

SELF TEST AND SELF REPAIR STRATEGIES
IN VLSI ARCHITECTURES FOR HIGH SPEED
DIGITAL CORRELATION

William Sinclair Blackley

A Thesis Submitted to the Faculty of Science,
University of Edinburgh, for the degree of
Doctor of Philosophy.

Department of Electrical Engineering
1985



A B S T R A C T

In this thesis, the concepts of self test and self repair are applied to a VLSI architecture for digital polarity correlation. A prototype correlator chip has successfully demonstrated the value of regular array architectures with built-in self test and self repair in the implementation of large area silicon systems.

The polarity correlation function is implemented using an overloading integrating counter technique. This technique permits direct cascading of individual correlator chips, without using additional components, to give complete flexibility in choice of correlator delay and resolution. Regularity, and a concerted strategy of design for testability in the chip's architecture, allow the correlator to perform self test and self repair in an economic and efficient manner. The built-in self test and self repair mechanisms automatically detect and eliminate failed channels in the VLSI circuit.

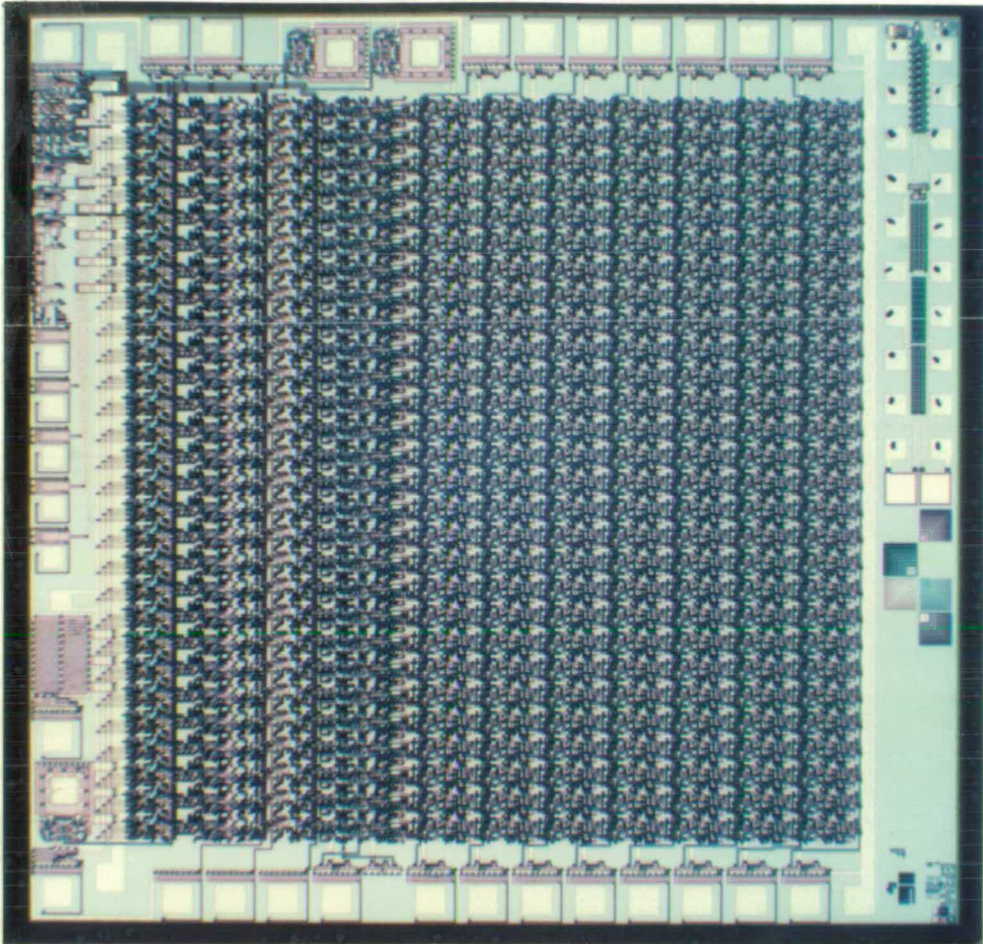
A review of correlation techniques in VLSI, and the concepts of fault tolerance and yield enhancement are presented. The correlator has been fabricated on a five-micron N-channel MOS process and results from the prototype chips are reported.

DECLARATION OF ORIGINALITY

This thesis, composed entirely by myself, reports on work conducted by myself in the Department of Electrical Engineering, University of Edinburgh.

Signed.

William Blackley



Photograph of Eu349 Digital Polarity Correlator Chip

Table of Contents

GLOSSARY	xi
 CHAPTER 1: INTRODUCTION	 1
1.1. VLSI: Linking Design and Test	1
1.2. Layout of Thesis	3
 CHAPTER 2: CORRELATION THEORY AND TECHNIQUES	 4
2.1. Introduction	4
2.2. Interpreting the Correlation Function	6
2.3. Historical Development	12
2.4. Correlation Principles	13
2.4.1. Random Data Concepts	13
2.4.2. Fundamental Estimation Errors	15
2.4.3. Discrete Time Correlation	17
2.5. Correlation Techniques	19
2.5.1. Quantisation of Input Data	20
2.5.2. Direct Analogue Correlation	22
2.5.3. Stieltjes Correlation	23
2.5.4. Relay Correlation	24
2.5.5. Multilevel Digital Correlation	25
2.5.6. Polarity-Coincidence Correlation	27
2.5.7. Modified Correlators: Dither	28
2.6. Polarity Correlation and the Overloading Integrating Counter Technique	30

2.6.1. Polarity Correlation	30
2.6.2. Overloading Counter Technique	34
2.7. Summary	38
 CHAPTER 3: INTEGRATED CIRCUIT CORRELATORS	 40
3.1. Introduction	40
3.2. Correlation Architectures	41
3.2.1. Serial Architecture	41
3.2.2. Parallel Architecture	43
3.2.2.1. Parallel architecture with temporal integration and spatial delay	43
3.2.2.2. Parallel architecture with temporal integration and spatial delay using the overload- ing counter technique	47
3.2.2.3. Parallel architecture with spatial integration and temporal delay	50
3.2.2.4. Parallel architecture with spatial integration and temporal delay using pipe-organ structures	56
3.2.3. Serial Parallel Architectures (DELTIC)	58
3.2.4. Systolic Architectures	60
3.2.4.1. Temporal Integration	60
3.2.4.2. Spatial Integration	62
3.3. Correlation Cube: The Difference Between Temporal and Spatial Integration	67
3.3.1. Correlator Architecture based on Spatial Integration	69
3.3.2. Correlation Architecture based on Temporal Integration	72
3.3.3. Display of Correlation Output	73
3.4. Summary	74

CHAPTER 4: VLSI DESIGN STRATEGIES FOR TESTABILITY AND FAULT TOLERANCE	76
4.1. Introduction	76
4.2. Test Philosophies and The Motivation Behind Design for Testability	78
4.3. Design for Testability Methods	81
4.3.1. Objectives	81
4.3.2. Ad Hoc Methods	82
4.3.3. Scan Methods	83
4.3.4. Built-In Self Test Methods	87
4.4. VLSI Design for Testability in the Eu349 Correlator Chip	91
4.5. Integrated Circuit Yield Statistics	94
4.5.1. Scope	94
4.5.2. Yield Loss due to Gross Defects	96
4.5.3. Yield Model for Random Defects	98
4.5.4. General Yield Model for VLSI Chips with No Redundancy	102
4.5.5. Yield Model for Chips with Redundancy	104
4.5.6. Cost of Redundancy	110
4.6. Yield Enhancement Techniques	112
4.6.1. Scope	112
4.6.2. Integrated Circuit Redundancy Schemes	113
4.6.2.1. Bypass schemes	114
4.6.2.2. Nearest neighbour schemes	116
4.6.2.3. Chaining schemes	116
4.6.3. Comparison of Redundancy Schemes	117
4.7. Yield Enhancement Features in the Eu349 Correlator Chip	119

4.8. Summary	121
CHAPTER 5: DESIGN AND TEST OF THE PROTOTYPE INTEGRATED CIRCUIT	123
5.1. Introduction	123
5.2. Architecture of the Basic Polarity Correlator	123
5.3. Architecture of the Correlator with Built-In Self Test and Self Repair Features	127
5.4. Design of the Eu349 Correlator	131
5.4.1. System Overview	131
5.4.2. Correlator Array Design	137
5.4.3. Peripheral Circuit Design	144
5.5. Test Strategy	144
5.5.1. Initial Test	144
5.5.2. Self test	145
5.5.3. Self repair	146
5.5.4. Run	146
5.6. Test System Configuration and Results	147
5.6.1. Test Configuration	147
5.6.2. Test Results	147
5.6.3. Yield Enhancement	159
5.7. Summary	161
CHAPTER 6: CONCLUSIONS	162
6.1. Summary of Work	162
6.2. Further Work	164

ACKNOWLEDGEMENTS	167
 APPENDIX 1. EU349 CORRELATOR DESIGN	168
A1.1. Introduction	168
A1.2. Silicon Design	169
A1.3. Peripheral Circuitry Design	172
A1.4. Power Supply Considerations	180
 APPENDIX 2. EU349 TEST SCHEDULE	181
A2.1. Introduction	181
A2.2. Test MCR as Shift Register	181
A2.3. Test OSR and DSR as Shift Registers	182
A2.4. Test MCR Effect on both OSR and DSR	182
A2.5. Test SET and CLEAR Features of DSR	183
A2.6. Test Latches and MCR Parallel Load	183
A2.7. Test Latches and OSR Parallel Load	184
A2.8. Self Test Sequence	185
A2.9. Self Repair Sequence	187
A2.10. Correlation Test Sequence	187
 APPENDIX 3. EU349 TEST CONFIGURATION	190
A3.1. Introduction	190
A3.2. Prototype Test Configuration	190
A3.3. DAS Data Probes	191
A3.3.1. 91A32 Data Acquisition Module	192
A3.3.2. 91P16 Pattern Generator Module	192
A3.3.3. 91P32 Pattern Generator Module	193

A3.4. Channel Specification 194

A3.5. Timing Diagram 196

A3.6. Trigger Specification 197

A3.7. Pattern Generator - Timing 198

A3.8. Pattern Generator Instruction Codes 199

A3.9. Pattern Generator - Program 201

APPENDIX 4. AUTHOR'S PUBLICATIONS 204

REFERENCES 205

GLOSSARY

<i>Symbol</i>	<i>Description</i>
α	Clustering parameter
A	Defect susceptible chip area
A_O	Chip area without redundancy
A_E	Chip area with redundancy
A_m	Module area
B	Bandwidth
BILBO	Built-In Logic Block Observer
BIST	Built-In Self Test
CCD	Charge Coupled Device
CLK	Clock
CMOS	Complementary Metal-Oxide-Semiconductor
CRC	Cyclic Redundancy Check
Δt	Sample interval
D	Average defect density
DAS	Digital Analysis System
DELTIC	Delay Line Time Compressor Correlator
DIL	Dual-In-Line
DSR	Data Shift Register
DUT	Device Under Test
ECL	Emitter Coupled Logic
EXNOR	Exclusive-NOR function
EXOR	Exclusive-OR function
$F_n(\tau)$	Coincidence function
FM	Figure of merit
$F\{x(t)\}$	Fourier transform of $x(t)$
G_{yx}	Cross power spectrum of $y(t)$ and $x(t)$
GND	Ground
I_{sat}	Value of drain current at saturation

<i>Symbol</i>	<i>Description</i>
IIL	Integrated Injection Logic
k	Sequence index variable
k	Process gain factor
λ	Average number of faults ($\lambda=AD$)
L	MOS transistor gate length
LFSR	Linear Feedback Shift Register
LSB	Least Significant Bit
LSSD	Level Sensitive Scan Design
m	Number of input sample pairs
MCR	Multiplexer Control Register
MISR	Multiple Input Signature Register
MOS	Metal-Oxide-Semiconductor
MUX	Multiplexer
N	Capacity of each integrating counter
NMOS	N-channel Metal-Oxide-Semiconductor
OD	Overload detect
OSR	Overload Shift Register
OVRFLO	Integrating counter overload signal
$\phi 1$	Clock phase 1
$\phi 2$	Clock phase 2
PCM	Pulse Code Modulation
PMOS	P-channel Metal-Oxide-Semiconductor
PRBS	Pseudo-Random Binary Sequence
q	Quantisation interval
$q(\tau)$	Contents of integrating counter
$r_{dyx}(\tau)$	Direct digital correlation function
$r_{nxx}(\tau)$	Normalised autocorrelation function of $x(t)$
$r_{pyx}(\tau)$	Polarity correlation function
$r_{ryx}(\tau)$	Relay correlation function
$r_{syx}(\tau)$	Stieltjes correlation function
$r_{xx}(\tau)$	Autocorrelation function of $x(t)$
$r_{xxx}(\tau_1, \tau_2)$	Triple correlation function
$r_{yx}(\tau)$	Crosscorrelation function of $y(t)$ and $x(t)$

*Symbol**Description*

RMS	Root Mean Square
σ_x^2	Mean square value of x
σ	Standard deviation
S_x	Spectrum of x
SAW	Surface Acoustic Wave
SGN	Signum, i.e. polarity
τ	Time delay
TB	Time-bandwidth product
V_{ds}	MOS transistor drain to source voltage
V_{gs}	MOS transistor gate to source voltage
V_{th}	MOS transistor threshold voltage
VBB	MOS back bias voltage supply
VDD	MOS transistor drain voltage supply
VIN	Input voltage
VLF	Very Low Frequency
VLSI	Very Large Scale Integration
VSS	MOS transistor source voltage supply
W	MOS transistor gate width
W/L	MOS gate aspect ratio
x^+	Quantised version of x
Y	Preassembly yield
Y_{CRD}	Correctable random defect yield
Y_E	Enhanced yield
Y_G	Yield due to gross defects
Y_R	Yield due to random defects
Y_{UNC}	Uncorrectable random defect yield
Y_{eff}	Effective yield
Y_m	Module yield

CHAPTER 1

INTRODUCTION

1.1. VLSI: Linking Design and Test

The maturing of silicon integrated circuit technology from large scale to very large scale integration, has improved performance, reduced costs and opened new systems applications. However, one important facet of integrated circuit technology lags dangerously behind the complexity potential of VLSI: establishing the integrity of the VLSI design in terms of initial design validation, manufacturing quality, and fault tolerance [1].

This thesis addresses the need to embody a testability scheme within the VLSI integrated circuit itself. It presents details of a digital polarity correlator architecture with built-in self test and self-repair mechanisms. Results obtained from a prototype integrated circuit chip fabricated in five-micron enhancement/depletion N-channel MOS technology demonstrate the concept.

Correlation techniques are widely used in communications, instrumentation, telemetry, sonar, radar, and in medical diagnosis. Important correlation properties include the ability to detect a desired signal in the presence of noise or other signals, to recognise specific patterns, and to determine time delays through various media. Electronic systems for computation of the correlation function have been available for many years, but they have been large and inefficient. With the development of VLSI, correlation can be performed efficiently and with

fewer components.

The integrated circuit to be described here, offers a digital implementation of the polarity correlation function using an overloading integrating counter technique [2]. The VLSI architecture offers high speed operation, long (programmable) integration time, and an arbitrarily long correlation time delay. The mathematical theory of correlation analysis, including the effects of finite sampling and quantisation is presented as a prerequisite to deriving the overloading integrating counter technique. From this theoretical base the architecture of the correlator chip is described.

The architecture consists of a linear cascade of identical correlation elements. The performance of the correlator depends on the serial connection of correctly functioning correlation elements. To optimise the performance and gain full advantage of the VLSI architecture, a design philosophy was adopted which includes design for testability, self test, and fault tolerance.

The question "why design for testability?" is answered by discussing some existing test philosophies. Various approaches exist, and each has its specific applications, but there is no general agreement on how to design for testability. The thesis examines the "ad hoc" testability approach, which consists of circuit partitioning and added test points. This is contrasted with the "structured" testability approach, where the test problem is solved at a much lower design level. The object of a structured approach is to reduce the sequential complexity of a logic network and thus aid test generation and verification.

Built-in test and self test techniques are also discussed. Built-in test techniques, when used in

conjunction with redundant circuitry and reconfiguration techniques in VLSI, provide the basis of self repairing systems. The ease with which built-in self test and self repair techniques have been employed in the VLSI architecture to be described here, is demonstrated by the very low overhead required in silicon area.

1.2. Layout of Thesis

In Chapter 2, a concise background and theory of correlation is presented. The effect of finite averaging time, discrete sampling, and quantisation of input data are discussed. Quantisation of the input data is used to link direct correlation to relay correlation, and to polarity correlation. The overloading integrating counter technique is then derived for the polarity correlator.

A review of silicon correlators is presented in Chapter 3. In this chapter a comparison is made between the various published architectures (including the one expounded by this thesis), that have been realised as silicon integrated circuits. In Chapter 4, the concept of design for testability is discussed, and the subject of integrated circuit yield statistics is introduced. Circuit redundancy is discussed as a method for achieving yield enhancement and fault tolerance.

The integrated circuit design is described in Chapter 5. The architecture of the basic correlator is shown modified to allow built-in self test and self repair. The performance of the prototype chip and the experimental results of the self repair concept are also presented in Chapter 5.

Chapter 6 summarises and highlights observations from the work. In addition, areas of special interest that may be considered for further investigation are identified.

CHAPTER 2

CORRELATION THEORY AND TECHNIQUES

2.1. Introduction

Correlation analysis is of great interest to engineers and scientists. A wide range of engineering applications of random data analysis centres around the determination of linear relationships between two or more sets of data. These linear relationships may be extracted in terms of a correlation function [3,4]. Correlation techniques are widely used in communications [5], sonar [6], radar [7,8,9], and medicine [10,11,12], where they are used to detect known signals in the presence of noise or other signals [13,14]. They have application in many areas such as spectral estimation [15], time response measurements of linear systems [16,17,18,19], pattern recognition [20,21,22,23,24], and time delay estimation [25,26,27] including flow measurement [28,29,30,31,32,33]. The bandwidths of the signals to be correlated vary from several Hz. in seismology and very low frequency (vlf) radio wave studies [34], to several MHz. in photon spectroscopy [35,36], radio astronomy [37,38], or plasma physics experiments [39,40,41], for example. Other fields in which correlators are useful include flaw detection and system health monitoring [42,43].

This chapter deals with the historical development of correlator systems, and the mathematical theory of correlation analysis. A brief summary of random data concepts is included, on account of the statistical nature of correlations. The concept of the ideal correlation

coefficient, which is computed over an infinite number of data sets, is related to the correlation function, which is computed over a single data set for a finite length of time. This relationship is crucial to the physical realisation of a correlation system.

A correlator consists of three basic elements: a delaying device, a multiplier, and an averager or integrator as shown in Figure 2.1.

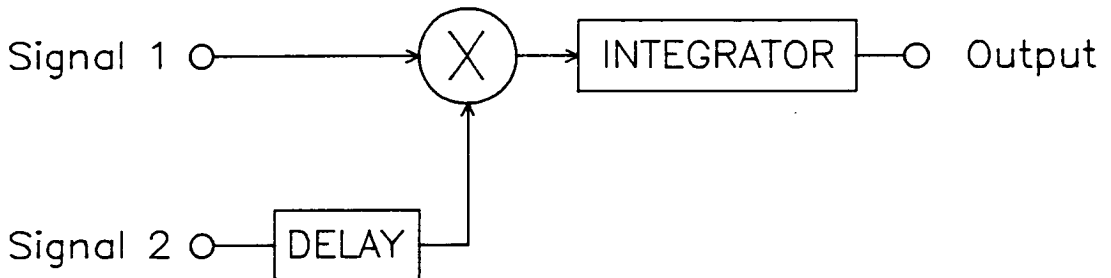


Figure 2.1. Basic elements of a correlator.

Direct implementation of the correlation function imposes a large processing cost. Consequently, considerable effort has been expended to devise approximations that will reduce the cost involved. Significant reductions are achieved when signals are converted to the sampled-data form, and the analogue integration process is replaced by one of summation. Further reductions follow when quantised signal representations are used. This chapter discusses the various forms of correlator which arise from the use of quantisation, of a varied degree, and the use of "dither" signals. In addition, the inevitable processing errors, which result from the necessary approximations to the ideal correlation coefficient, are examined. Firstly however, the interpretation of the correlation function shall be studied.

2.2. Interpreting the Correlation Function

Correlation functions may be divided into two categories: autocorrelation and crosscorrelation. The autocorrelation function $r_{xx}(\tau)$ of the time function $x(t)$ is defined as

$$r_{xx}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T x(t)x(t-\tau)dt \quad 2.1$$

where τ is a continuous time delay parameter. Autocorrelation represents a comparison of an input signal with a time delayed replica of itself. The autocorrelation function can yield useful information about the signal $x(t)$. For example, the value of the autocorrelation function at zero delay, is simply the mean square value σ_x^2 of the signal, that is,

$$r_{xx}(0) = \sigma_x^2 \quad 2.2$$

In addition, if the signal contains periodic components, then the resulting autocorrelation function will also exhibit periodic components. This feature is useful in recovering periodic signals buried in noise or other interference [13]. Other special properties of the autocorrelation function are

$$|r_{xx}(\tau)| \leq r_{xx}(0) \quad 2.3$$

and

$$r_{xx}(\tau) = r_{xx}(-\tau) \quad 2.4$$

A typical autocorrelation function is illustrated in Figure 2.2.

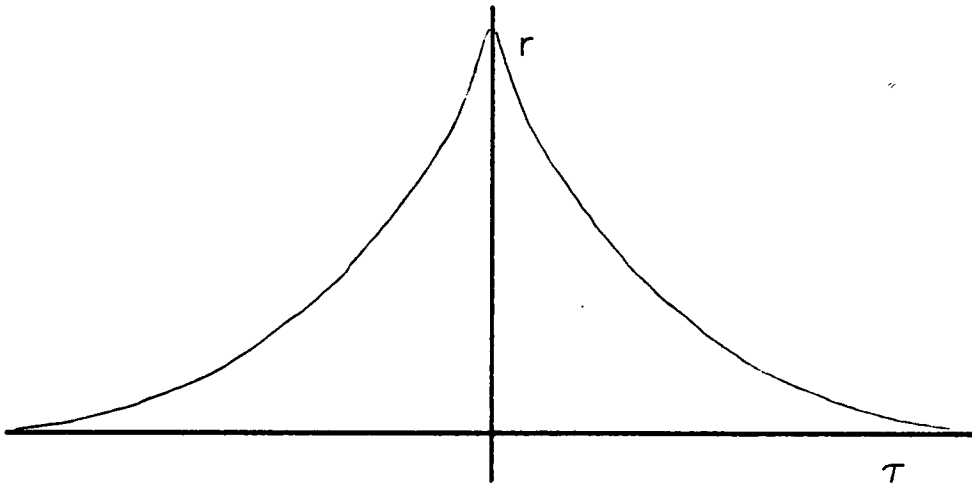


Figure 2.2. Autocorrelation function of a zero-mean signal.

The correlation between two signals $x(t)$ and $y(t)$ is given by the crosscorrelation function

$$r_{yx}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T y(t)x(t-\tau)dt \quad 2.5$$

where r_{yx} is simply the averaged product of y lagged with respect to x . When the value of the crosscorrelation is high for some value of lag τ , it can be said that x and y are similar, in some sense, at this lag value. Some special properties are

$$|r_{yx}(\tau)| \leq [r_{xx}(0)r_{yy}(0)]^{1/2} \quad 2.6$$

or

$$|r_{yx}(\tau)| \leq \sigma_x \sigma_y$$

where $\sigma_y^2 = r_{yy}(0)$, i.e. the mean square value of y , and

$$r_{yx}(\tau) \neq r_{yx}(-\tau) \text{ but } r_{yx}(\tau) = r_{xy}(-\tau) \quad 2.7$$

The most straightforward interpretation of the crosscorrelation function is in the context of time delay estimation [44,45,46,47]. Consider the propagation path shown in Figure 2.3.

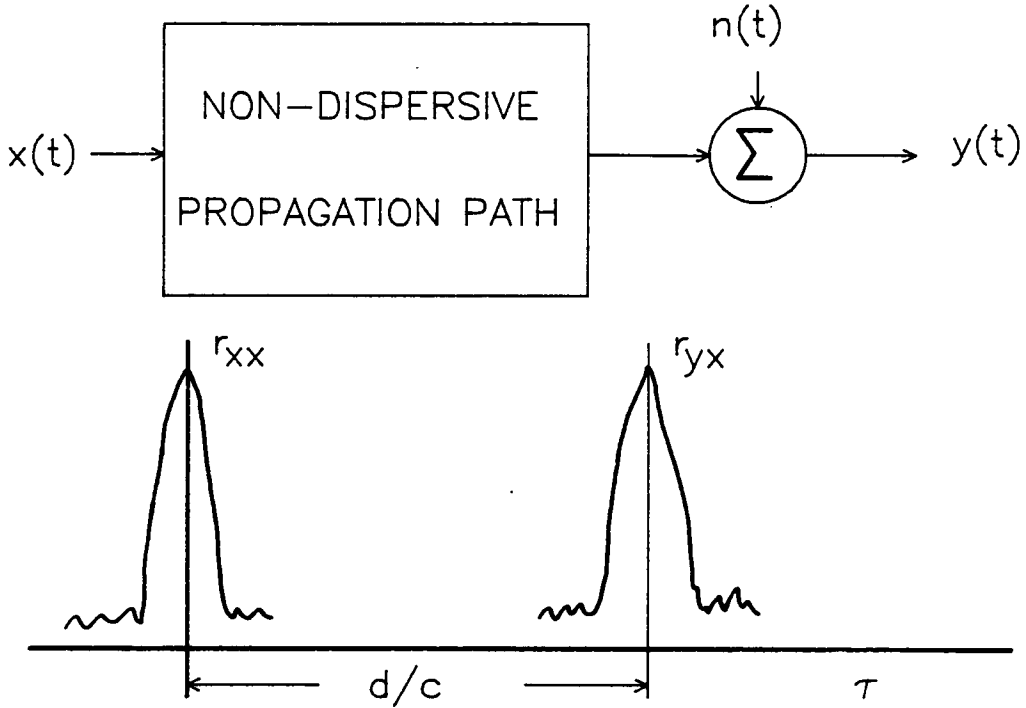


Figure 2.3. Non-dispersive propagation path.

In this example the signal, represented by $x(t)$, propagates through the nondispersive, linear path and combines with statistically independent noise $n(t)$, to produce the output response $y(t)$. Assuming, for simplicity, that the frequency response function of the propagation path is a constant $H(f) = K$, that the propagation distance is d , and that the propagation velocity is c , it follows that [4]

$$y(t) = Kx(t-d/c) + n(t) \quad 2.8$$

The crosscorrelation function between $x(t)$ and $y(t)$ is then

$$\begin{aligned} r_{yx}(\tau) &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T [Kx(t-d/c) + n(t)] \cdot x(t-\tau) dt & 2.9 \\ &= Kr_{xx}(\tau-d/c) \end{aligned}$$

So, in this simple example, the crosscorrelation function is given by the autocorrelation function of $x(t)$ multiplied by K and displaced in time to have a peak at $\tau_1 = d/c$. Thus, the crosscorrelation function can be used to determine either the distance d , the velocity c , or the time delay τ_1 of the propagation path. In realistic situations, such as in flow metering, the model is less straightforward. Turbulence in the flow causes the crosscorrelation to become asymmetrical about its maximum and adopt a skewed form [48,49], as shown in Figure 2.4.

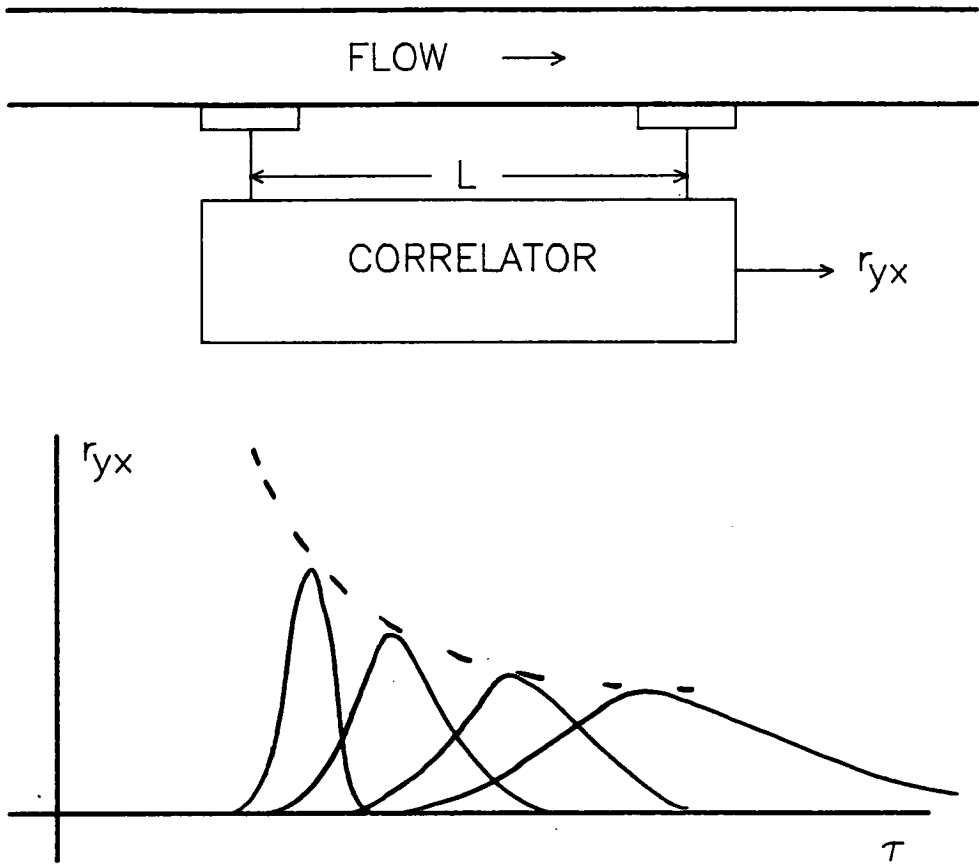


Figure 2.4. Skewing of crosscorrelation functions due to the effects of flow turbulence.

Normalised correlation functions are defined by the following expressions. Firstly, for autocorrelation,

$$r_{nxx}(\tau) = \frac{r_{xx}(\tau)}{r_{xx}(0)} , \quad -1 \leq r_{nxx}(\tau) \leq 1 \quad 2.10$$

and secondly, for crosscorrelation,

$$r_{nyx}(\tau) = \frac{r_{yx}(\tau)}{[r_{xx}(0)r_{yy}(0)]^{1/2}} , \quad -1 \leq r_{nyx}(\tau) \leq 1 \quad 2.11$$

where r_{nxx} and r_{nyx} are the normalised correlation

functions, and $r_{yy}(0)$ and $r_{xx}(0)$ are the mean square values of the signals y and x respectively.

Normalisation of the function makes interpretation clear when r_{nxx} or r_{nyx} equals ± 1.0 or zero. However when the result is less than 1.0 but greater than zero, the significance is less clear. To help with interpretation some associated functions are introduced. For the purposes of this thesis they are mentioned only briefly; a more detailed account is given by Roth [4]. Firstly, the correlation integral is closely related to the convolution integral. The only significant difference being the time reversal operation required by the convolution integral. For example,

$$T.r_{yx}(\tau) = y(t)*x(-t) \quad 2.12$$

where the star indicates the convolution of the two time functions. Another useful function is the cross-power spectrum, G_{yx} , which is the Fourier transform of the crosscorrelation function,

$$G_{yx} = F\{r_{yx}\} \quad 2.13$$

In addition, the cross-power spectrum may be obtained from the linear spectrums, thus:

$$G_{yx} = S_y S_x^* \quad 2.14$$

where $S_y = F\{y(t)\}$, $S_x = F\{x(t)\}$, and S^* indicates the complex conjugate of S . In addition, there are triple correlation functions, which are defined as the average product of the input signal at three instants in time, i.e. two time lags. Thus the triple correlation $r_{xxx}(\tau)$ is given by

$$r_{xxx}(\tau_1, \tau_2) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T x(t)x(t-\tau_1)x(t-\tau_2)dt \quad 2.15$$

A study of triple correlation is presented by Lohmann and Wirnitzer [50], and a correlator which can produce triple correlations has been reported by Corti et al [36]. In combination, therefore, these functions provide comprehensive and powerful tools for measurement and analysis.

2.3. Historical Development

In the 1950's, computation of the correlation function was performed using a variable delay line, a single multiplier, and a single integrator. The delay line had to be non-dispersive over the frequency range of interest. Various early methods using magnetic tape loops are reviewed by Cheney [51] and Lange [52]. In 1952 Brooks and Smith proposed a general purpose analogue computer for correlation functions [53]. The delay parameter is provided by staggered magnetic tape inputs. In the following year Bennett used a tapped delay line to replace the staggered tape inputs [54]. A complete integration period was required by these early systems for each successive value of the time delay τ , and although they produced accurate estimates of the correlation function, their computation time was too long for many applications.

A relentless demand for ever greater computational speed prompted the development of digital correlators. By quantising the input signals into two levels, the tasks of multiplication and integration become simple arithmetic procedures. Quantisation causes an increase in the variance of the output, but as we shall see later, the effect can be reduced by integrating over a longer time.

With the development of Large Scale Integration (LSI), it became more economically feasible to make tapped delay lines, and arrays of multipliers and integrators. This heralded an era of parallel processing where values

of the correlation function, for many values of the time delay parameter τ , are computed simultaneously. This improvement in processing power, made possible by LSI and VLSI, is essential for real time measurement and control applications. However, in each area of application of correlation analysis, there exists a need to compromise between sampling rate, the level of input quantisation, and the length of the integration period.

The remaining sections of this chapter deal with the fundamentals of correlation theory. The effects of sampling rate, input quantisation, and observation period are examined. To introduce the theory, a short summary of random data concepts is presented.

2.4. Correlation Principles

2.4.1. Random Data Concepts

Physical phenomena of interest in engineering are usually described in terms of amplitude versus time functions, known as "time history records". Many of these phenomena are "non-deterministic", or "random"; that is, each measurement produces a unique time history record which is not likely to be repeated, and cannot accurately be predicted.

In the case where the measurements of a physical phenomena are considered random, then the resulting time history record represents only one instance of what might have happened. To gain a fuller understanding of the phenomenon one must consider a set of all possible time history records that could have occurred. For example, a set of all time history records $x_i(t)$, $i=1,2,3,\dots$, is illustrated in Figure 2.5.

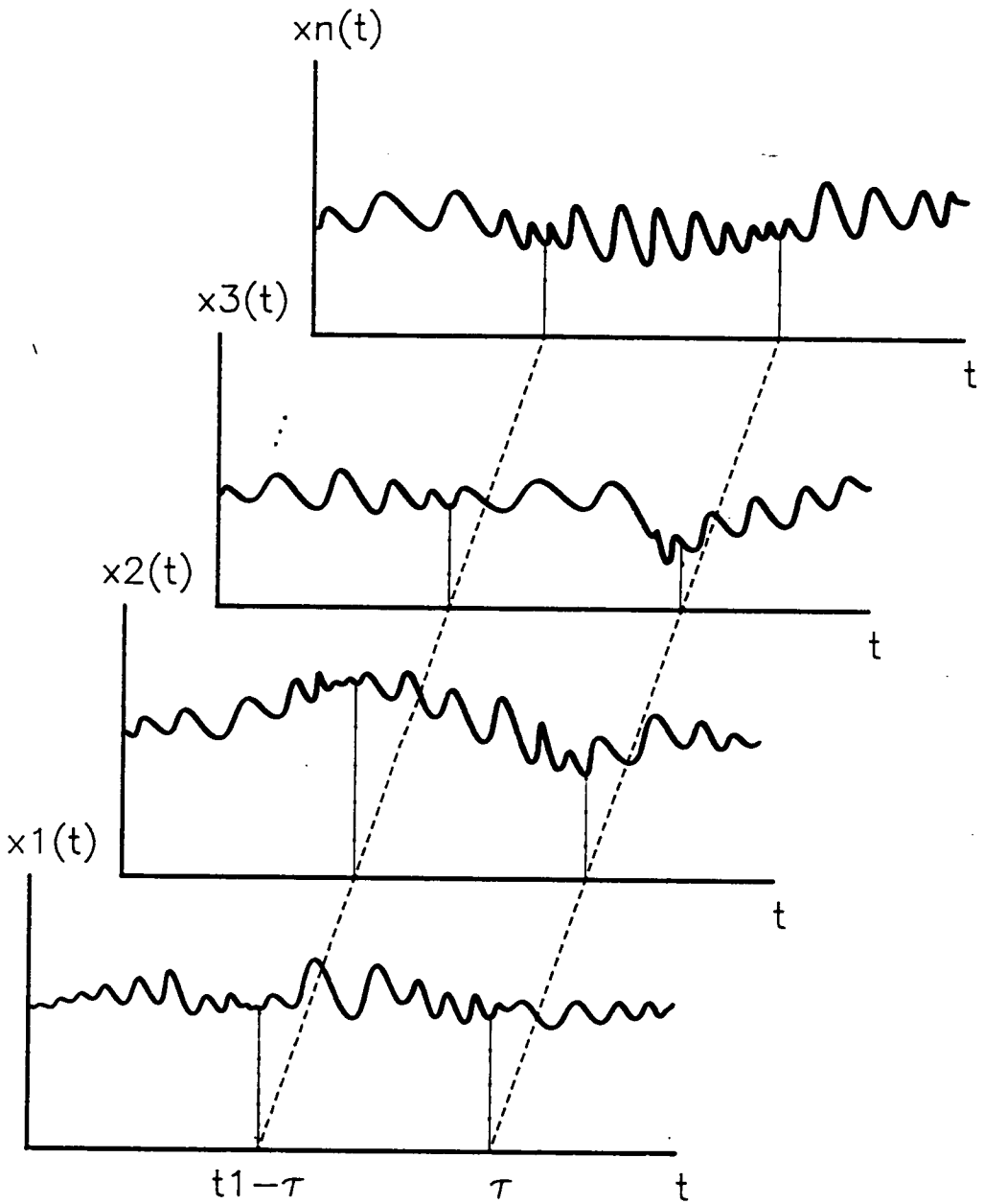


Figure 2.5. Ensemble of time history records defining a random process $\{x(t)\}$.

This is referred to as the "ensemble" that defines the random process $\{x(t)\}$. Given an ensemble of time history records, the average properties can be computed at any specific time t_1 and $t_1 - \tau$, that is, the autocorrelation

function at time delay τ , is given by

$$r_{xx}(t_1, \tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N x_i(t_1) \cdot x_i(t_1 - \tau) dt. \quad 2.16$$

In the general case, where one or more of the average values vary with time, the process is said to be "non-stationary". In the special case, where the average values are constant from one ensemble to the next, the process is said to be "stationary". For almost all stationary data the average values computed over the ensemble at time t_1 will equal the corresponding average values computed over all time from any single time history record. Thus the autocorrelation function may be written as

$$r_{xx}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T x(t)x(t-\tau)dt \quad 2.17$$

where $x(t)$ is any arbitrary record from the ensemble $\{x(t)\}$. The justification for the interchange of time and ensemble averaging is given by the ergodic hypothesis [55,56].

2.4.2. Fundamental Estimation Errors

In practice the number of data records available for analysis by ensemble averaging techniques, or the length of a data record used for analysis by time averaging techniques, will always be finite. Therefore, the average properties of the data can only be estimated and never computed exactly. As a result, certain errors arise. These are in addition to numerous potential errors from other sources, such as errors that might arise, for example, from input transducers, signal pre-processing, and analogue to digital conversion (quantisation).

The estimation errors can be divided into two classes: bias error and random error. Bias error is a systematic error that will appear with the same magnitude and in the same direction from one analysis to the next. Random error is a haphazard scatter in the results from one analysis to the next, of different samples from the same random data. It is a direct result of averaging over a finite number of time history records, or over a single record of finite length; it is therefore present in all analyses.

Random error is defined by the standard deviation of the estimate about its expected value, and it is often normalised to the parameter being estimated [3]. The normalised value is inversely proportional to the square root of the number of records N , or record length T . Hence to reduce the random error to half its value, the number of records, or the integration time, must be increased by a factor of four.

Also, for time averaged estimates, the normalised random error is inversely proportional to the square root of the data bandwidth B . This means that for those applications where the data bandwidth is very wide, as often occurs in communications, a relatively short record might provide highly accurate estimates. In contrast, for those applications where the data bandwidth is typically narrow, as occurs in studies of ocean movements or atmospheric turbulence, very long records may be required to obtain acceptably accurate results.

In the next section the discussion is extended to include the correlation function of discrete time sampled data, and then to discrete time quantised data.

2.4.3. Discrete Time Correlation

In the case of sampled inputs, the process of integration is replaced by one of summation, and Equations 2.1 and 2.5 may be rewritten as

$$r'_{xx}(k\Delta t) = \frac{1}{N} \sum_{n=0}^{N-1} x(n\Delta t) \cdot x(n\Delta t - k\Delta t) \quad 2.18$$

$$r'_{yx}(k\Delta t) = \frac{1}{N} \sum_{n=0}^{N-1} y(n\Delta t) \cdot x(n\Delta t - k\Delta t) \quad 2.19$$

where k , and n are integers. The notation $r'(k\Delta t)$ represents an approximation to the defined correlation function, but for convenience the approximation will be written simply as $r(k\Delta t)$. The analogue signals are normally sampled at equally spaced time intervals, Δt , with delay calculated at $k\Delta t$ intervals, where k is an integer from 0 to $K-1$, equal to the number of correlation points. In practice the maximum number of samples N (corresponding to the maximum integration time $N\Delta t$), is finite.

The sampling period Δt is related to the signal bandwidth B and the number of correlation points required to define the peak of the function. If we assume that we require p points within the peak region to define the peak position adequately, then the sampling period Δt is given by [57]

$$\Delta t = \frac{1}{B(p+1)} \quad 2.20$$

In certain applications, such as flow rate measurement or time delay estimation, a reduction can be achieved in the number of delay increments required to implement the function. At minimum flow velocities (that is, maximum time delays), the number of points computed to define

the peak far exceeds the number required to determine the position of the peak accurately. The amount of redundant information in these situations can be reduced by increasing the time delay increment at longer correlated time delays [58,16]. Alternatively, a variable sampling rate, which is derived from the flow velocity, may be used, although this approach is unreliable when there is a step change in the flow velocity [59].

As we have already seen, there are several sources of estimation error. Finite averaging time, finite bandwidth, noise, waveform sampling, and waveform quantisation all contribute to the variance of the result. Expressions relating the variance to the averaging time, bandwidth, and mean square signal to noise ratios have been derived theoretically and confirmed experimentally by several authors [60,61,16]. Sampling and quantisation both introduce noise and, in addition, sampling can limit bandwidth.

Intuitively, one would expect the accuracy of the correlation estimate to increase with increasing sampling rate. However, Kay [61] has shown that, for long averaging times (time bandwidth product greater than 25), the variance does not significantly reduce as the sampling rate increases. For short averaging times (time bandwidth product less than 25), the variance does reduce when the analogue waveform is sampled faster than the Nyquist rate. Sampling at twice the Nyquist rate appears to be a good compromise between the desires of minimising the mean square error and of maintaining a low sampling rate. This result concurs with the earlier analysis of Bowers et al [62].

2.5. Correlation Techniques

Direct implementation of the correlation function imposes a large processing cost. Considerable effort has been expended to devise approximations that will reduce this cost. Notable reductions are achieved when signals are converted to sampled-data form, and the analogue integration process is replaced by one of summation. Further reductions follow when quantised signal representations are used. This section is concerned with the classes of correlator which arise from the use of quantisation and dither. In Chapter 3, where integrated circuit correlators are reviewed, another facet of correlator implementation is introduced: that is, computation using parallel techniques, serial techniques, or a combination of the two.

Four basic types of correlator, resulting from the use of quantisation and dither, can be defined as follows:

1. Direct correlators, where both inputs are analogue.
2. Stieltjes correlators [63,64,65], where one input is quantised and the other is analogue. The relay correlator is a limiting case, where the digital channel is quantised to just two levels, +1 and -1.
3. Digital correlators, where both inputs are quantised. The polarity correlator is the limiting case of this class of correlator, where both input signals are quantised to two levels, +1 and -1, before correlation.
4. Modified correlators, where a dither signal is added to the digital input or inputs. The modified relay correlator is a special case of the modified Stieltjes correlator, and the modified polarity

correlator is the limiting case of modified digital correlators.

These classes of correlators are described in more detail in the following sections. Firstly, however, a brief discussion about quantisation is presented.

2.5.1. Quantisation of Input Data

Quantisation is the process of replacing analogue samples with approximate values taken from a finite set of allowed values [66]. Quantisation is employed in the design of correlators so that the benefits of digital circuit techniques may be exploited. It will be seen that the disadvantages arising from the errors introduced by quantisation, are often compensated by the reduction in correlator's cost and complexity.

There are various forms of quantiser, the more sophisticated of which add a minimum of distortion or quantisation noise to the signal. The simplest and most common form is the zero-memory quantiser. In this case, the output value is determined from only one corresponding input sample, independent of the values of earlier (or later) analogue samples applied to the quantiser input. More sophisticated, is the block quantiser which looks at a group, or block, of input samples simultaneously, and produces a block of output values to represent the corresponding input samples. Another class of quantisers, which could be described as sequential, includes digitising schemes such as delta modulation, differential PCM, and other adaptive versions. A sequential quantiser stores some information about previous samples and generates the present quantised output using both the current input and the stored information.

For the purposes of this thesis, we need not be concerned further with the details of quantisation, other than its effect on the realisation and accuracy of the correlation function. In this respect the most important parameter of quantisation is, the number of quanta, or quantum levels allocated to each input of the correlator. For example, coarse quantisation in both inputs of a correlator permits the use of much less complex circuits for multiplication and summation etc., than would be the case for one with finely quantised data. On the other hand, coarse quantisation leads to a degradation in the accuracy of the correlation function. However, the degradation in the output can be eliminated by averaging over a longer period, since the errors are essentially random. But, in the case of extremely coarse quantisation, i.e. two levels, significant bias errors are incurred which cannot be removed by simply extending the integration period. These bias errors are eliminated by the use of dither, or auxiliary signals [67,68]. A dither signal is added to the input of a digital correlator before quantisation. Unfortunately, dither signals introduce an additional source of random errors into the system, which, in turn, must be eliminated by integrating over a longer time [69].

Another form of quantisation, delta sigma quantisation, is the basis of a separate class of correlators. Delta sigma correlators are described by several authors [64,70,71,72,73,74], but they are beyond the scope of this thesis.

Quantisation can have a significant effect on the complexity of correlators. In 1962, Watts [75] presented a detailed analysis of the effect that quantisation has on correlator performance and derived a general form for multiplier correlators. The direct analogue correlator, the

digital correlator, and the Stieltjes correlator are all shown to be special cases of the general form.

Amplitude quantisation is a non-linear process. When such a non-linear operation is incorporated into a system, detailed analytical analysis of the system is made extremely difficult. The statistical analysis of such a system can be relatively easy because it is possible to investigate the statistical effects of quantisation in detail. It can be shown that, for many cases, quantisation is equivalent to the addition of random independent noise with a mean square value equal to one-twelfth the square of the quantisation interval [76]. Thus, the quantised signal x^+ is considered to be equal to the original signal x , plus the additive quantisation noise a . For example, $x^+ = x + a$, $y^+ = y + b$. The correlation function of two quantised signals x^+ and y^+ (with zero means) may then be expressed as

$$r_{y^+x^+} = r_{yx} + r_{ya} + r_{bx} + r_{ba} \quad 2.21$$

where r_{yx} is the correlation between the signals y and x , r_{ya} is the correlation between the signal y and the quantisation noise a , r_{bx} is the correlation between the signal x and the quantisation noise b , and r_{ba} is the correlation between the quantisation noise a and the quantisation noise b .

2.5.2. Direct Analogue Correlation

Analogue or continuous correlators are those correlators in which the signals are processed directly, without any form of amplitude distortion being used. They have been termed ideal correlators because, with the same input signals and noise, their signal to noise ratio is not exceeded by any other form of correlator. However, they

possess important disadvantages such as drift. The implementation of analogue correlators has been described in Section 2.3. The analogue multipliers used in them can be realised using transistor circuits, and the delay operation can be performed by LC circuits, or tape recorder systems. The integration can be achieved using current summing amplifiers or low pass filters.

When the analogue signals are sampled at the Nyquist rate, or faster, sampled data techniques, such as charge coupled devices, may be employed. Analogue correlators realised using integrated circuit techniques are described in Chapter 3.

2.5.3. Stieltjes Correlation

The Stieltjes correlator is a special form of the general configuration, in which one of the inputs is analogue and the other is coarsely quantised [63,64,65]. Since only one of the inputs is quantised, the output of a Stieltjes correlator is

$$r_{syx} = r_{yx} + r_{bx} \quad 2.22$$

The only error term is the term r_{bx} , which, even for quite coarse quantisation of y , can be extremely small. Watts [75] has shown that when the digital channel is quantised into three levels, the Stieltjes correlation function is related to the direct correlation function to within 1%. The circuitry required to implement this correlator represents a considerable saving in complexity when compared with direct correlation [77]. The correlation delay is implemented digitally in one channel. The multiplication may be performed by a digital-to-analogue converter with the analogue signal as its reference. A disadvantage of the Stieltjes correlator, as with all analogue

correlators, are the difficulties concerning drift.

2.5.4. Relay Correlation

A special case of the Stieltjes correlator is the relay correlator, which is illustrated in Figure 2.6.

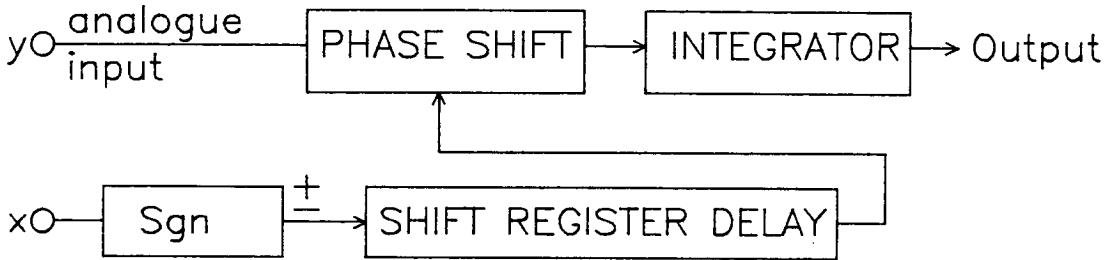


Figure 2.6. Basic configuration of a relay correlator.

In this case one of the inputs is quantised into two levels, denoted by the "sgn" operator, and the other is analogue. Sgn(x) means signum(x), a function of the value +1 for positive x and -1 for negative x. The output of the relay correlator r_{ryx} , for sampled inputs with Gaussian statistics, is related to the direct correlation function by

$$r_{ryx}(k\Delta t) = [2/\pi]^{1/2} r_{nyx}(\tau) \sigma_y \quad 2.23$$

where

$$r_{ryx}(k\Delta t) = \frac{1}{N} \sum_{n=0}^{N-1} y(n\Delta t) \cdot \text{sgn}[x(n\Delta t - k\Delta t)] \quad 2.24$$

and σ_y is the RMS value of the signal y, given by $[r_{yy}(0)]^{1/2}$. The relay correlator represents a compromise between the direct digital correlator and the polarity-coincidence correlator, both in terms of accuracy, and in terms of circuit complexity. To achieve results of the

same level of accuracy as the ideal analogue correlator, the integration time must be approximately 1.5 times longer.

2.5.5. Multilevel Digital Correlation

Direct digital correlation, or multilevel correlation, in which quantisation is done using more than two levels per channel, is illustrated in Figure 2.7.

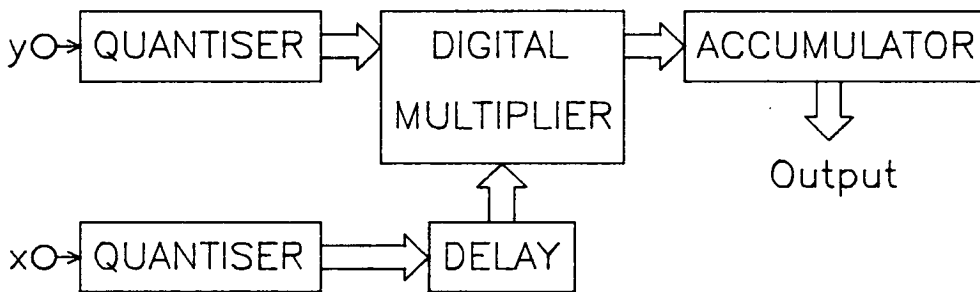


Figure 2.7. Basic configuration of digital correlator.

It has been shown that [76], in many cases, even for fairly large quantisation intervals (for example, the total range of x or y divided into eight intervals), the terms r_{ya} and r_{bx} , defined above, are negligible, and the term r_{ba} is also negligible, except when x equals y , in which case $r_{ba} = q^2/12$, where q is the quantisation interval. This is known as the Sheppard correction to the mean square for grouped data [78]. This correction is a simplification of a more general expression, given by Gersho [66], and assumes that the intervals of quantisation q_i are equal, that is, $q_i = q$, where i is an integer from 0 to $L-1$ in an L level quantiser. Thus, the output of the direct digital correlator r_{dyx} , using uniform quantisation, may be taken to be

$$\begin{aligned}
 r_{dyx} &= r_{yx} + q^2/12, & x=y, \tau=0 \\
 &= r_{yx}, & x \neq y, \text{ all } \tau \\
 &= r_{yx}, & x=y, \tau \neq 0
 \end{aligned} \tag{2.25}$$

The hardware realisation of a direct digital correlator involves complex digital circuitry but results in an accurate correlation estimate. A description of a 2-bit by 2-bit digital correlator for measuring the spectra of radio astronomy signals is given by Ables et al [79], and the Hewlett Packard correlator [80], which quantises the input signals into three levels in one channel and seven levels in the other, has been used successfully for many years. Another digital correlator, which resembles a 3-level by 3-level correlator is presented by Dewdney [81]. In this case the circuit complexity is reduced by accumulating the product transitions rather than the products themselves. The penalty incurred by this technique is a six percent decrease in output signal to noise ratio, when compared with a normal three level correlator. The loss in signal to noise ratio may be recovered by increasing the integration time, since integration time is proportional to the square of the signal to noise ratio. A special case of multilevel correlation is digital relay correlation. This is in addition to polarity correlation, which is discussed in the next section. A digital relay correlator, illustrated in Figure 2.8, averages the product of the quantised values of the y-input, and the polarity of the x-input, using a digital adder/subtractor and digital store.

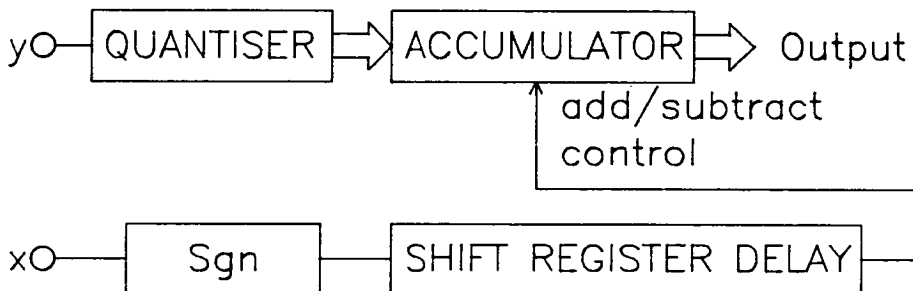


Figure 2.8. Configuration of a digital relay correlator.

The circuit complexity represents a compromise between full multilevel correlation and polarity correlation.

2.5.6. Polarity-Coincidence Correlation

A special case of digital correlation is the polarity-coincidence correlator, shown in Figure 2.9.

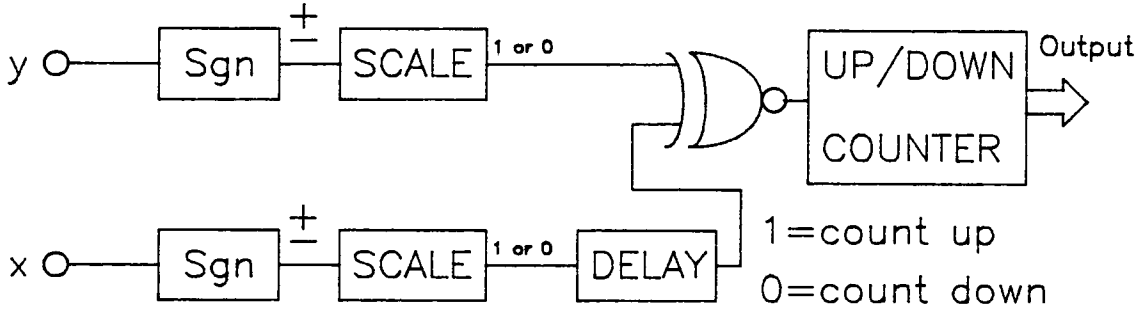


Figure 2.9. Configuration of a polarity correlator.

In this case the inputs are quantised to two levels, ± 1 , denoted by the sgn operator, as above. If the input signals have Gaussian statistics, the polarity correlation function r_{pyx} may be related to the direct correlation function by

$$r_{pyx}(k\Delta t) = \frac{2}{\pi} \arcsin r_{nyx}(\tau) \quad 2.26$$

where

$$r_{pyx}(k\Delta t) = \frac{1}{N} \sum_{n=0}^{N-1} \text{sgn}[y(n\Delta t)] \cdot \text{sgn}[x(n\Delta t - k\Delta t)] \quad 2.27$$

and $r_{nyx}(\tau)$ is the normalised correlation function of the signals y and x , and τ is the time lag between the two signals, as given in Equation 2.11. The arcsine relationship was first reported by Van Vleck in 1943 and subsequently by Van Vleck and Middleton in 1966 [82]. The hardware realisation of the polarity-coincidence

correlator is very much more simple than the direct digital correlator. The signal delay is implemented by a single-bit shift-register, multiplication is achieved by exclusive-NOR gates, and the integration process is performed by simple counters. The correlation estimate obtained from a polarity correlator is less accurate than one obtained from a direct digital correlator [83], and accordingly requires an integration time which is approximately 2.5 times longer, to achieve the same level of accuracy. Polarity correlation is treated in more detail in Section 2.6.

2.5.7. Modified Correlators: Dither

The significant bias errors incurred when extreme clipping operations, such as $\text{sgn}(x)$, are used, can be eliminated by adding a dither signal to the signal to be clipped, as shown in Figure 2.10.

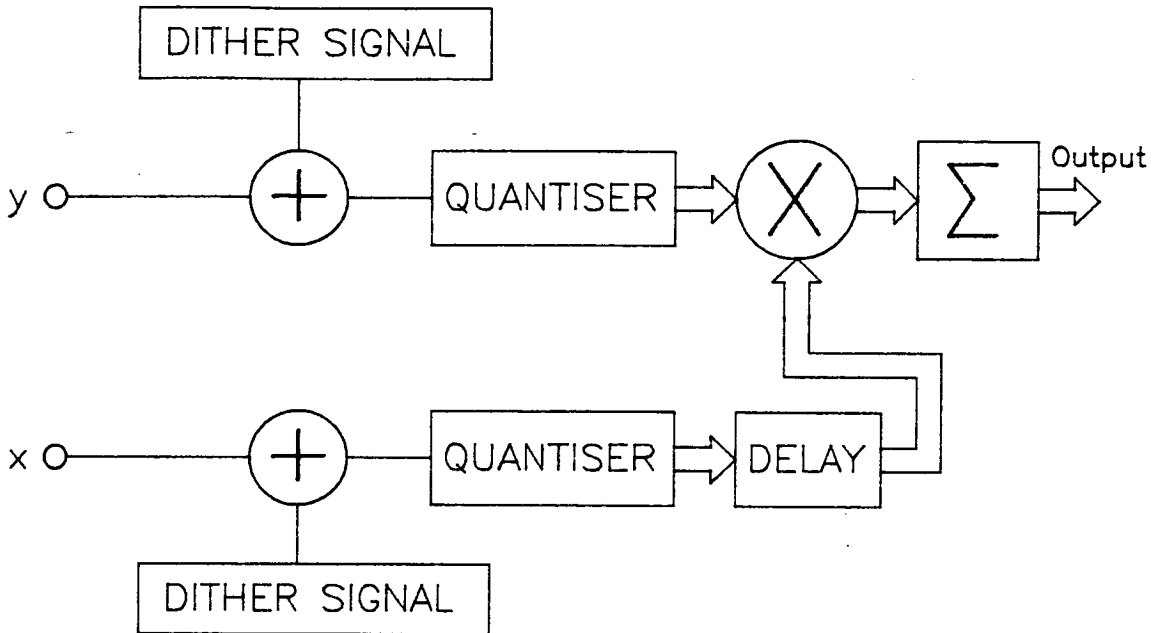


Figure 2.10. Configuration of a modified digital correlator.

It can be shown that

$$r_{pyx}(n\Delta t) = \frac{1}{A^2} \frac{r_{yx}(\tau)}{[r_{yy}(0)r_{xx}(0)]^{1/2}} \quad 2.28$$

where the signal input magnitude must be maintained equal to or less than the upper bound A on the amplitude of the dither signal. A detailed analysis of modified correlators has been presented by Berndt [84], and by Chang and Moore [67]. Landsberg and Cohen [85] have reported a modified digital correlator which uses three levels of quantisation in both channels.

A polarity correlator can be modified to give an unbiased output, and is applicable to any random process with bounded inputs. This modification is achieved by

adding uniformly distributed, statistically independent noise to each of the input signals before they are clipped. A wide range of random dither signals have been used to modify correlators, but it has been found, subsequently, that deterministic signals can be used if they have uniformly distributed amplitude values [86,68,67]. Dither signals have found numerous applications in fields such as communications, where it enables capture of a wanted signal despite the presence of unwanted interference [87], and control, where it improves the performance of quantised sampled-data systems.

2.6. Polarity Correlation and the Overloading Integrating Counter Technique

2.6.1. Polarity Correlation

Implementation of a high speed correlator requires an array of multipliers, delay elements, and accumulators, either analogue or digital. Polarity correlation methods minimise the complexity of the computational elements by discarding the magnitude information of the input sequences. Digital design techniques can then be employed to realise the multipliers by EXNOR gates, the delay elements by a digital shift register and the accumulators by simple counting circuits. This results in a more economical and more compact implementation than would otherwise be achieved, the penalty for which is an increase in integration time to obtain a correlation function with acceptable variance [88]. The polarity correlation function is nonlinearly related to the direct correlation function by the Van Vleck arc sine relation, Equation 2.26, for input sequences which have Gaussian statistics.

In Chapter 3 details are presented of previously reported techniques for obtaining the polarity correlation

function [89,90,91,92,93]. These techniques include parallel counters [94,95,96,97], which are not directly cascadable and hence non-optimal for VLSI implementation. The prototype chip described here is based on an interpretation of the polarity correlation function which permits the elimination of parallel counters and results in a highly regular correlator structure amenable to VLSI implementation. The structure also permits direct cascading of correlator stages. A block diagram of a correlator using this approach is shown in Figure 2.11.

