Gallium Arsenide Design Methodology and Testing of a Systolic Floating Point Processing Element

by

Andrew J. Beaumont-Smith, B.E.(Hons)

A thesis submitted in fulfilment of the requirement for the degree of

Master of Engineering Science

The University of Adelaide
Faculty of Engineering
Department of Electrical and Electronic Engineering
Adelaide, South Australia

November 1995

Contents

Abstract

Despite the recent advances in computing performance, there remain many signal processing tasks that are beyond the capabilities of current off-the-shelf computing systems. These tasks include matrix intensive operations such as real-time digital Kalman filtering, signal processing and computer simulation algorithms for electronic circuits and mechanical structures and thermal system modelling. This dependence on $O(n^3)$ matrix operations leads to a requirement for a parallel computer architecture in the form of a multi–dimensional array of processing elements.

A general purpose matrix processing engine is described in this thesis which deals in particular with the implementation of a single processing element which forms part of a two dimensional processing array. The processing element performs addition, multiplication and multiplication – accumulation of two floating point numbers.

An architecture for a class of digit-serial systolic ring floating point processing element is investigated and a $0.8\mu m$ gallium arsenide implementation is realised using Vitesse HGAAS–II technology. Gallium arsenide technology was chosen to implement the processing element because of its high speed and low power advantages over conventional technologies such as silicon ECL. Studies were conducted to develop an optimised logic class for this technology. A mixed logic approach using DCFL (direct coupled FET logic), SDCFL (source follower DCFL), SBFL (super buffer FET logic) was used.

A new physical layout strategy '*ring notation*' was developed which was shown to be suitable for the design of high speed circuits using these classes of logic. This strategy achieves good power supply isolation from high speed signal interconnects and high packing density for these types of circuits.

A CAD environment for gallium arsenide was developed which includes the modelling of circuit parasitics, layout, circuit extraction and technology files. Circuit primitives were designed including flip–flops, adders and multiplexers.

Architectural studies were carried out to determine the optimum architecture for this technology. It is shown that the *area–time* metric should be used to optimise these processors.

A four bit per digit implementation of the systolic–ring floating point processing element was realised for an extended floating point format. A chip was successfully fabricated using the HGAAS-II process and measured $3mm \times 5.7mm$. It contained $12,000$ devices and has a maximum operating speed of $300MHz$, producing $11Mflops$ for multiply – accumulate operations. The chip was tested and found to be fully functional at $128MHz$ (due to process variation) to produce a computation rate of $5Mflops$.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text.

I give consent to this copy of my thesis, when deposited in the University Library, being available for loan and photocopying.

Signed: Date: 5 Feb 1996

Acknowledgements

Firstly, I would like to thank my supervisors, Dr. Kamran Eshraghian and Dr. Cheng–Chew Lim for their guidance and assistance with the work and the preparation of this thesis.

I would like to thank Dr. Warren Marwood for the enlightenment he has given me through the work and papers we have written jointly, for proof reading the thesis and countless interesting late night discussions.

Thanks also to my colleagues in the department and from other Universities including Mr. Ali Moini, Mr. Michael Liebelt, Dr. Jens Jakobsen (Jydsk Telefon, Denmark), Mr. Eric Chu, Mr. Mike McGeever, Mrs. Song Cui, Mr. Tim Shaw and Mr. Gyudong Kim (Seoul National University). Mr. Mike McGeever also assisted this work by characterising the I–V curves of the fabricated MESFET devices.

The support of The Australian Research Council and the Sir Ross & Sir Keith Smith Foundation is gratefully acknowledged.

Lastly I would like to thank my wife, Natalie for her endless patience and assistance in proof reading.

ABS

List of Publications

The following is a list of publications by the author and colleagues which are related to this thesis.

A. Beaumont-Smith, W. Marwood, C.C. Lim and K. Eshraghian. "Design and Implementation of a GaAs Systolic Floating Point Processing Element". *Submitted to IEE Proceedings-E, Computers and Digital Techniques*, 1995.

A. Beaumont-Smith, W. Marwood and C.C. Lim. "A CMOS Linear Systolic Processing Element". *Proc. 13th Australian Microelectronics Conference*, pp. 74-79, July 1995.

A. Beaumont-Smith, W. Marwood, K. Eshraghian and C.C. Lim. "The Gallium Arsenide Implementation of a Systolic Floating Point Processing Element". *Proc. 12th Australian Microelectronics Conference*, pp. 255-260, October 1993.

W. Marwood and A. Beaumont-Smith. "The Implementation of a Generalised Systolic Serial Floating Point Multiplier". *Proc. APCCAS'92, IEEE Asia-Pacific Conference on Circuits and Systems*, pp. 513-518, December 1992.

W. Marwood and A. Beaumont-Smith. "The Architecture and Optimisation of Systolic Ring Processors". *Proc. TENCON '92: IEEE Region 10 Conference*, pp. 735-739, November 1992.

W. Marwood, C.C. Lim, K. Eshraghian and A. Beaumont-Smith. "Systolic Matrix Processor Architecture for Very High Speed Signal Processing". *Proc. IREECON International Convention*, 1991.

A. Beaumont-Smith, W. Marwood, C.C. Lim and K. Eshraghian, "Ultra High Speed Gallium Arsenide Systems: Design Methodology, CAD tools and Architecture". *Proc. Microelectronics '91, I.E.Aust Conference*, pp. 85-90, June 1991.

Software:

A. Beaumont-Smith. "EXT2HSP - A conversion program from MAGIC to HSPICE for GaAs circuits", *The University of Adelaide*, Adelaide, 1992.

A. Beaumont-Smith. "GAASNET V2.0 - A gallium arsenide network extractor", *Integrated Silicon Design Pty.Ltd.*, Adelaide, 1991.

List of Symbols

ϵ_r	Relative Dielectric Constant
ϵ_{eff}	Effective Dielectric Constant
σ	Conductivity *or* Standard Deviation (for Process Parameters)
ρ	Resistivity
Z_0	Characteristic Impedance
W_G	Channel Length
L_G	Gate Length
d	Gate to Channel Spacing
β	Transconductance Parameter *or* Number Base
α	Saturation Factor
λ	Channel Length Modulation Parameter
γ	Drain Voltage Induced Threshold Voltage Lowering Coefficient
δ	I_{ds} Feedback Factor for TOM Model *or* Skin Depth
b	Critical Field for Mobility Degradation
I_{ds}	Drain to Source Current
I_{DSS}	Drain–Source Saturation Current at Zero Gate–Source Voltage
I_D	Diode Current
I_s	Diode Saturation Current
V_d	Diode Voltage
V_{gs}	Gate to Source Voltage
V_{ds}	Drain to Source Voltage
V_{TO}	Threshold Voltage
$VGEXP$	Gate Voltage Exponent
V_{ds0}	Drain to Source Voltage for the Curtice II model
N	Diode Ideality Factor
T	Temperature in Kelvin
$q = 1.60212 \times 10^{-19}$	Electronic Charge

$k = 1.38062 \times 10^{-23}$	Boltzmanns Constant
$\epsilon_0 = 8.854 \times 10^{-12} F/m$	Permittivity of Free Space
$\mu_0 = 4\pi \times 10^{-7} H/m$	Permeability of Free Space
r	Number of Bits per Digit
s	Sign Bit
m	Number of Mantissa Bits
e	Number of Exponent Bits
q	Number of Systolic Cells
g	Number of Guard Digits
k	Number of Digits in a Floating Point Operand
p	Order of a Square Systolic Array
n_c	Number of Systolic Cells in a Systolic Ring
n_d	Number of Digit Delay Cells in a Systolic Ring
C	Number of Circulations of Operands in a Systolic Ring
A_{proc}	Total Active Area of a Systolic Processing Array
A_{pe}	Active Area of a Processing Element
T_{pe}	Time to Process One Set of Operands in a Processing Element
B_{proc}	Systolic Processor Bandwidth

List of Abbreviations

Al	Aluminium
CIF	Caltech Intermediate Form
DAS	Digital Acquisition System
DCFL	Direct Coupled FET Logic
DFET	Depletion FET
ECL	Emitter Coupled Logic
EFET	Enhancement FET
ESD	Electro–static Discharge
GaAs	Gallium Arsenide
Gflops	Giga Floating Point Operations per Second
HEMT	High Electron Mobility Transistor
LCC	Leaded Chip Carrier
MATRISC	MATrix Reduced Instruction Set Computer
MESFET	Metal Semiconductor FET
Mflops	Mega Floating Point Operations per Second
MIPS	Mega Instructions per Second
PE	Processing Element
SAGA	Self–Aligned Gate
SBFL	Super Buffer FET Logic
SDCFL	Source Follower Direct Coupled FET Logic
Si	Silicon
SI	Semi Insulating
SIMD	Single Instruction, Multiple Data
SPICE	Simulation Program for Integrated Circuit Estimation
TEM	Transverse Electro-Magnetic
WSI	Wafer Scale Integration

List of Figures

List of Tables

Chapter 1

Introduction

1.1 High Performance Computing

Despite the recent advances in computing performance, there remain many tasks that are beyond the capabilities of current off–the–shelf computing systems such as signal processing [LoLi95], real–time control [Marw90a, KuHw91] and computer simulation for electronic devices [NaHi91]. This is due to the rapid growth in algorithm development for better task performance and the general purpose nature of computer architecture design, so there will remain a niche market for application specific processors in the for–seeable future.

The recent trends in computer architecture have been toward massively parallel processors based on Reduced Instruction Set Computer (RISC) nodes with support for vector processing. The support for matrix algorithms has been at the software level but matrix computation rates in the range of Gflops are not possible on general purpose RISC machines due to non–optimal processor architectures and software overheads for matrix algorithms. A natural extension for these architectures is to provide hardware support to a RISC processor for a set of matrix operations such as the matrix outer product, matrix addition and subtraction, element–wise matrix multiplication (Schur or Hadamard product), tensor product and tensor sum. The matrix product is an order n^3 ($O(n^3)$) operator whereas matrix addition, subtraction and the element–wise product are $O(n^2)$ operators. A matrix processor which supports these tasks has been proposed by Marwood *et.al.* [Marw94, MaLi91]. This processor is referred to as a MATrix Reduced Instruction Set Computer (MATRISC) due to similarities to the RISC philosophies. The core of the processor architecture is characterised by a two dimensional mesh connected systolic array of parallel processing elements (PEs) as shown in Figure 1.1 [Marw94]. The systolic array is fed operands on two wavefronts by address generators from a high bandwidth main memory. The address generators are a programmable device that map the matrix operands from the linear main memory into the systolic array. The RISC/CISC CPU is used to perform scalar operations on the data since the array has significant performance penalties for simpler operations. Caches are used to increase the performance of the system through re–use of input operands and results without accessing main memory. The system bus links the main memory to the address generators, caches and a host workstation.

Hardware supported matrix operations provide an architecture [Marw94] which :

- provides a well defined framework for problem definition and expression
- allows serial code to implement scalar, vector or matrix algorithms
- utilises a well defined RISC architecture to describe the parallel architecture
- improves on current implementations by orders of magnitude in performance using current technologies

The implementation of the PEs in the the two dimensional systolic array uses systolic ring techniques to achieve scalable floating point precision, size and computation time. The PE architecture is described in Chapter 3 and allows the possibility of different size and speed PEs to be used as function of their location in the systolic array. This is done by trading off the length of floating point representation and the number of computation cells with the time taken to complete a floating point computation. It is then possible to match the order of the array to the matrix problem and maintain constant bandwidth as faster elements are placed towards the origin of the array as indicated in Figure 1.2 [Marw94]. For example, an array of order $\frac{N}{2}$ would contain PEs with computation rates twice as fast as an array of order N to maintain constant memory bandwidth and hence completely utilise the available bandwidth when operating on matrix operands of order $\frac{N}{2}$.

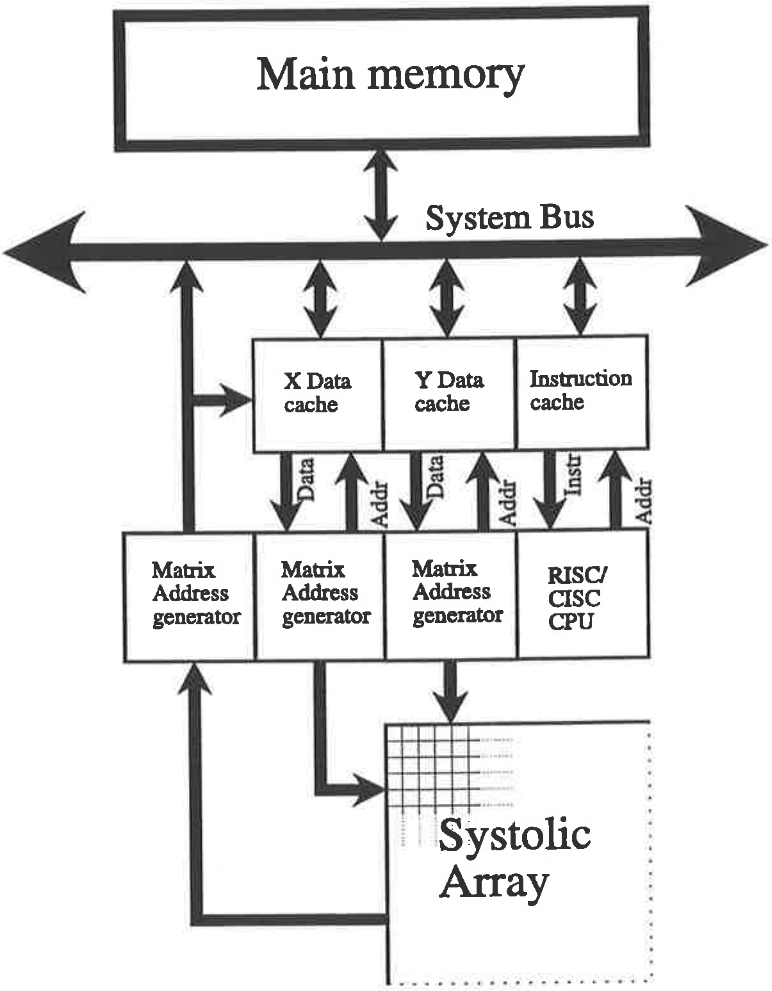


Figure 1.1: Matrix processor architecture (MATRISC).

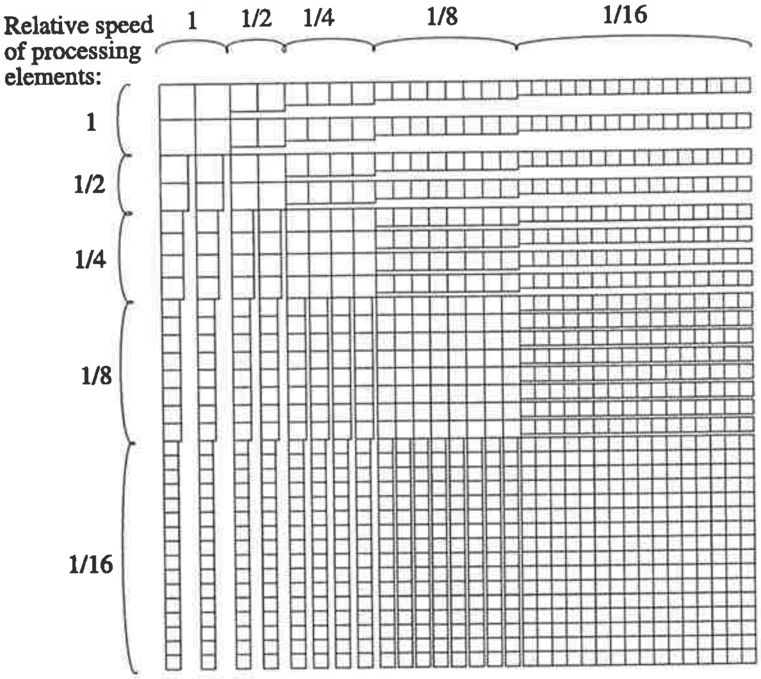


Figure 1.2: A constant bandwidth mesh connected systolic array.

Figures 1.3 and 1.4 show the simulated performance of a MATRISC processor presented in Marwood [Marw94]. The performance of the processor in Figure 1.3 is nearly doubled to 750*Mflops* by using a constant bandwidth array. The irregularities in the simulation are due to cache and memory behaviour. In Figure 1.4 a peak computation rate of 6 *Gflops* is achieved for an array of order 40.

For such a processor to support high computation rates, the technology for realisation of the PEs has a significant impact on the architecture and hardware realisation. Gallium Arsenide (GaAs) Metal Semiconductor FET (MESFET) technology is a contender for implementing high performance electronic components to rival silicon Emitter Coupled Logic (ECL) in speed and power performance [LoBu89, Eshr91]. Integration densities have also improved dramatically in recent years and more than one million devices may be integrated on a single chip in a gate array [Vite92].

In the following sections an overview of systolic arrays, matrix processing algorithms and an introduction to GaAs technology is presented.

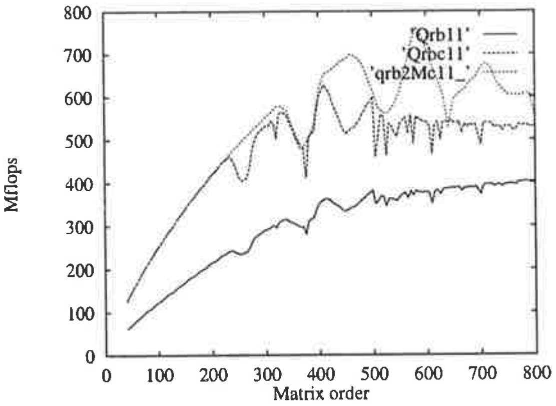


Figure 1.3: A performance comparison of a conventional systolic array and a constant bandwidth array when implementing a block QR factorisation algorithm.

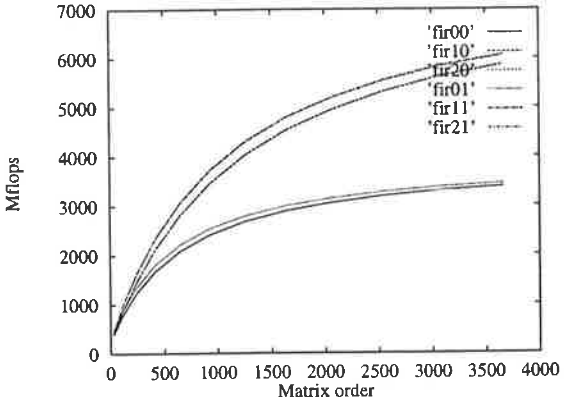


Figure 1.4: Simulated performance for FIR filters implemented on an order 40 MATRISC processor.

1.2 Systolic Processing for Matrix Computations

Systolic arrays have regular and modular structures that match the computational requirements of many algorithms. Their implementation requires that a wealth of subsumed concepts and engineering solutions be mastered and understood.
– J.Fortes and B.Wah, 1987: p.12 [FoWa87].

This statement reflects the fact that systolic array design covers many inter–related disciplines including mathematics, VLSI and Computer Architecture. Systolic arrays were first proposed by Kung and Leiserson in 1978 [KuLe78] as a parallel processing technique. Since this time many different systolic array algorithms and architectures have been proposed for applications including matrix arithmetic, signal processing, image processing, language recognition, relational database operations, data structure manipulation and character string manipulation [JoHu93, Kung88]. The term 'systolic' refers to the way data is pipelined rhythmically along the communication channels between an array of nodes. The nodes that the data visit may be arranged in a single– or multi–dimensional array with a fixed or configurable interconnection structure. The idea is to re–use data already entered into the systolic array as it passes through the pipeline to achieve very high computation rates. In this way, compute bound problems can be processed much faster than traditional computer architectures since each operand is re–used inside the array and not fetched and written to memory each time it is used. This has an advantage over conventional processor architectures by using a substantially lower memory bandwidth. This implies that the computation performance of bandwidth limited systems can be increased through the use of systolic arrays.

1.2.1 Algorithms and Architecture

Examples of matrix based systolic array architectures include the inner product step processor of Kung and Leiserson [KuLe78] which was a fundamental building block that could be configured as a linear, orthogonally– (mesh) or hexagonally–connected array. The proposed matrix based algorithms included matrix–vector multiplication using a linearly connected network, matrix multiplication using a hexagonal array and LU–decomposition of a matrix using a hexagonal array [MeCo80]. An inner product accumulate systolic cell connected in a mesh was proposed by Whitehouse and Speiser [WhSp81] in 1981. This was called the engagement processor and each cell at position i, j in the array computes the inner product, c_{ij} which is stored in each cell and given by

$$c_{ij} = \sum_{k=1}^{N} a_{ik} b_{kj}$$

where a_{ik} and b_{kj} are the elements of the matrices A and B. A 3×3 engagement processor is shown in Figure 1.5.

The efficiency for this systolic array approaches 100% for large matrices under the assumptions that end effects of the start and end of the matrix operation are ignored.

1.2.2 Granularity

The operation performed in each cycle by a PE can range from a bit–wise to a word–level operation such as multiplication and addition. This is the granularity of the systolic array. Bit level systolic arrays have low (or fine) granularity and use bit–serial data transfers between PEs. This work, as reported in [DeRe85] has a low Input/Output (I/O) requirement which is attractive for I/O constrained systems, however this limits the throughput of operands if they contain a large number of bits such as floating point numbers. Digit

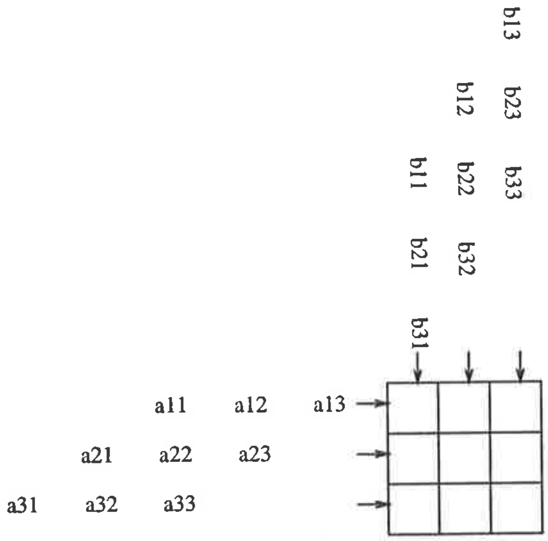


Figure 1.5: A 3 × 3 engagement processor with input matrices A and B.

serial arithmetic however is an area–time efficient method of performing high speed arithmetic calculations [HaCo90, CoHa92]. The digit size can be appropriately chosen for the throughput to match the design needs. Configurable systolic architectures such as the Configurable Highly Parallel (CHiP) computer [Snyd82] overcome the difficulty of being limited to a fixed array architecture as the connections between PEs are configurable through switches to suit the algorithm. Special–purpose single chip processors which can be used to build systolic arrays such as the Programmable Systolic Chip [FiKu83] and the Orthogonal Multi–Processor [HwDu90] based on i860 processors are flexible in the types of algorithms that can be run but too complex to implement since each node must be programmed. These are examples of systems with high (or coarse) granularity.

1.2.3 Implementations and Issues

Since systolic arrays are regular two dimensional structures of nodes or PEs, they lend themselves very well to Very Large Scale Integration (VLSI) implementation. This presents a number of VLSI design challenges since the order of systolic arrays tends to be large and the connections between the nodes can be complex as in the case of a reconfigurable array. Implementation techniques such as Wafer Scale Integration (WSI) and Multi–Chip Modules (MCM) have been used for integrating large VLSI systems [DoFr93].

These bring the circuits physically closer together which enables them to achieve higher throughput.

WSI is the process used to integrate and interconnect circuits on a single processed wafer where the individual chips are not cut from the wafer and packaged. MCMs are a structure housing two or more integrated circuits electrically connected to a common circuit base and interconnected by conductors in that base. WSI is simpler for the construction of large systolic arrays all using the same technology but the number of defects in the silicon substrate is constant. This implies a high probability that at least one node will not work in a WSI system, so fault tolerance through redundancy or array reconfiguration is required. The identification of a non functional PE using self testing and the replacement and/or bypassing of it become design considerations as the order of the systolic array grows [LiJe89]. Kanopoulos [Kano85] describes a bit–serial systolic array for signal processing applications that uses a self testing scheme and a voting circuit to identify single permanent faults and isolate a particular storage or arithmetic unit during processing. This increases the control overhead and circuit complexity ultimately resulting in a performance penalty. Furthermore, integration of memory chips and specialised processor chips is not possible using WSI if different process technologies are used. MCMs however are more difficult and expensive to construct but can be repaired since individual components may be replaced. Another advantage of MCM technology is being able to test chips before they are assembled.

A recent MCM implementation of a systolic array for matrix computation was the SCalable Array Processor (SCAP) [ClCl92, Marw94, MaCl95] which is the first systolic co–processor subsystem to implement the set of matrix operations {*multiplication, addition, element–wise multiplication, transposition, permutation*}. SCAP uses an IEEE single precision floating point data format and is coupled to a SUN SPARCstation 1. The processor module has four hundred 1.0*Mflops* processors and the system can perform matrix products around 150 times faster than a SUN SPARCstation 1. The scalable array of PE chips and data formatting chips were implemented in $1.2\mu m$ CMOS. Each PE chip contained a 4×5 array PEs in a mesh connected array. The ceramic MCM was constructed using MCM–C technology and contained a 5×4 array of PE chips which were wire bonded in place. A failure rate analysis of the completed MCMs gave a yield of around 30%. Given the high cost in producing each processor array, there was a high probability that at least one PE chip would have to be replaced or repaired. It is not possible to repair WSI systems in this manner.

Hein *et.al.* [HeZi87] report on the design of a GaAs systolic array for an adaptive null steering beamformer. The processing array was configured as a SIMD machine and specially designed parts included a 32–bit GaAs ALU, a 500*M–bits per second (bps)* Manchester encoder, a 200 MIPS, RISC, 8–bit microprocessor and a Manchester decoder. The clock rate was 120*MHz* and the system updates the coefficients for the multiple beams every $5\mu s$. This real–time performance was unmatchable at that time by any realisable uniprocessor system.

Fouts and Butner [FoBu91] proposed GASP, a GaAs supercomputer which contained a hexagonally connected homogeneous systolic array of PEs. The design uses 32–bit integer arithmetic and has a peak computation rate of 30,000 MIPS with 65 PEs. A MCM solution was proposed for integration of GASP with a 500*MHz* instruction issue clock and 1*GHz* subsystem clocks. System simulation predicted an improvement in performance by a factor of 8.3 and 457 over a Sun 4/280 for heap sorting and Gaussian elimination algorithms, respectively. Problems with the design were identified as the availability of high density and high speed (2*ns*) RAMs and relatively low pin count on hybrid modules limit data transfer rates. The processor dissipates 680 *W* and requires refrigerant cooling.

Most systolic array designs to date use integer arithmetic in the PEs. Some exceptions include the SAXPY Matrix–1 [FoSc87] and the Warp computer [AnAr87]. For general purpose use and to be compatible with existing RISC workstations, the PEs of the proposed MATRISC processor need to comply with the IEEE–754 floating point standard [ieee85].

1.3 Gallium Arsenide Technology

Gallium arsenide is the technology of the future, always has been, always will be.
(– humorous moment at a conference, source unknown, circa 1989.)

The group III–V compound gallium arsenide (GaAs) was first discovered in 1926, but its high speed potential as a semiconductor was not realised until the 1960's [PuEs88]. The first GaAs analogue products appeared in the 1970's with the development of IC fabrication and the advances in ion implantation in the 1980's have made digital GaAs VLSI technology a commercial reality in the 1990's. GaAs will not be a technology for mainstream computer and systems applications in the forseeable future due to continuing advances and research and development support for CMOS technology. The characteristics of GaAs make it suitable for specific niche applications where it shows a clear advantage over silicon implementations. These applications include communications devices such as an optical fibre front end that processes high speed serial data, automotive sensors and specialised high speed computers [Dyks90] such as the CRAY–3 [KiHe97]. Figure 1.6 shows a comparison of the speed versus power characteristics for GaAs, CMOS, BiCMOS, nMOS and ECL technologies. This shows that GaAs would be a favourable choice of technology where outright speed or speed and power are critical design parameters for a particular problem.

The advantages of GaAs material over silicon include: [LoBu89, Eshr91, Vite92, TriQ91, Rocc90, Giga91, Beau93]

- a six to seven times higher electron mobility than silicon. MESFETs with typical gate lengths of around $0.8\mu m$ with transit times as small as 10 to $15ps$ produce current gain–bandwidth products in the range of 15 to $25GHz$. This is a three to five times improvement over silicon.
- smaller interconnect capacities than silicon as a consequence of the substrate being a semi–insulator rather than a semiconductor.
- higher electron saturation velocity at lower electric field strengths than silicon. This

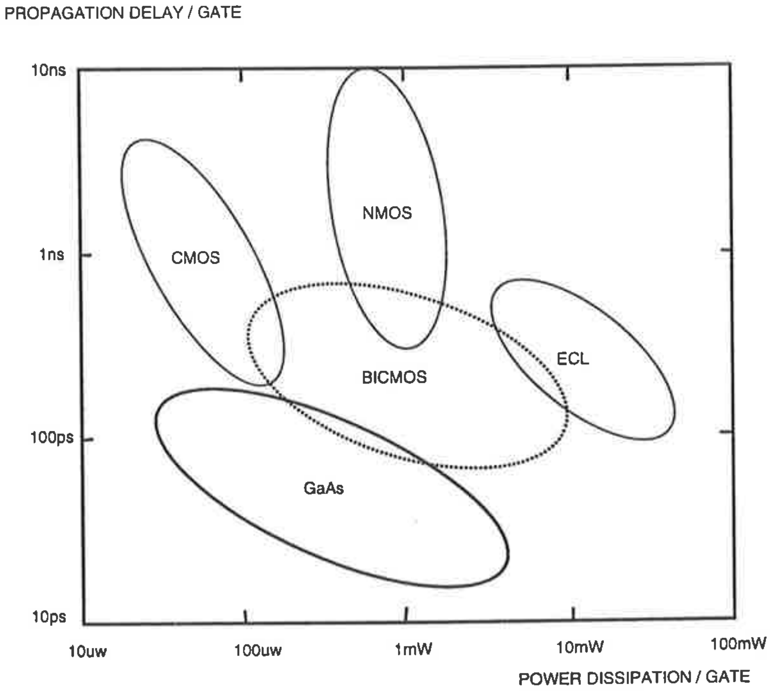


Figure 1.6: Speed versus power for GaAs, CMOS, BiCMOS, nMOS and ECL technologies.

implies faster switching speeds and up to a 70% reduction in power dissipation over silicon ECL.

- smaller speed–power product than silicon (ECL).
- the direct bandgap of GaAs allows the efficient radiative recombination of carriers. This provides a mechanism for integrated high bandwidth optical communications.
- there is no gate oxide to trap charges which makes the device more ionising radiation resistant than silicon. This is of benefit in space–borne applications.
- GaAs devices are more temperature tolerant due to the larger bandgap ($1.42eV$) of the material.

Disadvantages associated with GaAs circuits lie in problems with the device physics which mainly result in high fabrication costs and low yield. The disadvantages and reasons include:

- lower yield than achievable with silicon due to a large density of dislocations in the crystal lattice structure, brittleness of wafers and difficulty in controlling doping and threshold voltage over the wafer. Threshold voltage should be controlled to less than $20mV$ between devices, otherwise the circuit may become inoperable.

- there is no gate oxide in a MESFET to isolate the gate, therefore the gate may only be forward biased to around 0.7 to $0.8\,V$ before large currents begin to flow. This limits the voltage swing for many logic classes and makes them incompatible with other technologies such as CMOS.
- different device offset voltages produce an intrinsic bias in different parts of the circuit which degrades both the noise margin of logic circuits and the yield. Device offset in GaAs is caused by threshold variation, component mismatch and low frequency (1/f) noise. Material non–uniformity and threshold variation are the main contributors.
- backgating or sidegating in GaAs circuits cause a reduction in the drain current of a device when the substrate (backgate) or neighbouring device (sidegate) is biased negatively with respect to the source of that device. This causes an increase in the size of the space–charge layer at the channel/substrate interface which shifts the threshold voltage of the device higher. It is also dependent on the distance between active devices and can been reduced through ion implantation and mesa etching around active regions. The solution is to space devices farther apart which is not good either for device matching or for increasing circuit density.
- drain current hysteresis effects due to charges stored in the substrate traps. These are frequency dependent and have the most influence at frequencies less than $100Hz$.

Table 1.1 summarises the physical characteristics of GaAs and silicon [Glon88, Sze83].

1.3.1 Gallium Arsenide Devices

There are two distinct generations of GaAs devices [PuEs88]. First generation devices have typical switching delays of $70ps$ for a simple inverter and power dissipations around 0.1 to $0.2\,W$. GaAs devices include:

- depletion and enhancement mode MESFETs,

Physical Property	GaAs	Si
electron mobility (cm^2/Vs)	5000	800
maximum electron drift velocity (cm/s)	2×10^7	1×10^7
hole mobility (cm^2/Vs)	250	350
energy gap (eV)	1.42	1.12
gap type	direct	indirect
density of states in conduction band (cm^3)	5×10^7	3×10^{19}
maximum resistivity (Ωcm)	10^9	10^5
minority carrier life (s)	10^{-8}	10^{-3}
breakdown field (V/cm)	4×10^5	3×10^5
Schottky barrier height (V)	$0.7 - 0.8$	$0.4 - 0.6$

Table 1.1: Comparison of GaAs and silicon physical characteristics

- enhancement mode JFET,
- complementary enhancement mode JFET.

First Generation Devices

Complementary GaAs logic suffers from a poor P–type transistor due to its low mobility so high performance logic has been restricted to normally–off and normally–on classes employing MESFETs. The two types of MESFETs are enhancement and depletion mode. The enhancement mode MESFET has a positive threshold voltage and depletion mode MESFETs have a negative threshold voltage. In the fabrication process for MESFETs conductive transistor channels are formed by implanting silicon atoms into the substrate. A two step implant scheme is used to pattern the channels for enhancement and depletion mode devices where the threshold voltage is adjusted by the depth of the N- implant in the channel region. A refractory metal is then deposited to form the gate and a high dose $N+$ implant is used to lower the resistance in the source and drain regions. Ohmic contacts for the source and drain connections are subsequently formed. The proceeding steps include the patterning and deposition of dielectric films and interconnect metalisation. Since MESFET fabrication is a planar process, up to four layers of aluminium interconnect are able to be used with some processes. The GaAs MESFET has a similar lithographical process to silicon and the metalisation process is identical (except for airbridges). Eleven mask steps are needed for a three metal process which is half the number of mask layers required for silicon ECL. Further information on fabrication techniques for MESFETs has been presented elsewhere [PuEs88, TriQ91, Rocc90, LoBu89, Vite92, Giga91].

The MESFET is the most mature GaAs device and millions of devices are able to be integrated onto a single chip [Vite92] and many standard MESFET based products are available for applications such as encoding/decoding, multiplexing, crosspoint switches and gate array products.

Second Generation Devices

Second generation devices include the High Electron Mobility Transistor (HEMT) and Heterojunction Bipolar Transistor (HBT). These devices have different structures to achieve up to five times higher electron mobility than the first generation. For example, typical depletion mode Pseudo–morphic HEMT (PM–HEMT) devices have short circuit current gain–bandwidth products (f_t) of around 50 to $100GHz$. Of these devices the HEMT holds the most promise for future digital GaAs implementations. The HEMT device was first developed in 1980 [HiLa86] and has since progressed rapidly. These sub–micron gate devices have less than $10ps$ switching delays at $300K$ with a typical f_t of around $70GHz$ for quarter micron devices. HEMTs exploit the superior transport properties of electrons moving along the heterojunction interface between two lattice matched compound semiconductor materials which have been grown using molecular beam epitaxy. HEMTs are also known as two dimensional electron gas FET (TEGFET), modulation doped FET (MODFET) and selectively doped heterojunction transistor (SDHT), depending on the process or resultant device characteristics. They offer superior gain and speed among most known semiconductor devices [HiLa86].

Other more recent devices include the semiconductor–insulator–semiconductor FET (SISFET) and heterostructure insulated–gate FET (HIGFET). HEMTs are suitable for use in analogue MMIC and high speed digital circuits and have performance benefits over GaAs MESFETs including low noise, low power and high speed. Logic classes applicable to HEMT include high/low power buffered FET logic (BFL), direct coupled FET logic

(DCFL, for E/D mode processes), high/low power for both E/D and D mode source coupled FET logic (SCFL) and capacitor enhanced logic (CEL). LSI/VLSI applications are possible and some circuit applications employing HEMT devices include multipliers [Ber91, TaNi92] , static RAMs (SRAMs), ALUs and demultiplexers/multiplexers [Nowo91]. Other applications include high definition television and optical fibre telecommunication systems such as SONET.

Both generations of GaAs devices employ Schottky barrier diodes for tasks such as logic level shifting and ESD protection. Schottky barriers can be made with metals such as aluminium, platinum and titanium. They have low reverse currents ($< 1A/cm^2$) and high ideality factors (< 1.1).

Digital HEMT processes are not as mature as the digital MESFET processes and VLSI circuits suitable for computer systems applications are still currently the domain of the MESFET. The remainder of this thesis focuses on the Enhancement/Depletion (E/D) MESFET process.

1.4 Contribution of the Thesis

Due to the high speed nature of GaAs, diversions from traditional VLSI design principles are required at the transistor and chip architectural level. In deriving architectures suitable for high speed systems, the following problems must also be addressed:

- selection of logic classes suitable for high speed systems
- transmission line modelling of longer interconnects
- minimisation of crosstalk and inductive spikes through architecture design
- clock distribution across large chips, guaranteeing synchronism
- multi–chip interconnection
- thermal management of chips

The research in this thesis addressed these issues and the main contributions include:

- an evaluation of digital GaAs MESFET logic classes
- characterisation and modelling of GaAs interconnects and parasitics
- the design and optimisation of a matrix processor suitable for implementation in GaAs
- the design of a floating point systolic PE in GaAs
- testing of the high speed GaAs PE

1.5 Outline of the Thesis

Chapter 2 presents a discussion of digital GaAs circuits, design methodology and circuit models. GaAs MESFET devices and digital GaAs logic classes suitable for implementation are reviewed. A layout strategy called '*ring notation*' is developed for the physical layout of the circuit primitives. An analysis of GaAs interconnect structures and parasitics is presented to characterise the process for accurate modelling and simulation. CAD tools are discussed which were developed to help facilitate GaAs circuit design by modification of existing silicon CAD tools.

In **Chapter 3** digit–serial multiplication is reviewed and a parallel digit–serial multiplier presented for use in a systolic multiplier cell. A class of systolic ring PE is proposed using the systolic multiplier cell for floating point multiplication and accumulation of two input operands. A performance metric is derived by minimising the job time for a matrix product on a two–dimensional mesh connected array of PEs. The performance metric is evaluated for a target GaAs technology and number representation to determine the optimum PE architecture. The required memory bandwidth for an array of PEs is also discussed.

Chapter 4 presents the circuit design, layout, simulation and implementation of the GaAs systolic ring PE. Basic circuit elements including data flip–flops, a full adder, a toggle flip–flop and multiplexers are designed, simulated and a layouts produced. These are used to build the parts of the PE including the systolic cell, ring controller, flag generator, I/O multiplexer. A variable frequency clock generator is implemented to allow testing of the chip at different clock rates. A clock and power distribution system is also designed and implemented. The chip floorplan is presented and finally details of the fabricated PE chip.

Chapter 5 reviews the testing of the systolic ring PE chip. A test fixture is designed and constructed to facilitate testing of the PE chip. MESFET test devices are measured to characterise and verify the models used in the design of the PE chip. A test procedure is developed for low and high speed functional testing of the PE using a digital test system. This test procedure is subsequently used to verify the operation of the PE chip. The clock generator output frequency is measured over its range of operation and as a function of supply voltage to further characterise the process.

Chapter 6 presents a discussion of the work presented in the thesis and future work.

Chapter 2

Gallium Arsenide Technology

This chapter presents a discussion of digital GaAs circuit simulation and design methodology as the basis for designing the processing element chip. GaAs MESFET models are reviewed and digital GaAs logic classes suitable for implementation are presented and optimised for speed, area and noise margin. Circuit parasitics are also investigated and transmission line models are discussed for high speed interconnections on a GaAs substrate. Layouts of circuit primitives are designed using these results.

2.1 GaAs MESFET Device Modelling

GaAs digital circuits have small voltage swings and accurate modelling in all regions is required to predict circuit performance [LoBu89, Wing90]. The GaAs MESFET is characterised by operation in several regions; *cutoff* where the channel is pinched off by the gate depletion region and no drain current flows, *linear* region where the behaviour is similar to a resistor and *saturation* where the behaviour is similar to a current source due to velocity saturation of the electrons in the channel. The *inverse* and *subthreshold* regions are of secondary importance.

The physical parameters which make up the MESFET models may not correspond to the parameters in the models because a purely algebraic representation of the device has been used to curve fit the actual characteristics. The models which are used for computer simulations must be continuous in their derivatives for the result to converge.

Mathematical models suitable for simulation of the MESFET have been developed for use in SPICE–like circuit simulators using the JFET equations as a basis. Figure 2.1 shows the MESFET model that is most commonly used. The Curtice model [Curt80] uses a hyperbolic tangent function which fits all regions of the model and is continuous in all of its derivatives. The most developed equation is the Statz–Raytheon model [StNe87] which includes modelling effects due to velocity saturation.

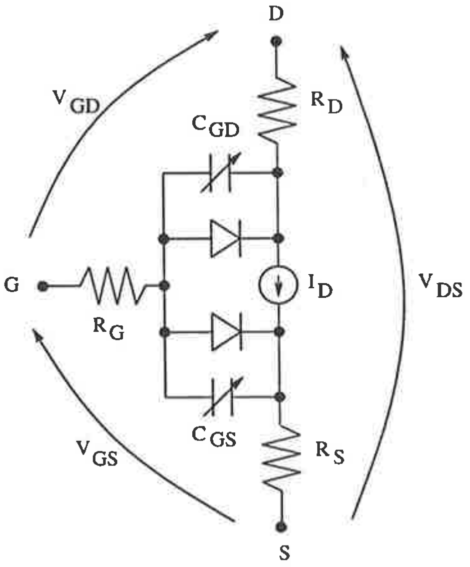


Figure 2.1: GaAs MESFET equivalent circuit.

2.1.1 Drain Current

Modelling of MESFET drain current characteristics is performed by curve fitting a formula to data and a range of MESFET compatible models are commonly available in most SPICE simulators. The drain current is set to zero for the cutoff region ($V_{gs} < V_{TO}$) and the equations for the linear and saturation regions are given below. A description of the parameters in the equations is given in the "List of Symbols" at the start of this thesis.

- Curtice model [Curt80]

$$I_{ds} = \beta(V_{gs} - V_{TO})^2(1 + \lambda V_{ds})tanh(\alpha V_{ds})$$

- Curtice model with user–defined gate voltage exponent and V_{gs} in the hyperbolic tangent function

$$I_{ds} = \beta(V_{gs} - V_{TO})^{VGEXP}(1 + \lambda V_{ds}) tanh(\alpha \frac{V_{ds}}{(V_{gs} - V_{TO})})$$

- Statz–Raytheon model [StNe87]

  For $V_{ds} < \frac{3}{\alpha}$

$$I_{ds} = \frac{\beta(V_{gs} - V_{TO})^2}{(1 + b(V_{gs} - V_{TO}))}(1 + \lambda V_{ds}) \left[1 - \left(1 - \alpha \frac{V_{ds}}{3}\right)^3\right]$$

  For $V_{ds} > \frac{3}{\alpha}$

$$I_{ds} = \frac{\beta(V_{gs} - V_{TO})^2(1 + \lambda V_{ds})}{(1 + b(V_{gs} - V_{TO}))}$$

- Meta software variable saturation model [Meta92]

  This is the same as the Statz–Raytheon model except more flexibility has been allowed by parameterising the gate voltage exponent ($VGEXP$) and the saturation exponent which is '3'. A more flexible model can be made by building a hybrid model from the Curtice, Statz–Raytheon and TriQuints Own (TOM) models [Goli91]. Note that Golio [Goli91] states that the saturation function is the hyperbolic tangent function which is contrary to the HSPICE manual, although the cubic saturation function is a truncated Taylor series representation of the *tanh* function.

$$I_{ds} = \frac{\beta(V_{gs} - V_{TO} - \gamma V_{ds})^{VGEXP}}{(1 + b(V_{gs} - V_{TO} - \gamma V_{ds}))}(1 + \lambda V_{ds}) \left[1 - \left(1 - \alpha \frac{V_{ds}}{3}\right)^3\right]$$

- TOM [McCa90]

  For $V_{ds} < \frac{3}{\alpha}$

$$I_{ds} = \frac{I_{ds0}}{(1 + \delta V_{ds} I_{ds0})}$$

  where $I_{ds0} = \beta(V_{gs} - V_{TO} + \gamma V_{ds})^{VGEXP} \left[1 - \left(1 - \alpha \frac{V_{ds}}{3}\right)^3\right]$

It is unclear whether the equation has been implemented correctly in HSPICE because the manual [Meta92] does not state any formula. HSPICE may have implemented the *tanh* function or the cubic approximation. The original paper states the former.

The common I_{ds} and capacitor parameters for the Statz capacitor model may be made independent of the Ids model. Some researchers such as Golio [Goli91] have *incorrectly* stated the TOM model by putting the hyperbolic tangent function outside of the feedback equation for I_{ds}. In the original paper [McCa90], this function is inside the feedback equation.

2.1.2 Diode

There are two Schottky gate diodes, one from gate to source and the other from gate to drain in MESFET devices. The diode current is given by:

$$I_D = I_s[exp(qV_d/NkT) - 1]$$

2.1.3 Parasitic Capacitances

Capacitors in the MESFET model characterise charge storage within the physical device which provides information about the transient operation and ultimate speed of a circuit. Figure 2.2 shows the physical interpretation of the MESFET parasitic capacitances with an exploded view of a typical device. The two major capacitances associated with a MESFET are the gate to source and gate to drain capacitances because the storage of charges in the gate to source and gate to drain depletion region and are non linear. For MESFET devices, the Statz–Raytheon capacitance model [StNe87] is a function of V_{ds} and V_{gs} and is an accurate analytical expression to use for large signal analysis [McCa90]. This model provides symmetric modelling of the gate to source and gate

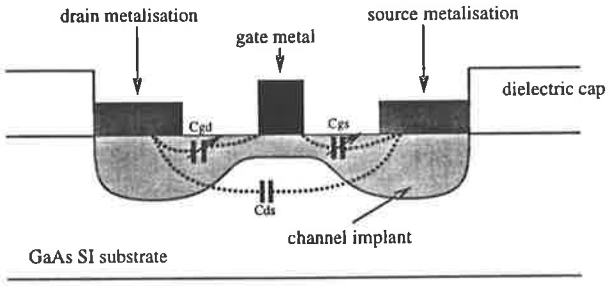


Figure 2.2: Cross section of a MESFET device.

to drain capacitances and is more accurate than the Curtice model [Curt80] which is a function of V_{gs} only. Other capacitances are the fringing capacitance from the gate depletion region to the source and drain because of the depletion layer extending beyond the edge of the gate, the smaller the gate length, the larger these capacitances will become. They are modelled by fixed capacitances connected from the gate to the source and drain intrinsic terminals. HSPICE does not accurately model the gate to source and gate to drain capacitance values at all bias conditions, therefore HSPICE can only be used up to $2GHz$.

2.1.4 Parasitic Resistances

The parasitic resistances of the MESFET occur in series with the drain and source connections. The physical interpretation is the resistance formed at the connection of the metal 1 layer to the diffusion via a low resistance ohmic contact. To minimise these resistances the source and drain connections must be placed as close to the gate as possible. In early models the values of the drain and source resistances were fiddled to enable a better fit to the MESFET characteristics but this caused the drain resistance to become quite large and the source resistance to drop to zero. This means that the symmetry of the device is lost (JFET model). However, later models overcame this deficiency (Statz–Raytheon Model [StNe87]) and the source and drain resistances regained their physical meaning. The series gate resistance is so small that it is often ignored.

2.1.5 Second Order Effects

Second order effects of Gallium Arsenide circuits are not necessarily taken into consideration for modelling purposes. Second order effects are listed below.

1. Backgating or sidegating is a similar to the body effect in MOSFETs. Sidegating is caused by negatively biased neighbouring FETs which cause the threshold voltage to increase and therefore reduce the drain current. Sidegating has the same effect but is due to a negative substrate bias. All active devices are susceptible to these effects including FETs, diodes and resistors. Even a horse shoe shaped resistor exhibits self sidegating although a positively biased guard ring around such devices can help. Sidegating is dependent on the distance between active devices and it has

been reduced through ion implantation and mesa etching around active regions. The backgating or sidegating effect cannot be modelled currently by extraction from a layout since it depends on the distance and relative potentials of surrounding active regions. TriQuint [TriQ91] claim that the backside metal should be biased at the highest power supply potential of the circuit and to allow $3\mu m/V$ separation between devices to help reduce this effect. The threshold voltage decreases rapidly below $10°C$ so worst case modelling would be at low temperature and high power supply voltages. Prediction of the backgating voltage is unreliable due to large variations in backgating from substrate to substrate. Thus backgating is ignored for low to moderate power supply voltages and higher than room temperature modelling. A constant substrate bias (backgate) may be specified and can be used with a model parameter, $K1$, in HSPICE to shift the threshold voltage of the simulated devices.

2. Drain current transient lag effects are due to deep level traps in the substrate below the channel which accumulate electrons injected into the substrate. It takes longer for the traps to release electrons than to capture them, hence the effect of overshoot in the drain current and slow recovery to a step in the drain to source voltage. This effect is also observed as an increase in the small signal output conductance by a factor of as much as three when the FET is in saturation. This is because the traps under the channel shield the drain to channel capacitance. This frequency dependent effect is modelled by changing the parameter LAMBDA in the MESFET model for the high and low frequency case. The effect is to increase LAMBDA for the high frequency case which increases the slope of the I-V curve in the saturation region. The function of variation of the drain–source characteristics with frequency is not a simple function. This also has a smaller effect of increasing the transconductance with increasing frequency although this is limited to a few percent and is not modelled. A higher quality substrate material with less traps would reduce these effects.

3. Subthreshold current flows from drain to source when the gate to source voltage is below the pinch off voltage. This occurs when the electrons are transported across the channel by diffusion and drift. The subthreshold current has little influence

on the DCFL circuits which spend most of their time operating in the saturation region.

4. Temperature dependence is characterised by two physical effects; the variation of the built in voltage of the channel/substrate interface and the channel transconductance factor, β. They both affect the threshold voltage of the MESFET but the built–in voltage increases the threshold as temperature is increased and β decreases the threshold voltage as temperature is increased. The net effect on the threshold voltage (and hence the drain current depends on the gate to source voltage) is complex and is not modelled. However, different libraries of models have been formulated to model devices at specific temperatures, usually two extremes such as $0°C$ and $125°C$. If operation at any other temperatures is required interpolation of the results would be the best option.

2.1.6 Simulating Worst Case

Worst case conditions for a particular device may include a selection of a poor parameter or group of parameters that cause poorer device operation and hence a reduction in voltage swing, noise margins and speed of operation. Worst case parameters are usually determined at $0°C$ and $125°C$ and characterised by variations in threshold voltage (either positive or negative), degraded transconductance (smaller), resistance (larger) and capacitance (larger). This is opposed to nominal circuit conditions where parameters are determined at room temperature (around $25°C$) which are used to initially verify the functionality of the circuit.

2.2 GaAs MESFET Logic Classes

Normally–on logic classes use only depletion mode MESFETs and typically require some voltage level shifting of the gate output to be compatible with the next stage. Larger supply voltages are needed than with the normally–off logic classes and so the power dissipation is higher. The complexity is also generally higher in normally–on logic classes but the speed may be greater than normally–off logic classes. Level shifting may be done by using Schottky diodes. Normally–on logic classes include [LoBu89, PuEs88, Eshr91, KaNa85, Wing90]:

- Buffered FET Logic (BFL)
- Capacitively Coupled Domino Logic (CCDL)
- Capacitor Coupled FET Logic (CCFL)
- Capacitor Diode FET Logic (CDFL)
- Feed–Forward Static Logic (FFSL)
- Inverted Common Drain Logic (ICDL)
- Schottky Diode FET Logic (SDFL)
- Source Coupled FET Logic (SCFL)
- Two–Phase dynamic FET Logic (TDFL)
- Unbuffered FET Logic (UFL)

Normally–off logic uses enhancement type MESFETs as a switch and depletion type MESFETs (or a resistor) as a load. Normally–off logic classes include [LoBu89, PuEs88, Eshr91, Wing90]:

- Direct Coupled FET Logic (DCFL)
- Feedback FET Logic (FBFL)
- FET FET Logic (FFL)
- Junction FET Logic (JFL)
- Pseudo Current Mode Logic (PCML)
- Quasi FET Logic (QFL)
- Super Buffer FET Logic (SBFL)
- Source Follower Direct Coupled FET Logic (SDCFL)
- Source Follower FET Logic (SFFL)

Other logic classes include Differential Pass Transistor Logic (DPTL) which has a low density due to its differential nature and the frequent buffering required. The logic classes above can be further broken down into static and dynamic logic. Dynamic logic has a minimum frequency of operation although the gates may be low power and simpler but static logic can operate down to DC. One requirement of the chip is complete testability which should be operable at clock frequencies down to DC, therefore a dynamic approach is not suitable.

The H–GAAS II E/D MESFET process, supplied by Vitesse Semiconductor Inc., USA and Thomson–CSF Semiconducteurs Specifiques, France (as a second source) has been tuned for using Normally–off DCFL derived logic families. A buffered logic family, Source follower Direct Coupled FET Logic (SDCFL), was mixed with an unbuffered logic family (DCFL) to optimise the speed and layout density for most of the chip. Super buffered DCFL (SBDCFL) was used where high drive capability is required including clock lines and long interconnects. Studies of this *mixed* logic approach have shown that good VLSI density, noise immunity and speed can be achieved [BeMa91].

The limits of operation of the logic classes are:

- Power supply: 1.2 to 2.5 V
- Temperature: 0 to 125°C
- 0.5σ *fast–fast* and 0.5σ *slow–slow* models

Unfortunately, accurate temperature models and process spread models were not available when this work was carried out.

The devices available and the corresponding models and device sizes are:

- Enhancement MESFET (EFET): $L = 1.2\mu m, 1.5\mu m$
- Depletion MESFET (DFET): $L = 1.2\mu m, 2.4\mu m, 3.2\mu m$

All gate lengths are specified "as drawn" and are shrunk by $0.4\mu m$ when processed. All device modelling was done using HSPICE [Meta92] and the models supplied by MOSIS for the "edgaas" process. Figure 2.3 shows the simulated I–V curve for a $0.8\mu m$ EFET.

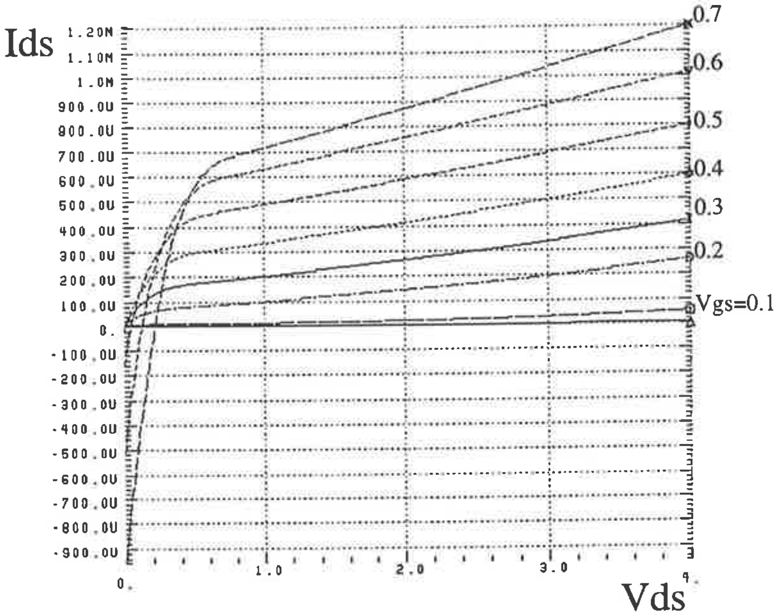


Figure 2.3: Simulated I–V characteristics for an EFET, $L = 0.8\mu m, W = 10\mu m$.

Appendix A contains the performance criteria and specifications for the classes of logic investigated.

2.2.1 Direct Coupled FET Logic

DCFL is the simplest logic class for digital GaAs design and has the smallest power–delay product of the current GaAs normally–off logic classes. It is comparable to nMOS in Silicon VLSI design. An EFET operates as a voltage controlled resistor which pulls the output down as a function of the applied gate voltage, while a DFET operating in the saturation region provides the active pull up as shown in Figure 2.4. (A resistor may replace the depletion mode MESFET in some cases.) When a DCFL gate drives another DCFL gate, the high level output of the first gate is clamped to about $0.7\,V$ by the Schottky diode at the input of the second gate. This limits the voltage swing of the gate and hence the noise margin. By varying the pull up and pull down MESFET widths', the gate can be tuned for speed, noise margin, power and load drive. The pull down to pull up MESFET *ratio* determines the noise margin, propagation delay and transition times.

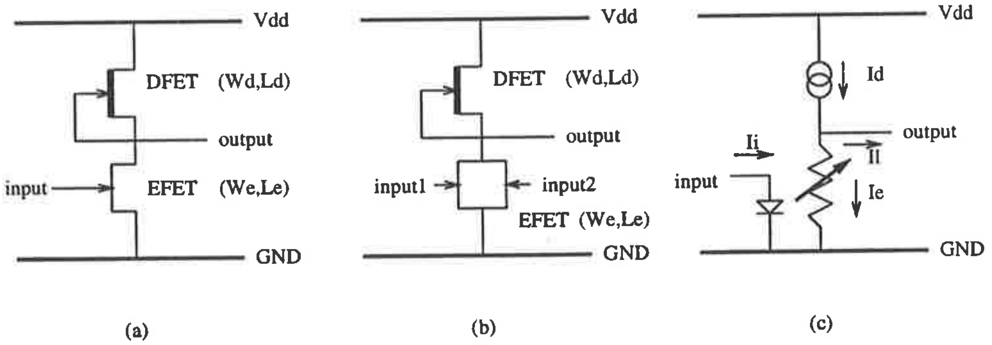


Figure 2.4: (a) DCFL inverter, (b) 2 input NOR gate, (c) equivalent circuit.

