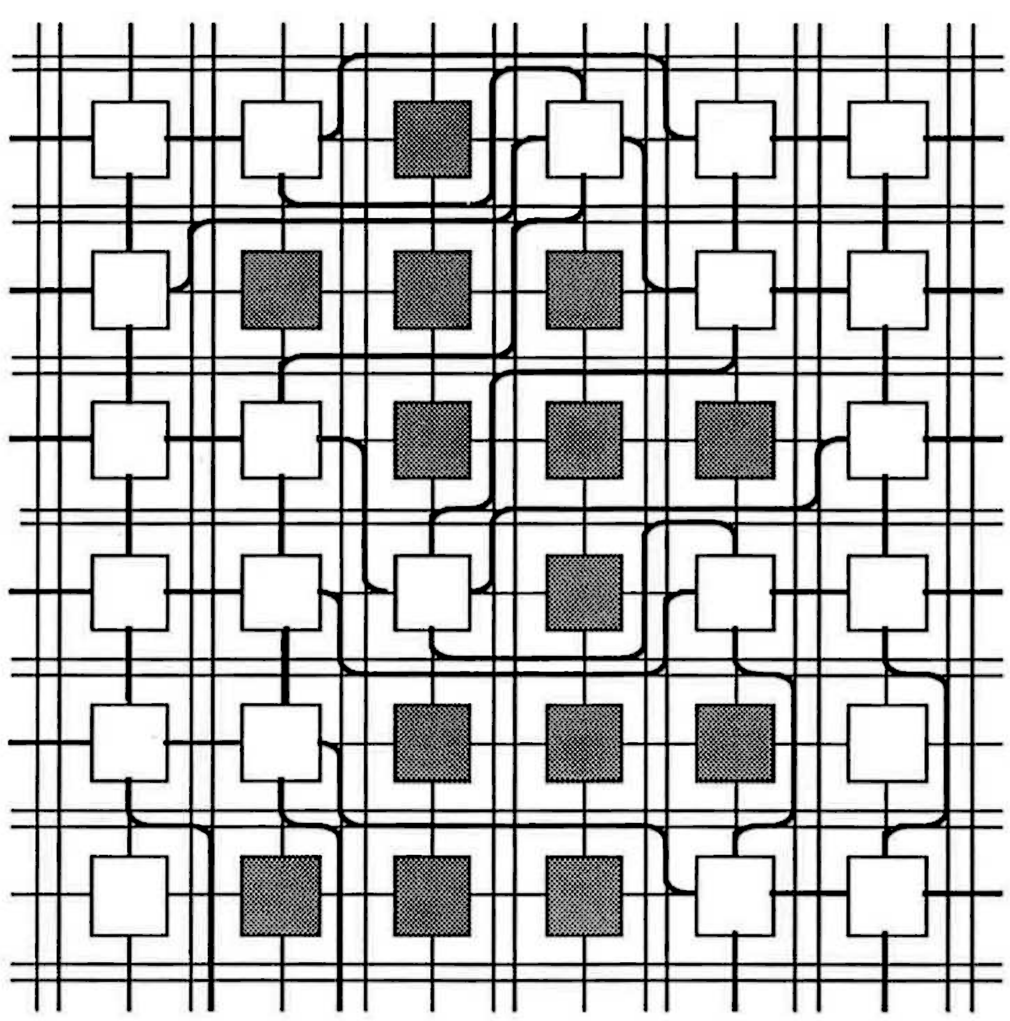


Figure 6.12: Example of a reconfigured array using the channel approach (double channel width.)

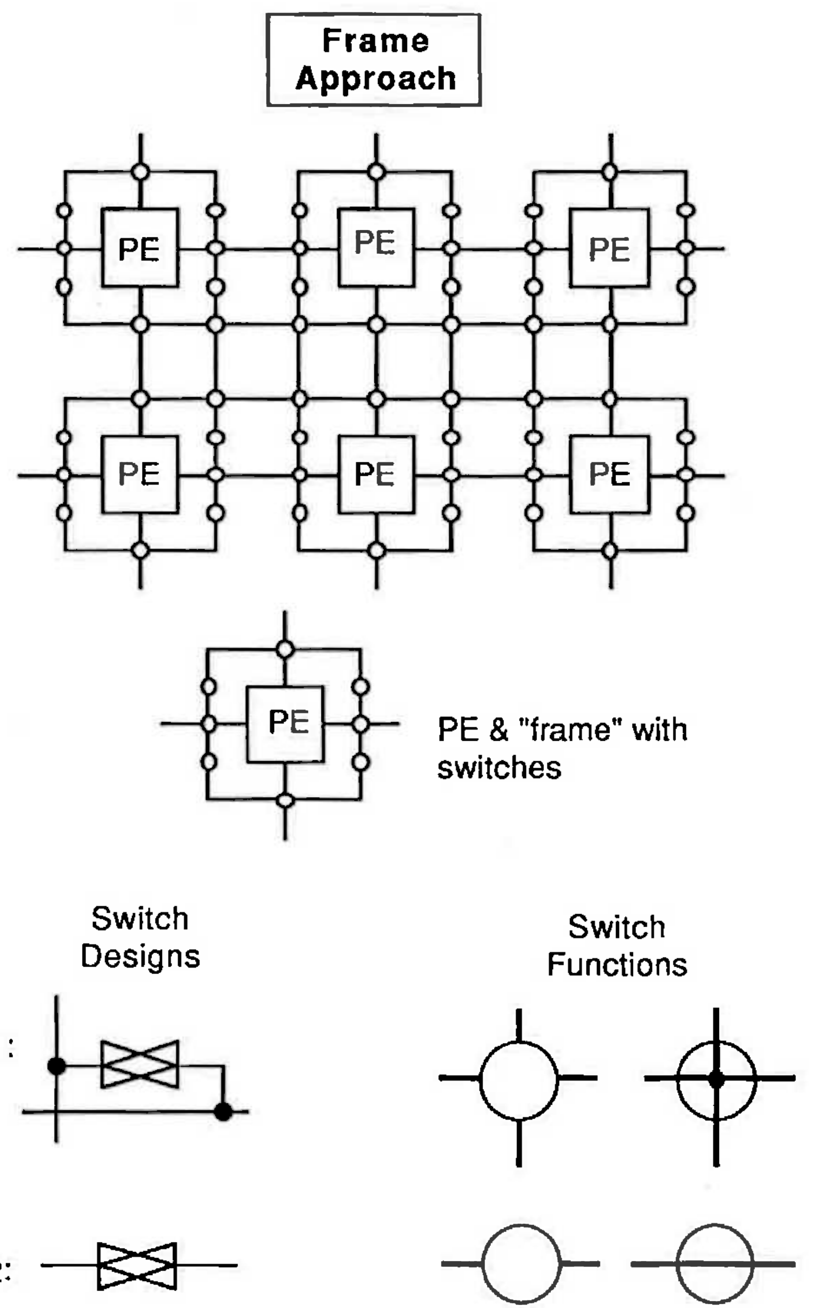


Figure 6.13: Frame approach to all-GI connected array reconfiguration.

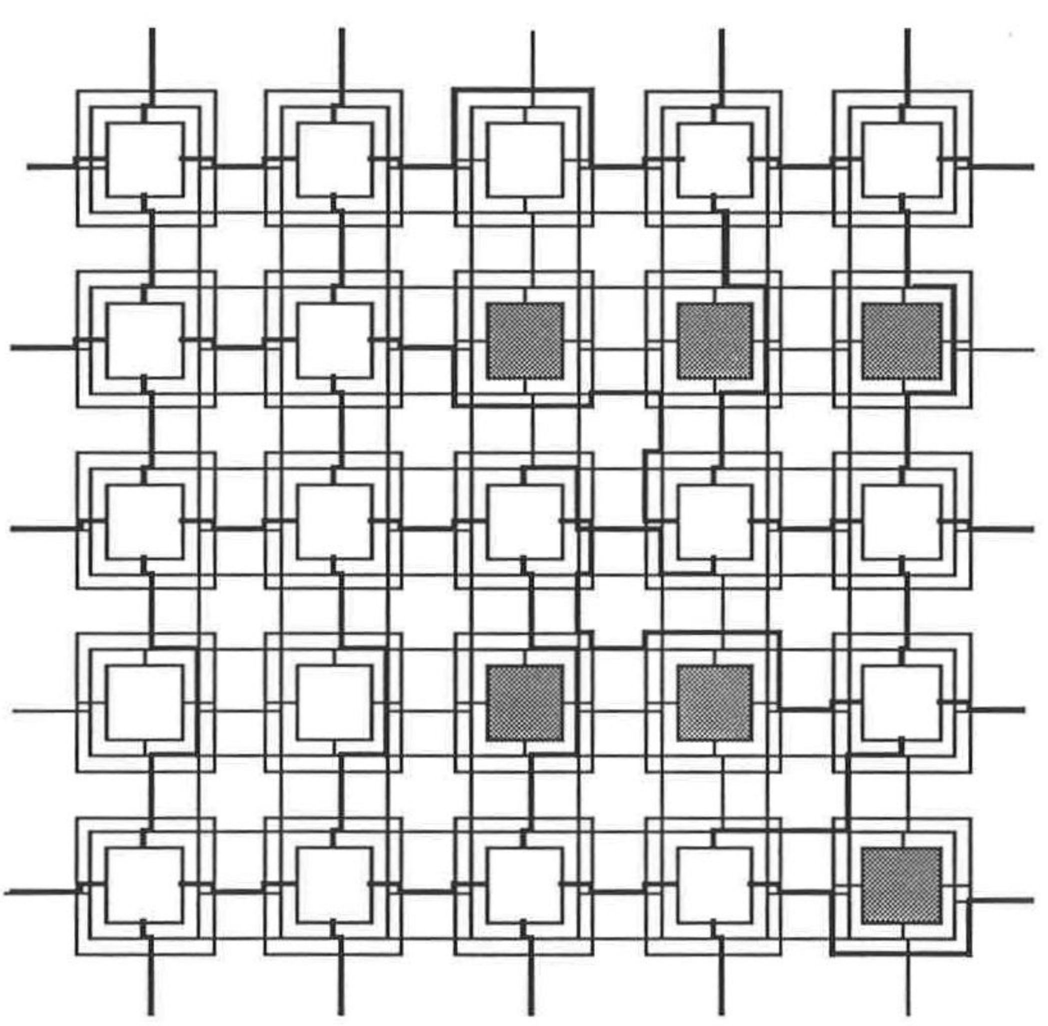


Figure 6.14: Example of a reconfigured frame approach array.

The error between $\overline{\xi}$ and $x = M\xi$ si given by

$$\delta \leq \frac{3\sigma}{\sqrt{N}} \tag{6.5}$$

where the variance σ^2 is

$$\sigma^2 = \frac{\sum_i^N (\xi_i - \overline{\xi})^2}{N}. \tag{6.6}$$

This can be used to determine the number of samples N required. If after S samples the error is δ_S and an error of δ_N is required then an estimate of the minimum number of samples required to achieve this is given by

$$N = S(\frac{\delta_S}{\delta_N})^2. \tag{6.7}$$

The Monte Carlo Method was used to determine utilization as a function of the number of faulty PEs for a variety of reconfiguration schemes. For each Monte Carlo step a pattern of faulty PEs was randomly produced. The appropriate reconfiguration algorithm was then applied to this fault pattern to determine either if (1) a properly configured array could be produced, or (2) what maximum size array could be configured, as appropriate depending on the particular reconfiguration scheme being studied.

6.3.2 Utilization Comparisons

A direct comparison of the three most basic (whilst useful) schemes for reconfiguring WSI mesh arrays in the presence of faults are presented in Figure 6.15. The schemes compared are the in-line row schemes given in Figure 6.4, the combined CE/LI scheme of Figure 6.5 and the best possible in-line rows scheme. In this last (hypothetical) scheme any good processor in a row can be used. This is included so as to ascertain the value of increasing the level of column steerability of the GI columns/in-line row approach (by adding extra horizontal column shift configuration lines.) The row bypass (Figure 6.2) and connecting element bypass (Figure 6.3) schemes are not included here because their utilization drops drastically for arrays with more than 10% faulty elements. For example, the row bypass approach achieves a utilization of 40% when 10% of PEs are faulty and a utilization of only 10% when 20% of PEs are faulty.

Interstitial redundancy, an LI-like scheme, [Singh, 1985] (Figure 6.6 shows an example with 100% redundancy) can produce superior utilizations to the GI column/in-line row scheme. Utilization results with different levels of redundancy are shown in Figure 6.16. R=150% interstitial redundancy produces a superior result when the percentage of faulty PEs is between about 30% and 50%. For R=200% this range is from 40%

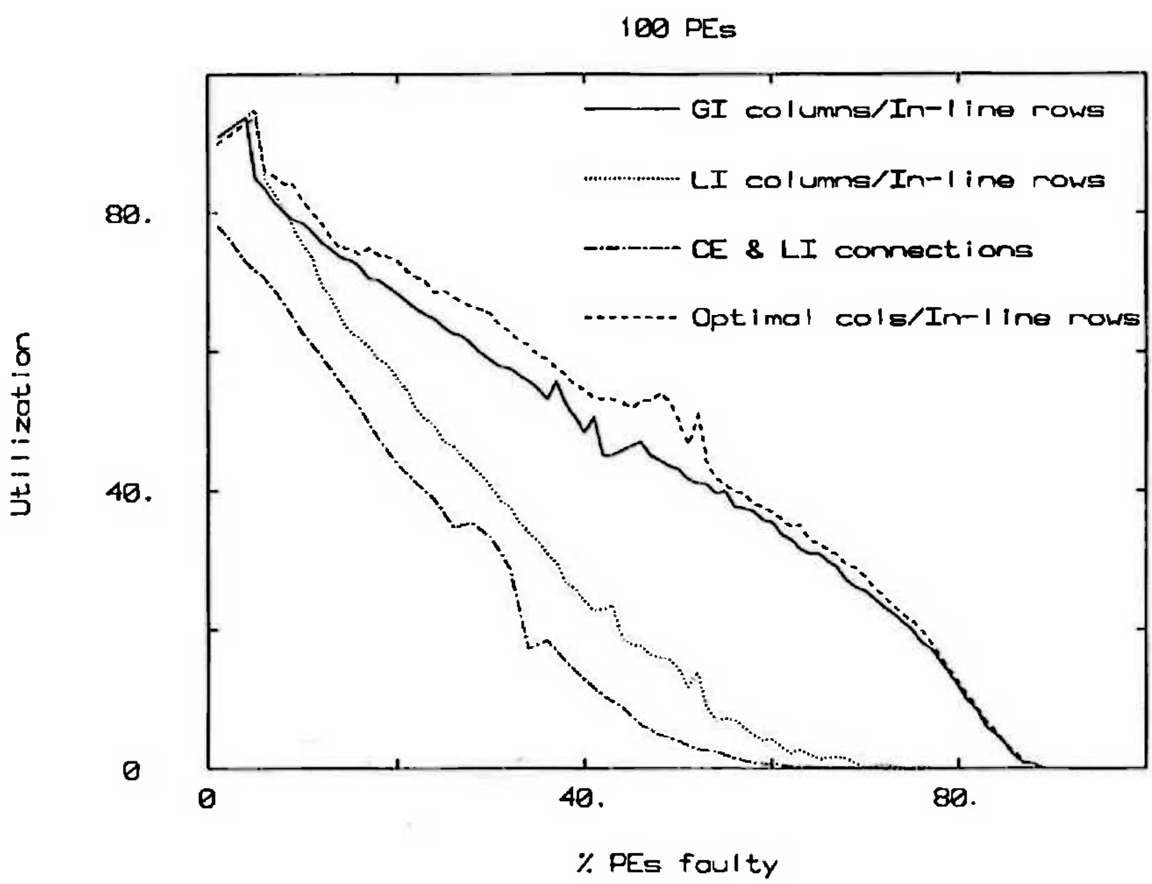


Figure 6.15: Comparison of the in-line row schemes and a combined CE/LI scheme.

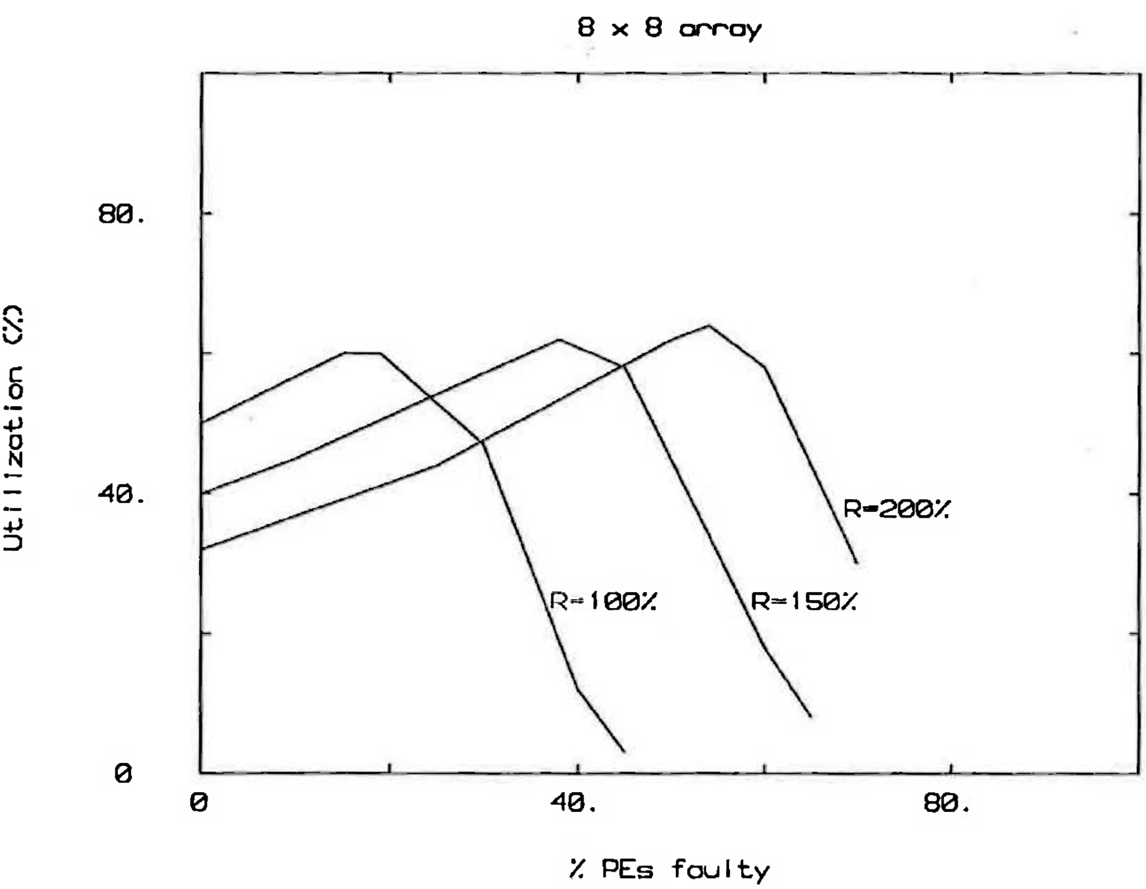


Figure 6.16: Utilization of Interstitial Redundancy. R is the level of redundancy.

to 70%. The switching complexity of this approach can be significantly higher than for the previously mentioned schemes though. For the R=200% case each PE has to have the capacity to be connectable to nineteen potential neighbours on all four sides. Singh doesn't describe the connection scheme that would allow this but its complexity will obviously be significant. In fact its actual implementation would most likely be more GI than LI simply to reduce the number of connections required.

As can be seen, amongst the schemes compared so far, the GI columns/in-line rows approach results in a generally superior utilization. The use of switched GI interconnect rather than direct LI interconnect brings a clear utilization advantage. The only costs are an extra line within each row and a more complex reconfiguration algorithm. The extra complexity of the reconfiguration algorithm rules out the possibility of the reconfiguration step being carried out by hardware internal to the array, as is often suggested for the LI columns approach. This is not such a disadvantage as it may seem however, because, for the WSI case, if no special precautions are taken this hardware can contribute significantly to the array kill area A_{kill} and seriously impact yield, as described in Chapter 3. In [Evans & McWhirter, 1986] a method is presented for overcoming this problem for self-organizing LI column arrays.

To further increase utilization requires the use of all-GI connected schemes. These allow the steerage of rows as well as columns around faulty PEs. A large number of various all-GI connected schemes were simulated by hand (as automating the task requires considerable effort) and the results are given in Figure 6.17. The all-GI schemes consistently produce utilization figures better than 70%, with the double channel and frame approaches providing a utilization of around 90% for this size array.

One major disadvantage however of these GI schemes is the complexity involved in determining an optimal reconfiguration. Generally it seems that the only solution is too carry out a heuristically guided depth first search to determine PE connectivity and then use a routing step to map the required connections onto the wires provided. Such a set of tools have been written to reconfigure the RVLSI wafers, which use a channel approach [Anderson, 1986]. A divide and conquer approach can also be used for certain arrays [Leighton & Leiserson, 1985].

Another approach, that has been suggested as a solution to reducing the complexity of carrying out this reconfiguration step, is to perform the assignment in a hierarchical fashion. This has been suggested by Hedlund [Hedlund & Snyder, 1984] and Su [Wang et al., 1987]. In Hedlund's original approach the PEs are grouped into logical blocks of twelve. Each block is tested to determine if there are four working PEs present. If each block in a column of blocks has four working PEs then all those blocks

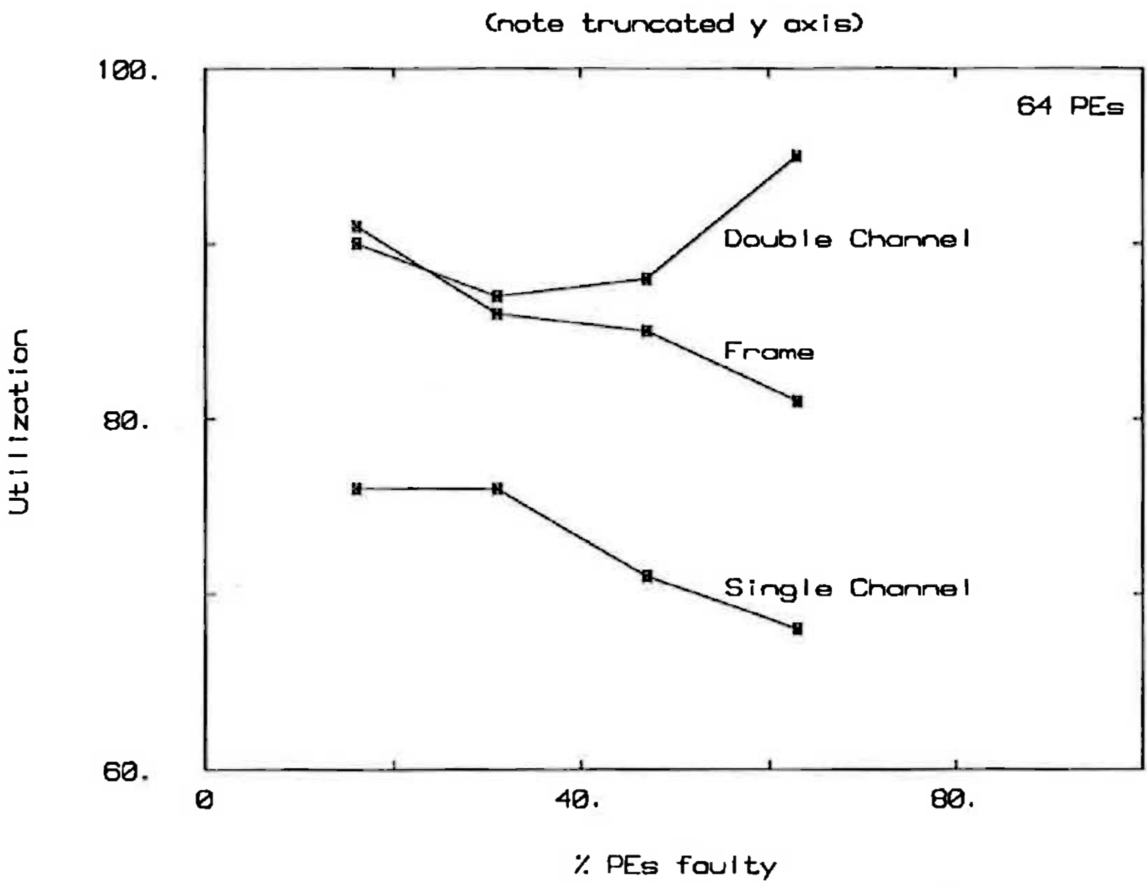


Figure 6.17: Utilization comparison of different all-GI connected approaches.