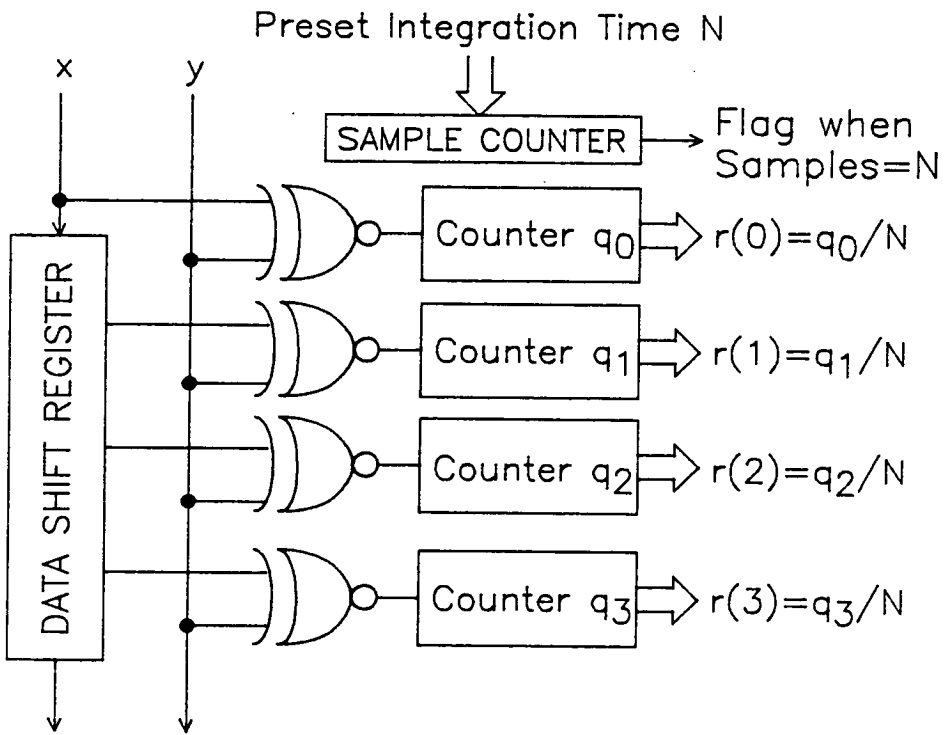


Figure 2.11. Block diagram of a polarity correlator.

Polarity correlation is based on the computation of the discrete function,

$$r_{pyx}(\tau) = \frac{1}{N} \sum_{n=0}^{N-1} (\text{sgn}[y_n] \cdot \text{sgn}[x_{n-\tau}]) \quad 2.29$$

which is based on Equation 2.27, but, for convenience, the

time lag $k\Delta t$ is replaced by the symbol τ , and the sampled signals of the form $y(n\Delta t)$ are replaced by a data sequence y with sequence index n . Complete positive correlation ($r_{pyx} = 1$) occurs when the polarities of the input samples (assuming the mean of both inputs to be zero) are at all times equal, yielding an average product of +1. Complete negative correlation ($r_{pyx} = -1$) occurs when the polarities of the input samples are never equal (inverse proportionality), yielding an average product of -1. In the case where the input samples are not related ($r_{pyx} = 0$) the sum of the positive products will equal the sum of the negative products and the average product will be zero.

Implementation of polarity correlation requires an analogue comparator circuit to convert $\text{sgn}[x]=x/|x|$ and $\text{sgn}[y]=y/|y|$ into logic 1 if the signal is positive and logic 0 if the signal is negative. Note that this definition means that a logic 0 represents -1 (see Section 2.5.6). The time delay τ between the two signals is achieved by using a digital shift register where a particular value of delay is defined by the product of the number of preceding shift register stages and the sample clock period, Δt . Multiplication is performed by the Boolean coincidence function, EXNOR, whose output is 1 only if the inputs are both equal. If time-successive values of the coincidence function $F_n(\tau)$ are summed in a digital counting circuit for a period T seconds, where $T = N\Delta t$, then the contents of the counter at the end of the period will be proportional to the relevant value of the correlation function. The EXNOR function can only be regarded as performing multiplication if the logic 0 is allowed to represent -1. Thus, a logic 1 in the coincidence signal would indicate 'increment by one' the contents of the counter and a logic 0 would indicate 'decrement by one' the contents of the counter. This would

necessitate the use of up-down counters which are undesirable from a VLSI circuit design point of view. However, it is possible to use simple up-counters whose contents, $q(\tau)$, can be related to the correlation function in the following way. Firstly the contents of an integrating counter are given by

$$q(\tau) = \sum_{n=0}^{N-1} F_n(\tau) \quad 2.30$$

where $F_n(\tau)$ is the coincidence function bit stream defined by

$$\begin{aligned} F_n(\tau) &= \frac{1}{2} + \frac{1}{2} \text{sgn}[y_n] \cdot \text{sgn}[x_{n-\tau}] \\ &= 1 \text{ or } 0 \end{aligned} \quad 2.31$$

Thus, by substituting into 2.30,

$$q(\tau) = \sum_{n=0}^{N-1} \frac{1}{2} + \sum_{n=0}^{N-1} \frac{1}{2} \text{sgn}[y_n] \cdot \text{sgn}[x_{n-\tau}] \quad 2.32$$

$$= \frac{N}{2} + \frac{N}{2} \cdot r_{\text{pyx}}(\tau) \quad 2.33$$

Hence,

$$r_{\text{pyx}}(\tau) = 2 \frac{q(\tau)}{N} - 1 \quad 2.34$$

where $r_{\text{pyx}}(\tau)$ is the polarity correlation function as given by Equation 2.29. Thus, Equation 2.34 gives a measure of the correlation function using the integration counter contents, $q(\tau)$, after sampling N times. At maximum positive correlation ($r_{\text{pyx}} = +1$) a maximum count $q(\tau) = N$ is obtained after sampling N times. In the case

of maximum negative correlation ($r_{pyx} = -1$), where the input samples are never equal, the coincidence signal is always zero, resulting in a zero count, $q(\tau) = 0$. In the case of zero correlation ($r_{pyx} = 0$), a count of $q(\tau) = N/2$ is reached after sampling N times.

2.6.2. Overloading Counter Technique

An alternative approach to polarity correlation is based on an integrating overloading counter technique [98,99,2], which eliminates the requirement for a value of $q(\tau)$ to be available. Instead, the correlation function is computed using the number of samples required to achieve overload count conditions, $q(\tau) = N$, in a given integrating counter. The concept of the technique is illustrated by Figure 2.12, which shows the relationship between the contents of an integrating counter, $q(\tau)$, and the number of samples, which is now a variable, m .

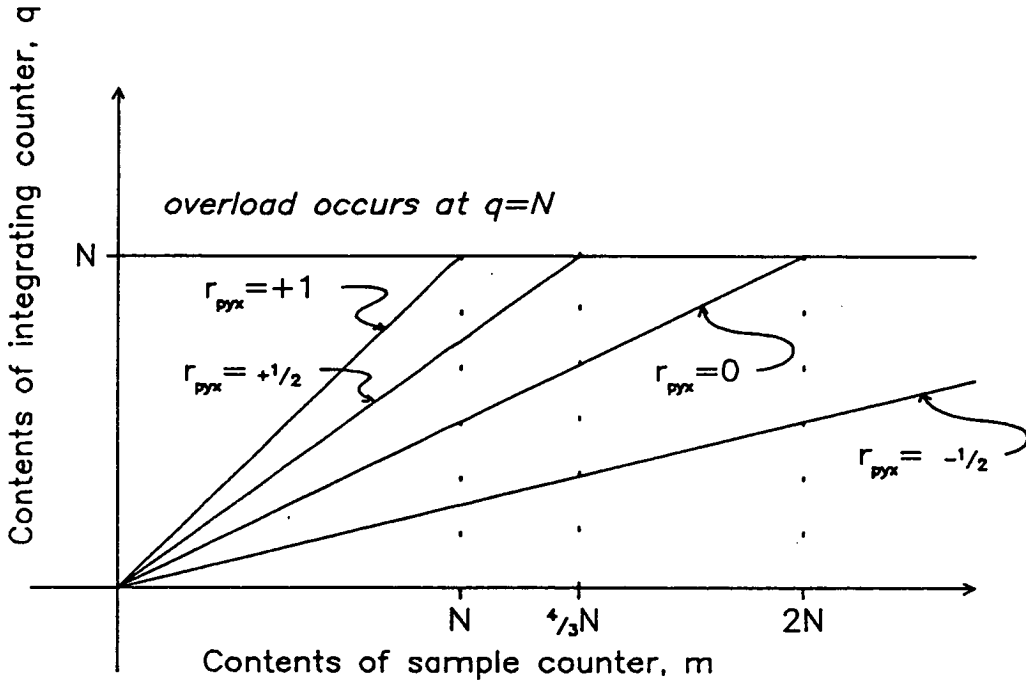


Figure 2.12. Relationship between the contents of the integrating counters and the number of input sample pairs.

The number of samples, m , can be related to the polarity correlation function by writing $q(\tau)$ as,

$$q(\tau) = N = \sum_{n=0}^{m-1} \left(\frac{1}{2} + \frac{1}{2} \text{sgn}[y_n] \cdot \text{sgn}[x_{n-\tau}] \right) \quad 2.35$$

$$= \frac{m}{2} + \frac{m}{2} \cdot r_{pyx}(\tau) \quad 2.36$$

where

$$r_{pyx}(\tau) = \frac{1}{m} \sum_{n=0}^{m-1} (\text{sgn}[y_n] \cdot \text{sgn}[x_{n-\tau}]) \quad 2.37$$

Hence, in this case,

$$r_{pyx}(\tau) = 2 \frac{N}{m} - 1, \quad \text{for } m \geq N \quad 2.38$$

where N is the *capacity* of the integrating counters and m is the number of samples required to achieve overload conditions in the integrating counter corresponding to time delay τ . An overload occurs after $m=N$ samples when correlation is maximum and positive. In the case of zero correlation an overload occurs after $m=2N$ samples and after an infinite number of samples when the correlation is maximum and negative. Note that an overload cannot occur until $m \geq N$.

A polarity correlator using the overloading counter technique thus comprises a delaying shift register connected to a parallel array of coincidence detectors and integrating counters. A block diagram of a polarity correlator using the overloading counter technique is shown in Figure 2.13.

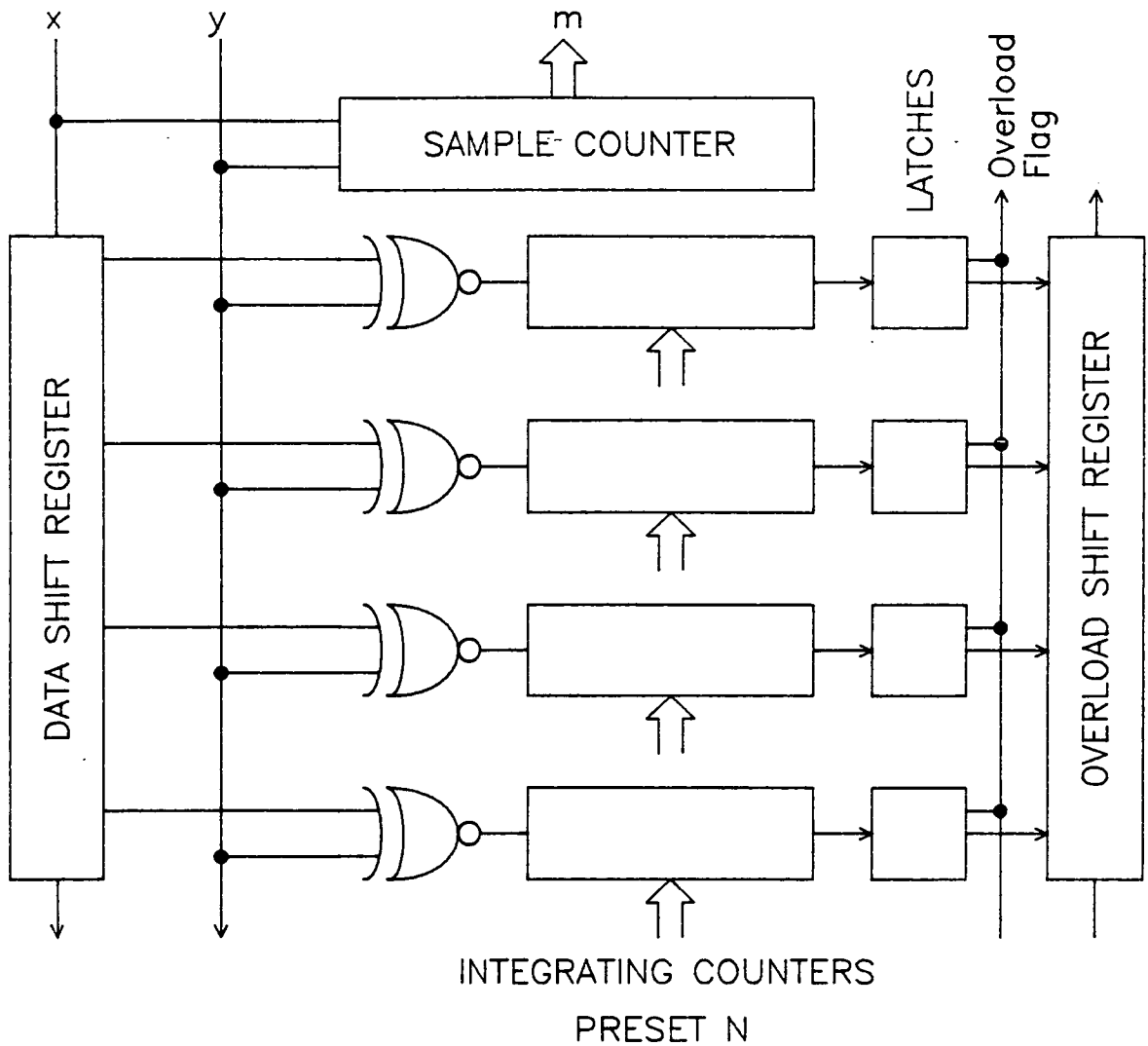


Figure 2.13. Polarity correlator block diagram using the overloading counter technique.

An overload pattern shift register is used to inspect the overload condition of the counters. The evolving pattern of overload states defines the correlation function shape and the time delay position of the first integrating counter to overload defines the position of the most significant peak of the function. A sample counter is included to count the number of input samples, m , so that the value of the correlation function may be computed for

any integrating counter to overload. If the maximum capacity of the sample counter is set to be twice the capacity of the integrating counters the significance range is limited to $1 \geq r \geq 0$. If it is required to cover the range $1 \geq r \geq -1$, two correlator circuits working in parallel can be used with one covering the positive range and the other covering the negative range.

Such a system is most suitably realised using integrated circuit technology and an early device implemented 12 stages of correlation using pMOS technology [28]. The correlator chip described in this thesis, consists of a linear cascade of identical correlation elements, which has been fabricated in 5 micron nMOS technology. The performance of the correlator depends on the serial connection of correctly functioning correlation elements. To optimise performance, and gain full advantage of the VLSI architecture, a design strategy was adopted, which includes testability, yield enhancement, and improves reliability. The design incorporates built-in self test (BIST) and self repair mechanisms, which automatically detect and eliminate failed correlation stages in the VLSI circuit [100,101,102,103].

2.7. Summary

In this chapter, correlation theory has been presented. It has been shown that, for stationary, ergodic signals, a temporal correlation function with finite integration time can approximate the true correlation coefficient. The effects of sampling, quantisation, and dither have been described. The main conclusion is that any physically realisable correlation system must compromise accuracy with integration time, or circuit complexity.

The overloading integrating counter technique for

polarity correlation has also been described, and the prototype correlator chip, featuring built-in self test and self repair mechanisms, has been introduced. Design details of a 28 stage prototype chip (termed the Eu349) are reported in Chapter 5. In the next chapter a review of silicon integrated circuit correlators is presented, in which the Eu349 chip's architecture, and how it relates to other integrated circuit correlators, is discussed.

CHAPTER 3

INTEGRATED CIRCUIT CORRELATORS

3.1. Introduction

Devices for computing correlation functions have been implemented using a variety of technologies and techniques. They span the entire gamut of signal processing techniques from optical signal processors to microcomputer systems; from surface acoustic wave devices to charge coupled devices; and from electronic systems built with small scale and medium scale integrated circuits, to full custom VLSI processors. This chapter reviews correlation techniques which have been realised by silicon integrated circuits. Implementations based on optical techniques [104,105,106,107], acousto-optical techniques [108,109], ultrasonic and surface acoustic wave (SAW) techniques [110,111,42] are beyond the scope of this discussion. So too are integrated optical correlators [112], which have received considerable attention and will find applications in parallel array signal processing problems such as real time image processing. Also excluded from this discussion are the microprocessor based correlators. These, in general, use a microprocessor to control a dedicated peripheral circuit which performs the delay, multiply, and accumulate operations [113,114,59,115]. In some cases however algorithms are used which allow the microprocessor to compute the correlation function with a minimum of additional circuitry. Examples of this are the zero crossing algorithm of Henry [116], and the skip algorithm of Fell [117].

In the remaining sections of this chapter silicon correlators are discussed. The architectural concepts, which distinguish the VLSI implementations, include serial correlators, parallel correlators, and serial/parallel correlators. The discussion also includes systolic arrays and examples are given for bit-systolic, word-systolic, linear, and two-dimensional systolic architectures.

3.2. Correlation Architectures

3.2.1. Serial Architecture

The basic elements of a correlator are shown in Figure 3.1. In a serial correlator, this configuration

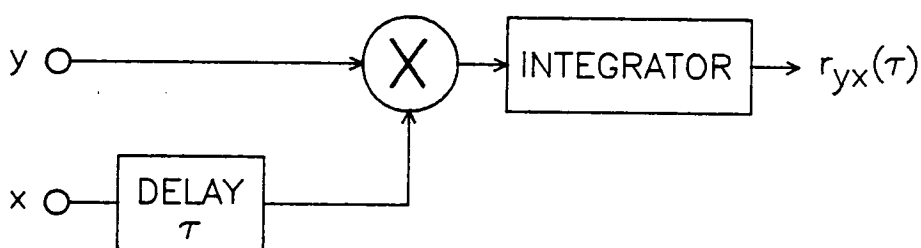


Figure 3.1. Serial correlator.

is implemented directly, and its operation is straightforward. The underlying principle can be described in terms of a *temporal* correlation lag and a *temporal* integration (architectures which implement *spatial* lags or *spatial* integrations are discussed in the next section). By making use of temporal techniques, the serial correlator minimises the circuitry required to implement the function. However the penalty for this simplicity is a long processing time. To compute, for example $r(\tau_1)$, the delay element is first set to the value τ_1 . Then, the input data sequences are multiplied, and the results are integrated. After the integration period, computation for

this single point of the function is complete. The entire computation is repeated for the next value of correlation delay, hence the term "temporal lag and temporal delay". Due to the long processing time, serial correlators are not common in VLSI implementations, although if the signal bandwidth is large and has stationary characteristics, then serial correlation is useful and very simple to implement. One instance of an integrated serial correlator, designed to verify a correlation algorithm which uses a pseudorandom dither signal, has been reported [86].

In general, there are two ways to increase the computation rate of a signal processing system. One is to use faster components and the other is to use concurrency. The last decade has seen an order of magnitude decrease in the cost and size of integrated circuit components, but only an incremental increase in component speed. With current technology, tens of thousands of gates can be fabricated on a single chip, but no gate is much faster than its TTL (Transistor-Transistor Logic) counterpart of ten years ago. Since the technology trend indicates a diminishing growth rate for component speed, any major improvement in computation rate must come from the concurrent use of many processing elements. The degree of concurrency is largely determined by the underlying algorithm. Optimum performance can be achieved when the algorithm is designed for the most effective degrees of pipelining and multiprocessing [118]. However, it must be noted that, when a large number of processing elements work simultaneously, coordination and communication problems become significant [119,120]. The objective, therefore, is to design algorithms which allow high degrees of concurrency, while employing only simple, regular communications and control. Direct cascading of cells for system expansion is also important. Systolic architectures, introduced by Kung and Leirserson [121], provide a

solution to the above objectives.

A systolic system consists of a set of synchronously clocked, interconnected cells, each of which are capable of performing some simple operation. The cells are usually connected together to form a systolic array or systolic tree. Information flows between cells in a pipelined fashion and communication with the outside world occurs only at the boundary of the array. Features to avoid in the design of a systolic system are global broadcasting of signals across the array, and fan-in of many outputs to a single computational element [122]. These criteria will be illustrated by the correlator architectures in the remaining sections of this chapter.

3.2.2. Parallel Architecture

In the previous section, a serial correlator was described as employing temporal delay and temporal integration. The first parallel correlation architecture to be discussed here achieves concurrency by replacing the temporal delay with spatial delay.

3.2.2.1. Parallel architecture with temporal integration and spatial delay

This parallel architecture is shown in Figure 3.2.

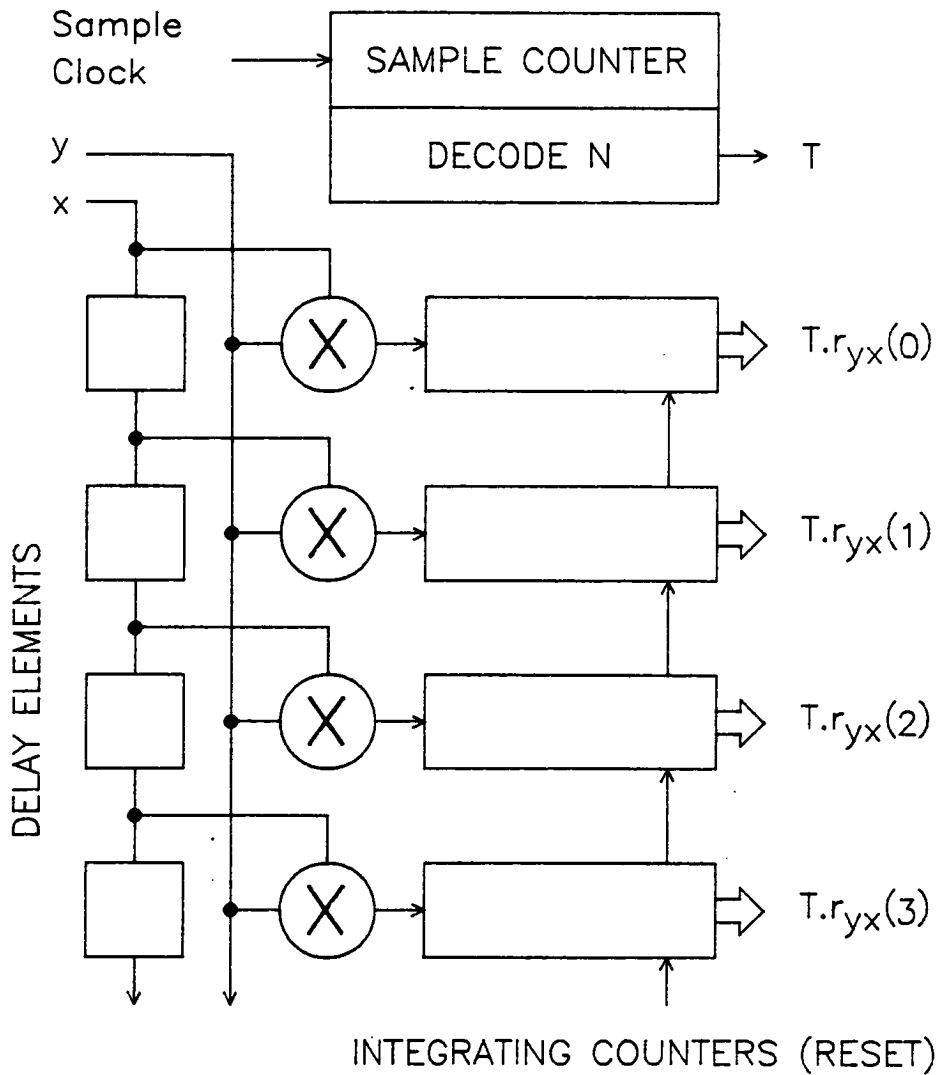


Figure 3.2. Parallel architecture with temporal integration and spatial delay.

It can be seen from Figure 3.2 that the correlation function is computed by an array of multipliers, integrators, and delay elements. Each point of the function is computed simultaneously by a dedicated multiplier and integrator. The delay operation is implemented by a tapped shift register. At each cycle of the computation all the delayed values of the input signal are available - hence the term "spatial delay architecture".

The architecture considered here has a major disadvantage in that a parallel output is required by each integrator. In the case of a digital circuit this output is around 8-16 bits wide for each point of the function. Direct communication with every integrator would involve a large number of output pins unless some form of multiplexing were used. An example of such an implementation (although not an integrated circuit) is reported by Corti et al [36], where each integrated result is shifted serially across the array of counter/registers to the output. This technique defeats the purpose of the spatial time delay by reintroducing a temporal operation at the output. The method is only advantageous when the integration period greatly exceeds the maximum time lag, or the integration period is too long to be implemented by a spatial integrator (spatial integration is discussed in section 3.2.2.3). The correlator designed by Corti was designed to correlate weak optical signals over a range of 108 delays with an integration period of approximately 65,000 sample periods. Currently, the maximum integration period using spatial integration is 512 sample periods, which is only possible using analogue current summing techniques [123,63]. Integration times in digital integrated correlators are much shorter. Chips which comprise 128 integrating stages are state of the art [90].

The architecture of Figure 3.2 may be optimised for VLSI implementation by incorporating the overloading integrating counter technique, which is discussed in the next section.

Another time-integrating correlator, but with a different architecture to the one described above, is reported by Burke et al [124]. The technique, which is illustrated in Figure 3.3, is peculiar to analogue correlators employing charge coupled devices (CCDs), and will

be discussed again in section 3.2.2.4. The correlation delay is achieved spatially, but in this case $\frac{1}{2}N(N+1)$ shift register cells are required, compared with N shift register cells in the previous architecture.

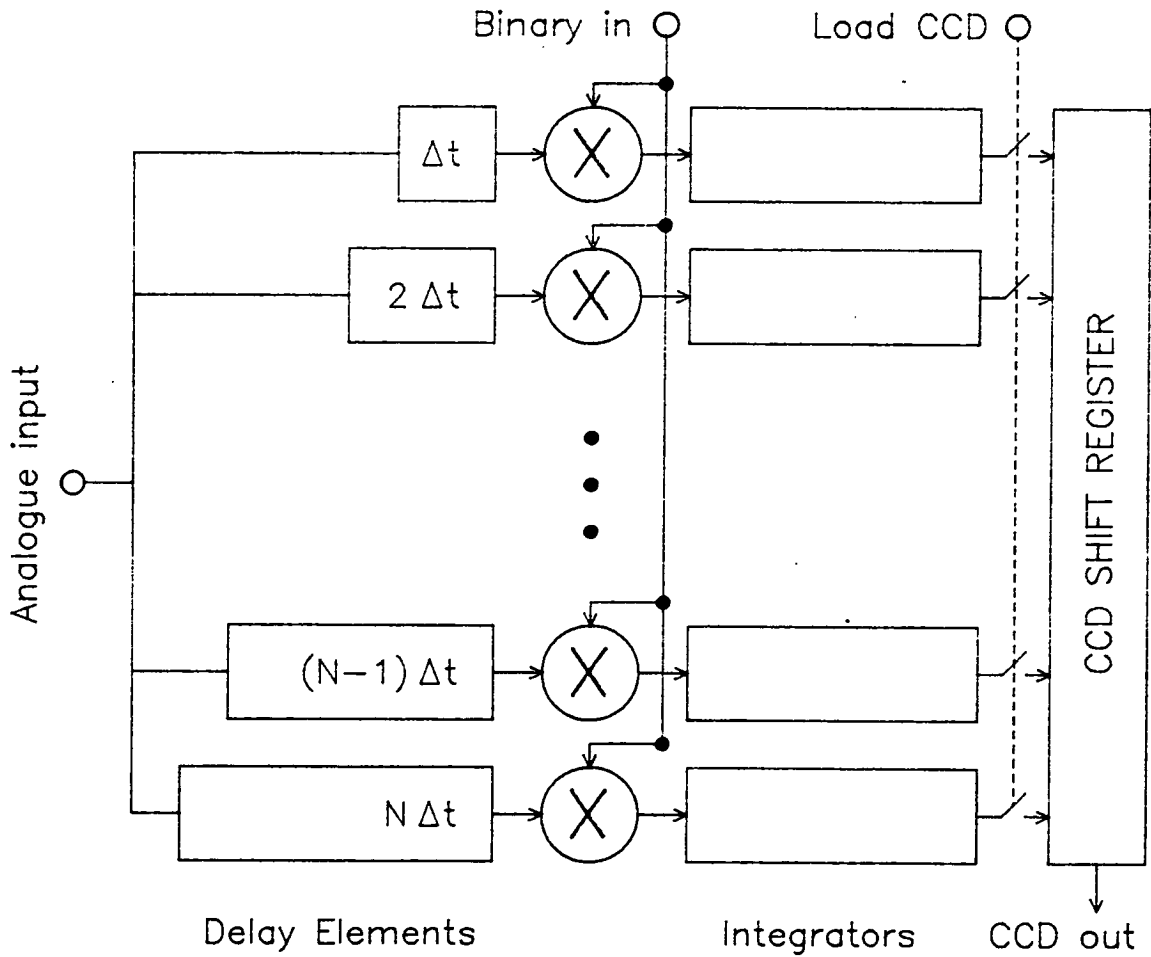


Figure 3.3. CCD time integrating correlator.

In the case of CCDs, there are advantages in using the larger array of CCD cells. The principal advantage this array has over its equivalent using only N cells is that the requirement for non destructive sensing of the CCD outputs is eliminated. This greatly simplifies the CCD design and clocking scheme.

3.2.2.2. Parallel architecture with temporal integration and spatial delay using the overloading counter technique

The architecture of the Eu349 chip falls into this category. It is shown in Figure 3.4. The arguments presented in the previous section apply also to this architecture.

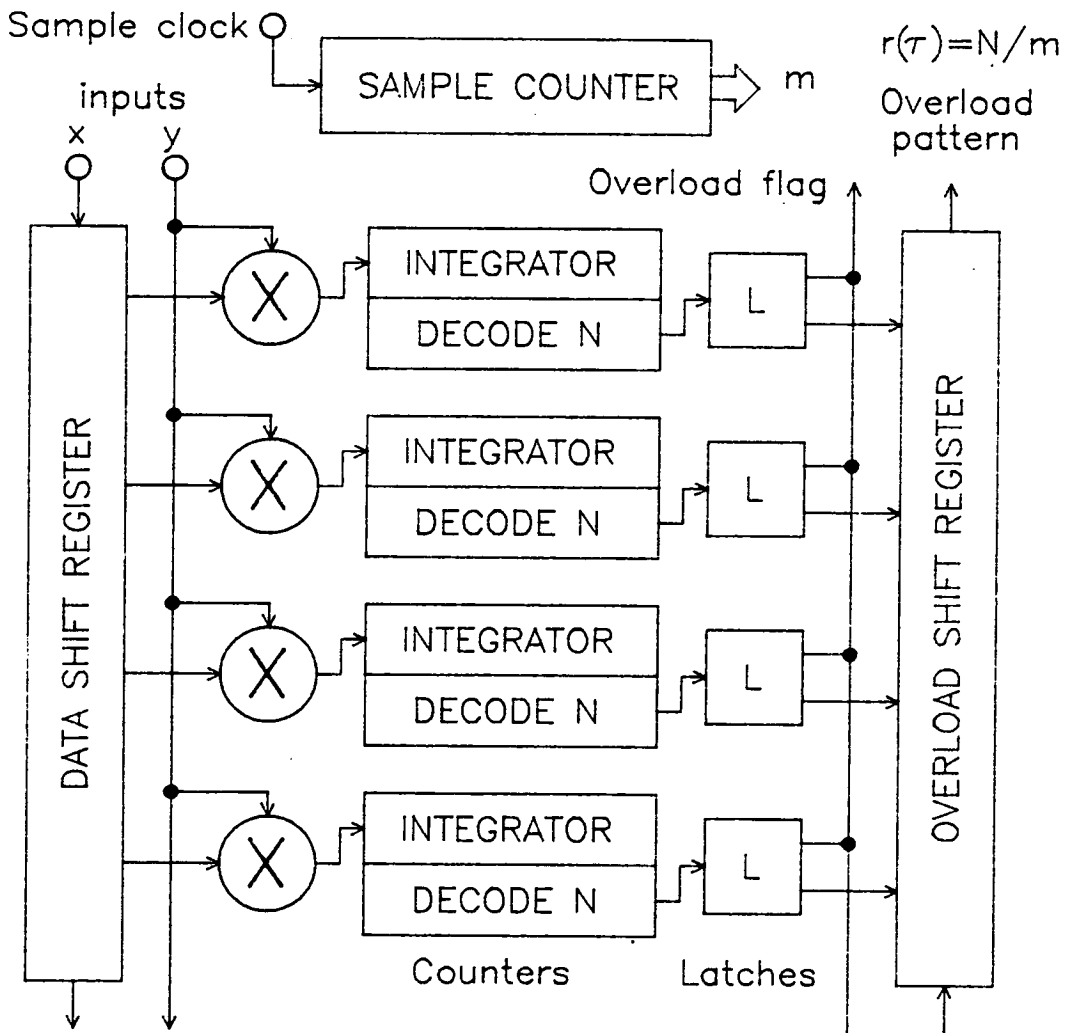


Figure 3.4. Architecture of overloading type correlator. A full description of the overloading counter technique is given in Chapter 2, but briefly the operation of the circuit is as follows: A small modification to the

integrating counters leads to a system which is much more suitable for realisation as a large scale integrated circuit. Instead of monitoring the total contents of each counter at the end of a predetermined integration period, the counters are arranged to indicate when a preset value is reached. Thus when a counter overloads (i.e. exceeds its preset capacity), an overload bit is stored in an associated latch. Clearly the first counter to overload indicates the position of the most significant peak of the polarity correlation function. If integration is allowed to proceed after the peak has been detected then progressively more counters will overload and the pattern of overload states will grow as shown in Figure 3.5. The envelope of the overload pattern describes the shape of the function.

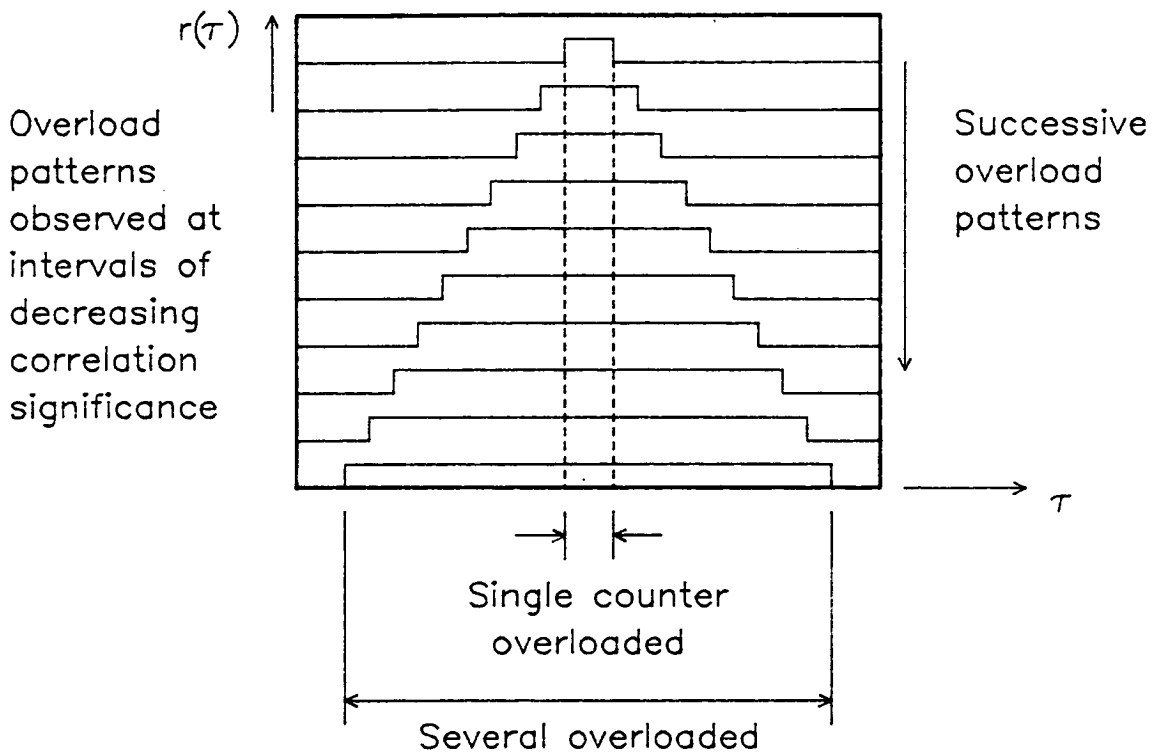


Figure 3.5. Overload patterns from overloading integrating counters.

This architecture can be described as a linear systolic array. Advantages include cascadability, without the need for external components; long programmable integration time; and nearest neighbour communications. There is no fan-in. Two versions of this architecture have been realised by silicon integrated circuits, one using pMOS technology [28,98], and the other, the Eu349 described by this thesis, in nMOS technology [2,100]. The Eu349 has, however, some novel design features which allow it to perform automatic self test and self repair [101,103,102]. This aspect of the design is discussed in Chapter 5.

3.2.2.3. Parallel architecture with spatial integration and temporal delay

Figure 3.6 shows the elements of spatial integration correlator. The operation of such an architecture is as follows: both signals are stored in registers, with taps at each stage connected to a parallel array of multipliers. The products are summed over the array to give the integrated result for a single value of correlation delay. Subsequent values of the correlation function at different delays are then computed by shifting one of the signals with respect to the other, and repeating the integration process.

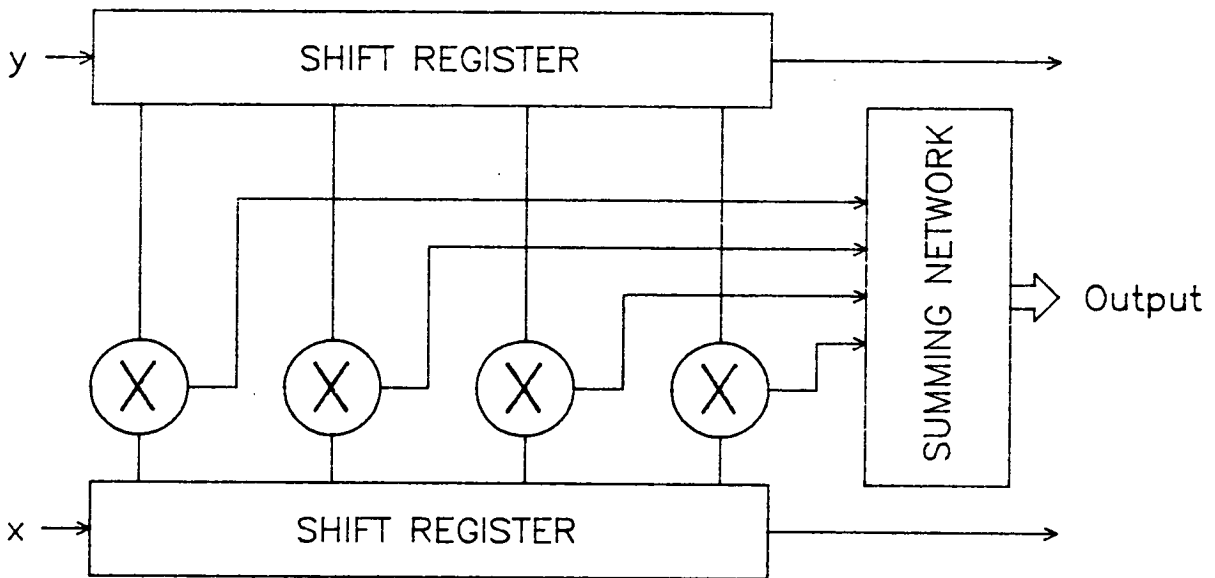


Figure 3.6. Parallel correlator using spatial integration.

The architecture is especially suited to analogue correlators due to the ease with which the summing network can be implemented using analogue techniques. A purely analogue correlator has been reported which consists of 64 cascaded stages to give integration over 64 samples [125,126]. MOS storage capacitors are employed for the "static" channel,

charge coupled devices are used for the "active" channel (i.e. the one to be shifted), and single MOS transistors perform the analogue multiplication. Currents from all the multiplier transistors are summed on a common source busbar and summing amplifier.

The majority of the analogue correlators in this category implement relay (analogue-binary) correlation. Again, CCD shift registers are used in the active channel and digital shift registers are employed in the static channel. Current summing is a method of integrating the sample products which requires less silicon area than digital methods. Relay correlators with 64 stages [127], 128 stages [128,129], and even 512 stages [63,123] have been reported. An example of an analogue/digital correlator is reported [130]. Analogue information is sampled and held at fixed sites on the chip and digital information is shifted past them. The digital channel, which is quantised into 7 bits, controls the selection of 7 binary area-ratioed MOS capacitors per correlator stage. The area penalty for employing a 7 bit digital channel is that the chip contains only 16 stages of correlation.

However, analogue techniques have serious disadvantages, not least with the CCDs. Complicated clocking schemes, clock breakthrough, bias, and leakage are some of the problem areas. Digital correlators are therefore desirable but generally require more silicon area to implement, unless the accuracy of polarity correlation is sufficient for the application. One digital polarity correlator architecture retains the analogue output and current summing technique in an attempt to enjoy the best of both worlds [131]. The chip consists of 64 stages of correlation each with its own current generator which feeds the current summing bus, as shown in Figure 3.7.



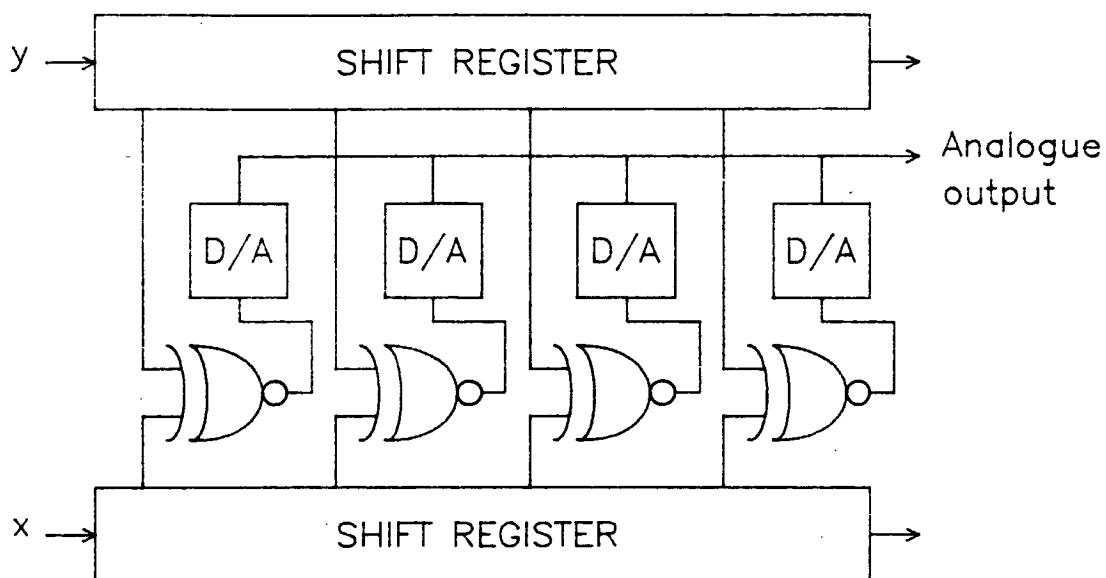


Figure 3.7. Digital correlator using current summing integration technique.

A system for spread spectrum communications based on this chip, is described by Saethermoen et al [132]. To allow for over sampling (twice the Nyquist rate), in-phase and quadrature correlation, and 4-bit quantisation, the system required a total of 16 stages of correlation per integration sample. The correlator described has an integration time of 1024 samples, which was achieved by cascading 256 correlator chips.

The architecture of a correlator employing all digital techniques consists of digital shift registers, digital multipliers and a digital summing network. A recent digital correlator chip [90,89] consists of four, 32-bit polarity correlator modules. Their individual outputs can be combined in a variety of ways to implement 1x4 bit quantised inputs, 2x2 bit quantisation, 2x1, or 1x1 bit inputs, all with a corresponding compromise in integration time. There is also a facility for quadrature signal correlation. In the case of a polarity correlator the

multipliers are EXNOR gates and the summing network is a parallel counter. A parallel counter is a combinational circuit that determines how many of its inputs are at a given logical state (usually logic 1) expressing the result as a parallel binary number of its outputs. Parallel counters have been extensively researched [133,94,95,134]. They are difficult to design, in that they lack modularity in an arithmetic sense. For example a large parallel counter can only be made from two smaller parallel counters by using extra components to combine the separate outputs. As a result, a large parallel counter is best designed recursively, starting from the minimum implementation (3-input full adder) and working up, geometrically. Such an approach is the basis of a silicon compiler for parallel counters reported by Cappello [135]. The architecture of a 31-bit parallel counter is shown in Figure 3.8. Parallel counters occupy a significant portion of the silicon area in digital correlator designs. Also, pipelining is normally employed to reduce the throughput rate, which increases the required area still further. Multi-valued logic techniques have been used to reduce the silicon area required by integrated parallel counters, as shown in Figure 3.9. Area savings of nearly 50% using quaternary logic have been reported [96,97]. Multi-valued logic circuits are most easily realised in technologies such as ECL and IIL [136]. This means that very high clocking rates (50 MHz) are possible in digital correlators employing this technique [92,93].

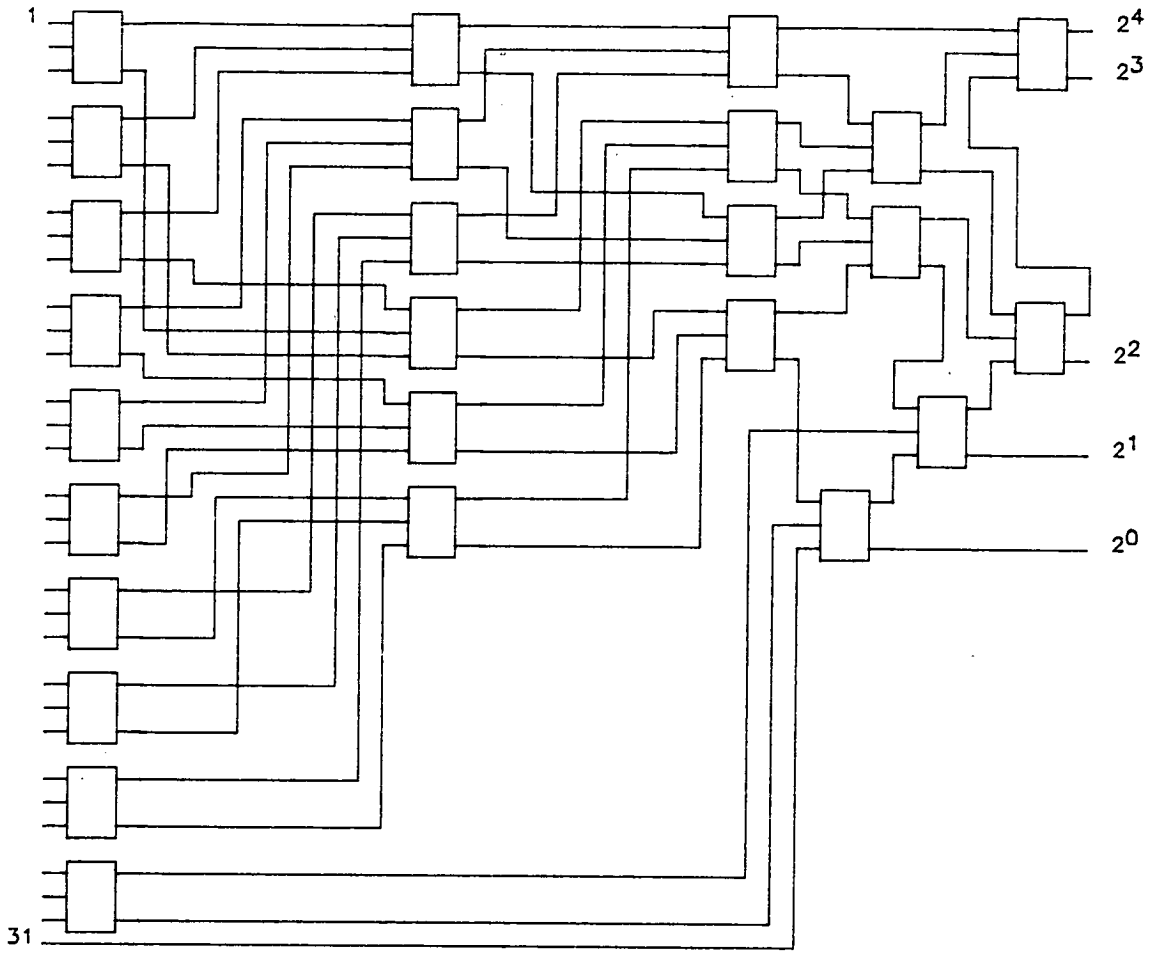


Figure 3.8. 31-bit binary parallel counter using binary full adders.

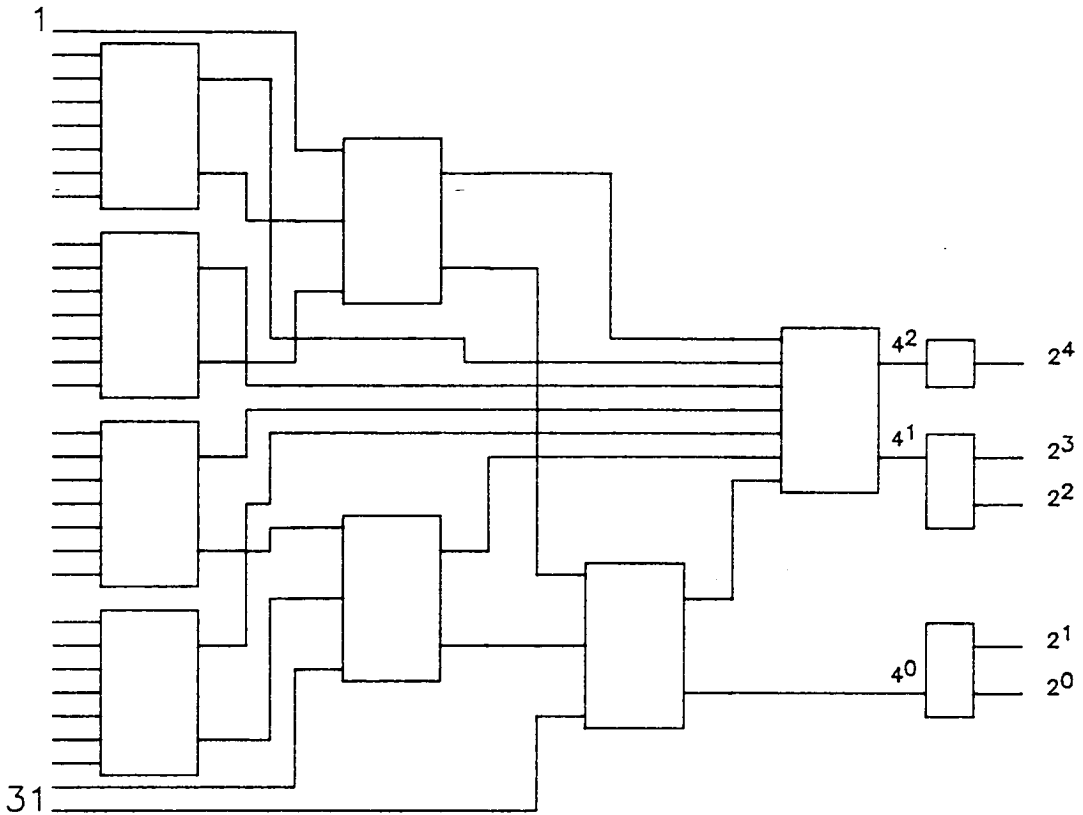


Figure 3.9. Quaternary logic parallel counter using full adder circuits.

Note the lack of modularity in the architecture of parallel counter, and the high degree of global interconnection and fan-in that is necessary. A method which solves these particular problems is the architecture shown in Figure 3.10. Here the summation is distributed along the correlator array. The operation of the circuit will be slower than one with a pipelined parallel counter unless pipelining is incorporated here also, and the circuit is operated in a systolic fashion. However, other problems are then introduced, since the summing network must be pipelined *along with* the other elements of the correlator.

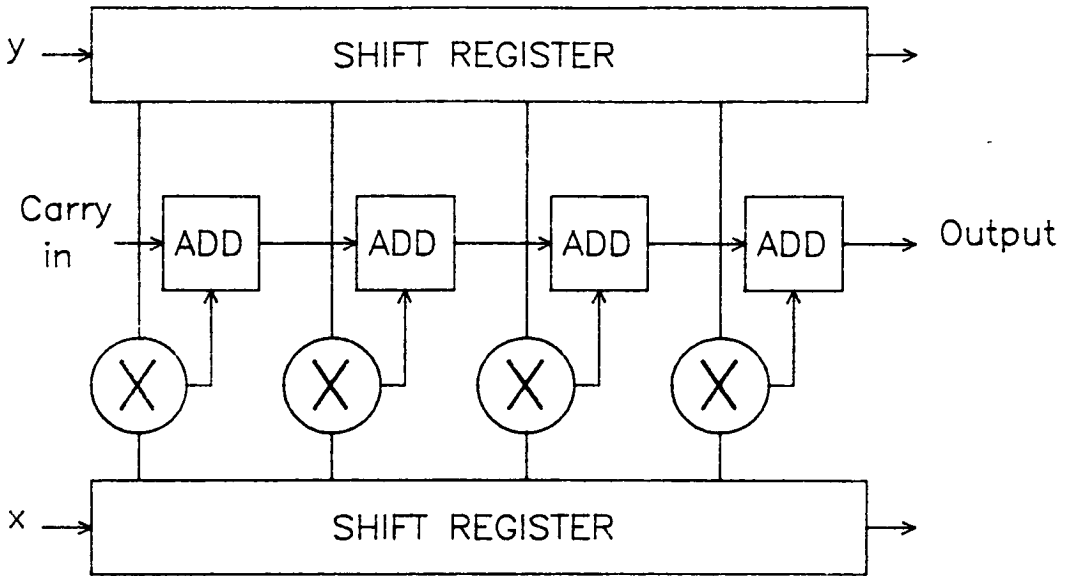


Figure 3.10. Spatial integration using distributed adders.

This is the basis of the systolic correlators discussed in section 3.2.4.

3.2.2.4. Parallel architecture with spatial integration and temporal delay using pipe-organ structures

A special architecture, shown in Figure 3.11 and termed a pipe-organ correlator, is equivalent to the conventional spatial integrating correlator of Figure 3.6. It arises from the fact that those CCDs, which do not require non-destructive sensing techniques, are much simpler to construct than their destructive sensing counterparts. Every delay element in the conventional architecture transfers its stored information to two inputs, namely the next delay element, and a multiplier. It is essential, therefore, that the information remains intact during the process. To avoid this situation, the same algorithm can be implemented using separate delay times for each correlation point. Every delay cell now feeds

only one input. The stored information may now be destroyed during a transfer operation. In CCD technology, the simplification in circuit design (and control) that the technique of destructive sensing permits is often worth the extra area involved [137,138]. Miller and Berry have described a pipe-organ correlator, where the extra area required is reduced by merging CCD cells in groups of four [65].

A dual of this architecture, which uses temporal integration, has been described in section 3.2.2.1.

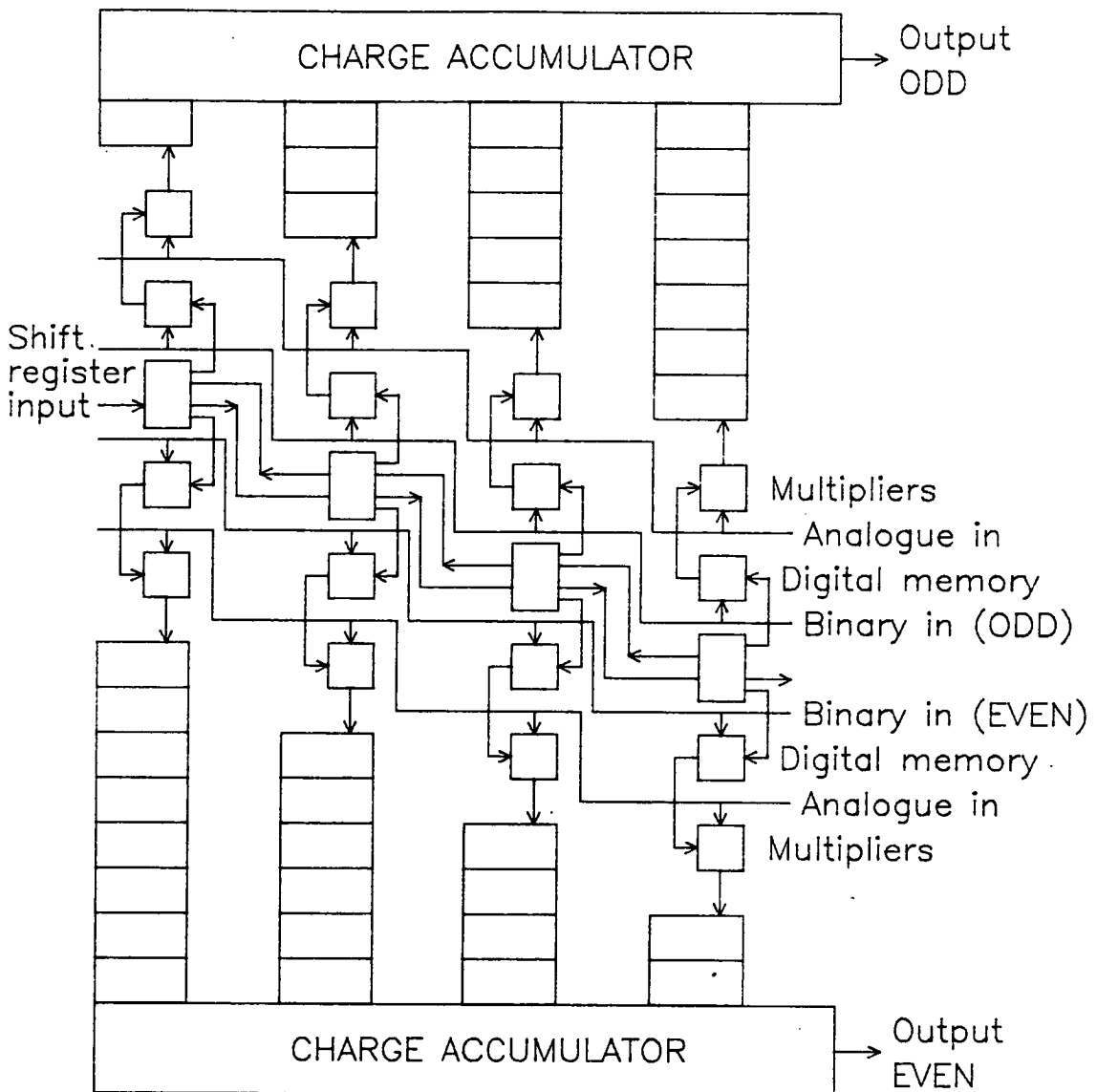


Figure 3.11. Spatially integrated pipe-organ correlator.

3.2.3. Serial Parallel Architectures (DELTIC)

The architectures described in the previous sections have comprised of functional elements, all of which operate at a common clock rate. This section deals with serial/parallel architectures, termed delay-line time compressor (DELTIC) correlators [6,139], where internal

circuitry operates at a higher rate than the sample rate. The configuration of a DELTIC correlator is shown in Figure 3.12.

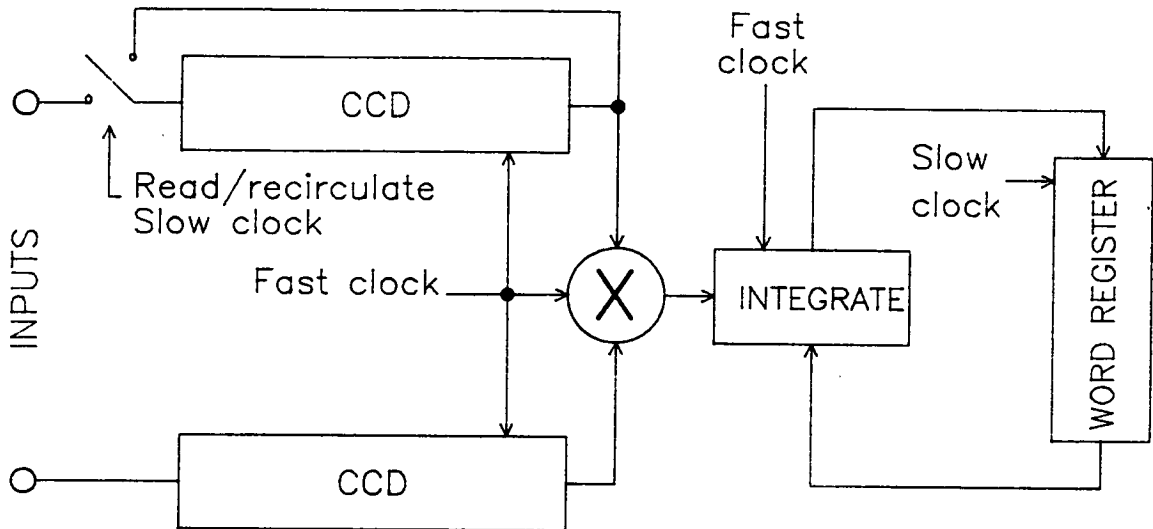


Figure 3.12. DELTIC correlator configuration.

Here the data is time compressed, or expanded in bandwidth, by a factor N , to permit a single, fast multiplier to perform the required $N \times N$ multiplications in a time equal to N input sample periods. Thus the data contained in the recirculating store must be recirculated at a rate of N times the input sample rate. For a fixed data record, the memory information is held for N complete recirculations before being replaced by a new record. In the case of a varying input signal, the oldest memory sample is replaced by a new input sample after each recirculation. Multiplying each sample of the recirculating data by a reference signal and integrating N samples provides one point of the correlation function. Further points are obtained on successive recirculations. The disadvantage of this architecture is that the correlation rate is limited by the speed of the single multiplication element. The concurrent use of an array of multipliers, as

described in section 3.2.2, increases the correlation rate significantly. It also renders a more modular design which is more suitable for VLSI implementation.

3.2.4. Systolic Architectures

Systolic architectures have been reviewed by several authors [122,121,140,141]. In this section, only those systolic architectures for integrated circuit correlation will be discussed. The algorithms that underlie these architectures can again be divided into two categories: spatially integrating correlators, and time integrating correlators.

3.2.4.1. Temporal Integration

The Eu349 correlator chip, shown in Figure 3.4 is a time integrating linear systolic array, which uses global control signals but no fan-in. There is only a single delay on both input and output shift registers, and the elements of the delay may be cascaded directly. This architecture can also be adapted to provide fault tolerant features by simply adding two multipliers and one latch per correlator stage. The circuit design of the Eu349 chip is discussed in detail in chapter 4. A time integrating correlator chip, similar in concept to the Eu349, has been reported by Barral and Moreau [142], and is illustrated in Figure 3.13.

