

Washington University in St. Louis

Washington University Open Scholarship

All Computer Science and Engineering
Research

Computer Science and Engineering

Report Number: WUCS-79-1

1979-04-01

Synchronization Strategies

Mishell J. Stucki and Jerome R. Cox Jr

Computing systems are now frequently composed of independently clocked subsystems that cooperate to perform the function desired for the whole. This type of architecture has many advantages and promises to be the standard for the foreseeable future. With the trend towards more and more gates per chip, the number of chips per subsystem gets smaller and smaller, and we can expect to soon see one or more subsystems per chip. This transition will require contributions from disciplines previously outside the field of chip design, and every issue will have to be carefully worked out beforehand because debugging chips of... **Read complete abstract on page 2.**

Follow this and additional works at: https://openscholarship.wustl.edu/cse_research



Part of the [Computer Engineering Commons](#), and the [Computer Sciences Commons](#)

Recommended Citation

Stucki, Mishell J. and Cox, Jerome R. Jr, "Synchronization Strategies" Report Number: WUCS-79-1 (1979). *All Computer Science and Engineering Research*.
https://openscholarship.wustl.edu/cse_research/868

Department of Computer Science & Engineering - Washington University in St. Louis
Campus Box 1045 - St. Louis, MO - 63130 - ph: (314) 935-6160.

Synchronization Strategies

Mishell J. Stucki and Jerome R. Cox Jr

Complete Abstract:

Computing systems are now frequently composed of independently clocked subsystems that cooperate to perform the function desired for the whole. This type of architecture has many advantages and promises to be the standard for the foreseeable future. With the trend towards more and more gates per chip, the number of chips per subsystem gets smaller and smaller, and we can expect to soon see one or more subsystems per chip. This transition will require contributions from disciplines previously outside the field of chip design, and every issue will have to be carefully worked out beforehand because debugging chips of this complexity is a difficult and costly task. This paper addresses one of those issues - the design of reliable synchronization logic for interfacing independently clocked subsystems. The design of this logic is not a normal exercise in clocked logic design because the operating environment is such that the response times of some flipflops will be unbounded, and an improper appreciation of this phenomenon can result in designs plagued by intermittent synchronization failures. The absence of a bound had been documented in the literature, but only from an experimental and analytic standpoint, and no generally applicable methodology for dealing with it has been suggested. As a result, it is not common knowledge among logic designers, and future systems are liable to suffer from it. The objective of this paper is to assist the logic designer by reviewing the basic phenomenon, characterizing it quantitatively, and presenting techniques for coping with it.

SYNCHRONIZATION STRATEGIES

M. J. Stucki and J. R. Cox, Jr.

WUCS-79-1

Department of Computer Science

Washington University

St. Louis, Missouri 63130

April 1979

This work has been supported in part by the Division of Research Resources
of the National Institutes of Health under Grant RR 00396

INTRODUCTION

Computing systems are now frequently composed of independently clocked subsystems that cooperate to perform the function desired for the whole. This type of architecture has many advantages and promises to be the standard for the foreseeable future. With the trend towards more and more gates per chip, the number of chips per subsystem gets smaller and smaller, and we can expect to soon see one or more subsystems per chip. This transition will require contributions from disciplines previously outside the field of chip design, and every issue will have to be carefully worked out beforehand because debugging chips of this complexity is a difficult and costly task.

This paper addresses one of those issues - the design of reliable synchronization logic for interfacing independently clocked subsystems. The design of this logic is not a normal exercise in clocked logic design because the operating environment is such that the response times of some flipflops will be unbounded, and an improper appreciation of this phenomenon can result in designs plagued by intermittent synchronization failures. The absence of a bound has been documented in the literature, but only from an experimental and analytic standpoint, and no generally applicable methodology for dealing with it has been suggested. As a result, it is not common knowledge among logic designers, and future systems are liable to suffer from it. The objective of this paper is to assist the logic designer by reviewing the basic phenomenon, characterizing it quantitatively, and presenting techniques for coping with it.

THE SYNCHRONIZATION PROBLEM

The specification sheet for a sequential device gives operating constraints such as setup times, hold times and maximum clock rates. These are constraints that must be met in order to assure a consistent interpretation of the input signals. Consider, for example, the common type D flipflop. If its data input value is logically defined during the specified setup and hold periods, the value seen by the flipflop when clocked will be unambiguous, and two or more flipflops presented with the same conditions will see the same value. However, if the constraints are not met, then interpretations may differ. For example, one flipflop could interpret a logically undefined input value as a 1 while another flipflop could interpret the same signal as a 0. Similarly, if the input signal is changing value, one flipflop could capture the before value while another flipflop could capture the after value. Consistent interpretations are guaranteed only for cases that meet the specified input constraints.

