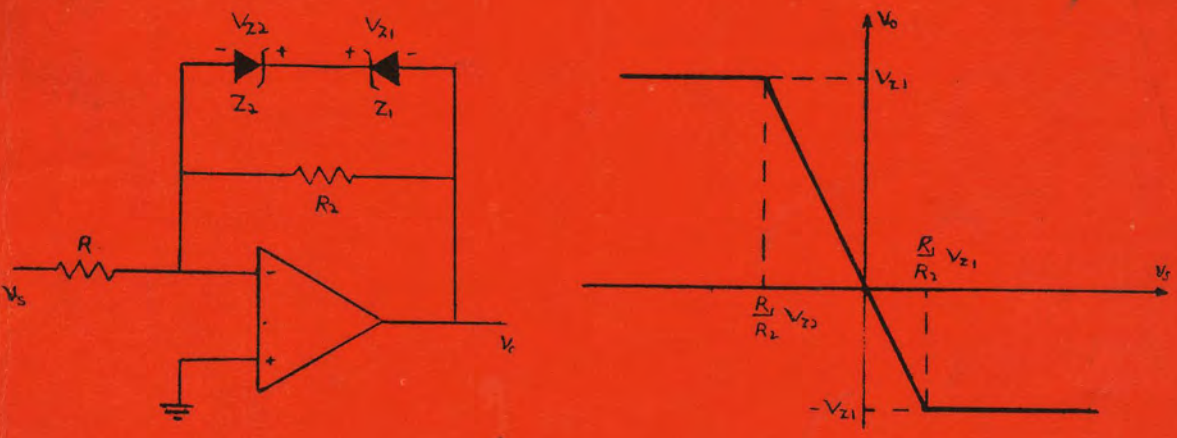
NA-72

PAN-COMMONWEALTH WORKSHOP
ON
MATHEMATICAL MODELLING
IN
CIRCUIT DESIGN



27TH APRIL - 1ST MAY 1992

KANDY, SRI-LANKA

ORGANISED BY :

COMMONWEALTH SCIENCE COUNCIL, LONDON, U.K.

NATURAL RESOURCES, ENERGY AND SCIENCE AUTHORITY OF SRI LANKA, COLOMBO, SRI LANKA

UNIVERSITY OF PERADENIYA, PERADENIYA, SRI LANKA

PREFACE

Engineering Design and Mathematical Modelling opens new avenues for innovation in the industrial and manufacturing sector. The advances made in techniques and tools for mathematical modelling in circuit designs has helped developed countries to increase the pace of progress thus widening the gap between them and the developing world. Hence countries like Sri Lanka which are making a tremendous effort towards industrial progress cannot afford to be complacent. It is because of this interest that Natural Resources, Energy and Science Authority agreed to co-sponsor an International Workshop on Mathematical Modelling in Circuit Designs for the benefit of scientists, engineers and entrepreneurs.

The workshop attracted some of the best known personalities in the field. Hence it was considered vital to document the resource material so that it may be useful to many others who may wish to make reference to it.

R.P. Jayewardene

NARESA, Colombo

24 April 1992.

TABLE OF CONTENTS

NA-12-1

APPLICATIONS OF CIRCUIT DESIGN

By

Dr R. Sivaloaganaadan

Project Officer - Technology Support

Commonwealth Science Council

Pall Mall

London SW1Y 6HX

1.0 INTRODUCTION

The enormous and widespread implications brought about today by the appearance of cheap, powerful and versatile microelectronic devices is comparable to those brought by the industrial revolution brought by the steam engine and electricity. Failure to react sensibly to the microelectronic revolution can hasten the relative decline and produce a substantial fall in living standards. The situation can become even more difficult to a great majority of developing countries already experiencing enormous difficulties in their efforts to industrialise. Inspite of the general decline in the worldwide economic growth, the growth in electronic goods have shown rapid progress [1]. Table I [1] shows the estimated world sales in millions of dollars of semi-conductors by major international firms for the years 72, 74, 76, 80, 83, 84 and 85. The foregoing facts show that a clear understanding on microelectronics and microelectronic circuits is necessary together with an understanding on their application areas.

TABLE I ESTIMATED WORLD SALES OF SEMICONDUCTORS

Firm	Location	72	74	76	80	83	84	85
Texas Instruments	United States	405	652	655	1580	1310	2350	1815
Motorola	United States	310	482	462	1100	1500	2255	1650
National Semiconductor	United States	75	204	263	770	690	1270	890
NEC	Japan	-	120	343	769	1084	1985	1950
Hitachi	Japan	135	195	240	n.a.	820	1690	1750
Toshiba	Japan	145	170	233	629	705	1460	1350
Intel	United States	45	n.a.	n.a.	575	685	1170	900
Fairchild	United States	165	324	307	566	400	n.a.	n.a.
Philips	Netherlands	225	398	385	942	n.a.	1150	950
Siemens	Fed. Rep of Germany	80	142	n.a.	420	n.a.	n.a.	n.a.
Fujitsu	Japan	-	120	343	419	500	815	950
Mostek	United States	n.a.	n.a.	n.a.	330	n.a.	n.a.	n.a.
Matusushita	Japan	70	150	254	300	n.a.	n.a.	870
Mitsubishi	Japan	70	80	94	254	n.a.	n.a.	n.a.

2.0 APPLICATIONS

The application of semiconductor devices mainly fall into two categories namely (i) use of individual components and (ii) use of integrated circuits. An integrated circuit can be devised to perform a set of functions. With given inputs or given operations from its memory it can produce an output which is built as 'fixed logic'. Alternately an integrated circuit can be programmable so that it is capable of being given fresh instructions to operate on a different logic. A microprocessor is a programmable circuit which with the addition of memory constitutes the core computer. A system can be controlled by a local microprocessor on the spot or it can be controlled remotely by a line to a big computer. The use of

microprocessors also overlap with the use of fixed logic circuits. In some circumstances it may be better to manufacture fixed logic circuits and in some other circumstances it may prove uneconomical. In some circumstances it may be best to use a microprocessor with a program to provide the necessary control even though there is no necessity for a change in the program or software. These all depend on the economies of mass production and the design engineer should have a clear understanding of the options and the associated constraints.

2.1 CONSUMER GOODS

There is a vast amount of consumer goods flooding the market. Some of them have simple switching and sensing devices while many of them have electronic circuits to operate and control the functions the product is expected to perform. Furthermore there can be innovative thinking in this area resulting new products in the market.

2.1.1 DOMESTIC APPLIANCES

This group considers appliances which one uses in the day to day life. Of these the minor appliances like iron and kettle are still having the traditional electromechanical switching and sensing devices. This is because of the lack of returns for the extra expenditure on the circuitry. But major appliances like automatic washing machines, radios, cassette players, electronic alarm clocks, video players, televisions and microwave ovens are made sophisticated by the introduction of electronic controls in the form of microprocessor and fixed logic controls. It is now common to have washing machines that run more than ten different washing programs which include cold wash to hot wash and tough wash to delicate wash, all of which are essentially different combinations of temperature, spin and waterflow. These washing programs are controlled by the microprocessors. The fundamental principle here is that the introduction of the control circuitry enables easy implementation of several programs, which otherwise would have been difficult or economically not viable for implementation.

2.1.2 CONTROL OF BOILERS, VENTILLATORS AND REFREGERATION EQUIPMENT

In hotels, warehouses and industrial complexes the operational costs of boilers, ventillators and refregeration equipment can be substantial. The problem is made complex by the change in weather and the change in load. Different programs need to be run to switch on and switch off and the loading conditions have to be changed to effect efficient operation. These operations are best controlled by a fixed logic circuit or a microprocessor.

2.1.3 ENGINE MANAGEMENT IN CARS

With the conciousness about the environment is growing, the emmission control of cars is becoming a major issue. Fixed logic controllers are ideal candidates to control ignition, fuel supply and air/fuel ratio to ensure ideal emmission and driving conditions. The quantity involved can justify the development of a purpose built controller.

2.1.4 TOYS AND GAMES

One of the evergrowing markets is that for toys and computer games. If one walks around toy shops now, he or she can see singing dolls, walking and talking teddies, music organs and many similar items which all employ circuits. All in all microelectronics has opened up an entirely new variety of markets.

2.2 INDUSTRIAL APPLICATIONS

If one considers an industrial process plant like that of a cement plant or an amonia plant here again there are several circuits. Such an industrial complex will consume several megawatts of power and there will be several power circuits. But the operational comfort is achieved by the measurement and control facilities which involve many electronic circuits. It can be said with confidence that the high capacity and operational efficiency achieved today would not have been possible if not for the electronic measurement and control system. Balasubramaniam [2] lists the electrical and instrumentation systems in the Lanka Cement Ltd factory at Kankesanthurai Sri-Lanka in the following way.

1 High voltage power supply systems and incoming transformers
2 Medium voltage power supply system and power distribution to various plant units
3 Protection systems to feeders and equipments
4 Low voltage power supply systems.
5 Electric drives
6 Motor protection, motor control and operation systems
7 Power electronics systems for variable speed drives
8 Power factor improvement systems
9 Dust arresting and collection systems
10 Electrical systems used in material blending proportioning and metal separation
11 Instrumentation and display of parameters at Central Control Room
12 Operational parameter control system
13 Emergency power supply systems
14 Illumination systems and service power supply and
15 Safety measure to personnel and plant units.

To discuss all these one could write a book. But to illustrate the point consider one item, instrumentation and display of parameters at central contol room, the 11th item in the list. A transducer is employed at the point where the measurement of a process parameter such as temperature or pressure is required. The transducer converts the process parameter into electrical signal. This often weak signal is received by a circuit sometimes called the transmission, which magnifies it and send it to the display. The display can be a digital one or a conventional one of the moving coil type.

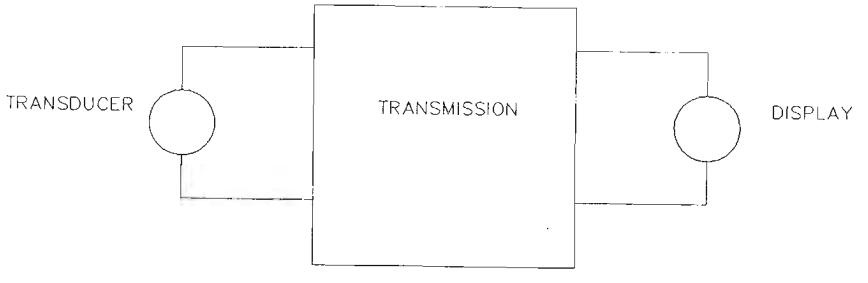


FIGURE 1 MEASUREMENT SYSTEM

Figure 1 shows the schematic representation of this measurement system. The transmission here plays an important role. The user of the system wants a linear display of the measured characteristic whereas the transducer may produce a nonlinear output signal. In such a situation the transmission circuit has to accommodate these variations and still produce a linear output. Here a digital output may be useful. Thus it can be seen that the microelectronics plays an important role in industrial control and opens new approaches for industrial control.

3.0 CIRCUIT SYNTHESIS AND ANALYSIS

A circuit system essentially has three constituents namely (i) circuit (ii) power source or excitation and (iii) response. The circuit is the collection of electrical elements such as the resistor, capacitor and transistor with details of their interconnections. The excitation or power source is the source of electrical energy connected to the circuit. The response is the output characteristic desired from the circuit. This can be a current, voltage, the ratio of amplitudes and so on. If the circuit and the source are given and the response is to be determined the problem is defined as analysis. If the power source and response are given and the circuit is to be determined the problem is said to be synthesis. Synthesis is the process of determining the circuit components and their interconnections to give a specified excitation-response characteristics. This depends very much on the experience and intution of the designer and as a result there can be more than one design for a given excitation-response characteristics.At the begining network design has been based primarily on an analysis approach in which a specific arrangement of elements was tested, results noted and compared with a specific response and the network modified to reduce variation. This trial and error approach depends mainly on the designer's experience.

Network synthesis is achieved through the following steps:
(i) Specifying a desired response due to a prescribed input
(ii) Formulating suitable functions to represent the network and
(iii) Determining an arrangement of components which yields the required output or response from the physical system.

Specifying the response and the definition of the mathematical functions is called the approximation stage while the procedures for decomposing the mathematical model to formulate a physically realizable collection of components is called the realization stage. Several literature deals with this process of circuit synthesis [3,4] and the associated techniques. Circuits are synthesized in various domains such as the Laplace domain, frequency domain and time domain. This process of circuit synthesis is good for simple and mostly analogue circuits. But digital circuits with very many operational conditions revolutionise the circuit synthesis process. A new aapproach to product development in the name of Mechatronics is developed in Japan in the recent past. The underlying philosophy of Mechatronics is based on the fact that the success of a present day product often depends on the integration of electronics and computing technologies and a properly conceived product exploits the development in electronics and circuit design at the early stage in the product and manufacturing process design. Mechatronics is the synergetic combination of precision mechanical engineering, electronic control and system thinking in the design of products and processes.

The preceding sections explain the basics of circuit design and one can now see their relative importance by considering som example situations.

4.0 EXAMPLE APPLICATIONS

This section presents examples illustrating the translation of the requirementsof a physical system to those of a circuit. The circuit could then be synthesized using the mathematical models of components and other microcircuits to simulate the final circuit. The circuit could then be physically realised.

4.1 DIGITAL CONTROL OF FEED HOPPER

In a cement factory limestone and clay are interground to form a powder called raw meal. This raw meal is burnt in the kiln to form clinker which when interground with 3to 5% of Gypsum results ordinary portland cement. One of the important process control function is the control of the raw meal supply to the kiln. The raw meal is pumped from the storage silos to an airstream (blower 1) which carries the raw meal to the feed hopper from which it is drawn by the weighfeeder to feed the kiln. The raw meal in the hopper is kept in the fluidised state by blower 2. The whole set up is given in figure 2. The raw meal pump should operate only when (i) weighfeeder is on (A) (ii) blower 1 is on (B) (iii) blower 2 is on (C) and (iv) the hopper is not full (D). Any violation would result in breakdown of the process. In the traditional setup limit switches and other mechanisms are used to achieve these requirements. In a mechatronic approach these could be controlled by a fixed logic circuit or a common microprocessor.

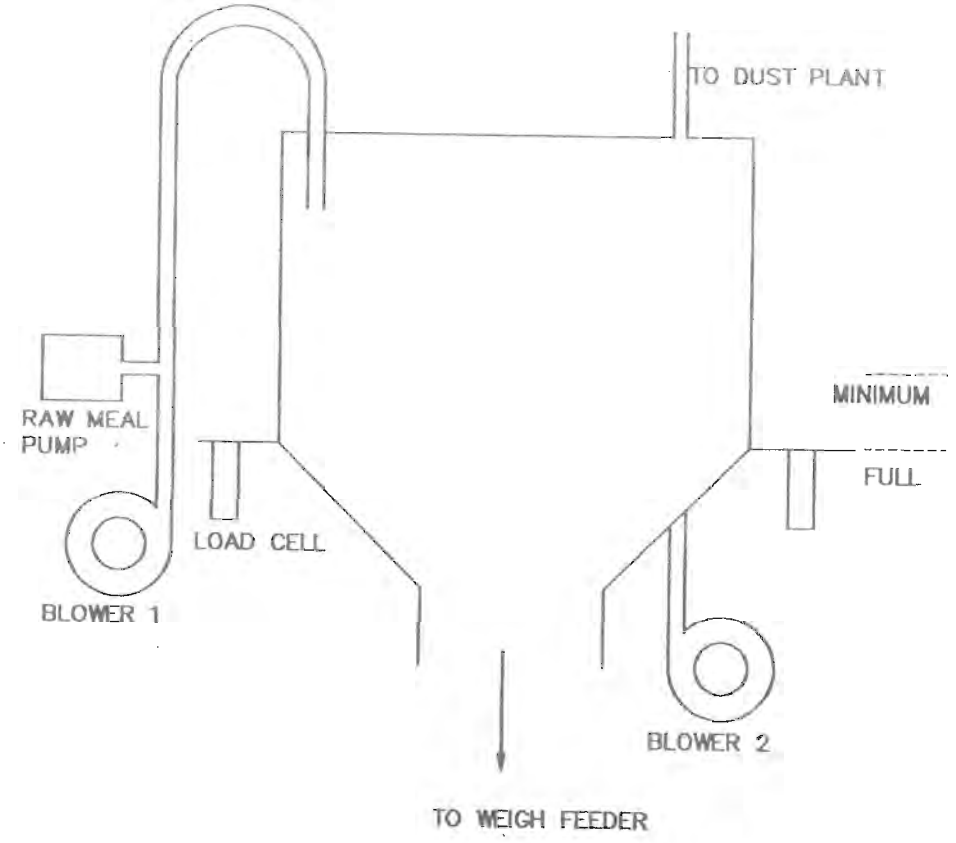


FIGURE 2 CONTROL OF FEED HOPPER

4.2 OPTICAL COUNTERS

The finished products from factories are loaded to lorries through conveyor belts and often it is a problem to keep an accurate count of the items loaded. Consider the situation in a

cement plant again. A photo sensor employed to measure the distance between it and the belt passing underneath. The passing of a cement bag would indicate a change in this distance. Figure 3 illustates the setup. The problem of counting is easy if the bags are oriented and come in equal spacing. But bags come in random orientation and at random spacings. Hence measurements have to be taken and calculations have to be performed before adding a bag to the bag counter. These could be performed well by a software driven processor.

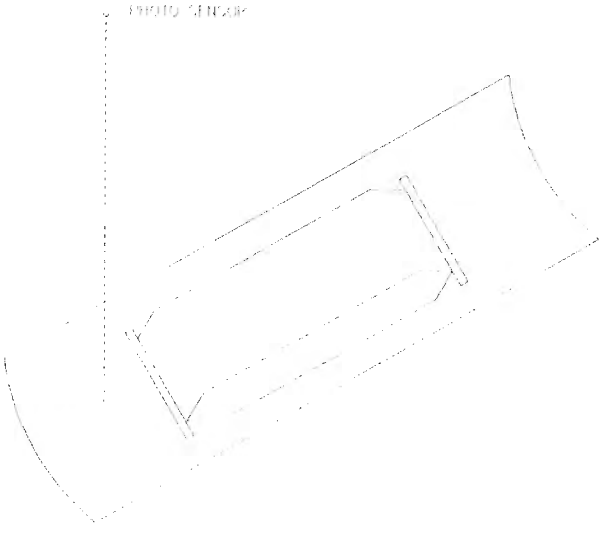


FIGURE 3 OPTICAL COUNTER

5.0 CONCLUSIONS

(i) The applications of circuitry vary widely.
(ii) Requirements of the circuit are first established as a mathematical model or otherwise before starting the design.

(iii) Synthesis techniques are used to meet these requirements.
(iv) Fixed logic and programmable controllers are used to perform several functions.
(v) Present day products require a combined mechanical and electronic or Mechatronic approach to design.

REFERENCES

1 Kaplinsky R., *Microelectronics and Employment Revisited*, Intenational Labour Office , Geneva 1987.

2 Balasubramaniam S., *Electrical and Instrumentation Systems*, Cement Industry in Sri-Lanka, A session at the Annual Sessions of IESL, 1984.

3 Valkenburg V., *Introduction to Modern Network Synthesis*, John Weily & Sons, 1960.

4 Weinberg L., *Network Analysis and Synthesis,* Mcgraw Hill Book Company Inc, 1962.

Device Modelling

Dr Alan Pearmain

Queen Mary & Westfield College, University of London

1. Why do we need device models?

Active electronic components - diodes, transistors, triacs etc. - have non-linear characteristics that prevent simple mathematical equations from being used to analyse the behaviour of the device. In computer-aided design we need to both verify and simulate the behaviour of our circuits before the circuit is committed to the considerable expense of fabrication. A model of the device, either consisting of a set of equations or a set of equivalent passive components is therefore required for circuit analysis. A possible alternative to a model is a table of look-up values that give the set of currents and corresponding voltage values for the device. This will be a set of discrete points with interpolation used to obtain intermediate values. The degree of granularity that is acceptable in such a table will depend on the application.

A problem with any device model is that real devices have a range of values for their parameters, depending on the variation of the processing. In the case of integrated circuits, for example, apart from variations in processing between batches there is also a variation between the chips that are in the centre of a wafer and those at the edge.

2. Device models for different simulators

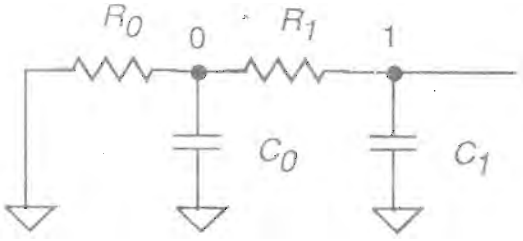
The complexity of the model that we require depends on the type of simulation or verification that we intend to perform. At the simplest for a digital circuit we could model transistors merely as switches, but this would only give us the most basic information about how the circuit performed. Alternatively, if we needed accurate timing information about an analogue (or high-speed digital) circuit with small geometry transistors we would need a complex model for the transistors, but we would obtain accurate waveforms for the output signals.

2.1. Models for digital simulation

If we are performing a complex digital design an accurate circuit simulation of the whole circuit that treated all the signals as true analogue waveforms would be impossible with existing circuit simulation programs and computer hardware. The circuit can only be simulated using idealised digital waveforms.

At the simplest transistors would be represented as switches, or logic gates as truth tables. However, we are more often interested in the gate-level timing

simulator. If we confine our circuit to CMOS technology then the gate delays can be calculated by considering the gate to consist of the gate capacitance of the transistor being driven and the channel resistance of the driving transistor.



At $t = 0$ capacitors C_0 and C_1 are both charged to V_{DD} and the circuit begins to discharge. The time constant of the process

$$T_D = R_0(C_0 + C_1) + R_1C_1 = R_0C_0 + (R_0 + R_1)C_1$$

There are several errors in this simple approach. One is that the capacitance values are voltage dependent and another is that the gate delay depends partly on the rate-of-change of input voltage. In the case of the *Crystal* simulator this is compensated for by using a resistance value that depends on the input slew rate.

2.2. Models for analogue simulation

These must be more complex and must attempt to represent the physical nature of the active devices. The parameters to be inserted in the models will depend on the fabrication process and must be derived from measurements of devices fabricated on the same line, and ideally as part of the same wafer, as the device used, although this is obviously not practical to derive parameters for every wafer.

The most widely used circuit simulator is *SPICE* or variations on SPICE such as *PSPICE HSPICE* and *MSPICE*. The models that are used in this simulator are the ones that have received most attention

3. SPICE models

SPICE has built-in models for the most common active components met in electronic design. There are simple techniques (using sub-circuits) for adding models for other devices. The built-in models are for the diode, JFET, MOSFET, GaAsFET and bipolar junction transistor.

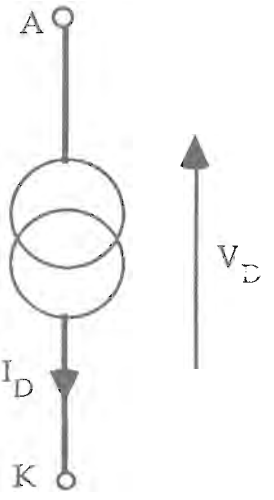
3.1. The diode models

There are three separate models used: one for dc, one for small-signal analysis and one for large-signal analysis.

The objective of the models is to predict the behaviour of a real device, but in some analysis ideal diode behaviour may be an adequate representation.

3.1.1. *dc model of the ideal diode*

The dc model for the ideal diode is a non-linear current source:



The standard forward characteristic of the ideal diode is then

$$I_D = I_S\left(\exp\left(\frac{eV_D}{kT}\right)-1\right)+V_D GMIN$$

for $V_D \geq -5\frac{kT}{e}$

The first part of this equation is the well-known equation for the relationship between the reverse saturation current and the forward current, but the GMIN term is a small conductance (default value 10^{-12} S) added in parallel with each diode to aid convergence in the simulation program.

The reverse characteristic can be represented by this same equation for

$$-5\frac{kT}{e} \leq V_D \leq 0 \text{ and}$$

$$I = -I_S + V_D \text{GMIN} \quad \text{for } V_D < -5\frac{kT}{e}$$

3.1.2. *dc model of the real diode*

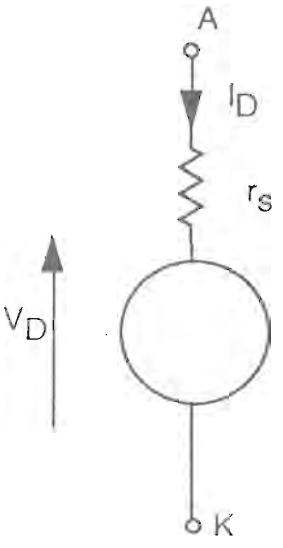
For the real diode dc. characteristic we have two parts to the forward-biased junction case: small bias and large bias. In the small bias case the situation is the same as for the ideal diode, and the reverse bias characteristic is

$$I_D = I_S\left(\exp\left(\frac{eV_D}{nkT}\right)-1\right)+V_D GMIN \text{ for } -5\frac{nkT}{e} \leq V_D \leq 0 \text{ and}$$

$$I = -I_S + V_D \text{GMIN} \quad \text{for } V_D < -5\frac{kT}{e}$$

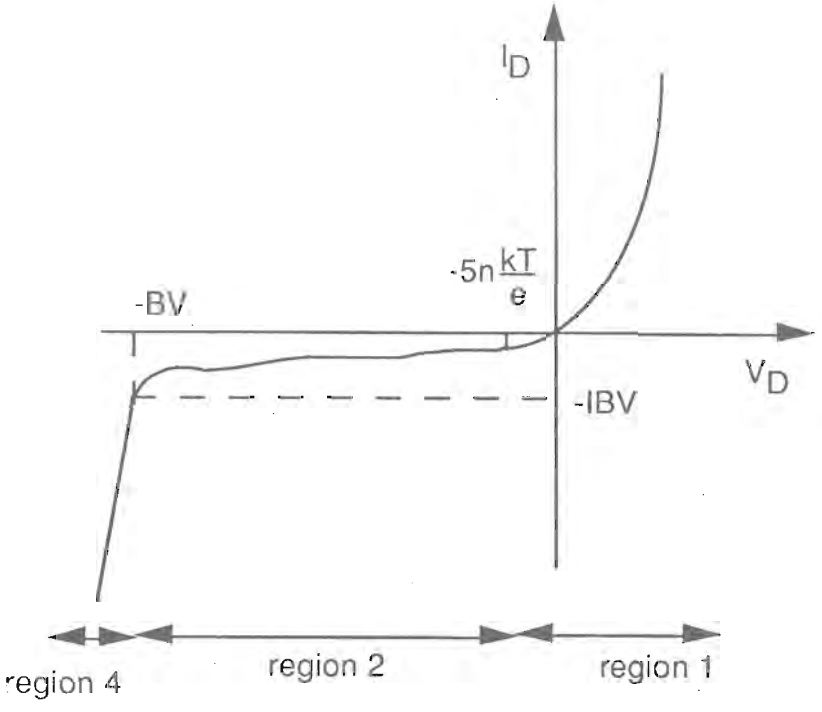
where n is the emission coefficient ($1 \leq n \leq 2$)

This is the same as the reverse-biased ideal diode except for the introduction of n.



For a large forward bias there are two extra effects: the ohmic resistance of the diode and high-level injection. The injection is not modelled separately, only by an equivalent value for the ohmic resistance. Therefore the equation is

$$V_D' = r_s + V_D$$



This leaves one more dc. effect to model - the breakdown under large reverse bias. This has four distinct regions:

Region 1 obeys the equation given for reverse bias of the real diode

$$I_D = I_S\left(\exp\left(\frac{eV_D}{nkT}\right)-1\right)+V_D GMIN$$

In region 2 the equation becomes

$$I_D = -I_S + V_D \text{GMIN}$$

The third region is the discrete point where V_D = -BV with BV being the breakdown voltage and here I_D = -IBV

The fourth region, beyond the breakdown voltage, has the equation

$$I_D = -I_S\left(\exp\left(\frac{-e(BV+V_D)}{kT}\right)-1+\frac{eBV}{kT}\right)$$

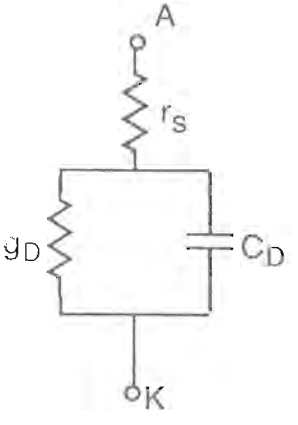
A convergence problem is given by these different equations for the different regions.

This therefore gives us a set of parameters that are required for the SPICE model.

IS	=	saturation current I_S
RS	=	ohmic resistance r_S
N	=	emission coefficient n
BV	=	breakdown voltage BV
IBV	=	breakdown current IBV

3.1.3. *The small-signal model*

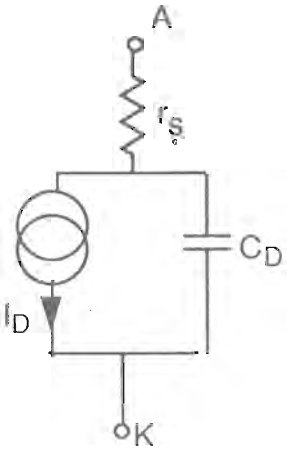
This will include the capacitance of the diode:



The values of C_D and g_D are calculated at the dc operating point. This gives a simple linear model with a simple equation.

3.1.4. *The large-signal model*

The more general model must include the effects of stored charge in the space charge and the minority-carrier injection.



The equations that result from this model are in terms of the zero-bias capacitance $C_d(0)$ and the built-in voltage ϕ_0

We get as our equations:

$$C_D = \tau_D \frac{dI_D}{dV_D} + C_d(0)\left(1 - \frac{V_D}{\phi_0}\right)^{-m} \quad \text{for } V_D < FC \times \phi_0$$

$$C_D = \tau_D \frac{dI_D}{dV_D} + \frac{C_d(0)}{F_2}\left(F_3 + \frac{mV_D}{\phi_o}\right) \quad \text{for } V_D \geq FC \times \phi_0$$

$$F_2 = (1 - FC)^{1+m}$$

$$F_3 = 1 - FC(1 + m)$$

The first term in the equations of C_D is the stored charge due to injected minority carriers term and the second term due to the space charge.

We therefore require some additional parameters:

CJO	=	zero-bias *pn* capacitance $C_d(0)$
FC	=	forward-bias depletion capacitance coefficient *FC*
TT	=	transit time τ_D
VJ	=	the *pn* potential ϕ_0
M	=	the *pn* grading coefficient

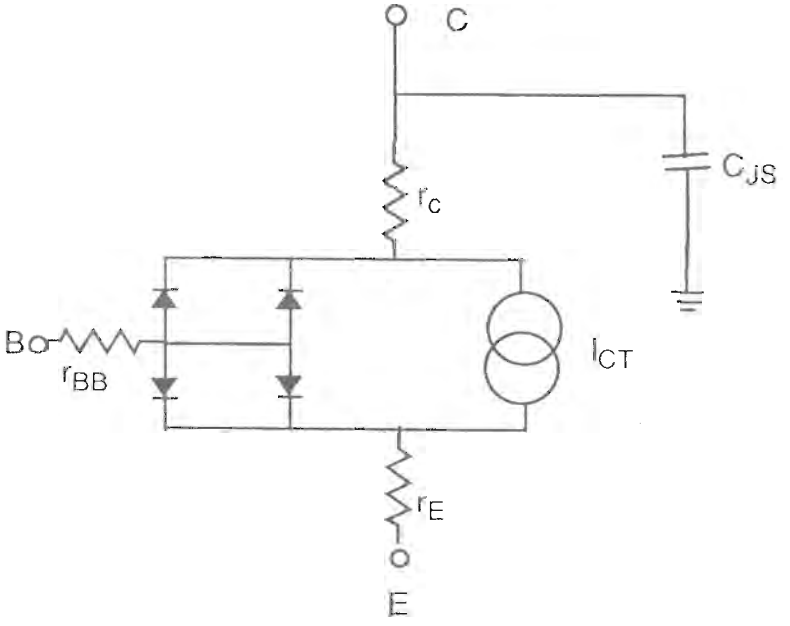
This gives some idea of the complexity of modelling even the diode in SPICE. There are additional elements to model the temperature-dependence of the model parameters.

3.2. The bipolar junction transistor models

The model used is the Gummel-Poon model [1] which will default to the Ebers-Moll model [2] if some of the parameters are omitted. Again there are dc, small signal and large signal models.

The Gummel-Poon model was developed to be suitable for computer simulations and it uses charge-integral relationships. It accounts for base-width modulation, charge storage, lifetime dependence on current, β dependence on current and some high-injection effects. There are 40 parameters to completely specify the model.

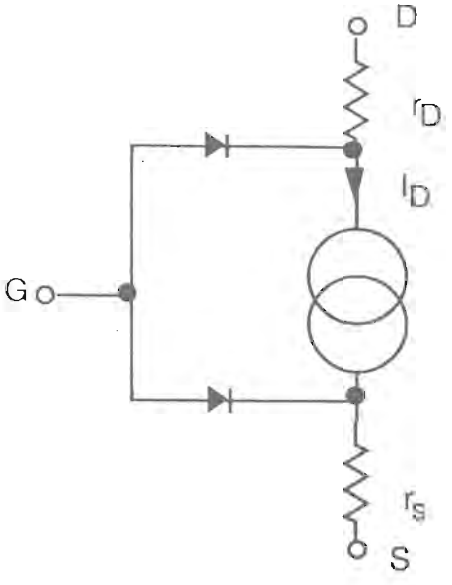
The static version of the model is:



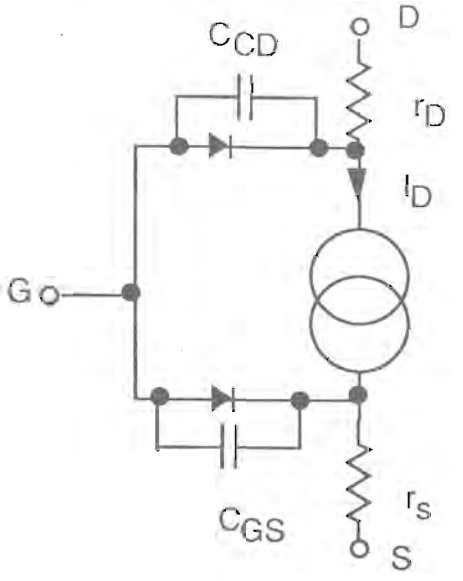
The detail of the model is in the specification of the diodes and current generator.

3.3. The junction field-effect transistor models

This is based on the Shichman and Hodges FET model [3].



dc model of a FET



Large signal model of a FET

The parameters required this time are:

CGS = Zero-bias gate-source capacitance

CGD = Zero-bias gate-drain capacitance

PB	=	Gate junction potential ϕ_0
FC	=	Forward bias depletion capacitance coefficient
m	=	Junction grading coefficient (set to 0.5)

3.4. The MOSFET models

MOSFET models are probably the most important since the devices in complicated integrated circuits are usually MOSFETs. SPICE offers three different models for MOSFETs, the level 1, level 2 and level 3 models.

3.4.1. Level 1 model

The level 1 model is basically the Shichman and Hodges model [3]. This gives two equations:

$$I_{DS} = KP\frac{W}{L-2X_{jl}}\left(V_{GS}-V_{TH}-\frac{V_{DS}}{2}\right)V_{DS}(1+\lambda V_{DS})$$ for the linear region and

$$I_{DS} = \frac{KP}{2}\frac{W}{L-2X_{jl}}(V_{GS}-V_{TH})^2(1+\lambda V_{DS})$$ for the saturation region

W and L are the length and width of the channel of the MOSFET.

The SPICE parameters that we require are:

GAMMA	=	Body-effect parameter
KP	=	transconductance parameter
LAMDA	=	channel length modulation
LD	=	Lateral diffusion (X_{jl})
PHI	=	surface inversion potential
VTO	=	Zero-bias threshold voltage

Some of these can be obtained from the physical parameters

NSUB	=	Substrate doping density
TOX	=	Thickness of oxide layer
UO	=	Surface mobility (μ_0)

3.4.2. *level 2 model*

Here the depletion region charge is modelled more accurately. The charge depends on the voltage in the channel and a model developed by Meyer [4] is used.

The equations for this model are:

$$I_{DS} = \frac{KP}{1-\lambda V_{DS}} \frac{W}{L-2X_{jl}} \left\{ \left(V_{GS} - V_{FB} - 2\phi_p - \frac{V_{DS}}{2} \right) V_{DS} - \frac{2}{3}\gamma \left[\left(V_{DS} - V_{BS} + 2\phi_p \right)^{1.5} \right] \right\}$$

for the linear region and

$$I_{DS} = I_{D,sat} \frac{1}{1-\lambda V_{DS}} \quad \text{for the saturation region.}$$

Other areas where the level 2 model is more accurate are that it allows for the variation of mobility with the gate electric field, the variation of channel length in the saturation region and the maximum velocity that charge carriers can obtain, together with some other second order effects. There are therefore several extra parameters required:

DELTA	=	Width effect on threshold voltage
NEFF	=	Total channel charge coefficient
NFS	=	Surface fast-state density
NSS	=	Surface state density
TPG	=	Type of gate material
UCRIT	=	critical electric field for mobility
UEXP	=	Exponential coefficient for mobility
UTRA	=	Transverse charge coefficient
VMAX	=	Maximum drift velocity of carriers
XJ	=	Metallurgical junction depth
XQC	=	Coefficient of channel charge share

3.4.3. *Level 3 model*

This model has been developed by Liu [5] for short-channel MOSFETs, i.e. channel lengths up to 2 μm. The objective is that the model should be both simpler and more accurate, although some work has suggested that it is not necessarily more accurate. It requires slightly fewer parameters than the level 2 model. As device lengths fall (0.8 μm is common in current fabrication practice) simulation of short-channel devices has become the most common task.

The current equation in the linear region has been simplified by a Taylor-series expansion.

$$I_{DS} = \beta\left(V_{GS} - V_{TH} - \frac{1+F_B}{2}V_{DS}\right)V_{DS}$$

where $$F_B = \frac{\gamma F_s}{2\sqrt{2\phi_p - V_{BS}}} + F_n$$

The short channel influences V_{TH}, F_S and β while the narrow channel effects influence F_n.

As the gate oxide is always thin in short-channel devices the surface electric field is almost always greater than the critical field so the equation for predicting the influence of field on mobility can be simpler.

The new parameters are:

ETA	=	Static feedback threshold
THETA	=	Mobility modulation

3.5. Short channel problems

There are a number of problems that become more important as the channel length becomes shorter. One is that the threshold voltage is effected by the channel length. Whilst this can be computed the computation is too time-consuming for circuit simulation. Another model has been proposed by Sakurai and Newton [6] that claims to give accurate results for short-channel devices with one third the computation time of the level 3 model and with a smaller number of parameters.

4. Measurement of SPICE parameters

The parameters must be extracted from measurements on devices that have been fabricated. This can prove very difficult for some of the parameters.

For BJT transistors the measurements used are I_B and I_C variation with V_{BE}. A program is usually used to convert values that can be measured to the parameters required, e.g. TF, the ideal total forward transit time, is determined from a measurement of the unity gain bandwidth f_T. SPICE gives default values for most parameters, but these may be considerably inaccurate for the fabrication process actually being used. The more parameter values that can be supplied from measurements on devices the more accurate simulation will be. Special drop-in structures on the wafer are used to determine SPICE parameters.

For MOSFETs measurement of the variation of I_{DS} with V_{GS} will give V_{TH} and KP for the level 1 model. If measurements are made at different values of

substrate voltage the value of GAMMA (body effect) can be determined. Level 2 parameters require measurements to be made on a number of devices with different channel lengths and widths. A full set of measurements can generally yield level 2 or level 3 parameters, but the quantity of data required is considerable.

5. Constants used

e = charge on an electron $=1.6 \times 10^{-19}$ C

k = Boltzmann's constant $= 1.38 \times 10^{-32}$ J/K $= 8.6 \times 10^{-5}$eV/K

6. References

[1] Gummel, H.K. and Poon, H.C., An Integral charge control model for bipolar transistors, *Bell Syst. Tech. J.*, 49, 827-851 (1970).

[2] Ebers, J.J. and Moll, J.L., Large-signal behaviour of junction transistors, *Proc IRE*, 42 1761-1772 (1954).

[3] Shichman, H. and Hodges, D.A., Modelling and simulation of insulated-gate field-effect transistor switching circuits, IEEE *J Solid-State Circuits*, SC-3, 285-289 (Sept 1968).

[4] Meyer, J.E., MOS models and circuit simulation, *RCA Review*, 32, 42-63 (March 1971).

[5] Liu, S., *A unified CAD model for MOSFETs, Memorandum* No. UCB/ERL M81/31, Electronics Laboratory, University of California, Berkeley (20 May 1981).

[6] Sakurai, T. and Newton, A.R., *A simple MOSFET model for circuit analysis and its application to CMOS gate delay analysis and series-connected MOSFET structure*, Memorandum No. UCB/ERL M90/19, Electronics Laboratory, University of California, Berkeley (16 March 1990).

7. Bibliography

Antognetti, P. and Massobrio, G., *Semiconductor device modelling with SPICE*, McGraw-Hill (1988).

Schlegel, E.S., A bibliography of Metal-Insulator-Semiconductor studies, *IEE Trans. Electron Devices*, ED-14, 728-749 (Nov. 1967).

Schlegel, E.S., Additional bibliography of Metal-Insulator-Semiconductor studies, *IEE Trans. Electron Devices*, ED-15, 951-953 (Dec. 1968).

MATHEMATICAL MODELING OF ANALOG CIRCUITS

Jiri Vlach
Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario, Canada

Abstract
Detailed modeling of analog circuits has reached a high level of maturity but new problems are arising. Classical simulators like SPICE can solve networks with hundreds of transistors but a need arises to simulate much larger networks. Technological advance also permit new developments, like switched-capacitor networks or switched power supplies. This contribution considers non-classical methods of analog simulation and gives references for reading.

1. Linear networks

Theory of linear networks is used mostly for the design of filters in frequency domain. In many cases this calls for the application of optimization and for calculation of sensitivities with respect to changes of network elements. The theory is often based on the application of Tellegen's theorem and on a special "adjoint" network. The steps are difficult to understand and we feel that a purely algebraic method leads to the same result in a much simpler way.

Let the system equation be described by

$$\mathbf{TX}=\mathbf{W} \tag{1.1}$$

We can solve it formally by writing

$$\mathbf{X}=\mathbf{T}^{-1}\mathbf{W} \tag{1.2}$$

Here T is the system matrix. If it is in modified nodal formulation, then it has the form

$$\mathbf{T}=\mathbf{G}+s\mathbf{C} \tag{1.3}$$

In order to get the derivative of the unknown vector X with respect to an element h, we differentiate (1.1)

$$\mathbf{T}\frac{\partial \mathbf{X}}{\partial h}+\frac{\partial \mathbf{T}}{\partial h}\mathbf{X}=\frac{\partial \mathbf{W}}{\partial h}$$

In modified nodal formulation the right hand side vector does not depend on the elements and we can simplify

$$\frac{\partial \mathbf{X}}{\partial h}=-\mathbf{T}^{-1}\frac{\partial \mathbf{T}}{\partial h}\mathbf{X} \tag{1.4}$$

Suppose now that we wish to get only one output, which can be either a nodal voltage or a difference of two nodal voltages. We define the output by

$$F=\mathbf{d}^t\mathbf{X} \tag{1.5}$$

where the vector $\mathbf{d}$ selects the desired output from the vector $\mathbf{X}$. Derivative of this function with respect to h is

$$\frac{\partial F}{\partial h}=\mathbf{d}^t\frac{\partial \mathbf{X}}{\partial h} \tag{1.6}$$

Inserting from (1.4) we obtain

$$\frac{\partial F}{\partial h}=-\mathbf{d}^t\mathbf{T}^{-1}\frac{\partial \mathbf{T}}{\partial h}\mathbf{X} \tag{1.7}$$

The row vector $\mathbf{d}^t\mathbf{T}^{-1}$ can be precomputed before sensitivity calculations. Denote this vector by $\mathbf{X}^a$ and solve

$$(\mathbf{X}^a)^t=-\mathbf{d}^t\mathbf{T}^{-1} \tag{1.8}$$

Postmultiply by $\mathbf{T}$, take the transpose and solve

$$\mathbf{T}^t\mathbf{X}^a=-\mathbf{d} \tag{1.9}$$

Once $\mathbf{X}^a$ is known, we use it in (1.7) to get the final formula

$$\frac{\partial F}{\partial h}=(\mathbf{X}^a)^t\frac{\partial \mathbf{T}}{\partial h}\mathbf{X} \tag{1.10}$$

Both vectors $\mathbf{X}$ and $\mathbf{X}^a$ are obtained by one LU decomposition and two forward-back substitutions. The sensitivity of the output with respect to any element costs *at most* one multiplication and two subtractions.

This method was used in the program NETOPT [1] for finding sensitivities with respect to elements. Similar steps were also used to derive sensitivity of group delay, available in the same program. Additional details are in [2].

2. Iterative methods

Modern analog simulators are based on a few fundamental discoveries which were made some 20 years ago:
(a) sparse matrix theory,
(b) formulations resulting in systems of algebraic-differential equations,
(c) stiffly stable integration methods for systems of algebraic-differential equations.
All three subjects are dealt with in [2].

Simulation can be only as accurate as are its models and accurate semiconductor models are quite complicated. As a result, classical simulators are excellent for networks with hundreds but not for networks with many thousands of transistors.

Parts of large networks are often idle, but sparse matrix solutions must go through the whole solution process at every step. Attempts to overcome this problem, called latency, resulted in the development of iterative methods [3,4]. Although many

modifications are possible, the principles have been known for many years under the names Gauss-Jordan and Gauss-Seidel. They can be easily explained on an example:

$$4x_1+x_2+2x_3=16$$

$$x_1+3x_2+x_3=10$$

$$x_1+2x_2+5x_3=12$$

or in matrix form

$$\mathbf{Ax=b} \tag{2.1}$$

As a preprocessing step modify the equations into the form

$$x_1=4-x_2/4-x_3/2$$

$$x_2=10/3-x_1/3-x_3/3$$

$$x_3=12/5-x_1/5-2x_2/5$$

For iteration use known values on the right and obtain new values on the left. This is repeated until convergence is reached.

The steps can be described mathematically by first partitioning the matrix **A** into

$$\mathbf{A=L+D+U}. \tag{2.2}$$

where **L** is lower triangular, **D** diagonal and **U** upper triangular portion of **A**. With this notation the Gauss-Jordan iteration becomes

$$\mathbf{Dx}^{k+1}=\mathbf{b}-(\mathbf{L+U})\mathbf{x}^k$$

or

$$\mathbf{x}^{k+1}=\mathbf{D}^{-1}\mathbf{b}-\mathbf{D}^{-1}(\mathbf{L+U})\mathbf{x}^k \tag{2.3}$$

All diagonal entries of **D** must clearly be nonzero. Convergence of the system will depend on the matrix

$$\mathbf{M}_{GJ}=\mathbf{D}^{-1}(\mathbf{L+U}) \tag{2.4}$$

A proof exists that convergence is guaranteed if $\mathbf{M}_{GJ}$ has all eigenvalues less than one in absolute value. A sufficient condition, which is easier to understand, states that the matrix **A** has to be strictly diagonally dominant,

$$|a_{ii}|>\sum|a_{ij}| \tag{2.5}$$

It means that the sum of absolute values of all nondiagonal entries in each row must be smaller that the absolute value of the diagonal entry. We note that this condition is automatically satisfied for a resistive network in nodal formulation.

The second, Gauss-Seidel method, is better. When any of the equations is solved, the result is substituted immediately into all equations below. The method can be expressed as

$$(\mathbf{L}+\mathbf{D})\mathbf{x}^{k+1}=\mathbf{b}-\mathbf{U}\mathbf{x}^{k}$$

or

$$\mathbf{x}^{k+1}=(\mathbf{L}+\mathbf{D})^{-1}\mathbf{b}-(\mathbf{L}+\mathbf{D})^{-1}\mathbf{U}\mathbf{x}^{k} \tag{2.6}$$

On the right we can define the matrix

$$\mathbf{M}_{GS}=(\mathbf{L}+\mathbf{D})^{-1}\mathbf{U} \tag{2.7}$$

Its eigenvalues must again be less than 1 in absolute values to guarantee convergence. This method converges faster than Gauss-Jordan. In fact, if the original A matrix can be reordered into a lower triangular form, then the solution is obtained in one iteration.

Relaxation can also be applied to nonlinear equations. In such case we have two iterative loops: the external one uses relaxation, the internal one solves a single nonlinear equation. In the Gauss-Jacobi method we operate in the internal loop on the jth equation

$$f_j(x_1^k \cdots x_{j-1}^k, x_j^{k+1}, x_{j+1}^k \cdots x_n^k)=0 \tag{2.8}$$

Only x_j changes, all other variables are held fixed, equal to the result of the previous external loop. The Gauss-Seidel iteration operates on

$$f_j(x_1^{k+1} \cdots x_j^{k+1}, x_{j+1}^k \cdots x_n^k)=0 \tag{2.9}$$

Only one variable, x_j, is changed in the inner iteration, the others are kept fixed. As in the linear case, the solution of the jth equation, x_j^{k+1}, is used immediately in the next equation.

Models of nonlinearities are usually such that each equation of the system can be written in the form

$$F[v(t),\dot{v}(t),w(t)]=0 \tag{2.10}$$

This means that a capacitor to ground appears at every node. The derivatives are replaced, for instance, by the backward Euler formula

$$x'_{n+1}=-\frac{1}{h}(x_n-x_{n+1})$$

which transforms the equations into a system of algebraic equations; they can be solved by one of the above methods [5].

Many additional considerations are needed for practical application. Relaxation iterations will be faster if each equation has only a few terms. This will be the case if there is no feedback and the signal processes in one direction from a node to the next one. MOS realization may have such form. Bipolar transistors will increase interconnections and as a result relaxation methods do not work well for bipolar networks. Inductors are always a problem. To get a system of algebraic-differential equations, inductors must be entered in impedance form. The inductance appears in the matrix on the main diagonal, but the its value may be very small. In order to keep this term dominant against the topological units which appear in the same row, the step size (which divides L) must be kept small.

Waveform relaxation is another method which attempts a rapid solution of large networks. Instead of stepping forward with small time steps for all equations, waveform relaxation integrates each equation over an interval. Let all nodes have a capacitance to ground and let all nodes have zero initial voltages. The derivative of the voltage, due to the presence of the grounded capacitor, is replaced by the Euler backward formula and the resulting nonlinear equation is solved. The integration is performed and the waveform is stored. Since the equation is nonlinear, we can use the Newton-Raphson iteration and drive it to convergence in each step. The same is done for the second equation, but the first variable already has a waveform, and its values are used. As one goes through the system, more and more waveforms are available from the previous solutions. The method is thus a modification of the Gauss-Seidel iterative scheme. It has some advantages as well as some disadvantages. Since the waveforms have to be stored, the requirements on storage grow rapidly. On the other hand, each equation can select naturally its step size and can exploit latency. The time steps time can differ for each equation and interpolation of the results must be used.

Many modifications of the above schemes can be devised. For instance, one does not have to work with one equation, but rather with small subnetworks. Each such section can be integrated as an analog network, with LU decomposition and step control. Relaxation is used only for the coupling of the subnetworks. Rearrangement of the equations according to the progress of the signal will also be useful; in fact, all the comments we made above apply to waveform relaxation as well.

3. Switch level simulation

Logic networks are usually large and their size prevents the use of classical simulators. In such case the only possibility is to simplify the transistor models to the extent that fundamental functions of the network can be tested, but all details are eliminated. Any number of such simplifications can be invented and programmed, but only after the whole work has been done can one be sure that the results correspond to the expectations.

We will introduce one simplification which proved useful. Let every node of the network have a capacitance to ground. This was already the case in relaxation methods. Suppose now that we model the MOS transistors by several levels of *constant* current, as sketched in Fig. 1. With these limitations the network consists of a number of current sources delivering constant currents. As a result, nodal voltages are known to be piecewise linear segments. The step size is not controlled by numerical integration but by simple evaluation of the next change of current level. Each change of current level results in a point which can be connected to the previous point by a straight line. Principles of the method can be found in [6,7,8].

4. Functional analog simulation

During the development process, the network is first known in its functional description, but not in terms of its transistor realization. In this case testing of the design has to be done by functional simulation where the gates are defined by their logic functions. Certain parameters which depend on technology must also be used.

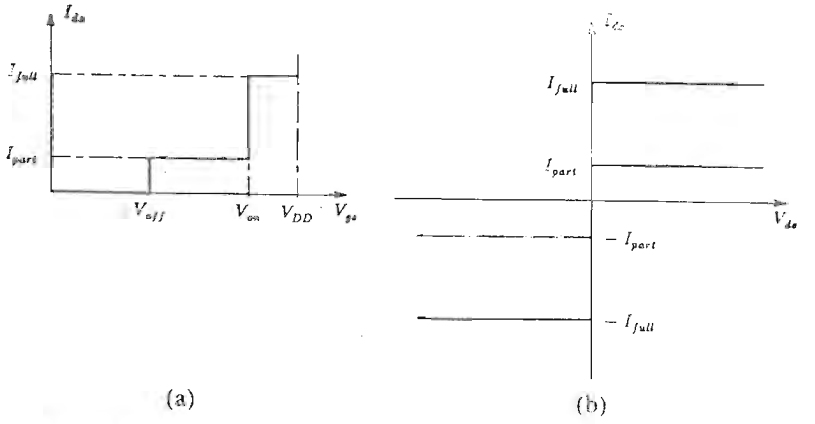


Fig 1. Characteristic of a transistor switch.
(a) Transfer characteristic,
(b) Output characteristic.

To use time functions we need to redefine the logic and allow for more than only two states of the Boolean algebra. This was done in [9] where it was proved that, at any instant of time, the output of an AND gate is defined by

$$v_{AND}(t)=\min\{v(i)t \mid i=1,2,...n\} \tag{4.1}$$

and the output of an OR gate by

$$v_{OR}(t)=\max\{v(i)t \mid i=1,2,...n\} \tag{4.2}$$

The NOT operation is defined by

$$v_{NOT}(t)=V_h+V_l-v_{in}(t) \tag{4.3}$$

where V_l corresponds to logic level zero and V_h to logic level one. They also correspond to the lowest and highest voltage used in the system realization.

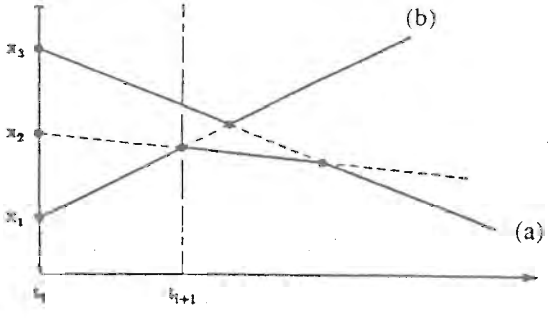


Fig 2. Input lines for three-input gate.
(a) output for AND operation,
(b) output for OR operation

We can select the type of curves of the time domain response and piecewise linear segments are clearly the simplest. As an example consider Fig. 2 where the thin lines represent three input signals of some gate. If the gate is an AND gate, then the output, according to (4.1), will be the lower bold line. It the gate is an OR gate, we get the upper bold line, compare with (4.2). The simulator moves in jumps from one intersection to the other one, connecting the points with straight lines.

For complete definition we must also consider the delay through the gate and limits on speeds of signals, given by technology. Details are in [9].

5. Switched networks

Switched networks have found two main applications: In communications as switched-capacitor networks and in electronically regulated power supplies, in various controllers etc.

Analysis of switched-capacitor networks is based on a very restricted set of elements: capacitors, ideal switches, independent or voltage controlled voltage sources. This restriction allows us the use charges and voltages as network variables. Time domain responses can be obtained without numerical integration. It is also possible to analyze such networks in frequency domain. The theory is fairly complicated and for details the reader is referred to [2,10]. Some important remarks are in order. It is a common practice to derive z-domain transfer functions. This can be done in a very structured way by using simplifications published in [11]. It should also be noted that the networks *do not* behave as digital networks, but rather as a mixture of analog and digital networks. Results obtained in the z-domain reflect actual responses only approximately because they are multiplied by a function of the type $\frac{\sin x}{x}$.

If all elements are allowed, the situation becomes much more complicated. For time domain analysis we must use numerical integration. The switching itself must be carefully considered, because Dirac impulses can occur. If the switching is periodic, frequency domain methods are available [12,13]. If internally controlled switches, like diodes or thyristors, are in the network, then only time domain integration remains as an option. Analysis and design of such networks was considered in detail in [14,15].

6. References

[1] E. Christen and J. Vlach: "NETOPT - A program for multiobjective design of linear networks. IEEE Trans. on Computer Aided Design, May 1988, pp. 567-577.

[2] J. Vlach and K. Singhal: Computer Methods for Circuit Analysis and Design. Van Nostrand Reinhold, New York 1983.

[3] A.R. Newton and A.L. Sangiovanni-Vincentelli: Relaxation-based electrical simulation. IEEE Trans. on Computer Aided Design, October 1984, pp. 308-331 (many references).

[4] J.K. White and A. Sangiovanni-Vincentelli: Relaxation Techniques for the Simulation of VLSI Circuits. Kluwer Academic Publishers, Boston 1987.

[5] S.P. Menzel and J. Vlach: A mixed-mode analogue and switch level simulator. International Journal of Circuit Theory and Applications, Vol. 19, No. 1, Jan.-Feb. 1991, pp. 35-50.

[6] G. Ruan, J. Vlach and J.A. Barby: Logic simulation with current-limited switches. IEEE Trans. on Computer-Aided Design, February 1990, pp. 133-141.

[7] G. Ruan, J. Vlach and J.A. Barby: Current-limited switch-level timing simulator for MOS logic networks. IEEE Trans. on Computer Aided Design, June 1988, pp. 659-667.

[8] C. Visweswariah and R.A. Rohrer: Piecewise approximate circuit simulation. IEEE Trans. on Computer Aided Design, July 1991, pp. 861-870.

[9] G. Ruan, J. Vlach, J.A. Barby and A. Opal: Analog functional simulator for multilevel systems. IEEE Trans. on Computer Aided Design, May 1991, pp.565-576.

[10] J. Vlach, K. Singhal and M. Vlach: Computer-oriented formulation of equations and analysis of switched-capacitor networks. IEEE Trans. on Circuits and Systems, September 1984, pp. 753-765.

[11] J. Vlach: Hand analysis of switched-capacitor networks. IEEE Circuits and Devices Magazine, November 1986, pp. 11-16.

[12] A. Opal and J. Vlach: Analysis and sensitivity of periodically switched linear networks. IEEE Trans. on Circuits and Systems, April 1989, pp. 522-532.

[13] A. Opal and J. Vlach: Consistent initial conditions of linear switched networks. IEEE Trans. on Circuits and Systems, March 1990, pp. 364-372.

[14] D. Bedrosian and J. Vlach: Time-domain analysis of networks with internally controlled switches. IEEE Trans. on Circuits and Systems-I. To appear in March 1992.

[15] D. Bedrosian an J. Vlach: Analysis of switched networks. International Journal of Circuit Theory and Applications, To appear in March 1992.

Workshop on Mathematical Modelling for Circuit Design

Sri Lanka, 1992

Tools for Analogue Circuit Design:

Present and Future

Robert Spence

Imperial College
London
England

Tools for Analog Circuit Design

Robert Spence
Imperial College, London

Simulation

Interaction

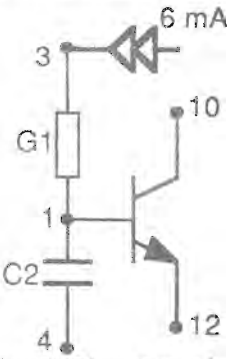
Sensitivity analysis

Optimisation

Tolerance Design

Automated Design

Computer-Aided Circuit Design:
Simulation



Circuit Diagram with node numbers

R 1 3 1000
C 1 4 1E-6
Q 12 1 10 50
etc

"Netlist" prepared by hand or as the output of a "schematic capture" package (e.g., MINNIE)

AC performance
DC performance
Time response

Simulator
e.g. SPICE

see below for detail

Equation formulation for DC performance

KCL @ node 1

$0 = G1(V1 - V3) + \ldots$

KCL @ node 3

$6 \times 10^{-3} = G1(V3 - V1) \ldots$

etc

If N nodes, then N linear simultaneous equations "nodal equations" in N unknown nodal voltages

$$\begin{bmatrix} I \end{bmatrix} = \begin{bmatrix} N \times N \text{ nodal conductance matrix} \end{bmatrix} \begin{bmatrix} V \end{bmatrix}$$

Solve (by Gaussian Elimination, LU Factorisation) to find V1, V2, ... Vn

Similar approach if nonlinear circuit, but iterations needed to move from 'guessed' answer to correct one.

THEN . .

RESULTS OF SIMULATION

V1 = 2.083 volts
V2 = ...

Equation formulation for AC performance

First: DC conditions provide small-signal parameters for nonlinear devices (e.g., g_m, r_{be}, r_o).

Then formulate *complex* nodal equations:

KCL @ node 1

$0 = j\omega C2(V1 - V4) + \ldots\ldots$

if N nodes, then we obtain N linear simultaneous equations in N complex voltages

$$\begin{bmatrix} I \end{bmatrix} = \begin{bmatrix} N \times N \textit{ complex} \text{ nodal admittance matrix} \end{bmatrix} \begin{bmatrix} V \end{bmatrix}$$

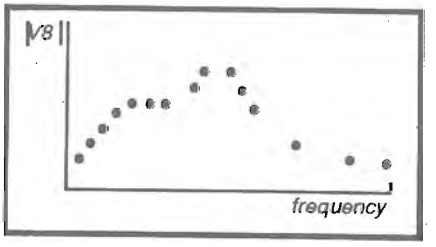
Select first frequency of interest

Solve (by Gaussian Elimination, LU Factorisation) to find V1, V2, ... Vn

then

Select second frequency of interest

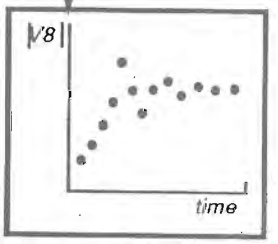
repeat for all frequencies



Equation formulation for Transient performance

More complex that for DC or AC simulation

For each time-point, it involves the solution of a linear resistive circuit



Interaction

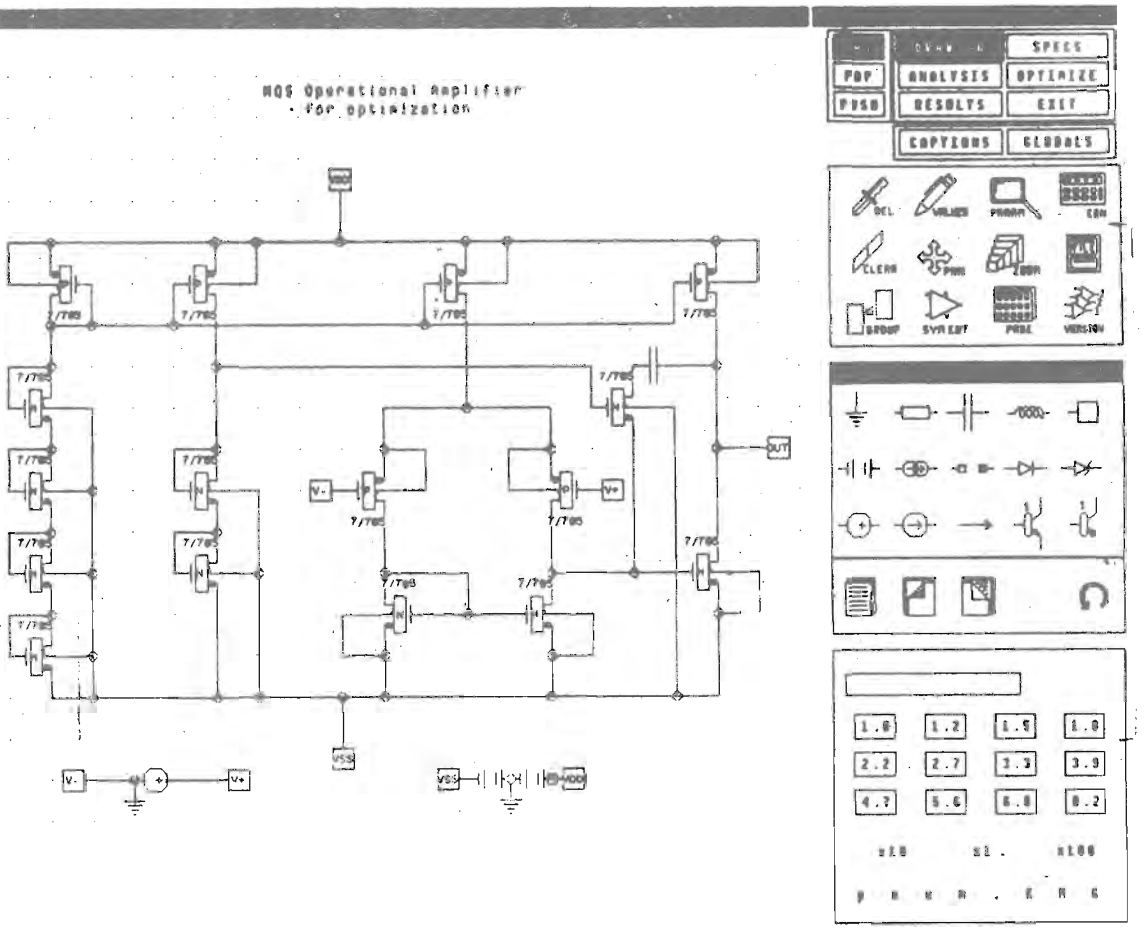
The use of a netlist to describe a circuit is extremely error-prone, and even a correct netlist cannot readily be interpreted by a human designer.

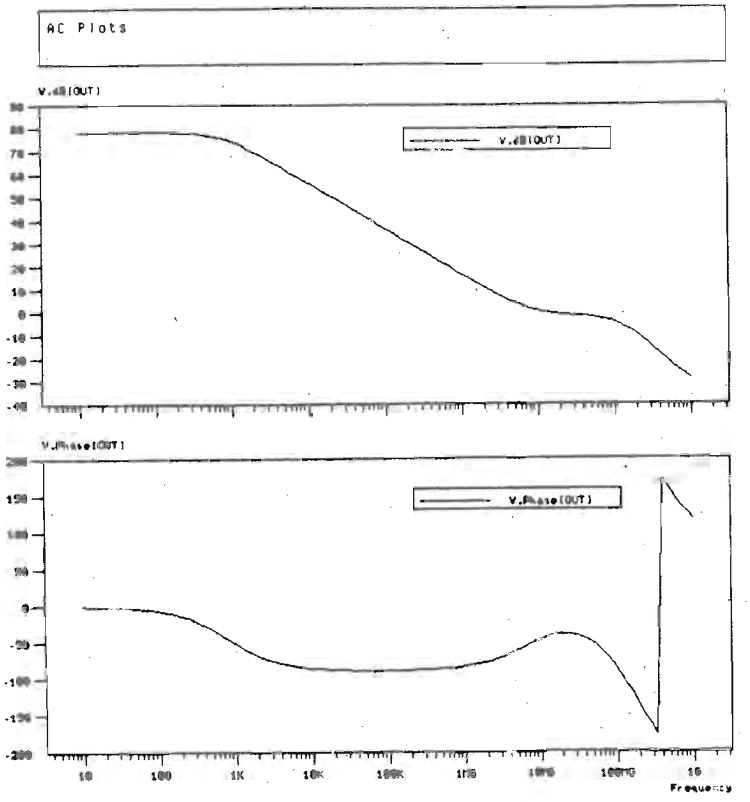
The human designer is used to sketching circuit diagrams on paper, so the MINNIE interface was developed to allow the designer to use the same method of circuit description when describing a circuit to a simulator.

