A SOLUTION TO A SPECIAL CASE OF THE SYNCHRONIZATION PROBLEM

William K. Stewart

SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN
ELECTRICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1983

Signature of Author [illegible]
Department of EE & CS, August 31, 1983

Certified by [illegible]
Thesis Supervisor

Accepted by ______________________
Departmental Chairman

A SOLUTION TO A SPECIAL CASE OF THE SYNCHRONIZATION PROBLEM

William K. Stewart

Submitted to the Department of Electrical Engineering and Computer Science
of the Massachusetts Institute of Technology on September 2, 1983
in partial fulfillment of the requirements for the Degree
of Bachelor of Science in Electrical Engineering

ABSTRACT

Perfectly synchronizing an asynchronous digital signal in bounded time is known to be impossible, since all bistable devices exhibit a region of metastability. In practice, "reliable synchronization" means the achievement of a synchronization failure rate comparable to hardware failure rates. Since metastable state decay times are exponentially distributed, an arbitrarily low synchronization failure rate can be achieved by performing the synchronization with a shift register of sufficient length. Unfortunately, this leads to a trade-off between failure rate and propagation delay. Low synchronization failure rates imply long propagation delays, which can seriously degrade the performance of delay-sensitive systems.

This paper describes a synchronizer that exhibits an arbitrarily low failure rate with a short *average* propagation delay for the special case of synchronizing a signal that is synchronous with some periodic signal to which the synchronizer has access.

Key words: synchronization, arbitration, metastability.

Thesis Supervisor: Stephen A. Ward, Associate Professor of Computer Science

To all the graduate students
of the Real Time Systems Group
who rejected this research topic.

ACKNOWLEDGEMENTS

The successful completion of this thesis would not have been possible without the assistance of many people.

Specifically, I would like to thank Steve Ward, Chris Terman, Rich Zippel, Al Mok, and Dave Goddeau for numerous technical discussions that had a significant impact on the direction taken by my research.

Lisa Chabot and Jon Powell were kind enough to proofread intermediate drafts and to make constructive stylistic comments.

Finally, Chris Terman and Dave Goddeau are almost completely responsible for the fact that this document is more-or-less properly formatted.

CONTENTS

1. The Synchronization Problem

It is impossible to synchronize an asynchronous digital signal in bounded time, with probability zero of failure to synchronize, for the same reason that no upper bound can be placed on the time required for a pencil perfectly balanced on its point to achieve a stable (e.g., horizontal) state. Knife-edge decision-making implies the existence of *metastable states*, states that persist until sufficiently perturbed by statistical phenomena such as Johnson noise.

Many researchers have directly observed metastability in bistable logic devices with both ordinary and sampling oscilloscopes [1,2,5]. Others have adduced statistical proof of the existence of metastability [3,4]. Furthermore, a general proof exists that all bistable devices, electrical, mechanical, hydraulic or otherwise, must exhibit a region of metastability [6]. (Nevertheless, as recently as 1977, articles claiming to present metastability-free synchronization schemes have been published in reputable journals [7].) As a practical matter, it is straightforward to construct a simple finite-state machine with an asynchronous input that fails regularly unless explicit steps are taken to avoid metastability. Such a machine is described in section 3.1.

Current practice recognizes that metastable state decay times are exponentially distributed since metastable state decay is a Poisson process [5,8]. Therefore, arbitrarily low synchronization failure rates can be achieved by allowing synchronizing flip-flops sufficient time to settle. An ubiquitous circuit embodying this principle is the *shift-register synchronizer*, which consists of n cascaded D flip-flops, $n=2$ being the most common case. If the clock to which the asynchronous input is to be synchronized has a period of T, the shift-register synchronizer allows $(n-1)T$ seconds for metastable states to settle. Although input transitions are delayed by a constant $(n-1)T$ seconds, the shift-register synchronizer exhibits a throughput of $\frac{1}{T}$, which is as fast as can be. As a concrete example, a clocked sequential circuit constructed with low-power Schottky TTL components and operating at a clock frequency of 10 mHz would require a single-stage shift-register synchronizer to synchronize an asynchronous input, 100 ns being more than sufficient to achieve a synchronization failure rate comparable to hardware failure rates, presumably a tolerable level.

