1994

A study and characterization of metastability in AT&T's ORCA FPGAs

Alan Cunningham
Lehigh University

AUTHOR:

Cunningham, Alan

TITLE:

A Study and Characterization of Metastability in AT&T's ORCA FPGAs

DATE: January 15, 1995

A Study and Characterization of Metastability in AT&T's ORCA FPGAs

by

Alan Cunningham

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy of the Degree of

Master of Science

in the Department of

Computer Science and Electrical Engineering

Lehigh University

December 1994

This Thesis is accepted and approved in partial fulfillment of the requirements for the Master of Science.

Dec. 9, 1994

Date

Thesis Advisor

Chairperson of Department

Acknowledgments

The author would like to recognize and express his appreciation to the following individuals for their contributions:

My advisor, Dr. Frank H. Hielscher, for his patience, help, and guidance.

Frederick J. Koons for providing invaluable advice on organization, formatting, and creating figures.

Barry Britton for providing technical information on the AT&T 's Field Programmable Gate Arrays.

My supervisor, James R. Fullerton, for allowing me the time I required to complete this Thesis.

My family and friends, who have supported me throughout this endeavor.

Table of Contents

List of Figures

Abstract

Most digital systems have at least one asynchronous input, where the input signal has no time reference with the system clock. These signals must first be synchronized before they can be used by the rest of the system. The most common device used for synchronization is the simple flip-flop. Due to the lack of a time reference between the asynchronous input and the system clock, there exists a small but finite probability that the flip-flop will go into a metastable state (in which the output is neither a logic zero nor a logic one).

Many integrated circuits, such as the TTL family and programmable logic devices (PLDs), have been fully characterized for their metastable characteristics. More recently, reconfigurable Field Programmable Gate Arrays (FPGAs) are used in system design. For these FPGAs only limited metastability characteristics are available, and these characteristics are a function of the specific design which they implement.

In this thesis, the theory of calculating the mean time before failure (MTBF) of synchronization is reviewed. Characterization of The AT&T Optimized Reconfigurable Cell Array (ORCA) FPGA was performed using both simulation and experimental methods. With these characterization data, system designers will be able to reduce synchronization failures.

Various design techniques have been evaluated which reduce the probability of synchronizer failures. Two of these techniques have been examined here in detail, the cascaded synchronizer and the multiple cycle synchronizer, both of which increase the MTBF by a

power of N. A comparison of several different commercial devices was made, with emphasis on the maximum operating frequency that each device could sustain while maintaining a mean time before failure of ten years. In this comparison, the AT&T ORCA FPGA was found to have the highest operating frequency.

1.0 Introduction

Synchronization refers to coincidence in time. In digital systems, synchronization is obtained when all components share a common periodic signal (clock). This clock controls the operations and interactions of all the components. Because the components use the same common clock it is easy to analyze, simulate, and test the system. For the system to be fully synchronous, all input and output timing must be derived from this clock. Usually, however the system has at least one asynchronous input, and is therefore it is not fully synchronous.

Asynchronism is defined as lack of concurrence, or absence of synchronism. In a digital system, an asynchronous input has no time relationship to the system clock, and therefore can occur anywhere within the system clock period. Examples of asynchronous inputs include bus arbitrations, telecommunications, I/O interfaces (e.g. a computer keyboard), and data acquisitions. To utilize an asynchronous signal, it must be first synchronized by the clock before being used by the system. The circuit used for synchronization most commonly is the simple flip flop.

Flip-flops have specific requirements which must be met for the device to operate correctly. Two specific requirements are setup time and hold time. Setup time is defined as the time allowed for the data signal to remain stable prior to the active edge of the clock. Hold time is the time allowed for the data signal to remain stable after the active edge of the clock signal. If the input signal violates either one of these conditions, the flip flop will operate in an anomalous behavior. This behavior is avoided in a synchronous system by insuring that the design meets all the setup time and the hold time requirements. Because a synchronous system has a common time reference, this is a trivial task. Unfortunately, with an asynchronous input there is no time reference between the input and the system clock. The input can occur with equal probability anywhere within the clock period. Therefore, at some point in time, the asynchronous signal will violate the flip-flop's setup and hold conditions, causing an anomalous output.

The anomalous behavior of the flip-flop is called metastability. Metastable is a Greek word meaning in-between. Here, the flip-flop is in-between logical states of a one or a zero. The behavior of the output during metastability is erratic. Three types of behavior have been observed: the output hovers between a logical-one and a logical-zero, the output oscillates between a logical-one and a logical-zero, the output propagation delay is longer than normal. Since the metastable state is unstable, the flip-flop will eventually resolve the metastable condition and reach a stable output state of either a logic one or logic zero, but the final state cannot be predetermined. The duration of the metastable state is also probabilistic in nature and can theoretically last forever. The ability of the flip-flop to resolve to a known state is a function of time and the flip-flop's characteristics. If the output of the flip-flop is sampled before it has resolved to a known state, non-binary information will be transmitted through the system.

The non-binary information can be interpreted differently by various components in the system. This situation will corrupt data and cause a system error. Furthermore, the error is a random, infrequent event.

System errors caused by metastability have been listed in literature as early as 1952. Lubkin [25] discussed synchronizer failures in the ENIAC computer. The designers of the ENIAC added an additional flip-flop to eliminate errors. Lubkin's paper discussed that although the designers reduced the probability of an error, they did not eliminate all errors. His paper furthermore derived mathematical equations for the metastability phenomena based on the probability of an error.

To reduce the probability of error, the system designer must comprehend the metastability concept, and fully understand the characteristics of the devices used for synchronization. This thesis will discuss the theory of metastability, characterize the AT&T Optimized Reconfigurable Cell Array (ORCA) Field Programmable Gate Array (FPGA), and list design techniques to reduce the probability of metastable events.

2.0 Metastability Theory

Asynchronous signals have no time-relationship with the system clock. These signals must be synchronized before being used by the rest of the system. To synchronize these signals, a storage element is used.

This storage element is a latch or flip-flop. A latch stores or "latches" data on the negative sense of the system clock. A flip-flop stores data on the positive edge of the system clock. The flip-flop is actually a master-slave configuration that is constructed of two latches. The first latch is the master while the second latch acts as a slave. Because the master-slave flip-flop is a combination of two latches, it is the preferred synchronizer.

The AT&T ORCA FPGA can implement over a hundred different types of flip-flops. Some of these flip-flops include JK flip-flops, toggle flip-flops, multiplexed flip-flops, clock-enabled flip-flops, preset flip-flops, clear flip-flops, and D flip-flops. Most of these flip-flops must be implemented with additional logic. This additional logic increases the setup time of the flip-flop. Setup time should be minimized when designing synchronizers. The D flip-flop requires no additional logic thereby having a low setup time. Therefore, a master-slave D-type flip-flop is used for synchronization of asynchronous input signals. Figure 2.1 shows the D flip-flop, and it's truth table.

Metastability is a phenomenon where the output of a flip-flop is undefined. The output may hover between stable logical states, it may oscillate between the stable states, or it may reach a stable state after a longer than normal propagation delay. This phenomenon is caused by marginal triggering of the flip-flop. The output goes into an anomalous behavior because the data has violated the specified setup and hold conditions.

In the case of a D type flip-flop, the time that the data must be stable at the D input of the device prior to the clock edge is known as the setup time, and the time that this data must remain stable after the clock edge is known as the hold time (figure 2.2). The data must satisfy both the setup time and hold time to insure that the flip-flop stores valid data, and to insure that the outputs present valid data after a specified propagation delay (T_p). T_p is specified as the time from the edge of the clock until the time that the output reaches a valid state.

If the data violates the setup time or hold time, the flip-flop output may go to an anomalous state for a time greater than T_p. It may take the outputs anywhere from a hundred picoseconds to a microsecond to reach a valid output level. The amount of additional time needed (beyond T_p) is the resolving time (T_r). This resolving time is statistically predictable, but is not deterministic because electrical noise is a factor in reaching the final point.

Figure 2.3 shows the variation in output delay with relation to the input timing of the flip-flop. The left portion of the graph shows that when the data input meets the required setup time, the device has a valid output after a predictable delay equal to T_p. The right portion shows that when the data input arrives after the clock, the output does not change states. The middle portion of the graph indicates the metastable region. If the data input changes near the clock edge, the output delay is longer than T_p. The closer the transition occurs to the clock edge the greater the delay. The region, where the input transition causes output delays longer than T_p, is the metastable time window (T_w). T_w is given by equation (2.1) [42]

$$T_w(t)=T_0*\exp(-(t - T_p)/\tau) \tag{2.1}$$

where T_p is the normal propagation delay of the flip-flop. τ is the time constant of resolution. This time constant is a function of the device specific characteristic such as the gain of the flip-flop and transistor parasitics.

Figure 2.4 shows a physical analogy of the metastability problem. A flip-flop like any other bistable system, has two minimum potential energy levels, separated by a maximum energy potential. A bistable system has stability at either of the two minimum energy points. The system can also have temporary stability, known as metastability, at the energy maximum. If nothing pushes it from the maximum energy point, the system will remain at this point indefinitely. A smooth ball on a hill is another bistable system. A ball placed on top of the hill will tend to roll toward one of the minimum energy levels. If left undisturbed at the top, it may remain there for an indeterminate amount of time. It is apparent from this figure that the characteristics of the hill affect how long the ball will stay there. The steepness of the hill is analogous to the gain of the flip-flop.

2.1 Analysis of the Simple Latch

The simplest form of the static latch [45] consists of a pair of cross-coupled inverters. Where the inverters represent the non-linearity of the flip-flop, and an RC network represents the dynamic portion. This simplified representation is shown in figure 2.5.

The transfer curve for the simplified circuit is shown in figure 2.6. The curve shows three possible solutions. Points A and B represent stable solutions for the latch. At both points the slope or gain of the circuit is less than one. Point M represents an unstable solution. At this point the gain is greater than one. Any change in the input voltage will be amplified forcing the output voltage to one of the stable solutions (A or B).

During the analysis, a few assumptions will be made. The absolute value of the gain |A| will be used. The output impedance of the inverter is R. The capacitance C consists of the output capacitance, the input capacitance, and the parasitic routing capacitance. Both the inverters are identical. Therefore, the latch is in the metastable state when $V_{QN} = V_Q = V_M$.

Writing the Kirchoff's equation for nodes QN and Q

$$(A{*}V_{QN} - V_Q)/R - C(dV_{QN}/dt) = 0 \qquad (2.2)$$

$$(A{*}V_Q - V_{QN})/R - C(dV_Q/dt) = 0 \qquad (2.3)$$

Rearranging equations (2.2) and (2.3) and subtracting them yields

$$RC^*d(V_{QN} - V_Q)/dt + (V_{QN} - V_Q) + A(V_{QN} - V_Q) = 0 \tag{2.4}$$

$$\text{let } V_{QN} - V_Q = V_d \tag{2.5}$$

The expression V_d represents the difference between the node voltages that causes the regenerative action to return the flip-flop outputs to stable states.

Replacing equation (2.5) into equation (2.4)

$$RC^*dV_d/dt + V_d + AV_d = 0 \tag{2.6}$$

Rearranging equation (2.6) yields

$$dV_d/dt + ((A + 1)/RC)V_d = 0 \tag{2.7}$$

Equation (2.7) is a linear differential equation with a solution of the form

$$y(x) = B^*\exp(Ax) \tag{2.8}$$

Therefore, the solution for equation (2.7) is

$$V_d(t) = B^*\exp(((A + 1)/RC)t) \tag{2.9}$$

In this equation, there are two factors with the following physical interpretations:

$$\tau = RC/(1 + A) \tag{2.10}$$

$$V_d(0) = B \tag{2.11}$$

τ is the time constant that controls how quickly the flip-flop resolves to a stable state. The variable $V_d(0)$ is the initial voltage difference between nodes V_Q and V_{QN} that cause the outputs to be metastable.

The final equation for the difference voltage between nodes Q and QN is equation (2.12).

$$V_d(t) = V_d(0)*\exp(t/\tau) \qquad (2.12)$$

Equation (2.12) shows that the difference voltage V_d will grow exponentially with time. This exponentially growth causes the flip-flop to resolve from a metastable state. The rate at which it resolves is dependent on τ. The constant τ is a function of the gain of the inverter, the output impedance of the inverter, and the capacitance of inverter. To reduce this constant, which would increase the resolving rate, the gain should be increased and the impedance and capacitance should be decreased.

When the flip-flop's outputs resolve from the metastable state, the final difference voltage occurs at $t = T_r$ This difference is the noise margin of the inverter (VIH - VIL). Using this value for $V_d(T_r)$, the initial voltage difference as a function of time can be solved.

$$V_d(t) = (VIH - VIL)*\exp(-T_r/\tau) \qquad (2.13)$$

2.2 Statistically Analysis of Metastability Failures

Again using the simple latch, a statistical equation for metastability failures can be found [18]. Relate the metastable voltage $V_d(t)$ to the metastable time window. In figure 2.7, the metastable voltage increases through the metastable time window at a rate of dV_d/dt. This rate is directly related to the slew rate given by

$$\text{slew rate} = dV/dt = (VDD - 0)/trise \qquad (2.14)$$

Using equation (2.14) the metastable window $T_w(t)$ can be determined

$$T_w(t) = V_d(t)*(dt/dv) = V_d(t)*(trise/VDD) \qquad (2.15)$$

Combining equation (2.13) and (2.15) the metastable time window $T_w(t)$ can be found

$$T_w(t) = (((VIH - VIL)/VDD)*Trise)*\exp(-T_r/\tau) \qquad (2.16)$$

Letting

$$T_0 = (((VIH - VIL)/VDD)*Trise) \qquad (2.17)$$

then equation (2.16) becomes

$$T_w(t) = T_0*\exp(-t/\tau) \qquad (2.18)$$

If the phase of the clock and data signal are un-correlated, which is the case for asynchronous signals, the probability density function is unity for one clock period. The probability that the flip-flop is metastable within a given clock period is given by

$$P = T_w(t)/T_c \qquad (2.19)$$

The number of occurrences of the metastable event depends on the frequency of the input signal. Also, since there are two transitions through the metastable time window, an additional factor of two has been included.

$$N = 2Tw(t)*F_c*F_d \qquad (2.20)$$

The mean time before failure (MTBF) is

$$MTBF = 1/N = 1/(2*T_w(t)*F_c*F_d) \qquad (2.21)$$

Combining equation (2.18) and equation (2.21)

$$MTBF = \exp(T_r/\tau)*(1/(2*T_0*F_c*F_d)) \qquad (2.22)$$

Equation (22) shows some interesting information about metastability error rates. The rate has an exponential dependence on the time (t) that is given to resolve the outputs to a stable state. Clearly, the first way to reduce errors is to increase t by waiting a little longer. Also, the rate has an exponential dependence on the inverse of τ. τ is a function of specific device characteristics such as gain, capacitance, and output impedance. By using a device with a better τ, the error rate will be reduced. The constant T_0 is also a function of device characteristics, but it also is affected by the input slew rate and the power supply voltage. Increasing the slew rate, will also improve the error rate.

2.3 An Example

As a simple example of a synchronizer, consider two D flip-flops from the Schottky TTL family, with specific values for the 74S74 (figure 2.8). If the asynchronous input changes during the metastable window of FF0, its output Q0 may become metastable until time T_r. If the output Q0 is still metastable at the beginning of the metastable window for FF1, then the synchronizer will fail because FF1 may have a metastable output.

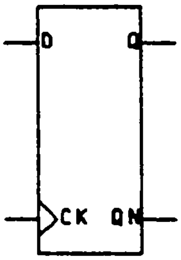
Using values from [47] for the 74S74, calculate the mean time before failure for the synchronizer using equation (2.22). The resolution time T_r is a function of the clock period (T_c), flip-flop delay (T_p), flip-flop setup time (T_{set}), the logic delay between the flip-flops (T_d), and routing delay between the flip-flops (T_{route}). The resolution time can be calculated by using equation (2.23).

$$T_r = T_c - T_p - T_{set} - T_d - T_{route} \qquad (2.23)$$

In this example, assume that there is no logic delay or routing delay between the flip-flops. The frequency of operation is 10 MHz resulting in a clock period of 100 nS. The flip-flop setup time is 20 nS, and the propagation delay of the flip-flop is 25 nS. The data rate for this example is 1 MHz. Using equation (23), the resolution time is 55 nS. The value of the metastability resolution constant (τ) is 1.7 nS. The constant T_0 is 1.0 mS. Substituting these values into equation (16), solve for the MTBF.

$$MTBF = (exp(55/1.7))/(2*1.0E-03*1E+07*1E+06) = 5.62E+3\ S$$

This value may seem large, but it actually states that there will be one failure in 1.56 hours. If this synchronizer is used in 1000 systems, there will be a system failure every 5.62 seconds!



