1986

Simulation analysis and design of asynchronous CMOS arbiter /

Sina Jafroodi
Lehigh University

SIMULATION ANALYSIS AND DESIGN OF ASYNCHRONOUS CMOS ARBITER

by

Sina Jafroodi

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in Electrical Engineering

Lehigh University

1986

This thesis is accepted and approved

in partial fulfillment of the requirements

for the degree of Master of Science

9/25/86

(date)

Professor in Charge

Chairman of Department

Acknowledgments

I would like to thank Professor Alfred K. Susskind and Professor Frank Hielcher for their helpful discussions and ideas.

TABLE OF CONTENTS

Abstract	1
1. Introduction	2
2. The Synchronization Problem	10
2.1 Noise Independence	15
2.2 Mean Time Between Failures	22
2.3 Metastable Detecting Circuits	27
2.4 A CMOS Metastable Detecting Circuit	31
3. Corrective Circuits	36
3.1 Output Characterization	36
3.2 Input Characterization	40
3.3 A Runt Pulse Removing Circuit	47
3.4 A Fault Tolerant CMOS Flip-Flop	49
4. Arbiter Design	54
4.1 Asynchronous Arbiter Module	55
4.2 The Arbiter Module	57
Conclusion and results	61
References	62
Vita	64

ABSTRACT

An arbiter must resolve the conflict between multiple requests for a common resource and allow only one processor to access the requested resource within a finite time. The design of asynchronous arbiters presents some difficulties because multiple input changes are allowed, and because inputs may change when the arbiter is not in a stable state. The flip-flop used as synchronizing element in such arbiters may therefore fail to make an unfaltering decision within a prespecified time. Significant system failures may result from this commonly observed problem.

This thesis deals with the behavior of flip-flops used as input synchronizers, in particular when they operate in the metastable state. An approach to calculate the minimum pulsewidth required by a flip-flop to assure avoiding the metastable state during transition period is presented. Moreover, two solutions are considered to remove runt pulses that may appear at the input or output terminals of such flip-flops.

An analysis of the mean time between failures of a flip-flop is represented. A fault tolerant CMOS flip-flop is proposed that is used in design of an asynchronous arbiter module. Finally the performance versus reliability of a system has been considered.

1. INTRODUCTION

A digital system may incorporate several independently clocked processors . These processors may need to share a common resource or a logical unit to carry out their intended task. Any such transfer of information between a processor and a common resource must be carried out in a time frame which is acceptable to both units. It is likely that within a very short period of time two or more processors require access to a particular common resource. An arbiter is the control mechanism used in resolving multiple requests by processors.

When arranging for communication between two processors or subsystems that do not share a common time reference, it is impossible to avoid the generation of logically undefined pulses or glitches. The conventional solution is to use these signals as inputs to a synchronizing element, typically a flip-flop. The assumption is thus made that the flip-flop will reach a stable state within a finite time after the trigger pulse is applied, even if the input to the synchronizing flip-flop was a runt pulse. A delayed clock pulse is then gated by the flip-flop output to produce a request or interrupt signal to the arbiter or central processor. The basic form of the synchronizer is indicated in Figure 1.1. Figure 1.2 shows selected output trajectories of a CMOS flip-flop in response to marginal triggering.

This assumption of guaranteed flip-flop stability in a finite time is not always valid. It has been shown qualitatively [1], that

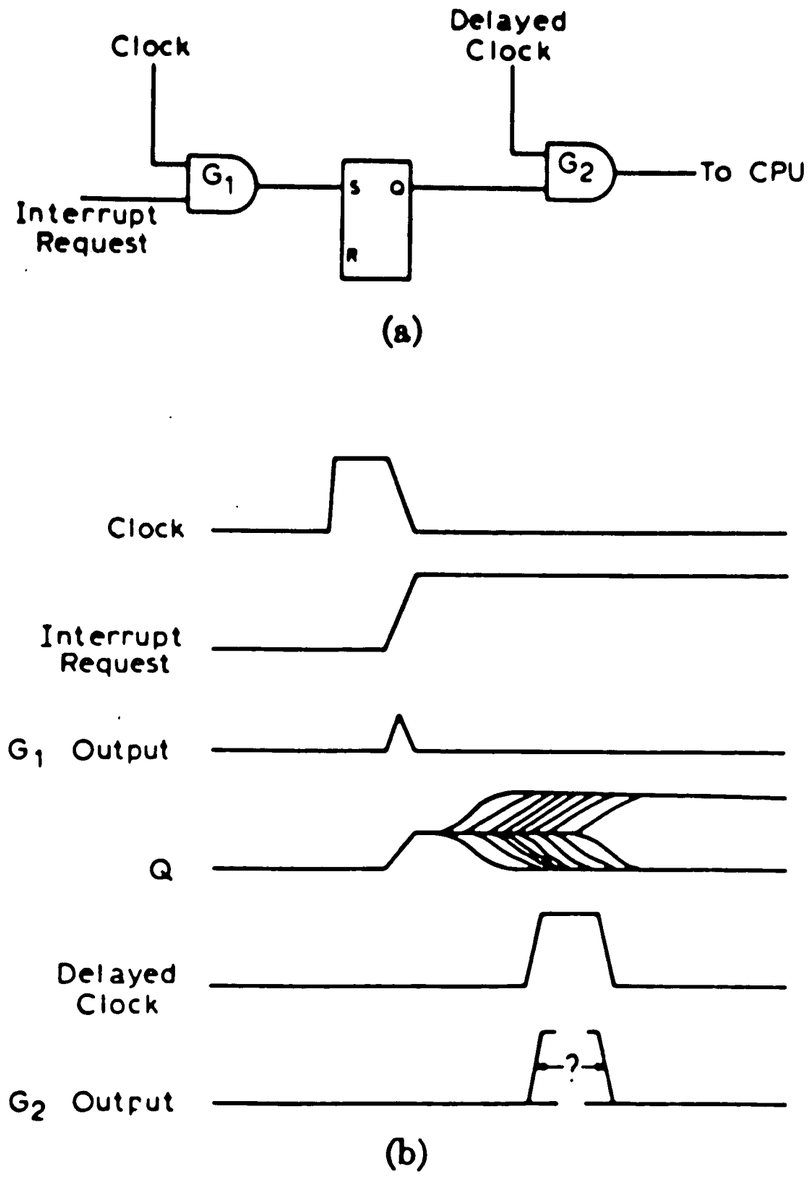

Figure 1.1. Synchronizer circuit. (a) Logic diagram and (b) typical waveforms [22].

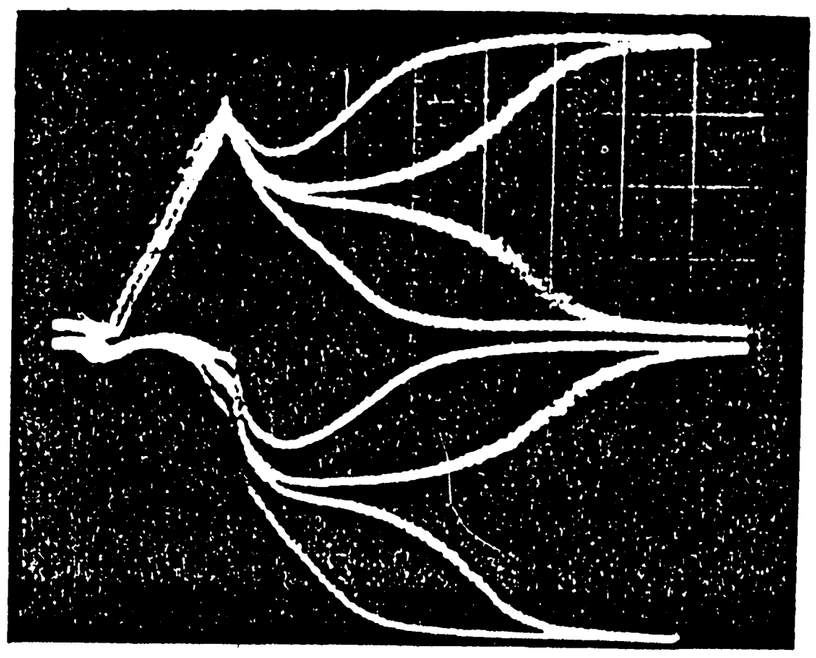

Figure 1.2. Selected output trajectories of CMOS cross-tied NAND type flip-flop in response to marginal triggering [5].

in a noise-free environment there is no fixed time interval sufficiently long to ensure that the flip-flop will, with probability 1 , reach a defined output state. Also shown in [1] is that for certain trigger pulse energy levels a metastable state is maintained for an indetermined period of time. The term metastable operation or metastable region refers to the prolonged transition time of a bistable device that may result if the input that causes the bistable to change state is a runt pulse. Moreover, a runt pulse or a logically undefined input may be described as an input to a particular device where a deterministic response at the output is not possible based on the properties of that particular input, i.e. the input does not have the properties to be classified as a logic 1 or a logic 0 by the receiving device.

Chaney and Molnar [2] have observed oscillatory and metastable behavior of flip-flops in response to logically undefined input conditions such as those that occur in arbiters and synchronizers. Chaney [3] has also measured and characterized the marginal triggering response of several types of flip-flops. He has shown that such characterization is essential to predict the reliability of synchronizer designs, even though they are seldom specified or measured by manufacturers of such devices.

Similar experiments performed by other authors [4]-[5], has revealed that failure rate variations tend to straight lines quite rapidly on a logarithmic scale as shown in Figure 1.3(a). Other

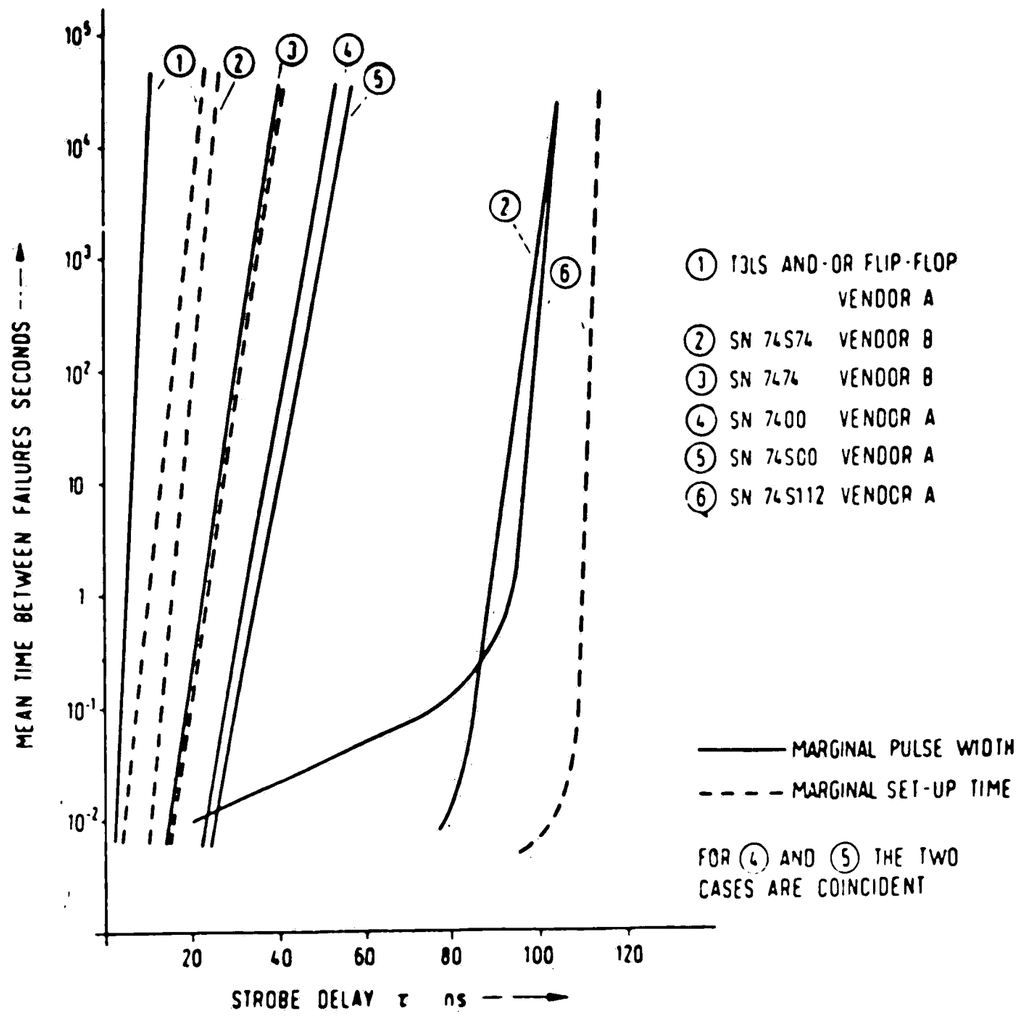

Figure 1.3(a). Failure rate for various flip-flops under marginal triggering conditions [4].

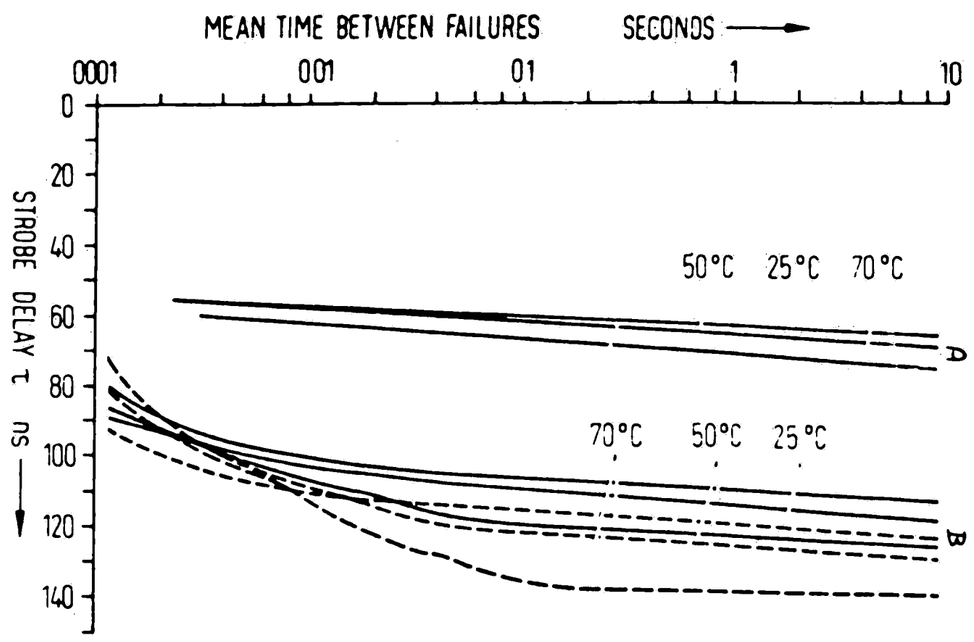

Figure 1.3(b). The effect of increase in ambient temperature on the failure rate of SN74S112 flip-flop [4].

findings indicate that as the temperature increases, so does the failure rate. The probable reason for this is attributed to a decrease in the amplification of the transistors and hence a decrease in the overall gain. With any decrease in the gain it takes more time for the output to settle into a stable state. Figure 1.3(b) represents the experimental results due to this effect under marginal triggering of two samples of a 74S112 flip-flop. Strobe delay τ is defined as the delay from the triggering edge of clock or data to the "strobe" input minus the normal propagation delay of the flip-flop. Also concluded is that flip-flops with more complex circuits have longer metastable durations.

Other authors [6], have concluded that some information on flip-flop response in the metastable region is required to predict its behavior. They have also indicated that a flip-flop circuit to be used as a synchronization or arbitration element must have a response which is fast and non-oscillatory. Having fulfilled these conditions will then allow one to construct systems with known reliability.

It is known that clocked systems have no equivalent synchronizing solution [7], i.e., a clocked input synchronizer may fail to produce a synchronized output in a finite time and there is no way to reduce the probability of failure to zero. This is due to the fact that it is not possible to design a bistable device that does not have a region of anomalous or metastable operation. Pechoucek [8] and Veendrick [9] have thoroughly analysed the anomalous behavior of input synchronizers and provided different methods to predict their rate of

failure. They have also proposed and evaluated different solutions to reduce the probability of failure.

An attempt was made by Wormald [10] to avoid the maloperation of synchronizers or interlocks due to metastable action when handling asynchronous signals. In this approach the possibility of setting a flip-flop to its metastable state is "removed" by insertion of a Schmitt trigger within the flip-flop's inner loop, as shown in Figure 1.4. This proposal, however, ignores proofs presented by several authors [11], that "every device that has at least two stable equilibrium states, regardless of how the devices are made, must have at least one region of unstable equilibrium", [12]. It has also been shown through experimental results [12], that the presence of the Schmitt trigger section significantly degrades the synchronization performance of the flip-flop. The primary intent of the experiment, as claimed by the author, was to challenge the belief that there is an electrical or logical circuit that will provide the synchronizer function, that has a probability of failure equal to zero.

As an analytical treatment to this unrestricted input change problem Unger [13] has shown how to design asynchronous systems that will operate in a "satisfactory manner" even where the occurrence of input signals are independent of one another. This discussion covers only single-output-change (SOC) functions, characterized by the fact that no more than a single change in the output state will occur when changing the input state once. Even though "satisfactory type" state

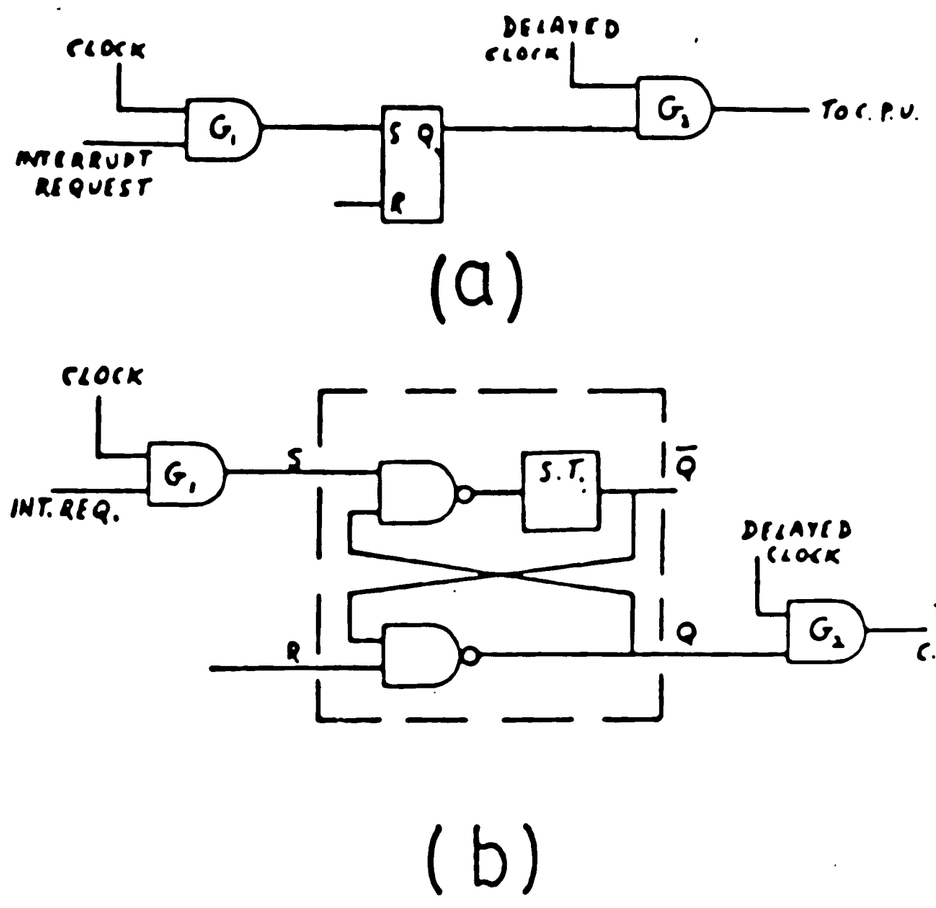

Figure 1.4. (a) Conventional synchronizer, (b) suggested circuit for input synchronizer flip-flops [10].

assignment can be achieved by design in compliance with his procedure, however, elimination of short input pulses can not be guaranteed and this is a constraint which is often violated by signals at the inputs to synchronizing elements and arbiters.

A practical design of a general purpose asynchronous arbiter has been presented by Plummer [14]. In this design some provisions that might become necessary in a given application have been accounted for. It has also been determined that the complexity added to the design has reduced its probability of failure. Less complex designs that are

more prone to failure have also been suggested by other authors [15]-[16]. One of the two designers [15], however, proposes some timing modifications to avoid possible maloperations.

Among the three asynchronous arbiters of [14] through [16] mentioned above, each design uses different circuitry to carry out a similar task. What is common in all designs, however, is the fact that each proposed arbiter will fail to perform the synchronization function under certain input combinations and particular timming sequence.

The purpose of this thesis is to find a logical function that would perform the synchronization function with very high probability of success. In order to accomplish this task simulations of synchronizing flip-flops have been performed. Also the behavior of flip-flops in the metastable state has been studied. Several input pulses with different pulsewidth, magnitude and shape have been used to achieve metastability in the flip-flops. The output behavior of these flip-flops under marginal triggering was studied and characterized.

Based on this characterization an approximation of the minimum pulsewidth required by a flip-flop was made. Using this result a logical circuit is designed that will guarantee the removal of all inputs with a pulsewidth less than that of the desired parameter. Moreover, a metastable detecting circuit is suggested that can be used in a new flip-flop configuration which is extremely tolerant of input timing.