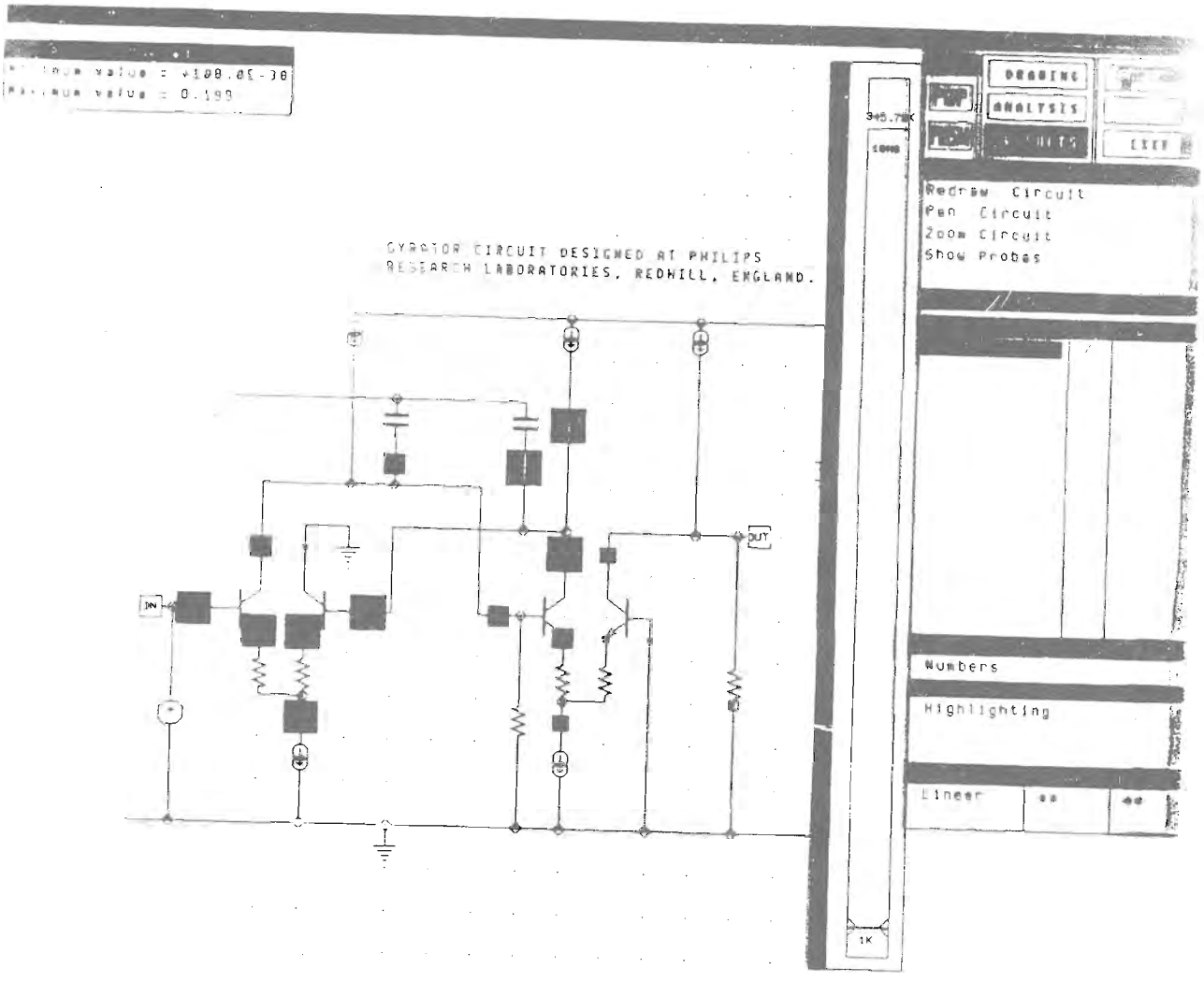
MINNIE, as well as allowing "circuit capture", offers a valuable post-processing facility. This post-processor takes the results of simulation and presents them, not only as ordinary graphs (which can easily be measured and resized, for example), but in a variety of innovative and effective ways.

The following illustrations were taken directly from the screen of the MINNIE display.

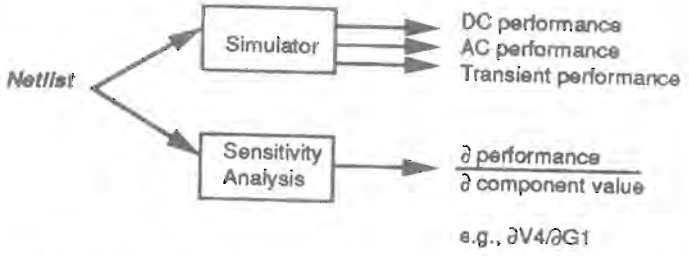
MINNIE is designed to be easily interfaced to any simulation package.


MOS Operational Amplifier
- For optimization
SPECS
POP
ANALYSIS
OPTIMIZE
PRSB
RESULTS
EXIT
OPTIONS
GLOBALS
OUT
V-
V+
VSS
VDD


AC Plots
V.dB(OUT)
V.dB(OUT)
V.Phase(OUT)
V.Phase(OUT)
Frequency


DRAWING
ANALYSIS
Redraw Circuit
Pen Circuit
Zoom Circuit
Show Probes
GYRATOR CIRCUIT DESIGNED AT PHILIPS
RESEARCH LABORATORIES, REDHILL, ENGLAND.
IN
OUT
Numbers
Highlighting
Linear
1K

Computer-Aided Circuit Design:
Sensitivity Analysis



Sensitivity analysis is carried out by numerical techniques similar to those used for the simulation of circuit performance. Sensitivity values can provide the circuit designer with useful information.

Sensitivity analysis is often carried out by the Transpose Circuit technique. For linear circuits this involves the analysis/simulation of two circuits:

Circuit 1 is the actual circuit of interest (See Figure A below)

Circuit 2 (the Transpose Circuit) has the *same topology as Circuit 1* but each component now has an admittance matrix which is the transpose of the matrix describing the corresponding component in Circuit 1. Thus, a resistor in Circuit 1 appears as the same resistor in Circuit 2 (similarly for L and C) but a transistor in Circuit 1 will be replaced, in Circuit 2, by a device described by the transpose of the Y-matrix of the transistor from Circuit 1.

The excitation of Circuit 2 must be appropriately arranged (see Figure B below)

Following the analysis of the two circuits, simple products of a voltage from Circuit 1 and the corresponding voltage from Circuit 2 provide the sensitivities of interest.
[see Brayton, R.K. and Spence, R. Sensitivity and Optimization, Elsevier, 1980]

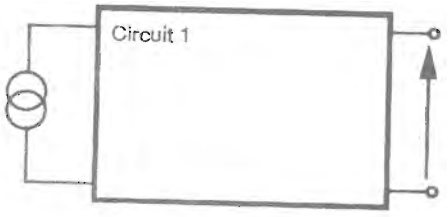


Figure A

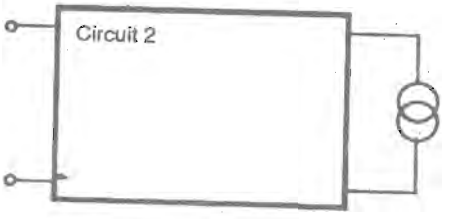
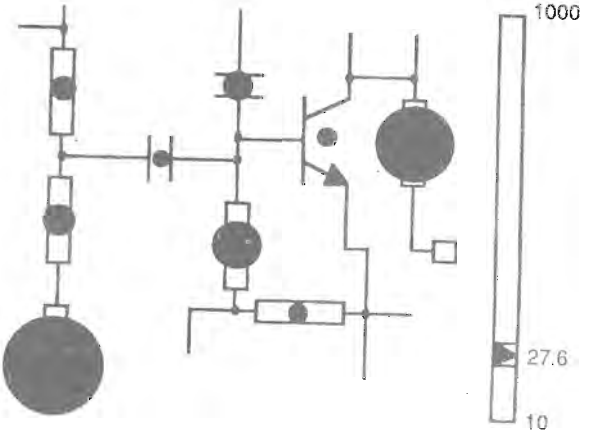
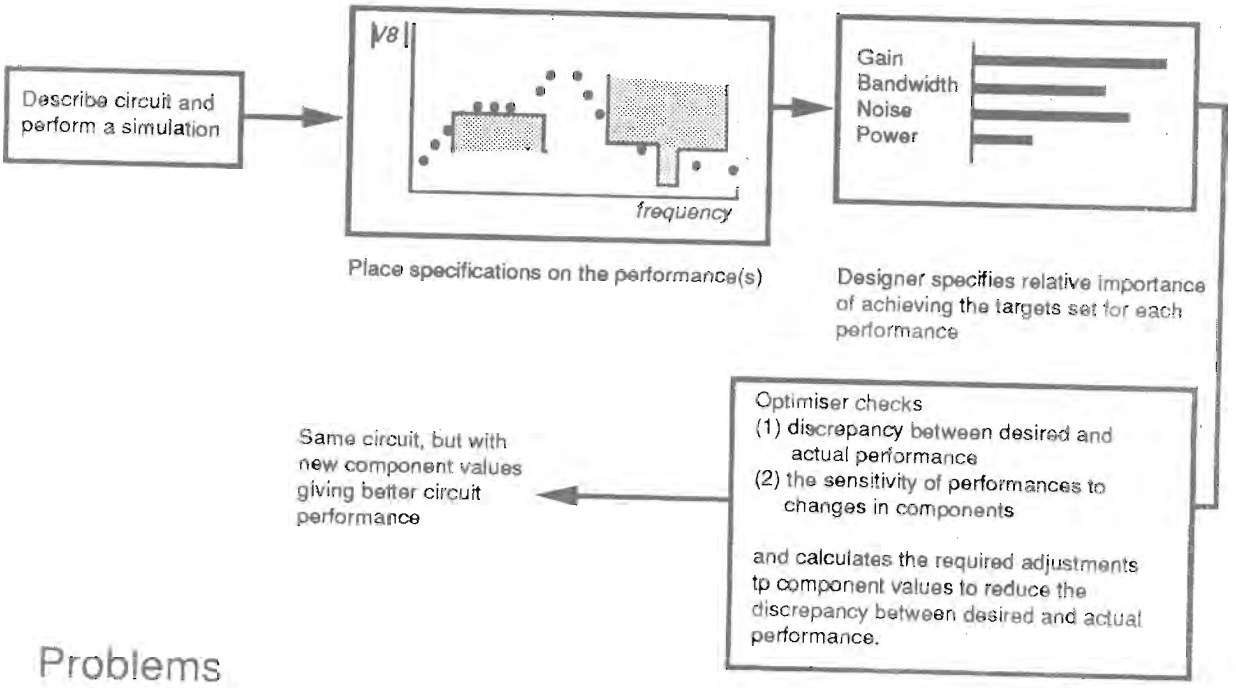


Figure B

Shown at right is one possible graphical presentation of the results of a sensitivity analysis carried out in the frequency domain. Circles superimposed on component symbols indicate the magnitude of the sensitivity of (say) voltage gain to changes in those components. Animation causes the display to change as the frequency range (shown at right) is scanned. If any frequency must be examined in detail, the frequency can be changed by moving the mouse on the scroll-bar.



Computer-Aided Circuit Design:
Optimisation (Automated Design)



Problems

Many optimisation algorithms exist, but are not easy to use. They can be made easy to use by the provision of a good interface between the human designer and the algorithm.

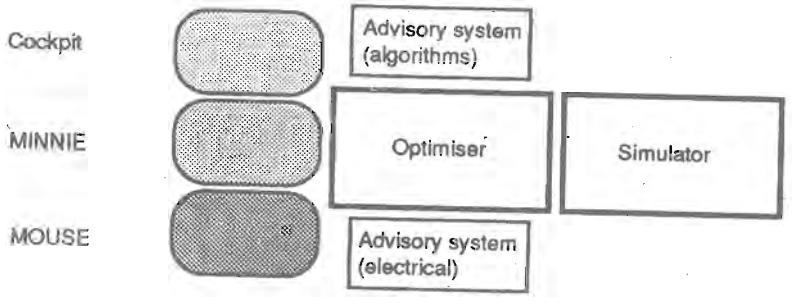
Solutions

CAD systems are being developed which provide a good interface between designer and algorithm, and which often contain other aids to design.

CoCo

One such system is called CoCo, for the control and observation of circuit optimisation.

CoCo also contains two expert advisory systems, one giving advice about algorithms, the other advice about circuits (e.g., "Transistor T23 is getting too hot").

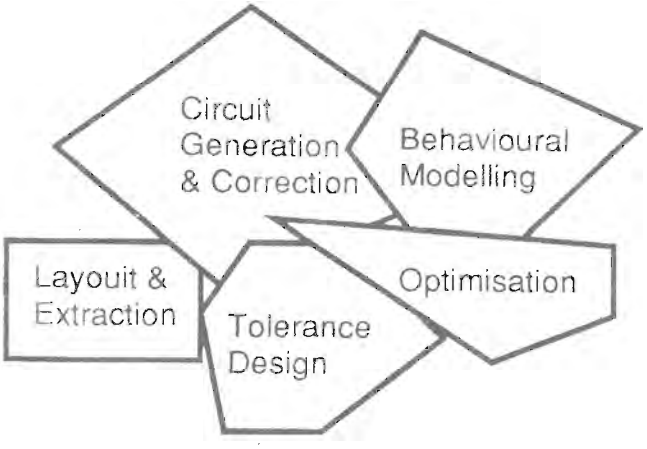


Automated Analog Design

Over the last 5 years, attempts have been made to fully automate the process of analogue circuit design. Systems which do so have been called "Analog Compilers". The input to an analog compiler comprises (1) specifications on the performance of the circuit required (" I need an amplifier with a gain of 60 dB, a slew-rate of . . .) and (2) details of the technology of the fabrication process (e.g., GaAs MESFETS, Silicon bipolar, . . .). The out put from the compiler is a layout which can be sent to the foundry.

CHIPAIDE

is an analog compiler proposed and demonstrated by Imperial College, London. It contains five interacting modules concerned with Circuit Generation and Correction, Layout and Extraction, Optimisation (see notes), Tolerance Design (see notes) and Behavioural Modelling.



Close interaction occurs between modules which comprise the CHIPAIDE analog compiler

NA-12-26

Visualisation within Engineering Design

Lynne Colgan*
Paul Rankin+
Robert Spence*

*Imperial College, London, U.K.
+Philips Research Laboratories, U.K.

Abstract

In the near future engineering design, conventionally carried out manually, will be enhanced by automated design techniques and supported by advice tendered by knowledge-based expert systems. To be effetive within such an environment, the designer must develop insight into a variety of functional relationships and constraints. By detailed reference to a novel exemplar design system we advocate visualisation as an efficient means of acquiring such insight.

1 Introduction

To say that engineering design is a complex activity is an understatement. Whether the artifact being designed is an oil-rig, a doll or a hi-fi amplifier, the designer is faced with a multitude of choices that must be made, trade-offs that need to be decided and factors that must be accounted for at different levels in the design.

Broadly speaking, design is carried out in two stages. The first is usually the proposal, by the designer, of the architecture or structure of the artifact which is most likely, in the designer's extrapolation, to be able to satisfy the requirements placed on its performance by a customer. In this paper we are concerned with the *subsequent* stage in which values of parameters of the constituent parts of the structure are adjusted to try to deliver a performance satisfying the market's requirements. A running illustration will refer to the design of electronic circuits, though the concepts are believed to be widely generalisable to other fields of engineering design.

An electronic circuit is an interconnection of components and, for decades, has been visualised in the form of a circuit diagram. In a modern computer-aided design (CAD) system such as the one described in this paper, the circuit diagram is sketched on a workstation screen, and values assigned to the components, by means of a mouse (Figure 1).

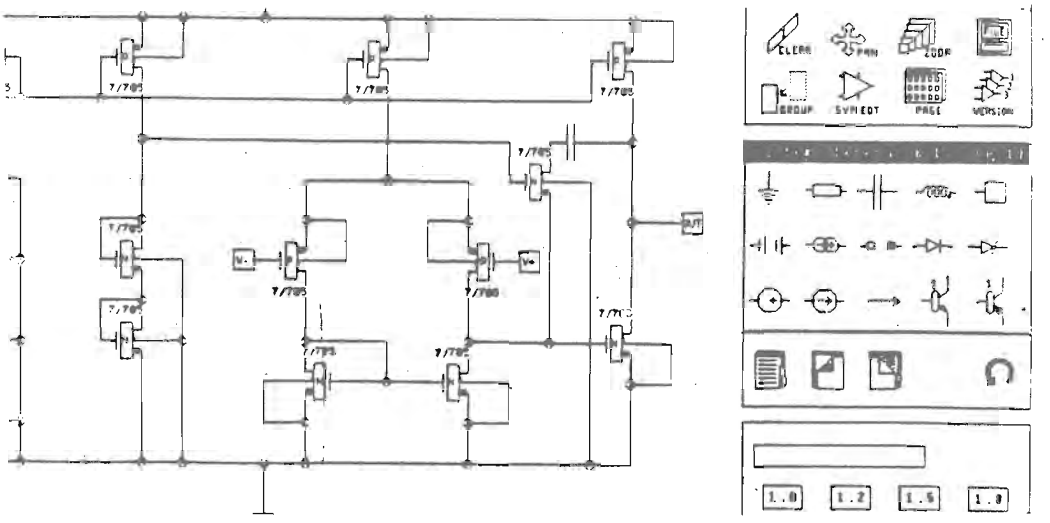


Figure 1 Detail of the diagram of an electronic circuit drawn on a workstation screen

In general all components influence all of the many voltages and currents - collectively known as **responses** - in the circuit, as suggested diagrammatically in Figure 2. Normally, such detail is not of prime interest: rather, it is the *integrated* effects of these responses, known as the **performances** of the circuit, to which the customer has referred in placing **specifications** on the circuit and which are of concern to the designer. Performances might include the amplification of the hi-fi, the output power and the level of unwanted noise.

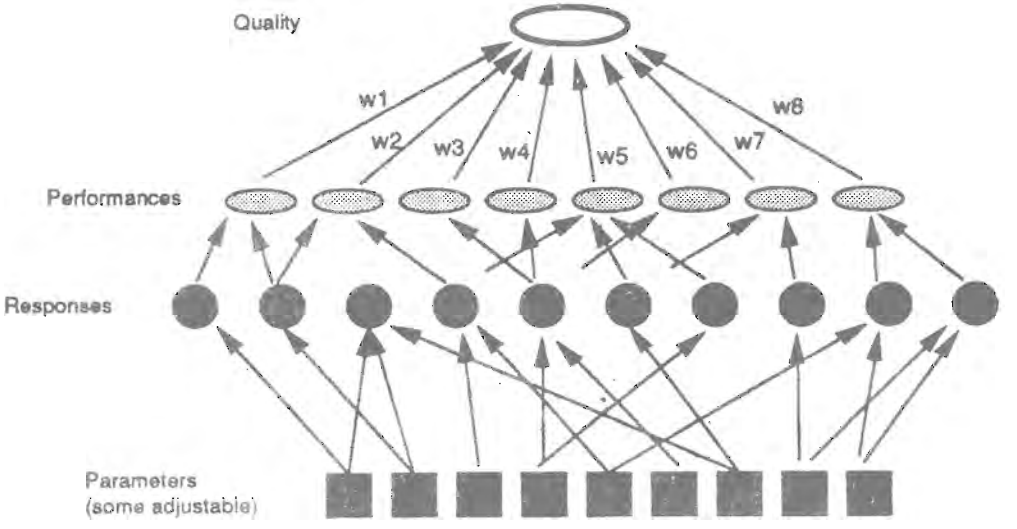


Figure 2 The relationships involved in engineering design.

Often, many satisfactory amplifiers can be designed that will satisfy the customer's specifications, in which case the question arises as to which is 'best' in some respect. Thus, an attempt might be made to minimise the amplification of the hi-fi, or maximise its output power, or minimise the level of unwanted noise. In the usual situation wherein the best of all worlds cannot be achieved, measures of importance (**weights**) have to be assigned to the manner in which the various performances contribute to some overall measure of the **quality** of the designed artifact (Figure 2).

The responses, and hence the performance and ultimately the quality of the artifact, can be varied by adjusting the values of certain components designated as **designable variables**. It is the task of varying the designable components in order to maximise the quality of the artifact that is **the activity of engineering design**, and which is exceedingly difficult even for an experienced designer. Contributory difficulties include, for example, trade-off effects where increasing a component value will degrade one performance while enhancing another.

Automated Design and Knowledge-based Advice

Before considering the potential of visualisation to ameliorate such difficulties, two important and recent advances in CAD must first be explained. One is the potential offered by numerical algorithms [1] capable of undertaking the ***automated*** **design** of an object. Such an algorithm senses (Figure 3) the existing quality of the artifact, computes - from its mathematical model of the artifact - a better combination of component values, and assigns those values to the components: iterative application can result in a considerable improvement in artifact quality. The other advance takes the form of an **advisory system**, based on a knowledge-base, which, like an experienced technician, relieves the master designer of tedious attention to design details.

Of course, effective use of specialist tools for automated design entails numerical knowledge which the engineer rarely possesses. A second, expert system to encapsulate the mathematical knowledge and numerical skills to run these CAD tools is implied.

Customer's Requirements

For the algorithm to be able to carry out automated design it must, in effect, be told what

performances are required from the artifact, and their relative importance. Again, visualisation is already inherent in the conventional manner in which these requirements are stated (e.g., upper and lower bounds on performance as in Figure 4) but is now additionally provided by the 'meter' metaphor (Figure 5) as well as a new slider bar presentation (Figure 6) of the relative weights placed on particular performances.

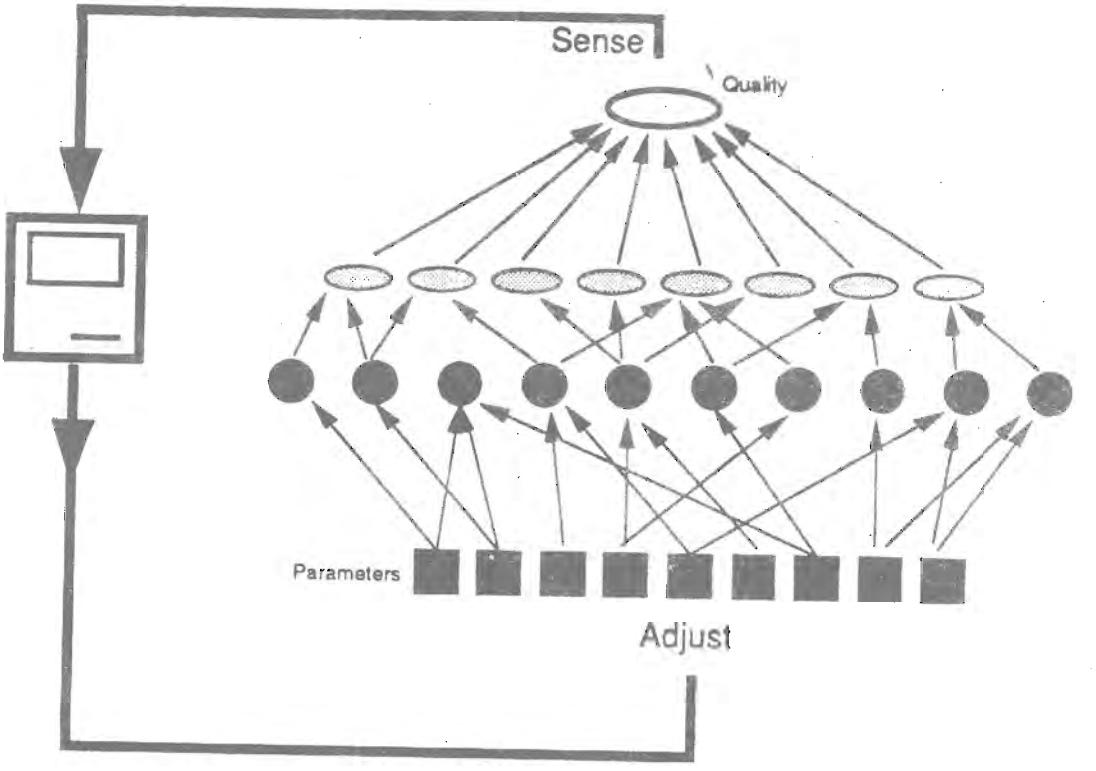


Figure 3 The automation of Engineering Design

Priorities

The designer can express his or her concern with performance in three essentially different ways. Certain performances are viewed as hard **constraints** ("these have *got* to be achieved"), others as soft **objectives** ("get as near to this ideal as possible"): both can be handled by an automated design algorithm. The third category is that of **alarms** which are indirectly activated by the knowledge-based system acting as an intelligent assistant, drawing the designer's attention to violations of good or common-sense design practice. The designer has the discretion to ignore or take account of these warnings.

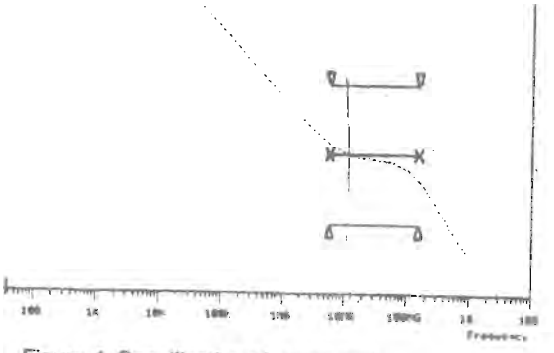


Figure 4 Specification of upper and lower limits of performance

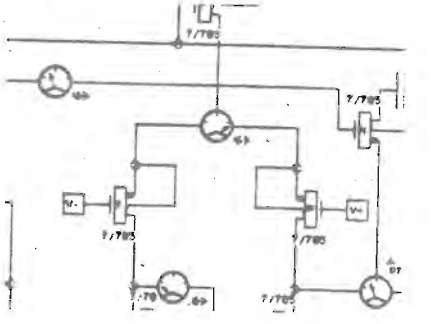


Figure 5 Use of the 'meter' metaphor to represent performance requirments

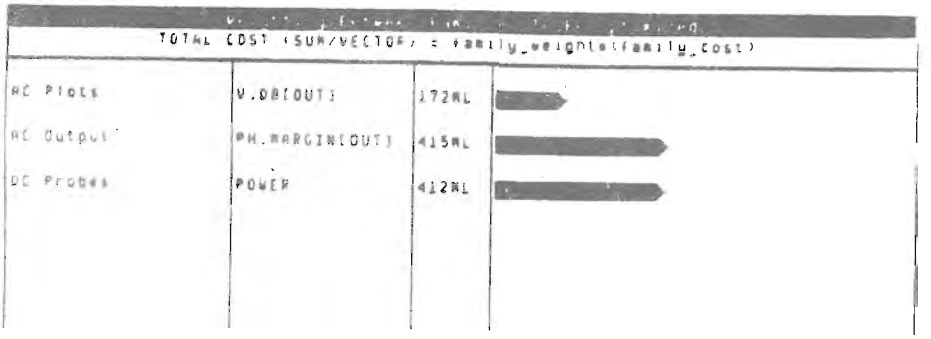


Figure 6 The selection of relative weights on performances

2 The CoCo Design System

An exemplar system called CoCo - for the Control and Observation of Circuit Optimisation - has been built [2] to support the designer of analogue electronic circuits by providing both automated design and advice [3] from two knowledge sources: one regarding electrical expertise and the other expertise with regard to the numerical aspects of automated design. It is relevant to this paper to remark that CoCo is *not* a 'toy' laboratory system: its use by industrial designers has required extensive software engineering of about 400,000 lines of code in 5 languages, calling for the investment of person-decades of investment at a total cost probably in the region of $2m. Only with such a realistic system can the innovations in visualisation described later be effectively evaluated in an industrial setting.

Perhaps the major focus of the CoCo system has been its concentration on providing effective interfaces with the human designer, these interfaces having a high graphical content and having evolved through careful user studies. Two such interfaces, for describing the designed object and expressing performance requirements, have already been illustrated in Figures 1, 5 and 6. However, the interface principally concerned with visualisation is called the **Cockpit** in view of the analogy with an aircraft cockpit: the designer uses the Cockpit in automatic pilot mode until he or she wants to interrupt and take control. In this way, the designer guides the system through the problem space until an optimum design for the artifact is found. The primary function of the Cockpit is to allow the designer to *monitor* the automated design process, to give an *overview* of design improvements and facilitate *interruption* and *interaction*.

3 The Cockpit

The relation between the Cockpit display and the functional description of the process of design is aided by comparison of the Cockpit display (Figure 7) with the previous Figure 3. Various aspects of the Cockpit display are discussed below.

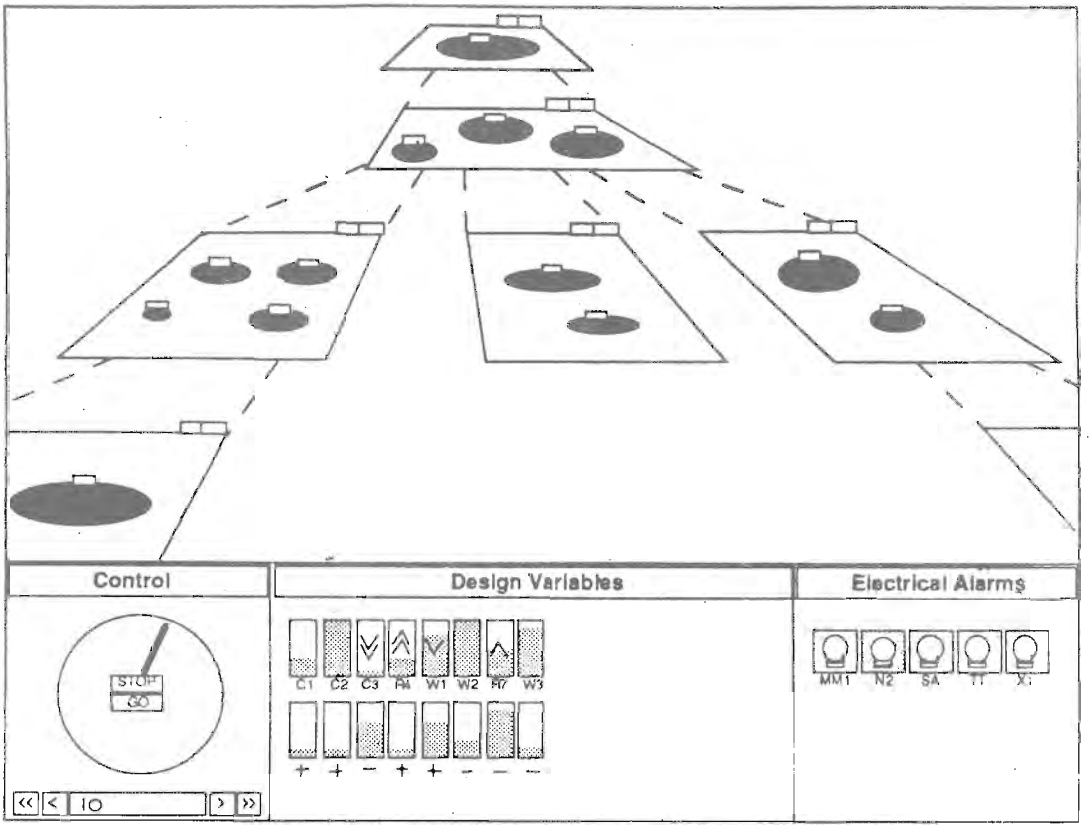


Figure 7 Outline Sketch of the Cockpit Display

The Hierarchy

The pseudo 3-D display in the Cockpit's main window makes explicit the hierarchical nature of the relation between the overall quality of the designed artifact (a single quantity represented at the top), the constituent performances specified by the customer (in the middle region) and the more detailed responses directly related to the component values (at the bottom). Scrolling, and customising the hierarchy by opening and closing branches, allows the designer to focus upon global or detailed design issues, an essential facility in engineering design.

The Sheets

The primitive elements from which the hierarchical display is constructed are known as **sheets**, on each of which is displayed one or more circles. Consider, for example, the topmost sheet. Since this sheet represents a single, scalar measure of the overall quality of the designed circuit, it contains only one circle. The size of the circle is a measure, not of the quality of the circuit but, rather, of an inverse measure - the *discrepancy* between desired and already achieved performance: the advantage of this measure is that, as ideal performance is approached, the size of the circle tends to zero.

Qualitative and Quantitative Display

As design proceeds, whether by manual and/or automated adjustment of component values, successful steps will be reflected by a reduction in size of the circle. The designer may wish merely to be aware of its qualitative variation: alternatively, quantitative detail may be required, in which case selection of the 'orientation' button (Figure 8a) will cause the view of the sheet to be changed from perspective to plan (Figure 8b), whereupon further menu selection can display either the numerical value of the objective or its history (Figure 8c). If a breakdown of the current sources for dissatisfaction in the design are required, the 'open/close' button triggers the unfolding of sheets one level lower in the hierarchy, on each of which one or more

circles can be displayed: for each circle the same facilities as just described are available including, of course, the facility to unfold more sub-sheets. The numerical weights relating sub-objectives to higher objectives can also be displayed.

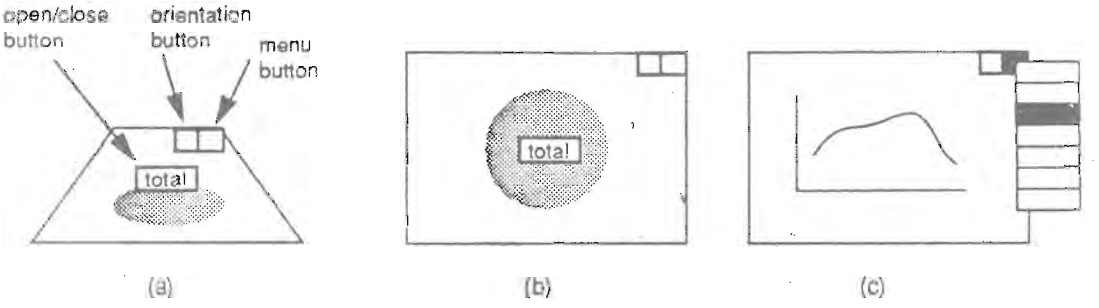


Figure 8 Sheet functionality

Objectives, Constraints and Alarms

Objectives values, encoded as circles, are not the only information displayed on the sheets. As discussed earlier, design requirements take three forms, hard constraints, soft objectives and alarms, all of which are typically relevant in a given design. The approach taken is to display all three on the sheets, though in a prioritised order. If a constraint is violated at a particular level of the hierarchy a rectangle appears at that and all parent levels, and suppresses any objective displays until the constraint is satisfied. With this arrangement a constraint violation can be traced down the hierarchy to see where the problem has arisen. Alarms carry the lowest priority since the designer can exercise discretion over conformance to the rules they represent. They are indicated on the sheet on which they occur, without suppressing either constraint or objective symbols. Since a sheet on which an alarm has appeared may have been folded away or scrolled out of sight, its activation is indicated in the Electrical Alarms window on the lower right of the Cockpit (Figure 7).

Sheet Geography

In passing, we note that the objective circles embody a visualisation technique first introduced by Sir Edward Playfair [4] in 1801, and more recently [5, 6] in 1970 and 1977 with added interaction and dynamics.

As one proceeds down the hierarchy it is likely that the objectives are associated with specific locations in the circuit being designed. Such spatial significance can also be preserved by displaying the objective circles on a stylised diagram of the circuit.

Designable Components

In the case of automated design, the designer must be kept aware of the manner in which components are being adjusted and the role they play in design improvement. It is for this reason that the lower central window of the Cockpit displays (Figure 7), as bars, the permissible ranges, initial values, current values and (as indicated by the number of superimposed chevrons), the rate of change of values of each designable component. Thus, if the designer notices (or is alerted by an alarm) that a component value is frequently stuck against one of its bounds, he or she may decide to re-run the automated design with a different bound, or one less design variable, or a structural circuit change. As for the sheets, pull-down menus allow the examination of the history of a designable component and many other details of interest.

The bars representing the values of designable variables are not provided simply for observation during optimisation: they could permit the adjustment of design variables through manual interaction.

Sensitivity Information

One of the aims of the circuit designer is to build up insight into the functional relation between

the adjustable components and the performances of the circuit, ranging from detailed responses to the overall quality. Again by the use of bars, and with identical interrogation facilities, information regarding the sensitivity of the overall quality to the designable variables is displayed immediately below the bars displaying their values. Knowledge of such sensitivities can significantly assist the designer in managing both automated and manual design.

Control Facilities

Facilities are provided which allow the designer to interact with the development of the design via controls that would ordinarily be found on a video recorder. 'Stop' and 'Go' buttons switch the system from automatic pilot mode to manual control. Fast forward and rewind buttons enable design progress to be reviewed. Other buttons allow the designer to move through the process step by step in either direction.

4 Evaluation

Evaluation of the effectiveness of all the visualisation mechanisms associated with the CoCo system will take an appreciable time to plan and execute, simply because a fair evaluation is only possible through the use of the CoCo system by real circuit designers engaged in the design of real industrial circuits. Structured evaluations of the basic system for automated design, involving the recording of deigners' voiced thoughts and plans as a design proceeds, have helped our modelling of the cognitive processes involved, but industrial evaluation of the Cockpit is still in the planning stage. In parallel, however, and based on our own observations, we have initiated the implementation of a SuperCard-based experimental facility to enable off-line experiments involving subjects to explore what we believe to be fundamental aspects of Cockpit use.

5 Future Developments

We envisage a convergence between the visualisation techniques and user interface styles employed in the monitoring of automatic processes and plants and those used for computer-aided drafting and simulation in design, and a rich field for investigation.

We also anticipate the emergence of interfaces, similar to the Cockpit, which facilitate the definition of a design problem as it is articulated by the designer: the benefits of visualisation in such an activity appear attractive

6 References

[1] Brayton, R.K. and Spence, R. 1980: *Sensitivity and Optimisation*, Amsterdam, Scientific Elsevier.

[2] Colgan, L., Rankin, P.R. and Spence, R. 1991: Steering Automated Design, *Proc. Conf. on AI in Design*, Edinburgh, June 1991.

[3] Gupta, A. and Rankin, P.R. 1991: Knowledge assistants for design optimisation, *Proc. Conf. on AI in Design*, Edinburgh, June 1991.

[5] Spence, R. and Drew, A.J. 1971: Graphical exploration in electrical circuit and model design, *Proc. 2nd Man-Computer Communication Seminar*, Ottawa, NRC, 61-70.

[6] Spence, R. and Apperley, M.D. 1977: The interactive graphic man-computer dialogue in computer-aided circuit design, *Trans. IEEE on Circuits and Systems*, CAS-24, 2, 49-61

[4] Tufte, E. 1983: *The Visual Display of Qualitative Information*, Cheshire, Conn., Graphics Press.

7 Acknowledgements

Partial support for the work described was provided by the United Kingdom ESRC/MRC/SERC Cognitive Science/HCI Joint Initiative. We are especially grateful to Simon Harriss, Maddy Brouwer-Janse, Mark Apperley, Ashok Gupta and Keith Hollis for their many and varied contributions to our research.

Tolerance analysis and design of electronic circuits

by **Robert Spence**
Imperial College of Science and Technology

An unwelcome factor in the design of electronic circuits is the variation in component values resulting from the tolerances associated with manufactured components. The corresponding variations in circuit performance can often be great enough to cause the specifications of a circuit to be violated. Although appropriate corrective action is not easy to establish, with suitable software improvements can be obtained through tolerance design.

Problems

Specifications for the performance of an electronic circuit required in quantity are received from a customer . . . A trial circuit is designed . . . A simulation of the circuit reveals that, with the chosen component values, the circuit's performance satisfies the specifications . . . Mass production begins . . . Soon, measurements on sample circuits reveal that some violate the specifications: they will therefore be unacceptable to the customer . . . More extensive measurements reveal that only about 55% of the circuits pass the specifications.

A typical scenario? The underlying cause of the unacceptable circuits is not hard to find: the variations in component values induced by the *tolerances* associated with all manufactured components. Component variations within these tolerances cause corresponding variations in circuit performance, sometimes to the extent that the specifications are violated.

The appropriate corrective action is, by contrast, not easy to establish. For example, the scenario described above, although unexpected and apparently unacceptable, may even turn out to be optimum in some sense! Often, however, a first trial design is not optimum, and some redesign is required. Design with a view to minimising the unwanted effects of component tolerances is called *tolerance design*.

Questions

Despite a knowledge of the underlying cause, the failure of a mass-produced circuit to achieve 100% manufacturing yield immediately stimulates a number of questions. Prominent among these are the following:

- Can the manufacturing yield be increased by changing the nominal values of the components, their tolerances remaining fixed?
- Since the cost of a component typically decreases as the tolerance increases, is there some optimum choice of tolerances which, although leading to less than 100% yield, nevertheless minimises the cost of acceptable circuits?
- Which specifications are seriously limiting the manufacturing yield? Which should the customer be persuaded to relax?

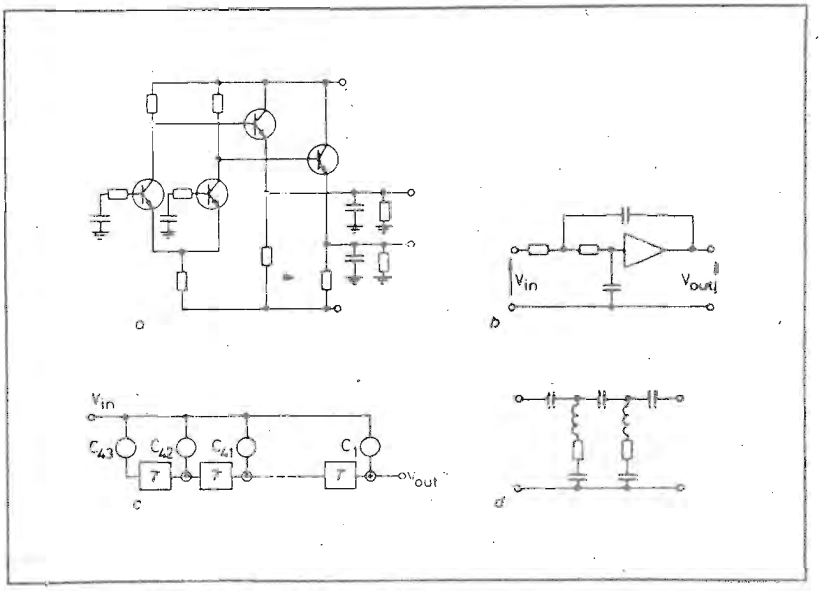


Fig. Successful applications of tolerance design
a An integrated current switch emitter follower for which useful reductions in power dissipation and worst-case delay were obtained
b A Sallen-Key filter whose performance variability was reduced
c An integrated transversal filter whose manufacturing yield was increased
d A highpass filter whose unit manufacturing cost was reduced

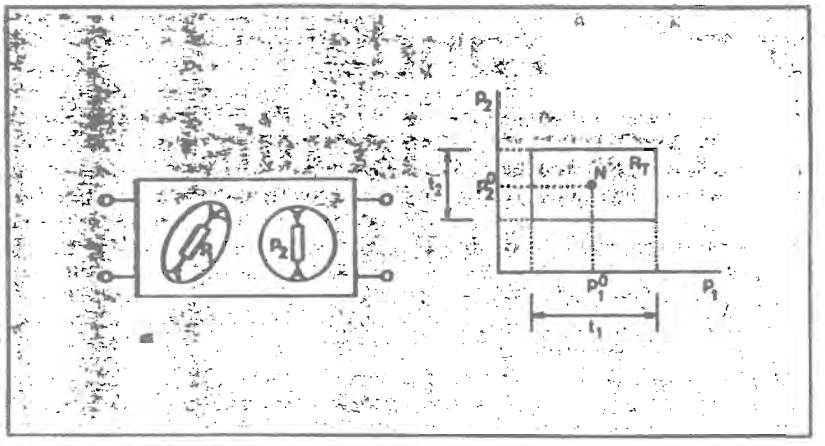


Fig. 2 Circuit with toleranced components
a Illustrative circuit containing two toleranced components
b The tolerance region in component parameter space, defined by the tolerances of the two components in Fig. 2a

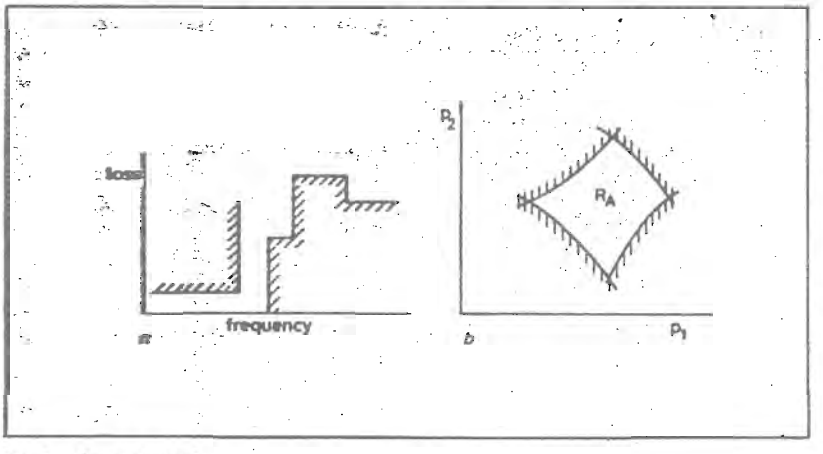


Fig. 3 The customer's specifications
a Specifications on the performance of the circuit of Fig. 2a
b The region of acceptability in component parameter space

- Is it possible to redesign the circuit so that the performance spread over the manufactured circuits is reduced?
- Can all the components in the circuit which would it be economical to measure before possible inclusion in the manufactured circuit with a view to maximising the yield?

These are just a few of the questions that a designer may ask, and for which the answers are often crucial to the economics of circuit manufacture. In what follows we shall show how the answers can be obtained by suitably designed software.

Capabilities

For the moment we shall ignore the details of tolerance design and how it is carried out, and first examine some examples of successful tolerance design. These examples will help to set the later discussion of design methods in context.

The circuit shown in Fig. 1a is an integrated current switch emitter follower. With its original component values it was designed by a very experienced integrated-circuit designer. Nevertheless, following the automated choice of new nominal component values, the power dissipation had been reduced by 28%, and the worst-case time delay by 30%, both very worthwhile improvements. The cost of this design exercise was approximately twice the cost of 67 analyses of the circuit [1].

The circuit shown in Fig. 1b is a Sallen-Key filter. It was required that the Q-factor should not only lie between 10 and 40, but should also exhibit a minimum variability between manufactured samples. Four iterations of a 'variability reduction' algorithm resulted in a very substantial reduction of the variance of Q [2].

The circuit shown in Fig. 1c is an integrated transversal filter involving coefficients determined by capacitors. The original design was put into manufacture and 600 samples were produced. The yield was 65%. Then the circuit was redesigned and again 600 samples were manufactured: the yield was found to have increased to 74%, an improvement that agreed closely with prediction [3].

The circuit shown in Fig. 1d is a high-pass filter. The cost of each component is inversely proportional to its tolerance. By means of an iterative algorithm the actual cost to the customer of each satisfactory circuit was minimised, the optimum design being one which led to slightly less than 100% yield. The cost of this exercise was essentially that of undertaking 600 analyses of the circuit [4].

We conclude from the examination of these four examples that worthwhile improvements appear to be obtainable through tolerance design.

Software requirements

Again, before plunging into the details of tolerance design, we consider what requirements the actual circuit designer (or his computer-aided design specialist) might place upon any methods that are available.

If the underlying theory of a method is easy to follow, the chance that the method will be put to practical use is considerably enhanced. Additionally, the task of software production should be kept as simple as possible. Furthermore, if existing tried and tested software can be exploited, then so much the better. While it is also desirable that the actual computational effort involved in tolerance design should be low, the state of the art is such that this cost is not trivial. Fortunately, ongoing research is attempting to reduce this cost [4].

Concepts, representations and terminology

For the newcomer to the field of tolerance design it is necessary to introduce some key concepts, to discuss the representation of tolerance-based effects and to define the terminology common to this field.

Parameter space

As an aid to exposition, we shall use as an illustrative example the trivially simple case of a circuit containing only *two* toleranced components, since the concepts thereby illustrated generalise, without restriction, to any size of circuit.

The circuit shown in Fig. 2a can be represented by a single point in *parameter space* (Fig. 2b), whose axes p_1 and p_2 are associated with the parameter values describing the two com-

parameters. For example, p_1 might be the resistance characterising a resistor, and p_2 might be the capacitance value characterising a capacitor. With each parameter will be associated a nominal value (p_1^0 and p_2^0), the two nominal values together describing the nominal circuit represented by the point N (Fig. 2b). In the realistic case of toleranced components, a tolerance range (t_1, t_2) will be associated with each parameter, thereby defining a tolerance region R_T within which the point representing any manufactured version of the circuit must lie (Fig. 2b).

Now consider the specifications imposed by the customer on the performance of the circuit. The specifications are not formulated in the context of the parameter space we have just discussed, but rather in an *output space* or *performance space*. For example, limits may be placed on the loss of a filter as a function of frequency, or the rise time of a switch (Fig. 3a).

In order to see whether and how the parameter tolerances most easily visualised in parameter space (Fig. 2b) cause the circuit to fail the specifications expressed in output space (Fig. 3a), some mapping from one space to the other must take place. For reasons which will gradually become apparent, we consider the specifications to be transformed from output space to parameter space (Fig. 3b), thereby defining a region of acceptability R_A. If the point describing a circuit lies within R_A, the circuit's performance will satisfy the specifications; if it does not, the circuit will have failed one or more of the specifications. Normally, the shape of R_A is quite irregular, with discontinuities and both concave and convex edges.

Before going further, it is important to note that the actual transformation of the specifications from output space to parameter space is *not* a trivial task: indeed, if it were, tolerance design would pose few problems and would be an inexpensive process to implement on a computer.

If we display both R_T and R_A in parameter space (Fig. 4a) we may find that R_T does not lie completely within R_A, in which case any manufactured circuits represented by points within the cross-hatched area will fail the specifications and the manufacturing yield (that fraction of mass-produced circuits passing the specifications) will be less than 100%. For the case in which a very large number of copies of the same circuit are manufactured, and where there is an equal likelihood of a parameter taking on any values within its tolerance range, the expected manufacturing yield is simply the ratio of two areas: the area of R_T that lies within R_A divided by the area of R_T. For different probability density distributions of the components, different 'weightings' will apply.

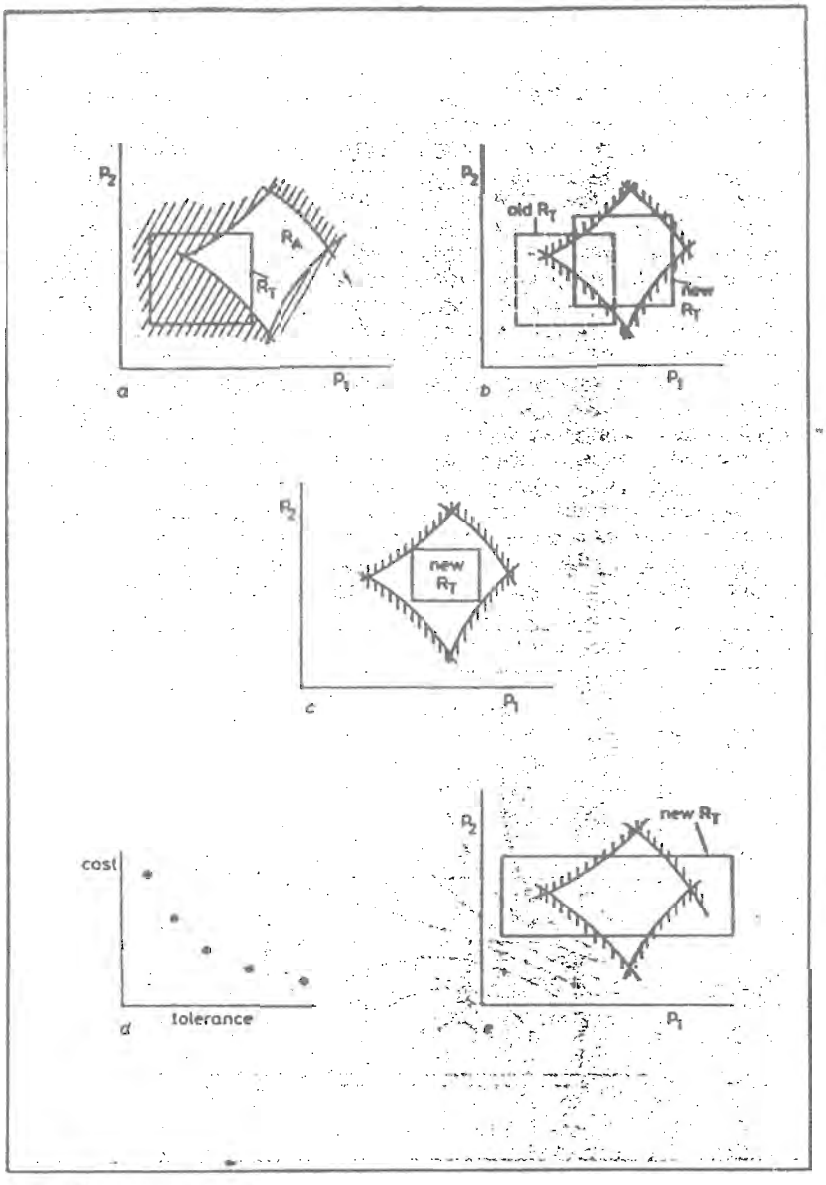


Fig. 4 Design centring and tolerance assignment

a The relative locations of the tolerance region and the region of acceptability determine the manufacturing yield

b A 'centring' of R_T within R_A increases the manufacturing yield

c Tighter tolerances increase the yield, but also increase the cost of each circuit

d A typical relation between the tolerance of a component and its cost

e The unit cost of this circuit may be less than that of Fig. 4c despite the lower yield

Design improvement

A simple means of increasing the manufacturing yield Y is suggested by Fig. 4a: it is to adjust the *nominal* parameter values, while leaving their tolerances fixed, so that R_T is more 'centrally located' within R_A (Fig. 4b). Such an adjustment of nominal parameters to increase manufacturing yield is called *design centring*.

Having centred the circuit, and achieved a greater yield, the question of parameter tolerances may arise. If these are open to choice, the designer will be aware that a tightening of tolerances (Fig. 4c) will lead to 100% manufacturing yield, although at a higher cost to the customer since the cost of a component is typically an inverse function (Fig. 4d) of its tolerance. Indeed, for this very reason, it may be more economical (Fig. 4e) to *widen* the parameter tolerances (i.e. increase t_1 and t_2) so that the cost of the reduced yield is more than offset by the reduced cost of the components. It is not difficult to

appreciate that there may exist an optimum set of parameter tolerances which ensures that the unit cost of a satisfactory circuit is minimised. The choice of parameter tolerances with a view to reducing circuit cost is called *tolerance assignment*.

For convenience, the activities of design centring (variable nominal, fixed tolerance) and tolerance assignment (variable tolerance, fixed nominal) have been separately identified and discussed, whereas they could be regarded as extremes of a combined process in which both nominals and tolerances are simultaneously adjusted.

The last aspect of tolerance design to be introduced here is *variability reduction*. The tolerance region R_{T_1} in Fig. 5 represents an initial design. The contours are those of constant circuit performance. Clearly, the circuit performance variability (from one manufactured sample to another) is reduced if nominal parameter values are adjusted (with fixed tolerances) to produce the new design represented by R_{T_2}.

Approaches to tolerance design

Broadly speaking, the past few years have seen the evolution of two principal approaches to tolerance design, reflecting two quite different ways of discovering the whereabouts of R_A in parameter space or, more precisely, the relative locations of R_A and R_T.

The *deterministic approach* attempts to construct a polygonal approximation to R_A by locating points on its boundary (Fig. 6). Knowledge of this approximation then allows the most appropriate design centre (i.e. nominal circuit) and set of parameter tolerances to be selected.

In the *statistical exploration approach*, the actual circuit manufacturing process is simulated by making a random selection of component values and predicting, by means of a circuit simulation package, the performance of each resulting circuit. The outcome is the location of 'pass' and 'fail' circuits within the tolerance region R_T (Fig. 7). Knowledge of their relative *numbers* allows the yield to be estimated, and knowledge of their *location* allows conclusions to be drawn concerning desirable adjustments to nominal parameter values and tolerances. Thus, in Fig. 7, it seems reasonable to move R_T in a 'north-easterly' direction to increase the yield.

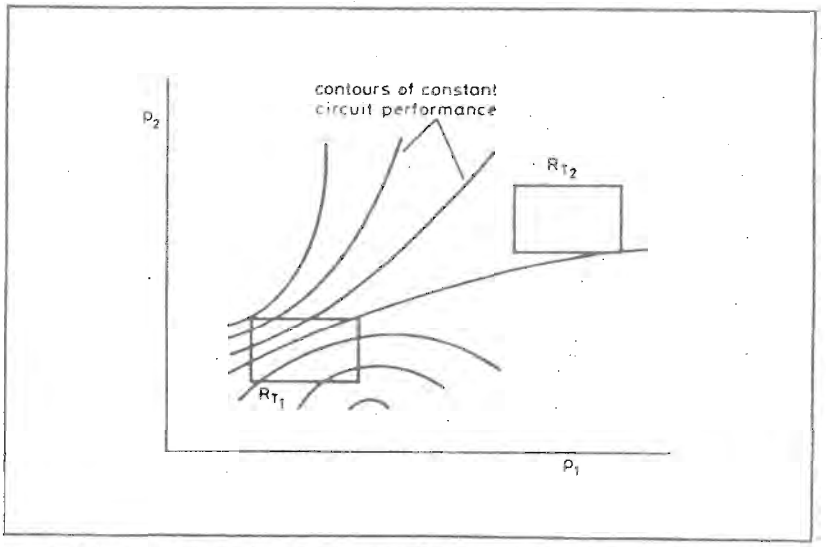


Fig. 5 Adjustment of nominal component values can reduce the variability of the performance of a mass-produced circuit

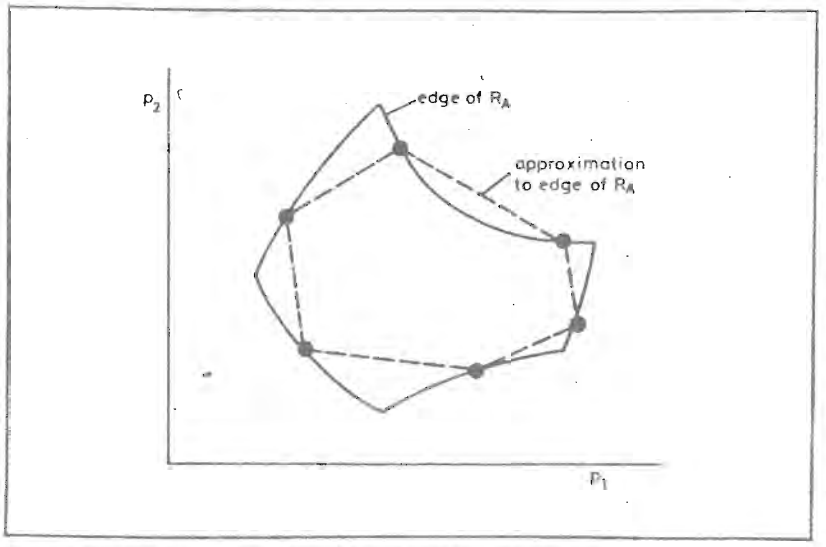


Fig. 6 A polygonal approximation to the region of acceptability

Statistical exploration

The deterministic approach has been widely reported in the literature, but suffers from the limitation that its computational cost increases very rapidly when more than about five or six components are subject to tolerances. This limitation is a very severe one, and we therefore concern ourselves exclusively with the statistical exploration approach. The advantages of this approach include the following:

- the computational efficiency with which it handles large circuits
- the opportunity it offers to make use of existing circuit analysis software
- the ease with which it can be understood, and the relative straightforwardness of its software implementation
- its ability to handle unusually shaped R_As, such as those shown in Figs. 8a and b.

Tolerance analysis

The first step in any attempt at tolerance design using the statistical optimisation approach is a tolerance analysis. Its primary objective is to estimate the manufacturing yield, although we shall see later that much additional and valuable design information can be derived from the results that are thereby generated.

The Monte Carlo method of tolerance analysis *simulates* the actual manufacturing process by selecting, at random, possible component values and computing the circuit's performance. A simple comparison with the specifications then establishes whether each 'sample' circuit is a 'pass' or a 'fail' (Fig. 9a). Typically, 300 samples might be analysed, although the actual number may vary widely. Simply by observing the fraction of the samples labelled 'pass', the yield can be estimated. Thus, if 150 out of 300 sample circuits passed the specifications, the yield estimate $\hat{Y}$ is 50%.

With some circuit simulation packages (for example ASTAP) a Monte Carlo facility is already provided, and is usually optimised with regard to computational cost. Otherwise, if only a circuit simulation package without this facility is available, it is a straightforward matter to implement a 'bolt-on' package to provide a Monte Carlo facility (Fig. 9*b*).

The smaller the number of Monte Carlo samples, the greater the possible error in estimating the yield by this method. The accuracy of a yield estimate can be quantified by establishing, theoretically, the likelihood that the actual yield Y will lie between two particular limits:

$$\bar{Y} - \varepsilon\sigma < Y < \bar{Y} - \varepsilon\sigma$$

where σ^2 is known as the variance of the yield estimate. If ε is (as is common) selected to be 2, then there is a probability of 95% that the true yield Y lies between the limits $\bar{Y} - 2\sigma$ and $\bar{Y} + 2\sigma$. Since the value of σ is given by

$$\sigma^2 = \frac{\bar{Y}(1 - \bar{Y})}{N}$$

where N is the number of Monte Carlo samples, we see that, following a Monte Carlo analysis, the yield estimate and its '95% confidence limits' can be computed.

From the above equation for the variance σ^2 we see that, for example, to halve the confidence interval, four times as many samples must be analysed at approximately four times the cost. That is the bad news. The good news is that the equation is independent of the number of components within the circuit. Thus, for a very large circuit, exactly the same number of samples are required to provide the same 95% confidence limits for a given estimated yield (the *cost* of each circuit analysis will, of course, increase).

Tolerance sensitivity

A Monte Carlo analysis can provide far more useful information than just a yield estimate. If the results of the analysis are stored we can easily and cheaply obtain some idea of how nominal values and tolerances can be adjusted to improve the design, as well as some insight into the trade-off between individual specifications and the manufacturing yield.

Parameter histograms

Let us assume that a Monte Carlo analysis has been carried out for a particular circuit, and that a database has been created, storing for each circuit sample the parameter values and the information (1 or 0) concerning whether the sample's performance was classified as 'pass' or 'fail'. From the information, and for a particular parameter p_i, two histograms can now be plotted showing the densities of passes and fails to a base of the parameter value (Fig. 10*a*). For ease of discussion the histograms are shown as continuous curves, even though in practice the p_i-axis will be divided into a number of class intervals so that the histograms are discrete in nature.

An immediate reaction to the pair of histograms shown in Fig. 10*a* might be that a greater proportion of 'pass' circuits is likely to be manufactured if the nominal value of p_i is reduced somewhat; to a value in the vicinity of the point X. Alternatively, if both upper and lower limits to p_i can be adjusted independently, then the yield might be increased if the upper limit is set to the value Z, arbitrarily chosen as the value of p_i for which the histograms intersect.

Aware of the design guidelines that such histograms may provide, Norman Elias (a pioneer of the statistical sampling approach to tolerance design) suggested that some measure of the overlap between the 'pass' and 'fail' histograms associated with a given parameter p_i 'is a direct measure of the influence of parameter p_i on circuit yield . . . The more sensitive the circuit yield is to any particular component, the less should be the overlap between values of that parameter in passing and failing circuits' [5]. Once a Monte Carlo analysis has been carried out, it is a relatively straightforward task to compute, for every parameter within the circuit, the corresponding value of the overlap, so that the designer's attention can be drawn to those components with small overlap.