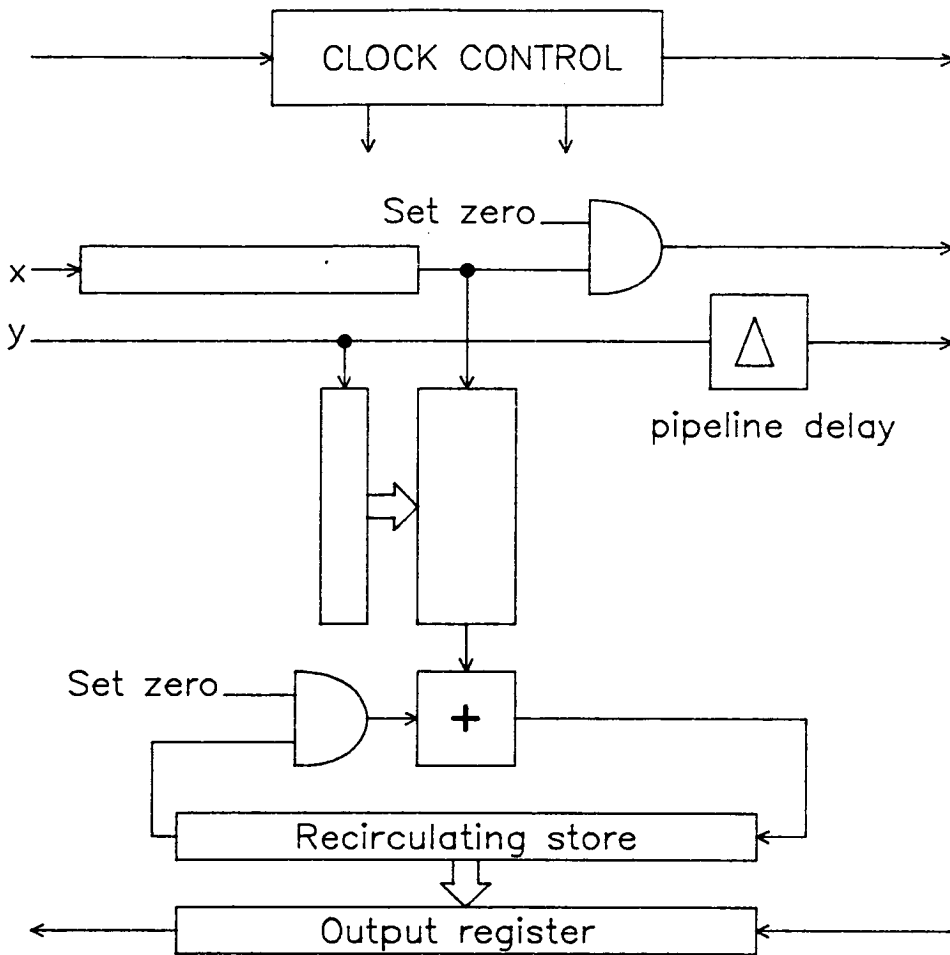


Figure 3.13. Bit-serial systolic correlator (single stage).

The correlator stage shown in Figure 3.13 can compute a single correlation point integrated over a maximum of 512 samples. The samples are 12-bit two's complement numbers and the chip contains 11 identical stages. The architecture is bit serial. From the view point of this discussion, however, the main difference between this architecture and that of the Eu349 is the extra pipeline delay between each stage. The control signals propagate through the pipeline from one stage to the next; thus global control signals are avoided. Disadvantages of this architecture include low correlation rate due to the bit-serial

implementation (300 kHz maximum); integration time limited to 512 samples; and only 11 parallel stages of correlation per chip.

The remaining examples here, of systolic correlation chips, employ spatial integration. The conceptional difference between time integration and spatial integration is treated in Section 3.3.

3.2.4.2. Spatial Integration

In devising systolic architectures all the possible permutations of the three quantities (reference, input, and results) and the two parameters (moving or stationary) are explored. For example, the architecture shown in Figure 3.6 can be described as having "stationary reference signal, moving input signal and stationary output". At each shift cycle the stationary outputs fan-in to the single summing network. A similar situation is shown in Figure 3.10, the only difference being the summing network, which is now distributed over the array.

Another permutation is shown in Figure 3.14. In this example, which is the architecture of a correlator chip described by Snelling and Penn [143], the summing network, and input signal channel, are pipelined. If the summing network alone were pipelined, an architecture is produced which has a stationary reference signal, moving input signal, and a moving output signal, which will not compute a correlation function unless the adjacent bits of the input signal (and hence the output signal) are separated by zeros. One alternative solution, adopted by Snelling and Penn, is to introduce a pipeline delay at each stage in the input signal, as well as in the summing network. The correlator described by Snelling and Penn is also pipelined into bit slices. The complete architecture allows

1-bit x 8-bit correlation, but integration is only over 8 samples. Another alternative solution, which is described by Kawahara [144], is to remove the delay entirely from the input signal. This architecture is shown in Figure 3.15. The chip described by Kawahara computes a 3-bit x 4-bit correlation function integrated over 32 samples. The output word size is 11 bits.

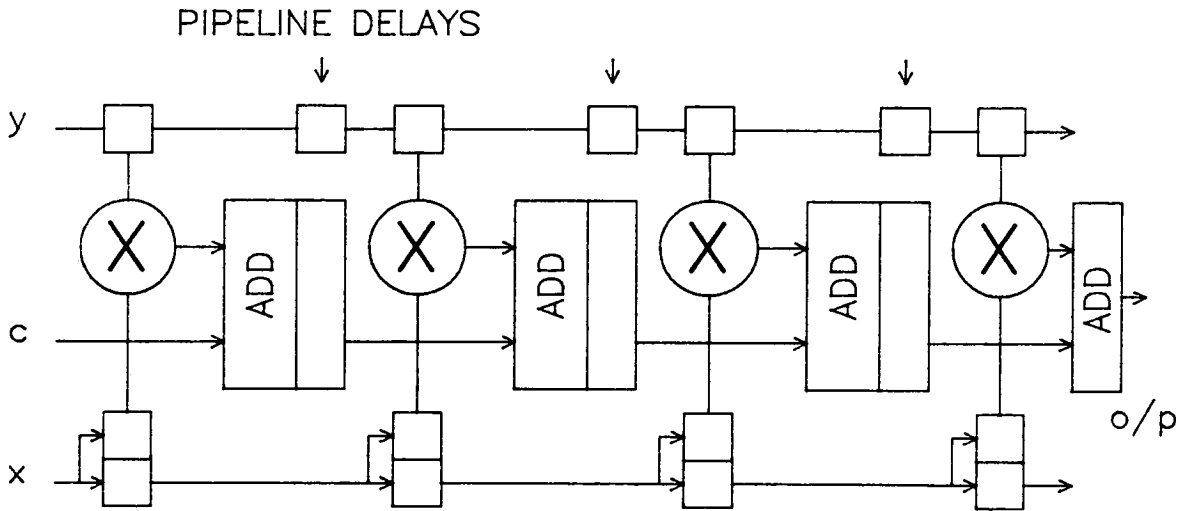


Figure 3.14. Systolic correlator with pipelined summing network and input register.

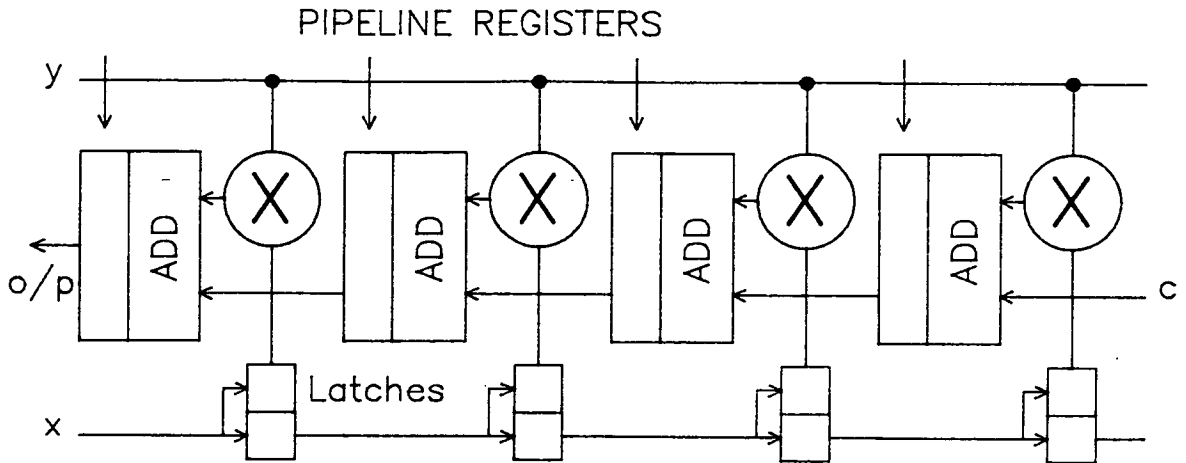


Figure 3.15. Correlator architecture with pipelined summing network and global input signal.

Finally, a two dimensional systolic array of simple 1-bit processor and memory cells, which can compute correlation functions, is described by McWhirter et al [145,146]. The silicon implementation of the architecture [147] provides 4-bit x 1-bit correlation, employing spatial integrations over 64 data samples. The correlation algorithm uses a moving reference signal, moving input signal and moving results. Zeros are interspersed between adjacent bits of the input data words and the reference words to achieve the desired interaction of the components. The architecture of the correlator is shown in Figure 3.16.

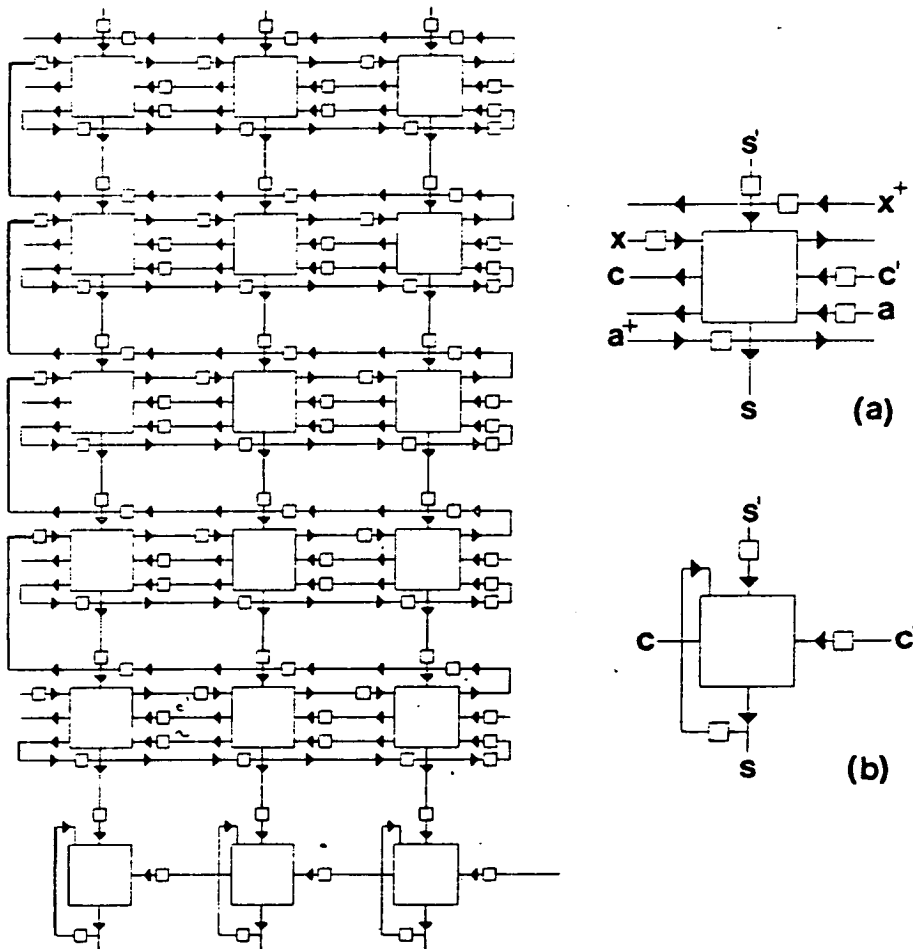


Figure 3.16. Two-dimensional systolic correlator of McWhirter et al.

As a result of the interspersed zeros and the continual contra-flow of the data and reference bits, a diamond shaped region of valid interaction propagates down the array, as shown in Figure 3.17.

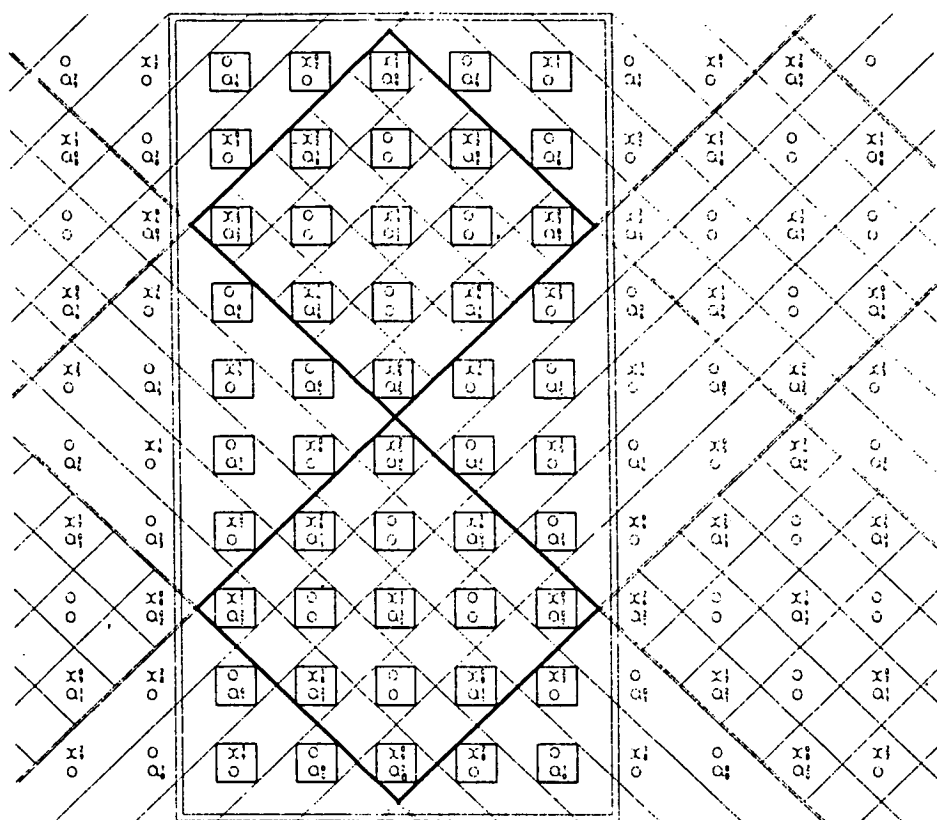


Figure 3.17. Data flow in the systolic correlator of McWhirter et al.

The partial products inside the interaction area eventually reach the bottom edge of the array where they are accumulated by the adder cells (marked (b) in Figure 3.16). Only those partial products which are relevant to the particular correlation point being accumulated, will have any effect, since all others will have a zero in one, or both of the multiplicands. The correlator operates in a bit serial manner, and produces a valid result every $4N-1$ clock cycles (for two's complement numbers), where N is the length (integration time) of the array. A CMOS realisation of this architecture, where $N=64$, operating at 20 MHz could provide 16-bit results at a rate of just under 100 kHz. An important disadvantage, therefore, of

this type of systolic array is one of throughput, particularly if the array is large. Another disadvantage is that arrays must be cascaded geometrically to allow for internal word growth in the partial products. In practice, truncation is used to limit the permitted word growth.

3.3. Correlation Cube: The Difference Between Temporal and Spatial Integration

In this section the contrasting architectures of time integrating correlators and spatially integrating correlators are discussed. There are two points to note in particular: time delay implementation and integration technique. Figures 3.18 and 3.19 show respectively spatial integration and time integration architectures.

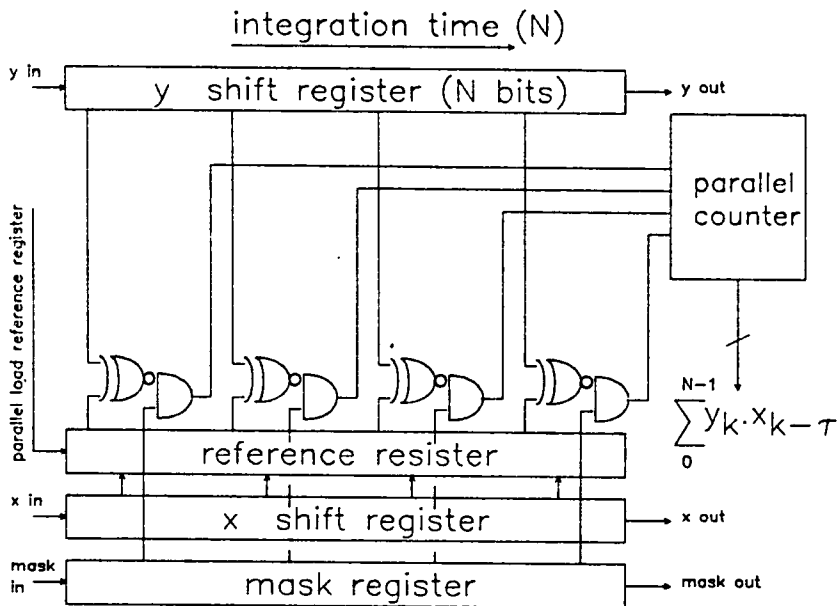


Figure 3.18. Correlator architecture using a single parallel integrator.

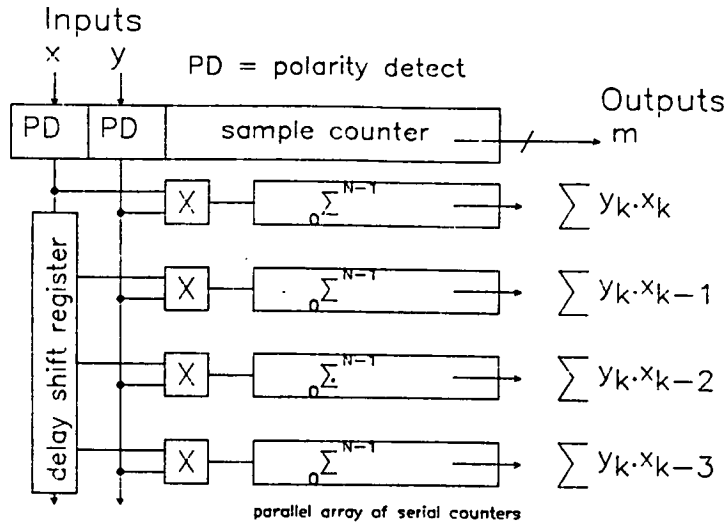


Figure 3.19. Eu349 architecture using a parallel array of serial integrators.

In Figure 3.18, the relative delay between the signals is achieved by dumping the x register contents into the reference register. In this way the x signal is held stationary while the y signal is shifted past. The time delay window is given by the period between the x register parallel dumps. In contrast the Eu349 architecture delays only one input signal. Hence each stage of the correlator introduces a unit of time delay between the input signals. For the Eu349 the time delay window is given by the number of correlator stages, and may be increased easily by cascading the Eu349 chips.

Integration time in the Eu349 is governed by the capacity of the integrating counters, which is programmable. Thus the integration time may be varied from 1 to 32,766, regardless of the number of correlation stages in the cascade.

In the architecture of Figure 3.18, the integration time is determined by the length of the correlator, that is, the number of bits in the shift registers.

Integration time is therefore short. The integration time may be increased by cascading the chips, but this is difficult because the individual chip outputs (typically 7 bits for an integration time of 64) must be added together using external circuitry [94,95].

The differences in the architectures of Figures 3.18 and 3.19, may be summarised by viewing correlation in three dimensions: function amplitude, time delay, and integration time. A correlation cube to demonstrate this is shown in Figure 3.20. Both architectures incorporate two physical dimensions and one time dimension. It can be seen in Figure 3.20 how the two physical dimensions of the architectures occupy orthogonal slices of the correlation cube.

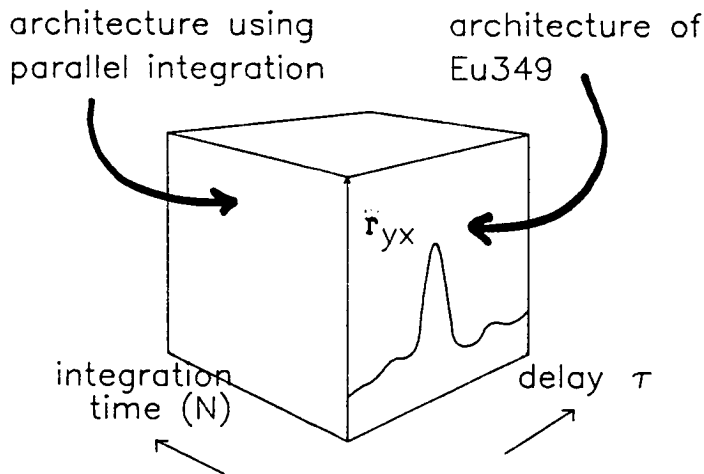


Figure 3.20. The correlation cube showing the relationship between the two contrasting architectures.

3.3.1. Correlator Architecture based on Spatial Integration

For this architecture, the correlation equation

$$r(\tau) = \sum_{k=0}^{N-1} y_k x_{k-\tau} \quad 3.1$$

is implemented by *storing* the reference signal y_k in a (maskable) register latch of N stages, where N represents the correlator integration time. The input signal x_k is shifted along a tapped shift register and at the k th tap, on each clock cycle, the product $y_k \cdot x_{k-\tau}$ is produced. Summing these individual products in a single (parallel) counter produces the desired correlation function. The output is one b -bit value of the function for each clock cycle. Effectively, the parallel counter is integrating the products for all k , from 0 to $N-1$, simultaneously, at a single delay value τ , per clock cycle.

Increasing code lengths by cascading individual correlator chips is complicated by the need to add together the b -bit words from individual parallel counters. In this parallel counter structure the correlator integration time is less than (if masking is used), or equal to the reference code length. The correlator, therefore, has three degrees of freedom: integration time, correlation lag or delay, and correlation amplitude. These may be represented on the correlation cube as shown in Figure 3.21.

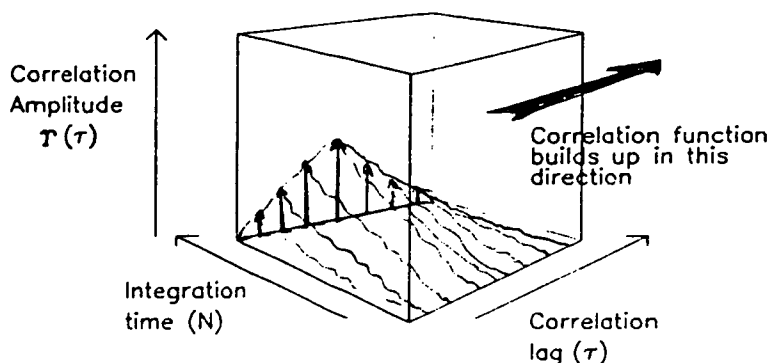


Figure 3.21. Correlation cube for the spatially integrating correlator.

The correlation function of Figure 3.22, which has fixed integration time N , and produces one value of $r(\tau)$ for each lag τ , would be depicted as shown in Figure 3.21, on the rear face of the cube, since the integration over N samples is effectively performed instantaneously within the parallel counter.

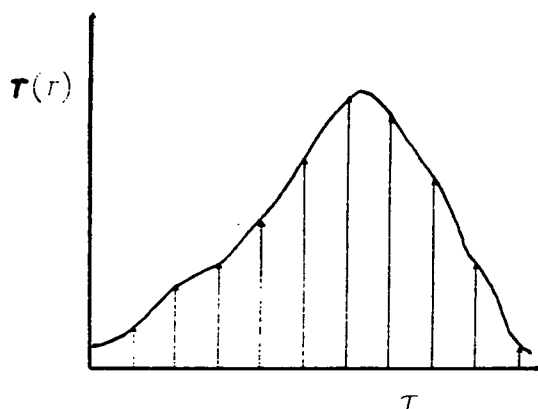


Figure 3.22. The correlation function from the spatially integrating architecture.

3.3.2. Correlation Architecture based on Temporal Integration

In the Eu349, the correlation equation

$$r(\tau) = \sum_{k=0}^{N-1} y_k x_{k-\tau} \quad 3.2$$

is implemented. The input signal x_k is shifted through a tapped shift register and at each tap the product $y_k \cdot x_{k-\tau}$ is produced. Note that, in contrast to the previous architecture, the Eu349 does not involve on-chip latching of the reference waveform y_k , and that each individual sample of y_k is applied simultaneously to all stages. A serial counter/integrator on each stage integrates the product values for all values of k , up to a maximum of $N-1$. The value N , which represents the integration time, is simply the preset capacity of the serial counters, and has nothing whatsoever to do with the number of stages in the Eu349 correlator. Each integrating counter in the Eu349, is dedicated to integrating the product values for one particular value of lag or delay. The value of lag is determined by the position of the counter in the overall array. Thus the contents of the τ th counter, after an integration time of N , will be

$$\sum_{k=0}^{N-1} y_k x_{k-\tau} = r(\tau), \quad 3.3$$

which is exactly the function produced by the spatially integrating correlator.

The main difference between the two architectures can be visualised with reference to the correlation cube.

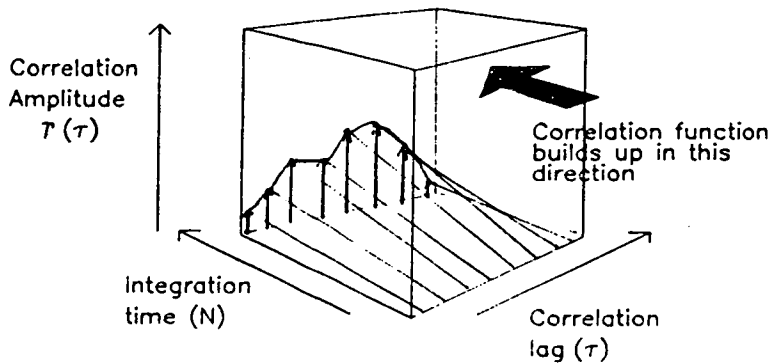


Figure 3.23. Correlation cube for the correlator Eu349.

Whereas in the spatially integrating correlator, the integration over N samples is performed "instantaneously" in the parallel counter to produce one value of the correlation function per clock cycle, in the Eu349, the *array* of serial counters simultaneously offer the values of the correlation function at *all* time lags as a function of integration time. Thus, after an integration time of N clock cycles, the values in the serial counters will represent the final correlation function, identical to the result from the spatially integrating device.

In the Eu349, the values $r(\tau)$ can be read out from the array of counters to yield the correlation function. In contrast to the spatially integrating device, the Eu349 offers direct cascading of individual chips without requiring the use of additional circuitry, to increase the maximum length of reference code and lag value, whilst offering an *independent* variation of integration time N by the presettable serial counters. Also, the correlation rate is not affected by the size of the array.

3.3.3. Display of Correlation Output

With the spatially integrating correlator, the b -bit output of the correlation function is obtained each clock cycle. With the Eu349, a latency of N clock cycles (N is

the integration time through the correlation cube of Figure 3.23) is required before the correlation function can be read out. As an alternative display mechanism, the overloading counter technique can be used to provide a bit-serial output of the correlation function for applications where the integration time is significantly greater than the maximum lag value, or reference code length. Here, when one of the presettable serial counters overloads, a flag is set, the overload status (one bit for each counter) is read out for all counters from a serial shift register, and the time lag of the correlation function peak (see Figure 3.23) can be determined. On later clock cycles (representing lesser correlation significance), several other counters will have overloaded, so that points around the main correlation peak, and other lesser peaks of the function may be displayed (see Figure 3.24).

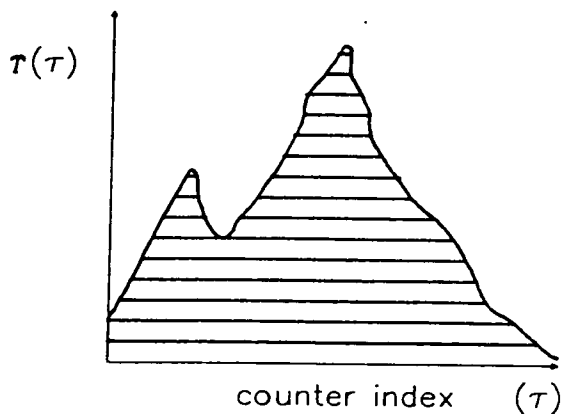


Figure 3.24. Display technique for the Eu349 correlator.

3.4. Summary

In this chapter, several implementations of silicon correlators have been discussed. The architectures may be classified by observing whether time integrating or spatially integrating techniques have been used. The

difference between these two concepts has been illustrated by the correlation cube. Further segregation of correlator architectures may be made by observing which computational techniques have been used, namely bit serial, bit parallel, polarity, systolic etc.

Parallel and concurrent techniques are employed to an ever increasing extent in integrated circuit correlators. However there exists a compromise between using a large number of very simple concurrent operations, and using a small number of complex cells, to achieve a common objective. In the DELTIC correlator, a single, fast, multiplier is used. In the systolic correlator of McWhirter et al. delay, multiply, and add operations are distributed over a large 2-dimensional array of simple cells. However, partial products are only generated in cells within an interaction region and these in turn are only used to form a product on every alternate clock cycle. Furthermore, to achieve useful integration times a large array of cells is required, and to increase the integration time requires cells to be cascaded. Normally this would not be a disadvantage; it is in fact preferable for VLSI architectures to be modular and cascadable. However the output rate of this correlator is inversely proportional to the size of the array.

The architecture of the Eu349 correlator achieves a balance between concurrency, cascadability and correlation rate. The architecture is concurrent in that each point of the correlation function is computed in parallel. The architecture is directly cascadable, and the correlation rate is independent of the length of the array.

CHAPTER 4

VLSI DESIGN STRATEGIES FOR TESTABILITY AND FAULT TOLERANCE

4.1. Introduction

The concepts of design for testability and fault tolerance in integrated circuit design become important as feature sizes shrink and chip sizes increase. The chip described in this thesis embodies a design for testability strategy and provides yield enhancement and fault tolerance through the use of redundancy. These two topics are discussed in this chapter.

Design for testability addresses the two major facets of the chip testing problem: test pattern generation, and test response verification. At the circuit complexities presented by VLSI the need to design testable logic circuits is crucial, and considerable work has been done in recent years in devising design strategies that produce highly testable circuits [148, 149, 150, 151, 152]. Testability can be achieved by:

- a) *ad hoc* partitioning of a VLSI design into small testable modules or stages,
- b) the inclusion of a systematic testability scheme, such as scan path, and
- c) built-in test and self test strategies, and associated data compression techniques.

Fault tolerance is undoubtedly a desirable property in

any electronic system. In order to take full advantage of VLSI, the design strategy should include techniques for fault tolerance and yield improvement. Examples of these techniques are

- a) modified design rules, which reduce the probability of yield loss due to critical spacings, or random defects (defect avoidance),
- b) replication of critical circuits with associated majority voting schemes (concurrent fault tolerance), and
- c) modified VLSI architectures in which redundant circuit modules can be switched into operation to compensate for defective areas (nonconcurrent fault tolerance).

In this thesis, attention will be restricted to non-concurrent schemes, referred to in paragraph (c) above.

Increased design and implementation costs should be expected when redundancy is incorporated into a VLSI design. A figure of merit can be defined, however, which takes into account the improvement of yield and the increase in implementation cost. The yield enhancement scheme is worthwhile, when the figure of merit is greater than unity i.e. the cost of the redundant chips will be lower than the cost of the nonredundant chips. The figure of merit for redundantly designed chips is a maximum when approximately 10% of the circuit is redundant [153]. This implies that chips can be designed around an optimum amount of additional circuitry to improve yield.

The Eu349 chip described in this thesis, has been designed for testability and fault tolerance. Furthermore, the design strategy allows faulty stages to be

detected and eliminated automatically. The circuit design is presented in Chapter 5, but as a precursor, the subjects of VLSI design for testability, and design for fault tolerance and yield enhancement will be reviewed in this chapter.

4.2. Test Philosophies and The Motivation Behind Design for Testability

With the increase in complexity of logic that can be fabricated on a VLSI chip, there is a growing problem in validating the logical behaviour of the chip at manufacture. Traditional test techniques require the derivation of input test stimuli, and associated output responses. Exhaustive testing of circuits demands the consideration of all possible logic states in which a circuit can exist. This strategy rapidly becomes uneconomical in complex, or deeply sequential circuits, since the costs and times involved in test pattern generation grow exponentially with increasing circuit complexity [152]. Techniques to reduce the number of test stimuli are based on the use of fault models and a knowledge of the internal structure of the circuit. The most common fault model is the stuck-at model. More comprehensive models are possible [154] but they substantially increase the difficulty of test pattern generation and do not offer any significant compensating advantages [151].

The efficiency of a test set is measured by its fault coverage, which, in the case of a stuck-at fault model, refers to the percentage of possible stuck-at faults the test set will detect. Fault simulation is commonly used in logic circuit testing to evaluate whether a generated test set does indeed detect the faults it was intended to detect. It is also used to compute the fault coverage.

There are a number of difficulties with this approach. Firstly, a fault model is required. In VLSI circuits the classical assumption that only single stuck-at faults need be modelled is not sufficient [154]. More comprehensive models are possible, but they increase the task of test pattern generation. Secondly, test pattern generation is required. Automatic test pattern generation [155] is very costly and typically does not provide a sufficiently high fault coverage. For sequential circuits at VLSI complexity, automatic test generation is extremely difficult, and manual generation is time consuming and error prone [149].

One method which avoids the problem of producing a specific test pattern is random testing [156]. In this case a relatively large number of random patterns are applied to the circuit under test. If the response is found to match the expected circuit response, then it can be assumed, within a specified confidence limit, that the circuit is fault-free. Random testing has been found to be an extremely effective means for fault detection in combinational circuits, but its effectiveness in dealing with sequential circuits is not easily defined [151].

An alternative to gate level testing is functional testing. This approach has the advantage that tests can be generated without having a detailed knowledge of the gate structure of the chip. the problem with functional testing, however, is that the only way to be certain that the circuit is fault-free, is to perform an exhaustive functional test. Since exhaustive testing is only feasible for circuits which have few inputs and few sequential states, then functional testing, on its own, is not a practical approach to VLSI testing.

From the foregoing discussion, it can be concluded

that testing becomes increasingly difficult as designs approach VLSI complexity. Methods used to reduce the amount of test data reduce, in turn, the fault coverage, and in any case are difficult to automate for large circuits. The only solution to these problems is to reduce the complexity of VLSI circuits, at least with regard to testing. Hence the term "design for testability". Figure 4.1 shows a comparison between test costs with, and without, design for testability. The test costs without design for testability grow exponentially with increasing complexity, in contrast to the almost linear characteristics of test costs for circuits which incorporate a design for testability scheme [157].

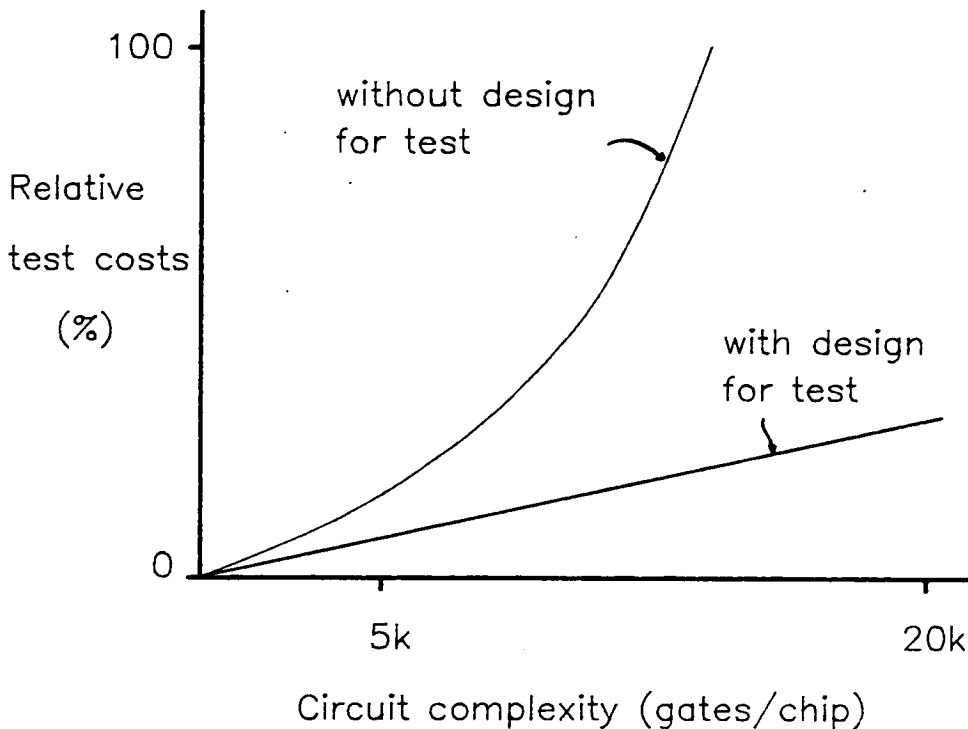


Figure 4.1. Comparison of test costs with, and without, design for testability.

This thesis addresses the need to embody a testability scheme within the VLSI integrated circuit itself, and describes a methodology which makes this possible for well

structured systems.

4.3. Design for Testability Methods

4.3.1. Objectives

Testability involves two important concepts: controllability and observability. Controllability is the ability to establish a circuit in a controlled initial state, and observability is the ability to observe externally, the internal states. Design for testability involves increasing the controllability and observability of the constituents of a design by decomposing the overall design into more manageable elements. The cost of design for testability can be measured by the number of additional package pins required for test purposes, the number of additional test circuits required, and any loss in performance resulting from design for testability techniques.

Increased circuit complexity reduces fabrication yield [153]. Thus, the increased chip costs involved in using extra silicon area for test purposes must be weighed against the savings in test costs, which are usually reflected by test time. Typically the use of test circuits which increase the chip area by approximately 10% is considered reasonable [158]. The variation of relative test costs with test circuit area overhead is shown in Figure 4.2 [157].

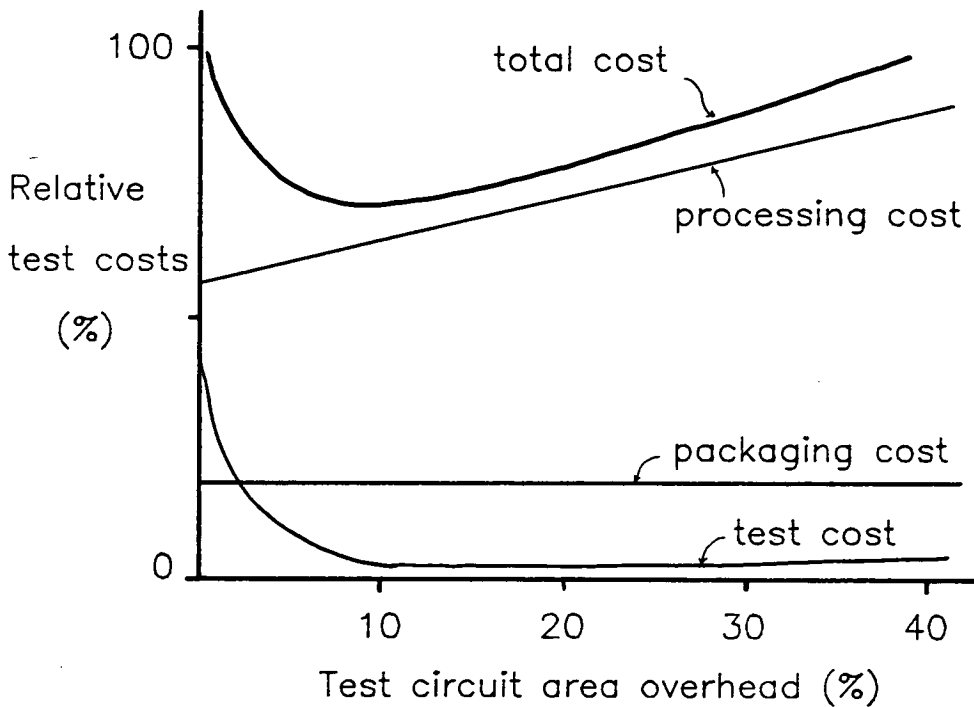


Figure 4.2. Typical variation of chip costs as a function of test circuit area overhead.

The significance of the test circuit area overhead depends on the type, and application, of the chip being designed. For low cost, high volume, modest performance designs, an acceptable test overhead is around 10%, whereas for high performance, low volume applications, test overheads of 100% may be acceptable.

4.3.2. Ad Hoc Methods

Ad hoc approaches to design for testability are in fact simply guidelines on how to improve the testability of a particular circuit. The testability problem has to be addressed again and again with every new design. The most common *ad hoc* method is circuit partitioning with added test points. This allows the circuit to be split into functional sub-modules, each of which may be accessed

and tested individually. The type of circuit architecture is important to the choice of *ad hoc* testability scheme. For example, bus structured circuits, such as microprocessors, are easily partitioned, using the busses as test points. However, with growing VLSI complexity, additional design for testability schemes must be employed within the sub-modules.

4.3.3. Scan Methods

The scan path method of design for testability enhances the controllability and observability of a VLSI circuit by allowing access to the internal states of a circuit [159]. The principle of the technique is to provide additional facilities within the circuit, so that the storage devices can be tested separately from the rest of the circuit; the future state of the internal variables can be set to any desired value independent of their present values; and the values of the internal variables can be accessed and observed directly. These facilities can be achieved by establishing a scan path through the storage devices, as shown in Figure 4.3. The scan path operates in two modes. In normal mode, the storage devices in the scan path are not linked together and the normal operation of the circuit is not affected. In scan mode, the storage devices are linked to form a shift register. The serial input and serial output provide controllability and observability to the internal states of the circuit, when in scan mode.

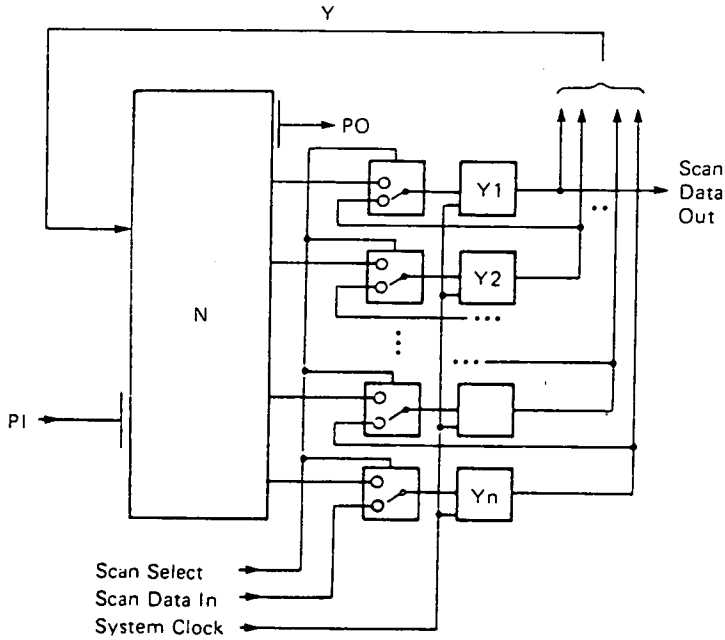


Figure 4.3. Scan path in design for testability.

Level sensitive scan design (LSSD) is a method of constructing scan paths which relies on strict design rules and guidelines. They are designed so that their operation is as independent as possible from the circuit's a.c. parameters, such as degraded rise and fall times, degraded propagation delays, or other faults that may introduce race or hazard conditions. As a result, the potential effect of failure mechanisms that cause timing faults is reduced [160].

The method of testing using scan path is as follows. Firstly the scan path is itself tested. This is done by selecting scan path mode, i.e. the storage elements configured as shift register. The status and operation of each storage device is tested using the Scan Data In, Scan Data Out, and Clock facilities shown in Figure 4.3. The test procedure uses a flush test followed by a shift test. Flush test begins by initialising the storage elements to 0. Then a single 1 is clocked through the scan path from

the Scan Data In input to the Scan Data Out output. The test can be repeated with a single 0 flushed through a background of 1s. Flush test checks the ability of each storage device to assume a 0-state and a 1-state, and the ability to transfer the stored state to the output. Shift test consists of clocking the sequence 00110011... through the scan path shift register. This sequence exercises each storage device through all combinations of present state and future state [158].

Secondly, the circuitry between scan path nodes can be tested. This is done by selecting scan path mode and shifting a predetermined test pattern into the storage devices. Also, a set of test vectors are applied to the primary inputs. Then the circuit is switched to normal operation. The steady state output response of the circuit under test can now be clocked into the storage devices. Finally, scan path mode is reselected, and the contents of the storage devices are clocked out. These values, plus the values directly observable on the primary outputs, can be compared with the expected fault-free response.

The total test time is determined mainly by the number of stages in the scan path which, in turn, is determined by the number of individual logic blocks to be tested. Optimum scan testing requires the inclusion of a complete scan path which leaves no sequential logic circuits during test mode. However, speed, performance, or area constraints, may restrict the use of this technique, with the result that parts of the circuit are sequential during the test.

The implementation overhead of a scan path test strategy, in terms of additional design and silicon area, depends on the basic structure of the circuit, and the

availability of circuit elements that are suitable for conversion into scan path elements. The simplest form of scan path test strategy is to add scan path shift registers to the VLSI design. Clearly this involves increased circuit area. A more attractive scan path implementation involves functional conversion of existing circuit elements into the required reconfigurable storage elements, thus reducing test area overheads to a minimum. Such a strategy is often forced upon the designer by the architecture and design software of semi-custom integrated circuits. In the UK5000 gate array [161], for example, the rows of uncommitted logic cells are sandwiched between rows of predefined LSSD latches. When the designer requires a storage element he is forced to use one of the LSSD latches. In this way the design is guaranteed testable. In the case of the Eu349 correlator design, functional conversion of existing circuitry has been extensively used.

The effect of scan paths on circuit performance is only of importance when additional scan path register stages have to be included in the design. Otherwise, only increased loading and routing need be considered.

The primary advantage of the scan path method is that as few as three extra circuit pins need be used to allow test-enable, and data input and output. However, the scan path merely allows access to internal circuit nodes to enhance controllability and observability. Testing circuits that have scan paths incorporated still requires external test pattern generation, and test response monitoring, to derive the test result.

4.3.4. Built-In Self Test Methods

Built-in self test (BIST) is a design for testability strategy in which test pattern generation and circuit response monitoring is performed within the system. This can be done either concurrently or nonconcurrently. Concurrent (on-line) methods use a variety of error-detecting, error-correcting, and self-checking codes. Nonconcurrent methods require an external activation which initiates the built-in test and inhibits the normal function of the circuit. The advantages of self test are that the test may be repeated as and when necessary during the service life of the system, and not simply at manufacture. For example, the system may be configured to initiate a self test automatically at each power-on. This thesis is primarily concerned with nonconcurrent self test methods, and the remainder of this section shall deal with two implementation techniques for built-in test. These are signature analysis [162,163], and BILBO (Built-In Logic Block Observation) [164,165].

In built-in test, it is essential that the the test pattern is short, or at least that it can be generated easily by a small amount of additional circuitry. The same criterion applies to test response monitoring. Test pattern generation can be simplified by using pseudo-random binary sequences (PRBS) [166] which are easy to generate on chip using a simple linear feedback shift register (LFSR), as shown in Figure 4.4.

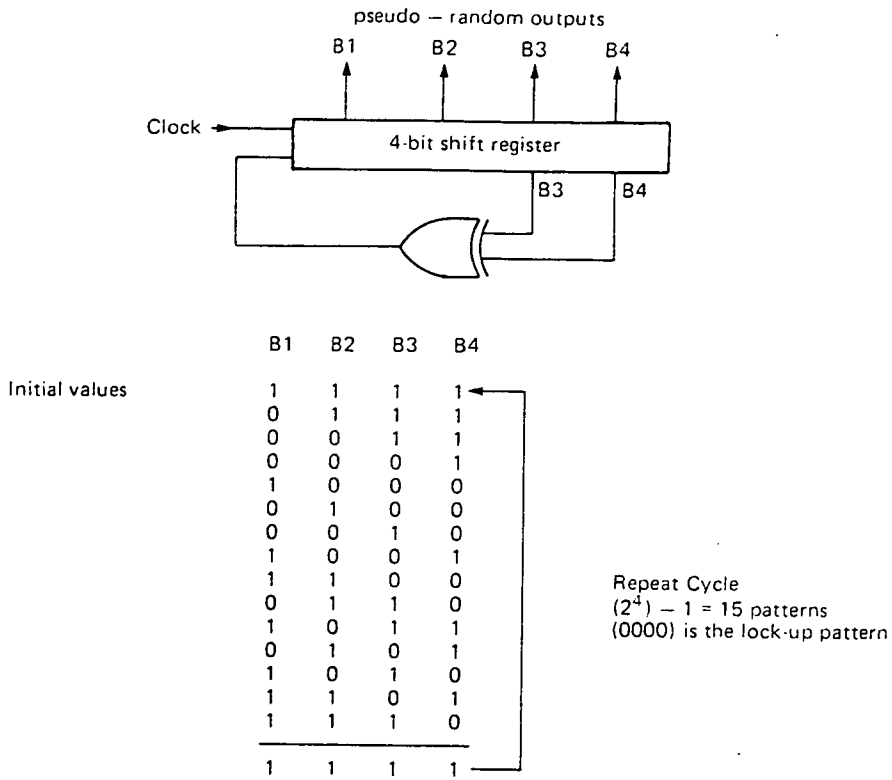


Figure 4.4. Configuration of a PRBS generator.

Data compression techniques, such as signature analysis, can greatly reduce the problem of test response monitoring [167,156]. Signature analysis is carried out using a linear feedback shift register, adapted to perform cyclic redundancy checking (CRC) on the test response sequence, as shown in Figure 4.5.

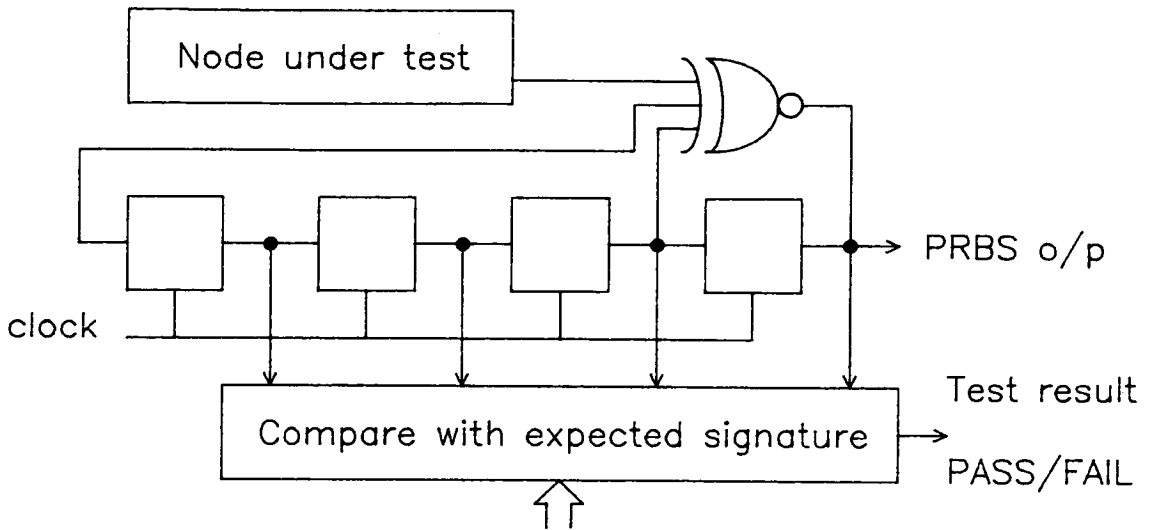


Figure 4.5. Signature analysis register.

The test sequence is sampled and clocked into the shift register. The contents of the shift register are influenced not only by the next sampled value of the test sequence, but, by virtue of the feedback structure, by the current contents of the register. In this way, any corruption of the sampled bit stream causes a corresponding corruption in the contents of the shift register. At the end of the test period, the accumulated contents of the shift register represents the signature of the node under test. The signature is compared with the expected fault-free signature, and a match indicates that the node is fault-free; a mismatch indicates that the response sequence is corrupt in some way. For CRC signatures, the probability of a corrupt data stream generating the same signature as the fault-free data stream is extremely low, quickly approaching 2^{-n} as the length of the data stream exceeds the length n of the shift register [162].

The BILBO technique [164,165] is a recent innovation which draws together all the main elements of design for testability, including pseudo-random test pattern

generation, scan path, and signature analysis. The technique reduces the test overhead by exploiting the shift register elements, common to all three schemes. The basic BILBO element is illustrated in Figure 4.6.

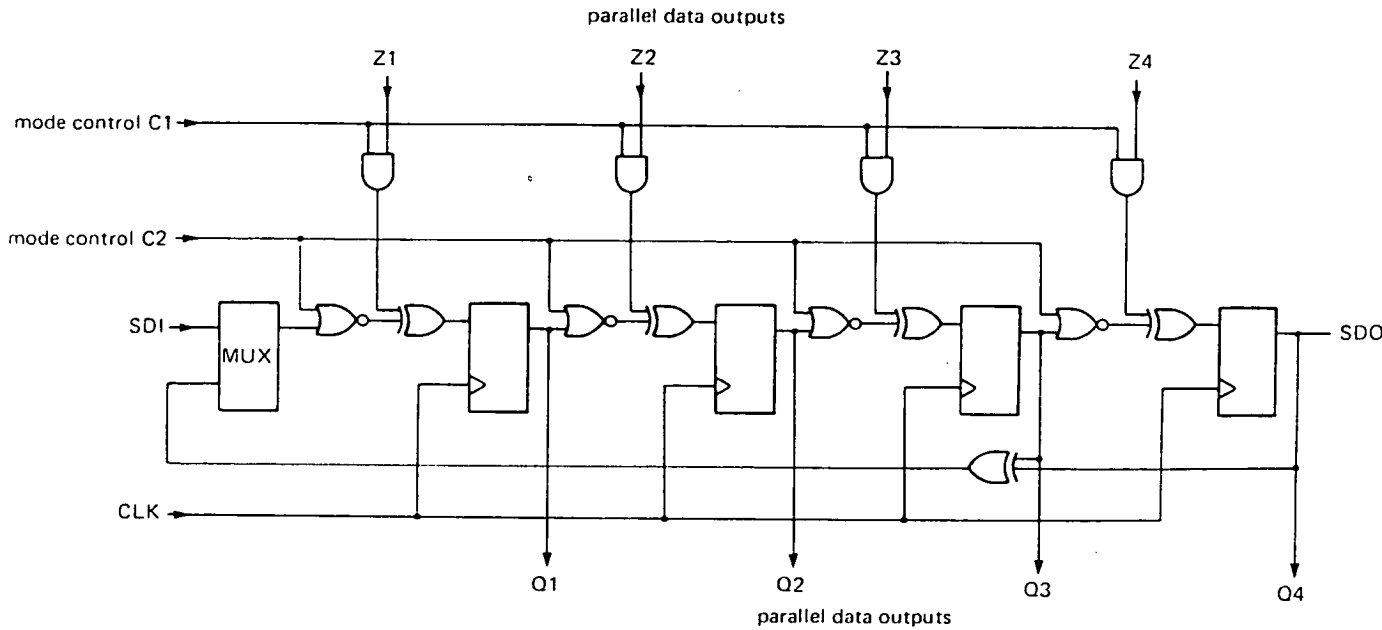


Figure 4.6. Basic BILBO element.

Each BILBO consists of a latch register and some additional gates for shift and feedback operations. Four different functional modes can be selected using the two mode controls C1 and C2 [164]. In the first mode ($C1 = 1, C2 = 1$), each latch is independent and can be used in normal operation. In the second mode, the BILBO is configured as a shift register and operates as a scan path ($C1 = 0, C2 = 0$). In the third mode ($C1 = 1, C2 = 0$), the BILBO is functionally converted into a multiple input signature register (MISR), and in the fourth mode ($C1 = 0, C2 = 1$) the latches are reset.

Multiple input signature registers can perform either pseudo-random sequence generation, or signature analysis. PRBS generation is achieved by setting the parallel inputs

to zero. As a signature analysis register the BILBO can operate in two modes: serial input, or parallel input. In serial input mode, the test data is clocked into Z1 while the remaining parallel inputs are held at zero. In parallel input mode, the test sequences are clocked into some, or all of the Z-inputs. The theory of multiple input signature analysis is complex, and is beyond the scope of this thesis. The most important aspect of signature analysis, as regards this thesis, is that the probability of fault detection is very high. It can be shown that the probability of detecting errors from L input vectors of m bits each, by an n bit MISR is [167]

$$P = 1 - \frac{2^{mL-n}-1}{2^{mL}-1} \quad 4.1$$

assuming all error sequences to be equally likely.

4.4. VLSI Design for Testability in the Eu349 Correlator Chip

This section contains a summary of the design features that are relevant to the Eu349 correlator chip. Details concerning the operation of the architecture have been described in Chapter 3; details of the chip design will be presented in Chapter 5.

A block diagram showing the main elements of the prototype correlator is presented in Figure 4.7. The figure shows an array of coincidence detectors and integrating counters, whose inputs and outputs are linked together by two shift registers, the data shift register (DSR), and the overload shift register (OSR). In test mode the DSR and OSR act as scan paths, while the integrating counters perform signature analysis. Signature analysis provides a self test of the integrating counters, and after a

complete test period, the counters contain the compressed signatures of each correlator stage. There are only two primary data inputs to the correlator, therefore an exhaustive functional test is possible, and only requires four different test patterns. Each test pattern, however, must be repeated for the number of clock cycles necessary to complete the signature analysis. In a complete test period, four integrating counter self tests, and one exhaustive test of the coincidence detectors will have occurred.

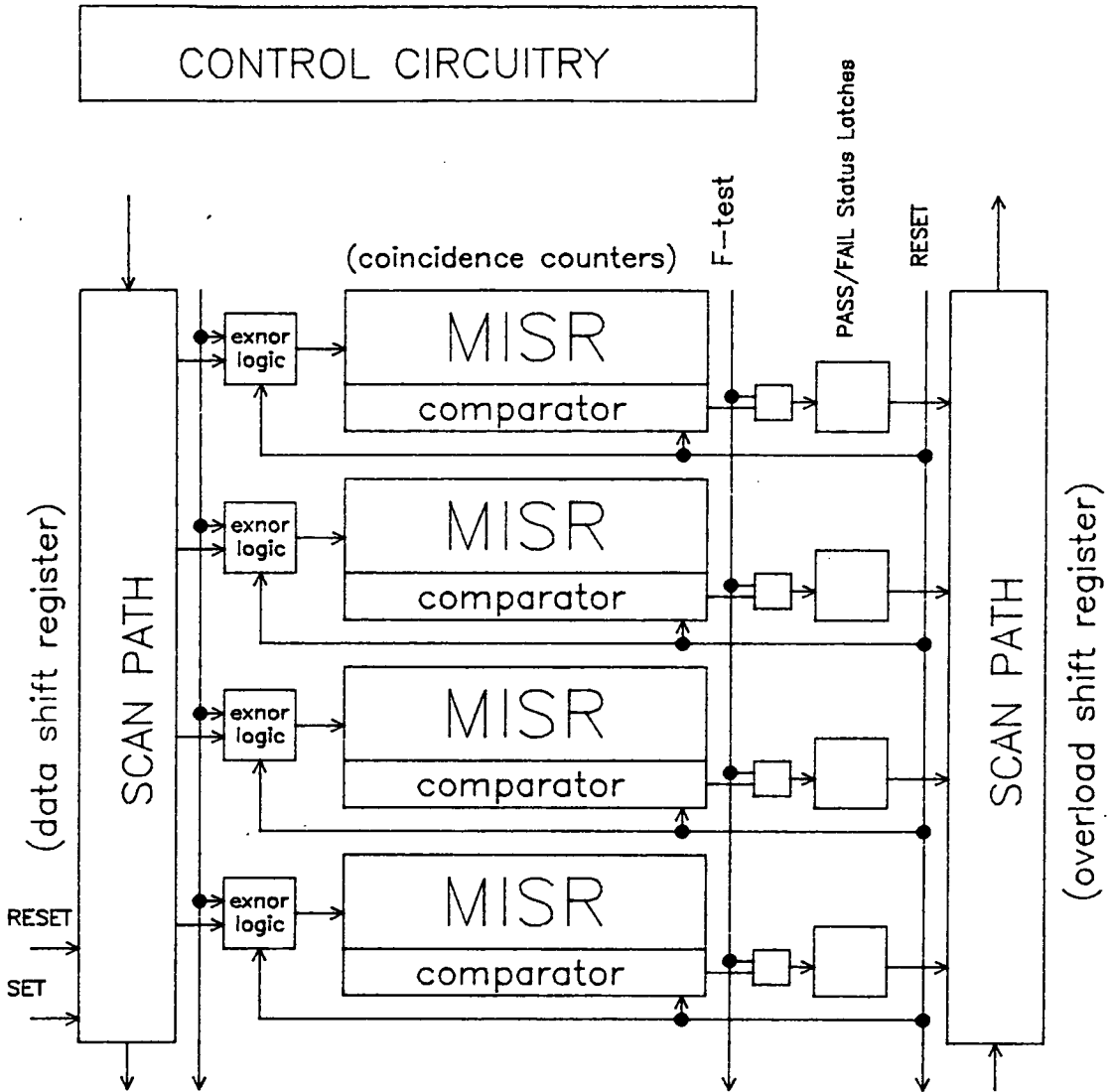


Figure 4.7. Correlator block diagram showing Built-In Self Test features.

At the end of each integrating counter self test the signature must be checked for correctness. This is done by an external signal called Fidelity-Test (F-Test), and the result, a single GO/NO-GO status bit is stored in the associated overload latch. The results of the full test for each correlator stage in the array may then be examined using the overload shift register in scan path mode.

The architecture of the Eu349 correlator uses the results of the self test (the GO/NO-GO status bits) to provide yield enhancement and fault tolerance. This aspect of the design is summarised in Section 4.7, and is addressed in more detail in Chapter 5.

4.5. Integrated Circuit Yield Statistics

4.5.1. Scope

The integrated circuit described in this thesis contains approximately 7500 MOS transistors interconnected to perform a specific electronic function. The probability that all the devices and their interconnections will function correctly depends on the control exercised during the series of complex processing steps used in their manufacture. The fraction of chips that satisfy the final test programme is called the yield. This section of the thesis deals with the mathematical models used to predict yield. Yield statistics are important in both controlling a semiconductor fabrication process, and in predicting the yield of future semiconductor products. They are also essential for analysing (or anticipating) the effectiveness of a yield enhancement scheme. It is this particular application of yield statistics that is of primary interest here.

The yield associated with integrated circuit fabrication can be divided into three parts. The first part results from catastrophic defects, such as wafer breakage, missing or erroneous processing steps etc., which prevent the circuits ever reaching final test. These defects will not be included in the discussion. The second part, known as pre-assembly test yield, deals with localised process defects, and the third part takes account of faults caused by packaging. The main area of interest here is the

second yield category, the pre-assembly test yield. This yield can be divided into two classes. Firstly, there are gross yields, which are the result of gross defects, such as process parameter variations, causing large areas of the wafer to fail, and secondly, random yields, which are the result of random defects, such as thin oxide pinholes.

The dependence of yield upon chip area has been extensively studied in the literature. Various theories have been presented, and analytical expressions derived to fit statistical data based on defect density distributions [168,169,170,171,172,173,174]. The work is based on random defect distributions, and the papers differ in their treatment of various defects being distinguishable or indistinguishable from each other.

Attempts at yield calculations that take redundancy into account have largely concerned memory chips [175,176,177]. The model presented by Schuster [177] is based on the exponential dependence of yield on the active chip area. The defects are separated as correctable, uncorrectable, and gross imperfections, and the net yield is calculated as the product of these three independently calculated yields. Stapper et al [175] have described a yield model with redundancy based on the Gamma distribution of defects. They then use mixed Poisson statistics to derive a yield expression to describe the yield of redundantly designed memory chips.

Researchers have also been concerned with redundancy in non-memory VLSI chips [178,153,179], and they all agree that redundantly designed circuits have more chance of working than nonredundant designs. Mangir et al [153], in their model, account for the effects of the complexities of areas, connectivities between different areas, and the effect of regularity of interconnections, which would

affect the processing tolerances, and hence yield.

Before describing in detail a yield model for random defects, it is necessary to describe the yield losses due to gross defects.

4.5.2. Yield Loss due to Gross Defects

Gross defects, which are normally associated with errors in the process parameters, may cause large areas, or entire wafers, to have no functioning chips. Examples of these parameters are transistor gain, threshold voltages, contact resistance, and parasitic capacitances. Entire wafers will fail if the values of these parameters fall outside of their specified range. In marginal cases parts of wafers may fail, as shown in Figure 4.8. Gross yield losses may also be caused by errors in photolithographic processes. Examples of these are over or under exposure of the photosensitive resist material, optical distortions, and misalignment of mask patterns. The failures do not cause the chips to fail in random patterns on the wafer. This is why they must be treated separately from random defects.

