A Gallium Arsenide Mutual Exclusion Element

Nick Michell, Erik Brunvand, and Kent Smith

UUCS-93-029

Department of Computer Science
University of Utah
Salt Lake City, UT 84112 USA

Dec. 1993

A Gallium Arsenide Mutual Exclusion Element

Nick Michell, Erik Brunvand, and Kent Smith*
University of Utah
Dept. of Computer Science
3190 Merrill Engineering Building
Salt Lake City, Utah 84112

Abstract

A mutual exclusion element is a key component in building asynchronous and self-timed circuits. As part of our effort to design high performance self-timed circuits, we have designed a mutual exclusion element in gallium arsenide. This circuit has been fabricated in a 1.2μ process and tested. A test circuit using matched delay lines was used to verify that the circuit operates correctly when input requests arrive simultaneously.

1 Introduction

Concurrency plays a major role in the design of modern digital systems. Digital circuits, for example, are inherently concurrent at a very low level. Even in a small circuit, whether implemented as an integrated circuit, or as a collection of higher level circuit modules, there may be a great many parts of the circuit active at the same time. In addition, many applications of VLSI are of systems that are concurrent at a higher level.

Unfortunately, a fundamental problem exists whenever these concurrent systems lack a common time reference and engage in communication. The problem involves the difficulty of building a reliable circuit to synchronize the signals used to communicate [1, 2]. This problem shows up in a variety of situations ranging from asynchronous inputs of standard synchronous systems, to communicating asynchronous state machines, to any situation when a resource is shared between two or more processes that lack a common time reference. The circuit used to perform the synchronization is usually called an *arbiter* [3].

*This work is supported in part by DARPA award J.FBI.89.102, and in part by NSF award MIP-9115372

Arbiters are especially common in asynchronous and self-timed systems where there is no common time reference for the system and parts of the system cooperate through communication [4]. These types of systems are becoming popular as the cost of distributing a global synchronizing signal such as a clock continues to grow in advanced VLSI technologies [5, 6, 7].

A mutual exclusion element, or interlock element, is the key component of an arbiter. The function of this element is to prevent two signals, which may arrive within a very short time interval, from ever being asserted simultaneously. Signals are presented at the input to the mutual exclusion element. Given a sufficiently long time interval between the arrival of inputs, these signals are passed through the mutual exclusion element in the order in which they arrive. However, if the difference in arrival time is small, either choice is acceptable as long as only a single output of the mutual exclusion element is asserted. The mutual exclusion element must guarantee this behavior in the face of possible metastable behavior of the sampling circuitry [1].

2 Mutual Exclusion Element Design

A standard method for building a mutual exclusion element involves a cross-coupled latch followed by a threshold detector. A generalized mutual exclusion element of this type is shown in Figure 1. Inputs are captured by the cross-coupled latch. If the inputs are separated by enough time, this latch will set or reset quickly. However, if the inputs arrive close enough together in time, the latch could exhibit metastable behavior in which both of the outputs are at the same level until the feedback in the latch causes it to settle into one stable state.

The latch outputs are connected to the inputs to the threshold detector. As long as these inputs differ by less than the threshold, neither output will be asserted. A difference in the inputs to the detector is reinforced by the positive feedback of the cross-coupled latch. By the time the threshold detector fires, the positive feedback of the latch ensures that the state will not change without a change in the inputs. A restriction on the environment is that requests may not be deasserted until they are acknowledged, so no change in the inputs will occur.

An important point to note is that the propagation delay of this circuit is unbounded. While the mutual exclusion circuit will never assert both outputs simultaneously, two inputs arriving at the same time may cause the latch to go into the metastable state, and the circuit can remain in this state for an indeterminate length of time [1].

An NMOS mutual exclusion element from [4] is shown in Figure 2. *Req1* and *Req2* may occur

concurrently, however, the mutual exclusion element ensures that only one of *Ack1* and *Ack2* is asserted. The threshold detector uses two depletion pullups and a pair of pass transistors, where the control voltage for one pass transistor is the input to the other. Neither *Ack1* nor *Ack2* will be pulled low until the voltages at the output of the latch differ by at least the threshold voltage of the enhancement device.