If a particular parameter exhibits a small overlap, what steps should be taken by the designer? Elias describes a number of ways of exploiting the infor-

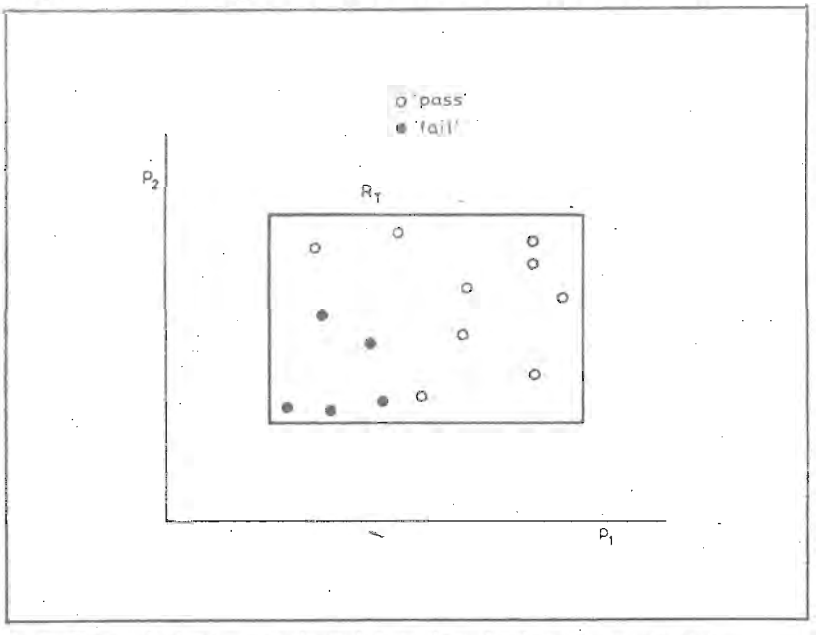


Fig. 7 The result of testing a random selection of simulated manufactured circuits

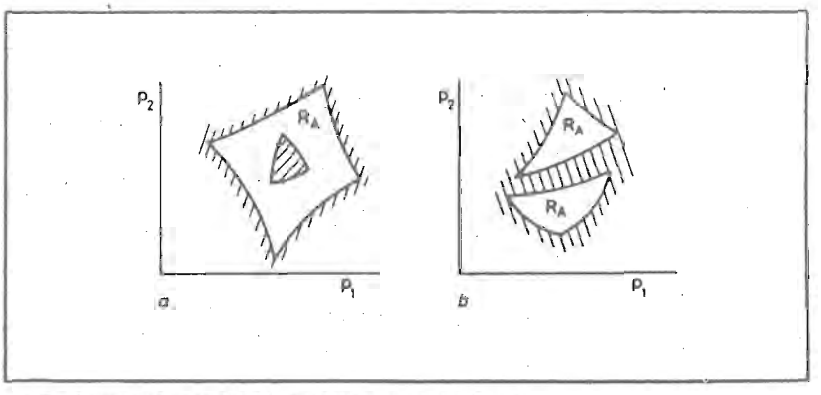


Fig. 8 Regions of acceptability can take on unusual shapes
a A more complex region of acceptability
b A region of acceptability composed of separate parts

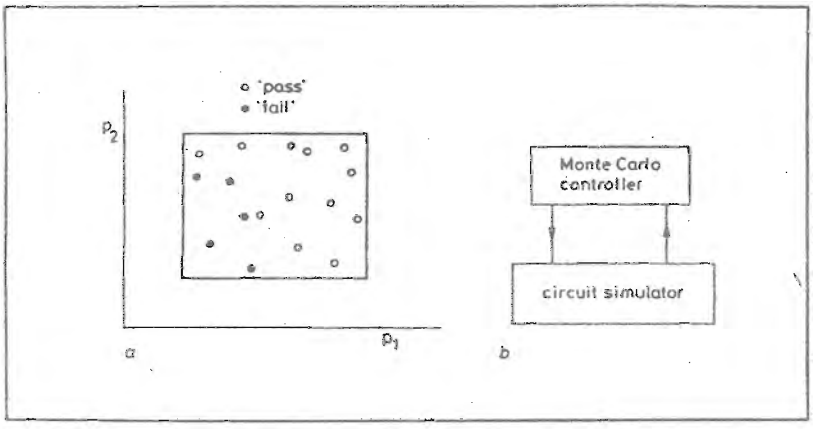


Fig. 9 Monte Carlo analysis

a The result of a Monte Carlo analysis

b An existing circuit analysis package being controlled by new software to carry out a Monte Carlo analysis

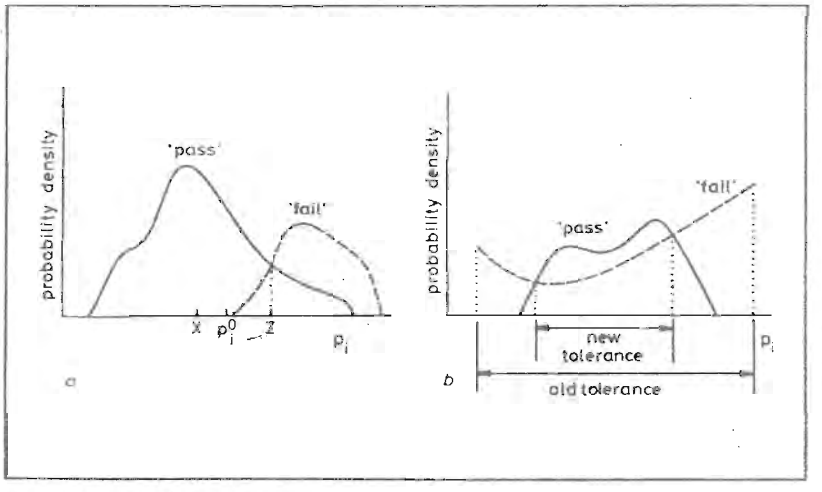


Fig. 10 The use of Elias plots

a Histograms showing the likelihood of a 'pass' or 'fail' circuit being associated with a given value of a component parameter

b Possible organisation of a tolerance-tightening scheme

mation contained within the histograms. For example, he describes a method of tolerance assignment (although, to be precise, it could only lead to a tightening of tolerance limits) in which the upper and lower limits on p_i are adjusted to the values for which $p_{pass} = p_{fail}$; in other words, the values for which the histograms intersect. An example showing old and new ranges for a parameter is shown in Fig. 10*b*. Although Elias used a statistically based proof that such a scheme would always increase the manufacturing yield, it was assumed that the limits on only *one* parameter were adjusted, all other parameter limits remaining fixed. There is therefore no assurance that the application of the technique simultaneously to more than one parameter will lead to a yield increase. In any case, simultaneous application of the tolerance-tightening algorithm to a number of parameters may result in 'overkill', in the sense that the increased cost associated with the new limits is disproportionately higher than the savings that accrue from the yield increase.

Elias has also shown [6] how the concepts discussed above can be applied to the selection of components before manufacture, this with a view to increasing the manufacturing yield.

Specification sensitivity

If, during the Monte Carlo analysis, and for each sample, the value of *all* circuit responses on which specifications are placed are saved, it is a straightforward matter to explore the trade-off between specifications and the manufacturing yield.

For details of the basis of this algorithm the reader is referred to the literature [7].

Design centring

A popular, effective and easily understood technique for design centring is called the 'centres-of-gravity' method: it is also straightforward to implement in software [8].

Imagine (Fig. 11*a*) that a Monte Carlo analysis of the original circuit design has already been carried out, and that 'pass' and 'fail' samples have been identified. A simple calculation will establish the location, in parameter space, of the centre of gravity P of the 'pass' samples and the centre of gravity F of the 'fail' samples (Fig. 11*b*). The line PF is taken as the direction in which the nominal circuit N should be moved, in the direction F → P: such a direction seems intuitively satisfying, but cannot be proved to be optimal in the general case. The best point along FP at which to relocate the nominal circuit is not so obvious. Satisfactory results have in fact been obtained by relocating N at P (Fig. 11*c*).

There is no assurance that this adjustment of the nominal component values (while keeping the tolerances fixed) will result in an increased yield. Therefore, after selecting the new nominal point, a fresh Monte Carlo analysis must be performed to see if the new yield estimate exceeds the previous one. If it does, the new nominal point may be accepted. If the yield estimate is still not 100%, an additional step of design centring may be undertaken. Thus, the centres-of-gravity method of design centring is essentially iterative.

An example of the progress of a design centring exercise is shown in Fig. 11*d*. In five iterations, each involving a Monte Carlo analysis using 100 samples, the yield estimate is seen to increase from 65% to 94%. Such an exercise is monitored by the designer, who decides when to terminate the progress, a decision normally affected significantly by economic considerations. In this example, a special 'technique, allowing 'old' Monte Carlo samples to be 're-used', reduced the number of circuit analyses required from 500 to 217 [8].

One extremely important advantage of the centres-of-gravity method is that the number of samples required for each Monte Carlo analysis appears to be independent of the complexity of the circuit: in most cases, the yield estimate has appeared to have achieved a value close to the maximum after about four, five or six iterations.

Another advantage of the algorithm is that it is not restricted to a particular class of circuit: nonlinear circuits are just as easily handled as linear ones. In fact, the basic requirement is some means of computing, from the parameter values p_i associated with a circuit, the performance F of that circuit (Fig. 12*a*). Then the only additional soft-

ware that is required is a parametric adjustment supervisor (Fig. 12*b*). Its function is, from a knowledge of nominal parameter values and tolerances, to generate the p_i associated with each sample of a Monte Carlo analysis, and to store information as to whether the resulting circuit performance F, when compared with specifications, is found to be acceptable or not. Thereupon the supervisor chooses new nominal parameter values according to the centres-of-gravity algorithm. The philosophy of a 'bolt-on' supervisor is particularly attractive to those designers who have invested a great deal of time, money and effort in becoming skilled in the use of a particular package (for example SPICE) and do not wish to amend it in any way, but who do wish to exploit the advantages offered by design centring.

One key factor of design centring has not yet been mentioned, and is associated with the fact that the Monte Carlo analysis performed at each iteration provides only an estimate of yield, not its actual value. Thus the possibility exists that the choice of a new nominal point may increase the yield estimate but cause a *decrease* in the *actual* yield! There is therefore a need to compute some measure of confidence in the correct ranking of two yield estimates: such methods are available [8]. Naturally, the designer is always anxious to pay for as small a number of circuit analyses as possible, in which case it is important to be aware of a technique known as *correlated sampling* [9]. If the sample points in the new tolerance region have the same relative spatial location as in the previous tolerance region, then the confidence in the correct ranking of the two yield estimates will be higher than if a new random sampling of the new tolerance region were employed.

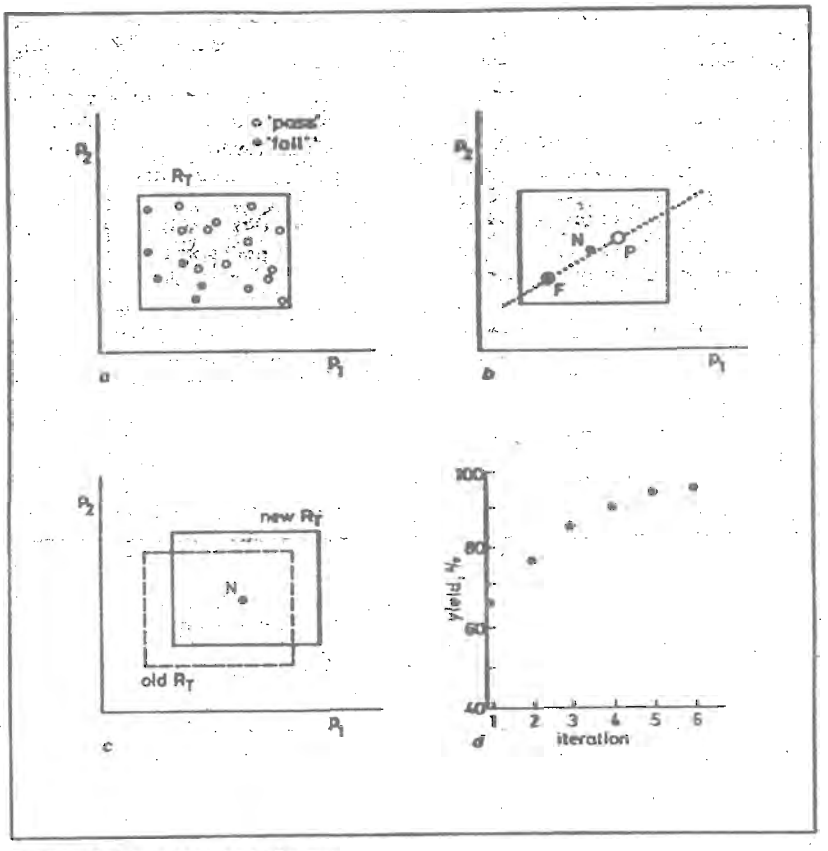


Fig. 11 A design centring technique
a The result of a Monte Carlo analysis of a circuit
b Identification of the centres-of-gravity of the 'pass' and 'fail' circuits of Fig. 11*a*
c A new tolerance region obtained by choosing as the new nominal design, the centre of gravity of the 'pass' circuits (see Fig. 11*b*)
d A typical outcome of a six-iteration design centring exercise

Tolerance assignment

The choice of component tolerances must take into account the cost of each component. Often the cost is inversely proportional to the tolerance, in which case the cost of an entire circuit can be expressed in the form

$$\text{cost} = A + \sum_{\substack{\text{all} \\ \text{components}}} \frac{a_i}{t_i}$$

where t_i is the tolerance associated with the ith component. However, if the manufacturing yield is less than 100%, this cost has to be recovered from the sale of only the acceptable circuits. Since they are a fraction Y of the total, the circuit cost C for each satisfactory circuit is

$$C = \frac{1}{Y}\left[A + \sum_{\substack{\text{all} \\ \text{components}}} \frac{a_i}{t_i}\right]$$

The task of tolerance assignment is to minimise C by the choice of suitable component tolerances, under the condition that the nominal values are fixed. Other realistic cost models are available, but in every case it is necessary, when attempting to minimise C, to have available some estimate of Y.

As is the case for design centring, a number of algorithms are available to assist the designer with tolerance assignment. The one we select here for illustration is chosen to acquaint the novice with a new class of algorithms within the statistical exploration approach.

The existence of this new class arises from our ability [1] to compute, *at very little cost*, the differential sensitivities $\partial F/\partial p_i$ of circuit performance F with respect to each parameter p_i. This ability leads to the possibility of replacing many of the costly circuit analyses demanded by a Monte Carlo analysis by a much simpler calculation. For example by using a truncated Taylor series approximation to the circuit response F,

$$F = F_0 - \sum \left.\frac{\partial F}{\partial p_i}\right|_{F_0} \Delta p_i$$

where Δp_i is the change in parameter p_i, knowledge of the value F_0 of F at one point in parameter space is sufficient to allow an approximate but *inexpensive* calculation of F at points in the immediate vicinity. For this reason a 'sensitivity based' approach to tolerance design has emerged, and will now be illustrated in the context of an effective method of tolerance assignment.

Consider a circuit containing two toleranced components. Their tolerances define, in component space (Fig. 13*a*), a tolerance region R_T. Assuming, as we do for tolerance assignment, that the nominal values of all parameters are fixed, R_T is equivalently defined by a single point in *tolerance space* (Fig. 13*b*).

With the initial choice of tolerances, the yield Y must first be estimated in order to determine the cost C from Eqn. 1. This is achieved by a conven-

tional Monte Carlo analysis. The effect on C of different combinations of tolerances is then examined in the following way. A new point A in tolerance space is selected randomly, although at a prescribed distance from the original point (Fig. 13c). This tolerance perturbation Δt_A from the original point defines a new tolerance region $R_{T(A)}$ in parameter space (Fig. 13*d*). Assuming that R_T and $R_{T(A)}$ do not differ widely, the truncated Taylor series approximation can be used to predict the value of F at any point within $R_{T(A)}$. The procedure adopted is first to ensure that the sensitivity $(\hat{c}f/\hat{c}p_i)$ — as well as F — is computed for each of the Monte Carlo samples within R_A. Then, within $R_{T(A)}$, the same geographical disposition of sample points is chosen that was used for R_T, thus simplifying the calculation of the estimated F values within $R_{T(A)}$ using the truncated Taylor series approximation. A subsequent comparison with the specifications allows the yield Y_A and cost C_A associated with the new point A in tolerance space to be estimated.

The estimate of cost is carried out for a number of points in tolerance space randomly chosen to lie on a hypersphere centred at the original point (Fig. 13e). With this information, a number of methods are available for choosing a new point in tolerance space representing a new — and hopefully improved — choice of component tolerances. In the tolerance assignment exercise illustrated in Fig. 1*d*, the least-squares method was used to fit a hyperplane to the cost estimates in tolerance space. In a direction opposite to the slope of this hyperplane a uniform line search was then undertaken (inexpensively, using the Taylor series approximation) to determine the minimum-cost point. Once a cost minimum has been located the tolerances are appropriately adjusted. According to the judgment of the designer, the process may be repeated until an acceptable design is obtained.

For the moment, the benefits that can accrue from the availability of inexpensive sensitivity information are tempered by the fact that not many simulation packages provide sensitivity information. For this reason, the approach of a bolt-on parameter adjustment supervisor is not so attractive.

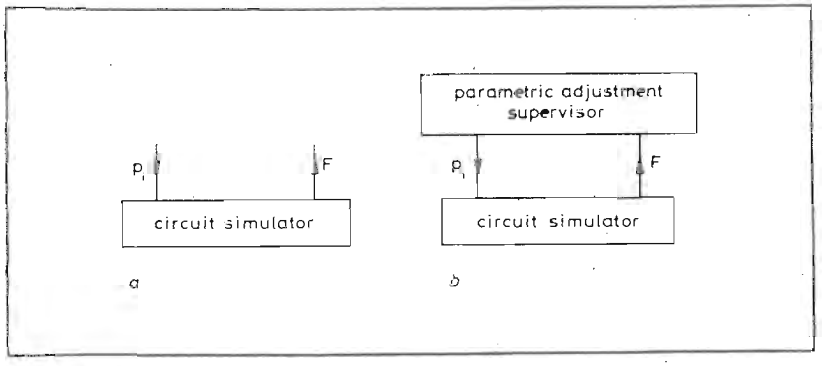


Fig. 12 The use of an existing circuit simulator for design centring
a Representation of a conventional circuit simulation package
b A 'bolt-on' parametric adjustment supervisor causes the responses of selected Monte Carlo samples to be generated, stored and used for design centring

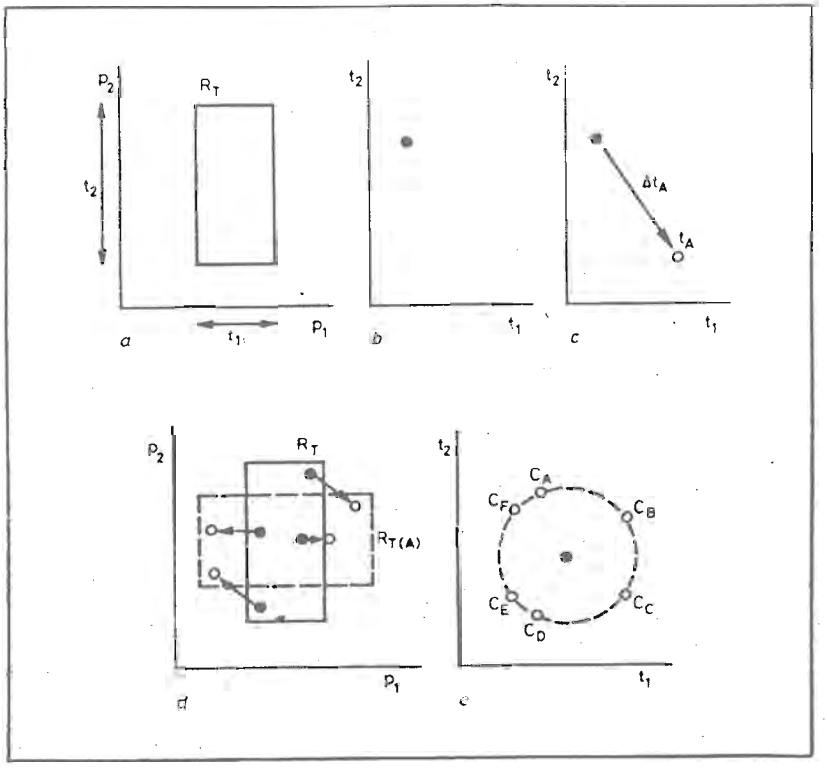


Fig. 13 An approach to tolerance assignment

Industrial application

Although research is still in progress, software packages for tolerance design are finding their way into industrial use; in some cases, a package has acquired the 'however-did-we-manage-without-it?' accolade. To our knowledge, only one commercially available package offers a tolerance design facility: it is SCADS, expected to be available in the summer of 1984 from I. P. Sharp for bureau use and eventually for an IBM PC.

Necessarily, we have had to discuss the topic of tolerance design largely in isolation; how would these techniques actually be used in practice? While there may be occasions when a full tolerance design exercise involving a number of iterations may be required, it will frequently be the case that the very crudest form of design centring will be used merely to see if any useful improvement is possible. For example, the situation typified by the point A in Fig. 14, and which resulted in a considerable loss of production to a company, could have been brought to the embarrassed designer's notice by a single Monte Carlo exploration involving about 20 to 50 samples, and a useful redesign could have been effected at what, by comparison, would have been a trivial cost. Even a move to a point such as B, which is still far from the design centre, would constitute an improvement.

Throughout this paper we have mentioned the potential economic advantage offered by an improvement in yield or a reduction in unit cost. But this advantage is usually impossible to quantify beforehand. If a circuit is already well designed, then *no* improvement can be brought about by tolerance design, and the major outcome of the exercise will be a measure of con-

fidence that the design is close to optimum in the tolerance sense. Even if the circuit is not well designed, any improvement may well be negated by the computational cost of carrying out the tolerance design. Just as for most other tools used by the designer, the decision when to use tolerance design will become more and more instinctive as experience is accumulated.

Since a tool can often become much more effective if it is used interactively, the programming of the human-computer interaction associated with tolerance design is of vital importance. For example, an ability to monitor easily the progress of a Monte Carlo analysis, or to effect a comparison with an earlier iteration, can help the designer to gain insight into what is happening, and thereby to guide the tolerance design. Such a facility can [4] lead to considerable savings in computational effort.

The future

Despite the results that have been described here, and many others reported in the literature, research in tolerance design is still proceeding. A major effort, for example, is being directed towards a reduction in the computational effort involved.

At one time, the number of circuit analyses required was sometimes measured in thousands. More recently, the number was in the low hundreds. Happily, methods are now becoming available involving considerably less than 100 analyses, although it must always be borne in mind that the accuracy of a Monte Carlo yield estimate is governed solely by the number of samples, and for 'reasonable accuracy' this number would rarely fall far below 100.

Another, largely unresolved, problem is associated with components for which only a discrete number of 'preferred' values and tolerances are available. A common approach is to ignore such restrictions and then, at the completion of design centring and tolerance assignment, to choose the 'nearest' values. Naturally, a new Monte Carlo analysis is required to estimate the yield.

Methods of tolerance design are now being extended and modified to suit the characteristic requirements of integrated circuits. An introduction to this direction of research is provided by Ref. 10.

To dig deeper

Apart from the References provided, two publications may be of interest to the reader who wishes to dig deeper into the subject. One is a special issue of *IEE Proceedings Part G* on tolerance analysis and design [11], which contains 13 papers dealing with various aspects of the subject, and provides an additional source of references. The other is a review paper published recently in the *IEEE Proceedings* [10].

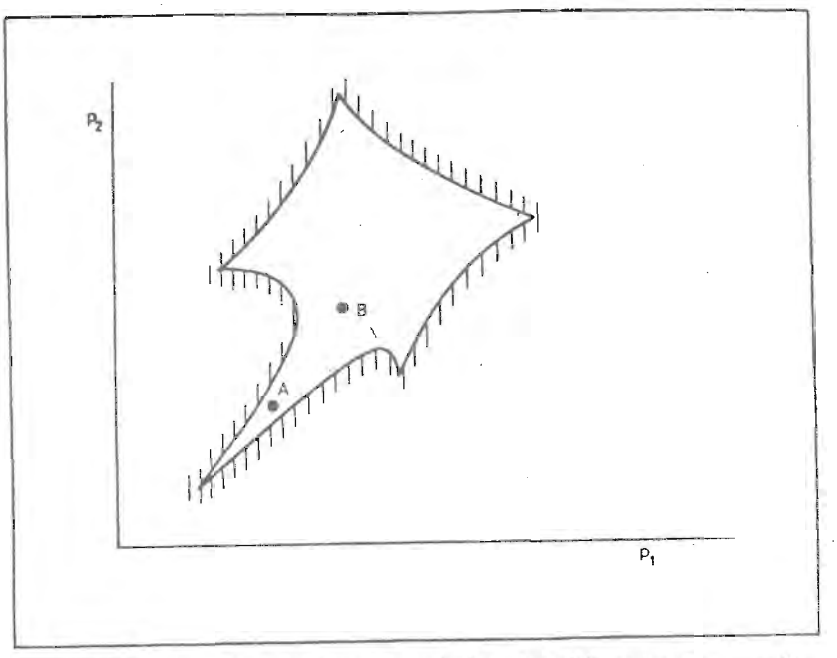


Fig. 14 A badly placed design A is too close to the boundary of the region of acceptability

Acknowledgments

The author's task has been assisted immensely by colleagues and acquaintances with whom he has collaborated in research or in the presentation of courses in tolerance design. They include Ajoke Ilumoka, Nick James, Ian Jones, Nicos Maratos, Paul Rankin, Randeep Soin and Eric Wehrhahn. Collaboration with Ajoke Ilumoka, Ken Chakawata and John Burgess on the implementation of tolerance design methods within the package SCADS is also gratefully acknowledged.

References

1 BRAYTON, R. K., and SPENCE, R.: 'CAD of electronic circuits: sensitivity and optimisation' (Elsevier, 1980)
2 ILUMOKA, A., and SPENCE, R.: 'Reduction of tolerance effects in the 2nd-order Sallen-Key filter', *IEE Proceedings Part G, Electronic Circuits & Systems*, 1981, **128**, (4), pp. 163–166
3 KNAUER, K., and PFLEIDERER, H.-J.: 'Yield enhancement realized for analogue integrated filters by design techniques', *ibid.*, 1982, **29**, (4), pp. 122–126
4 ILUMOKA, A., and SPENCE, R.: 'A sensitivity-based approach to tolerance assignment', *ibid.*, 1982, **129**, (4), pp. 139–149
5 ELIAS, N. J.: 'New statistical methods for assigning device tolerances', Proceedings of IEEE International Symposium on Circuits and Systems, Newton, MA, USA, Apr. 1975, pp. 329–332
6 ELIAS, N. J.: 'The application of statistical simulation to automated analog test development', *IEEE Transactions*, 1979, **CAS-26**, pp. 513–517
7 GEFFERTH, L.: 'Specification sensitivity and its use in system design', *IEE Proceedings Part G, Electronic Circuits & Systems*, 1982, **129**, (4), pp. 181–185
8 SOIN, R. S., and SPENCE, R.: 'Statistical exploration approach to design centring', *ibid.*, 1980, **127**, (6), pp. 260–269
9 BECKER, P. W.: 'Finding the better of two designs by Monte Carlo techniques', *IEEE Transactions*, 1974, **R-23**, pp. 242–246
10 BRAYTON, R. K., HACHTEL, G. D., and SANGIOVANNI-VINCENTELLI, A.: 'A survey of optimization techniques for integrated circuit design', *Proceedings IEEE*, 1981, **69**, pp. 1334–1362
11 Special issue on tolerance analysis and design, *IEE Proceedings Part G, Electronic Circuits & Systems*, 1982, **129**, (4), pp. 109–204.

Dr. R. Spence is with the Department of Electrical Engineering, Imperial College of Science & Technology, Exhibition Road, London SW7 2BT, England

APPLICATION OF LOW COST ELECTRONIC DESIGN AUTOMATION TOOLS - THE USE OF MATHEMATICAL MODELLING IN CIRCUIT DESIGN AND SIMULATION

by T. RAJA SEGARAN and T. LINGANATHAN

CONTENTS

"Application of low cost electronic design automation tools - the use of mathematical modelling in circuit design and simulation".

1.0 Introduction

This paper describes the experiences of the authors in starting up a business which involves Circuit Design, Circuit Prototyping and Testing, Schematic Capture and Printed Circuit Board or PCB design. It is hoped that the text will serve as a guide to assist participants aspiring to become entrepreneurs in these exciting fields.

R.L. Electronics Corporation is a small but innovative company that engages in projects involving circuit design, testing and fabrication, including the design of printed circuit boards. Its core business is providing Electronic Engineering Solutions and this has involved designing analogue circuits and digital circuits, including microprocessor-based circuits. Some examples of circuits include signal conditioning circuits for the Building Automation industry interfacing various types of sensors to controllers, the design of Miniature Controllers for Factory and Process Automation projects, Alarm and Enunciator circuits for Process Monitoring stations, Instrumentation circuits for testing semi-conductor devices, PC-based Data Acquisition Circuits, PC-based Control Circuits, PC-based Communications Circuits and PC-based Instrumentation Circuits. The company was formed in 1982 and has evolved from a company using traditional Electronic Design Automation tools to one that uses some amount of Design Automation. In the 10 years of its existence, R.L.Electronics has ventured into many aspects of the industry, all involving electronics circuit design, most of which uses single chip microcomputers.

The paper will discuss one of the products designed by R.L. Electronics Corporation, which is part of the Home Security and Automation system being marketed by the company. This project was chosen because it was a project that was initiated in 1983 and has undergone numerous changes, not only changes in the circuit but also in the use of the tools to design and test the circuit.

2.0 Home Security and Automation System

The Home Security and Automation System developed by the company performs all the functions of any basic alarm system and some additional functions such as switching on the living room lights, switching on the hot water heater, controlling an automatic gate, has automatic telephone dialling capability, and so on. It has user programmable features and can be customised to meet a wide range of user requirements. However, this paper will focus on the security functions of the unit, and so the attention will be focused on the alarm circuit only.

A short history of the evolution of this alarm circuitry will help illustrate some of the issues that entrepreneurs must face when designing such a product (a typical consumer product).

The first model of the alarm circuit was a simple circuit using some unijunction transistors and some triacs. The unit was a Burglar Alarm system and had two zones of operation. It had fixed Entry and Exit delays, and could drive a 10 watt siren. The unit was sold for about US$80.00, and the cost of production was considered to be US$40.00. The selling price of the unit was very attractive. The cost of production was based on the assumption that large volumes of the unit can be sold. The strategy was to market low cost alarms and to manufacture them in large volumes. The price was attractively low, and the future of the product looked very good.

The first problem encountered by the company was the fixed Entry and Exit times. To change the times, a capacitor had to be changed. Despite the low costs, customers had differing view-points as what these times should be and each unit had to be specifically tailored to meet the customers specification. To keep costs low, a fair number of units had to be manufactured at once and this tied up financial resources, leaving limited funds for marketing. Because, the volumes never materialised, only few units at a time were assembled, and so the margins were extremely small. Tailoring each unit was not practical because of the additional cost of the labour involved. Adjusting the Entry/Exit times usually involved making several trips to the customers premises after the unit was installed and changing the specific capacitor. A newer design allowing simple trimming of a potentiometer did not help resolve the problem. Infact, the margins became smaller and the venture soon became unprofitable.

It soon became clear that the market for alarms in Malaysia was not large and the concept of high volume production was not yet feasible. Around about this time, some Taiwan manufactured alarms were brought into the market by some competitors. The cost of these alarms were about half the cost of R.L. Electronics' alarms. The Taiwan units were also more professionally packaged and offered better features.

The company immediately planned to design a competitive alarm unit offering similar features. By now it was already clear that large volume production was not feasible. As a result the projected cost of the new unit was very much higher than the Taiwanese unit. The new design also had too many components, making the printed circuit board much more complex. Because of the number of components involved, a prototype had to be developed, simulated and tested. To do this the company would have to invest in test equipment such as oscilloscopes, pulse generators and multimeters. The cost of these equipment was also high at that time. It was anticipated that the entire project from start to finish would take about 8 months. Also because of the large number of components, the reliability of the product was in question and so further testing would be required.

After much thought, the company decided that it was too ambitious for such a small company to consider high volume production, especially since financial resources were limited. So a different strategy was adopted - to enter the higher alarm market and offer a "relatively" low cost microprocessor based alarm. This was more of a "niche" market catering for the wealthier customers. Such a strategy will provide the necessary business experience, and then the capital to venture into, perhaps, high volume production later. So a new alarm system was developed based on Intel's 8748 single chip microcomputer. This chip was chosen because of its relatively low cost and the large number of electronic component shops stocking the item in Malaysia. The chip also had a built-in EPROM and this provided the flexibility of altering programs. The use of a single chip microcomputer also reduced the number of components, allowed the development of better user-interface features, allowed customising the system to meet individual user demands, and allowed incorporation of additional features such as the automatic switching of lights, auto-dialing and so on. The customising feature of the alarm made the unit attractive to the customers. The EPROM

inside the 8748 microcomputer chip made the customising feature easy to undertake. The system was also designed around a standard off the shelf housing to reduce the packaging costs.

The next task was to develop the product. At that time (1985), the cost of microprocessor development tools were very expensive and unaffordable. Program development had to be done in machine language. The programs were written on a piece of paper, and the chips then programmed and tested. To avoid complications and difficulties in debugging, the program to be developed must be simple. To do this, a subset of the complete instruction set for the 8748 chip was defined and all programming had to be done using these simple instructions only. This helped create a range of software modules such as Entry/Exit timing modules, Alarm On Module and so on. These modules soon became time tested modules and the continuous use of these prevented future errors and wastage of resources.

Appendix 5 provides a data sheet for the 8748 single chip microcomputer, and the instruction subset. The entire program for the alarm system was initially developed using these few instructions only. In the early stages, most of the other instruction sets were not used. Much later on after the product had been field tested, and when additional features needed to be added on, the other instruction sets were used. By then some amount of development tools were available and these helped speed up the debugging process.

Appendix 1 provides the circuit for the alarm unit. It can be seen that the circuit was a very simple circuit and had few components only. Much of the sophisticated features of the alarm were carried by the single chip microcomputer and the programs written for the chip.

The first 8748-based alarm system was developed in about 6 weeks. Because there were fewer components, the printed circuit board designed was the single-sided type, and this reduced the artwork costs and subsequently the printed circuit board or PCB costs itself. In low volumes (10 pcs or so) the price of the PCB was about US$3.50. The new 8748 based alarm unit was priced at about US$200.00 each.

Today, the company still markets this product as the Home Security and Automation system. Additional features have been added-on to the basic alarm circuit. The company has also entered the low cost alarm market but instead of designing its own unit, is marketing an imported

unit.

Appendix 2 discusses some additional issues that a new entrepreneur must consider when entering a business venture.

3.0 Traditional Tools for Circuit Design

In the early stages, most of the circuits designed were done manually. The process involves designing the circuit on a piece of paper, building it on a protoboard or breadboard and then testing the circuit. Changes were made as and when necessary and after the final circuit was made, a hard-wired version of the circuit was built. This prototype circuit was normally field tested in an actual application situation. For example, in the case of the alarm systems, several alarm systems were hard-wired and delivered to customers premises. The customers were usually given discounts to encourage them to cooperate. Subsequent units were designed partly in PCB form and partly hard-wired. After extensive testing, the final printed circuit boards were designed and fabricated. Because of the high cost of fabricating the printed circuit boards, printed circuit board layouts had to be done right the first time. As a result, the design of the printed circuit board was a long and elaborate process. Typically, it would take about 2 to 4 months to design a sophisticated double-sided printed circuit board.

Traditionally, the most important tool in any circuit design project would be the oscilloscope. Although, multimeters may suffice in some cases, the oscilloscope allows the designer to look out for unexpected situations such as oscillations and noise.

Most small businesss today still use the traditional method in designing circuits. It is still a cost effective system and is effective for most circuit designs. The time taken to design the circuits may be a little tedious and long, but when funds are limited, there is no other alternative. A new entrepreneur would probably have to start off his first venture using this traditional method.

4.0 Modern Tools for Circuit Design

The wide availability of computers and software has helped automate some of the design processes in circuit design and development. These tools called Electronic Design Automation tools can undertake Schematic Capture, PCB Layout Design and Circuit Simulation. No longer is it necessary to build a new circuit on a breadboard and test it using oscilloscopes and function generators. Infact, an entire circuit can be designed and tested without having to touch a single component. These automation tools can even convert a schematic diagram into a printed circuit board layout automatically.

The use of design automation tools have had a great impact on reducing the development cycle of new products. There are less errors in the products developed and all aspects of the design can be modelled and studied. In circuit design, Electronic Design Automation tools have significantly reduced the lead times for circuit prototyping. It has also reduced the chances of errors, greatly improved the accuracy of documentation, and has significantly reduced the documentation time. Malfunctions in circuit design can be detected early and worst case performance characteristics can be deduced. Circuits can be tested under many different circumstance and under varying degree of conditions without having to build and rebuild the circuit. There is no danger of damaging any components during the early stages of the design. Modelling different circuits helps the designer chose a circuit design before building his prototype. It also helps determine the correct tolerances for components.

Because of limited financial resources, a new entrepreneur would have to start his business by employing the traditional methods of circuit design. Subsequently, he can start investing in the electronic design automation tools. This can be done in stages and on a critical need basis. There are certain kinds of electronic designs that are too complex that it would be impossible to do using the traditional method, or the amount of investment in specialised instruments will supercede the cost of investing in electronic design automation tools. This includes designing high speed digital circuit and also high frequency analogue circuits. In such cases, the entrepreneur may have to opt for the Design Automation tools from the beginning.

It may not be wise for a new entrepreneur to immediately invest in electronic design automation tools. He would need to be familiar with the various types of tools and the benefits they will bring to his business. Some tools may be too difficult to use that the designer will prefer to go back to the traditional method of designing.

5.0 Electronic Design Automation Tools

The development of any product that involves the use and design of electronic circuitry will involve the following stages :

1. defining the functions and specifications of the product,
2. conceptual design and flow-charting,
3. deciding on the type of components to be used,
4. creating the actual circuit,
5. proto-boarding,
6. testing and debugging,
7. printed circuit board design,
8. fabrication and assembly of printed circuit board;
9. testing and debugging the prototype system, and
10. packaging.

In deciding on the tools that would be required, the entrepreneur today has the choice of deciding between a wide range of "low cost" electronic design automation tools.

Some of the Electronic Design Automation tools that would be required in a project involving circuit design includes Schematic Capture, Printed Circuit Board or PCB layout, and Circuit Simulation. Generally, most low cost circuit simulation tools can either perform Analog Circuit simulation or Digital Circuit simulation, but not both simultaneously. But recently, there are some new software packages that can undertake mixed circuit simulations. Some simulation packages even support certain microprocessors, and can undertake simulation based on the program written for the processor. The tools that are available are numerous and they vary in terms of prices and performance capabilities between one vendor and the other. Because of the flexibility of the PC, its relatively low price, and the wide range of support available, a PC-based solution will offer the most cost effective solution. Increasingly, there are also tools being offered for the Mackingtosh computer.

Projects involving microprocessors will require additional tools such as simulators, emulators and software development tools. Popular single chip microcomputers such as the 8X51 have many third party vendors offering a full range of low cost development tools (such as high language support eg. BASIC, C-language), emulators, cross-compilers and even simulators. Some of these tools even include a programmer. Appendix 3 provides a list of vendors offering low cost PC-based

microprocessor development tools.

The word "low cost" has come to mean a lot of different things to a lot of different people. Packages ranging from a few hundred US dollars to tens of thousands of US dollars are considered low cost. To be fair, the prices of Electronic Design Automation tools used to be in the hundreds of thousands to millions of US dollars. These tools could run on mini-computers and on mainframes only. Infact, the idea of the PC based engineering workstation is a relatively new idea. Years ago, workstations were based on mini-computers. Only recently with the introduction of the more powerful 80386 and 80486 based PCs, has the gap between the PC and the minicomputer been reduced to the point that there is now no clear definite boundary between what constitutes a mini and what constitues a PC. So what is "low cost"? To be practical, "low cost" is whatever you yourself can afford. Most of the tools mentioned above are classified as low cost and are probably affordable in the west, but in the east they are still far too expensive. This will be especially so for an entrepreneur in a developing country entering the business world. It would also not be practical to expect prices to come down any lower than they already are. The entrepreneur can only hope to develop his business to a point that these tools can become affordable. Furthermore, packages that are very low in cost may not be very user friendly and so will require much time and effort to use. For example, there are circuit simulation tools that require the user to draw a circuit on a paper and translate the circuit into a series of codes and these codes are then fed into the computer. Such tools may be very difficult to use, they are succumb to errors and may not be of any use to the entrepreneur. But students studying mathematical modelling may find them beneficial. A designer on the other hand, would prefer to input his circuit in schematic form (say, by using a schematic editor) and the software package will then simulate this circuit.

A designer can on a single PC-based work station undertake the following tasks : design and draft his entire circuit using a Schematic Capture software; Simulate and Test his design using a Circuit Simulation software, and perform timing analysis of his circuit; develop a program for any microprocessor, on the PC using Assembler, or C-language or any other high level language; Simulate and Test his program using a Simulator software; cross-compile into the specific microprocessor, down-load the code and program the chip (in the case of a single chip microcomputer); design

the PCB layout; send the layout to the vendor in a disk or via a modem; and photo-plot using a Gerber or photo-plotter and send the film to the PCB manufacturer for fabrication. He can then proceed to document his work and write the project report using the same PC. He can also use a mechanical drafting software package such as AUTOCAD to design the casing or packaging and panels for the project.

The amount of funds that the entrepreneur has, will determine the kind of tools that he can afford. Being familiar with all the different tools available and the costs will help the entrepreneur in deciding the tools to purchase, but getting to know all the various tools and their capabilities is impractical. There are far too many tools having different levels of capabilities and limitations. However, there are some guidelines that an entrepreneur can follow.

In identifying electronic design automation tools, it is preferable to choose vendors who have been in the business for a long time, and offer a range of products that can be upgraded as and when the financial resources are available. The vendor can start with a lower and affordable system and then as the need arises and financial resources become available, move on to more sophisticated electronic design tools. It is also to an entrepreneurs advantage to chose tools that can be integrated to work together. For example, the schematic drawn by the Schematic Capture software, should be easily read by the Circuit Simulation software. Similarly, the PCB layout software can read the schematic without having to translate the software and make changes to the original schematic file. One of the ways to ensure this is by selecting a vendor that offers the entire range of design tools. But in most cases this may be too expensive. Furthermore, the vendors who specialise in simulation software offer better features than those who offer the integrated versions.

There are also vendors who offer a scaled down version of their design software free of charge. These tools will not accept more than 30 or so components but otherwise it is a fully functioning system. New entrepreneurs may take advantage of these tools to help them get started and to become familiarised with the use of electronic design tools.

Appendix 4 provides a list of vendors supplying electronic design automation tools but it should be noted that the list is not a comprehensive list of vendors. These are merely some of the tools that the authors are familiar with.

One of the ways an entrepreneur can optimise the use of his funds is to tailor his designs to that of the Electronic Design Automation Tools that he can afford. The entrepreneur should ensure that the project can be tailored so that its success will not depend largely on the availability of expensive Electronic Design Automation tools that are beyond his reach. One way of tailoring his designs is to make use of a limited selection of components. Although in general most entrepreneurs undertaking circuit design would like to have the freedom to chose whatever components they so desire, in practice this may not be beneficial.

There is a tendency for new entrepreneurs, especially those in the technical field to opt for designing and developing their own tools. For example, an entrepreneur might want to build his own development board based on some information available in magazines and data sheets - this generally diverts the entrepreneur from his original objectives and in the end he would have utilised all his financial resources and time in undertaking this board, but the original goal is yet to be achieved.

In the short term, investing in an electronic design tool may not shortened the development cycle. Infact the development cycle may become longer. The designer has to still go through the learning curve. There is still a lot of effort required by the designer in learning all the features, creating all the necessary libraries, and learning to model his circuit correctly. Using the traditional method, stray capacitance and residual resistance effects are already present during the breadboarding, but in circuit modelling, unless the designer includes these in his design, the design tool will completely ignore these effects. An inexperienced circuit designer who attempts to design sophisticated circuits using mathematical modelling software packages, may design unreliable circuits. In ensuring efficient use of electronic design automation tools, the designer must build a collection of commonly used circuits (based on the actual performance of these circuits), have a check list to ensure that all relevant data are entered, and to make sure that the libraries of compo-

nents are correct. For example, the designer undertaking PCB design will still need to know the thicknesses of PCB track widths for different current values. Most low cost PC-based design tools will not automatically deduce this. Track widths, non-standard component hole spacing, via sizes, spacing between tracks for different voltage levels, and so on must still be deduced by the designer. The autoplacement and autorouting feature of PCB design software will not take into account proper placement and routing in terms of noise. The software will be mainly concerned with optimum spacing and so will route the PCB accordingly. The designer must take control and ensure that sensitive circuits are properly placed, and tracks are appropriately routed. In simulation software, mathematical modelling provides idealistic performance characteristics ignoring the real situation of parasitic capacitance and inductance, not to mention noise and glitches. The designer must ensure these are taken into account, and should be aware that the software package will not automatically take these into account.

Like all other tools in life, if not used properly electronic design automation tools will create more problems and will be a waste of financial resources and time.

6.0 Other Tools

Besides electronic design tools, there are also PC-based instruments that offer a one-stop station to test prototype circuits. These instruments are used after the circuit is actually fabricated, to test and verify the actual operations of the circuit. Such instruments include PC-based oscilloscopes, multimeters, function generators and logic analyzers. These tools come in the form of a plug-in card and software. For example the PC-based oscilloscope comprises of a plug-in high speed data acquisition card and the software provided collects the data and presents in graphical form, so as to model the functions of the oscilloscope. The PC-based function generators are used to generate all sorts of signals which are then injected into the circuit to be tested. At strategic points of the circuit, the signals are measured and displayed by the PC-based oscilloscope.

Some advantages of such instruments are its lower cost and flexibility of use. These tools form part of the engineering workstation and so the workbench is not cluttered with instruments. Data collected can be stored and used later for reference or further analysis. The data can be easily ported to other statistical packages for further in-depth analysis. The data collected can also be used during the documentation stage for fast and accurate documentation.

The real time data collected can be easily compared with data collected during the simulation stage. This real time data can be processed mathematically using different modelling software to see the effects of such processing activities. For example, the data can be processed by some filter models to see the effect of such filters. The processed data can then be graphically presented on the monitor for viewing. This would be as if an actual variable filter circuit had been added on to the circuit being tested. These features will help identify potential problems of a circuit and enhance the reliability of the circuit. It will also help shorten the development cycle for prototype circuits.

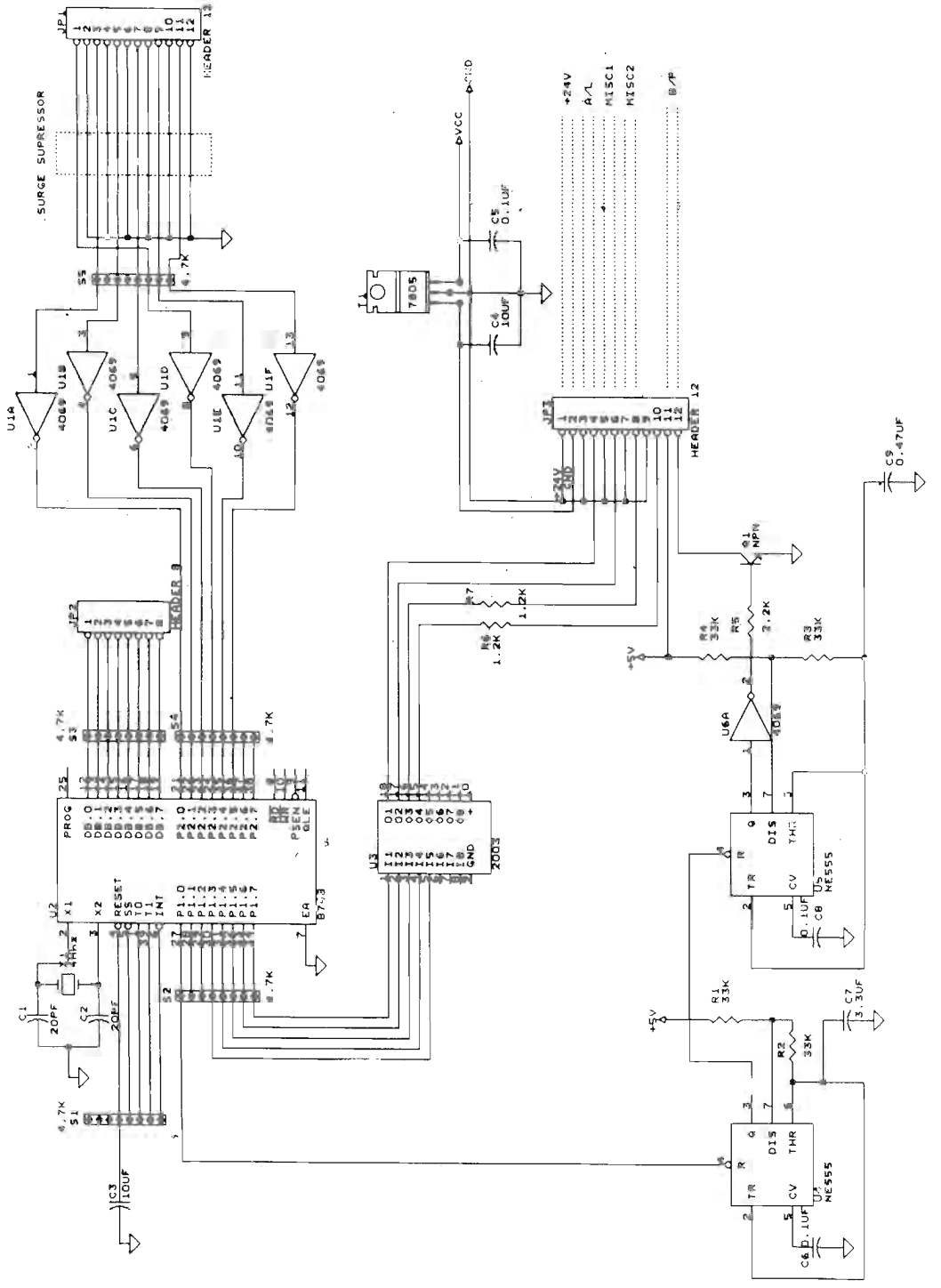
PC-based test instruments, such pulse generators, function generators, multimeters and oscilloscopes, are tools that can be incorporated onto an engineering workstation to further improve the productivity and efficiency of designing and developing circuits.

7.0 Conclusion

It cannot be argued that the use of sophisticated Electronic Design Automation tools will help improve the productivity, efficiency and the accuracy with which new products can be developed, especially products involving electronic circuits. The use of mathematical modelling not only in circuit designs, but also in the development and testing of microprocessor based systems, has significantly reduced the development time for these products. The small entrepreneur can take advantage of the availability of low cost PC-based design automation tools. Realising the limitations of the low cost tools and taking steps to guide engineering designs so as to work within these limitations will lead to optimum utilization of these tools.

APPENDIX I : CIRCUIT DIAGRAM FOR THE ALARM SYSTEM

The diagram below shows the complete schematic of the Alarm system. It centers around the 8748 single chip microcomputer, some driver chips and some buffers. The circuit itself is simple and the entire operation is centred around the program for the 8748 chip. Today the design of such a project can be done on the IBM compatible PC using some relatively low cost Electronic Design Automation tools.



APPENDIX 2 : ISSUES FACED BY A NEW ENTREPRENEUR

Before even attempting to undertake any circuit design, the entrepreneur must already have identified the marketing prospects for the product under consideration. It would be advantagous to look at similar products already available in the market, particularly in relation to the costs and the features in the products. It may be easier and cheaper to just import the product and market it. The risks are small and the effort involved may be minimal. Technical people tend to get caught up in the novelty and the challenge of designing something but the marketing aspects of the product are often neglected. Entrepreneurs must always be vary of this - ones own perception of the market alone is not enough, opinions of others must be sought and then a decision made on the viability of the project. Usually secondary data available in local trade journals and in the Government Statistics department (such as import/export figures) may help put things in the right perspective, and help with the decision making. The idea of getting entrapped into undertaking a challenging project that has absolutely no market potential is a very serious possibility for any new entrepreneur.

Having decided on the market potential, a hopeful entrepreneur must then decide on how much capital is available. In most countries, government funds may be available but generally these are never accessible to new entrepreneurs entering the industry for the first time. These funds are generally for relatively established entrepreneurs with some sort of a track record and are seeking to expand their business. The new entrepreneur will generally have to rely on his own source of funds. Keeping in mind the limited source of funds, the entrepreneur must now decide on the type tools that would be necessary to undertake the project. The cost and availability of the tools will affect the ultimate cost and completion times for the project.

The entrepreneur should keep a diary of the events and work involved, particularly the amount of labour hours involved in a particular project. This will help ascertain the cost of the item when it is ready to be marketed. The data collected will also help the entrepreneur to evaluate the worthiness of the business that he has engaged in, Simple economic principle of "opportunity" cost is useful in considering starting a business especially if the intent is purely to make money. Neglect-

ing this aspect will result in a young entrepreneur wasting his talents and investing his limited financial resources and labour in a venture that may take his entire youth, and result in a product that would have been obsoleted by the time it is completed.

The notion that an item can be developed and sold in large volumes must be dispelled from the begining. Consider the market - most items involving electronics in any developing countries (especially home security systems) will definitely face a small market by virtue of the people's poor purchasing power. Dispelling high volume production will also help avoid falling into the trap of volume pricing a new product, and then ending up selling small volumes at small margins. The idea of large volume production carries with it many other complexities that a small entrepreneur will have to consider, especially in terms of the capital required. Next forget the so-called government assistance programs and the small businessman loans that are frequently mentioned. They are not easy to obtain, and nobody will consider giving a loan to a person with just an idea in mind. Electronics is a complex thing to many bankers and if they cannot see a prototype, they will not finance the project. Generally, most loans will require collateral, and without such collateral most loans will be difficult to come by. Such assistance programmes and schemes are generally well-intended, but the realities of the business world and the risks involved prevents even the most dynamic bank from divesting its funds to an unknown entrepreneur. The new entrepreneur, must think small, and his business plan must not assume any assistance from outside sources. A more appropriate scheme for a new entrepreneur would be to build a small range of affordable (probably "niche" products) that can be easily assembled and sold on an order basis. These products must not be products that are already being manufactured on a large scale such as television sets, cassette recorders, and so on. These companies already enjoy large scale production, and by sheer economies of scale they can price their products very competitively. Go for the products that will not be required in large volumes as these products will not be attractive to the large electronic manufacturers. The security system mentioned is one such product.

Also go for products that can be expanded to meet future requirements. The basic designs can also be considered for future products. Designing circuits is very simple - any electronic or

hobby magazine will offer ideas But the key thing is to have the circuit time tested - that is, able to withstand voltage spikes, surges, variations in the domestic power supply, and noise factors. A circuit which has lasted through the years without much problem will generally not give much problems. Never under estimate the operations of a circuit - no matter how simple it may be. The most problems in circuit design have been with simple circuits. The more simpler the circuit the greater the potential for problems. This is probably due to the human-nature of being cautious when a complicated circuit is being designed but being complacent whenever a simple circuit is designed. Ensure that all the necessary safety measures and protection circuits are included, such as the reverse diode for inductive circuits, the spike clamping diode, the surge arrester and so on are included in the circuit. Since production is of low volume, the additional cents spent on these components will not significantly bite into the profits to be earned. Later if and when high volume production is considered, the high capital invested will allow some additional advanced tools to be used to further simulate and deduce optimum design and so reduce components.

The first consideration that any aspiring entrepreneur would have to face is what components to use in any circuit design. Whatever component that is stocked and is available at the local electronic outlet should preferably be used in the designs. A list of such items should be made and used as a reference whenever a new design is being considered. Stocking components is an expensive affair and this should be avoided. Secondly, using components readily available by local shops will help develop a healthy relationship between the entrepreneur and the supplier, and this may result in certain business advantages such as extended credit terms and discounts.

Before embarking on the circuit design, the packaging of the product has to be considered. Because of the low volume production, a standard off-the-shelf casing should be considered. A survey of local shops will generally help identify a suitable and attractive casing. The size and dimensions of the casing will help determine the PCB size of the circuit to be designed.

The next task is the design of the circuit itself. Once the circuit has been designed and tested, the products can be assembled and marketed. The most important item in any business is the marketing aspect. Brochures must be made, several demonstration units set-up and some advertisements must be inserted into the daily newspapers on a regular basis. Product pricing must

take into consideration the advertising and promotion costs, the transportation costs, the salesman's commissions and the networking costs. A certain percentage for future product development should also be factored into the cost of the product.

Having developed and marketed his product, the entrepreneur should realise that this only the begining.

APPENDIX 3 : Microprocessor Development Systems

There are quite a number of vendors providing powerful development tools for debugging microprocessor based systems. Some of these include in-circuit emulators, source level debuggers and performance analyzers. Most of these are PC-based systems and they provide a complete debugging environment for both the hardware and the software on the microprocessor based-systems. These tools support high language compilers such as C, PL/M, Pascal, and BASIC. The features offered by these include trace, multiple breakpoints, program performance analysis, real-time trace without stopping the processor, single-stepping, and some even offer built-in logic analyzers.

In-circuit emulators comprise of a unit with a plug in pod. The pod replaces the microprocessor in the application circuit and the entire unit then takes the role of the microprocessor. The device allows the designer to execute programs and view the status of registers, bus, inputs and outputs. Programs can be executed line by line. Some even provide timing analysis to indicate the running times of the programs. Programs can be written in Assembler or High Level Languages and compiled before running. Most units allow the user the flexibility of using third party Assemblers or Compilers.

Some examples of vendors who provide such development systems are provided below.

1. Nohau Corporation,
   51 East Campbell Avenue,
   Suite 107E,
   Campbell,
   CA 95008
   U.S.A.
   Telephone Number : 408-866-1820

2. Orion Instruments, Inc.
   180 Independence Drive,
   Menlo Park,
   California 94025
   U.S.A.
   Telephone Number : 415-327-8800
   Fax Number : 415-327-9881

3. Micro Amps Ltd,
66 Smithbrook Kilns,
Cranleigh,
Surrey,
GU6 8JJ
United Kingdom
Telephone Number : 0483-268999
Fax Number : 0483-268397

4. Applied Microsystems
3333 Bowers Avenue,
Suite 220,
Santa Clara,
CA 95054
U.S.A.
Telephone Number : 408-727-5433
FAx Number : 408-727-9011

5. Sunrise Electronics, Inc.,
524 S. Vermont Ave.,
Glendora,
CA 91740
U.S.A.
Telephone Number : 818-914-1926

6. Step Engineering
661, E. Arques Ave.,
P.O. Box 3166,
Sunnyvale,
CA 94088,
U.S.A.
Telephone Number : 408-733-7837
Fax Number : 408-773-1073

7. Huntsville Microsystems, Inc.
P.O. Box 12415
Huntsville,
AL 35815
U.S.A.
Telephone Number : 205-881-6006
Fax Number : 205-882-6701

8. Advanced Microcomputer System, Inc.,
1321 N.W. 65th Place,
Ft. Lauderdale,
Florida 33309
U.S.A.
Telephone Number : 305-975-9515
Fax Number : 305-975-9698

9. Oasys
230 Second Avenue,
Waltham,
MA 02154
U.S.A.
Telephone Number : 617-890-7889

10. MetalLink Corporation
325 E. Elliot Rd., Suite 23,
Chandler,
Arizone 85225
U.S.A.
Telephone Number : 602-926-0797
Fax Number : 602-926-1198

Some of the above companies also offer Compilers, Assmblers, Simulators, and High Level Language editors.

Vendors offering cross assemblers are listed below :

1. Enertec Inc.,
19 Jenkins Ave.,
Box 1312,
Lansdale,
PA 19446
U.S.A.
Telephone Number : 215-362-0966

APPENDIX 4 : Electronic Design Automation Tools

These include Schematic Design tools, PCB Layout Tools, Digital Circuit Simulation and Analogue Circuit Simulation Tools. Some examples of PC-based electronic design automation tools are provided

Schematic Capture Software

These are tools used to design and draft circuits on the PC. They enable the designer to design electronic circuits faster and more accurately, using the sophisticated graphical features of the PC. Other features include a library of components, different levels of hierarchy, pop-up menus for easy user interface, automatic check for electrical rules, cross reference parts, bill of materials, plotting and printing capabilities, and so on. A list of vendors providing such solutions has been provided below :

1. PROTEL SCHEMATIC
   Technopark,
   Dowsings Point,
   Hobart,
   Australia
   G.P.O. Box 204,
   Hobart 7001
   Telephone Number : 002-73-0100
   Fax Number : 002-73-0944

2. ORCAD
   3175, N.W. Aloclek Drive,
   Hillsboro,
   Oregon U.S.A. 97124
   U.S.A.
   Telephone Number : 503-690-9881
   Fax Number : 503-690-9891

3. Tango Schematic Capture
   Accel Technologies, Inc.
   6825 Flanders Drive,
   San Diego,
   CA 92121
   U.S.A.
   Telephone Number : 619-554-1000
   Fax Number : 619-554-1019

4. Easy-PC (Schematic and PCB Layout)
   Number One Systems Limited,
   Harding Way,
   St. Ives,
   Huntingdon,
   Cambs,
   England,
   PE17 4WR
   Telephone Number : 0480-61778

5. P-CAD
   Cadam EDA Headquarters,
   1290 Parkmoor Avenue,
   San Jose,
   CA 95126,
   U.S.A.
   Telephone Number : 408-971-1300
   Fax Number : 408-279-3752

Circuit Simulation Software

Usually vendors provide either digital or analogue circuit simulation software only. Some of these tools will allow the schematics captured by the software above to be simulated. Most of these packages also incorporate a schematic editor. Features include check for glitches, predicting circuit board performance for propagation delays and timing measurements, worst case analysis, Monte Carlo Analysis and so on. Some vendors who provide such tools are listed below.

1. ORCAD (Digital Simulation only)
   3175, N.W. Aloclek Drive,
   Hillsboro,
   Oregon U.S.A. 97124
   U.S.A.
   Telephone Number : 503-690-9881
   Fax Number : 503-690-9891

2. Intusoft (Analogue Simulation)
   P.O. Box 710,
   San Pedro,
   CA 90733-0710
   U.S.A.
   Telephone Number : 310-833-0710
   Fax Number : 310-833-9658

3. MicroSim Corporation (Analogue and Mixed Signal)
20, Fairbanks,
Irvine,
CA 92718
U.S.A.
Telephone Number : 714-770-3022

4. P-CAD
Cadam EDA Headquarters,
1290 Parkmoor Avenue,
San Jose,
CA 95126,
U.S.A.
Telephone Number : 408-971-1300
Fax Number : 408-279-3752

SUSIE - The real time digital simulation and design verification tool.

Introduction

SUSIE is a universal electronic breadboard that simulates any schematic logic design on personal computers and workstations.

SUSIE includes logic simulators and test vector generators thereby making it a complete design verification environment.

It eliminates breadbording and the need for Test Equipment, thereby making it a highly efficient and versatile. Test cost and design time is greatly reduced.

Features

Three important features that make a simulator a practical design verification tool :

1. Interactive operation; Recompilation wastes valuable time.

2. Easy to use test vector generation; especially when it comes to entering thousands of design stimulators (test vectors).

3. End to end solution; A complete set of current libraries must be available including PLD's and processors.

Susie has all of the above and more.

* Real time interactive simulation.
  Susie is interactive and without visible compilations. You can toggle switches, move jumpers, load and modify ROM, RAM, single step MPU programs, overide output signals.

* Built in logic analyzer.

* Built in programmable test vector generator.
  A graphics oriented test vector editor.

* Accurate to 10 pico seconds.

* Automatic error detection of timing violations and bus conflicts.

* Equivalent capacity from 10,000 gates and above.

* Simulation speeds in excess of 2,000,000 gates per second.

* Component libraries for TTL, CMOS, ECL, GaAs, PLD, FPGA, ASIC, etc.

* IC modelling of custom devices.

* Supports JEDEC and hex files.

* Supports all popular schematic capture software.

* SUSIE has three modes of verifiaction:

  a) timing driven functional simulator.

  b) unit propagation based glitch detector.

  c) High precision timing simulator.

THE INTUSOFT (ANALOGUE) SIMULATION SYSTEM

ICAPS is a complete simulation system and includes 4 program modules each performing a different function.

1) Spice Net - Schematic Entry

2) Pre Spice - Model Libraries

3) Is Spice - SPICE Simulation Libraries

4) Intu Scope - IsSpice post processing module

Together these modules provide an integrated simulation system that will allow the user to :

1) Enter a schematic into a computer.

2) Access SPICE models form an extensive library of over 1200 parts.

3) Simulate circuits in all disciplines : power, rf, mixed signal, digital, filters, etc.

4) Perform high speed circuit simulation including Bode plots, frequency response, operating points, Fourier analysis, etc.

5) Construct graphs of the outputvoltage and current waveforms.

6) Produce report quality printouts of the results.

SpiceNet : Schematic Entry

SpiceNet is a schematic entry program. It is designed to be a direct interface to ISpice , the analog and mixed signal simulator. It greatly eases the burden of of creating a spice netlist, redy for simulation, directly from the schematic.

Unlike other schematicpackages, which are geared for digital circuitry or PCB layout, SpiceNet supports all facets of the Spice netlist including node numbering, component values, spice control statements and models.

SpiceNet part placement is straightforward, designed to be faster than pencil and paper. Most components can be placed on the schematic with a single keystroke. For

documentation and circuit debugging, waveforms and node voltages can be displayed directly on the screen. SpiceNet has a direct interface to IsSpice allowing simulation to be performed directly from the schematic.

IsSpice : Analog Circuit Simulation

IsSpice performs several different kinds of analyses: DC includind operating point, transfer function, sensitivity, curve families, input/output impedances, AC including Bode plots , noise and distortion analysis, and non linear Transient including Fourier analysis.

IsSpice is completely integrated with SpiceNet, PreSpice and IntuScope allowing the simulation process to appear virtually transparent to the end user.

IsSpice will accept input from any schematic entry program that produces a Berkeley SPICE compatible netlist.The output from IsSpice is an ASCII text file (same syntax as Berkeley Spice 2G.6) that can be viewed with any text editor.

PreSpice : Model Libraries/Advanced Analyses

PreSpice addds an extensive array of over 1200 SPICE device models.

The SPICE Device Libraries in PreSpice contain a wide variety of models including diodes, zeners, bjt's op-amps, comparators, transformers, non-linear magnetics, SCR's, power MOsfets, PWM's, SC filters, analog behavioral models, digital logic gates, switches, opto-isolators, transmission line models, crystals, etc.

Models are stored in ASCII Text files that can be viewed and edited.

IntuScope : Graphical Waveform Processing

IntuScope is an interactive graphical data processing programe especially designed to diplay , analyse and manipulate IsSpice output data. IntuScope can display waveforms from any Berkeley SPICE compatible program, as well as user entered data.

Input format : IntuScope accepts data files that conforms to the format of Berkeley SPICE 2G.6 .

Graph Format : Ability to display and compare data from several files and several analy-

sis types on the same graph. Scaling formats include : Semilog, Histogram, Cumulative Probability.

Waveform operations :

RMS, Peak to peak, Mean, maximum, minimum, Square-root, Standarad deviation, cursor location.

Add, Subtract, Multiply, Divide using numbers, numbers and waveforms and two waveforms.

Trigonometric functions, log, power, e(x), polar to rectangular conversion, group delay, gain/phase margin, propagation delay, rise/fall time and convolution.

PCB Layout Software

These tools have features such as autoplacement of components, automatic routing, multi-layer, route optimization to minimize vias and track length, support for photo-plotters, printing and plotting capability, extensive library of components, design rule checking and bill of materials. Some examples of vendors providing such solutions are as follows :

1. PROTEL SCHEMATIC
   Technopark,
   Dowsings Point,
   Hobart,
   Australia
   G.P.O. Box 204,
   Hobart 7001
   Telephone Number : 002-73-0100
   Fax Number : 002-73-0944

2. ORCAD
   3175, N.W. Aloclek Drive,
   Hillsboro,
   Oregon U.S.A. 97124
   U.S.A.
   Telephone Number : 503-690-9881
   Fax Number : 503-690-9891

3. Tango PCB
   Accel Technologies, Inc.
   6825 Flanders Drive,
   San Diego,
   CA 92121
   U.S.A.
   Telephone Number : 619-554-1000
   Fax Number : 619-554-1019

4. P-CAD
   Cadam EDA Headquarters,
   1290 Parkmoor Avenue,
   San Jose,
   CA 95126,
   U.S.A.
   Telephone Number : 408-971-1300
   Fax Number : 408-279-3752

NOTE : THE ABOVE NAMES OF VENDORS IS FOR REFERENCES ONLY THERE ARE MANY

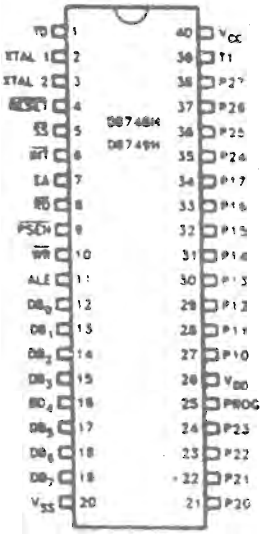


Figure 3. Pin Configuration

Table 1. Pin Description (40-Pin DIP)

Symbol	Pin No.	Function
V_{SS}	20	Circuit GND potential.
V_{DD}	26	+ 5V during normal operation.
		Programming power supply (+ 21V).
V_{CC}	40	Main power supply; + 5V during operation and programming.
PROG	25	Output strobe for 8243 I/O expander.
		Program pulse (+ 18V) input pin during programming.
P10–P17 Port 1	27–34	8-bit quasi-bidirectional port.
P20–P23 P24–P27 Port 2	21–24 35–38	8-bit quasi-bidirectional port. P20–P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for 8243.
DB0–DB7 BUS	12–19	True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.
T0	1	Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CKL instruction.
		Used during programming.
T1	39	Input pin testable using the JT1, and JNT1 instructions. Can be designated the timer/counter input using the STRT CNT instruction.
$\overline{INT}$	6	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active low) interrupt must remain low for at least 3 machine cycles for proper operation.
$\overline{RD}$	8	Output strobe activated during a BUS read. Can be used to enable data onto the bus from an external device. Used as a read strobe to external data memory. (Active low)

Table 1. Pin Description (40-Pin DIP) (Continued)

Symbol	Pin No.	Function
$\overline{RESET}$	4	Input which is used to initialize the processor. (Active low) (Non TTL V_{IH})
		Used during programming.
$\overline{WR}$	10	Output strobe during a bus write. (Active low) Used as write strobe to external data memory.
ALE	11	Address latch enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.
$\overline{PSEN}$	9	Program store enable. This output occurs only during a fetch to external program memory. (Active low.)
$\overline{SS}$	5	Single step input can be used in conjunction with ALE to "single step" the processor through each instruction.
EA	7	External access input which forces all program memory fetches to reference external memory. Useful for emulation and debug. (Active high.)
		Used during (18V) programming.
XTAL1	2	One side of crystal input for internal oscillator. Also input for external source. (Non TTL V_{IH}.)
XTAL2	3	Other side of crystal input.

