GaAs Design Methodology & Performance Estimates for Very High Speed Circuits Using Normally-Off Classes of Logic

K.Eshraghian, A.Blanksby, R.Sarmiento†, C.C. Lim

Centre for Gallium Arsenide VLSI Technology
University of Adelaide, Adelaide, SA 5005, Australia
† Centro de Microelectrónica Aplicada
University of Las Palmas, Las Palmas de Gran Canaria, Spain

Abstract

This paper compares three compatible normally–off classes of digital logic, namely DCFL, SDCFL and SFFL, that have been found to be suitable for Gallium Arsenide VLSI implementation. A Merged Logic *approach that exploits the advantages of each of the logic classes to provide superior circuit performance is demonstrated. A design methodology using* Ring Notation *for mapping of logic in a systematic way is also presented and the underlying principles highlighted using the design of a Polynomial Evaluator Processing Element.*

1 Introduction

Gallium Arsenide has shown potential as a technology suitable for the implementation of Very High Speed Systems due to high electron mobility and the semi–insulating nature of the substrate. The lower parasitics result in significant reduction in dynamic power dissipation when compared to Silicon. Other advantages include optoelectrical properties, higher temperature tolerance and greater radiation hardness. These properties make GaAs a good contender for high speed applications including digital signal processing, broadband communications, supercomputing, satellite communications, etc. With the availability of foundries offering an Enhancement/Depletion Self–Aligned process with a sub–micron feature size, GaAs VLSI systems in a true sense are now achievable.

In this paper the characteristics and performance behaviour of three different classes of Normally–Off static logic which have shown promise for VLSI implementation are presented, namely Direct Coupled FET Logic (DCFL), Source–Follower Direct Coupled FET Logic (SDCFL) and Source–Follower FET Logic (SFFL). The basic logic structure for each class is characterised and optimised in terms of noise margin, speed, area, dissipation, and fan–out sensitivity.

A *Merged Logic* approach to circuit design is proposed in this paper. This concept combines DCFL with SDCFL and SFFL so that the advantages of each logic class are exploited and circuit performance is achieved which is superior to that obtained if the circuit had been implemented with any one of the logic classes exclusively.

When realising Very High Speed VLSI circuits, clocking technique and communication strategy, power distribution and obviously speed become critical design issues. Thus performance of the logic gates are influenced by the design methodology. *Ring Notation* [1, 2], a novel approach for the mapping of circuits, is used to translate DCFL, SDCFL, and SFFL circuits into compact, high performance layouts. To facilitate a better understanding of this design methodology the design of a Polynomial Evaluator Processing Element is used to illustrate the concept of Merged Logic as part of an overall strategy for a GaAs VLSI implementation.

2 Gallium Arsenide Technology

2.1 GaAs Process Technology

Previous work [3, 4] undertaken at the Centre for Gallium Arsenide VLSI Technology identified the Vitesse H–GaAs–II $0.8\mu m$ Technology as a process suitable to realise high density VLSI chips. Moreover it is available from several sources such as Thomson C.S. (France) and Fujitsu (Japan). The technology provides variable gate length transistors and up to four layers of metal interconnects resulting in compact layouts with low power dissipation.

When designing GaAs digital circuits having VLSI complexity, variations in transistor parameters and hence circuit performance due to process spread must be addressed at the beginning of the design cycle. For E/D based logic the dominant effect is the change in transistor threshold voltages. An increase in V_{te} due to process spread results in a *Slow* E–MESFET while a decrease in V_{te} gives a *Fast* E–MESFET. Similarly a *Fast* D–MESFET is produced when V_{td} is more negative and a *Slow* D–MESFET results when V_{td} is more positive. As the H–GaAs–II process uses an Additive–Implantation technique, circuits are produced with transistors which are predominantly Fast–Fast or Slow–Slow and not Fast–Slow or Slow–Fast.