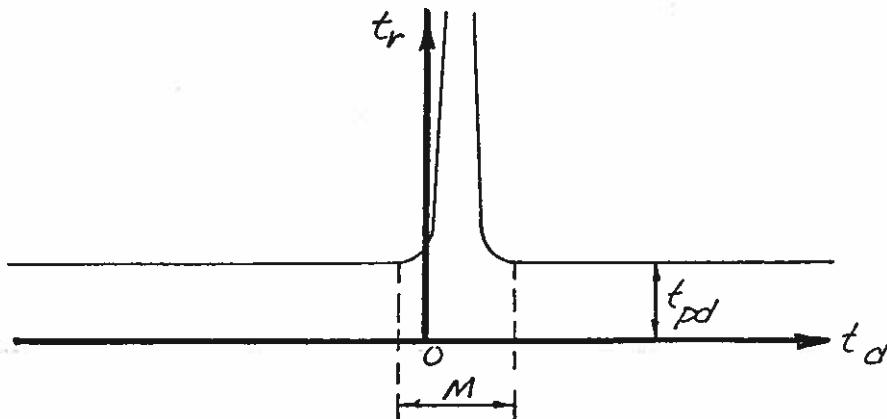
Preserving consistency is a primary concern in the design of systems, and the proper functioning of a system requires that all devices that depend on the value of a given signal at a given time see the same value, i.e., they all see a 0 or they all see a 1. The logic designer satisfies this requirement by trying to assure that the input constraints are met for each device. In

cases where he cannot adequately control an input signal, for example a signal originating external to the system, he solves the consistency problem by allowing only one observer and therefore only one interpretation. The traditional method is to gate the signal into a flipflop and then use the flipflop's output value as the signal to be processed by the system. The flipflop is said to synchronize the input signal because it gives the designer a copy which he can control.

In point of fact, the copy is not totally under the designer's control. Although it is common knowledge that violating the input constraints of a flipflop can cause unusual behavior and although it is common practice to allow for a longer than normal propagation delay in these applications, experimental studies [1, 2, 3, 4, 5, 6] and theoretical studies [7, 1, 8] indicate that the response times are actually unbounded. It is thus possible for the output signal of the flipflop to violate the input constraints of other devices and, by so doing, cause inconsistencies. This accounts for the occasional synchronization failures experienced by existing systems.

RESPONSE TIME

When the input constraints of a flipflop are violated, the input event can leave the circuit in a non-stable state and stabilization then occurs as a result of regenerative feedback. The response time depends on the time needed to stabilize and that depends on the state from which stabilization begins. This relationship is developed analytically in Appendix A for an NMOS flipflop constructed from two cross-coupled inverters. If we add some input gating to that circuit and create a latch with data input D and clock input C, then the sketch below illustrates the dependence of response time on input condition. The ordinate t_r is the circuit's response time and the abscissa t_d is the time of occurrence of a data change as measured from the nearest clock event. The normal response time of the circuit is t_{pd} , and interval M is the range of values of t_d for which the response time is greater than normal.



The value of t_r becomes arbitrarily large near the middle of M. In particular, for any proposed bound t_b , we see that there exists an interval within M for which $t_r > t_b$. The probability of exceeding this bound is thus equal to the probability of t_d occurring in that interval. This observation is quantified in the following equations which are based on the stabilization time properties of the model in Appendix A and on a uniform distribution of t_d .

$$P(t_d \in M) = \frac{I_M}{I_C} \quad (1)$$

$$P(t_r > t_b \mid t_d \in M) = \frac{1}{k + (1-k)e^{(t_b - t_{pd})/\tau}} \quad (2)$$

$$P(t_r > t_b) = \frac{I_M}{I_C} \cdot \frac{1}{k + (1-k)e^{(t_b - t_{pd})/\tau}} \quad (3)$$

$$\cong \frac{I_M}{I_C} \cdot \frac{e^{-(t_b - t_{pd})/\tau}}{(1-k)} \quad t_b - t_{pd} \geq 5\tau \quad (4)$$

$$= \frac{T_0}{I_C} \cdot e^{-t_b/\tau} \quad t_b - t_{pd} \geq 5\tau \quad (5)$$

Parameter I_M is the duration of interval M, parameter I_C is the clock period and is assumed to be constant, and t_b is a bound greater than or equal to t_{pd} . Parameters k and τ are circuit parameters as defined in Appendix A, with k being a positive fraction less than 1 and τ being a time constant with values on the order of a few nanoseconds.

Equation 5 is a convenient form because there are only two circuit-dependent parameters, T_0 and τ , and an experimental method for estimating them is given in [9]. Equation 5 is also in agreement with results obtained experimentally and analytically for other technologies, including TTL, ECL, and a tunnel-diode memory element. There is evidence suggesting that it is not a good estimator for values of t_b close to t_{pd} . Equation 3 seems reasonable in that region but this has yet to be verified experimentally.

IMPLICATIONS

Given that Equation 5 is reasonably accurate, two observations can be made. The first is that there is no finite value of t_b for which $P(t_r > t_b) = 0$. This supports the contention that response time is unbounded. The second observation is more complex and requires the following preliminaries.

An error is the occurrence of a response time greater than the response time available in the application. A failure is an inconsistency caused by an error. Failures occur less often than errors because consistent interpretations are possible even if input constraints are violated. The expected number of errors is denoted $E_e(t_a)$ and, as derived in Appendix B, is given by

$$E_e(t_a) = P(t_r > t_a) \cdot \lambda \cdot t \tag{6}$$

where t_a is the available response time, where λ is the average rate of change of the signal at the data input of the flipflop, and where t is the time over which the errors are counted. $E_e(t_a)$ is the least upper bound for the expected number of failures and is thus a good measure of reliability. The expected number of errors decreases rapidly for larger values of t_a . In particular,

$$E_e(t_a + x) \approx E_e(t_a) \cdot 10^{-x/2.3\tau} \tag{7}$$

The values of τ that have been measured for TTL, ECL, and NMOS circuits are close to 2 nanoseconds. This means that the expected number of errors is reduced by a factor of about 150 each time t_a is increased by 10 nanoseconds. Extremely low error rates can thus be obtained for t_a in the tens of nanoseconds. For example, the box below gives the parameters for a reasonably busy TTL application. The expected number of errors is 1.24 every 10 years. For all practical purposes, the design does not fail.