2. An Interesting Special Case

The canonical example of an asynchronous event in the life of a computer is the depressing of a teletype key by a user. It is neither necessary nor possible to improve upon the performance of the shift-register synchronizer in such a case. Consider instead a pair of finite state machines that are in communication with one another. Each machine has its own clock. These clocks may or may not be of the same nominal frequency, and even if they are, the phase relationship between them is unspecified. A pair of independently clocked processors competing for a shared resource is an example of such a system. A common strategy is to require that each machine treat the current state of the other as a totally asynchronous input, capable of changing at any time. To do so, however, is to ignore the fact that a clocked sequential circuit cannot change state except immediately after ticks of its clock. If each FSM were provided with the clock signal of the other, the metastable-state settling-time penalty need be paid only when absolutely necessary.

2.1. How it works

A scheme for constructing a minimum-average-latency synchronizer is sketched in the following sections.

2.1.1. Definitions

LCLK: the local clock; i.e., a locally-generated signal consisting of periodic, positive-going edges.

FCLK: a foreign clock; i.e., any signal consisting of periodic, positive-going edges.

GO: an aperiodic signal synchronous with FCLK; i.e., all transitions of GO are guaranteed to occur between 0 and τ_f seconds after a positive-going edge of FCLK, where $\tau_f \ll T_f$, the period of FCLK.

2.1.2. Problem

Synchronize GO with LCLK, delaying GO only as necessary to avoid arbitration failure.

2.1.3. Plan

As usual, apply GO to the D input of an edge-triggered D flip-flop clocked by LCLK (hereinafter called the *synchronizer*). Create a *window* signal associated with LCLK; i.e., create a train of positive pulses of duration τ_l and with period T_l, the period of LCLK. The duty cycle and phase of this window signal should be such as to make the positive pulses coincide with the times during which the input of an edge-triggered D flip-flop clocked by LCLK *must not* change if the data set-up and hold time requirements of the flip-flop are to be respected. Similarly, create a window signal associated with FCLK consisting of a train of positive pulses of duration τ_f and with period T_f, the period of FCLK. The duty cycle and phase of this window signal should be such as to make the positive pulses coincide with the times during which the output of an edge-triggered D flip-flop clocked by FCLK *may* change, as determined by the maximum clock-to-output propagation delay of the flip-flop.

The synchronizer is susceptible to metastability just when these window signals overlap, since then and only then is it possible for GO to be in transition when it mustn't be. Because the LCLK and FCLK window signals are periodic, it is possible to predict which local clock pulses will be accompanied by overlapping LCLK and FCLK windows and to protect the synchronizer from metastability by disabling it at the proper time. Specifically, a pulse detector must monitor the logical AND of advance copies of the LCLK and FCLK window signals, closing a latch inserted between GO and the synchronizer as necessary to prevent changes in GO from causing synchronizer metastability.

Since the degree of overlap between the windows can vary continuously from none to total, ANDing advance copies of the two window signals can produce runt pulses. Therefore, the advance copies must be sufficiently in advance to allow the pulse detector to recover from metastable states.

2.2. Performance

Assuming that all possible phase relationships between FCLK and LCLK are equally likely, the probability that any randomly chosen FCLK window will collide with an LCLK window is

$$P = \frac{\tau_f + \tau_l}{T_l} \qquad (2.1)$$

Hence the synchronization process can take at most one cycle, and the average number of cycles required to recognize a change in GO is just

$$N = \frac{\tau_f + \tau_l}{T_l} \tag{2.2}$$

3. TTL Implementation

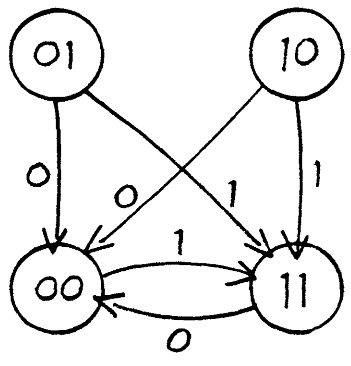
This section describes TTL realizations of two previously mentioned devices:

- the synchronization failure detector of section 1, and
- the minimum-average-latency synchronizer of section 2.