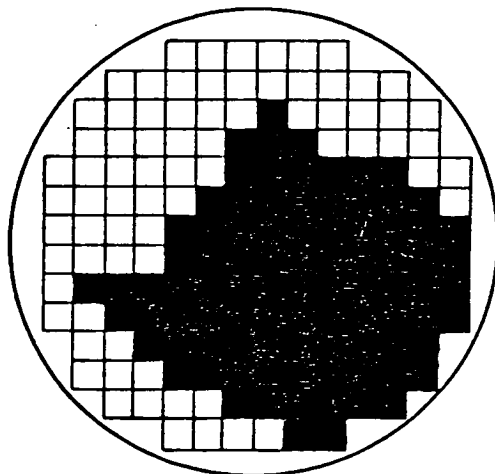


Figure 4.8. Wafer map showing gross yield. The shaded chips are functioning correctly.

Special test circuits for measuring the process parameters are usually fabricated on the wafer, either in the free space between the chips (the scribe channel), or in reserved areas of each chip (test stripe), or in a small number (5-6) of chip size replacement "drop-ins". Mask misalignment can also be measured in this way. The fraction of test devices whose measured parameters lie within the required range, contributes to the gross yield.

Stapper [180] gives an example of the relative yield losses occurring in the manufacture of a 64k-bit random access memory (RAM) chip. These are reproduced in Figure 4.9. The actual values of the yield losses are proprietary information and have not been published. Note that the parametric yield accounts for less than 5% of the total yield loss.

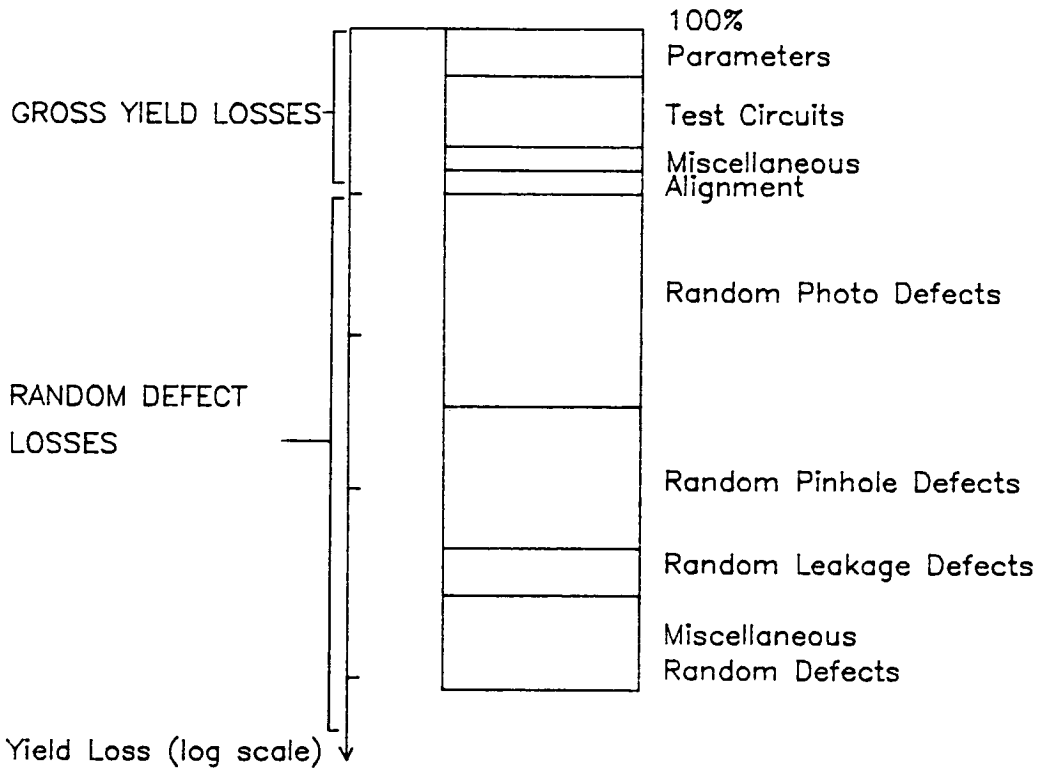


Figure 4.9. Relative yield losses. Random defects cause most of the losses.

Figure 4.9 represents data obtained from a specific process line for a specific product, and therefore may not be applicable to other circuit types or fabrication facilities.

4.5.3. Yield Model for Random Defects

The data shown in Figure 4.9 indicates that random defects cause approximately five times as many chips to fail than gross defects. Random defect models are, therefore, an important factor in semiconductor yield statistics.

Due to the nature of random defects, and to the complexity of the fabrication process, it is impossible to

tell whether observed defects will cause actual chip failures. Therefore, the random defect model must be divided into two parts. The first part deals with the average number of failures or faults that can be caused by a large number of different defect mechanisms. The second part deals with the statistical distribution of the average number of faults per chip. According to this theory, each defect type is associated with a probability that it will cause a failure. This probability can be multiplied by the number of defects in the corresponding category to obtain the average number of failures or faults per chip. This must be done for each defect type. Several failure models that have been developed for this purpose are described by Stapper et al [180]. However, this theory leads to very cumbersome expressions involving hundreds of terms, the data for many of which would be very difficult to obtain. Fortunately, for the purposes of this thesis, a simpler model using a single average defect density will suffice.

The simple theory using Poisson statistics on a random distribution of faults, predicts that the yield is proportional to the exponential of the average number of faults per chip, or the chip area (if the fault distribution is constant). In practice, however, it has been observed that the defect distribution is non-uniform and the yield falls off less sharply, but nevertheless significantly with increasing chip area [181,182]. To account for this, a wide range of random defect models have been reported. Price [169], and later Mangir et al [153], maintained that defects should be modelled by Bose-Einstein statistics. Others have favoured Maxwell-Boltzman statistics [183,170,184]. Stapper et al [180] discuss Poisson, Binomial, and Generalised Negative Binomial statistics, and conclude that each one of these may be applied to yield theory. The correct model is the one

which fits the data best, and according to Stapper et al the Generalised Negative Binomial distribution is the most suitable for modelling present day semiconductor manufacturing.

When simple Poisson statistics are used, the yield Y_R due to random defects is given by

$$Y_R = e^{-\lambda} \quad 4.2$$

where λ is the average number of faults, given by the product of the defect susceptible chip area A , and the average defect density D ,

$$\lambda = AD \quad 4.3$$

However, the average value of faults per chip λ varies from chip to chip, from wafer to wafer, and from batch to batch. To take account of these variations a yield model that uses the sum of many thousands of fault terms is required. The sum may be approximated by an integral. The yield is then given by

$$Y_R = \int_0^{\infty} e^{-\lambda} g(\lambda) d\lambda \quad 4.4$$

where $g(\lambda)$ is a probability distribution function of faults per chip. This model was reported by Murphy in 1964 [168] with uniform and triangular distributions given for $g(\lambda)$. Murphy's results, however, took no account of the fact that defects in semiconductor fabrication tend to cluster. A more suitable yield model reported by Stapper in 1973 [171] uses a Gamma distribution for $g(\lambda)$, and an expression for yield is obtained of the form

$$Y_R = (1 + \sigma^2/\lambda)^{-\lambda^2/\sigma^2} \quad 4.5$$

where λ is the mean, and σ is the standard deviation of the Gamma distribution. Defining a constant

$$\alpha = \lambda^2 / \sigma^2 \quad 4.6$$

gives

$$Y_R = (1 + \lambda / \alpha)^{-\alpha} \quad 4.7$$

This distribution is known as the Generalised Negative Binomial distribution. The parameter α depends on the spread of the fault distribution and takes into account the clustering of defects.

Clustering is believed to be caused by the aggregation of particles that have collected in the manufacturing equipment. When shaken loose by vibrations, pressure changes, etc., these clumps of particles form clouds in the fluids used for processing the integrated circuits. Where such clouds reach the wafer surface, particles are clustered. Even when contaminating particles are uniformly distributed in the fluids, they are electrostatically attracted to the nearest edge of the wafer. This leads to edge clustering, a phenomenon in IC fabrication that has been widely observed [185,186,170,187].

A comparison of yield models is shown in Figure 4.10.

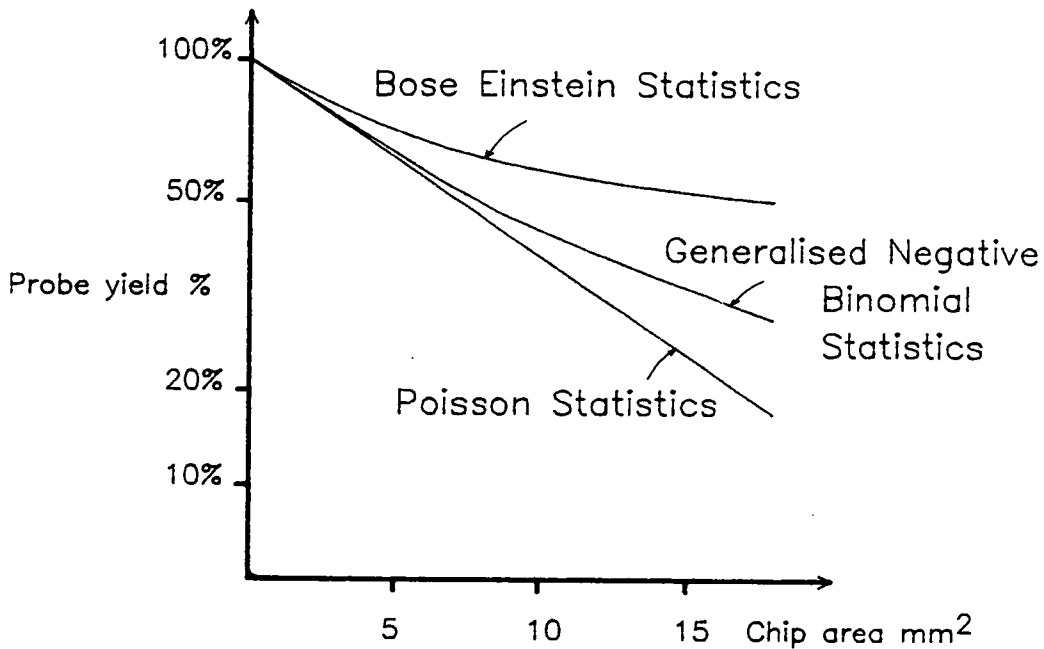


Figure 4.10. Comparison of yield models.

It is interesting to note, that the expressions for yield given in Figure 4.10, can be linked to each other by the value of the clustering parameter. Low values of α are used to model severe clustering. When $\alpha=1$ the yield model in 4.7 takes the form $Y_R = (1+\lambda)^{-1}$ which is the same, *mutatis mutandis*, as the Bose-Einstein model reported by Price [169], and Mangir et al [153]. When α approaches infinity, 4.7 approaches $e^{-\lambda}$, which is the same as the simple Poisson model in 4.2. In this case there is no clustering, i.e. a uniform distribution.

4.5.4. General Yield Model for VLSI Chips with No Redundancy

The yield model described in the previous section can be considered a general model for random yield. To complete the model, gross yield, which was discussed in

Section 4.5.2, must be taken into account. Gross yields may be incorporated into the general model simply as yield multipliers. Thus, pre-assembly yield Y may be expressed by

$$\begin{aligned} Y &= Y_G Y_R \\ &= Y_G (1 + \lambda / \alpha)^{-\alpha} \end{aligned} \quad 4.8$$

where Y_G is the average yield due to gross defects, listed in Section 4.5.2.

In practice, it is a difficult task to obtain consistent data for even these key features of a yield model. This is due to several important causes:

- a) The information is proprietary and is rarely disclosed.
- b) State of the art processes often change more quickly than the data can be compiled.
- c) A yield model that has been derived for one particular process will often not apply to another process, even of the same type.
- d) A yield model that has been derived for one particular circuit will often not apply to a new circuit, because of the dependency on circuit complexity and interconnect [153].

Therefore, the general yield expression is often simplified by assuming the values $\alpha \rightarrow \infty$ and $Y_G = 1$, to produce the Poisson yield expression,

$$Y = Y_R = e^{-\lambda} \quad 4.9$$

which assumes a uniform distribution of faults. This expression tends to produce a lower yield estimate than is observed in practice, so it can be considered as a lower bound. The upper bound can be expressed by the Bose-Einstein model, which is obtained by setting $\alpha=1$. This results in the expression for yield,

$$Y = (1+\lambda)^{-1} \quad 4.10$$

4.5.5. Yield Model for VLSI Chips with Redundancy

In the previous sections it has been stated that integrated circuit yield is reduced by gross defects, and by faults caused by random defects in the materials and photolithography. In a yield enhancement scheme, where faulty circuit stages are replaced by redundant stages, or in a scheme where faulty stages are simply bypassed to leave a partially functioning chip, it has been observed that the defect susceptible portion of the chip is divisible into two areas [177]. The first area is where random defects can cause failures in the circuit stages or modules (the words stage and module are synonymous in this context). Defects in this area are correctable by replacing the faulty module. The remaining defect susceptible portion of the chip is uncorrectable, and any defects occurring in this area cause the chip to fail. Uncorrectable circuitry includes redundancy switching circuits, chip test status latches, clock lines and interconnect, input/output buffers etc. The net yield Y_E after the enhancement scheme has been implemented, is, therefore, the product of the gross defect yield Y_G , the correctable random defect yield Y_{CRD} , and the uncorrectable random defect yield Y_{UNC} , that is [188],

$$Y_E = Y_G \cdot Y_{UNC} \cdot Y_{CRD} \quad 4.11$$

With the aid of a block diagram of the integrated circuit, shown in Figure 4.11, expressions for the correctable and uncorrectable yields can be obtained.

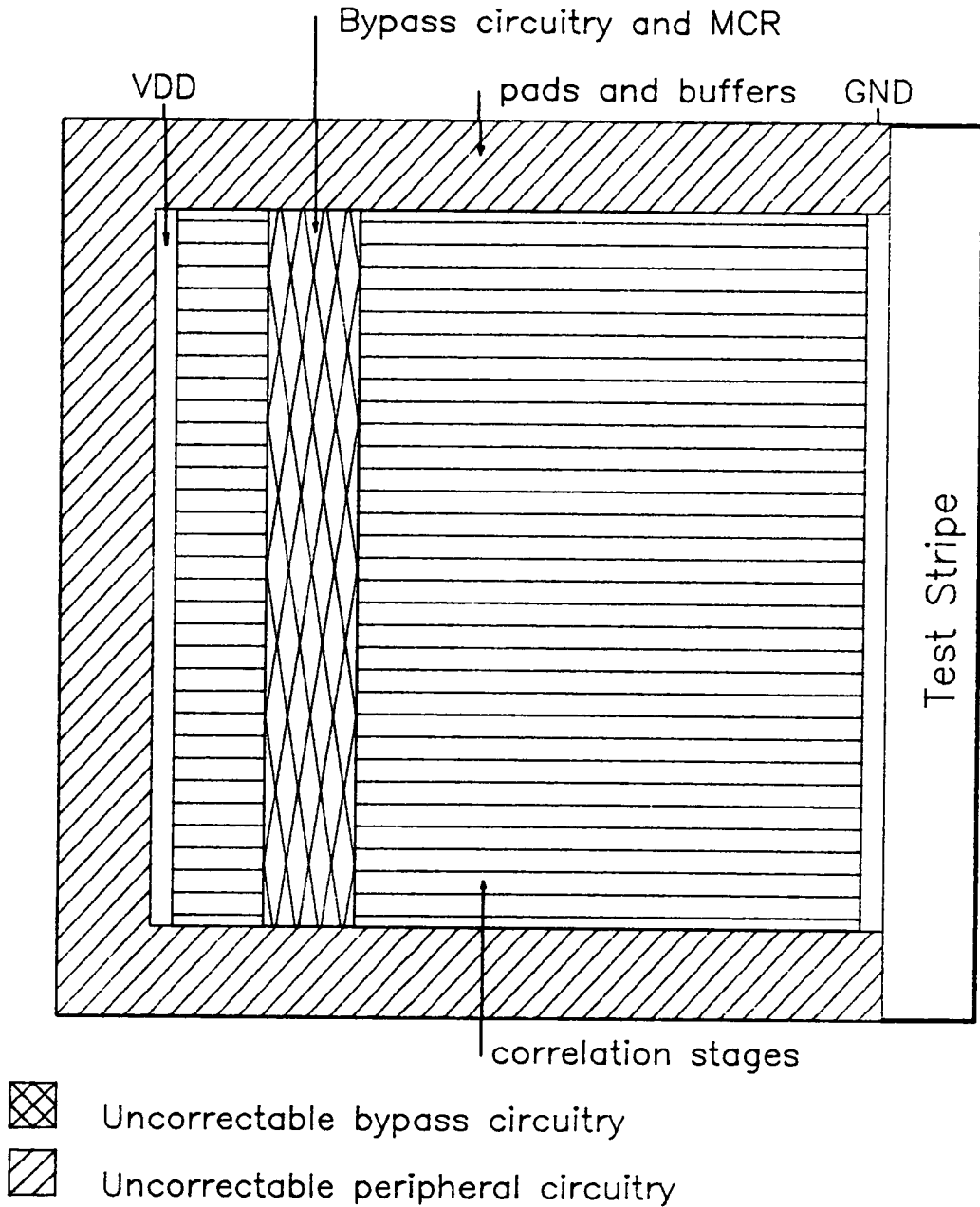


Figure 4.11. Block diagram of Eu349, showing correctable and uncorrectable area in the yield enhancement scheme.

Figure 4.11 shows an array of correlation stages

surrounded on three sides by pad drivers and buffer circuitry, some miscellaneous logic, and power and clock lines. This area, shown hatched, is uncorrectable. In addition, the area shown cross hatched contains the multiplexer control register; this too is uncorrectable. The yield of the hatched and cross-hatched region is denoted Y_{UNC} and is expressed by

$$Y_{UNC} = e^{-DA_{UNC}} \quad 4.12$$

where A_{UNC} is the uncorrectable area, and D is the defect density.

The module yield Y_m can be calculated by any of the yield models discussed in Section 4.5.2. For simplicity, a Poisson defect distribution will be assumed here. Thus, an expression for the module yield is

$$Y_m = e^{-DA_m} \quad 4.13$$

where A_m is the module area. This expression can now be used to derive an expression for the correctable random defect yield Y_{CRD} .

The correctable yield of a one dimensional array of identical modules, each having the probability Y_m of working, is determined using binomial statistics as follows.

If there are no spare (redundant) modules, then the yield of the array is simply

$$Y_{CRD} = Y_m^N \quad 4.14$$

where N is the required number of modules in the array. If there is one spare module, the yield becomes

$$Y_{CRD} = Y_m^{N+1} + (N+1)Y_m^N(1-Y_m) \quad 4.15$$

Here the first term represents the probability that all $N+1$ modules are functioning, and the second term represents the $N+1$ possible combinations of N working modules and one defective module.

Extending this approach to the case where there are N required modules and S spare modules in the array, then the probability of having at least N working modules from an array of $N+S=M$ modules is

$$Y_{CRD} = \sum_{j=0}^S \frac{M!}{(M-j)!j!} Y_m^{(M-j)} (1-Y_m)^j \quad 4.16$$

where $\frac{M!}{(M-j)!j!}$ represents the number of possible combinations of M modules taken j at a time.

Finally, by substituting 4.16 into 4.11, the expression for the enhanced yield can be written as

$$Y_E = Y_G Y_{UNC} \sum_{j=0}^S \frac{M!}{(M-j)!j!} Y_m^{(M-j)} (1-Y_m)^j \quad 4.17$$

where $Y_m = e^{-DA_m}$ and $Y_{UNC} = e^{-DA_{UNC}}$. This expression has been evaluated for a range of parameter values, and the results, enhanced yield versus redundancy, are plotted in Figures 4.12(a), 4.12(b), and 4.12(c). In each figure, the total number of modules M remains constant, and the gross yield has been normalised to unity. The following observations can be made:

- a) The yield saturates after a certain amount of redundancy. This occurs when the yield of the correctable areas approaches unity, and increasing redundancy ceases to have a significant effect on overall yield.

- b) The increase in yield is greatest for chips with the highest defect density. Therefore redundancy is most effective in low yielding processes.
- c) As the uncorrectable area increases, the yield decreases.

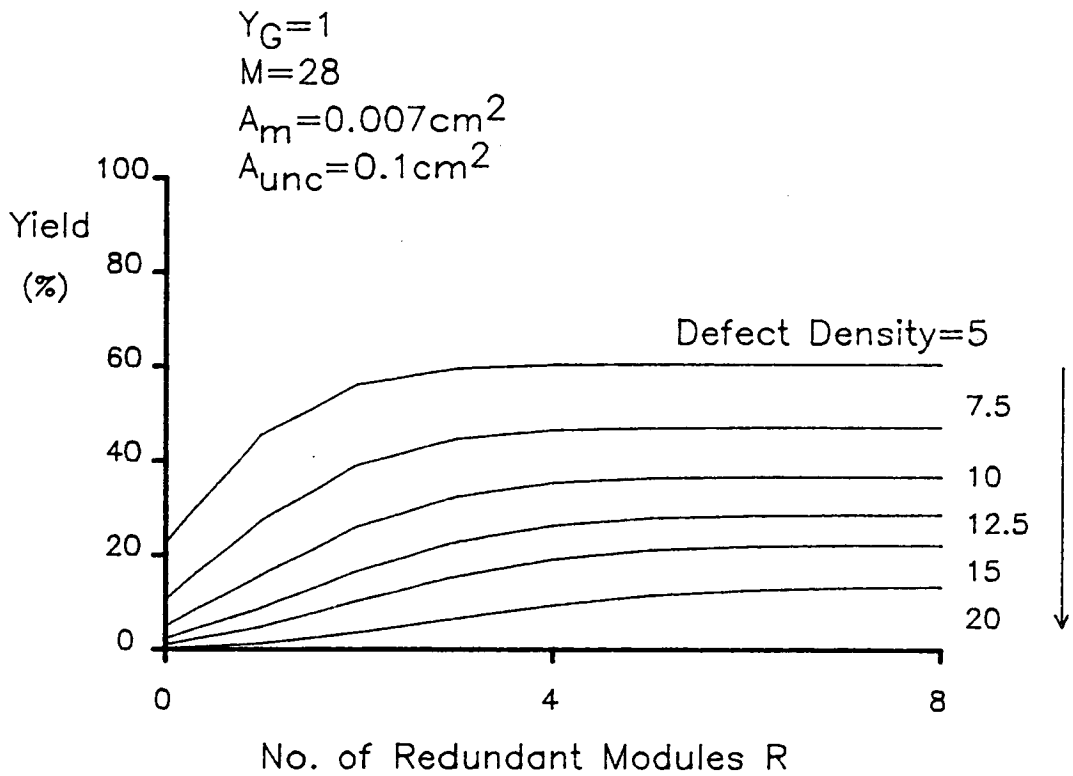


Figure 4.12(a). Yield vs. Redundancy for various defect densities.

$Y_G=1$
 $M=28$
 $A_m=0.007\text{cm}^2$
 $D=10\text{ defects/cm}^2$

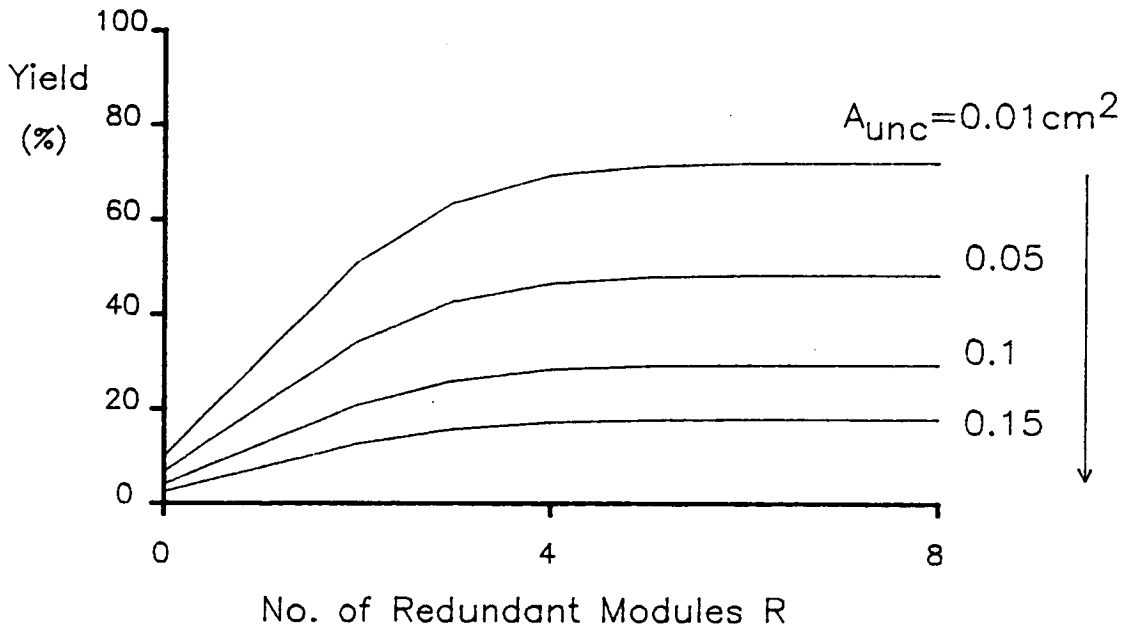


Figure 4.12(b). Yield vs. Redundancy for various values of uncorrectable area.

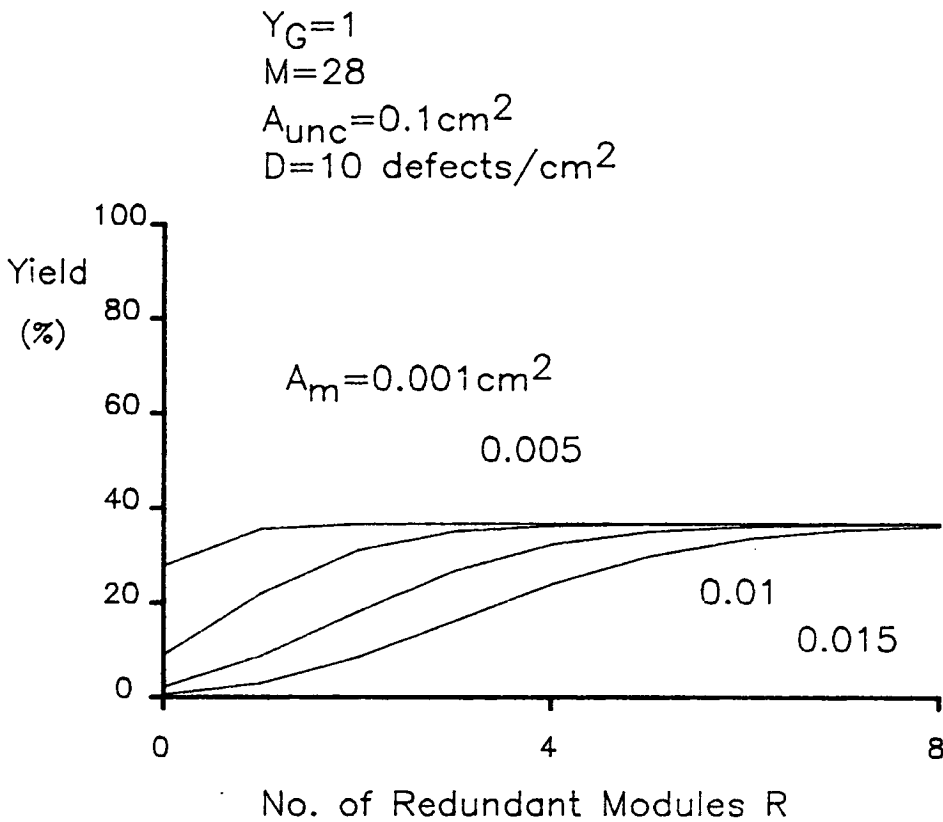


Figure 4.12(c). Yield vs. Redundancy for various values of module area.

4.5.6. Cost of Redundancy

The increase in net yield is not obtained without cost. The redundant circuits require extra chip area. Therefore, there are fewer chips per wafer. Also the redundancy scheme requires switching mechanisms, or additional circuitry which is uncorrectable and thus detracts from the yield. The compromise between the increase in yield due to the action of a redundancy scheme, and the decrease in yield due to the implementation of such a scheme is discussed by many authors on yield enhancement [177,153,176].

The effective yield of a yield enhancement scheme is

found using the enhanced yield Y_E and the proportional increase in chip area that is required to implement the scheme. The penalty due to the increased area is expressed by ratio of chip area without redundancy A_0 , and the area with redundancy A_E . The effective yield is defined as the product of the enhanced yield and the area penalty term,

$$Y_{\text{eff}} = Y_E \frac{A_0}{A_E} \quad 4.18$$

For cost considerations a figure of merit FM may be defined. This takes into account the relative yield improvement and the relative increase in area required. The figure of merit is defined as

$$FM = (Y_E/Y_0)(A_0/A_E) \quad 4.19$$

where Y_0 is the yield without redundancy. If $FM > 1$ then a cost advantage is attained by the use of redundancy. Figure 4.13 shows how the figure of merit varies with redundancy. The relationship indicates that a circuit can be designed around an optimum amount of redundancy.

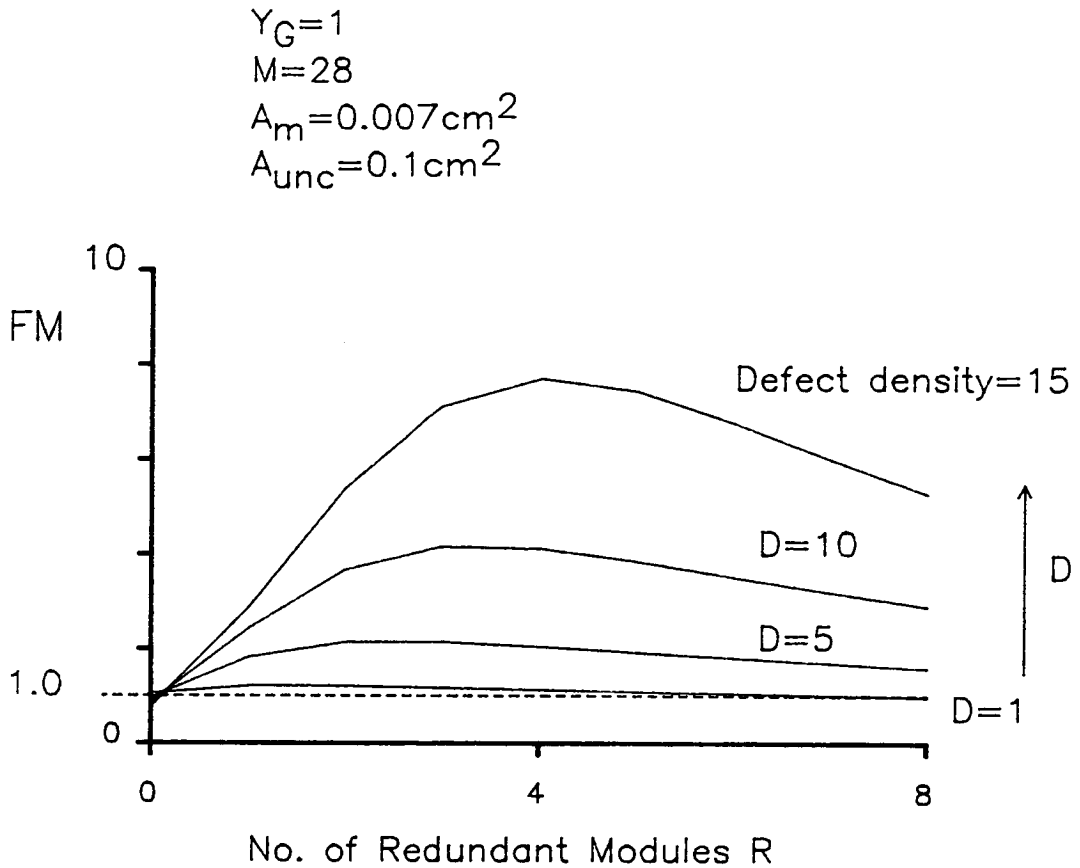


Figure 4.13. Figure of Merit vs. Redundancy for various values of defect density.

4.6. Yield Enhancement Techniques

4.6.1. Scope

This section deals with the concept of yield enhancement. Attention is focused, however, on yield enhancement techniques which incorporate redundant circuits and switching mechanisms, so that faulty circuit elements may be replaced by redundant ones after an initial test period. The discussion, therefore, does not include "on-line" self checking circuits [189].

4.6.2. Integrated Circuit Redundancy Schemes

There are many techniques for implementing redundancy schemes. These range from non-volatile "once only" configurations, which normally are carried out by the manufacturer, to the volatile schemes which may be configured as often as necessary by the host system. For example, discretionary interconnect layers provide a method of repair in non-volatile schemes, as do fusible links, and laser personalisation. Electrically programmable storage elements, and programmable links, are used to configure volatile redundancy schemes. Further more, latches and other electrically alterable configurations can be reprogrammed in the field if necessary. Thus, redundancy included on chip for yield improvement purposes can be used for field maintenance and improve reliability.

The design effort in VLSI is minimised by using regular repeated architectures. Also, chips with a large number of identical cells are the most obvious candidate for yield enhancement. Memory chips certainly have such an architecture and were among the first to benefit from redundancy techniques. They are particularly suitable since there is no interaction between the cells. As the interconnection complexity increases, either an increasing amount of circuitry must be dedicated to routing faulty cells, or a less flexible use of the spares has to be accepted. In 1967 Tammaru and Angell [178] proposed the concept of treating groups of interconnected elements, rather than single gates, as the smallest units to be tested and replaced with spares. In this manner the complexity and cost of reconfiguration can be reduced. Architectures for yield enhancement which are of interest here, consist of an array of identical cells. There are several reconfiguration schemes. These schemes fall into three categories: bypass schemes, nearest neighbour, and

simple chaining schemes.

4.6.2.1. Bypass schemes

Bypass schemes use a fixed sequence of cells but extra data paths are available so that one or more faulty cells may be bypassed, depending on which scheme is employed. These schemes should not be confused with chaining schemes which are described in Section 4.6.2.3. In the bypass scheme, the switching mechanism is part of the cell. Therefore, defective switching mechanisms can be bypassed. In the chain scheme, the switching network is regarded as separate from the array cells and accordingly must be defect free.

Examples of some bypass schemes are shown in Figure 4.14. These schemes have been compared by Moore and Day [187].

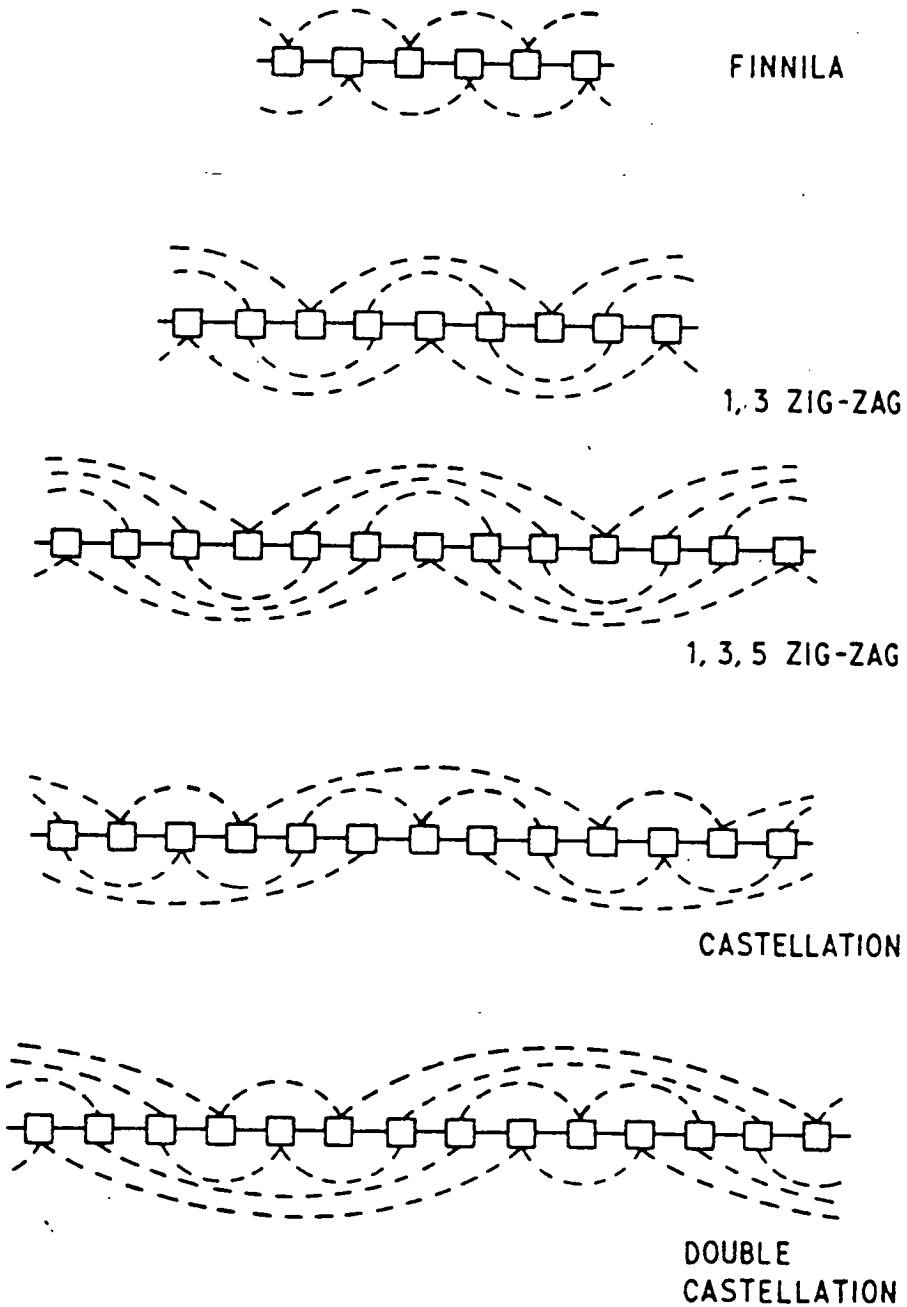


Figure 4.14. Bypass schemes.

They were selected because they contain no crossovers and can be mapped compactly into silicon. An example of the silicon layout for the 1,3 zig-zag scheme is shown in Figure 4.15.

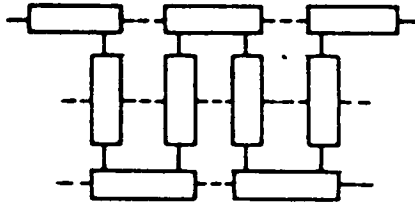


Figure 4.15. Layout of the 1,3 zig-zag scheme.

4.6.2.2. Nearest neighbour schemes

The nearest neighbour concept for yield enhancement is best suited to two dimensional arrays of regular cells as shown in Figure 4.16.

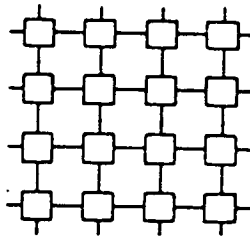


Figure 4.16. Nearest neighbour scheme.

The scheme depends on each cell being able to take any one of its neighbours as its successor. The resulting path therefore, is not fixed as in the bypass and chain schemes, but may snake around in any desired pattern [190,191].

4.6.2.3. Chaining schemes

This category contains the simplest of all yield enhancement architectures. A chaining scheme consists of a fixed array of cells which can either be connected into a chain or not. The main advantage of this scheme is its simplicity, in both implementation and configuration. The concept can easily be extended to 2 dimensional arrays and

to architectures with interconnectivity which would be too complex for the alternative schemes [192]. The main drawback of this scheme is that, unlike the others, the switching network must be defect free, because it is uncorrectable. The chain scheme, for a linear array is shown in Figure 4.17.

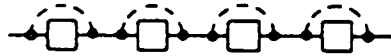


Figure 4.17. Chain scheme for yield enhancement.

4.6.3. Comparison of Redundancy Schemes

The differences between the above yield enhancement schemes can be described in terms of the amount by which they improve yield, and their implementation costs. The nearest neighbour schemes will not be considered because they are not suitable for linear array architectures, such as that of the Eu349 correlator chip. This leaves the bypass schemes and the chain schemes.

The degree of yield enhancement that may be obtained using a bypass scheme, is determined, to a large extent, by the complexity of the scheme, and by the routing algorithm. In one of the simplest algorithms, a faulty cell would enable the bypass route which connects its natural predecessor to its natural successor. This algorithm, however, only works for single faults, since two or more consecutive faulty cells result in total chip failure. This is illustrated by bypass 'a' in Figure 4.18. In a more sophisticated scheme, one that can implement a more complex routing algorithm, a faulty cell would enable bypass 'b' in Figure 4.18.

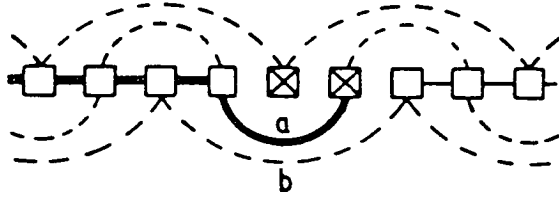


Figure 4.18. Bypass scheme with two consecutive faults.

Thus, an increase in fault tolerance is achieved by providing more bypass routes. A study by Moore and Day [187] has shown that yields are improved by using more complex zig-zag schemes, but for the same yield improvement the castellation schemes have consistently higher cost overheads, and therefore may be rejected.

Bypass schemes have an advantage that the switching circuitry is an integral part of the array cells. Thus, defective switches can be tolerated by the scheme. A disadvantage is that special terminating cells are required at the ends of the array. These cells select the desired start and finish paths, and connect them to the input and output pads respectively, as shown in Figure 4.19.

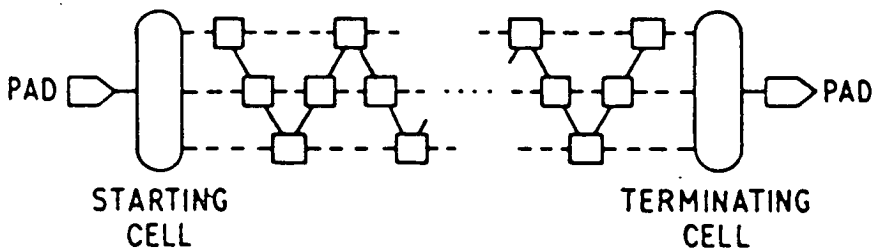


Figure 4.19. Terminating a bypass architecture.

By far the simplest yield enhancement technique is the chain scheme. In this case a faulty cell merely

enables its own bypass. It has an advantage that no special terminating cells are required. Also, it can tolerate consecutive faulty cells, although it is possible that signals may be required to go through several bypass switches before reaching the next working cell. The additional delays through these switches may reduce the maximum attainable clock rate. This aspect of a chain scheme may be viewed as a restriction in the number of consecutive faults that may be tolerated, if the circuit is to operate up to a specified maximum clock rate. The only serious disadvantage associated with the chain scheme is that the switching logic is critical and must work. Thus, the yield of chip is given, approximately, by the critical, uncorrectable areas of the chip.

Due to its simplicity, the chain scheme can easily be extended to provide yield enhancement in arrays that have more than single inter-cell connections, and to two dimensional arrays. The two dimensional array digital correlator, reported by McCanny and McWhirter [192], incorporates this type of yield enhancement technique.

4.7. Yield Enhancement Features in the Eu349 Correlator Chip

This section contains a summary of the yield enhancement features that have been used in the design of the Eu349 correlator chip. Details concerning the design can be found in Chapter 5.

In the Eu349 digital correlator, there are two interconnections per cell which require switching mechanisms. Therefore, in view of the expandability, and low cost, the chain scheme has been chosen to provide yield enhancement in the correlator array. The disadvantage that is associated with the chain scheme, in that the switching network

is uncorrectable, is considered to be outweighed by the advantages listed above, since the area of the uncorrectable switching network is less than 10% of the area of the correctable correlator array.

The yield enhancement scheme in the Eu349 correlator consists of, for each correlator stage, two multiplexers (one per inter-cell connection), and one controlling latch. The fault status of each correlator stage is stored in its associated latch. The stored information then controls the multiplexers; a faulty stage causes the relevant inputs to be switched to the respective outputs, thus isolating the faulty stage. A block diagram of the yield enhancement features of the Eu349 chip are shown in Figure 4.20. Details of the design are given in Chapter 5.

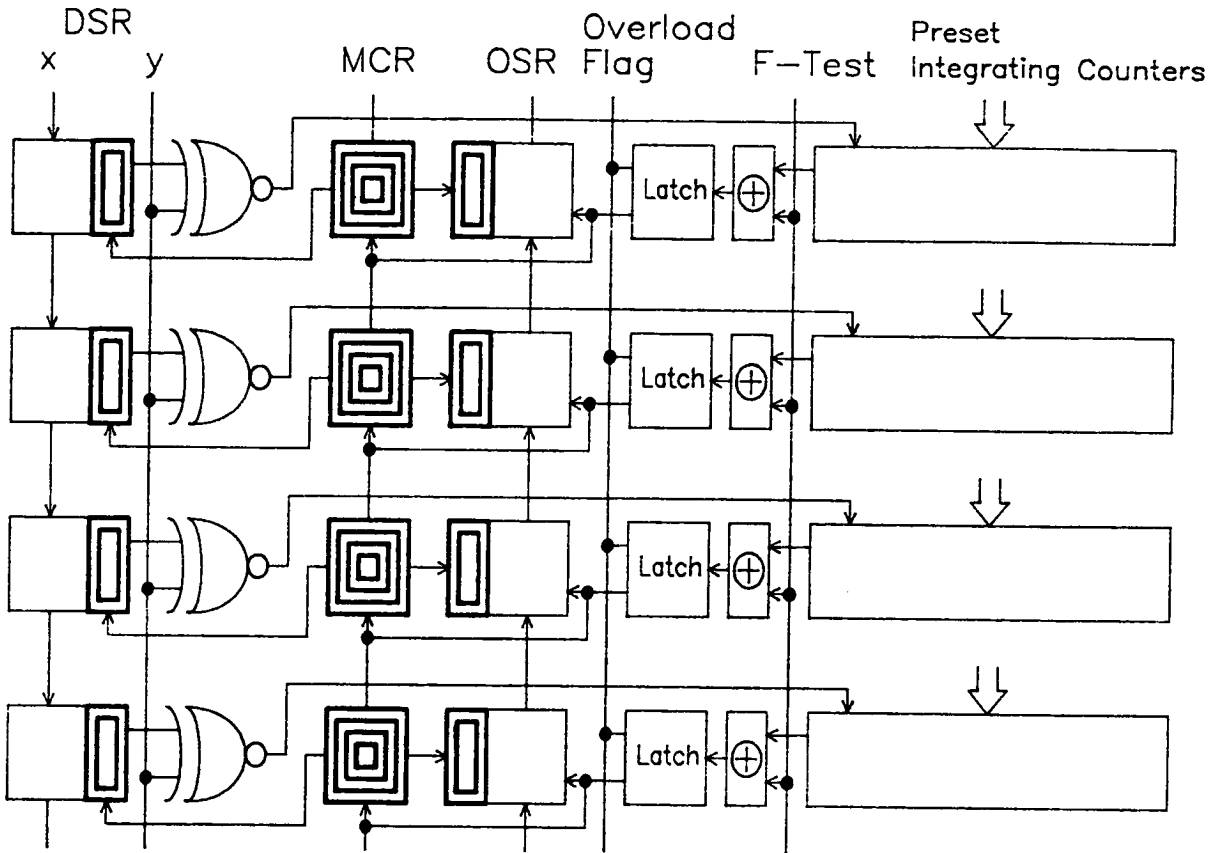


Figure 4.20. Yield enhancement features of the Eu349 correlator chip.

4.8. Summary

Two important subjects in integrated circuit design, namely testability and yield, have been discussed. Methods by which the testability of a design may be enhanced have been described and summarised for the particular case of the Eu349 correlator chip. Close linking of design and test has enabled the architecture of the Eu349 to achieve a high degree of testability at a very low overhead.

Yield enhancement through the use of redundant circuitry is of central importance to the design of the Eu349 correlator chip. Yield models and yield enhancement

techniques, have been described. A binomial model for the yield of redundantly designed circuits has been presented, and the curves produced by the model show that the yield of an array of identical modules, such as in the Eu349 correlator, increases rapidly with the addition of spare modules, but saturates to a level determined by the defect density and the uncorrectable area of the chip.

In the next chapter, details of the design, and test results of the correlator chip are presented.

CHAPTER 5

DESIGN AND TEST OF THE PROTOTYPE INTEGRATED CIRCUIT.

5.1. Introduction

In this chapter, details of the prototype chip design are presented. A "top-down" approach is adopted, and in Section 5.2, the description begins with reference to a "floor plan" of the basic correlator system. This floor plan represents the architecture of the chip before built-in self test and repair features are added.

In Section 5.3, the architecture is shown modified, to allow built-in self test and repair, and the circuitry required to perform self test and self repair is discussed. In Section 5.4, the integrated circuit design is described.

In Section 5.5, the test strategy is described. This section refers to test programs and configuration procedures for the Tektronix Digital Analysis System (DAS 9100). In Section 5.6, the test results for the batch of 130 chips are presented. The results show the effectiveness of the test strategy and yield enhancement scheme.

5.2. Architecture of the Basic Polarity Correlator

The theory of polarity correlation using the overloading integrating counter technique is presented in Section 2.6. Figure 5.1 shows the architecture of a correlator that implements the technique. The VLSI architecture offers high speed operation, long (programmable) integration time, and an arbitrary range of correlation time

delay or resolution. It consists of a data shift register, or DSR, connected to a parallel array of coincidence detectors and integrating counters. The counters each have a single bit output which indicates the overload condition of the counter. This overload output is latched, to be transferred subsequently, to the overload shift register, or OSR. The pattern held in the OSR may then be shifted serially off chip to display the correlation function. An additional output from the chip is the overload flag, which indicates when the first overload has occurred.

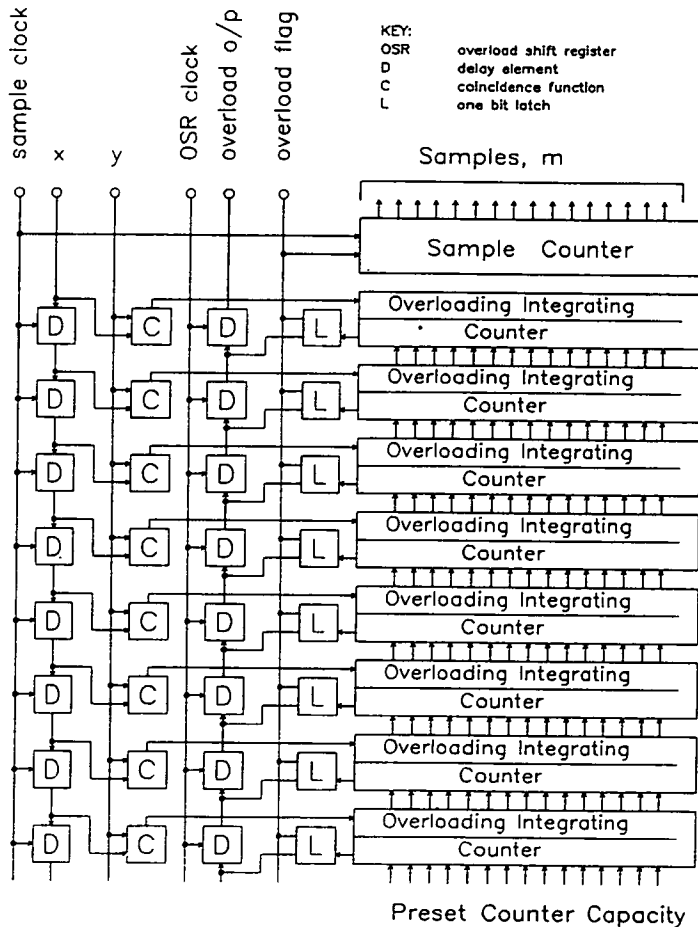


Figure 5.1. Architecture of the basic polarity correlator using the overloading integrating counter technique.

The control circuitry required for this architecture

consists of a sample counter that has twice the capacity of the integrating counters, and additional circuitry to monitor the overload flag. Two modes of operation are necessary: peak detection, and function display.

In peak detection mode, the objective is to locate the first integrating counter to overload. It operates as follows:

Correlation commences with a reset pulse which clears the overload latches and presets the integrating counters to their start value. This sets their capacity to N . After at least N input sample pairs, the overload flag signals the arrival of the first overload, which represents the most significant peak of the correlation function. The contents of the sample counter m are used to compute the significance, or the ordinate, of the detected correlation peak, using Equation 2.38. The time lag, or the abscissa of correlation peak, is then calculated by transferring the contents of the latches to the overload shift register, and counting the leading zeros in the pattern as it is shifted out. The system is then reset and correlation begins once more. Successive overload patterns may be viewed as a pulse train whose frequency is inversely proportional to the time lag at peak correlation. In other words, the frequency is proportional to the flow velocity (in a correlation based flow meter for example).

Display mode operates similarly to peak detection mode except that the system is not reset after the occurrence of the first overload. Instead, at suitable intervals of correlation significance (that is, at regular intervals along the vertical axis of the correlation function), the contents of the OSR are shifted out and displayed, as discussed in Section 3.2.2.2 and in Section

3.3.3.

The initial design criteria for the basic correlator are as follows:

Cascadability: the number of correlation points, or the resolution of the time delay axis, is determined by the number of cascaded correlation stages. In the prototype design, there is no limit to the number of stages that may be cascaded.

Long, programmable integration time: in correlation applications where the time-bandwidth product (TB) is low, as discussed in Chapter 2, long integration times are required. The integrating counters in the Eu349 prototype chip have a capacity of approximately 2^{15} states. To maintain flexibility, and allow the correlator to address both high and low TB applications, the integrating counters must be programmable.

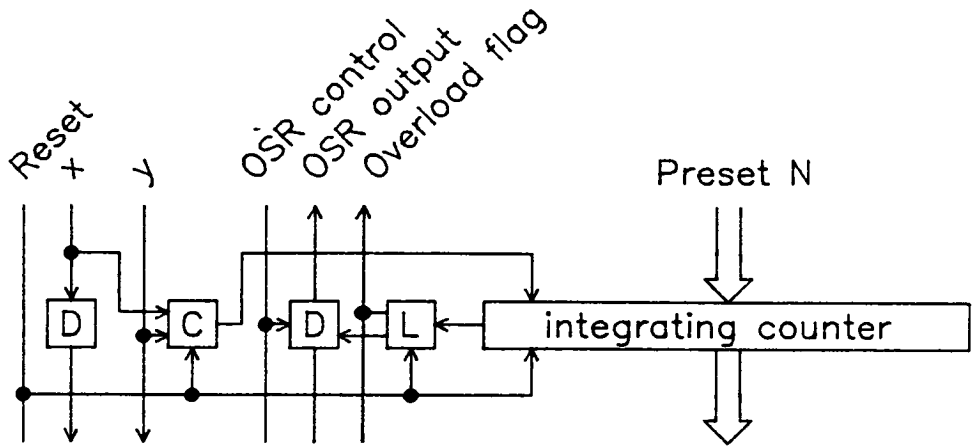
Design style, static or dynamic: there are two important considerations here. First, the integrating counters are required to count at a rate determined by the number of coincidences in the data. In other words, the count rate is proportional to the correlation of the input bit streams, and therefore may vary from zero to the sample rate. Consequently the design of the integrating counters must be static, regardless of the factors that determine the sampling rate. Second, the shift registers, OSR and DSR, may be static or dynamic depending on the sampling rate requirements. However, to allow flexibility in choice of sampling rate, these shift registers must be static. Using static registers for the OSR and DSR functions, incurs a small area penalty of approximately 2%.

Note that the architecture in Figure 5.1 differs from that in Figure 2.14, in that the output of the integrating

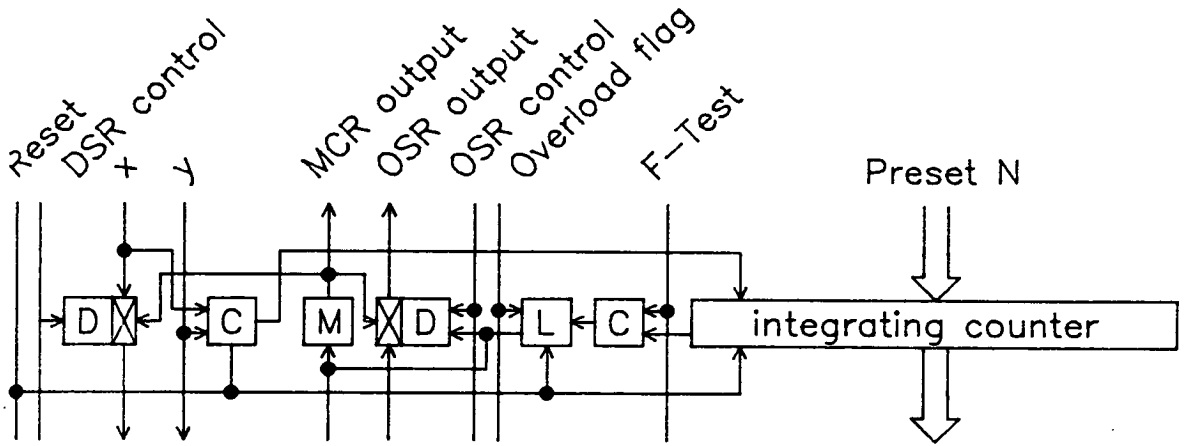
counter comes from the left of the counter rather than the right. This detail allows the DSR, the coincidence gates, the OSR, and associated latches to be laid out close to each other on the silicon. The significance of this facet of the architecture will emerge in the next section, when the modifications to the basic correlator architecture to accommodate the self test and self repair philosophy are described.

5.3. Architecture of the Correlator with Built-In Self Test and Self Repair Features

The VLSI architecture considered here, consists of a long series connection of identical correlation stages. If any stage suffers faults during manufacture, or becomes faulty during service, the whole chip will fail. A self test and self repair strategy has been devised to overcome this problem. The self test sequence is started each time the chip is switched on; any faulty stages discovered as a result of the test are automatically bypassed. This reconfigures the working stages into a continuous serial connection. Thus, faults that develop during the working life of the chip are automatically eliminated every time the chip is switched on. Modifications to the basic correlator architecture, to accommodate the self test and self repair philosophy, are shown in Figure 5.2. Figure 5.2(a) shows the basic correlator stage; Figure 5.2(b) shows the basic stage modified to perform built-in self-test and self-repair. A block diagram showing 8 stages of the array is presented in Figure 5.3. The architecture is well structured and thus maps easily on to silicon.



(a)



(b)

KEY: D = Delay M = Multiplexer control
 C = Coincidence gate
 L = Latch X = Multiplexer

Figure 5.2(a). Basic correlator stage; (b). Correlator stage with built-in self-test and self-repair circuitry.

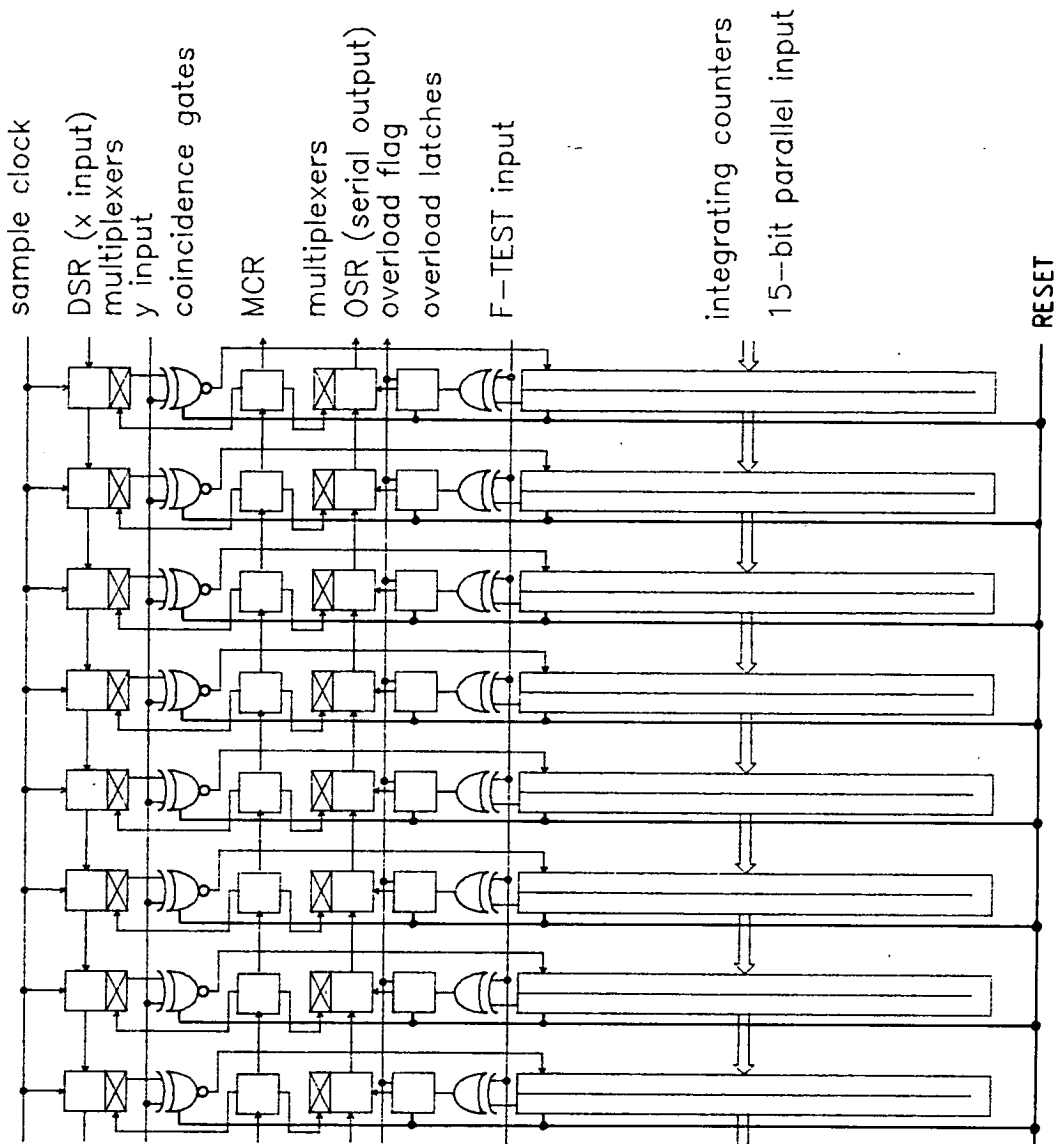


Figure 5.3. Block diagram of the correlator array, showing 8 of the repeated stages.

The principal additions for self-test are the input signal "F-test" (for function test), and its associated anticoincidence detector (EXOR) at the "set" input of each latch. Also, there is a parallel set and reset facility in the data shift register. All other circuitry required by the test strategy already exists as part of the basic

correlator. The principal additions for self repair are the multiplexer control register, or MCR, and 2:1 multiplexers on the data shift register and overload shift register outputs.

In test mode, the DSR, MCR, and OSR shift registers act as scan paths, and signature analysis is performed by the integrating counters. The result of the signature analysis is compared with the known good signature using the F-test input. The results are latched for subsequent use in the self repair scheme. (The test sequence is described in detail in Section 5.5.) Testability is achieved using functional conversion to such an extent, that the silicon area overhead is only 2%. This is illustrated by Figure 5.4, which shows the floor plan of one correlation stage in the Eu349 chip.