Parameters

Application	Circuit (TTL)	Design
$I_C = 100$ nsec	$\tau = 1.8$ nsec	$t_a = 60$ nsec
$\lambda = 10^5$ changes/sec	$T_0 = 10^{3.07}$ nsec	

This allows us to make our second observation. Even though there is no absolute bound, there are bounds that are adequate from an engineering standpoint. These engineering bounds differ from absolute bounds in that they depend on application parameters as well as on circuit parameters. They must therefore be determined on a case by case basis.

BOUND BASED DESIGNS

The simplest and most economical technique for coping with the absence of an absolute bound is to use an engineering bound. The reliability of this approach depends on the statistical properties of the input signal, on the sample rate, on the circuit parameters, and on the available response time. While the first of these is not under the designer's control, the others are, and this section discusses some of the ways in which he can exercise this control.

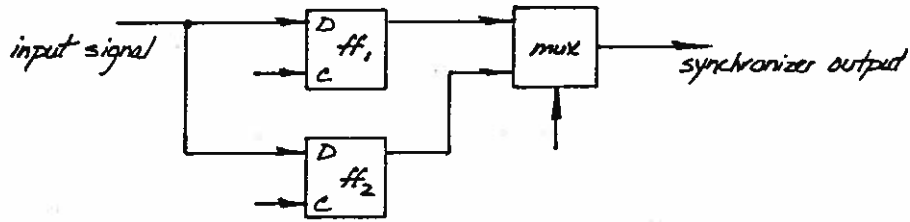
The designer's task in using this approach is to provide enough response time, and a crucial first step is to determine how much is enough. If the statistical properties of the input signal are appropriate, the required time can be estimated by picking an acceptable error rate and solving Equation 6 for t_a . This calculation requires values for λ , T_0 , and τ , where λ is estimated from an analysis of the input environment and where T_0 and τ are estimated for the flipflop circuit that the designer would like to use. Estimating the circuit parameters is a problem because manufacturers do not provide this information. However, representative values for different technologies can be found in the following references: TTL [3, 5], ECL [3, 5], NMOS [9], tunnel diode [1, 5]. If the statistical properties of the input signal do not allow the use of Equation 6, the designer must analyze the properties and develop an equivalent equation. In either case, the derived value is the minimum response time that should be made available. We denote that time by t_b because it is the bound that will be assumed in the design process. The response time that is actually available is denoted t_a , and it is the designers task to guarantee that $t_a \geq t_b$.

Once a value has been determined for t_b , the designer can make a quick design pass and determine if $t_a \geq t_b$. If it is, fine. If not, more time can be made available by using more than one flipflop in the synchronizer. For example, consider an application with the parameters shown in the box below.

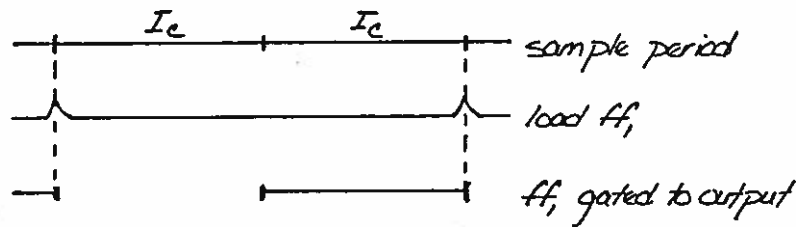
Parameters

Application	Circuit (NMOS)	Design
$I_C = 40$ nsec	$\tau = 1.6$ nsec	$t_a = 30$ nsec
$\lambda = 10^5$ changes/sec	$T_0 = 20$ nsec	

Assuming that Equation 6 applies, a t_a of 30 nanoseconds means that about 31 errors per day can be expected. Since this is too high an error rate, we increase the available response time by using the design shown below.



In this design, the flipflops take turns receiving samples so that, even though the input signal is being sampled every 40 nsec, the clock period for a given flipflop is 80 nsec. The multiplexor gates the data from the flipflops onto a single output path in such a way that the data from a given flipflop is on the path during the sample period preceding the reloading of the flipflop. This is illustrated in the timing diagram below.



The available response time is now the original 30 nsec, plus 40 nsec due to halving the loading rate, minus 10 nsec because of propagation delay through the multiplexor. This is more than adequate since $t_a = 60$ nsec means about 7 errors every 12 thousand years. However, if it were not, more flipflops would be used. The table below shows how the available time increases with the number of flipflops.

