Gallium Arsenide Bit-Serial Integrated Circuits

by

S. C. K. Lam

A thesis submitted to the Faculty of Science,
University of Edinburgh,
for the degree of
Doctor of Philosophy

Department of Electrical Engineering
1990

Abstract

Bit-Serial architecture and Gallium Arsenide have essentially been mutually exclusive fields in the past. Digital Gallium Arsenide integrated circuits have increasingly adopted the conventional approach of bit-parallel structures that do not always suit the properties and problems of the technology. This thesis proposes an alternative by using a least significant bit first bit-serial architecture, and presents a group of "cells" designed for signal processing applications.

The main features of the cells include the extensive use of pseudo-dynamic latches for pipelining, modularity, and programmability. The logic circuits are mainly based on direct-coupled FET logic. They are also compatible with silicon ECL circuits. The target clock rate for these cells are 500MHz, at least ten times faster than previous silicon bit-serial circuits. The differences between GaAs and silicon technologies meant that the cells were designed from circuit level upwards.

Further to these cells, a multi-level signaling scheme has been developed to substantially alleviate off-chip signaling. Synchronisation between signals is simplified, improving even further on the conventional bit-serial system, especially at the high bit-rates encountered in GaAs circuits.

For on-chip signals, a single phase clock scheme has been developed for the GaAs cells, which maintains the low clock loading and high speed characteristics of the pseudo-dynamic cells, while substantially simplifying clock distribution and generation. Two novel latch designs are proposed for this scheme.

Test results available have already proved the concepts behind the two-phase clocking scheme, the latches, and the multi-level signaling scheme. Further tests are taking place to establish their speed performance.

Declaration of Originality

This thesis was composed entirely by myself. All the circuit design work described in this thesis, including the software modifications in Chapter 4, were carried out solely by myself, when I was a member of a research group in the Department of Electrical Engineering, University of Edinburgh.

Acknowledgement

The author is most grateful to my Supervisor, Dr. Martin Reekie, whose help, inspiration, and patience, have made it possible to finish this work. I am very grateful to my second Supervisor, Dr. Brian Flynn, who gave me a great deal of help and support over the years. I would also like to thank Professor John Mavor, whose kind support in extending my contract has enabled me to complete my thesis.

Numerous colleagues in the Department have helped me in the past. In particular, I would like to thank Bob Mhar and Dr. C. H. Lau, for their help and friendship. Thanks are also due to Dr. Alan Murray for help with the diagrams, to Mrs. Caroline Burns for providing photocopying facilities, and to Bruce Hassall and David Stewart for keeping the computers going.

Finally, I would like to thank Dr. Ross McTaggart and Honeywell for circuit fabrication and testing.

[redacted]

S. C. K. Lam

Contents

Chapter 1

Introduction

This thesis is the result of two and a half years of research carried out in this Department on gallium arsenide (GaAs) digital ICs. The main objectives of this work were to design a library of cells, based on GaAs MESFET technology, and combine it with a bit-serial architecture to produce highly modular and efficient integrated circuits for signal processing. The original work in this thesis can be divided into three main parts: bit-serial GaAs cells, multi-level signaling, and single-phase latches. The cell designs were partly based on direct-coupled field effect transistor logic (DCFL), and adopted a unified bit-serial approach, tightly pipelined logic, and psuedo-static latches.

Although this project was essentially based on GaAs metal-semiconductor field effect transistor (MESFET) technology, the circuit techniques and system architecture are also applicable to modulation doped field effect transistors (MODFETs) or high electron mobility transistors (HEMTs) technologies. MODFETs are GaAlAs heterojunction FETs that are very similar to GaAs MESFET technology. Converting MESFET based circuits for MODFET/HEMT fabrication only requires minor modifications at the circuit level. Some of the cells in this work were appropriately modified for fabrication (Chapter 5), illustrating their flexibility.

To summarise the scope of this project, a library of cells had been designed and an FIR filter was used as a case study of integrating these cells. Work was done to incorporate a suitable model into SPICE2G.6. The verification of the designs were mostly done by the SPICE simulations. Some of the cells were fabricated at a late stage of the project and some testing is still under way in the United States. A test IC was fabricated in NMOS in silicon to demonstrate the special registers used for the cells. A single wire 3-level chip-to-chip signaling technique was developed to minimise synchronisation problems of off-chip signals without sacrificing throughput. A two wire scheme, each having 3 voltage levels, was also investigated as an extension of the concept, leading towards radix-8 data representations.

With a project of such a time span, a brief history of the events that led to its present and final form is appropriate. Initial interests in digital GaAs ICs in this Department stemmed from work done by Dr. A. David Welbourn and Dr. Brian W. Flynn in 1983. An SERC funded project was initiated in 1986, with Standard Telecommunications Laboratories (STL) agreeing to provide free fabrication. Owing to a management commercial decision at STL in 1987, the processing facilities were no longer available. (This also applied to other research projects related to STL at the time.) Subsequent to this setback, limited unofficial links with Honeywell Sensors and Signal Processing Laboratory (SSPL) were obtained. Delays in obtaining mask insertion meant that in the present state of this project, some of the cells were fabricated in a non-standard Honeywell MESFET process (Chapter 5), and they will be fully assessed at SSPL. Export restrictions also meant that the test ICs are unlikely to be available for full assessment in this country within the time available. However, the low speed test results obtained so far have been positive and support the simulations. The test results are included in an Appendix at the end of this thesis. The circuits that have not been fabricated are therefore supported by simulation results only, using the modified SPICE2G.6 program, discussed in Chapter 4.

GaAs has always been considered a high risk material for making ICs simply because of its possible obsolescence from competing compound materials and from the well established Si ECL technology. The arguments on GaAs versus Si have continued if not heightened during the course of this project and Chapter 2 will highlight some of the recent developments in high speed ICs and make observations on the future of GaAs in the various application areas reviewed. Research in bit-serial computation in silicon are also reviewed in Chapter 2. The Chapter ends with a discussion on the reasons for combining GaAs and bit-serial techniques.

Chapter 3 provides an introduction to basic GaAs DCFL design. The similarities and important differences between DCFL and silicon NMOS are highlighted, allowing silicon designers to transfer to DCFL design, and provide background to the circuits discussed in later Chapters.

In Chapter 4, the operation of the GaAs MESFET, device structure, and its modelling are discussed in more detail. Its operation is compared to the common silicon MOSFET and JFET devices to further emphasise its different properties and problems. A Honeywell model and its incorporation into the SPICE FORTRAN source code, and comparison with measured data, are discussed in detail.

The rest of the thesis centres on the original circuit design work. Chapter 5 describes in detail the adopted bit-serial architecture, the design of the register structure, and the GaAs cells. The layouts of the cells as multi-project chips and their simulations are also shown. Modifications for a real MODFET process were also discussed to show the flexibility of the designs. To illustrate the application of the cells, a brief case study of an FIR filter is then discussed.

In Chapter 6, a novel 3-level voltage signaling technique and its associated decoding/encoding circuitry are described. This approach was also extended into the 2-wire radix-8 technique mentioned above and its properties and circuitry are discussed in detail. Timing limitations of two-phase clocks systems are substantially reduced by single-phase clocking. Two unique single-phase pipelining latch designs are proposed in this Chapter.

Finally, the thesis is concluded in Chapter 7 by identifying the limitations of the GaAs cells as they stand, and suggest possible future extensions of this work. The results and achievements of this work are also summarised.

Chapter 2

Introduction to GaAs and Bit-serial circuits

The two main areas of interest in this work, GaAs and bit-serial circuits, had previously been separate major subjects in their own right, and a fair amount of literature exist in both areas. The scope and structure of this survey is defined by the following objectives: to introduce the basic concepts of GaAs and bit-serial circuits, review previous important developments in each field, and focus on the idea of combining GaAs technology and bit-serial techniques.

The survey consists of three main sections: GaAs integrated circuits, bit-serial circuits, and a final discussion. In each of the first two main sections, initial discussions cover some historical aspects of the subject, fundamental concepts, and their advantages and disadvantages compared to other technologies or circuit architectures. In the case of GaAs, the basic concepts will also include some material and device properties, and common circuit configurations. They serve to provide a silicon designer familiar with metal-oxide-semiconductor FET (MOSFET) and junction FET (JFET) technologies with the basic background to "transfer" to GaAs devices. More detailed discussions of device operation, fabrication, and modelling are dealt with in Chapter 4. Recent and important developments are discussed to indicate possible future trends of research in each area. These will include some products and systems available commercially or under development. They are especially relevant to GaAs, owing to its continuing "high risk" designation and limited acceptance within the semiconductor industry, and the continuous threat from silicon to its survival.

Finally, the third section of this chapter makes observations based on the previous sections concerning the problems of GaAs VLSI utilisation in systems, and puts forward the case for combining bit-serial architecture and GaAs as an alternative, thus laying the basic foundation of this work.

2.1. GaAs integrated circuit technology

Since the early 60's, GaAs has been known to possess certain electrical properties which make it attractive for three main categories of integrated circuits: high speed digital, analogue, and microwave [71,127]. The second type covers wide bandwidth operational amplifiers, and data converters. Switched-capacitor filters have also recently been of interest [141]. The third type covers microwave amplifier stages, mixers, and oscillators, and are often called monolithic microwave integrated circuits (MMICs). While noting the significant overlap in device modelling and fabrication between the three categories, this thesis shall concentrate on digital circuits. Wherever appropriate, comparisons are made to silicon circuits to highlight the state of development and viability of GaAs in a particular area.

In the next sub-section, the basic properties of GaAs materials, devices, and common circuit configurations are first introduced to provide the background for later discussions.

2.1.1. Material properties

The basic material properties of GaAs have been well known. Along with Indium Phosphide (InP), GaAs is a binary III-V compound semiconductor which promises dramatic advantages over silicon. A number of quantitative comparisons of its properties are available in the literature [43,127]. A mainly qualitative summary of several of its properties in comparison with silicon illustrates its fundamental attractions:

(i) 6 times higher electron mobility;

(ii) 1.5 times higher electron saturation velocity;

(iii) High resistivity, semi-insulating substrate;

(iv) Higher electron band gap energy;

(v) Large velocity overshoot at low electric fields;

(vi) Direct band gap structure;

(vii) Higher radiation hardness.

The advantages of higher mobility and saturation velocity are self-evident. The higher band gap energy implies a wider operating temperature range. The semi-insulating substrate promises lower device to substrate and interconnect to substrate parasitic capacitances, and simpler device isolation. For MMICs, these also mean that inductors, capacitors, and resistors can all be compactly placed on the same substrate (hence monolithic).

Velocity overshoot is a particularly significant phenomenon in III-V compound semiconductors, such as GaAs and InP [13,71]. This is qualitatively illustrated in Fig. 2.1. The amount of overshoot can be quite marked — as high as two to three times of the saturation velocity — and was found to be very sensitive to channel doping, electric field, initial injection energy, and transient distance. Note that from the diagram that overshoot occurs at a much lower field in GaAs than in silicon. This phenomenon is most applicable to field effect transistors at submicron and ultra-submicron levels of gate/channel lengths, and was found to be small in silicon.

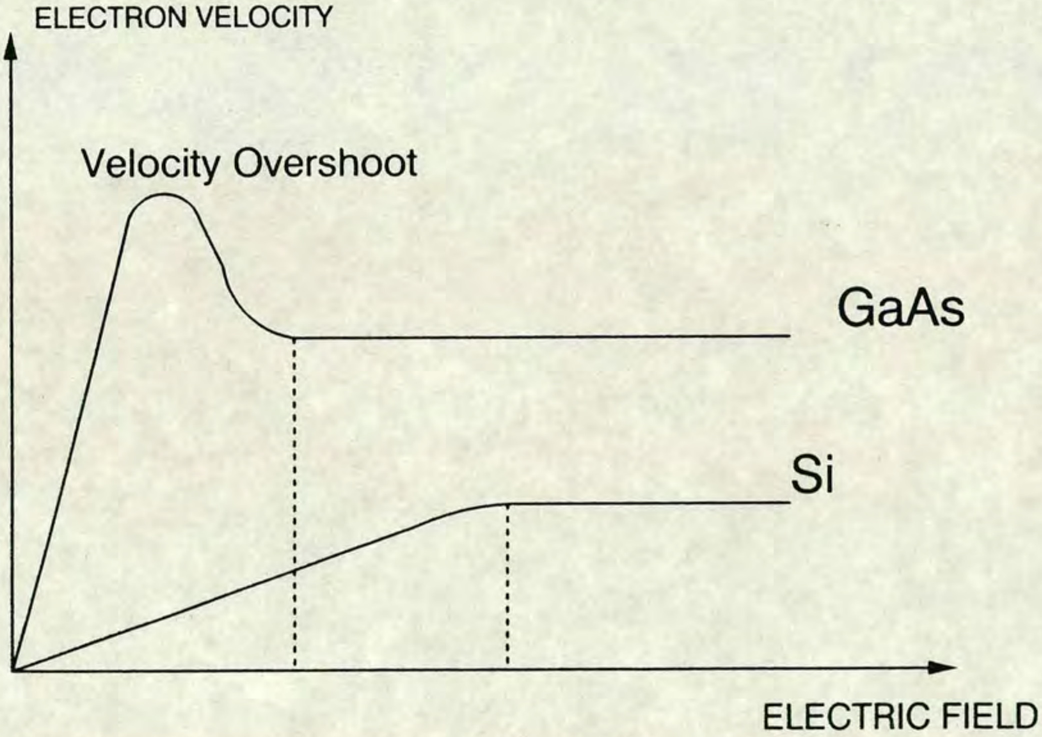


Fig. 2.1 Electron drift velocity vs. electric field in GaAs and Si

The direct band gap structure of GaAs results in its ability to emit infra-red light, and in theory fully integrated opto-electronic circuits can be realised [43]. In addition, GaAs is sensitive to radiation in the microwave frequency range. These properties make it attractive for fibre-optic systems.

Offsetting these advantages over silicon, GaAs unfortunately has several detrimental material properties:

(i) 2 times lower thermal conductivity;

(ii) Brittleness;

(iii) Lower hole mobility;

(iv) Orientation dependency of electrical properties.

The implications of (i) and (ii) are generally higher manufacturing costs. The low hole mobility (lower than that of silicon) rules out complementary structures where speed is of major concern. Orientation dependency implies less scope for layout compaction, because all active devices have to be parallel to each other.

Although the advantages of GaAs materials over silicon mentioned above are impressive, they do not indicate how or whether they can be utilised to the same degree in circuits and systems. In practice, factors such as device geometry, and circuit and system architectures, limit the speed advantage to much less than that suggested by the 6 times higher electron mobility. Examples of direct comparative studies have been published and will be discussed. First, the fundamental difference between silicon and GaAs IC technologies — the active device — is discussed.

2.1.2. Device constraints in GaAs

The early stages of an IC technology is inevitably entirely driven by device research and fabrication. The first and most important distinction between GaAs and silicon devices is the unavailability of a good quality oxide in GaAs, making insulated gate field effect transistors (IGFET) impractical, though not entirely impossible [32,36]. As a result, present digital GaAs circuits are based on the MESFET and the JFET, with the former being dominant. The operational and modelling problems of the MESFET are dealt with in more detail in the next Chapter. It is sufficient in this discussion to note this important difference and the immediate consequences of using a MESFET. First consider the fundamental differences in the switching behaviour between the MESFET and the two currently dominant devices in silicon, the MOSFET and the bipolar junction transistor (BJT). Table 2.1 below gives such a comparison.

Characteristic	MESFET	MOSFET	BJT
gate-source/drain	Schottky barrier	Insulated gate	pn junction
conduction	unipolar	unipolar	bipolar
switching/clamping	Schottky barrier	No clamping	pn junction
input impedance	low when ON	capacitive, high resistance	low when ON

Table 2.1 Qualitative comparison of MESFET, MOSFET, and BJT.

Fig. 2.2 illustrates schematically the different conduction mechanisms in the three devices. It is clear from these that, as a switch, the MESFET is very different from a MOSFET, even though both are FET devices, but shares some similarity to the BJT owing to its gate-source/drain junction. However, whereas a BJT has two pn junctions, the MESFET has one depletion region directly underneath the channel which also controls the flow of majority carriers between the drain and source. Also, the Schottky barrier has a different turn on characteristic compared to a pn junction, and has a lower forward biased voltage (V_{on}).

One device that does have many similarities to the MESFET is the JFET. The JFET was the precursor to the MESFET and was proposed as early as 1951 by Shockley. The first successful device was fabricated 7 years later [90]. In a JFET, a pn junction exists between the gate and the channel, instead of the metal/semiconductor contact in a MESFET. As a result, the JFET has a higher barrier height than the MESFET and has a larger gate voltage swing. The JFET is quite widely used for input stages in precision silicon operational amplifiers, such as a number of precision amplifiers available from Analog Devices.

The JFET was initially found to be less manufacturable in GaAs, and a simpler and more planar structures was sought after in the early stages of the technology. The GaAs MESFET was invented in 1965 by Carver Mead [24]. (The first silicon MOSFET was made in 1960 for similar reasons.) While the JFET has been used

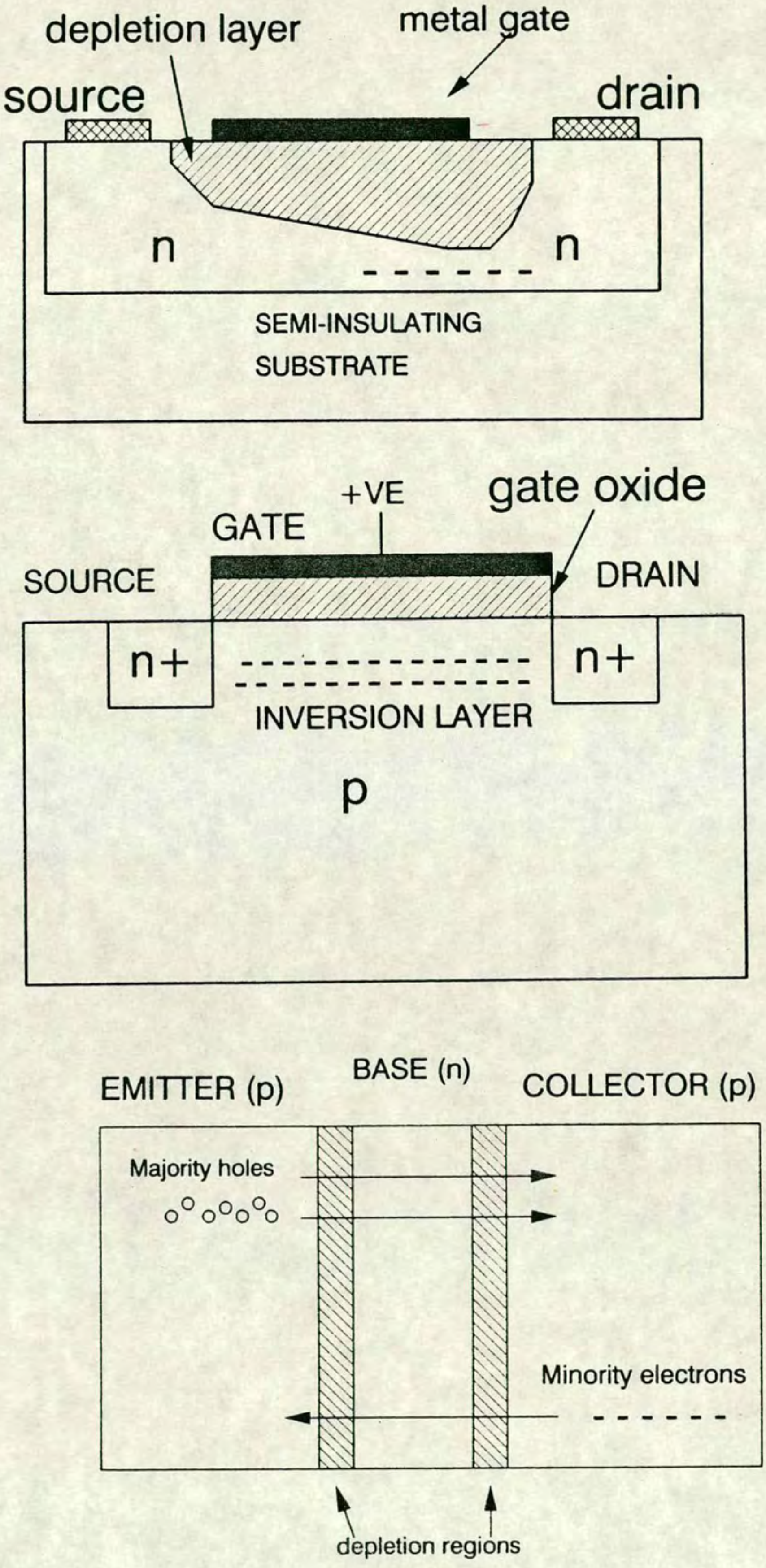


Fig. 2.2 Conduction mechanisms in MESFET, MOSFET, and BJT devices

increasingly in GaAs, the MESFET device still dominates in VLSI.

With the GaAs MESFET, both enhancement and depletion devices are available. Depending on the process and application, threshold voltages vary from −2 V to −0.5V for the depletion device, and +0.1 V to +0.3 V for the enhancement device. The symbols adopted for this and all subsequent discussions for these two MESFET's are shown in Fig. 2.3.

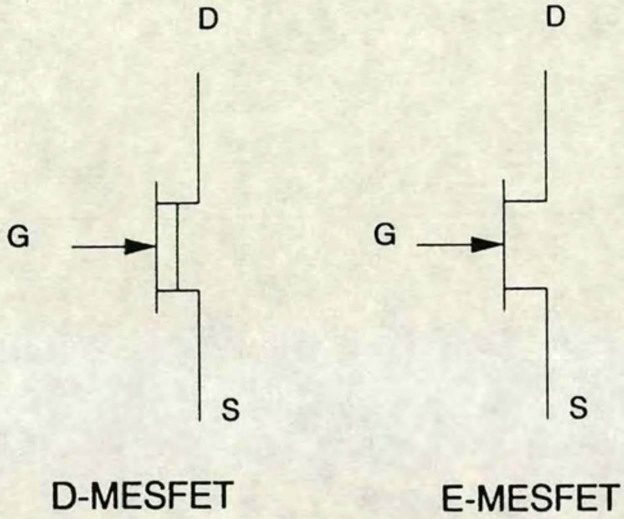


Fig. 2.3 Symbols used for D-MESFET and E-MESFET.

The other important active device is the Schottky diode, used extensively in level shifting and switching. It is commonly formed by a shorted MESFET, as shown in Fig. 2.4. If it is used for switching rather than level shifting, a separate mask may be introduced for the anode for compactness [24]. A reversed biased diode can also be used as a capacitor for level shifting [150], as will be seen later.

Passive devices such as resistors, capacitors, and inductors are mainly used in op-amps and MMICs, and are not discussed here. The next section will discuss the various logic circuit configurations centred on the MESFET and Schottky diode.

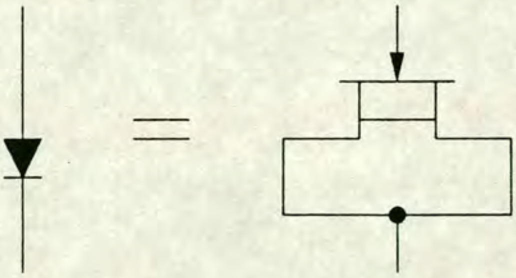


Fig. 2.4 Schottky diode made from a D-MESFET.

2.1.3. Logic circuit configurations

The relatively recent expansion of GaAs research predictably resulted in a fair number of different logic configurations generally aiming to find a suitable compromise between the often conflicting requirements in speed, power, and size. Owing to the limited number of devices available in GaAs, most logic families can be divided into two broad categories: D-MESFET only, and both E-MESFET and D-MESFET. They will be denoted as D-MESFET and E/D-MESFET logic respectively. The latter includes complementary logic circuits (both p and n channel devices).

D-MESFET logic

In this type of logic, level shifting is invariably required between logic stages in order to switch off the D-MESFET. The most common family is the Buffered FET Logic (BFL). The basic gate is made up of a logic block followed by a level-shifting buffer, as shown in Fig. 2.5. The number of level shifting diodes (made up of D-MESFET's) depend on the threshold voltage of the particular process. Most of the power is dissipated in the buffer stage, which has quiescent current flowing from V_{dd} to a negative V_{ss} for both HIGH and LOW output states. The power of the first stage is small provided that the pull-down D-MESFET is not forward biased. With this two-stage structure, the output of the first stage is effectively decoupled from fanout gate and interconnect capacitances, and the overall propagation delay is very small. The large voltage swing and the buffered output ensure good noise margins and fanout capability. To offset these, the normally-on buffer stage results in low power efficiency. Values of 34ps at 0.5μm gate lengths [150] are attainable with this structure and is therefore most suitable for applications where ultimate speed is required. The high power dissipation, ranging from several milliwatts to 50mW per gate depending on speed, and the large number of components per gate, exclude it from levels of integration higher than LSI. Low power versions of BFL (LP-BFL), using less level shifting diodes and D-MESFET's with higher (less negative) threshold voltages, tend to have little speed advantage over other logic families reviewed later on.

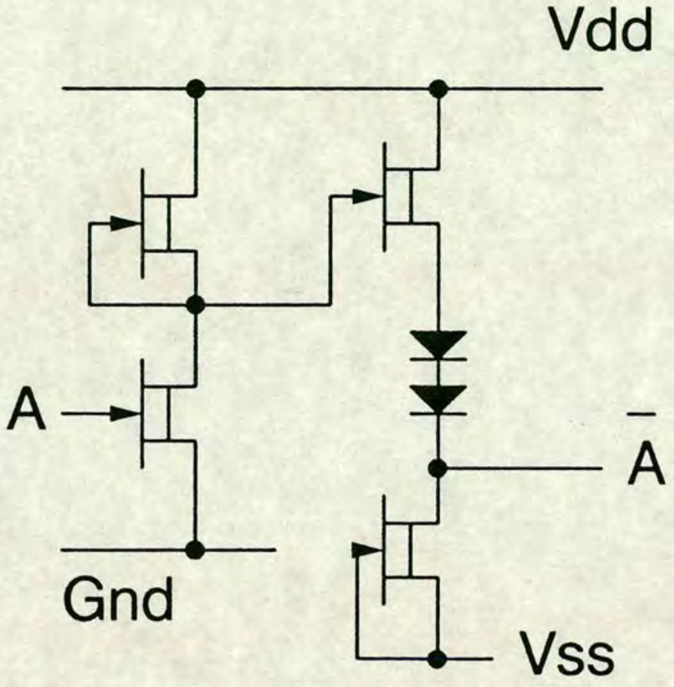


Fig. 2.5 Buffered FET logic (BFL).

To overcome the high power dissipation of BFL, several alternatives had been introduced. These include unbuffered FET logic (UFL) [74], Schottky diode FET logic (SDFL) [24], and Capacitor-coupled FET Logic (CCL) [75].

In UFL, the pull-up D-MESFET device in the BFL output buffer is removed to save power, at the expense of fanout capability. It is identical to BFL in all other respects. For low fanout logic, this was found to be a good alternative to BFL.

SDFL is similar in principle to UFL. But unlike BFL and UFL where logic switching is performed by a NOR gate with multiple pull-down D-MESFETs, level shifting and input logic switching are combined by using Schottky diodes and a D-MESFET, which now forms the input stage, as shown in Fig. 2.6. The output stage is an inverter which produces a maximum possible swing of up to V_{dd}. A NOR gate is formed if more than one switching diodes are connected in parallel at the input stage. A negative V_{ss} supply is again necessary. Additional level shifting diodes may be necessary if the D-MESFETs have very negative threshold voltages. The input diodes are usually small area Schottky diodes while the level shifting diode is made from a D-MESFET [24,25]. Significant power and size reductions resulted from SDFL, but with longer delays compared to BFL. Typical figures of propagation delay and power are 75ps and 0.3mW respectively [151]. SDFL has similar insensitivity to device threshold voltage variations, but has worse fanout capability due to the unbuffered output. Fanin, however, is substantially better than BFL. (Here fanin is defined as the number of inputs in a NOR gate and fanout is the number of logic gates of the same size being driven.) A recently published chip consisting of 2000 gate array cells [147], which also used a push-pull output stage, demonstrated high integration and low power. Unfortunately, gate delay was too large compared to even silicon bipolar gate arrays. The speed/power compromise might have been too severe in this particular case.

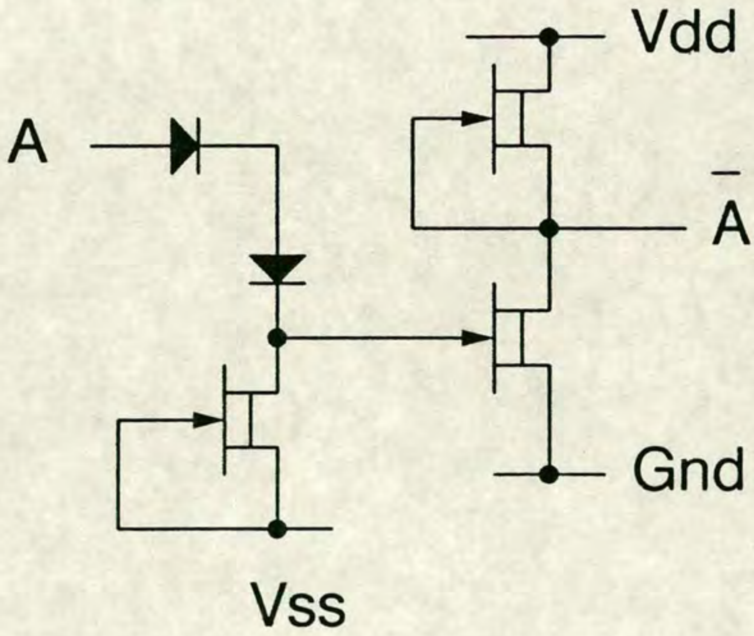


Fig. 2.6 Schottky diode FET logic (SDFL).

One interesting variation of SDFL exists, and is called inverted common drain logic (ICDL) [1]. This has essentially the same structure as SDFL, but uses an ion implanted resistor instead of small area diodes. The reasons for using a resistor were ease of process control and lower power dissipation. Experimental results showed that although speed-power product was reduced, speed suffered as result.

Another important attempt at solving the level shifting problem was the use of capacitor coupling between logic gates, initially called capacitor-coupled logic (CCL) [75]. Instead of forward biased diodes, reversed biased Schottky diodes are used as capacitors, as shown in Fig. 2.7. With a correctly chosen diode to MESFET ratio, charges are transferred as the input level changes, providing level shifting with minimal static power dissipation, and without the need for an extra negative supply. It was also designed to be insensitive to device variations, and produced figures of 55ps gate delay at 3.5mW/gate from ring-oscillator results [150]. Initial charging of the capacitors is necessary and a lower cut-off frequency exists. These were not considered to be major drawbacks for telecommunication applications at which it was aimed at. The ratio of the reverse biased diode capacitance to the total load capacitance determines charge transfer efficiency and hence fanout capability. With large fanouts, penalty is paid in the form of large diodes, with significant parasitics which need to be charged by the input gate during a transition. Since the output is also unbuffered, as in SDFL, CCL gate delay is more sensitive to fanout compared to BFL.

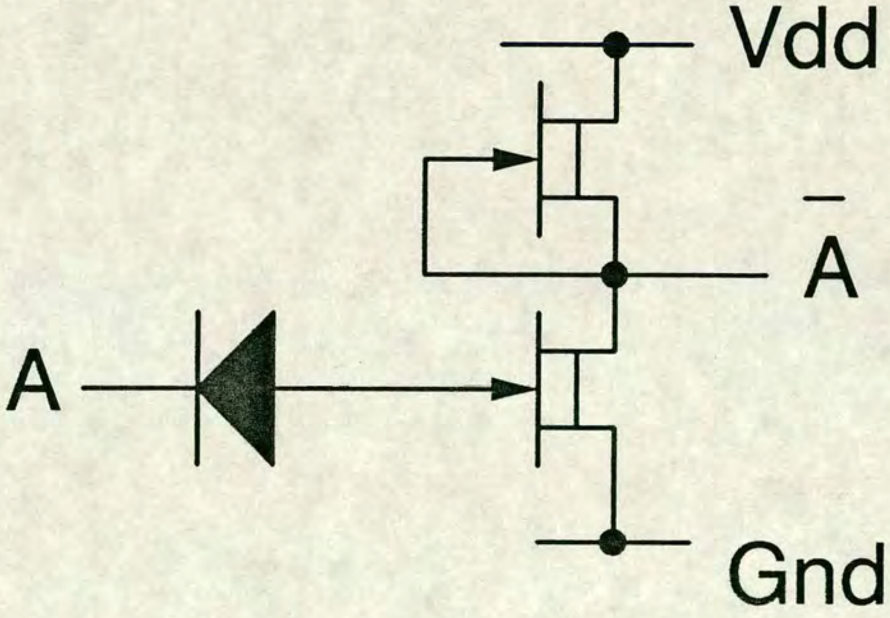


Fig. 2.7 Capacitor-coupled FET logic (CCL).

A variation of the basic CCL principle, called Capacitor diode-coupled FET logic (CDFL), improved on the speed and power performance further, and was adopted for a high speed logic family [25]. In the CDFL, extra diodes and pull-down device are added between stages to improve fanout capability. A CDFL inverter gate is shown in Fig. 2.8. A further variation of CDFL, super capacitor FET logic (super-CFL), uses a super buffer stage to further improve output current drive [76]. All versions of CCL are capable of higher integration than BFL (up to LSI), and have better speed-power products, but cannot match BFL in ultimate speed.

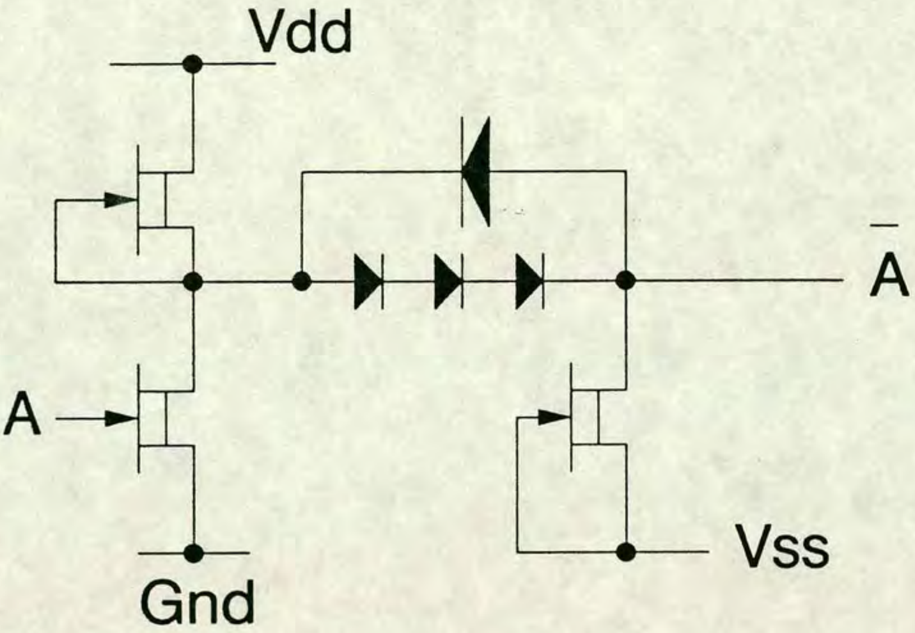


Fig. 2.8 Capacitor diode FET logic (CDFL).

All the D-MESFET logic discussed so far operate in voltage mode. The GaAs equivalent of silicon ECL is generally called source-coupled FET logic (SCFL). Its basic structure, shown in Fig. 2.9, consists of source-coupled pairs and current sources biased by voltage references, with the outputs buffered by source following level shifters, similar to those in BFL. This implies that the output is in voltage mode while the logic block operates in current mode. As a result, it is also insensitive to threshold variations, and has all the advantages of current mode logic [127]. Recent results [44,53,148] appeared to be very favourable both in power and speed compared to BFL, producing figures of 50ps of propagation delay for a NOR gate (fanout unknown). A low power version called LSCFL gave figures of 24ps and 2.4mW/gate [55] for a fanout of 1, 0.4μm gate length, plus 255 μm of interconnect. The buffered outputs ensured low fanout sensitivity. The SCFL family is relatively new compared to BFL, and is clearly a major contender at the MSI to

LSI level, where high speed and acceptable power dissipation are required.

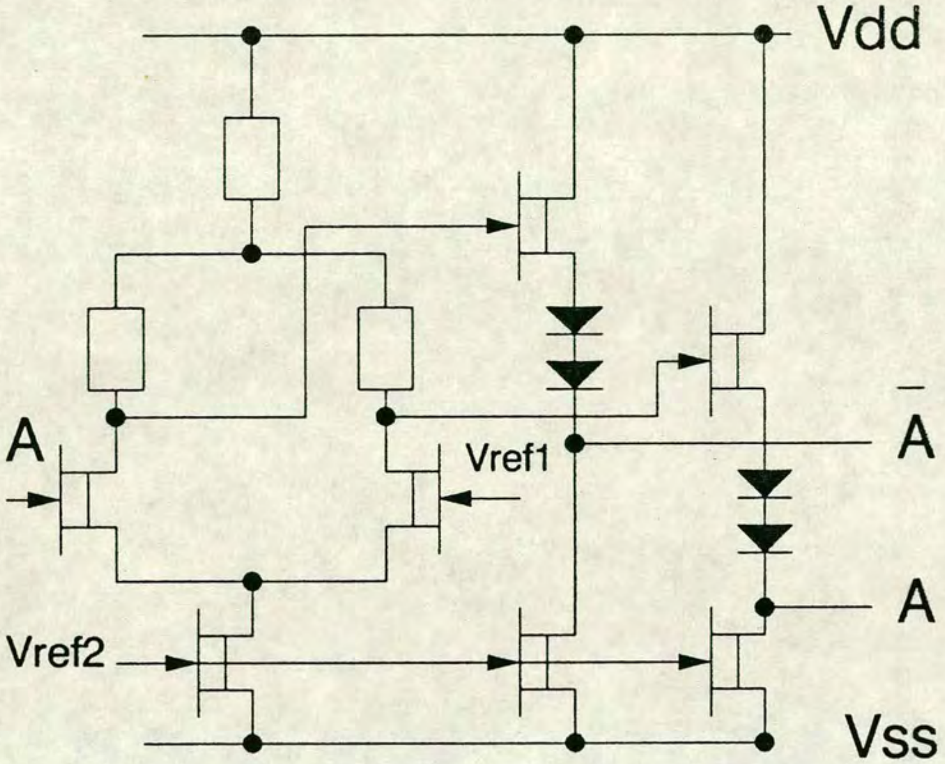


Fig. 2.9 Source-coupled FET logic (SCFL).

Finally, an important issue worth mentioning is compatibility between the different D-MESFET logic structures, and between GaAs and silicon ECL. In the former case, all the D-MESFET configurations discussed above require two supply rails (V_{dd} and V_{ss}), and are directly compatible to each other with the same process. In a custom design, they can be mixed at will to optimise area, speed, and power on the same chip. Where ECL compatibility is necessary, the V_{ss} supply rails can be split and level-shifted, as shown in Fig. 2.10, to share the conventional ECL rails. A total of two or more power supplies are normally necessary, depending on the silicon ECL family used. Since D-MESFET logic is generally process tolerant and has a large logic swing, the level-shifted supply rails are not detrimental to noise margins.

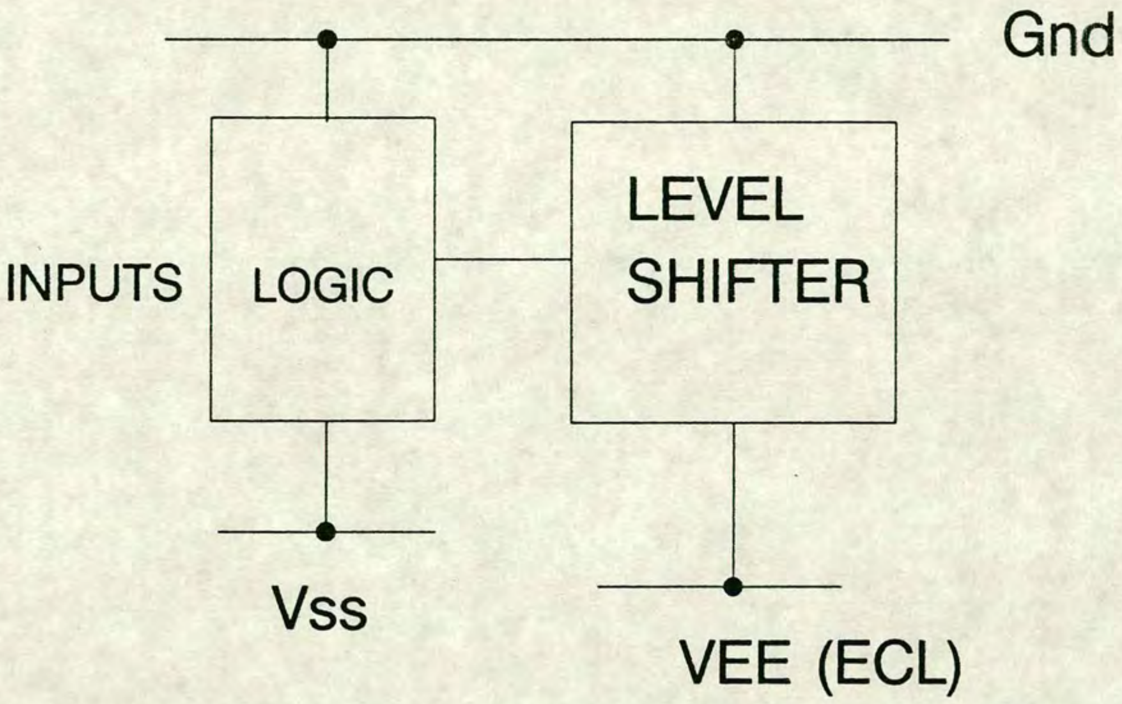


Fig. 2.10 Level-shifted supply rails for ECL compatibility.

E-MESFET/D-MESFET (E/D) logic families

The major drawbacks of D-MESFET logic are clearly the high power dissipation and component count per gate. For commercial survival, high integration and functionality are mandatory. Although low power D-MESFET logic families such as SDFL and SCFL did go some way in improving speed-power product, they are all generally considered to be too large for VLSI. The simplest solution is to have silicon NMOS-like static logic, using both E-MESFET and D-MESFET devices — direct-coupled FET logic (DCFL).

The basic DCFL inverter is shown in Fig. 2.11. The pull-down E-MESFET avoids the need for level shifting between stages, hence the name. The voltages shown in Fig. 2.11 are for a DCFL gate with a non-zero fanout. If the gate is not clamped, as might be the case when it is driving a source follower, then the output HIGH voltage can rise to V_{dd}. The immediate advantages are small size, low power, and single supply operation. It is therefore widely accepted as the most suitable logic family for VLSI in GaAs.

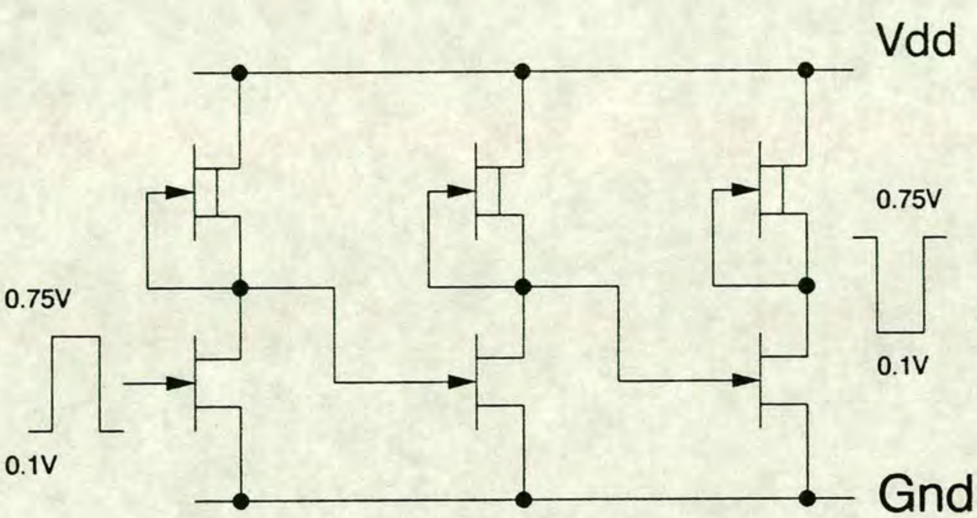


Fig. 2.11 Direct-coupled FET logic (DCFL) inverter chain.

In DCFL, major drawbacks caused by the unavailability of a IGFET device are small logic swing (typically 0.1V to 0.75V), low fanin and hence functionality per gate, and high fanout sensitivity. Stringent process control over threshold voltage is also crucial. Together with high costs, lower yields, and lower noise margins compared to D-MESFET logic families, DCFL was considered to be too high a risk to adopt as a standard logic family until recently. Self-aligned gate ion implanted E/D processes have improved to such an extent in the last two to three years that DCFL has become a viable commercial alternative to silicon ECL [73]. The basics of DCFL design are discussed in more detail in the next Chapter.

Although DCFL was primarily envisaged for VLSI rather than ultimate speed, impressive figures of under 100ps gate delay at 0.3mW/gate were achievable even from a gate array [73]. The high fanout sensitivity would increase delay closer to 200ps, comparable to the better silicon ECL gate arrays. The most important advantage of GaAs, however, is the much lower power dissipation. A more detailed look at silicon ECL performance later on will confirm that GaAs DCFL is commercially viable mainly because of its power advantage.

In order to reduce its fanout sensitivity, various forms of buffering were introduced. One method used a super-buffer output stage, and is called super-buffered FET logic (SB-DCFL) [73]. The basic super-buffer is similar to its silicon NMOS counterpart, and is shown in Fig. 2.12. Because of the shared gate terminal between the two stages of the buffer, a compact layout of the logic gate is possible, despite its apparent complexity.

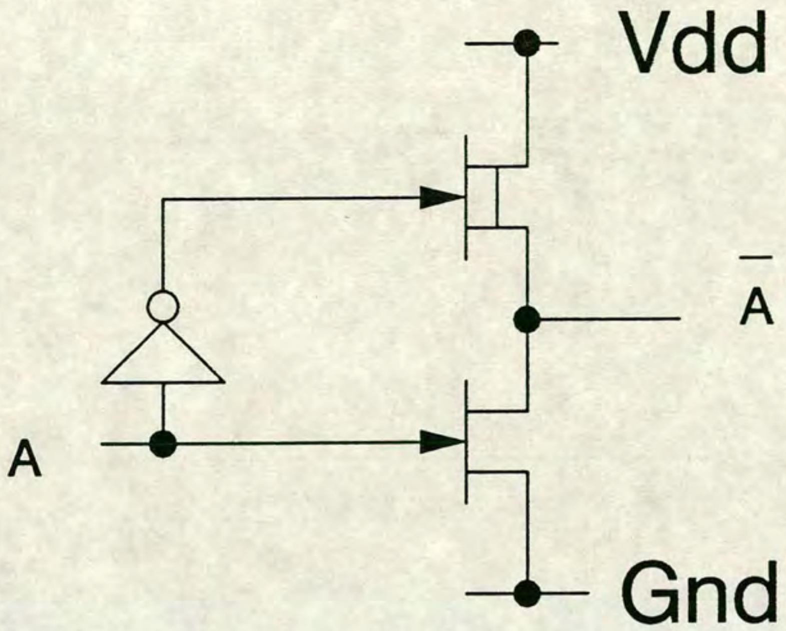


Fig. 2.12 E/D super-buffer used in super-buffered FET logic (SB-DCFL).

Other forms of output buffering use source followers, increasing fanout capability at the expense of dc power dissipation. This is used in one of the various forms of a logic family called low pinch-off FET logic (LPFL) [101]. LPFL was envisaged as a compromise between DCFL and BFL, and used MESFET's with threshold voltages close to 0V and single supply operation. At least six forms of LPFL had been proposed [151]. They follow the lines of modified forms of D-MESFET logic families such as BFL, UFL, SDFL, and ICDL. One example, which uses source follower buffering, is shown in Fig. 2.13. They generally have the BFL advantages of process tolerance and good fanout capability, and lie between BFL and DCFL in terms of speed, power, and size. The output swing is about 100mV higher than DCFL. This was especially attractive in the early stages of DCFL development when the swing was as low as 500mV due to the low Schottky forward voltage (V_{on}). Noise margins were also improved with LPFL. However, an examination of recent GaAs circuits showed that the interest in LPFL had declined owing to improvements in processing, although the concept was clearly a significant one.

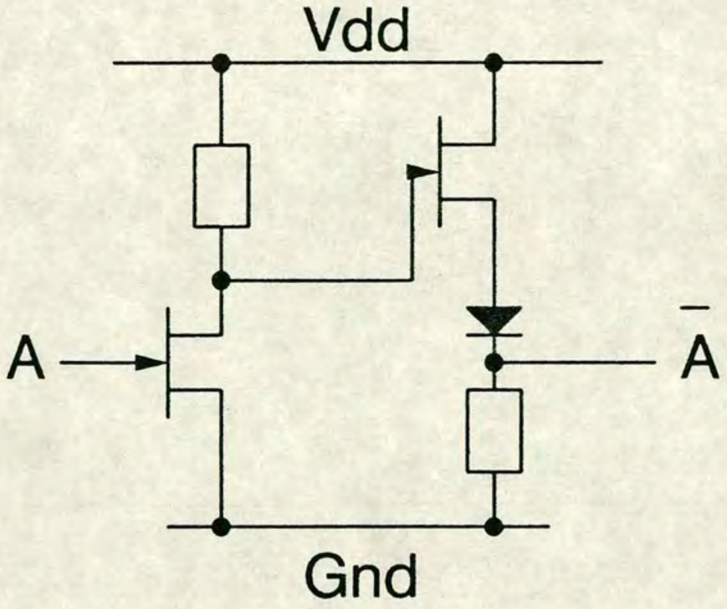


Fig. 2.13 One form of low pinch-off FET logic (LPFL).

In current mode form, one notable attempt was to use SCFL with an E/D-MESFET process. The basic concept remained unchanged, but with E-MESFET replacing most of the D-MESFET devices in the differential pairs and voltage references. Simulated results showed that an E/D SCFL frequency divider operated up to 3 GHz at 3 times power dissipation and same circuit density, compared to the DCFL equivalent operating up to 1.5 GHz [148]. Although 4-bit digital-to-analogue and analogue-to-digital converters were designed and tested successfully using E/D SCFL, full test results were not disclosed and simulated advantages could not be confirmed. The increased speed was largely attributed to the much better fanout capability, at the expense of power. Another important feature was the ease of integrating digital, analogue, and even microwave circuits on the same chip. The first two of these had been demonstrated by the converters already mentioned.

These different variations of E/D logic had aimed at striking a reasonable compromise between the low power of DCFL and the high speed of BFL. Although the hole mobility of GaAs is lower than that of silicon, quasi-complementary and complementary logic had been used in GaAs to reduce power even further than conventional DCFL. Particular areas of interest are memories and gate arrays.

In the gate array sector, a quasi-complementary output stage was used with a custom input logic block. This logic is called Gain FET logic (GFL) (named after the company), and is shown in block form in Fig. 2.14. Power is reduced by avoiding a dc current path in the output stage. Full details of the input logic block were not disclosed, and was claimed to have better input noise margins than DCFL. Performance figures of 180ps at 700μW/gate for an inverter were also quoted. Although these appear to be worse than conventional DCFL, the increased fanin and fanout capability and the resulting higher functionality per gate meant that much less gates are needed than DCFL for the same logic function. An earlier comparison of quasi-complementary logic [151] showed that for large fanouts, this approach was fastest compared to inverters with resistive and D-MESFET pull-up loads. Unfortunately, it also had the worst fanin capability. GFL had apparently solved the fanin problem, while inevitably increasing total propagation delay through the gate, confirmed by the quoted figures. In terms of size, direct comparison was not available. The largest array offered has 1444 cells. A 14 × 16 cross-point switch was fabricated and worked up to 500Mbits/s into 50Ω, at a power dissipation of 1.1W. Other GaAs gate arrays are reviewed in a later section.

In the case of complementary logic, the most notable attempts were SRAMs ranging from 4K to 16K [70,146]. Apart from the reduction in power by using n and p channel devices, the other distinguishing feature of these designs is the use of JFETs rather than MESFETs. Figures for the 16K SRAMs were 16.1ns access time, 678mW dissipation, and 3% yield. Much better access times are possible in GaAs RAMs which used non-complementary logic, but the low power and good yield may prove complementary logic to be viable. However, it has not been adopted for other types of circuits. A 32-bit RISC microprocessor made with the same JFET process used n-channel devices only [45]. The speed degradation was clearly considered to be unacceptable for most circuits.

Finally, in terms of compatibility, the problem of different supply voltages is less severe owing to the lower currents involved, compared to D-MESFET logic. In DCFL and its variations described so far, V_{dd} is normally chosen to be sufficiently high to saturate the D-MESFET pull-up device, but low enough to minimise dc power. Values can range from +1V to +2V, and are therefore not directly compatible with any silicon technology. A simple solution had been to use the same supplies as silicon CMOS or ECL, and connect external resistors in series between the GaAs supply terminal and the board V_{dd}. This is clearly inefficient if a large number of DCFL chips are involved, and an extra rail with on-chip level translators would be necessary in such cases, as with the D-MESFET logic families

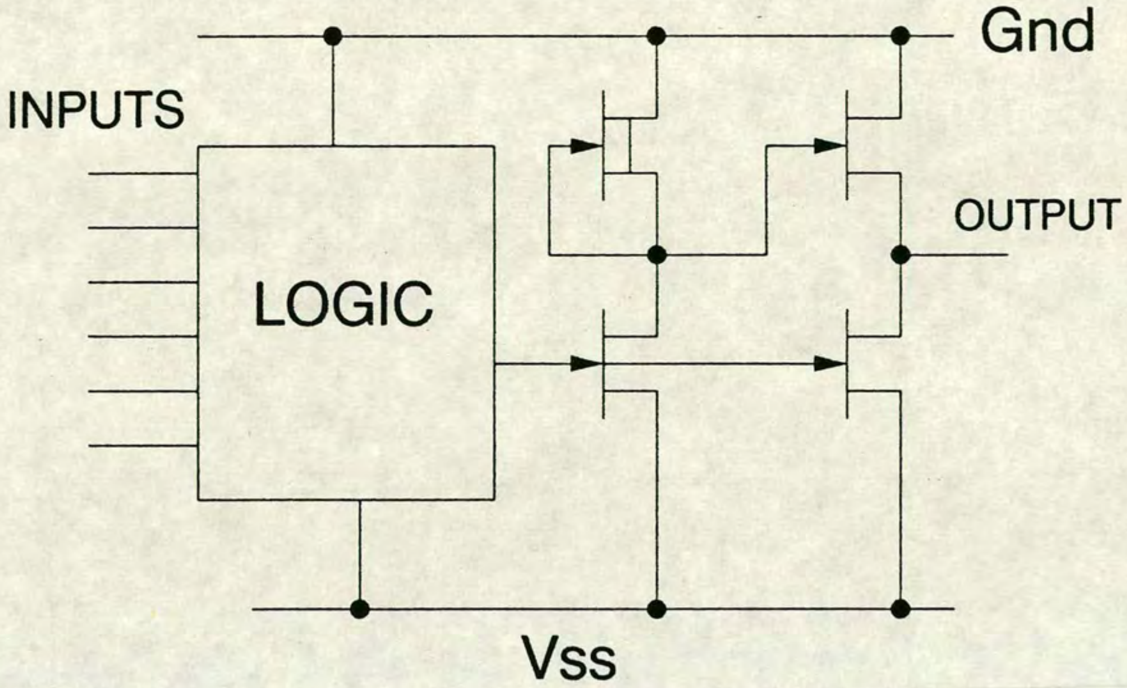


Fig. 2.14 Block diagram of gate array cell with quasi-complementary output stages.

discussed above. (ECL compatibility is discussed in more detail in Chapter 3.)

One interesting configuration that avoided either of these board level adaptations is the stacked DCFL logic [121]. This consisted of upper and lower DCFL blocks with an interface block referenced to a virtual ground (VG). The block diagram is shown in Fig. 2.15. Test results of a prescaler IC, using a V_{dd} voltage of 3.3V, showed 31% reduction in circuit current compared to the resistor and conventional DCFL arrangement. The IC operated up to 900MHz. A three-stack structure was considered to be possible and should reduce current even further. Direct comparison of speed was not available. The main disadvantage of stacked circuits is clearly gate size, limiting integration level.

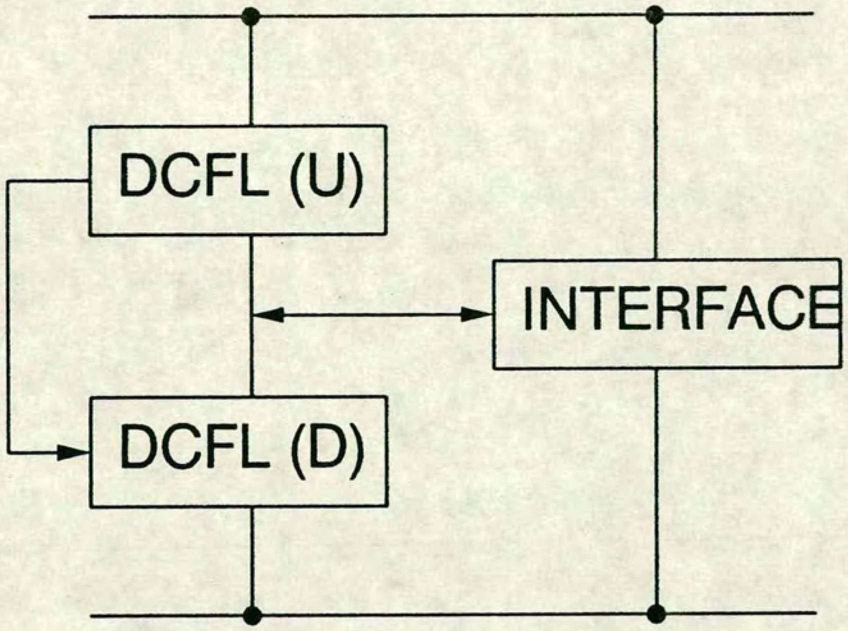


Fig. 2.15 Stacked DCFL logic structure.

2.1.4. Circuits and systems

In the last sub-section, a fair number of logic families, spanning the two extremes of D-MESFET BFL and complementary E/D JFET logic, were discussed. It is clear that no one single family can dominate and the application in question ultimately dictates what combinations or variations of these families should be used. In this section, recent results of GaAs circuits, some of which are commercially available, are reviewed according to their functional areas. There are four main application areas in which digital GaAs ICs are used:

(i) Signal processing.

(ii) Communications.

(iii) Test and measurement.

(iv) Supercomputers.

Signal processing and communications are closely related areas, while the other areas also overlap with the others. Although no sharp distinction shall be made on the applications of GaAs circuits in this review, a few words on signal processing are appropriate here.

Signal processing

Owing to military interests in the States and Europe and commercial interests in Japan, communications and signal processing are the major areas that attracted funding for GaAs. These have made possible advances such as the jump from 4K to 16K SRAMs in 2 years. The material advantages of GaAs in terms of radiation hardness and potential speed were the fundamental attractions, although silicon advances, in response to GaAs, have forced the speed argument more and more towards lower power.

Signal processing covers a wide range of circuit functions, ranging from custom FFT processors to general purpose microprocessors. These include:

(i) Arithmetic logic, such as multipliers and adders, and registers.

(ii) Memories.

(iii) Microprocessors.

(iv) Application specific processors.

Other equally important SSI to MSI circuits are A/D and D/A converters, and sample and holds. Due to their semi-analogue nature, they are not reviewed here. The next section will first review the core of all signal processing systems.

Arithmetic logic

The most important elements in arithmetic are the adder and the multiplier. Apart from their obvious functions, the multiplier is often used to assess a certain new process or logic configuration due to its circuit complexity. The speed of the multiplier also determines the processing power of a system for the same reason. Recent published results on adders and multipliers are summarised in Table 2.2.

It can been seen from this Table that low power, medium to high speed, 4-bit circuits were most developed. Two of the circuits, the 32-bit adder and 8×8 multiplier, exceeded 1W in power dissipation, as a direct result of using D-MESFET logic.

Assuming good accuracy of manufacturer's figures, a comparison of off-the-shelf products between silicon and GaAs should indicate possible real system advantages. For a silicon ECL 100K series 4-bit ALU [94], the worst case adder propagation delay from input to carry output was quoted at 8.3ns (excluding setup and hold times), with a power dissipation of 920mW. Worst case test figures for the Vitesse 4-bit slice ALU listed in the Table, also available commercially, were 4.7ns at 240mW power dissipation for the same function. Whereas these figures point towards significant improvements by using the GaAs parts, a customised silicon ECL ALU may match them in speed. The greater than 3 times power advantage is more significant in the context of a complete system, as other GaAs system comparisons will confirm in later discussions.

Circuit and performance (worst delay/power)	Ref.
(1) 4-bit carry look-ahead generator, 1.55ns, 34mW, DCFL.	[117].
(2) 4-bit ripple carry adder, 1.25ns, 180mW, lp-BFL.	[107].
(3) 4-bit adder-accumulator, 340 MHz, 500mW, BFL.	[26].
(4) 8-bit adder, 11ns, 236mW, SDFL.	[147].
(5) 32-bit adder, 2.9ns, 1.2 W, BFL.	[157].
(6) 4X4 multiplier, 2.5ns, 40mW, DCFL.	[23].
(7) 4X4 multiplier, 1.4ns, 1.6 W, CDFL.	[76].
(8) 5X5 multiplier, 4ns, 133 μW/gate, DCFL.	[17].
(9) 8X8 multiplier, 5.25ns, 1-2mW/gate, SDFL.	[24].
(10) 8X8 multiplier, 6ns, 876mW, JFET DCFL.	[40].
(11) 16X16 multiplier, 10.5ns, 952mW, DCFL.	[96].
(12) 4-bit slice ALU, 4.7ns, 240mW, DCFL.	[49].

Table 2.2 Performance of arithmetic logic elements

In contrast to this favourable comparison of "standard" products, a laboratory silicon CMOS 16×16 multiplier, based on 0.6μm gate lengths, gave a typical multiply time of 7.4ns and dissipated 400mW [103]. This is significantly better than the GaAs DCFL multiplier figure of 10.5ns worst case delay, achieved 4 years earlier

(Table 2.2). A more detailed examination revealed that the GaAs multiplier used a classic parallel array architecture, while the CMOS circuit used a Booth's algorithm array scheme to reduce the number of adder stages (Chapter 6). Also, the GaAs MESFETs used in this example had gate lengths of 2μm. Projected performance with 1μm gate lengths was 6.5ns. Unfortunately, attempts at improving on the GaAs figures are not available in recent publications and therefore an up-to-date speed comparison cannot be made. Nevertheless, these results demonstrate that at this LSI level, the well known advantages of GaAs materials over silicon do not guarantee correlation with circuit performance, and that circuit architecture becomes more and more important as integration level increases.

Memories

The fabrication of memory components has traditionally been the most difficult for any process owing to the need for small feature sizes and high yield. Interests in GaAs have been mainly in SRAMs. Table 2.3 below summarises a number of recent achievements.

Referring to the Table, it can be seen that DCFL based logic dominated all sizes. In the 1K category, fastest access time was 1ns with the second lowest power of 300mW. Lowest power was achieved with a 0.7μm process. In the 4K category, best access time was again 1ns. Lowest power was achieved at 200mW with a complementary JFET process. The largest memory size in GaAs is 16K, and produced best access time of 7ns at 2.1W. Lowest power was again from a complementary JFET process, which was also the slowest. Overall, no GaAs SRAM could achieve subnanosecond access time, and power covers a range of 200mW to 2.1W. For comparison, a 4K high electron mobility transistor (HEMT) RAM was capable of 0.5ns access time, but at a high power of 5.7W (Table 2.3).

In GaAs, the processing problems that most affected memory ICs are temperature gradients in annealing furnaces, handling damage, surface contamination, residual polishing damage, and the small diameter of wafers [140]. The effects of the high dislocation density in un-doped GaAs wafers were less clear, and were thought to affect E/D-MESFET memories more than D-MESFET circuits [125]. Indium doped wafers were found to reduce dislocations by at least 4 times. They were also more expensive and brittle, restricting their general acceptance.

In terms of circuit design, several problems have hampered the acceptance of large SRAMs in GaAs [137]. These include non-standard power supplies, low tolerance to supply, temperature, and process variations. Together with typical yield figures of

1% to 3% for 16k SRAMs, they are still a significant way from widespread system replacement of silicon parts.

In contrast, 1K SRAMs are relatively well developed, with parts available commercially that can directly replace silicon ECL 100K and TTL memories. A silicon ECL 256×4 bit SRAM gave worst case read access time of 10ns and power dissipation of 1W [88]. Test results of pin compatible GaAs parts produced figures of 2.5ns and 1.5W [137]. Therefore the GaAs SRAMs have a better speed-power product, but higher power dissipation. This was likely to be the result of using some current-mode logic in this GaAs SRAM. Yield was high at 30%. Other laboratory results have shown much better speed-power product than these standard parts. It was found to be generally true that when ECL or TTL compatibility are necessary, power had to be sacrificed.

Circuit and performance (read access time/power)	Ref.
(1) 1K SRAM, 3.6ns, 68mW, 2μm DCFL.	[157].
(2) 1K SRAM, 1.3ns(min.), 1.4W, DCFL.	[86].
(3) 1K SRAM, 1ns, 300mW, DCFL.	[134].
(4) 1K SRAM, 3.6ns (worst), 1.9W, E/D CDFL.	[31].
(5) 1K SRAM, 2.5ns (min.), 1.5W, DCFL, TTL/ECL pin compatible.	[137].
(6) 1K SRAM, 1.4ns (min.), 210mW, 0.7μm DCFL.	[35].
(7) 1K SRAM, 2.3ns (mean), 550mW, 1μm DCFL.	[48].
(8) 4K SRAM, 2.6ns (mean), 890mW, 1μm DCFL.	[48].
(9) 4K SRAM, 1ns, —, 0.7μm,DCFL.	[82].
(10) 4K SRAM, 0.5ns, 5.7W, 0.5μm,HEMT	[99].
(11) 4K SRAM, 10ns (typical), 200mW (typical), complementary JFET.	[100].
(12) 4K ROM, 1.2ns, 3.75W, 0.5μm,DCFL.	[56].
(13) 4K SRAM, 7ns (worst), 850mW, 1μm DCFL.	[79].
(14) 4K ROM, 1.2ns, 1.9W, 1μm CDFL.	[16].
(15) 4K DRFM, 1.4ns (min.), 1.5W, DCFL.	[153].
(16) 16K SRAM, 20ns (typical), 1W, 1μm,D-MESFET.	[143].
(17) 16K SRAM, 22.5ns (worst), 678mW, complementary JFET.	[146].
(18) 16K SRAM, 7ns, 2.1W, 0.7μm DCFL.	[81].

Table 2.3 GaAs Memories.

At the 4K-bit size, a comparison of manufacturers' figures of standard parts is favourable towards GaAs. The silicon ECL 10474-15 4K RAM achieved worst case chip select access time of 8ns [88], compared to 3.5ns of the Vitesse GaAs VE12G474 of the same memory size [18].

Moving away from standard parts, the comparisons are not as clear cut. One example of silicon performance is a 5K ECL SRAM with a best access time of 0.85ns at 2.4W [14]. The technology used was an advanced self-aligned 1.2μm process. At a similar time, two independent GaAs 4K SRAM produced figures of 1.3ns and 1.4W [86], and 2.2ns and 890mW [48], for minimum access times and power respectively. The larger size of the ECL RAM implies that it has slight advantage over GaAs on both speed-power product and speed. The relative states of development of the three processes involved are difficult to assess. In the case of 16K SRAMs, the 3.5ns and 2W performance of one silicon example [51] compares favourably to the 7ns and 2.1W of a GaAs SRAM [81]. These laboratory results imply that at the present state of GaAs development, it has no advantage over silicon in memories larger than 4K bit.

Microprocessors

The first major initiative in GaAs microprocessors was the 32-bit reduced instruction set computer (RISC) program, started in 1984 by the Defence Advanced Research Projects Agency (DARPA). This processor was targeted specifically for the advanced on-board signal processor (AOSP) for a broad range of military missions [91]. The RISC architecture, as opposed to CISC (complex instruction set computer) popular in silicon, was considered to be the only viable one, owing to the limited integration level, fast switching speed, high dependency of fanin and fanout, and the non-existence of dynamic logic of GaAs technology available at the time. More importantly, this program had drawn attention to the need for different architectural approaches to GaAs VLSI in general.

Other efforts at making GaAs microprocessors include 4-bit slice and 8-bit slice processors. Test results of these efforts and the 32-bit RISC processor are listed in Table. 2.4 below.

Processor	Comment	Ref.
(1) 4-bit slice processor chip set.	ALU: 6.5ns t_{pd},1W.	[49].
(2) 8-bit slice.	150 MIPS (6.6ns), 9.2W, D-MESFET.	[37].
(3) 32-bit RISC.	60MIPS (16.5ns clock cycle), 4.7W, JFET DCFL.	[45].

Table 2.4 GaAs microprocessors

The 4-bit slice chip set consisted of a central processor, a carry-look-ahead generator, a microcontroller, and a 4K SRAM. They were designed as direct replacements of Am2900 (Advanced Microdevices) series silicon ECL chip set. This effort was therefore significant within the GaAs industry as the first to offer such a chip set. Projected performance of 4×4-slice system was 55MHz clock rate. Direct experimental comparison with a silicon system is not available. However, judging from the low clock rate with 4 cascaded 4-bit slices, speed advantage over 16-bit silicon CMOS microprocessors is likely to be limited. The projected performance for an integrated 16 bit processor was more practical at 125 MHz. These figures do demonstrate that advantages from fast switching speeds on-chip are easily crippled by going off-chip.

The other two processors were directed more at military systems. Both the 8-bit and 32-bit processors improved on the 4-bit processor by a significant amount. The 32-bit RISC processors also demonstrated VLSI capability in GaAs, particularly using direct-coupled FET logic (DCFL) — the MD484 (McDonnell Douglas) CPU contained 21606 transistors and was fully functional.

Communication between processors in multi-processor systems is an important issue in both GaAs and silicon microprocessors. Two GaAs circuits in this area have been published. The first is a 32-bit asynchronous serial data transceiver chip set [102]. The chips have been tested at 500Mb/s using 16 32-bit silicon microprocessors. Power was rated at 1W. The other circuit is an 8-bit bus logic IC designed specifically for a parallel data transfer network [64]. A test cycle time of 10ns has been achieved, at a high power dissipation of 7W. By using 48 ICs, 4Gb/s transfer rate was considered to be possible.

In terms of speed, the 60MIPS GaAs RISC chip compares favourably with the 10MIPS achievable in 32-bit silicon chips at present. But a commercially available 64-bit RISC chip from Integraph [29] is already capable of 50MHz clock rate and 20MIPS in CMOS. Its ECL version under development is aimed at 100MHz and 80MIPS. Therefore the speed advantage of GaAs in this area is not clear cut. The availability of supporting software is clearly also a major factor in the acceptance of a new microprocessor. It is likely that future GaAs circuits will need to take this into account if they are to compete with silicon.

Application specific processors

Most of the ICs mentioned so far were designed for one general functionality per chip, such as multiplication or memory. Very few published results were available on dedicated signal processing circuits and systems, while noting that some military circuits are not publishable. (Optical fibre transmission circuits are discussed in a later section.) Table. 2.5 lists 3 recent examples.

Among these systems, the highest integration level was used in the FIR filter, which utilised 13376 sites in a gate array IC, and a 4K SRAM. The 400MHz clock rate achieved has demonstrated the practicality of GaAs gate arrays in a high speed system, while noting that such speed should be achievable with silicon ECL gate arrays. More importantly, it was noted that higher performance and lower power dissipation resulted from using a unique GaAs I/O structure, rather than traditional ECL or TTL levels. Full details of this structure were not disclosed.

The third project in the Table was impressive in its scope. In terms of performance, the most notable was the half-FFT butterfly in achieving 500MHz clock rate with 88 custom SSI and MSI GaAs ICs. The operations involved two 16×8 complex multiplications, additions, subtractions, and serial-parallel shifts. All the ICs had ECL interfaces, driving 100Ω loading. An RF memory in the Table [22] has also demonstrated 1GHz capability in signal processing by using a number of GaAs ICs.

Circuit/system	Ref.
(1) Numerically controlled oscillator IC for direct digital frequency synthesis and phase modulation. Consists of pipelined accumulator, phase modulator, phase adder, two ROMs, and waveform generator. DCFL-based. Tested functional, speed unknown.	[80].
(2) Low pass filter (2 ICs) in sub-band tuner for adaptive tuning to narrow band signals. Custom FIR made from a 4K SRAM and a 10K gate array. 400 MHz clock rate. DCFL-based.	[8].
(3) 7 ICs (i) Radio frequency memory. Tested to 1GHz, 35 GaAs ICs. (ii) Half FFT butterfly. Tested to 500MHz, 88 GaAs IC. (iii) Digital convolver. (iv) Phase shift keying code generator. (v) Timing incoming detector. (vi) 8-bit A/D processor. (vii) Fibre-optic transmission system, Tested to 700MHz, 40 GaAs ICs. Microstrip boards made with up to 88 GaAs ICs (custom D flip-flops, 4-bit universal shift register, serial-parallel multiplier accumulator, sample/hold, and comparator).	[22].

Table 2.5 Signal processing systems

It would appear from these different examples that custom high integration ICs with GaAs I/O levels would combine to produce systems performing complex signal processing functions at 1GHz clock rates or above.

Communications circuits

Telecommunication is one area that sees continuous growth. The optical properties of GaAs, already mentioned earlier, and its high switching speed have been a major focus for integrated optical transmit/receive (transceiving) systems. Some the circuit and system achievements are listed in Table. 2.6 below.

Most of the chips or chip sets listed have been tested in a real transmission environment. For example, the 2.4Gb/s chip set from BTRL has been tested in a 50km link using single-mode fibres [130].

Apart from systems, a number of frequency dividers, flip-flops, multiplexers (MUX), and demultiplexers (DEMUX) have been reported to demonstrate the suitability of GaAs for telecommunication. Figures of 1.2Gb/s for multiplexers [150] and 10GHz for frequency division [116] had already been achieved in 1983/84. More recent circuits include a 26.5 GHz BFL 1/2 divider based on 0.2μm MESFETs [63], and 4 Gb/s 16:1 and 1:16 MUX/DEMUX circuits [53]. Other switching circuits included 12Gb/s 2-bit MUX/DEMUX ICs [58], and 16×16 [5], 8×8 [47], and 4×4 [96] cross-point switches at up to 3Gb/s input data rate. The 16×16 cross-point switch contained 10000 transistors and dissipated only 800mW.

Existing optical fibre systems have been based on 140Mb/s, 565Mb/s, 1.6Gb/s, and 1.7Gb/s rates [130,46], and were built from discrete silicon components at the higher bit rates. The GaAs circuits have been concentrating on the imminent standard of 2.4Gb/s and above. In all cases, high reliability, low power, and low cost are the most important issues. Reliability is particularly important in the high current laser drivers.

Circuit and test results.	Ref.
(1) PRBS generator/error detector. 1023 bits at 2Gb/s, UFL.	[74].
(2) Fibre-optic system. Prototype made with 40 GaAs ICs, 350MHz, BFL.	[22].
(3) Optical transmitter. 1.1Gb/s, 20mA peak current, SCFL.	[144].
(4) Buffer store for optical transmission. 2Gb/s,,1μm CDFL.	[122].
(5) opto-electronic receiver. Photodetector, amplifier, demux, 6GHz, DCFL.	[98].
(6) complete transmit/receive chip set. 2.4GHz	[34].
(7) optical transmission system. 2.4Gb/s, CDFL.	[130].
(8) Transceiver chip set for computer networks (HOT ROD). 1Gb/s.	[19].

Table 2.6 Optical communication circuits

Most ambitious among the 2.4Gb/s systems was the complete GaAs chip set from Hitachi (Item 6 [34]), available commercially since late 1988. This consisted of seven chips, which included laser driver, 4:1 MUX, 1:4 DEMUX, decision circuit, pre-amp, AGC amp, and main amp. Maximum operating frequency of 3GHz was achieved by the 1:4 DEMUX, at a power dissipation of 1.5W. 0.8m gate lengths were used, although the logic configuration was unclear. A commercial silicon bipolar chip set was not available for direct comparison. However, a review of laboratory circuits based on "standard" 2μm silicon bipolar processes [113], showed that silicon was well capable of the necessary speed, and in the case of 2.4Gb/s, dissipated less power with the laboratory samples. Although the comparative study gave no indication as to the practical utilisation, reliability, and cost of these laboratory chips, it would appear that in a multi-chip system, GaAs chips do not have conclusive advantages over silicon at the 2.4Gb/s rate.

Above 2.4Gb/s, 9.95Gb/s was considered to be a possible future requirement [58,113]. Silicon was also considered to be suitable for this bit rate by using advanced self-aligned gate processes [113]. GaAs laboratory circuits have already demonstrated their capability with circuits such as the 26GHz frequency divider and 12Gb/s 2-bit MUX/DEMUX mentioned earlier. Silicon have produced 5.8Gb/s 2-bit MUX and 8 Gb/s frequency divider ICs. From these figures, one can conclude that silicon has not yet achieved 10Gb/s, but has the necessary potential.

The possibility of integrating optical and logic functions onto the same GaAs substrate may prove to be more important than ultimate circuit speed at the moment commercially. This had been realised in the 6GHz integrated receiver (Item 5, [98]). The chip consisted of a photodetector, a three-stage amplifier, and a DCFL-based demultiplexer. This was not possible in silicon, and should point the way forward for ICs where opto-electronics, switching, and even MMIC functions exist on the same chip.

On the computer networking front, two GaAs ICs provide the fastest serial data link commercially available in any technology. The "Hot Rod" chip set (Item 8, [19]) consists of a GA9011 transmitter and GA9012 receiver, based on a proprietary TTL-compatible GaAs technology from Gazelle. They were capable of transceiving data at 1Gb/s, compared to the 175Mb/s of the fastest silicon chip set from Advanced Micro Devices [3]. The silicon chip set is based on the Fibre Distributed Data Interface (FDDI) standard, defined by an ANSI committee [61]. The GaAs chips were designed on a similar standard, but with a 40-bit interface, rather than 10-bit of the silicon chip set. They were aimed at easing communication bottlenecks between workstations in a distributed network, and at critical server nodes. The next generation Hot Rod will aim at 10Gb/s.

Objectively, the lack of comparable circuits means that this chip set may well be ahead of its time, and may have difficulty in finding networks that require this bit rate. Also, from earlier discussions concerning optical fibre transmission, present silicon technology is well capable of achieving similar bit rates. On the other hand, since all the error encoding and decoding functions are fully integrated into the GaAs chips, it can be used by any networking protocol designed for 1Gb/s bit rates or beyond. Therefore the chip set may have a new standard all to itself until a faster chip set is available. This was the case when the AMD chip set was introduced.

Supercomputers

Supercomputers and mini-supercomputers have found wider general use in the 80's in numerically intensive computing, interactive design and graphics. They have been based on massively parallel architectures and large number of silicon ECL processors and physical memories, and usually require liquid cooling systems. The processors are normally LSI to MSI integrated circuits designed for highest switching speed at the expense of power dissipation. GaAs was therefore considered to be particularly suitable, owing to its speed potentials.

Two systems have been announced that will use GaAs technology. They are listed in Table 2.7 below, together with their projected performance. The subject of computer benchmarking is complex and the figures should be treated only as a guideline. Cray-3 was claimed to be the fastest supercomputer if the figures are realised. The Prisma system is targeted for the mini-supercomputer market. The fact that the maker of the first supercomputer is strongly backing the use of GaAs is in itself significant, even though its real market potential is likely to be small compared to other GaAs applications.

Computer	Description	Ref.
Cray-3	16 processors, 16 giga-flops.	[68].
Prisma	Based on Sparc RISC, 250 MIPS.	[50].

Table 2.7 Supercomputer systems

The backing of GaAs for Cray-3 was based on a comparative study with silicon ECL [68]. The study found that silicon could match a GaAs system in performance only by using more chips and power. In addition, it was considered that experience learned from designing the Cray-3 would be useful for Cray-4, possibly based on a MODFET or HEMT technology (the theory of MODFET/HEMT can be found in [30]), which have many similarities to GaAs MESFET technology.

Test results are not yet available on either system. Cray-3 has been delayed by circuit design and board level problems [59]. GaAs circuit yield, however, was not found to be a problem. The success or failure of these computers will essentially determine the future prospects of GaAs in this area.

Gate array and standard cells

Apart from custom military and civil communication circuits, gate arrays and related semi-custom have also been a major focus of GaAs foundries, especially for start-ups like Vitesse and TriQuint. For those companies which do not manufacture silicon, it is a commercial necessity to diversify.

In silicon, gate arrays have been used in all types of applications that require a fast design turnaround time. A fair number of GaAs standard cell and gate array families have been published. Most of them are commercially available to compete head-on with silicon. One example of laser programmable logic devices (PLD) also exists. These are listed in Table. 2.8 below.

Details	Commercial availability/Ref.
(1) 6K gates, SDFL. 1.1ns, 500μW/gate at fanout=2 plus 1500μm line. 12×12 multiplier, 82.5ns and 1.7W.	No.[105].
(2) 15K gates, DCFL. 68ps unloaded. D-flip flops, 450ps delay.	Vitesse.[73].
(3) 3K gates, E/D BFL. 125ps unity fanout. divider at 1 GHz, control chip for test system at 750MHz, ECL/TTL compatible.	TriQuint.[119].
(4) 500 gates, 0.5μm BFL. 44ps unloaded, 5mW. counter at 4.1GHz, 2:1 MUX at 4.6Gb/s, 1:2 DEMUX 5Gb/s, 4:1 MUX 3.6Gb/s, ECL compatible.	NEC.[50].
(5) 6K, CDFL. 76ps unloaded, 1.2mW. 16-bit serial/parallel circuit at 852MHz and 952mW, ECL compatible.	Toshiba.[136].

Table 2.8 GaAs gate arrays

Details	Commercial availability/Ref.
STANDARD CELLS	
(1) BFL gates, buffers, flip-flops. 80ps unity fanout, 5mW. 4-bit serial/parallel register at 1.3GHz, flip-flops at 2GHz, ECL compatible.	Thomson.[118].
(2) 22K gates max., logic, I/O, RAM, ROM, ALU, register file. DCFL, 100ps, 0.3mW 2-input NOR. 3.5GHz clock rate.	Vitesse.[138].
MACROCELL ARRAYS	
(3) 250 gates, SCFL. 74ps loaded delay, 2.4mW/gate. Flip-flops at 7.5GHz, 2×2X-point switch at 2GHz, ECL compatible.	No.[55].
PROGRAMMABLE LOGIC DEVICES (PLD)	
(4) 165 MHz clock rate, t_{pd}=7.5ns. Contains 6 buried registers and 8 output logic registers. TTL pin compatible.	Gazelle.[27].

Table 2.9 Standard cells, macrocell arrays, and PLDs

In silicon, two ECL macrocell arrays have produced impressive results. The first one consisted of 7K gates configurable as functional cells such as latches and logic gates, based on current mode logic [52]. A 24-bit parallel multiplier was fabricated as a test chip and produced a multiply time of 12.8ns. This compares favourably with a 12×12 multiplier, designed with a gate array, listed in Table 2.8 (item 1). An unloaded inverter gate delay of 50ps and a power dissipation of 1.84mW were also achieved. Another macrocell array had produced 43ps current mode logic basic gate delay and 4.3ns 16-bit multiplication [132]. No results are available for comparison with a GaAs macrocell array multiplier. However, the 30ps basic gate delay and 7.5GHz flip-flop toggle frequency of the GaAs macrocell array, based on a "low power" SCFL configuration (item 1, Table 2.9), showed decided advantage over these two recent silicon examples. Power consumption per gate of the GaAs cells is also 40% lower than the best of the two silicon macrocells. The significant similarities between SCFL and ECL has been mentioned earlier and their gate functionalities are comparable. The small differences between the gate delays of the GaAs and silicon macrocells are less significant in a real system context when interconnection delays are included. Therefore the power advantage will again be more apparent than speed.

In the area of gate arrays, they tend to be slower than macrocells owing to the lower functionality per cell and hence longer interconnections on-chip. The loaded gate delays of the GaAs arrays listed in Table 2.8 range from about 100ps for unity fanout, to 588ps for a fanout of 3. Most of the arrays have been demonstrated with circuits of LSI complexity. Most notable of these was the use of a 3K gate array for a digital tester control IC (Table 2.8). 2295 gates were utilised in the array and it operated up to a clock rate of 750MHz.

Virtually all major semiconductor companies offer silicon gate arrays commercially. Gate speed and power figures are broadly similar to the GaAs arrays listed here. In terms of complexity, 30K cells have been available in silicon [12], compared to 15K in GaAs [73]. At the chip level, a previous review of gate arrays in general concluded that the GaAs arrays based on MESFETs would result in more efficient gate utilisation, compared to silicon ECL [12]. Therefore total power dissipation is likely to be 1/4 to 1/3 of equivalent functions in silicon.

In the case of standard cells and PLDs, speed performance appeared to be between those of gate arrays and macrocell arrays. The 22K-gate commercially based standard cell family leads the way in terms of complexity in GaAs, with claimed future increase to 50K gates, based on DCFL. Direct comparison of experimental results with silicon is not possible owing to the different cell functions and designs.

Test and measurement systems

In previous sections, the progress of GaAs in the main functional areas have been reviewed. Most of the published laboratory results have been purely for demonstration and research purposes. Examples of insertion into real systems mentioned so far have been largely in optical fibre transmission systems. A small but significant area of digital GaAs IC usage is testing and instrumentation. SSI to MSI custom and standard ICs were used in critical functions where silicon was unsuitable. Some of these systems are listed in Table 2.10.

System and ICs used	Ref.
(1) Anritsu 5GHz pattern generator and error detector. T flip-flops from NEC.	[108].
(2) Interface Technology, 8-channel pattern generator, 1GHz. Shift registers from Gigabit Logic.	same.
(3) Outlook Technologies, 250MHz pattern generator. Output drivers from Gigabit Logic.	same.
(4) Astro Design Inc., video signal generator. Standard logic ICs from Gigabit Logic.	same.
(5) Hewlett Packard, 1GHz digital oscilloscope. Custom sample/hold and logic ICs.	same.

Table 2.10 Test and measurement systems

Noting the gap between circuits and system insertions, at least 11 military projects had also been initiated to replace or implement specific systems with custom GaAs ICs, despite large cuts in U.S. government funding for digital GaAs. These include replacement of CMOS components in a Digital Map Computer for aircrafts and in the On-board Processor in a spacecraft, a signal processor for anti-tank missiles, and modem and synthesiser for army Small Unit Radio [110]. For these DARPA funded programs, companies are expected to go forward with qualification and installation on successful demonstration of the components. These projects will possibly determine the long predicted broad acceptance of digital GaAs technology.

2.2. Bit serial circuits and systems

Bit serial architectures have been well explored in silicon. It was generally accepted that they are particularly suitable for certain types of signal processing applications, and in general provide a compromise between functionality per chip, and throughput rate and performance. This section looks at the basic concepts of such architectures, and review previous results in silicon to illustrate the strengths and weaknesses of bit serial in general.

By definition, bit-serial systems process a data word one bit at a time. (Some degree of parallelism has emerged in recent designs, discussed later on). The data word is spread out sequentially in time, and may appear least significant bit (lsb) or most significant bit (msb) first, depending on the implementation. Fig. 2.16 illustrates the concept with an arbitrary operator and compares it with a bit parallel system.

A common feature of bit-serial systems is the use of one or more framing pulses to indicate the position of the system word. They are often chosen to be coincident with the lsb or msb of the word. The most popular number format is two's complement, chosen for its arithmetic efficiency. The main features of bit-serial systems are summarised below:

(i) Msb or lsb first, two's complement number representation.

(ii) Fixed point or floating point.

(iii) Synchronous operators with fixed latencies.

(iv) Multiple precision.

In addition, timing control of operators can range from single phase, 2-phase non-overlapping, to 4-phase clocking in silicon. As an example, Fig. 2.17 shows the timing diagram of an lsb first system which adopts a two-phase non-overlapping clocking scheme. This is essentially the convention adopted for the GaAs circuits discussed in later Chapters.

One of the earliest works on bit-serial signal processing systems was that of Jackson et. al. in 1968 [62]. Several fundamental properties were identified from this work that remain relevant to all bit-serial systems:

(i) Digital filters can be made up of simple circuits (operators).

(ii) The operators are highly modular.

(iii) They can be easily adapted to a wide range of filter forms.

(iv) The operators can be time multiplexed to increase hardware efficiency.

In this pioneering work of Jackson, various filters of first, second, third, and sixth order were integrated and multiplexed into a touch-tone receiver system. Since then, significant advances have been made in bit-serial systems both in research and commercial applications. Some of the most important examples are listed in Table 2.11.

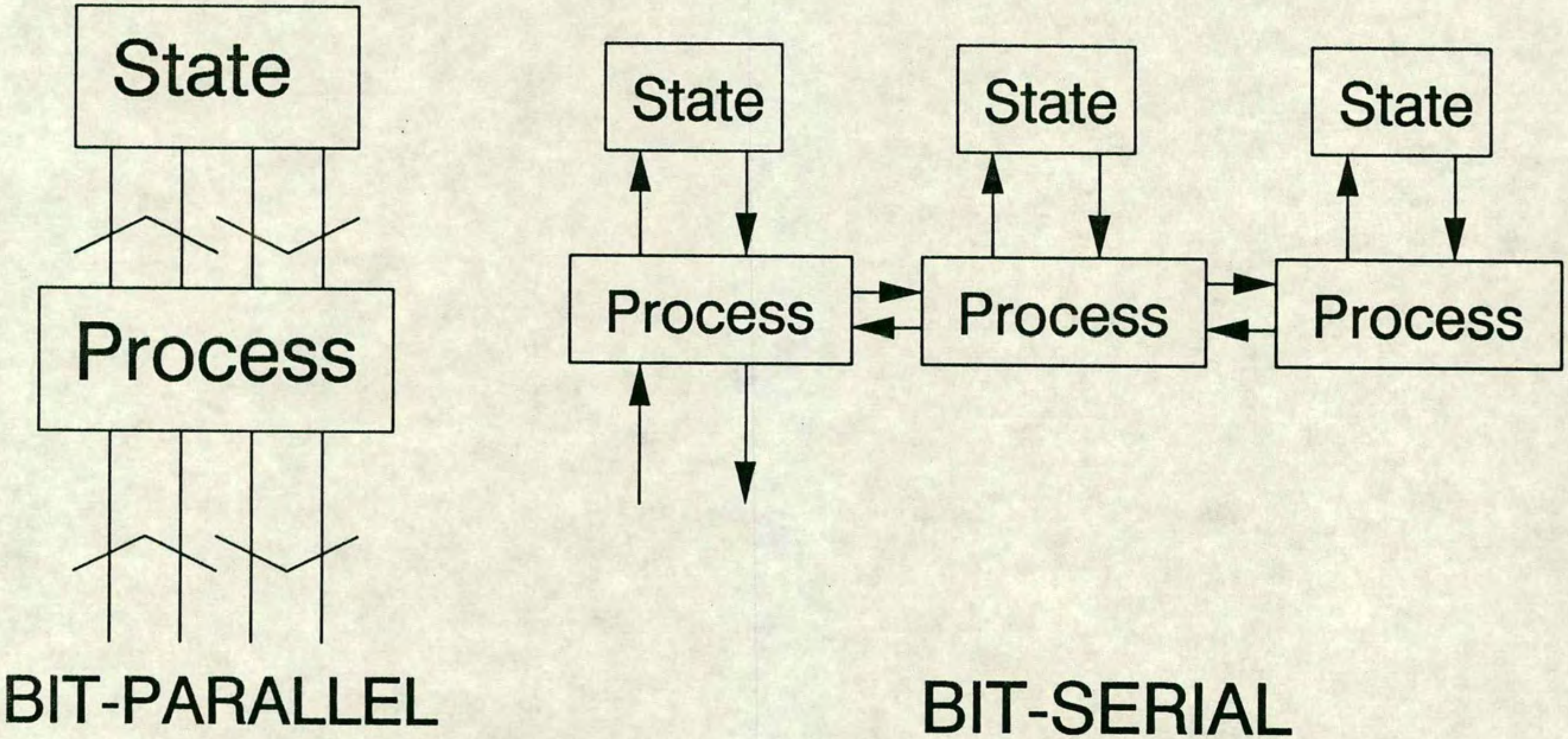


Fig. 2.16 Bit-serial and bit-parallel systems.