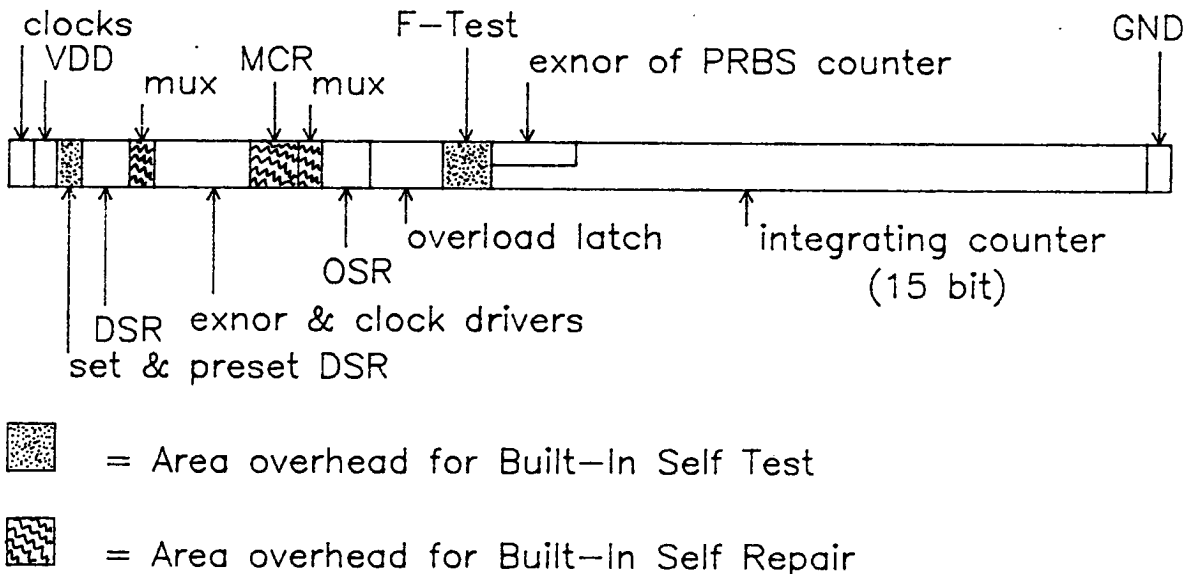


Figure 5.4. Floor plan of one correlation stage in the Eu349 chip, illustrating the relative areas of the components in the design.

The self-repair technique requires that the data

shift register, and the overload pattern shift register, each have a 2:1 multiplexer connected to their outputs. The technique requires a multiplexer control register for storing the control information for these multiplexers. The multiplexer control register is the key feature in the self repair scheme. After the self-test sequence, the MCR contains the pass/fail status for each stage. In the case of a failure, the input and output registers of the correlator stage are bypassed via the multiplexers, so that the malfunctioning stage is short-circuited. The number of functioning stages on the chip can be read out serially from the MCR by reconfiguring it as a shift register. This parameter represents the maximum attainable correlation delay (or resolution) and can be used for chip reject/accept decisions in production test. The self-test and repair sequence may be repeated as required during the service life of the chip.

The layout of the MCR and its associated multiplexers, is simplified by manipulating the architecture of the correlator stage so that the DSR and OSR are laid out close to each other topographically, as discussed in the previous section. As a result, the overhead for self-repair is not greater than 6% additional silicon area. The self repair features are shown cross-hatched in Figure 5.4.

5.4. Design of the Eu349 Correlator

5.4.1. System Overview

A prototype digital correlator featuring self test and self repair has been fabricated on a six-micron N-channel MOS process. The prototype design, shown in Figure 5.5, contains 28 parallel stages of correlation, each of which implements the block diagram in Figure 5.2(b).

The area of the chip is 5.08 mm by 5.08 mm, and the chip contains approximately 7500 transistors.

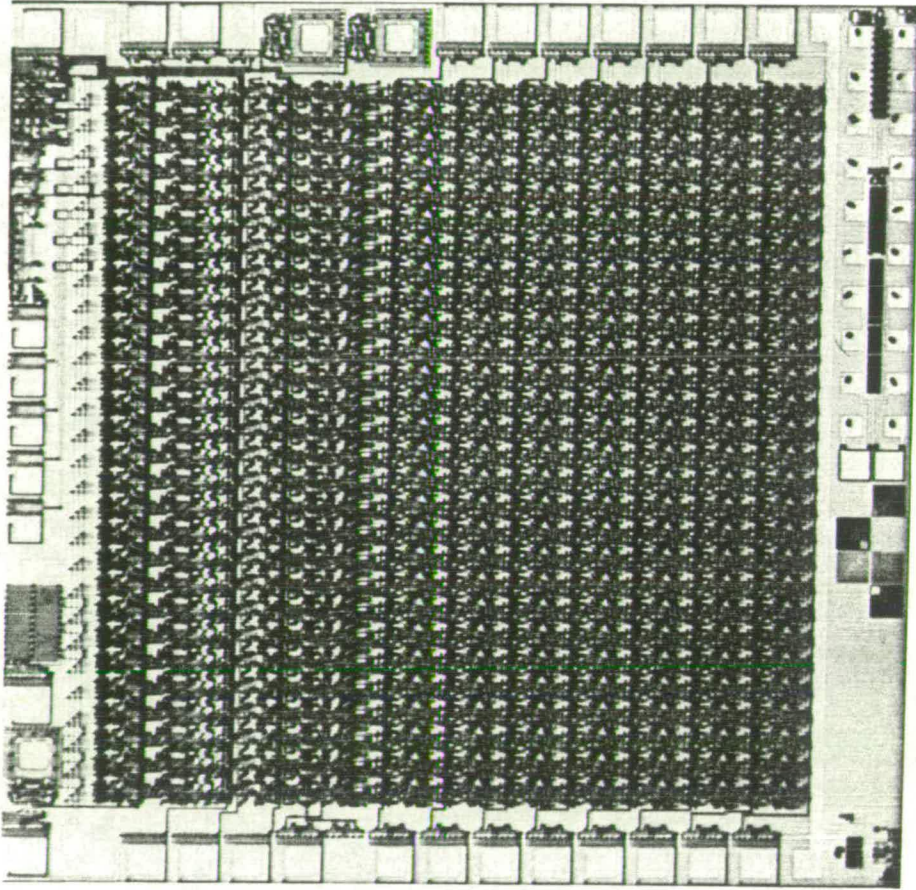


Figure 5.5. Eu349 correlator chip.

A floor plan of the chip is shown in Figure 5.6. This figure identifies the main features of the chip layout, and shows the relatively large, dense area occupied by the integrating counters. The integrating counters, which are made from linear feedback shift registers, are realised in the shape of a ring to minimise the length of the feedback connection. Thus, the propagation delays between each shift register stage are approximately equal.

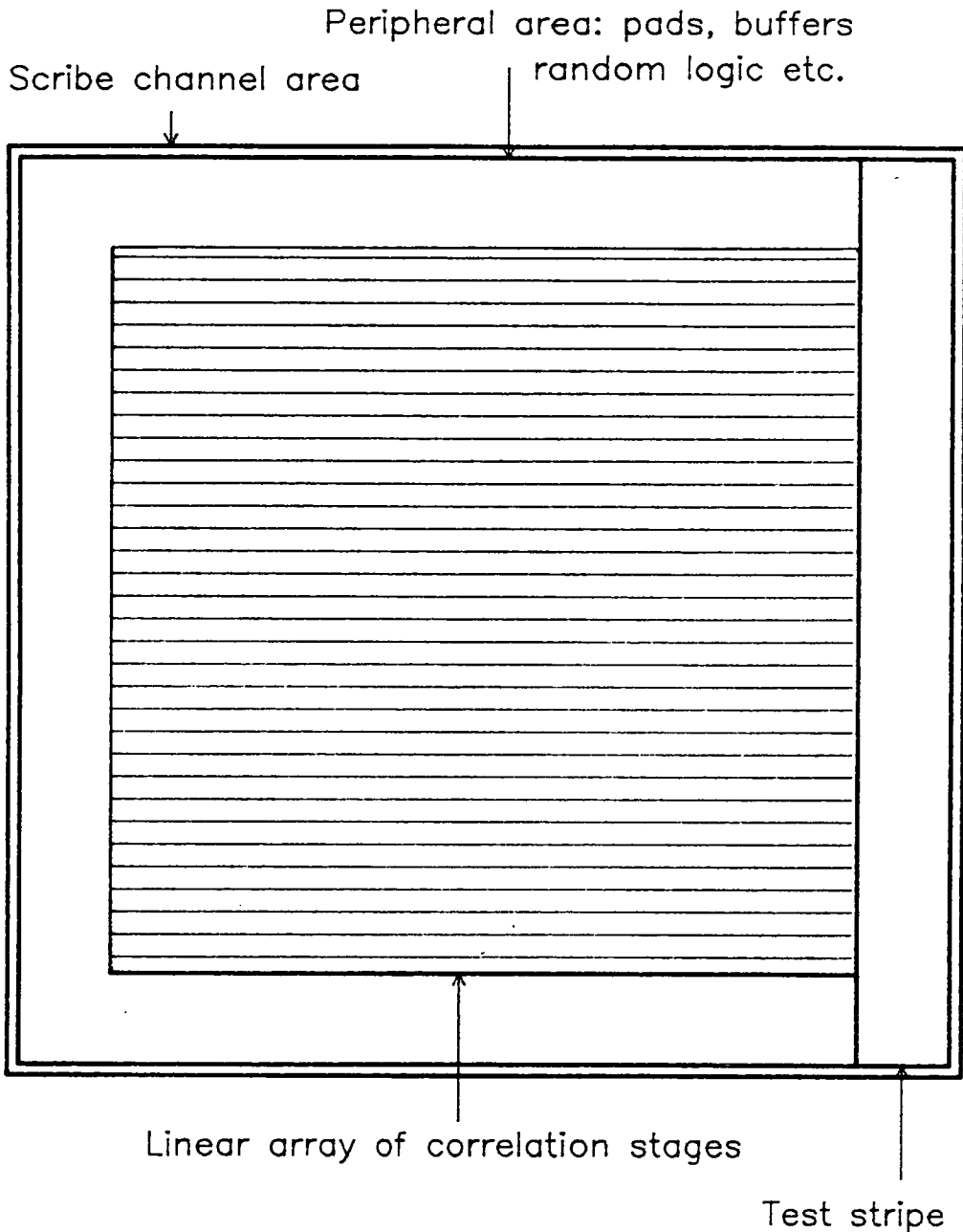


Figure 5.6. Eu349 correlator chip floor plan.

The correlator design uses a two phase non-overlapping clock system. The phases are denoted ϕ_1 and ϕ_2 , respectively, and in general, data is sampled on ϕ_1 , and stored on ϕ_2 . The design is semistatic through out. This means that during one clock phase (in this case ϕ_2)

the stored state can be maintained indefinitely. Thus, the clock frequency, and therefore the sampling frequency of the correlator can range from dc to 4 MHz (for this fabrication process).

The prototype devices have also been packaged in 40 pin dual-in-line ceramic packages. The pin designations of the Eu349 device is shown in Figure 5.7.

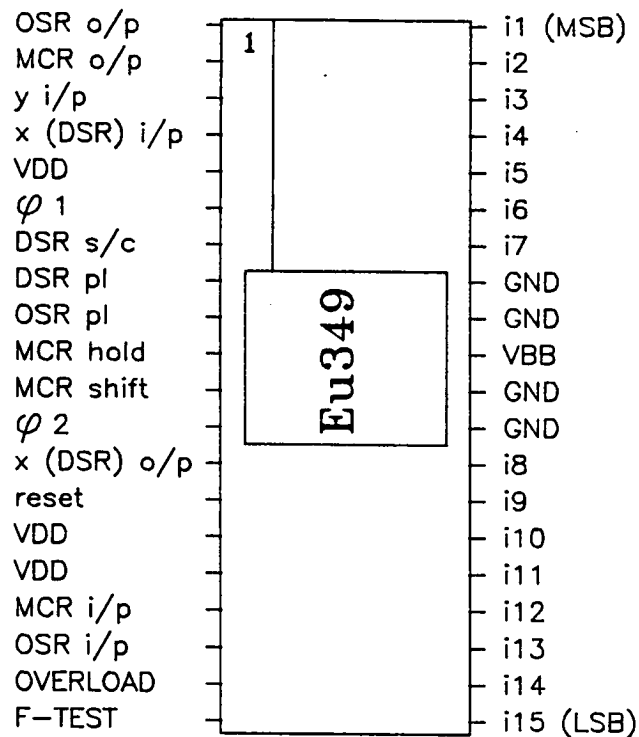


Figure 5.7. Pin designation for packaged Eu349 correlator chips.

A summary of the functional description of each pin is given in Table 5.1

TABLE 5.1 PIN-OUT FUNCTIONAL DESCRIPTION		
FUNCTION		PIN NUMBER
Inputs	x (DSR) i/p	4
	y i/p	3
	MCR i/p	17
	OSR i/p	18
	F-TEST	20
	i1 - i15	21-28, 34-40
Controls	DSR s/c	7
	DSR pl	8
	OSR pl	9
	MCR hold	10
	MCR shift	11
	Reset	14
Outputs	OSR o/p	1
	MCR o/p	2
	x (DSR) o/p	13
	OVERLOAD	19
Clocks	$\Phi 1$	6
	$\Phi 2$	12
Supplies	VDD	5, 15, 16
	VSS	29, 30, 32, 33
	VBB	31

Truth tables for the operation of the control signals to the overload shift register, data shift register, and multiplex control register are listed in Tables 5.2 to 5.4

respectively.

TABLE 5.2 OSR TRUTH TABLE	
OSR pl	Effect
L	Serial shift from input pin
H	Parallel load from latches

TABLE 5.3 DSR TRUTH TABLE		
DSR pl	DSR s/c	Effect
L	L	Serial shift from input pin
L	H	Serial shift from input pin
H	L	Parallel load zeros (CLEAR)
H	H	Parallel load ones (SET)

TABLE 5.4 MCR TRUTH TABLE		
MCR shift	MCR hold	Effect
L	L	Parallel load MCR from latches
L	H	Hold MCR contents stationary
H	L	Serial shift MCR
H	H	Serial shift MCR

The chip design is divisible into two main areas: correlator array circuitry, and peripheral circuitry. Each area can be subdivided further into its component nMOS modules. The nMOS modules are listed in Appendix 1.

5.4.2. Correlator Array Design

The correlator array is composed of 28 identical stages of correlation. This number is arbitrary, but is determined by the amount of available silicon area. At the time of design, the maximum available chip size measured 5.08 mm by 5.08 mm. of which a border 0.5 mm wide is required for mandatory test structures and peripheral circuitry. Thus an area of approximately 4 mm by 4 mm is available for the layout of the correlator array; enough silicon for 28 stages.

Each correlator stage, designated Module STG100, implements the circuit shown in Figure 5.8. The floor plan and layout of module STG100 is shown in Figure 5.9.

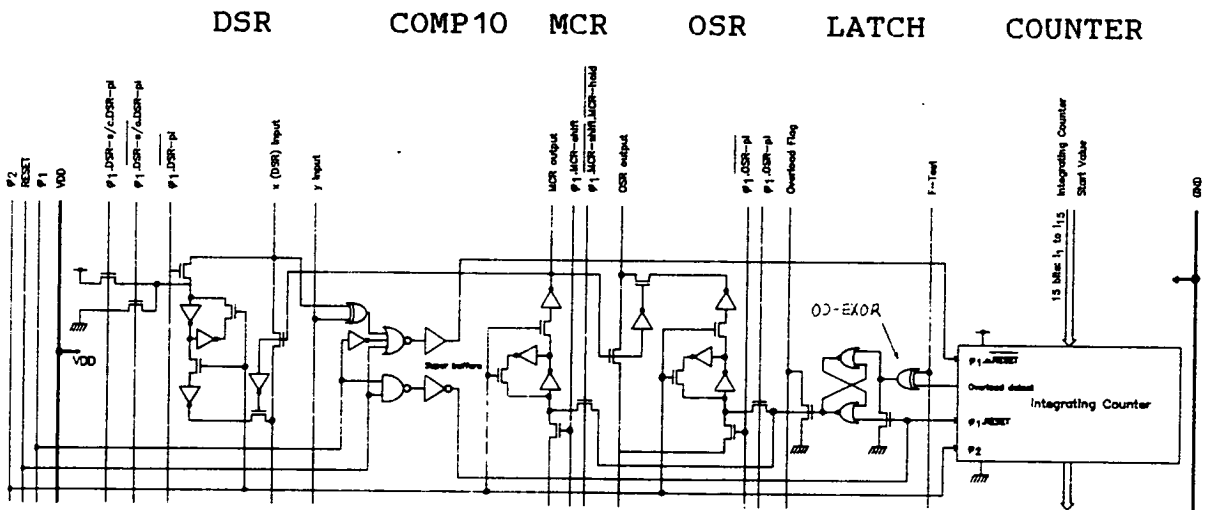


Figure 5.8. Circuit schematic of the repeated correlator stage in the Eu349 device.

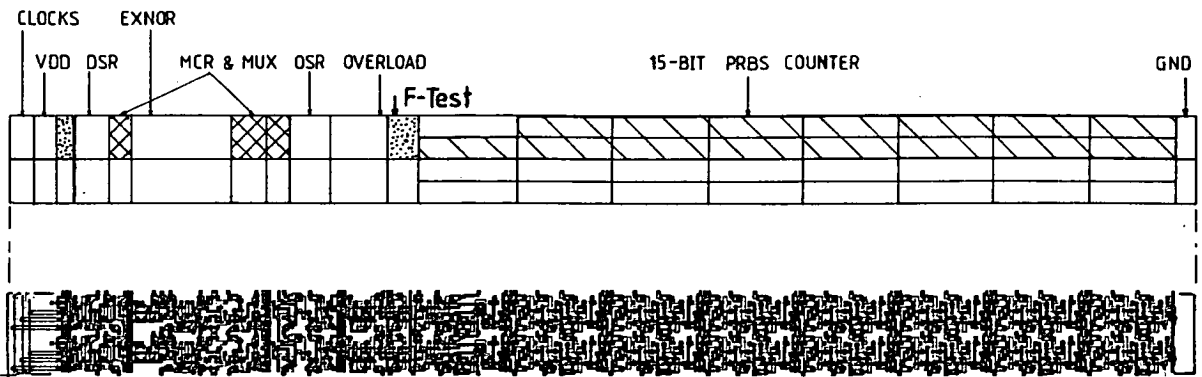


Figure 5.9. Floor plan and nMOS layout of Module STG100. This represents the repeated correlator stage.

The interstage connections are either bit serial, nearest neighbour communications, or globally broadcast control signals. This allows the stages to form a serial cascade by simple abutment in the y direction. All connections to and from the correlator array are made via the peripheral circuitry. Connections to the outside world are made through pads in the peripheral area, where input protection and buffering takes place. Some control signals to the array are generated in the peripheral area, and therefore the inputs to the chip, as shown in Figures 5.6 and 5.7, differ from the inputs to the correlator array, as shown in Figure 5.8. The circuitry of the peripheral area is summarised in the next section.

Referring to Figure 5.8, the data shift register, DSR, consists of three inverters and five pass transistors. The three pass transistors that form the input to the shift register select the required input source. In one instance the source is the x data input (the x data output from the previous correlation stage), and in the other cases the input sources are VDD and GND, so that the register may be set or cleared respectively. These data

selectors only operate during $\phi 1$ (that is, while phase 1 of the clock is high). Thus, data is transferred from the selected source to be stored, dynamically, on the gate of inverter Inv1 under the control of $\phi 1$. Static storage is implemented during $\phi 2$. Inverter Inv2 provides a feedback loop that regenerates the stored state so that it may be stored indefinitely (so long as $\phi 2$ remains high to enable the feedback loop). The stored information is also transferred to the register output under the control of $\phi 2$. The same basic semistatic shift register circuitry can be found in the DSR, OSR, MCR, and the shift register stages that make up the integrating counter.

The integrating counter is a 15 element PRBS counter. The feedback is the logical exclusive NOR of the 14th and 15th output. Thus the count length is 2^{15} less one forbidden state where all 15 registers contain logical ones. All of the other possible combinations are legal, and one such combination may be used to indicate that the counter has reached capacity. The simplest combination to detect using nMOS circuitry is the all zero state which requires a 15-input NOR gate. To detect the all ones state, which would be required if exclusive OR feedback were used, a 15 input NAND gate is required. In nMOS, it is desirable to construct NOR gates in preference to NAND gates; therefore exclusive NOR feedback has been implemented.

The 15-bit shift register in the integrating counter is laid out on silicon in the form of a ring. The benefit of doing this is that the length of the feedback connection is minimised, and thus the delays between stages of the shift register are approximately equal.

The integration time, or counter capacity may be programmed by presetting the combination of ones and zeros that represents the starting point for the count sequence.

The counter then counts from this starting sequence and produces an "overload detect", or OD, pulse when the all zeros state is reached. The combination of ones and zeros, or start word, that represents a particular integration time is derived by simulating the action of the integrating counter *in reverse*. The simulation program takes as its input the required integration time in clock cycles; it then steps back through the PRBS sequence from the all zero state for the specified number of clock cycles and prints the combination of ones and zeros at that point in the sequence. Table 5.5 lists some integration times and corresponding integrating counter start words.

TABLE 5.5		
Integrating Counter Start Words		
Integration time (clock cycles)	Counter Start Code (binary)	(hex)
5	000000000010101	0015
10	000001010101010	02AA
15	101010101010101	5555
32	011001100110001	3331
64	001011010011100	169C
128	010010001110001	2471
256	011100011100010	38E2
512	001110110001011	1D8B
1024	010011101001111	274F
2048	011000011111110	30FE
4096	001111000000011	1E03
8192	000011111110000	07F0
16384	000000011111111	00FF
32766	100000000000000	4000

Referring again to Figure 5.8, the operation of the circuit is as follows. The x data and the y data inputs are compared by the comparator module, designated COMP10. This module has two functions: first, to produce the count pulse for the integrating counter, and second, to synchronise the RESET pulse with ϕ_1 . Producing the count pulse for the integrating counter, requires the module to perform the logical exclusive NOR of the x and y data, to perform the necessary logic so that the count pulse is disabled during a RESET, to synchronise the count pulse with ϕ_1 , and to provide adequate buffering for the count and RESET pulse.

The purpose of the RESET pulse is to clear the overload latch and load a preset start word into the integrating counter. The input to each shift register stage in the integrating counter can come from one of two sources: the preceding shift stage, or the preset inputs i_1 to i_{15} . The selection is controlled by either a count pulse or a RESET pulse and the operation must be mutually exclusive. To prevent both events occurring at the same time, the RESET signal disables the generation of a count pulse. There is similar reasoning behind the design of the other shift register control signals. The input select controls for the DSR are designed to mutually exclusive, as are the controls for the MCR and OSR respectively.

After the RESET pulse has cleared the overload latch, and preset the integrating counter, the sampled data is shifted along the DSR. When the x and y inputs are equal (at the correlator stage under discussion) a count pulse is generated which increments the integrating counter. Eventually, as this operation continues, the counter reaches the overload state, i.e. all zeros, and produces an overload detect pulse. In normal circuit operation, the F-test signal is held low, and the overload detect

pulse sets the overload latch. This in turn causes the output signal "overload flag" to change state (high to low), which indicates that a correlation peak has been detected. The overload flag is a wired-OR output, so that the Eu349 devices may be arbitrarily cascaded.

Under the control of OSR-pl (parallel load control for the OSR), the values stored by the overload latches are transferred, in parallel, to the overload shift register OSR. Then, again under the control of the OSR-pl, the overload pattern is shifted off chip via the OSR serial output.

In the above discussion, it has been assumed that the MCR contains the necessary bit pattern to configure the cascade of correlator stages into a continuous serial connection of correctly functioning stages. The method by which this is performed is described in Section 5.5, but the circuitry used to perform self repair is discussed here.

The MCR is similar in design to the OSR and DSR. The controls to the MCR allow it operate in three modes: serial shift, parallel load, and hold. The output of MCR controls the multiplexers at the outputs of the DSR and OSR. When a logic one is stored in MCR element of a particular correlator stage the DSR and OSR inputs are short circuited to their respective outputs. When this is done, the affected correlation stage serves only to link together its two immediate neighbours. The overall effect, therefore, is that correlation stages may be selected and eliminated from the correlation array by inserting logic ones into the relevant bit positions in the MCR. The built-in self test and self repair procedure is a method whereby faulty stages may be identified and eliminated automatically.

During self test, a correctly functioning correlator stage produces two overload detect pulses. The overload detect output is continuously compared with the expected good output using the F-test input and the exclusive OR gate, denoted OD-EXOR in Figure 5.8. Any deviation from the expected good output sets the overload latch. Thus, faulty correlation stages have their overload latches set during the self test period. The number of faulty stages in a cascade, may be determined by transferring the contents of the overload latches to the OSR and shifting the pattern off chip. The number of ones in this pattern represents the number of faulty stages in the cascade. Self repair is carried out by transferring the contents of the overload latches to the MCR. This is done using the MCR-hold and MCR-shift controls in combination (both controls low).

The multiplexers associated with the DSR and OSR outputs each consist of one inverter and two pass transistors. In normal circuit operation, the "bypass" transistor is turned off, and the "output" transistor is turned on (see Figure 5.8). When a correlator stage is identified as being faulty the bypass transistor is turned on, and the output transistor is turned off. Thus, the input data to a subsequent correlator stage will have passed through n bypass transistors and one output transistor, when n preceding, contiguous stages have been identified as faulty. If n is greater than three, then the operation of the DSR (or OSR) is degraded due to the excessive delay introduced by the series connection of output and bypass transistors. (The delay in a series connection of four pass transistors is approximately equal to the delay in one inverter.) This system, therefore cannot guarantee to repair more than three consecutive faults. However, the probability that more than three consecutive faults will occur is very low, and can be estimated to be less than

10^{-5} for an overall yield of 20%.

5.4.3. Peripheral Circuit Design

The peripheral circuitry consists of input and output pads, power supply pads, buffer circuits, and some random logic for generating or synchronising control signals. The peripheral circuitry is described in Appendix 1.

5.5. Test Strategy

The test strategy for the correlator consists of built-in self test and self repair procedures. These procedures are *off-line*, therefore, they are distinct from, and do not impede the normal operation of the correlator in "run" mode. The test strategy is divisible into three parts which are summarised here. A detailed step by step test schedule is listed in Appendix 2. The three parts of the test strategy are: initial test, self test, and self repair.

5.5.1. Initial Test

During the initial test period three tests are carried out on the critical elements of the design, namely the scan path registers. These registers (DSR, MCR and OSR) and their various control functions are not covered by the self test and repair strategy, and therefore must be tested to check that the subsequent self test and repair procedures are possible. The initial test sequence is as follows:

- a) Test MCR, OSR and DSR as shift registers and measure their delay. This is done using a flush test, as described in Section 4.3. The MCR must be flushed with zeros and held static while the DSR and OSR are tested.

- b) Test the effect of the MCR on the DSR and OSR registers. This is done by shifting n "ones" into the MCR and then measuring the delay of the DSR and OSR registers, which should each be reduced by n .
- c) Test the parallel load facilities of the DSR, OSR and MCR registers, and the set and reset facilities of the overload latches.

5.5.2. Self test

In the self-test period, a full functional test of the correlation array takes place. In this test sequence (b) is repeated four times according to the possible combinations of the two binary input signals, x and y . Initially the MCR must be flushed with all zeros and held static.

- a) Reset latches and integrating counters. The counters are loaded with 4000 (in hexadecimal), a number that corresponds to the maximum integration time of $2^{15}-2 = 32766$ sample clock cycles, as described in Section 5.4.2.
- b) Set up the input conditions x and y and set or clear the DSR register as required. Shift x and y through correlator for 32766 clock cycles. When the inputs are equal, F-test must be set HIGH to coincide with the expected overload detect pulse.
- c) Parallel load latches into OSR. The overload pattern may be shifted out for observation.

5.5.3. Self Repair

The self repair sequence follows the self test sequence. During the self test sequence the overload signal is compared with the expected value of overload signal. Any deviations from the expected signal results in a logic 1 stored in the corresponding latch. Thus, when the self test sequence has finished the logic 1's and 0's stored in the latches are the results of the self test, where a logic 1 indicates a faulty stage. The self repair operation transfers this information to the MCR which in turn causes the faulty stages to be bypassed. The net effect is a series connection of correctly operating correlation stages. The following sequence is required.

- a) Parallel load MCR.
- b) Hold MCR static.

5.5.4. Run

The run period follows automatically after the self-test and repair sequence is completed. After the test period the number of zeros stored in the MCR represents the number of correctly operating correlation stages. The following sequence may occur during the run period.

- a) Monitor the overload flag status, and/or display output from the OSR.
- b) Compute ordinate and abscissa of correlation peak.
- c) Reset, and repeat correlation.

5.6. Test System Configuration and Results

5.6.1. Test Configuration

The test equipment used to carry out the functional test comprises a Tektronix Digital Analysis System (DAS 9100), one dual power supply unit, and a purpose-built test-jig.

The test-jig incorporates power supply decoupling, one external load resistor for the wired-OR overload output from the correlator chip, and a 40 pin dual-in-line (DIL), zero insertion force IC socket. The test-jig provides an interface between the DAS and the device under test (DUT), which is either a packaged chip or a probed chip on a wafer. In both cases electrical connection is made via the 40 pin DIL socket.

Initially 10 packaged chips, which had passed a visual inspection, were functionally tested. However, many more samples were required to demonstrate the yield enhancement capability of this design, so the remaining wafers were probe-tested. The Eu349 chip was fabricated as part of a multi-project wafer, with only 24 chip sites per wafer. Consequently only 130 candidates were available for testing.

5.6.2. Test Results

The results are divisible into two parts. These are chip verification results, and yield enhancement results. Chip verification consists of initial test sequence results, and self test and repair sequence results. These results are demonstrated here using display material from the Tektronix DAS. Yield enhancement results are discussed in Section 5.6.3.

The initial test sequence is shown in Figure 5.10. These tests are equivalent to those described in Appendix 2, Sections A2.2 to A2.7, but are abbreviated and linked together to form a continuous display. These abbreviated elements of the initial test sequence, and part of the self test sequence, are small enough to fit into a single DAS pattern generator program, and the resulting data sequences are short enough to fit into the DAS acquisition memory. This short hand method allows a large number of devices to be checked easily. Chips that pass this test can then be given a more exhaustive test according to Appendix 2.

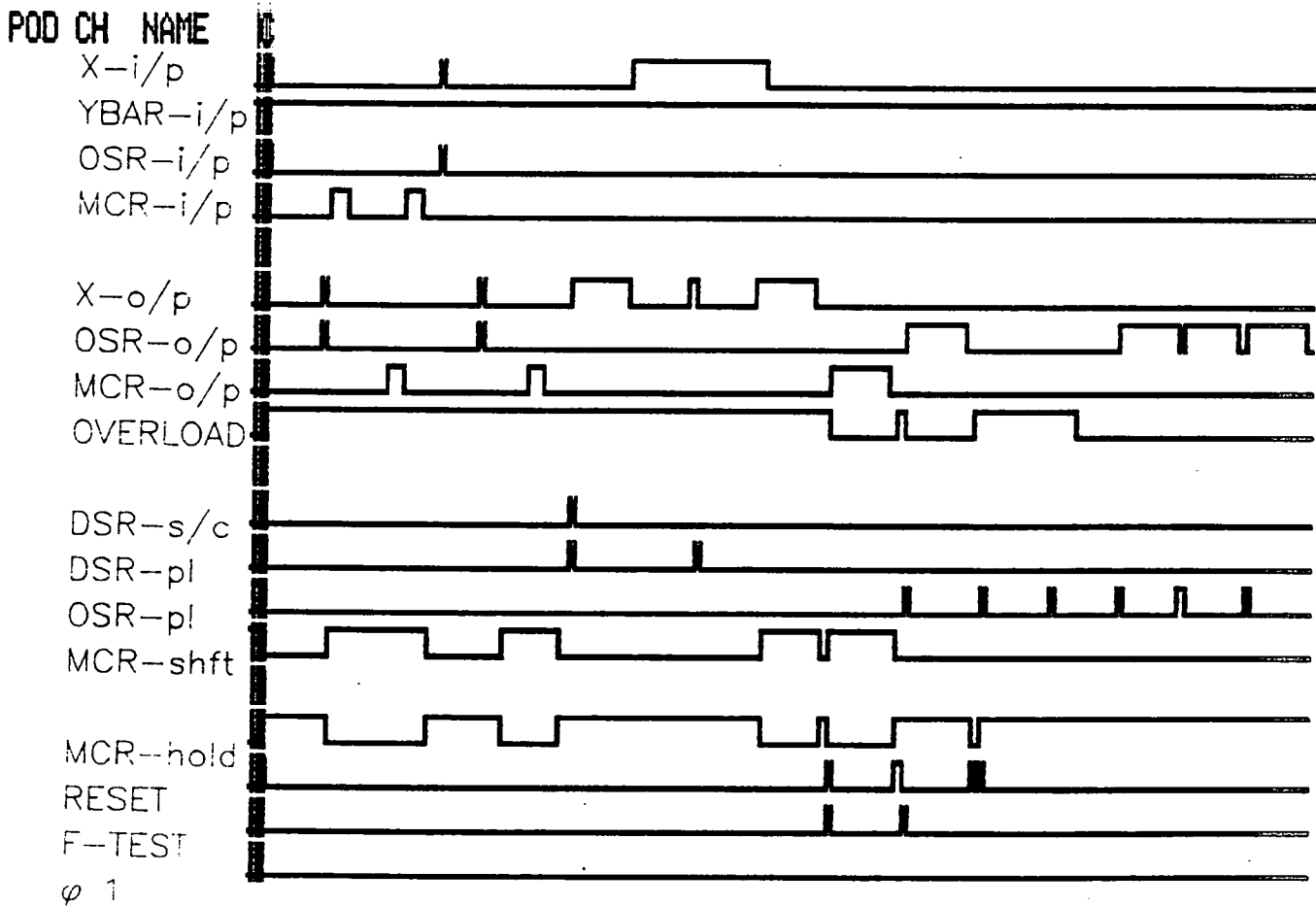


Figure 5.10. Results for initial test sequence.

The left hand side of Figure 5.10 is shown expanded

in Figure 5.11. The figure shows 16 traces. The top four traces show the inputs to the device under test, and the group of four traces below these represent the outputs. The remaining eight traces are the chip control signals. This figure shows tests that verify the function of the MCR, DSR and the OSR.

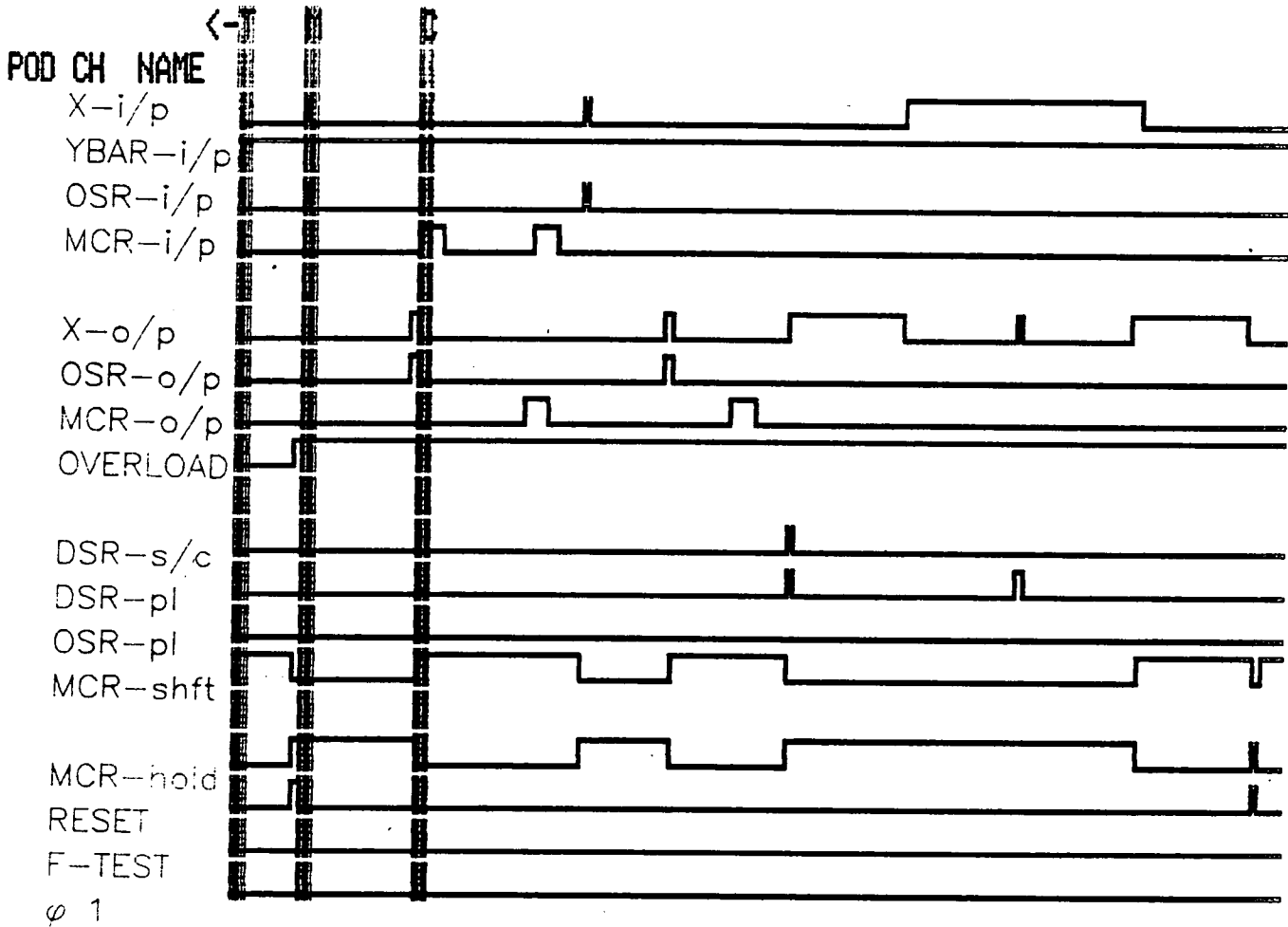


Figure 5.11. Tests to verify the function of the MCR, DSR, and the OSR.

In Figure 5.11 there are three dense vertical lines labelled "T", "M", and "C", for "trigger", "marker", and "cursor" respectively. The sequence of events before the marker are concerned with flushing zeros through the MCR, DSR, and OSR. At the marker, the MCR controls indicate that the contents of the MCR are being held static, that

is, the MCR is neutralised. Also at the marker, single logic ones are presented to the x and the OSR inputs. After 28 clock cycles these logic ones, against a background of zeros, have shifted through the registers and appear at the x and OSR outputs. The point at which they appear is marked by the cursor. The time delay between the marker and the cursor is given in the top left of the figure in a line starting "C - M", and is shown to be 28 μ s. This shows a sample of test A2.3.

Starting at the cursor position, and moving to the right, a similar test is shown with logic ones being shifted through the MCR. Although it is not shown explicitly, the delay through the MCR is also 28 μ s. This shows a sample of test A2.2. The next sequence tests the effect of the MCR on the other shift registers (test A2.4). The sequence starts where the MCR input goes HIGH for the second time. This MCR input pattern represents a group of three consecutive logic ones which are shifted into the MCR and held static. Then a simple flush test is performed on the DSR and OSR, by shifting single logic ones into the DSR and OSR against a background of zeros. The resulting delay through these registers can be measured as before, and is shown here to be 25 μ s. This test sequence is completed by flushing the MCR with zeros to neutralise it for the next test. In doing this, the logic ones that had been held static in the MCR, can be seen emerging from the MCR output.

The next test sequence represents test A2.5, and is concerned with the SET and CLEAR features of the DSR. The sequence starts 28 clock cycles before the point where the control signals DSR-pl and DSR-s/c go HIGH. At the time when these signals go HIGH, a background of zeros have been shifted through the DSR. DSR-pl and DSR-s/c then cause the DSR to parallel load all ones, which can be

observed as a bank of 28 logic ones in the *x* serial output. After this bank of ones has shifted out, the *x* input is set HIGH, and a background of ones is established in the DSR. When the pattern reaches the output, the signal DSR-pl is pulsed HIGH again, this time with DSR-s/c LOW, and the DSR is cleared. The result of this action can be seen as a large gap before the final block of ones in the *x* output.

The second part of the initial test sequence is shown in Figure 5.12. This figure shows the waveforms relating to tests A2.6 and A2.7, where the MCR and OSR parallel load operations, and the overload latches set and clear operations are tested.

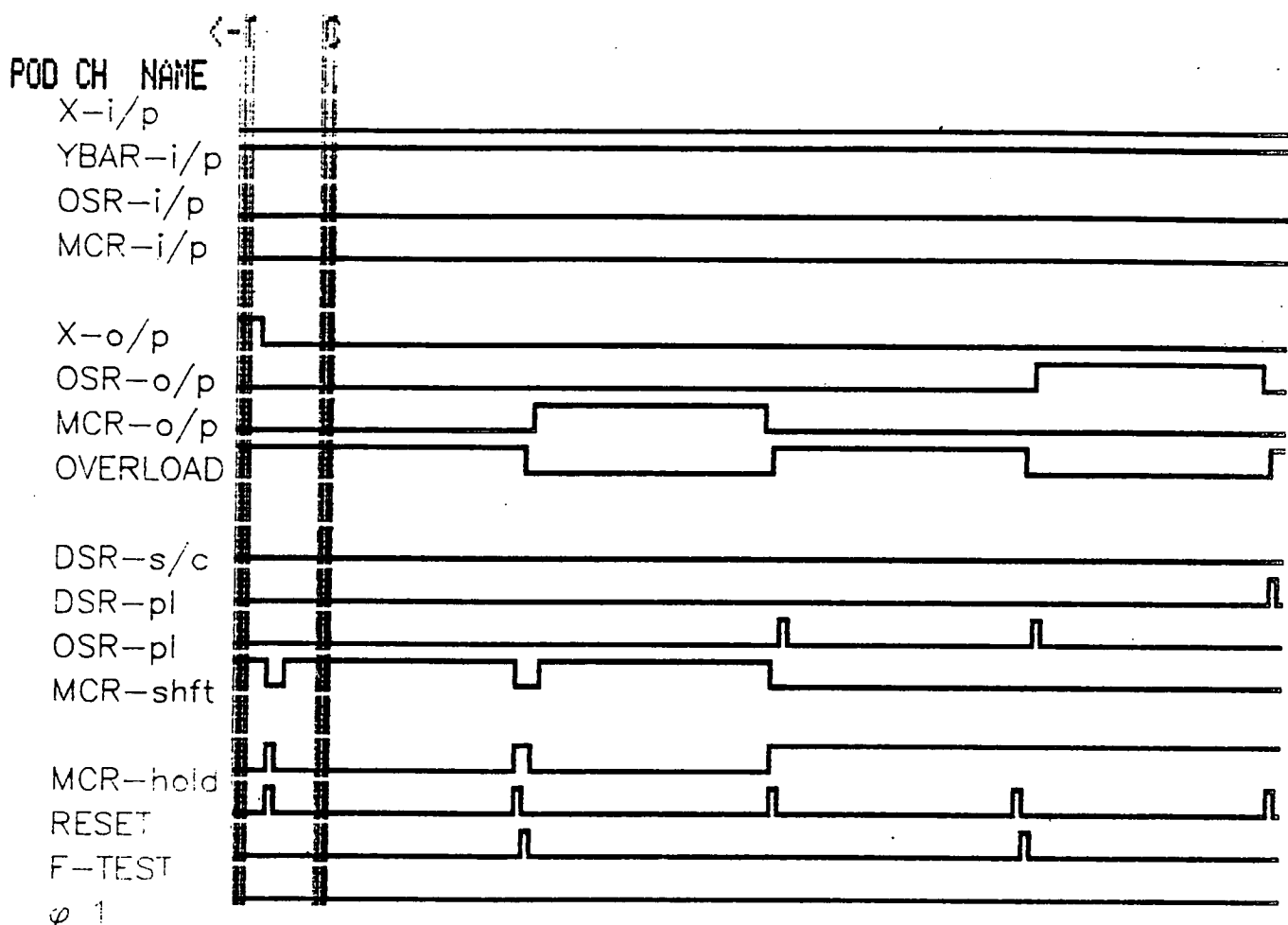


Figure 5.12. Initial test sequence relating to tests A2.6 and A2.7.

The sequence for test A2.6 begins with a RESET pulse to clear the overload latches. There immediately follows a control combination (MCR-shift LOW, MCR-hold LOW) which transfers the contents of the latches to the MCR. The MCR controls are then changed (MCR-shift HIGH) to shift the MCR contents out through the MCR output for observation. Since the latches were reset, the observed output should be all zeros, as can be seen in the MCR output in the region around the cursor in Figure 5.12. To complete the test, this sequence is repeated with one additional feature: the F-test pulse which immediately follows the RESET. In this respect the F-test signal works correctly,

and sets the overload latches. The contents are then transferred as before, and shifted out for observation. The bank of logic ones, as expected, can be seen on the MCR serial output.

Test sequence A2.7 is similar to A2.6 except that the OSR is tested instead of the MCR. Two pulses on the OSR-pl control indicate where the bank of zeros, and the bank of ones begin, respectively, on the OSR serial output.

Figure 5.13 and 5.14 show some of the input and output waveforms from two correlator chips, that have occurred during the self test and repair period. For display purposes the integration time of the correlator has been reduced to just 15 clock cycles. Figure 5.13 shows the correlation output of a "golden chip", that is, a fully functional chip, while Figure 5.14 shows the output of a chip that has one failed stage. The top four traces in each figure represent the inputs to the device. In each figure the x and y inputs sequence through their four possible combinations in accordance with the test strategy described in Section A2.8.

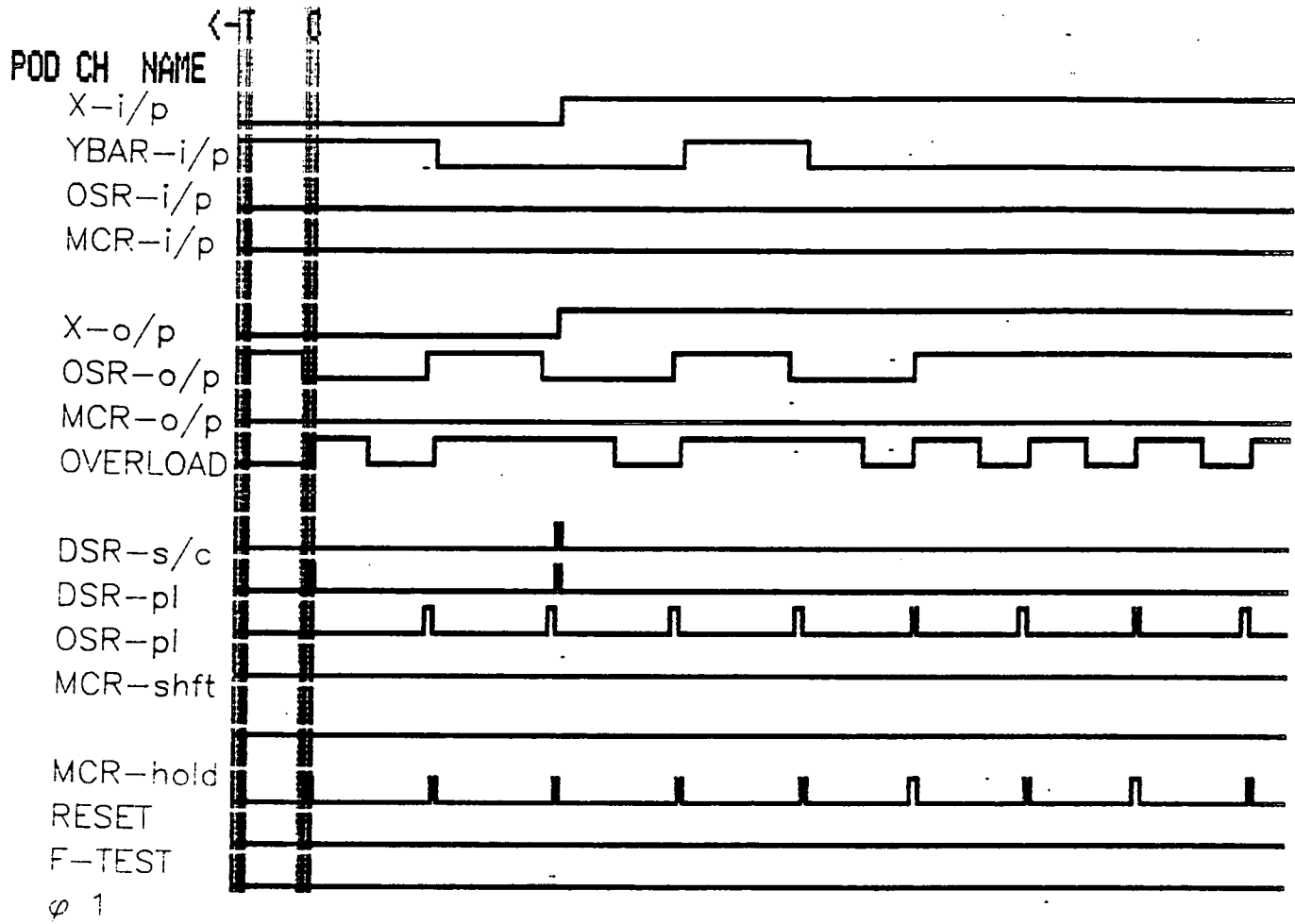


Figure 5.13. Self test sequence for fully functional, or "golden chip".

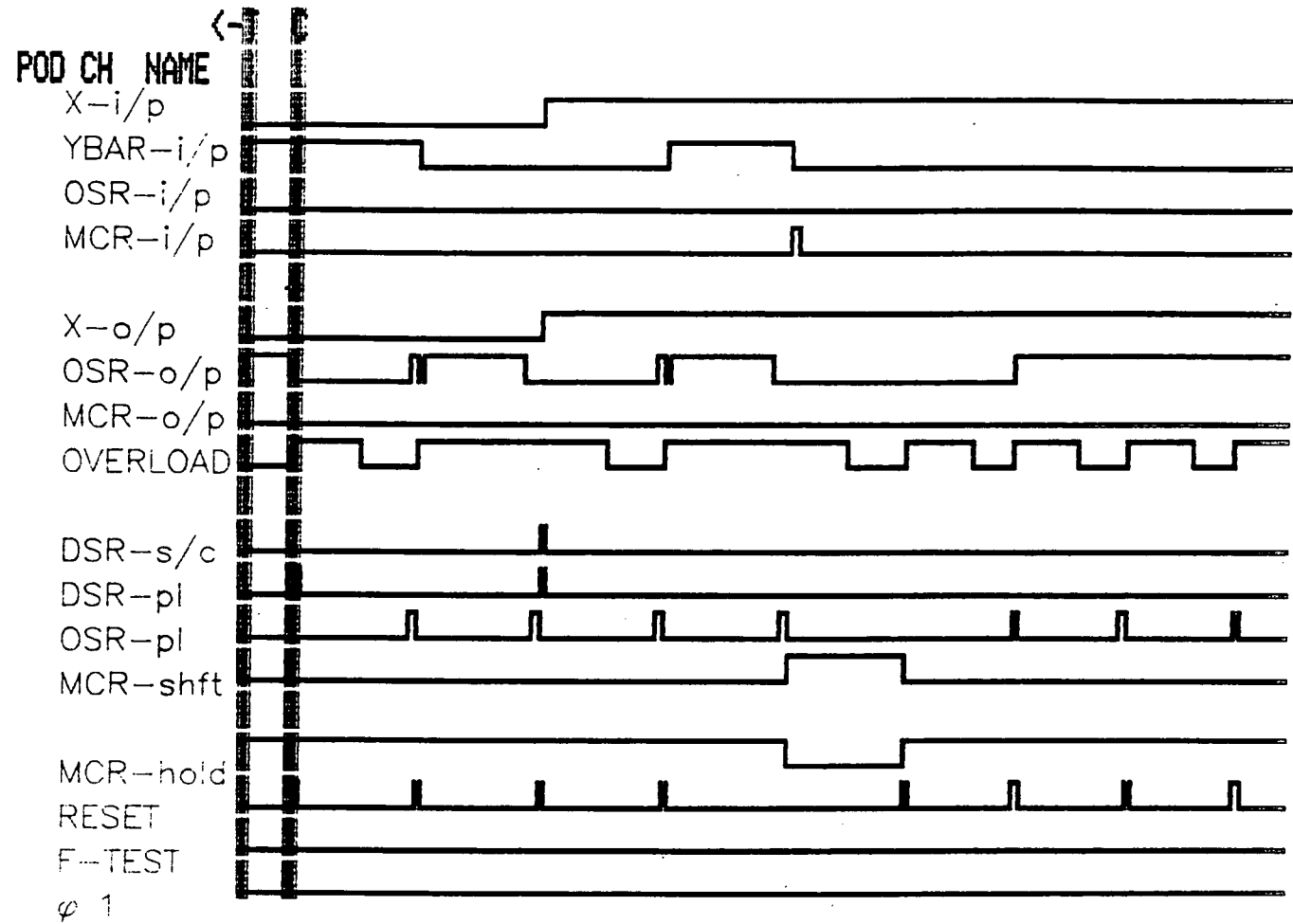


Figure 5.14. Self test and repair sequence for a chip with one faulty correlation stage.

The significant points to note in Figures 5.13 and 5.14 are the MCR input and the OSR output. All the other signals are the same for both chips, with the exception of the MCR control signals, MCR-hold and MCR-shift. With reference to Figures 5.13 and 5.14 and moving left to right from the cursor, the overload output (OVRFLO) has changed from logic 1 to 0. This indicates that at least one integrating counter has overloaded after the prescribed period of 15 clock cycles. This result is expected since the inputs have been equal, x and y both zero over this period.

When OVRFLO next goes high, the correlator has been reset and the next correlation test, with $x = 0$ and $y = 1$, is begun. Also at this time, the overload pattern, that is, the contents of the latches, are transferred to the OSR and shifted out for display. Now we can see the difference between the "golden chip", Figure 5.13 and the faulty chip, Figure 5.14. The OSR should contain a series of 28 logic ones and in Figure 5.14 there is a logic 0 in position number 2, indicating a fault in stage 2. The correlation test is repeated for the remaining combinations of x and y , and the fault is again exposed on the OSR output in the case where x and y are both equal to 1.

Self repair is then carried out on the faulty chip. A single logic 1 is shifted into bit position 2 of the MCR. This causes stage 2 to be bypassed. The correlation test, with x and y are both equal to 1, is repeated several times at a period of 27 rather than 28 and the incorrect logic 0 on the OSR output has been eliminated. The result is a "golden chip" containing 27 stages of correlation.

Figure 5.15 shows an expanded view of the repair sequence. The part of the figure labelled "A" represents the correlation overloads for the input combination $x = y = 1$. The overload pattern is displayed on the OSR serial output, and it should contain a continuous block of logic ones. However, with an apparent stuck-at-zero fault in stage 2 there is a zero at this position.

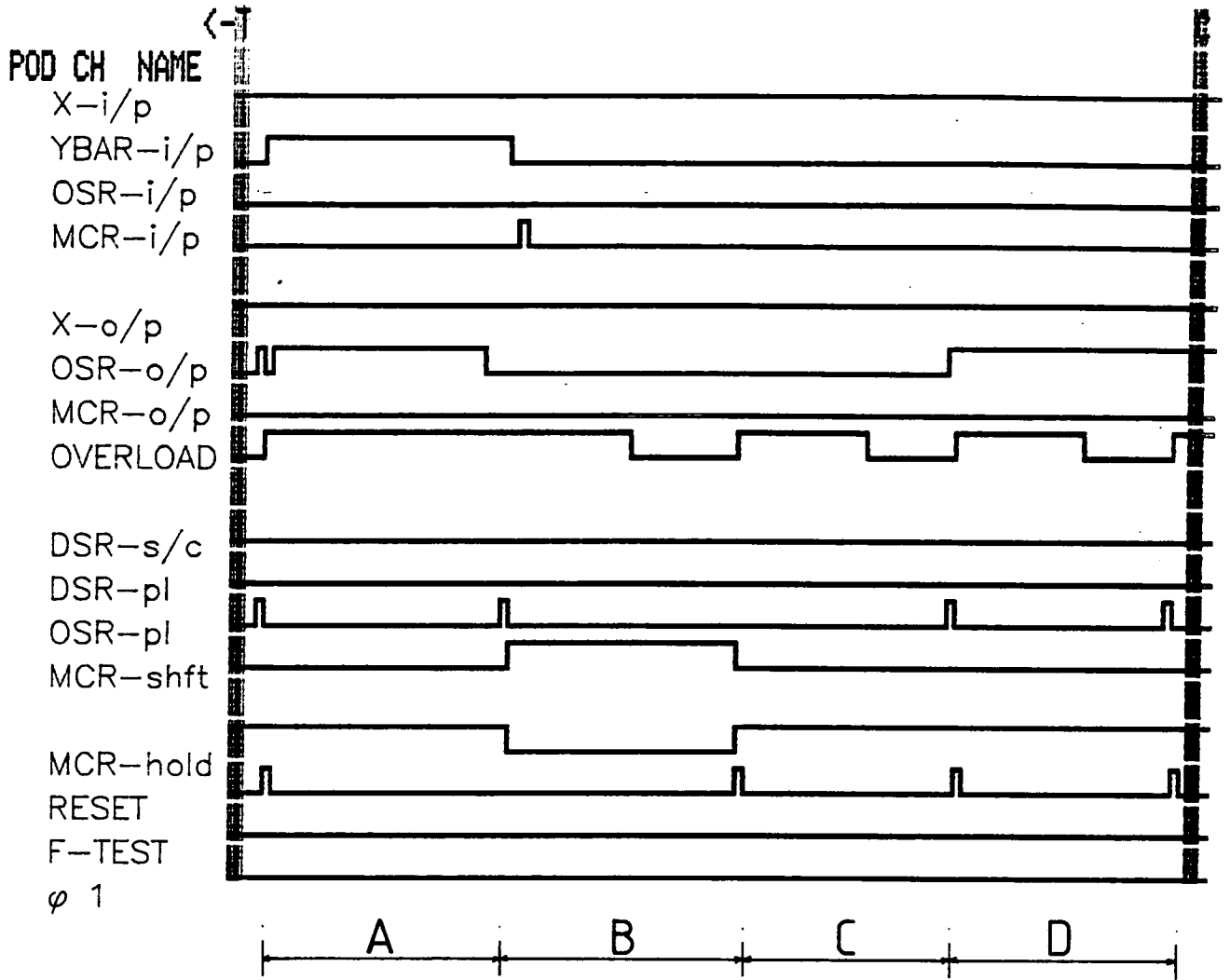


Figure 5.15. Zoom in on the self repair sequence.

Part "B" shows the logic one in the MCR input being shifted into the bit position of the MCR that corresponds to the second stage in the correlator array. Part "C" represents a correlation of the input combination $x = y = 1$. The overload output can be seen to go LOW, as expected, after the prescribed 15 clock cycle integration time. Part "D" of the figure shows the overload pattern displayed on the OSR serial output. The period between the RESET and OSR-pl pulses is now 27 clock cycles so that the OSR is reloaded with correlation results before data shifted from its serial input appears at the serial

output.

The full self test and self repair sequence, as described in Appendix 2, uses the F-test signal to emulate the expected overload pattern. The action of the F-test signal is to invert the overload pattern shown in Figure 5.15, so that it contains a logic one at the position of the faulty stage. The self repair sequence would then simply transfer this bit pattern, in parallel, to the MCR.

The correlation test, as described in Section A2.10, is similar to the self test except for the action of the F-test signal. However, the self test sequence shown in Figures 5.13 and 5.14, is a modification of test A2.8 that demonstrates both self test and correlation test. Therefore correlation test need not be treated separately.

5.6.3. Yield Enhancement

This section contains the results of the first 130 processed chips. Figure 5.16 shows a chart of number of chips plotted against number of working stages. It shows that 29 of the 130 candidates passed the initial test and that 27 of these yielded more than 20 stages of correlation.

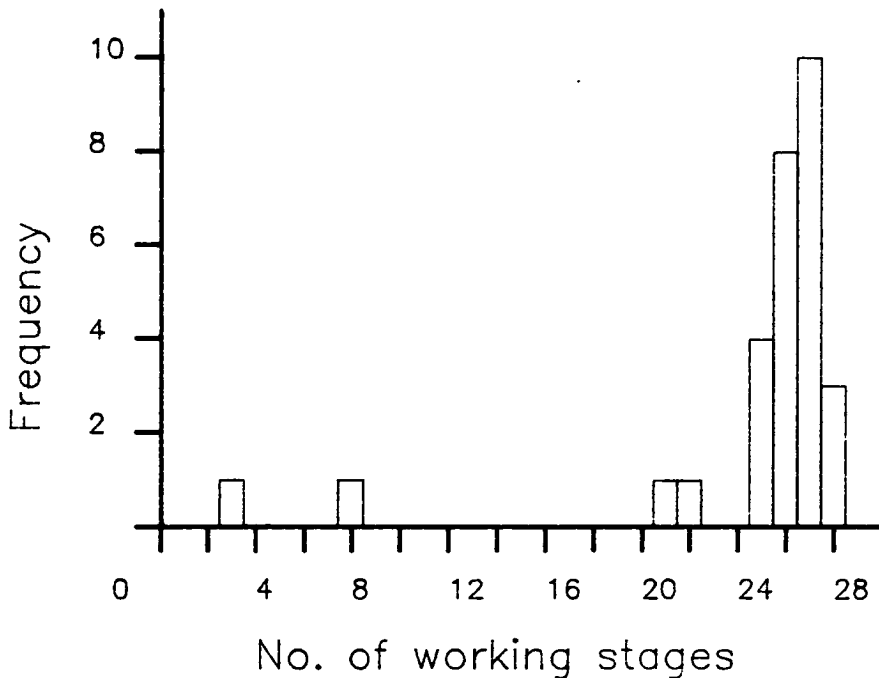


Figure 5.16. Distribution of functioning stages.

Listed below are the test results for each wafer. The multi-project wafers each contained 24 correlator chips.

TABLE 5.6
RESULTS OF CHIP TEST

Candidates tested	Without Self Repair (100% working)	With Self Repair (at least 75% working)
Packaged (10)	0	2
Wafer #1 (24)	1	5
Wafer #2 (24)	0	5
Wafer #3 (24)	0	6
Wafer #4 (24)	0	0
Wafer #5 (24)	2	9
TOTALS (130)	3	27
YIELD (%)	2.3	20.7

Although these results are based on a small statistical population (130 chips), they show nevertheless a strong agreement with the theoretically predicted figures. For example, the expected distribution of number of working stages, as predicted by Equation 4.16, is shown in Figure 5.17.

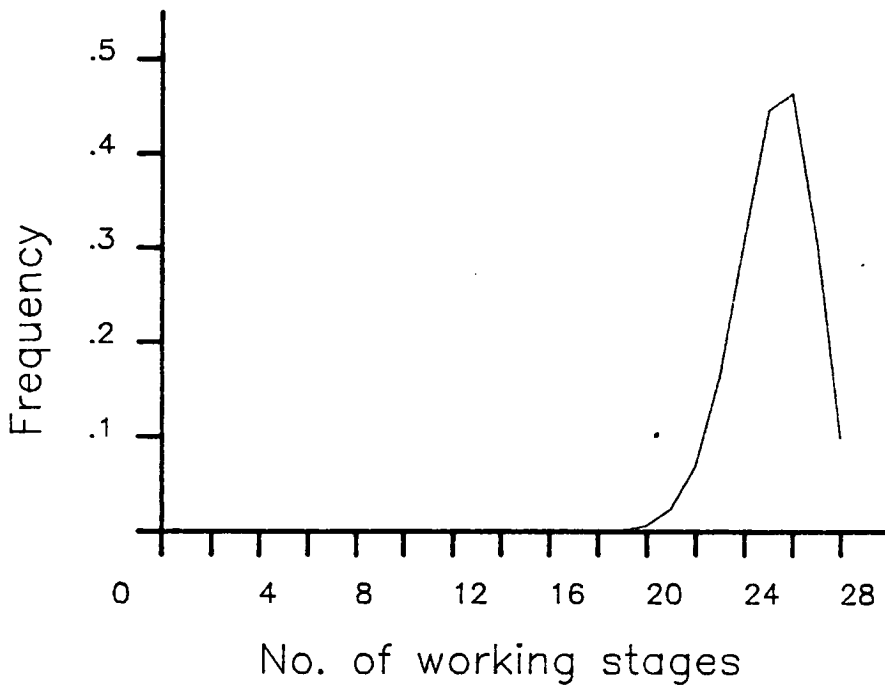


Figure 5.17. Distribution of working stages according to Equation 4.16.