2. THE SYNCHRONIZING PROBLEM

Design of synchronous systems presents some serious limitations, which are made worse as the dimensions are scaled down and as chips become larger. The difficulties of moving information from point to point within a single clock period is one of a few to be named. Another limitation is the difficulty of managing very large designs in which all system parts must operate in sequential mode and produce a deterministic output within a single clock period.

These considerations provide some motivation to divide a system into modular parts and require that the parts be independently timed. In such systems where each part is a synchronous subsystem of its own, information transfered from one part to another must be synchronized to the receiver's clock. The primitive method to bring a data or control signal **A** into synchronization with a system with clock signal **B** is to use signals **A** and **B** as inputs to an NAND gate. The output of such a gate can then be used in a system with clock **B** as a synchronized data or control signal. By using this method of synchronization there exist a possibility that a runt pulse will occur at the output of the NAND gate if signals **A** and **B** overlap for a short peroid of time, as shown in Figure 2.1(a).

In the event that the signal that appears at **C** is used as a trigger pulse for a data transfer, it is likely that device **D** might indicate that data transfer has taken place at time t_1 while device **E**

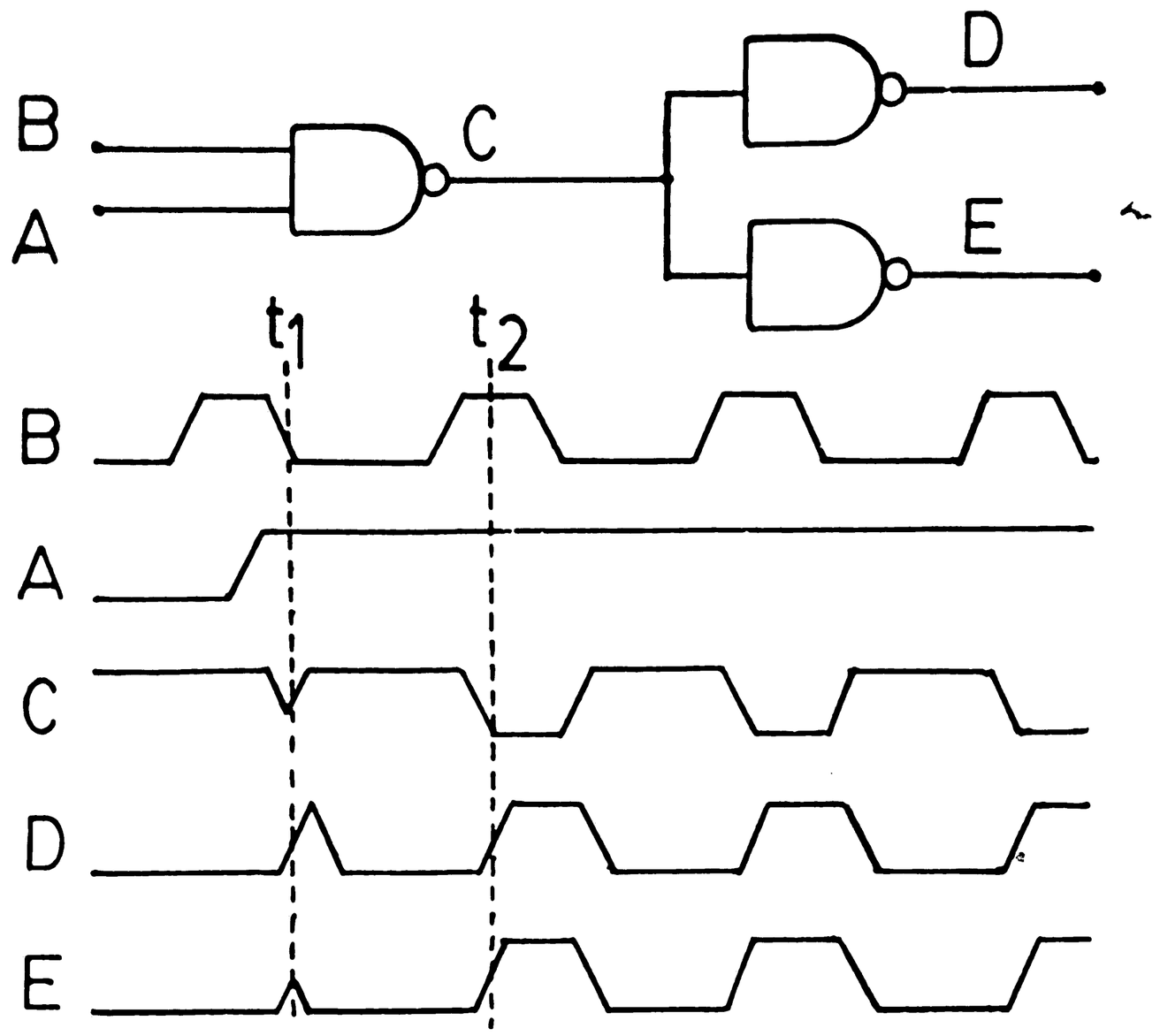

Figure 2.1(a). A NAND gate synchronizing circuit and typical waveforms.

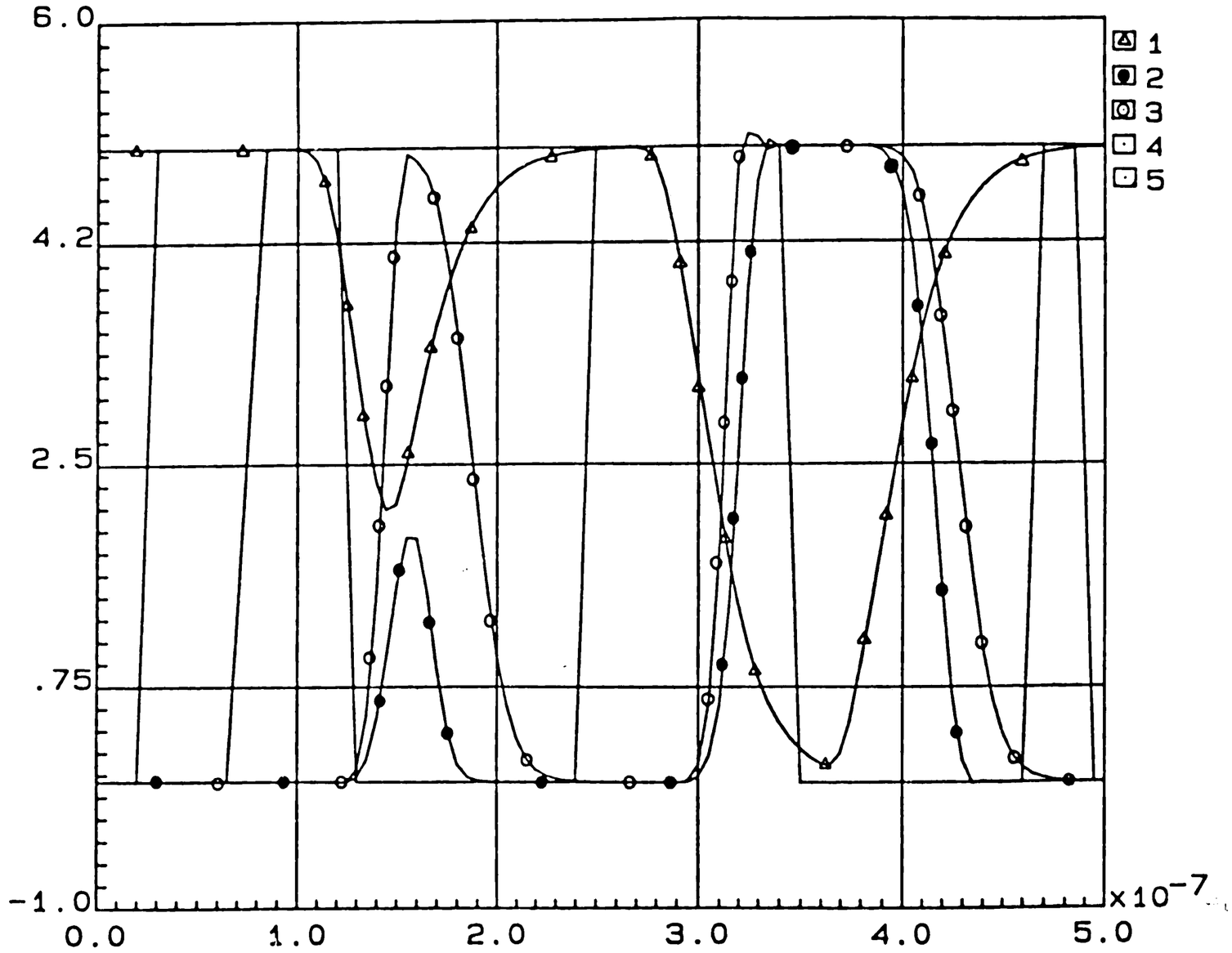

Figure 2.1(b). Output (volts) vs. time (seconds). Simulation result of a NAND gate synchronizer circuit. Curve 1 is the output at C. Curve 2 and 3 are outputs at D and E.

has allowed the transfer to take place at time t_2 . The simulation results of such circuitry is shown in Figure 2.1(b).

An alternative method is to use signal C as an input to a bistable device in order to resolve the discrepancy among the receiving devices as shown in figure 2.2(a). This discrepancy arises in interpretation of the logic level of the runt pulse. The assumption made by the designer is that the pulse that appears at C will set the flip-flop to a logic 0 or 1 within some prespecified time. However, there is still a statistical possibility that for certain input combinations at A and B, the flip-flop may enter its metastable or half-set state for an extended period of time. As a result the system will get out of step. The simulation results of such configuration has yielded similar results to the conditions stated in this paragraph and extended periods of metastability was also observed as shown in Figure 2.2(b).

A final but similar approach uses a D-type flip-flop such that the D input is connected to the line where an external or asynchronous signal A is expected. The clock terminal of such a flip-flop is then triggered with the clock signal B of the receiving system. The Q terminal of such flip-flop will then produce the synchronized equivalent of signal A. It should be noted that sequential devices such as flip-flops are normally designed with some operating constraints such as an input setup time and hold time requirements with respect to the clock input.

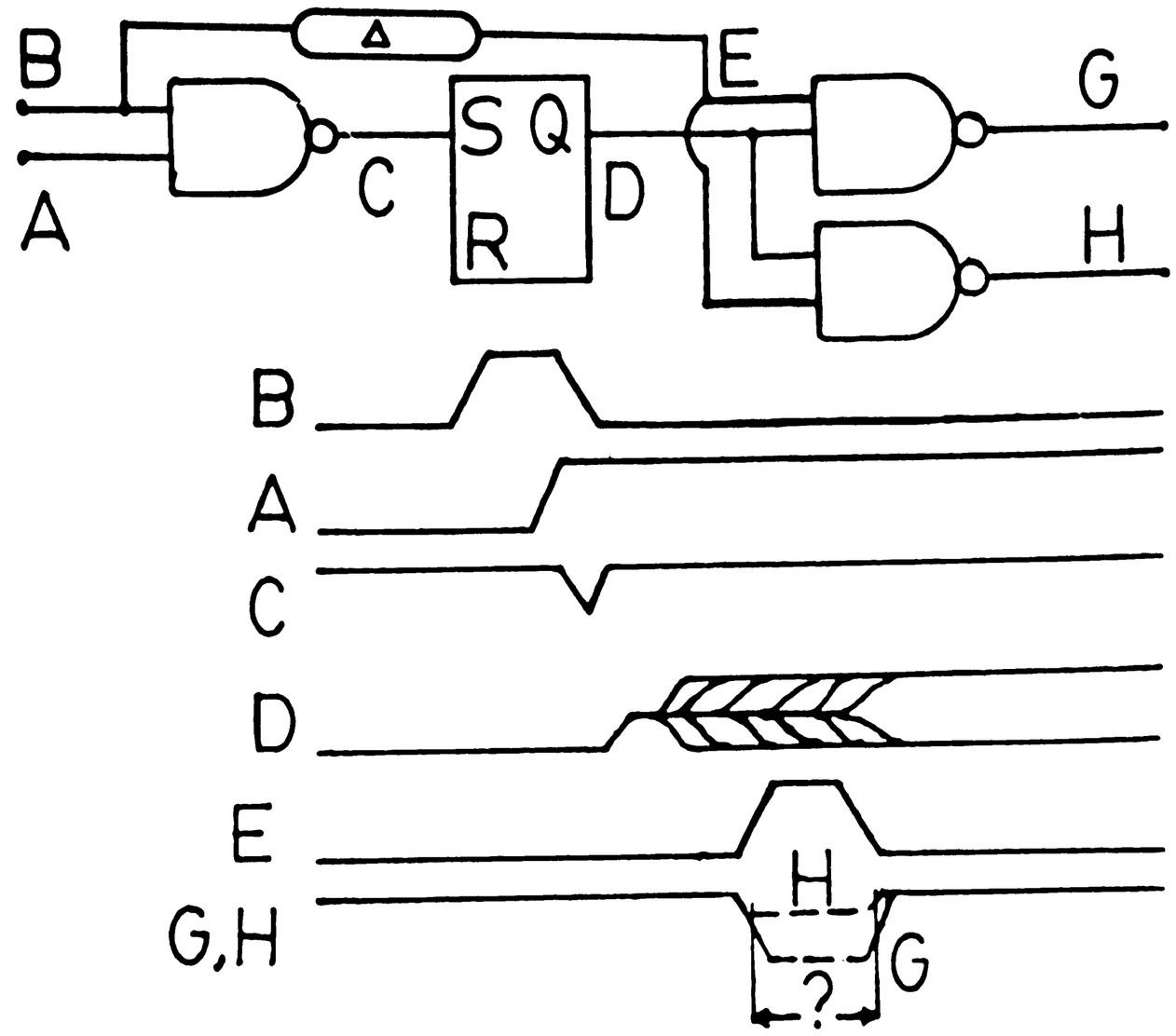

Figure 2.2(a). (a) an S-R input synchronizer circuit, (b) typical waveforms.

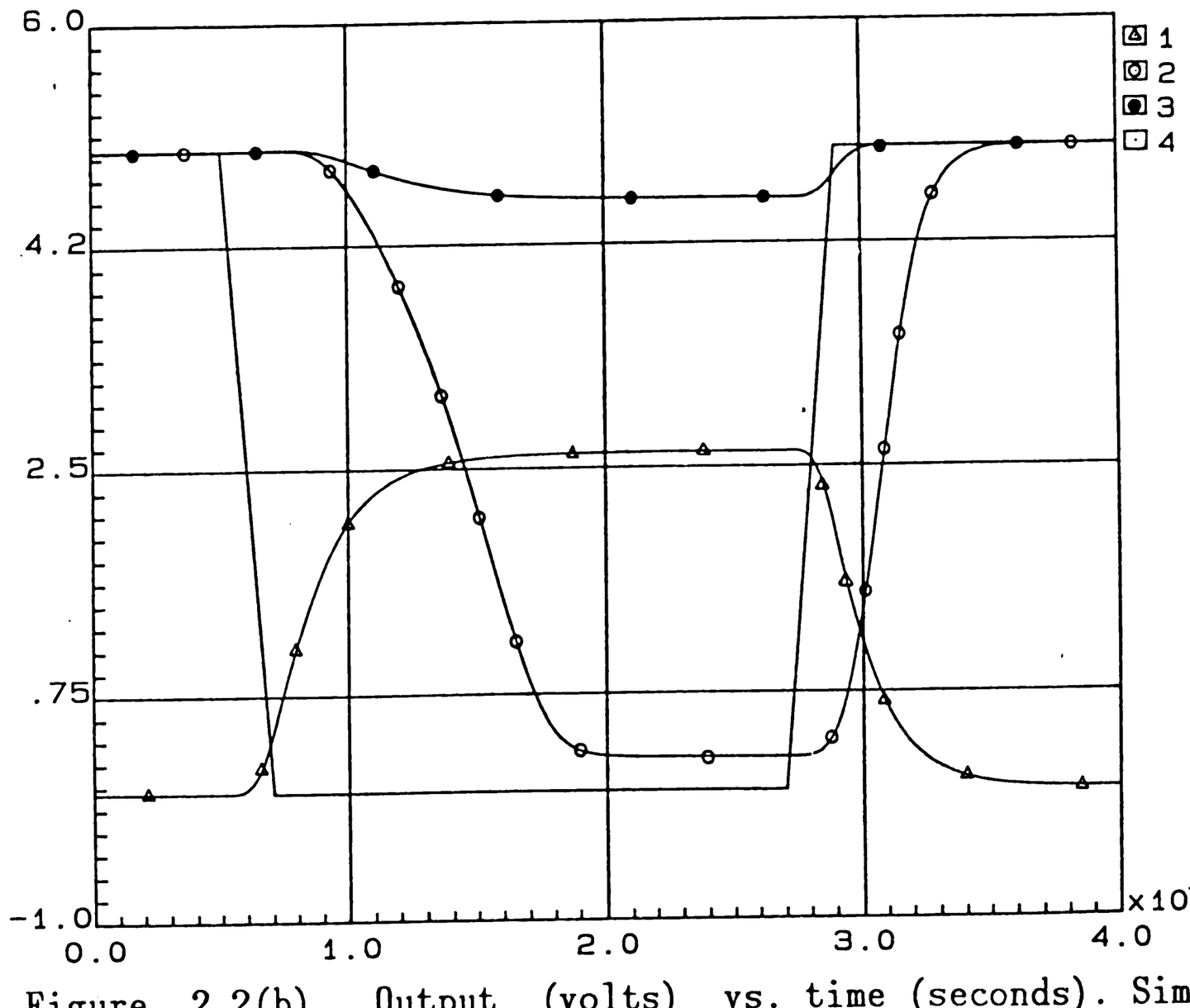

Figure 2.2(b). Output (volts) vs. time (seconds). Simulation result of an S-R input synchronizer circuit. Curve 1 is the output at D, and curves 2 and 3 are outputs at G and H.

The term setup time can be defined as a definite time, in which an input must be maintained at a constant value prior to the application of a clock pulse. Hold time is a definite time that an input must not change after the application of the triggering edge of a clock pulse. These are the constraints that must be met in order to assure that the bistable won't be forced into its astable state. Since there is no timing boundaries on when the signal **A** may occur with respect to the clock **B**, the constraints set by the manufacturer of such devices are likely to be violated. Moreover, the interpretation of the logic level seen at the output of the flip-flop may be different. For example, one flip-flop could interpret a logically undefined value as a logic **1**, while another flip-flop could interpret the same signal as a logic **0**.

A similar misinterpretation of a logic level could occur if the input signal to state registers of a sequential network is changing value. As a result, some state register flip-flops could capture the value of an input signal before the change of that signal while others may respond to the value of an input signal after the change has occured at the input, producing a state transition that is correct for neither value.

In order to guarantee the proper functioning of a synchronous sequential system with zero probability of failure it is required that all components in the system have zero probability of malfunction. Moreover, it is essential that all devices that depend on the value of

a given signal at a time see the same value. Thus preserving consistency in a system should be a primary concern.

2.1 NOISE INDEPENDENCE

Consider a flip-flop consisting of two NAND gates as shown in Figure 2.3(a). The condition in which both gates of the flip-flop are operating as inverting amplifiers and in which both gates are carrying the same current is an unstable equilibrium [16]. If the circuit is displaced from its initial state V_M due to a noise or impulse signal, it will continue to move in the direction in which it was displaced until further excursion is limited by the NAND gate nonlinearities. The voltage transfer characteristics of such flip-flop is shown in Figure 2.3(b). This is due to the rapid regenerative action found in a flip-flop configuration. An equivalent circuit for such a flip-flop can be approximated by two stages, each containing an inverter with amplification -A followed by an RC filter with time constant $\tau = RC$ (Figure 2.4). Thus solving the first order differential equations of the given model yields [9]:

$$v_1 = \lambda_1.\exp[(A-1)t/\tau] + \lambda_2.\exp[(-A-1)t/\tau] \qquad (2.1)$$

$$v_2 = -\lambda_1.\exp[(A-1)t/\tau] + \lambda_2.\exp[(-A-1)t/\tau] \qquad (2.2)$$

where λ_1 and λ_2 are integration constants.

We denote the initial conditions at time t=0 as:

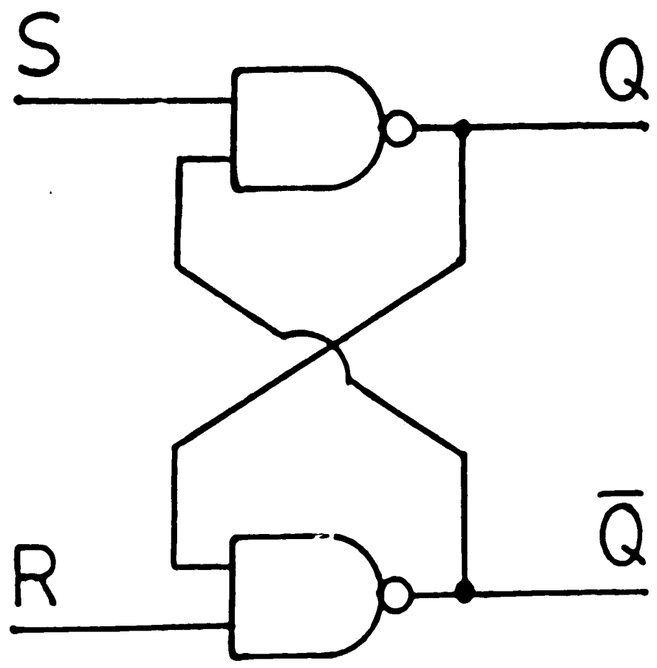

Figure 2.3(a). A NAND type set-reset flip-flop.