In high speed circuits with small voltage swings, the noise margin is the most critical parameter to consider when designing for correct circuit operation. The noise margin may be increased by reducing the on resistance of the EFET. This is achieved by increasing the EFET width with respect to the DFET load at the expense of decreasing the speed of the gate. Figure 2.5 shows the drain current of a DFET when used as a pull up device ($V_{gs} = 0$) for different device sizes. A relatively small (40%) change in current drawn from the supply in the high and low logic states also contributes to circuit stability. In this respect, DCFL produces a quieter power bus since the DFET operates in saturation as a current source. Increasing the power supply voltage pushes the DFET further into saturation and makes the changes in supply current even smaller at the expense of power dissipation. To achieve low power the current in the DFET must be made small, so a gate length of $3.2\mu m$ or $2\mu m$ can be used. For a DFET gate length of $L_d = 3.2\mu m$, a DCFL inverter driving another DCFL load can only drive 1 fan–out. For larger fan–outs or for driving longer wires a DFET gate length of $L_d = 2\mu m$ must be used. A major drawback of DCFL is its poor load drive capability since the DFET is always on and the switching EFET must supply current to both pull the load down and also supply the

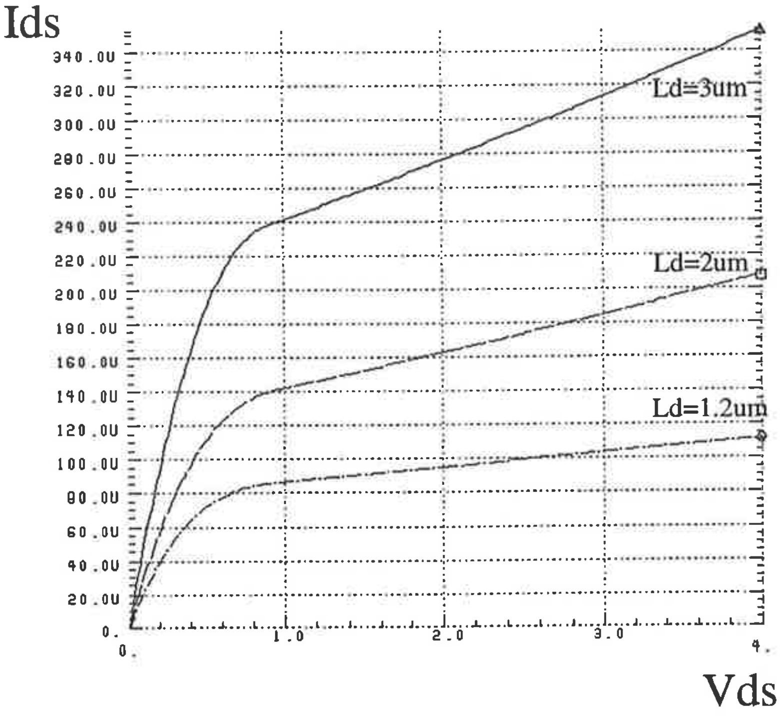


Figure 2.5: Drain current for a DFET with $V_{gs} = 0$ for a 1.2, 2 and 3μ gate length.

load DFET. Complementary logic classes drive the load with just one device while the other is cut off. Figure 2.7 shows that the speed of a DCFL inverter as the load (fan–out) increases is quite linear and fan–outs greater than three or four lead to long gate delays. A buffered logic should be used to drive these higher loads. The wire delay of $70\mu m$ of interconnect is approximately equal to 1 fan–out.

Figure 2.6 shows the average noise margin ($\frac{NM_L+NM_H}{2}$) as a function of EFET width (W_e) with a fan–out of three for a three input NOR gate with only one input signal driven and the other two inputs tied to ground. Definitions of noise margin parameters and measurement techniques are presented in Appendix A. The average noise margin degrades as the fan–out and fan–in increase and as less of the input signals are driven high indicating that this will be the worst case noise margin. The high noise margin in this case is close to zero at around $W_e = 10\mu m$ but the low noise margin is always above 150mV. Since this is the absolute worst case $W_e = 8\mu m$ was chosen. Table 2.1 shows the characteristics of DCFL circuits that were simulated using HSPICE for different device sizes and input conditions.

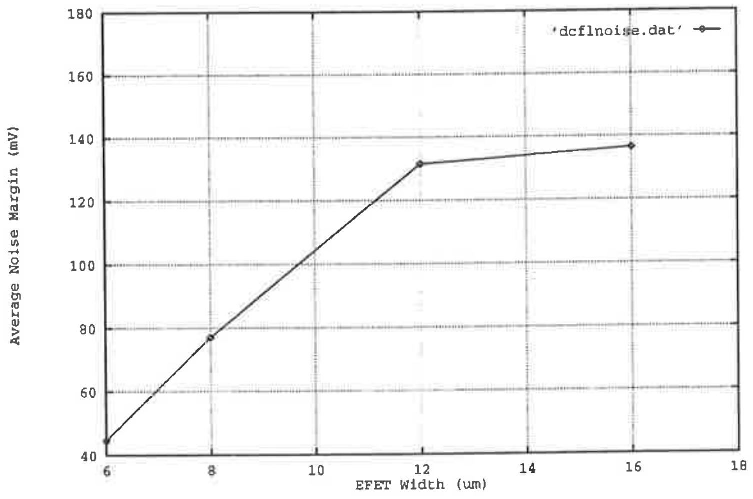


Figure 2.6: Average noise margin of a three input DCFL NOR gate as a function of W_e.

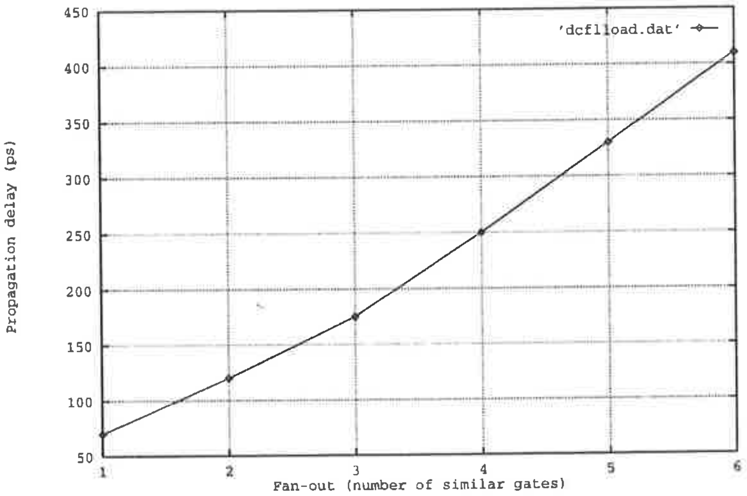


Figure 2.7: Propagation delay of a DCFL inverter as a function of fan–out (capacitive load).

input	Fan	Fan	W_e	L_d	I_{dss}	V_{swing}	NM_H	NM_L	Delay	Power	Area
condition	-in	-out	μm	μm	μA	V	mV	mV	ps	mW	μm^2
-	1	1	6	3.2	70	0.54	70	70	95	0.14	685
-	1	2	6	3.2	70	0.47	63	70	128	0.14	294
-	1	3	6	3.2	70	0.44	60	70	195	0.14	294
1 GND	2	1	6	3.2	70	0.52	73	195	111	0.14	500
tied	2	1	6	3.2	70	0.57	128	159	119	0.14	500
tied	3	3	6	2.4	112	0.39	106	167	153	0.22	685
2 GND	3	3	6	2.4	112	0.47	-27	216	128	0.22	685
tied	3	3	8	2.4	112	0.39	126	151	174	0.22	780
2 GND	3	3	8	2.4	112	0.47	-62	216	169	0.22	780
2 GND	3	3	12	2.4	112	0.46	26	211	250	0.22	880
2 GND	3	3	16	2.4	112	0.45	77	196	305	0.22	970

Table 2.1: Table of simulated DCFL gates with $L_e = 1.2\mu m$, $W_d = 2\mu m$, $Vdd = 2V$ and $T = 70°C$.

The DCFL design guidelines used for an inverter and NOR gates are as follows:

- **DCFL with one fan–out**
  - $W_d = 2\mu m, L_d = 3.2\mu m$
  - $W_e = 6\mu m, L_e = 1.2\mu m$
  - fan–in = 3 maximum
- **DCFL with two or three fan–outs**
  - $W_d = 2\mu m, L_d = 2.4\mu m$
  - $W_e = 8\mu m, L_e = 0.8\mu m$
  - fan–in = 3 maximum

2.2.2 Source Follower Direct Coupled FET Logic

SDCFL is a buffered version of DCFL to improve the load drive capability, voltage swing and noise margin. The buffer is a source follower using an EFET as a pull up and a DFET as a pull down load as shown in Figure 2.8(a). The output of the DCFL stage is clamped at two diode drops. The first diode is across the EFET in the source follower stage and the second is the input diode of the DCFL load. The voltage swing at the input to the SDCFL buffer stage is improved over the DCFL class. There is a V_{gs} voltage drop across the EFET in the source follower, so the logic low level is improved. A negative supply for the source follower may be used to further improve the voltage swing as shown in Figure 2.8(b). A separate negative supply may be used for the ground of the buffer stage to further improve the noise margin and isolate the higher switching currents of the source follower from the DCFL stage. However, the extra power rail requires extra area for power distribution and the small benefit in noise margin performance (10 to $20mV$) did not justify this overhead. There is a trade off in sizing the ratio of the EFET and the DFET in the source follower because a larger DFET can discharge the output node faster but the EFET must be made larger to supply enough current for the DFET as well as charge up the load capacitance. A large load capacitance leads to longer fall times, since the current supplied by the DFET is constant, but the current supplied by the EFET is correspondingly higher for higher load capacitances, since V_{gs} will be larger. The current

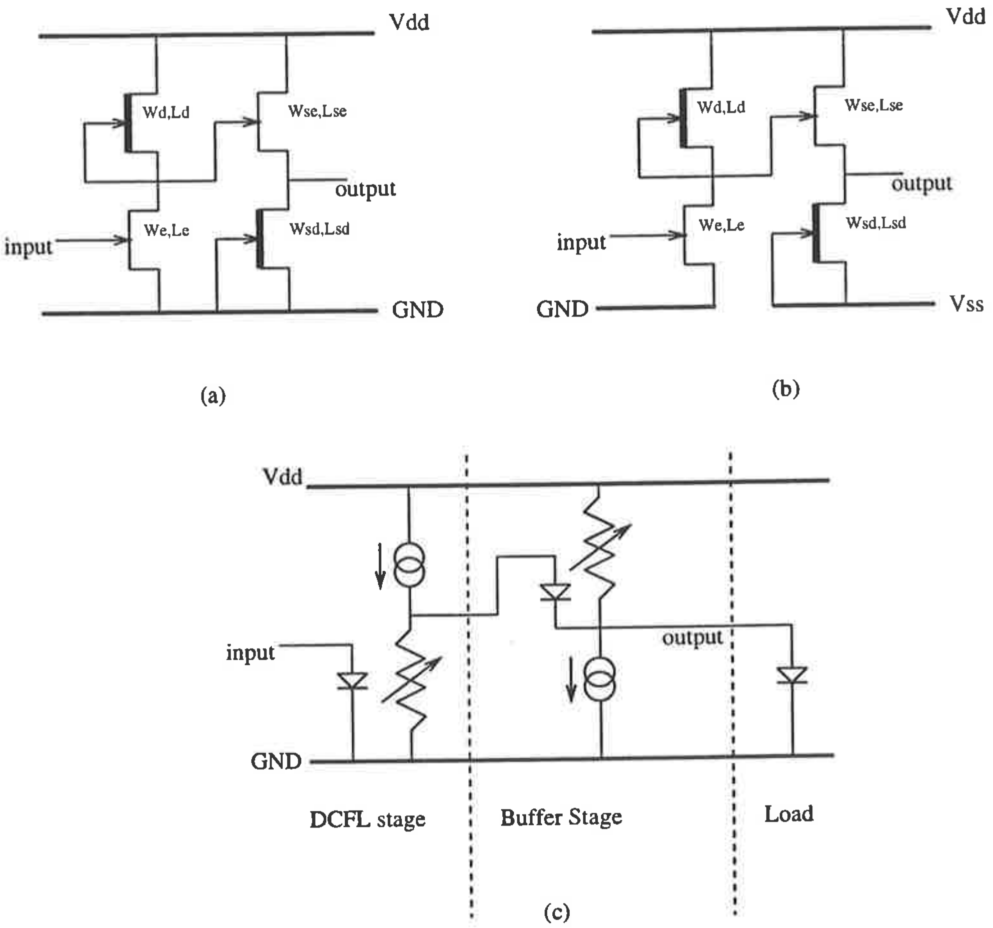


Figure 2.8: (a) SDCFL inverter, (b) SDCFL inverter with extra supply, (c) equivalent circuit. .

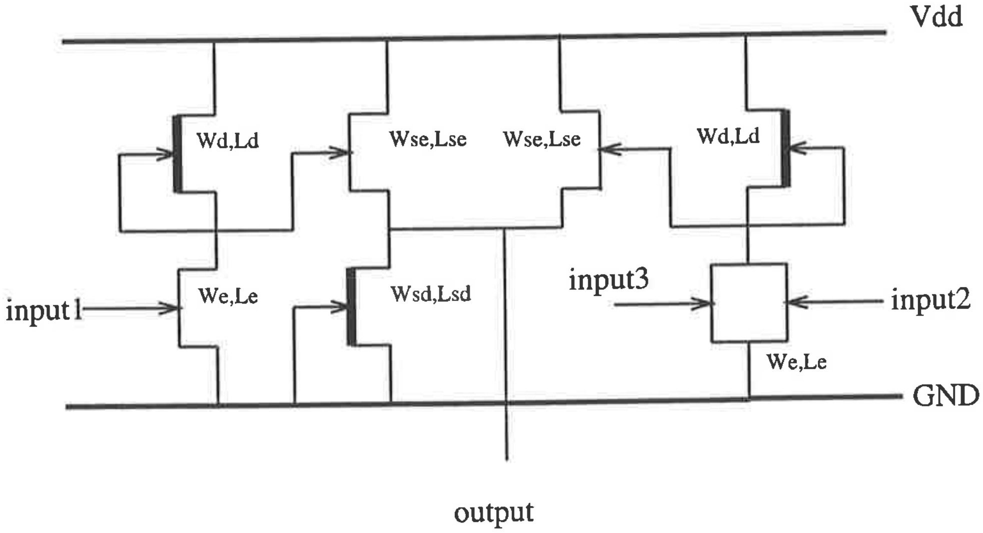


Figure 2.9: OR–AND–INVERT (OAI) logic structure.

W_e μm	W_d μm	L_d μm	W_se μm	I_{dss} μA	V_{swing} V	NM_H mV	NM_L mV	Delay ps	Power mW	Area μm^2
6	2	2.4	10	283	0.59	22	322	202	566	1030
4	2	2.4	10	290	0.57	-42	360	189	580	998
8	2	2.4	10	310	0.53	69	291	218	620	1062
6	2	2.4	12	320	0.54	31	318	197	640	1062
10	2	2.4	10	309	0.50	87	269	240	618	1094
6	2	3.2	10	230	0.54	107	260	269	460	1030
6	3	3.2	10	280	0.545	50	311	242	560	1050
6	4	3.2	10	346	0.544	1.2	346	211	692	1062

Table 2.2: Table of simulated 3–input SDCFL NOR gates with 2 inputs tied to GND and a fan–out of 5, $L_e = L_se = L_sd = 1.2\mu m$, $W_sd = 2\mu m$, $Vdd = 2V$ and $T = 70°C$.

through the source follower depends on the logic state and therefore switching transients are produced in the power rails which lead to ground bounce and noise injection into other circuits. This is caused by the DFET coming in and out of saturation when the circuit switches. The current in the output logic low state is around $150\mu A$ and $700\mu A$ in the output logic high state. The change in current is significant (460%) compared to DCFL (150%) because the buffer switches off in the logic low state. However, SDCFL has a higher noise margin than DCFL and can have a fan–in of up to five.

OR–AND–INVERT (OAI) structures can be made using SDCFL, as shown in Figure 2.9, which are both compact and fast and implement the following logic function (with three fan–in per gate):

$$\begin{aligned} Z &= \overline{A+B+C} + \overline{D+E+F} \\ &= \overline{A}.\overline{B}.\overline{C} + \overline{D}.\overline{E}.\overline{F} \\ &= \overline{(A+B+C).(D+E+F)} \end{aligned}$$

where A, B, C, D, E and F are inputs and Z is the output. Table 2.2 shows the characteristics of SDCFL circuits that were simulated using HSPICE for different device sizes.

The design guidelines used for SDCFL circuits were:

- **SDCFL with up to five fan–outs**
  - $W_d = 2\mu m, L_d = 2.4\mu m$
  - $W_e = 8\mu m, L_e = 1.2\mu m$
  - $W_s d = 2\mu m, L_s d = 1.2\mu m$
  - $W_s e = 10\mu m, L_s e = 1.2\mu m$
  - fan–in = 3 maximum

2.2.3 Super Buffer FET Logic

SBFL improves the load capacitance drive capability of DCFL and SDCFL utilising a push–pull super buffer as shown in Figure 2.10. The disadvantage of using this gate is the noise produced on the power rails because of a conduction path from *Vdd* to ground when the gate changes state. With limited use and careful power rail design, it can be successfully used to drive high loads such as clock lines, buses and high fan–out loads. It has a higher noise margin ($192mV$) than DCFL and SDCFL but a higher power dissipation, and the current changes in the high and low state by 200%. The use of this gate is restricted to an inverter driver only since logic gates are more complex and require more area than DCFL or SDCFL.

Optimising the gate ratios is similar to SDCFL except the devices in the output stage have the same sizes. The design guideline used for SBFL is as follows:

- **SBFL with up to seven fan–outs**
  - $W_d = 2\mu m, L_d = 2.4\mu m$
  - $W_e = 8\mu m, L_e = 1.2\mu m$
  - $W_s e = 12\mu m, L_s e = 1.2\mu m$
  - $W_s e1 = 12\mu m, L_s e1 = 1.2\mu m$

For driving high loads such as clock lines the ratio of these sizes is adhered to. Careful modelling is required to make the transitions fast for clock edges.

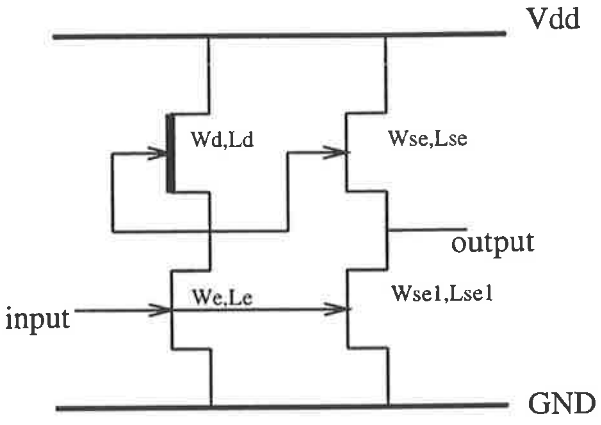


Figure 2.10: SBFL inverter.

2.2.4 Performance Comparison

Table 2.3 is a summary of the characteristics of the DCFL, SDCFL and SBFL inverters discussed.

Parameter	DCFL Inverter	SDCFL Inverter	SBFL Inverter
Delay (ps)	70	120	120
Noise Margin (mV average)	107	142	192
Power (mW average)	0.170	0.550	0.600
Max. fan–in	3	3	3
Max. fan–out	3	5	7
Delay/fan–in (ps)	21	20	-
Delay/fan–out (ps)	75	22	11
Voltage swing (mV)	580	650	
Logic Threshold (mV)	311	331	331
% change in current between logic states	150	460	100
Number of devices for an N–input gate	N+1	N+3	2N+2
Power–delay product (fJ)	10	35	40

Table 2.3: GaAs logic circuit characteristics for DCFL, SDCFL and SBFL.

2.3 Design Methodology

The characteristics of digital GaAs technology which require special attention so as to avoid performance penalties include:

- Lower integration levels than possible with silicon. This is because of the larger device width which is a consequence of using ratioed logic. Complex logic gates aren't available (unlike CMOS).
- Lower yields than equivalent silicon systems.
- Logic families suited to VLSI are characterised by supply voltages approaching the thresholds of the transistors and also suffer from poor noise margins.
- The rise times of 'long' interconnects are degraded by transmission line reflection coefficients at impedance discontinuities.

Reducing device size leads to lower parasitics and high integrationer levels, but designers only have control over how the circuits are realised in layout.

2.3.1 Layout Style

Ring notation is used to layout the logic gates and has been previously reported [BeMa91, SaCa92, Eshr91a]. This technique made the rapid layout of the regular high performance GaAs circuits required for the chip possible. The circuit design should take the following practices into consideration:

- the minimisation of coupling and clock feedthrough by using separate clock and signal lines
- the minimisation of ground bounce by making the circuits quiet when switching current to ground
- close placement of the devices to achieve good device matching
- separation of larger devices to minimise sidegating
- placement of all gates in one direction (horizontal) to gain maximum mobility through the GaAs crystal lattice

- minimisation of interconnect lengths and coupling

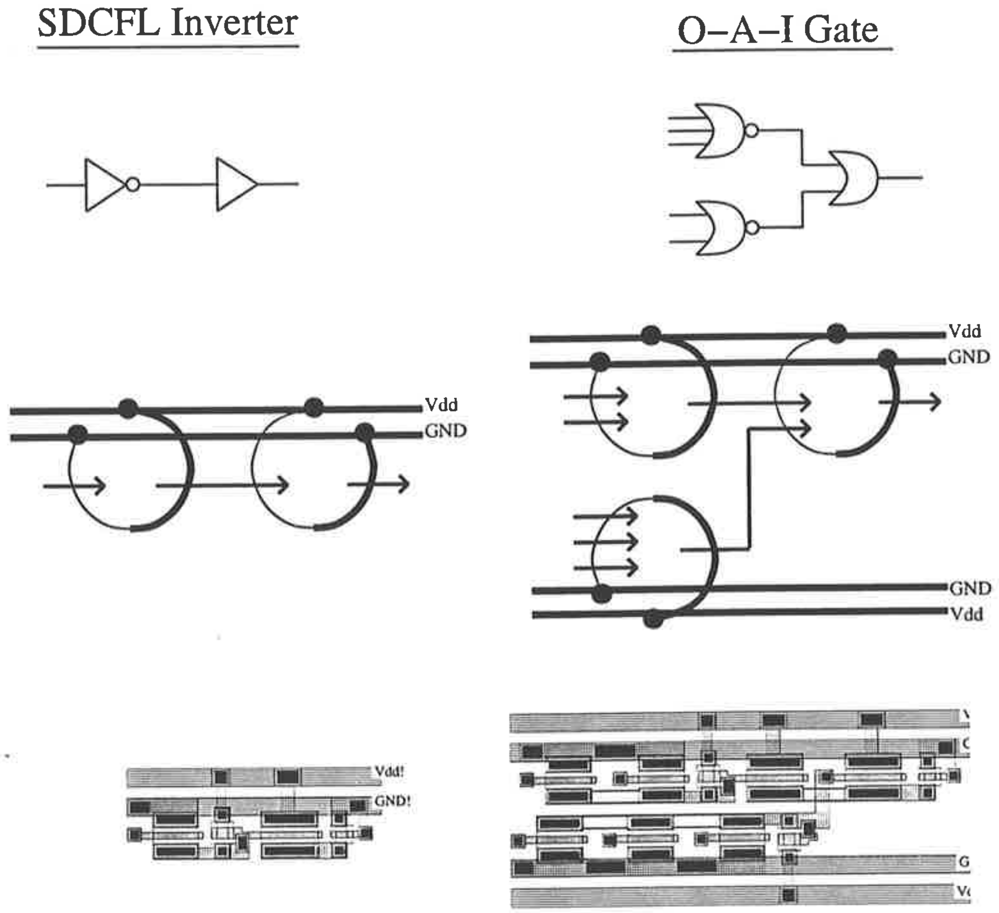
- reduction of inductance and increase capacitance of power buses

- high packing density

Traditional CMOS layout technique involves placing logic in between the ground and *Vdd* power buses. Ring notation places the power buses bunched next to, or on top of, each other. The connections to the transistors and other devices are in the form of rings from the *Vdd* to the ground as shown in Figure 2.11. Enhancement mode MESFETs are drawn as a dashed line while depletion mode MESFETs are drawn as solid lines. Gate connections are drawn as an arrow head crossing the line and interconnections between gates are simply drawn as lines. This provides a simpler method than stick style for nMOS for GaAs designers to layout subcells. By placing the power rails close together the capacitance between them is increased and separating the signals into a wiring channel between the gates and power rails leads to a quieter power bus. Figure 2.12 shows the physical layers used for layout of the GaAs circuits and the key for *ring notation* used in Figure 2.11. Some of the basic building blocks of the layout are a DCFL 2 input NOR gate Figure 2.13, a DCFL 3 input NOR gate Figure 2.14, a SDCFL source follower buffer Figure 2.15 and a SDCFL source follower OR gate Figure 2.16.

2.3.2 Design Tools

CAD for integrated circuits requires a layout editor with some automatic design rule checking, a layout network extractor and circuit simulators. Mapping to various formats such as CIF and CALMA are also required. The layout tool MAGIC [Magi90] was used to design the PE chip. A GaAs network extractor called 'gaasnet' [Beau91] was developed from the ISD phase–1 design suite but its use was discontinued as MAGIC had developed beyond gaasnets parasitic extraction capabilities. A program called 'ext2hsp' [Beau92] was written to generate spice decks suitable for input directly into the HSPICE [Meta92] circuit simulator. This checks transistor models against MESFET types and gate lengths to ensure correct selection of the correct device simulation model. Another script called 'ext2sp' does label name substitution in the SPICE deck. The MAGIC



SDCFL Inverter

O–A–I Gate

Figure 2.11: Schematic, *ring notation* and layout for a SDCFL inverter and OAI structure.

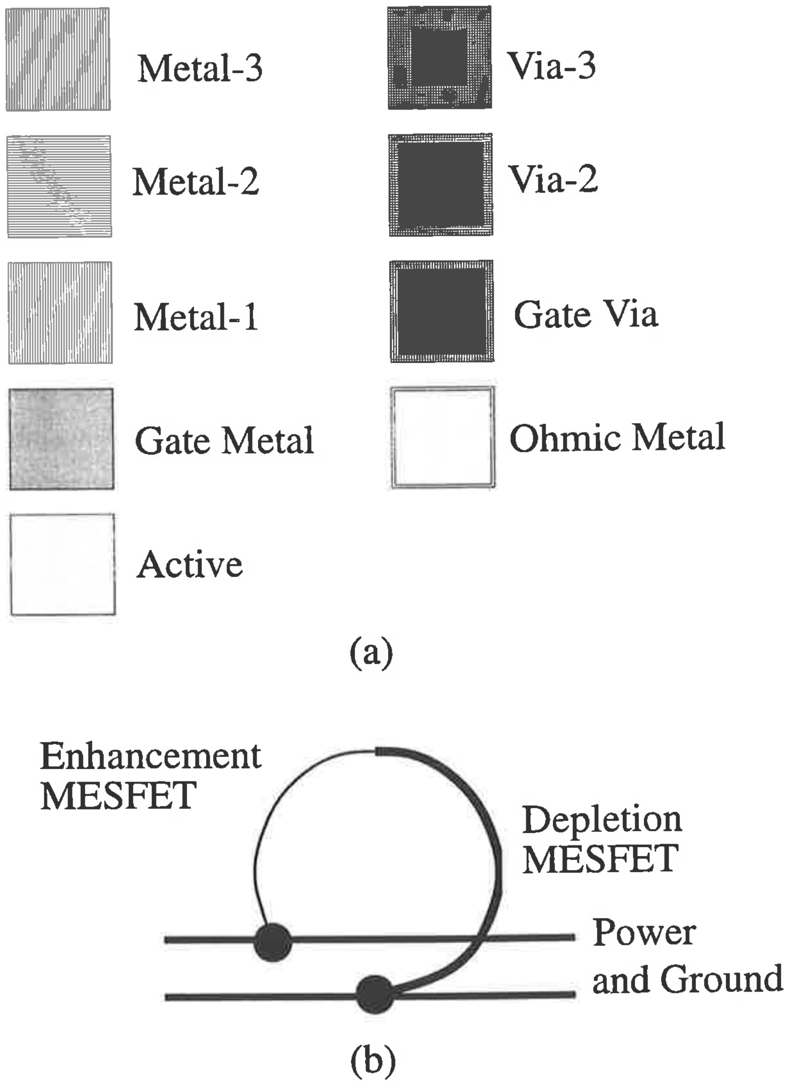


Figure 2.12: (a) Physical layers used for layout and (b) key for *ring notation.*

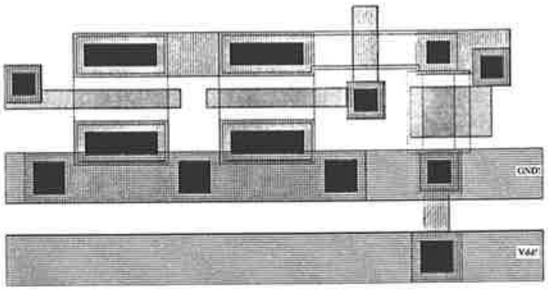


Figure 2.13: DCFL 2 input NOR gate layout.

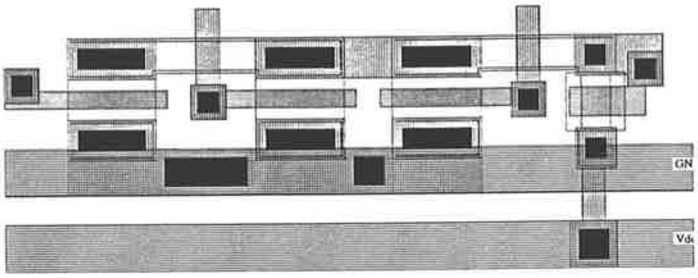


Figure 2.14: DCFL 3 input NOR gate layout.

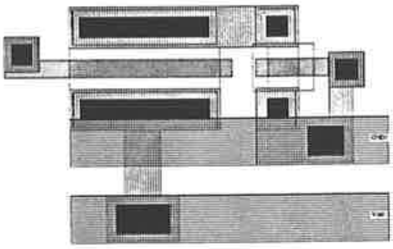


Figure 2.15: SDCFL buffer layout.

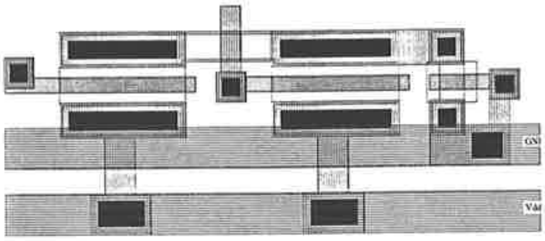


Figure 2.16: SDCFL 2 input OR gate buffer layout.

technology file 'edgaas.tech' was supplied by MOSIS[1] and modified to both correct design rule information and enhance circuit extraction. The circuit simulator IRSIM was used for functional simulation since it uses a simpler transistor model than SPICE for fast turn–around. A HGAAS–II parameter file was written for IRSIM based on timing parameters from HSPICE.

2.4 Circuit Modelling and Parasitic Extraction

2.4.1 Interconnect Analysis

Circuit parasitics play a major role in determining the ultimate performance of an integrated circuit. As the operating frequencies increase, particularly with high speed technologies such as GaAs MESFET and HEMT, the nature of the parasitics change from being mainly capacitive (e.g. CMOS) to a combination of inductive, resistive and capacitive. The relative magnitudes of these parasitic elements on a GaAs chip were investigated. As the wavelengths of the signals become comparable to the length of the interconnect the transmission line (TL) effects become important, as reflections from discontinuities may degrade circuit performance. Transmission line effects become significant at the chip level when signal rise times go below 150ps [Bako90].

The electric field lines of an interconnection may terminate at the adjacent interconnection lines because integrated circuits are inherently densely packed and this particularly occurs in higher metal layers of a multi-level interconnection. The adjacent lines are not uniform in structure and hence there is not such a stable capacitance to ground. This means that the crosstalk between conductors must be carefully modelled for high speed circuits. A software package, 'Raphael' [TeMo93], was used to model and analyse the interconnect structures in the following sections.

[1] MOS Information Service at the University of Southern California.

2.4.2 Capacitance

Bulk GaAs is a semi-insulating material, so the capacitance to the substrate is lower than for CMOS technology. The capacitance to substrate of a single wire on silicon is about $0.08fF/\mu m$ [Rocc90] whereas it is around $0.05fF/\mu m$ for wires on GaAs substrates. Figure 2.17 shows the inter–nodal capacitance for two neighbouring wires as a function of metal pitch [Beau93]. This shows that the dominant capacitance load is the nearest neighbour in a planar metallisation process. The total capacitance per micron length of the centre interconnect (Figure 2.18) is the sum of the capacitance to substrate and the inter–nodal capacitances to neighbouring structures. This shows that the substrate thickness has a negligible effect on the capacitance per micron length of the wire but the effect of the dielectric constant is more significant (25% increase). To study the coupling capacitances, five conductors on a substrate were simulated (Figure 2.19). The coupling capacitance as a function of the design rule are plotted in Figure 2.20 where C_{xy} denotes the capacitive coupling from electrode 'x' to electrode 'y'. The coupling to the closest neighbour accounts for 80 to 90% of the total capacitance for tightly spaced lines while other lines do not contribute significantly. It is interesting to note that there is no mini-

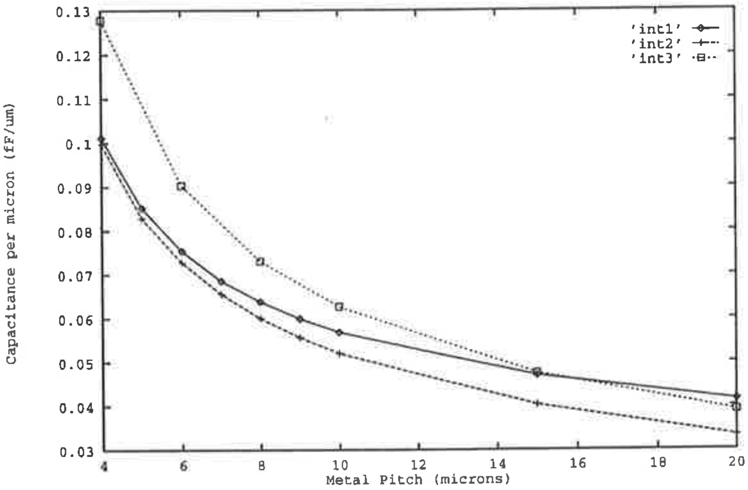


Figure 2.17: Inter–nodal capacitance of two neighbouring wires on $100\mu m$ [$\epsilon_r = 4(+)$], $450\mu m$ [$\epsilon_r = 4(\diamond)$] and $100\mu m$ [$\epsilon_r = 8(\sqcap)$] thick GaAs SI substrate with a backplane and dielectric constant for the inter–level dielectric.

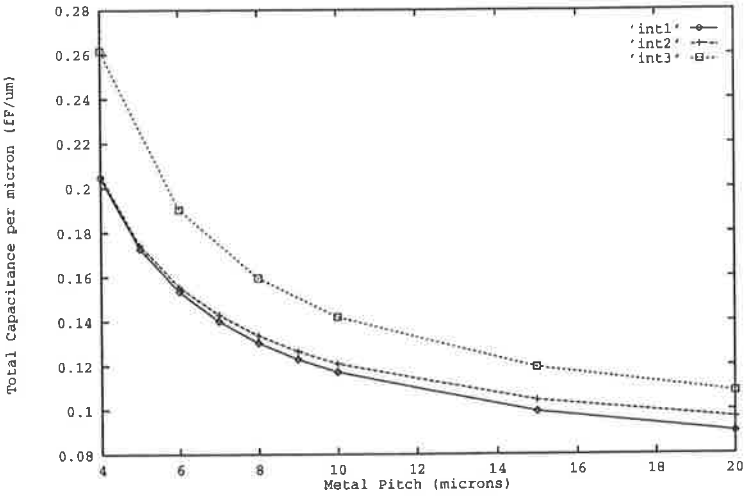


Figure 2.18: Total capacitance of the centre $2\mu m$ wide wire on $100\mu m$ [$\epsilon_r = 4(+)$], $450\mu m$ [$\epsilon_r = 4(\diamond)$] and $100\mu m$ [$\epsilon_r = 8(\sqcap)$] thick GaAs SI substrate with a backplane and dielectric constant for the inter–level dielectric.

mum total capacitance for these structures, as in silicon technology, as the design rule is changed [Rocc90]. To minimise the total capacitance the interconnects must be spaced as far apart as is practical or allowed.

The total capacitance of the wires was plotted as a function of the design rule in Figure 2.21. As expected the capacitance increases with the design rule. The capacitance converges as proximity becomes less important and the major influence becomes the capacitance to substrate. The capacitance of wire 1 equals wire 5 and wire 2 equals wire 4 because of symmetry.

2.4.3 Characteristic Impedance

A coplanar waveguide TL can be used to analyse the characteristic impedance as shown in Figure 2.22. The line has semi-infinite ground planes placed either side (it is assumed that adjacent strips of an interconnect will be ground or a reference).

The formula for an ideal coplanar waveguide [LoBu89] ignores the effect of a ground plane and tends to overestimate the line capacitance, hence underestimating the characteristic

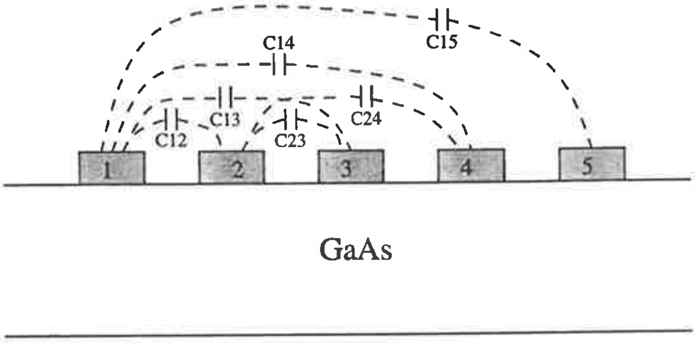


Figure 2.19: Five equally spaced conductors on a GaAs substrate.

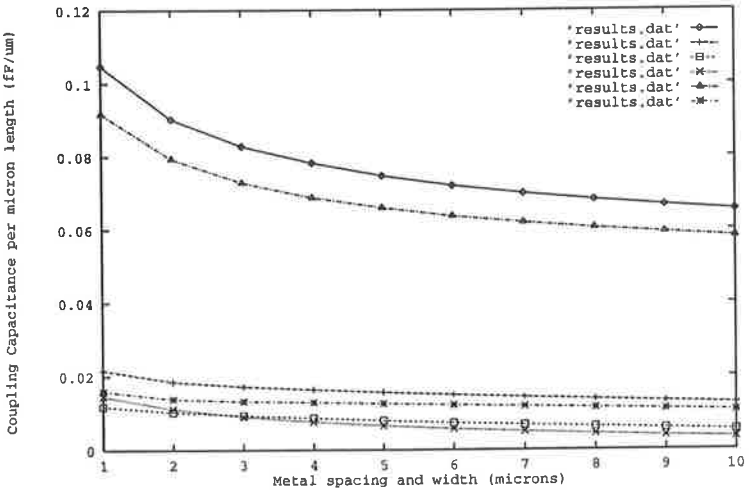


Figure 2.20: Coupling capacitance between electrodes for 5 equal width and spacing electrodes on $100\mu m$ thick GaAs SI substrate embedded in dielectric ($\epsilon_r = 4$), $2\mu m$ thick with a backplane metallisation ($\diamond = C12$, $+ = C13$, $\sqcap = C14$, $\times = C15$, $\triangle = C23$, $* = C24$).

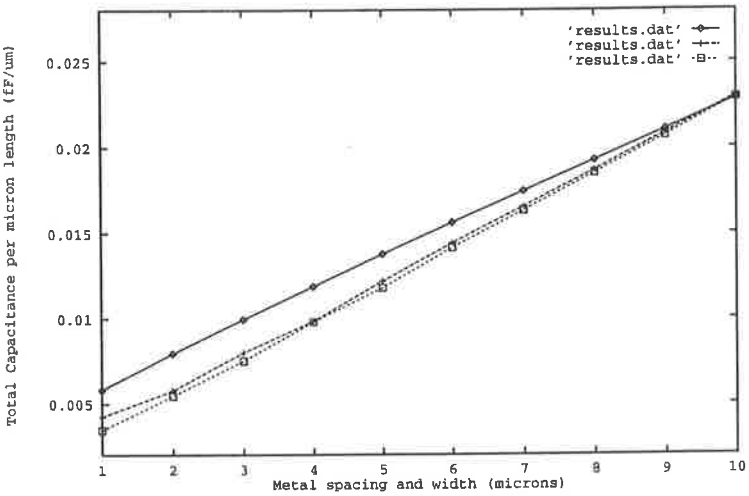


Figure 2.21: Total coupling capacitance for electrodes 1 (◇), 2 (+) and 3 (⊓) for 5 equal width and spacing electrodes on $100\mu m$ thick GaAs SI substrate embedded in dielectric ($\epsilon_r = 4$), $2\mu m$ thick with a backplane metallisation.

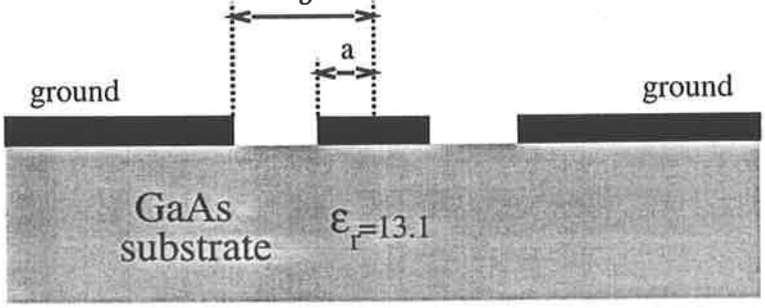


Figure 2.22: Coplanar waveguide (cross section).

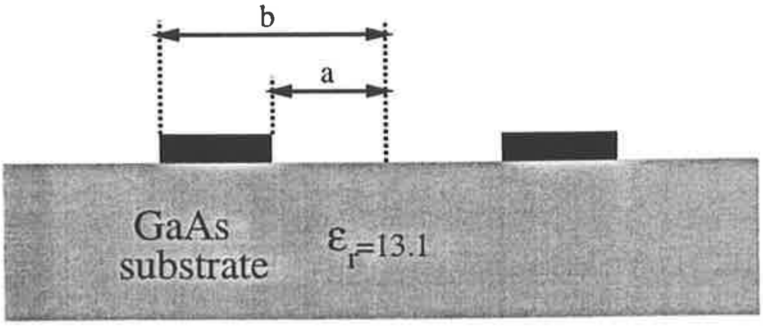


Figure 2.23: Coplanar strips (cross section).

impedance. Thus:

$$Z_0 = \frac{30\pi}{\sqrt{\epsilon_{eff}}} \frac{K(k')}{K(k)} \tag{2.1}$$

where

$$\epsilon_{eff} = 1 + \frac{\epsilon_r - 1}{2} = 7.05(GaAs), \quad k = \frac{a}{b}, \quad k' = \sqrt{1-k^2}$$

and the $K(k)$ is the elliptic integral of the first kind and $K'(k)$ is the complementary function. Tables and formulae for this function can be found in [LoBu89]. The validity of this equation assumes that the substrate thickness is much greater than the line spacing b, which is greater than the line width, a. Consider lines of equal width and spacing, this implies $k = \frac{1}{3}$ and the elliptic function becomes $\frac{K(k)}{K(k')} = 0.64$. Substituting into equation 2.1 gives the characteristic impedance of the line, $Z_0 = 22.7\Omega$. A correction may be applied to account for the increase in effective dielectric constant due to a thin dielectric coating, e.g. polyimide. The effective dielectric constant is multiplied by B, which is given by:

$$B = 1 + \frac{\epsilon_L - 1}{\epsilon_r + 1}\left[1 - exp(\frac{-4.6t}{a+b})\right] \tag{2.2}$$

If $a = 1\mu m$ and $b = 3\mu m$ and the thickness t of polyimide is $2\mu m$ and the effective dielectric constant is $\epsilon_L = 4.0$, then by equation 2.2, $B = 1.19$ and ϵ_{eff} can be corrected. Substituting into equation 2.1 produces the characteristic impedance, $Z_0 = 20.8\Omega$. This value seems quite low compared to the results of the two-dimensional simulation of three parallel interconnects using Raphael, shown in Figure 2.24.

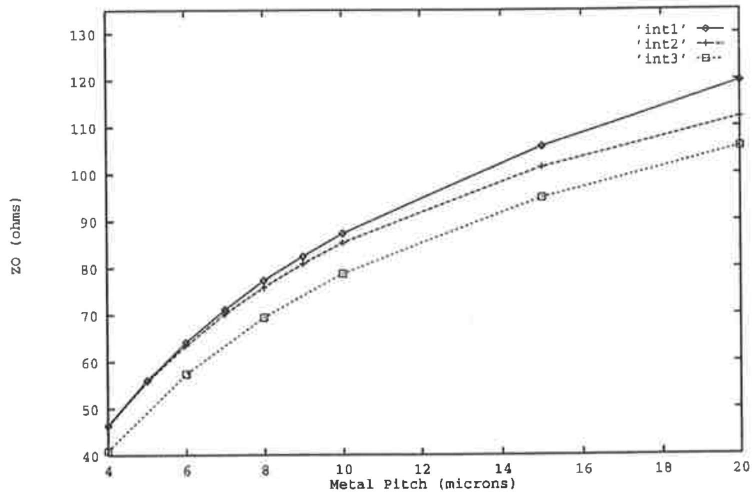


Figure 2.24: Characteristic impedance of the centre $2\mu m$ wide wire on $100\mu m$ [$\epsilon_r = 4(\diamond)$], $450\mu m$ [$\epsilon_r = 4(+)$] and $100\mu m$ [$\epsilon_r = 8(\sqcap)$] GaAs SI substrate with a backplane and dielectric constant for the inter–level dielectric.

A more accurate approach may be to consider two coplanar strips that are at some distance from other lines as shown in Figure 2.23. The characteristic impedance may be given by:

$$Z_0 = \frac{120\pi}{\sqrt{\epsilon_{eff}}} \frac{K(k)}{K(k')} \tag{2.3}$$

where the parameters are defined as being the same as the coplanar waveguide case above. For a line width and spacing of $2\mu m$, the characteristic impedance given by equation 2.3 becomes $Z_0 = 83.2\Omega$. Other methods of characterisation, such as microstrip analysis, are not appropriate since the distance to the backplane is around 0.1 to $0.5mm$ (if a backplane exists). This gives an w/h ratio of several hundred for VLSI type interconnections where w is the width of the interconnect and h is the height of the strip above the ground plane. The characteristic impedance of a microstrip line is given by:

$$Z_0 = \frac{60}{\sqrt{\epsilon_{eff}}} ln(\frac{8h}{w} + \frac{w}{4h})$$

which is valid for $\frac{w}{h} < 10$ [LoBu89], and ϵ_{eff} is given by:

$$\epsilon_{eff} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2}(1 + \frac{10h}{w})^{-\frac{1}{2}}$$

The width of the interconnect would have to be $72\mu m$ for a $100\mu m$ thick substrate with a grounded backplane and a 50Ω characteristic impedance line. For $h = 0.1mm$ and $\epsilon_r = 13.1$ the width of the line is $10\mu m$ which is too wide and not suitable for densely packed integrated circuits. The impedance is not well controlled to ground since the coupling to the nearest interconnect will dominate. A coplanar waveguide structure may be built if a ground plane is placed either side of the interconnect. This would be suitable for making controlled impedance lines connected to pads for wafer probing.

2.4.4 Resistance

The series resistance of an interconnect increases as the feature size is scaled down. This may lead to RC delays in signal lines and ohmic drops in power lines. The resistance in an aluminium line is given by:

$$R = \rho\frac{l}{wt}$$

where $\rho = 2.74 \times 10^{-6}\Omega.cm$ is the bulk resistivity of aluminium, l is the length, w is the width and t is the thickness of the line. Wire resistance is a function of $\frac{1}{w}$. Figure 2.25 shows the resistance per micron of a wire versus the design rule (width and spacing) modelled using Raphael with the equally spaced five parallel trace model from the Raphael Interconnect Library [TeMo93].

The skin effect is the exponential decay of the electric field as it penetrates the conductor at high frequencies and increases the resistive loss. The skin depth is given by:

$$\delta = \frac{1}{\sqrt{\frac{\omega\mu_0\sigma}{2}}}$$

where ω is the frequency in rad/s, μ_0 is the permeability and σ is the conductivity of the conductor. The bandwidth of a $100ps$ wide gaussian pulse is about $3GHz$, so for aluminium wires, $\sigma = \frac{1}{\rho} = 3.65 \times 10^5(\Omega cm)^{-1}$ and $\delta = 1.5\mu m$. This is much greater than the thickness of the interconnect $(0.5\mu m)$ so the resistance is just the DC resistance.

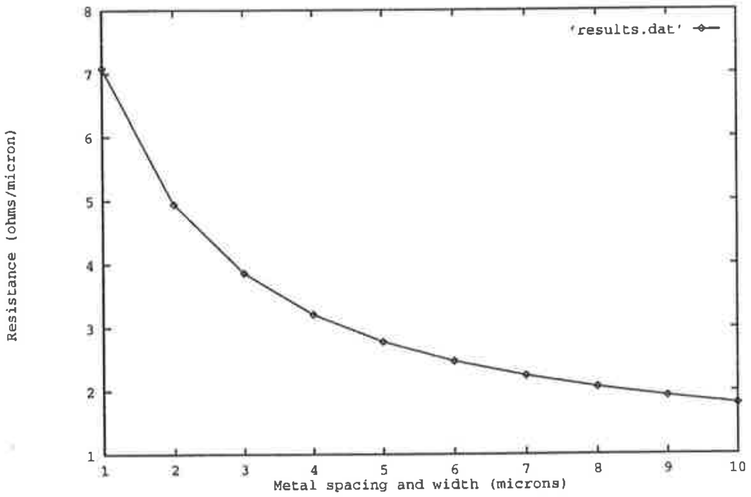


Figure 2.25: Resistance ($10^{-2}\Omega/\mu m$) of a wire as a function of design rule.

2.4.5 Inductance

The partial self and mutual inductances of interconnect structures may be found using some simple formulas derived by Grover and reproduced in reference [TriQ92]. The partial self inductance for a rectangular bar may be approximated by the following equation:

$$L = \frac{\mu}{2\pi} l \left[ln \frac{2l}{w+t} + \frac{1}{2} \right]$$

where w is the width of the line, l is the length, t is the thickness and $\mu = 4\pi\ nH/cm$. For a $2\mu m$ wide line which is $0.5\mu m$ thick and $10\mu m$ long, the total self partial inductance is $5.1pH$ or $0.51pH$ per micron length. For a $100\mu m$ line, the partial self inductance is $97pH$ or $0.97pH/\mu m$. The total inductance per unit length l of two parallel lines when they form part of a complete loop may be approximated by [TriQ92]:

$$\begin{aligned} L_t/l &= (L_{self,1} + L_{self,2} - 2M_{1,2})/l \\ &= \frac{\mu}{\pi} \left[ln(\frac{d}{w+t}) + 1.5 \right] \end{aligned}$$

where it is assumed that the length is much greater than the spacing d between the lines (as is usually the case in a dense integrated circuit). The permeability is μ, w is the width and t is the thickness of the conductor. If $d = w = 2\mu m$ and $t = 0.5\mu m$ the total inductance per unit length is $0.51pH/\mu m$. These results compare well with the results of the two–dimensional simulation shown in Figure 2.26 of coplanar strips on a GaAs substrate. Figure 2.26 shows the self inductance plus the mutual inductance per micron length and Figure 2.27 shows the mutual inductance per micron length only. Note that the difference between Figure 2.26 and Figure 2.27 is the self inductance. Figures 2.28 and 2.29 show the mutual and self plus mutual inductance per micron length of five equal width and spaced interconnects on a $100\mu m$ thick substrate. The same simulation input file was used for the capacitance calculation (from the Raphael Interconnect Library) using the same conditions. The results show an asymptotic decrease of the inductance per micron length with an increasing design rule. These results agree with the results for the three conductor case shown in Figures 2.26 and 2.27.

2.4.6 Line Delay

It has been shown [Bako90] that for various line lengths and driver impedances the interconnect delay for aluminium is constant and minimum past a width of $2\mu m$. The RC delay is proportional to the square of the line length whereas the electromagnetic LC delay is proportional to length. The RC delay along an interconnect of length l is the time to charge the end of the line to 50% of the final value and is given by:

$$t_{RC} = 0.69RCl^2w \tag{2.4}$$

where R is the sheet resistance, w is the width and C is the capacitance per unit length of the conductor. The electromagnetic transit delay is given by:

$$t_{LC} = l\sqrt{LC}$$

where l is the length of the interconnect and $L = 6 \times 10^{-6} H/m$ is the inductance per unit length and $C = 0.2 \times 10^{-9} F/m$ is the capacitance per unit length for $2\mu m$ aluminium wires. The critical line length at which an interconnect must be treated as a TL occurs when the rise time t_r of the signal is the same as the time of flight down the line, t_f. Substituting in the formula [Bako90]:

$$2.5t_f = t_r$$

for a $100ps$ rise time which is routinely observed in GaAs HEMT technology gives $t_f = 40ps$. By using the electromagnetic delay equation (2.4) the critical length of wire is $l = 1.1mm$.

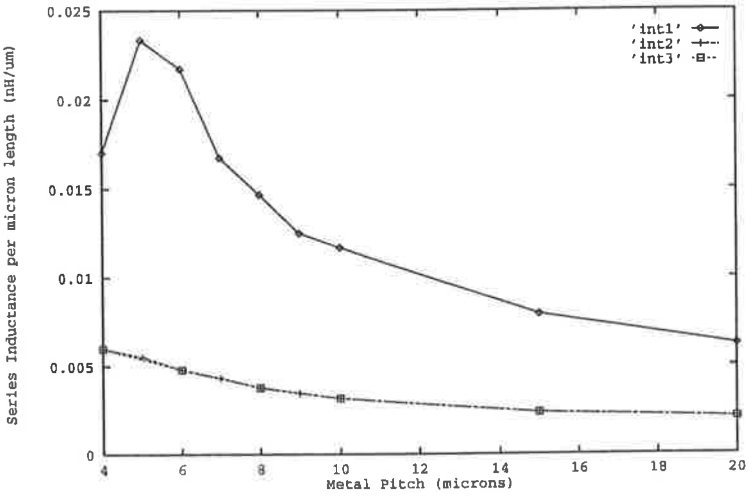


Figure 2.26: Self plus mutual inductance of the centre $2\mu m$ wide wire on $100\mu m$ [$\epsilon_r = 4(+)$], $450\mu m$ [$\epsilon_r = 4(\diamond)$] and $100\mu m$ [$\epsilon_r = 8(\sqcap)$] thick GaAs SI substrate with a backplane and dielectric constant for the inter–level dielectric.

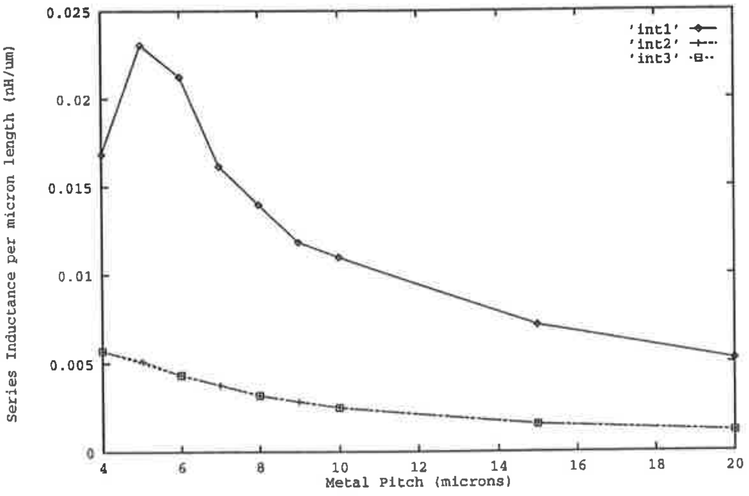


Figure 2.27: Mutual inductance of the centre $2\mu m$ wide wire on $100\mu m$ [$\epsilon_r = 4(+)$], $450\mu m$ [$\epsilon_r = 4(\diamond)$] and $100\mu m$ [$\epsilon_r = 8(\sqcap)$] thick GaAs SI substrate with a backplane and dielectric constant for the inter–level dielectric.

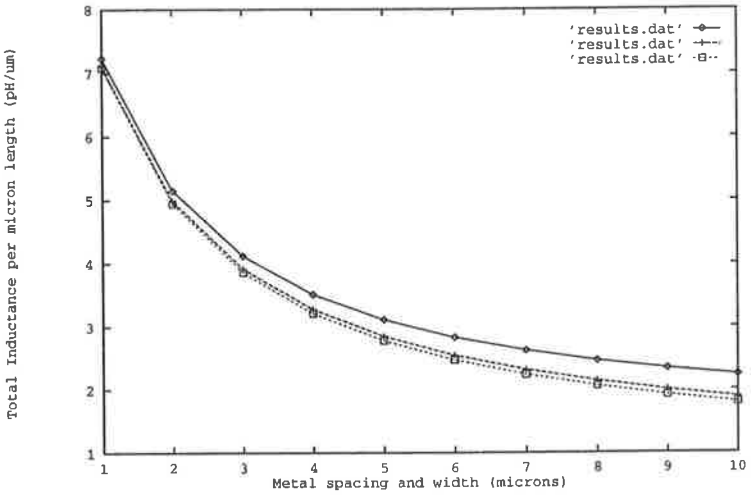


Figure 2.28: Self plus mutual inductance of 5 equal width and spaced interconnects on a $100\mu m$ GaAs SI substrate with a backplane ($\diamond = L11 = L55$, $+ = L22 = L33$, $\sqcap = L33$).