<u>Number of Flipflops</u>	<u>Available Response Time (nsec)</u>
1	30 (given)
2	$30 + 1 \cdot 40 - 10 = 60$
3	$30 + 2 \cdot 40 - 10 = 100$
4	$30 + 3 \cdot 40 - 10 = 140$

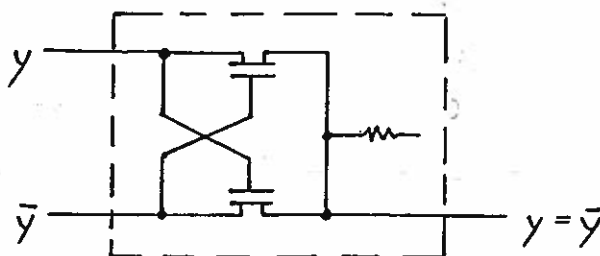
The designer can thus arrange any amount of response time that is called for. There are situations, however, in which t_b must be kept small. The reason is that t_b directly affects the time that it takes for the system

to detect an input change, and performance considerations can require that the detection time be as small as possible. There is only one way that the value of t_b can be reduced without sacrificing reliability. This is to use a flipflop circuit with a smaller value of T_0 and τ . Of the two parameters, τ has the most significant effect, and reducing τ by a factor will reduce the value of t_b by about the same factor. The analysis in [9] indicates that both parameters scale linearly with feature size in NMOS circuits. The VLSI logic designer can thus control the parameters to the extent that manufacturing constraints will allow. In systems constructed of discrete components, extremely small values of T_0 and τ can be obtained by using flipflops constructed of tunnel diodes.

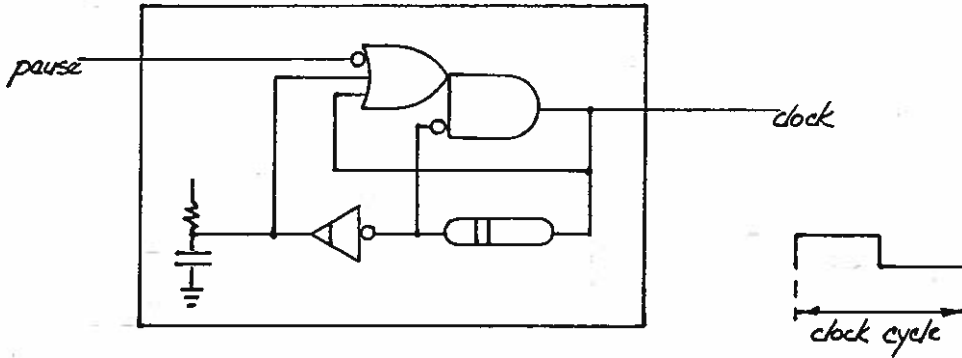
AN ADAPTIVE DESIGN STRATEGY

The design strategy presented in this section differs from that of the preceding section in that reliability is not based on engineering bounds. This scheme monitors the flipflop and, in the case of response times that are long enough to cause errors, it suspends system operation until the response is completed. There are thus no errors. The basic components of this scheme are a flipflop circuit whose response state (stable or unstable) can be monitored, a circuit to do the monitoring, and a clock circuit whose operation can be suspended between clock cycles. The latter is necessary because the monitor suspends system operation by suspending the operation of the clock circuit. This freezes the state of the entire system and guarantees that the flipflop data will not be used until it is stable and logically defined.

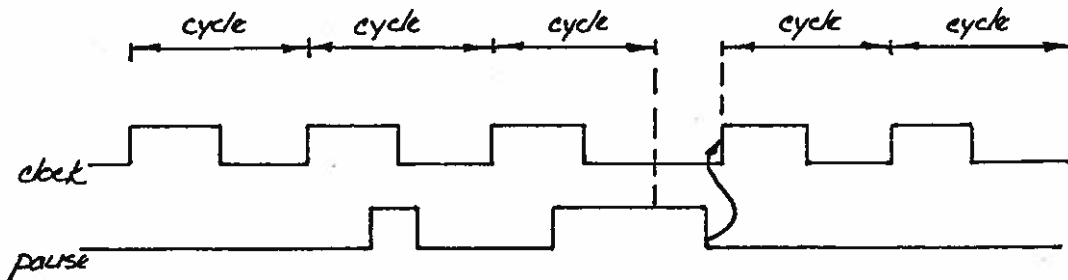
The flipflop circuit used in this design must have an output behavior during a response episode that is uniquely different from its behavior between episodes. The flipflop in Appendix A is such a circuit. During a response episode, the circuit is unstable and its complementary outputs are anti-complementary, that is, they have similar values. The signals can be fairly static, hovering near the middle of the HIGH and LOW logic bands, or they can oscillate, but whatever their behavior happens to be, they will have nearly equal values for the entire episode. This means that the existence of the unstable state can be detected with a difference comparator whose output is a logic 1 if and only if its input signals differ by less than some appropriate amount. The exclusive NOR circuit shown below is just such a detector. Its output is HIGH as long as the input signals differ by less than the threshold voltage of the input gates. The circuit designer sets this threshold at a value that will distinguish the stable and unstable states.



The clock circuit used in this design strategy must be capable of having its operation suspended between clock cycles. The logic diagram for such a circuit is shown below, along with the output waveform for a clock cycle. The waveform can be used as the master clock signal in systems using edge-triggered logic. For systems using pulse-triggered logic, pulses can be generated by adding appropriate output logic.