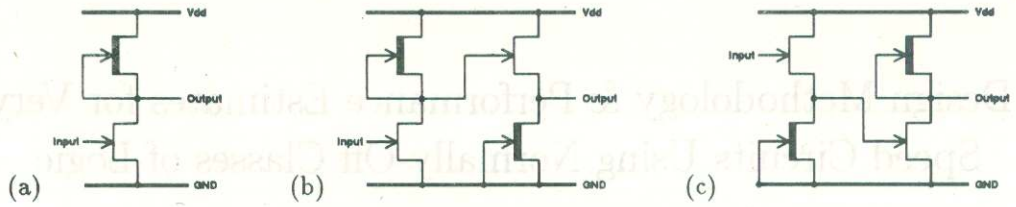

Figure 1: (a) DCFL Inverter (b) SDCFL Inverter (c) SFFL Inverter

2.2 DCFL, SDCFL & SFFL

The simplest logic class in GaAs is Direct Coupled FET Logic (DCFL). It uses a Depletion MESFET transistor as the active load and Enhancement MESFET transistors to implement logic functions. Fig. 1(a) gives the basic DCFL inverter structure where the circuit designer must select the transistor gate length L and width W of each MESFET to achieve the required performance for each logic gate. The supply voltages are usually Vdd = 2.0V and GND = 0.0V although Vdd = 0.0V and GND = -2.0V are often used to facilitate interfacing to ECL (Emitter Coupled Logic) circuits in Silicon. An optimal DCFL inverter with satisfactory logic levels and noise margin at different process spread was found to have a β ratio in the range 10–12 where β is defined as $\frac{W_{enh}:L_{enh}}{W_{dep}:L_{dep}}$

Although DCFL is the simplest and fastest of the static logic classes it has several shortcomings, the most notable being the low noise margin ($\approx$ 100mV). This is due to the output voltage being limited by the barrier height of the MESFET Schottky diode at the input to the next DCFL stage ($\approx$ 700mV). Other limitations include the sensitivity of the gate delay to fan–in, fan–out, and load capacitance and the small temperature stability margin of the basic gate.

In order to improve noise margins and circuit fan–out performance two additional Normally–Off classes of static logic are considered in this paper, SDCFL and SFFL. SDCFL, as the name implies, consists of a DCFL stage using a source–follower as a buffer on the output. In SDCFL, the voltage levels at the input and output are DCFL compatible. In contrast SFFL precedes a DCFL stage with a source–follower as a buffer on the input which results in different logic levels at the input and output. Fig. 1(b) and Fig. 1(c) give the basic inverter for the SDCFL and SFFL classes of logic respectively. The DCFL, SDCFL, and SFFL inverters can be expanded into NOR gates by adding additional E–MESFETs in parallel in the input stage of each gate. Due to degradation of performance, circuits in our design methodology are restricted to parallel branches in the input path, that is OR/NOR gates only are permitted.

2.3 Ring Notation

Performance goals in terms of small area, low power and high speed are very important for the general acceptance of GaAs technology. However a design methodology for easy design structuring and improved reliability is needed in GaAs. In this study *Ring Notation* [1, 2] is presented as an intermediate method of describing the layout of a circuit designed using the *Merged Logic* approach proposed. Ring Notation allows signal paths to be highlighted and the interconnection strategy to be formulated before the design proceeds to a layout. Complex structures can be readily and systematically mapped thus providing an easy translation method from symbolic notation to mask layout and hence ensuring the portability of designs. In the Ring Notation the transistors are laid–

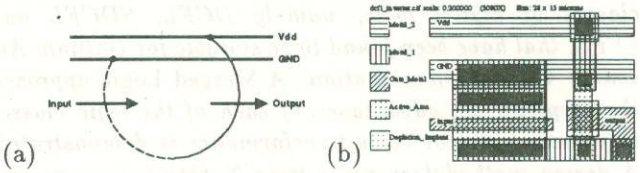

Figure 2: DCFL Inverter (a) Ring Notation (b) Layout.

out horizontally while the GND bus is placed between the circuit and Vdd bus, which reduces their self–inductance and hence susceptibility to current transients. Fig. 2(a) illustrates the Ring Notation for a DCFL inverter where the solid line represents the Depletion MESFET and the dashed line represents the Enhancement MESFET. The corresponding mask layout is given in Fig. 2(b) demonstrating the power of this abstraction technique.