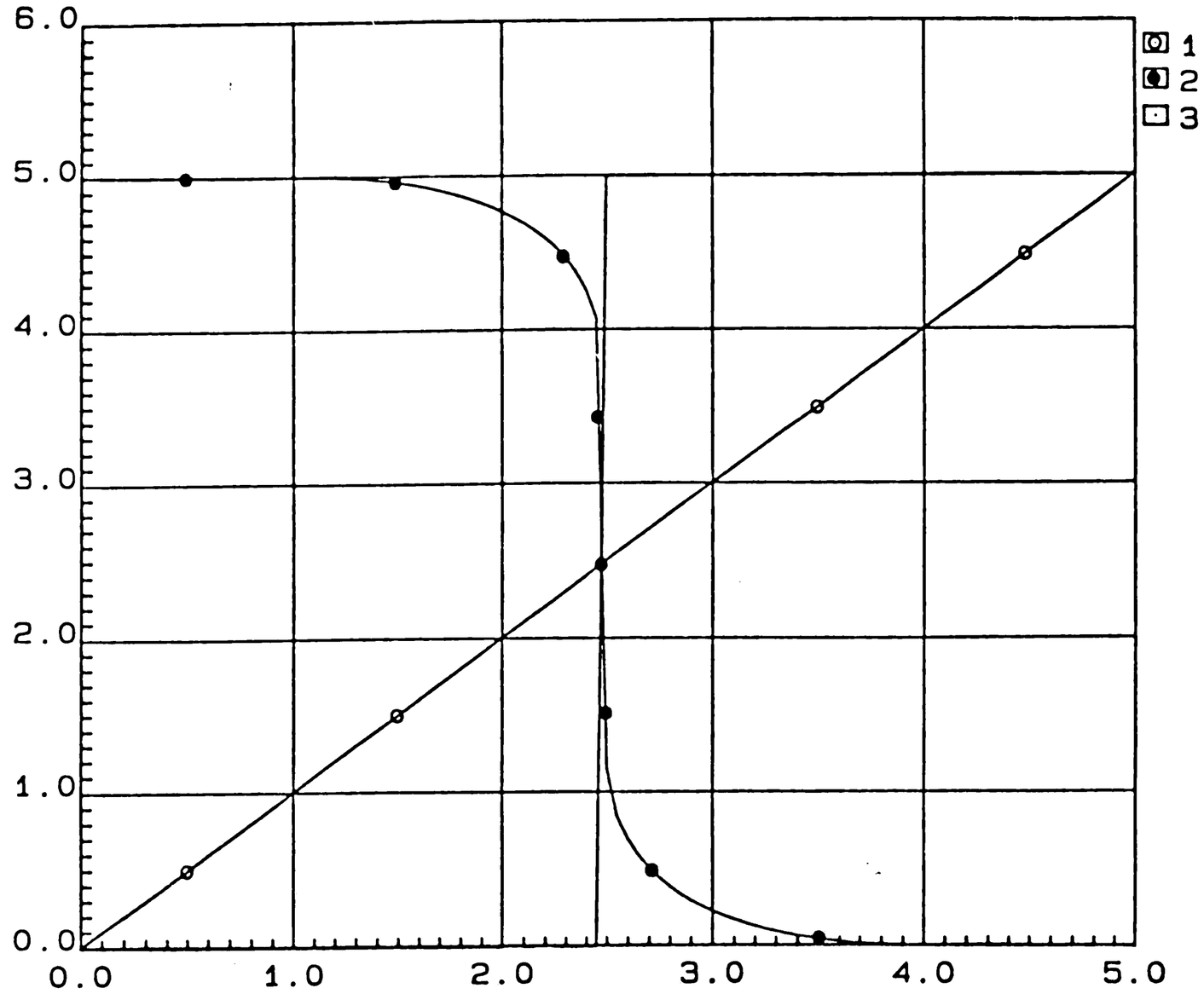

Figure 2.3(b). Output (volts) vs. input (volts). Voltage transfer characteristics of a NAND gate with midpoint voltage V_M.

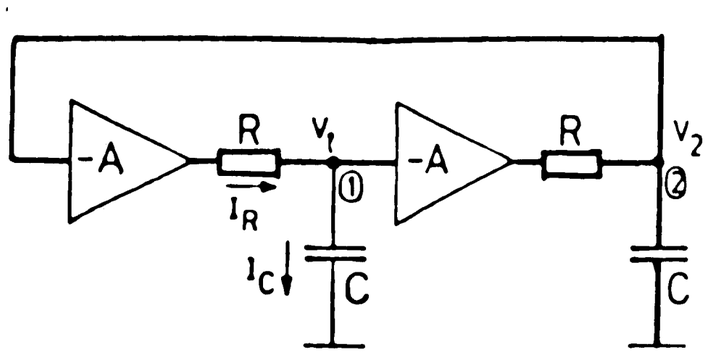

Figure 2.4. Equivalent first-order small-signal model of a flip-flop [9].

$$v_1 = v_{01} = \lambda_1 + \lambda_2 \tag{2.3}$$

$$v_2 = v_{02} = -\lambda_1 + \lambda_2 \tag{2.4}$$

hence at any time t after the sample moment the voltage at v_1 and v_2 can be expressed as [9]:

$$v_1 = \frac{1}{2}\{(v_{01}-v_{02})\exp[(A-1)t/\tau] + (v_{01}+v_{02})\exp[(-A-1)t/\tau]\} \tag{2.5}$$

$$v_2 = \frac{1}{2}\{(-v_{01}+v_{02})\exp[(A-1)t/\tau] + (v_{01}+v_{02})\exp[(-A-1)t/\tau]\} \tag{2.6}$$

If we consider only node 1 and assume that the second inverter is operating in the linear region at sample moment (t=0), we can see that the voltage v_{02} at node 2 will be equal to the voltage v_{01} at node 1 times the amplification constant -A.

$$v_{02} = -A \cdot v_{01} \tag{2.7}$$

where the value -A is an amplification factor in linear region. Thus we can obtain a new relationship for v_1 by substituting equation 2.7 into equation 2.5 and therefore:

$$v_1 = \frac{1}{2} v_{01}\{(1+A)\exp[(A-1)t/\tau] + (1-A)\exp[(-A-1)t/\tau]\} \tag{2.8}$$

It can be seen from equation 2.8 that at any time t after the sample moment t=0 each value $v=v_{01}$ is multiplied by the same factor inside the brackets. Since the values of v_{01} are uniformly distributed at time t=0, so the values of v_1 will remain uniformly distributed at any given time [9]. The second term inside the brackets represents a decaying exponential which will tend to zero quite rapidly as time increases. Thus we can omit this term and rewrite equation 2.8 as follows:

$$v_1 = \frac{1}{2} v_{01}\{(1+A)\exp[(A-1)t/\tau]\} \tag{2.9}$$

We can simplify equation 2.9 as follows:

$$v_1 = K_0 \cdot v_{01} \cdot \exp[(A-1)t/\tau] \tag{2.10}$$

The constant K_0 is a circuit parameter and is a function of design and technology used and will vary from device to device.

Equation 2.10 describes the output trajectories of a flip-flop in response to uniformly distributed input samples v_{01} (at time t=0). The behavior of the output v_1 as expressed in equation 2.10 is illustrated in Figure 2.5. It can be deduced from equation 2.10 that at any given time $t=t_s$ after the sample moment, there exist an equal number of trajectories or states in equally sized regions $\delta(v)$ over the magnitude of v_1. This is due to the fact that v_{01} is uniformly distributed at time t=0. Moreover, the output behavior of a flip-flop in the metastable state follows similar trajectories as expressed in equation 2.10. These trajectories, however, will be centered at the voltage V_m.

In order to account for the effects of noise on these trajectories consider two regions of outputs R_1 and R_2 of size $\delta(v)$ at time $t=t_s$ after the sample moment. Also consider the number of trajectories in each region being N_1 and N_2 where $N_1=N_2$ as shown in Figure 2.6. We can assume that at time $t=t_s$ there exists a disturbing voltage with magnitude V_a at node 1. By superimposing this disturbing voltage over all trajectories, the output states or trajectories in

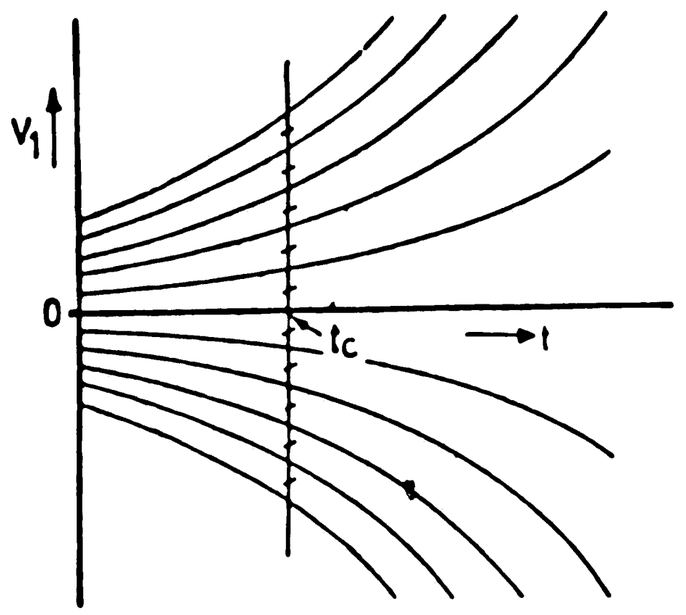

Figure 2.5. Output trajectories of a flip-flop for uniformly distributed sample values (at t=0) [9].

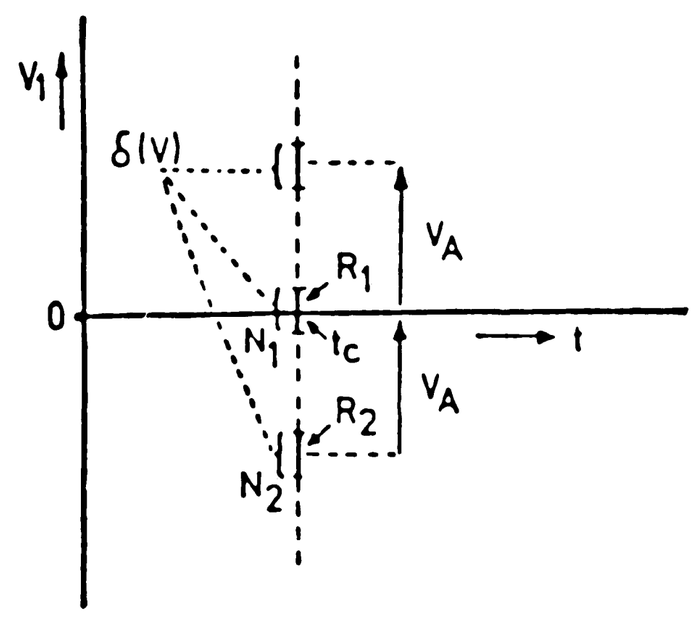

Figure 2.6. The number of states N_1 that is forced out of the metastable region by a disturbing voltage V_A is replaced by the number of states N_2 that is forced back into the metastable region [9].

region R_1 of size $\delta(v)$ are moved out of the metastable state due to the voltage V_a and at the same time the trajectories in region R_2 of size $\delta(v)$ are shifted into the metastable state. Since the number of trajectories in region R_1 are equal to the number of trajectories in region R_2 $(N_1=N_2)$ due to the uniform distribution of v_{01} (at time t=0), it follows that a disturbing voltage of any magnitude does not affect the average number of trajectories or metastable states that last longer than a certain time $(t=t_s)$ over a large number of events.

A similar analogy applies to circuit noise, which can be considered as a series of disturbing voltages over time with a zero mean. Thus at any time after the sample moment and within or after the transition period of a flip-flop, circuit noise does not disturb the average state distribution , and as a result does not change the average duration and number of metastable states. Moreover, it has been derived [9] that the average number of metastable states M_s, lasting longer than a certain time t_s among N_0 samples which are uniformly distributed between logical 0 and logical 1 is given by:

$$M_s = N_0.\exp[(-A+1)t_s/\tau] \qquad (2.11)$$

Even though circuit noise randomly affects each individual metastable state, the average number of metastable states during s time period is independent of noise. The expression given above is applicable to any flip-flop that can be modeled as two inverting amplifiers with amplification -A followed by an RC-filter after each

inverter with a time constant τ = RC. Thus to decrease the number metastable states over a large period of time, the factor (A-1)/τ has to be maximized.

2.2 MEAN TIME BETWEEN FAILURES

Synchronous sequential circuits are guaranteed reliable provided that the network components do not malfunction and that certain operating specifications are met. Thus a system is reliable given that the system components have zero probability of failure. It is possible that a system fails to function properly even though when there exist no component failure within the system, i.e. the failures are caused by the occurrence of certain parasitic inputs which can not be prevented by the imposition of any input requirements. Typical input requirements for a synchronous system might be that of set-up or hold time of the incoming data with respect to the occurrence of a clock pulse, or minimum pulsewidth required for the system clock.

Systems comprised of several independently clocked modules may require the use of several input synchronizers for each module to handle asynchronous signals from other modules or systems. The failure rate of a synchronizer is directly proportional to the frequency of the asynchronous inputs. As the frequency and number of asynchronous inputs grow, synchronizer failures occur more rapidly. This is due to the fact that the response time of synchronizing elements such as D-type flip-flops is not only a function of the input value and clock

but also a function of the time of occurrence of an input signal with respect to the clockpulse.

When a synchronizing flip-flop is placed in its non-stable or metastable state, stabilization occurs as a result of regenerative feedback. The response time of such flip-flop will depend on the state from which stabilization begins. Consider a system timing configuration as shown in figure 2.7. In this example all flip-flops are represented as positive-edge triggered devices with zero set-up and hold time.

Suppose the data input to flip-flop 1 is an asynchronous signal and is random with respect to the clock. As long as data transitions occur well before the rising edge of the clock signal, the output of flip-flop 1 will switch with a normal propagation delay. If we assume a propagation delay of 10 nsec and a clock period of 100 nsec, then flip-flop 2 will have a stable input after a total delay of 60 nsec after the occurence of the clock. Thus flip-flop 2 will have a valid data for 40 nsec and similarly flip-flop 3 will get a valid data for 70 nsec before the next clock.

This system configuration, however, will fail if a data transition occurs which is very near the clock edge and will result in abnormally long delays before the final transition of Q_1. Suppose that a data transition has occured very near the rising edge of a clock-

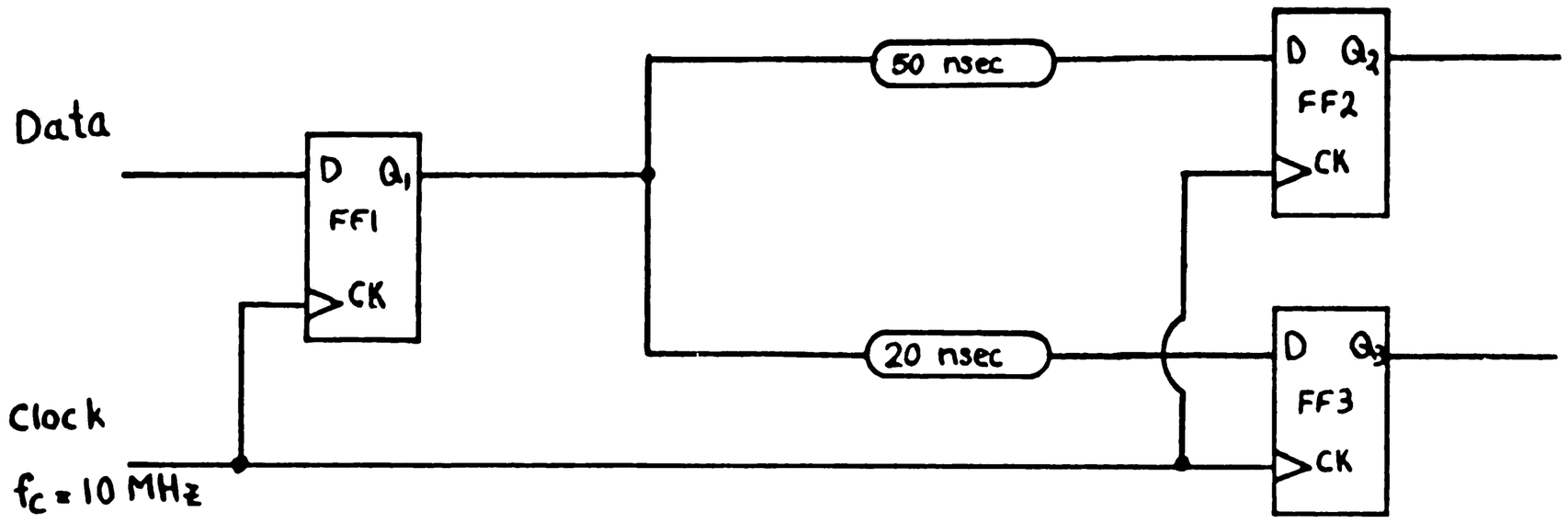

Figure 2.7. System timing example for failure discussion.

pulse. Moreover, suppose that this transition results in an output transition which is 50 nsec longer than normal. In this case, the transition will reach flip-flop 2, 10 nsec after the positive edge of the clock and thus will not affect the output of flip-flop 2. The same transition, however, will reach flip-flop 3, 20 nsec before the occurrence of the clock, and therefore will change Q_3 to a new value that corresponds to the latest data transition. As a result two different parts of the system will have different information as to what data value appeared at the input. For example, one part of the system might think an interrupt has occured, while other parts are processing their normal routine.

It can be seen in the example above that even though a component failure has not occured the system itself will malfunction due to prolonged response time. As the frequency of the system clock and the number of data transitions increase, the frequency of these failures will also increase. In order to calculate a system failure rate due to synchronization problem consider a D-type flip-flop with asynchronous input. Figure 2.8 shows output delay versus input timing for such flip-flop.

The origin in figure 2.8 corresponds to the inputs that are concurrent with the triggering edge of the clock. At the left side, inputs occuring well before the clock edge cause an output transition one propagation delay (t_{pd}) after the clock. At the right side, inputs arriving after the clock edge cause no output transitions at all. In the center, however, input transitions near the clock edge cause extra delays. There is no bound to this delay for an input transition that occurs exactly at the clock edge.

We can now calculate the mean time between failures of a flip-flop following these definitions. An error is the occurence of a response time t_e, which is greater than the normal response time of a flip-flop. A failure is an inconsistency caused by an error. Errors occur more often than failures because not all errors will result in an inconsistent interpretation of a signal. An uncertainty window function $t_w(t)$ can be defined as the range of input values to a flip-

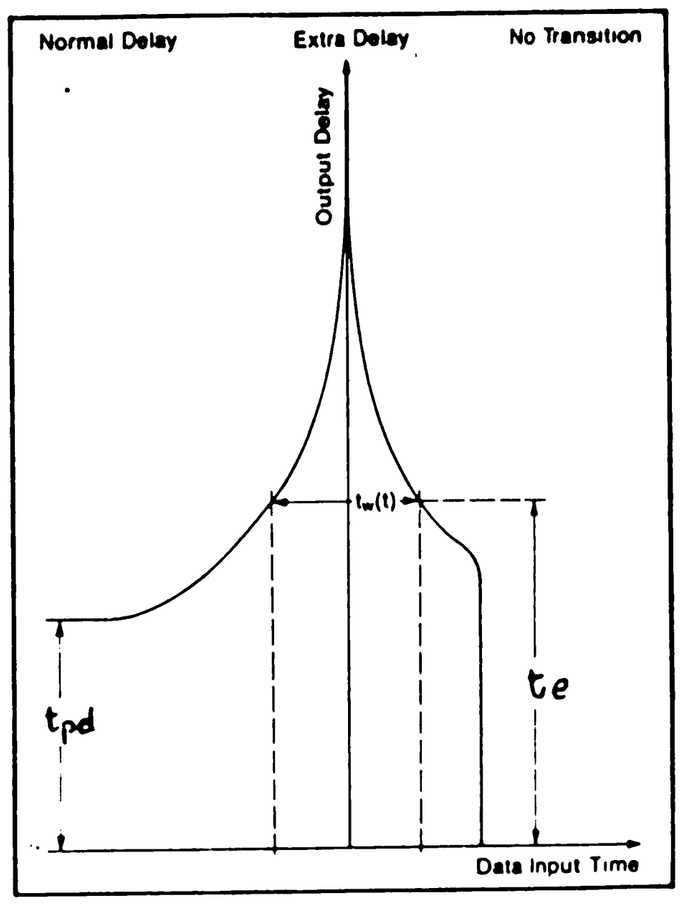

Figure 2.8 Flip-flop output delay vs. data input time.

flop for which an error occurs. The width of this window spans both sides of the trigerring edge of the clock. As can be seen in figure 2.8 errors that have longer excess delays have smaller uncertainty window, i.e., they are less likely to occur than errors with short excess delays. The general form for the window function in regenerative flip-flops is given in the following equation [3]:

$$t_w(t) = T_0 . \exp^{-(t_e - t_{pd})/\tau} \qquad (2.13)$$

where t_{pd} is the normal propagation delay of the flip-flop and τ is the time constant of resolution of the flip-flop. T_0 is a circuit

parameter and is obtained through experimental results. Methods for obtaining T_0 are given in [3], [19] and [21].

Using this window function we can now express the mean time between failures of a system due to a flip-flop error as follows:

$$MTBF(t) = [f_c . f_d . t_w(t)]^{-1} \qquad (2.14)$$

where the window function $t_w(t_f)$ is expressed as follows:

$$t_w(t) = T_0 . \exp^{-(t_e - t_{pd})/\tau} \qquad (2.15)$$

where f_c and f_d are the clock and average data input frequencies.