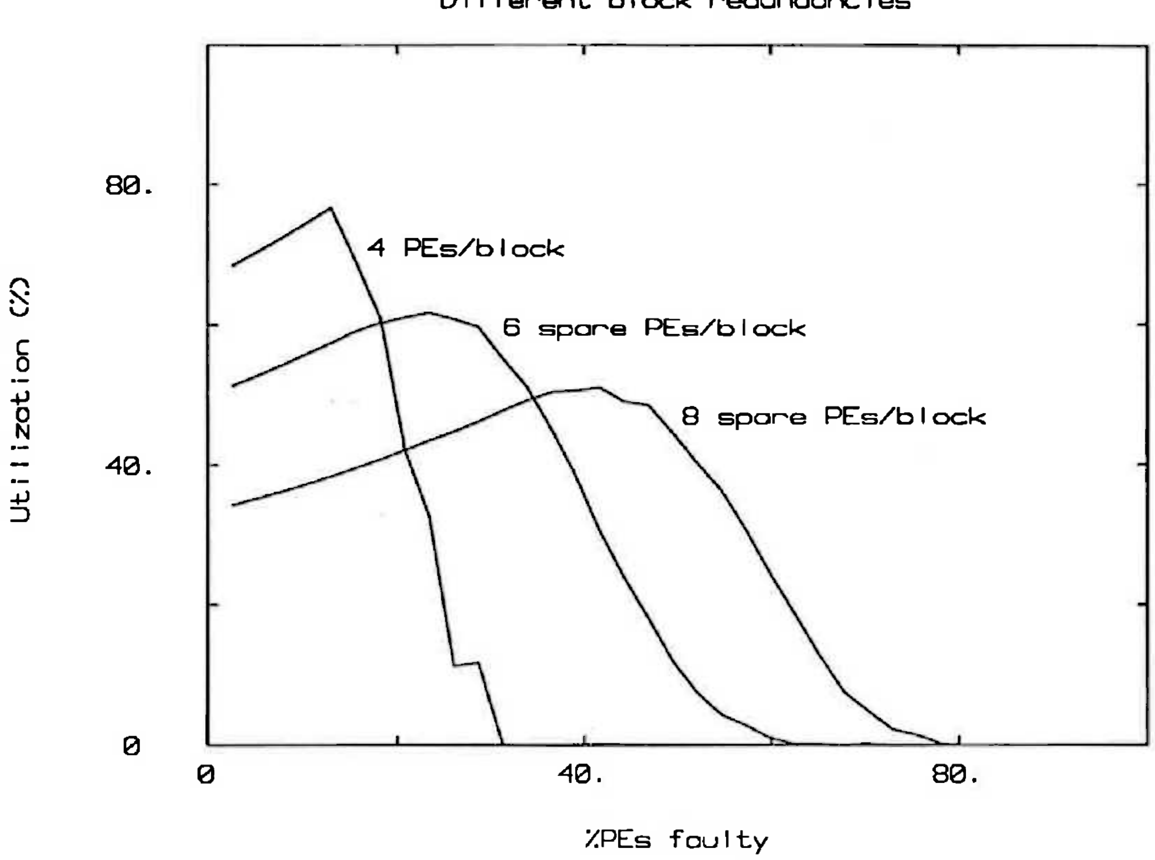


Figure 6.18: Utilization vs. % of PEs faulty for a number of variations on Hedlund's hierarchical scheme. The labels refer to the number of PEs that are considered spare in each block of 12 PEs.

are used. If any of the blocks don't have four working PEs then the whole column of blocks is bypassed. A number of variations are of course possible on this approach. The most obvious one is to vary the number of PEs allocated as spares in each block. Utilization with a varying number of PEs is given in Figure 6.18. Hedlund's arrays have exactly the same basic physical structure as the double channel all-GI arrays illustrated earlier. Though greatly simplifying the reconfiguration step, utilization is reduced by between 20% and 30% by adopting this hierarchical reconfiguration strategy, rather than tackling the full complexity.

Another possible variation is to improve the manner in which blocks that do not have the required number of working PEs are handled. Rather than bypassing a whole column for each such bad block a more efficient method could be used, such as applying the GI column/in-line row scheme to the blocks. The utilization that can be achieved

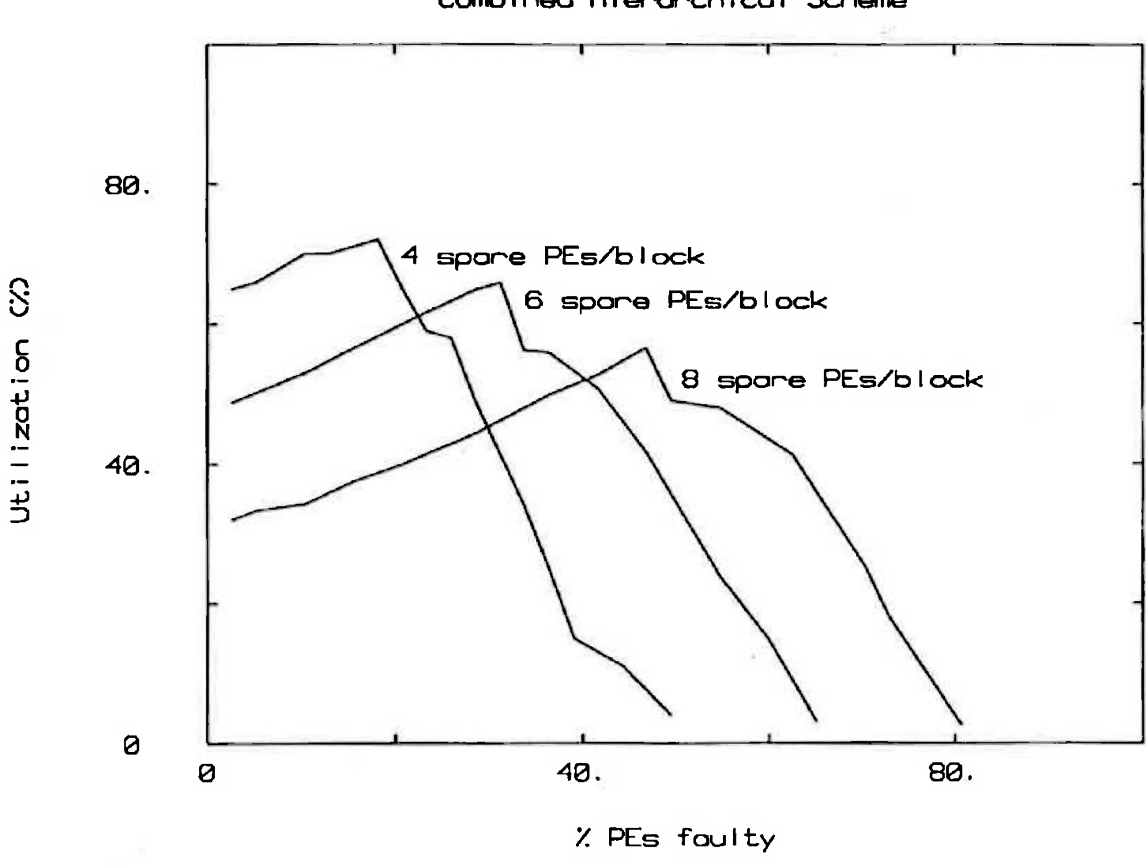


Figure 6.19: Utilization achieved by combining the hierarchical approach with the GI-column/in-line row reconfiguration scheme for the blocks.

using this approach is given in Figure 6.19. The only difference between these plots and the previous plots in Figure 6.18 is that the right hand declining tails of the new plots are higher. The optimal points have not changed as at these points there are usually no bad blocks. Thus doing this only brings a limited return. The double channel physical structure could be retained for this approach as the inter-block channels are sufficient to embed a GI columns/in-line row scheme. So this enhancement comes at no cost.

Another possible enhancement is to dynamically change the size of the block, and the number of working PEs required in the block to obtain an optimum utilization. This approach retains the 2 channel structure of Hedlund's original array and does give some yield advantage. In more detail, the scheme works by scanning the array from left to right considering blocks of fixed height (3 PEs high, as before), but of variable width, and with a variable number of good PEs being required from each block. (The number

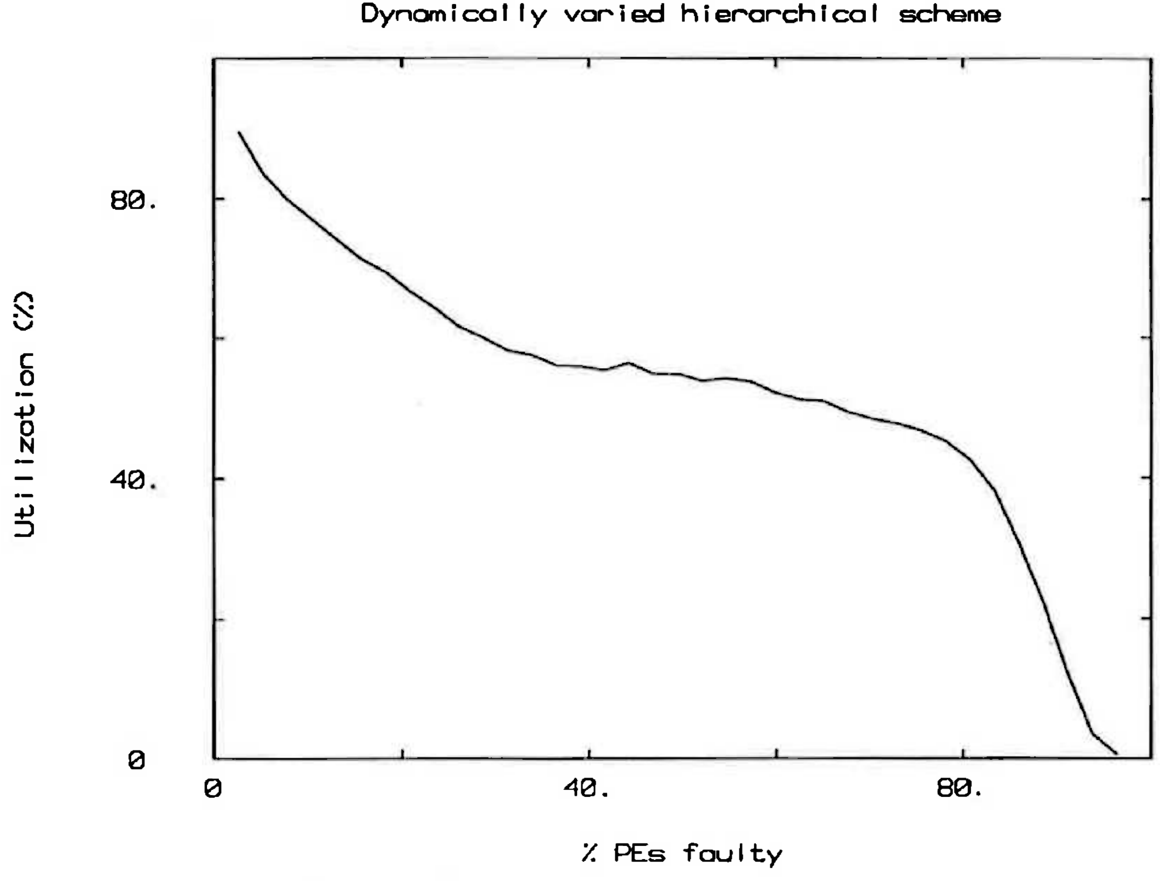


Figure 6.20: Utilization of hierarchical scheme using varied width columns.

of PEs to be used is kept to an even number however so as to ensure routability of the final result.) A number of widths and PE numbers are considered until the optimum usage of PEs is obtained for that variable width column. Consideration is then moved to the next column. As shown in Figure 6.20 this results in an generally improved utilization over the fixed column width and combined hierarchical approaches at no cost at all. Its biggest advantage however is that the utilization plot is not peaky. As discussed before the number of defects per wafer can vary significantly between wafers. Thus to most efficiently use all the arrays produced a consistently high utilization is preferred, and a peaky utilization result does not provide this. This is assuming that a range of product sizes can be used. The interstitial redundancy scheme also suffers from this limitation of producing an optimum utilization only for a restricted range of % PEs faulty.

It is worth noting that some suggested hierarchical schemes (*eg.* such as in

[Wang et al., 1987]) use a different reconfiguration structure between blocks to the structure used within them. By fixing the block size in the hardware the possible application of dynamically changing the block size is prevented. The benefit of such schemes though could be that the number of switches in a connection path may be reduced when compared with the approaches being presented here. However no-one has presented such a scheme.

Another enhancement, that will not be explored here, is the sharing of PEs between blocks. Like the variable sparing enhancement this will most likely mainly affect the tails of the distributions in Figure 6.18 rather than raise the utilization peaks.

It is evident in all of the utilization plots above that the graphs are not monotonically smooth. The reason is that the possible array sizes are quantized to a sub-set of the non prime numbers. This sub-set differs with the scheme under investigation. For any of the in-line schemes the number of connected PEs must be a multiple of the number of rows. For the all-GI schemes, the number of PEs used tend to have numerical factors that are not very dissimilar in value. As the number of good PEs available decrease the number of PEs used tend to jump between these quanta in a non continuous fashion.

6.3.2.1 Effect of array size

What confidence do we have that the results presented above track for array sizes other than the ones presented? To test for this effect some schemes were simulated with a number of different array sizes.

Figure 6.21 shows the effect of array size on utilization results of the In-line rows/LI columns approach. Here the difference can be up to 20%.

In the interstitial redundancy approach utilization decreased by over 10% when the array size was quadrupled [Singh, 1985].

It would be expected that Hedlund's hierarchical scheme would suffer a slightly decreased utilization as the number of rows of blocks is increased and a constant utilization as the number of columns of blocks is changed. Its hierarchical nature results in the latter, whilst the fact that a single unusable block leads to the loss of the whole column naturally leads to the former conclusion.

In order to maintain a constant utilization for the all-GI channel approach arrays, it is necessary to increase the channel width at the rate of $O(lglgN)$, as N increases, where N is the number of PEs in the unreconfigured array [Leighton & Leiserson, 1985]. The similarity between the channel approach and the frame approach suggests that the

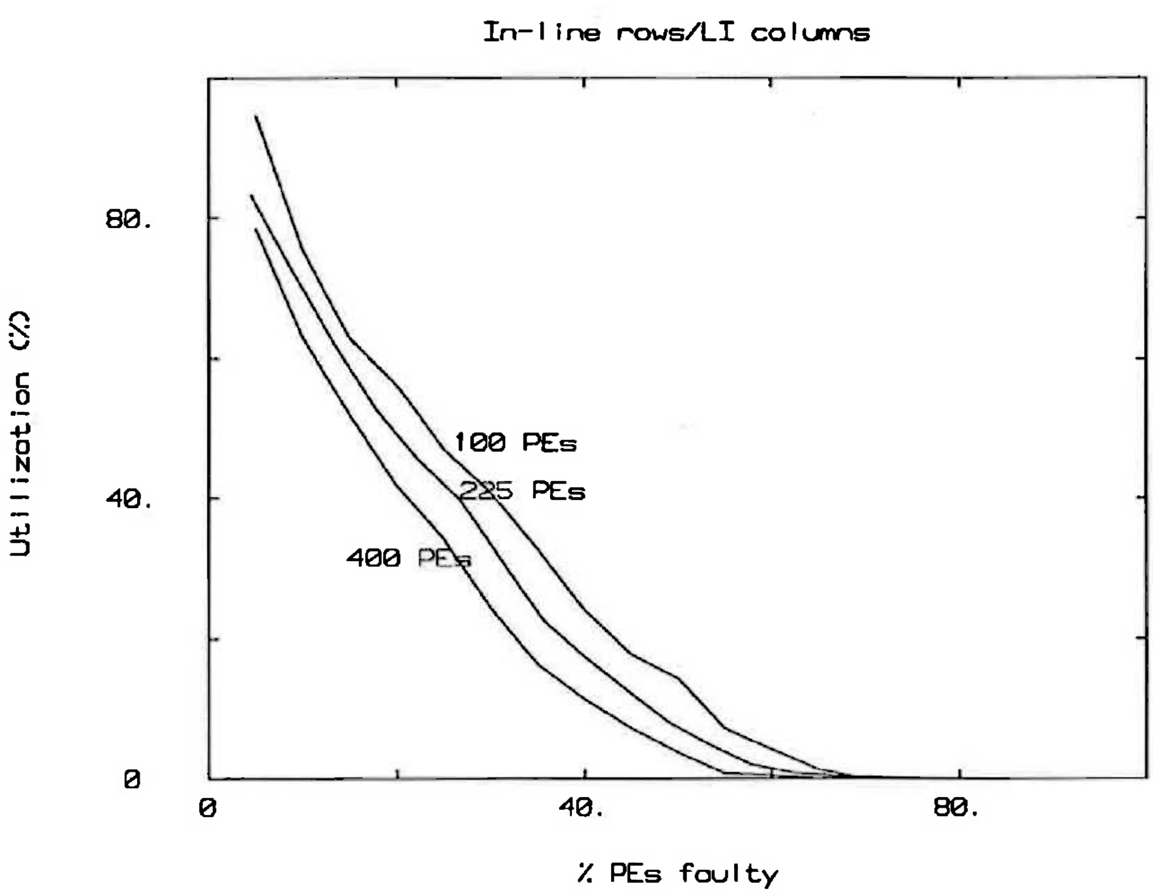


Figure 6.21: Effect of array size on utilization of in-line rows/LI columns scheme.

utilization of the latter will also degrade with increasing N unless additional frames are added at this rate.

6.3.2.2 Effect of Clustering

Clustering enters the models discussed here in three ways:

1. In the base yield model the α parameter reflects the level of clustering as discussed in Chapter 3. The presence of clustering results in a greater variance in defect densities between wafers and chips than if clustering were not present.

2. In obtaining a fault tolerant yield model the probability distribution between the number of faults and the number of faulty PEs can be adjusted to take account of the presence of clustering. This was discussed in Section 3.5.3 where its effect was shown to be minimal.

3. The utilization U (or C_{ijNR}) can be affected by clustering. A higher probability of there being neighbouring faulty PEs may change these figures.

This section discusses the effect of the last point.

The sensitivity of the utilization results to clustering was determined by simulating reconfiguration of faulty arrays under two separate assumptions:

1. The assignment of faulty PEs to array locations was purely random.

2. There was a finite probability the adjacent PEs were faulty.ch other. This probability was set here at the high figure of 0.8. In the Monte Carlo simulation, as faulty PEs were assigned to array locations, a probability of 0.8 was assigned to the possibility of that faulty PE being placed adjacent to the last faulty PE.

The first assumption was used in determining the utilization figures given above. Results under the second assumption are compared with the first here. These comparisons are given in Figures 6.22 to 6.27.

No scheme's utilization improves under these clustering conditions. However, the degradation of the in-line and all-GI schemes is not as bad as it is for the others. The degradation of utilization under clustering conditions is worst for Hedlund's scheme because if a cluster of faults falls in a block then the whole column of blocks may be lost. The dynamic hierarchical approach degrades less under clustering assumptions than the fixed block size hierarchical approach.

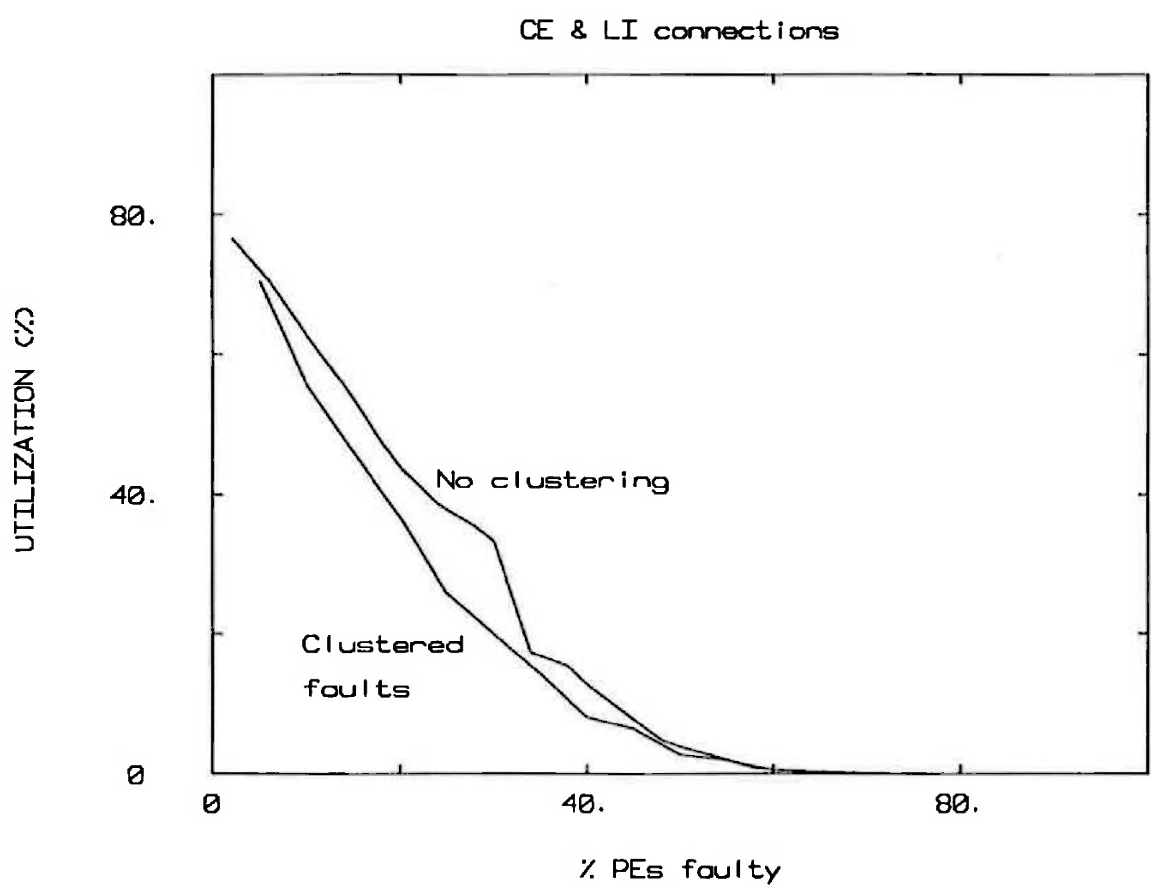


Figure 6.22: Utilization of the CE & LI connections only scheme under assumptions of clustering and no clustering.