Inputs		Outputs	
D	CK	Q	QN
0	↑	0	1
1	↑	1	0

Figure 2.1: D Flip-Flop with Truth Table

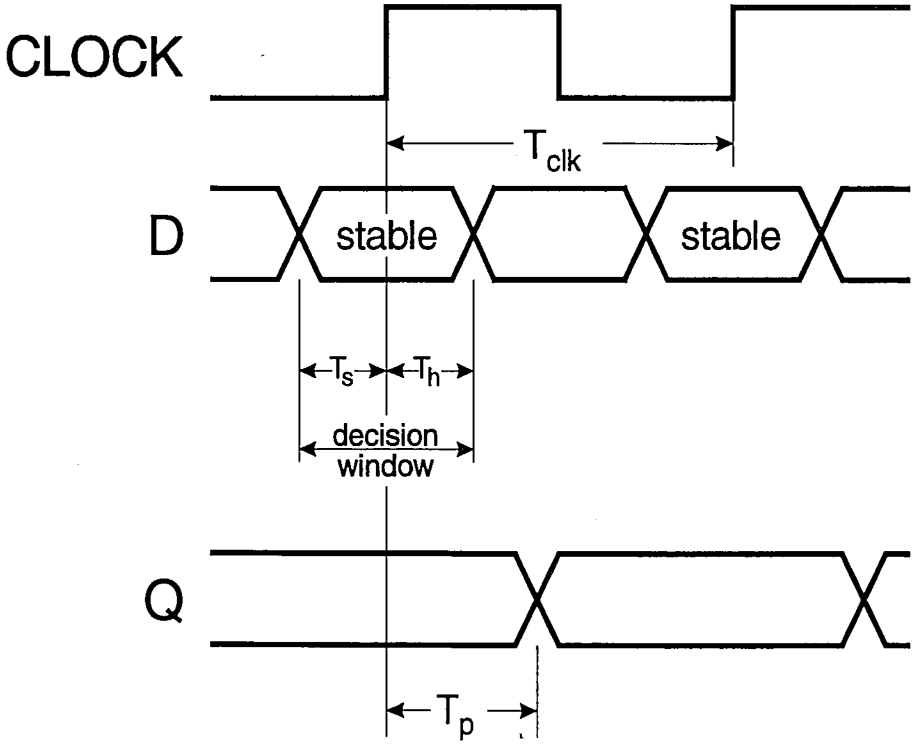


Figure 2.2: Flip-Flop Timing Parameters

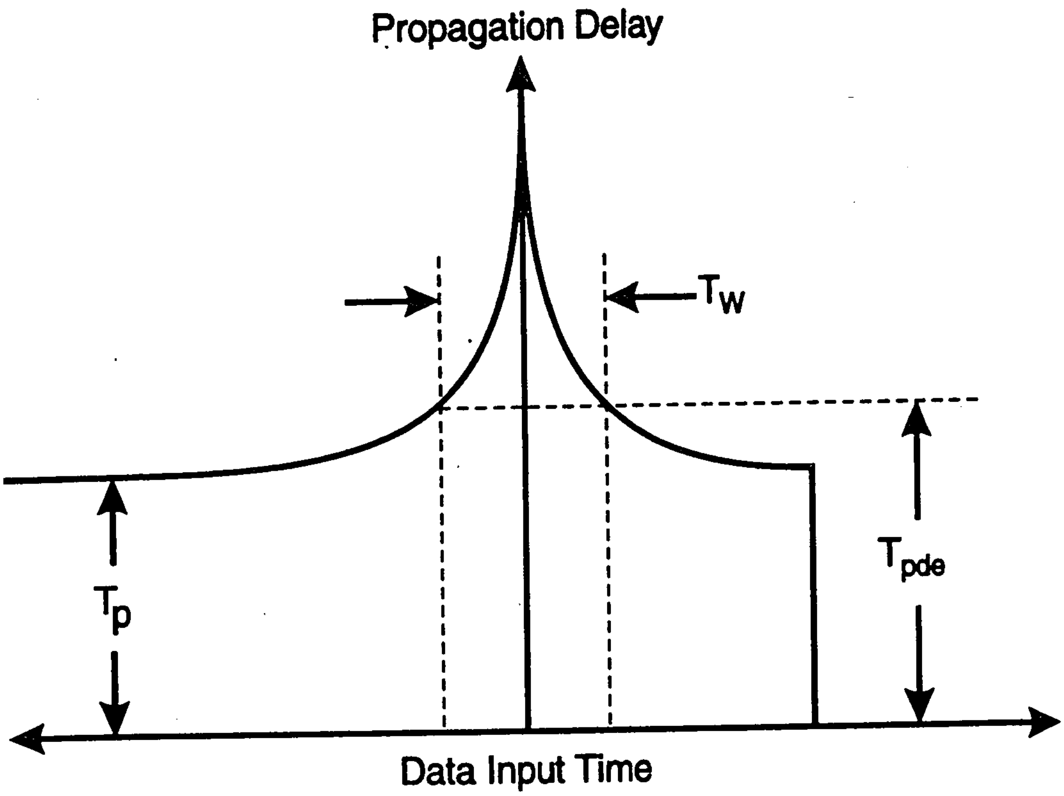


Figure 2.3: Metastable Time Window

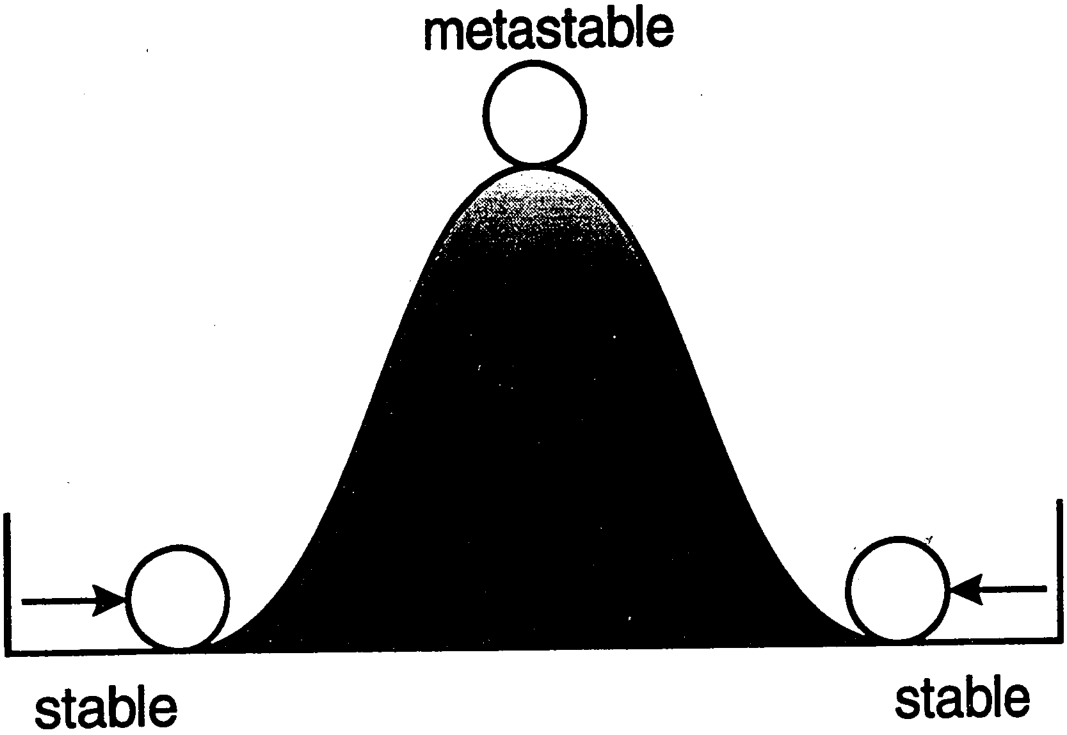


Figure 2.4: A Smooth Ball on a Hill

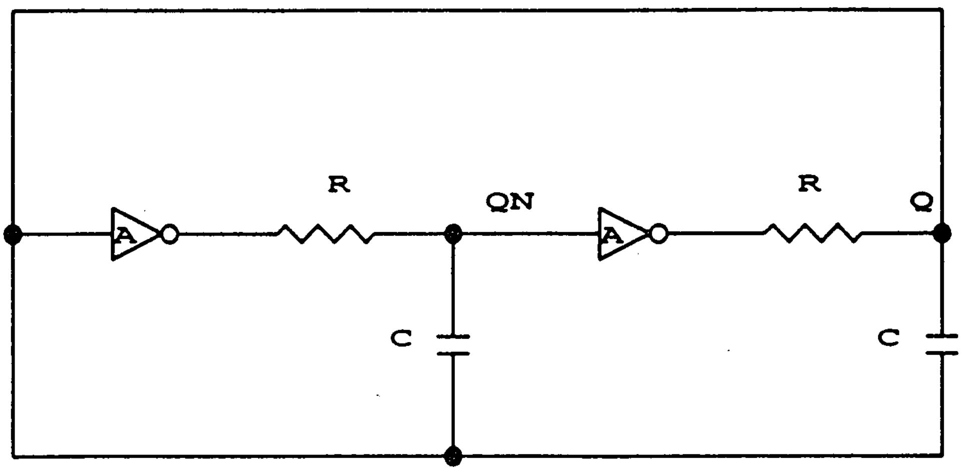


Figure 2.5: Simple Latch

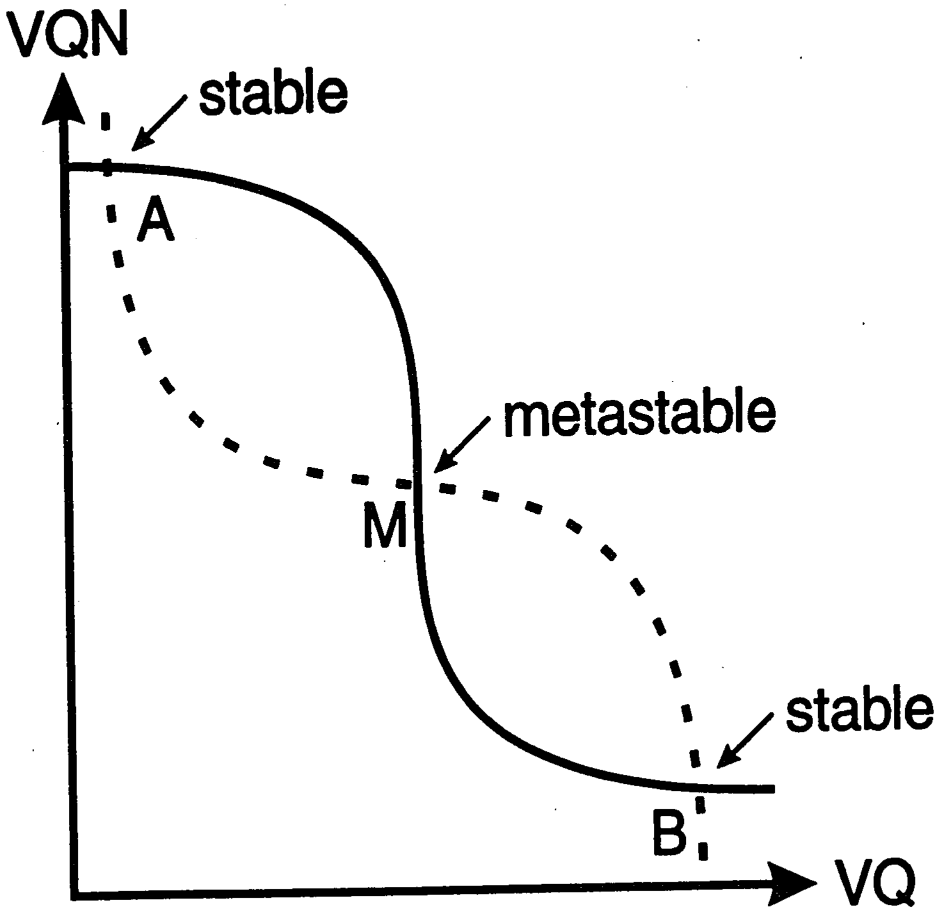


Figure 2.6: Transfer Curve for Simple Latch

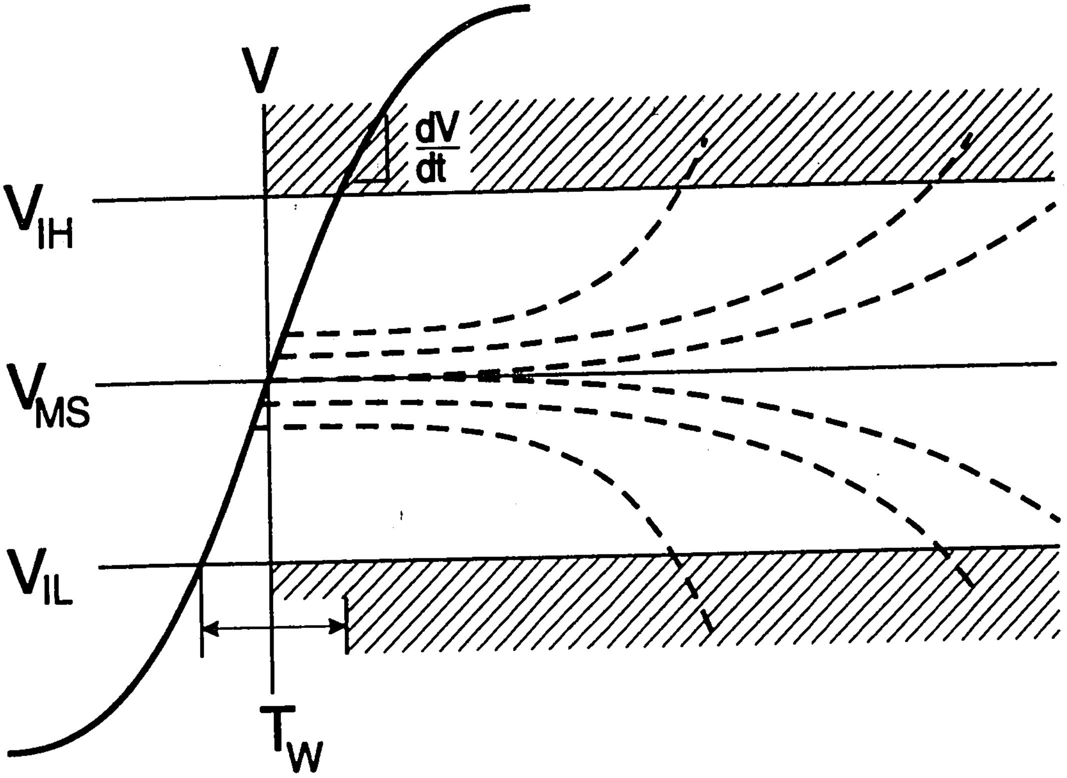


Figure 2.7: The Metastable Voltage as a Function of Time

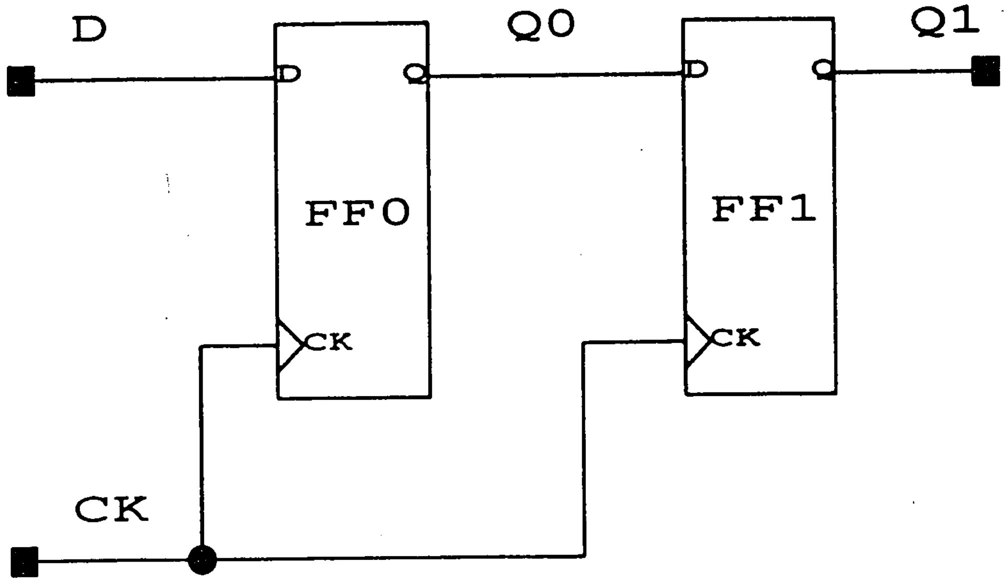


Figure 2.8: Simple Synchronizer

3.0 Discussion of AT&T's FPGA

Masked Programmable Gate Arrays (MPGAs) allow the implementation of powerful digital circuits. The MPGA consists of rows of transistors interconnected to implement a desired design. Connections within the rows implement the logic gates while connections between the rows join the gates together. Additionally, logic surrounds the rows providing input and output connections to the MPGA's external pins. In an MPGA, all the mask layers that define the circuitry are predefined by the manufacturer except the final metal mask layers. These metal layers are customized to connect the transistors in the array, therefore implementing the desired circuit. MPGA's have a large non-recurring engineering (NRE) charge required to create the metal mask layers and to manufacture the chip. This NRE charge can cost anywhere from $20,000 to $40,000. Furthermore, MPGA's require extensive manufacturing effort, taking several weeks to several months to create a device.

Similar to an MPGA, a Field Programmable Gate Array (FPGA) consists of an array of elements that can be interconnected. Unlike an MPGA, the element's logic and the interconnection of these elements are programmable by the user. Since the user controls the connectivity and logic of the FPGA, there is no NRE charge for creating mask layers or for manufacturing the chip. Furthermore, because there is no manufacturing effort a device can be created in days instead of months.

An FPGA has a two-dimensional array of cells that implement the digital logic of the design (Figure 3.1). The array is surrounded by a ring of input/output (I/O) cells that connect to the FPGA's external pins. An interconnecting structure connects the logic blocks together and connects the logic blocks to the I/Os. All the components of the FPGA including the logic blocks, the I/O cells, and the interconnect are user-programmable. The user-programmable element varies with technology. The programming technologies used in commercial products are: Static Read-Access Memory (SRAM) cells, anti-fuses, Erasable Programmable Read-Only Memory (EPROM) transistors, and Electrical Erasable Programmable Read-Only Memory (EEPROM) transistors. The two prevalent technologies for FPGAs are SRAM and anti-fuse.

SRAM FPGAs are volatile, and therefore these FPGAs must be programmed each time the device is powered-up. During power-up, the programming information for the FPGA is sent from a storage element such as a ROM or a disk. By changing the programming information, the FPGA can be modified quickly (a few milliseconds) while on a board. The modification or re-configurability of these FPGAs allows many design changes that are invaluable during the prototyping of a system.

In SRAM FPGAs, there are two types of programmable connection elements. The first element is a pass transistor (Figure 3.2a) which can connect a logic cell to a metal segment, or two metal segments together. The gate of the pass transistor is controlled by an SRAM bit. When the SRAM bit is a one, there is a low resistance (on-resistance) path through the transistor creating a connection. When the SRAM bit is a zero, there is a high resistance (off-resistance) path through the transistor breaking the connection. This on-resistance affects the performance of the FPGA. A pass transistor has an on-resistance that varies from 1000 ohms to 2000 ohms. The capacitance for the transistor varies from 10 fF to 20 fF. Because of these high values for resistance and capacitance, SRAM based FPGAs can have large interconnect delays. The second element is a multiplexer (Figure 3.2b). Here, the SRAM bits control which of the multiplexer's inputs should be connected to its output. This is used to tie several wires to a single input.

The chip area required for SRAM FPGAs is quite large. This is because five transistors are required for each RAM cell, and there are additional transistors needed for the pass transistors or multiplexers. A major advantage of SRAM FPGAs is that they can be manufactured in a standard CMOS process technology.

Anti-fuse FPGAs are non-volatile. The anti-fuse is normally a high impedance, but can be fused into a low impedance when programmed with a high voltage. This high voltage is provided by a third-party programmer. After an anti-fuse FPGA is programmed, it cannot be modified.

The anti-fuse is a square structure that consists of three layers: the bottom layer is positively-doped silicon (n+), the middle layer is a dielectric, and the top layer is poly silicon. The anti-fuse is programmed by placing a high voltage across the anti-fuse terminals. This programming generates heat in the dielectric causing it to melt and form a conductive link between the doped silicon and the poly silicon. Metal wires are connected to the bottom layer and to the top layer of the anti-fuse. Therefore when programmed, the anti-fuse provides a low resistance (on-resistance) connection between the two wires. The on-resistance for the anti-fuse varies from 300 ohms to 500 ohms. Also, the capacitance value varies from 3 fF to 5 fF. With low resistance and capacitance values, Anti-fuse FPGAs have small interconnect delays.

Anti-fuse FPGAs require a small chip-area for the programming element. Yet this small area is offset by the area required for the high-voltage transistors needed to handle the programming voltages. A major disadvantage of anti-fuse FPGAs is that they require modification to the standard CMOS process technology.

In this thesis, only FPGAs that have published metastability data will be looked at. However, they are other FPGAs that may have excellent metastability characteristics. The four FPGAs considered are the Actel ACT1, the QuickLogic Q12X16-2, the Xilinx XC3030-70, and the AT&T ORCA.

The Actel ACT1 is an anti-fuse based FPGA fabricated in a 2.0 micron CMOS process [1]. The ACT1 FPGA family has devices with logic densities from 1,200 gates to 2,000 gates. Figure 3.3 shows the ACT1 logic cell. The logic cell is small and simple. Logic is implemented in the cell by using a configuration of multiplexers. This cell can implement any function of two variables, most functions of three, some of four, up to a total of 702 logic functions. Flip-flops must also by implemented using these multiplexers. The anti-fuse used for interconnect has an on-resistance of 400 ohms, and a capacitance of 4 fF.

The QuickLogic Q12X16-2 is an anti-fuse based FPGA fabricated in a 0.65 micron CMOS process [35]. The QuickLogic FPGA family has devices with logic densities from 1,500 gates to 12,000 gates. Figure 3.4 shows the QuickLogic function cell. In the cell, logic is implemented by using the AND gates and the multiplexer gates. This cell also includes a dedicated flip-flop. In the QuickLogic FPGA, a via-link anti-fuse is used for interconnect. This type of anti-fuse has an on-resistance of approximately 65 ohms, and a capacitance of 1.3 fF.

The Xilinx 3020-70 is an SRAM based FPGA fabricated in a 1.25 micron CMOS process [51]. The Xilinx 3000 family of FPGA has gate densities from 2,000 gates to 9,000 gates. The Xilinx 3000 FPGA is very similar to the AT&T ORCA FPGA, and in fact was the basis for the AT&T FPGA. Figure 3.5 shows the Xilinx function cell. Here, the logic is implemented in look-up tables (LUTs). The cell includes two dedicated flip-flops. One of the flip-flops can have an input that bypasses the LUT, decreasing the flip-flops' setup time. The other flip-flop input must come through the LUT, increasing that flip-flops' setup time. The outputs of the flip-flops go through a buffer before leaving the logic cell. In the Xilinx FPGA, a pass transistor is used for interconnect. This pass transistor has a high on-resistance of 1500 ohms, and a high capacitance of 15 fF.

The AT&T ORCA is an SRAM based FPGA fabricated in a 0.5 micron CMOS process [4]. This FPGA family has gate densities from 3,500 gates to 26,000 gates. As stated previously, the AT&T FPGA architecture was based on the Xilinx FPGA. For simplicity, the AT&T FPGA logic cell has double the logic, and double the number of flip-flops of the Xilinx FPGA. This architecture will be described in depth later in this section. Logical circuits are implemented in LUTs. There are four dedicated flip-flops. All the data inputs for the flip-flops can come from either the LUTs, or these inputs can bypass the LUTs. Also, the outputs of the flip-flops are buffered before leaving the logic cell. In the AT&T FPGA, a pass transistor is used for interconnect. This pass transistor has an on-resistance of 500 ohms, and a capacitance of 5fF.

The AT&T ORCA FPGA (figure 3.6) consists of array of programmable logic cells (PLCs) surrounded by programmable input/output cells (PICs). Programmable routing resources are used for PLC-to-PLC connections, and PLC-to-PIC connections.

The programmable logic cells (PLCs) provide the functional elements for constructing the digital circuit. Each PLC has a combinatorial logic section, and a storage section. Each PLC has nineteen possible inputs and six possible outputs. Figure 3.7 shows the resources of the PLC.

The PLC's combinatorial logic uses a 64-bit look-up table (LUT) memory to implement Boolean functions. The PLC can be configured to generate any function of six inputs, any two functions of five inputs, or four functions of four inputs (with some shared inputs), and several functions of eleven inputs. The PLC can also be configured to implement arithmetic functions such as a four-bit counter, or four-bit adder/subtractor. Alternately, a PLC can be used as a 16X4 memory cell or two 16X2 memory cells.

The PLC has four registers. These registers can either be configured as a flip-flop or a latch. The inputs for these registers can come from either the LUTs, or they can bypass the LUTs to reduce the input setup time. A dedicated 2-1 multiplexer can also be used to select the input to the register. The outputs of the PLC can come directly from the LUTs or they can be registered before leaving the PLC. The registers in the PLC have many programmable features such as a global set/reset, a local set/reset, a clock enable, and a clock inversion.

The programmable input/output cells (PICs) are located on the periphery of the array. They provide the interface between the external package pins of the device and the internal logic. Each PIC (figure 3.8) can be configured to be either an input, an output, or a bidirectional I/O. Inputs can be configured as either TTL or CMOS compatible. To allow zero hold time on the internal registers, the input signal can be delayed. Pull-up or pull-down resistors are available on the inputs to reduce power consumption. The output slew rate is also programmable to reduce ground bounce.

Programmable interconnections in the FPGA are used to provide routing paths to connect inputs and outputs of the PICs and PLCs. All interconnections are composed of metal segments and programmable switching elements. The metal segments are broken into direct connections, connections that span one PLC, connections that span four PLCs, connections that span half the array, and connections that span the whole array. The programmable switching elements are the pass transistor, and the multiplexer. The pass transistor can connect a PLC output to a metal wire, or connect two metal wires together. The multiplexer connects one of its inputs to its output.

The ORCA FPGAs range in density from 3,000 gates to 26,000 gates. They use SRAM programming elements to implement complex digital design. Because these devices are user-programmable, they have no NRE charges and have quick design cycles.