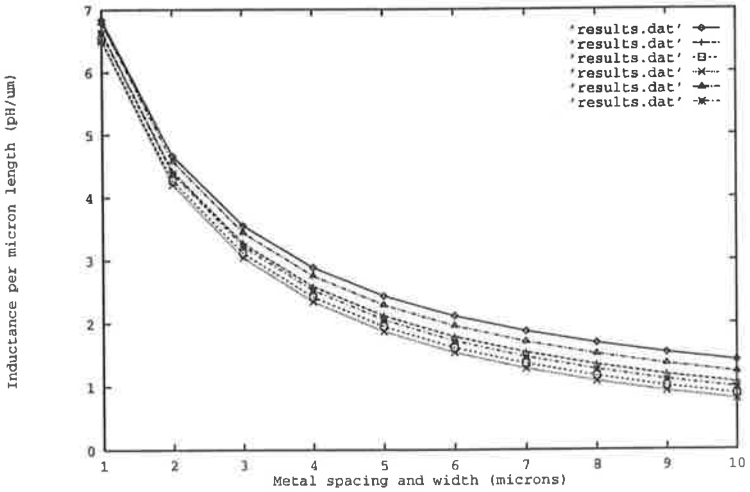


Figure 2.29: Mutual inductance of 5 equal width and spaced interconnects on a $100\mu m$ GaAs SI substrate with a backplane ($\diamond = L12$, $+ = L13$, $\sqcap = L14$, $\times = L15$, $\triangle = L23$, $* = L24$).

2.4.7 Source Impedance of the Driving Gate

The source resistance of the driving circuit relative to the characteristic impedance of the line determines the behaviour of the signal on the line. If the source resistance is low compared to the line impedance, reflections may be observed on the line. However, if the source resistance is high compared to the line impedance, a lumped capacitor approximation may be used since the voltage at the end of the line will rise slowly. The characteristic impedance of packed $2\mu m$ aluminium lines ranges from 47 to 150Ω with the pitch ranging from 4 to $20\mu m$ as determined by two–dimensional simulation using Raphael. The source impedance of a source follower is typically several hundreds of ohms. Short lines ($\leq 600\mu m$, as determined in the resistance calculation) have a total line resistance less than 18Ω for $2\mu m$ wide aluminium wire, therefore $\frac{R_{line}}{Z_0} \simeq 0.5$ and $\frac{R_{gate}}{Z_0} \geq 2$. It has been stated that a lumped capacitor model may be used in this case for a delay accuracy within 10 percent [LoBu89]. If the size of the driver is increased (R_{gate} becomes smaller) or the line is long, a more complex model for the interconnect should be used.

2.4.8 Interconnect Models

An interconnect line may be modelled as; a lumped capacitance, L–shaped RC circuit, a hybrid–π circuit, T model, T2 model, nRLC segments, ideal TL, or a lossy TL [LoBu89, Bako90]. The choice of model depends on the required accuracy and the effects to be modelled. Two models were considered, a lumped capacitance model and a lossy TL which are the simplest and the most advanced models, respectively. For circuit simulation, the simplest model satisfying the required accuracy should be chosen. VLSI interconnect lengths typically fall into two groups, short local gate to gate and long cell to cell connections. The gate to gate connection was considered to be around $100\mu m$ long and the cell to cell connection to be around $1mm$.

Interconnections with a high speed technology were studied using some P-HEMT circuits which are potentially faster than MESFET circuits [Beau93]. The device models were characterised from prototype P-HEMT devices fabricated by the Department of Electronics Engineering at Seoul National University, Korea. A source follower model was chosen to drive the circuit which is used to buffer the output of the SCFL gates. This uses a $10\mu m$ wide P-HEMT, two level shift diodes and a minimum size current source ($300\mu A$) with a $5\,V$ supply. A $1.5\,V$ step input is applied to the input of a buffered source follower to launch a signal into the model and the waveform response at each end of the line is measured.

Lumped Capacitor Model

A lumped capacitor cannot model any ringing effects on long lines but can be reasonably accurate if the waveforms are well behaved. It is the simplest interconnection model for implementation on a circuit simulator. The lumped capacitive load of a $100\mu m$ and a $1mm$ line is $23fF$ and $230fF$, respectively. Figure 2.30 shows the simulation of a step input applied to these lines. The lumped capacitor model is valid for small buffer stages and short lines in which the effects of series resistance and inductance can be ignored.

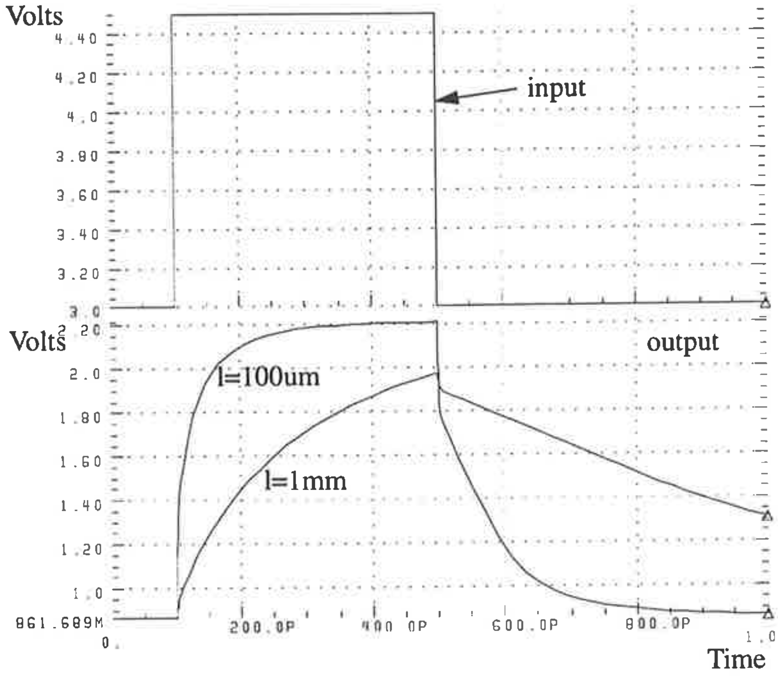


Figure 2.30: Transient analysis of a $100\mu m$ and $1mm$ line modelled as a lumped capacitor.

Lossy Transmission Line Model

The lossy TL model is a complete model of a line with distributed series resistance, inductance and shunt capacitance. The value of shunt conductance is negligible and is ignored. The number of lumped sections in the model can be increased to improve the accuracy but the simulation time is longer. The parameters for the TL model are from the results of the Raphael simulation for three, $2\mu m$ width and spacing interconnects on $100\mu m$ thick SI GaAs substrate embedded in a dielectric with $\epsilon_r = 4$.

- $C = 0.2056 \times 10^{-9} F/m$
- $L = 5.984 \times 10^{-6} H/m$
- $R = 3 \times 10^{-8} \Omega/m$

Figure 2.31 shows the transient analysis of this circuit for a $100\mu m$ and a $1mm$ line with the results for the lumped capacitor model superimposed. Good accuracy was achieved with 20 lumped sections. A $5mm$ line was simulated to observe any TL effects with this model. The results of a transient analysis is shown in Figure 2.32. The ripples in the

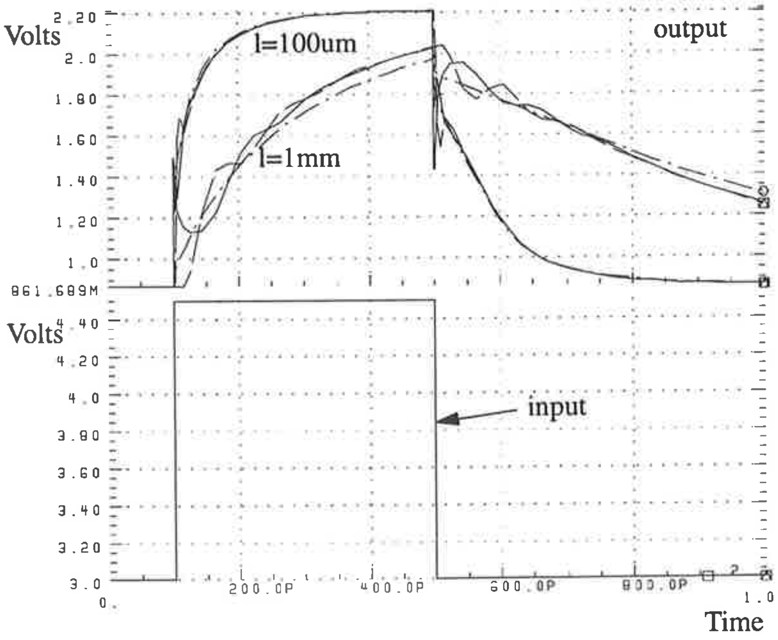


Figure 2.31: Transient analysis of a $100\mu m$ and $1mm$ line modelled as a lossy TL and a lumped capacitor.

waveforms at the driver and receiver, due to reflections, can clearly be seen.

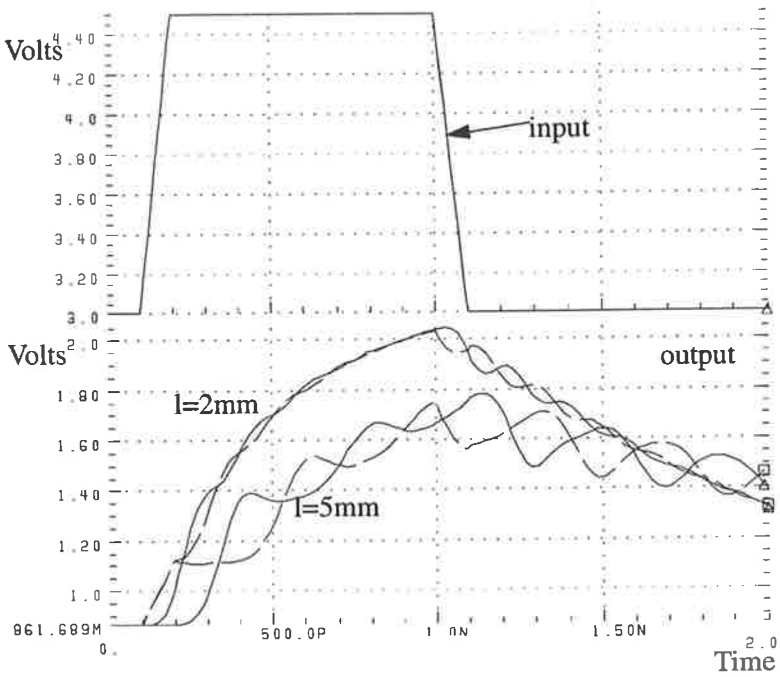


Figure 2.32: Transient analysis of a $2mm$ and a $5mm$ line modelled as a lossy TL (signals plotted at start and end of the TL).

Effect of Resistance

The lossy TL circuit was re–simulated without the distributed resistance. The results of a transient analysis are shown in Figure 2.33. Comparing the lossless TL (Figure 2.33) with the lossy TL (Figure 2.31) shows there is no difference in the results. Interconnects of this type may be considered as lossless without loss of simulation accuracy.

2.4.9 Crosstalk

Crosstalk is a signal transition on a wire which influences the signal on another neighbouring wire. Crosstalk is mainly due to inter–nodal coupling capacitances between lines in VLSI circuits. Crosstalk may be minimised by using a circuit layout style where lines do not run parallel for 'long' distances, adjacent signal lines are spaced far apart and crossovers are avoided. Ground lines either beside or above a signal reduce crosstalk to other signals, although this may cause noise on the ground lines. Placing alternate ground

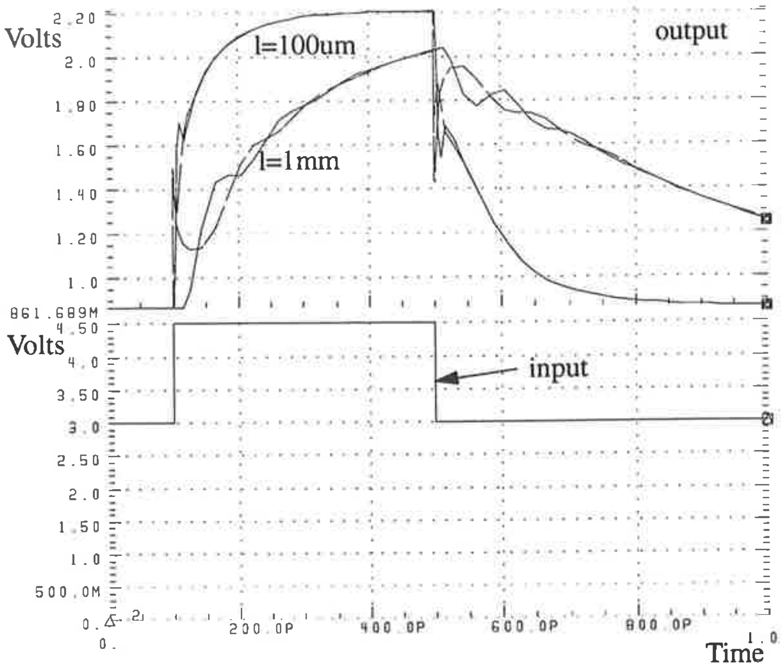


Figure 2.33: Transient analysis of a $100\mu m$ and $1mm$ line modelled as a lossless TL.

lines in a circuit uses more area although this must be used in some instances, particularly at the chip boundary. SCFL gates can have differential input and output signals which means there are twice as many signal lines to consider than in other logic classes such as DCFL. There is a virtual ground between differential signals so placing a power bus between the differential wires spaces them further apart as there is less signal-signal coupling. This also means that the power bus should be cleaner than if there were just a single coupled signal line. The capacitive coupling between lines plotted as a function of spacing can be seen in Figure 2.20.

2.4.10 Package Parasitics

Parasitics associated with the packaging limit the communication bandwidth across a chip boundary. The bond wire, trace and external lead from the package form a TL which may be modelled as a coplanar waveguide or a stripline if the package has a metal floor and lid. A first order equivalent 'T' circuit model for a pad and package bond for a 24 pin leadless chip carrier (LCC) [LoBu89] is shown in Figure 2.34. The effective trace impedance is 100Ω. TriQuint [TriQ92] propose a complex model for their MLC (multi-

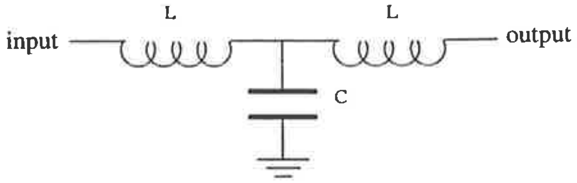


Figure 2.34: Equivalent model for a LCC pad and bond, $L = 1.4nH$, $C = 40fF$.

layer ceramic) packages so the model must be determined by the package structure. The pad to pin characteristic impedance is 50Ω with a pad to pin delay of about $80ps$. First order effects of pad, bond wire and package lead parasitics can be modelled and are probably sufficient to characterise the circuit. The bond wire inductance is about 1 to $2nH$ for LCC type packages [TriQ92] which can be an order of magnitude higher for needle probes and dual-in-line (DIL) type packages.

2.4.11 Pad Structures

A three-dimensional field analysis using Raphael [TeMo93] was performed for a single pad on a $450\mu m$ thick GaAs substrate with a backplane metallisation. This showed the total pad capacitance to be $26.9fF$ for a $100\mu m$ square pad and $34.2fF$ for an $80\mu m$ square pad. The same simulation was performed for three co-linear pads for which the total and inter-pad coupling is shown in Table 2.4. The total coupling referred to ground

Pad size (μm)	Pad pitch (μm)	ϵ_r	Substrate thickness (μm)	C_{pad1} (fF)	C_{pad2} (fF)	$C_{pad1,2}$ (fF)	$C_{pad1,3}$ (fF)
100	150	13.1	450	38.06	41.21	11.3	2.18

Table 2.4: Total and inter-pad capacitance simulation for three adjacent bonding pads on a SI GaAs substrate.

(if the pad either side of the centre pad is grounded) is $41fF$. This agrees with the results published by TriQuint [TriQ92] and is higher than for the case of a single pad ($26.9fF$). The capacitance of a square pad from the following formula which uses an approximate microstrip formula [LoBu89] is given by:

$$C_{pad} = 4\pi\epsilon_0 \frac{\epsilon_{eff} W}{ln(8h/W)} - \epsilon_0 \left(\frac{\epsilon_r W^2}{h}\right)$$

where

$$\epsilon_{eff} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2(1 + 12(h/W))^{0.5}}$$

W is the size of a square pad and h is the height of the pad above the backplane. Table 2.5 shows C_{pad} for various sizes using this formula and the results agree closely with the previous estimate.

W(μm)	h (μm)	ϵ_{eff}	$C_{pad}(fF)$
100	100	8.73	39.0
80	100	8.56	29.4
50	100	8.26	14.8
20	100	7.83	4.45
100	450	7.87	22.9
80	450	7.14	15.8

Table 2.5: Pad capacitance for various pad sizes and substrate thicknesses ($\epsilon_r = 13.1$).

2.4.12 Power Supply and Ground Lines

Current density must be kept below a limit to prevent electromigration and subsequent open circuits. The safe limit for the current density of aluminium interconnect is $2 \times 10^5 A/cm^2$ up to a temperature of $125°C$. This equates to $1mA/\mu m$ width for $0.5\mu m$ thick lines. Inductance in power and ground lines may cause voltage transients due to their self inductance. To calculate the magnitude of the voltage spike the self inductance and change in current per unit time need to be determined. The magnitude of the voltage spike is given by the familiar equation:

$$\Delta V = L \frac{\Delta I}{\Delta t}$$

where ΔI is the change in current in time Δt. This may be used to check the variation in supply voltage on a power bus for a group of logic gates as they switch current from one state to another. Ohmic drops in the power supply need to be avoided. The resistivity of a $0.5\mu m$ thick aluminium interconnect is $0.06\Omega/square$. If the current is at its highest allowable density of $1mA/\mu m$ width, the voltage drop per unit length becomes $60mV/mm$ or 3% of a $2V$ supply. Typically, a 5% supply variation ($100mV$) can be tolerated but this depends on the noise margin of the logic gates.

2.5 Summary

A review of MESFET models has been presented suitable for use in SPICE circuit simulators. The MESFET models used are from Vitesse Semiconductor and are implemented using the Statz–Raytheon model in the HSPICE circuit simulator. GaAs MESFET digital logic classes are also reviewed and DCFL, SDCFL and SBFL are chosen for implementing GaAs digital circuits. These logic classes are optimised for speed, area and noise margin by adjusting the sizes of the MESFETs in the logic gates. A design methodology is developed along with a layout style called '*ring notation*' which is used to design layout primitives including NOR, OR, source followers and OR–AND–INVERT structures. A study of the interconnect parasitics and suitable models for the behaviour of high speed signals on GaAs substrates is given. The study included capacitive coupling, resistance, inductance, characteristic impedance, line delay and source impedance. An evaluation of interconnect models found that short lines ($< 600\mu m$) need to be modelled as a lumped capacitance but long wires may be modelled as non-lossy transmission lines. A SPICE deck can be extracted directly from the layout using a program called 'ext2sp'. The interconnect extraction is limited to inter–nodal and ground lumped capacitance. SPICE circuits for long wires must be created by hand and incorporated into the extracted circuit. Package parasitics and current densities are discussed and a value for bond pad capacitance is calculated.

Chapter 3

Systolic Ring Processing Element

The PE forms the basic computational unit in the mesh connected systolic array. It performs multiplication, addition and multiplication–accumulation of the two input operands. Architectures for a class of digit–serial systolic ring floating point PEs are targeted for fabrication in Gallium Arsenide technology and which are optimised for matrix processing is discussed in this chapter. Digit–serial multiplication and floating point numbers are discussed and a systolic cell is presented with two models for floating point multiplication. The PE is subsequently improved with additional circuits to allow it to perform floating point addition. A performance metric is derived by minimising the total job time for a matrix product using a systolic array. This performance metric is used to optimise the architecture of the PE. The memory bandwidth requirement of the systolic array is also discussed.

3.1 Digit–Serial Multiplication

Previous work on integer digit–serial processing techniques can be found in [HaCo90], [CoHa92] and [Parh89] and in the discussion in Chapter 1. Much of this work is concerned with the partitioning of a parallel operation into a sequence of smaller–radix digit–serial operations. To derive a digit–serial multiplier cell, consider X, an M–digit number $\{x_0, x_1, \ldots, x_{M-1}\}$ represented in base β as:

$$X = \sum_{i=0}^{M-1} x_i \beta^i \tag{3.1}$$

Similarly Y, an N–digit number in base β is represented as:

$$Y = \sum_{i=0}^{N-1} y_i \beta^i \tag{3.2}$$

The product of X and Y is given by:

$$XY = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} x_i y_j \beta^{i+j} \tag{3.3}$$

Let each digit x_i and y_j of X and Y have a r–bit binary representation given by:

$$x_i = \sum_{k=0}^{r-1} x_{ik} 2^k \tag{3.4}$$

and

$$y_j = \sum_{l=0}^{r-1} y_{jl} 2^l \tag{3.5}$$

Substituting equations 3.4 and 3.5 into equation 3.3 gives:

$$XY = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sum_{k=0}^{r-1} \sum_{l=0}^{r-1} x_{ik} y_{jl} 2^{k+l} \beta^{i+j} \tag{3.6}$$

Re–writing the innermost summation gives:

$$XY = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sum_{k=0}^{r-1} \left[\sum_{l=0}^{r-1-k} x_{ik} y_{jl} 2^{k+l} + \beta \sum_{l=r-k}^{r-1} x_{ik} y_{jl} 2^{k+l-r} \right] \beta^{i+j} \tag{3.7}$$

Let

$$A_{ijk} = \sum_{l=0}^{r-1-k} x_{ik} y_{jl} 2^{k+l} \tag{3.8}$$

and

$$B_{ijk} = \sum_{l=r-k}^{r-1} x_{ik} y_{jl} 2^{k+l-r} \tag{3.9}$$

Then equation 3.7 can be re–written as:

$$XY = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sum_{k=0}^{r-1} \left[A_{ijk} \beta^{i+j} + B_{ijk} \beta^{i+j+1} \right] \tag{3.10}$$

The A_{ijk} are associated with partial digit sums of weight β^{i+j} and the B_{ijk} are associated with partial sums of weight β^{i+j+1}.

Expanding the summation over j gives:

$$\begin{aligned} XY &= \sum_{i=0}^{M-1}\sum_{k=0}^{r-1}\Big[A_{i0k}\beta^{i} + B_{i0k}\beta^{i+1} + A_{i1k}\beta^{i+1} + B_{i1k}\beta^{i+2} + \\ &\qquad A_{i2k}\beta^{i+2} + B_{i2k}\beta^{i+3} + \ldots A_{i(N-1)k}\beta^{N-1} + B_{i(N-1)k}\beta^{N}\Big] \\ &= \sum_{i=0}^{M-1}\sum_{k=0}^{r-1}\Big[A_{i0k}\beta^{i} + \sum_{j=1}^{N-2}\left(A_{i(j+1)k} + B_{ijk}\right)\beta^{i+j+1} + B_{i(N-1)k}\beta^{N}\Big] \end{aligned} \qquad (3.11)$$

Expanding the A_{ijk} and B_{ijk} terms in this equation gives:

$$\begin{aligned} XY = \sum_{i=0}^{M-1}\Bigg[&\sum_{k=0}^{r-1}\sum_{l=0}^{r-1-k} x_{ik}y_{0l}2^{k+l}\beta^{i} + \\ &\sum_{j=0}^{N-2}\Bigg[\sum_{k=0}^{r-1}\sum_{l=0}^{r-1-k} x_{ik}y_{(j+1)l}2^{k+l} + \sum_{k=0}^{r-1}\sum_{l=r-k}^{r-1} x_{ik}y_{jl}2^{k+l}\Bigg]\beta^{i+j+1} \\ &+\sum_{k=0}^{r-1}\sum_{l=r-k}^{r-1} x_{ik}y_{(N-1)l}2^{k+l-r}\beta^{N}\Bigg] \end{aligned} \qquad (3.12)$$

Consider a four-bit per digit representation ($r = 4$). The two terms in the inner brackets give the following for (x_iy_j) and (x_iy_{j+1}):

$$\sum_{k=0}^{3}\sum_{l=4-k}^{3} x_{ik}y_{jl}2^{k+l} = \begin{array}{r} x_1y_32^4+ \\ x_2y_32^5+x_2y_22^4+ \\ x_3y_32^6+x_3y_22^5+x_3y_12^4+ \end{array} \qquad (3.13)$$

and

$$\begin{aligned} &\sum_{k=0}^{3}\sum_{l=0}^{3-k} x_{ik}y_{(j+1)l}2^{k+l} = \\ &\qquad x_0y_32^3+x_0y_22^2+x_0y_12^1+x_0y_02^0+ \\ &\qquad x_1y_22^3+x_1y_12^2+x_1y_02^1+ \\ &\qquad x_2y_12^3+x_2y_02^2+ \\ &\qquad x_3y_02^3 \end{aligned} \qquad (3.14)$$

where the terms in equation 3.13 are the high–order components of the digit product (x_iy_j) and the terms in equation 3.14 are the low–order components of the product (x_iy_{j+1}). The i, j and $j+1$ subscripts on the right hand side of equations 3.13 and 3.14

have been omitted for clarity of presentation.

This re-formulation of the XY product shows that partial products of weight β^{i+j+1} are formed by summing the digit product $\left(x_i y_{(j+1)}\right)$ with the digit product $(x_i y_j)$. To form digits of weight β^{i+j+1} a structure is required which in each time period can compute and accumulate the two different partial products from adjacent time periods, and then accumulate the result with partial products computed in other cells. The function required is:

$$Z = XY + PP + V \tag{3.15}$$

where PP is the r–bit partial product input and V is the r–bit carry. X and Y are r–bit input operands. The term XY implies the pipelined computation and accumulation of the high– and low–order digits as discussed. To implement this function a pipelined parallel multiplier structure is used. Pipelining of the high–order output of this multiplier with one level of registers delays the high–order digit by one clock cycle. This delayed digit is then fed back to the V input during the next computation to allow its accumulation with the next low–order digit, and so forms the desired term in equation 3.12. The structure of a multiplier which implements this operation is shown in Figure 3.1 and is a direct result of equation 3.15. The need for the single level of registers to properly sequence the digit–wise addition within the multiplier allows the minimisation of the critical path. In fact it is possible to almost halve the number of delays present in the critical path with an appropriate placement of these registers, as shown in Figure 3.2. The immediate consequence is that the optimised multiplier can function at double the clock speed of a conventional parallel multiplier array. The number of registers required for the optimised multiplier with the shortest critical path $(2n - 1)$ is approximately two–thirds of the number of registers required for the direct implementation $(3n - 1)$, as can be seen by comparing Figures 3.1 and 3.2. The first use of this re–organised multiplier array was reported by Braun [Brau63] and is modified to include the sum of two extra nibbles in the least significant digit.

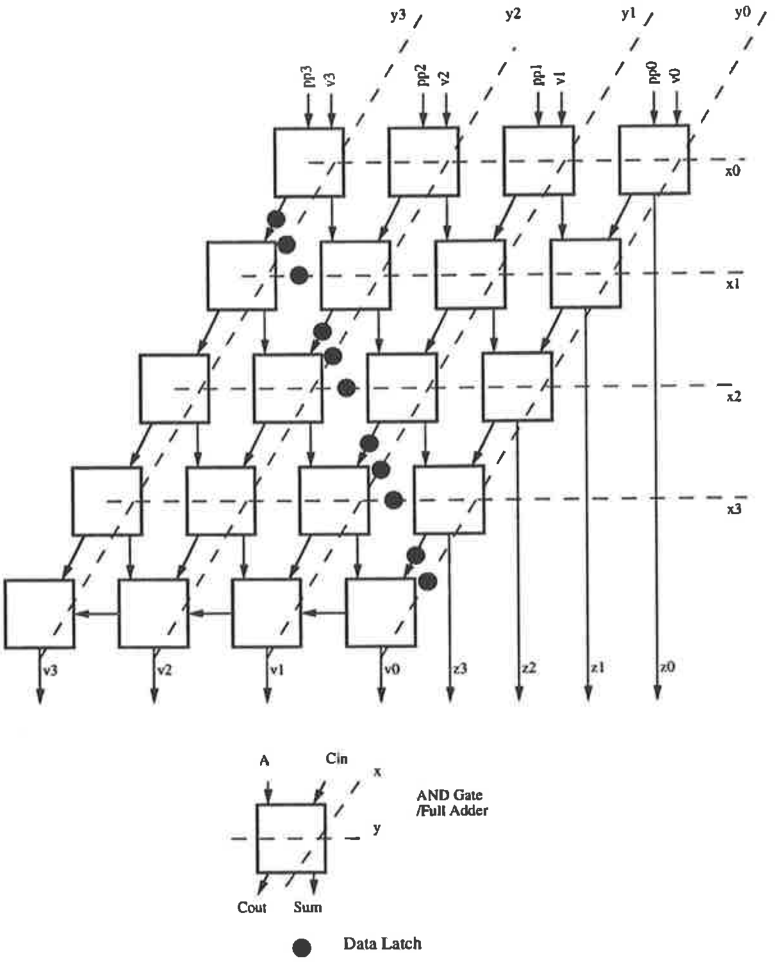


Figure 3.1: A pipelined four–bit per digit multiplier.

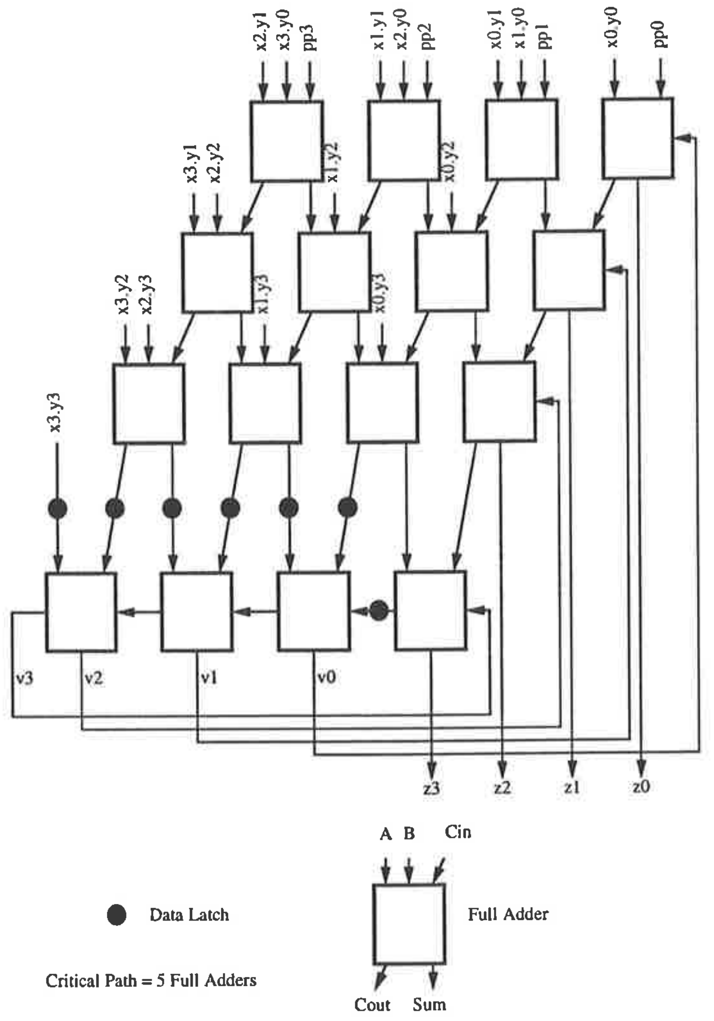


Figure 3.2: A pipelined four–bit per digit multiplier optimised for both area and critical path.

3.2 Digit–Serial Floating Point Multiplication

3.2.1 Floating Point Numbers

To be compatible with most of todays scientific and engineering computers, a general purpose co-processor should use a floating point standard such as IEEE–754 (1985) [ieee85] for implementation of arithmetic logic units. Work on floating point numbers and floating point arithmetic can be found in [ieee85, Ster74, Zyne88]. The floating point representation in its most basic form is characterised by four integers: the base, β, the precision m, a sign bit s and the exponent range e. A floating point number, N is given by:

$$N = (-1)^s \times 0 \cdot d_0 d_1 d_2 d_3 \ldots d_{m-1} \times \beta^{x_0 x_1 x_2 x_3 \ldots x_{e-1}} \tag{3.16}$$

where d_i are the mantissa bits, x_i are the exponent bits for a signed exponent field and $s = 0$ or 1 is the sign bit. The fractional part of the number is to the right of the '$\cdot$' and β is 2 for binary numbers. In addition, a signaling, a quiet Not a Number (NaN) and the two infinities ($\pm\infty$) must be encoded in the representation.

3.2.2 A Systolic Cell for Floating Point Multiplication

The digit–serial multiplier shown in Figure 3.3 is constructed from a linear array of simple systolic cells and implements the multiplication algorithm presented in the previous section. The operands indicated in Figure 3.3 are a digit–serial sequence and the **mode** input aligns the $\{X\}$ and $\{Y\}$ input operand formatting as the operands pass through each cell. The systolic cell is shown in Figure 3.4 and consists of a number of delay cells, multiplexers, a $r \times r$–bit parallel multiplier and a $2r$–bit adder.

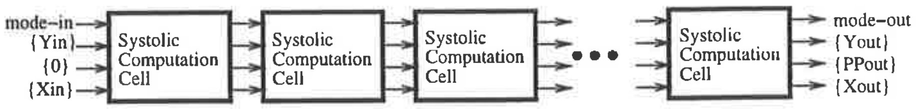

mode–in
{Yin}
{0}
{Xin}
Systolic Computation Cell
Systolic Computation Cell
Systolic Computation Cell
Systolic Computation Cell
mode–out
{Yout}
{PPout}
{Xout}

Figure 3.3: A digit–serial multiplier array.

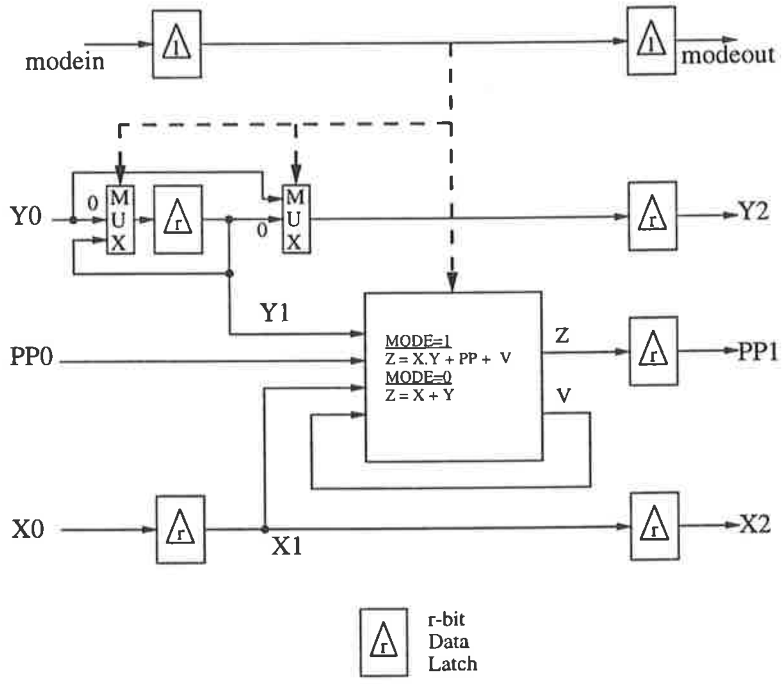


Figure 3.4: A digit–serial multiplier cell.

The speed of this multiplier is limited by the maximum speed of the multiplication and addition elements. The cell control comes from the **mode** input and operates the multiplexers and the digit–serial multiplier (shown in Figure 3.4) to reconfigure the cell depending on the operation required. The simple cell control together with the reformulation of the multiplication algorithm in terms of a pipelined digit multiplication leads to an elegant systolic multiplier cell.

3.2.3 Digit–Serial Floating Point Multiplier Model

A multiplier model implementing floating point multiplication is described briefly. Let $\{X\}$ and $\{Y\}$ be two sequences of digits entered in parallel into a machine M and let $\{Z\}$ be a sequence of digits output from the machine. The sequences are constructed from k digit 2–tuples. Each 2–tuple represents a discrete floating point number and consists of an ordered exponent and mantissa number pair. Each number is entered least significant digit first. The first e digits in a 2–tuple represent the exponent, and the remaining $(k-e)$ digits represent the mantissa. A **mode** signal is used to differentiate between exponent and mantissa digits. Let the machine M be constructed from q identical cells. Further, let the state of the machine at time n be $\{S_p(n, X_0, X_1, X_2, Y_0, Y_1, Y_2, P) : p = 1, ..., m\}$ where the state variables X_i, Y_i and P represent storage nodes for digits where i is a non-negative integer. The states X_i and Y_i are indicated on the systolic cell in Figure 3.4. The behaviour of the p^{th} cell in the machine is defined by the following recurrence relations:

$$
\begin{aligned}
X_0(p,n) &= X_2(p-1,n) \\
X_1(p,n) &= X_0(p,n-1) \\
X_2(p,n) &= X_1(p,n-1) \\
Y_0(p,n) &= Y_2(p-1,n) \\
Y_1(p,n) &= Y_0(p,n-1) && ik + 2p \leq n < ik + 2p + e + 1 \\
&= Y_1(p,n-1) && ik + e + 2p + 1 \leq n < (i+1)k + 2p \\
Y_2(p,n) &= Y_1(p,n-1) && ik + 2p + 1 \leq n < ik + 2p + e + 1 \\
&= Y_0(p,n-1) && ik + e + 2p + 1 \leq n < (i+1)k + 2p + 1 \\
P(p,n) &= X_1(p,n) + Y_1(p,n) && ik + 2p \leq n < ik + 2p + e \\
&= P(p-1,n-1) + X_1(p,n)Y_1(p,n) && ik + 2p + e \leq n < (i+1)k + 2p
\end{aligned}
$$

In the following, a two-dimensional mapping $n = ki + j + 1$ is used to express the n^{th} digit of the linear input and output sequences in terms of the i^{th} 2-tuple. Using this mapping, the one-dimensional sequences $\{X\}$, $\{Y\}$ and $\{Z\}$ can all be written in the form $\{w(i,j) : \forall i \geq 0 : 0 \leq j < k\}$ where the element $w(i,j)$ is the j^{th} digit of the i^{th} 2-tuple.

Lemma 1: The X_1 state of the p^{th} cell $X_1(p,n)$ is expressed in terms of the input digit sequence $\{X\}$ as

$$X_1(p,n) = X\left(i, \langle n - 2p \rangle_k\right) \tag{3.17}$$

where $< . >_k$ is the remainder *modulo k*.

Proof: By induction.

Lemma 1 states that the digit sequence through $X_1(p,n)$ is circular.

Lemma 2: The Y_1 state of the p^{th} cell $Y_1(p,n)$ is expressed in terms of the input digit sequence $\{Y\}$ as

$$Y_1(p,n) = Y\left(i, \langle n - 2p \rangle_k\right) \quad ik + 2p \leq n < ik + 2p + e \tag{3.18}$$

where $< . >_k$ is the remainder *modulo* k.

Proof: By induction.

Lemma 2 states that the Y_1 state is a cyclic sequence of exponent digits for the given range.

Lemma 3: The Y_1 state of the p^{th} cell of the machine M is given in terms of the digit sequence

$$Y_1(p,n) = Y(i, p+e-1) \quad ik + 2p + e \leq n < (i+1)k + 2p \tag{3.19}$$

Proof: By induction.

Lemma 3 states that the Y_1 state for the p^{th} cell is the $(p+e-1)^{th}$ digit in the sequence for all n in the range given, *i.e.* the Y digit is stored so it can be multiplied with X digits to form partial product terms.

The following theorem can be proven:

Theorem: For the machine inputs $\{X\}$ and $\{Y\}$ defined above, the state $P(p,n)$ of the p^{th} stage of the machine M at time n is given by the following:

$\forall i \geq 0 : 0 \leq j < e$ where $j = n - 2p$

$$P(p,n) = x(i,j) + y(i,j) \tag{3.20}$$

$\forall i \geq 0 : e + p - 1 \leq j < k - 1$ where $j = n - p - 1$

$$P(p,n) = \sum_{s=0}^{p-1} x(i, j-s)y(i, s+e) \tag{3.21}$$

$\forall i \geq 0 : 0 \leq r < p$ where $r = n - p$

$$P(p,n) = \sum_{s=0}^{p-1-r} x(i, k-1-s)y(i, s+r+e) \tag{3.22}$$

Proof: By Induction.

The theorem can be interpreted as follows:

- In the interval defined by equation 3.20 the exponent elements of the input 2–tuples are added independently in every cell. Only the final cell contributes to the output digit sequence.

- In the interval defined by equation 3.21, the digits output from the p^{th} cell ($p < q$) are the low–order digits of the product of the i^{th} input mantissae. The expression is not defined for $p = q$ as the low–order digits do not reach the last cell of the machine and do not constitute any part of the output digit sequence.
- In the interval defined by equation 3.22 the digits output from the last cell of the machine (when $p = q$) are the most significant digits of the product of the i^{th} input mantissae.

Figure 3.2.3 illustrates the movement of data through the Y operand path of an array of four cells which implement these recurrences. Note that no mantissa digits are output from the final cell (Cell 4). It is not desirable to lose the input operand since it should be passed to the next processing element in a systolic array. To overcome this an equivalent set of recurrences which can be shown to implement the same multiplication algorithm are:

$$
\begin{aligned}
X_0(p,n) &= X_2(p-1,n) \\
X_1(p,n) &= X_0(p,n-1) \\
X_2(p,n) &= X_1(p,n-1) \\
Y_0(p,n) &= Y_2(p-1,n) \\
Y_1(p,n) &= Y_0(p,n-1) && ik+2p \le n < ik+2p+e+1 \\
&= Y_1(p,n-1) && ik+e+2p+1 \le n < (i+1)k+2p \\
Y_2(p,n) &= Y_1(p,n-1) && ik+2p \le n < ik+2p+e+1 \\
&= Y_0(p,n-1) && ik+e+2p+1 \le n < (i+1)k+2p \\
P(p,n) &= X_1(p,n)+Y_1(p,n) && ik+2p \le n < ik+2p+e \\
&= P(p-1,n-1)+X_1(p,n)Y_1(p,n) && ik+2p+e \le n < (i+1)k+2p
\end{aligned}
$$

where i is a non–negative integer. These recurrences differ from the earlier set only in the definition of the terms involving Y_2. The modified definition of Y_2 has two advantages:

1. one gate is removed from the implementation of the control signals driving the multiplexers for the Y operands, and more significantly,
2. the digit sequence output from the Y port of the last cell in the machine is identical to the digit sequence input to the Y input ports of the first cell of the machine.

Time	Cell 1		Cell 2		Cell 3		Cell 4		
n	y_0	y_1	y_2						
			y_0	y_1	y_2				
1	e_0				y_0	y_1	y_2		
2	e_1	e_0					y_0	y_1	y_2
3	e_2	e_1	e_0						
4	e_3	e_2	e_1	e_0					
5	m_0	e_3	e_2	e_1	e_0				
6	m_1	m_0	e_3	e_2	e_1	e_0			
7	m_2	m_0	m_1	e_3	e_2	e_1	e_0		
8	m_3	m_0	m_2	m_1	e_3	e_2	e_1	e_0	
9	e_0	m_0	m_3	m_1	m_2	e_3	e_2	e_1	e_0
10	e_1	e_0	e_0	m_1	m_3	m_2	e_3	e_2	e_1
11	e_2	e_1	e_0	m_1	e_0	m_2	m_3	e_3	e_2
12	e_3	e_2	e_1	e_0	e_0	m_2	e_0	m_3	e_3
13	m_0	e_3	e_2	e_1	e_0	m_2	e_0	m_3	e_0
14	m_1	m_0	e_3	e_2	e_1	e_0	e_0	m_3	e_0
15	m_2	m_0	m_1	e_3	e_2	e_1	e_0	m_3	e_0
16	m_3	m_0	m_2	m_1	e_3	e_2	e_1	e_0	e_0
17	e_0	m_0	m_3	m_1	m_2	e_3	e_2	e_1	e_0

Figure 3.5: Operand movement through a four cell linear array of recurrence cells.

Time	Cell 1		Cell 2		Cell 3		Cell 4		
n	y_0	y_1	y_2						
			y_0	y_1	y_2				
1	e_0				y_0	y_1	y_2		
2	e_1	e_0					y_0	y_1	y_2
3	e_2	e_1	e_0						
4	e_3	e_2	e_1	e_0					
5	m_0	e_3	e_2	e_1	e_0				
6	m_1	m_0	e_3	e_2	e_1	e_0			
7	m_2	m_0	m_1	e_3	e_2	e_1	e_0		
8	m_3	m_0	m_2	m_1	e_3	e_2	e_1	e_0	
9	e_0	m_0	m_3	m_1	m_2	e_3	e_2	e_1	e_0
10	e_1	e_0	m_0	m_1	m_3	m_2	e_3	e_2	e_1
11	e_2	e_1	e_0	m_1	m_0	m_2	m_3	e_3	e_2
12	e_3	e_2	e_1	e_0	m_1	m_2	m_0	m_3	e_3
13	m_0	e_3	e_2	e_1	e_0	m_2	m_1	m_3	m_0
14	m_1	m_0	e_3	e_2	e_1	e_0	m_2	m_3	m_1
15	m_2	m_0	m_1	e_3	e_2	e_1	e_0	m_3	m_2
16	m_3	m_0	m_2	m_1	e_3	e_2	e_1	e_0	m_3
17	e_0	m_0	m_3	m_1	m_2	e_3	e_2	e_1	e_0

Figure 3.6: Operand movement through a modified four cell linear array of recurrence cells.

The second item has major significance to the testing and verification of a multiplier implementation. Figure 3.6 illustrates the movement of data through an array of cells which implement the Y recurrences. Note that the Y output from successive cells rotates the mantissa sequence by one digit so that at the final stage the sequence is identical to the input to the first stage.

3.3 A Systolic Ring Floating Point Processing Element

Floating point multiplication has been considered in previous sections of this chapter. To complete the design of a PE floating point addition must be included. The floating point multiplication algorithm is not significantly more complex than the integer algorithm. However the algorithm for floating point addition or accumulation is substantially more complex than the integer operation due to a need for the denormalisation of one operand. In previous work the denormalisations were handled using dedicated shift units [AwTa93, BrBa92] and time optimal implementations for pipelined scalar and scalar multiplication [AwTa93, TaNi92]. The new digit–serial architecture unifies the floating point multiplication and addition operations into one architecture where the operands move through a reconfigurable systolic computation cell. A new systolic ring PE shown in Figure 3.7 implements the combined function of multiplication and accumulation and appears at the block schematic level to be identical to that used for multiplication. The difference is an increased complexity in both the systolic cells and the single logic element. Interconnections are made only to nearest neighbours, as is characteristic of systolic architectures.

The PE consists of an I/O–control unit and a circular ring of delay and systolic cells. The systolic cells perform multiplication and accumulation on the input operands. (A schematic of the systolic cell is shown in Figure 3.4.) The systolic cells implement recurrence relations to perform the operations denormalisation, multiplication, addition depending on the instruction field to the ring which is encoded into the **mode** digit. The **mode** input also differentiates between the exponent and mantissa elements of the

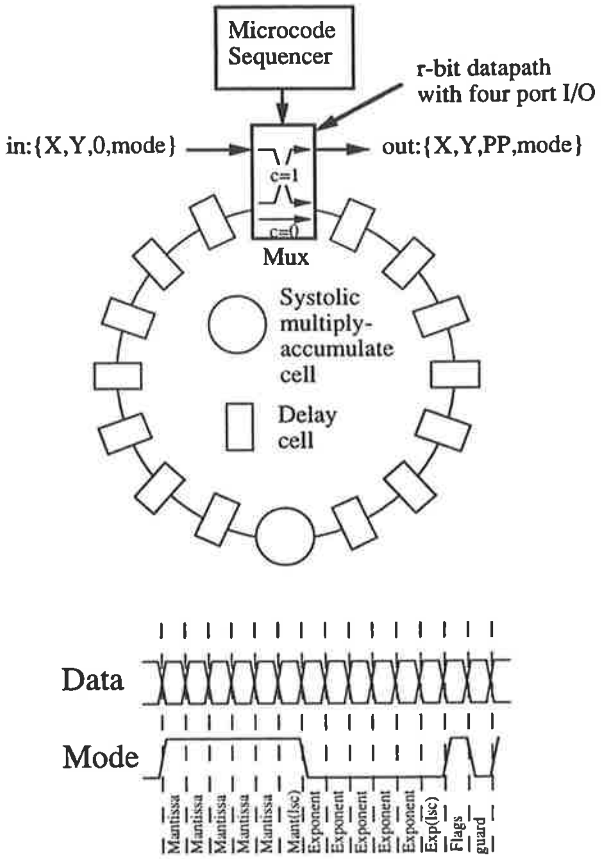


Figure 3.7: The systolic ring multiply/accumulate processing element.

operands. It allows the processing of different formats. For an m digit mantissa, it is necessary to apply m recurrences to compute the product. The data format required by the ring processor is shown in Figure 3.7. The last element in the systolic ring is a delay cell. The number of delay cells in a ring is chosen so that the length of the ring is equal to the length of the operands.

The operation can be described as follows: two operands, X and Y are input through a multiplexer into the ring with a **mode** signal, the ring is closed and the operands are circulated an integral number of cycles. The ring is then opened to output the results and input the next operands at the same time. Consider an operand format of k digits[1], $M = \lceil \frac{m}{r} \rceil$ of which represent the mantissa, and the remaining $k - M$ represent the exponent, sign, instruction and guard digits. A systolic ring can be constructed from q cells and $k - 2q$ state registers, where $q \leq M/2$. Note that m is the number of mantissa bits and r is the number of bits per digit as defined previously. The state register cells may be lumped or distributed. The k digits of the operands are input into the ring and the M recurrences are applied by circulating the operands $[M/q]$ times for the multiplication, and $[M/q] + 1$ times for accumulation. The next computation is fully pipelined. New operands are entered into the ring as the results of the previous computation are being output. The length of the ring is determined by the floating point operand.representation. For a representation of m mantissa bits and e exponent bits in an r–bit per digit representation, the ring has a length [2] L given by:

$$L = \frac{m}{r} + \frac{e}{r} + g \tag{3.23}$$

where g is the number of guard digits. It is assumed that $\lceil \frac{m}{r} \rceil = \frac{m}{r}$ and $\lceil \frac{e}{r} \rceil = \frac{e}{r}$. The number of operand circulations around the ring which are required to complete an operation is determined by the ratio of the number of digits in the mantissa to the number of systolic cells in the ring, n_c. The number of circulations, C, required for a multiplication is $\frac{m}{rn_c}$ and for an accumulation $\frac{m}{rn_c} + 1$. In the illustrated format in Figure 3.7 three digits are dedicated to instruction and guard digits. The limiting case for a processor is a single computational cell with $k - 2$ delay cells. In this case the ring can process operands whose

[1] k now includes a sign and guard digit, mantissa and exponent digits

[2] Length refers to the number of storage cells for an operand in the systolic ring.

specifications range from $k-4$ mantissa digits and a single exponent digit to $k-4$ exponent digits and a single mantissa digit. This PE provides a wide range of possible dynamic range and precision options in a single hardware implementation. The architecture can be optimised with respect to the number of systolic cells n_c, the number of circulations of the operands in the processing element C, and the number of bits per digit r. The cost of this flexibility is the number of recirculations needed for each product. The consequence of the two architectural improvements is a different systolic cell. Partial products formed in the cells of the multiplier are accumulated with the uncommitted PP multiplier input. This allows the accumulation to be performed in parallel with the multiplication and so does not contribute to the overall cell cycle time. A less detailed version of the cell as presented previously is shown in Figure 3.8. The function of the cell is (from equation 3.15):

$$Z = XY + PP + V$$

This cell function is used in two ways during floating point multiplication. During the mantissa multiplication the algorithm implemented is:

$$PP_{out} = XY + PP_{in} + V$$

where V is the high–order digit generated by the pipelined multiplier and PP_{in} is the partial product input from the previous cell. During the exponent addition mode, the following function is implemented:

$$\begin{aligned} PP_{out} &= X \times 1 + V + Y \\ &= X + Y \end{aligned}$$

The value of V is zero in this part of the computation as there is no high–order output from the product $X \times 1$. This implements the exponent addition in each cell.

Figure 3.9 shows the logical function of the systolic cell when it is performing a multiplication operation on the mantissa of two floating point operands. The output from the systolic ring multiplier (high–order digits denoted by XY') using the results of the section 3.1, equation 3.12 is:

$$XY' = \sum_{i=0}^{N-1} \sum_{k=0}^{q-1} \left[A_{i0k}\beta^i + \sum_{j=1}^{N-2-i} \left(A_{i(j+1)k} + B_{ijk} \right) \beta^{i+j+1} + B_{i(N-1-i)k}\beta^N \right] \tag{3.24}$$

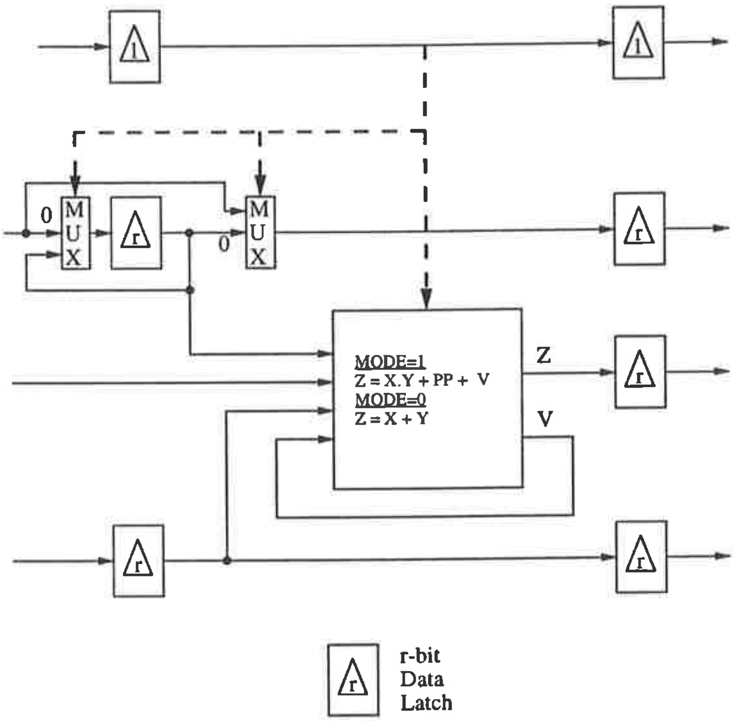


Figure 3.8: The systolic multiply/accumulate cell.

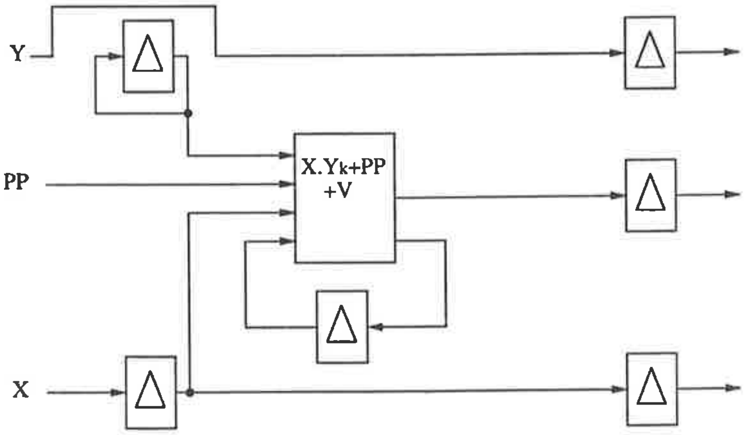


Figure 3.9: The logical function of the systolic cell during multiplication.

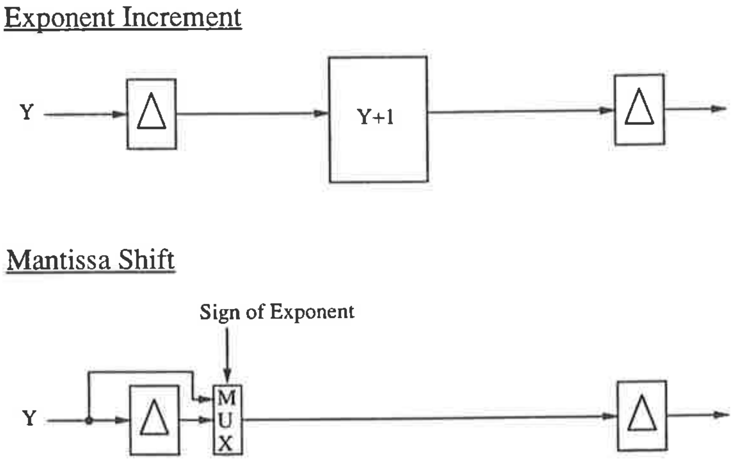


Figure 3.10: The logical function of the systolic cell during denormalisation.

Figure 3.10 shows the logical function of the systolic cell when it is performing a denormalisation operation for the floating point accumulation function. Two operations are required in this mode; one to increment an exponent difference, and the other to shift the appropriate mantissa. Both operations are performed by the cell on the exponent and mantissa fields of the required operand.

3.4 Performance Metric of a Rectangular Systolic Array Processor

It has been common to use a variety of performance metrics when designing arithmetic units. These metrics are typically functions of execution time, power and area. Rather than restrict the study to the optimisation of a single PE using an arbitrary metric, a more realistic goal is the minimisation of the total job time for the computation of large order matrix products when executed on a square array of PEs. Implementation of a real systolic array constructed from a rectangular array of elementary inner–product–accumulate processors imposes some physical constraints upon the system architect. It is assumed that the following two constraints are realistic:

1. the design is limited by some maximum area of active circuitry, determined by either physical limitations such as thermal dissipation, or cost;
2. the design is limited by some maximum memory bandwidth.