3 A Gallium Arsenide Mutual Exclusion Element

Since a high quality oxide cannot be grown in gallium arsenide, MESFETs instead of MOSFETs are typically used. Direct-coupled FET Logic (DCFL) circuits resemble NMOS circuits and offer the best speed power product of any static GaAs logic family. The key difference for the designer is the non-insulating gate of the MESFET. The gate of the MESFET is connected to the source and drain regions by a Schottky diode, which clamps the gate voltage about 700mV above the drain. Series connected transistors, such as pass transistors, work very poorly and are seldom used in DCFL circuits since they reduce the already slim noise margins[8]. While the NMOS version of the mutual exclusion element could probably be used reliably in GaAs with proper transistor sizing, we decided to take a different approach.

A new mutual exclusion circuit can be designed by considering the generalized case. What is required is an input latch and a threshold detector. The latch is unchanged; it is still a cross-coupled NOR gate. The threshold detector is built from a differential amplifier and a pair of source followers. The key requirement for this component is that there is a range of input voltages where neither output is asserted. SPICE simulation of the DC transfer characteristics shows that this is indeed the case (Figure 3). A schematic of the mutual exclusion element is shown in Figure 4, and the composite layout is shown in Figure 5. The mutual exclusion cell has been designed for use with the PPL tool suite [9, 10], a two-dimensional tiled layout system.

4 Test Results

A circuit containing the mutual exclusion element was submitted to MOSIS and fabricated in the Vitesse 1.2μ process. In order to test the mutual exclusion element, the setup in Figure 6 was used. Two matched, voltage-controlled delay lines accept an identical input and generate the two requests to the mutual exclusion element. One delay line has a reference voltage as its control input, while the other is driven from an external source. This allows us to precisely adjust the relative timing of the two requests. When the relative delay is long, the mutual exclusion element will choose the first input that arrives. When