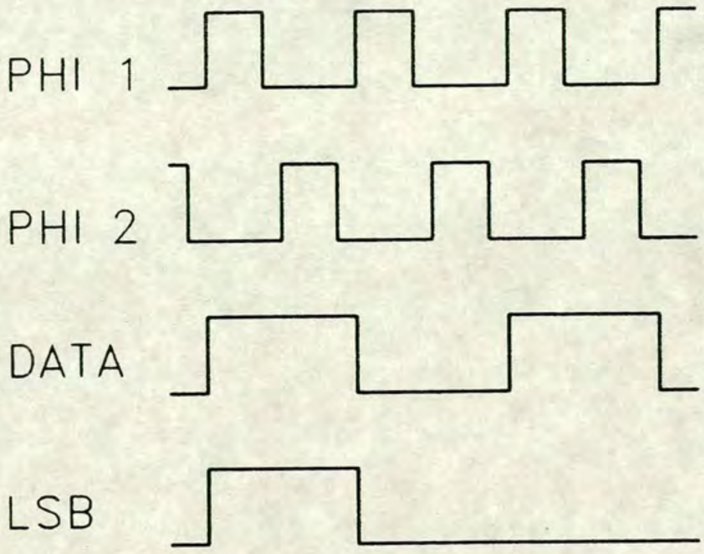


Fig. 2.17 Example of bit-serial system timing.

The first comprehensive analysis of bit-serial systems in comparison to bit-parallel, was given by Lyon [78]. General issues such as timing, hierarchy, formats, pipelining strategy, and functional parallelism, and multiplexing, were covered. Functional parallelism is the use of a large number of simple operators connected in parallel, with appropriate logic to combine the partial results generated by each operator. This type of array structure is used to overcome the disadvantages of bit-spreading on a single data stream for very high bandwidth applications. One example of this was the 64-point FFT processor of Powell and Irwin [109]. At the other extreme, operators can be shared and used to process unrelated data by multiplexing several slower channels, thus maximising hardware efficiency.

The first bit-serial Booth's multiplier was built by Lyon [77]. Subsequent to this, the FIRST and SECOND [123] silicon compilers have exploited and advanced the timing and modularity of bit-serial architecture to good effect. These projects have also resulted in a different type of parallelism by introducing a twin-pipeline technique and single-phase clocking. In a similar direction, higher radix computation have also been used to increase system throughput. Serial multipliers were extensively investigated by Smith [123].

Circuit/system	Year and ref.
(1) High-pass, low-pass, band-pass, and band-rejection filters in a touch-tone receiver.	1968.[62].
(2) Two's complement pipeline multiplier study. First bit-serial implementation of Booth's recoding algorithm.	1976.[77].
(3) 64 point Fast Fourier Transform processor.	1978.[109].
(4) Matrix-vector multiplier. Multiplies 3X3 matrix with a 3×1 vector at 6MHz.	1985.[67].
(5) Serial/parallel multiplier. Novel "automultiplier" architecture.	1987.[124].
(6) Twin-pipe, single phase clocked multiplier.	1988.[123].
(7) Vector quantiser for real-time image coding.	1989.[111].
(8) Reconfigurable multiplier. Designed for testability and reconfigurable to utilise good ones.	1989.[9].
(9) Digital audio D/A conversion IC. Includes FIR and pulse-density modulation D/A converter on one chip.	1988.[97].

Table 2.11 Silicon bit-serial circuits and systems

Judging from the range of applications where bit-serial techniques were used, as evident in the Table, its complementary role to bit-parallel computation has been well demonstrated. Perhaps the most recent and significant of these was the use of the bit-serial format in ICs used in commercially available compact discs players. In the Philips D/A conversion chip, SAA7320 [97,129], a 16-bit four-times oversampling FIR filter, 32-times oversampling linear interpolator, noise shaper, and a 1-bit switched capacitor D/A converter and analogue low pass filter, were integrated into one chip. The chip uses a high speed clock of 11.3 MHz, and a serial word format called Inter-IC Sound (I^2S) [54,139]. The I^2S bus consisted of an msb first two's complement serial data of variable real data word length, a Word Select line (to indicate which of the stereo channels is being sent), and a clock. Previous bit-parallel systems required the use of 3 chips to perform the same circuit functions. The use of bit-serial techniques in this instance has resulted in substantial reduction in physical board space, and an improvement in linearity and perceived sound quality.

In general, these systems have exploited the two inherent and unique advantages of bit-serial architecture:

(i) Simple signal interconnections between operators and chips — only a signal wire and a framing wire are all that were necessary.

(ii) The functional efficiency of operators — only one adder is needed in an n-bit bit-serial multiplier, as opposed to n adders in a bit-parallel carry multiplier. More functional elements can be placed in one chip.

These properties result in more efficient and cost-effective use of silicon and board area. Set against these is the need for a high speed clock, in order to obtain the same system throughput rate.

2.3. Discussion

In the previous sections, the basic concepts and important results of digital GaAs ICs, and to a lesser extent, bit-serial circuits and systems, have been introduced and reviewed. It is the aim of this section to make observations on the general state of and the technical problems facing GaAs technology, and discuss how bit-serial architecture can be used to advantage.

Concerning the state of GaAs technology as a whole, its volatility is evident. One of the facts that have emerged from the review of GaAs systems is that GaAs IC insertions into real systems were almost exclusively of SSI to MSI complexity, and were used only at critical functional bottlenecks where silicon proved unsuitable. The commitment that a system designer had to make is still too big for most systems to contemplate a custom or even semi-custom GaAs design. In many ways, the announcement of the Cray-3 had helped to sustain the initial technical and commercial interests in digital GaAs MESFET technology, and prevented it from being relegated to purely an interim **research** technology before heterojunction devices are mature enough for manufacturing. Apart from the obvious silicon competition, other problems have prevented it from "taking off":

(i) Volatility of foundries.

Strategic alliances exist (Cray/Gigabit Logic, and Motorola/Honeywell) which improve the situation slightly. However, most major semiconductor companies have pulled out of digital GaAs foundry services.

(ii) Lack of second sourcing — there was only one announcement of linking of two GaAs processes [28].

(iii) Lack of a complete product line of circuit building blocks.

The high cost of GaAs parts is clearly important, but will fall quickly if volume production is achieved. On the technical side, a number of observations can be made from previous sections:

(i) Speed advantage over silicon decreases quickly with increasing integration level.

This is partly due to the use of MESFETs in GaAs, which have lower transconductances. In addition, direct coupled FET logic (DCFL), the family most suitable for VLSI, has high fanin/fanout sensitivity. As functional complexity increases, so does gate delay (or number and size of gates used to compensate). This is confirmed by the results of large GaAs memories, multipliers, and microprocessors discussed earlier. Also, a direct comparison of identical 4-bit ALU designs fabricated with a 0.6μm silicon NMOS process and a 0.6μm recessed (non-self-aligned) gate GaAs process showed only 1.5 times speed advantage in favour of GaAs [92]. The power advantage was found to be more significant. Both silicon and GaAs technologies have improved significantly since the time of this direct comparison. Nevertheless, it implies that the use of exactly the same circuit architecture in GaAs only results a slight improvement in speed, compared to even silicon NMOS technology.

(iii) Compatibility.

Although most ICs were designed for ECL or TTL level and power supply compatibility, power and performance suffer as a result, again owing to the exclusive use of GaAs MESFETs. This further reduced the gap between silicon and GaAs parts.

(iv) Low integration level and yield.

The lower integration level in GaAs implies that more ICs are needed for a complex system function. This clearly increases interconnection, power, and speed penalties. In military projects, the standard cells approach was found to be most popular where performance and complexity were important, while gate arrays and macrocell arrays were used for slower functions and fast turnaround designs [39].

The processing and yield related aspects of GaAs are discussed in Chapter 4. In the related subject of noise margins, it was not identified as a major problem in published results. In fact, other factors have more drastic effects on yield (Chapter 4). In practice, the "relative" noise margins within a voltage swing are more important than the "absolute" values. For the same reason, silicon ECL has a smaller logic swing and hence lower absolute noise margins than CMOS, but does not present problems in use. GaAs DCFL has similar percentage noise margins as silicon ECL. More details of DCFL design are given in Chapter 3.

(v) Chip interconnection and packaging.

At speeds of a hundred MHz or more, off-chip interconnection becomes a major concern both in silicon and GaAs. This is even more severe in GaAs because of the properties of MESFET and DCFL mentioned earlier. A GaAs output buffer designed for a 50Ω transmission line can dissipate from 20 to 60mW. When signal rise times decrease, cross-talk and synchronisation become problematic, especially with the larger number of chips involved owing to the lower integration level, and system speed would need to be sacrificed. Delay equalisation on a large scale, as was necessary in Cray-3, had proved to be both expensive and technically difficult. In a bit-parallel system, an n-bit wide data bus may be connected to several destinations within or beyond a board, with each bit-line requiring proper termination. Together with the need for chip packages and logic boards to maximise proximity and minimise wire inductances, these problems have proved to be challenging theoretical topics in their own right [39].

It is clear from these points that different architectural approaches to GaAs, and indeed high speed technologies in general, are necessary. Complex bit-parallel systems so dominant in silicon are simply not practical in GaAs. The fact that there is no existing commercial or research effort to make a single-chip Complex

Instruction Set Computer (CISC) processor in GaAs implies that even if such integration level could be achieved, the speed advantage might not be significant enough to justify the effort. The 32-bit GaAs RISC chips clearly point the way towards future microprocessors, although competition from 64-bit silicon ECL chips again limits its real market appeal, as mentioned earlier.

Another useful example of system performance comparison was a simulated study of computers based on GaAs MESFET, HEMT, and silicon CMOS and ECL ICs [38]. It was found that 1μm GaAs MESFET was 1.5 times faster than 1μm ECL at the system level. Identical logic design and architecture were used in this comparison. It is clear from the delay analysis of this study that only limited advantage can be gained from faster logic gates because off-chip delays are not scaled by the same amount. Off-chip driver delays of the MODFET/HEMT ICs were only 50% less than those of ECL in this study, although loaded gate delays were 75% smaller than ECL. The imbalance is worsened as on-chip delays reduce even further. In addition, chip-to-chip line delays do not scale as easily as gate delays. These limiting factors can only be alleviated by using architectures that maximise on-chip processing functionality, and minimise off-chip interconnections.

In the work presented in subsequent Chapters, the adoption of a bit-serial architecture was proposed as a possible solution to these problems. The inherent properties of bit-serial processing discussed earlier clearly matches those required by high speed technologies. The simpler logic design, and ease of pipelining and partitioning would allow complex and modular structures to be integrated on a chip to perform signal processing functions.

Chapter 3

GaAs DCFL design

GaAs DCFL was reviewed as a logic configuration in Chapter 2. In this Chapter, the basics of DCFL design in general, and in relation to the Honeywell process in particular, are discussed. Similarities and differences between DCFL and silicon NMOS static circuits are highlighted. This serves to provide an NMOS designer with the necessary knowledge to crossover to GaAs DCFL design. The methods and ratioing stated will be taken as standard rules that apply in the rest of this thesis, unless otherwise stated.

3.1. The DCFL inverter and NOR gate

The DCFL inverter (Fig. 3.1) models on the silicon NMOS static inverter, and has been introduced in Chapter 2. However, two major differences between the MESFET and the MOSFET must be born in mind: the clamping action and the gate current of the Schottky barrier gate when it is forward biased. These result in several DC and transient effects unique to GaAs DCFL, as will be seen later on.

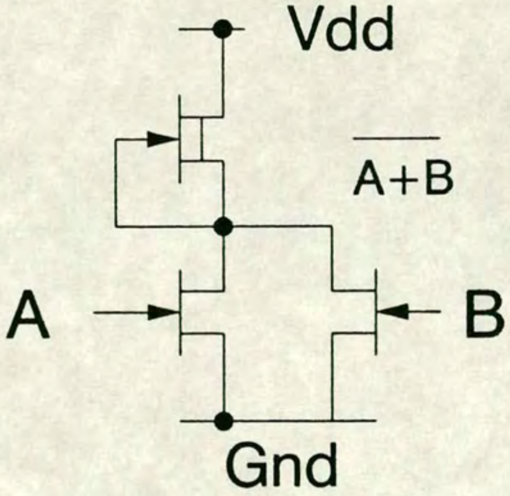


Fig. 3.1 DCFL NOR/inverter gate.

The traditional but inaccurate way of numerically modeling the MESFET was as a "modified" JFET device. A JFET device has three regions of operation: cutoff, ohmic or active, and saturation regions. The regions are illustrated in Fig. 3.2. However, the most commonly used MESFET model, based on Curtice [20], has only 2 regions: cutoff and active. The main difference is quantitative, especially in the vicinity of the transition between the active and saturation regions (the "knee"), as shown in Fig. 3.2, normalised to the JFET curve for comparison. For ease of discussion, the MESFET will be thought of as having the same three regions as a JFET. Fig. 3.3 shows a measured dc curve for a 20μm D-MESFET device based on Honeywell data.

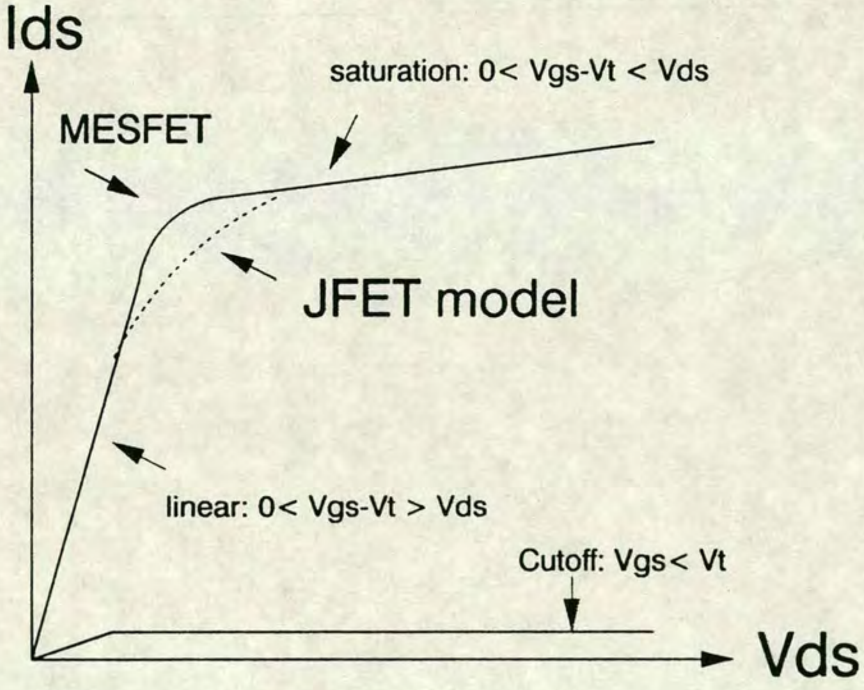


Fig. 3.2 Qualitative difference in I-V curve between JFET and MESFET, and regions of operation

With respect to the ratioing of a NOR/inverter, it can be approximately calculated by equating drain currents as in silicon. But unlike NMOS, the forward gate currents drawn by driven gate inputs do affect the DC conditions and need to be taken into account. It is convenient at this point to define the following ratioing formula and notations, which will be used in subsequent discussions:

$$\beta_R = \frac{\frac{W}{L}_{\text{pull-down}}}{\frac{W}{L}_{\text{pull-up}}}$$

where:

W = gate width

L = gate length

$\frac{W}{L}$ = aspect ratio of transistor

The gate length and width of a transistor are defined as real dimensions. (The difference between real and drawn dimensions are small and are usually ignored in the Honeywell process.) In the Honeywell process, the following figures apply:

$L_{E-MESFET} = 1\mu m$
$L_{D-MESFET} = 1.5\mu m$
$\beta_R = 3$

With these values, it follows that:

$$\frac{W_{E-MESFET}}{W_{D-MESFET}} = 2$$

The β_R value of 3 takes into account noise margins, gate current loading, and process threshold variations. If a more negative V_t voltage is used for the D-MESFET, then a higher β_R ratio is necessary, as in NMOS. With the Honeywell rules, the process gain factor K' of the E-MESFET is about twice that of the D-MESFET, while their threshold voltages V_t are +0.2V and −0.5V respectively. With these figures, the DCFL inverter logic voltages are 50mV for the LOW state and 750mV for the HIGH state, for a fanout of unity.

At this point, it is useful to define the meanings of fanout and fanin of a logic gate when they are specified as quantities:

$$\text{fanout} = \frac{\text{(Total W of driven E-MESFETs)}}{\text{(W of pull-down device in the driving logic gate)}}$$

fanin = Total number of pull−down devices in a logic gate

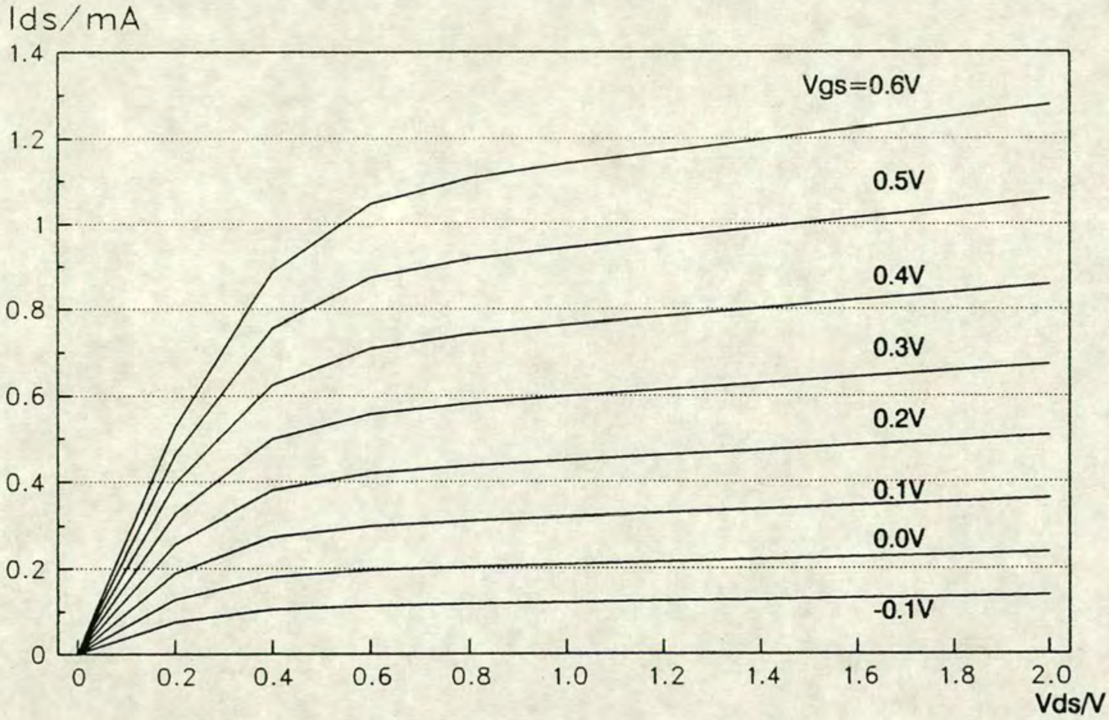


Fig. 3.3 DC characteristic of 20μm D-MESFET

The choice of the value of V_{dd} is determined by the following criteria:

(i) V_{dd} should be high enough to ensure the pull-up D-MESFET is in saturation, as in NMOS. This is easily met by by having V_{dd} greater than the forward diode clamping voltage.

(ii) Unlike NMOS, it also needs to supply enough DC current to keep the fanout Schottky gates forward biased when the output is HIGH.

(iii) IR drop on the on-chip supply rail and power supply variations need to be taken into account.

(iv) Speed/power requirements need to be satisfied. Higher speed can be obtained by using a high V_{dd}, provided that the necessary increase in β_R (to maintain noise margins) does not offset the gain.

(v) A high V_{dd} ($> 2\times(V_{gs}-V_t)$) would aggravate hysteresis (not equivalent to body-effect in NMOS) in the D-MESFET pull-up device. A more detailed discussion on hysteresis is given in Chapter 4.

Taking these into account, DCFL should operate satisfactorily with V_{dd} above 1.2V, while 2V is a reasonable upper limit. Honeywell rules specifies a value of 1.7V±10%.

In Chapter 2, the high sensitivity of the transient behaviour (gate delays) of DCFL to fanout, compared to other buffered logic families, was mentioned as one of its drawbacks. Another important problem is the sensitivity of DC noise margins to fanin and fanout. These are again the direct results of using MESFETs. The mechanisms behind them are worth discussing.

First consider the DC problems. The effect of increased fanin is directly due to leakage currents. Even when the gate-source voltage of an E-MESFET is below the threshold voltage, a leakage drain current still exists. This is more severe in a MESFET because the channel region is n type, as opposed to p type in an n channel MOSFET. This implies that for a NOR gate with all pull down devices supposedly OFF, each increase in fanin would result in a slightly lower output logic HIGH voltage. With the Honeywell process, an E-MESFET in the cutoff state is approximately equivalent to a 50kΩ resistor, and would result in about 10mV loss in logic swing for each added fanin. Therefore DCFL design rules normally limit gate fanin to a maximum of 4 to 6, depending on the process. This in turn results in limited logic input functionality per gate.

The effect of increased fanout on logic HIGH voltage is more severe. To illustrate the basic cause, the driving pull-up device is replaced by an ideal current source and the fanout gates are thought of as Schottky diodes, as shown in Fig. 3.4. As the fanout is doubled, the current flowing into each fanout is halved. This corresponds to a lower forward gate voltage on the diode curve, thus resulting in a lower logic HIGH.

In practice, the D-MESFET pull-up device has a finite output conductance, and the drain current increases as the source voltage is dropped. However, with improved device fabrication, the output conductance is likely to be reduced in the future, and this current compensation would be small.

A second and more important phenomenon tends to worsen the situation. This is due to the fact that a MESFET has finite gate-source resistances. The overall forward characteristic is significantly different from that of a Schottky diode, as shown in Fig. 3.5. Note the much more linear form of the MESFET forward current. Therefore the reduction in gate current due to increased fanout would result in an even larger drop in forward voltage and logic HIGH voltage than with

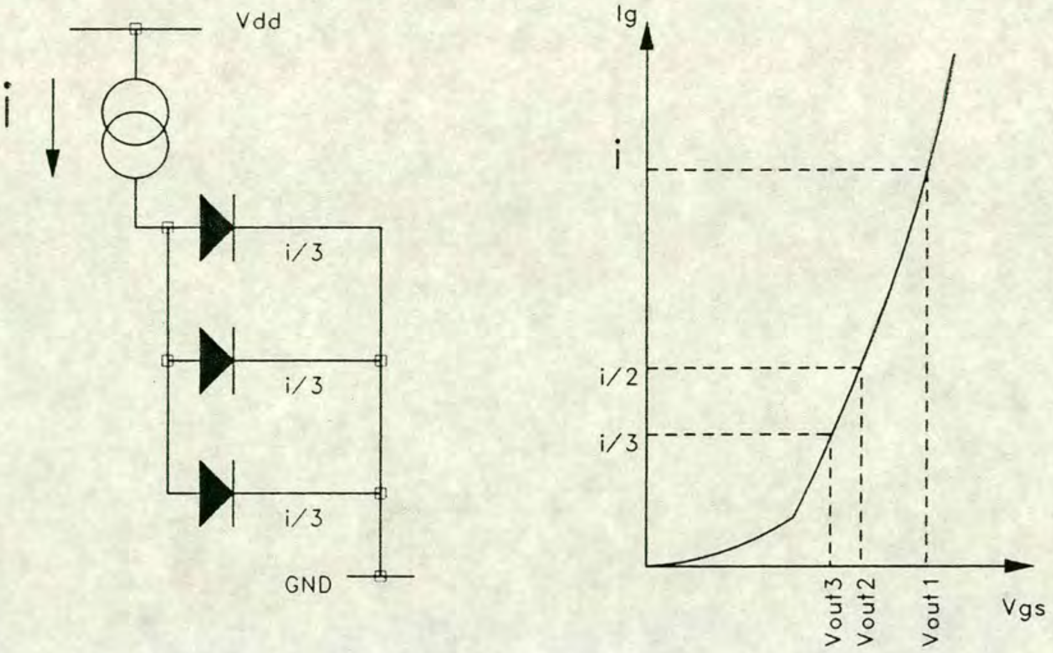


Fig. 3.4 Logic HIGH degradation with increasing fanout.

a Schottky diode. To illustrate this quantitatively, simulated outputs for inverters with fanouts of 1 to 3 are shown in Fig. 3.6. This also contributes to the better noise margins of Schottky diode FET logic, SDFL, reviewed in Chapter 2. The equivalent model of the MESFET is not discussed at this point. It is sufficient to note that this is the major device factor that causes noise margin degradation, and limits fanout to an absolute maximum of 4 in most DCFL designs. Similar to output conductance, these resistances should reduce with improved processing and scaling of gate lengths. Nevertheless, together with the fanin effects, they limit the functionality of DCFL NOR gates.

The transient effects of increased fanout follow from the current division reasoning. A common and simple delay model can be given by the sum of unloaded gate delay and total capacitive delay (including interconnection and fanout gate capacitances) at the output. In DCFL, as in NMOS, the latter dominates, and is directly proportional to fanout. Self-aligned gate devices have overlap capacitances of the order of 10 fF for a 20μm device. Capacitance between closely spaced interconnections are more significant and can amount to several hundred fF for 100μm lengths, and are therefore the main RC delay contributions. At unity fanout, loading delay typically represents 50% to 70% of total gate delay. In addition, the lower logic HIGH voltage with increased fanout means that the driven

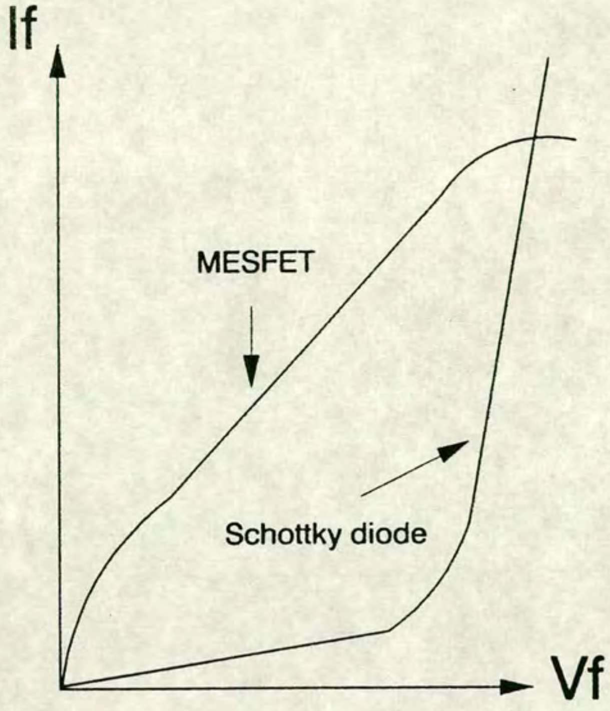


Fig. 3.5 Comparison of MESFET and Schottky diode forward characteristics.

gates are turned ON weaker, increasing their basic (unloaded) gate delays. The basic β_R value of 3 takes into account noise margins in this situation. It is not general practice to increase the size of driven gates in GaAs, unlike NMOS, to compensate for the lost swing. This would simply increase the fanout loading on the driving gate and reduce logic swing and speed even further. However, increasing one driven gate size at the expense of another less critical gate, but without increasing β_R, is a valid technique of increasing speed.

The effect of extra fanin is less severe and is mainly due to the additional drain-source capacitance. Drain-source capacitance is similar to or smaller than gate capacitances. A distributed logic gate, where the fanin MESFETs are physically distant from each other, is possible in DCFL, and would clearly increase delay contributions from fanin.

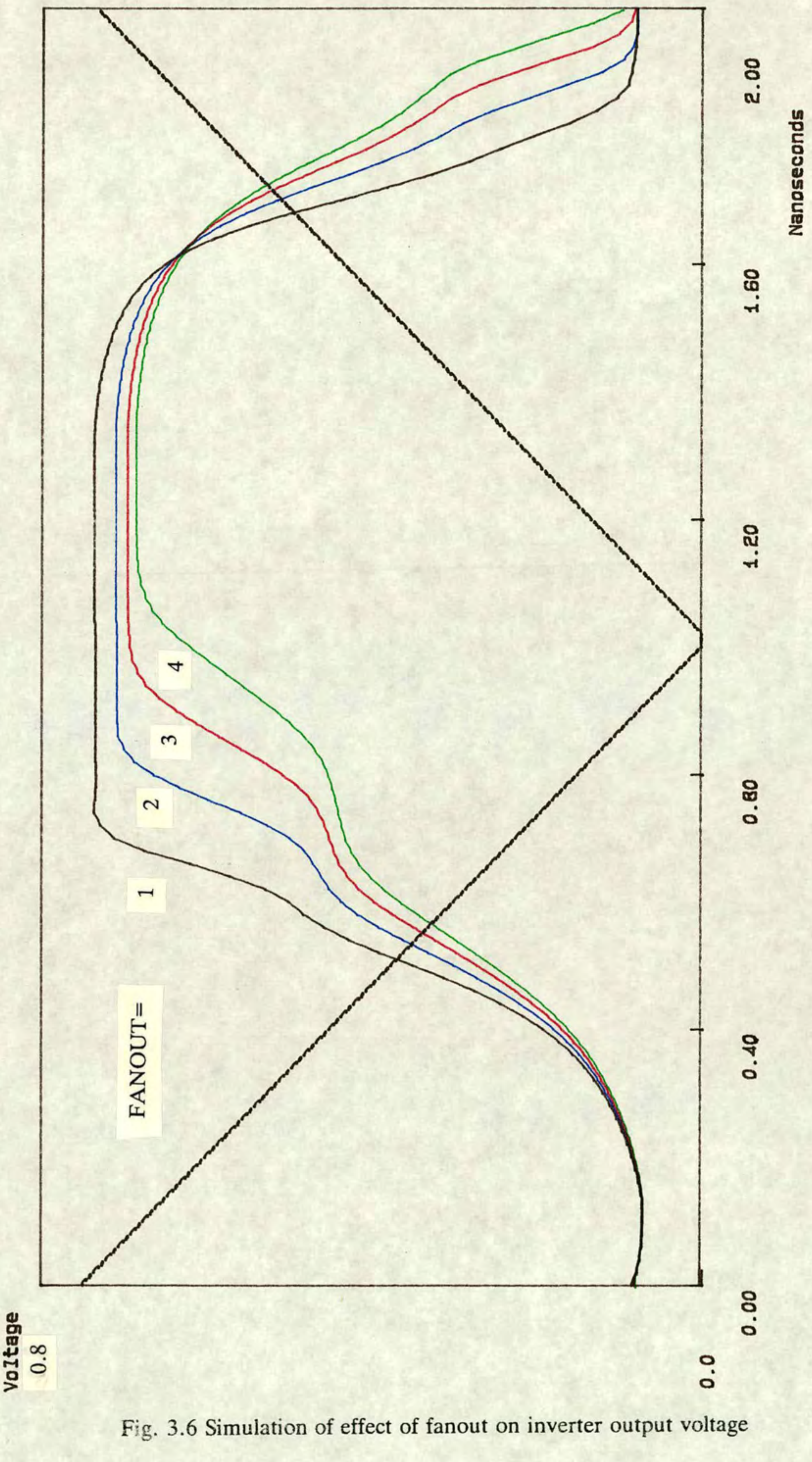


Fig. 3.6 Simulation of effect of fanout on inverter output voltage

3.2. Other logic gates

NAND gates with two pull-down devices in series are permissible in DCFL (Fig. 3.7), but are generally best avoided. The disadvantages of the increase in sizes and capacitances from the doubling of gate widths are clear. The other drawback is due to the combination of gate voltage clamping and the elevated drain voltage of the upper E-MESFET device. The upper device is significantly weaker than the lower device, dominating the overall switching delay. Unlike a MOSFET which can swing to V_{dd}, a reduction of 50mV in gate-source voltage corresponds to about 20% reduction in static drain current. The device would need to further increased in width to compensate if speed is critical. Noise margins are accounted for in the standard β_R. Alternatively, it can be driven by a "dedicated" gate. Therefore the use of NAND/complex gates should be strictly limited in DCFL. In most cases, it is more efficient to use inverters and an OR gate.

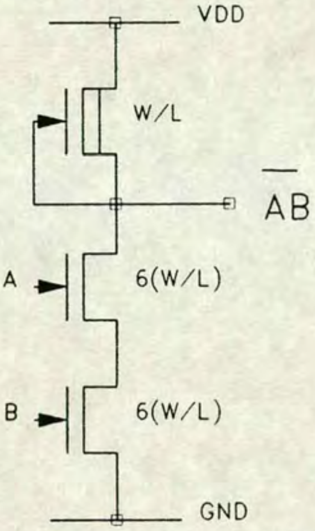


Fig. 3.7 Two input NAND gate in DCFL

The OR gate has less of a problem in DCFL in terms of size and speed. This is shown in Fig. 3.8. But because of the clamping action of MESFETs, none of the inputs can be used to drive other NOR/inverter or NAND gates for correct operation.

3.3. Delay optimisation and interconnections

In general, fanouts should be scaled uniformly in the critical path to less than 3. This should result in fairly random gate width distribution and lead to more compact layout. Fanin signals should also be equalised in time as far as possible, by introducing OR and inverter chains or non-standard sizing, to eliminate glitches. Interconnections should always be included, or estimated as far as possible in the

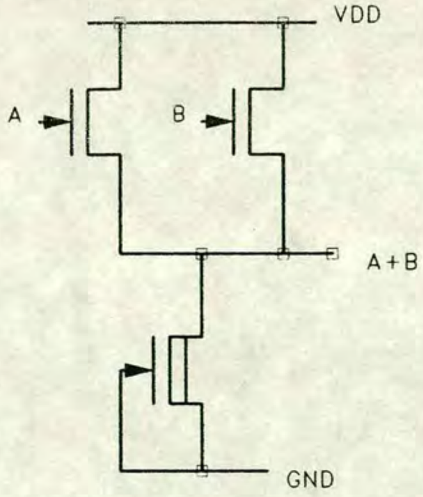


Fig. 3.8 Two input OR gate in DCFL

design process. In cases where a large fanout is unavoidable, super buffers can be used as in NMOS.

Capacitive coupling and switching noise in GaAs MESFET circuits is less severe than silicon MOSFET owing to the semi-insulating substrate and Schottky gate. Nevertheless, gates with large fanouts and long interconnections are prone to cross-coupling with adjacent parallel lines due to fringing (line to line) capacitances, and should be routed carefully.

Finally, transmission line effects of on-chip interconnections can be ignored with the signal rise and fall times involved in this work, which range from 80ps to 200ps, depending on the fanout of the driving gate. For circuits operating at on-chip clock rates of less than 5GHz and chip dimensions of up to a few millimetres, impedance matching is still largely unnecessary [89]. When signals are going off-chip, a microstrip line length of less than a few inches is still satisfactory for GaAs DCFL circuit speeds. Longer lines will require power consuming matched impedance drivers (typically 40mW to 60mW per output buffer). At the circuit speed for which the GaAs cells in this work are designed, packaging and board design are less of a problem. Standard leadless chip carriers (LCC) can be used.

3.4. ECL compatibility

In Chapter 2, the incompatibility of the single DCFL power supply rail with silicon has already been discussed. One common solution is to connect V_{dd} in the DCFL circuits to Ground, and use a negative supply ($-1.7V$ to $-2V$) for the lower rail, previously at 0V. The major drawback of this method is that pull-down devices in DCFL are directly affected by variations on the negative supply. On-chip, these variations should have little effect over noise margins, provided that the IR drop is carefully controlled. However, if a large enough number of chips are involved such that they have different effective on-chip negative supply voltages, the chips will not operate properly. In this case, the normal DCFL positive rail has to be used, in addition to negative ECL supply and termination rails.

The other compatibility issue is the logic swing. The ECL logic swing is typically between -1.7V to -0.9V, i.e. about 0.8V. This is slightly larger than the DCFL swing magnitude of about 0.7V, and a small voltage gain or attenuation is needed at the I/O buffers. The level shifting itself is not very power consuming if the inverted supply arrangement is used. Larger penalty is incurred by the 50Ω termination at the output buffer for ECL compatibility.. If the normal DCFL +ve supply is used, more power is consumed in level shifting between the +ve DCFL and the −ve ECL levels. This work adopted the approach of an inverted DCFL supply, with the assumption that the number of bit-serial GaAs chips are sufficiently small not to cause noise margin problems between chips. The output buffer is discussed in Chapter 5.

3.5. Conclusion

The properties and design methodology for DCFL design have been discussed. The similarities and differences between DCFL and silicon NMOS have been analysed qualitatively. Complex logic functions tend to require more logic gates than in NMOS owing to the constraints on fanin and fanout in DCFL. In theory, all logic functions can be implemented with NOR/inverter gates only — the preferred path in DCFL. This means that little optimisation of logic functions is possible, and they are best implemented in sum of min-term forms. These inherent properties of DCFL necessitated the design of the GaAs cells (Chapter 5) from first principles, and the resulting implementations have little resemblance to their silicon NMOS counterparts [21].

Chapter 4

GaAs MESFET device and SPICE modelling

In this Chapter, the GaAs MESFET device structure, operation, and fabrication are discussed. The incorporation of a suitable model into SPICE is described in detail.

The late fabrication of the cells meant that they were not fully tested in time. Therefore simulations are used in subsequent Chapters to support the design concepts where necessary. The work on the recoding of SPICE described in this Chapter adopted a GaAs MESFET model from Honeywell, which was based on that of Curtice [20], and was carried out independently. Although no new model was proposed in this work, the coding exercise provides the background for future improved models to be included, and the details are described in detail in this Chapter.

To illustrate the accuracy of the incorporated model, simulated results of the DC characteristics are compared to measured data from the Honeywell. A critical appraisal of the limitations of the modified SPICE model will be identified from these comparisons.

4.1. GaAs MESFET device operation

The MESFET device is the dominant device used in GaAs, but not in silicon. In Chapter 2, the reasons for this limitation in GaAs have been discussed and its differences from silicon NMOS in terms of circuit design have also been discussed in Chapter 3. It is useful to look at the small signal operation of the MESFET by first reviewing the other more common FET devices.

The MOSFET device

The n-channel silicon MOSFET (Fig. 2.2) is made up of n-type drain and source regions within a p-type well. It is turned ON if a high enough positive gate voltage is applied such that enough minority carriers (electrons) are attracted towards the region underneath the thin gate oxide to form a conducting channel between the drain and source (inversion of the p-type substrate). The insulated gate means that negligible current flows from the gate to the channel even when the MOSFET is

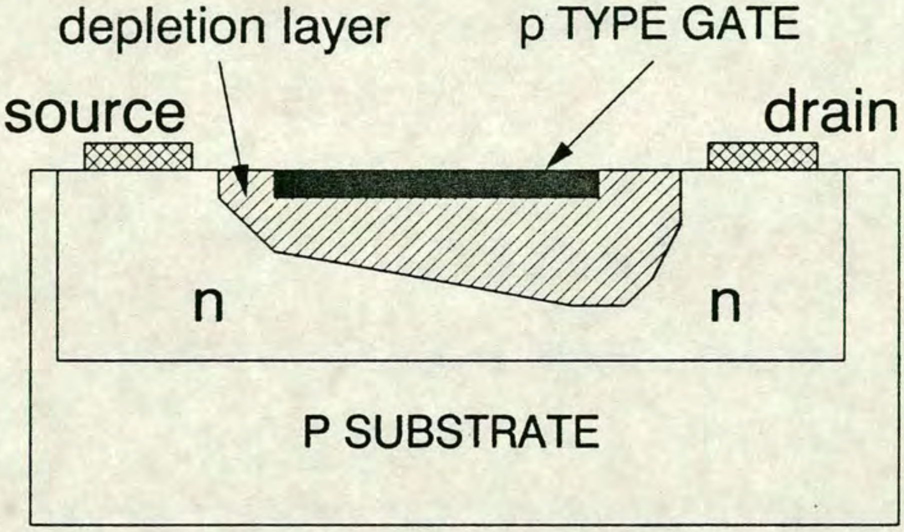


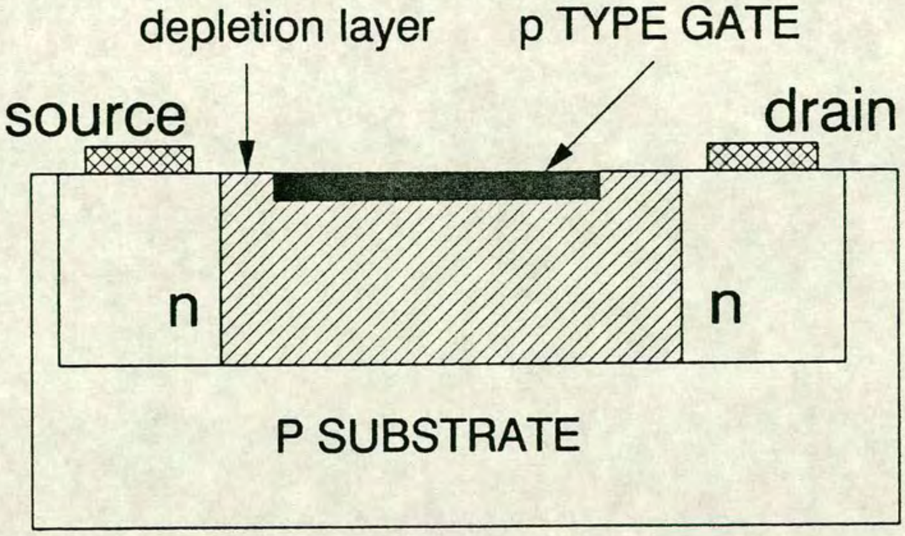
Fig. 4.1 The JFET structure

ON. The gate voltage of a MOSFET can in theory be increased to any value less than the dielectric breakdown voltage of the oxide (typically of the order of +30V). Both n and p channel devices are widely used because of the similar hole and electron mobilities.

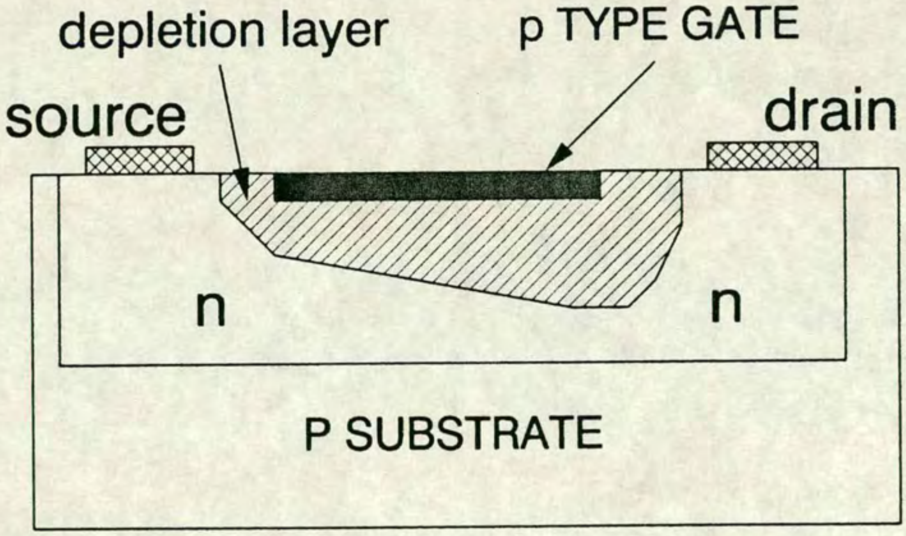
The JFET structure

The JFET device is closest to the MESFET in operation, and is common in analogue silicon circuits such as op-amps input stages [4]. This is due to its superior input noise and slew rate performances compared to MOSFET devices — a direct result of the non-insulated, low capacitance pn junction gate. Fig. 4.1 shows the basic JFET structure. Whereas the MOSFET has a p-type substrate between the drain and source, the JFET has a thin n-type layer between and including the drain and source in an n-channel device.

In the JFET, the gate is a p-type semiconductor in contact with the n-type active area underneath, thus forming a depletion layer. The JFET operates by using its gate voltage to control the thickness of this depletion layer. Taking the depletion-mode or normally-ON JFET for this discussion, if the gate voltage is made negative enough such that the depletion layer extends into the normally conducting n-type



(a) Cutoff region



(b) Ohmic region

Fig. 4.2 (a,b) The JFET in cutoff and ohmic regions.

channel and reaches the bottom of the channel and virtually across the whole channel (Fig. 4.2(a)), then the device is said to be "cut-off". In this state, majority carrier current ceases to flow, and a small leakage current flows from drain to source.

If the gate voltage is increased beyond a certain threshold voltage V_t, there is still a depletion layer which does, to some extent, restrict current flow from drain to source (Fig. 4.2(b)). In the ohmic or linear region, the restriction of current flow changes linearly with gate voltage, thus forming a voltage controlled resistor.

If the gate-source voltage is made increasingly positive, the depletion region shrinks further and starts to reduce in width from the source end of the channel. At this point the gate-source pn junction is forward biased (Fig. 4.3), and gate-source current increases significantly as the depletion recedes until breakdown occurs.

Current saturation in the channel occurs if the drain-source voltage is sufficiently large to cause a constriction at the drain end of the channel owing to the reverse gate-drain bias, such that velocity saturation occurs owing to the high electric field within that constriction (Fig. 4.4). This constriction region increases in width with the drain voltage, thus forming the constant low conductance saturation region of the I-V curve. The value of this conductance (G_{ds}) is particularly important in the design of current sources, and can vary from a few ohms to few tens of ohms, depending on the process.

So far, no distinction is made between the silicon and the GaAs JFET. At this point, it is important to state a major difference between their operations in the saturation region. The difference arises from the fact that velocity saturation occurs at a much lower electric field in GaAs than in silicon (Chapter 2). In silicon, current saturation starts when the depletion region near the drain end of the device literally pinches off the channel by extending to the bottom of the channel, restricting further increases in current, but **without** velocity saturation of the electrons. Therefore in silicon, the voltage conditions when pinch-off occurs (the pinch-off voltage or region) is effectively coincident with the saturation region for all gate voltages, and the two terms can usually be used interchangeably. This is not the case with GaAs, where current saturation occurs at the onset of electron velocity saturation. Pinch-off only occurs at a much lower gate voltage (i.e. higher reverse bias at the drain end of the device), when the device is close to cut-off. The two regions are only coincident for a small gate voltage range, and therefore the pinch-off region is not important in subsequent discussions, and a GaAs JFET is considered to have three regions of operation: cut-off, ohmic, and saturation.

It is clear from this discussion that the JFET has the following properties that the MOSFET does not possess:

(i) The input gate voltage swing can be limited by the pn junction.

(ii) Significant current flows through the gate if the junction is forward biased.

The GaAs MESFET structure

The GaAs MESFET is essentially the JFET with a metal-semiconductor junction, i.e. a Schottky barrier, in place of the pn junction. The gate metal is often a refractory metal of Ti and W compounds, which can withstand the high processing temperatures. Fig. 4.5 shows the cross-section of the basic GaAs MESFET. MESFETs are usually fabricated by selective ion implantation. The n-channel sits directly on a semi-insulating GaAs substrate, with ohmic metals in contact with the n+ drain and source regions. Buffer layers are sometimes used between channel and substrate. Apart from the mechanism involved in saturation discussed above, its switching properties are essentially the same as the JFET discussed above. The main differences are quantitative, and are a direct result of the Schottky barrier. The fabrication of the MESFET is discussed in a later section.

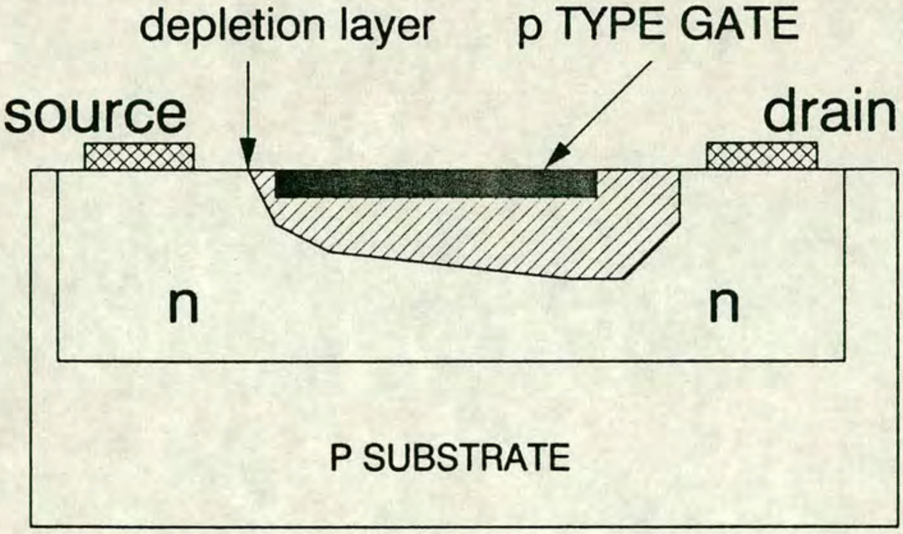


Fig. 4.3 JFET in forward biased mode.

One of the main quantitative effects of a Schottky barrier in the MESFET is the diode "ideality factor". The exponent in the standard diode forward current equation is scaled down by an "ideality factor" ($N_i > 1$), which is normally equal to 1 in pn junction diodes. This means that the rate of increase of forward current with respect to voltage is **reduced** for the same saturation current. On the other hand, because of the higher carrier density of metal, the reverse saturation current is larger, compensating for the larger N_i. The diode turn on voltage (V_{on}) is also lower due to the lower barrier height, which depends on the surface properties of the semiconductor and not on the metal work function [133]. Overall, a Schottky diode turns ON faster, and has comparable forward current to a pn diode at a lower gate voltage, but unfortunately also clamps the gate voltage to a lower value. The low V_{on} of the MESFET compared to the JFET reduces the voltage swing in DCFL even further. This prompted efforts to use JFETs in place of MESFETs in some circuits (Chapter 2), and to use molecular beam epitaxy (MBE) to make MESFETs that have a higher turn on voltage, but without sacrificing switching speed [87].

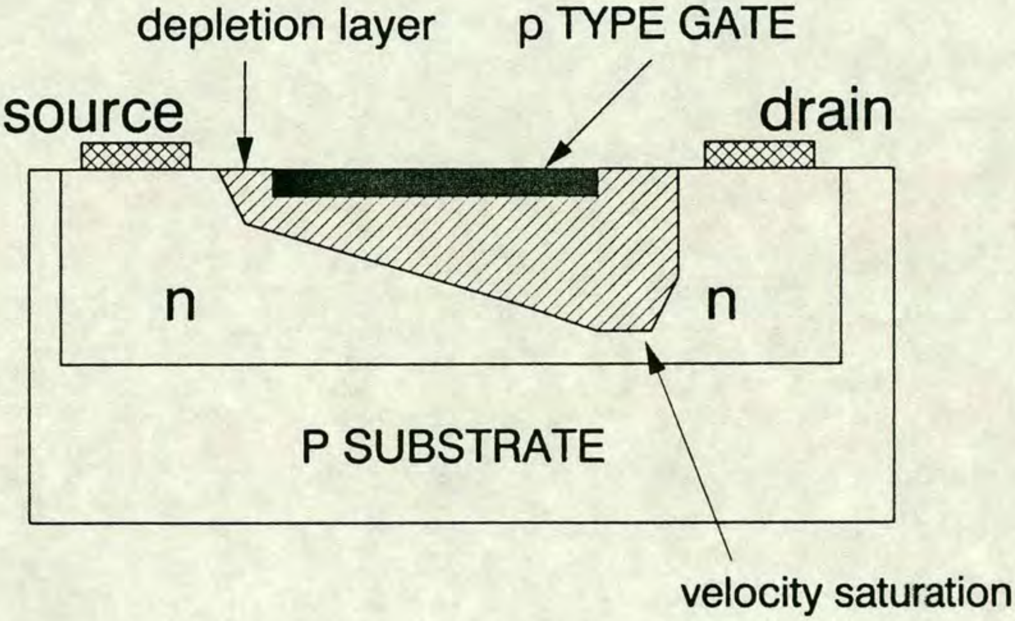


Fig. 4.4 JFET in saturation.

Another important issue associated with the Schottky gate is the gate current. In Chapter 3, the quasi-linear forward characteristic (I_{gs} vs. V_{gs}) was mentioned. This is due to the finite gate-source/drain and drain/source resistances. In logic and switching circuits, the effect of drain-source voltage on forward biased gate current is also important. This directly affects fanout, power, and in the case of pass transistors, clock/signal coupling and breakthrough. At this point, it is useful to introduce the small signal equivalent circuit of the MESFET, based on Curtice [20], as shown in Fig. 4.6.

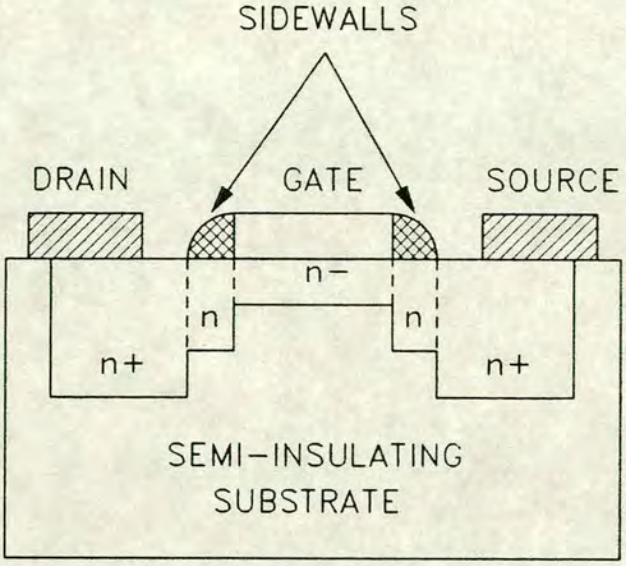


Fig. 4.5 GaAs MESFET cross-section

Consider the case when the gate-source junction is forward biased at 0.7V (V_{GS}). The effective voltage across the ideal gate-source Schottky diode is modulated by the voltage drops across R_G and R_S in Fig. 4.6. If V_{DS} is small, then I_{ds} is also small and the MESFET is in the ohmic region. Most of V_{GS} drops across R_G and the diode and the gate current is large. A maximum is reached if V_{DS} is 0V, as in the case of a MESFET used as a level shifting diode. For a 10μm E-MESFET, relatively small by DCFL standards, the maximum gate current is of the order of 0.1mA. To put this figure into perspective, the same device at a V_{DS} of 0.1V, typical of the logic LOW voltage in DCFL, produces a drain current of about 0.25mA. Since V_{DS} does not reach 0V in practice, the gate current is about half of the maximum of 0.1mA (at $V_{DS} = 0$). Together with gate current division when it is driven by a loaded logic gate, this is not as severe as it appears in DCFL. If the device is used as a pass transistor in the forward biased mode, the situation is much worse. A clock driver by design needs to supply enough current to drive the designed number of transistors and therefore will result in a large gate current mixing with the signal being passed, i.e. clock breakthrough. This is potentially disastrous, for example in a memory cell, and care is needed to prevent this happening. Chapter 5 will discuss in detail how pass transistors are used in this work.

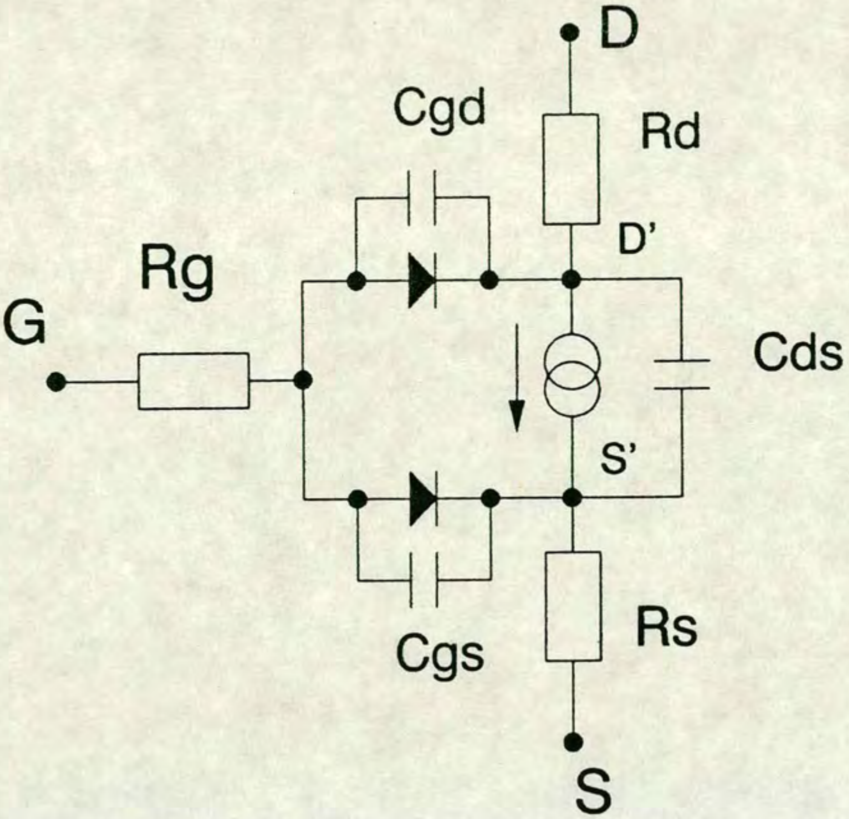
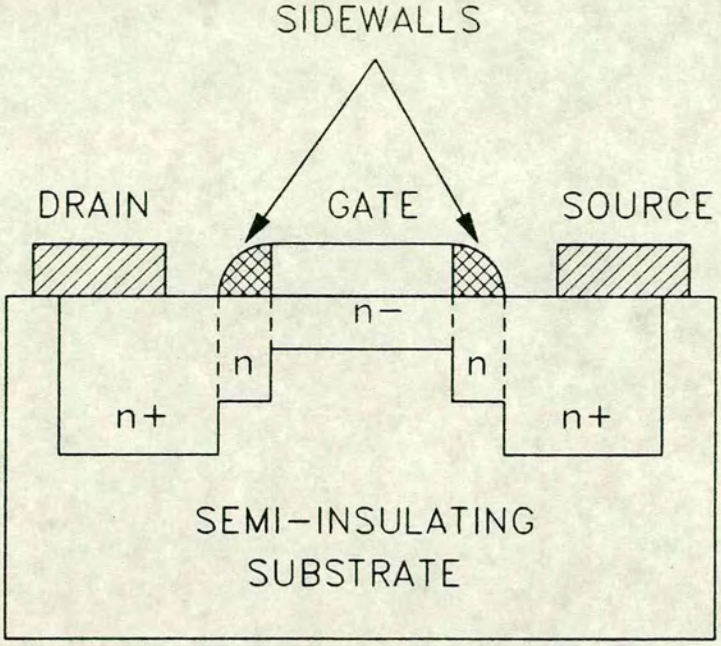


Fig. 4.6 Small signal equivalent circuit of the GaAs MESFET.

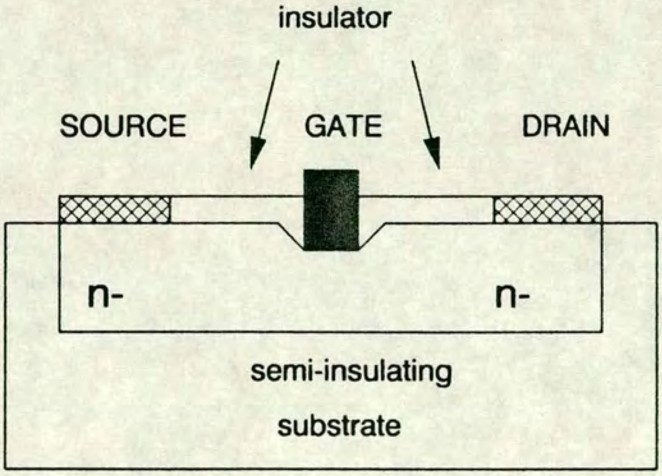
4.2. GaAs MESFET processing

In this section, related aspects of the fabrication process, and a number of important effects and deficiencies unique to the MESFET, and their processing solutions, are discussed. Although the equipments used in GaAs IC fabrication are essentially the same as those in silicon, the GaAs MESFET has a different set of electrical problems. These include effects such as sidegating/backgating and hysteresis, and yield degradation factors. References are also made to two different processes that this work was involved with.

GaAs MESFET's are commonly made by selective ion implantation to provide the accurate control of electrical properties required by VLSI circuits. The two most common device structures adopted in GaAs are self-aligned gate and recessed gate MESFET's [10,69,86,120,149]. Fig. 4.7 shows their two cross-sections. To distinguish them in this discussion, these two processes will be denoted by RG and SAG respectively.



(a) Self-aligned gate GaAs MESFET



(b) Recessed gate MESFET

Fig. 4.7 Two most common GaAs MESFET structures

Table 4.1 below gives a brief summary of the typical processing steps and the design layers involved in SAG [87] and RG [69,120,149] processes.

The process step which distinguishes SAGFET and non-SAGFET processes is the timing of the formation of the transistor source and drain regions. In RG, the E-MESFET gate metal layer, together with the active area mask, are used to define the area above the channel where the device surface is to be etched. Such etching reduces the channel thickness so that it is cut-off for zero gate-source voltage. The gate of a D-MESFET is deposited without etching so that the transistor threshold voltage is negative.

In some other GaAs processes the gates of E-MESFETs and D-MESFETs are both recessed, but by different amounts. This was found to increase the process gain factor (K) [120]. However, when this structure is used, the gate/drain and gate/source distances are less controllable when compared to the SAGFET and unpredictable asymmetry can result, as has been demonstrated by measured results [149]. This effect can be detrimental, to a limited extent, to the performance of circuits which use the MESFET as a bi-directional device, such as the pass transistors in the GaAs cells described in Chapter 5.

The problem of device asymmetry can be largely eliminated by the use of the SAGFET structure. In the SAG process, the source and drain regions are formed after the gate (or a dummy gate) has been laid down. SiO_2 sidewalls are introduced to increase the gate-drain and gate-source distances, in order to improve breakdown characteristics and to minimise gate overlap capacitances.

Other features of the SAG process designed for GaAs VLSI include planar contacts (using a via plug), p-type buried layer, and refractory gate metal (tungsten silicide). The via plug structure used in the SAG process (Table 4.1) is particularly important. The via is made by filling the contact window between metal 1 and 2 with a metal "plug" before the metal 2 proper is deposited, thus providing a fully planar structure, allowing better compaction of the circuits, and improving yield [134].

	RG		SAG	
Step	Design Layer	Process	Design Layer	Process
(i)	(1)	Active layer implant.	(1)	p-buried layer implant.
(ii)		Annealing.	(1)	E-MESFET channel implant.
(iii)	(2)	Ohmic contact.	(2)	D-MESFET channel implant.
(iv)		Isolation implant.		Annealing.
(v)	(3)	D-MESFET gate/metal 1.	(3/1)	E/D Schottky gate metal.
(vi)		Gate recess etch.		SiO_2 sidewall.
(vii)	(4)	E-MESFET gate/metal 1.		N+ source/drain implant.
(viii)	(5)	Insulator.		Rapid annealing.
(ix)	(6)	Contact window.	(4)	Ohmic contacts.
(x)	(7)	Metal 2.	(3)	Metal 1.
(xi)		Insulator.		Insulator.
(xii)	(8)	Bond pad windows.	(5)	Contact windows/via metal.
(xiii)			(6)	Metal 2.
(xiv)				Insulator.
(xv)			(7)	Bond-pads.

Table 4.1 Comparison of selective and non-selective
ion-implant processes (SAG and RG).

Advanced features, that are being incorporated into the Honeywell SAG [87] process, include polyimide dielectric and MBE grown active layers. They are significantly more expensive than Si_3N_4 and ion implanted processes. MBE devices are used when high β MESFETs are required, but with exactly the same design rules as the standard SAG process. A normal implanted MESFET has a typical transconductance (g_m) of 250 $mSmm^{-1}$. The latter technique can produce MESFET's with g_m as high as 400 $mSmm^{-1}$.

4.3. Electrical properties of the GaAs MESFET device

Each process level, from active layer implantation to passivation, closely affects the overall circuit performance, yield, and cost. This section looks at the major electrical properties of and problems in GaAs that affect circuit designs. Attempts to overcome these "imperfections" of the GaAs MESFET operation by additional processing steps are also discussed.

Firstly, it is useful to compare two different processes and identify the critical electrical parameters involved. A summary of the main device parameters for 20μm/1μm (gate width/length) E-MESFETs and D-MESFETs of two typical RG and SAG processes is given in Table 4.2 below.

Referring to Table 4.2, the most critical difference between the two processes is in the D-MESFET threshold voltage, resulting in much lower inverter ratioing for direct-coupled FET logic (DCFL) gates and hence smaller logic gates for the SAG process. Also, SAG has higher α and β, implying a faster MESFET overall than that of RG, even though it has slightly higher gate-source and gate-drain capacitances and λ value. The precise quantitative meaning of these device parameters are discussed in the modeling section later on. Table 4.3 below compares the typical propagation delays of DCFL inverters fabricated with the SAG and RG processes for two fanout conditions.

	RG		SAG	
Parameter	E-MESFET	D-MESFET	E/D-MESFET	Unit
V_t	0.15	-1.08	0.2/-0.5	V
α	3.3	2.4	4.5	V^{-1}
β	1.46	0.82	4.6/2.46	mAV^{-2}
λ	0.09	0.03	0.172	V^{-1}
I_f	1500	980	20	fA
N_i	1.42	1.46	1.21	-
R_s	145	90	65.9/42.5	Ω
R_d	145	90	65.9/42.5	Ω
C_{gs}	7.8	7.8	10	fF
C_{ds}	7.8	7.8	10	fF

Table 4.2 Device parameters of two GaAs processes

Fan-out	RG	SAG	Units
1	113	80	ps
2	191	110	ps

Table 4.3 Inverter t_{pd} for two GaAs processes

From Table 4.3, the speed advantage of SAG over RG and the large effects of fanout on gate delay in both cases are evident. Although this simple comparison shows that the self-aligned gate process is superior, the result may not be indicative of all such processes, and is still a matter for debate. Nevertheless, it would appear that GaAs should follow the same path as silicon, where MOSFET processes have evolved into using exclusively SAGFET structures, as process control improves.

Process control and noise margin

The effects of deviations in device characteristics on noise margin and functional yield, within a chip and over a wafer, are clear. Of particular importance to digital circuits is the standard deviation (σ) of the threshold voltages (V_t), because of its direct effects on noise margins in logic gates.

Noise margin has been a major concern in using DCFL, compared to other circuit configurations in GaAs (Chapter 2), owing to its smaller logic swing. Excessive process variations of device parameters over a large chip area can reduce the noise margins of some logic gates to zero, resulting in very low functional yield. In GaAs, standard deviations of 50mV to 100mV in V_t were not uncommon until recently, meaning that fluctuations in V_t of 100mV to 200mV were possible [24] (assuming Gaussian distribution). Noting that the V_t of an E-MESFET device is typically +200mV and the logic HIGH in DCFL gates is +700mV, these deviations are potentially disastrous. This situation has been greatly improved in the SAG process, and other recent SAGFET and RGFET processes [120,149]. Table 4.4 below lists the typical σ of V_t for the SAG and RG processes.

Process	SAG	RG
E-MESFET	25 mV	60 mV
D-MESFET	38 mV	120 mV

Table 4.4 σ values of V_t of two MESFET processes

Looking at the SAG process more closely, close range V_t deviation (within a 5×5 mm^2 area) could be as good as 6 mV for the E-MESFET [87]. Also, Monte Carlo simulations carried out at Honeywell have shown that chips with 10^5 to 10^6 transistors could have a yield of 90%, depending on the actual circuit. These simulations have taken into account process variations in V_t, β, and Schottky diode forward turn on voltage. These figures imply that noise margins should not be the major yield limiting factor in GaAs VLSI, and had been corroborated by test results [87]. However, if non-DCFL or unconventional circuit structures are used, as in the GaAs cells in this work, noise margins must be considered carefully before they are applied in the large scale (Chapter 5). Other yield hazards exist, and are discussed next.

Yield hazards in processing

The Monte Carlo figures mentioned above did not take into account two processing problems. These are metallisation/dielectric stress (including metal 2 and passivation stress) and electrostatic damage (ESD) [66,87]. ESD was a particularly surprising cause of chip failure and was identified only recently, as it had been considered a problem only associated with silicon, especially MOSFET. This can be solved by input protection, and proper grounding precautions during processing at increased costs. The stress problem is more complex and attempts to solve it include the use of alternative dielectrics. Examination of available yield figures reveal that it is likely to remain one of the major yield hazards. For instance, yield figures of 50% to 60% for MSI chips based on BFL (Chapter 2) could be attributed more to metallisation than to noise margins, especially as BFL has a much higher noise margin than DCFL (Chapter 3). In terms of design and layout, large number of long interconnects on the same chip are best avoided as far as possible.

Other long standing problems include the brittleness of and high defect densities in GaAs wafers, and the lack of large diameter wafers. These do not directly affect circuit designs and their simulations, but clearly increase costs in manufacturing. A recent solution, that has yet to see wide spread use, is to use GaAs-on-Si wafers [65,126,154]. These studies showed that circuits built on GaAs-on-Si have similar or better yield than GaAs wafers, provide large diameter wafers at a low cost, improved thermal conductivity and reliability, and provide the prospect for integrating both GaAs and Si circuits, as well optical devices on the same chip. To offset these advantages, circuits are generally slower than GaAs-on-GaAs circuits, which may be unacceptable for most manufacturers, especially military related ones, and prevent them from being used on a large scale.

4.3.1. Detrimental device properties of the GaAs MESFET

Apart from process control and noise margins, several other problems exist that, to different extents, affect circuit design and modeling. These are:

(i) Channel widening effect.

(ii) Hysteresis.

(iii) Backgating/sidegating effect.

(iv) Short channel effects.

Channel-widening effect

This effect is the increase in output conductance of a MESFET under dynamic conditions, compared to that of the dc value. This is illustrated qualitatively in Fig. 4.8. The effect is believed to be caused by electron injection from the channel into the substrate. The higher drain current under dynamic conditions, compared to the normal DC value, results in an apparent increase in device width. In terms of device parameters, this is equivalent to an increase in λ (Table 4.2). The exact mechanism is beyond the scope of this discussion. More details can be found in publications [11,43,128].

The decrease in output impedance of the device at high frequencies (frequency dispersion) clearly complicates design and simulation. Output buffers driving low impedance loads require more elaborate techniques to overcome this decrease. In terms of simulation, different figures of λ for DC and AC analysis can be used to account for this effect. More important is other effects caused by or related to this electron injection, including hysteresis and backgating discussed below.

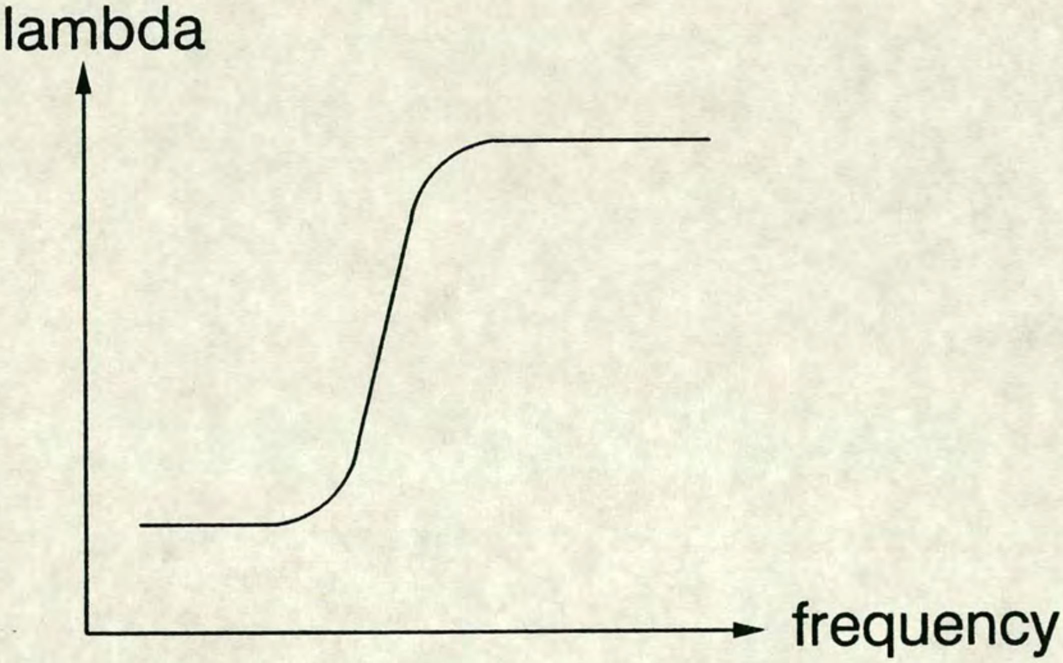


Fig. 4.8 The channel widening effect (frequency dispersion).

Hysteresis in I-V characteristics

Often simply called hysteresis, it can be identified as low frequency changes in the I-V characteristic of a device under dynamic conditions. Fig. 4.9 illustrates this hysteresis effect qualitatively. This effect is additional to the channel-widening effect described above. As V_{ds} is increased, the I-V curve of the device follows that of the arrow in the Figure. It should be noted that this phenomenon has no relationship to the body effect in silicon [83]. Although the exact cause of hysteresis is still a matter of debate among device scientists, it is clear that it is different from the body effect.

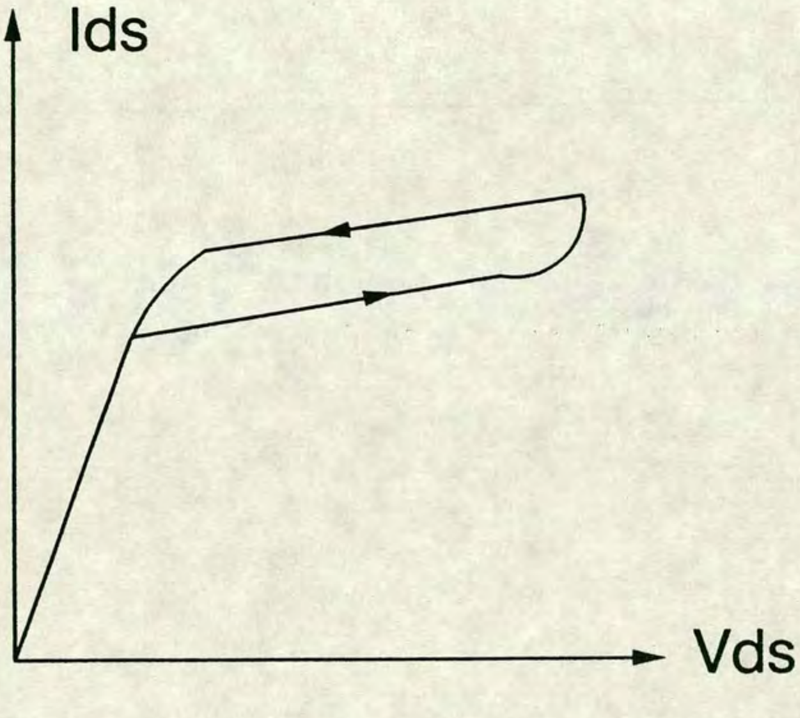


Fig. 4.9 Hysteresis of drain current in GaAs MESFET

The exact cause of hysteresis is still a matter for debate among device physicists. It has previously been attributed to surface interface properties and surface states [128]. Recent attempts in explaining and reducing it showed very close links with the backgating/sidegating effect discussed later on. The latter has been partly attributed to electron injection from the channel into the substrate (which also causes the channel-widening effect). Electrically active deep-level traps (DLT's) in the substrate are charged by this current injection [33,72,128].

Although the cause of hysteresis may possible be the same as the other effects in the GaAs MESFET, it had been found that it has two unique properties:

(i) The effect is significant only when the device is deep into saturation (typically $V_{ds} > 2\times(V_{gs} - V_t)$.

(ii) The time constant of the fluctuations depends on the carrier density in the channel close to the substrate, and the DLT density and occupancy in the substrate. This can range from milliseconds to minutes.

The immediate effect of (i) is a slow change in the small signal output conductance of a device in deep saturation. In the context of GaAs DCFL, again using the inverter as an example, the D-MESFET pull-up device is always saturated and can be affected by this effect. However, the channel-widening effect is more severe due to its high frequency dispersion effects. Also, hysteresis can be effectively avoided by choosing a low enough V_{dd} supply voltage so that the D-MESFET load is not deep into saturation so that it does not occur. Other factors also determine the value of V_{dd} and have already been discussed in Chapter 3.

On the other hand, hysteresis can affect analogue circuits more than switching circuits. These include I/O buffers, op-amps, and analogue-to-digital converters. Any current and voltage references within a circuit have to take into account low frequency drifts and offsets, as well as complicated thermal considerations [43]. Output buffers (for 50Ω lines) and level converters (such as GaAs to ECL voltage levels), tend to be rather complicated, consume a lot of power, and introduce large propagation delays [135], when they are required to operate over wide temperature ranges and at high speeds.

To reduce hysteresis, various additional processing methods have been successful to different extents. In the Honeywell SAG process, a p-type buried layer, often called a "buffer" layer, automatically generated in the pattern generation (PG) phase, serves to suppress the injection of electrons from the channel into the substrate [134]. The change in output conductance from hysteresis was found to be reduced by at least a factor of 3. In DCFL logic circuits which use a carefully chosen V_{dd} supply voltage, hysteresis is effectively suppressed.

Backgating/sidegating effect

This is commonly identified as the switching-off effects on a MESFET by an adjacent Schottky gate. This greatly increases the problem of device isolation/compaction in GaAs VLSI, and offsets the advantage of having a semi-insulating substrate. The effect has the property of occurring only when the sidegate voltage has fallen below a certain process dependent value (typically -4V). This is particularly detrimental to buffered-FET logic (BFL), and to a smaller extent, to certains circuits of the GaAs cells in this work, owing to the widespread use of negative voltages (Chapter 5).

Sidegating is believed to be caused by a combination of current injection from channel to substrate and avalanche breakdown of surface states in the channel [128]. The result is the modulation of depletion layer depth near the channel substrate interface such that a nearby channel can be turned OFF by a sidegate at or below a certain negative potential.

Owing to its relationship to hysteresis and the channel-widening effect, the p-buried layer had been found to be also very effective in reducing it [134]. This was confirmed by other similar methods [57,15]. The isolation from the sidegate is increased if the buried layer is connected to the most negative potential in the system [57].

In the Honeywell process, a 10μm separation between an unrelated negative gate voltage below -4V and an active area is still necessary. Although a 10μm separation is still a large dimension compared to other design rules, this is not a major problem in this work because very few gate voltages are at or near the sidegate "threshold".

Short channel effects

Short channel effects occur as the channel length is reduced to 1μm or less. They include poor cut-off characteristics, a large increase in diode ideality factor, and threshold voltage variations at short (submicron) gate lengths [134]. A direct result of these effects is the destruction of the expected proportional increase in β value with shorter gate length. The p buried layer in SAG again has been found to substantially reduce these effects [57]. It was also found to improve K value, V_t uniformity, and soft error immunity, the latter by reducing the collected charges induced by alpha particle radiation using the high p-barrier [82].

4.4. SPICE modelling of the MESFET model

This section describes the details of the MESFET model and its coding into the standard SPICE2G.6. The original SPICE2G.6 has a JFET model which has been known to have a number of inaccuracies if applied to the GaAs MESFET [20]. These can be summarised as:

(i) In the DC characteristic of I_{ds} versus V_{ds}, poor agreement between the JFET model at the transition region between ohmic and saturation regions and the actual measured MESFET curves.

(ii) The different λ at low and high frequencies of the MESFET, a result of the channel-widening effect, is not modelled.

(iii) The gate-source and gate-drain diode I-V characteristic over-estimates the diode forward current for gate voltages above 0.5V but correctly estimates the current below 0.5V if the correct ideality factor is used. If an ideality factor of unity is used, then the situation is reversed.

The problem of λ in (ii) is not severe depending on the process being used. Unfortunately, the other two problems cannot be solved without choosing a curve fitted set of parameters which do not bear physical resemblance to those of the real device.

Fig. 4.10 shows the small signal equivalent circuit of the JFET model. The I_{ds}—V_{ds} characteristic is comprised of three separate regions: cutoff, ohmic, and saturation. The corresponding forward equations for the drain current source are as follows:

(i) Cutoff region:

$I = 0$

(ii) Ohmic region:

$$I_{ds} = \beta(\frac{W}{L})(2(V_{gs}-V_t)-V_{ds})(1+\lambda V_{ds})V_{ds} \tag{4.1}$$

(iii) Saturation region:

$$I_{ds} = \beta(\frac{W}{L})(V_{gs}-V_t)^2(1+\lambda V_{ds}) \tag{4.2}$$

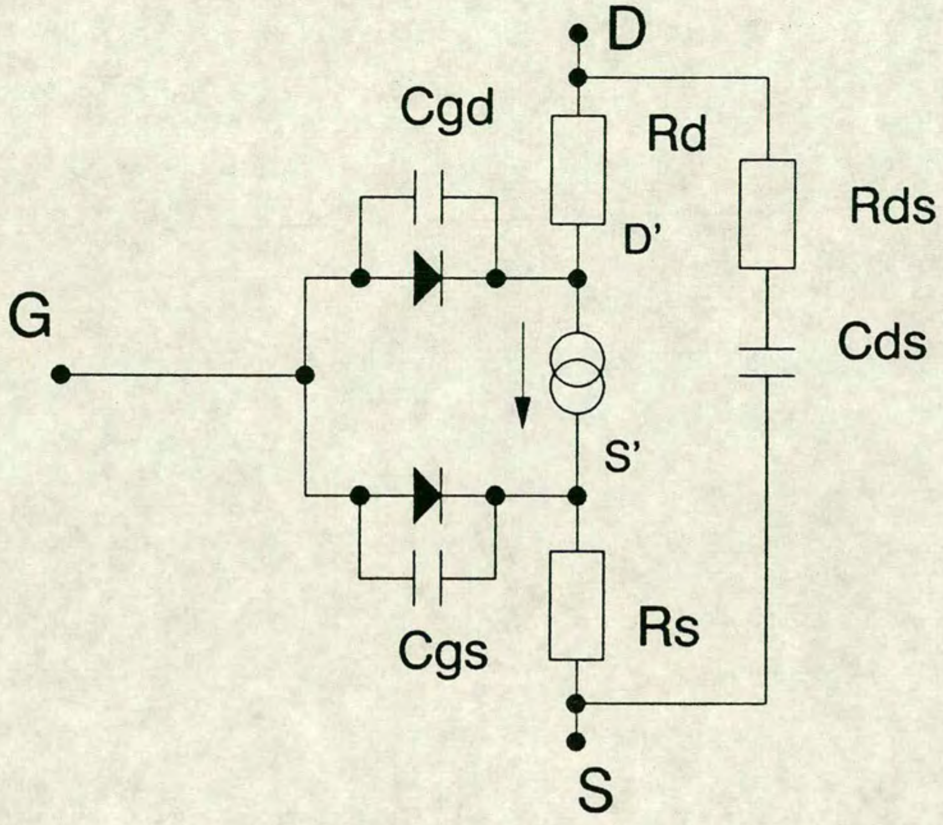


Fig. 4.10 Small signal equivalent circuit of GaAs JFET

4.4.1. The Curtice MESFET Model

The Curtice MESFET model [20] is a relatively simple analytical model well suited to time-domain analysis. The parameters can be evaluated either from experimental measurements or from accurate detailed models. Variations based on this model largely follow the lines of the hyperbolic tangent function correction factor described below, and have been widely accepted in industry for general use [41,42], including at Honeywell. Its suitability for MODFET modelling is as yet not entirely proven and therefore the SPICE modifications will be limited to MESFET only.

Compared to the JFET equations, the Curtice equations are comprised of two regions only: cutoff and active regions. The corresponding forward response equations are:

(i) Cutoff region:

$I_{ds}=0$

(ii) Active region:

$$I_{ds} = \beta(\frac{W}{L})(V_{gs}-V_t)^2(1+\lambda V_{ds})\tanh(\alpha V_{ds}) \tag{4.3}$$

With these equations, the model effectively combines the previously separate ohmic and saturation regions in the JFET into one region using the hyperbolic tangent function, and solves the errors in the "knee" region of the transfer characteristic mentioned earlier. It does not solve the problem of a variable λ and therefore a RC network solution, or a different λ for transient analysis should still be necessary if the process requires it. In the case of the Honeywell process, this was found to be unnecessary [87].

One inaccuracy of this model is the zero I_{ds} in the cutoff region. Here a simple solution is to add a parallel resistor across the drain-source terminals. The Honeywell value is 50 kΩ.

The diode forward current equation that can be used with this model can be represented by this equation:

$$I_g = I_f(e^{\frac{qV}{nkT}} - 1) \tag{4.4}$$

where

I_f = diode reverse saturation current

q = electron charge

V_f = diode forward voltage

n = diode ideality factor

k = Boltzmann constant

T = temperature in Kelvin

Finally, before going onto the actual SPICE modifications, a more general form of Curtice's MESFET I_{ds} equation (Eq. 4.5) has seen widespread use simply because of its flexibility in matching specific process requirements or abnormalities [42]. This is achieved by removing the restriction on the exponents in the current equation as follows:

$$I_{ds} = \beta(\frac{W}{L})[(V_{gs}-V_t)^N + \lambda(V_{gs}-V_t)^M V_{ds}]\tanh(\alpha V_{ds}) \tag{4.5}$$

M and N are exponent factors that can be extracted or calculated from curve tracer measurements. If M and N are taken to be 2 in this equation, then it simplifies to the Curtice equation (Eq. 4.3). This is the form of the current equation adopted for coding into SPICE2G.6.

4.4.2. SPICE modifications

SPICE is a well established program [95,145] designed for circuit level simulations, with emphasis on IC engineering. Numerous versions of SPICE exist commercially, based on the original versions and revisions developed at Berkeley for non-commercial purposes. The discussions in this section will only involve the Berkeley version SPICE 2G.6.

Most major electronic devices are catered for in SPICE by internally fixed models. The models for semiconductor devices are based on well proven results from device research and allow the user to specify some device parameters, but without the provision for readily changing the actual model. The latter property is one of the weaknesses of SPICE, compared to some other simulation programs like ASTAP [7].

In the 2G.6 version on which this work is based, high voltage and power devices and components are not modelled as thoroughly as semiconductor devices used in ICs. These include device breakdown and large signal characteristics, and transformers. SPICE is also well known for its susceptibility to non-convergence, owing to its integration algorithm. In addition, the option for simulating at different temperatures appears not to be functional — the default temperature of 300K applies to all simulations.

SPICE 2G.6 was written in FORTRAN 77, and is made up of 9 program files. In the UNIX operating system, each file is denoted by a file name with an ".F" extension (e.g. SPICE.F is the main program). Each program performs a specific task with a number of subroutines. The functions of these 9 files are as follows (it is important to note that UNIX distinguishes between upper and lower case letters in file names, but not in program statements):

(i) SPICE.F — main program.

(ii) ACAN.F — AC analysis, including frequency response, noise, and distortion analysis.

(iii) DCOP.F — prints the DC operating points of non-linear circuit elements.

(iv) DCTRAN1.F — controls DC transfer curve analysis, operating points, and transient analysis.

(v) DCTRAN2.F — performs DC and transient analysis for all semiconductor devices

(vi) ERRCHK.F — performs pre-processing and general error checking on SPICE circuit input data.

(vii) OVTPVT.F — performs tabular listings of analysis results and generate plots for lineprinters.

(viii) READIN.F — performs input processing of SPICE, including circuit and element descriptions, and models.

(ix) SETUP.F — sets up the sparse matrix used in SPICE for analysis.

The modifications that were made to SPICE 2G.6 to incorporate the modified-Curtice model MESFET model (Eq. 4.5) can be divided into three main categories:

(a) Lexical changes that affect the input and output of symbols and numbers.

(b) Changes to internal program memory management which affect the transfer of data and constants between the subroutines.

(c) Arithmetic changes that directly affect the numerical computations associated with the incorporated Honeywell (general Curtice) model.

A simplified block diagram of the relationship between the modified subroutines is shown in Fig. 4.11. For ease of maintenance, the modified subroutines are separated from the program files to which they originally belonged. The relevant program file and subroutine modifications are summarised below:

(1) PROGRAM SPICE & SUBROUTINE FIND.

SUBROUTINE FIND was part of SPICE.F. Its function is to check that a circuit element or subcircuit in the SPICE input is legitimate and is not duplicated, and to produce error messages or allocate memory and identification for its storage where appropriate.

This subroutine calls a dynamic memory manager subroutine which expands the data structure associated with each element type, and manages efficient searching and linking. It must be informed of any new parameters associated with any element.