5.7. Summary

In this chapter, the architecture and design of the Eu349 digital correlator has been described. Additions to the basic architecture, that make possible built-in self test and self repair strategies have been discussed. The net result of the design strategy, that closely links design to test, is a well structured, and regular VLSI architecture.

The test results show the correct operation of the device as a correlator, and demonstrate the principles of self test and self repair.

CHAPTER 6

CONCLUSIONS

6.1. Summary of Work

This thesis has described built-in self test and self repair strategies in VLSI architectures for digital correlation. In Chapter 2, correlation theory was presented. Correlation techniques from analogue through to digital polarity implementations were discussed. It has been shown that, for stationary, ergodic signals, a temporal correlation function with finite integration time can approximate the true correlation coefficient. The effects of sampling, quantisation, and dither have been described. The main conclusion is that any physically realisable correlation system must compromise accuracy with integration time, and measurement time with circuit complexity.

The overloading integrating counter technique for polarity correlation has also been described, and the prototype correlator chip, featuring built-in self test and self repair mechanisms, has been introduced.

In Chapter 3, several implementations of silicon correlators have been discussed. The architectures may be classified by observing whether time integrating or spatially integrating techniques have been used. The difference between these two concepts has been illustrated by the correlation cube. Further segregation of correlator architectures may be made by observing which computational techniques have been used, namely bit serial, bit parallel, polarity, systolic etc.

Parallel and concurrent techniques are employed to an ever increasing extent in integrated circuit correlators. However there exists a compromise between using a large number of very simple concurrent operations, and using a small number of complex cells, to achieve a common objective. In the DELTIC correlator, discussed in Section 3.2.3, a single, fast, multiplier is used. In the systolic correlator, discussed in Section 3.2.4.2, delay, multiply, and add operations are distributed over a large 2-dimensional array of simple cells. However, partial products are only generated in cells within an interaction region and these in turn are only used to form a product on every alternate clock cycle. Furthermore, to achieve useful integration times a large array of cells is required, and to increase the integration time requires cells to be cascaded. Normally this would not be a disadvantage; it is in fact preferable for VLSI architectures to be modular and cascadable. However the output rate of this correlator is inversely proportional to the size of the array.

The architecture of the Eu349 correlator achieves a balance between concurrency, cascadability and correlation rate. The architecture is concurrent in that each point of the correlation function is computed in parallel. The architecture is directly cascadable, and the correlation rate is independent of the length of the array.

In Chapter 4, two important subjects in integrated circuit design, namely testability and yield, have been discussed. Methods by which the testability of a design may be enhanced have been described and summarised for the particular case of the Eu349 correlator chip. Close linking of design and test has enabled the architecture of the Eu349 to achieve a high degree of testability at a very low overhead.

Yield enhancement through the use of redundant circuitry is of central importance to the design of the Eu349 correlator chip. Yield models and yield enhancement techniques, have been described. A binomial model for the yield of redundantly designed circuits has been presented, and the curves produced by the model show that the yield of an array of identical modules, such as in the Eu349 correlator, increases rapidly with the addition of spare modules, but saturates to a level determined by the defect density and the uncorrectable area of the chip.

In Chapter 5, the architecture and design of the Eu349 digital correlator has been described. Additions to the basic architecture, that make possible built-in self test and self repair strategies have been discussed. The net result of the design strategy, that closely links design to test, is a well structured, and fault tolerant VLSI architecture.

The test results show the correct operation of the device as a correlator, and demonstrate the principles of self test and self repair. Results from the first batch of processed wafers have demonstrated that yield can be improved considerably at a very low cost in circuit overhead; the initial sample's yield enhancement factor was 9.0 for 130 chips tested. In addition, any of these chips can be given an exhaustive functional test in less than 150 ms at 1 MHz.

6.2. Further Work

The work described by this thesis provides a significant base for further research. Both the self repair aspect of the VLSI architecture, and the advantages it holds for high speed digital correlation would be worth further investigation. One such project would involve

redesigning the correlator array on to a wafer of its own so that many thousands of the chips may be made. With such large numbers of test candidates, a comprehensive yield model for the fabrication process could be established. Another research topic would be to expand the correlation architecture and self repair technique to multibit direct-digital correlation.

The investigation of large area silicon systems is rapidly becoming an important topic in microelectronics research. The correlator architecture discussed here would play a significant role in the development of a wafer scale, or large area silicon system. If, for example, the correlator were fabricated on a $2\mu\text{m}$ CMOS process, the $7\text{ mm} \times 7\text{ mm}$ chip would contain approximately 256 parallel stages of correlation. Cascades of these chips would provide very attractive high speed, high resolution correlation systems.

In the prototype device, the control circuitry has not been included on chip. An interesting situation can be envisaged where each chip contains the required control circuitry to supervise any arbitrary length of correlation cascade. When these chips are cascaded, either discretely or as part of a wafer scale system, a second tier of fault tolerance can be introduced. This situation would be achieved if each correlator control circuit could be isolated from the correlation system. The system would consist of a cascade of identical chips, each with their own controller. However, only one controller in the entire cascade may be active at any time. The important fact is that it would not matter which controller was active. Thus, for a cascade of four correlator chips, there would be three redundant control circuits. The active control circuit, in addition to controlling the correlation array, would also control the other redundant control circuits.

This controller, the "master" chip, would signal all other control circuits to adopt their transparent mode. The system would be reconfigurable. Thus, in addition to the normal self repair and reconfiguration of the correlator array stages, the "master" controller can be reselected, if it is found to be defective. This concept has been investigated in a Master of Science degree project, and silicon layout has been produced for an overloading correlator with such a "master" controller [193].

In conclusion, the design of regular, cascadable VLSI architectures for high speed digital correlation, coupled with low circuit-overhead self test and self repair strategies, holds potential for the fabrication of high yielding large area silicon systems.

ACKNOWLEDGEMENTS

I offer sincere thanks to Dr. Mervyn A. Jack and Dr. James R. Jordan, my supervisors in this research work. I would also like to thank my colleagues in the Department of Electrical Engineering, and Wolfson Microelectronics Institute for their help and encouragement. Thanks are also due to the staff of the Edinburgh Microfabrication Facility for processing the integrated circuits.

I also thank my wife Alison, and my family, for their constant help and encouragement.

APPENDIX 1

EU349 CORRELATOR DESIGN

A1.1. Introduction

The Eu349 prototype polarity correlator is a monolithic n-channel MOS integrated circuit. The VLSI structure implements polarity correlation using an overloading integrating counter technique. The device architecture permits the direct cascading of individual correlator chips without the need for additional components, to give complete flexibility in choice of correlator delay and resolution. Additional features include programmable integration time, built-in self test, and built-in self repair capabilities.

The prototype device consists of a cascade of 28 identical correlation stages. Each stage comprises a delay element (DSR), an exclusive-NOR gate for the multiplication/comparison process of correlation, a 15-bit programmable integrating counter, and a counter overload latch. In addition the chip contains a parallel-in/serial-out shift register (OSR) for serially shifting the values of the correlation function off chip, and a parallel-in/parallel-out shift register (MCR) used to control the self repair multiplexers. There are two multiplexers per stage.

A1.2. Silicon Design

The Eu349 correlator chip consists of a correlator array and peripheral circuitry. The correlator array is composed of 28 identical modules. Each module, or correlation stage, consists of a further level of sub-modules. The design is structured in such a way that correlator stages may be cascaded by simple abutment. The correlator chip is composed of the following hierarchy:

```
<chip>      :   <array><peripheral circuitry>
<array>     :   <28 x <STG100>>
<STG100>    :   <DSR10><COMP10><MCR10><OSR10><OL10><PN100>
<PN100>     :   <PN30><7 x <PN10>><PN20>
```

where the module names have the following meanings:

```
<STG100>    :   Correlator Stage
<DSR10>     :   Data Shift Register and Mux.
<COMP10>    :   Comparator and PN100 Clock Buffer
<MCR10>     :   Multiplexer Control Register
<OSR10>     :   Overload Shift Register and Mux.
<OL10>      :   Overload Latch
<PN100>     :   Integrating Counter
<PN30>      :   Stage 15 and Feedback EXNOR
<PN10>      :   Repeated Section of Counter
<PN20>      :   Link between Stages 7 and 8
```

A block diagram of the integrating counter, module PN100, is shown in Figure A1.1. The counter consists of a cascade of 15 semistatic shift register stages, and one exclusive NOR module. The reasons for implementing the integrating counter in the shape of a ring with semistatic shift register elements, and the reasons for choosing exclusive NOR feedback instead of exclusive OR, are

discussed in Section 5.4.2. The overload detect circuitry, which consists of a 15 input NOR gate, is distributed throughout the counter.

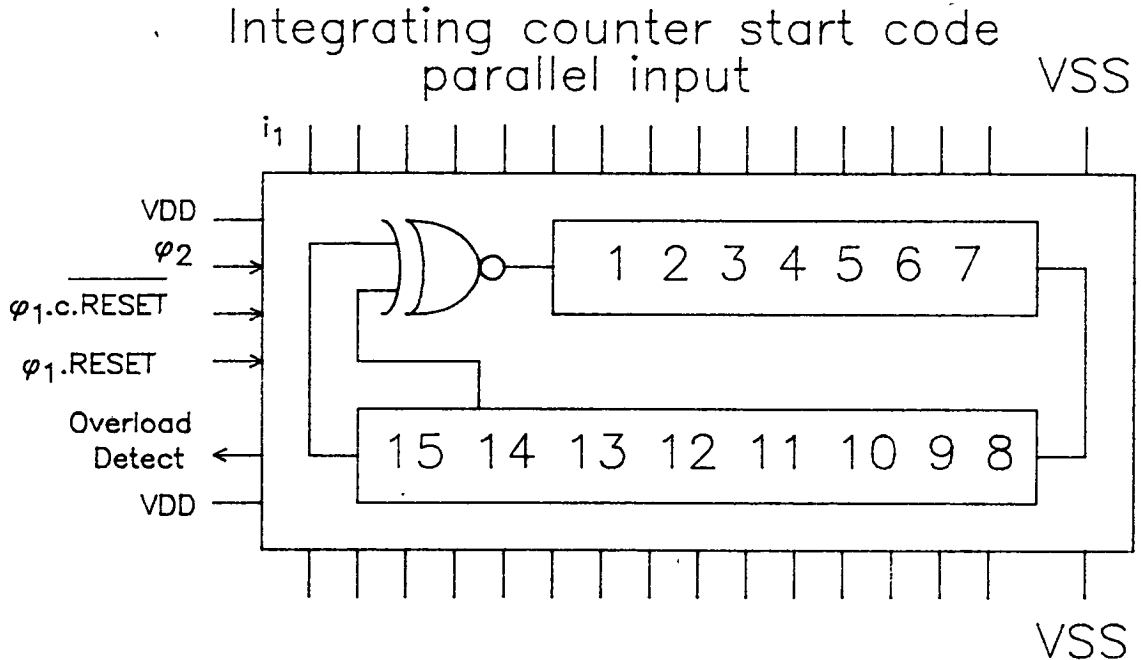


Figure A1.1. Block diagram of integrating counter.

The integrating counter is composed of three sub-modules: PN10, PN20, and PN30. Module PN10 contains two shift register stages: stage n and stage $15 - n$, where $1 \leq n \leq 7$, and is replicated seven times along the integrating counter. Module PN20 completes the connection between shift register stages 7 and 8, and provides the VSS connection to the correlator array. Module PN30 provides the exclusive NOR feedback connection of the integrating counter, incorporates shift register stage 15 and the depletion mode pull-up transistor that forms part of the overload detect NOR gate.

The shift register stages that make up the integrating counter are also used in other modules in the

correlator design. Simulation results for this basic semistatic shift register stage are shown in Figure A1.2. The input data for this simulation is shown in Figure A1.3. Figure A1.2 shows four voltage traces. The non-overlapping clocks, ϕ_1 and ϕ_2 , are drawn together in the top grid. The middle grid shows the input waveform to the shift register, and the lower grid shows the output waveform. The figure shows that the input data, which is sampled on ϕ_1 , appears on the output when ϕ_2 becomes active. The simulation shows the shift register working at 4 MHz.

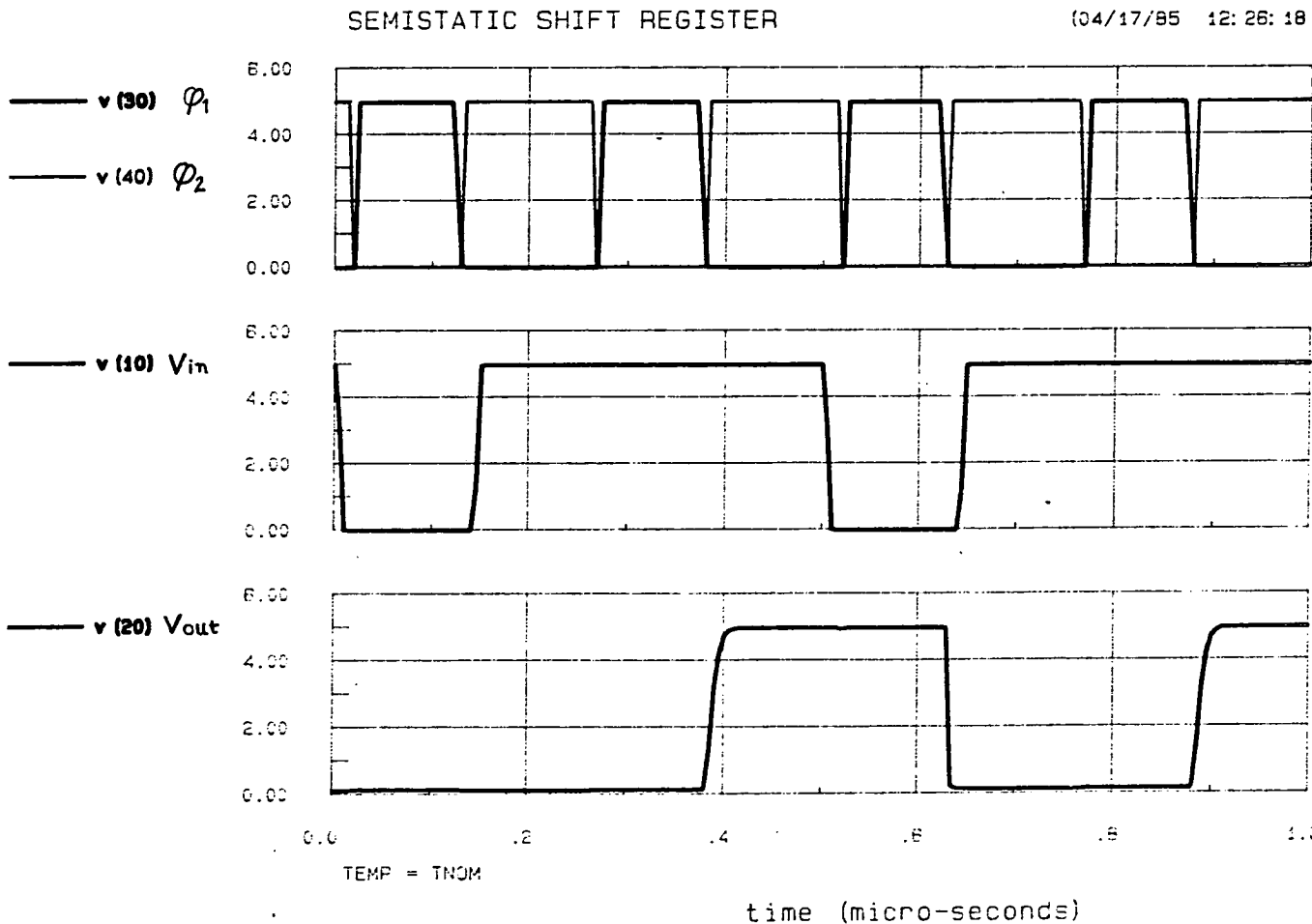


Figure A1.2. Simulation of basic shift register element.

```
SEMISTATIC SHIFT REGISTER
.SUBCKT INVK8 10 20 500 100
ME1 20 10 0 100 MENH2 6U 12U
MD1 500 20 20 100 MDEP2 24U 6U
.ENDS INVK8
.SUBCKT INVK4 10 20 500 100
ME1 20 10 0 100 MENH2 6U 12U
MD1 500 20 20 100 MDEP2 12U 6U
.ENDS INVK4
.SUBCKT MINPAS 10 20 30 100
MP1 10 20 30 100 MENH2 6U 6U
.ENDS MINPAS
.SUBCKT SRSS 10 20 30 40 500 100
XP1 10 30 35 100 MINPAS
XP2 35 40 45 100 MINPAS
XP3 55 40 65 100 MINPAS
XN1 35 55 500 100 INVK8
XN2 55 45 500 100 INVK4
XN3 65 20 500 100 INVK8
.ENDS SRSS
VDD 500 0 DC 5
VBB 100 0 DC -2.5
VP1 30 0 PULSE 0 5 20N 4N 2N 100N 250N
VP2 40 0 PULSE 5 0 16N 4N 2N 110N 250N
VIN 5 0 PULSE 5 0 2N 8N 4N 134N 500N
XSR1 10 20 30 40 500 100 SRSS
CLOAD 20 0 0.05P
.TRAN 5N 1000N
.GRAPH TRAN V(10) V(20) V(30) V(40)
.WIDTH OUT=80
.MODEL MENH2 NMOS (LEVEL=2 VTO=0.75 GAMMA=0.46
+CGSO=4.5E-10 CGDO=4.5E-10 CJ=1.0E-4 CJSW=1.0E-9 JS=1.0E-7
+TOX=8E-8 NSUB=8.5E14 NFS=1E10 XJ=1.5U LD=1.25U UO=700
+UEXP=0.1 UTRA=0.3 VMAX=5E4 NEFF=3.0 XQC=0.4 DELTA=1.0)
.MODEL MDEP2 NMOS (LEVEL=2 VTO=-4.7 GAMMA=0.7
+CGSO=4.5E-10 CGDO=4.5E-10 CJ=1.0E-4 CJSW=1.0E-9 JS=1.0E-7
+TOX=8E-8 NSUB=2.0E15 NFS=1E10 XJ=1.5U LD=1.25U UO=550
+UEXP=0.1 UTRA=0.3 VMAX=5E4 NEFF=3.0 XQC=0.4 DELTA=1.0)
.END
```

Figure A1.3. Input data for shift register simulation.

A1.3. Peripheral Circuitry Design

The peripheral circuitry consists of input and output buffers. It also contains three combinational logic circuits for the generation of control signals. The output

buffers are standard library designs, slightly modified to fit the available space. The input buffers are designed according to the capacitive load that they must drive so that the chip can operate at 4 MHz. The capacitance is determined by calculating the number of gates to be driven, and the area of interconnect. A buffer with a drive capability of 5pF in a rise time of 20ns is adequate for all but two input pads. The remaining inputs are $\phi 1$, which requires a drive capability of 12pF in 20ns, and $\phi 2$, which requires a drive capability of 65pF in 20ns. Circuit simulations for each of these buffers are shown in Figures A1.4 to A1.9.

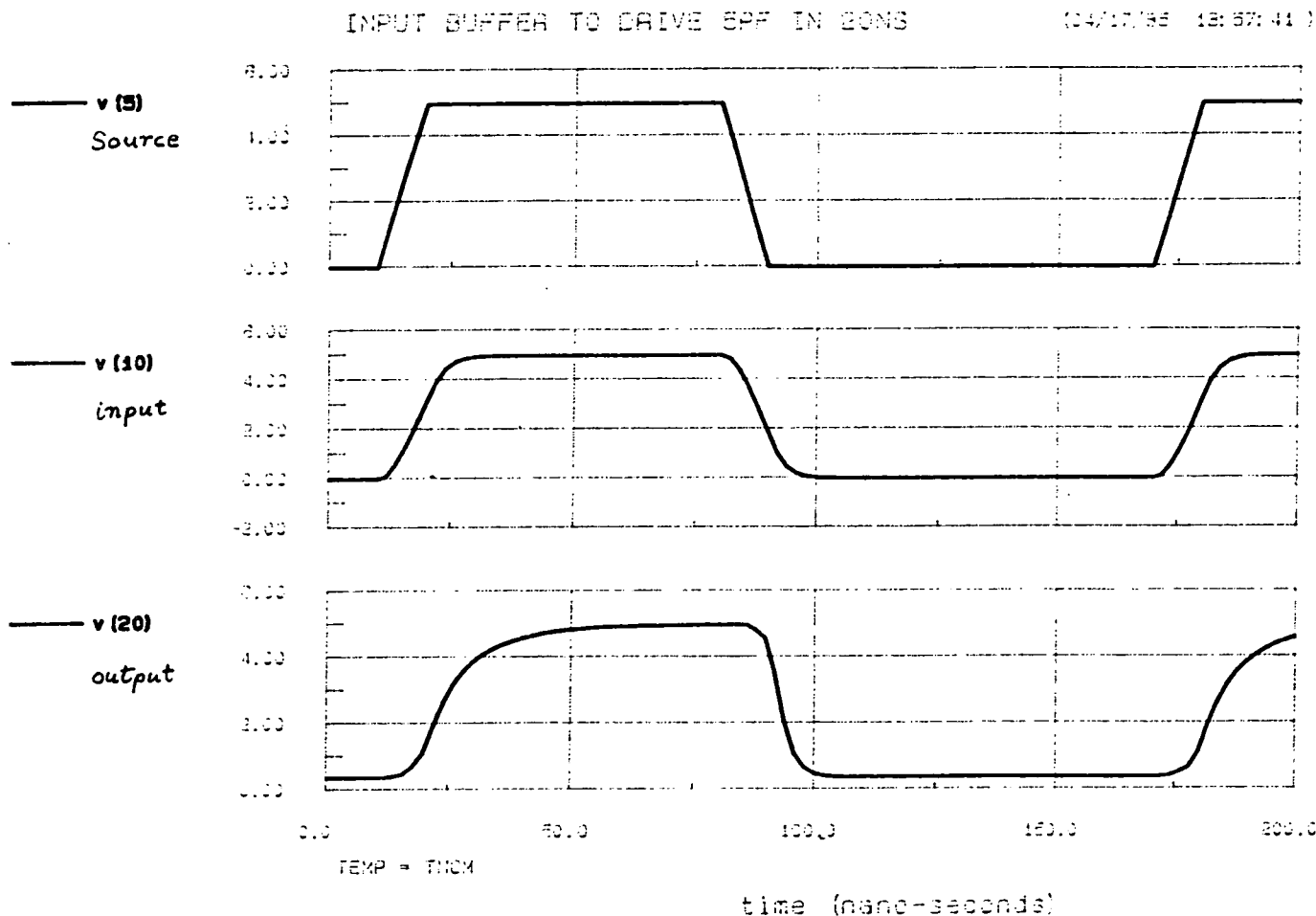


Figure A1.4. Simulation results for 5pF input buffer.

```
INPUT BUFFER TO DRIVE 5PF IN 20NS
.SUBCKT BUFF 10 30 500 100
ME1 20 10 0 100 MENH2 6U 96U
MD1 500 20 20 100 MDEP2 6U 24U
ME2 30 20 0 100 MENH2 6U 96U
MD2 500 10 30 100 MDEP2 6U 24U
ME3 500 10 30 100 MENH2 6U 72U
.ENDS BUFF
VDD 500 0 DC 5
VBB 100 0 DC -2.5
VIN 5 0 PULSE 0 5 10N 10N 10N 60N 160N
RCABLE 5 6 50
CCABLE 6 0 50P
RP 6 10 1K
CP 10 0 1P
X1 10 20 500 100 BUFF
CL 20 0 5P
.TRAN 2N 200N
.GRAPH TRAN V(5) V(10) V(20)
.MODEL MENH2 NMOS (LEVEL=2 VTO=0.75 GAMMA=0.46
+CGSO=4.5E-10 CGDO=4.5E-10 CJ=1.0E-4 CJSW=1.0E-9 JS=1.0E-7
+TOX=8E-8 NSUB=8.5E14 NFS=1E10 XJ=1.5U LD=1.25U UO=700
+UEXP=0.1 UTRA=0.3 VMAX=5E4 NEFF=3.0 XQC=0.4 DELTA=1.0)
.MODEL MDEP2 NMOS (LEVEL=2 VTO=-4.7 GAMMA=0.7
+CGSO=4.5E-10 CGDO=4.5E-10 CJ=1.0E-4 CJSW=1.0E-9 JS=1.0E-7
+TOX=8E-8 NSUB=2.0E15 NFS=1E10 XJ=1.5U LD=1.25U UO=550
+UEXP=0.1 UTRA=0.3 VMAX=5E4 NEFF=3.0 XQC=0.4 DELTA=1.0)
.END
```

Figure A1.5. Simulation data for 5pF input buffer.

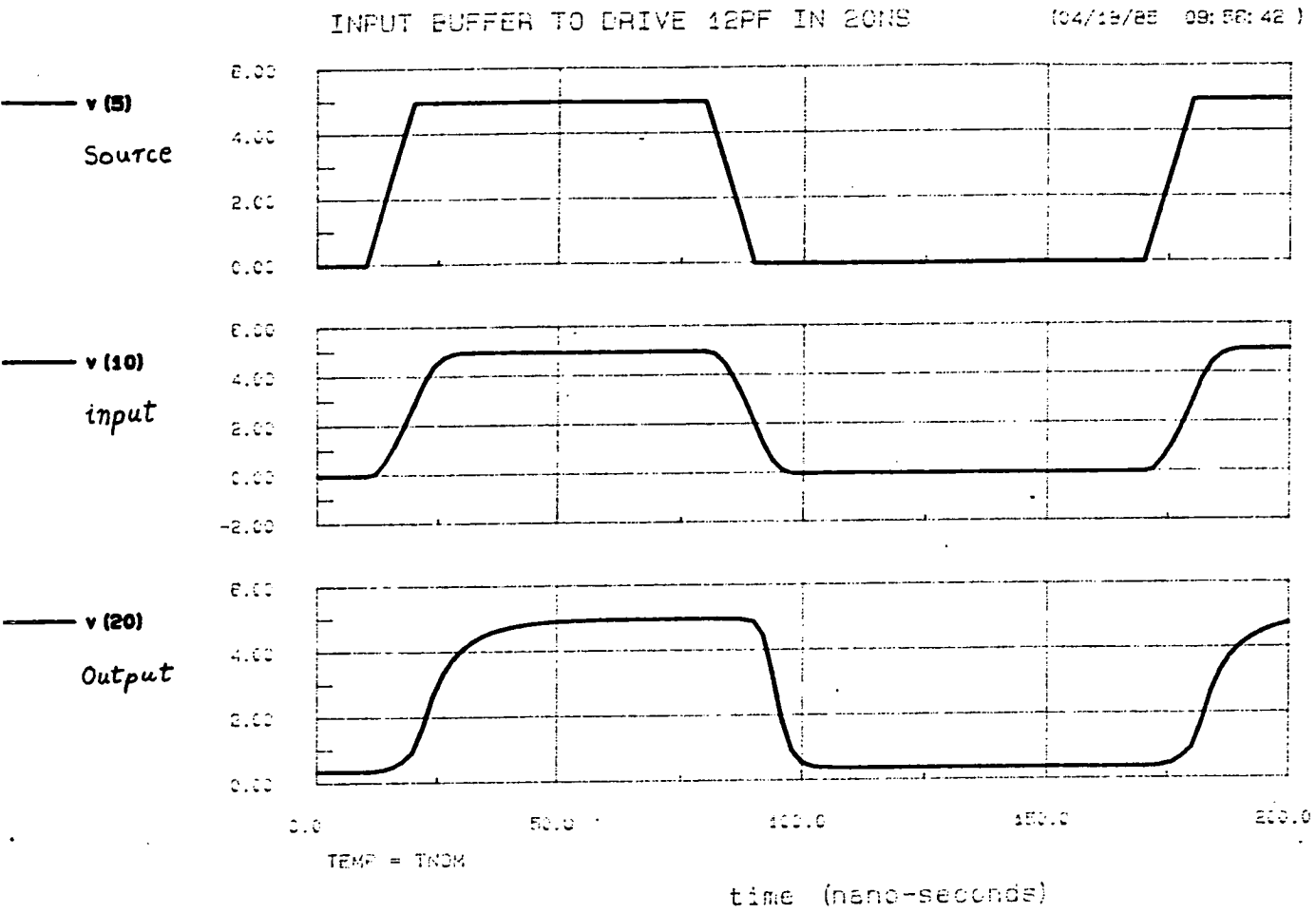


Figure A1.6. Simulation results for 12pF input buffer.

```
INPUT BUFFER TO DRIVE 12PF IN 20NS
.SUBCKT BUFF 10 30 500 100
ME1 20 10 0 100 MENH2 6U 96U
MD1 500 20 20 100 MDEP2 6U 24U
ME2 30 20 0 100 MENH2 6U 384U
MD2 500 10 30 100 MDEP2 6U 96U
ME3 500 10 30 100 MENH2 6U 288U
.ENDS BUFF
VDD 500 0 DC 5
VBB 100 0 DC -2.5
VIN 5 0 PULSE 0 5 10N 10N 10N 60N 160N
RCABLE 5 6 50
CCABLE 6 0 50P
RP 6 10 1K
CP 10 0 1P
X1 10 20 500 100 BUFF
CL 20 0 15P
.TRAN 2N 200N
.GRAPH TRAN V(5) V(10) V(20)
.MODEL MENH2 NMOS (LEVEL=2 VTO=0.75 GAMMA=0.46
+CGSO=4.5E-10 CGDO=4.5E-10 CJ=1.0E-4 CJSW=1.0E-9 JS=1.0E-7
+TOX=8E-8 NSUB=8.5E14 NFS=1E10 XJ=1.5U LD=1.25U UO=700
+UEXP=0.1 UTRA=0.3 VMAX=5E4 NEFF=3.0 XQC=0.4 DELTA=1.0)
.MODEL MDEP2 NMOS (LEVEL=2 VTO=-4.7 GAMMA=0.7
+CGSO=4.5E-10 CGDO=4.5E-10 CJ=1.0E-4 CJSW=1.0E-9 JS=1.0E-7
+TOX=8E-8 NSUB=2.0E15 NFS=1E10 XJ=1.5U LD=1.25U UO=550
+UEXP=0.1 UTRA=0.3 VMAX=5E4 NEFF=3.0 XQC=0.4 DELTA=1.0)
.END
```

Figure A1.7. Simulation data for 12pF input buffer.

INPUT BUFFER TO DRIVE 65PF IN 20NS

104/10/95 10:08:53

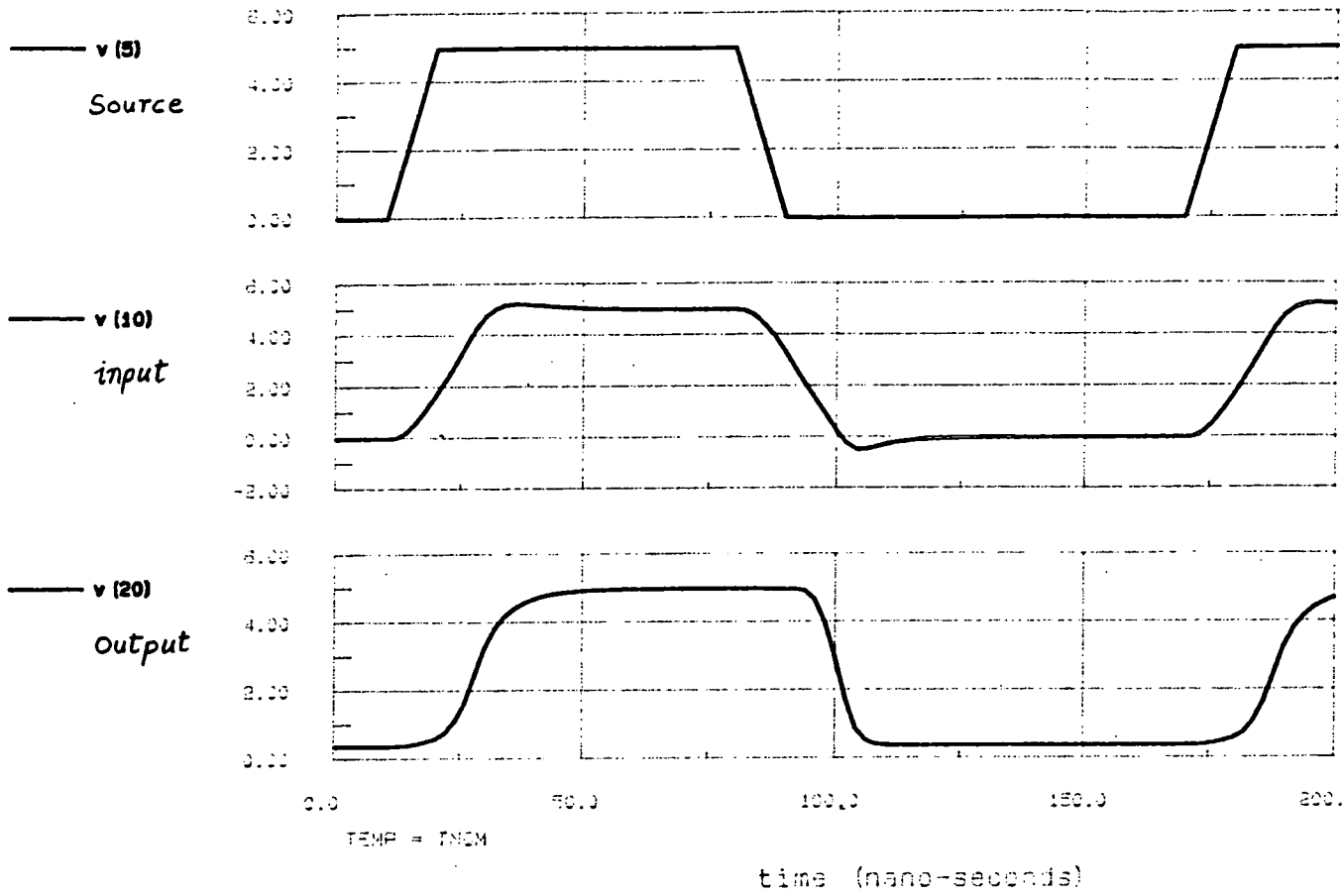


Figure A1.8. Simulation results for 65pF input buffer.

```
INPUT BUFFER TO DRIVE 65PF IN 20NS
.SUBCKT BUFF65 10 30 500 100
ME1 20 10 0 100 MENH2 6U 384U
MD1 500 20 20 100 MDEP2 6U 96U
ME2 30 20 0 100 MENH2 6U 3072U
MD2 500 10 30 100 MDEP2 6U 768U
ME3 500 10 30 100 MENH2 6U 2304U
.ENDS BUFF65
VDD 500 0 DC 5
VBB 100 0 DC -2.5
VIN 5 0 PULSE 0 5 10N 10N 10N 60N 160N
RCABLE 5 6 50
CCABLE 6 0 50P
RP 6 10 1K
CP 10 0 1P
X1 10 20 500 100 BUFF65
CL 20 0 100P
.TRAN 2N 200N
.GRAPH TRAN V(5) V(10) V(20)
.MODEL MENH2 NMOS (LEVEL=2 VTO=0.75 GAMMA=0.46
+CGSO=4.5E-10 CGDO=4.5E-10 CJ=1.0E-4 CJSW=1.0E-9 JS=1.0E-7
+TOX=8E-8 NSUB=8.5E14 NFS=1E10 XJ=1.5U LD=1.25U UO=700
+UEXP=0.1 UTRA=0.3 VMAX=5E4 NEFF=3.0 XQC=0.4 DELTA=1.0)
.MODEL MDEP2 NMOS (LEVEL=2 VTO=-4.7 GAMMA=0.7
+CGSO=4.5E-10 CGDO=4.5E-10 CJ=1.0E-4 CJSW=1.0E-9 JS=1.0E-7
+TOX=8E-8 NSUB=2.0E15 NFS=1E10 XJ=1.5U LD=1.25U UO=550
+UEXP=0.1 UTRA=0.3 VMAX=5E4 NEFF=3.0 XQC=0.4 DELTA=1.0)
.END
```

Figure A1.9. Simulation data for 65pF input buffer.

Combinational logic is included in the peripheral circuitry to generate control signals for the DSR, OSR and MCR. These circuits are shown in Figures A1.10 to A1.12 respectively. The circuits perform input buffering in addition to their logic functions.

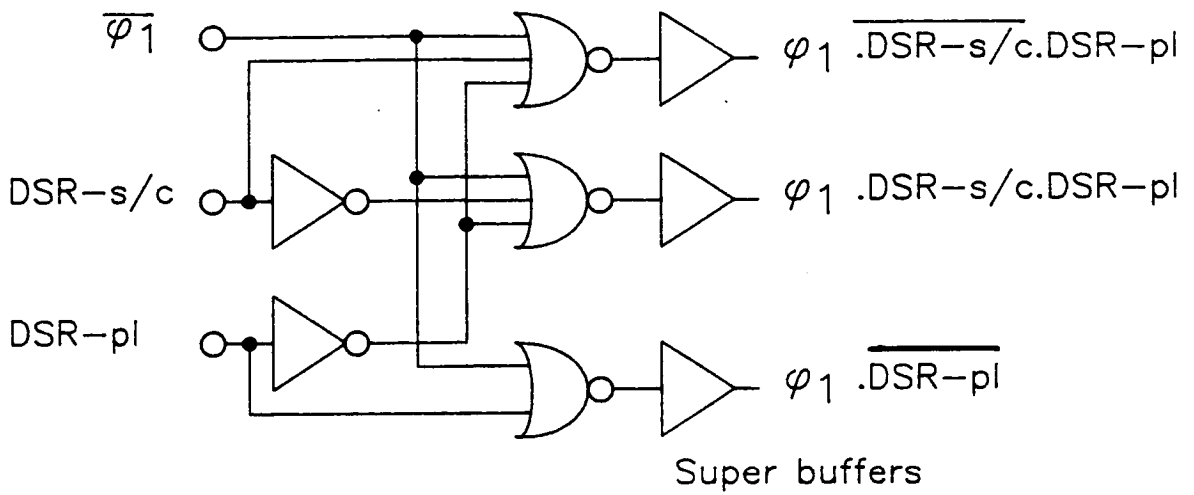


Figure A1.10. Logic for generating DSR control signals.

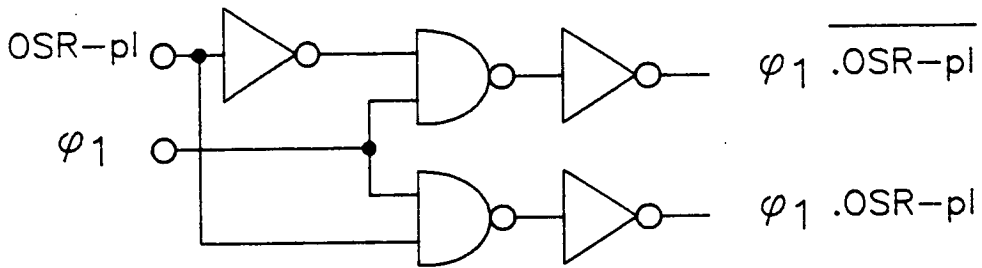


Figure A1.11. Logic for generating OSR control signals.

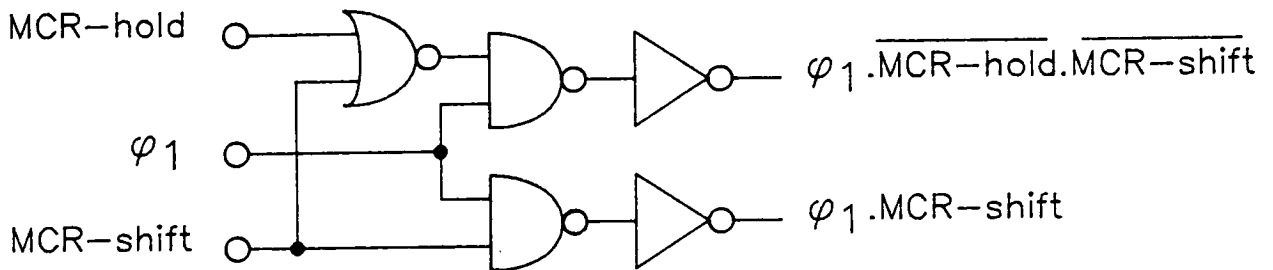


Figure A1.12. Logic for generating MCR control signals.

A1.4. Power Supply Considerations

Supply currents for logic gates are calculated under the approximations that the pull-down's resistance is negligible, and that the pull-up transistor is in saturation. Thus,

$$I_{\text{sat}} = k' \frac{W}{L} (V_{\text{gs}} - V_{\text{th}})^2, \quad V_{\text{ds}} \geq (V_{\text{gs}} - V_{\text{th}}), \quad \text{A1.1}$$

$$\approx 0.3 \frac{W}{L} \text{ mA}, \quad \text{A1.2}$$

where:

$$k' \approx 20 \mu\text{A}/\text{V}^2,$$

$$V_{\text{gs}} \approx 0\text{V},$$

$$V_{\text{th}} \approx -4\text{V}, \text{ and}$$

$$V_{\text{ds}} \approx 5\text{V}.$$

W and L are the width and length of the active area of the pull-up device.

The VDD and VSS metal line width is determined by the current carrying capability of the aluminium tracks. The metal migration limit is estimated to be $1 \text{ mA}/\mu\text{m}^2$, where the metal thickness is $1 \mu\text{m}$. Thus, a $10 \mu\text{m}$ metal track can carry 10 mA . Taking the worst case condition where all inverters in the correlator chip are turned on, a supply current estimate can be estimated to be $\sim 150 \text{ mA}$. Therefore, the VDD and VSS metal line widths must be $150 \mu\text{m}$. However, since there are two each VDD and VSS pads, the major power line widths can be reduced to $75 \mu\text{m}$.

APPENDIX 2

EU349 TEST SCHEDULE

A2.1. Introduction

The Test Schedule consists of a series of simple test procedures. Each procedure is presented as a subsection containing a list of the individual tests required to verify a specific chip function.

The test schedule procedures are summarised in Table A2.1.

TABLE A2.1 Summary of Test Schedule	
A2.2.	Test MCR as shift register.
A2.3.	Test OSR and DSR as shift registers.
A2.4.	Test MCR effect on both OSR and DSR.
A2.5.	Test SET and CLEAR features of DSR.
A2.6.	Test latches and MCR parallel load.
A2.7.	Test latches and OSR parallel load.
A2.8.	Verify self test sequence.
A2.9.	Verify self repair sequence.
A2.10.	Verify correlation performance.

A2.2. Test MCR as Shift Register

The Multiplexer Control Register is a critical element in the correlator circuit. It is required to be functional and neutralised before other tests are

attempted. (It is neutralised by flushing with zeros and holding the all zero state static.)

1. Perform a flush test on the MCR. (Flush test is described in Section 4.3.3.)
2. Perform a shift test by shifting the pattern 00110011... through the MCR. (Shift test is described in Section 4.3.3.)
3. Check that the time delay between MCR input and MCR output is 28 clock cycles.
4. Neutralise MCR.

A2.3. Test OSR and DSR as Shift Registers

The MCR must be neutralised before carrying out this test.

1. Neutralise MCR.
2. Perform a flush test on both the OSR and DSR.
3. Perform a shift test on both the OSR and DSR.
4. Check that the time delays through the OSR and DSR are 28 clock cycles.

A2.4. Test MCR Effect on both OSR and DSR

This test should be repeated with the input patterns 11001100..., 01100110..., 00110011... and 10011001....

1. Shift a binary pattern containing n ones into the MCR.

2. Hold MCR contents static.
3. Flush test, and shift test the OSD and DSR.
4. Check that the time delays through the OSR and DSR are $28 - n$ clock cycles. Note that the bypass circuitry is not guaranteed to work correctly if n contains blocks of more than 3 ones together.

A2.5. Test SET and CLEAR Features of DSR

Neutralise the MCR before performing this test.

1. Parallel load logic ones into DSR (SET DSR). This is done by asserting the control signals DSR-pl and DSR-s/c both HIGH.
2. Serially shift 28 zeros into DSR and observe DSR output.
3. Serially shift 28 logic ones into DSR.
4. Parallel load zeros into DSR (CLEAR DSR). Assert DSR-pl control HIGH, and DSR-s/c control LOW.
5. Serially shift 28 logic ones into DSR and observe DSR output.

A2.6. Test Latches and MCR Parallel Load

Neutralise the MCR before performing this test. This test sets/resets the overload latches, and transfers the latch contents to the MCR. The normal input to the latch, that is, the overload detect output from the integrating counter, cannot be disabled. Therefore, the integrating counter start word must be set to 4000-hex, or some other suitably large number, to prevent the integrating counter generating an overload detect output during the test.

(The RESET control also loads the integrating start word.)

1. Reset all latches using RESET control signal.
2. Parallel load latches into MCR. Assert MCR-shift LOW; MCR-hold LOW.
3. Shift 28 zeros into MCR and observe MCR serial output. Assert MCR-shift HIGH.
4. Reset again all latches using RESET control signal.
5. Set all latches using F-Test control signal. Assert F-test HIGH.
6. Parallel load MCR.
7. Shift 28 zeros into MCR and observe MCR serial output.

A2.7. Test Latches and OSR Parallel Load

Neutralise the MCR before performing this test. This test sets/resets the overload latches, and transfers the latch contents to the OSR. The normal input to the latch, the overload detect output from the integrating counter, cannot be disabled. Therefore, the integrating counter start word must be set to 4000-hex, or some other suitably large number, to prevent the integrating counter generating an overload detect output during the test. (The RESET control also loads the integrating start word.)

1. Reset all latches using RESET control signal.
2. Parallel load latches into OSR. Assert OSR-pl HIGH.

3. Shift 28 zeros into OSR and observe OSR serial output. Assert OSR-pl LOW.
4. Reset again all latches using RESET control signal.
5. Set all latches using F-Test control signal. Assert F-test HIGH.
6. Parallel load OSR.
7. Shift 28 zeros into OSR and observe OSR serial output.

A2.8. Self Test Sequence

The self test sequence consists of four repetitions of a single test. The test begins by resetting the correlator and setting up the initial conditions to each correlation stage. The objective of the test is to make each correlator stage correlate the same data. They should all then produce the same result which can easily be verified using the F-test signal to emulate the expected good response of the correlator array. The test is repeated for each of the four combinations of input data, $xy = 00, 01, 11, 10$. The F-test signal is set HIGH to correspond to the expected overload detect pulse at the end of the integration period of the "00" input combination. It is again set HIGH for the duration of the expected overload at the end of the integration period of the "11" combination. At all other times F-test is LOW.

Neutralise the MCR before performing this test.

1. Reset all latches and load 4000-hex into integrating counters using the RESET control. This number represents the maximum integration time of the correlator.

2. Clear the DSR using the DSR-pl and DSR-s/c controls (DSR-pl HIGH; DSR-s/c LOW). Set up the input conditions with $x = 0$ and $y = 0$, and shift zeros into the DSR.
3. Emulate the expected response of the circuit with F-test signal. In this case F-test is set HIGH to coincide with the expected overload.
4. Set up the input conditions with $x = 0$ and $y = 1$, shift zeros into the DSR.
5. Emulate the expected response of the circuit with F-test signal. In this case F-test is held LOW.
6. Set the DSR using the DSR-pl and DSR-s/c controls (DSR-pl HIGH; DSR-s/c HIGH). Set up the input conditions with $x = 1$ and $y = 1$, shift logic ones into the DSR.
7. Emulate the expected response of the circuit with F-test signal. In this case F-test is set HIGH to coincide with the expected overload.
8. Set up the input conditions with $x = 1$ and $y = 0$, shift logic ones into the DSR.
9. Emulate the expected response of the circuit with F-test signal. In this case F-test is held LOW.
10. Parallel load latches into the OSR.
11. Shift contents of the OSR to observe the number of detected faults.

A2.9. Self Repair Sequence

This test directly follows the self test sequence. The self test status of the correlator array is stored in the overload latches. The self repair sequence consists simply of transferring this information to the MCR.

1. Parallel load MCR using the controls MCR-shift and MCR-hold (both LOW).
2. Hold contents of MCR static (MCR-shift LOW; MCR-hold HIGH).
3. The self test procedure may now be repeated, without neutralising the MCR, to verify the self repair procedure.

A2.10. Correlation Test Sequence

Correlation test is similar to the self test sequence except that the F-test signal is inactive, that is, held LOW. As in self test, the data inputs are cycled through the four combinations 00, 01, 11, 10. This test is performed using a short integration time so that the results of all four input combinations may be displayed together on the DAS screen. The integration time is chosen to be less than the length of the correlator array, which in the case of one Eu349 device is 28 delay stages. This means that when the input conditions are such that an overload should occur after the specified integration time, then it can be displayed by parallel loading the OSR and shifting out its contents every 28 clock cycles.

The correlation test may be carried out either before or after self repair has been carried out. If it is to be performed before self repair, then the MCR must be neutralised; OSR parallel load and RESET pulses should occur

every 28 clock cycles. In this case the test will show the response of all correlation stages, including faulty ones if they exist. When the test is performed after self repair, the MCR must be held static to maintain the configuration of the array. Also, the OSR-pl and RESET pulses must occur every n clock cycles, where n represents the *effective* length of the correlator array, i.e. n is the number of zeros held in the MCR.

Correlation test requires the following sequence of events:

1. Reset all latches and load 5555-hex into integrating counters using the RESET control. This number represents an integration time of 15 clock cycles. It is implied here that 5555-hex will be loaded on each subsequent RESET pulse.
2. At the same time clear the DSR using the DSR-pl and DSR-s/c controls (DSR-pl HIGH; DSR-s/c LOW). Also, set up the input conditions with $x = 0$ and $y = 0$, and shift n zeros into the DSR.
3. Parallel load the OSR (OSR-pl HIGH) and RESET correlator.
4. At the same time set up the input conditions with $x = 0$ and $y = 1$, shift n zeros into the DSR.
5. Parallel load the OSR (OSR-pl HIGH) and RESET correlator.
6. At the same time set the DSR using the DSR-pl and DSR-s/c controls (DSR-pl HIGH; DSR-s/c HIGH). Set up the input conditions with $x = 1$ and $y = 1$, shift n logic ones into the DSR.

7. Parallel load the OSR (OSR-pl HIGH) and RESET correlator.
8. At the same time set up the input conditions with $x = 1$ and $y = 0$, shift n logic ones into the DSR.
9. Parallel load the OSR (OSR-pl HIGH) and RESET correlator.
10. Shift contents of the OSR to observe the results of paragraphs 7 and 8 above.

APPENDIX 3

EU349 TEST CONFIGURATION

A3.1. Introduction

The specific configuration for the Digital Analysis System (DAS) is described here. The description is divided into seven subsections which describe the individual menus and POD configurations, from the data probes, data acquisition, display and triggering menus, to the pattern generator program and instruction codes.

A3.2. Prototype Test Configuration

The bonding diagram and pin arrangement of the prototype IC is shown in Figure A3.1.

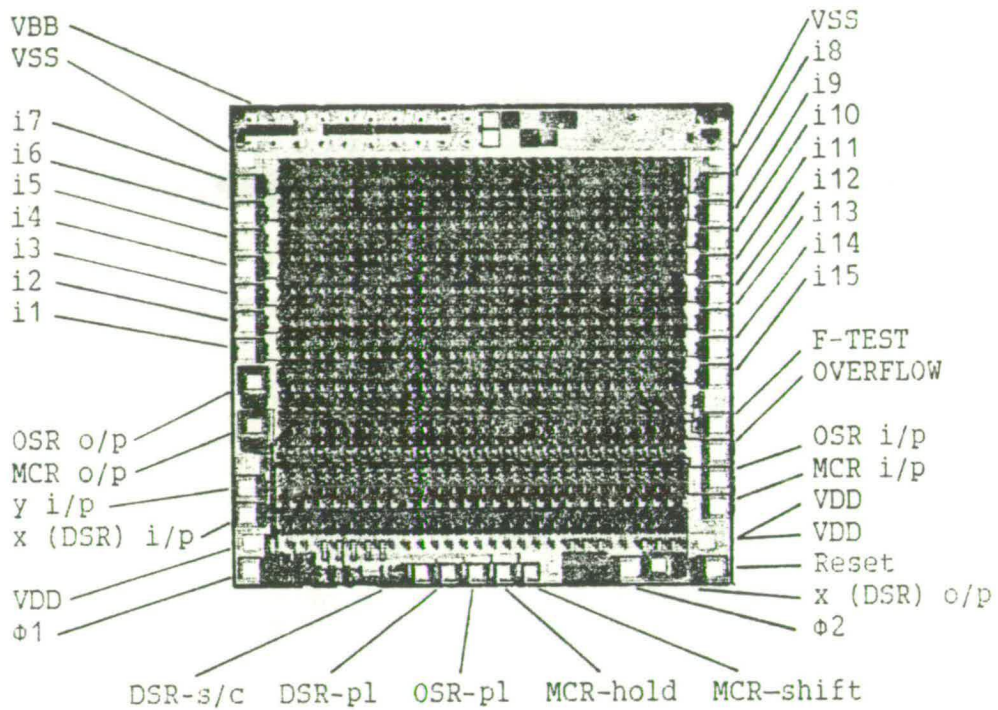


Figure A3.1. Eu349 bonding diagram.

Patterns of input and control data are generated by the "pattern generator" section of the DAS. Outputs from the chip under test, and if required, inputs and control signal are acquired and displayed by the "data acquisition" section of the DAS. DAS input and output is achieved through data probes.

A3.3. DAS Data Probes

The DAS is composed, for the purposes of this test, of the following modules:

1. One 91A32 Data Acquisition Module.
2. One 91P16 Pattern Generator Module.

3. One 91P32 Pattern Generator Module.

Each module has specific data probe connections, which are described here.

A3.3.1. 91A32 Data Acquisition Module

One 91A32 Data Acquisition Module with a maximum of 32 data channels at 25 MHz. is installed. In this case 16 channels are used to acquire and display inputs to, and outputs from the chip under test. These channels are listed in Table A3.1.

TABLE A3.1			
POD Assignments for 91A32 Data Acquisition Module			
Channel	Function	Channel	Function
POD 2A 0	OSR o/p	POD 2B 0	MCR-hold
POD 2A 1	MCR o/p	POD 2B 1	MCR-shift
POD 2A 2	y i/p	POD 2B 2	x (DSR) o/p
POD 2A 3	x (DSR) i/p	POD 2B 3	RESET
POD 2A 4	$\phi 1$	POD 2B 4	MCR i/p
POD 2A 5	DSR-s/c	POD 2B 5	OSR i/p
POD 2A 6	DSR-pl	POD 2B 6	OVRFLO
POD 2A 7	OSR-pl	POD 2B 7	F-TEST
POD 2A Q	...	POD 2B Q	...

A3.3.2. 91P16 Pattern Generator Module

One 91P16 Pattern Generator Module with a maximum of 16 data channels at 25 MHz. is installed. In this case 11 channels are used to provide all the control signals necessary for the correlator chip. These channels are listed in Table A3.2.

TABLE A3.2			
POD Assignments for 91P16 Pattern Generator Module			
Channel	Function	Channel	Function
POD 1B 0	MCR i/p	POD 1C 0	MCR-hold
POD 1B 1	OSR i/p	POD 1C 1	MCR-shift
POD 1B 2	y i/p	POD 1C 2	-
POD 1B 3	x (DSR) i/p	POD 1C 3	-
POD 1B 4	DSR-pl	POD 1C 4	OSR-pl
POD 1B 5	DSR-s/c	POD 1C 5	-
POD 1B 6	-	POD 1C 6	-
POD 1B 7	RESET	POD 1C 7	F-TEST
POD 1B STRB	$\phi 1$	POD 1C STRB	$\phi 2$
POD 1B CLK	-	POD 1C CLK	-

A3.3.3. 91P32 Pattern Generator Module

One 91P32 Pattern Generator Module with a maximum of 32 data channels at 25 MHz. is installed. In this case 15 channels are used to provide the parallel input, i1 to i15, to the integrating counters of the correlator chip. These channels are listed in Table A3.3.

TABLE A3.3			
POD Assignments for 91P32 Pattern Generator Module			
Channel	Function	Channel	Function
POD 4A 0	i1 (LSB)	POD 4B 0	i9
POD 4A 1	i2	POD 4B 1	i10
POD 4A 2	i3	POD 4B 2	i11
POD 4A 3	i4	POD 4B 3	i12
POD 4A 4	i5	POD 4B 4	i13
POD 4A 5	i6	POD 4B 5	i14
POD 4A 6	i7	POD 4B 6	i15
POD 4A 7	i8	POD 4B 7	...
POD 4A STRB	...	POD 4B STRB	...
POD 4A CLK	...	POD 4B CLK	...

The permitted values which may be given to the bits i1 to i15 are summarised in the section on Pattern Generator Instruction Codes, Section A3.8.

A3.4. Channel Specification

The Channel Specification menu is for controlling the display format of the data acquisition channels. It divides channels into groups, sets display radix and polarity values, and determines probe input thresholds.

Table A3.4 shows the grouping of the acquisition channels and their POD IDs into data inputs, data outputs, and control signals.

TABLE A3.4 Grouping of Acquisition Data			
Group	Name	POD ID	Function
A	x (DSR) i/p	2A 3	Data Inputs
	y i/p	2A 2	
	OSR i/p	2B 5	
	MCR i/p	2B 4	
B	x (DSR) o/p	2B 2	Data Outputs
	OSR o/p	2A 0	
	MCR o/p	2A 1	
C	OVRFLO	2B 6	Overflow Flag
D	DSR-s/c	2A 5	DSR Control
	DSR-pl	2A 6	
E	OSR-pl	2A 7	OSR Control
F	MCR-shift	2B 1	MCR Control
	MCR-hold	2B 0	
O	RESET	2B 3	Reset Latches & Load Counters
1	F-TEST	2B 7	Set Latches or Fault Repair
2	$\phi 1$	2A 4	Chip's $\phi 1$

The display radix and polarity fields (not shown) are set to binary and positive respectively. The probe input thresholds are all set to 2.6 volts MOS. PODs 2D and 2C, which are not required in the correlator test, are unsigned.

A3.5. Timing Diagram

Once in memory, acquired data may be displayed in a timing diagram format. In this format the DAS displays up to 16 logic waveforms representing the high and low states in each clock cycle. Screen editing is used for viewing different portions of memory, altering the display magnification, and for labelling and rearranging the channel orders.

Table A3.5 shows how the channels are labelled and rearranged for the correlator test.

TABLE A3.5 Arrangement and Labelling of Channels for Display		
POD ID	Name	Display
2A 3	x (DSR) i/p	input data
2A 2	y i/p	
2B 5	OSR i/p	
2B 4	MCR i/p	
2B 2	x (DSR) o/p	output data
2A 0	OSR o/p	
2A 1	MCR o/p	
2B 6	OVRFLO	
2A 5	DSR-s/c	CONTROL SIGNALS
2A 6	DSR-pl	
2A 7	OSR-pl	
2B 1	MCR-shift	
2B 0	MCR-hold	CHIP'S $\phi 1$
2B 3	RESET	
2B 7	F-TEST	
2A 4	$\phi 1$	

A3.6. Trigger Specification

The Trigger Specification menu is for controlling the modules used during data acquisition. It specifies which modules are used, their clock rates, clock qualifiers, and trigger parameters.

For the correlator test only one 91A32 data acquisition module is used, and it is operated from the DAS

internal clock. The trigger word is positioned at the beginning of the acquisition memory. The acquisition memory of the DAS is not large enough to store all data from the correlator chip during a complete test, so suitable trigger words must be specified to acquire the desired portion of the test results.

A3.7. Pattern Generator - Timing

The Timing sub-menu of the pattern generator is for entering the characteristics of the strobe signals asserted in the Program sub-menu. It is also used to select the pattern generator's start mode, either single step or run.

Figure A3.2 shows the Timing sub-menu. It indicates that STROBE 0, which is the output line labelled STRB from POD 1B, is set up to perform the $\phi 1$ function in the correlator chip; and that STROBE 1, the STRB output line from POD 1C, is set up to perform the $\phi 2$ function. (See also Table A3.2.)

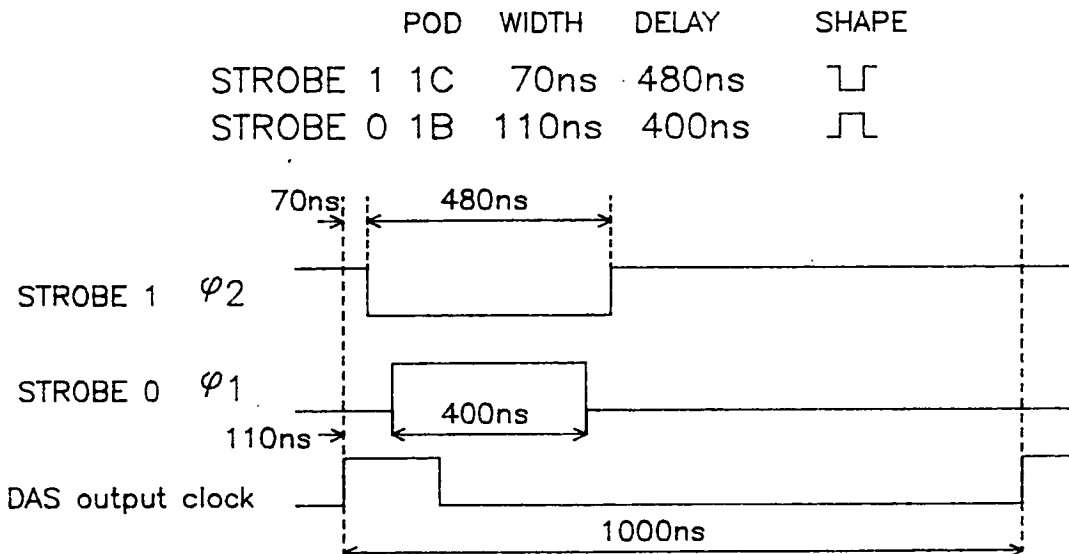


Figure A3.2. DAS Pattern Generator Timing Sub-Menu and clock strobes.

Figure A3.2 illustrates the specified features of the two strobe signals. STROBE 0 is the short duration clock phase. When STROBE 0 is high, information is transferred from one circuit element to another. STROBE 1 is the long duration clock phase. During this phase information is stored and maintained by the semistatic clocking scheme adopted in the correlator design.

A3.8. Pattern Generator Instruction Codes

This section provides a key to the pattern generator program instructions. The program, which is given in the next section, consists of a sequence of in-line instructions, each containing, *inter alia*, three fields for generating a bit pattern on the 48 (maximum) output lines. In this case, the fields of interest are the two relating to the PODs 4B and 4A, and PODs 1B and 1C.

Tables A3.6 and A3.7 show a list of correlator chip functions and their associated codes in hexadecimal. Table A3.6 is concerned with data input signals to the chip, while Table A3.7 is concerned with the function control signals.

