2-1974

Macromodular Computer Design, Part 1, Volume 4, The Synchronizer "Glitch" Problem

Computer Systems Laboratory, Washington University

MACROMODULAR COMPUTER DESIGN PART 1. DEVELOPMENT OF MACROMODULES

VOLUME IV

THE SYNCHRONIZER "GLITCH" PROBLEM

Technical Report No. 47

FINAL REPORT - FEBRUARY, 1974
CONTRACT SD-302 (ARPA)

COMPUTER SYSTEMS LABORATORY
WASHINGTON UNIVERSITY
ST. LOUIS, MISSOURI

MACROMODULAR COMPUTER DESIGN

FINAL REPORT - CONTRACT SD-302

FEBRUARY, 1974

Technical Report No. 47

PART 1 - DEVELOPMENT OF MACROMODULES

VOL. IV - THE SYNCHRONIZER "GLITCH" PROBLEM

This work has been supported by the Advanced Research Projects Agency of the Department of Defense under Contract SD-302 and by the Division of Research Facilities and Resources of the National Institutes of Health under Grant RR-00396.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U.S. Government.

Computer Systems Laboratory
Washington University
St. Louis, Missouri

ABSTRACT

There is a fundamental problem in synchronizing communication between any two concurrently operating digital systems that lack a common time reference. This problem involves the inability to build a completely reliable synchronizer or arbiter that will work in a prescribed amount of time. Stimulated by the need for an interlock macromodule design of predictable reliability, the inability to find evidence of previous studies, and indications that this problem has been responsible for significant reduction in the reliability of several commercial machines, we undertook theoretical and experimental studies of this problem. The results to date of these studies are documented in this volume.

TABLE OF CONTENTS

1. INTRODUCTION

It has long been known that there is a problem in communication between two concurrently operating digital computer systems that lack a common time reference, but until quite recently it has been generally believed that there was a safe solution. The difficulty arises when one system attempts to obtain information about the state of another system while that other system may be undergoing a change of state. Depending upon timing, the inquiring system may perceive the observed system as being in the state it had before the transition, in the state that it took after the transition, or in some other state having little or no relationship to either of these.

In clocked systems, the traditional solution to this problem is a device called a synchronizer, which in its basic form consists of a flip-flop whose input is formed by combining a binary level from the observed system with a clock pulse from the inquiring system. In theory, it is assumed that the state of this flip-flop at the time of the clock pulse next following the reading of the input level represents either a logical "0" or a logical "1", and can safely be used as an input to subsequent operations controlled by later clock pulses.

Unfortunately, if the coincidence in time between the level change and the clock pulse is such as to produce a reduced pulse input into the flip-flop, it is possible that the flip-flop will not reach a stable state by the time of the next clock pulse, and that its outputs will not correspond to the defined logical "0" and "1" for the inquiring system. It appears to be true that no finite value for the time interval between clock pulses will guarantee that the synchronizing flip-flop will be in a stable state at the time of the clock pulse following the pulse used to sample the input level.

The effects of such a failure of synchronization are far worse than merely an error in determining the state of the observed system. Since the output of the flip-flop is not in a state that is defined logically, one cannot predict with certainty what will happen to the control of the inquiring system that is using the flip-flop output. The problem is generally dealt with in the design of clocked systems by allowing a sufficiently long time interval, T, between clock pulses so that the probability that the flip-flop has not stabilized by the time its state is sensed is acceptably small. Remarkably enough, there is little information in the literature that would enable one to predict what T should be for a specified circuit and failure probability.

Our recognition of the depth of the problem began early in the development of macromodules (late 1965), when it was recognized that there does not seem to be any way to design general asynchronous control logic that provides for concurrent multiple asynchronous interactions with a single processor which does not, for some timing relationships of the input signals, provide a flip-flop with a marginal input signal. Therefore some tests were conducted to see what effects these "runt" pulses might have on the flip-flops that we were using. The results of these tests, which were similar to the results shown in the photographs of Fig. 2, clearly showed that a flip-flop did not always reach a stable state in the time allowed.

During 1966, the source of this problem and possible ways to avoid it were discussed by us and debated extensively. Late in 1966, we made our first informal attempt to record our understandings and feelings about this problem, which we called "the Glitch". (See Appendix A, which is a reprint of [1].) This report included a philosophical argument that synchronizers must have imperfect reliability due to the glitch phenomenon. In our attempts to avoid the problem by increasing the circuit complexity (playing the game of "musical Glitch"), we were often able to obscure the problem but never able to solve it. This report also presented a development of a mechanical system analogy (a system for which a potential energy curve is applicable) to show that circuit "noise" does not affect the average response of a bistable to input energies in the neighborhood of the "marginal energy" input that would, in the absence of noise, leave the bistable in the metastable state for ever.

Also presented in this report was an arbiter circuit based on detecting the metastable region of the flip-flop that may receive a marginal energy input. The use of this circuit, which appears to allow both a short average throughput time and high reliability, requires that the interacting processes stop while the arbiter circuit is making its decision.

From 1966 to 1971 several designs for interlock macromodules were proposed [2,3,4,5]. Each of these schemes incorporated a metastable detection circuit of the type described. Also during this period a detailed study was done of the behavior of a tunnel diode latch [6].

With few exceptions, our attempts to interest others in the problem during this period were met either with disbelief or the attitude, "it's an interesting problem, but it doesn't have any noticeable effect on today's synchronizer designs". At least two papers concerning this problem were submitted for publication during this period, one by Couranz and Wann of our laboratory, and one by another party, that were rejected on the grounds of lack of general interest to the readership.* A third paper, by I. Catt, was published as a short note in the IEEE computer transactions [7] even though the editor of the transactions "...was quite sure that the problem I (Catt) was discussing did not exist. However, he (the editor) agreed to publish it as a footnote because it might generate some interest and discussion. In fact, the response was nil." (quote from Catt at the 1972 workshop on synchronizer failures [8].) One other early but obscure discussion of the synchronizer problem is found in Gray [9]. Although this book was published in 1963, the section dealing with the synchronizers was not brought to our attention until 1971. Part of the reason is that Gray's discussion is in chapter 6, "Digital Computer Circuit Analysis", section 6.21; which is a 4-page section between 6.20, "Graded-Base Transistors", and 6.22, "Pulse Transformers".

Showing of photographs of a misbehaving flip-flop at the December, 1971, ARPA IPT contractors' meeting, and at a workshop on modular computer systems in St. Louis, generated interest in a workshop on synchronizer failures. This workshop was quickly organized by us and held in April of 1972. The consensus arrived at during this workshop can perhaps be summarized:

* It is interesting to note both these papers were resubmitted in 1973 and both were accepted for publication. [30,31]

1) The problem is fundamental in any communication between two systems not sharing a common time reference, and it is not possible to build a completely reliable synchronizer or arbiter that will work in a prescribed amount of time. (Nevertheless, theorists postulate such devices and use them as components of systems which people then build.)

2) There is no adequate theoretical treatment predicting the failure probability of a synchronizer due to this class of mechanisms, and new failure modes are still being discovered.

3) Specifications of existing integrated circuits are not adequate to permit the evaluation of a particular circuit for synchronizer or arbiter service, and representatives of the semiconductor industry expressed pessimism with respect to any useful response from manufacturers of semiconductors.

4) There is considerable evidence suggesting that present systems are crashing and failing at significant rates due to this problem.

These conclusions, and our own need for an interlock macromodule design of predictable reliability, stimulated efforts on our part toward further theoretical and experimental study of the problem. The evidence from the workshop that significant synchronizer reliability problems had occurred in the IMP used in the ARPANET, and in several commercial machines made by Honeywell (DDP-516) and Digital Equipment Corporation (PDP-10 and PDP-11/45) also had encouraged us to redouble our efforts to make the problem more widely known.

2. OBSERVATIONS OF FLIP-FLOP RESPONSES TO MARGINAL INPUTS

Although it has been known for some time by others [7,9,13,14] as well as by us that flip-flops, in response to marginal input conditions, can have outputs that are logically undefined and that the probability the flip-flop has settled from this undefined region in any given time interval is less than one, very little experimental verification has been done prior to the work of T. Chaney, who has observed and photographed many different types of flip-flop misbehavior [6,10,11]. Some of the methods used to obtain these photographs are discussed in Appendix B.

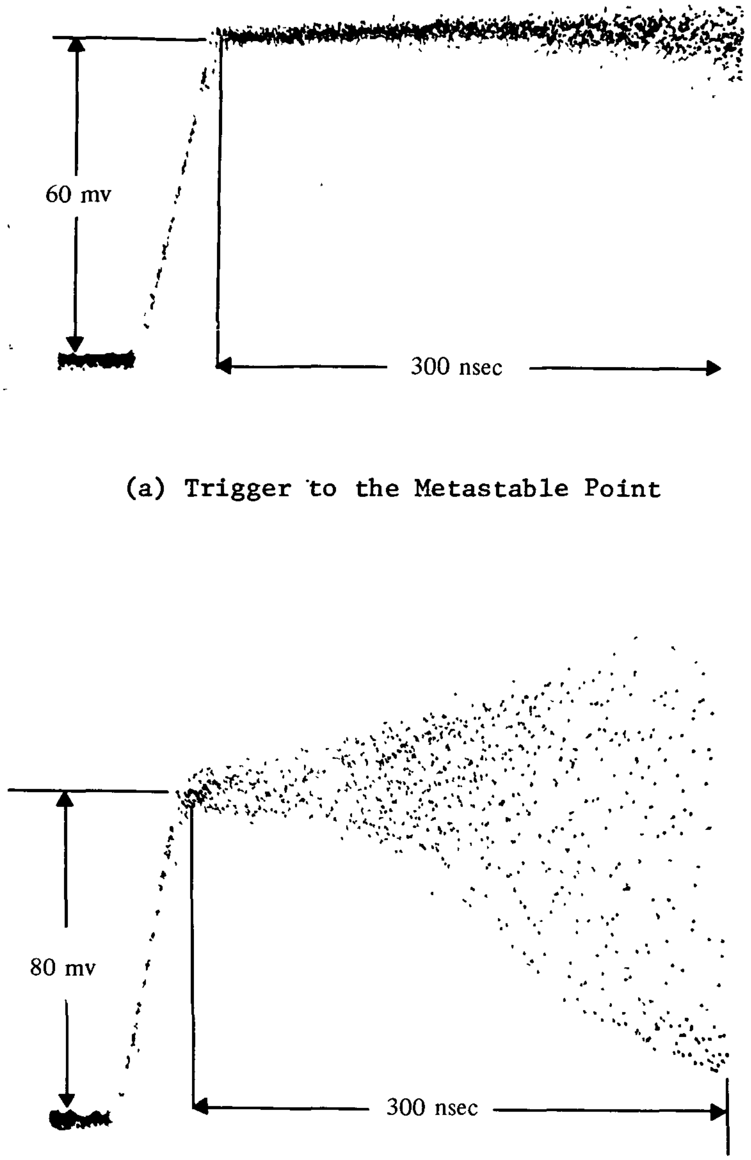
In one mode of flip-flop behavior, the output hovers for an indeterminate time at a metastable value somewhere intermediate between the defined "0" and "1" output levels. This mode is typical of flip-flop circuits that have a small signal propagation time to signal rise time ratio. The sampling oscilloscope photographs of Figure 1 show, by dot density, the relative probability that the tunnel diode flip-flop has not settled when the input triggering amplitude is adjusted to a marginal size. This flip-flop has been slowed down by the addition of a 200-pf capacitor in parallel with the tunnel diode so that better measurements can be taken to verify a model [6]. Figure 1a indicates the circuit response with a fixed trigger amplitude, and Figure 1b indicates the response when the trigger amplitude is modulated.

Figure 2 shows the response of an emitter-coupled logic (ECL) clocked R-S flip-flop (MOTOROLA MC1016) when the clock input signal is switched off as the data input signal is changing. The sampling oscilloscope photo shows the relative probability that the flip-flop has not settled. The photographs from a real-time oscilloscope show the details of some individual trajectories.

In another common mode of anomalous behavior, the Q and $\overline{Q}$ outputs oscillate in phase a number of times between the "0" and "1" states before finally coming to rest out of phase. This mode is typical of flip-flops constructed from gates with large propagation time to rise time ratios. Transistor-transistor logic (TTL) R-S type flip-flops are often constructed by cross-tying two NAND gates. Figure 3 shows the resulting behavior when a runt pulse is supplied to one input of the flip-flop. The mode of behavior of R-S flip-flops constructed from low-power TTL (the SN74L00 gates) is intermediate between the hovering and oscillating modes.

The flip-flop reaction shown in Figure 2 may be explained in simple terms by considering two inverting gates connected in series as shown in Figure 4. With the switch open, bias the input of the first gate such that the output of the second gate is equal to the input bias voltage ($V_Y=V_{IN}$). Then connect the output of the second gate to the input of the first gate by closing the switch (thus forming a flip-flop) and remove the bias input.

In the absence of noise sources, the system is at an unstable equilibrium point and will stay there forever. Since any infinitesimally small energy source will cause the flip-flop to leave this point, the presence of circuit noise will cause the flip-flop to switch after some period of time.

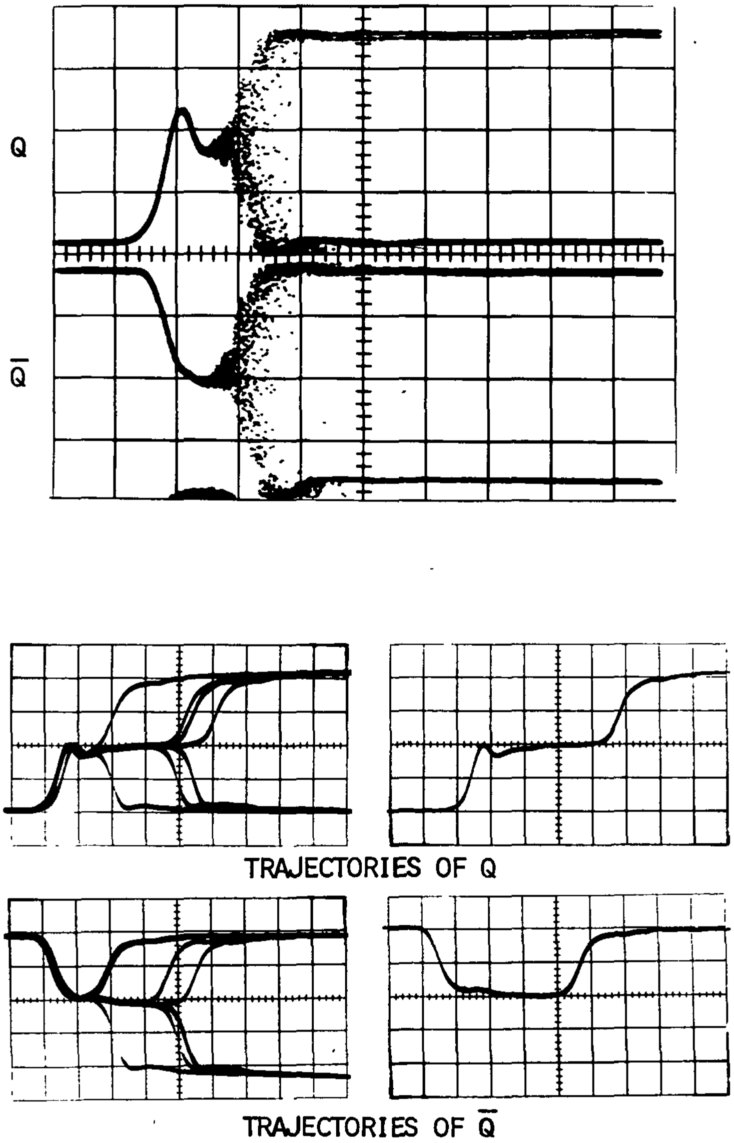


(a) Trigger to the Metastable Point

(b) Trigger with Triangular Modulation

Figure 1. Sampling oscilloscope photographs of a tunnel diode type flip-flop triggered into the metastable region with and without amplitude modulation.