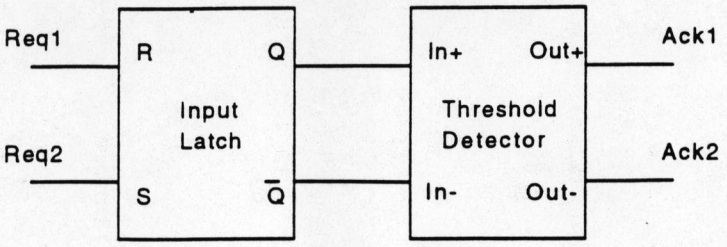

Figure 1: Generalized Mutual Exclusion Element

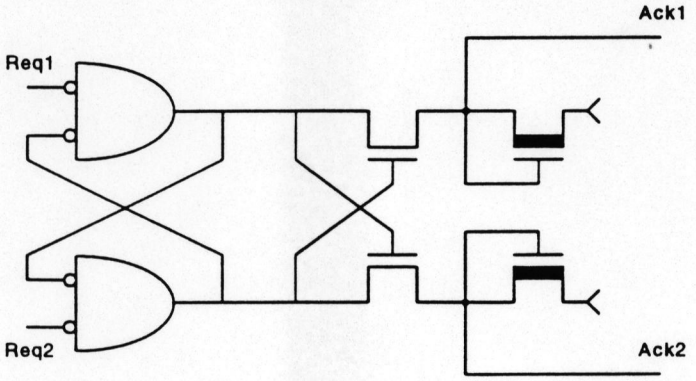

Figure 2: NMOS Mutual Exclusion Element

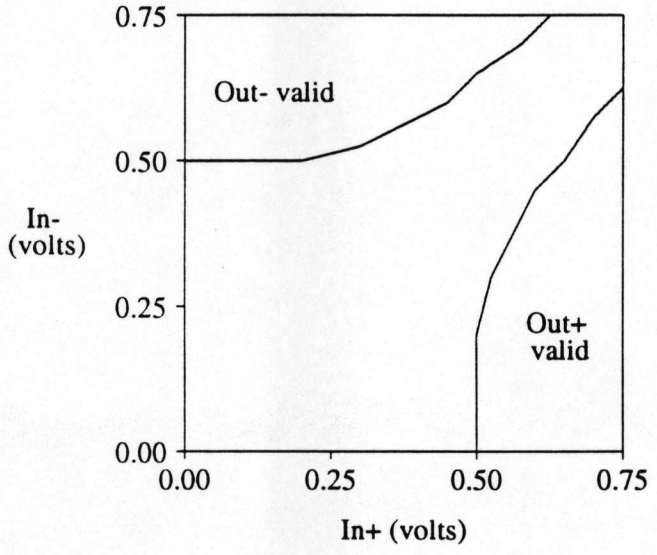

Figure 3: DC Transfer Characteristics of Threshold Detector

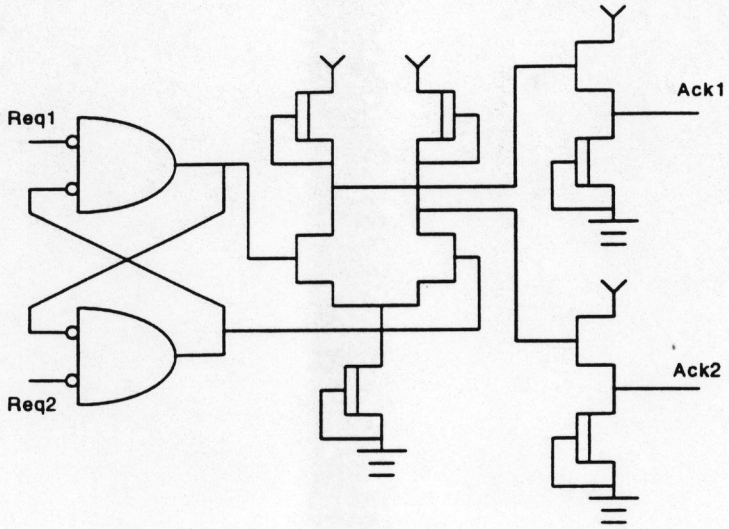

Figure 4: GaAs Mutual Exclusion Element

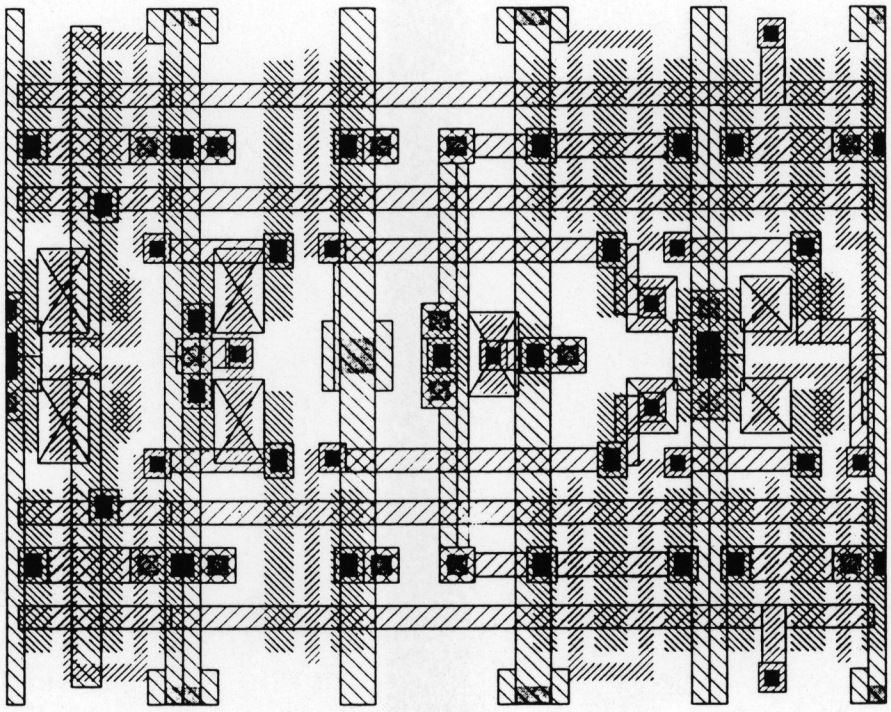

Figure 5: Composite Layout of GaAs Mutual Exclusion Element

the two signals are nearly coincident, the mutual exclusion element should take longer to decide which output to raise as the input latch resolves in favor of one or the other input.