To analyse the performance of the processing array it is assumed that the architecture of the PE allows variation in both execution time and chip area. Architecture classes ranging from bit–serial to fully parallel implementations of PEs are two extremes of these variables. Let the chip area of a PE be A_{pe} and let T_{pe} be the execution time required to process one set of operands or a wavefront in a PE. It is assumed that the area constraint is A_{proc}, so the processor active area is less than or equal to A_{proc} and the maximum bandwidth constraint for the processor is B_{proc}. Let p be the order of a square systolic array. The number of PEs is p^2. Under the above assumptions, the number of PEs in the system is expressed as a function of the total active area, A_{proc} as:

$$p^2 = A_{proc}/A_{pe} \tag{3.25}$$

or

$$p = \sqrt{A_{proc}/A_{pe}} \tag{3.26}$$

The number of operands required to drive the array inputs for each wavefront entered into the array is $2p$. The time to fetch these operands from memory or the time for one wavefront is $T_{wf} = 2p/B_{proc}$. This is an upper bound for the execution time of each PE. Thus the bandwidth constraint provides an execution time constraint for the processing elements of the form:

$$T_{pe} \leq 2p/B_{proc} \tag{3.27}$$

Hence:

$$B_{proc} \leq 2p/T_{pe}$$

and substituting equation 3.26 gives:

$$B_{proc} \leq \frac{2\sqrt{A_{proc}}}{T_{pe}\sqrt{A_{pe}}} \tag{3.28}$$

Consider the execution of a product of square matrices of order N on a systolic array of order p, where $N >> p$ and for simplicity $N \bmod p = 0$. The number of partitions to be computed in the product is $\lceil N/p \rceil^2$, where $\lceil x \rceil$ represents the least integral value greater than or equal to x. The pipelined time to compute each partition is given by the time for the array to process all of the wavefronts in any given partition. If the processing time for one wavefront is T_{pe}, then the time to compute one partition is $N \times T_{pe}$. It is convenient to assume that all partitions are the same size, in which case the time required to compute the matrix product, T_{job} is:

$$\begin{aligned} T_{job} &= NT_{pe}\left\lceil \frac{N}{p} \right\rceil^2 \\ &= \frac{N^3}{A_{proc}} T_{pe} A_{pe} \end{aligned} \tag{3.29}$$

Start–up delays are ignored and it is assumed that all partition computations are fully pipelined. Under these assumptions the job execution time is minimised by minimising the area–time (AT) product of the PEs. If the bandwidth constraint (3.27) is considered in terms of the area constraint (3.25), the following expression relates the area and time metrics of the PE:

$$T_{pe}\sqrt{A_{pe}} \leq \frac{2\sqrt{A_{proc}}}{B_{proc}} \tag{3.30}$$

A *matched* system is one in which the processor fully utilises the available bandwidth from the memories to obtain maximum processing performance. For a *matched* system, the area and time metrics of the elements are related by:

$$T_{pe}\sqrt{A_{pe}} = \frac{2\sqrt{A_{proc}}}{B_{proc}}$$

Hence, the bandwidth requirement of a matched processor is (by rearranging 3.30):

$$B_{proc} = \frac{2\sqrt{A_{proc}}}{T_{pe}\sqrt{A_{pe}}} \tag{3.31}$$

3.5 Architecture Optimisation

To optimise the PE over the range of architectural possibilities a number of GaAs technologies have been studied. The HGAAS–II [Vite92] process has been characterised in terms of the properties of a limited set of fundamental circuits, typically a logic gate, a 2:1 multiplexer, a full adder and a data latch. Estimates of the area and time performance

Circuit Element	Area (m^2)	Gate Delay (ms)
NOR gate	$a_g = 0.9 \times 10^{-9}$	$t_g = 0.15 \times 10^{-3}$
2:1 Mux	$a_m = 3 \times a_g$	$t_m = 2 \times t_g$
Full adder	$a_a = 11 \times a_g$	$t_a = 2 \times t_g$
Register	$a_r = 7 \times a_g$	$t_r = 5 \times t_g$
Multiplier	$a_{mul} = r^2 \times a_a + (2r-1)a_r$	$t_{mul} = (2r+3)t_g$

Table 3.1: Area and time metrics of characteristic circuits for a E/D GaAs process.

metrics of these cells is given in Table 3.1. Typical values for propagation delay and area of a GaAs NOR gate are $150ps$ and $900\,\mu m^2$, respectively.

The number of delay cells, n_d, can be expressed in terms of the delay length of the ring, L, and the number of computation cells, n_c, where each computation cell has two delays, $n_d = L - 2n_c$. It can be seen from Figure 3.4 that the systolic cell area, A_{cell}, can be approximated by the area of its constituents: seven r–bit registers, two 2–bit registers, three r–bit multiplexers and an $r \times r$ pipelined digit–serial multiplier. An estimate of ten gates has been used for instruction decoding. Using the characteristics of this E/D GaAs process shown in Table 3.1, the area estimate for the systolic cell is:

$$A_{cell} = 7ra_r + 4a_r + 3ra_m + 10a_g + r^2a_a + (2r-1)a_r = Ar^2 + Br + C$$

where $A = a_a$, $B = 9a_r + 3a_m$ and $C = 3a_r + 10a_g$. The area of a delay cell is $A_d = (4r+2)a_r$. Thus the area of a PE, A_{pe}, can be written as:

$$A_{pe} = n_cA_{cell} + n_dA_d + A_{con}$$

where A_{con} is the area of the PE control. The delay of a systolic cell, T_{cell}, is assumed to be determined by the setup and hold time of the registers plus the critical path delay of the digit–serial multiplier discussed earlier. The multiplier critical path contains $r+1$ full adders and one AND gate. Letting the gate delay be t_g the total multiplier delay is $(2r+3)t_g$ and the register delay is $5t_g$ giving a total cell delay of $T_{cell} = 2(r+4)t_g$. Using equation 3.23, the time to circulatc an operand through the ring, T_c, is given by:

$$T_c = (\frac{m}{r} + \frac{e}{r} + g)2(r+4)t_g$$

The number of circulations required for a multiplication is $\frac{m}{rn_c}$ and for an accumulation $\frac{m}{rn_c} + 1$. The PE execution time, T_{pe}, is given by multiplying the cell execution time by the number of clocks to complete the required multiply/accumulate operation as follows:

$$\begin{aligned} T_{pe} = & \quad (\tfrac{2m}{rn_c} + 1)(\tfrac{m}{r} + \tfrac{e}{r} + g)T_{cell} \\ & = (\tfrac{1}{r^2}\tfrac{16m(m+e)}{n_c} + \tfrac{1}{r}(\tfrac{4m(m+e)}{n_c} + 8(m + e + \tfrac{2mg}{n_c})) \\ & \quad +2m + 2e + \tfrac{4mg}{n_c} + 8g + 2rg)t_g \end{aligned}$$

The AT product of the PE can be represented by the polynomial expression

$$A_{pe}T_{pe} = (a_0r^2 + a_1r + a_2 + a_3r^{-1})(t_0r^{-2} + t_1r^{-1} + t_2 + t_3r) \qquad (3.32)$$

where

$a_0 = 11n_ca_g,\ \ a_1 = n_c(B - 8a_r) + 4a_rg,$

$a_2 = n_c(C - 4a_r) + 4a_r(m + e) + 2ga_r + A_{con},$

$a_3 = 2(m + e)a_r,$

$t_0 = \frac{16m(m+e)}{n_c},$

$t_1 = \frac{4m(m+e)}{n_c} + 8m + 8e + \frac{16mg}{n_c},$

$t_2 = 2m + 2e + 8g + \frac{4mg}{n_c},\ \ t_3 = 2g.$

The evaluation of the partial derivative of the AT product with respect to the number of cells gives a result which is always negative. Hence, the AT product is minimised by using the maximum number of systolic cells $n_c = \frac{\partial m}{\partial 2r}$ which requires two circulations of the operands for a multiplication or denormalisation and three for an accumulation operation. Expressing the number of cells n_c as a function of the number of bits per digit, r allows the AT product to be expressed as a function of r:

$$\begin{aligned} A_{pe}T_{pe} = \quad & [r(\tfrac{mA}{2} + 4a_rg) + 0.5m(B - 8r) + 4a_r(m + e) + 2ga_r + A_{con} \\ & +0.5r^{-1}m(C - 4a_r) + 2(m + e)a_r]10((m + e)/r + g)(r + 4)t_g \qquad (3.33) \end{aligned}$$

In the above analysis it has been assumed that all functions are continuous and differentiable. These assumptions neglect physical restrictions such as the requirement for the number of delay cells to be integral. As a consequence the model was extended to incorporate into equation 3.33 the following constraints: $\frac{e}{r} \leftarrow \lceil\frac{e}{r}\rceil$, $\frac{m}{r} \leftarrow \lceil\frac{m}{r}\rceil$ and $n_d \leftarrow \lceil(\frac{m}{r} + \frac{e}{r} + g) - 2n_c)\rceil$. This defines the discrete AT model.

3.5.1 Area–Time Model Evaluation

Evaluation of the resulting expression derived in equation 3.32 for a mantissa length of 32–bits, and an exponent length of 16–bits gives a three–dimensional plot of the AT metric for the systolic ring processor shown in Figure 3.11. This is a function of both the number of bits per digit, and the number of systolic cells in the systolic ring.

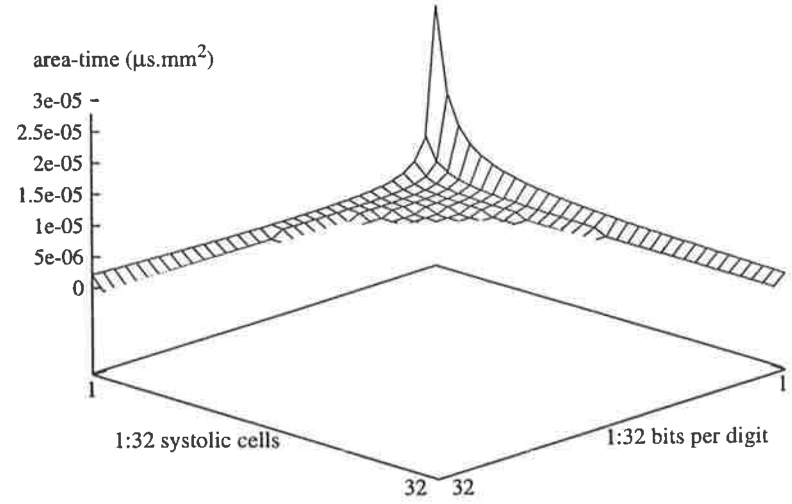


Figure 3.11: The AT metric for the systolic ring multiplier.

Figure 3.12 is a plot of the evaluation of a continuous time and discrete time model for the AT performance metric versus the number of bits per digit. Figure 3.12 is for multipliers with 32 mantissa bits, 16 exponent bits and two guard or control digits and implemented with the maximum number of systolic cells permissible in a ring *e.g.* $m/2r$ cells. Figure 3.12 shows that a PE with 4–bits per digit would be AT minimum for this particular processor and number format. Local minima in the AT product for the discrete time curve are associated primarily with $\frac{m}{r} = \lceil \frac{m}{r} \rceil$ and so for mantissa lengths which differ from this example, the optimal number of bits per digit may differ from four. It is apparent from the graph that the continuous model represents a lower bound for the AT performance metric of the processor. The AT product over all r and n_c has also been evaluated for a number representation with 64 mantissa bits, 16 exponent bits and two guard or control digits. The results of this evaluation are shown in Figure 3.13 and it can be seen that 4–bits per digit is still an optimal solution for double precision numbers. This would require a simple lengthening of the systolic ring to achieve fast

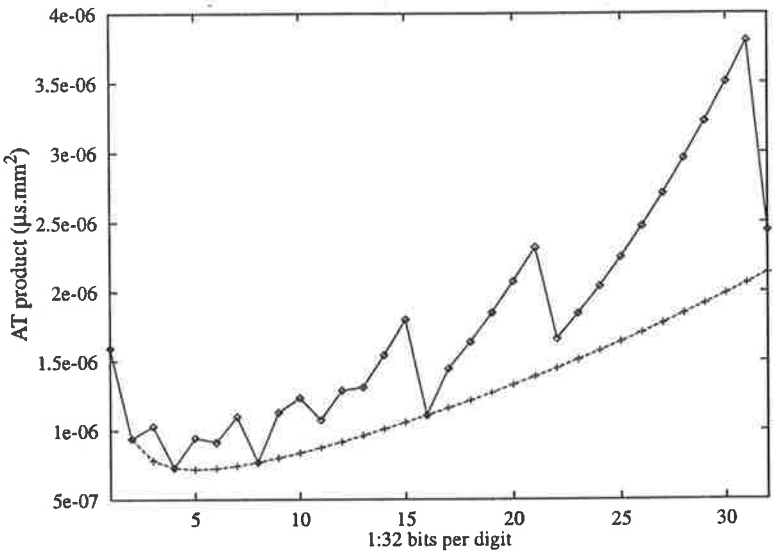


Figure 3.12: The AT metric for the systolic ring multiplier for a continuous model and a constrained model with the maximum number of systolic cells versus the number of bits per digit for $m = 32$ and $e = 16$.

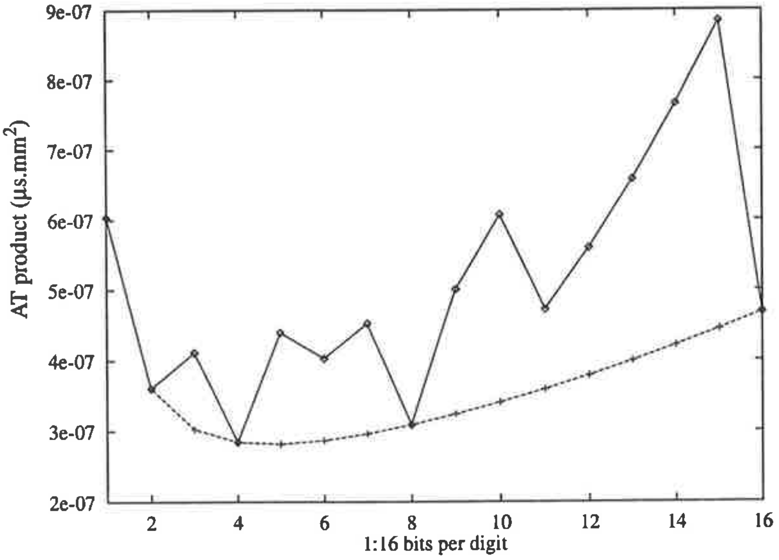


Figure 3.13: The AT metric for the systolic ring multiplier for a continuous model and a constrained model with the maximum number of systolic cells versus the number of bits per digit for $m = 64$ and $e = 16$.

double precision computation using the systolic array.

3.5.2 Processor Bandwidth Requirement

The bandwidth requirements of the ring under the constraint of a given maximum area are determined by equation 3.31which is plotted as a function of both the number of bits per digit and the number of systolic cells in the systolic ring in Figure 3.14. As the number of systolic cells increase, the bandwidth requirement rises sharply but, as the number of bits per digit in the number representation increases the bandwidth curves saturate. This is due to the simple models used to implement the arithmetic units. If arithmetic speed–up techniques were used, such as booth encoding for larger bit per digit implementations, the bandwidth curves would continue to rise in line with the increased performance of the system.

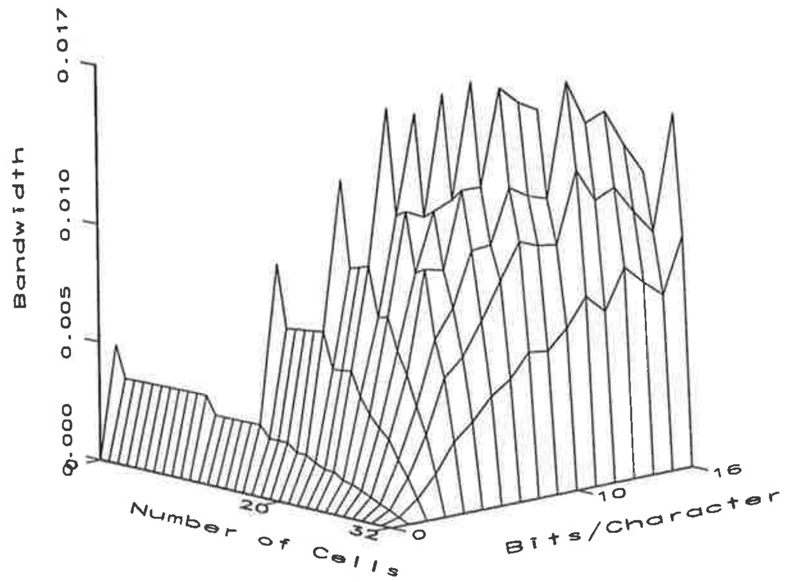


Figure 3.14: The bandwidth metric for the systolic ring processing element under the constraint of constant area.

Since the processor performance is limited by the available bandwidth, it is useful to know what architectures are under– or over–utilised. The constraint on the system imposed by the memory bandwidth in equation 3.28 can be rewritten as:

$$B_{proc} - \frac{2\sqrt{A_{proc}}}{T_{pe}\sqrt{A_{pe}}} \leq 0 \tag{3.34}$$

Equation 3.34 is graphed in Figure 3.15 for a range of values of r and n with an active area of ten million gates assumed for the complete processor. Figure 3.15 shows that while equation 3.34 is greater than zero the constraint is met and the processor bandwidth is fully utilised. When the expression equals zero the processor and memory subsystem requirements are matched, and when the expression becomes negative, the bandwidth is no longer fully utilised. An increase in the order of the array can return the bandwidth to full utilisation. The available memory bandwidth is $400 Mbytes/s$. A significant set of possibilities exist for the PE architecture which meet the bandwidth constraint.

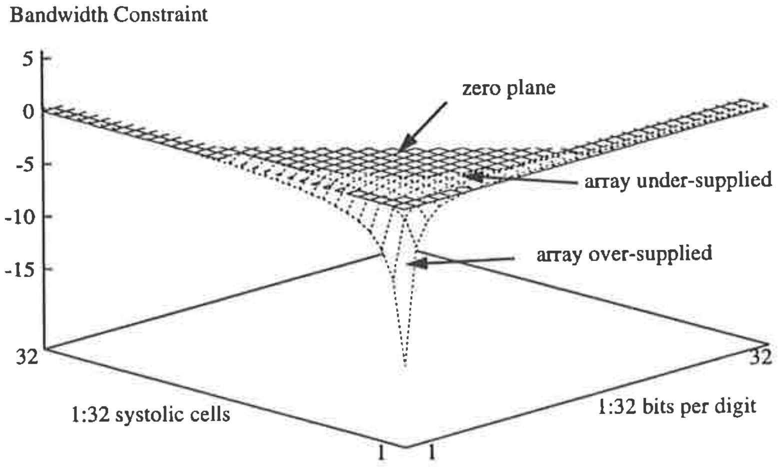


Figure 3.15: The bandwidth constraint for the systolic ring processing element for ten million gates.

3.6 Summary

Digit–serial arithmetic has been studied for a systolic cell which performs multiplication. The algorithm for the multiplication of an N digit number by an M digit number, where the digit base β is 2^r, is rewritten in terms of the binary representation of the digits. The result is used to show that for digit–serial multiplication a full $r \times r$–bit multiplication of digit pairs is not required at each time step. In particular, it shows that the digits which contribute to the output sequence are formed from the accumulation of partial multiplications whose critical paths are approximately half that of a $r \times r$–bit parallel multiplier. A systolic ring digit–serial multiplier is described which uses a single-level pipelined parallel multiply/accumulate cell to implement the partial multiplications indicated by the decomposition. An architecture for a systolic floating point processing element which can perform multiplication, accumulation and denormalisation of two floating point operands has been proposed. The performance of a rectangular systolic array of PEs was analysed using the metrics of area, time and bandwidth. It was found that the AT metric should be minimised to give the smallest job time for a matrix product. The architecture has been optimised for use in a class of systolic array processors to perform matrix computations by minimising the AT metric of the processing element. An optimal implementation has been shown to consist of arithmetic units with 4–bits per digit (nibble) and four systolic cells in the ring for HGAAS–II GaAs technology using an IEEE single precision format.

Chapter 4

Design, Layout and Simulation

4.1 Introduction

In this chapter the PE is designed and simulated and a layout produced for remote fabrication. The building blocks for the systolic ring PE include a data flip–flop, toggle flip–flop, full adder, multiplexers, clock generation, clock distribution circuits and bonding pads. A conventional 6–NOR gate data flip–flop is modified and several new versions produced to incorporate clear and preset functions. The signal integrity of the data flip–flop is critical for the correct operation of the chip since data storage circuits take up most of the layout area. Various adder circuit implementations constructed from DCFL and SDCFL classes are studied. The full adder circuit is used throughout the PE and so the area–time characteristics of each implementation was studied to find a minimum to satisfy the processor model from Chapter 3. The architectural studies presented in the previous chapter show clearly that four–bits per digit is an optimal implementation for a particular class of PE using the HGAAS–II process. It was decided to implement the PE with the following requirements:

- use digital GaAs process (from Thomson–CSF, $0.8\mu m$ SAGA)
- area available is $15mm^2$ due to cost constraints
- PE chip functions
  - floating point multiplication
  - floating point addition (includes denormalisation)

  - floating point flags

- extended single precision floating point format
- state machine controller for data I/O
- individual circuits must be testable, therefore include separate test structures

Schematics and layouts are produced using basic circuits for the systolic cell, systolic ring controller, delay cell and flag checking. A variable speed clock generator was needed and a design based on ring oscillators and was produced. Clock distribution for a systolic ring is studied. Design constraints were derived from data flip–flop timing characteristics so the correct transfer of data could be achieved. The physical clock distribution circuit must be carefully simulated and includes buffers and a H–tree interconnection structure to distribute the clock signal to the data flip–flops in the PE. Power circuits are designed and checked for current density limits and voltage variation limits due to resistance and inductance in the power rails. A floorplan of the chip is presented and a final layout produced. Fabrication and packaging details are also discussed.

4.2 Floating Point Representation

An extended floating point representation is defined which exceeds both the dynamic range and precision of the single precision IEEE–754 standard [ieee85]. Referring to equation 3.16 in Chapter 3, the specified minimum number of bits in the representation is $e \geq 11$, $m \geq 32$ where e is the number of exponent bits and m is the number of mantissa bits. The exponent bias is unspecified, however the minimum exponent value is $E_{min} \leq -1022$ and the maximum exponent value is $E_{max} \geq 1023$. Note the encoding of non–zero values may only be used in extended formats. To align the number of bits with the total number of digits in the PE for a 4–bit per digit implementation $e = 12$ and $m = 32$. The operand format consists of eight mantissa digits, three exponent digits, one flag and one guard digit. The flag digit contains a zero flag and a sign flag. The guard digit stores the most significant digit of the result. This architecture implements a multiplication with two circulations of the data around the ring, so the processor requires a total of 55 clocks to perform a multiplication and accumulation. Rounding is not implemented because an extended number format is used and all bits are carried to the next operation unmodified. Conversion to other formats for floating point compatibility must be done externally to the PE. The final architecture of the PE chip is shown in Figure 4.1.

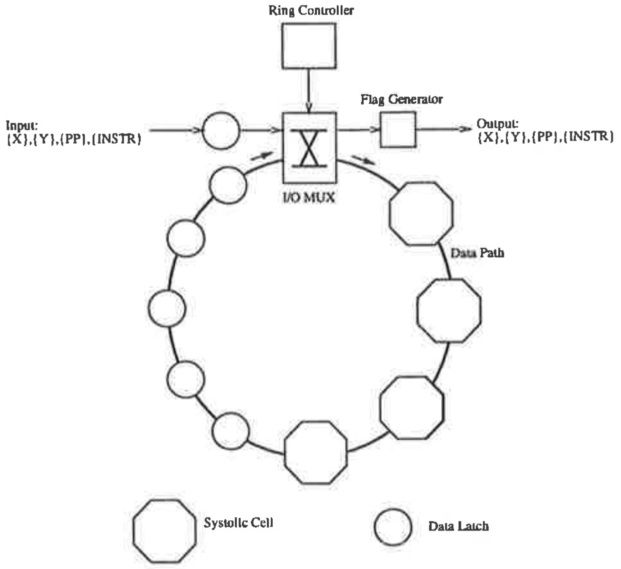


Figure 4.1: Processing element architecture.

4.3 GaAs Circuit Design, Simulation and Layout

This section deals with the design of the components of the chip that form the building blocks for the PE.

4.3.1 Data Flip–Flop

A data latch is required to store data that is recirculated around the systolic ring. The following requirements are considered essential in the design:

- operation from DC to $1GHz$
- area–time efficient storage since latches make up around 50% of a layout
- ability to be cleared or preset into a state

- single ended clock input
- complementary output available

Master–slave latches may be implemented in a variety of ways using current mode techniques or usual logic gate (e.g. NOR). Since a small power supply voltage (1 to 2 V) is used, current mode techniques cannot be used since they require a higher power supply voltage to correctly bias the circuit. Multiple power supply voltages may be used to overcome this but this increases the layout complexity. Figures 4.2, 4.3 and 4.4 show different half latches using simple gates which are transparent on half of the clock cycle. These may be implemented in Scheme 1 (Figure 4.5) where two phase non–overlapping clocks are used or Scheme 2 (Figure 4.6) where a single clock line is used to avoid distributing two clock phases. Note that in Scheme 2 only half of the logic is active at any time but may be more clock skew tolerant than Scheme 1.

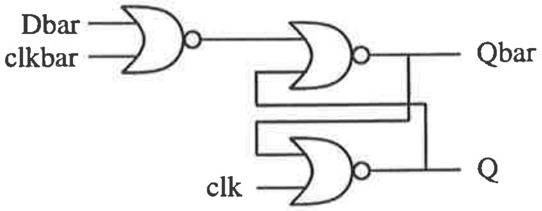


Figure 4.2: Master–slave half latch (1).

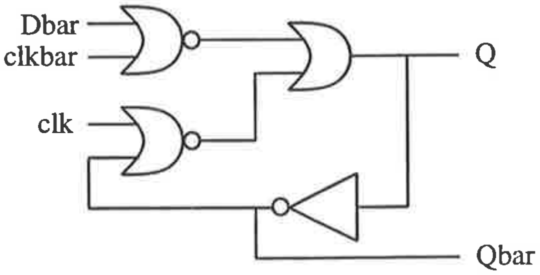


Figure 4.3: Master–slave half latch (2).

The 6–gate data flip–flop is a good approach and has been used successfully by foundries such as Vitesse [Vite92] and TriQuint [TriQ91]. The basic design for this latch is a

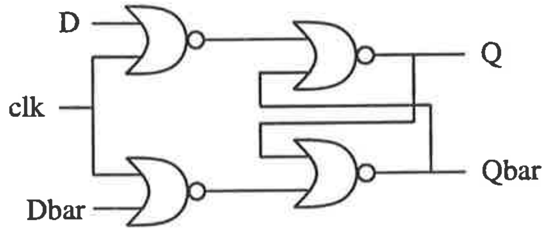
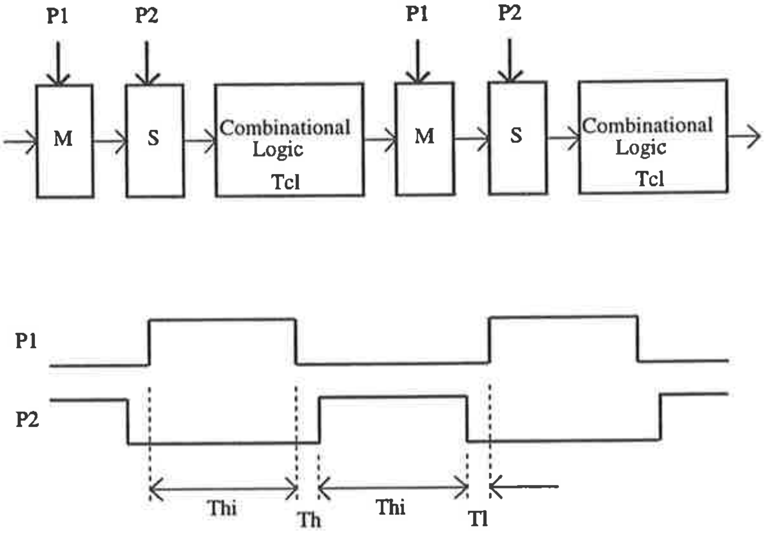


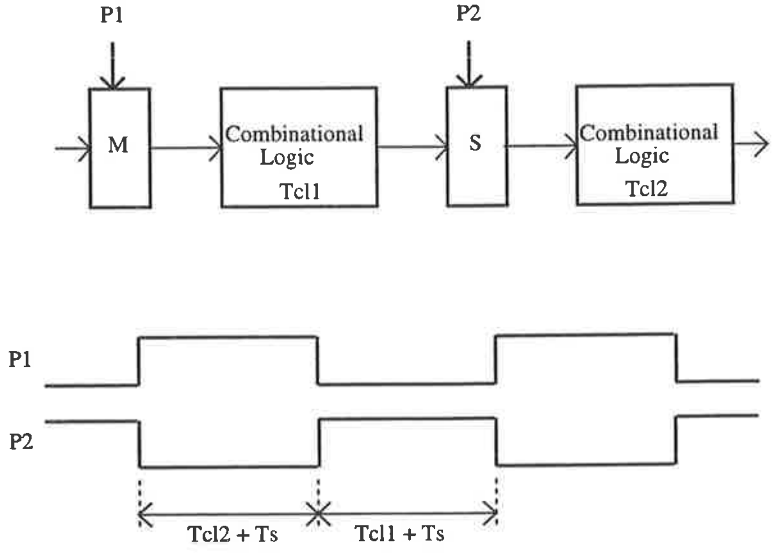
Figure 4.4: Master–slave half latch (3).



Two phase non–overlapping clocks, P1 and P2.

Timing constraint: Thi + Tl > Tcl, Th is the hold time.

Figure 4.5: Clock Scheme 1.



Single phase clock, P1 with P2 generated at each slave.

Figure 4.6: Clock Scheme 2.

negative edge triggered data flip–flop with a single D input as shown in Figure 4.7. Figure 4.8 shows the same data flip–flop but with $\overline{D}$ being required as well. Figures 4.9, 4.10 and 4.11 show similar arrangements but with a clear signal incorporated. All flip–flops have the Q and $\overline{Q}$ outputs available and are negative edge triggered. Table 4.1 shows the characteristics of the data flip–flops. Data flip–flops that have both D and $\overline{D}$ inputs will toggle at a higher frequency and may make the overall circuit operation slightly faster if $\overline{D}$ does not have to be generated. The interconnection would be simpler and the number of gates used would be reduced slightly if a single input was used.

A problem can occur for the latch shown in Figure 4.9; if the clock is low and $\overline{D}$ is low when the clear goes low, a '1' is latched to Q instead of '0'. This problem disappears if clear goes low when the clock is high. To overcome this, the clear signal must be held low long enough for $D = 0$ to propagate to the next latch. Alternatively, the circuit shown in Figure 4.10 overcomes the problem by gating the $\overline{D}$ input with the clear signal. The latch shown in Figure 4.11 is impractical for design in GaAs since there is a 4–input NOR gate required with a fan out of three which would be slow and have poor noise

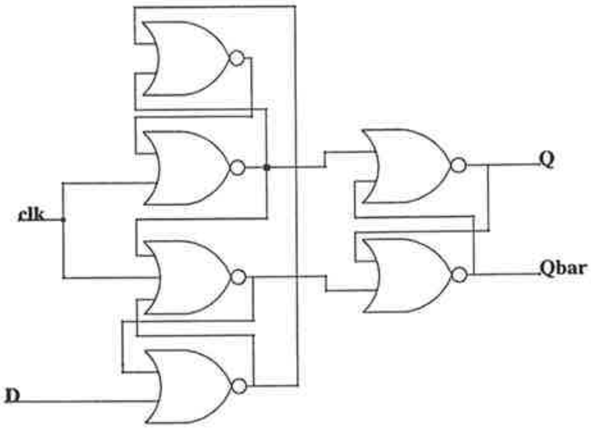


Figure 4.7: Data flip–flop 1: Schematic of a 6–NOR data flip–flop with a single input.

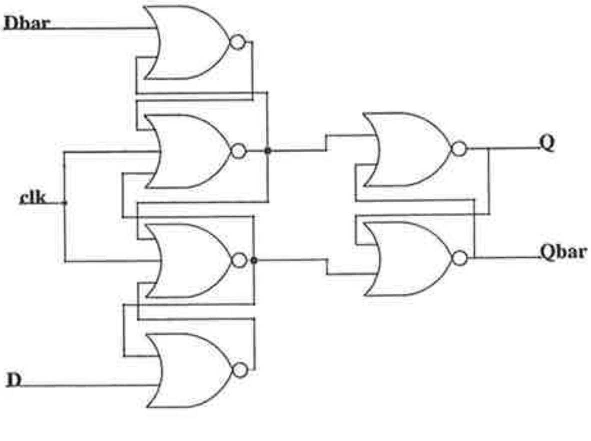


Figure 4.8: Data flip–flop 2: Schematic of a 6–NOR data flip–flop with D and $\overline{D}$ inputs.

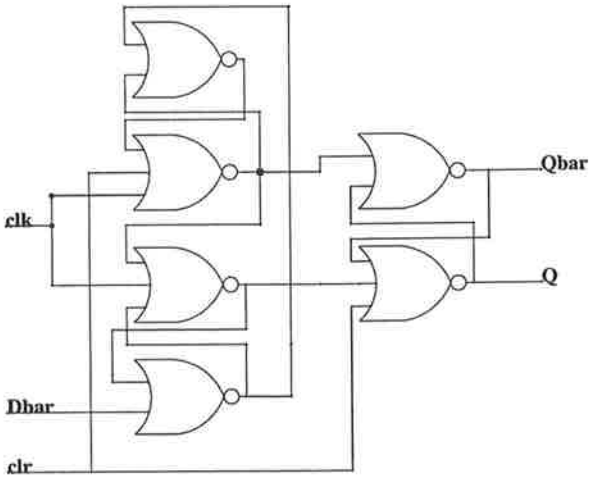


Figure 4.9: Data flip–flop 3: Schematic of a 6–NOR data flip–flop with clear.

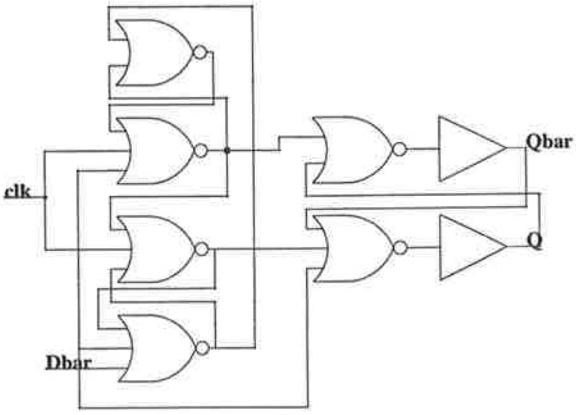


Figure 4.10: Data flip–flop 4: Schematic of a 6–NOR data flip–flop with improved clear.

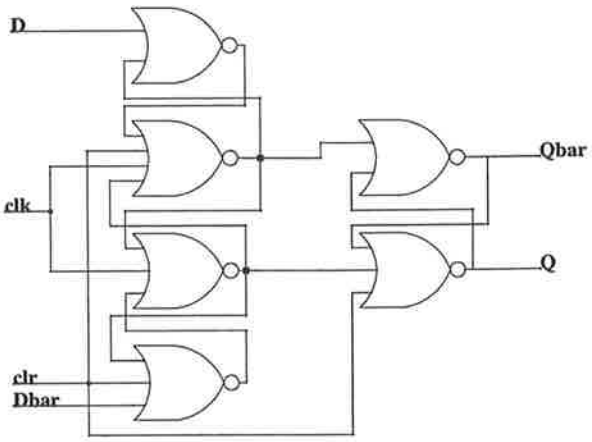


Figure 4.11: Data flip–flop 5: Schematic of a 6–NOR data flip–flop with clear, D and $\overline{D}$ inputs.

data flip–flop (Figure No.)	1 (4.7)	2 (4.8)	3 (4.9)	4 (4.10)	5 (4.11)
set–up time (gate delays)	2	1	2	2	1
hold time (gate delays)	3	3	3	3	3
input	D	$\overline{D}$,D	$\overline{D}$	$\overline{D}$	D,$\overline{D}$
output	Q,$\overline{Q}$	Q,$\overline{Q}$	$\overline{Q}$,Q	$\overline{Q}$,Q	$\overline{Q}$,Q
clear	-	-	asynch.	asynch.	asynch.
input	$\overline{D}$	D,$\overline{D}$	D	D	$\overline{D}$,D
output	$\overline{Q}$,Q	$\overline{Q}$,Q	Q,$\overline{Q}$	Q,$\overline{Q}$	Q,$\overline{Q}$
preset	-	-	asynch.	asynch.	asynch.

Table 4.1: Characteristics of the data flip–flops.

immunity. The latch in Figure 4.10 is considered the best alternative if a reset or clear function is required otherwise the latch in Figure 4.11 could be used. Ring notation for the latch with clear or preset is shown in Figure 4.12. Figure 4.13 shows the resulting layout and Figure 4.14 is a SPICE simulation of the flip–flop operating correctly at 1 *GHz*.

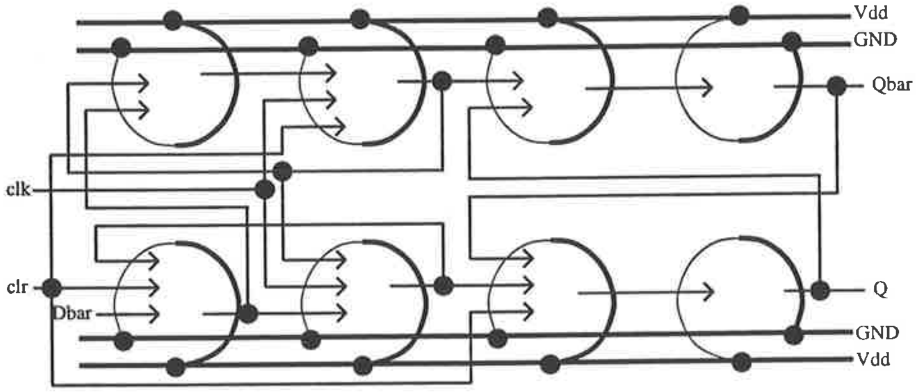


Figure 4.12: Ring notation of a GaAs data flip–flop with clear or preset.

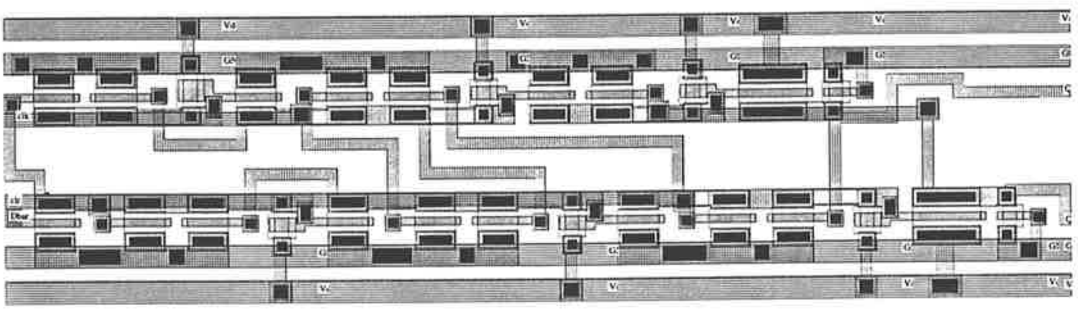


Figure 4.13: Layout of a GaAs data flip–flop using ring notation.

Set–up and Hold

The set–up time of a latch is the time before the clock edge where the input must be held stable. The hold time of a latch is the time after the clock edge where the input must be held stable. To find the point at which a latch becomes metastable, a simulation of the latch was carried out with a transition occurring near the clock edge. The set–up time was found to be 150*ps*, the hold time is 120*ps* and the propagation delay is 360*ps*.

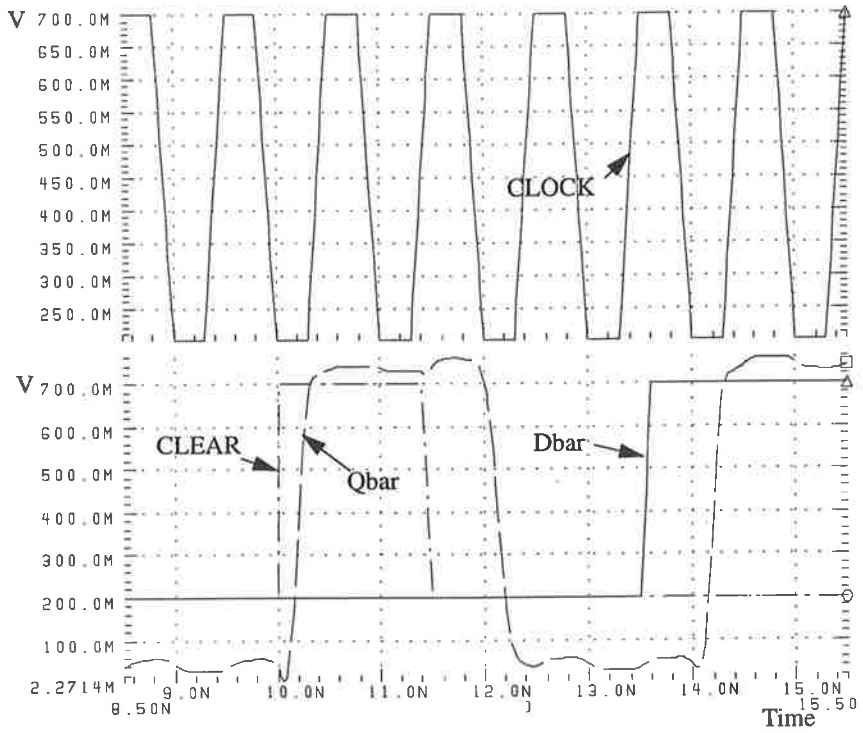


Figure 4.14: SPICE transient response simulation of a GaAs data flip–flop with a $1GHz$ clock.

Toggle Flip–Flop

Toggle flip–flops are needed in the clock divider circuit and operate by inverting their outputs.at each clock cycle. The toggle flip–flop was constructed from a data flip–flop with the outputs fed back to the inputs, $D = \overline{Q}$, $\overline{D} = Q$. The resulting layout is shown in Figure 4.16.

4.3.2 Full Adder

Full- and half adders with equal sum and carry times were required for the digit–serial multiplier accumulator. The equations for the sum and carry terms generated from the a, b and c inputs were:

$$\begin{aligned} H_k &= a \oplus b \\ S &= H_k \oplus c \\ C &= a.b + H_k.c \end{aligned} \qquad (4.1)$$

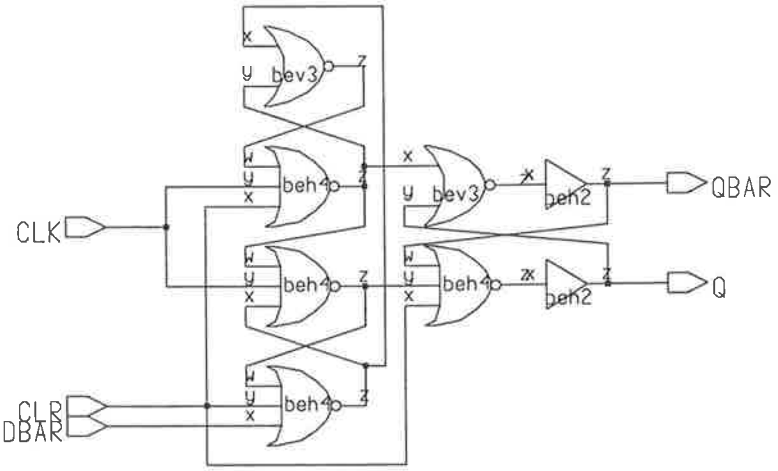


Figure 4.15: Final version of the data flip–flop schematic with clear.

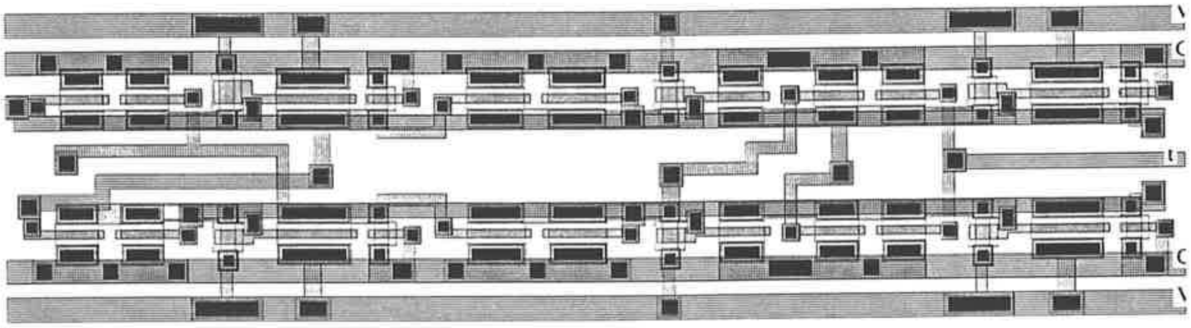
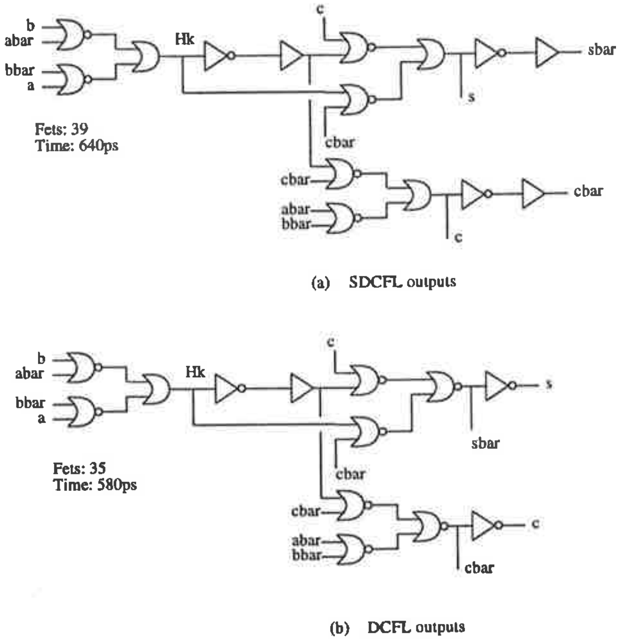


Figure 4.16: Toggle flip–flop layout.

The signals available were a, $\overline{a}$, b, $\overline{b}$, c and $\overline{c}$ and the outputs S, $\overline{S}$, C and $\overline{C}$ should be produced. There are several ways to generate the sum and carry using either DCFL, SDCFL, or a combination of both. Figures 4.17 and 4.18 show various implementations using equations 4.1.



(a) SDCFL outputs

(b) DCFL outputs

Figure 4.17: SDCFL implementation of a full adder using adder half equations with (a) SDCFL outputs and (b) DCFL outputs.

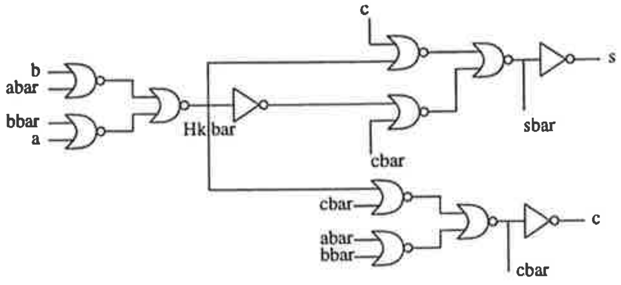


Figure 4.18: DCFL implementation of a full adder using adder half equation with DCFL outputs.

The full adder sum and carry equations are written below:

$$\begin{aligned} S &= abc + \overline{a}.\overline{b}.c + a.\overline{b}.\overline{c} + \overline{a}.b.\overline{c} \\ C &= a.b + a.c + b.c \end{aligned} \qquad (4.2)$$

Various independent implementations for sum and carry generation are shown in Figures 4.19 and 4.20, respectively, using equations 4.2. The adder may be composed of any combination of these circuits. The circuit area is proportional to the number of devices if the same layout strategy is used. Table 4.2 shows the area (device)–delay product for each full adder implementation assuming a load of two external devices for each output.

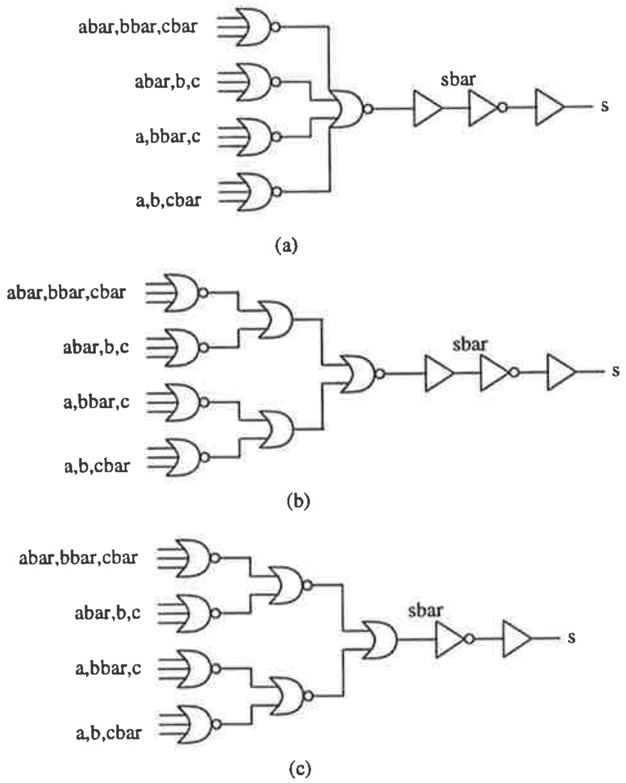


Figure 4.19: Full adder sum generation circuits.

The device–delay product for the full adder in Figure 4.17a is 24,960. The full adder in Figure 4.17b is not considered since the DCFL outputs would be unable to drive the required load. The delay for the cases in Table 4.2 is less than for Figures 4.17 and 4.18 however the area is much larger. Glitches are output events that occur before a

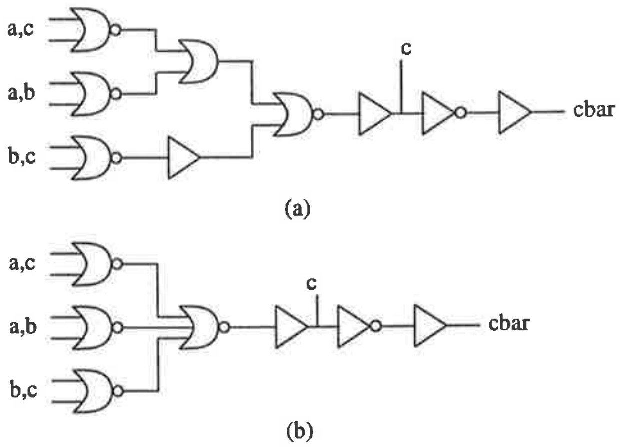


Figure 4.20: Full adder carry generation circuits.

sum/carry (Figure) (number of FETs,delay)	sum-a (4.19a) (27 FETs,481ps)	sum-b (4.19b) (31 FETs,530ps)	sum-c (4.19c) (29 FETs,432ps)
carry-a (4.20a) (23 FETs,553ps)	27,650	29,862	28,756
carry-b (4.20b) (19 FETs,503ps)	23,138	26,500	24,144

Table 4.2: Area (device)–delay(*ps*) product for various full adder implementations.

combinational circuit stabilises. In high speed circuits glitches should be avoided because they cause additional noise to be generated on power and signal buses. In the case of the full adder, glitches can arise due to unequal delay paths for the cases where the half sum is generated. The circuits summarised in Table 4.2 have almost equal delay and are glitch free if the inputs are synchronous. Another consideration is the amount of routing between cells. In all cases, three inputs and their inverse need to be provided which must be propagated between cells. A method to slightly reduce the number of devices and eliminate the need to generate the inverse of each signal is shown in Figures 4.21 and 4.22. The inputs are only $\overline{a}, \overline{b}$ and $\overline{c}$ which are used directly by the carry generation circuit to generate $\overline{Carry}$ as shown in Figure 4.21 with a delay of $370ps$ which is less than other implementations since $Carry$ does not have to be generated. The results of the first set of gates are $a.c$, $a.b$ and $b.c$ which are fed into the sum generation circuit in Figure 4.22. The result, $\overline{S}$ is calculated:

$$\overline{S} = a.b.c + a.\overline{a.c}.\overline{a.b} + b.\overline{a.b}.\overline{b.c} + c.\overline{a.c}.\overline{b.c}$$

$$= a.b.c + a.\overline{b}.\overline{c} + \overline{a}.b.\overline{c} + \overline{a}.\overline{b}.c$$

The delay through the sum path is $550ps$. The device–delay product for this implementation is 26,400 which is nearly the same as the cases in Table 4.2 except the carry path is shorter. The implementations shown in Figures 4.21 and 4.22 were used in the final design. The layout of the full adder is shown in Figure 4.23.

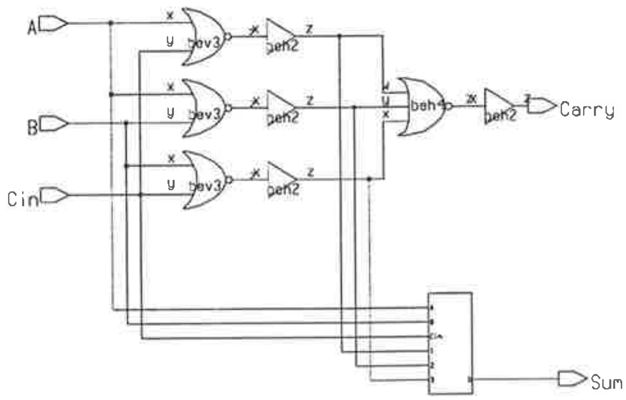


Figure 4.21: Carry generation circuit used in the final design.

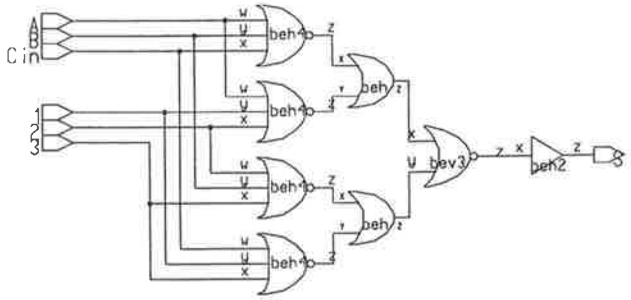


Figure 4.22: Sum generation circuit used in the final design.

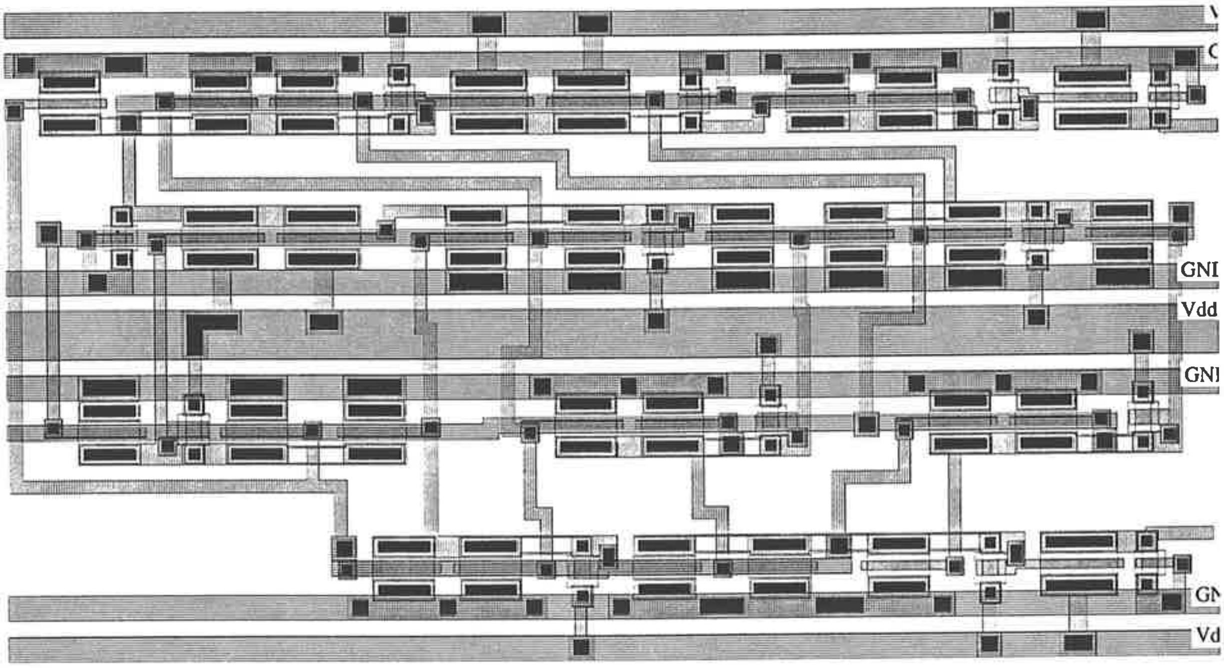


Figure 4.23: Full adder layout.

4.3.3 Systolic Cell

The systolic cell implements digit–serial arithmetic on the three operands X, Y and PP where X, Y and PP are single precision floating point numbers which have mantissa, exponent, flag and guard fields. The design of the systolic cell presented in Chapter 3, Figure 3.4 is modified to also perform the accumulation operation. In the following discussion, a subscript 'm' indicates the mantissa part and 'e' denotes the exponent part of an operand. Each number is entered least significant digit first. There are three modes of operation: multiply, add and denormalisation. These are determined by the 'c' and 'd' bits of the instruction nibble shown in Table 4.3.

INSTR[c]	INSTR[d]	Cell Operation
0	0	floating point addition
0	1	floating point multiplication
1	0	denormalisation

Table 4.3: Systolic cell instructions.

Multiplication Mode

In multiplication mode, the Y operand is the multiplicand and the X operand is the multiplier. To form partial product terms, the first nibble of the Y mantissa is stored and multiplied with each nibble of the X operand and the result is accumulated with the input partial product and output to PP_{out}. The Y mantissa is nibble–wise rotated as it passes through each systolic cell. To complete a mantissa multiplication, the mantissa must pass through the same number of systolic cells as there are nibble–digits in the mantissa. Hence, the number of systolic cells in the ring multiplied by the number of complete rotations of the operands around the ring must equal the number of mantissa nibble–digits in the mantissa. The cell function is used in two ways during floating point multiplication. During the mantissa multiplication the algorithm implemented is

$$PP_{out} = (XY + V) + PP_{in}$$

where V is the high–order digit generated by the pipelined multiplier, PP_{in} is the partial product input from the previous cell. During the exponent addition mode, the following function is implemented:

$$\begin{aligned} PP_{out} &= (X \times 1 + V) + Y \\ &= X + Y \end{aligned}$$

The value of V is zero in this part of the computation as there is no high-order output from the product $X \times 1$. This implements the exponent addition in each cell.