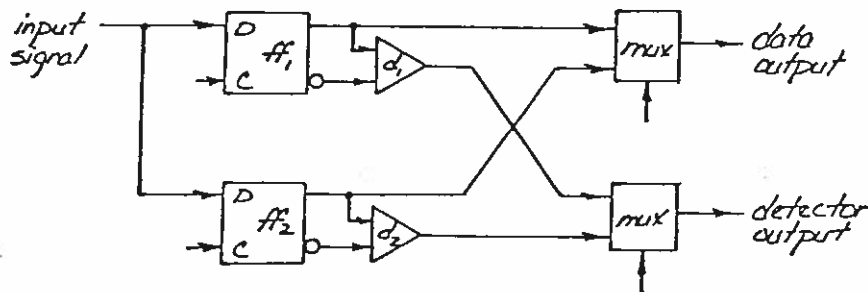
This circuit honors a pause request that is present at the end of a cycle by postponing the next cycle for as long as the request is present. When the request disappears, the next cycle begins immediately. This is illustrated in the diagram below. Note that postponement does not occur if a pause request disappears before the end of a cycle. This is convenient because it means that the detector can request a pause as soon as a response episode begins. If a pause is not actually needed, the request will drop before the end of the cycle and there will be no effect on clock operation.



Since flipflop response time has no absolute bound, it is possible for the operation of the clock to be suspended for an unacceptably long time. To protect against such an occurrence, the clock circuit contains a time-out delay which will automatically restart it if a pause becomes excessive. The length of the time-out delay is controlled by the RC network and is set appropriate to the application.

Proper operation of the clock circuit requires that a pause request not be raised too close to the beginning or end of a clock cycle. This is not a problem in practice because the time at which a pause request is raised is completely predictable. There is no timing constraint on the dropping of a request.

The frequency with which pauses occur and the duration of the pauses are of obvious concern since they affect the performance of the system as a whole. These measures are dependent on the statistical properties of the input signal, on the sample rate, on the circuit parameters, and on the available response time. As in the case of the bound-based strategy, all but the first are under the designers control. Furthermore, since the conditions that lead to an error in the bound-based strategy are the same conditions that lead to a pause in the adaptive strategy, the techniques for controlling error rate can also be used to control pause rate. For example, the multi-flipflop scheme can be used as shown below.



Each flipflop has its own detector d_i and the detector outputs are multiplexed in exactly the same manner as the flipflop outputs. Since the detector for a given flipflop will not be gated through the multiplexor until the sample period preceding the re-loading of the flipflop, a pause will not occur unless instability lasts for at least 2 sample periods. The extension of this scheme to N flipflops is obvious, and a response episode would have to last N sample periods in order to cause the clock to pause. The designer can also affect the pause rate by his choice of flipflop. Circuits with smaller values of τ and T_0 will have shorter response times and will therefore cause fewer pauses. The shorter response times also mean shorter pauses.

Thus, the designer can arrange any amount of response time that is called for. In the case of a uniform input distribution, this time can be estimated by picking an acceptable pause rate and solving Equation 6 for t_a . (The pause rate replaces the error rate in this calculation.) The distribution of pause durations for this case is essentially independent of the available response time. Given that the available time is at least $t_{pd} + 5\tau$, the average duration will be τ , and 98% of the durations will be less than 5τ . Because the pauses tend to be so short, many applications will be able to tolerate a modest pause rate, and as a result, the available response time

can be less than would be required in a bound-based design. This is an advantage in applications where the detection delay must be small.

An important aspect of the adaptive scheme is its ability to resolve the situation where two interacting systems are operating at about the same frequency and phase, and as a consequence, the signals sent between them violate the flipflop constraints with great regularity. In a bound-based design, the error rate would be inordinately high, but in an adaptive design, pauses would perturb the relative operating phase and would tend to disrupt the unfavorable situation, reducing the likelihood of an inordinately high pause rate.

DISCUSSION

Future VLSI designs may use multiple independent clocks that give rise to opportunities for synchronizer failures and ensuing system failures. These failures will be characterized by transient and elusive symptoms that compound the problems of circuit design and verification. Thus careful specification and design must precede the use of a strategy based on bounding the number of synchronization errors. Knowledge of both circuit dependent and application dependent parameters is required for the success of this strategy. Unfortunately, published circuit data are fragmentary, test procedures are unstandardized, manufacturer's specifications are nonexistent, and the application dependent parameters may be inaccessible.

One approach to this difficult situation is to design synchronizers with such a large margin of safety that sizable errors in estimating application and circuit parameters can be tolerated. This is practical since the probability of synchronizer failure is reduced by about two orders of magnitude for each 10 nsec added to the allowed flipflop response time.

Thus, a pragmatic solution exists to the synchronizing problem whenever estimates of circuit and application parameters are available and appropriate. There are, however, important circumstances in which a stochastic model of the input to the synchronizer may be inappropriate. Consider, for example, the composition of large systems through the interconnection of modular subsystems chosen from a set of basic module types. This powerful and attractive approach to VLSI design leads to a class of systems with great diversity in the detailed structure of intermodule interaction. Increased care is required in module design to assure that all legitimate compositions of the basic module types will operate correctly. Synchronization is a particularly vexing problem in this context if each module is independently clocked. No a priori estimate of application parameters is available to the module designer, and even if it were, intermodule interactions may not be well modeled by a stochastic process.

Multiple independently clocked subsystems within a single VLSI chip present a particularly interesting and relevant example of this modeling difficulty. If several local clocks are fabricated on the same chip, they

are likely to operate with very nearly the same period unless special precautions are taken. Although most of the time for most systems, operation would be trouble-free, a particular set of processes operating in subsystems driven by a particular set of clocks could lead to an essentially deterministic pattern of transitions that would violate the flipflops input constraints. Because of uniformity in fabrication, in temperature coefficient, and in sensitivity to external fields, response times of synchronizers in such a VLSI system might be consistently long, and the opportunities for system failure might be increased even beyond that experienced in current experiments designed to measure synchronizer failures.