The array "LVAL" stores the maximum number of double precision parameters needed for a particular SPICE element. The actual parameter names and values are stored in different arrays (see later). In the case of the SPICE JFET model, there are 12 user specifiable parameters and 4 derived parameters. Therefore the LVAL(23) value, representing the entry for the JFET, is 16+1 or 17 (with one null entry). Since this value is a constant, the FORTRAN 77 "DATA" statement is used to enter the value into the array. From equations 4.4 & 4.5, there are 5 extra parameters for the modified-Curtice model. Therefore, LVAL(23) was changed to 22 via the "DATA" statement. The same modification was made to LVAL(23) in "PROGRAM

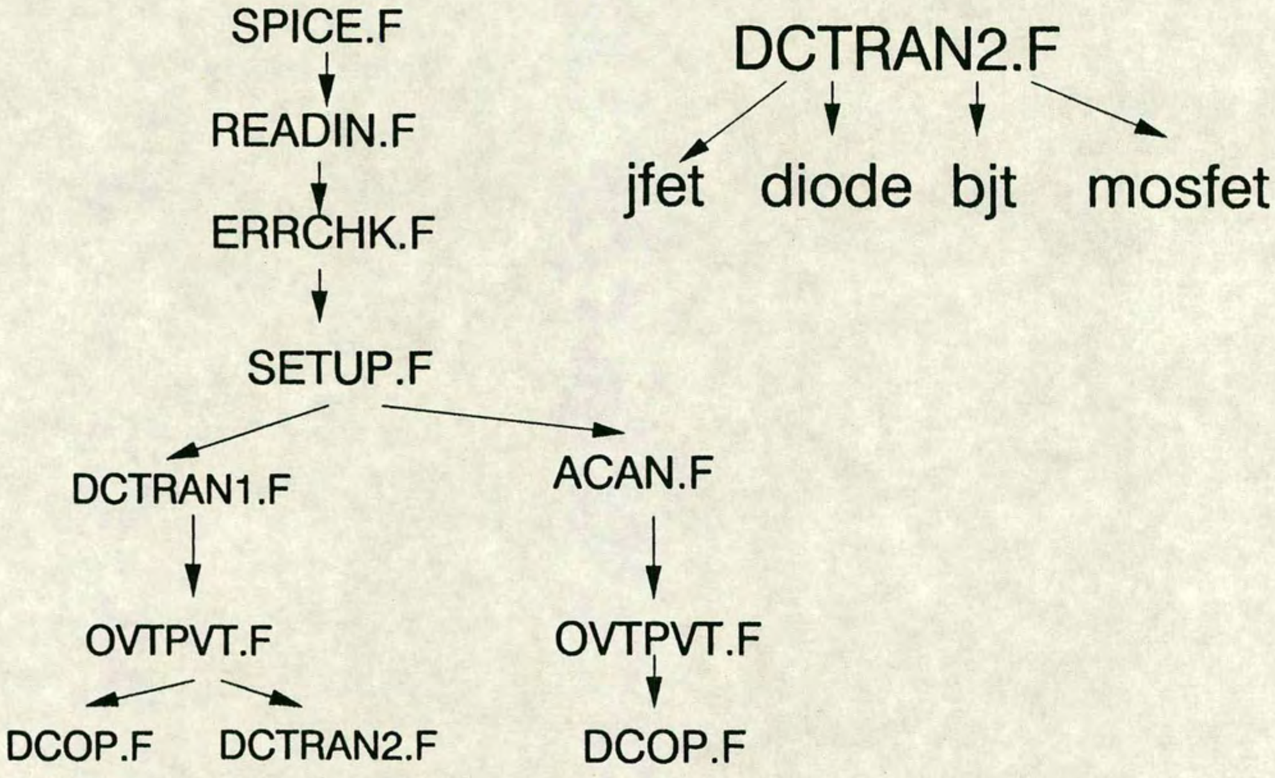


Fig. 4.11 Relationship between SPICE2G.6 main subroutines.

SPICE" (the main program).

(2) SUBROUTINE READIN is the main routine in READIN.F. Its function has already been mentioned above.

An array "AMPAR" stores the reserved names of the model parameters for each of the 4 semiconductor model types, namely DIODE, BJT, JFET, and MOSFET. The number of entries for each model is the same as the "LVAL" entry mentioned above for that model, except in the case of the BJT. The dummy entries for BJT were due to previous changes to the model.

A second array, "IPAR", indicates the starting position of the the parameters in "AMPAR" for each model type. The changes to these two arrays are as follows:

(a) DIMENSION AMPAR(115) changed to DIMENSION AMPAR(120).

(b) Values of IPAR(4) and IPAR(5) are changed to 77 and 119 respectively, in the DATA statement.

(c) Addition of 5 parameter reserved words in the DATA statement for array AMPAR. The 5 new reserved words are:

Parameter	Reserved word
α	ALPHA
n_i	NI
M	M
N	N
C_o	CO

Internally, these same reserved words are used as FORTRAN variable names for double precision computation.

(3) SUBROUTINE MODCHK was part of ERRCHK.F, and performs the task of checking and pre-processing the model parameters before computation. Arrays "AMPAR" and "IPAR" are changed as in SUBROUTINE READIN above. Other changes include arrays "DEFVAL", "IFMT", and "IVCHK". These are described below.

Array "DEFVAL" stores the default values for each of the model parameters. Therefore its size was increased by 5 (DIMENSION statement) and 5 new default values are added to the array.

The value of each entry in array "IFMT" indicates three properties of a SPICE parameter:

(a) If IFMT(I) (the Ith entry in the array) is greater than 2, then the parameter corresponding to AMPAR(I) will always be printed in the model parameter list in the SPICE output file. Otherwise, the value is printed only if it is user-specified in the SPICE input file.

(b) If IFMT(I) is equal to 1 or 3, the associated parameter will be printed out without an exponent.

(c) If IFMT(I) is equal to 2 or 4, the value will be printed with an exponent.

The necessary changes are therefore an increase of the array size by 5, and 5 new entries to indicate how the new modified-Curtice parameters should be printed in the output file.

Array IVCHK indicates if a parameter in the array AMPAR can have negative values or not. Again, the Ith value in this array corresponds to the Ith parameter in AMPAR. If IVCHK(I) is equal to -1, then a negative value is allowed for that parameter. If it is equal to 0, then that parameter has to be positive (or zero). Therefore 5 new entries are added to IVCHK for the new parameters.

Also in ERRCHK.F, the main subroutine ERRCHK requires a change in LVAL(23) from 17 to 22, as in PROGRAM SPICE and SUBROUTINE FIND.

(4) SUBROUTINE ADDELT adds an element in the SPICE input file to the existing circuit definition. Array LVAL needs to be changed as in the other subroutines.

(5) SUBROUTINE JFET performs the transient and dc analysis for the JFET device. It was part of DCTRAN2.F. The changes are crucial to this work and the complete modified subroutine is listed in Appendix 1, with extensive comments. They are summarised below:

(a) The diode equation was changed to include the ideality factor (N_i), as in Eq. 4.4.

(b) The equations for I_{ds}, g_{ds}, and g_m were changed to the modified-Curtice formulae. For the forward mode of operation, the following relationships apply [95]:

$$g_m = \frac{\delta I_{ds}}{\delta V_{gs}}$$

$$g_{ds} = \frac{\delta I_{ds}}{\delta V_{ds}}$$

These two equations can be derived directly from Eq. 4.5 and are given in Appendix 2.

In the inverse mode, g_{ds} is given by [6]:

$$g_{ds} = \frac{\delta I_{ds}}{\delta V_{ds}} - g_m$$

(6) UNIX system file MAKEFILE, is commonly used to manage the compilation/re-compilation of a large number of program files to form the final binary code output file for execution. The modifications to this file is system-specific and is therefore detailed in Appendix 3 for reference purposes.

4.4.3. Comparison with measured results

In this section, comparisons of simulated and measured results for the Honeywell process are presented to illustrate the limitations of the model. The measured and simulated I-V curves for the E-MESFET and D-MESFET are compared in Figs. 4.12 & 4.13 respectively. With this set of measured results, the V_t of the D-MESFET was –0.4V, instead of the nominal -0.5V. It is common in the Honeywell process to "step" the value of V_t on several batches of the wafers [87], by varying the implant depth. This serves to provide an assessment of process control for future wafer runs, as well as a choice of circuits with different speed and power tradeoffs. The V_t of E-MESFETs are usually left unchanged from the nominal value of +0.2V, as in this case.

By inspection of the curves, they showed good agreement with each other, especially at higher V_{gs} values. Table 4.5 below lists the percentage errors of the two MESFETs at a V_{gs} of +0.7V, i.e. under forward bias. The E-MESFET showed

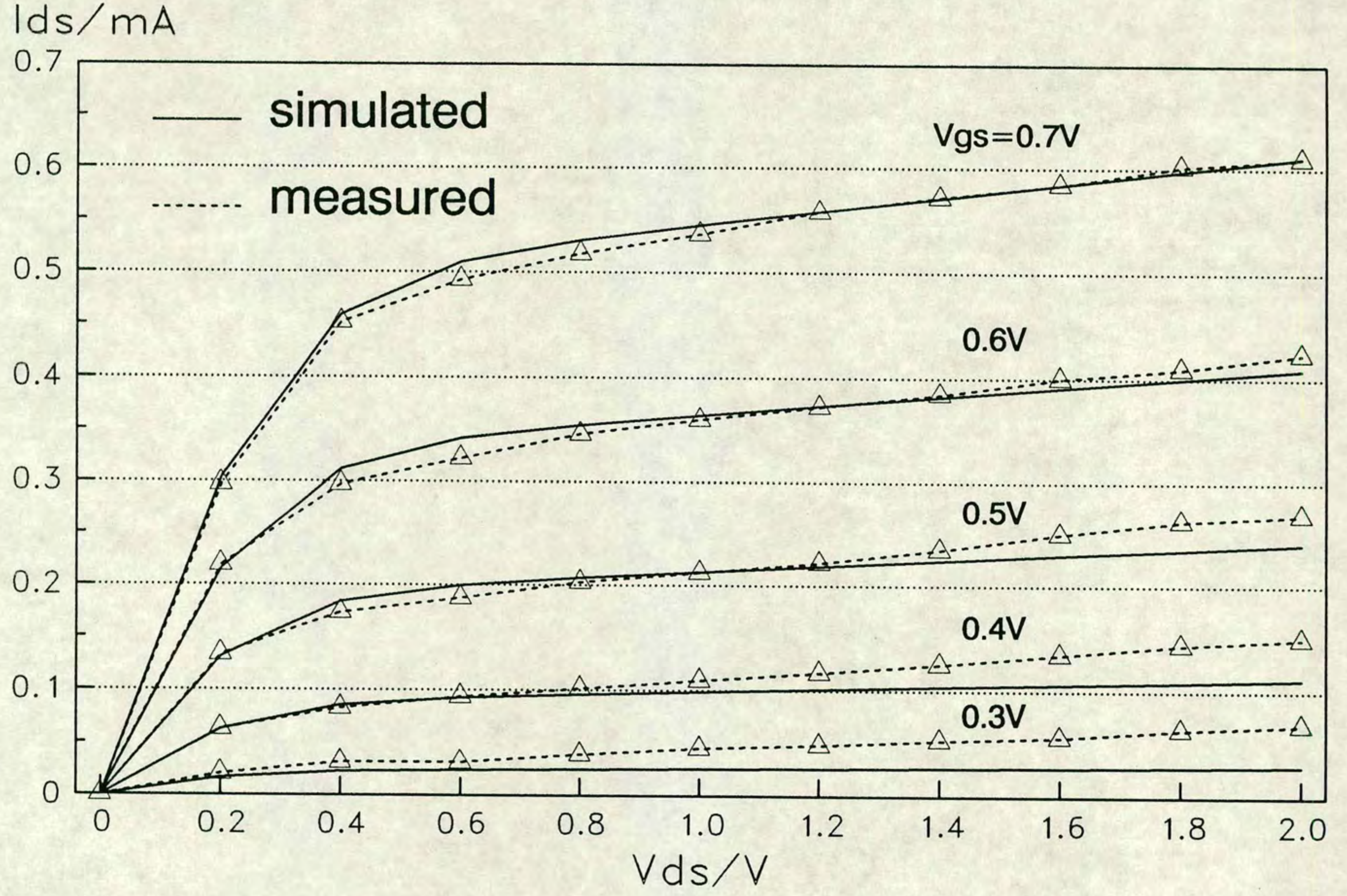


Fig. 4.12 Comparison of simulated and measured E-MESFET I-V characteristics.

excellent agreement (±4%) with the measured result, with a ±4μA error in curve reading. The errors for the D-MESFET were larger than that of the E-MESFET. (The curve error is also larger at ±6μA owing to different scales.) At the DCFL V_{dd} supply voltages of 2V or less, this translates to an error of about 6% or less between the simulated and measured results of the D-MESFET. This is in line with the fact that in the Honeywell self-aligned gate process, the D-MESFET requires two separate implant steps while the E-MESFET actives area requires one. The D-MESFET also has a larger spread in V_t as a result. In addition, the simulated result was obtained by changing only V_t in the modified-Curtice model parameters, to the −0.4V of the measured D-MESFET. All other model parameters are left unchanged. In processing terms, a deliberate change in V_t from the nominal value also changes the value of β, usually larger than that at a more negative V_t.

V_{ds}/V	E-MESFET	D-MESFET
0.2	−3.6%	+9.3%
0.4	+2%	+3.9%
0.8	+0.95%	+2.5%
1.2	+0.55%	+2.4%
1.6	+1.2%	+6.4%
2.0	−7.8%	+5.9%

Table 4.5 I_{ds} percentage errors at V_{gs} of 0.7V for various V_{ds}.

In the context of DCFL circuits, a more critical comparison of the results is at the logic boundaries of DCFL. For the D-MESFET, the results at V_{gs} equal to 0V and V_{ds} less than 2V are most important. A detailed look at the I_{ds} curves at V_{gs}=0V (Fig. 4.13) shows that the errors are less than approximately ±15% between V_{ds} values of 0.8V and 2V (Table 4.6). Since the DCFL logic swing is 0.1V to 0.75V, equivalent to V_{ds} between 0.95V and 1.6V across the D-MESFET in the Honeywell process, this maximum error figure applies. At V_{ds} values of 0.4V and lower, much larger errors were observed. At V_{ds} of 0.4V, a large error of +44% was observed. The error at 0.2V was approximately 33%. These errors are not severe in terms of DC behaviour since they are well beyond the normal operating range of a D-MESFET load in DCFL logic circuits.

V_{ds} (V)	Error in I_{ds} (%)
0.2	+33
0.4	+44
0.8	+15.3
1.6	−9.5
2.0	−13.7

Table 4.6 I_{ds} errors of simulated
D-MESFET at $V_{gs}=0$

In the case of the E-MESFET, its switching nature means that the whole range of V_{gs} is important, while V_{ds} larger than 0.75V can be ignored. By analysing the results, it was found that very good agreement between simulation and measurement was maintained for V_{gs} down to 0.4V, with errors of 6.5% or less, over the crucial V_{ds} range. At 0.3V, the lowest V_{gs} available for comparison, the simulated results underestimated I_{ds} by as much as 100% at a V_{ds} of 1.6V. The error reduces to +30% (overestimates I_{ds}) at lower V_{ds} value. Although a direct comparison of currents at even smaller V_{gs} is not possible, it is safe to assume that the simulation underestimates the leakage current of the E-MESFET when it is close to and within cutoff. Therefore more care needs to be taken in terms of fanin design of the DCFL gates.

Under dynamic conditions, first consider the case when the input to a DCFL inverter changes from logic LOW to HIGH. Here V_{gs} of the E-MESFET rises from approximately +0.1V to +0.75V, and V_{ds} falls from +0.75V to +0.1V. Therefore the whole family of DC curves within these voltage ranges need to be assessed. Referring to Fig. 4.12, the largest errors occur when V_{gs} is below 0.3V, where the simulated I_{ds} current would be significantly **smaller** than that of the real device. In the case of the D-MESFET device, V_{ds} would rise from about 0.95V to 1.6V (assuming a V_{dd} of 1.7V), while V_{gs} is fixed at 0V. Within this range (Fig. 4.13), the simulated curve fits the measured curve very well and only the errors in the E-MESFET need to be considered. Since the simulated **fall** time of the output depends on the pull-down current of the E-MESFET, which is smaller than that of the real device, the fall time would be **larger** (overestimation) in the simulations.

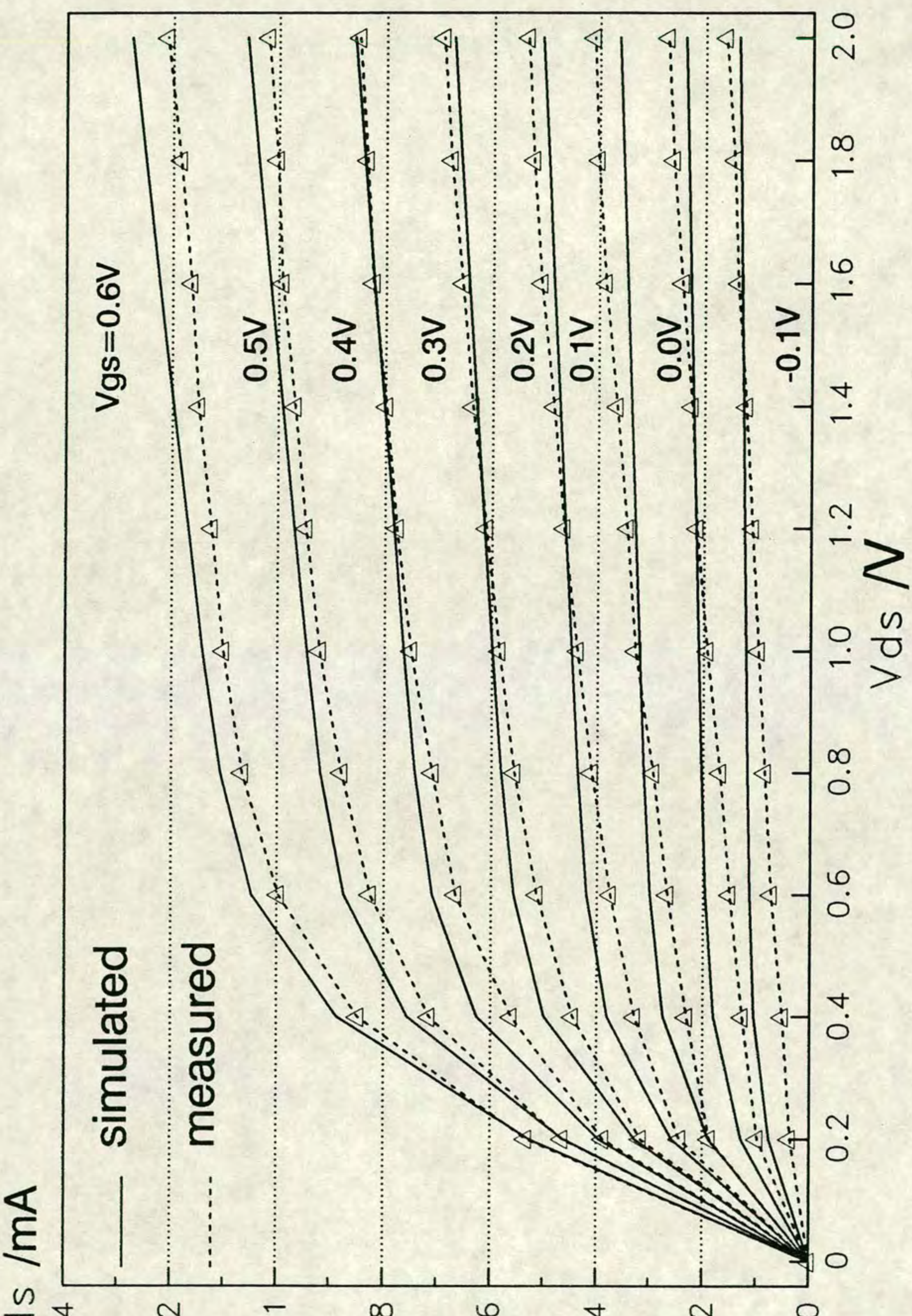


Fig. 4.13 Comparison of simulated and measured D-MESFET I-V characteristics.

In the opposite case where the input to the inverter switches from HIGH to LOW, the simulated **rise** time would be **smaller** (underestimation) than the real inverter. This is because the rise time depends on the difference between the D-MESFET and E-MESFET drain currents. Since the E-MESFET underestimates the drain current at low V_{gs} voltages, the charging current available from the D-MESFET is overestimated, and the simulated rise time of the output would be less than with real devices.

Overall, the opposite natures of the errors in rise and fall times mean that they tend to compensate each other over a period of transitions. Simulated propagation delay for an inverter with a fanout of 2 produced a figure of approximately 124ps. This compares reasonably well with the 110ps quoted by Honeywell [87].

With these specific simulation error figures and conditions in mind, the following overall strategy for design and simulation can be drawn up:

(i) Fanin design rules are strictly followed owing to the underestimation of leakage currents.

(ii) Dynamic nodes should be avoided.

(iii) Circuits are oversimulated by at least 15% in speed. This is to account for the 15% maximum error in I_{ds} of the simulated D-MESFET device.

Together with the general rules discussed in Chapter 3, and the inclusion of process variations of device parameters, simulations made with this modified SPICE program can be used to assess circuit designs with reasonable confidence. The test results obtained from test circuits also confirm the correctness in functionality of the simulations. These will be discussed in more detail in the next Chapter.

Chapter 5

GaAs Bit-Serial Cells and Systems

In Chapter 2, previous work based on bit-serial architectures in silicon was reviewed, and the inherent advantages over bit-parallel circuits were identified. In this Chapter, the design of the bit-serial cells are discussed in some detail. Simulation results are presented to verify their correctness where necessary.

In Chapter 2, it was concluded that some of the problems of GaAs MESFET technology and design can be alleviated by adopting bit-serial techniques. In particular, the increased functionality and efficiency of operators and of the chip as a whole, single-wire communication rather than bus logic, and the ease of partitioning and pipelining, are most attractive. Bit-serial circuits are also inherently more testable than bit-parallel circuits [21].

In a single bit-stream system where a complete system word is represented as a 2's complement signal on one wire, a major drawback is the need for a higher clock rate to compensate for the time-spreading of an n-bit system word. Increasing the number of pipelining stages does alleviate this problem, but at the expense of hardware overhead. Ultimately, a single stream bit-serial circuit designed for maximum possible clock rate cannot match the speed of a fully pipelined bit-parallel operator. This point can be illustrated by the 10.5ns multiply time of a 16×16 bit parallel multiplier [96]. A bit-serial Booth's multiplier [78] would require a clock rate of approximately 3 GHz to achieve the same speed. (The latency of a Booth's multiplier is discussed later.) Although this clock rate may be achievable by pushing GaAs DCFL to its limit, the bit-parallel figure is by no means the fastest possible. This can only be overcome by introducing some form of parallelism.

On the plus side, whereas the bit-parallel chip consisted of 224 full adders and 16 half adders on a $4 \times 4\text{mm}^2$ die, a bit-serial multiplier would consist of 8 full-adders and some control logic connected in series, leaving space for other functions to be integrated. However, the large number of adders in the parallel multiplier form a very regular array, while the bit-serial adder is larger and less easy to compact with the additional control and recoding logic. Therefore the overall space advantage is not as large as the numbers might suggest, but still substantial nonetheless. The space advantage of bit-serial circuit is illustrated with an FIR example later on.

To put the circuits discussed in this Chapter into perspective, they are single bit-stream circuits that were designed for a nominal clock rate of 500MHz, and the circuits were pipelined to an extent adequate for this speed. It was not one of the design objectives to explore the limits of the process in question — an impossible exercise especially in this case, where a foundry service was not used and a number of important parameters were variables in such an experimental process (Appendix 15). Instead, they were aimed at demonstrating the functionality and flexibility achievable in GaAs by adopting a bit-serial architecture, using special circuit techniques. At the chosen clock rate, they operate approximately 10 times faster than previously published bit-serial circuits in silicon [85,114].

The differences between GaAs and silicon devices in general, and between DCFL and silicon NMOS circuit design in particular, have already been discussed. It is clear that a direct "mapping" of silicon logic design onto DCFL is not possible and therefore the GaAs circuits have been designed from the basic latch structure upwards. Circuit engineering aspects are discussed in some detail in the subsequent sections. But before going on to the details , it is important to define the adopted signal conventions. These are discussed in the next section.

Conventions

A number of fixed timing conventions and terms, based on those of Renshaw and Denyer [21], are adopted in this work. These are summarised below:

(i) LSB-first fractional 2's complement fixed point data format with two sign bits (guard bits).

(ii) Two-phase non-overlapping clocking scheme, $\phi 1$ and $\phi 2$.

(iii) On-chip data signals are stable during $\phi 2$.

(iv) Off-chip data is stable during $\phi 1$, if the inputs and outputs are pipelined. Otherwise, all signals are stable during $\phi 2$.

Fig. 5.1 gives an example, showing the 4 system signals and their relationships. In addition, the cells in this work were designed for the following speed specifications:

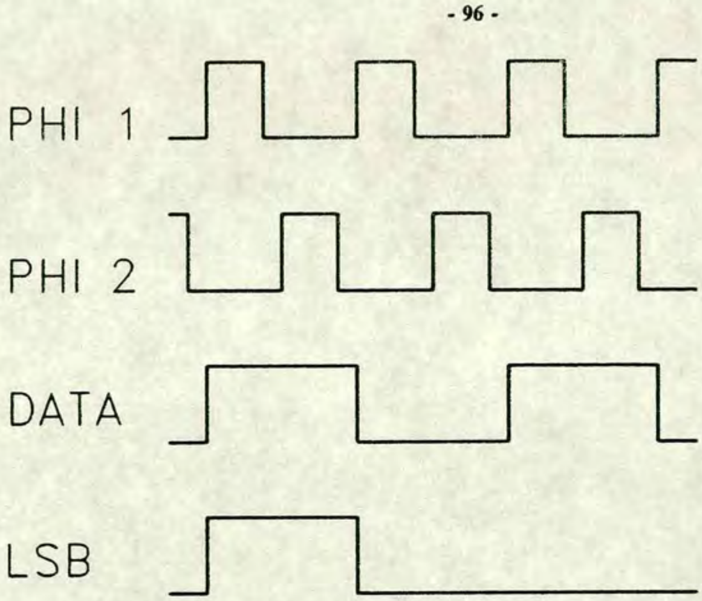


Fig. 5.1 System timing convention

(i) Maximum clock rate ($f_{clock(max)}$) of 500MHz.

(ii) Maximum data word rate (or sampling rate) is given by:

$$\text{data rate}_{max} = \frac{f_{clock(max)}}{(\text{system word length})}$$

A data word is defined by the word sample that is to be processed by a bit-serial system. A system word is defined by the internal word used for computation within the system. Both word lengths are initially programmable for a certain application, but cannot be changed when the system is in operation. The system word length is at least 1 bit longer than the data word length because 2 sign bits are needed for correct operation of the multiplier. If the system word length is more than 1 bit longer than the data word, then binary 0's may be inserted at the least significant positions of the system word when the data word is programmed into the system word (if the data word is not scaled down before processing). Since the system word appears lsb first, these 0's are called "leading" 0's.

The data word length is restricted by the precision of analogue-to-digital converters, since the maximum system word length is programmable up to 31 bits. The choice of system word length depends on the precision of the arithmetic required and the type of the system being designed. In a recursive system, the required system word length will also be determined by the operator latency owing to time align signals fed back from different parts of the system. They incur the biggest penalty on the maximum sampling rate achievable with the system.

The latency of an operator is defined by the number of clock cycles between the least significant bits of its input(s) and output, as illustrated by an example in Fig. 5.2. All operators have fixed latencies, independent of system word length.

5.1. Latch structure requirements

By utilising pipelining as extensively as possible to maximise the system throughout rate and area/time efficiency, the penalty of bit-wise processing can be partly overcome. The pipelining latch design is therefore crucial to this work. This section analyses in detail the requirements on the latch structure for use in GaAs bit-serial circuits. They form the basis of comparing different conventional designs and choosing the adopted latch design, discussed in subsequent sections. Firstly, the timing constraints of pipelining are identified.

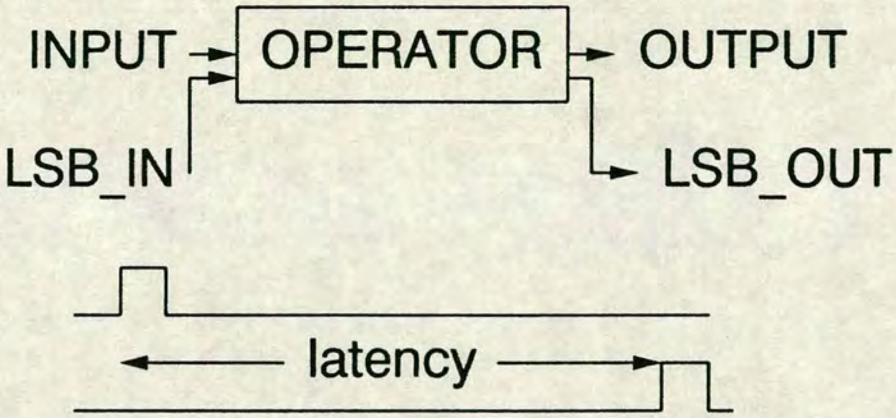


Fig. 5.2 Latency of an operator

Pipelining timing constraints

The use of pipelining to increase data throughput rate is a well established technique and involves the partitioning of logic with latches so that data can be clocked in at higher rates. The maximum allowable clock rate in such a system is given by:

$$\mathrm{clock}_{\mathrm{max}} = \frac{1}{2(\mathrm{delay}_{\mathrm{latch}} + \mathrm{delay}_{\mathrm{logic}} + \mathrm{delay}_{\mathrm{wire)}}} \quad \text{— Eq. 5.1}$$

Here it is assumed that all delay terms are taken as propagation delays (t_{pd}), and that the clock has 50% duty cycle. Propagation delay (t_{pd}) is typically defined as the delay between the 50% points of input and output signal voltages.

As technology improves and device switching delays are of the order of 10ps, the interconnection term becomes more and more dominant. If Josephson junctions [38] were used for these designs, the logic gates become more like zero-delay functions, and the $delay_{wire}$ terms and the fanout capability of the circuits would become the limiting factor.

Given that t_{pd} are in the region of 100ps to 200ps in the Honeywell GaAs process, the interconnection delays can be considered insignificant within a latch. The wire delay between a latch and related logic can be significant and comparable with t_{pd}, depending on the particular pipeline stage. However, it is safe to assume that the latch and logic delays are dominant under most circumstances.

In designs with low throughput rates, the latch or register delay is insignificant compared to the total logic gate delay, which might be 5 to ten times that of the latch. As the number of pipelining stages are increased, as is necessary and inherent in bit-serial circuits, even to the extent of pipelining every logic gate delay, the logic and latch delays become similar in magnitude. Taken even further, the logic can be absorbed into the latch/register structure, depending on the latch and logic design. This latter technique is discussed in the context of the chosen design methodology in a later section. If the latter can be achieved, then the $delay_{latch}$ and $delay_{logic}$ terms are merged together in the equation above. Therefore, ultimately, the design of the latch or register is the dominant factor that limits the speed of bit-serial circuits in terms of circuit design.

Latch properties

At this point, it is clear that the pipeline latches or registers should ideally possess all of the following properties:

(i) Compactness.

(ii) Low power dissipation.

(iii) Good noise margins.

(iv) Insensitivity to clock skew.

(v) Low clock and signal loading.

(vi) Possibility of merging of logic and latch functions for ultimate throughput.

Compactness and low power dissipation are mandatory in VLSI, especially in circuits that make extensive use of latches. Pipelining inevitably increases total circuit area compared to non-pipelined circuits (both bit-serial and bit-parallel). Therefore the more compact the latches are, the smaller is the penalty in area and power dissipation, and more levels of pipelining can be applied.

GaAs DCFL has demonstrated clear advantages over silicon ECL in power dissipation in commercially available gate arrays (Chapter 2). This should not be sacrificed for the sake of higher throughput and integration if GaAs is to remain competitive.

Noise margin considerations in GaAs DCFL circuits have already been discussed in Chapter 2 & 3. It was concluded that with the Honeywell process, noise margin should not be the major yield factor even in VLSI systems. However, if non-DCFL circuits are involved, the issue needs to be considered carefully before they are used.

Insensitivity to clock skew can be measured by the maximum permissible amount of relative skew between the clock signals driving a two-phase latch. (Here two-phase includes overlapping and non-overlapping clocks.) This type of skew is called local clock skew. Global skew represents the skew between one phase of the clock signals at two different physical driving points on an IC. The two are clearly related numerically but local skew is of concern only in this discussion.

With any real system of significant size, local and global clock skew always exist. For a latch, local clock skew depends on the distributed parasitic capacitances, gate capacitive loading, and the interconnect and contact resistances at the tapping points of the clocks for the latch in question. Distributed capacitances are not always easy to estimate and simulate in large circuits. A compact and simple latch again would reduce the problems of estimating them. Careful clock distribution and buffering are also important in reducing the local skew.

Finally, referring back to the maximum possible clock rate in a pipelined circuit (Eq. 5.1), if the latch structure lends itself to be modified such that the latch delay term can be eliminated and merged with the logic delay term, then even higher throughput would result. This is clearly not always possible and can be considered as an optional rather than a major requirement of a latch design.

5.2. Comparison of latch structures

In this section, different latch designs are discussed and compared, based on the above criteria. While DCFL is the assumed and preferred family, latches that uses pass transistors will also be considered.

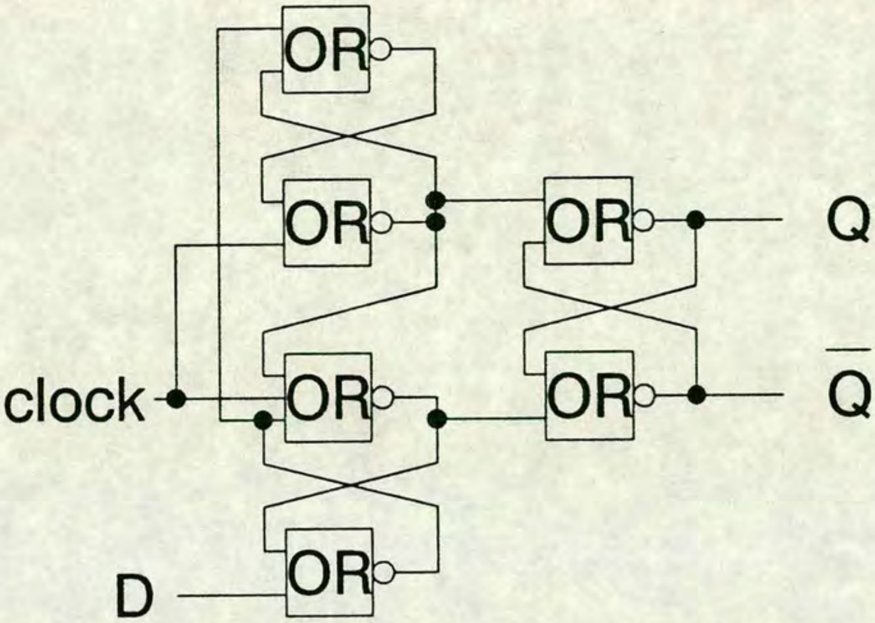


Fig. 5.3(a) Fully static latch I

Latches can be divided into three types: fully-static, fully-dynamic, and pseudo-dynamic. Fully-static latches are first considered.

Fully static latches

Fig. 5.3(a) and Fig. 5.3(b) shows two different fully static latches. They are edge-triggered flip-flops commonly specified as a "standard" cell in GaAs. The latch in Fig. 5.3(a) is negative edge-triggered, while that in Fig. 5.3(b) is positive edge-triggered. They can be directly implemented in DCFL with properly buffered and aligned clock signals. The worst case gate delay through both latches in Fig. 5.3 is $3\times t_{pd}$, assuming that the data input signal is stable before clock and $\overline{\text{clock}}$ changes in each case. Because of the delay between cause and effect, the restrictions on local skew are eased and the clock edges do not need to be very sharp. This tends to counteract the loading effects on the clock signals caused by the DCFL inputs. A further advantage of this type of latch is the inherent ability of static circuits to

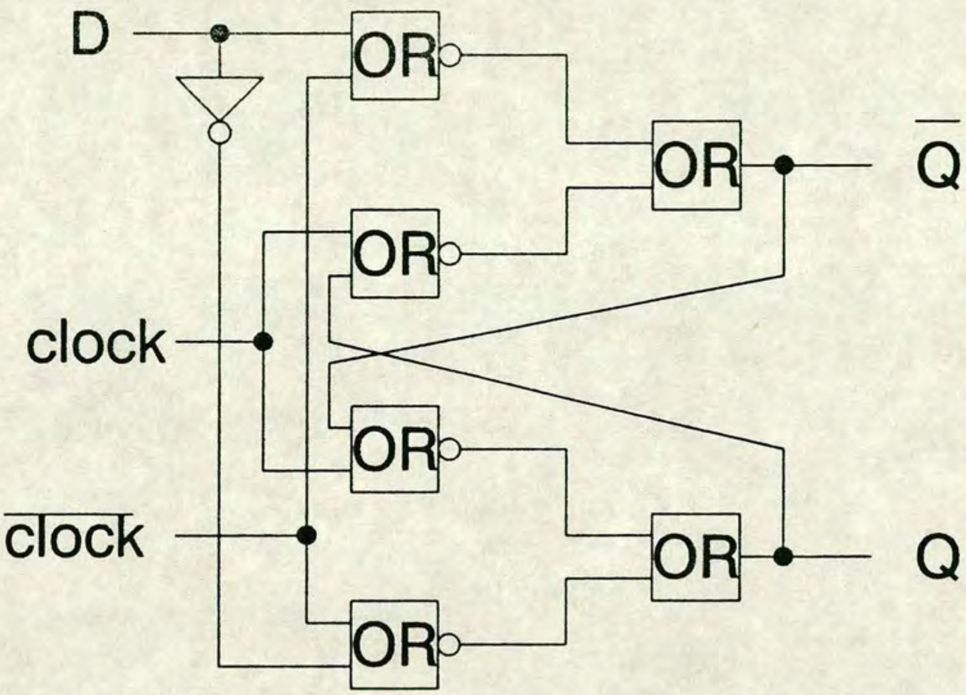


Fig. 5.3(b) Fully static latch II

retain their output states even if the clocks are stopped. As a result, they are reasonably reliable and easy to use; hence their popularity.

The major drawbacks of these latches are large layout area, DCFL power consumption, and DCFL clock and signal loading. If they were to be used extensively, as required in GaAs bit-serial systems, the overhead in these parameters have to be weighed against their conventionality.

Fully-dynamic and pseudo-dynamic latches

By their very nature, dynamic circuits rely on the storage of charge at critical signal carrying lines and nodes. In GaAs, the semi-insulating substrate and the forward biased gate-drain or gate-source junction result in high leakage, low capacitance nodes. These properties impose a lower limit on the clock rate for correct circuit operation. Also, if one of the clock signals in a two-phase system is stopped, a fully-dynamic latch cannot retain the data. This rules out any simple standby state in a dynamic latch.

One other problem with floating charge storage nodes is that they are notably susceptible to noise. This is especially detrimental to dynamic latches that consist of a DCFL inverter driven by a pass transistor, such as that shown in Fig. 5.4, because

DCFL has a relatively small swing, and the noise margins need to be restored by increasing ratioing.

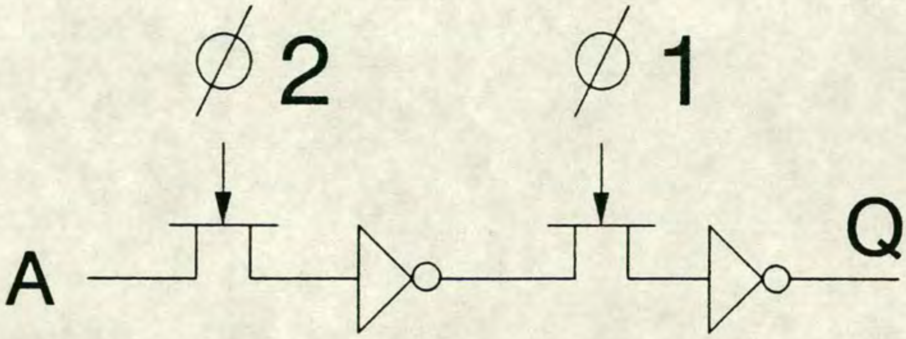


Fig. 5.4 Fully dynamic GaAs latch

Another related problem, well known in both dynamic and pseudo-dynamic silicon MOSFET circuits, is the capacitive coupling between the gate-drain and gate-source junctions. This introduces spikes at the input and output of the pass transistor as it switches on, especially if the voltage difference between the clock and logic signals is large. However, provided that the driving logic gate is large enough to charge or discharge any dynamic nodes at the required rate, these spikes do not affect the correct operation of a latch. Also, the MESFET does not have an insulated gate and thus has a lower gate capacitance. For instance, at zero gate bias, the gate capacitance of a MESFET in the Honeywell process is approximately 0.5 $fF/\mu m^2$ [87], compared to a figure 1.4 $fF/\mu m^2$ for a MOSFET in a $1\mu m$ process [106]. If the pass transistor is an E-MESFET, a different coupling mechanism exists which is more severe, in addition to its longer t_{pd} than a D-MESFET, as will be seen later.

Overall, the main advantages of compactness and low power dissipation will have to be weighed against its frequency limit and noise problems, when considered for extensive use in general purpose pipelining. However, they should not be ruled out for circuits where their disadvantages are not important.

Pseudo-dynamic latches

Psuedo-dynamic latches have been widely used in silicon NMOS circuits [83], an example of which is shown in Fig. 5.5. Here an enhancement mode MOSFET pass transistor is used so that it can be driven by the output of a normal logic gate. Compared to fully-dynamic latches, they do not have a lower operational frequency limit, but are generally larger in size because of the refresh circuit. They can work down to DC provided that certain conditions are satisfied by the two phase clocks. The same principle can be readily applied to GaAs, although there are major differences in terms of circuit engineering. Figs. 5.6(a) and 5.6(b) show two versions of psuedo-dynamic latches that can be implemented in GaAs, using E-MESFET and D-MESFET pass transistors respectively. Detailed analysis of these two latches reveals their problems and the ways they differ from the latch as implemented in silicon.

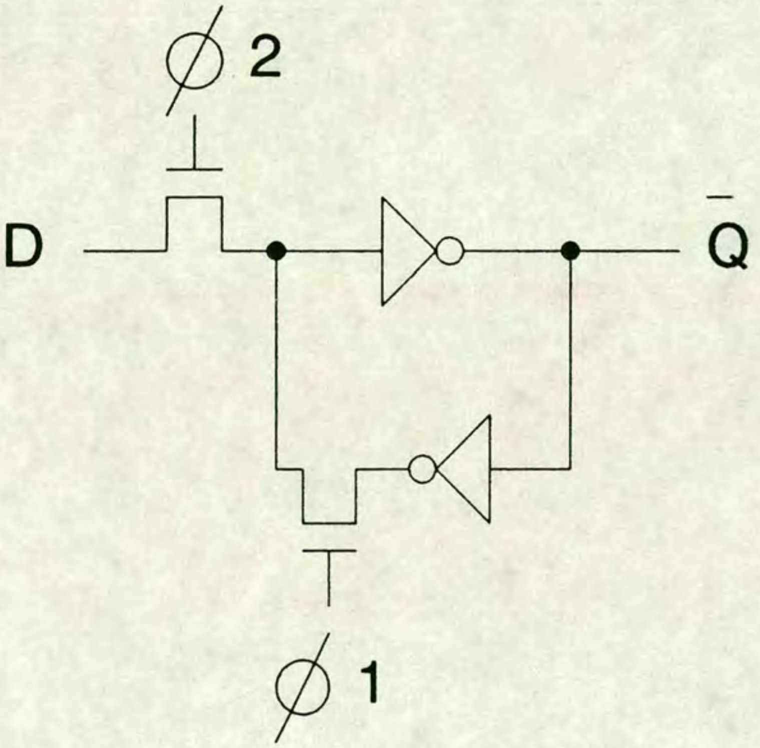


Fig. 5.5 Silicon NMOS pseudo-dynamic latch

From a logic point of view, the GaAs and silicon based latches are identical. To accept new data, the forward pass transistor is turned ON while the feedback transistor is OFF. The data is latched, refreshed and isolated from the latch input by reversing the states of the pass transistors. It is crucial that when T1 is ON, the data input signal is stable. In addition, real skewed clock signals, like those shown in Fig. 5.7, must satisfy the following conditions for correct latch operation:

(i) The overlap time must be less than the total gate delay between two pipeline stages.

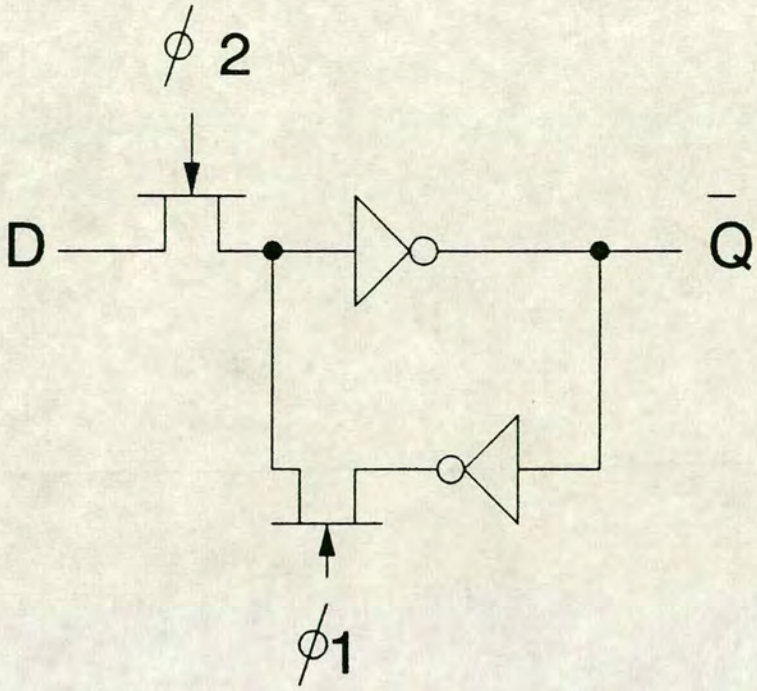


Fig. 5.6(a) GaAs pseudo-dynamic latch with E-MESFET pass transistor

(ii) The non-overlap time must be less than the minimum discharge time for a HIGH signal at the feedforward and refresh inverter inputs.

If these requirements are satisfied at all times, then the latches should work down to DC. Unfortunately, the similarities between the GaAs and silicon based latches end here. Their differences are highlighted by first analysing the GaAs E-MESFET pass transistor latch, shown in Fig. 5.6(a), the direct counterpart of the silicon NMOS latch.

The E-MESFET pass transistor latch

First consider the aspect of voltage levels in the GaAs enhancement pass transistor latch (E-MESFET latch). Problems arise when two or more latches are driven by the same clock signal. For simplicity of argument, consider the situation where two E-MESFET latches are driven by two independent input signals. There are 4 possible combinations of input states. To illustrate the problems of this latch, first consider two input states in particular, as listed below and shown in Fig. 5.8:

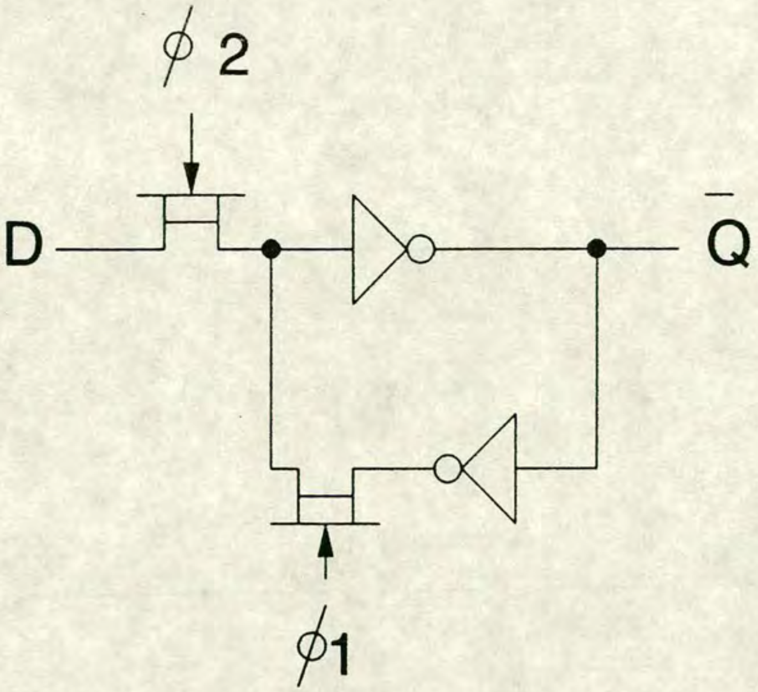


Fig. 5.6(b) GaAs pseudo-dynamic latch with D-MESFET pass transistor

(i) All inputs at logic HIGH of 0.75V, Fig. 5.8(a).

(ii) One input at logic LOW of 0.1V, Fig. 5.8(b).

For simplicity, the refresh paths of the latches are omitted in the diagrams. Assume that the clock signal is capable of a full swing of 0.1V to V_{dd}, as with the output of a DCFL inverter which is not clamped by a fanout inverter/NOR gate.

In Fig. 5.8(a), the clock goes HIGH and tries to pass two logic HIGH signals through the pass transistor and change the logic LOW at the input of the inverter (taken to be the source terminal of the pass transistor in this discussion, even though self-aligned gate MESFETs are symmetrical). Since logic LOW is about 0.1V, the clock rises to a maximum voltage of 0.85V until it is clamped by the Schottky gate. As the inverter input voltage rises due to the drain current from the pass transistor, the clamped clock voltage also rises in unison. Eventually V_s of the pass transistor is clamped by the inverter, and the clock signal is clamped at about 1.5V. This effectively links the clock and data voltages together. In this case, apart from the significant gate current drain on the clock and the power dissipation over two diode drops, the operation of the latch is not affected adversely. The gate current in fact helps to sustain V_s (the input to the feedforward inverter).

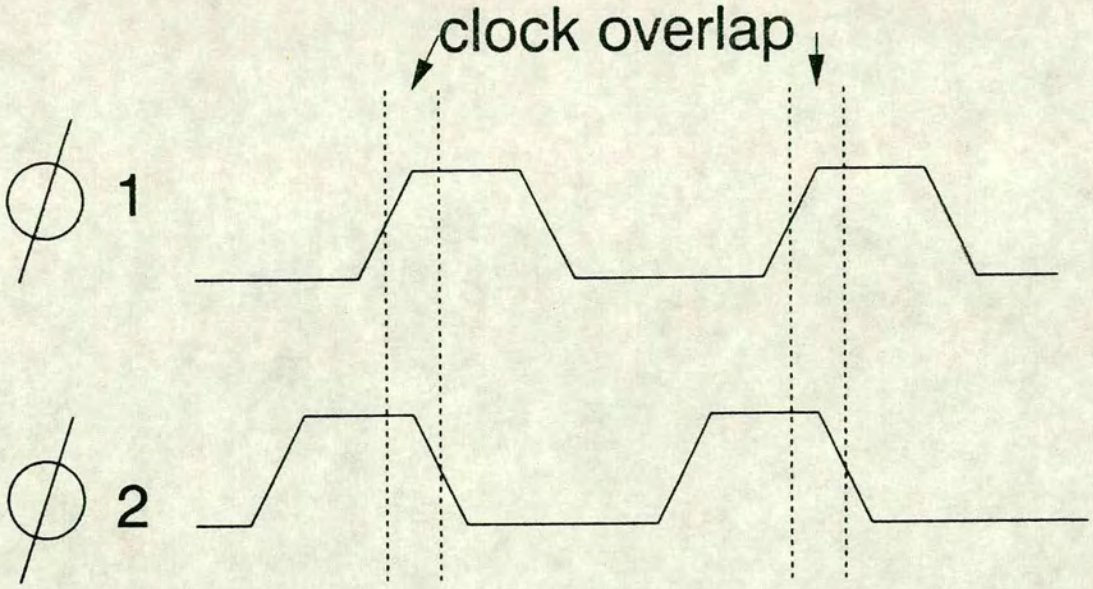


Fig. 5.7 Skewed 2-phase clock signals

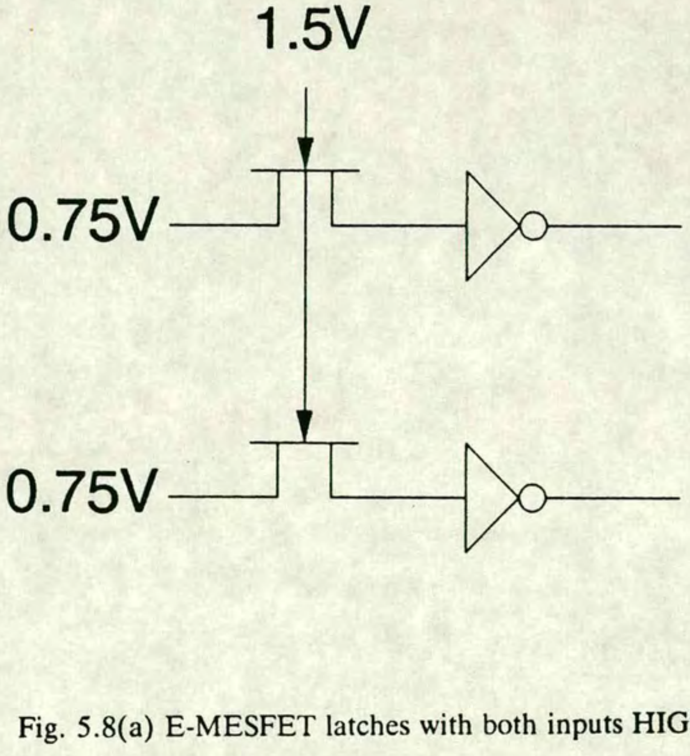


Fig. 5.8(a) E-MESFET latches with both inputs HIGH

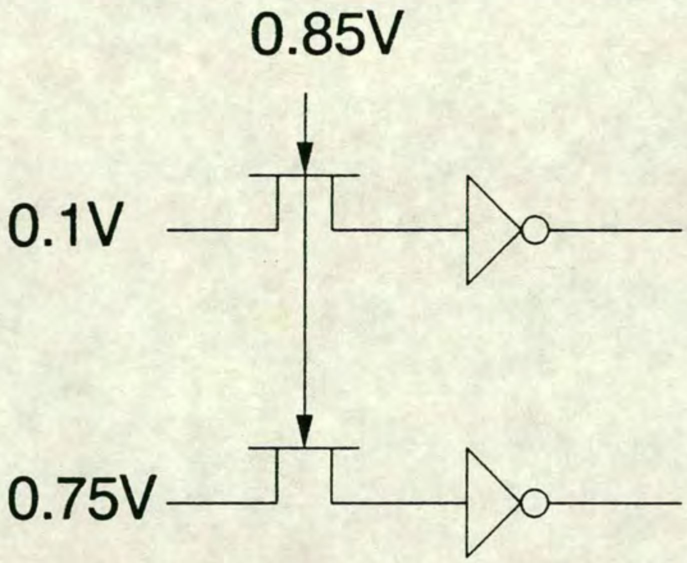


Fig. 5.8(b) E-MESFET latches with one input HIGH

In the second case, shown in Fig. 5.8(b), where one input is LOW (0.1 V) and the other is HIGH (0.75 V), the situation is more complex. The LOW input signal now clamps the maximum clock HIGH voltage to about 0.85V, in contrast to the previous case of 1.5V. As a result, the V_s of the other pass transistor can only rise to an absolute maximum value of $V_{gs}-V_t$ (about 0.65V in this process). Fig. 5.9 shows the simulation of this case, with standard inverter sizes (40μm E-MESFET gate width) and pass transistors/inverter ratio of 1:1. In practice, even this value cannot be reached due to other adverse factors discussed later. Noise margin is clearly affected. Inevitably, this latch is also slower than the other owing to the rising channel resistance of the E-MESFET as it approaches cut-off.

More important is the fact that in this case, the V_s of the pass transistors in two (or more) independent latches are now linked to each other through the clock line. If a fluctuation appears on the LOW signal to which the clock is clamped, perhaps due to clock/signal and signal/signal capacitive coupling, the clock also fluctuates. The exact amount of variation depends on the size of the pass transistor and the parasitic capacitances. A positive going fluctuation will pass through the clock line and cause the output of the other pass transistor to rise, which is beneficial to the HIGH signal. A negative fluctuation will pull down the clock line and reduce the

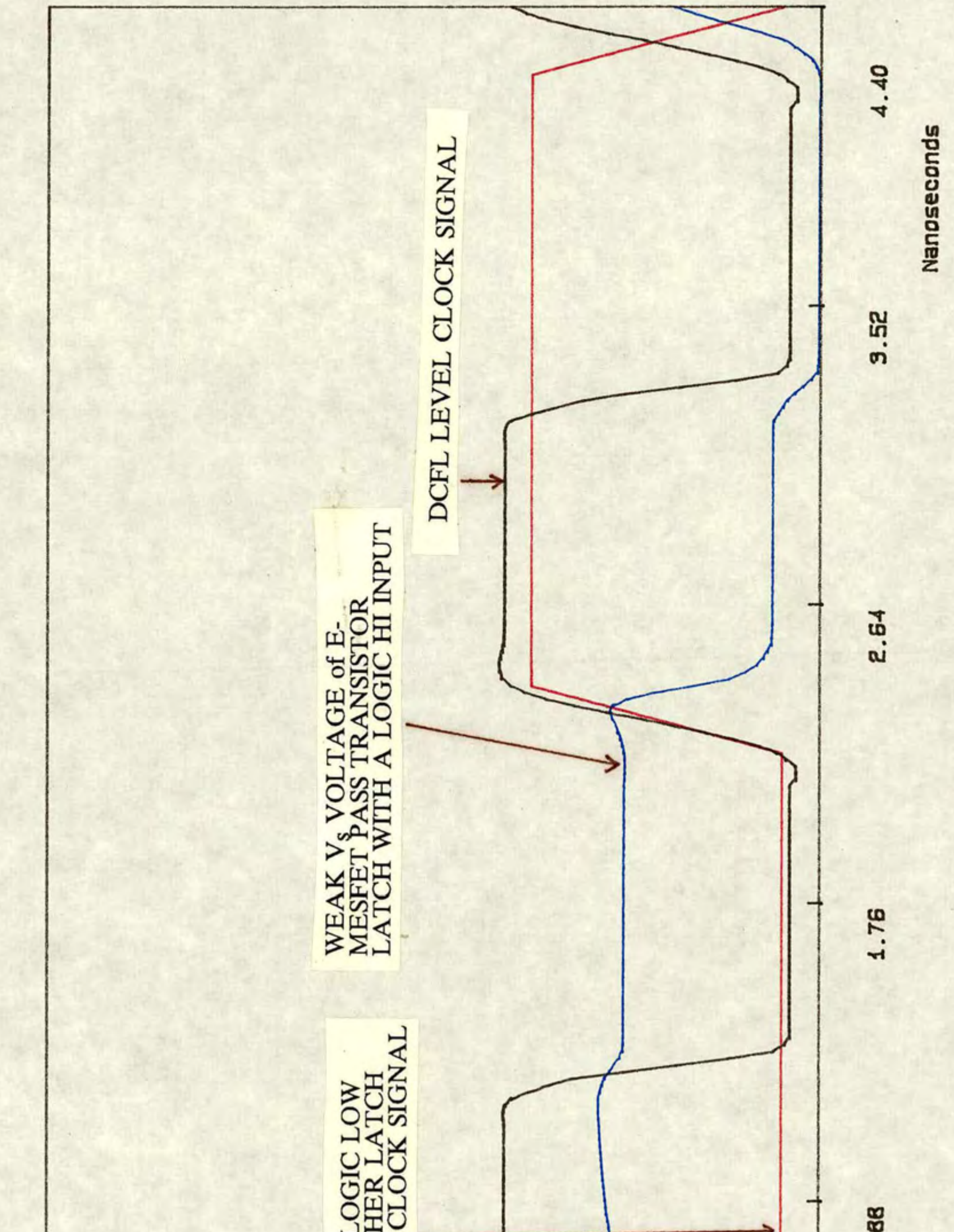


Fig. 5.9 Simulation of E-MESFET latch, one input LOW

HIGH voltage of an unrelated pass transistor. On the other hand, fluctuations on the HIGH input signal do not affect the clock line and other signals, except through a limited amount of capacitive coupling, because the gate-drain junction of the pass transistor is not forward biased.

In a third possible input condition, where both latch inputs are LOW, the situation is similar to the all HIGH case, and fluctuations on either input do not affect the other through the clock line.

Now further consider the problem of noise margins in the mixed input case. As mentioned earlier, V_{source} of the pass transistor can only rise to a theoretical maximum of 0.65V (with V_t of +0.2V in Honeywell process). This is a reduction of 100mV compared to the nominal logic HIGH, at the input to the inverter. Unfortunately, this maximum of +0.65 V cannot be reached in practice because the pass transistor is in series with the inverter E-MESFET, which is not strongly forward biased. Together they form a potential divider, as illustrated in Fig. 5.10. The voltage drop across the pass transistor, δV, depends on the DC resistances of the two E-MESFETs, and can be reduced by ratioing their gate widths. This contrasts with silicon NMOS where ratioing of pass transistor and inverter is not normally necessary. A larger ($W_{pass\ transistor}$) to ($W_{inverter}$) ratio will result in a smaller δV. Using a very large pass transistor cannot totally eliminate the problem and has the drawbacks of higher clock loading and larger latch size. In practice, with the Honeywell process, a ratio of 1:1 was found to be acceptable in practice (Fig. 5.9). Therefore, a further small voltage degradation of typically 50mV across the pass transistor cannot be avoided.

Another voltage degradation factor stems from the ratioing and fanin sensitivity of DCFL gates. So far it is assumed that logic LOW is at 0.1V. In real circuits, it is not uncommon that the pass transistor may be driven by a multi-input NOR gate, which produces a "strong" logic LOW of 50mV or even less when all its inputs are ON. Also, to maintain noise margin, it is common practice to increase β_R ratio beyond that needed for a nominal 0.1V LOW to compensate for possible IR drop on the ground rail. This means the logic HIGH voltage of the clock signal of the pass transistor will be clamped to an even lower voltage, limiting the maximum V_s even further.

All these factors together make V_{source} significantly lower and weaker than normal DCFL levels, requiring very high β_R ratio and in turn increasing the load on the pass transistor and overall size. Fig. 5.11 shows a simulation of the E-MESFET GaAs latch driven by a 3-input NOR gate with all its inputs ON. The V_{source} of the

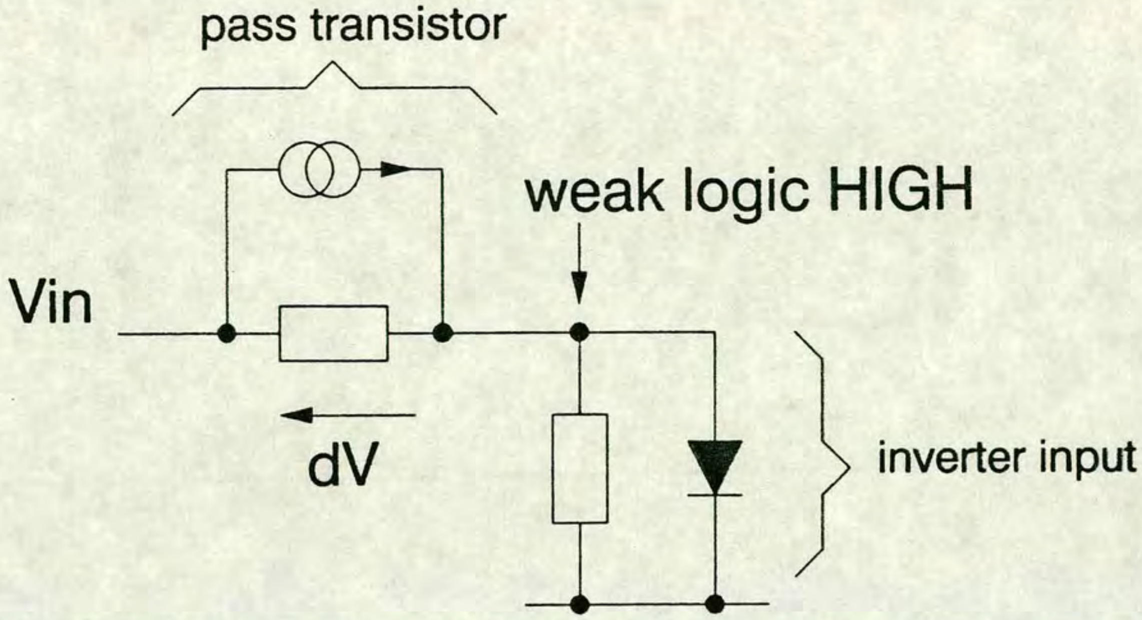


Fig. 5.10 Simplified E-MESFET pass transistor latch equivalent circuit

pass transistor reached approximately 0.55 V, a 200mV drop from the nominal logic HIGH voltage. For worst case design, an inverter β_R ratio of at least 15:1 will be needed, compared to the normal 3:1 ratio adopted for "standard" circuits. An even higher ratio is needed if a lower D-MESFET V_t is used. Even with such high β_R ratio, capacitive coupling is not yet taken into account. If the ratio of pass transistor to inverter gate is 1:1, then all pass transistors (both forward and refresh in a psuedo-dynamic latch) will need to be increased by the same factor. Therefore in order to make the E-MESFET latch totally reliable under worst case conditions, the resulting physical size, clock loading, and power dissipation would make it impractical.

Finally, although one can restore the noise margin of the inverter to the same value as the normal DCFL gate, the absolute noise margin between the HIGH and LOW input states is still significantly reduced because of the combination of the small voltage swing and the proximity of the threshold voltage. This is best understood by looking at the two transfer curves of a normal DCFL inverter and one with a high β_R, as shown qualitatively in Fig. 5.12. Allowing for process variations in V_t

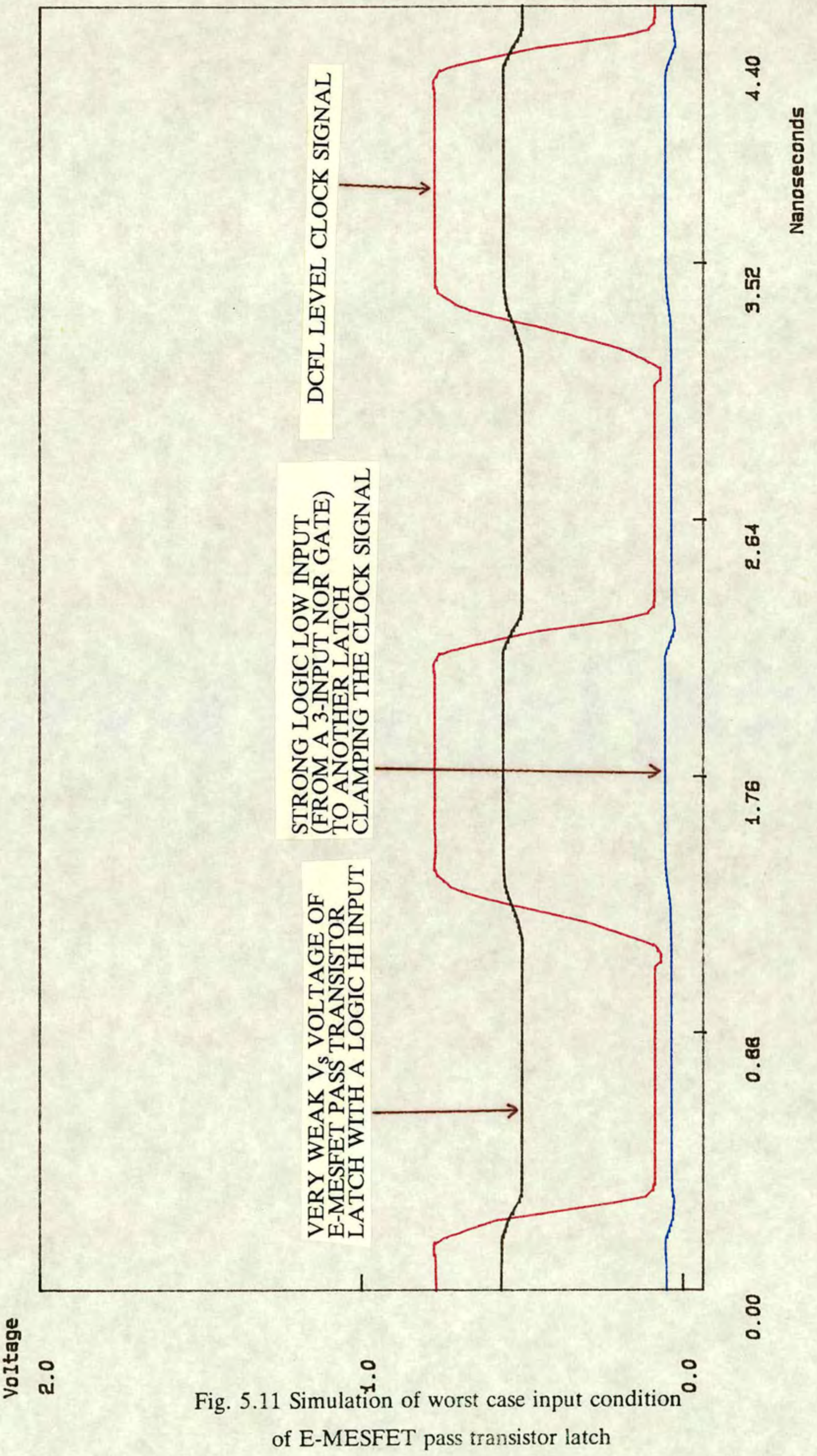


Fig. 5.11 Simulation of worst case input condition of E-MESFET pass transistor latch

and IR drop on the Ground rail, which directly affect logic switching thresholds, large scale use of such latches becomes unreliable.

All the above arguments can easily be extended to the general case of any number of E-MESFET latches. These problems also apply to any steering logic implemented with E-MESFET pass transistors. In the latter case, the logic and steering signals are voltage coupled through any forward biased gates.

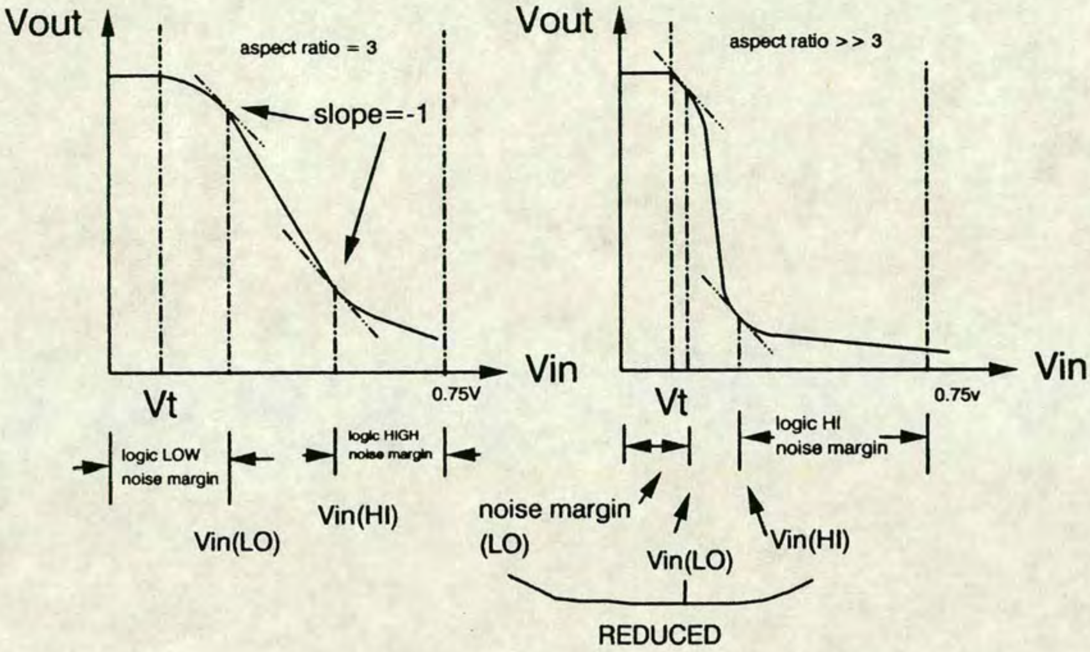


Fig. 5.12 Relationship between transfer curve and noise margin in a high-ratio DCFL inverter

Previous work in GaAs has used E-MESFET pass transistor latches in an SSI scale [116]. They have also been used in transfer logic [23], using E-MESFETs, and showed promising results. The points raised in the above discussions have not been fully addressed in these works. Their practicality for general pipelining in a large chip is unclear due to the small scale use of the former and the different usage of the latter. Nevertheless, the E-MESFET pass transistor latch appears to be unattractive in terms of circuit engineering. Its sole advantage is that it can be driven directly by DCFL gates. In the next section, the D-MESFET version is discussed and its feasibility is compared to the E-MESFET.

The D-MESFET pass transistor latch

Now consider the case of using the D-MESFET as a pass transistor. By virtue of its negative V_t ($-0.5V$ in the Honeywell process), several important advantages can be identified:

(i) There is a larger operating gate voltage range before it is forward biased.

(ii) The gate-drain or gate-source junction does not need to be forward biased to obtain the same DC drain-source current as a forward biased E-MESFET of the same size.

(iii) As a pass transistor, a smaller sized D-MESFET is capable of the same speed (or same t_{pd}) as an E-MESFET.

Advantage (i) is self-evident. In (ii), some figures of the DC drain current at a V_{ds} of 0.65V (logic HIGH − logic LOW voltage) help to demonstrate the point. At a gate voltage of +0.25V, a D-MESFET produces approximately the same drain current as an E-MESFET at a forward biased gate voltage of 0.75V. Since the D-MESFET is not forward biased, lower gate current and hence clock loading, and lower latch power dissipation result. The clock generator can also be smaller. Ideally the clock driver should act as a voltage source and have zero output resistance. In practice, the lowest possible non-zero resistance can be obtained by operating a MESFET in the ohmic region (Chapter 2), i.e. at a high V_{gs} and a low V_{ds}, and ensuring that it does not go into saturation under maximum current loading conditions. This can only be achieved by using a sufficiently large device under forward bias. The D-MESFET pass transistor clearly alleviates this situation significantly.

An equally important consequence of (ii) is that the diode coupling problem of the E-MESFET pass transistor identified earlier is totally eliminated.

In terms of speed and size, advantage (iii) follows from the other points, especially when the D-MESFET is passing a logic HIGH signal. The negative V_t means that it does not approach as close to cutoff as the E-MESFET, when its output source voltage rises. Therefore a smaller width ratio (pass transistor : inverter pull-down device ratio) is sufficient for the same speed. The reduced size results in lower capacitive loading and coupling, and a more compact layout.

The main disadvantage of using the D-MESFET is that it cannot be driven directly by normal DCFL gates. Level-shifting and an extra voltage rail are necessary to switch the pass transistors OFF. This means that a lot of power may have to be dissipated at the level shifters in the clock generator, rather than in the pass transistors themselves. Depending on the clock voltage levels chosen, this may offset the advantage in terms of power, while the other advantages remain unchanged.

Following the same line of analysis as the E-MESFET pass transistor latch, consider the case that two D-MESFET pass transistors in two latches are driven by the same gate signal, with one latch input at logic LOW and the other at logic HIGH. A closer look at the voltage requirements reveal that for the D-MESFET to pass the logic LOW voltage of 0.1 V, a gate voltage of 0.1 V or more is adequate. However, for the latch with a logic HIGH input at 0.75V, the pass transistor gate voltage must be high enough to pass the signal, but should ideally avoid forward biasing the gate-drain/gate-source junctions. In the case of the E-MESFET, it was noted that the source voltage of the pass transistor cannot rise any further when

$V_{gs} < V_t$ (cutoff condition)

where

$V_{gs} = V_g - V_s$

i.e.

$V_g < V_s + V_t$

In the case of the D-MESFET passing a DCFL logic HIGH, cut-off occurs when:

$V_g < 0.25V$

Since the upper limit of the gate voltage is set by the forward biasing of the D-MESFET with an input LOW voltage, cut-off induced voltage degradation across the pass transistor can be avoided by ensuring that:

$0.25V < V_g < 0.85V$

The issue of noise margin due to V_{ds} drop across the pass transistor is the next to be considered. Assuming that the clock signal satisfies the lower and upper limits above, the channel resistance again forms a potential divider with the DCFL inverter being driven. Since the D-MESFET does not approach cutoff, the output

HIGH voltage, V_{source}, is limited by the channel IR drop, and **not** by the channel cut-off, unlike the E-MESFET case. This IR drop can again be reduced by ratioing the D-MESFET size to that of the pull-down E-MESFET in the DCFL inverter being driven. If the nominal HIGH voltage of the clock is taken to be +0.5 V, then a ratio of 1:2 is enough to ensure correct operation with adequate speed, and without being in danger of forward biasing any junctions. To turn the D-MESFET pass transistor OFF, a gate voltage of −0.5V should provide ample noise margin, since the logic LOW DCFL input voltage is always higher than 0V. From this discussion, the nominal swing of the clock signal can be taken to be ±0.5V.

Overall, the D-MESFET is clearly superior to the E-MESFET as a pass transistor, but with the disadvantage of the need for level shifting. By adopting the sizing and ratioing mentioned above, it is simple to illustrate the compactness of the pseudo-dynamic latch, as shown in Fig. 5.13. The pseudo-dynamic latch has 6 transistors. Two of these latch forms a shift register. Note that the power rails are not shown in this plot. Compared to a full static latch, the reduction in area is approximately 65%.

A direct comparison of speed between the E-MESFET and D-MESFET latches demonstrate the advantage of the latter, owing to the D-MESFET not being close to cut-off as its source voltage rises. This is shown in Fig. 5.14.

Owing to late fabrication of the GaAs circuits, a silicon NMOS analogue of this latch was fabricated and tested successfully. Two other versions of the latch were also on the same test chip. Their results are presented later on in this Chapter. They served to demonstrate the concept of using this type of latch on transistors which draw significant gate currents, such as the GaAs MESFETs.

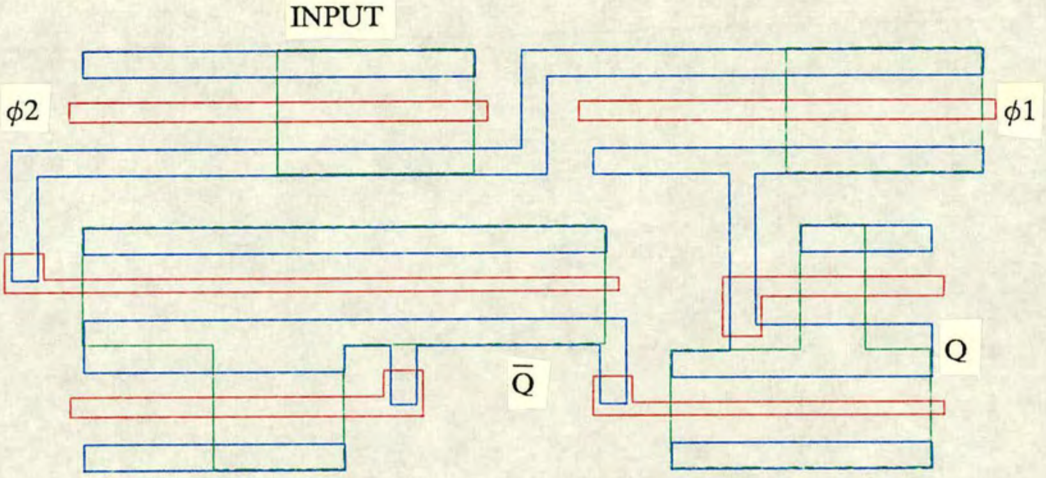


Fig. 5.13 Layout plot of pseudo-dynamic latch

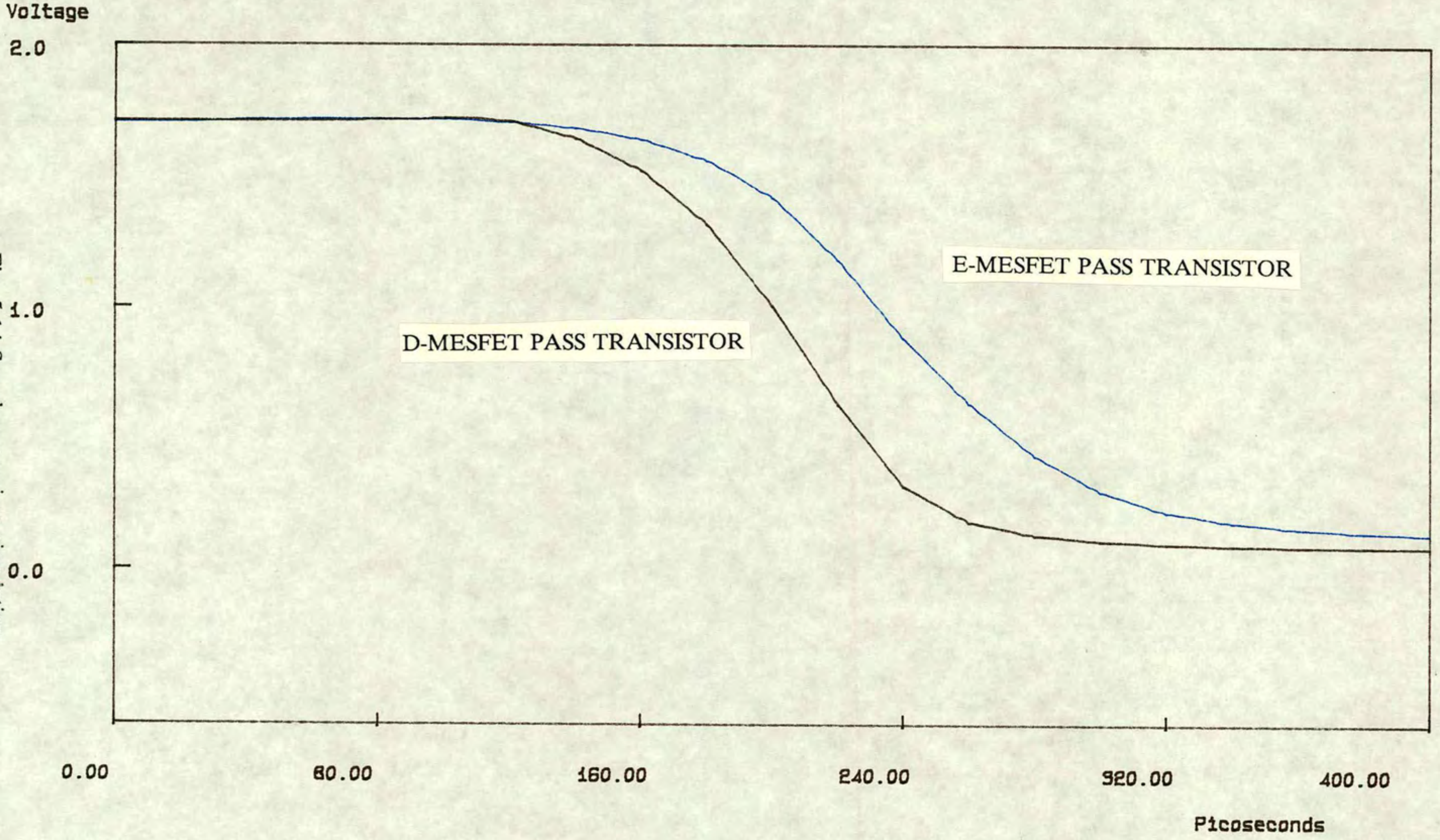


Fig. 5.14 Speed comparison simulation of E-MESFET and D-MESFET latches

5.3. Variations of the D-MESFET pseudo-dynamic latch

So far, the D-MESFET pass transistor latch has been chosen for its compactness, low power and clock loading properties. In addition to these properties, this section discusses how it can be readily modified and adapted in various ways to further enhance its suitability for general pipelining.

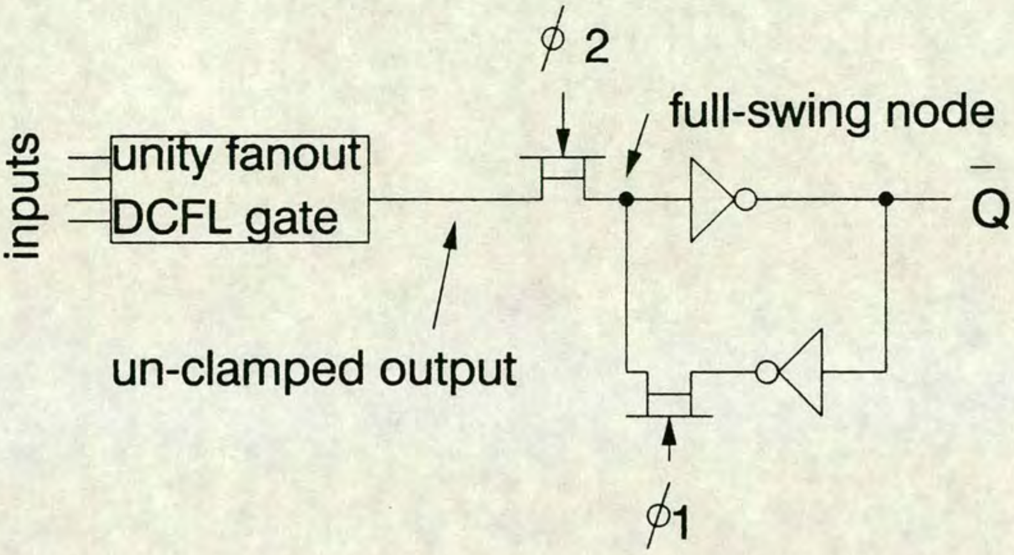


Fig. 5.15 Modified pseudo-dynamic latch with no weak HIGH signals

The voltage drop across the D-MESFET pass transistor and the resulting weakening of the input signal is clearly a significant drawback of this latch compared to the fully static latch. Although the noise margin of the input inverter can be restored, the weak signal appears at the most crucial node of the latch structure, and is still a potential hazard under real clocking conditions, when this node can become floating. If an E-MESFET pass transistor is used, then the weak signal cannot be eliminated because of the upper limit imposed by the gate voltage, irrespective of the input voltage (V_{drain}). With a D-MESFET pass transistor, no such limit exists on the source voltage because the junctions are not clamped to any input signals. Therefore it is possible to totally eliminate the "weak" node if the input HIGH signal to the latch is not clamped to 0.75V by a DCFL gate, and is allowed to to rise to V_{dd}. If this is satisfied, then the V_{source} of the D-MESFET can rise to the full DCFL HIGH voltage of +0.75 V. Such a configuration is shown in Fig. 5.15, which does not have a "weak" logic HIGH voltage at any point in the circuit. An IR drop still exists through the pass transistor, but no longer degrades the signal being latched.

Compared to fully-static latches, this technique is still substantially smaller in terms of area. This method is preferable in very large chips where the process control may be variable, and where noise margins are crucial, at the expense of the extra layout area and loss of latch symmetry.

One simple variation of the D-MESFET latch is to have a reset function in the forward path, as shown in Fig. 5.16. This allows the output of the latch to be forced to LOW, and is used in the programmable parallel-to-serial converter discussed later on.

Another useful, though not always applicable, variation is the merging of logic and the latch function into one circuit, as shown in Fig. 5.17. In this example, the logic NOR gate has replaced the usual inverter and feeds back its output into one of its inputs, thus eliminating one gate delay between pipeline stages. Because of the different refresh mechanism, the latched signal is *not* the input signal as in a proper latch, but is the output of the logic function in question. Clearly, this is only acceptable if the input signal has a logic fanout of 1. If a fanout of more than 1 is needed, then an extra latch can be connected in parallel with the logic, but sharing the same pass transistor. The problem with this method is the need for ratioing the pass transistor to the **total** size of the inverter pull-down devices (double that of the normal latch in this case). If the fanout of the signal is large (> 3), a very large pass transistor is necessary to maintain speed and limit IR drop. This poses a major clock/signal coupling problem due to the gate-source and gate-drain capacitances, even though the junctions are not forward biased. This in itself limits the use of this technique. Alternatively, separate latches can be used with the same logic signal, increasing fanout loading on that signal, but without the need to use one large pass transistor.

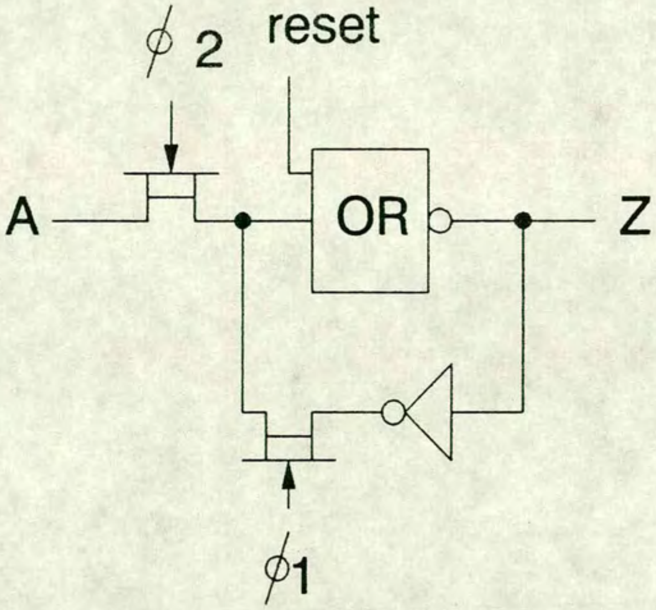


Fig. 5.16 Pseudo-dynamic latch with reset

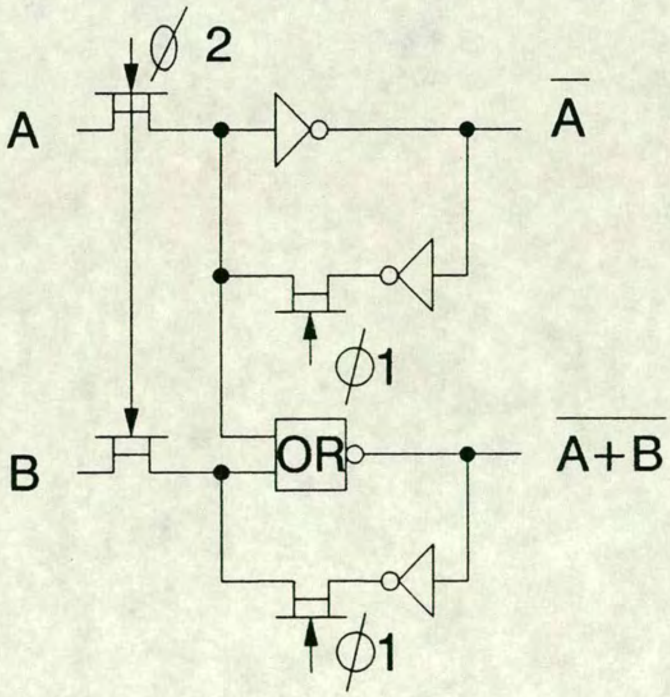


Fig. 5.17 Pseudo-dynamic circuit with merged logic and latch functions

5.4. A silicon NMOS study of the pseudo-dynamic latch

Three silicon NMOS analogue of the GaAs latch structure discussed above have been fabricated and tested. They were aimed at emulating the principle of operation of the dynamic and pseudo-dynamic latches in GaAs, where the transistors draw significant gate currents, by using readily available silicon technology.