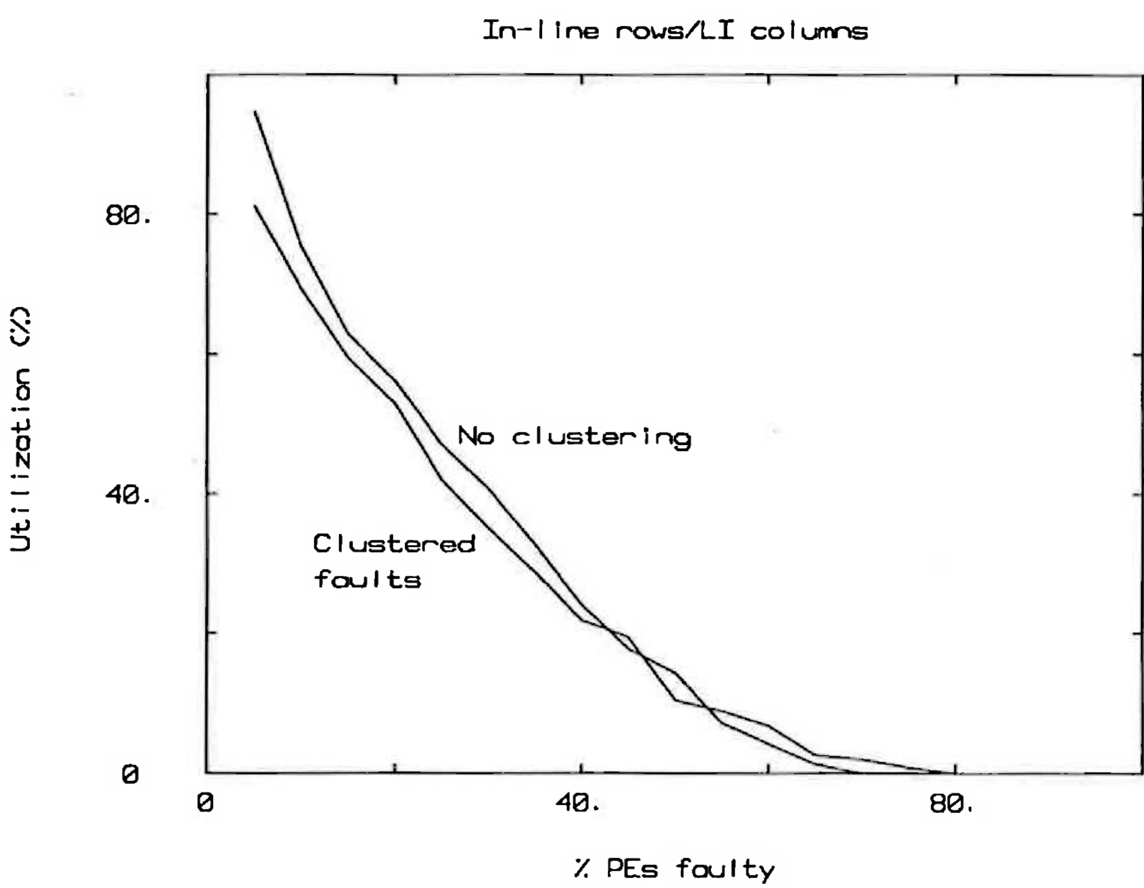


Figure 6.23: Utilization of the in-line rows/LI columns scheme under assumptions of clustering and no clustering.

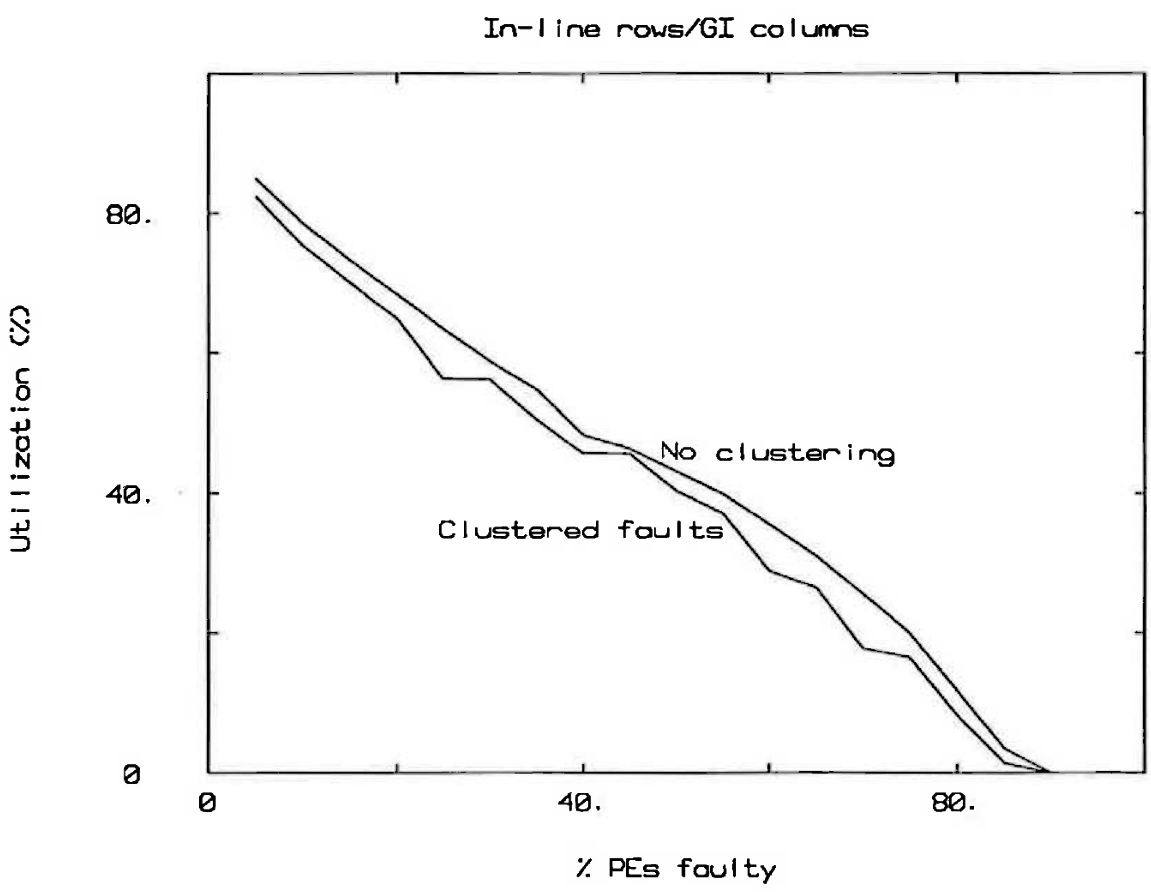


Figure 6.24: Utilization of the in-line rows/GI columns scheme under assumptions of clustering and no clustering.

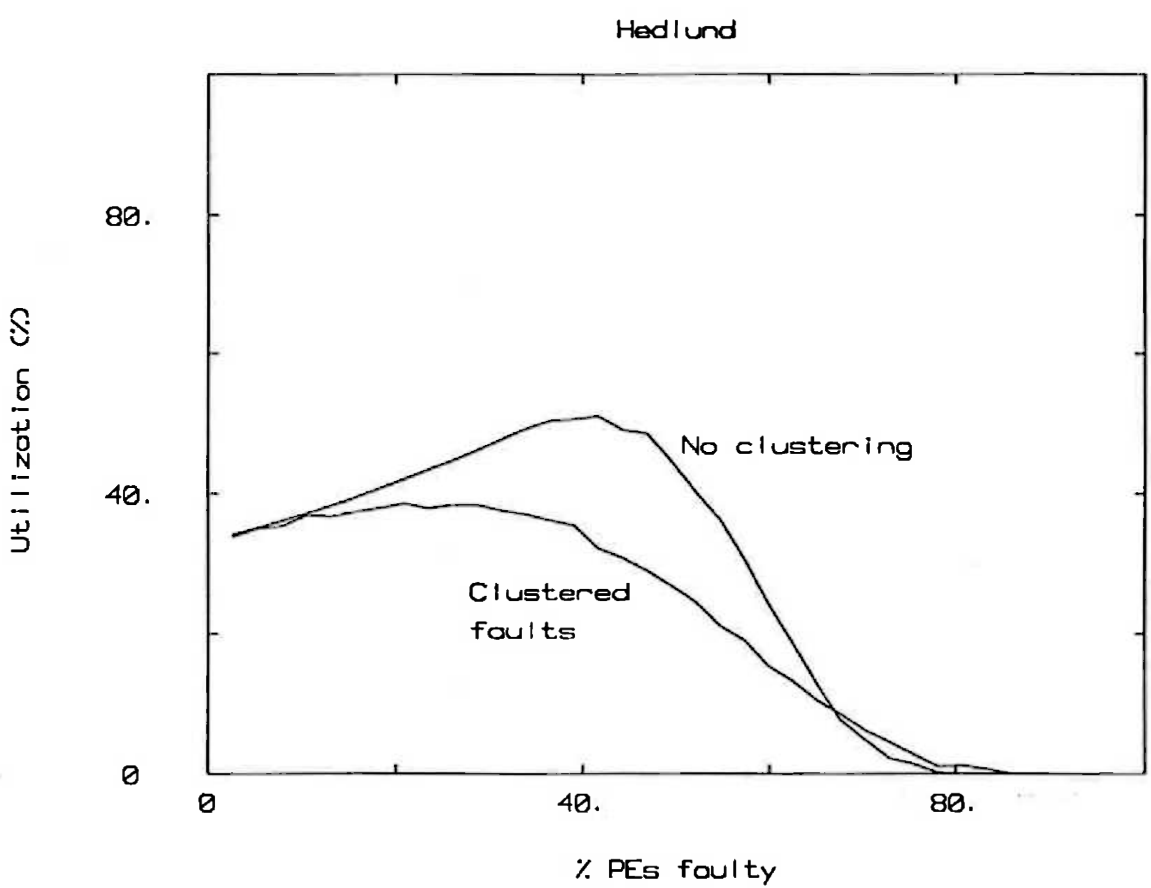


Figure 6.25: Utilization of Hedlund's scheme under assumptions of clustering and no clustering.

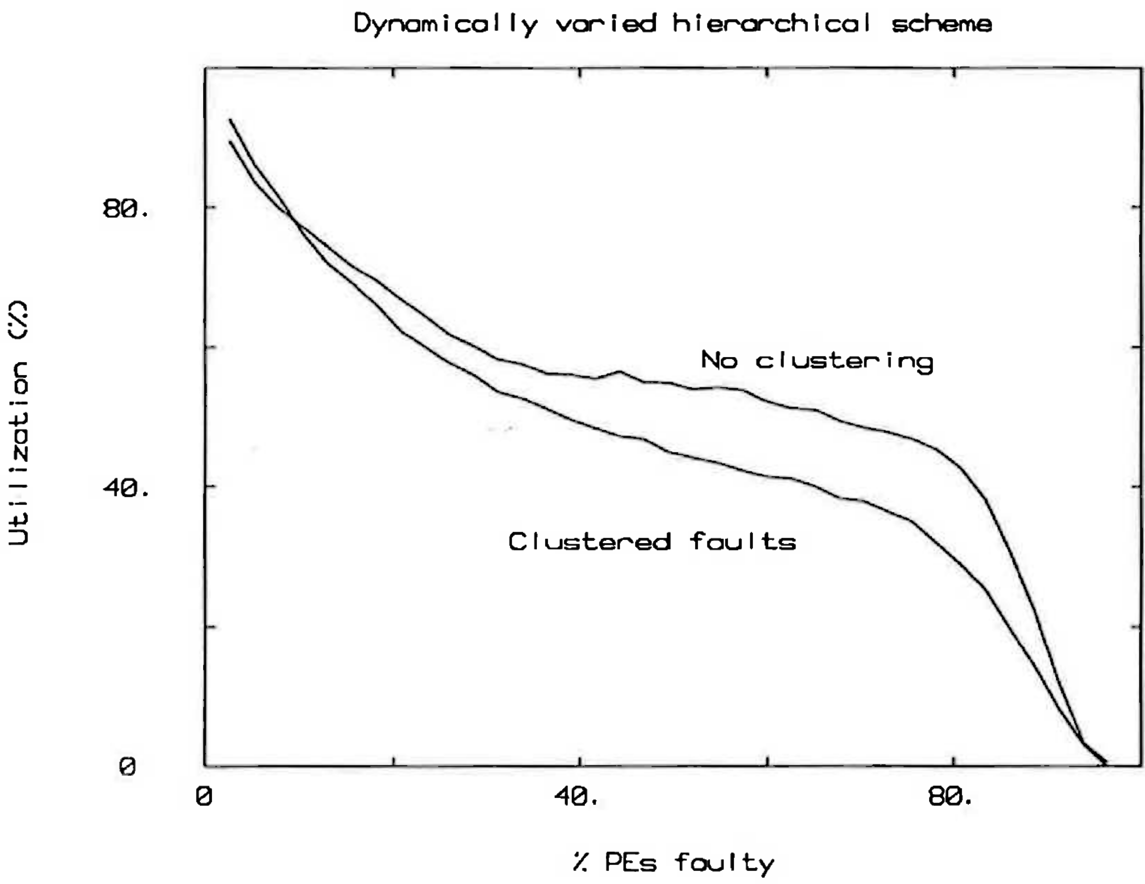


Figure 6.26: Utilization of the dynamically varied block size hierarchical scheme under assumptions of clustering and no clustering.

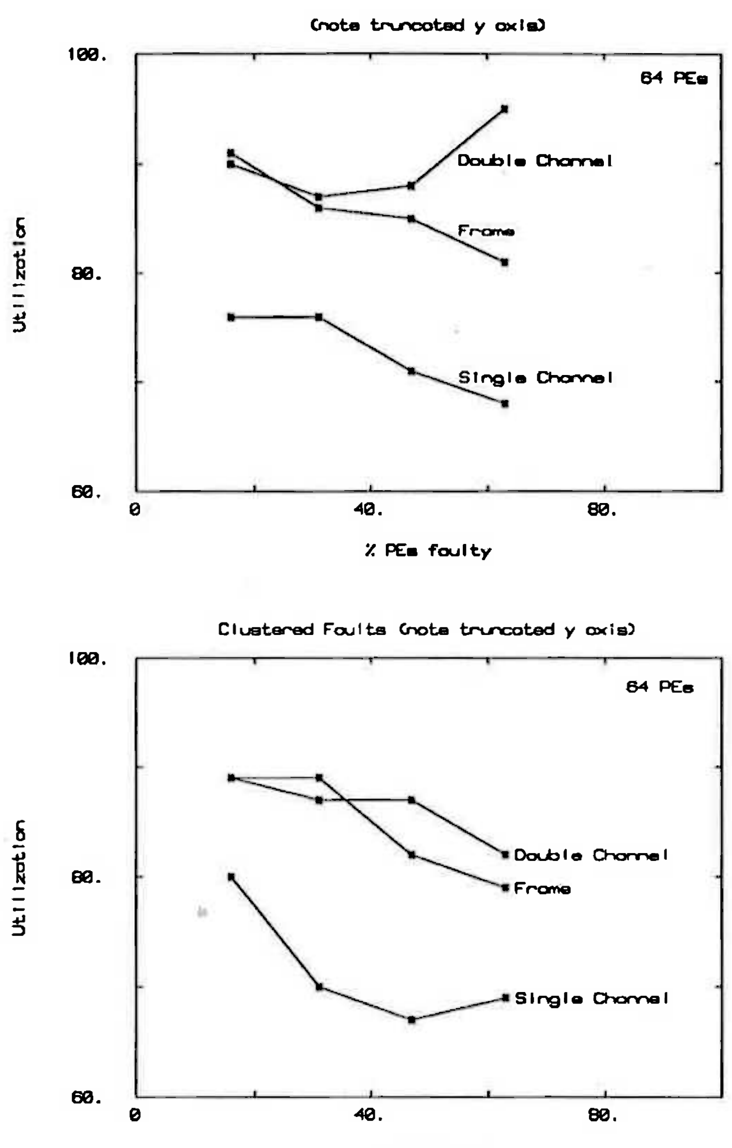


Figure 6.27: Utilization of all-GI schemes under assumptions of clustering and no clustering.

6.3.3 Speed Comparisons

The speed of a WSI processor array depends on a variety of factors:

1. *Implementation Technology.* This refers to both the IC fabrication technology as well as the technology used to implement reconfiguration. Most of this section will be devoted to speed comparisons based on implementation in a CMOS process with CMOS switches being used to route reconfigured lines. This is actually the worst case but it is an important case, as it is the most widely available technology for WSI implementation. Use of hard fusing/joining technologies can sometimes bring substantial speed improvements at the expense of a loss of post-processing flexibility and longer lead time and cost. This was discussed in detail in Chapter 4. The use of a BiCMOS process to produce bipolar switches for reconfiguration routing offers a lot of promise for improving speeds.

2. *Number of switches in the switched path.* (To be referred to as *switch length.*) This is mainly a function of the reconfiguration scheme used, though it will vary with the number of faulty PEs present.

3. *Length of interconnect in switched path.* This depends primarily on the size of the PE, and secondarily on the type of reconfiguration scheme being used.

4. *Usage of schemes to increase speed.* These were discussed in Chapter 4 and an example was given in Chapter 5. These depend on the application as well as the particular reconfiguration scheme being used.

In general the worst path length for a particular number of faults is not always the same. For example consider the graph in Figure 6.28. Here the relative frequencies of different maximum switch lengths are plotted against the percentage PEs that are faulty for the in-line rows set of schemes.

The distribution of maximum path lengths is quite sharply peaked, particularly for lower levels of faults. This means that if speed is important the reconfigured arrays with the longest switch lengths can be rejected, without significantly impacting yield and utilization. For higher levels of faults, where the peak is not quite as sharp, it is possible to make some tradeoff between speed and utilization.

Both of these effects can be seen by referring to the 50% PE fault distribution curve in Figure 6.28. For this array, with 100 PEs, the longest possible switch length is 11 switches (the bypass for an in-line scheme is the consists of one switch for each PE bypassed plus one switch at each end; thus the longest bypass in a 10 x 10 array is 11

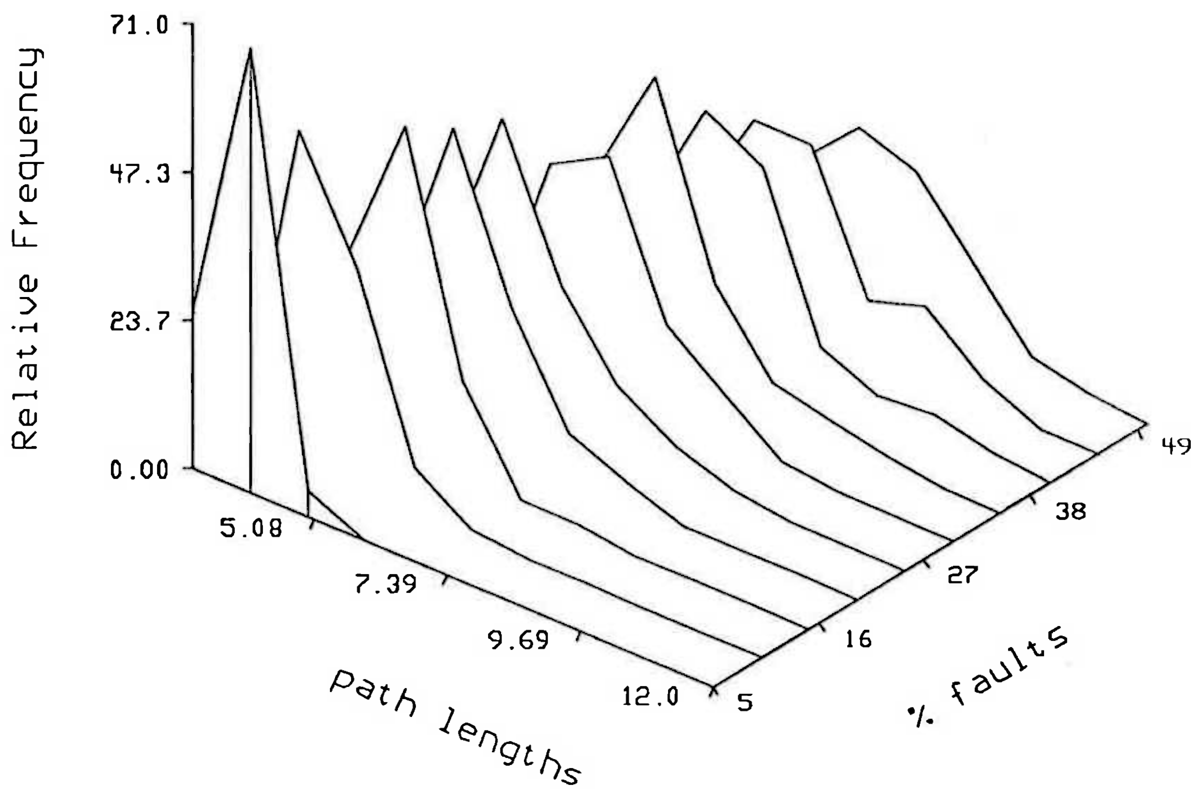


Figure 6.28: Relative frequencies of maximum switch lengths, in terms of number of switches in that path for the in-line rows/GI columns scheme. The raw array consisted of 100 PEs.

switches – 9 bypassed PEs + one switch at each end.) This also corresponded to the longest switched path out of all the samples taken. However if the worst 4% of the reconfigured arrays are rejected then the longest switched path is only 9 switches. This has a less than 4% effect on utilization because the now rejected arrays also contain the fewest PEs. This illustrates the first effect.

If still better performance is required the distribution is flat enough to allow one to use the second effect. By rejecting the worst 18% of arrays the maximum switch length becomes 8. Rejecting 44% of the reconfigured arrays reduces this length to 6. This corresponds to a 23% speed improvement over a maximum length of 9 but only reduces utilization by about 15% (from 45% to 30%.)

The switch lengths about to be presented were determined with the extreme longest paths rejected (*ie.* taking account of the first effect above.) The switch lengths for the in-line rows/GI columns scheme are given in Figure 6.29.

The number of switches required in the longest switched path can be determined readily for other schemes. In the LI/CE scheme the longest switch length contains four switches:– one at each end and two for a CE in the middle. For Hedlund's hierarchical scheme the longest chain is generally eighteen switches, this being the longest possible chain that can be formed in the process of connecting four PEs within a block of twelve. Naturally the case where a whole column is bypassed will produce a deterioration – 26 switches or worse. However such cases are rare if the utilization is at its optimum point on its utilization vs. % PEs faulty plot.

Switch lengths for the all-GI scheme are plotted in Figure 6.30 for the non-clustered case and Figure 6.31 for the clustered fault case. It can be seen from these results, as well as Figure 6.27 showing utilization, that the Frame approach achieves utilizations roughly similar to those achieved by the double channel approach, whilst the switch lengths are shorter than those of the single channel approach. The trade off is however that the Frame scheme uses twice as many wires as the double channel approach.