Experimental results produced by the metastability detector are presented, and the performance of the minimum-average-latency synchronizer is computed.

3.1. Synchronization failure detector

Synchronizer metastability will cause random behavior in all finite-state machines for which the value of a single input variable affects more than one state variable. Therefore, a machine susceptible to errors due to synchronization failure need have only two state variables. Such a machine is described by the following state-transition diagram:



Note that except for a start-up transient, this FSM simply shuttles back and forth between the two states marked 00 and 11 depending on the value of the input. If the synchronizing flip-flop enters the metastable state with the result that the next-state values are invalid when the next clock pulse arrives, each state flip-flop must independently and arbitrarily choose a value for its output. If one flip-flop should latch a 1 while the other latches a 0, an illegal transition will have been executed. The state of the machine is monitored by an XOR gate. For each clock period during which the FSM resides in state 01 or state 10, the XOR gate enables a synchronous counter, which thereby keeps a running total of the number of synchronization failures. The schematic of a low-power Schottky TTL implementation of such a machine operating with a clock period T_l of 100 ns appears in Appendix 1.

For the 74LS175 quadruple D flip-flops used, the manufacturer specifies a minimum data-to-clock setup time t_s of 10 ns and a maximum clock-to-output propagation time t_p of 22 ns. An active digital delay line with a propagation delay t_d of 60 ns simulates the combinational logic portion of the FSM. Since

$$t_p + t_d + t_s < T_l \tag{3.3}$$

this machine is "properly" designed. Therefore, all illegal state transitions must be the result of synchronizer metastability. Furthermore, if a two-stage shift-register synchronizer is created by prepending to the synchronizing flip-flop another 74LS175 D flip-flop clocked by LCLK, *illegal state transitions cease to occur.*

3.2. Minimum-average-latency synchronizer

Given local and foreign clocks characterized by

$$T_l = 100\,\text{ns} \tag{3.4}$$

$$T_f = 125\,\text{ns} \tag{3.5}$$

the task is to implement the minimum-average-latency synchronizer described in section 2.2.

3.2.1. Circuit topology

The schematic of a Schottky TTL implementation of the synchronizer appears in Appendix 2; Appendix 3 contains a timing diagram.

A single-shot constructed from a delay line, an inverter, and a NOR gate produces EN. Two more similarly-constructed single-shots generate the FCLK and LCLK window signals FCLKWIN and LCLKWIN. A delay line adjusts the phase of FCLKWIN with respect to LCLKWIN so that their relationship models the relationship that will exist between FCLK and LCLK slightly less than two LCLK periods to come. FCLKWIN and LCLKWIN are ANDed to produce NO, which triggers a two-stage pulse detector by asynchronously clearing its first stage. This stage is synchronously set at the beginning of every LCLK cycle, and remains set unless cleared by NO. Since NO is capable of being a runt pulse, the first stage is susceptible to metastability. However, any activity on NO must occur while LCKLWIN is logically high, leaving the first stage the better part of an LCLK period to settle before being sampled by the second stage. The output of the pulse detector, OK, is ORed·with EN to produce OPEN, which enables the D latch that converts GO to GOSAFE. OK keeps the latch open during LCLK cycles in which synchronizer metastability is impossible. Only when GO can change near the next active LCLK edge does OK allow EN to close the latch part-way through the cycle. The trailing edge of EN occurs sufficiently in advance of the next active edge of LCLK that GO cannot occur *both* close enough to an active LCLK edge to trigger the pulse detector and close enough to the trailing edge of EN to cause the D latch to become metastable. Consequently, GOSAFE is guaranteed to be valid in the vicinity of active edges of LCLK. Furthermore, GOSAFE never lags GO by more than one LCLK period.