As mentioned earlier, NMOS transistors differ in their insulated-gate structure and operation from the MESFET. The circuit schematics of the three fabricated latches are shown in Fig. 5.18. Their design principle had been to introduce an additional MOSFET at the input of every inverter. The gate voltage of this transistor was designed to be variable at an external pad, to emulate the gate current drawn by a GaAs MESFET. This MOSFET was laid out as a long gate length device, and acts as a resistor rather than a genuine pull-down device. With this MOSFET (Fig. 5.18) fully ON, by applying 5V to its gate terminal, the leakage current would be in excess of a fully forward biased GaAs MESFET, in order to demonstrate the correct operation of the pseudo-dynamic latch down to DC under such adverse conditions.

A photograph of the test chip is shown in Fig. 5.19. The oscilloscope traces of the three different latches are shown in Figs. 5.20 to 5.22. Fig. 5.20 showed that with a large leakage current, the latch without a refresh path failed at clock rates **below** approximately 10MHz. Both of the other two latches, which have passive and clocked feedback paths, worked correctly virtually down to DC. The clock signals used for the tests were 2-phase overlapping ϕ and $\overline{\phi}$ signals, instead of the normal non-overlapping clocks. The correct operation of the two feedback latches therefore also demonstrated their timing robustness.

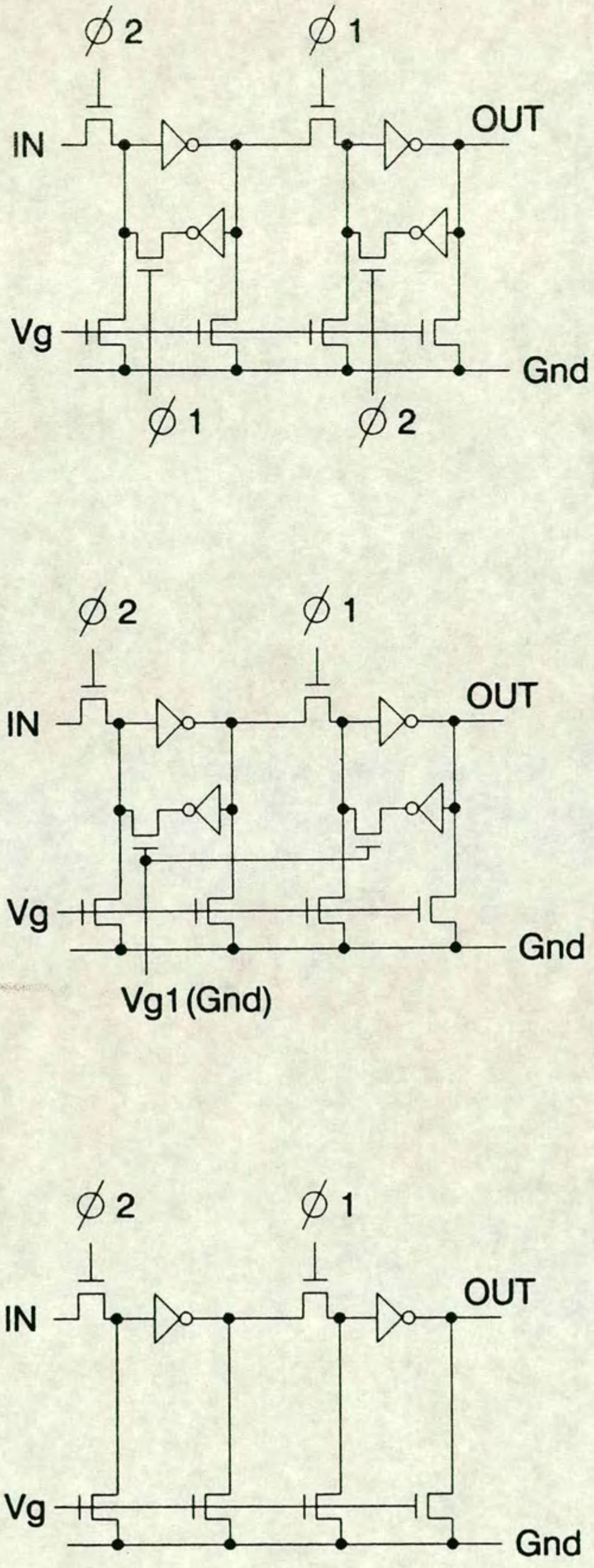


Fig. 5.18 Three forms of silicon NMOS latches with large gate current drain that were fabricated

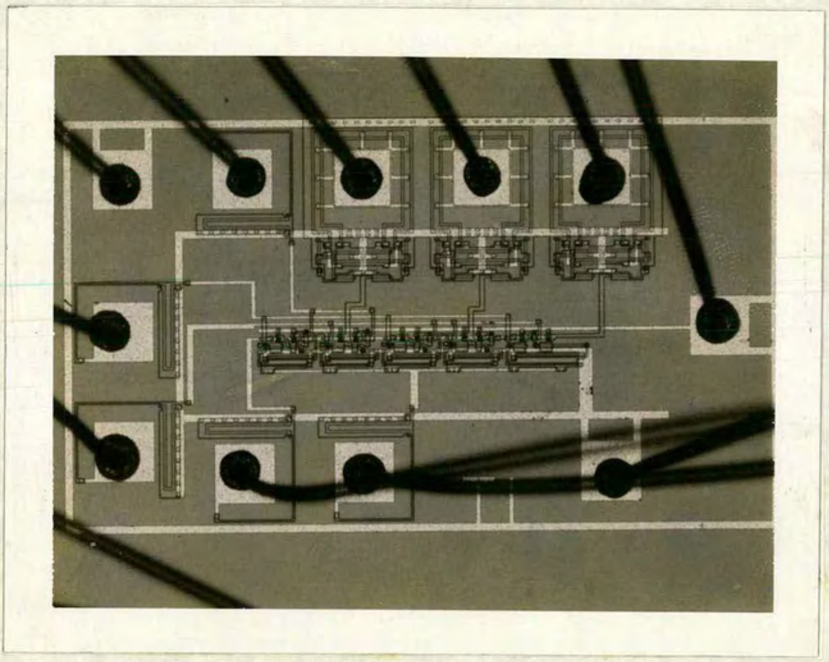


Fig. 5.19 Micrograph of the silicon NMOS test chip

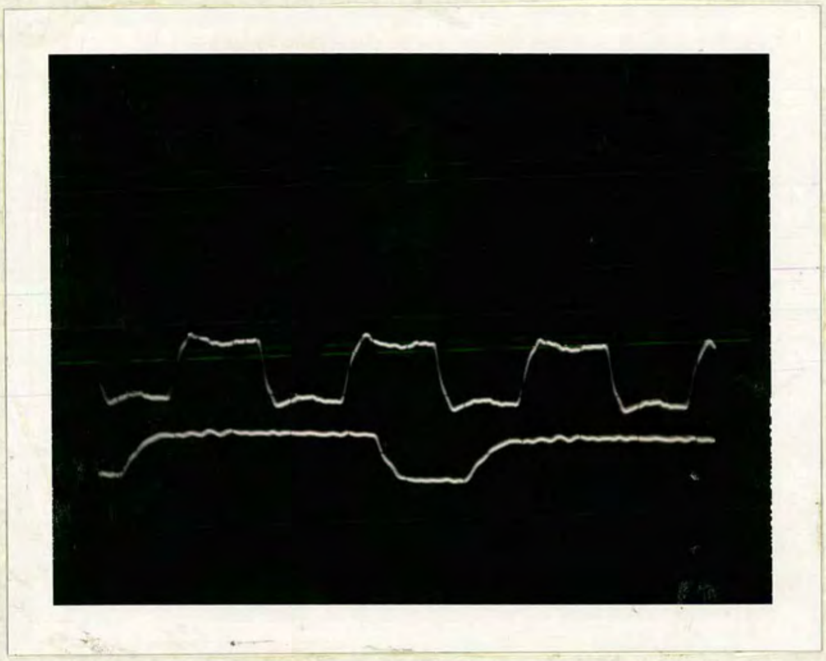


Fig. 5.20 Oscilloscope trace of fully dynamic latch failure
below 10MHz clock rate
(top: clock, bottom: output)

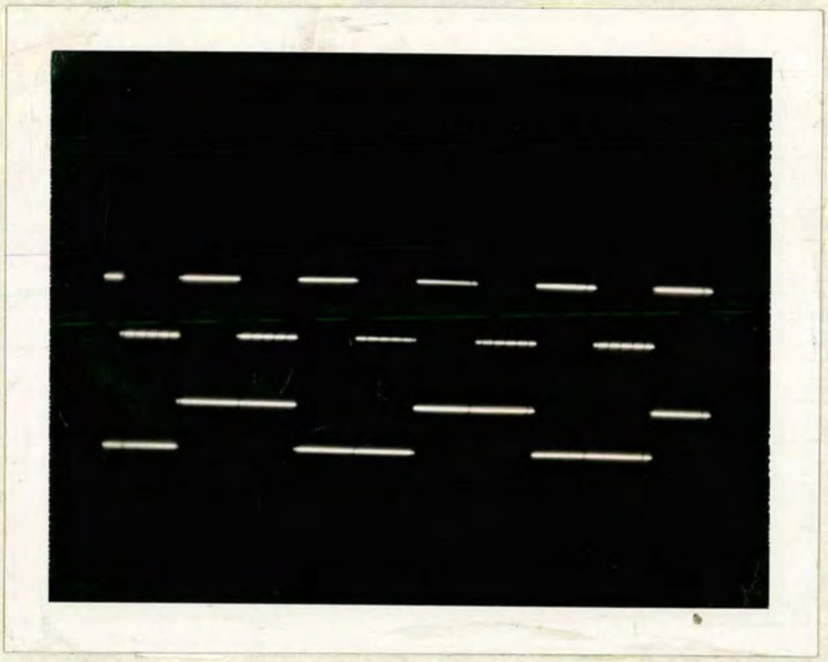


Fig. 5.21 Oscilloscope trace of psuedo-dynamic latch shifting at 1kHz clock rate (top: clock, bottom: output)

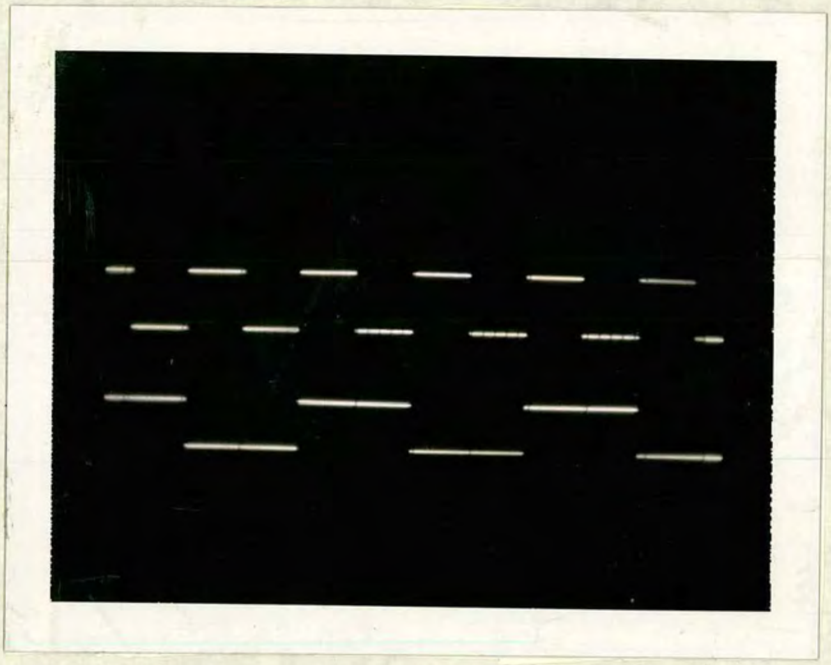


Fig. 5.22 Oscilloscope trace of latch with passive refresh, shifting at 1kHz clock rate (top: clock, bottom: output)

5.5. The GaAs Cell Designs

With the observations discussed above, the D-MESFET pass transistor latch was adopted for general pipelining in the GaAs cells[†], to be described in this section.

A group of cells have been designed and laid out over a period of time and some of them have been modified for a MODFET mask set, rather than a GaAs MESFET mask set as were originally intended. Because of the similarities between MODFET and MESFET, the logic designs were in fact identical and electrical design differences exist mainly in the level shifting in the PRBS and clock generators. These will be highlighted in a later section. However, problems in processing meant that the MODFET-modified cells were eventually re-fabricated in a MESFET process. As a result, there was a mismatch between the submitted designs and the MESFET Schottky diode V_{on} voltage. The V_{on} voltage of MODFET is about 1.1 V, compared to 0.8V of the MESFET process. The effects of this are discussed in Appendix 15 with the test results. Owing to this combination of process related changes, the discussions in this section will mainly concentrate on the cells as designed for the GaAs MESFET technology, unless otherwise stated.

In Chapter 2 (Section 2.2), it was established that a small number of operators are sufficient for a large number of signal processing applications. These include common filter structures such as FIR, and application-specific processors such as wave filters [112]. The GaAs cells that have been designed are summarised by the following logic and I/O operators:

(i) Multiplier;

(ii) Adder;

(iii) 8 bit programmable parallel to serial converter;

(iv) 8 bit programmable serial to parallel converter;

(v) 2-phase clock generator;

(vi) Non-inverting output buffer;

(vii) Non-inverting ECL compatible buffer.

[†] Published work by the author.

(viii) Counter.

First we consider the clock generator design, which is critical to the operation of the pipelining latches in all the operators.

5.5.1. The 2-phase clock generator

The clock generator was designed to have an output swing of ±0.5 V, with minimal phase non-overlap, in order to prevent nodes from losing signals through forward biased gate junctions. A standard cross-coupled NOR structure was used to give the appropriate non-overlap. The schematic of the generator is shown in Fig. 5.23. The inverters preceding the inputs of the NOR gates serve to produce square pulses from the $\phi 0$ clock signal. $\phi 0$ can be a sine wave with a dc offset of +0.35 V, or a square pulse with DCFL voltage levels. The output of the NOR gates are buffered, which then drive level shifters to produce the pull-up and pull-down logic signals for each phase. These DCFL signals are then level shifted to produce the actual driving signals for the push-pull output stage. The design of the level shifter and the output stage merit some discussion.

First consider the push-pull output stage. Noting that the output swing required is nominally ±0.5V, a negative rail of −0.5 V or less is required for the output stage. With a suitable rail, a simple pull-down device, properly sized, can achieve the required LOW clock level.

Producing the HIGH voltage of +0.5 V is more complicated owing to the fact it is not a multiple of a diode drop (relative to Ground). The pull-up device in whatever form has to strictly limit the output voltage to avoid forward biasing the D-MESFET pass transistors. An earlier discussion also identified that the clock generator should be as close to being a voltage source as possible. But as will be seen later on, this is not easy to attain with a MESFET.

Three options of pull-up configuration are available: a D-MESFET device, an E-MESFET, or both devices in parallel with a common gate terminal. When a D-MESFET pull-up is used in parallel or on its own, the output voltage can rise to V_{dd} if it is not forward biased. A source follower output stage can be ruled out because of its poor current drive capability. This leaves push-pull and related output stages where one or more of the pull-up or pull-down devices is off in one output state. Since the HIGH output voltage is +0.5V, voltage limiting cannot be achieved by the pull-up device alone, and the stage driving the pull-up device has to limit its gate voltage, and perform level shifting if necessary. This arrangement is

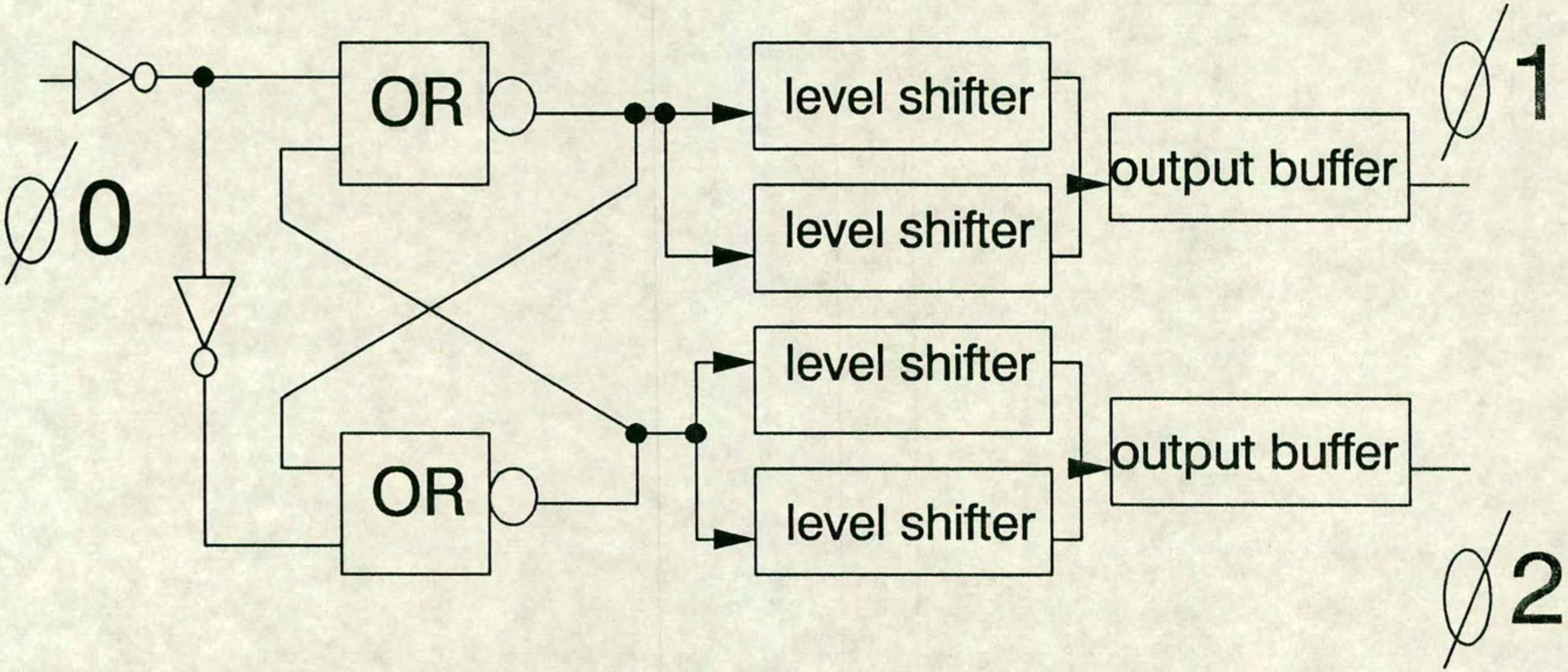


Fig. 5.23 Schematic of the clock generator

shown in Fig. 5.24, and will be referred to in subsequent discussions.

Consider the case of using a single E-MESFET pull-up device. In this case, the difficulty lies in designing an appropriate voltage limiter. The E-MESFET can be chosen to operate in one of two modes: forward biased or non forward biased gate in the quiescent state.

First assume that it is not forward biased. The simplest way to limit the output voltage is to operate it near cutoff by limiting its gate voltage to $+0.5V+V_t$, i.e. $+0.7V$. Since this is close to one diode drop, a simple diode level shifting stage, with a negative V_{ss} rail, can provide the voltage limiting. The total gate current drawn by the D-MESFET pass transistors needs to be taken into account, even though they are not forward biased, and the gate voltage has to be higher than $+0.7V$ to compensate for the IR drop across the channel. The advantages of this are low power and a single negative rail. Unfortunately, it has the same problem of a slow rising output voltage, as in the E-MESFET pass transistor discussed earlier, and a large device is necessary to obtain a sharp clock edge with a large capacitive load. Otherwise, this circuit is simple to use.

Now consider the case where the output pull-up is forward biased in the clock output HIGH state. In this case, a limit of $+0.5V+0.75V$, i.e. $+1.25V$, on the gate voltage is necessary. Since the V_{ds} of the pull-up device is $V_{dd}-(+0.5V)$, about $+1.2V$, the E-MESFET is firmly in saturation, and has an undesirably high output impedance. This would result in high sensitivity of output voltage to the total current drawn from the device, and is clearly not ideal as a voltage source. A low output impedance is possible only in the active region of the MESFET. Unfortunately, with a V_{ds} of $+1.2V$, the E-MESFET cannot possibly be operated in the active region. This would require a gate-source voltage of at least $+1.2V+V_t$, i.e. $+1.4V$ — impossible to achieve with a MESFET. In practice, the size of the E-MESFET can be designed to match the current and voltage requirements for a certain maximum number and sizes of pass transistors being driven, and the total capacitive loading are known. Total capacitive loading, which includes gate and interconnect capacitances, is more difficult to estimate due to its distributed nature. Therefore it is common to use conservative capacitance values for simulations, and the estimated total capacitance loading on a node is unlikely to be accurate. Since this clock driver configuration behaves more like a current source than a voltage source, the quiescent output HIGH voltage can vary significantly from the desired $+0.5$ V in the fabricated circuit. Since

$$i = C_{tot}\frac{dV}{dt}$$

then,

$$V \propto \frac{1}{C_{tot}}$$

where

C_{tot} = total capacitance on a node

V = quiescent voltage on a capacitive node

Therefore a 20% overestimation of C_{tot} would represent an approximate error of −17% in the clock HIGH voltage. This also means that a clock driver designed for a certain "known" load cannot be used without re-sizing for a different load. For instance, a capacitive load 50% less than the designed load would produce an error in quiescent voltage of +100% (of the nominal HIGH of +0.5V), more than sufficient to forward bias the pass transistors.

Another problem with the forward biased MESFET is the need to maintain gate forward bias. This means the pull-down device needs to be partially ON and properly ratioed to the pull-up device, with a DC path from V_{dd} to V_{ss}. Power dissipation is increased, and efficiency is reduced, compared to the push-pull principle.

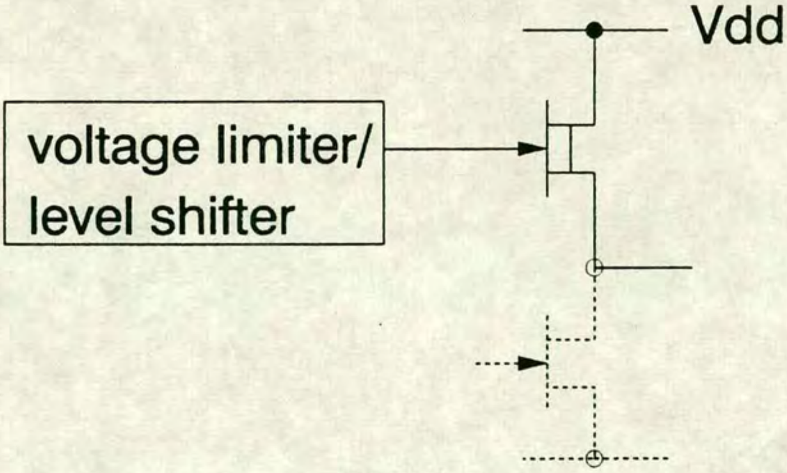


Fig. 5.24 Clock generator output stages arrangement

At the gate voltage limiting stage, the +1.25V limit cannot be met by simple diode shifting, and requires a two stage amplifier which produces the desired amount of attenuation of a (unclamped) DCFL HIGH voltage of V_{dd} (+1.7V).

Now consider the case of using a D-MESFET as the pull-up device. The main drawback in this case is that the voltage limiter now has to go down to at least -0.5 V$-V_t$, i.e. -1 V, in order to turn it OFF. This is because the source voltage of the D-MESFET pull-up is required to go down to -0.5 V, when the output pull-down device is ON. Therefore, a second negative voltage rail of -1 V or less, in addition to the -0.5 V, is necessary. As with the E-MESFET, the D-MESFET can be operated near cut-off or in saturation when the output voltage has stabilised. In addition, it is now possible to operate it in the active region because of its negative V_t.

Suppose that the device is to be forward biased so that it operates in the active region. To maintain forward bias, a large pull-down load has to be used to prevent the output voltage from rising above +0.5V because of the negative V_t. Such a circuit is shown in Fig. 5.25. The disadvantage is that a DC path now exists in both the HIGH and LOW output states, requiring ratioing, and increasing power dissipation. A closer look at the necessary ratioing also reveals that the load device needs to be 6 to 7 times the size of the pull-up device, in order to keep it forward biased — a totally impractical solution since the pull-up device needs to be large to drive to output load. The situation is worse than the E-MESFET because of the negative V_t. With this reasoning, active region operation cannot be easily achieved.

At this point, it is clear that it is difficult to operate a MESFET as a voltage source in the active region. A compromise would be to use the device in the saturation region, i.e. as a voltage limited current source. This is far more practical because of the much wider saturation region, and dissipates less power because the device no longer needs to be forward biased. The drawback is that, as in DCFL logic gates, the saturated output current is now divided by the pass transistors gate currents. The maximum current drive available must be large enough to supply a certain full load condition, and avoid forward biasing the pass transistors if the same circuit is used to drive a small load.

Now consider operating the D-MESFET close to cutoff, when the output voltage is +0.5V. In the cutoff state, its gate voltage must not rise above $+0.5\text{V}+V_t$, i.e. 0V. However, since it also has to supply current to drive the pass transistors, a higher gate voltage is necessary. Since V_{ds} is much larger than $V_{gs}-V_t$, the device is in the saturation region as required. This voltage limiting is achieved by using a level

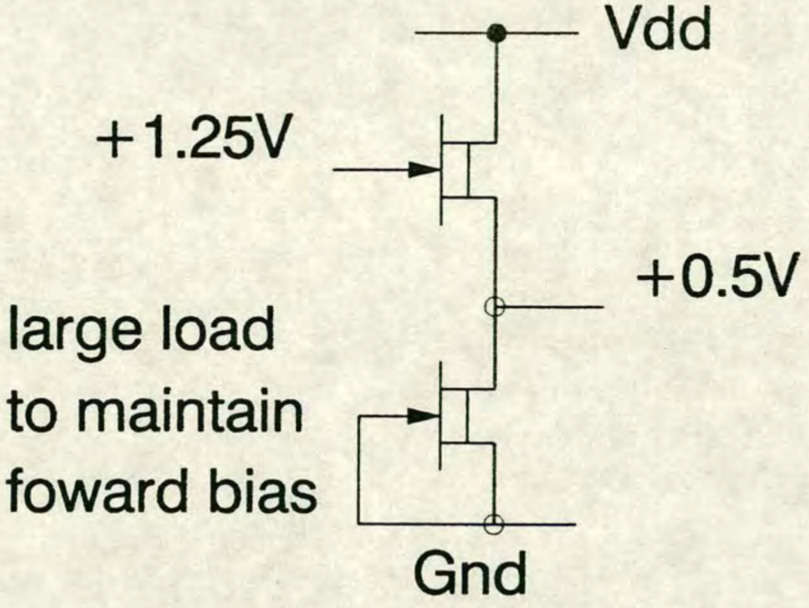


Fig. 5.25 Output driver with a passive pull-down load

shifter with 2 diode drops, as shown in Fig. 5.26. Together with the finite V_{ds} drop across the D-MESFET source follower, this shifter will produce a HIGH level-shifted voltage of +0.1 V, if it is driven by a full swing (+0.1 to 1.7 V) DCFL logic gate. This drives the output D-MESFET device so that it operates as a voltage limited current source as required.

Under low current loading conditions, the push-pull stage can only rise to an absolute maximum of +0.6 V when the pull-up is cutoff, still avoiding forward biasing the D-MESFET pass transistors, with a safety margin of 150mV. If adverse conditions such as V_t variations and noise cause the pass transistors to be forward biased, then the extra current drain on the output device will increase the channel voltage drop and cause the output voltage to fall. In terms of clock rising edge, the D-MESFET is significantly better than the E-MESFET, using the same reasoning as in pass transistors. A smaller gate width can again be used to attain the same speed.

Overall, the single D-MESFET pull-up was chosen for its speed, at the expense of the need for two negative voltage rails. The two diode drops required in the level shifter (Fig. 5.26) also mean a higher dissipation in the clock generator. The resulting output stage and the associated voltage levels are shown in Fig. 5.27. A single D-MESFET pull-up device and an E-MESFET pull-down device, in parallel with a small D-MESFET load were used. Each of the large MESFET devices can be laid out as a number of smaller inter-digitated transistors in parallel. The choice

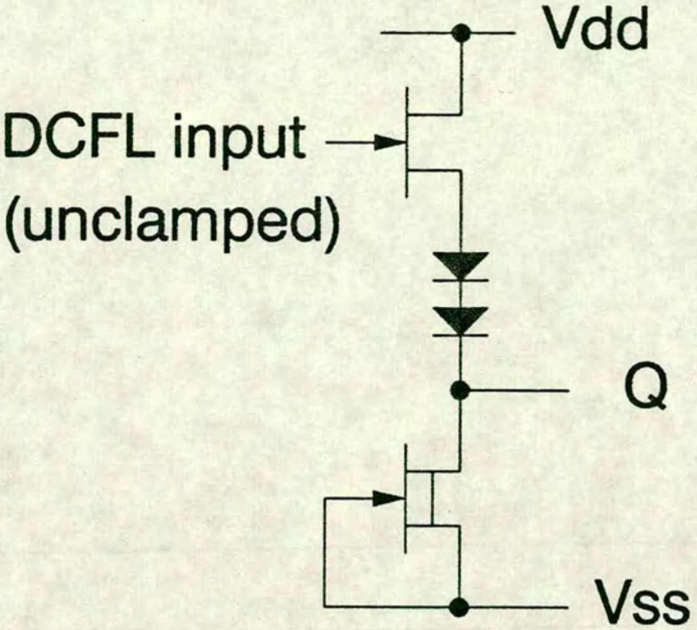


Fig. 5.26 Level shifter in clock generator, with 2 diode drops

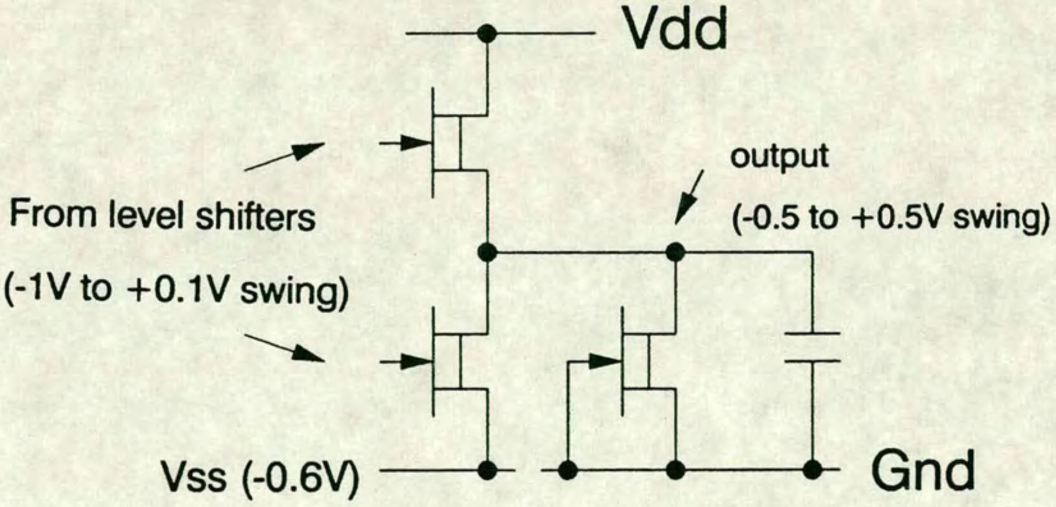


Fig. 5.27 Output buffer circuit of clock generator

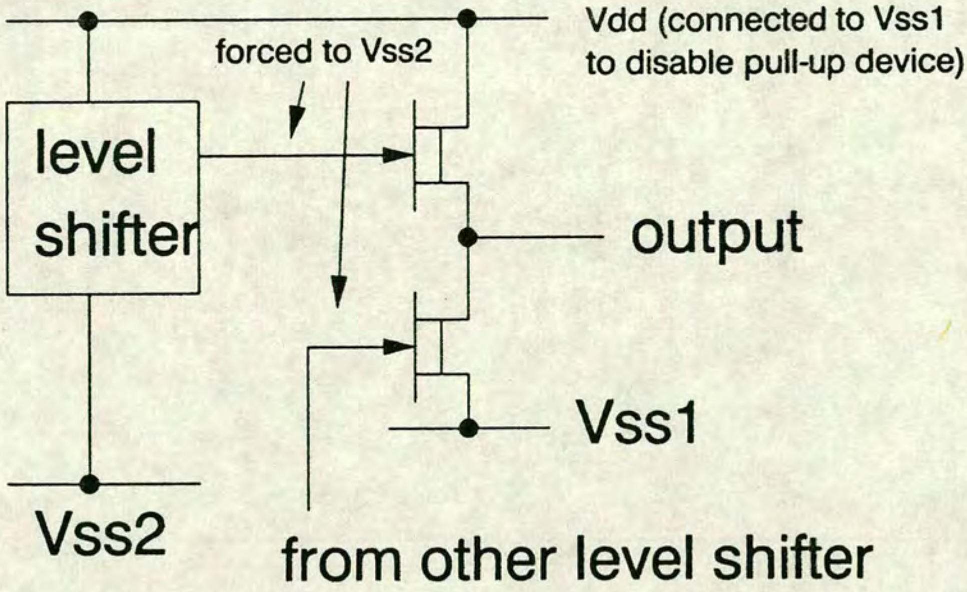


Fig. 5.28 Connections to disable clock generator output stages

of this output stage can be explained as follows.

The most negative rail in the circuit is at −1.7 V (Fig. 5.27). This was chosen such that a small D-MESFET pull-down device could be used in the level shifter stage. This rail presents the most negative backgate potential to other active areas and must be carefully routed. The output stage was designed to drive 3 pF at 500 MHz. The driver sizes had been conservatively designed and the small D-MESFET load to Vss1 was added to reduce the sensitivity of output high level to load conditions as compared to a pure push-pull configuration. The output stage uses a −0.6 V rail to ensure the pass transistors are truly OFF, and to provide extra margin for clock skew, without the danger of excessive non-overlap. The latter is important to avoid loss of latch data due to off time leakage through forward biased inverters. The tight control over clock generation and distribution necessary in a 2-phase non-overlapping system ultimately limits its appeal for technologies faster than GaAs MESFET and MODFET technologies. This problem is looked at later in Chapter 6.

The clock generator has its own independent V_{dd} and Gnd rails for switching isolation. This also lends itself to simple disabling. Fig. 5.28 shows the method of disabling the output stages by switching off the output devices, producing a high impedance state. This allows the clock lines to be connected to external clock signals.

Finally, a simulation of the clock generator at 600MHz is shown in Fig. 5.29. The input is a $\phi 0$ sine wave with a DC offset. The SPICE input file is given in Appendix 4.

5.5.2. The bit-serial adder

The function of the bit serial adder was to add two time aligned bit-serial data streams. An option is also available to use the same cell for subtraction. There is no provision for output overflow and therefore pre-scaling of the input words may be necessary.

There are a number of possible implementations of the full adder [21,90,123]. A full comparison of all these circuits is beyond the scope of this work. The need to take into account low fanout capability and to use only NOR gates and inverters in GaAs design (Chapter 3) has to be weighed against the objective of having a short critical path in the adder circuitry. If the critical path is made up of more than 5 gate delays, then two pipeline stages, instead of one, will be required, in order to partition the logic for the 500MHz nominal clock rate. This introduces an extra cycle or one bit of latency into the adder cell— a severe penalty. For this reason, separate sum and carry blocks are used for their speed advantages. These are shown in Figs. 5.30 and 5.31. The extra loading on the input signals of these blocks are not severe owing to the simplicity of the logic circuits, which operate on the signals one bit at a time — an inherent advantage of bit-serial circuits.

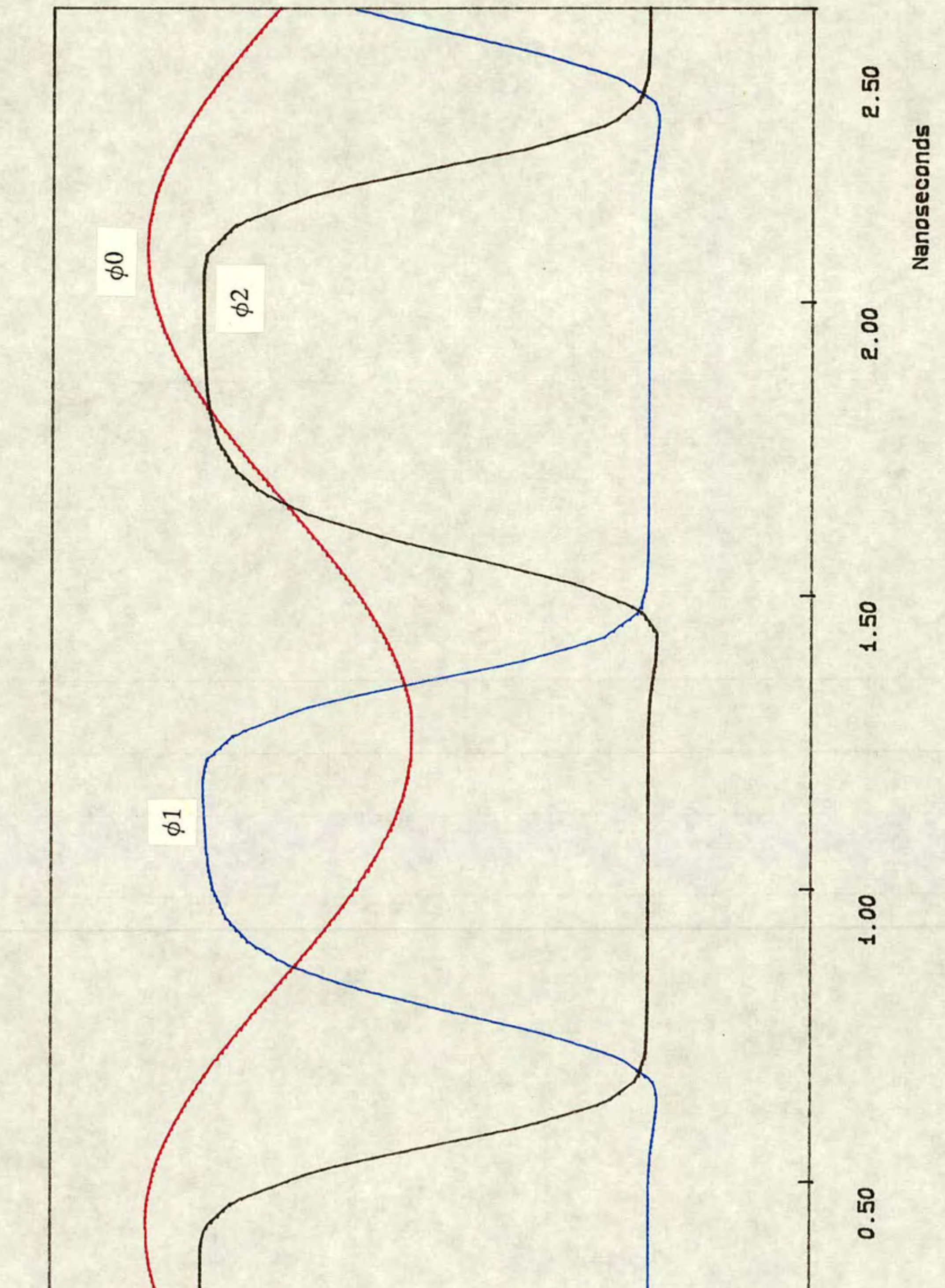


Fig. 5.29 Simulation result of 2-phase clock generator at 600MHz

Pipelining is used extensively to partition the delays into small sections and the largest number of gate delays with this design is 3 propagation delays between 2 pipeline stages, excluding the pass transistor delay. This is equivalent to a total worst case delay of about 700ps, adequate for the nominal clock rate of 500MHz. The block diagram of the complete bit-serial adder is shown in Fig. 5.32. For flexibility, a selector (XNOR gate) is included so that the adder can also be used as a subtractor. In the case of subtraction, the output is (A-B), where the signals A and B are as shown in the diagram. The latency of the adder/subtractor is 1 clock cycle. Since no over-flow control was included in this design, and since a fixed point format was adopted, the filter designer must pre-scale the data signals entering the bit-serial system prior to processing, by programming the data word position within the system word. The necessary amount of pre-scaling can be calculated using a number of filter design programs available [112]. The pre-scaling depends on the number of adders and multipliers present in the system. Since a fractional data format was adopted in this architecture, multiplication always scales down the modulus of the system data word, opposing the possible scaling up effects of addition and subtraction. The actual placement of the data word is performed by the parallel to serial converter (P/S converter), described in a later section.

The adder/subtractor circuit also forms the core of the multiplier, and its simulation result is therefore not presented here.

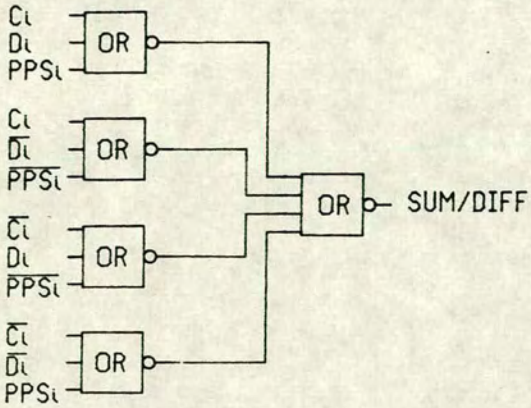


Fig. 5.30 Sum/difference logic schematic

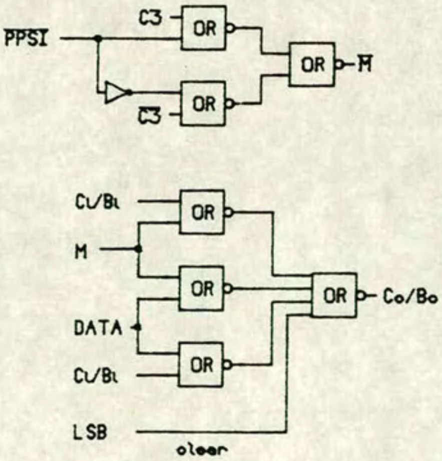


Fig. 5.31 Carry/borrow logic schematic

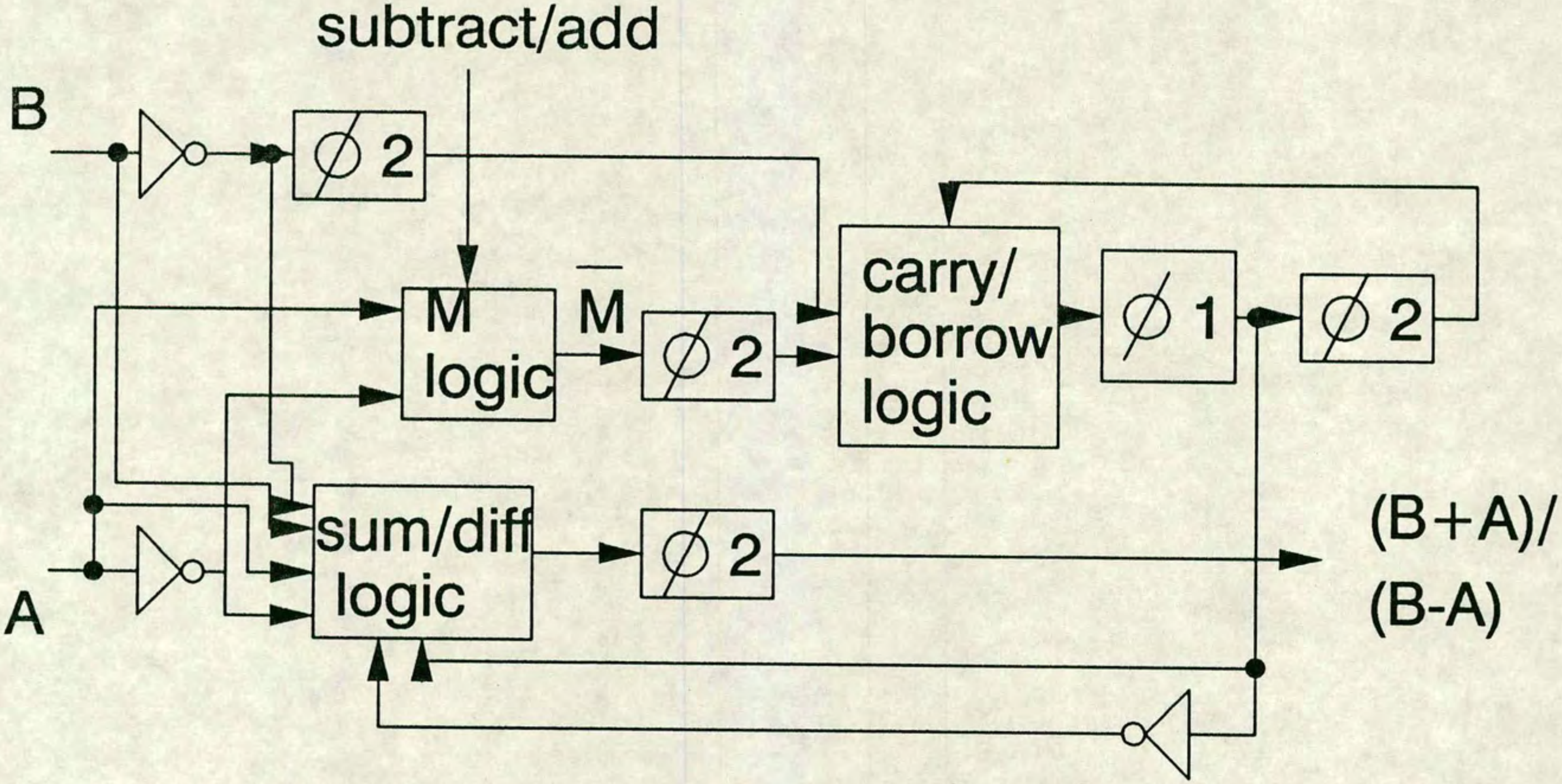


Fig. 5.32 The Bit-Serial Adder Block Diagram

5.5.3. The Multiplier

The latency of a digital filter largely depends on the latency of any multiplier it may contain. Normally, it is best to reduce the latency of a bit-serial filter (reduced group delay), and therefore efforts should be made to reduce the latency of the multiplier as far as circuit area and complexity will allow. The Modified Booth's Algorithm (MBA) is particularly suitable in this respect because it has a smaller latency than the basic carry-save add-shift algorithm [123], and requires a smaller number of adders overall.

The MBA is based on a 5-level encoding of the coefficient word [77]. Each coefficient word is divided into triplets of bits (3 consecutive bits), with the lsb of each triplet overlapping with the msb of the adjacent triplet. Each triplet is recoded to perform one of the 5 operations ($\times 0$, $\times +1$, $\times -1$, $\times 2$, $\times -2$) on the data word. The data word, to be multiplied with the coefficient, is normally called the multiplicand, and is not recoded in any way in the MBA. Therefore, the size of the complete multiplier is dependent only on the coefficient word length.

To implement the MBA, we define a multiplier slice as a circuit block which performs the MBA multiplication between one triplet of coefficient bits and the data word. Fig. 5.33 shows the recoding grouping and the corresponding multiplier slice mapping. With this scheme, the number of slices for an N bit coefficient is equal to N/2 (assuming that N is even).

In order to understand the implementation of the MBA multiplier in GaAs, it is necessary to detail the MBA recoding scheme. An example multiplication is also given to illustrate the MBA in operation. A simple mathematical treatise of the MBA is available in Delhaye [23], in which a 4×4 bit-parallel GaAs multiplier was implemented. The 5-level recoding scheme is given in Table 5.1 below.

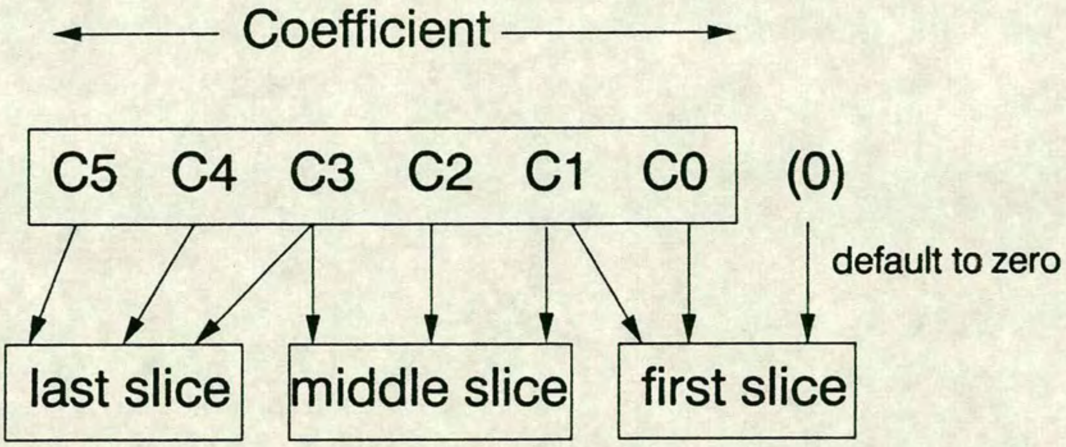


Fig. 5.33 Modified Booth's algorithm coefficient grouping for recoding

coeff. segment c3	 c2	 c1	operation
0	0	0	(1/4) ppsi + (data bit × 0)
0	0	1	(1/4) ppsi + (data bit × 1)
0	1	0	(1/4) ppsi + (data bit × 1)
0	1	1	(1/4) ppsi + (data bit × 2)
1	0	0	(1/4) ppsi - (data bit × 2)
1	0	1	(1/4) ppsi - (data bit × 1)
1	1	0	(1/4) ppsi - (data bit × 1)
1	1	1	(1/4) ppsi + (data bit × 0)

Table 5.1: The MBA recoding scheme

In this Table, "ppsi" represents the partial product sum input from the adjacent, less significant coefficient multiplier slice. Therefore each row represents an addition or subtraction of the the data or (data×2) word with or from the partial product sum of the previous slice. An addition or subtraction on (data bit×2) is equivalent to operating on the adjacent, less significant bit in the data word, i.e. an arithmetic left shift. So all ×2 entries can be implemented with one bit delays, and then selecting between the delayed and the normal data stream. Overall, the implementation of this algorithm consists of the adder/subtractor described in the previous section, delay elements, and recoding logic circuitry to select addition or addition, and data or (data×2).

To see how a bit-serial MBA multiplier works, Table 5.2 below gives an example multiplication between these data and coefficient words:

coefficient = 1.00110

data = 00.0111

The notations in the following Table 5.2 are:

(i) $ppsi_n$=partial product sum input of the nth multiplier slice

(ii) $ppso_n$=partial product sum output of the nth multiplier slice

In this example, the 6 bit coefficient word requires a multiplier with 3 slices. For simplicity of discussion, these slices are denoted by "first-slice" for the least significant triplet, "middle-slice" for the centre triplet, and "last-slice" for the most significant triplet. The final output product is $ppso_3$ truncated by 1 bit, i.e. 11.1010, assuming a fixed word length of 6 bits. It is important to note that the coefficient has **one** sign bit, as opposed to the two bits in the data word. The need for two bits in the data word will becomes clear later on. This example shows the MBA multiplication as a parallel process, the results of which should be implicitly interpreted to appear bit-serially in the actual circuit implementation. Nevertheless, it serves to illustrate a number of important properties of the algorithm that needs to be implemented in the design:

coeff. word			1 0	0 1	1 0
data word			0 0	0 1	1 1
coeff. group	symbol	operation	result		
slice 1:	ppsi$_1$	× 0	0 0 0	0 0	0 0
coeff. triplet=100	data×2	left shift	0 0 0	1 1	1 0
(least significant)	ppso$_1$	ppsi$_1$−data×2	1 1 1 1 1	1 1	1 0
slice 2:	ppsi$_2$	=ppso$_1$	1 1 1 1 1	1 1	
coeff. triplet=011	data×2	left shift	0 0 0 1 1	1 0	
(centre bits)	ppso$_2$	ppsi$_2$+data×2	0 0 0 0 0 1 0	1 0	
slice 3:	ppsi$_3$	=ppso$_2$	0 0 0 0 0 1 0		
coeff. triplet=100	data×2	left shift	0 0 0 1 1 1 0		
(most significant)	ppso$_3$	ppsi$_3$−data×2	1 1 1 0 1 0 0		

Table 5.2 : Example multiplication using the modified Booth's algorithm

(i) The lsb of the least significant coefficient triplet is always 0.

(ii) The partial product sum input (ppsi$_1$) of the first-slice is always 0.

(iii) Between two slices, the data word is shifted left by 2 bits (×4) before being used to form the next ppso. In conventional uncoded multiplication, the data is shifted only by one bit. In the MBA, each slice caters for effectively 2 bits of the coefficient word (3 bits with 1 bit overlap), hence the double shifting.

(iv) In each slice, the carry/borrow for the lsb addition/subtraction is always 0.

(v) In each slice, the partial product sum output (ppso$_n$) needs to be sign extended by 2 bits to accommodate the 2 bit shifted data word so that they have the same word length for addition/subtraction in the next slice.

(vi) The ppso output of the last slice is sign extended by 1 bit only.

These properties result in the bulk of extra control logic, in addition to the adder/subtractor logic, that are necessary to form the MBA multiplier slice. Also, the first, middle, and last slices are slightly different. Delay elements are also required to time align the data word and the partial product sum outputs between the slices. The number of delays per slice is equal to the latency of each slice. More details of the control logic that implement these properties of MBA listed above are described in the next sub-section.

The multiplier slice design

There are 3 inputs signals to each slice: data, partial product sum, and the coefficient. The first two signals are bit-serial by default, and constantly vary from word to word. With the coefficient word, choices include either fixed or adaptive coefficients, and bit serial or parallel coefficient loading, such as from an on-chip memory [123]. In this version, the coefficient word is entered in parallel, and has only one sign bit for bit efficiency (unlike the data word which requires 2 sign bits). The coefficient word is fixed for a certain application. A bit-serial multiplier with parallel coefficient loading is often called a serial-parallel multiplier (not to be confused with a serial-parallel converter, discussed later on).

The next major design issue is the total latency of the multiplier, which should be minimised as far as possible. A serial-parallel multiplier has a smaller latency because the recoding of the coefficient can be achieved in parallel. In this version of the MBA multiplier, the latency is given by (1.5M-1), where M is the coefficient word length. In a serial-serial multiplier where both the data and coefficient words are entered serially, the latency is larger (= 1.5M+2) [21].

Consider the first of the m/2 slices (the first-slice). From earlier discussions, the one-bit full adder/subtractor has a latency of 1 clock cycle, and is also the basic latency of the multiplier slice. Since the MBA operates on triplets of coefficient bits with a one bit overlap, the data word (or data×2) entering the adjacent middle-slice needs to be shifted by 2 bits before it is added or subtracted to the partial sum output of the first slice. This constitutes two bits of latency. Since addition/subtraction takes one clock cycle, the data word needs to be delayed by a total of 3 clock cycles before entering the next slice, so that they are properly aligned for the next triplet of the coefficient. Therefore the total latency of the first-slice is 3.

This latency figure also applies to the middle-slice, which is identical to the first-slice, except that the partial product sum input is not always 0.

The situation is different in the last-slice. Since the output of the last-slice is not cascaded to another MBA slice, the data word no longer needs to be left shifted by 2 bits for time alignment. Therefore in theory, the latency of the last-slice is only one bit, due to the adder/subtractor. However, the entries which add/subtract data×2 in the MBA (Table. 5.1) must still be accommodated, whether or not the coefficient triplet happens to fall under those 2 entries. In two's complement arithmetic, data×2 is equivalent to increasing the word length by 1 bit and introducing a 0 at the lsb. Since a fixed word length format is adopted here, this 1 bit growth in word length can only be accommodated by discarding the msb of the data word, i.e. one of the two sign bits. After addition/subtraction in the last slice, this lost sign bit needs to be restored so that the output product has the same format as the data words, hence the one bit sign extension. Therefore, together with the unavoidable adder/subtractor delay, the total latency of the last-slice is 2 bits.

From this analysis, the total latency of a serial-parallel multiplier is given by:

$$\text{multiplier latency} = (\frac{m}{2} - 1) \times 3 + 2$$

$$= 1.5m - 1$$

The issues of truncation of the partial products sums follow directly from the discussion on the latency. The 2-bit shifting of the data word between slices (×4) necessitates a 2 bit sign extension on the partial product sum for correct relative positioning in the next slice. This extension automatically truncates the two least significant bits of the next result of the addition/subtraction in the same slice. Thus the total word length remains unchanged. This was illustrated by the example of MBA multiplication in Table 5.2. In the last-slice, the one bit sign extension truncates the lsb of the next addition/subtraction in the last-slice. Therefore overall, for an M bit coefficient, the number of truncated bits is given by:

$$(\frac{m}{2} - 1) \times 2 + 1 = m - 1$$

$$= \text{coefficient word length} - \text{number of sign bits}$$

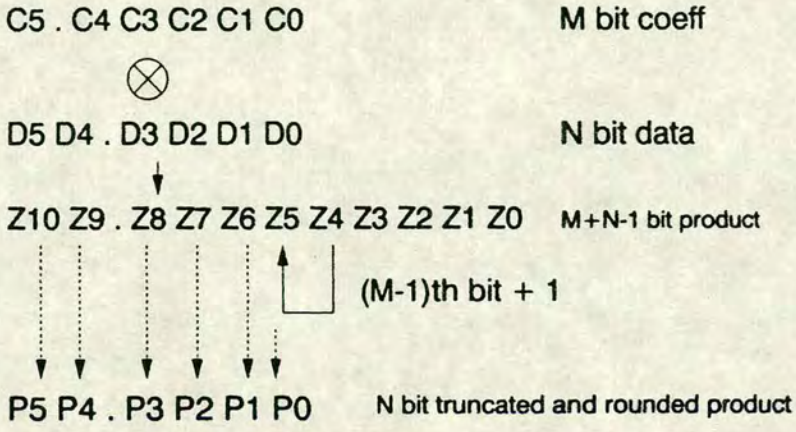


Fig. 5.34 Rounding of the product between an n bit
m bit coefficient

In terms of fractional arithmetic, this equation also illustrate the position of the fractional point in the truncated product. Since the coefficient has one sign bit, the product of data×coefficient grows by $m-1$ bits, which needs to be truncated to maintain word length. The truncated word, as achieved by the sign extensions described above, therefore always has the correct fractional point and number of sign bits.

It follows that if the coefficient has 2 sign bits instead of 1, i.e. same as the data word format, then only $m-2$ bits needs to be truncated — one less than before. If such a coefficient is used, it means that there is no need for sign extension in the last slice of the MBA multiplier at all, and the slice and multiplier latency is reduced by 1. The circuitry in the last-slice is also simplified, at the expense of one redundant sign bit in the coefficient. However, if the coefficient is computed by the bit-serial system adaptively, and therefore has 2 sign bits, then no inefficiency results.

From the above discussion, the data word convention defined in the introduction of this Chapter becomes clear. The necessity for at least 2 sign bits or guard bits is a direct result of the data×2 entries in the MBA (Table 5.1). Since the msb of the data word is discarded, as mentioned earlier, two sign bits are needed to retain its sign. If only one sign bit is present in the original data word, the sign information would be totally lost. The sign extension circuit therefore extends this sign bit as appropriate.

Multiplier product rounding

So far in this discussion, one or two bits of the partial product sum are truncated in each slice without regard to their binary value. Truncation errors should normally be reduced as far as possible, and rounding becomes necessary.

Consider the issue of implementing rounding in the Modified Booth's Algorithm multiplier. When an n bit data word is multiplied with an m bit coefficient (with one sign bit), the total word length of the product is given by $(n+m-1)$, without truncation. If the lsb is denoted as bit 1, then the rounding of this product is equivalent to adding binary 1 to the $(m-1)$th bit of the product. This is illustrated in Fig. 5.34. Since $m-1$ bits are truncated in the multiplier, this addition rounds the most significant of the truncated bits into the lsb of the final product.

To implement this in practice, the LSB pulse can be used to advantage. Since the LSB pulse is normally coincident with bit 1 of the original data input word into the first-slice of the multiplier, it can be delayed by the appropriate number of clock cycles to provide the binary 1 needed for the rounding. Since the rounding position is the $(m-1)$th bit, then $(m-1-1)$ delays on the LSB signal are necessary. This signal is readily available because the LSB signal is delayed by 3 cycles (as is the data signal) between two slices in the multiplier. An appropriate tap on one of the delayed LSB signals would provide the rounding binary 1. This is then simply fed back into the partial product sum input of the first-slice to achieve the rounding operation.

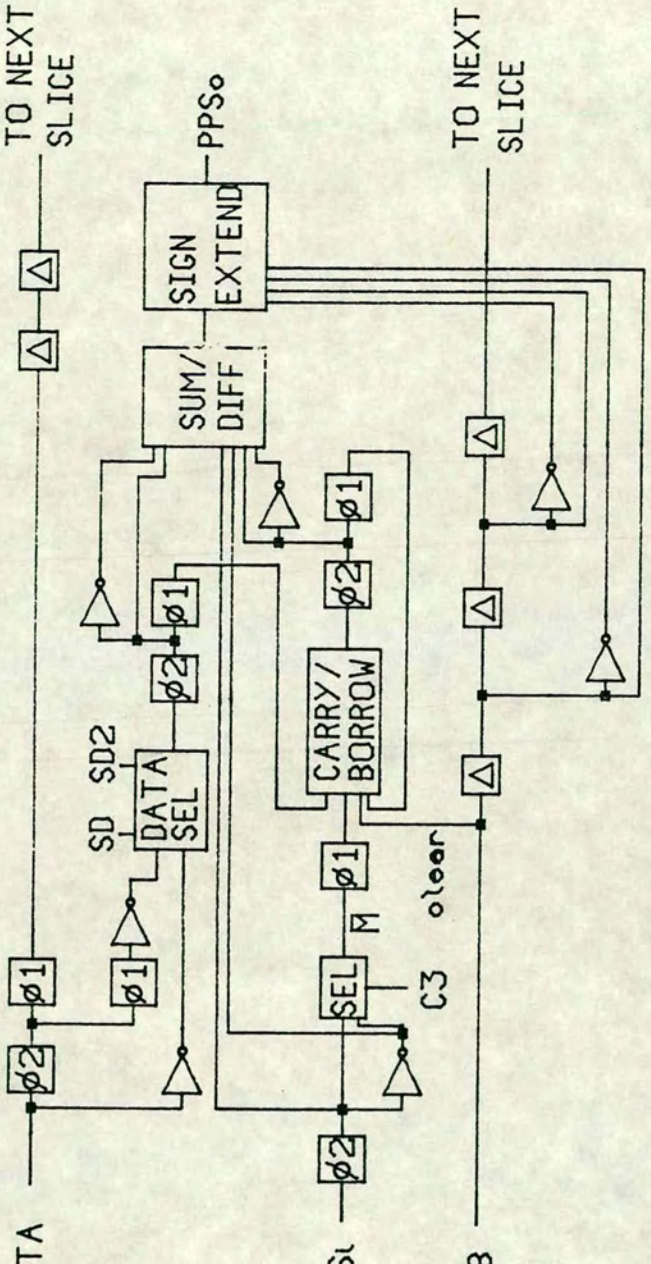


Fig. 5.35 Block diagram of middle-slice of MBA multiplier

Multiplier slice logic schematics

Based on the properties of the MBA multiplier listed earlier, the logic design can be readily worked out. Since the adder/subtractor design forms the core of the multiplier, this section only summarises the recoding and sign extension logic required at the interface between the adder/subtractors in each slice. Again, the logic is pipelined appropriately for the nominal clock rate of 500MHz.

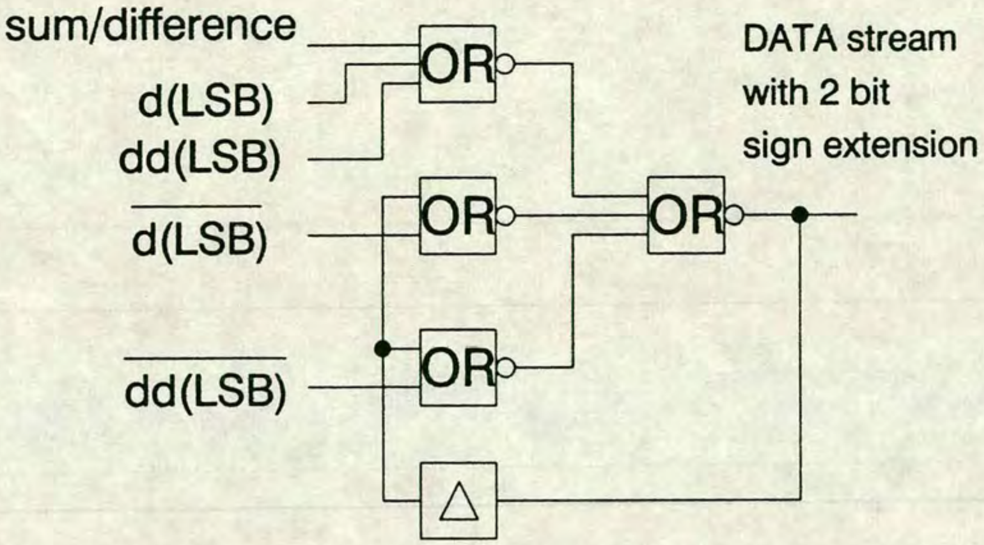


Fig. 5.36 Two bit sign extension schematic

The block diagram of a middle-slice is shown in Fig. 5.35. The sum/difference and the carry/borrow blocks are as in the adder but with a simple but important change: the sum/difference block used in the adder/subtractor (Fig. 5.30 and 5.31) is modified to produce its $\overline{\text{sum/difference}}$. This is simply achieved by replacing each logic input signal with its complement. This is necessary because in the multiplier slice, in Fig. 5.35, the pipelining arrangement is such that the $\overline{\text{sum/difference}}$ logic signal is required. The normal pseudo-dynamic latch is inverting and signal polarities have to be matched to this pipelining arrangement.

The schematic of the sign extend circuits for the middle-slice (two bit extension) and last-slice (one bit extension) are shown in Fig. 5.36 and 5.37 respectively. The

LSB signal, together with its 1 bit delayed version, are used to select either the previous sign bit (sign extend), or to use the current partial product sum output bit, as the output of the multiplier slice that is to be cascaded to the next slice.

The recoding logic block produces two output signals to select either the DATA or the 1 bit delayed DATA (data×2) signal for the sum/difference block, according to the MBA scheme in Table 5.1. An XNOR gate is used to achieve this selection. The recoding and data selection logic schematics are shown in Figs. 5.38 and 5.39.

Finally, two simulations of a middle multiplier slice is shown in Fig. 5.40 & 5.41. The input data and the correct output word are given under the Figures. The signals are also labelled. In both cases, the coefficient triplet for the slice is given by:

coefficient triplet = 100

The coefficient triplet of 100 represents the following calculation, according to the MBA (Table 5.1):

$$\mathrm{PPSO} = \mathrm{PPSI} - (2\times \mathrm{DATA})$$

The inherent one bit latency of the adder/subtractor can be seen from the simulation results as one clock cycle delay between PPSO and DATA. Again, note that all results appear lsb first, and should be read accordingly. The sign extension process (and hence truncation) was not shown for simplicity. The two simulations demonstrated the correct operation of the multiplier slice, as well as that of the adder/subtractor circuit discussed earlier. A simulation of the slice, with variations in the MESFET threshold voltages of +100mV, is shown in Fig. 5.42. Correct operation was still maintained with such a large change in device parameters, in excess of 2 times the standard deviations of the V_t voltages. The multiplier slice contained approximately 246 MESFETs, and dissipated a DC power of 41.7mW in the simulations. The SPICE input file is given in Appendix 5.

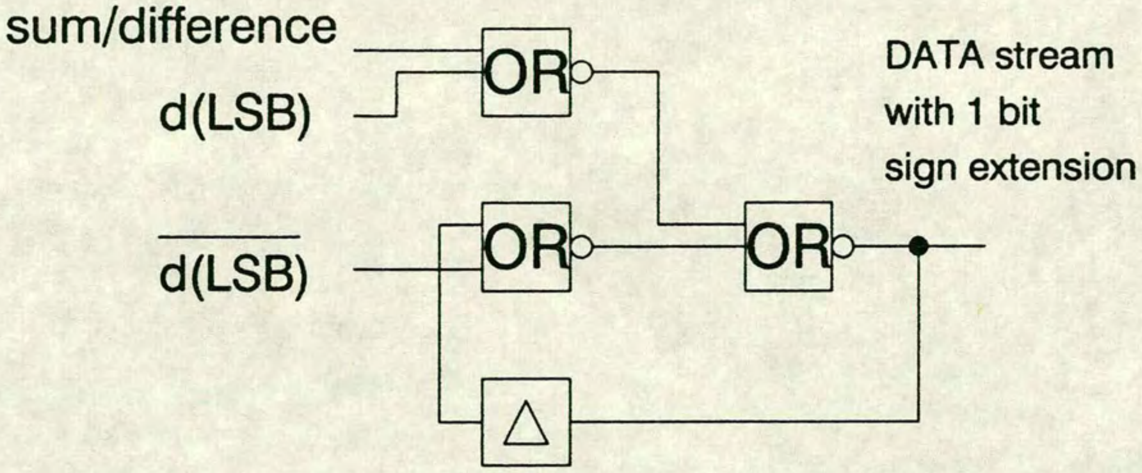


Fig. 5.37 One bit sign extension schematic for last-slice

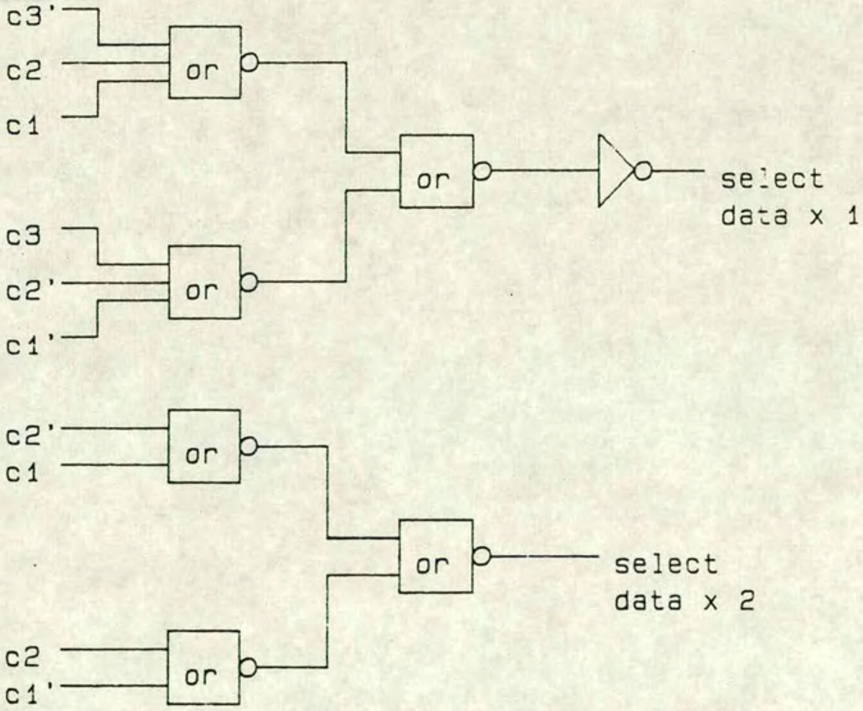


Fig. 5.38 Recoding logic schematic

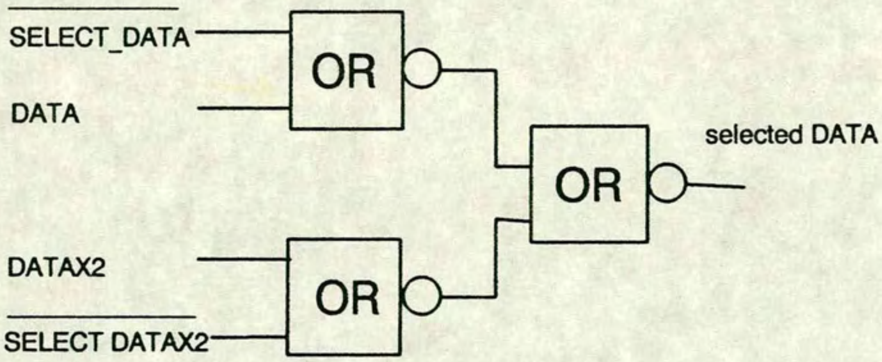

$\overline{\text{SELECT\_DATA}}$
DATA
OR
OR
selected DATA
DATAX2
OR
$\overline{\text{SELECT DATAX2}}$

Fig. 5.39 Data/data×2 selection logic

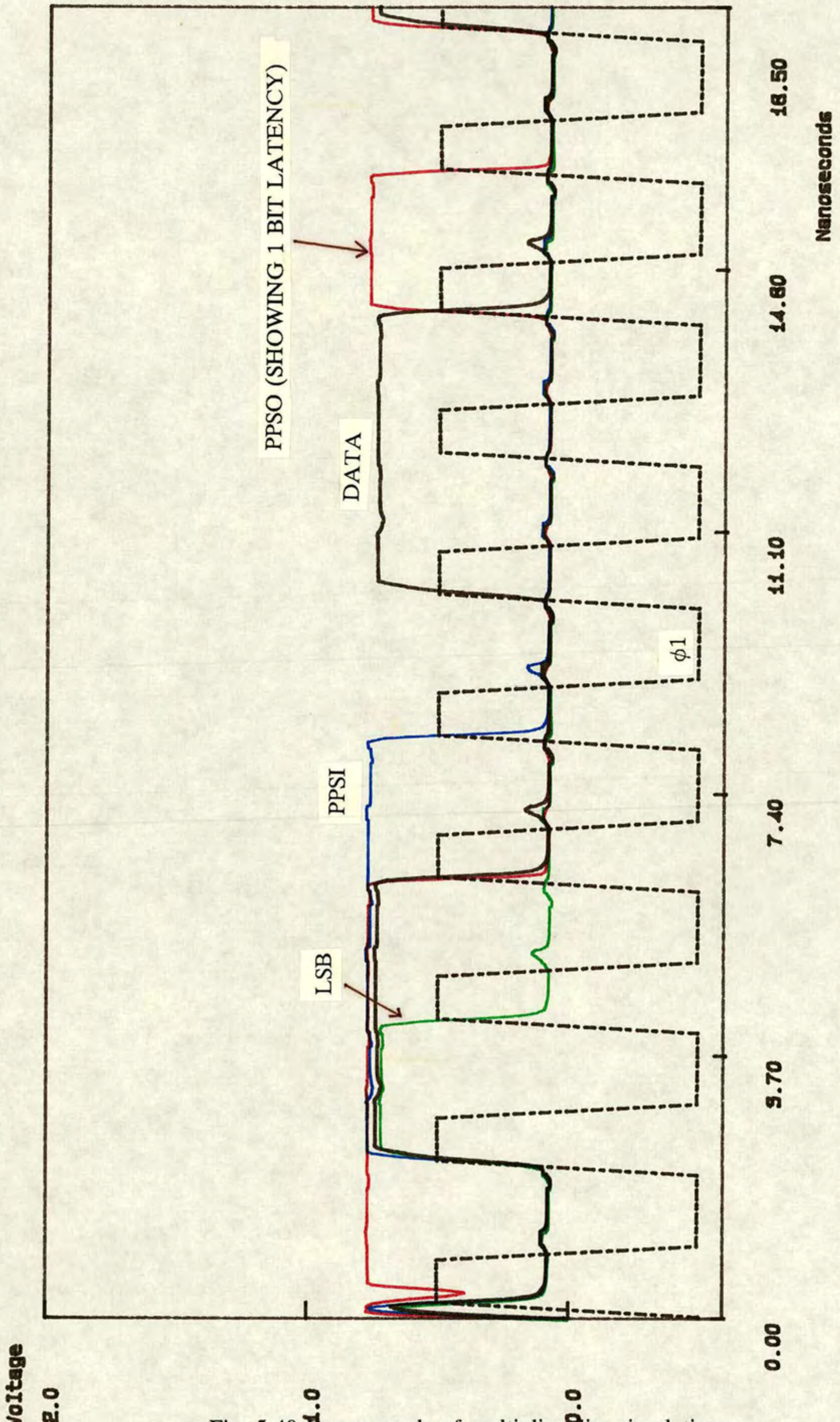


Fig. 5.40 One example of multiplier slice simulation:
DATA=110011, PPSI=000111, OUTPUT=10100001.

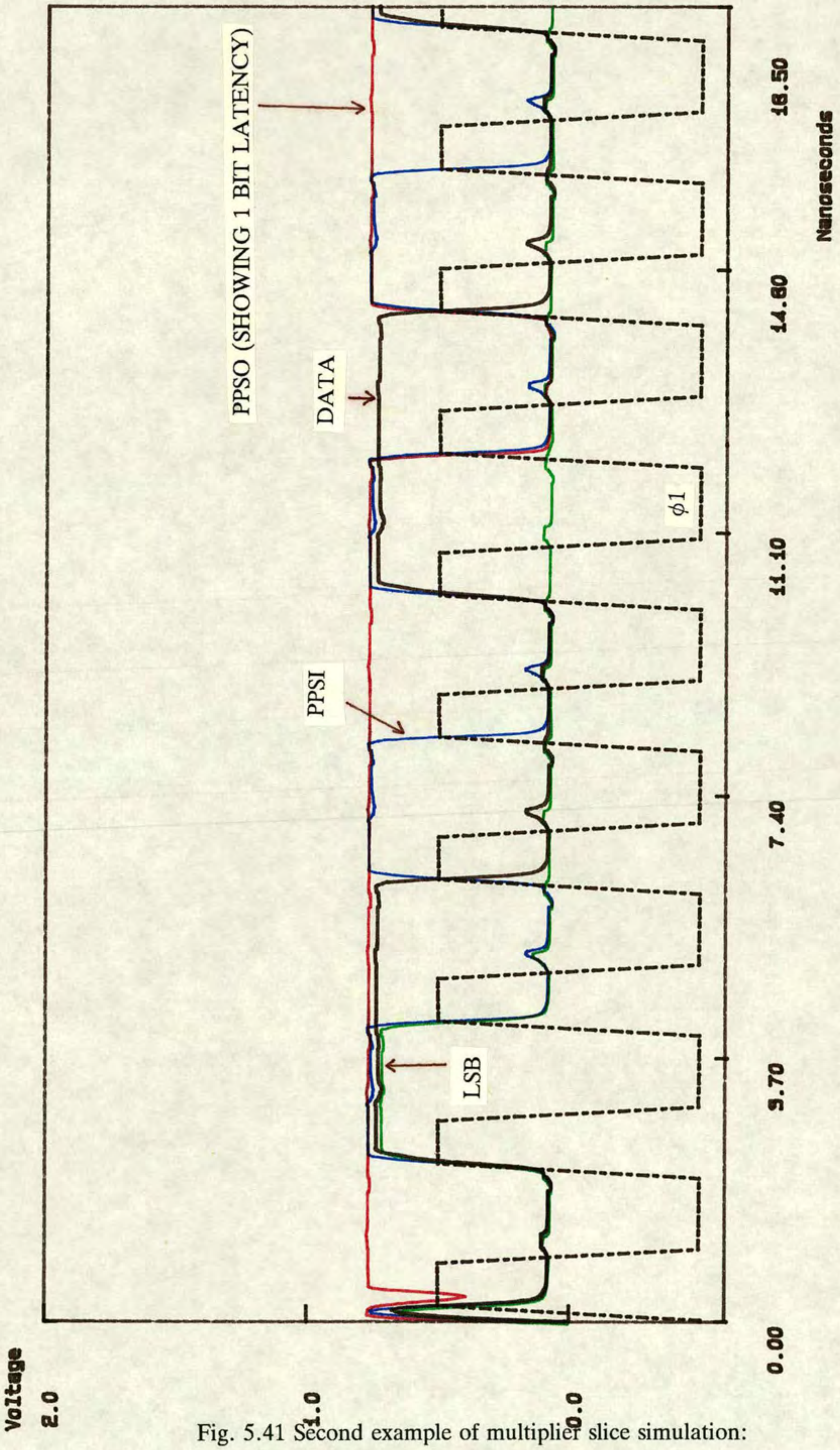


Fig. 5.41 Second example of multiplier slice simulation: DATA=110011, PPSI=010101, OUTPUT=11101111.

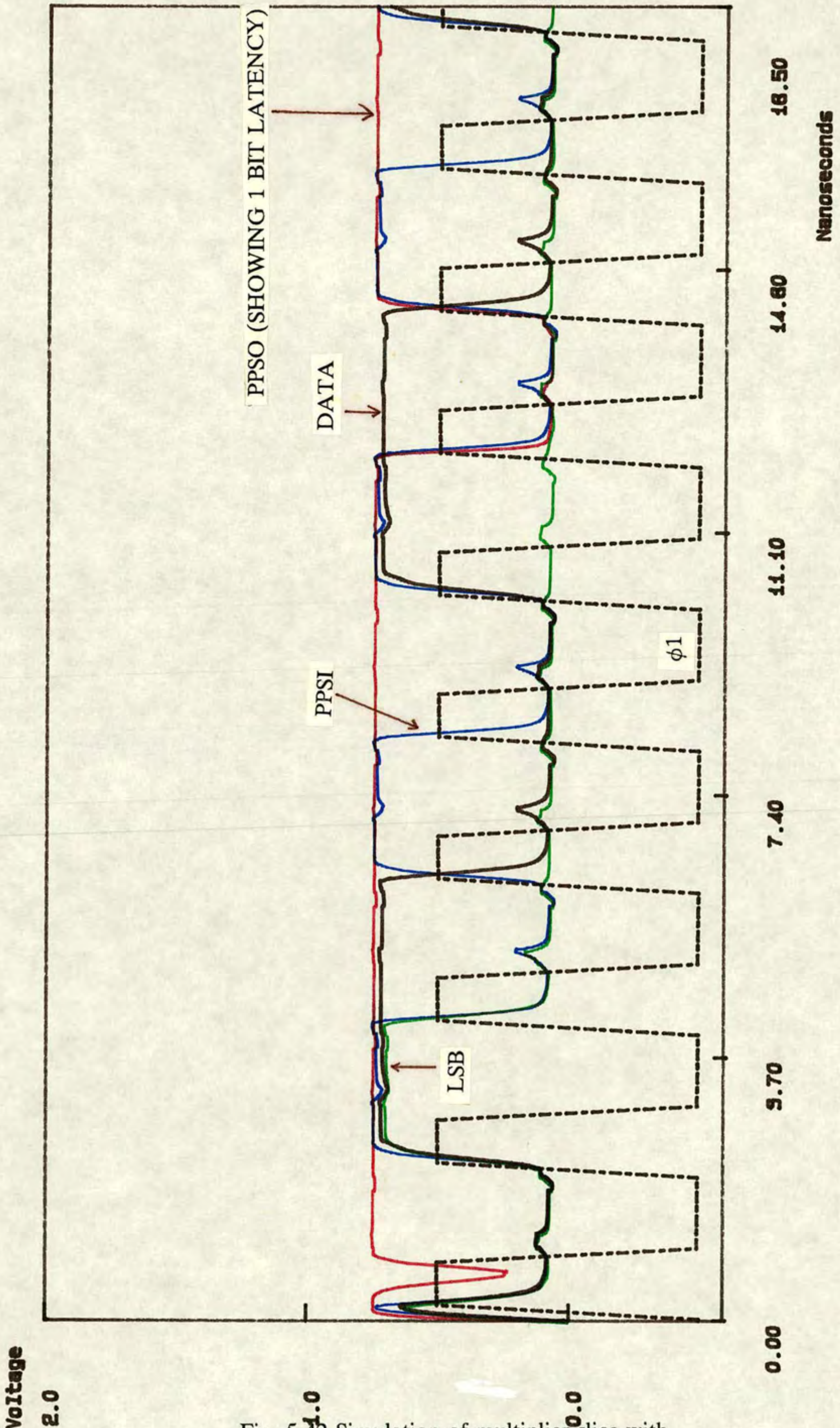


Fig. 5.42 Simulation of multiplier slice with

100mV change in V_t

5.5.4. The Parallel to Serial Converter (P/S converter)

The parallel to serial converter converts an 8 bit parallel word into a serial word according to the adopted data format. The converter allows a parallel data word, such as that from an analogue-to-digital converter, to be placed at any pre-programmed position within a 9 to 31 bit system word. The minimum system word length of 9 bits provides for the 2 guard bits required for the multiplier to function correctly. A shorter system word length is allowed by this converter (down to a minimum of 3 bits), but such short word lengths are impractical in real filters.

Two 5 stage pseudo-random binary sequence (PRBS) generators are used as counters to programme the data word position. PRBS are maximum length codes and are therefore efficient in terms of area usage. However, due to their pseudo-random count progression, they are not the natural choice for counters. PRBS generators are commonly implemented by cascading a number of shift registers, with appropriate taps from the outputs of the shift registers, feed through an exclusive-NOR (XNOR) or exclusive-OR (XOR) gate. An example of the former (with an XNOR feedback) is shown in Fig. 5.43. If n shift registers are used, then the maximum number of distinct n bit code is 2^n-1, as opposed to 2^n in an n bit binary counter. The one "missing" code is normally the all HIGH state, i.e. (111...), and is called the lockout state. Since the XNOR of two logic 1's is also 1, the shift registers outputs remain unchanged and will not generate the PRBS. This state must be taken into account in the design.

In order to use PRBS sequence generators as counters, the registers are loaded at the beginning of each counting cycle with a programmable starting code word, and then allowed to go through the PRBS sequence until a pre-determined "endpoint" (EP) is reached. The endpoint is chosen so that the logic required to detect it is as simple as possible. The programmable starting point determines the length of the count, and can be any code within the $2^n - 1$ combinations, including the EP itself. Therefore, the fact that the progression of such a PRBS counter is pseudo-random is not a problem. The user is given a table to choose the appropriate starting code for all allowable count lengths.

Compared to other common counters such as binary and Johnson counters, the PRBS counter has a number of advantages. Owing to the simple single logic gate endpoint detection, the counter is free of ripple delays. The small component count is particularly suitable to GaAs with its low fanout capability. Word length programming is also simple. One disadvantage is that the maximum count is one less than that of a binary counter for the same number of shift registers. This is not detrimental if the correct number of stages are used. A 5 stage PRBS counter is used for this application, which allows a maximum system word length of 31 bits, more than adequate for most filters. The other disadvantage is the lock-out state of 11111. This can be simply solved by choosing a suitable endpoint, as will be described below.

The two 5-stage PRBS counters used in the P/S converter are denoted by PRBS1 and PRBS2. Fig. 5.44 shows the block diagram of the converter. Apart from the counters, an 8 bit shift register block stores the new parallel-loaded data word and produces the serial output data for the bit-serial system. Control logic, again pipelined where possible, generates signals from the two counters. The appropriate presets for all allowable data positions and system word lengths are summarised in Table 5.3.

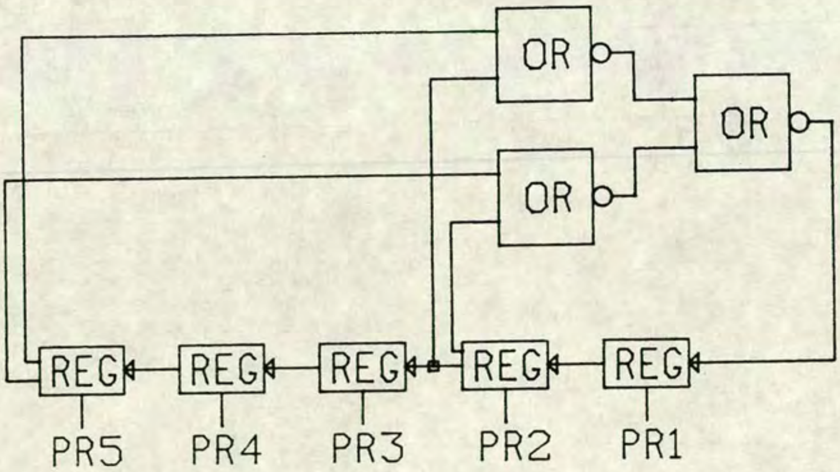


Fig. 5.43 A 5-stage PRBS generator

PRBS register preset word.					P/S converter		S/P converter
Direction of shifting: Q1→Q5					System word length	Number of leading 0's in system word	msb position of data word within system word
Q1	Q2	Q3	Q4	Q5			
0	1	1	1	1	31	17	—
1	0	1	1	1	30	16	31
0	1	0	1	1	29	15	30
1	0	1	0	1	28	14	29
0	1	0	1	0	27	13	28
0	0	1	0	1	26	12	27
0	0	0	1	0	25	11	26
1	0	0	0	1	24	10	25
0	1	0	0	0	23	9	24
0	0	1	0	0	22	8	23
1	0	0	1	0	21	7	22
1	1	0	0	1	20	6	21
1	1	1	0	0	19	5	20
0	1	1	1	0	18	4	19
0	0	1	1	1	17	3	18
0	0	0	1	1	16	2	17
0	0	0	0	1	15	1	16

Table 5.3: Register presets for the PRBS counters
in the P/S and S/P converters

PRBS register preset word.					P/S converter		S/P converter
Direction of shifting: Q1→Q5					System word length	Number of leading 0's in system word	msb position of data word within system word
Q1	Q2	Q3	Q4	Q5			
0	0	0	0	0	14	0	15
1	0	0	0	0	13	30	14
1	1	0	0	0	12	29	13
0	1	1	0	0	11	28	12
0	0	1	1	0	10	27	11
1	0	0	1	1	9	26	10
0	1	0	0	1	8	25	9
1	0	1	0	0	7	24	8
1	1	0	1	0	6	23	7
0	1	1	0	1	5	22	6
1	0	1	1	0	4	21	5
1	1	0	1	1	3	20	4
1	1	1	0	1	2	19	3
1	1	1	1	0	—	18	—

Table 5.3 (contd.)

For convenience of discussion, the bit-serial signal format is shown again in Fig. 5.45. PRBS1 counts the whole system word length and controls the sampling rate of the input parallel word. It also produces the signals to load the registers in PRBS2 and itself with their pre-selected parallel starting points, and to enable/disable the shifting of the input data word in the data registers.