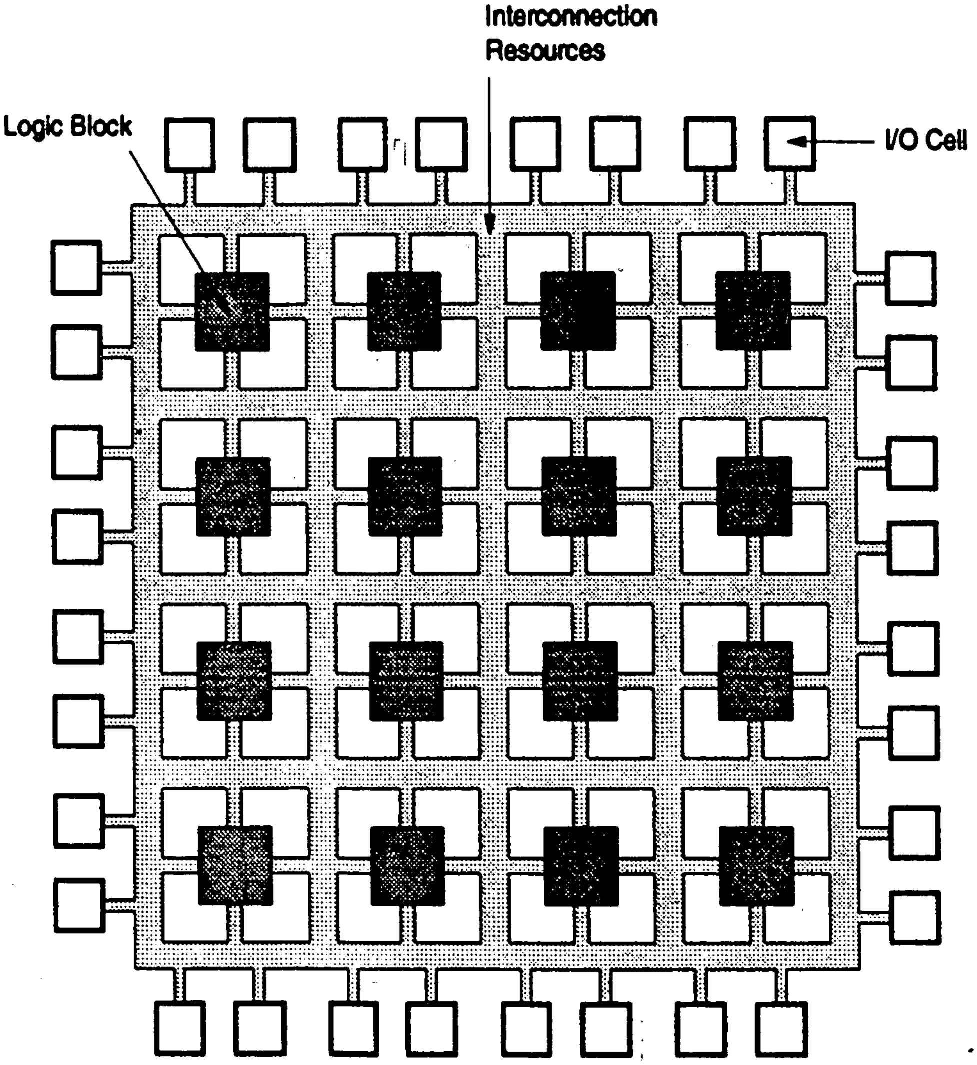
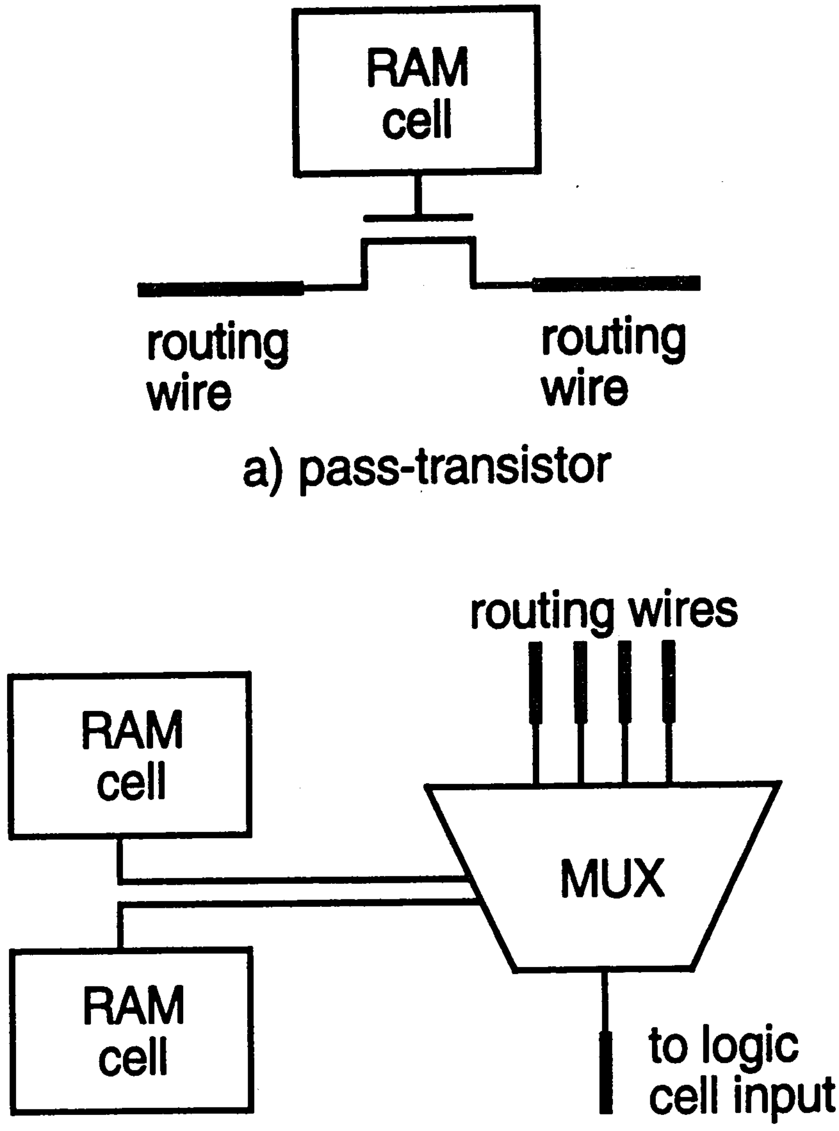


Figure 3.1: Block Diagram of a Field Programmable Gate Array [7]



a) pass-transistor

b) multiplexer

Figure 3.2: SRAM Programming Elements

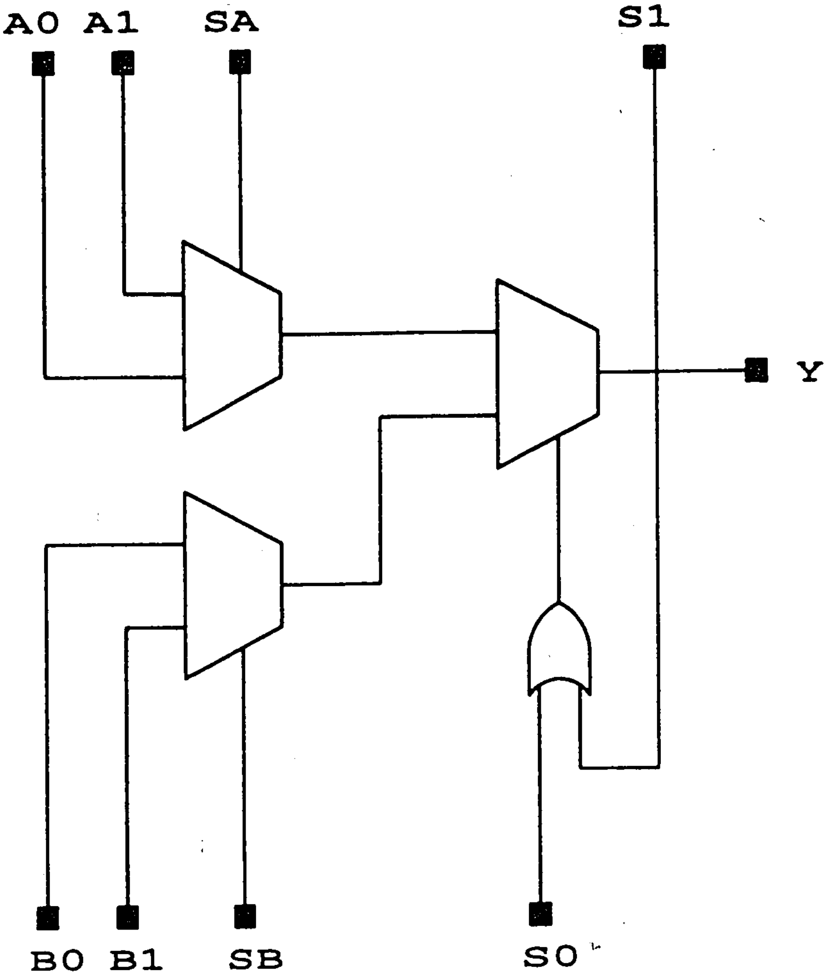


Figure 3.3: Actel ACT1 Logic Cell

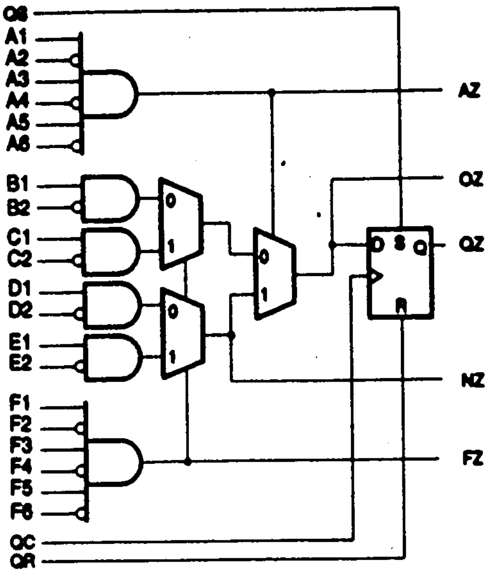


Figure 3.4: QuickLogic Function Cell [35]

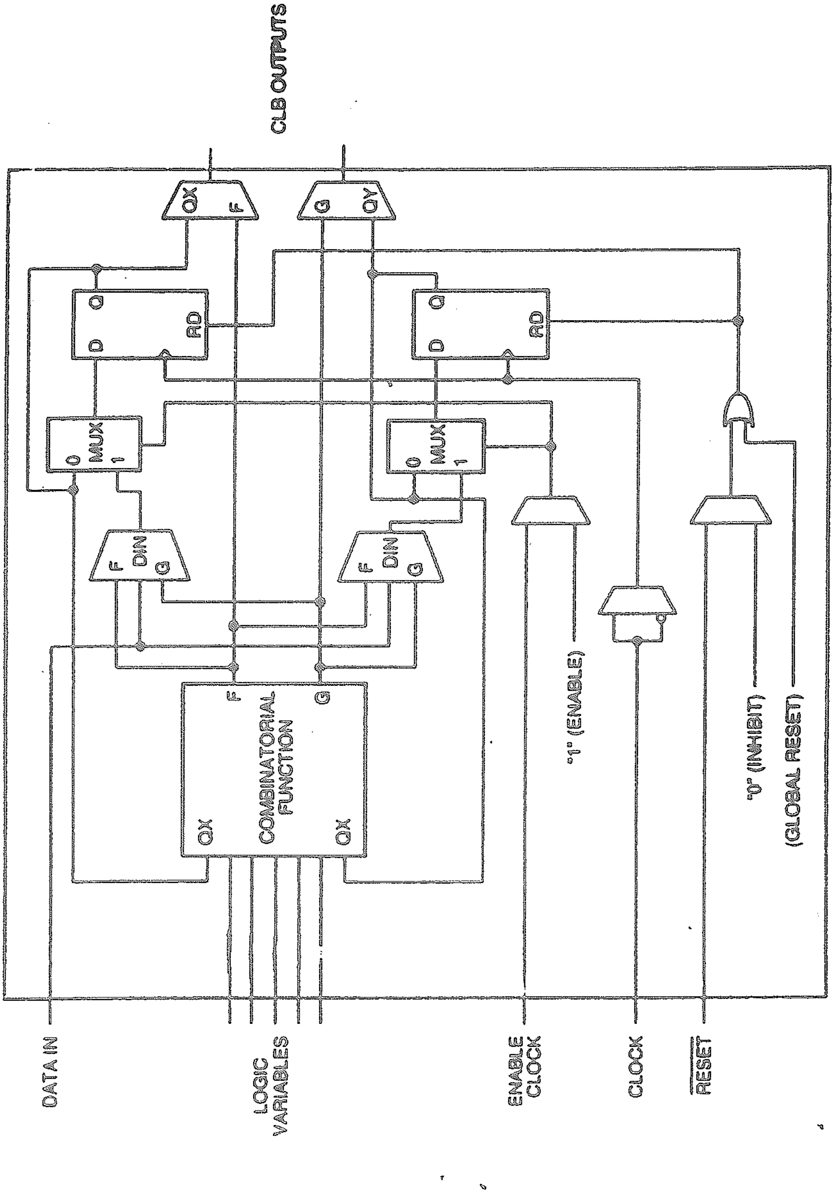
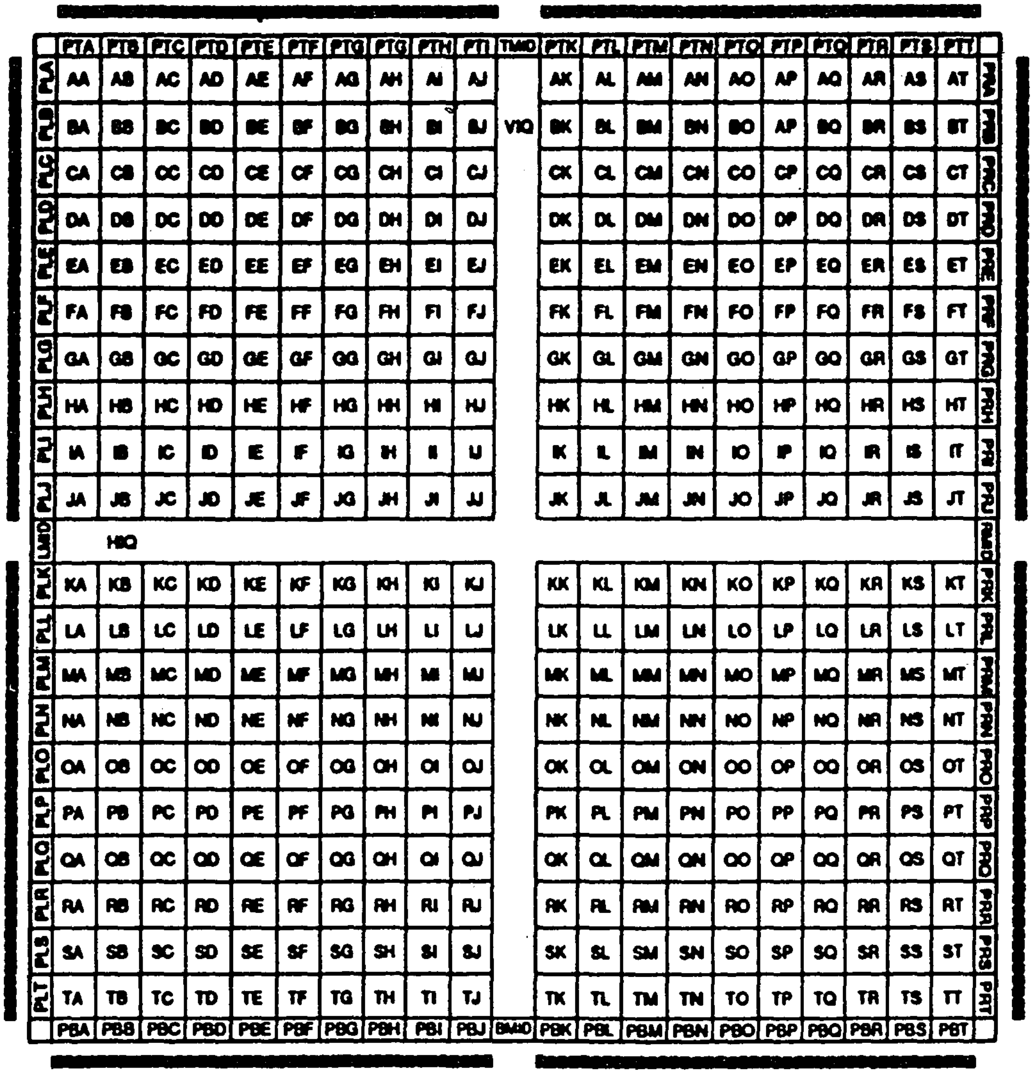


Figure 3.5: Xilinx XC3000 Function Cell [51]



	PTA	PTB	PTC	PTD	PTE	PTF	PTG	PTG	PTH	PTI	TMID	PTK	PTL	PTM	PTN	PTO	PTP	PTQ	PTR	PTS	PTT	
PLA	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ		AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	PRA
PLB	BA	BB	BC	BD	BE	BF	BG	BH	BI	BJ	VIQ	BK	BL	BM	BN	BO	AP	BQ	BR	BS	BT	PRB
PLC	CA	CB	CC	CD	CE	CF	CG	CH	CI	CJ		CK	CL	CM	CN	CO	CP	CQ	CR	CS	CT	PRC
PLD	DA	DB	DC	DD	DE	DF	DG	DH	DI	DJ		DK	DL	DM	DN	DO	DP	DQ	DR	DS	DT	PRD
PLE	EA	EB	EC	ED	EE	EF	EG	EH	EI	EJ		EK	EL	EM	EN	EO	EP	EQ	ER	ES	ET	PRE
PLF	FA	FB	FC	FD	FE	FF	FG	FH	FI	FJ		FK	FL	FM	FN	FO	FP	FQ	FR	FS	FT	PRF
PLG	GA	GB	GC	GD	GE	GF	GG	GH	GI	GJ		GK	GL	GM	GN	GO	GP	GQ	GR	GS	GT	PRG
PLH	HA	HB	HC	HD	HE	HF	HG	HH	HI	HJ		HK	HL	HM	HN	HO	HP	HQ	HR	HS	HT	PRH
PLI	IA	IB	IC	ID	IE	IF	IG	IH	II	IJ		IK	IL	IM	IN	IO	IP	IQ	IR	IS	IT	PRI
PLJ	JA	JB	JC	JD	JE	JF	JG	JH	JI	JJ		JK	JL	JM	JN	JO	JP	JQ	JR	JS	JT	PRJ
LMID		HIQ																				RMID
PLK	KA	KB	KC	KD	KE	KF	KG	KH	KI	KJ		KK	KL	KM	KN	KO	KP	KQ	KR	KS	KT	PRK
PLL	LA	LB	LC	LD	LE	LF	LG	LH	LI	LJ		LK	LL	LM	LN	LO	LP	LQ	LR	LS	LT	PRL
PLM	MA	MB	MC	MD	ME	MF	MG	MH	MI	MJ		MK	ML	MM	MN	MO	MP	MQ	MR	MS	MT	PRM
PLN	NA	NB	NC	ND	NE	NF	NG	NH	NI	NJ		NK	NL	NM	NN	NO	NP	NQ	NR	NS	NT	PRN
PLO	OA	OB	OC	OD	OE	OF	OG	OH	OI	OJ		OK	OL	OM	ON	OO	OP	OQ	OR	OS	OT	PRO
PLP	PA	PB	PC	PD	PE	PF	PG	PH	PI	PJ		PK	PL	PM	PN	PO	PP	PQ	PR	PS	PT	PRP
PLQ	QA	QB	QC	QD	QE	QF	QG	QH	QI	QJ		QK	QL	QM	QN	QO	QP	QQ	QR	QS	QT	PRQ
PLR	RA	RB	RC	RD	RE	RF	RG	RH	RI	RJ		RK	RL	RM	RN	RO	RP	RQ	RR	RS	RT	PRR
PLS	SA	SB	SC	SD	SE	SF	SG	SH	SI	SJ		SK	SL	SM	SN	SO	SP	SQ	SR	SS	ST	PRS
PLT	TA	TB	TC	TD	TE	TF	TG	TH	TI	TJ		TK	TL	TM	TN	TO	TP	TQ	TR	TS	TT	PRT
	PBA	PBB	PBC	PBD	PBE	PBF	PBG	PBH	PBI	PBJ	BMID	PBK	PBL	PBM	PBN	PBO	PBP	PBQ	PBR	PBS	PBT	

Figure 3.6: AT&T ORCA FPGA [4]

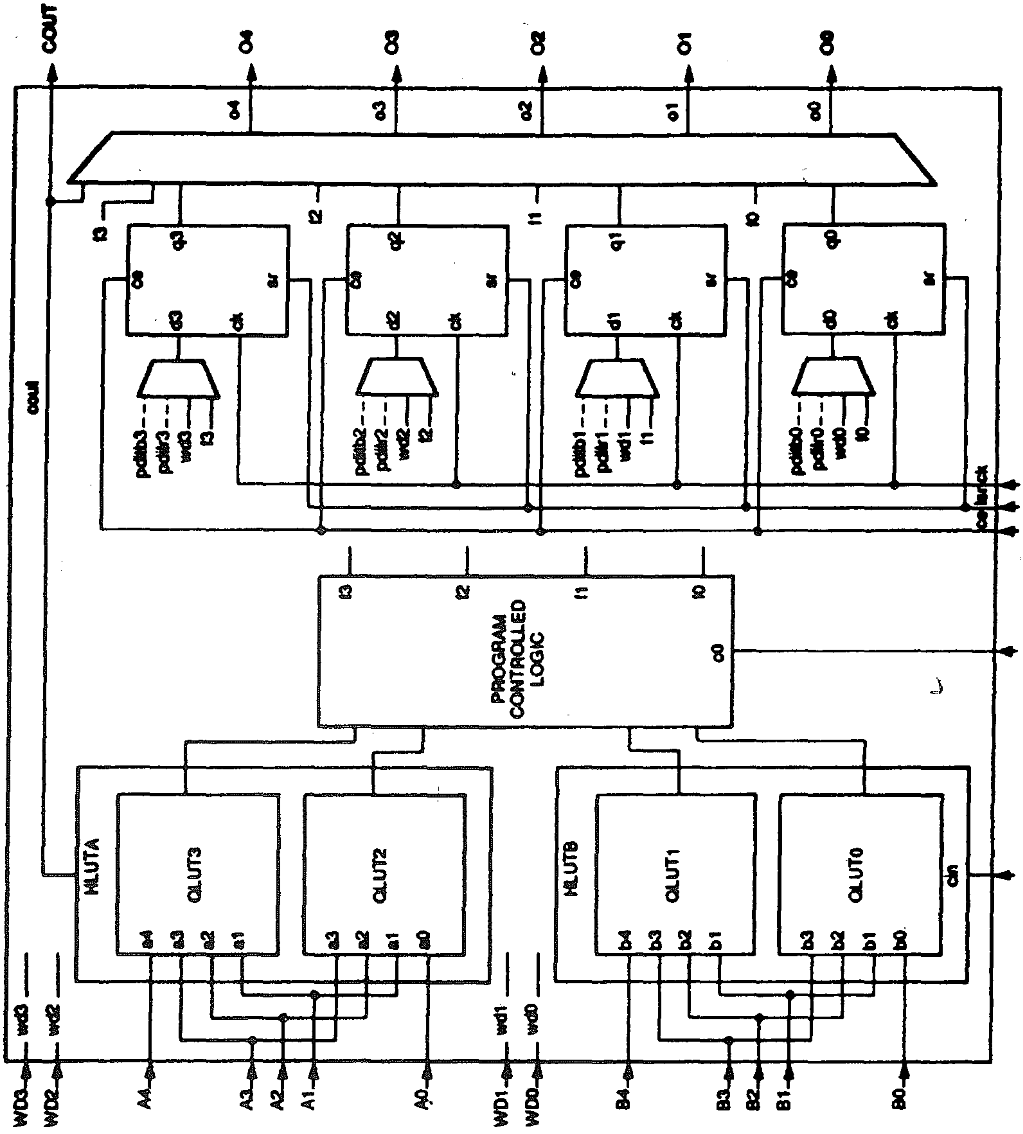


Figure 3.7: ORCA Programmable Logic Cell [4]

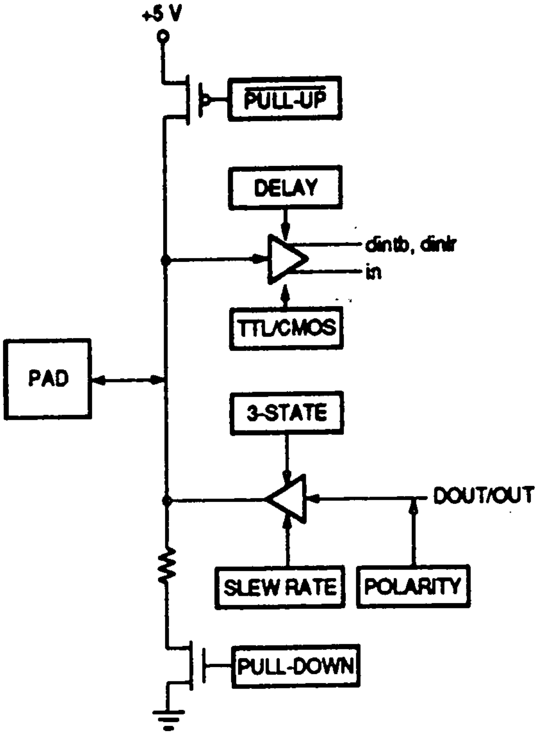


Figure 3.8: ORCA Programmable Input/Output Cell [4]

4.0 Simulation of Metastability

Although the information gathered through simulation may not absolutely predict real world devices, it does offer some advantages that are not always available in the real world. Simulation allows the probing of internal nodes which gives further insight into the metastable phenomena. Furthermore, various environmental effects can be easily studied such as processing parameters, power supply variation, and temperature differences. Also, simulation allows the circuit designer to evaluate several techniques to improve the resolving performance of the flip-flop, or to find a resolving problem before chip fabrication.

4.1 Previous Work

In the literature, there are three simulation methods used to determine metastability characteristics. All three have their advantages and disadvantages that will be further discussed in this section. The three methods are small signal AC analysis [23], forcing the flip-flop into a metastable state during transient analysis [20], and transient analysis to find the metastable window [31].