3.2.2. Delay line values

The effective set-up time t_s' of the 74LS175 D flip-flop is equal to its intrinsic set-up time plus the propagation delay t_d of the 74S373 D latch that precedes its D input:

$$t_s' = t_s + t_d = 10\,\text{ns} + 7\,\text{ns} = 17\,\text{ns} \tag{3.6}$$

Since the required minimum data hold-time of the 74LS175 is 0 ns,

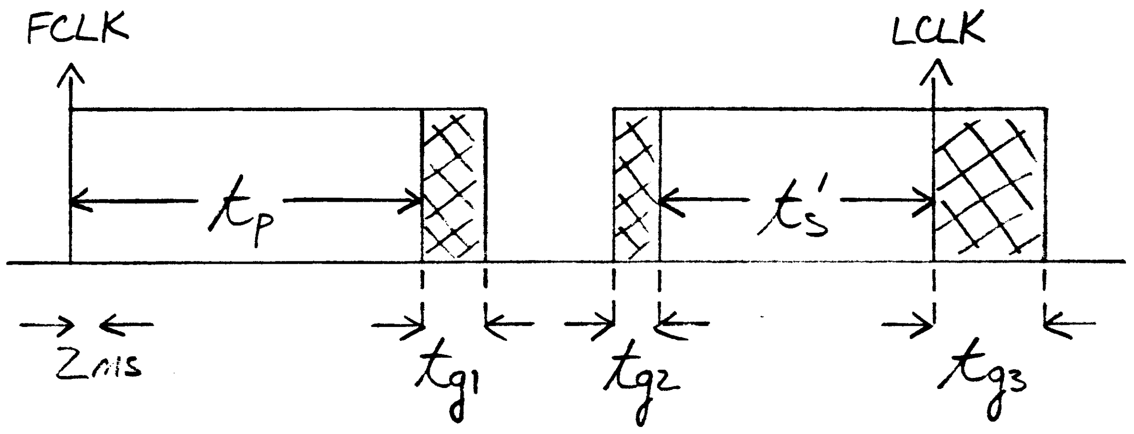
$$\tau_l = t_s' + t_h = 17\,\text{ns} \tag{3.7}$$

Furthermore,

$$\tau_f = t_p = 22\,\text{ns} \tag{3.8}$$

since the maximum clock-to-output propagation delay of a 74LS175 D flip-flop is 22 ns.

Each of the delay lines used to generate τ_f and τ_l is paralleled by an inverter possessing a maximum propagation delay $\varepsilon = 3\,\text{ns}$. In addition, a pulse at least 7 ns wide is necessary to reliably clear the 74S74 flip-flop. Therefore, a 7 ns guard band must be provided between the leading edge of FCLKWIN and the trailing edge of LCLKWIN, and vice versa. Keeping in mind that the windows will be created with delay lines tapped every 5 ns, the guard bands are apportioned as shown in the figure below:



Taking everything into consideration, the delay lines that determine the foreign and local clock window widths should have the following minimum values:

$$\tau_f' = \tau_f + t_{g_1} + \varepsilon = 22\,\text{ns} + 4\,\text{ns} + 3\,\text{ns} = 29\,\text{ns} \tag{3.9}$$

$$\tau_l' = \tau_l + \tau_{g_2} + \tau_{g_3} + \varepsilon = 17\,\text{ns} + 3\,\text{ns} + 7\,\text{ns} + 3\,\text{ns} = 30\,\text{ns} \tag{3.10}$$

Given the above, it is theoretically possible for GO to change as much as

$$t_s' + t_g + \varepsilon + t_p = 17\,\text{ns} + 7\,\text{ns} + 3\,\text{ns} + 22\,\text{ns} = 49\,\text{ns} \tag{3.11}$$

before an active LCLK edge and for EN to close the D latch. In order to avoid metastability in the D latch, it is therefore necessary for the trailing edge of EN to occur no more than 45 ns after each active LCLK edge. Choosing 40 ns for the value of the delay line that determines the duty cycle of EN is appropriate.

These delay line values constrain LCLK to be logically high for between 40 ns and 60 ns, and they constrain FCLK to be logically high for between 30 ns and 95 ns.

