LABORATORY FOR
COMPUTER SCIENCE

MASSACHUSETTS
INSTITUTE OF
TECHNOLOGY

MIT/LCS/TM-220

CIRCUIT ANALYSIS OF SELF-TIMED ELEMENTS
FOR
NMOS VLSI SYSTEMS

Tam-Anh Chu

May 1982

545 TECHNOLOGY SQUARE, CAMBRIDGE, MASSACHUSETTS 02139

MIT/LCS/TM-220

CIRCUIT ANALYSIS OF SELF-TIMED ELEMENTS
FOR
NMOS VLSI SYSTEMS

Tam-Anh Chu

May 1982

Circuit Analysis of Self-timed Elements

For NMOS VLSI Systems[1,2]

by

Tam-Anh Chu

May, 1982

[1] This project was supported by the National Science Foundation under research grant 9715255-MCS.

[2] This technical memorandum is a summarized version of Chapters 2 and 3 of the author's Master's thesis "Circuit Analysis of Self-Timed Elements for a VLSI Router Module", submitted to the department of EECS in May 1981.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Laboratory for Computer Science

Cambridge Massachusetts 02139

Abstract

Scaling of VLSI digital systems introduces new problems to the design of synchronous systems, due to the disproportional increase in wire delays with the decrease in transistor sizes. On the other hand, the asynchronous self-timed design approach, which has been traditionally less attractive, offer a number of advantages for VLSI. Also, this approach can be directly incorporated into a structured design methodology for Packet Communication Architectures. This paper considers a practical self-timed design methodology and studies its implementation in nMOS. The C-element and the arbiter circuit, two main circuit components of self-timed systems, are analyzed to allow the evaluation of the design approach.

Key Words : Self-timed, dual-rail code, reset signaling, C-element, metastable, arbiter circuit.

1.Introduction

The emergence of the VLSI technology, allowing the integration of systems of more than ten thousand transistors on a single chip has initiated an avalanche of new applications, the most visible of which is the one-chip microcomputers. Currently, the dominant VLSI technology is one using the nMOS process, whose main component is the n-channel MOS field-effect transistors. Up to date, VLSI systems have mostly been designed as synchronous ones, which can be systematically partitioned into the data path and control logic sections. The control circuits run off a clock whose rate is limited by the maximum delay of the combinational logic paths between registers. The data path receives clock and control signals from the control section to carry out the desired sequence of operations. Utilization of the well disciplined synchronous methodology for VLSI system design has proven to be very successful, so long as the mapping from logic circuit diagrams to physical circuits on the silicon surface goes through a step-by step translation process that takes into account actual electrical behavior of the underlying technology.

Currently, efforts are made to reduce even further the size of the basic active devices so that larger systems can be integrated on chips. This process is called scaling. Since the early 70's, the MOS transistors have been scaled down by a factor of two every two years or so. Even though this exponential trend is expected to decelerate in the near future, it is projected that by the late 80's, VLSI chips containing more than a million active devices are feasible. However, as the transistors are scaled down, a set of new problems, unimportant in large transistors, emerges. They are related to the fundamental physical characteristics of the device: the subthreshold leakage current is no longer negligible; the effects of short and narrow channel devices become important; and more importantly, the disproportional increase of interconnect delays with circuit delays, making it more and more difficult for abstracting digital systems away from the electrical components and their characteristics. As discussed in [10], interconnect wires have distributed resistances and capacitances, and the transmission of signals on wires is administered by a process called diffusion. As wires are scaled down by a factor of α, the delay in transmitting a signal scales up quadratically, i.e., by α^2. Scaling bears serious consequences on the distribution of global clock and control signals of synchronous digital systems : since systems contain more circuits, global signals have to reach more places and also travel longer distances (relative to transistor size), hence more idle time must be allow for signal skews; in addition, since wire delays increase quadratically, the clock rate has to be reduced proportionally, resulting in unacceptably inefficient operation of the system.

The existence of these new problems make the synchronous system design approach less favorable as a candidate for VLSI design. On the other hand, asynchronous self-timed design methodology provides a solution to the above problems. The key idea of this methodology is *localization of communication*. A self-timed system is composed of a large number of atomic entities called self-timed modules, interconnected in such a way that one communicate only with its neighboring modules, through a signaling convention enforcing the correct sequence of operation of modules. In such a system, control is distributed locally, and no global signals exist. The delay of the system now reflects the average delays of constituent modules, instead of the worst case delay of some modules. Moreover, the signaling convention used ensures that inter-module communication is insensitive to wire delays, therefore, alleviates substantially the problems due to timing prevailing in synchronous systems. Thus, one can expect self-timed methodology to become an important approach for VLSI system design, as it allows the design of systems that are insensitive to interconnect delays.

From another point of view, self-timed methodology offers a definite advantage. In packet-switching architectures [4], there exists a high degree of similarity between the high level description of such architecture and the description of self-timed hardware systems. For example, the data flow graph representation of an asynchronous programming language [5] exhibits parallelism, and distributed control and communication of functions or operators. This fact permits a direct mapping from the high-level language to the hardware circuits. A structured design methodology has been suggested in [7], in which self-timed system design is approached top-down, from a high level hardware description language, through several steps of refinement and finally to the actual circuit realization.

This paper describes the analyses of nMOS circuits used to implement self-timed modules. The electrical characteristics and behaviors of basic self-timed circuit elements are investigated, allowing the evaluation of a practical self-timed design methodology.

As will be presented later, a set of self-timed modules can be implemented using basic digital building blocks such as the and,or gates, etc...,as demonstrated in [9]. There are two important basic elements - the C-element and the arbiter circuit - which deserve careful and thorough understanding. The C-element is indispensable as it enforces the correct timing operation of self-timed modules, while the arbiter circuit is difficult to design because of the existence of the metastable state [1,12] in its operation.

The first part of the paper presents a self-timed design methodology and its characteristics. A set of self-timed modules is then examined and shown informally to function correctly according to the methodology. Next, the C-element is studied, as an effort to validate its functionality in self-timed modules. And lastly, the arbiter circuit and its metastable operation are considered. A theory of metastable operation of bistable circuits is developed in order to provide an understanding of the circuit behavior and therefore, permitting the optimization of design parameters for the arbiter circuit.

2.A self-timed methodology

2.1.Characteristics

The correct operation of a self-timed system requires that constituent modules comply to a number of system constraints. Self-timed modules are atomic entities and can not be decomposed. A self-timed system consists of a number of modules interconnected according to some rules, necessary to prevent deadlocks. Self-timed modules have the following characteristics :

2.1.1.Input/Output Specifications of Self-timed Modules

The core of a self-timed module is a combinational logic (C/L) circuit. The timing restrictions and relations between the inputs and outputs of a C/L circuit are expressed through the weak conditions [1-1]. In essence, they simply stipulate that output changes must follow and not precede input changes, and input changes can not occur until previous output changes have settled. Figure 2.1 shows these specifications.

2.1.2.Signaling convention and the dual rail code

The reset signaling convention, usually referred to as handshake protocol, implements inter-module communication. The protocol forms a closed loop communication path between modules through the use of request/acknowledge signal pairs to make a successful transfer of data. The assertion of the request signal indicates that the inputs to a module are all defined, while the assertion of the acknowledge signal indicates that the outputs of the module are all defined, and therefore, inputs are allowed to change. Figure 2.2 depicts the reset signaling protocol, which is composed of the active and the reset phase. During the active phase, data are defined, in the reset phase, data are reset to the undefined (or spacer) state.

In implementing the modules, the dual rail code is employed to represent a signal. A dual-rail coded signal consists of two signal lines, thus is able of expressing four distinct states. The spacer state (00) indicates the absence of data, the zero- and one- data states (01 and 10) represent the data value of 0 and 1, respectively, and finally, the illegal state (11) corresponds to an illegal transition of data lines, therefore, allowing the inclusion of some fault-detection mechanism in the modules. The reset signaling protocol permits only transitions between spacer and data states, but not transitions between data states themselves, as shown in Figure 2.3.

Using the dual rail code, the request signal is no longer explicitly needed, because it is embedded in the input data. This property is very important as it makes the transmission of data insensitive to wire delays. If an explicit request signal were used, the time skews between the request signal and data could cause incorrect interpretation by the receiving module, and hence correct operation would depend on careful matching of wire delays. Figure 2.4 shows a self-timed module and its timing specification, using the dual rail coded signals.

2.1.3.Equipotential regions

An equipotential region is one in which a signal can be treated as identical everywhere, that is, the signal requires a negligible amount of time to equalize all potential differences within the designated region. This notion is fundamental in any self-timed methodology. A basic assumption in the synthesis of self-timed modules is that in a module, wire delays are negligible, whereas delays in logic gates are arbitrary but finite[6]. This is equivalent to stipulating that self-timed modules have to reside completely within equipotential regions. In any integrated circuit technology, limits of such regions can be defined, based on the electrical characteristics of interconnects and circuits. Particularly, in nMOS technology, equipotential regions are defined as regions within which signals settle in less than τ, the transit time of a MOS transistor[8]. As stated in [11], normally, these limits are much larger than the size of self-timed modules, and hence, no special care is required.

The notion of equipotential region also brings up another interesting and important point : self-timed modules can be considered to be contained in equipotential regions, communicating with each other reliably through the use of the reset signaling protocol. Thus, the protocol must be implemented whenever signals are to be transmitted between regions, inside an equipotential region, this requirement is not neccessary.

2.1.4.Basic elements used in the design of Self-timed modules

Basic elements needed to implement self-timed modules include the common logic gates (and, or,...), the C-element and the arbiter circuit. The design of a set of self-timed modules [7] can be constructed exclusively from these elements, as will be presented in the following section. Figure 2.5 shows the C-element, its logic realization and transition table. It is implemented as a simple asynchronous state machine by feeding the output of the majority circuit back to one of its inputs. Its output goes high when all its inputs goes high, low when all its inputs goes low. Figure 2.6 shows the

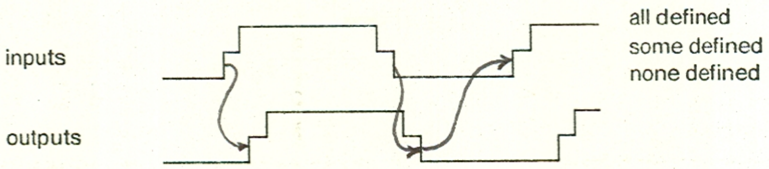


Figure 2.1.The weak conditions
on inputs/outputs relation of C/L

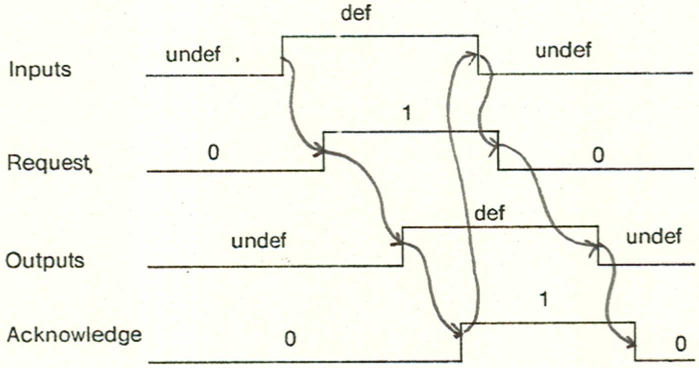


Figure 2.2. Reset Signaling protocol for self-timed modules

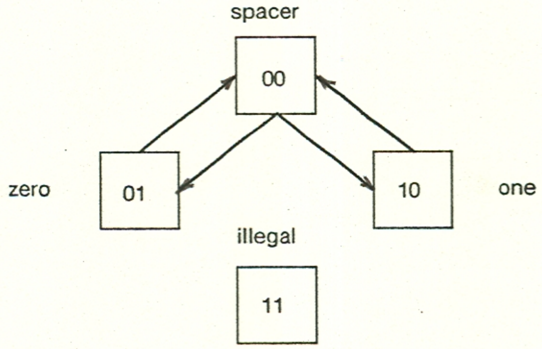


Figure 2.3.The dual rail code representation of signal

in the Reset signaling protocol

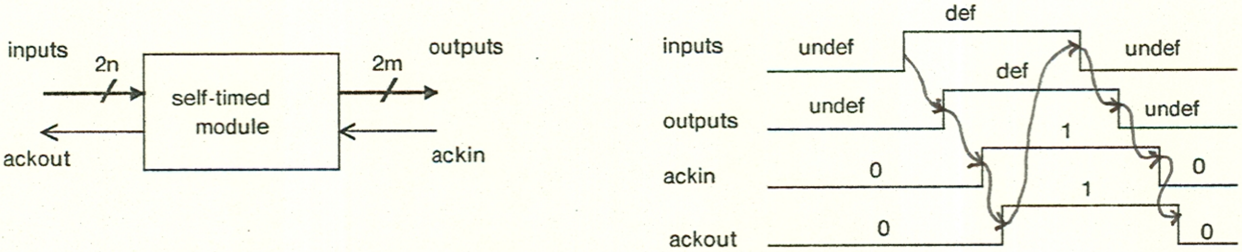
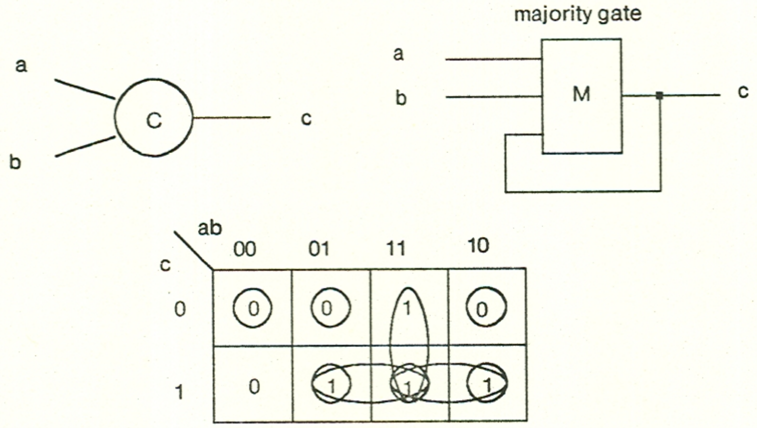