Small signal AC analysis is the least CPU intensive method. Through small signal AC analysis, the gain-bandwidth is found. An assumption is made that the metastable resolution constant (τ) is the inverse of the gain-bandwidth product. This assumption is only true when the flip-flop is constructed of simple inverters without feedback capacitors. Also, this method can only predict τ. It cannot determine the second metastability constant T_0, which is required to determine synchronizer failures.

A flip-flop can be forced into a metastable state, by setting the input node and output node to the same voltage near the middle of the voltage swing. If a transient analysis is performed while the flip-flop is in the metastable state, the output of the flip-flop will resolve to a stable logical state. Close to the metastable voltage, the exponential growth of the output voltage is the metastable resolution constant (τ). This procedure requires only one transient analysis, and therefore does not require extensive simulation time. Unfortunately, the constant τ must be determined from the linear region of the output voltage. If the linear region is not fully understood, the value of τ can be erroneous. Also, this method does not allow the prediction of T_0.

In section two, the metastability time window is discussed, and is shown graphically in figure 2.1. The metastability window is a range of input times that will produce flip-flop output delays greater than the normal propagation delay (T_p). This window can be fully described using many transient analyses. After describing the window, the constants τ and T_0 can be determined. This procedure requires extensive simulation time, but because of recent advances in computer performance, the whole procedure takes only a few hours. Also, since this procedure is straightforward, a program can be written to automate it.

4.2 Procedure

The method used in this thesis is the transient analysis to determine the metastable window. Before beginning a description of the procedure, a few parameters will be defined that will be used in the discussion. The time after t = zero, when the clock transitions from a low to a high, is the clock delay (T_{dc}). The time after t = zero, when the data transitions from a high to a low, is the data delay (T_{dd}). The difference between these two delays is the setup time for the flip-flop (T_{set}). The delay from the clock transition to the output (QN) transition is the propagation delay of the flip-flop (T_p). Within the metastable window, the setup time for the flip-flop affects the propagation delay. As T_{set} gets smaller and smaller, T_p gets longer and longer. There is a point where T_{set} is so small that the output no longer transitions. Define this point t = zero for the metastable window ($T_w(0)$).

First, find the point $T_w(0)$. Because the metastable window has an exponential growth, this point must be defined to a fine point of resolution. Here, $T_w(0)$ will be found to the femtosecond (1E-15 seconds). To do this continue to reduce the setup time until the output no longer changes state. The clock delay is held constant while the data delay is moved closer and closer to the clock edge. The data delay that causes $T_w(0)$ is typically within a time region of 2 nS around the clock edge. To find the value of $T_w(0)$ to the fS, would require two million simulations. Therefore, a search method is used to determine the point. This search method will be shown as an example using the worst-case conditions.

An example of the algorithm used for finding $T_w(0)$:

Perform a transient analysis at each value of T_{dd}, when T_{dd} varies from 5 nS to 7 nS by 100 pS The output does not change state at 6.1 nS.

Perform a transient analysis at each value of T_{dd}, when T_{dd} varies from 6.00 nS to 6.10 nS by 10 pS The output does not change state at 6.01 nS.

Perform a transient analysis at each value of T_{dd}, when T_{dd} varies from 6.000 nS to 6.010 nS by 1 pS The output does not change state at 6.007 nS.

Perform a transient analysis at each value of T_{dd}, when T_{dd} varies from 6.0060 nS to 6.0070 nS by 100 fS The output does not change state at 6.0061 nS.

Perform a transient analysis at each value of T_{dd}, when T_{dd} varies from 6.00600 nS to 6.00610 nS by 10 fS The output does not change state at 6.00610 nS.

Perform a transient analysis at each value of T_{dd}, when T_{dd} varies from 6.006090 nS to 6.006100 nS by 1 fS The output does not change state at 6.006097 nS.

The failure point is thus 6.006097 nS. Therefore, $T_w(0)$ is 6.006096 nS which is the last T_{dd} that causes the output to change state. The setup time at this point is T_{dc} - T_{dd} which is equal to a negative 6.096 pS.

Now that the point $T_w(0)$ is found, the rest of the metastability time window must be depicted. Again, since the window is exponential, ten simulations must be performed at each decade from one fS to one nS. Using the worst case processing as an example, the methodology will be described.

An example of the algorithm for depicting the metastable time window:

Perform a transient analysis at each value of T_{dd}, when T_{dd} varies from 6.006086 nS to 6.006096 nS by 1 fS Measure and store the value of T_p for each analysis.

Perform a transient analysis at each value of T_{dd}, when T_{dd} varies from 6.005996n to 6.006076n by 10fs Measure and store the value of T_p for each analysis.

Perform a transient analysis at each value of T_{dd}, when T_{dd} varies from 6.005096 nS to 6.005896 nS by 100 fS Measure and store the value of T_p for each analysis.

Perform a transient analysis at each value of T_{dd}, when T_{dd} varies from 5.996096 nS to 6.004096 nS by 1 pS Measure and store the value of T_p for each analysis.

Perform a transient analysis at each value of T_{dd}, when T_{dd} varies from 5.906096 nS to 5.986096 nS by 10 pS Measure and store the value of T_p for each analysis.

Perform a transient analysis at each value of T_{dd}, when T_{dd} varies from 5.006096 nS to 5.806096 nS by 100 pS Measure and store the value of T_p for each analysis.

After the data that depicts the metastability window has been collected, it must be graphed. The metastability time window is placed on the logarithmic-scale vertical axis, and the corresponding propagation delay is placed on the linear-scale horizontal axis. A straight line dependence is found with the slope determining the resolution time constant τ, and the intercept determining the constant T_0. Since the data does not all fall on a straight line, linear regression techniques are used to find τ and T_0. The equation for the line was given in section two, and is also listed in this section.

$$T_w(t) = T_0 * \exp(-t/\tau) \tag{4.1}$$

4.3 Results

Simulations were performed on three conditions: worst case fast (best processing, VDD = 5.5 VDC, T = -55 C), nominal (typical processing, VDD = 5.0 VDC, T = 25 C), and worst case slow (worst processing, VDD = 4.5 VDC, T = 125 C). The results are listed in table 4.1.

TABLE 4.1: Simulation Results

Process	Temp. (C)	VDD (V)	T_p (nS)	τ (pS)	T_0 (S)
WCF	-55	5.5	0.57	86	2.43E-11
TYP	25	5.0	1.01	155	3.92E-11
WCS	125	4.5	1.84	282	8.99E-11

Looking at the worst-case slow (WCS) results. As expected, the results for this condition were the worst. Figure 4.1 shows a simplified schematic of the AT&T ORCA flip-flop. The ORCA flip-flop is a D master-slave flip-flop designed in a 0.5 micron CMOS technology. Switches are created out of pass-transistors, and the latching is accomplished with cross-coupled inverters. The master-slave circuit actually consists of two D latches that operate on different clock polarities.

Figure 4.2 shows the normal operation of the flip-flop. The setup time at the D-input is 1nS. A close-up of the flip-flop output node (V_{QN}) and the first D latch output node (VG6) is shown in figure 4.3. The propagation delay for the flip-flop is 1.84 nS when the T_{set} = 1 nS.

Figure 4.4 shows the metastable operation of the flip-flop. The setup time at the D-input is -6.096 pS. Figure 4.5 shows a closer look at the important flip-flop nodes. Node VG6 sits at an anomalous state of 1.8 volts for 3 nS before resolving to a low state. The output node V_{QN} has a normal characteristics, but it switches to a low state 3 nS later than it would normally switch for a proper setup time. The propagation delay for the flip-flop is 4.87 nS when T_{set} = -6.096 pS.

Figure 4.6 shows the output V_{QN} with a 1 nS setup time (solid line), and the output V_{QN} with a -6:096 pS setup time (dotted line). Both outputs have similar characteristics, but with different propagation delays.

The graph of the metastable window T_w versus the flip-flop propagation delay is shown in figure 4.7. When T_w = 1 fS, the propagation delay is at its greatest value of 4.87 nS. When T_w = 1 nS, the propagation delay is at its lowest value of 1.84 nS. The graphical data forms a straight line from T_w = 1 fS to T_w = 1 pS. After 1 pS, the data becomes asymptotic near the normal value of propagation delay. This shape of the data matches the theoretical curve in Figure 2.1. From linear regression of the data, the value for the metastability constant τ is 282 pS, and the constant T_0 is equal to 8.99E-11 S. Also included is the graph for worst-case fast conditions (figure 4.8), and the graph for the nominal conditions (figure 4.9).

The results showed a large variation in the metastable resolution constant τ. For worst-case fast, τ is equal to 86 pS, while for worst-case slow, τ is equal to 282 pS. Because of the exponential growth of the mean time before (MTBF) equation, small differences in τ will cause order of magnitude changes in the MTBF error rate. As with any type of design methodology, the most pessimistic parameter should be used to design an effective synchronizer.

Using the example from section two, the MTBF can be calculated for an AT&T ORCA FPGA. The values of propagation delay and setup time from the AT&T ORCA FPGA Data Book [4] are utilized.

$$F_c = 10 \text{ MHz}$$

$$F_d = 1 \text{ MHz}$$

$$T_p = 3.0 \text{ nS}$$

$$T_{set} = 0.2 \text{ nS}$$

$$T_{route} = 1.0 \text{ nS}$$

$$\tau = 282 \text{ pS}$$

$$T_0 = 8.99\text{E-}11 \text{ S}$$

$$T_r = 100 - 3.0 - 0.2 - 1.0 = 95.8 \text{ nS}$$

$$\text{MTBF} = (\exp(95.8/0.282))*(1/(2*1\text{E+}07*1\text{E+}06*8.99\text{E-}11)) = 1.91\text{E+}144 \text{ S}$$

Clearly at this clock frequency the synchronizer would never fail. Yet if the clock frequency is increased to 100 MHz and the data frequency is increased to 10 MHz, there would be a failure every one and half hours. These error rates are calculated with the worst-case slow parameters. However if the worst-case fast parameters are used, there will be a failure every 1.27E+15 centuries! Obviously, the resolution constant has a large affect on the error rate.