Table 2. Instruction Set

Mnemonic	Description	Bytes	Cycles
ACCUMULATOR			
ADD A, R	Add register to A	1	1
ADD A, @R	Add data memory to A	1	1
ADD A, #data	Add immediate to A	2	2
ADDC A, R	Add register with carry	1	1
ADDC A, @R	Add data memory with carry	1	1
ADDC A, #data	Add immediate with carry	2	2
ANL A, R	And register to A	1	1
ANL A, @R	And data memory to A	1	1
ANL A, #data	And immediate to A	2	2
ORL A, R	Or register to A	1	1
ORL A, @R	Or data memory to A	1	1
ORL A, #data	Or immediate to A	2	2
XRL A, R	Exclusive or register to A	1	1
XRL A, @R	Exclusive or data memory to A	1	1
XRL A, #data	Exclusive or immediate to A	2	2

Mnemonic	Description	Bytes	Cycles
ACCUMULATOR (Continued)			
INC A	Increment A	1	1
DEC A	Decrement A	1	1
CLR A	Clear A	1	1
CPL A	Complement A	1	1
DA A	Decimal adjust A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
INPUT/OUTPUT			
IN A, P	Input port to A	1	2
OUTL P, A	Output A to port	1	2
ANL P, #data	And immediate to port	2	2
ORL P, #data	Or immediate to port	2	2
INS A, BUS	Input BUS to A	1	2
OUTL BUS, A	Output A to BUS	1	2
ANL BUS, #data	And immediate to BUS	2	2
ORL BUS, #data	Or immediate to BUS	2	2
MOVD A, P	Input expander port to A	1	2

Table 2. Instruction Set (Continued)

Mnemonic	Description	Bytes	Cycles
INPUT/OUTPUT (Continued)			
MOVD P, A	Output A to expander port	1	2
ANLD P, A	And A to expander port	1	2
ORLD P, A	Or A to expander port	1	2
REGISTERS			
INC R	Increment register	1	1
INC @R	Increment data memory	1	1
DEC R	Decrement register	1	1
BRANCH			
JMP addr	Jump unconditional	2	2
JMPP @A	Jump indirect	1	2
DJNZ R, addr	Decrement register and skip	2	2
JC addr	Jump on carry = 1	2	2
JNC addr	Jump on carry = 0	2	2
JZ addr	Jump on A zero	2	2
JNZ addr	Jump on A not zero	2	2
JT0 addr	Jump on T0 = 1	2	2
JNT0 addr	Jump on T0 = 0	2	2
JT1 addr	Jump on T1 = 1	2	2
JNT1 addr	Jump on T1 = 0	2	2
JF0 addr	Jump on F0 = 1	2	2
JF1 addr	Jump on F1 = 1	2	2
JTF addr	Jump on timer flag	2	2
JNI addr	Jump on INT = 0	2	2
JBb addr	Jump on accumulator bit	2	2
SUBROUTINE			
CALL addr	Jump to subroutine	2	2
RET	Return	1	2
RETR	Return and restore status	1	2
FLAGS			
CLR C	Clear carry	1	1
CPL C	Complement carry	1	1
CLR F0	Clear flag 0	1	1
CPL F0	Complement flag 0	1	1
CLR F1	Clear flag 1	1	1
CPL F1	Complement flag 1	1	1
DATA MOVES			
MOV A, R	Move register to A	1	1
MOV A, @R	Move data memory to A	1	1
MOV A, # data	Move immediate to A	2	2

Mnemonic	Description	Bytes	Cycles
DATA MOVES (Continued)			
MOV R, A	Move A to register	1	1
MOV @R, A	Move A to data memory	1	1
MOV R, # data	Move immediate to register	2	2
MOV @R, # data	Move immediate to data memory	2	2
MOV A, PSW	Move PSW to A	1	1
MOV PSW, A	Move A to PSW	1	1
XCH A, R	Exchange A and register	1	1
XCH A, @R	Exchange A and data memory	1	1
XCHD A, @R	Exchange nibble of A and register	1	1
MOVX A, @R	Move external data memory to A	1	2
MOVX @R, A	Move A to external data memory	1	2
MOVP A, @A	Move to A from current page	1	2
MOVP3 A, @A	Move to A from page 3	1	2
TIMER/COUNTER			
MOV A, T	Read timer/counter	1	1
MOV T, A	Load timer/counter	1	1
STRT T	Start timer	1	1
STRT CNT	Start counter	1	1
STOP TCNT	Stop timer/counter	1	1
EN TCNTI	Enable timer/counter interrupt	1	1
DIS TCNTI	Disable timer/counter interrupt	1	1
CONTROL			
EN I	Enable external interrupt	1	1
DIS I	Disable external interrupt	1	1
SEL RB0	Select register bank 0	1	1
SEL RB1	Select register bank 1	1	1
SEL MB0	Select memory bank 0	1	1
SEL MB1	Select memory bank 1	1	1
ENT0 CLK	Enable clock output on T0	1	1
NOP	No operation	1	1

8748 INSTRUCTION SUBSET

1	NOP	The No Operation Instruction	00H
2	OUTL Pp,A	Output Accumulator Data to Port 1 Output Accumulator Data to Port 2	39H 3AH
3	RET	Return Without PSW Restore	83H
4	MOV A,#data	Move Immediate Data to Accumulator	23H
5	JZ address	Jump If Accumulator Is Zero	C6H
6	JNZ address	Jump If Accumulator Is Not Zero	96H
7	INC A	Increment Accumulator	17H
8	IN A,Pp	Input Port 1 or Data to Accumulator Input Port 2 or Data to Accumulator	09H 0AH
9	DEC A	Decrement Accumulator	07H
10	CLR A	Clear Accumulator	27H
11	CALL address	Subroutine Call	14H
12	ADD A,#data	Add Immediate Data to Accumulator	03H
13	JMP address	Jump to Location	04H

REFERENCES

Intel 8-Bit Embedded Controllers Data Book 1990
Radio Electronics magazine
Circuit Cellar magazine
EDN magazine
Protel Schematic manual
Protel AUTOTRAX manual
SUSIE demonstration disk and literature
INTUSOFT demonstration disk and literature
MicroSim literature
ORCAD demonstration and literature
P-CAD demonstration disk and literature

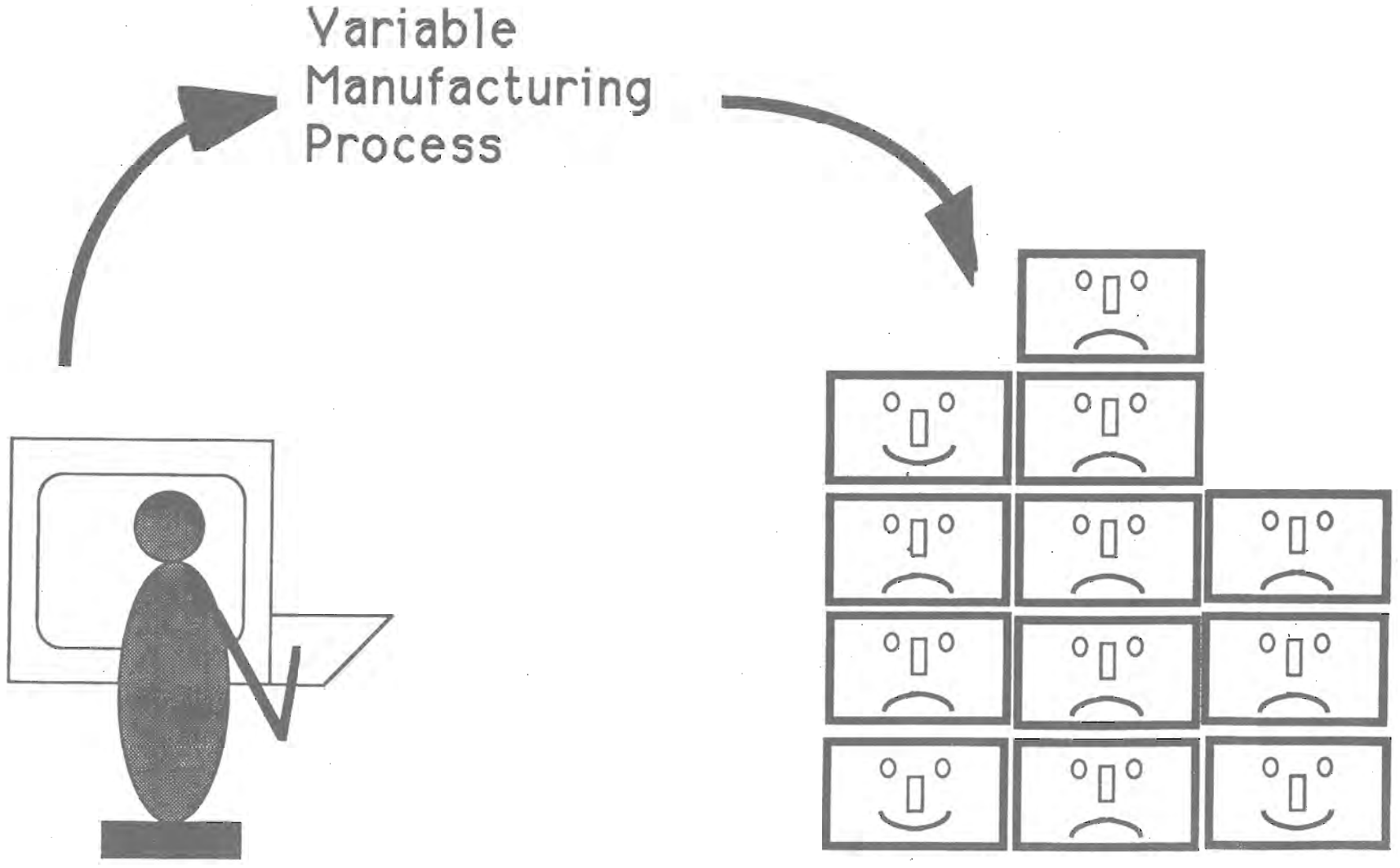
Workshop on Mathematical Modelling for Circuit Design

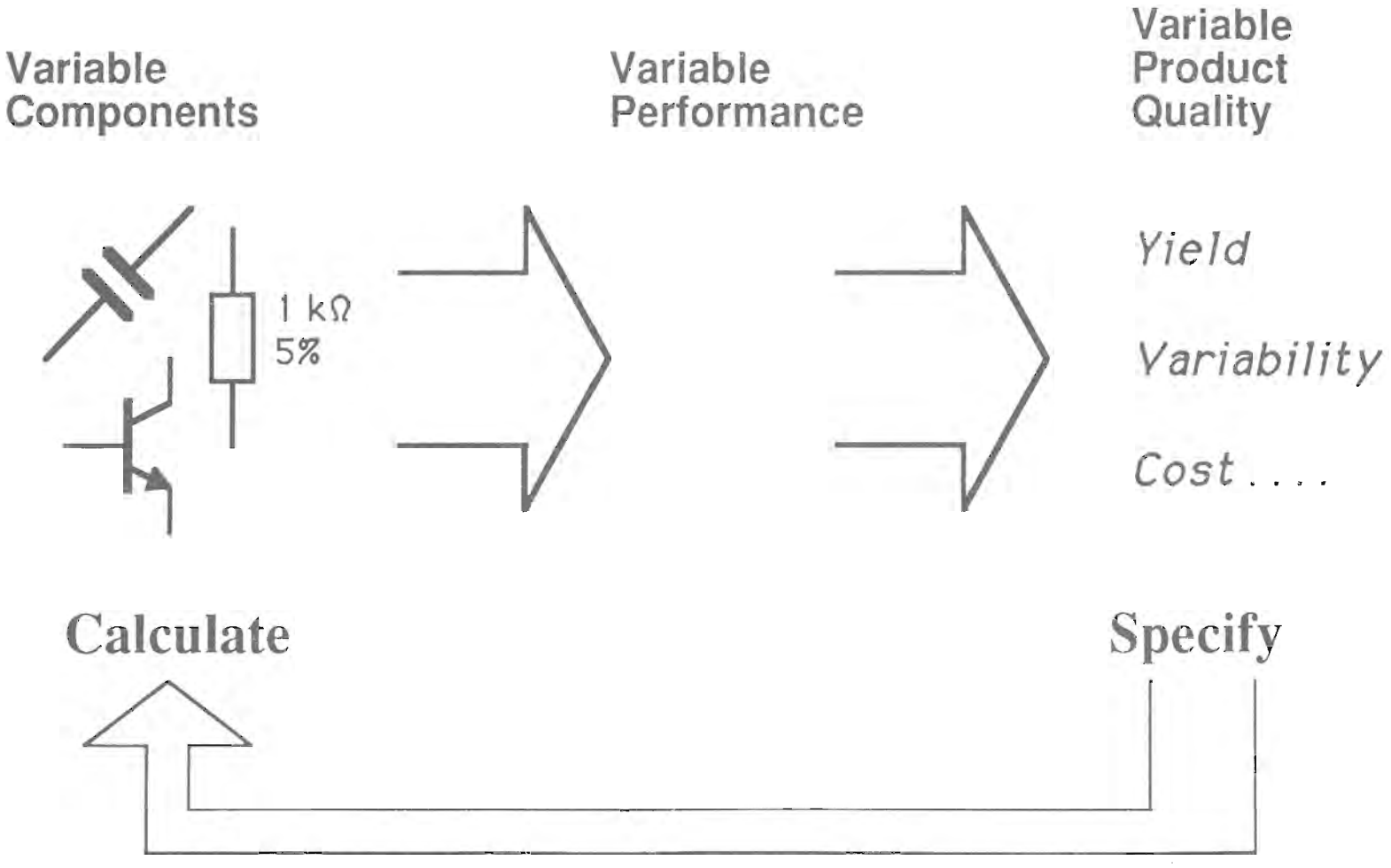
Sri Lanka, 1992

Tolerance Design of Electronic Circuits

Robert Spence

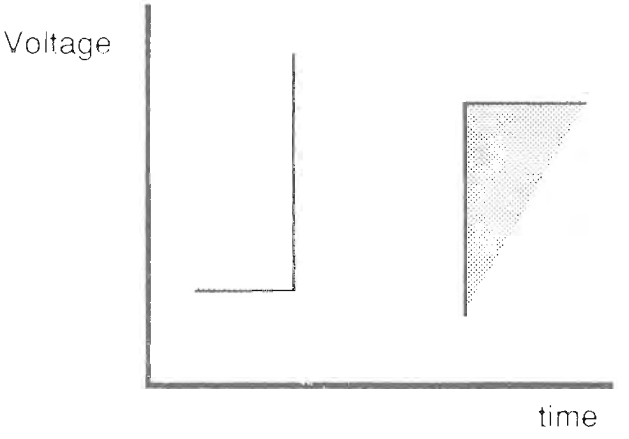
Imperial College
London


Variable
Manufacturing
Process

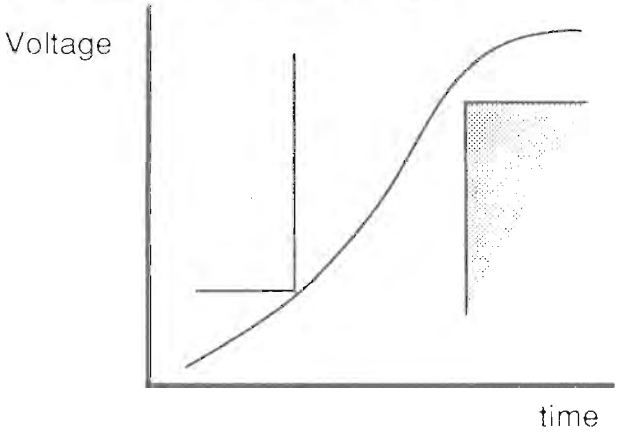

Variable
Components
Variable
Performance
Variable
Product
Quality
1 kΩ
5%
Yield
Variability
Cost . . . .
Calculate
Specify

Tolerance Effects

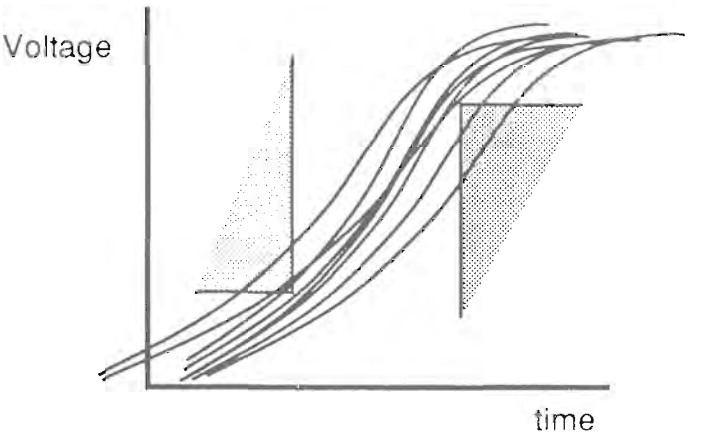
Specification on Inverter response



Response of nominal design



Response of 1000 manufactured inverters
(few samples shown)



The Designer's Questions

1

Is the yield particularly sensitive to the nominal value of any component ? If it is, which component(s) is it sensitive to ?

2

What are the optimum nominal values of the components which will lead to maximum manufacturing yield, and what is the value of the yield ?

3

Is there a trade-off between tolerance and yield ? Wider tolerance components cost less but lead to lower yield, which costs more - where is the optimum ?

The Designer's Questions

4

How sensitive is the yield to the various performance specifications ? What is the relative difficulty of meeting different specifications ? What specification changes should I negotiate ?

5

Is it possible to redesign my circuit economically so that the performance variability over a satisfactory fraction of the manufactured samples is reduced, making the circuit more saleable ?

The Designer's Questions

6

Of all the components in my circuit, which, if any, would it be economic to measure before connection into the manufactured circuit, with a view to maximising the manufacturing yield ?

Yield Increase

Adjustment of nominal parameter values

Example

Siemens integrated filter

predicted increase in yield 11%

measured increase in yield 11%

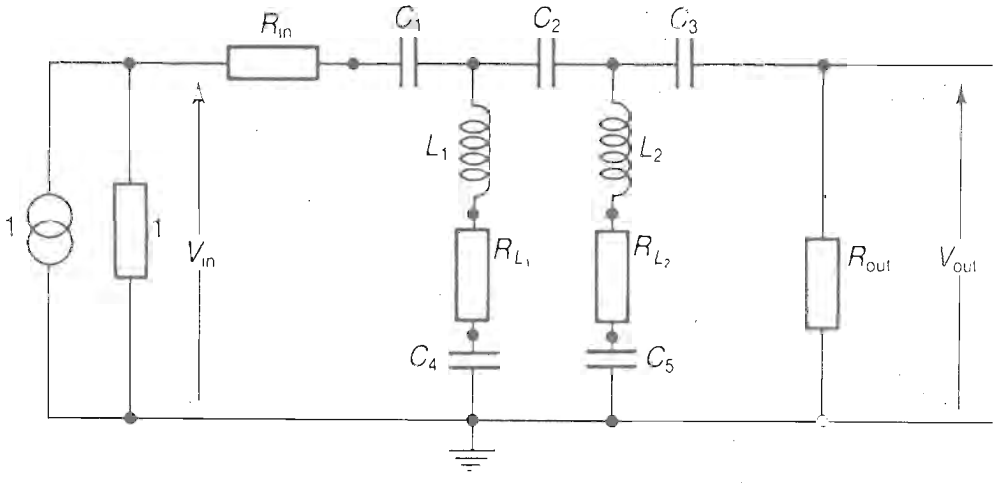
Design Centering

Cost Reduction

Adjustment of component tolerances

Example

Bandpass filter



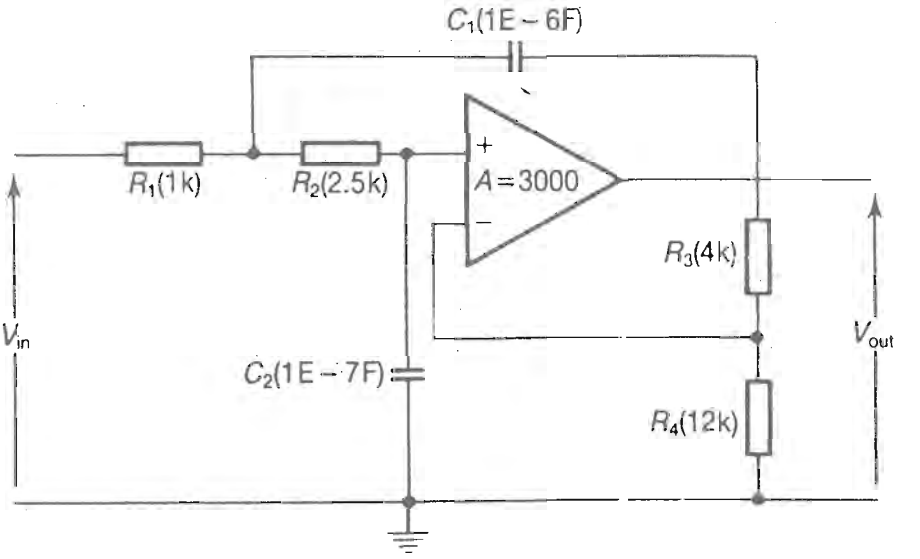
Manufacturing cost reduced from

4.3 units to 3.5 units

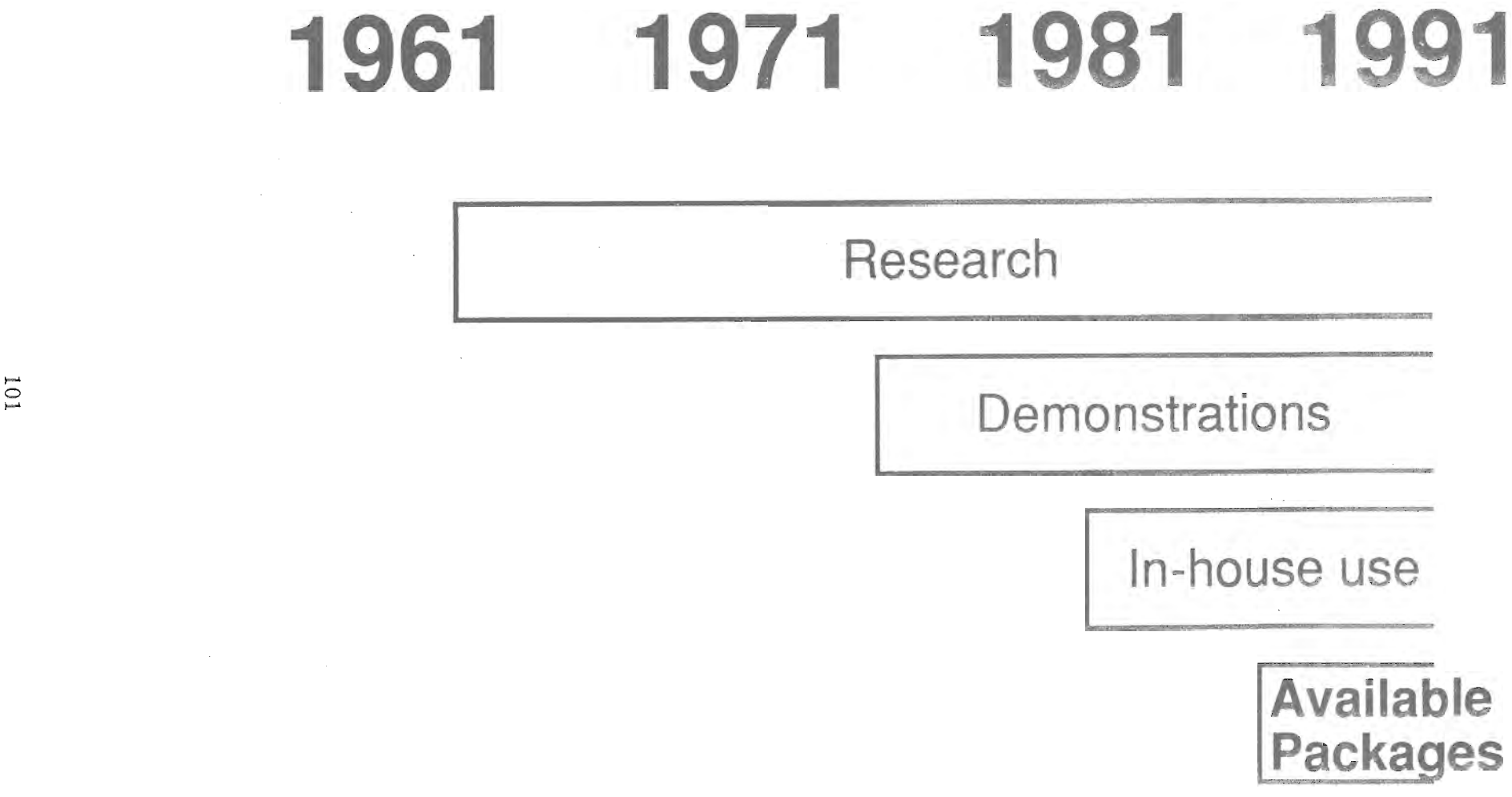
Variability Reduction

Adjustment of nominal parameter values

Example



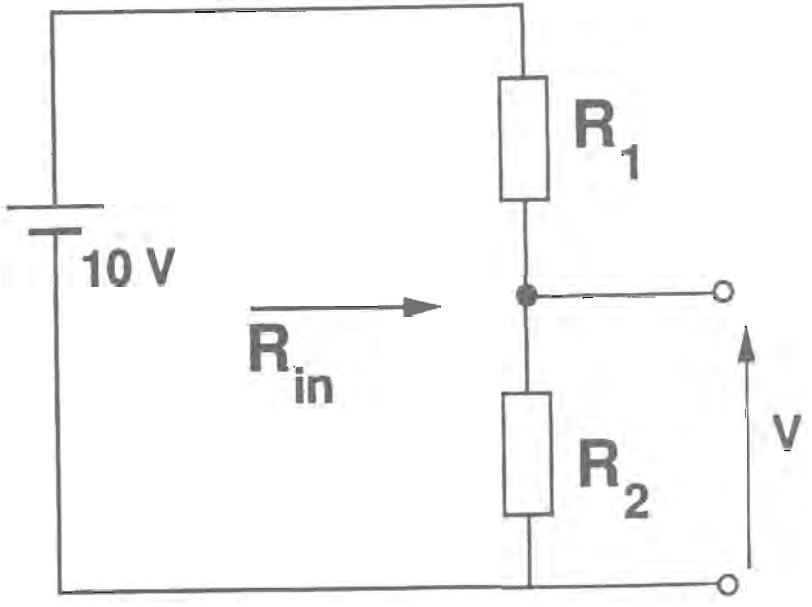
Variability of selectivity reduced from 8.6 to 0.1

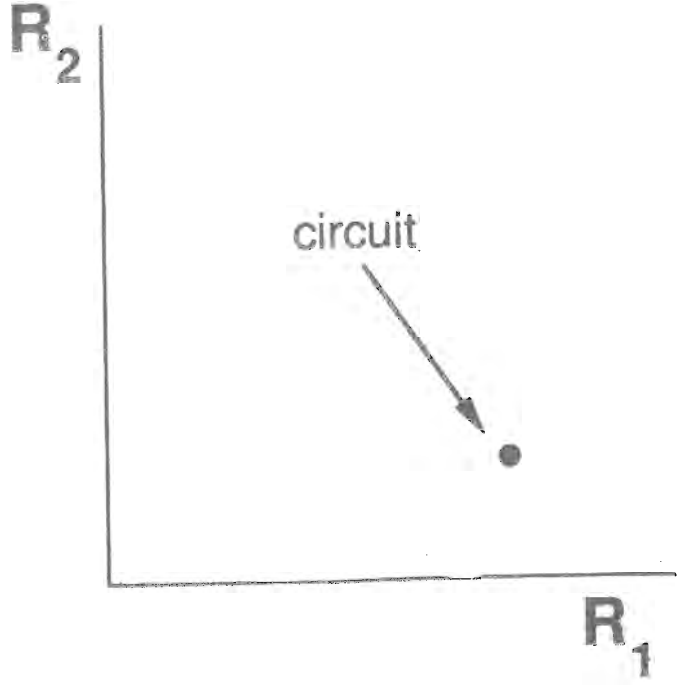

1961
1971
1981
1991
Research
Demonstrations
In-house use
Available Packages

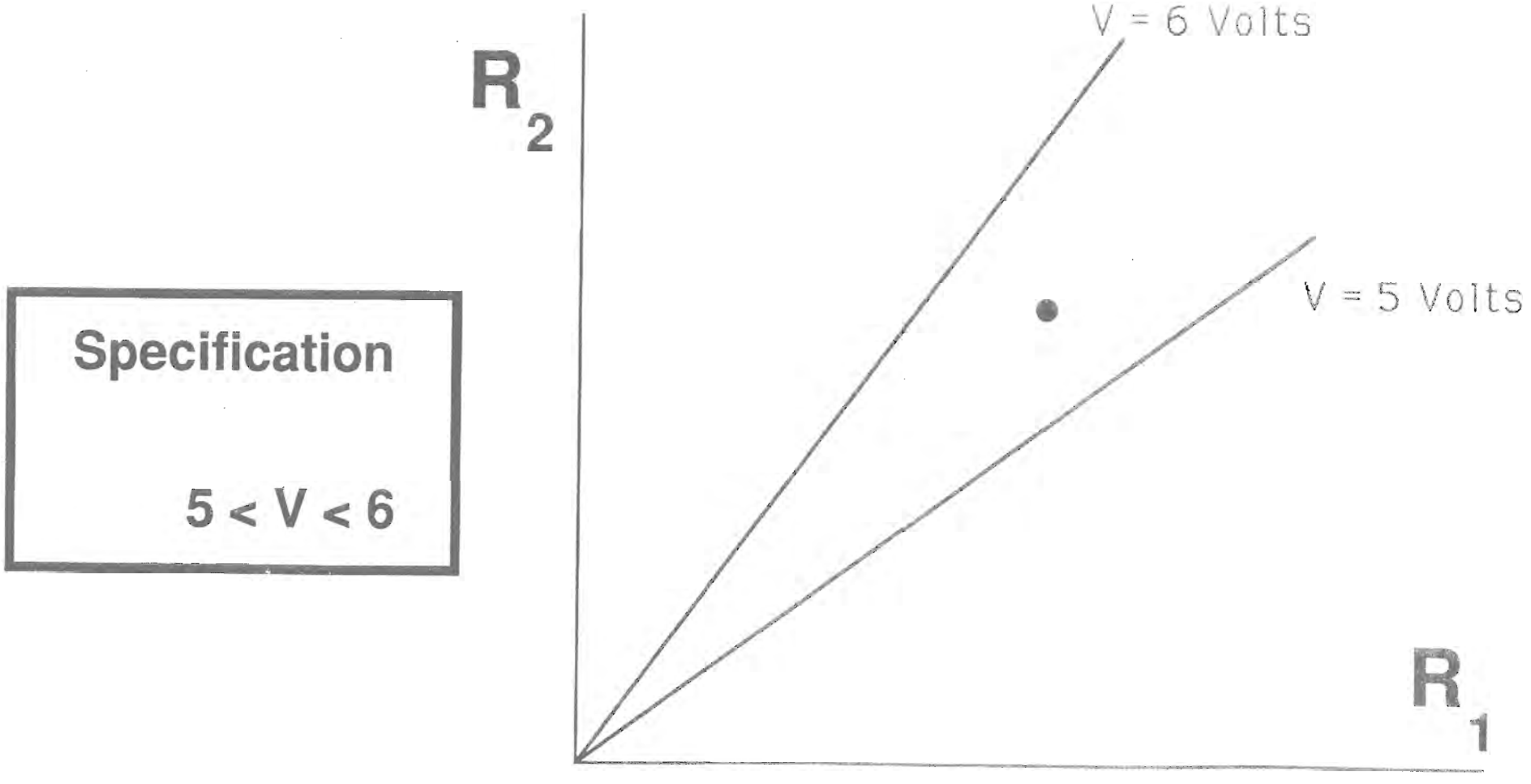
Today

- Algorithms proven via realistic examples
- Good take-up by industry
- Software available off-the-shelf
- Leading to reliability design

Mature field, ready for exploitation & development


10 V
R1
R2
Rin
V

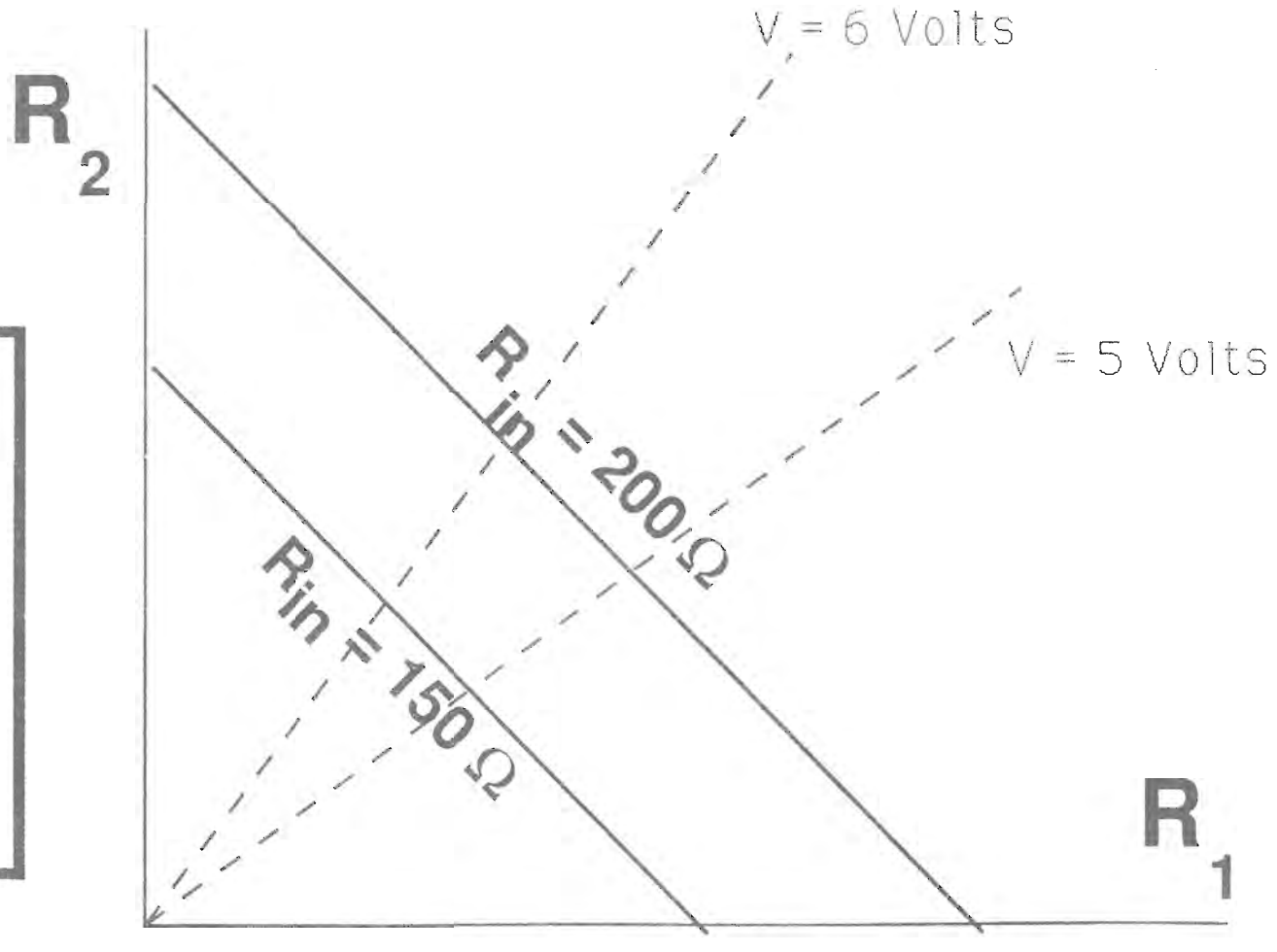

R2
circuit
R1


Specification
5 < V < 6
R2
V = 6 Volts
V = 5 Volts
R1

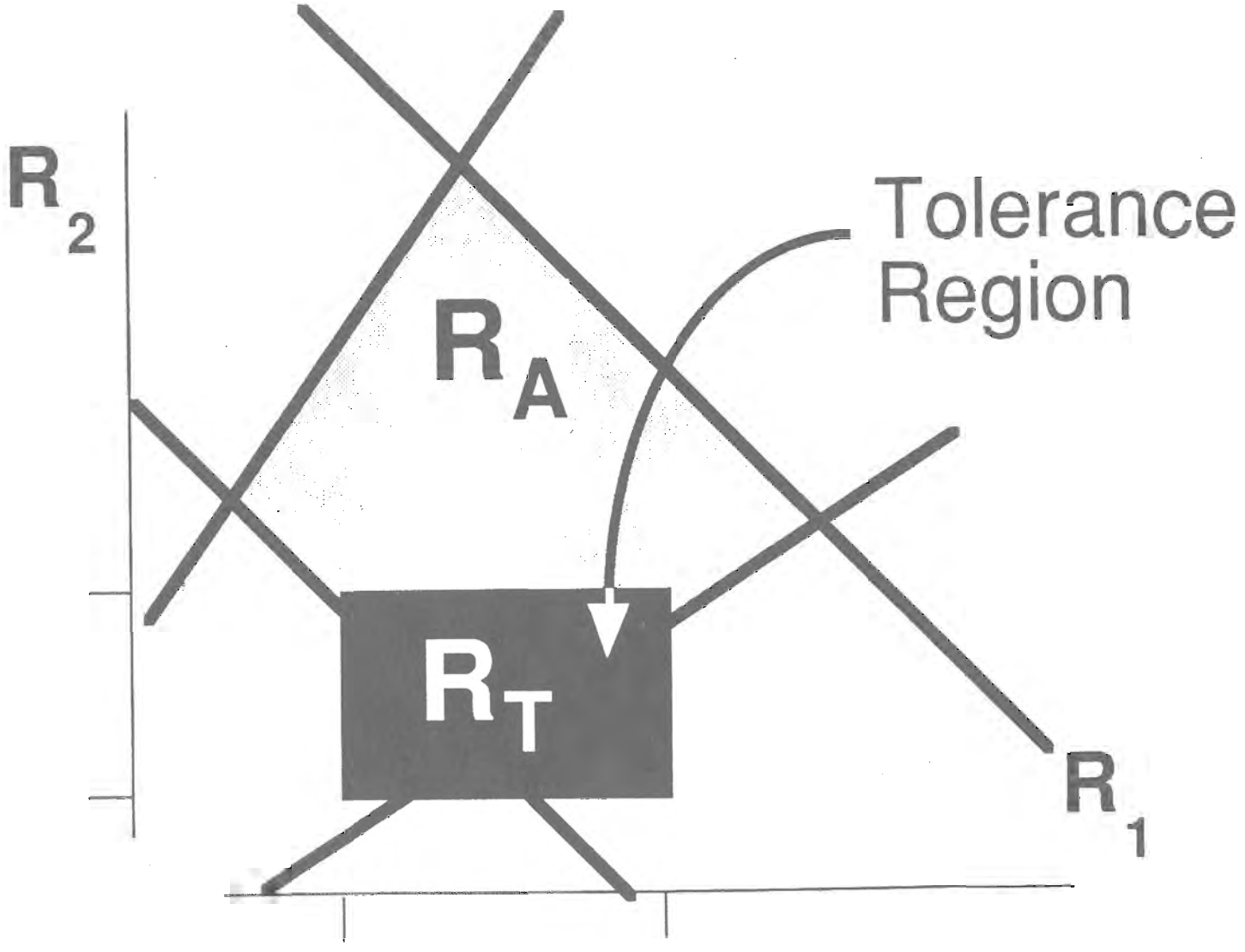
Specifications

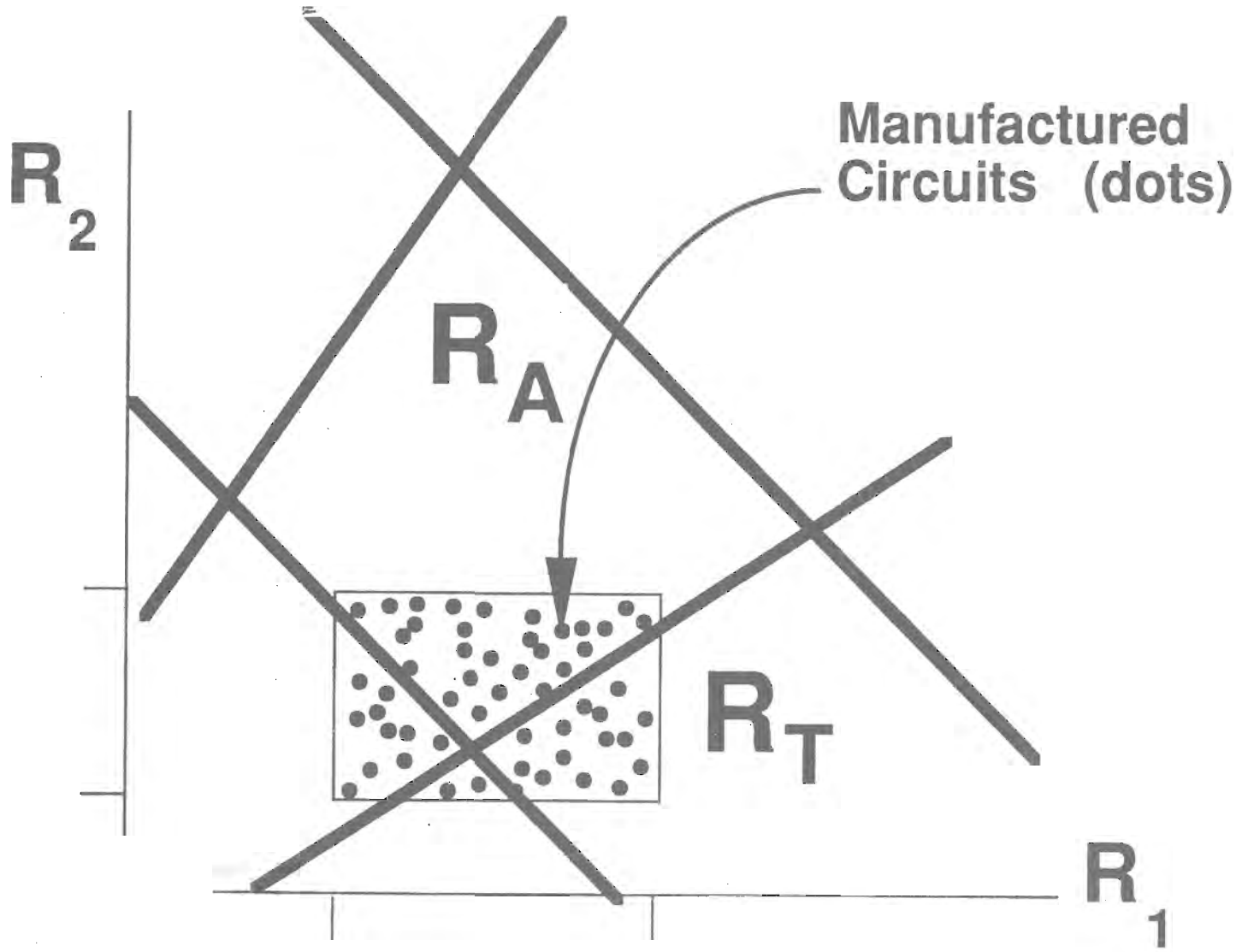
$5 < V < 6$

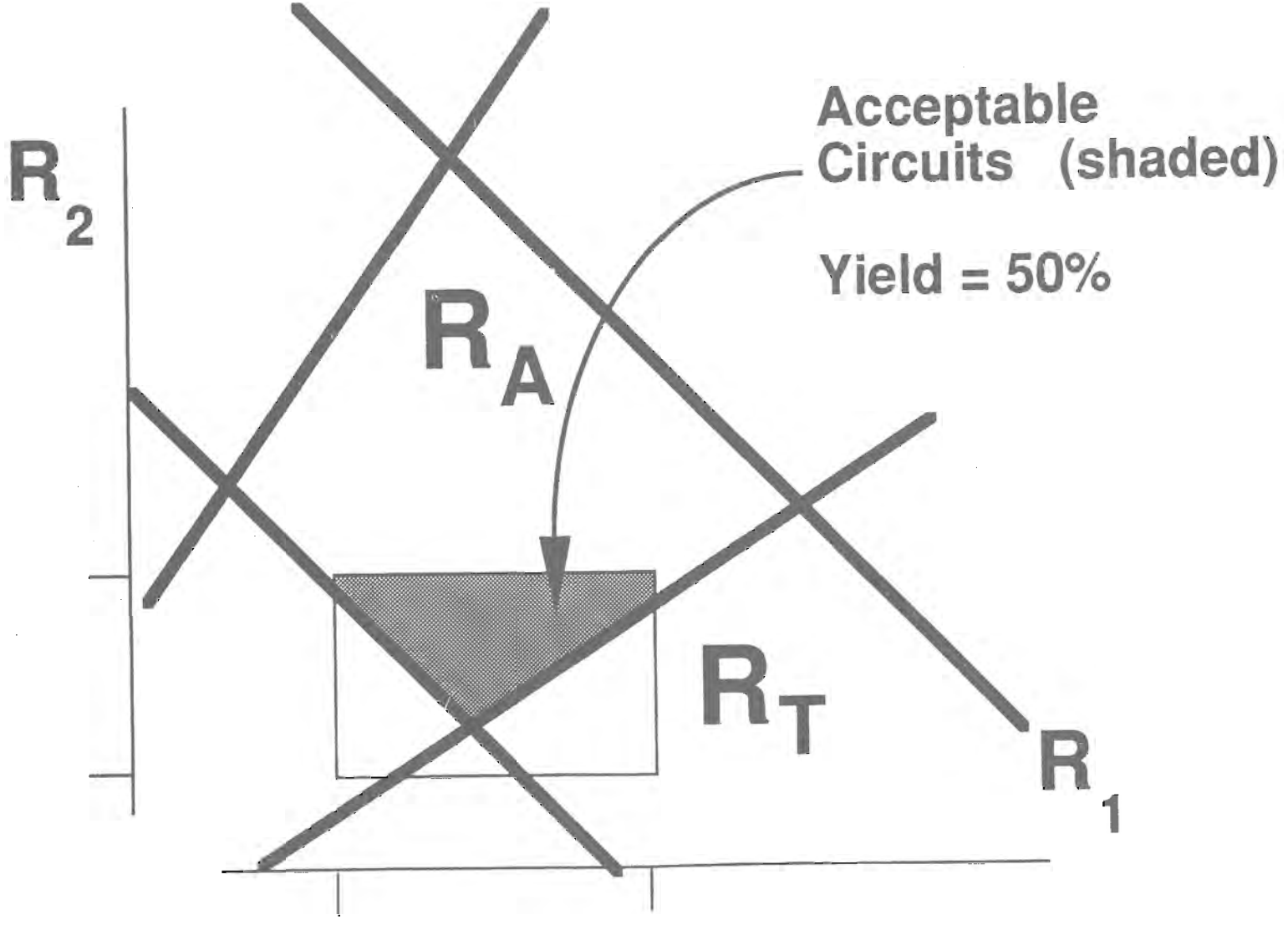
$150 < R_{in} < 200$

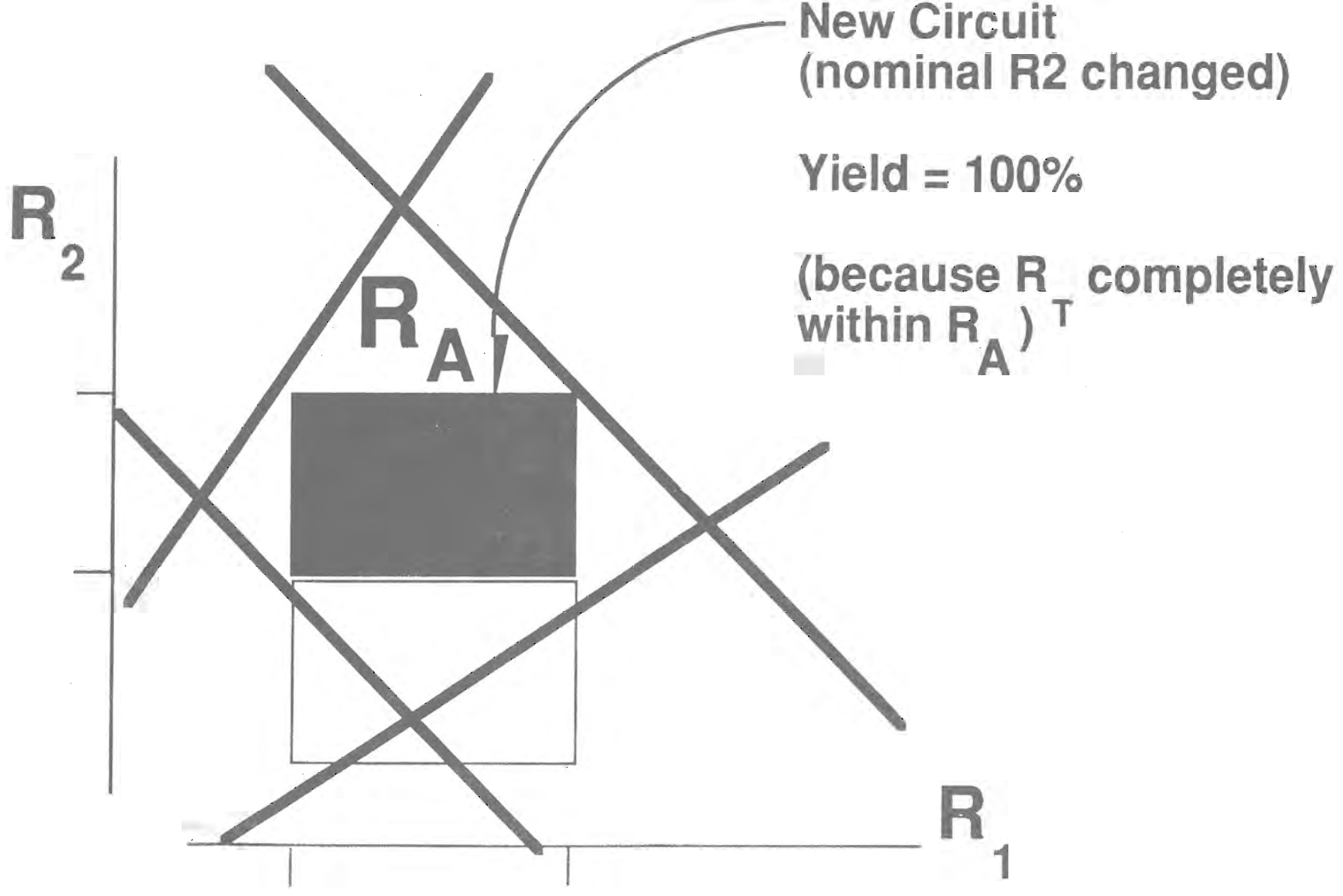


Region of Acceptability
R 1
R in = 200 Ω
V = 5 Volts
R A
V = 6 Volts
Rin = 150 Ω
R 2


R2
RA
Tolerance
Region
RT
R1

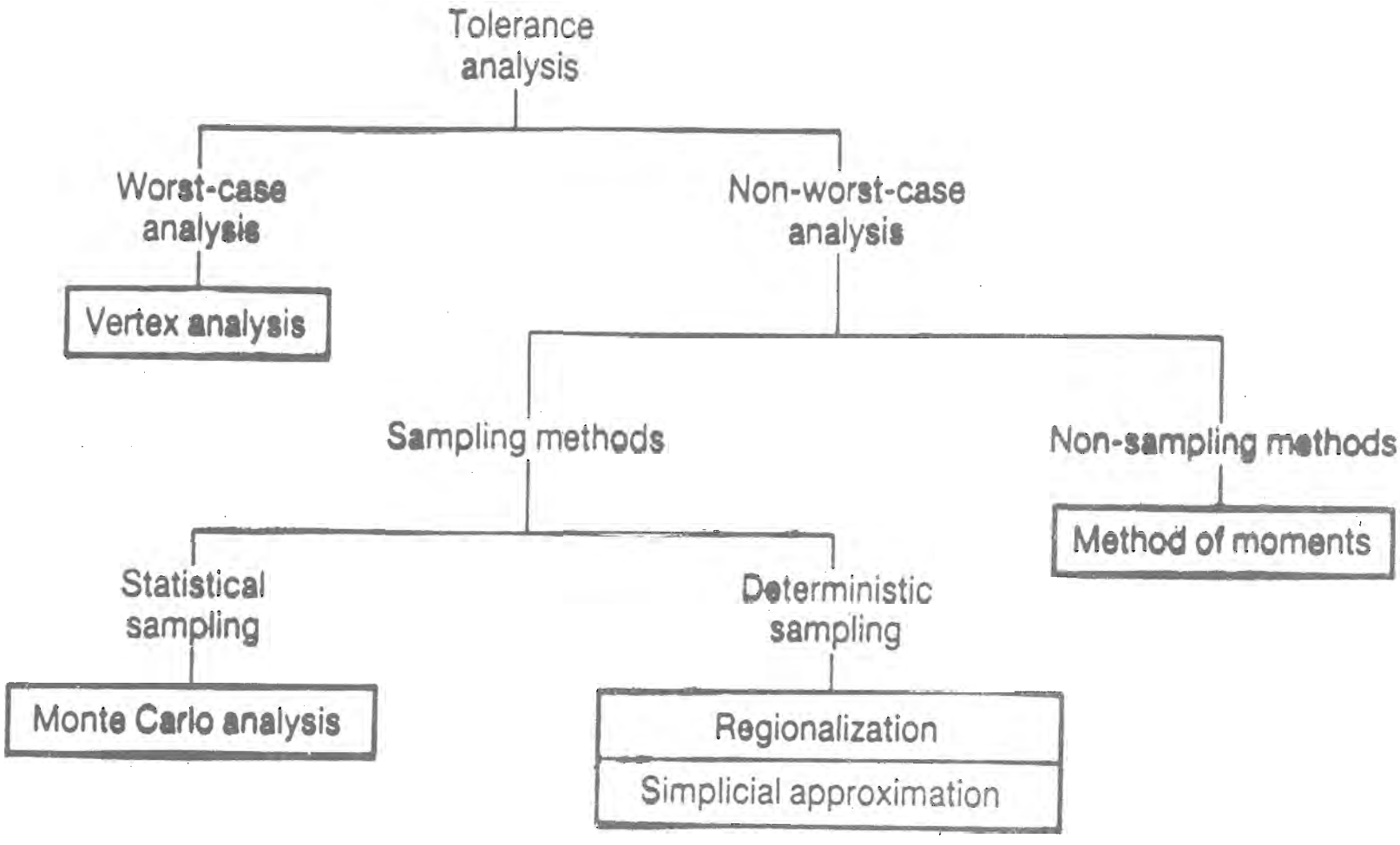

Manufactured
Circuits (dots)
R 2
R A
R T
R 1


R2
Acceptable
Circuits (shaded)
Yield = 50%
RA
RT
R1


New Circuit
(nominal R2 changed)
Yield = 100%
(because R_T completely within R_A)
R2
R_A
R1

What's the Problem ?

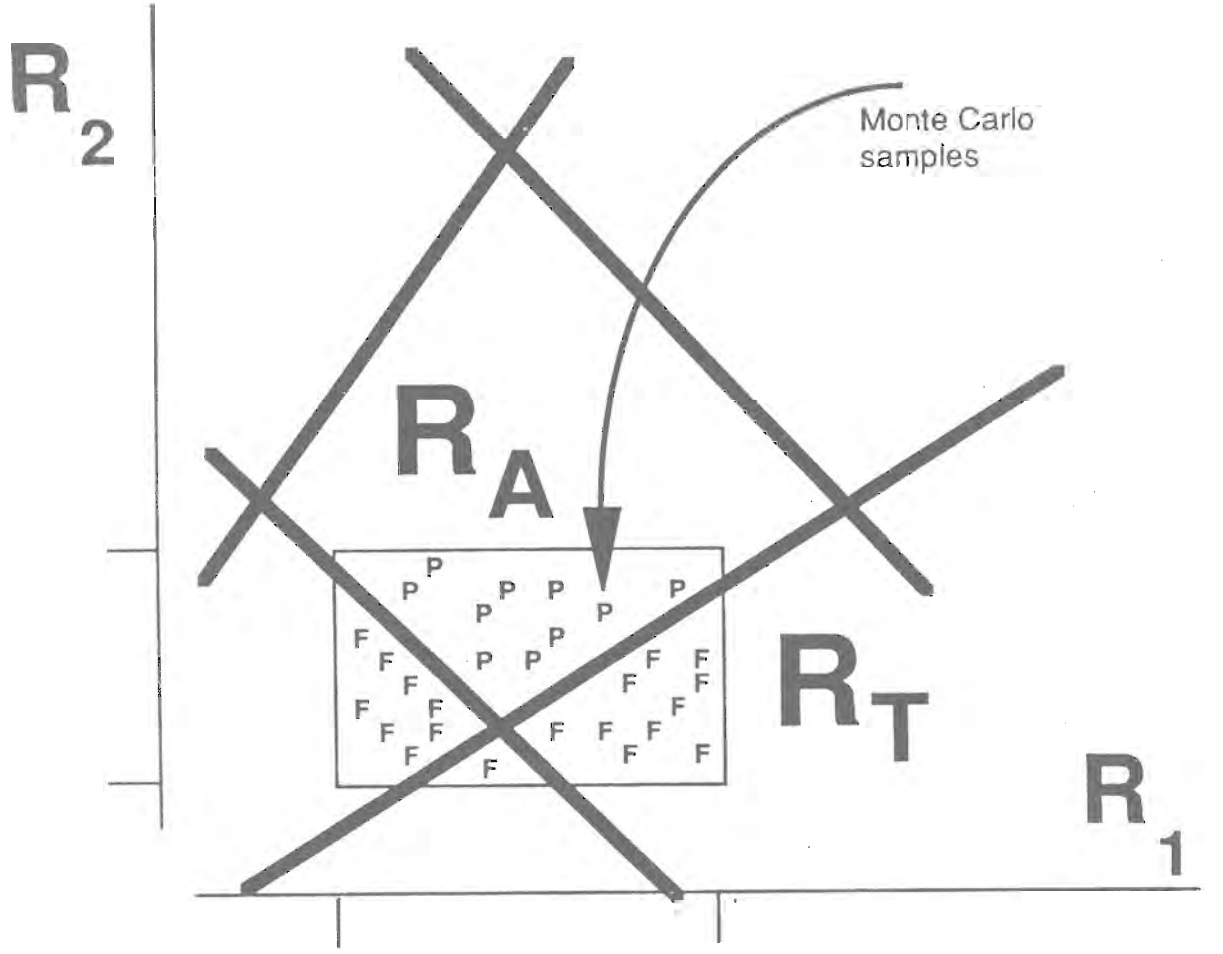
- Many components, not two
- R_T and R_A are multidimensional
- R_A unknown, very expensive to compute


Tolerance analysis
Worst-case analysis
Non-worst-case analysis
Vertex analysis
Sampling methods
Non-sampling methods
Method of moments
Statistical sampling
Deterministic sampling
Monte Carlo analysis
Regionalization
Simplicial approximation

Tolerance Analysis

Monte Carlo sampling

N samples randomly selected according to the parameter probability density functions. For each sample, performance evaluated and rated as pass or fail.



Suppose M samples satisfy the performance specfications.

Then, M/N is an *estimate* of the yield, and denoted by $\hat{Y}$

Monte Carlo sampling (continued)

Monte Carlo analysis, as well as generating N responses and estimating the yield, can allow a statement of the following form to be made:

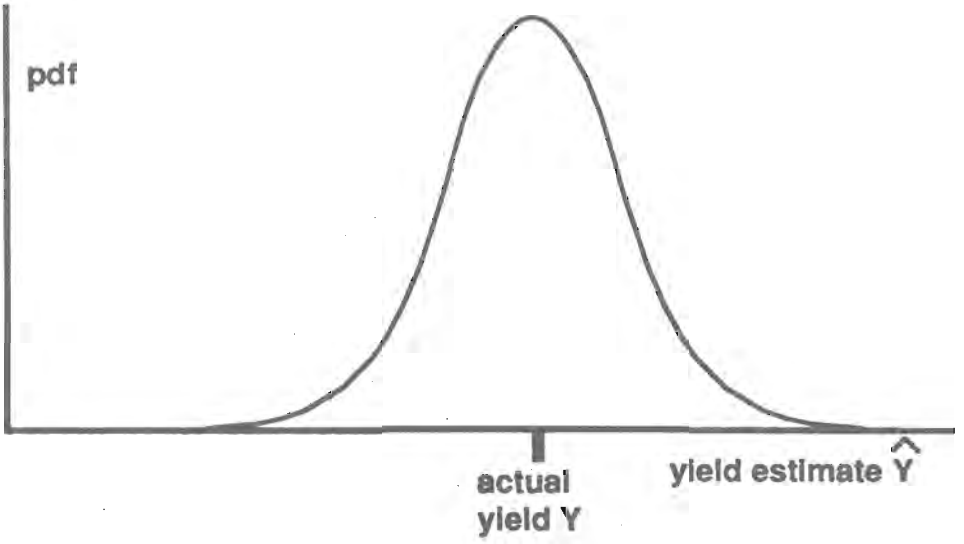
The estimate of yield is 63% and we can be 95% confident that the actual yield lies between 51% and 75%

An advantage of Monte Carlo analysis is that the confidence limits are independent of the number of parameters involved, and only depend upon the number of samples

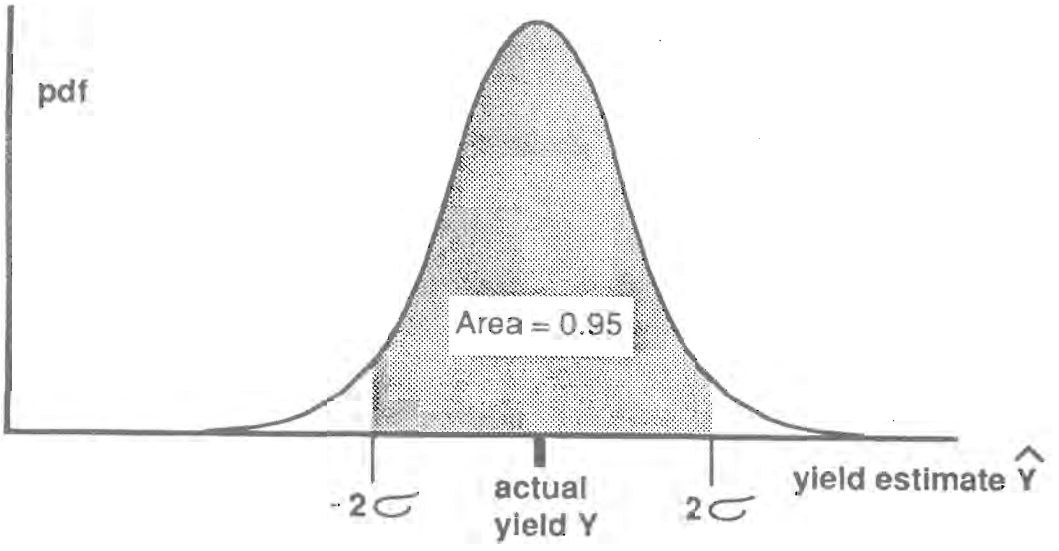
If repeated many times, a Monte Carlo analysis will produce different estimates of yield. In the limit of an infinite number of Monte Carlo analyses, the probability density function of the yield estimates can be shown to be Gaussian, with a standard deviation sigma given by

$$\sigma = \sqrt{\frac{Y(1 - Y)}{N}}$$

Since Y is unknown, the estimate of Y is used to calculate the standard deviation of the yield estimate



Monte Carlo sampling (continued)



Characteristics of Monte Carlo sampling

\+ **Any parameter distribution**

\+ **Predictable accuracy**

\+ **Dimensionally independent accuracy**

\+ **Provides valuable tolerance sensitivity and design information**

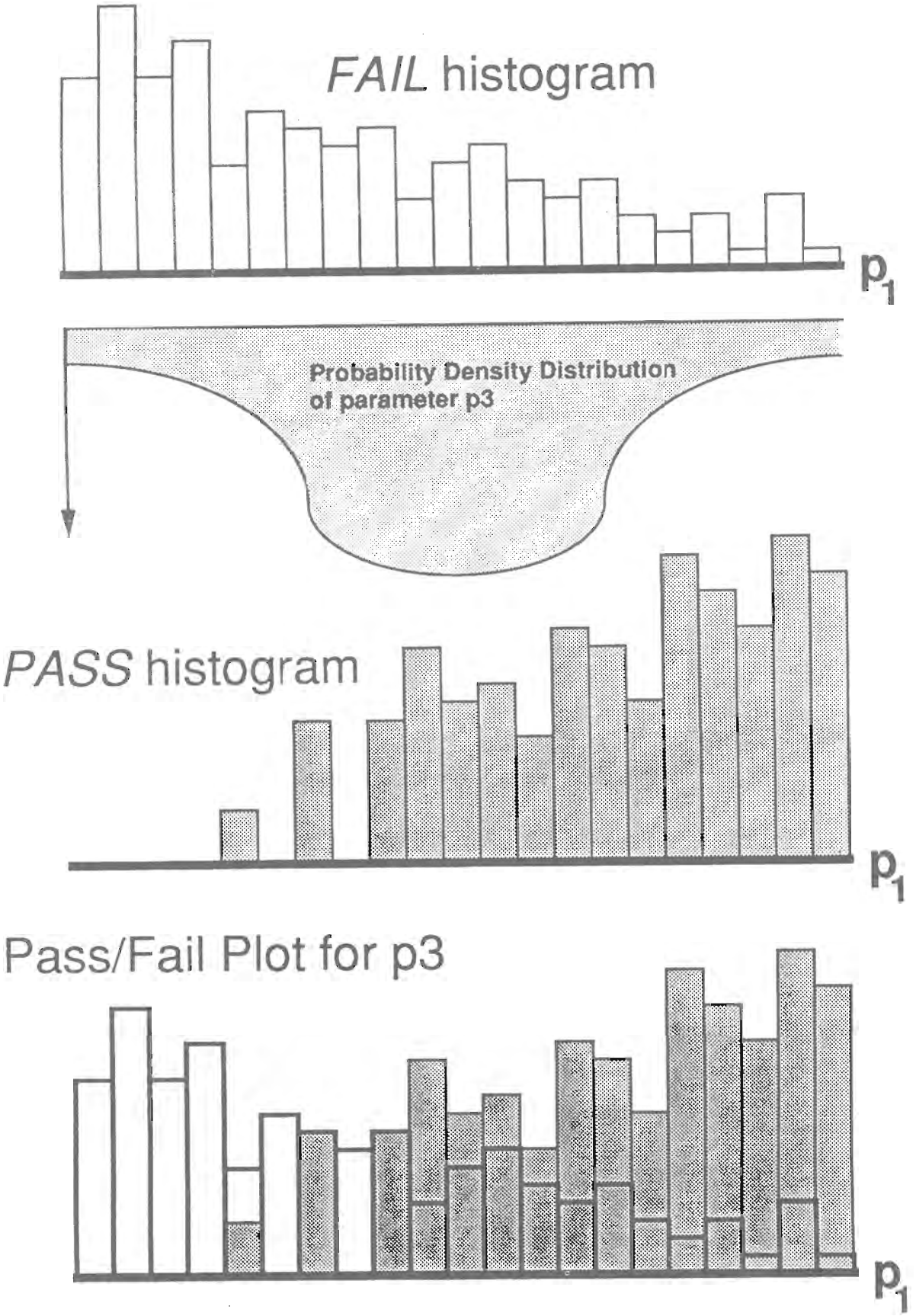
— **Accuracy proportional to $N^{-1/2}$**

Monte Carlo analysis results

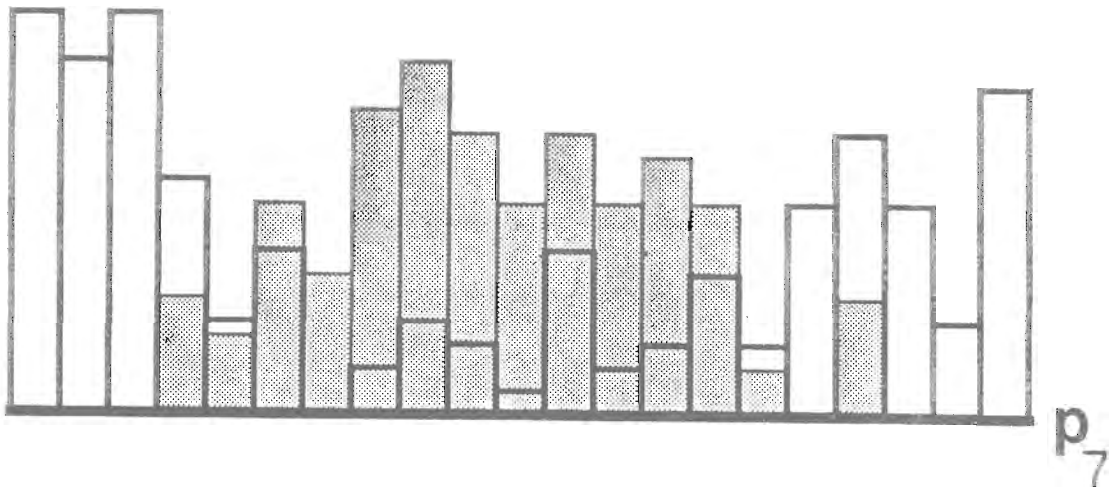
sample	p_1	p_2	p_3	p_4	pass/fail
1	5.04				1
2	5.18				0
3	4.96				1
4	4.82	values			1
5	5.11				0
6	4.98				0
7	5.08				1
etc.	etc.	etc.	etc.	etc.	etc.