Figure 2.4.Self-timed module and its port signals

Inputs and outputs are dual-rail coded

Port signals obey reset signaling protocol arbiter circuit, consisting of a front-end set-reset flip-flop and a threshold detecting circuit. The threshold circuit is needed to prevent the illegal voltage level associated with the metastable state from appearing at the output. The operation of the arbiter circuit will be discussed more in a later section.



c \ ab	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Figure 2.5. The C-element, its realization and transition table

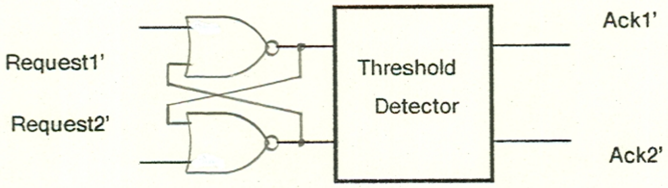


Figure 2.6. Logic diagram of the Arbiter circuit

3. The Self-timed modules

A set of self-timed modules has been designed [7], their implementation in nMOS using the Mead and Conway's design rules has been demonstrated [9]. The electrical and timing characteristics of nMOS self-timed modules using ratioed logic has been verified by SPICE in [3]. In the following, the logic diagrams of self-timed modules are presented with their behavioral descriptions. Note that all the above characteristics of self-timed methodology are included in every module, namely :

1. Dual rail coded data are used in the signaling protocol.

2. Within a module, C-elements are use to enforce the weak conditions mentioned in Section 2.

3. Modules are small enough to reside completely in equipotential regions.

4. Common nMOS ratioed logic gates are used.

3.1.AND module

The AND (combinational self-timed modules are capitalized) module has two input ports and one output port (each has two signal lines), as in Figure 3.1. Other combinational modules such as OR, NAND, NOR can be obtained by inverting the port signals according to DeMorgan's theorem. The acknowledge signals are conveyed using the C-element.

3.2.Switch module

The switch module has one data and one control input port. In Figure 3.2, the heavy directed arcs represent signal bundles, large circles are replicated gates. Data are switched to either the T (true) or the F (false) output port, depending on the value of the control input.

3.3.Multiplexor module

The multiplexor module has two data ports, one control input ports and one data output port. In Figure 3.3, the control input connects either the T or F data input port to the data output port.

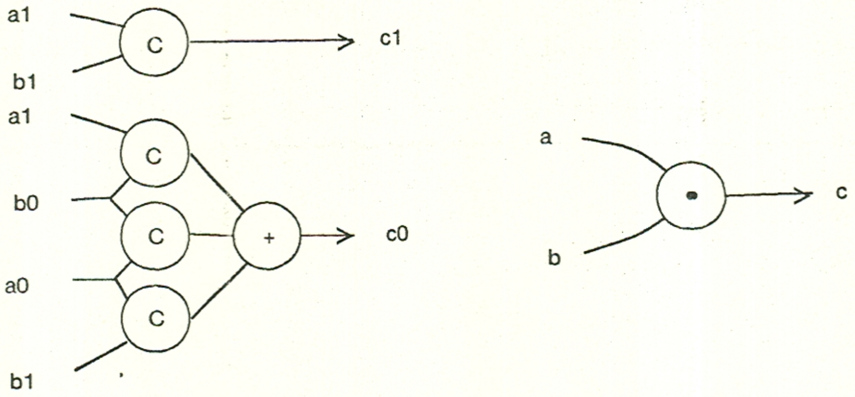


Figure 3.1.AND module symbol and implementation

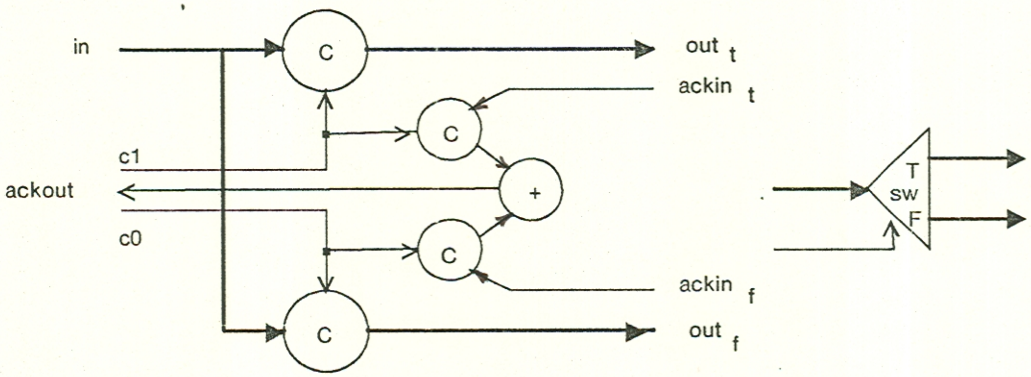


Figure 3.2. Switch module symbol and impıementation

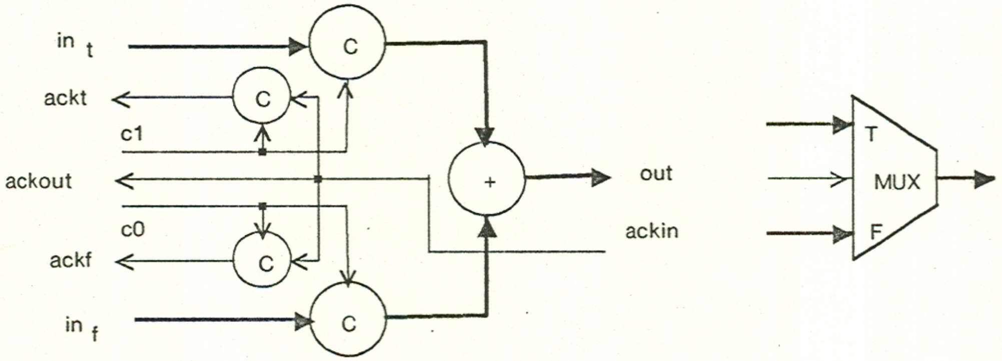


Figure 3.3. Merge module symbol and implementation

Notes:

1.Heavy lines indicate dual-rail coded signal bundles

2.Large circles indicate replicated elements

3.4.Sink module

The sink module is also called the packet detector. It accepts input data and generates the acknowledge signal to the source. As shown in Figure 3.4, it contains a Data detector and a Spacer detector, used for detecting the data and spacer state of the input, respectively. Two methods of implementing the data detector is shown in Figure 3.5, using exclusive-or gates or or-gates. The spacer detector is a multiple input or-gate.

3.5.Source module

The source module generates a constant value. When acknowledged, it is reset to the spacer state. Figure 3.6 shows the 0- and 1-source.

3.6.Register module

The register module stores input data in C-elements and acknowledges the source immediately after the reception of data. They are reset to spacer state when the following module accepts the data and acknowledges. Because of this, registers are connected in pairs, one is in data while the other is in spacer state. A feedback path must contains at least three register modules.Figure 3.7 shows the implementation of the register module.

3.7.Merge module

The merge module shown in Figure 3.8 accepts two inputs, passes the one arriving first to the output port and blocks the other. In case they arrive concurrently, the arbiter resolves by arbitrarily passing one through. The nor- and and-gates are needed to keep the arbiter engaged until both the input and acknowledge have been reset.

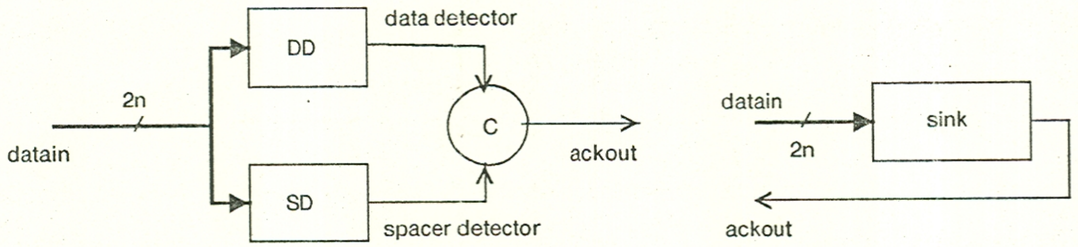
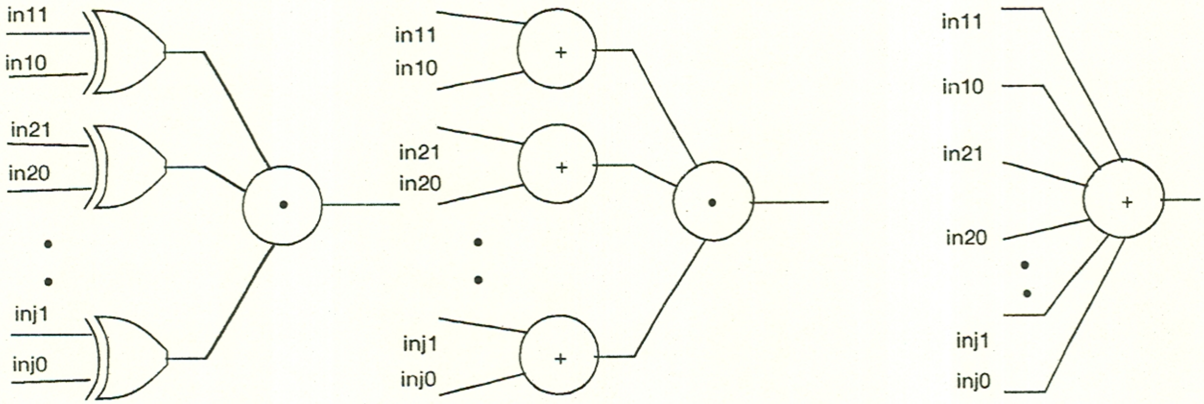