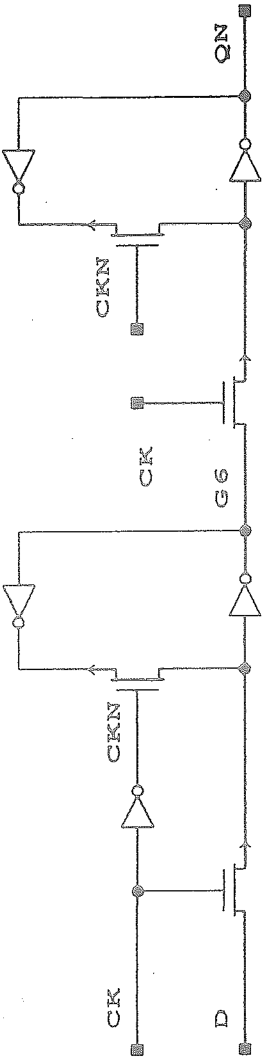


Figure 4.1: Simplified Schematic of the AT&T ORCA Flip-Flop

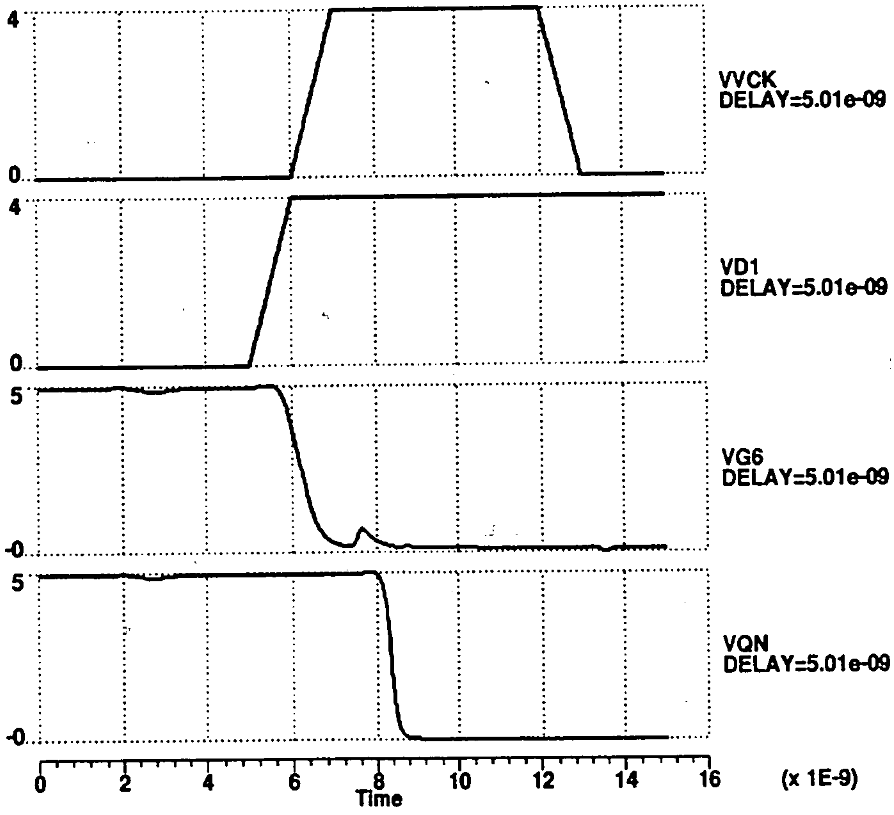


Figure 4.2: Normal Operation of the Flip-Flop

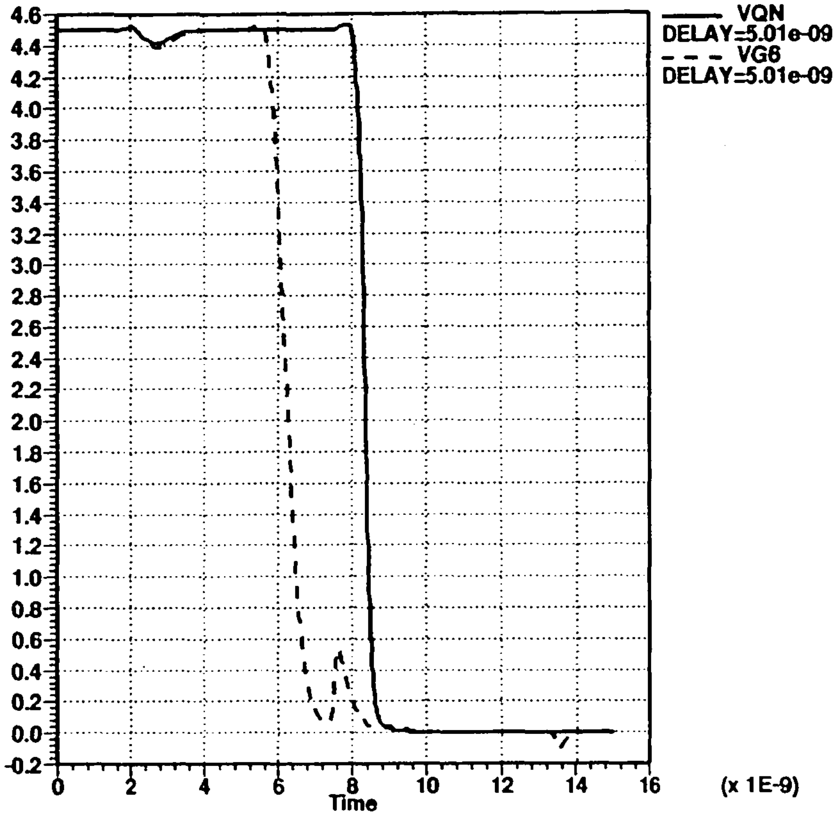


Figure 4.3: Close-Up of Normal Operation

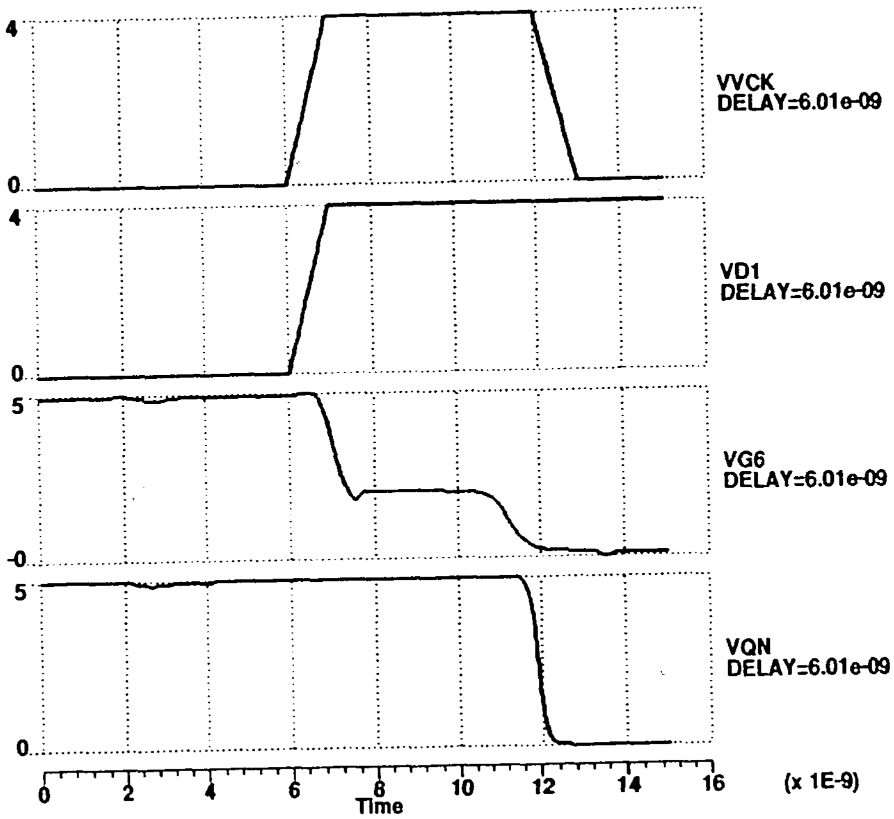


Figure 4.4: Metastable Operation of the Flip-Flop

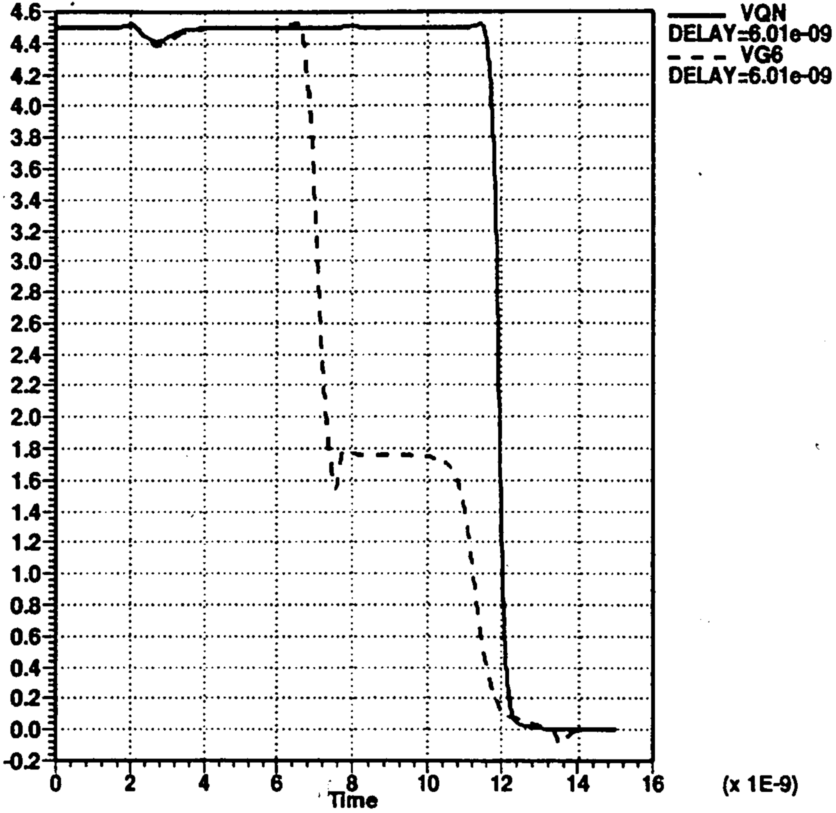


Figure 4.5: Close-Up of Metastable Operation

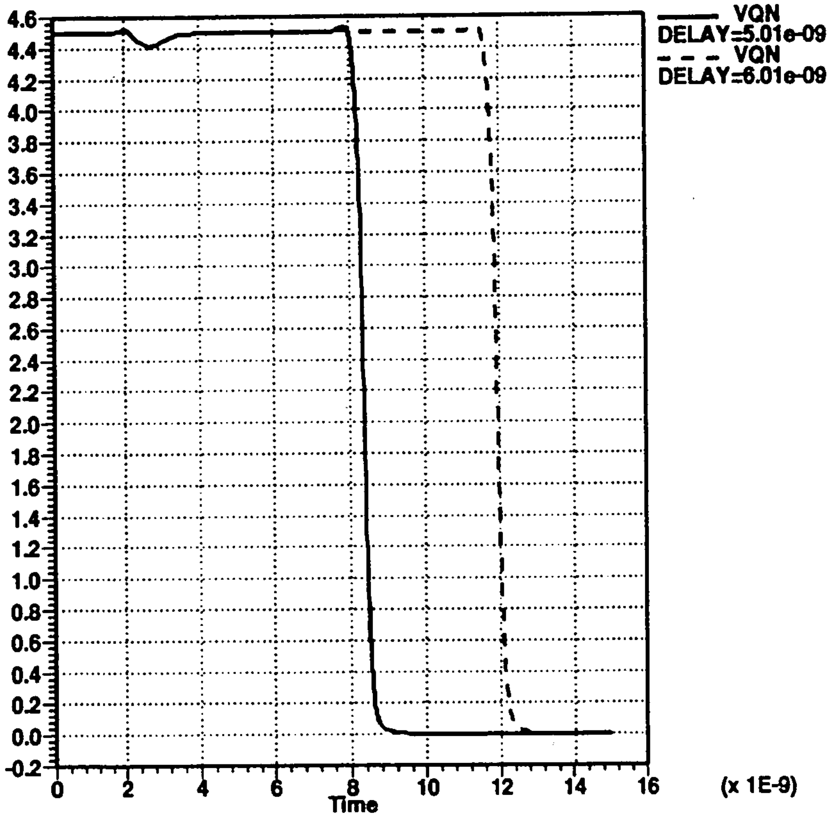


Figure 4.6: Comparison of Metastable Output and Normal Output

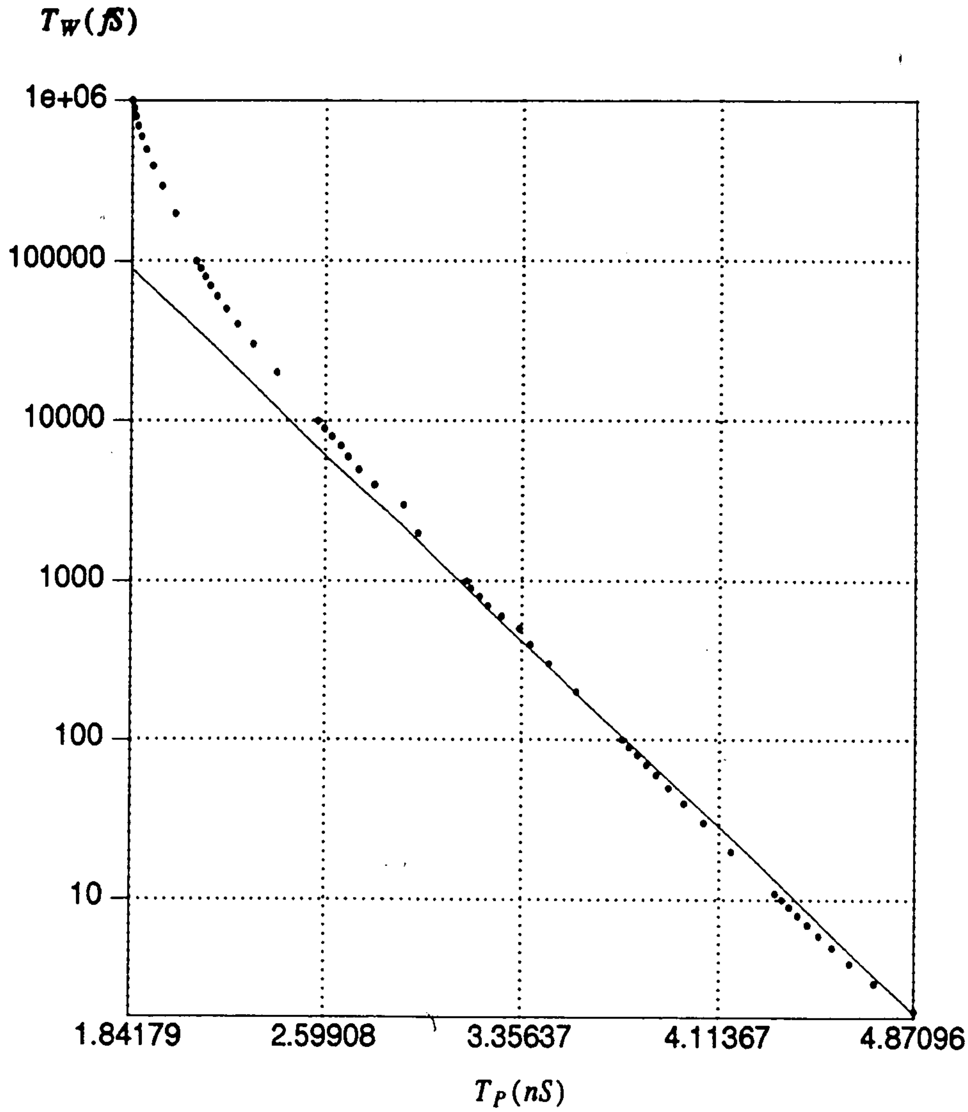


Figure 4.7: Graph of Metastable Window for Worst Case Slow Conditions

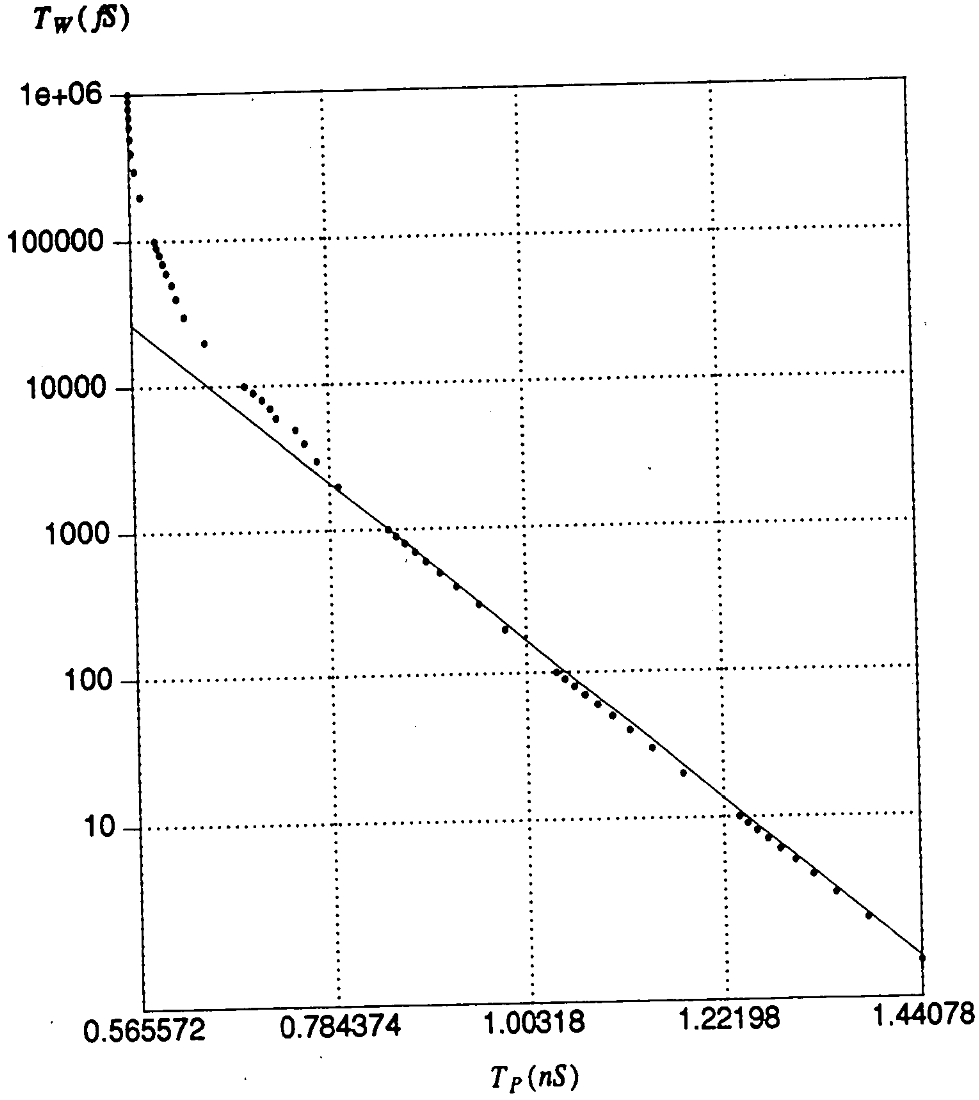


Figure 4.8: Graph of Metastable Window for Worst Case Fast Conditions

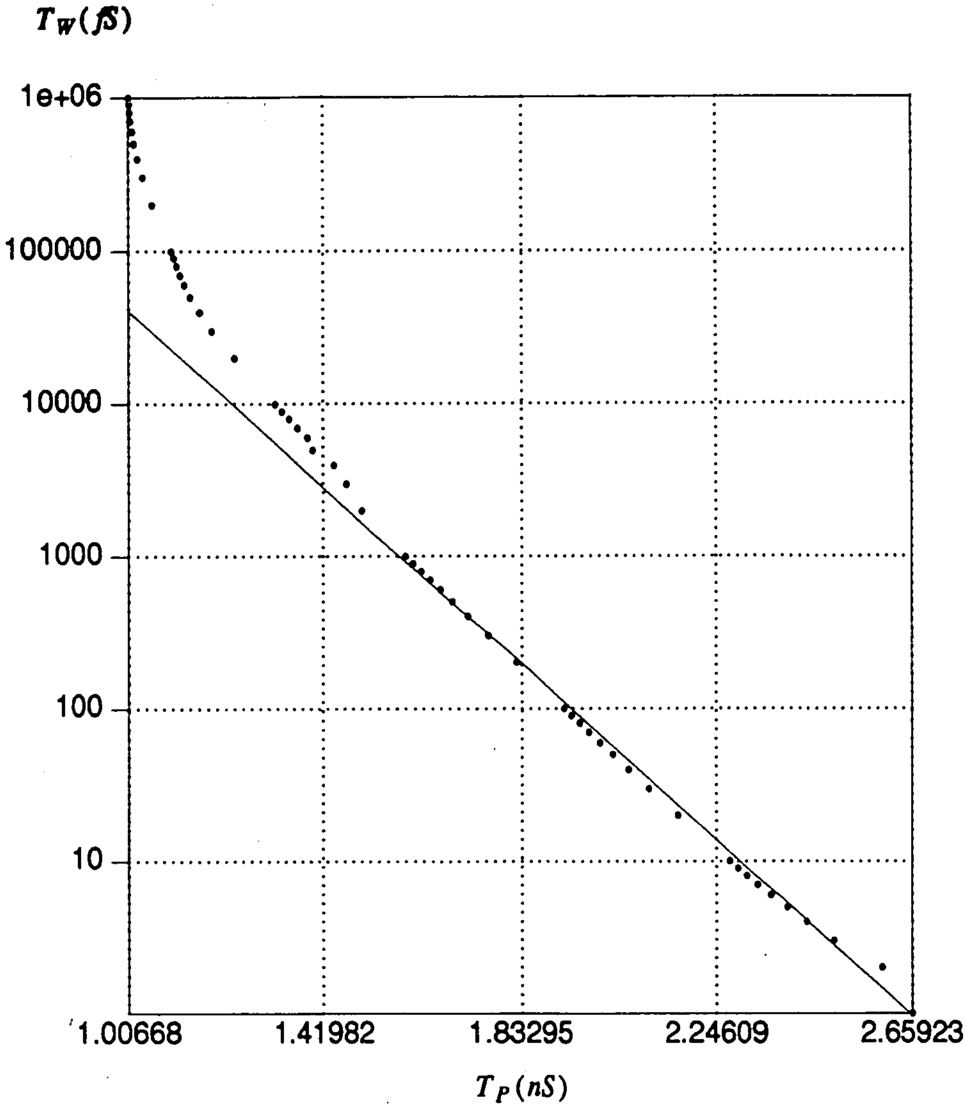


Figure 4.9: Graph of Metastable Window for Nominal Conditions

5.0 Testing of Metastability

In characterizing synchronizer failures, actual device testing is the only fail-safe method of obtaining real-world results. There have been many types of experimental methods for determining synchronizer failures. Three types used extensively in the literature will be discussed. The three types are the intermediate voltage sensor, the output proximity sensor, and the late transition sensor.

The intermediate voltage sensor [32] is an analog method of determining failures. In this methodology, two voltage comparators are used to determine if the output voltage lies between two set voltages (typically the noise margin of the device). When the output voltage remains within the two set voltages, the detection circuitry flags a metastable error. This methodology works well when detecting voltage levels in the threshold region of latch type flip-flops. Unfortunately, the flip-flops tested here exhibit increased delay, but normal output transitions. Also, because of the analog nature of this method, it is difficult to automate it.

The second methodology uses an output proximity sensor [43]. This sensor determines when the flip-flop outputs' Q and QN have approximately the same voltage. When the voltages at Q and QN are approximately equal, the sensor flags a metastable event. Similar to the first method, there is no way to detect errors for flip-flops with normal output transitions, but who have increased delays. It is also difficult to automate this procedure.

The first two methods sense voltage levels in the threshold region of latches. The flip-flops tested here have normal output transitions but exhibit increased propagation delays.

Therefore, the late transition sensor method is used [20]. In this method, a metastable state is not observed, but instead is inferred by a synchronizer failure. Figure 5.1 shows the late transition detector. The data input is asynchronous in relation to the clock signal. Since the data is asynchronous, it will violate the flip-flops setup time and hold time requirements. When these requirements are violated, the output of FF0, Q0, will become metastable. Output Q0 is sampled by flip-flop FF1 and flip-flop FF2. Flip-flop FF1 samples the output Q0 one clock cycle after FF0 has sampled the data input. Flip-flop FF2 samples the output Q0 one clock cycle after FF0 has sampled the data input, and after a set delay. If the output Q0 goes into a metastable state, and remains in that state for greater than one clock cycle, but less than one clock cycle and the set delay, then the outputs of flip-flop FF1 and FF2 will differ. When these outputs differ, the output of the XOR gate transitions from a zero state to a one state. This transition is a metastable error. These errors are counted over an interval of time, and are used to calculate the mean time before failure.

5.1 Test Circuit

A top level schematic for the test circuit is shown in figure 5.2. The metastable test circuit consists of four sections. The error section determines the errors. Counting the errors is accomplished by the counter section. A timer is used to control the time interval that errors are counted. The clock divider generates a 1 Hz signal which is used by the timer.

The error detector (figure 5.3) has two functions: determine the maximum frequency of operation for the circuit (FMAX), and detect metastable events. By holding the test input low, the circuit is in the FMAX mode. The maximum frequency of operation is determined by the critical path from flip-flop FF0 to flip-flop FF2. Flip-flop FF0 triggers on the positive edge of the clock while flip-flop FF2 triggers on the negative edge of the clock. Also, the output Q0 must pass through three levels of logic before arriving at the D input of flip-flop FF2. The maximum frequency for this circuit is given by

$$FMAX = 1/(2*(1/FC) - T_p - 3*T_d - T_{set}) \qquad (5.1)$$

where F_c is the clock frequency, T_p is the propagation delay of flip-flop FF0, T_d is the logic delay, and T_{set} is the setup time of flip-flop FF2.

The reason for having the three levels of logic in front of flip-flop FF2 is to reduce the maximum frequency of operation. Also, by triggering flip-flop FF2 on the negative edge of the clock, the maximum frequency of operation is also reduced. The maximum frequency of operation is reduced to approximately 50 MHz. At this frequency the internal counters can be used to count metastable events. Furthermore, by operating at this frequency, there is less concern about high frequency effects such as ringing and reflections.

When the test input is held high, the circuit is in the metastable detection mode. Here, the circuit is trying to detect metastable errors from flip-flop FF0. A clock input signal is provided on the CK input. An asynchronous data signal is provided on the D input. Since the input signal is asynchronous, the signal will violate the setup time or hold time of flip-flop FF0. When these violations occur, FF0's output Q0 will go into a metastable state delaying its transition. The output Q0 feeds the D input of FF1, and the D input of FF2 after going through three levels of logic. Flip-flops FF0 and FF1 trigger on the positive edge of the clock while flip-flops FF2 and FF3 trigger on the negative edge of the clock. If the flip-flop FF2 samples output Q0 before Q0 transitions, and flip-flop FF1 samples output Q0 after Q0 transitions, then there will be a difference in the output states of flip-flop FF1 and flip-flop FF2. A logical function produces an ERROR signal when output Q0 equals output Q1, but does not equal output Q2.

The error counter (figure 5.4) counts the number of metastable events. This section is made up of a 24-bit binary counter, and six 16X8 ROMs. The counter can tally up to 16,777,215 errors. Outputs of the counter feed the address inputs of the 16X8 ROMS. These ROMs act as decoders They decode the binary data from the counters to hexadecimal data for a liquid crystal display which is on the test board.

The counter is controlled by a timer circuit (figure 5.5). This timer circuit determines the time interval for counting metastable events. The timer is composed of an 8-bit counter and an 8-bit comparator. The counter counts from zero seconds to two hundred and fifty-five seconds. The counter output is compared with the settings of eight switches. When the counter's output is equal to the switch settings, the error counter is stopped. The switches are located on the test board. By modifying the switch settings, the time interval can be varied from zero seconds to two hundred and fifty-five seconds. The 8-bit counter is clocked by a 1 Hz signal.

The 1 Hz signal is generated by the clock divider circuit (figure 5.6). A 26-bit decade counter divides down a 4 MHz input signal. The 4 MHz input signal comes from an accurate clock oscillator which is located on the test board.

The test circuit incorporates all the pieces required for metastable testing: the detector, the counter, and the timer. Without the test circuit, test equipment would have been required to count the events, and to set the time interval. Therefore, this circuit reduces the amount of equipment required to do metastability testing.

5.2 Test Setup

The experimental setup (figure 5.7) includes various pieces of equipment. The clock signal is provided by an HP 8130A which is a 300 MHz pulse generator. An HP 8116A, 50 MHz pulse generator, provides the asynchronous data signal. There is no time reference between the two independent signal generators. To monitor the input and output signals, an HP 16500A digital scope is used. The power supply voltage is supplied by an HP 6102 power supply.

A four layer printed circuit board with two signal planes, a ground plane, and a power plane is used to perform the metastability measurements. On the board there is a 208-pin socket. This socket allows testing of all the AT&T ORCA FPGA's from the 1C03 (3500 gates) to the 2C26 (26,000 gates). All the inputs and outputs connect to the board with BNC connectors. To match the impedance of the coax cables which bring the input signals from the signal generators, the inputs are terminated on the board with a 50-ohm resistor. The metastable errors are shown on a 7-segment liquid crystal display. A clock oscillator is used to generate an accurate 4 MHz clock that is used as the reference input for the timer. The interval for the timer is set by a bank of dip-switches. A push-button switch is used to clear the metastable counter and the interval timer. Another push-button starts the loading of the configuration information. This configuration information comes from a Serial EEPROM which is also located on the test board.

5.3 Procedure

The testing procedure begins by configuring the Field Programmable Gate Array. As discussed in section 3, the AT&T ORCA FPGA is an SRAM based device. Therefore, before using the FPGA, it must be loaded with the design information. Here, the device is loaded with information from a 128K-bit serial EEPROM.

Once the device is configured, maximum operating frequency must be found. During this part of the testing, the test input to the error detect circuitry is held low. The D input for FF0 is now its own inverted output. Flip-flop FF0 now acts like a divide-by-two circuit with the output one half the frequency of the incoming clock signal. After placing the device into test mode, the output of flip-flop FF2 is monitored with the digital oscilloscope while increasing the clock frequency. When Q2 no longer changes state, the flip-flop is at its maximum operating frequency (FMAX), which is significant because this is the point where the time allowed for output Q0 to resolve to a known state is zero nanoseconds. To increase the resolving time (T_r), the clock period is increased.

The test mode is turned off. The asynchronous signal generator now drives the D input of flip-flop FF0. Starting at FMAX, clock period (Tc) is increased by 0.1 nS, and the data frequency (F_d) is increased by 0.1 MHz. By keeping the product of F_c and F_d a constant, a straight line relationship for the data is obtained. At each increment the total failures are measured over a sixty-second interval. T_c and F_d are increased until there are no failures during the time interval. This procedure is completed for five samples of the AT&T ORCA FPGA, and for the FPGA sample #3 across the power supply operating range of 4.5 VDC to 5.5 VDC.

The data gathered from the experimental procedure is in a hexadecimal format. It must be first converted to decimal. Also, the data was gathered in a sixty-second interval, but mean time before (MTBF) is in units of seconds. Therefore, the errors must be divided by sixty.

After the data has been tabulated, it must be graphed. The mean time before failure data is placed on the logarithmic-scale vertical axis, and the corresponding resolution time is placed on the linear-scale horizontal axis. A straight line dependence is found with the slope determining the resolution time constant τ, and the y-intercept of the line determining the constant B. Since not all the data falls onto a straight line, linear regression techniques are required to find τ and B. The equation for the line is

$$\mathrm{MTBF} = B*\exp(T_r/\tau) \tag{5.2}$$

where B is given by

$$B = 1/(2*F_c*F_d*T_0) \tag{5.3}$$

5.4 Results

The first step was to check the premise that the flip-flop would exhibit increased propagation delay when setup times and hold times are violated. To validate this premise, the flip-flop FF0's output Q0 was monitored with an oscilloscope. Sample #1 with VDD = 4.5 VDC is used. The clock period is set to 19 nS, and the data frequency is set to 19 MHz. Figure 5.8 shows a scope plot of the data signal and the clock signal with the scope in normal mode. In the accumulate mode, the waveforms show that the data has no timing relationship to the clock. Figure 5.9 is the scope plot with the scope in the accumulate mode for thirty seconds. The data waveform moves with each capture of the scope.

The output Q0 of the flip-flop FF0 was monitored for 24 hours with the scope in the accumulate mode. Figure 5.10 shows the scope plot for Q0. From the plot, it can be observed that the propagation delay of Q0 increased by 1.06 nS, when Q0 transitioned from a low to a high. This increased delay is related to setup and hold violations caused by the asynchronous data input. Another point of interest is that the propagation delay of Q0 only increased by 500 pS, when Q0 transitioned from a high to a low. This leads to the theory that the flip-flop may have better characteristics when the output is going low.

After the premise was validated, the metastability characteristics of five samples were tested. The power supply voltage (VDD) was set to 5.0 volts DC, and the ambient temperature was at 25 degree Celsius. The results are listed in table 5.1.

TABLE 5.1: Experimental Results for Five Samples

Sample	FMAX (MHz)	τ (pS)	T_0 (S)
1	51.7	97	7.52E-11
2	55.2	205	7.94E-12
3	54.6	150	2.98E-11
4	48.1	112	1.33E-11
5	52.1	122	4.81E-11
Average	52.3	137	3.49E-11

For sample #3, figure 5.11 shows the graph of the mean time before failure (MTBF) versus the resolution time (T_r). The measured data points fall close to the line which is given by equation (5.2). Only at $T_r = 0.1$ nS, does that data diverge from the line. This divergence may be caused by different characteristics in this region, or possible FMAX failures. From a linear regression of the data the metastability constant τ is equal to 150 pS, and the constant B is equal to 1.68E-05 S. Since the product of the F_c and F_d equaled 1E+15 S, T_0 can be calculated using equation (5.3). The constant T_0 is equal to 29.8 pS.

Looking at the data in table 5.1, there is some interesting information. First, there is a wide variation in the metastable resolution constant τ. Sample #1 has the lowest value of τ (97 pS), while sample #2 has the highest value of τ (205 ps). As was discussed previously, small variations in τ will cause order of magnitude differences in the mean time before failure. Second, it would be expected that the fast devices would have the lowest τ, but devices #2 and #3 have the highest FMAX and the highest τ. Devices #1 and #4 have the lowest FMAX and the lowest τ. It seems that the process parameters which improve delay may increase the resolution constant.

Sample #3 was then tested to find the effects of power supply variation. The device was tested at VDD = 4.5 VDC, VDD = 5.0 VDC, and VDD = 5.5 VDC. Table 5.2 and figure 5.12 show the results of the tests.

TABLE 5.2: Experimental Results Over Power Supply Range

VDD (V)	**FMAX (MHz)**	**τ (pS)**	**T_0 (S)**
4.50	50.8	179	5.14E-11
5.00	54.6	150	2.98E-11
5.50	56.2	94	4.03E-11

The results for power supply variation were as expected, with the lowest power supply voltage causing the highest value of τ, and the highest power supply voltage having the lowest value for τ. However, the variation in τ did not change linearly with the power supply variation. In fact, the value of τ for VDD = 5.5 VDC was much better than the value of τ for VDD = 5.0 VDC. However, the value of τ for VDD = 4.5 VDC was not much worse than the value of τ for VDD = 5.0 VDC.

The average results for the five samples were 137 pS for τ, and 34.9 pS for T_0. These values correlate well with the nominal results from simulation (τ = 155 pS and T_0 = 39.2 pS).

Based on these values, figure 5.13 is a graph comparing MTBF versus T_r for both simulation and experimental results. Since there is a good correlation between experimental results and simulation results, simulation could be utilized to predict the effects of temperature and process variation.