Photographs Reprinted From Technical
Report 15 [6]

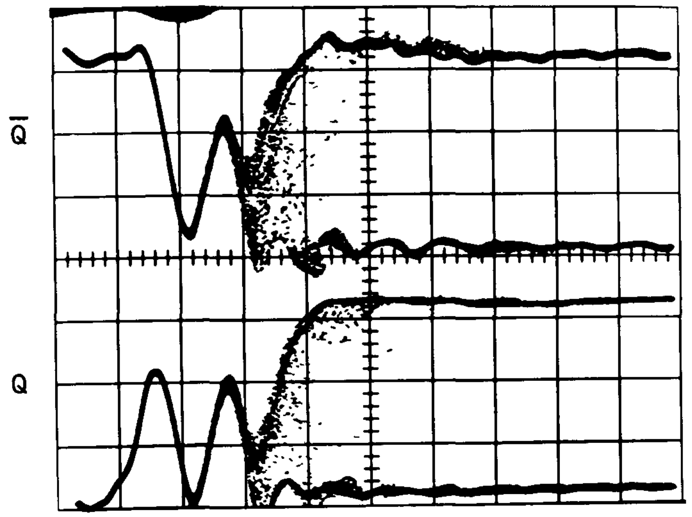


TRAJECTORIES OF Q

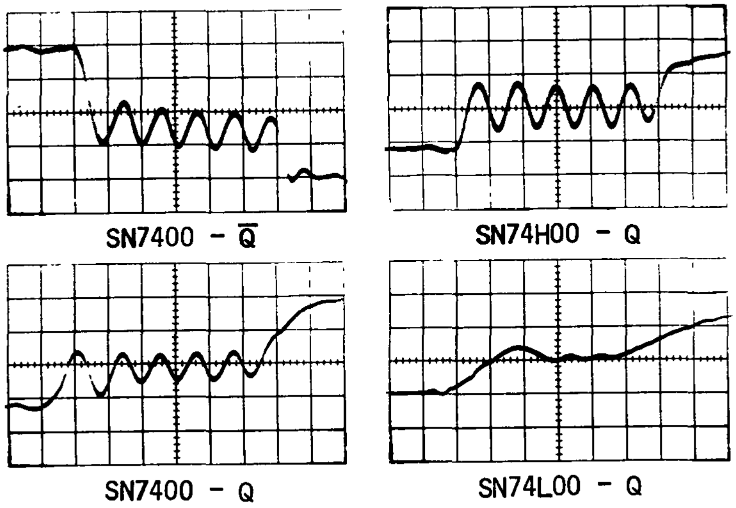
TRAJECTORIES OF $\overline{Q}$

Figure 2. Sampling and real-time oscilloscope displays of the response of an ECL clocked R-S flip-flop (Motorola MC1016) to the clock input signal being switched off as the data input signal is changing. 0.2 V/div. for real-time photographs; 10 nsec/div. for all photographs.

Reprinted from COMPCON - 72 Digest [10]



Flip-Flop Constructed from SN7400 Gates


SN7400 - $\overline{Q}$

SN74H00 - Q

SN7400 - Q

SN74L00 - Q

Selected Trajectory from a Flip-Flop Constructed from
Type Gate Indicated

Figure 3. Response of TTL R-S flip-flops, constructed by cross-tying two NAND gates, to a runt pulse at one input. 1 V/div., 10 nsec/div.

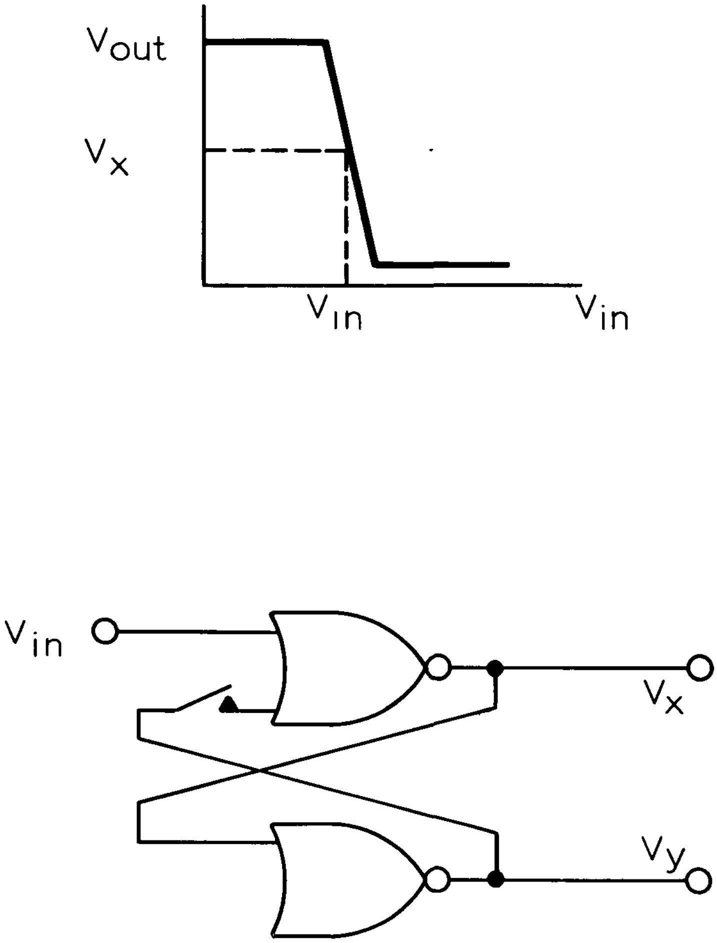


Figure 4. Simple flip-flop.

3. THEORETICAL STUDIES

The most commonly used model for metastable behavior of flip-flops is essentially that presented by Gray [9], which assumes that the system is linear and unstable with a single pole on the real axis in the right half-plane. In the noise-free version of this model, the system output is approximately of the form:

$$V(t) = [V(0) - V_o] \exp(\frac{t}{T})$$

where V(0) is the initial output voltage, V_o is the metastable voltage, and T is a time constant characteristic of the circuit.

This model predicts the commonly observed result that the tail of the decision time frequency distribution is exponential in form, and Couranz [6] has shown that for the conditions he chose this noise-free model predicts results essentially identical to those obtained assuming the presence of reasonable amounts of Gaussian noise. On the other hand, Hurtado [16] has demonstrated by analysis that there are conditions under which noise can have a large effect.

While useful, this simple model is not adequate to explain a number of observations, particularly those showing oscillatory behavior. It is also too specific to be useful in discussing the fundamental limits of synchronizer performance. Hurtado, in our laboratory, has undertaken development of a more comprehensive theory based upon the general theory of bistable and multistable dynamical systems representable by a set of simultaneous non-linear differential equations, relating the values of state variables and their derivatives. In his preliminary results, he has found that for such noise-free systems the "glitch" problem is inescapable. The theoretical treatment of the effects of noise upon this conclusion is not complete, but there is not any reason at present to expect that such systems with noise can prove perfectly reliable synchronization either.

What is of particular interest in these studies is the fact that the input interval as well as the free behavior interval of the system can be represented by this class of model, which also appears general enough to account for all types of flip-flop behavior yet observed. Another aspect of this model is the likelihood of being able to justify the use of noise-free models and to ascertain the conditions under which they are good approximations.

Other theoretical studies in our laboratory by Srinivasan and Chaney [12], stimulated by difficult-to-explain observations of the behavior of flip-flops constructed by Seitz [15] from TTL gates with totem-pole outputs, have discovered a new set of unpleasant phenomena relating to positive feedback within the TTL output circuit. Under some conditions, particularly when inputs to the gate are in the logically undefined region, the circuit can show negative resistance at its input, output, and power supply terminals. When such gates are cross-tied to construct flip-flops and these flip-flops are marginally triggered, very complex and potentially hazardous behaviors can be observed. Painstaking attempts are now underway to classify these behaviors and account for each type theoretically, using Ebers-Moll circuit models for the transistors in the TTL devices.

Aside from its relevance to the synchronizer, this work suggests the need for great caution in using TTL logic with slowly changing inputs.

4. ANALYSIS OF A SYNCHRONIZER DESIGN

Although examples of poor synchronizer designs are abundant, we chose to analyze carefully a commercially available TTL-type integrated circuit that has been designed specifically for use as a synchronizer in order to illustrate how the marginal energy condition affects the reliability of a synchronizer. This device, whose logic diagram is shown in Figure 5, is advertised as a "Dual Pulse Synchronizers/Drivers" in which "Latched Operation Ensures that Output Pulses are not Clipped" [17].

The function of this circuit is to gate the next clock (C) pulse(s) through the synchronizer after an interrupt occurs at the output of gate B. The A-B flip-flop provides wave shaping for the interrupt input and as such is not part of the synchronizer circuit. To simplify the discussion, we will disable the A-B flip-flop by setting the R input low and the S2 input high. S1 then controls the output of gate B and will be considered the interrupt input. Note that reasonable circuit performance requires that the interrupt input signal be longer than one clock period.

The circuit is activated by switching S1 low. If the mode (M) input is grounded, beginning with next clock pulse, all following clock pulses will be gated through the synchronizer (through gate J) until S1 is switched back high. If the M input is held high, only the next clock pulse will be gated through. S1 must then be switched back high (to reset the D-E flip-flop), then low again to produce another output pulse.

There are two cases of marginal energy input in this circuit that cause anomalous behavior: the case when an interrupt arrives (S1 switching low) about the time the clock pulse is switching high and, for the chain of pulses mode (M grounded), the case when the interrupt is switching off (S1 switching high) as another clock pulse is arriving.

First let us consider the marginal energy case that occurs when the asynchronous input is stopping the clock pulse chain about the time the next clock pulse is switching high. The problem occurs when the input timing is such that input No. 2 to gate J and input No. 1 to gate I are both switched high at the same time. The flip-flop, consisting of gates I and J, is then left to decide which output will be a "1" and which will be a "0". Some typical responses are shown in Figure 6. These output waveforms were produced by carefully adjusting the time between the input signals. The circuit produced responses like these pulses over an input signal timing variation of approximately 10 psec (10^{-11} sec). Note that the Y output pulse in Figure 6 is a logical high for more than 20 nsec. The waveforms shown represent a family of pulses of varying width. Y output pulses of this form as wide as 40 nsec. have been detected.

The apparent timing conflict between the description of the circuit performance and the input pulse waveforms shown in Figure 6 can be rationalized by noting that the 3-gate path of the S1 input pulse (through gates B, H, and I) to the output of gate I is through internal gates which typically have propagation times that are much shorter than the typical propagation delay of an output gate such as gate J. Therefore, the output response of gate J and gate I can occur at the same time with the input pulse timing shown in Figure 6.

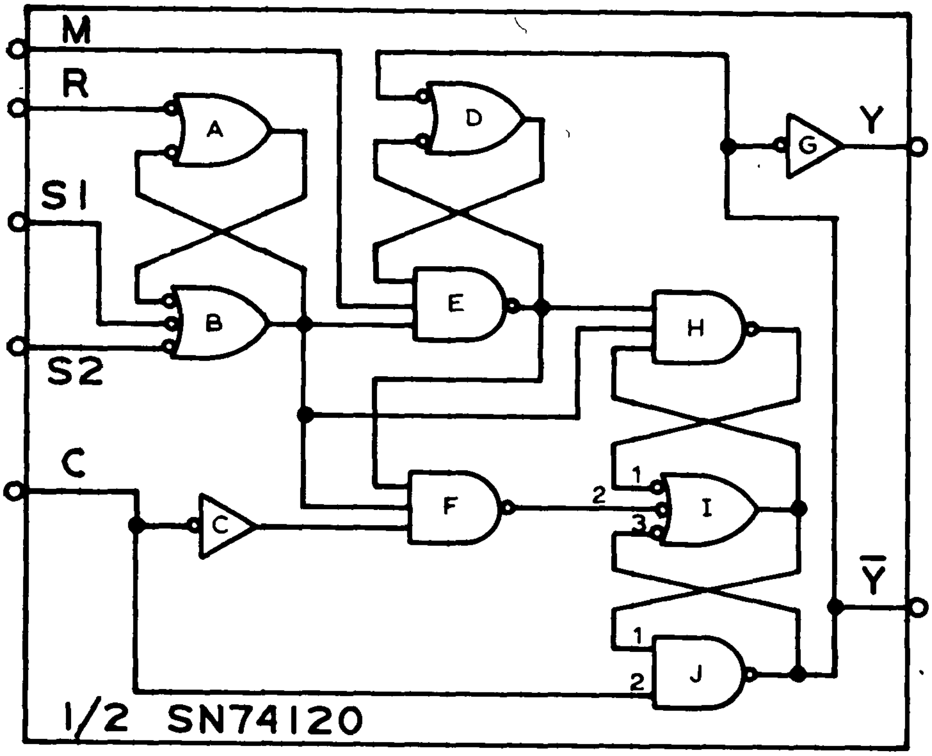
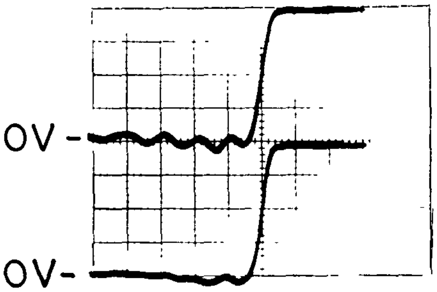


Figure 5. Logic Diagram of the SN74120 "Dual Purpose Synchronizers/Drivers".

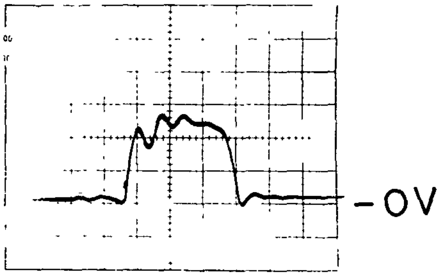


LEADING EDGE OF INPUT PULSES (Pulse Width is 2 μsec)

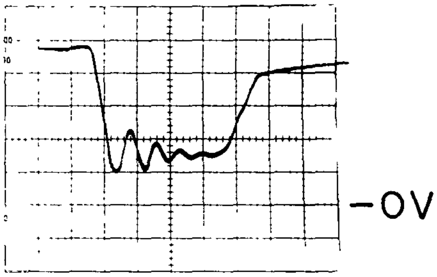
signal applied to "S1" input

signal applied to "C" input

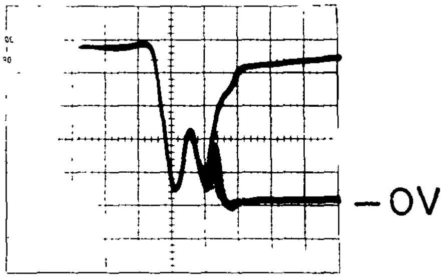
ALL OSCILLOSCOPE PHOTOS ARE 10 nsec./div. and 1 volt/div.



SINGLE RUNT (CLIPPED) PULSE FROM "Y" OUTPUT



SINGLE RUNT (CLIPPED) PULSE FROM "$\overline{Y}$" OUTPUT



FAMILY OF "$\overline{Y}$" RESPONSES TO INPUTS SHOWN

Figure 6. Some typical responses of a SN74120 to the input signals shown. (Pins M and R were held low, and S2 was held high.)

The other marginal energy case, which exists for both modes of operation, occurs when the first clock pulse is gated through. If the asynchronous input, S1 , is switched low about the time the clock pulse is switching high, it is possible to produce a negative-going runt pulse at input No. 2 of gate I. At the time this runt pulse arrives at I, the output of I is negative and all other inputs to the H-I-J latch arrangement are high. If the runt pulse has sufficient energy to switch either the H-I or the I-J flip-flop, the other flip-flop will also switch. If the two flip-flops have different input energy thresholds, the problem occurs when the flip-flop with the lowest threshold is driven into the middle region. The runt pulse could drive the H-I flip-flop into the middle region, and if the H-I flip-flop later switched, the clock pulse would be gated through J late. Thus the clock pulse delay through the synchronizer would be large. On the other hand, the runt pulse might drive the I-J flip-flop into the middle region, resulting in outputs much like those shown in Figure 6. Other modes involving all 3 gates at once are also possible.

To illustrate one of the possible effects of these marginal energy conditions, consider using this synchronizer driver circuit in the single output pulse mode. The first pulse at $\bar{Y}$ will set the D-E flip-flop, which will then inhibit any further output pulses until the D-E flip-flop is again reset by the interrupt signal. If the input time relationships and internal gate thresholds are such that the first $\bar{Y}$ pulse is a runt pulse with enough energy to set the D-E flip-flop, but not enough energy to propagate through the external logic gates, then the logic circuit which detects the synchronizer/driver output and then later notifies the external device that the request has been serviced will never produce a pulse. Thus the interrupt input will never be reset, and the external device will stop working for no apparent reason.

This synchronizer/driver circuit is but one example of a circuit that, except for this one fatal flaw, appears to be a good design. Because the effects of marginal energy input conditions were apparently not understood by the designer, an important product has turned out to have a serious hidden flaw.