TABLE A3.6		
DAS Instruction Codes for Correlator Input Signals		
Input Data	POD4BA	POD1CB
x, DSR (0)	0000	0000
x, DSR (1)	0000	0008
y (0)	0000	0004
y (1)	0000	0000
OSR (0)	0000	0000
OSR (1)	0000	0020
MCR (0)	0000	0000
MCR (1)	0000	0001

TABLE A3.7		
DAS Instruction Codes for Correlator Control Signals and Integrating Counters Start Value (ICSV).		
Chip Function	POD4BA	POD1CB
OSR serial shift	0000	0000
OSR parallel load	0000	1000
DSR serial shift	0000	0000
DSR set all ones	0000	0030
DSR set all zeros	0000	0010
MCR serial shift	0000	0200
MCR parallel load	0000	0000
MCR hold contents	0000	0100
RESET load & reset	ICSV	0080
F-TEST set latches	0000	8000

A particular program instruction is obtained by OR-ing the required function and data input codes in Tables

A3.6 and A3.7.

Whenever the RESET function is selected, which resets the correlator latches and loads the integrating counters to their start value, then that value must be supplied in the menu field for PODs 4B and 4A. Illegal start values are:

1. 15 zeros (0000-hex or 8000-hex), which represents a zero integration time.
2. 15 ones (7FFF-hex or FFFF-hex), which represents an infinite integration time.

Apart from these two conditions there are 32766 permitted values. Some values, produced by simulating the integrating counter, are listed in Table 5.5.

A3.9. Pattern Generator - Program

The Program sub-menu of the DAS pattern generator is for entering the program instructions, and for selecting the output clock and strobe signals.

The pattern generator program is listed in Figure A3.3. The output clock in this case, is derived from the DAS internal master clock, and the clock period, specified by the field on the third line of the menu, is 1 μ s.

The Interrupt, Pause, and Inhibit signals are not used in the correlator test, and their inactive, default values are selected.

PATTERN GENERATOR: PROGRAM				INTERRUPT: CALL ON 1		
CLOCK: 105		MOS + 2,500		PAUSE ON: 1		INHIBIT ON: 1
SEQ	LABEL	POD4DC HEX	POD48A HEX	POD1C8 HEX	INSTRUCTIONS	STROBES
0	STFT	0000	0000	0204	REPEAT	30 012
1		0000	4000	0104	GOTO	PROG 012
2	OSR	0000	0000	010E		012
3		0000	0000	0104	REPEAT	30 012
4		0000	0000	0104	RETURN	012
5	MCR	0000	0000	0205	REPEAT	6 012
6		0000	0000	0204	REPEAT	30 012
7		0000	0000	0205	REPEAT	6 012
8		0000	0000	0204	REPEAT	6 012
9		0000	0000	0104	CALL	OSR 012
10		0000	0000	0204	REPEAT	30 012
11		0000	0000	0204	RETURN	012
12	SET	0000	0000	0134		012
13		0000	0000	0104	REPEAT	30 012
14		0000	0000	010C	REPEAT	30 012
15		0000	0000	011C		012
16		0000	0000	010C	REPEAT	30 012
17		0000	0000	0204	REPEAT	30 012
18		0000	0000	0204	RETURN	012
19	FMC	0000	4000	0104		012
20		0000	0000	0104		012
21		0000	0000	0004		012
22		0000	0000	0204	REPEAT	30 012
23		0000	0000	0104	RETURN	012
24	FOSR	0000	4000	0104		012
25		0000	0000	0104		012
26		0000	0000	1104		012
27		0000	0000	0104	REPEAT	30 012
28		0000	0000	0104	RETURN	012
29	HEAL	0000	4000	0104		012
30		0000	0000	0004		012
31		0000	0000	0104	RETURN	012
32	PL0	0000	34E4	1104		012
33	L0	0000	0000	0104	REPEAT	30 012
34		0000	0000	1104	GOTO	L0 012
35	PL1	0000	34E4	1108		012
36	L1	0000	0000	0108	REPEAT	30 012
37		0000	0000	1108	GOTO	L1 012
38	PL10	0000	0002	110C		012
39	L10	0000	0000	010C	REPEAT	30 012
40		0000	0000	110C	GOTO	L10 012
41	PL01	0000	0002	1108		012
42	L01	0000	0000	0108	REPEAT	30 012
43		0000	0000	1108	GOTO	L01 012
44	PROG	0000	0000	0104	CALL	OSR 012
45		0000	0000	0104	CALL	MCR 012
46		0000	0000	0104	CALL	SET 012
47		0000	0000	0104	CALL	FMC 012
48		0000	0000	0104	CALL	FOSR 012
49		0000	0000	0104	CALL	HEAL 012
50		7000	0000	0104	GOTO	PL0 012

Figure A3.3. Pattern Generator Program for Prototype IC Test.

The program consists of a sequence of in-line pattern generator code which is divided into seven columns. These are, from left to right in Figure A3.3:

SEQ: program sequence number. Each number corresponds to one program line. Altogether there are 245 sequences.

LABEL: up to four characters for labelling a specific program line, and up to 32 labels may be assigned. Labels are for use with GOTO, CALL, and INTERRUPT CALL instructions.

POD4DC: 16 data output lines; not used in this program

POD4BA: 16 data output lines; 15 lines are used to generate the integrating counters' start value (see Section A3.8).

POD1CB: 16 data output lines; 11 lines are used to generate the control signals and the input signals for the correlator chip (see also Section A3.8).

INSTRUCTIONS: program instructions for code compression, which include: CALL, GOTO, RETURN, REPEAT, HOLD, COUNT, and HALT. The REPEAT, HOLD, and COUNT instructions require numerical parameters; no more than six unique numerical parameters may be shared among them.

STROBES: the numbers in this columns' refer to the strobes as defined in the Timing sub-menu, which are to be asserted during the current program sequence. In this case STROBE 0 represents $\phi 1$ on chip; STROBE 1 represents $\phi 2$ on chip. Note that when the pattern generator is started any strobes asserted on SEQ 0 are ignored. These strobes will only be asserted if SEQ 0 is accessed again by a loop or call.

APPENDIX 4

AUTHOR'S PUBLICATIONS

VLSI DIGITAL POLARITY CORRELATOR BASED ON AN OVERLOADING COUNTER TECHNIQUE

Indexing terms: Integrated circuits, VLSI

A VLSI structure to implement a digital polarity correlator using an overloading integrating counter technique is reported. The implementation permits direct cascading of individual correlator chips without using additional circuits, to give complete flexibility in choice of correlator delay and resolution. The design considered offers significant performance advantages in high-speed correlation applications.

Introduction: Correlation is based on computation of the correlation function, $r_{yx}(\tau)$, where

$$r_{yx}(\tau) = \frac{1}{N} \sum_{k=0}^N y_k x_{k-\tau} \quad (1)$$

where y_k and x_k are analogue or digital sampled data sequences. Implementation of a high-speed correlator requires, therefore, an array of multipliers, delay elements and accumulators, either analogue or digital. Polarity correlation methods minimise the complexity of the computational elements by discarding the magnitude information of the input sequences. Digital design techniques can then be employed to realise the multipliers by EXNOR gates, the delay elements by a digital shift register and the accumulators by simple counting circuits. This results in a more economical and more compact implementation than would otherwise be achieved, the penalty for which is an increase in integration time to obtain a correlation function with acceptable variance.¹ The polarity correlation function is nonlinearly related to the (direct) correlation function, eqn. 1, by the Van Vleck arc sine relation² for input sequences which have Gaussian statistics.

Previously reported techniques for obtaining the polarity correlation function have included parallel counters,^{3,4} which are not directly cascadable and hence nonoptimal for VLSI implementation. This letter describes an interpretation of the polarity correlation function which permits the elimination of parallel counters and results in a highly regular correlator structure amenable to VLSI implementation. The structure also permits direct cascading of correlator stages. Details of a 18-stage prototype correlator chip based on this approach are included.

Polarity correlation: Polarity correlation is based on computation of the discrete function

$$r_{yx}(\tau) = \frac{1}{N} \sum_{k=0}^N (\text{sgn } [y_k] \text{sgn } [x_{k-\tau}]) \quad (2)$$

Complete positive correlation ($r_{yx} = 1$) occurs when the polarities of the input samples (assuming the mean of both inputs to be zero) are at all times equal, yielding an average product of +1. Complete negative correlation ($r_{yx} = -1$) occurs when the polarities of the input samples are never equal (inverse proportionality), yielding an average product of -1. In the case where the input samples are not related ($r_{yx} = 0$) the sum of the positive products will equal the sum of the negative products, and the average product will be zero.

Implementation of polarity correlation requires an analogue comparator circuit to convert $\text{sgn } [x] = x/|x|$ and $\text{sgn } [y] = y/|y|$ into logic 1 iff the signal is positive and logic 0 if the signal is negative. The time delay τ between the two signals is achieved by using a digital shift register where a particular value of delay is defined by the product of the number of preceding shift register stages and the sample clock period P . Multiplication is performed by the Boolean coincidence function EXNOR, whose output is 1 only if the inputs are both equal. If time-successive values of the coincidence function $F_k(\tau)$ are summed in a digital counting circuit for a period T seconds, where $T = NP$, then the contents of the counter at the end of the period will be proportional to the relevant value of the correlation function. The EXNOR func-

tion can only be regarded as performing multiplication if the logic 0 is allowed to represent -1. Thus, a logic 1 in the coincidence signal would indicate 'increment by one' the contents of the counter, and a logic 0 would indicate 'decrement by one' the contents of the counter. This would necessitate the use of up-down counters which are undesirable from a VLSI circuit design point of view. However, it is possible to use simple up-counters whose contents $q(\tau)$ can be related to the correlation function in the following way. First, the contents of an integrating counter are given by

$$q(\tau) = \sum_{k=0}^N F_k(\tau) \quad (3)$$

where $F_k(\tau)$ is the coincidence function bit stream defined by

$$F_k(\tau) = \frac{1}{2} + \frac{1}{2} \text{sgn } [y_k] \text{sgn } [x_{k-\tau}] = 1 \text{ or } 0 \quad (4)$$

Thus, by substituting into eqn. 3,

$$q(\tau) = \frac{N}{2} + \frac{N}{2} r_{yx}(\tau) \quad (5)$$

Hence,

$$r_{yx}(\tau) = 2 \frac{q(\tau)}{N} - 1 \quad (6)$$

where $r_{yx}(\tau)$ is the polarity correlation function as given by eqn. 2. Thus eqn. 6 gives a measure of the correlation function using the integration counter contents $q(\tau)$ after sampling N times. At maximum positive correlation ($r_{yx} = +1$) a maximum count $q(\tau) = N$ is obtained after sampling N times. In the case of maximum negative correlation ($r_{yx} = -1$), where the input samples are never equal, the coincidence signal is always zero, resulting in a zero count, $q(\tau) = 0$. In the case of zero correlation ($r_{yx} = 0$), a count of $q(\tau) = N/2$ is reached after sampling N times.

Overloading counter technique: An alternative approach to polarity correlation is based on an integrating overloading counter technique,⁵ which eliminates the requirement for a value of $q(\tau)$ to be at all times available. Instead, the correlation function is computed using the number of samples required to achieve count conditions, $q(\tau) = N$, in a given integrating counter. The concept of the technique is illustrated by Fig. 1, which shows the relationship between the contents of

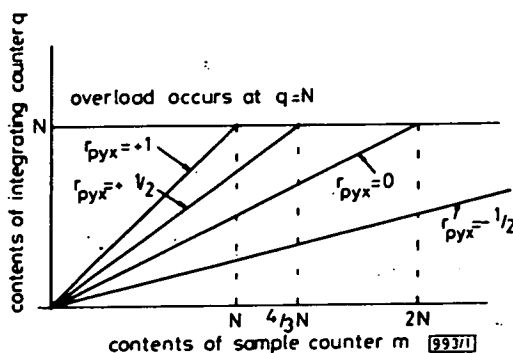


Fig. 1 Relationship between an integrating counter overload and the contents of the sample counter

an integrating counter $q(\tau)$ and the number of samples, which is now a variable m . The number of samples m can be related to the polarity correlation function by writing $q(\tau)$ as

$$\begin{aligned} q(\tau) &= N = \sum_{k=0}^m \left(\frac{1}{2} + \frac{1}{2} \text{sgn } [y_k] \text{sgn } [x_{k-\tau}] \right) \\ &= \frac{m}{2} + \frac{m}{2} r_{yx}(\tau) \end{aligned} \quad (7)$$

where

$$r_{yx}(\tau) = \frac{1}{m} \sum_{k=0}^m (\text{sgn } [y_k] \text{sgn } [x_{k-\tau}]) \quad (8)$$

ence, in this case,

$$r_{p_{yx}}(\tau) = 2 \frac{N}{m} - 1 \quad \text{for } m \geq N \quad (9)$$

where N is the capacity of the integrating counter and m is the number of samples required to achieve overload conditions in the integrating counter corresponding to time delay τ . An overload occurs after $m = N$ samples when correlation is maximum and positive. In the case of zero correlation an overload occurs after $m = 2N$ samples and after an infinite number of samples when the correlation is maximum and negative. Note that an overload cannot occur until $m \geq N$.

A polarity correlator using the overloading counter technique thus comprises a delaying shift register connected to a parallel array of coincidence detectors and integrating counters. An overload pattern shift register is used to inspect the overload condition of the counters. The evolving pattern of overload states defines the correlation function shape and the time-delay position of the first integrating counter to overload defines the position of the most significant peak of the function. A sample counter is included to count the number of input samples m so that the value of the correlation function may be computed easily for any integrating counter to overload. If the maximum capacity of the sample counter is set to be twice the capacity of the integrating counters the significance range is limited to $1 \geq r \geq 0$. If it is required to cover the range $1 \geq r \geq -1$, two correlator circuits working in parallel can be used with one covering the positive range and the other covering the negative range.

Such a system is most suitably realised using integrated-circuit technology and an early device implemented 12 stages of correlation using p -MOS technology. Fig. 2 shows the

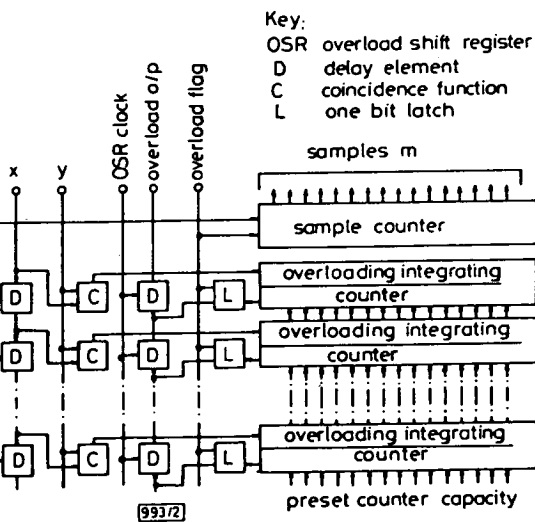


Fig. 2 Layout diagram of polarity correlator using the overloading integrating counter technique

block diagram of a polarity correlator with additional control circuitry to realise a technique for displaying the correlation function and to provide built-in self test and self repair. The built-in self test and self repair mechanism automatically detects and eliminates failed channels in the VLSI circuit. The failed channels are short-circuited to maintain a series connection of correctly operating channels. Design parameters of a prototype chip, containing 28 parallel stages of correlation and fabricated on a $5 \mu\text{m}$ n -channel MOS process are 4 MHz sample rate with integration time programmable to a maximum of 2^{15} samples. The architecture shown allows direct cascading of chips, without using additional components, to give correlation delays of arbitrary length. Sample rates up to 40 MHz with up to 512 parallel stages of correlation per chip can be expected from available VLSI fabrication processes.

Conclusions: A VLSI structure has been described which offers an attractive digital implementation of a high-speed polarity correlator. Individual chips may be directly cascaded to realise a correlator with arbitrary resolution or delay, in contrast to other digital correlator circuits, particularly those using the parallel counter technique, which cannot be easily cascaded and do not render regular VLSI structures. Furthermore the operating speed of parallel counter based correlators is limited by carry signal propagation delays whereas the correlator described here, which is composed mainly of simple shift register stages can approach the maximum clocking rate of a chosen VLSI fabrication process. The architecture described, through built-in self test and self repair techniques, offers enhanced production yield and in-service reliability.

Acknowledgments: This work was carried out under a UK Science & Engineering Research Council grant.

W. S. BLACKLEY

M. A. JACK

J. R. JORDAN

20th July 1983

Department of Electrical Engineering
 University of Edinburgh, King's Buildings
 Mayfield Road, Edinburgh EH9 3JL, Scotland

References

- 1 HAYES, A. M., and MUSGRAVE, G.: 'Correlator design for flow measurement', *Radio & Electron. Eng.*, 1973, 43, pp. 363-368
- 2 VAN VLECK, J. H., and MIDDLETON, D.: 'The spectrum of clipped noise', *Proc. IEEE*, 1966, 54, pp. 2-19
- 3 SWARTZLANDER, E. E.: 'Parallel counters', *IEEE Trans.*, 1973, C-22, pp. 1021-1024
- 4 DADDA, L.: 'Composite parallel counters', *ibid.*, 1980, C-29, pp. 942-946
- 5 JORDAN, J. R., and BECK, M. S.: 'Correlation function display and peak detection', *Electron. Lett.*, 1972, 24, pp. 602-604

AGARD

ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT

7 RUE ANCELLE 92200 NEUILLY SUR SEINE FRANCE

**Paper Reprinted from
Conference Proceedings No.361**

**DESIGN FOR TACTICAL AVIONICS
MAINTAINABILITY**

NORTH ATLANTIC TREATY ORGANIZATION



BUILT-IN TEST AND SELF REPAIR MECHANISMS IN A DIGITAL CORRELATOR INTEGRATED CIRCUIT

by

W.S. Blackley, M.A. Jack, J.R. Jordan

Department of Electrical Engineering

University of Edinburgh

King's Buildings

Mayfield Road

Edinburgh

Scotland

EH9 3JL

SUMMARY

A VLSI digital correlator architecture which incorporates built-in self test and self repair mechanisms is described. The architecture offers testability and reliability, and the overhead for the test and repair circuitry is only one latch and two multiplexers per correlator stage. The correlator has been fabricated on a 5-micron nMOS process and results from the first batch of processed chips are reported.

INTRODUCTION

The advantages in terms of increased complexity, improved performance, reduced costs and new systems applications made available as silicon integrated circuit technology matures from the level of large scale integration (LSI) to very large scale integration (VLSI) have been widely recognised. However, one important facet of integrated circuit technology which lags dangerously behind the complexity potential of VLSI, is the problem of establishing the integrity of the VLSI design in terms of initial design validation, manufacturing quality and longer term operational reliability [1,2].

This paper addresses the need to embody a testability scheme within the VLSI integrated circuit itself and presents details of a digital polarity correlator architecture with built-in self test (and self-repair) mechanisms. The concept is demonstrated using results obtained from a prototype integrated circuit chip which has been fabricated in 5-micron enhancement/depletion n-channel MOS technology.

Correlation techniques are widely used in communications, instrumentation, computers, telemetry, sonar, radar, medical and other signal processing systems [3,4,5]. The desirable properties of correlation include the ability to detect a desired signal in the presence of noise or other signals; the ability to recognise specific patterns, and the ability to measure time delays through various media.

Electronic systems for computation of the correlation function have been available for many years, but they have been large and inefficient. With the development of VLSI, correlation can be performed efficiently now, with a minimal number of components.

The correlator chip presented here, consists of a linear cascade of identical correlation elements. The performance of the correlator depends on the serial connection of correctly functioning correlation elements. To optimise the performance and gain full advantage of the VLSI architecture a design strategy was adopted which includes testability, yield enhancement, and reliability improvement.

TEST STRATEGY REQUIREMENTS IN VLSI DESIGN

A VLSI test strategy must ideally allow for a range of differing test environments to be experienced by the circuit during its operational service. These environments can be summarised as:

- a) prototype characterisation; to include design validation and parametric testing.

- b) production test; to include yield enhancement features.
- c) service or maintenance test; to include self-repair features.

In prototype characterisation it is essential to identify and localise individual faults to enable fault diagnosis and correction. Prototype faults may be process-related faults statistically distributed over a processed wafer, or they may be design faults (errors) such as forgotten contact holes, wrong interconnections or excessive signal delays. Prototype testing is invariably carried out by the designer(s) using automatic test equipment (ATE), microprobing or electron beam facilities.

Production test requirements include both process quality checks and functional checks. Process quality control is achieved either by a number of chip-size "drop-replacements" spaced over the wafer or by using a small test area on each chip. Measures of transistor parameters, contact resistance and capacitance values are made to check production tolerances. In production test, functional (and parametric) tests must be minimised since here testing time and costs are important. Functional tests need only yield a limited number of the significant internal states since it is not generally possible to redesign or repair at this stage.

In maintenance and systems test, fault diagnosis is precluded so a simple YES/NO-GO indication for the circuit is adequate.

The correlator architecture considered here incorporates design for test which offers the potential of valid use at each stage in the life of a VLSI circuit. To appreciate the ease with which this architecture has been adapted to perform self test and self repair, the concept of polarity correlation and its silicon realisation must be discussed.

POLARITY CORRELATION

Polarity correlation is based on the computation of the discrete function,

$$r_{pyx}(\tau) = \frac{1}{N} \sum_{k=0}^N (\text{sgn}[y_k] \cdot \text{sgn}[x_{k-\tau}]) \quad (1)$$

where $r(\tau)$ is the value of the correlation function between two signals, x and y . Complete positive correlation occurs when the polarities of the input samples (assuming the mean of both inputs to be zero) are at all times equal, yielding an average product of +1. Complete negative correlation occurs when the polarities of the input samples are never equal (inverse proportionality), yielding an average product of -1. In the case where the input samples are not related, the sum of the positive products will equal the sum of the negative products and the average product will be zero.

Implementation of polarity correlation requires an analogue comparator circuit to convert $\text{sgn}[x]=x/|x|$ and $\text{sgn}[y]=y/|y|$ into logic 1 if the signal is positive and logic 0 if the signal is negative. The time delay τ between the two signals is achieved by using a digital shift register where a particular value of delay is defined by the product of the number of preceding shift register stages and the sample clock period, P . Multiplication is performed by the Boolean coincidence function, EXNOR, whose output is 1 only if the inputs are both equal. If time-excessive values of the coincidence function are summed in a digital counting circuit for a period T seconds, where $T = NP$, then the contents of the counter at the end of the period will be proportional to the relevant value of the correlation function.

Polarity correlation methods minimise the complexity of the computational elements by discarding the magnitude information of the input sequences. Digital design techniques can then be employed to realise a more economical and more compact implementation than would otherwise be achieved, the penalty for which is an increase in integration time to obtain a correlation function with acceptable variance [6]. The polarity correlation function is nonlinearly related to the (direct)

correlation function by the Van Vleck arc sine relation [7] for input sequences which have Gaussian statistics.

Previously reported techniques [8,9] for obtaining the polarity correlation function have included parallel counters [10,11] which are not directly cascadable and hence non-optimal for VLSI implementation. This paper describes an interpretation of the polarity correlation function which permits the elimination of parallel counters and results in a highly regular correlator structure amenable to VLSI implementation. As a consequence direct cascading of correlator stages to any arbitrary level is possible.

The structure is based on an integrating overloading counter technique [12,13], in which the correlation function is computed using the number of input samples taken to achieve overload count conditions, in a given integrating counter. An overload flag bit for each counter is used instead of the counter contents. This reduces the complexity of the structure to bit-serial input and output. The number of input samples, m , can be related to the polarity correlation function by [14]

$$r_{pyx}(\tau) = 2\frac{N}{m} - 1 \quad \text{for } m \geq N \quad (2)$$

where N is the capacity of the integrating counter and m is the number of samples required to achieve overload conditions in the integrating counter corresponding to time delay τ . An overload occurs after $m = N$ samples when correlation is maximum and positive. In the case of zero correlation an overload occurs after $m = 2N$ samples and after an infinite number of samples when the correlation is maximum and negative. Note that an overload cannot occur until $m \geq N$.

A polarity correlator using the overloading counter technique is shown in Figure 1. It comprises a delaying shift register connected to a parallel array of coincidence detectors and integrating counters. An overload pattern shift register is used to inspect the overload condition of the counters. The evolving pattern of overload states defines the correlation function shape and the time delay position of the first integrating counter to overload defines the position of the most significant peak of the function. A sample counter is included to count the number of input samples, m , so that the value of the correlation function may be computed for any integrating counter to overload. If the maximum capacity of the sample counter is set to be twice the capacity of the integrating counters the significance range is limited to $1 \geq r \geq 0$. If it is required to cover the range $1 \geq r \geq -1$, two correlator circuits working in parallel can be used with one covering the positive range and the other covering the negative range.

CORRELATOR ARCHITECTURE FOR SELF TEST AND SELF REPAIR

The VLSI architecture considered here consists of a long series connection of identical correlation stages. If any one of these stages suffers faults during manufacture or becomes faulty during service then complete chip failure will be experienced. A self-test and self-repair structure has been devised to overcome this problem. The self-test sequence is initiated each time the chip is switched-on and any faulty stages discovered as a result of these tests will be automatically bypassed so that the working stages are reconfigured to form a continuous serial connection. Faults developing during the working life of the chip will thus be automatically eliminated every time the chip is switched on. The self-test control circuit must offer high reliability and therefore employs redundant circuit techniques, however assuming fault conditions to be evenly distributed over the chip area it can be expected that the majority of faults will be experienced in the large area taken by the integrating counters. Using these self-test and repair strategies, an overall manufacturing yield of good working chips is enhanced and longer working life can be expected.

The principal additions to the basic correlator stage of Figure 2(a) to allow it to perform built-in self-test and self-repair are shown in Figure 2(b). The delay shift register (DSR) and the overload pattern shift register (OSR) each have a 2 to 1 multiplexer added and a multiplexer control register (MCR) has been included to store the control information for these multiplexers. Full functional testing is

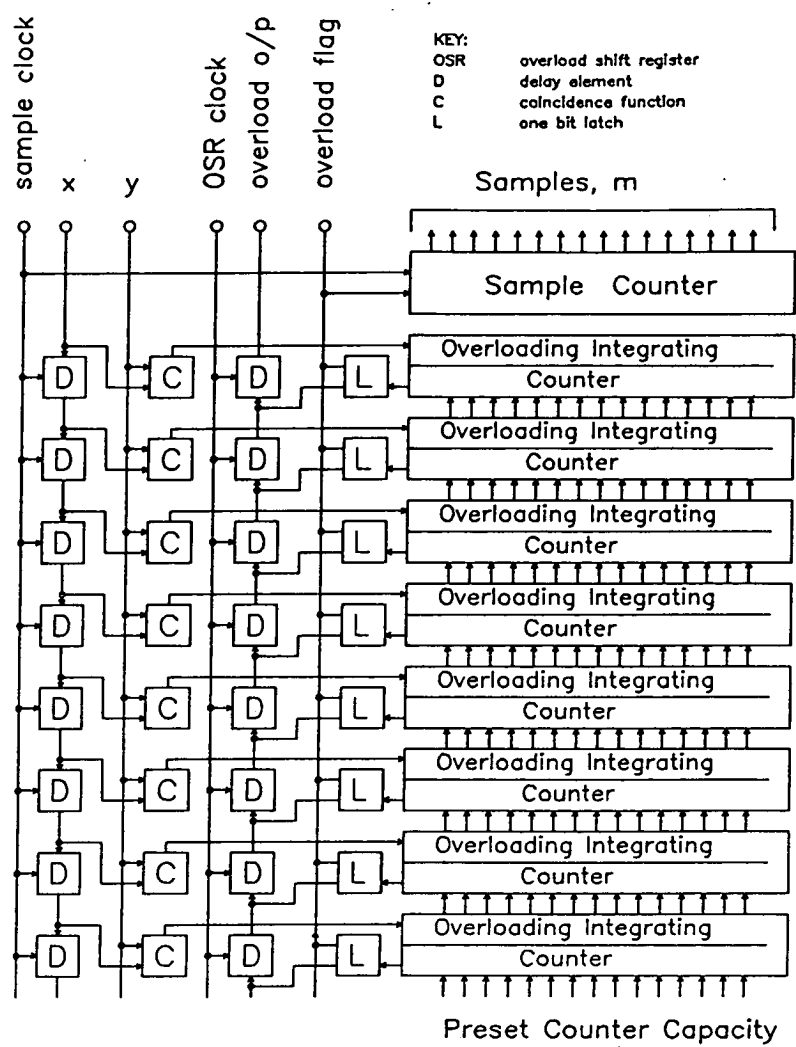


Figure 1. Layout diagram of digital polarity correlator using the overloading integrating counter technique.

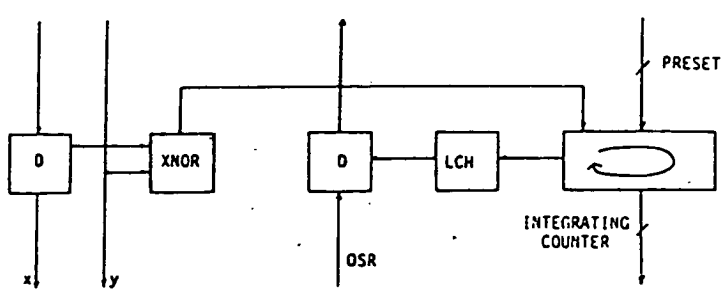


Figure 2*(a). Basic correlator stage.

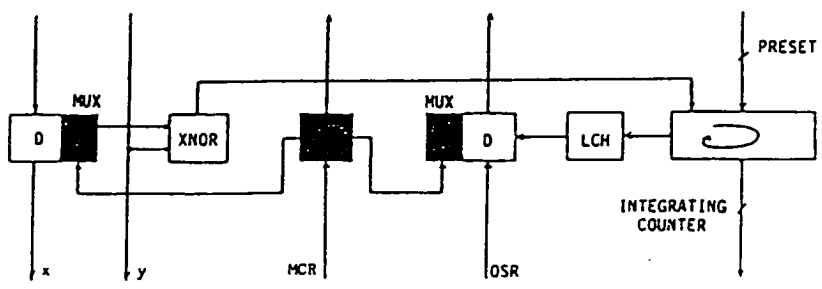


Figure 2 (b). Correlator stage with built-in self- test and self-repair mechanisms.

possible due to the extent of the link between design and test. A large degree of circuit partitioning is incorporated in the design and this, coupled with the DSR, OSR and MCR shift registers acting as scan paths [15] allows all the internal states to be controlled and observed.

The key feature in the self repair mechanism is the Multiplexer Control Register (MCR) which, after the self-test sequence, contains the pass/fail status for each stage. A circuit schematic of the MCR and one multiplexer is shown in Figure 3. In the case of a failure the input and output registers of the correlator stage are bypassed using the multiplexers, thereby short circuiting the malfunctioning stage. The number of functioning stages on the chip can be read out serially from the MCR by reconfiguring it as a shift register. This parameter represents the maximum attainable correlation delay and can be used for chip reject/accept decisions in production test. The self-test and repair sequence may be repeated as required during the service life of the chip.

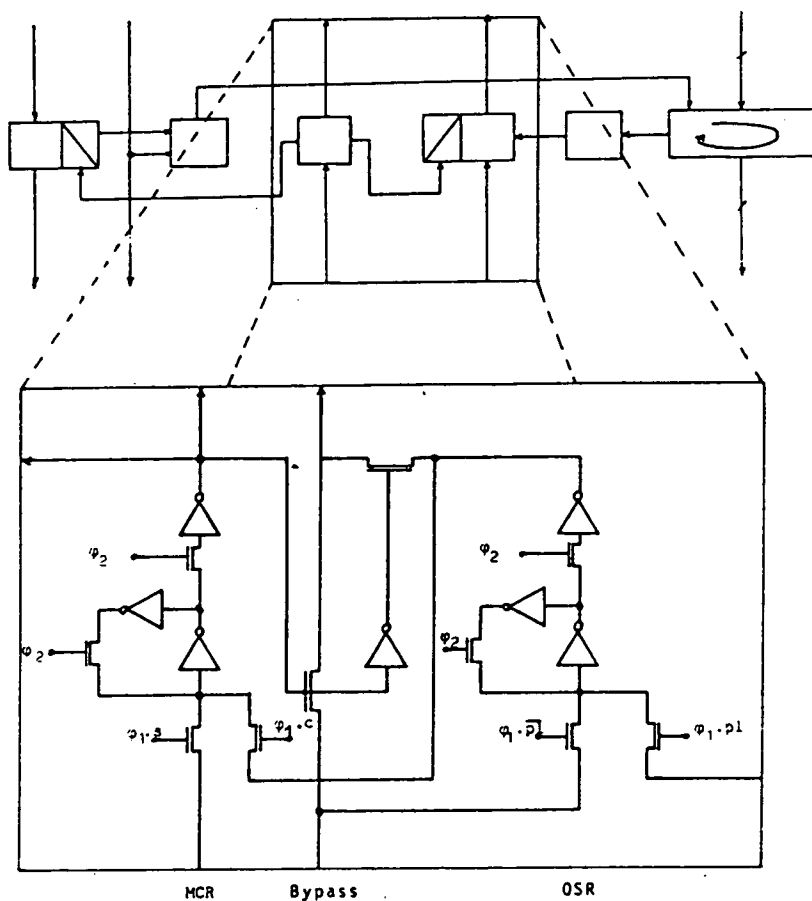


Figure 3. Zoom in on floor plan: Bypass loop and multiplexer control register.

TEST STRATEGY

The correlator operates in three distinct modes: initial test, self-test and repair, and run. During the initial test period three simple tests are carried out on the most basic elements of the design, namely the scan path registers. These registers (DSR, MCR and OSR) and their various control functions are tested to check that a chip is acceptable immediately after fabrication. The initial test sequence is as follows:

- 1) Test DSR, OSR and MCR as shift registers and measure their delay.
- 2) Test the effect of the MCR on the DSR and OSR registers. This is done by shifting n 'ones' into the MCR and then measuring the delay of the DSR and OSR registers, which should each be reduced by n .
- 3) Test the parallel load facilities of the DSR, OSR and MCR registers.

The self-test period is where the chip effectively tests itself and reconfigures its registers so that all of the working stages are connected in series. In this test the following sequence is repeated four times according to the possible combinations of the two binary input signals, X and Y.

- 1) Reset Latches and Integrating Counters. The counters are loaded with 4000-hex, a number corresponding to the maximum integration time of $2^{15} - 2 = 32766$ sample clock cycles.
- 2) Set up the input conditions (X and Y) by setting or clearing the DSR register as required. Shift X and Y through correlator for 32766 clock cycles.
- 3) Parallel load Latches into OSR. The overload pattern may be shifted out for observation.

The self repair sequence follows the self test sequence. During the self test sequence the overload signal is compared with the expected value of overload signal and any deviations form the expected signal results in a logic 1 stored in the corresponding Latch. Thus, when the self test sequence has finished the logic 1's and 0's stored in the Latches are the results of the self test, where a logic 1 indicates a faulty stage. The self repair operation essentially transfers this information to the Multiplexer Control Register which in turn causes the faulty stages to be bypassed. The net effect is a series connection of correctly operating correlation stages.

The run period follows automatically after the self-test and repair sequence is completed. Note that after the test the contents of the MCR may be inspected to ensure that enough of the correlator stages are working to satisfy the requirements of the system into which the chip is to be installed.

PROTOTYPE DESIGN

A prototype digital correlator featuring self-test and self-repair has been fabricated on a 5 micron n-channel MOS process. The prototype design contains 28 parallel stages of correlation, each of which implements the block diagram of Figure (b). The area of the chip is 5.08mm by 5.08mm.

The layout of the two parallel stages of correlation is shown, annotated, in Figure 4. Each stage is composed of cells which may be repeated by abutting in the x-direction. The largest cell is the presettable PRBS counter which has 15 shift register stages, and thus a maximum count of approximately 32K samples. The layout of the presettable counter is in the form of a ring in order to minimise the circuit delays between each shift register stage. The correlator design is semi-static throughout. This means that the clock frequency and thus the sampling frequency of the correlator can range from d.c. to 4MHz. (for this fabrication process). From Figure 4 it may be seen that the presettable counter occupies most of the active area of the chip. Also shown is the area taken up by the self-test and repair circuitry. The overhead for self-test and self-repair is approximately 6%.

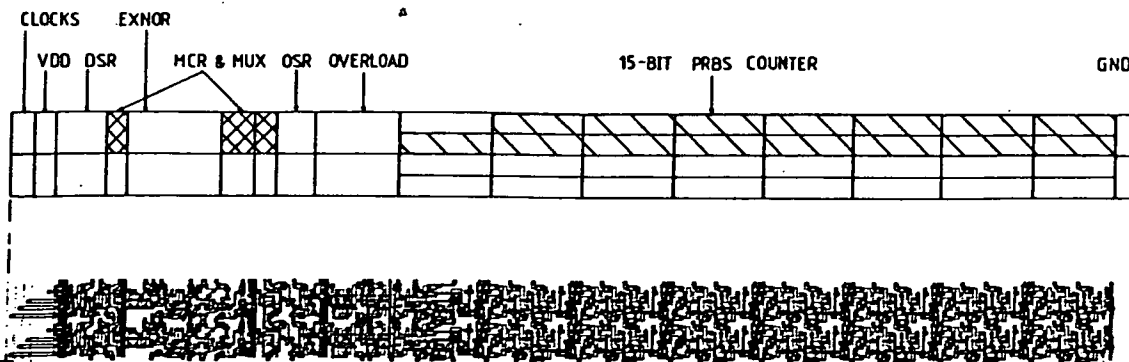
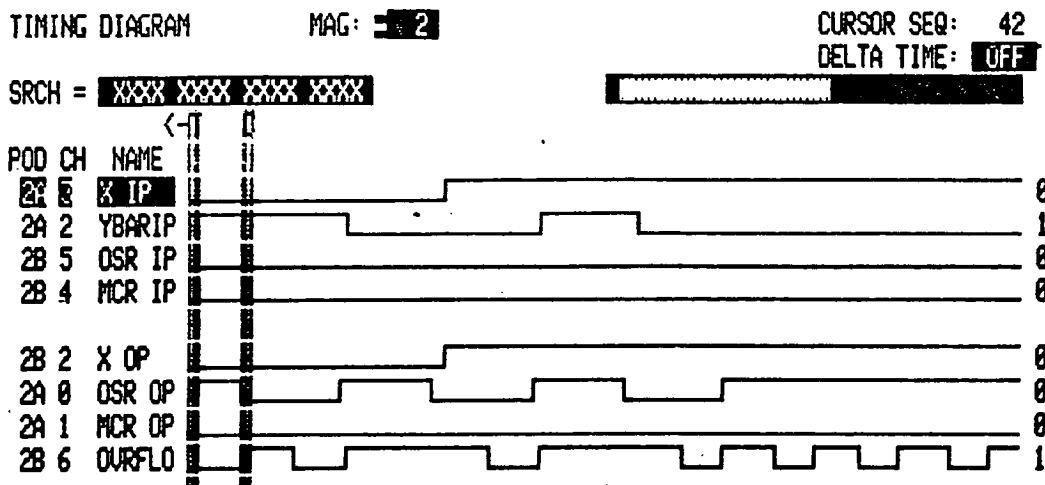


Figure 4. nMOS layout of 2 stages of correlation.

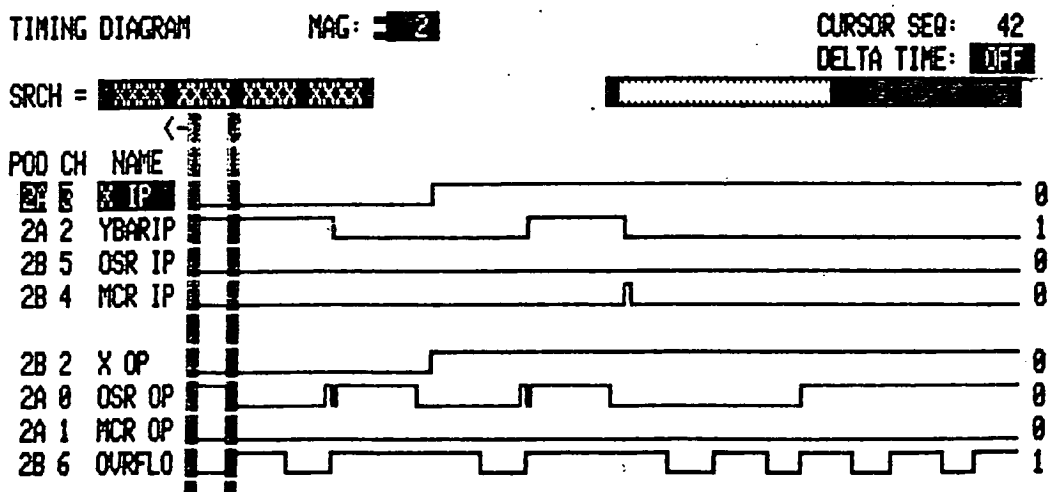
TEST RESULTS

The correlator chip has been functionally tested using a Tektronix DAS 9100 Digital Analysis System coupled to a Teledyne Probe Station. Initially 10 packaged chips, which had passed a visual inspection, were functionally tested. However many more samples were required to demonstrate the yield enhancement capability of this design so the remaining wafers were probe-tested. Unfortunately, only 130 candidates were available for testing since the chip was fabricated as part of a multi-project wafer. More wafers are however, to be processed.

Figure 5(a) and (b) show some of the input and output waveforms from two correlator chips, that have occurred during the self test and repair period. For display purposes the integration time of the correlator has been reduced to just 15 clock cycles. Figure 5(a) shows the correlation output of a 'golden chip', while Figure 5(b) shows the output of a chip which has one failed stage. The top four traces in each figure represent the inputs to the device. In each figure the X and Y inputs sequence through their four possible combinations in accordance with the test strategy described above. For clarity, the control signals which cause, for example, parallel load OSR, or reset counters, have not been shown.



(a).



(b).

Figure 5. Test results from two correlator chips. A 'golden chip' (a) and a chip with one faulty stage (b).

The significant points to note in Figure 5 are the Multiplexer Control Register input (MCR IP) and the Overload Shift Register output (OSR OP). All the other shown signals are the same for both chips. With reference to Figure 5 and moving left to right from the cursor, the overload output (OVRFLO) has changed from logic 1 to 0. This indicates that at least one of the integrating counters has overloaded after the prescribed period of 15 clock cycles (see above). This result is expected since the inputs have been equal ($X=0$, $Y=0$) over this period.

When OVRFLO next goes high, the correlator has been reset and the next correlation test ($X=0$, $Y=1$) is begun. Also at this time, the overload pattern, i.e. the contents of the Latches are transferred to the OSR and shifted out for display. Now we can see the difference between the 'golden chip', Figure 5(a) and the faulty chip, Figure 6(b). The OSR should contain a series of 28 logic 1's and in Figure 6(b) there is a logic 0 in position number 27, indicating a fault in stage 27. The correlation test is repeated for the remaining combinations of X and Y, and the fault is again exposed on the OSR output in the case where $X = Y = 1$.

Self repair is then carried out on the faulty chip. A single logic 1 is shifted into bit position 27 of the MCR which causes stage 27 to be bypassed. The correlation test, with $X=Y=1$ is repeated several times at a period of 27 rather than 28 and the incorrect logic 0 on the OSR output has been eliminated. The result is a 'golden chip' containing 27 stages of correlation.

YIELD ENHANCEMENT

This section contains the results of the first 130 processed chips. The results are preliminary and the sample is small. Figure 6 shows a chart of Number of Chips plotted against number of working stages. It shows that 29 of the 130 candidates passed the initial test and that 27 of these yielded more than 20 stages of correlation.

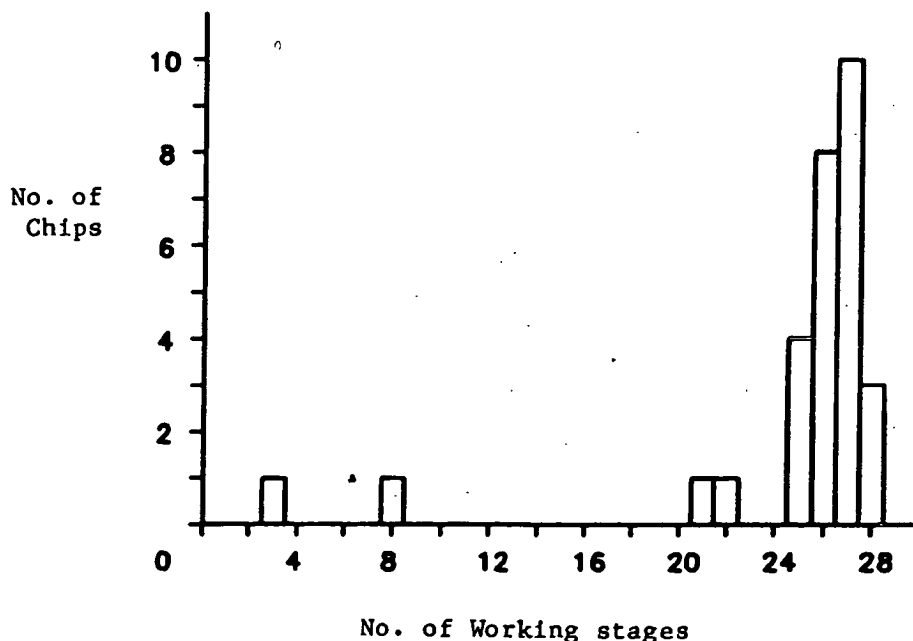


Figure 6. No. chips vs. No. working stages.

Listed below are the test results for each wafer. The multi-project wafers each contained 24 correlator chips.

	Without Self-Repair (i.e. 28 stages working)	With Self-Repair (i.e. >20 stages working)
Packaged (10 candidates)	0	2
Wafer #1 (24 candidates)	1	5
Wafer #2 (24 candidates)	0	5
Wafer #3 (24 candidates)	0	6
Wafer #4 (24 candidates)	0	0
Wafer #5 (24 candidates)	2	9
	=====	=====
TOTALS (130 candidates)	3	27
YIELD with no yield enhancement: 2.3%		
YIELD with yield enhancement: 20.7%		

CONCLUSIONS

A digital polarity correlator architecture which incorporates all of the required features of a built-in self-test and repair strategy has been described. The test strategy will carry the design through all of the varying test requirements to be encountered by the chip. Incorporating extra stages of correlation on-chip permits the use of self-repair mechanisms for enhanced production yield and in-service reliability.

The VLSI structure offers an attractive digital implementation of a high speed polarity correlator. Individual chips may be directly cascaded to realise a correlator with arbitrary resolution or delay, in contrast to other digital correlator circuits, particularly those using the parallel counter technique, which cannot be easily cascaded and do not render regular VLSI structures. Furthermore, the operating speed of parallel counter based correlators is limited by carry signal propagation delays whereas the correlator described here, which is composed mainly of simple shift register stages can approach the maximum clocking rate of a chosen VLSI fabrication process.

The results from the functional testing of the first batch of processed chips have been reported. They demonstrate that a considerable improvement in yield can be obtained at a very low circuit overhead. A yield enhancement factor of 9.0 has been obtained for the initial sample of 130 chips. In addition this chip can be given a exhaustive functional test in less than 150ms at 1MHz.

References

1. T. W. Williams, "Design for Testability: What's the Motivation?," VLSI Design, pp. 21-23 (October 1983).
2. E. B. Eichelberger and E. Lindbloom, "Trends in VLSI Testing," pp. 339-348 in VLSI 83, ed. F. Anceau and E. J. Aas, ELsevier Science Publishers B. V. (North Holland) (1983).
3. J. S. Bendat and A. G. Piersol, Engineering Applications of Correlation and Spectral Analysis, John Wiley and Sons, Chichester (1980). A Wiley Interscience Publication
4. Y. W. Lee, T. P. Cheatham Jr., and J. B. Wiesner, "Application of Correlation Analysis to the Detection of Periodic Signals in Noise," Proc. IRE, Vol. 38, pp. 1165-1171 (October 1950).

- . J. E. Tanner and C. Mead, "A Correlating Optical Motion Detector," Proc. Conf. on Advanced Res. in VLSI, MIT, Cambridge, MA, pp. 57-64 (January 1984).
- . A. M. Hayes and G. Musgrave, "Correlator design for flow measurement," The Radio & Electronic Engineer, Vol. 43, pp. 363-368 (June 1973).
- . J. H. Van Vleck and D. Middleton, "The Spectrum of Clipped Noise," Proc. IEEE, Vol. 54, pp. 2-19 (January 1966).
- . J. Eldon, "Correlation - A Powerful Technique for Digital Signal Processing," Application Notes, TRW LSI Products, California, Vol. TP-17, pp. 1-22 (1981).
- . K. W. Current, "A High Data-Rate Digital Output Correlator Design," IEEE Trans. Comput., Vol. C-29, pp. 403-405 (May 1980).
0. E. E. Swartzlander Jr., "Parallel Counters," IEEE Trans. Comput., Vol. C-22, pp. 1021-1024 (November 1973).
1. L. Dadda, "Composite Parallel Counters," IEEE Trans. Comput., Vol. C-29, pp. 942-946 (October 1980).
2. J. R. Jordan and M. S. Beck, "Correlation Function Display and Peak Detection," Electron. Lett., Vol. 8, pp. 602-604 (November 1972).
3. W. S. Blackley, M. A. Jack, and J. R. Jordan, "Digital Polarity Correlator," UK Patent Application Nos. 8306797 and 8300699 (11th March 1983).
4. W. S. Blackley, M. A. Jack, and J. R. Jordan, "VLSI Digital Polarity Correlator Based on an Overloading Counter Technique," Electron. Lett., Vol. 19, pp. 761-762 (September 1983).
5. T. W. Williams and K. P. Parker, "Design for Testability - A Survey," IEEE Trans. Comput., Vol. C-31, pp. 2-15 (January 1982).

ACKNOWLEDGEMENTS

This work was carried out under a UK Science & Engineering Council Research Council grant.

This chip's test and repair overhead is only one latch and two multiplexers per correlator stage. Yield on the first batch processed was enhanced nine to one.

A Digital Polarity Correlator with Built-in Self Test and Self Repair

William S. Blackley, Mervyn A. Jack, and James R. Jordan
University of Edinburgh

The maturing of silicon integrated circuit technology from large-scale to very large scale integration has improved performance, reduced costs, and opened new systems applications. However, one important facet of integrated circuit technology lags dangerously behind the complexity potential of VLSI: establishing the integrity of the VLSI design in terms of initial design validation, manufacturing quality, and long-term operational reliability.^{1,2}

This article addresses the need to embody a testability scheme within the VLSI integrated circuit itself. It also presents details of a digital polarity

correlator architecture with built-in self-test and self-repair mechanisms. Results obtained from a prototype integrated circuit chip fabricated in five-micron enhancement/depletion N-channel MOS technology demonstrate the concept.

Correlation techniques are widely used in communications, instrumentation, computers, telemetry, sonar, radar, medical, and other signal processing systems.³⁻⁵ Desirable correlation properties include the ability to detect a desired signal in the presence of noise or other signals, to recognize specific patterns, and to measure time delays through various media.

Electronic systems for computation of the correlation function have been available for many years, but they have been large and inefficient. With the development of VLSI, correlation can be performed efficiently and with fewer components.

Our correlator chip consists of a linear cascade of identical correlation elements. The performance of the correlator depends on the serial connection of correctly functioning correlation elements. To optimize performance and gain full advantage of the VLSI architecture, we adopted a design strategy that includes testability, enhances yield, and improves reliability.

An earlier version of this article appeared in the *International Test Conference Proceedings*, October 1983.

Summary

Correlation techniques are widely used in communications, instrumentation, computers, telemetry, and other signal processing systems to detect a desired signal in the presence of noise, to recognize patterns, and to measure time delays. With the development of VLSI, correlation can be performed efficiently with a minimal number of components.

The correlator chip presented in this article consists of a linear cascade of identical elements; failure of any one element causes complete chip failure. Therefore, we devised a self-test and self-repair structure to automatically bypass faulty stages.

The overhead for self test and self repair was approximately six percent of the chip area. The results of functional testing of the first batch of processed chips demonstrated a nine-to-one yield enhancement and an exhaustive functional test time of less than 150 milliseconds. The self-repair mechanism provides high in-service reliability.

Test strategy requirements VLSI design

Ideally, a VLSI test strategy allows the circuit to experience a range of test environments during its operational service. In summary, these environments are

- prototype characterization, to include design validation and parametric testing;
- production test, to include yield-enhancement features; and
- service or maintenance test, to include self-repair features.

In prototype characterization, it is essential to identify and localize individual faults to enable fault diagnosis and correction. Prototype faults can be process-related and statistically distributed over a processed wafer, or they can be design faults (errors), such as omitted contact holes, wrong interconnections, or excessive signal delays. Prototype testing is invariably carried out with automatic test equipment, microprobing, or electron beam facilities.

Production test requirements include both process quality checks and functional checks. Process quality control is achieved by means of a number of chip-size, drop-in replacements spaced over the wafer or by dedicating a small area on each chip to testing.

Transistor parameters, contact resistance, and capacitance values are measured in order to check production tolerances. In production test, functional (and parametric) tests must be minimized, since testing time and costs are important. Functional tests need only yield a limited number of the significant internal states, since it is not generally possible to redesign or repair at this stage.

In maintenance and systems test, fault diagnosis is precluded; a simple GO/NO-GO indication for the circuit is adequate.

The correlator architecture considered here incorporates a design for test with the potential for valid use at each stage in the life of a VLSI circuit. The ease with which this architecture has been adapted to perform self test and self repair can only

be discussed within the context of polarity correlation and its silicon realization.

Polarity correlation

Polarity correlation is based on the computation of the discrete function

$$r_{pyx}(\tau) = \frac{1}{N} \sum_{k=0}^N (\text{sgn}[y_k] \cdot \text{sgn}[x_{k-\tau}]) \quad (1)$$

where $r(\tau)$ is the value of the correlation function between two signals, x and y . $\text{Sgn}[x]$ means $\text{signum}[x]$, a function of the value $+1$ for positive x and -1 for negative x . Complete positive correlation occurs when the polarities of the input samples (assuming the mean of both inputs to be zero) are at all times equal, yielding an average product of $+1$. Complete negative correlation occurs when the polarities of the input samples are never equal (inverse proportionality), yielding an average product of -1 . In the case where the input samples are not related, the sum of the positive product will equal the sum of the negative products, and the average product will be zero.

Implementation of polarity correlation requires an analog comparator circuit to convert $\text{sgn}[x] = x/|x|$ and $\text{sgn}[y] = y/|y|$ into logic 1 if the signal is positive and logic 0 if the signal is negative. The time delay τ between the two signals is achieved by using a digital shift register in which the product of the number of preceding shift register stages and the sample clock period P define a particular value of delay.

Multiplication is performed by the Boolean coincidence function, EX-NOR, whose output is 1 only if the inputs are equal. If time-successive values of the coincidence function are summed in a digital counting circuit for a period T seconds, where $T = NP$, then the contents of the counter at the end of the period will be proportional to the relevant value of the correlation function.

Polarity correlation methods minimize the complexity of the computational elements by discarding the

magnitude information of the input sequences. Digital design techniques can then be employed to realize a more economical and compact implementation than could otherwise be achieved. The penalty is the increased integration time needed to obtain a correlation function with acceptable variance.⁶ The polarity correlation function is nonlinearly related to the (direct) correlation function by the Van Vleck arc sine relation⁷ for input sequences with Gaussian statistics.

Previously reported techniques^{8,9} for obtaining the polarity correlation function have included parallel counters,^{10,11} which are not directly cascable and hence nonoptimal for VLSI implementation. Our interpretation of the polarity correlation function permits elimination of parallel counters and results in a highly regular cor-

**Our interpretation of the
polarity correlation function
permits elimination of parallel
counters and results in a
highly regular correlator
structure amenable to VLSI
implementation.**

relator structure amenable to VLSI implementation. As a consequence, correlator stages can be directly cascaded to any arbitrary level.

The structure is based on an integrating overloading counter technique^{12,13} in which the correlation function is computed by using the number of input samples needed to reach overload count conditions in a given integrating counter. Under such conditions, an overload flag bit for each counter is used instead of the counter contents. This reduces the complexity of the structure to bit-serial input and output. The number of input samples m can be related to the polarity correlation function by

$$r_{pyx}(\tau) = 2 \frac{N}{m} - 1 \quad \text{for } m \geq N \quad (2)$$

where N is the capacity of the integrating counter and m is the number

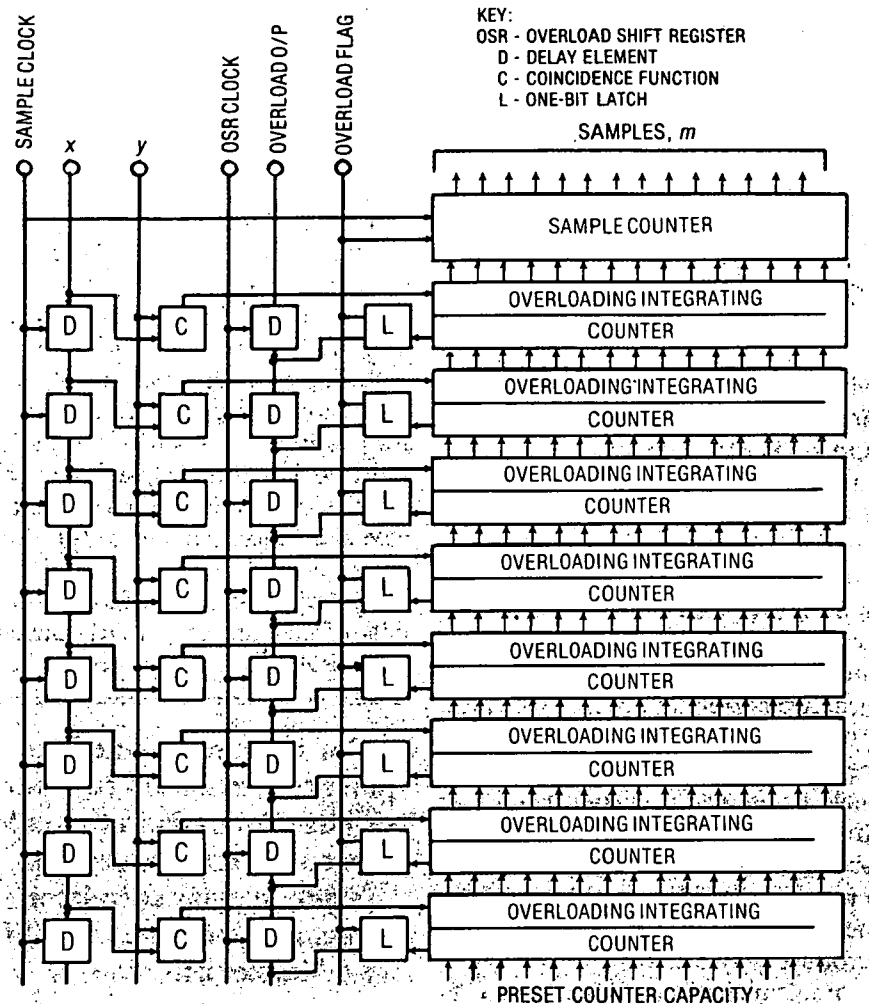


Figure 1. Layout diagram of digital polarity correlator that uses the overloading integrating counter technique.

of samples required to achieve overload conditions in the integrating counter corresponding to time delay t .¹⁴ An overload occurs after $m = N$ samples, when correlation is maximum and positive. In the case of zero correlation, an overload occurs after $m = 2N$ samples and after an infinite number of samples, when the correlation is maximum and negative. An overload cannot occur until $m \geq N$.

Figure 1 shows a polarity correlator that uses the overloading counter technique. It consists of a delaying shift register connected to a parallel array of coincidence detectors and integrating counters. An overload pattern shift register inspects the overload condition of the counters. The evolu-

ing pattern of overload states defines the correlation function shape, and the time-delay position of the first integrating counter to overload defines the position of the most significant peak of the function. A sample counter is included to count the number of input samples m , so that the value of the correlation function can be computed for any overloaded integrating counter. If the maximum capacity of the sample counter is set to be twice the capacity of the integrating counters, the significance range is limited to $1 \geq r \geq 0$. If it is required to cover the range $1 \geq r \geq -1$, two correlator circuits working in parallel can be used—one to cover the positive range, one to cover the negative range.

Correlator architecture for self test and self repair

The VLSI architecture considered here consists of a long series connection of identical correlation stages. If any one of these stages suffers faults during manufacture or becomes faulty during service, the whole chip will fail.

We have devised a self-test and self-repair structure to overcome this problem. The self-test sequence is initiated each time the chip is switched on; any faulty stages discovered as a result of these tests are automatically bypassed. This reconfigures the working stages into a continuous serial connection. Faults developing during the working life of the chip are thus automatically eliminated every time the chip is switched on.

Since the self-test control circuit must offer high reliability, it employs redundant circuit techniques. Assuming fault conditions to be evenly distributed over the chip area, the majority of faults are likely to occur in the large area occupied by the integrating counters.

Figure 2a shows the basic correlator stage; Figure 2b shows the principal additions that allow it to perform built-in self test and self repair. The delay shift register, or DSR, and the overload pattern shift register, or OSR, each have a two-to-one multiplexer. They also have a multiplexer control register, or MCR, for storing the control information for these multiplexers.

Close linking of design and test makes full functional testing possible. The design incorporates a high degree of circuit partitioning. The partitioning—coupled with the DSR, OSR, and MCR shift registers, which act as scan paths¹⁵—allows all the internal stages to be controlled and observed.

The multiplexer control register is the key feature in the self-repair mechanism. After the self-test sequence, the MCR contains the pass/fail status for each stage. Figure 3 shows a circuit schematic of the MCR and one multiplexer.

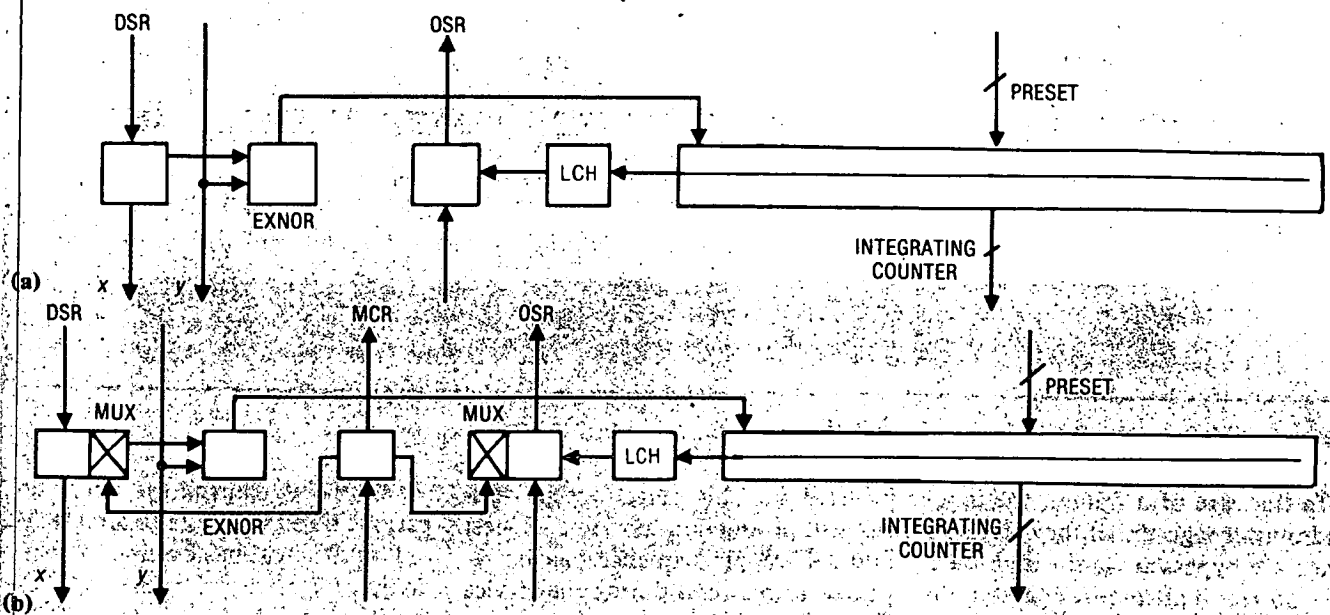


Figure 2. Basic correlator stage (a); correlator stage with built-in self-test and self-repair mechanisms (b).