Under these circumstances, an adaptive synchronization strategy appears to be preferable to a probabilistic one. Synchronization is then error free even though the total time to complete a computation may be increased slightly.

CONCLUSION

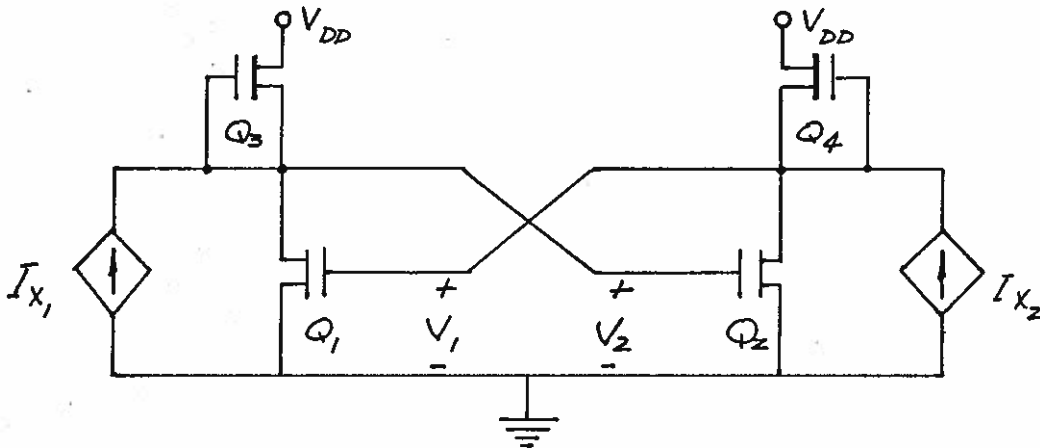
The synchronization problem has been described and synchronizer strategies presented that exemplify the trade-offs possible between failure probability, sample rate, detection time and computation time. The combination of a flipflop and an instability detector, called an indicating flipflop by Kinnement [5], has been known for more than a decade [7, 10]. Until recently the use of an indicating flipflop has been limited to asynchronous systems, but its incorporation in an adaptive synchronization strategy [11] makes possible error-free synchronization at the expense of variability in the clock period. This strategy and the more familiar probabilistic strategy provide the designer with alternatives for the management of the synchronizer problem. Thus, designers who continue to ignore the problem cannot use the excuse that the problem is fundamentally insoluble. Only design and implementation costs stand in the way of reliable synchronizers. With achievement of the anticipated economies of VLSI, we believe these costs are much less than the verification and maintenance costs associated with poorly designed synchronizers.

ACKNOWLEDGEMENTS

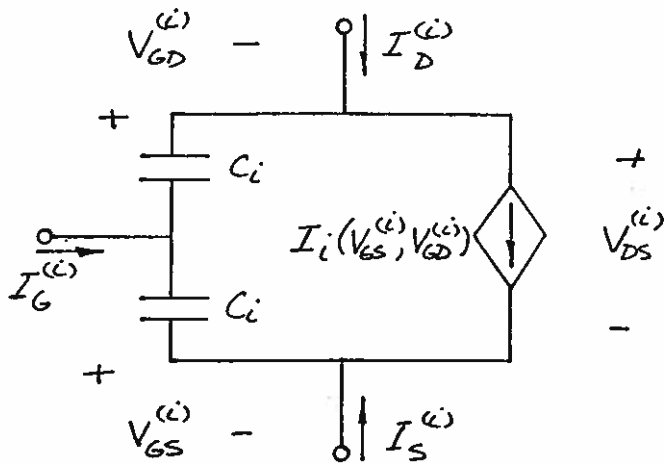
A number of the ideas in this paper were suggested or independently developed by others. Chuck Seitz of Cal Tech built a pausable clock for an Evans and Sutherland graphics system marketed in the early 1970s. Ivan Sutherland designed the MOS instability detector in 1976. Pechoucek independently reported coupling an indicating flipflop and a pausable clock in a paper in 1976. Tom Chaney, Fred Rosenberger and Charlie Molnar have all contributed ideas to this paper through many discussions. We wish to acknowledge here our thanks for their help.

Appendix A

The simple MOS flipflop shown below exhibits the essential features of synchronizer behavior.



Each MOS transistor is modeled by the following equivalent circuit.