Figure 3.4. Sink module symbol and implementation

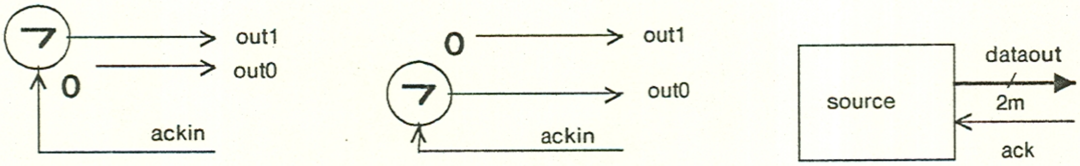


DD using ex-or gates

DD using or gates

SD using multiple input or gate

Figure 3.5. Implementations of Data and Spacer Detector



0-source

1-source

Figure 3.6. Source module symbol and implementations

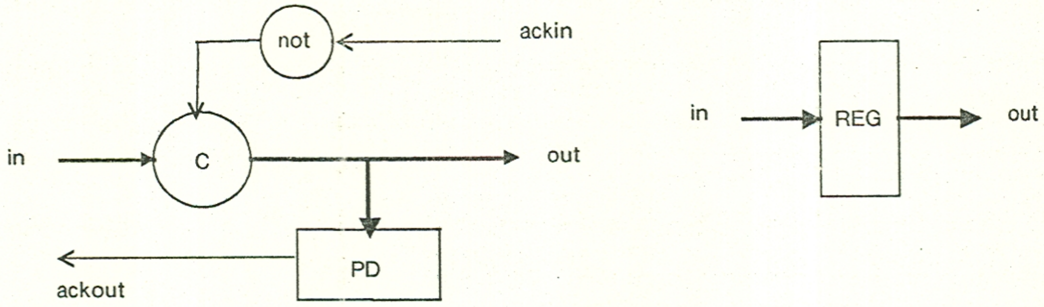


Figure 3.7. Register module symbol and implementation

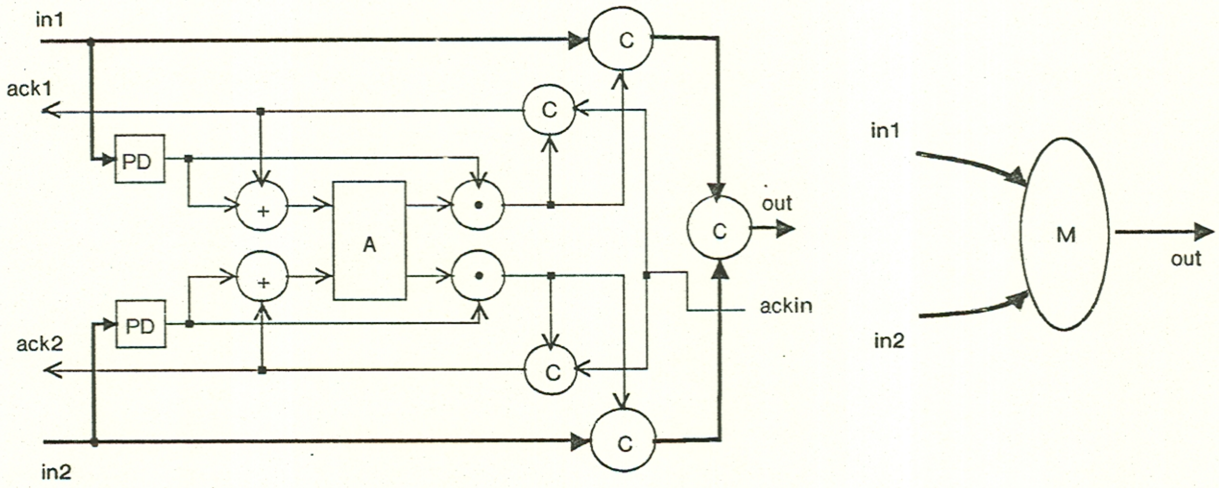


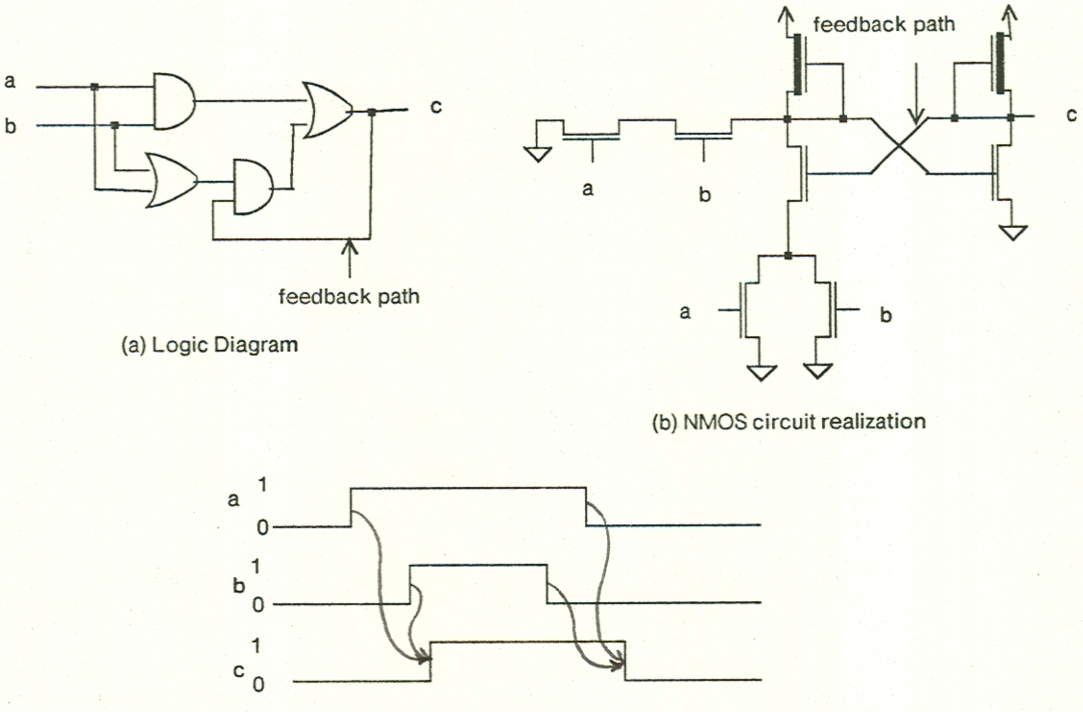
Figure 3.8. Merge module symbol and implemenantation

4. Analysis of the C-element

4.1. Description and function

The C-element is one of the mostly used elements in the implementation of self-timed modules. In Figure 4.1, its logic diagram, nMOS circuit design and timing diagram are shown. As mentioned earlier, the C-element is a simple asynchronous state machine, whose output is the only state of the circuit. From the transition table, the output changes to 1 only when both inputs change to 1, and to 0 when both inputs changes to 0, in case where two inputs assume different values, the output remains in its previous state. The logic diagram is derived from the transistor circuit of the C-element, the feedback path can be detected in both circuits.

The main function of the C-element is to synchronize the signals. As shown in Figure 4.1c, it is the device that enforces the *weak conditions* in self-timed modules, as it waits for all changes to occur before making an output transition.



(a) Logic Diagram

(b) NMOS circuit realization

(c) Timing

Figure 4.1. The C-element

4.2.Condition for correct operation of the C-element

Consider the case as depicted in Figure 4.2. The C-element has a certain delay in the feedback path, which can be represented by a lumped delay D. Assume that this C-element is a part of a self-timed module. When both inputs a and b go high, output c goes high. If somehow, its succeeding module receives the data and then acknowledges, forcing one of the inputs a or b to go low before output c feeds back through the delay D, then output c will return to low. This scenario exemplifies a possible condition under which the C-element malfunctions, resulting in faulty operation of the self-timed module. This sequence can be seen in the transition table : initially, the circuit is in total state abc = 000, with input transition 00->11, the output becomes 1 but the feedback state variable c' is still in state 0, thus the total state is in 110. If input transition 11->10 then occurs, the total state shall be 100, instead of 101.

Thus a simple condition can be imposed on the circuit to guarantee correct operation, being that the interval between successive input changes has to be larger than the delay D of the feedback path. If input changes are slower than the feedback change, then the circuit reaches its stable total state before the next input change, hence, no malfunction occurs. Note that this condition is almost always met as the feedback path is a short wire contained within an equipotential region.

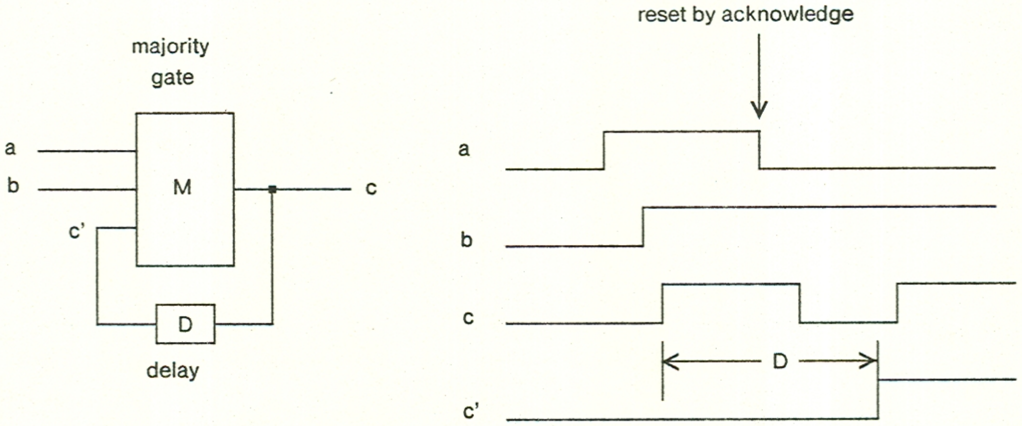


Figure 4.2. Malfunction in the C-elelment
due to feedback delay

4.3.Hazards due to unrestricted input changes

Output hazards of combinational circuits have little effect on the operation of synchronous systems, as they are allowed to settle before being latched into registers. On the opposite, hazards are intolerable in asynchronous systems, because any transition of output or state variables triggers other transitions immediately ; the circuit operates autonomously, and does not depend on any clock timing. For this reason, it is necessary to analyze circuits used in self-timed modules and define the constraints under which no hazard will ever occur. The constraints must then be followed strictly, or failure due to hazards may result. In this section, hazard conditions of the C-element is analyzed, allowing one to evaluate its functionality in self-timed design.

The transition table in Figure 2.5 shows that input transitions in the same direction do not cause output hazards. Only input changes in opposite direction, as shown in Figure 4.3, can cause hazards. These changes are the sequences 01 – 00 – 10 and 01 – 11 – 10, the first sequence causes a 0-hazard, while the second causes a 1-hazard. Note that hazards can occur if the separation between two input changes are within a critical window, outside the window, no hazards result. Let δ_1 and δ_2 be the critical windows of separations of input changes causing the 0- and 1-hazard, respectively, one can analyze two cases to determine δ_1 and δ_2.

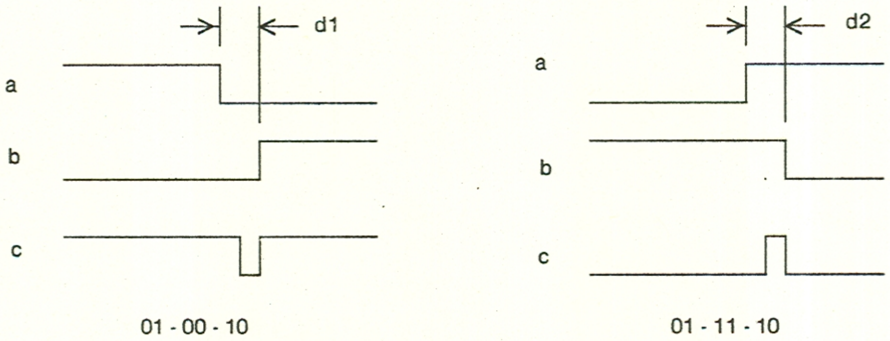


Figure 4.3. Input transitions causing hazards
d1,d2 are critical window values

4.3.1. Input Sequence 01 - 00 - 10

For this input sequence, the input and-gate of the C-element is always off, and the circuit can be modeled as in Figure 4.4, the input transitions can be replaced by a negative pulse, whose width is δ_1. In Figure 4.4, initially, output c (node 1) is high. When the pulse is applied, node 1 starts going low, while node 2 starts going high. If the pulse switches to high when V_1 is still greater than V_2, the output returns to its high value, a glitch appears at the output. If the input pulse switches high when V_1 is already less than V_2, V_1 continues to go low as V_2 goes high, thus the output settles at a new value. If the input pulse switches high when V_1 and V_2 are almost equal, the circuit can 'hang' in the metastable state, an illegal logic state corresponding to a voltage level called V_{inv}, defined as the logic threshold of the inverter. Thus, one can define the critical window δ_1 to be the pulse duration causing a metastable state output. The Appendix contains SPICE simulation results for this case. If the C-element is designed according to Mead and Conway design rules (4:1 logic ratio) and has a fan-out equal to k, δ_1 is given by

$$\delta_1 = \tau k + 9\tau$$

This equation is obtained using the zeroth order approximation, and it has been verified by SPICE simulation, and the plot of δ_1 versus fan-out k in Figure 4.6 shows that it is very close to experimental results.