5. EXPERIMENTAL DETERMINATION OF SETTLING TIME PROBABILITY

In order to provide data on the settling time probability of flip-flops, two SN74120's with different date codes, and some ECL type flip-flops (Motorola MC1016) were tested. The probability the output is not yet resolved, shown in Figure 7, results from the test data taken, which covers the full 8 decade range shown. The SN74120's were tested under the conditions described in Figure 6 so that the results would reflect the performance of TTL R-S flip-flops. The tests were conducted with the input signal timing variation being controlled, using an adjustable air-dielectric coaxial line, within the 10 psec "window" required to produce long resolution times. The "window" size is a weak function of the input signal rise/fall times. A change of input rise/fall times from 3 nsec to 20 nsec changed the window width from 10 psec to 20 psec for the ECL circuits.

A pulse was generated with a width equal to the length of time the flip-flop was in the middle region. The length of each output pulse was quantized into one of several time intervals, and the outputs of the quantization circuit were connected to a set of counters. The test results were numbers in each of a set of counters. Figure 7 is a plot of these numbers normalized to the beginning of the long resolution times. A more detailed description may be found in Appendix C.

These test data are useful if a flip-flop is allowed a minimum of 16 to 21 nsec (depending on circuit type) to "settle" before the output is used by other logic. An equation of the form $P = \exp\left(-\frac{t-D}{T}\right)$ may be used to fit the curves in Figure 7. P is the probability that the output has not yet resolved; t is the time, measured from the input clock transition, allowed for resolving; D may be thought of as an extended propagation time, like that of a comparator to a small input overdrive; and T is calculated from the slope of the curve. The following constants were derived for the two types of flip-flops tested, along with an indication of the accuracy obtainable from the test data.

TABLE 1

	ECL (MECL II type)	TTL (7400 type)
D	16 nsec ± 1 nsec	21 nsec ± 2 nsec
T	2.1 nsec ± 0.1 nsec	1.8 nsec ± 0.2 nsec

Designers should be aware that the circuit model used to predict the simple equation used above does not include the negative input and output impedance exhibited over part of the active region by TTL gates with active pull-up output stages [12]. Although the 7400 type flip-flops tested behaved as predicted by the simple model, some flip-flops, made from Schottky-clamped gates (SN74S00), have been observed to remain in the middle region for long periods of time (seconds) when tested under marginal input conditions [15]. Also, a more complex Schottky-clamped latch, an Intel 3404 6-bit latch, was tested. The probability that the flip-flop has not yet settled from the middle

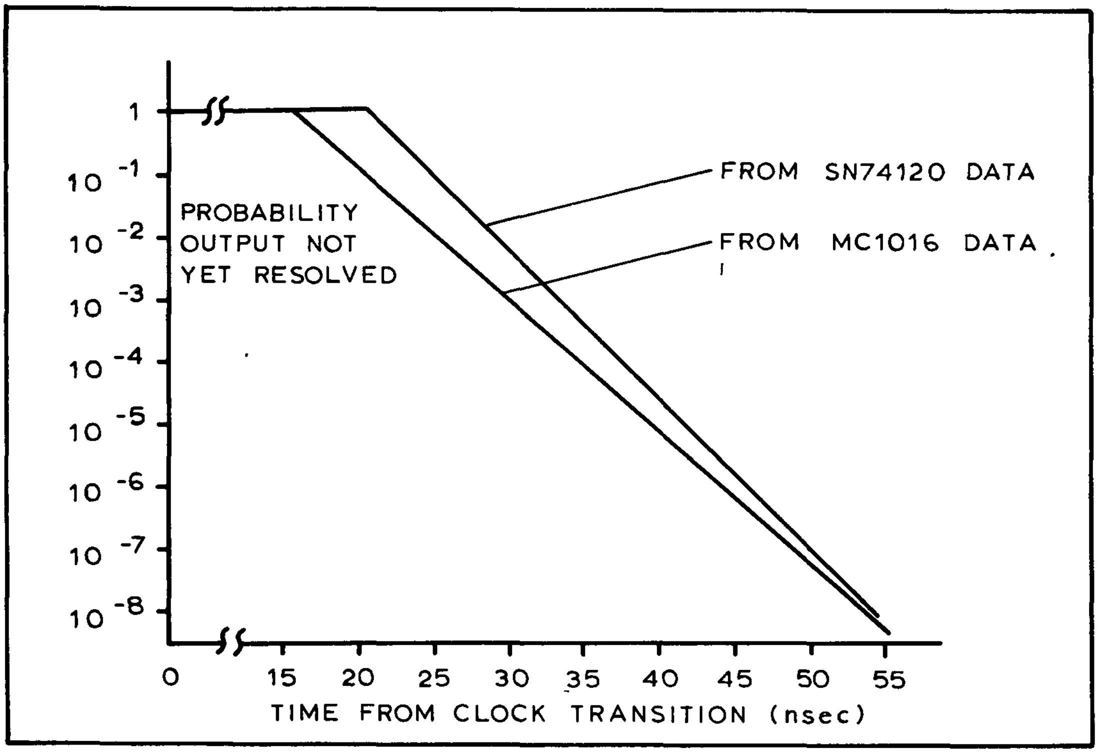


Figure 7. Probability flip-flop has not settled from middle region.

region curve for this latch was to the right of the SN74120 curve in Figure 7 with a slope of a little less than the slope for the MC1016. The internal circuit used to form the latches in the 3404 is different than the circuit used in the SN74120. Therefore, at the present time, the results of the 3404 tests is only another indication that Schottky-clamped latches may make particularly bad synchronizers.

Until the negative impedance effects are understood, a conservative synchronizer design should use either ECL, or TTL gates with passive pull-up output stages. If active pull-up TTL gates must be used, Schottky-clamped circuits should be avoided.

6. RELIABILITY OF SYNCHRONIZERS

Returning to the SN74120, consider its use in a system with a clock rate of 1 MHz and an external asynchronous input signal average rate of 100 Hz. If we assume the interrupt signal meets the conditions defined in Appendix E and the input "window" is 10 psec, a clipped pulse will be produced at the output of the synchronizer:

$$\begin{array}{l}\text{Mean Period}\\ \text{Between}\\ \text{Clipped Pulses}\end{array} = \frac{1}{(100\ \text{Hz})\ (10^{6}\text{Hz})\ (10\text{x}10^{-12}\text{sec})} = 1000\ \text{sec} \doteq 17\ \text{min.}$$

Most of these clipped pulses will probably not affect the operation of the system, however; for comparison consider a synchronizer of the type shown in Figure 8, which uses two edge-triggered latches. The asynchronous input is strobed into the first latch on the one clock edge, then strobed into the second latch on the next clock edge. The first latch thus has a full clock period to settle. This synchronizer will not fail if the first latch settles in less than a clock period, or mathematically (see Appendix E for equation development):

$$\begin{array}{l}\text{Mean Period (Sec) for Output not Yet}\\ \text{Resolved at End of Clock Period}\end{array} = \frac{\exp\left(\dfrac{\frac{1}{\text{C.R.}} - \text{D}}{\text{T}}\right)}{(\Delta t)\ (\text{C.R.})\ (\text{Average I.R.})}$$

"C.R." is the clock rate in PPS; "I.R." is the interrupt rate in PPS; "Δt" is the input "window" in sec (about 10 psec for the ECL and TTL circuits tested); and "D" and "T" are derived from experimental data (see Table 1).

If we further assume that the clock rate and the average interrupt rate are related by some factor, A=I.R./C.R., the curves in Figure 9 result. These curves show that a small increase in clock rate can produce a dramatic loss of reliability. Also note that the data rates used as an example for the SN74120 give a mean time between failures of 10^{187} centuries for this two flip-flop type synchronizer.

The synchronizer circuit of Figure 8 is a special case of a synchronizer which strobes as asynchronous input into a flip-flop, then waits as long as possible before using the output. Figure 7 can be used to calculate a mean failure rate for any waiting period. If at least a 110-nsec waiting period is allowed, the mean predicted time between failures for typical circuits is years. At the other end of the spectrum, if less than a 30-nsec waiting period is allowed, the predicted mean time between failures will approach seconds.*

*For applications which need a synchronizer that operates reliably with a shorter waiting time, a faster-resolving bistable element is needed. Some tunnel diodes resolve very fast, with values of T (Table 1) of 60 psec or less, and D less than 1 nsec. Thus a settling time of less than 5 nsec for a fast tunnel diode is equivalent to a settling time over 110 nsec for the ECL and TTL flip-flops. An ECL type integrated circuit synchronizer which utilizes a tunnel diode bistable is being developed at the M.I.T. Lincoln Laboratory [18].

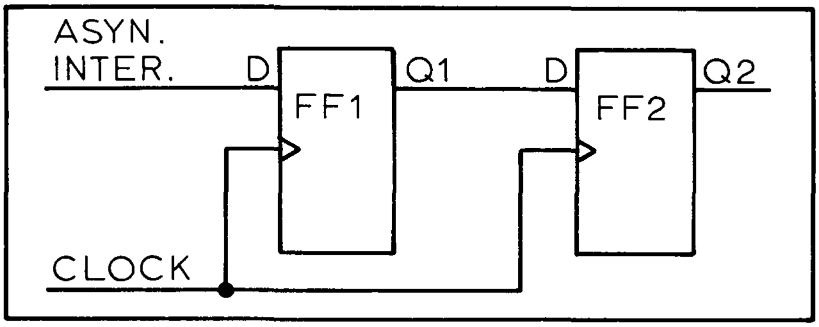


Figure 8. Two flip-flop type synchronizer.

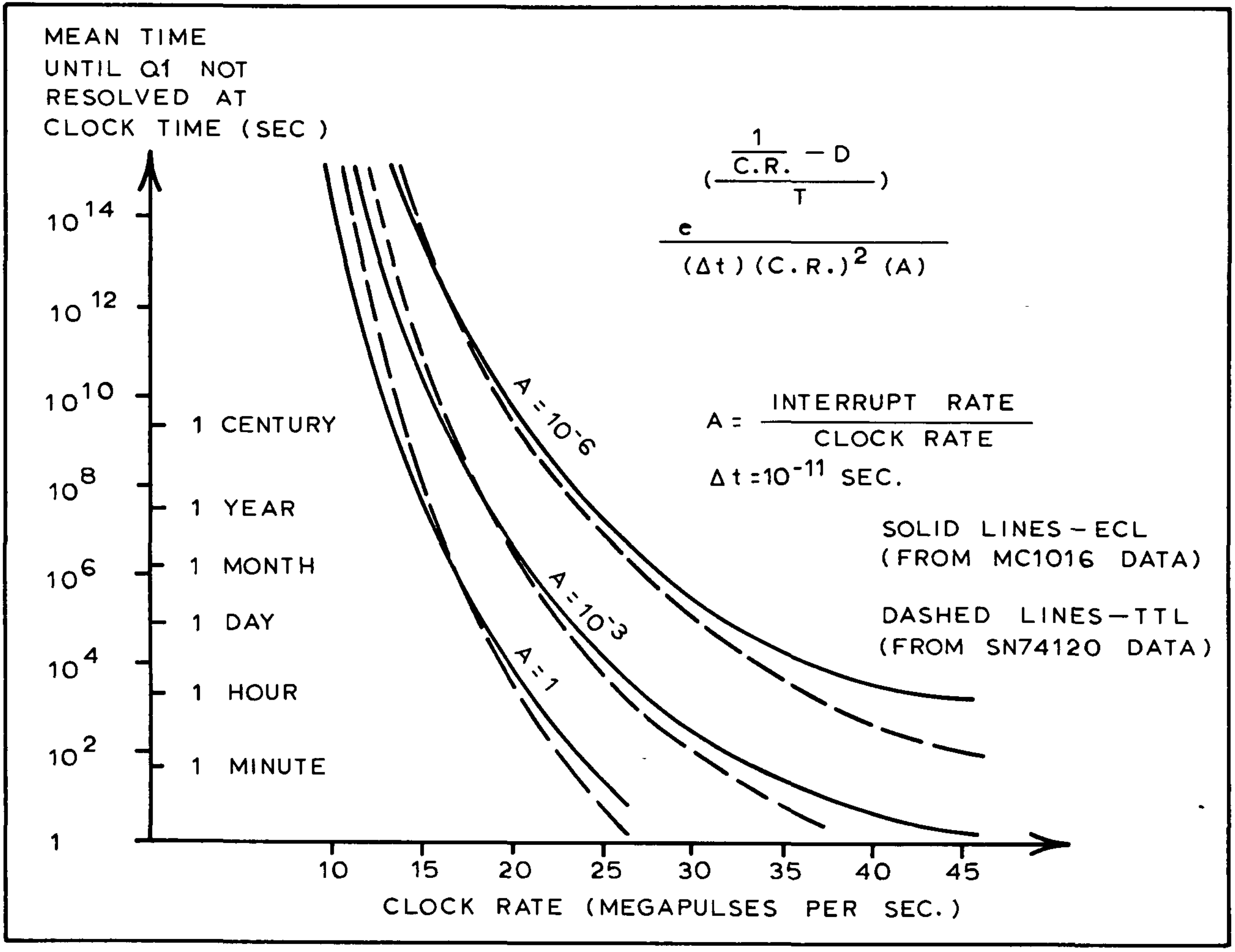


Figure 9. Probability that FF1 may cause FF2 to have an undefined output (for synchronizer of figure 8) with values of "D" and "T" from Table 1. (The interrupt signal is assumed to be statistically independent of the clock signal and all previous interrupt signals, and have a uniform distribution in an average interval of $(I.R.)^{-1}$.)

7. ARBITER SOLUTIONS FOR ASYNCHRONOUS SYSTEMS

A variation of the clocked synchronizer circuits previously described has been proposed for use in asynchronous systems to provide arbitration between signals [19]. However, it appears much more natural to adopt an entirely different approach, which allows a short average throughput time and high reliability by incorporating a circuit which detects the settled flip-flop state [1]. This approach, however, does require that the two interacting processors stop when the arbitrating circuit is making a decision.

The ECL waveforms shown in Figure 2 suggest that a comparator can be used to detect when a flip-flop output is in the middle region. Such a detecting circuit is shown in Figure 10. This circuit requires that when Q and $\bar{Q}$ differ by less than V volts, both comparators will enable the AND. The two batteries of V volts can be replaced by resistor divider or diode-resistor networks.

Although the outputs of TTL flip-flops usually oscillate in phase a number of times before settling, the common mode rejection characteristic of the comparator allows this type circuit to be used in TTL designs as well as ECL designs. The M output of this circuit can be used to inhibit action until the flip-flop is settling. Then, after a short delay to allow the flip-flop time to finish settling, the output of the flip-flop can be polled and the appropriate action taken.

Various members of the Computer Systems Laboratory have designed ECL versions of arbiters which used a circuit of the type shown in Figure 10 [1,2,3,4,5], and a Schottky TTL circuit using a different type offset detection has been proposed by Seitz [21]. However, as an example of a possible way the circuit of Figure 10 can be used, let us consider a simplified version of the circuit used in the INTERLOCK module [20]. A block diagram of this simplified interlock is shown in Figure 11, and a logic diagram of the central part of the interlock design is given in Figure 12. This simplified interlock uses level signals as opposed to the transition signals used by macromodules.

This interlock design is an answer-back type scheme. For example, a level change at I_1 will produce a level change at D_1. The circuitry receiving the D_1 signal is expected to return a U_1 signal when it is ready for another D_1 input signal. The interlock then returns the D_1 level to the inactive state and produces a C_1 level change. The circuitry that produced the I_1 input signal must then return the I_1 signal to the inactive state, and finally the interlock returns the C_1 level to its inactive state.

The logic diagram shown in Figure 12 includes that part of the interlock circuit that controls the priority between inputs (I_1 has the higher priority), and the part that resolves the arrival times of the inputs. The numbers shown in the logic element symbols are Motorola MECL II part numbers.