The two capacitors C_i split the gate capacitance and include other parasitic capacitance between gate and drain and between gate and source. The current source represents the static drain characteristics expressed as a function of $V_{GD}^{(i)}$ and $V_{GS}^{(i)}$ instead of the more usual, but less symmetric form in terms of $V_{GS}^{(i)}$ and $V_{DS}^{(i)}$,

$$I_i(V_{GS}^{(i)}, V_{GD}^{(i)}) = \begin{cases} \frac{\epsilon \mu_e W_i}{2DL_i} \left[(V_{GS}^{(i)} - V_{TH}^{(i)})^2 - (V_{GD}^{(i)} - V_{TH}^{(i)})^2 \right] ; V_{GS}^{(i)}, V_{GD}^{(i)} > V_{TH}^{(i)} \\ \frac{\epsilon \mu_e W_i}{2DL_i} \left[(V_{GS}^{(i)} - V_{TH}^{(i)})^2 \right] ; V_{GS}^{(i)} > V_{TH}^{(i)}, V_{GD}^{(i)} < V_{TH}^{(i)} \\ \frac{\epsilon \mu_e W_i}{2DL_i} \left[-(V_{GD}^{(i)} - V_{TH}^{(i)})^2 \right] ; V_{GS}^{(i)} < V_{TH}^{(i)}, V_{GD}^{(i)} > V_{TH}^{(i)} \end{cases}$$

(see for example [12]). In these equations ϵ is the oxide region permativity, μ_e is the channel mobility, W_i is the width of the i^{th} transistor gate, D is the oxide depth, L_i is the length of the i^{th} transistor gate and $V_{TH}^{(i)}$ is the threshold voltage for the i^{th} transistor. Each of the voltages within the circuit can be expressed in terms of the node voltages V_1 , V_2 and V_{DD} . Node equations can be written at the drains for Q_1 and Q_2 which include the two current sources I_{x_1} and I_{x_2} . These sources represent the effects of input gating circuitry and are non-zero whenever a change of state is initiated for the flipflop.

$$2C_2(\dot{V}_2 - \dot{V}_1) - (C_1 + C_4)\dot{V}_1 - I_2(V_2, V_2 - V_1) + I_4(0, V_1 - V_{DD}) + I_{x_2} = 0 \quad (A-1)$$

$$2C_2(\dot{V}_1 - \dot{V}_2) - (C_2 + C_3)\dot{V}_2 - I_1(V_1, V_1 - V_2) + I_3(0, V_2 - V_{DD}) + I_{x_1} = 0 \quad (A-2)$$

We assume symmetry such that $C_1=C_2$ and $C_3=C_4$. Furthermore, the dimensions of Q_1 and Q_2 are the same as are those for Q_3 and Q_4 . Thus $I_1(,)$ is the same function as $I_2(,)$. A similar equivalence holds for $I_3(,)$ and $I_4(,)$. These simplifications lead to the pair of equations

$$2C_1(\dot{V}_2 - \dot{V}_1) - (C_1 + C_3)\dot{V}_1 - I_1(V_2, V_2 - V_1) + I_3(0, V_1 - V_{DD}) + I_{x_2} = 0 \quad (A-3)$$

$$2C_1(\dot{V}_1 - \dot{V}_2) - (C_1 + C_3)\dot{V}_2 - I_1(V_1, V_1 - V_2) + I_3(0, V_2 - V_{DD}) + I_{x_1} = 0 \quad (A-4)$$

Next we observe that the metastable point occurs for $\dot{V}_1 = \dot{V}_2 = 0$ and vanishing inputs with solution of both (A-3) and (A-4) given by

$$I_1(V_0, 0) = I_3(0, V_0 - V_{DD}) \quad (A-5)$$

such that both drains have the same voltage V_0 . The value of V_0 depends on $V_{TH}^{(1)}$, $V_{TH}^{(2)}$ and the fraction $\beta = \frac{W_1 L_3}{L_1 W_3}$. The symmetry evidenced in (A-5) is also present in (A-3) and (A-4) allowing us to limit our consideration to trajectories in the plane (V_1, V_2) for $V_1 > V_2$. We assume throughout that V_1 and V_2 are non-negative.

The trajectories in (V_1, V_2) of interest start near (V_0, V_0) since otherwise no metastable behavior would occur. These trajectories will end at (V_H, V_L) where these two voltages are solutions of

$$I_1(V_L, V_L - V_H) = I_3(0, V_H - V_{DD}) \quad (A-6)$$

$$I_1(V_H, V_H - V_L) = I_3(0, V_L - V_{DD}) \quad (A-7)$$

These trajectories lie near the line $V_1 + V_2 = 2V_0$. This is particularly true for the case $2V_0 = V_{DD}$ and $V_H + V_L = V_{DD}$, not unlikely conditions. Using these approximations and solving for the difference mode voltage $V_D = V_1 - V_2$ we get

$$(5C_1 + C_3)\dot{V}_D = \frac{\epsilon \mu_e W_1}{2DL_1} \left\{ V_D V_{DD} - V_D^2 - \frac{1}{\beta} V_D V_{DD} \right\} + I_{x_1} - I_{x_2} \quad (A-8)$$

where the further approximations $V_{TH}^{(1)} = 0$ and $V_{TH}^{(3)} = V_{DD}$ have been made. This equation is to be solved for $t > t_{pd}$ for which $I_{x_1}, I_{x_2} = 0$. To simplify matters normalize the voltage by V_{DD} and collect terms

$$\tau \dot{v} = v - \frac{\beta}{\beta-1} v^2 \quad (A-9)$$

where $\tau = \frac{(5C_1+C_3)2DL_1\beta}{\epsilon\mu e W_1 V_{DD}(\beta-1)}$ and $v = \frac{V_D}{V_{DD}}$. (A-9) has the solution

$$v = \frac{1}{\frac{1}{v_H} + \left(\frac{1}{v_0} - \frac{1}{v_H}\right) e^{-(t-t_{pd})/\tau}} ; \quad t \geq t_{pd} \quad (A-10)$$

where we note that

$$v \Big|_{t=t_{pd}} = v_0 \quad (A-11)$$

$$v \Big|_{t=\infty} = v_H = \frac{\beta-1}{\beta} \quad (A-12)$$

so that v_0 and v_H are the initial normalized difference voltage at $t=t_{pd}$ and the final voltage, respectively.