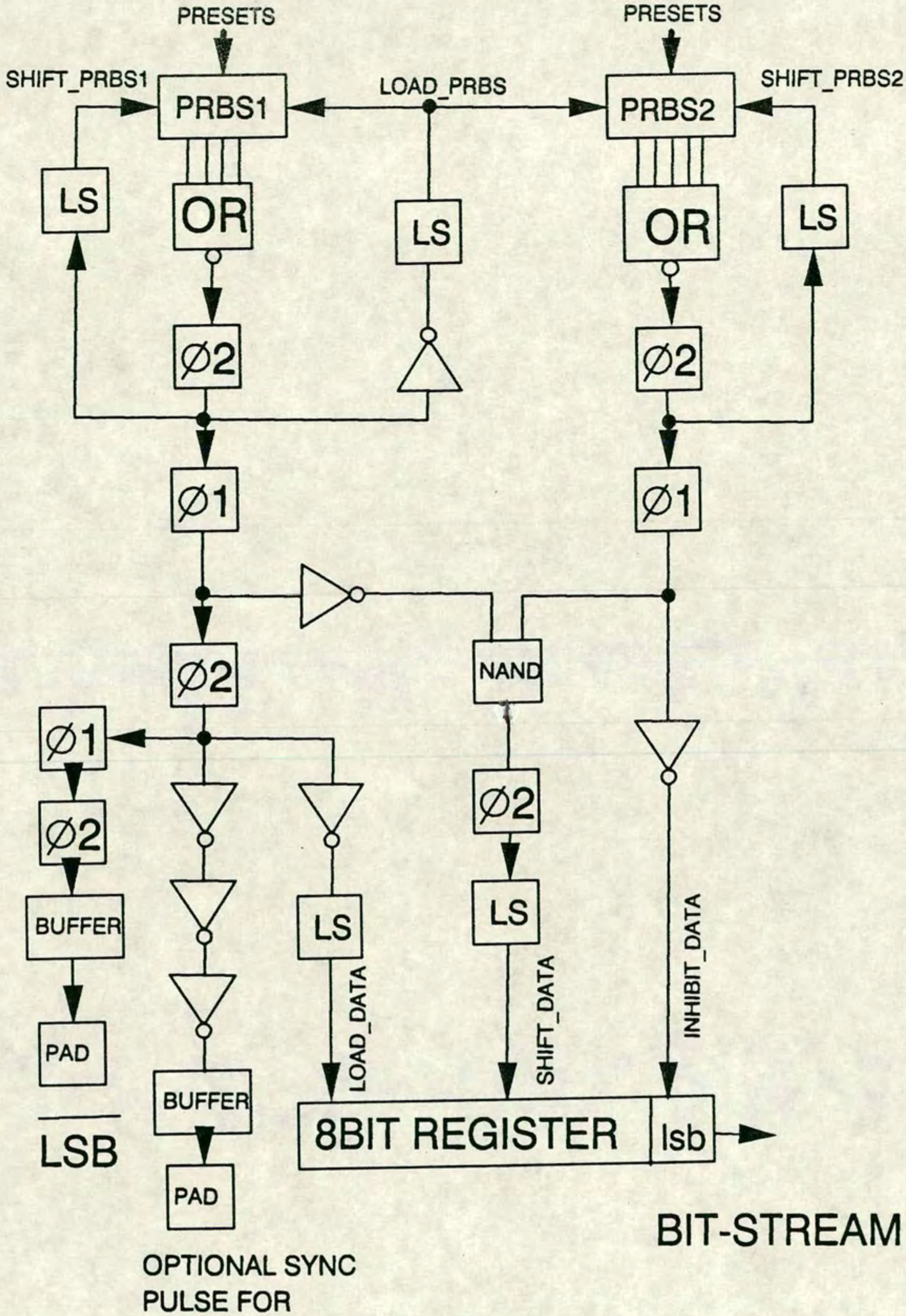


Fig. 5.44 The Parallel-to-serial converter block diagram

PRBS2 counts the number of optional leading 0's that should appear before the lsb of the actual data word (Fig. 5.45). It produces the signals to shift the PRBS2 registers during counting, to inhibit the data output stream, and to load the data registers with the external parallel data word.

The timing diagram of the major control signals in the schematic is shown in Fig. 5.46. The details of their generation and functions are described below:

(i) EP1 — The endpoint signal of PRBS1.

EP1 goes HIGH when the outputs of the PRBS1 registers are 1111X, where X is the "don't care" state. This endpoint condition takes care of the lockout state of 11111. A 4-input NOR gate is used to generate EP1. This signal is programmed to go HIGH every N clock cycles, for an N-bit system word. It is also used to generate the loading signal for both of the PRBS1 and PRBS2 counters. If the counter is in the lockout state initially, then EP1 is HIGH, and loads the registers with the chosen preset word (according to Table 5.3), which appears at the register outputs one clock cycle later. The registers then starts shifting until the endpoint is reached, thus completing one counting cycle.

(ii) EP2 — The endpoint signal of PRBS2.

The endpoint state for PRBS2 is 00000. This state was chosen, instead of the simpler 1111X state used in the PRBS1, owing to its different function. As mentioned earlier, PRBS2 controls the position of the data word within a longer system word. To achieve this, the PRBS2 counter is loaded with the chosen preset 5-bit word at the same time as PRBS1. This preset word (Table 5.3) represents the number of desired leading 0's in the system word (Fig. 5.45). But unlike PRBS1, shifting is disabled after the endpoint EP2 has been reached. Since shifting is controlled by the EP2 signal itself, EP2 remains HIGH until the next loading operation. During this period, both the loading and shifting signals are LOW, and the registers must retain its endpoint state. The choice of 00000 simplifies the design of the register in PRBS2.

In the case where no leading 0's are required in the system word, the endpoint state of 00000 is chosen as the preset word, and EP2 is always HIGH.

A 5 input NOR gate is used for detection. Note that although this endpoint state does not overcome the lockout state of 11111, PRBS2 is loaded every system word cycle. Since PRBS1 is lockout-free, PRBS2 will always return to normal counting after a maximum of 31 clock cycles.

(iii) LOAD_PRBS — Signal which loads both PRBS counters. This is given by:

$$\text{LOAD\_PRBS} = \frac{\Delta}{2}(\text{EP1})$$

where $\frac{\Delta}{2}(\text{signal})$ = signal delayed by ½ a clock cycle.

Level shifting for this signal is necessary before it can be applied to the registers, because the pass transistor loading of the registers uses D-MESFETs.

(iv) SHIFT_PRBS1 — Signal which shifts the PRBS1 registers during counting. This is given by the complement of (iii):

$$\text{SHIFT\_PRBS1} = \frac{\Delta}{2}\overline{(\text{EP1})}$$

Level shifting is again necessary.

(v) SHIFT_PRBS2 — Signal to shift the PRBS2 registers. It is given by:

$$\text{SHIFT\_PRBS2} = \frac{\Delta}{2}\overline{(\text{EP2})}$$

If no leading 0's are required, this signal is always LOW.

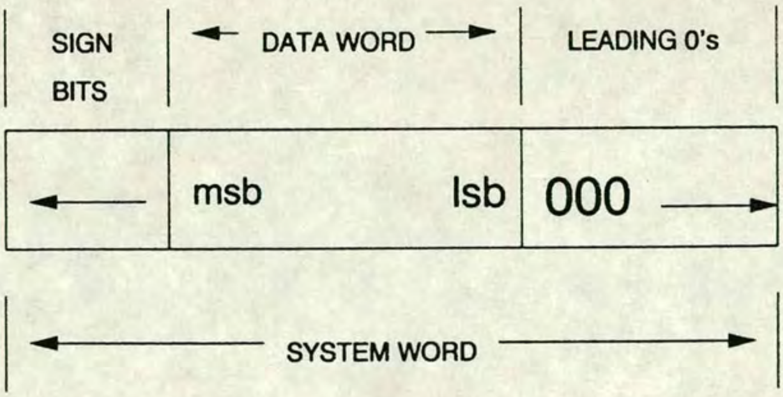


Fig. 5.45 The GaAs bit-serial system word format

(vi) DATA_INHIBIT — Signal to produce the leading 0's. This operates by forcing 0's at the output of the data registers with a NOR gate. This is given by:

$$\text{DATA\_INHIBIT} = \Delta\overline{(\text{EP2})}$$

If no leading 0's are required, this signal is always LOW.

(vii) LOAD_DATA — Signal to sample the parallel inputs and load them into the data registers. Level shifting is again necessary. This is given by:

$$\text{LOAD\_DATA} = \frac{(3\Delta)}{2}(\text{EP1})$$

where $\frac{(3\Delta)}{2}(\text{signal}) = (\text{signal})$ delayed by 1.5 clock cycles

(viii) SHIFT_DATA — Negative going signal to shift the data registers after the leading 0's have appeared at the end of the PRBS2 cycle.

During loading of the 8-bit shift registers and masking of the serial data stream using DATA_INHIBIT, the shift registers should retain the loaded parallel data and be prevented from shifting. This signal is given by:

$$\text{SHIFT\_DATA} = \overline{\frac{(3\Delta)}{2}(\text{EP1}) + (3\frac{\Delta}{2})\overline{(\text{EP2})}}$$

$$= 3\frac{\Delta}{2}(\overline{\text{EP1}}\ \overline{\text{EP2}})$$

In practice, the complement of this signal is produced using a NAND gate, and then pipelined through a latch to a level shifter before it is applied to the DATA registers.

(ix) LSB — Signal to indicate the lsb of the serial data stream. This is simply given by:

$$\text{LSB} = 2\Delta(\text{EP1})$$

where $2\Delta(\text{signal})$ is (signal) delayed by two clock cycles.

Note that signals going off-chip in Fig. 5.44 (LSB, SYNC) are pipelined and are delayed an additional half clock cycle.

The schematics for the PRBS1, PRBS2, and DATA registers are shown in Figs. 5.47, 5.48, and 5.49 respectively. They are derived from the D-MESFET pseudo-dynamic latch, with a loading path added for a fixed preset input for each of the PRBS registers. The PRBS2 register is slightly modified to maintain the correct endpoint state when shifting is disabled at the endpoint (EP2). By choosing

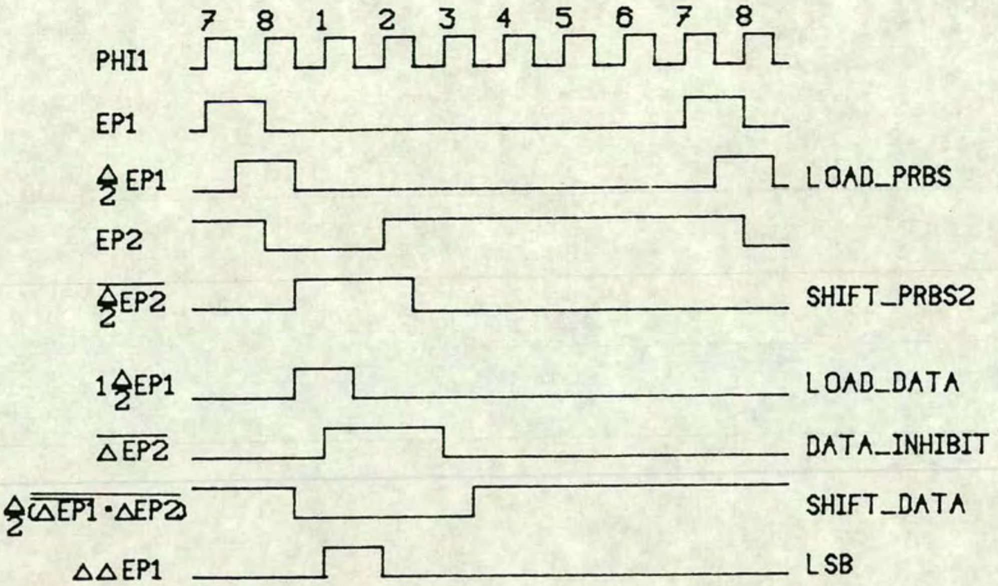


Fig. 5.46 Timing diagram of control signals in the parallel-serial converter

an endpoint state of 00000, the necessary modification to the shift register is minimised, and only a small pull-up device need to be added.

The DATA register has a passive refresh path such that it retains the loaded data when both SHIFT_DATA and LOAD_DATA signals are LOW, i.e. when the leading 0's are being output. The loading and shifting sequence must follow the correct order to prevent the data from being corrupted by the data from the preceding stage. This requirement is matched by the control logic that generate the SHIFT_DATA and LOAD_DATA signals. The refresh path uses a unique configuration of a D-MESFET pass transistor and a D-MESFET diode in parallel, to provide full non-clocked refresh. Its operation is discussed in more detail in Chapter 6.

The level shifter used here is similar to that used in the clock generator, except that only one diode drop is needed. This is because this level shifter only needs to shift down to a minimum of -0.5 V, rather than the -1 V in the clock generator. The shifter gives a swing of ± 0.5 V to drive the D-MESFET pass transistors. The extra negative supply rail of -0.6V can be the same as that in the clock generator. However, for testing and isolation purposes, it should be from a separate pad. In each instance where it was used, one level shifter roughly equivalent to the size of two inverters is sufficient to drive all the D-MESFET's in the PRBS counter and data shift registers. Since the P/S converter is needed only at the system interface with bit-parallel circuits, and the D-MESFET's are never forward biased, the size and power dissipation of these level shifters do not become a major penalty. The compactness of the latches more than compensates for the extra area needed for the small number of level shifters and the extra voltage rail.

Fig. 5.50 shows a simulation of the P/S converter at a clock rate of 500MHz, with an 8 bit parallel data word of 01010101, a 13 bit system word length, and 3 leading 0's in the system word. Note the msb of the data word is repeated as the sign bit. Fig. 5.51 shows a simulation with the same inputs and clock rate, but programmed for no leading 0's. Fig. 5.52 shows a simulation with the same conditions as Fig. 5.50, but at a clock rate of 714MHz (clock period of 1.4ns). The converter contains about 470 transistors, and dissipates a DC power of about 90mW. Although these are fairly large figures, only one converter is needed in a complete bit-serial system. A representative SPICE input file is given in Appendix 6.

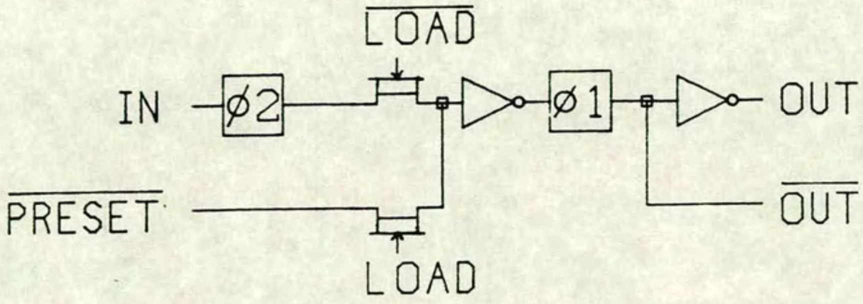


Fig. 5.47 The PRBS1 shift register

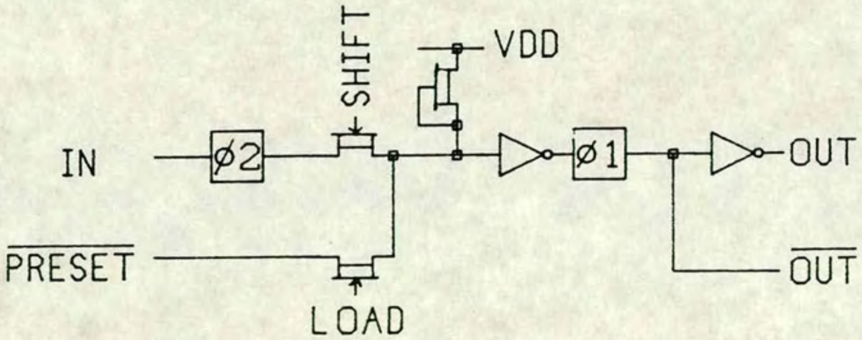


Fig. 5.48 The PRBS2 shift register

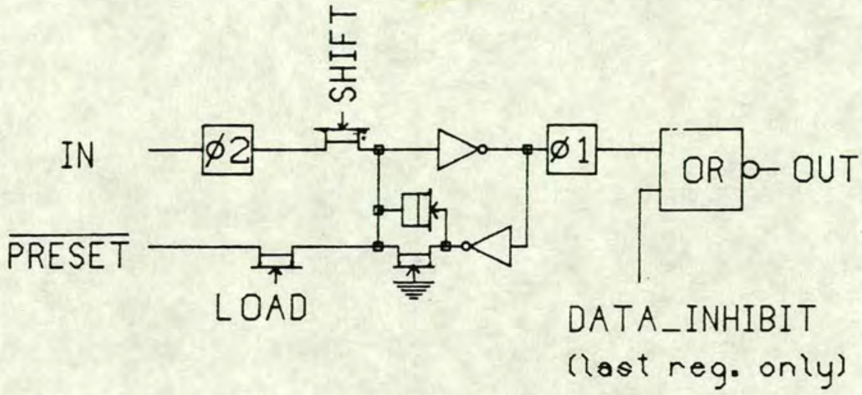


Fig. 5.49 The loadable data register for the 8-bit converter

5.5.5. The Serial to Parallel Converter (S/P converter)

The function of the S/P converter is to extract the data word of a fixed length from the system word, possibly for a digital-to-analogue converter. The schematic of the converter is shown in Fig. 5.53. In this version, the output is an 8 bit parallel word that can be interfaced with ECL buffers to drive 50Ω at low speed. The nominal maximum word rate is approximately 55.6 M(samples/s), for a system clock rate of 500 MHz (500/9 Msamples/s).

Referring to Fig. 5.53, the single PRBS counter used here counts the most significant bit position of the pre-programmed data word and load the parallel latches at the appropriate clock cycle. The design of the S/P converter is much simpler than the P/S converter because less programmability is involved. The counter design is same as PRBS1 in the P/S converter in the previous section and therefore Table 5.3. applies. A data-ready signal, denoted by SYNC, allows external timing interface.

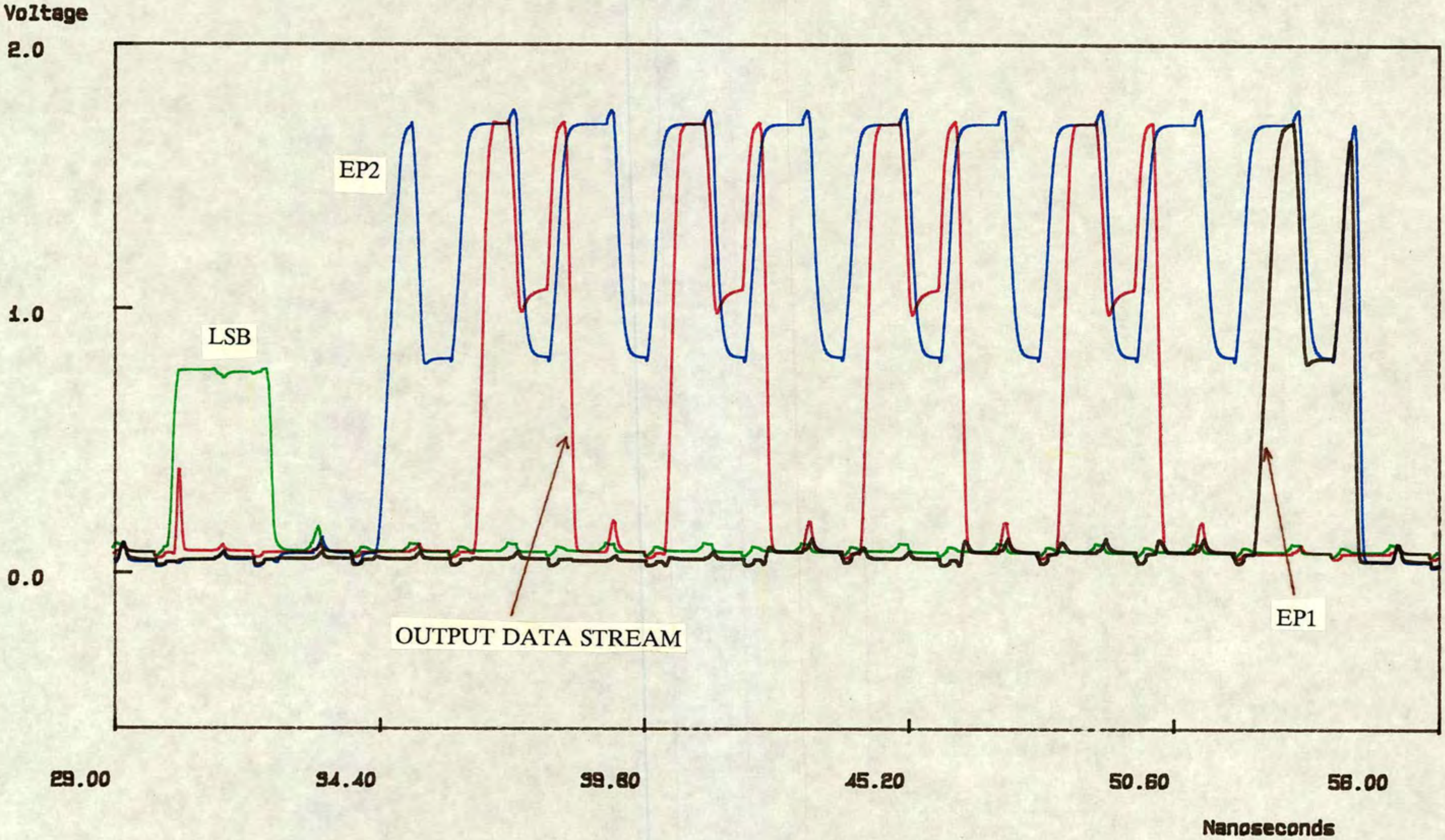


Fig. 5.50 Simulation result of the parallel-serial converter

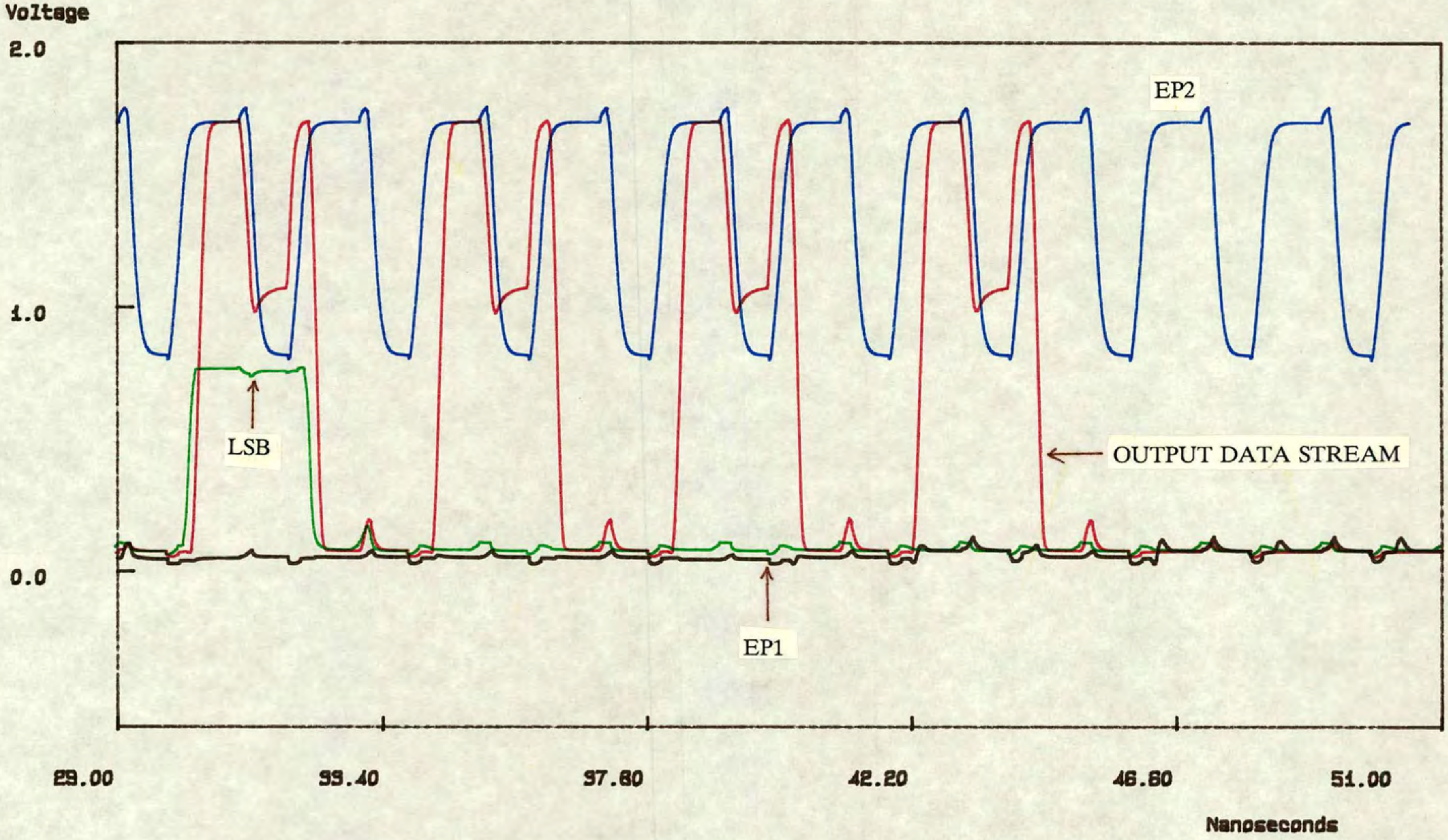


Fig. 5.51 Second simulation example of P/S converter

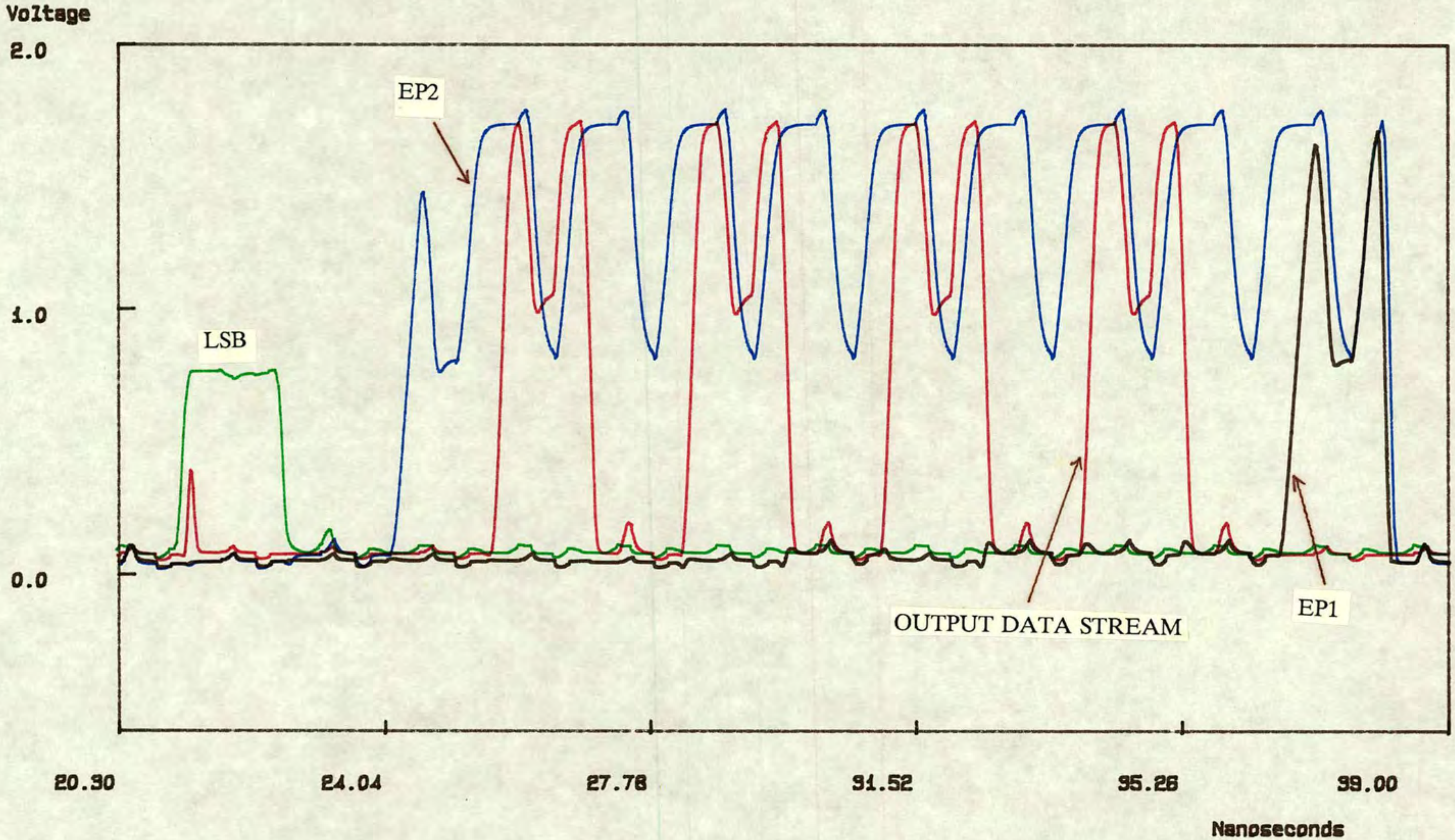


Fig. 5.52 Simulation of P/S converter at 714MHz

The main control signals are summarised below:

(i) EP — Endpoint signal from the PRBS counter.

EP is HIGH when the PRBS registers are 1111X. This caters for the lockout condition.

(ii) LOAD_PRBS — Level shifted signal to load all 5 PRBS registers, given by:

$$\text{LOAD\_PRBS} = \frac{\Delta}{2}(\text{LSB})$$

Level shifting is necessary before being applied to the registers.

(iii) LOAD_DATA — Loads the latches with the parallel word. This is given by:

$$\text{LOAD\_DATA} = \Delta(\text{EP})$$

Level shifting is again necessary.

(iv) $\text{SYNC} = \overline{\text{LOAD\_DATA}}$

This pulse is used for external synchronisation.

A simulation of the S/P converter at a clock rate of 714MHz is shown in Fig. 5.54. The input and output signals are as labelled on the diagram. It is not practical to plot all of the 8 parallel outputs on the same plot. Therefore only the three most significant bits are shown. In this example, the converter was programmed to retrieve 8-bit data within a system word given by:

system data input = 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1

msb position of 8 bit data = 15

where lsb is bit 1.

Therefore the retrieved data is given by:

8−bit parallel output data = 0 1 1 0 0 1 1 0

In this example, the system and data words have very different word lengths. In the case of a data word length being 15 bits, then the msb of the data word would be at position 15 and the lsb is at position 1. The timing of the S/P converter caters for this extreme case because the loading of the parallel registers occur one clock cycle after the endpoint is reached (the LOAD_DATA signal).

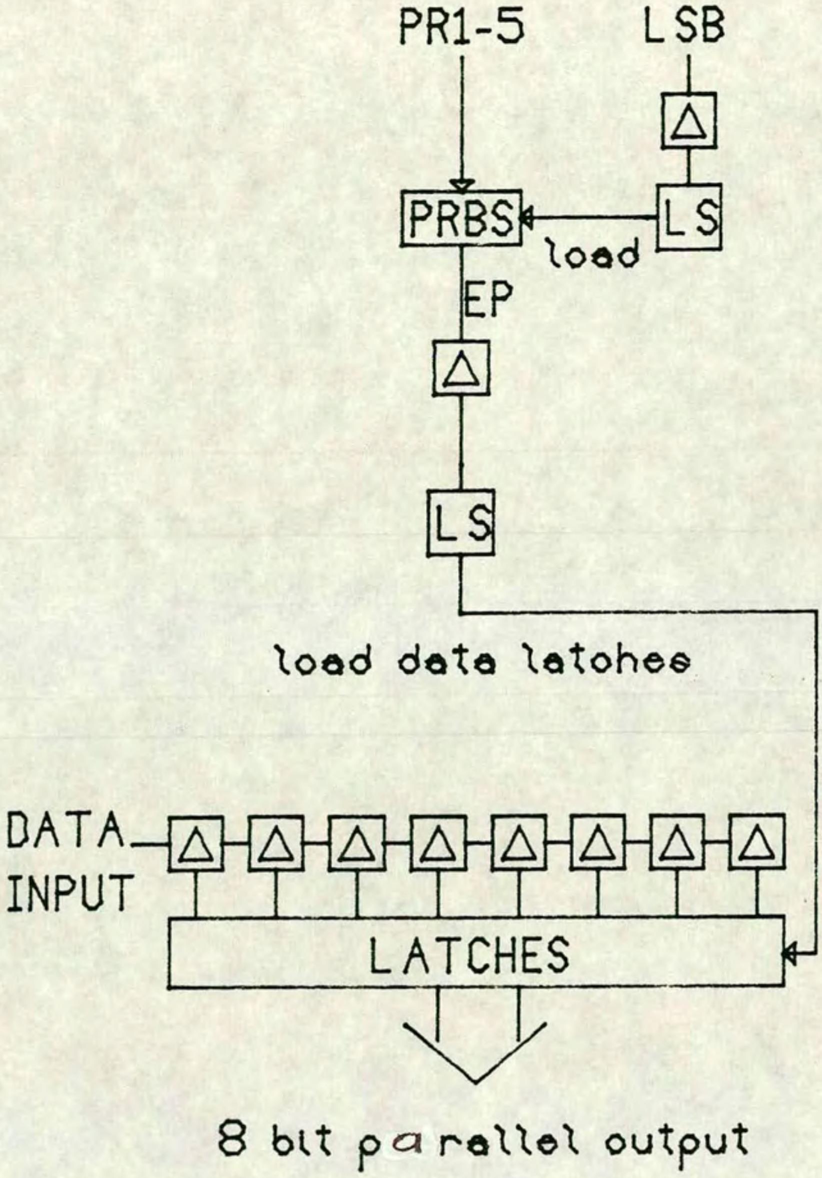


Fig. 5.53 The S/P converter block diagram

The S/P converter contains approximately 304 MESFETs, and dissipates a DC power of 58.3mW. Again, as with the P/S converter, only one converter is needed in the system. The SPICE input file is given in Appendix 7.

5.5.6. The PRBS counter chip

This chip design was placed on the test chip specifically for testing as a stand-alone circuit element. It provides an assessment of the performance of the counter, which is one of the limiting factors on system speed. The only inputs required are the two-phase clock signals. The PRBS counter is programmed for maximum count by loading the registers with 10000 (equivalent to a preset word of 01111), one clock cycle after endpoint goes HIGH. The output is therefore a pulse with a pulse width of one clock cycle and a period of 31 clock cycles. The endpoint is the same as that in the S/P converter in the previous section, i.e. 1111X. The major control signals follow similar principles as the S/P converter, but are used differently, and are listed below:

(i) EP — Endpoint signal. Goes HIGH when the register outputs are 1111X.

(ii) LOAD — Loads PRBS counter with 01111. This is given by:

LOAD = Δ(EP)

Level shifting is required before it can be applied.

(iii) OUTPUT — A GaAs level output pulse to indicate loading of PRBS. This is given by:

OUTPUT = Δ(EP)

The SPICE simulation of this counter is shown in Fig. 5.55. This shows in more detail the operation of the counter as used in the two converters. The endpoint signal and the derived loading signal for the PRBS registers are shown, as well as one of the register outputs. The clock rate was 714MHz. The counter was programmed to repeat every 5 clock cycles. It contained 121 MESFETs and dissipated 24.8mW of DC power. The SPICE input file is given in Appendix 8.

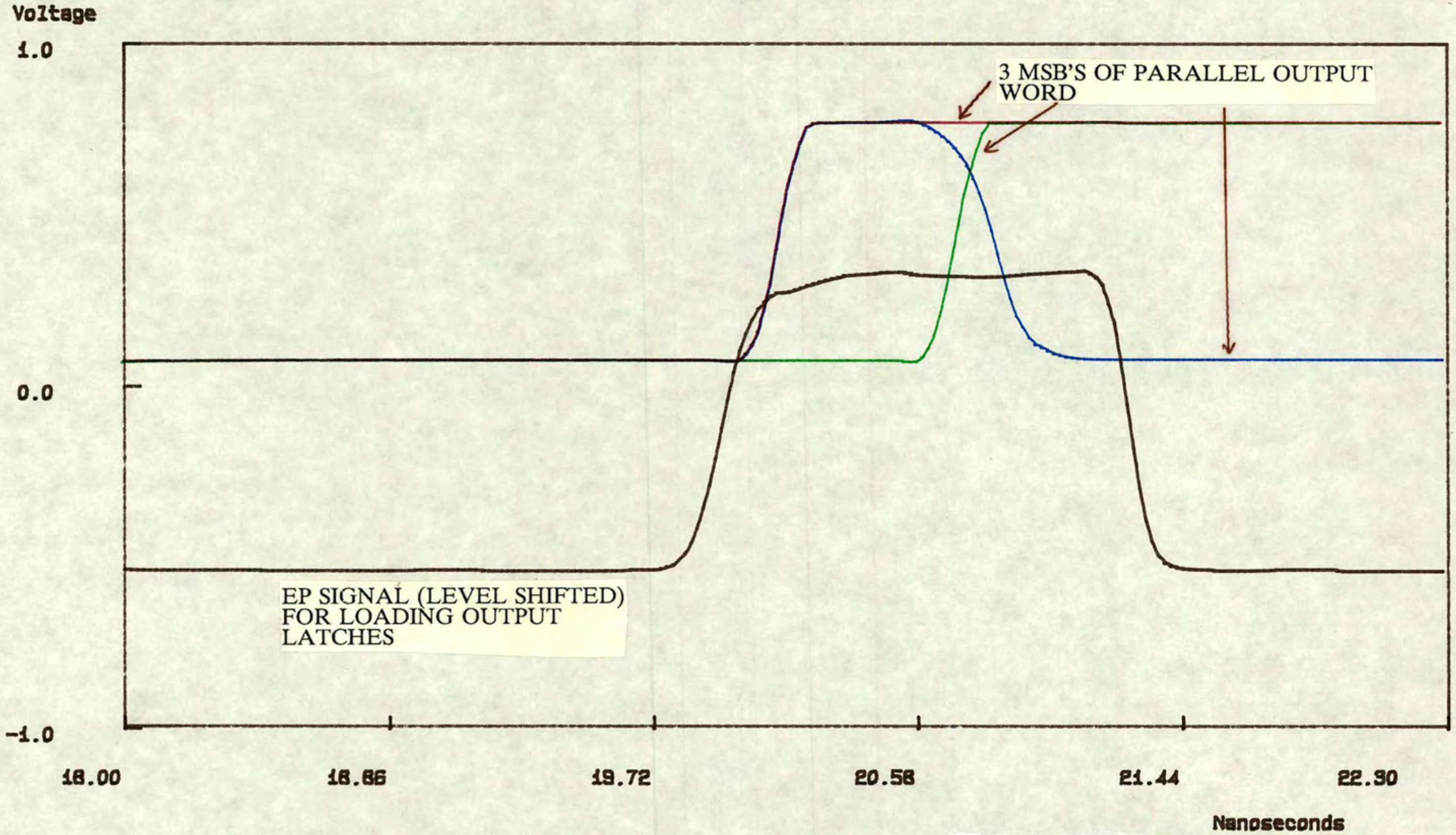


Fig. 5.54 Simulation result of the serial-parallel converter

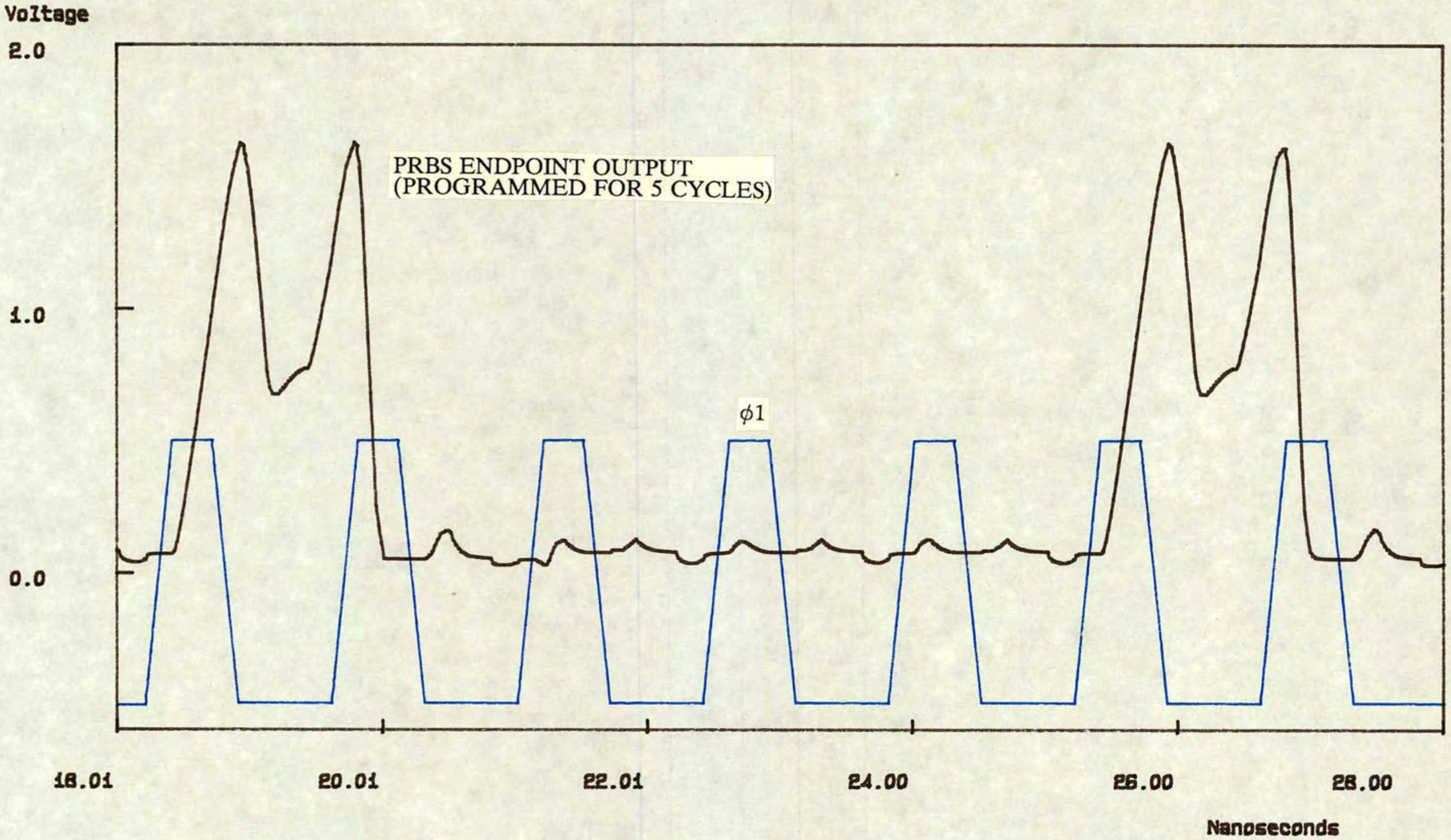


Fig. 5.55 Simulation result of PRBS counter

5.5.7. I/O buffers

5.5.7.1. A low speed ECL level output buffer

This is a simple non-inverting open-source buffer designed for low speed testing purposes, or in conjunction with the parallel outputs of the S/P converter. The circuit schematic is shown in Fig. 5.56. The buffer is made up of an open source E-MESFET output device, driven by 2 large inverters. It was designed to be capable of driving a 50Ω resistive load, in parallel with a 3pF capacitive load, at a nominal maximum signal rate of 50MHz. The outputs levels were designed to be ECL compatible.

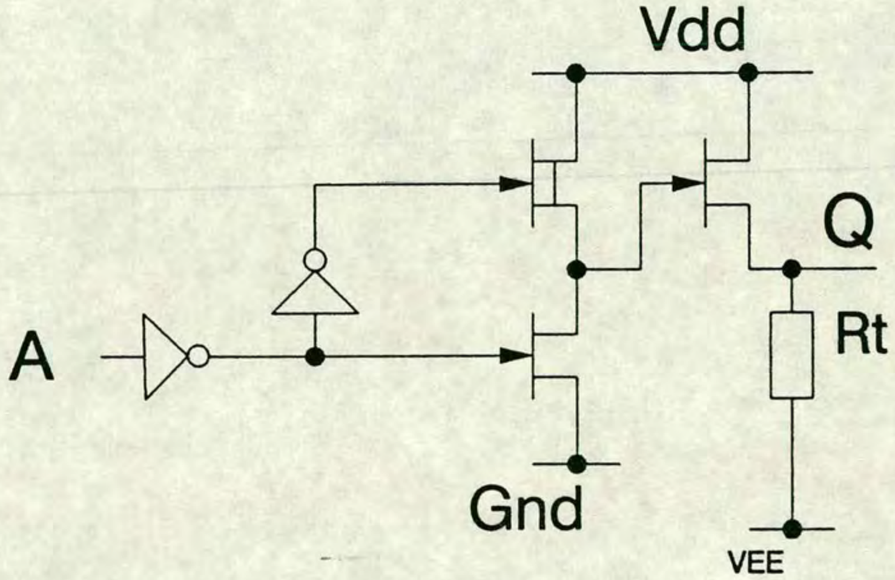


Fig. 5.56 Low speed GaAs to ECL buffer

In terms of layout, the large output E-MESFET pull-up transistors were inter-digitated and have increased source-drain separation for better thermal dissipation. The thermal properties of large GaAs MESFETs depend on the separations of the drain terminals and the substrate thickness [152]. A simulation of this buffer is shown in Fig. 5.57. The data rate was 200Mb/s, with a load capacitance of 6pF.

Note that 1.7V should be subtracted from the output voltage levels due to the level-shifted power rails. The buffer dissipated 11mW of DC power. The SPICE input file is given in Appendix 9.

5.5.7.2. Other I/O buffers

Apart from the GaAs to ECL level buffer, three other possible types of I/O buffers should also be considered:

(i) GaAs DCFL input protection buffer;

(ii) GaAs DCFL output buffer;

(iii) ECL level to GaAs DCFL level converter.

The ECL to GaAs level converter, item (iii), is not catered for among the cells designed, and requires external circuitry. Such a converter is relatively simple because of the low speed data inputs, such as that from an analogue-to-digital converter. Figs. 5.58 and 5.59 show the GaAs to GaAs I/O buffers (buffers (i) & (ii)). For GaAs level inputs, a large diode to Ground was used to prevent overdrive of the associated logic gate. The pull-up E-MESFET and D-MESFET devices in Fig. 5.59 are laid out as one device, sharing a common Schottky gate, to save space and interconnection. The dual MESFET pull-up configuration is particularly useful in providing a fast rise time but without the need to use a large D-MESFET. The E-MESFET provides initial pull-up current, and then the D-MESFET takes over as the output voltages rises and the former approaches cut-off. The D-MESFET also allows a full swing of up to V_{dd}, if it is not clamped by the receiving end of the signal.

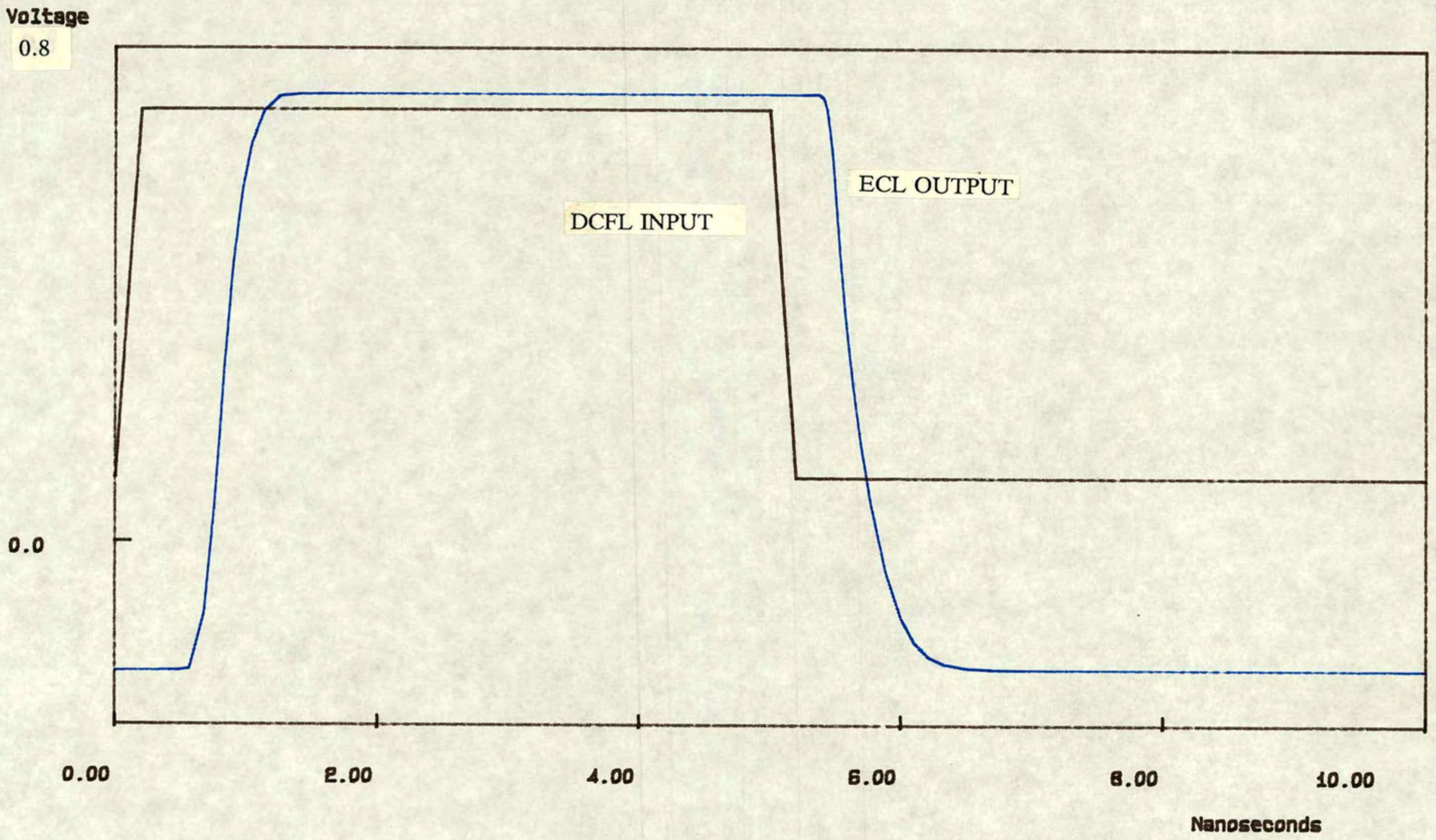


Fig. 5.57 Simulation result of GaAs-ECL output buffer

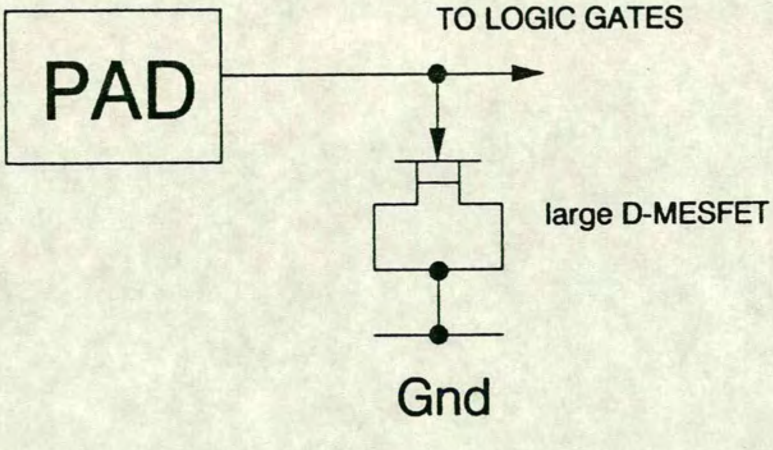


Fig. 5.58 GaAs level input protection circuit

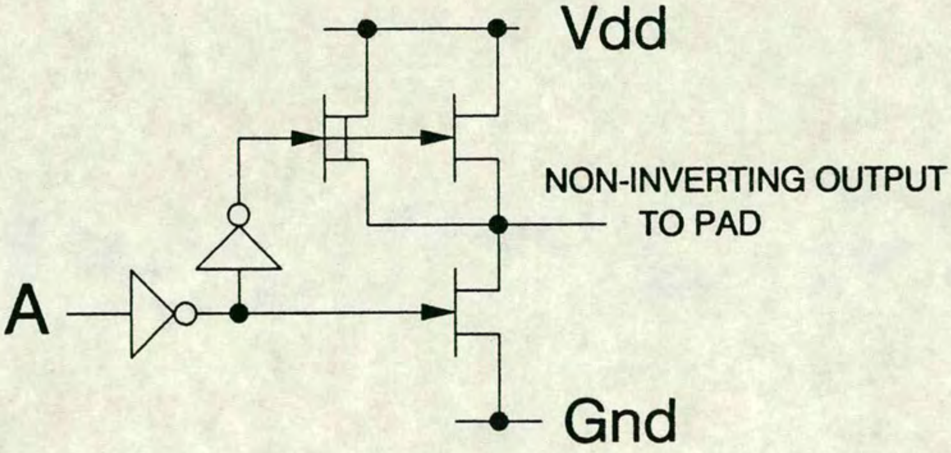


Fig. 5.59 GaAs level output buffer

The SPICE simulation results of the GaAs/GaAs output buffer are shown in Figs. 5.60. The bit rate is 1000Mb/s. The buffer is intended for use with a clock rate of 500MHz and a 500Mb/s data rate. However, it is essential that its speed is comparable to on-chip delays, even if the I/O buffers are pipelined. The buffer dissipated 52.4mW of DC power. The SPICE input file is included in Appendix 10.

5.6. The Layouts

The designs are plotted in Figs. 5.61 to 5.65. Fig. 5.66 also shows the layout of the clock generator on its own. The geometry is Honeywell's 1μm SAG MESFET process. For D-MESFET devices, the gate length is usually 1.5μm. Diodes are typically laid out as drain-source shorted MESFET's. For devices substantially wider than 50 μm, they are split into parallel devices each of 50 μm or smaller. In the converters where level shifting was extensively used, negative rails are separated from unrelated active areas to avoid backgating. The layers are summarised below:

(i) Active area — green;

(ii) Schottky metal — red;

(iii) Ohmic metal — not plotted;

(iv) N-prime implant — not plotted;

(v) Metal 1 — blue;

(vi) Contact window — black;

(vii) Metal 2 — orange;

(viii) Overglass — not plotted.

The Vdd and Vss rails are unbroken and are laid out on metal 2. The clock phases are on metal 1 initially, then internally distributed on metal 2. The negative going clock signal levels do not present the possibility of backgating (Chapter 4) even in dense areas of the layout. The negative level of -0.5 V was considered to be harmless with the current process. Since these cells are laid out as test chips, it is essential that they can be easily tested and debugged if problems arise. The main additional features that were introduced in the layouts to this end are summarised below:

(i) The negative supplies for the gate and source of the level shifters are separated to allow for fine adjustment if necessary. The level shifters for logic circuits, such as those in the P/S and S/P converters, and those in the clock generator also had separate negative rails for flexibility of adjustment. This results in a large number of ultimately redundant pads for these supplies. In a final design, these extra pads can be eliminated.

(ii) The two phases of the clock are also connected to "bare" (unprotected, non-voltage limited) pads to allow for external clocking if the internal clock generator fails.

(iii) Some devices in the output buffers have physical width of larger than 50 μm to avoid cross-overs at the gate or output.

(iv) The D-MESFET pull-ups use a gate length of 1.5 μm, in order to allow for possible adjustment of D-MESFET implant doses during processing [87].

Most of the extra pads in these test chips are unnecessary in an integrated system. Apart from the redundant supply pads, the parallel presets for the PRBS converters will also be unnecessary for a given fixed application. As a result, rather inefficient use of GaAs area can be seen in some of the layout plots.

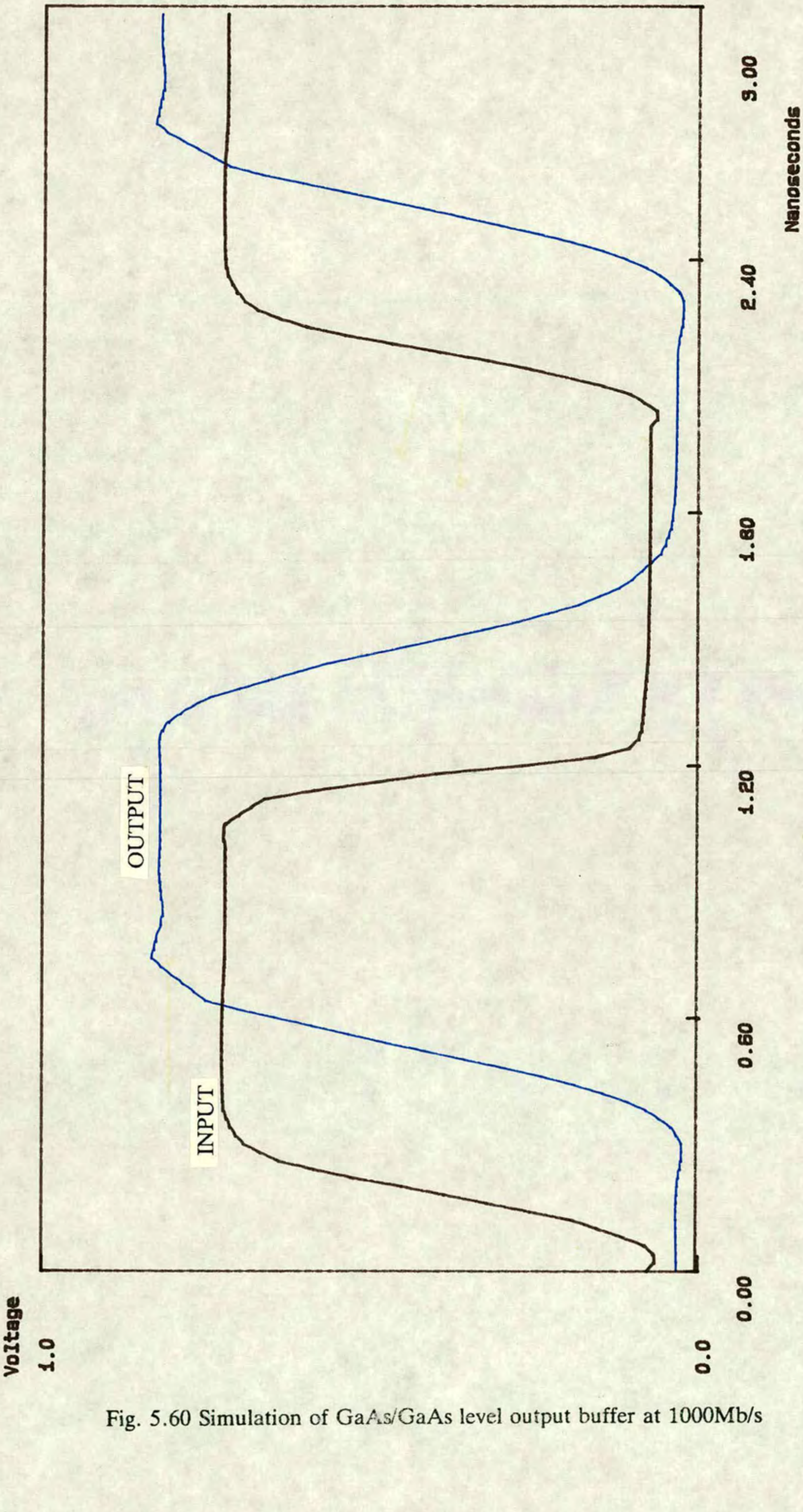


Fig. 5.60 Simulation of GaAs/GaAs level output buffer at 1000Mb/s

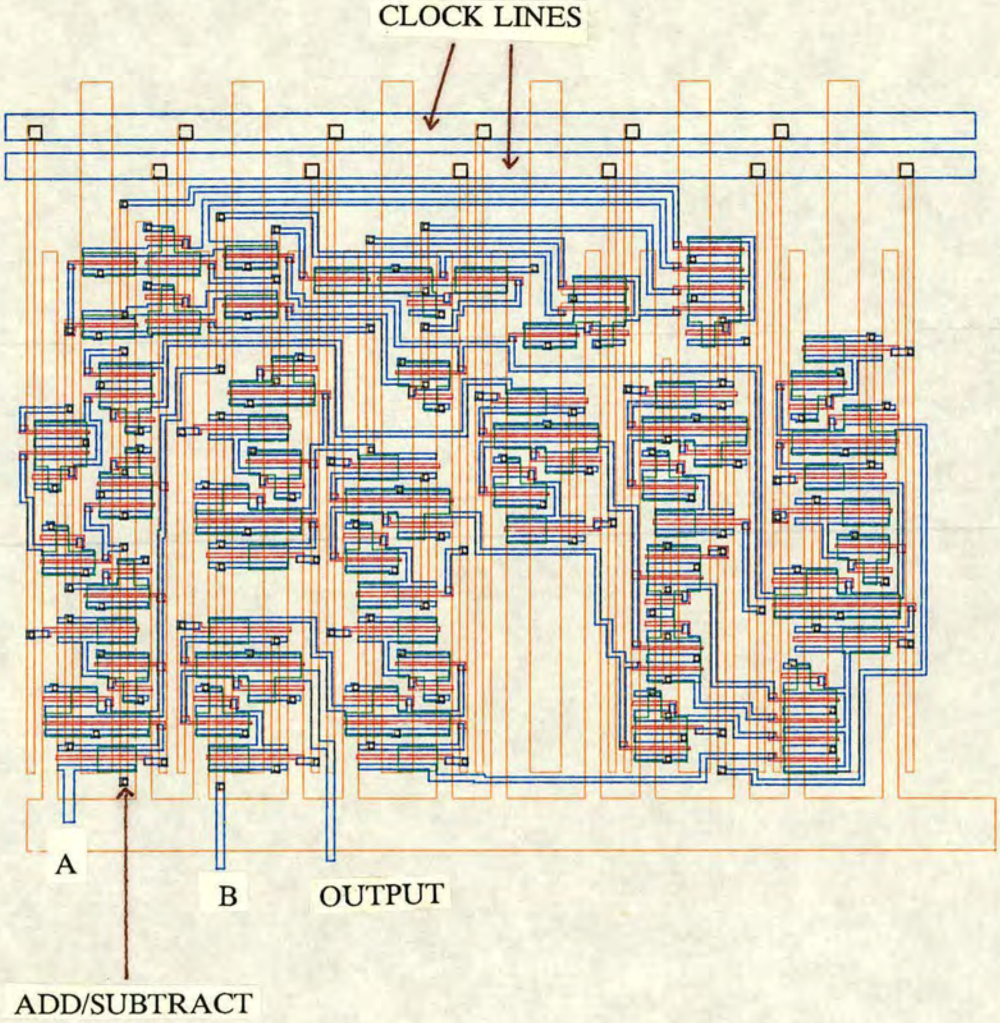


Fig. 5.61 The adder layout

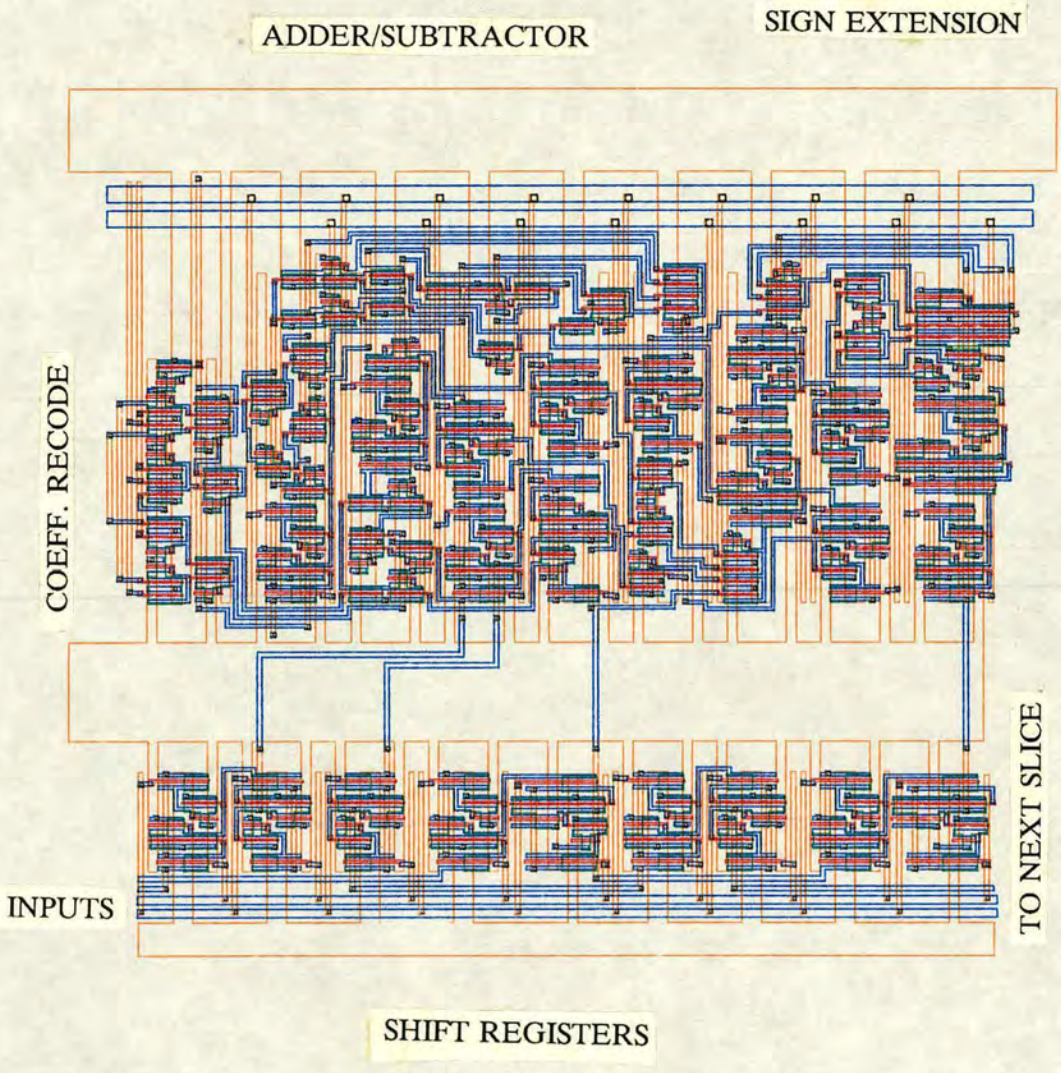


Fig. 5.62 The multiplier layout

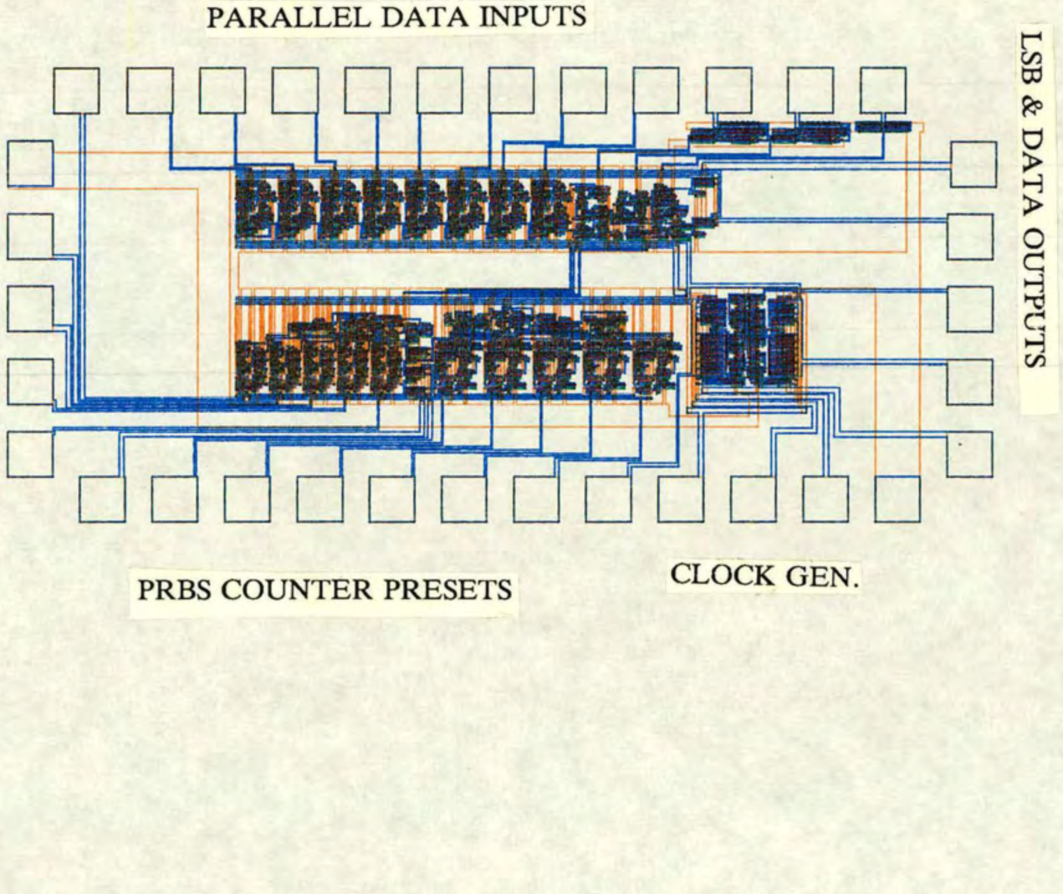


Fig. 5.63 The Parallel-to serial converter layout

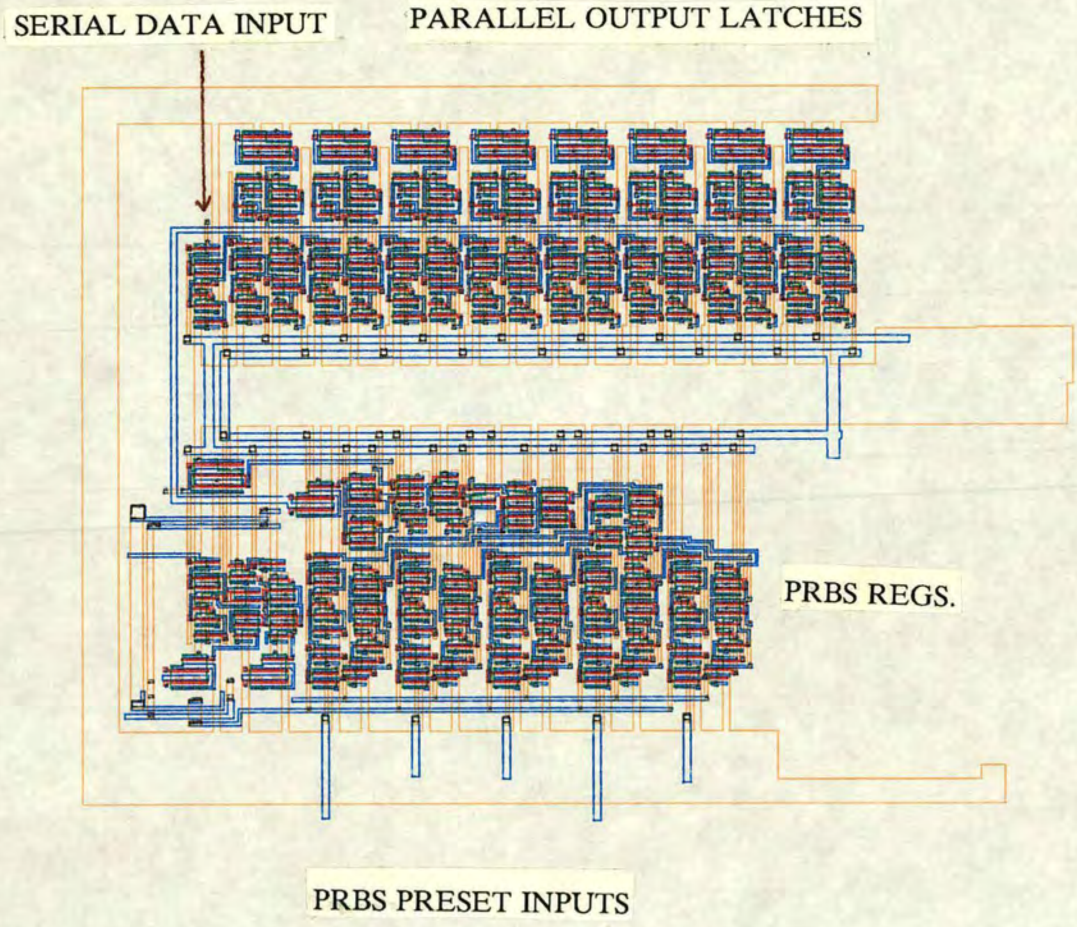


Fig. 5.64 The Serial-to-parallel converter layout

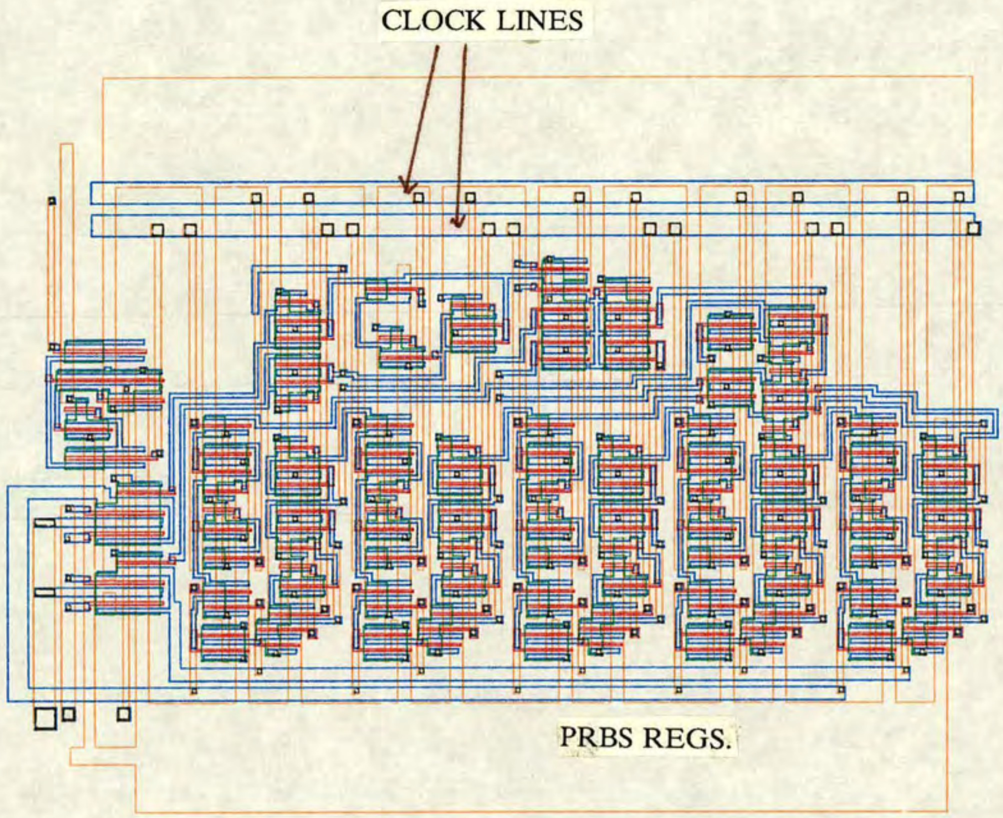


Fig. 5.65 The PRBS counter layout

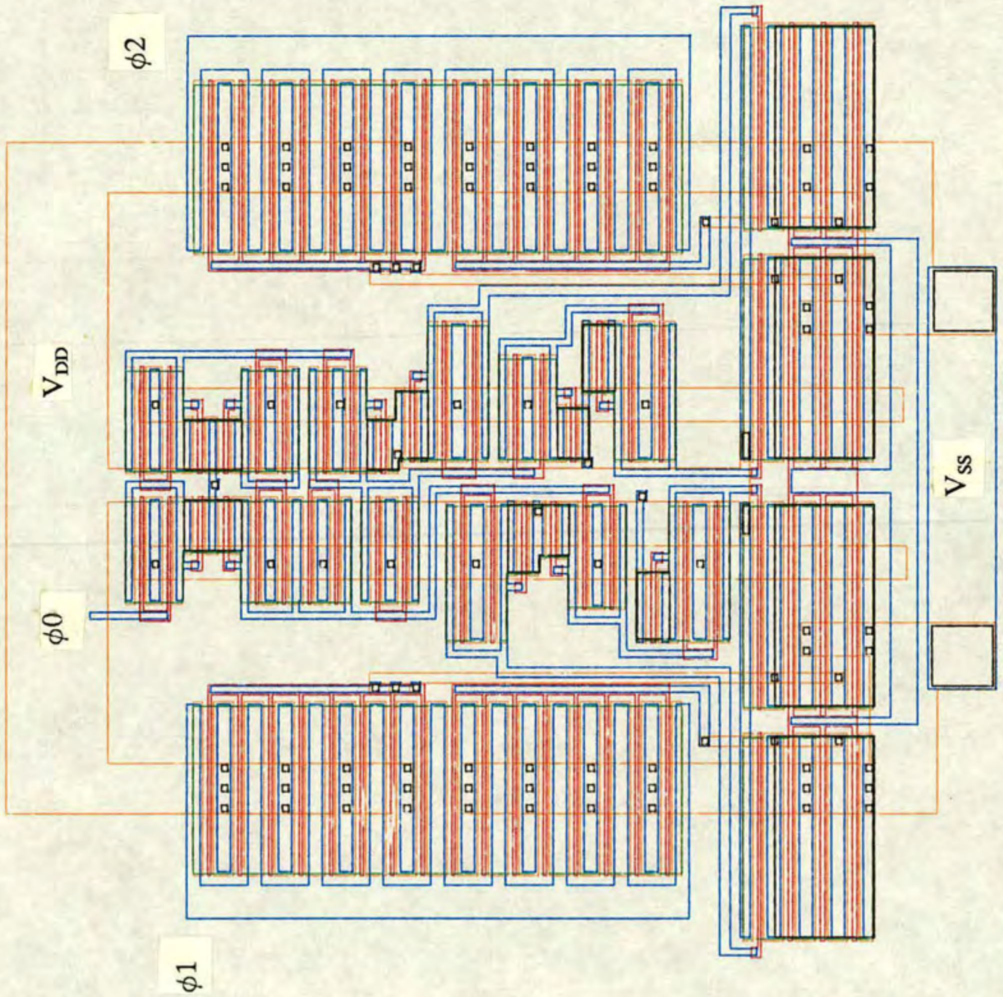


Fig. 5.66 The 2-phase clock generator layout

5.7. The MODFET mask set

The GaAs designs described above had originally been intended for fabrication on a MESFET process. The lack of mask set space meant that some of these designs were eventually placed on a MODFET mask set in standard frames. The changes and details of these frames can be summarised below:

(i) A total of 3 frames have been submitted: two 10×6 and one 4×5 (10×6 meaning a 10 pads by 6 pads frame, i.e. a total of 28 pads). These conform to the probe card dimensions used by Honeywell.

(ii) The first 10×6 frame contains a clock generator driving one multiplier slice, one adder, and one unit delay element.

(iii) The second 10×6 frame contains a clock generator driving one PRBS counter and one 3-level generation and detection circuit (Chapter 6).

(iv) The 4×5 frame contains two DCFL to ECL output level buffers for 50Ω lines.

(v) The MODFET cells have been designed to be tested on their own rather than as a system. This is because of the space limitations of the standard frames. For instance, one 10×6 frame cannot accommodate a full 6 bit multiplier unless the clock generator is omitted. Therefore their I/O ports are not pipelined. However, the multiplier slice can be cascaded to form a workable but incomplete and inefficient multiplier of any bit-length. Therefore it can be tested as a system, with the adder and delay elements, at a lower speed than in an integrated system.

(vi) The projected performance of the MODFET cells (individually) is 800MHz clock rate, i.e. a data rate of 800Mbit/s.

Figs. 5.67 to 5.69 shows the floor plan of the 3 frames that were submitted. Since all cells have already been plotted, there is no need to plot all 3 frames in detail. Each of the two 10×6 frames occupies an area of 1.54mm×0.9mm, while the 4×5 frame occupies 0.58mm×0.74mm.

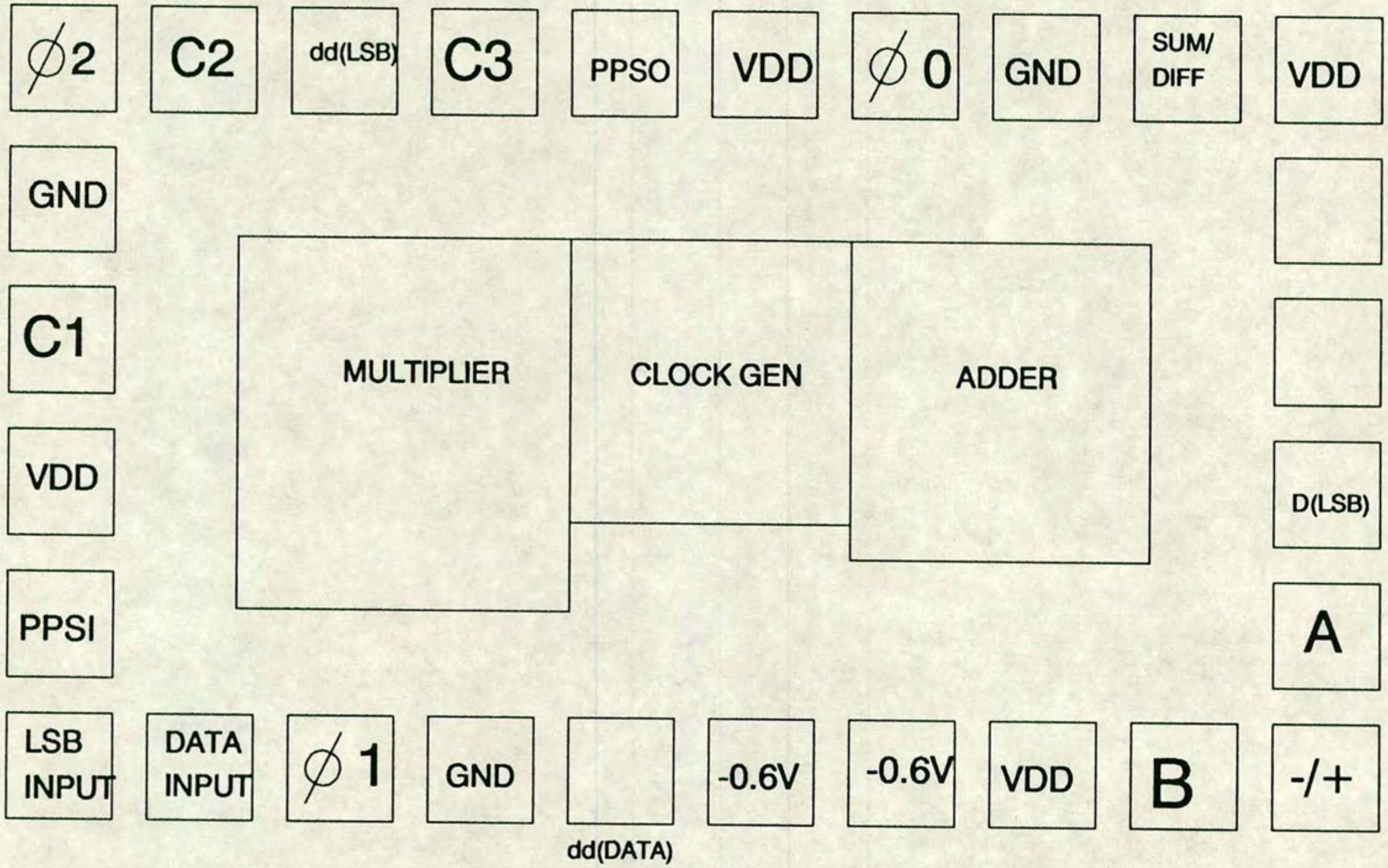


Fig. 5.67 MODFET 10×6 frame I, with multiplier, adder.

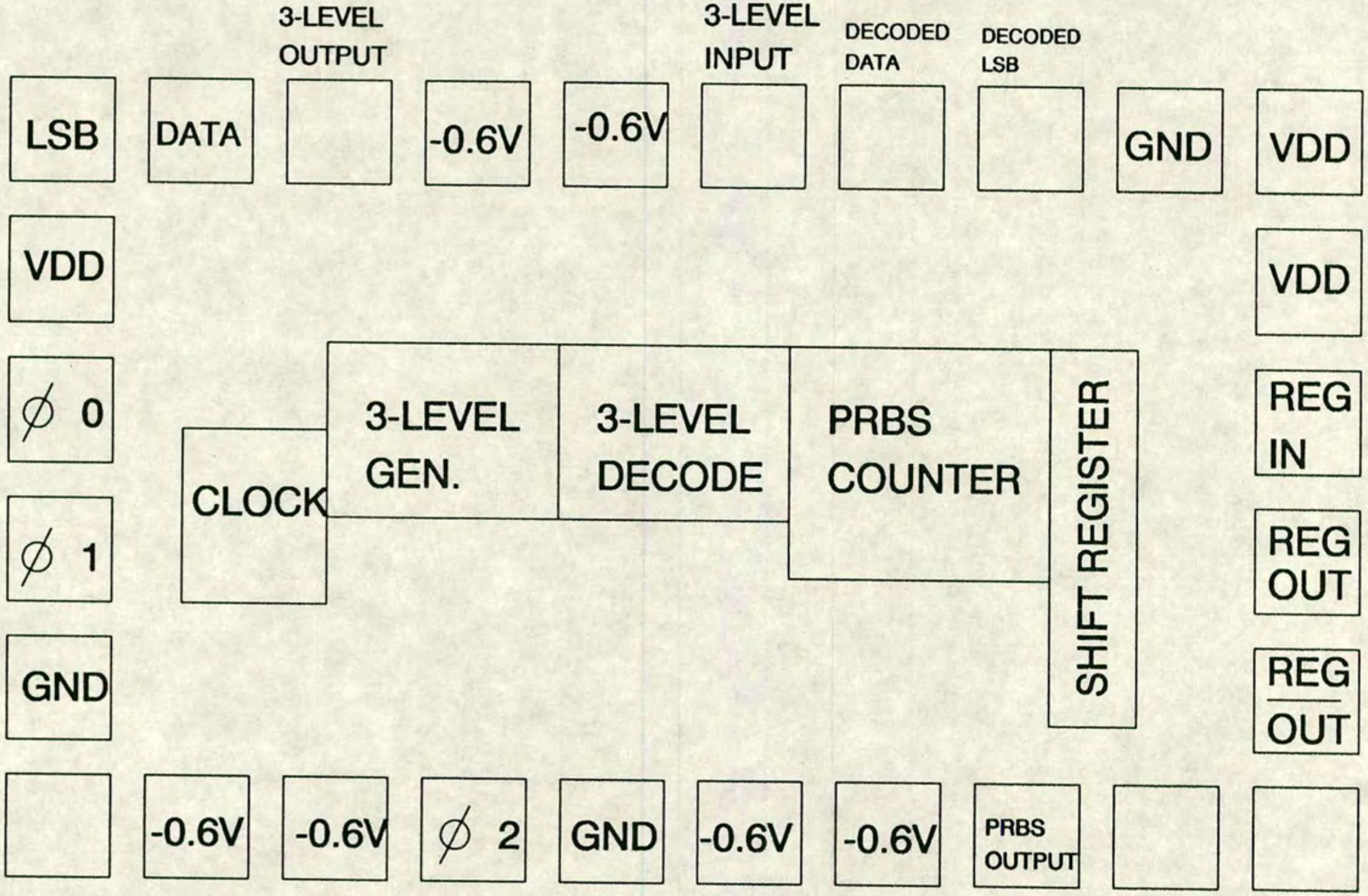


Fig. 5.68 MODFET 10×6 frame II, with 3-level circuit and counter

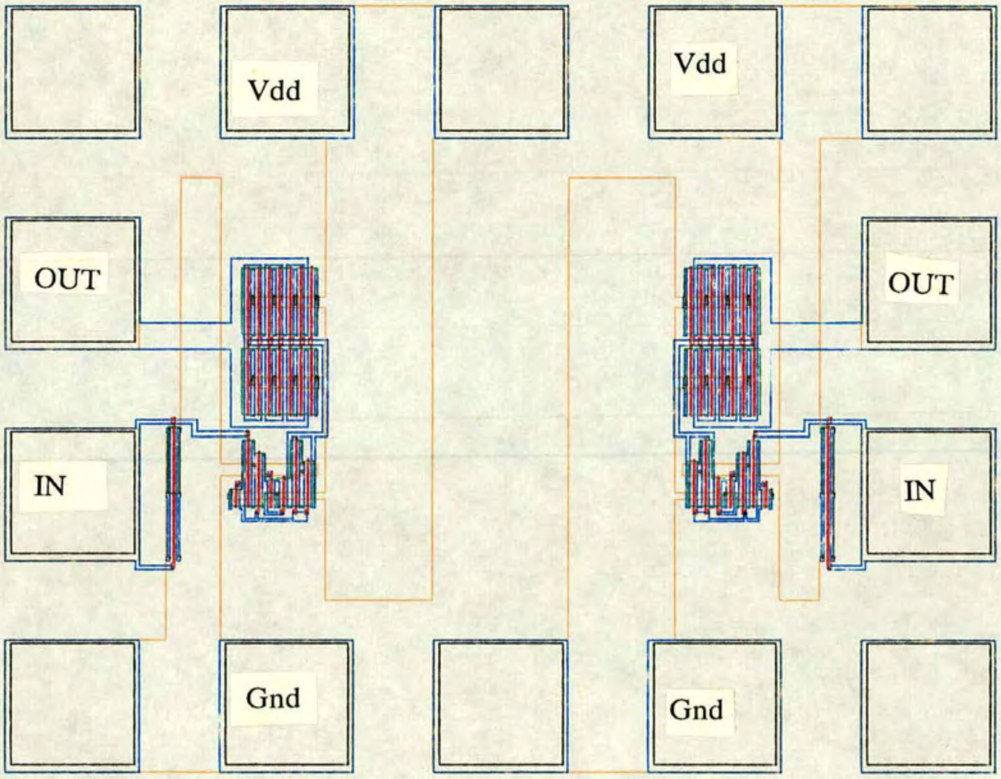


Fig. 5.69 MODFET 4×5 frame, with ECL buffers

5.8. Test results

The bit-serial cells that were laid out in the MODFET-modified mask set have been fabricated. Problems encountered in the MODFET processing have resulted in these designs being fabricated with a MESFET process. Even with this unexpected change, the clock generator, pseudo-dynamic latches, and the GaAs output buffers have been tested at low speed, using standard Honeywell probe cards, and the results were found to be promising. These results, together with results on circuits described in the next Chapter, are included in Appendix 15.

FIR implementation study

Although only some of the cells were fabricated, and they were not enough to implement a practical test system, it is useful to illustrate their use with a simple FIR filter. The theory of FIR filters are not discussed in detail here. This can be found in a number of references [104]. Fig. 5.70 shows the block diagram of one possible implementation of such a filter. Based on the sizes of the existing cells and without optimisation, a floorplan of such a filter is shown in Fig. 5.71. The largest operator is clearly the multiplier, while compaction ultimately depends on the system word length, which causes the size of the word delay element (T) to increase linearly. All other elements, however, are independent of data word length. The die would measure approximately $3 \times 3\text{mm}^2$ for a coefficient word length of 6 bits, and would have 26 pads. Most pads are occupied by the three parallel coefficients. Since the bit-serial approach meant that there are an abundant number of pads, the parallel coefficients are not detrimental. In the unlikely case that the reverse is true, they can be easily multiplexed.

Assuming that the data word length is 24 bits (24 shift registers occupies half the physical height of a filter slice) with 6-bit coefficients, the chip would dissipate about 1.2W, including I/O buffers and the clock generator.

5.9. Summary

In this Chapter, the design and simulation of the GaAs cells have been discussed in detail. An FIR filter was presented as a simple integration case study. The test results of the fabricated circuits have successfully demonstrated some of the main elements of these cells. The D-MESFET pseudo-dynamic latches used for pipelining, and the ±0.5V two-phase clock generators were found to be fully functional. Apart from demonstrating the basic circuit ideas, they also imply the functional correctness of the SPICE simulations, based on modifications to incorporate a Honeywell MESFET model, which have been presented for the cells that have not yet been tested or fabricated.