3 Optimisation of Merged Logic

In the *Merged Logic* approach DCFL is used predominantly to achieve high packing density and maximum circuit performance. SDCFL is used to drive capacitive loads and realise the *And–Or–Invert (A–O–I)* function while SFFL is used to implement large fan–out. In this way significant performance improvement can be achieved. The performance of different logic classes have been evaluated in terms of noise margin, power dissipation, fan–in, fan–out, Average Transition Time and Average Delay. The noise margin has been measured using the Maximum Square Method [5]. The rise–time (t_{rise}) and the fall–time (t_{fall}) were calculated when the output voltage crossed the 10% and 90% points of the output voltage swing, $V_{oH} - V_{oL}$. The high–to–low (t_{phl}) and low–to–high (t_{plh}) propagation delays were measured from the 50% voltage swing of the input to the 50% voltage swing of the output.

The performance of the DCFL, SDCFL and SFFL inverters for several fan–out is summarised in Table 1. It is evident that with increasing fan–out the performance of DCFL is degraded by increasing rise–time, t_{rise} and

Table 1: **Performance of Optimal DCFL, SDCFL and SFFL Inverters.**

Logic Class	Area (μm x μm)	Fan Out	Power (μW)	Noise Margin (mV)	V_{oH} (mV)	V_{oL} (mV)	t_{rise} (ps)	t_{fall} (ps)	t_{phl} (ps)	t_{plh} (ps)
DCFL	21.9 x 15.8	**1**	200	140	640	102	125	82	91	31
		3	204	114	591	102	256	109	147	64
SDCFL	36.0 x 15.8	**1**	592	158	669	27	128	99	89	69
		3	608	125	620	113	320	236	162	168
SFFL	38.5 x 15.8	**1**	644	375	659	122	111	91	81	75
		3	654	235	607	122	206	136	132	120

decreasing V_{oH}. This is due to the reduced ability of the D–MESFET to drive the E–MESFETs of the fan–out load. The fan–in capability of DCFL is restricted by the drain–to–source leakage current of the E–MESFET which, when multiplied by the fan–in, reduces V_{oH}. The SDCFL class of logic does offer some advantages to the circuit designer. Fig. 4(a) demonstrates the ability of SDCFL to drive capacitive loads significantly faster than DCFL. Another important application for SDCFL is the *Wired–Or* structure given in Fig. 3 which implements the logical expression given by (1) with the delay of a single SDCFL NOR gate, typically $150ps$ (the delay for an equivalent DCFL structure is about $250ps$).

$$Z = \overline{A+B} + \overline{C+D} = \overline{(A+B)\cdot(C+D)} \qquad (1)$$

This makes the SDCFL Wired–Or gate a good choice when implementing logical expressions which can be manipulated into the And–Or–Invert form of (1). The fan–out of SDCFL, however, is limited by the ability of the D–MESFET of the source–follower to discharge the gate capacitances of the following stages which can be significant. The SFFL class of logic has the ability to drive

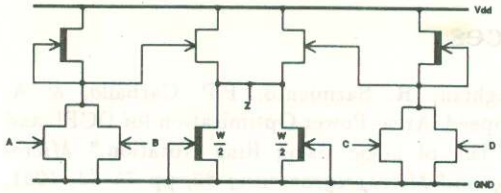

Figure 3: SDCFL Wired–Or Structure.

quite large fan–out. Fig. 4(b) gives the comparative fan–out performance of DCFL and SFFL. This means SFFL circuits could be used to implement clock drivers and buffer circuits when large fan–out is required. The fan–in of SFFL is limited by the on drain–to–source current of the input E–MESFET which, when multiplied by the fan–in, drives the Schottky diode present at the gate of the output E–MESFET excessively and V_{oL} is degraded. Both SDCFL and SFFL gates dissipate more power and occupy more area than a DCFL gate. Thus a *Merged Logic* approach to circuit design is a natural progression from the results of this comparative study. It involves identifying critical parts of the circuit where SDCFL or SFFL could be used to improve circuit performance such as using SDCFL to drive capacitive loads and realise the A–O–I function, and SFFL to implement large fan–out.