These all-GI plots were produced for 8×8 (64 PEs) arrays. A result presented in [Leighton & Leiserson, 1985] states that to maintain a constant utilization for channel approach arrays the number of switches in the longest connecting wire must grow at a rate of at least $\Omega(\sqrt{lgN})$ where N is the number of PEs in the array. Given the similarity in complexity between the frame approach and the channel approach it is assumed that this result would also apply to the frame approach.

It will also be assumed that this minimum growth rate can also be applied to the in-line rows schemes, though the rate of growth appears to be faster for PE fault rates above 60%. At these large fault rates most of each row is bypassed and thus the switch

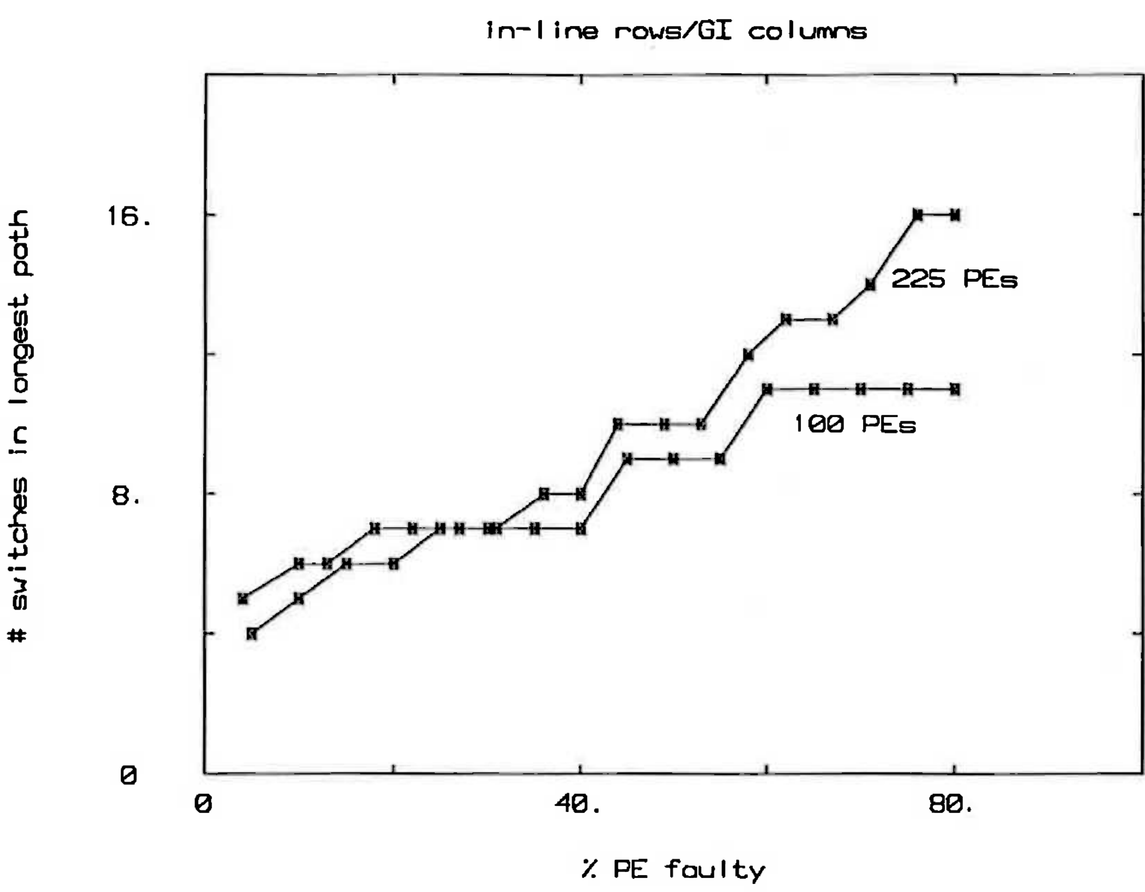


Figure 6.29: Number of switches in in-line rows/GI columns scheme for a number of array sizes.

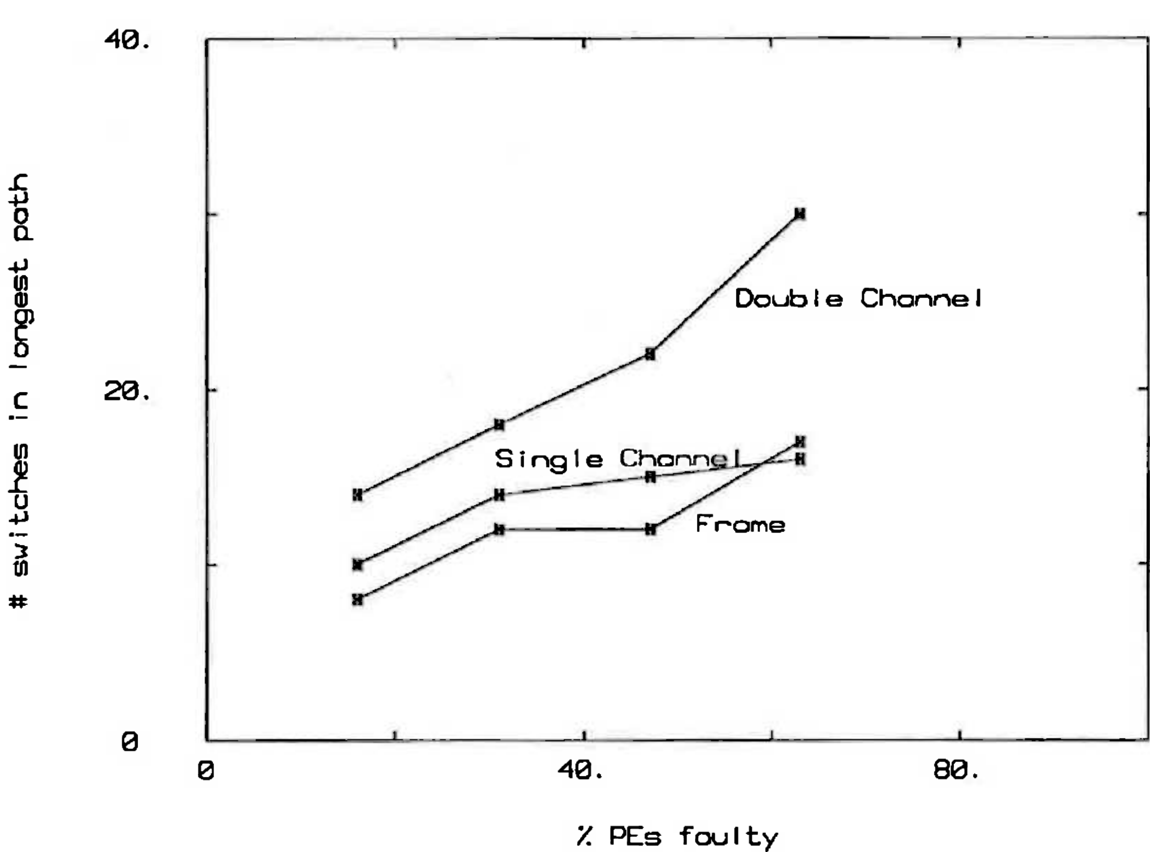


Figure 6.30: Number of switches in longest path for all-GI schemes.

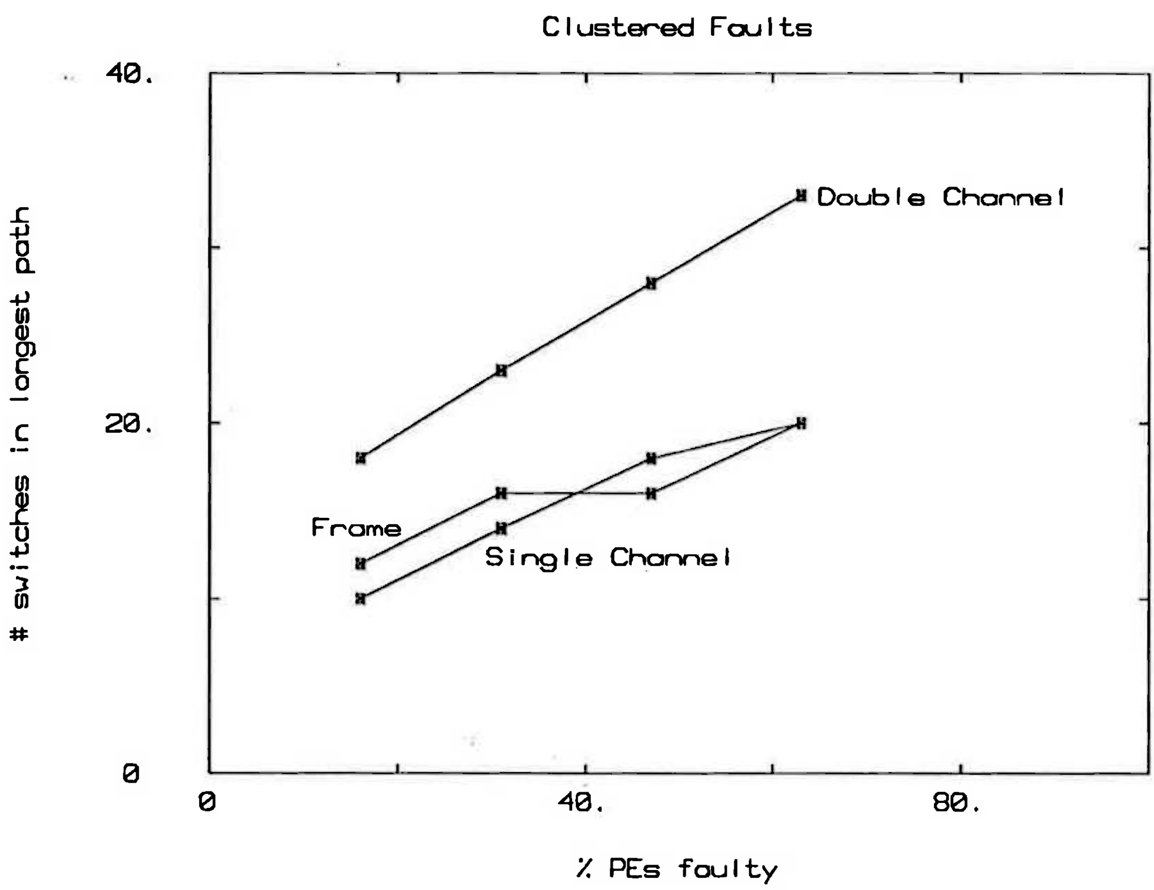


Figure 6.31: Number of switches in longest path for all-GI schemes – clustered faults.

length grows as the row length grows - $O(\sqrt{N})$.

A length of switched interconnect can be reduced to a lumped RC equivalent (see Chapter 5 and Figure 6.32). As the resistance of the metal interconnects is negligible compared with the resistance of the switches, a simple expression for the path delay would be:

$$\tau_d = R_1 C_{total} + \sum_{i=1}^{N_{sw}} \sum_{j=i}^{N_{sw}} R_i C_j \tag{6.8}$$

where R_i is the resistance of a switch, C_j is the combined capacitance of a switch's input and output stages and the metal interconnect, and N_{sw} is the number of switches. R_1 is the driving source resistance and C_{total} is the sum of all the C_j.

As described in Chapter 5, the sizes of the switches can be varied to increase the speed, and in some cases (notably the frame approach) extra drivers can also be inserted. This of course is at the expense of additional switching area overhead.

An approximate optimum can be determined by ignoring R_1 and assuming that the C_j are all equal. Then as the switch size S is increased, R_i decreases as r/S whilst C_j increases as $c + kS$, where r, c and k are constants. Equation 6.8 can then be approximated by

$$\tau_d = N_0 r \frac{c + kS}{S} \tag{6.9}$$

where

$$N_0 = \sum_{i=1}^{N_{sw}} \sum_{j=i}^{N_{sw}} 1 \tag{6.10}$$

The optimum of this expression occurs when

$$S = \frac{1}{2}\left(1 + \sqrt{1 + \frac{4c}{k}}\right). \tag{6.11}$$

For 2.5μm technology,

$$\begin{aligned} c &\approx 2 \times 10^{-16} l_m \text{ Farad} & (6.12) \\ k &\approx 7 \times 10^{-14} \text{ Farad} & (6.13) \\ r &\approx 2k\Omega & (6.14) \end{aligned}$$

where l_m is the length of metal interconnect between switches.

Using these approximations the optimum switch size was determined for different combinations of inter-switch wiring length and number of switches in the path. These optimum speeds are plotted in Figure 6.33. The optimum speed was found to be independent of the inter-switch wiring length. The switch size S associated with the optimum speed however was always found to be 10,000 or more! Fortunately near

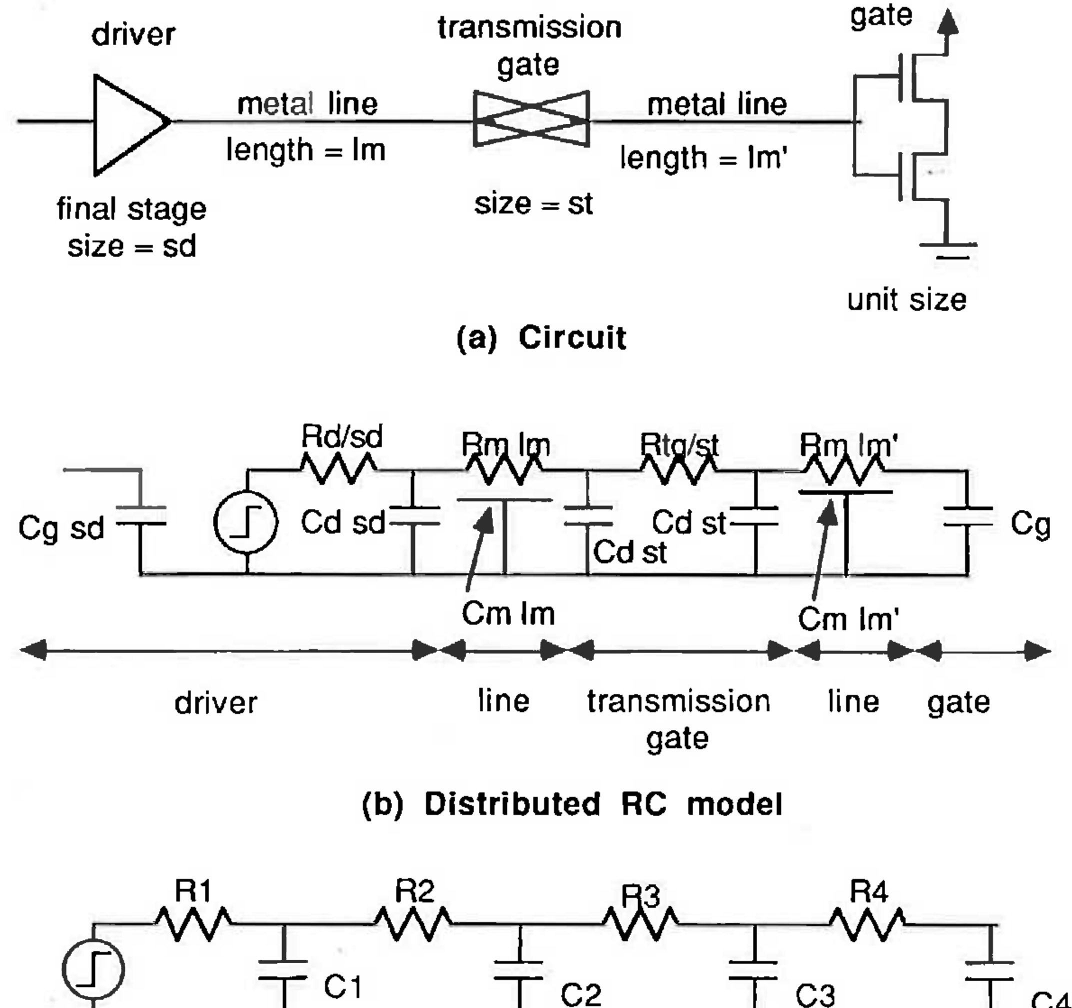


Figure 6.32: A length of reconfigurable interconnect and its equivalent RC model.

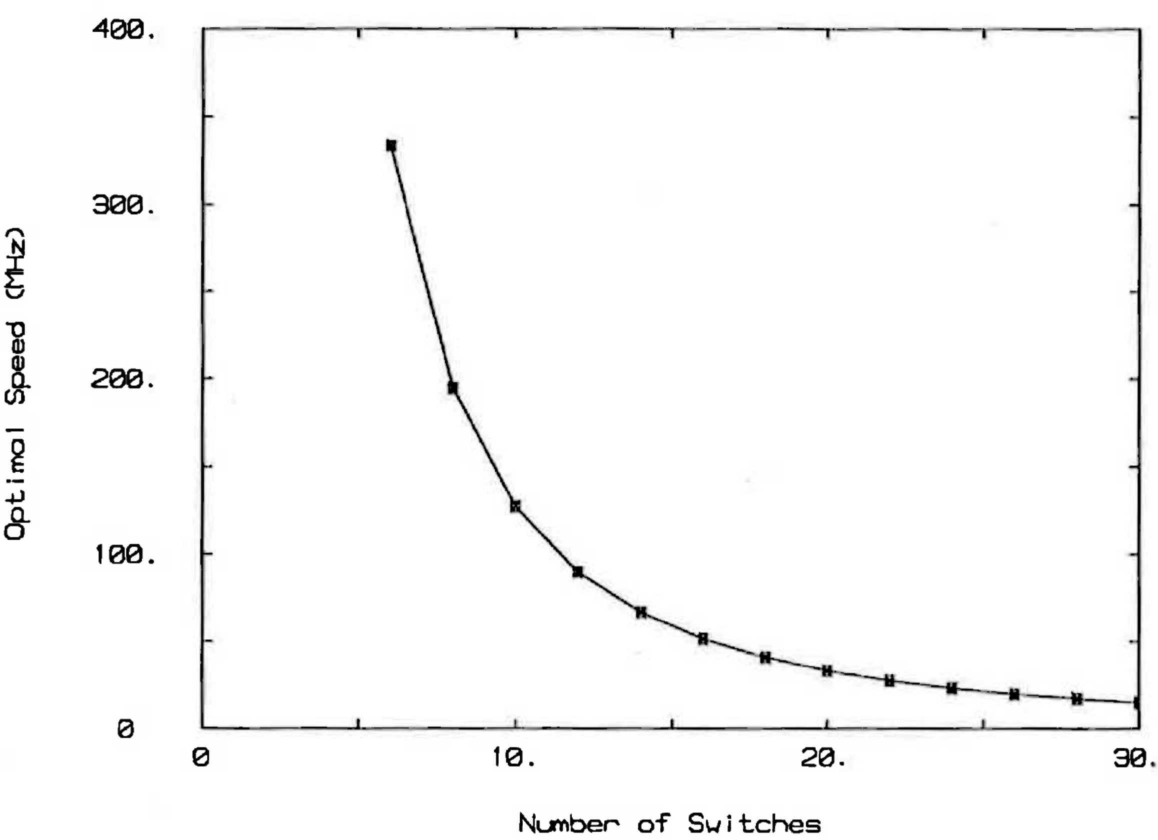


Figure 6.33: Optimum communications speed for different switch numbers and inter-switch wiring distances.

optimum speeds can be achieved using much smaller switch sizes. Figure 6.34 shows the speeds achieved for switch size $S = 10$.

As can be seen in this figure the speed is relatively independent of the inter-switch wiring length. Thus these results can be used as a rough guide for comparing the speeds of different schemes with different maximum switch numbers, the different and varying inter-switch lengths producing only a small variation in communications speed when compared to the number of switches.

Combining the rough calculations presented in Figure 6.33 with the results presented earlier for switch lengths, the relative communications speeds of the different schemes can be calculated. These are presented in Figure 6.35. Here only the number of switches in the longest switched path is accounted for, the differing lengths of interconnect between the switches in different schemes being ignored. (The results given in Figure 6.34 indicate that the error in this assumption is small.) Clearly the in-line rows variants and the frame approach enjoy some advantage when it comes to communications speed.

Of course the speed of a processor array may be limited by the speed of the processor rather than the interconnect. In the technology used to calculate these rough results the fastest processors have a speed of about 80MHz. Special techniques were required to achieve this however and speeds of 40MHz or less are common.

6.4 Overall Comparison of Different Scheme Samples

A summary of the scheme utilizations and optimum speeds, along with their product is given in Table 6.1 for different levels of PE fault rate. ("Cols" means "Columns" in all of these tables.) Care should be taken in drawing conclusions from this table as the speeds are very roughly calculated and the speed of the connected PE will often be slower than the speed of the reconfigured connections. However these figures clearly indicate the ascendancy of the more basic schemes for lower PE fault rates and a preference towards all-GI schemes at higher rates.

The schemes compared in Table 6.1 were chosen because they are representational of the wider variety of schemes available. From the utilization plots some of them are also considered as being the best choice of schemes available or, from the literature, the most popular. The LI columns/in-line rows scheme is perhaps the most popular in the literature, followed closely by the channel approach schemes. The CE/LI scheme

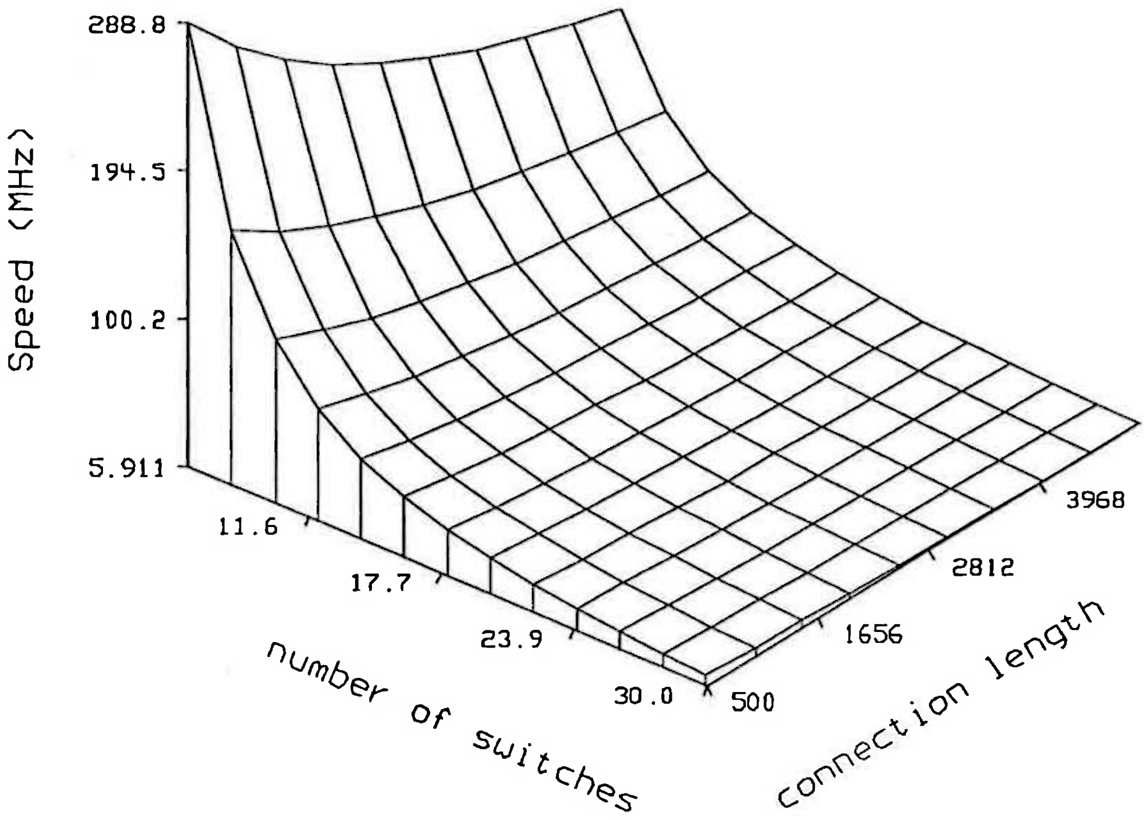


Figure 6.34: Communications speed for different switch numbers and inter-switch wiring distances with the switch size set to $S = 10$.