Although the resolving time for a perfectly symmetric flip-flop may be infinite for $t_d = 0$, non-symmetrical properties, including those due to nonuniform chip temperature or due to asymmetric voltages on circuit capacitances and also the effects of noise can cause an offset to the value of t_d for which the response time is infinitely long. Moreover, hysteresis or history dependence can create a situation where, a flip-flop may favor setting if it was set the previous time and favor resetting if it was reset the previous time.

2.3 METASTABLE DETECTING CIRCUITS

In order to avoid possible failures in fixed-period clocked systems and synchronizers due to extended periods of flip-flop response time, a synchronizing scheme that uses a clock with extensible recurrence time has been suggested [8]-[18]. In this

approach extensible separation between consecutive clock events is possible using a metastable detecting circuit to control the generation of the clock events. This circuitry is devised by Stucki and Cox [18], using an XNOR circuit at the Q and $\overline{Q}$ outputs of a flip-flop as shown in Figure 2.9. It should be noted that a flip-flop must exhibit an output behavior during its metastable state that is uniquely different from its output behavior during stable states.

It is known that metastability in CMOS flip-flops occurs at the unity amplification point or midpoint voltage V_m of their voltage transfer characteristic. Moreover, during this episode the outputs Q and $\overline{Q}$ have similar values and are fairly static around the value V_m. As a result, the existence of the metastable state can be detected with a voltage comparator where an output of 1 would only occur if the input signals are within a certain threshold of one another. The designer thus has control in setting what threshold should be used to distinguish between stable and the metastable states.

If we assume that both NMOS gates in Figure 2.9 have the same threshold voltage V_T, then the operation of the circuit can be summarized as follows. As long as both outputs of the flip-flop are in stable states, one of the two gates in the XNOR circuit will always be ON, causing a voltage drop across the load and a low output. As outputs Q and $\overline{Q}$ start a new transition and their respective output

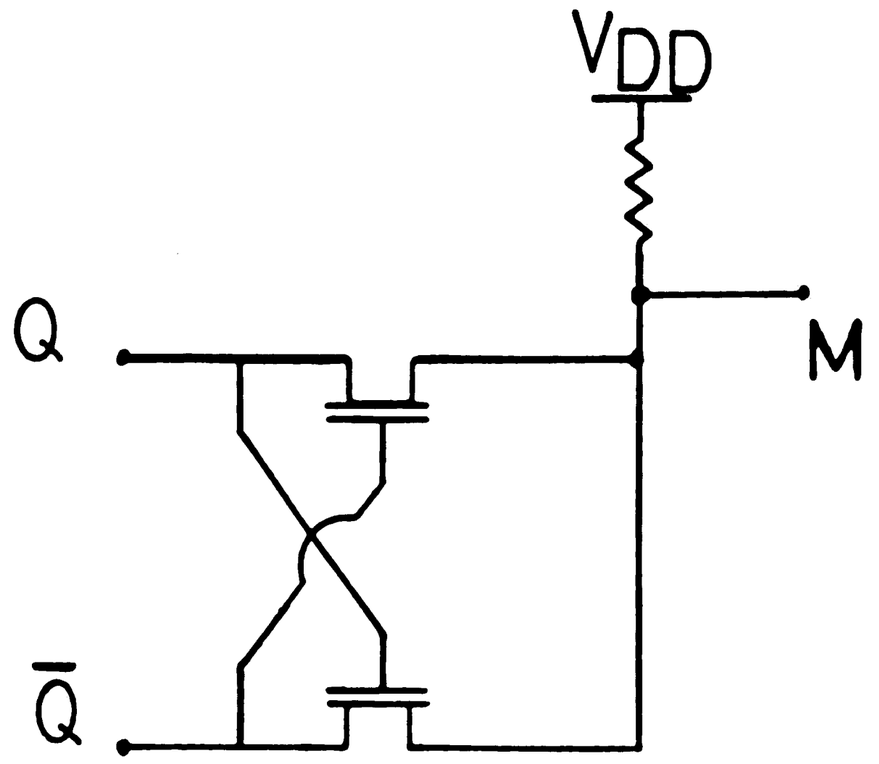

Figure 2.9. Metastable state detecting XNOR circuit [18].

voltages become closer to each other, at some point in time they will come to within less than the threshold voltage of the input gates of the metastable state detecting XNOR circuit. At this point in time the output will rise and display a logic 1 to indicate a metastable state has occurred in the flip-flop circuit.

This particular circuitry has some shortcomings in the sense that for transitions where no metastability has occurred the detecting circuit will generate a runt pulse. This is due to the fact that during each transition at the outputs of the flip-flop there exist a crossover point where the outputs Q and $\bar{Q}$ will have the same value. However, the detection circuitry will display an output of 1 from the

time when the outputs are within V_T volts of one another and becoming closer, until the time where the difference between Q and $\bar{Q}$ is V_T volts and increasing. This phenomenon may not be of any concern to a designer if the transition time of the flip-flop under the test is very fast. However, this output will become larger as the transition time between states becomes longer.

In the clock scheme suggested by Stucki and Cox [18], this event should pose no hazard to proper functioning of the system since a pause functions as a delay given the condition that metastability in the flip-flop exceeds the end of a clock cycle. Another drawback in this scheme is that since flip-flop response time has no absolute bound it is possible for the operation of the clock to be suspended for an unacceptably long time. The designer has added a time-out delay circuit to prevent against such an occurrence. The length of the time-out delay is determined by the designer and is set by using proper values for the RC network.

It should be noted that a time-out can restart the clock at a time when a metastable state is still present at the output of the flip-flop that has initiated a pause. As a result there exists a probabilistic possibility that the system may fail due to a premature time-out prior to complete stabilization of the flip-flop.

2.4 A CMOS METASTABLE STATE DETECTING CIRCUIT

In this section a new circuit is presented which is suitable for a circuit design prior to fabrication of a flip-flop. The circuitry used to improve the reliability of a flip-flop should be accounted for at design time along with the flip-flop since the circuit specifications depend on the flip-flop's properties and must be built in on the chip.

Consider the voltage transfer characteristic of a flip-flop as shown in Figure 2.10. Further consider the fact that metastability occurs at the voltage V_m in CMOS flip-flops. If this voltage level is used as an input to a NAND gate that has a voltage transfer characteristic such that its midpoint voltage V_{m1} is less than V_m then the value of V_m will be translated as a logic 1. Similarly if the same value V_m be used as an input to a different NAND gate for which V_{m2} is greater than V_m , then the value V_m will be translated as a logic 0.

Now consider a NAND gate that has a voltage transfer characteristic for which V_{m1} is less than V_m. We use the outputs Q and $\overline{Q}$ of a flip-flop whose midpoint voltage is V_m, as the two inputs to this NAND gate where $V_{m1}<V_m$ by some margin that is determined by the designer. For the given circuit if a metastable condition arises where the output values $Q=\overline{Q}=V_m$ the NAND gate will see the output values of the flip-flop as logic 1 and thus will produce an output of 0. At any

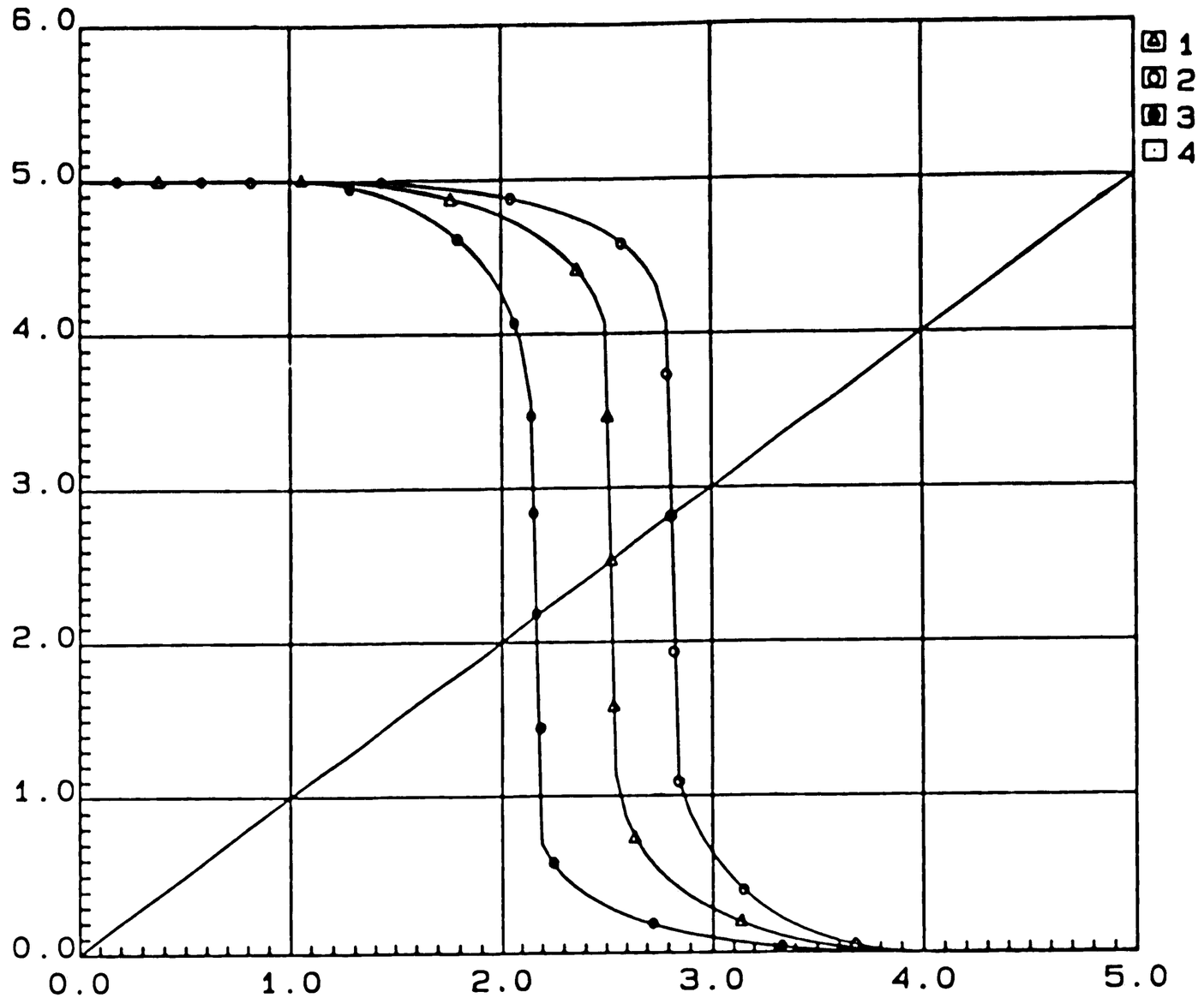

Figure 2.10. Output (volts) vs. input (volts). Voltage transfer characteristics of three NAND gates. Curve 1, 2 and 3 have midpoint voltages V_M, V_{M2} and V_{M1} respectively.

other time where the output values of the flip-flop are complementary the output of the NAND gate will be a logic 1. Thus a logic 0 at the output of the NAND gate can be used as an indicator of a metastable state at the output of the flip-flop.

We can use the signal produced by the NAND gate and improve the flip-flop circuit by using it as an output disable signal as shown in Figure 2.11. It should be noted that this signal may be used in other ways as suited in a particular application. In this case we design the circuit such that for the condition where a non-stable output is

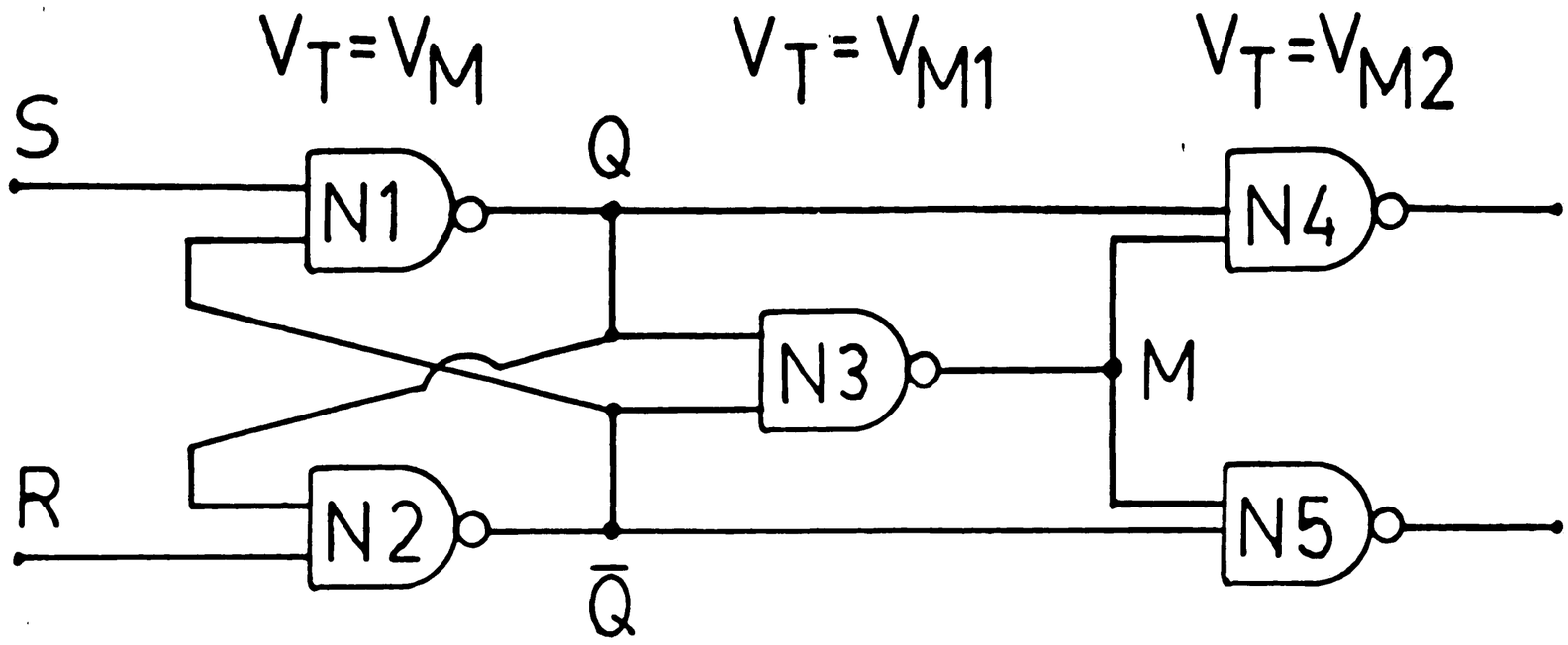

Figure 2.11. Logic diagram for a metastable state detecting circuit with output disable gates.

present at the flip-flop we would keep both outputs at 1 until they have resolved to a stable state at which time we would release the outputs Q and $\bar{Q}$.

In order to obtain the above function we use each of the outputs Q and $\bar{Q}$ as inputs to two NAND gates where the second input will be the output of the metastable state detecting NAND gate. If we use a NAND gate midpoint voltage V_{m2} where $V_{m2} > V_m$ then the only time a stable output is available is when the flip-flop is in a stable state.

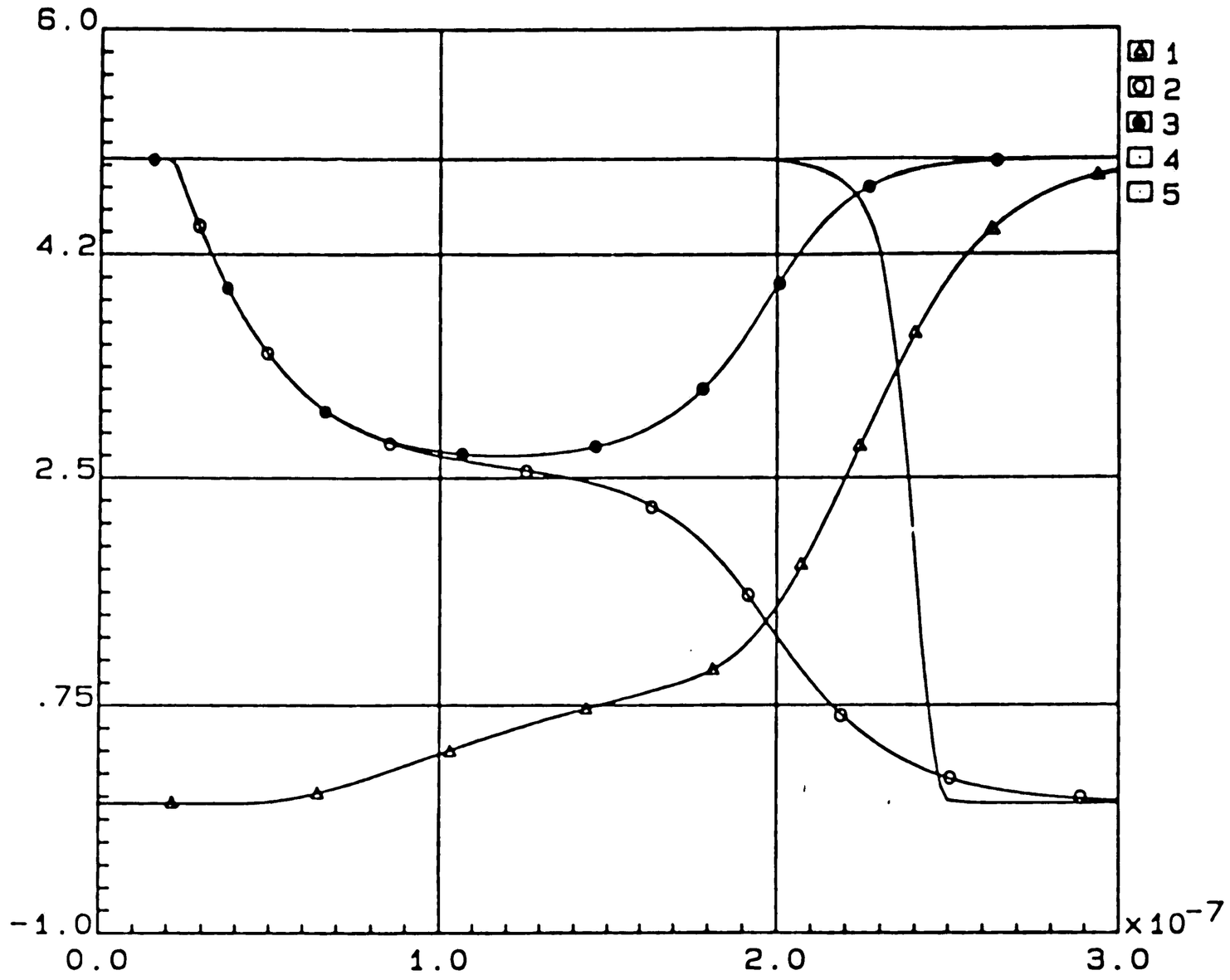

Figure 2.12. Output (volts) vs. time (seconds). Simulation result of a NAND type metastable state detecting circuit. Curves 2 and 3 are the outputs of a flip-flop. Curve 1 is the MSS indicator signal.

The circuit described above is suitable for asynchronous arbiters and input synchronizers since the response time of the circuit is determined by the time the flip-flop at the input stage may spend in the metastable state. However, since all the circuitry proposed for this configuration is internal to a chip, it is estimated that the flip-flop would have much smaller load capacitance and much smaller transition time than flip-flips without this circuit. Therefore, the delay that is experienced at the output of such circuitry will always be comparable or less than the delays observed in flip-flops without a disable function. The simulation results of

such circuit is shown in Figure 2.12 and verifies the functionality of this circuit.

3. CORRECTIVE CIRCUITS

In thc previous chapter two different circuits were presented to detect metastability in a flip-flop. Although circuits can be designed based on either one of the detecting circuits in order to prevent the appearance of a metastable state at the output of a flip-flop, neither circuit is capable of providing some means to avoid entering a metastable state in a flip-flop. In this chapter an attempt has been made to prevent metastability in flip-flops without imposing any constraints on the inputs to the flip-flop.

3.1 OUTPUT CHARACTERIZATION

It is desired to design some circuitry to prevent a flip-flop from entering its metastable state. In order to do so we need to know more about the output characteristics of a flip-flop under input conditions that cause a flip-flop to enter its metastable state. Consider an S-R flip-flop consisting of two NOR gates as shown in figure 3.1(a). The CMOS representation of such flip-flop is shown in figure 3.1(b). We can create an input condition for this flip-flop such that it would enter its metastable state. The difficulty in characterizing such input condition arises from the fact that the input used is an output of some other gate itself. Thus we need to use some well defined inputs for which the flip-flop will produce similar output characteristics as for the case where the input used is a pulse from another gate.

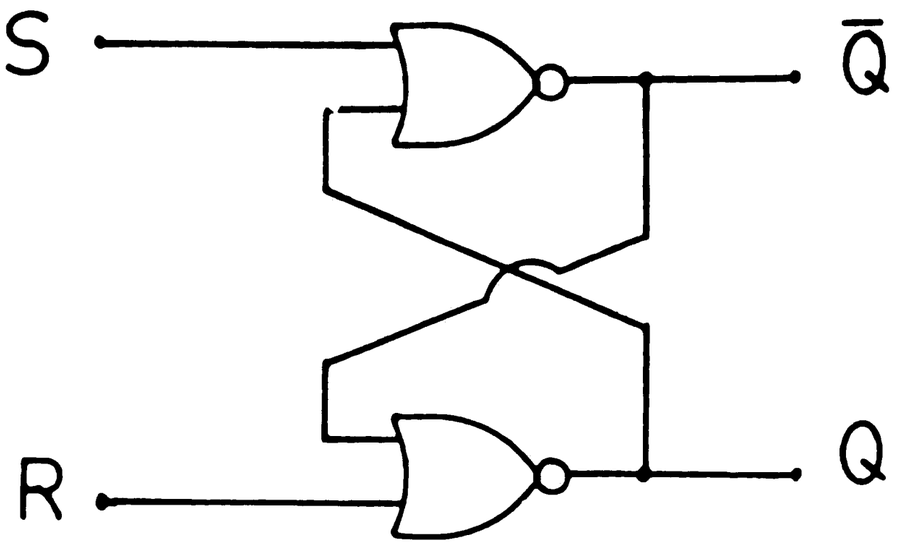

Figure 3.1(a). A NOR type S-R flip-flop.