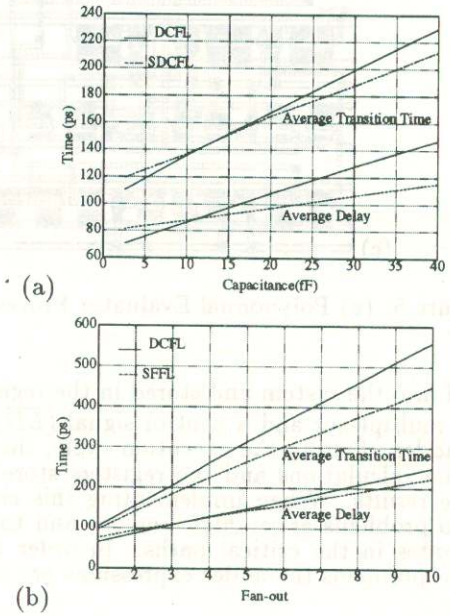

Figure 4: (a) Capacitive Load Performance of DCFL and SDCFL (b) Fan–out Performance of DCFL and SFFL.

4 The Polynomial Evaluator

In real–time image processing, there often exists a need for a very fast polynomial evaluator (PE). The general representation of an n order polynomial is given by:

$$y(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0 \qquad (2)$$

In order to evaluate this polynomial it is necessary to transform it into a bit–serial form [6] suitable for a GaAs implementation. The PE can calculate any point on the curve from an initial starting point $y(x_i)$, using a step size h, with the application of a recursive addition process on the individual bits of both the initial point and a set of previously calculated coefficients, $d_1, d_2, \ldots, d_n$.

$$\begin{aligned} y(x_{i+1}) &= y(x_i) + d_1(x_i) \\ d_1(x_{i+1}) &= d_1(x_i) + d_2(x_i) \\ &\vdots \\ d_{n-1}(x_{i+1}) &= d_{n-1}(x_i) + d_n(x_i) \end{aligned} \qquad (3)$$

where $d_n(x_{i+1}) = d_n(x_i)$ and $x_{i+1} = x_i + h$. Fig. 5(a) shows the basic PE cell structure. A single PE cell may be replicated horizontally to increase n, the polynomial order, and replicated vertically and pipelined to increase the bit size. The initial starting point and coefficient bits

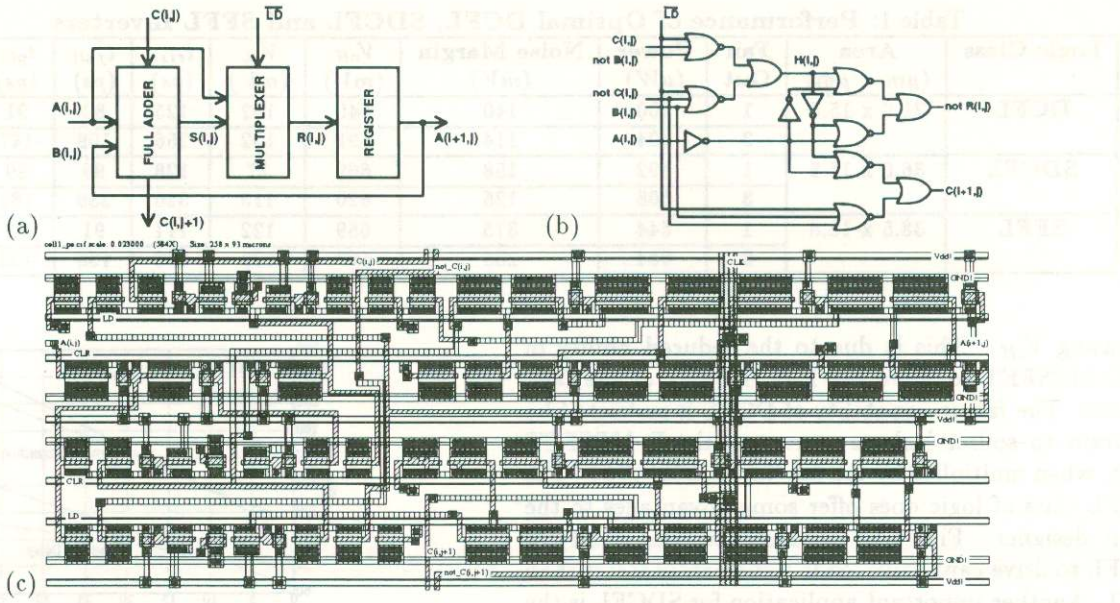

Figure 5: (a) Polynomial Evaluator Processing Element Cell (b) Logic representation (c) Mask layout.