When an input signal arrives at I_1 or I_2, there is a period of time (15-20 nsec) before the W signal at the two bottom 1010 gates inhibits any further inputs, or closes the "window". If a signal is arriving at the other

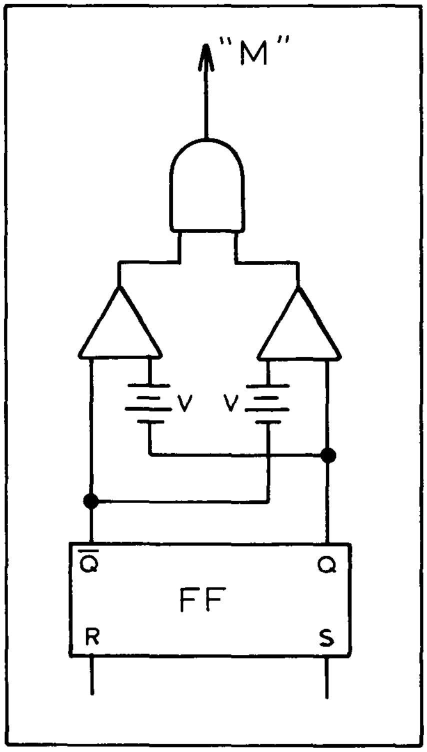


Figure 10. Heart of an arbitrating circuit for asynchronous systems.

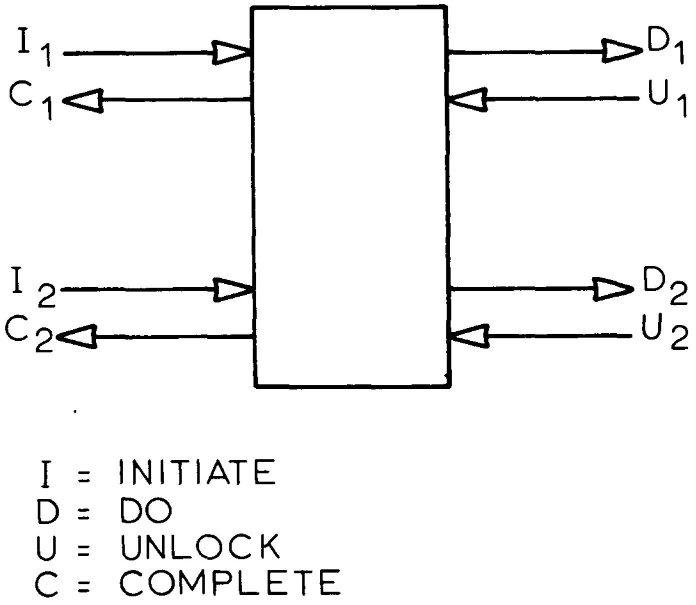


Figure 11. Simplified macromodular interlock.

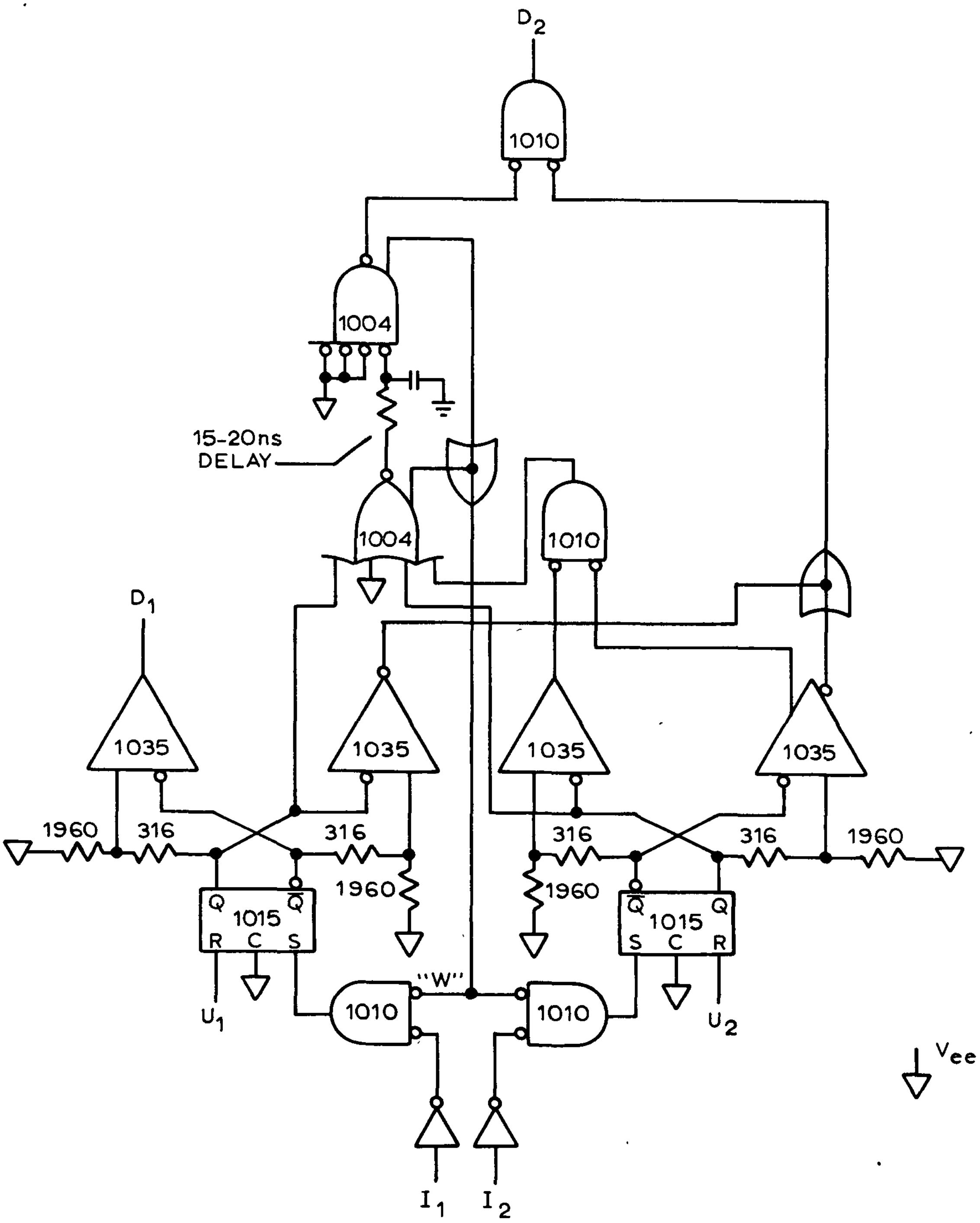


Figure 12. Central part of interlock design (for level signals).

I input just as the W signal is arriving, the 1015 may receive a marginal energy input pulse. If the other I input arrives before the W signal, then both 1015 flip-flops will be set and the outputs, D_1 and D_2, will be activated in turn. If the second 1015 receives a marginal energy pulse and enters the middle region, the interlock circuit will still function in a logically defined manner.

The 1960-ohm and 316-ohm resistors replace the V-volt batteries shown in Figure 10. The right-hand section (the U_2 side) of the logic diagram includes a copy of Figure 10 consisting of the 1015 flip-flops, the two resistor divider networks, the two 1035 comparators, and the 1010 (whose output is equivalent to the M output of Figure 10). The left-hand side (the U_1 side) is somewhat abbreviated because if the left-hand 1015 is in the middle region and later settles to a one, D_1 will have first priority, and therefore the output of the 1035 comparator can be used as the D_1 output. Also, it is sufficient that the lower-priority U_2 side only know that the U_1 1015 flip-flop output is not a zero.

Asynchronous bus-oriented computers can use an arbiter design of this type. The slower devices connected to the bus, which cannot be reasonably stopped, such as tape drives and disks, could use a delay-type synchronizer (such as the type shown in Figure 8) without appreciably degrading the system performance.

8. CONCLUSION

Since the April 1972 workshop, we have served as a consulting resource and a clearinghouse for information concerning this problem, and have tested circuits and provided advice and information to others, particularly the C.mmp project at Carnegie Mellon University and the ARPANET project at Bolt, Beranek & Newman. We have had many dozens of requests for reports and reprints, and feel that we have made an important contribution to awareness and understanding of this problem.

There is still a gap between our experimental observations and available theoretical explanations, particularly concerning the behavior during the input pulse. The experimentally observed input "window" is not well defined, but seems to be related in unexplained ways to circuit parameters such as internal noise, gain, and frequency response. The relating of experimental evidence and theory to achieve circuit performance predictions based on easily measured parameters, and the understanding of optimum design of flip-flops for synchronizer service, are yet to be realized.

There are still widely used flip-flop circuit types which have not been examined, particularly those circuits which use field-effect transistors. As recently as last summer, senior representatives of companies marketing LSI computer chips based on the MOS technology were not aware of the Glitch problem, although their computers have internal synchronizers.

Our appreciation of the richness and complexity of the set of problems associated with synchronizers and arbiters continues to grow with each discovery of a new mode of misbehavior or a new system problem possibly caused by synchronizer failure, and it is somewhat surprising that the problem is taking so long to be widely appreciated. Despite the early publications of Gray [9] and Catt [7], from 1966 until late 1971 we found little or no evidence of understanding or concern about this problem, despite numerous efforts to make it known. Some evidence of understanding the problem has developed since the April 1972 workshop [22,23,24,25].

We feel that it is critically important that this problem be fully understood and that an awareness of its implications be aroused in system and circuit designers, who are otherwise apt to be trapped into having to make an unhappy choice, late in the design of a system, between severely degraded performance or marginal reliability. The trend toward higher clock speeds, more asynchronous operation, and multiprocessor systems is rapidly increasing the vulnerability to this problem, and there is a serious need for better information for designers.

The best promise, at present, for a trouble-free synchronizer appears to lie in the use of a fast tunnel diode as the bistable element, since it can offer highly reliable synchronization in times an order of magnitude faster than needed for similar reliability with bipolar transistor circuits. The design of a convenient integrated circuit interface for employing such circuits as part of an ECL 10,000 system is underway at Lincoln Laboratoiy, but an accelerated effort toward commercial availability of such circuits would be wise.

Although the synchronization problem appears to be fundamental, system designers, through ignorance, frequently choose arrangements that create needless exposure to the problem. Such examples have been found in interface and bus designs, and in the design of multi-processor systems. These errors sometimes cause trouble, and sometimes they appear not to. On occasion they are easy to correct, but in other cases they have been deeply embedded.

In closing, there are several pieces of advice that can be offered to aid the system designer. First, minimize the unessential use of synchronizers when an acceptable alternative is possible. Second, centralize sychronization tasks in one or a few synchronizers; if there is trouble, there are fewer circuits to study or replace. Third, test subsystems using synchronizers under conditions that produce marginal inputs, and observe the behavior for unexpected phenomena. Fourth, choose circuits and devices that have more predictable behavior if at all possible, such as ECL, or ordinary TTL, rather than Schottky TTL devices. We are unaware of any experimental studies of this problem in MOS devices. Fifth, be wary if available decision times are less than 100 nanoseconds, and take extreme precaution if the time available is less than 50 nanoseconds. Sixth, consider in high-performance systems the use of an asynchronous arbiter that detects the making of a decision, rather than one which allows a fixed decision time. Seventh, design the synchronizer to allow the decision-making element as much time to settle as the system constraints will readily allow. Finally, ask yourself whether an erratic system could possibly be due to this problem; it is difficult to make a Glitch on the laboratory bench, and nearly impossible to find one when looking for another cause.

9. APPENDIX A: REPRINT OF TECHNICAL MEMORANDUM NO. 10, "THE GLITCH PHENOMENON" (1966)

9.1 INTRODUCTION

This paper deals with what has become known as the "Knife Edge" problem. In essence, the problem is that of resolving two independent events such that no ambiguity arises. It should be emphasized that at that point of uncertainty it makes no difference how the decision is resolved, just so long as a firm decision is made in a finite amount of time.

The treatment given in this paper, although somewhat lacking in formal mathematical rigor, is not entirely cursory. Rather, emphasis has been placed on an intuitive understanding of the problem, its solution, and finally the implications of the solution. The problem will be considered in two parts. First, the ideal noiseless case,which is admittedly an abstraction, but nonetheless instructive; and second, the "real world" noisy case with its further probabilistic complications.

Let us first begin by familiarizing ourselves with exactly what the problem is. Consider a woman sitting in the living room of a house having two doors, front and back, and likewise two doorbells. She has two pet doorbell rules which are simply these: 1) She doesn't answer a doorbell that hasn't been rung. 2) She answers the first one she hears. This solution works as long as the milkman and the postman don't come at the same time. Having foreseen that this event might occur one day, she made the arbitrary decision that should they both ring at the same time, she will give priority to the front door. Content in the knowledge she has solved her problem, she waits in the living room for the doorbell to ring. One formidable day, the postman and the milkman both decided to deliver at exactly the same time. However, as fate may have it, the milkman's finger reached the back doorbell just a millisecond or two before the postman's finger reached the front doorbell. Now, the poor lady of the house was fraught with indecision. It appears that two milliseconds is on the threshold of her ability to distinguish between two simultaneous events, and two non-simultaneous events. In a word, she could not decide whether the back doorbell rang first, or whether they both rang simultaneously.

The postman and the milkman, both being very busy men (i.e. having more than one delivery that must be made), cannot wait an arbitrarily long time for her to make up her mind. They are somewhat indifferent as to which she answers first, just so long as she makes up her mind. But there she sits, steadfastly adhering to her rule, and unable to resolve which of the two events occurred; simultaneity, or non-simultaneity. Hence, this pathological event caused her system to "hang up".

Being a resourceful individual, she devised a means of making an arbitrary decision in such a circumstance. Her new rule was, when in doubt flip a coin, and abide by the outcome of the coin toss. However clever this may have seemed to her, in so doing she has inadvertently complicated, but not cured, the essential difficulty, for now she has to decide when she is in doubt, and when she isn't (i.e., when to use the coin, and when not to). Thus, again on that pathological Monday, we find the postman and the milkman, each at their respective doors, with the poor frustrated lady trying to decide whether or not to use the coin.

The poor lady keeps trying to change the rules, hoping that there is some magic formula that will not hang her up in indecision, but each time her efforts are frustrated. The name of this game has affectionately been called "musical glitch". The point of this story is that regardless of the set of rules that is made to resolve all possible cases of two independent events into one of two groups, there is always a pathological case that will "hang up" your system for an arbitrarily long, but finite amount of time.

First, we will attempt to prove that the above statements are true in general, and second, we will try to show that living with this fact, one may still devise a "glitchless" system that will eventually make a firm decision in a theoretically unspecifiable length of time. It should be emphasized that the only assumption made in this treatment is that all phenomena considered exhibit essentially continuous behavior. However, it should be noted that the same arguments apply even for discrete particle behavior. Nevertheless, this case will be ignored for the present.

9.2 THE GLITCH PHENOMENON

Throughout this analysis, we shall speak in terms of energy, as this is the fundamental unit of both the electrical and mechanical analogies of which we will speak. This analysis is only applicable to ergodic systems, or those systems for which a potential energy curve is applicable. Let us begin with some fundamental definitions:

1. A stable state is a relative minimum in the Potential Energy function.

2. A metastable state is a relative maximum or inflection point in the Potential Energy function (i.e., any point of zero gradient that is not a relative minimum).

Theorem I: Between any two stable states, there exists at least one metastable state.

Assumption: All systems found in nature are essentially continuous* (i.e., the potential curve is continuous at all points).

Lemma: Between any two relative minimums, there must be a relative maximum.

Theorem I follows by definition from the Lemma, hence it is sufficient to prove the Lemma. However, the Lemma follows directly from Rolle's Theorem [26], and is also obvious by induction.

*In the macroworld of electronics, discrete electrons are considered in such numbers that the statistical behavior of the system may be considered continuous.

Thus, by choosing our definitions judiciously, a basic truth becomes immediately apparent: All devices having two stable states have a third metastable state in between.

Noiseless, Dissipationless Case

We shall begin our analysis by examining the noiseless, dissipationless case. First we need to add a few more definitions to our vocabulary.

3. The metastable point is that point of zero potential gradient such that should the system be displaced from this point by an amount δ, where δ is vanishingly small, the system will pass onto another state. (See Figs. 13 and 15.)

4. The metastable line is the local continuum formed by all points of zero gradient, terminated on at least one side by a line of negative gradient. (That point of metastable line termination may be considered a metastable point. See Figs. 14 and 15.)

5. The metastable region is that region about the metastable point or line in which, for the purposes of measurement, it is impossible to resolve whether the system is on the metastable point (or line), or not.