Quite costly . . . .

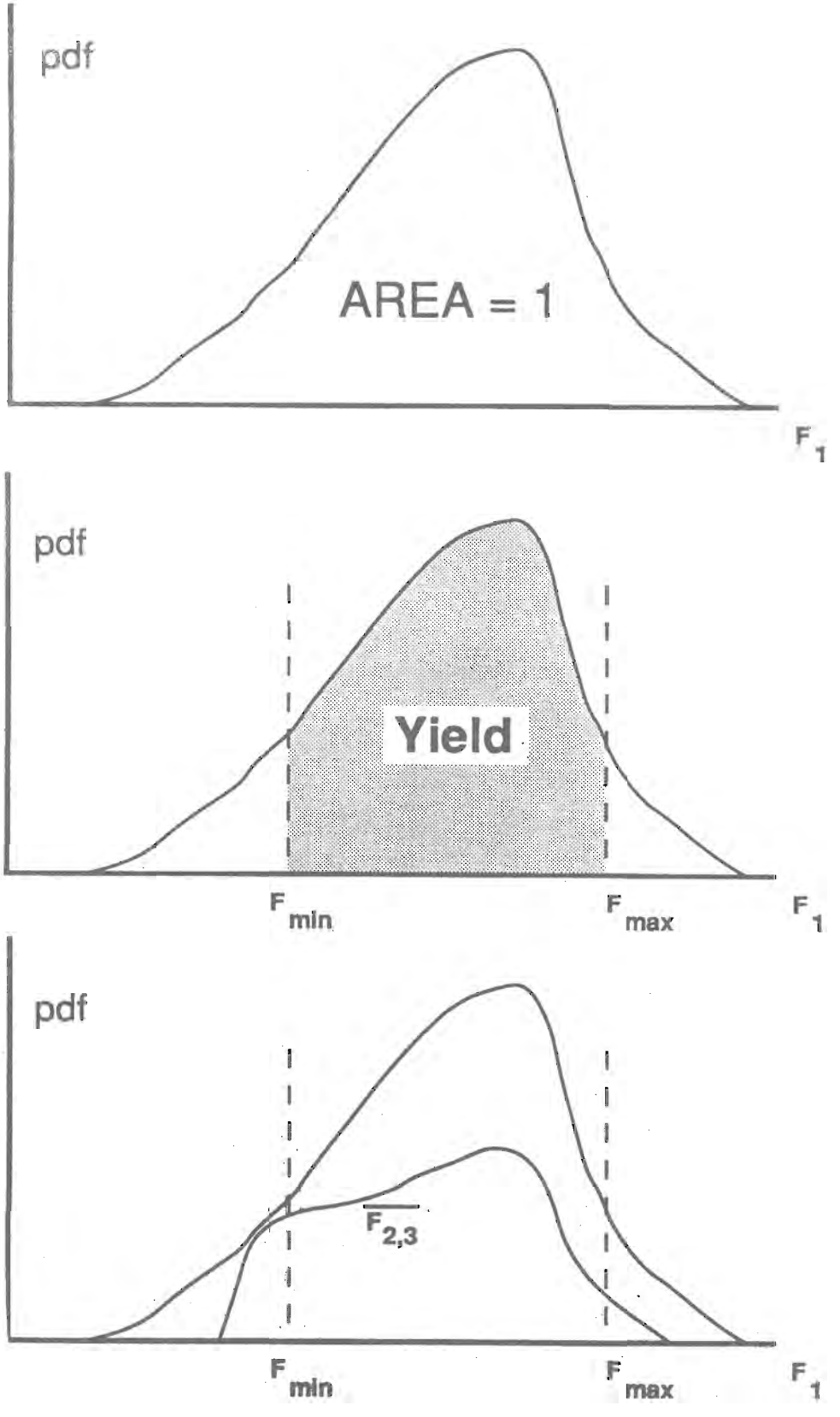
Interpretation ?

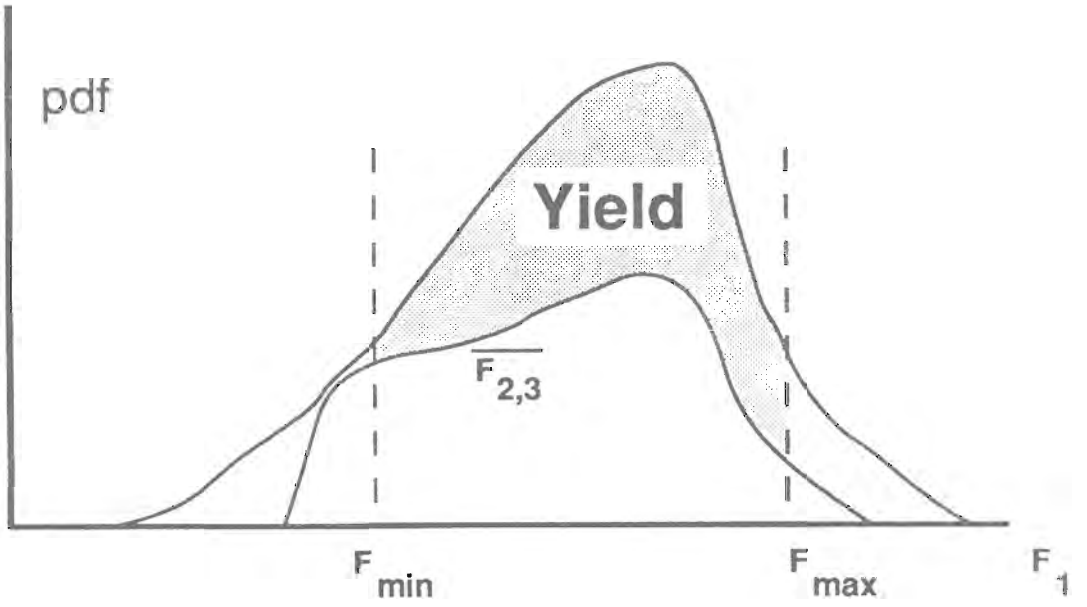

FAIL histogram
p_1
Probability Density Distribution
of parameter p3
PASS histogram
p_1
Pass/Fail Plot for p3
p_1

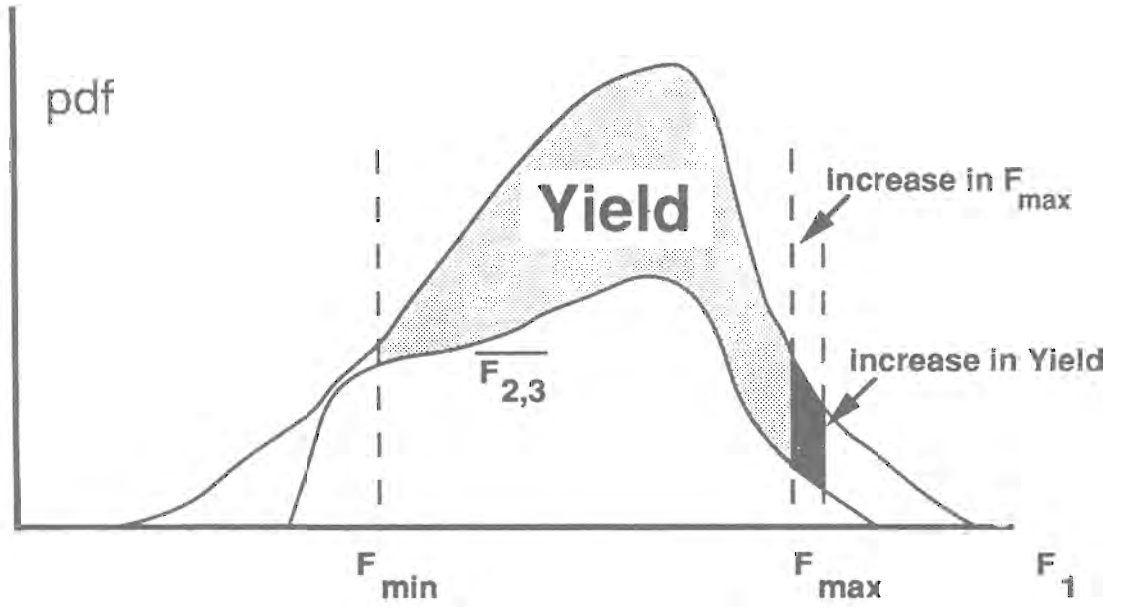
PASS/FAIL Plot for p7



Specification Sensitivity




pdf
Yield
$\overline{F_{2,3}}$
F_{min}
F_{max}
F_1


pdf
Yield
Increase in F_{max}
Increase in Yield
$\overline{F_{2,3}}$
F_{min}
F_{max}
F_1

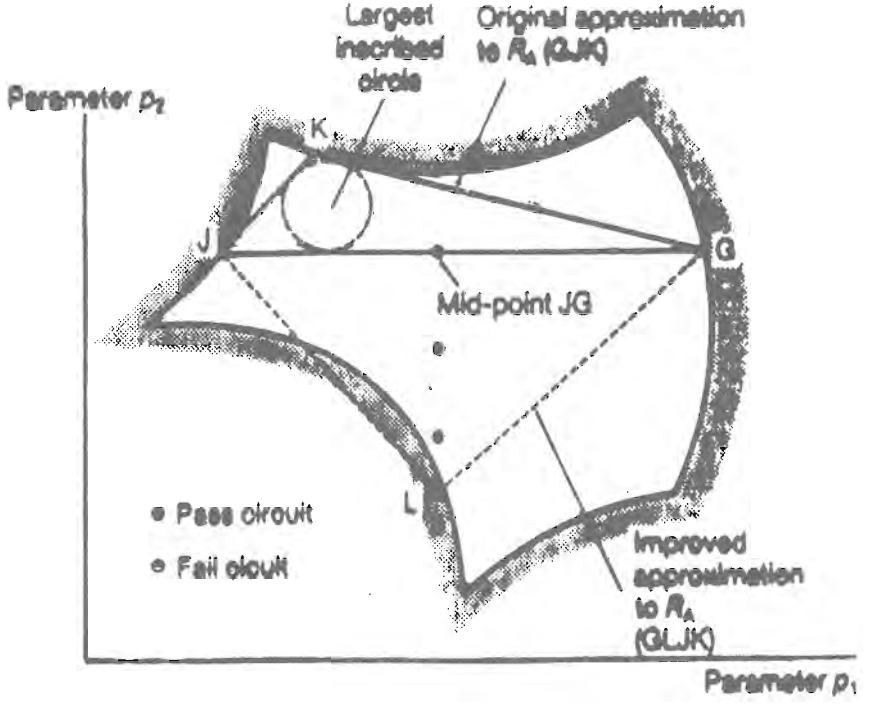


Figure 6.8
Expansion of the approximation to R_A to include a new point (L) identified by a search from the midpoint of the largest side of the original approximation (GJK) tangential to the largest inscribed circle.

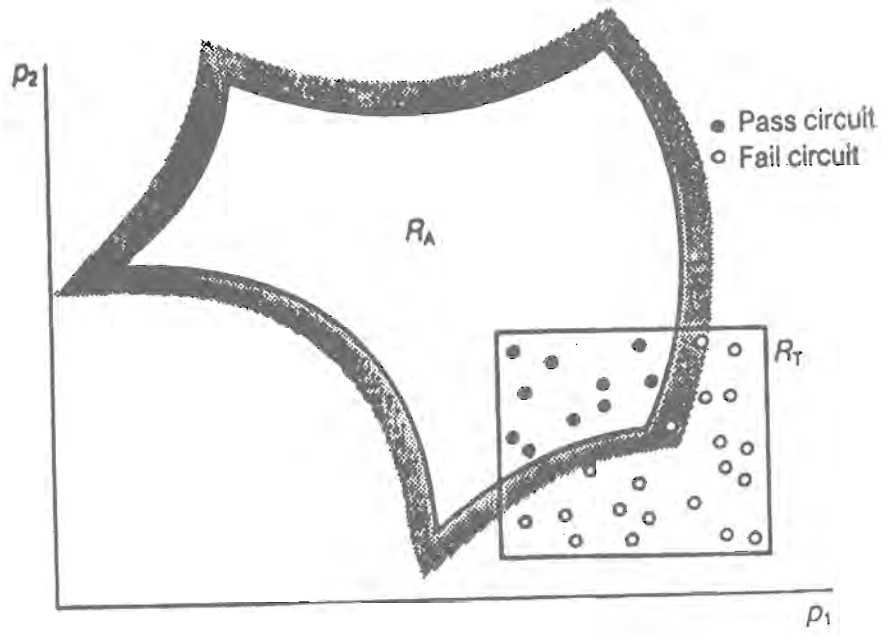


Figure 6.10
Statistical exploration involving Monte Carlo analysis: the simulation of randomly selected circuits within the tolerance region. The estimated yield is the fraction (here 10/30 or 33%) of simulated circuits which satisfy all the specifications.

Table 6.1 The advantages and disadvantages of the deterministic and statistical exploration approaches.

Deterministic	Statistical exploration
Advantages	*Advantages*
Good for 'worst case'	Independent of number of toleranced parameters
Efficient for small number of parameters	No assumptions about the nature of R_T, R_A
Mathematical rigour	Yield accuracy predictable
Disadvantages	*Disadvantages*
Only useful for small number (up to 5–8) toleranced parameters	Expensive to guarantee high yields (e.g. 99%)
Maximizes only a lower bound on yield	Largely heuristic algorithms employed
Limiting assumptions about nature of R_T, R_A and probability distributions	
No means of predicting the accuracy of the yield estimate: only a lower bound can be found under certain conditions (e.g. if R_A is convex)	

Table 6.2 Some criteria to assist the evaluation of a tolerance design scheme.

(1) How general is the method?
  (a) Are assumptions made about R_T, R_A and $\phi(p)$?
  (b) What is the maximum number of toleranced parameters?
  (c) What is the maximum number of design variables?

(2) What information does it give me?
  (a) Will it do design centring, tolerance assignment, ... and for what aspects of circuit performance?
  (b) Can I easily change the specifications?
  (c) Is it better for worst-case design (e.g. 100%) or less than 100% yield?
  (d) How accurate is the method?
  (e) What are its convergence properties like?

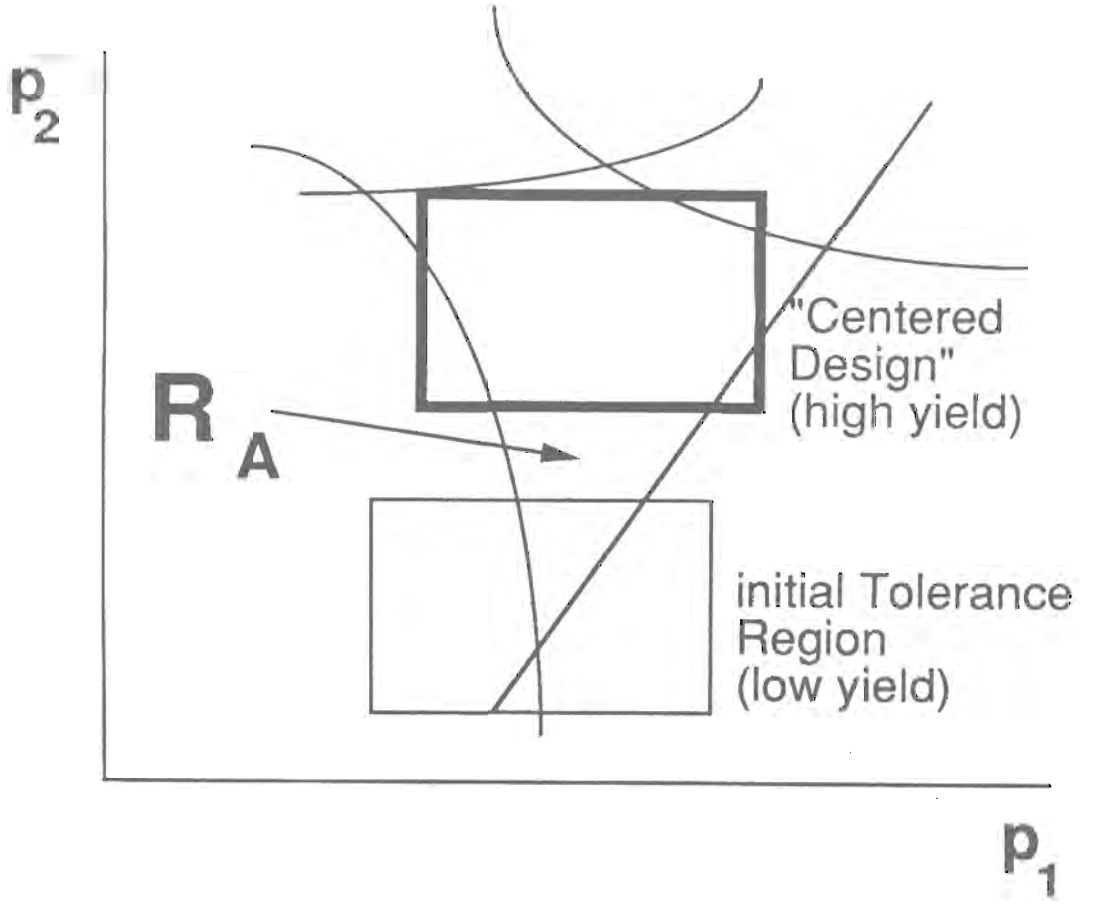
(3) What does it cost to use?
  (a) How many circuit analyses to find the yield or R_A?
  (b) How many circuit analyses (typically) to improve the yield?
  (c) How severe are the computational overheads?

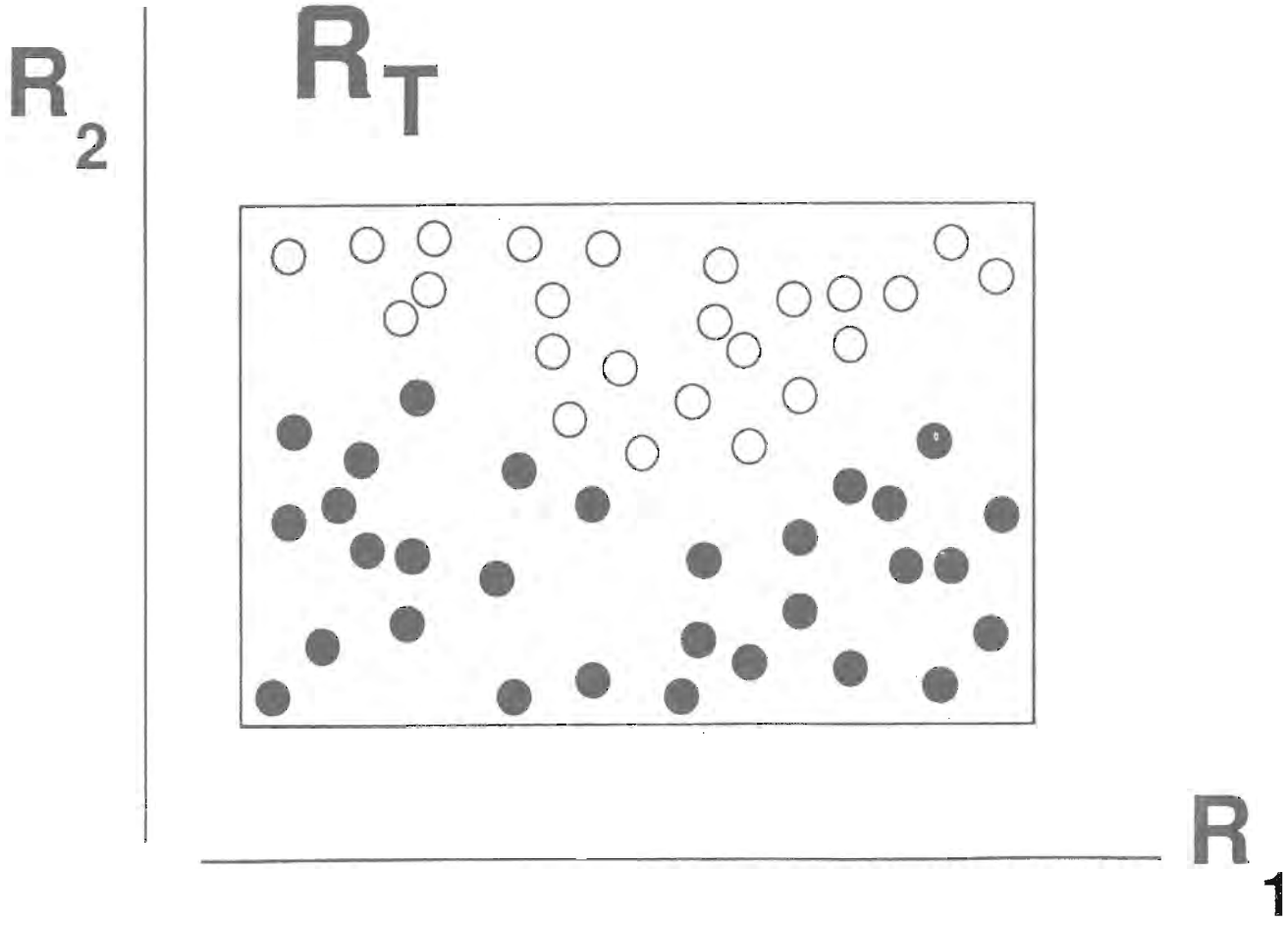
(4) Ease of implementation
  (a) Are the mathematics comprehensible?
  (b) Can the method use an already existing circuit analysis package?

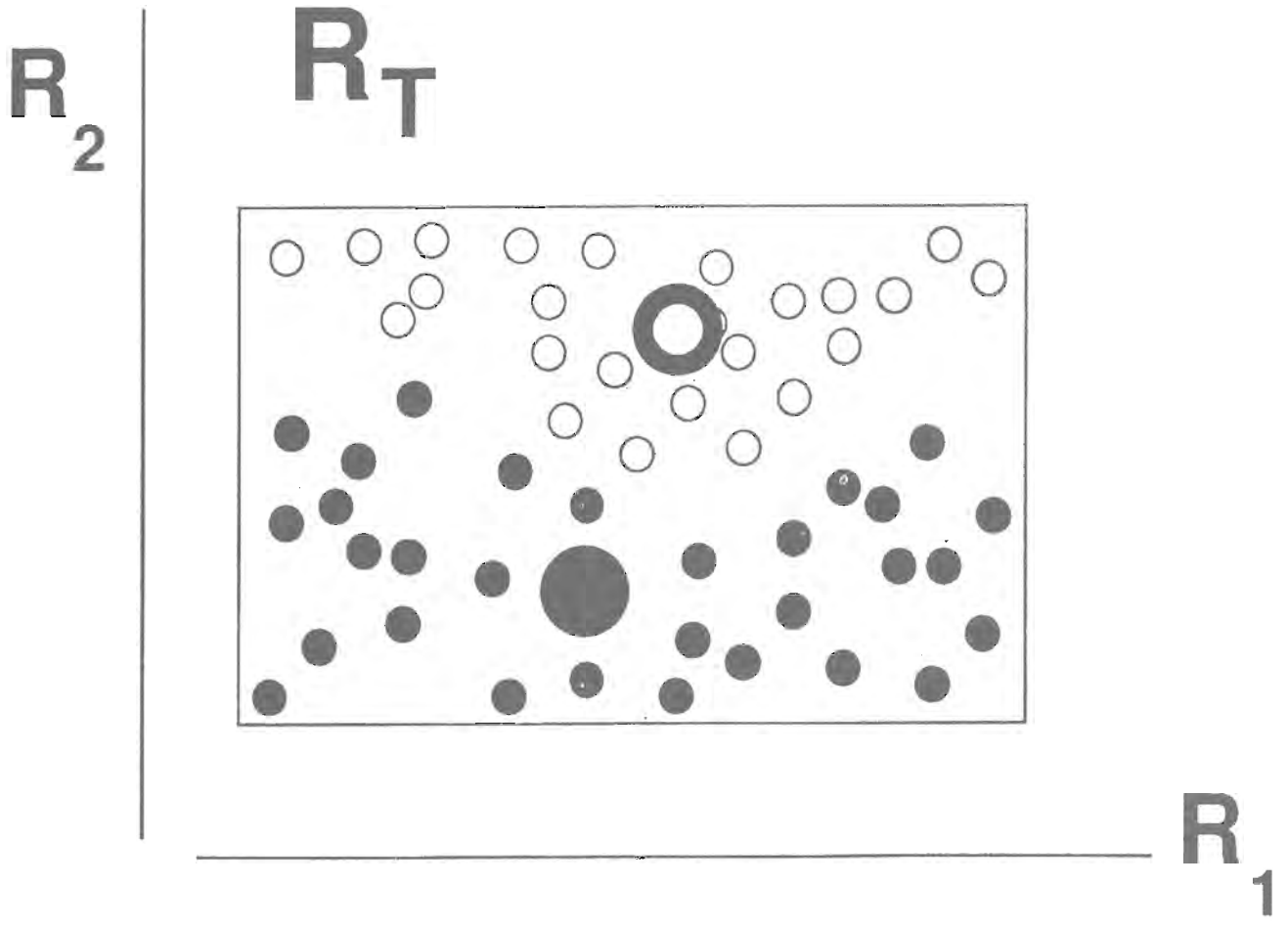
Design Centering

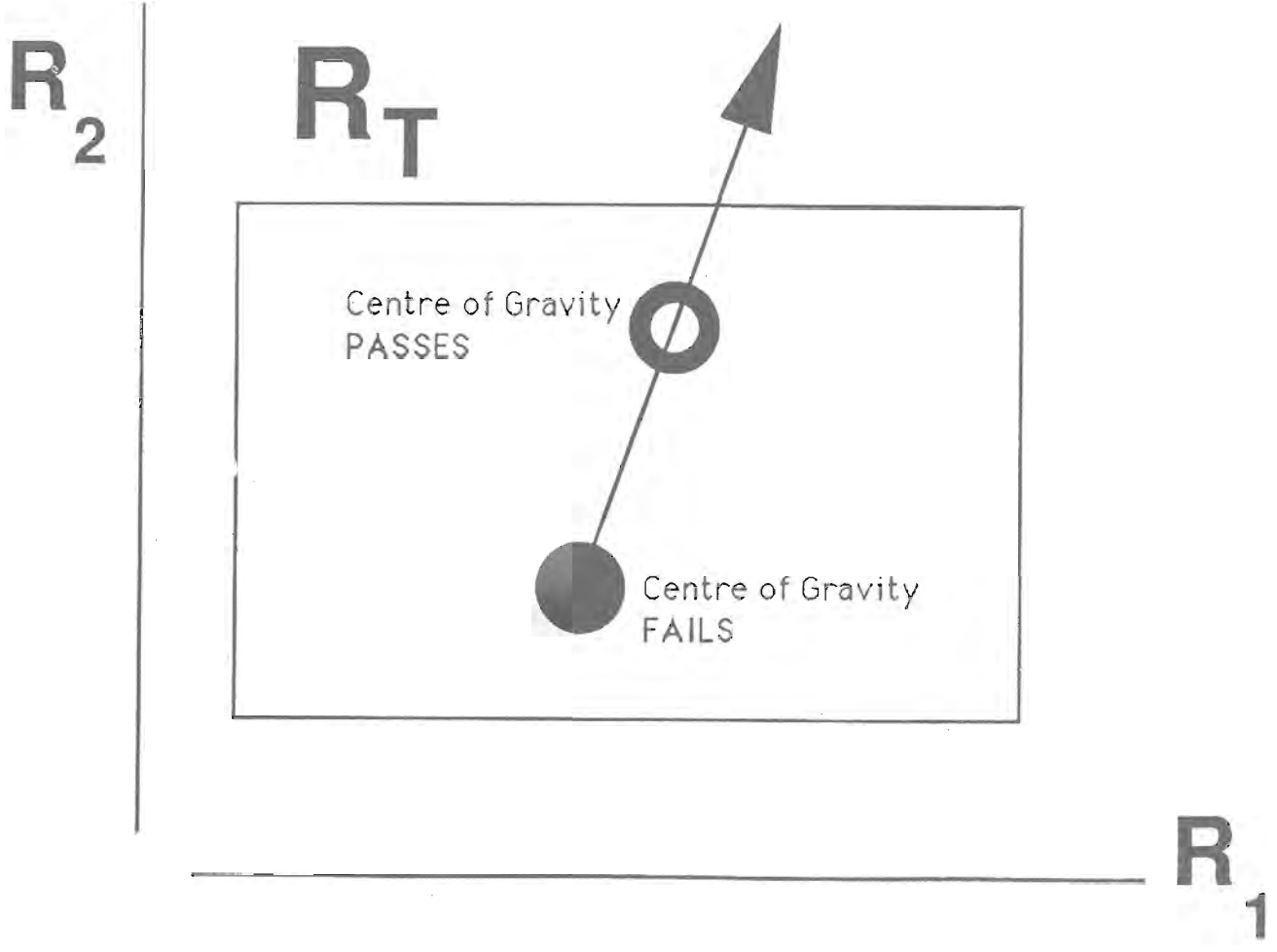
Often, the Tolerance Region R_T will not fit completely inside the Region of Acceptability R_A, so that 100% yield cannot be achieved.

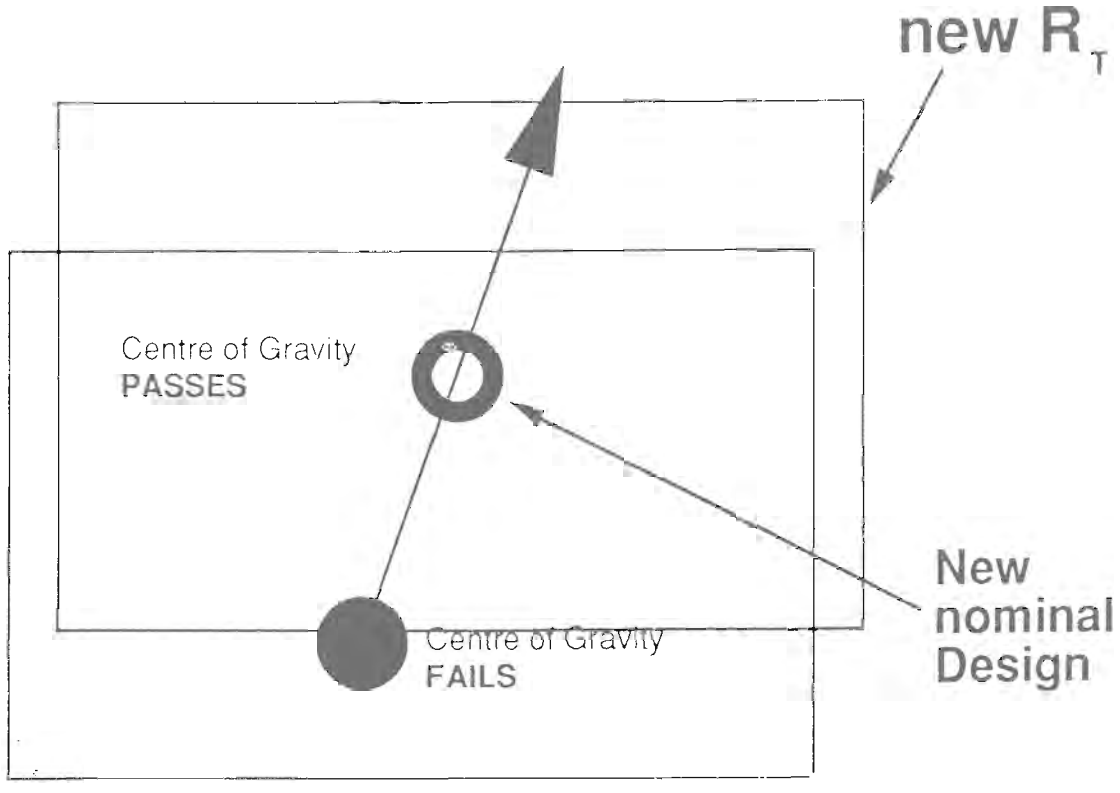
Adjustment of the nominal design (without changing tolerances) to ensure the maximum yield is referred to as Design Centering.



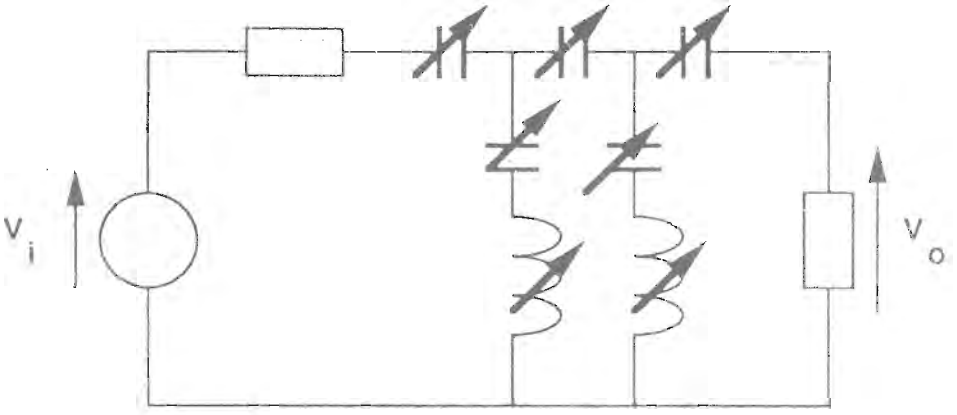

R2
RT
R1


R2
RT
R1

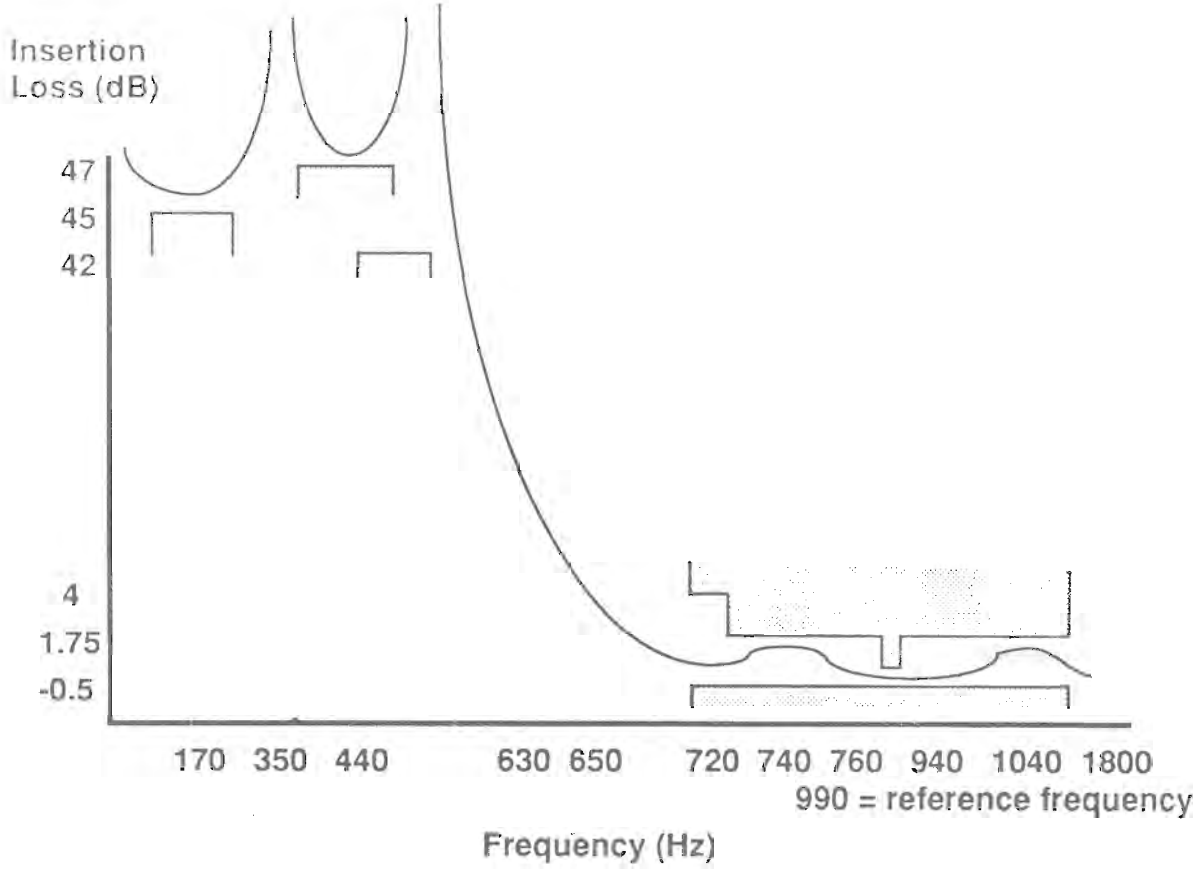

R2
RT
Centre of Gravity
PASSES
Centre of Gravity
FAILS
R1


new R_T
Centre of Gravity
PASSES
Centre of Gravity
FAILS
New
nominal
Design

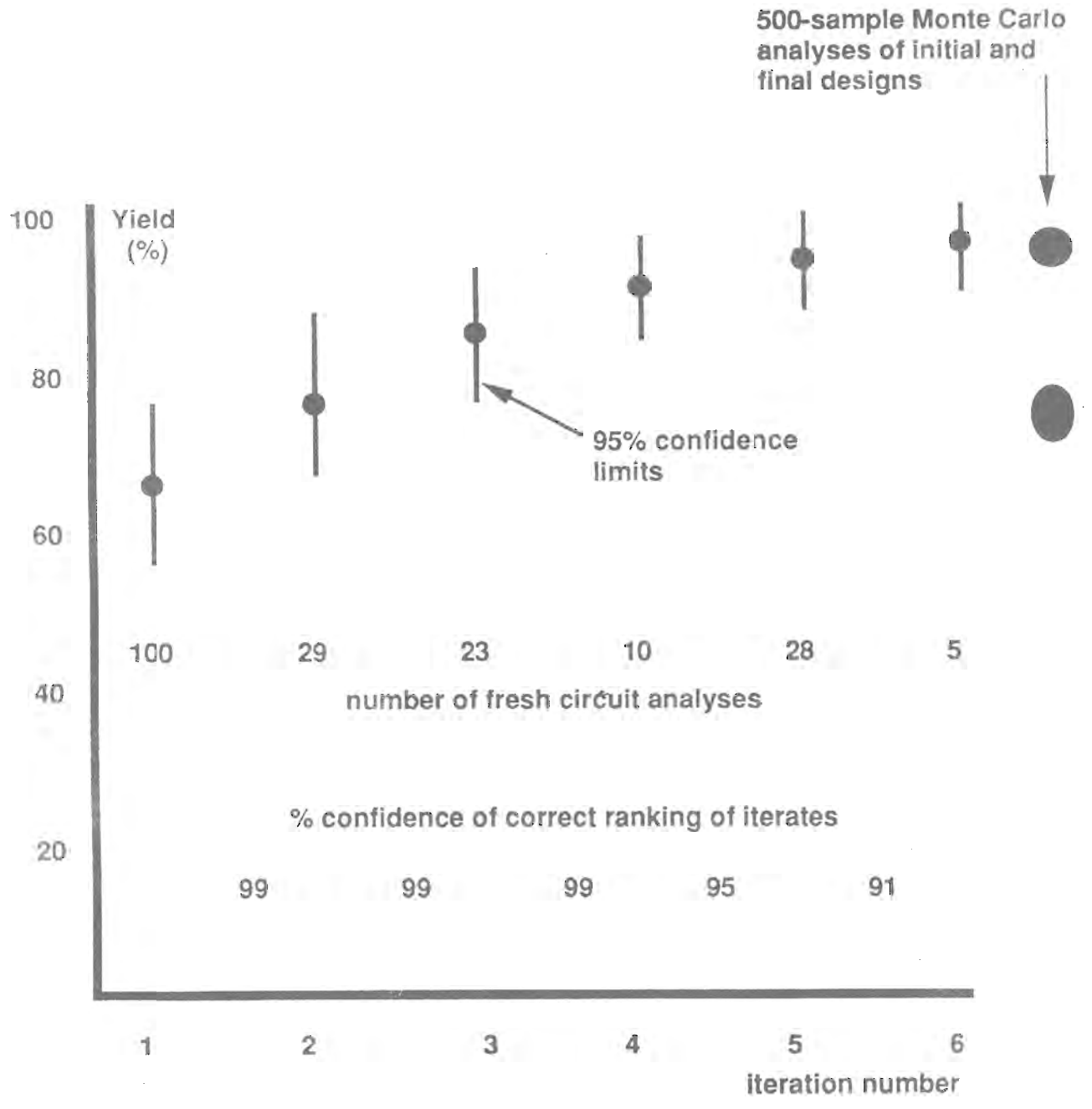
Circuit



Performance Specifications

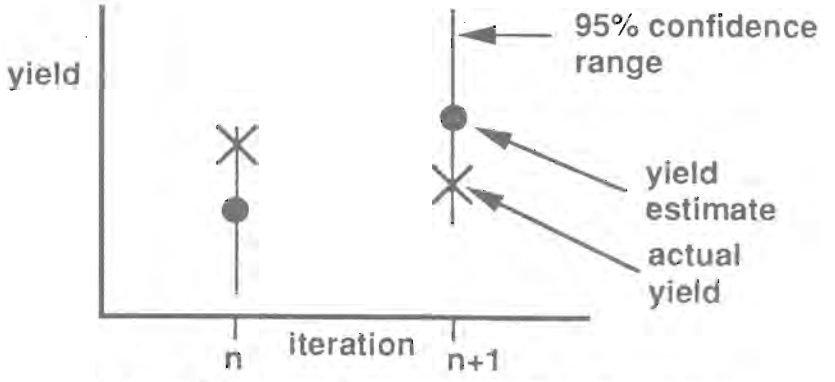


Design Centering by the Centers-of-Gravity Algorithm

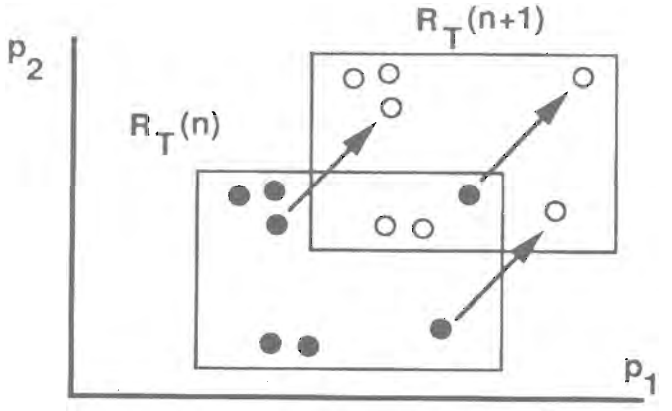


Design Centering by the Centers-of-Gravity Algorithm

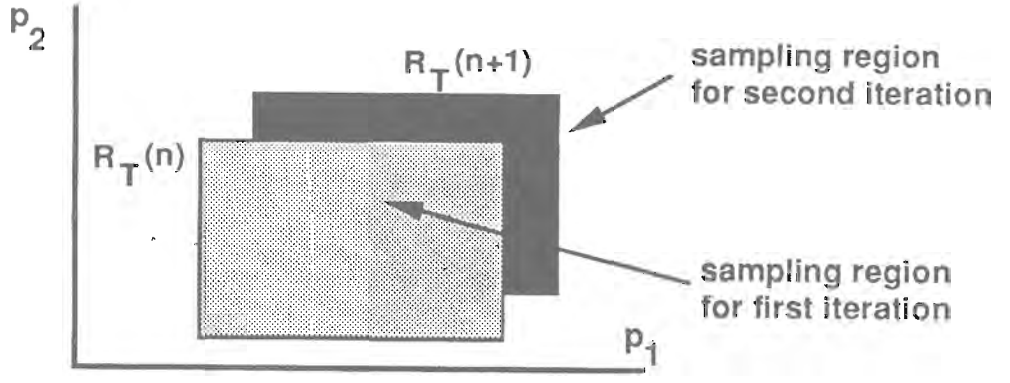
Problem: increase in yield estimate -
- but does the actual yield increase or decrease ?



Correlated Sampling (allows the confidence ranking of iterates)



Common Points Sampling (allows confidence ranking of iterates and saves on samples)



NA-12-133

SPECIFICATION CAPTURE AND YIELD ENHANCEMENT IN AN INTERACTIVE ENVIRONMENT

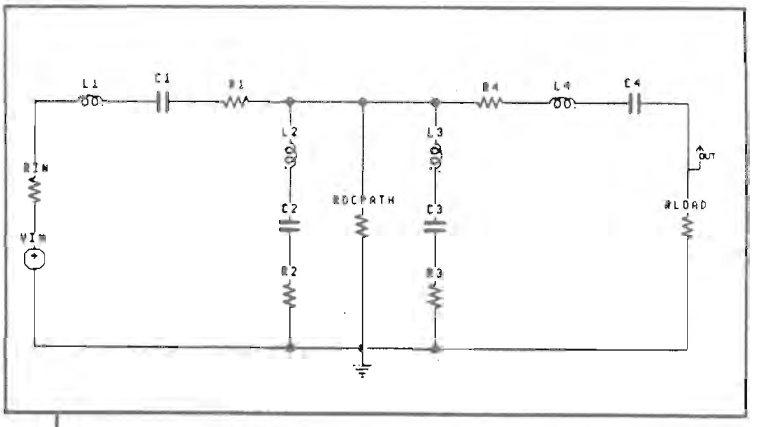
Rik de Boer, Paul Jennings and Paul Rankin

Increasing Manufacturing Yield

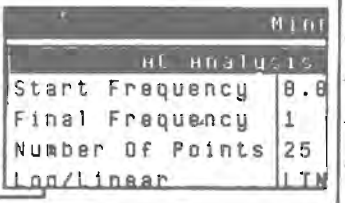
We describe a new tool for the analogue circuit designer, a tool concerned particularly with component tolerances and the unwanted effect these have on the performance of a mass-produced circuit. It assists in the redesign of a circuit to increase the manufacturing yield and reduce component costs. Like some conventional tools, this one simulates the effect of component variations on circuit performance and, by checking against performance specifications, estimates the manufacturing yield. Unlike others, however, it not only provides the designer with diagnostic information regarding especially sensitive parameters and specifications, but will also automatically adjust the nominal values of components in order to maximise the yield. Any parameter, such as the width of a device or the value of a resistor, may be designated as adjustable by the designer.

A design sequence

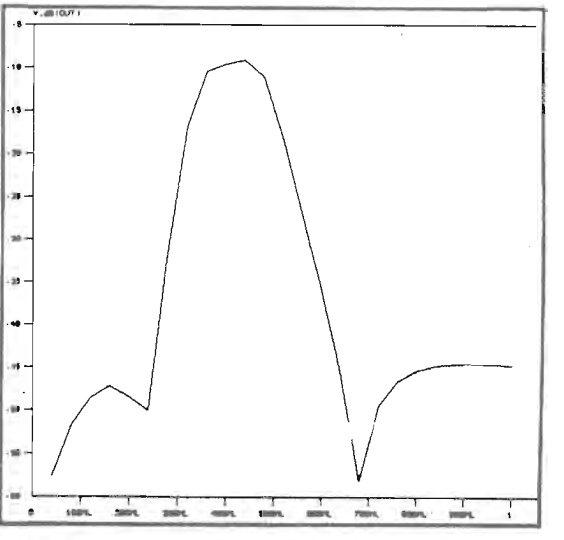
Since the purpose of this paper is to describe a new tool, none of the underlying theory, already published [1], will be referred to. Rather, the exposition will be largely in 'storyboard' form, illustrating typical use of the tool. All screen shots are of the MINNIE analogue circuit design environment as supplied by Interactive Solutions Ltd. and in general commercial use.



Using simple schematic entry, the designer describes a band-pass filter, and selects analysis in the frequency domain.

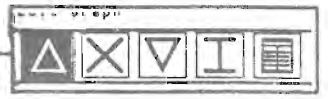


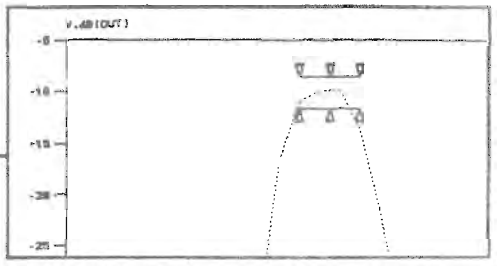
After the display of the simulated performance



Mr. de Boer and Mr. Jennings are with EES Inc., 16 The Grampians, Shepherds Bush Road, London W6 7LW and Mr. Rankin is with Philips Research Laboratories, Redhill, Surrey.

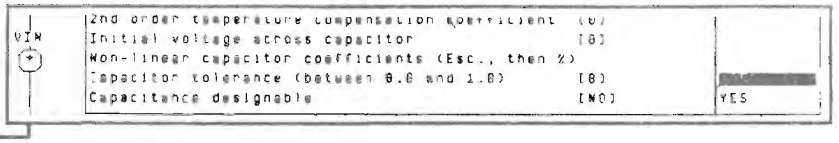
the designer then uses simple icons to attach the performance specifications, in the form of upper and lower limits, to the displayed results.

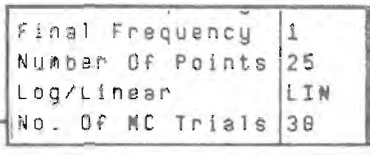


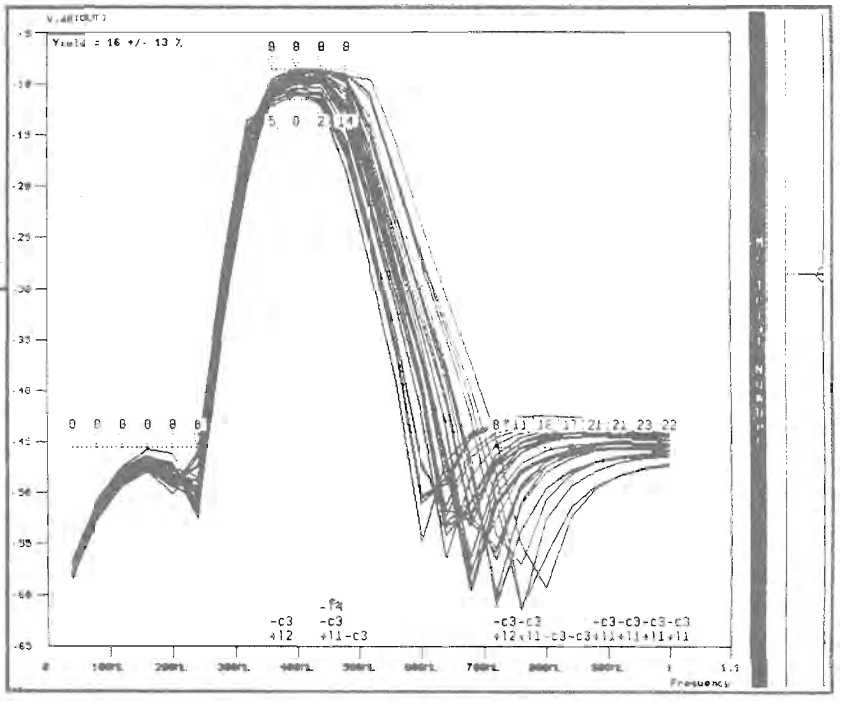


Component Tolerances

Component tolerances are assigned by selection from component parameter menus, and designable parameters are identified. A Monte Carlo analysis to estimate performance spread is initiated following selection of the number of samples (simulations) but mindful of the fact that more simulations means more accuracy. The result of the Monte Carlo analysis not only provides an estimate of the yield (here 16% with 95% confidence that the actual yield lies between 3% and 29%), but also a display of all simulated performances. The specification limits can be examined in detail to see how many samples fail each limit.

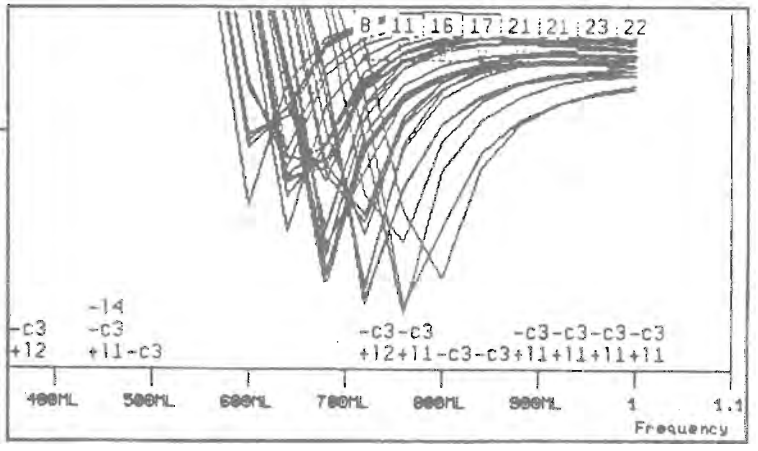




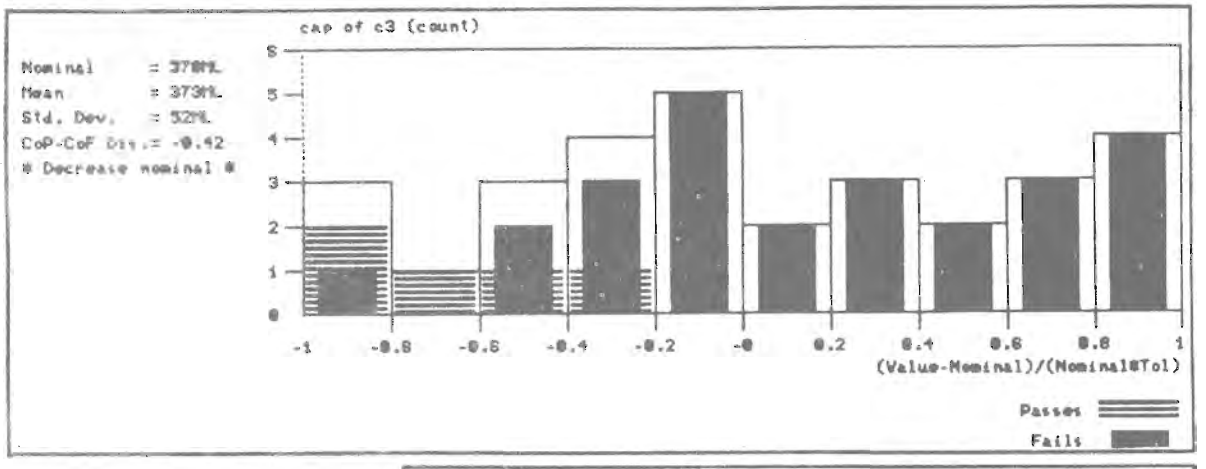


Diagnosis

Diagnostic information, showing which parameters were principally responsible for the unsatisfactory yield, is displayed on the independent axis. Further diagnostic information which indicates in some detail which nominal parameter values should be



adjusted is provided in the form of Pass/Fail plots. These plots show, for each parameter taken separately, the distribution of passing and failing circuits in the Monte Carlo sample. Thus, for C3 (see below), many more circuits passed the specifications when C3 was towards the lower end of its range suggesting, as does the displayed message, that its nominal value might usefully be reduced.



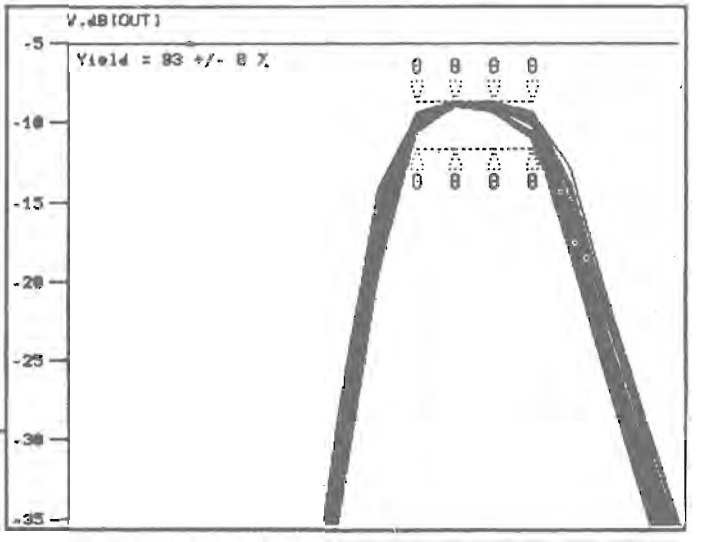
Pass/Fail plots can also indicate whether there is scope for increasing a component tolerance, thereby lowering its cost, or if a tightening of the tolerance is necessary to improve the yield.

Centres of Gravity (Auto Step Factor)

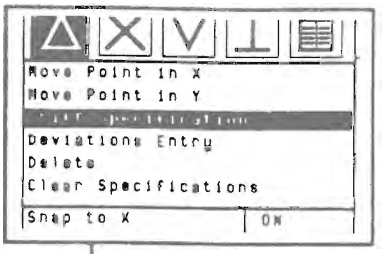
Parameter name	Current	Suggest
ind of l1	1.65	1.6781
cap of c1	82ML	81.774M
ind of l2	470ML	474.54M
cap of c2	1	1.0035
ind of l3	150ML	149.93M
cap of c3	378ML	384.15M
ind of l4	1.8	1.8288
cap of c4	82ML	81.537M

Tolerance Design

The suggested new values which, on request, will automatically be assigned as the nominal values of the designable parameters, are displayed in a Table (see above). This provides the designer with the opportunity, for what ever reason, to temporarily hold one or more designable parameters fixed. After MINNIE has automatically made the required adjustments, the designer is then free to carry out a second Monte Carlo analysis on the new circuit to check its estimated yield and, if thought appropriate, to repeat the process as many times as necessary: three, four or five times might be typical. In the selected example of the bandpass filter, five iterations led to an increase in estimated yield from 16% to 93% and, as the display indicates, a smaller performance spread.

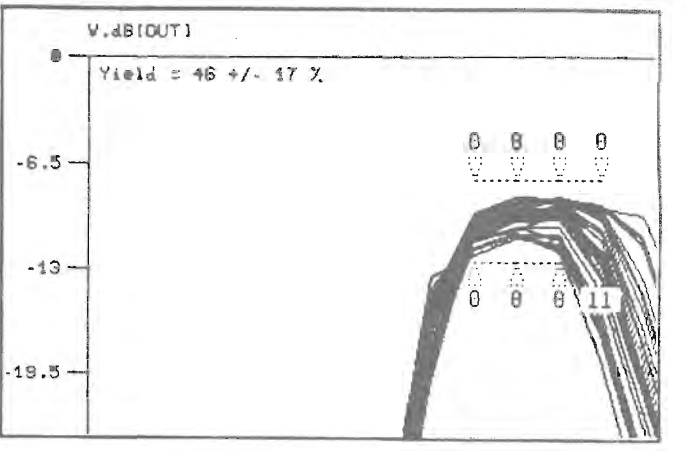


Higher accuracy in the yield estimate can always be obtained from a final Monte Carlo analysis with a much larger number of samples.



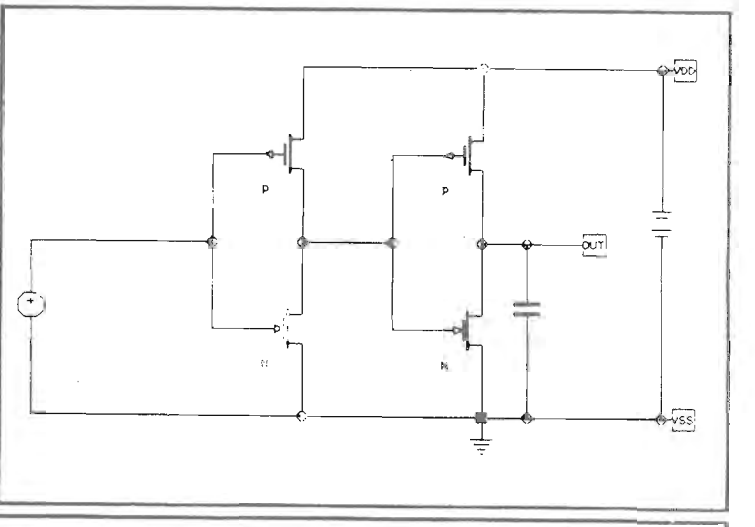
Yield-Specification Trade-off

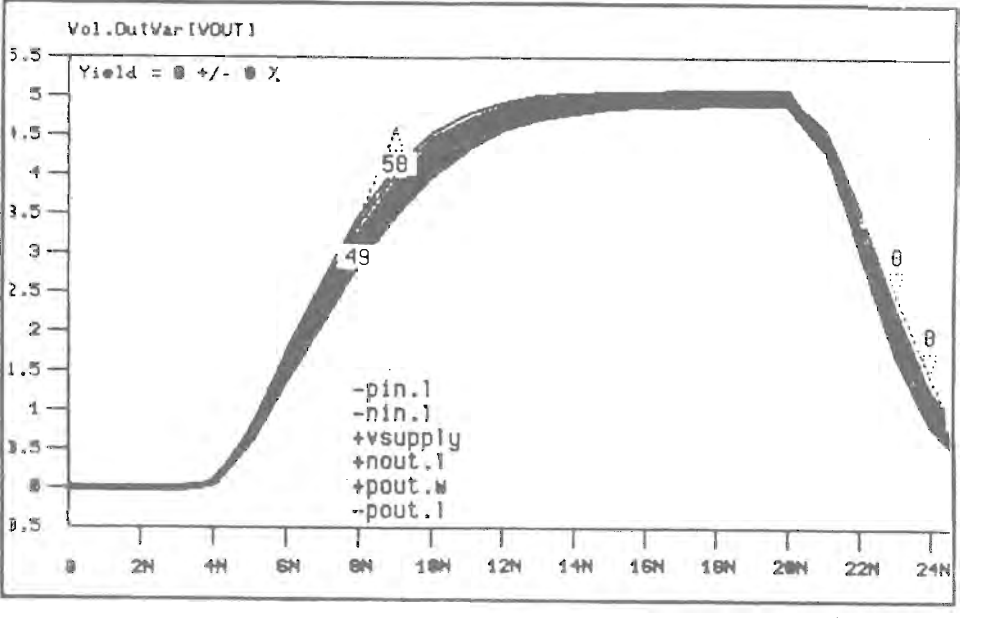
If the designer wishes to see how a change in the specification limits affects the yield, this can be explored by interactively changing the specification limits, whereupon the new yield and the new number of failures for each limit are immediately displayed. Indeed, the fact that the *partial* yields with respect to each specification are displayed enables the designer to quickly ascertain where the yield is being lost and to investigate, interactively, any potentially conflicting specifications



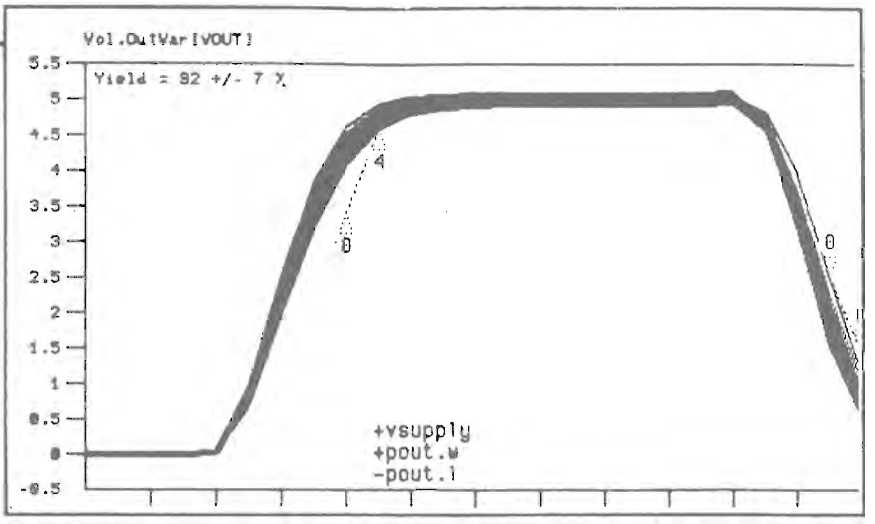
Integrated Circuit Design

As already remarked, the designable parameters can include the lengths and widths of integrated devices. An illustrative example is provided by a CMOS buffer whose transient response to a defined input waveform was specified, as was the input capacitance of the buffer. Tolerances were assigned to the supply voltage (±2%), the capacitive load (±5%), and the device lengths and widths (±5%). From an initial design providing zero estimated yield, eight iterations were required to arrive at a new design: only the second stage widths needed adjusting.