If $f(t)$ is a periodic function with period T, then

$$f(t) = f(t - nT) \qquad n = 0, \pm 1, \pm 2, \ldots \tag{3.12}$$

If $f(t)$ is to be *advanced* by A seconds, then

$$f(t)' = f(t + A) = f(t + A - nT) = f(T - D) \tag{3.13}$$

where

$$D = nT - A > 0 \qquad n > \frac{A}{T} \tag{3.14}$$

is the amount by which $f(t)$ must be *delayed*, and conversely. The use of a two-stage pulse detector to monitor NO requires local and foreign clock window overlaps to be predicted between one and two local clock periods in advance. Therefore, LCLKWIN is effectively advanced by

$$2T_l - t_s' - t_{g_2} = 200\,\text{ns} - 17\,\text{ns} - 3\,\text{ns} = 180\,\text{ns} \tag{3.15}$$

and the smallest non-negative amount by which FCLKWIN can be delayed is 250 ns − 180 ns = 70 ns.

3.2.3. Performance

The average latency exhibited by the synchronizer described in the previous section can be no greater than

$$N = \frac{\tau_f + \tau_l}{T_l} = \frac{30\,\text{ns} + 30\,\text{ns}}{100\,\text{ns}} = 0.60 \tag{3.16}$$

local clock cycles, or 60 ns.

The theoretical minimum latency is

$$N = \frac{t_p + t_s + t_h}{T_l} = \frac{22\,\text{ns} + 10\,\text{ns} + 0\,\text{ns}}{100\,\text{ns}} = 0.32 \tag{3.17}$$

local clock cycles, or 32 ns.

The latency of a two-stage shift-register synchronizer is 1.00 local clock cycles, or 100 ns.

3.3. Experimental results

Experiments were conducted with the synchronization failure detector whose schematic appears in Appendix 1, GO being derived from FCLK with a divide-by-two circuit. The number of synchronization failures that occurred in 60 seconds of operation with each of five randomly-chosen Fairchild 74LS175 D flip-flops was:

1	1487
2	9579
3	1430
4	825
5	4601

These data are indicative of the exponential dependence of flip-flop settling time on poorly specified digital amplifier characteristics such as small-signal gain-bandwidth product.

No synchronization failures were recorded in 36 hours of operation with either a single-stage shift-register synchronizer or the minimum-average-latency synchronizer whose schematic appears in Appendix 2.

4. Conclusions

With a significant increase in complexity and with considerable attention to detail, it is possible to better the performance of a shift-register synchronizer, provided that the input to be synchronized is synchronous with some periodic signal to which the synchronizer has access.