Metastable failures can be catastrophic in systems. Therefore, when calculating failures, the most conservative numbers for the metastability constants should be used. Using the values from sample #2, the example from section 4 is repeated.

$$F_c = 100 \text{ MHz}$$
$$F_d = 10 \text{ MHz}$$
$$T_p = 3.0 \text{ nS}$$
$$T_{set} = 0.2 \text{ nS}$$
$$T_{route} = 1.0 \text{ nS}$$
$$\tau = 205 \text{ pS}$$
$$T_0 = 7.94 \text{ pS}$$

$$T_r = 10 - 3.0 - 0.2 - 1.0 = 5.8 \text{ nS}$$

$$\text{MTBF} = (\exp(5.8/0.205))*(1/(2*1E+08*1E+07*7.94E-12)) = 1.22E+08 \text{ S}$$

The synchronizer would fail every 1.22E+08 seconds or 3.87 years. For a thousand systems, there would be a synchronizer failure every 1.42 days.

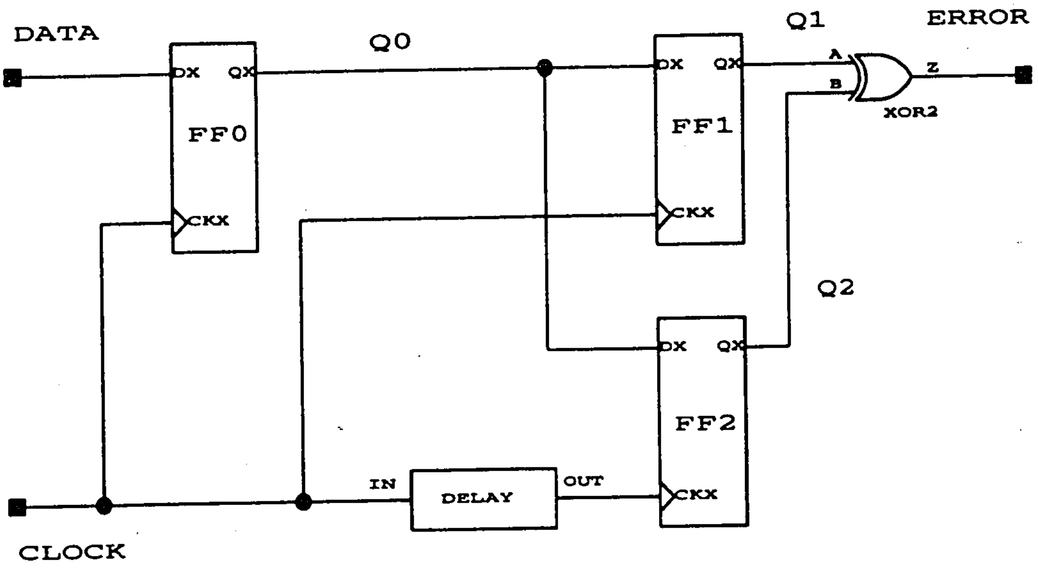


Figure 5.1: Late Transition Detector

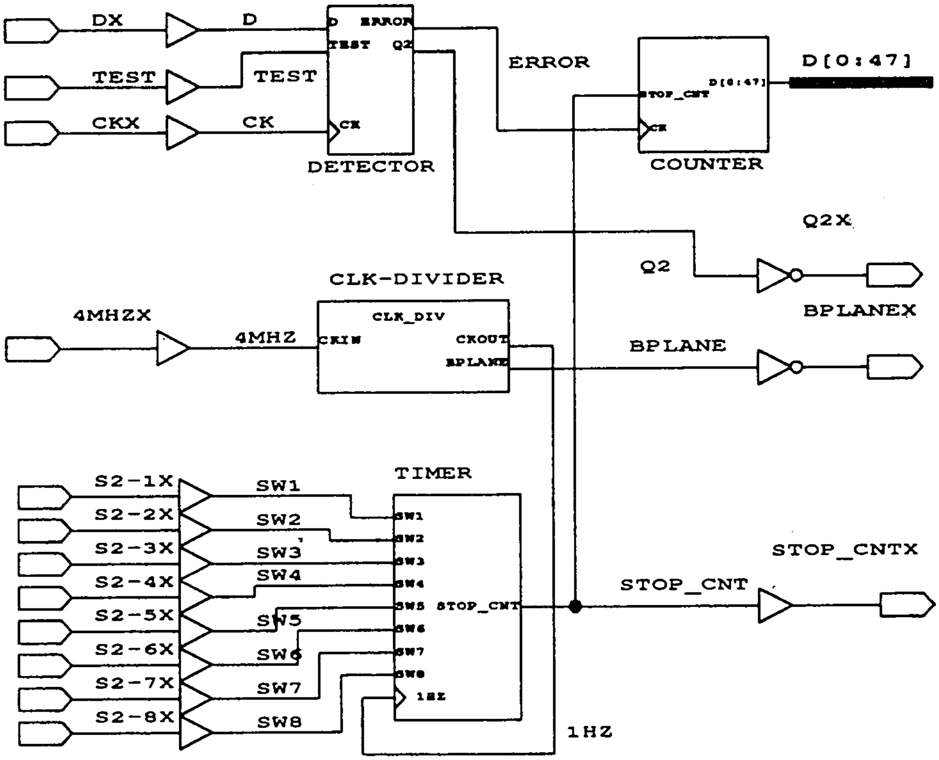


Figure 5.2: Top Level Schematic of Test Circuit

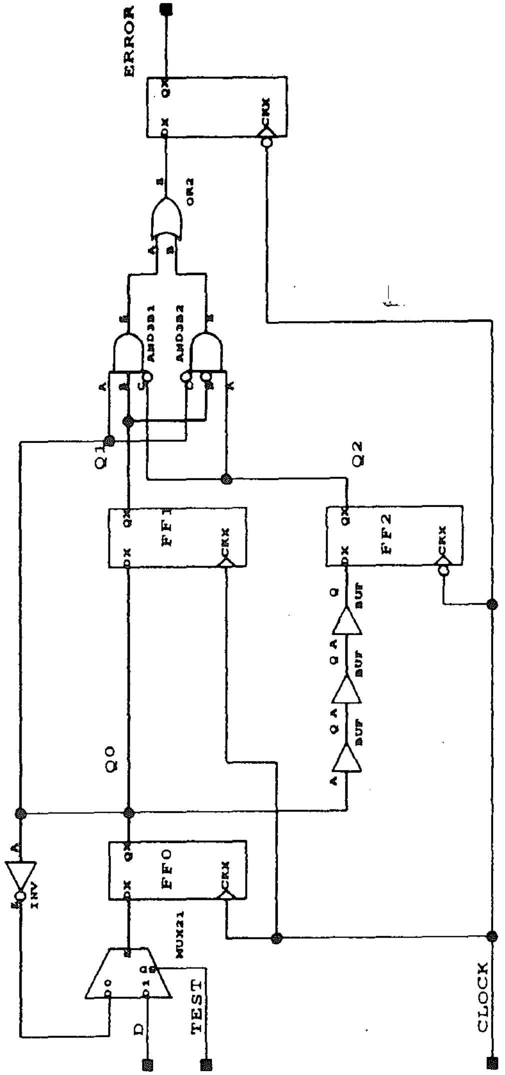


Figure 5.3: Schematic of Error Detector

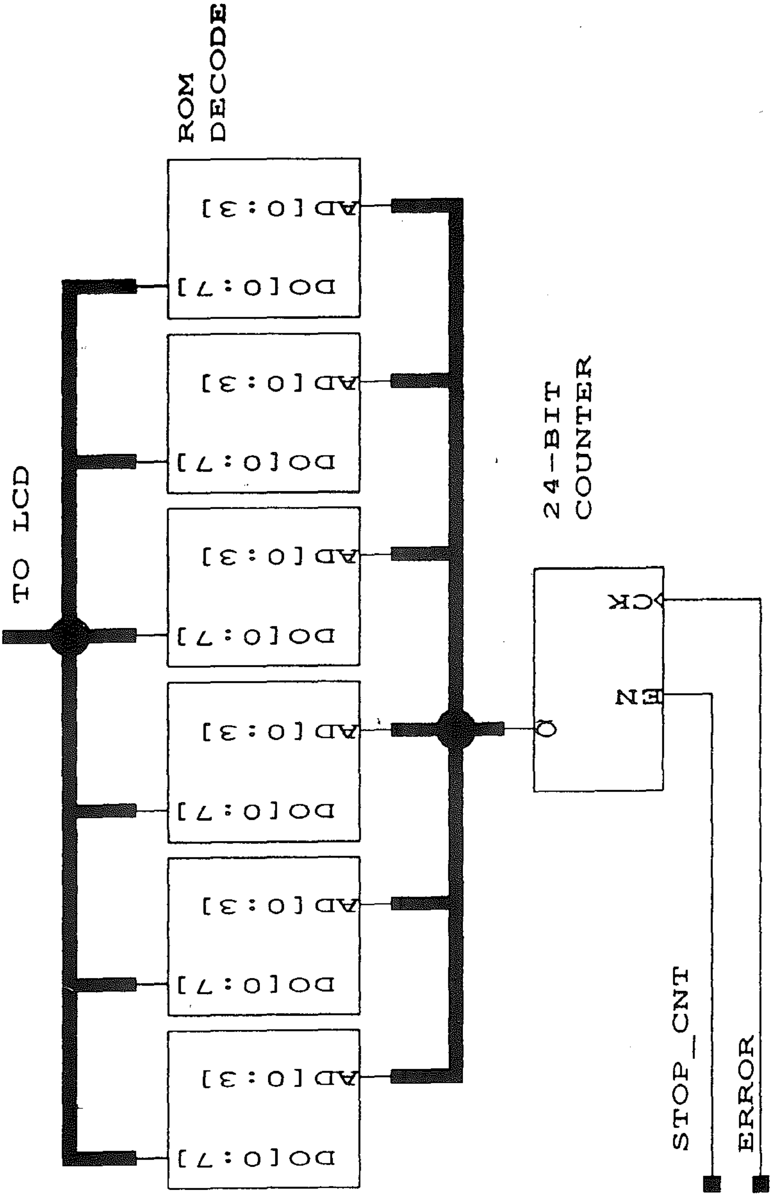


Figure 5.4: Schematic of Error Counter

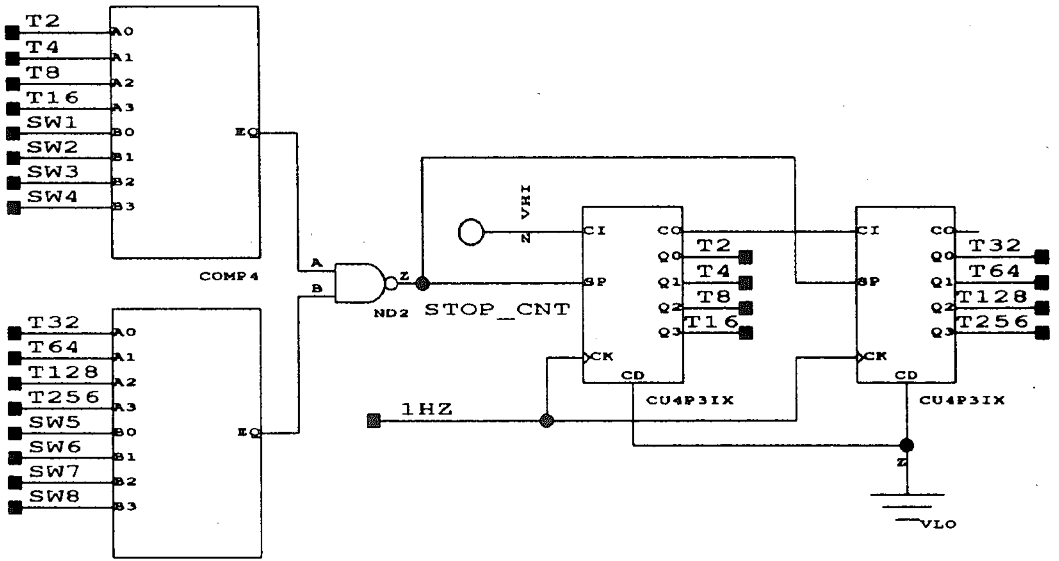


Figure 5.5: Schematic of Timer

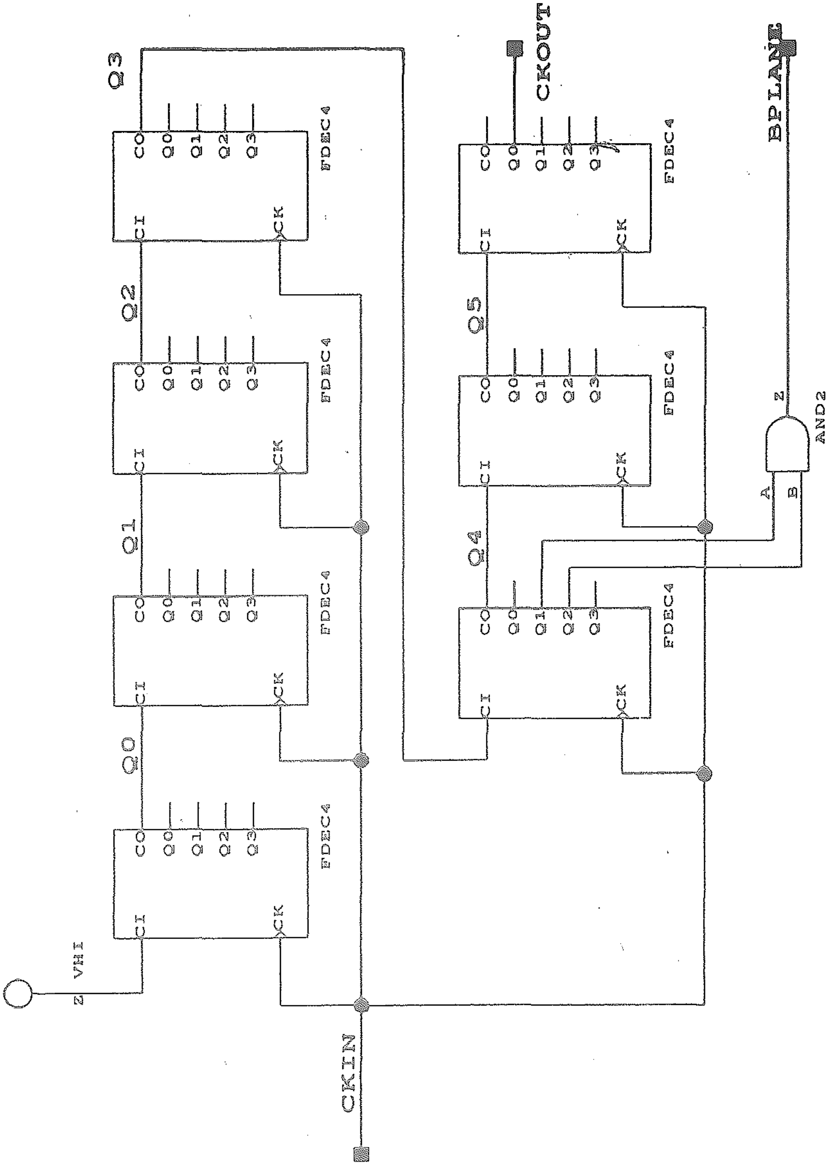


Figure 5.6: Schematic of Clock Divider

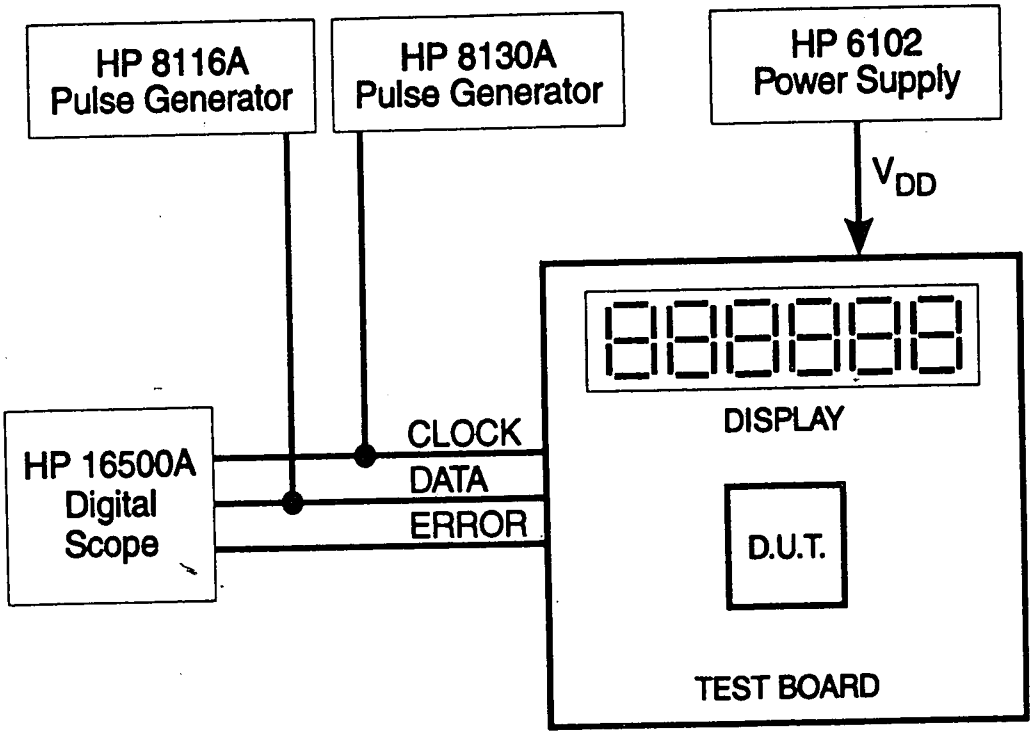
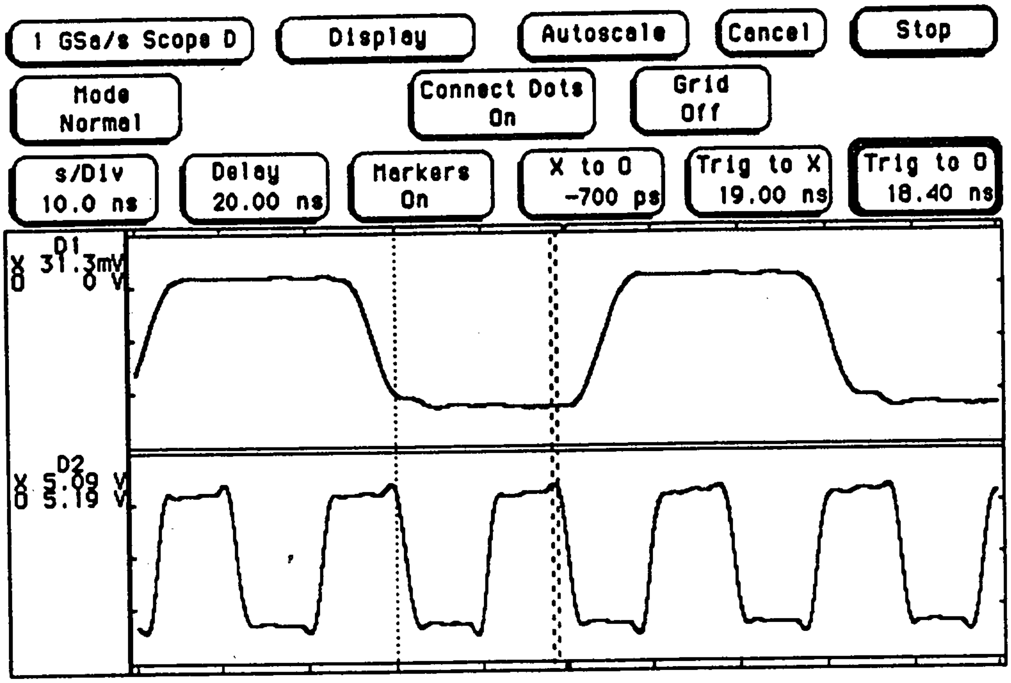