The floorplan of a 6-bit coefficient third order FIR has been presented, and showed the space efficiency and modularity of bit-serial operators achievable in a small chip area. With the example, a maximum word length of 24 bits, limited by the area taken up by the word delay element (T), would mean that the chip is capable of processing at a minimum of 20.8 Msamples/s, assuming a clock rate of 500MHz. This provides much wider bandwidth than silicon MOSFET-based bit-serial circuits. The integration level of present GaAs processes is also high enough to accommodate the filter on one single $3\times3\text{mm}^2$ chip — a major advantage of the bit-serial architecture. This would not be possible with a bit-parallel approach.

Table 5.4 below summarises the size and power dissipation of the main cells. All sizes exclude pads

cell	sizeμm	power dissipation/mW	number of MESFETs
multiplier slice	600×500	41.7	246
adder	387×252	30	167
clock generator	270×270	62	28
S/P converter	570×600	58.3	304
P/S converter	1000×550	90	470
shift register	90×120	2.3	12

Table 5.4 Summary of main cells

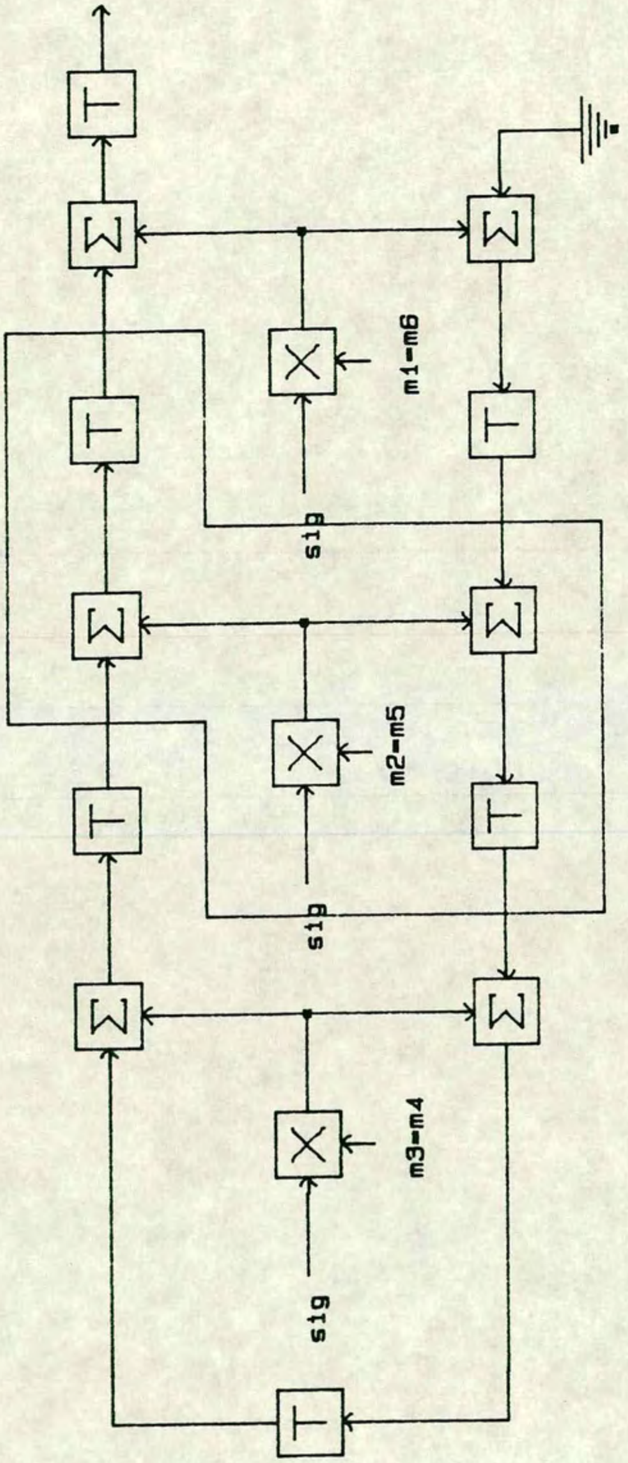


Fig. 5.70 FIR structure implementation

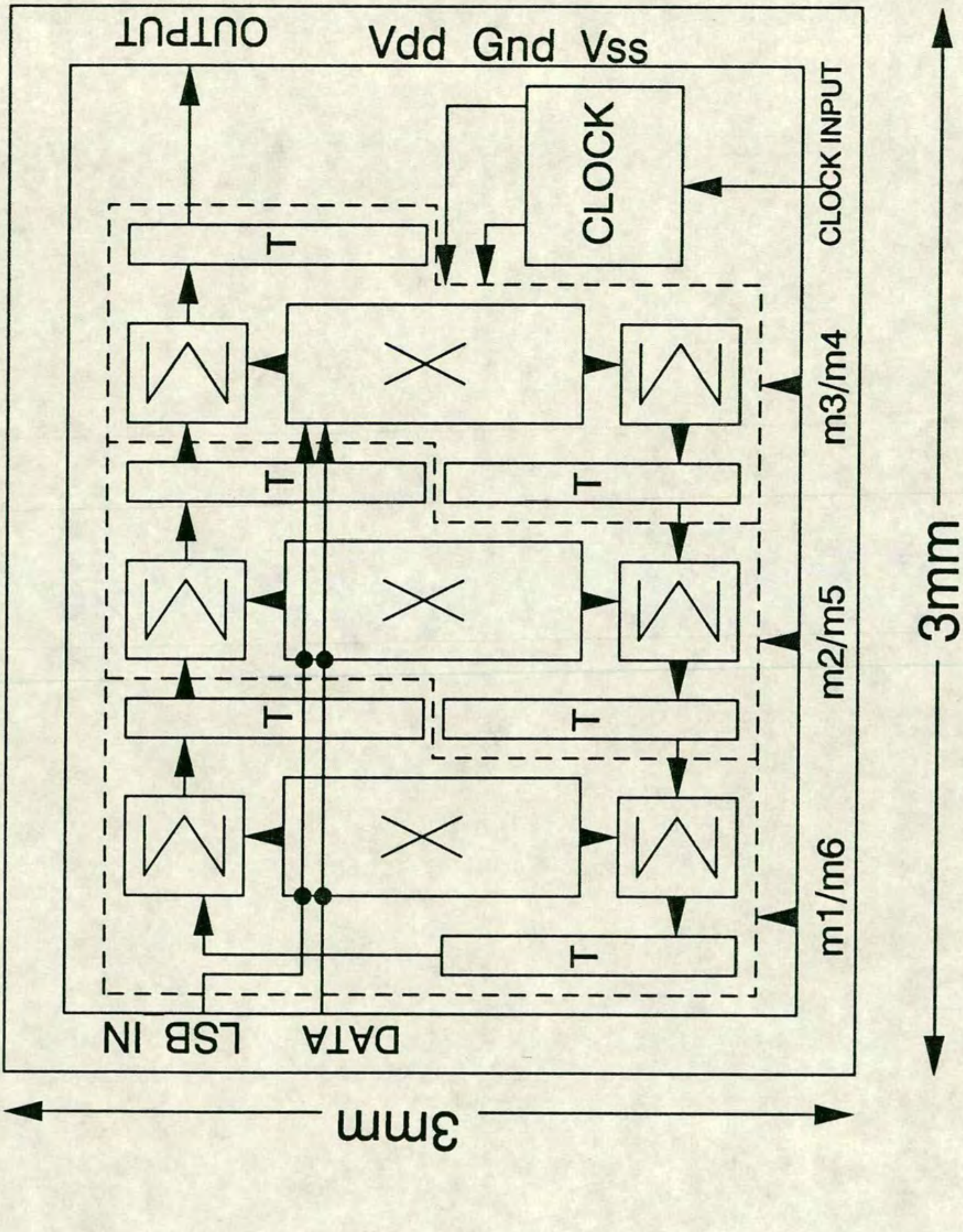


Fig. 5.71 Floorplan of an FIR filter chip using the GaAs cells

5.10. Performance enhancement

The performance of these cells could obviously be enhanced by increasing the clock rate to a predicted rate of 800MHz, by using MODFET technology. Without redesign, the present GaAs cells should work up to 500MHz, allowing for the limitation of simulations. By optimising the layout and clock distribution in an integrated chip, 1GHz clock rate should be possible with a MODFET process, since the loaded gate delay is below 100ps [2]. Above 1 GHz, apart from changes in transistor sizes, interconnection delays will be much more significant and it is likely that the 2-phase clocking scheme will become a limiting factor. Also, matched impedance lines will become mandatory when going off-chip. Single-phase clocking is an attractive alternative, which avoids relative skew of the two phases. It poses its own set of problems in terms of clock loading. Such a scheme is proposed and discussed in Chapter 6.

Other techniques that had been shown to enhance the performance of silicon bit-serial systems, such as twin-pipe and radix-4, are also attractive. These techniques introduce a certain degree of parallelism into the architecture to achieve a doubling or quadrupling in throughput, with a significant (but not directly proportional) increase in operator areas. Because of the limited complexity of GaAs chips at present, functionality may be compromised for system word rate so that the off-chip delays dominate in a system context, as with bit-parallel systems.

Apart from on-chip enhancements, off-chip signals will always be a major problem and worsen if MODFET/HEMT technology is used. In the next Chapter, a novel chip to chip signaling technique is described which can be adapted to the existing GaAs and MODFET cells, and further enhance the advantages of bit-serial circuits. This technique is then expanded towards a limited degree of parallelism.

Chapter 6

Multi-level signaling and single-phase clocking schemes

This Chapter describes two separate schemes that were aimed at enhancing the efficiency of and reducing timing problems in the GaAs bit-serial cells discussed in the last Chapter.

The first scheme is centred on a re-structuring of off-chip signals to increase their efficiency. A multi-level off-chip signaling technique was developed, fabricated, and tested. A brief overview of previous multi-level systems is first given to determine the feasibility of GaAs implementation. Previously published examples of such systems in silicon, and one example of multi-level circuits in GaAs, are used to highlight the main design aspects. Extensions of this technique into higher-radix signal formats are also proposed and discussed.

The second scheme involves the use of single phase clocking to simplify clock distribution and reduce clock skew. The circuit designs are unique in their utilisation of both static and pseudo-dynamic latches, and both D-MESFETs and E-MESFET for logic switching.

6.1. Overview of multi-level signals and systems

It is important to first identify the need for introducing multi-level signals into bit-serial GaAs circuits. The conventional bit-serial system transfers data via one signal wire and one word framing wire, both on-chip and off-chip. In this work, the framing wire carries a pulse one clock cycle wide and coincident with the lsb of the data signal. The adopted convention is shown again in Fig. 6.1. Such signals belong to a class of non-return-to-zero (NRZ) formats where the data bits are consecutive both in state and in level. The output buffer for the framing signal needs to be capable of the same speed as that for the data signal. This is inefficient in space and power because one whole wire and output buffer is devoted to a pulse which only goes high every 8 to 31 clock cycles (depending on the system word length).

Secondly, and more importantly, synchronisation between the data and framing signals are essential for correct system operation. Depending on the bit rate, the distance between chips, and whether board to board communication is necessary,

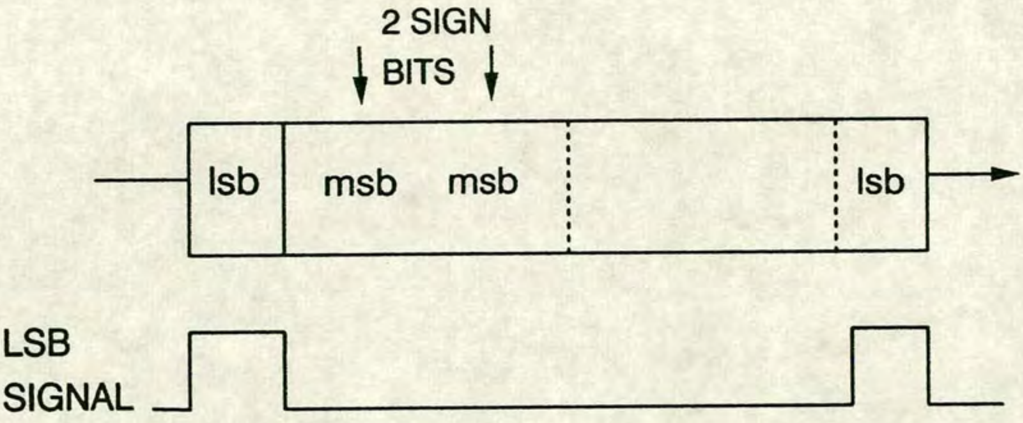


Fig. 6.1 Data and framing signal convention, with two guard bits.

careful line matching and delay equalisation may be necessary.

For these two reasons, it is therefore desirable to develop a scheme such that the two signals are combined into one signal carried on one wire. With such a scheme, reductions in I/O buffer area and power consumption may result, and the synchronisation of the data and framing signals are guaranteed. With improving technology, the latter advantage is particularly relevant, as the bit-rate is increased well beyond the nominal 500Mbit/s of the GaAs cells in this work.

Note that while these advantages would also apply to previous silicon CMOS and NMOS bit-serial systems [123], the efforts involved in combining the signals are largely unnecessary owing to the much lower bit rates and power dissipation of output buffers (especially in CMOS).

In terms of actual implementation, it is common in serial time division multiplexed (TDM) digital communication systems to combine data and framing signals by adding extra "control bits", as well as recoding for unique detection and error correction [131]. NRZ signals are also common in these systems because of their lower bandwidth requirements. This method normally requires significant circuit overhead in extracting the data, typically using registers, counters, and decode logic. In addition, the system word rate is reduced due to the redundancy bits. Such a method is therefore not applicable to bit-serial systems.

The alternative is to introduce a third voltage level for the LSB signal. Noting that there are 2 guard bits available in the adopted system data format, the most significant guard bit (i.e. the msb of the system data word) can be replaced by a third voltage level representing the LSB signal, and then recovered at the receiving end. Although this means that the extra level is now coincident with the msb of the data word, it will still be called LSB for the sake of consistency with the conventional GaAs cells. Its function remains exactly the same as before — to indicate the lsb of the data word — even though it occurs one clock cycle before the lsb of the data word it represents. Details of its implementation are discussed later on.

In a broader context, the advantages of combining the data and LSB signals clearly also applies to multi-level systems in general. If multi-level logic is also adopted, additional improvements in area efficiency and speed can result. A possible drawback of using multi-level signals is yield degradation due to low noise margins between the voltage or current levels in a large chip. These are largely device and process dependent. Incompatibility with conventional DCFL systems also precludes their general usage. The problem of noise margins can be solved by sacrificing speed and increasing the separation between the levels. The overall compromises can easily outweigh the advantages and should be carefully assessed.

There are a number of options that should be considered when implementing a system with composite signals. These include:

(i) Composite signaling on-chip only, off-chip only, or both on and off-chip.

(ii) Current or voltage mode signals.

(iii) Multi valued logic circuit implementation as well as multi-voltage level signaling.

A review of related circuits in silicon and GaAs technologies is useful in identifying the factors affecting these issues.

Silicon implementations of multi-level signals

In silicon MOSFET, on-chip multi-voltage level signals have been used in a ROM design [115]. The primary objective was to reduce the number of cells per word. A 4-level storage cell could store 2 bits of data at a time. This signal was stored and sensed by cells and decoders made up of MOSFET's with 4 different threshold voltages using selective implantation. Since the signals existed on-chip only, these memories could be used as standard components without modifications to other chips. Such a scheme was used in a 128k ROM in this example. The speed and yield of such a device was found to be similar to that of a normal 2-state ROM of similar die size, while the memory density was increased by about 50%. The four voltage levels were between 0V and V_{dd}. Since the data in a ROM is fixed, the noise margin between signal levels is not as crucial. This is not the case if it were used in a RAM, where signals have to be sensed along long lines during read/write cycles, and the circuit design becomes more difficult.

Implementations and design methodologies for multi-level logic circuits have been proposed by others [84,93]. In these instances, it was found to be more convenient to compute and generate the signals in current mode, instead of as voltages. This can be generally called Current-mode Logic (CML). Proposed circuits exist both using bipolar junction transistors (BJT's) and MOSFET's. In the case of BJT's [84], Integrated Injection Logic (I^2L) circuits [90] are adapted for multi-valued logic by having several different current injectors (MVI^2L), as shown in Fig. 6.2. Multi-collector BJT's are used to advantage in reducing chip area. They also have typical transconductances of several thousand, particularly important for current sources where a very low output conductance is desirable. In one MOSFET implementation [93], current injection is achieved using voltage controlled current sources and current mirrors. Since MOSFET's are also symmetrical and bidirectional, they are used to steer currents into and out of logic gates, or act as a buffer to regenerate degraded currents.

Current-mode multi-level logic circuits have some advantages in area over binary circuits. More importantly, currents can be conveniently added or subtracted at one node — not possible with voltage signals — thus allowing efficient implementation of arithmetic operations. The main disadvantage is that currents are split into two halves if used to drive two logic gates. Therefore, the fanout of current-mode logic gates is restricted to 1.

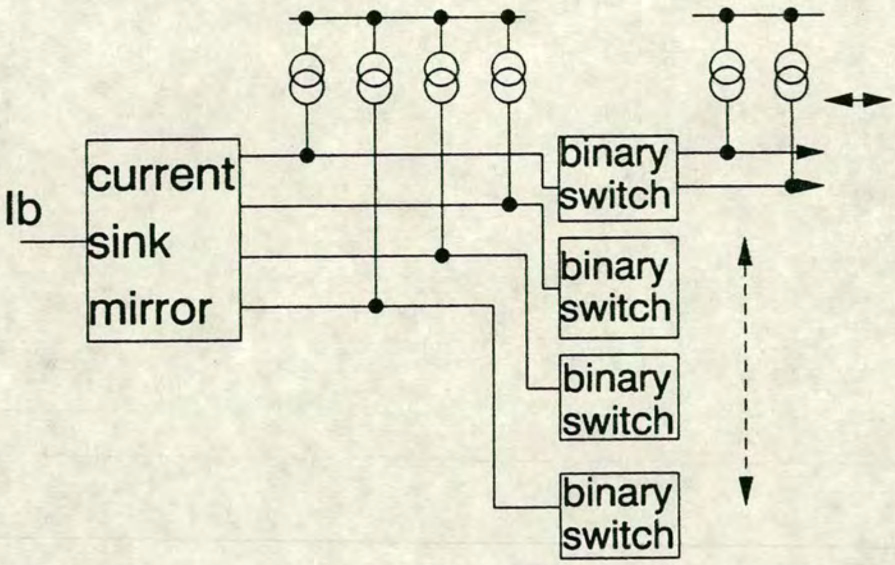


Fig. 6.2 Multi-valued I²L.

Other multi-level signals take the form of composite analogue signals such as video signals, and tri-state logic which has a high impedance state. The former is of more relevance here since the high impedance state is not a valid detectable state. In video signals, a square synchronising voltage pulse is added to the analogue video signal at regular intervals to frame the horizontal scanning of the television [131]. The next section looks at an example of multi-valued logic in GaAs, and conclude as to the feasibility of implementing current and voltage-mode multi-level signals.

Feasibility of GaAs multi-level circuits

First consider implementing current-mode signals in GaAs. Current sources and mirrors used in these circuits ideally should have a very high output resistance. In the current sources commonly used in ICs, such as the compact three transistor Widlar-based current source [90] shown in Fig. 6.3, the output resistance of the mirror is equal to the output resistance of the output FET device. Compared to bipolar devices, FET devices have relatively low values of small signal output resistance, R_{ds}, owing to the channel-widening effect (Chapter 4). This applies both to the silicon MOSFET and the GaAs MESFET. Therefore they are not as good for current sourcing as bipolar transistors.

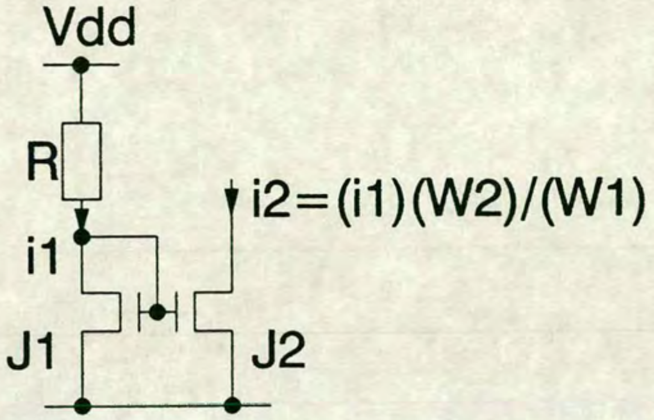


Fig. 6.3 Widlar-based current source using FET devices

To increase R_{ds}, heterojunction devices (MODFET/HEMTs and HBTs) can be used to advantage, as used in a number of examples of dynamic frequency dividers in the form of emitter/source coupled pairs, one of which operated up to 34.8GHz [156]. Alternatively, R_{ds} can be increased by using the MESFET versions of the Wilson and cascode current mirrors [90], at the expense of increased size and power. Since CML circuits in whatever form require a current source in every logic gate (with several different values in multi-level circuits), these device aspects contribute to the fact that CML has not been implemented in GaAs MESFET technology.

In the voltage mode, one example of multi-level signals in GaAs was proposed in [142]. Based on a different multi-valued algebra from the McClusky proposal [84], the signals were generated and computed directly with specially designed DCFL-like logic gates. Each gate performs a "literal" function, and in theory can be extended to any fixed radix. The major drawback of this technique, as in the silicon ROM, was the need for having a different MESFET threshold voltage for each of the logic voltage levels. Different supply rails were also necessary for the voltage levels. With the present level of threshold voltage control achievable in GaAs processes (Chapter 4), the noise margin fluctuations of a system with 4 or more different V_t values over a large chip area is likely to result in lower yield. Standard deviations of V_t values increases with the number of implant steps (as illustrated by the larger deviations for the D-MESFET compared to the E-MESFET in the Honeywell process, discussed in Chapter 4). Also, their fluctuations do not necessarily cancel each other. Together with the extra number of masks and process steps, and associated yield hazards, the cost of such chips will be substantially increased. The extra number of voltage rails also make board level packaging and decoupling more complicated. These tend to offset the advantages of multi-level logic in terms of circuit area efficiency. However, when many voltage levels are involved, multi-thresholding is unavoidable in order to maintain a high logic density.

From the discussions above, it can be observed that whether in voltage or current mode, the electrical properties and the processing of GaAs MESFET's largely determines the overall merits of implementing multi-level circuits, and that large number of levels are unlikely to be efficient and cost-effective in the GaAs MESFET technology as it stands. However, by limiting the number of levels and carefully choosing the voltage levels to minimise circuit overhead, the 3-level generation and detection circuits in this work provide a compromise between multi-values logic and normal binary signaling. This is discussed in the next section.

6.2. The off-chip 3-level signaling scheme

Returning to the bit-serial data format, it was established earlier that the msb of the system data word can be replaced by a third voltage level to represent the LSB signal because it represents a redundant sign bit†. With such a scheme, the system data word is totally unchanged and retains the normal DCFL levels off-chip. The only decoding circuit necessary at the receiving end is to detect the third level to produce the LSB signal, and recover the sign bit at the msb position. After detection, the two signals are then compatible with the bit-serial circuits described in Chapter 5, without the need for complicated decoding. The very important choice of the third voltage level essentially depends on the simplicity of generation and detection, and the noise margins between the levels. This merits some discussion.

† Co-published work by the author.

In silicon n-channel MOSFET devices, the substrate must be tied to the most negative part of the IC to prevent channel/substrate forward biasing. Therefore any additional voltage levels have to be between the MOSFET threshold voltage and the V_{dd} rail, as was the case in the multi-state ROM [115]. In GaAs, the substrate is semi-insulating and is normally grounded, except in circuits where sidegating/backgating is highly critical due to large negative voltages (Chapter 4). Therefore negative voltage is a viable option. Overall, there are three possible ranges for the third voltage level: within, above, or below the normal DCFL logic swing. Consider each range in turn.

For the voltage range within the DCFL logic swing, i.e. between 0.1V and 0.7V, unacceptable degradation of the noise margins will clearly be inevitable. This option can be definitely ruled out.

For voltages above the normal logic HIGH, i.e. greater than 0.7V, the Schottky diode drop can be used as a convenient reference step. With 3 voltage levels, multi-threshold voltage MESFETs can in fact be avoided. The diode drop is used often in GaAs (apart from its clamping effect on logic swing), as in Schottky Diode FET Logic (SDFL). Since only one extra level is required in this instance, consider using the nominal voltage of 1.5 V (2 diode drops).

To generate this third voltage, the circuit shown in Fig. 6.4 may be used. This circuit uses the existing GaAs V_{dd} supply rail, and all E-MESFETs have the normal V_t of +0.2 V. The conditions for the three output voltages are:

(i) +0.1V — E-MESFET pull-down ON — output = $\overline{\text{DATA}}$.

(ii) +0.75 V — both pull-up and pull-down E-MESFETs ON — output = LSB.

(iii) +1.5 V — pull-up E-MESFET ON — output = $\overline{\text{DATA}}$.

A major drawback of this circuit is the very high drain/source currents in the output devices for generating the DCFL-equivalent 0.75V level, when both the pull-up and pull-down devices are ON.

Unique detection of the 1.5 V level and the normal DCFL levels requires two detector gates in parallel, producing a unique 2-bit binary word for each level. This can be achieved by two of the four possible circuits designed for this task, shown in Fig. 6.5. Their input/output states for the 3 input voltage levels are also shown. Note that all four circuits do not clamp the input voltage to one diode drop in order for them to be able to detect the +1.5V level. By using either the top 2 circuits or the bottom 2 circuits in the Figure, three unique 2-bit words are obtained for the three voltage levels. (Simulations of the generation/detection circuits will be presented in a later section.) Note that the normal single diode input protection cannot be used with this high level. The static power dissipation of the output driver alone is about 40 mW. If this circuit is used in a 3-level system, the power advantage of combining the 2 wires into 1 is small compared to two 2-level output buffers, and the extra area needed for thermal dissipation of the high current devices would also reduce the area advantage. Now consider the third voltage range.

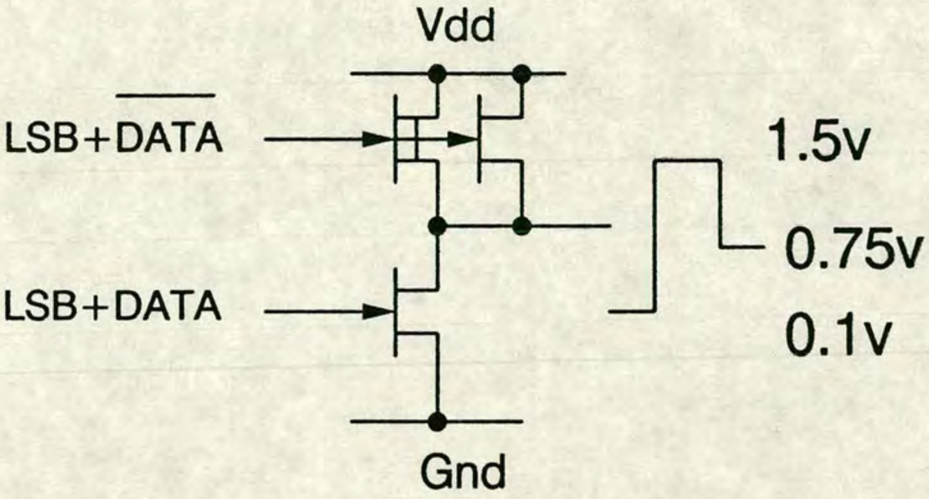


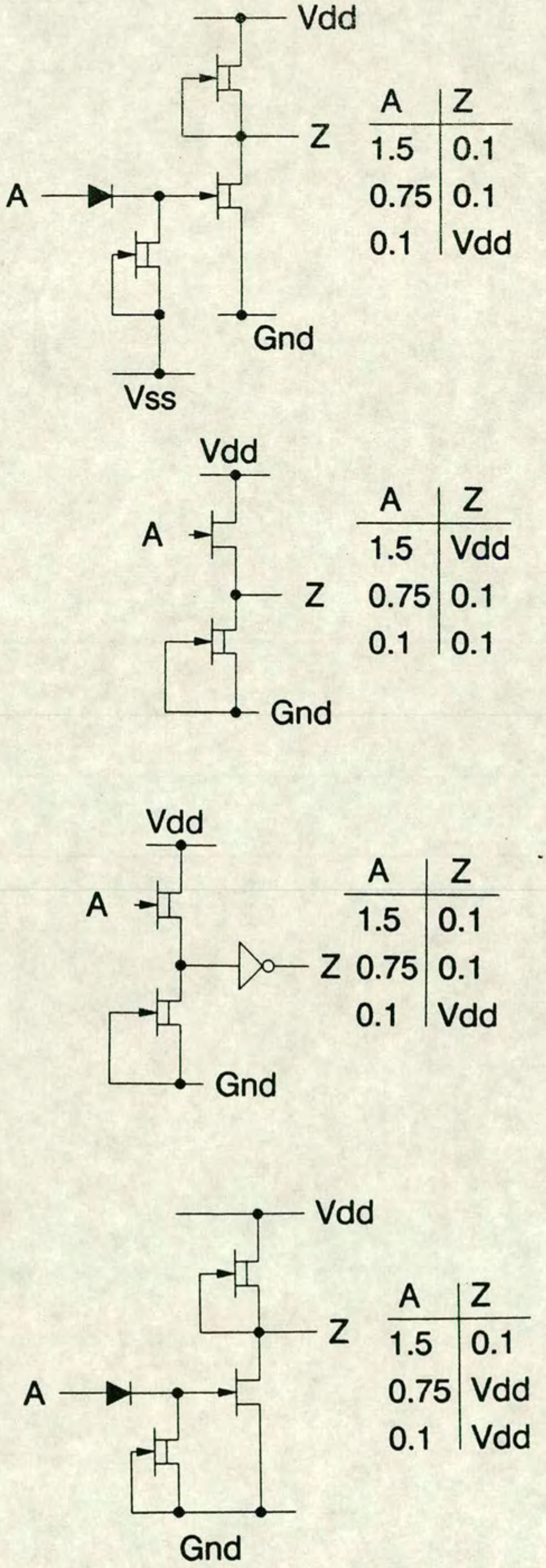
Fig. 6.4 The circuit for generating
a 1.5 V 3-level composite signal

With the third possible voltage range, i.e. below the normal logic LOW, a negative voltage rail is necessary, and problems of sidegating/backgating need to be considered. Since the −0.5 V level has already been used in the GaAs cells discussed previously, this is the convenient choice in this range. Negative voltages are widely used in Buffered FET logic circuits in GaAs (Chapter 2). Sidegating is not a problem with this voltage with the Honeywell process and this choice should be further considered.

To generate this −0.5V voltage, the normal 2-level output buffer can be modified by adding an extra pull-down device and three level shifters, as shown in Fig. 6.6. The existing −0.6 V rail (in the on-chip clock generator) can be used. The level shifters are used to drive each of the output devices to ensure that only one output MESFET is ON under static conditions. This minimises power dissipation and avoid the high current conditions in the 1.5 V circuit (Fig. 6.4).

At the receiving or decoding end (Fig. 6.7), a D-MESFET can be used in a specially ratioed DCFL-based inverter. This inverter produces a DCFL logic HIGH output only when the input is −0.5 V or less. Beyond this D-MESFET inverter, all signals have the normal DCFL swing. Fig. 6.8 illustrates, with idealised signals, the generated and detected outputs for a data word example of 11001. Note that the two guard bits are not restored with the circuit in this diagram for clarity. The guard bit can be restored by the circuit in Fig. 6.9.

From this discussion, it is apparent that the "negative" range of the workable GaAs voltage spectrum is best suited to the 3-level off-chip signaling scheme. The resulting circuits should not compromise throughput rate, and have all of the advantages of combining the LSB and DATA signals, and do not require any additional process or device steps and supply rails other than the existing ones. More details are discussed in the next sub-section.



A	Z
1.5	0.1
0.75	0.1
0.1	Vdd

A	Z
1.5	Vdd
0.75	0.1
0.1	0.1

A	Z
1.5	0.1
0.75	0.1
0.1	Vdd

A	Z
1.5	0.1
0.75	Vdd
0.1	Vdd

Fig. 6.5 Four possible detector gates for +1.5V 3-level circuit

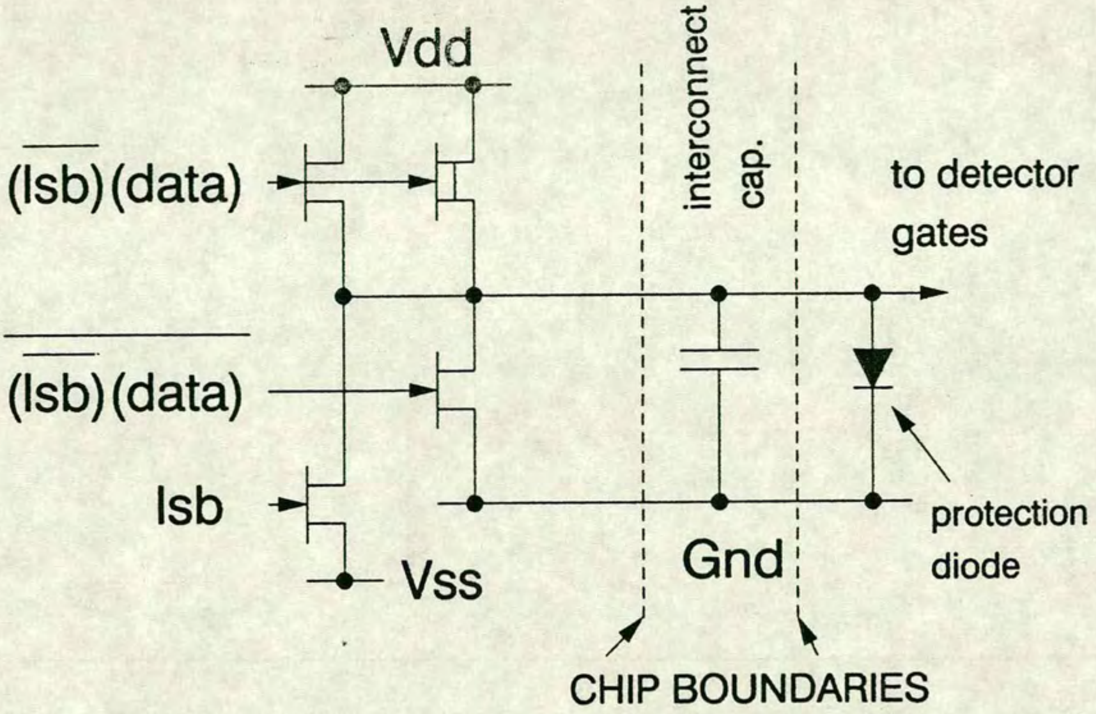


Fig. 6.6 3-level generating circuit
for a −0.5V voltage level.

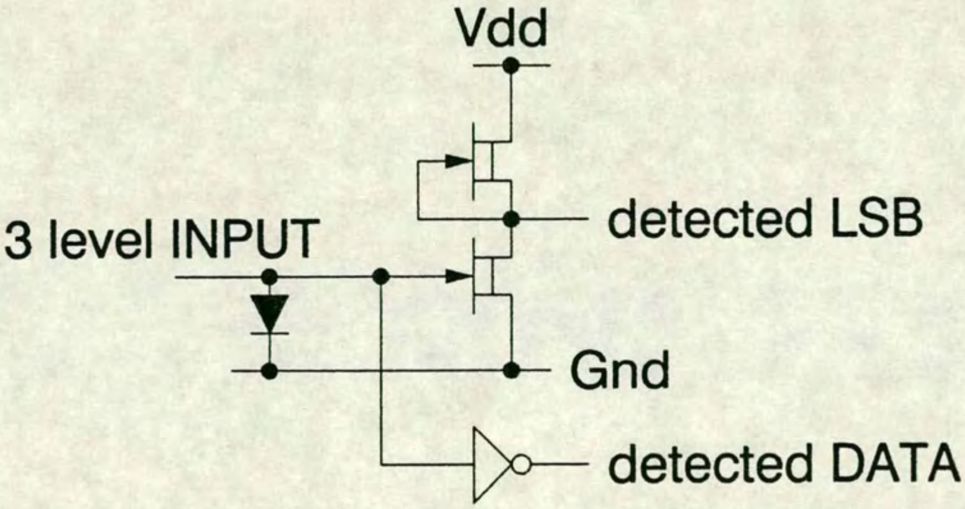


Fig. 6.7 3-level detecting circuit for
−0.5V level.

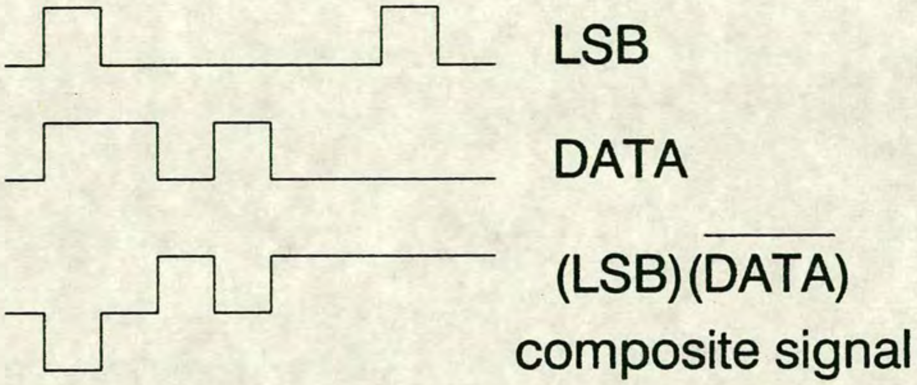


Fig. 6.8 Idealised example of composite signal format

The circuit details and simulation results

Having adopted the −0.5V as the third level, the circuits in Fig. 6.6 & 6.7 were fabricated. Minor modifications were made to the circuits for a MODFET mask set. These are mainly in the level shifting stage preceding the output buffer. A single E-MESFET device was used for the pull-up device in the output buffer. Since the modifications are minor due to the similar structures between the MESFET and MODFET, further discussions shall concentrate on the MESFET.

Timing aspects of off-chip signaling of the 3-level circuit were identical to that of the 2-level system. The I/O buffers were pipelined with pseudo-dynamic latches. The output buffer used a combined E/D-MESFET pull-up device, and a large input protection diode was used as before.

The logic which combines the LSB and the DATA signals is simple and self-evident from the diagram. The sizing of the NOR gates, inverters, level shifter, and the output MESFETs were designed for a capacitive load of 3pF. Most of the power is dissipated in the output devices and they need to be distributed and placed as close to the pads as possible.

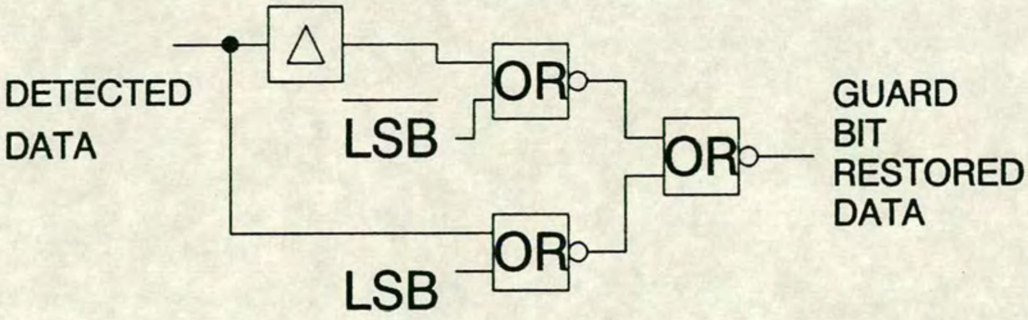


Fig. 6.9 The guard bit restoring circuit

The inverter that detects the −0.5V signal uses a D-MESFET pull-down device. The ratioing of this inverter is such that when the input voltage is 0V or higher, the output is 0.1V (DCFL LOW). The noise margin between the 0V and −0.5V levels is important. Simulations have found that the depletion inverter (for detecting the −0.5 V) has a noise margin of approximately 300mV for the logic LOW level, similar to that of conventional DCFL inverters.

The 3-level circuit was designed for the nominal clock rate of 500MHz, i.e. a data bit rate of 500Mb/s The SPICE simulation results are shown in Fig. 6.10 & 6.11. The input file is included in Appendix 11. The latency of one clock cycle due to pipelining can be seen from the curves. Note that when reading the results, the signals are lsb first. The data bit rate was 700Mb/s. A 5pF capacitor was used to represented between-chip interconnect. Correct detection and latching were maintained when simulated at this, higher than required, clock rate. The total static power dissipation of this circuit was approximately 29mW, as compared to (2 × 22.3mW) for the conventional 2-level system. Estimated layout area is 30% less than the previous system. Including the guard bit restoring logic, the overall area would be about the same, while power dissipation would still be significantly lower because of the single output buffer.

Test results

The 3-level encode and decode circuits were fabricated on MODFET mask set. The layout of the generating and detecting circuits are shown separately in Figs. 6.12 & 6.13. The decode circuit did not include the guard bit restoring logic. The low speed test showed perfect functionality in the encoding circuit. The LSB signal was successfully recovered from the decoding circuit. A layout fault meant that the DATA signal could not be observed off-chip. If scanning electron microscope was available for debugging, it should show that the DATA signal was decoded correctly on-chip. Full-speed testing will need to be done with the chip bonded and packaged, and these results are not yet available. More details of the test results are given in Appendix 15.

6.3. Extensions of the 3-level scheme

The 3-level circuit and its test results have demonstrated that multi level signals are feasible. The main objective had been to ensure the synchronisation of the data and framing signals. This section proposes other schemes which build upon these results, and considers other possible uses of multi-level signals, which may or may not relate to that of the 3-level circuit.

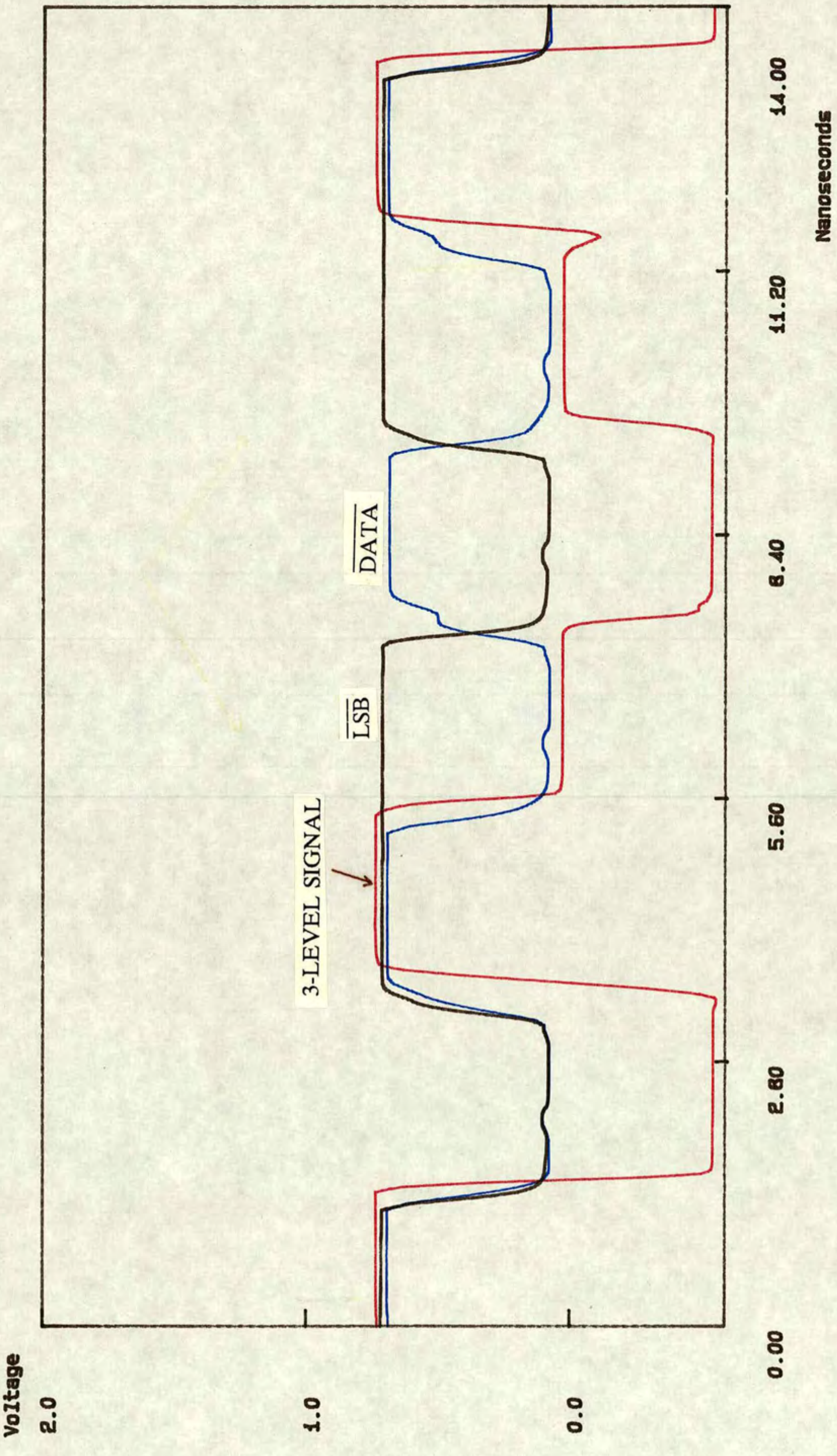


Fig. 6.10 Simulation of 3-level generation (−0.5V).

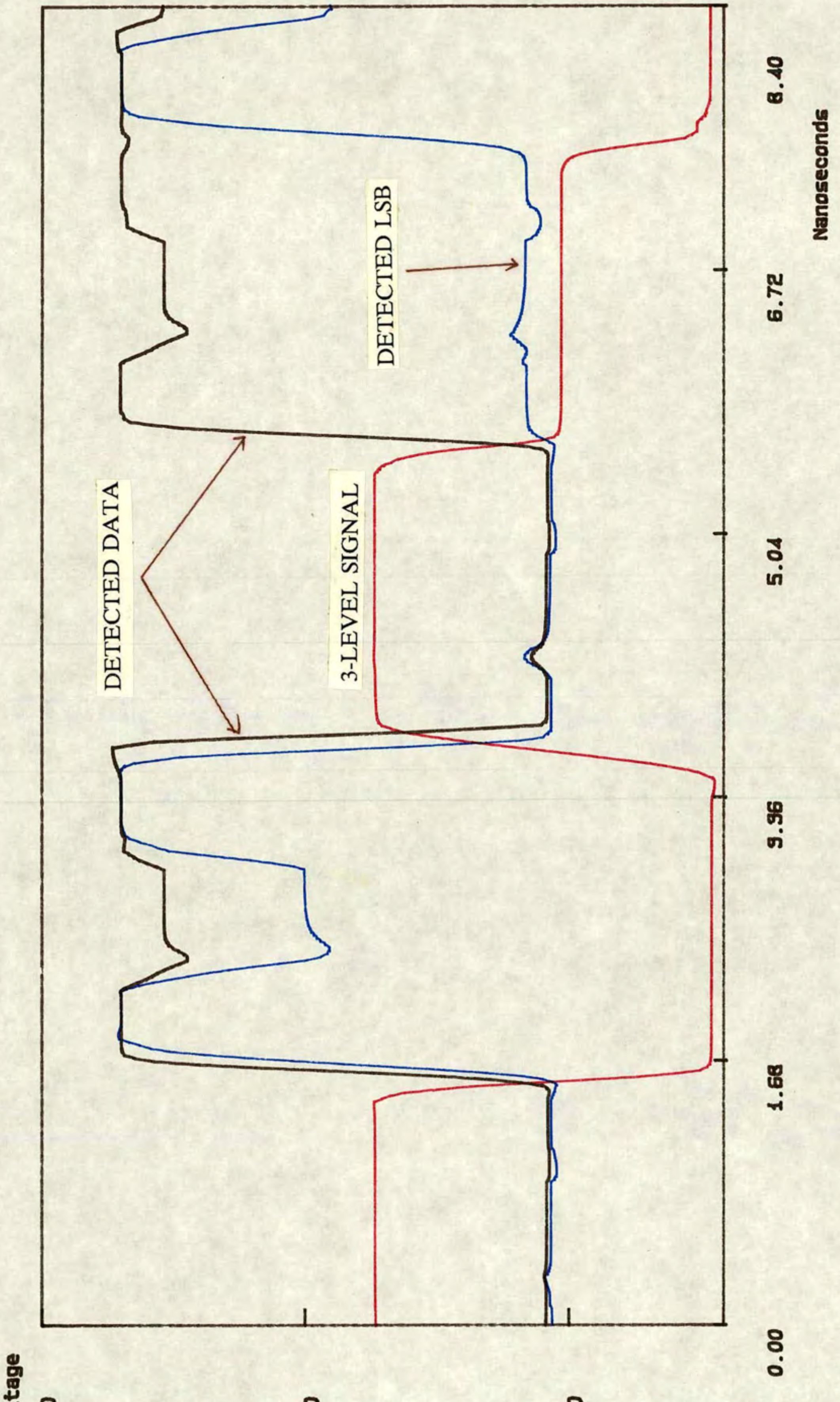


Fig. 6.11 Simulation of 3-level detection (--0.5V).

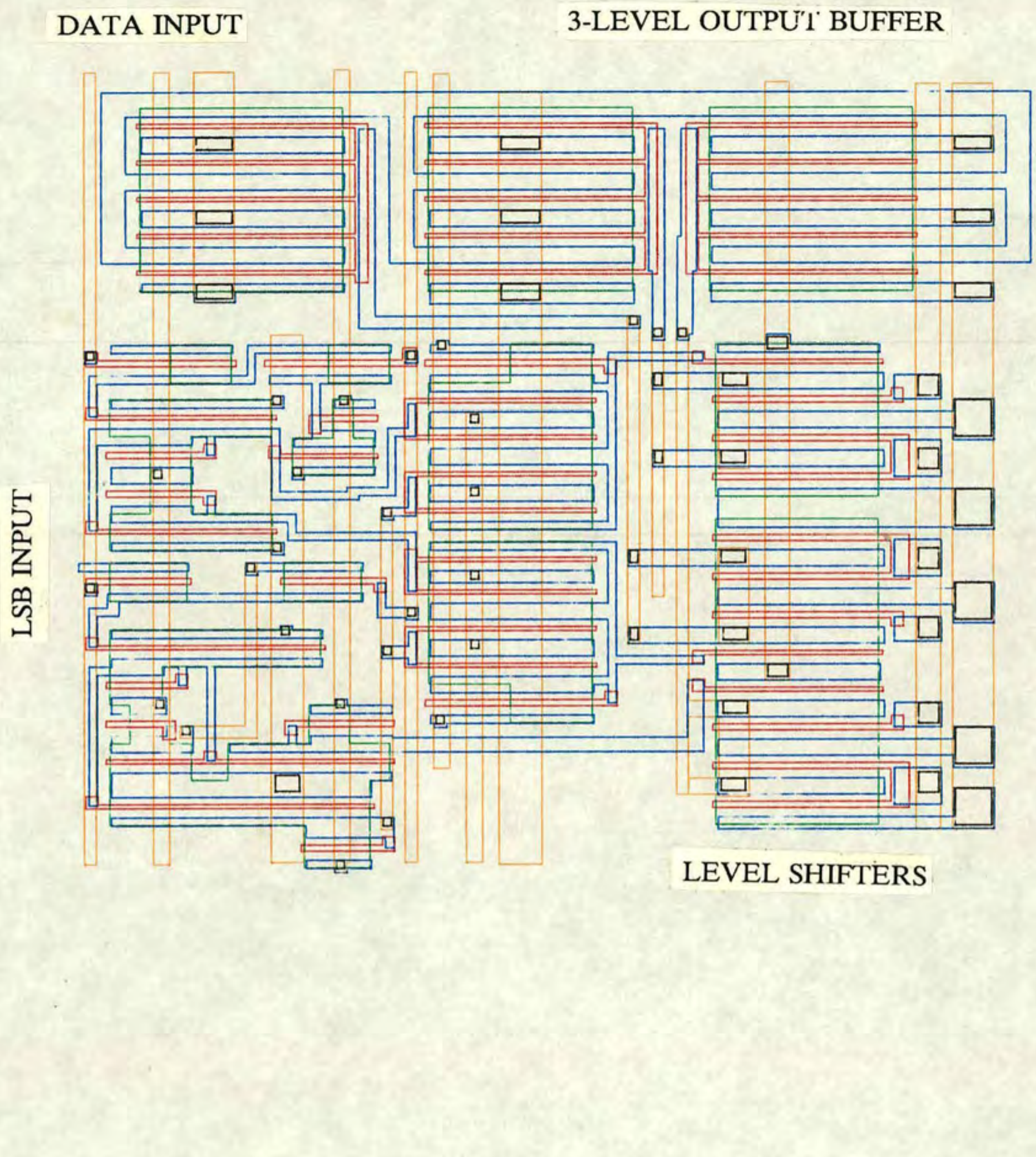


Fig. 6.12 Layout of 3-level generating circuit

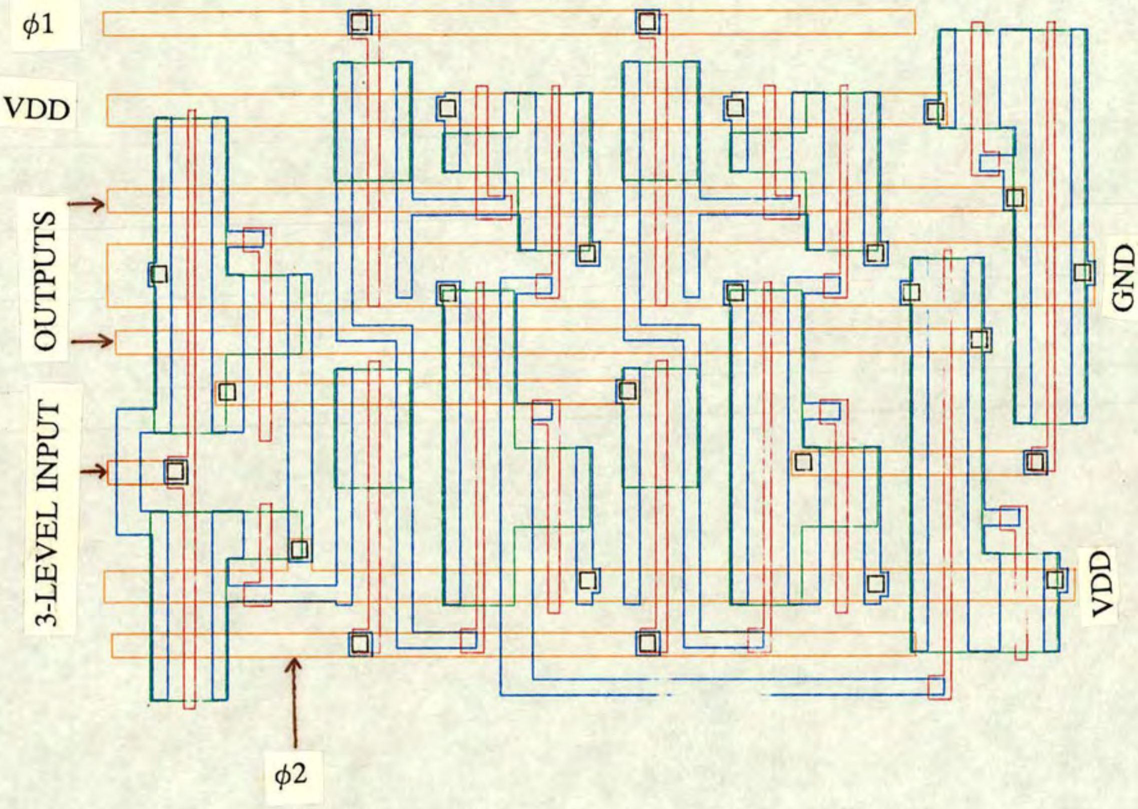


Fig. 6.13 Layout of 3-level detection circuit

6.3.1. 4-level off-chip signaling

In an earlier section, it was established that multi-level logic circuits in voltage mode have yet to be proven by practical results, and are unlikely to be practical. Also, current-mode logic is not as attractive in GaAs MESFET technology as in silicon bipolar technology. Therefore, the discussion will further concentrate on off-chips signaling only. A 4-level (voltage) signal is a logical extension, and is considered here.

One possible use of a 4-level signal is to represent 2 bits of data at a time, i.e. radix-4. (Here the word "radix" is used in the sense of binary numbers, not as in fast Fourier transform machines.) The significant advantage of such a system is the halving of the off-chip data transition rate, making it easier to propagate and less prone to timing and logic state errors, such as over longer distances on or between circuit boards. This is particularly attractive for the GaAs MESFET technology due to the penalties of high current buffers, and for faster technologies such as MODFET/HEMT. Off-chip interconnect capacitances do not scale as on-chip technology improvements. The definite reduction in dynamic switching power is therefore very relevant. For example, even at a clock rate of 2 GHz, the data transition rate would still be a relatively low 1000 Mb/s, if a 4-level signal was used.

The major disadvantages are the circuit overhead needed for encoding and decoding, and the larger propagation delay. Although it may be possible to pipeline the logic and buffers such that the throughput rate is not sacrificed, the extra delay is still detrimental in a recursive system, because it needs to be accounted for in the system word length.

An additional drawback arises from the data/framing signal requirement of this bit-serial work. A 4-level signal used in the radix-4 mode cannot be used to represent the LSB signal as well. The LSB signal will need to be carried on a separate wire as in the conventional system. Therefore overall, using one 4-level signal is not as attractive as a 3-level signal. A 4-level signal may be more useful in combination with other multi-level signals, as will be seen in a later section. The circuit design aspects of generating a 4-level signal are first considered in some detail.

In the 3-level signaling discussions, it was decided that the high voltage level of 1.5 V can be used as an extra level if necessary, but with the drawback of very high static power dissipation for the 0.75 V level. The output buffer circuit (Fig. 6.4) can be combined with the -0.5 V 3-level circuit to generate 4 levels, i.e. 1.5 V, 0.75 V, 0.1 V, and -0.5 V. These will also be denoted as $+1$, 1, 0, and -1

With the previous forms of the source following shifter (Fig. 6.15), the voltage swing of the input is +0.1 V to V_{dd} (assuming there is no clamping from a DCFL gate). This voltage is then normally shifted by dropping across one to three diodes, depending on its specific purpose. The E-MESFET pull-up device is OFF when the input is LOW, so that the DC power of the shifter is minimised. When a output current drive is necessary, a D-MESFET can be used. The voltage gain inherent to this circuit configuration is less than or equal to 1.

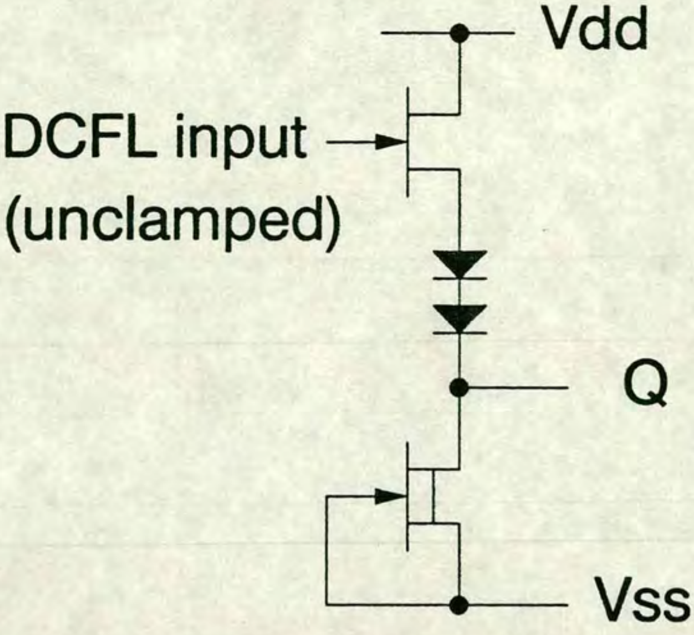


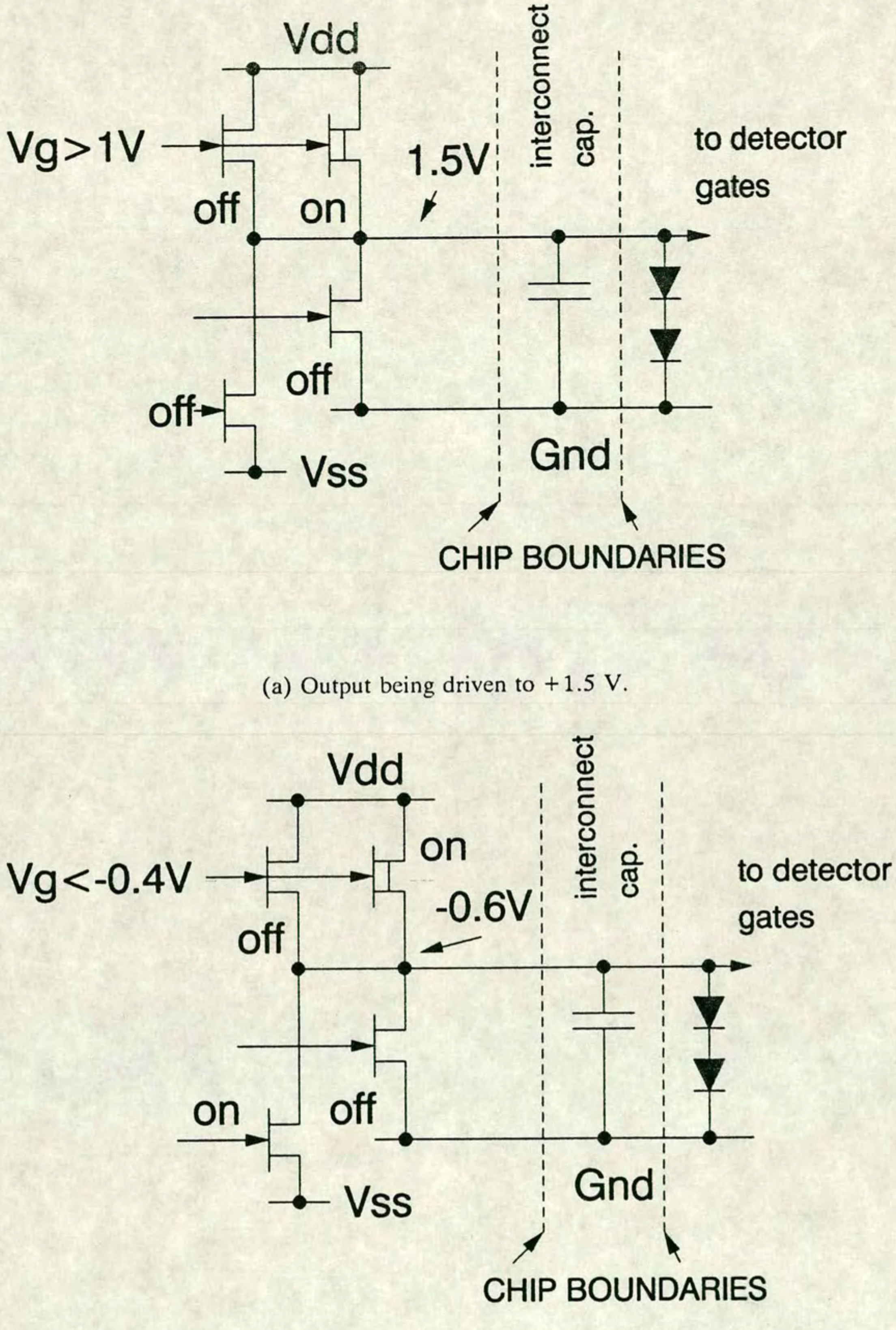
Fig. 6.15 Source follower level shifter.

In the case of the 4-level circuit, the output from the shifter needs to be able to swing from −0.5 V to +1.5 V in order to drive the pull-up device (Fig. 6.14) appropriately. This requirement can be best understood by referring to Figs. 6.16(a) & 6.16(b).

In Fig. 6.16(a), the shifter is attempting to drive the output pull-up devices so that common source voltage rises to the +1.5 V level. This is possible only if V_{gs} of the D-MESFET pull-up device is larger than −0.5 V, the V_t of the D-MESFET, so that it is still ON. Therefore, even a single diode drop from V_{dd}, normally +1.7 V, at the gate of the D-MESFET will turn it OFF. This implies that the level shifter must produce a HIGH output of close to V_{dd}. For the pull-down devices in the 4-level output buffer, this is not necessary.

In the opposite situation, Fig. 6.16(b), the output voltage is being pulled down to −0.5 V and all the other devices should ideally be OFF. Therefore the level shifter LOW output needs to drop to −0.5 V. The D-MESFET pull-up device is still ON under this condition, and necessitates proper ratioing between the pull-up and pull-down devices.

This magnitude of the ideal desired swing is therefore 2.3 V (=1.7+0.6 V). Since the output is clamped to +1.4 V, a swing of 2 V is necessary in practice. This is difficult to achieve with the conventional source following level shifter, because the necessary voltage gain is much larger than 1. One solution is to replace the E-MESFET pull-up with a D-MESFET and no diode drops, as shown in Fig. 6.17. Again assuming that the shifter is driven by single fanout DCFL gate, the output voltage is now capable of rising to V_{dd} in the HIGH state. But since the pull-up is always ON, the pull-down device will need to be very large for the LOW output of −0.5 V. Also, when the input is at +0.1 V (DCFL LOW), the pull-down device needs to pull against the increasingly forward biased pull-up as its source voltage falls. The static power dissipation is clearly very large and the efficiency of such a level shifter is very low, as in class-A amplifiers.



(a) Output being driven to +1.5 V.

(b) Output being driven to −0.5 V.

Fig. 6.16 Two conditions of 4-level output circuit

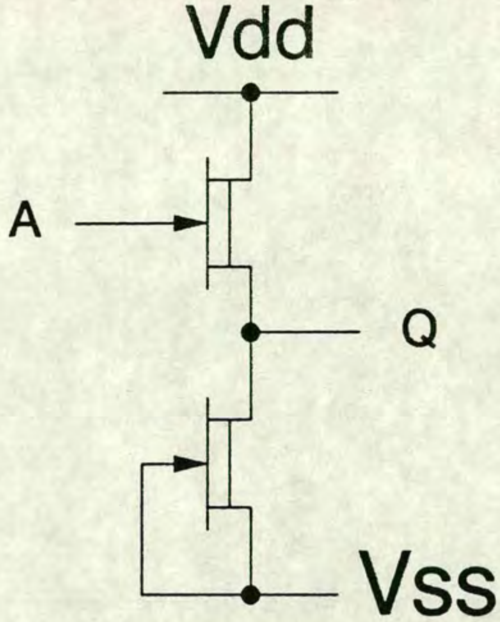


Fig. 6.17 Level shifter with large output swing

In order to avoid these problems, an inverting configuration can be used, as shown in Fig. 6.18. This circuit has some similarity with the D-MESFET Schottky diode FET logic (SDFL), described in Chapter 2. But unlike SDFL, the level shifting is achieved by a DCFL-like inverting output stage, which has the pull-down device connected to −0.6 V, rather than 0V. In order to provide the correct voltages to switch the E-MESFET, a small switching diode and a D-MESFET load are used as shown. Assume that this circuit is driven by a unity fanout DCFL logic gate. When the DCFL gate output is LOW, the D-MESFET load pulls down to −0.6V, and turns the switching E-MESFET of the level shifter OFF. The output of the level shifter is capable of rising to V_{dd}, if it is not clamped by any diodes. When the input voltage is HIGH, then the output of the shifter is pulled down to −0.6V. Therefore this configuration has a maximum swing from V_{ss} to V_{dd}.

In terms of ratioing, the DCFL logic gate driving this circuit needs to be modified. Since the switching diode is always forward biased, and the series connected D-MESFET load is always ON, the β_R $((W/L)_{pull-down}/(W/L)_{pull-up})$ ratio of the logic gate needs to be larger than the normal value of 3.

In terms of power dissipation of the level shifter, since the output stage MESFETs are both ON only in the LOW (−0.5 V) state, most of the power is dissipated in the input DC path formed from the DCFL gate-diode-load connection.

The simulation results of this level shifter is shown in Fig. 6.19. The power dissipation is about 11.3mW, including that of the DCFL gate.

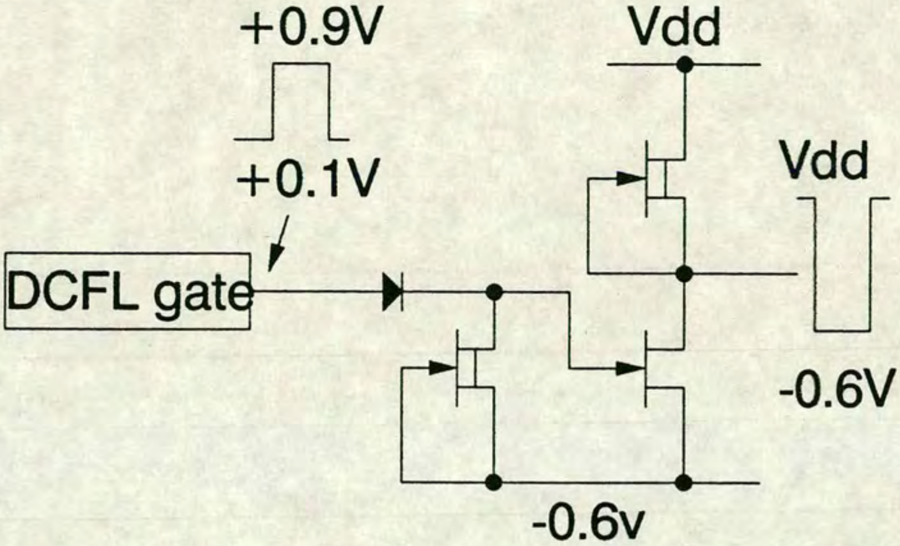


Fig. 6.18 Improved level shifter with gain

A variation of this shifter configuration is to use the same inverting output stage, but buffer its input with a conventional source following shifter, as shown in Fig. 6.20. This has the advantage of eliminating the aforementioned permanent DC path. The disadvantages are the extra propagation delay through the two stages, and that the DCFL gate is restricted to a fanout of one because a full swing (larger than +0.1 V to +0.75 V) is required. In this application, the additional delay is not acceptable, although the power dissipation is lower. Therefore the first circuit is preferred.

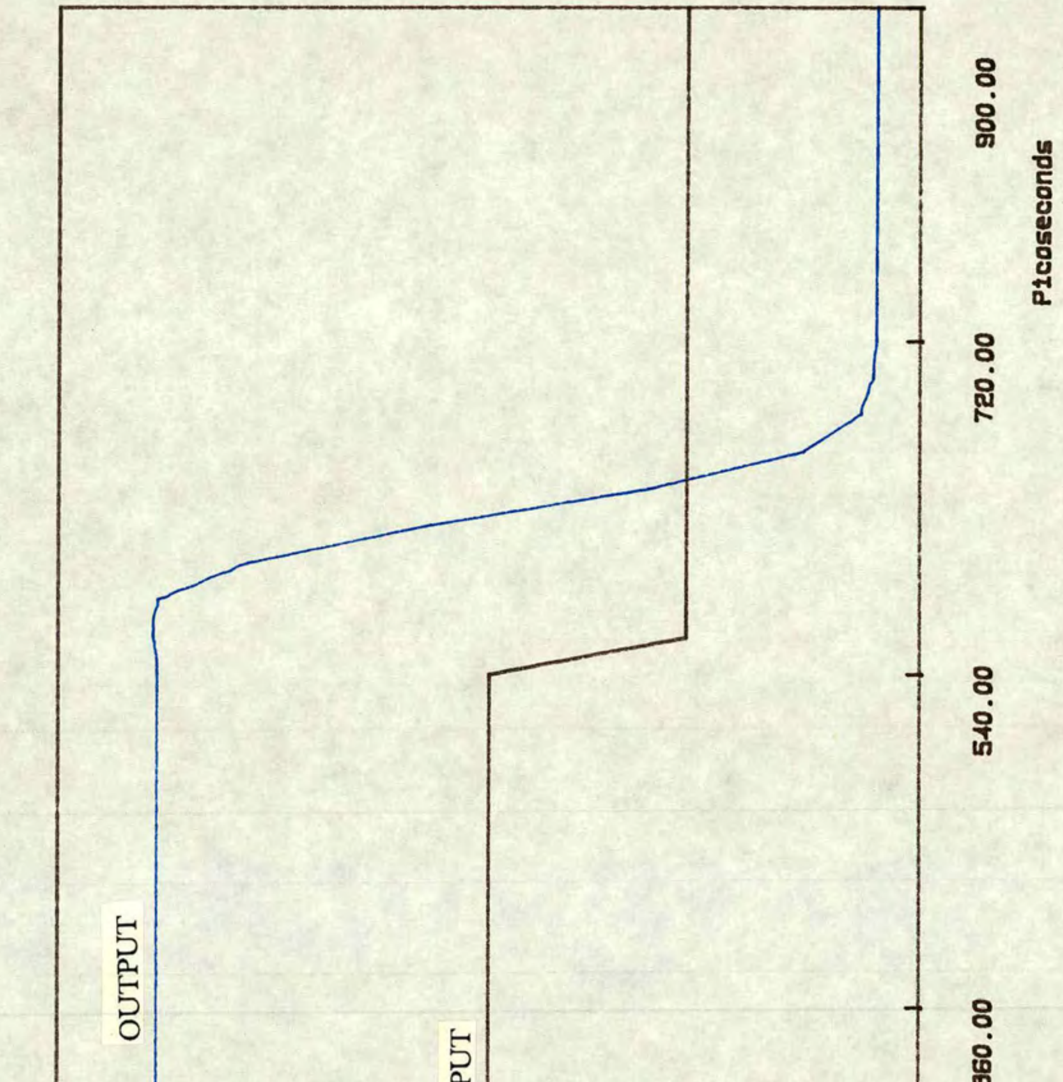


Fig. 6.19 Simulation results of improved level shifter

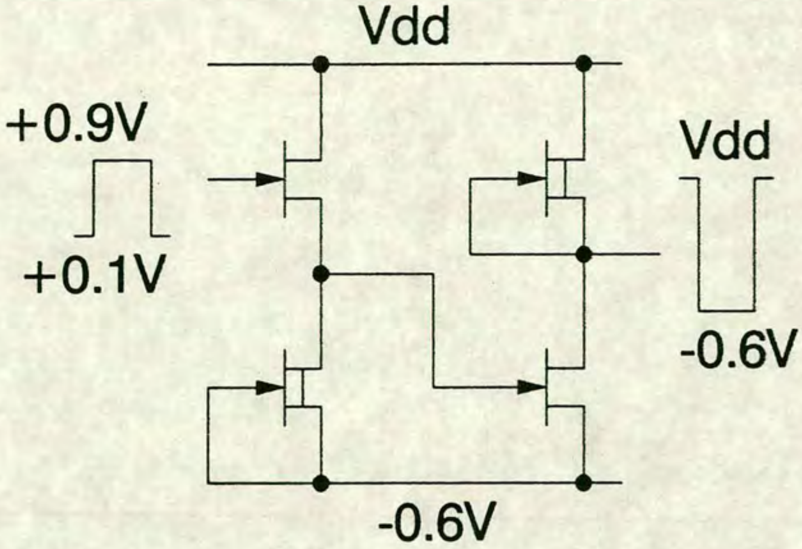


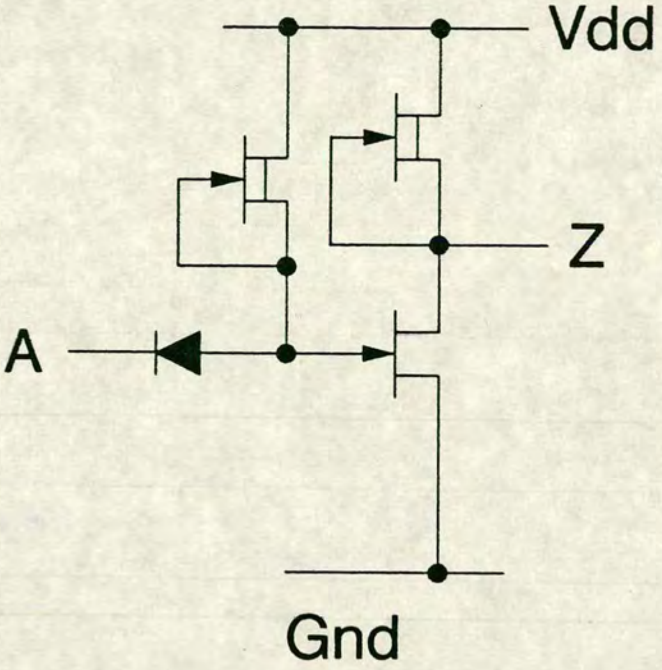
Fig. 6.20 Two-stage level shifter with large swing

4-level signal detection

The task of detecting a composite 4-level signal is not as simple as in the 3-level cases. In each of the previous 3-level schemes (both the positive and negative versions), the detection is done by two gates which detect the extra level and the normal DCFL levels separately. They produce two outputs which can then be used directly, or decoded with normal DCFL logic. This is only workable provided that the operations of the two detector gates are independent of each other. With a 4-level signal and its large voltage range (−0.5V to +1.5V), the two detection circuits for +1.5V and −0.5V cannot simply be combined together and used to provide unique detection of all 4 levels. This is because the gate that detects the −0.5V level will clamp the input signal to +0.75V. Therefore, a different circuit is used to detect the −0.5V signal, as shown in Fig. 6.21.

To obtain 4 unique three-bit codes from the composite signal, this circuit should be used in conjunction with two of the detector gates discussed for the +1.5V 3-level circuit (Fig. 6.5). The function of this circuit is to detect the −0.5 V level, using a reversed biased diode in series with an inverter. This avoids the need for multi-threshold MESFETs. Its operation relies on the correct biasing of the input to the DCFL inverter. If the input signal is +1.5 V or +0.75 V, the diode is OFF and the inverter input is at +0.75 V, producing a DCFL LOW output. If the input signal is

+0.1 V, then the diode is slightly forward biased, still maintaining a +0.75 V into the inverter. When the 4-level signal falls to −0.5 V or less, the diode is forward biased and pulls down the inverter input to LOW, producing a HIGH DCFL output. The ratioing of the inverter needs to be reduced from the normal value of 3 to take into account the weak LOW input voltage.



A	Z
1.5	0.1
0.75	0.1
0.1	0.1
-0.5	Vdd

Fig. 6.21 Detector gate for detecting −0.5 V in the 4-level scheme

One possible combination of the decoded 3-bit code is listed in Table 6.1 below. The simulation results of the 4-level generation and detection circuits are shown in Figs. 6.22, 6.23 & 6.24. Their input files are given in Appendix 12 & 13. Again, the subsequent use of this code depends on what the 4-level signal is used to represent. The overall propagation delay of the detection/generation circuitry and the encoding/decoding logic will clearly affect its use within a bit-serial system. These related issues are discussed in the next section.

Voltage level	Z1	Z2	Z3
+1.5	1	0	0
+0.75	0	0	0
+0.1	0	1	0
−0.5	0	1	1

Table 6.1 4-level decoding scheme

6.3.2. Higher radix representations using multi-level signals

In this section, ways of making use of combinations of 2-level, 3-level, and 4-level signals to further enhance the advantages of bit-serial techniques in GaAs are discussed.

The primary aim of utilising off-chip multi-level signaling is to make more efficient use of signal buffers and lines, in order to alleviate the limitations of the MESFET and problems arising from the high bit-rates involved in bit-serial systems. To extend this concept further, 2 separate wires each carrying multi-level signals can be used to represent a higher-radix data word, as well as the LSB framing signal. If the two wires have M and N levels, the number of states is clearly $M \times N$. The efficiency of such a scheme is determined by the number of redundant states among the $M \times N$ states, the radix of the represented word (data transition rate), and the hardware overhead in encoding and decoding. In the last section, it was stated that using a 4-level signal to represent 2 bits of data for each state is rather inefficient because a separate, full bandwidth wire is then required for the LSB signal, offsetting the advantage of transmitting at half the normal bit-rate. By permutating with other signals, this inefficiency can be removed. The most useful combinations are briefly described below:

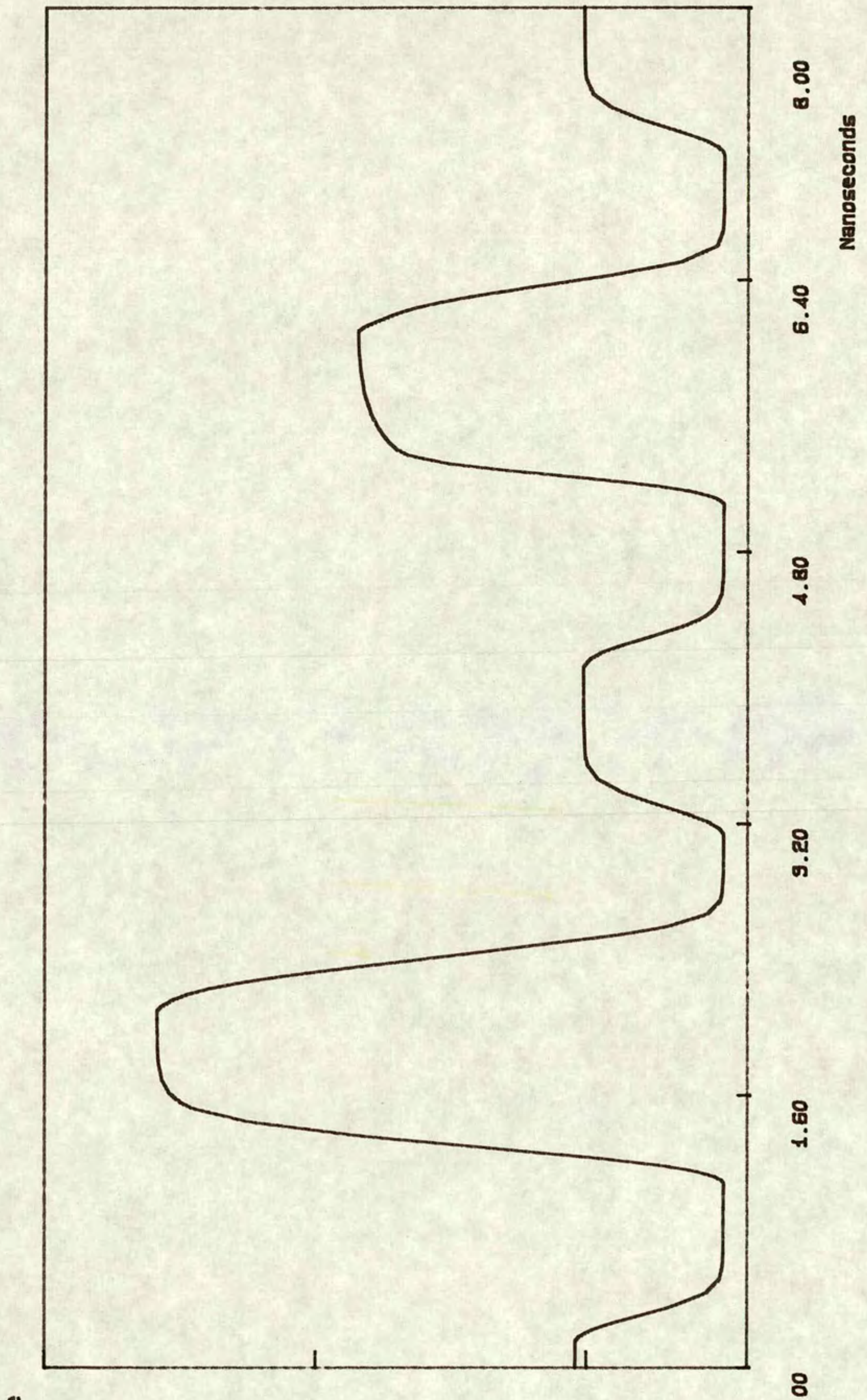


Fig. 6.22 Simulation of 4-level generation circuit.

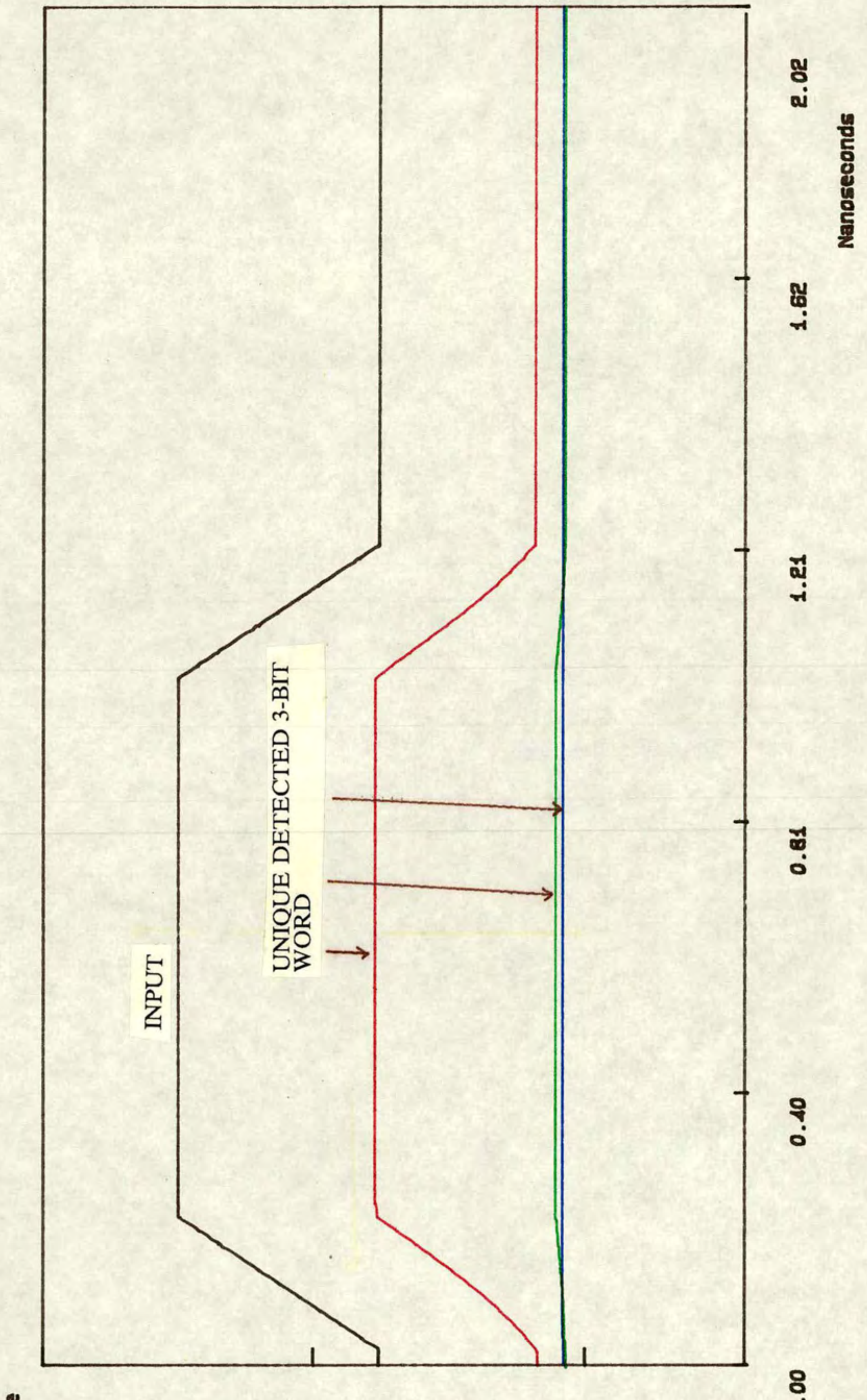


Fig. 6.23 Simulation of 4-level detection circuit (+1,1)

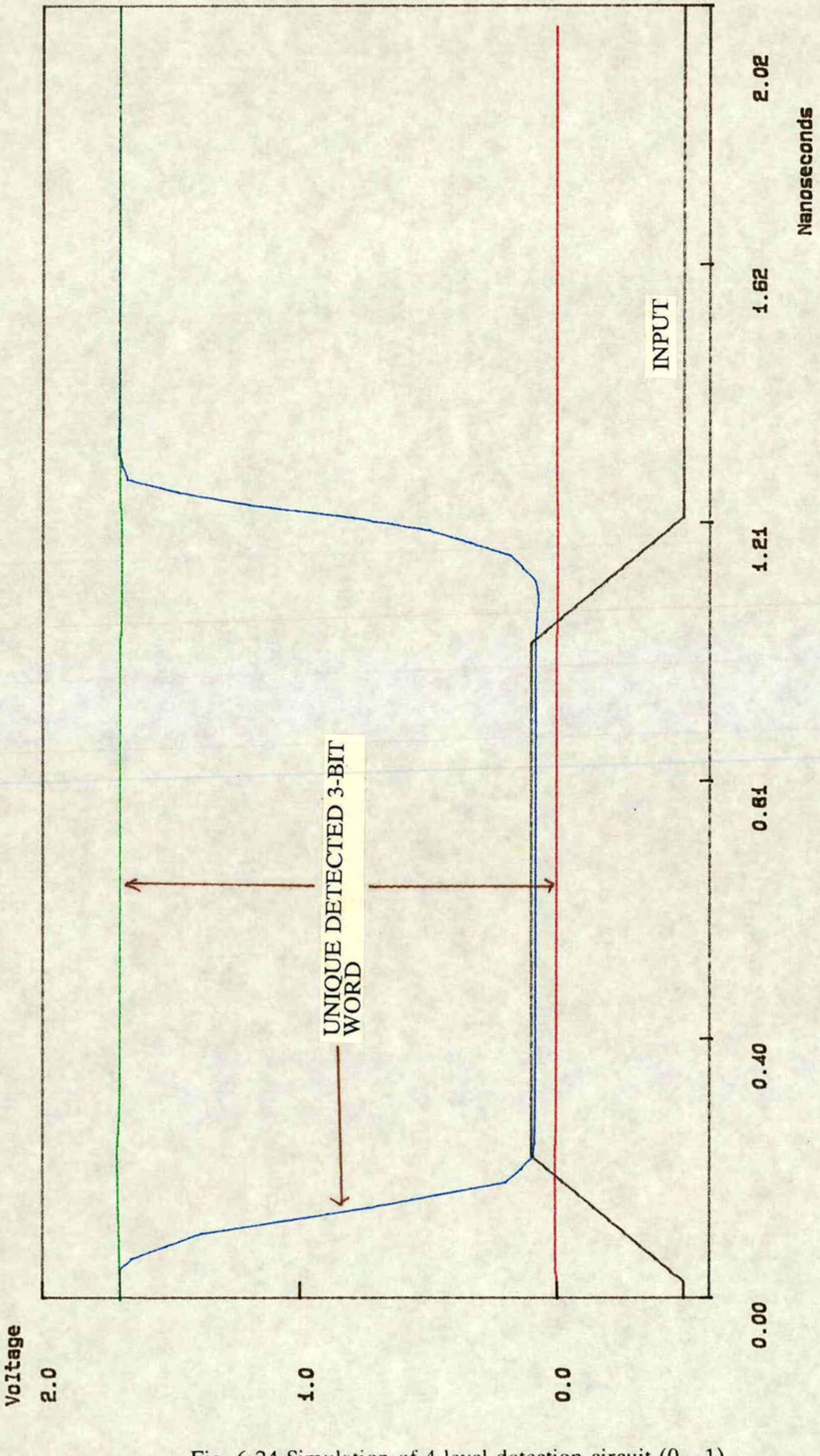


Fig. 6.24 Simulation of 4-level detection circuit $(0, -1)$.