4.3.2. Input Sequence 01 - 11 - 10

This input sequence can cause a 1-hazard. Similar to above, the circuit in Figure 4.5 can be used to study the C-element output hazards. Initially, output c is low, depending on the input pulse width δ_2, the output may settle to a new value, have glitches, or enter the metastable state, as in Figure 4.5. The critical window in this case is given by

$$\delta_2 = \frac{\tau}{n} k + \tau$$

where n is a proportionality constant. SPICE simulation gives n~8. Figure 4.6 shows the plot of δ_2 versus fan-out.

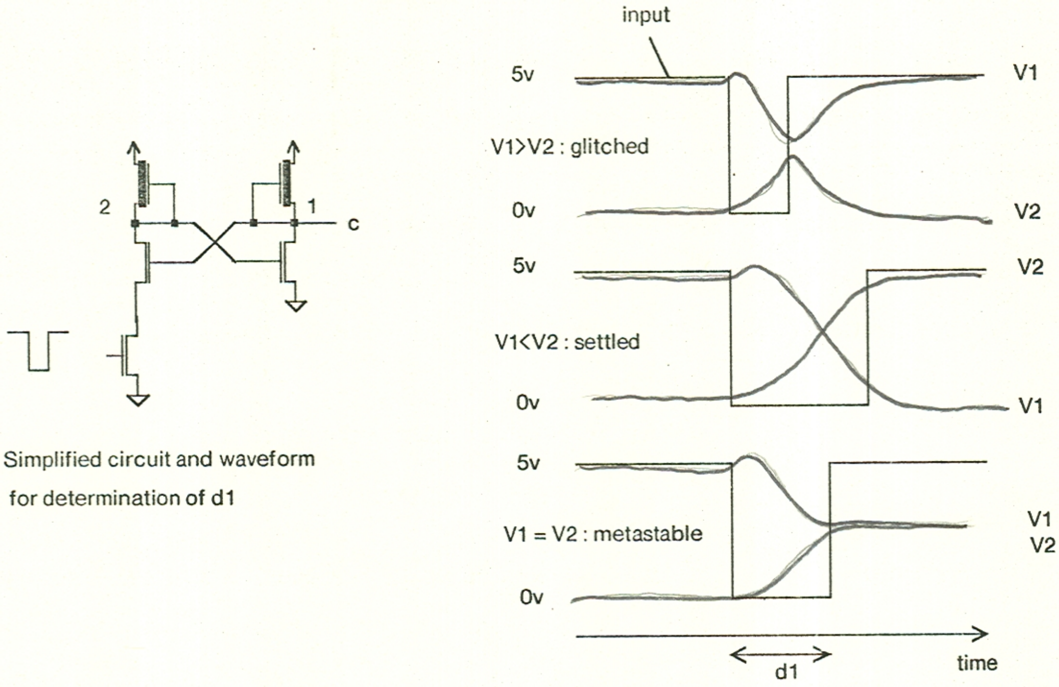


Figure 4.4. Simplified circuit and waveform for determination of d1

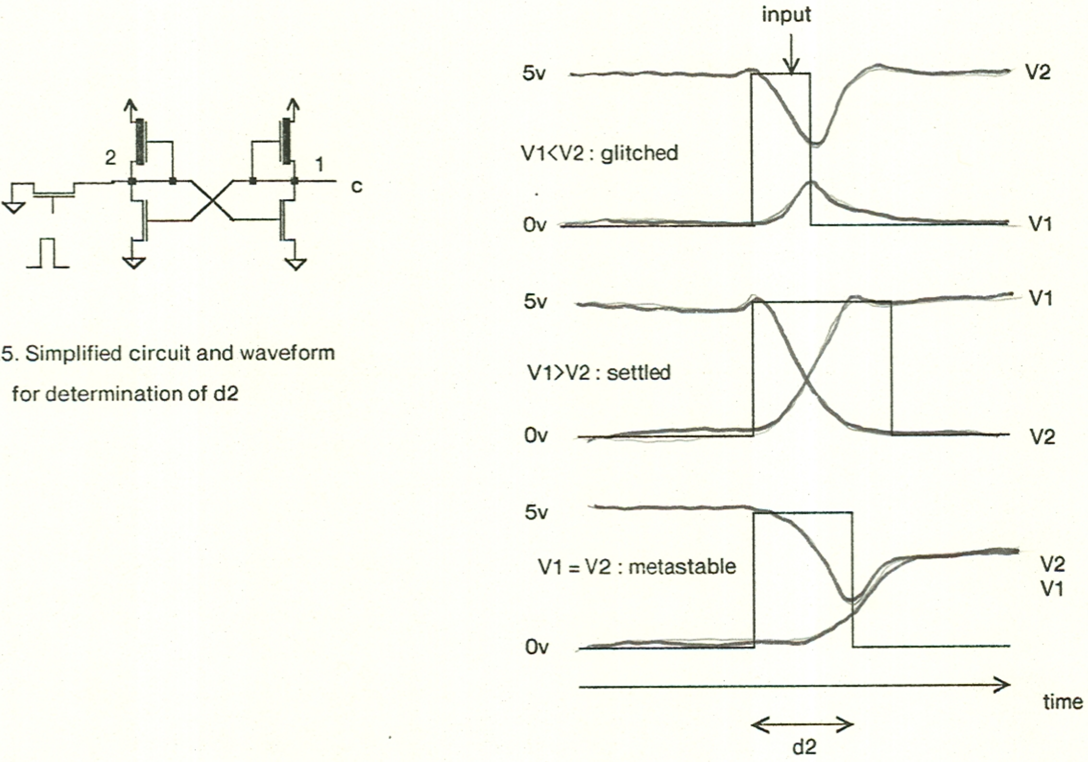


Figure 4.5. Simplified circuit and waveform for determination of d2

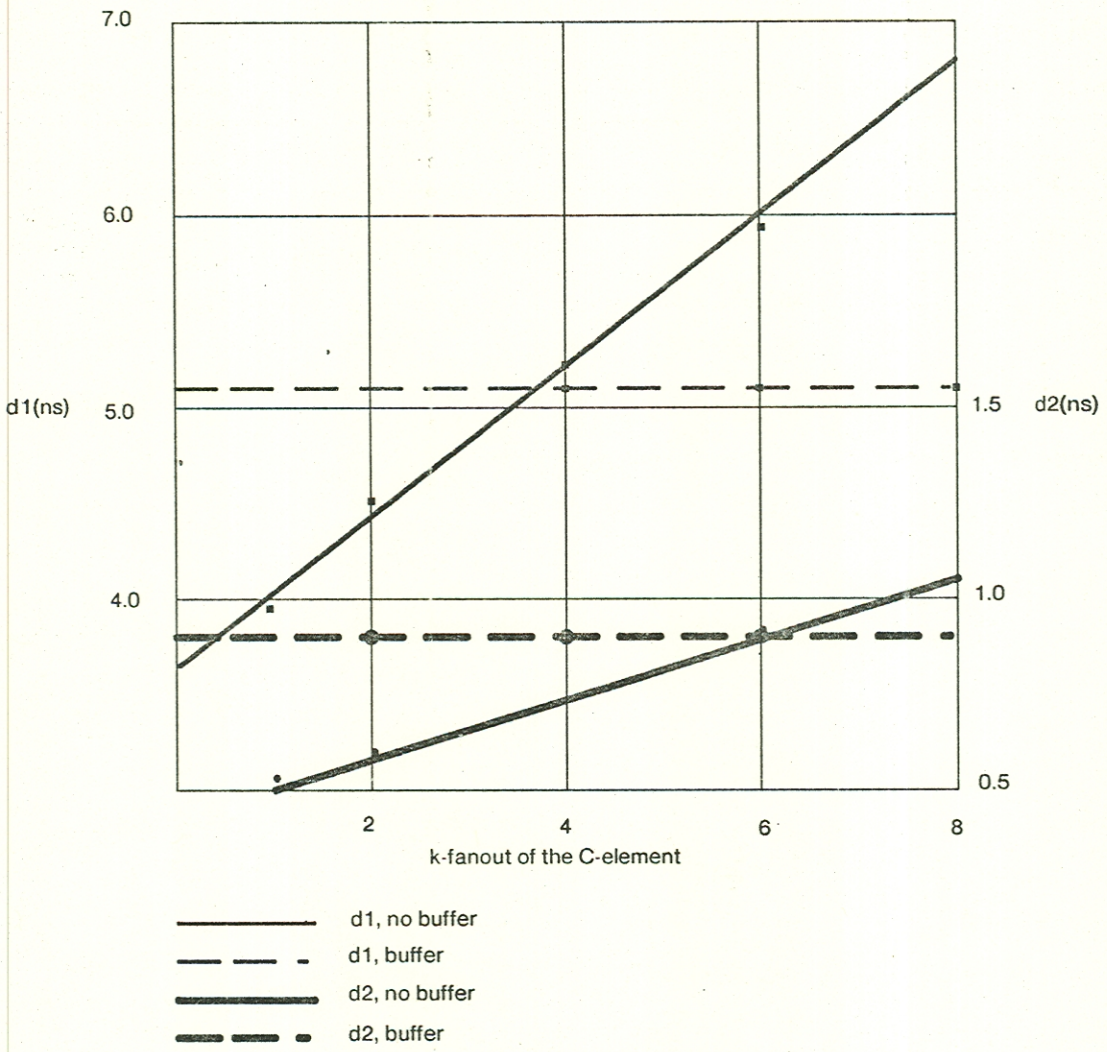


Figure 4.6.Plot of d1, d2 as function of fanout k

4.4.Constraints on input changes

From the above analysis, it is clear that the separation between input changes in *opposite direction* should not be less than the critical windows δ_1 and δ_2. Other input changes do not cause hazards and therefore are permitted. Within the context of self-timed system, it is noted that C-elements are used to synchronize a group of signals which always make the same transition (either from low to high or from high to low) in the same phase of the reset signaling cycle. Thus neighboring transitions are usually in the same direction. If the active and reset phases are far apart, one can be sure that signal transitions in *opposite directions* do not occur close to each other. This argument indicates that the above constraints on input changes can be easily met, or almost always met if the set of modules discussed in section 3 are used exclusively. The reason is that except the register module, all of the rest return acknowledge signals after delays much larger than δ_1 and δ_2, therefore, the reset phase is sufficiently apart from the active phase of a reset signaling cycle.

4.5.A buffered C-element

As seen previously, the critical window values are proportional to the fan-out or output loading of the C-element. Thus, it may be desirable in some cases, where the loading is significant, to used a C-element with buffered output. Figure 4.7 shows its nMOS circuit diagram. The output buffer is a configuration called superbuffer, providing almost symmetrical rise time and fall time for the output, its rise time is approximately four times faster than the regular C-element's. The buffer isolates the internal structure and the feed back path of the C-element from output load, thus keeping the values of δ_1 and δ_2 constant, as shown in Figure 4.6.

The disadvantages are it requires more space (layout size increases by 7.4%) and dissipates more static power (70% more). Thus the use of buffered C-element is recommended when the output load is large, such as in the case of large fan-outs or long interconnects.

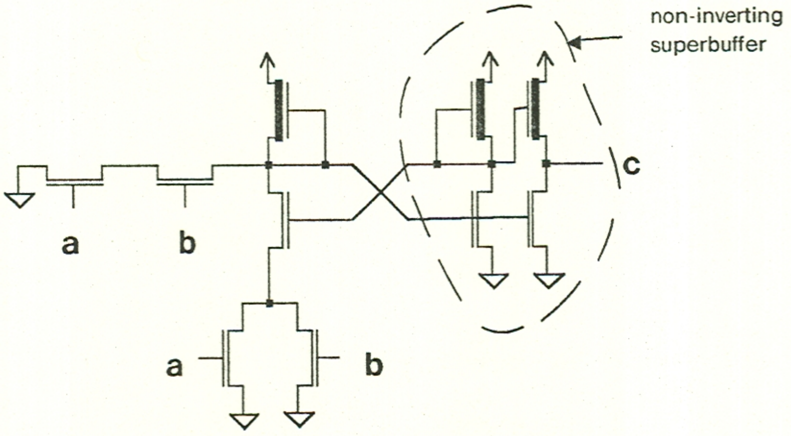


Figure 4.7. Buffered C-element

5. Analysis of the Arbiter Circuit

The arbiter circuit is needed to arbitrate between two asynchronous requests. As shown in Figure 2.6, it is comprised of a set-reset flip-flop and a threshold detector circuit. The flip-flop accepts asserted-low request inputs, and it performs interlocking function by locking out the other input when one input is passed to the output. As has been well known, the flip-flop is a bistable device and has a meta-stable state, at which the input and output of the gates are at the same voltage level called the logic threshold of the inverter. In this state, the circuit is stable, in the sense that the voltage levels are sustained by the gates feeding back to one another. However, the presence of a small perturbation will cause the flip-flop to come out of the metastable state due to the regenerative action of the cross-coupled devices.