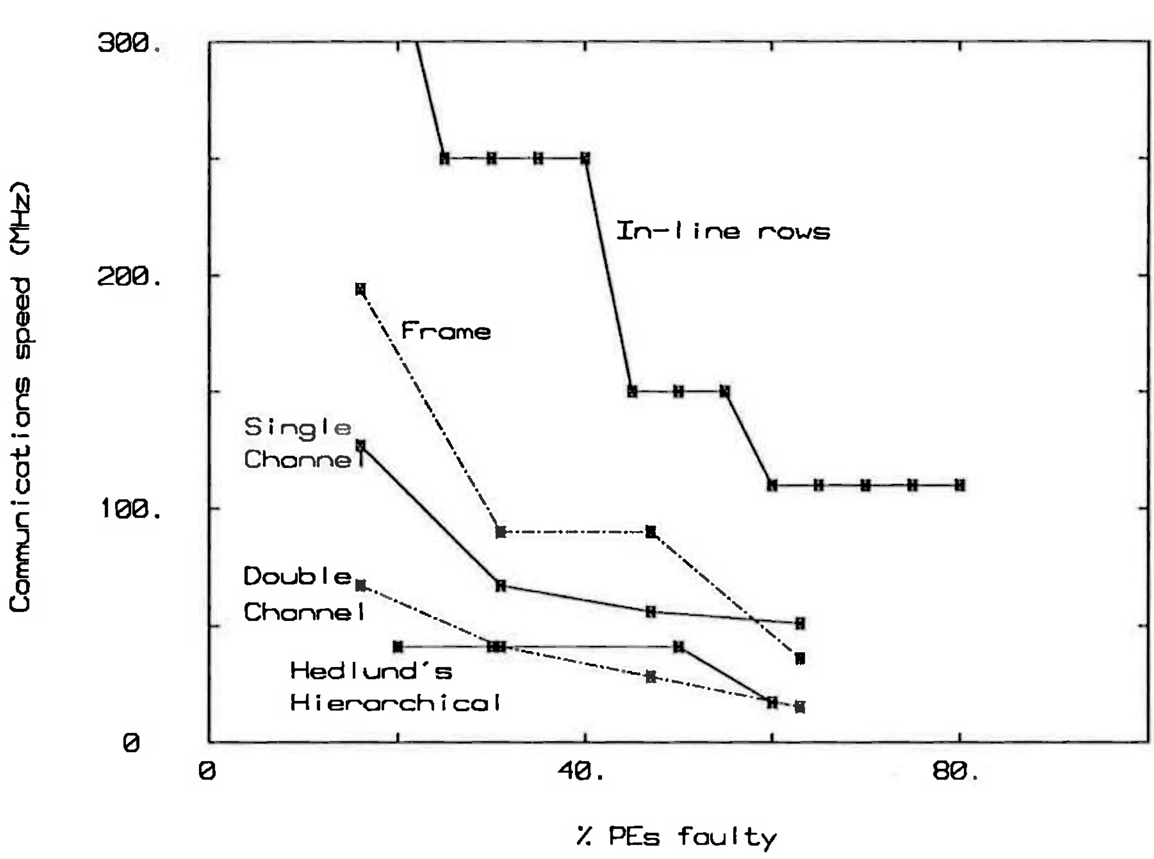


Figure 6.35: Approximate speed comparison of different redundancy approaches.

Scheme:	CE/LI	LI cols/ in-line rows	GI cols/ in-line rows	Hedlund	Single Channel	Double Channel	Frame
20% faulty PEs							
Utilization(%):	44	56	68	76	72	90	92
Speed (MHz):	330	190	190	40	120	60	170
Speed × Util.:	146	106	130	30	86	54	156
40% faulty PEs							
Utilization(%):	13	24	49	52	74	88	86
Speed (MHz):	330	130	130	40	60	30	90
Speed × Util.:	43	31	64	21	44	26	77
60% faulty PEs							
Utilization (%):	2	5	36	53	68	94	82
Speed (MHz):	330	45	45	20	50	20	45
Speed × Util.:	7	2	16	11	34	19	37

Table 6.1: Comparison of Utilization and optimum communications speed at different failure rates.

is included to give a comparison with an all LI scheme. The GI columns/in-line rows and the Frame schemes are included because of their good utilizations.

Clearly however the advantage goes to smaller PEs, with lower fault rates. At a 20% fault rate the frame, CE/LI and GI columns/in-line rows schemes produce the best Speed × Utilization figures. Table 6.1 doesn't tell the whole story however because it doesn't take account of the area overhead, particularly the switch area overhead that would be required to achieve the optimal speeds given.

Table 6.2 summarizes the approximate area overhead impacts of these reconfiguration schemes. The A_{wire} figure indicates the area overhead for each PE in terms of the number of channel units A_c required. One channel unit is the area taken by one bus width of wiring running along one edge of the PE. The PEs are assumed to be square. The A_{switch} column indicates the area of the switch overhead per PE in terms of the number of transmission gate switches times their unit size A_S. Here these two area contributions are considered independent. In actual practice they would be tightly inter-related and the total area overhead may easily be greater than that of these two areas added together. This would be layout dependent. The effect of the wiring and switch area overheads on yield is also discussed in this table. In terms of the yield model presented in Chapter 3, the relevant contributions of this overhead to the module area A_{mod}, wafer kill area A_{kill}, or otherwise is indicated. In this column A_{recon} refers to the total area overhead required for reconfiguration.

The effect of the wiring overheads on yield related areas have only been worked out approximately, though the methodology is based on that presented in Chapter 3. For the CE/LI scheme approximately $\frac{1}{5}$ of the reconfiguration area is actually used by each PE and thus contributes to the wiring area. The expressions for the in-line rows scheme's yield areas are approximations of those presented in Chapter 3. The single channel case is self explanatory. For the double channel scheme approximately just more than $\frac{1}{10}$ of the wiring provided is used. For the frame scheme, typically a fault anywhere in about half a frame prevents that frame from being used, and thus denies access to the enclosed PE.

To make a direct comparison of the effect of wiring overhead on reconfiguration scheme choice, detailed assumptions would have to made about the PE size, communications bus-width, switch size and reconfiguration scheme layout and thus is best only done on a case by case basis. In general it is worth noting that A_c increases at the rate of $\sqrt{A_{PE}}$ where A_{PE} is the PE area and A_S is constant with changing PE size. Thus the larger area overheads of the all-GI schemes, particularly the frame scheme, have a lower yield impact as the PE size increases. Naturally as the PE size increases its fault

Scheme	A_{wire}	A_{switch}	*Wiring impact on yield*
CE/LI	$6A_c$	$14A_S$	A_{mod} increased by approx. $\frac{1}{5}A_{recon}$
LI columns/in-line rows	$4A_c$	$4A_S$	A_{kill} increased by approx. $\frac{1}{4}A_{recon}$, A_{mod} increased by approx. $\frac{1}{6}A_{recon}$
GI columns/in-line rows	$5A_c$	$5A_S$	A_{kill} increased by approx. $\frac{1}{5}A_{recon}$, A_{mod} increased by approx $\frac{1}{5}A_{recon}$
Single Channel	$2A_c$	$12A_S$	Every faulty channel prevents access to two PEs.
Double Channel	$4A_c$	$32A_S$	A_{mod} increased by approx. $\frac{1}{10}A_{recon}$
Frame	$8A_c$	$24A_S$	A_{mod} increased by approx. $\frac{1}{2}A_{recon}$

Table 6.2: Area effects of reconfiguration wiring and switching overhead.

rate increases also.

To sample the choices available to the array designer a few approximate calculations of $E(P)$ under different conditions are given in Table 6.3 for schemes using switches of size=5, and in Table 6.4 for schemes using size=1 switches. All calculations assume that the total available area, excluding that required for power distribution, is 100 mm by 100 mm and that the communicating bus width is four bits with an interwire pitch of $6\mu m$. For a $2.5\mu m$ technology it was calculated that the size=5 switches would occupy an area of approximately $375\mu m^2$ and the size=1 switches would occupy an area of approximately $110\mu m^2$. Using the areas given in Table 6.2 and assuming no layout inefficiencies (*ie.* $A_{wire} + A_{switch}$ forms the complete area overhead) the area overheads per PE fabricated were calculated as follows: Size=5 switches:

$$\text{CE/LI:} \quad 0.144W_{PE} + 21 \text{ mm}^2 \tag{6.15}$$

$$\text{LI columns/in-line rows:} \quad 0.096W_{PE} + 6 \text{ mm}^2 \tag{6.16}$$

$$\text{GI columns/in-line rows:} \quad 0.12W_{PE} + 7.5 \text{ mm}^2 \tag{6.17}$$

$$\text{Single Channel:} \quad 0.048W_{PE} + 18 \text{ mm}^2 \tag{6.18}$$

$$\text{Double Channel:} \quad 0.096W_{PE} + 48 \text{ mm}^2 \tag{6.19}$$

$$\text{Frame:} \quad 0.192W_{PE} + 36 \text{ mm}^2 \tag{6.20}$$

Size=1 switches:

$$\text{CE/LI:} \quad 0.144W_{PE} + 1.54 \text{ mm}^2 \tag{6.21}$$

$$\text{LI columns/in-line rows:} \quad 0.096W_{PE} + 0.44 \text{ mm}^2 \tag{6.22}$$

$$\text{GI columns/in-line rows:} \quad 0.12W_{PE} + 0.55 \text{ mm}^2 \tag{6.23}$$

$$\text{Single Channel:} \quad 0.048W_{PE} + 1.32 \text{ mm}^2 \tag{6.24}$$

$$\text{Double Channel:} \quad 0.096W_{PE} + 3.52 \text{ mm}^2 \tag{6.25}$$

$$\text{Frame:} \quad 0.192W_{PE} + 2.64 \text{ mm}^2 \tag{6.26}$$

where W_{PE} is the length of one side of the PE.

To simplify the calculations $E(P)$ was approximated as

$$E(P) = \text{\% of PEs that are good} \times \text{Utilization} \times \text{Number of PEs fabricated} \tag{6.27}$$

where the number of PEs fabricated was calculated with the wiring area accounted for, and the PE yield was calculated approximately as

$$Y_{PE} = \frac{1}{1 + D_0 A_m} \tag{6.28}$$

Scheme	A_{recon}/PE	N_{total}	$E(P)$	Y_{kill}	$E_{tot}(P)$	$U_{area} = \frac{E(P)A_{PE}}{A_{total}}$
			PE area = 25 sq. mm			
CE/LI	21.7 mm^2	214	9			0.0225
LI cols/in-line rows	6.84 mm^2	318	35	0.06	2	0.0875
GI cols/in-line rows	8.1 mm^2	302	76	0.06	4	0.19
Single Channel	18.24 mm^2	231	47			0.1175
Double Channel	48.5 mm^2	136	63			0.1575
Frame	37 mm^2	161	59			0.1475
			PE area = 50 sq. mm			
CE/LI	22 mm^2	68	0			0.0
LI cols/in-line rows	6.68 mm^2	176	2	0.10	0	0.01
GI cols/in-line rows	8.35 mm^2	171	26	0.10	2	0.13
Single Channel	18.3 mm^2	146	27			0.135
Double Channel	48.7 mm^2	101	36			0.18
Frame	37.4 mm^2	114	30			0.15

Table 6.3: $E(P)$ of 100 mm square arrays with different reconfiguration schemes. $D_0 =$ 0.03 defects/mm^2. *Switch size is 5.*

where A_m is the PE area plus that percentage of the wiring area per PE that impacts A_{mod} as given in Table 6.2. For the single channel approach the contribution of A_{recon} to A_{mod} was taken as $2A_{recon}$. This does not accurately reflect the correct handling of this situation as described in Chapter 3 but will serve as a suitable approximation for our immediate purposes.

The calculations for $E(P)$ were thus determined and are given in Tables 6.3 and 6.4. D_0 was taken as 0.03 defects/mm^2. The last two columns in the tables take account of the contribution of reconfiguration wiring and switching area to the array kill area, as per Table 6.2. Y_{kill} is the yield of arrays that work at all. $E_{tot}(P) = E(P) \times Y_{kill}$. From these results a strong case can be made for making the in-row configuration wiring in the in-row schemes defect tolerant, for example as illustrated in Figure 6.36. The additional overhead that would be introduced by doing this is minimal and the resultant $E(P)$ would be close to those indicated in the $E(P)$ column of these tables.

The area utilization measure (the fraction of the area that is effectively used after reconfiguration) is also given in these tables.

The speed for each situation just described was calculated using the approximate method presented in Section 6.3.3. The number of switches required was determined from the results presented in that section. The switch length for in-line rows schemes was scaled as $\sqrt{ln(N)}/\sqrt{ln(100)}$ over the switch length for the 10×10 array. For the all-

Scheme	A_{recon}/PE	N_{total}	$E(P)$	Y_{kill}	$E_{tot}(P)$	$U_{area} = \frac{E(P)A_{PE}}{A_{total}}$
			PE area = 1 sq. mm			
CE/LI	1.68 mm^2	3726	2891			0.2891
LI cols/in-line rows	0.536 mm^2	6510	5683	0.04	227	0.5683
GI cols/in-line rows	0.67 mm^2	5988	5228	0.04	209	0.5228
Single Channel	1.37 mm^2	4222	4095			0.4095
Double Channel	3.62 mm^2	2166	2101			0.2101
Frame	2.83 mm^2	2610	2427			0.2427
			PE area = 4 sq. mm			
CE/LI	1.83 mm^2	1715	916			0.3664
LI cols/in-line rows	0.632 mm^2	2159	1345	0.09	121	0.538
GI cols/in-line rows	0.79 mm^2	2087	1481	0.09	133	0.5924
Single Channel	1.41 mm^2	1846	1164			0.4656
Double Channel	3.71 mm^2	1297	1039			0.4156
Frame	3.02 mm^2	1423	1003			0.4012
			PE area = 9 sq. mm			
CE/LI	1.97 mm^2	911	284			0.2556
In cols/in-line rows	0.728 mm^2	1028	531	0.15	80	0.4779
GI cols/in-line rows	0.91 mm^2	1009	628	0.15	94	0.5652
Single Channel	1.46 mm^2	955	537			0.4833
Double Channel	3.81 mm^2	781	542			0.4878
Frame	3.22 mm^2	818	547			0.4923
			PE area = 25 sq. mm			
CE/LI	2.26 mm^2	367	28			0.07
LI cols/in-line rows	0.92 mm^2	385	50	0.27	13	0.125
GI cols/in-line rows	1.15 mm^2	382	135	0.27	36	0.3375
Single Channel	1.56 mm^2	376	200			0.5
Double Channel	4.0 mm^2	344	232			0.58
Frame	3.6 mm^2	350	224			0.56

Table 6.4: $E(P)$ of 100 mm square arrays with different reconfiguration schemes. $D_0 =$ 0.03 defects/mm^2. *Switch size is 1.*

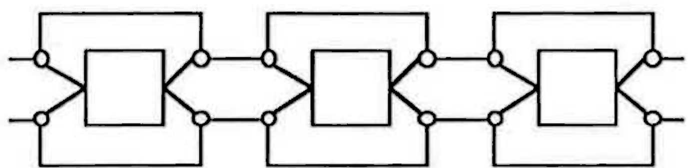


Figure 6.36: Improving in-line row schemes' effectiveness by making the row reconfiguration wiring itself defect tolerant.

GI schemes the switch length was scaled as $\sqrt{ln(N)}/\sqrt{ln(64)}$ as a multiple of the switch length results for the 8×8 array presented in Section 6.3.3. The schemes presented have a varying distance between switches. Instead of taking account of these varying distances explicitly an average inter switch distance was assumed. These distances were: $\frac{3}{5}W_{PE}$ for the CE/LI scheme; W_{PE} for the in-line rows schemes; $W_{PE}/2$ for the single channel scheme; $W_{PE}/4$ for the double channel scheme; and $W_{PE}/3$ for the Frame scheme. The calculated speeds, along with $E(P)$, and $E(P) \times A_{PE} \times$ speed are presented in Table 6.5 for size=5 switched schemes and Tables 6.6 and 6.7 for size=1 switched schemes. The A_{total} factor used in Equation 6.2 is constant and is thus left out in these tables.

The speed of the interconnect may not always limit the speed of the array. For example, the PEs may be slower, or it may be possible to allow communications to take place over several clock cycles. Extra rows have been added to these tables to reflect this. The rows labeled "(PE speed)" reflect the situation where the PE is slower than the interconnect. Two situations are given :- a very fast PE (for the generation of technology being represented) of speed 80MHz, and a typically fast PE of speed 50MHz. It may also be possible for the array to operate at speeds faster than the interconnect allows under single clock phase communications. If multiple clock cycle or asynchronous communications are possible then the array speed may be faster. This is represented in the rows labeled "(speed-up)".

General conclusions should only be drawn from the trends presented in these tables with caution, as the results are drawn from specific samples only and are also rough calculations. The sources of error in these calculations are:

1. The use of an approximate equation for determining E(P).

   By calculating the the percentage of PEs that are faulty using a simple yield expression the beneficial effects that defect clustering give to the yields of large area devices are ignored. The difference in the number of defects per wafer between wafers would tend to benefit $E(P)$ as low defect wafers would have a higher utilization than higher defect wafers. As this is true for the CE/LI and in-line row schemes these would benefit more from this fact than the all-GI schemes.

2. The difference in utilization with array size is ignored.

   Generally utilization decreases with array size. As the all-GI arrays have fewer PEs than the other schemes then these schemes would benefit more from this, particularly for small PE sizes. In addition the utilization of the CE/LI

Scheme	*Speed (MHz)*	$E(P)$	$E(P) \times A_{PE} \times$ Speed
$A_{PE} = 25$ mm^2			
CE/LI	272	9	61,200
(PE limit)	50		11,250
LI cols/in-line rows	31	35	27,125
(speed-up)	50		43,750
GI cols/in-line rows	31	76	58,900
(speed-up)	50		95,000
Single Channel	20	47	23,500
(speed-up)	50		58,750
Double Channel	16	63	25,200
(speed-up)	50		78,750
Frame	40	59	59,000
(speed-up)	50		73,750
$A_{PE} = 50$ mm^2			
CE/LI	189	0	0
LI cols/in-line rows	10	2	10,00
(speed-up)	50		5,000
GI cols/in-line rows	10	26	13,000
(speed-up)	50		65,000
Single Channel	16	27	21,600
(speed-up)	50		67,500
Double Channel	9	36	16,000
(speed-up)	50		90,000
Frame	20	30	30,000
(speed-up)	50		75,000

Table 6.5: Comparison of different schemes with switch size=5.

Scheme	*Speed (MHz)*	$E(P)$	$E(P) \times A_{PE} \times$ Speed
$A_{PE} = 1$ mm^2			
CE/LI	305	2891	881,755
(PE limit)	80		231,120
(PE limit)	50		144,550
LI cols/in-line rows	250	5683	1,420,750
(PE limit)	80		454,640
(PE limit)	50		284,150
GI cols/in-line rows	250	5228	1,307,000
(PE limit)	80		418,240
(PE limit)	50		261,400
Single Channel	305	4095	1,249,000
(PE limit)	80		327,600
(PE limit)	50		204,750
Double Channel	160	2101	336,160
(PE limit)	80		168,000
(PE limit)	50		105,000
Frame	150	2427	364,050
(PE limit)	80		194,160
(PE limit)	50		121,350
$A_{PE} = 4$ mm^2			
CE/LI	190	916	696,160
(PE limit)	80		293,120
(PE limit)	50		183,200
LI cols/in-line rows	50	1345	269,000
GI cols/in-line rows	50	1481	296,200
Single Channel	20	1164	93,120
(speed-up)	50		232,800
Double Channel	12	1039	41,560
(speed-up)	50		207,800
Frame	40	1003	100,300
(speed-up)	50		200,600

Table 6.6: Comparison of different schemes with switch size= 1.

Scheme	*Speed (MHz)*	$E(P)$	$E(P) \times A_{PE} \times$ Speed
$A_{PE} = 9$ mm^2			
CE/LI	108	284	276,000
(PE limit)	80		204,480
(PE limit)	50		127,800
LI cols/in-line rows	25	531	119,475
(speed-up)	50		238,950
GI cols/in-line rows	25	628	141,300
(speed-up)	50		282,600
Single Channel	16	537	77,328
(speed-up)	50		241,650
Double Channel	18	628	101,736
(speed-up)	50		282,600
Frame	31	547	152,613
(speed-up)	50		246,150
$A_{PE} = 25$ mm^2			
CE/LI	89	28	62,300
(PE limit)	50		35,000
LI cols/in-line rows	7	50	8,750
(speed-up)	50		62,500
GI cols/in-line rows	7	135	23,625
(speed-up)	50		168,750
Single Channel	6	200	30,000
(speed-up)	50		250,000
Double Channel	3	232	17,400
(speed-up)	50		290,000
Frame	6	224	33,600
(speed-up)	50		280,000

Table 6.7: Comparison of different schemes with switch size=1.

scheme would be severely down on the numbers assumed as the number of PEs is increased. This is not reflected in the tables.

3. Approximate method of calculating speed.

   Experience from the ADVICE simulations discussed in Chapter 5 indicate that the speeds calculated here are high. They would be expected to be uniformly high. So this affects the speed figures by only a constant factor. However, the attendant scaling down in the $E(P) \times A_{PE} \times$ Speed figures would tend to make these figures favour the slower schemes more.

4. The difference in switch length with array size.

   An attempt was made to adjust the switch length on the basis of array size. Though the factors used have a logical basis, the accuracy of the resultant switch lengths used is difficult to ascertain.

Nevertheless certain trends are evident and are worth commenting upon:

Every scheme had a $E(P) \times A_{PE} \times$ Speed figure calculated for it for at a speed of 50MHz. These figures are thus proportional to the area utilization measure mentioned in Section 6.1 and calculated in Tables 6.3 and 6.4. The best area utilization is 0.59 for the the GI columns/in-line rows scheme with $A_{PE} = 4\text{mm}^2$. The next best are 0.58 for the double channel scheme with $A_{PE} = 25\text{mm}^2$ (switch size = 1), and 0.56 for the frame scheme at $A_{PE} = 25\text{mm}^2$, the GI columns/in-line rows scheme for $A_{PE} = 9\text{mm}^2$ and the LI columns/in-line rows scheme at $A_{PE} = 1\text{mm}^2$. This ordering of results reflects two factors:

1. The reconfiguration wiring overhead is quite large relative to the PE areas for $A_{PE} = 1$ mm^2 thus making the area utilization for this size PE poor. Larger PEs were required to efficiently utilize the area.