The response time t_r can now be found in terms of the stabilization threshold v_S for the normalized difference voltage merely by substituting $v=v_S$ and $t=t_r$ in (A-10). Solving for the initial voltage v_0 required to produce a particular response time t_r yields

$$\frac{v_0}{v_S} = \frac{1}{k + (1-k)e^{(t_r-t_{pd})/\tau}} ; \quad t_r \geq t_{pd} \quad (A-13)$$

where the abbreviation $k = \frac{v_S}{v_H}$ has been used. Define the window for marginal triggering to be the input conditions that lead to $|v_0| < v_S$ or equivalently $t_r > t_{pd}$. For input edges different by $t_d \in M$ we assume these conditions on v_0 hold and furthermore assume that a uniform distribution for t_d leads to a

uniform distribution for v_0 . These assumptions yield the following result for an arbitrary bound t_b .

$$\begin{aligned} P(t_r > t_b) | t_d \in M &= P(t_r > t_b | |v_0| < v_S) \\ &= P(|v_0| < v_B | |v_0| < v_S) = \frac{v_B}{v_S} \end{aligned} \quad (\text{A-14})$$

where v_B is the initial voltage that causes the flipflop to stabilize at precisely the bound t_b . Substituting (A-13) with $t_r = t_b$ and $v_0 = v_B$ into (A-14) we obtain Equation 2 in the text.

Appendix B

Assume the number of logic signal transitions for $t \geq 0$ can be described as a counting process $\{N(t), t \geq 0\}$ having stationary, independent increments with unit jumps; namely, a Poisson process,

$$P(N(t)=n) = e^{-\lambda t} \frac{(\lambda t)^n}{n!} \quad (\text{B-1})$$

$$E[n] = \lambda t$$

Random selection with probability p from this process gives a new counting process $\{M(t), t \geq 0\}$ (see [13]), also Poisson, where

$$P(M(t)=m) = e^{-\mu t} \frac{(\mu t)^m}{m!} \quad (\text{B-2})$$

$$E[m] = \mu t = p\lambda t.$$

REFERENCES

1. Couranz, G. R. An Analysis of Binary Circuits Under Marginal Triggering Conditions, Computer Systems Laboratory, Washington University, St. Louis, Missouri, Technical Report #15, November 1969. Also see Couranz, G. R. and Wann, D. F., "Theoretical and Experimental Behavior of Synchronizers Operating in the Metastable Region", IEEE Trans. on Computers, C-24, 1975.
2. Chaney, T. J. and Molnar, C. E., "Anomalous Behavior of Synchronizer and Arbiter Circuits", IEEE Trans. on Computers, C-22:421-422, April 1973.
3. Chaney, T. J., The Synchronizer 'Glitch' Problem, Computer Systems Laboratory, Washington University, St. Louis, Missouri, Technical Report #47, February 1974.
4. Wann, D. F., Molnar, C. E., Chaney, T. J. and Hurtado, M., "A Fundamental Problem Associated with the Physical Realization of Certain Classes of Petri-Nets", Conf. on Petri-Nets and Related Methods, MIT, Cambridge, Massachusetts, S. S. Patil - Editor, July 1975, to be published. Copies available as Technical Memorandum #215, Computer Systems Laboratory, Washington University, 724 South Euclid Avenue, St. Louis, Missouri 63110.
5. Kinniment, D. J., Woods, J. V., "Synchronization and Arbitration Circuits in Digital Systems", Proc. of the IEEE, (England), Vol. 123, No. 10:961-66, October 1976.
6. Marino, L. R., "The Effect of Asynchronous Inputs on Sequential Network Reliability", IEEE Trans. on Computers, C-26, No. 11:1082-90, November 1977.
7. Littlefield, W. M. and Chaney, T. J., "The Glitch Phenomenon", Computer Systems Laboratory, Washington University, St. Louis, Missouri, Technical Memorandum #10, December 1966.
8. Hurtado, M., Dynamic Structure and Performance of Asymptotically Bistable Systems, D. Sc. Dissertation, Department of Electrical Engineering, Washington University, St. Louis, Missouri, 1975. Also see Hurtado, M. and Elliot, D. L., "Ambiguous Behavior of Logic Bistable Systems", Proceedings Thirteenth Annual Allerton Conference on Circuit and System Theory, University of Illinois at Urbana-Champaign, 1975.
9. Chaney, T. J. and Rosenberger, F. U., "Characterization and Scaling of MOS Flip Flop Performance in Synchronization Applications", Proceedings of Conference on Very Large Scale Integration Architecture, Design and Fabrication, Pasadena, California, January 22-24, 1979.

10. Adams, R. L., Castaldo, D. R. and Kurz, G. W., "Real-time Detection of Latch Resolution Using Threshold Means", U. S. Patent 3,515,998, June 1970.
11. Pechoucek, M., "Anomalous Response Times of Input Synchronizers", IEEE Trans. on Computers, C-25, No. 2:133-139, February 1976.
12. Gray, P. E., and Searle, Electronic Principles, Wiley, 1968.
13. Parzen, E., Stochastic Processes, Holden-Day, 1962.