Figure 5.7: Metastability Test Setup



Mode
Normal

Connect Dots
On

Grid
Off

s/Div
10.0 ns

Delay
20.00 ns

Markers
On

X to O
-700 ps

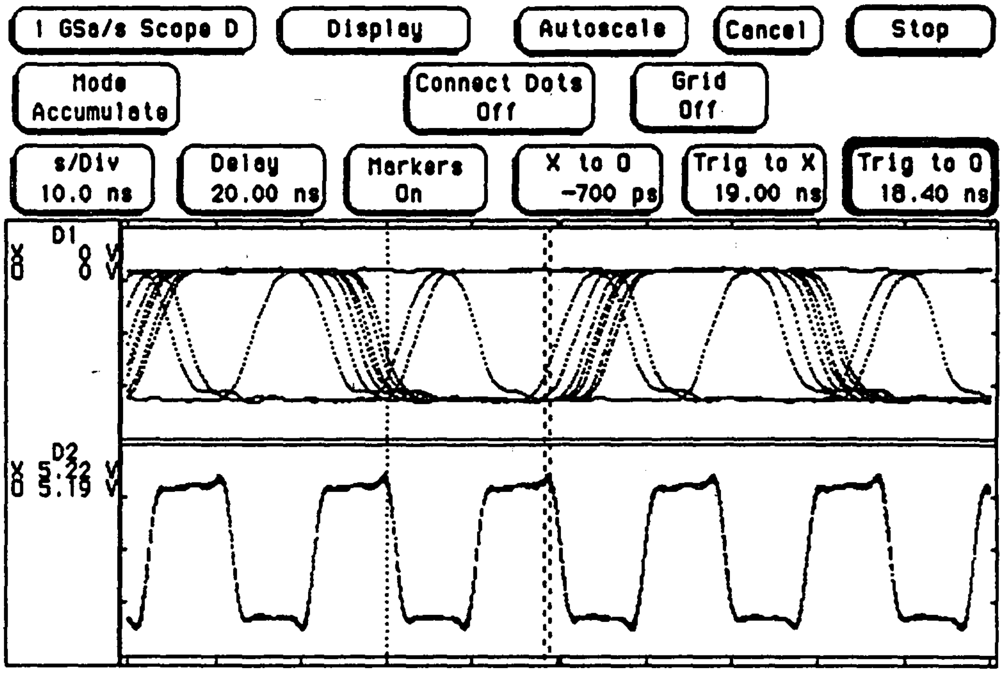
Trig to X
19.00 ns

Trig to O
18.40 ns

D1
X 31.3mV
O 0 V

D2
X 5.09 V
O 5.19 V

Figure 5.8: Scope Plot of Data and Clock with Scope in Normal Mode



Mode Accumulate

Connect Dots Off

Grid Off

s/Div 10.0 ns

Delay 20.00 ns

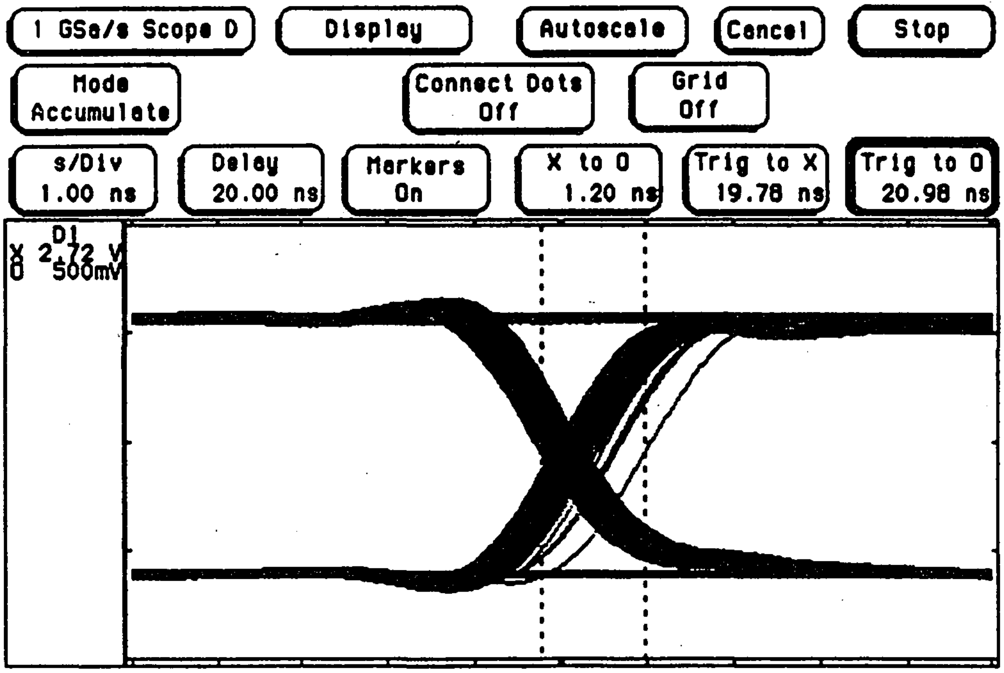
Markers On

X to O -700 ps

Trig to X 19.00 ns

Trig to O 18.40 ns

Figure 5.9: Scope Plot of Data and Clock with Scope in Accumulate Mode



Mode Accumulate

Connect Dots Off

Grid Off

s/Div 1.00 ns

Delay 20.00 ns

Markers On

X to O 1.20 ns

Trig to X 19.78 ns

Trig to O 20.98 ns

Figure 5.10: Scope Plot of Output with Scope in Accumulate Mode for 24 Hours

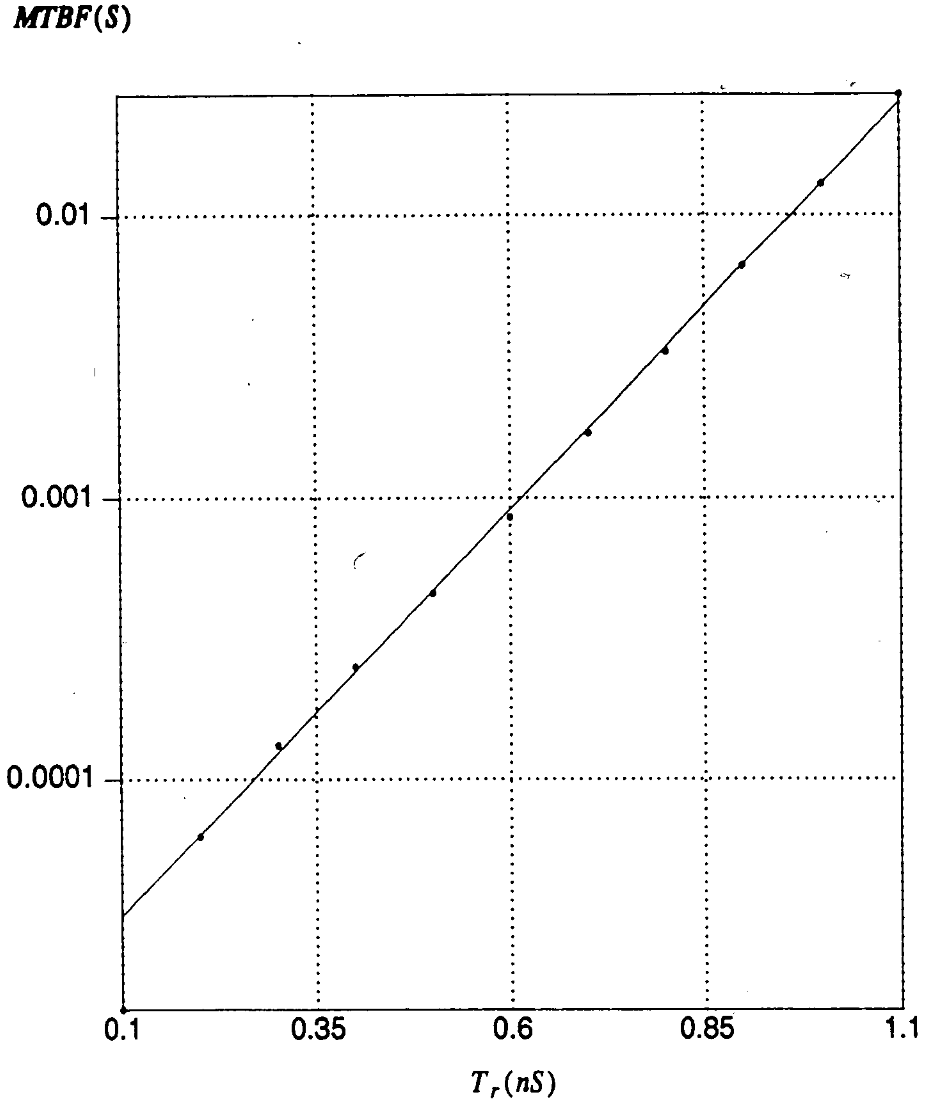


Figure 5.11: Graph of MTBF Versus T_r for Device Sample #3

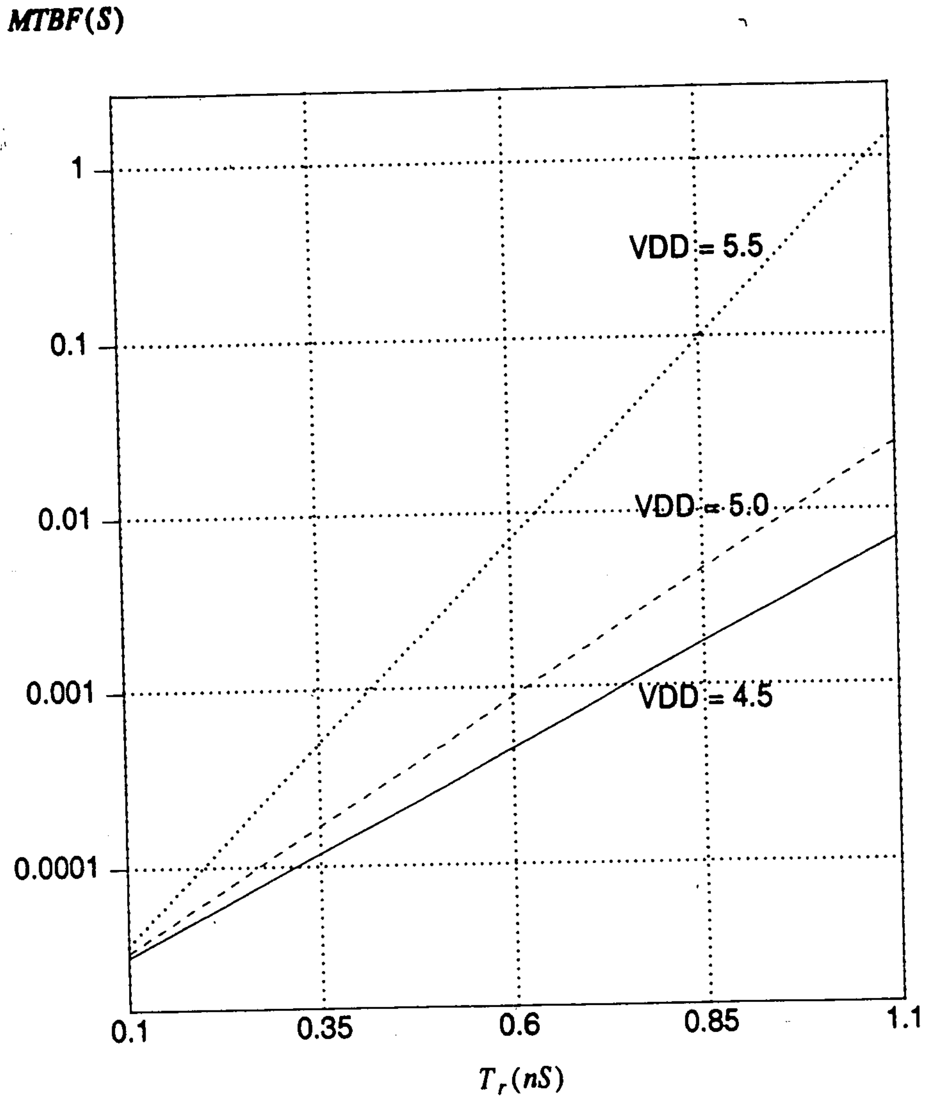


Figure 5.12: Graph of MTBF Versus T_r for VDD = 4.5 V, 5.0 V, and 5.5 V

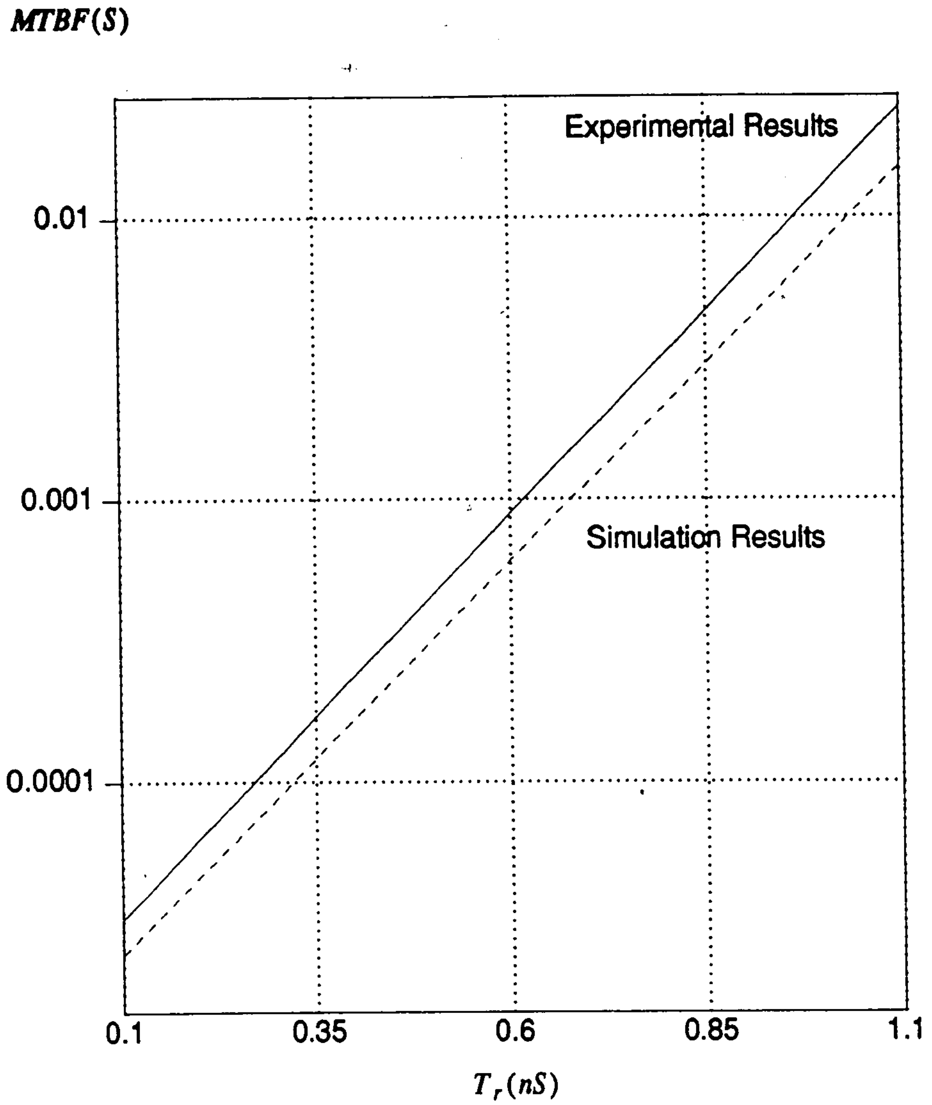


Figure 5.13: Graph of MTBF Versus T_r for Experimental and Simulation Results

6.0 Improving Metastability

Metastability is an unavoidable problem that occurs when an asynchronous signal must be synchronized. Designers have tried special circuits to eliminate this problem [24]. Unfortunately, these circuits did not reduce the problem, but instead increased its likelihood. Since metastability cannot be avoided, we must strive to reduce the probability of it occurring. This section will describe the techniques that should be used to reduce metastable errors.

6.1 Avoid unnecessary synchronization events.

Asynchronous signals should have only one input point to the system. At this point, the input should be synchronized, before being used by the rest of the system. Also, all internal design techniques should be synchronous. Asynchronous practices such as clock-gating are sensitive to temperature, power supply, and process variations. Small changes in these parameters can cause metastability.

6.2 Use the fastest parts available.

Most standard products are available in different speed grades. The manufacturer selects the fastest parts of the normal process distribution. These parts are used for applications that need the highest performance. The fastest speed grade with a reasonable yield is typically about 40 percent faster than the worst case specification. AT&T ORCA FPGAs have two speed grades. The -3 speed is the highest performance part available. It is 25 percent faster than the -2 speed grade part. The -3 speed grade refers to nominal processing parameters, and the -2 speed grade refers to slow processing parameters.

Resolving time (T_r) is a function of the clock period (T_c), the flip-flop delay (T_p), the flip-flop setup time (T_{set}), the logic delay between the flip-flops (T_d), and the routing delay between the flip-flops (T_{route}). By using the fastest part available, the flip-flop's delay and the flip-flop's setup time are reduced. Thereby, reducing the resolution time. Since the ratio of T_r to τ is an exponential function, changing to a faster speed grade reduces the error rate dramatically, as seen from the equation

$$\mathrm{MTBF} = (\exp(T_r/\tau))*(1/(2*T_0*F_c*F_d)) \tag{6.1}$$

where

$$T_r = T_c - T_p - T_{set} - T_d - T_{route} \tag{6.2}$$

To illustrate the effect of speed grades, the example from section five will be used for a -2 speed grade device. Utilizing equation (6.1), equation (6.2), and the following parameters, solve for the MTBF

$$F_c = 100 \text{ MHz}$$
$$F_d = 10 \text{ MHz}$$
$$T_p = 3.9 \text{ nS}$$
$$T_{set} = 0.5 \text{ nS}$$
$$T_{route} = 1.0 \text{ nS}$$
$$\tau = 205 \text{ pS}$$
$$T_0 = 7.94 \text{ pS}$$

$$T_r = 10 - 3.9 - 0.5 - 1.0 = 4.6 \text{ nS}$$

MTBF = (exp(4.6/0.205))*(1/(2*1E+08*1E+07*7.94E-12)) = 3.50E+05 S

The synchronizer will fail every 3.50E+05 seconds. From section five, the failure rate was 1.22E+08 seconds for a -3 speed grade. By changing from a -3 device to a -2 device, the error rate has worsened by three orders of magnitude. This error rate translates into a decrease from 3.87 years to 4.05 days.

6.3 Reduce the routing delay.

With Field Programmable Gate Arrays, there are routing delays between logic elements. This routing delay can constitute a significant part of the total delay, varying from less than a nanosecond to greater than twenty nanoseconds. Part of the resolution time is due to this routing delay. To reduce this delay, the designer must specify the maximum delay on the route connecting the two flip-flops when placing and routing the FPGA. The delay can be checked by performing back-annotation simulation on the design, or by checking the design's static timing information.

6.4 Avoid having a logic function between the synchronizer flip-flops.

The delay through logic functions will decrease the time for the output to resolve to a proper state. By decreasing the resolution time, the MTBF decreases. Unfortunately this cannot always be avoided, since in many devices the input signal must pass through logic before getting to the flip-flop. This additional delay is detrimental to a synchronizers ability to reduce errors.

6.5 Reduce the setup time for the flip-flop.

Setup time is a portion of the resolution time. Therefore decreasing the setup time will increase the mean time before failure. Some devices have direct inputs to their flip-flops, which can significantly reduce the flip-flop setup time. The AT&T ORCA FPGA allows direct inputs to it's flip-flops, and by using these direct inputs the flip-flop setup time is reduced from 1.8 nS to 0.2 nS. This difference of 1.6 nS has an exponential affect on the MTBF. Utilizing equation (6.1), equation (6.2), and the following parameters, solve for the MTBF

$$F_c = 100 \text{ MHz}$$
$$F_d = 10 \text{ MHz}$$
$$T_p = 3.0\text{nS}$$
$$T_{set} = 1.8\text{nS}$$
$$T_{route} = 1.0 \text{ nS}$$
$$\tau = 205 \text{ pS}$$
$$T_0 = 7.94 \text{ pS}$$

$$T_r = 10 - 3.0 - 1.8 - 1.0 = 4.2\text{nS}$$

$$\text{MTBF} = (\exp(4.2/0.205))*(1/(2*1\text{E}{+}08*1\text{E}{+}07*7.94\text{E}{-}12)) = 4.98\text{E}{+}04 \text{ S}$$

With a setup time of 1.8 nS, the synchronizer will fail every 4.98E+04 seconds (13.8 hours). From section five, the synchronizer will fail every 1.22E+08 seconds (3.87 years) with a setup time of 0.2 nS. Therefore, the mean time before failure has declined by four orders of magnitude by decreasing the setup time by 1.6 nS.

6.6 Increase the clock period.

The MTBF equation (6.1) is directly proportional to the clock period. Also, the clock period constitutes a major portion of the resolution time. By increasing the clock period, T_r increases, causing an exponential increase in MTBF. Unfortunately, the clock period is a function of the specific system, and it may be impossible to increase the clock period.

6.7 Increase the slew-rate of the input signal.

In section two, the metastability time window was calculated. This calculation showed that the value of the window is proportional to the slew rate. Therefore, the faster the input signal passes through the window, the longer the time before a synchronizer failure.

6.8 Increase the number of clock cycle.

By increasing the number of clock cycles, a metastable output has effectively a resolution time equal to the number of clock cycles times the clock period. This increase in the resolution time reduces the probability of errors by approximately the exp(N) where N is equal to the number of clock cycles. There are two design methods that increase clock cycles. These two methods are the cascaded synchronizer and the multiple cycle synchronizer.

6.8.1 Use a cascaded synchronizer.

The cascaded synchronizer uses a shift register of flip-flops instead of one flip-flop (figure 6.1). The assumption is that the first flip-flop in the shift register will resolve the metastable state. If the metastable state is not resolved, then each successive flip-flop with equal probability will attempt to resolve the state. The overall probability of this synchronizer is of the Nth power of the failure probability of a single flip-flop synchronizer. The mean time before failure equation for the cascaded synchronizer is given by

$$\mathrm{MTBF} = ((\exp(T_r/\tau))*(1/(2*T_0*F_c*F_d))\text{^}N \qquad (6.3)$$

where

$$T_r = T_c - T_p - T_{set} - T_d - T_{route} \qquad (6.4)$$

6.8.2 Use a multiple cycle synchronizer.

The multiple cycle synchronizer uses a divided down system clock (figure 6.2). Here, the system clock is divided down by a divide-by-N counter where N is the number of states. The counter output feeds the clock input of the synchronizer flip-flops. These flip-flops have an effective clock period of N times the system clock period. By increasing the clock period, the time allowed to resolve from a metastable state is also increased. With increased resolution time, the mean time before failure also increases. The MTBF equation for the multiple cycle synchronizer is given by

$$\mathrm{MTBF} = ((\exp(T_r/\tau))*(1/(2*T_0*F_c*F_d)) \qquad (6.5)$$

where

$$T_r = N*T_c - T_p - T_{set} - T_d - T_{route} \qquad (6.6)$$

The multiple cycle synchronizer will have a higher MTBF than the cascaded synchronizer. The difference in MTBF is due to the multiple cycle synchronizer having a larger resolution time for the same value of N, as can be seen by comparing equations (6.4) and (6.6). In equation (6.6) the flip-flop delay, the flip-flop setup time, the logic delay, and the routing delay are subtracted from the effective clock period only once. In equation (6.4) these delays must be subtracted N times.

To illustrate the difference in MTBF between the cascaded synchronizer and the multiple cycle synchronizer, the example from section four will be used. For the cascaded synchronizer with N equal to two, the MTBF can be calculated using equations (6.3) and (6.4).

$$F_c = 100\text{ MHz}$$
$$F_d = 10\text{ MHz}$$
$$T_p = 3.0\text{ nS}$$
$$T_{set} = 0.2\text{ nS}$$
$$T_{route} = 1.0\text{ nS}$$
$$\tau = 205\text{ pS}$$
$$T_0 = 7.94\text{ pS}$$
$$N = 2$$

$$T_r = 10 - 3.0 - 0.2 - 1.0 = 5.8\text{ nS}$$

MTBF = ((exp(5.8/0.205))*(1/(2*1E+08*1E+07*7.94E-12)))^2 = 1.49E+16 S

The synchronizer will fail every 1.49E+16 seconds or 4.72E+08 years. For the multiple cycle synchronizer again with N equal to two, the MTBF can be calculated using equations (6.5) and (6.6).

$$F_c = 50 \text{ MHz}$$

$$F_d = 10 \text{ MHz}$$

$$T_p = 3.0 \text{ nS}$$

$$T_{set} = 0.2 \text{ nS}$$

$$T_{route} = 1.0 \text{ nS}$$

$$\tau = 205 \text{ pS}$$

$$T_0 = 7.94 \text{ pS}$$

$$N = 2$$

$$T_r = 20 - 3.0 - 0.2 - 1.0 = 15.8 \text{ nS}$$

MTBF = ((exp(15.8/0.205))*(1/(2*5E+07*1E+07*7.94E-12)) = 3.74E+29 S

The multiple cycle synchronizer will fail every 3.74E+29 seconds or 1.19E+22 years. At this frequency the multiple cycle synchronizer has an MTBF that is fourteen orders of magnitude better than the cascaded synchronizer.

Although the cascaded method and the multiple cycle method significantly reduce metastable errors, they cannot be used in all systems. The larger the value of N, the longer it takes for an asynchronous input change to be seen by the synchronous system. Fortunately in typical microprocessor systems, most asynchronous inputs are events, interrupts, or DMA requests. These inputs do not have to be recognized at the system frequency. However in memory access, it is important that the entire system memory be synchronous with the system clock. For example, the refresh requests of dynamic memory must be acknowledged within one clock period. A simple flip-flop must be used to synchronize this input.

Since the multiple cycle synchronizer samples data at a clock frequency of N times the systems clock frequency, it may miss data changes that the cascaded synchronizer would have captured. Another disadvantage is that the divide-by-N counter injects skew into the clock signal used by the multiple cycle synchronizer. This clock skew limits the use of the synchronizer in high frequency applications.

Using the techniques described in this chapter, a designer can design an adequate synchronizer. However, what is adequate? A good rule-of-thumb is a mean time before failure of ten years for all systems shipped. For example, there is ten asynchronous inputs in our system that need to be synchronized. We are planning to ship one hundred thousand systems. Using a cascaded synchronizer with N=2, the MTBF would be 4.72 centuries.