When two requests arrive at almost the same time, the flip-flop may enter the metastable state. The threshold detector functions as a voltage comparator, its outputs are asserted only when the inputs are at different voltage levels, thus the voltage levels associated with the metastable state are prevented from appearing at the outputs.

To have a better picture of the metastable state behavior of the flip-flop, consider the case when two inputs are asserted (low) at almost the same time, the outputs then are equal to the logic threshold V_{inv}. In Figure 5.1, the nor gates can be replaced by inverters, with outputs V_1 and V_2 both equal toV_{inv}. In this initial state, the bistable device can be studied through small signal models with parameters evaluated at $V = V_{inv}$.

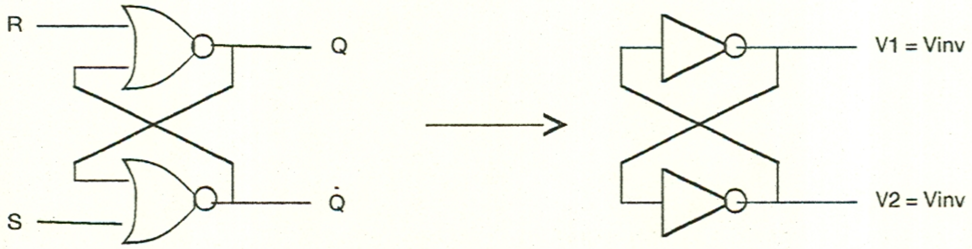


Figure 5.1. A bistable device in the metastable state

5.1.NMOS transistor and Inverter models

In this section, models for nMOS transistor and inverter are briefly explained. More complete discussion can be found in [3]. Figure 5.2 shows the dynamic small signal model for the n channel MOS device. The device contains two dependent current sources with transconductances g_m and g_{mb} between the gate and the source, and the bulk and the source, respectively. The drain to source conductance is g_d. There are four principal capacitances from the gate and bulk to the drain and the source, being C_{gd}, C_{gs} and C_{bd}, C_{bs}, respectively. These capacitances are composed of intrinsic and extrinsic components, the latter arise from the overlap regions of the gate over the source and the drain. Figure 5.2 also shows the physical regions of the nMOS structure where these capacitances manifest themselves. Note that except for the extrinsic parts of the capacitances, all parameters are dependent on the region of operation of the device, being different in the nonsaturation and saturation region. The bulk terminal is commonly known as 'back gate' because its effect on the channel current is the same as that of the gate but at a lesser degree.

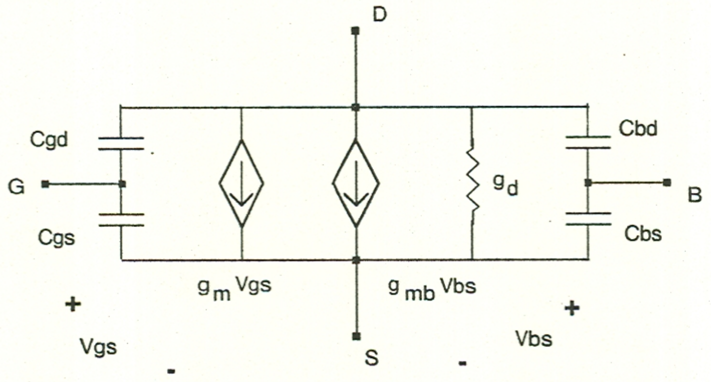


Figure 5.2a. Small signal model of n-channel MOSFET

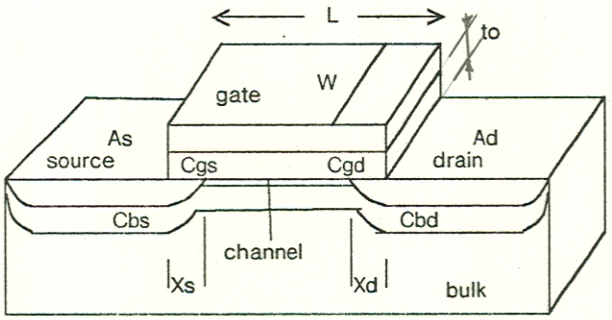


Figure 5.2b.Physical region of the MOS capacitances

The inverter model used in the bistable device is depicted in Figure 5.3. Some parameters disappear because they are short circuited (capacitors) or open-circuited (dependent sources). Capacitance C_{gdpd} of the pull down device is divided in two, using Miller's theorem. In Figure 5.3, the components are defined as follows (subscripts pd and pu denote pullup and pulldown device)

$$C_{in} = C_{gspd} + C_{gdpd}(1 - A_0)$$

$$C_t = C_{bdpd} + C_{gdpd}(1 - \frac{1}{A_0}) + C_{gdpu} + C_{bspu}$$

$$A_0 = \text{Gain of the inverter at } V = V_{inv}$$

$$= - \frac{g_{mpd}}{g_d} = - \frac{2K_{pd}(V_{inv} - V_{TE})}{K_{pu}(V_{inv} - V_{TD} - V_{DD})}$$

$$g_d = \text{conductance of the pull up device at } V = V_{inv}$$

$$= K_{pu}(V_{inv} - V_{TD} - V_{DD})$$

where

$$K_{pu} = \frac{\mu\varepsilon_0}{2t_0}(\frac{W}{L})_{pu}$$

V_{TD} = threshold voltage of the pull up depletion transistor

and

μ = electron surface mobility

ε_0 = permittivity of Silicon dioxide

W,L = width and length of transistor channel region

t_0 = thickness of the oxide layer

Also, since it is necessary to determine the transient behavior of the bistable, the circuit model in Figure 5.3 has a DC supply voltage Vdd and a dependent current source given by

$$I(V) = K_{pd}(V - V_{TE})^2$$

where

$$K_{pd} = \frac{\mu\varepsilon_0}{2t_0}(\frac{W}{L})_{pd}$$

V_{TE} = threshold voltage of the pull down enhancement transistor

Note that the value of current source is given for saturation region operation because at $V = V_{inv}$, the pull down transistor is in saturation.

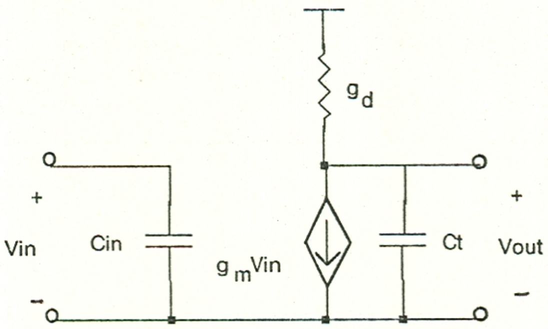


Figure 5.3. Inveter circuit model at V = Vinv

5.2.Metastable region operation of the bistable circuit

The circuit model of the bistable can be constructed using the above inverter model. In Figure 5.4, the bistable circuit model is shown. Generally, the inverters can be assumed to have identical parameters, since they are laid out symmetrically close to each other. In Figure 5.4,

$$C_{tot} = \text{total capacitance seen by node 1 or 2}$$
$$= C_t + C_{in} + C_{wire}$$

$$C_{wire} = \text{capacitance of the feedback interconnects}$$

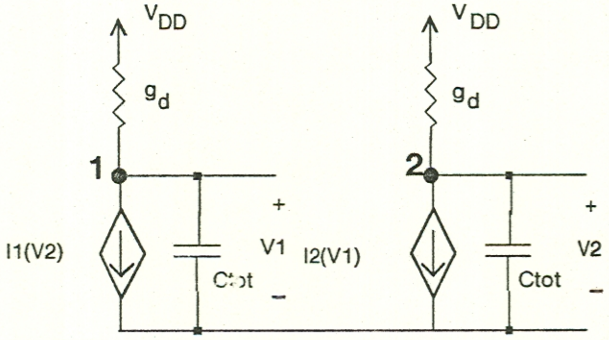


Figure 5.4. Circuit model of bistable device at V = Vinv

Over the range of interest, that is, around V_{inv}, the following assumptions are valid :

1. g_d is almost constant

2. the sum of V_1 and V_2 is approximately constant, and

$$V_1 + V_2 \sim 2V_{inv}$$

Initially, the bistable is in metastable state, a small perturbation ΔV causes V_1 and V_2 to exit the metastable state, thus

$$V_1(0) = V_{inv} - \Delta V$$

$$V_2(0) = V_{inv} + \Delta V$$

Using Kirchoff's current law at nodes 1 and 2 gives

$$(V_{DD} - V_1)g_d = K_{pd}(V_2 - V_{TE})^2 + C_{tot}\frac{dV_1}{dt} \quad (1)$$

$$(V_{DD} - V_2)g_d = K_{pd}(V_1 - V_{TE})^2 + C_{tot}\frac{dV_2}{dt} \quad (2)$$

Letting $V = V_1 - V_2$, subtracting (2) from (1), and eliminating the squared terms, assuming $V_1 + V_2 \sim 2V_{inv}$, one obtains

$$[2K_{pd}(V_{inv} - V_{TE}) - g_d]V - C_t\frac{dV}{dt} = 0$$

or

$$\frac{dV}{dt} - \frac{1}{RC_{tot}}V = 0$$

where

$$R = \frac{1}{2K_{pd}(V_{inv} - V_{TE}) - g_d} = \frac{1}{g_d(A_0 - 1)}$$

$$= \frac{k}{K_{pd}}\frac{1}{2k(V_{inv} - V_{TE}) - (V_{inv} - V_{DD} - V_{TD})}$$

$$k = \frac{(L/W)_{pu}}{(L/W)_{pd}}$$

so

$$\tau_r = RC_{tot} = \frac{C_{tot}}{g_d(A_0 - 1)}$$

This equation shows that V_1 and V_2 exit the metastable state exponentially with time constant τ_r, that is

$$V_1(t) = V_{inv} - \Delta V \exp(\frac{t}{\tau_r})$$

$$V_2(t) = V_{inv} + \Delta V \exp(\frac{t}{\tau_r})$$

Figure 5.5 shows output voltages V_1 and V_2, which have been verified by SPICE simulation in the Appendix.

Also, the probability of *remaining* in the metastable state at time t', as determined in [2], is given by

$$F(t') = C_1 \tau_r \exp(\frac{-t}{\tau_r})$$

Where C_1 is a proportionality constant. Thus, τ_r characterizes the electrical and probabilistic behavior of the bistable in the metastable state.

5.3. Parameters optimization

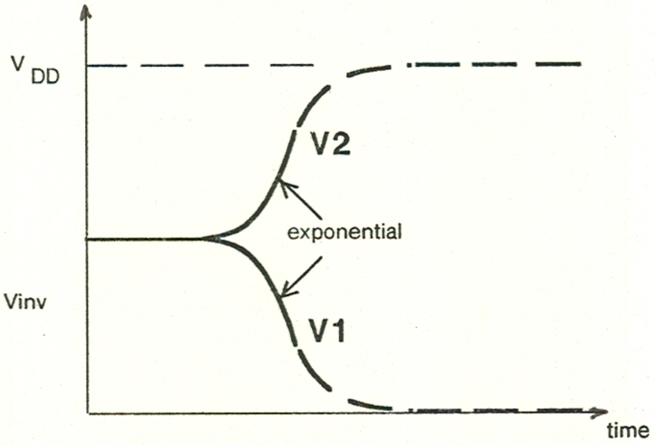


Figure 5.5. Output voltages when bistable exits the metastable

From the above analysis, it is seen that an optimal design for the arbiter circuit must have the smallest value possible for τ_r. With τ_r minimized, V_1 and V_2 exit the metastable at their maximum rates, and the probability F(t') is also minimized.