4.1. Applications

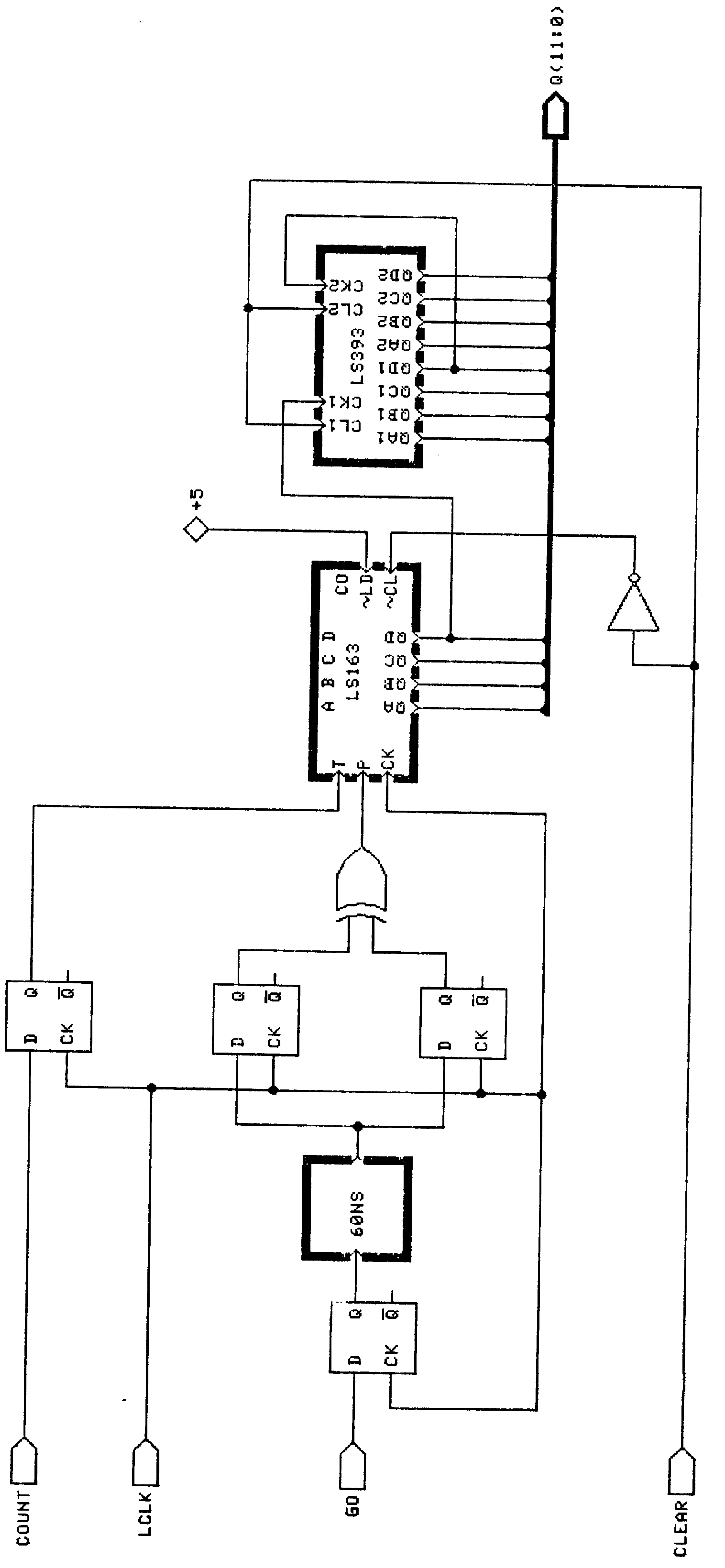
The synchronization strategy discussed in this paper applies whenever two or more independently-clocked synchronous sequential circuits must communicate and latency is an issue. For example, consider that the limiting performance factor in many computer systems is memory access time. If a number of processors must contend for the use of an asynchronous data bus, metastable-state settling time puts an absolute floor underneath memory access time. Even if the bus is synchronous, it may be inconvenient or impossible for all potential bus masters to operate on clocks that are phase-locked to a global timing signal. Particularly if the peak-to-average utilization ratio of the bus is high, overall performance may be degraded by enforcing global synchrony. True synchronization may be impractical because the system is too widely distributed.

Concrete examples of multiple-processor systems that exhibit one or more of these characteristics include the Nu Machine, a personal work-station developed by the M.I.T. Laboratory for Computer Science, and Concert, a general-purpose multiprocessing system under construction by the same group.