Addition Mode

In addition mode, the X and Y operand exponents are assumed to be equal and the function of the PE is to add up the mantissa. The Y_e (Y exponent) is directed to $PP_{out,e}$ exponent and the mantissa result is $PP_{out,m} = X_m + Y_m$. In terms of the systolic cell function, $Z = XY + PP + V$ when the X input is set to '1' and the X operand is multiplexed to PP when the mantissa is passed through the cell. Every cell performs this computation but only one cell is required to perform the mantissa addition. Since the PP_{in} to a systolic cell is blocked, all cells perform the mantissa addition and routing of Y_e but only the last cell actually provides the result PP_{out} at the output.

Denormalisation mode

The function of denormalisation is to mantissa shift the smaller of the two operands and increment the exponent. Only the systolic cell functionality is tested here. Each cell operates independently and when presented with an instruction to denormalise, the Y_m operand is shifted with respect to the X_m operand by '1' digit by bypassing a delay cell. The exponent field is incremented by '1' by using the cell function:

$$PP_{out,e} = Z = 1 \times Y + V + 1$$

One circulation of the ring shifts the Y operand by 4 digits. For full denormalisation capability an exponent subtraction would determine which operand is to be denormalised and the operands are exchanged if necessary. A difference counter would then determine how many cells the operand would need to pass through to align the mantissae. To complete a cycle of the ring any excess instructions would be NOPs. A schematic of the 4–bit digit–serial multiplier cell is shown in Figure 4.24 which incorporates the extra functions required for denormalisation and addition. The schematic for the complete systolic cell is shown in Figure 4.25 and the corresponding layout is shown in Figure 4.26. A SPICE simulation of the critical path through the digit–serial multiplier carry path

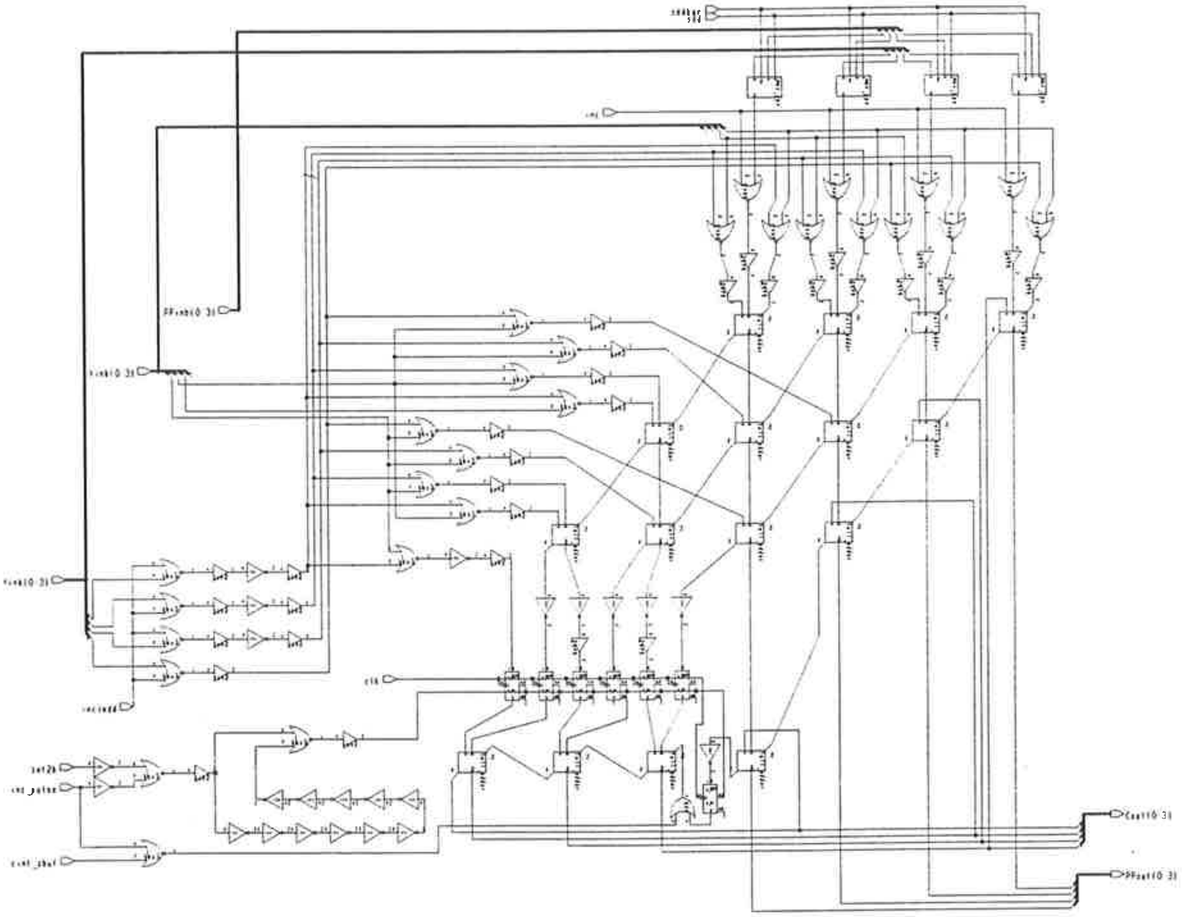


Figure 4.24: Nibble–serial multiplier schematic.

(Figure 4.27) shows a delay of $3.2ns$ which indicates that the maximum clock speed at which the processor will function correctly is greater than $300MHz$. It takes 55 clock cycles to do a multiplication–accumulation, therefore the floating point performance which can be expected from the device is approximately 11*Mflops*.

4.3.4 I/O Pads

The input and output pads interface the chip to the outside world. They have a low voltage swing of around $0.8V$ and provide a high bandwidth electrical interface to the chip. The pads used in this chip were originally from The University of California Santa

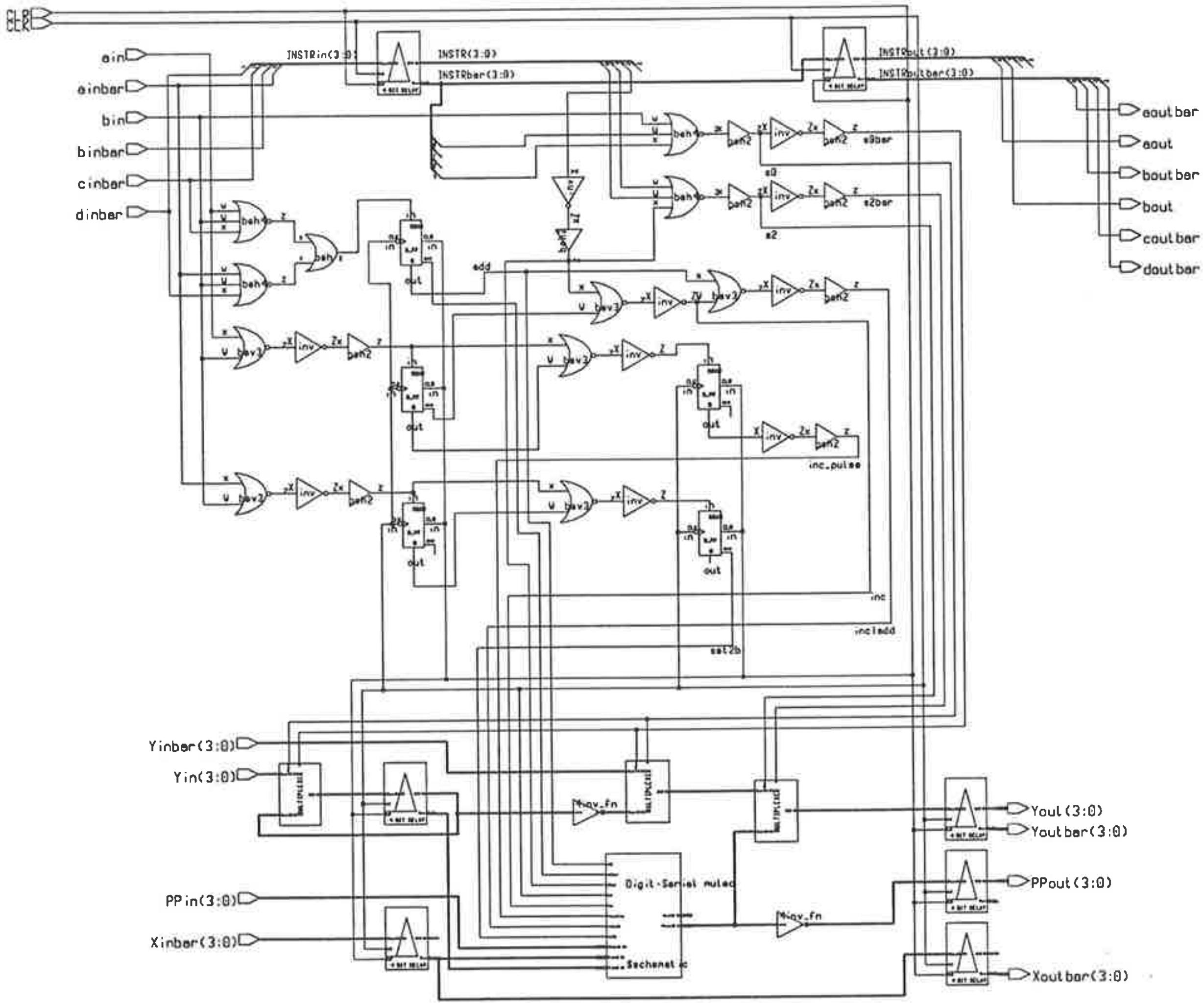


Figure 4.25: Schematic of the systolic cell.

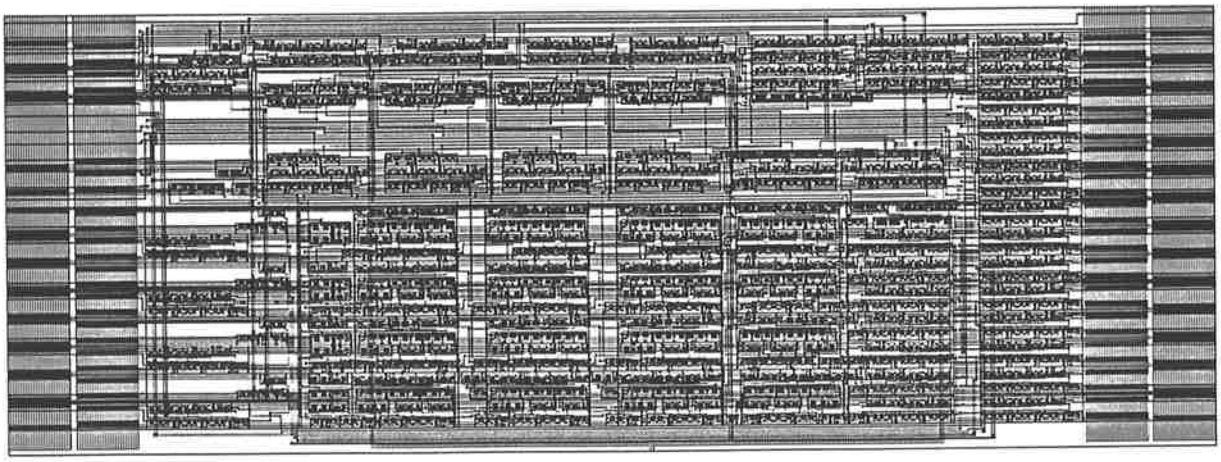


Figure 4.26: Layout of the systolic cell.

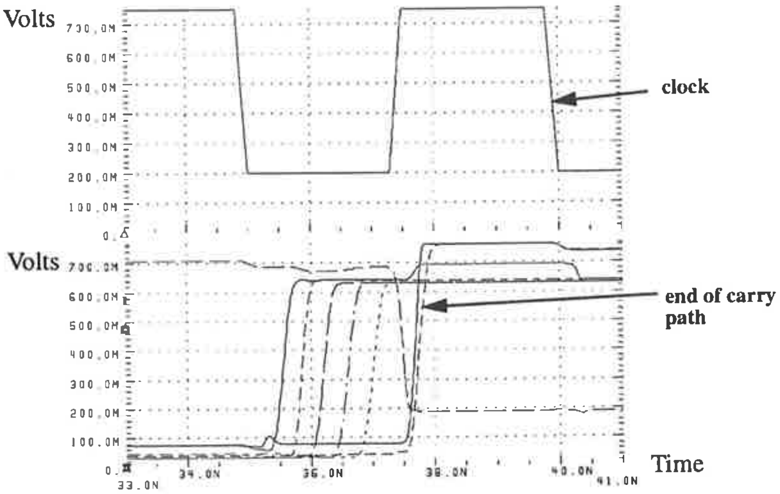


Figure 4.27: SPICE simulation of the critical path through the digit–serial multiplier.

Barbara and supplied by MOSIS. They were modified to suit our pad–ring requirements.

Input Pad

The non–inverting input pad protects the chip from ESD and allows a signal to enter onto the chip with a high bandwidth up to $500MHz$. A schematic of the input pad is shown in Figure 4.28 and the layout is shown in Figure 4.29. Figure 4.30 shows a simulation of the input pad. The pad requires an external reference voltage, *VREF*, to be supplied to the comparators in the pad. It was found $VREF = 0.7\,V$ provides a symmetrical response. On the top simulation in Figure 4.30 the signal *pad-in* is the input signal with a $1.3\,V$ amplitude and a $1ns$ rise and fall time. The three signals *out-tt*, *out-ss1* and *out-ss2* correspond to the DCFL signal on the chip for *typical*, 1σ– and 2σ–*slow* MESFET parameters, respectively. The delay through the pad ranges from 1 to $2ns$. The lower simulation in Figure 4.30 shows the current drawn from the pad supply, *I(Vddp)*, and from the input, *I(Vin)*. There is negligible current drawn in the low state but around $1.6mA$ is drawn from the supply in the high state giving a power dissipation of $3.2mW$. There is a large change in current drawn from the supply when switching between logic states.

Output Pad

A schematic of the non–inverting output pad is shown in Figure 4.31 and the correspond-

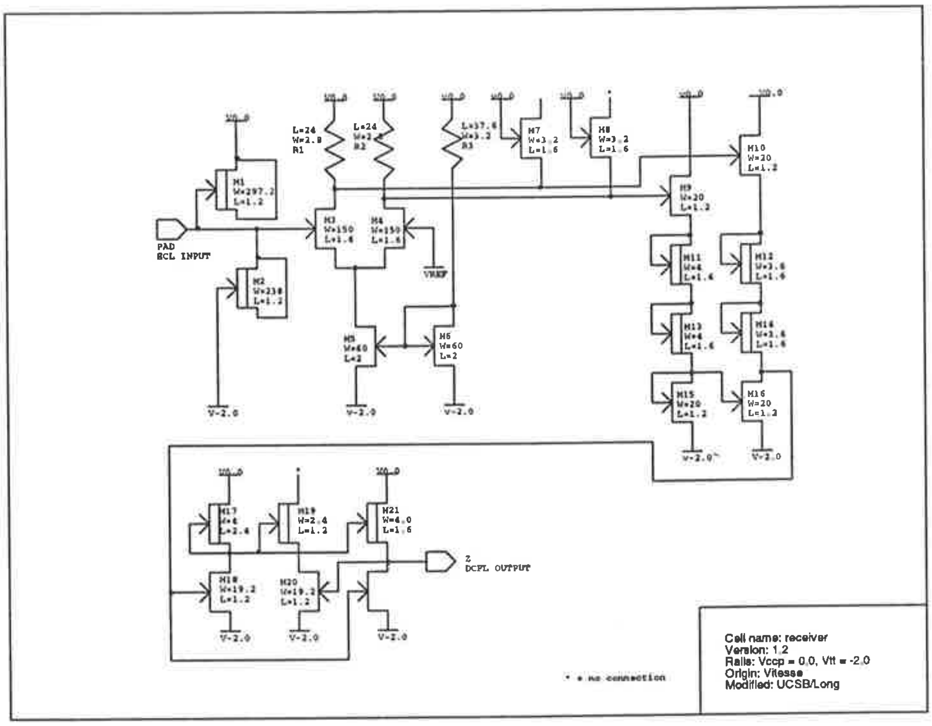


Figure 4.28: Schematic of the input pad.

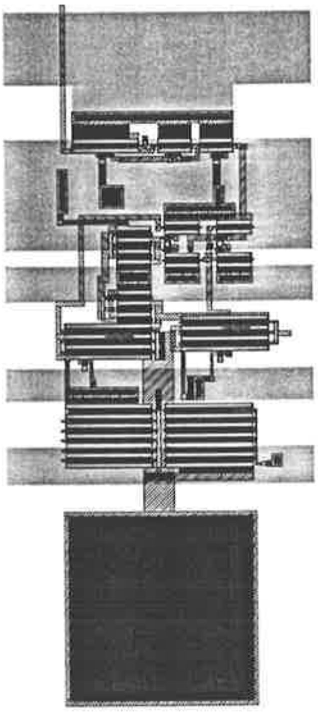


Figure 4.29: Layout of the input pad.

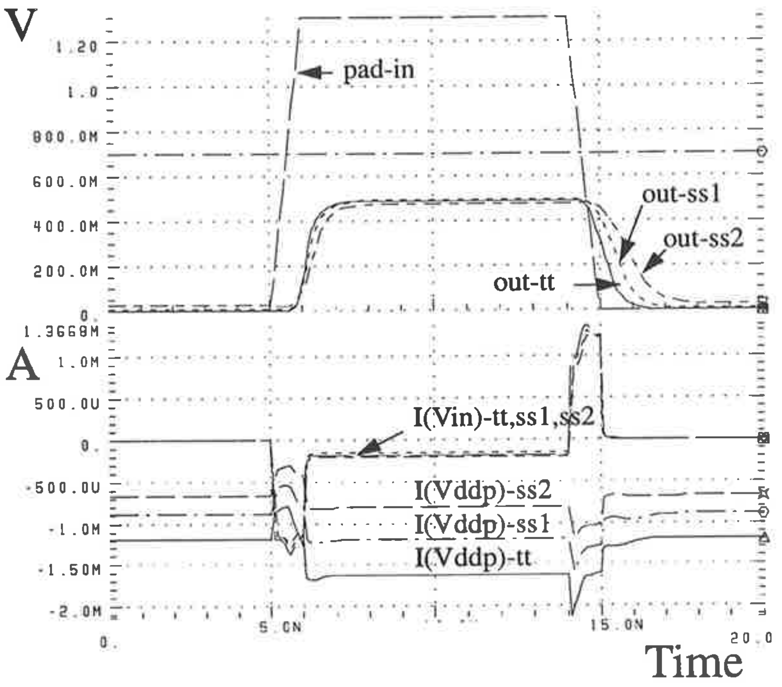


Figure 4.30: Simulation of an input pad receiver with *VREF*= 0.7*V* showing input, output response and the current drawn using *typical–typical*, *slow–slow–1* and *slow–slow–2* process parameters.

ing layout is shown in Figure 4.32. A simulation of the output driver pad is shown in

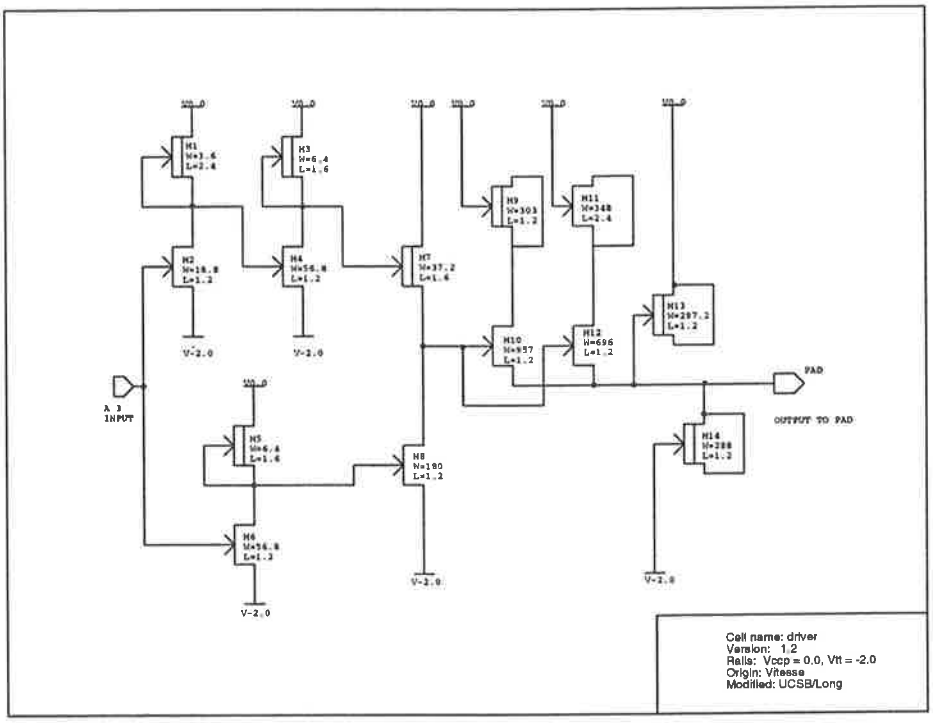


Figure 4.31: Schematic of the output pad.

Figure 4.33. The upper simulation in Figure 4.33 shows the response of the pad to a rising and falling edge on the chip (*chip-out*) with a $700mV$ amplitude and a $1ns$ rise and fall time. The output pad drives into an external 50Ω, $1pF$ load. The three output responses *pad-tt*, *pad-ss1* and *pad-ss2* correspond to *typical*, $1\sigma-$ and $2\sigma-$*slow* MESFET parameters, respectively. The delay through the pad is less than $1ns$. The lower simulation in Figure 4.33 shows that the current drawn from the pad supply, *I(VDDP)*, is negligible in the low state but around $2mA$ in the high state giving a power dissipation of $4mW$. There is a large changc in current drawn from the supply when switching between logic states as with the input pad.

4.3.5 Ring Controller

A finite state machine controller was used to control the I/O of data from the systolic ring. The input to the controller is the instruction inputs a and b in both the input (a_{ip}, b_{ip}) and inside the ring (a_{ring}, b_{ring}). The controller has eight states, $s0$-$s7$ and two

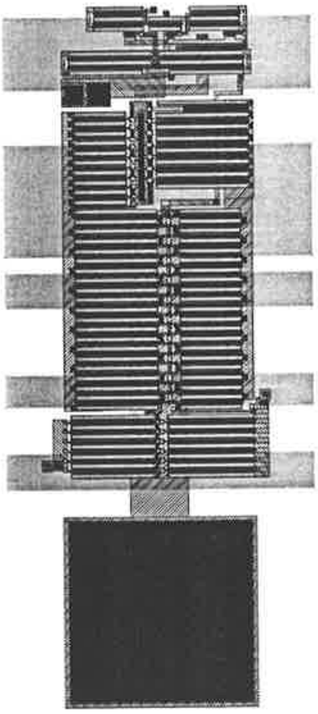


Figure 4.32: Layout of the output pad.

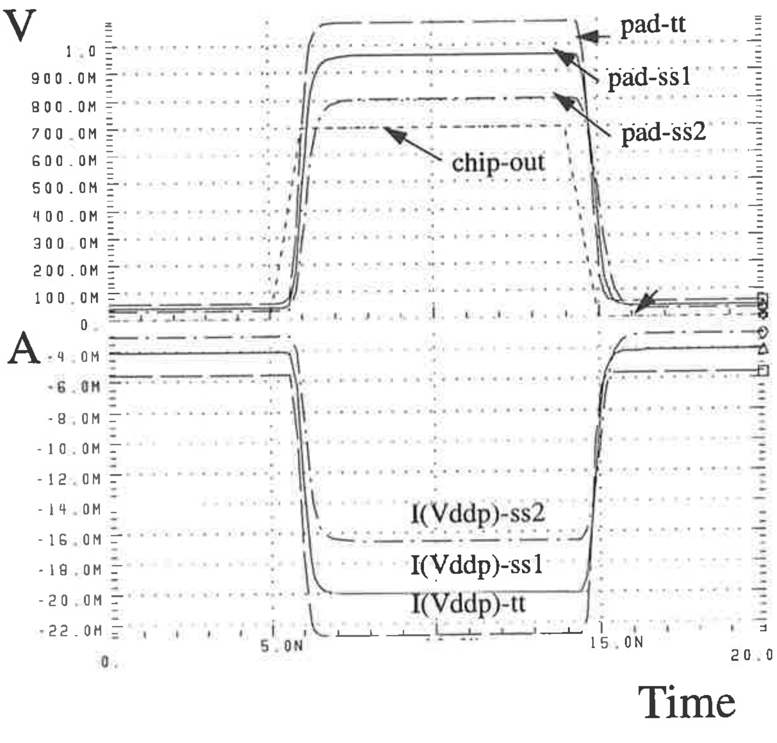
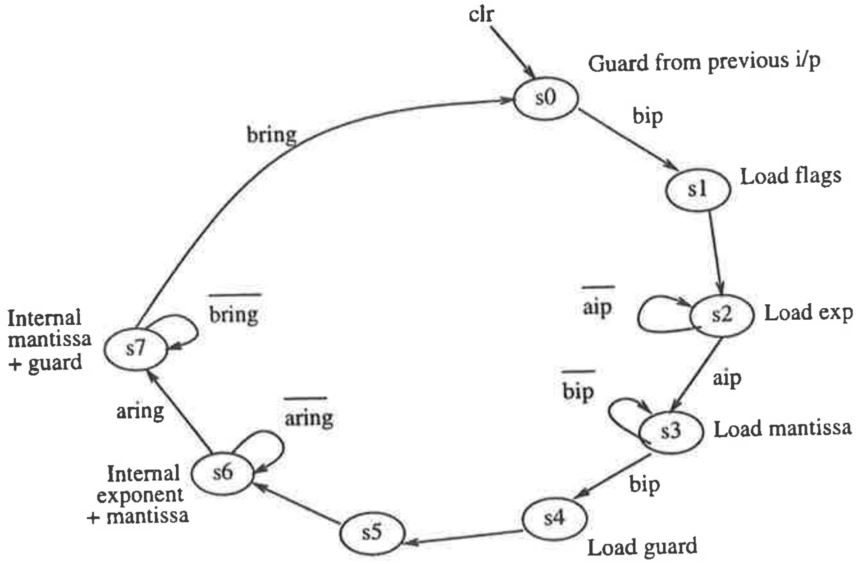


Figure 4.33: Simulation of an output pad showing the voltage response and current drawn for *typical–typical*, *slow slow-1* and *slow–slow–2* process parameters.

outputs, I_0 and I_p. The state transition diagram is shown in Figure 4.34 and I_0 opens the I/O multiplexer for the ring to feed the operands into the ring. Once the operands are loaded, the ring is closed and one more circulation of the operands is carried out to complete the computation. The ring is then opened and the operands and result are output at the same time the new operands are loaded (i.e. the cycle is repeated). The input/output multiplexer to the ring is controlled by the I_0 signal. Currently, I_p is not used. The states $s1$, $s2$ and $s3$ are used by the flag generation circuit to generate the flag bits. The schematic diagram of the controller is shown in Figure 4.35 and a functional simulation using IRSIM is shown in Figure 4.36 which verifies the operation. The outputs are multiplexed between the first cell and the ring by the multiplexer under the control of the ring controller. The schematic of a 4-bit multiplexer is shown in Figure 4.37a. Four of these are used to build the I/O multiplexer shown in Figure 4.37b.



Outputs: $Io = Io.\overline{s0}.\overline{s5} + s0$

$Ip = Ip.\overline{s4}.\overline{s0} + s0$

Figure 4.34: State transition diagram for the ring controller.

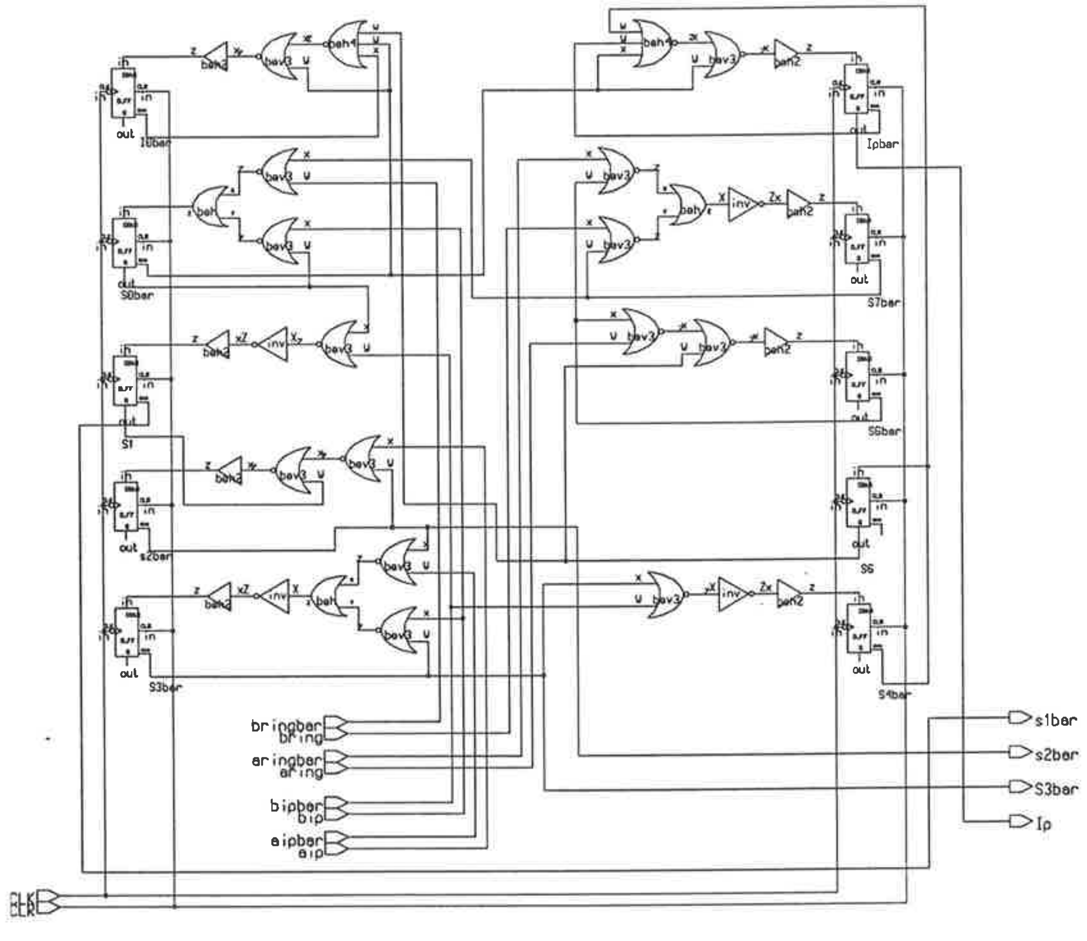


Figure 4.35: Circuit schematic of the ring controller.

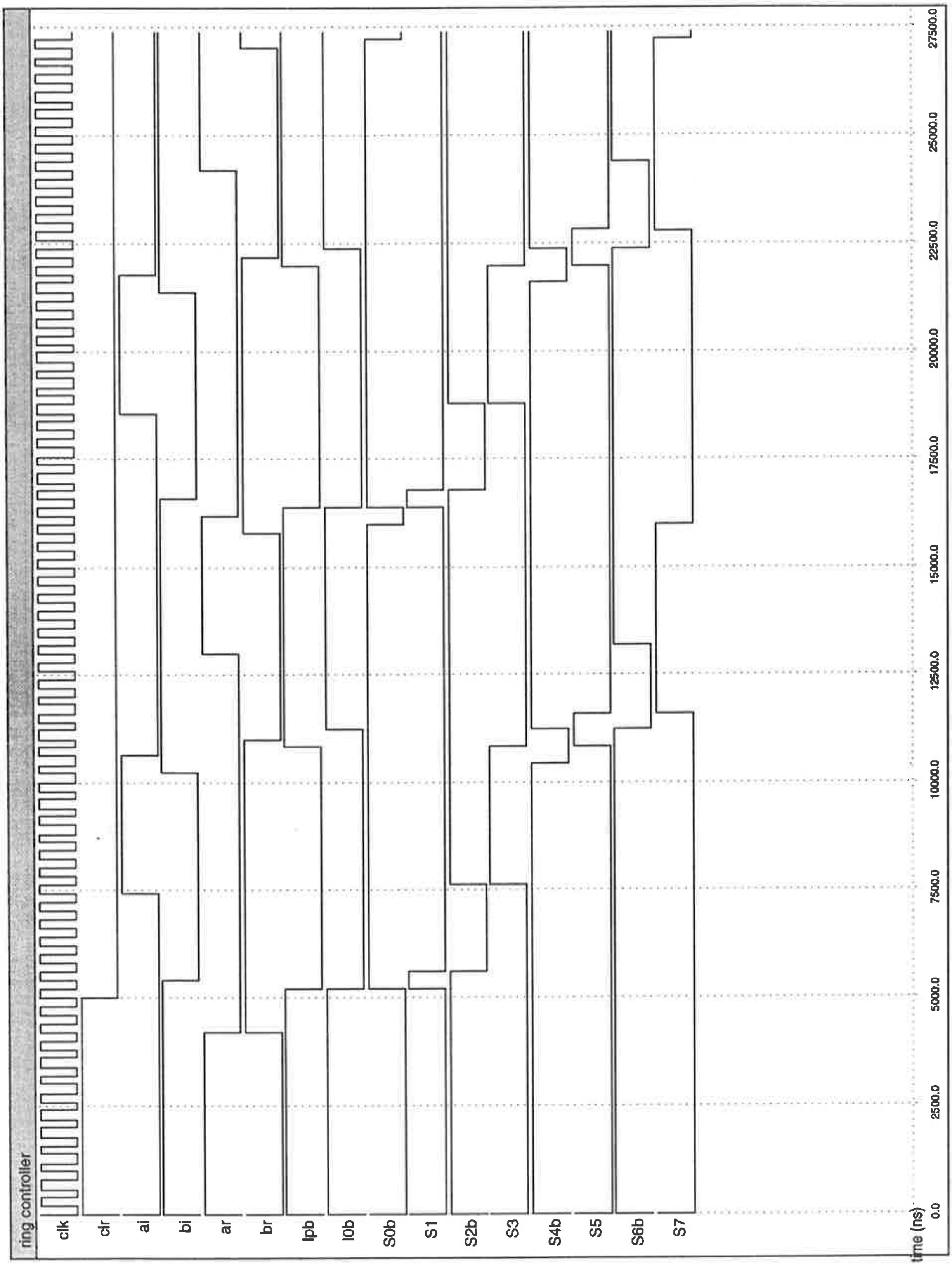


Figure 4.36: Functional simulation of the ring controller using IRSIM.

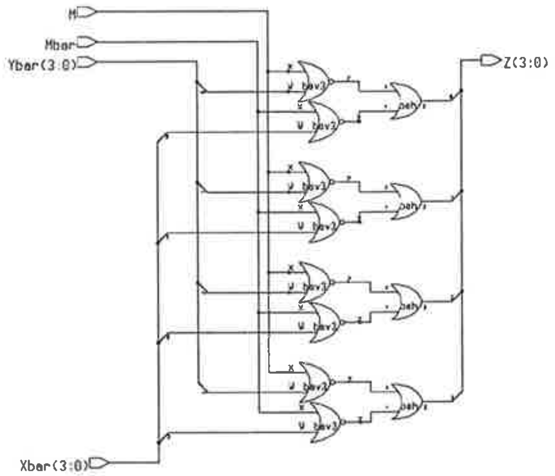


Figure 4.37a: Schematic of a 4–bit multiplexer.

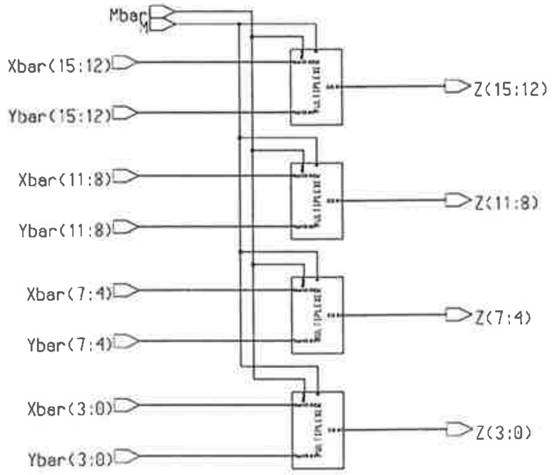


Figure 4.37b: Schematic of the I/O multiplexer.

4.3.6 Flag Checking

The flag field for both X and Y input operands and the result P_{out} have the specification shown in Table 4.4. Z is the zero bit which is set if the X or Y operand mantissa is zero

flag bit	0	1	2	3
X_{in}	-	-	-	$X_{m,sign}$
Y_{in}	-	-	-	$Y_{m,sign}$
P_{out}	-	-	Z	$X_{m,sign} \oplus Y_{m,sign}$

Table 4.4: Specification of input and output operands flag nibble.

and indicates the result, P_{out} is zero. $X_{m,sign}$ and $Y_{m,sign}$ are the sign bits for the mantissa of the X and Y operands, respectively. These are fed into an EX–OR function to form the sign result for the multiplication. The circuit schematic of the flag generation circuit is shown in Figure 4.38 and the signals $s1$, $s2$ and $s3$ from the ring controller are used to define when the flag checking circuit is operational. The resulting layout is shown in Figure 4.39.

4.3.7 Clock Generation

A two speed single–phase clock is generated on–chip using a DCFL ring oscillator which can run at either 1*GHz* or 600*MHz*. To provide a range of possible clock speeds for both low speed functional testing, high speed performance verification, and process characterisation the output from the two speed oscillator is divided by two additional modulo–4 counters. External signals are used to multiplex between the direct and derived clocks as well as an external clock source. The internally generated clock frequencies are 37.5, 62.5, 150, 250, 600 and 1000*MHz*. There is no constraint on the external clock frequency and can therefore be used for DC testing. Figure 4.40 shows the clock architecture, Figure 4.41 shows the layout and Table 4.6 shows the clock control signals. The signal *RST* is the active high ring oscillator reset signal and the rate signal determines the path through the ring as shown in Table 4.5. *CLKpad* is the external clock input. A SPICE simulation of the oscillator is shown in Figure 4.42.

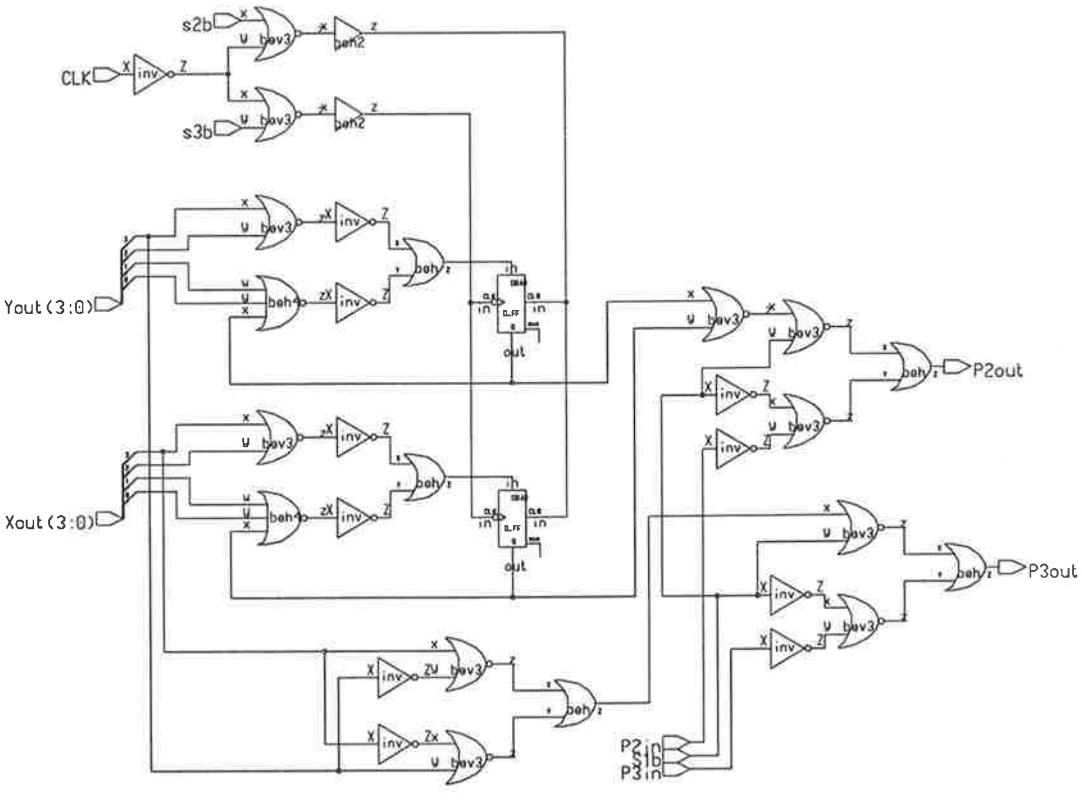


Figure 4.38: Schematic of the flag generation circuit.

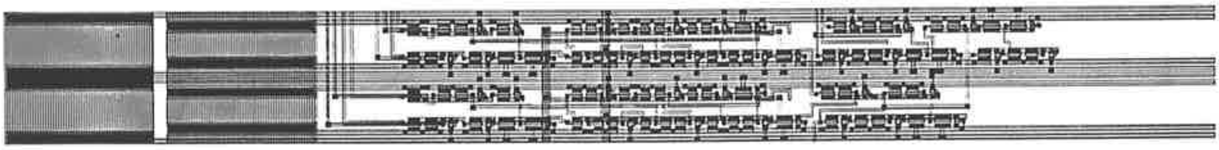


Figure 4.39: Layout of the flag generation circuit.

rate	DCFL inverter	DCFL NOR	SDCFL NOR	total logic depth
0	11	1	1	13
1	5	1	1	7

Table 4.5: Logic length for the two ring oscillator configurations.

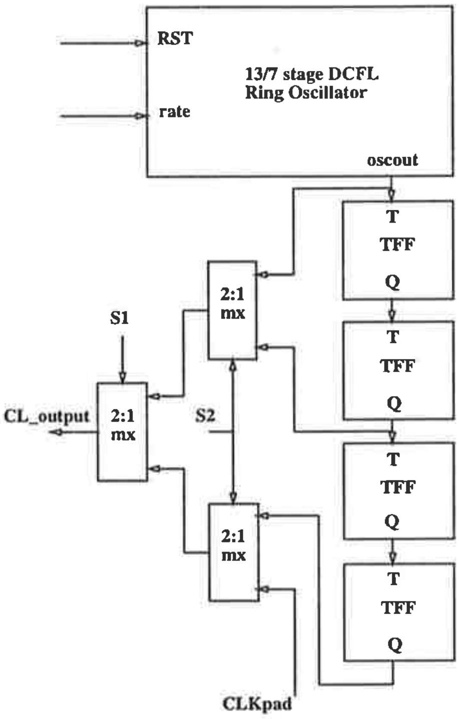


Figure 4.40: Clock architecture.

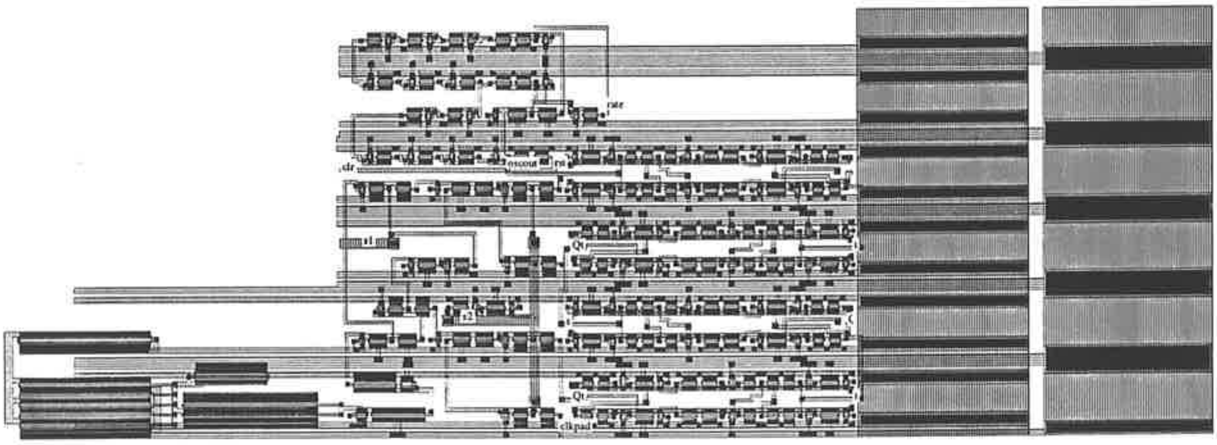


Figure 4.41: Clock generator layout.

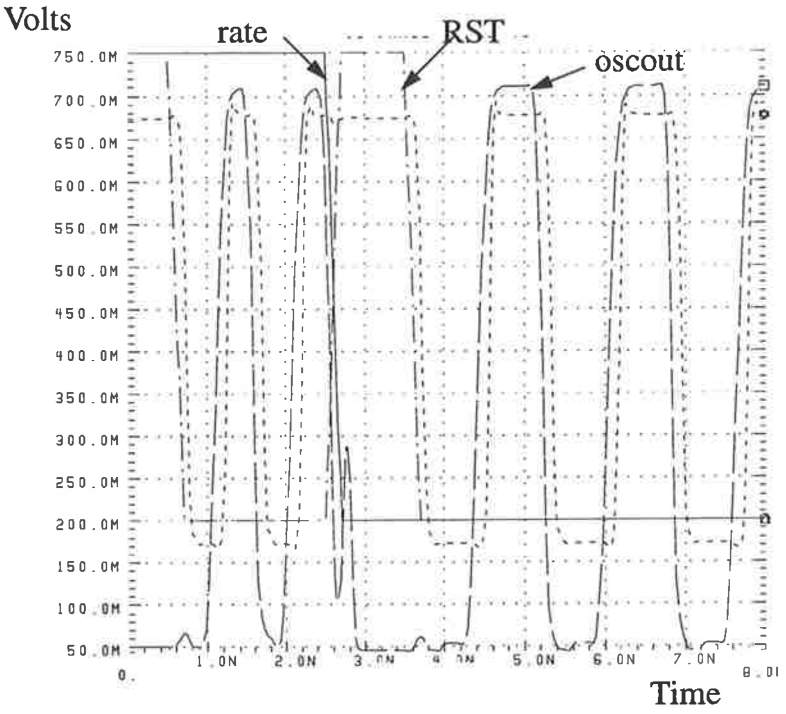


Figure 4.42: Clock generator transient simulation for two clock rates.

CL_output (*MHz*)	*S1*	*S2*	*rate*	RST
1000	0	0	1	0
250	0	1	1	0
CLKpad	1	0	1	0
62.5	1	1	1	0
600	0	0	0	0
150	0	1	0	0
CLKpad	1	0	0	0
37.5	1	1	0	0
0	-	-	-	1

Table 4.6: Clock control signals.

4.3.8 Clock Distribution

There is a significant overhead in generating and distributing a synchronous clock across a VLSI system [Come92]. A global synchronous clock scheme could be used for small arrays, but for large arrays clock synchronisation in a fully synchronous system will not work if significant clock skew exists between iso–synchronous zones. To overcome this problem, techniques such as clock frequency multiplication in each chip from a global lower rate clock and balancing skew using H–Trees [Bako90] may be used. As the order of the array increases, the communication between the PEs would need to be asynchronous, but each PE would have its own synchronous clock which may be derived from a global or locally generated clock. The PE is characterised by a ring of registers some with a delay path between adjacent registers and some without delay. A set of clocked data latches $\{L\}$ can be described in terms of the following timing characteristics:

- T_{cl} is the clock period
- T_h is the hold time
- T_s is the set–up time
- T_p is the propagation delay from the clock to the output
- T_i is the time at which latch L_i is clocked

The condition under which data is transferred correctly between the adjacent latches, L_i and L_{i+1} in a linear array is

$$-T_{cl} + T_p + T_s < \Delta T_i < T_p - T_h$$

where the clock skew $\Delta T_i = T_{i+1} - T_i$. In an ideal synchronous system there would not be any clock skew, that is:

$$\Delta T_i = 0 \ \ \forall i$$

Consider a ring structure of N latch elements shown schematically in Figure 4.43 for the case of $N = 8$. Extending the definition of a linear array to a ring, the skew, ΔT_i between two adjacent L_i and $L_{(i+1) \ mod \ N}$ latches is:

$$\Delta T_i = T_{[(i+1) \ mod \ N]} \ - \ T_i$$

where $1 \leq i \leq N$. For any synchronously clocked ring, whether it is ideal or non–ideal,

$$\sum_{i=1}^{N} \Delta T_i = 0$$

The sum of the clock skews in a closed ring is zero. Figure 4.43 shows an example of an imperfectly clocked ring from which it can be seen that any positive clock skew must be matched by an equivalent negative skew. Clock signal between latches 3 and 4 has a positive clock skew while the clock signal between latches 4 and 5 has a negative clock skew (Figure 4.43). Clock signal between latches 1 and 2 has no clock skew. It is also apparent that there is no upper limit to total skew across j elements provided that the constraint on each ΔT_i is satisfied. The simple design constraint which was adopted to guarantee correct operation of the ring was that *any* two latches in the ring should be able to communicate with each other. Hence, for latches L_i and L_j, $\Delta T_{i,j}$ has an upper bound given by:

$$\Delta T_{i,j} < T_p - T_h, \quad \forall i, j$$

For the logic family used, $T_p = 360ps$, $T_h = 120ps$ and hence $\Delta T_{i,j} < 240ps$.

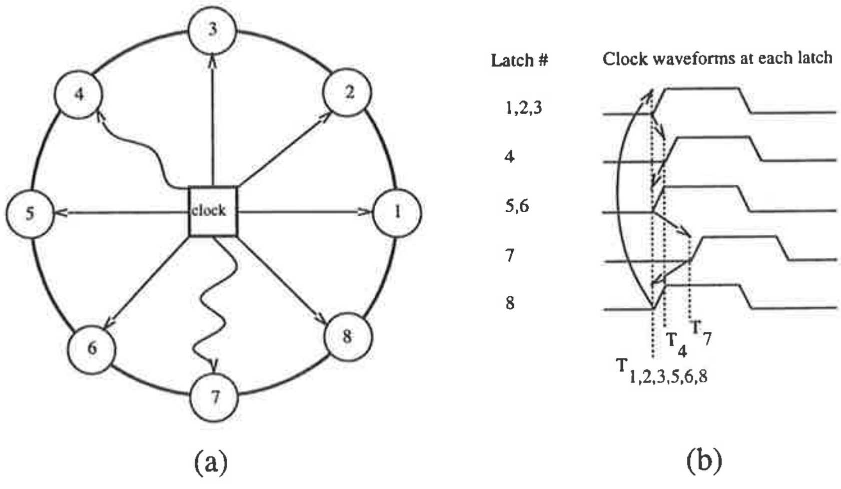


Figure 4.43: (a) A clocked ring where the arrows indicate the delay from the clock generator to the latch, (b) Clock timing where arrows indicate clock skew between adjacent latches.

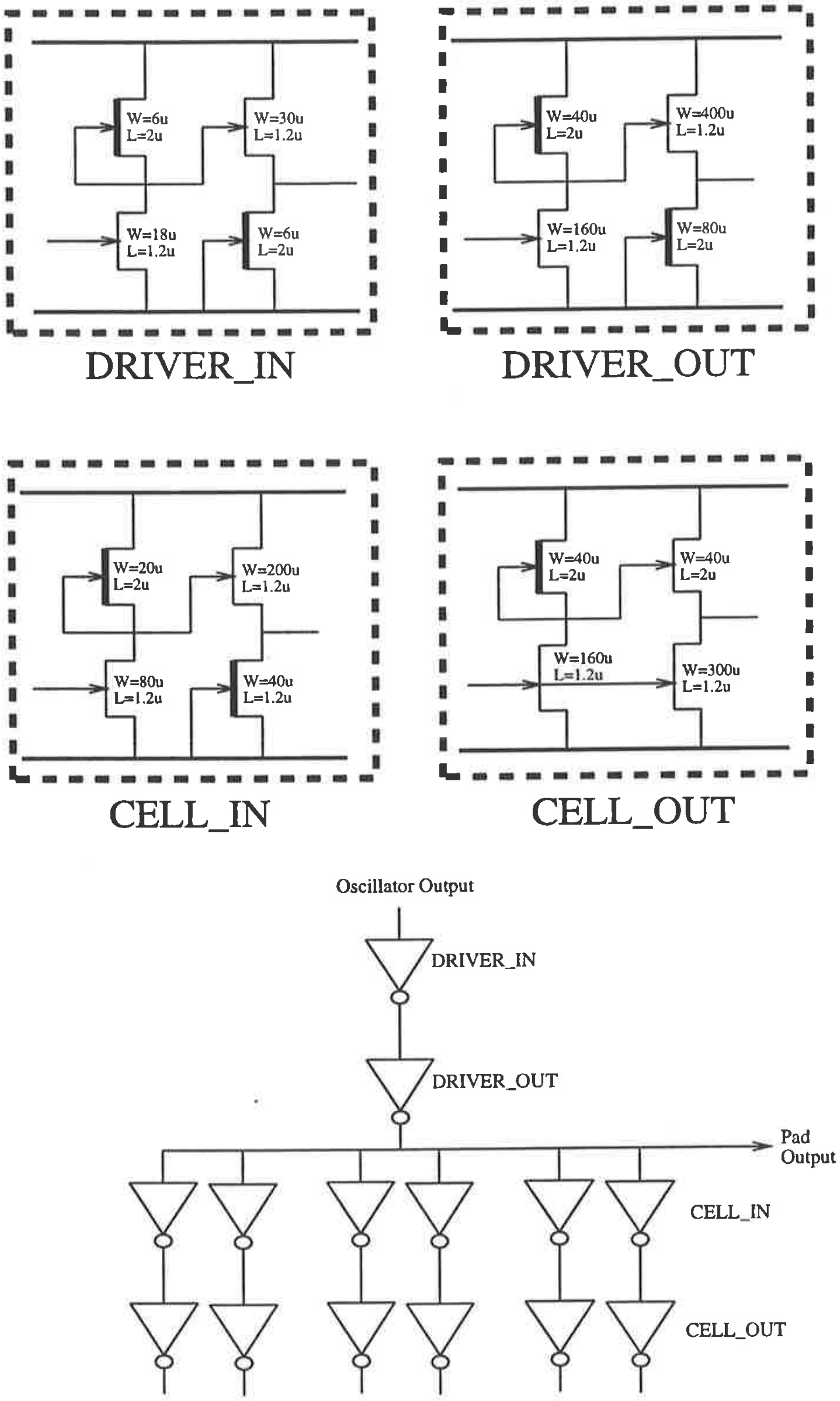


Figure 4.44: Two stages of super buffers used to drive the clock tree.

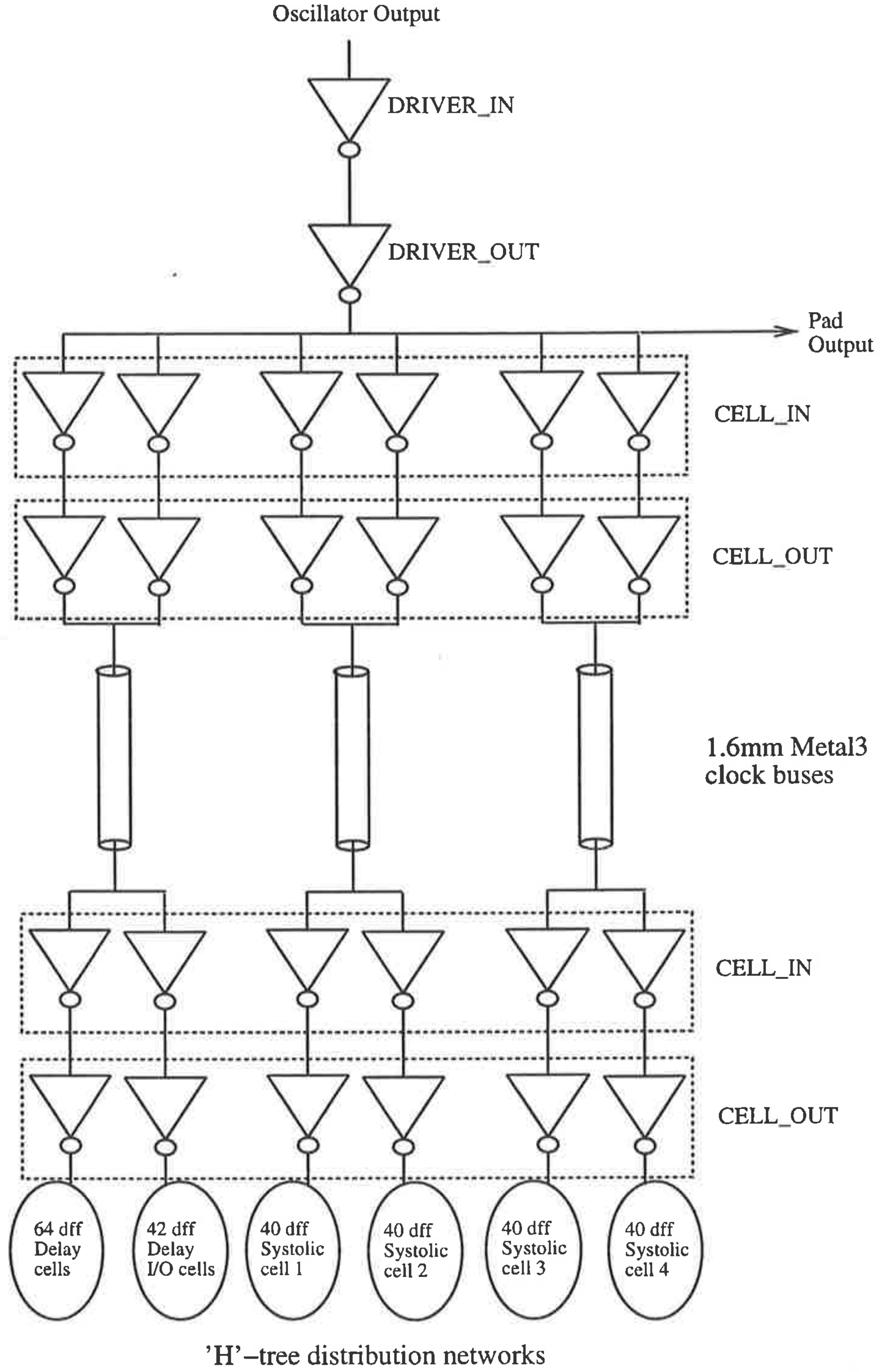


Figure 4.45: Schematic of the simulated clock distribution network.