Usually, the gain $A_0 \gg 1$, therefore the ratio k is much greater than 1, so R can be approximated by

$$R \sim \frac{1}{2K_{pd}} \frac{1}{V_{inv} - V_{TE}}$$

$$\sim \frac{L^2}{\mu C_{gspd}} \frac{1}{V_{inv} - V_{TE}}$$

therefore,

$$\tau_r \sim \frac{1}{\mu(V_{inv} - V_{TE})} L^2 \frac{C_{tot}}{C_{gspd}}$$

In order to minimize τ_r, one has to minimize

1. the channel length L
2. the capacitance ratio $\frac{C_{tot}}{C_{gspd}}$.

Since the total capacitance C_{tot} is approximately

$$C_{tot} = C_{gspd} + C_{stray}$$

or

$$\frac{C_{tot}}{C_{gspd}} = 1 + \frac{C_{stray}}{C_{gspd}}$$

Therefore, minimizing the capacitance ratio requires maximizing the gate capacitance C_{gspd}.

Thus, an optimal design of the bistable contains inverters (or nor-gates) with large pulldown transistor. Since the channel length has to be minimized, the pull down transistor is wide, therefore it has large channel width and minimum channel length. Figure 5.5b shows a layout of such inverter using Mead and Conway's design rules. Note that the pull up is kept small so that k is large and g_d is insignificant, according to the previous approximations. The large geometry also has the additional advantage of being less sensitive to layout misregistration.

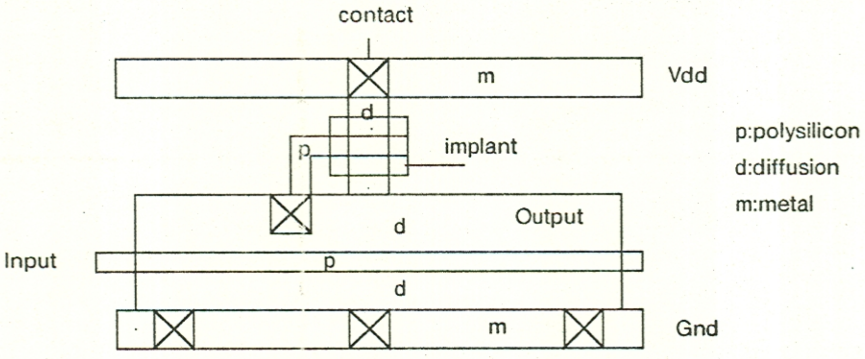


Figure 5.6. Layout of an Inverter
5

5.4.Design considerations

In designing an operational nMOS arbiter circuit, there are a number of practical aspects one has to take into consideration. This section investigates those aspects relevant to the operation of the arbiter circuit.

5.4.1.Threshold detectors

The threshold detector is required at the output of the flip-flop to prevent the illegal voltage level associated with the metastable state from appearing at the output of the circuit. Two commonly used designs are shown in Figure 5.6, together with their transfer characteristics. The first design employs two cross-coupled nMOS enhancement transistors to compare the voltages V_1 and V_2. If the voltage difference is greater than the threshold voltage V_{TE}, then one of the transistors will be turned on, grounding the input to a low voltage, and the circuit produces asserted-low outputs. The second design uses inverters with hysteresis to suppress the metastable output. In its transfer characteristic, it is important that the metastable voltage V_{inv} be covered by the hysteresis loop, as shown in Figure 5.6. Of the two design, the first one is preferred because it can be more easily implemented and it works for any value of the metastable voltage, as long as their difference is less than the threshold of the transistor. If the outputs of the flip-flop oscillate in metastable state, the oscillations are in-phase, and thus, will be suppressed.

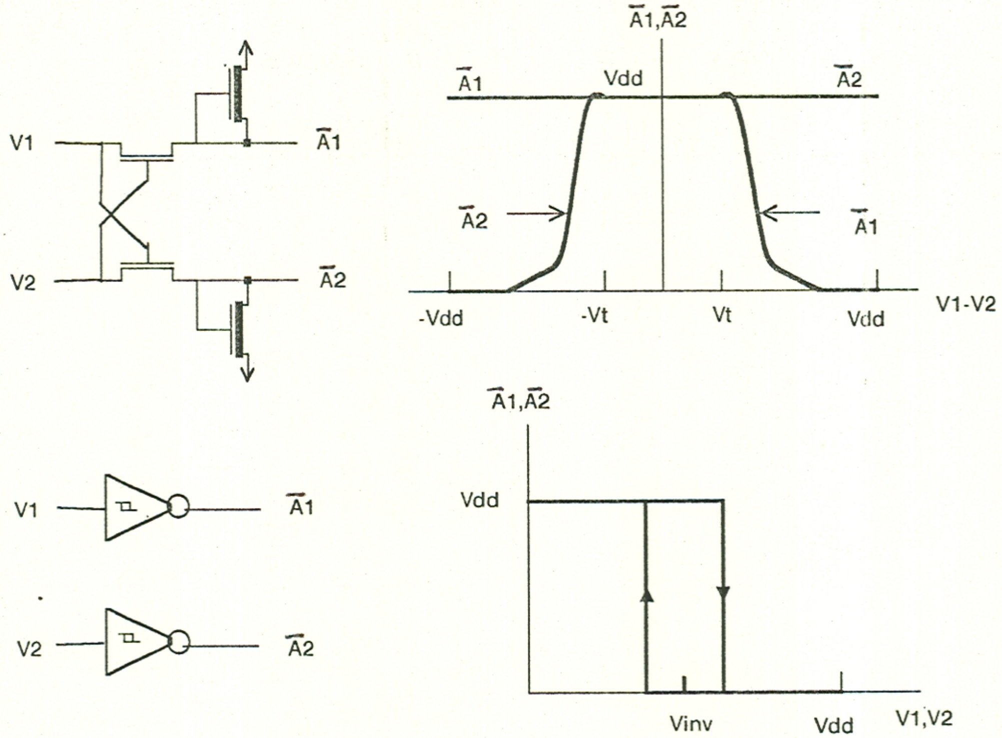


Figure 5.6. Designs of the threshold detector

5.4.2. Mismatch in output capacitances

As discussed in [3], if the outputs of the flip-flop see different capacitances due to mismatch in layout and wire capacitances, V_1 and V_2 will still exit the metastable state exponentially, but with different time constants τ_{r1} and τ_{r2}. The voltage difference between V_1 and V_2 still increases monotonically, therefore, does not affect the operation of the threshold detector. Operation of the circuit in this case has been correctly verified by SPICE simulations (see the Appendix).

5.4.3. Oscillation

The above analysis of the arbiter shows that there is no oscillation in the metastable state. SPICE simulations are performed for three cases : in the first, the feedback interconnects are assumed to have no delays; in the second, small RC delays are incorporated into the feedback paths; in the third, noninverting buffers are inserted. Figure 5.7 shows the circuits and the metastable outputs for these cases. The results indicate that oscillation occur in the third case, and both outputs oscillate in-phase until they exit the metastable state.

In considering the metastable operation of the circuit, each nor-gate can be modeled as a linear system with a single pole. The first case can not oscillate as each gate provides only a 90 degree phase shift. The second case may oscillate if the gain of the gates are large enough, since each gate and its output RC delay can provide 180 degree phase shift. In the third case, the series of nor gate and buffer can be considered as a 3-pole system, and since the gain is very large (equal to the gain product of 3 stages), the output can oscillate easily. At the frequency of oscillation, each branch (nor gate and buffer included) provides 180 degree phase shift. Thus, the feedback to the input of a nor-gate is positive, causing oscillation.

5.4.4.Geometry of Layout

The logic threshold V_{inv} is dependent on the geometry of the circuit. If the nor-gates have grossly different geometries, their V_{inv}'s might differ by more than the threshold voltage V_{TE} of the threshold detector transistors. In this case, the detector may not function correctly.

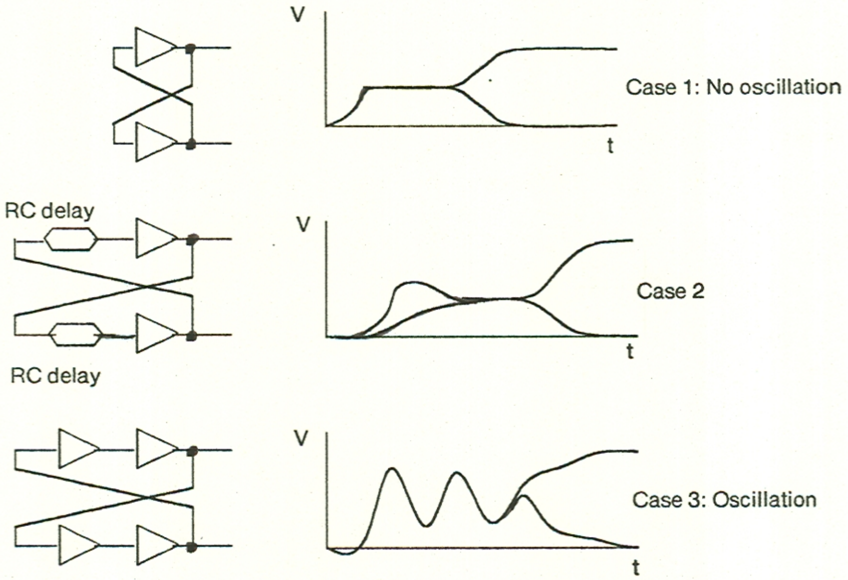


Figure 5.7. Simulation results for different cases of feedback connection

Hence for correct operation of the arbiter circuit, it is required that

$$|V_{inv1} - V_{inv2}| < V_{TE}$$

The logic threshold V_{inv} can be approximated by

$$V_{inv} = V_{TE} - V_{TD}\left(\frac{Z_{pd}}{Z_{pu}}\right)^{1/2}$$

where

Z = length to width ratio of transistor channel = L/W

Let $\Delta(P)$ be the maximum absolute variation in P, and assume that

$$\Delta L = \Delta W = \Delta D$$

then, as shown in [3],

$$\Delta Z = \frac{\Delta D}{W}\left(1 + \frac{L}{W}\right)$$

and

$$\Delta Z_{pd} < \Delta Z_{pdmax} = \frac{V_{TE}}{|V_{TD}|}\,(Z_{pu}Z_{pd})^{1/2}\;\frac{1}{1 + \dfrac{Z_{pd}W_{pd}(1+Z_{pu})}{Z_{pu}W_{pu}(1+Z_{pd})}}$$

$$\Delta Z_{pu} < \Delta Z_{pumax} = \frac{V_{TE}}{|V_{TD}|}\,(Z_{pu}Z_{pd})^{1/2}\;\frac{Z_{pu}/Z_{pd}}{1 + \dfrac{Z_{pu}W_{pu}(1+Z_{pd})}{Z_{pd}W_{pd}(1+Z_{pu})}}$$

For example, for a design of the arbiter with

Pull up : $W = 4\lambda \quad L = 2\lambda$

Pull down : $W = 32\lambda \quad L = 2\lambda$

and $V_{TE} = 1.0$ v, $V_{TD} = -4.0$ v

The above formulas give

$\Delta Z_{pumax} = 0.207$, so $\Delta D < 0.552\lambda$

$\Delta Z_{pdmax} = 0.018$, so $\Delta D < 0.542\lambda$

The overall requirement for the geometry of the arbiter circuit is that the maximum variation in width and length must be less than about 0.54λ.

5.4.5. A layout for the nMOS Arbiter

The design of the arbiter in Figure 5.9 has been optimized. The arbiter circuit with device sizes are given. A layout of the arbiter circuit has been produced, and the circuit has been fabricated through a class project. Initial testing shows that the arbiter works correctly, even though further testing and evaluation are required to actually quantify its operation.