We have been able to verify that this is in fact the case, as shown in Figure 7. This figure shows the output *Out1* from Figure 6 in response to different settings of the control voltage *Vctl*. When the delay is long, the mutual exclusion element chooses output *Out1* quickly and exclusively as shown in the top trace. As the delay is lessened, the input signals arrive closer in time and the mutual exclusion element takes longer to decide which output to assert. Notice in the bottom trace that the output transition has moved to the right by around 300pS. Also notice that in response to the closely separated inputs, the mutual exclusion element is choosing the other output much of the time and thus *Out1* is not asserted on every cycle. The increase in delay of the output signals is variable, as indicated by the widening of the rising edge of *Out1*, but the average increase is about 300pS, which is close to what our simulations predicted. The increase in the delay of the bottom trace is due only to an increase in the resolving time since *Out1* is determined by the reference delay and not the variable delay.

5 Conclusions

We have designed a gallium arsenide mutual exclusion element and demonstrated that the fabricated circuit operates correctly. In response to well spaced inputs, it quickly asserts the corresponding output. When the input signals arrive very close together in time, the mutual exclusion element may take a little longer to decide which output to raise, but whichever it chooses, it will never raise both outputs simultaneously.

The mutual exclusion element is a key component for building asynchronous circuits, but it is only one piece. We are currently extending our existing GaAs cell set [11] to support the design of high-performance asynchronous as well as synchronous circuits.

References

[1] T. J. Chaney and C. E. Molnar, "Anomalous behavior of synchronizer and arbiter circuits," *IEEE Transactions on Computers*, vol. C-22, pp. 421–422, April 1973.

[2] T. J. Chaney, "A comprehensive bibliography on synchronizers and arbiters," Tech. Rep. 306B, Washington University, Institute for Biomedical Computing, Oct 1985.

[3] C. L. Seitz, "Ideas about arbiters," *Lambda*, pp. 10–14, First Quarter 1980.