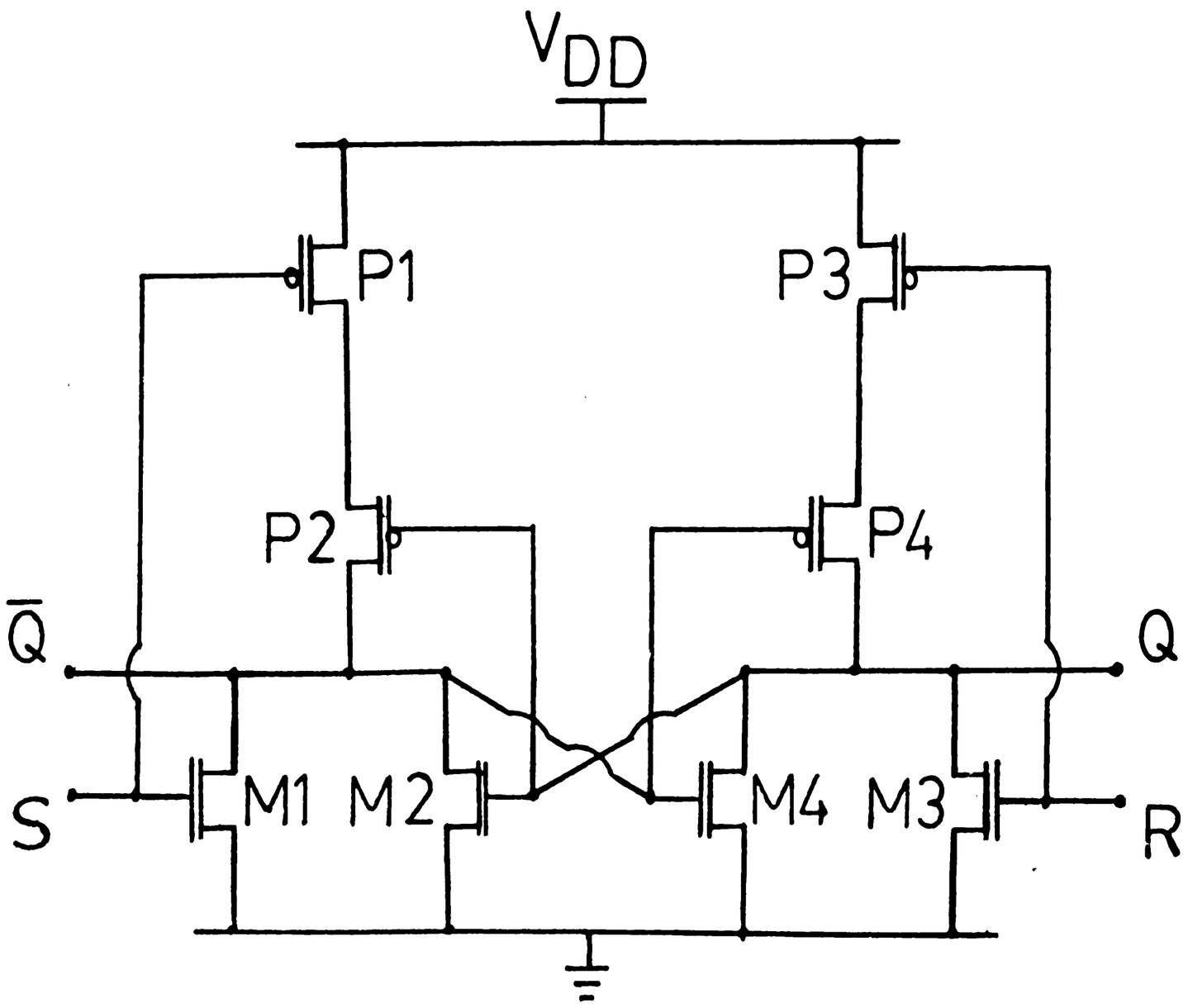

Figure 3.1(b). CMOS representation of a NOR-NOR S-R flip-flop.

Since our intent is to characterize the output behavior at this stage we need to create metastability at the output terminals as explained here. Consider an S-R flip-flop as shown in figure 3.1(b). If we hold both set and reset inputs of this flip-flop at logic 1, the outputs Q and $\bar{Q}$ will be at logic 0. By dropping both inputs to logic 0 simultaneously, the outputs Q and $\bar{Q}$ will start a transition from logic 0 and gradually reach a metastable state. This is due to the fact that as long as both inputs are high transistors M1 and M3 are on and transistors P1 and P3 are off. Therefore M1 and M3 will create a path to ground and thus outputs Q and $\bar{Q}$ will be low. This in turn will keep M2 and M4 off and P2 and P4 on.

As soon as the set and reset inputs are lowered to logic 0, transistors M1 and M3 will be off as well as M2 and M4. In turn P1 and P3 will be on along with P2 and P4.This will result into a gradual charge build-up at the outputs Q and $\bar{Q}$ at a rate that is directly proportional to the time constant of the flip-flop (τ=RC). As the charge continues to build up at the outputs, transistors M2 and M4 will gradualy begin to turn-on, whereas transistors P2 and P4 are becoming less and less conductive. At some point in time the voltage level at the outputs Q and $\bar{Q}$ will reach a point such that the conduction in transistors M2 and M4 will be equal to the conduction in their corresponding PMOS gates, P2 and P4. If we assume a symmetrical flip-flop, i.e., both NOR gates have similar dimentions and I/O

characteristics to one another, the equilibrium in voltage level and conduction will occur simultaneously at Q and $\bar{Q}$. Thus the flip-flop will be in an unstable equilibrium or metastable state until some nonlinear properties in the flip-flop drive it out of that state.

The total amount of charge build-up at the output terminals Q and $\bar{Q}$ at the time of unstable equilibrium is:

$$Q_Q = \int_{t_1}^{t_2} I_{P2}(t)dt - \int_{t_1}^{t_2} I_{M2}(t)dt \qquad (3.1)$$

$$Q_{\bar{Q}} = \int_{t_1}^{t_2} I_{P4}(t)dt - \int_{t_1}^{t_2} I_{M4}(t)dt \qquad (3.2)$$

where the limit t_1 is the time prior to the high to low transition at the set and reset inputs. The limit t_2 is the time when conduction in transistor P2 is equal to M2 and similarly conduction in transistor P4 is equal to M4. Equations 3.1 and 3.2 are obtained through numerical evaluation of simulation results. If we assume that the load condition at output terminals are the same, then the total charge stored at Q and $\bar{Q}$ will be the same at the time when flip-flop reaches its unstable equilibrium state, and therefore we have:

$$Q_Q = Q_{\bar{Q}} = C_{tot}.V_M \qquad (3.3)$$

It can be seen that the total charge stored during an unstable equilibrium state is directly proportional to the total load

capacitance of a flip-flop. Therefore an accurate calculation of C_{tot} will enable us to calculate Q_Q and $Q_{\overline{Q}}$ and vice versa.

3.2 INPUT CHARACTERIZATION

We would like to investigate the input conditions for which the output of a flip-flop will enter a metastable state. Consider an S-R flip-flop consisting of two NOR gates as shown in figure 3.1(a). Moreover, assume that the flip-flop is reset, i.e., the output $\overline{Q}$=1. We must find a pulse with certain magnitude and width that has enough energy to cause metastability at the output of this flip-flop. Several pulses were used that contained this amount of energy to leave the flip-flop in its half-set state.

All pulses used appeared to have equal amount of energy above a certain threshold. Moreover, no transition was observed at the output of the flip-flop for input pulses that had a magnitude equal to V_M volts or less. Figures 3.2 through 3.5 show the simulation results of the output behavior of this flip-flop in response to various input conditions.

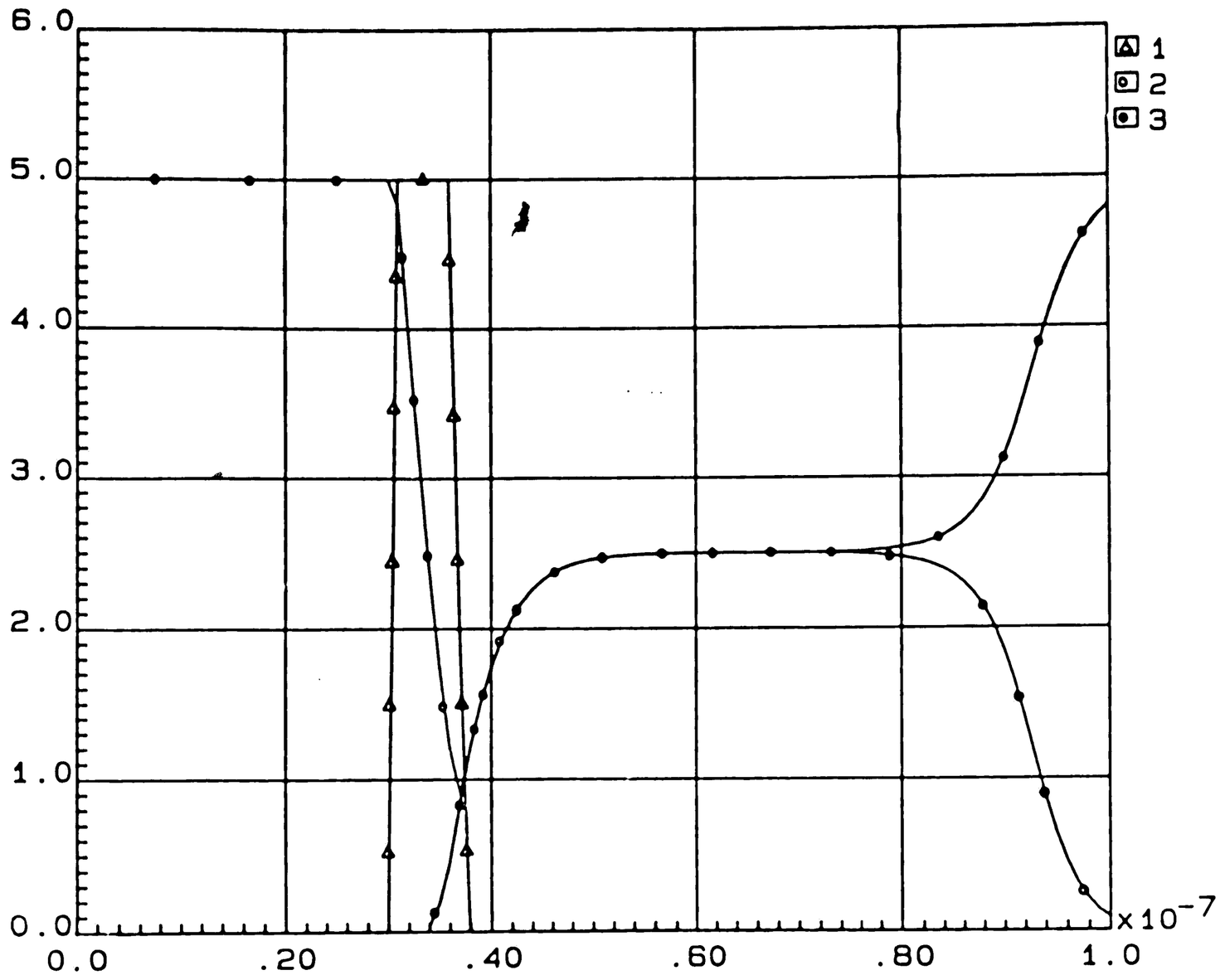

Figure 3.2(a). Output (volts) vs. time (seconds). Curve 1 is the input to S terminal. Curves 2 and 3 are the Q-bar and Q outputs.

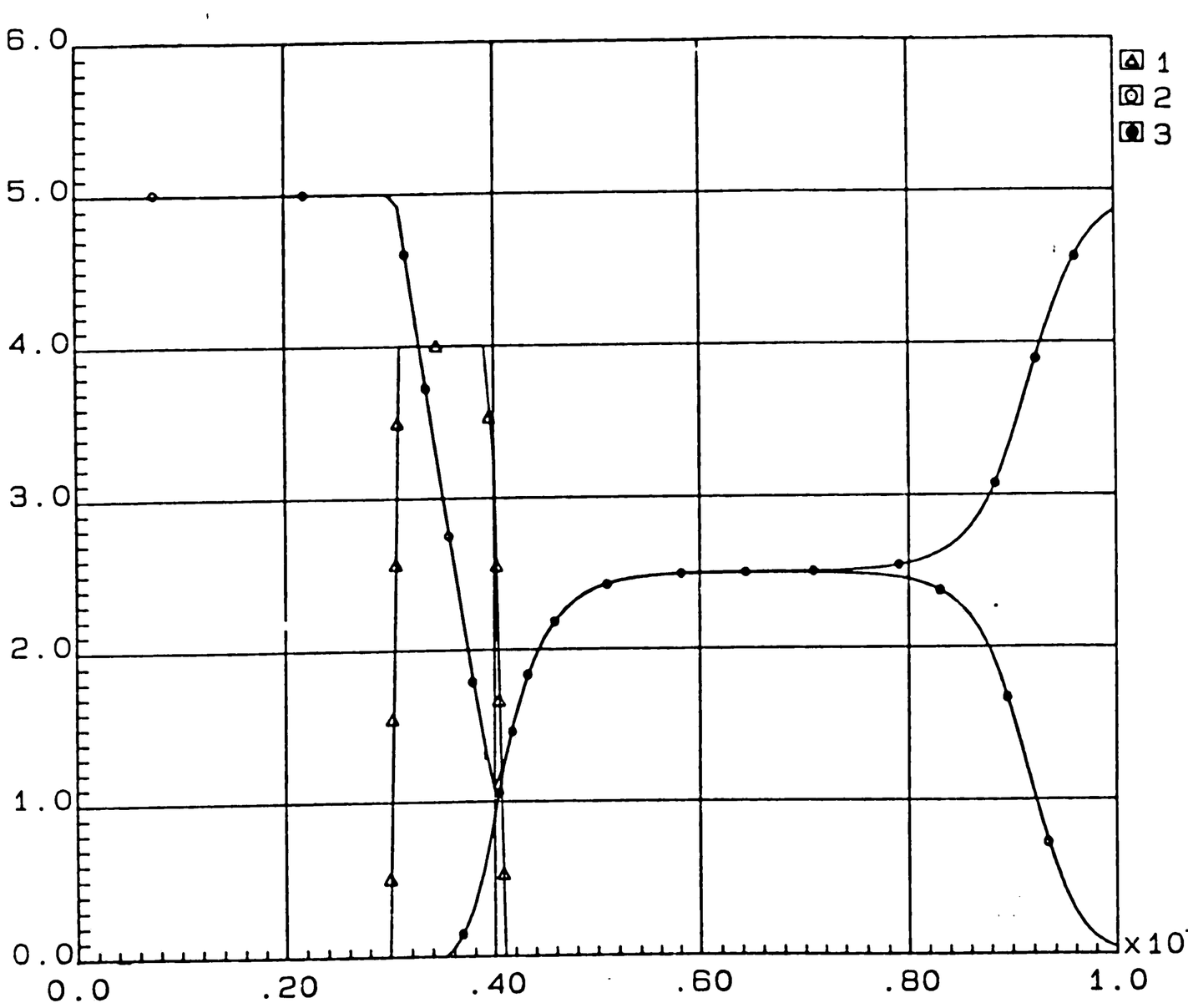

Figure 3.3(a). Output (volts) vs. time (seconds). Curve 1 is the input to S terminal. Curves 2 and 3 are the Q-bar and Q outputs.

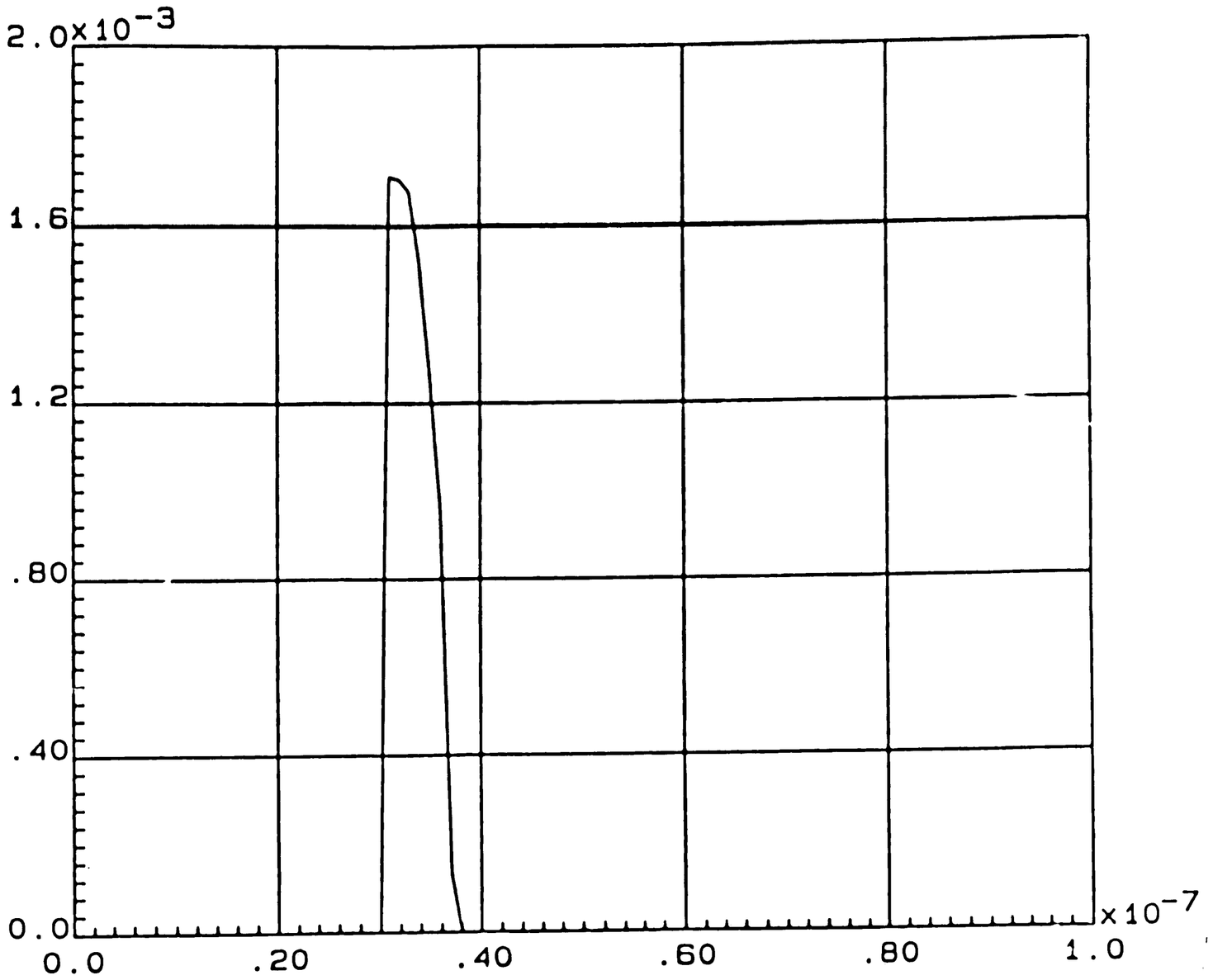

Figure 3.2(b). Current (amps) vs. time (seconds). Current through gate M1 for input as shown in figure 3.2(a).