(a) One 3-level signal and one 2-level signal

Four of the six states can be used for radix-4 data, and one state for LSB. There is one redundant state. Data rate is half of 2-level system.

(b) Two 3-level signals

The 9 states can be used to represent 3 bits of data per transition (8 states), with one state for the LSB signal. There is no redundant state. Data rate is $\frac{1}{3}$ of 2-level system.

(c) Two 4-level signals.

This can be used to represent radix-16 format, but requiring a separate LSB wire. This has the most redundant states (15) due to the extra 2-level wire. The only advantage is that the data transition rate is $\frac{1}{4}$ of the 2-level system.

Irrespective of their relative efficiency, the most important advantage of signaling at a reduced rate in all of these combinations is the reduced possibility of data error due to clock jitter and mismatched lines. If extra latency is acceptable (while the data rate is maintained), as in non-recursive systems, then the wide pulse widths of these signals allow easier detection, and board-to-board delays can be "absorbed" into the low bit rate.

The most efficient combination is clearly the 2×3-level one. While the 2×4-level option has the most redundant states, the low data transmission rate means that the extra LSB wire, which has the same $\frac{1}{4}$ bandwidth as the two 4-level wires, is not as large a wastage in area and power as in the normal 2-level system.

As an example, the block diagram of the encoding/decoding scheme for 2×3-level is shown in Fig. 6.25. The logic circuitry is shown in Figs. 6.26 and 6.27. Note that the decoding is logically simplified by choosing a suitable encoding scheme, given by Table 6.2 below. Although the logic appears to be complicated, especially in the encoding section, it is fairly modular and repetitive, thus facilitating compaction in actual physical area. The symbols correspond to those used in the diagrams and are summarised as follows:

X = most significant 3-level signal

Y = least significant 3-level signal

XLSB = least significant bit (binary) of X decoded signal

XMSB = most significant bit of X decoded signal

YLSB = least significant bit of Y decoded signal

YMSB = most significant bit of Y decoded signal

DATA			3-level signals		decoded signals			
D3	D2	D1	X	Y	XMSB	XLSB	YMSB	YLSB
0	0	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0	1
0	1	1	1	−1	0	0	1	1
1	0	0	0	1	0	1	0	0
1	0	1	0	0	0	1	0	1
1	1	1	0	−1	0	1	1	1
0	1	0	−1	1	1	1	0	0
1	1	0	−1	0	1	1	0	1
LSB	-	-	−1	−1	1	1	1	1

Table 6.2 2×3-level encode/decoding scheme.

The logic circuitry recodes 3 bit of DATA at a time and the LSB signals into six signals to drive the two 3-level output buffers. At the receiving end, each 3-level signal is decoded into a 2-bit word, i.e. a total of 4 bits as shown in the Table. Referring to the Table, this scheme is devised such that 3 of the decoded bits (XLSB, YMSB and YLSB) correspond directly or indirectly to the encoded 3-bit data from the transmitting end, simplifying decoding substantially. Note that for a DATA of 110, the decoded XLSB-YMSB-YLSB word is 101, same as that for a DATA of 101. This is also true for DATA inputs 010 and 100. However, the decoded XMSB bit in each case allows them to be distinguished, and results in the decoding logic in Fig. 6.27.

By nature of the 1/3 transmission rate and radix-8 representation, the LSB signal has a pulse width of 3 bits in duration. As a result, the three most significant bits of the normal radix-2 bit-serial data is not represented. Therefore, in order to preserve the sign bit information, 4 sign bits are needed in the radix-2 data word, as opposed to the normal format of 2 sign bits. This penalty renders this scheme unsuitable for short system data word lengths. A further restriction is clearly the need to have data word lengths equal to a multiple of 3, compared to a multiple of 2 in the conventional system word.

The wastage of the 3 most significant bits and the additional 2 sign bits can be avoided by using one 3-level wire plus one 4-level wire (3-level+4-level). This provides a total of 12 states. 8 of these states can be used to represent the radix-8 DATA as in the 2×3-level scheme. The remaining 4 states can be used for the 4 possible most significant 3 bits of data **and** the LSB signal (noting that the LSB signal is in fact coincident with the most significant 3 bits). With the 2 sign bits format of the conventional system, these 4 possible states are:

110 111 001 000

Therefore, with this 3-level+4-level scheme, there is no redundant state, and the 4-level signal is used far more efficiently than the 2×4-level scheme suggested earlier.

Finally, the circuit overhead involved in the recoding/encoding circuitry in this example can only be justified if off-chip signals are limited, such as between very large chips. Approximately 30 gates are needed for the encoding/decoding, and have to be traded off for the 1/3 off-chip transmission rate. Twice this number of gates are needed if the 3-bit decoded data needs to be converted back to the conventional bit-serial format. The latency induced by the conversion is unimportant in non-recursive systems, or in systems with a long word length. If the latency and the overhead are acceptable, the low data rate virtually guarantees error-free communication over circuit boards. With the off-chip signals lasting 3 clock cycles, board to board delays less than 3 clock cycles can be easily accommodated by introducing more delay stages at the receiving end, without the need to match lines accurately.

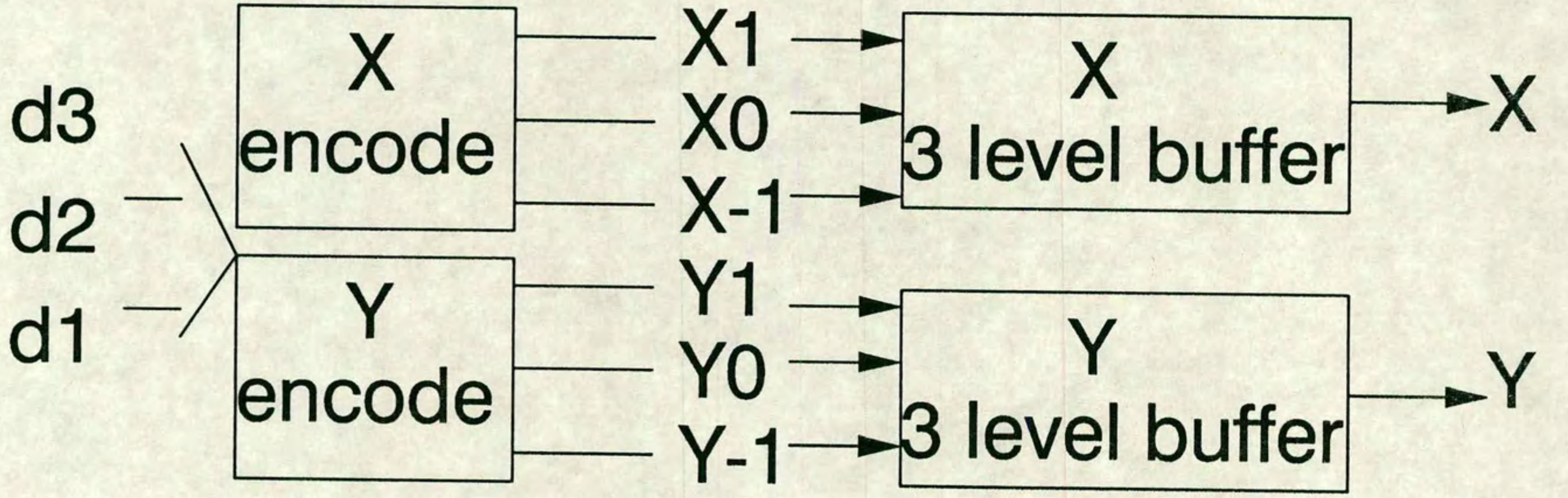


Fig. 6.25 Block diagram of 2×3 level circuit

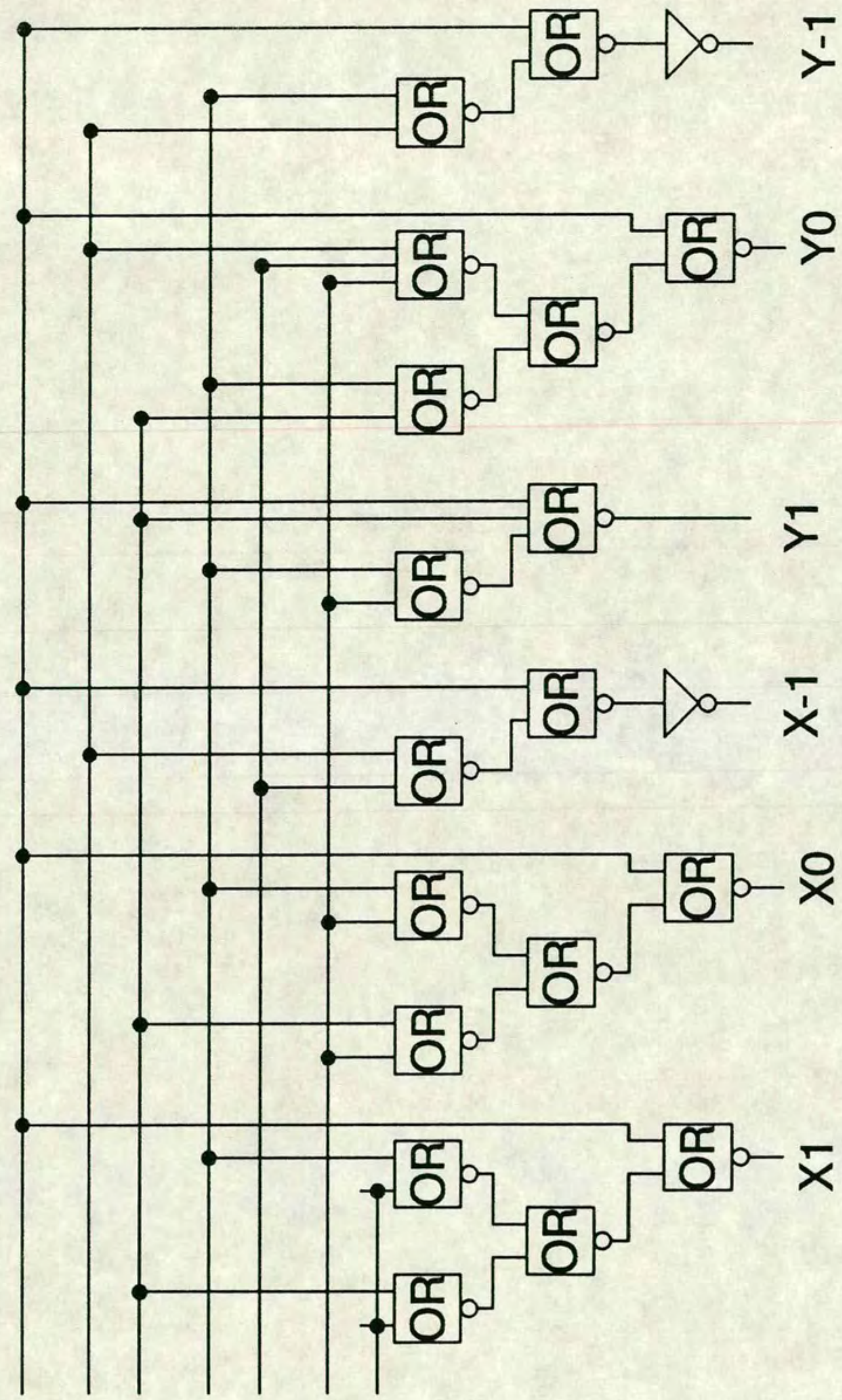


Fig. 6.26 2×3 level encoding logic

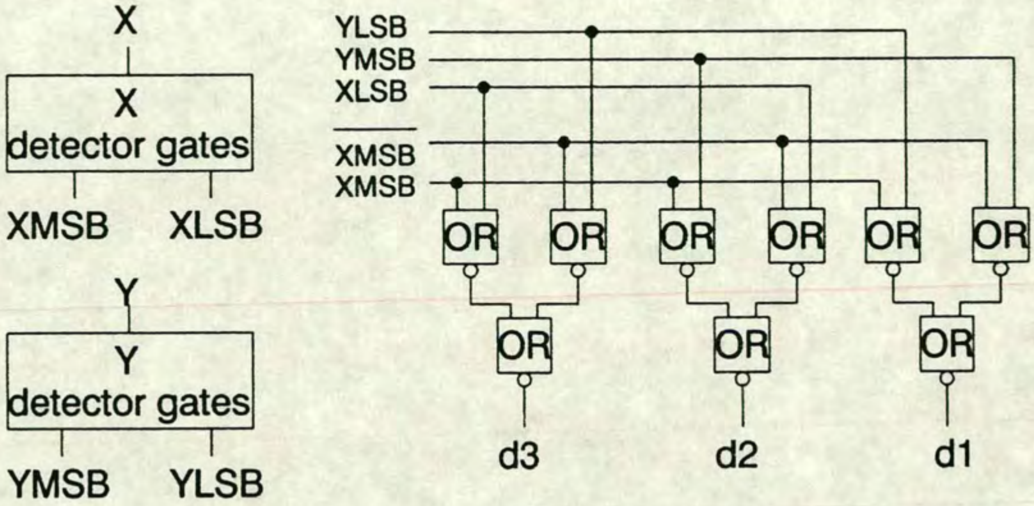


Fig. 6.27 2×3 level decoding logic

6.4. GaAs single phase clocking

One of the limitations of the GaAs cells as they stand is the two-phase clocking scheme. With two clock phases, three possible undesirable situations exist:

(i) $\phi 1$ skewed relative to $\phi 2$ locally.

(ii) $\phi 1$ or $\phi 2$ skewed relative to itself globally.

(iii) $\phi 1$ and $\phi 2$ skewed by different extents globally.

The conditions for correct operation of the type of latches used have already been set out earlier. As technology improves or when more pipelining stages are used, these conditions become more and more stringent. The most critical requirement of these is that the maximum overlap time between the two clock phases must be less than the minimum propagation delay between two pipeline stages. In the worst case, this amounts to one DCFL gate delay plus one pass transistor delay. With the

Honeywell process, this is approximately 300ps. With a MODFET/HEMT process, this reduces to about 100ps to 150ps. Whereas local clock skew in silicon is insignificant compared to gate delays, enabling 4-phase or even 6-phase systems to be realised [123], the GaAs and MODFET/HEMT figures necessitate careful clock distribution and estimation of parasitic capacitances right down to clock lines that are in close proximity.

A single phase system has clear advantages in this respect. Here, a single phase system is defined as having one clock phase both locally and globally within a chip. Note that a system with ϕ and a locally or globally generated $\overline{\phi}$ is a two-phase system. In a single-phase system, the $\phi 1$ and $\phi 2$ latches are replaced by an active-high latch and an active-low latch. They have previously been denoted by π and μ latches respectively [123]. As their names imply, the π-latch accepts data when the single clock signal ϕ is HIGH. The μ-latch is activated when ϕ is LOW. Since both pipelining latches use exactly the same clock signal, the possibility of local clock skew is totally eliminated. The only type of skew possible is the relative skew between ϕ signals that are physically distant from each other, as opposed to the three possible skews in the two-phase system.

Other advantages of a single-phase scheme include:

(i) Only one internal clock signal needs to be generated and distributed.

Even though the loading on the single clock may be more than doubled, the simplicity of clock generation more than compensates for this.

(ii) The whole period of the clock signal is used in full.

There is no longer the need to maintain a non-overlap between clock phases by the clock generator and the associated buffering.

(ii) One external clock can be used easily, without having to regenerate the two-phases on every chip, or distribute two clock signals globally on a circuit board.

The single clock phase inevitably means that the pseudo-dynamic latch, with a feed-forward path and a refresh path, cannot be used. The next section discusses previous types of single-phase latches, and proposes two new π and μ latch designs.

The π and μ latch designs

In silicon, the π and μ latches have been based on the designs shown in Figs. 6.28 & 6.29 respectively [123]. In their fully-static forms, each latch contains two 3-input complex gates. However, complex gates are undesirable in GaAs DCFL design (Chapter 3), and the logic of these two gates must preferably be re-arranged into NOR functions. First consider the μ latch.

In the μ latch, the OR-NAND complex gates (Fig. 6.29) can be simply replaced by two 2-input NOR gates. It is desirable to maintain the principle of a low clock current loading, for reasons set out in Chapter 5 in the discussion of the pseudo-dynamic latches and the clock generator. The most important of these is that a forward biased MESFET draws a large gate current. When the signal in question is the clock, which drives hundreds of latches each having one to two clock inputs, the clock generator needs to supply a current of the order of 10mA simply to maintain forward bias under DC conditions. The situation is worse in a single-phase system, by its very nature. Therefore it is even more important here to avoid the DCFL equivalent of the μ and π latches.

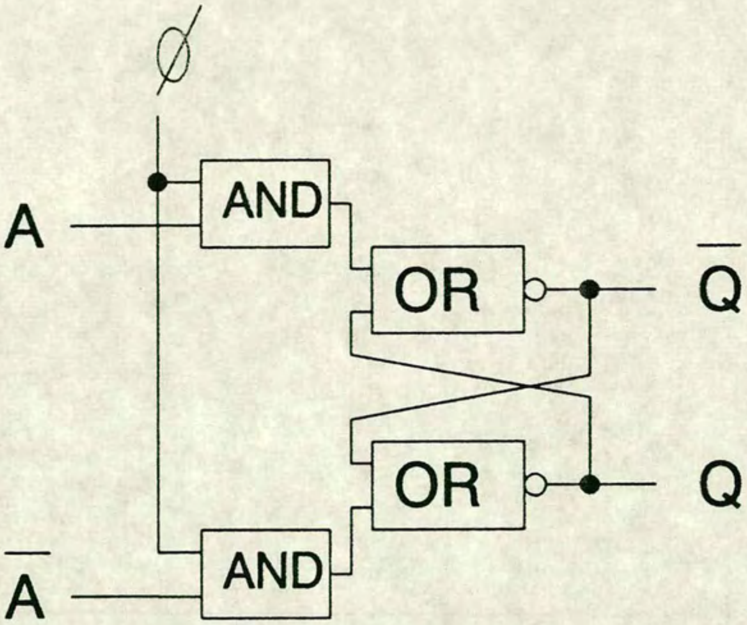


Fig. 6.28 Fully static silicon π latch.

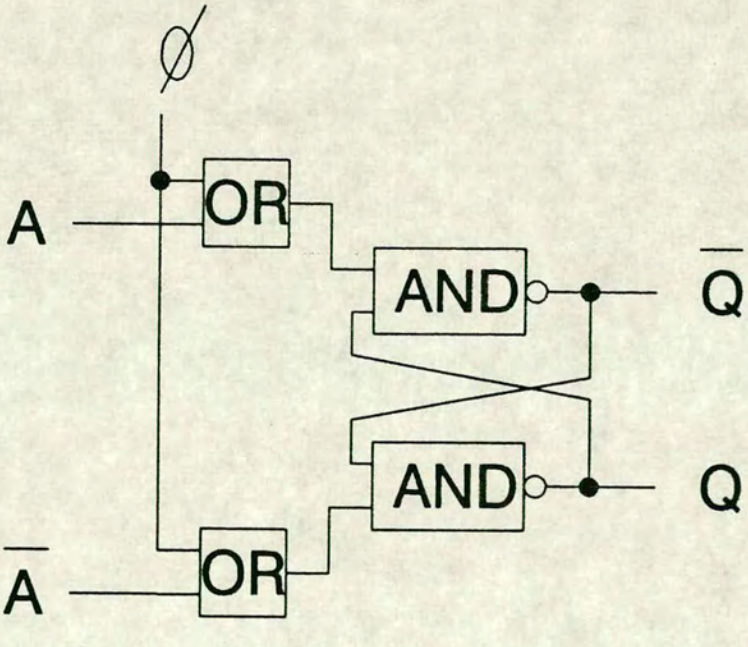


Fig. 6.29 Fully static μ latch.

In the pseudo-dynamic latches in Chapter 5, D-MESFET pass transistors and a ±0.5V clock swing were used for this purpose. In the new μ latch proposed here, the same clock swing is used, while the D-MESFET is used in a unique NOR gate configuration. This configuration and its symbol are shown in Fig. 6.30. The NOR gate uses both a D-MESFET and a E-MESFET in parallel for the pull-down logic. and will be denoted as a D-NOR gate. It is an extension of the special inverter (a D-INVERTER) described in Chapter 5, used to detect the multi-level signal. The functioning of the D-INVERTER has been proven experimentally in the 3-level circuit (Appendix 15). When used in the μ latch, the D-MESFET pull-down device is driven by the ±0.5V clock signal. The resulting μ latch is shown in Fig. 6.31.

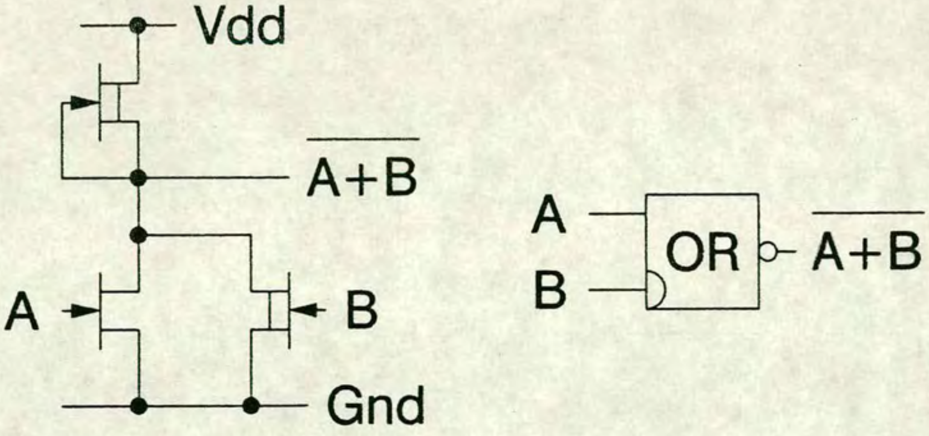


Fig. 6.30 E/D-MESFET NOR gate.

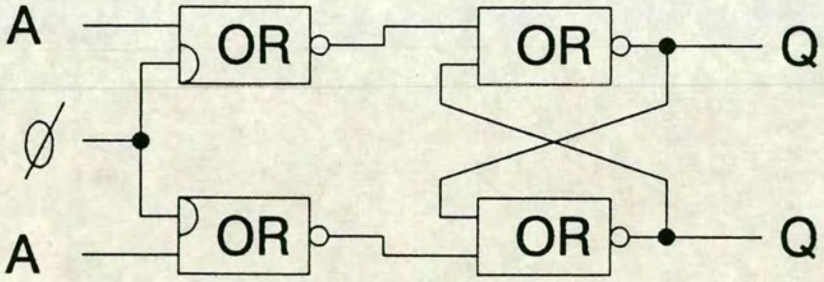


Fig. 6.31 GaAs μ latch.

In the case of the π latch, the logical equivalent of the AND-NOR complex gate inevitably requires the use of one NAND gate, which is undesirable though allowable in GaAs DCFL. If a NAND gate has to be used, then it is simpler to use a complex gate instead, because both have similar disadvantages (Chapter 5). To solve this problem, and to maintain low clock loading, the pseudo-dynamic latch can be used with a "resistive" refresh path, rather than a clocked path. The resulting π latch is shown in Fig. 6.32.

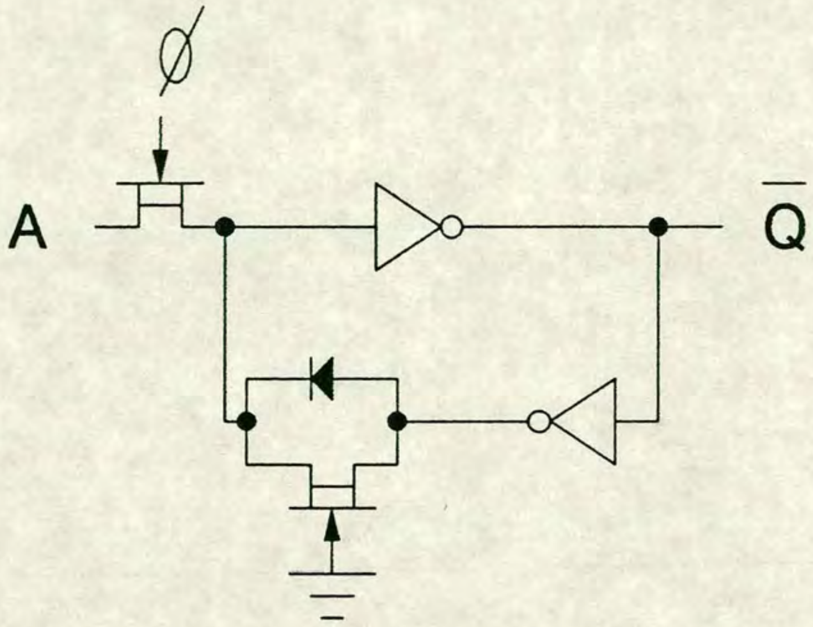


Fig. 6.32 GaAs π latch.

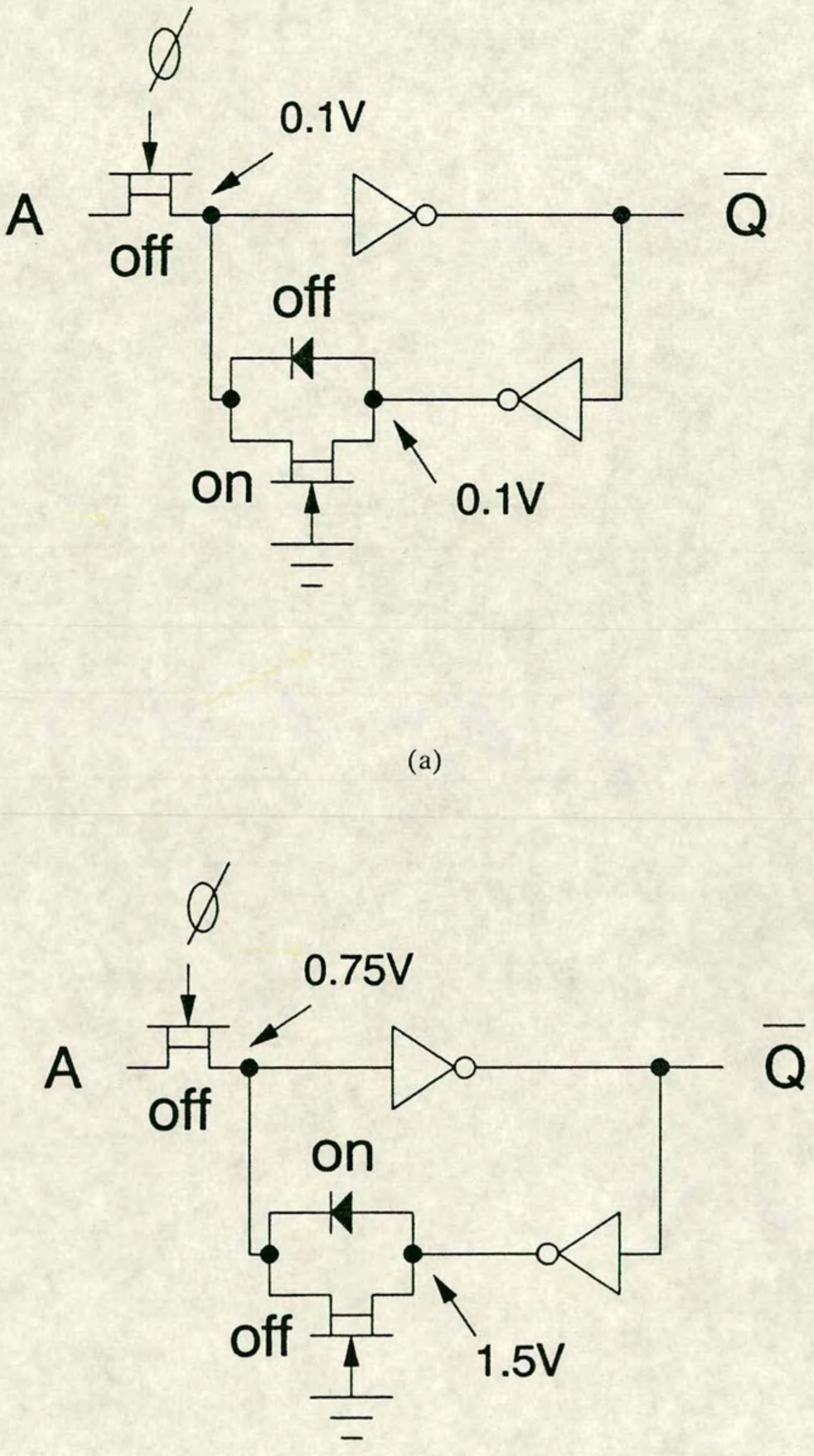
Referring to Fig. 6.32, in order to retain the signal after ϕ has changed to LOW, a unique arrangement is used in the refresh path. This consists of a D-MESFET with V_g at 0V, in parallel with a second D-MESFET connected as a diode. The first D-MESFET acts as a resistive path for a LOW input latch signal. The diode is used to retain a HIGH input signal. This diode is necessary because the V_t of the D-MESFET in the Honeywell process is -0.5V. If the input signal to the latch is logic HIGH, the feedback signal would be $+0.75$V or higher and the grounded-gate D-MESFET would be OFF, thus failing to refresh the signal. Both of the D-MESFETs in the refresh path can be small devices and do not need to be rigidly ratioed to the forward inverter, unlike the refresh path of the pseudo-dynamic latch. This is best explained by Fig. 6.33(a) & 6.33(b), showing the two refresh states. Only one D-MESFET is ON or forward biased in each state, and no voltage degradation results from the feedback path.

The drawback of this refresh technique is that when ϕ is ON, the input signal may conflict with the old data in the latch. Since the refresh D-MESFETs do not need to be ratioed, the hazard condition can minimised by ensuring that they are minimum dimension devices, and that the DCFL gate driving the μ-latch is at least twice the size of the refresh inverter. Fig. 6.34 shows a simulation of a π—μ—π latch arrangement, clocked at 1 GHz. The input file is in Appendix 14.

One clear problem of using these two latches is their asymmetry, both in size and speed. First consider their difference in size.

Although the μ latch is larger in size and has two clock inputs, the pipelining strategy essentially determines the number of π and μ latches needed for a certain function. The situation is not different from a two-phase system, where there can be more $\phi1$ latches than $\phi2$ latches or vice versa. Therefore this should not pose a major layout problem compared to the two-phase system.

In terms of speed, the slower μ latch, owing to the two full gate delays, ultimately determines maximum speed if two latches are directly cascaded. In the pipelined logic circuits in this work, the longest logic path between stages determines maximum speed, and the partitioning strategy can be chosen to take the different latch delays into account.



(a)

(b)

Fig. 6.33 Two conditions of GaAs π latch.

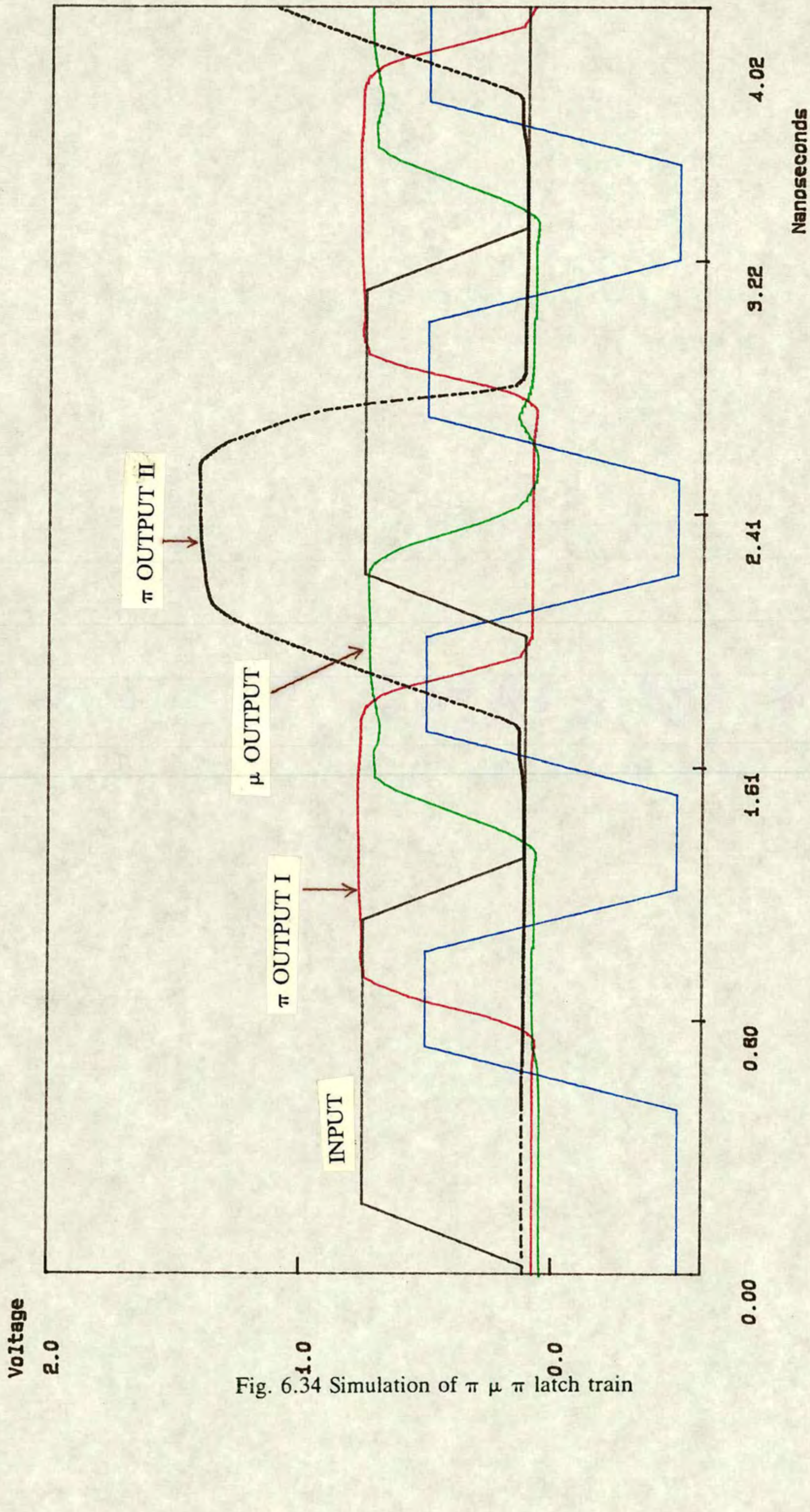


Fig. 6.34 Simulation of π μ π latch train

Finally, with the properties of these two latches known, it is important to establish the condition for their correct operation. In the worst case situation, where the faster π latch is directly driving the μ latch, the limit of correct data transfer is when ϕ goes from LOW to HIGH, turning the π latch on. The μ latch must turn off before the output of the π latch changes, to prevent the data in the μ latch from being corrupted by the new data entering π. This depends on the gate delay through the D-NOR gate in the μ latch, and the total delay through the π latch, and must satisfy the following:

$$t_{pd(D-NOR)} < t_{pd(pass\ transistor)} + t_{pd(inverter)}$$

The exact values depend on their relative sizes, and need to be designed such that this is satisfied. Noting that right hand side has two delay terms, this is easily achieved in practice. The simulation results shown in Fig. 6.34 used inverter sizing identical to the $\phi 1/\phi 2$ latches, and the D-NOR gate has a fanout of 2.

6.5. Conclusion

In this Chapter, two novel schemes to further enhance the advantages of the GaAs bit-serial cells have been discussed. The multi-level signaling scheme had been demonstrated to be feasible by the test results of a 3-level encode/decoding circuit. The single-phase latches are also unique in their circuit design, and significantly simplifies clock timing, loading, and distribution on and off chip. Total compatibility with the standard GaAs cells in Chapter 5 are also assured.

Chapter 7

Discussion and conclusion

Further work should include the design of bit-serial digital-to-analogue (D/A) and analogue-to-digital (A/D) converters. Bit-parallel converters, as used in most time domain instrumentation systems, have proved to be technically difficult in the laboratory, partly due to process variations. More severe are the GaAs MESFET properties such as channel-widening effect, hysteresis and sidegating, which affect non-linear circuits more than digital circuits [108].

An interesting project would be to integrate digital bit-serial filters with switched-capacitor-based converters which adopt the the pulse-density modulation (PDM) principle. These types of converters, as used in silicon, have been found to place less stress on the DC behaviour of a technology, and provide superior linearity. However, a high speed clock was necessary for bit-stream processing. Recently published GaAs switched-capacitor filters have shown promising results [141]. The simple FIR example using the GaAs cells has shown the space efficiency of the bit-serial cells. Combining them to form mixed-mode ICs should be a distinct possibility and a challenging exercise.

Another very useful cell for the bit-serial library would be on-chip RAM. This could be used to store filter coefficients in an adaptive system or intermediate states in a multiplexed system.

With a complete set of cells, clocked at the same nominal rate as the existing cells (500MHz), the circuit principles proposed in this work should allow the design and fabrication of either a single chip (if chip yield improves) or a small chip set which would implement digital filters capable of processing video frequency signals.

There is some potential for GaAs bit-serial circuits to be designed using a "compilation" approach, as bit-serial systems are now. However, the different fanin and fanout properties of GaAs logic circuits, together with the more stringent timing requirements due to the higher clock rates, will require extensive modifications to the embedded simulator and layout synthesis software for this to be effective. It is difficult to predict whether the degradation in speed compared to full-custom designs would be acceptable. It may be found that GaAs gate arrays will provide a more acceptable compromise between design time and performance.

Finally, the higher radix approach introduced in Chapter 6 should ideally be developed fully into high-radix computation. The example used in the earlier discussion encodes and decodes 3 bits of data at a time, which parallels with the modified-Booth's algorithm. This may be used to advantage to design a more efficient multiplier, without the need to convert the radix-8 format back to radix-2. Since the multiplier normally forms the core of filters, other arithmetic elements should also follow easily if this can be achieved. Their area-speed compromises can then be weighed against the one-bit approach adopted in the cells described in Chapter 5.

Conclusion

This thesis has described the circuit engineering aspects of a number of GaAs bit-serial cells, a multi-level off-chip signaling scheme, and single-phase clocked latches. The modification of SPICE2G.6 to incorporate a suitable MESFET model has also been discussed in some detail. The test results now available have demonstrated the correctness of the novel principles and circuit techniques described.

Because GaAs technology is based on the MESFET transistor, a device which has a Schottky diode gate and hence a low input impedance and transconductance, it is sufficiently different from any silicon technology, even NMOS, to necessitate an independent design approach. Thus a simple transfer of NMOS designs onto GaAs by replacing the MOSFETs with MESFETs is impossible, despite the apparent similarities between static NMOS circuits and GaAs direct-coupled FET logic.

Clearly the bit-serial techniques employed in this work cannot solve all the problems presented by the use of MESFETs. No single architecture can possibly achieve that ideal, and compromises between the speed, power, and level of functionality of a chip must inevitably be made. However, the nature of GaAs technology implies that locally intensive computational blocks, avoiding excessive chip-to-chip connections, are favourable. This is also recognised by other general research into architectures for MESFET-related high speed technologies. In this respect, the main properties of bit-serial circuits, namely the efficiency of operators and utilisation of chip area together with the small number of off-chip interconnects, are particularly attractive.

In the circuits presented in this work the locality, modularity, and the functionality of a chip have been emphasised at the expense of the overall system throughput rate. Whereas these same tradeoffs have been accepted and used to advantage in silicon, they have been largely ignored in GaAs, and work using this technology has concentrated more on the traditional approach of bit-parallel circuits with longer and longer word lengths, so attempting to compete directly with silicon ECL. The resulting system insertion rate of these bit-parallel chips has been very slow to date, and consistently below market forecasts and expectations. This work offers an alternative circuit technique which may be particularly attractive for signal processing applications.

As part of this work a multi-level signaling technique was developed which significantly improved the efficiency of chip-to-chip communications and also reduced synchronisation problems. This was achieved through a combination of composite signals and reduced data transmission rates.

Lastly, single-phase latches were proposed which would effectively minimise problems arising from clock skew and simplify clock distribution and generation. These were inherently compatible with the existing GaAs cells, but would require more modifications to the GaAs cells than would the multi-level circuits.

Test results are presented which demonstrate the low speed functionality of the 3-level encode/decode circuits and the clock generator. Other circuits such as the multiplier and adder have been fabricated but, due to difficulties in exporting GaAs ICs from the United States, are yet to be tested. Simulations have demonstrated their correct functionality and have proved (as far as possible using simulation) that the circuit principles behind the psuedo-dynamic latch, the multi-level signaling technique and the unconventional ±0.5V clocking scheme are correct. As the simulation technique employed has itself been proven to give a fair indication of actual functionality (through the chips that have been measured) it is expected that these other circuits will operate as expected. Further testing, which will be carried out in the near future, will include high speed tests on bonded chips.

Appendix 1

Modified SPICE JFET subroutine

This Appendix contains the modified JFET subroutine in full. Other modifications are less extensive and are described in the main text in Chapter 4. Note that some of the codes have been split into two or more lines for formatting, and will not compile if used without change.

```
      subroutine jfet
      implicit double precision (a-h,o-z)
c
c     this routine processes jfets for dc and
c     transient analyses.
c
c
c     lower case comments mark modifications. the
c     main ones are
c     (i)  jfet current equation in cutoff, ohmic,
c     and sat. regions;
c     (ii) diode equation.
c
c
c spice version 2g.6  sccsid=tabinf 3/15/83
      common /tabinf/ ielmnt,isbckt,nsbckt,iunsat,
         nunsat,itemps,numtem,
     1   isens,nsens,ifour,nfour,ifield,icode,
         idelim,icolum,insize,
     2   junode,lsbkpt,numbkp,iorder,jmnode,iur,iuc,
         ilc,ilr,numoff,isr,
     3   nmoffc,iseq,iseq1,neqn,nodevs,ndiag,iswap,
         iequa,macins,lvnim1,
     4   lx0,lvn,lynl,lyu,lyl,lx1,lx2,lx3,lx4,lx5,lx6,
         lx7,ld0,ld1,ltd,
     5   imynl,imvn,lcvn,nsnod,nsmat,nsval,icnod,
         icmat,icval,
     6   loutpt,lpol,lzer,irswpf,irswpr,icswpf,icswpr,
         irpt,jcpt,
     7   irowno,jcolno,nttbr,nttar,lvntmp
c spice version 2g.6  sccsid=cirdat 3/15/83
      common /cirdat/ locate(50),jelcnt(50),nunods,
      ncnods,numnod,nstop,
     1   nut,nlt,nxtrm,ndist,ntlin,ibr,numvs,numalt,numcyc
c spice version 2g.6  sccsid=status 3/15/83
      common /status/ omega,time,delta,delold(7),
      ag(7),vt,xni,egfet,
     1   xmu,sfactr,mode,modedc,icalc,initf,
         method,iord,maxord,noncon,
     2   iterno,itemno,nosolv,modac,ipiv,
         ivmflg,ipostp,iscrch,iofile
c spice version 2g.6  sccsid=knstnt 3/15/83
      common /knstnt/ twopi,xlog2,xlog10,root2,
      rad,boltz,charge,ctok,
     1   gmin,reltol,abstol,vntol,trtol,
         chgtol,eps0,epssil,epsox,
     2   pivtol,pivrel
c spice version 2g.6  sccsid=blank 3/15/83
      common /blank/ value(200000)
      integer nodplc(64)
      complex cvalue(32)
      equivalence (value(1),nodplc(1),cvalue(1))
c
c
      dimension vgso(1),vgdo(1),cgo(1),cdo(1),cgdo(1),
      gmo(1),gdso(1),
     1   ggso(1),ggdo(1),qgs(1),cqgs(1),qgd(1),cqgd(1)
      equivalence (vgso(1),value( 1)),(vgdo(1),value( 2)),
     1            (cgo (1),value( 3)),(cdo (1),value( 4)),
     2            (cgdo(1),value( 5)),(gmo (1),value( 6)),
     3            (gdso(1),value( 7)),(ggso(1),value( 8)),
     4            (ggdo(1),value( 9)),(qgs (1),value(10)),
     5            (cqgs(1),value(11)),(qgd (1),value(12)),
     6            (cqgd(1),value(13))
c
c
      loc=locate(13)
   10 if ((loc.eq.0).or.(nodplc(loc+25).ne.0)) return
      locv=nodplc(loc+1)
      node1=nodplc(loc+2)
      node2=nodplc(loc+3)
      node3=nodplc(loc+4)
      node4=nodplc(loc+5)
      node5=nodplc(loc+6)
      locm=nodplc(loc+7)
      ioff=nodplc(loc+8)
      type=nodplc(locm+2)
      locm=nodplc(locm+1)
      loct=nodplc(loc+19)
c
c  dc model parameters
c
c  the allocation of parameters within value
c  are as follows:
c
c     index       name        comment
c
c     ---------------------------------------
c     locm+1      vto         threshold voltage
c         +2      beta        beta
c         +3      xlamb       lambda
c         +4      gdpr        drain admittance
c         +5      gspr        source admittance
c         +6      czgs        cgs
c         +7      czgd        cgd
c         +8      phib        built-in gate potential
c         +9      csat        is
c         +10     kf          flicker noise coeff.
c         +11     af          .... exponent
c         +12     fc          cap. ceff.
c         +13     alpha       alpha
c         +14     ni          diode ideality factor
c         +15     m           ids eqn. first exponent
c         +16     n           ids eqn. sec. exp.
c         +17     co          cap. coeff. for vds dep.
c         +18     f1          first cap. eqn. coeff.
c         +19     f2          sec. ...
c         +20     f3          3rd ....
c         +21     vcrit       ref. volt. for limiting
c
      area=value(locv+1)
      vto=value(locm+1)
      beta=value(locm+2)*area
      xlamb=value(locm+3)
      gdpr=value(locm+4)*area
      gspr=value(locm+5)*area
      csat=value(locm+9)*area
      vcrit=value(locm+21)
c
c  the following 5 are additional parameters
```

```
c  for an accurate jfet
c  model based on curtice.
c
      alpha=value(locm+13)
      xdni=value(locm+14)
      xm=value(locm+15)
      xn=value(locm+16)
      co=value(locm+17)
      m=xm
      n=xn
      vt1=vt*xdni
c
c  initialization
c
      icheck=1
      go to (100,20,30,50,60,70), initf
   20 if(mode.ne.1.or.modedc.ne.2.or.nosolv.eq.0)
      go to 25
      vds=type*value(locv+2)
      vgs=type*value(locv+3)
      vgd=vgs-vds
      go to 300
   25 if(ioff.ne.0) go to 40
      vgs=-1.0d0
      vgd=-1.0d0
      go to 300
   30 if (ioff.eq.0) go to 100
   40 vgs=0.0d0
      vgd=0.0d0
      go to 300
   50 vgs=vgso(lx0+loct)
      vgd=vgdo(lx0+loct)
      go to 300
   60 vgs=vgso(lx1+loct)
      vgd=vgdo(lx1+loct)
      go to 300
   70 xfact=delta/delold(2)
      vgso(lx0+loct)=vgso(lx1+loct)
      vgs=(1.0d0+xfact)*vgso(lx1+loct)-xfact*
         vgso(lx2+loct)
      vgdo(lx0+loct)=vgdo(lx1+loct)
      vgd=(1.0d0+xfact)*vgdo(lx1+loct)-
          xfact*vgdo(lx2+loct)
      cgo(lx0+loct)=cgo(lx1+loct)
      cdo(lx0+loct)=cdo(lx1+loct)
      cgdo(lx0+loct)=cgdo(lx1+loct)
      gmo(lx0+loct)=gmo(lx1+loct)
      gdso(lx0+loct)=gdso(lx1+loct)
      ggso(lx0+loct)=ggso(lx1+loct)
      ggdo(lx0+loct)=ggdo(lx1+loct)
      go to 110
c
c  compute new nonlinear branch voltages
c
  100 vgs=type*(value(lvnim1+node2)-value(lvnim1+node5))
      vgd=type*(value(lvnim1+node2)-value(lvnim1+node4))
  110 delvgs=vgs-vgso(lx0+loct)
      delvgd=vgd-vgdo(lx0+loct)
      delvds=delvgs-delvgd
      cghat=cgo(lx0+loct)+ggdo(lx0+loct)*delvgd+
          ggso(lx0+loct)*delvgs
      cdhat=cdo(lx0+loct)+gmo(lx0+loct)*delvgs+
          gdso(lx0+loct)*delvds
     1   -ggdo(lx0+loct)*delvgd
c
c  bypass if solution has not changed
c
      if (initf.eq.6) go to 200
      tol=reltol*dmax1(dabs(vgs),dabs(
            vgso(lx0+loct)))+vntol
      if (dabs(delvgs).ge.tol) go to 200
      tol=reltol*dmax1(dabs(vgd),dabs(
            vgdo(lx0+loct)))+vntol
      if (dabs(delvgd).ge.tol) go to 200
      tol=reltol*dmax1(dabs(cghat),dabs(
          cgo(lx0+loct)))+abstol
      if (dabs(cghat-cgo(lx0+loct)).ge.tol) go to 200
      tol=reltol*dmax1(dabs(cdhat),dabs(
          cdo(lx0+loct)))+abstol
      if (dabs(cdhat-cdo(lx0+loct)).ge.tol) go to 200
      vgs=vgso(lx0+loct)
      vgd=vgdo(lx0+loct)
      vds=vgs-vgd
      cg=cgo(lx0+loct)
      cd=cdo(lx0+loct)
      cgd=cgdo(lx0+loct)
      gm=gmo(lx0+loct)
      gds=gdso(lx0+loct)
      ggs=ggso(lx0+loct)
      ggd=ggdo(lx0+loct)
      go to 900
c
c  limit nonlinear branch voltages
c
  200 ichk1=1
      call pnjlim(vgs,vgso(lx0+loct),vt1,vcrit,icheck)
      call pnjlim(vgd,vgdo(lx0+loct),vt1,vcrit,ichk1)
      if (ichk1.eq.1) icheck=1
      call fetlim(vgs,vgso(lx0+loct),vto)
      call fetlim(vgd,vgdo(lx0+loct),vto)
c
c  determine dc current and derivatives
c
c  note introduction of diode ideality
c  factor xdni and modifications
c  to diode and related equations
c
  300 vds=vgs-vgd
      if (vgs.gt.-5.0d0*vt1) go to 310
c     ggs=-csat/vgs
      ggs=-csat/vgs+gmin
      cg=ggs*vgs
      go to 320
  310 evgs=dexp(vgs/vt1)
c     ggs=csat*evgs/vt1
      ggs=csat*evgs/vt1+gmin
c     cg=csat*(evgs-1.0)
      cg=csat*(evgs-1.0)+gmin*vgs
  320 if (vgd.gt.-5.0d0*vt1) go to 330
c     ggd=-csat/vgd
      ggd=-csat/vgd+gmin
      cgd=ggd*vgd
      go to 340
  330 evgd=dexp(vgd/vt1)
c     ggd=csat*evgd/vt1
      ggd=csat*evgd/vt1+gmin
c     cgd=csat*(evgd-1.0)
      cgd=csat*(evgd-1.0)+gmin*vgd
  340 cg=cg+cgd
c
c  compute drain current and derivitives
c  for normal mode
c
c  set up common variables for forward and
c  inverse drain, gm, and
c  gds equations.
c  note that:    gm = dids/dvgs or dids/dvgd
c  and           gds = (dids/dvds)
c                      for normal mode
c                    = (dids/dvds) - gm
```

```
c                      for inverse mode
c
  400 alphv=alpha*vds
      epos=dexp(alphv)
      eneg=dexp(-alphv)
      tanhp=(epos-eneg)/(epos+eneg)
      epos2=dexp(alphv+alphv)
      eneg2=dexp(-alphv-alphv)
      sech2=4.0d0/(epos2+eneg2+2.0d0)
      xlamvds=xlamb*vds
c
c  check if normal or inverse mode
c
      if (vds.lt.0.0d0) go to 450
      vgst=vgs-vto
      vgstm=vgst**m
      vgstn=vgst**n
      vgstm1=vgst**(m-1)
      vgstn1=vgst**(n-1)
c
c  normal mode, cutoff region
c
c
      if (vgst.gt.0.0d0) go to 410
      cdrain=0.0
c     cdrain=vds/50.0d6
      gm=0.0d0
c     gds=2.0d-8
      gds=0.0
      go to 490
c
c  the original code for the cutoff region
c
c     cdrain=area*1.214d-7*vds
c     gm=0.0d0
c     gds=area*1.214d-7
c     go to 490
c
c  normal mode, saturation region and linear
c  region as well!
c
c  the main eqns. are cdrain, gm, and gds. they
c  are based on the
c  curtice model.
c
c  cdrain = drain current eqn.
c  gm = transconductance
c  gds = gain
c
  410 cdrain=beta*tanhp*(vgstm*xlamvds+vgstn)
      gm=beta*tanhp*(xm*xlamvds*vgstm1+xn*vgstn1)
      gds=beta*(xlamb*tanhp*vgstm
     1   +alpha*sech2*(vgstn+xlamvds*vgstm))
      go to 490
c
c
c  compute drain current and derivitives for inverse mode
c
  450 vgdt=vgd-vto
      vgdtm=vgdt**m
      vgdtn=vgdt**n
      vgdtm1=vgdt**(m-1)
      vgdtn1=vgdt**(n-1)
c
c  inverse mode, cutoff region
c
      if (vgdt.gt.0.0d0) go to 460
      cdrain=0.0
c     cdrain=vds/50.0d6
      gm=0.0d0
      gds=0.0
c     gds=2.0d-8
      go to 490
c
c  the original code for the cutoff region
c
c     cdrain=area*1.214d-7*vds
c     gm=0.0d0
c     gds=area*1.214d-7
c     go to 490
c
c  inverse mode, saturation region
c
c  and linear region
c
  460 cdrain=beta*tanhp*(vgdtn-xlamvds*vgdtm)
      gm=beta*tanhp*(xn*vgdtn1-xlamvds*xm*vgdtm1)
      gds=beta*(alpha*sech2*(vgdtn
     1    -xlamvds*vgdtm)-xlamb*tanhp*vgdtm)-gm
c
c  compute equivalent drain current source
c
  490 cd=cdrain-cgd
      if (mode.ne.1) go to 500
      if ((modedc.eq.2).and.(nosolv.ne.0)) go to 500
      if (initf.eq.4) go to 500
      go to 700
c
c  charge storage elements
c
  500 czgs=value(locm+6)*area
      czgd=value(locm+7)*area
      phib=value(locm+8)
      twop=phib+phib
      fcpb=value(locm+12)
      fcpb2=fcpb*fcpb
      f1=value(locm+18)
      f2=value(locm+19)
      f3=value(locm+20)
      czgsf2=czgs/f2
      czgdf2=czgd/f2
      if (vgs.ge.fcpb) go to 510
      sarg=dsqrt(1.0d0-vgs/phib)
      qgs(lx0+loct)=twop*czgs*(1.0d0-sarg)
      capgs=czgs/sarg
      go to 520
  510 qgs(lx0+loct)=czgs*f1+czgsf2*(f3*(vgs-fcpb)
     1   +(vgs*vgs-fcpb2)/(twop+twop))
      capgs=czgsf2*(f3+vgs/twop)
  520 if (vgd.ge.fcpb) go to 530
      sarg=dsqrt(1.0d0-vgd/phib)
      qgd(lx0+loct)=twop*czgd*(1.0d0-sarg)
      capgd=czgd/sarg
      go to 560
  530 qgd(lx0+loct)=czgd*f1+czgdf2*(f3*(vgd-fcpb)
     1   +(vgd*vgd-fcpb2)/(twop+twop))
      capgd=czgdf2*(f3+vgd/twop)
c
c  store small-signal parameters
c
  560 if ((mode.eq.1).and.(modedc.eq.2).
      and.(nosolv.ne.0)) go to 700
      if (initf.ne.4) go to 600
      value(lx0+loct+9)=capgs
      value(lx0+loct+11)=capgd
      go to 1000
c
c  transient analysis
c
  600 if (initf.ne.5) go to 610
```

```
      qgs(lx1+loct)=qgs(lx0+loct)
      qgd(lx1+loct)=qgd(lx0+loct)
  610 call intgr8(geq,ceq,capgs,loct+9)
      ggs=ggs+geq
      cg=cg+cqgs(lx0+loct)
      call intgr8(geq,ceq,capgd,loct+11)
      ggd=ggd+geq
      cg=cg+cqgd(lx0+loct)
      cd=cd-cqgd(lx0+loct)
      cgd=cgd+cqgd(lx0+loct)
      if (initf.ne.5) go to 700
      cqgs(lx1+loct)=cqgs(lx0+loct)
      cqgd(lx1+loct)=cqgd(lx0+loct)
c
c  check convergence
c
  700 if (initf.ne.3) go to 710
      if (ioff.eq.0) go to 710
      go to 750
  710 if (icheck.eq.1) go to 720
      tol=reltol*dmax1(dabs(cghat),dabs(cg))+abstol
      if (dabs(cghat-cg).ge.tol) go to 720
      tol=reltol*dmax1(dabs(cdhat),dabs(cd))+abstol
      if (dabs(cdhat-cd).le.tol) go to 750
  720 noncon=noncon+1
  750 vgso(lx0+loct)=vgs
      vgdo(lx0+loct)=vgd
      cgo(lx0+loct)=cg
      cdo(lx0+loct)=cd
      cgdo(lx0+loct)=cgd
      gmo(lx0+loct)=gm
      gdso(lx0+loct)=gds
      ggso(lx0+loct)=ggs
      ggdo(lx0+loct)=ggd
c
c  load current vector
c
  900 ceqgd=type*(cgd-ggd*vgd)
      ceqgs=type*((cg-cgd)-ggs*vgs)
      cdreq=type*((cd+cgd)-gds*vds-gm*vgs)
      value(lvn+node2)=value(lvn+node2)-ceqgs-ceqgd
      value(lvn+node4)=value(lvn+node4)-cdreq+ceqgd
      value(lvn+node5)=value(lvn+node5)+cdreq+ceqgs
c
c  load y matrix
c
      locy=lvn+nodplc(loc+20)
      value(locy)=value(locy)+gdpr
      locy=lvn+nodplc(loc+21)
      value(locy)=value(locy)+ggd+ggs
      locy=lvn+nodplc(loc+22)
      value(locy)=value(locy)+gspr
      locy=lvn+nodplc(loc+23)
      value(locy)=value(locy)+gdpr+gds+ggd
      locy=lvn+nodplc(loc+24)
      value(locy)=value(locy)+gspr+gds+gm+ggs
      locy=lvn+nodplc(loc+9)
      value(locy)=value(locy)-gdpr
      locy=lvn+nodplc(loc+10)
      value(locy)=value(locy)-ggd
      locy=lvn+nodplc(loc+11)
      value(locy)=value(locy)-ggs
      locy=lvn+nodplc(loc+12)
      value(locy)=value(locy)-gspr
      locy=lvn+nodplc(loc+13)
      value(locy)=value(locy)-gdpr
      locy=lvn+nodplc(loc+14)
      value(locy)=value(locy)+gm-ggd
      locy=lvn+nodplc(loc+15)
      value(locy)=value(locy)-gds-gm
      locy=lvn+nodplc(loc+16)
      value(locy)=value(locy)-ggs-gm
      locy=lvn+nodplc(loc+17)
      value(locy)=value(locy)-gspr
      locy=lvn+nodplc(loc+18)
      value(locy)=value(locy)-gds
 1000 loc=nodplc(loc)
      go to 10
      end
```

Appendix 2

Derivation of G_{ds} and G_m equations

The conductance equations used in the modified SPICE code are derived here. The basic Curtice [19] current equation is given by:

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

Given that:

$$\frac{\delta(\tanh u)}{\delta u} = \mathrm{sech}^2 u$$

Then G_{ds} is given by:

$$G_{ds} = \frac{\partial I_{ds}}{\partial V_{ds}}$$

$$= \beta(V_{gs}-V_t)^2\left[\frac{\partial\tanh(\alpha V_{ds})}{\partial V_{ds}}+\lambda\frac{\partial(V_{ds}\tanh(\alpha V_{ds}))}{\partial V_{ds}}\right]$$

$$= \beta(V_{gs}-V_t)^2\left[\alpha\,\mathrm{sech}^2(\alpha V_{ds})+\lambda(\tanh(\alpha V_{ds})+\alpha V_{ds}\mathrm{sech}^2(\alpha V_{ds})\right]$$

$$= \beta(V_{gs}-V_t)^2\left[\alpha\,\mathrm{sech}^2(\alpha V_{ds})(1+\lambda V_{ds})+\lambda\tanh(\alpha V_{ds})\right]$$

G_m is given by:

$$G_m = \frac{\partial I_{ds}}{\partial V_{gs}}$$

$$= 2\beta(1+\lambda V_{ds})(V_{gs}-V_t)$$

In the reverse mode of operation, the same equations apply, except that all instances of V_{ds} are now replaced by $-V_{ds}$.

Appendix 3

UNIX "makefile" for compiling modified-SPICE routines

The "makefile" in the UNIX operating system allows any number of files to be compiled and linked together to produce one binary file for execution. To execute the commands in the makefile, the following UNIX command is used:

```
make makefile
```

The "makefile" itself is generally of the form:

```
<target> : <sub-targets>

<UNIX commands>
```

The words within brackets designate meanings rather than the actual words used in the "makefile". Although the "make" command is described in the UNIX manual pages, the exact format and execution of the "makefile" is unclear, and is therefore discussed in some detail here.

The execution of the <UNIX commands> within the "makefile" depends on the first <target> in the file. A <target> is simply a name given to represent the string of commands to be executed under that target. The first <target> forms the top of the target tree of the "makefile". The <sub-targets> specified by the target form the branches of the tree one level lower. Each sub-target can then specify other <sub-target> and <UNIX commands> entries using the same format, and so on indefinitely. If a <sub-target> does not form any further <target> entry in the "makefile", then that branch of the target tree is terminated. Execution of the commands under a target entry is from bottom of the tree upwards. In other words, the target tree is searched until a terminated branch is reached. The commands under that branch is executed, and control moves one level up to search for other terminations. This process carries on until the top target and its associated commands are executed.

The relationship between a target/sub-target and the files to be compiled (in whatever programming language) is as follows. A <sub-target> can either be filenames or names that represent the commands associated with the sub-target.

If the sub-targets are filenames, then the commands are executed if and only if any one of those files have been modified.

If the sub-target is not a filename, then the associated commands are always executed. This is equivalent to searching for a filename that does not exist in practice, and is automatically assumed to have been modified.

A dummy target entry is possible if the there is no sub-targets under a target entry. Any commands under that target entry will only be executed if the following UNIX command is used:

```
make target
```

The "makefile" for the modified SPICE subroutines is listed below:

```
b = /u/user2/ee/elee48/gaspice/bin
l = ../lib

cflags = -o $(defines)
fflags = -w

objects = spice.o ../acan.o ../dcop.o ../dctran1.o dctran2.o \
errchk.o ../ovtpvt.o readin.o ../setup.o jfet.o \
modchk.o find.o addelt.o unix.o

spice2g6: $(objects)
    fortran -o spice2g6 $(objects)

install: spice2g6 $l
    strip spice2g6; mv spice2g6 /u/user2/ee/elee48/bin/ecips

clean:
    rm -f *.o spice2g6

$l:
    mkdir $l
```

Appendix 4

Clock generator SPICE input file

```
 the clock generator obviously
*
* this version has the following features
* (1) one dfet pull-up.
* (2) one efet pull-down.
* (3) the dfet is driven by a level shifter
* conected to vdd and -2.8v
*     equiv. to -4.5v in ecl 100k series. it has
* all dfet devices for
*     large current (and large power)
* (4) the efet is driven by a level shifter
* connected to vdd and -0.6v,
*     of similar construction as that drving the dfet.
* (5) a dfet and cap. load are connected to simulate
* pass transistors
*     etc.
* the power is rather high due to the -2.8v supply,
* as opposed to the
* previous -1.7v
*
 *
 .options limpts=50000 itl5=0
 .width out=75
 *
 *
 .subckt inv1 3 2 1
 jd1 1 3 3 jdep 6.67 off
 je1 3 2 0 jenh 20 off
 c1 2 0 20f
 .ends
 *
 .subckt inv1a5 3 2 1
 jd1 1 3 3 jdep 10
 je1 3 2 0 jenh 30
 c1 2 0 25f
 .ends
 *
 .subckt inv2 3 2 1
 jd1 1 3 3 jdep 13.5 off
 je1 3 2 0 jenh 40 off
 c1 2 0 20f
 .ends
 *
 .subckt inv3 3 2 1
 jd1 1 3 3 jdep 20 off
 je1 3 2 0 jenh 60 off
 c1 2 0 20f
 .ends
 *
 .subckt nor1 4 3 2 1
 jd1 1 4 4 jdep 6.67 off
 je1 4 3 0 jenh 20 off
 je2 4 2 0 jenh 20 off
 c1 2 0 25f
 c2 3 0 25f
 .ends
 *
 .subckt nor2 4 3 2 1
 jd1 1 4 4 jdep 13.5 off
 je1 4 3 0 jenh 40 off
 je2 4 2 0 jenh 40 off
 c1 2 0 25f
 c2 3 0 25f
 .ends
 *
 .subckt nor3 4 3 2 1
 jd1 1 4 4 jdep 20 off
 je1 4 3 0 jenh 60 off
 je2 4 2 0 jenh 60 off
 c1 2 0 25f
 c2 3 0 25f
 .ends
 *
 .subckt nor4 4 3 2 1
 jd1 1 4 4 jdep 26.7 off
 je1 4 3 0 jenh 80 off
 je2 4 2 0 jenh 80 off
 c1 2 0 25f
 c2 3 0 25f
 .ends
 *
 .subckt latch 5 2 3 7 1
 jdp1 2 3 4 jdep 20 off
 xinv1 5 4 1 inv2
 xinv2 6 5 1 inv1
 jdp2 6 7 4 jdep 20 off
 c1 2 0 20f
 c2 3 0 20f
 c3 6 0 20f
 c4 7 0 20f
 c5 4 0 20f
 .ends
 *
 .subckt delay 6 3 4 5 1
 xlatch1 2 3 4 5 1 latch
 xlatch2 6 2 5 4 1 latch
 .ends
 *
 *
 .subckt lspu 4 3 2 1 6
 je1 1 2 3 jdep 30
 jdi 4 3 4 jdep 20
* jd2 5 4 5 jdep 20
 jd1 4 6 6 jdep 80
 c1 2 0 25f
 .ends
 *
 .subckt lspd 4 3 2 1 6
 je1 1 2 3 jdep 60
 jdi 4 3 4 jdep 60
* jd2 5 4 5 jdep 20
 jd1 4 6 6 jdep 100
 c1 2 0 25f
 .ends
*
 .subckt clock 16 17 4 5 30 3 12 13 2 1 100 200 120 7
 *
 xinv00 20 2 1 inv3
 xinv01 30 20 1 inv3
 xinv1 3 30 1 inv3
 xnor1 4 5 30 1 nor3
 xnor2 5 4 3 1 nor3
 *
 xinv2 6 4 1 inv2
 xinv3 7 6 1 inv3
 xinv4 8 4 1 inv2
 *
 xinv5 9 5 1 inv2
```

```
xinv6 10 9 1 inv3
xinv7 11 5 1 inv2
*
xls1 12 120 7 1 100 lspu
xls2 13 130 8 1 200 lspd
*
xls3 14 140 10 1 100 lspu
xls4 15 150 11 1 200 lspd
*
xbuff1 16 12 13 1 200 buff
xbuff2 17 14 15 1 200 buff
*
.ends
*
.subckt buff 4 2 3 1 200
jd1 1 2 4 jdep 400 off
jd2 4 3 200 jdep 50 off
jd3 4 3 200 jenh 400 off
c1 2 0 20f
c2 3 0 20f
.ends
*
vdd 1 0 1.7
vss 100 0 -2.8
vss1 200 0 -0.6
*
vin 2 0 sin(0.40 0.35 600e6 0 0)
*vin1 20 0 pulse(0.1 0.75 0.64n 0.2n 0.2n 1.8n 4n)
*
*vphi1 3 0 pulse(-0.6 0.7 0 0.2n 0.2n 0.6n 2n)
*vphi2 4 0 pulse(-0.6 0.7 1n 0.2n 0.2n 0.6n 2n)
*
xclock 3 4 5 6 7 8 9 10 2 1 100 200 120 122 clock
c1 3 0 3p
c2 4 0 3p
*
*xdel1 11 20 3 4 1 delay
*
```

Appendix 5

Multiplier slice SPICE input file

This input file is for a middle slice in a Modified-Booth's Algorithm multiplier.

```
 to investigate a bit
 *
 .options limpts=10000000 itl5=0
 .width out=75
 *
 *
 .subckt latch 1 2 3 4 5 6 7
 jdpass1 2 3 4 jdep 20 off
 *
 * note that the phi2 inverter is 2 times
* larger than the
 * phi1 inverter to make sure that in case
* both pass transistors
 * are slightly on at the same time, the input
* data always wins
 je1 5 4 0 jenh 40 off
 jd1 1 5 5 jdep 13.5 off
 je2 6 5 0 jenh 20 off
 jd2 1 6 6 jdep 6.5 off
 *
 jdpass2 6 7 4 jdep 20 off
 c1 2 0 20f
 c2 4 0 20f
 c3 5 0 20f
 c4 6 0 20f
 c5 3 0 20f
 c6 7 0 20f
 .ends
 *
 * latch1 is different only in the input tg
* and inverter size
 *
 .subckt latch1 1 2 3 4 5 6 7
 jdpass1 2 3 4 jdep 30 off
 *
 * note that the phi2 inverter is 2 times
* larger than the
 * phi1 inverter to make sure that in
* case both pass transistors
 * are slightly on at the same time, the input
* data always wins
 je1 5 4 0 jenh 60 off
 jd1 1 5 5 jdep 20 off
 je2 6 5 0 jenh 20 off
 jd2 1 6 6 jdep 6.5 off
 *
 jdpass2 6 7 4 jdep 30 off
 c1 2 0 20f
 c2 4 0 20f
 c3 5 0 20f
 c4 6 0 20f
 c5 3 0 20f
 c6 7 0 20f
 .ends
 *
```

```
*
 *
 .subckt latch2 5 3 2 1
 * this is a simple inverting latch with
* a static refreshing
 * path rather then the two phase stuff in
* latch1 and 2
 *
 * the nodes are:
 * 5 = output
 * 2 = input
 * 3 = latch signal or phi2
 *
 jd1pass1 2 3 4 jdep 20 off
 jd1 1 5 5 jdep 13.5
 je1 5 4 0 jenh 40
 xinv1 6 5 1 inv
 jdpass2 6 0 4 jdep 10
jdiode 4 6 4 jdep 5
 *
 .ends
 *
 .subckt latchcl 1 2 3 4 5 6 7 8
 jdpass1 2 3 4 jdep 20 off
 *
 * this is for a reset function, so that the
* lsb of the
*  datax2 can be forced to 0
*
 je1 5 4 0 jenh 40 off
 jd1 1 5 5 jdep 13.5 off
 je3 5 8 0 jenh 40 off
 je2 6 5 0 jenh 20 off
 jd2 1 6 6 jdep 6.5 off
 *
 jdpass2 6 7 4 jdep 20 off
 c1 2 0 20f
 c2 4 0 20f
 c3 5 0 20f
 c4 6 0 20f
 c5 3 0 20f
 c6 7 0 20f
 .ends
*
 .subckt delay 9 2 1 3 7
 xlatch1 1 2 3 4 5 6 7 latch
 xlatch2 1 5 7 8 9 10 3 latch
 .ends
 *
 * delay1 is made for inhibiting the carbor signal
* using the lsb pulse
 * at node 12
 *
 .subckt delay1 9 2 1 3 7
 xlatch1 1 2 3 4 5 6 7 latch
 *
 * xlatch2 has an enlarged input tg
 *
 xlatch2 1 5 7 8 9 10 3 latch1
 .ends
 *
 * delay2 is identical to delay except the extra
* output node at the
 * slave latch tg
 *
 .subckt delay2 9 8 2 1 3 7
 xlatch1 1 2 3 4 5 6 7 latch
 xlatch2 1 5 7 8 9 10 3 latch1
 .ends
*
```

```
* delay3 has a static refreshed master stage
* so the gating can be
* control by logic rather than simply phi1 or 2
*
 .subckt delay3 9 8 2 1 3 7
 xlatch1 1 2 3 4 5 6 7 latch2
 xlatch2 1 5 7 8 9 10 3 latch
 .ends
*
 *
 *
 .subckt reg 8 1 2 3 6 30 60 9 100
 xlat1 1 2 3 4 5 99 60 latch
 xlat2 1 5 6 7 8 98 30 latch
* the tg and inverter are for loading the
* start bit into the
* master stage of the register
 jdepass2 9 100 4 jdep 40 off
 c1 4 0 20f
 c2 100 0 20f
 c3 9 0 20f
 .ends
*
* this is a level shifter specifically for
* driving the huge buffers
* within the clock generator. the supplies are 1.
* 7v and -2.5v.
*
 .subckt level 5 2 1 100
  jd1 1 2 3 jenh 100
  jd2 4 3 4 jdep 100
  jd3 5 4 5 jdep 100
  jd5 5 100 100 jdep 45
  c1 2 0 20f
  c2 3 0 20f
  c3 4 0 20f
  c4 5 0 20f
 .ends level
 *
 *
 *
 * the next subckt is the
* level shifter specifically for
 * shifting a single control signal to drive at
* least 6 tg’s
 *
 .subckt level1 5 2 1 6
 jd1 1 2 3 jdep 30 off
 jdio1 4 3 4  jdep 20 off
 jdio2 5 4 5 jdep 20 off
 jd2 5 6 6 jdep 10 off
 c1 2 0 10f
 .ends
*
*
 .subckt buf 4 2 3 5 1
  jd1 1 2 4 jdep 600 off
  jd2 4 3 5 jdep 400 off
  c1 2 0 20f
  c2 3 0 20f
 .ends
 *
 .subckt inv 3 2 1
 je1 3 2 0 jenh 20 off
 jd1 1 3 3 jdep 6.5 off
 c1 2 0 20f
 .ends
 *
 .subckt inv1 3 2 1
 je1 3 2 0 jenh 40 off
 jd1 1 3 3 jdep 13.5 off
 c1 2 0 20f
 .ends
*
 je1 3 2 0 jenh 56
 jd1 1 3 3 jdep 18.5
 c1 2 0 10f
*
 .subckt xnor 2 3 4 5 6 1
 xnor1 7 3 4 1 nor
 xnor2 8 5 6 1 nor
 xnor3 2 7 8 1 nor
 .ends
 *
 .subckt nor 4 3 2 1
 jd1 1 4 4 jdep 6.5 off
 je1 4 2 0 jenh 20 off
 je2 4 3 0 jenh 20 off
 c1 2 0 20f
 c2 3 0 20f
 .ends
 *
 .subckt nor3 5 4 3 2 1
 jd1 1 5 5 jdep 6.5 off
 je1 5 2 0 jenh 20 off
 je2 5 3 0 jenh 20 off
 je3 5 4 0 jenh 20 off
 c1 2 0 20f
 c2 3 0 20f
 c3 4 0 20f
 .ends
*
 .subckt nor4 6 5 4 3 2 1
 jd1 1 6 6 jdep 6.5 off
 je1 6 5 0 jenh 20 off
 je2 6 4 0 jenh 20 off
 je3 6 3 0 jenh 20 off
 je4 6 2 0 jenh 20 off
 c1 2 0 20f
 c2 3 0 20f
 c3 4 0 20f
 c4 5 0 20f
*
 .subckt bignor2 5 3 2 1
 jd1 1 5 5 jdep 10 off
 je1 5 2 0 jenh 80 off
 je2 5 3 0 jenh 80 off
 c1 2 0 20f
 c2 3 0 20f
 .ends
*
*
 .subckt bignor3 5 4 3 2 1
 jd1 1 5 5 jdep 10 off
 je1 5 2 0 jenh 80 off
 je2 5 3 0 jenh 80 off
 je3 5 4 0 jenh 80 off
 c1 2 0 20f
 c2 3 0 20f
 c3 4 0 20f
 .ends
*
 .subckt bignor4 5 3 2 1
 jd1 1 5 5 jdep 6 off
 je1 5 2 0 jenh 48 off
 je2 5 3 0 jenh 48 off
 c1 2 0 20f
 c2 3 0 20f
 .ends
*
```

```
*
jdpass 2 3 4 jdep 20 off
*
jd1 1 5 5 jdep 10 off
je1 5 4 0 jenh 80 off
jd2 1 6 6 jdep 6.5 off
je2 6 5 0 jenh 20 off
jd3 1 7 7 jdep 6.5 off
je3 7 5 0 jenh 20 off
*
jdres 7 0 4 jdep 10 off
jdiode 4 7 4 jdep 5
c1 3 0 20f
c2 5 0 35f
c3 6 0 20f
c4 7 0 20f
c5 2 0 20f
c6 4 0 35f
*
* this is the clock generator
* complete with clock inhibit
* please refer to ’cl9’ for full details.
*
 .subckt clockgen 23 20 2 11 12 1 200 100
 *
 xinv0 3 2 1 inv1
 xinv1 24 3 1 inv1
 xinva 25 24 1 inv1
 *
 *
 *
 * cross-coupled nor’s
 *
 xnor1 4 24 9 1 bignor2
 xnor2 9 25 4 1 bignor2
 *
 * buffering to drive level shifters for push-pull stage
 *
 *
 xinv2 5 4 1 inv1
 xinv6 10 9 1 inv1
 *
 *
 xinv10 14 4 1 inv1
 xinv11 15 5 1 inv1
 *
 xlevel1 18 14 1 100 level
 xlevel2 19 15 1 100 level
 *
 xbuf1 20 19 18 200 1 buf
 *
 xinv12 16 9 1 inv1
 xinv13 17 10 0 0 1 bignor3
 *
 xlevel3 21 16 1 100 level
 xlevel4 22 17 1 100 level
 *
 xbuf2 23 22 21 200 1 buf
 *
 .ends
*
* node conventions for datcon1,2
*
* cn-1 = 2
* cn   = 3
* cn+1 = 4
* cn-1’ = 5
* cn’  = 6
* cn+1’ = 7
*
```

```
* others are internal signals
*
* datcon1 produces signal to select 1 x data
*
* datcon2 produces signal to select 2 x data
*
* together they perform the booth's recoding
*
* actually the signal generated (10) is (select data)'
xnor1 8 3 5 1 nor
xnor2 9 6 2 1 nor
xnor3 10 9 8 1 nor
*
* as above, the signal (10) is (select datax2)'
xnor1 8 7 3 2 1 nor3
xnor2 9 4 5 6 1 nor3
xnor3 10 9 8 1 nor
*
*
* for the next subckt:
*
*    2 = (1 x data)'
*    3 = datcon1          (from datcon1)
*    4 = (2 x data)'
*    5 = datcon2          (from datcon2)
*
*  this does the actual xnor to get datax1,
* datax2, or 1 (if
*  both 3 & 5 are 1)
*
* note the output (8) is (selected data)', and is
* then latched later on
xnor1 6 3 2 1 nor
xnor2 7 5 4 1 nor
xnor3 8 7 6 1 bignor4
*
*
* for the sum and carry/borrow subckts
*
* a = 2
* b = 3
* c = 4     (can be carry or borrow)
* a' = 5
* b' = 6
* c' = 7
* these can be reversed to give sum' at node 12
xnor1 8 4 3 2 1 nor3
xnor2 9 6 5 4 1 nor3
xnor3 10 7 6 2 1 nor3
xnor4 11 7 5 3 1 nor3
xnor5 12 11 10 9 8 1 nor4
*
*
* for the next subckt only:
*
* cn+1  = 8
* cn+1' = 9
*
* cn+1 is the msb in the coeff bit pair
* if it is 1 then subtract (same as 2's complement conv.)
* if 0 then add
*
*
*
* circuit for (b + a) and (b - a)
*
* note that the order of nodes is essential
* 12 = m
* 2 = data
* 4 = ci
* 16 = co
xnor4 13 12 2 1 nor
xnor5 14 12 4 1 nor
xnor6 15 2 4 1 nor
xnor7 16 15 14 13 17 1 nor4
*
*
* three (rather awkward) power rails
*
vdd 1 0 1.7
*vss1 2 0 -0.6
*vss2 3 0 -2.5
*
* the sine wave: (offset amplitude freq delay decay)
*
*vclock 4 0 sin(0.35 0.35 500e+06 0 0)
*
* data = 110011 (only true for one word !)
* ppsi = 010101
*
vlsb 7 0 pulse(0.1 0.75 20p 200p 200p 1.8n)
vdata 11 0 pulse(0.1 0.75 20p 200p 200p 3.8n 8n)
vppsi 100 0 pulse (0.1 0.75 20p 200p 200p 1.8n 4n)
*
* 5 = phi1 uninihited
* 6 = phi2 uninhibited
*
*
* these are coefficients = 100
*
vcn+1 10 0 0.75
vcn 21 0 0
vcn-1 22 0 0
*xclock 5 6 4 0 0 1 2 3 clockgen
*
* the following are delays to
* produce realistic lsb, and
* data signals from ideal pulses. see
* notes for node designations.
* also sets up ppsi at node 101
* 8=lsb
* 12=data
* 101= ppsi
xdelay1 8 7 1 6 5 delay1
xdelay2 12 11 1 6 5 delay
xdelay3 101 100 1 6 5 delay
*
*
* here comes the real circuit
*
*
* 3 delay stages for lsb. to inhibit the carry/
* borrow for the next
* slice
*
*
xdelay4 14 8 1 6 5 delay
xdelay5 15 14 1 6 5 delay
xdelay6 16 15 1 6 5 delay
*
*
* sets up the complements of the coeffs.
*
*
xinv1 23 10 1 inv
xinv2 24 21 1 inv
xinv3 25 22 1 inv
*
* decodes the coeff. to select the data or datax2
*
```

```
xseldata 26 22 21 25 24 1 datcon1
xsel2data 27 22 21 10 25 24 23 1 datcon2
*
*
* produce data'
xinv4 30 12 1 inv
*
* this taps the data stream for the
* datax1 and datax2 signals
*
xdat 320 30 26 29 27 1 dat
*
* phi2 latch to give selected data for
* sum/diff circuit later on
* 32 = selected data
* 33 = 32' = selected data'
xlatch1 1 320 6 321 32 322 5 latch
* a large inverter to get selected data'
xinv5 33 32 1 inv2
*
* generates phi1 latched selected data for
* carry/borrow circuit later on
xlatch2 1 33 5 51 52 53 6 latch
*
* the following delays are for the serial data.
* the output of the
* first one is tapped to give datax2. the tapping is
* done at the output
* of the tg at the slave stage.
*
* node 12 is the (real) data signal
* node 41 is the data signal for the next slice
* node 293 is the phi2 latch data'
*
xlatch3 1 12 6 292 293 294 5 latch
xdelay7 31 293 1 5 6 delay
xdelay8 40 31 1 5 6 delay
xlatch4 1 40 5 410 41 411 6 latch
* 29 = (lsb inhibited datax2)' = 291'
* (goes into dat earlier on)
*
xlsbcl 1 293 5 290 291 295 6 8 latchcl
xinvx 29 291 1 inv
*
* 39 = phi2 latched ppsi'
* 36 = 39' = ppsi (logically)
xlatch5 1 101 6 390 39 391 5 latch
*
xinv6 36 39 1 inv1
* note that inputs are swapped in polarity to give sum'
*  35 = ci
*  37 = ci'
* 34 = ppso
* 55 = phi1 latched ppso
*
xpsum 34 35 36 32 37 39 33 1 psum
xlatch9 1 34 5 54 55 56 6 latch
* generate m' for carry/borrow
xmbar 38 39 10 36 0 1 xnor
* phi1 latch before entering carbor
xlatch6 1 38 5 42 43 44 6 latch
*
xcarbor 45 43 49 52 8 1 carbor
*
* phi2 latch for ci (35) and ci' (37) inputs of sum.diff
xlatch7 1 45 6 46 37 47 5 latch1
xinv7 35 37 1 inv
*
xlatch8 1 37 5 48 49 50 6 latch
```

Appendix 6

Parallel-to-serial converter SPICE input file

```
p/s and s/p converters cascaded(p in p out)
 *
 .options limpts=1000000 itl5=0
 .width out=75
 *
 *
 *
 .subckt latch 1 2 3 4 5 6 7
 jdpass1 2 3 4 jdep 20 off
 *
 * note that the phi2 inverter is 3 times
 * larger than the
 * phi1 inverter to make sure that in
 * case both pass transistors
 * are slightly on at the same time, the
 * input data always wins
 je1 5 4 0 jenh 40 off
 jd1 1 5 5 jdep 13.5 off
 je2 6 5 0 jenh 20 off
 jd2 1 6 6 jdep 6.5 off
 *
 jdpass2 6 7 4 jdep 20 off
 c1 2 0 8f
 c2 4 0 8f
 c3 5 0 8f
 c4 6 0 8f
 c5 3 0 8f
 c6 7 0 8f
 .ends
 *
 .subckt latch1 1 2 3 4 5 6 7
 jdpass1 2 3 4 jdep 40  off
 *
 * note that the phi2 inverter is 3
 * times larger than the
 * phi1 inverter to make sure that in
 * case both pass transistors
 * are slightly on at the same time, the
 * input data always wins
 je1 5 4 0 jenh 40 off
 jd1 1 5 5 jdep 13.5 off
 je2 6 5 0 jenh 20 off
 jd2 1 6 6 jdep 6.5 off
 *
 jdpass2 6 7 4 jdep 40 off
 c1 2 0 8f
 c2 4 0 8f
 c3 5 0 8f
 c4 6 0 8f
 c5 3 0 8f
 c6 7 0 8f
 .ends
*
 .subckt delay 9 2 1 3 7
 xlatch1 1 2 3 4 5 6 7 latch
 xlatch2 1 5 7 8 9 10 3 latch
 .ends
 *
 *
 *
 .subckt ct1reg 13 11 2 101 100 102 3 7 1
* 13 = del(data)
* 11 = del(data)'
 xlat1 1 2 3 4 5 6 7 latch
* the 2 pass transistors are for loading or shifting
* an inverter is used to "restore" the signal
* 100 = load'
* 102 = load
* 101 = preset value
 jdp1 5 100 8 jdep 20
xinv1 9 8 1 inv1
jdp2 101 102 8 jdep 20
* second inverter to give correct polarity
 xlat2 1 9 7 10 11 12 3 latch
xinv2 13 11 1 inv
 c1 4 0 8f
 c2 100 0 8f
 c3 9 0 8f
 c4 5 0 15f
 c5 102 0 20f
 .ends
*
 .subckt ct2reg 13 11 2 101 100 102 3 7 1
* almost identical to ct1reg, except auto-pullup
* 13 = del(data)
* 11 = del(data)'
 xlat1 1 2 3 4 5 6 7 latch
* the 2 pass transistors are for loading or shifting
* an inverter is used to "restore" the signal
* 100 = load'
* 102 = load
* 101 = preset value
 jdp1 5 100 8 jdep 20
xinv1 9 8 1 inv1
jdp2 101 102 8 jdep 20
* extra pull-up to force register output to 0 when not
* loading or shifting
jd1 1 8 8 jdep 3.5 off
* second inverter to give correct polarity
 xlat2 1 9 7 10 11 12 3 latch
xinv2 13 11 1 inv
 c1 4 0 8f
 c2 100 0 8f
 c3 9 0 8f
 c4 5 0 15f
 c5 102 0 20f
 .ends
*
 .subckt datareg 13 11 2 101 100 102 3 7 1
* 13 = del(data)
* 11 = del(data)'
 xlat1 1 2 3 4 5 6 7 latch
* the 2 pass transistors are for loading or shifting
* an inverter is used to "restore" the signal
* 100 = load'
* 102 = load
* 101 = preset value, which is inverted at the
* output (13)
 jdp1 5 100 8 jdep 20
xinv1 9 8 1 inv1
* passive feedback for retaining data if
* neither shifting nor loading
xinv2 14 9 1 inv
jdres 14 0 8 jdep 10
jdiode 8 14 8 jdep 5
jdp2 101 102 8 jdep 20
* second inverter to give correct polarity
 xlat2 1 9 7 10 11 12 3 latch
```

```
xinv3 13 11 1 inv
 c1 4 0 8f
 c2 100 0 8f
 c3 9 0 8f
 c4 5 0 15f
 c5 102 0 20f
 .ends
 *
 .subckt inv 3 2 1
 je1 3 2 0 jenh 20 off
 jd1 1 3 3 jdep 6.5 off
 c1 2 0 8f
 .ends
*
 .subckt inv1 3 2 1
 je1 3 2 0 jenh 40 off
 jd1 1 3 3 jdep 13.5 off
 c1 2 0 8f
 .ends
 *
jd1 1 4 4 jdep 6.5 off
je2 4 3 5 jenh 40 off
je3 5 2 0 jenh 40 off
c1 2 0 20f
c2 3 0 20f
*
 .subckt xnor 2 3 4 5 6 1
 xnor1 7 3 4 1 nor
 xnor2 8 5 6 1 nor
 xnor3 2 7 8 1 nor
 .ends
 *
 *
 .subckt endnor 7 2 3 4 5 6 1
 jd1 1 7 7 jdep 6.5 off
 je1 7 2 0 jenh 20 off
 je2 7 3 0 jenh 20 off
 je3 7 4 0 jenh 20 off
 je4 7 5 0 jenh 20 off
 je5 7 6 0 jenh 20 off
 c1 2 0 8f
 c2 3 0 8f
 c3 4 0 8f
 c4 5 0 8f
 c5 6 0 8f
 .ends
 *
 *
 .subckt nor 4 3 2 1
 jd1 1 4 4 jdep 6.5 off
 je1 4 2 0 jenh 20 off
 je2 4 3 0 jenh 20 off
 c1 2 0 8f
 c2 3 0 8f
 .ends
 *
 .subckt nor3 5 4 3 2 1
 jd1 1 5 5 jdep 6.5 off
 je1 5 2 0 jenh 20 off
 je2 5 3 0 jenh 20 off
 je3 5 4 0 jenh 20 off
 c1 2 0 8f
 c2 3 0 8f
 c3 4 0 8f
 .ends
*
je1 1 2 3 jdep 15 off
jdi 4 3 4 jdep 20 off
jd1 4 5 5 jdep 55 off
c1 2 0 15f
```

```
*
*
 .subckt prbs1 2 3 4 5 6 7 8 9 10 11 101 102
+ 1000 2000 1
 *
*
 * preset word = 10000, note inversion at inputs
*also the reg word is read in this node order
* 2-4-6-8-10
 xreg1 2 3 12 0 101 102 1000 2000 1 ct1reg
 xreg2 4 5 2 1 101 102 1000 2000 1 ct1reg
 xreg3 6 7 4 1 101 102 1000 2000 1 ct1reg
 xreg4 8 9 6 1 101 102 1000 2000 1 ct1reg
 xreg5 10 11 8 1 101 102 1000 2000 1 ct1reg
 *
 xxnor 12 4 11 5 10 1 xnor
 *
 .ends
 *
 .subckt prbs2 2 3 4 5 6 7 8 9 10 11 101 102
+ 1000 2000 1
 *
 * preset word = 00111
* note that owrd is read in this order: 2-4-6-8-10
 xreg1 2 3 12 1 101 102 1000 2000 1 ct2reg
 xreg2 4 5 2 1 101 102 1000 2000 1 ct2reg
 xreg3 6 7 4 0 101 102 1000 2000 1 ct2reg
 xreg4 8 9 6 0 101 102 1000 2000 1 ct2reg
 xreg5 10 11 8 0 101 102 1000 2000 1 ct2reg
 *
 xxnor 12 4 11 5 10 1 xnor
 *
 .ends
*
*.subckt ptos
*
xprbs1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 1 prbs1
* 2-11 = regs outputs and complements
* 12 = load' = shift
* 13 = load
* 14 = phi2
* 15 = phi1
*
xprbs2 16 17 18 19 20 21 22 23 24 25 26 13 14
+ 15 1 prbs2
* 16 - 25 are the outputs of the 5 regs and
* their compl.
* 26 = shift
* 13 = load
* 28 = ep1
* 29 = ep2
xendnor1 28 3 5 7 9 0 1 endnor
xendnor2 29 16 18 20 22 24 1 endnor
*
xlat1 1 28 14 30 31 32 15 latch1
xinv1 33 30 1 inv
xinv2 34 31 1 inv
* 13 = load both prbs1/2
* 12 = shift prbs1 = 44' logically
xlsload 13 34 1 100 ls
xlshift1 12 33 1 100 ls
* 36 = del(ep1)
* 39 = 1.5del(ep1)'
xlat2 1 31 15 35 36 37 14 latch
xlat3 1 36 14 38 39 40 15 latch
* 42 = lsb pulse
xlat4 1 39 15 41 42 43 14 latch
* 47 = to load parallel data bits
xinv3 46 39 1 inv1
xlsldata 47 46 1 100 ls
```

```
*
* 48 = del(ep1)', used for nand gate for shift serial
* data later on
xinv4 48 36 1 inv
* 50 = 0.5del(ep2)'
xlat5 1 29 14 49 50 51 15 latch1
* 26 = shift prbs2
xinv5 52 49 1 inv
xlshift2 26 52 1 100 ls
* 55 = del(ep2)
* 57 = inhibit data
xlat6 1 50 15 54 55 56 14 latch
xinhdata 57 55 1 inv
* produce shift data signal
* only nand gate in whole design
* 63 = shift data signal
xnan1 58 48 55 1 nand
xlat7 1 58 14 59 60 61 15 latch1
xinv6 62 59 1 inv
xlshdata 63 62 1 100 ls
 * the following are arbitrary parallel inputs
* to the parallel regs
 *
 *
 * the parallel/serial data regs.
 * parallel data = 01010101 (note inversion in
* subcircuit call)
* 85 = lsb
* 64 = msb
 * sign bit/msb is repeated until next word
 xdatreg1 64 65 64 1 63 47 14 15 1 datareg
 xdatreg2 67 68 64 0 63 47 14 15 1 datareg
 xdatreg3 70 71 67 1 63 47 14 15 1 datareg
 xdatreg4 73 74 70 0 63 47 14 15 1 datareg
 xdatreg5 76 77 73 1 63 47 14 15 1 datareg
 xdatreg6 79 80 76 0 63 47 14 15 1 datareg
 xdatreg7 82 83 79 1 63 47 14 15 1 datareg
 xdatreg8 85 86 82 0 63 47 14 15 1 datareg
* nor gate for inhibiting data
* 88 = coded bit stream
* 89 = 1 cycle delayed data
xnor 88 86 57 1 nor
xdel1 89 88 1 14 15 delay
* 90 = 1 cycle delayed lsb
xdel2 90 42 1 14 15 delay
 *
*.ends
*
vdd 1 0 1.7
vss 100 0 -0.6
*
* artificial clocks
vphi1 15 0 pulse(-0.5 0.5 20ps 200ps 200ps 0.6n 2n)
vphi2 14 0 pulse(-0.5 0.5 1020ps 200ps 200ps 0.6n 2n)
*
```