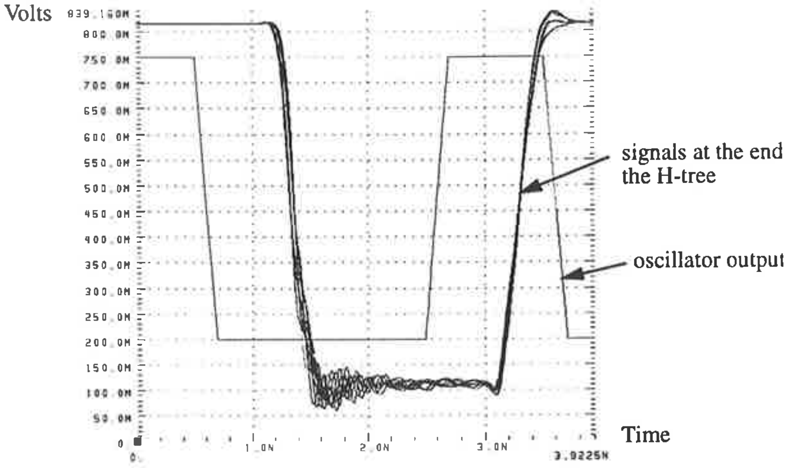


Figure 4.46: SPICE simulation of the clock distribution across the chip.

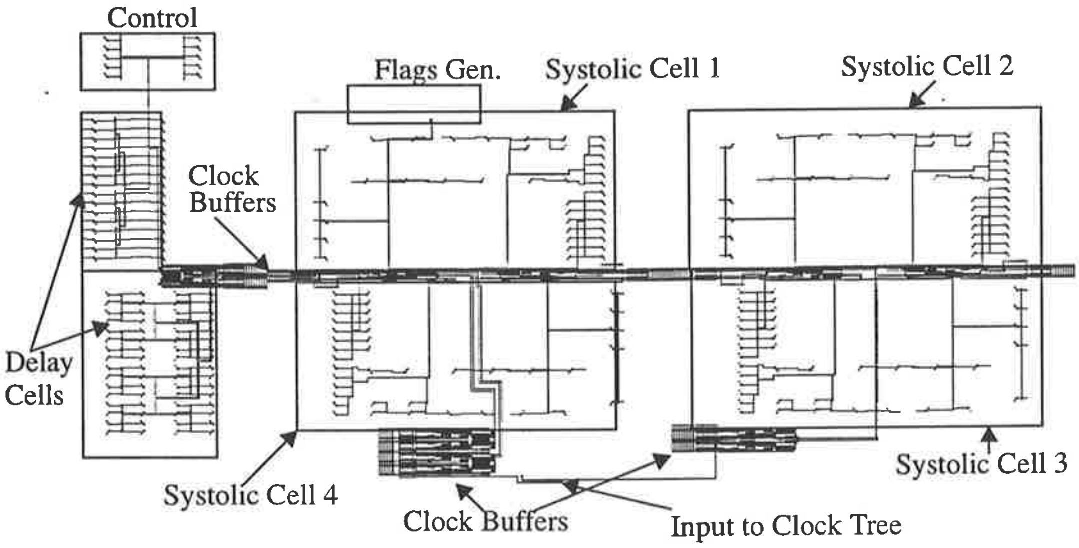


Figure 4.47: Layout of the clock distribution circuit.

Equal length clock lines using an H–tree layout style were used to distribute the clock to the 266 flip–flops (532 EFET loads) in the chip. Figure 4.47 shows the layout of the clock distribution system including buffers. A two stage system of super buffers was simulated and optimised to drive the long buses in the clock tree. Figures 4.44 and 4.45 show the design of the first and second stages of the super buffers. Transmission line models were also used to simulate the clock system and Figure 4.46 shows a SPICE simulation of the clock waveforms at the leaves of the H–tree. The simulation shows that the clock skew, $\Delta T_{i,j}$ between any two latches has been controlled to within $100ps$ to ensure correct latch operation.

4.3.9 Systolic Ring

The complete processing element is a systolic ring consisting of three elements and a ring controller. The first is an I/O logic element in which I/O and logical operations are performed. The second is a systolic cell which implements two distinct recurrence relations upon operands circulating in the ring. The selection of the appropriate recurrences to be applied at a given time is determined by an instruction nibble, *INSTR*, which circulates with the operands and includes **mode** which was defined previously. The third element of the ring is a delay cell. The number of delay cells in a ring is chosen so that the length of the ring is equal to the length of the operands. The mantissa length is 32–bits and $r =$ 4–bits per digit. Therefore, there are eight mantissa digits and if there are four systolic cells in the ring, two circulations of the operands is required. The schematic of the PE is shown in Figure 4.48 which includes the controller (Figure 4.35), the flag generation circuit (Figure 4.38), four systolic cells (Figure 4.25) and six 16–bit delay cells (Figure 4.50) which are built from the 4–bit delay elements shown in Figure 4.49.

4.3.10 Floorplan

The floorplan of the overall chip is shown in Figure 4.51. The ring structure of the processing element is in the centre of the chip with the I/O on one side of the chip and test structures on the other. Some basic circuit elements were placed on the chip as test structures in the event that the PE did not work.

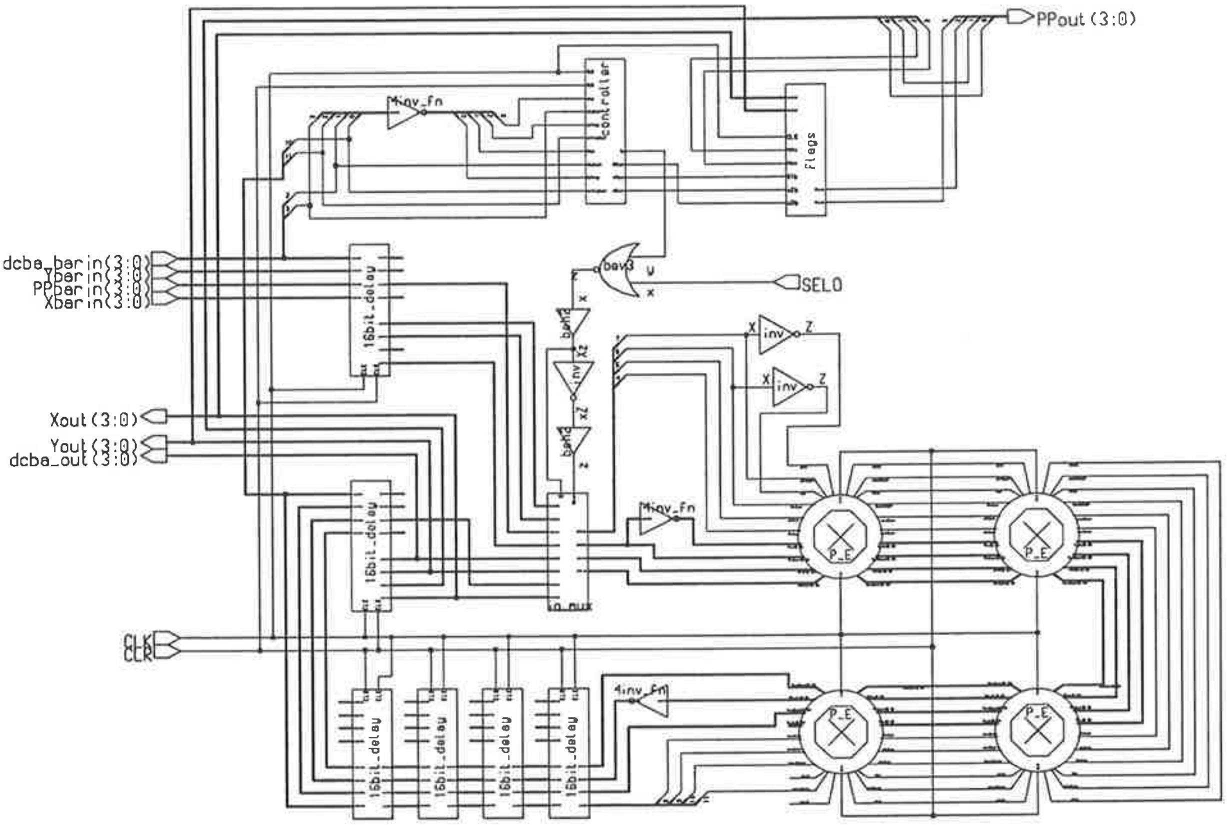


Figure 4.48: Schematic of the systolic ring processing element.

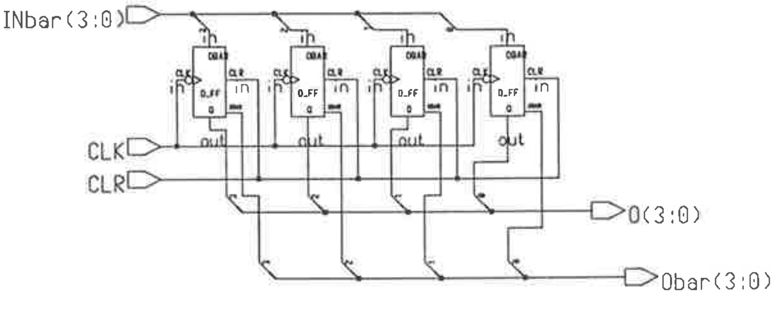


Figure 4.49: A 4-bit delay element.

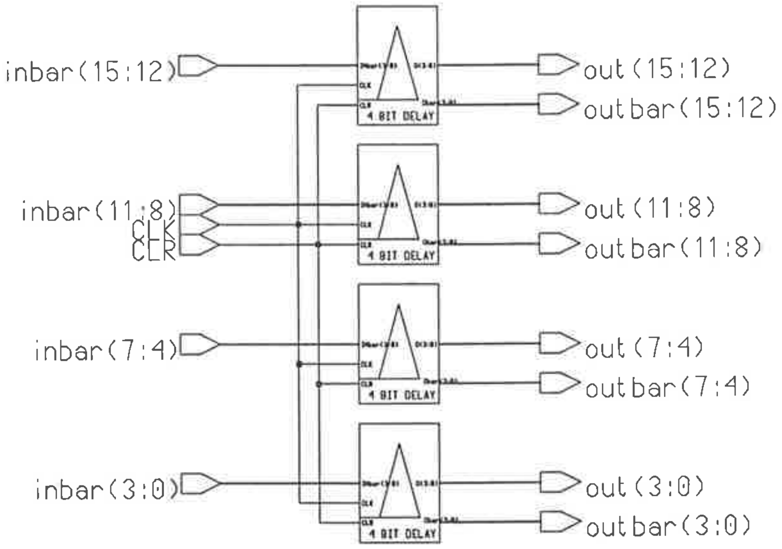


Figure 4.50: A 16–bit delay element used in the systolic ring.

4.3.11 Power Circuit

The design specification for the power distribution circuit adopted was that the voltage should not drop by more than 5% due to resistive effects across the chip. The design of the power distribution circuit took into consideration the maximum allowable current density, the self inductance and maximised the supply to ground capacitance of the power buses. Simulated total power dissipation of the PE chip is $2.2\,W$ with a $2\,V$ supply. The power is dissipated using an 132–pin Multi–Layer Ceramic (MLC) package with a finned heat-sink. Buffered DCFL is the major logic class used. It is a normally–on class in which there is little dynamic power dissipation. The voltage swing is small ($0.6\,V$) and the operation of a DCFL gate is to switch current from the pull–down FET in the output logic low state to the forward biased Schottky diode on gate of the load device. Ring notation places gate structures in a local area so the change in current into this local area is small.

Clock Distribution

The clock distribution and the power rails for the logic circuits were separated to minimise noise coupling. The peak current in the clock circuit for the feed wires is $20mA$. The current density was set to $2mA/\mu m$ and so the widths of the metal–3 feed wires were calculated accordingly.

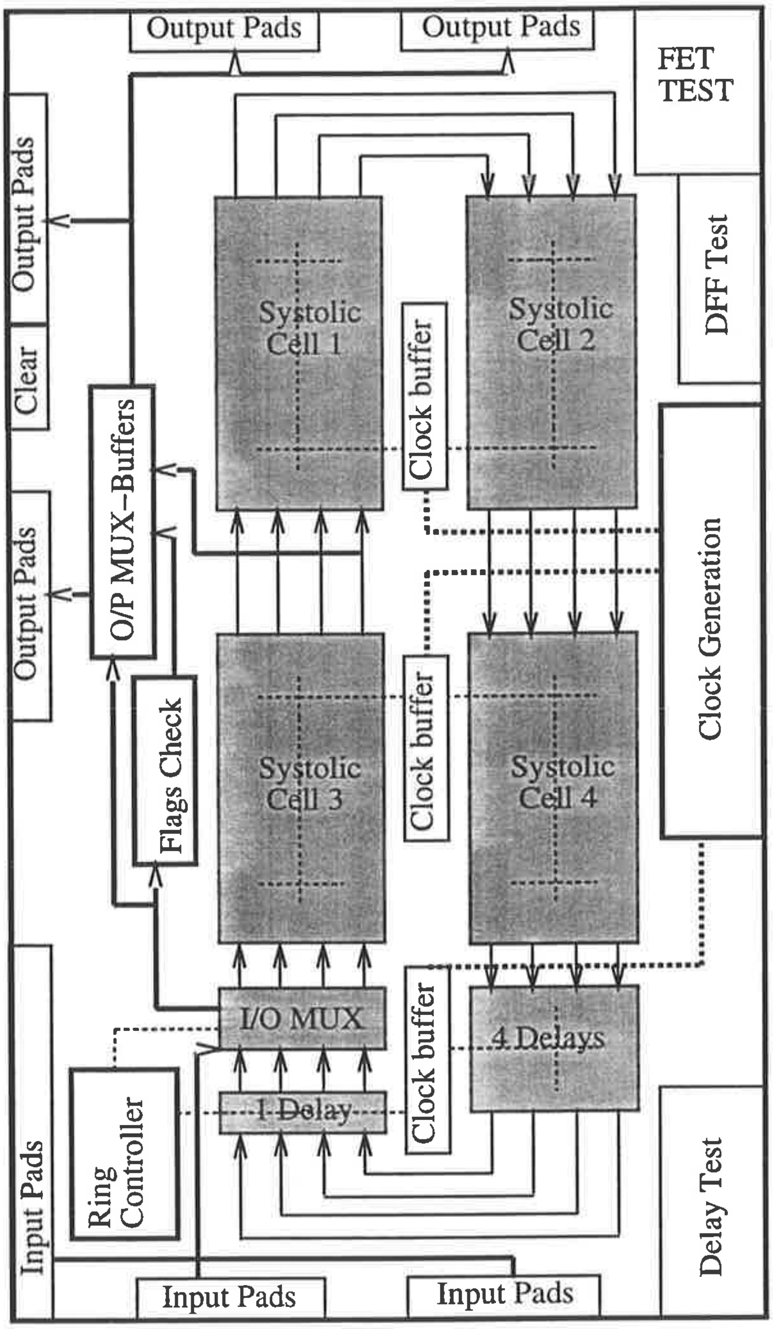


Figure 4.51: Floorplan of the processing element chip including test structures.

Current Density Limits in Power Buses

Each data flip–flop dissipates around $1mW$ of power and a full adder requires $2mW$. For each power bus there are four full adders and four latches which require $12mW$ or $6mA$ of current. Power supply rails fed from each end have a maximum current density at the ends of $3mA$. A systolic cell draws $100mA$ of current and is double–end fed so the $240\mu m$ wide power bus has a peak of $50mA$ passing through it to give a current density of $0.5mA/\mu m$. This is 20% of the metal–3 current density limit of $2.8mA/\mu m$. The twenty–four metal–2 buses which take the current to the circuits from the metal–3 buses are each $3\mu m$ wide to produce a maximum current density of $0.66mA/\mu m$. A gold bond wire is $25\mu m$ in diameter which produces a cross sectional area of $491\mu m^2$. The maximum current density for gold is $J_{max,Au} = 6 \times 10^5 A/cm^2$ and therefore a bond wire can only carry $2.95A$. A $160\mu m$ wide pad made from metal–3 has a current limit of $179mA$, therefore the maximum current through a pad is limited by the metal–3 connection from the pad. The input pad draws $3mA$ and the output pads draw $25mA$ of current and therefore, one set of Vdd and GND pads should only supply four output pads.

Inductance Limit

Consider the inductance and subsequent voltage induced in a metal–2 power line running through a systolic cell. For two parallel metal–2 wires with a pitch of $2\mu m$ and a width of $1\mu m$, the mutual inductance is given by [LoBu89]:

$$L = \frac{120\pi}{c} K$$

where $K = 0.33$ is an elliptic integral function which depends on the width and pitch of the wires and c is the speed of light. In this case $L = 8nH/cm$. Note that these wires are thinner than actual metal–2 power distribution wires and the inductance is overestimated. The voltage induced between the wires is given by:

$$\Delta V = L \frac{\Delta I}{\Delta T}$$

The current change in the power supply for a single data flip-flop is approximately $0.4mA$ as seen in Figure 4.52. A single Vdd bus may supply ten data flip–flops where, in the worst case, they all switch to the same state at the same instant and the current change

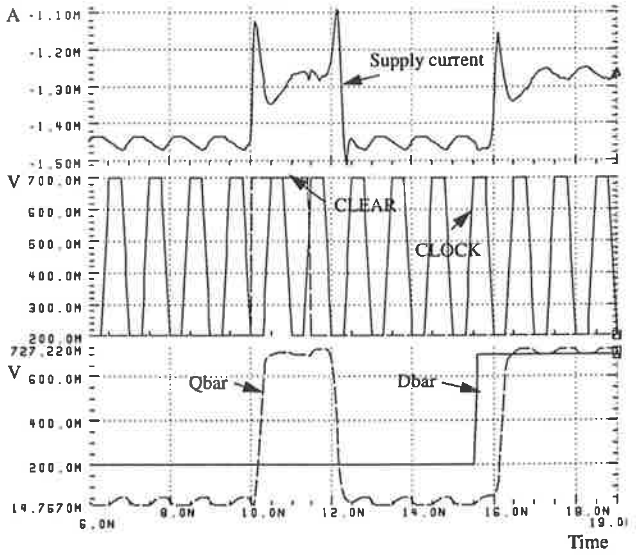


Figure 4.52: Data flip–flop simulation showing supply current.

will be $4mA$. For a $1mm$ long Vdd bus and a $4mA$ current ramp for a typical rise time of $100ps$, $\Delta V = 32mV$. This is less than our design limit in Chapter 3 which was 5% of $2V$ ($100mV$).

4.4 Fabrication and Packaging

The chip was successfully fabricated by Thomson–CSF Semiconducteurs Specifiques, France using the HGAAS–II process licenced from Vitesse Semiconductor Inc., USA on their first fabrication run with this process. The chip was fabricated in a Gallium Arsenide $0.8\mu m$ E/D MESFET process. The total chip size including pads and test structures is $3.1mm \times 5.8mm$ and includes 16,000 devices. The dimensions of the PE by itself are $1.7mm \times 4.5mm$ giving an active area of $7.5mm^2$ with 12,000 devices resulting in a device density of 1600 $FETs/mm^2$. The chip was bonded into a 132/84 pin MLC package supplied by TriQuint Semiconductor [TriQ91]. This is a high speed package which supports the special requirements of very high performance ICs. There are 84 signal lines and two power supplys with internal decoupling capacitors between the internal power and ground planes which minimise switching noise on the power supplies. Signals are carried on 50Ω controlled impedance transmission lines between the package leads and the cavity bond pads. The package can easily handle power dissipation of $1.5W$ and up to 4 to 5W with a finned heat–sink. The package is mounted upside–down to allow the heat–sink to be attached to the back of the package. The gull wing leads make it suitable for surface mounting or contact mounting using an elastomer ring. The delay of signals through the package range from 70ps to 110ps. Appendix B shows the pin assignment to the package. A micrograph of the fabricated GaAs systolic PE chip after packaging is shown in Figure 4.53 with the floorplan overlaid.

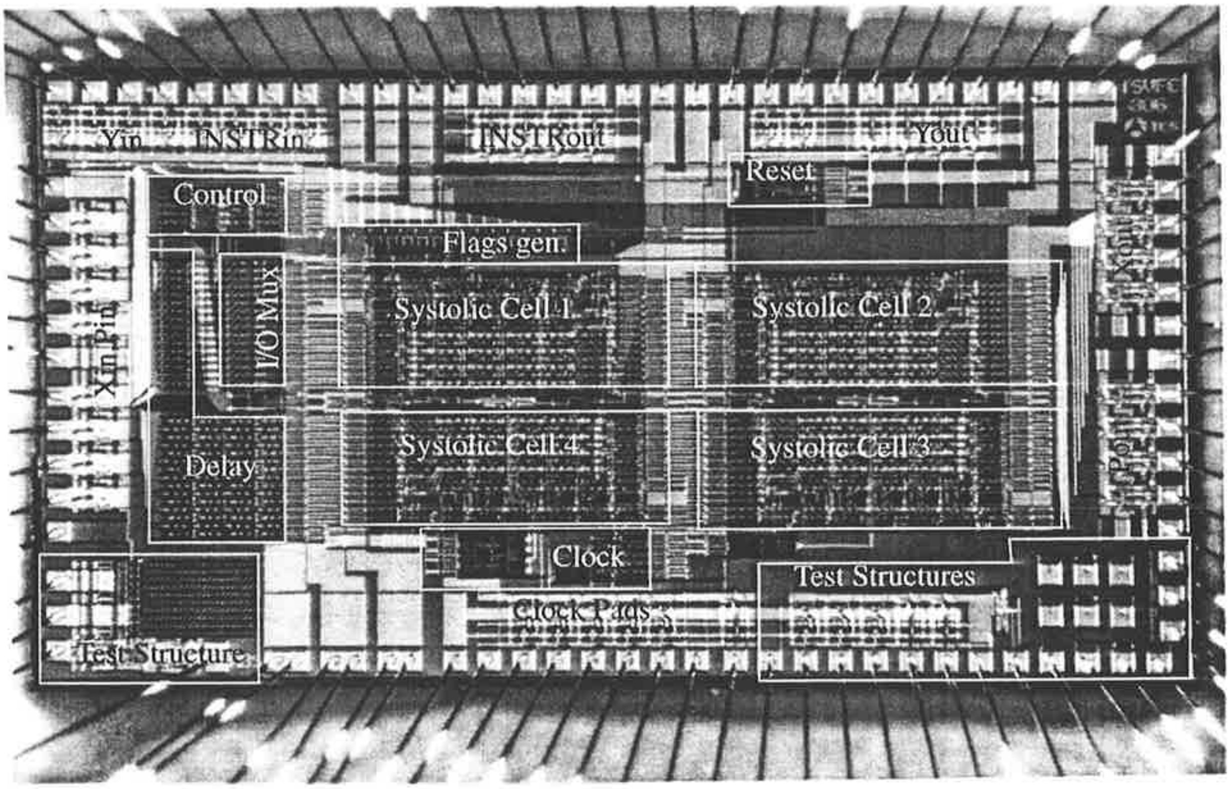


Figure 4.53: Micrograph of the fabricated GaAs systolic PE chip with the floorplan overlaid.

Chapter 5

Testing the Processing Element Chip

This chapter presents the test environment, test procedure and results of testing the PE chip. A test jig was designed and constructed to mount the PE chip and interface it to a digital tester and other test equipment. The wires from the test jig to the probe tips of the digital tester must propagate digital signals at a $300MHz$ rate without significant distortion. The discrete enhancement and depletion mode MESFETs are characterised. A digital tester was used to test the functionality of the PE.

5.1 Test Fixture

A custom designed test fixture (test jig) provides a platform to quickly test the packaged chips. The 132/84 pin MLC package is surface mounted to the PCB with an elastomer ring (pressure contact) to allow easy mounting and unmounting of test chips. The footprint is the same as that which would be used for solder reflow assembly. The board is designed to interface to the Tektronix DAS–9200 digital tester. The connections are standard gold PCB pins with a 0.1" (inch) pitch. A thin low dielectric double sided Teflon PCB provides controlled impedance lines to the test interface pins. Solder pads are provided to connect chip resistors or capacitors to either end of the board trace.
The test fixture provides:

- a precise environment for high speed digital circuits
- easy mounting/unmounting of packaged chips in a 132/84 MLC package

- 132 external pin connection
- direct interface to the Tektronix DAS
- controlled impedance lines
- provision for chip resistors or capacitors at either end of a trace

5.1.1 PCB Design

The dimensions of the PCB is 4.1" square and the top and bottom artwork of the board are shown in Figures 5.1 and 5.2, respectively.

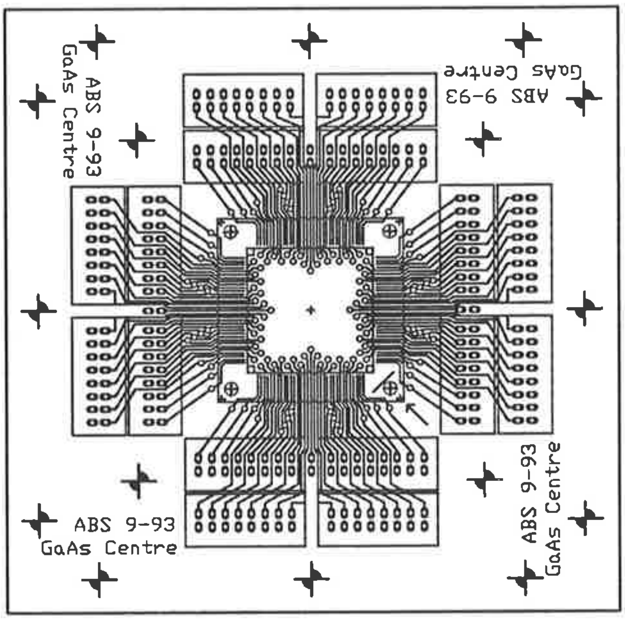


Figure 5.1: Top layer of the test fixture PCB.

The output pads of the chip drive into a resistor to ground whose resistance is nominally 50Ω. The design parameters are:

- signal traces of 0.015" wide and 0.010" spacing (0.025" pitch)
- available board material is Teflon ($\epsilon_r = 2.55$), double sided 1-ounce copper. Thickness is either $\frac{1}{16}$" or $\frac{1}{64}$"

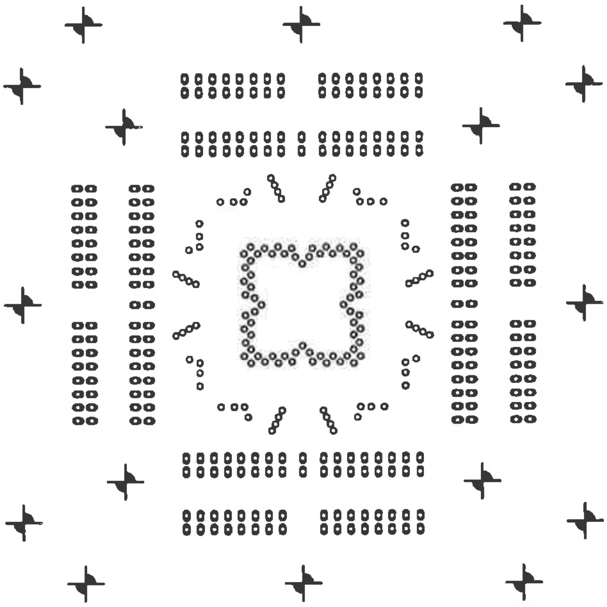


Figure 5.2: Bottom layer negative of the test fixture PCB.

- pins to the DAS must be a signal–ground pair which can be arranged in a group of eight or individually. The pin spacing in both directions is 0.100". They must be gold to make reliable contact and to prevent metal migration of tin into the DAS probe tips
- due to the compact nature of the test board, trace lengths could not be equalised so there is a difference in delay between some traces

The traces on the PCB are treated as lossy transmission lines. The structure is a microstrip where the backside ground plane reflects the signal to produce its dual. The impedance may be approximated by:

$$Z_0 = \frac{87}{\sqrt{\epsilon_r + 1.41}} ln \left[\frac{5.98h}{0.8w + t} \right]$$

where $\epsilon_r = 2.55$ is the relative dielectric constant of Teflon, $t = 0.0013$" is the thickness of the copper, $w = 0.015$" is the width of the track and $h = \frac{1}{64}$"(0.0156) is the thickness of the dielectric. The characteristic impedance is $Z_0 = 85\Omega$. A board thickness of $\frac{1}{16}$" produces a characteristic impedance of 145Ω. To produce a characteristic impedance of

54Ω on a $\frac{1}{64}$" board, the track width must be doubled to $w = 0.030$" which is impractical in this application. A board trace may be designed in one of three ways:

- line terminating resistor to ground (Rt)
- line source resistor to ground and terminating resistor to ground (Rs, Rt)
- line series source resistor and terminating resistor to ground (Rss)

Provision for source and terminating resistors or capacitor to ground for each signal line were made. HSPICE was used to model a signal trace driven by one of the output pads through a bond wire, lead, transmission line and terminating in a probe. The equivalent circuit is shown in Figure 5.3. To determine the effects of various board parameters on the signals, the following simulations were done:

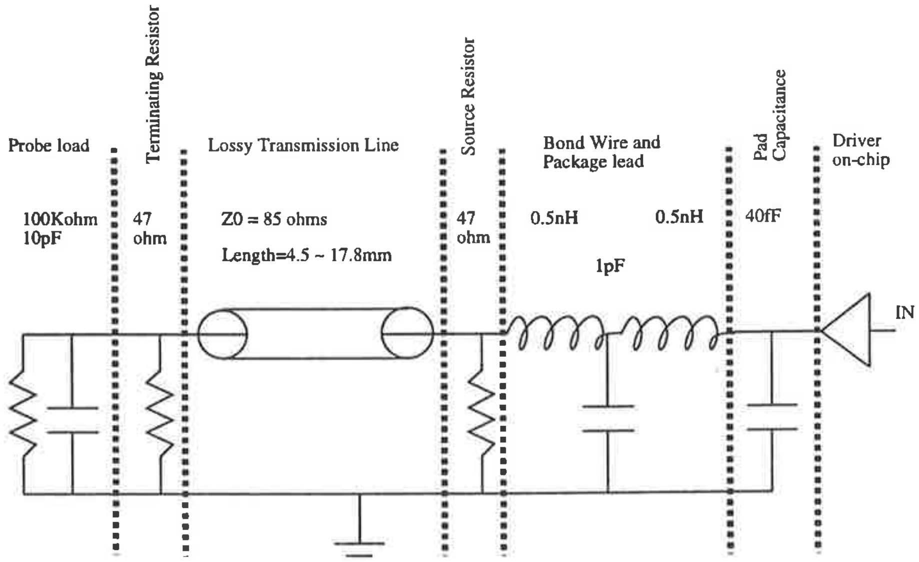


Figure 5.3: Equivalent circuit of a signal driven off chip.

- Using a $\frac{1}{64}$" thick Teflon PCB:
  - A $40mm$ long line with no source resistance and terminating resistors of 25, 50, 75 and 100Ω. Figure 5.4 shows the 50Ω load has the best damping although there is too much ringing at the start of the TL.

  - A $40mm$ long line with a 50Ω terminating resistance for source resistance values of 25, 50, 75 and 100Ω. Figure 5.5 shows the 75Ω source resistor gives the best result with a slight overshoot and a $1\,V$ swing.
  - A $40mm$ long line with a 50Ω terminating resistance and a 50Ω source resistance with track widths of $0.3048mm$, $0.381mm$ and $0.5mm$ (Figure 5.6).
  - A $40mm$ long line with a 25Ω terminating resistance and a 25Ω source resistance with track widths of $0.3048mm$, $0.381mm$ and $0.5mm$ (Figure 5.7). This has a smaller output voltage swing than the 50Ω case (Figure 5.6).

- Using a $\frac{1}{16}$" thick Teflon PCB:
  - A $40mm$ long line with a 25Ω terminating resistance and a 25Ω source resistance with track widths of $0.3048mm$, $0.381mm$ and $0.5mm$ (Figure 5.8).

All simulation results show the on–chip pad input signal as well as the response at the start (Rs) and end (Rt) of the PCB line. These simulations show there is negligible difference in the response due to changes in track width. The source and terminating resistors should be about the same for a good response (fast rise time with a small overshoot). The delay from the input to the PCB pin is about $1ns$. Figure 5.9 shows a signal being driven onto the chip through a board trace. The board is $\frac{1}{64}$" thick and the line is terminated with a resistor, Rt, near the chip. Using these results, the board was designed and the length of the tracks measured and resimulated to determine the skew and the resistor values to be used. There are four cases considered; short lines and long lines with source resistors underneath or alongside the chip. This arises because of area constraints around the chip. The four possible interconnect types are:

- $17.8mm$ line with the source resistor under the chip
- $19.05mm$ line with the source resistor outside the chip
- $32.6mm$ line with the source resistor outside the chip
- $30.48mm$ line with the source resistor under the chip

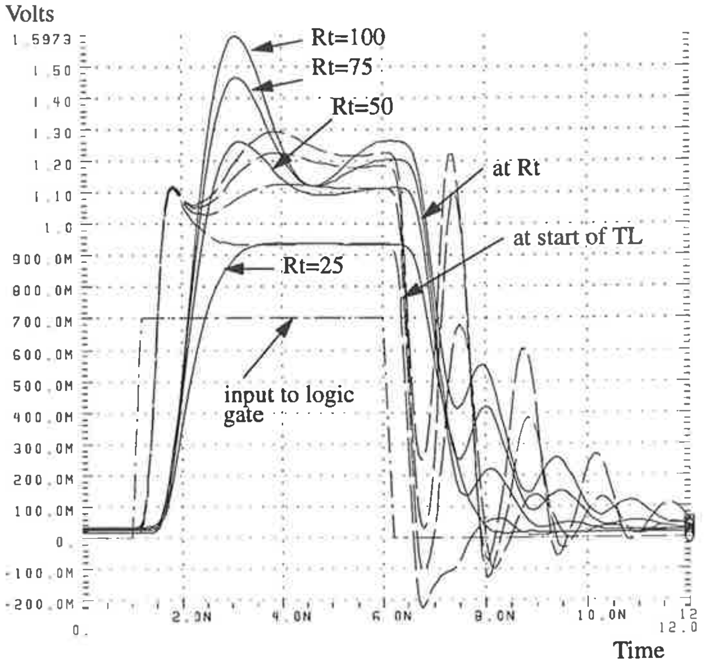


Figure 5.4: Simulation of a $40mm$ long line with no Rs and Rt = 25, 50, 75 and 100Ω.

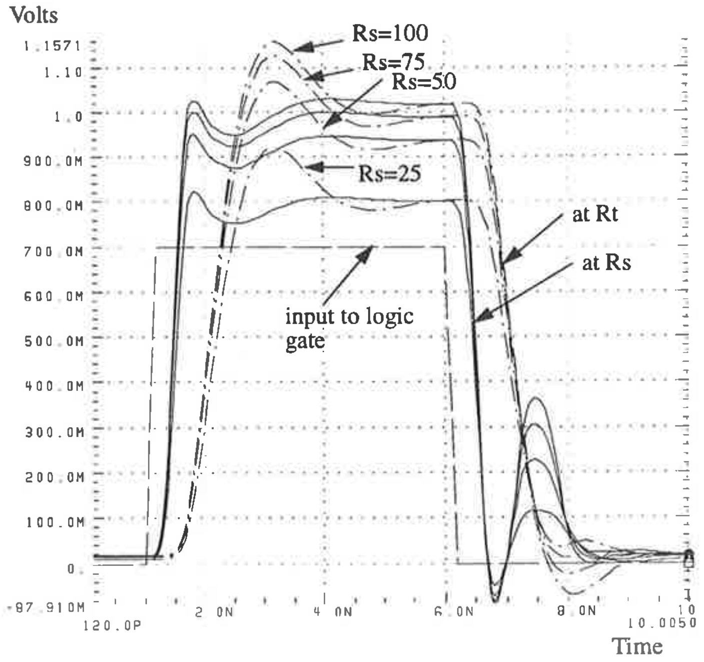


Figure 5.5: Simulation of a $40mm$ long line with Rt = 50Ω and Rs = 25, 50, 75 and 100Ω.

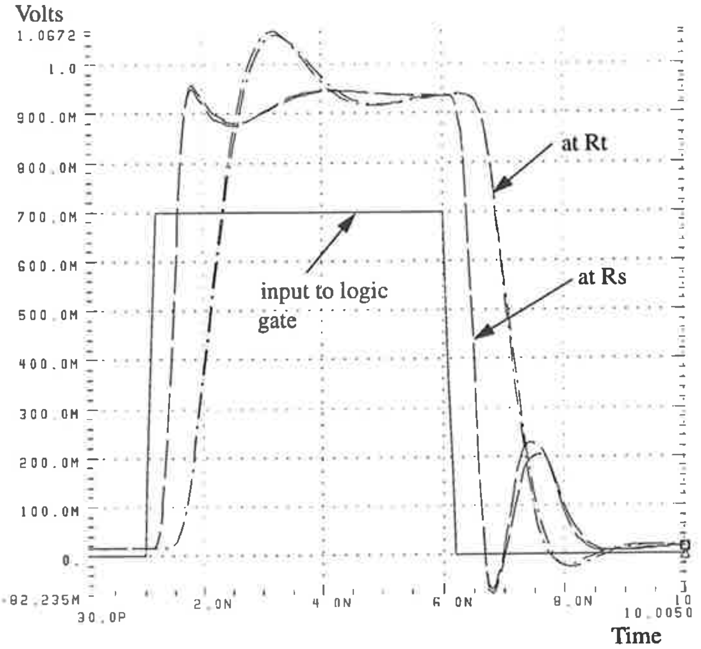


Figure 5.6: Simulation of a $40mm$ long line with $Rt = 50\Omega$ and $Rs = 50\Omega$ for track widths of $0.3048mm$, $0.381mm$ and $0.5mm$.

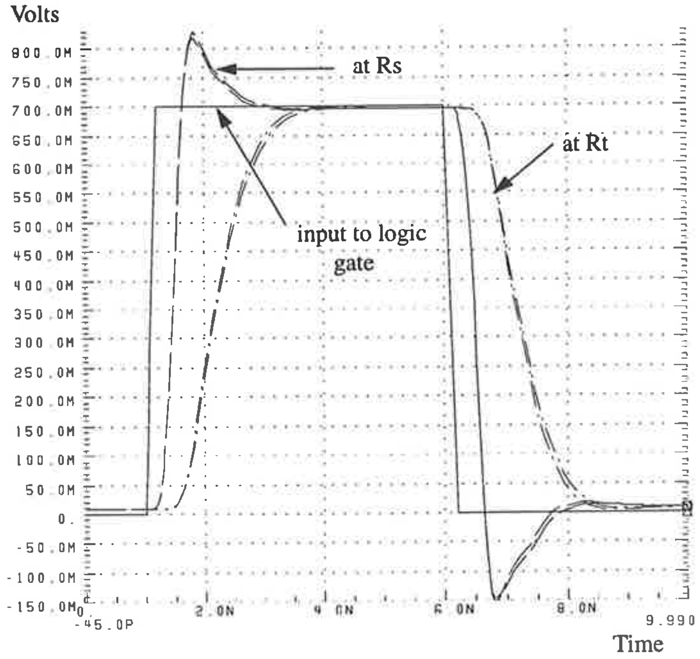


Figure 5.7: Simulation of a $40mm$ long line with $Rt = 25\Omega$ and $Rs = 25\Omega$ for track widths of $0.3048mm$, $0.381mm$ and $0.5mm$.

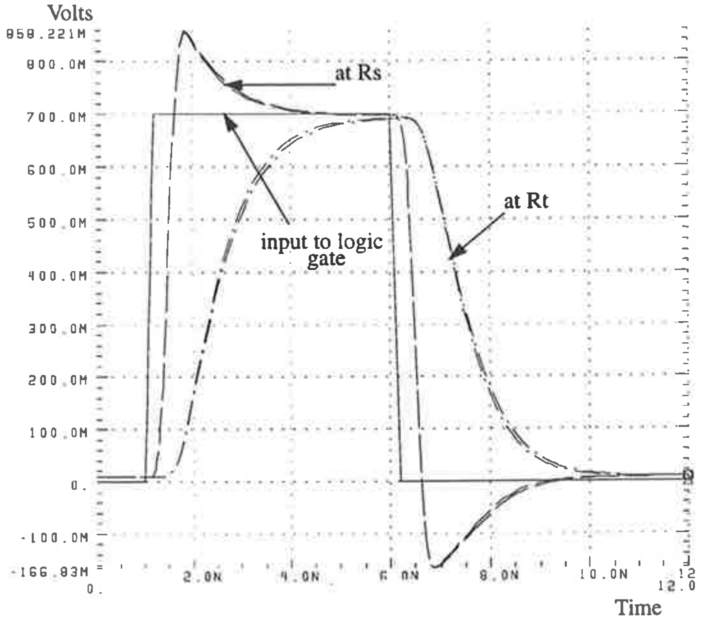


Figure 5.8: Simulation of a $40mm$ long line with $Rt = 25\Omega$ and $Rs = 25\Omega$ for track widths of $0.3048mm$, $0.381mm$ and $0.5mm$. The PCB is $\frac{1}{16}$" thick.

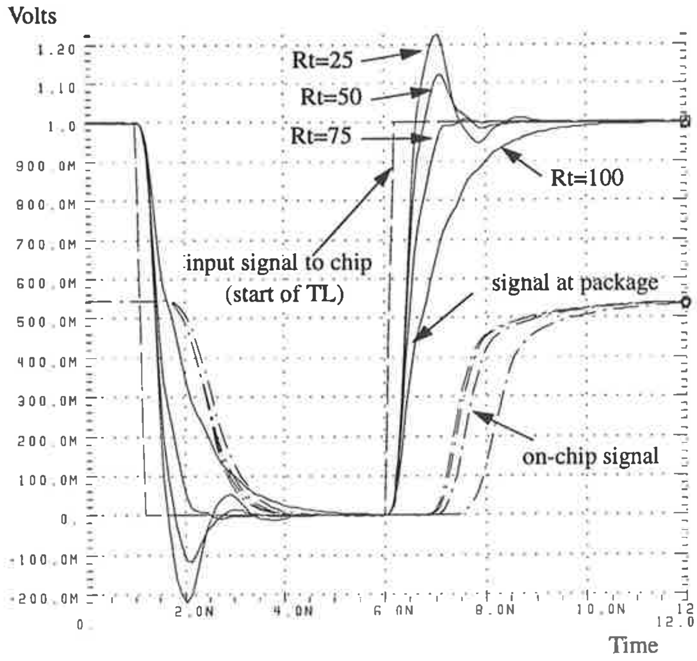


Figure 5.9: Simulation of a $40mm$ long line with Rt = 25, 50, 75 and 100Ω, signal is being driven onto the chip.

The final design details are:

- track width is $0.381mm(0.015")$
- track pitch is $0.635mm(0.025")$
- PCB thickness is $0.396mm$ $(\frac{1}{64}")$
- source resistor to ground is 47Ω
- terminating resistor to ground is 47Ω

Figure 5.10 shows a simulation of the four line types with 47Ω source and terminating resistors to ground. The maximum skew between the signal lines is around $350ps$ and the worst case skew between two signal lines from the pad to the PCB pin is $390ps$. The output voltage swing is $0.9\,V$ but the resistors can be increased to around 68Ω to obtain a larger swing at the expense of more ringing.

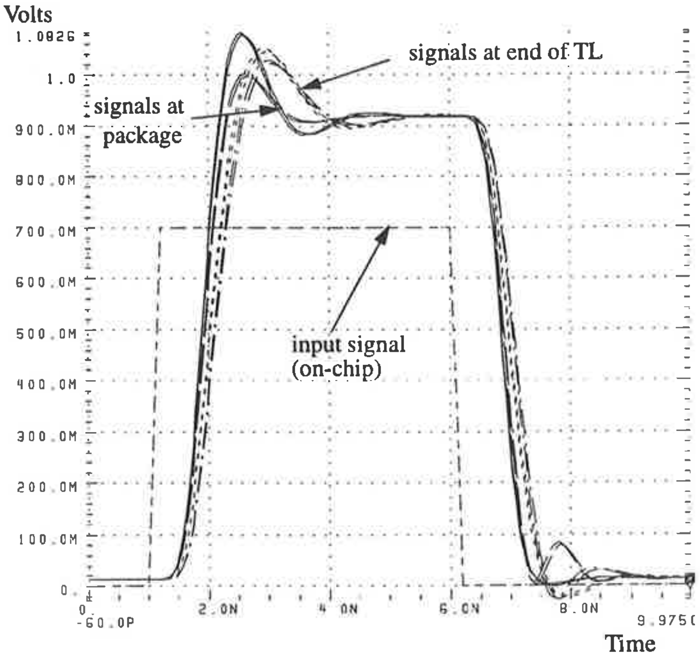


Figure 5.10: Simulation of the four possible interconnect types on the PCB with 47Ω source and terminating resistors.

5.1.2 Construction

The test jig was made from aluminium with a sealed cavity under the PCB. Figure 5.11 shows the details of the test jig. The terminating resistors and capacitors are a surface mount 805 type package which are low inductance and suitable for high frequency applications. Terminating resistors for chip output lines are mounted next to the PCB pins as shown in Figure 5.12. Source resistors are mounted as close to the chip as possible. All power pins have decoupling capacitors (0.1 to $0.47\mu F$) connected to ground near the chip. A photograph of the test jig with the chip mounted is shown in Figure 5.13.

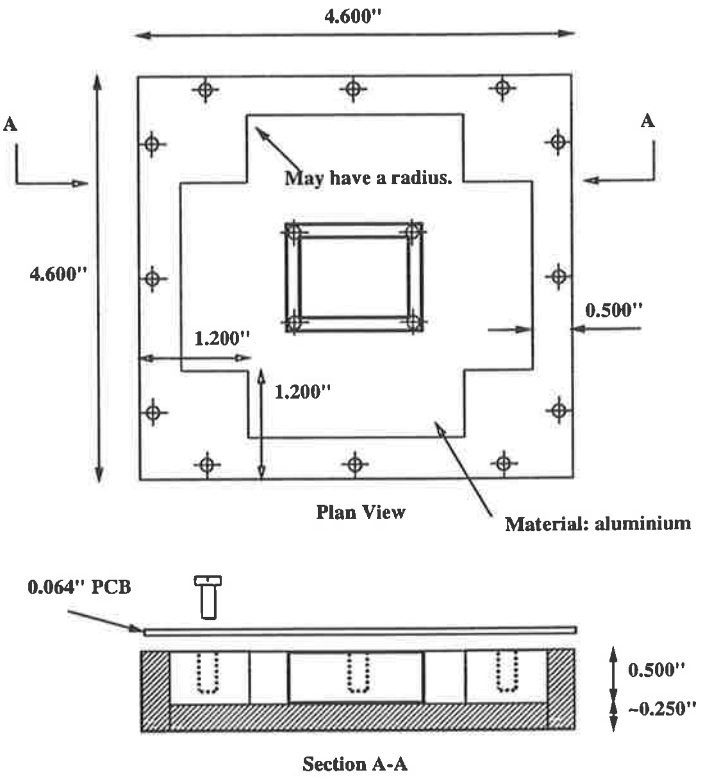


Figure 5.11: High speed test jig for 132/64 MLC packages.

5.2 Test Equipment and Set–up

The arrangement of power supply connections to the chip is critical to the correct operation of the chip due to the large current drawn ($1A$) and the high speed nature of

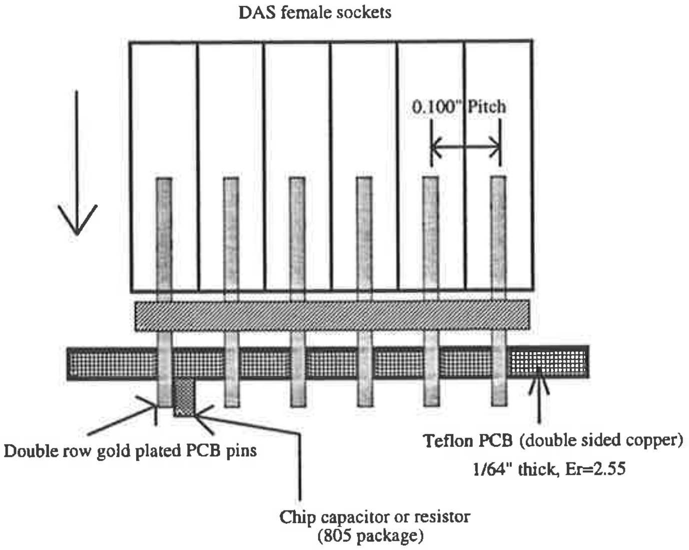

DAS female sockets
0.100" Pitch
Double row gold plated PCB pins
Teflon PCB (double sided copper)
1/64" thick, Er=2.55
Chip capacitor or resistor
(805 package)

Figure 5.12: Cross section through the PCB.

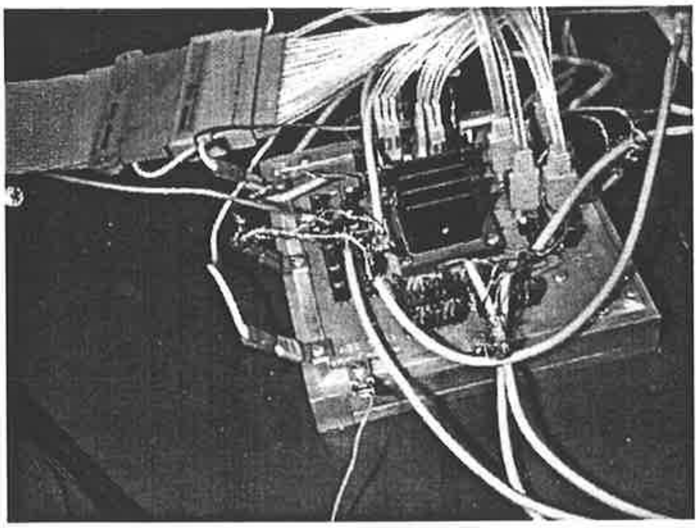


Figure 5.13: Photograph of the high speed test jig with a chip and heat-sink installed.

the I/O signals. Appendix C contains a description of the power supply connections and sequencing to avoid ground loops, ground bounce and crosstalk. Low speed functional and high speed testing was carried out using a Tektronix Digital Analysis System (DAS) 9200. The DAS has two 92S16 pattern generation cards (18 signal lines at $50MHz$) and a 92A96 data acquisition card which can monitor up to 24 channels at a $400MHz$ acquisition rate. The support software allows test vectors to be generated, test results displayed and stored. Appendix C contains a description and specifications of the DAS and its pattern generation and acquisition modules. A LeCroy 9360 digital storage oscilloscope was used for measurement of high speed signals and was useful in de–bugging the system. The oscilloscope has a sampling rate of $5GSamples/s$ and a $600MHz$ internal bandwidth. The active probes used have a $1GHz$ bandwidth.

5.3 Circuit Testing

Testing of the following structures was performed:

- enhancement and depletion MESFETs
- systolic cell
- systolic ring
- clock generation circuit

In all testing, the heat–sink was in place and the chip was allowed to reach a steady operating temperature. The input and output of data to and from the ring is controlled by the ring controller which is clocked at the same rate as the rest of the chip. For testability purposes, two outputs are provided. One is from the 16–bit output of the first systolic cell and the other is the 16–bit output of the ring. This allows independent functional testing of both a systolic cell and the systolic ring which can be carried out under DC conditions with external clock control. High speed testing of the systolic ring is carried out by loading the input operands into the systolic ring at low speed for one circulation around the ring at which point the ring closes. The clock for the chip is provided by the DAS. The clock is then switched to being internally generated (up to $1GHz$) and the operands are recirculated internally at high speed while the output is monitored for the result. Appendix C shows the channel allocation and the test programs used to test the PE chip.

5.3.1 Practical Terminations

The output of the ring oscillator was used to check the terminations on the test jig. The terminating resistance for adequate voltage swing was too small with the two 68Ω resistors at each end of the PCB. The source resistor was removed and the voltage swing increased to around $1.1V$. This may be due to a larger than expected series resistance at the source of the PCB trace possibly due to contact resistance of the package on the PCB. Figure 5.14 shows the clock output waveform with a 67Ω terminating resistor measured using the oscilloscope. From Figure 5.14 the delay of the clock input to output signal is around 2 to $2.5ns$ and the rise and fall time of the output clock was measured at $1ns$. However, the measurement of the signal will be limited by the bandwidth of the oscilloscope.

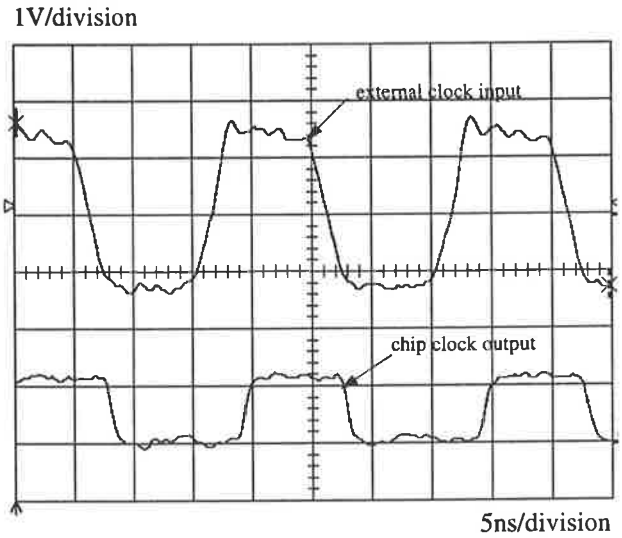


Figure 5.14: External clock input and chip clock output waveforms.

5.3.2 Fix 1 : Ground Bounce

Initial testing showed that there was a severe ground bounce problem with the four 'PP' operand output pads. The problem can be seen in Figure 5.15 where signal '1' is 'p2out' and signal '2' is the ground for that set of pads measured at the pin of the chip. 1.6 V peak–peak of bounce was observed when any of the Pout signals change. The ground connection for the four 'PP' pads is not connected to the internal ground plane of the package but to a signal line which was externally grounded. This was carried out because of pin limitations in the package. The solution was to solder the package pin (114) to an adjacent ground pin (115) that is connected to the ground plane thus considerably shortening the ground loop and the same signals are shown in Figure 5.16. This illustrates the magnitude of noise that may be generated through a deficient power supply and ground scheme.

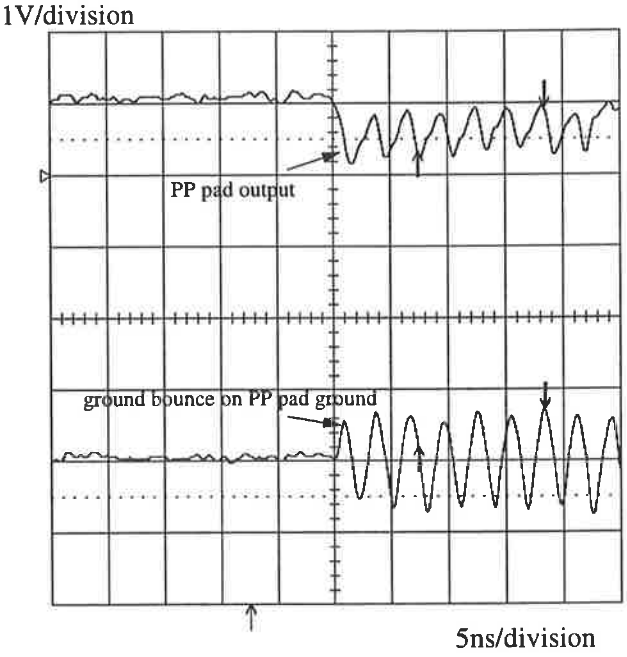


Figure 5.15: Signal p2out and the pad ground showing ground bounce.

5.3.3 Fix 2 : Separate Power Supplies

To improve the stability of the circuits, the circuit and pad supplies should be separated. Unfortunately, a bonding error left both a circuit supply and the pad supply for 'Xout' connected to 'PWR2' in the package. Since the circuit has several other sources for power, the bond wire to the centre *Vdd* supply on the chip (connected to 'P2') was removed and pins 33, 34, 99 and 100 were connected to *Vddp* instead of *Vdd*. Separating the power

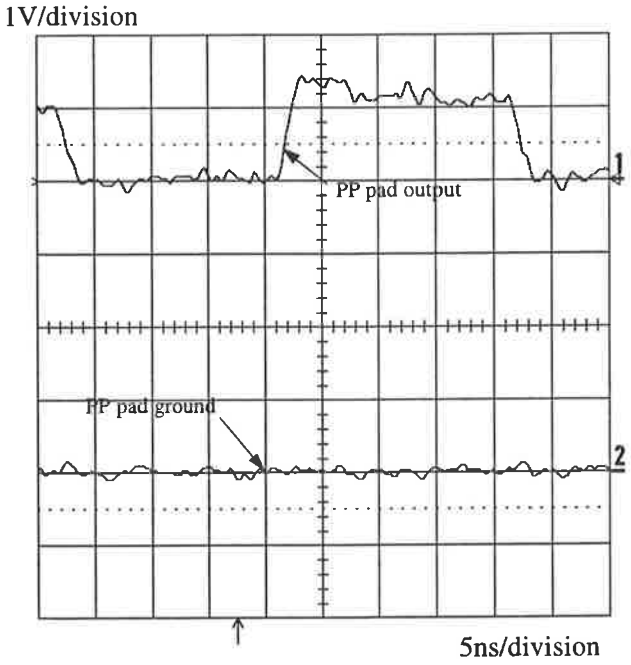


Figure 5.16: Signal p2out and the pad ground with the ground bounce solved.

supplies made the circuit considerably more stable.

5.4 Fingered MESFET Test Structures

In order to allow verification of the SPICE models used in the design, an enhancement mode fingered MESFET and a depletion mode fingered MESFET were fabricated. The layouts of the EFET and DFET are shown in Figures 5.17 and 5.18, respectively. The fingered structure is an economical way of producing a wide transistor capable of handling large currents and provides some degree of protection against damage by the testing equipment. A Tektronix transistor tracer was used to test the MESFETs.