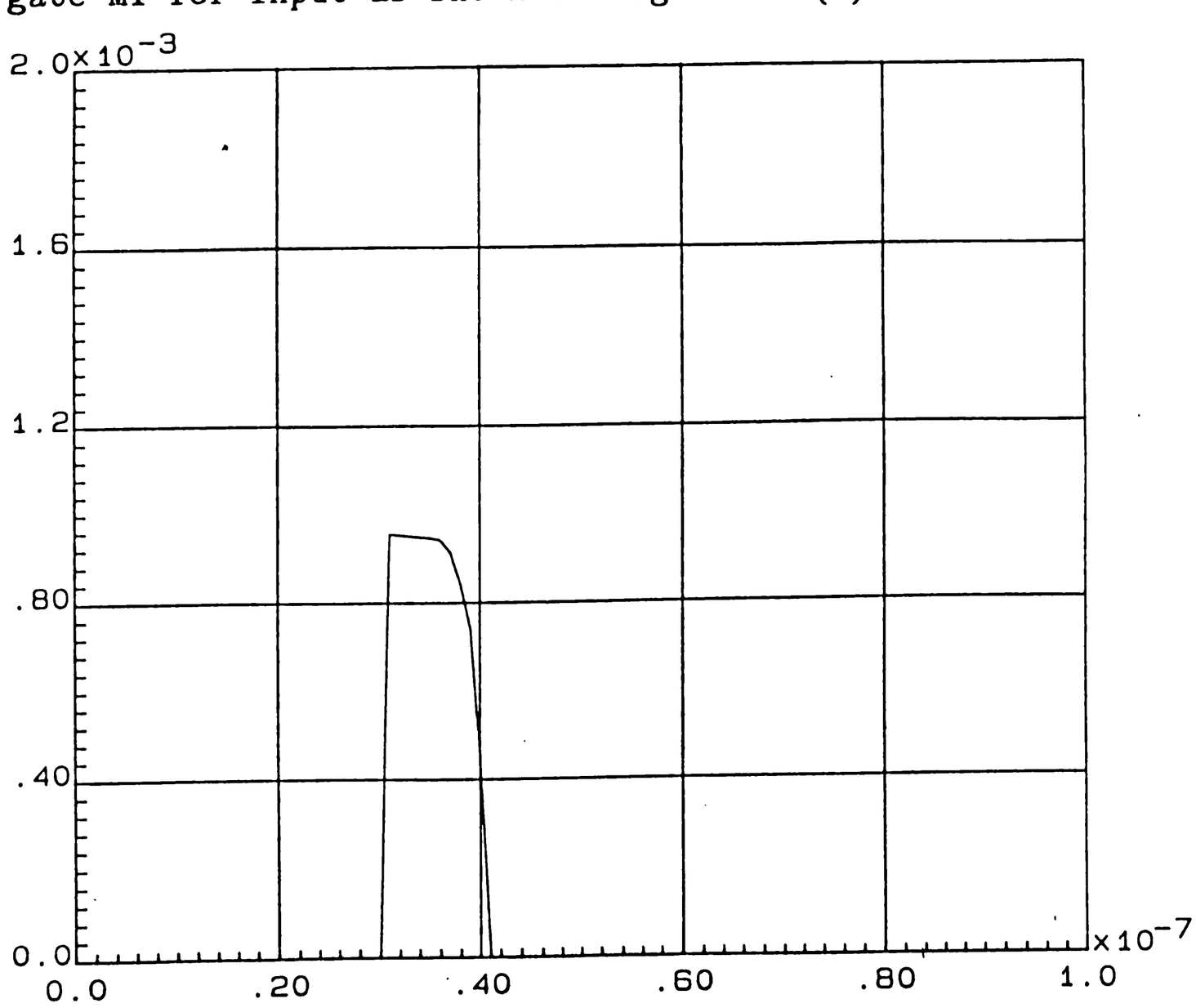

Figure 3.3(b). Current (amps) vs. time (seconds). Current through gate M1 for input as shown in figure 3.3(a).

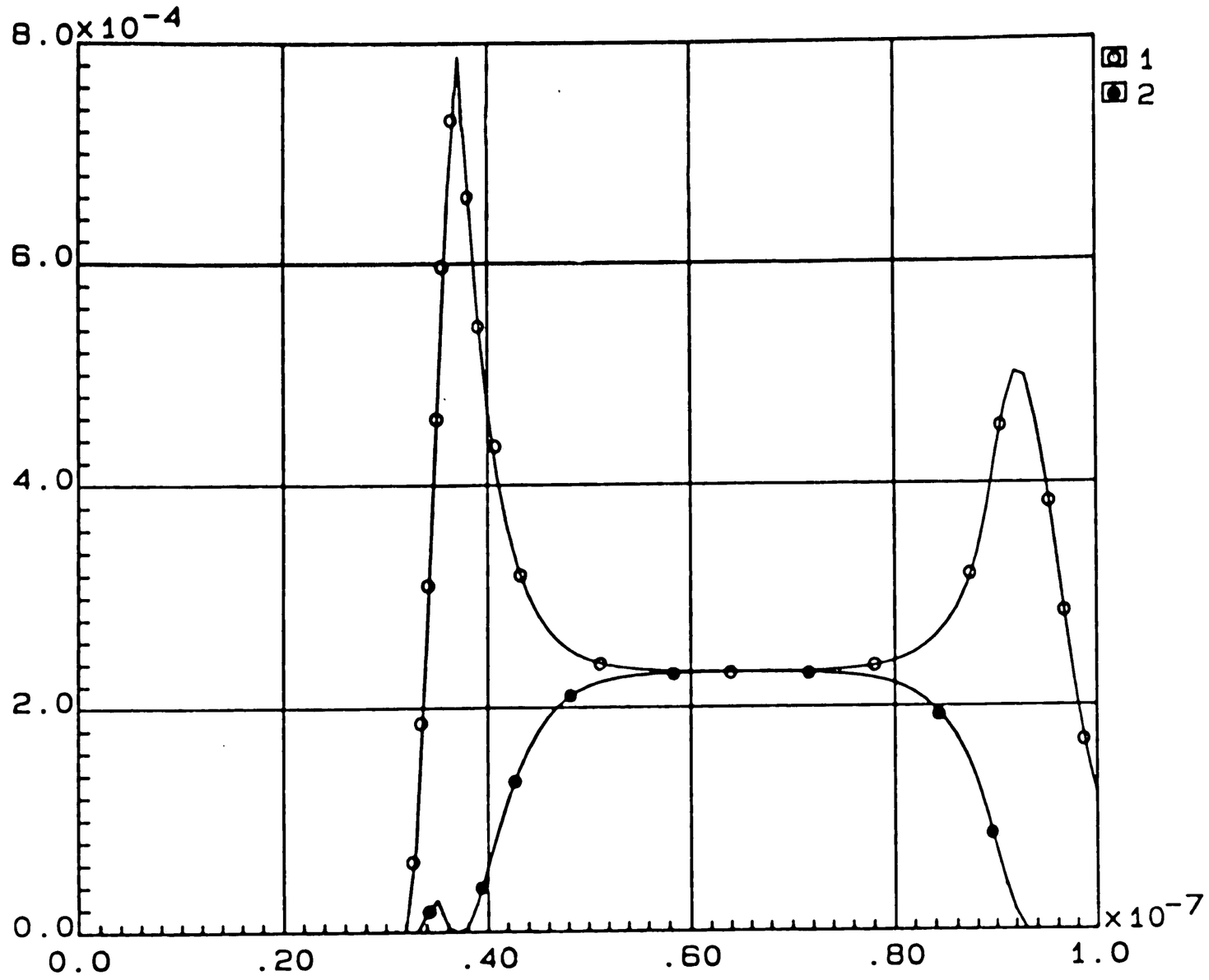

Figure 3.2(c). Current (amps) vs. time (seconds). Curve 1 and 2 are currents through gates P4 and M4 for input as shown in 3.2(a).

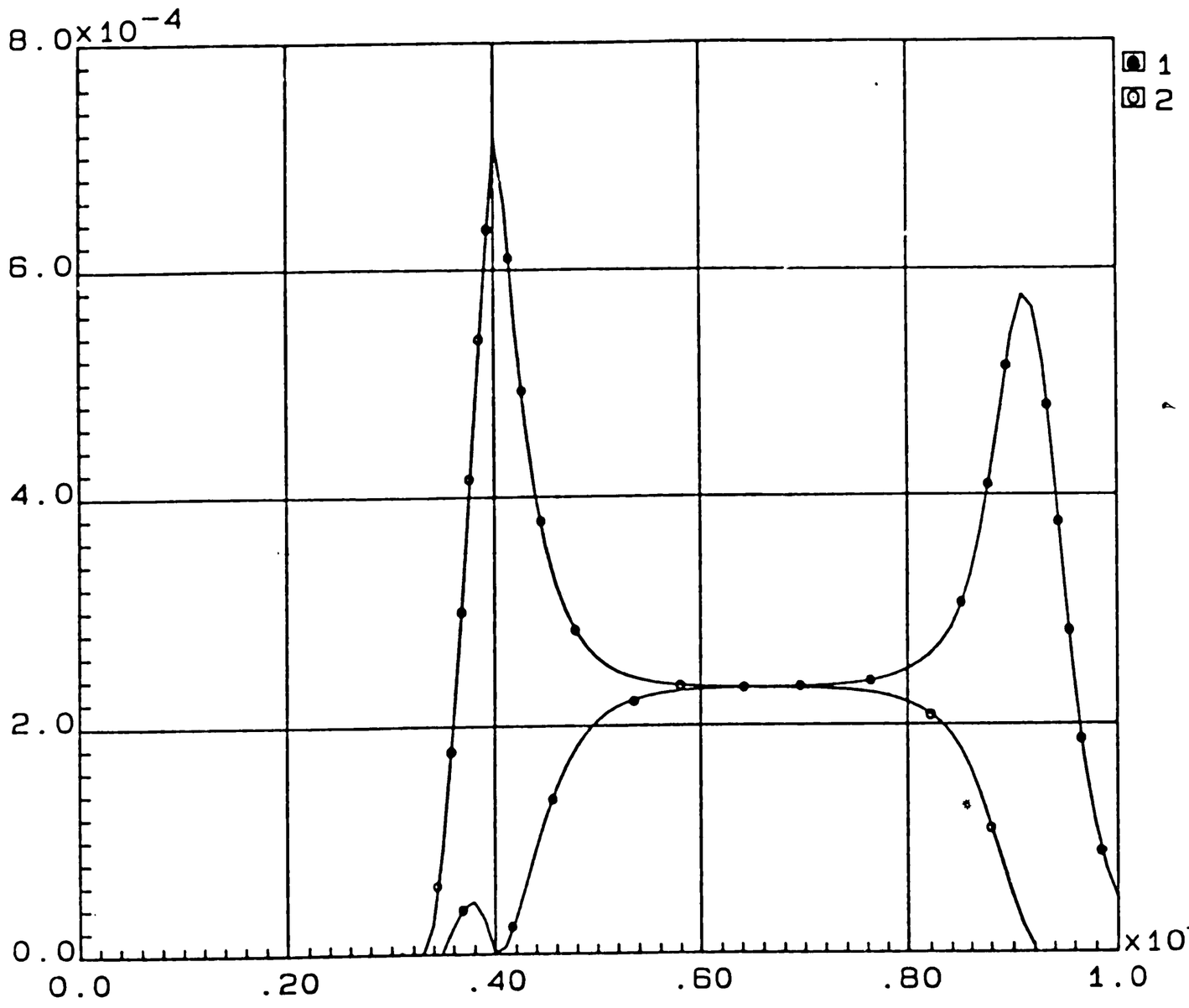

Figure 3.3(c). Current (amps) vs. time (seconds). Curves 1 and 2 are currents through gates P4 and M4 for input as shown in 3.3(1).

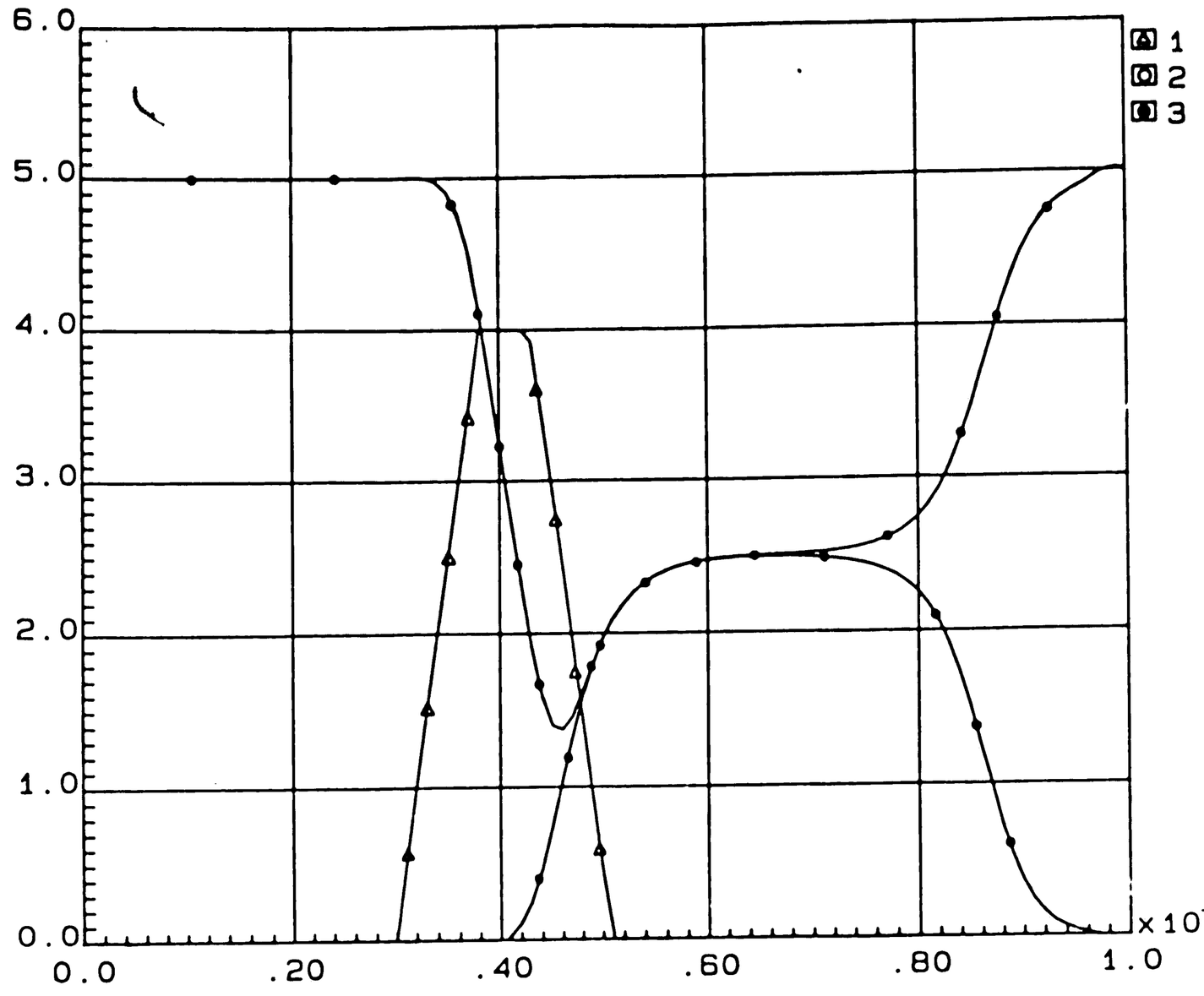

Figure 3.4(a). Output (volts) vs. time (seconds). Curve 1 is the input to S terminal. Curves 2 and 3 are outputs Q-bar and Q.

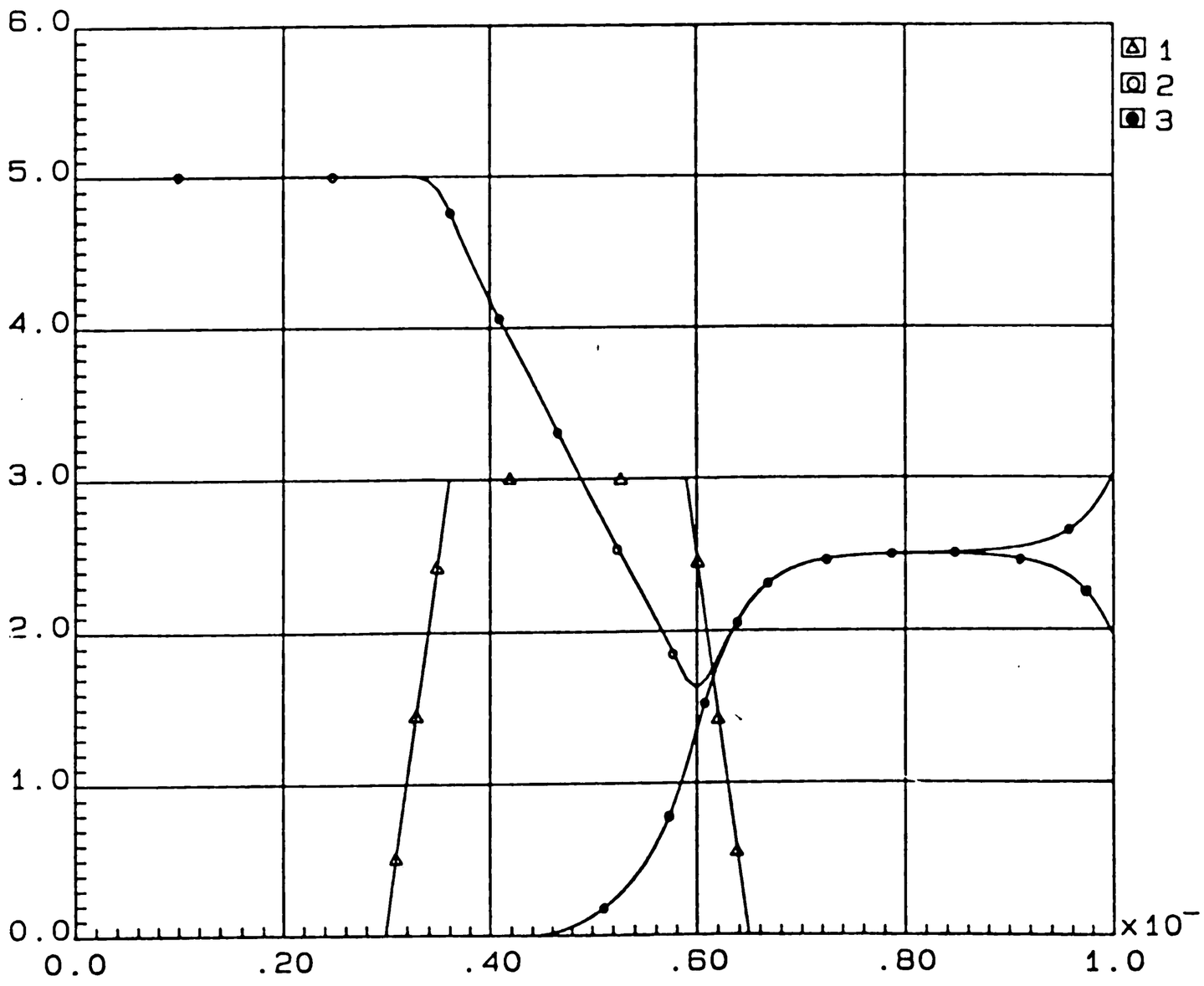

Figure 3.5(a). Output (volts) vs. time (seconds). Curve 1 is the input to S terminal. Curves 2 and 3 are outputs at Q-bar and Q.

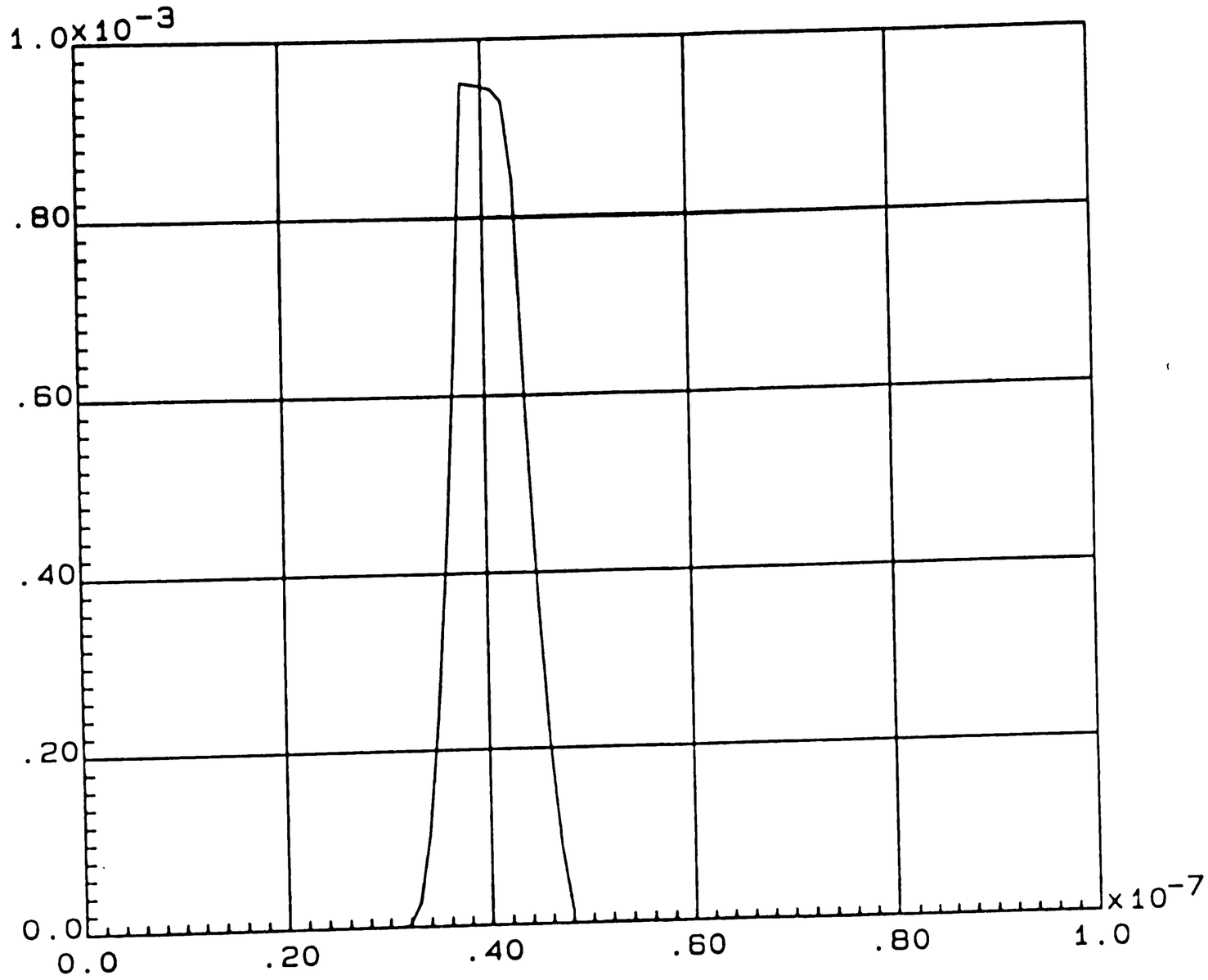

Figure 3.4(b). Current (amps) vs. time (seconds). Current through gate M1 for input as shown in figure 3.4(a).