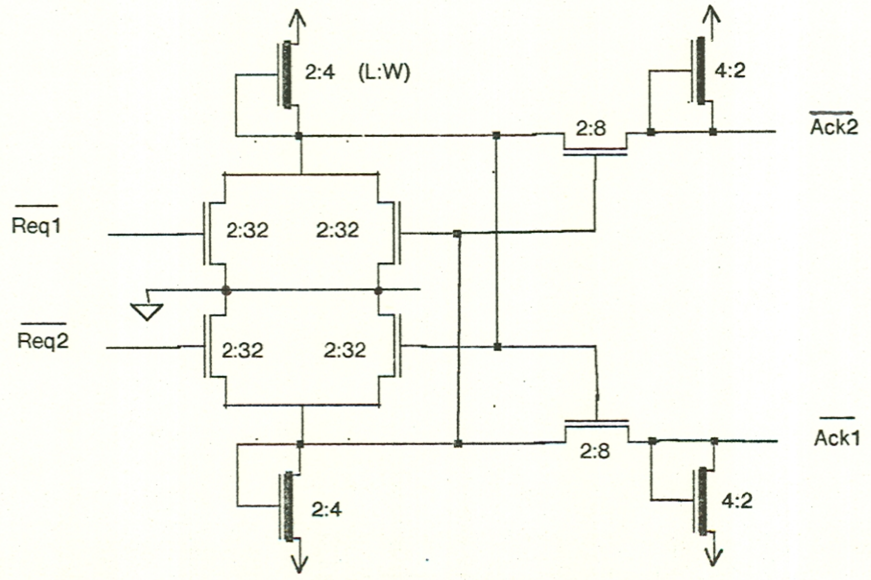


Figure 5.9. NMOS design of the Arbiter circuit

6.Conclusion

The analysis of self-timed elements in nMOS has shown that a practical self-timed system can be realized. Analysis of the C-element reveals that it can produce glitches if consecutive input changes in *different directions* are within a critical window, defined as the separation of two input changes causing the C-element to enter the metastable state. However, it is reasoned that the signaling protocol used prohibits such input transitions, hence guaranteeing the correct operation of the C-element. The study of the arbiter circuit using the bistable model in the metastable state and other design considerations lead to an optimized design of the arbiter circuit. This design, as suggested by Seitz, proves to be effective in preventing the metastable state voltage from appearing at the desired outputs.

There are several areas left unanswered in this paper: in the area of design methodology, it would be necessary to arrive at an established theory for self-timed systems, dealing with issues such as deadlocking, signaling convention, fault modeling, etc.; in the area of system realization and implementation, one would consider the circuit topology for other hardware components needed to comprise a rich set of system configurations. Also, it would be beneficial to investigate other technologies for VLSI such as CMOS (Complementary MOS), I^2L (Integrated Injection Logic), which give low power dissipation, higher level of integration and other advantages.

7. Acknowledgments

The author would like to thank Prof. Jack Dennis for his guidance and helpful suggestions, Prof. Arvind for his supports. Clement Leung, Andy Boughton, Willie Lim and Randy Bryant have spent many hours for discussions and proof reading, their help is deeply appreciated.

References

[1] Chaney, T J and Molnar, C E, Anomalous Behavior of Synchronizer and Arbiter Circuits, IEEE Trans. on Computers, C-22, April 1973.

[2] Chaney, T J and Rosenberger, F, Characterization and Scaling of MOSFET Performance in Synchronizer Applications, Proceedings of CalTech Conference on VLSI, January 1979.

[3] Chu, T, Circuit Analysis of Self-Timed Elements for a VLSI Router Module, SM Thesis, Department of EECS, June 1981.

[4] Dennis, J B, Packet Communication Architectures, Proceedings of the 1975 Sagamore Conference on Parallel Processing, IEEE, New York 1975.

[5] Dennis, J B, First Version of a Data Flow Procedure Language, Lecture Notes in Computer Science, Volume 12, Springer Verlag 1974.

[6] Dennis, J B and Patil, S S, Speed-Independent Asynchronous Circuits , CSG Memo No. 54, Project MAC, MIT January 1971.

[7] Leung, C K C, On a Design Methodology for Packet Communication Architectures based on a Hardware Design Language, CSG Memo, LCS, MIT 1979.

[8] Mead, C and Conway, L, Introduction to VLSI Systems, Addison Wesley 1980.

[9] Ries, P,A VLSI Implementation of a Two by Two Packet Router, CSG Memo 197, LCS, MIT, July 1980.

[10] Seitz, C, Self-Timed VLSI Systems, Proceedings of CalTech Conference on VLSI, January 1979.

[11] Seitz, C, System Timing, Chapter 7 of Mead and Conway's Introduction to VLSI Systems, Addison Wesley, 1980.

[12] Wann, D F et al., A Fundamental Problem Associated with the Physical Realization of Certain Classes of Petri Nets, TR125 Computer Systems Laboratory, Washington University, St. Louis, July 1975.

Appendix

This appendix contains output of SPICE simulations.

Plate 1 : Simulations for the C-element. The separation of the inputs changing in opposite directions determines whether the output has glitches, settles to a new state, or entering the metastable state.

Plate 2 : Simulation of the arbiter circuit at normal temperature. The metastable voltage does not appear at the outputs of the threshold detector.

Plate 3 : Simulation of the arbiter circuit with different load capacitances for the set-reset flip-flop.

Plate 4 : Simulation of the arbiter circuit with buffers in the feedback connections. The flip-flop outputs oscillate in phase and finally exit the metastable region. The threshold detector is shown to prevent the oscillation from appearing at the outputs.

Plate 1

Simulation of the C-element

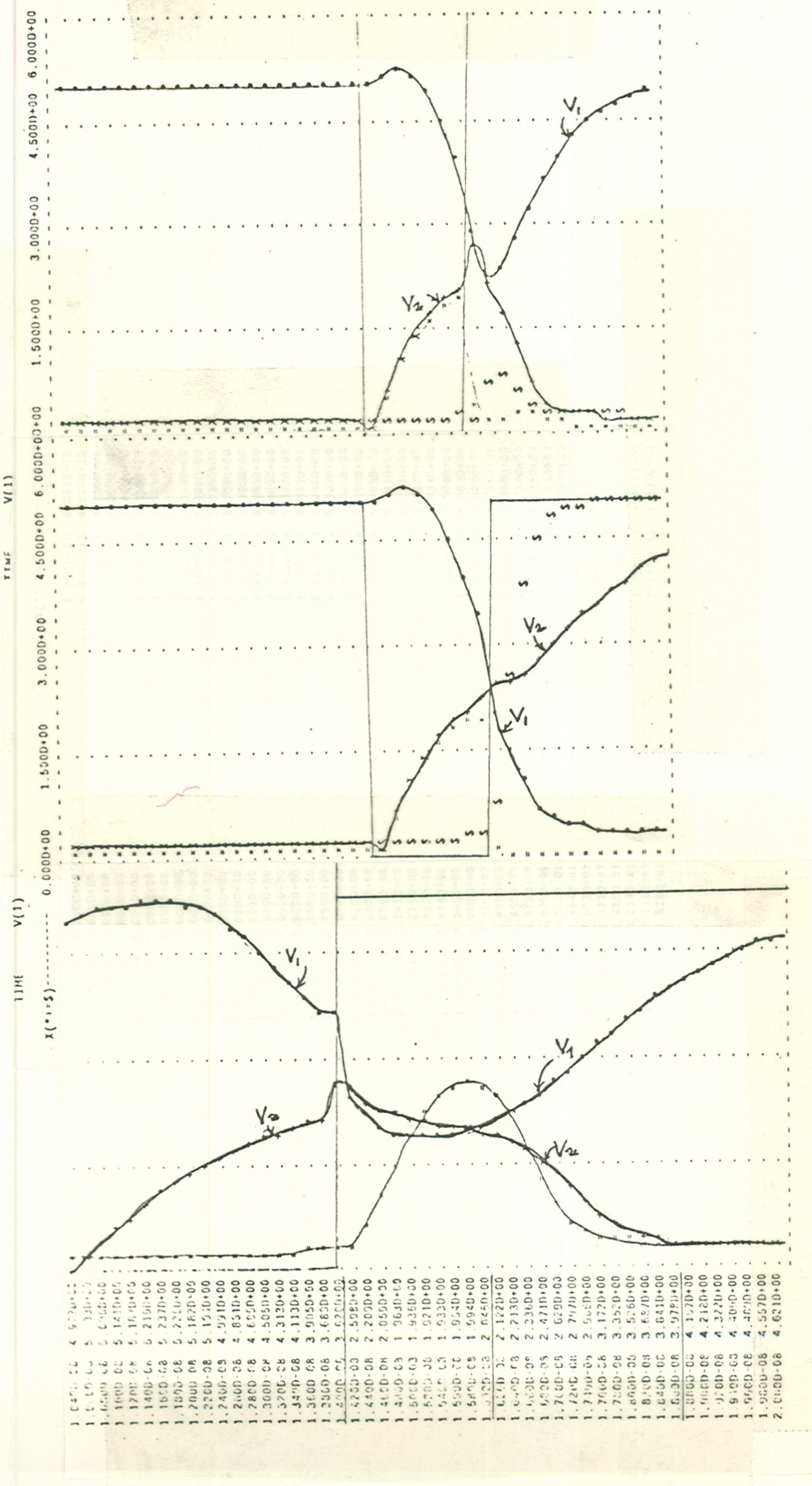
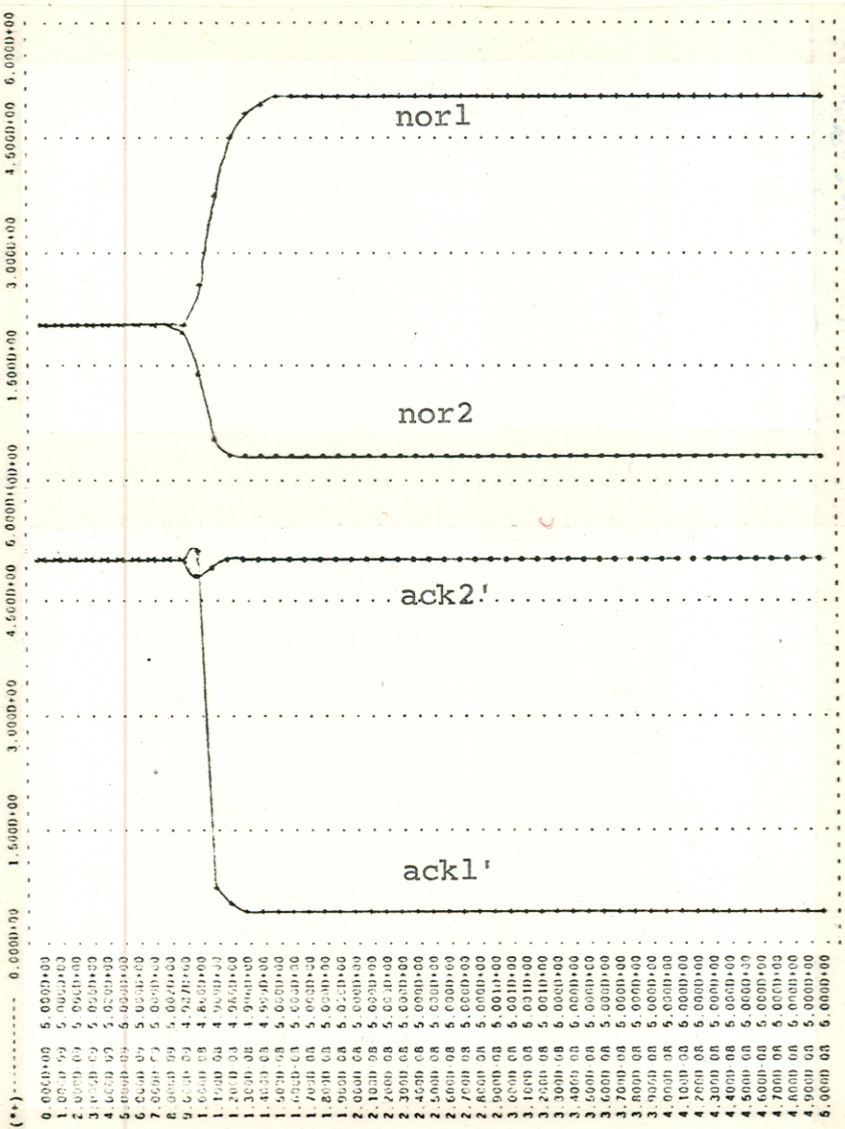