2. The GI columns/in-line rows scheme produced better area utilization for lower PE failure rates, while the double channel and frame approaches produced better utilization for higher PE failure rates. The approximate PE failure rates for the cases illustrated were: 3% for $A_{PE} = 1$ mm^2; 10% for $A_{PE} = 4$ mm^2; 20% for $A_{PE} = 9$ mm^2; 45% for $A_{PE} = 25$ mm^2;and 65% for $A_{PE} = 50$ mm^2. This merely reflects the higher utilizations achievable by the all-GI schemes at higher PE failure rates. At these higher rates the relative area overheads of the reconfiguration schemes are reduced also.

Naturally the area utilization figures for the switch size=5 results are much lower due to the larger area overhead of the bigger switches.

In the size= 1 case examples the area utilization was maintainable across a wide range of PE sizes. This is slightly surprising as the area utilization may have been expected to fall with increasing PE fault rate. This result arose because of the high utilizations achievable with the all-GI schemes. As the PE size grew their relative area overhead diminished and these schemes could be used productively. This is in slight contrast with the partitioning result of Chapter 3 which effectively stated that functional utilization increased with smaller PE sizes. This latter result assumed a constant utilization with PE size and fault rate, thus making it a result more suitably applied to linear arrays or any other structure that maintains a fairly constant utilization.

As a comparison consider normally diced chips. The combined width of two sets of pads and the inter die sawing kerf varies between 1mm and 1.5mm. Combined with the yields mentioned above this would produce area utilization measures of $U_{area} = 0.29$ for $A_{PE} = 4\text{mm}^2$, $U_{area} = 0.35$ for $A_{PE} = 9\text{mm}^2$, and $U_{area} = 0.33$ for $A_{PE} = 25\text{mm}^2$. Obviously the area utilizations achieved through WSI mesh arrays are often much higher than that achieved through normal chip production techniques.

The area utilizations that can be achieved using linear arrays can be much higher though. As mentioned in Chapter 4, linear arrays can achieve 100% utilization for a lower area overhead than any of the schemes here. With these conditions area utilizations of 0.8 or better are easily possible.

Comparison of the $E(P) \times A_{PE} \times$ Speed factors, taking achievable communications speed into account, shows the LI columns/in-line rows scheme for $A_{PE} = 1\ \text{mm}^2$ to be the best scheme. However an array speed of 250MHz is unrealizable so the lower speed figures should be looked at. Considering this, the GI columns/in-line rows scheme for $A_{PE} = 4\ \text{mm}^2$ again achieves the best result, with the LI columns/in-line rows scheme for $A_{PE} = 1\text{mm}^2$ coming a close second. If larger PEs are required, or the other schemes used, then the results achievable are much worse, at best half of the GI columns/in-line rows for $A_{PE} = 4\text{mm}^2$ result. As the PEs get larger and/or the all-GI schemes are used, the number of switches in the connected paths increases slowly and the speed of these paths decreases dramatically, thus producing this result. However it should be stressed that the speed calculations presented in these tables are perhaps the least accurate of all the factors considered.

For PEs larger than 4mm^2 speeds of a maximum of 50MHz are most likely. For these PE areas the frame scheme produces the best result for $A_{PE} = 9\text{mm}^2$ and a close second best result for $A_{PE} = 25\text{mm}^2$. The single channel approach also produces a

good result for $A_{PE} = 25\text{mm}^2$.

It is worth noting however that out of the three all-GI schemes the Frame scheme produces $E(P) \times \text{Area}_{PE} \times$ Speed factors that are always better than the other two, due to its combination of high utilization and a relatively low number of switches. The cost comes in the form of at least twice as many wires as required by the other two all-GI schemes. In the reconfiguration area overhead equations presented earlier in this section more of the reconfiguration area overhead came from the switches than from the wires. All of this suggests that the best reconfiguration approach would be one that maintains a high utilization (60% – 80%) whilst minimizing the number of switches, both in terms of their total number but also in terms of the number in a switched path. The number of wires required by the scheme does not have the same impact. The limited utilizations achieved by the CE/LI and the in-line row approaches indicates that such a scheme must retain an all-GI nature in order to achieve this level of utilization.

Comparison of Tables 6.5 and 6.7 indicate that the $E(P) \times A_{PE} \times$ speed factors get worse as the transistor sizes are scaled for improved speed. This arises because the area increase required by these larger transistors is greater than the speed increase gained. This larger reconfiguration area also negatively impacts PE failure rates, through its contribution to A_{mod} in the yield equation. Note however that the speed figures for the frame scheme do not include the beneficial effects of intermediate drivers, as discussed in Chapter 5. There it was shown that the introduction of these drivers increased the speed by about 30%. The area overhead was increased to a little more than that required by the double channel approach. Assume that the extra area overhead reduces N_{total} to 95% of those of the double channel approach, as given in Table 6.3. Then $E(P)$ becomes 47 for $A_{PE} = 25\text{mm}^2$, and 25 for $A_{PE} = 50\text{mm}^2$ and $E(P) \times A_{PE} \times$ Speed becomes 30,500 for $A_{PE} = 25\text{mm}^2$, and 32,500 for $A_{PE} = 50\text{mm}^2$. These represent about an 7% improvement on the results in Table 6.5, which unfortunately still does not bring it up to the levels achievable fore size=1 switches.

6.5 Improved Schemes

Improving on the area utilization results of the previous section would be difficult. However many of the PE speeds could use a substantial improvement, particularly for large PE sizes. The best way to achieve this is by decreasing the switch lengths of the connected paths, even if it means an increase in the switch and wiring area overhead.

Going by the tables above, the GI columns/in-line rows scheme is a reasonably

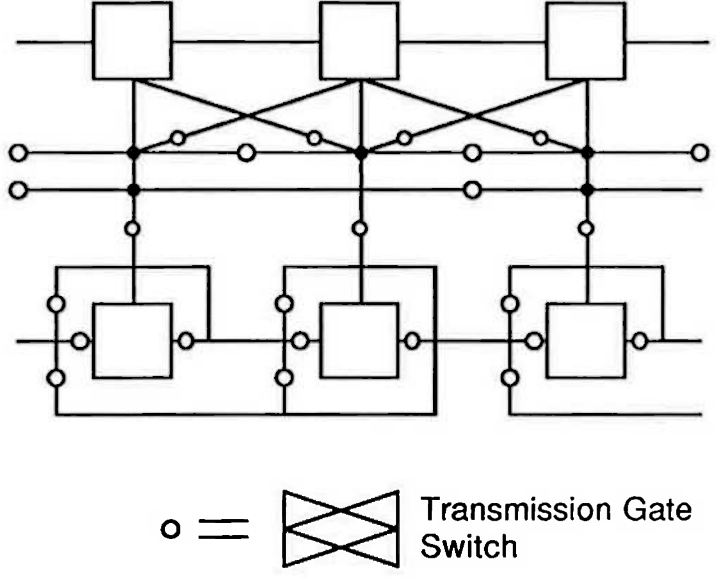


Figure 6.37: GI columns/in-line rows scheme modification to improve speed.

all round successful scheme, as long as the in-line rows connections are made defect tolerant. As this requires extra bypass lines anyway, it should be possible to use fewer switches in these additional bypass lines so as to reduce the total number of switches in any connection. If this is done to the horizontal bypass, then it will also have to done to the vertical column steering lines, as illustrated in Figure 6.37. Doing this should approximately double the speed of the array, whilst only slightly increasing the area overhead and slightly improving the utilization.

The all-GI approaches were successful for larger PE sizes and failure rates. However the high number of switches in any switched path seriously affected system speeds. One suggestion that can be used to increase this speed is illustrated, for the vertical connections only, in Figure 6.38. The area overhead is eight wires and 32 switches per PE (the corner connection essentially comes for free in any layout) which is higher than the overhead for the double channel approach. Its utilization would be similar to those achievable by the double channel approach and thus its area utilization measure would be similar also. However its speed will be up to 50% faster than the speeds achievable by the single channel and frame approaches, making its $E(P) \times A_{PE} \times$ Speed figures better than any other all-GI approach.

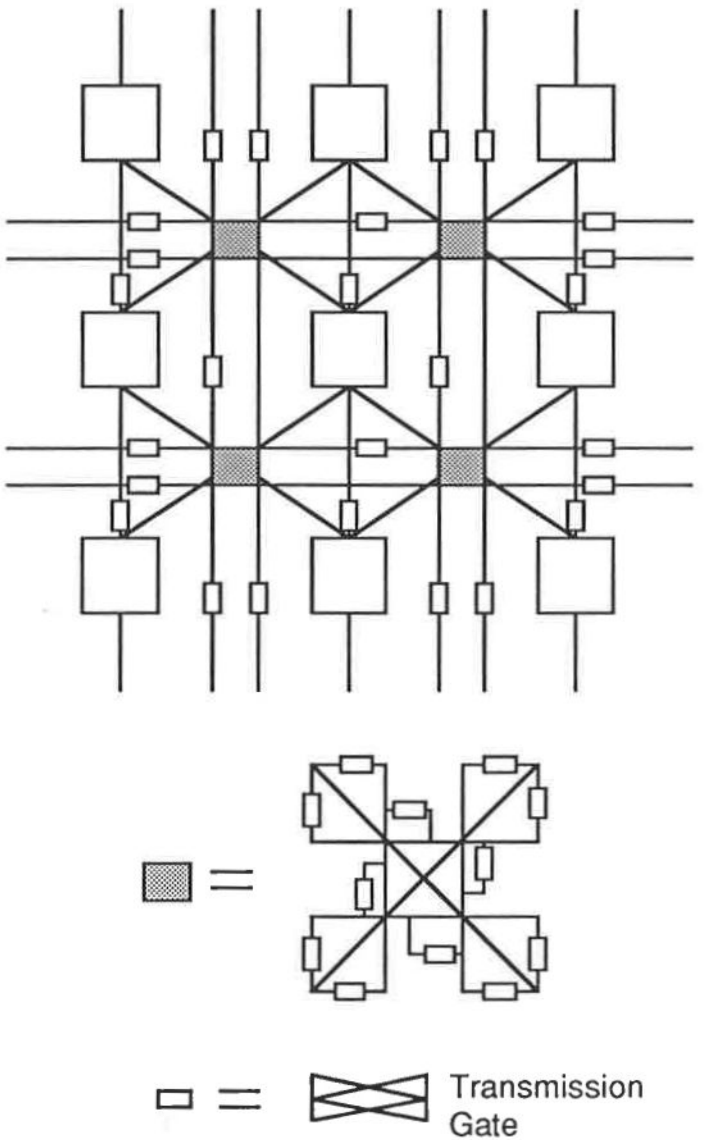


Figure 6.38: Vertical connections for an increased speed all-GI scheme. The horizontal connections are identical in nature.

Chapter 7

Conclusions

7.1 Defect Tolerance and Advanced Packaging

The main aim of fault tolerant VLSI is to increase the current size of economically viable integrated circuits through the use of defect tolerant structures. Wafer Scale Integration (WSI) is the extreme of this process. However in many respects WSI is as much a statement in packaging as in chip size. This is because, with current defect levels, WSI is best used to implement extremely regular arrays. Thus no new architectures can arise by moving to WSI, just a new means of packaging old ones.

A number of potential advantages can be realized in this medium. These are increased reliability, increased speed, increased packaging density, decreased cost, and decreased power consumption.

Another means of obtaining several of these improvements, without the constraints introduced on architecture is by using Hybrid WSI. By packaging VLSI and other parts as a silicon on silicon wafer it is anticipated that improvements can be made in speed and power consumption, and hopefully cost and reliability, at least when compared with other high density packaging techniques. Introducing WSI and HWSI packaging media also allows one to consider the prospect of three dimensional packaging, with a stack of wafers being interconnected through their interiors.

Achieving the final objective of either defect tolerant VLSI or full WSI requires the satisfaction of many enabling objectives:

1. A yield model must be developed that is suitable for describing these media;
2. A mechanism must be introduced allowing the reconfiguration of the parts;
3. A testing procedure must be established that achieves a high fault coverage of those circuits for which spares have been provided; and

4. Novel solutions may be required for a number of more routine problems, such as the satisfactory distribution of power and clocks on the wafer surface, the removal of heat if the power density is high, and possibly the integration of such a high pin count "chip" into a system.

In this thesis considerable emphasis was given to presenting solutions that lead to the achievement of the first two enablers above. The last two points were not given detailed consideration, but were at least addressed for the WSI and HWSI parts that were designed and produced to serve as examples of such systems.

7.2 Yield Modeling for Fault Tolerant VLSI and WSI

How does one decide when one should use defect tolerance and what methods should be applied? These decisions can only be made with the aid of accurate and useful yield models.

Application of defect tolerance to chips is approached in this thesis through a specific and a generalized case study. In the specific case study, a VLSI chip was analyzed to determine where defect tolerance can effectively be applied. Despite the existence on this chip of a number of candidate structures that could have used defect tolerance, only one structure was able to benefit from these techniques.

The generalized case study was tackled with the intention of producing a rule of thumb about what level of granularity is required before defect tolerance can be usefully applied. It was found that the percentage of chip area devoted to an array structure was a more important criterion for the suitability of defect tolerance than the actual granularity of that structure. A broad range of examples were evaluated. From these it was found that 20% – 30% of a VLSI or ULSI chip should be covered by a regular structure before consideration should be given to applying defect tolerance techniques to that structure. Given the increasing interest in using defect tolerance for ULSI chip yield improvement this is an important limitation, precluding many architectures from the potential benefits, but also indicating many more that could benefit.

A yield model is presented for the analysis of fault-tolerant, particularly wafer-scale, arrays. Two important extensions are made to the model. Firstly it was reformulated so as to calculate the expected number of PEs $E(P)$ that can be connected into a wafer-scale array. This form is often more relevant to the WSI case, is

more useful for comparison and can be easier to calculate.

Secondly a methodology for accounting for wiring defects in a wafer-scale array has also been presented. It was shown that this accounting can change conclusions reached about the relative worth of various redundancy schemes. It is a significant improvement on the previous approaches of either assuming the interconnect to be defect free, or assuming that any interconnect fault results in array failure.

The relationship between granularity and yield is discussed for wafer-scale arrays. This relationship introduces new complexities into making architectural decisions about the processing elements that go into the array. An example is given by considering the possible benefits that can be gained by partitioning a proposed array architecture into smaller elements. In non-fault tolerant applications, array element architectures can be evaluated by using a Area $\times$ Time metric. For wafer-scale arrays a Area $\times$ Time $\times$ Yield metric is suggested as being more appropriate.

7.3 Making WSI and HWSI Parts Using the Frame Scheme

The frame scheme for reconfiguring mesh arrays demonstrates a particular trade off in reconfiguration schemes and as such offers a number of advantages. In particular it offers a number of advantages over the channel approach to reconfiguring mesh arrays for a lot of cases.

It maintains as high a utilization as any similar scheme, whilst achieving a higher communications speed. A high speed is reached by two mechanisms. Firstly, fewer switches are required for any PE to PE connection, and secondly, the nature of the scheme allows intermediate drivers to be used. The trade off is that more area overhead is required to achieve the same level of utilization.

The "frame scheme" is so named because reconfiguration is achieved through intimate association of a frame with each PE. This is in contrast with most reconfiguration schemes that logically separate the PEs from the interconnect. This means that intermediate drivers can be placed in such a way as to increase the communications speed as compared with similar non frame approaches. It also means that control of a group of switches can be centralized on each frame. This reduces the programming overhead of the array. For non-frame approaches, if no distributed algorithm for control exists, then each switch must be individually programmed. For the frame approach however the possible switch configurations are known in advance, meaning

that the frame can be programmed by selecting the appropriate configuration rather than selecting the state of each individual switch. A decoder is then applied to the configuration choice to actually program the switches.

WSI and HWSI examples of the frame scheme were fabricated using an FIR filter as the processing element. Both were implemented on the same wafer, with the monolithic WSI array occupying one half of the wafer and the HWSI array occupying the other half. Difficulties were experienced that arose both from the size of the project in terms of the number of transistors and also the particular requirements of a WSI project. With such a large number of transistors present the CAD tools were stretched to their limits. Circuit extraction would have taken over several days of computer time if the whole circuit was extracted but due to its regular nature only partial extraction was required. Similarly only switch level simulation was possible. The projects were constructed by laying out the parts symbolically, compacting them, and then assembling them with a channel router. This high level of automation resulted in a loss of control over the layout and thus a somewhat inefficient layout.

7.4 Comparison Studies of Fault Tolerant Mesh Arrays

With an average of about 100 faults per wafer a large number of spares are required to achieve Wafer-Scale Intergration. This level of sparing can be best achieved by limiting the WSI architecture to an array architecture. The spares are best used by just fabricating an array of processing elements together with a reconfigurable interconnect, so that good processing elements only are connected. How does one decide what processing element architecture and reconfigurable interconnect approach should be used? The answers are actually interdependent.

If the target application gives a linear speed-up in solution time with the addition of processors then the best combination of PE design and scheme is the one which maximizes

$$\frac{E(P) \times \text{Area}_{PE} \times \text{speed}}{\text{Area}_{total}}. \tag{7.1}$$

where the factor $\frac{E(P) \times \text{Area}_{PE}}{\text{Area}_{total}}$ is the area utilization measure, as defined by Koren. The speed would be constrained either by the PE or the interconnect.

If the target application does not give a linear speed-up with the addition of processors then the PE design would be constrained to one that achieved the maximum

functionality, and the interconnect design would be one that achieved maximum area utilization whilst giving a sufficient speed.

The array architecture is also a critical factor. Linear arrays are far more suited to WSI than any other architecture. The examples used in Chapter 6 showed that linear arrays can achieve at least a 1.5 times better area utilization than mesh arrays and diced and packaged chips. However there are many applications where the communications demand necessitates the use of the more richly interconnected mesh array rather than a linear array. Mesh arrays can also achieve better area utilization than diced and packaged chips.

In the case examples evaluated, the area utilization was maintained across a wide range of PE sizes. Thus, for the range of examples considered, the fact that a PE will be used in a defect tolerant array can be ignored when considering the design of that PE. This ignores the detrimental effects on speed that large PEs and higher PE failure rates can have.

Speed is accounted for by considering the E(P)×Area$_{PE}$×Speed metric. On this basis, out of the examples evaluated, the best reconfiguration schemes were the LI columns/in-line rows scheme for the smallest PEs, the GI columns/in-line rows scheme for medium size PEs and the double channel and frame followed closely by the single channel for large PEs. However these results don't consider either the possibility of the PE limiting the speed or the possibility of using a mechanism to effectively increase the speed of the interconnect. Techniques to do this include:

1. Use of schemes with fewer switches between PEs.

   At the end of Chapter 6 some such schemes were suggested. Generally more area overhead is required to do this. Modest gains in speed and $E(P) \times A_{PE} \times$ speed can be achieved though.

2. Use "hard" (fast) switches, such as laser fuses and joins.

   The interconnect configured for the RVLSI program normally achieves communications speed faster than 20MHz, though special process steps are required for these joins. Higher resistance joins can also be formed from structures available on a normal CMOS fabrication line. These joins are still faster however than CMOS switches of the same size and thus can be profitably used instead of these switches in the reconfiguration schemes investigated in Chapter 6. Effort should be made to minimize the number of joins required, particularly the number of joins required in a single path, for speed, area and economy reasons.

3. Use of larger faster switches and intermediate drivers.

    These can achieve a useful increase in speed only at the expense of a significant decrease in area utilization.

4. Use of a BiCMOS process to form bipolar switches.

    This sounds very promising (achieving ECL switch speeds with CMOS density) but without process details it is difficult to evaluate.

5. Use of intermediate latches to pipeline the interconnect.

    This assumes that the extra communication latency is not a problem. However it seems a very promising solution, at the cost of the extra area required for the bidirectional latches. A problem that has to be solved for mesh arrays is that of synchronization :– either by equating all the pipeline lengths, or by accounting for their differing lengths.

All of the schemes considered in the evaluated examples have a mainly GI nature (General Interconnect – connections through a general switched network.) Schemes with a mainly LI (Local Interconnect – point to point only) nature achieve too low a utilization with too much overhead to be universally successful.

One option with the all GI schemes, is to configure them using a hierarchical algorithm. This greatly simplifies the complexity of reconfiguring these types of arrays. A hierarchy is achieved by grouping the PEs into blocks and applying different reconfiguration techniques inside and outside of these blocks. This can be done using one homogeneous all GI structure or by using different switch structures inside and outside of the blocks. The former is preferred not only because it is easier to design, but also because a flexible hierarchical scheme can be applied to it to improve the utilization. Such a scheme is suggested, and shown to be superior, in Chapter 6. However even with these enhancements hierarchical schemes return up to 20% – 30% lower utilizations when compared with tackling the full complexity of the reconfiguration problem.

As a result of the examples evaluated in Chapter 6 suggestions can be made as to how to produce better mesh reconfiguration schemes. From these examples it was found that:

- The number of switches in a switched path dramatically impacts the speed;
- The area overhead of the switches is much higher than the area overhead of the wires;

- Any reconfiguration area that contributes to the chip kill area can lead to a drastically reduce overall yield.

With these points in mind a couple of improved schemes are suggested at the end of Chapter 6. These are an adaptation of the GI columns/in-line rows scheme, and an adaptation of the frame scheme.

By using the correct mix of techniques described above it is quite possible to produce WSI mesh arrays that achieve better silicon utilization than their diced chip counterparts, as well as having lower packaging costs, higher reliability and a smaller system volume. Speed can be maintained or even improved on the PCB mounted equivalent at the same time, depending on the application.

7.5 Future Work

WSI can bring a lot of advantages to array based systems. The techniques required to realize these advantages are to the most part fairly well understood. The main effort that is needed now is the design and construction of practical systems. In the process of solving these real problems the scheme parameters determined in Chapter 6 can be further refined. This would allow a more accurate comparison of scheme approaches. If necessary, new reconfiguration schemes could be developed for the application along the lines suggested towards the end of Chapter 6. Together this work will quickly produce a library of knowledge about various schemes indicating their area overheads and speed parameters. Also details of the yield model, accounting for wiring, can be confirmed and extended. This library of knowledge can be used by future designers to quickly, and decidely, design WSI products.

Not all of the techniques applicable are completely understood yet. The application of BiCMOS to WSI is an area yet to be explored at all. The production of pipelined reconfiguration schemes to increase speed at the expense of latency is a very promising idea that can use refinement. These, and other, speed gaining techniques need to be well understood if large PE mesh arrays are to be produced.

Appendix A

Delay Models

The full RC and modified RC delay equations for each of these cases are given below.

Case 1 (Drive at source only)

The delay tau_{d1} for the simple RC model is

$$
\begin{aligned}
\tau_{d1} &= (stages - 1)(R_G C_D + 4 R_G C_G) + R_G C_D \\
&\quad + 7 R_G C_M / sd + 22 R_G C_D st / sd \\
&\quad + R_{TG}(28 C_M + 77 C_D st)/st
\end{aligned} \tag{A.1}
$$

The meanings of all the symbols are given at the end of this appendix. The delay tau_{d2} for the complex RC model is

$$
\begin{aligned}
\tau_{d2} &= (stages - 1)\sqrt{0.48 R_G^2 (C_D + 4C_G)^2 + 0.5((C_D + 4C_G)^2 / G_M) R_G} \\
&\quad + \sqrt{\begin{array}{c} 0.48 \frac{R_G^2}{sd^2}(C_D sd + 7C_M + 22 C_D st)^2 \\ +0.5(C_D sd + 7C_M + 22 C_D st)/G_M (C_D + 4C_G) R_G \end{array}} \\
&\quad + \frac{R_{TG}}{st}(28 C_M + 77 C_D st)
\end{aligned} \tag{A.2}
$$

where all the symbols have the same meaning as above except that G_M is the transconductance I_{out}/V_{in} of an inverter.