The new design exhibited an estimated yield of 92% and a smaller spread in response. Again, it would be possible to interactively adjust the specifications related to the rising and falling parts of the waveform in order to see the influence on the estimated yield.



Complex Performance Specifications

The tool we describe can handle, concurrently, performance specifications within more than one domain (DC, AC and Transient) and, within any one domain, specifications on a number of properties (e.g., gain magnitude, impedance and current phase in the frequency domain). Also, specifications can, if desired, be allowed to vary as a function of external parameters such as temperature and power supply voltage.

Conclusions

Failure to take account of component tolerances at the design stage is unfortunately commonplace, with the attendant risk of enormously expensive circuit failures. The tool described allows the effect of tolerances not only to be taken into consideration, but compensated for, at a very early stage in the design. This results in a direct reduction in design time, since a comprehensive assessment of the circuit's performance with respect to its specifications can be made. Overdesign can be eliminated, with consequent reductions in manufacturing costs, since the circuit is more robust. The new tool therefore provides a direct means of increasing manufacturing profitability.

Acknowledgements

Much of the development reported here was carried out at Twente University, Enschede, The Netherlands. The specification editor was developed in collaboration with Philips Research Laboratories, Redhill, U.K.

References

[1] Spence, R. and Soin, R.S., Tolerance Design of Electronic Circuits, Addison-Wesley, 1988.

[2] Rankin, P.J. and Siemensma, J.M., 'Analogue Circuit Optimization in a Graphical Environment', Proc. ICCAD, pp.372-375, 1989.

VLSI Design

Dr Alan Pearmain

Queen Mary & Westfield College, University of London

1. Suitable applications for VLSI design

The range of possible design styles for VLSI design means that Application-Specific Integrated Circuits (ASICs) are now suitable for a vast range of electronic circuit implementation. The term VLSI (Very Large Scale Integration) design is often used synonymously with ASIC design, although VLSI should only apply to high-density design. No exact definition exists, but definitions that have been proposed are that VLSI is design in a technology with minimum feature size of <2μm, or that it is design with more than 100,000 transistors on a chip. The first definition incorporates virtually all current ASIC design whereas the second would be more restrictive.

The areas where ASIC design is not practical are those where large powers must be handled but there are possible design routes with ASICs for almost all other applications.

Common to all ASIC design is the need for Computer-Aided Design (CAD) tools. The level of complexity of the design process makes these essential and the fabricator will require output on tape or disk from the CAD system as input to the fabrication process.

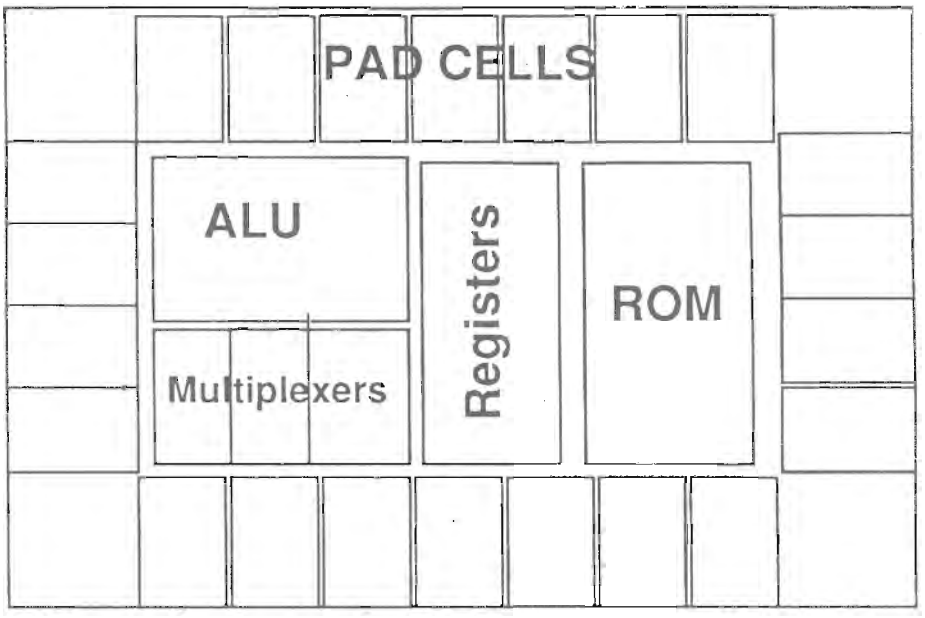
The available design style are: full custom, standard cell, gate array or field-programmable gate array. The one used will depend on the performance required from the circuit being designed, the anticipated production run for the device, the timescale required for delivery and the capital available.

ASICs are suitable even for prototype work and field-programmable gate arrays are becoming very popular in the development of relatively complicated prototypes.

2. Types of VLSI design

2.1. Full Custom

In full-custom design the designer has complete freedom to place transistors anywhere on the chip and to make any interconnections that he requires. The design will have a functional block structure with pad cells for connections off-chip around the edges. The figure below shows a typical microprocessor arrangement.



Advantages:

- The design can use the latest fabrication technology.
- The highest signal speeds are possible.
- The highest levels of complexity are possible.
- Maximum use of silicon area makes the marginal cost in volume production the lowest

Disadvantages:

- The CAD tools required are more expensive & require considerable computing resources.
- The time taken to develop a new design may be high, in some cases > 1 year.
- Skilled designers are required.
- High NRE costs. A complete set of masks is required - at least 9 masks costing around $15,000 in addition to the design costs.

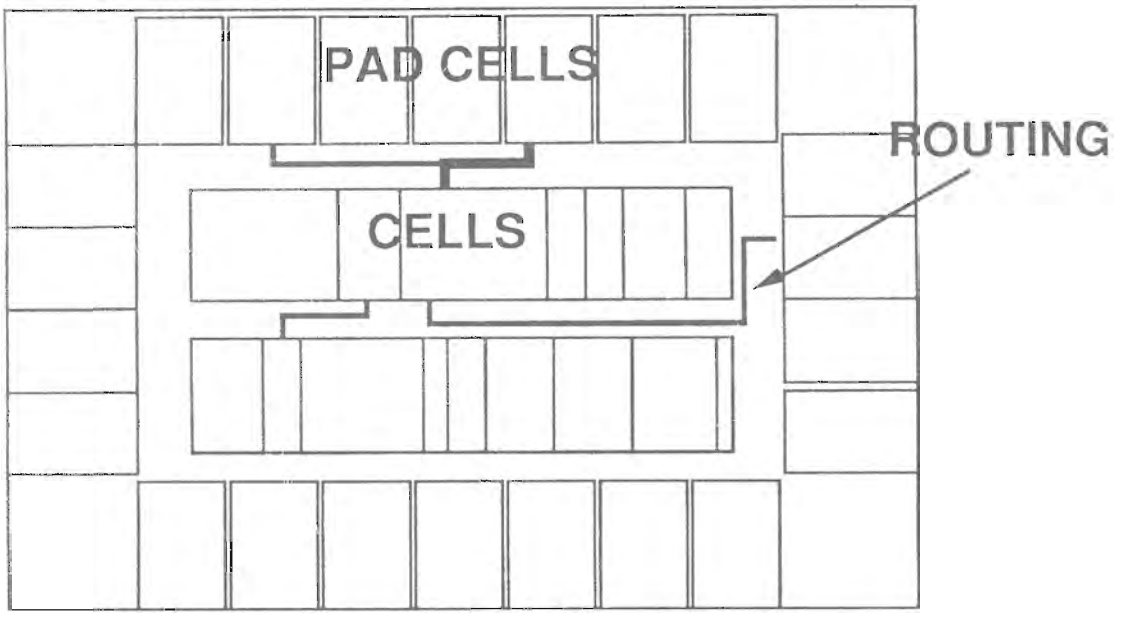
An example of a full-custom chip is a memory chip. Here performance and complexity are essential and the production run is likely to be many million parts. Much less complex parts are also likely to be full-custom if the highest speeds are required.

2.2. Semi-custom

There are a number of design tools that automate the design process by using pre-designed and characterised blocks. In some tools the blocks have a high level of complexity e.g. a microprocessor core, whilst in others the individual blocks are at the level of basic logic gates but the tool has intelligence to

interconnect these gates to make more complex structures. An example of the latter type of tool is the ES2/Cadence SOLO 1400 tool.

The final chip consists of logic blocks with routing pathways between. The size of the routing channels is automatically adjusted in accordance with the requirements of the auto-router. A typical arrangement of a standard cell chip is shown below.



Advantages:

- Higher speeds and higher complexity are available than with gate arrays.
- Pre-characterised proven cells are used in the design, so design confidence is high.
- A high degree of design automation is available, so limited extra skills are required and the design time can be short - perhaps as short as a week if the design is not very complicated.

Disadvantages

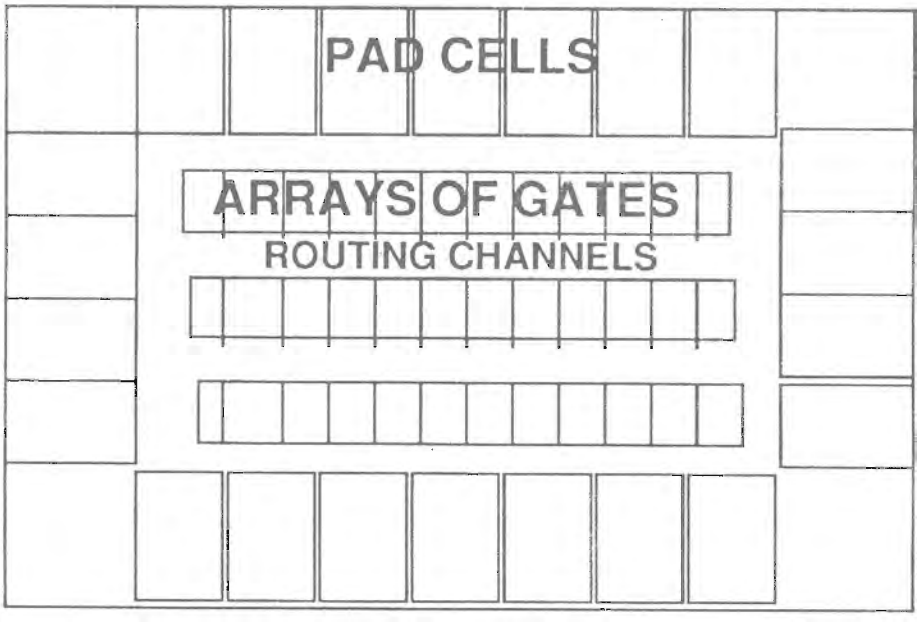
- A complete set of special masks is required for production, so the mask costs are as high as for a full-custom design.
- The prototype turn-round times are likely to be at least 9 weeks (four to five weeks is possible if you have a good relationship with the fabricator.
- The CAD tools are still relatively expensive and you require workstation computing power.

Many moderate volume parts are designed as semi-custom standard cell parts. Major electronics manufacturers have their own standard-cell libraries that they have developed in-house for their needs, but standard cell libraries are available for purchase, generally at a high cost.

2.3. Gate Array

This is the design technique favoured by smaller electronics companies. The cost of producing ASICs of moderate complexity can be very low by this technique. Major fabricators such as Hitachi, Texas Instruments or LSI Logic produce standard chips in large volume that consist of a sea of gates. They also supply software for inputting a design of interconnections between gates in a graphical form. This software will include a simulator and generally be capable of running on a reasonably powerful PC. When the designer is satisfied with the results of his simulation he ships the disk containing the design to the fabricator. The fabricator then adds the interconnection layers to the standard array so that the array implements the special logic required by the customer.

A typical simple gate array will look like that shown below



The more advanced arrays do not have separate routing channels but run the routing over the gates. Arrays of over 250,000 gates are available and arrays as small as 200 gates are also available.

Advantages:

- Start-up costs are relatively low.
- Rapid design & prototyping are possible.
- There is a high probability of success.
- Few new skills are required.

Disadvantages:

- The available complexity is limited.
- The device speed is lower than for full-custom or standard cell design.

The silicon area used is relatively high and the cost per part for high volume is higher than with full-custom or semi-custom techniques.

Gate arrays have two or more routing layers but it is still not possible to use a very high percentage of the available gates in practice. In a sea-of-gates design only about 60% of the gates may be usable because of routing problems.

Some companies will provide ten or so prototype gate arrays for low-complexity designs for around $500 and the turn-round time for fabrication should be better than two weeks. The design time should only be a few days unless it is a complex design once the learning curve for the software has been overcome.

2.4. Field-Programmable Gate Array

These are the most recent ASIC introduction. There are two major varieties, arrays that are fuse-programmable and arrays that are programmable via a non-volatile memory and are therefore re-programmable. In either case there is a special programming board that plugs into a PC and the programming is instant once the design is completed.

In one manufacturer's offering the design system is priced at $10,000 with the 2,000 gate parts priced at $63.50 each. The unit cost is therefore high and would not be attractive for volume production. The speed is 70MHz.

At a much lower level of complexity programmable logic devices are available with up to 258 product terms.

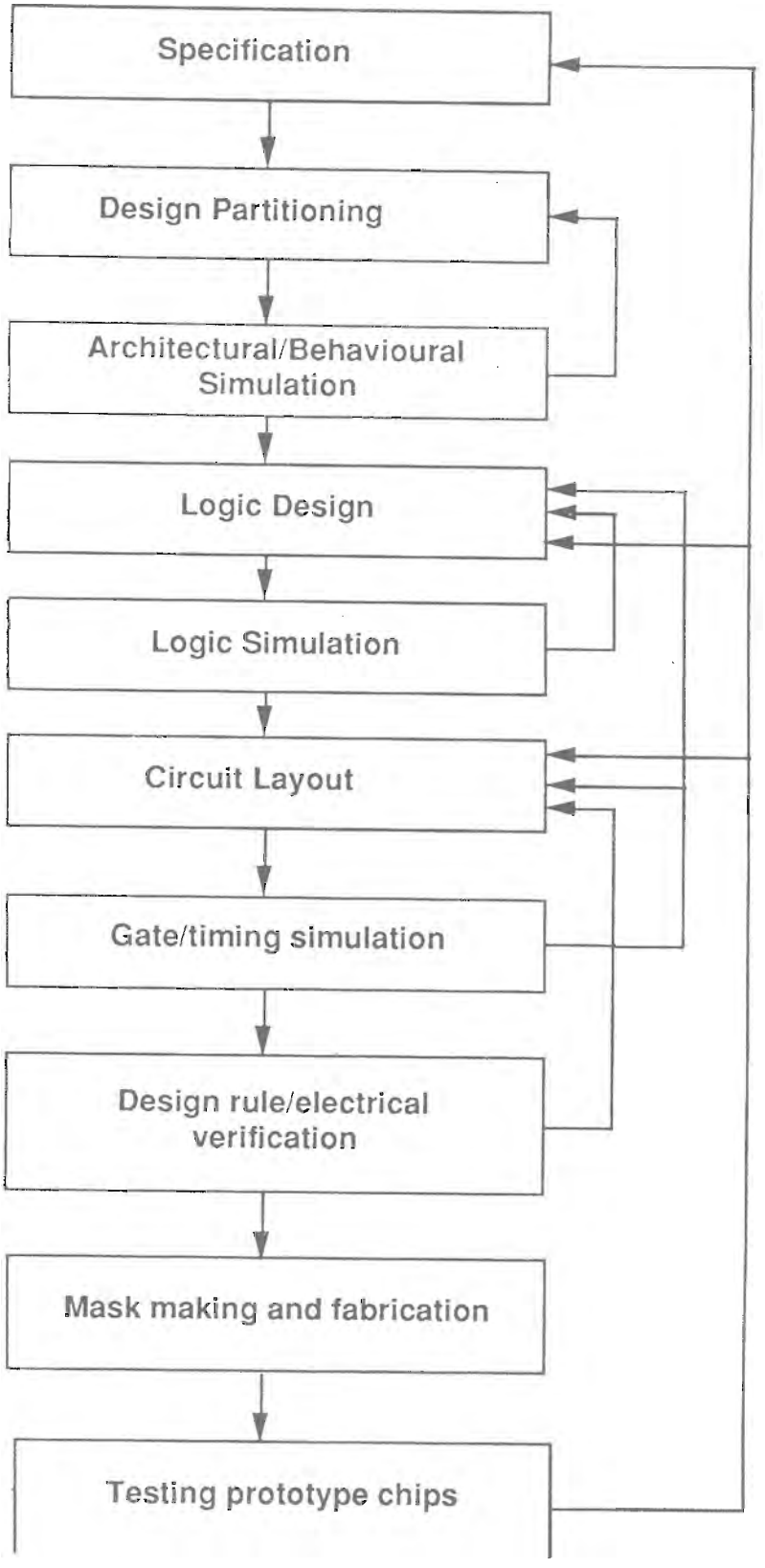
2.5. Analogue Design

Most ASIC design software has been aimed at digital design but some cell libraries contain a few analogue blocks in addition to digital blocks and there are some gate arrays that contain analogue elements. With the growth in popularity of BiCMOS technology this trend is likely to accelerate.

There are still very few support tools for custom analogue VLSI design, but Cadence, for example, now have such a tool as part of their design system.

3. The design cycle

The standard design cycle for custom ASIC design is:



There are CAD tools to support almost every step in this process and in a good CAD system these tools are well integrated so that there is a minimum of data conversion required.

4. Typical design tools and environments

At the specification level of design a hardware design and desription language (HDDL) is required. Such a language is an extension of a programming language and the one that is probably going to become the standard is VHDL. This has Ada as the underlying language structure. Tools to support VHDL that include simulation are just becoming available but they generally only support a subset of the language. The language is rich but does not support design down to the layout level. VHDL originates from a U.S. Department of Defence initiative. An alternative originating from the U.K. Ministry of Defence is ELLA, which is sold as a complete system including simulator and a support environment. In either case the tools require a workstation such as a SUN, Hewlett Packard or DEC machine.

A number of tools support schematic capture of logic design. Some, such as ORCAD can be used as a capture tool for a number of logic simulators. Complete design systems such as those sold by Mentor and by Cadence have schematic capture and logic simulation closely combined, together with graphical techniques for specifying and viewing the results of the simulation. A well-known logic simulator that has no built-in graphical front end is HiLo. The simulators themselves will run on a variety of workstation and minicomputer platforms with the graphical pre- and post-processing running on major workstations. With the introduction of X-windows interfaces to the graphical parts X-terminals of any description can be used, although not all vendors have introduced licensing systems to allow this.

Circuit layout requires good quality graphics with colour to represent the different mask layers for full-custom design. If the design system attempts checks of the layout against the set of fabrication design rules as the design progresses considerable computing power is also required. The latest systems require 16 MByte of RAM and systems of about 20 SpecMarks for reasonable performance.

There are a number of other checking tools available on integrated design systems. These check electrical connectivity, the correspondence between the logic design and the layout, the testability of the circuit and they will also generate test vectors.

5. The relationship between design, tools and testing for VLSI

It is very easy to design an integrated circuit that it is impossible to test. There may be a million transistors on the chip but only 128 pins to access the system. It is therefore vital to consider the testing problem at the beginning of the design process. Often extra circuitry is added just for testing. This

approach is called built-in self test (BIST). Programs are available both to evaluate the degree of testability of a chip and to assist in designing BIST structures.

The eventual test vectors are generally too complex for manual production so the logic simulator has extra modules that will produce test patterns. These will be directly transferred by disk or network to an automatic tester.

6. Mathematical modelling in VLSI design

All the simulation steps require mathematical modelling and the testability analysis also requires modelling.

A current area of research is in formal verification of designs. If a design is to be used in a safety-critical application then the design needs a more formal verification process than simulation can provide. Attempts have been made to formally prove the correctness of a design mathematically. This is difficult and in cases where it has been done the layout has not been formally proven to implement the design.

Another algorithm problem arises from the checking of the correctness of the layout against a set of design rules. As the minimum feature size of fabrication processes decreases the number and complexity of the fabrication rules increases significantly. This makes it difficult to code the rules and makes the checking operation slow.

Analogue deign still has very few aids and this is an area wide open for modelling.

7. Bibliography

Dillinger, T., *VLSi Engineering*, Prentice-Hall (1988).

Fabricius, E.D., *Introduction to VLSI Design*, McGraw-Hill (1990).

Geiger, R.L., Allen, P.E. and Strader, N.R., *VLSI Design Techniques for Analogue and Digital Circuits*, McGraw-Hill (1990).

Mukherjee, A., *Introduction to NMOS and CMOS VLSI Sytems Design*, Prentice-Hall (1986).

Reinhard, D.K., *Introduction to Integrated Circuit Engineering*, Houghton-Mifflin (1987).

Weste, N.H.E. and Eshragian, K., *Principles of CMOS VLSI Design*, Addison-Wesley (1985).

VERY HIGH AND ULTRA HIGH SPEED VLSI TECHNOLOGIES

KAMRAN ESHRAGHIAN

Centre for Gallium Arsenide VLSI Technology
The University of Adelaide
South Australia

1 Introduction

The rapidly emerging area of High Speed and Ultra High Speed processing that now underpins the transformation of much of the ideas into working systems necessitates for evolutionary changes in both the technology and the strategy that would facilitate the ability for physical mapping of such systems . The systems that are mostly affected, and indeed in the next decade or so would require an ever increasing processing power include real time signal processors and image processors, computer vision, telecommunications, biomedical systems and personal super computers, just to name a few.

Since the invention of the transistor in 1947, and the development of the very first integrated circuit at the beginning of 1960, there has been four generation of ICs. Now, we are beginning to witness the emergence of the fifth generation of ICs that are characterized by complexities in excess of 1.8 million devices on a single chip. Over the past several years, CMOS technology has become the dominant fabrication process for relatively high performance and cost effective VLSI circuits and systems. For example, today's microprocessors are able to handle some 80 - 100 million operations per second, but the circuitry that manages communications with other processors and in particular with the memory, is just too slow to keep up with this data rate. The speed/power limitations that are becoming apparent have brought about the need for development of other technologies such as Gallium Arsenide and BiCMOS.

The compound Gallium Arsenide is rather a different material to silicon. It was first discovered in 1926. However its potential as a high speed semiconductor was not realized until 1960's. For the same power consumption, GaAs is about half an order of magnitude faster than emitter-coupled logic (ECL), the fastest of the silicon family. The speed advantage is primarily the result of Gallium Arsenide's high electron mobility. Other advantages of Gallium Arsenide over silicon include its higher temperature tolerance, radiation hardness and optoelectronic properties. The last point permits efficient integration of electronic and optical devices, on the GaAs IC, and is somewhat critical for Ultra High Speed systems of the future. Gallium Arsenide has a direct bandgap, whereas silicon has an indirect bandgap. This means that energy level transitions in silicon, as illustrated in Figure 1, take place with momentum change. Whereas energy level transition, in GaAs, can take place with momentum conservation, allowing photon emission. Thus, GaAs is both a good emitter, as well as a good receiver of light. This creates new opportunities towards on-chip integrated optical communications since interconnects in Very High and Ultra High speed systems have significant influence upon the performance of the system. The bandwidth requirements associated with a very fast pulse requires the designer to have better appreciation of the imposed constraints. Although at present the capacitance and resistance of interconnect including those of contacts are a practical limitation on scaling of minimum geometry devices, the resistivity of the substrate also provides further constraints on the propagation of signal.

The propagation of a signal in both the semi conducting material such as silicon substrate and semi insulating material such as Gallium Arsenide can be described through the two modes of operation:

•Slow-wave mode;
•Dielectric quasi-TEM mode.

In Slow-wave mode the propagation of the signal is through the bulk of the semiconductor. In this mode the substrate such as silicon, appears as a lossy medium which subsequently attenuates the signal. Alternatively in the dielectric quasi-TEM mode the transmission of a signal occurs through the bulk of the substrate with very small propagation losses which is typical of Gallium Arsenide. These limitations could be overcome through the incorporation of optical interconnects. Thus, Gallium Arsenide is emerging as a possible contender for Ultra High speed systems.

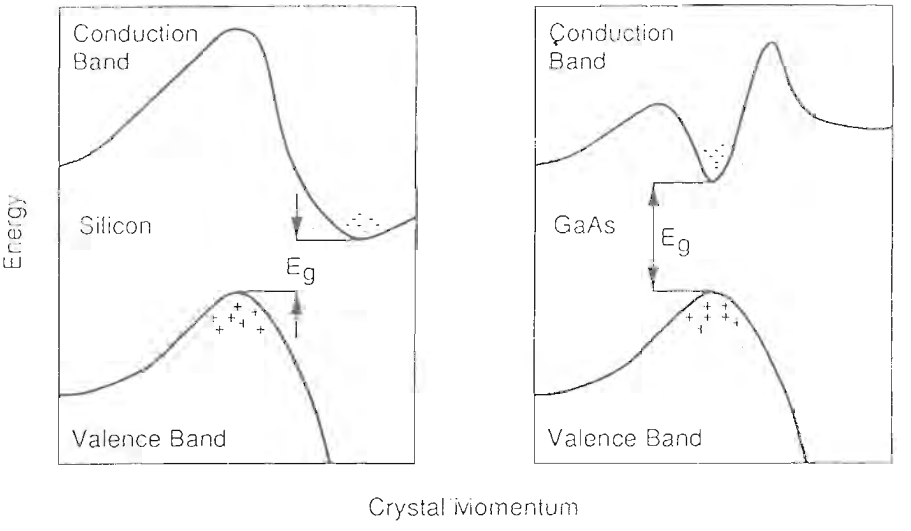


Figure 1. Energy Band Structure of GaAs vs. Silicon

Optical interconnect can either take the form of optical fibres or thin film paths (e.g. aluminium oxide) integrated on the chip carrier as shown in Figure 2. Another form is the free-space interconnect where focussed or unfocused light is broadcast over an area, whole or system. This can be utilized to supply a high speed global clock to a VLSI circuit without the usual routing complexity. Here an opaque masking layer would be used to shield parts of the chip from unwanted light.

In the late 1980's US manufacturers of digital Gallium Arsenide circuitry bowed to the reality of a predominantly silicon world. They began to introduce GaAs integrated support circuits which had neither quirky power supply requirements nor idiosyncratic input/output voltage levels. This support circuitry provided the basis for GaAs chips that looked and acted like silicon chips but were faster and dissipated some 50% to 75% less power than their silicon ECL counterpart. This turning point brought about a significant influence upon the future trend for implementation of Ultra High Speed, high performance VLSI systems.

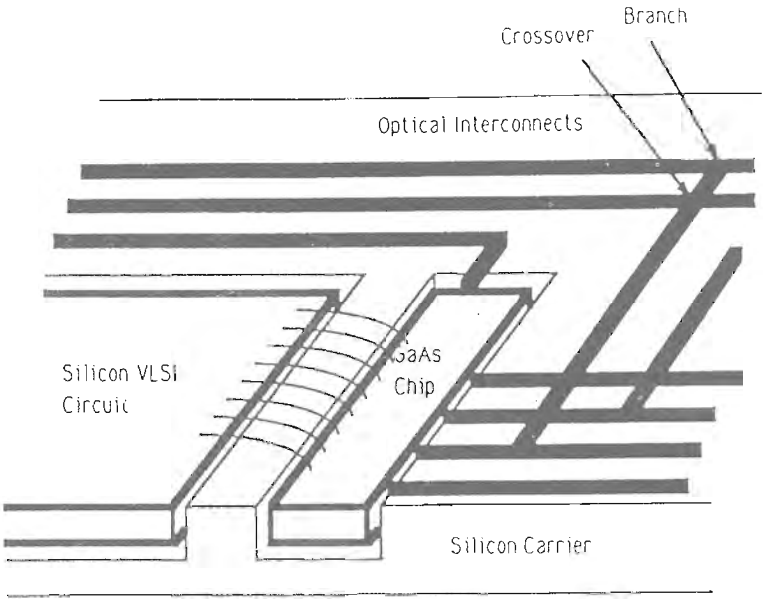


Figure 2. Optical Interconnect

Paralleling the developments in CMOS technology, some evolving technological changes are also beginning to take place in the area of BiCMOS where some of the advantages of the Bipolar transistor are being utilized as CMOS speed enhancer. Main advantages of Bipolar technology over MOS, are greater current drive per unit area, low delay sensitivity to load variation (make excellent line drivers, and decoders for memory elements and fast on-chip cache), better device matching, low sensitivity to process variation and linear performance (sense amplifiers and current mirrors). This means that GaAs technology may be utilized for front-end processor sections of high-speed single stream processors for digital data (e.g. 1-2 gigasamples/second), usually generated by way of very wideband image sensor, detectors or the like. This fast data stream can subsequently be subdivided into lower rate parallel streams suitable for processing in silicon CMOS/BiCMOS subsystems at lower frequencies. Figure 3 illustrates such a concept and highlights the critical paths for communications.

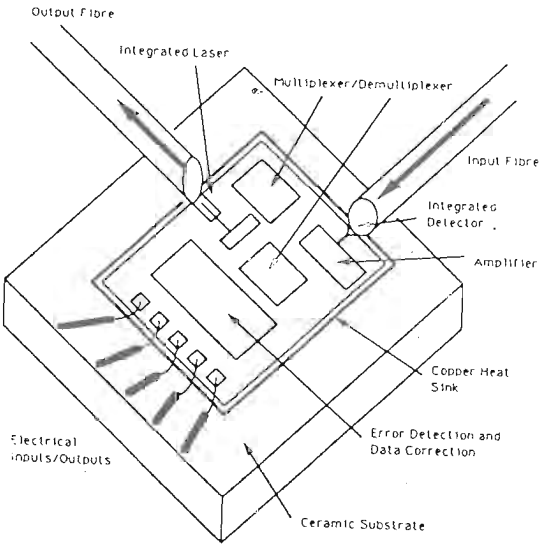


Figure 3. Partitioning of High Speed Systems

The necessity for high clock rates in the "front end" processors is compounded since it is very likely that 10-20 microcycles of the processor may be required to preprocess each incoming data sample. This means an input data rate of 10^8 bytes/second might demand a system clock rate of 2 GHz. By mixing GaAs and Silicon CMOS/BiCMOS technologies, it becomes possible to exploit high system clock rates in a number of systems, including high bandwidth signal and image processors, memories, microprocessors, telecommunication

2 Performance Requirements

The Integrated Circuit performance without further qualification is somewhat subjective. However for the purpose of comparison it is possible to quantify it in terms of the three parameters, power dissipation, functional throughput rate, and communications limits. In computationally intensive structures usually one has to address the best approach that can be pursued to partition a given system. For example, the question that may arise in this process is the choice as to whether one creates a very high speed single channel or to take the alternative approach of partitioning the system into multiple parallel channels clocked at a lower frequency. The technology suitable for Very High and Ultra High Speed VLSI systems needs to satisfy the following criteria:

- Very low propagation delay possibly less than 100 ps/gate;
- Low gate dissipation in the order of 100μW/gate;
- Very low dynamic switching energy, i.e. less than 0.1 picojoule;
- Very high level of integration, greater than 50,000 gates;
- High process yield.

Figure 4 shows the level of integration in terms of the switching energy for a number of technologies for a typical chip dissipation of 2 Watts.

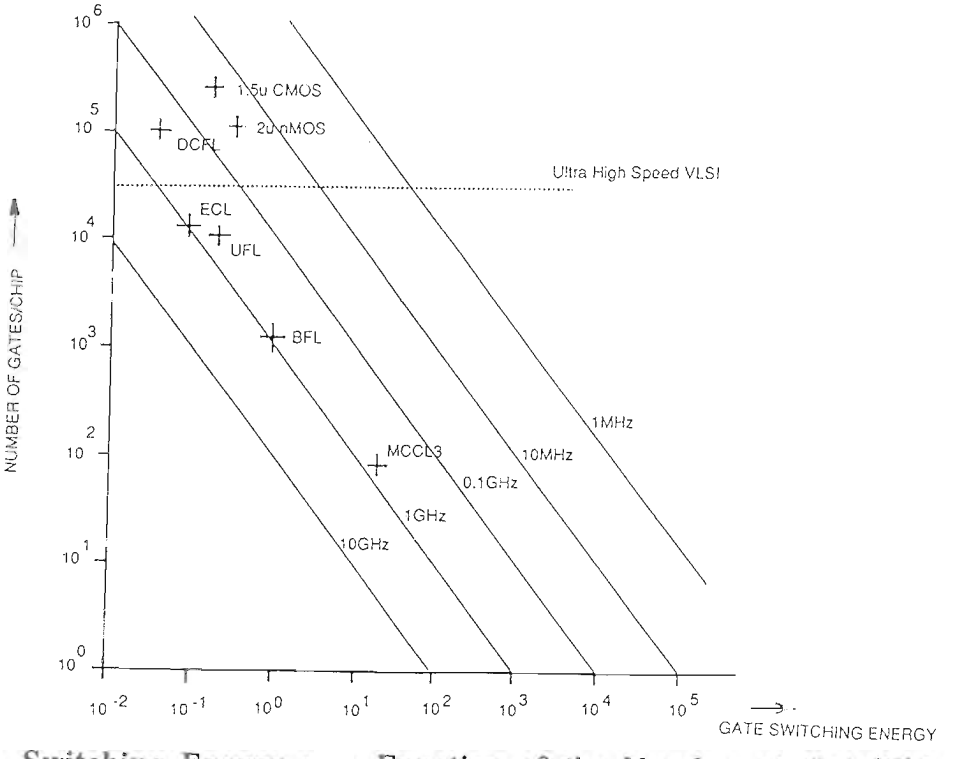


Figure 4. Switching Energy as a Function of the Number of Gate/Chip

2.1 BiCMOS Technology Adaptation

For high performance VLSI systems, it is possible to take the advantage of Bipolar transistors to enhance the performance of memory components and I/O drivers. In a Bipolar transistor the Base-Emitter V_{BE} varies linearly with temperature T (which can be kept reasonably constant over the chip area), and logarithmically with base width W_B, base doping N_A, electron mobility μ_n, and collector current I_c. This means we can achieve ΔV_{BE} of a few millivolts over a circuit/chip area. For high-speed logic, there are a number of design trade-offs. The design compromises centre on the fact that to maintain emitter injection efficiency, the base doping level N_A must be limited to a fraction of a percent of the emitter doping. The active carrier is the minority carrier in the base (i.e. electrons in an NPN transistor). It is in fact up to the rather modest density N_A of holes in the base to establish electrical connection from the control electrode to the active region with small lateral base resistance R_B. Performance improves strongly as electron transit time through the base is reduced by lowering the basewidth W_B. However lowering of the basewidth results in high base resistance, unless the emitter strip width W_E is made very small. Furthermore if the base width is lowered too far, i.e. $W_B < 1000Å$, with the reasonably modest base doping N_A, the numbers of doping atoms in a cube of dimension W_B on each side becomes so small (in the order of 100) that simply the Gaussian statistical variations in this number can lead to Emitter-Collector punchthrough in a statistically significant number of transistors.

Collector resistance together with Collector-Substrate capacitance also have a dominant effect on the propagation delay of a BiCMOS gate. To overcome such performance related issues process enhancement is necessary. It should be noted that BiCMOS flow could add up to five levels to the CMOS core.

- Buried subcollector (high performance NPN device requires heavily-doped subcollector)
- Topside deep n^+ collector contact
- p^- base implant
- Emitter pattern
- Bipolar trench isolation (provides high density with low capacitance).

If a BiCMOS is retrofitted into an existing CMOS process as shown in Figure 5, in the absence of process enhancements, then the above design issues need to be carefully considered during the design phase and included as part of the overall design methodology.

2.2 CMOS Based BiCMOS Process

Looking at the technology base it is readily recognizable that, a CMOS-based BiCMOS process compromises the performance of the NPN transistor, while Bipolar-based BiCMOS process compromises CMOS devices. Although a generic BiCMOS process is uncompromised, the BiCMOS for VLSI applications tend to be CMOS dominated as the result of the need for highest possible CMOS performance and density. Therefore the option that one should pursue needs to be based upon an existing, and readily accessible n-well CMOS process that has been modified by inclusion of a p^- base implanted mask to form the p-base of the NPN transistor in the n-well. By inclusion of BiCMOS in critical paths of VLSI circuits, a speed enhancement by a factor of 2 and possibly 3 becomes possible.

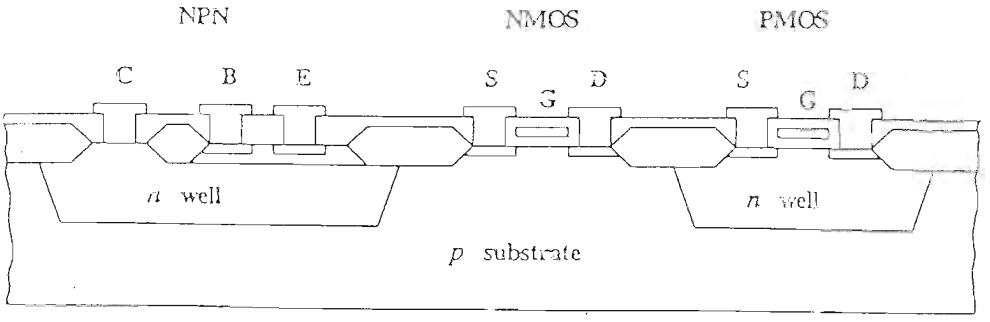


Figure 5. Structure for BiCMOS Retrofitted to an Existing CMOS Process

3.0 Summary

In order to have a better appreciation of the technology base for Very High and Ultra High speed technologies Table 1 is provided.

Table 2.3 Comparison Between CMOS, BiPolar and GaAs Technologies

CMOS	Bipolar	GaAs
• Low dissipation	• High dissipation	• Medium dissipation
• High I/P impedance	• Low I/P impedance	• High I/P impedance
- low drive current	- high drive current	- below ØB
• High Noise Margin	• Medium Noise Margin	• Low Noise Margin
• Medium speed	• High speed	• Very High speed
-high voltage swi	-low voltage swing	-low voltage swing
• High packing density	• Low packing density	• High packing density
• High delay sensitivity	• Low delay sensitivity	• High delay sensitivity
to load - Fanout	to load - Fanout	to Fanin and Fanout
• Low output drive	• High output drive	• Low output drive
• gm Vin	• gm eVin	• gm Vin
• Bidirectional	• Unidirectional	• Bidirectional possible
• Ideal switching device	• Not ideal switching device	• Reasonable switching device
• Medium ft	• High ft at low current	• Very high ft
• Indirect gap	• Indirect gap	• Direct gap
		-good light emitter
• Mask level 12 to 16	• Mask level 12 to 20	• Mask level 6 to 10

The table highlights a number of factors including dissipation, clocking frequency, integration density and cost that influence the design base.

Design Methodology and Layout Style for Very High Speed Circuits and Subsystems

Kamran Eshraghian

Centre for Gallium Arsenide VLSI Technology
The University of Adelaide
South Australia

1 Introduction

The VLSI era has brought about with it the concept of Design Methodology which has for some years now facilitated rapid development of complex integrated circuits. With recent developments of automatic Computer Aided Design tools it appears that such an approach is becoming less important than in the past. However with the emergence of Very High speed VLSI technologies where performance is an important aspect of the design criteria, once again, design methodology is beginning to feature as an important part of the chip design cycle. Design methodology in general can assist to a large degree towards creation of an effective design environment for designers whereby it becomes possible not only to save chip area and time during the manual design cycle, but more importantly to provide a better insight to automatic tool designers to create simple and efficient translators and compilers to satisfy the needs of a particular architecture. Thus adequate understanding of *Methodology* permits the designers to make correct selection of CAD tools suitable for a particular class or classes of architectures. Design methodology usually is described in terms of a class of circuits needed to created either directly or through partitioning a given architecture.

The classical design methods for integrated circuits can be grouped as:

- Logic Arrays
- Gate-Arrays
- Sea-of-Gates
- Standard Cells
- Semi-Custom
- Full-Custom
- Macro Cells.

The Macrocell concept being a combination of Semi-Custom and Custom design, most suited for complex integrated circuits having optimal performance, has gained rapid acceptance. However, the need for global CAD tools places considerable constraint on its effective implementation. Therefore the full-Custom design approach still is the most viable alternative for the design of high performance VLSI chips.

2 Design Domains

The design of an integrated circuit can be described in terms of *Behavior*, *Structural* and *Physical* domains. Fundamentally, Behavior Domain describes to the designer as to the manner in which a design should function. This is usually presented in terms of an algorithm. The Structural Domain highlights the necessary interconnection of modules needed to achieve the required behavior. This level of abstarction inverably cab be represented by the *Logical Architecture*. Finally what actually is needed physically to achieve the required structure and hence the behavior is embedded within the scope of Physical Domain. This level corresponds to the *Physical Architecture* commonly referred to as the *Floor Plan*. Usually realization of a satisfactory Floor Plan may require several iterations through the design domains. It is at this level of the design cycle where the communications strategy together with power distribution are highlighted.

2.1 Structured Design and Floor Planning

The structured design begins with the concept of hierarchy, in which complex functions are divided into less complex and manageable subfunctions. For Very High speed systems, system timing has a dramatic influence upon the composition of the *Floor Plan*. This means algorithm transformation and decomposition process must be tightly coupled to both local and global communication strategies. The steps involved with algorithm transformation result firstly in realization of a *Logical Architecture* from which it becomes possible to derive the *Physical Architecture*. In composing the Logical Architecture from the behavioral description, there is a degree of transparency associated with both clocking and data transfer strategies, and for that matter power distributions is not fully described at this level. This means obvious problems with signal skew and noise behavior may not be highlight, and hence performance could be significantly degraded if adequate measures are not incorporated at this level of abstraction. This means design methodology and a layout style for given classes of logic and technology become an important part of the design environment.

2.2 Logic for High Speed Processing

There are two main approaches towards logic design using the first generation GaAs MESFETs as the base technology:

- Normally-on logic
- Normally -off logic.

The normally-on logic utilize depletion mode MESFETs which are 'ON' devices. When used as switching elements they are required to be turned OFF. Thus, a number of circuit techniques have been developed over the years to facilitate logic turn-off. The approaches in this class of logic include:

- Unbuffered FET Logic (UFL);
- Buffered FET Logic (BFL);
- DMESFET Schottky Diode FET Logic (SDFL);
- Capacitor-Coupled FET Logic (CCFL);
- Capacitor-diode FET Logic (CDFL).

The normally-off logic use enhancement mode MESFETs as the switching element. Although several approaches have emerged during the last few years the following structures have shown the most promise:

- Direct-Coupled FET Logic (DCFL)
- Buffered DCFL
- Source-Follower FET Logic (SFFL)
- Pseudo-Current-Mode Logic (PCML).

2.2.1 Normally-0n Logic Gates

Depletion mode devices are basic switching elements for this class of circuits. Since D type MESFETs are ON devices then to facilitate turn-off, a negative voltage is needed at the gate. This means two supply buses together with a level shifting network are necessary for proper circuit operation. For example, the Unbuffered FET logic (UFL) uses depletion mode MESFETs together with Schottky diodes to perform a logic function. The basic structure for an Unbuffered FET logic together with the transfer characteristics are shown in Figure 1. As can be seen there are two basic functional blocks:

- Input logic
- Voltage shifter.

To ensure that both the input and output voltage levels of the logic gates are compatible, level shifting is implemented after the input logic circuitry stage. Schottky diodes D1 and D2 together with the pull-down DMESFET, T3, provides the mechanism for this function. Logic functions can simply be realized by modifying the input logic circuitry.

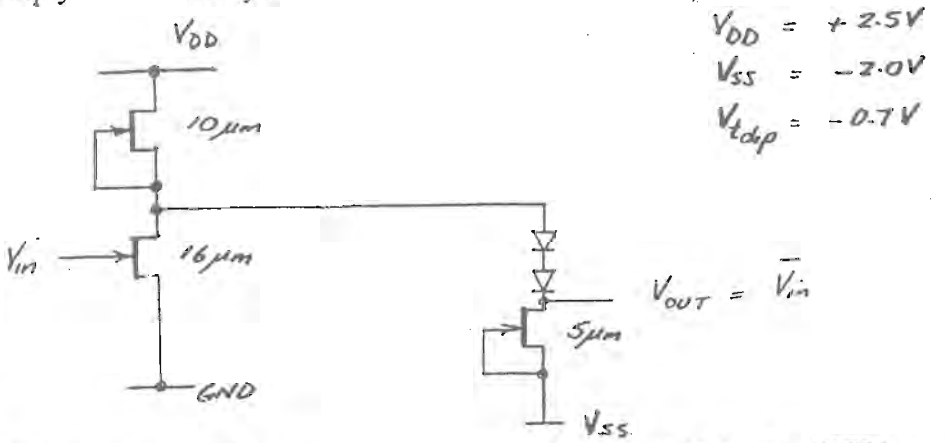


Figure 1. Basic structure for Unbuffered FET logic (UFL)

During an input transition i.e. logic '0' to logic '1' change, the D type MESFETs, T1 and T2, charge and discharge the load capacitance C_L. The Schottky diode associated with the gate of the next stage limits the output swing to approximately 0.8V.

Sizing of the transistors in the gate is influenced by the subthreshold current. For wide transistors, the leakage around MESFETs could shunt sufficient load current which reduces the magnitude of the V_o 'High' level. This in turn will influence the *Noise Margin* associated with the gate.

As can be observed, during the entire period, there is a static current I_D that flows through the load device T1, the level shift diodes D1 and D2, and the pull-down transistor T3. The UFL gate exhibits a good noise margin, and furthermore, gate delays in the order of 100 *pS* can be expected. However the main limitation of UFL is it's sensitivity to load variations. To overcome the influence of load variation that could be considerable, a source follower is included as part of the output stage. The modified structure refereed to as a Buffered FET Logic (BFL) gate is shown in Figure 2.

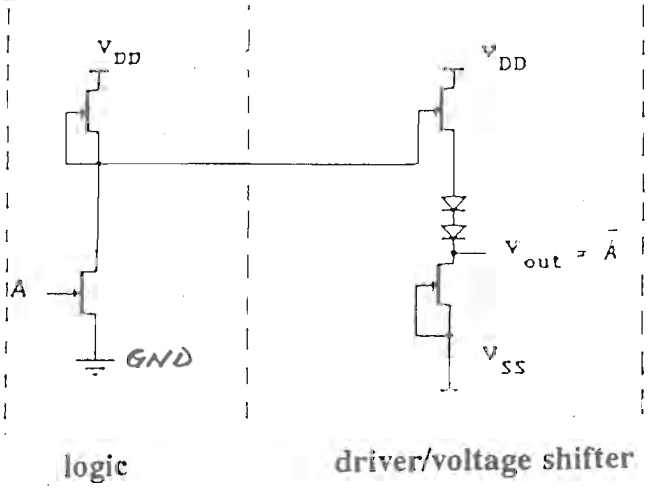


Figure 2 Basic Gate Structure for Buffered FET Logic

Typical delay of a BFL gate is in the order of 50 *pS* . Since the saturation drain-to-source current for a depletion type MESFET decreases with increasing temperature, the switching point of a BFL gate characteristics varies typically within the range of +0.3 to +0.4 mV/oC. The noise margin is determined by the Schottky gate turn-on voltage which is typically in the order of 0.7V to 0.8V. A limitation that is a characteristic of this class of logic is the need for two power supplies. Thus additional complexity in routing of power buses can be expected. Gate dissipation for UFL is in the order of 1 - 2 mW, and approximately 5 - 10 mW for BFL. Furthermore due to the complexity of the basic gate structure and dissipation, this class of logic is unsuitable for Very Large Scale Integration.

2.2.2 Normally OFF Logic

In this class of logic both the depletion mode and the enhancement mode transistors are used. This class of logic resemble closely the structure of nMOS logic in which the present generation of VLSI designers are most familiar with. Because of the structure of the gate, ratio rule which establishes the device size applies. The sizing determines the performance of the basic gate both in terms of power-speed product and Noise Margin. The enhancement mode FET is utilized as the switching device, while the depletion type device acts as the load. Figure 3 illustrates the basic structure for a DCFL inverter together with the kind of Noise Margins that can be expected from such circuits. From the figure it is evident that there is no need for level shifting circuitry as was the case for Normally On logic.Also a single power supply bus is necessary. The allowable output voltage is limited by the barrier height of the Schottky gate diode.This means only small voltage swing is possible from DCFL circuits. Thus relatively a small noise margin is characteristic of this class of logic. Changes in Pull-Up to Pull-Down ratio or the threshold voltage variation for the DCFL inverter results in a shift in the Vo vs. Vin characteristics which further degrades the noise margin. Therefore, determination of the width ratios for the pull-up and pull-down devices is an important aspect of the optimization process.

DCFL gate dissipates typically about 80 - 100μW with the associated delay of about 10 - 50 ps. This is considerably less than the "Normally On Logic" families. Thus, the logic appears as a suitable contender for Very High Speed VLSI systems.

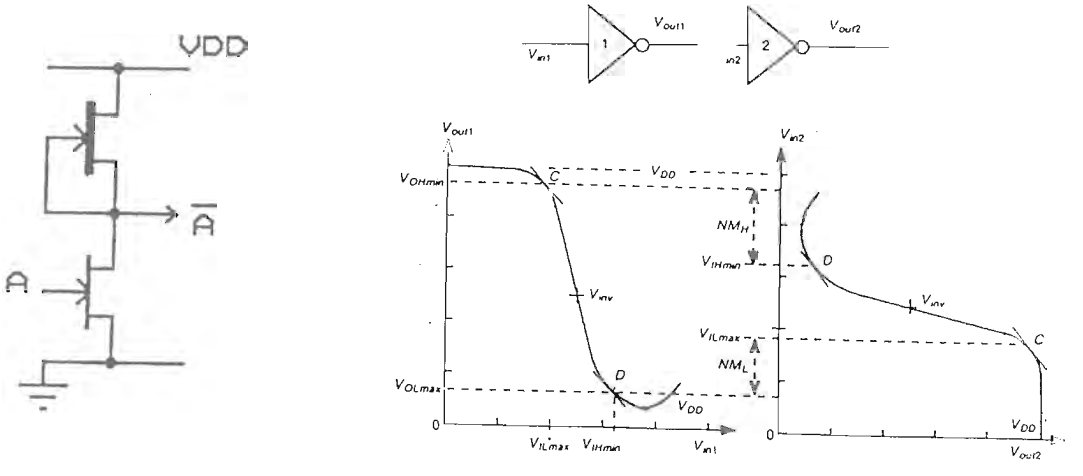


Figure 3 Direct Coupled FET Logic (DCFL)

DCFL circuits have weak load drive capability. This implies that the delay associated with a gate increases with an increase in both the Fan-out as well as the interconnect line lengths. Introduction of supper buffer as shown in Figure 4 can alleviate much of the problem at the expense of extra area. Usually the basic DCFL gate is used for light load conditions, while the supper buffers are used where larger loads are to be driven.

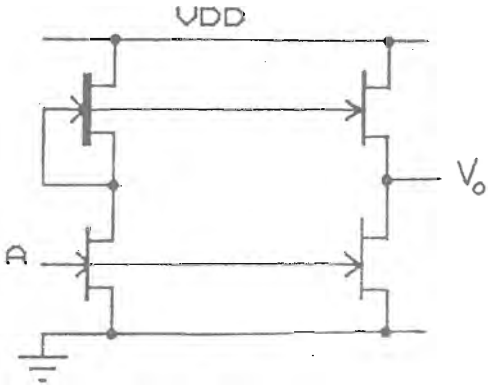


Figure 4 Inverter with Supper Buffer

An alternative arrangement providing improved noise margins are the Source Follower DCFL (SDCFL), and Source-Follower FET logic(SFFL) as shown in Figure 5. The two logic family have both a power dissipation and switching delay that are comparable with DCFL family, however with larger fan-out and noise margins. The improved noise margin in SDCFL is brought about as the result of the pull-up transistor (enhancement mode) being able to be turned-off, thus permitting the source follower output to pull-down all the way towards zero voltage.

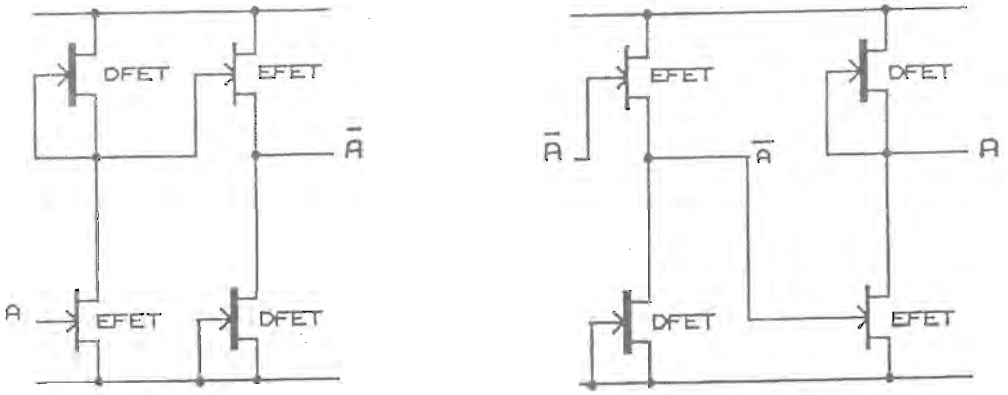


Figure 5 Structures for SDCFL and SFFL

An interesting aspect of the three classes of logic namely, DCFL, SDCFL and SFFL is that they can be merged together within the frame work of the design methodology. This approach is shown in Figure 6.

2.3 Design Methodology for DCFL, SDCFL and SCFL Classes of Logic

Communication paths between logic blocks including, both signal and power buses have significant influence upon the performance of Very high speed VLSI structures. The designers of high performance digital systems need to appreciate the bandwidth requirements of a high speed signal and the related constraints such as crosstalk, reflection and attenuation and distortion when a communication path acts as a transmission line. Fast transitions also bring about noise on the power bus. Thus layout methodology has a significant influence upon performance.

The placement of V_{DD} and GND lines must be placed such as to reduce their self inductance, and hence their susceptibility to current transients. from the results of Coplanar Strip Line and

Coplanar Waveguide models, the inductance of the power buses is reduced by a factor of two to three when they are placed in the proximity of one another.

The new approach referred to as "***Ring Notation***" exploits this advantage and enables the designer to layout the skeleton of a circuit rapidly, paying particular attention to power and signal buses between adjacent circuitry.

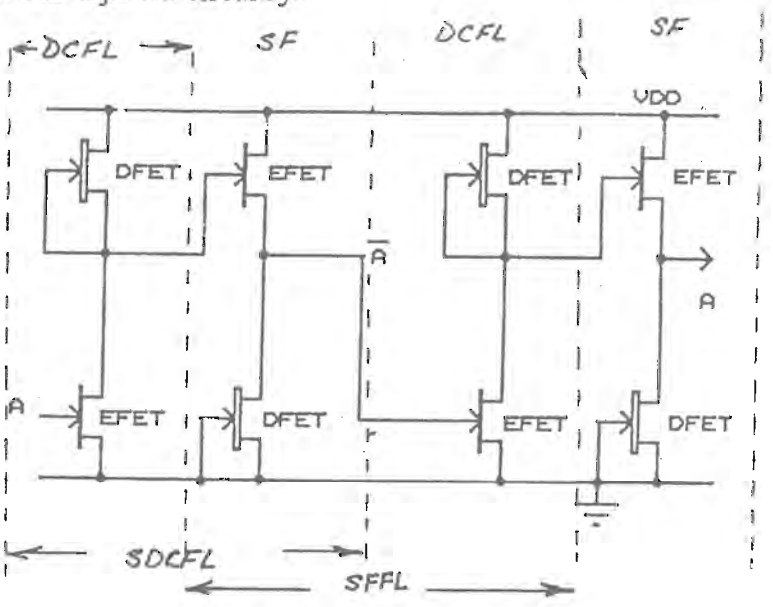


Figure 6. Merged Structure

The layout approach for a basic DCFL inverter using this notation is shown in Figure 7. As can be observed translation from the *Ring Notation* to its equivalent symbolic form or alternatively to mask layout is straight forward.

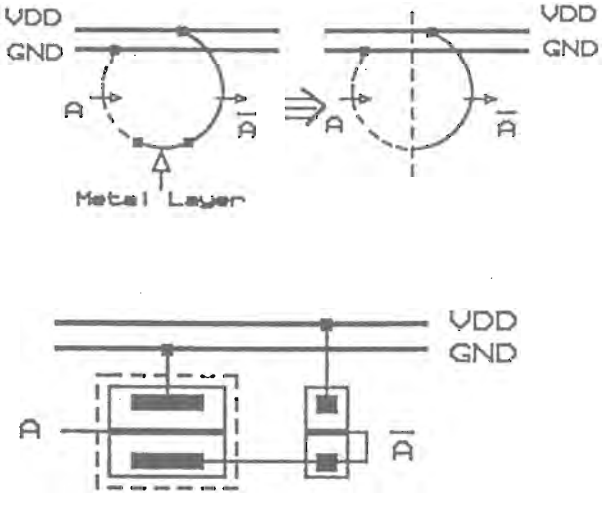


Figure 7. Methodology for Layout of DCFL Inverter

Since in this technology we use NOR gates only then representation of NOR gates can be simplified by eliminating the parallel branches of the inputs. The approach is illustrated in Figure 8.

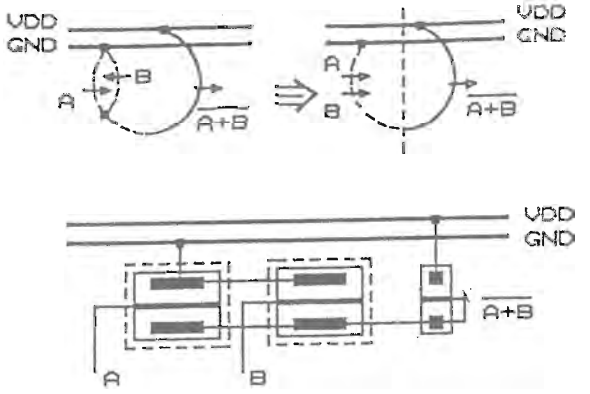


Figure 8. DCFL NOR Gate using Ring Notation

By the way of an example Figure 9 illustrates the overall strategy used for layout an XOR gate using the merged approach for the three classes of logic.

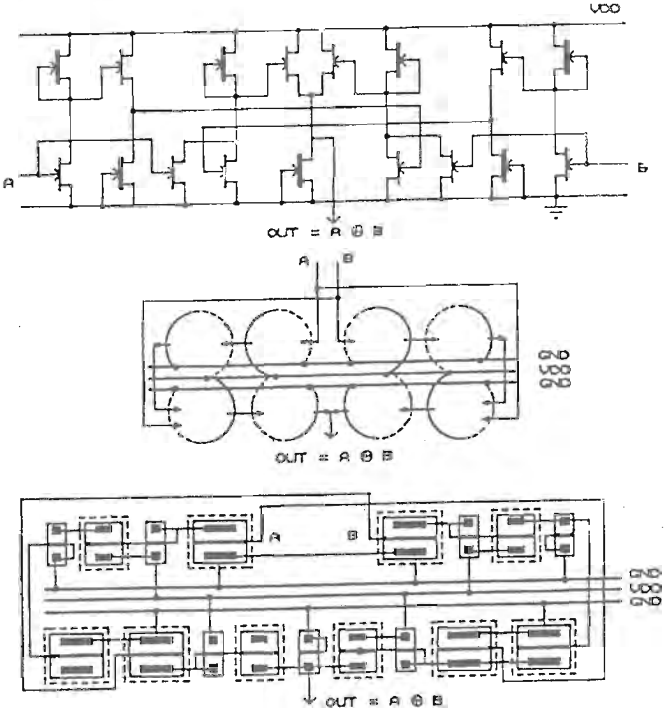


Figure 9 Methodology for Layout of Merged Logic Classes

2.4 Electrical and Geometrical Equivalence

During the translation of circuit description into layout, the related issue of "*Electrical Equivalence*" as distinct from "*Geometrical Equivalence*" must be addressed. The layout strategy must take into consideration the signal paths to minimize the skew. For example Figure 10, show the structure for a divide-by-four complementary clock divider. Each NOR gate is represented by its equivalent data flow diagram. Weights in accordance to the Fanin(x) and Fanout(y) requirements are assigned to each path. To reduce the interconnect delay and data skew, the data flow diagram can be rearranged such that the lengths of the feedback paths are minimized. This task sometimes is achieved by identifying axis of symmetry within a functional block.

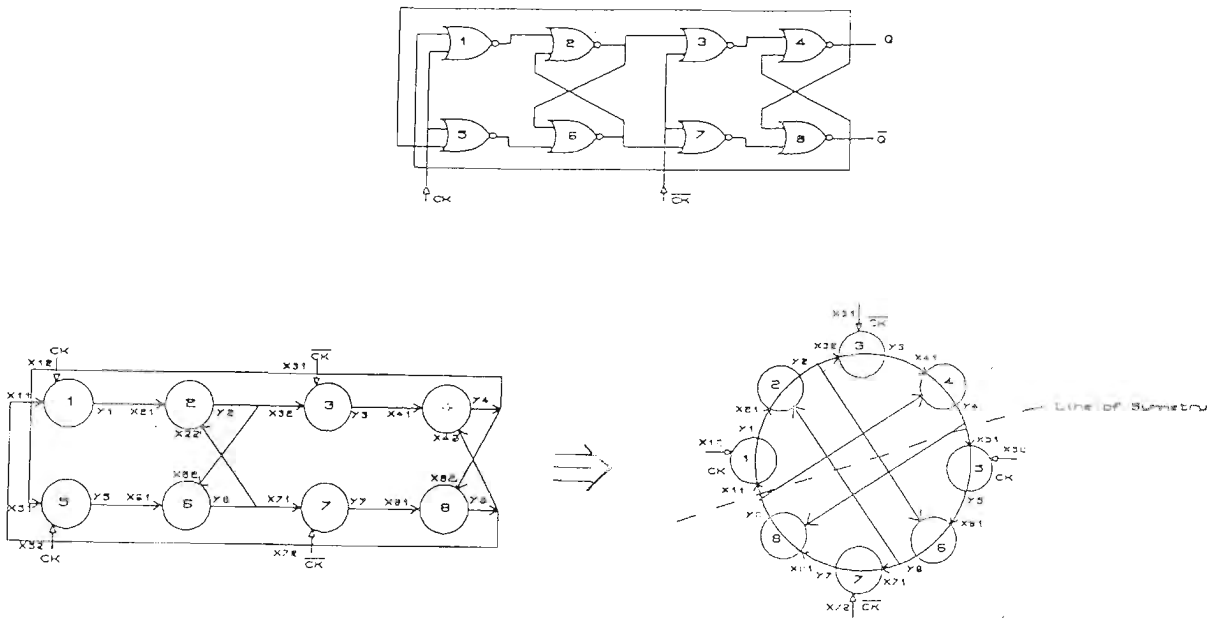


Figure 10. Data Flow Diagram for Complementary Clock Divider

The general layout strategy at a higher level of hierarchy is shown in Figure 11. There is one row of MESFETs on either side of the horizontal power bus. There is the interconnect channel between each pair of rows of MESFETs. This means individual MESFETs can be sized for performance without affecting the pitch of a basic functional block.

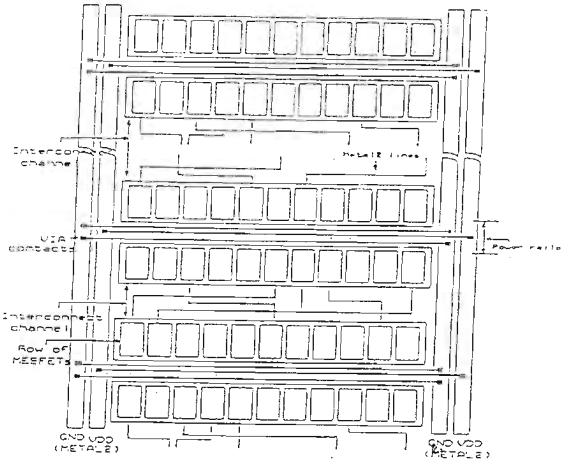


Figure 10 Layout Strategy for a Subsystem

3 Summary

The new design methodology and layout style for DCFL, SDCFL and SFFL classes of logic has shown considerable promise. The concept of Ring Notation has been demonstrated to be a useful mapping approach for high speed GaAs circuits by considering the influence of power and signal buses during the layout phase.

Acknowledgements

The contributions towards this program provided by Ms Song Cui and A Beamont-Smith from The Centre for Gallium VLSI Technology, Mr E. Bushehri and Mr R. Bayford of Middlesex Polytechnic U.K. is very much appreciated.

OPTIMISATION OF HIGH SPEED VLSI SYSTEMS: PHYSICAL MAPPING

KAMRAN ESHRAGHIAN

Centre for Gallium Arsenide VLSI Technology
The University of Adelaide
South Australia

1 Introduction

One of the major tasks that needs to be addressed early on in the design of High Speed and Very High Speed VLSI systems is the optimisation process. The very first step at the systems level is the realisation of a Floor Plan which by necessity entails some kind of Algorithm Transformation. Before mapping a system into either a Gallium Arsenide technology or for that matter into a high speed silicon technology, careful consideration must be given to a number of factors, including:

- regularity
- planarity
- intercell communication and interaction
- intercell routing
- clocking
- power distribution & routing
- control
- chip area
- dissipation

As a system's complexity and operational speed increases, its optimisation also changes as a number of different factors begin to influence its creation. For High Speed VLSI Systems clever partitioning aims at reducing implicit complexity by minimising the amount of interaction between subsystems. It is crucial that circuits or subsystems interacting with high frequency, are physically proximate, otherwise one may pay severe penalties for long, high-bandwidth interconnects.

The steps involved during physical mapping of algorithms usually enables us to highlight:

- inherent parallelism in the algorithm
- determine maximum throughput
- establish trade off between latency and throughput, that may lead to pipelining.

The approach that can be pursued during the optimisation process of a Very High Speed VLSI System can best be illustrated by examining a typical design cycle. As an example an image processing system for aerospace applications will be considered for such an exercise.

2 Image Processing

In order to realise high speed image post-processing functions as part of an imaging system for an aerospace vehicle, our first task is to identify the relevant areas, which include:

- Image registration
- Image enhancement
- Image classification.

Both image classification and image enhancement are important for aerospace imaging systems. However, in the following sections discussion is limited to image registration for the sake of simplicity.

2.1 Image registration

Image registration is the aligning of two images of the same scene from different perspectives. The change in perspective may be caused by movement of the aerospace vehicle or by virtue of two imaging systems mounted on different positions on the same craft. One image sensor may be infrared and the other may operate in the visible band, and so the two images may have different pixel resolutions.

Image registration is performed by the application of *image warping*, which is simply transforming an image from one coordinate system to another. A traditional example of warping is the cartographer's transformation of a map of the globe into a flat coordinate system. For example this class of processing may be implemented by a polynomial evaluator.

Thus, for smart aerospace vehicles, where real-time registration would be necessary during the link-up between two vehicles, a high speed GaAs hardware engine such as a polynomial evaluator could be desirable for performing such a task. Furthermore, for irregular shaped objects, such as other aerospace craft, the same polynomial evaluator can be used to find the centre of mass of the object, as an origin or reference point for registration.