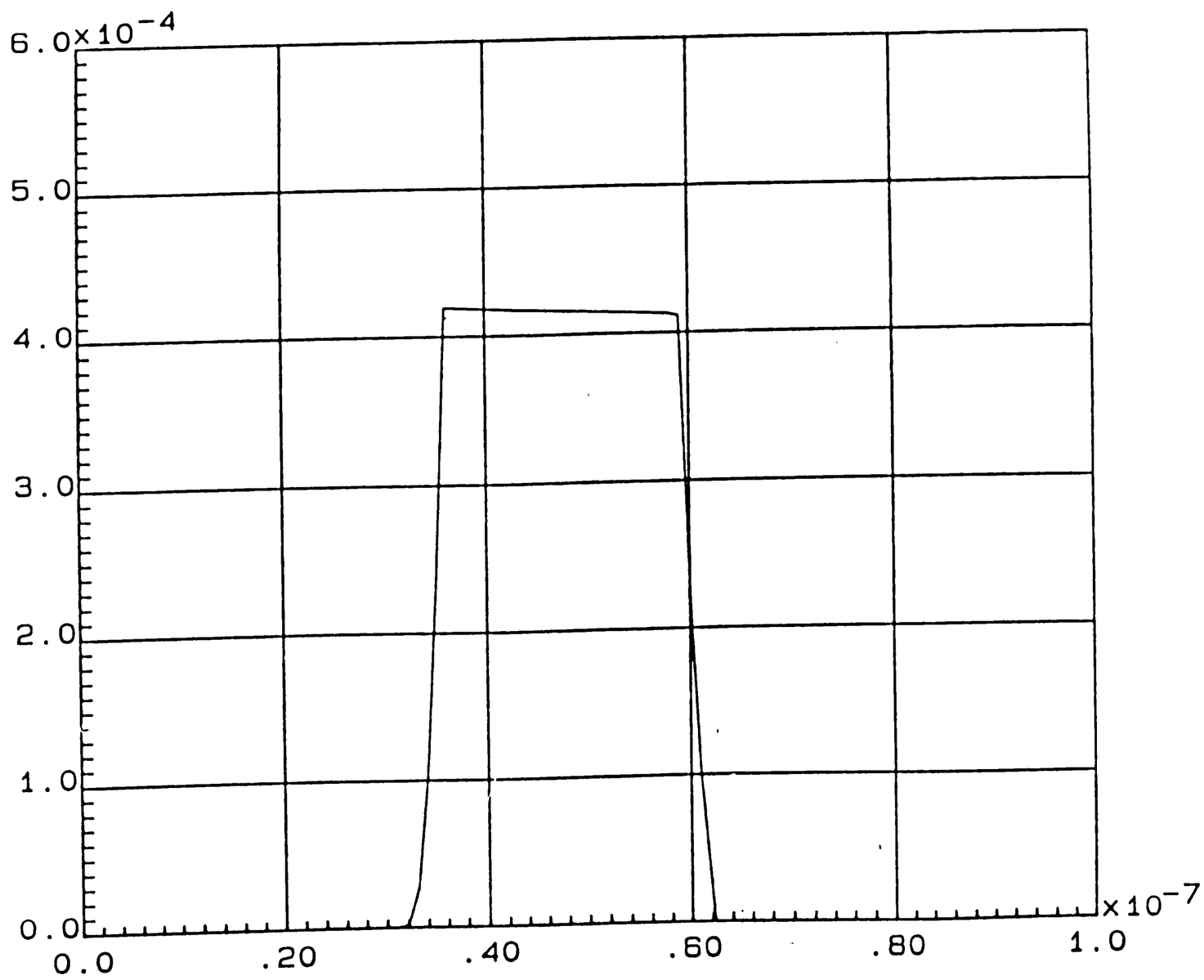

Figure 3.5(b). Current (amps) vs. time (seconds). Current through gate M1 for input as shown in figure 3.5(a).

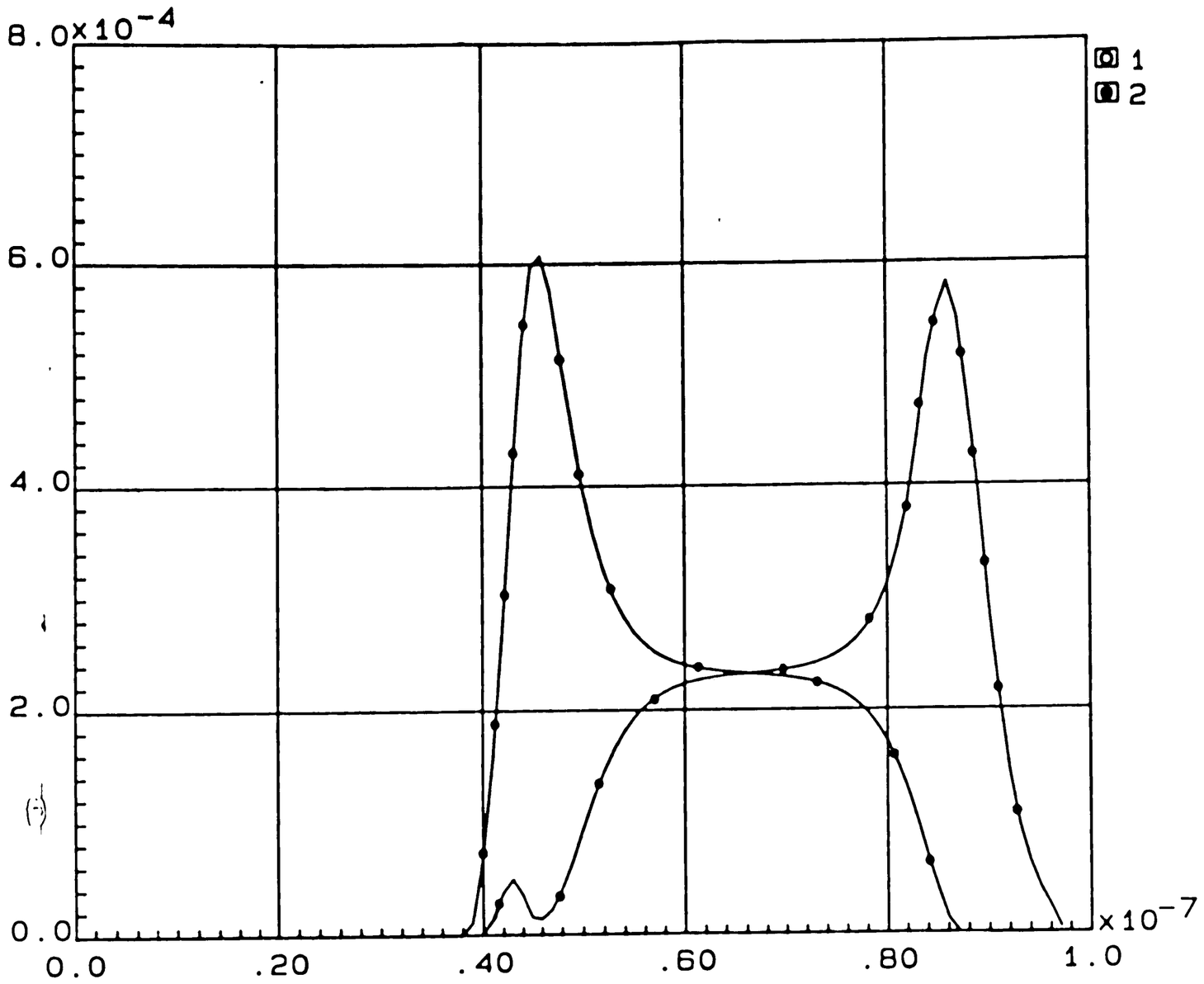

Figure 3.4(c). Current (amps) vs. time (seconds).

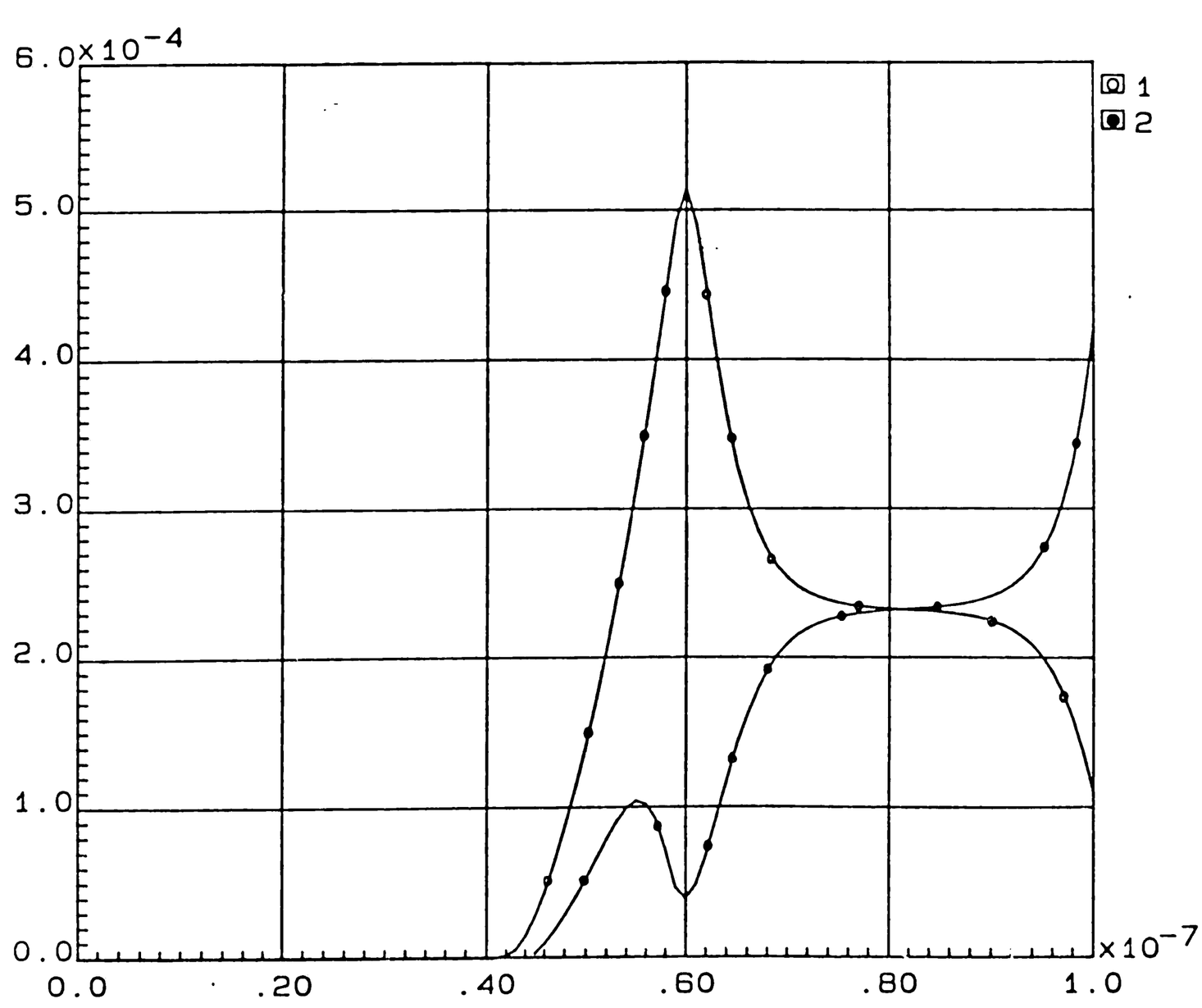

Figure 3.5(c). Current (amps) vs. time (seconds).

3.3 A RUNT PULSE REMOVING CIRCUIT

It is known that extended periods of metastability is caused by inputs to a flip-flops that have certain amount energy above a flip-flop midpoint voltage V_m. This energy level can be defined through simulation results and is independent of the shape of the input. It is intended to design a circuit that would prevent a flip-flop from entering a metastable state regardless of what form of input was used to trigger the flip-flop. The circuit must function such that when the signals used as inputs to the flip-flop have insufficient amount of energy to trigger the flip-flop to a SET or RESET state they should be removed from the inputs to the flip-flop. On the contrary, all pulses that have enough energy to SET or RESET a flip-flop, without causing an extended metastable state should pass through the circuit.

The requirements stated above can be implemented using a circuit as shown in Figure 3.6. The circuit consist of a three input majority decoder that would have an output of 1 whenever any two out of the three inputs are high. The output will remain low for any other input combinations. The input to a flip-flop is used as input A as the first entry. A delayed version of the first input which we would call B, is used as the second entry and finally a feedback from the output back to third input C, would complete the circuit. The amount of the delay at the second input B is determined by the designer and is dependent on the flip-flop properties and load characteristics.

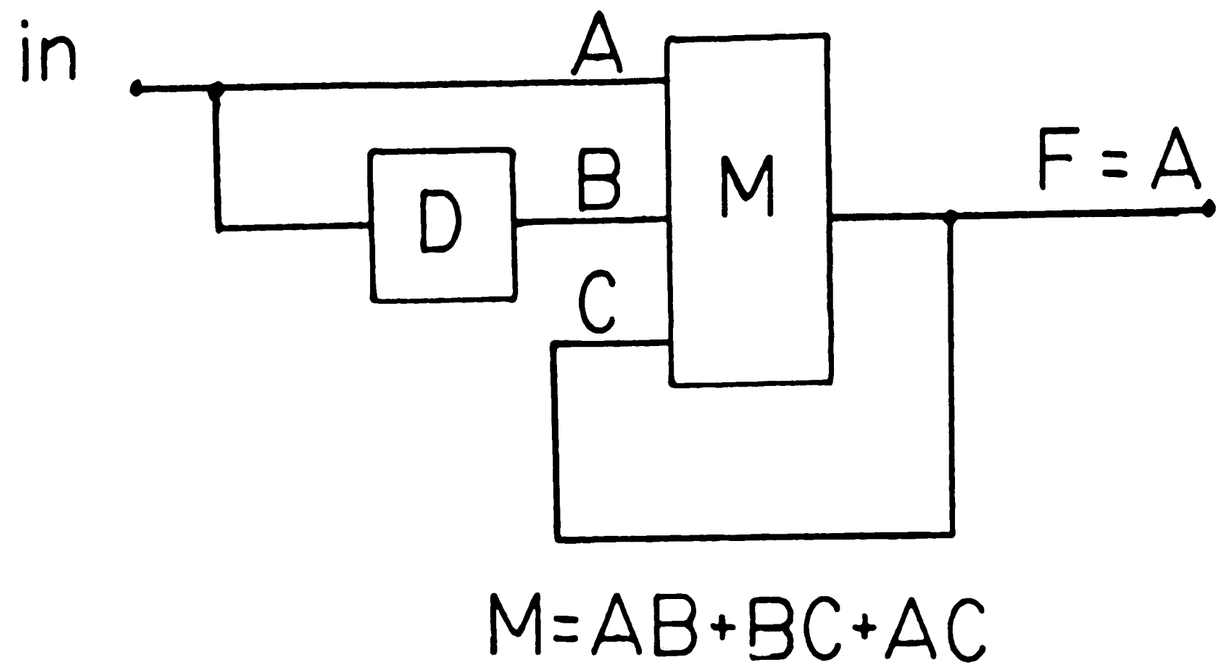

Figure 3.6. Logic diagram of a runt pulse removing circuit.

When an input at terminal A goes from a logic low to a logic high, the output will remains at zero. After a delay of D when the input appears at node B the output will go from a logic low to a logic high only if the input at A is still high. As soon as the output F becomes high the input C will become high too. At this time if the input is removed from A the output will remain high for another D seconds after the input A is removed, since inputs B and C are still forming a majority.

In this configuration if the signal at A goes low before B goes high there will not be a majority and thus the circuit will ignore A . Therefore by setting the delay D such that it is equal or greater than the minimum pulsewidth that is desired at the inputs of a flip-flop,

we are assured that there will be no output signal from the circuit that has a pulsewidth less than that of the desired pulse. Moreover, since this pulsewidth is set by varying the delay D and is based on the flip-flop characteristics it will always have enough energy to SET or RESET the flip-flop.

There are some advantages and disadvantages in using this configuration in order to avoid metastability in flip-flops. The first and most important is that regardless of what the input conditions are, every input will be delayed for a time period of D seconds plus the propagation delay of the circuit. This delay will have a negative effect on the performance of the circuit. The second drawback is that all inputs that have a long enough pulsewidth to go through the circuit will appear at the output with a slightly reduced pulsewidth. The amount of this reduction is about one gate delay and thus the designer must take this loss into consideration. Finally, any change that changes the load capacitance of the flip-flop may require redefining the delay element D. Thus if the circuit described above is realized as an internal or on chip part to the flip-flop it may not be possible to make any changes once the circuit is fabricated.

3.4 A FAULT-TOLERANT CMOS FLIP-FLOP

It is seen that there exists some circuit to prevent a flip-flop from entering a metastable state. However, as described in section 3.3

this circuit exhibits several disadvantages. In this section a new design for a CMOS flip-flop is presented which is shown through simulation results to be able to avoid entering a metastable state regardless of the input conditions with very high probability of success. The circuit diagram of this flip-flop is shown in figure 3.8 and it is described below.

The circuit is comprised of two stages each containing a NAND type S-R flip-flop. The first stage contains a metastable state detecting circuit as described is section 2.4. Each output of the first stage is then used as an input to the flip-flop in the second stage in the fashion shown. The first stage acts as a driver for the second stage. The second stage, however, will have two different phases during its operation, which we call normal phase and memory phase, respectively.

During the normal phase of operation the flip-flop in the first stage is in a stable state such that its outputs are either set or reset. The second flip-flop in this phase will simply respond to the output of the first stage. In the second phase the flip-flop in the first stage is in the metastable or unstable equilibrium state due to some input conditions. The metastable state detecting gate N3 will exhibit a logic 0 output to indicate the occurence of a metastable condition in the first stage. This in turn will cause both N4 and N5 to display a logic 1.

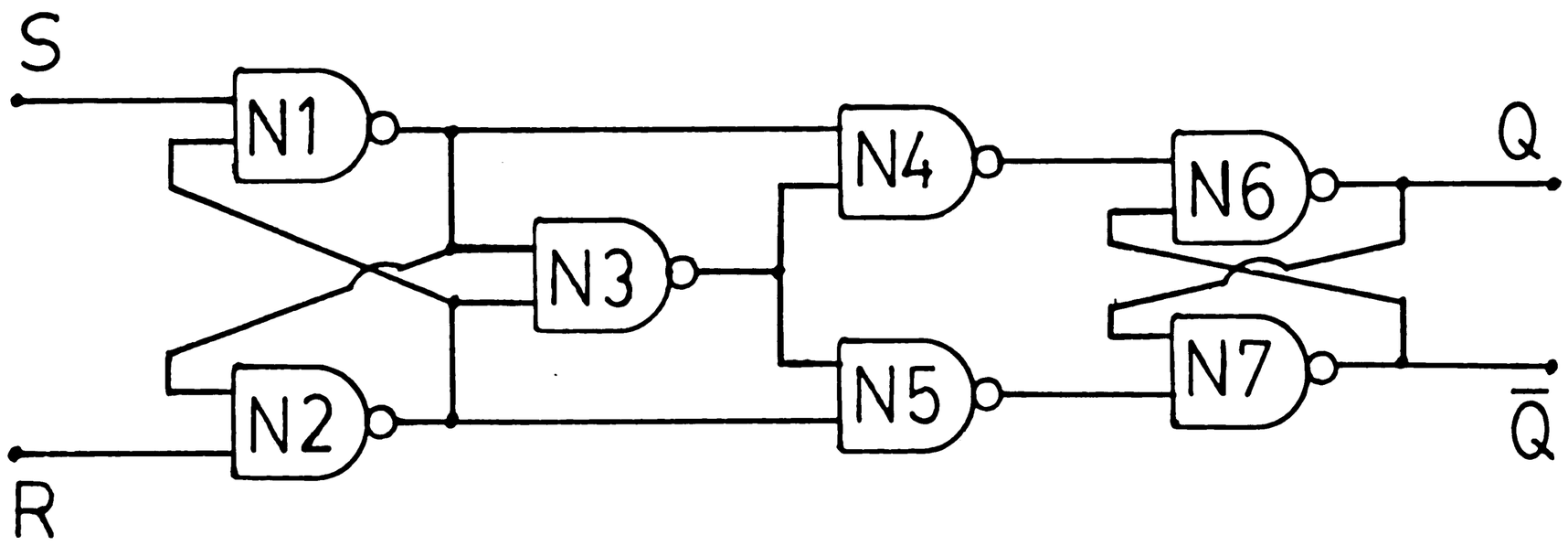

Figure 3.8. Logic diagram of a fault tolerant CMOS flip-flop.

Presence of a logic 1 at the inputs of NAND gates N6 and N7 will cause no change of states at the output terminals, therefore for as long as flip-flop of the first stage is in the metastable state there will be no change or transitions at the output of the second stage. As soon as the first flip-flop comes out of metastable state, one of the outputs at N1 or N2 will go low causing the output of N3 to go high. This output in a NAND combination of a high output (at N1 or N2), will cause a low input at N6 or N7 and will either set or reset the flip-flop of the second stage accordingly.

It is extremely unlikely for a metastable state to occur at the flip-flop of the second stage. This is due to the fact that the only time such an event may occur is when there is a high to low runt pulse at one of the inputs to the second flip-flop. However, the only circumstances under which a runt pulse may appear at one of the inputs of the second flip-flop is if the first flip-flop experiences two consecutive metastable states during a very short period of time. This implies that the mean time between failures of the first flip-flop is in the order of a few tens of nanoseconds. Since all the capacitive load that is seen by the first flip-flop is internal to the chip, it is very unlikely that the first flip-flop stays in an unstable equilibrium state for a long period of time. Moreover, It is possible to improve the MTBF of the first stage by allowing longer settling time for the flip-flop.