Case 2 (2 drivers in ring bus)

$$
\tau_{d1} = 2(stages - 1)(R_G C_D + 4 R_G C_G)
$$

$$+ \frac{R_G}{sd}(2C_Dsd + 7C_M + 22C_Dst + C_G)$$
$$+ \frac{R_{TG}}{st}(16C_M + 48C_Dst + 3C_G) \quad (A.3)$$

$$\begin{aligned}\tau_{d2} = & (stages - 1)\sqrt{0.48R_G^2(C_D + 4C_G)^2 + 0.5((C_D + 4C_G)^2/G_M)R_G} \\ & + \sqrt{\begin{array}{c} 0.48\frac{R_G^2}{sd^2}(C_Dsd + 3C_M + 9C_Dst)^2 \\ +0.5(C_Dsd + 3C_M + 9C_Dst)(C_D + 4C_G)\frac{R_G}{G_M} \end{array}} \\ & + \frac{R_{TG}}{st}(6C_M + 15C_Dst + 3C_G) \\ & + (stages - 1)\sqrt{0.48R_G^2(C_D + 4C_G)^2 + 0.5((C_D + 4C_G)^2)\frac{R_G}{G_M}} \\ & + \sqrt{\begin{array}{c} 0.48\frac{R_G^2}{sd^2}(C_Dsd + 13C_Dst + 4C_M)^2 \\ +0.5(C_Dsd + 4C_M + 13C_Dst)(C_D + 4C_G)\frac{R_G}{G_M} \end{array}} \\ & + \frac{R_{TG}}{st}(10C_M + 33C_Dst) \end{aligned} \quad (A.4)$$

Case 3 (inter PE drivers)

$$\begin{aligned}\tau_{d1} = & 3(stages - 1)(R_GC_D + 4R_GC_G) \\ & + \frac{R_G}{sd}(3C_Dsd + 7C_M + 21C_Dst + 3C_G) \\ & + \frac{R_{TG}}{st}(12C_M + 35C_Dst + 5C_G) \end{aligned} \quad (A.5)$$

$$\begin{aligned}\tau_{d2} = & 3(stages - 1)\sqrt{0.48R_G^2(C_D + 4C_G)^2 + 0.5((C_D + 4C_G)^2)\frac{R_G}{G_M}} \\ & + \sqrt{\begin{array}{c} 0.48\frac{R_G^2}{sd^2}(C_Dsd + 2C_M + 6C_Dst + C_G)^2 \\ +0.5(C_Dsd + 2C_M + 6C_Dst + C_G)(C_D + 4C_G)\frac{R_G}{G_M} \end{array}} \\ & + \sqrt{\begin{array}{c} 0.48\frac{R_G^2}{sd^2}(C_Dsd + 3C_M + 8C_Dst + C_G)^2 \\ +0.5(C_Dsd + 3C_M + 8C_Dst + C_G)(C_D + 4C_G)\frac{R_G}{G_M} \end{array}} \\ & + \sqrt{\begin{array}{c} 0.48\frac{R_G^2}{sd^2}(C_Dsd + 2C_M + 7C_Dst)^2 \\ +0.5(C_Dsd + 2C_M + 7C_Dst)(C_D + 4C_G)\frac{R_G}{G_M} \end{array}} \\ & + \frac{R_{TG}}{st}(12C_M + 35C_Dst + 3C_G) \end{aligned} \quad (A.6)$$

Case 4 (Fully buffered rings)

$$
\begin{aligned}
\tau_{d1} \;=\;& 9(stages\;-\;1)(R_G C_D\;+\;4R_G C_G) \\
&+\frac{R_G}{sd}(9C_D sd\;+\;4.5C_M\;+\;26C_D st\;+\;9C_G) \\
&+\frac{R_{TG}}{st}(4.5C_M\;+\;17C_D st + 9C_G)
\end{aligned}
\qquad (A.7)
$$

$$
\begin{aligned}
\tau_{d2} \;=\;& 9(stages\;-\;1)\sqrt{0.48R_G^2(C_D\;+\;4C_G)^2\;+\;0.5(C_D\;+\;4C_G)^2\frac{R_G}{G_M}} \\
&+\sqrt{\begin{array}{c}0.48\frac{R_G^2}{sd^2}(C_D sd\;+\;0.5C_M\;+\;2C_D st\;+\;C_G)^2 \\ +0.5(C_D sd\;+\;0.5C_M\;+\;2C_D st\;+\;C_G)(C_D\;+\;4C_G)\frac{R_G}{G_M}\end{array}} \\
&+2\sqrt{\begin{array}{c}0.48\frac{R_G^2}{sd^2}(C_D sd\;+\;0.5C_M\;+\;6C_D st\;+\;C_G)^2 \\ +\;0.5(C_D sd\;+\;0.5C_M\;+\;6C_D st\;+\;C_G)(C_D\;+\;4C_G)\frac{R_G}{G_M}\end{array}} \\
&+\frac{R_{TG}}{st}(4.5C_M\;+\;17C_D st\;+\;9C_G)
\end{aligned}
\qquad (A.8)
$$

where

sd = gate size of final driver stage (width/length)

st = gate size of transmission gate (width/length)

R_G = on resistance of minimum size gate in the driver

R_{TG} = on resistance of minimum size transmission gate

C_G = input capacitance of minimum size inverter

C_D = output capacitance of minimum size inverter

C_M = capacitance of unit length (here $10000/mu$m) metal line

The actual values for these parameters were extracted from the ADVICE model for this process. R_G was obtained by measuring the average ON resistance of a transmission gate under ADVICE. As C_D involves both area and edge effects a value was used that gives a good result for a large range of transistor sizes and is thus not quite the true minimum.

Appendix B

Published Papers

K. Eshraghian, R.C. Bryant, A. Dickinson, D.S. Fensom, P.D. Franzon, M.T. Pope, J.E. Rockliff, and G. Zyner. The Transform and Filter Brick: A new architecture for signal processing. *VLSI 85, Tokyo Japan,* pages 129–138, August, 1985.

P.D. Franzon. Yield modeling for fault tolerant VLSI. In W. Moore, A. McCabe and R. Urquhart, editors, *Systolic Arrays, papers presented at the first international workshop on systolic arrays, Oxford, England*, pages 207–216, Adam Hilger, Bristol and Boston, 1986.

L.A. Hornak, S.K. Tewksbury, M. Hatamian, A. Ligtenberg, B. Sugla, and P. Franzon. Through-wafer optical interconnects for multi-wafer Wafer-Scale integrated architectures. In *SPIE 86, San Diego California*, August, 1986.

P.D. Franzon. Interconnect strategies for fault tolerant 2D VLSI arrays. In *ICCD-86*, pages 230–234, October, 1986.

S.K. Tewksbury, M. Hatamian, P. Franzon, L.A. Hornak, C.A. Siller, Jr., and V.B. Lawrence, FIR filters for high sample rate applications. *IEEE Communications*, July, 1987.

P. Franzon, M. Hatamian, L.A. Hornak, T. Little, and S.K. Tewksbury, Fundamental interconnection issues. *AT&T Technical Journal*, pages 13–30, September, 1987.

P. Franzon and S.K. Tewksbury. 'Chip Frame' scheme for reconfigurable mesh-connected arrays. In *Wafer Scale Integration II: International workshop on*

Wafer Scale Integration, Uxbridge England, Elsevier Science Publishers B.V. (North Holland), September, 1987.

P.D. Franzon and K. Eshraghian. Achieving ULSI through defect tolerance. To appear in *Int. J. of Computer Aided VLSI Design.*, Spring, 1989.

Bibliography

[Abraham & Agarwal, 1986] Abraham, J. and Agarwal, V., 1986. *Test generation for digital systems*, chapter 1, pages 1–94. Prentice Hall.

[Ackland et al., 1986] Ackland, B., Ahuja, S., DeBenedictis, E., London, T., Lucco, S., and Romero, D., 1986. MOS timing simulation on a message passing multiprocessor. In *ICCD-86*, pages 446 – 450, October.

[Ambekar et al., 1987] Ambekar, S., Hamilton, W., and Cole, T., 1987. Systems Packaging. *AT&T Technical Journal*, 66(4):81–95, July/August.

[Amdahl, 1984] Amdahl, C., 1984. *Electronic Design*, 3 May.

[Anderson, 1986] Anderson, A., 1986. Computer aided design and testing for RVLSI. In Jesshope, C. and Moore, W., editors, *Wafer Scale Integration (Proceedings of a workshop held in Southampton from 10 July to 12 July 1985)*, pages 216–222, Adam Hilger, Bristol and Boston.

[Athas & Seitz, 1988] Athas, W. and Seitz, C., 1988. Multicomputers: Message-passing concurrent computers. *IEEE Computer*, 21(8):9–25, August.

[Aubusson & Catt, 1978] Aubusson, R. and Catt, I., 1978. Wafer-scale integration – A fault tolerant procedure. *IEEE JSSC*, SC–13(3):339–693, June.

[Aubusson & Gledhill, 1978] Aubusson, R. and Gledhill, R., 1978. Wafer-scale integration – Some approaches to the interconnection problem. *Microelectronics*, 9(1):5 – 10.

[Avizienis & Laprie, 1986] Avizienis, A. and Laprie, J., 1986. Dependable computing: From concepts to design diversity. *Proc. IEEE*, 74(5):629 – 638, May.

[Barrett, 1987] Barrett, J. J., 1987. Electroplated copper conductors in ESPRIT 824. In *Proc. International Workshop on Wafer-Scale Integration*, 23-25 September.

[Barrett et al., 1986] Barrett, J., Smits, E., and Moran, P., 1986. A copper tracking technique for wafer scale integration. In Saucier, G. and Trilhe, J., editors, *Wafer Scale Integration (Proceedings of the IFIP WG 10.5 workshop, Grenoble, France 17-19 March, 1986)*, pages 291–300, North Holland.

[Bartlett et al., 1987] Bartlett, C., Segelken, J., and Teneketges, N., 1987. Multi-chip packaging design for VLSI based systems. In *Proc. 37th Electronic Components Conference*, pages 518–525, May.

[Bennett, 1985] Bennett, E., 1985. Inova makes its move in risky wafer–scale work. *Electronics*, December 2.

[Bentley & Jesshope, 1986] Bentley, L. and Jesshope, C., 1986. The implementation of a two-dimensional redundancy scheme in a wafer scale high-speed disk memory. In Jesshope, C. and Moore, W., editors, *Wafer Scale Integration (Proceedings of a workshop held in Southampton from 10 July to 12 July 1985)*, pages 187–197, Adam Hilger (Bristol and Boston).

[Bhatt & Leiserson, 1982] Bhatt, S. and Leiserson, C., 1982. How to assemble tree machines. In *14th ACM Symp on Theory of Computing*, pages 77 – 84.

[Blodgett & Barbour, 1982] Blodgett, A. and Barbour, D., 1982. Thermal conduction module: A high performance multilayer ceramic package. *IBM J. Res. Dev.*, 26(1), January.

[Bogdan, 1987] Bogdan, A., 1987. Electrically programmable silicon circuit board expands package options. *Hybrid Circuit Technology*, :9–14, April.

[Buehler, 1985] Buehler, M., 1985. *Microelectronic test chips for VLSI electronics*, pages 202 – 248. IEEE Computer Society Press/North Holland. reprinted from "VLSI Electronics: Microstructure Science", 1983 (Academic Press).

[Buslenko et al., 1966] Buslenko, N., Golenko, D., Shreider, Y., Sobol, I., and Sragovich, V., 1966. *The Monte Carlo Method.* Pergamon Press. Translated from Russian by G.J. Tee.

[Calhoun & MacNamee, 1972] Calhoun, D. and MacNamee, L., 1972. A means of reducing custom LSI interconnection requirements. *IEEE Journal of Solid State Circuits*, 7:395 – 404, October.

[Calhoun, 1969] Calhoun, D., 1969. A means of reducing custom LSI interconnection requirements. In *Proc. AFIP Fall Joint Computer Conference*, pages 99–109.

[Carter & Guise, 1983] Carter, D. and Guise, D., 1983. Analysis of signal propagation dealys and chip level performance due to on-chip interconnections. In *ICCD-83*, pages 218 – 221, IEEE.

[Chamberlain, 1984] Chamberlain, J., 1984. Branched labrynith wafer-scale integrated circuit. September. US Patent 4471483.

[Chapman, 1986] Chapman, G., 1986. Laser linking technology for RVLSI. In Jesshope, C. and Moore, W., editors, *Wafer Scale Integration (Proceedings of a workshop held in Southampton from 10 July to 12 July 1985)*, pages 204–216, Adam Hilger, Bristol and Boston.

[Chapman et al., 1987] Chapman, G., Raffel, J., Canter, J., and Rhodes, F., 1987. Advances in laser link technology for wafer-scale circuits. In *International Workshop on Wafer-Scale Integration*, 23–25 September.

[Chaturvedi, 1988] Chaturvedi, P., 1988. Wafer scale integration: a review. *Microelectronics*, 19(2):4 – 35.

[Chen & Strojwas, 1987] Chen, I. and Strojwas, A., 1987. Realistic yield simulation for VLSIC structural failures. *IEEE Trans. CAD*, CAD-6(6):965 – 980, November.

[Chen, 1969] Chen, A., 1969. Redundancy in LSI memory array. *IEEE J. Solid State Circ.*, :291 – 293, October.

[Chong & McEnroe, 1985] Chong, F. and McEnroe, P., 1985. A high density multichip memory module. In *Proc. WESCON*, November.

[Colbry et al., 1987] Colbry, B., Juki, M., Savicki, J., Schmidt, D., and Teneketges, N., 1987. High performance crosspoint switch implemented in CMOS on AVP. In *ICCD-87*, pages 113–116, IEEE.

[Cole, 1985] Cole, B., 1985. Wafer-scale faces pessimism. *Electronics Week*, :49–53, April 1.

[Deierling, 1987] Deierling, C. E., 1987. *Guidelines for the use of digital GaAs ICs*. GigaBit Logic Inc., January. Application Note 2.

[Denyer & Renshaw, 1985] Denyer, P. and Renshaw, D., 1985. *VLSI signal processing: A bit-serial approach.* Addison-Wesley.

[Dicken, 1988] Dicken, H., 1988. Demystifying ASIC costs. *VLSI Systems Design*, :86 – 94, May.

[Dingwall, 1968] Dingwall, A., 1968. High-yield-processed bipolar LSI arrays. In *Int. Electron Dev. Meet., Washington D.C.*, October. Abstract only available, reported in [Warner, 1981].

[Donlan et al., 1985a] Donlan, B., McDonald, J., Taylor, G., Steinvorth, R., and Bergendahl, A., 1985. Computer-Aided Design for wafer scale integration. *VSLI Design*, :43 – 42, April.

[Donlan et al., 1985b] Donlan, B., Taylor, G., Steinvorth, R., Bergendahl, A., and McDonald, J., 1985. Wafer-scale integration using discretionary microtransmission line interconnections. In Jesshope, C. and Moore, W., editors, *Wafer Scale Integration (Proceedings of a workshop held in Southampton from 10 July to 12 July 1985)*, pages 31 – 45, Adam Hilger.

[Donlan et al., 1986] Donlan, B., McDonald, J., Steinworth, R., Taylor, G., Dhodhi., and Bergendahl, A., 1986. The wafer transmission module. *VLSI Systems Design*, :54–90, January.

[Egawa et al., 1980] Egawa, Y., Wada, T., Ohmori, Y., Tsuda, N., and Masuda, K., 1980. A 1–Mbit full wafer MOS RAM. *IEEE JSSC*, SC-15(4):677–691, August.

[Ele, 1985a] *Electronics*, :41–42, December 9.

[Ele, 1985b] *Electronics*, :11, November 11.

[Ele, 1986] TRWs superchip passes first milestone. *Electronics*, :49–54, July 10.

[Ele, 1987] *Electronics Weekly*, (1375):21, July 29.

[EPP, 1987] Amorphous silicon promises to replace standard substrates. *Electronic Packaging and Production*, :13, April.

[Eshraghian et al., 1985] Eshraghian, K., Bryant, R., Dickinson, A., Fensom, D., Franzon, P., Pope, M., Rockliff, J., and Zyner, G., 1985. The Transform and Filter Brick: A new architecture for signal processing. In *VLSI 85, Tokyo Japan*, August.

[Evans & McWhirter, 1986] Evans, R. and McWhirter, J., 1986. A hierarchical test strategy for self-organising fault-tolerant arrays. In Moore, W., McCabe, A., and Urquhart, R., editors, *Systolic Arrays (Proceedings of Oxford Conference)*, pages 229 – 238.

[Evans, 1985] Evans, R., 1985. A self-organising fault-tolerant, 2-dimensional array. In *VLSI-85, Tokyo*, pages 233 – 242, August 26-28.

[Ferris-Prabhu, 1985a] Ferris-Prabhu, A., 1985. Defect size variations and their effect on the critical area of VLSI devices. *IEEE J. Solid State Circ.*, SC-20(4):878 – 880, August.

[Ferris-Prabhu, 1985b] Ferris-Prabhu, A., 1985. Modeling the critical yield area in yield forecasts. *IEEE J. Solid State Circ.*, SC-20(4):875 – 878, August.

[Ferris-Prabhu, 1988] Ferris-Prabhu, A., 1988. Yield implications and scaling laws for submicron devices. *IEEE Trans. Semiconductor Manu.*, 1(2):49 – 60, May.

[Finilla & Love, 1977] Finilla, C. and Love, H., 1977. The associative linear array processor. *IEEE Trans Comp*, C–26(2):112–125, February.

[Finnila & Love, 1978] Finnila, C. and Love, H., 1978. The associative array processor. *IEEE Trans. Comp.*, 13(3):339 – 344.

[Fischetti, 1984] Fischetti, M., 1984. Why Trilogy dropped WSI. *IEEE Spectrum*, :37, October.

[Franzon & Tewksbury, 1987] Franzon, P. and Tewksbury, S., 1987. 'Chip Frame' scheme for reconfigurable mesh-connected arrays. In *International workshop on Wafer Scale Integration, Uxbridge England*, September.

[Franzon, 1985] Franzon, P., 1985. *Yield and reliability models for fault tolerant VLSI.* Technical Report, University of Adelaide, August.

[Franzon, 1986a] Franzon, P., 1986. Interconnect strategies for fault tolerant 2D VLSI arrays. In *ICCD-86*, pages 230–234, October.

[Franzon, 1986b] Franzon, P., 1986. Yield modeling for fault tolerant VLSI. In W. Moore, A. M. and Urquhart, R., editors, *Systolic arrays, papers presented at the first international workshop on systolic arrays, Oxford, England July, 1986*, pages 207–216, Adam Hilger, Bristol and Boston.

[Franzon et al., 1987] Franzon, P., Hatamian, M., Hornak, L., Little, T., and Tewksbury, S., 1987. Fundamental interconnection issues. *AT&T Technical Journal*, :13–30, September.

[Fried, 1984] Fried, J., 1984. Wafer Scale Integration of Pipelined Processors. In *Proc ICCD 84*, IEEE.

[Fussel & Varman, 1982] Fussel, D. and Varman, P., 1982. Fault-tolerant wafer-scale architectures for VLSI. In *9th IEEE Symp. on Computer Architecture*, pages 190 – 198, April.

[Gaverick & Pierce, 1983] Gaverick, S. and Pierce, E., 1983. A single wafer 16-point 16 MHz FFT processor. In *Custom Integrated Circuits Conf.*, pages 244–248, May.

[Genestier et al., 1986] Genestier, P., Jay, C., and Saucier, G., 1986. A reconfigurable microprocessor for Wafer Scale Integration. In Saucier, G. and Trilhe, J., editors, *Wafer Scale Integration (Proceedings of the IFIP WG 10.5 workshop, Grenoble, France 17-19 March, 1986)*, pages 13–30, Elsevier Science Publishers B.V. (Amsterdam).

[Gentile et al., 1984] Gentile, G., Sami, M., and Terzoli, M., 1984. Design of switches for self-reconfiguring VLSI array structures. In *Microprocessing and Micrprogramming 14*, pages 99 – 108, North Holland.

[Giffels et al., 1987] Giffels, C. A., Gashler, R. J., Morabito, J. M., and Striny, K. M., 1987. Interconnection media. *AT&T Technical Journal*, 66(4):31–44, July/August.

[Girard et al., 1987] Girard, P., Pitoulet, B., Valenza, M., and Lorival, R., 1987. Electron beam switching of floating gate MOS transistors. In *International Workshop in Wafer Scale Integration*, IFIP, 23-25 September.

[Greene & Gamal, 1984] Greene, J. and Gamal, A. E., 1984. Configuration of VLSI arrays in the presence of defects. *J. ACM*, 31(4):694 – 717, October.

[Greub et al., 1987] Greub, H., Stoiantschewsky, S., Lahouar, S., Lopata, D., Pham, D., McDonald, J., and Creedon, T., 1987. FRISC - A fast reduced instruction set computer using advanced bipolar differential logic and wafer scale hybrid packaging. In *ICCD-87*, pages 478–484, IEEE.

[Grinberg et al., 1984] Grinberg, J., Nudd, G., and Etchells, D., 1984. A cellular VLSI architecture. *IEEE Computer*, :69–81, January.

[Gulett, 1981] Gulett, M., 1981. A practical method of predicting IC yields. *Semicond. Int.*, 4:87 – 94, February.

[Gupta & Lathrop, 1972] Gupta, A. and Lathrop, J., 1972. Yield analysis of large integrated-circuit chips. *IEEE J. Solid-State Circ.*, SC-7(5):389 – 395, October.

[Gupta et al., 1974] Gupta, A., Porter, W., and Lathrop, J., 1974. Defect analysis and yield degradation of integrated circuits. *IEEE J. Solid State Circuits*, SC-9(3):96 – 103, June.

[Gupta et al., 1982] Gupta et al., A., 1982. 5V EPROM – Springboard for autoprogrammable systems. *Electronics*, :121–125, Feb. 10.

[Ham, 1978] Ham, W., 1978. Yield-area analysis: Part I – A diagnostic tool for fundamental integrated-circuit process problems. *RCA Review*, 39:231 – 249, June.

[Harden & Strader, 1988] Harden, J. and Strader, N., 1988. Architectural yield optimization for WSI. *IEEE Trans. Comp.*, 37(1):88 – 110, January.

[Hassan & Agarwal, 1986] Hassan, A. and Agarwal, V., 1986. A modular approach to fault-tolerant binary tree architectures. *IEEE Trans. Comp.*, C-35, April.

[Hatamian & Cash, 1987] Hatamian, M. and Cash, G., 1987. Parallel bit-level pipelined VLSI design for high-speed signal processing. *Proc. IEEE*, September.

[Hedlund & Snyder, 1984] Hedlund, K. and Snyder, L., 1984. Systolic architectures – A wafer scale approach. In *Proc. ICCD-84*, pages 604–610, IEEE.