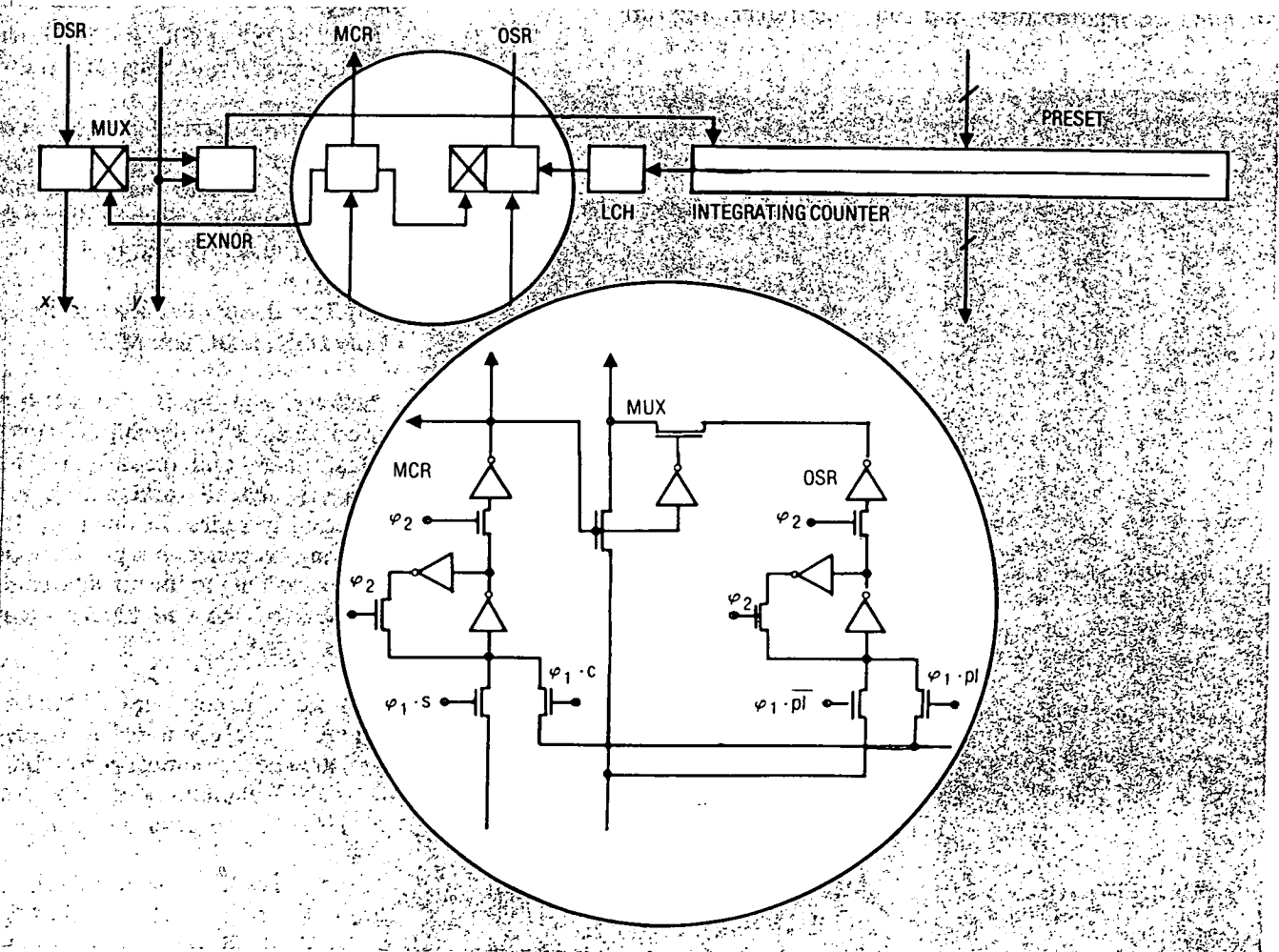


Figure 3. Zoom-in on floor plan: bypass loop and multiplexer control register.

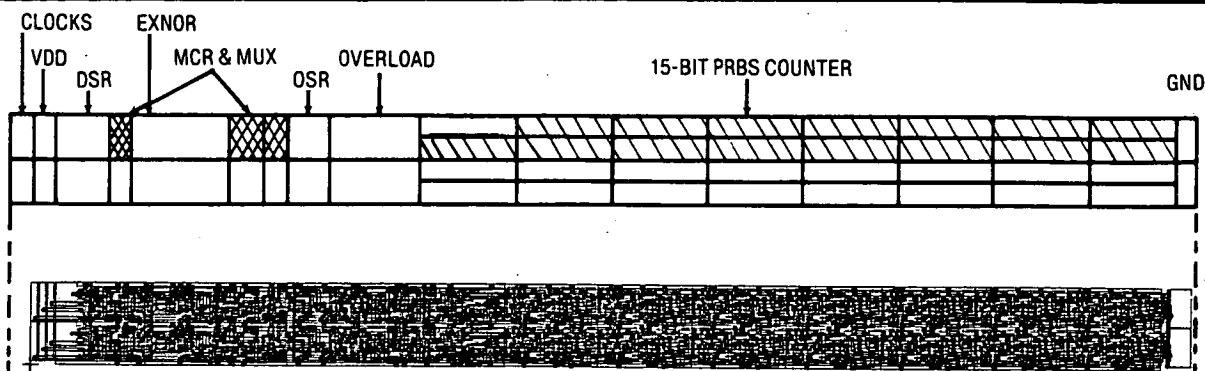


Figure 4. NMOS layout of two stages of correlation.

In the case of a failure, the input and output registers of the correlator stage are bypassed via the multiplexers, so the malfunctioning stage is short-circuited. The number of functioning stages on the chip can be read out serially from the MCR by reconfiguring it as a shift register. This parameter represents the maximum attainable correlation delay and can

be used for chip reject/accept decisions in production test. The self-test and self-repair sequence can be repeated as required during the service life of the chip.

Test strategy

The correlator operates in three distinct modes: initial test, self test and repair, and run.

Initial test. During the initial test period, three simple tests are carried out on the most basic elements of the design, the scan-path registers. These registers—DSR, MCR, and OSR—and their various control functions are tested to determine whether a chip is acceptable immediately after fabrication. The initial test sequence is as follows:

- (1) Test DSR, OSR, and MCR as shift registers and measure their delay.

- (2) Test the effect of the MCR on the DSR and OSR. This is done by shifting n ones into the MCR and then measuring the delay of the DSR and OSR, which should each be reduced by n .

- (3) Test the parallel load facilities of the DSR, OSR, and MCR.

Self test and repair. The self-test period occurs when the chip effectively tests itself and reconfigures its registers so that all working stages are connected in series. In this test, the following sequence is repeated four times, according to the possible combinations of the two binary input signals, x and y .

- (1) Reset latches and integrating counters. The counters are loaded with 4000-hex, a number corresponding to the maximum integration time of $2^{15} - 2 = 32,766$ sample clock cycles.

- (2) Set up the input conditions (x and y) by setting or clearing the DSR as required. Shift x and y through correlator for 32,766 clock cycles.

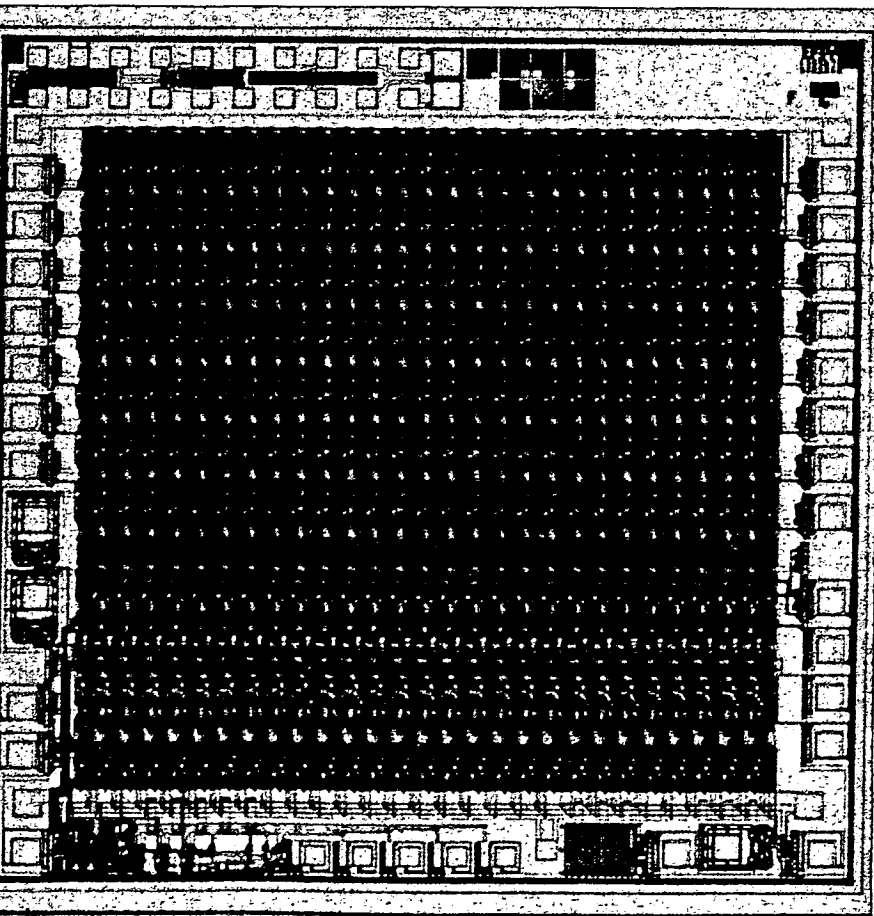


Figure 5. Chip plot of digital correlator featuring self-test and self-repair mechanisms.

3) Parallel load latches into OSR. The overload pattern can be shifted for observation.

The self-repair sequence follows the self-test sequence. During the self-test sequence, the overload signal is compared with the expected value of the overload signal. Any deviation from the expected signal results in a logic 1, which is stored in the corresponding latch. Thus, when the self-test sequence has finished, the logic 1s and 0s stored in the latches indicate the results of the self test; a logic 1 indicates a faulty stage.

The self-repair operation essentially transfers this information to the MCR, which in turn causes the faulty stages to be bypassed. The net effect is a series connection of correctly operating correlation stages.

Run. The run period automatically follows the self-test and repair sequence. After the test, the contents of the MCR can be inspected to ensure that the number of working correlator stages meets the requirements of the system in which the chip is to be installed.

Prototype design

A prototype digital correlator featuring self test and self repair has been fabricated on a five-micron channel MOS process. The prototype design contains 28 parallel stages of correlation, each of which implements the block diagram in Figure 2b. The area of the chip is 5.08 mm by 5.08 mm.

Figure 4 shows, with annotations, the layout of the two parallel stages of correlation. Each stage consists of cells that can be repeated by abutting in the y direction. The largest cell is the presettable PRBS counter, which has 15 shift register stages and thus a maximum count of approximately 32K samples.

The presettable counter is laid out in the form of a ring to minimize the circuit delays between each shift register stage.

The correlator design is semistatic throughout. This means that the clock frequency and thus the sam-

pling frequency of the correlator can range from dc to 4 MHz (for this fabrication process).

Figure 5 shows a plot of the complete chip area. The presettable counter occupies most of the active area of the chip. Figure 5, of course, includes the self-test and repair circuitry; the overhead for self test and self repair is approximately six percent.

Test results

The correlator chip has been functionally tested with a Tektronix DAS 9100 digital analysis system coupled to a Teledyne probe station. Ten packaged chips that had passed a visual inspection were functionally tested. Because many more samples were required to demonstrate the yield enhancement capability of this design, the remaining wafers were probe-tested. Unfortunately, only

130 candidates were available for testing, since the chip was fabricated as part of a multiproject wafer. More wafers are to be processed.

Figure 6 shows some of the input and output waveforms from two correlator chips. They occurred during the self-test and repair period. For display purposes, the integration time of the correlator has been reduced to just 15 clock cycles. Figure 6a shows the correlation output of a "golden chip," while Figure 6b shows the output of a chip with one failed stage. The top four traces in each figure represent the inputs to the device. In each figure, the x and y inputs sequence through their four possible combinations in accordance with the test strategy described above. For clarity, we have omitted some control signals—those that cause, for example, parallel load OSR or reset counters.

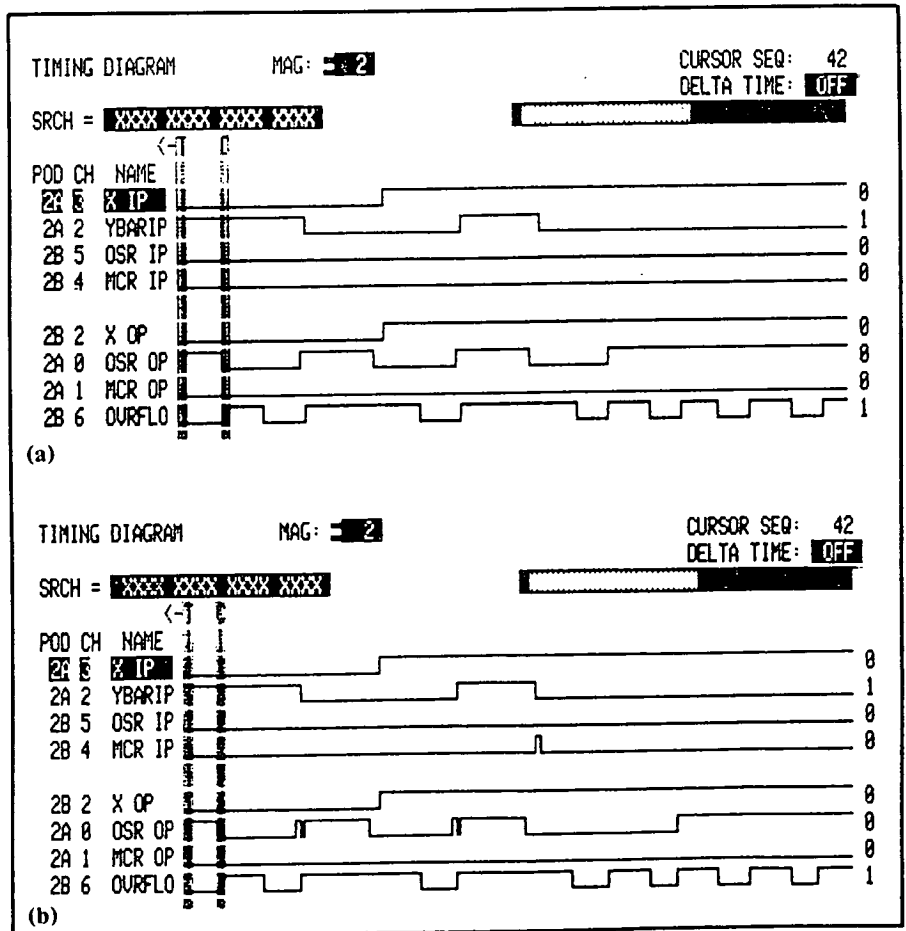


Figure 6. Test results from two correlator chips: a "golden chip" (a) and a chip with one faulty stage (b).

The significant points to note in Figure 6 are the multiplexer control register input—MCR IP—and the overload shift register output—OSR OP. All other shown signals are the same for both chips.

Moving left to right from the cursor in Figure 6, the overload output—OVRFLO—has changed from logic 1 to 0. This indicates that at least one of the integrating counters has overloaded after the prescribed period of 15 clock cycles. This result is expected, since the inputs have been equal ($x=0$, $y=0$) over this period.

When OVRFLO next goes high, the correlator has been reset and the next correlation test ($x=0$, $y=1$) begins. Also at this time, the overload pattern—that is, the contents of the latches—is transferred to the OSR and shifted out for display.

Now we can see the difference between the golden chip, Figure 6a, and the faulty chip, Figure 6b. The OSR should contain a series of 28 logic 1s; in Figure 6b, a logic 0 is in position number 27, indicating a fault in stage 27. The correlation test is repeated for the remaining combinations of x

and y : the fault is again exposed on the OSR output in the case where $x=y=1$.

Self repair is then carried out on the faulty chip. A single logic 1 is shifted into bit position 27 of the MCR, causing stage 27 to be bypassed. The correlation test, with $x=y=1$, is repeated several times at a period of 27 rather than 28 to eliminate the incorrect logic 0 on the OSR output. The result is a golden chip containing 27 stages of correlation.

The yield enhancement results are preliminary, and the sample is small—130 processed chips. Figure 7 charts the number of chips against number of working stages. It shows that 29 of the 130 candidates passed the initial test and that 27 of these yielded more than 20 stages of correlation.

Table 1 lists test results for each wafer. The multiproject wafers each contained 24 correlator chips.

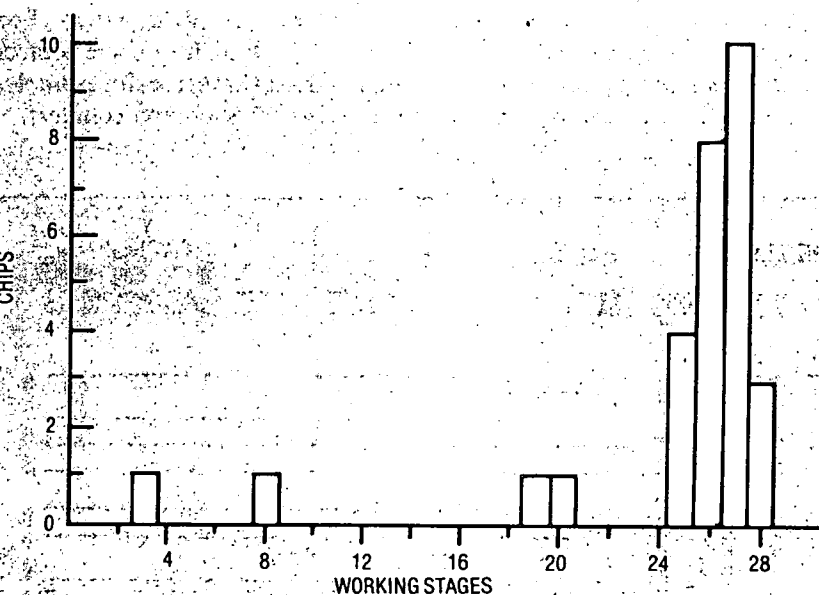


Figure 7. Number of chips vs. number of working stages.

Table 1.
Test results for each wafer.

	Without Self Repair (28 stages working)	With Self Repair (> 20 stages working)
Packaged (10 candidates)	0	2
Wafer 1 (24 candidates)	1	5
Wafer 2 (24 candidates)	0	5
Wafer 3 (24 candidates)	0	6
Wafer 4 (24 candidates)	0	0
Wafer 5 (24 candidates)	2	9
TOTALS (130 candidates)	3	27

YIELD with no yield enhancement: 2.3 percent

YIELD with yield enhancement: 20.7 percent

The VLSI structure offers an attractive digital implementation of a high-speed polarity correlator. Individual chips can be directly cascaded to realize a correlator with arbitrary resolution or delay, in contrast to other digital correlator circuits—particularly those using the parallel counter technique—which cannot be easily cascaded and do not render regular VLSI structures. Furthermore, the operating speed of parallel counter-based correlators is limited by carry signal propagation delays. The correlator described here, which is composed mainly of simple shift register stages, can approach the maximum clocking rate of a chosen VLSI fabrication process.

Functional testing of the first batch of processed chips has demonstrated that yield can be improved considerably at a very low cost in circuit overhead; the initial sample's yield enhancement factor was 9.0 for 130 chips. In addition, any of these chips can be given an exhaustive functional test in less than 150 ms at 1 MHz.

The time taken in linking design and test has proved to be time well spent. ■

Acknowledgments

This work was carried out under a United Kingdom Science and Engineering Research Council grant.

References

1. T. W. Williams, "Design for Testability: What's the Motivation?" *VLSI Design*, Vol. 4, No. 6, Oct. 1983, pp. 21-23.
2. E. B. Eichelberger and E. Lindbloom, "Trends in VLSI Testing," in *VLSI 83*, F. Anceau and E. J. Aas, eds., Elsevier Science Publishers B. V. (North Holland), Amsterdam, 1983, pp. 339-348.
3. J. S. Bendat and A. G. Piersol, *Engineering Applications of Correlation and Spectral Analysis*, John Wiley and Sons, Chichester, U.K., 1980.
4. Y. W. Lee, T. P. Cheatham, Jr., and J. B. Wiesner, "Application of Correlation Analysis to the Detection of Periodic Signals in Noise," *Proc. IRE*, Vol. 38, Oct. 1950, pp. 1165-1171.
5. J. E. Tanner and C. Mead, "A Correlating Optical Motion Detector," *Proc. Conf. Advanced Research VLSI*, MIT, Cambridge, Mass., Jan. 1984, pp. 57-64.
6. A. M. Hayes and G. Musgrave, "Correlator Design for Flow Measurement," *The Radio & Electronic Engineer*, Vol. 43, No. 6, June 1973, pp. 363-368.
7. J. H. Van Vleck and D. Middleton, "The Spectrum of Clipped Noise," *Proc. IEEE*, Vol. 54, No. 1, Jan. 1966, pp. 2-19.
8. J. Eldon, "Correlation—A Powerful Technique for Digital Signal Processing," *Application Notes*, Vol. TP-17, TRW LSI Products, Los Angeles, Calif., 1981, pp. 1-22.
9. K. W. Current, "A High Data-Rate Digital Output Correlator Design," *IEEE Trans. Computers*, Vol. C-29, No. 5, May 1980, pp. 403-405.
10. E. E. Swartzlander, Jr., "Parallel Counters," *IEEE Trans. Computers*, Vol. C-22, No. 11, Nov. 1973, pp. 1021-1024.
11. L. Dadda, "Composite Parallel Counters," *IEEE Trans. Computers*, Vol. C-29, No. 10, Oct. 1980, pp. 942-946.
12. J. R. Jordan and M. S. Beck, "Correlation Function Display and Peak Detection," *Electronics Letters*, Vol. 8, No. 24, Nov. 1972, pp. 602-604.
13. W. S. Blackley, M. A. Jack, and J. R. Jordan, "Digital Polarity Correlator," U.K. Patent Application Nos. 8306797 and 8300699, Mar. 11, 1983.
14. W. S. Blackley, M. A. Jack, and J. R. Jordan, "VLSI Digital Polarity Correlator Based on an Overloading Counter Technique," *Electronics Letters*, Vol. 19, No. 19, Sept. 1983, pp. 761-762.
15. T. W. Williams and K. P. Parker, "Design for Testability—A Survey," *IEEE Trans. Computers*, Vol. C-31, No. 1, Jan. 1982, pp. 2-15.



William Blackley is a member of the Integrated Systems Group in the Department of Electrical Engineering, University of Edinburgh. His current research interests include custom and semicustom integrated circuit design for digital signal processing and testability and yield enhancement.

Blackley received his BSc in engineering science (electrical) from the University of Edinburgh in 1979. After working briefly for Racal Microwave and Electronic Systems, Ltd., he returned to the University of Edinburgh in 1980 as a research associate. He is now studying for his PhD.



Mervyn A. Jack is a lecturer in the Department of Electrical Engineering, University of Edinburgh, where he has taught since 1979. From 1975 until 1979, he was a research fellow of the university, studying the design and application of Fourier transform processors based on surface acoustic wave and charge-coupled devices. From 1971 to 1975, he worked as a project engineer with Microwave and Electronic Systems, Ltd., Edinburgh.

Jack received his BSc and MSc in electronic engineering from Heriot-Watt University, Edinburgh, in 1971 and 1975, respectively. He received his PhD from the University of Edinburgh in 1978. Jack is a member of the IEE.



James R. Jordan joined the Department of Electrical Engineering, University of Edinburgh, in 1969 after industrial engineering experience with EMI Electronics, Ltd., and teaching experience at Teeside Polytechnic. He is now a senior lecturer specializing in teaching system theory and electronic instrumentation to undergraduate students and reliability and fault detection methods to postgraduates. His principal research interest is the application of LSI circuits and microelectronic fabrication techniques to electronic instrumentation and transducers.

Jordan received his MSc from the University of Surrey in 1967 and PhD from the University of Bradford in 1973.

The authors' address is Department of Electrical Engineering, University of Edinburgh, King's Buildings, Mayfield Rd., Edinburgh, EH9 3JL Scotland.

References

1. T. W. Williams, "Design for Testability: What's the Motivation?," *VLSI Design*, Vol. 4, pp. 21-23 (Oct., 1983).
2. W. S. Blackley, M. A. Jack, and J. R. Jordan, "VLSI Digital Polarity Correlator Based on an Overloading Counter Technique," *Electronics Letters*, Vol. 19, pp. 761-762 (Sept., 1983).
3. J. S. Bendat and A. G. Piersol, *Engineering Applications of Correlation and Spectral Analysis*, John Wiley and Sons, Chichester (1980). A Wiley Interscience Publication
4. P. R. Roth, "Effective Measurements Using Digital Signal Analysis," *IEEE Spectrum*, Vol. 8, pp. 62-70 (Apr., 1971).
5. D. A. Gandolfo, J. R. Tower, L. D. Elliott, E. J. Nossen, and L. W. Martinson, "CCD's for Spread Spectrum Applications," pp. 90-96 in *Proc. International Specialist Seminar on Case Studies in Advanced Signal Processing.*, IEE (Sept., 1979).
6. W. B. Allen and E. C. Westerfield, "Digital Compressed-Time Correlators and Matched Filters for Active Sonar," *J. Acoustical Society of America*, Vol. 36, pp. 121-139 (1964).
7. S. Cacopardi, "Applicability of the Relay Correlator to Radar Signal Processing," *Electronics Letters*, Vol. 19, pp. 722-723 (Sept., 1983).
8. J. R. Forrest and D. J. Price, "Digital Correlation for Noise Radar Systems," *Electronics Letters*, Vol. 14, pp. 581-582 (Aug., 1978).
9. D. J. Price, "Correlation Processing in Noise Radar," pp. 8/1 - 8/4 in *Colloquium on Correlation Processing*, IEE Colloquium digest No. 1979/32, Savoy Place, London (May, 1979).
10. F. J. Taylor, V. Shenoy, C. P. Olinger, and F. Wasserman, "Aneurysm Detection Using One-Bit Correlation," *Medical and Biological Engineering and Computing*, Vol. 17, pp. 443-448 (July, 1979).

11. F. J. Looft, III and W. J. Heetderks, "Real Time Correlator for Detecting Single Units in Peripheral Nerve," *IEEE Trans. Biomedical Engineering*, Vol. BME-25, pp. 564-567 (Nov., 1978).
12. S. E. Fu and J. S. Lee, "A Video System for Measuring the Blood Flow Velocity in Microvessels," *IEEE Trans. Biomedical Engineering*, Vol. BME-25, pp. 295-297 (May, 1978).
13. H. Ekre, "Polarity Coincidence Correlation Detection of a Weak Noise Source," *IEEE Trans. Information Theory*, Vol. IT-9, pp. 18-23 (Jan., 1963).
14. Y. W. Lee, T. P. Cheatham Jr., and J. B. Wiesner, "Application of Correlation Analysis to the Detection of Periodic Signals in Noise," *Proc. IRE*, Vol. 38, pp. 1165-1171 (Oct., 1950).
15. C. M. Rader, "An Improved Algorithm for High Speed Auto Correlation with Applications to Spectral Estimation," *IEEE Trans. Audio Electroacoustics*, Vol. AU-18, pp. 439-441 (Dec., 1970).
16. B. W. Finnie, "Digital Correlation Techniques for Identifying Dynamic Systems," Ph.D. Thesis, University of Edinburgh (May, 1965).
17. D. Bassi, "Pseudorandom Digital Cross Correlator for Impulse Response Measurements," *Review of Scientific Instruments*, Vol. 51, pp. 795-798 (June, 1980).
18. R. J. Polge and E. M. Mitchell, "Impulse Response Determination by Cross Correlation," *IEEE Trans. Aerospace and Electronic Systems*, Vol. AES-6, pp. 91-97 (Jan., 1970).
19. J. A. M. McDonnell and J. Forrester, "Polarity Coincidence Techniques for Correlation Function Measurement and System Response Evaluation," *The Radio and Electronic Engineer*, Vol. 40, pp. 165-172 (Oct., 1970).
20. L. P. Horwitz and G. L. Shelton, Jr., "Pattern Recognition Using Autocorrelation," *Proc. IRE*, Vol. 49, pp. 175-185 (Jan., 1961).
21. J. E. Tanner and C. A. Mead, "A Correlating Optical Motion Detector," pp. 57-64 in *Proc. Conf. Advanced Research in VLSI*, MIT, Cambridge, MA. (Jan., 1984).

22. D. I. Barnea and H. F. Silverman, "A Class of Algorithms for Fast Digital Image Registration," *IEEE Trans. Computers*, Vol. C-21, pp. 179-186 (Feb., 1972).
23. H. Murakami and B. V. K. Vijaya Kumar, "Correlation of Binarized Images," *IEEE Trans. Aerospace and Electronic Systems*, Vol. AES-19, pp. 322-328 (Mar., 1983).
24. J. S. Boland, L. J. Pinson, E. G. Peters, G. R. Kane, and W. W. Malcolm, "Design of a Correlator for Real Time Video Comparisons," *IEEE Trans. Aerospace and Electronic Systems*, Vol. AES-15, pp. 11-19 (Jan., 1979).
25. M. Azaria and D. Hertz, "Time Delay Estimation by Generalized Cross Correlation Methods," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-32, pp. 280-285 (Apr., 1984).
26. J. P. Ianniello, "Time Delay Estimation Via Cross-Correlation in the Presence of Large Estimation Errors," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-30, pp. 998-1003 (Dec., 1982).
27. Special Issue, "Time Delay Estimation," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-29, (June, 1981). Edited by G. C. Carter
28. J. R. Jordan and R. G. Kelly, "Integrated Circuit Correlator for Flow Measurement," *Measurement and Control*, Vol. 9, pp. 267-270 (July, 1976).
29. T. S. Durrani and C. A. Greated, *Laser Systems in Flow Measurement*, Plenum Press, New York and London (1977).
30. F. Boonstoppel, B. Veltmen, and F. Vergouwen, "The Measurement of Flow by Cross Correlation Techniques," pp. 110-124 in *Proc. Conf. Industrial measurement techniques for on-line computers*, IEE Conf. Publication No. 43, London (June, 1968).
31. W. Matthes, W. Riebold, and E. De Cooman, "Measurement of the Velocity of Gas Bubbles in Water by a Correlation Method," *Review of Scientific Instruments*, Vol. 41, pp. 843-845 (June, 1970).
32. M. Intaglietta and W. R. Tompkins, "System for Measurement of Velocity of Microscopic Particles in Liquids," *IEEE Trans. Biomedical Engineering*, Vol. BME-18, pp. 376-377 (Sept., 1971).

33. W. R. Tompkins, R. Monti, and M. Intaglietta, "Velocity Measurement by Self Tracking Correlator," *Review of Scientific Instruments*, Vol. 45, pp. 647-649 (May, 1974).
34. D. Jones, "An On-Board Digital Correlator for Spacecraft VLF Radio Wave Studies," *IEEE Trans. Geoscience Electronics*, Vol. GE-12, pp. 9-18 (1974).
35. R. Jones, "The Single-Clipped Digital Malvern Correlator," pp. 7/1 - 7/4 in *Colloquium on Correlation Processing*, IEE Colloquium digest No. 1979/32, Savoy Place, London (May, 1979).
36. M. Corti, A. De Agostini, and V. Degiorgio, "Fast Digital Correlator for Weak Optical Signals," *Review of Scientific Instruments*, Vol. 45, pp. 888-893 (July, 1974).
37. P. C. Egau, "Correlation Systems in Radio Astronomy and Related Fields," *IEE Proc. Part F*, Vol. 131, pp. 32-39 (Feb., 1984).
38. L. R. Allen and R. H. Frater, "Wideband Multiplier Correlator," *IEE Proc.*, Vol. 117, pp. 1603-1608 (Aug., 1970).
39. P. J. Kindlmann and E. B. Hooper, Jr., "High Speed Correlator," *Review of Scientific Instruments*, Vol. 39, pp. 864-872 (June, 1968).
40. M. Fukao, "A Wide Band Correlator," *Review of Scientific Instruments*, Vol. 42, pp. 783-788 (June, 1971).
41. J. C. Brenot, J. A. Fayeton, and J. C. Houver, "Fast Multichannel Time Correlator for Coincidence Experiments in Atomic Physics," *Review of Scientific Instruments*, Vol. 51, pp. 1623-1629 (Dec., 1980).
42. B. B. Lee and E. S. Furgason, "An Evaluation of Ultrasound NDE Correlation Flaw Detection Systems," *IEEE Trans. Sonics and Ultrasonics*, Vol. SU-29, pp. 359-369 (Nov., 1982).
43. C. M. Beck, R. M. Henry, B. T. Lowe, and A. Plaskowski, "Autocorrelation Function Parameters Used to Indicate Incipient Blockage in a Pneumatic Transport System," *Electronics Letters*, Vol. 18, pp. 705-706 (Aug., 1982).

44. H. Meyr and G. Spies, "The Structure and Performance of Estimators for Real-Time Estimation of Randomly Varying Time Delay," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-32, pp. 81-94 (Feb., 1984).
45. C. H. Knapp and G. C. Carter, "The Generalized Correlation Method for Estimation of Time Delay," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-24, pp. 320-327 (Aug., 1976).
46. J. N. Bradley and R. L. Kirlin, "Delay Estimation by Expected Value," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-32, pp. 19-27 (Feb., 1984).
47. J. C. Hassab and R. E. Boucher, "Optimum Estimation of Time Delay by a Generalized Correlator," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-27, pp. 373-380 (Aug., 1979).
48. H. Meyr, "Application of Digital Signal Processing in Measuring," pp. 431-438 in *Signal Processing II: Theories and Applications*, ed. H. W. Schussler, Elsevier Science Publishers B.V. (North Holland) (1983).
49. H. Meyr, "Delay-Lock Tracking of Stochastic Signals," *IEEE Trans. Communications*, Vol. COM-24, pp. 331-339 (Mar., 1976).
50. A. W. Lohmann and B. Wirnitzer, "Triple Correlations," *Proc. IEEE*, Vol. 72, pp. 889 - 901 (July, 1984). Invited Paper
51. P. W. Cheney, "A Digital Correlator Based on the Residue Number System," *IRE Trans. Electronic Computers*, Vol. 10, pp. 63-70 (Mar., 1961).
52. F. H. Lange, *Correlation Techniques*, Iliffe Books Limited, London (1967).
53. F. E. Brooks and H. W. Smith, "A Computer for Correlation Functions," *Review of Scientific Instruments*, Vol. 23, pp. 121-126 (Mar., 1952).
54. W. R. Bennett, "The Correlatograph. A Machine for Continuous Display of Short Term Correlation," *Bell Systems Tech. J.*, Vol. 32, pp. 1173-1185 (Sept., 1953).

55. Y. W. Lee, *Statistical Theory of Communication*, John Wiley and Sons Inc., New York (1960).
56. A. B. Carlson, *Communication Systems*, McGraw-Hill Kogakusha Ltd., Tokyo (1975).
57. R. P. Keech, "The KPC Multichannel Correlation Signal Processor for Velocity Measurement," *Trans. Inst. Measurement and Control*, Vol. 4, pp. 43-52 (Jan. - Mar., 1982).
58. J. R. Jordan and B. A. Manook, "Correlation-Function Peak Detector," *IEE Proc.*, Vol. 128, Part E, pp. 74-78 (Mar., 1981).
59. J. Coulthard and R. P. Keech, "A Six-Channel Microprocessor Controlled Correlator," pp. 4/1 - 4/6 in *Colloquium on Correlation Processing*, IEE Colloquium digest No. 1979/32, Savoy Place, London (May, 1979).
60. A. M. Hayes and G. Musgrave, "The Variance of Time Delay Estimates from Cross Correlation Functions," pp. 2/1 - 2/3 in *Colloquium on Correlation Processing*, IEE Colloquium digest No. 1979/32, Savoy Place, London (May, 1979).
61. S. M. Kay, "The Effect of Sampling Rate on Autocorrelation Estimation," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-29, pp. 859-867 (Aug., 1981).
62. F. K. Bowers, D. A. Whyte, T. L. Landecker, and R. J. Klingler, "A Digital Correlation Spectrometer Employing Multiple-Level Quantization," *Proc. IEEE*, Vol. 61, pp. 1339-1343 (Sept., 1973).
63. D. A. Gandolfo, J. R. Tower, J. I. Pridgen, and S. C. Munroe, "Analog-Binary CCD Correlator: A VLSI Signal Processor," *IEEE Trans. Electronic Devices*, Vol. ED-26, pp. 596-603 (Apr., 1979).
64. D. Lagoyannis, "Stieltjes-Type Correlator Based on Delta-Sigma Modulation," *IEE Proc.*, Vol. 128, Part G, pp. 9-14 (Feb., 1981).
65. R. S. Miller and M. B. Berry, "A Merged Pipe Organ Binary-Analog Correlator," *IEEE J. Solid-State Circuits*, Vol. SC-17, pp. 20-27 (Feb., 1982).

66. A. Gersho, "Principles of Quantization," *IEEE Trans. Circuits and Systems*, Vol. CAS-25, pp. 427-36 (July, 1978).
67. K. Y. Chang and A. D. Moore, "Modified Digital Correlator and its Estimation Errors," *IEEE Trans. Information Theory*, Vol. IT-16, pp. 699-706 (Nov., 1970).
68. J. J. Freeman, "The Action of Dither in a Polarity Coincidence Correlator," *IEEE Trans. Communications*, Vol. COM-22, pp. 857-862 (June, 1974).
69. L. Cheded, P. A. Payne, and S. M. Jawad, "High Speed Digital Cross-correlator Design for Multifrequency Response Analysis," *The Radio and Electronic Engineer*, Vol. 53, pp. 229-234 (June, 1983).
70. D. Lagoyannis, "Correlator Based on Delta-Sigma Modulation," *Electronics Letters*, Vol. 12, pp. 253-254 (May, 1976).
71. S. Nakamura, "A Digital Correlator Using Delta Modulation," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-24, pp. 238-243 (June, 1976).
72. W. N. Cheung, "Correlation Measurement by Delta Sigma Modulation," *IEEE Trans. Indust. Electron. Contr. Instrum.*, Vol. IECI-26, pp. 88-92 (May, 1979).
73. R. E. H. Bywater, W. Matley, and D. Brock, "Design of a flexible phase reversal modulation correlator," *The Radio and Electronic Engineer*, Vol. 46, pp. 129-135 (Mar., 1976).
74. L. F. Rocha, B. Cernuschi-Frias, and C. Orda, "Convolution and Correlation Using Delta Modulators," *Proc. IEEE*, Vol. 68, pp. 1024-1026 (Aug., 1980).
75. D. G. Watts, "A General Theory of Amplitude Quantization with Applications to Correlation Determination," *IEE Proc.*, Vol. 109, Part C, pp. 209-18 (1962).
76. B. Widrow, "A Study of Rough Amplitude Quantization by Means of Nyquist Sampling Theory," *IRE Trans. Circuit Theory*, Vol. CT-3, pp. 266-276 (Dec., 1956).
77. L. C. Andrews, "Analysis of a Cross Correlator with a Clipper in One Channel," *IEEE Trans. Information Theory*, Vol. IT-26, pp. 743-746 (Nov., 1980).

78. W. F. Sheppard, "On the Calculation of the Most Probable Values of Frequency - Constants, for Data Arranged According to Equidistant Divisions of a Scale," *Proc. London Mathematical Society*, Vol. 29, p. 353 (1898).
79. J. G. Ables, B. F. C. Cooper, A. J. Hunt, G. G. Moorey, and J. W. Brooks, "A 1024-Channel Digital Correlator," *Review of Scientific Instruments*, Vol. 46, pp. 284-295 (Mar., 1975).
80. G. C. Anderson and M. A. Perry, "A Calibrated Real Time Correlator/Averager/Probability Analyser," *Hewlett Packard J.*, Vol. 21, pp. 9-15 (1969).
81. P. E. Dewdney, "Product Transition Correlator," *Review of Scientific Instruments*, Vol. 51, pp. 1548-1552 (Nov., 1980).
82. J. H. Van Vleck and D. Middleton, "The Spectrum of Clipped Noise," *Proc. IEEE*, Vol. 54, pp. 2-19 (Jan., 1966).
83. L. C. Andrews, "The Output PDF of a Polarity Coincidence Correlation Detector," *IEEE Trans. Aerospace and Electronic Systems*, Vol. AES-10, pp. 712-714 (Sept., 1974).
84. H. Berndt, "Correlation Function Estimation by a Polarity Method Using Stochastic Reference Signals," *IEEE Trans. Information Theory*, Vol. IT-14, pp. 796-801 (Nov., 1968).
85. D. Landsberg and A. Cohen, "Fast Correlation Estimation by Random Reference Correlator," *IEEE Trans. Instrumentation and Measurement*, Vol. IM-32, pp. 438-442 (Sept., 1983).
86. P. G. A. Jespers, M. G. Windal, and T. Watteyne, "An Integrated Binary Correlator Module," *IEEE J. Solid-State Circuits*, Vol. SC-18, pp. 286-290 (June, 1983).
87. C. R. Cahn, "Performance of Digital Matched Filter Correlator with Unknown Interference," *IEEE Trans. Communication Technology*, Vol. COM-19, pp. 1163-1172 (Dec., 1971).
88. A. M. Hayes and G. Musgrave, "Correlator Design for Flow Measurement," *The Radio and Electronic Engineer*, Vol. 43, pp. 363-368 (June, 1973).

89. J. A. Eldon and J. D. Haight, "New CMOS Chip Facilitates Multibit Correlation," pp. 44.2 in *Proc. International Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, IEEE, San Diego, CA. (1984).
90. J. A. Eldon, "Digital Correlators Suit Military Applications," *Electronic Design News (EDN)*, pp. 148-160 (Aug., 1984).
91. J. A. Eldon, "Correlation - A Powerful Technique for Digital Signal Processing," pp. 1-22 in *Application Notes*, TRW LSI Products, La Jolla, CA. (1981).
92. K. W. Current, "A High Data-Rate Digital Output Correlator Design," *IEEE Trans. Computers*, Vol. C-29, pp. 403-405 (May, 1980).
93. K. W. Current and D. A. Mow, "Digital Correlator Design with Four-Valued Threshold Logic," pp. 237-241 in *Digest of Papers, International Symp. Circuits and Systems (ISCAS)*, (1978).
94. E. E. Swartzlander, Jr., "Parallel Counters," *IEEE Trans. Computers*, Vol. C-22, pp. 1021-1024 (Nov., 1973).
95. L. Dadda, "Composite Parallel Counters," *IEEE Trans. Computers*, Vol. C-29, pp. 942-946 (Oct., 1980).
96. K. W. Current, "Pipelined Binary Parallel Counters Employing Latched Quaternary Logic Full Adders," *IEEE Trans. Computers*, Vol. C-29, pp. 400-403 (May, 1980).
97. K. W. Current and D. A. Mow, "Implementing Parallel Counters with Four-Valued Threshold Logic," *IEEE Trans. Computers*, Vol. C-28, pp. 200-204 (Mar., 1979).
98. J. R. Jordan and M. S. Beck, "Correlation Function Display and Peak Detection," *Electronics Letters*, Vol. 8, pp. 602-604 (Nov., 1972).
99. W. S. Blackley, M. A. Jack, and J. R. Jordan, "Digital Polarity Correlator," UK Patent Application Nos. 8306797 and 8300699 (Mar., 1983).
100. W. S. Blackley, "Digital Polarity Correlator Integrated Circuit Featuring Built-In Self Repair Structures for Yield Enhancement and High Reliability," pp. 12/1 - 12/10 in *Proc. 4th International Conference on Custom and Semi-Custom ICs*, Prodex Seminars Ltd., in association with the IEE., London

(Nov., 1984).

101. W. S. Blackley, M. A. Jack, and J. R. Jordan, "A Digital Polarity Correlator with Built-In Self Test and Self Repair," *IEEE Design and Test of Computers*, Vol. 1, pp. 42-49 (May, 1984).
102. W. S. Blackley, M. A. Jack, and J. R. Jordan, "Built-In Test and Self Repair Mechanisms in a Digital Correlator Integrated Circuit," pp. 12/1 - 12/10 in *Proc. Conf. Design for Tactical Avionics Maintainability*, North Atlantic Treaty Organisation, Advisory Group for Aerospace Research and Development (NATO - AGARD), AGARD Conf. Preprint No. 361, Brussels, Belgium (May, 1984).
103. W. S. Blackley, M. A. Jack, and J. R. Jordan, "A Digital Polarity Correlator Featuring Built-In Self Test and Self Repair Mechanisms," pp. 289-294 in *Digest of papers, International Test Conf.*, IEEE, Philadelphia, PA. (Oct., 1983).
104. M. A. Monahan, K. Bromley, and R. P. Bocker, "Incoherent Optical Correlators," *Proc. IEEE*, Vol. 65, pp. 121-129 (Jan., 1977).
105. D. Casasent, "Coherent Optical Pattern Recognition," *Proc. IEEE*, Vol. 67, pp. 813-825 (May, 1979).
106. T. M. Turpin, "Spectrum Analysis Using Optical Processing," *Proc. IEEE*, Vol. 69, pp. 79-92 (Jan., 1981). Invited Paper
107. T. W. Cole, "New Class of One-Bit Digital Auto Correlator," *Electronics Letters*, Vol. 16, pp. 86-88 (Jan., 1980).
108. W. T. Rhodes, "Acousto-Optical Signal Processing: Convolution and Correlation," *Proc. IEEE*, Vol. 69, pp. 65-79 (Jan., 1981). Invited Paper
109. A. Korpel, "Acousto-Optics - A Review of Fundamentals," *Proc. IEEE*, Vol. 69, pp. 48-53 (Jan., 1981). Invited Paper
110. R. A. Becker, R. W. Ralston, and P. V. Wright, "Wide-Band Monolithic Acoustoelectric Memory Correlators," *IEEE Trans. Sonics and Ultrasonics*, Vol. SU-29, pp. 289-298 (Nov., 1982).

111. C. M. Elias, "An Ultrasonic Pseudorandom Signal-Correlation System," *IEEE Trans. Sonics and Ultrasonics*, Vol. SU-27, pp. 1-7 (Jan., 1980).
112. C. M. Verber, R. P. Kenan, and J. R. Busch, "Design and Performance of an Integrated Optical Digital Correlator," *J. Lightwave Technology*, Vol. LT-1, pp. 256-261 (Mar., 1983).
113. G. Comoretto, "A Microprocessor-Controlled Multichannel Counter for a Digital Autocorrelator," *J. Phys. E: Sci. Instrum.*, Vol. 16, pp. 836-839 (1983).
114. L. Basano, P. Ottonello, and E. Schiavi, "Improvements in the Design of Time-Delay Correlators," *J. Phys. E: Sci. Instrum.*, Vol. 16, pp. 840-843 (1983).
115. B. Wenk, "Aspects of Correlator Design for Industrial Applications," pp. 89-92 in *Signal Processing II: Theories and Applications*, ed. H. W. Schussler, Elsevier Science Publishers B.V. (North Holland) (1983).
116. R. M. Henry, "An Improved Algorithm Allowing Fast On-Line Polarity Correlation by Microprocessor or Minicomputer," pp. 3/1 - 3/4 in *Colloquium on Correlation Processing*, IEE Colloquium digest No. 1979/32, Savoy Place, London (May, 1979).
117. R. Fell, "Microprocessor-Based Cross-Correlators Using the "Skip" Algorithm," pp. 25-32 in *Proc. Conf. the Influence of Microelectronics on Measurements Instruments and Transducer Design*, IEE Conf. Publication No. 55, Manchester, UK. (June, 1982).
118. J. R. Jump and S. R. Ahuja, "Effective Pipelining of Digital Systems," *IEEE Trans. Computers*, Vol. C-27, pp. 855-865 (Sept., 1978).
119. A. L. Fisher and H. T. Kung, "Synchronizing Large Systolic Arrays," pp. 44-52 in *Proc. SPIE Vol. 341, Real Time Signal Processing V*, The Society of Photo-Optical Instrumentation Engineers, Arlington, VA. (May, 1982).
120. S. Y. Kung and R. J. Gal-Ezer, "Synchronous versus Asynchronous Computation in Very Large Scale Integrated (VLSI) Array Processors," pp. 53-65 in *Proc. SPIE Vol. 341, Real Time Signal Processing V*, The Society of Photo-Optical Instrumentation Engineers, Arlington, VA. (May, 1982).

121. C. A. Mead and L. A. Conway, *Introduction to VLSI Systems*, Addison-Wesley, Reading, MA. (1980).
122. H. T. Kung, "Why Systolic Architectures?," *IEEE Computer*, pp. 37-46 (Jan., 1982).
123. E. G. Magill, D. M. Grieco, R. H. Dyck, and P. C. Y. Chen, "Charge-Coupled Device Pseudo-Noise Matched Filter Design," *Proc. IEEE*, Vol. 67, pp. 50-60 (Jan., 1979).
124. B. E. Burke, D. L. Smythe, D. J. Silversmith, W. H. McGonagle, R. W. Mountain, and B. J. Felton, "A 10MHz. CCD Time-Integrating Correlator," pp. 256-257 in *Digest of Papers, International Solid State Circuits Conf.*, IEEE, New York, NY. (1983).
125. P. B. Denyer, J. Mavor, and J. W. Arthur, "Miniature Programmable Transversal Filter Using CCD/MOS Technology," *Proc. IEEE*, Vol. 67, pp. 42-50 (Jan., 1979).
126. J. Mavor, J. W. Arthur, and P. B. Denyer, "Analogue CCD Correlator Using Monolithic MOST Multipliers," *Electronics Letters*, Vol. 13, pp. 373-374 (June, 1977).
127. E. P. Herrmann and D. A. Gandolfo, "Programmable CCD Correlator," *IEEE Trans. Electronic Devices*, Vol. ED-26, pp. 117-122 (Feb., 1979).
128. J. R. Jordan, "Integrated Circuit Relay Correlator for Measurement System Applications," *Electronics Letters*, Vol. 15, pp. 366-367 (June, 1979).
129. W. D. Pritchard and J. N. Gooding, "Design and Application of a Cascadable Binary Weighted Analogue Correlator," pp. 241-246 in *Proc. 5th International Conf. Charge Coupled Devices (CCD'79)*, ed. J. Mavor, University of Edinburgh, Centre for Industrial Consultancy and Liason, Edinburgh (1979).
130. Y. A. Haque and M. A. Copeland, "Design and Characterization of a Real-Time Correlator," *IEEE J. Solid-State Circuits*, Vol. SC-12, pp. 642-649 (Dec., 1977).
131. J. L. Buie and D. R. Breuer, "A Large-Scale Integrated Correlator," *IEEE J. Solid-State Circuits*, Vol. SC-7, pp. 357-363 (Oct., 1972).

132. N. A. Saethermoen, B. Skeie, and S. Prytz, "Digital SOS/MOS Correlator: Basic System Component in Experimental Army Spread Spectrum Radio," pp. 73-78 in *Proc. Conf. the The Impact of High Speed and VLSI Technology on Communications Systems*, IEE Conf. Publication No. 230, London (Dec., 1983).
133. C. C. Foster and F. D. Stockton, "Counting Responders in an Associative Memory," *IEEE Trans. Computers*, Vol. C-20, pp. 1580-1583 (Dec., 1971).
134. D. D. Gajski, "Parallel Compressors," *IEEE Trans. Computers*, Vol. C-29, pp. 393-398 (May, 1980).
135. P. R. Cappello and K. Steiglitz, "A Fast Tally Structure and Applications to Signal Processing," pp. 25A.4 in *Proc. International Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, IEEE, San Diego, CA. (1984).
136. K. W. Current, "High Density Integrated Computing Circuitry with Multiple Valued Logic," *IEEE J. Solid-State Circuits*, Vol. SC-15, pp. 127-131 (Feb., 1980).
137. J. C. White, J. M. Keen, M. F. Hamer, D. V. McCaughan, and J. R. Hill, "A Fast 32 Point Analogue Correlator," pp. 237-240 in *Proc. 5th International Conf. Charge Coupled Devices (CCD'79)*, ed. J. Mavor, University of Edinburgh, Centre for Industrial Consultancy and Liason, Edinburgh (1979).
138. R. A. Haken, "An Electronically Programmable Transversal Input Filter," *IEEE J. Solid-State Circuits*, Vol. SC-17, pp. 34-39 (Feb., 1982).
139. R. T. F. Williams, "Correlators Using M.O.S.T.'s in Sonar Applications," *J. Sound and Vibration*, Vol. 9, pp. 161-168 (1969).
140. W. M. Gentleman and H. T. Kung, "Matrix Triangularisation by Systolic Arrays," pp. 19-26 in *Proc. SPIE Vol. 298, Real Time Signal Processing IV*, The Society of Photo-Optical Instrumentation Engineers (1981).
141. J. M. Jover and T. Kailath, "Design Framework for Systolic-Type Arrays," pp. 8.5 in *Proc. International Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, IEEE, San Diego, CA. (1984).

142. H. Barral and N. Moreau, "Circuits for Digital Signal Processing," pp. 44.9 in *Proc. International Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, IEEE, San Diego, CA. (1984).
143. W. E. Snelling and J. E. Penn, "A Fully Pipelined, Bit-Sliced, VLSI Correlator," pp. 313-320 in *Proc. Digital Signal Processing - 84*, ed. V. Cappellini and A. G. Constantinides, Elsevier Science Publishers B.V. (North Holland) (1984).
144. S. K. Kawahara, R. P. O'Connell, and J. G. Peterson, "A One-Micron Bipolar VLSI Convolver," pp. 226-227 in *Proc. International Solid-State Circuits Conf. (ISSCC)*, IEEE (Feb., 1981).
145. J. G. McWhirter, J. V. McCanny, and K. W. Wood, "Novel Multibit Convolver/Correlator Chip Design Based on Systolic Array Principles," pp. 66-73 in *Proc. SPIE Vol. 341, Real Time Signal Processing V*, The Society of Photo-Optical Instrumentation Engineers, Arlington, VA. (May, 1982).
146. R. A. Evans, D. Wood, K. Wood, J. V. McCanny, J. G. McWhirter, and A. P. H. McCabe, "A CMOS Implementation of a Systolic Multi-Bit Convolver Chip," pp. 227-235 in *VLSI 83*, ed. F. Anceau and E. J. Aas, Elsevier Science Publishers B.V. (North Holland) (1983).
147. A. G. Corry and K. Patel, "Architecture of a CMOS Correlator," *GEC J. Research*, Vol. 1, pp. 35-38 (1983).
148. T. W. Williams and K. P. Parker, "Design for Testability - A Survey," *Proc. IEEE*, Vol. 71, pp. 98-112 (Jan., 1983). Invited paper.
149. E. J. McCluskey and S. Bozorgui-Nesbat, "Design for Autonomous Test," *IEEE Trans. Computers*, Vol. C-30, pp. 866-875 (Nov., 1981).
150. C. H. Chen, "Designing Testable Synchronous Logic," pp. 89-94 in *Digest of papers, International Test Conf.*, IEEE, Philadelphia, PA. (1981).
151. P. K. Lala, "Current Problems in VLSI Testing and Testability," *The Radio and Electronic Engineer*, Vol. 54, pp. 415-423 (Oct., 1984).

152. M. T. M. R. Segers, "The Impact of Testing on VLSI Design Methods," *IEEE J. Solid-State Circuits*, Vol. SC-17, pp. 481-486 (June, 1982). Invited Paper
153. T. E. Mangir and A. Avizienis, "Fault Tolerant Design for VLSI: Effect of Interconnect Requirements on Yield Improvement of VLSI Designs," *IEEE Trans. Computers*, Vol. C-31, pp. 609-615 (July, 1982).
154. P. Banerjee and J. A. Abraham, "Generating Tests for Physical Failures in MOS Logic Circuits," pp. 554-559 in *Digest of papers, International Test Conf.*, IEEE, Philadelphia, PA. (1983).
155. S. G. Papaioannou, "Optimal Test Generation in Combinational Networks by Pseudo Boolean Programming," *IEEE Trans. Computers*, Vol. C-26, pp. 553-560 (June, 1977).
156. J. Savir and P. H. Bardell, "On Random Pattern Test Length," pp. 95-106 in *Digest of papers, International Test Conf.*, IEEE, Philadelphia, PA. (1983).
157. G. Grassl, "Design for Testability," pp. 1-36 in *Proc. NATO Advanced Study Institute on VLSI Design*, North Atlantic Treaty Organisation, Louvain, Belgium. (1980).
158. R. G. Bennetts, *Design of Testable Logic Circuits*, Addison-Wesley Publishing Company, London (1984).
159. M. J. Y. Williams and J. B. Angell, "Enhancing Testability of Large Scale Integrated Circuits via Test Points and Additional Logic," *IEEE Trans. Computers*, Vol. C-22, pp. 46-60 (Jan., 1973).
160. E. I. Muehldorf, "Designing LSI Logic for Testability," pp. 45-49 in *Proc. IEEE Semiconductor Test Conf.*, (1976).
161. J. R. Grierson, "The UK5000 Project," pp. 1/1 - 1/4 in *Proc. 3th International Conference on Custom and Semi-Custom ICs*, Prodex Seminars Ltd., in association with the IEE., London (Nov., 1983).
162. R. A. Frohwerk, "Signature Analysis: A New Digital Field Service Method," *Hewlett-Packard J.*, pp. 2-8 (May, 1977).
163. H. J. Nadig, "Signature Analysis - Concepts, Examples, and Guidelines," *Hewlett-Packard J.*, pp. 15-21 (May, 1977).

164. B. Konemann, J. Mucha, and G. Zwiehoff, "Built-In Logic Block Observation Techniques," pp. 37-41 in *Digest of papers, International Test Conf.*, IEEE, Philadelphia, PA. (1979).
165. B. Konemann, J. Mucha, and G. Zwiehoff, "Built-In Test for Complex Digital Integrated Circuits," *IEEE J. Solid-State Circuits*, Vol. SC-15, pp. 315-318 (June, 1980).
166. N. Benowitz, D. F. Calhoun, G. E. Alderson, J. E. Bauer, and C. T. Joeckel, "An Advanced Fault Isolation System for Digital Logic," *IEEE Trans. Computers*, Vol. C-24, pp. 489-497 (May, 1975).
167. D. K. Bhavsar and R. W. Heckelman, "Self Testing by Polynomial Division," pp. 208-216 in *Digest of papers, International Test Conf.*, IEEE, Philadelphia, PA. (1981).
168. B. T. Murphy, "Cost Size Optima of Monolithic Integrated Circuits," *Proc. IEEE*, Vol. 52, pp. 1537-1545 (Dec., 1964).
169. J. E. Price, "A New Look at Yield of Integrated Circuits," *Proc. IEEE*, Vol. 58, pp. 1290-1291 (Aug., 1970).
170. A. Gupta and J. W. Lathrop, "Yield Analysis of Large Integrated Circuit Chips," *IEEE J. Solid-State Circuits*, Vol. SC-7, pp. 389-395 (Oct., 1972).
171. C. H. Stapper, "Defect Density Distribution for LSI Yield Calculations," *IEEE Trans. Electronic Devices*, Vol. ED-20, pp. 655-657 (July, 1973).
172. R. S. Hemmert, "Poisson Process and Integrated Circuit Yield Prediction," *Solid-State Electron.*, Vol. 24, pp. 511-515 Pergamon Press Ltd., (1981).
173. K. Saito and E. Arai, "Experimental Analysis and New Modelling of MOS LSI Yield Associated with the Number of Elements," *IEEE J. Solid-State Circuits*, Vol. SC-17, pp. 28-33 (Feb., 1982).
174. S. C. Seth and V. D. Agrawal, "Characterizing the LSI Yield Equation from Wafer Test Data," *IEEE Trans. Computer-Aided Design*, Vol. CAD-3, pp. 123-126 (Apr., 1984).

175. C. H. Stapper, A. N. McLaren, and M. Dreckmann, "Yield Model for Productivity Optimization of VLSI Memory Chips with Redundancy and Partially Good Product," *IBM J. Research and Development*, Vol. 24, pp. 398-409 (May, 1980).
176. B. F. Fitzgerald and E. P. Thoma, "Circuit Implementation of Fusible Redundant Addresses on RAMs for Productivity Enhancement," *IBM J. Research and Development*, Vol. 24, pp. 291-298 (May, 1980).
177. S. E. Schuster, "Multiple Word/Bit Line Redundancy for Semiconductor Memories," *IEEE J. Solid-State Circuits*, Vol. SC-13, pp. 698-703 (Oct., 1978).
178. E. Tammaru and J. B. Angell, "Redundancy for LSI Yield Enhancement," *IEEE J. Solid-State Circuits*, Vol. SC-2, pp. 172-182 (Dec., 1967).
179. I. Koren and M. A. Breuer, "On Area and Yield Considerations for Fault Tolerant VLSI Processor Arrays," *IEEE Trans. Computers*, Vol. C-33, pp. 21-27 (Jan., 1984).
180. C. H. Stapper, F. M. Armstrong, and K. Saji, "Integrated Circuit Yield Statistics," *Proc. IEEE*, Vol. 71, pp. 453-470 (Apr., 1983).
181. J. Bernard, "The IC Yield Problem: A Tentative Analysis for MOS/SOS Circuits," *IEEE Trans. Electronic Devices*, Vol. ED-25, pp. 939-944 (Aug., 1978).
182. W. R. Moore, "A Review of Fault-Tolerant Techniques for the Enhancement of Integrated Circuit Yield," *GEC J. Research*, Vol. 2, pp. 1-15 (1984).
183. B. T. Murphy, "Comments on 'A New Look at Yield of Integrated Circuits'," *Proc. IEEE*, Vol. 59, p. 1128 (July, 1971).
184. G. E. Moore, "What Level of LSI is Best for You?," *Electronics*, Vol. 43, pp. 126-130 (Feb., 1970).
185. T. Yanagawa, "Yield Degradation of Integrated Circuits Due to Spot Defects," *IEEE Trans. Electron Devices*, Vol. ED-19, pp. 190-197 (Feb., 1972).
186. A. Gupta, W. A. Porter, and J. W. Lathrop, "Defect Analysis and Yield Degradation of Integrated Circuits," *IEEE J. Solid-State Circuits*, Vol. SC-9, pp. 96-102 (June, 1971).

187. W. R. Moore and M. J. Day, "Yield Enhancement of a Large Systolic Array Chip," *Microelectronics Reliability*, Vol. 24, pp. 511-526 (1984).
188. P. L. Meyer, *Introductory Probability and Statistical Applications*, Addison-Wesley Publishing Co., Reading, MA. (1970).
189. R. M. Sedmak, "Implementation Techniques for Self Verification," pp. 267-278 in *Digest of papers, International Test Conf.*, IEEE, Philadelphia, PA. (1980).
190. H. T. Kung and M. S. Lam, "Fault Tolerance and Two Level Pipelining in VLSI Systolic Arrays," pp. 74-83 in *Proc. Conf. Advanced Research in VLSI*, MIT, Cambridge, MA. (Jan., 1984).
191. R. W. Linderman and W. H. Ku, "A Three Dimensional Systolic Array Architecture for Fast Matrix Multiplication," pp. 34A.6 in *Proc. International Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, IEEE, San Diego, CA. (1984).
192. J. V. McCanny and J. G. McWhirter, "Yield Enhancement of Bit Level Systolic Array Chips Using Fault Tolerant Techniques," *Electronics Letters*, Vol. 19, pp. 525-527 (July, 1983).
193. I. Kale, "A CMOS Digital Polarity Correlator with Built-In Self-Test and Self-Repair," MSc. Project Report MSP26, University of Edinburgh (Sept., 1984).