are loaded into the system and stored in the registers by using the multiplexer and a control signal ($\overline{LD}$) to bypass the adder. During the execution cycle, the adders perform the calculations and the registers store the intermediate results. When implementing this circuit in GaAs, two problems arise: high fan–out and too many levels of gates in the critical paths. In order to overcome these problems the adder expressions are modified as follows:

$$R_{i,j} = S_{i,j}LD + A_{i,j}\overline{LD} = (B_{i,j} \oplus C_{i,j})LD \oplus A_{i,j} = H_{i,j} \oplus A_{i,j} \quad (4)$$

$$S_{i,j} = A_{i,j} \oplus B_{i,j} \oplus C_{i,j} \quad (5)$$

$$C_{i+1,j} = A_{i,j}(B_{i,j} \oplus C_{i,j})LD + B_{i,j}C_{i,j}LD = H_{i,j}A_{i,j} + B_{i,j}C_{i,j}LD \quad (6)$$

The logic representation of these equations are shown in Fig. 5(b). The XOR operations are realised with the SDCFL Wired–Or structure. The register is designed in DCFL with six gates. Although the output of the register has a fan–out of five, it is not possible to buffer it with SFFL due to incompatibility of logic levels. A reduction in fan–out is required which can be achieved by using an inverter for the input connecting one processing element stage to the next stage, $A_{(i,j)}$.

The mask layout of the Polynomial Evaluator Processing Element is given in Fig. 5(c). Simulations of this cell indicate a maximum clock frequency of 1 GHz, with a mean delay per logic gate of $96ps$. Power consumption is $6.5mW$ with a basic cell area of $258\mu m \times 94\mu m$, giving a density of $\approx$ 3200 transistors per square millimetre.

5 Conclusion

Three compatible, normally–off classes of logic, DCFL, SDCFL, and SFFL, have been optimised taking into account the effect of process spread. From this study a *Merged Logic* approach for designing high performance VLSI circuits has emerged based on the use of each class of logic where it is best suited. A design methodology using *Ring Notation* has been introduced and adopted for the design of Merged Logic. Ring Notation produces high density, full custom layouts and is very suitable for the symbolic representation of the mask layout. Using this method a Polynomial Evaluator Processing Element for a real–time image processor has been designed and simulated with a maximum clock frequency of 1 GHz for typical circuits with the average speed of a simple logic gate being $90ps$. This design has been optimised to function over the entire range of the process spread and a density of 3200 transistors/mm^2 has been achieved.

References

[1] K. Eshraghian, R. Sarmiento, P.P. Carballo, & A. Núñez, "Speed–Area–Power Optimisation for DCFL and SDCFL Class of Logic Using Ring Notation." *Microprocessing and Microprogramming 32,* pp 75–82, 1991, North Holland.

[2] K. Eshraghian, "Fundamentals of Ultra–High Speed Systems: GaAs VLSI Technology." In press, Prentice Hall.

[3] A. Beaumont-Smith & K. Eshraghian, "Comparison of Gallium Arsenide MESFET Process Technologies." *Technical Report GAAS92–14,* Centre for Gallium Arsenide VLSI Technology, August 1992.

[4] K. Eshraghian, E. Chu, A. Moini & S. Cui, "Design and Comparison of GaAs Static Logic Suitable for VLSI Implementation." *Technical Report GAAS92–15,* Centre for Gallium Arsenide VLSI Technology, November 1992.

[5] S. Long & S. Butner, "Gallium Arsenide Digital Integrated Circuit Design." McGraw Hill, 1990, pp 146–147.

[6] K. Eshraghian, "Optimisation of High Speed VLSI Systems: Physical Mapping." *Commonwealth Workshop on Mathematical Modelling in Circuit Design,* pp 161–177, Kandy, Sri Lanka, April 1992.