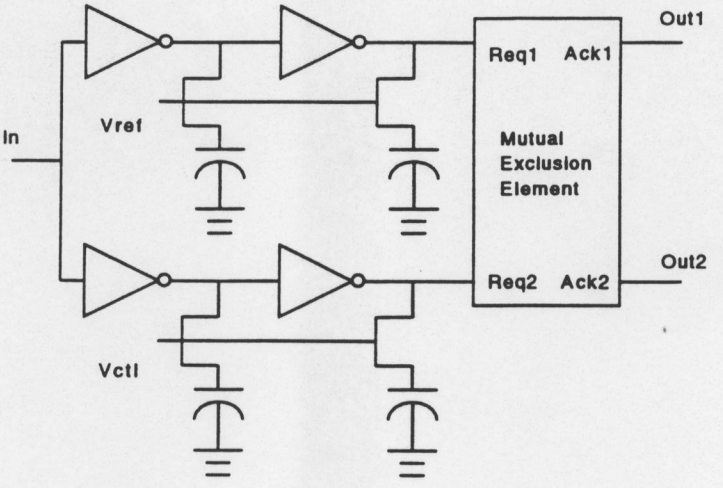

Figure 6: Test Setup

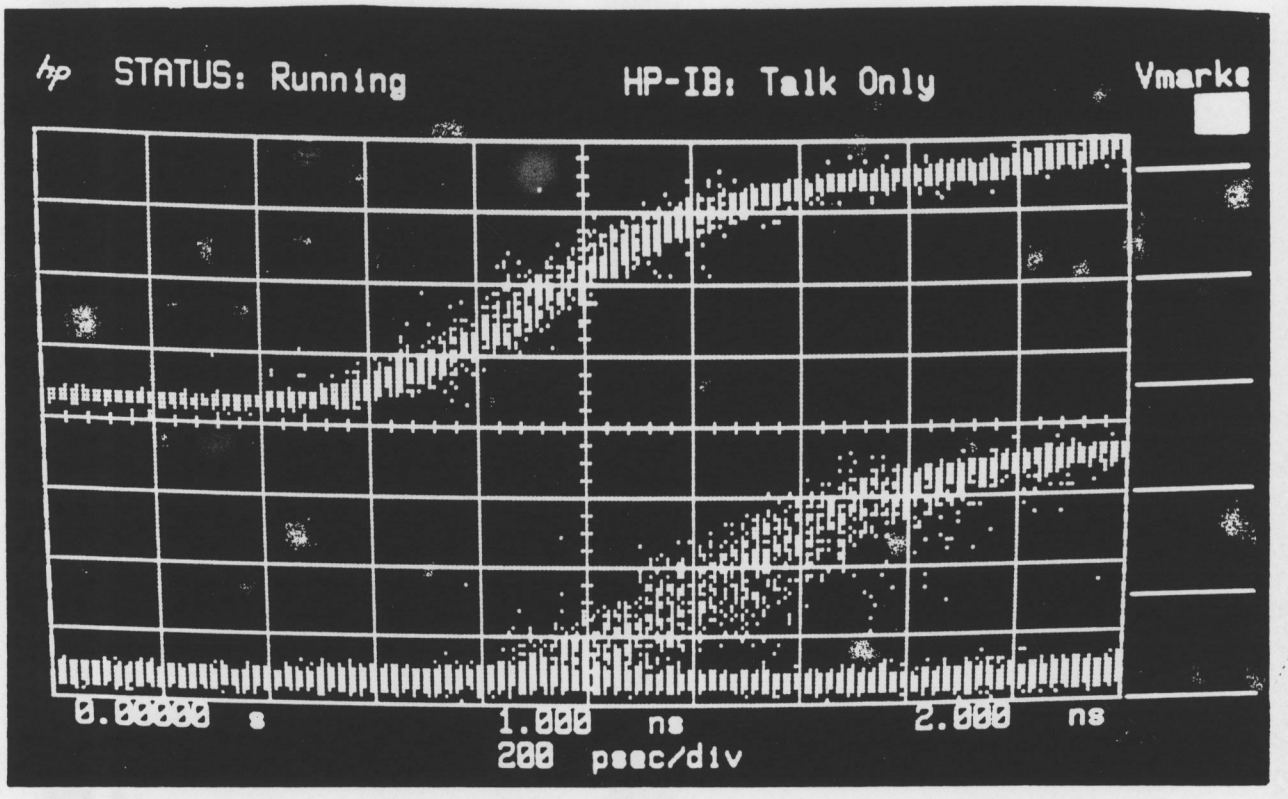

Figure 7: Oscilloscope Traces

[4] C. L. Seitz, "System timing," in *Mead and Conway, Introduction to VLSI Systems*, ch. 7, Addison-Wesley, 1980.

[5] A. J. Martin, "Compiling communicating processes into delay insensitive circuits," *Distributed Computing*, vol. 1, no. 3, 1986.

[6] E. Brunvand, *Translating Concurrent Communicating Programs into Asynchronous Circuits*. PhD thesis, Carnegie Mellon University, 1991. Available as Technical Report CMU-CS-91-198.

[7] I. Sutherland, "Micropipelines," *CACM*, vol. 32, no. 6, 1989.

[8] S. Long and M. Sundaram, "Noise margin limitations on gallium arsenide VLSI," *IEEE J. Solid State Circuits*, Aug. 1988.

[9] J. Gu and K. F. Smith, "A structured approach for VLSI circuit design," *Computer*, November 1989.

[10] Bonneville Microelectronics, Inc., Salt Lake City, Utah, *Path-Programmable Logic: A Design System for Integrated Circuits*, 1990.

[11] N. Michell, "GaAs PPL cell set," Technical Report UUCS–93-027, University of Utah, 1991.