Appendix 7

Serial-to-parallel converter SPICE input file

```
p/s and s/p converters cascaded(p in p out)
 *
 .options limpts=1000000 itl5=0
 .width out=75
 *
 *
 *
 .subckt latch 1 2 3 4 5 6 7
 jdpass1 2 3 4 jdep 20 off
 *
 * note that the phi2 inverter is
* 3 times larger than the
 * phi1 inverter to make sure that in case
* both pass transistors
 * are slightly on at the same time, the input
* data always wins
 je1 5 4 0 jenh 40 off
 jd1 1 5 5 jdep 13.5 off
 je2 6 5 0 jenh 20 off
 jd2 1 6 6 jdep 6.5 off
 *
 jdpass2 6 7 4 jdep 20 off
 c1 2 0 20f
 c2 4 0 20f
 c3 5 0 20f
 c4 6 0 20f
 c5 3 0 20f
 c6 7 0 20f
 .ends
 *
 .subckt latch1 1 2 3 4 5 6 7
 jdpass1 2 3 4 jdep 40  off
 *
 * note that the phi2 inverter is
* 3 times larger than the
 * phi1 inverter to make sure that in case
* both pass transistors
 * are slightly on at the same time, the input
* data always wins
 je1 5 4 0 jenh 40 off
 jd1 1 5 5 jdep 13.5 off
 je2 6 5 0 jenh 20 off
 jd2 1 6 6 jdep 6.5 off
 *
 jdpass2 6 7 4 jdep 40 off
 c1 2 0 20f
 c2 4 0 20f
 c3 5 0 20f
 c4 6 0 20f
 c5 3 0 20f
 c6 7 0 20f
 .ends
*
 .subckt delay 9 2 1 3 7
 xlatch1 1 2 3 4 5 6 7 latch
 xlatch2 1 5 7 8 9 10 3 latch
 .ends
 *
 *
 *
 .subckt ct1reg 13 11 2 101 100 102 3 7 1
* 13 = del(data)
* 11 = del(data)’
 xlat1 1 2 3 4 5 6 7 latch
* the 2 pass transistors are for loading
* or shifting
* an inverter is used to "restore" the signal
* 100 = load’
* 102 = load
* 101 = preset value
 jdp1 5 100 8 jdep 20
xinv1 9 8 1 inv1
jdp2 101 102 8 jdep 20
* second inverter to give correct polarity
 xlat2 1 9 7 10 11 12 3 latch
xinv2 13 11 1 inv
 c1 4 0 20f
 c2 100 0 20f
 c3 9 0 20f
 c4 5 0 30f
 c5 102 0 20f
 .ends
*
 *
 .subckt inv 3 2 1
 je1 3 2 0 jenh 20 off
 jd1 1 3 3 jdep 6.5 off
 c1 2 0 20f
 .ends
*
 .subckt inv1 3 2 1
 je1 3 2 0 jenh 40 off
 jd1 1 3 3 jdep 13.5 off
 c1 2 0 20f
 .ends
 *
 .subckt biginv 3 2 1
 je1 3 2 0 jenh 80 off
 jd1 1 3 3 jdep 10 off
 c1 2 0 20f
 .ends
*
*
 .subckt xnor 2 3 4 5 6 1
 xnor1 7 3 4 1 nor
 xnor2 8 5 6 1 nor
 xnor3 2 7 8 1 nor
 .ends
 *
 *
 .subckt endnor 7 2 3 4 5 6 1
 jd1 1 7 7 jdep 6.5 off
 je1 7 2 0 jenh 20 off
 je2 7 3 0 jenh 20 off
 je3 7 4 0 jenh 20 off
 je4 7 5 0 jenh 20 off
 je5 7 6 0 jenh 20 off
 c1 2 0 20f
 c2 3 0 20f
 c3 4 0 20f
 c4 5 0 20f
 c5 6 0 20f
 .ends
 *
 *
 .subckt nor 4 3 2 1
 jd1 1 4 4 jdep 6.5 off
 je1 4 2 0 jenh 20 off
 je2 4 3 0 jenh 20 off
 c1 2 0 20f
 c2 3 0 20f
 .ends
 *
 .subckt nor3 5 4 3 2 1
 jd1 1 5 5 jdep 6.5 off
 je1 5 2 0 jenh 20 off
 je2 5 3 0 jenh 20 off
 je3 5 4 0 jenh 20 off
 c1 2 0 20f
 c2 3 0 20f
 c3 4 0 20f
 .ends
*
je1 1 2 3 jdep 15 off
jdi 4 3 4 jdep 20 off
jd1 4 5 5 jdep 55 off
c1 2 0 30f
*
*
* this is inverting latch
* note that like a normal clocked latch,
* the input should change
* on a different clock edge from
* the gate signal. otherwise
* output can be ambiguous.
jdpass 2 3 4 jdep 20 off
*
jd1 1 5 5 jdep 13.5 off
je1 5 4 0 jenh 40 off
*jd2 1 6 6 jdep 6.5 off
*je2 6 5 0 jenh 20 off
jd3 1 7 7 jdep 6.5 off
je3 7 5 0 jenh 20 off
*
jdres 7 0 4 jdep 10 off
jdiode 4 7 4 jdep 5 off
c1 3 0 20f
c2 5 0 35f
*c3 6 0 20f
c4 7 0 20f
c5 2 0 20f
c6 4 0 35f
*
 *
 .subckt prbs1 2 3 4 5 6 7 8 9 10 11 101
+ 102 1000 2000 1
 *
* 100 - 140 are the 5 prsets
 * preset word = 10000, note inversion at inputs
 xreg1 2 3 12 0 101 102 1000 2000 1 ct1reg
 xreg2 4 5 2 1 101 102 1000 2000 1 ct1reg
 xreg3 6 7 4 1 101 102 1000 2000 1 ct1reg
 xreg4 8 9 6 1 101 102 1000 2000 1 ct1reg
 xreg5 10 11 8 1 101 102 1000 2000 1 ct1reg
 *
 xxnor 12 4 11 5 10 1 xnor
 *
 .ends
 *
*
*
*.subckt stop
*
* 2-11 prbs reg outputs and compls.
* 12 = shift
* 13 = load
* 14 = phi2
* 15 = phi1
```

```
xprbs 2 3 4 5 6 7 8 9 10 11 12 13 14 15 1 prbs1
*
* this detects the endpoint 11110
* 16 = ep
xendnor 16 3 5 7 9 0 1 endnor
*
*
* 18 is a fake lsb signal
* 13 is level shifted lsb used to load prbs1
xinv1 17 18 1 inv1
xls1 13 17 1 100 ls
*
* 12 = shift prbs regs
xinv2 19 18 1 inv
xinv3 20 19 1 inv1
xls2 12 20 1 100 ls
*
* delay to generate signals for
* loading parallel output latches
xlat1 1 16 14 21 22 23 15 latch
xlat2 1 22 15 24 25 26 14 latch1
*
* 27 = del(ep)
* 29 = load output latch
xinv4 27 24 1 inv1
xls3 29 27 1 100 ls
*
* 31 = fake data input
* 210 = real serial data stream
xinv5 210 31 1 inv1
*
* the following are the serial regs. for the
* input data stream
*
* 210 = serial data
* note that 208 is lsb when loading into outlatches
xreg100 1 210 14 209 201 219 15 latch
xreg101 202 201 1 15 14 delay
xreg102 203 202 1 15 14 delay
xreg103 204 203 1 15 14 delay
xreg104 205 204 1 15 14 delay
xreg105 206 205 1 15 14 delay
xreg106 207 206 1 15 14 delay
xreg107 208 207 1 15 14 delay
*
*
* the following are the output latches
* they are loaded whenever the endpoint
* is detected and therefore
* synchronous with phi1
*
* 211 is msb of parallel word
* 218 is lsb ................
* 29 = loading signal from above
xoutlat1 211 201 29 1 outlat
xoutlat2 212 202 29 1 outlat
xoutlat3 213 203 29 1 outlat
xoutlat4 214 204 29 1 outlat
xoutlat5 215 205 29 1 outlat
xoutlat6 216 206 29 1 outlat
xoutlat7 217 207 29 1 outlat
xoutlat8 218 208 29 1 outlat
*
*.ends stop
*
vdd 1 0 1.7
vss 100 0 -0.6
*
*
* artificial clocks
```

```
vphi1 15 0 pulse(-0.5 0.5 20ps 200ps 200ps 0.6n 2n)
vphi2 14 0 pulse(-0.5 0.5 1020ps 200ps 200ps 0.6n 2n)
*
* data = ..00110011
* lsb is used directly to load the prbs1, so
* it changes with phi2
vlsb' 18 0 pulse(0.75 0 1020p 200p 200p 1.8n)
vdata' 31 0 pulse(0.75 0 20p 200p 200p 3.8n 8n)
*
```

Appendix 8

PRBS counter SPICE input file

```
p/s and s/p converters cascaded(p in p out)
 *
 .options limpts=1000000 itl5=0
 .width out=75
 *
 *       at 714mhz clock rate
 *
 .subckt latch 1 2 3 4 5 6 7
 jdpass1 2 3 4 jdep 20 off
 *
 * note that the phi2 inverter is 3 times
 * larger than the
 * phi1 inverter to make sure that in case
 * both pass transistors
 * are slightly on at the same time, the input
 * data always wins
 je1 5 4 0 jenh 40 off
 jd1 1 5 5 jdep 13.5 off
 je2 6 5 0 jenh 20 off
 jd2 1 6 6 jdep 6.5 off
 *
 jdpass2 6 7 4 jdep 20 off
 c1 2 0 25f
 c2 4 0 25f
 c3 5 0 25f
 c4 6 0 25f
 c5 3 0 25f
 c6 7 0 25f
 .ends
 *
 .subckt latch1 1 2 3 4 5 6 7
 jdpass1 2 3 4 jdep 40  off
 *
 * note that the phi2 inverter is 3 times larger than the
 * phi1 inverter to make sure that in case both pass transisto
 * are slightly on at the same time, the input data always win
 je1 5 4 0 jenh 40 off
 jd1 1 5 5 jdep 13.5 off
 je2 6 5 0 jenh 20 off
 jd2 1 6 6 jdep 6.5 off
 *
 jdpass2 6 7 4 jdep 40 off
 c1 2 0 25f
 c2 4 0 25f
 c3 5 0 25f
 c4 6 0 25f
 c5 3 0 25f
 c6 7 0 25f
 .ends
*
 .subckt delay 9 2 1 3 7
 xlatch1 1 2 3 4 5 6 7 latch
 xlatch2 1 5 7 8 9 10 3 latch
 .ends
 *
 *
 *
 .subckt ct1reg 13 11 2 101 100 102 3 7 1
```

```
* 13 = del(data)
* 11 = del(data)'
 xlat1 1 2 3 4 5 6 7 latch
* the 2 pass transistors are for loading or shifting
* an inverter is used to "restore" the signal
* 100 = load'
* 102 = load
* 101 = preset value
 jdp1 5 100 8 jdep 20
xinv1 9 8 1 inv1
jdp2 101 102 8 jdep 20
* second inverter to give correct polarity
 xlat2 1 9 7 10 11 12 3 latch
xinv2 13 11 1 inv
 c1 4 0 25f
 c2 100 0 25f
 c3 9 0 25f
 c4 5 0 15f
 c5 102 0 20f
 .ends
*
 *
 .subckt inv 3 2 1
 je1 3 2 0 jenh 20 off
 jd1 1 3 3 jdep 6.5 off
 c1 2 0 25f
 .ends
*
 .subckt inv1 3 2 1
 je1 3 2 0 jenh 40 off
 jd1 1 3 3 jdep 13.5 off
 c1 2 0 25f
 .ends
*
 .subckt xnor 2 3 4 5 6 1
 xnor1 7 3 4 1 nor
 xnor2 8 5 6 1 nor
 xnor3 2 7 8 1 nor
 .ends
 *
 *
 .subckt endnor 7 2 3 4 5 6 1
 jd1 1 7 7 jdep 6.5 off
 je1 7 2 0 jenh 20 off
 je2 7 3 0 jenh 20 off
 je3 7 4 0 jenh 20 off
 je4 7 5 0 jenh 20 off
 je5 7 6 0 jenh 20 off
 c1 2 0 25f
 c2 3 0 25f
 c3 4 0 25f
 c4 5 0 25f
 c5 6 0 25f
 .ends
 *
 *
 .subckt nor 4 3 2 1
 jd1 1 4 4 jdep 6.5 off
 je1 4 2 0 jenh 20 off
 je2 4 3 0 jenh 20 off
 c1 2 0 25f
 c2 3 0 25f
 .ends
 *
 .subckt nor3 5 4 3 2 1
 jd1 1 5 5 jdep 6.5 off
 je1 5 2 0 jenh 20 off
 je2 5 3 0 jenh 20 off
 je3 5 4 0 jenh 20 off
 c1 2 0 25f
 c2 3 0 25f
 c3 4 0 25f
 .ends
 *
je1 1 2 3 jdep 15 off
jdi 4 3 4 jdep 20 off
jd1 4 5 5 jdep 55 off
c1 2 0 8f
*
 *
 .subckt prbs1 2 3 4 5 6 7 8 9 10 11 101 102 1000 2000 1
 *
 * preset word = 01101, note inversion at inputs
* the word is read in the direction of shift, i.e.
* 0>1>1>0>1
 xreg1 2 3 12 1 101 102 1000 2000 1 ct1reg
 xreg2 4 5 2 0 101 102 1000 2000 1 ct1reg
 xreg3 6 7 4 0 101 102 1000 2000 1 ct1reg
 xreg4 8 9 6 1 101 102 1000 2000 1 ct1reg
 xreg5 10 11 8 0 101 102 1000 2000 1 ct1reg
 *
 xxnor 12 4 11 5 10 1 xnor
 *
 .ends
*
*
*
* 2-11 prbs reg outputs and compls.
* 12 = shift
* 13 = load
* 14 = phi2
* 15 = phi1
xprbs 2 3 4 5 6 7 8 9 10 11 12 13 14 15 1 prbs1
*
* this detects the endpoint 11110
* 16 = ep
xendnor 16 3 5 7 9 0 1 endnor
*
*
* 13 is level shifted lsb used to load prbs1
xinv1 17 22 1 inv1
xls1 13 17 1 100 ls
*
* 12 = shift prbs regs
xinv3 20 21 1 inv1
xls2 12 20 1 100 ls
*
* delay to generate signals for loading parallel output latche
xlat1 1 16 14 21 22 23 15 latch1
*
vdd 1 0 1.7
vss 100 0 -0.6
*
*
* artificial clocks
vphi1 15 0 pulse(-0.5 0.5 20ps 200ps 200ps 0.3n 1.4n)
vphi2 14 0 pulse(-0.5 0.5 720ps 200ps 200ps 0.3n 1.4n)
```

Appendix 9

GaAs-to-ECL output buffer SPICE input file

```
the ecl buffer (driven by a gaas buffer)
* simulated at 100mhz (200mb/s)
* also just works at 500b/s
*
* 6pf load and a diode to limit swing.
* note that termination is ecl -2v (or
* gaas -0.3v if vdd=gnd)
*
jd1 1 3 3 jdep 13.5
je1 3 2 0 jenh 40
c1 2 0 20f
*
jd1 1 3 3 jdep 26.667
je1 3 2 0 jenh 80
c1 2 0 20f
*
vdd 1 0 1.7
vin 2 0 pulse(0.1 0.7 0n 0.2n 0.2n 4.8n 10n)
*
*********************************
* a version of gaas output buffer (with 50 ohm
* termination for driving
* the ecl buffer
* (normal buffer also fine)
* in emergency, this buffer can just be used as
* an ecl buffer
*
xinv1 3 2 1 inv1
xinv2 4 3 1 inv2
*
je1 1 4 5 jenh 200
r1 5 0 50
c1 5 0 3p
*
************************************************
*
* the real ecl buffer, complete with input protection diode
*
jdiod 0 5 0 jdep 100
*
vss 11 0 -0.3
*
xinv3 6 5 1 inv2
xinv4 7 6 1 inv1
*
jd1 1 7 8 jdep 58
je2 8 6 0 jenh 70
*
jeout 1 8 9 jenh 300
jdiode 0 9 0 jdep 100
vmet 9 10 0
r2 10 11 50
c2 10 11 6p
*
*************************************************
```

Appendix 10

GaAs output buffer SPICE input file

```
check the output pad buffer (non-inverting)
*
jd1 1 3 3 jdep 40
je1 3 2 0 jenh 120
c1 2 0 25f
*
jd2 1 2 2 jdep 26.5
je2 2 100 0 jenh 80
*
jd3 1 4 4 jdep 26.5
je3 4 3 0 jenh 80
c2 3 0 35f
*
jepd 5 3 0 jenh 260
jepu 1 4 5 jenh 100
jdpu 1 4 5 jdep 25
c3 4 0 40f
c4 5 0 3p
jdi1 0 5 0 jdep 100
*
vdd 1 0 1.7
vin 100 0 pulse(0.7 0.1 0 200p 200p 800p 2n)
```

Appendix 11

3-level circuit SPICE input file

```
investigate the lsb/data mixer
* this is th improved version of it. the main
* problem lied in the
* level shifters because of the conflicting
* requirements for high and low
* shifts. one could use the new level shifter as
* in level4 stuff but
* because it is inverting it plces a high
* penalty on propagatiuon delay
* thru. the output devices.
*
 .options limpts=1000000 itl5=0
 .width out=75
 *
 *
 .subckt inv 3 2 1
 jd1 1 3 3 jdep 13.5
 je1 3 2 0 jenh 40 off
 c1 2 0 25f
 r1 2 0 1g
 r2 3 0 1g
 .ends
 *
 .subckt inv1 3 2 1
 jd1 1 3 3 jdep 8.333
 je1 3 2 0 jenh 25 off
 c1 2 0 25f
 r1 2 0 1g
 r2 3 0 1g
 .ends
 *
 .subckt inv2 3 2 1
 jd1 1 3 3 jdep 6.6667
 je1 3 2 0 jenh 20 off
 c1 2 0 25f
 r1 2 0 1g
 r2 3 0 1g
 .ends
 *
 .subckt nor 4 3 2 1
 jd1 1 4 4 jdep 13.5
 je1 4 3 0 jenh 40        off
 je2 4 2 0 jenh 40        off
 c1 2 0 25f
 c2 3 0 25f
 r1 2 0 1g
 r2 3 0 1g
 r3 4 0 1g
 .ends
 *
 .subckt nor1 4 3 2 1
 jd1 1 4 4 jdep 27
 je1 4 3 0 jenh 80        off
 je2 4 2 0 jenh 80        off
 c1 2 0 25f
 c2 3 0 25f
 r1 2 0 1g
 r2 3 0 1g
 r3 4 0 1g
 .ends
 *
 .subckt nor2 4 3 2 1
 jd1 1 4 4 jdep 6.667
 je1 4 3 0 jenh 20        off
 je2 4 2 0 jenh 20        off
 c1 2 0 25f
 c2 3 0 25f
 r1 2 0 1g
 r2 3 0 1g
 r3 4 0 1g
 .ends
 *
 *
* as in ls1, this is compromise between 2
* conflicting requirements for
* high nad low states, only much more severe.
* therefore the shifted low
* is not very good at all and the pull-down devices
* would have to pull against
* a slightly on pull-up device
 je1 1 2 4 jdep 30
* jd1 4 3 4 jdep 80       off
 jd2 4 5 5 jdep 80
 c1 2 0 25f
 r1 2 0 1g
 r2 4 0 1g
 .ends
 *
* a ratio fo 1/1.5 is used. though not ideal,
* its is compromise between
* a fast rise and shifted logic low closer to -0.5v
 je1 1 2 4 jenh 40
* jd1 4 3 4 jdep 80
 jd2 4 5 5 jdep 60
 c1 2 0 25f
 r1 2 0 1g
 r2 4 0 1g
 .ends
*
 *
 xls1 10 101 1 2 ls
xls2 9 102 1 2 ls1
xls3 7 3 1 2 ls1
 je1 1 10 11 jenh 150 off
* jd1 1 10 11 jdep 25     off
 *
 c1 10 0 16f
 c2 9 0 16f
 c3 7 0 16f
 c4 3 0 16f
 c5 101 0 16f
 c6 102 0 16f
 c7 3 0 16f
 *
 je3 11 9 0 jenh 150 off
 je2 11 7 2 jenh 180 off
 *
 jdin 0 11 0 jdep 100
* cout 11 0 5p
*
 .subckt depinv 3 2 1
 jd1 1 3 3 jdep 6
 jd2 3 2 0 jdep 12        off
 r1 2 0 1g
 r2 3 0 1g
 .ends
 *
 .subckt or 3 2 1
```

```
je1 1 2 3 jenh 15
jd1 3 0 0 jdep 15        off
c1 2 0 25f
r1 2 0 1g
r2 3 0 1g
.ends
*
.subckt latch 5 2 3 7 1
jd1 2 3 4 jdep 20 off
xinv1 5 4 1 inv
xinv2 6 5 1 inv2
jd2 6 7 4 jdep 20 off
c1 2 0 25f
c2 6 0 25f
r1 2 0 1g
r2 3 0 1g
r3 4 0 1g
r4 5 0 1g
r5 6 0 1g
r6 7 0 1g
.ends
*
.subckt delay 6 3 5 4 1
xlat1 2 3 4 5 1 latch
xlat2 6 2 5 4 1 latch
.ends
*
vdd 1 0 1.7
vss 2 0 -0.6
vphi2 100 0 pulse(-0.5 0.5 1.1n 0.2n 0.2n 0.6n 2n)
vphi1 200 0 pulse(-0.5 0.5 0.1n 0.2n 0.2n 0.6n 2n)
vlsb 3 0 pulse(0.1 0.7 0.2n 0.2n 0.2n 1.8n 6n)
vin 4 0 pulse(0.1 0.7 0.2n 0.2n 0.2n 1.8n 4n)
*
* lsb'
xlat1 5 3 100 200 1 latch
*
.nodeset v(5)=0.1 v(6)=0.7 v(15)=1.7
* 2 lsb's, one full one clamped
xinv2 6 5 1 inv
xinv3 15 5 1 inv1
*
* 15 is full and used to drive enh pd.
*
* data'
xlat2 8 4 100 200 1 latch
*
* data
xinv5 13 8 1 inv
*
* lsb nor data' to drive enh pd to ground
xnor1 9 6 8 1 nor1
*
.nodeset v(8)=0.1 v(13)=0.7 v(9)=0.1 v(10)=0.1
* lsb nor data to drive pu.
xnor2 10 6 13 1 nor
*
* the new output driver subcircuit !
*
xout 11 10 9 15 1 2 1000 9000 1500 outbuf
* dep. inverter to detect lsb
xdinv1 14 11 1 depinv
*
.nodeset v(11)=0.1 v(23)=1.7 v(14)=0.1
* invverter to detect data
xinv6 23 11 1 inv
*
* for lsb. 17 is lsb latched at phi1
xlat3 16 14 200 100 1 latch
xinv4 17 16 1 inv2
```

```
 xlat4 25 16 100 200 1 latch
 *
 .nodeset v(16)=0.1 v(17)=0.7 v(25)=0.1
 *
 * latch for data’. 18 is data, but not msb restored
 *
 * the next bit restores the msb.
 * stage
 *
 xlat5 18 23 200 100 1 latch
 xdel1 19 18 200 100 1 delay
 *
 xnor3 20 18 17 1 nor2
 xnor4 21 19 16 1 nor2
 xnor5 22 21 20 1 nor2
 *
 xlat6 24 22 100 200 1 latch
 .nodeset v(18)=0.1 v(19)=0.1 v(20)=0.1 v(21)=0.7
+ v(22)=0.1 v(24)=0.1
 r1 16 0 1g
 r2 18 0 1g
 r3 19 0 1g
 r4 11 0 1g
 r5 24 0 1g
 r6 25 0 1g
 *
 *
 *
```

Appendix 12

4-level circuit SPICE input file

```
check out the levels
 je1 1 2 4 jdep 18 off
 je2 1 2 4 jenh 40 off
* jd1 4 3 4 jdep 80        off
 jd2 4 5 5 jdep 50 off
 c1 2 0 20f
 .ends
 je1 1 2 4 jenh 55 off
* jd1 4 3 4 jdep 80        off
 jd2 4 5 5 jdep 40       off
 c1 2 0 20f
 .ends
 .subckt invdou 5 2 1 6
jd1 1 3 3 jdep 90 off
je1 3 2 0 jenh 90 off
*
jdi1 4 3 4 jdep 10 off
jd2 1 5 5 jdep 50 off
je2 5 4 6 jenh 85 off
jd3 4 6 6 jdep 40 off
*
 je1 1 2 3 jdep 55 off
 jd1 4 3 4 jdep 80        off
 jd2 4 5 5 jdep 40       off
 c1 2 0 20f
 .ends
*
 *
* xls1 10 101 1 2 ls
xivnd1 10 101 1 2 invdou
xls2 9 102 1 2 invdou
xls3 7 3 1 2 invdou
* je1 1 10 11 jenh 100 off
 jd1 1 10 11 jdep 200 off
 *
* r1 1 10 10g
 c1 10 0 30f
 c2 9 0 30f
 c3 7 0 30f
 c4 3 0 30f
 *
 je3 110 9 0 jenh 600 off
 je2 110 7 2 jenh 600 off
 *
jdi1 12 110 12 jdep 50 off
jdi2 0 12 0 jdep 50 off
 cout 110 0 5p
vdd 1 0 1.7
vss 100 0 -0.6
*vlow 200 0 0.1
vin1 6 0 pulse(0.1 0.75 1020p 0 0 1n 4n)
vin2 10 0 pulse(0.1 0.75 3020p 0 0 1n 2n)
vin3 5 0 pulse(0.1 0.75 20p 0 0 1n 2n)
*
xout 2 20 6 10 5 1 100 4 8 9 outbuf
vcur 2 20 0
```

Appendix 13

4-level detection circuit SPICE input file

```
check out the 14 detection scheme
*
*
jd1 1 3 3 jdep 10
je1 3 2 0 jenh 30
*
xinv 3 2 1 inv
jd1 1 2 2 jdep 10
jdi1 4 2 4 jdep 20
*
je1 1 2 3 jenh 15
jd1 3 0 0 jdep 30
*
jd1 1 2 3 jdep 10
jd2 3 0 0 jdep 10
xinv 4 3 1 inv
*
vdd 1 0 1.7
vinplus 2 0 pulse (0.75 1.5 20p 200p 200p 0.8n 2n)
vinminus 7 0 pulse (-0.5 0.1 20p 200p 200p 0.8n 2n)
* for 0.75 and 1.5 levels
xdet1 3 2 1 4 det1
xdet2 5 2 1 det2
xdet3 6 2 1 70 det3
* for 0.1 and -0.5 levels
xdet4 8 7 1 9 det1
xdet5 10 7 1 det2
xdet6 11 7 1 12 det3
* in practice only one set of 3 is used
* this is just to show all possible combinations
```

Appendix 14

Single phase latches SPICE input file

```
check out a version of single phase latch
*
***********************************************
* two single phased latches in pie and mule
* (1) pie is active on hi of phi
* (2) mule is active on lo of phi
* (3) mule is fully static and uses 4 nor gates
*     the input nor gates havea dfet
*     pull-down for phi
* (4) pie is like the pseudo-dynamic latches
*     with a permanent feedback
*     therefore spikes and race at node v(9) which
*     drives a pie latch
*
jd1 1 3 3 jdep 13.5
je1 3 2 0 jenh 40
c1 2 0 35f
*
jd1 1 3 3 jdep 3.5
je1 3 2 0 jenh 10
c1 2 0 35f
*
jd1 1 4 4 jdep 7
je1 4 3 0 jenh 20
jd2 4 2 0 jdep 14
c1 2 0 35f
c2 3 0 35f
*
jd1 1 4 4 jdep 13.5
je1 4 3 0 jenh 40
jd2 4 2 0 jenh 40
c1 2 0 35f
c2 3 0 35f
*
jdp1 2 3 4 jdep 20
xinv1 5 4 1 inv1
xinv2 6 5 1 inv2
jdio1 4 6 4 jdep 5
jdp2 6 0 4 jdep 5
c1 3 0 35f
c2 6 0 35f
*
xnor1 5 2 4 1 nor1
xnor2 6 3 4 1 nor1
xnor3 7 5 8 1 nor2
xnor4 8 7 6 1 nor2
*
vdd 1 0 1.7
vin 2 0 pulse (0.1 0.75 20p 200p 200p 0.9n 2n)
*vinbar 5 0 pulse (0.75 0.1 20p 200p 200p 1.8n 4n)
vphi 100 0 pulse (-0.5 0.5 520p 200p 200p 0.3n 1n)
*
xpie1 5 4 2 100 3 1 pie
xinv1 6 5 1 inv1
xmu 9 10 6 5 100 7 8 1 mule
xpie2 11 12 9 100 13 1 pie
```

Appendix 15

GaAs circuit test results

This Appendix describes in more detail the test results available from the 3 MODFET mask sets fabricated so far. It has been stated that the circuits were modified for a MODFET process, but were eventually fabricated in a GaAs MESFET process owing to processing difficulties. The main difference between the two was found to be:

V_{on} of MODFET = 1.1V

V_{on} of fabricated MESFET = 0.8V

The V_{on} of the fabricated MESFET process is higher than a standard MESFET (0.75V), but substantially lower than that of a MODFET. The V_t's of the fabricated MESFETs were similar to that of the designs. Owing to this substantial difference in V_{on}, all level shifting stages (within the clock generator, and the PRBS counter) would have **less** voltage shifting than intended. For instance, if two diode drops were used, then the actual shifting voltage would be 0.6V less than with MODFET. In particular, this would affect the clock generator and level shifting in the PRBS counter, and the 3-level generation circuits, due to the use of negative voltages. However, one of the safety features built into the level shifting stages was externally adjustable V_{ss} and gate voltages of the pull-down loads. Adjustments are likely to be necessary to obtain good voltage swings and in some cases, correct functionality. Nevertheless, the 300mV difference in V_{on} may prove to be too large to compensate for some level shifting stages, especially those in the PRBS counter, which uses the normal source follower type stage.

Although level shifting is likely to be affected, all DCFL-based circuits should operate functionally. A large noise margin had been built into the MODFET-modified chips by increasing the β_R ratio to 6, rather than the normal value of 3 for MESFETs. This was due to the different β values of the D-MODFET and D-MESFET devices. Unfortunately, this also means that the circuits would be expected to operate slower in the MESFET process.

Test results

The test results available so far are:

(i) Two-phase clock generator — fully functional, shown in Fig. A15.1

(ii) 3-level generation circuit — fully functional, as shown in Figs.A15.2 & A15.3. In Fig. A15.3, clock feedthrough is clearly visible. This was due to non-optimum supply voltages used for this particular example.

(iii) 3-level detection circuit — LSB detection fully functional, shown in Fig. A15.4. Note the half clock cycle shift between input and output due to pipelining with a pseudo-dynamic latch.

A layout fault was found in the output buffer of the 3-level detection circuit at the DATA signal output. Although the DATA signal was likely to have been decoded correctly internally, an inverter was grounded, rendering the output buffer ineffective. A SEM photograph should reveal the actual signal if the chips were available.

The oscilloscope photographs were taken at low clock rates on probe cards to initially check for functionality, as are evident from the scope phtos. Future high speed tests in Honeywell will be on bonded chips. Although high speed tests are not yet available, these test results have successfully demonstrated the operating concepts of the ±ve clock generator, the pseudo-dynamic latches, and the 3-level generation and detection circuits.

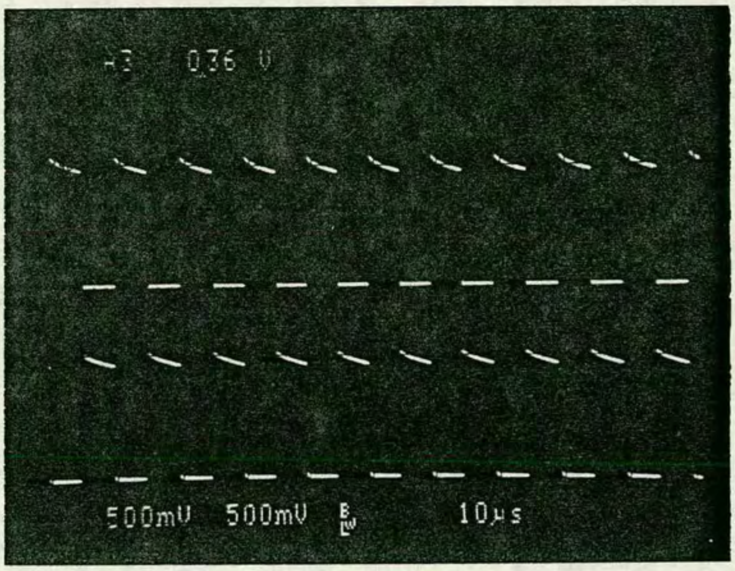


Fig. A15.1 Two-phase non-overlapping clock signals.

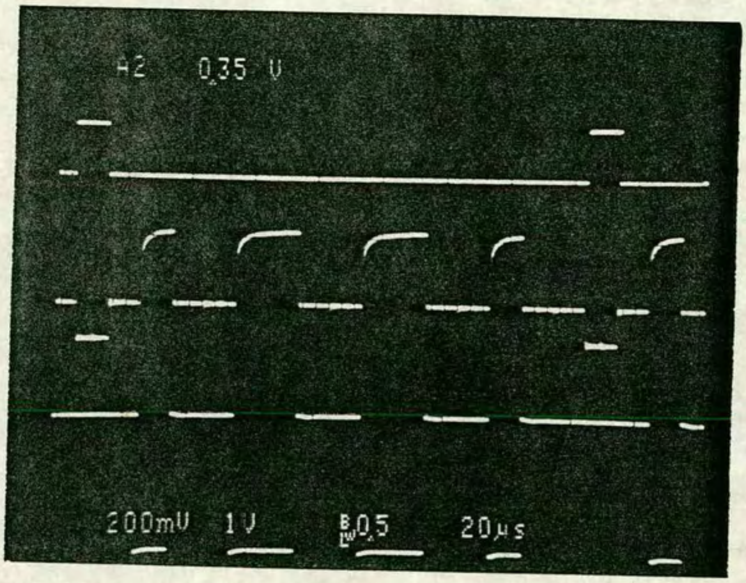


Fig. A15.2 3-level generation circuit: LSB signal (top),
3-level output signal (middle), DATA input signal (bottom).

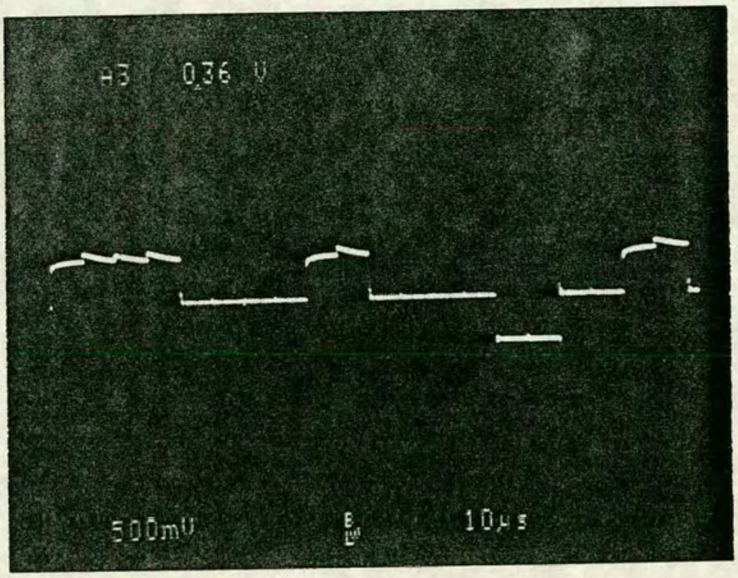


Fig. A15.3 Different example of 3-level output signal
(clock feedthrough due to incorrect power supply voltages).

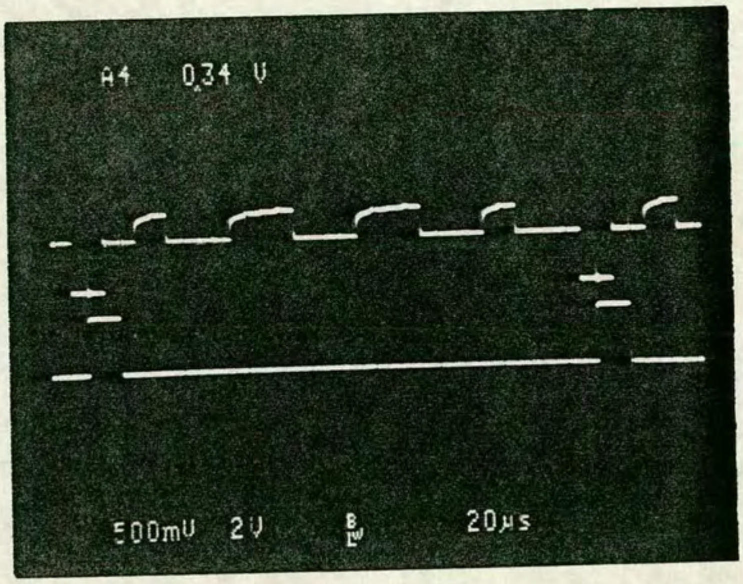


Fig. A15.4 3-level detection circuit: 3-level input signal (top), detected LSB signal (bottom).

Bibliography

[1] I. M. Abdel-Motaleb, W.C. Rutherford, L. Young, "GaAs Inverted Common Drain Logic (ICDL) and Its Performance Compared with Other GaAs Logic Families", Solid-State Electronics, Vol. 30, No. 4, 1987.

[2] A. Akinwande et. al., "A High Performance (AlGa)As/GaAs MODFET Butterfly Adder Chip For FFT Computation", IEEE GaAs IC Symposium Digest 1989, pp. 53-56.

[3] *Am7968/Am7969 Data Sheets*, Advanced Micro Devices, January 1986.

[4] *Data-Acquisition Handbook*, Analog Devices, Vol. 1: Integrated Circuits, 1984.

[5] C. J. Anderson et. al., "A GaAs MESFET 16×16 Crosspoint Switch at 1700 Mbit/s", IEEE GaAs IC Symposium Digest 1988, pp. 91-94.

[6] P. Antognetti, G. Massobrio, *Semiconductor Device Modeling with SPICE*, McGraw Hill Book Company, 1988, pp. 134.

[7] "ASTAP User's Manual", IBM Computers, 1983.

[8] J. Babbitt et. al., "GaAs Sub-Band Tuner", IEEE GaAs IC Symposium Digest 1989, pp. 45-48.

[9] M. A. Bayoumi, "Reconfigurable Testable Bit-Serial Multiplier for DSP Applications", IEE Proceedings, Vol. 136, Part E, No. 6, November 1989.

[10] S. W. Bland, D. Wood, J. Mun, "Self-alignment techniques for GaAs MESFET ICs", Journal of the Institute of Electronic and Radio Engineers, Vol. 57, No. 1.

[11] P. Bonjour et. al., "Saturation Mechanisms in 1μm gate GaAs FET with Substrate Interface Barrier", IEEE Transactions on Electron Devices, Vol. ED-27, 1980.

[12] D. Bursky, "Large, Speedy Arrays Accelerate Systems", Electronic Design International, December 1988, pp. 37-43.

[13] F. Capasso, *Gallium Arsenide Technology*, Chapter 8, pp.303-328, Howard W. Sams & Co., 1985.

[14] C. T. Chuang et. al., "A Subnanosecond 5kbit Bipolar ECL RAM", IEEE Journal of Solid-State Circuits, Vol. 23, No. 5, October 1988.

[15] C. L. Chen et. al., "Reduction of Sidegating in GaAs Analog and Digital Circuits Using a New Buffer Layer", IEEE Transactions on Electron Devices, Vol. 36, No. 9, September 1989.

[16] J. Chen et. al., "A 1.2ns GaAs 4K Read Only Memory", IEEE GaAs Symposium Digest 1988, pp. 83-86.

[17] H. K. Chung et. al., "High Speed and Ultra Low Power GaAs MESFET 5×5 Multipliers", IEEE GaAs Symposium Digest 1986, pp. 15-18.

[18] B. C. Cole, "GaAs LSI Goes Commercial", Electronics, September 1986, pp. 57-60.

[19] B. C. Cole, "These Chips Move Data at 1 Gbit/s", Electronics, December 1989, pp. 71.

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

[21] P. Denyer, D. Renshaw, *VLSI Signal Processing: A Bit-serial Approach*, Addison-Wesley Publishing Company, 1985.

[22] A. Dubois et. al., "GaAs IC's for High Speed Signal Processing", IEEE Journal of Solid-State Circuits, Vol. SC-23, No. 3, June 1988.

[23] E. Delhaye, C. Rocher, J. C. Baelde, J. M. Gibereau, M. Rocchi, "A 2.5ns 40mW 4×4 GaAs Multiplier in Two's Complement Mode", IEEE Journal of Solid-State Circuits, Vol. SC-22, No. 3, June 1987.

[24] R. C. Eden, "The Development of the First LSI GaAs Integrated Circuits and the Path to the Commercial Market", Proceedings of the IEEE, July 1988, Vol. 76 No.7, pp. 756-777.

[25] R. C. Eden, "Capacitor Diode FET Logic (CDFL) Circuit Approach For GaAs D-MESFET ICs", IEEE GaAs IC Symposium Digest, 1984.

[26] C. G. Ekroot, S. I. Long, "A GaAs 4-bit Adder-Accumulator Circuit for Direct Digital Synthesis", IEEE Journal of Solid-State Circuits, SC-23, No. 2, April 1988.

[27] Electronics, January 1989.

[28] "Microwave Semiconductor and Triquint Share GaAs Chip Process", Electronics, 31 March 1988.

[29] Electronics, March 1989.

[30] R. Eppenga, M. F. H. Schuurmans, "Theory of the GaAs/AlGaAs Quantum Well", Philips Technical Review, Vol. 44, No. 5, November 1988.

[31] A. Fiedler, J. Chung, D. Kang, "A 3ns/Kx4K Static Self-Timed GaAs RAM", IEEE GaAs IC Symposium Digest 1988, pp. 67-70.

[32] G. G. Fountain et. al., "GaAs MIS Structures with SiO_2 Using a Thin Silicon Interlayer", Electronics Letters, 1 September 1988, Vol.24 No.18, pp. 1134-1135.

[33] Y. Fujisaki, N. Matsunaga, "The Origin of Gate Hysteresis and Gate Delay of MESFETs Appear Under Low Frequency Operation", IEEE GaAs IC Symposium Digest 1988, pp. 235-238.

[34] "GaAs IC's for Opto Telecommunications", Hitachi Data Sheets, August 1988.

[35] B. Gabillard, T. Ducourant, C. Rocher, M. Prost, J. Maluenda, "A 200mW GaAs 1K SRAM with 2ns Cycle Time", IEEE Journal of Solid-State Circuits, Vol. SC-22, No. 5, October 1987.

[36] A. F. Galashan, S. W. Bland, "Application of Pt/a-Si: H Gate GaAs FETs To Wide Noise Margin Direct-coupled FET Logic", Electronics Letters, 28 September 1989, Vol.25 No. 20, pp. 1344-1345.

[37] R. V. Gautier et. al., "A 150-MOPS GaAs 8-bit Slice Processor", IEEE Journal of Solid-State Circuits, Vol. SC-23, No. 5, October 1988.

[38] T. R. Gheewala, "System Level Comparison of High Speed Technologies", IEEE GaAs IC Symposium Digest 1984, pp. 245-250.

[39] B. K. Gilbert, G. W. Pan, "The Application of Gallium Arsenide Integrated Circuit Technology to the Design and Fabrication of Future Generation Digital Signal Processors: Promises and Problems", Proceedings of the IEEE, July 1988, pp. 816-834.

[40] K. Gonoi et. al., "A GaAs 8×8-bit Multiplier/Accumulator Using JFET DCFL", IEEE Journal of Solid-State Circuits, Vol. SC-21, No. 4, August 1986.

[41] R. Goyal, N. Scheinberg, "An Accurate GaAs Depletion MESFET Model Suitable for Digital and Analog GaAs LSI Circuit Design", Electronic Design Automation 1987, pp. 201-205.

[42] R. Goyal, "MESFET Model Improves Accuracy, Cuts CPU Time", Microwave &RF, April 1987, pp. 71-77.

[43] D. Haigh, J. Everard, Ed., *GaAs Technology and its Impact on Circuits and Systems*, Peter Peregrinus Ltd., 1989.

[44] H. Hamano et. al., "8 Gbit/s GaAs Logic ICs for Optical Fibre Communication Systems", Electronics Letters, 24 November 1988, Vol. 24, No. 24.

[45] D. Harrington et. al., "A GaAs 32-Bit RISC Microprocessor", IEEE GaAs IC Symposium Digest 1988, pp. 87-90.

[46] Y. Hatta et. al., "A GaAs IC Set for Full Integration of 2.4Gb/s Optical Transmission Systems", IEEE GaAs IC Symposium Digest 1988, pp. 15-18.

[47] S. Hayano et. al., "A GaAs 8×8 Matrix Switch LSI for High-Speed Digital Communications", IEEE GaAs IC Symposium Digest 1987, pp. 245-248.

[48] T. Hayashi et. al., "Novel Circuit Technology for ECL-Compatible GaAs Static RAMs with Small Access Time Scattering", IEEE Journal of Solid-State Circuits, SC-22, No. 5, October 1987.

[49] N. Hendrickson et. al., "A GaAs Bit-slice Microprocessor Chip Set", IEEE GaAs IC Symposium Digest 1987, pp. 197-200.

[50] H. Hirayama et. al., "4 Gb/s GaAs Gate Arrays with 0.5μm WSi-Gate MESFET Technology", IEEE GaAs IC Symposium Digest 1989, pp. 325-328.

[51] N. Homma et. al., "A 3.5ns 2W 20mm^2, 16kbit ECL Bipolar RAM", IEEE Journal of Solid-State Circuits, SC-21, No. 5, October 1986.

[52] H. Ichino et. al., "A 50ps 7K-Gate Masterslice Using Mixed Cells Consisting of an NTL Gate and an LCML Macrocell", IEEE Journal of Solid-State Circuits, SC-22, No. 2, April 1987.

[53] M. Ida, N. Kato, T. Takada, "A 4-Gbits/s GaAs 16:1 Multiplexer/1:16 Demultiplexer LSI Chip", IEEE Journal of Solid-State Circuits, Vol. 24, No. 4, August 1989.

[54] I^{2}S Bus Specification, Philips Electronic Components and Materials, 1988.

[55] M. Ino et. al., "30ps 7.5GHz GaAs MESFET Macrocell Array", IEEE Journal of Solid-State Circuits, Vol. 24, No. 5, October 1989.

[56] M. Ino et. al., "A 1.2ns GaAs 4kb Read Only-Memory Fabricated by 0.5μm Gate BP-SAINT", IEEE GaAs IC Symposium Digest 1987, pp. 189-192.

[57] K. Inokuchi et. al., "Suppression of Sidegating Effect for High Performance GaAs ICs", IEEE GaAs IC Symposium Digest 1987, pp. 117-120.

[58] K. Ishida et. al., "12 Gbps 2-bit Multiplexer/Demultipliexer Chip Set for the SONET STS-192 System", IEEE GaAs IC Symposium Digest 1989, pp. 317-320.

[59] W. R. Iversen, "It Looks Like an '89 Debut for the Cray-3 After All", Electronics, December 1989.

[60] W. R. Iversen, "Kieley's Challenge: Guiding a Startup in a New Computer Technology', Electronics, April 1989, pp. 135-136.

[61] V. Iyer, S. J. Joshi, "FDDI's 100Mbps protocol improves on 802.5 spec's 4Mbps limit", Electronics Design News, May 2 1985.

[62] L. B. Jackson, J. F. Kaiser, H. S. McDonald, "An Approach to the Implementation of Digital Filters", IEEEE Transactions Audio Electroacoustics, Vol. AU-16, September 1968.

[63] J. F. Jensen, L. G. Salmon, D. S. Deakin, M. J. Delaney, "26GHz GaAs Room-Temperature Dynamic Divider Circuit", IEEE GaAs IC Symposium Digest 1987, pp. 201-204.

[64] A. Kameyama et. al., "An 8-bit slice GaAs Bus Logic LSI for a High-Speed Parallel Processing System", IEEE GaAs IC Symposium Digest 1989, pp.105-108.

[65] K. Kaminishi, "GaAs on Si Technology", Solid State Technology, November 1987, pp. 91-97.

[66] M. G. Kane, P. Y. Chan, S. S. Cherensky, D. C. Fowlis, "A 1.5GHz Programmable Divide-by-N GaAs Counter", IEEE Journal of Solid-State Circuits, Vol. 23, No. 2, April 1988.

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

[68] D. Kiefer, J. Heightley, "Cray-3: A GaAs Implemented Supercomputer System", IEEE GaAs IC Symposium Digest 1987, pp. 3-6.

[69] C. G. Kirkpatrick, "Making GaAs Integrated Circuits", Proceedings of the IEEE, July 1988, pp. 792-815.

[70] R. Krein et. al., "A 4K×1 Bit Complementary E-JFET Static RAM", IEEE GaAs IC Symposium Digest 1987, pp. 185-188.

[71] P. H. Ladbrooke, *MMIC Design: GaAs FETs and HEMTs*, Artech House, 1989.

[72] L. E. Larson, "An Improved GaAs MESFET Equivalent Circuit Model for Analog Integrated Circuit Applications", IEEE Journal of Solid-State Circuits, Vol. SC-22, No. 4, August 1987.

[73] G. Lee, S. Canaga, B. Terell, I. Deyhimy, "A High Performance GaAs Gate Array Family", IEEE GaAs IC Symposium Digest, 1989, pp. 33-36.

[74] C. A. Liechti, R. Jolly, M. Namjoo, "GaAs Integrated Circuits for Error-Rate Measurement in High-Speed Digital Transmission Systems", IEEE Journal of Solid State Circuits, Vol. SC-18, No. 4, August 1983.

[75] A. W. Livingstone, A. D. Welbourn, G. L. Blau, "Manufacturing Tolerance of Capacitor Coupled GaAs FET Logic Circuits", IEEE Electron Device Letters, Vol. EDL-3, No. 10, October 1982.

[76] K. S. Lowe, "High-Speed GaAs 4×4 Bit-Parallel Multiplier Using Super Capacitor FET Logic", Electronics Letters, 9 April 1987, Vol. 23, No. 8.

[77] R. F. Lyon, "Two's Complement Pipeline Multipliers", IEEE Transactions Communications, Vol. COM-24, April 1976.

[78] R. F. Lyon, "A Bit-Serial VLSI Architectural Methodology for Signal Processing", VLSI-81, Academic Press 1981.

[79] H. Makino et. al., "A 7ns/850mW GaAs 4Kbit SRAM Fully Operative at 75C", IEEE GaAs IC Symposium Digest 1988, pp. 71-74.

[80] M. L. Macchia, B. Crawforth, B. Grung, "Flight GaAs Numerically Controlled Oscillator", IEEE GaAs IC Symposium Digest 1989, pp. 49-52.

[81] S. Matsue et. al., "A Soft Error Improved 7ns/2.1W GaAs 16kb SRAM", IEEE GaAs IC Symposium Digest 1989, pp. 41-44.

[82] N. Matsunaga et. al., "GaAs MESFET Technologies with 0.7μm Gate Length for 4kb 1ns Static RAM", IEEE GaAs IC Symposium Digest 1987, pp. 129-132.

[83] J. Mavor, M. A. Jack, P. B. Denyer, "Introduction to MOS LSI Design", Addison Wesley Publishing Company, 1983, pp. 18-61.

[84] E. J. McClusky, "Logic Design of Multivalued I^2L Logic Circuits", IEEE Transactions on Computer, Vol. C-28, No. 8, August 1979.

[85] M. S. McGregor, P. B. Denyer, A. F. Murray, "A Single-phase Clocking Scheme for CMOS VLSI", Proc. Conf. Advanced Research in VLSI, Cambridge MA, M.I.T. Press, 1987, pp. 257-271.

[86] W. V. McLevige, C. T. M. Chang, A. H. Tadikken, "An ECL-Compatible GaAs MESFET 1kbit Static RAM", IEEE Journal of Solid-State Circuits, SC-22, No. 2, April 1987.

[87] R. I. McTaggart, Honeywell Sensors and Signal Processing Laboratory, private communication.

[88] *MECL Device Data*, Motorola Semiconductors, 1986.

[89] P. J. T. Mellor, "GaAs Integrated Circuits for Telecommunications Systems", British Telecom Technology Journal, Vol. 5, No. 4, October 1987.

[90] J. Millman, *Microelectronics*, McGraw-Hill Book Company, 1987.

[91] V. Milutinovic, "GaAs Microprocessor Technology", IEEE Computer Magazine, 1986, pp.10-81.

[92] A. Mitonneau et. al., "Direct Experimental Comparison of Submicron GaAs and Si NMOS MSI Digital IC's", IEEE GaAs IC Symposium Digest 1984, pp. 3-6.

[93] O. J. Morales, B. K. Arnold, F. Collins, "Beyond Binary — An Introduction to Multilevel Arithmetic and Its Implementation", Electronic Engineering, October 1988, pp. 45-56.

[94] *Mullard technical handbook: ECL 100000 family*, Book 4, Part 10, 1985.

[95] L. W. Nagel, "SPICE2: A Computer Program to Simulate Semiconductor Circuits", Electronics Research Laboratory, Memorandum No. ERL-M520, 9 May 1975.

[96] Y. Nakayama et. al., "A GaAs 16×16 bit Parallel Multiplier", IEEE Journal of Solid State Circuits, Vol. SC-18, No. 5, October 1983.

[97] S. S. Nethisinghe, "Introduction to Bit Stream A/D, D/A Conversion", Philips Components, September 1988.

[98] "News and Events", IEEE Circuits and Devices Magazine, Vol. 6, No. 1, Jan 1990.

[99] S. Notomi et. al., "A High Speed 1K×4-Bit Static RAM Using 0.5μm-gate HEMT", IEEE GaAs IC Symposium Digest 1987, pp. 177-180.

[100] J. Notthoff et. al., "A 4K×1 Bit Complementary E-JFET Static RAM", IEEE GaAs IC Symposium Digest 1987, pp. 185-188.

[101] G. Nuzillat, G. Bert, F. Damay-Kavala, C. Arnodo, "High-Speed Low-Power Logic IC's Using Quasi-Normally-Off GaAs MESFET's", IEEE Journal of Solid State Circuits, Vol. SC-16, No. 3, June 1981.

[102] Y. Ogawa et. al., "A 32-bit GaAs Asynchronous Serial Data Transmitter and Receiver Chip Set for Parallel Data Processing", IEEE GaAs IC Symposium Digest 1989, pp. 329-332.

[103] Y. Oowaki et. al., "A Sub-10-ns 16×16 Multiplier Using 0.6μm CMOS Technology", IEEE Journal of Solid State Circuits, Vol. SC-22, No. 5, October 1987.

[104] A. V. Oppenheim, R. W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall International, 1989, pp. 290-402.

[105] A. Peczalski et. al., "A 6K GaAs Gate Array with Fully Functional LSI Personalization", IEEE Journal of Solid State Circuits, Vol. SC-23, No. 2, April 1988.

[106] W. M. Penney, L. Lau, Ed., *MOS Integrated Circuits*, Van Nostrand Reinhold Company, 1972, pp.59.

[107] E. H. Perea et. al., "A GaAs Low-Power Normally-On 4 Bit Ripple Carry Adder", IEEE Journal of Solid State Circuits, Vol. SC-18, No. 3, June 1983.

[108] V. Peterson, "Applications of GaAs ICs in Instruments", IEEE GaAs IC Symposium Digest 1988, pp. 191-194.

[109] N. R. Powell, J. M. Irwin, "Signal Processing with Bit-Serial Word-Parallel Architectures", SPIE Vol. 154, Real-Time Siganl Processing, 1978.

[110] A. Prabhakar, "Digital Gallium Arsenide Upgrades For Military Systems", IEEE GaAs IC Symposium Digest 1989, pp. 15-17.

[111] P. A. Ramamoorthy, B. Potu, T. Tran, "Bit-Serial VLSI Implementation of Vector Quantizer for Real-Time Image Coding", IEEE Transactions on Circuits and Systems, Vol. 36, No. 10, October 1989.

[112] H. M. Reekie, N. Petrie, J. Mavor, P. B. Denyer, C. H. Lau, "The Design and Implementation of Digital Wave Filters using Universal Adaptors", IEE CRSP Special Issue on The Design and Application of Digital Signal Processors. Proc IEE (F), Vol. 131, No. 6, pp 615-622 Oct. 1984.

[113] H. M. Rein, "Multi-Gigabit Per Second Silicon Bipolar IC's for Future Optical-Fiber Transmission Systems", IEEE Journal of Solid State Circuits, Vol. SC-23, No. 7, June 1988.

[114] D. Renshaw, C. H. Lau, "Race-free Clocking of CMOS Pipelines Using a Single Global Clock", IEEE Journal of Solid-State Circuits, Vol. 25, No. 3, June 1990.

[115] D. A. Rich, K. L. C. Naiff, K. G. Smalley, "A Four-State ROM Using Multilevel Process Technology", IEEE Journal of Solid State Circuits, Vol. SC-19, No. 2, April 1984.

[116] M. Rocchi, B. Gabillard, "GaAs Digital Dynamic IC's for Applications up to 10GHz", IEEE Journal of Solid-State Circuits, Vol. SC-18, No. 3, June 1983.

[117] R. N. Sato et. al., "Performance of Carry Lookahead Generator Circuit Fabricated with Gallium Arsenide", IEEE Journal of Solid-State Circuits, Vol. SC-22, No. 1, June 1987.

[118] G. Schou et. al., "Fully-ECL Compatible GaAs Standard Cell Library", IEEE Journal of Solid-State Circuits, Vol. SC-23, No. 3, June 1988.

[119] D. J. Schwab et. al., "Application of a 3000-Gate GaAs Gate Array in the Development of a Gigahertz Digital Test System", IEEE Journal of Solid-State Circuits, Vol. SC-18, No. 3, June 1983.

[120] Y. D. Shen et. al., "An Ultra High Performance Manufacturable GaAs E/D Process", IEEE GaAs IC Symposium Digest 1987, pp. 125-128.

[121] S. Shimizu, N. Koide, "Proposal of GaAs Stacked DCFL Circuit", Electronics Letters, 26 October 1989, Vol. 25, No. 22.

[122] P. J. Smith, P. Birdsall, C. T. Mallett, "GaAs Buffer Store Components for Gbit/s Optical Fiber Transmission Systems", IEEE Journal of Solid-State Circuits, Vol. SC-24, No. 3, June 1989.

[123] S. G. Smith, P. B. Denyer, *Serial-Data Computation*, Kluwer Academic Publishers, 1988, pp. 9-48 & pp. 165-188.

[124] S. G. Smith, "Serial/Parallel Automultiplier", Electronics Letters, Vol. 23, No. 8, 9 April 1987.

[125] P. H. Singer, "Material Requirements of GaAs ICs and Future III-V Devices", Semiconductor International, November 1986, pp. 68-71.

[126] P. H. Singer, "Ford Puts GaAs on Silicon", Semiconductor International, September 1988, pp. 17-18.

[127] R. Soares Ed., *GaAs MESFET Circuit Design*, Artech House, 1988.

[128] I. Son, T. W. Tang, "Modelling Deep-Level Trap Effects in GaAs MESFETs", IEEE Transactions on Electron Devices, Vol. 36, No. 4, April 1989.

[129] *Stereo CMOS DAC for Digital Audio Systems*, SAA7320 Development Data, Philips Components, January 1988.

[130] A. Stevenson et. al., "A 2.4 Gbit/s long-reach optical transmission system", Journal of British Telecom Technology, Vol. 7, No. 1, January 1989.

[131] F. G. Stremler, *Introduction to Communication Systems*, Addison-Wesley Publishing Company, 1982, pp. 651-666.

[132] M. Suzuki, M. Hirata, S. Konaka, "43ps 5.2GHz Macrocell Array LSI's", IEEE Journal of Solid-State Circuits, Vol. SC-23, No. 5, October 1988.

[133] S. M. Sze, *Physics of Semiconductor Devices*, 2nd Edition, John Wiley & Sons, 1981, pp. 246-248.

[134] K. Tan et. al., "A Submicron Self-Aligned Gate GaAs MESFET Technology for Low Power Subnanosecond Static RAM Fabrication", IEEE GaAs IC Symposium Digest 1987, pp. 121-124.

[135] S. Taylor, "A 100ps GaAs Pin Driver Suitable for High-Speed Test Systems", IEEE GaAs IC Symposium Digest 1987, pp. 91-94.

[136] T. Terada et. al., "A 6K-Gate GaAs Gate Array with a New Large Noise-Margin SLCF Circuit", IEEE Journal of Solid-State Circuits, Vol. SC-22, No. 5, October 1987.

[137] W. C. Terrell, "Direct Replacement of Silicon ECL and TTL SRAMs with High Performance GaAs Devices", IEEE GaAs IC Symposium Digest 1988, pp. 79-82.

[138] K. Thame, "GaAs Standard Cells Offer High Integration", What's New in Electronics, July 1989, pp. 100.

[139] "Third-generation decoding ICs for CD players", technical publication, Philips Electronic components and materials, No. 261.

[140] R. N. Thomas et. al., "Status of Device-Qualified GaAs Substrate Technology for GaAs Integrated Circuits", Proceedings of IEEE, July 1988, pp. 778-791.

[141] C. Toumazou, D. G. Haigh, S. J. Harrold, K. Steptoe, J. I. Sewell, R. Bayruns, "400MHz Switching Rate GaAs Switched Capacitor Filter", Electronics Letters, 29 March 1990, Vol. 26 No. 7.

[142] J. G. Tront, D. D. Givone, "A Design for Multiple-Valued Logic Gates Based on MESFET's", IEEE Transactions on Computers, Vol. C-28, No. 11, November 1979.

[143] C. Tsen et. al., "A Manufacturable Low-Power 16K-Bit GaAs SRAM", IEEE GaAs IC Symposium Digest 1987, pp. 181-184.

[144] K. Utsumi et. al., "Gigabit Optical Transmitter GaAs MESFET IC", Electronics Letters, Vol. 23, No. 8, 9 April 1987.

[145] A. Vladimirescu, S. Liu, "The Simulation of MOS Integrated Circuits Using SPICE2", Electronics Research Laboratory, Memorandum No. UCB/ERL M80/7, February 1980.

[146] C. Vogelsang et. al., "Complementary GaAs JFET 16K SRAM", IEEE GaAs IC Symposium Digest 1988, pp. 71-74.

[147] T. T. Vu et. al., "Low-Power 2K-Cell SDFL Gate Array and DCFL Circuits Using GaAs Self-Aligned E/D MESFET's", IEEE Journal of Solid-State Circuits, Vol. 23, No. 1, February 1988.

[148] T. T. Vu et. al., "The Performance of Source-Coupled FET Logic Circuits that Use GaAs MESFET's", IEEE Journal of Solid-State Circuits, Vol. 23, No. 1, February 1988.

[149] C. F. Wan et. al., "A Comparative Study of GaAs E/D MESFETs Fabricated with Self-Aligned and Non-Self-Aligned Processes", IEEE GaAs IC Symposium Digest 1987, pp. 133-136.

[150] A. D. Welbourn, "The Design and Implementation of Gallium Arsenide Digital Integrated Circuits", Ph.D. Thesis, March 1988, University of Edinburgh.

[151] A. D. Welbourn, "Gigabit Logic: a Review", IEE Proceedings, October 1982, Vol. 129, Part I, No. 5.

[152] S. H. Wemple, H. Huang, "Thermal Design of Power GaAs FETs", GaAs FET Principles and Technology, Artech House 1982, pp. 309-348.

[153] W. White et. al., "Integration of GaAs 4kbit Memory with 750 Gate Logic for Digital RF Memory Applications", IEEE GaAs IC Symposium Digest 1989, pp. 37-40.

[154] M. Wilson et. al., "GaAs-on-Silicon: A GaAs IC Manufacturer's Perspective", IEEE GaAs IC Symposium Digest 1988, pp. 243-246.

[155] R. Yamamoto et. al., "Design and Fabrication of Depletion GaAs LSI High-Speed 32-Bit Adder", IEEE Journal of Solid State Circuits, Vol. SC-18, No.5, October 1983.

[156] Y. Yamauchi et. al., "A 34.8 GHz 1/4 Static Frequency Divider using AlGaAs/GaAs HBTs", IEEE GaAs IC Symposium Digest 1989, pp. 121-124.

[157] N. Yokoyama et. al., "A GaAs 1K Static RAM Using Tungsten Silicide Gate Self-Aligned Technology", IEEE Journal of Solid State Circuits, Vol. SC-18, No.5, October 1983.

Author's Publications

[1] S. Lam, M. Reekie, B. Flynn, J. Mavor, "Gallium Arsenide Bit-Serial Cells for Digital Filters", 1988 Saraga Colloquium on Electronics Filters, May 1988, Digest No. 1988-77, pp. 9/1-9/3.

[2] H. M. Reekie, S. C. K. Lam, B. W. Flynn, "Single-wire GaAs IC Communication Technique", Electronics Letters, Vol. 25, No. 1, 5 January 1989.

Gallium Arsenide Bit-Serial Cells for Digital Filters

Stephen Lam, Martin Reekie, Brian Flynn, John Mavor

1. Introduction

This paper describes the design of some GaAs bit-serial arithmetic and control cells that form part of a cell library intended for use in the implementation of digital filters. Device processing and fabrication is being carried out by Honeywell Physical Sciences Center using an advanced self-aligned 1μm gate enhancement/depletion MESFET process. The individual cells have been assembled into multi-project chips for initial fabrication and design verification. Once validated the cells will be used to design a simple FIR filter and a universal adaptor [1] for wave digital filters.

GaAs has long been known for its superior speed capabilities. Digital GaAs IC fabrication technology has matured in recent years and LSI products are becoming commercially available for the first time. Current major applications and developments include a 32 bit GaAs processor [2], 16-kbit SRAM's [2], and the central processor electronics for the Cray 3 supercomputer [3] . Digital signal processing (DSP) is another area where GaAs offers significant speed advantages. Bit serial architectures provide a consistent, modular, testable, and efficient methodology for general DSP. They also minimise the problem of data communication between chips at high bit rates. The ultimate throughput in bit serial circuits can be very high using pipelined architectures. The objective of the present work is to realise the highest data rates possible by combining GaAs IC technology with pipelined bit-serial structures. To this end cells suitable as building blocks for such systems have been designed.

2. The cells

The cells include a programmable serial to parallel converter, a parallel to serial converter, an adder, an ECL buffer, a 6 bit multiplier, a programmable word delay and a clock generator. The nominal data word length of the serial/parallel converters in the current batch of designs is 8 bits, but this can be readily extended to 31 bits for specific applications. All of the other cells, with the exception of the multiplier, can be used for any word length. A counter has been designed for testing purposes. Brief details of each design are given in later sections.

3. System timing

Two signal paths are required to pass information between cells. The first is the single wire carrying the bit-serial word itself (LSB first) and the other carries a signal which pulses high when the LSB of the signal is available. The system word length is 3 to 31 bits, always longer

The authors are with the Department of Electrical Engineering, University of Edinburgh.

than the input data word to allow for bit-growth and to maintain accuracy. The word format is fixed point fractional 2's complement. In order to maintain good dynamic range, the start of the data word is delayed a number of clock cycles after the start of the system word. This delay is pre-programmed into the serial to parallel converter. The system uses two phase non-overlapping clocks at a nominal rate of 500 MHz.

4. Circuit configuration

The most common circuit configuration is direct-coupled FET logic (DCFL), which does not require level shifting between logic gates and is therefore most suitable for designs that require high density and low power dissipation. The basic inverter consists of a depletion load pull-up and an enhancement pull-down device. The structure of the cells is based on the psuedo-dynamic DCFL latch. Fully static edge triggered latches are too large for use in pipelined bit-serial circuits.

Although the latches used here are similar in appearance to NMOS, several features of their operation are different. By nature of the Schottky gate MESFET structure, a significant gate current flows when the enhancement pull-down device is ON. The transmission gates (TG) are depletion devices and are driven by clock signals that never forward bias the Schottky gate. This reduces the current drain on the clock signals and prevents the data from being mixed with the clock. Data leakage into the clock signals would affect the operation of other TG's and lead to further signal degradation. The clock signals need to be sufficiently negative going, to fully turn OFF the depletion TG's. Where non-overlapping clocks are used, their separation must be small to prevent the latched signal being lost as gate current in the following inverters. The TG's and the driven inverters need to be correctly ratioed to allow good signal transfer. The forward inverter is twice the size of the feedback inverter to prevent hazards with the next stage at worst case clock skew.

5. The cell designs

5.1. The converters

These converters allow an eight bit-parallel system to interface correctly with the bit-serial components fabricated as part of this work. The bit-serial words used are at least nine bits long and may be substantially greater, the actual number of bits and the optimum position of the eight data bits being determined by the system application. Therefore the parallel to serial converter must be programmable to allow the eight bits of data to be placed at any position within the bit-serial word. Any extra LSBs are set to zero while any guard-bits (extra MSBs) are set equal to the MSB of the eight-bit data word. PRBS counters are used to carry out this operation because of their inherent simplicity. Their disadvantage is their "un-natural" way of counting which implies that the system operator requires a table of "equivalents"; to have 3 guard bits may require the value "15" to be entered. This converter also produces the LSB pulses required in a bit-serial system.

The serial to parallel converter uses only one PRBS counter to count up to the MSB of the data word and then loads the eight bit word into output latches, which have ECL compatible outputs.

All the PRBS have been designed to allow a maximum count of 31.

5.2. The adder and multiplier

The multiplier uses a modified Booth's algorithm with two levels of encoding. Under this scheme each slice encodes three bits of the coefficient word. The least significant bit overlaps the msb of the previous coefficient slice and so a multiplier with a six bit coefficient, as in this mask set, requires three slices.

The multiplier slice consists of a carry/borrow block, a sum/difference block, a data selector, a one or two bit sign extension block, and the coefficient recoder. The blocks are controlled by the LSB pulse from the previous slice. The LSB pulse is delayed by the the appropriate latency to maintain system timing, as in all other bit serial cells. The scheme and the word format require the first, middle and last slices to have slightly different sign extension and recoding circuits. In the slice the logic blocks are partitioned as far as possible without increasing the slice latency so that a high clock rate can be maintained throughout the circuits.

The latency of the multipliers in a module or system is the major factor in determining the system latency. In a recursive system the system latency determines the word length, and therefore the ultimate sampling rate for the system. The modified Booth's algorithm reduces latency without causing an excessive increase in complexity. The latency of a multiplier is given by (1.5m - 1) for an m bit coefficient word, and is independent of the data word length.

5.3. The clock generator

The clock generator is a cross-coupled NOR structure with two level shifting buffer stages at the output. The output "high" and "low" signals are +0.5V and -0.5V, respectively. The need for the negative "low" value arises from the use of depletion pass-transistors in the rest of the system. The output stage comprises two push-pull depletion devices and a depletion pull-down device. The push-pull drivers are driven by two buffered FET logic type level shifters which are themselves driven by normal DCFL gates.

5.4. The variable word delay

This block provides a programmable word delay of three to thirty one bits. Twenty eight delay elements are tapped by three levels of multiplexers (mux), made up of eight 4:1, two 4:1, and one 2:1 mux. The LSB pulse also goes through the same process as the system word. Only twenty eight stages are needed because all the cell inputs and outputs are pipelined and the mux's introduce 1 bit of latency. This block is most useful in FIR filters.

6. Future work

The cells will be individually tested for functionality and then an FIR filter such as that shown in Fig. 1 will be built. Because of the chip to chip latencies, ultimate filter performance will not be obtained but the modularity and flexibility of bit serial techniques will be demonstrated.

The next step will be to optimise the cells and integrate them into an adaptor so that a wave filter can be implemented, as shown in Fig. 2. Wave digital filters have many excellent stability properties that are directly inherited from the reference analogue filters from which they are derived. Their main disadvantage is their complexity and the resulting low sample-rate. This work should allow construction of wave filters which will be usable at very high frequencies.

7. References

[1] REEKIE, H M, PETRIE, N, MAVOR, J, DENYER, P B and LAU, C H: "The Design and Implementation of Digital Wave Filters using Universal Adaptors", IEE CRSP Special Issue on The Design and Application of Digital Signal Processors. Proc IEE (F) Vol 131, no 6, pp 615-622 Oct. 1984.

[2] Military/Aerospace Newsletter, Electronics, Mar 31 1988.

[3] Technical Digest, 1987 IEEE GaAs IC Symposium, Portland, Oregon.

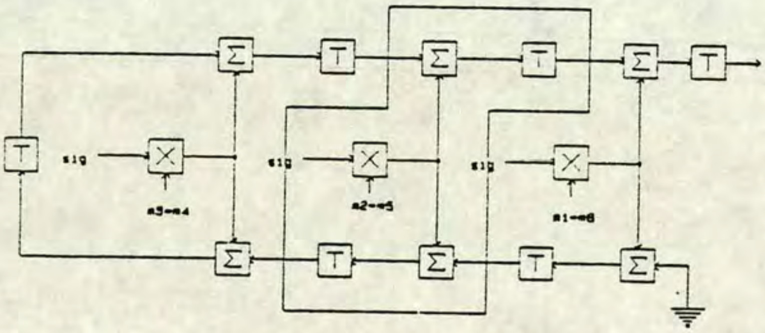


Fig. 1 The FIR filter

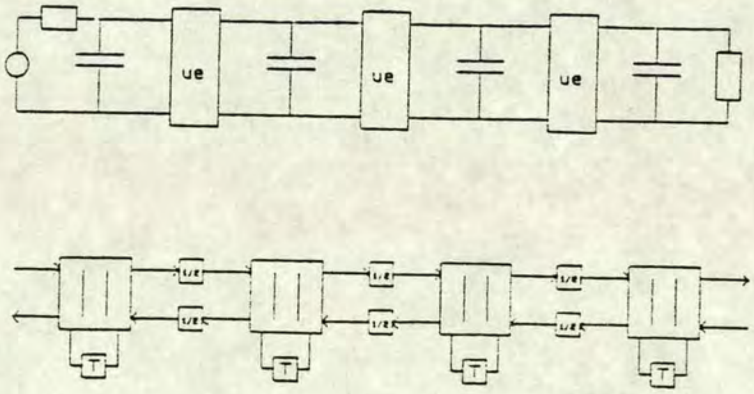


Fig. 2 The wave filter implemented from parallel adaptors

By choosing a third level below the normal logic 0, in this case −0·5 V, problems in both compatibility with conventional DCFL circuits and generation are eliminated. Provided the gate-source and protection diodes at the receiving end do not break down, an incoming negative voltage appears as logic 'low'. Generation would require an extra pull-down device to a negative rail of −0·6 V, already available in the GaAs cells for the on-chip clock generator.[1] This voltage is also well above the backgating potential of the process. There is the further advantage that the negative-going MSB signal is restricted to the *I/O* pads and buffers only. Fig. 2 shows an idealised example of the composite signal corresponding to a data word '11001'.

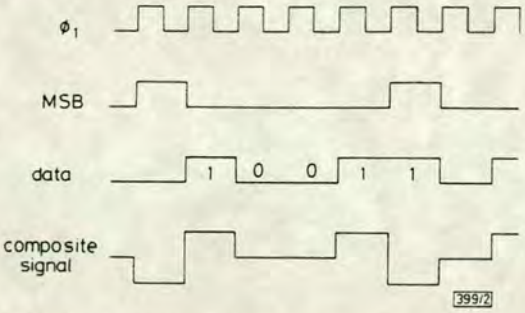


Fig. 2 *Single-wire composite signalling format*

Note that while this technique would also be possible in other technologies, for example CMOS, output buffers and interconnect are not such a problem at lower clock rates, and so its advantages are not so apparent. Also, it is not possible in CMOS to have signals outside the V_{SS}–V_{DD} range.

I/O circuits and simulation: Fig. 3 shows the generation and detection circuitry, including the inverting *I/O* latches. The output buffer is similar to the buffer used in the conventional two-wire system, and consists of a composite enhancement/depletion MESFET pull-up device driven logically by $\overline{\text{MSB}}$. $\overline{\text{DATA}}$, and two enhancement pull-down devices driven by DATA . $\overline{\text{MSB}}$ and MSB. The MSB signal is level-shifted for correct switching. The sizes of the NOR gates, inverters, level shifter and the output MESFETs are matched for optimum capacitive drive. Most of the power is dissipated in the output devices, which need to be distributed and placed as near to the pads as possible. The receiving end employs a protection diode, a carefully ratioed inverter with a depletion pull-down device to detect the negative level, and a latch for the data signal. The depletion inverter was simulated to have a noise margin of about 400 mV for the 'off' level. This circuit was designed for a nominal clock rate of 500 MHz, or a data bit rate of 250 MHz.

This circuit was simulated using a modified SPICE 2G.6 program based on the Curtice MESFET model.[3] The models were matched to the self-aligned gate E/D process from Honeywell Physical Sciences Center. The threshold voltages for the enhancement and depletion devices are 0·2 V and −0·5 V, respectively. Fig. 4 shows the generated composite signal, representing $\overline{\text{DATA}}$ and $\overline{\text{MSB}}$, and the latched MSB and MSB-restored data signals, with the associated one bit latency from pipelining. The bit rate was 250 MHz. A 5 pF capacitor represented the between-chip interconnect. The data words were '001' followed by '110' when reading the results. Note that the signals were LSB first. Correct detection and latching were maintained when simulated at 700 MHz clock rate and when worst-case process deviations were included. The total power dissipation of this circuit was approximately 29 mW, as compared to (2 × 22·3 mW) for the conventional system. The estimated layout area is about 40% less than the previous system.

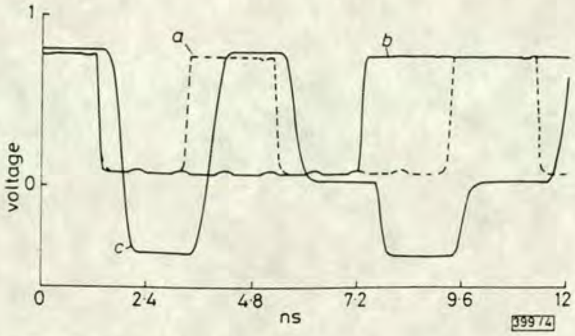


Fig. 4 *SPICE simulation results*
a MSB *b* Data *c* Composite signal

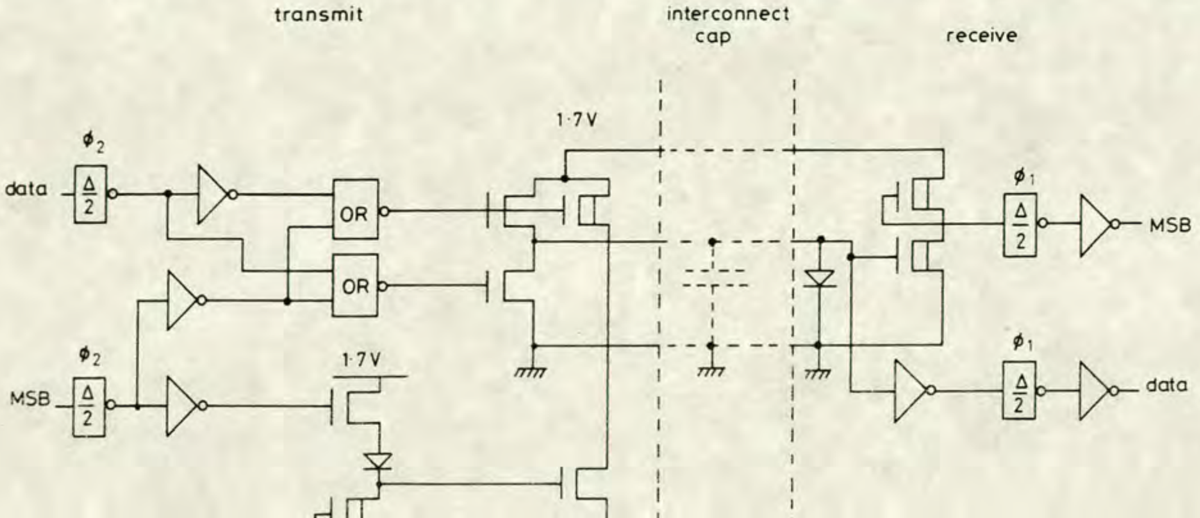
Conclusion: A three-level off-chip signalling technique was designed and simulated for GaAs ICs. The results showed significant reduction in total *I/O* circuit area and power consumption. Furthermore, the throughput rate is not affected and off-chip signal synchronisation is maximised, making it ideal for the next generation of bit-serial GaAs cells and other bit-serial GaAs systems in general.

Acknowledgment: We are grateful to the UK Science & Engineering Research Council for funding this project, and Honeywell Inc., Physical Sciences Center, USA, for providing processing.

H. M. REEKIE
S. C. K. LAM
B. W. FLYNN

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

18th August 1988



By choosing a third level below the normal logic 0, in this case $-0{\cdot}5$ V, problems in both compatibility with conventional DCFL circuits and generation are eliminated. Provided the gate-source and protection diodes at the receiving end do not break down, an incoming negative voltage appears as logic 'low'. Generation would require an extra pull-down device to a negative rail of $-0{\cdot}6$ V, already available in the GaAs cells for the on-chip clock generator.[1] This voltage is also well above the backgating potential of the process. There is the further advantage that the negative-going MSB signal is restricted to the *I/O* pads and buffers only. Fig. 2 shows an idealised example of the composite signal corresponding to a data word '11001'.

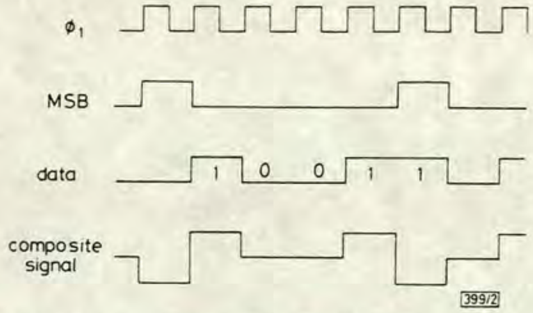


Fig. 2 *Single-wire composite signalling format*

Note that while this technique would also be possible in other technologies, for example CMOS, output buffers and interconnect are not such a problem at lower clock rates, and so its advantages are not so apparent. Also, it is not possible in CMOS to have signals outside the V_{SS}–V_{DD} range.

I/O circuits and simulation: Fig. 3 shows the generation and detection circuitry, including the inverting *I/O* latches. The output buffer is similar to the buffer used in the conventional two-wire system, and consists of a composite enhancement/depletion MESFET pull-up device driven logically by $\overline{\mathrm{MSB}}$. $\overline{\mathrm{DATA}}$, and two enhancement pull-down devices driven by DATA . $\overline{\mathrm{MSB}}$ and MSB. The MSB signal is level-shifted for correct switching. The sizes of the NOR gates, inverters, level shifter and the output MESFETs are matched for optimum capacitive drive. Most of the power is dissipated in the output devices, which need to be distributed and placed as near to the pads as possible. The receiving end employs a protection diode, a carefully ratioed inverter with a depletion pull-down device to detect the negative level, and a latch for the data signal. The depletion inverter was simulated to have a noise margin of about 400 mV for the 'off' level. This circuit was designed for a nominal clock rate of 500 MHz, or a data bit rate of 250 MHz.

This circuit was simulated using a modified SPICE 2G.6 program based on the Curtice MESFET model.[3] The models were matched to the self-aligned gate E/D process from Honeywell Physical Sciences Center. The threshold voltages for the enhancement and depletion devices are 0·2 V and $-0{\cdot}5$ V, respectively. Fig. 4 shows the generated composite signal, representing $\overline{\mathrm{DATA}}$ and $\overline{\mathrm{MSB}}$, and the latched MSB and MSB-restored data signals, with the associated one bit latency from pipelining. The bit rate was 250 MHz. A 5 pF capacitor represented the between-chip interconnect. The data words were '001' followed by '110' when reading the results. Note that the signals were LSB first. Correct detection and latching were maintained when simulated at 700 MHz clock rate and when worst-case process deviations were included. The total power dissipation of this circuit was approximately 29 mW, as compared to (2 × 22·3 mW) for the conventional system. The estimated layout area is about 40% less than the previous system.

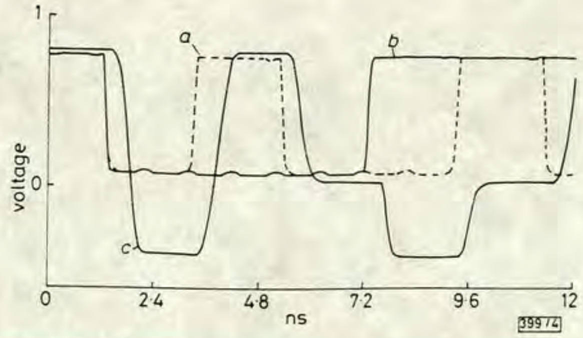


Fig. 4 *SPICE simulation results*
a MSB *b* Data *c* Composite signal

Conclusion: A three-level off-chip signalling technique was designed and simulated for GaAs ICs. The results showed significant reduction in total *I/O* circuit area and power consumption. Furthermore, the throughput rate is not affected and off-chip signal synchronisation is maximised, making it ideal for the next generation of bit-serial GaAs cells and other bit-serial GaAs systems in general.

Acknowledgment: We are grateful to the UK Science & Engineering Research Council for funding this project, and Honeywell Inc., Physical Sciences Center, USA, for providing processing.

H. M. REEKIE
S. C. K. LAM
B. W. FLYNN

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

18th August 1988

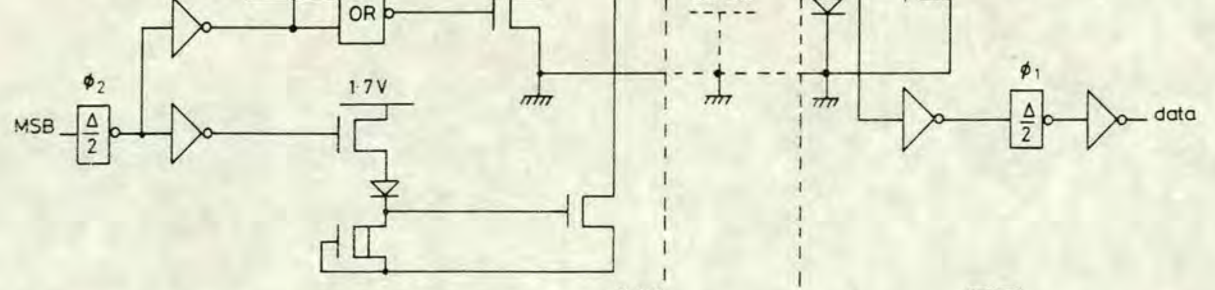


Fig. 3 *Composite signal generation and detection circuit*