3 Polynomial Evaluator

The polynomial evaluator is an example of a type of processing element (PE) that performs a reasonably complex function whilst still being architecturally simple. Although the implementation decided upon for this particular application is by no means the only approach nor is it the fastest, however, it works reliably, and with minimum chip area. An important aspect of the design methodology is the need for both top-down and bottom-up approaches. Top-down design is necessary to allow the behavioural description to be realised correctly, and bottom-up design is needed to ensure that the algorithm chosen to implement the PE is such that it can be readily mapped into Gallium Arsenide, and furthermore is consistent with the accepted design principles and limitations of the technology.

The class of logic that was chosen for the implementation of this functional element is Source-Follower Direct-Coupled FET Logic (SDCFL). This choice is based upon the favourable physical characteristics that this class of logic displays, namely high noise-margins, good temperature stability, and also good fan-out capabilities. The power dissipation and switching delays also are comparable with that of simpler DCFL.

3.1 Behavioural Model

The general representation of an arbitrary order polynomial is given by,

$$y(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0 \tag{1}$$

The behaviour of the Polynomial Evaluator is very simple to describe. It requires a set of coefficients $a_n, a_{n-1}, ..., a_0$ of an arbitrary order polynomial, and a step size, h, and then proceeds with an execution cycle to obtain the next data point. Such an execution cycle is repeated until no more data is required from the polynomial, at which time new coefficients and step size is loaded and the process is once again repeated.

3.2 Algorithmic Solution

As a first step, it is necessary to examine the algorithm and ensure it can be mapped into the appropriate technology; in this case the chosen technology being Gallium Arsenide.

The general form of a polynomial may be re-written in the nested form:

$$y(x) = ((...(a_n x + a_{n-1})x + a_{n-2})...)...)x + a_0$$

As can be seen, in this form, to evaluate points on the general curve, N multiplications and N additions are necessary. The multiply operation is significant as it can consume considerable amount of chip area and t herefore leads to high dissipation.

However, by examining the polynomial, it is possible to transform it into a form in which the only operation necessary for the evaluation of any point on the curve from the initial starting point is found by the simple application of an recursive addition process on the individual bits of both the initial point and a set of coefficients. The decomposition of the algorithm, into the recursive structure, can be expressed in the Taylor series form:

$$y(x_0 + h) = y(x_C) + h\dot{y}(x_0) + \frac{h^2}{2}\ddot{y}(x_0) + ... + \frac{h^n}{n!}y^{(n)}(x_0) \tag{2}$$

where:

x_0 = some value of x for which the derivatives $\dot{y}, \ddot{y}, ..., y^{(n)}$ exist,

$h = x - x_0$

and

x = the point at which evaluation of y is required.

Equation 2 may be re-written as:

$$y(x_0 + h) = y(x_0) + d_1(x_0) \tag{3}$$

where:

$$d_1(x_0) = h\dot{y}(x_0) + \frac{h^2}{2}\ddot{y}(x_0) + ... + \frac{h^{n-1}}{(n-1)!}d_1^{(n-1)}(x_0) \tag{4}$$

The polynomial may be reduced to a recursive set options by iteration of this process, ie.

$$d_1(x_0+h) = d_1(x_0) + h\dot{d}_1(x_0) + \frac{h^2}{2}\ddot{d}_1(x_0) + \ldots + \frac{h^{n-1}}{(n-1)!}d_1^{(n-1)}(x_0) \tag{5}$$

$$d_2(x_0) = h\dot{d}_1(x_0) + \frac{h^2}{2}\ddot{d}_1(x_0) + \ldots + \frac{h^{n-1}}{(n-1)!}d_1^{(n-1)}(x_0) \tag{6}$$

$$d_1(x_0+h) = d_1(x_0) + d_2(x_0) \tag{7}$$

This process may be continued until the first derivative is zero.

The polynomial may then be re-written in the new form, ie.

$$y(x_0+h) = y(x_0) + d_1(x_0)$$

$$d_1(x_0+h) = d_1(x_0) + d_2(x_0)$$

$$d_2(x_0+h) = d_2(x_0) + d_3(x_0)$$

$$\vdots$$

$$d_{n-1}(x_0+h) = d_{n-1}(x_0) + d_n(x_0)$$

$$d_n(x_0+h) = d_n(x_0)$$

where, in the most general terms:

$$d_j(x_0+h) = d_j(x_0) + h\dot{d}_j(x_0) + \frac{h^2}{2}\ddot{d}_j(x_0) + \ldots + \frac{h^{n-j}}{(n-j)!}d_j^{(n-j)}(x_0) \tag{8}$$

and

$$d_j(x_0) = h\dot{d}_{j-1}(x_0) + \frac{h^2}{2}\ddot{d}_{j-1}(x_0) + \ldots + \frac{h^{n-j+1}}{(n-j+1)!}d_{j-1}^{(n-j+1)}(x_0), d_0 = y \tag{9}$$

If

$$x_1 = x_0 + h,$$

then given that

$$d_j(x_0), j = 1, I, n, y(x_0),$$

it becomes possible to calculate exactly

$$y(x_1), d_j(x_1), j = 1, I, n.$$

Hence, any point at an integral number of h away from x may be evaluated by the following recursion:

$$y(x_{i+1}) = y(x_i) + d_1(x_i)$$

$$d_1(x_{i+1}) = d_1(x_i) + d_2(x_i)$$

$$d_2(x_{i+1}) = d_2(x_i) + d_3(x_i)$$

$$d_{n-1}(x_{i+1}) = d_{n-1}(x_i) + d_n(x_i)$$

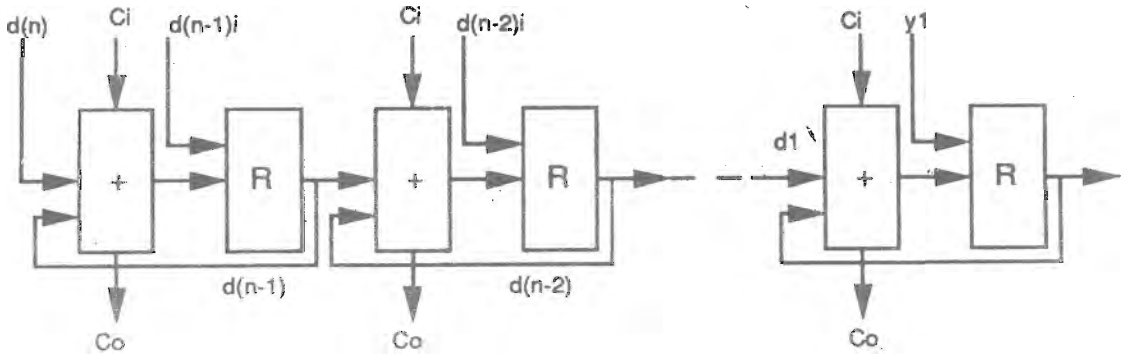


Figure 1: Block diagram of bit-serial algorithmic solution.

$$d_n(x_{i+1}) = d_n(x_i)$$

where

$$x_{i+1} = x_i + h$$

The system of equations above are significant because they require no multiplications. There is a trade-off though, since the initial conditions expressed in Eqn. 9 are required to be calculated.

If $x_0 = 0$, the problem is significantly reduced, because only the highest order derivative is non-zero, ie.

$$d_j(x_0) = \frac{h^{n-j+1}}{(n-j+1)!} d_{j-1}^{(nj+1)}(x_0) \tag{10}$$

this system may be calculated quickly or possibly stored for retrieval when required.

It should be noted that the manipulations required to convert an arbitrary polynomial into the form that is required by this algorithm are not trivial. This means the coefficients required by the hardware may take some time to calculate.

The possibility is to have coefficients pre-computed for a number of common polynomials and starting values. Approximations may then be made using interpolations for values that are required that do not coincide exactly with any stored set of coefficients. It is likely that for evaluations of polynomials requiring a large number of points from the same polynomial, significant speed increases in the evaluation would be obtained if the coefficients were calculated at run-time, since the time spent on this calculation would be offset by the large number of interpolations necessary if the 'approximate from stored values' method used.

3.3 Functional Specification

From the recursion relations it is readily observed that the transformed algorithm now requires an iterative addition process only. The logical architecture in the mapping process is shown in Fig. 1. Basically the required functional blocks to realise the algorithm are simply adders and registers. The adders perform the calculations and the registers initially store coefficients, and then store intermediate values as each block is clocked during the execution cycle.

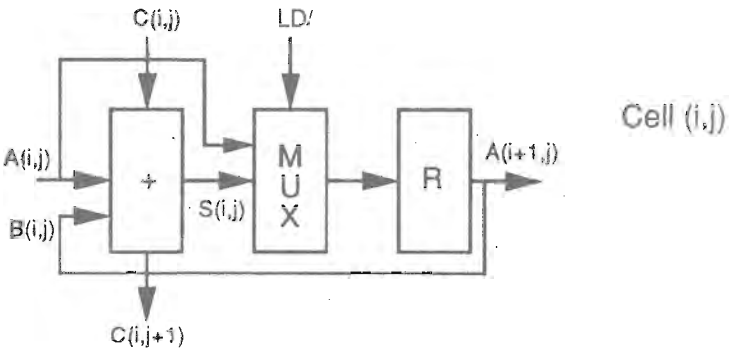


Figure 2: PE cell structural layout.

With reference to Fig. 1 the notation being implemented is as follows:

$d(n-j)$ = Coefficient $(n-j)$, all except $d(n)$ change as execution cycle is run.
$d(n-j)i$ = Initial coefficient $(n-j)$, loaded at start-up.
$y1$ = Initial evaluation point.
C_i = Carry in
C_o = Carry out.

The coefficients $d(n-j)i, d(n)$ must be loaded initially into the system. This can be achieved by the inclusion of a 2 way multiplexor, so that when coefficients are being loaded into the system, the adders are by-passed and no execution cycles take place. With the addition of this requirement, the system now has two states: Load, and Execute. It is possible to use an extra 1-bit signal to realise this control.

3.4 Structural Description (Floor Plan)

A single PE cell consists of the three functional elements, an adder, a register and a diplexor. These elements may be connected as shown in Fig. 2 to form the PE cell for the polynomial evaluator.

The approach lends itself readily to replication and minimum intercell communications and routing. A single PE cell may be replicated both horizontally to increase the order of polynomial and vertically to increase the bit size of the results.

3.5 Physical Description

The physical description entails design and layout of the basic elements of the PE namely the adder, the register and the diplexor.

3.5.1 The Adder

For the sake of simplicity, initially a simple ripple-through carry adder appeared adequate without overly increasing the complexity of the cell.

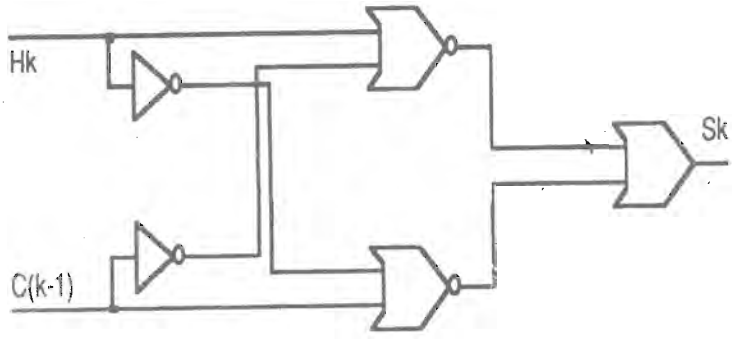


Figure 3: Generation of sum result.

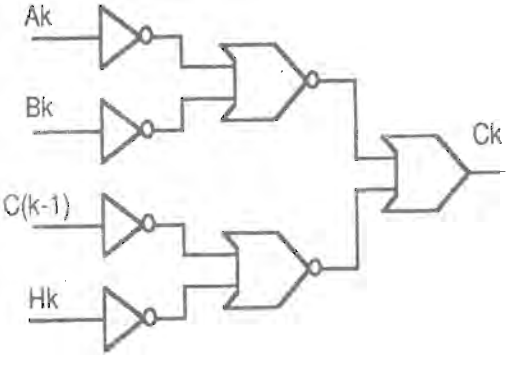


Figure 4: Generation of carry result.

Since the class of logic that has been decided upon for the implementation of the PE is SDCFL GaAs, it is necessary to manipulate the equation into a form which contains only inverters and NOR gates. The OR gate in SDCFL is of the 'wired-OR' form, and hence can only be used between NOR gates and inverters. Following is the logic expressions for the three required results and their transformed equivalents.

Half-Carry: $H_k = \overline{A_k}B_k + A_k\overline{B_k} = \overline{\overline{A_k}\,\overline{B_k}} + \overline{\overline{A_k}B_k}$

Sum: $S_k = H_k\overline{C_k} + \overline{H_k}C_k = \overline{\overline{H_k}\,\overline{C_k}} + \overline{H_k\overline{C_k}}$

Carry: $C_k = A_kB_k + H_kC_k = \overline{\overline{A_kB_k}} + \overline{\overline{H_kC_k}}$

Figs. 3 and 4 illustrate the logical transformation of two of the three expressions.

3.5.2 The Register

The register chosen is by necessity an edge triggered device. The data at the inputs would be changing while the data at the output is still needed for the next PE cell to the right. Therefore transparent

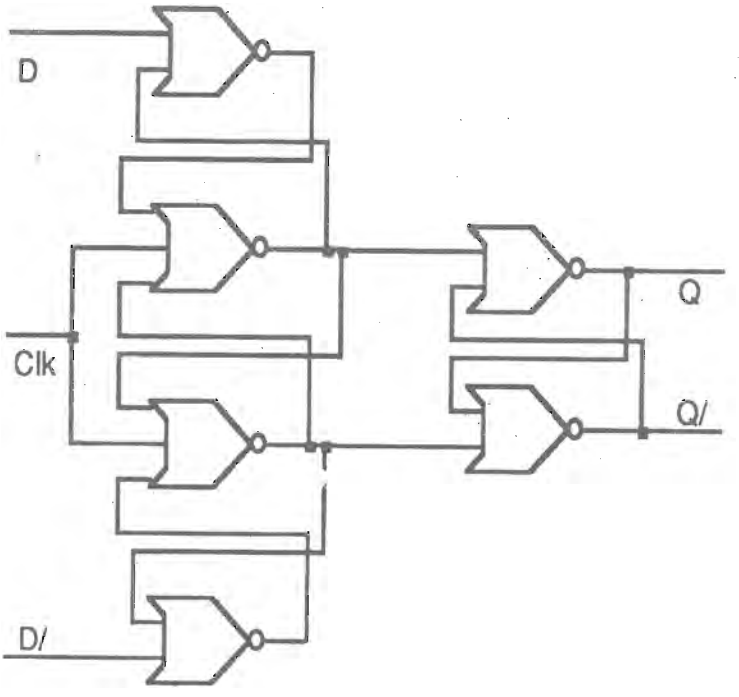


Figure 5: Register logical representation.

latches cannot realise this property without placing constraints on the clocking strategy. Thus a negative edge-triggered 'D' type latch as shown in Fig. ?? was implemented.

3.5.3 The Diplexor

The logic equation that describes the diplexor, as shown in Fig. 6, is given by:

$$\overline{Z} = I_0\overline{LD} + I_1 LD$$

The cell and was realised using two SDCFL 'NOR' gates with their outputs in a 'Wired-OR' configuration.

3.5.4 The PE Cell

The PE cell in its entirety consists of the adder, diplexor and latch connected in such a way so that the input to the latch is selectable between the A input to the adder, and the S output of the adder. Depending on the state of the control input, LD, either (a) the system passes constant values through the latches (latch input = A of adder), ie. loading data coefficients, or (b) the system executes an evaluation cycle and produces a result (latch input = S of adder). Since the output of the diplexor is the complement of the selected input, it is necessary to connect the diplexor output to the $\overline{D}$ input of the latch in order to have the correct value latched. The D input of the latch is connected to the output of the NOR gate adjacent to the $\overline{D}$ input. This allows the latch to be operated without the

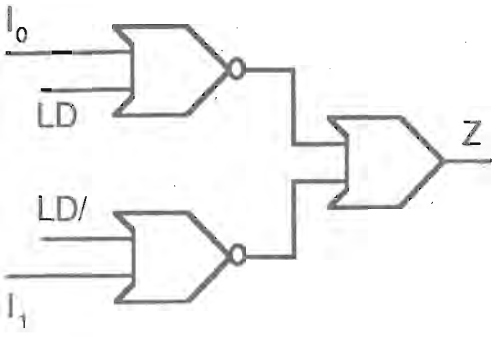


Figure 6: Diplexor logical representation.

normally required complementary input. The Q output of the latch is connected to the B input of the adder, to realise the feedback path and hence the recursive property of the cell.

3.6 Physical Layout

Each single PE cell contained the three functional elements, the adder, the D-latch and the diplexor. The adder cell was initially designed using the 'ring notation,' ie. a design methodology most suitable for this class of logic. The floor plan for the implementation of the PE is shown in Fig. 7. It encompasses all the necessary design criteria, including placement of the input and output signals so that the PE cell may be assembled with minimum length routing paths between sub-circuits.
The Ring Notation diagrams for the three functional elements are shown in Figs. 8, 9 & 10.

In the diagrams, the thick horizontal lines represent the power and ground buses. All the buses have GND, VDD and VSS rails, Bus (1) also contains LD and $\overline{LD}$ signal, bus (2) has the Clk signal for the register, and bus (3) has the feed-forward by-pass path for the A_k signal through the adder, and the feedback path for the Q output to the B_k input through the register.

In the layout, the adder occupied 5 rows of gates. This was fortunate due to the fact that the diplexor occupied one row, and the D-latch 4 rows. This permited the Adder to be placed on the left of the cell, with the diplexor and the latch placed vertically (relative to each other) to the right of the adder. The full cell arrangement is illustrated by Fig. 11.

Since this placement of devices occupies 5 horizontal rows, it was not possible to stack these cells vertically, because two rows share a single set of power and ground buses. Therefore it is required that an even number of rows of devices be present in order to replicate any block vertically. Hence it was necessary to layout another PE cell below the first, and an anti-symmetric arrangement with respect to the power and ground buses. This arrangement occupied now a total of 10 rows.

In the 2 PE cell arrangement, shown in Fig. 12, the clock (clk) and control signals (LD and $\overline{LD}$), as well as the feed-forward A signal (to the input of the diplexor), and the feedback Q signal (to the B of the adder), are separated from the VSS and VDD rails. This is due to the high level of crosstalk and capacitive effects which introduce noise into the supply rails when there are fast switching transients on signal lines nearby.

The PE will only load coefficient data when the circuit connected to the input buffer has data available, and similarly the PE will only clock an execution cycle when the circuit connected to the output has

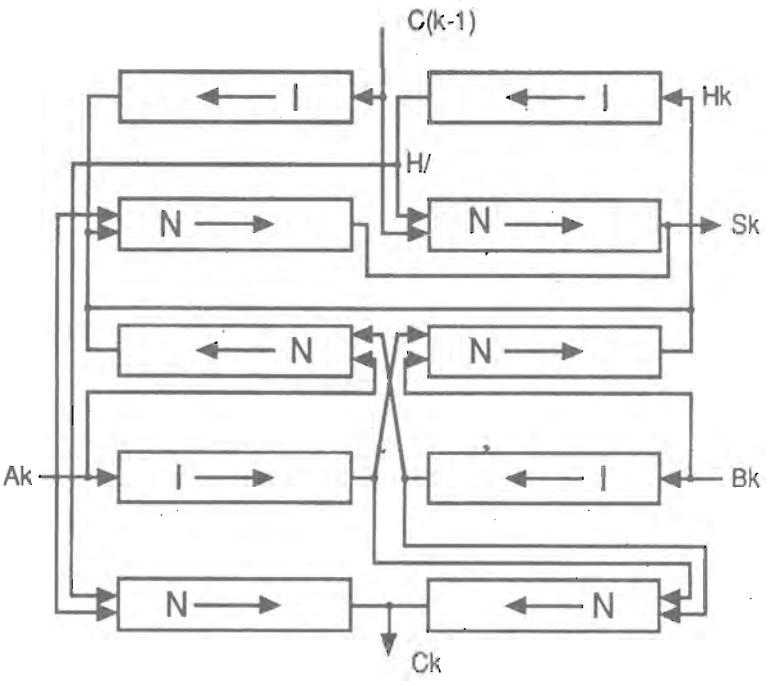


Figure 7: Adder floor plan block diagram.

indicated that it is ready to receive data. The LogicWorks^{TM} circuit diagram of the adder, register unit and the PE cell, together with corresponding simulation results are shown in Figs. 13, 14 and 15.

The propagation delay of each cell in a horizontal manner is approximately 2.8ns. In the vertical direction, it is necessary to examine the propagation delay of the carry signal through each cell. Since during each evaluation cycle data is only moved from one cell to its immediate right neighbour, the horizontal delay is independent of the polynomial order. The vertical delay, however, is a different matter. The carry signal must propagate through the entire bit-depth of the array and allow the adder output data to settle in the most significant bit before any clocking of the MSB row can take place. It does not appear advantageous to clock the lesser significant bits before the entire arrays adder output is stable.

The estimation of the delay time can now be made. Using the data obtained from the adder simulation plot, the carry signal delay is in the worst case 1200ps. Therefore we must wait $(n-1)$ times this long (for n bits deep array), plus the horizontal delay. As the bit depth increases, the horizontal delay contributes relatively less to the overall delay. Note that this is only the theoretical minimum clock delay for an evaluation cycle, and a data load cycle will have considerably less delay involved, since no carries are required, and furthermore the adder is by-passed resulting in a very fast load cycle (compared to the execution cycle). The delay (for the execution cycle) is therefore estimated to be 741200ps (carry prop. delay) = 8.4ns + 142800ps (PE cell prop. delay) = 11.2ns, giving a theoretical clock speed of 90MHz.

Although this figure could be construed as being fairly slow for a technology such as GaAs, the most

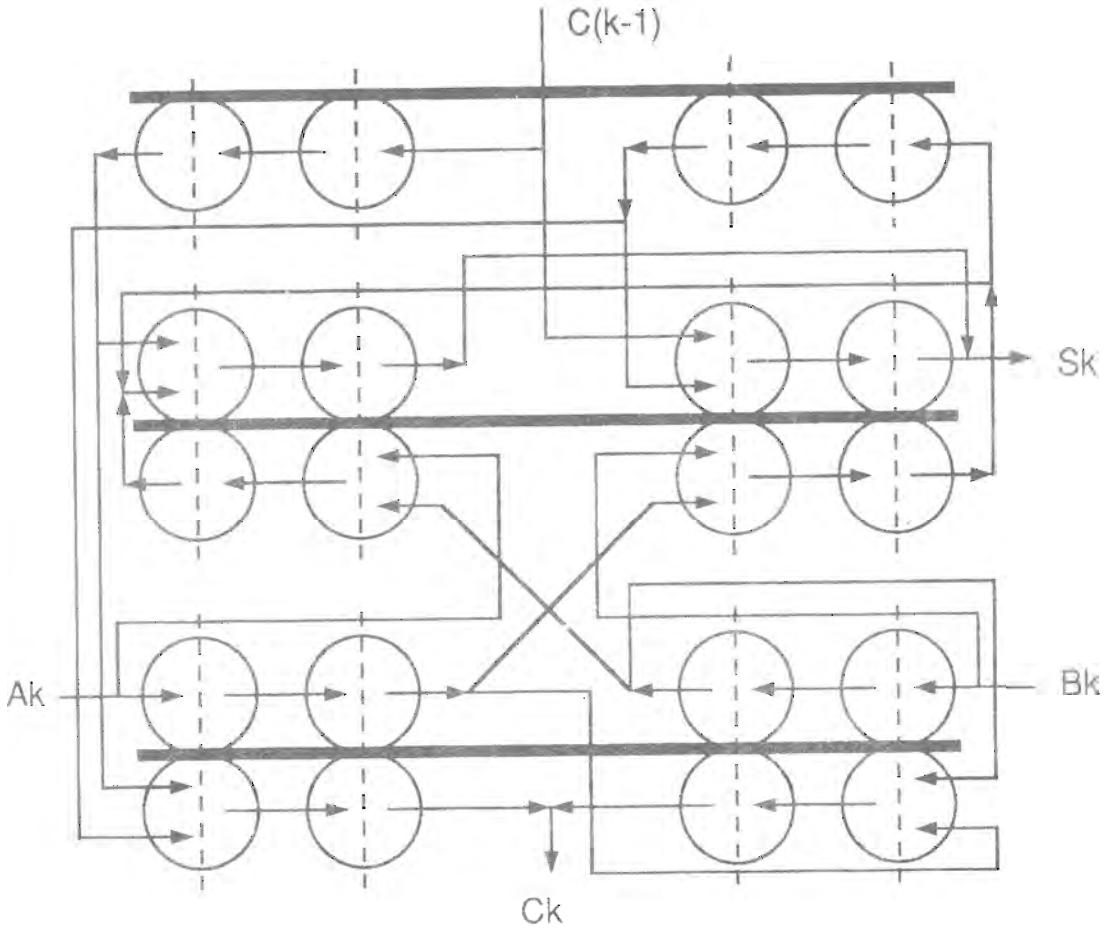


Figure 8: Adder cell ring notation.

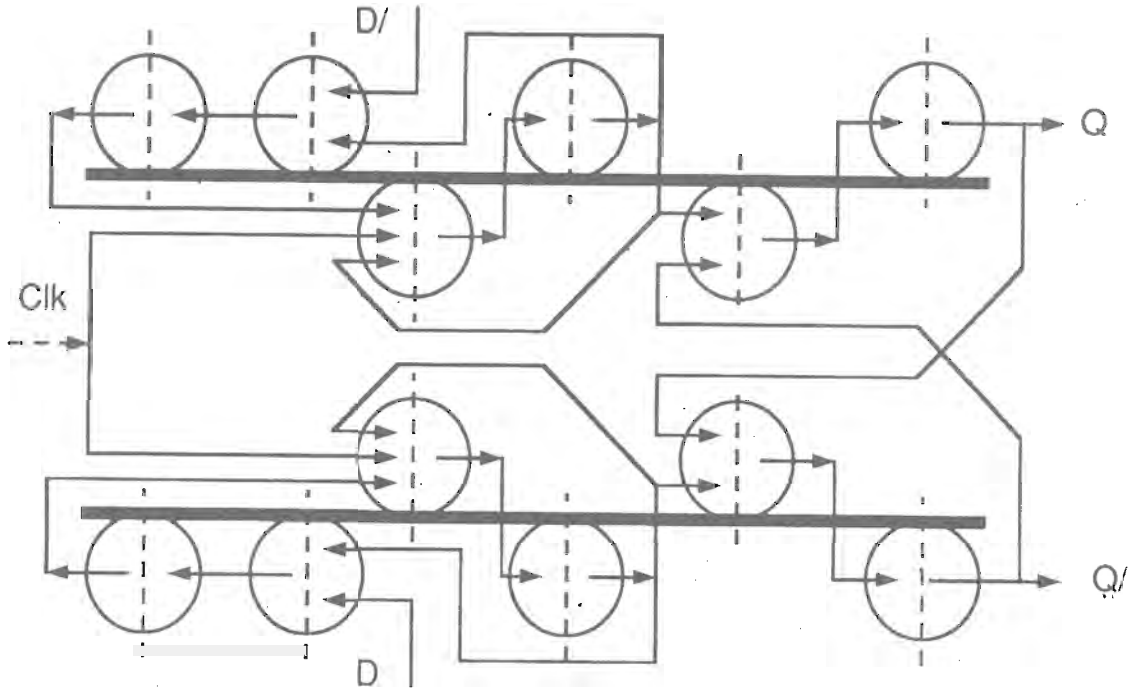


Figure 9: Register cell ring notation.

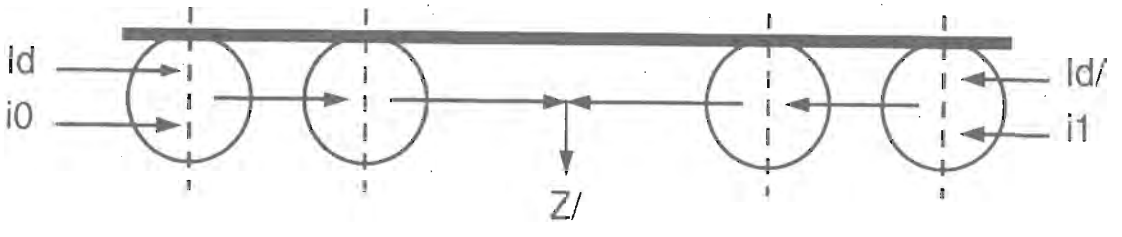


Figure 10: Diplexor cell ring notation.

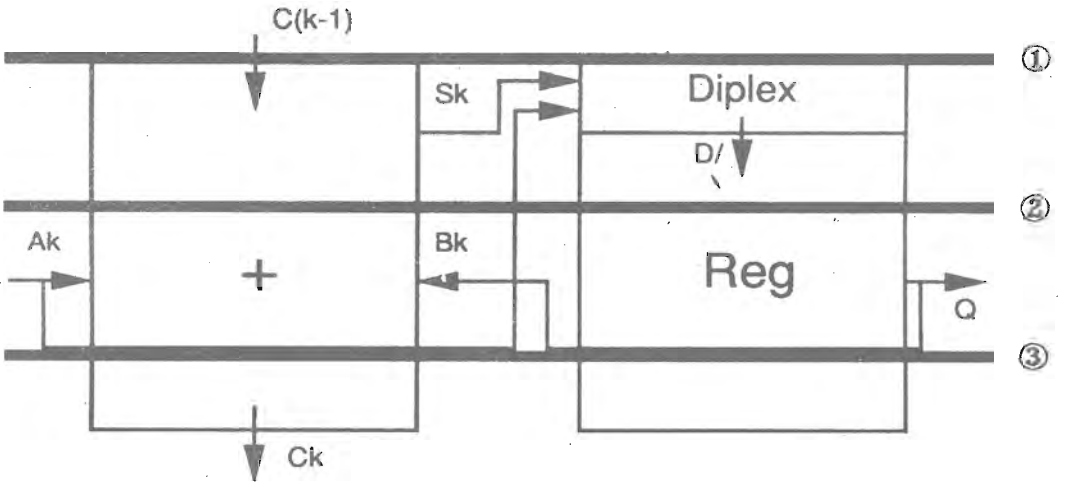


Figure 11: PE cell structural block diagram.

serious problem occurring here is the time complexity of the array being proportional to the bit-depth of the implementation. If this dependency could be removed, or at least reduced, significant savings could be made in the propagation delays of the array. One possible method of realising this would be to substitute the present ripple-through carry adder with a device such as a 4-bit carry lookahead adder. This would significantly reduce the time taken for the carry to propagate vertically through the array, and hence improve the overall performance of the circuit.

4 Acknowledgements

The significant contributions provided by Sam Mosel and Derek Abbott towards the preparation of this material is gratefully acknowledged. The support provided by the Sir Ross and Sir Keith Smith Foundation towards this research program is also noted.

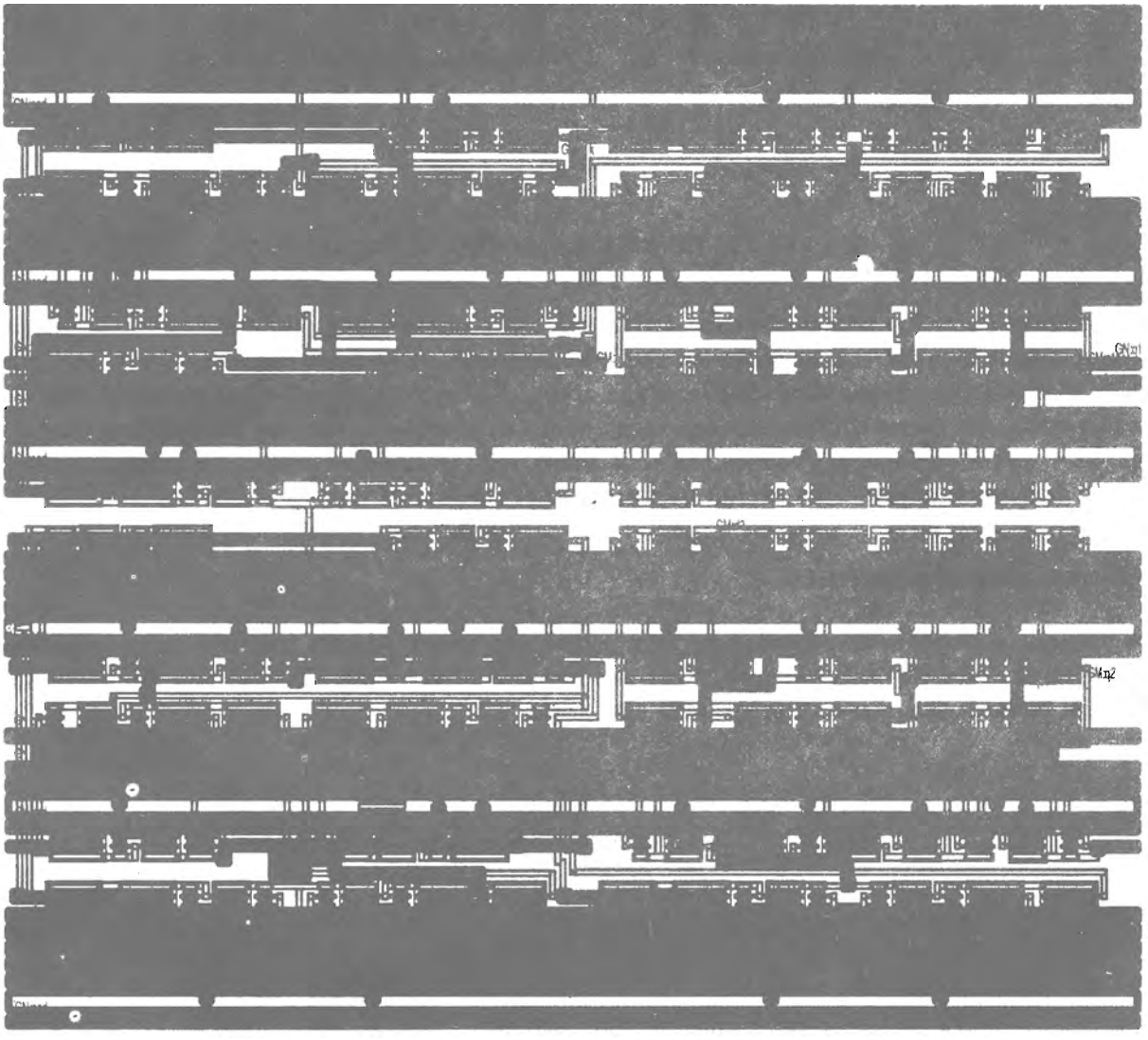


Figure 12: Polynomial evaluator cell.

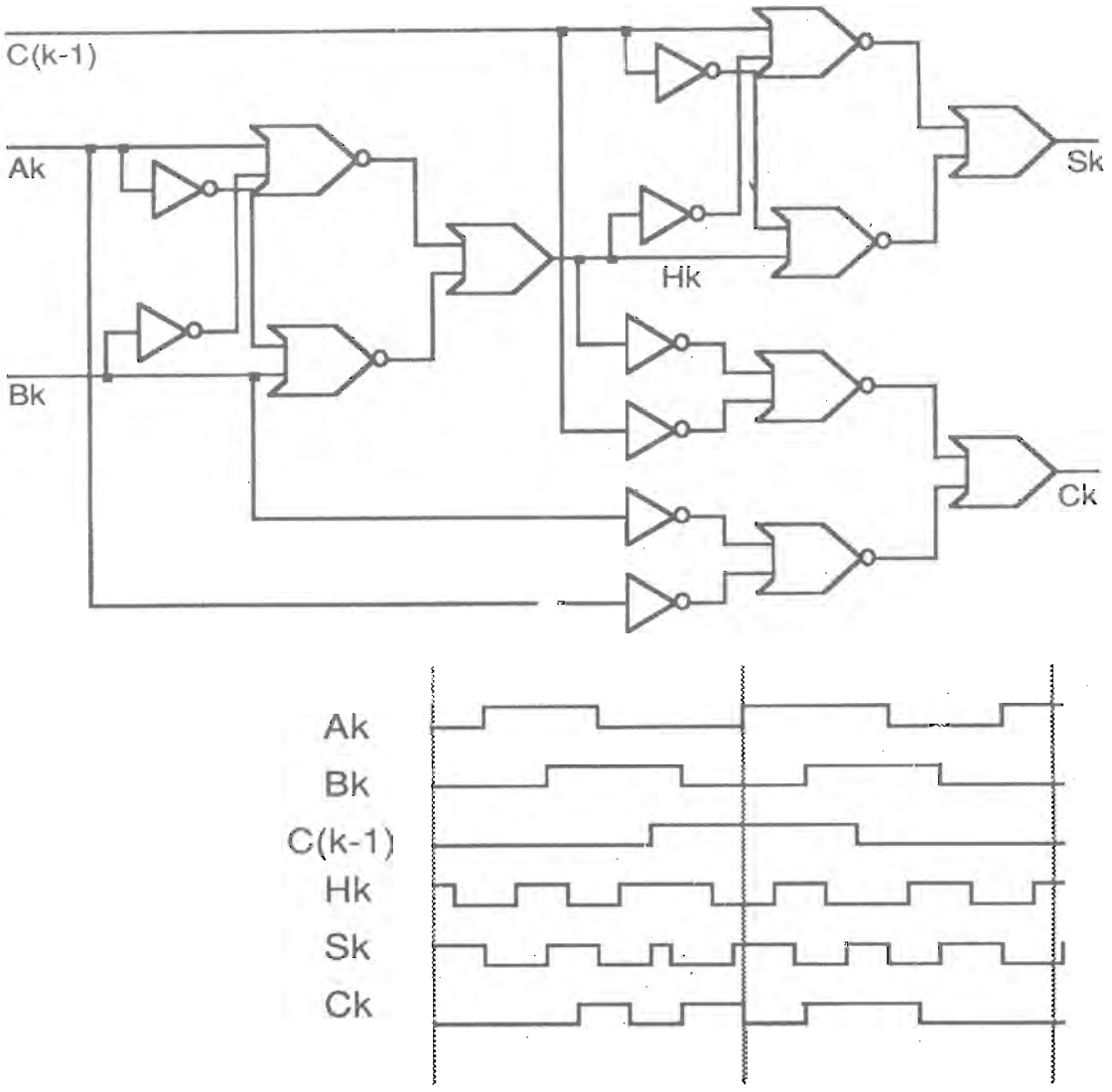


Figure 13: Adder and simulation.

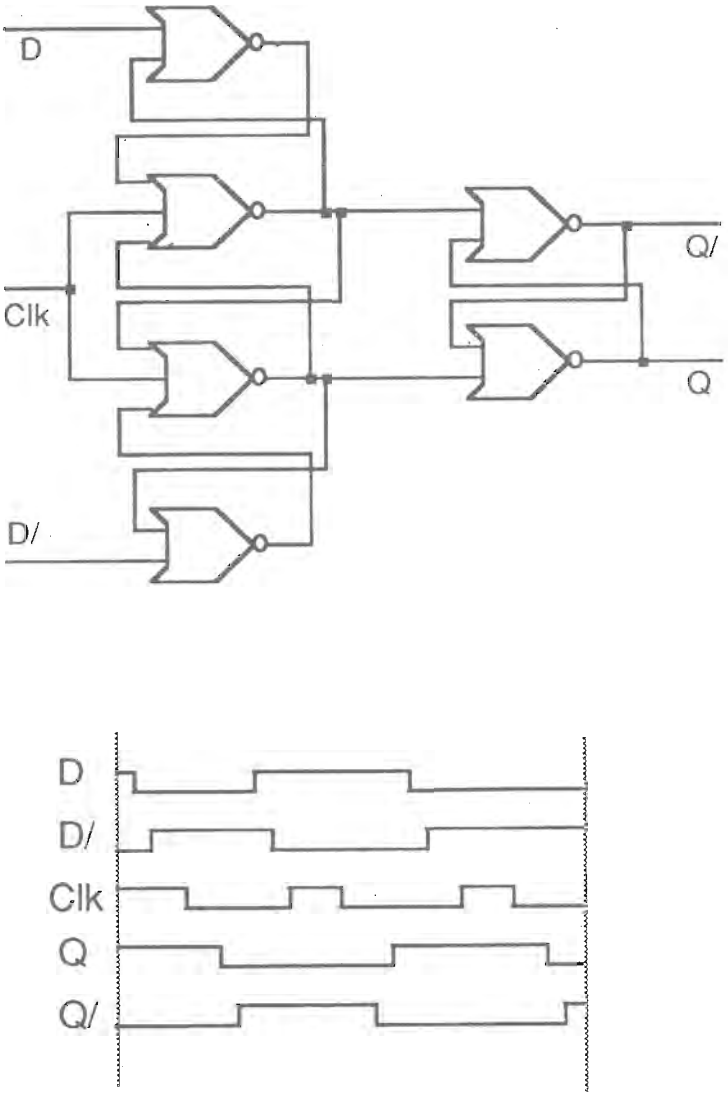


Figure 14: Register and simulation.

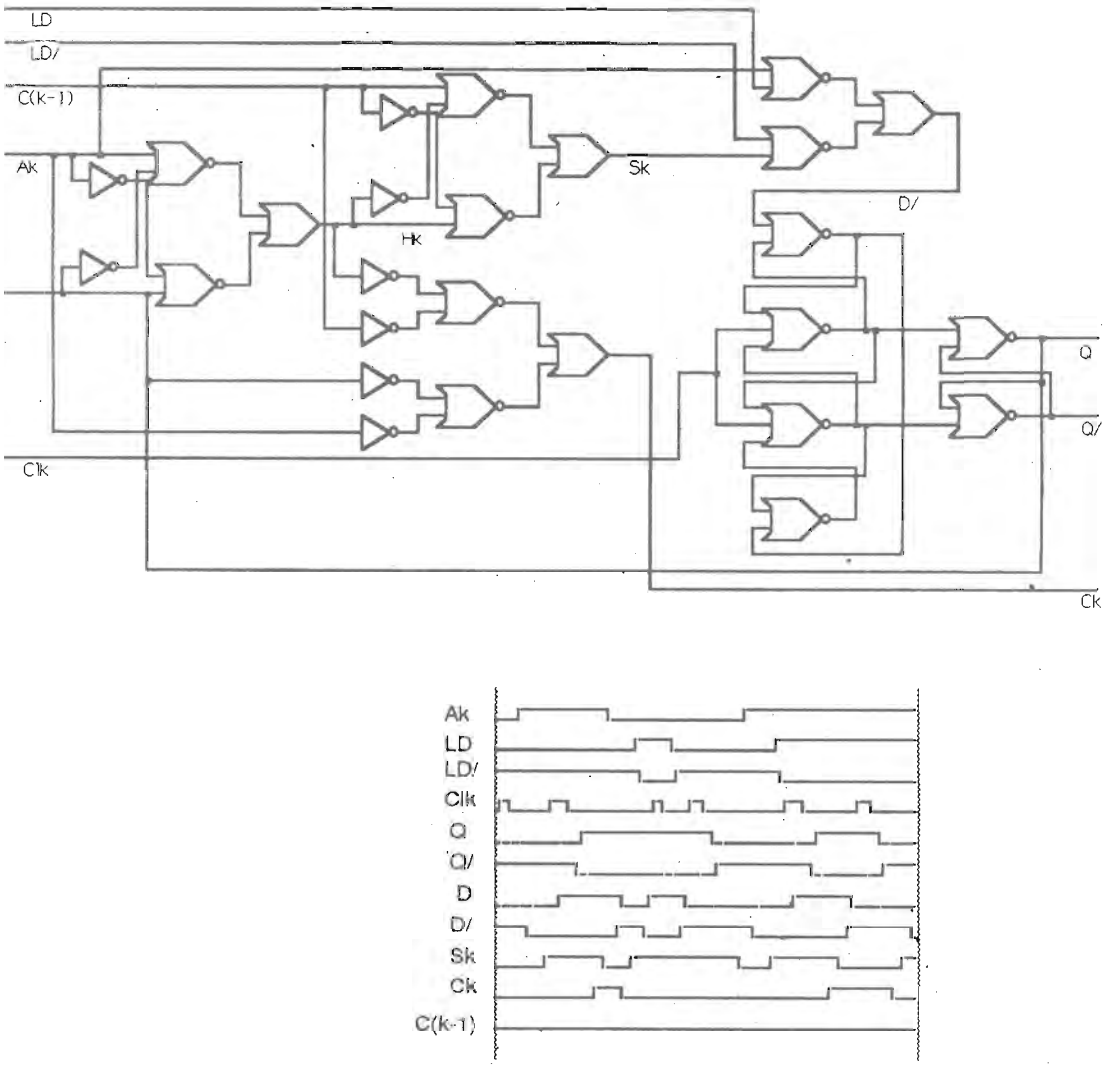


Figure 15: Block diagram of bit-serial algorithmic solution with PE cell structural layout below.

Modelling for Analogue Systems

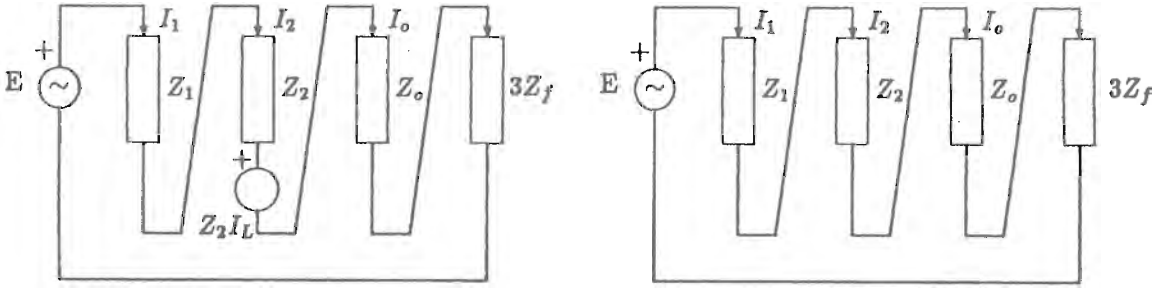
P R P Hoole and T Kirubarajan
Faculty of Engineering
University of Peradeniya
Peradeniya
Sri Lanka.

1 INTRODUCTION

This paper provides a class-room type tutorial to introduce some of the basic concepts involved in mathematical modelling. Without sacrificing the depth of engineering concepts involved in mathematical models, the following sections seek to highlight the importance of modelling in the design and development of both devices and systems. The paper is organized as follows: section 2 presents the identification of a signal using circuit models to detect high impedance(low current) faults. Section 3 discusses the limitations of circuit models, and gives an example of a case where electromagnetic field analysis is required. Modelling of a microwave remote sensing system to detect tilts and rotation of a body is described in section 4, while in section 5 some microwave circuit design-related issues are discussed. In section 6 we develop a circuit model for magnetic devices with hysteresis losses.

2 ELECTRIC POWER SIMULATION

Symmetrical component theory is used in all asymmetrical fault analysis of power system circuits. The computer simulation results are used to design the protection system and settings for the network. However, the mathematical models for the network assume that the fault currents are much greater than the load current. Due to this when the fault currents are very small, the protective system fails to detect the fault leading to long-term damage to equipment or loss of generators. In Figure 1(a) is presented a modified equivalent circuit for line-to-ground fault. The model used by commercial power system fault simulation packages is given in Figure 1(b).



(a) High Impedance Fault (b) Low Impedance Fault

Figure 1. Interconnection of Networks

What is significant is that for high-impedance(low fault current) faults, there is an additional voltage or current source which appears in the negative sequence network. In other words the pre-fault positive sequence load currents induce post fault back emf in the negative sequence network.

The currents for the network in Figure 1(a) are

$$I_1 = \frac{E_1 - Z_2 I_L}{Z_1 + Z_2 + Z_o + 3Z_F} \quad (1)$$

$$I_2 = \frac{E_1 - Z_2 I_L}{Z_1 + Z_2 + Z_o + 3Z_F} + I_1 \quad (2)$$

$$I_o = \frac{E_1 - Z_2 I_L}{Z_1 + Z_2 + Z_o + 3Z_F} \quad (3)$$

and for Figure 1(b) we obtain a set of equivalents by simply setting $I_L = 0$ in (1) - (2). Consider now an 11kV synchronous generator, connected to a 11/132 kV transformer feeding unity power factor 10MW load. The relevant data for the system are as follows

1. Generator: $X_1 = 0.15$ p.u., $X_2 = 0.1$ p.u., $X_o = 0.03$p.u. all on 10 MVA base. Star point of winding solidly earthed.
2. 11/132 kV Transformer: $X_1 = X_2 = X_0 = 0.1$ p.u. on a 10 MVA base. 11 kV winding delta connected and 66kV winding star connected with star point solidly earthed.
3. Feeder: $X_1 = X_2 = 0.05$ p.u., $X_0 = 0.15$ p.u. on a 10MVA base.

Calculate the fault current when a single-phase-to-earth fault occurs at the load end of the terminal, for fault impedance Z_f is equal to (i) 3 Ω (ii) 3000 Ω (iii) $j3000$ Ω.

The 3 Ω, 20 Ω fault impedances have the following p.u. values: 0.00172 and 1.72. The load current is 43.7 A or 1 p.u.

(1) For a 0.0172 p.u. fault impedance, we may ignore the load current.

$$I_1 = I_2 = I_0 = \frac{E_1 - Z_2 I_L}{Z_1 + Z_2 + Z_o + 3Z_F} \quad (4)$$

$$Z_1 = j0.25\text{p.u.}, Z_2 = j0.25\text{p.u.}, Z_o = j0.25\text{p.u.} \quad (5)$$

$$I_f = 3.75 \text{ p.u.} = 164 \text{ A.} \quad (6)$$

We have ignored the $3Z_f$ term, since $Z_f \ll Z_1$.

(ii) When $Zf = 1.72$, we must resort to 1 - 2.

$$I_1 = 0.1966 - j0.0766 \quad (7)$$

$$I_2 = 1.2 - j0.766 \quad (8)$$

$$I_o = 0.2 - j0766 \quad (9)$$

$$I_f \approx 0.6 - j0.2 \quad (10)$$

Comments: When $Z_f = 0.0172(3\ \Omega)$, the fault current is 3.75 p.u., which is significantly greater than the load current, and thus may be detected by a conventional over current relay. The negative sequence current in this case is 1.25 p.u. When $Z_f = 1.72(3000\ \Omega)$, the fault current is about 0.6 p.u., which is insufficient to trigger an overcurrent relay. However, the key to detecting the high impedance fault may be the negative sequence current. Prior to the high-impedance fault, $I_2 = 0$, the post fault $\|I_2\| \doteq 1.4$p.u.

In this simple tutorial exercise we have illustrated how circuit modelling of a power system subject to a high impedance (low current) fault, could be used to identify the signal which permits us to detect the presence of this elusive faulted state. We have developed a general computer package HIGHZ to simulate high impedance faults on multi-busbar power systems.

3 ELECTROMAGNETIC SIMULATION

In this workshop we have virtually concentrated on circuit design based on circuit simulations. Devices and systems are modelled by resistors, inductors, capacitors, voltage sources and current sources. But these circuit elements are basically models of the more fundamental electromagnetic phenomena. Maxwell's equations for electromagnetic fields, the potential definitions, and the concepts of resistance, inductance and capacitance combine to produce the common expressions of circuit analysis. The assumptions upon which circuit theory is based are as follows:

1. filamentary conductor defines the closed path or circuit
2. the maximum dimensions of the current are small compared to a wavelength
3. displacement current is confined to capacitors
4. magnetic flux is confined to inductors
5. imperfect conductivity is confined to resistors

Under certain conditions the circuit ceases to obey the laws of circuit theory (e.g. if the frequency is sufficiently high) and we would have to resort to electromagnetic simulators for analysis and design of a device or system. For instance, design of transistors, VLSI process-fabrication control, and eddy current systems require finite-element models of the electromagnetic phenomena.

An important application of eddy current simulators is non-destructive testing. Consider the simple one-dimensional problem shown in Figure 2.

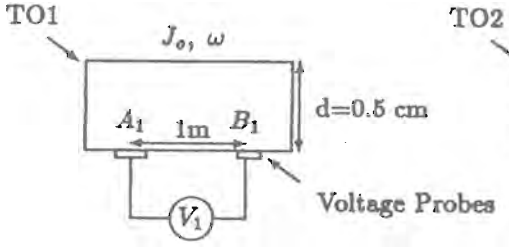


Figure 2(a)

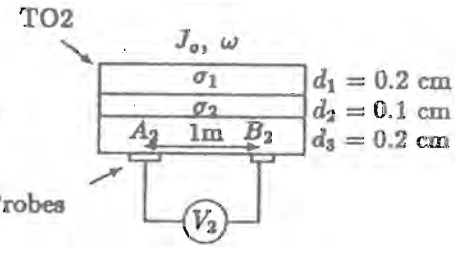


Figure 2(b)

We are given two test objects of identical geometry and dimensions. The test object TO1 is in good order. When a current carrying conductor (shown shaded in Figure 2(a)) is impressed on it, the voltage picked u p across probes A_1B_1 is given by

$$V_1 = \frac{J_1}{\sigma_1} \cdot l = \frac{J_1}{\sigma_1} \text{ volts} \quad (l = 1m) \tag{11}$$

Given current density J_o and frequency ω in the source conductor, J_1 may be found from $\nabla^2 J = j\omega\mu\sigma J$ giving

$$J_1 = J_o e^{-d/\delta_1} e^{-jd/\delta_1} \tag{12}$$

where skin depth $\delta_1 = \sqrt{2/\omega\mu\sigma_1}$. Therefore

$$V_1 = \frac{J_o}{\sigma_1} e^{-d/\delta_1} e^{-jd/\sigma_1} \text{ volts} \tag{13}$$

In test object TO2, a small 0.1cm thick region has deteriorated and subsequently its material properties have changed. Assuming that the test object is an electrical conductor, we assume that conductivity in this region has dropped to σ_2. (Notes: (1) if the test objects were magnetic

materials, the change will be in magnetic conductivity μ (ii) in geotechnical engineering, similar technique in employed to obtain data about the earth's crust. σ_1 may be the conductivity of rock, and σ_2 conductivity of gold.) For test object TO2,

$$J_2 = J_o e^{-d_1/\delta_1} \cdot e^{-jd_1/\delta_1} \cdot e^{-d_2/\delta_2} \cdot e^{-jd_2/\delta_2} \cdot e^{-d_3/\delta_3} \cdot e^{-jd_3/\delta_3}, \tag{14}$$

where $\delta_2 = \sqrt{2/\omega\mu\sigma_3}$. Thus

$$\begin{aligned} V_2 &= \frac{J_o}{\sigma_1} e^{-(d_1/\delta_1 + d_2/\delta_2 + d_3/\delta_3)} e^{-j(d_1/\delta_1 + d_2/\delta_2 + d_3/\delta_3)} \\ &= \frac{J_o}{\sigma_1} e^{-(2d_1/\delta_1 + d_2/\delta_2)} e^{-j(2d_1/\delta_1 + d_2/\delta_2)}, \end{aligned} \tag{15}$$

since $d_1 = d_3$.

Comparing equations (13) and (15), we note that V_1 and V_2 differ both in magnitude and phase. In non-destructive testing, we seek to establish (a) that the material or device has deteriorated (b) the values or d_2 and σ_2.

Let $J_o = 10^4\ A/m^2, \sigma_1 = 10^7, \sigma_2 = 0.5 \cdot 10^7, \omega = \pi \cdot 10^5, \mu = \mu_o$. Then $\delta_1 = 0.711$ mm, $\delta_2 = 1.006$ mm. The magnitudes of V_1 and V_2 are $\|V_1\| = 0.88\ \mu$Volt and $\|V_2\| = 3.3\ \mu$Volt.

A comparison of $\|V_1\|$ and $\|V_2\|$ indicates that there is change in material properties in TO2. In non-destructive testing, the question is, having measured V_2, can we determine d_2 and σ_2. This is termed the *inverse problem*.

We shall briefly illustrate how techniques commonly employed in adaptive signal processing may be applied to solve the inverse problem. The basic system is shown in Figure 3(a).

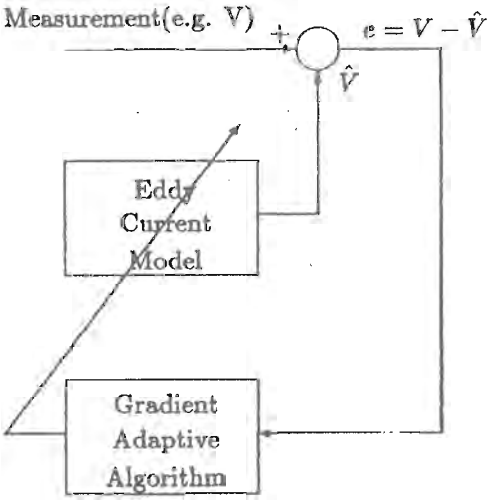


Figure 3(a)

The measured potential (V_2 is compared with that obtained from the mathematical model (e.g. finite element algorithm). For the problem defined in Figure 2(b), say $d_2 = c$, $d_1 + d_3 = d - c$. We need to obtain σ_2^* (orδ_2) and c. We obtain $\hat{V}_2$ from the mathematical model using an initial guess of δ_{20} and c_0. In the subsequent iterations we seek to move δ_{2i} and c_i towards their correct values by minimizing the error $V_2 - \hat{V}_2$. An intelligent way to move δ_{2i} and c_i toward the correct values is to use the gradient estimator:

$$\delta_2(i+1) = \delta_2(i) - \Delta_\delta \frac{\partial F_i}{\partial \delta_2} \tag{16}$$

$$c(i+1) = c(i) - \Delta_c \frac{\partial F_i}{\partial c}, \tag{17}$$

(18)

where

$$F = \frac{1}{2}(V_2 - \hat{V}_2)^2 \tag{19}$$

where V_2 is defined by (15). Hence

$$\frac{\partial F_i}{\partial \delta_2} = -\frac{c}{\delta_2^2} V k e^{-c\Delta} + \frac{2k^2 c}{\delta_2} e^{-2c\Delta} \tag{20}$$

$$\frac{\partial F_i}{\partial c} = \Delta V k e^{-c\Delta} - 2\Delta k^2 e^{-2c\Delta} \tag{21}$$

$$k = \frac{J_o}{\sigma_1} e^{-d/\delta_1} \tag{22}$$

$$\Delta = \frac{1}{\delta_2} - \frac{1}{\delta_1}. \tag{23}$$

By repeatedly solving for $\delta_2(i+1)$ and $c(i+1)$, we finally arrive at the correct values of δ_2 and c when $F \approx 0$.

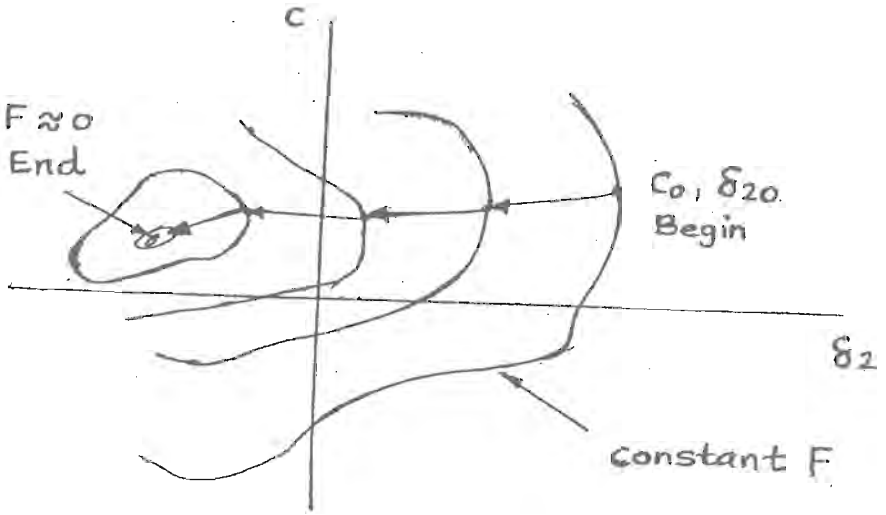


Figure 4.

The inverse problem is generally more complex than shown in Figure 2(b). The material change may not extend over the full length of the object, but be confined to a limited area so that a one-dimensional analysis is not adequate. For such cases we have developed our own two-dimensional eddy current simulator on an IBM-PC compatible.

4 MODELLING FOR REMOTE SENSING

A helicopter is one of the many vehicles and control systems driven by electric machinery where the position of the rotor could be used to track the state and the trajectory of the vehicle or system. In tracking a helicopter, for example, the helicopter body attitude angles and rotor blade tip path plane angles may be used in a nonlinear tracker. Body attitude angles are useful in

tracking both fixed wing aircraft and helicopters. With helicopters, additional information may be obtained by using the tip path plane of the rotor blades, which are oriented with respect to the helicopter fuselage. The tip angle is directly related to the main rotor thrust which influences the flight of the helicopter. The orientation measurement of the tip path plane provides valuable information on the translation motion of the helicopter. When, for instance, the pilot wishes to translate forward, he or she will tilt the tip path plane down towards the nose of the helicopter. This will result in both a downward motion of the nose and a forward component of the thrust.

Using the Doppler signature, where the scattered field due to the advancing and retarding blades may be filtered out, we want to develop a state estimator to estimate the radius and the angle of rotation of the rotor. The rotating blade is modelled as a homogeneous cylinder of radius r and length $2l$. For simplicity he we consider the case of pure rotational motion of the cylinder. To derive the state estimator we use the Doppler signature of the vibrating sensor. The incident field U_i and the magnetic field back scattered by the cylindrical sensor U_{sh} are given by

$$U_i = \left(1 - \frac{{\alpha_1}^2 {r_{eq}}^2}{4}\right) - i\alpha_1 r_{eq}\left(1 - \frac{{\alpha_1}^2 {r_{eq}}^2}{8}\right)\cos\phi_{eq}, \tag{24}$$

$$U_{sh} = i{\alpha_1}^2\left(\frac{1}{4} + \cos\phi_{eq}\right)\sqrt{\frac{\pi}{2\alpha_1 r_{eq}}}\exp\left\{-i\left(\alpha_1 r_{eq} - \frac{1}{4}\pi\right)\right\}, \tag{25}$$

where $\alpha_1 = (\omega/c)rN = \beta r$, ω is the signal frequency, r the radius of the cylinder, N the refractive index, and c the speed of light. In the above equations r_{eq}, and ϕ_{eq} are the apparent or equivalent radius of the cylinder by which the incident signal is scattered back and the apparent surface angle of the cylinder respectively. To illustrate the physical meaning of apparent radius and apparent surface angle and to derive the expressions for them, we consider the following problem.

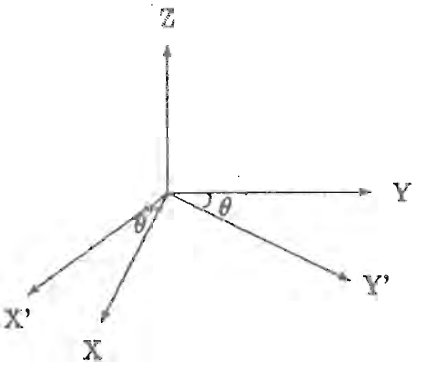


Figure 5. Rotation about Z' axis

Let the original state of the sensor is specified by the axis X'Y'Z'. When it is rotated by an angle θ about X' axis as shown in Figure 5, the new state it defined by the axis XYZ, where XYZ is related to X'Y'Z' by Euler transformation as given in Equation (26).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \tag{26}$$

If the radius of the cylinder is r, and its surface angle, which is measured along the cylinder surface from the reference position, is ϕ, then the apparent or equivalent radius r_{eq}, by which the incident wave is scattered back, and the apparent surface angle ϕ_{eq} are found to be

$$r_{eq} = r \tag{27}$$

$$\phi_{eq} = \phi - \theta. \tag{28}$$

The sensor(Doppler radar) measures the magnitude and phase of scattered fields. Here in deriving the estimator we use only the magnitude and phase measurements of the scattered magnetic field. We assume that the magnitude measurement A_m, and the phase measurement B_m are corrupted by white noise η_1 and η_2 with variances ${\sigma_1}^2$ and ${\sigma_1}^2$ respectively. Therefore the measurements are given by

$$A_m = {\alpha_1}^2 \left(\frac{1}{4} + \cos\phi_{eq}\right) \sqrt{\frac{\pi}{2\alpha_1 r_{eq}}} + \eta_1 \tag{29}$$

$$B_m = \frac{3\pi}{4} - \alpha_1 r_{eq} + \eta_2. \tag{30}$$

We now proceed to develop an appropriate state estimator to determine the radius r of the rotor, and its angle rotation θ using maximum likelihood criterion. The estimated values of r and θ, $\hat{r}$ and $\hat{\theta}$ respectively, are found to satisfy

$$A_m - \frac{\omega N \hat{r}}{c} \sqrt{\frac{\omega N \pi}{2c}} \left(\frac{1}{4} + \cos(\phi - \hat{\theta})\right) = 0 \tag{31}$$

$$\left[\frac{2}{{\sigma_2}^2}\left(B_m - \frac{3\pi}{4}\right) + \frac{\omega^2 N^2 \pi}{2c^2 {\sigma_1}^2}\left(\frac{1}{4} + \cos(\phi - \hat{\theta})\right)^2\right] \hat{r} + \frac{2\omega N}{c{\sigma_2}^2}\hat{r}^3 - \frac{A_m}{{\sigma_1}^2}\sqrt{\frac{\omega N \pi}{2c}}\left(\frac{1}{4} + \cos(\phi - \hat{\theta})\right) = 0. \tag{32}$$

Therefore the state estimator finds the approximate values of r, and θ using the above two equations. We assume that the statistical properties of the noise terms are known. The analytical solutions to these equations can also found to be

$$\hat{r} = \frac{1}{2}\sqrt{\frac{(3\pi - 4B_m)c}{\omega N}} \tag{33}$$

$$\cos(\phi - \hat{\theta}) = \frac{2A_m c}{\omega N}\sqrt{\frac{2}{\pi(3\pi - 4B_m)}} - \frac{1}{4}. \tag{34}$$

Hence we have derived the state estimator which will estimate the rotation and the radius of a rotating cylinder from the magnetic field scattered back by it.

5 MICROWAVE CIRCUIT ANALYSIS AND DESIGN

Future microwave system requirements will include increased RF power, higher frequencies, wider bandwidth, lower noise levels and smaller higher density packaging at lower cost. Microwave systems are used in satellite communications, navigational equipment for ships, radar systems and phased-array modules for information transfer systems. GaAs FETs monolithic circuits and hybrid circuits are of importance in the state-of-the-art developments of microwave technology. Mathematical modelling of computer-aided design helps to improve performance and reduce costs. The monolithic approach promises cost effective mass production of very large number of modules (10^6), and the hybrid approach also provides small sized models in small quantities but more versatile. Both circuits use GaAs devices. Advance4d modelling and computer simulation, measurement and design optimization are invaluable tools to realize these technologies.