Now that we've shown that there exists a metastable state between every pair of stable states, we shall try to demonstrate that it is possible to become "hung up" in the metastable region for an arbitrarily long period of time, depending on the kinetic energy of the system as it passes through the metastable state. We may now generate a Taylor Series expansion about a point in the metastable region. The only restriction on the series expansion is that there be no more terms in the expansion than there are continuous derivatives in the function. Hence, we require, for a first-order approximation, that the potential function and its first derivative be continuous at that point. To do this, we again fall back on our one assumption of nature exhibiting continuous behavior in the macroworld of which we are speaking. The point which we pick will be the metastable point, or the end point on the metastable line. If we define our coordinate system about this point, then our Taylor series reduces to a simple MacLaurin series:

$$P(x) = P(0) + P'(0)\cdot x/1! + P''(0)\cdot x^2/2! + \ldots. \qquad (1)$$

Since the force on the system is given by the gradient of the potential, the acceleration, or second derivative of x with respect to time, is proportional to the gradient of the potential:

$$\ddot{x} = -k\cdot\frac{dP}{dx} = -k[P'(0) + P''(0)\cdot x + \ldots.] \qquad (2)$$

Let us examine the case where the metastable state is a relative maximum. (See Fig. 13.) In this case the second derivative in the metastable region is negative, except on the metastable line (see Figs. 14 and 15), while the first derivative is zero. Thus, let us define a positive constant, a= stable

b= metastable
(point)

c= stable

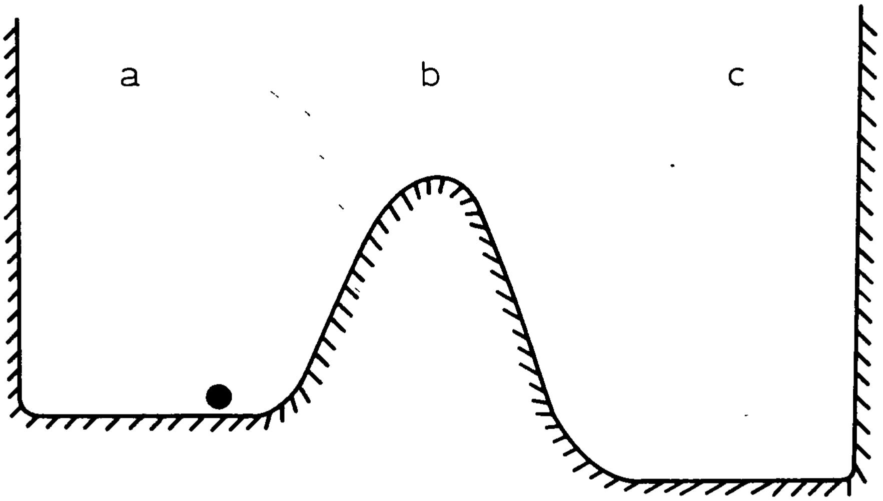


Figure 13. Three-state system with metastable point.

a= stable

b= metastable
(line)

c= stable

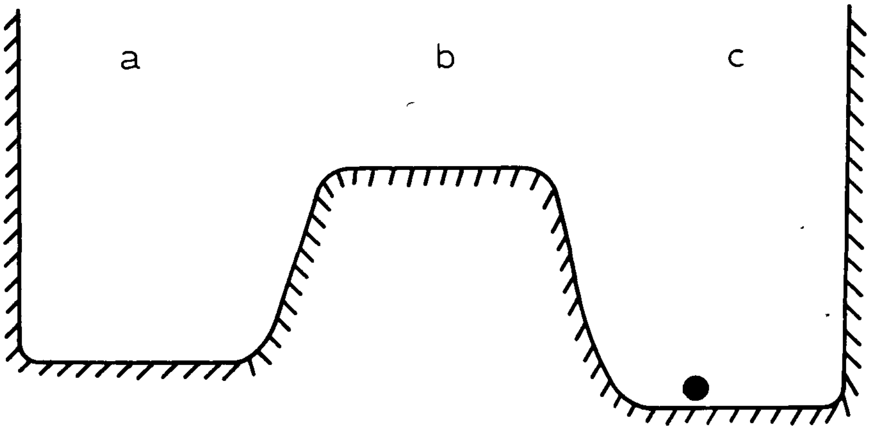


Figure 14. Three-state system with metastable line.

a= stable

b= metastable
(line)

c= metastable
(point)

d= stable

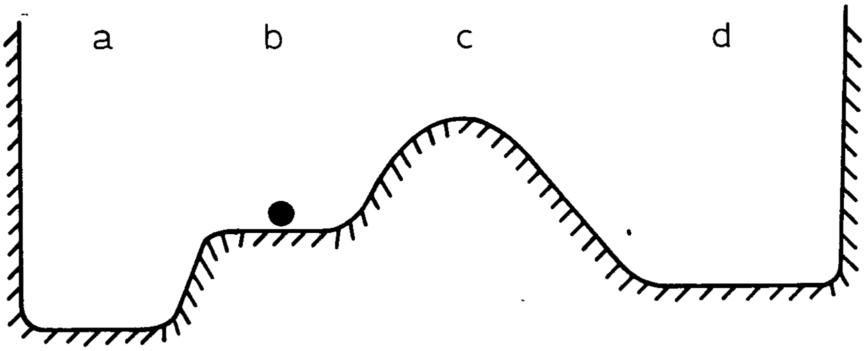


Figure 15. Three-state system with metastable point and metastable line.

$$K^2 = -k \cdot P''(0). \tag{3}$$

Hence:

$$\ddot{x} = K^2 x + \ldots. \tag{4}$$

within the metastable region. As usual, we shall throw out the higher-order terms as negligible for very small values of x. The general solution to this second-order differential equation is the familiar hyperbolic sine and cosine function:

$$x = A \sinh Kt + B \cosh Kt. \tag{5}$$

The diverging quality that characterizes the metastable state is immediately apparent in the solution. We shall now consider the boundary conditions for the only two possible situations that a continuous world will allow.

I. System initially at rest at $x = x_o$:

Then, $\dot{x}(0) = 0 = A$ and $x(0) = x_o + B$. Thus, the specific solution for this case is

$$x = x_o \cdot \cosh Kt. \tag{6}$$

Now, we ask, how long will it take for the system to reach the edge of the metastable region, $x = x_m$? Thus, solving for t_m, we find:

$$t_m = \frac{1}{K} \cdot \cosh^{-1}\left(\frac{x_m}{x_o}\right). \tag{7}$$

Note that for x_o arbitrarily small, t_m will be arbitrarily large. In the limit, we see:

$$\lim_{x_o \to 0} \frac{1}{K} \cdot \cosh^{-1}\left(\frac{x_m}{x_o}\right) = \infty. \tag{8}$$

This simply says that if we should start the system with zero initial velocity on the metastable point, it will stay there ad infinitum. However, the probability of being able to locate a system at a predetermined point in a continuous line is zero: hence, we can only approach this limit, we can never, in fact, reach it. This particular case is characteristic of the case where a ball rolls up a hill, but never quite makes it to the top. Somewhere, just before the top, it comes to rest. It is at this point that we start our clock. We may conclude that depending upon how close the system gets to the metastable point, the time to get out of the metastable region will be arbitrarily long.

II. System having initial velocity v_o as it passes through the metastable "point at $t = 0$":

Then, $x(0) = 0 = B$ and $\dot{x}(0) = v_o = A \cdot K$. Thus, the specific solution for this case is

$$x = \frac{v_o}{K} \cdot \sinh Kt. \tag{9}$$

Now, we ask, how long will it take for the system to reach the edge of the metastable region, $x = x_m$? Thus, solving for t_m, we find:

$$t_m = \frac{1}{K} \cdot \sinh^{-1}\left(\frac{Kx_m}{v_o}\right). \tag{10}$$

Note that for v_o arbitrarily small, t_m will be arbitrarily large. In the limit, we see:

$$\lim_{v_o \to 0} \frac{1}{K} \cdot \sinh^{-1}\left(\frac{Kx_m}{v_o}\right) = \infty. \tag{11}$$

Again, this simply says that if we should start the system with zero initial velocity on the metastable point, that it will stay there ad infinitum. This case describes the situation where the ball has gone over the top of the hill, and we now wish to know how long it will take to finally get out of the metastable region.

Thus, cases I and II describe all possible situations that can occur in the transition between two stable states. Either it starts from state A to state B (see Figure 13), but does not make it over the potential hill, or it does make it over the hill, with a finite velocity. In either case, the system may become "hung up" in the metastable region for an arbitrarily long time, depending only on the initial conditions of the system. Thus, it must be concluded that it is impossible to place a maximum time limit on the length of time that might be spent in the metastable region. One can only assign probabilities as to the length of time that might be spent, depending on the distribution of initial conditions.

Dissipative Case

We shall now complicate the picture slightly by adding dissipation to the system. By adding damping (friction proportional to the speed of the system: $F = -\dot{x}$), we allow the system to come to rest at the bottom of one of the stable states. If we critically dampen the system, then it should come to rest monotonically (i.e., with no overshoot or oscillation). However, it can be shown that this will not affect the overall performance of the system, as outlined in the previous section, within the metastable region except to modify the argument of the hyperbolic functions. However, there is one region of performance which is radically changed, and that is the performance on a hypothetical metastable line. Given that it is possible to create a line of zero potential for a finite length, it would indeed be possible to start the system on this line. Any initial velocity the system would be given could be entirely dissipated before the system reached the edge of the metastable line (see Figure 14). Hence, in the absence of noise, the system could indeed come to rest on the metastable line. However, we are rescued from this situation by the reality of noise. Thus, by adding dissipation to the system, we have further forced ourselves to analyze the noisy situation.

Noisy Dissipative Case

Let us look at our metastable line once again. The equation of motion on this line is now:

$$\ddot{x} = -f\cdot\dot{x} + A(t), \qquad (12)$$

where f is the coefficient of moving friction and A(t) is a gaussian random noise term with a mean power value of g_o*. The solution to this equation was proposed by Einstein in 1905 [27]. This essentially describes the motion of a free particle in the presence of noise. This random walk phenomenom is described by chemists as Brownian Motion. If our particle started at the origin, then the expected mean will remain the origin as t goes to infinity. However, the standard deviation will diverge to infinity. Einstein's famous equation for the standard deviation is:

$$\overline{x^2} = \frac{2g_o}{f}\, t \qquad (13)$$

Thus, surprisingly enough, the system can be expected to diverge from the origin with a variance linearly dependent upon time. Hence, if we begin our system in the center of a metastable line of length 2L, then the expected time that the system will reach the edge is given by:

$$\overline{t_L} = \frac{f}{2g_o} \cdot L^2 \qquad (14)$$

Now, the only remaining question is how noise will affect the motion of the system in general. Thus, we will now solve for the generalized noisy damped case:

$$\ddot{x} = -f\cdot\dot{x} + c\cdot x + A(t) \qquad (15)$$

Again, we are solving this within the metastable region where our approximation of a potential function of the nature of equation (1) is valid. The solution to this equation is as follows:

<u>Variance</u>: $$\overline{x^2} = \frac{g_o}{c} + \exp(-ft)\cdot(x_o^2 + \frac{go}{c})\cdot(\cosh w't + \frac{f}{2w'}\sinh w't)^2 \qquad (16)$$

*In the case of purely thermal noise: g_o = KT, where T is the absolute temperature, and K is Boltzmann's Constant.

Mean: $$\overline{x} = x_o \exp(-f/2)\cdot(\cosh w't + \frac{f}{2w'} \sinh 2't) \qquad (17)$$

where $x_o = x(t=0)$ $\qquad w' = \sqrt{\frac{f^2}{4} + c}$ $\qquad \dot{x}(0) = 0$

The interesting thing to note from the statistics of the solution is the fact that although the variance is dependent upon the noise power, the expected value of x is completely independent of the noise power. In other words, the expected performance of the system is identical to that in the absence of noise. However, one must qualify the statement to the effect that the noise energy should never exceed the height of the potential barrier that separates the two states, or the system will lose all determinism whatsoever.

Thus, the total effect of adding noise to the system was to rescue the theory from the metastable line case, but not change the general performance of the system as originally outlined in the noiseless case.

Trinary Solution

Summarizing, we have shown that between any two stable states there exists a metastable state through which it may take an arbitrary length of time to pass, depending on the initial conditions. If your system does not allow ambiguous decisions, but does allow an arbitrary length of time to make a decision, then there is a solution to the problem. Simply stated, the solution is to indicate a delay while the decision is being made. Once the device is entirely committed to one of two alternatives, then the device should be interrogated as to its decision. In this way, there is no danger of the device ever yielding an ambiguous decision. Needless to say, while the device is in the Paused mode, the system is "hung up", but this is a price that one must pay for absolute certainty (in the absence of noise).

This solution is implied from the realization that three states, not two, exist. Hence, we redefine our two-state device as a three-state device (see Figure 16). The new state will be the Paused state, and its area must at least include the metastable region, but beyond that restriction the boundaries are arbitrary. Let us now look at a Boolean transition table, to see how we might encode this. (See Figure 17.)

Note that our transition points, x_a and x_b, are judiciously chosen so that there is a finite gradient passing through them. This insures the fact that the system cannot get hung up at the transition points. Note also that the coding is essentially a Gray code in that no races can possibly exist. The nature of the gradient is such that the system may pass from A to P, and then back to A again; however, once it has left P for B it must go all the way. This becomes clear if one considers the ball and hill analogy. You can roll the ball so that it doesn't quite make it over the hill and rolls back again, but once over the hill it keeps going. In traversing from A to B, there is absolute certainty that once the ball passes over the summit, it will continue on to B.

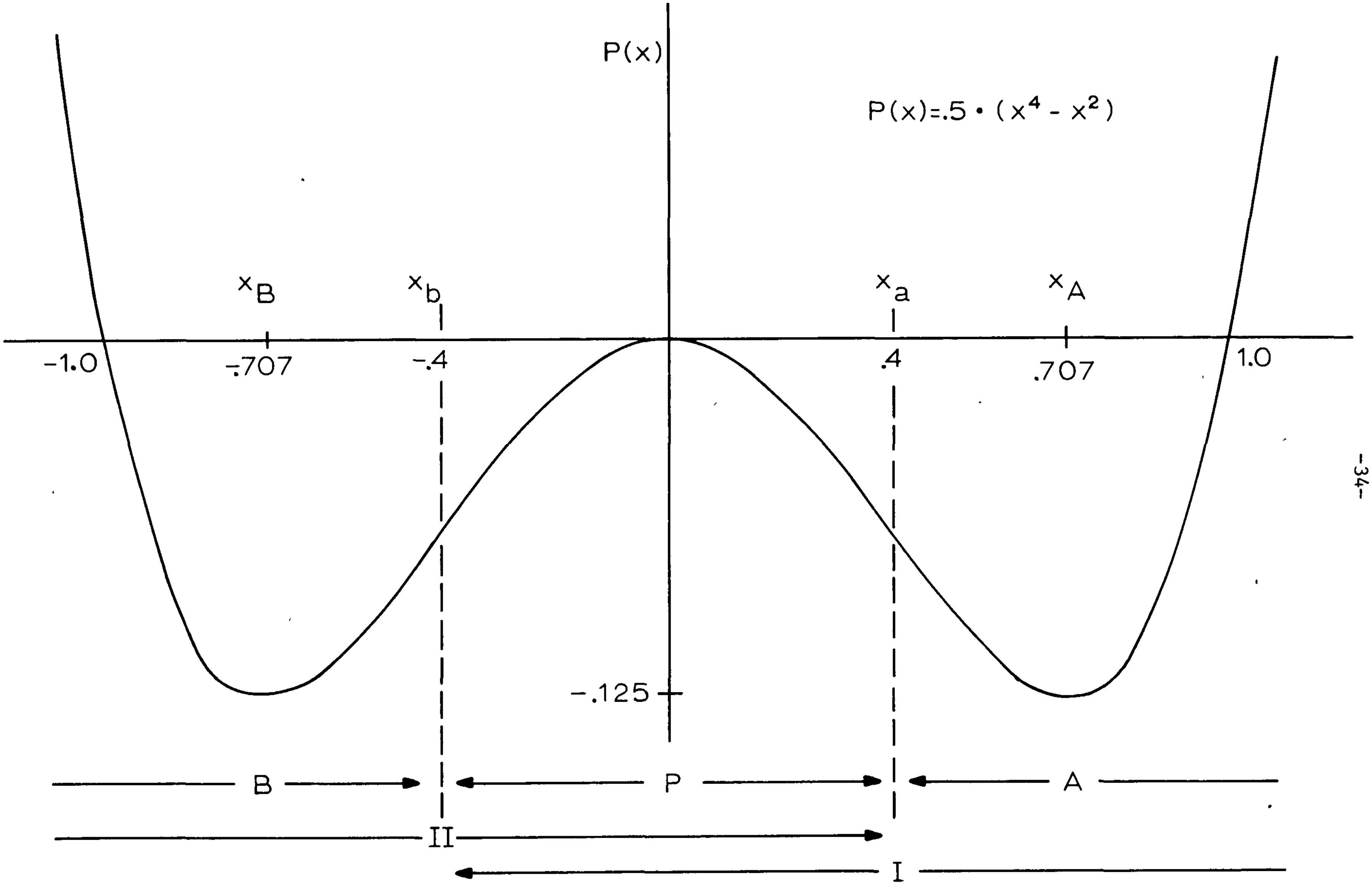
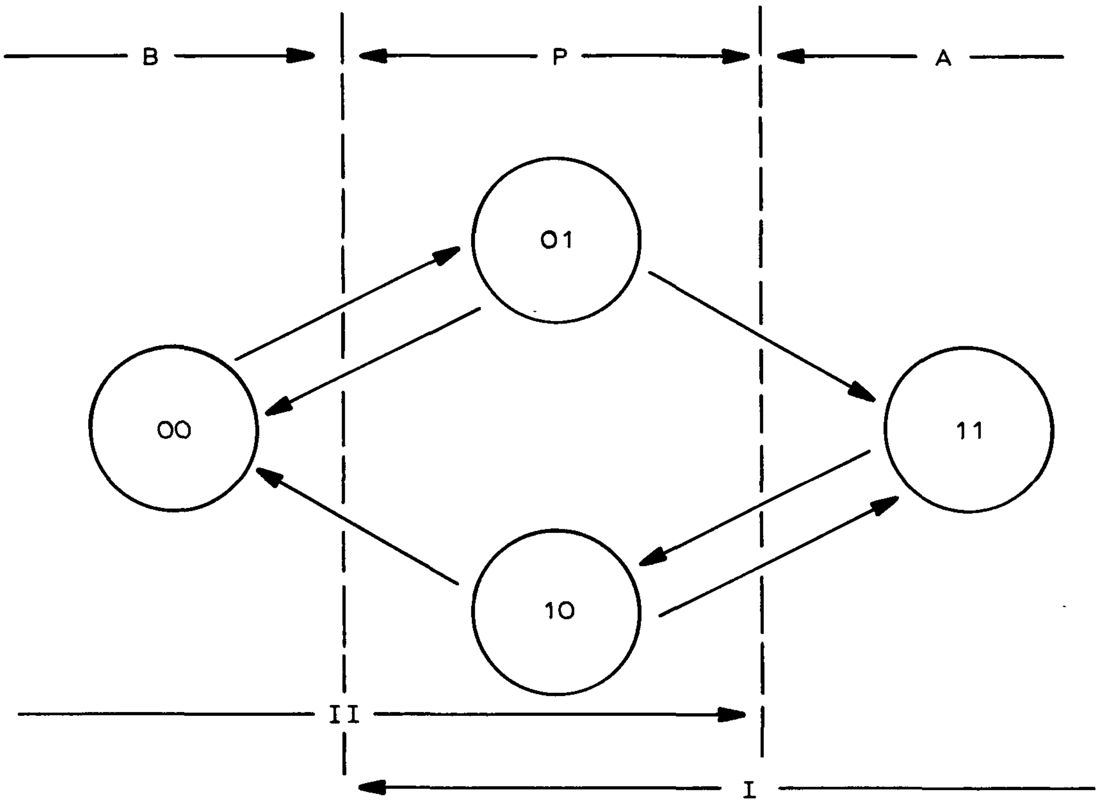