It is clear that this flip-flop configuration has several advantages inherent in its structure. For instance, it is possible to

have a mean time between failure of a few hundred nanoseconds in the first stage, while the second stage flip-flop is realizing a correct response without exhibiting a metastable condition. Moreover, the propagation delay of the circuit is only slightly increased, since the added circuit is internal and does not introduce much inceased capacitance. Finally, since a stable output is all that is presented at the outputs of this flip-flop, this configuration is ideal for asynchronous arbiter structures. This feature prevents the occurence of inconsistent interpretation of a logic level among the receiving devices. In the next chapter this flip-flop is used to design a simple asynchronous arbiter module.

4. ARBITER DESIGN

Arbiters are used in multiprocessor systems where they process and resolve the conflict of asynchronous and simultaneous requests by different processors to access a common memory or shared unit. Asynchronous arbiters rely on input synchronizers and flip-flops in order to record any request signal initiated from a processor. Any failure that may occur in the input synchronizers used in such arbiters may adversely affect the performance of such modules. Thus arbiters are prone to failures due to the consequences that may arise in the event of an input synchronization failure.

Resolving multiple and asynchronous requests implies the design of a control mechanism in an arbiter module that will operate under certain criteria as explained below:

a) at most one processor or request signal may be serviced or acknowledged by the arbiter at any time,

b) each and every request signal must be acknowledged by the arbiter in a finite time,

c) initiation of the request signals are entirely random and independent of one another.

The arbiter module described in this chapter satisfies the requirements mentioned above.

4.1 ASYNCHRONOUS TWO-BIT ARBITER

The arbiter described here is a design based on request/grant signaling convention and is specifically intended for the implementation of arbiter trees. The signaling convention used in this arbiter is shown in Figure 4.1 and can be described as follows:

When the grant lines G_0 and G_1 are low, a device may initiate a request signal for the use of a shared unit by raising its request line R_0 or R_1 . The arbiter will in return allocate the shared unit to a particular device by raising its grant line G_0 or G_1 . At the completion of a request cycle the device resets its request line which in turn will cause the arbiter to reset the corresponding grant line. The function described above can be implemented by the circuit shown in Figure 4.2.

The S-R flip-flop used records which device has initiated a request signal and responds by raising its corresponding grant line. When a request is initiated by one of the processors, the input gate of the second port is disabled after a short delay through the inverter. This will prevent the flip-flop from changing state until the original request line has been reset. If the unit is in use at the time a second device requests access to the unit, the second request will not be granted until the unit is released by the other user. In the event that both users initiate continuous and overlapping requests, the unit will be shared on an alternating basis.

In the event that both devices initiate a request simultaneously while the arbiter is idle, the decision will depend upon the

this circuit exhibits several disadvantages. In this section a new design for a CMOS flip-flop is presented which is shown through simulation results to be able to avoid entering a metastable state regardless of the input conditions with very high probability of success. The circuit diagram of this flip-flop is shown in figure 3.8 and it is described below.

The circuit is comprised of two stages each containing a NAND type S-R flip-flop. The first stage contains a metastable state detecting circuit as described is section 2.4. Each output of the first stage is then used as an input to the flip-flop in the second stage in the fashion shown. The first stage acts as a driver for the second stage. The second stage, however, will have two different phases during its operation, which we call normal phase and memory phase, respectively.

During the normal phase of operation the flip-flop in the first stage is in a stable state such that its outputs are either set or reset. The second flip-flop in this phase will simply respond to the output of the first stage. In the second phase the flip-flop in the first stage is in the metastable or unstable equilibrium state due to some input conditions. The metastable state detecting gate N3 will exhibit a logic 0 output to indicate the occurence of a metastable condition in the first stage. This in turn will cause both N4 and N5 to display a logic 1.

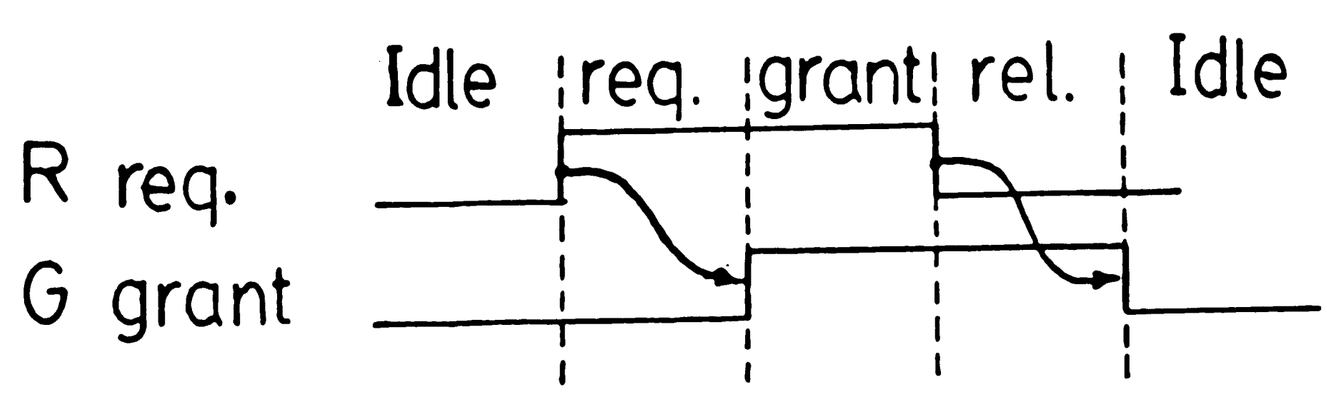

Figure 4.1. Request-grant signaling convention.

Figure 4.2. Two-bit arbiter.

characteristics of the arbiter. Moreover, there exist a hazard in the circuit when simultaneous requests are initiated since there is a possibility that both grant lines g_0 and g_1 will go high. The arbiter configuration, however, will prevent this signal from reaching G_0 and G_1 by using the metastable state detecting flip-flop described in section 3.4. The metastable state detecting gate N5 as shown in figure 4.2, will keep both outputs of gate N6 and N7 at a logic 1 for as long as g_0 and g_1 are high. As soon as either g_0 or g_1 return to zero the output of N5 will go high causing a high to low transition at either N6 or N7. A low input at N8 or N9 will then set the corresponding grant line.

4.2 THE ARBITER MODULE

The two-input asynchronous arbiter module described above can be used in arbiter trees to implement multiinput arbiters as shown in figure 4.3. Each of the arbiter modules consists of a two-input arbiter with an additional set of request (R_c) and grant (G_c) lines to be used in cascade interconnections of the modules. If the cascaded grant line G_c is low, and either of the two request lines into the arbiter module is raised, the cascaded request line R_c will be raised. As soon as the next higher level module responds by raising the cascaded grant line G_c, the lower level module passes the grant signal

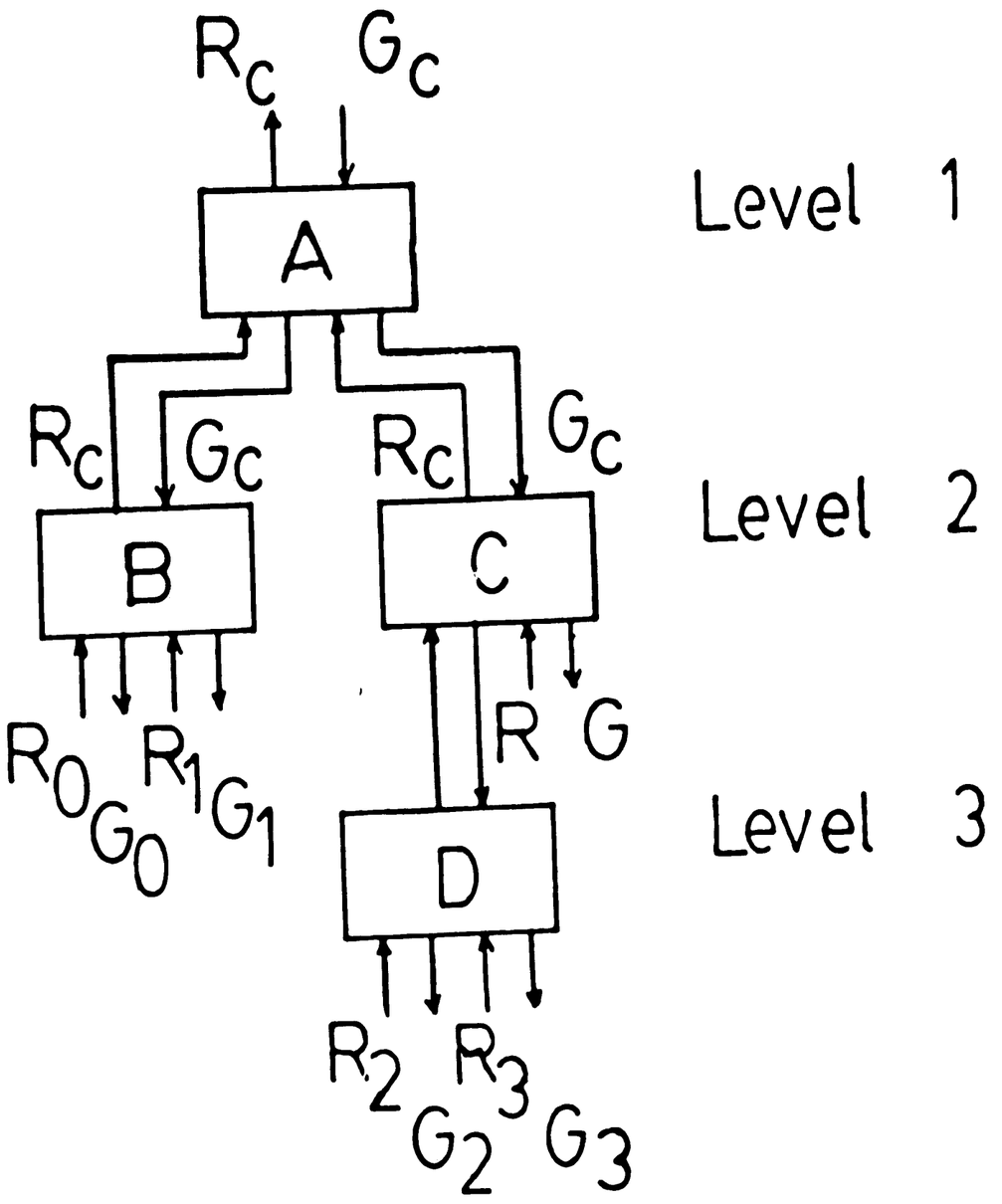

Figure 4.3. An arbiter tree using two-bit arbiter module.

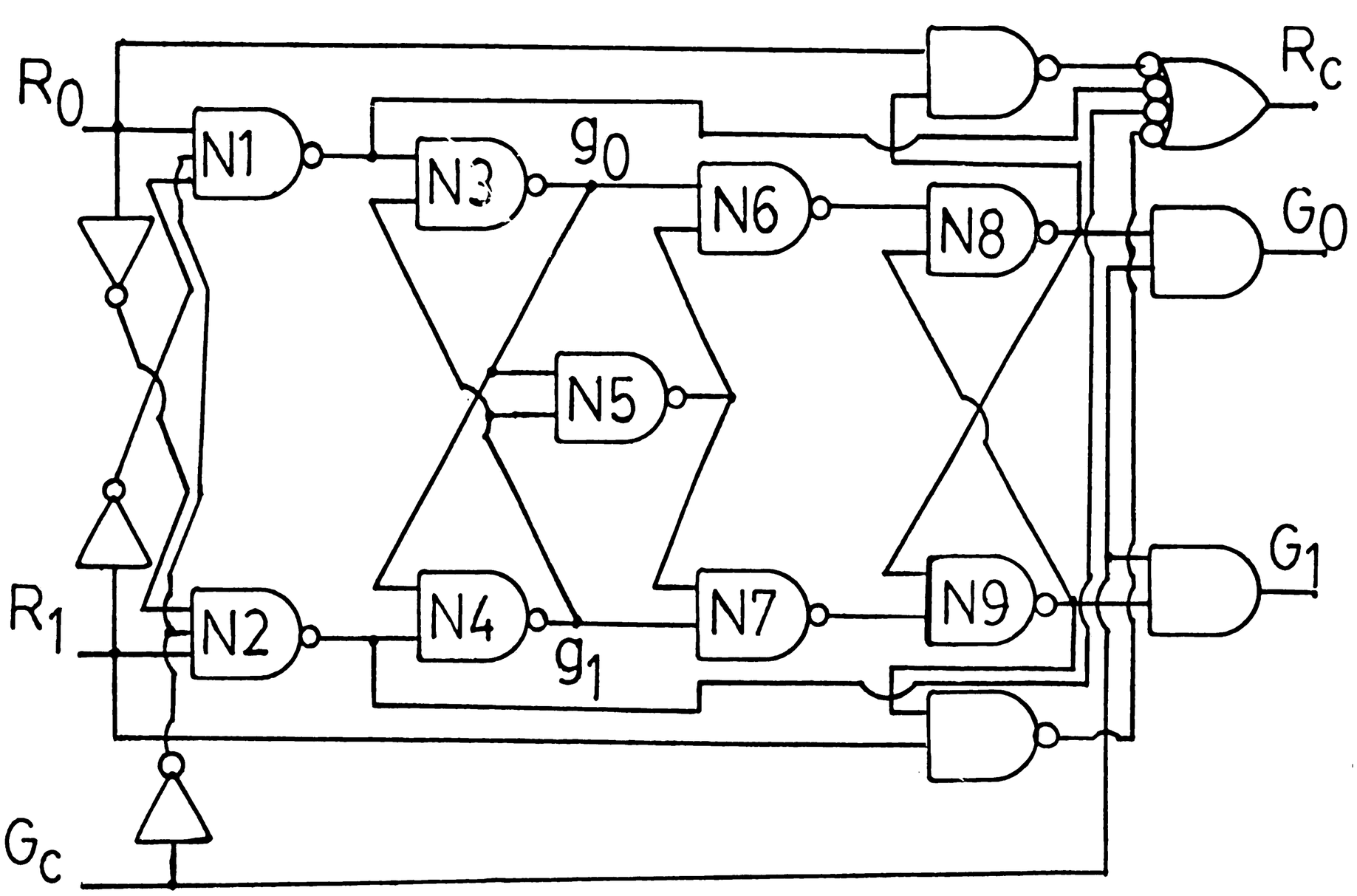

Figure 4.4. Two-bit arbiter module.

back to the requesting device. At the completion of data transfer between the requesting device and the shared unit, the requesting device releases the shared resource by lowering its request line. This in turn causes the arbiter to release the cascaded request line R_c. In the event that another request line is already high, the module waits until G_c is lowered before making a new request by raising R_c. Thus conflicts between two modules at any level is resolved at the next higher level.

A circuit representation of a two input arbiter module in given in figure 4.4, which is an extention of the two-input arbiter shown in figure 4.2. When G_c is low, a logic 1 input at either R_0 or R_1 wiil cause R_c to go high. When the cascade request R_c is acknowledged, i.e. G_c is raised. then either G_0 or G_1 is raised depending on the origin of the request. This signal is sent to the requesting device and the inputs to the module are disabled by G_c to prevent any changes in the flip-flop until the end of data transfer. Upon the completion of a transfer the requesting device will lower its request line R_0 or R_1 and consequently will cause R_c to be lowered. After the cascade grant line G_c is cleared, any new request signal at R_0 or R_1 will start a new cycle.

The arbiter module described here follows the request-grant signaling convention [14]. It is possible to devise a request-acknowledge convention as described by Plummer [14]. In order to realize this convention the grant lines can be ORed to generate a request signal and the grant signal from the shared unit in an AND combination with the grant lines will produce an acknowledge signal.

It should be noted that in an arbiter tree, each module provides equal service to its requesting lines, i.e. both request lines have equal priority . Consider the condition where the shared unit is never idle, and all users generate continous request signals for service. The fraction of the service given to a requesting device will be equal to $(2^n)^{-1}$ where n is the level of arbiter in which the requesting device resides. For example, in figure 4.3 under continous request condition, the tree allocates one-quarter of the service to each of the devices 0,1 and 4 while devices 2 and 3 get one-eighth of service each.

CONCLUSION AND RESULTS

It is seen in this thesis that it is possible to design a synchronizer circuit that has a very high probability of success. This design, however, was only possibles after a thorough analysis of the behavior of flip-flops in the metastable state was performed. It is also shown that other approaches in solving the problem of synchronization are possible. Each approach , however, must base the design upon the technology used or a particular application. Further studies are needed to fully characterize the input condition for which a flip-flop enters the metastable state. After this characterization, it may be possible to provide circuits that perform the synchronization function with zero probability of failure.

The arbiter module represented here is a basic design and there exist several other approaches to design an asynchronous arbiter. It should be noted, however, that there are several approaches to design an arbiter in all of which the synchronization function is performed using some kind of a flip-flop. Therefore, it is essential that a reliable flip-flop be designed and used in the structure of an arbiter, since an arbiter failure is usually attributed to a failure in the synchronization mechanism and eventually the flip-flop used.

REFERENCES

[1] I.Catt ,"Time loss through gating of asynchronous signal pulses," IEEE Trans. Electron. Comput. (Short Notes), Vol. EC-15, pp. 108-111, Feb. 1966.

[2] T.J.Chaney and C.E.Molnar ,"Anomalous behavior of synchronizer and arbiter circuits," IEEE Trans. Comput. (Corresp.), Vol. C-22, pp. 421-422, Apr. 1973.

[3] T.J.Chaney ,"Measured flip-flop responses to marginal triggering," IEEE Trans. Comput., Vol. C-32, pp. 1207-1209, Dec. 1983.

[4] W.Flieschhammer and O.Dortok,"The anomalous behavior of flip-flops in synchronizer circuits," IEEE Trans. Comput., Vol. C-28, pp. 273-276, Mar. 1979.

[5] G.Lacroix, P.Marchegay and G.Piel,"Comments on The Anomalous Behavior of Flip-Flops in Synchronizer Circuits," IEEE Trans. Comput., Vol. C-31, pp. 77-78, Jan. 1982.

[6] D.J.Kinniment and J.V.Woods,"Synchronization and arbitation circuits in digital systems," Proc. IEE, Vol. 123, pp. 961-966 Oct. 1976.

[7] T.J.Chaney ,S.M.Ornstein and W.M.Littlefield ,"Beware the synchronizer," presented at COMPCON-72 IEEE Comput. Soc. Conf., San Francisco, CA., pp. 317-319, Sept. 12-13, 1972.

[8] M.Pechoucek ,"Anomalous response times of input synchronizers," IEEE Trans. Comput., Vol. C-25, pp. 133-139, Feb. 1976.

[9] H.J.M.Veendrick,"The behavior of flip-flop used as synchronizers and prediction of their failure rate," IEEE J.Solid-State Circuits, Vol. SC-15, pp. 169-176, Apr. 1980.

[10] E.G.Wormald ,"A note on synchronizer or interlock maloperation," IEEE Trans. Comput. (Corresp.), Vol. C-26, pp. 317-318, Mar. 1977.

[11] M.Hurtado and D.L.Elliott,"Ambigious behavior of logic bistable systems," in Proc. 13th Annual Allerton Conf. on Circuits and

Systems theory, University of Illinois, Urbana-Champaign, Oct 1-3, 1975, pp. 605-611.

[12] T.J.Chaney ,"Comments on A Note on Synchronizer or Interlock Maloperations," IEEE Trans. Comput., Vol. C-28, pp. 802-804, Oct. 1979.

[13] S.H.Unger,"Asynchronous sequential switching circuits with unrestricted input changes," IEEE Trans. Comput., Vol. C-20, pp. 1437-1444, Dec. 1971.

[14] W.W.Plummer ,"Asynchronous arbiters," IEEE Trans. Comput., Vol. C-21, pp. 37-42, Jan. 1972.

[15] R.C.Pearce, J.A.Field and W.D.Little ,"Asynchronous arbiter module," IEEE Trans. Comput., Vol. C-24, pp. 931-932, Sept. 1975.

[16] P.Corsini ,"Self-synchronizing asynchronous arbiter," Digital Processes, Vol. 1, pp. 67-73, Spring 1975.

[17] J.Millman and H.Taub,"Pulse and Digital Circuits," New York: McGraw-Hill, 1956.

[18] M.J.Stucki and J.R.Cox ,"Synchronization trategies," Presented at Conf. on VLSI Architecture, Des. and Fabrication, California Inst. of Technol., Jan. 1979.

[19] F.Rosenberger and T.J.Chaney ,"Flip-flop resolving time test circuit," IEEE J.Solid-State Circuits, Vol. SC-17, pp. 731-738. Aug. 1982.

[20] P.A.Stoll,"How to avoid synchronization problems," VLSI Design, Nov.-Dec. 1982, pp. 56-59.

[21] Chaney,T.J. and Rosenberg,F.U.,"Characterization and scaling of MOS flip-flop performance in synchronization applications", Proceedings of Conference on VLSI Architectures, Design and Fabrication, Pasadena, California, Jan. 22-24, 1979.

[22] Couranz,G.R. and Wann,D.F.,"Theoretical and experimental behavior of synchronizers operating in metastable region," IEEE Trans. on Comput., Vol. C-24, pp. 604-616, June 1975.

VITA

Sina Jafroodi, son of Mr. and Mrs. Hassan Jafroodi, was born on July 22, 1960, in Tehran, Iran. He attended Tehran Conservatory of Music from Sept. 1968 to June 1978 practicing violin. He received his Bachelor of Engineering degree from SUNY at Stony Brook. He continued his studies at Lehigh University as a Teaching and Research Assistant.

His main areas of interest are in device modeling of Ga-As FET and Ga-As digital integrated circuits and applications in MSI and LSI circuits.