Plate 2

Simulation of the Arbiter circuit at 27°C

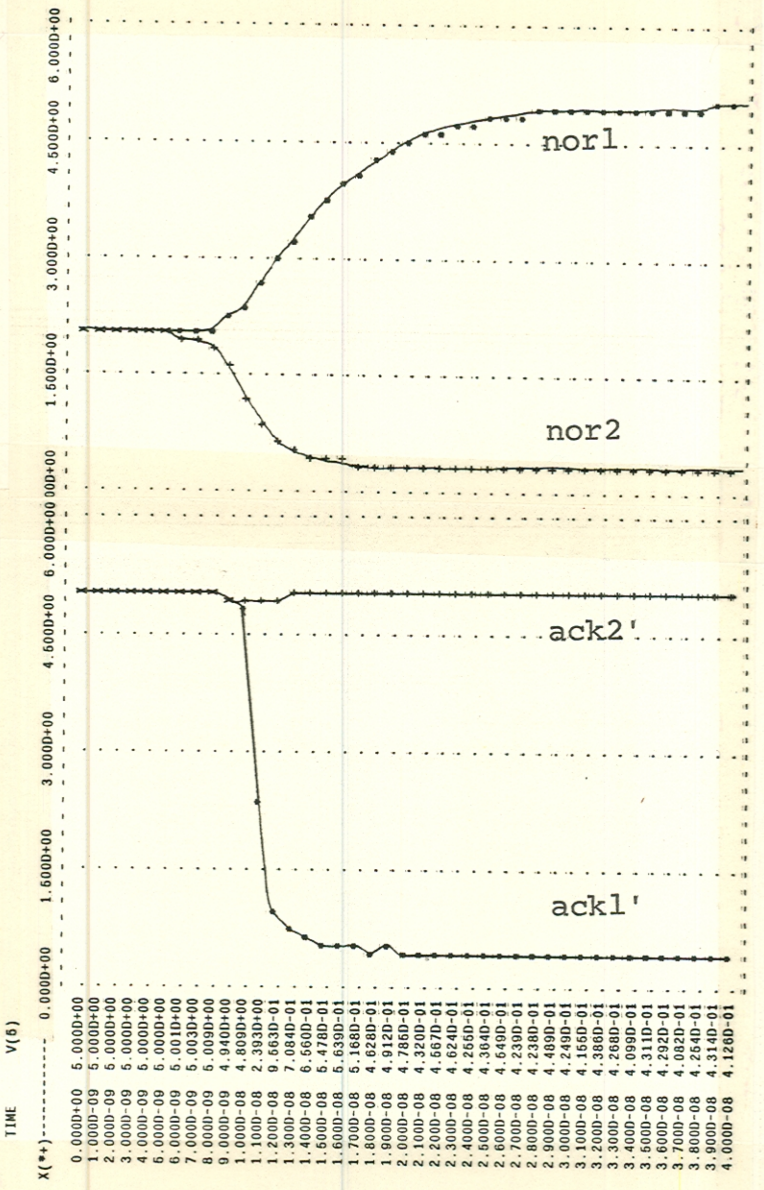


```
      TIME        V(5)

X(*+)-------------  0.000D+00

 0.000D+00   5.000D+00 .
 1.000D-09   5.000D+00 .
 2.000D-09   5.000D+00 .
 3.000D-09   5.000D+00 .
 4.000D-09   5.000D+00 .
 5.000D-09   5.000D+00 .
 6.000D-09   5.000D+00 .
 7.000D-09   5.000D+00 .
 8.000D-09   5.002D+00 .
 9.000D-09   4.927D+00 .
 1.000D-08   4.860D+00 .
 1.100D-08   4.909D+00 .
 1.200D-08   4.965D+00 .
 1.300D-08   4.996D+00 .
 1.400D-08   4.999D+00 .
 1.500D-08   5.000D+00 .
 1.600D-08   5.000D+00 .
 1.700D-08   5.000D+00 .
 1.800D-08   5.000D+00 .
 1.900D-08   5.000D+00 .
 2.000D-08   5.000D+00 .
 2.100D-08   5.000D+00 .
 2.200D-08   5.000D+00 .
 2.300D-08   5.000D+00 .
 2.400D-08   5.000D+00 .
 2.500D-08   5.000D+00 .
 2.600D-08   5.000D+00 .
 2.700D-08   5.000D+00 .
 2.800D-08   5.000D+00 .
 2.900D-08   5.001D+00 .
 3.000D-08   5.001D+00 .
 3.100D-08   5.001D+00 .
 3.200D-08   5.001D+00 .
 3.300D-08   5.000D+00 .
 3.400D-08   5.000D+00 .
 3.500D-08   5.000D+00 .
 3.600D-08   5.000D+00 .
 3.700D-08   5.000D+00 .
 3.800D-08   5.000D+00 .
 3.900D-08   5.000D+00 .
 4.000D-08   5.000D+00 .
 4.100D-08   5.000D+00 .
 4.200D-08   5.000D+00 .
 4.300D-08   5.000D+00 .
 4.400D-08   5.000D+00 .
 4.500D-08   5.000D+00 .
 4.600D-08   5.000D+00 .
 4.700D-08   5.000D+00 .
 4.800D-08   5.000D+00 .
 4.900D-08   5.000D+00 .
 5.000D-08   5.000D+00 .
```

Plate 3

Simulation of the Arbiter circuit

with different flip-flop output capacitances



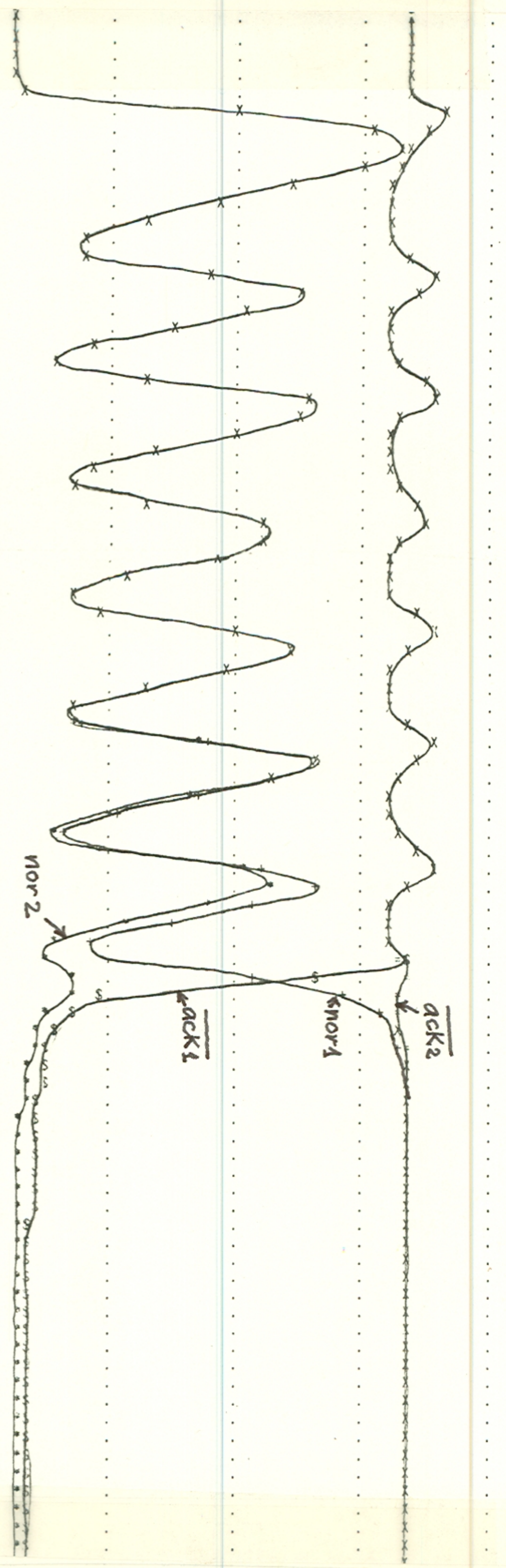
```
TIME        V(6)
X(*+)----------
0.000D+00   5.000D+00
1.000D-09   5.000D+00
2.000D-09   5.000D+00
3.000D-09   5.000D+00
4.000D-09   5.000D+00
5.000D-09   5.000D+00
6.000D-09   5.001D+00
7.000D-09   5.003D+00
8.000D-09   5.009D+00
9.000D-09   4.940D+00
1.000D-08   4.809D+00
1.100D-08   2.393D+00
1.200D-08   9.563D-01
1.300D-08   7.084D-01
1.400D-08   6.560D-01
1.500D-08   5.478D-01
1.600D-08   5.639D-01
1.700D-08   5.168D-01
1.800D-08   4.628D-01
1.900D-08   4.912D-01
2.000D-08   4.786D-01
2.100D-08   4.320D-01
2.200D-08   4.567D-01
2.300D-08   4.624D-01
2.400D-08   4.255D-01
2.500D-08   4.364D-01
2.600D-08   4.649D-01
2.700D-08   4.239D-01
2.800D-08   4.238D-01
2.900D-08   4.489D-01
3.000D-08   4.249D-01
3.100D-08   4.155D-01
3.200D-08   4.386D-01
3.300D-08   4.268D-01
3.400D-08   4.099D-01
3.500D-08   4.311D-01
3.600D-08   4.292D-01
3.700D-08   4.082D-01
3.800D-08   4.254D-01
3.900D-08   4.314D-01
4.000D-08   4.126D-01
```

Plate 4

Simulation of the Arbiter circuit

with buffers in the feedback connections of the flip-flop

```
8.500D-09   3.517D-01 .
9.000D-09   3.517D-01 .
9.500D-09   3.517D-01 .
1.000D-08   3.517D-01 .
1.050D-08   4.213D-01 .
1.100D-08   2.985D+00 .
1.150D-08   4.579D+00 .
1.200D-08   4.965D+00 .
1.250D-08   4.476D+00 .
1.300D-08   3.670D+00 .
1.350D-08   2.771D+00 .
1.400D-08   1.930D+00 .
1.450D-08   1.136D+00 .
1.500D-08   1.137D+00 .
1.550D-08   2.706D+00 .
1.600D-08   3.773D+00 .
1.650D-08   3.136D+00 .
1.700D-08   2.210D+00 .
1.750D-08   1.272D+00 .
1.800D-08   8.357D-01 .
1.850D-08   1.888D+00 .
1.900D-08   3.841D+00 .
1.950D-08   3.803D+00 .
2.000D-08   2.960D+00 .
2.050D-08   2.018D+00 .
2.100D-08   1.261D+00 .
2.150D-08   1.039D+00 .
2.200D-08   1.952D+00 .
2.250D-08   3.361D+00 .
2.300D-08   3.326D+00 .
 .300D-08   2.734D+00 .
4 350D-08   1.750D+00 .
4.400D-08   1.039D+00 .
4.450D-08   1.363D+00 .
4.500D-08   3.020D+00 .
4.550D-08   3.688D+00 .
4.600D-08   2.929D+00 .
4.650D-08   1.978D+00 .
4.700D-08   1.114D+00 .
4.750D-08   1.123D+00 .
4.800D-08   2.620D+00 .
4.850D-08   3.987D+00 .
4.900D-08   3.409D+00 .
4.950D-08   2.504D+00 .
5.000D-08   1.539D+00 .
5.050D-08   9.033D-01 .
5.100D-08   1.396D+00 .
5.150D-08   3.156D+00 .
5.200D-08   3.456D+00 .
5.250D-08   2.626D+00 .
5.300D-08   1.662D+00 .
5.350D-08   9.040D-01 .
5.400D-08   7.853D-01 .
5.450D-08   1.115D+00 .
5.500D-08   1.043D+00 .
5.550D-08   8.013D-01 .
5.600D-08   6.885D-01 .
5.650D-08   6.167D-01 .
5.700D-08   5.597D-01 .
5.750D-08   5.250D-01 .
5.800D-08   4.952D-01 .
5.850D-08   4.796D-01 .
5.900D-08   4.641D-01 .
5.950D-08   4.485D-01 .
6.000D-08   4.377D-01 .
6.050D-08   4.321D-01 .
6.100D-08   4.265D-01 .
6.150D-08   4.208D-01 .
6.200D-08   4.162D-01 .
6.250D-08   4.126D-01 .
6.300D-08   4.090D-01 .
6.350D-08   4.054D-01 .
6.400D-08   4.031D-01 .
6.450D-08   4.021D-01 .
6.500D-08   4.012D-01 .
6.550D-08   4.003D-01 .
6.600D-08   3.992D-01 .
6.650D-08   3.980D-01 .
6.700D-08   3.967D-01 .
6.750D-08   3.955D-01 .
6.800D-08   3.949D-01 .
6.850D-08   3.951D-01 .
6.900D-08   3.953D-01 .
6.950D-08   3.955D-01 .
7.000D-08   3.952D-01 .
7.050D-08   3.946D-01 .
```


nor 2
$\overline{ack1}$
nor1
$\overline{ack2}$