The drain–source voltage was swept over a 0 to 4 V range on testing each MESFET. The gate–source voltage is varied using a step voltage control and choosing the number of steps to be traced. A capacitor was connected between the drain and source terminals of each MESFET to prevent oscillations between the drain and source. This is caused by the parasitic capacitances of the MESFET and inductance and capacitance of the test circuit acting as a resonant circuit and oscillating producing a negative resistance effect in the transistor characteristic. The value of capacitance used must be larger than the

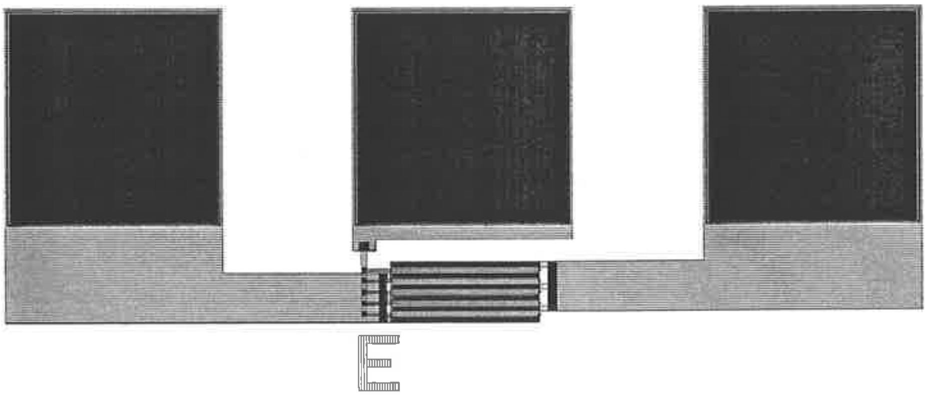


Figure 5.17: Layout of a fingered enhancement mode MESFET (5 fingers × 74.8μ wide).

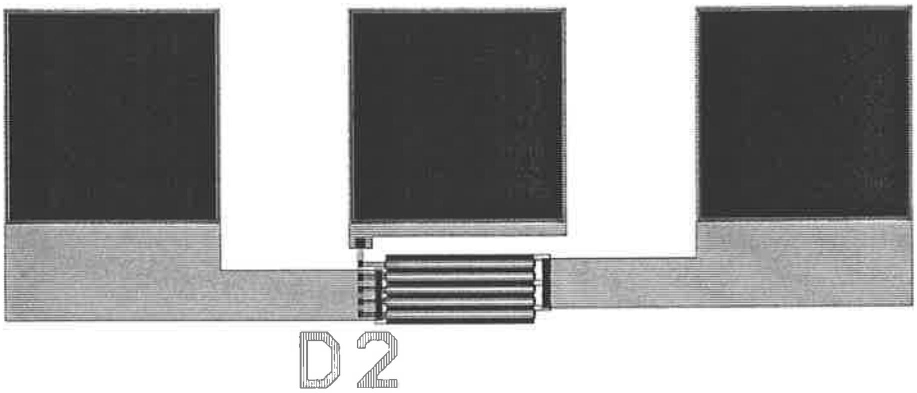


Figure 5.18: Layout of a fingered depletion mode MESFET (5 fingers × 74.8μm wide).

intrinsic capacitance of the MESFET. A $0.1\mu F$ chip capacitor was used and mounted on the PCB as close to the PE chip package as possible.

Enhancement Mode MESFET

The tracer was set to provide six $0.1\,V$ steps of gate voltage starting at $0.2V$, *i.e.* $V_{gs} = 0.2$, 0.3, 0.4, 0.5, 0.6, $0.7\,V$ to test the EFET. A photograph of the transistor tracer screen showing the EFET I–V characteristics is shown in Figure 5.19. The horizontal scale is 0.5 drain–source volts/division and the vertical scale is $5mA$ of drain current/division. It can be seen in Figure 5.19 that there is a large amount of hysteresis present in the transistor characteristics. This is due to charge being trapped in the substrate below the transistor. As a consequence of the high resistance of the semi–insulating GaAs substrate, the charge is dissipated slowly and so hysteresis is predominately a low frequency effect. The low scan frequency of the transistor tracer (approximately $1kHz$ maximum) causes a noticeable hysteresis effect in the characteristics, however, at higher frequencies the problem diminishes. The amount of hysteresis will also depend on the amount of charge stored in the substrate and hence on V_{gs} and V_{ds}. This effect is *not* modelled. To

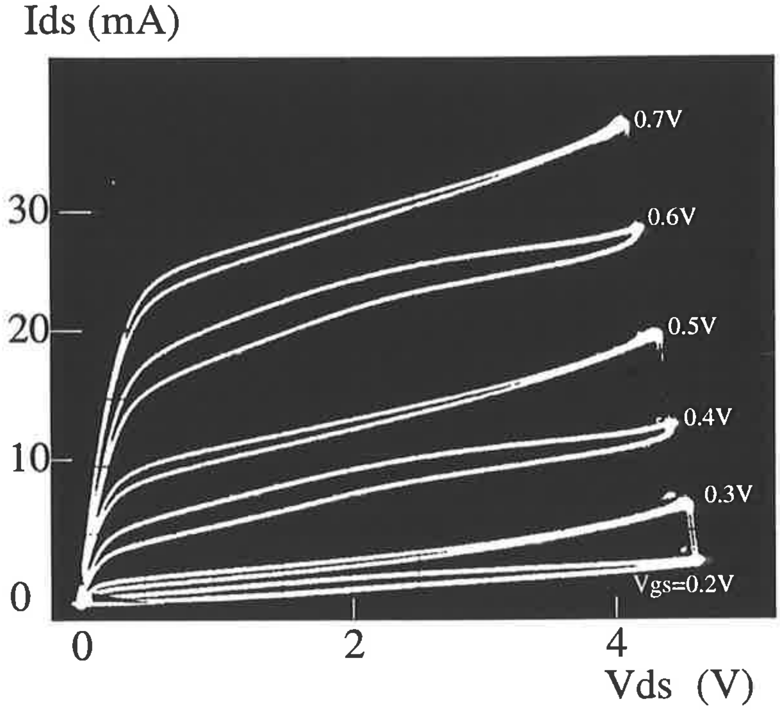


Figure 5.19: Photograph of the EFET I–V characteristics from the curve tracer.

facilitate a comparison between simulation and measurement, the I–V characteristics were measured from the photograph and imported into Matlab. Because the hysteresis effect is not modelled, the midpoint of each curve was taken. To determine the appropriate simulation characteristics temperature and process variation must be found. During measurement, the ambient temperature was about 30°C so the temperature of the device was estimated to be 30 to 50°C. Different process variation models were tried on the SPICE deck generated from the transistor layout until the closest match to the measured characteristics was found. The results of the HSPICE simulation at 50 °C using *typical–typical* process parameters is shown in Figure 5.20 superimposed on the measured I–V curve. It can be seen that reasonable agreement was obtained between measurement and simulated data.

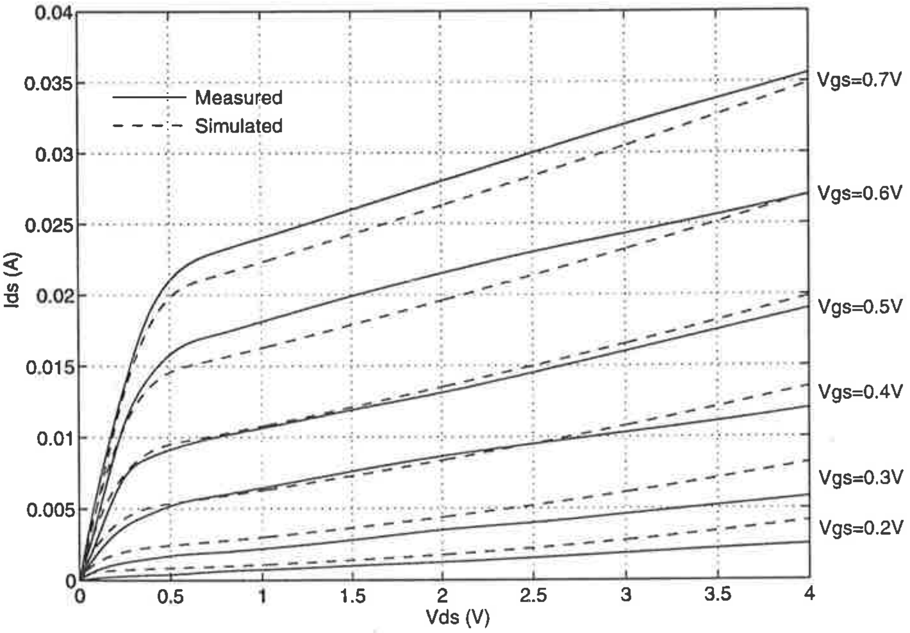


Figure 5.20: Comparison of measured and simulated EFET I–V characteristics using *typical* process parameters.

Depletion Mode MESFET

The DFET was tested using the same procedure used for the EFET. Due to a lack of response from the DFET on the same chip the EFET characteristics were taken from, a second chip had to be used to test the DFET. This chip was found to be slower than the first chip. The tracer was set up to provide eight steps for the gate–source voltage at 0.2 volts per step starting at -1.2 volts, *i.e.* $V_{gs} = -1.2, -1.0, -0.8, -0.6, -0.4, -0.2, 0, 0.2\,V$. A photograph of the DFET I-V characteristics is shown in Figure 5.21. The scales used are 0.5 volt/division for drain–source voltage (V_{ds}) and $10mA$/division for drain–source current (I_{ds}). The hysteresis effect is clearly evident. HSPICE simulations

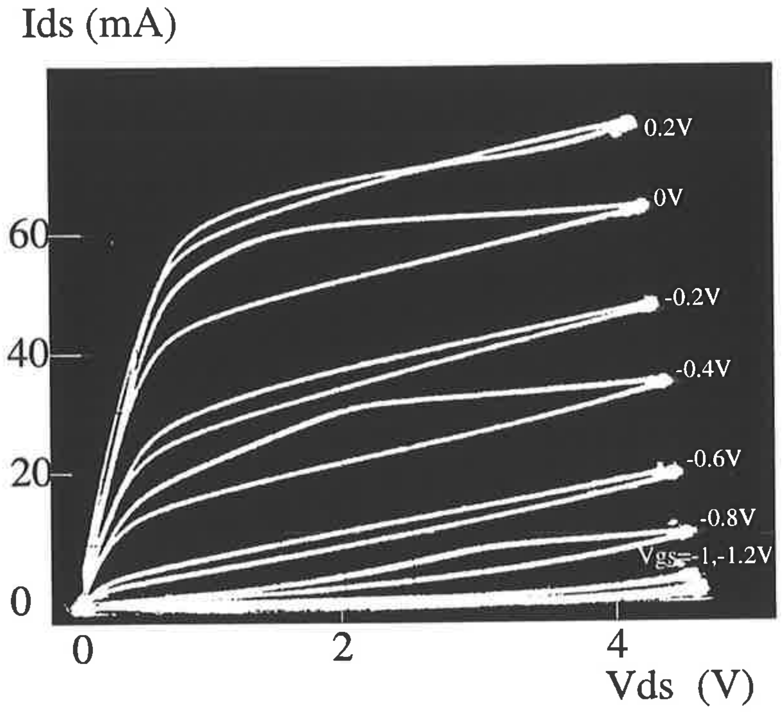


Figure 5.21: Photograph of the DFET I–V characteristics from the curve tracer.

were also done for the DFET. The closest simulation was found using 1σ–*slow* process parameters at 50 °C. A graph comparing the measured and simulated FET characteristics is shown in Figure 5.22.

Discussion

It can be seen that in both cases the model provides a reasonable approximation of the measured transistor characteristics. At low values of V_{gs}, the model appears to over–estimate the measured characteristics, while at high values of V_{gs}, there appears to be an under–estimation. Middle values of V_{gs} show close correlation. There are some factors which may account for these discrepancies:

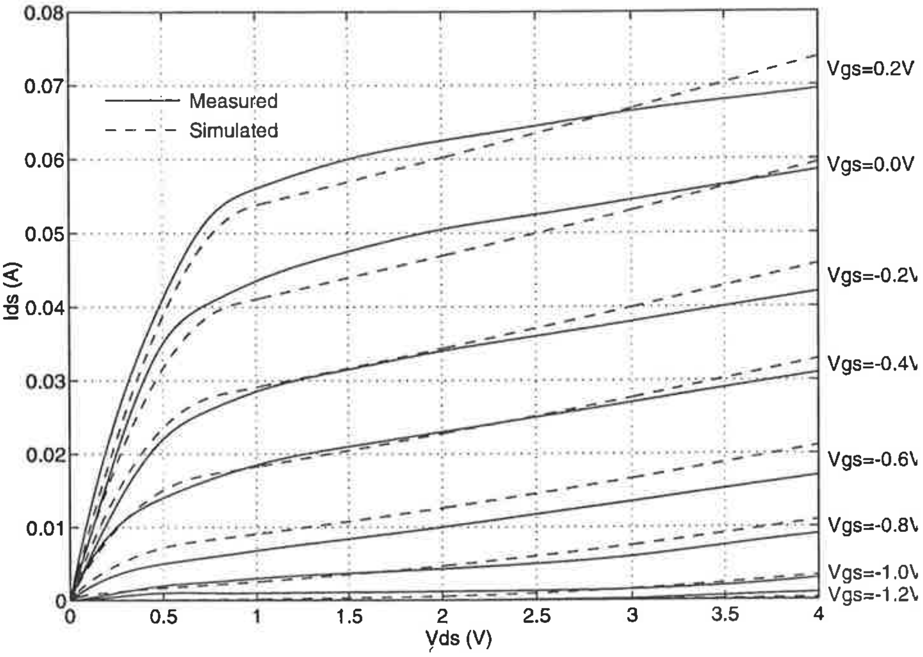


Figure 5.22: Comparison of measured and simulated DFET I–V characteristics.

- The models were not designed for such wide transistors, they were determined using 10 μm wide MESFETs. It is known that transistor characteristics do not scale linearly [PuEs88].

- The simulation does not take into account the geometry of the finger structure, and treats the fingered MESFET simply as five MESFETs in parallel.

- The models were sourced from Vitesse Semiconductor Inc. though MOSIS. The chips were actually fabricated at Thomson–CSF. Although the process has been replicated, there will undoubtedly be some variation in performance between the two. Thomson–CSF has not supplied models derived from their foundry process.

- The DFET is rotated by 90 degrees with respect to the GaAs crystal plane alignment position for maximum transconductance. This may account for the slow DFET.

The results of these measurements indicate reasonable validity of the supplied models.

5.5 Systolic Cell Functional Testing

The functionality of the systolic cell was tested using exhaustive test data generated by the C Program which checked the design functionality using IRSIM. The chip and DAS were configured as detailed in Appendix C. To configure the clock for externally applied input, pin 22 (*CKs1*) is connected to the logic level high power supply and pins 20 (*CKs2*), 23 (*CKstop*) and 25 (*CKrate*) are connected to GND.

5.5.1 Generating Test Vectors

For the systolic cell, the $SELO$ input should be tied high to multiplex the input operands permanently to the first systolic cell. In this mode the output of the PE is permanently tied to the output of the first systolic cell. The length of the mantissa and exponent operands may be arbitrarily long provided they follow the format shown in Figure 5.23. The first test was to check that the X, Y and $INSTR$ fields operate correctly in multiplication mode at a $50MHz$ clock rate. Figure 5.24 shows the correct result of zero by zero. In the systolic cell test mode, the X, Y and PP operands are inverted in the test result figures. Figure 5.25 shows the systolic cell operating in multiplication mode where the input operands are $X_{in} = 0.00000001 \times e^{001}$ and $Y_{in} = 0.00000001 \times e^{001}$. The convention used in this chapter is that the signal S_{out} denotes $S - out$ in hexadecimal format shown in the DAS test result figures. The outputs $X_{out} = 0.00000001 \times e^{001}$,

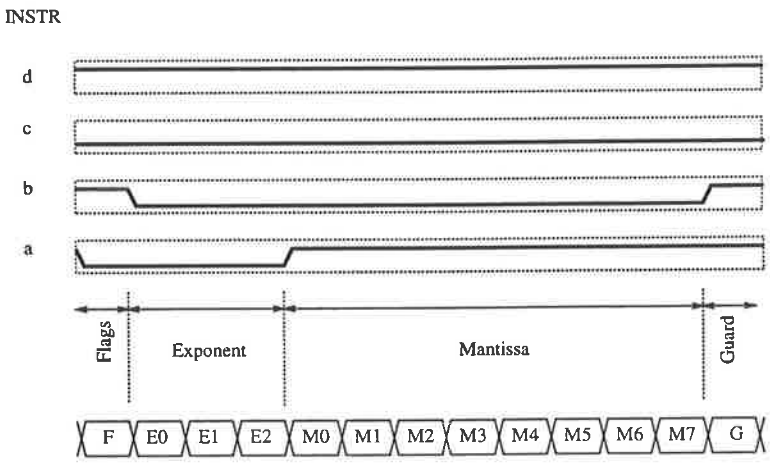


Figure 5.23: Instruction nibble for multiplication mode.

$Y_{out} = 0.10000000 \times e^{001}$ and $INSTR_{out}$ are correct for three successive cycles. Y_{out} has been shifted eight places in the mantissa and the exponent is unchanged due to the systolic cell holding the first mantissa digit and releasing it at the end of the mantissa having multiplied it with each of the X mantissa digits. The X and Y exponents are digit-serially added. The resultant partial product is $P_{out} = 0.00000001 \times e^{002}$.

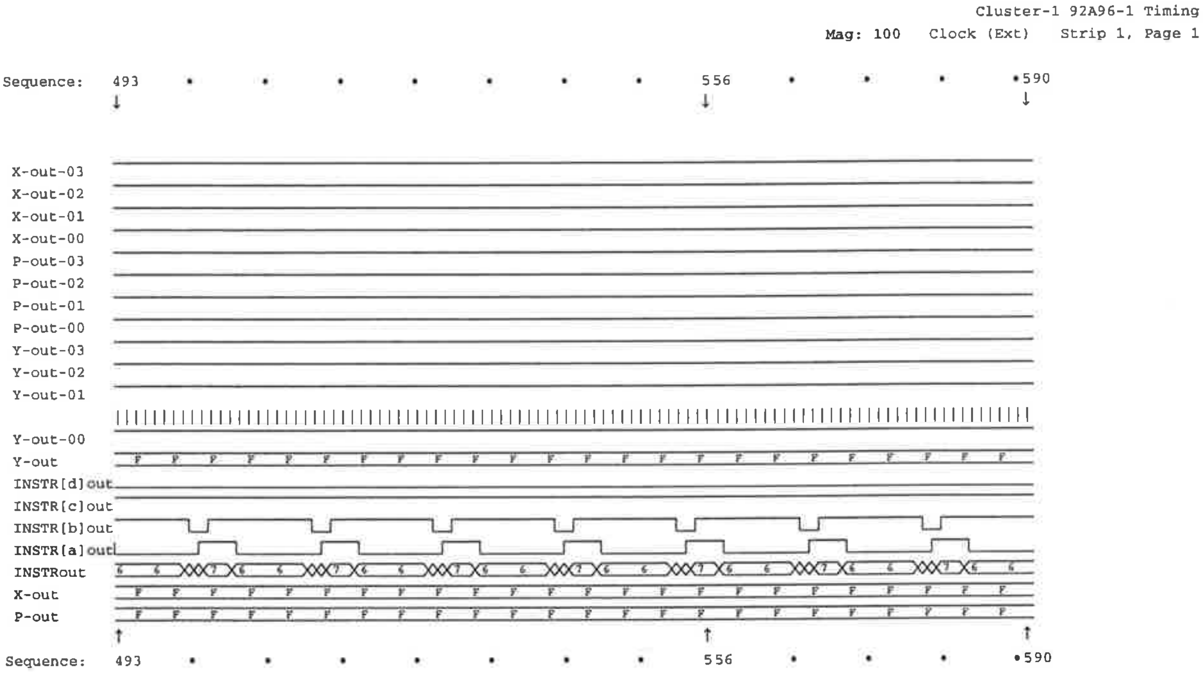


Figure 5.24: Systolic cell testing in multiplication mode at $50MHz$ (zero by zero).

Figure 5.26 shows the acquired data from the DAS of the systolic cell in multiplication mode at $50MHz$. These results show the correct operation of the systolic cell.

5.6 Systolic Ring Testing

The maximum clock speed of the DAS is $50MHz$ for data generation, so a scheme to determine the maximum clock frequency was devised given that data can be acquired at a $400MHz$ rate. Operands are loaded into the ring under external low speed clock control up to $50MHz$. The internal high speed clock is stopped while the clock generator multiplexers configure the clock to run at one of the specified speeds. The stop signal is released and the outputs are monitored while the operands are recirculated and unloaded from the systolic ring. The three operations, add, multiply and denormalise were tested

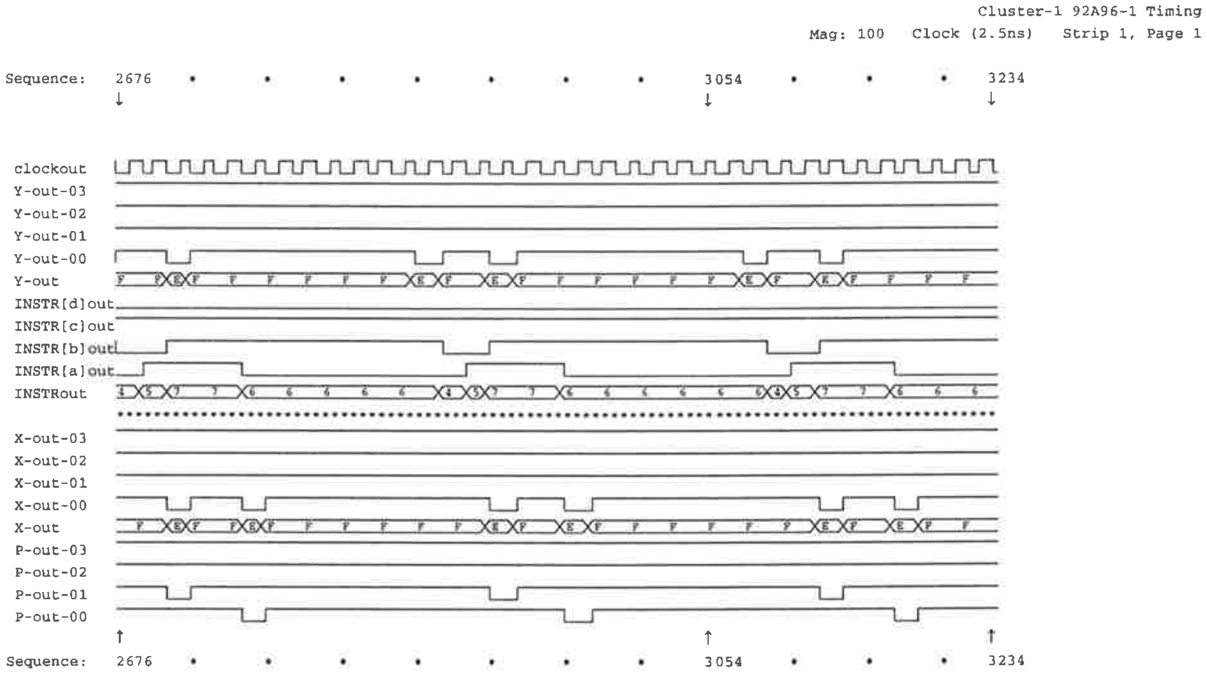


Figure 5.25: Systolic cell testing in multiplication mode at 50MHz.

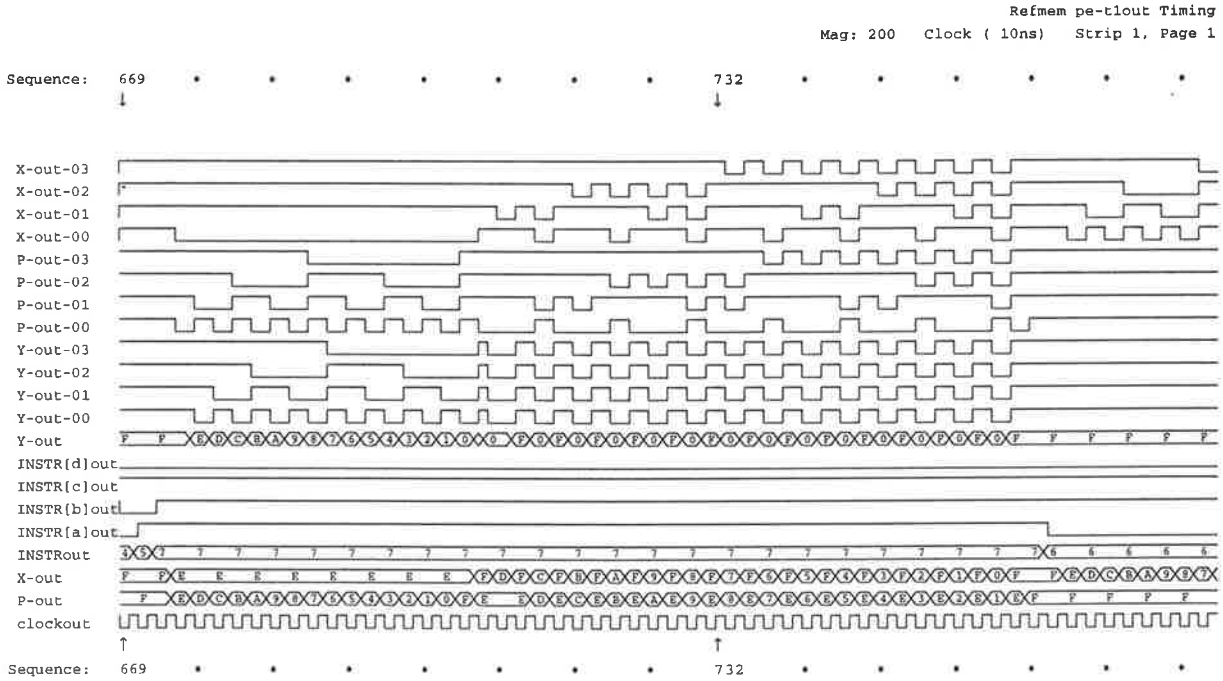


Figure 5.26: Results of the systolic cell in multiplication mode at 50MHz.

with the internal clock generator configured for nominal $250MHz$ operation. The actual frequency was measured using the oscilloscope to be $128MHz$. The result of an addition of:

$$X_{out} = 0.00000C21 \times e^{041}$$

and

$$Y_{out} = 0.00000831 \times e^{042}$$

is:

$$P_{out} = 0.00001452 \times e^{083}$$

after the second appearance of the data at the output (unloading cycle) of Figure 5.27. The first cycle is the single recirculation. This computation excludes the denormalisation operation. Although the input operands do not have the same exponent, the addition of the X and Y mantissa is correct. Figure 5.28 shows the PE operating in floating point denormalisation mode where the instruction given to every systolic cell is to denormalise the Y operand. The resultant Y_{out} mantissa is shifted by eight digits (four cells with two circulations) and the exponent has been incremented by eight where:

$$Y_{in} = 0.00000831 \times e^{042}, X_{in} = 0.00000C21 \times e^{046}$$

and P_{out} is the result of adding X and the shifted Y operands:

$$P_{out} = 0.00001452 \times e^{04A}$$

The denormalisation operation has been shown to work with this architecture, however more control is required to subtract the exponents and control how many digit denormalisations are to be carried out before the addition takes place. This has not been implemented in this chip. Figures 5.29a, 5.29b, 5.29c and 5.29d show the systolic ring PE operating at $91MHz$ in multiplication mode on a variety of input operands. Note the resultant P_{out} is the high order result of the multiplication. Figures 5.30a, 5.30b and 5.30c show the PE operating at $128MHz$ in multiplication mode and shows the final circulation of the operands around the ring and output. The exponent field is where $INSTRout = 1000$ and the mantissa field is where $INSTRout = 1001$. In some cases, the clock was not completely recovered due to a poor 50Ω load from the oscilloscope. The computation is $P_{out} = X_{in} \times Y_{in}$ where:

$$X_{in} = X_{out} = 0.041F9060 \times e^{00B}, Y_{in} = Y_{out} = 0.FA5F6802 \times e^{0FC}$$

and

$$P_{out} = 0.04085C612 \times e^{107}$$

Note that the Y operand has been rotated by four digits in the first circulation of data.

Figure 5.31 shows the PE operating in multiplication mode at $350MHz$ (clock operation set for $600MHz$ *typical–typical*). The inputs were $Y_{in} = 0.FA5F6802 \times e^{0FC}$, $X_{in} = 0.041F9060 \times e^{00B}$. While the result, P_{out} is not correct, $INSTR_{out}$, $X_{out} = 0.041F9060 \times e^{00B}$ and the exponent of Y_{out} and most of the mantissa are correct. This shows high speed synchronous clock operation is possible. The computation failure is in the critical paths of the chip not being able to complete the computation before the next clock cycle. These results show that the PE chip is fully functional, however, the chips tested were at least 0.5σ–*slow* as seen in the slow oscillator speeds in the next section.

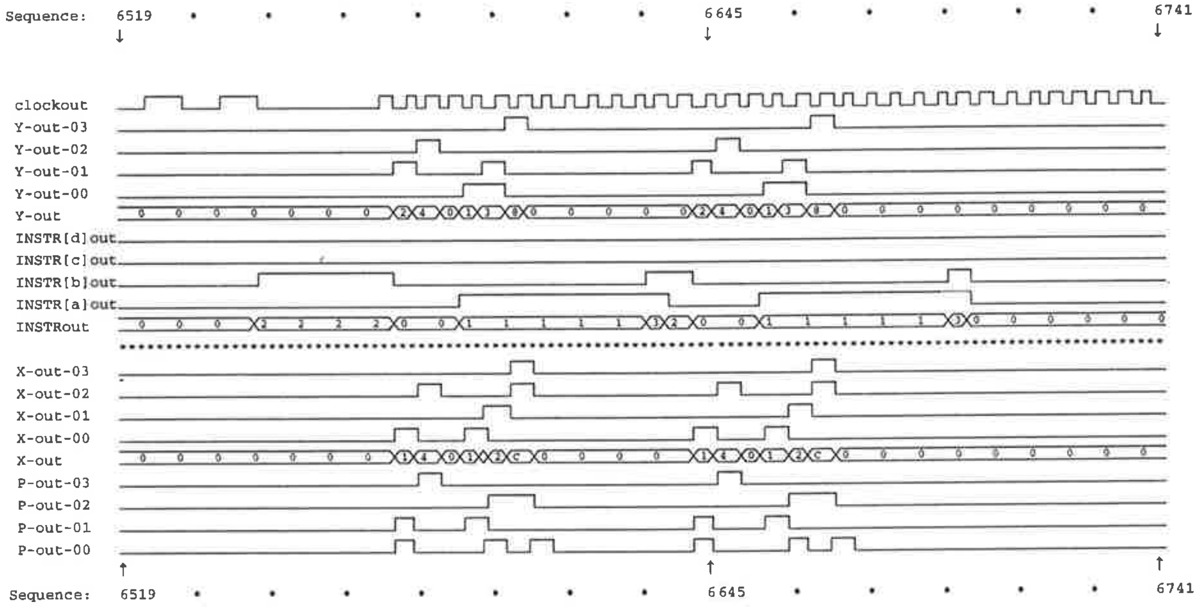


Figure 5.27: Test results of the PE in floating point addition mode at 128MHz.

5.7 Clock Generation Circuit

To test the clock generation circuit, the chip should be powered in the same way as the systolic cell case. The multiplexer control inputs that control the clock output frequency

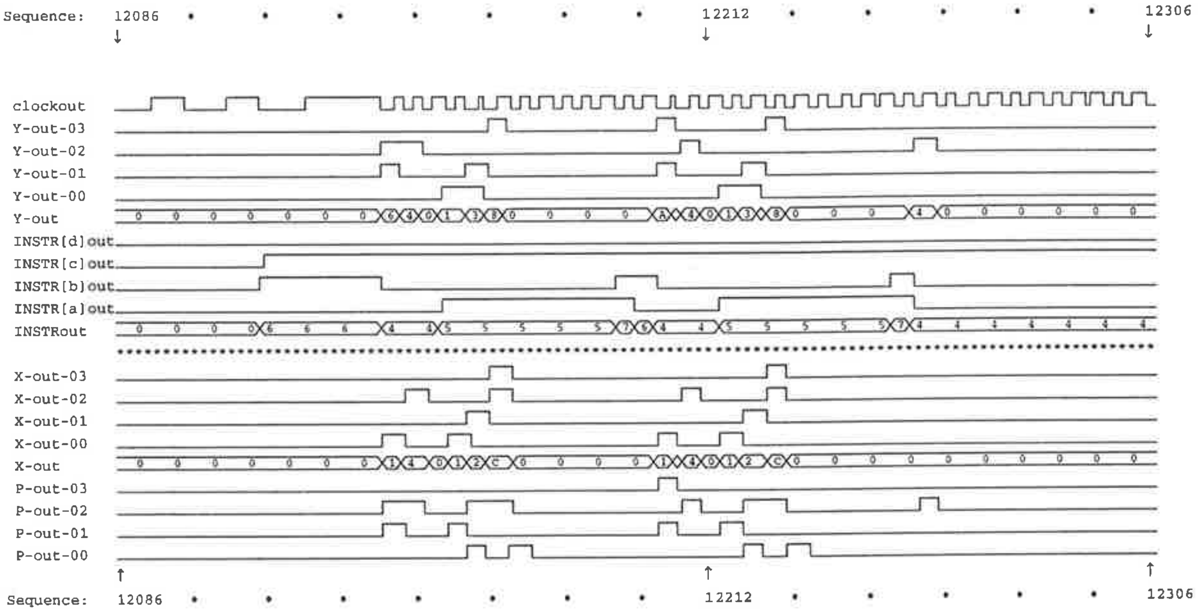


Figure 5.28: Test results of the PE in floating point denormalisation mode at 128MHz.

(*CKrate*, *CKs1* and *CKs2*) were connected as shown in Table 5.1 to achieve the desired output frequency. *CKstop* is the active high reset for the ring oscillator and was tied to logic low. *CKin* is the external clock input which was connected to the DAS pattern generation probe *clk* signal. The clock output (*CKout*) had no terminating resistor on the PCB but was connected to the Tektronix 7514 Storage Oscilloscope instead through a 50Ω coaxial line to the sampling head which has a 50Ω input impedance. The clock frequency was measured using a frequency meter.

Figure 5.32 shows the output of the clock generator set for 37.5*MHz* operation with a measured frequency of 30.12*MHz*. In most cases the frequency was stable to ±10*KHz*. There was feedthrough observed from the base clock frequency (482*MHz* measured) in the high output state with $Vdd = 1.89\,V$. Exactly eight cycles were observed indicating the ×16 clock was being fed through one of the multiplexer stages. Figures 5.34 and 5.35 show the variation of oscillator frequency with power supply voltage for a ring length of seven ($CKrate = 1$) and thirteen ($CKrate = 0$) gates, respectively (set $Vddp = 2.0\,V \pm 20mV$, $VREF = 0.7\,V \pm 20mV$, $Vhigh = 1.3\,V \pm 20mV$). Observed rise and fall times fall between 1.0 to 1.5ns. It was also noted that $Vdd > 1.75\,V$ produced feedthrough of the master clock in the logic high period of the output. This would indicate that the logic

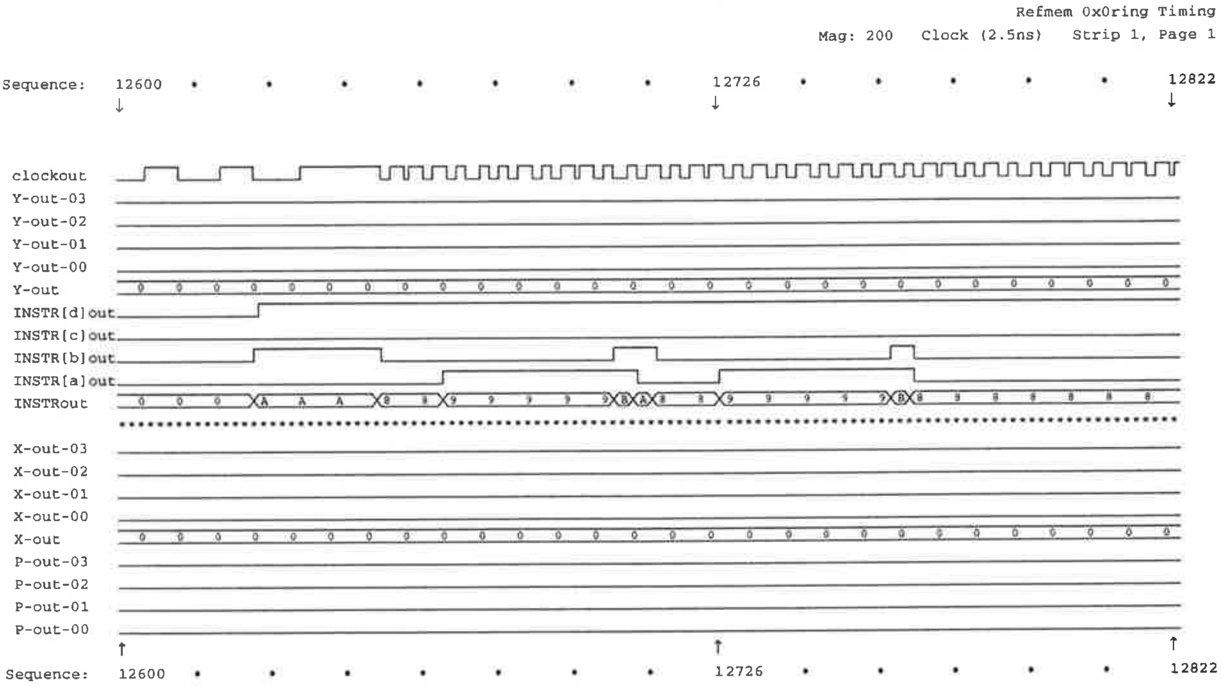


Figure 5.29a: Systolic ring operating in floating point multiplication mode at $91MHz$ where $X_{in} = 0 \times e^0$, $Y_{in} = 0 \times e^0$, $P_{out} = 0 \times e^0$.

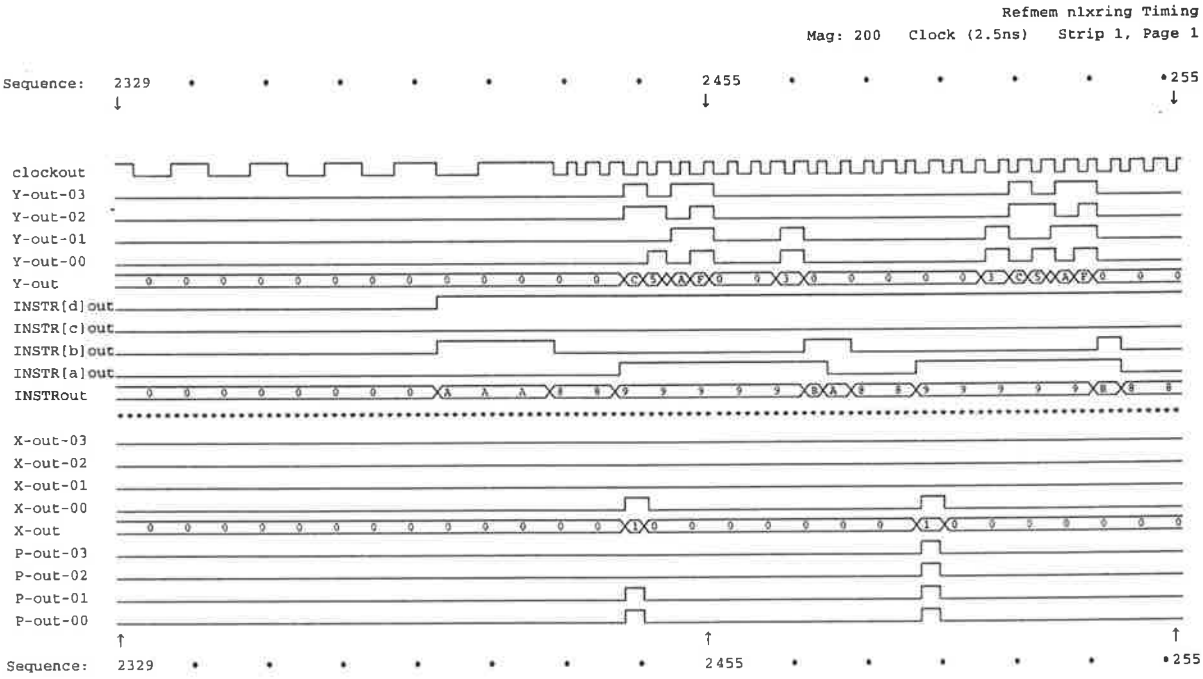


Figure 5.29b: Systolic ring operating in floating point multiplication mode at $91MHz$ where $X_{in} = 0.00000001 \times e^0$, $Y_{in} = 0.FA5C3000 \times e^0$ and $P_{out} = 0.0000000F \times e^0$.

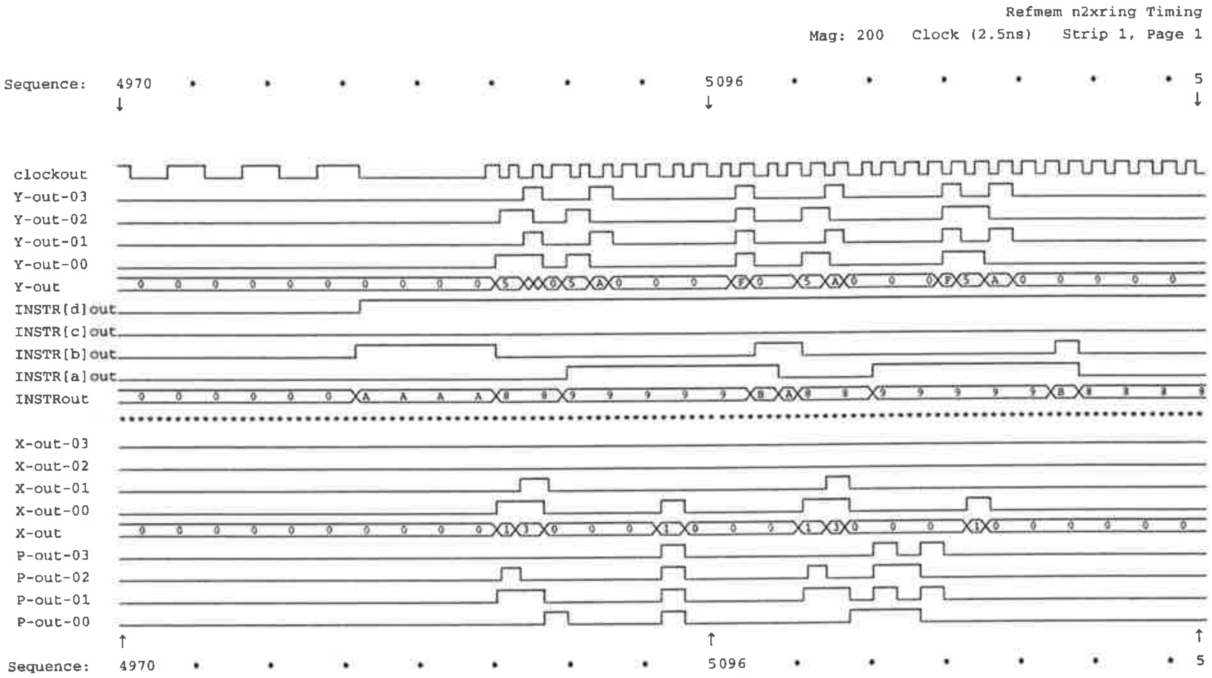


Figure 5.29c: Systolic ring operating in floating point multiplication mode at $91MHz$ where $X_{in} = 0.00010000 \times e^{031}$, $Y_{in} = 0.00A5F100 \times e^{0F5}$ and $P_{out} = 0.000000A5F \times e^{126}$.

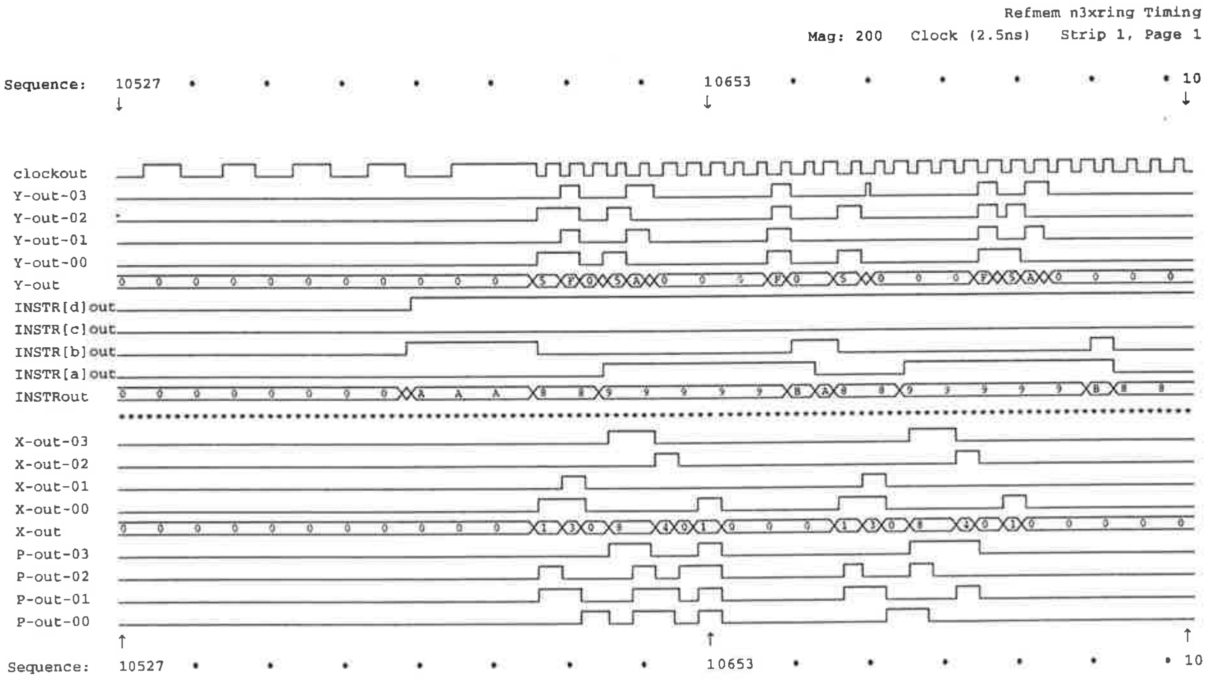


Figure 5.29d: Systolic ring operating in floating point multiplication mode at $91MHz$ where $X_{in} = 0.00010488 \times e^{031}$, $Y_{in} = 0.00A5F000 \times e^{0F5}$ and $P_{out} = 0.000000A8D \times e^{126}$.

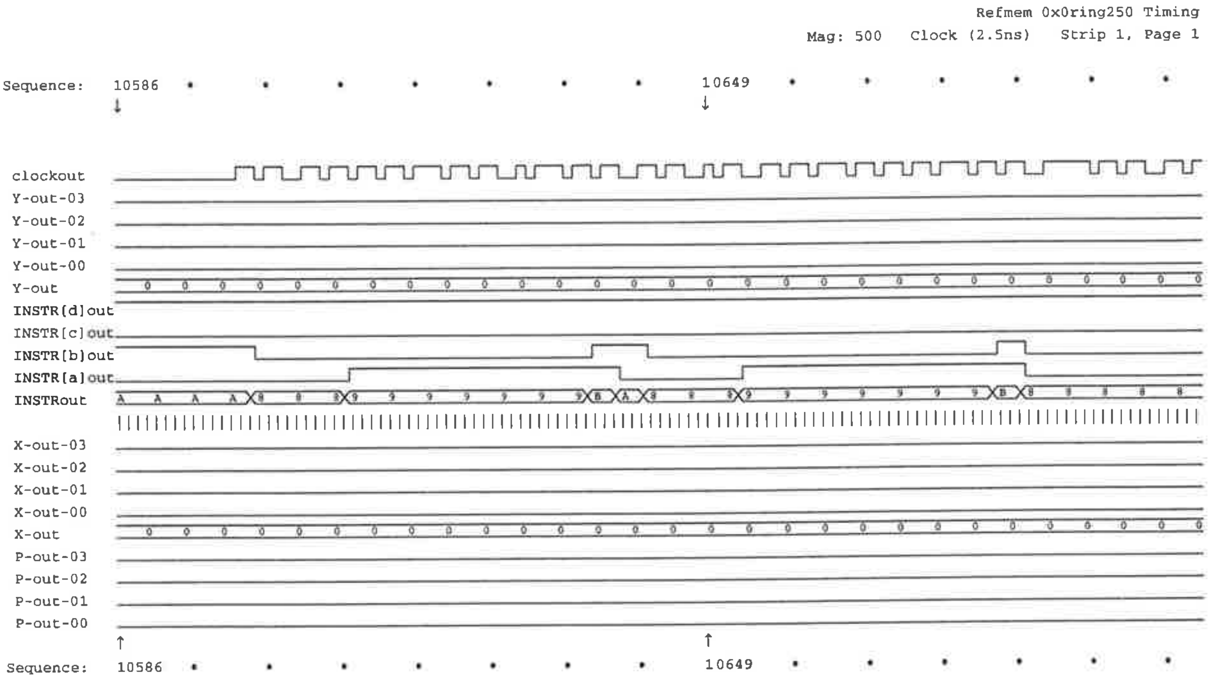


Figure 5.30a: Systolic ring operating in floating point multiplication mode at $128MHz$ where $X_{in} = 0 \times e^0$, $Y_{in} = 0 \times e^0$ and $P_{out} = 0 \times e^0$.

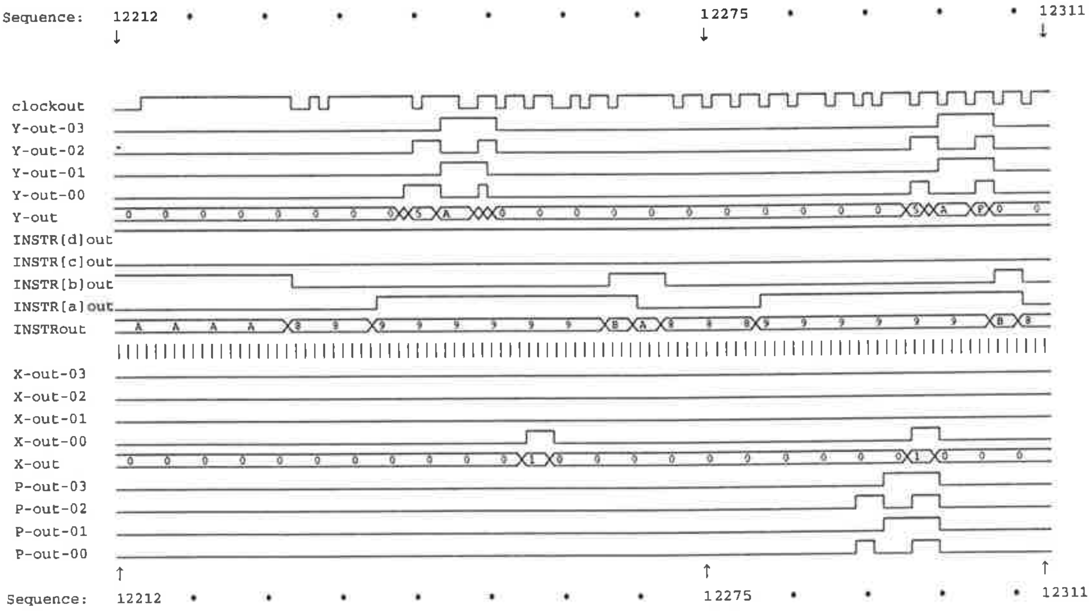


Figure 5.30b: Systolic ring operating in floating point multiplication mode at $128MHz$ where $X_{in} = 0.00100000 \times e^0$, $Y_{in} = 0.FA500000 \times e^0$ and $P_{out} = 0.000FA5000 \times e^0$.

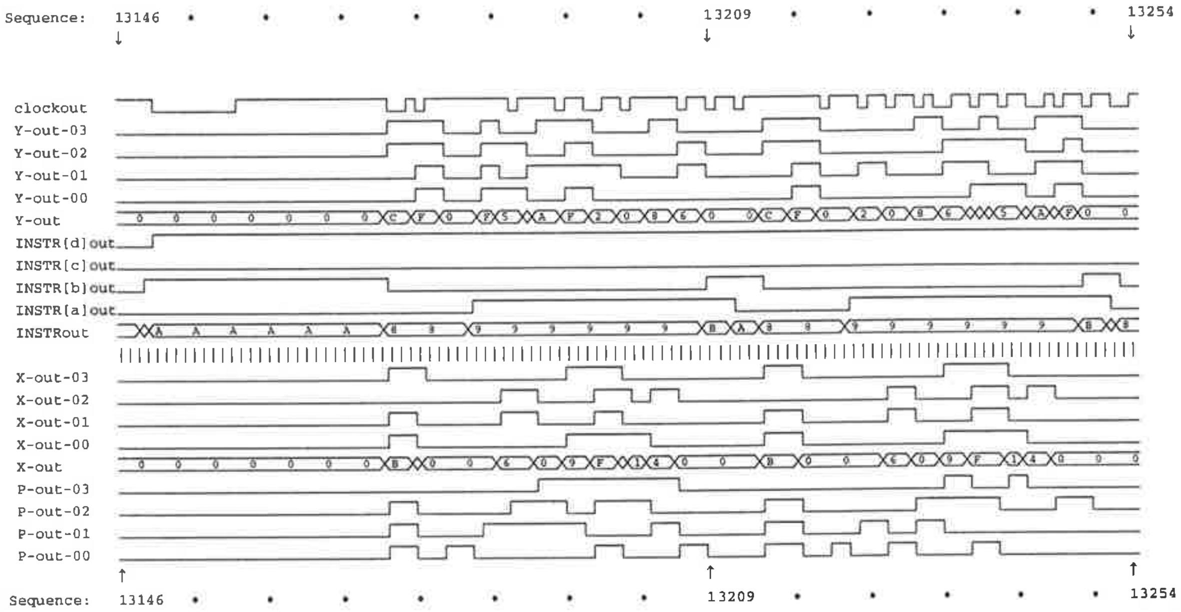


Figure 5.30c: Systolic ring operating in floating point multiplication mode at 128MHz where $X_{in} = 041F9060 \times e^{00B}$, $Y_{in} = FA5F6802 \times e^{0FC}$ and $P_{out} = 0.04085C612 \times e^{107}$

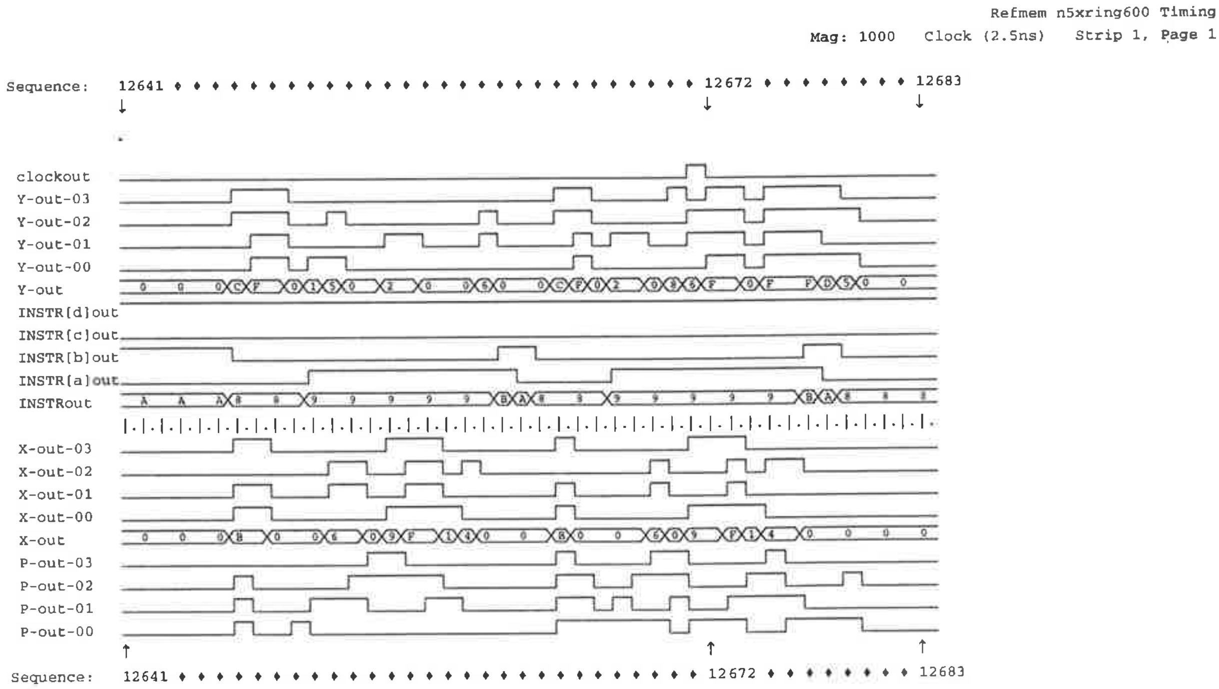


Figure 5.31: Test results of the PE in floating point multiplication mode at 350MHz.

signal	CKout	CKout	CKout	CKs1	CKs2	CKrate
process spread	*typical*	0.5 σ–*slow*	-			
source	Simulated	Simulated	Measured			
Vddc (*V*)	2.0 *V*	2.0 *V*	1.7 *V*			
temp. (°*C*)	75	75	75(est.)			
frequency	1000	763	747	0	0	1
(*MHz*)	250	191	187	0	1	1
	62.5	47.7	46.7	1	1	1
	600	461	441	0	0	0
	150	115	110	0	1	0
	37.5	28.9	27.3	1	1	0

Table 5.1: Simulated and observed clock frequencies for different clock rates.

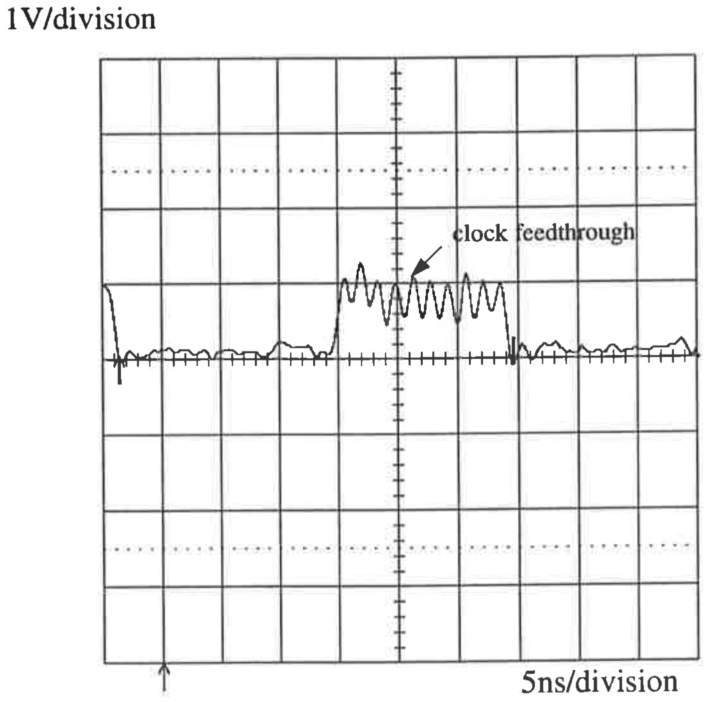


Figure 5.32: Clock generator set for 37.5*MHz* operation showing feedthrough from the source (*Vdd* = 1.89 *V*).