[Hedlund, 1982] Hedlund, K., 1982. *Wafer Scale Integration of CHiP processors.* PhD thesis, Purdue University.

[Hedlund, 1985] Hedlund, K., 1985. WASP - A wafer scale systolic processor. In *ICCD-85*, pages 665 – 671, IEEE.

[Hedlund, 1987] Hedlund, K., 1987. Private Communication.

[Hemmert, 1981] Hemmert, R., 1981. Poisson statistics and integrated circuit yield prediction. *Solid State Electronics (GB)*, 24:511 – 515.

[Hoover, Jr. et al., 1987] Hoover, Jr., C. W., Harrod, W. L., and Cohen, M. I., 1987. The technology of interconnection. *AT&T Technical Journal*, 66(4):2–12, July/August.

[Hornak & Tewksbury, 1987] Hornak, L. and Tewksbury, S., 1987. On the feasibility of through-wafer optical interconnects for hybrid wafer-scale-integrated architectures. *IEEE Trans. ED*, ED-34(7):1557–1563, July.

[Hornak et al., 1986] Hornak, L., Tewksbury, S., Hatamian, M., Ligtenberg, A., Sugla, B., and Franzon, P., 1986. Through-wafer optical interconnects for multi-wafer Wafer-Scale integrated architectures. In *SPIE 86, San Diego California*, August.

[Horowitz, 1983] Horowitz, M., 1983. *Timing models for MOS circuits.* PhD thesis, Stanford University, CA, December.

[Howells & Agarwal, 1988] Howells, M. and Agarwal, V., 1988. A reconfiguration scheme for yield enhancement of large area binary tree architectures. *IEEE Trans. Comp.*, C-37(4):463 – 468, April.

[Hsia et al., 1979] Hsia, Y., Chang, G., and Erwin, F., 1979. Adaptive wafer scale integration. In *Proc. 11th International Conference on Solid State Devices, Tokyo, 1979; Japanese J. App Phys, Volume 19 (1980), Supp. 19–1, pp. 193–202.*

[Hu, 1979] Hu, S., 1979. Some considerations in the formulation of IC yield statistics. *Solid State Electronics*, 22:205–211.

[Hu, 1984] Hu, S., 1984. Yield-I. Analysis of yield formulas, and Yield-II. A computer experiment. *IEEE Electron Devices Newsletter*, 69:4 – 10, March.

[Huang et al., 1983] Huang, C., Nunne, W., Spielberger, D., Mones, A., Fett, D., and Hampton, F., 1983. Silicon packaging – A new packaging technique. In *Proc. IEEE Customs Integrated Circuits Conf.*, pages 142–146, May.

[Hurst, 1985] Hurst, S., 1985. *Custom-specific integrated circuits.* Marcel Dekker. Appendix C.

[ICC, 1987] Keynote Address, ICCD-87.

[Inf, 1984] Where Amdahl failed. *Infomatics*, :4–5, August.

[Ipri & Sarace, 1977] Ipri, A. and Sarace, J., 1977. Integrated circuit process and design rule evaluation techniques. *RCA Review*, 38:323 – 350, September.

[Ipri, 1980] Ipri, A., 1980. Evaluation of CMOS transistor related design rules. *RCA Review*, 41:537 – 548, December.

[Ivey, 1987] Ivey, P., 1987. September. Personal communications.

[Ivey et al., 1987] Ivey, P., Huch, M., Midwinter, T., Hurat, P., and Glesner, M., 1987. Design of a large systolic array in Wafer-Scale technology. In *International Workshop in Wafer-Scale Integration*, September.

[J. Canter et al., 1986] J. Canter et al.., 1986. A laser-induced ohmic link for wafer-scale integration in standard CMOS processing. In *44th Annual device research Conf. Proc., Amherst*, June. paper VB-6.

[Jesshope & Moore, 1986] Jesshope, C. and Moore, W., 1986. *Wafer Scale Integration*, pages 137–139. Adam Hilger.

[Johnson, 1984] Johnson, R., 1984. The significance of wafer scale integration in computer design. In *ICCD-84*, pages 101–105, IEEE.

[Jones, 1987] Jones, S., 1987. Wafer-scale integration: Status of an art. In *International Workshop in Wafer-Scale Integration*, September.

[Katevenis & Blatt, 1985] Katevenis, M. and Blatt, M., 1985. Switch design for soft-configurable WSI systems. In *Chapel Hill Conference on VLSI*, pages 197 – 219.

[Katevenis, 1983] Katevenis, E., 1983. *Reduced instruction set computer architectures for VLSI.* PhD thesis, U.C. Berkeley.

[Kitano et al., 1980] Kitano, Y., Kohda, S., Kikuchi, H., and S, S., 1980. A 4-Mbit full-wafer ROM. *IEEE Trans. ED*, ED-27(8):1621–1628, August.

[Klass, 1986] Klass, P., 1986. Contractors use different techniques to meet VHSIC-2 objectives. *Aviation Week & Space Technology*, :123–128, October 20.

[Knausenberger & Schaper, 1984] Knausenberger, W. and Schaper, L., 1984. Interconnection costs of various substrates – The myth of cheap wire. *IEEE Trans. Components, Hybrids, Manafact. Technol.*, CHMT-7(3), September.

[Koren & Breuer, 1984] Koren, I. and Breuer, M., 1984. On area and yield consider-atitions for fault-tolerant VLSI processor arrays. *IEEE trans. Comp.*, C-33(1):21 – 27, January.

[Koren & Pradhan, 1985] Koren, I. and Pradhan, D., 1985. Introducing redundancy into VLSI designs for yield and performance enhancement. In *IEEE Fault Tolerant Computing Symposium 15*, pages 330 – 335, June.

[Koren & Pradhan, 1986] Koren, I. and Pradhan, D., 1986. Yield and performance enhancement through redundancy in VLSI and WSI multiprocessor systems. *Proc. IEEE*, 74(5):699 – 711, May.

[Koren et al., 1988] Koren, I., Koren, Z., and Pradham, D., 1988. Designing interconnection buses in VLSI and WSI for maximum yield and minimum delay. *IEEE J. Solid State Circ.*, 23(3):859 – 865, June.

[Kuhn, 1983] Kuhn, R., 1983. Interstitial fault tolerance – A technique for making systolic arrays fault tolerant. In *Proc. 16th Annual Hawaii Int. Conf. on System Sciences*, pages 215 – 224.

[Kung & Lam, 1984] Kung, H. and Lam, M., 1984. Wafer-scale integration and two level pipelined implementation of systolic arrays. *J. Parallel and Distributed Computing*, 1:32 –64.

[Lathrop et al., 1967] Lathrop, J., Clark, R., Hull, J., and Jennings, R., 1967. A discretionary wiring system as the interface between design automation and semiconductor array manufacture. *Proc. IEEE*, 55(11):1988–1997, November.

[Lea, 1986] Lea, R., 1986. WASP: A WSI processor for structured data processing. In Jesshope, C. and Moore, W., editors, *Wafer Scale Integration (Proceedings of a workshop held in Southampton from 10 July to 12 July 1985)*, pages 140–158, Adam Hilger (Bristol and Boston).

[Lea, 1987] Lea, R., 1987. WASP: A reconfigurable WSI associative string processor for highly-parallel information processing. In *International Workshop on Wafer-Scale Integration*, IFIP, September.

[Leighton & Leiserson, 1985] Leighton, T. and Leiserson, C., 1985. Wafer-scale integration of systolic arrays. *IEEE Trans. Comp.*, C-34(5):448 – 461, May.

[Lewis, 1984] Lewis, E., 1984. Optimization of device area and overall delay for CMOS VLSI designs. *Proc. IEEE*, 72(6):670 – 689, June.

[Lin et al., 1986] Lin, H., Rajapakse, R., Dhodhi, M., Jin, H., and McDonald, J., 1986. Use of the focused ion beam for unified lithography, testing and repair in wafer scale packaging. In *ICCD-86*, pages 225–229, IEEE.

[Little, 1986] Little, T., 1986. Private communication.

[Lowrie & Fuchs, 1987] Lowrie, M. and Fuchs, W., 1987. Reconfigurable tree architectures using subtree orientated fault tolerance. *IEEE Trans. Comp.*, C-36(10):1171 – 1182, October.

[Lukaszek et al., 1986] Lukaszek, W., W.Yarbrough., Walker, T., and Meindl, J., 1986. CMOS test chip for process problem debugging and yield prediction experiments. *Solid State Tech.*, :87 – 93, March.

[Lyman, 1986] Lyman, J., 1986. The latest wafer-scale design is a hybrid. *Electronics*, :28, March 17.

[Lyman, 1987] Lyman, J., 1987. Silicon on silicon hybrids are coming into their own. *Electronics*, :47–48, May 28.

[Lyon, 1984] Lyon, R., 1984. *MSSP: A bit-serial multiprocessor for signal processing*, pages 64 – 75. IEEE Press.

[MacDonald & Neish, 1982] MacDonald, N. and Neish, G., 1982. An algorithmic approach to the testing of a WSI system. In *IEEE Test Conference*, pages 590 – 600.

[Mahalingam, 1985] Mahalingam, M., 1985. Thermal management in semiconductor device packaging. *Proc. IEEE*, 73(9):1396–1404, September.

[Mallory et al., 1983] Mallory, C., Perloff, D., Hasan, T., and Stanley, R., 1983. Spatial yield analysis in integrated circuit manufacturing. *Solid State Technology*, :121 – 127, November.

[Maly, 1985] Maly, W., 1985. Modeling of lithography related yield losses for CAD of VLSI circuits. *IEEE Trans. CAD*, CAD-4(3):166 – 177, July.

[Mangir, 1984] Mangir, T., 1984. Sources of failures and yield improvement for VLSI and restructurable interconnects for RVLSI and WSI: Part I– Sources of failure and yield improvement for VLSI. *Proc. IEEE*, 72(6):690–709, June.

[Mann & Rhodes, 1986] Mann, J. and Rhodes, F., 1986. A wafer scale DTW multiprocessor. In *Proceedings of ICASSP'86, Tokyo, Japan*, April.

[Manning, 1977] Manning, F., 1977. An approach to highly–integrated computer maintained cellular arrays. *IEEE Trans. Comp*, C–26(6):536–552, June.

[Marwood & Clarke, 1985] Marwood, W. and Clarke, A., 1985. Fault tolerant systolic architectures. In *Proc. 3rd National Workshop on Fault Tolerant Computing*. Monash, Melbourne.

[Meindl, 1984] Meindl, J., 1984. Ultra large scale integration. *IEEE Trans ED*, ED-31(1):1555–1561.

[Messner, 1987] Messner, G., 1987. Cost-density analysis of interconnections. *IEEE Trans. Components, Hybrids, Manafact. Technol.*, CHMT-10(2):143–151, June.

[Moore & Day, 1984] Moore, W. and Day, M., 1984. Yield enhancement of a large systolic array chip. *Microelectron. Reliab.*, 24(3):511 – 526.

[Moore & Mahat, 1985] Moore, W. and Mahat, R., 1985. Fault-tolerant communications for wafer-scale integration of a processor array. *Microelectron. Reliab.*, 25(2):291 – 294.

[Moore, 1970] Moore, G., 1970. What level of LSI is for you? *Electronics*, :126 – 130, February 16.

[Moore, 1986] Moore, W., 1986. A review of fault tolerant techniques for integrated circuit yield. *Proc. IEEE*, May.

[Moore, 1987] Moore, W., 1987. Private communications.

[Murphy, 1964] Murphy, B., 1964. Cost-size optima of monolithic integrated circuits. *Proc. IEEE*, 52:1537 – 1545, December.

[Negrini et al., 1985] Negrini, R., Sami, M., and Stefanelli, R., 1985. Fault-tolerance approaches for VLSI/WSI arrays. In *4th Phoenix Computer and Communication Conference*, pages 460 – 468.

[Neugebauer, 1984] Neugebauer, C., 1984. Approaching Wafer Scale Integration from the packaging point of view. In *ICCD 84*, pages 115–120, IEEE, October.

[Newman, 1985] Newman, M., 1985. Is WSI too big? Selecting an appropriate component size for redundant logic. In *Proceedings VLSI 85.*, pages 97 – 108, August.

[Nicolas, 1987] Nicolas, G., 1987. Hybrid connections on Silicon. In *Workshop on Wafer Scale Integration*, IFIP, 23–25 September.

[Okabe et al., 1972] Okabe, T., Nagata, M., and Shimada, S., 1972. *Elec. Eng. Japan.*, 92:135 –.

[Pateras & Rajski, 1988] Pateras, S. and Rajski, J., 1988. Self-Reconfiguring interconnection network for a fault tolerant mesh-connected array of processors. *Electronics Letters*, 24(10):600 – 602, 12 May.

[Paz & Lawson, 1977] Paz, O. and Lawson, T., 1977. Modification of Poisson statistics: Modeling defects induced by diffusion. *IEEE J. Solid-State Circuits*, SC-12(5):540 – 546, October.

[Peltzer, 1983] Peltzer, D., 1983. Wafer-scale integration: The limits of VLSI? *VLSI Design*, :43–47, September.

[Price, 1970] Price, J., 1970. A new look at yield of integrated circuits. *Proc. IEEE*, :1290 – 1291, August. See also Proc. IEEE, July, 1971, pp. 1128 by B.T. Murphy.

[Raffel, 1986] Raffel, J., 1986. The RVLSI approach to wafer-scale integration. In Jesshope, C. and Moore, W., editors, *Wafer Scale Integration (Proceedings of a workshop held in Southampton from 10 July to 12 July 1985)*, pages 199–203, Adam Hilger, Bristol and Boston.

[Raffel et al., 1985] Raffel, J., Anderson, A., Chapman, G., Konkle, K., Mathur, B., Soares, A., and Wyatt, P., 1985. A wafer-scale digital integrator using restructurable VLSI. *IEEE Trans. Electron Devices*, ED-32:479–486.

[Raghavendra et al., 1983] Raghavendra, C., Avizienis, A., and Ercegovac, M., 1983. Fault-tolerance in binary tree architectures. In *Fault Tolerant Conference*, pages 360 – 363.

[Rainal, 1979] Rainal, A., 1979. The transmission properties of various styles of printed wiring boards. *Bell System Technical Journal*, 58(5):995, May/June.

[Rhodes, 1986] Rhodes, F., 1986. Applications of RVLSI to signal processing. In Jesshope, C. and Moore, W., editors, *Wafer Scale Integration (Proceedings of a workshop held in Southampton from 10 July to 12 July 1985)*, pages 199–203, Adam Hilger, Bristol and Boston.

[Robinson & Sauve, 1977] Robinson, R. and Sauve, S., 1977. *Analysis of failed parts on naval avionics systems.* Report D180-22840-1, The Boeing Company, October.

[Rosenberg, 1983a] Rosenberg, A., 1983. The Diogenes approach to testable fault-tolerant arrays of processors. *IEEE Trans. Comp.*, C-32(10):902 – 909, October.

also see "Comments" in IEEE Trans. Comp., Jan., 1986, pages 93 – 94, by I. Koren.

[Rosenberg, 1983b] Rosenberg, A. L., 1983. Three dimensional VLSI: A case study. *Journal of the Association of Computing Machinery*, 30(3):397–416, July.

[Rosenberg, 1984] Rosenberg, A., 1984. On designing fault-toelrant VLSI processor arrays. In *Advances in Computing Research*, pages 181 – 204, JAI Press.

[Sami & Stefanelli, 1983] Sami, M. and Stefanelli, R., 1983. Reconfigurable architectures for VLSI processing arrays. In *Proc. Nat. Comp. Conf., Los Angeles*, pages 565 – 577, AFIPS.

[Sami & Stefanelli, 1984] Sami, M. and Stefanelli, R., 1984. Fault-tolerance of VLSI processing arrays: The time-redundant approach. In *Proc. 1984 Real Time Systems Conf., Austin TX*, pages 200 – 207, IEEE, December.

[Sami & Stefanelli, 1986] Sami, M. and Stefanelli, R., 1986. Reconfigurable architectures for VLSI processing arrays. *Proc. IEEE*, :712 – 722, May.

[Saucier, 1986] Saucier, G., 1986. Terminology for WSI design and test. In *Wafer Scale Integration (Proceedings of the IFIP WG 10.5 workshop, Grenoble, France 17-19 March, 1986)*, pages *xiv* – *xx*.

[Seeds, 1967] Seeds, R., 1967. Yield, economic, and logistical models for complex digital arrays. In *IEEE Communication Convention Record*, pages 60 – 61, Pt. 6, April.

[Singh, 1985] Singh, A., 1985. An area efficient redundancy scheme for wafer scale processor arrays. In *ICCD-85*, pages 505 – 509.

[Snyder, 1982] Snyder, L., 1982. Introduction to the Configurable, Highly Parallel Computer. *IEEE Computer*, V–15(1):47–56, January.

[Stapper, 1973] Stapper, C., 1973. Defect density distribution for LSI yield calculations. *IEEE Trans ED*, :655 – 657, July.

[Stapper, 1975] Stapper, C., 1975. On a composite model to the IC yield problem. *IEEE J. Solid-State Circ.*, :537 – 539, December.

[Stapper, 1976] Stapper, C., 1976. LSI yield modeling and process monitoring. *IBM J. Res. Develop.*, :228 – 234, May.

[Stapper, 1981] Stapper, C., 1981. Comments on "Some considerations in the formulation of IC yield statistics. *Solid State Electronics (GB)*, 24:127 – 132.

[Stapper, 1982] Stapper, C., 1982. Yield model for 256K RAMs and beyond. In *Int. Solid State Circ. Conf.*, pages 12 – 13.

[Stapper, 1983] Stapper, C., 1983. Modeling of integrated circuit defect sensitivities. *IBM J. Res. Dev.*, 27:549 – 557, November.

[Stapper, 1984a] Stapper, C., 1984. Modeling of defects in integrated circuit photolithographic patterns. *IBM J. Res. Develop.*, 28(4):461 – 475, July.

[Stapper, 1984b] Stapper, C., 1984. Yield model for fault clusters within integrated circuits. *IBM J. Res. Develop.*, 28(5):636 – 640, September.

[Stapper, 1985] Stapper, C., 1985. The effects of wafer to wafer defect density variations on integrated circuit defect and fault distributions. *IBM J. Res. Develop*, 29(1):87 – 97, January.

[Stapper, 1986a] Stapper, C., 1986. The defect sensitivity effect of memory chips. *IEEE J. Solid State Circ.*, SC-21(1):193 – 198, February.

[Stapper, 1986b] Stapper, C., 1986. On yield, fault distributions, and clustering of particles. *IBM J. Res. Develop.*, 30(3):326 – 338, May.

[Stapper et al., 1980] Stapper, C., McLaren, A., and Dreckman, M., 1980. Yield model for productivity optimization of VLSI memory chips with redundancy and partially good product. *IBM J. Res. Develop.*, 24(3):398 – 349, May.

[Stapper et al., 1982] Stapper, C., Castrucci, P., Maeder, R., Rowe, W., and Verhelst, R., 1982. Evolution and accomplishments of VLSI yield management at IBM. *IBM J. Res. Develop.*, 26(5):532 – 545, September.

[Stapper et al., 1983] Stapper, C. H., Armstrong, F. M., and Saji, K., 1983. Integrated circuit yield statistics. *Proc. IEEE*, 71(4):453 – 470, April.

[Tammaru & Angell, 1967] Tammaru, E. and Angell, J., 1967. Redundancy for LSI yield enhancement. *IEEE J. Solid State Circ.*, SC-2:172 – 182, December.

[Taylor et al., 1985] Taylor, G., Donlan, B., McDonald, J., Bergendahl, A., and Steinvorth, R., 1985. The wafer transmission module – Wafer-scale integration packaging. In *IEEE Custom Int. Circ. Conf.*, pages 55 – 58.

[Tewksbury & Hornak, 1987] Tewksbury, S. and Hornak, L., 1987. Future physical environments and concurrent computation. In *1987 Princeton Workshop on Algorithms, Architecture and Technology Issues in Models of Concurrent Computation.*

[Tewksbury et al., 1987] Tewksbury, S., Hatamian, M., Franzon, P., L.A. Hornak, C.A. Siller, J., and Lawrence, V., 1987. FIR filters for high sample rate applications. *IEEE Communications*, July.

[Trilhe & Saucier, 1987] Trilhe, J. and Saucier, G., 1987. WSI – The challenge of the future. In *Proc. COMPEURO, Hamburg*, pages 531–542, IEEE.

[Tsuda & Satoh, 1987] Tsuda, N. and Satoh, T., 1987. Hierarchical redundancy for a linear array sorting chip. In *Proc. IFIP International Workshop on Wafer-Scale Integration, Brunel, England*, September.

[Turley & Herman, 1974] Turley, A. and Herman, D., 1974. LSI yield projections based upon test pattern results: An application to multilevel metal structures. *IEEE Trans. Parts, Hyb., Pack.*, PHP-10(4):230 – 234, December.

[Ueko et al., 1984] Ueko, Y., Minagaw, C., Oka, M., and Ishimoto, A., 1984. A defect-tolerant design for full-wafer memory LSI. *IEEE JSSC*, SC–19(3):319–329, June.

[Walker & Director, 1986] Walker, H. and Director, S., 1986. VLASIC: A catastrophic fault yield simulator for integrated circuits. *IEEE Trans. CAD*, CAD-5(4):541 – 556, October.

[Walker, 1986] Walker, H., 1986. *Yield simulation for integrated circuits.* PhD thesis, Computer Sci. Dept. Carnegie-Mellon Univ.

[Waller, 1988] Waller, L., 1988. The Pentagon will settle for not-so-super VHSIC superchip. *Electronics*, :34 – 35, 28 April.

[Wang et al., 1987] Wang, M., Cutler, M., and Su, S., 1987. On-line error detection and reconfiguration of array processors with two-level redundancy. In *Proc. CompEuro 87 on VLSI and Computers, Hamburg West Germany*, May.

[Warner, 1974] Warner, R., 1974. Applying a composite model to the IC yield problem. *IEEE J. Solid-State Circ.*, SC-9(3):86 – 95, June.

[Warner, 1981] Warner, R., 1981. A note on IC-yield statistics. *Solid State Electronics (GB)*, 24(11):1045 – 1047.

[Watari, 1987] Watari, T., 1987. The NEC SX supercomputer technology. In *ICCD 87*, pages 54–57, IEEE, October.

[Weste, 1981] Weste, N., 1981. MULGA - An interactive symbolic layout system for the design of integrated circuits. *Bell System J.*, 60(6):823 – 858.

[Wilkinson, 1987] Wilkinson, M., 1987. June. Personal Communications (Anamartic Pty. Ltd.).

[Woods, 1986] Woods, M. H., 1986. MOS VLSI reliability and yield trends. *Proceedings of the IEEE*, 74(12):1715–1729, December.

[Wooley et al., 1987] Wooley, B., Horowitz, M., Pease, R., and Yang, T., 1987. Active substrate system integration. In *ICCD-87*, pages 468–471, IEEE, October.

[Wyatt et al., 1984] Wyatt, P., Raffel, J., Chapman, G., Burns, J., and Herndon, T., 1984. Process considerations in restructurable VLSI for wafer-scale integration. In *IEDM 84*, pages 626–629.

[Yanagawa, 1969] Yanagawa, T., 1969. Influence of epitaxial mounds on the yield of integrated circuits. *Proc. IEEE*, 57(9):1621 – 1628, September.