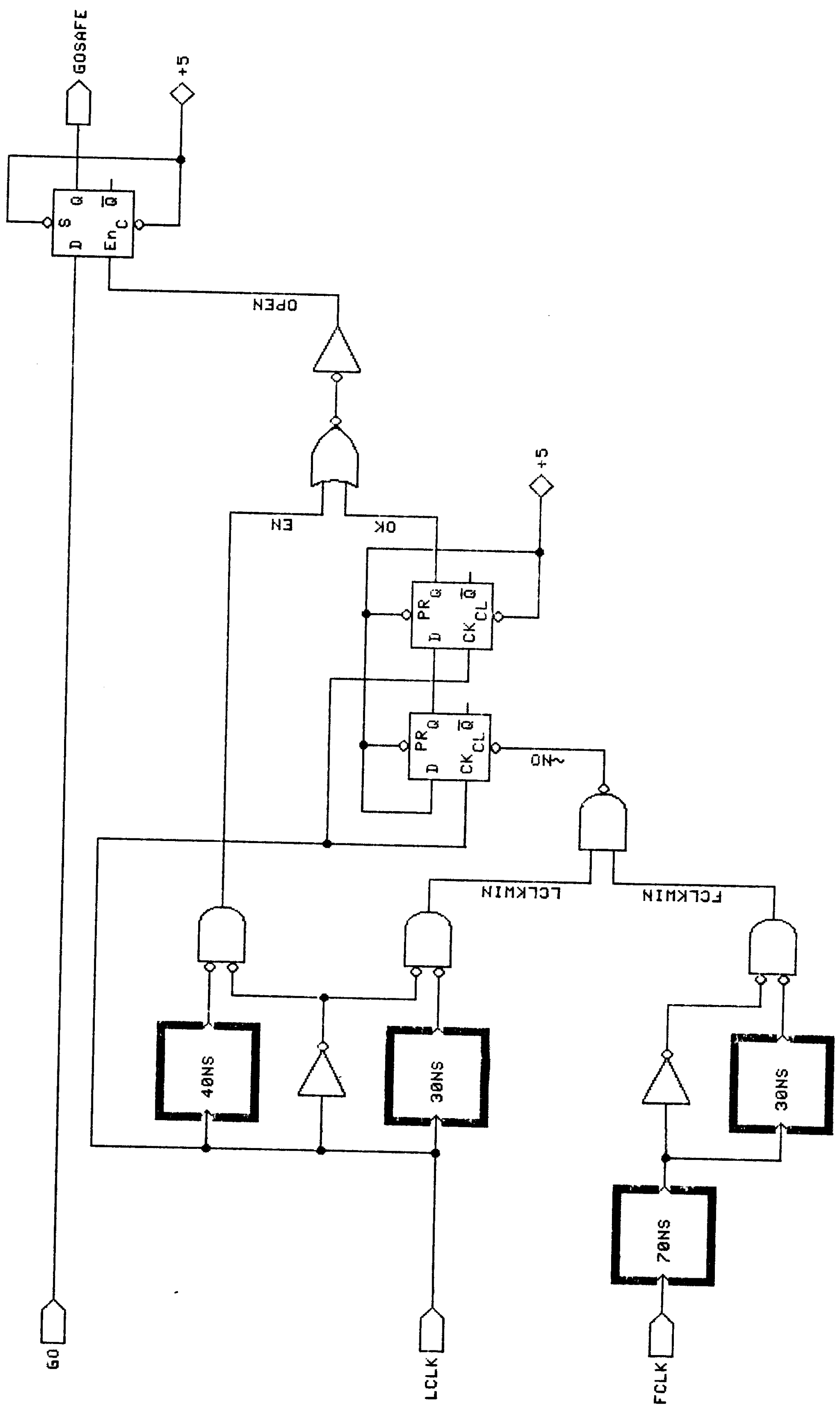
4.2. Further research

A difficulty with the current design is that it requires *a priori* knowledge of the local and foreign clock frequencies. Ideally, the synchronizer would automatically adjust itself to any clock frequencies within a reasonable range. Current speculation centers around the use of a voltage-controlled oscillator contained within a phase-locked loop as an alternative to a fixed delay line to produce the necessary phase shift in the foreign clock window signal.