Figure 16. P(x) for three-state system.



STATE	CODING	TRUTH TABLE		
		I	II	
A	11	1	0	
P	01	1	1	$A = I \cdot \overline{II}$
P	10	1	1	$P = I \cdot II$
B	00	0	1	$B = \overline{I} \cdot II$

Figure 17. State diagram.

There is one matter left to be pointed out, and that is the following: There is a finite maximum delay between the initiation of the transition and the time that it takes to reach the Paused state. If the system has not reached the Paused state by this time, one can be absolutely certain that it never will, and hence one can assume that the system will continue to remain in the same state. Given the equation of motion of the system, this maximum time is easy to compute.

Then let $x = f(t)$, where $x(t=0) = x_a$. Then

$$t_{max} = f^{-1}(x_A) - f^{-1}(x_a). \tag{18}$$

To aid in the understanding of the previous analysis, an example has been worked out for the case of a simple "ideal" flip-flop.

Example: It is desired to make an asequential gate. This gate will be designed such that a pulse of known length will be gated against a level whose value (Boolean) as a function of time is statistically independent of the pulse. The output of this will be a pulse of defined length on one of two separate lines, depending on the value of the level at the time of gating. The schematic of this system is shown in Figure 18. It should be quickly noted that the successful performance of this system depends solely on the nature of the trinary flip-flop. It is this flip-flop that must resolve any ambiguities that arise in the relative timing of the two events, Y and Z. This flip-flop also determines the values of the delays D_1 and D_2. D_3 is only constrained to have a duration longer than the value of the pulse generated by the Pulse Generator, PG.

Let us now thoroughly examine our trinary flip-flop. We will assume it to contain an "ideal" flip-flop with a potential curve as shown in Figure 16:

$$P(x) = .5(x^4 - x^2) + P_o. \tag{19}$$

Note that this potential function displays all characteristics that we would desire in a flip-flop; it is continuous and symmetric, containing only two stable states with one unavoidable metastable state, and a fast-rising potential wall bounding the stable states at the extremes. For convenience, we will arbitrarily define $P_o = 0$. We will define a voltage of $x \geq .4$ volts to be state A, and a voltage of $x \leq -.4$ volts to be state B. State P is defined for $-.4<x<.4$ volts. Hence, the offset voltage indicated in Figure 19 will be .4 volts. The equation of motion for the dissipationless system is simply given by:

$$\ddot{x} = -k^2 \cdot \text{grad}(P(x)) = k^2x - 2k^2x^3. \tag{20}$$

State of the art integrated flip-flops have a k in the order of 10^8. The equation of motion contains the dissipation term whose value was chosen to critically dampen the system:

$$\ddot{x} = k^2x - 2k^2x^3 - kf\dot{x}. \tag{21}$$

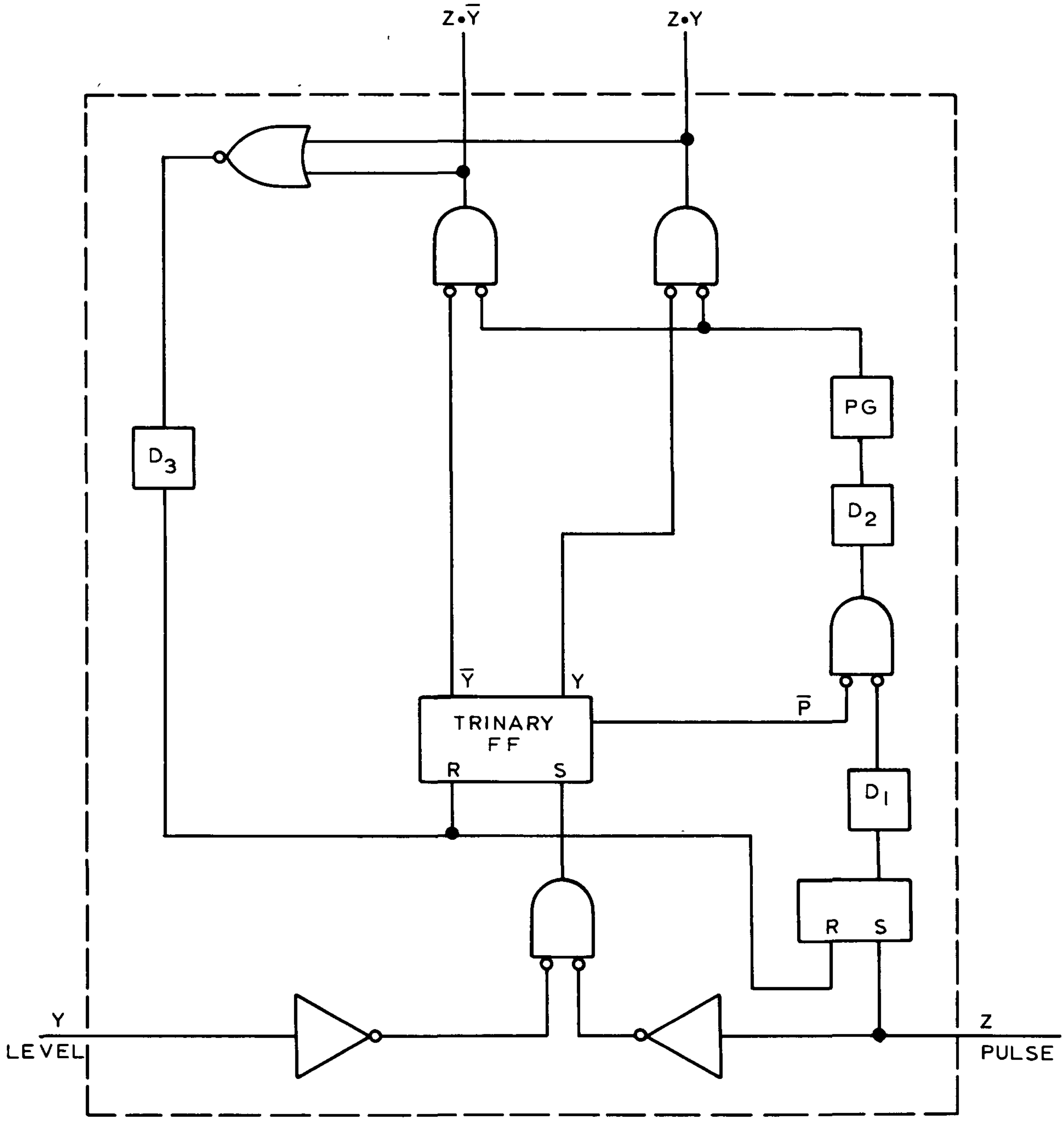


Figure 18. Asequential gate.

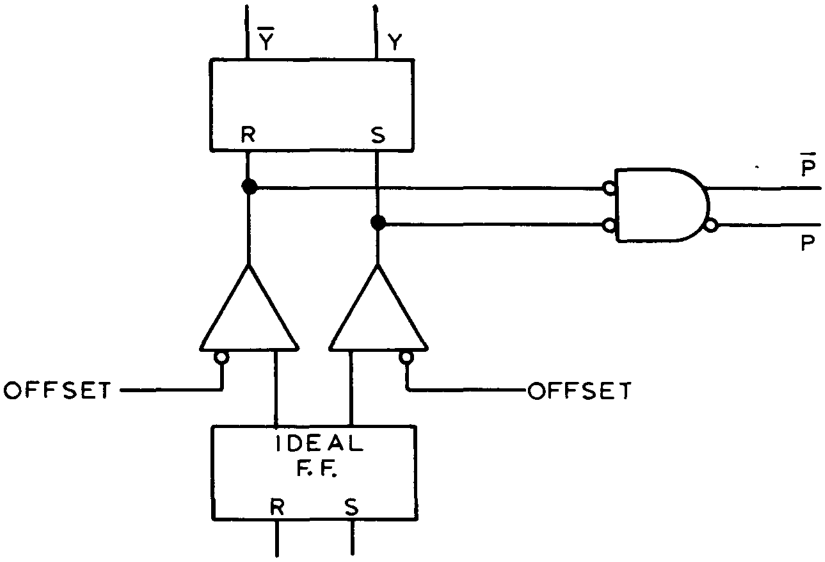


Figure 19. Trinary flip-flop.

For the purposes of critical damping, the coefficient of viscous friction is dependent on the initial conditions. We will choose our f such that the case for maximum initial velocity is critically damped. Thus, for any case having a smaller value of initial velocity, the system will be overdamped. In this manner, we can be certain that the states will be approached monotonically. Thus, we will arbitrarily choose our maximum initial velocity to be ten times that necessary to carry the system over the transition barrier (i.e., the maximum initial energy impulse is to be 100 times that necessary to pass over the barrier) in the absence of friction. The value of f necessary to critically dampen the system under these conditions was found to be f = 3.

The first problem is to determine what the value of delay D_1 should be. This delay time should be sufficiently long that one can be absolutely sure that if the system is going to reach the Paused state at all, it will have done so by this time; in which case it will either be in the Paused state, or will have passed out of it, having made up its mind. We can determine what an outer bound on this time is by simply seeing how long the undamped case takes to pass from x_a to x_A given zero initial velocity. We know that this represents an upper bound since with friction, the system must start at a much higher velocity at x_A in order to reach x_a at all. Consequently, it will reach x_a much more quickly with damping than without. Note that this is only true going uphill; the reverse is true going downhill. Thus, it will suffice to simply solve equation 20. The general solution to this equation is

$$x = \sqrt{\frac{2+K}{2k}} \cdot \mathrm{cn}\,(\lambda t+\phi \mid m_1),\ m_1 = \frac{\lambda+k}{2\lambda^2},\ \text{for } \lambda \geq k \tag{22}$$

$$x = \sqrt{\frac{2\lambda^2}{2k}} \cdot \mathrm{dn}\,(\lambda t+\phi \mid m_2),\ m_2 = \frac{2\lambda^2-k}{\lambda^2},\ \text{for } \lambda \leq k\,. \tag{23}$$

These solutions are elliptic functions [28,29] where λ and φ are determined from the initial conditions. The dn(u) function represents the trapped case where the system is oscillating about either x_A or x_B. The cn(u) function represents the untrapped case where the system contains sufficient kinetic energy to pass over the potential barrier separating the two states. To determine the upper bound on D_1, we are considering the trapped case with the following initial conditions:

$$x(0) = x_a = .4 \qquad \dot{x}(0) = 0\,.$$

Therefore, solving for λ and φ, we find:

$$x = .9\ \mathrm{dn}(.83\,kt + 2.26) \text{ with a squared modulus: } m = \frac{2\lambda^2-k}{\lambda^2} = .8\,. \tag{24}$$

Now, we wish to know the traversal time from $x = x_a = .4$ to $x = x_A = .707$.

$$t_{max} = f^{-1}(x_A) - f^{-1}(x_a) = \frac{1}{.83k;}\ \frac{dn^{-1}(.707)}{.9} - \frac{2.26}{83k} = \frac{1.72}{k} \qquad (25)$$

For $k = 10^8$, then, $t_{max} = 17.2$ ns $= D_1$.

The only constraint on D_1 is that it be long enough to insure that the final output of the trinary flip-flop has switched from the Paused state to either A or B. Observing Figure 19, you will notice that this amounts to the transition time of the upper flip-flop. Since this is being set by a level change, and not a pulse, we can read the maximum transition time off the specification sheet, thus defining D_2.

D_3 need be no longer than the length of the pulse from the pulse generator. Thus, if the desired output is a 100-ns pulse, D_3 should also be set to 100 ns. It should be noted that this circuit can only be called again after the trinary flip-flop is reset. This essentially clears the circuit, thus readying it for reuse.

Now that we've defined the parameters of the circuit, we need only ask what the expected delay time of this circuit will be, and more generally, what the delay probability density will look like. Let us begin by computing the minimum delay time of the system. Observing Figure 18, we can see that the total delay time is given by:

$$T_{tot} = t_{ff} + D_1 + t_g + t_{trinary} + D_2 + t_{pg} + t_g\ , \qquad (26)$$

where t_{ff} = flip-flop delay, t_g = gate delay, and t_{pg} = pulse generator delay. The only undetermined delay left is the trinary flip-flop delay. From Figure 19, we can see that this can be further broken down into:

$$t_{trinary} = t_{ff} + t_g + t_{idealff}\,. \qquad (27)$$

Thus, what we wish to compute is the delay probability distribution of the ideal flip-flop.

The maximum expected delay time within T_m seconds is a function of both the parameters of the flip-flop and the shape and length of the input pulse which fires the flip-flop. For our example, the maximum expected delay time is given by:

$$t_{max}(T_m) \approx 26 \cdot \ln(N \cdot T_m \cdot t_o/23) \text{ ns, for } N \cdot T_m \cdot t_o >> 23, \qquad (28)$$

where: T_m = mean time between glitches

n = duty rate of the ideal flip-flop given in #/sec
t_o = transition time of the input pulse.

A glitch is said to occur whenever the delay is greater than t_{max} seconds.

The minimum transition time was found to be about 23 ns in any case. Thus, using this approximate formula, one may now specify a mean time between glitches. If one designs a circuit in which a delay of $t > t_{max}$ causes a failure, then using equation 28 one can specify the mean failure rate. If, however, one designs his system such that it does nothing until the asequential gate makes up its mind, then no failure will ever occur; and one may simply use equation 28 to find the mean time between delays greater than t_{max}. The most important point to note here is that the mean time between glitches is an exponentially increasing function of the expected maximum delay time. Thus, if one specifies a t_{max} long enough, one can quickly exceed the expected lifetime of the integrated circuit itself.