For computer aided design, it is essential to obtain accurate device s-parameter measurements from the manufacturer. The parameters depend to a large extent on the test mount used for characterization (for example, the S_{12} value for the transistor may be considerably modified by feedback within the test mount). Typical values of the S_{11}, S_{12}, S_{21}, and S_{22}, parameters at 12GHz for 50Ω characteristic impedance input and output lines are respectively: $0.614\angle 107$, $0.026\angle -86.6$, $1.218\angle -69$, $0.85\angle -165.8$ (50Ω mount). A low-noise microwave amplifier consists of RF input to

circulator, low noise interstage amplifier, PIN attenuator, circulator, high gain interstage amplifier, circulator connected to RF output. To design a single state low-noise amplifier (Figure 6) we need the models of the following circuit elements:

1. $p1$ lumped 150Ω resistor
2. $p2$ 87Ω, 90° impedance transmission line
3. $p3$ 50Ω, 60° impedance transmission line
4. $p4$ 50Ω, 130° impedance transmission line
5. $p5$ FET device model with S parameters

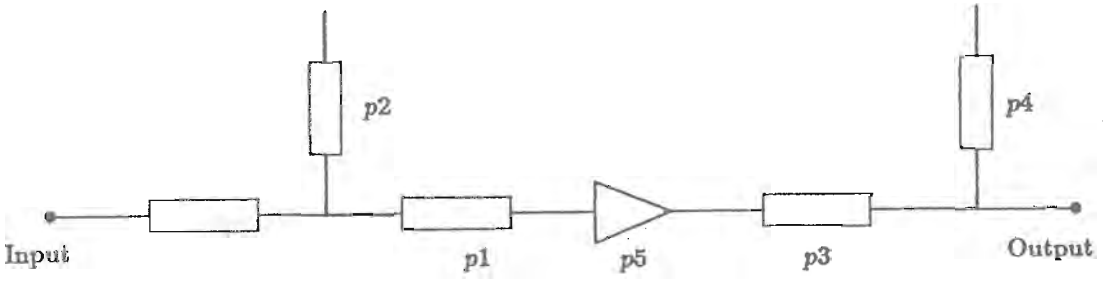


Figure 6(a) Low-noise amplifier

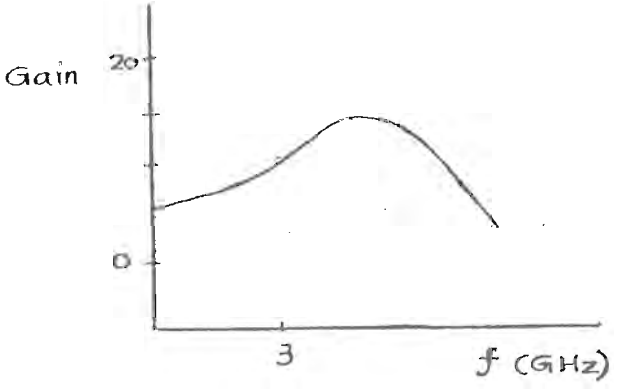


Figure 6(b) Frequency domain computer simulation result

6 Modelling Nonlinear Magnetic Devices

Electronic control circuits are often designed to drive nonlinear devices such as micromotors. In order to achieve an efficient circuit, where the losses in the motor are minimized, we need an accurate model of the nonlinear circuit elements of the motor. Current circuit simulation packages are designed to handle linear R, L and C circuit elements or where nonlinearities are introduced, rather crude models are employed. For instance, the B-H curve is not only linearised, but the hysteresis loss, often very significant in fast switching circuits, are ignored. We shall present here an elementary circuit model of nonlinear magnetic devices driven by a sinusoidal source. Work is currently underway to develop a more generalized circuit model.

When an external field H_e is applied to a magnetic material, the flux density

$$B = \mu_0 H_e + N B_e = \mu_0[1 + N B_e/(\mu_0 H_e)] H_e = \mu H_e, \tag{35}$$

where N is the number of magnetic domains. Differentiating (35) and re-arranging we get

$$\begin{aligned} dB/dt &= (\mu_0 + B_e \partial N/\partial H) dH/dt + B_e(\partial N/\partial X) dx/dt \\ &= \mu'(dH/dt) + B_e(\partial N/\partial X)v, \end{aligned} \tag{36}$$

$$\tag{37}$$

where μ is the reversible permeability and v is the velocity of the domain movement. The magnetic field H_d due to the domain movement is

$$H_d = (1/s) B_e(\partial N/\partial X)v, \tag{38}$$

and the resultant field

$$\begin{aligned} H &= H_e + H_d \\ &= B/\mu + (1/s)[dB/dt - \mu' dH/dt], \end{aligned} \tag{39}$$

$$\tag{40}$$

where $s = [dB/dt - \mu' dH/dt]/H$, the hysteresis coefficient. Typical values for a ferrite are $\mu = 100$, $\mu' = 0.01$, $s = 2000$.

For sinusoidally time varying fields, using $d/dt = j\omega$, we obtain from (39)

$$\mu = B/H = \mu_R(\omega) + j\mu_I(\omega), \tag{41}$$

where

$$\mu_R(\omega) = \mu \frac{s^2 + \omega^2 \mu \mu'}{s^2 + \omega^2 \mu^2} \tag{42}$$

$$\mu_I(\omega) = -\frac{\omega s(\mu - \mu')}{s^2 + \omega^2 \mu^2}, \tag{43}$$

where μ and μ' are permeabilities defined in Figure 7(a). Note that the model accounts for the hysteresis loss as well.

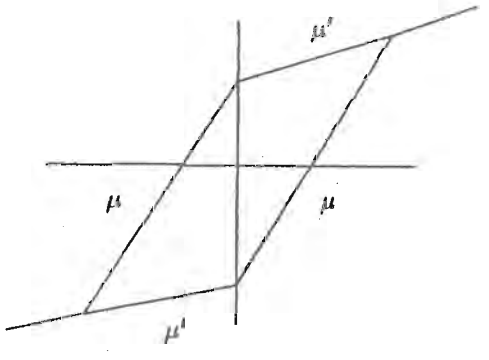


Figure 7(a)

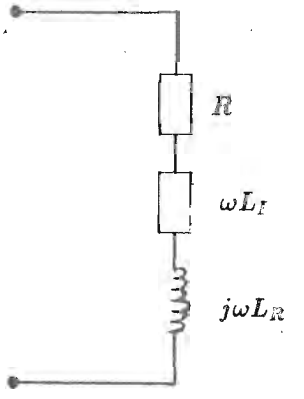


Figure 7(b)

For circuit analysis, the magnetic device is represented by an impedance Z.

$$\begin{aligned} Z &= R + j\omega L \\ &= R + j\omega(L_R - j\omega L_I) \\ &= (R + \omega L_I) + j\omega L_R \qquad (44) \\ L = \mu(\omega)/\mu_0 L_0 &= \mu_R(\omega/\mu L_0 - j\mu_I(\omega)/\mu_0 L_0 = L_R - j\omega L_I. \qquad (45) \end{aligned}$$

The ωL_I term represents the loss due to hysteresis L_0 is the d.c. inductance and R the resistance of the winding. The equivalent circuit of the nonlinear magnetic device is shown is Figure 7(b).

7 REFERENCES

1. Wijekoon, N. A. B., Hoole, P. R. P.: 'PC Based Analysis of High Impedance Faults: Mathematical Formulations'. Submitted to IEEE Transactions on Power.

2. Wijekoon, N. A. B., Hoole, P. R. P., Hoole, S. R. H.: 'A Novel Observation on Negative Sequence Fields in Synchronous Machines and a Modified Circuit Model'. Proceedings of the International Symposium on Nonlinear Phenomena, Japan, 1992.

3. Samarakoon, K. B., Ekanayake, J. B., Hoole, P. R. P., Hoole, S. R. H.: 'Modelling Nonlinear Actuators for Electronics Control Circuit Design Packages'. Proceedings of the International Symposium on Nonlinear Phenomena, Japan, 1992.

4. Vidthiyani, S. S., Kirubarajan, T., Hoole, S. R. H., Hoole, P. R. P.: 'A Novel State Estimator for Rotation and Tilts in Helicopters'. Proceedings of the International Symposium on Nonlinear Phenomena, Japan, 1992.

5. Saotine, H., Hayano, S., Miyazak, A., and Saito, Y.: 'A Representation of Magnetization Characteristics for Computational Magnetodynamics'. Proceedings of the International Symposium on Nonlinear Phenomena, Japan, 1992.

NA-12-189

Identifying Chaos in Electrical Circuits Using Computer Simulation Techniques

T. Kirubarajan
Department of Computer Sciences
Faculty of Engineering
University of Peradeniya
Peradeniya
Sri Lanka.

ABSTRACT

Nonlinear phenomena observed in electrical circuits and previously neglected as deterministic noise have now been found to be nonlinearities which could be analysed both theoretically and numerically. Chaos is a special type of nonlinearity frequently encountered in many analogue electrical circuits. In this paper a coherent scheme to check whether an electrical circuit is chaotic using computer simulation techniques is described. The simulation techniques will be useful in the design stage in that circuits positively identified as chaotic can be redesigned so as to eliminate the chaotic behaviour.

1 INTRODUCTION

Until very recently, linear circuit theory used to be applied for all electrical circuits and an electrical engineer did not use to think much about possible nonlinear characteristics of circuits when designing electrical systems. Even if an electrical system works well initially, due to component aging and wear and tear, it might become nonlinear and respond unpredictably. The distortion of signals and the generation of harmonics in some electrical systems when they have been in operation for long are due to such non-linear characteristics. In the past, such effects were neglected as deterministic electrical noise.

The possibility of nonlinear characteristics suggests that the steady state system response can be chaotic - i.e., the system can respond unpredictably. The limit set or the steady-state response of a non-linear system can either be an equilibrium point, or a periodic solution or a quasi-periodic solution that is the sum of many periodic solutions. Apart from these limit sets there exists another one - chaos. Chaos can be defined as none of the above, that is, as bounded steady-state behaviour that is not an equilibrium point, not periodic and not quasi-periodic. The fundamental nature of a chaotic system is the sensitive dependence on initial conditions: given two distinct initial conditions arbitrarily close to one another, the trajectories emanating from these initial conditions diverge, at a rate characteristic of the system until they become completely uncorrelated. Chaos is identified in many forms and in many man-made and natural systems including almost all branches of Engineering. Simple chaotic electrical systems include randomly blinking neon bulbs, computers printing out streams of random data, electric motors suddenly spinning out of control, etc.

To avoid the malfunctioning of electrical circuits, it is necessary to predict chaos using either laboratory experiments, computer simulations or mathematical proofs. If such identification in the early stages of designing is possible, then the system designer can avoid the design parameter values for which chaos is positively identified. Therefore one needs a coherent scheme that allows to foresee and avoid chaos in electrical circuits.

To identify whether a system is chaotic, a number of analytical methods are available [1] [2] [3]. However, most practical electrical systems are so nonlinear that it is impossible to study them analytically. Then, computer simulations provide the only mechanism to analyse such systems.

With the advent of cheap but powerful personal computers, it has become possible to carry out numerically intensive studies on nonlinear systems. In this paper a coherent scheme to analyse nonlinear systems and to identify chaos is presented. Using a third order nonlinear system, which roughly simulates a controller for an electrical motor, the method is illustrated.

2 COMPUTER SIMULATIONS

2.1 System Definition

To illustrate the chaos identification method, we use a third order system that contains a dead-zone relay element and simulates an electric motor control system. The sample system is given in Figure 1. The relay element in the system corresponds to a current amplifier that saturates at $\pm a$ Amps. The scheme applies mathematical proofs, simulates the third order system using computer simulation techniques, and compares the simulation results with the characteristics of well known chaotic systems.

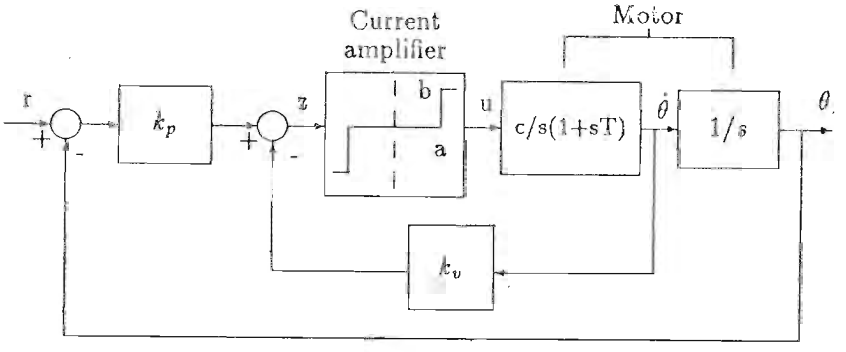


Figure 1. A third order nonlinear system

Since computer simulation methods are essentially numerical, we need to specify the system dynamics as a set of first order differential equations. If we denote the states θ, the velocity $\dot{\theta}$, and acceleration $\ddot{\theta}$ by x_1, x_2, and x_3 respectively and if the reference input is zero, then the system equations are given by

$$\dot{x}_1 = x_2 \tag{1}$$

$$\dot{x}_2 = x_3 \tag{2}$$

$$\dot{x}_3 = -\frac{1}{T}x_3 + \frac{c}{T}u, \tag{3}$$

$$\text{where} \quad u = \begin{cases} 0 & (\| z \| < a) \\ b \cdot \mathrm{sgn}(z) & (\| z \| \geq a) \end{cases}, \tag{4}$$

$$\text{and} \quad \mathrm{sgn}(z) = \begin{cases} 1 & (z < 0) \\ -1 & (z > 0) \end{cases}. \tag{5}$$

Any simulation program which can solve first order differential equation can be used for the simulation. We use MATLAB which solves first order ODEs using the fourth and fifth order, variable step-size Runge-Kutta method. Though the mechanism provided by MATLAB to solve the equations is very slow, MATLAB is chosen because it can be used very easily to implement the other algorithms to analyse the system. Alternatively, one can use any other program which would give a faster mechanism to solve the equations, for example RegSim or NAG, and then import the solution into MATLAB for further analysis. For the simulations the following values

are chosen: $c = 3$, $T = 0.1$, $k_p = 15$, $a = 5$, $b = 10$, and $r = 0$. The MATLAB file which defines the system dynamics is given in the Appendix.

2.2 Trajectories

The variation of time waveforms of $x = [x_1 \ x_2 \ x_3]^T$ with time and those of trajectories in the phase plane and their dependence on initial conditions are important in determining the behaviour of any system. Figure 2 gives the waveforms for a set of initial conditions. They exhibit some anomalous characteristics which we would not expect from an electrical motor. When the simulations are done for different sets of initial conditions, it is found that the magnitudes and the positions of the pulses and oscillations which characterize the waveforms vary unpredictably when the initial conditions are slightly perturbed from their original values.

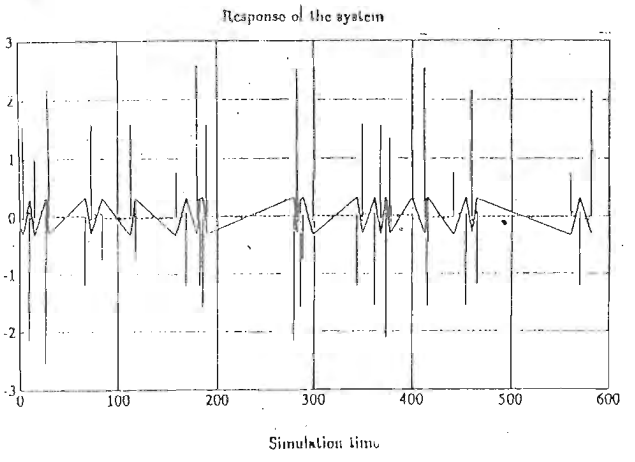


Figure 2. Waveforms with initial conditions [-.01 -0.05 -0.1]
($—\ x_1$ ..... x_2 - - - x_3)

It is not possible to predict the steady state behaviour of the system for a set of given initial conditions and the only way to find out the steady state response is to simulate the system for the required initial condition. The is the fundamental characteristic of any chaotic system [4] [5]. This behaviour could be explained from a theoretical point as well. Theoretically the system has infinitely many equilibrium points $x_{eq} = [x_{1eq} \ 0 \ 0]^T$, where $-\frac{1}{3} \leq x_{1eq} \leq \frac{1}{3}$. None of the equilibrium points are attracting and the steady state behaviour of the system cannot be a steady motion.

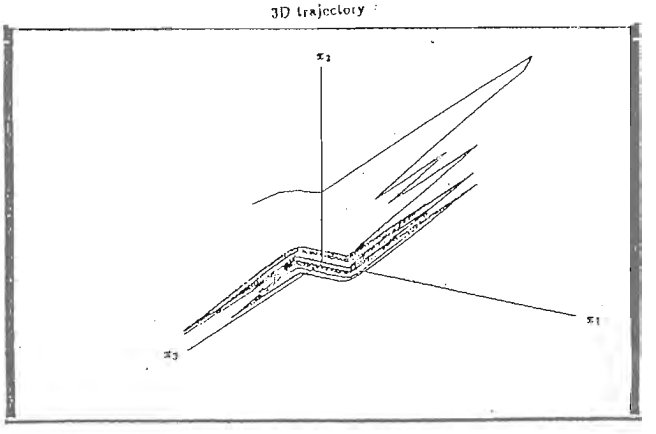


Figure 3. 3D trajectory with initial conditions [-0.5 1.0 0.5]

cycles known as chatter. The spectra of limit cycles have spikes at harmonics of the fundamental frequency indicating the periodicity.

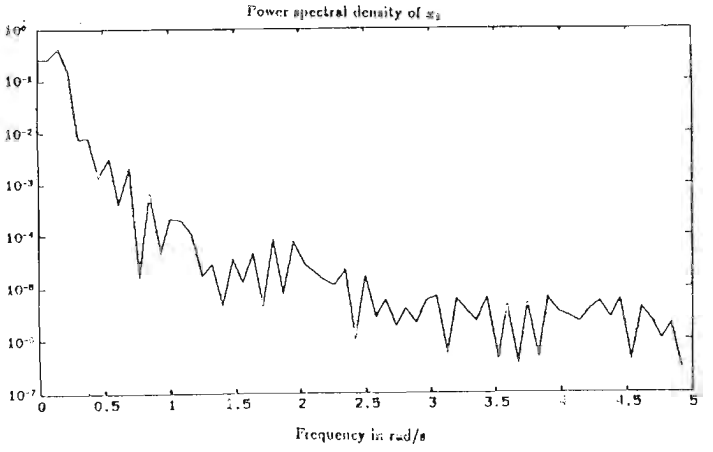


Figure 5. PSD of x_1 for initial conditions [0 0.04 0]

2.4 Poincaré Map

Poincaré map(PM) replaces the flow of a third order continuous system with a second order discrete system. It ensures that its limit sets correspond to the limit sets of the underlying flow and bridges the gap between continuous and discrete systems [6]. Also the order reduction resulting from the Poincaré method is another useful feature. PM gives the intersection of the trajectory with any given plane. To obtain the Poincaré map, a two dimensional hypersurface Σ, which divides the state space into two regions, Σ_+ for which the dot product $h \cdot (x - x_\Sigma)^T > 0$ and Σ_- for which $h \cdot (x - x_\Sigma)^T < 0$, where h is the normal vector to the plane Σ and x_Σ is an arbitrary point lying on Σ. If Σ is chosen properly, then the trajectory under observation will repeatedly pass through Σ crossing from Σ_- into Σ_+ etc. as illustrated in Figure 6.

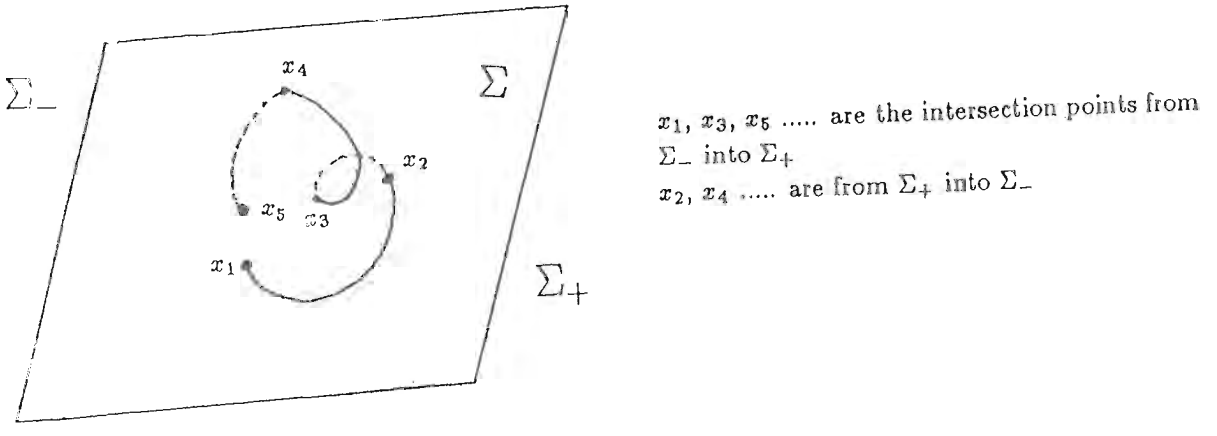


Figure 6. Two sided Poincaré map

Using the sign variations in the dot products between successive iterations, it is possible to extrapolate the point of intersection with Σ and thus obtain the PM. If the product of the dot products is negative, this indicates an intersection between the trajectory and the plane in either

direction. A sample PM on the plane whose normal vector is given by [1 1 1] and which passes through [0 0 0] is given in Figure 7.

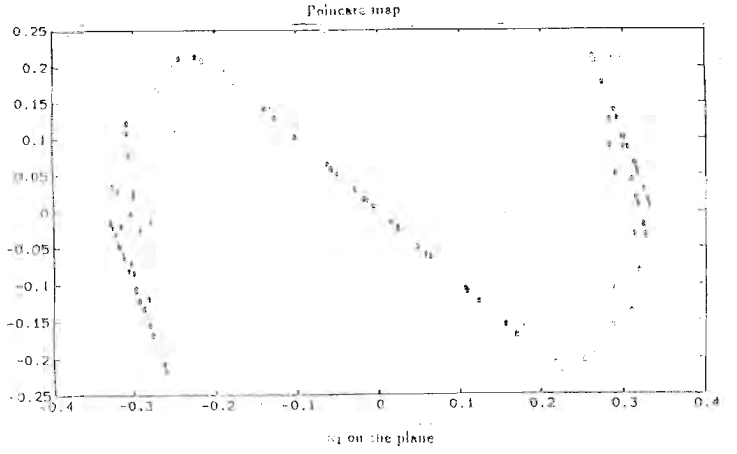


Figure 7. PM on plane [1 1 1] through [0 0 0] with initial conditions [-0.1 -0.05 -0.1]

It is clear from the plot that the attractor has a fine, distinctive structure. The shape(not the values) remains the same for different sets of initial conditions and on different plane. As more and more points are added, there appear to be layers within layers. The distorted S shape of the attractor can be compared with that of Duffing's equation, which is a well known chaotic system [6]. This similarity across scale is characteristic of chaotic systems and is known as *fractals* [4]. The layers represent the stretching and folding of the attractor which in the electrical system can correspond to friction or damping.

2.5 Dimension of the Attractor

The dimension of a dynamical system is the number of state variable that are used to describe the dynamics of the system. The dimension of a non-chaotic limit set is always an integer while a strange attractor has a non-integer dimension. The state space definition does not allow non-integer values. Hence an alternative definition for the dimension of a limit set is needed. There are a number of definitions of dimension of a strange attractor. Correlation dimension, capacity dimension and information dimension are some of them. We consider correlation dimension which is representative of other similar definitions.

To calculate the dimension of an attractor consider the following problem: cover the attractor with spherical volume elements each with diameter ϵ. Let $N(\epsilon)$ be the minimum number of such volume elements needed to cover the attractor. Correlation dimension D_c which is of probabilistic nature is defined as

$$D_c = \lim_{\epsilon \to 0} \left(\frac{\ln \sum_{i=1}^{N(\epsilon)} P_i^2}{\ln \epsilon} \right) \tag{6}$$

where P_i is the relative frequency with which a typical trajectory enters the i^{th} volume element. If N points of the trajectory have been collected by simulation and the corresponding correlation is defined as

$$C(\epsilon) = \lim_{N \to \infty} \frac{1}{N^2} \{ \text{ the number of pairs of points } (x_i, x_j)_{i \neq j} \text{ such that } \| (x_i - x_j) \| < \epsilon \} \tag{7}$$

it can be proved that the correlation dimension is given by

$$D_c = \lim_{\epsilon \to 0} \frac{\ln C(\epsilon)}{\ln \epsilon}. \tag{8}$$

However, from Figure 3 we can see that there exists a region of state space enclosing all possible equilibrium points which is large enough so that no trajectory leaves the region: all initial conditions outside this region evolve into this region and remain inside for all subsequent time. This behaviour can be compared with that of Lorenz attractor [4]. The 3D trajectory exhibits another feature common to many chaotic systems: the trajectories evolve as a random mixture of two types of behaviour. The trajectory loops for a while in the region $x_1 > 0$ and then passes over to the region $x_1 < 0$ where it loops for a while before returning to the first region. The number and size of loops between transitions follow no set pattern and they appear to be random. This behaviour can be compared with that of Lorenz attractor.

The system shrinks towards some final asymptotic structure. The volume shrinking nature is explained by the fact that the system is dissipative. The divergence of the system is found to be negative and therefore the ensemble of states will constantly decrease in volume. The dissipative nature of the system is observed in Figure 3 as well.

2.3 Probability Density and Power Spectrum

Since the waveforms of the system vary randomly in steady state, it is necessary to analyse the probability distribution(PD) of the values of x obtained after successive iterations. Figure 4 shows the probability distribution(subjected to a scalar factor) that the value of x_1 lies in a particular bin during a sampling interval. Only the values for which $\| x_1 \| < \frac{1}{3}$ are considered.

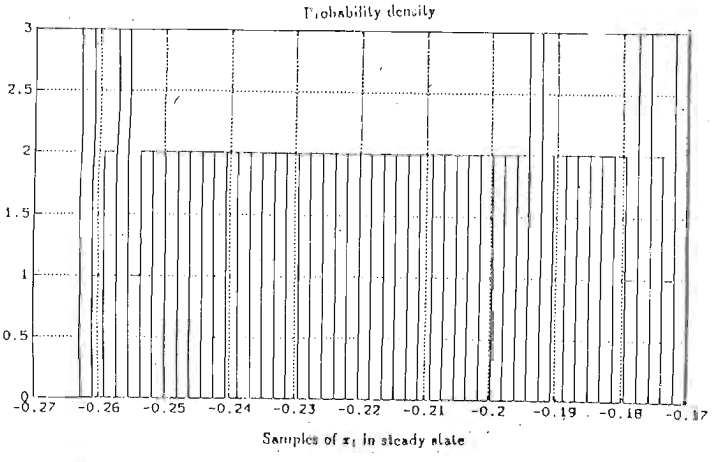


Figure 4. PD of x_1 for initial conditions [-10 5 -2]

When the system is simulated for different sets of initial conditions, the PD plots reveal that the distribution is either purely random and does not conform to any known distribution or it is very close to uniform distribution. The uniform distribution is an important feature in that, it confirms that the equilibrium point can lie anywhere so that $\| x_{1_{eq}} \| < \frac{1}{3}$. As explained in Section 2.2 this characteristic is expected for this system. The equiprobable nature or even the pure random nature of probability distribution suggests that the system can be chaotic in that it is impossible to predict its behaviour in steady state. Had the distribution been, say, Gaussian, as is the case in many random processes found in nature, the behaviour of the system could have been predicted to some reasonable extent.

A sample power spectral density(PSD) of x_1 calculated using a 128 point FFT and Hamming window method is given in Figure 5. The noise-like spectrum is characteristic exhibited by all chaotic systems. In addition to the broad-band component, they contain spikes indicating the predominant frequencies of the solution. Figure 5 also shows the low amplitude, high frequency limit

A plot giving the value of D_c is given in Figure 8. The slope in the linear region varies between 1.3 and 1.6. The slopes slightly depend on the value of N. As N increases, i.e., as the attractor is clearly defined the slope becomes smoother. Clearly the correlation dimension D_c given by the slopes is not an integer and hence this suggests that the limit set can be chaotic.

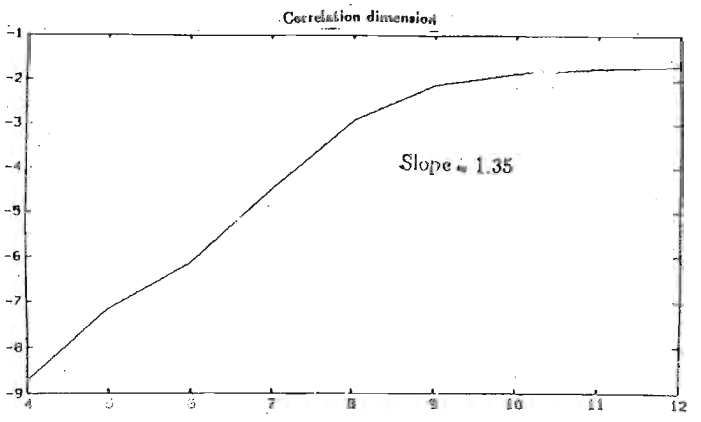


Figure 8. Correlation dimension of the attractor for initial conditions [0 0.04 0]

Dimension is a lower bound on the number of degrees of freedom of the attractor and the number of state variables to describe the dynamics of the system. A correlation dimension of $1.3 < D_c < 1.6$ implies that the limit set has two degrees of freedom and that the motion of the attractor can be modelled by a second order system. This observation is consistent with that made in the 3D trajectory plot given in Figure 3. It can be seen in Figure 3 that the trajectories move along an arbitrary plane while completing each loop. For that plane one component of the state remains constant throughout a loop. Therefore the trajectory has only two degrees of freedom while going around a loop.

2.6 Liapunov Exponents

If we denote current state at time t given that the initial condition is x_0 as $X_t(x_0)$, the system equation can be written as

$$\dot{X}_t(x_0) = f(X_t(x_0)), \qquad X_0(x_0) = x_0. \tag{9}$$

Differentiating (9) with respect to x_0 we get

$$\frac{\partial \dot{X}_t(x_0)}{\partial x_0} = \frac{\partial f(\dot{X}_t(x_0))}{\partial X_t(x_0)} \cdot \frac{\partial X_t(x_0)}{\partial x_0}, \qquad \frac{\partial X_t(x_0)}{\partial x_0} = I. \tag{10}$$

If we define $\Phi_t(x_0) = \partial X_t(x_0)/\partial x_0$, then equation (10) becomes,

$$\dot{\Phi}_t(x_0) = \frac{\partial f(X_t(x_0))}{\partial X_t(x_0)} \cdot \Phi_t(x_0), \qquad \Phi_t(x_0) = I, \tag{11}$$

and this is known as the variational equation [4].

If for the third order system under consideration $m_1(t), m_2(t)$ and $m_3(t)$ are the eigen values of $\Phi_t(x_0)$, the Liapunov or the characteristic exponents λ_i for $i = 1, 2, 3$ of x_0 are given by

$$\lambda_i = \lim_{t \to \infty} \frac{1}{t} \ln \| m_i(t) \|, \qquad i = 1, 2, 3 \tag{12}$$

whenever the limit exists. Liapunov exponents indicate the average rate of contraction if $\lambda_i < 0$ or expansion if $\lambda_i > 0$ of the trajectory in a particular subspace near a particular limit set. Let λ_i be arranged such that $\lambda_1 \geq \lambda_2 \geq \lambda_3$.

Using the signs of the Liapunov exponents it is possible to identify the nature of a limit set. One of the Liapunov exponents of an autonomous system unless the limit set is an equilibrium point is always zero. Sensitive dependence occurs only in an expanding trajectory. Therefore chaotic limit sets have a positive exponent. By definition chaotic system is bounded and therefore the sum of the exponents has to be negative. Hence the necessary but sufficient condition for a continuous system to be chaotic is that it must at least be of order three and its Liapunov exponents must satisfy the condition that $\lambda_1 > 0, \lambda_2 = 0, \lambda_3 < 0$ and $\lambda_3 < -\lambda_1$ [3] [6]. When the values of the Liapunov exponents are calculated numerically, they are found to satisfy the above condition. Therefore one could conclude that the system is chaotic.

2.7 Some Practical Considerations

Though the methods illustrated above using a third order system as an example positively indicate that the system is chaotic, they may not reveal the true nature of the steady state system response. By definition a chaotic system is greatly sensitive to initial conditions. This implies that an arbitrarily small error eventually affects the macroscopic behaviour of the system. Truncation and round-off errors are inherent to any numerical algorithm. Hence numerical simulation may not reflect the actual behaviour of the system it simulates. This aspect is very important here, since the system under consideration is very sensitive to initial conditions. The best accuracy with which MATLAB can solve the first order ODEs over reasonable time is ± 0.01. Hence it has to be assumed that the system is insensitive to variations less than 0.01 in magnitude in the initial conditions. Similar arguments will apply for other systems.

Since digital computers are discrete-time in nature, an integration algorithm models a continuous-time system by a discrete-time system. Thus the actual system being simulated in not a differential equation but a difference equation. Therefore, even if the computer simulations exhibit chaos, the real system which the simulations represent may not do so in practice. Because of this the results obtained are to be interpreted carefully. Another problem is that for a given accuracy (or local truncation error) a larger step size is taken with the fourth and fifth order Runge-Kutta method than with a method utilizing lower order approximations. Therefore the simulations become more discrete.

The values of different definitions of dimension are used to predict chaos. However, it is impossible for a single number such as, for example, correlation dimension, to describe the complex structure of a limit set. The result depends on N, the number of points considered and this is finite contradicting the original definition of dimension. Therefore the values obtained as dimensions have to be used in conjunction with other experimental results in deciding the nature of the limit set.

3 CONCLUSIONS

Many computer simulation techniques to identify chaos in an electrical system have been discussed in this paper. Using these techniques, it should be possible to test for any exotic limit sets like chaos before fabricating an electrical circuit. Although high computing power is required to apply the above techniques for a circuit of very high order, they provide a coherent tool, particularly for analogue circuits, by which one can foresee and eliminate highly non-linear characteristics by changing the system parameters. Such techniques also create an awareness among electrical engineers of the possibility of nonlinear characteristics and chaos in cases where such behaviour is not expected.

4 APPENDIX

```
function xdot=motor(t,x)

% # The m-file motor.m #

% For the 3D system explained in Section 2.1,
% the equations simulate an
% electric motor containing relay.
% r, kp, kv, T, a, b and c should be global variables
% z and u are local variables.

z = kp*(r-x(1)) - kv*x(2) ; % input to relay

% simulate relay
if abs(z) < a, u = 0 ;
   elseif z >= a, u = b ;
   else u = -b ;
end

% define the equations
xdot(1) = x(2) ;
xdot(2) = x(3) ;
xdot(3) = -(1/T)*x(3) + (c/T)*u ;

xdot = xdot(:); % forces xdot to be a column vector
```

5 REFERENCES

1. Brockett, R. W.: 'On Conditions Leading to Chaos in Feedback Systems'. Proceedings of IEEE Conference on Decision and Control, IEEE Press, New York, 1982.

2. Sparrow, C. T.: 'Chaos in a Three-Dimensional Single Loop Feedback System with a Piecewise Linear Feedback Function'. Journal of Mathematical Analysis and its Applications, Volume 83 (1982), pp. 275-291.

3. Ruelle, D.: 'Chaotic Evolution and Strange Attractors'. Cambridge University Press, Cambridge, 1989.

4. Deavany, R. L.: 'Chaotic Dynamical Systems'. The Benjamin/Cummings Publishing Co., Inc., California, 1986.

5. Thompson, J. M. T., and Steward, H. B.: 'Nonlinear Dynamics and Chaos'. John Wiley and Sons, London, 1986.

6. Parker, T. S., and Chua, L. O.: 'Practical Numerical Methods for Chaotic Systems'. Springer-Verlag New York, Inc., New York, 1989.

Category theoretic modelling of digital circuits and systems

M.S.J. Johnson and R.F.C. Walters

School of Mathematics and Computing, Macquarie University
and School of Mathematics and Statistics, Sydney University
AUSTRALIA

Abstract

This paper is a review of theoretical work being carried out in Sydney which is aimed at developing a new foundation for models of circuits and systems. The work is based on *category theory* which is a branch of pure mathematics which has been shown to be valuable in a wide range of applications in mathematics, computer science and physics. One of the principal goals is the development of a calculus to aid in the specification, refinement and verification of digital circuits and systems.

1 Introduction

Recently category theory has been widely recognised as a valuable tool for modelling computational systems. Most of the work has been focussed on problems in theoretical computer science including concurrency, denotational semantics and functional programming (see for example [2]). Among the few examples of lower level analyses are the work at Glasgow in hardware design and verification (Sheeran [16] [17] and Murphy [15]) and the work at Calgary in hardware verification [3].

In Sydney we have taken the view that the low and high level work should be integrated — if the mathematical models are at the appropriate level of generality then circuit modelling techniques should work whether the devices in the circuit are elementary semiconductor devices, logic gates, purpose built hardware (perhaps implementing some particularly useful software), or individual processors interacting in a parallel processing environment.

This document reports the development of category theoretic modelling techniques which work at both the digital circuit (low) level and at the digital systems (high) level.

The theory has been used at the systems level to model information systems' specification in consulting work with Telecom Australia, and has proved very valuable. Applications at the low level are now being considered and we are working on developing appropriate categorically based design tools to assist hardware specification.

In the following sections we will review some elementary notions from category theory and then outline:

- The modelling of combinational circuits in categories with products; the modelling of control systems in categories with coproducts; and a formulation of the mathematical duality between computation and control (Kasangian and Walters [11]).
- The theory of *distributive categories*. Distributive categories have the structure required to integrate high level control and low level computation (Walters [18] [21] and Cockett [5]).
- The mathematical notion of *imperative program* which uses state based descriptions in a distributive category to model digital systems (Walters [20] and Khalil and Walters [12]).
- The use of *profunctions* to provide standard techniques for dealing with the complexity of state based descriptions (by data-hiding) and for constructing larger models from given modules (Johnson and Walters [10]).
- An example of the use of distributive categories in the specification of digital systems (Dampney, Johnson and Monro [6] and Johnson [9]).
- High dimensional models of computation based on n-categories (Johnson [7] and [8]).

The main thrust of the Sydney work, originally pioneered by Walters, is the use of carefully controlled state space descriptions of circuits and systems, using ideas motivated by the mathematical treatment of dynamical systems in category theory.

It should be remembered that most modern modelling work evolves through three phases:

1. The development, if necessary, of appropriate mathematical theory
2. The implementation of software tools to assist in calculations in the theory
3. The tuning of the tools for the particular application problem domain of interest.

The work reported here falls squarely in the first phase, although there is now work in Sydney on phase 2. Nevertheless the work has been carried into a

number of particular application domains (phase 3 and Section 7 reports one example of this.

Special note: In the printed version of this paper not all of the above areas have been thoroughly treated but full details are available from the authors. We have also excerpted liberally from our previously published works and encourage the reader to refer to them for further details as required.

2 Category theory

The standard textbook on category theory is [14], which could be consulted for the many details not presented here. There are several newer books including Barr and Wells [2] and Walters [21] which introduce category theory via its applications to computer science. These latter may be more appropriate for readers with a background in computing or electrical engineering.

A *category* consists of a set of *objects* and a set of *morphisms* (also called *arrows*). A morphism f from the object X to the object Y is written $f : X \to Y$. Morphisms $f : X \to Y$ and $g : Y \to Z$ compose associatively to give a morphism $g \circ f : X \to Z$ and there are identity morphisms for the composition.

In particular instances of categories the arrows are often functions of some sort: There is the category Set whose objects are sets and whose arrows are functions, and there are many other examples of categories of mathematical objects and appropriate functions including groups with group homomorphisms as arrows, topological spaces with continuous maps as arrows, and vector spaces with linear maps as arrows. However the abstract definition makes no limitations on the nature of arrows — they simply must have some associative composition and specified identity arrows. Thus there are examples where the arrows are not functions, including any preordered set with elements as objects and an arrow $x \to y$ if and only if $x \leq y$; and the category of matrices with natural numbers as objects, $n \times m$ matrices as arrows $m \to n$ and with matrix multiplication as composition.

One of the important insights of category theory is that constructions in categories can be defined in terms of so called *universal properties* which are properties principally involving arrows. This shifts the emphasis of the mathematics from looking at the internal structure of objects to looking at the arrows between objects and it has resulted in much systematization and simplification. It seems that the essence of a construction is often captured by a universal property rather than by the details of how to build the constructed object.

Two basic constructions in categories are *products* and *coproducts*. The product of two objects A and B is written $A \times B$ and generalizes the notion of cartesian product of sets. The product in Set is essentially the cartesian product, the product in the category of groups and homomorphisms is the pointwise product, and the product in a preordered set is the least upper bound (if it exists). More generally there may be products of families of objects. The co-

product of A and B is written $A + B$ and generalizes the notion of disjoint union (discriminated union) of sets. The universal properties that are used to define products and coproducts are *dual* which means that one property can be obtained from the other by reversing the arrows. This shows the precise relationship between products and coproducts and allows proofs involving say coproducts to be *dualized* to give proofs involving products. We will see later that a similar duality applies between computation and control.

Two other dual categorical notions are the *initial object*, denoted here by O, and the *terminal object*, denoted here by I. Initial and terminal objects satisfy particularly simple universal properties: In a category C, an object I is terminal when for each object X of C there is a unique arrow from X to I. In fact a terminal object is a special case of a product — I satisfies the universal property of the product of the empty family of objects (and dually for initial objects and empty coproducts). See one of the standard texts for details.

It should be noted that the existence of products, coproducts and other objects satisfying universal properties is not guaranteed. Many categories contain most of the objects that we could hope for — they are *finitely complete* (have products of finite families and other *finite limits*) and *finitely cocomplete* (have coproducts of finite families and other *finite colimits*). Other categories contain few limits or colimits.

Subcategories are obtained by taking some of the objects and some of the arrows of a category, but of course they must be closed under composition and have all the necessary identities. The *full subcategory* determined by a collection of objects has as objects just those, and as arrows all arrows between those objects.

Computer scientists often consider objects as types: O corresponds to an empty type and I corresponds to a type containing only one thing. The object $I+I$ (the coproduct of I and I) is often called B, and corresponds to the Boolean type with two elements called *true* and *false*. There are two canonical arrows $I \rightarrow I + I$ which will also be called *true* and *false*. Many categories, including finitely complete categories, allow the following construction of subtypes: if $F : X \rightarrow I + I$ is a morphism then there is a subtype (or *subobject*) of X written as $\lceil F \rceil \rightarrowtail X$ which intuitively is the subtype of X containing those elements a of X such that $F(a)$ is true. There is also the subtype $\lceil \neg F \rceil$ containing those elements a of X such that $F(a)$ is false. Subtypes of products $G : X \times Y \rightarrowtail I + I$ play an important role and may be identified with *relations* between X and Y.

3 Computation and control

It is a remarkable fact that just as products and coproducts are dual, so too are combinational circuits and loop-free control systems (for example loop-free flow charts). The precise dual models are very simple, but the important challenge is to get their specification just right. The surprising thing is that two apparently

independent aspects of models of computation obey exactly the same, but for duality, laws. We will briefly outline the relationship.

A combinational circuit can be viewed as a Boolean function $B^n \to B^m$ for some n and m where $B = I + I$ is a two point set. Let **Comb** be the full subcategory of **Set** determined by the B^n for all natural numbers n. It is easy to check that this category has all finite products. It is also easy to demonstrate that any function $B^n \to B^m$ can be obtained from the function NAND : $B^2 \to B^1$ by using the universal property of products. To see that this is just the usual completeness of NAND we need to investigate which constructions the universal property of products provides.

The universal property of a product $X \times Y$ with projections $p_1 : X \times Y \to X$, $p_2 : X \times Y \to Y$ says that for any object Z and any pair of arrows $f : Z \to X$ and $g : Z \to Y$ there is a unique arrow $(f, g) : Z \to X \times Y$ such that $p_1 \circ (f, g) = f$ and $p_2 \circ (f, g) = g$. From this we can deduce the existence of a number of arrows including for each X, $\Delta_X : X \to X \times X$ ($f = g = Id_X$); for each X and Y, *twist* : $Y \times X \to X \times Y$ ($f = q_2$ and $g = q_1$ where q_1 and q_2 are the projections of $Y \times X$); and for each $h : X' \to X$, $k : Y' \to Y$, a map $h \times k : X' \times Y' \to X \times Y$ ($f = h \circ p'_1$ and $g = k \circ p'_2$). These amount in turn to fanning one wire or bus out onto two; crossing a pair of wires or buses; and putting two combinational circuits in parallel, and these, together with composition in the category which amounts to putting circuits in series, are the usual operations permitted in constructing combinational circuits.

In other words the decompositions of arrows of **Comb** into arrows constructed from say NAND by use of the universal property of products, are specifications of combinational circuits to compute the given function.

The constructions available in tracing the flow of control in a system are precisely dual to those provided in constructing circuits. In a circuit two wires give two values, a product. In a control system two flows represent alternative possibilities, a coproduct. The Δ for circuits corresponds to the $\nabla : X + X \to X$ which amounts to "forgetting" which X has control and simply returning the value of X (this is what happens when flows of control converge). Finally *twist* : $X + Y \to Y + X$ is just a cross of flows of control.

For more details see Kasangian and Walters [11].

We will see in Section 4 how control and computation can be modelled together and in Section 5 how the categorical models of circuits and control can be extended to clocked non-combinational circuits and flow charts with feedback.

4 Distributive categories

Section 3 illustrates the modelling of computation in categories with products and control in categories with sums. Of course most hardware has aspects of computation *and* control. In this section we integrate the previous models by

developing *distributive categories*. Distributive categories have been shown to provide a good model at all levels, from digital circuit design to complex software systems.

Distributive categories were so named by Walters [18] who demonstrated their importance in modelling datatypes. In an early (1967) note Lawvere [13] pointed out the significance of the distributive law in such constructions as '*if ... then ... else*'. Others who have studied distributive categories include Arbib and Manes [1] and Cockett [4]. There is an abundance of examples of distributive categories since any cartesian-closed category with sums, and hence any topos, is a distributive category.

In a category with (assigned) finite products the following operations exist as a consequence of the universal property of products. Given an object X and a terminal object I there is a unique arrow $!_X : X \to I$. Associated with a product $X \times Y$ there are the projection arrows $p_1 : X \times Y \to X$, $p_2 : X \times Y \to Y$. Given an object X there is a diagonal arrow $\Delta_X : X \to X \times X$. Given arrows $f_1 : X_1 \to Y_1$, $f_2 : X_2 \to Y_2$ there is the product of the two arrows $f_1 \times f_2 : X_1 \times X_2 \to Y_1 \times Y_2$. Given two objects X, Y there is an arrow $twist_{X,Y} : X \times Y \to Y \times X$.

The dual operations in a category with (assigned) finite sums are denoted respectively $!_X : O \to X$ (O the initial object); the injections $i_1 : X \to X + Y$, $i_2 : Y \to X + Y$; the codiagonal $\nabla : X + X \to X$; the sum of two arrows $f_1 + f_2 : X_1 + X_2 \to Y_1 + Y_2$, and $twist_{X,Y} : X + Y \to Y + X$.

Definition 1 A category is *distributive* if it has assigned finite sums and products, and the following arrows are isomorphisms:

$$!_{X \times O} : O \longrightarrow X \times O \qquad \text{and}$$

$$\nabla_{X \times (Y+Z)}((1_X \times i_1) + (1_X \times i_2)) : X \times Y + X \times Z \longrightarrow X \times (Y + Z).$$

Example 2 The category **Sets** of sets and functions is a distributive category.

The one-point set $\{*\}$ is terminal. The empty set is initial. The product of two sets is the cartesian product. The sum is the disjoint union; we denote the elements of $X + Y + Z$ by $(x, 0)$, $(y, 1)$ or $(z, 2)$ depending in which component of the sum the element lies. The projections of the product, the diagonal arrow, the injections of the sum are the obvious functions suggested by their names. The codiagonal function ∇_X is given by

$$\begin{array}{rccc} \nabla_X : & X + X & \longrightarrow & X \\ & (x, 0) & \longmapsto & x \\ & (x, 1) & \longmapsto & x \end{array}$$

The function $twist_{X,Y} : X \times Y \to Y \times X$ takes (x, y) to (y, x); the function $twist_{X,Y} : X + Y \to Y + X$ takes $(x, 0)$ to $(x, 1)$ and $(y, 1)$ to $(y, 0)$.

Finally the isomorphism of the distributive law is given by:

$$\begin{array}{ccc} X \times Y + X \times Z & \longrightarrow & X \times (Y + Z) \\ ((x, y), 0) & \longmapsto & (x, (y, 0)) \\ ((x, z), 1) & \longmapsto & (x, (z, 1)) \end{array}$$

Remark 3 As remarked in Section 2 we denote $I + I$ as B to indicate that $I + I$ is the Boolean algebra of truth values. We will often denote the truth value *false* by 0, and *true* by 1.

Remark 4 Since 1989 the theory of distributive categories has been developed along a number of lines. There are many special classes of distributive categories that play roles in particular applications but for our purposes the basic notion will (mostly) suffice.

5 State based models of digital systems

In this section we show how state spaces can be used to model simple digital systems, and in the next section we introduce tools which assist in the modelling of much more complex systems. In this section we will concentrate on the modelling of imperative programs, but these programs include clocked digital circuits with devices as built-in functions and arbitrary flows of control with decisions and operations as built-in functions. This section is taken from Walters [20].

Briefly, imperative programming can be seen to be the construction of dynamical systems from a given set of built-in data types and functions using the operations available in a distributive category. The behaviours of a program are the behaviours (orbits) of the dynamical system. We will present here some informal state space models of imperative programs and refer the reader to Walters [20] and Khalil and Walters [12] for formal definitions.

We begin by describing a simple special case – *isolated* imperative programs; that is, imperative programs with no input.

The notion of imperative program is always relative to some given class of *built-in functions*.

Definition 5 An *isolated imperative program* $\mathcal{P}$ in a language with a given class of built-in functions is a set X, called the state space of the program, and a function $f : X \to X$, constructed from the built-in functions using only the operations of a distributive category — that is, identities, composition, product, sum, and the distributive law. A behaviour of the imperative program is an initial state $x_0 \in X$ and the sequence of states obtainable from that state by repeated application of f; that is the sequence:

$$x_0 \mapsto f(x_0) \mapsto f^2(x_0) \mapsto f^3(x_0) \mapsto \ldots$$

The following example should make clear the intention of this definition.

Example 6 A program to calculate $n!$ in terms of built-in functions *multiply* : $\mathbf{Z} \times \mathbf{Z} \to \mathbf{Z}$, *subtract_1* : $\mathbf{Z} \to \mathbf{Z}$, and $test_{x>0} : \mathbf{Z} \to B \ (= I + I)$.

Take the state space (= the space of global variables) to be $X = \mathbf{Z} \times \mathbf{Z}$ ($\mathbf{Z}$ the integers). We will denote an element of X by (p, k); our intention is that p stand for the partial product and k for the decreasing factor.

Take the initial state to be $x_0 = (1, n)$, n a positive integer. Then the function $f : X \to X$ of the imperative program is given by

$$f(p,k) = \begin{cases} (p \cdot k, k-1), & (k > 0) \\ (p,k), & (k \leq 0) \end{cases}$$

The behavior of the program beginning with initial state $(1, n)$ is the sequence

$$(1,n) \mapsto (1 \cdot n, n-1) \mapsto (1 \cdot n \cdot (n-1), n-2) \mapsto \cdots (n!, 0) \to (n!, 0) \mapsto \cdots$$

That is, for all sufficiently large m, we have $f^m(1,n) = (n!, 0)$ which is what is meant by saying that the program calculates $n!$.

We have not completed the argument. The function $f : X \to X$ must be expressed in terms of the built-in functions using the operations of a distributive category — in fact, the expression of the function in this form *is* the program. But f can be expressed as the following composite:

$$\begin{array}{ccccccc}
\mathbf{Z}^2 & \xrightarrow{1_Z \times \Delta_Z} & \mathbf{Z}^3 & \xrightarrow{1_{Z^2} \times test_{x>0}} & \mathbf{Z}^2 \times (I+I) & \xrightarrow{\cong} & \mathbf{Z}^2 + \mathbf{Z}^2 \\
(p,k) & \longmapsto & (p,k,k) & \longmapsto & \begin{cases} (p,k,0) & (k \leq 0) \\ (p,k,1) & (k > 0) \end{cases} & \longmapsto & \\
\end{array}$$

$$\begin{array}{ccccc}
\xrightarrow{1_{Z^2}+g} & & \mathbf{Z}^2 + \mathbf{Z}^2 & \xrightarrow{\nabla} & \mathbf{Z}^2 \\
\longmapsto & \begin{cases} (p,k,0) & (k \leq 0) \\ (p \cdot k, k-1, 1) & (k > 0) \end{cases} & \longmapsto & \begin{cases} (p,k) & (k \leq 0) \\ (p \cdot k, k-1) & (k > 0) \end{cases} &
\end{array}$$

where $g : \mathbf{Z}^2 \to \mathbf{Z}^2$ is the composite

$$\mathbf{Z}^2 \xrightarrow{1_Z \times \Delta_Z} \mathbf{Z}^3 \xrightarrow{mult \times 1_Z} \mathbf{Z}^2 \xrightarrow{1_Z \times sub\_1} \mathbf{Z}^2$$

The next two examples are not programs, but functions which may be used in the construction of programs.

Example 7 Any function g between finite sets may be constructed using the operations available in distributive categories, without using any given functions, as follows:

$$m \cdot I \cong I + I + \cdots + I \xrightarrow{i_{g(1)} + i_{g(2)} + \cdots + i_{g(m)}} n \cdot I + n \cdot I + \cdots + n \cdot I \xrightarrow{\nabla} n \cdot I$$

This means, in particular, that any Boolean function $B^k \to B^l$ is available, and hence tests may be constructed from given tests by Boolean operations.

Example 8 Given two functions $f, g : X \to X$ and a test $t : X \to B$ we can construct the function which takes x to $f(x)$ if the test $t(x)$ is satisfied, else it takes x to $g(x)$, as follows:

$$X \xrightarrow{\Delta} X \times X \xrightarrow{1_X \times t} X \times B \xrightarrow{\cong} X + X \xrightarrow{g+f} X + X \xrightarrow{\nabla} X.$$

Of course, this was used in the program in Example 6.

Using Examples 7 and 8 and appropriate data types, it is clearly possible to construct the usual control structures used in programming.

Definition 9 An *imperative program* with input alphabet Σ is a set X, the state space, and to each element $a \in \Sigma$ a function $f_a : X \to X$ constructed out of the built-in functions using the operations available in a distributive category.

As a first example let us construct an imperative program, out of the same built-in functions as in Example 6, which allows a positive natural number to be input and which calculates the factorial function.

Example 10 Take the state space to be $X = \mathbb{Z}^2 \times (I + I) \cong \mathbb{Z}^2 + \mathbb{Z}^2$. We will call the two components $\mathbb{Z}^2$ of the state space the 0-*mode* and the 1-*mode* of the program. Take the alphabet to be $\Sigma = \{clock\} \cup \mathbf{N}$, ($\mathbf{N}$ the natural numbers).

By the universal property of sums, a function $h : X \to X$ corresponds to two functions $h_0, h_1 : \mathbb{Z}^2 \to X$. So we need to construct functions $f_{clock,0}$, $f_{clock,1}$, $f_{n,0}$, $f_{n,1}$ ($n = 1, 2, 3, \cdots$). (The index indicates the mode.) The required functions are

$$\begin{aligned}
f_{clock,0}(p,k) &= \begin{cases} (p \cdot k, k-1, 0), & (k > 0) \\ (p, k, 1), & (k \leq 0) \end{cases} \\
f_{clock,1}(p,k) &= (p, k, 1) \\
f_{n,0}(p,k) &= (p, k, 0) \\
f_{n,1}(p,k) &= (1, n, 0)
\end{aligned}$$

The intention is that in 0-mode only the clock input is enabled, and the program computes $n!$. When the computation is finished control is passed to the numerical input. If a number is input the control is passed immediately back to the clock input.

It is clear from our experience with Example 6 that these functions can be constructed from the built-in functions using the constructions available in a distributive category.

Example 11 Example 7 shows that if the state space X is finite all possible functions are available to construct a program. Hence any finite state machine can be modelled by an imperative program. No built-in functions are necessary — only the operations of a distributive category.

6 Profunctions

State space models are often avoided because they are unwieldy, principally because the calculus of state space descriptions has not been well understood. In this section we describe profunctions and record some of their basic properties. Profunctions were introduced by Khalil and Walters [12] (under the name pseudofunctions) to aid in the development of programs (and this section follows closely their development). They capture clearly the calculus of state space descriptions, and have been used by Johnson and Walters [10] to provide a new semantics of programming languages. Software support tools for calculations with profunctions are currently being developed. The tools will then be used as the foundations for a circuit analysis CAE package.

Definition 12 Let $\mathbf{C}$ be a distributive category. An arrow $\phi : X + U + Y \longrightarrow X + U + Y$ in $\mathbf{C}$ is said to *idle* in Y if $\phi \circ i_3 = i_3$, where i_3 is the injection of Y into $X + U + Y$.

A *profunction* or *functional process*, ϕ, from a set X to a set Y, denoted $\phi : X \rightsquigarrow Y$, is an function $\phi : X + U + Y \to X + U + Y$ which idles in Y such that for each $x \in X$ there exists a natural number n_x with $\phi^{n_x}(x) \in Y$.

The set U will be referred to as the set of *local states* of the profunction ϕ.

Proposition 13 *(Profunctions yield functions)* *Let $\phi : X \rightsquigarrow Y$ be a profunction, then there is a function $\overline{\phi} : X \to Y$ defined by $\overline{\phi}(x) = \phi^{n_x}(x)$, the function obtained by iterating ϕ.*

Proposition 14 *(Functions are profunctions)* *For any arrow $f : X \to Y$ in* **Set** *the arrow $f' = \nabla_{X+Y} \circ (i_2 \circ f + i_2) : X + Y \to X + Y$ is a profunction such that $\overline{f'} = f$.*

Proposition 15 *(Profunctions compose)* *If $\alpha : X \rightsquigarrow Y$ and $\beta : Y \rightsquigarrow Z$ are profunctions, with local states in U and V respectively, then*

$$\alpha ; \beta = (1_{X+U} + \beta) \circ (\alpha + 1_{V+Z}) \quad : \quad X + W + Z \longrightarrow X + W + Z$$

is a profunction from X to Z, with local states in $W = U + Y + V$, and with the property that:

$$\overline{\alpha ; \beta} = \overline{\beta} \circ \overline{\alpha}.$$

Proposition 16 *(Case and parallel profunctions)* *If $\phi : X \rightsquigarrow Y$ and $\psi : X' \rightsquigarrow Y'$ are profunctions with local states U and U' respectively, then*

1. $\phi \vee \psi = a^{-1} \circ (\phi + \psi) \circ a$ *where*

$$a = (1_{X+U} + twist_{(X'+U'),Y} + 1_{Y'}) \circ (1_X + twist_{X',U} + 1_{U'+Y+Y'})$$

is a profunction from $X + X'$ *to* $Y + Y'$ *with local states in* $W = U + U'$, *and with the property that:*

$$\overline{\phi \vee \psi} = \overline{\phi} + \overline{\psi}.$$

2. $\phi \wedge \psi = b^{-1} \circ (\phi \times \psi) \circ b$ *where*

$$b = \delta_2 \circ (\delta_0 + 1_{(U \times (X'+U'+Y')} + \delta_1)$$

(and δ_0, δ_1, δ_2 *are distributive law arrows) is a profunction from* $X \times X'$ *to* $Y \times Y'$ *with local states in* $W = (X \times (U' + Y')) + (U \times (X' + U' + Y')) + (Y \times (X' + U'))$, *and with the property that:*

$$\overline{\phi \wedge \psi} = \overline{\phi} \times \overline{\psi}.$$

Corollary 17 *("Universal properties" for products and coproducts)* *For* $i = 1, 2, \ldots, n$ *if* $\phi_i : X_i \rightsquigarrow X$ *and* $\psi_i : X \rightsquigarrow X_i$ *are profunctions; then there exist profunctions* $\phi : X_1 + X_2 + \cdots + X_n \rightsquigarrow X$ *and* $\psi : X \rightsquigarrow X_1 \times X_2 \times \cdots \times X_n$ *such that*

$$\begin{aligned} \bar{\phi}(x) &= \overline{\phi_i}(x), \quad \text{if } x \in X_i, \\ \bar{\psi}(x) &= (\overline{\psi}_1(x), \overline{\psi}_2(x), \ldots, \overline{\psi}_n(x)). \end{aligned}$$

Proof. Take

$$\begin{aligned} \phi &= (\phi_1 \vee \phi_2 \vee \ldots \vee \phi_n) ; \nabla \\ \psi &= \Delta ; (\psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_n). \end{aligned}$$

Proposition 18 *(Iteration of profunctions)* *Given a profunction* $\mu : X \rightsquigarrow X + Y$ *and a function* $ord : X \to \mathbb{N}$, *such that if* $\mu(x) \in X$, *then* $ord(\mu(x)) < ord(x)$; *then there exists a profunction* $\nu : X \rightsquigarrow Y$ *such that*

$$\overline{\nu}(x) = \overline{\mu}^{m_x}(x), \quad \text{for some } m_x \leq ord(x) + 1$$

Proof. Let U be the set of local states of the profunctions μ, and take

$$\nu = (\nabla_X + i_{U,X} + 1_Y) \circ (1_X + twist_{U,X} + 1_Y) \circ \mu$$

where the local states of ν are in $W = U + X$.

This collection of results shows that profunctions admit the constructions expected for a calculus of state spaces. The adequacy of the calculus of profunctions is further demonstrated by the work of Johnson and Walters [10] which uses this calculus to give a state-based semantics for programming languages (resolving the disjunction between operational and denotational semantics). That work is a systems application of a technique which is effective at both the systems' and circuits' levels.

7 Modelling information systems

The theory developed in the preceding sections has been used at the systems application level with some success. One of the areas that has been investigated with particular success is the methodology of designing information systems, especially using what has been called the *Entity-Relationship-Attribute approach (ERA)* (see Dampney, Johnson and Monro [6] and Johnson [9]). This section records informally some observations from that work.

Designing an information system, like designing a clocked digital circuit, is essentially about specifying a universal algebra. However, because of the need to specify computation and control the appropriate *syntactic category* or *theory* must be a distributive category. In fact, in order to obtain the full query language automatically one must take a finitely complete distributive category.

Next, to take the state-based view of the system in operation we need to understand the interaction between the *static* specification and the *dynamic* model of the system. This interaction is not yet fully worked out, but some preliminary results appear in [6].

The principal issue identified in the investigation of the relationship between the static and dynamic models is the modelling of *constraints*. Often constraints have been left implicit in information models. However, when a model or a part of a model is to have several instantiations or "views" for which compliance must be checked it is necessary to make the constraints explicit, and important changes in the methodology have resulted from this need.

Like most aspects of an information system, constraints themselves can be classified as *static* or *dynamic*.

A static constraint is a constraint which will never change and which must be satisfied for the life of the information system. Such constraints, although often left implicit in specifications, are frequently apparent as commutative diagrams in the static model. This has led to an important change in information systems CASE tools — they are beginning to require the specification of commutative diagrams rather than leaving the constraints to be determined by the perceived semantics and added in an ad hoc fashion.

Of course many constraints are not static. For example constraints which require that only certain classes of employee carry out certain tasks must be changed with changes in management directives or in industrial awards. Such constraints must be modelled as entities in an information model so that they can themselves be updated as changes occur. Although there had been attempts to incorporate these entities in some commercial systems, they had not been successful because of the limitations of the mathematical theory on which the models were based, principally the failure to work in a distributive category and the absence of limits.

The universal properties which are available in distributive categories with finite limits are important in specifying constraints because sometimes the entities required for a constraint are not present in the information system spec-

ification, simply because they can be calculated by the query language when required. Nevertheless, we may need to use these entities to specify constraints at the design stage, long before the query language is added.

The work has led to a slogan in information systems specification: All structural (as opposed to numeric) constraints should be determined at design time by the introduction of limit entities and the specification of commutative diagrams.

The immediate future of this work is the development of better CASE tools incorporating the category theoretic model, and the quest for a full understanding of the interaction of the static and dynamic models.

8 Higher dimensional models

An alternative modelling technique which is used in Sydney is the use of n-categories [7] [8]. For $n = 2$ these higher dimensional categories have long been used in category theoretic investigations. Recently n-categories for $n > 2$ have been found to be valuable in developing *coherence theorems* and these theorems are now recognised as important in transformations of chip designs. Coherence theorems were also important in the foundations of the work on the computation-control duality (see [19, Section 3]). However, these models will not be treated in detail here.

Acknowledgement

The authors gratefully acknowledge the support of the Australian Research Council (ARC).

References

[1] M.A. Arbib and E.G. Manes, *Algebraic approaches to program semantics*, Springer Verlag, (1986).

[2] M. Barr and C.Wells, *Category theory for computing science*, Prentice Hall, (1990).

[3] G. Birtwhistle and J.R.B. Cockett, personal communication, May 1990.

[4] J.R.B. Cockett, Distributive logic, Research Report CS-89-01, Department of Computer Science, University of Tennessee, (1989).

[5] J.R.B. Cockett, Introduction to distributive categories, *Macquarie Computing Reports*, 90-052C (1990), 31pp.

[6] C.N.G. Dampney, M.S.J. Johnson and G.P. Monro, A mathematical foundation for ERA, *Proc of the Institute for Mathematics and its Applications* (1991), 77–84.

[7] M.S.J. Johnson, The combinatorics of n-categorical pasting, *Journal of Pure and Applied Algebra* **63** (1989), 211–225.

[8] M.S.J. Johnson, Linear term rewriting systems are higher dimensional string rewriting systems *Proc of the Institute for Mathematics and its Applications* (1991), 101–110.

[9] M.S.J. Johnson The Telecom T.I.M.E. model, consultant's report, (1991).

[10] M.S.J. Johnson and R.F.C Walters, Profunctions and program semantics, in preparation.

[11] S. Kasangian and R.F.C. Walters, The duality between flow charts and circuits, *Bull. Austral. Math. Soc.* **42** (1990), 71–79.

[12] W. Khalil and R.F.C. Walters, An imperative language based on distributive categories II, *University of Sydney Pure Mathematics Research Reports* **91-29** (1991), 17pp.

[13] F.W. Lawvere, Theories as categories and the completeness theorem, *Journal of Symbolic Logic* **32** (1967), 562.

[14] S. Mac Lane, *Categories for the working mathematician,* Springer Verlag, (1971).

[15] D. Murphy, Formal design of regular architectures, *Algorithms and parallel VLSI architectures II,* (1991), North Holland, 6pp.

[16] M. Sheeran, Describing and reasoning about circuits using relations, *Proceedings of the workshop on theoretical aspects of VLSI design* (1990), CUP.

[17] M. Sheeran, Categories for the working hardware designer, *Hardware specification, verification and synthesis: Mathematical aspects* LNCS **408** (1989), 380–402.

[18] R.F.C. Walters, Datatypes in distributive categories, *Bull. Austral. Math. Soc.* **40** (1989), 79–82.

[19] R.F.C. Walters, The free category with products on a multigraph, *Journal of Pure and Applied Algebra* **62** (1989), 205–210.

[20] R.F.C. Walters, An imperative language based on distributive categories, to appear in *Mathematical Structures for Computer Science.*

[21] R.F.C. Walters, *Categories and Computer Science*, Carslaw Publications (1991), to appear, Cambridge University Press (1992).

School of Maths & Computing
Macquarie University
N.S.W. 2109
AUSTRALIA

School of Maths & Stats
University of Sydney
N.S.W. 2006
AUSTRALIA