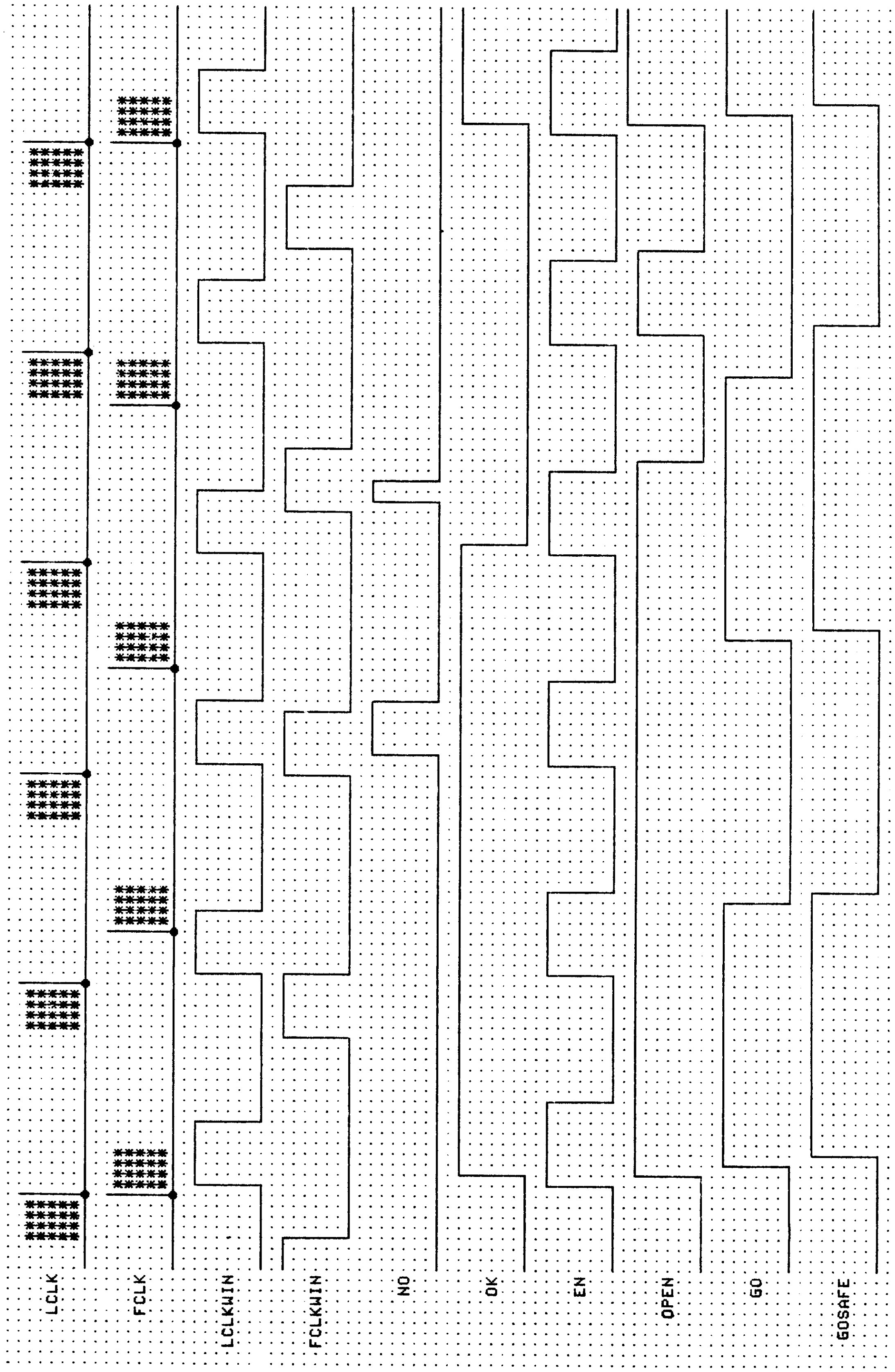
Appendix 1: Synchronization failure detector schematic



Appendix 2: Minimum-average-latency synchronizer schematic



Appendix 3: Minimum-average-latency synchronizer timing diagram



REFERENCES

[1] Chaney, T. J., and C. E. Molnar, "Anomalous Behavior of Synchronizer and Arbiter Circuits," IEEE Transactions on Computers, vol. C-22, pp. 421-422, April 1973.

[2] Chaney, T. J., "Comments on 'A Note on Synchronizer or Interlock Maloperation,'" IEEE Transactions on Computers, vol. C-28, pp. 802-804, October 1979.

[3] Pechoucek, M., "Anomalous Response Times of Input Synchronizers," IEEE Transactions on Computers, vol. C-25, pp. 133-139, February 1976.

[4] Fleischhammer, W., and O. Dortok, "The Anomalous Behavior of Flip-Flops in Synchronizer Circuits," IEEE Transactions on Computers, vol. C-28, pp. 273-276, March 1979.

[5] Couranz, G. R., and D. F. Wann, "Theoretical and Experimental Behavior of Synchronizers Operating in the Metastable Region," IEEE Transactions on Computers, vol. C-24, pp. 604-616, June 1975.

[6] Hurtado, M., "Structure and Performance of Asymptotically Bistable Dynamic Systems," D.Sc. Dissertation, Washington Univ., St. Louis, Mo., 1974.

[7] Wormald, E. G., "A Note on Synchronizer or Interlock Maloperation," IEEE Transactions on Computers, vol. C-26, pp. 317-318, March 1977.

[8] Kinniment, D. J., and D. G. B. Edwards, "Circuit Technology in a Large Computer System," Radio and Electronic Engineer, vol. 43, pp. 435-441, July 1973.