9.3 SUMMARY

We have shown that the glitch problem will arise whenever a Boolean decision must be made regarding two independent events. Thus, in general, we may say that this problem is inherent in the interaction between two asequential machines. The greater the interaction, the more prevalent the problem. Up to this time, computers have been essentially sequential machines, with this problem only arising between the central processor and the input-output equipment. However, with the advent of parallel mode computers, the problem has become more and more pressing.

We have shown that this problem may be overcome if within each sequential machine, the operation is paused until the trinary flip-flop has made its decision. Thus, this problem may be overcome as long as we put no time limit on the decision-making process.

So, let us look once more at the waiting woman. Her problem is one of adjustment to the fact that once every pathological Monday, the decision-making process may take an arbitrarily long time. Thus, as long as the postman and milkman realize that this will happen only once in a Blue Moon, they should be content to wait out her decision every time that Blue Moon does occur.

10. APPENDIX B: PHOTOGRAPHING FLIP-FLOPS IN MIDDLE REGION

The experimental evidence, both the photographs and the probability of escape data, presented in this section required a special test set-up to repeatably drive the flip-flop under test into the middle region. To achieve these requirements, the set-up had to produce two pulses that had smooth rise and fall transitions which could be varied in time with respect to each other, including the ability to "fine tune" the timing between the pulses over a 10-psec interval.

A diagram of the circuit used to drive the flip-flop under test into the middle region is shown in Figure 20. This system consists of a pulse generator, a circuit that produces two parallel output pulses from one pulse generator pulse (the "two for one box"), a set of coaxial cables and air lines of different lengths, and an adjustable air line for fine tuning. All signals are passed via coaxial cable with proper line terminations. Most of the cable beyond the "two for one" pulse producing box uses General Radio type GR874 connectors to minimize line impedance discontinuities. For ECL type flip-flop testing, the pulse generator is adjusted to provide a -0.8V pulse. A +4.5V pulse is used for testing TTL type flip-flops. The TTL "two for one" box will, with the transistors used, follow pulses with as short as 3-nsec rise and fall times. The ECL box will follow 2-nsec rise and fall times.

This setup is adequate to produce the sampling oscilloscope photographs. However, additional circuitry is required to produce the single trajectory photographs. A block diagram of this circuitry is shown in Figure 21. A circuit of the type shown in Figure 10 is used to detect the logically undefined state. FF2 is set if the flip-flop under test remains in the undefined region longer than the time for which D3 is adjusted. If FF2 is set, the oscilloscope will be triggered by the delayed (thru D5) pulse generator pulse. The 50-ft. cable delay line, which is driven from an active oscilloscope probe (a Tektronix P6045 F.E.T. probe), is needed to compensate for the triggering circuit delay. The dashed-in D1 delay is required, if the flip-flop under test has a transition-sensitive clock input or if the flip-flop is an R-S type, to reset the flip-flop after each event. For R-S type flip-flops, the C and D signals are fed into an AND gate, with either C or D inverted. The output of the AND gate is connected to the R input of the flip-flop.

Adjusting the test setup involves connecting an oscilloscope to the output of the flip-flop under test and adjusting the lengths of the input coaxial cables until a runt pulse is produced by the flip-flop. The adjustable air line can then normally be used to produce the desired results. There are two general guidelines which make the adjustment easier.

1. Use a low duty cycle and always set the flip-flop under test to one state for most of the period, so that all internal conditions, due to both temperature and voltage changes, can settle before the next input pulse is applied. For TTL circuits, a pulse rate of 2 to 10 KHz with a pulse width of only 200 nsec may be required. ECL flip-flops usually function properly with pulse rates of up to 500 KHz.

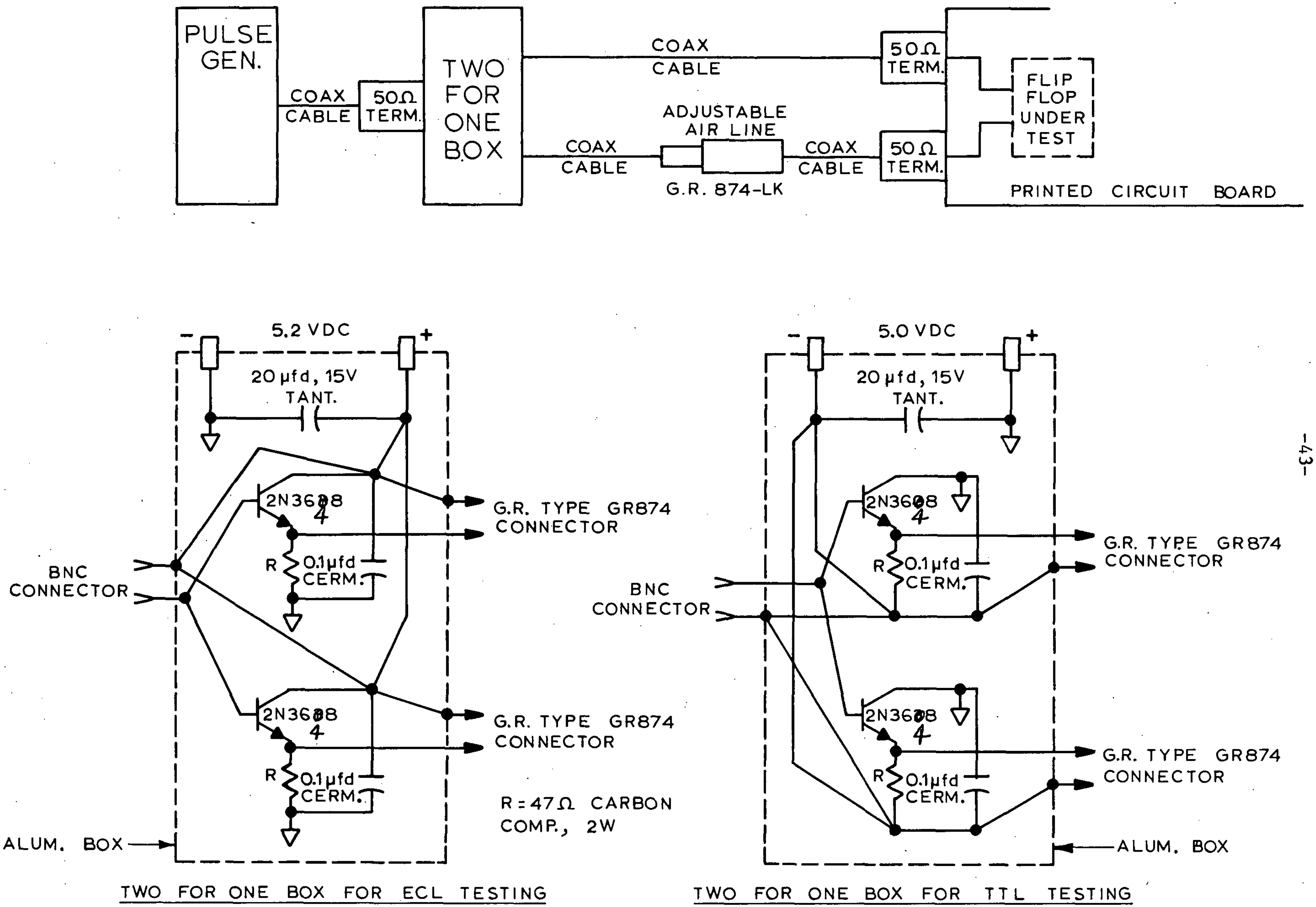


Figure 20. Test set-up to repetitively drive flip-flops into middle region, with details of "two for one" boxes.

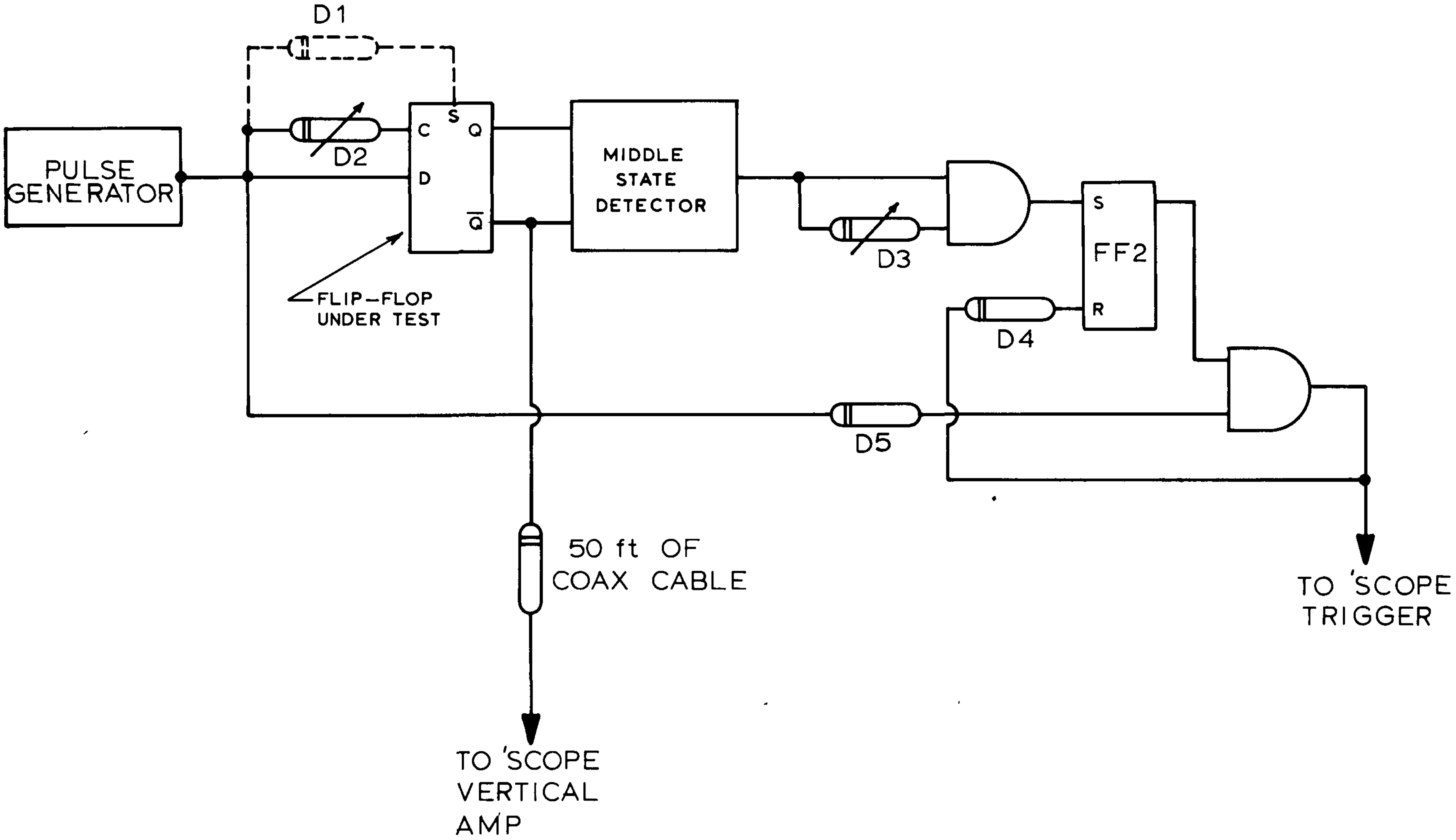


Figure 21. Block diagram of circuitry for producing single-trajectory photographs.

2. TTL flip-flops, with active pull-up output stages, should have a 2KΩ to 5KΩ resistor connected from each flip-flop output to ground. These resistors are needed to prevent the output that is high from being so high that the output pull-up transistor is reverse biased.

11. APPENDIX C: OBTAINING PROBABILITY OF ESCAPE DATA

The test set-up for obtaining data to calculate the data values used to plot the curves in Figure 7 is shown in Figure 22. The flip-flop under test is driven with the same circuit, described in Appendix B, that is used to produce photographs. The circuit in Figure 22 with the output going to the power supply sense circuit is used as feedback to compensate for drift. The conditions necessary to cause the flip-flop to produce long resolving times are critical enough that a small change in power supply voltage (10 to 20 millivolts typically) can change the flip-flop from never switching to always switching. To verify that this feedback does not affect the results, tests were conducted with and without the feedback connected,with different size filter capacitors, and with the feedback biased to produce mostly "ones", then mostly "zeros". The results of all these tests were, within the accuracy given in Table 1, identical.

In Figure 22, the delay line (made of a series of gates), the AND gates, and the latches comprise a circuit that provides a pulse to a counter if the flip-flop under test stayed in the undefined region for a minimum length of time. The last counter, N6, is a check counter to determine the number of pulses that did not propagate through the delay line gates during a given run. This last counter is required because the output of the Middle State Detector can be a marginal pulse.

After the test circuit is allowed to run for a period of time, the numbers in each of 6 counters are recorded. The number of times the flip-flop settled in the time interval between N3 and N4 is (N3-N4). The ratios of the number of settlings during different time intervals is the slope of the probability of escape curve. If we assume the probability of no escape is an exponential function of time, then the probability of no escape curve will be a straight line on a semilog plot, or:

$$\text{Slope} = \ln \frac{N2 - N3}{N3 - N4}.$$

To allow the use of this equation, the time intervals between the taps on the gate delay line were carefully adjusted to be equal.

Several test runs were made, each run with a different propagation time adjustment of the first 4 gates (the dashed-in adjustable delay line) until there were 10 to 20 data points over the 8-decade plot shown in Figure 7. The counter values from each test run were normalized against the count in counter No. 6; the data points were then calculated and plotted as Figure 7.

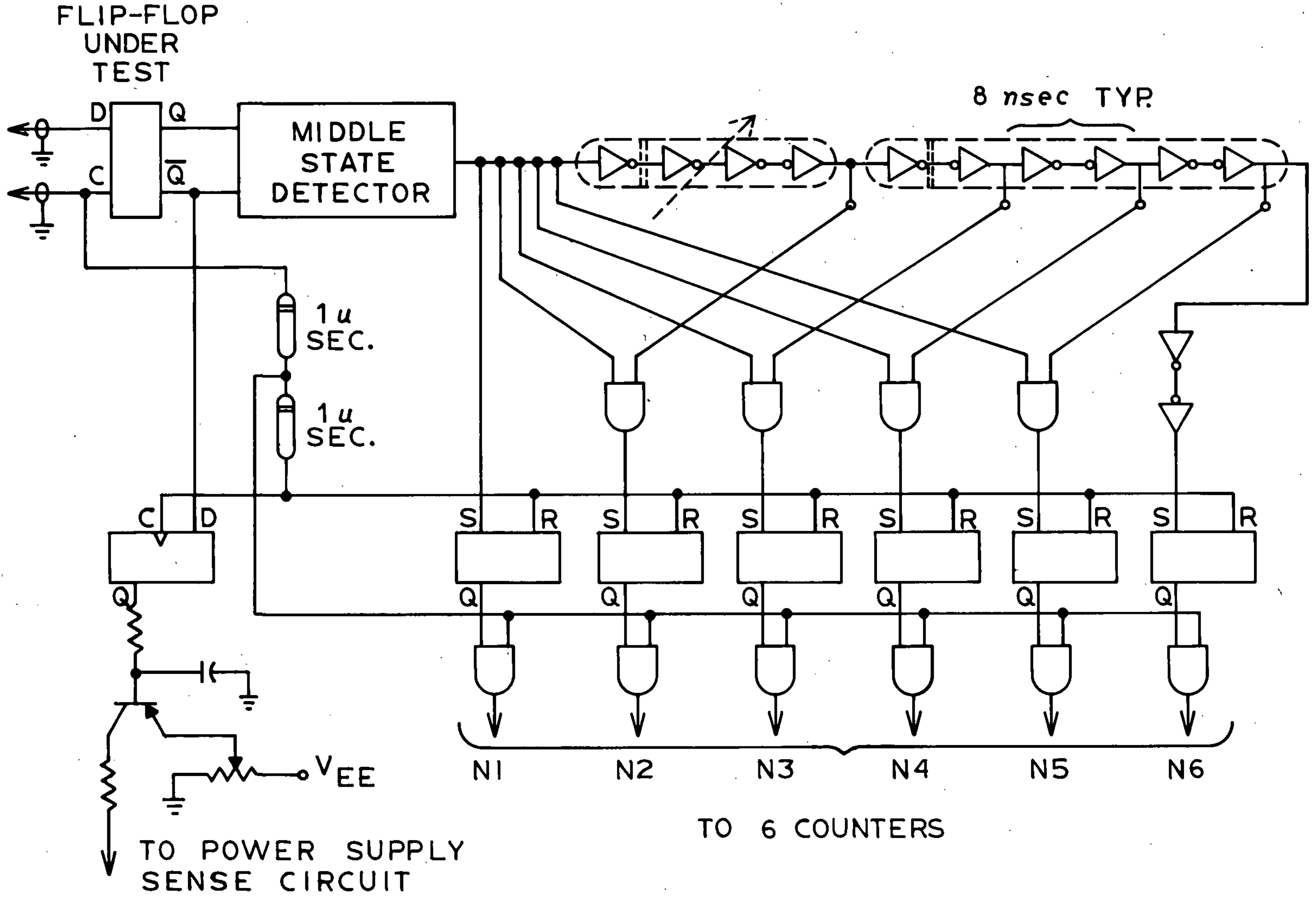


Figure 22.` Test set-up for obtaining probability of escape data.