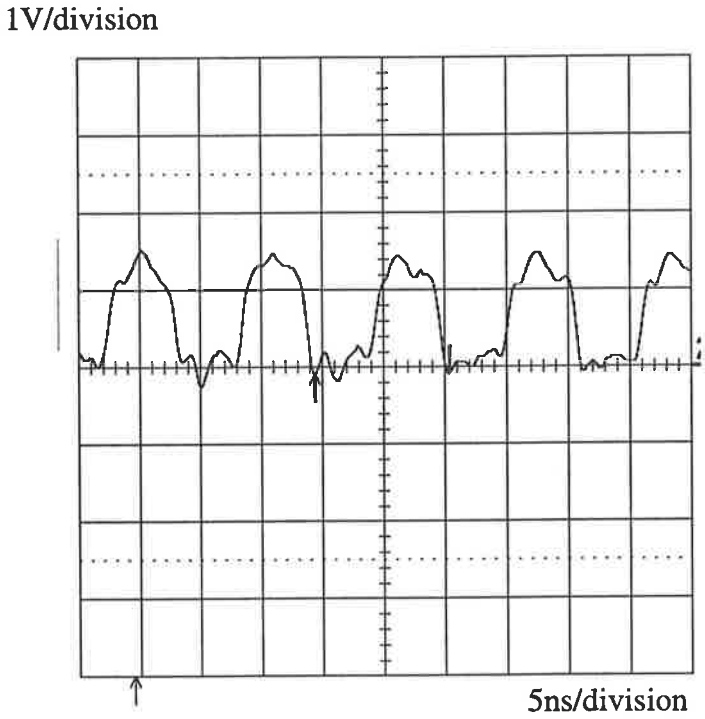


Figure 5.33: Clock generator set for 150*MHz* operation showing measured frequency of 91*MHz*.

threshold has just been reached in the multiplexer circuits of the clock generator, and is therefore unlikely that the rest of the chip would function correctly above 1.75 V. Figure

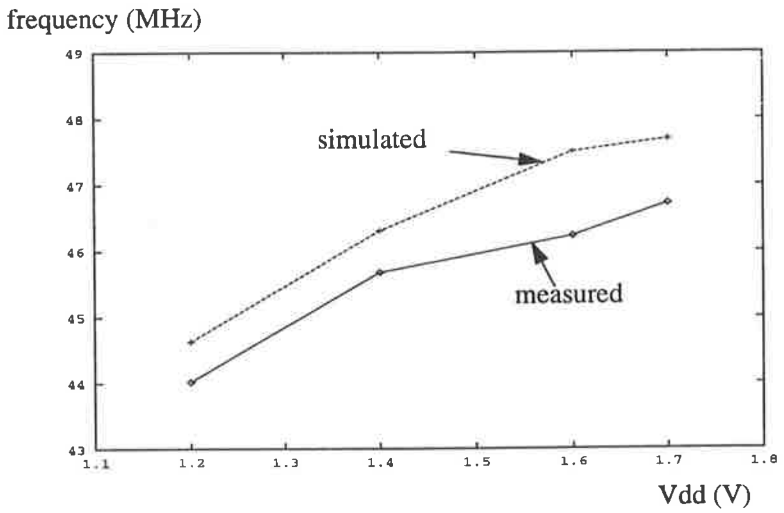


Figure 5.34: Variation of clock frequency with power supply voltage for $CKrate = 1$.

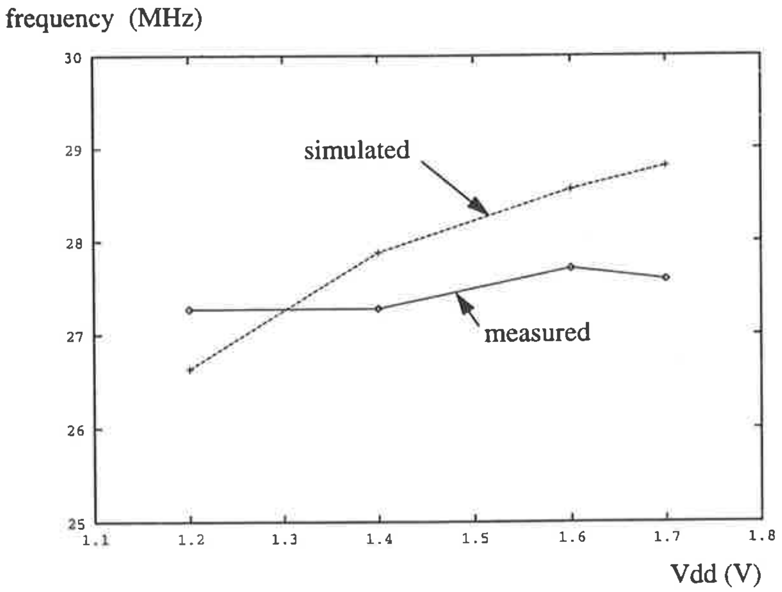


Figure 5.35: Variation of clock frequency with power supply voltage for $CKrate = 0$.

5.36 shows the variation of oscillator output amplitude (peak–peak) with pad power supply voltage ($Vddp$) for a ring length of seven ($CKrate$= 1) and thirteen ($CKrate$= 0) gates (set $CKs1 = 1$, $CKs2 = 1$, $VREF = 0.7\,V \pm 20mV$, $Vhigh = 1.3\,V \pm 20mV$).

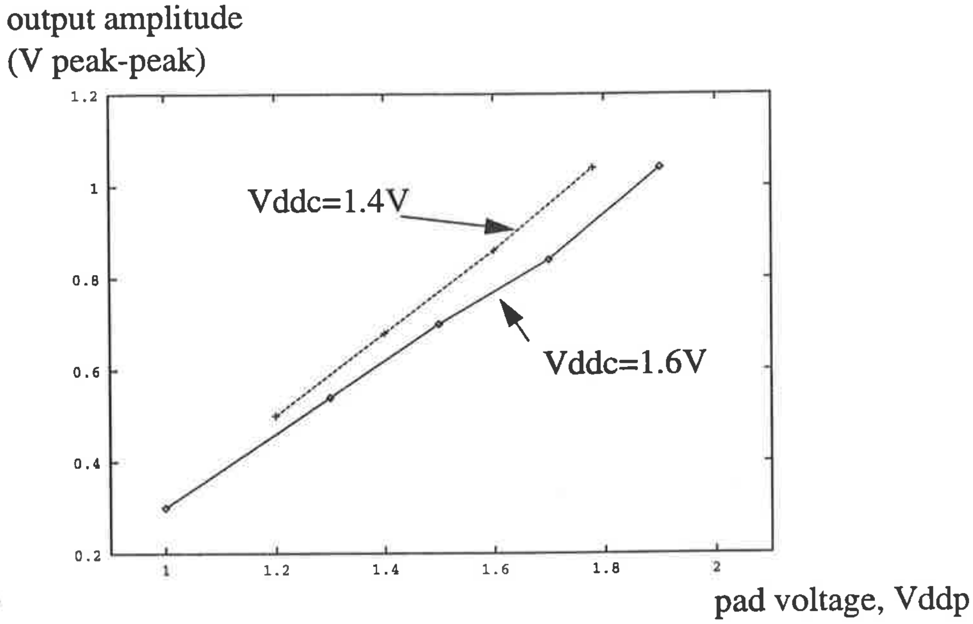


Figure 5.36: Variation of peak–peak output voltage with pad power supply voltage for $Vddc = 1.6\,V$, $CKrate = 1$ and $Vddc = 1.4\,V$, $CKrate = 0$.

Chapter 6

Discussion and Future Work

6.1 Discussion

This thesis has presented the design methodology, simulation, implementation and testing of a systolic ring floating point processing element (PE).

Gallium Arsenide (GaAs) was chosen as the technology for implementing the PE because of its speed and power advantages over silicon and to assess new architectures to make best use of the characteristics of GaAs. GaAs technology was studied in Chapter 2 including a review of MESFET and HEMT technology, large signal MESFET models, and MESFET logic classes. It was found that DCFL, SDCFL and SBFL may be mixed to provide a library of primitive cells from which circuits can be made. The logic classes were optimised for delay, area and noise margin to achieve both the smallest and fastest possible circuits. A set of design guidelines were established to build circuits. A *ring notation* layout style was presented which improves the performance, area and regularity of circuit structures for GaAs over the more tradition styles of CMOS design. An abstract design style (analogous to stick diagrams in CMOS design) was developed to aid the full custom layout. Circuit primitives included inverters, NOR gates, a 2–input OR gate and buffers to drive large fan–out loads. Circuit parasitics were investigated for GaAs circuits. The interconnect parasitics have a significant effect on circuit operation due to the fast transition times of the logic gates. These parasitics cause crosstalk, ringing and poor signal delays if not modelled properly. An electromagnetic field simulator, 'Raphael' was used to study circuit interconnect structures. Models for interconnects including transmission line (TL), and a lumped capacitance were simulated using HSPICE. It was found that for short wires ($< 600\mu m$) a lumped capacitor model may be used with little error to model the wire. For longer wires, a simulation using a more complex model such as a lossless TL should be used. To facilitate full custom GaAs design, some design tools were modified from their use in silicon design. A program 'ext2sp' was developed during the course of the research to correctly extract GaAs devices and parasitics from the layout for simulation in HSPICE. Technology files were further developed for use with 'MAGIC'. Models for parasitics of pads, bonding wires and package leads were investigated and their magnitude found from simulations.

A new PE architecture for integrated multiplication and accumulation of two floating point numbers was developed in Chapter 3. A digit–serial multiplication algorithm was presented and a simple digit–serial multiplier was designed for the case of 4–bits per digit. The digit–serial multiplier cell is a re–organised parallel multiplier which is pipelined and optimised for fast propagation through the critical path (critical path length is five full adder delays). A model for a systolic cell was presented which used the digit–serial multiplier and when systolic cells were placed in a linear array, they performed multiplication on two arbitrary precision floating point numbers. The systolic cell was then extended to perform the basic functions of floating point accumulation, namely denormalisation and addition. A ring of systolic cells and delay cells can perform these floating point operations and a range of architectures is possible which is variable in the precision of the operands, the number of bits per digit in the number representation and the number of systolic cells around the ring. To optimise this architecture for a MATRISC processor, it was necessary to develop a performance metric. The appropriate performance metric was shown to be *area* × *time* (*AT*) for these types of systolic array processors by minimising the total job time for a matrix product on a rectangular systolic array. The model was then evaluated for an IEEE extended single precision floating point number representation and two circulations of data in the PE using the Vitesse HGAAS–II (E/D MESFET) process as the target technology. The results show that an optimal implementation in the target GaAs technology consists of arithmetic units with four bits per digit and four systolic cells in the ring.

To build the physical layout of the PE chip, fast area efficient data flip–flops were designed based on the edge triggered 6-NOR flip–flop and adapted for use with DCFL and SDCFL. A toggle flip–flop was designed based on the data flip–flop. GaAs full adder circuits were investigated and a small AT metric implementation was chosen. Other circuits designed include a ring controller to control the I/O of operands from the PE, a flag checking circuit, multiplexers and a clock generator based on a variable length ring oscillator which can run up to $1GHz$ and has a selectable output from divider stages. A clock distribution system was designed based on the H–Tree approach to minimise clock skew to the 266 flip–flops on the chip. A design approach was developed for clocking synchronous rings of latches which found that the sum of the skew between all latches in a closed ring is zero. The clock distribution and buffer circuits were simulated using TL models to show that the skew between adjacent latches was less than $100ps$ to guarantee correct data transfer. The power circuit was designed to have less than a 5% fluctuation across the chip due to inductive spikes and ohmic losses and to be within safe limits for metal migration.

The chip was successfully fabricated in a GaAs $0.8\mu m$ E/D MESFET process by Thomson–CSF Semiconducteurs Specifiques, France on their first fabrication run with the HGAAS–II process licenced from Vitesse Semiconductor, USA. A micrograph of the chip is shown in Figure 6.1. The total chip size including pads and test structures is $3.1mm \times 5.8mm$ and includes $16,000$ devices. The dimensions of the processing element is $1.7mm \times 4.5mm$ giving an active area of $7.5mm^2$ with $12,000$ devices resulting in 1600 $devices/mm^2$. The chip was bonded into a 132/84 pin MLC package with a heat–sink. A test fixture was designed and constructed to facilitate testing of the PE chip. A $\frac{1}{64}$" thick Teflon PCB was designed with PCB pins connecting the pressure mounted chip to the tester. The GaAs I/O pads have signal transitions of well under $500ps$ with a $1.3\,V$ swing, so the PCB wires were modelled with source and terminating resistors to find the best response. Terminating resistors (68Ω) were used in the final test circuit and signal skew between lines was less than $350ps$.

The chips were tested and oscillator speeds were measured as a function of supply voltage. The chips tested were found to operate correctly except that most showed oscillator speeds corresponding to 0.5σ–*slow* simulations. The power dissipation was $1.5\,W$ at $Vdd = 1.5\,V$. Most chips would not work at $Vdd = 2\,V$ which was the designed supply voltage. This may be due to the process spread observed. The functionality of the chips was tested using a Tektronix DAS–9200. Input operands for testing were generated and programmed into the DAS and the results in Chapter 5 show the correct operation of the systolic cell and the complete systolic ring with operation at 128*MHz*. At this clock rate 5*Mflops* was achieved. Synchronous operation was shown for a 350*MHz* clock rate except that the result was incorrect due to failure of the critical path, however the X operand was fully recovered from the ring. For *typical* process parameters, a maximum clock speed above 300*MHz* would be expected with a corresponding computation rate of 11*Mflops* for the PE chip.

The PE is a computation node in a two dimensional mesh connected systolic array which forms part of a proposed MATRISC processor to perform matrix operations. The simulated performance of such a device when executing matrix problems is in the range of *Gflops* which is well in excess of the capabilities of current generation engineering workstations.. The MATRISC processor closely follows the RISC philosophy of providing a smaller set of commonly executed hardware operations. It is proposed that the matrix hardware extension be integrated into a RISC processor system.

This thesis has shown that high performance computing components can be implemented using GaAs technology if proper consideration is given to the characteristics of the technology to produce optimal processor architectures.

The development of GaAs layout strategies and systolic ring architectures for the PE has been published previously [BeMa91]. The design of the systolic matrix processor and the optimisation of the PE architecture for a target technology has been reported elsewhere [MaBe92a, MaBe92]. The work on the design, layout and simulation of the PE chip has been published in reference [BeMa93]. Finally, the design, simulation and testing of the PE chip is reported in reference [BeMa95].

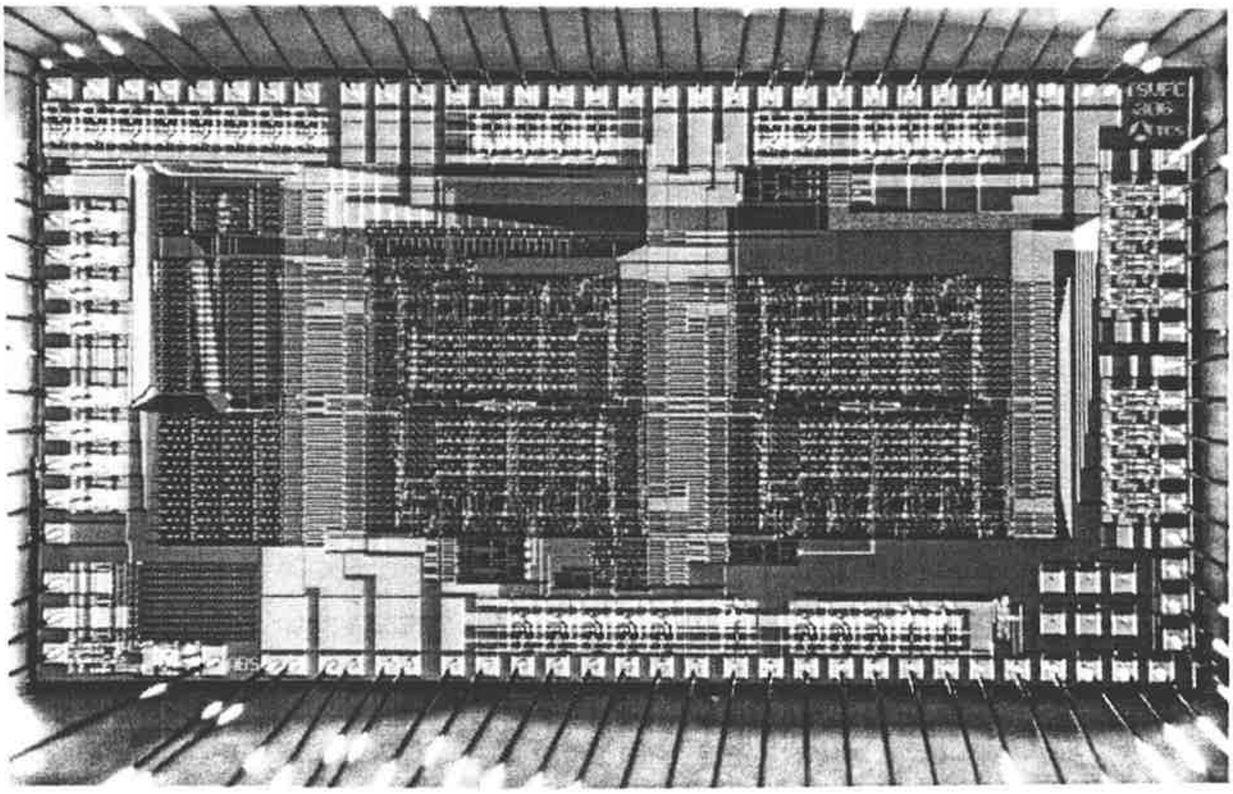


Figure 6.1: Micrograph of the fabricated GaAs systolic PE chip.

6.2 Future Work

Following a top–down methodology for future work:

- A complete MATRISC system needs to be thoroughly studied and simulated. Such a study must take into account the types of algorithms that suffer large performance penalties when executed on conventional computer systems. The optimal MATRISC architecture should also take into account current memory speeds and sizes to provide adequate bandwidth for such a processor. Additional components such as caches will also improve system performance. Design tools such as VHDL may be used for the system simulation task, however, *real* data on the size and speed of the components of the MATRISC system will need to be studied possibly by building some components.

- By breaking down the MATRISC system into components and studying each part to determine a method for improving processor performance, parts such as memories, scalar processors and caches are best found as "off the shelf" items. The following require special design to gain optimum system performance:

  - Processing elements do the computation work in the systolic array. They must be both small in chip area and fast in execution speed. They must also operate on multi–precision data, such as double precision floating point. Ideally, the PEs should be fault tolerant and have a simple built–in self test mechanism so faulty PEs can be bypassed or replaced. General future trends for PEs of this type will be an increase in complexity, reconfigurability (multi-precision and format), faster and more of them integrated onto a single chip to allow larger arrays to be built.
  - Buses link the system together and must provide an efficient way of transferring data to maximise throughput for a variety of matrix algorithms. The maximum allowable pin density and data transfer speed determine the bus speeds and hence the system bandwidth and overall performance.
  - A possible solution for the memory is 'Rambus' [Ramb93] which provides multiple $500 Mbytes/s$ channels which may meet the bandwidth requirements for a MATRISC processor.
  - System integration is a significant problem when designing large systems with chips fabricated using different technologies such as memories, PEs and caches. A fine–line PCB solution can be used at the top level but due to the high data transfer rates (up to $500 MHz$) between chips, a multi–chip module technology should be used. This improves both the density and hence the execution speed of the system.

- Better logic families for faster circuit operation with higher levels of integration and low power are needed. Complex gate structures in GaAs may provide an alternative solution to the AT metric studied in this thesis.

- The currently available technology and integration techniques drive the possible range of architectural solutions in computer design. GaAs was investigated because of its advantages in both speed and power over silicon technology. The solution may lie in a different technology in the future as processes improve, integration levels become higher and new logic classes are investigated. Motorola has announced a complementary GaAs process, 'CGaAs' which holds much promise for the future of high speed technologies. The largest deficiency of MESFET technology is the poor integration level when compared to a similar gate length CMOS technology. CGaAs has an integration level similar to that of CMOS and is claimed to be faster than DCFL using MESFETs. The process is also simpler than CMOS with no substrate contacts required which saves chip area. This would make CGaAs an attractive alternative for future processor design.

Appendix A : GaAs Digital Logic Performance Specifications

Measurements are carried out on the middle gate in a chain of three identical gates so the input and output to the gate under observation are realistic. Voltage swing was measured as the difference in output voltage in the static logic low and the static logic high state. The speed or delay of a logic gate was measured from the time at 50% of the input voltage swing to 50% of the output voltage swing in response to an input with a voltage swing and slew rate the same as the output. The rise or fall time of the gate was measured as the time for the signal to rise or fall from 20% to 80% of the output voltage swing. This differs from conventional CMOS which takes 10% to 90% of the output voltage swing. This is because GaAs has a much smaller voltage swing and the relative noise in each logic state is higher that for CMOS. These characteristics are shown in Figure A.2. A

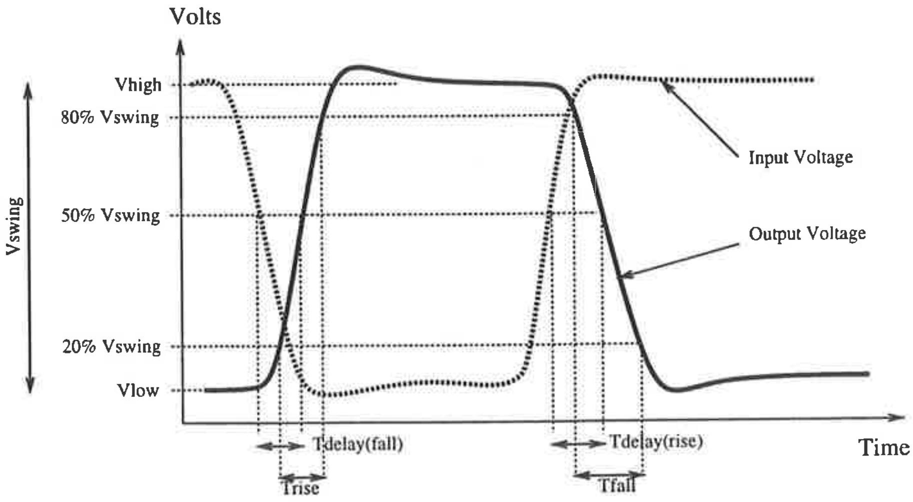


Figure A.2: Voltage swing, delay, rise and fall time measurements.

discussion of noise margins can be found in [Hill86, Lohs79, Haus93] and [Wing90] has a discussion relevant to GaAs.

Noise margin for a logic gate is defined as the maximum amount of noise applied to the input in each logic state while the output remains in the correct state. There are static and dynamic noise margins for any gate but here we only consider the static noise margins since robust operation under static conditions guarantees the dynamic operation [Lohs79]. The high (NM_H) and low (NM_L) static noise margins with reference to Figure A.3 are:

$$NM_H = Voh - Vih$$

$$NM_L = Vil - Vol$$

A negative noise margin indicates the logic gate will not settle into that logic state. There are five ways to measure the static noise margin of a logic gate as defined in [Haus93]:

- NSC (negative slope criteria) selects the unity gain point in the gate input-output voltage characteristics as the switching point where the gate moves from a logic state to a metastable state. This may be calculated mathematically which can be useful.

- MSC (maximum sum criteria) of $NM_H + NM_L$ is identical to the NSC method for most transfer characteristics but may predict a zero for one of the noise margins since it is not concerned with the individual values.

- MNSC (modified negative slope method) is used in many textbooks and leads to unconservative results for noise margin and has a poor theoretical basis [Haus93] and therefore is not used.

- MEC (maximum equal criteria) or mirror and maximum square method [HiLa86] constrains the high and low noise margins of the gate to be equal ($NM_H = NM_L$) and produces the worst case equal noise margin. This may be too restrictive on the optimisation of a logic gate and gives a more average result of high and low noise margins.

- MPC (maximum product criteria) maximises the area of a rectangle and hence maximises $NM_H \times NM_L$ and is the preferred method.

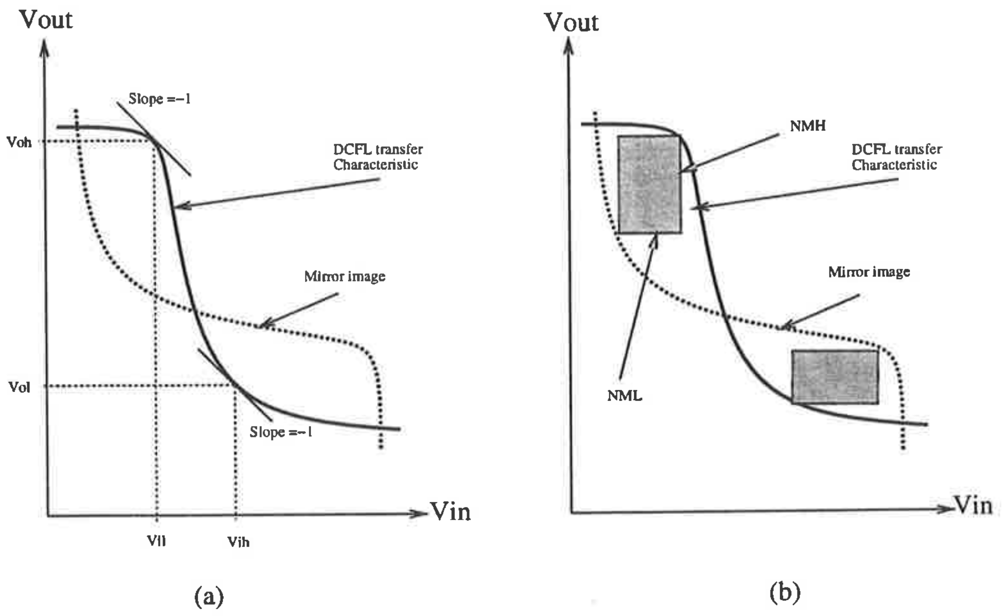


Figure A.3: Noise margin measurement methods using (a) NSC and (b) MPC or MEC if the rectangle becomes square.

Only the last two techniques, namely MEC and MPC give valid results for a wide range of gate transfer curves. The DCFL transfer curves shown in Figure A.3 are quite symmetric. For this case we can use either the NSC, MPC or MEC methods which would all give similar results. Noise margin was measured using the gain method where the points have unity gain. This means that the transition point from logic high to the metastable state is where the rate of change of output divided by input is unity.

Appendix B : PE Chip Pin Allocation

The following is the package pin to signal allocation for the PE chip and a description of power supply signals. The pins are numbered clockwise from the angled corner of the package. Note that when the package is mounted upside–down, the numbers run anticlockwise when viewed from the top. Key to symbols:

NC – not connected

Vdd – Pad and circuit power +1.5 V (decoupled to GND)

Vddp – Pad power +2 V

Vddc – Circuit power +1.5 V

GNDp – Pad ground 0 V

GNDc – Circuit ground 0 V

GND – Package ground plane 0 V (decoupled to Vdd)

VREF – Input pad reference supply +0.7 V

TSD – Depletion mode MESFET test structure

TSE – Enhancement mode MESFET test structure

TSFF – Data flip–flop test structure

TSINV – DCFL inverter gates test structure

CK – Clock generation circuit

Package Pin	Signal Name	Function	Package Pin	Signal Name	Function
1	PWR1	Vddp	67	PWR1	Vddp
2	s1	TSFFVREF	68	s43	cinb
3	s2	TSFFVddc	69	s44	dinb
4	GND	GND	70	GND	GND
5	s3	NC	71	s45	Vddc
6	s4	TSFFqbar	72	s46	Vddc
7	GND	GND	73	GND	GND
8	s5	TSFFq	74	s47	NC
9	s6	TSFFclear	75	s48	NC
10	GND	GND	76	GND	GND
11	s7	TSFFdbar	77	s49	dout
12	s8	TSFFclock	78	s50	cout
13	GND	GND	79	GND	GND
14	s9	TSFFVddp	80	s51	bout
15	s10	CKout	81	s52	aout
16	GND	GND	82	GND	GND
17	s11	CKVREF	83	s53	GNDc
18	GND	GND	84	GND	GND
19	s12	CKin	85	s54	NC
20	s13	CKs2	86	s55	NC
21	GND	GND	87	GND	GND
22	s14	CKs1	88	s56	RESETbar
23	s15	CKstop	89	s57	SELO
24	GND	GND	90	GND	GND
25	s16	CKrate	91	s58	NC
26	s17	CKVddp	92	s59	y3out
27	GND	GND	93	GND	GND
28	s18	NC	94	s60	y2out
29	s19	Vddc	95	s61	y1out
30	GND	GND	96	GND	GND
31	s20	Vddc	97	s62	y0out
32	s21	NC	98	s63	Vddp
33	PWR2	Vddp (*)	99	PWR2	Vddp (*)
34	PWR2	Vddp (*)	100	PWR2	Vddp (*)
35	s22	NC	101	s64	NC
36	s23	NC	102	s65	NC
37	GND	GND	103	GND	GND
38	s24	NC	104	s66	Vddc
39	s25	TSINVin	105	s67	Vddc

Table B.1: Assignment of pins to signals, power and ground.

Package Pin	Signal Name	Function	Package Pin	Signal Name	Function
40	GND	GND	106	GND	GND
41	s26	TSINVout	107	s68	NC
42	s27	TSINVVdd	108	s69	x0out
43	GND	GND	109	GND	GND
44	s28	VREF	110	s70	x1out
45	s29	x3inb	111	s71	x2out
46	GND	GND	112	GND	GND
47	s30	p3inb	113	s72	x3out
48	s31	x2inb	114	s73	GNDp
49	GND	GND	115	GND	GND
50	s32	p2inb	116	s74	p0out
51	GND	GND	117	GND	GND
52	s33	x1inb	118	s75	p1out
53	s34	p1inb	119	s76	p2out
54	GND	GND	120	GND	GND
55	s35	x0inb	121	s77	p3out
56	s36	p0inb	122	s78	Vddp
57	GND	GND	123	GND	GND
58	s37	y0inb	124	s79	TSDdrain
59	s38	y1inb	125	s80	TSDgate
60	GND	GND	126	GND	GND
61	s39	y2inb	127	s81	TSDsource
62	s40	y3inb	128	s82	TSEdrain
63	GND	GND	129	GND	GND
64	s41	ainb	130	s83	TSEgate
65	s42	binb	131	s84	TSEsource
66	PWR1	Vddp	132	PWR1	Vddp

Table B.2: Assignment of pins to signals, power and ground (cont.).

(*)- PWR2 was connected to Vdd, see chapter 5, Fix 2.

Appendix C : Test Equipment Set–up

The following is a description of the test environment set–up and equipment used for testing the PE chip.

Power Supply Connections and Sequencing

All pad power supply circuits were driven separately from the logic circuit supply to minimise noise and ground bounce. In addition separate high logic level and pad voltage references were used. The nominal circuit supply voltage (Vdd, $Vddc$) is $1.5\,V$ ($\pm5\%$) and the pad power supply ($Vddp$) should be set to $2.0\,V$ ($\pm20mV$). The input pads require a reference voltage ($VREF$) of $0.7\,V$ and the high logic level reference should be set to $1.3\,V$ ($\pm20mV$). The total current drawn from all power supplies is around $1A$.

The circuit power supply voltage should not be higher than 50% above its nominal value for any long period of time since excessive power dissipation may damage the chip. The input and output pads are single ended and no differential signals are used and each output pad is of the open–source type and can sink up to $27mA$ of current. The signal levels on *all* pads are 1.0 to $1.3\,V$ for logic high and $0\,V$ for logic low. All ground connections to power supplies were connected to a common point on the circuit chassis to prevent current loops. All power supply leads were shielded to prevent electromagnetic coupling with the shields were connected back to the test jig chassis ground. The chip was configured as follows (with reference to the pin allocation in Appendix B):

- all pins marked GND have a shorting PCB jumper connected to the adjacent grounded pin

- pins 17 and 44 are connected to $VREF$ supply
- pins 1, 26, 66, 67, 98, 122, 132 are connected to the pad power supply ($Vddp$)
- pins 29, 31, 33, 34, 71, 72, 99, 100, 104, 105 are connected to the circuit power supply

Before the power for ANY power supply was turned on, the output voltage was set to zero and then turned up to the correct voltage to avoid spikes from the power supply entering the circuit. The input pad reference voltage supply was switched on first, then the pad supply, the high logic level reference supply and finally the circuit supply.

Test Equipment

DAS 9200 High Speed Digital Tester

The Tektronix Digital Analysis System (DAS) 9200 is a tool which provides the operating environment for pattern generator modules, data acquisition modules, and the software to control them. The software is configured to run via a host system using an X–windows interface. The software allows programming of the pattern generation modules with test vectors after which, the program can be run which sends signals to the chip under test. The resulting signals read on the data acquisition probes (the chip outputs) can then be displayed.

Pattern Generation Module

The 92S16 pattern generator module connects the two P6464 pattern generator pods to the DAS. The pods are labelled $A0$ and $A1$. The P6464 provides a total of nine signal lines (bits 0–8) and a clock line (clk). The maximum clock frequency available is $50MHz$ and the P6464 can supply signals at either TTL or ECL levels. The probe tips should be directly connected to the PCB gold connectors with the signal label (white) on the innermost pin and the reference (black label) on the outermost pin. The specifications of the 92S16 pattern generator are shown in Table C.3. The P6464 receives power via three sense leads connected to the probe: a red line for V_H (voltage high), a black line for V_L (voltage low) and a green line for ground. The three power wires were connected

Characteristic	Specification
Maximum Clock Frequency (period)	$50 MHz$ ($20 ns$)
High Voltage (V_H)	$-0.5V$ to $+5.5V$ @ $100mA + I_{LOAD}$
Low Voltage (V_L)	$+0.3V$ to $-5.5V$ @ $100mA + I_{LOAD}$
$V_H - V_L$	$4.8V$ to $5.2V$
TTL V_L Out	$V_L + 0.8V$
TTL V_H Out	$V_H - 1.1V$
ECL V_L Out	$V_H - 1.75V$
ECL V_H Out	$V_H - 1V$
Current Capability	$20mA$ (sink or source)

Table C.3: DAS 92S16 Pattern Generator Specifications.

for TTL output as follows: $V_H = +2.4\,V$, $V_L = -2.6\,V$ and the green wire was connected to the test jig chassis ground. There must be a connection between the power supply ground and the circuit chassis ground. This provides a logic high of $1.3\,V$ and a logic low of $-2.0\,V$ at the probe tip. A negative logic low will not affect the chip operation since it must only be below the logic threshold to be off. ECL logic levels could not be used since the voltage swing was too small ($0.8\,V$).

Data Acquisition Module

The 92A96 data acquisition module was used in 24 channel high speed acquisition mode with a resolution of $2.5ns$. The signal side of the probe tips is marked with a colour and should point towards the chip.

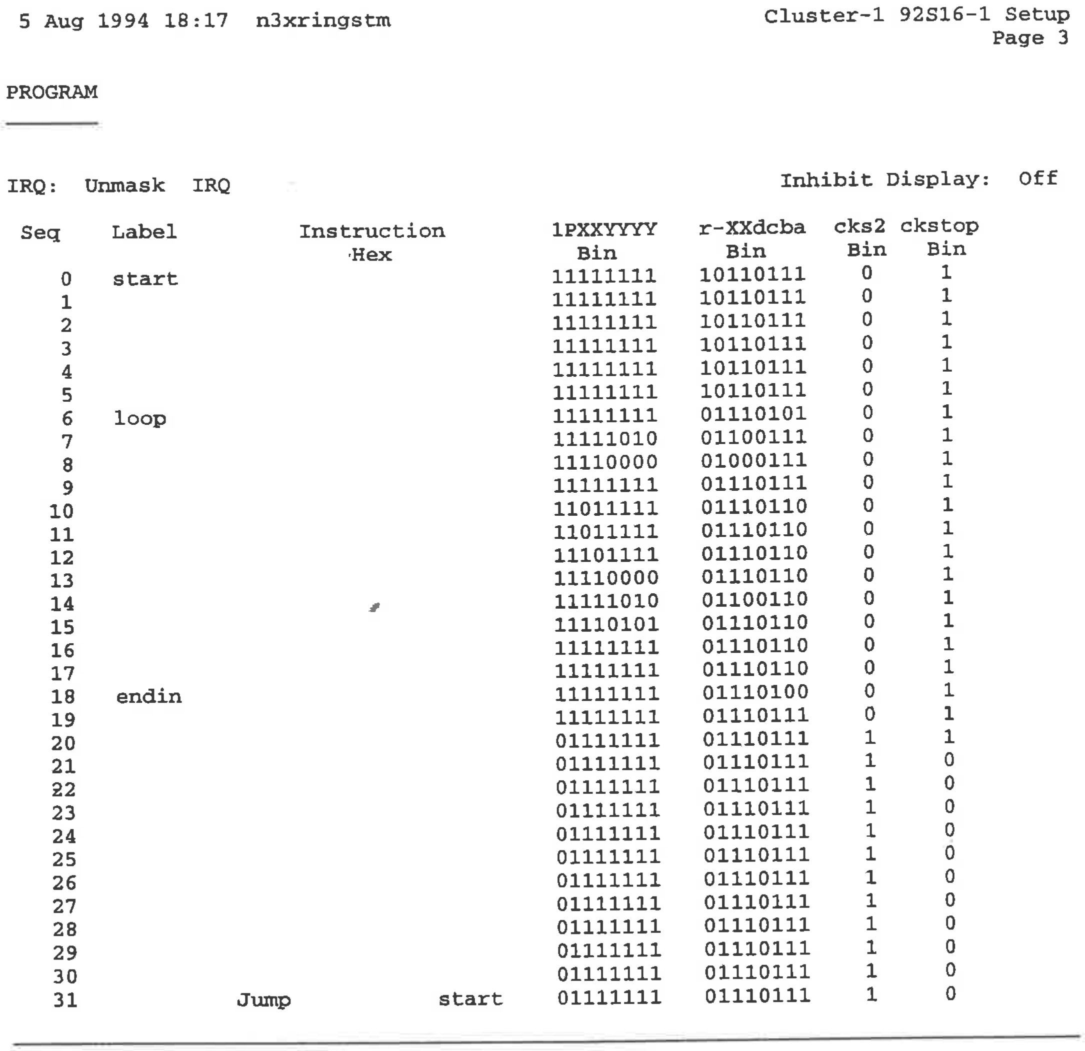
DAS probe allocation

The DAS was configured through the software interface as follows:

- define a cluster ('Sys–Config' menu) of the 92A96–1 and 92S16–1 modules
- under the 'Cluster Setup' menu, select run and start modes as normal and the stop mode as manual to allow debugging of the test head while the DAS keeps running
- 92S16–1 module:
  - under the 'Config' menu, select the clock as internal, $20ns$ which is the fastest rate.

Pattern Generator Pod	Pin Number	Signal
7A-0	58	Y0inb
7A-1	59	Y1inb
7A-2	61	Y2inb
7A-3	62	Y3inb
7A-4	48	X2inb
7A-5	45	X3inb
7A-6	56	P0inb
7A-7	53	P1inb
7A-8	50	P2inb
7A-clk	-	Not Connected
7B-0	64	ainb
7B-1	65	binb
7B-2	68	cinb
7B-3	69	dinb
7B-4	55	X0inb
7B-5	52	X1inb
7B-6	88	RESETbar
7B-7	89	SELO
7B-8	47	P3inb
7B-clk	19	CKin
Acquisition Probe	Pin Number	Signal
A0-0	81	aout
A0-1	80	bout
A0-2	78	cout
A0-3	77	dout
A0-4	97	Y0out
A0-5	95	Y1out
A0-6	94	Y2out
A0-7	92	Y3out
A1-0	108	X0out
A1-1	110	X1out
A1-2	111	X2out
A1-3	113	X3out
A1-4	116	P0out
A1-5	118	P1out
A1-6	119	P2out
A1-7	121	P3out
C0-0	15	CKout

Table C.4: DAS probe allocation.



```
 5 Aug 1994 18:17   n3xringstm                              Cluster-1 92S16-1 Setup
                                                                              Page 3

PROGRAM
_______

IRQ:  Unmask  IRQ                                          Inhibit Display:   Off

 Seq    Label          Instruction          1PXXYYYY    r-XXdcba   cks2  ckstop
                          Hex                  Bin         Bin      Bin    Bin
   0    start                               11111111    10110111    0      1
   1                                        11111111    10110111    0      1
   2                                        11111111    10110111    0      1
   3                                        11111111    10110111    0      1
   4                                        11111111    10110111    0      1
   5                                        11111111    10110111    0      1
   6    loop                                11111111    01110101    0      1
   7                                        11111010    01100111    0      1
   8                                        11110000    01000111    0      1
   9                                        11111111    01110111    0      1
  10                                        11011111    01110110    0      1
  11                                        11011111    01110110    0      1
  12                                        11101111    01110110    0      1
  13                                        11110000    01110110    0      1
  14                                        11111010    01100110    0      1
  15                                        11110101    01110110    0      1
  16                                        11111111    01110110    0      1
  17                                        11111111    01110110    0      1
  18    endin                               11111111    01110100    0      1
  19                                        11111111    01110111    0      1
  20                                        01111111    01110111    1      1
  21                                        01111111    01110111    1      0
  22                                        01111111    01110111    1      0
  23                                        01111111    01110111    1      0
  24                                        01111111    01110111    1      0
  25                                        01111111    01110111    1      0
  26                                        01111111    01110111    1      0
  27                                        01111111    01110111    1      0
  28                                        01111111    01110111    1      0
  29                                        01111111    01110111    1      0
  30                                        01111111    01110111    1      0
  31             Jump            start      01111111    01110111    1      0
```

Figure C.4: DAS test vector program to test the systolic ring floating point multiplication.

Bibliography

[AnAr87] M. Annaratone, E. Arnould, T. Gross, H.T. Kung, M. Lam, O. Menzilcioglu and J.A. Webb. "The Warp Computer: Architecture, Implementation and Performance". *IEEE Transactions on Computers*, C-36(12) pp. 1523–1538, December 1987.

[AwTa93] M. Awaga and H. Takahashi. "The mVP 64-Bit Vector Coprocessor: A New Implementation of High-Performance Numerical Computation". *IEEE Micro*, pp. 24–36, October 1993.

[Bako90] H. Bakoglu. *Circuits, Interconnections and Packaging for VLSI.* Addison-Wesley, 1990.

[Beau91] A. Beaumont-Smith. "GAASNET V2.0 - A gallium arsenide network extractor". *Integrated Silicon Design Pty.Ltd.*, Adelaide, 1991.

[Beau92] A. Beaumont-Smith. "EXT2HSP - A conversion program from MAGIC to HSPICE for GaAs circuits". *The University of Adelaide*, Adelaide, 1992.

[Beau93] A. Beaumont-Smith. "SCFL circuit design and Designer Interface for InGaAs/AlGaAs HEMT". *Seoul National University Report*, Department of Electronics Engineering, July 1993.

[BeMa91] A. Beaumont-Smith, W. Marwood, C.C. Lim and K. Eshraghian. "Ultra High Speed Gallium Arsenide Systems: Design Methodology, CAD tools and Architecture". *Proc. Microelectronics '91, I.E.Aust Conference*, pp. 85–90, June 1991.

[BeMa93] A. Beaumont-Smith, W. Marwood, K. Eshraghian and C.C. Lim. "The Gallium Arsenide Implementation of a Systolic Floating Point Processing Element". *Proc. 12th Australian Microelectronics Conference*, pp. 255–260, October 1993.

[BeMa95] A. Beaumont-Smith, W. Marwood, C.C. Lim and K. Eshraghian. "Design and Implementation of a GaAs Systolic Floating Point Processing Element". *Submitted to IEE Proceedings-E, Computers and Digital Techniques*, 1995.

[BeMa95a] A. Beaumont-Smith, W. Marwood and C.C. Lim. "A CMOS Linear Systolic Processing Element". *Proc. 13th Australian Microelectronics Conference*, pp. 74–79, July 1995.

[Ber91] M. Berroth, V. Hurm, U. Nowotny, A. Hulsmann, G. Kaufel, K. Kohler, B. Raynor and Jo. Schneider. "A 2.5nS 8x8-b Parallel Multiplier Using 0.5μm GaAs/AlGaAs Heterostructure Field Effect Transistors". *Microelectronic Engineering 15*, Elsevier Science Publishers B.V.,pp. 327–330, 1991.

[Brau63] E.L. Braun. *Digital Computer Design, Logic, Circuitry, and Synthesis.* Academic Press, 1963.

[BrBa92] R.B. Brown, P. Barker, A. Chandna, T.R. Huff, A.I. Kayssi, R.J. Lomax, T.N. Mudge, D. Nagle, K.A. Sakallah, P.J. Sherhart, R. Uhlig, and M. Upton. "GaAs RISC Processors", *Proc. IEEE GaAs IC Symposium*, pp. 81–84, 1992.

[ClCl92] A.P. Clarke, R.J. Clarke, I.A. Curtis and W. Marwood. "A Floating Point Matrix Arithmetic Processor: An Implementation of the SCAP Concept". *Proc. APCCAS '92, IEEE, IREE and IEAust Asia-Pacific Conference on Circuits and Systems*, December 1992.

[CoHa92] P. Corbett and R. Hartley. "Designing Systolic Arrays Using Digit-Serial Arithmetic". *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, Vol. 39, No. 1, January 1992.

[Curt80] W. Curtice. "A MESFET Model for Use in the Design of GaAs Integrated Circuits". *IEEE Transactions on Microwave Theory and Techniques*, Vol. MTT-28, No. 5, May 1980.

[DeRe85] P.B. Denyer and D. Renshaw. *VLSI Signal Processing: A Bit-Serial Approach.* Addison-Wesley, England, 1985.

[DoFr93] D.A. Doane and P.D. Franzon (Editors). *Multichip Module Technologies and Alternatives - The Basics.* Van Nostrand Reinhold, New York, 1993.

[Dyks90] J.A. Dykstra. "High-Speed Microprocessor Design with Gallium Arsenide Very Large Scale Integrated Digital Circuits". Ph.D. Thesis, The University of Michigan, 1990.

[Eshr91] K. Eshraghian. "Fundamentals of Very High Speed Systems: Gallium Arsenide VLSI Technology Course Notes". *Centre for GaAs VLSI Technology*, The University of Adelaide, South Australia, 1991.

[Eshr91a] K. Eshraghian, R. Sarmiento, P.P. Carballo and A. Nunez. "Speed-area-power optimization for DCFL and SDCFL class of logic using ring notation". *Microprocessing and Microprogramming*, 32, (1–5), pp. 75–82, 1991.

[FiKu83] A.L. Allan, H.T. Kung, L.M. Monier, H. Walker and D. Yasunori. "Design of the PSC: A Programmable Systolic Chip". *Third Caltech Conference on Vrey Large Scale Integration*, Pasadena, CA, USA, pp. 287–302, March 1983.

[FoBu91] D.J. Fouts and S.E. Butner. "Architecture and Design of a 500-MHz Gallium-Arsenide Processing Element for a Parallel Supercomputer". *IEEE Journal of Solid State Circuits*, Vol. 26, No. 9, pp. 1199–1211, September 1991.

[FoSc87] D.E. Foulser and R. Schreiber. "The SAXPY Matrix 1: A General Purpose Systolic Computer". *IEEE COMPUTER*, pp. 35–43, July 1987.

[FoWa87] J.A.B. Fortes and B.W. Wah. "Systolic Arrays – From Conception to Implementation". *IEEE COMPUTER*, pp. 12–17, July 1987.

[Giga91] *GaAs IC Data Book and Designer's Guide.* GigaBit Logic. 1991.

[Glon88] M. Gloanec et.al. *GaAs Digital Integrated Circuits* (chapter 8, GaAs MESFET Circuit Design), Artech House, 1988.

[Goli91] J.M. Golio. *Microwave MESFETs & HEMTs.* Artech House, 1991.

[HaCo90] R. Hartley and P. Corbett. "Digit-Serial Processing Techniques". *IEEE Transactions on Circuits and Systems*, Vol. 37, No. 6, pp. 707–719, June 1990.

[Haus93] J.R. Hauser. "Noise Margin Criteria for Digital Logic Circuits". *IEEE Transactions on Education*, Vol. 36, No. 4, pp. 363–368, November 1993.

[HeZi87] C.E. Hein, R.M. Zeiger and J.A. Urbano. "The Design of a GaAs Systolic Array for an Adaptive Null Steering Beamforming Controller". *IEEE COMPUTER*, pp. 92–93, July 1987.

[HiLa86] A.J. Hill and P.H. Ladbroke. "High Electron Mobility Transistors (HEMTS) - A Review". *GEC Journal of Research*, Vol. 4, No. 1, pp. 1–14, 1986.

[Hill86] C.F. Hill. "Noise margin and noise immunity in logic circuits". *Microelectron.*, Vol. 1, pp. 16–21, April 1968.

[HwDu90] K. Hwang, M. Dubois, D.K. Panda, S. Rao, S. Shang, A. Uresin, W. Mao, H. Nair, M. Lytwyn, F. Hsieh, J. Liu, S. Mehrotra and C.M. Cheng. "OMP. A RISC-based multiprocessor using orthogonal-access memories and multiple spanning buses". *Proc. ACM International Conference on Supercomputing*, pp. 7–22, Amsterdam, June 1990.

[ieee85] *IEEE STANDARD FOR BINARY FLOATING POINT ARITHMETIC*, ANSI/IEEE Std 754-1985, pp. 260–270, 1985.

[Come92] R. Comerford. "How DEC developed the Alpha", *IEEE Spectrum*, pp. 26–31, July 1992.

[JoHu93] K.T. Johnson, A.R. Hurson and B. Shirazi. "General-Purpose Systolic Arrays". *IEEE COMPUTER*, pp. 20–31, November 1993.

[Kano85] N. Kanopoulos. "A Bit-Serial Architecture for Digital Signal Processing". *IEEE Transactions on Circuits and Systems*, Vol. CAS-32, No. 3, March 1985.

[KaNa85] S. Katsu, S. Nambu, A. Shimano and G. Kano, "A Source Coupled FET Logic – A New Current-Mode Approach to GaAs Logics", *IEEE Transactions on Electron Devices*, Vol. ED-32, No. 6, pp. 1114–1118, June 1985.

[KiHe97] D. Kiefer and J. Heightley. "CRAY-3: A GaAs Implemented Supercomputer System". *Proc. IEEE GaAs IC Symposium*, pp. 3–6, 1987.

[KuHw91] S.Y. Kung and J-N. Hwang. "Systolic Array Designs for Kalman Filtering". *IEEE Transactions on Signal Processing*, Vol. 39, No. 1, January, 1991.

[KuLe78] H.T. Kung and C.E. Leiserson. "Systolic Arrays (for VLSI)". *Proc. Symposium on Sparse Matrix Computations and their Applications*, Duff and Stewart Editors, 1978.

[Kung88] S.Y. Kung. *VLSI Array Processors.* Prentice Hall, 1988.

[LiJe89] C.–M. Liu and C.–W.Jen. "Design of algorithm-based fault-tolerant VLSI array processor". *IEE Proceedings*, Vol. 136, Pt. E, No. 6, November 1989.

[LoBu89] S. Long and S. Butner. *Gallium Arsenide Digital Integrated Circuit Design.* McGraw-Hill, New York, 1989.

[Lohs79] J. Lohstroh. "Static and Dynamic Noise Margins of Logic Circuits". *IEEE Journal of Solid State Circuits*, Vol. SC-14, No. 3, pp. 591–598, June 1979.

[LoLi95] P.L Lozo, C.C. Lim and D. Nandagopal. "Translation Invariant Pattern Recognition: A Real–time Neural Network Architecture Based on Biological Visual Spatial Attention". *Australian Journal of Intelligent Information Processing Systems*, Vol. 2, No. 1, Autumn 1995.

[MaBe92] W. Marwood and A. Beaumont-Smith. "The Architecture and Optimisation of Systolic Ring Processors". *Proc. TENCON '92: IEEE Region 10 Conference*, pp. 735–739, November 1992.

[MaBe92a] W. Marwood and A. Beaumont-Smith. "The Implementation of a Generalised Systolic Serial Floating Point Multiplier". *Proc. APCCAS'92, IEEE Asia-Pacific Conference on Circuits and Systems*, pp. 513–518, December 1992.

[MaCl95] W. Marwood, A.P. Clarke, T.C. Thrum, O. Reinhold and M. Wise. "A Multi-Chip Module Technology and Application". *Proc. 13th Australian Microelectronics Conference*, pp. 40–45, July 1995.

[MaLi91] W. Marwood, C.C. Lim, K. Eshraghian and A. Beaumont-Smith. "Systolic Matrix Processor Architecture for Very High Speed Signal Processing". *Proc. IREECON International Convention*, 1991.

[Magi90] R.N. Mayo et.al. "1990 DECWRL/Livermore Magic Release", *WRL Research Report 90/7*, Western Digital Laboratory, September 1990.

[Marw90] W. Marwood. "A Generalised Systolic Ring Serial Floating Point Multiplier". *Electronics Letters*, Vol. 26, No. 11, pp. 753–754, May 1990.

[Marw90a] W. Marwood and C.C. Lim. "A GaAs Systolic Processor for Implementing a Kalman Filter", *Proc. I.E.Aust. Conference, Microelectronics '90*, 1990.

[Marw91] W. Marwood. "A Generalised Systolic Ring Serial Floating Point Multiplier". *PCT Patent Application No. PCT/AU91/0027*, July 1991.

[Marw91a] W. Marwood. "A Generalised Systolic Ring Serial Floating Point Adder and Accumulator". *Australian Patent*, July 1991.

[Marw94] W. Marwood. "An Integrated Multiprocessor for Matrix Algorithms". Ph.D. Thesis, Department of Electrical and Electronic Engineering, The University of Adelaide, South Australia, 1994.

[McCa90] A.J. McCamant, G.D. McCormack and D.H. Smith. "An improved GaAs MESFET Model for SPICE". *IEEE Transactions on Microwave Theory and Techniques*, Vol. 38, pp. 822–824, June 1990.

[MeCo80] H.T. Kung and C.E. Leiserson. "Systolic Arrays for VLSI", chapter in C. Mead and L. Conway. *Introduction to VLSI Systems.* Addison-Wesley, October 1980.

[Meta92] *HSPICE User's Manual.* META Software. Version H92, 1992.

[NaHi91] T. Naritomi, H. Aso and M. Kimura. "A Fast Processor for 3-D Device Simulation Using Systolic Arrays". *Systems and Computers in Japan*, Vol. 22, No. 1, pp. 39–47, 1991.

[Nowo91] U. Nowotny, M. Lang, M. Berroth, V. Hurm, A. Hulsmann, G. Kaufel, K. Kohler, B. Raynor and Jo. Schneider. "20Gbit/s 2:1 Multiplexer Using

0.3μm Gate Length Double Pulse Doped Quantum Well GaAs/AlGaAs Transistors" , *Microelectronic Engineering 15*, Elsevier Science Publishers B.V., pp. 323–326, 1991.

[Parh89] K.K. Parhi. "Nibble-Serial Arithmetic Processor Designs via Unfolding". *Proc. 1989 Int. Symp. on Circuits and Systems*, pp. 635–640, 1989.

[PuEs88] D.A. Pucknell and K. Eshraghian. *BASIC VLSI DESIGN - Systems and Circuits.* Prentice Hall, 1988.

[Ramb93] *RAMBUS – ARCHITECTURAL OVERVIEW.* Rambus Inc. Mountain View, California USA, 1993.

[Rocc90] M. Rocchi. *High Speed digital IC technologies.* Artech House, 1990.

[SaCa92] R. Sarmiento, P.P. Carballo and A. Nunez. "High speed primitives of hardware accelerators for DSP in GaAs technology". *IEE proc.-G*, Vol. 139, No. 2, pp. 205–216, April 1992.

[Snyd82] L. Snyder. "Introduction to the Configurable, Highly Parallel Computer", *IEEE Computer*, pp. 47–56, January 1982.

[StNe87] H. Statz, P. Newman, I.W. Smith, R.A. Pucel and H.A. Haus. "GaAs FET Device and Circuit Simulation in SPICE". *IEEE Transactions on Electron Devices*, Vol. ED-34, February 1987.

[Ster74] P.H. Sterbenz. *Floating-Point Computation.* Prentice-Hall, 1974.

[Sze83] S.M. Sze. *VLSI TECHNOLOGY.* McGraw-Hill, 1983.

[TaNi92] L.R. Tate, R.J. Niescier, A.C. Hu, J. Scorzelli, W. Leung, C.H. Tzinis, P.J. Robertson and A. Baca. "32 Bit GaAs HFET IEEE Floating Point Multiplier", *IEEE GaAs IC Symposium*, pp. 85–88, 1992.

[TeMo93] *Raphael Interconnect Analysis Program Manual, Version 2.* Technology Modelling Associates Inc. 1993.

[TriQ91] *Gallium Arsenide IC Design Manual.* TriQuint Semiconductor Corp. 1991.

[TriQ92] *GaAs IC Design Course Notes.* TriQuint Semiconductor Corp. May 1992.

[Vite92] *Foundry Design Manual, Version 5.* Vitesse Semiconductor Corp. 1992.

[Vite92a] *1992 Product Data Book.* Vitesse Semiconductor Corp. 1992.

[WeEs85] N.H.E. Weste and K. Eshraghian. *Principles of CMOS VLSI Design – A Systems Perspective.* Addison-Wesley, October 1985.

[WhSp81] H.J. Whitehouse and J.M. Speiser. "SONAR Applications of Systolic Array Technology", *Conference Record, IEEE EASCON*, Washington, D.C., November 17–19, 1981.

[Wing90] O. Wing. *Gallium Arsenide Digital Circuits.* Kluwer Academic Publishers, 1990.

[Zyne88] G.B. Zyner. "Design of Arithmetic Systems in VLSI". Ph.D. Thesis, The University of Adelaide, October 1988.