12. APPENDIX D: SCHMITT TRIGGER ACTION IN TTL GATES

Figure 23 illustrates the circuit response of a TTL gate of the type shown in Figure 24 to an input signal which begins at ground and is slowly increased towards V_{CC}. The hysteresis region corresponds to the input voltage range V_1-V_2. For an increasing input voltage, the output follows the path A,B,C,D. At D, transistors Q_2 and Q_3 are in the linear active region and Q_4 is in saturation. The diode D and the now forward-biased collector-base diode of Q_4 form a low impedance path between the emitters of Q_2 and Q_3. The circuit may now be viewed as a Schmitt trigger circuit. Further increase of the input voltage above point D results in regenerative action, which causes Q_3 to go more and more towards cutoff and Q_2 more into conduction. This regenerative action continues until Q_3 is cut off and Q_2 is saturated. Beyond this point in input voltage, the output is along E,F. If the input is decreased now, the output follows F,E,G. At G, Q_2 comes back into conduction and Q_2 enters the linear active region. Since Q_4 is still saturated, the regenerative action starts again and the output jumps to H. Further decrease of the input voltage makes the output follow the path C,B,A.

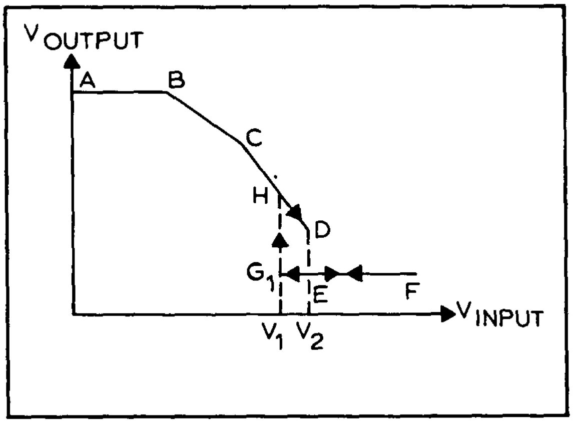


Figure 23. Transfer characteristic of SN7400.

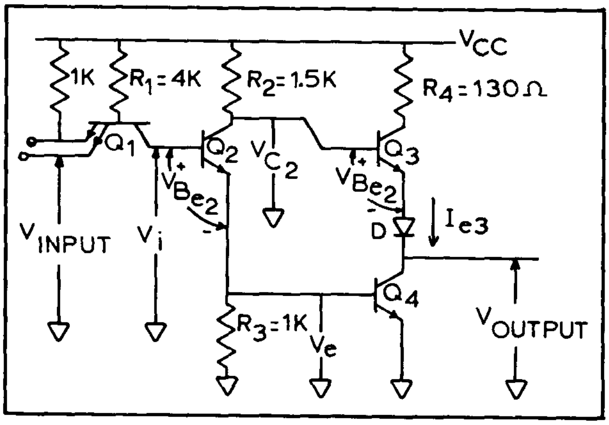


Figure 24. Schematic of 1/4 SN7400 (with one input active).

13. APPENDIX E: DEVELOPMENT OF EQUATIONS FOR TWO-FLIP-FLOP TYPE SYNCHRONIZER BEHAVIOR

The mean period, in seconds, for the output of FF1 of Figure 8 to not yet be resolved at the end of a clock period is (with some assumptions) the inverse of the product of the number of times per second FF1 enters the middle region at the beginning of the clock period times the conditional probability that the flip-flop is in the middle region at the end of the clock period given it enters this region at the start of the clock period.

The "input window" of a flip-flop is the range of differences in time between the inputs which can cause the flip-flop to enter the middle region. The number of times FF1 enters the middle region at the beginning of the clock period is dependent on this input window.

Experimental observations indicate that the time position of the inputs *within the input window* is independent of the output response of the flip-flop. If the input signal timing relationship is changed by varying the length of an adjustable air line while the output of the flip-flop is being observed on an oscilloscope, the flip-flop output seems to almost "jump" from never switching to sometimes switching with some long resolution times and then, with more air line adjustment, to always switching. All of this takes place with a change in length of the air line of approximately 1/4 cm. Light travels 1/4 cm. in approximately 10 psec.

Within the window, the flip-flop output response is then treated as being independent of the input condition, for 3 reasons. The first is the experimental observations; the second is that a mathematical model that would allow a better definition has not been developed; and the third is that, as related to the effect of inaccuracies of some of the other parameters, the inaccuracy of treating the window this way appears to have little effect on the accuracy of the resulting predicted system reliability.

Therefore, with this assumption, the *portion* of each second that the flip-flop can be caused to enter the middle region by the interrupt signal is: [FF1 Window (sec.)] [Clock Rate (PPS)].

If we assume that the arrival of *each* interrupt signal is statistically independent of the arrival of a clock signal and of all previous interrupt signals, and the time distribution of each interrupt signal is uniform in the *average* interval of $(\text{interrupt data rate})^{-1}$, then:

$$\left.\begin{array}{l}\text{Number of Entries}\\ \text{Into Middle Region}\\ \text{Per Second}\end{array}\right\} = (\text{FF1 Window})\cdot(\text{Clock Rate})\cdot(\text{Average Interrupt Rate})$$

The probability that FF1 is in the middle region at the end of a clock period, given that it entered the middle region at the start of the clock period, is derived from Figure 7 with the time (measured from the input clock transition) allowed for resolving being $(\text{Clock Rate})^{-1}$:

$$\text{Prob. Still In} = \exp\left(- \frac{1/\text{C.R.} - \text{D}}{\text{T}}\right),$$

where C.R. is the clock rate, and D and T are derived from experimental data (see Table 1).

The mean period (in seconds) for the output of FF1 of Figure 8 to not be resolved at the end of a clock period is then:

$$\begin{array}{l}\text{Mean Period (sec) for}\\ \text{Output Not Yet Resolved}\\ \text{At End of Clock Period}\end{array} = \frac{\exp\left(\frac{1/\text{C.R.} - \text{D}}{\text{T}}\right)}{(\Delta t)\ (\text{C.R.})\ (\text{Average I.R.})}$$

where C.R. is the clock rate in PPS, I.R. is the interrupt rate in PPS, Δt is the input "window" in seconds, and D and T, which are experimentally derived, have units of seconds.

14. ACKNOWLEDGMENTS

To a large extent, the "Glitch problem" has been the concern of the entire Laboratory staff, with many helpful suggestions and discussions coming from many different members. In addition to those whose names appear in the references, special thanks is given to Dr. Fred Rosenberger, who has made many helpful observations and suggestions; to Mr. David Shupe, who converted the faint oscilloscope photographs into publishable prints; and to Mrs. Christine Coaker, who provided assistance in the preparation of the text.

15. REFERENCES

1. W.M. Littlefield and T.J. Chaney, "The Glitch Phenomenon", Computer Systems Laboratory Technical Memorandum No. 10, Washington University, St. Louis, Mo., Dec. 1966.

2. W.M. Littlefield, "Interlocken", Computer Systems Laboratory Technical Memorandum No. 26, Washington University, St. Louis, Mo., June 1967.

3. K. Harada, "A Proposal on Interruption and Priority Control Schemes", Computer Systems Laboratory Technical Memorandum No. 36, Washington University, St. Louis, Mo., Aug. 1967.

4. G.R. Couranz, "An Unclocked Two-Line Interlock", Computer Systems Laboratory Technical Memorandum No. 55, Washington University, St. Louis, Mo., Feb. 1968.

5. K. Harada, "Diversified Interlock Module", Computer Systems Laboratory Technical Memorandum No. 63, Washington University, St. Louis, Mo., July 1968.

6. G.R. Couranz, An Analysis of Binary Circuits Under Marginal Triggering Conditions, Computer Systems Laboratory Technical Report No. 15, Washington University, St. Louis, Mo., Nov. 1969.

7. I. Catt, "Time Loss Through Gating of Asynchronous Logic Signal Pulses", IEEE Transactions on Electronic Computers (Short Notes), EC-15:108-111, Feb. 1966.

8. I. Catt, "My Experience With the Synchronizer Problem", presented at the Workshop on Synchronizer Failures, Washington University, St. Louis, Mo., April 27-28, 1972.

9. H. Gray, Digital Computer Engineering, Prentice-Hall, Englewood Cliffs, New Jersey, pp. 198-201, 1963.

10. T.J. Chaney, S.M. Ornstein, and W.M. Littlefield, "Beware the Synchronizer", in Digest of Papers -- COMPCON '72, IEEE Computer Society Conference, pp. 317-319, Sept. 1972.

11. T.J. Chaney and C.E. Molnar, "Anomalous Behavior of Synchronizer and Arbiter Circuits", IEEE Transactions on Computers, C-22:421-422, April 1973.

12. R. Srinivasan and T.J. Chaney, "On the Hysteresis of the TTL Circuit With Totem-Pole Output", Proceedings of the Eleventh Annual Allerton Conference on Circuit and System Theory, pp. 96-105, Oct. 1973.

13. P. Mars, "Study of the Probabilistic Behavior of Regenerative Switching Circuits", Proceedings of the IEEE, 115:642-668, May 1968.

14. D.J. Wheeler, "System Design for Non-Engineering Students and the Synchronization Problem", Proceedings of the Conference on Teaching of Computer Design, University of Newcastle upon Tyne, England, pp. 88-94, Sept. 1971.

15. C.L. Seitz, Personal Communication, Computer Science Department, The University of Utah, Salt Lake City, Utah, June 1972.

16. M. Hurtado, "Mathematical Study on the Effect of Noise in a Negative-Resistance Bistable Device", Computer Systems Laboratory Technical Memorandum No. 189, Washington University, St. Louis, Mo., April 1973.

17. Staff, New TTL MSI for Design Engineers, Texas Instruments Inc., Houston, pp. 15-19., 1972.

18. J. Raffel, R. Berger, and S. Pezaris, Personal Communications, M.I.T. Lincoln Laboratory, Lexington, Mass., Feb. 1973, Oct. 1973.

19. W.W. Plummer, "Asynchronous Arbiters", IEEE Transactions on Computers, C-21:37-42, Jan. 1972.

20. M. Pepper, "A Macromodular Interlock", presented at the Workshop on Synchronizer Failures, Washington University, St. Louis, Mo., April 27-28, 1972.

21. C.L. Seitz, Personal Communication, Computer Science Department, The University of Utah, Salt Lake City, Utah, Jan. 1972.

22. C.G. Bell, J. Grason, and A. Newell, Designing Computers and Digital Systems Using PDP-16 Register Transfer Modules, Digital Press, Digital Equipment Corp., Maynard, Mass., pp. 256-258, 1972.

23. D.J. Kinniment and D.B.G. Edwards, "Circuit Technology in a Large Computer System", The Radio and Electronic Engineer, 43:435-441, July 1973.

24. "Decisions, Decisions", Scientific American, 22 :43-44, April 1973.

25. R. Keller, "Towards a Theory of Universal Speed-Independent Modules", IEEE Transactions on Computers, C-23:21-33, Jan. 1974.

26. George B. Thomas, Jr., Calculus and Analytic Geometry, Addison-Wesley Publishing Co., Reading, Mass., 1960.

27. G.E. Uhlenbeck and S. Ornstein, "On the Theory of Brownian Motion", Physical Review, Vol. 36, Sept. 1, 1930, from Selected Papers on Noise and Stochastic Processes, Dover Publications, New York, 1954.

28. W.J. Cunningham, Introduction to Nonlinear Analysis, McGraw-Hill, New York, 1938.

29. L.N. Milne-Thomson, Jacobian Elliptic Function Tables, Dover Publications, New York, 1950.

30. D. Mayne and R. Moore, "Minimize Computer 'Crashes'", Electronic Design, 22:168-172, April 26, 1974.

31. G.R. Couranz and D.F. Wann, "Theoretical and Experimental Behavior of Synchronizer Operating in the Metastable Region", submitted to IEEE Transactions on Computers, July 1973.

BIBLIOGRAPHY (for Appendix A)

Wilbur B. Davenport and W.L. Root; Random Signals and Noise, McGraw-Hill, New York, 1958.

Anthanasios Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw-Hill, New York, 1965.

BIBLIOGRAPHY (related work in progress)

Marco Hurtado, "Dynamic Structure and Performance of Asymptotically Bistable Systems", Doctor of Science dissertation, Department of Electrical Engineering, Washington University, St. Louis, Missouri,1974.

Ramaswamy Srinivasan, "Anomalous Behavior of Transistor-Transistor Logic Circuits With Totem-Pole Output", Doctor of Science dissertation, Department of Electrical Engineering, Washington University, St. Louis, Missouri,1974.

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY *(Corporate author)* Computer Systems Laboratory Washington University St. Louis, Missouri	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3 REPORT TITLE

THE SYNCHRONIZER "GLITCH" PROBLEM

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Final Report 4/1/65 through 12/1/73

5 AUTHOR(S) *(First name, middle initial, last name)*

Thomas J. Chaney

6 REPORT DATE February, 1974	7a. TOTAL NO OF PAGES 54	7b. NO OF REFS 31

8a. CONTRACT OR GRANT NO DOD (ARPA) Contract SD-302 b. PROJECT NO ARPA Project Code No. 655 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) Volume IV of Part 1 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* Technical Report No. 47

10 DISTRIBUTION STATEMENT

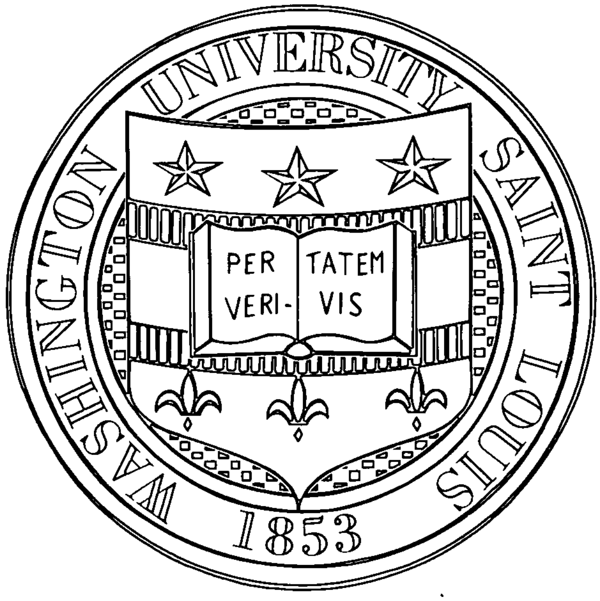
Distribution of this document in unlimited.

11. SUPPLEMENTARY NOTES	12 SPONSORING MILITARY ACTIVITY ARPA - Information Processing Techniques, Washington, D.C.

13 ABSTRACT

There is a fundamental problem in synchronizing communication between any two concurrently operating digital systems that lack a common time reference. This problem involves the inability to build a completely reliable synchronizer or arbiter that will work in a prescribed amount of time. Stimulated by the need for an interlock macromodule design of predictable reliability, the inability to find evidence of previous studies, and indications that this problem has been responsible for significant reduction in the reliability of several commercial machines, we undertook theoretical and experimental studies of this problem. The results to date of these studies are documented in this volume.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Asynchronous Computer Systems						
Arbiter Circuits						
Synchronizers						
Glitch Phenomenon						
Interlock						
Interrupt Systems						
Switching Circuits						
Priority Control Schemes						
Decision Theory						


WASHINGTON UNIVERSITY SAINT LOUIS
PER VERI- TATEM VIS
1853