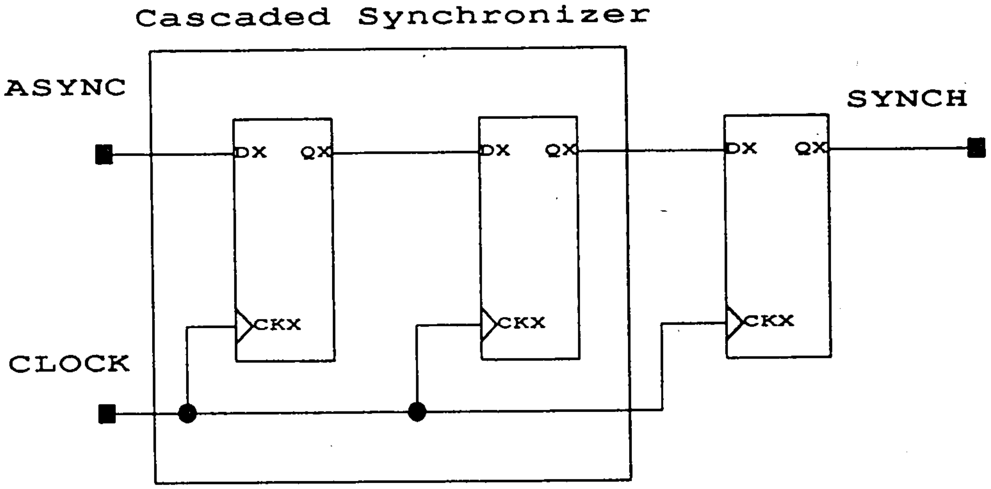


Figure 6.1: Cascaded Synchronizer

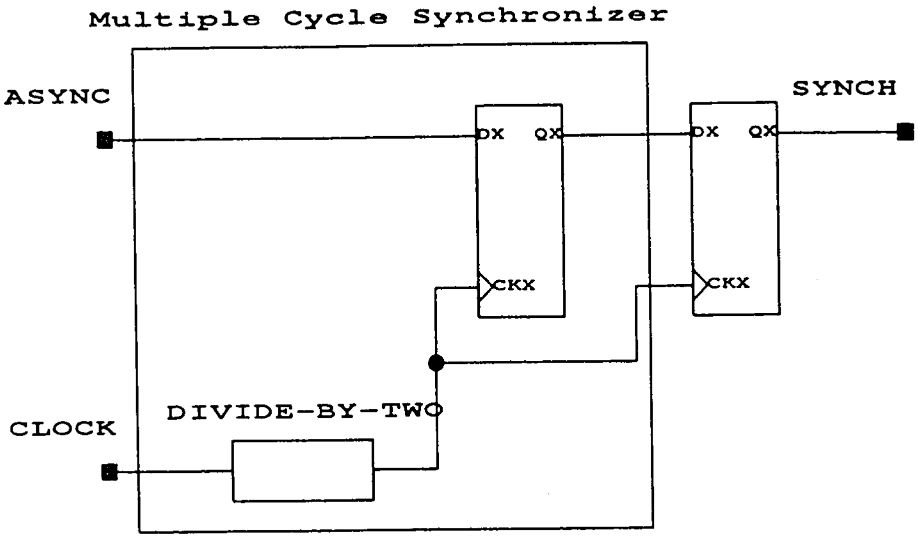


Figure 6.2: Multiple Cycle Synchronizer

7.0 Comparison of Different Devices

One of the first rules in reducing metastability is to use the best synchronizer available. In this section, many different commercial devices were compared. For this comparison, results from data sheets and technical papers were used. These results were then compared with the results of the AT&T ORCA FPGA. Comparisons were made with gate arrays, standard products, programmable logic devices, and other field programmable gate arrays For comparison purposes, a mean time before failure of ten years is used. Also, there are ten asynchronous inputs, and one hundred thousand systems shipped. Thus, the required MTBF is 3.15E+14. Using this value of MTBF with equations (7.1), (7.2), and (7.3), the maximum system frequency for each device can be calculated. Table 7.1 shows the results of the comparison.

$$\mathrm{MTBF} = (\exp(T_r))*(1/(F_c*F_d*2*T_0)) \qquad (7.1)$$

where

$$T_r = 1/F_c - T_{delay} \qquad (7.2)$$

where

$$T_{delay} = T_p + T_{set} + T_{route} + T_d \qquad (7.3)$$

The results used in this section are assumed to be for a power supply voltage of 5.0 VDC, and an ambient temperature of 25 C. The input data rate is 10 MHz.

TABLE 7.1: Comparison of Commercial Devices

Device	T_{delay} (nS)	τ (pS)	T_0 (S)	FMAX (MHz)
FD1	8.0	236	7.52E-11	56
74S74	45.0	1700	1.00E-03	7
SN74ABT7819	14.0	300	7.00E-12	39
85C220	10.0	220	5.20E+01	40
ACT1	12.0	216	5.00E-10	46
QL12X16-2	3.8	185	1.23E-10	79
XC3020	13.0	248	5.11E-10	42
ORCA	4.2	205	7.94E-12	80

7.2 Gate Arrays

In cell-based gate arrays functional elements are created by connecting the transistors in a cell together, and then if needed connecting the cells together. By changing these connections, any type of logical function can be created. Storage elements such as flip-flops can also be implemented in this fashion. Unfortunately these cell-based flip-flops have capacitance at these connection points. This capacitance degrades the resolving characteristics of these flip-flops. Another factor that determines the metastability characteristics is the loading on the flip-flop output. In a gate array the output of the flip-flop routes to other logic without first being buffered. This output can be loaded differently depending how the design is implemented. This difference in loading can complicate the analysis of the flip-flop's metastability behavior.

The LSI gate array flip-flop FD1 is fabricated in a 1.5 micron CMOS process. Using the parameters from reference [20], the maximum frequency of operation is calculated which can sustain an MTBF of 3.15E+14. The maximum frequency is 56 MHz.

7.3 Standard Products

Standard products typically have good metastability characteristics because their flip-flops are custom designed. These custom design techniques typically improve the flip-flops' performance, and metastability characteristics. In some cases, the flip-flops output drives externally off-chip. When this output is not buffered, the loading from external components can have dramatic effects on the standard products flip-flop.

The Texas Instruments quad D flip-flop 74S74 is designed in a Shottky-clamped TTL bipolar process [47]. The 74S74 has flip-flop outputs that drive externally. External parasitic capacitance and inductance have harmful effects on the flip-flop. The outputs of these devices oscillate when in a metastable state. This oscillation is partly caused by these external parasitics. The 74S74 flip-flop had the worst performance sustaining only a maximum frequency of 7 MHz.

The Texas Instruments FIFO SN74ABT7819 is fabricated in a 0.8 micron BICMOS process [44]. These FIFOs have internal synchronizers for the asynchronous input signals. These synchronizers are designed with two cascaded flip-flops. The flip-flops are designed for optimal metastability characteristics. Also, the delay between the flip-flops is minimized. The effects of one of the flips-flops in the cascade is studied. The SN74ABT7819 flip-flop can sustain a maximum performance of 39 MHz. The actual synchronizer has two cascaded flip-flops. This cascade allows this synchronizer to operate at a frequency of greater than 50 MHz. The FIFO flip-flops must drive off-chip.

7.4 Programmable Logic Devices

Programmable logic devices (PLDs) typically comprise an array of AND gates connected to an array of OR gates. The output of the OR gates can be registered with a flip-flop. The PLD flip-flop is custom designed for high performance, and good metastability characteristics. Unfortunately, the data input must pass through the AND-OR arrays before getting to the flip-flop. The delay through the arrays increases the flip-flop setup time therefore decreasing the resolution time.

The Intel' 85C220 is a Programmable Logic Device fabricated in a 1.0 micron CMOS technology. Using the results from reference [8], the maximum operating frequency is 40 MHz to obtain the desired MTBF.

7.5 Field Programmable Gate Arrays

Field Programmable Gate Arrays (FPGAs) are comprised of an array of logic cells surrounded by a ring of input/output cells. An interconnecting structure connects the logic blocks together. The logic cells implement all the logic required for the design, and any storage requirements. In this section, four types of FPGAs were studied. These four FPGAs were chosen, because they were the only ones to have published metastability characteristics. More advanced FPGAs are available with probably better characteristics, but they could not be compared because of a lack of data. The four FPGAs studied are the Actel ACT1, the QuickLogic Q12X16-2, the Xilinx XC3020-70, and the AT&T ORCA.

The Actel ACT1 is an anti-fuse based FPGA fabricated in a 2.0 micron CMOS process [1]. Figure 3.3 showed the ACT1 logic cell. Flip-flops must be implemented using multiplexers, and thus these flip-flops have poor performance, large setup times, and poor metastability characteristics. The anti-fuse used for interconnect has an on-resistance of 400 ohms, and a capacitance of 4 fF. These low values of resistance and capacitance reduce the interconnect delay. The maximum operating frequency is 40 MHz for an MTBF of 3.15E+14 seconds. It should be noted that Actel now has a family of FPGAs with a dedicated flip-flop in the logic cell. This family, ACT3, is fabricated in a 0.8 micron CMOS process. No metastable information was available for this part, but it probably has much better characteristics than the ACT1 family of FPGAs.

The QuickLogic Q12X16-2 is an anti-fuse based FPGA fabricated in a 0.65 micron CMOS process [35]. Figure 3.4 showed the QuickLogic function cell. This logic cell includes a dedicated flip-flop. This dedicated flip-flop has excellent metastability characteristics, but has a large setup time because there is no direct data input to the flip-flop. In the QuickLogic FPGA, a via-link anti-fuse is used for interconnect. This type of anti-fuse has an on-resistance of approximately 65 ohms, and a capacitance of 1.3 fF. Due to the low resistance and capacitance values, the QuickLogic FPGA has the lowest interconnect delay of all the FPGAs studied. To maintain an MTBF of 3.15E+14, the maximum operating frequency of the QuickLogic part is 79 MHz.

The Xilinx 3020-70 is an SRAM based FPGA fabricated in a 1.25 micron CMOS process [51]. Figure 3.5 showed the Xilinx function cell. This function cell includes two dedicated flip-flops that have good metastability characteristics. One of the flip-flops can have an input that bypasses the look-up table (LUT), decreasing that flip-flops setup time. The other flip-flop input must come through the LUT, increasing that flip-flops setup time. The outputs of the flip-flops go through a buffer before leaving the logic cell. This buffer eliminates the effects of loading on the metastability characteristics. In the Xilinx FPGA, a pass transistor is used for interconnect. This pass transistor has a high on-resistance of 1500 ohms, and a high capacitance of 15 fF. Because of these high resistance and capacitance values, the Xilinx FPGA has the highest interconnect delay. The maximum operating rate for the FPGA to maintain the required error rate is 42 MHz. It should be noted that Xilinx now has a more recent family of FPGAs. This family, XC4000, is fabricated in a 0.6 micron CMOS process. There was no available metastable information for this part, but it should have much better characteristics than the XC3000 family of FPGAs.

The AT&T ORCA is an SRAM based FPGA fabricated in a 0.5 micron CMOS process [4]. Figure 3.7 shows the AT&T ORCA logic cell. Within the logic cell, there are four dedicated flip-flops that have excellent metastability characteristics. All the data inputs for the flip-flops can come from either the LUTs, or these inputs can bypass the LUTs. By bypassing the LUTs, the flip-flops setup time is reduced. With a lower setup time, the resolution time decreases. Also, the outputs of the flip-flops are buffered before leaving the logic cell. This buffering eliminates the effects of loading on the metastability characteristics. In the AT&T FPGA, a pass transistor is used for interconnect. This pass transistor has a high on-resistance of 500 ohms, and a high capacitance of 5 fF. Because of these high resistance and capacitance values, the AT&T FPGA has a high interconnect delay. However, the delays for AT&T FPGA are smaller than the delays for the Xilinx FPGA. To sustain an MTBF of 3.15E+14, the maximum operating frequency of the AT&T ORCA FPGA is 80 MHz.

To create the best synchronizer, a number of commercial devices were compared. The benchmark for comparison was the maximum system frequency, that would sustain a mean time before failure of ten years. Furthermore, the system has ten asynchronous inputs, and one hundred thousand systems will be shipped. Thus, the required MTBF is 3.15E+14. The AT&T ORCA FPGA had the highest operating frequency of 80 MHz while the Texas Instruments 74S74 had the lowest operating frequency of 7 MHZ.

8.0 Conclusion

Metastability is a phenomenon where the output of a flip-flop remains in an undefined state for an indeterminate period of time. This phenomenon is caused by violating the setup time and hold time of the flip-flop. Typically, these violations are unavoidable when the flip-flop is used to synchronize an asynchronous signal. The task of the system designer is to reduce the probability of a metastable event. To quantify metastable events, an equation which determines the mean time before failure (MTBF) is used.

In this thesis, there is a review of the theory of calculating MTBF. This theory uses a simple latch as an example. By using the latch, the constants that affect this phenomenon are analyzed. The first constant τ affects how quickly the latch will resolve to a defined state (one or zero). The second constant T_0 affects how likely the input signal will cause a metastable state. By using τ and T_0 in conjunction with the data frequency and clock frequency, the MTBF can be predicted.

Field Programmable Gate Arrays (FPGAs) are similar to Gate Arrays, but are user-programmable instead of mask-programmable. These FPGAs are used increasingly in system design. However, few of these devices have been characterized for metastability.

The AT&T Optimized Reconfigurable Cell Array (ORCA) FPGA was characterized in this thesis through simulation. An algorithm was created to determine a flip-flop's metastable window. After the window was determined, the metastable constants τ and T_0 were calculated. Simulations were performed over the full range of processing, temperature, and power supply conditions.

Experimental methods were also performed to determine the metastability constants. A test setup was constructed using a test board, and various test equipment. This setup in association with a special test circuit was used to find the constants τ and T_0. Five samples were tested. One of the samples was then tested over the power supply range of 4.5 VDC to 5.5 VDC.

Design techniques to reduce MTBF were then evaluated. The most dramatic reductions were produced by two techniques: the cascaded synchronizer, and the multiple cycle synchronizer. Both of these techniques increased the mean time before failure by a power of N.

A number of commercial devices were compared for their metastable characteristics. For comparison purposes, a mean time before failure of ten years was used. The maximum frequency of operation for each device to sustain this MTBF was determined. The AT&T ORCA device had the highest operating frequency of 80 MHz.

The AT&T ORCA FPGA had the highest operating frequency because it is fabricated in a state-of-the-art 0.5 micron technology, has dedicated flip-flops in the logic cells, has direct data inputs to the flip-flops, and has buffered outputs. The 0.5 micron technology reduces capacitances, thereby increasing performance. The dedicated flip-flops have increased gain which gives the flip-flops excellent metastability characteristics. Input setup time for the flip-flops is reduced by using direct inputs. This decreased setup time gives a longer resolution time which in turn causes a longer mean time before failure. Finally by buffering the outputs of the flip-flops, their metastability characteristics are not affected by loading.

References

1. Actel, ACT Family Field Programmable Gate Array DATABOOK, Actel, Inc., Sunnyvale, CA, March 1991.

2. Anceau, F. "A Synchronous Approach for Clock VLSI Systems" IEEE J. Solid-State Circuits, Vol. SC-17, No. 1, February 1982.

3. Antognetti, P., Massobrio, G., Semiconductor Device Modeling with Spice, McGraw-Hill 1988.

4. AT&T, The ORCA 2C Field-Programmable Gate Arrays Data Sheet, AT&T, Allentown, PA, May 1994.

5. Birkner, John, M., "Understanding Metastability," Wecon 87 (San Francisco, CA, Nov. 17-19, 1987) Electronics Convention Management, Los Angeles, CA, 90045, 16/3.

6. Bolton, Martin, "A Guided Tour of 35 Years of Metastability Research" Wecon 87 (San Francisco, CA, Nov. 17-19, 1987) Electronics Convention Management, Los Angeles, CA, 90045, 16/4.

7. Brown, Stephen, D., Francis, Robert, J., Rose, Jonathan, Vranesic, Zvonko, G., Field-Programmable Gate Arrays, Kluwer Academic Publishers, Boston, MA, 1992.

8. Browns, Thom, "Metastability Characteristics of Intel PLDs," Intel Application Note, AP-336, Intel Corporation, September 1993.

9. Bursky, Dave "Clock-free Macrocells Simplify Asynchronous System Design" Electronics Design, July 1988.

10. Catt, I., "Time Loss Through Gating of Asynchronous Logic Signal Pulses," IEEE Trans. Electronic Computers, pp. 108-111, February 1966

11. Chaney, T. J., Ornstein, S., M., Littlefield, W., M., "Beware the Synchronizer," COMPCON-72 IEEE Computer Society Conference, San Francisco, CA, pp. 317-319, September 12-14, 1972.

12. Chaney, T., J., "Comments on 'A Note on Synchronizer and Interlock Maloperation'." IEEE Trans. on Computers, Vol. C-28, pp. 802-804, October 1979.

13. Chaney, T., J., Molnar, C., E., "Anomalous Behavior of Synchronizers and Arbiter Circuits," IEEE Trans. on Computers, Vol. C-22, No. 4, pp. 421-422, April 1973.

14. Couranz, G., R., Wann, D., F., "Theoretical and Experimental Behavior of Synchronizers Operating in the Metastable Region," IEEE Trans. on Computers, Vol. C-24, No. 6, pp. 604-616, June 1975.

15. Dike, Charles, "A Metastability Primer," Signetics Standard Products Data Book, AN219, pp. 280 - 282, Signetics, Inc., San Jose, CA, November 1989

16. Flannagan, S., T., "Synchronization Reliability in CMOS Technology," IEEE J. Solid-State Circuits, Vol. SC-20, No. 4, pp. 880-882, February 1987.

17. Gabara, T., J., Cyr, G., J., Stroud, C., E., "Metastability of CMOS Master/Slave Flip-Flops," AT&T Bell Laboratories Technical Memorandum, June 28, 1991.

18. Glasser, L., A., Dobberpuhl, D., W., The Design and Analysis of VLSI Circuits, Addison-Wesley Publishing Company, Inc., Reading, MA, 1980.

19. Hodges, D., A., Jackson, H., G., Analysis and Design of Digital Integrated Circuits, McGraw-Hill, Inc., 1983.

20. Horstman, J., U., Eichtel, H., W., Coates, R., L., "Metastability of CMOS Latch/ Flip-Flop," IEEE J. Solid-State Circuits, Vol. SC-24, No. 1, pp. 141-146, February 1989.

21. Jackson, T., A., Albicki, A., "Analysis of Metastable Operation in RS CMOS Flip-Flops," IEEE J. Solid-State Circuits, Vol. SC-22, No. 1, pp. 57-64, February 1987.

22. Kelly, Bob, "Minimize Metastability in 50 MHz State Machines," Phillips Components-Signetics Programmable Logic Devices Data Book, pp. 619-627, Phillips Components-Signetics, Inc., April 1990.

23. Kim, L., S., Dutton, R., "Metastability of CMOS Latch/Flip-Flop," IEEE J. Solid-State Circuits, Vol. SC-25, No. 4, pp. 942-951, August 1990.

24. Kleeman, L., Cantoni, A., "Metastable Behavior in Digital Systems," IEEE Design & Test of Computers, pp. 4-19, December 1987.

25. Lubkin, S., "Asynchronous Signals in Digital Computers," Mathematical Tables and Other Aids to Computation, Vol. 6, No. 40, pp. 238-241, October 1952.

26. Marino, L., R., "The Effects of Asynchronous Inputs on Sequential Network Reliability," IEEE Trans. on Computers, Vol. C-26, pp. 1082-1090, December 1977.

27. Muller, R., S., Kamins, T., I., Device Electronics for Integrated Circuits, John Wiley & Sons, 1986.

28. Marrin, Ken "Metastability Haunts VMEbus and Multibus II System Designers," Computer Design, August 1985.

29. Mead, Carver, Conway, Lynn, Introduction to VLSI Systems, Addison-Wesley Publishing Company, Inc., Reading, MA, 1980.

30. Melchiorre, Robert, "A Study of Metastability in CMOS Latches," MSEE thesis, Lehigh University, May 1992.

31. Nguyen, Hoang, "How to Detect Metastability Problems" ASIC & EDA, pp. 16-24, February 1993.

32. Nootbar, Keith, Spehn, Richard, "Design, Testing, and Application of a Metastable-Hardened Flip-Flop," Wecon 87 (San Francisco, CA, Nov. 17-19, 1987) Electronics Convention Management, Los Angeles, CA, 90045, 16/2.

33. Pechoucek, "Anomalous Response Times of Input Synchronizers," IEEE Trans. on Computers, Vol C-25, No. 2, pp. 133-139, February 1976.

34. Pfannkoch, T., A., "ADVICE 1N User's Guide, "AT&T Bell Laboratories Technical Memorandum, April 18, 1988.

35. QuickLogic, "Metastability Report for FPGAs," QuickLogic Data Book, pp. 5-23 to 5-26, QuickLogic, Inc., San Jose, CA, 1994

36. Rosenberg, F., Chaney, T., J., "Flip-Flop Resolving Time Test Circuits," IEEE J. Solid-State Circuits, Vol. SC-17, pp. 731-738, August 1982.

37. Rubin, Kim, "Metastability Testing in PALs.".Wecon 87 (San Francisco, CA, Nov. 17-19, 1987) Electronics Convention Management, Los Angeles, CA, 90045, 16/1.

38. Sakurai, T., "Optimization of CMOS arbiter and Synchronizer Circuits with Submicrometer MOSFET's," IEEE J. Solid-State Circuits, Vol. SC-23, No. 4, pp. 901-906, August 1988.

39. Shoji, M., "Mechanisms of Long-lasting Metastable States in CMOS D-Latches," AT&T Bell Laboratories Technical Memorandum, April 5, 1991.

40. Signetics, "A Metastability Primer," Signetics Standard Products Data Book, AN220, pp. 283-285, Signetics, Inc., San Jose, CA, November 1989

41. Signetics, "Metastability Tests for the 74F786 - A 4 input Asynchronous Bus Arbiter," Signetics Fast Products Data Book, AN217, pp. 7-96 to 7-99, Signetics, Inc., San Jose, CA, July 18, 1988.

42. Stoll, P., A., "How to Avoid Synchronization Problems," VLSI Design, pp. 56-59, November 1982.

43. Stucki, M., J., Cox, J., R., "Synchronization Strategies," Proceedings of the Caltech Conference on VLSI, pp. 357-374, Caltech, January 1979.

44. Texas Instruments, The High Performance FIFO Memories Data Book, Texas Instruments, Inc., Dallas, TX, 1994.

45. Veendrick, H., J., M., "The Behavior of Flip-Flops Used as Synchronizers and Prediction of Their Failure Rate," IEEE J. Solid-State Circuits, Vol. SC-15, No. 2, pp. 169-176, April 1980.

46. Wakerly, John "A Designers Guide to Synchronizers and Metastability, Part 1 and 2," VLSI Design, September 1987.

47. Wakerly, John, F., "A Designers Guide to Synchronizers and Metastability," Center for Reliable Computing Technical Report, CSL, TN #88-341, Computer Systems Laboratory, Departments of Electrical Engineering and Computer Science, Stanford University, Stanford, CA, February, 1988.

48. Weste, Neil, H., E., Eshraghian, Kamran, Principles of CMOS VLSI Design: A Systems Perspective, Addison-Wesley Publishing Company, Inc., Reading, MA, 1988.

49. Wormald, E., G., "A Note on Synchronizer and Interlock Maloperation," IEEE Trans. on Computers, Vol C-26, pp. 317-318, March 1977.

50. Wormald, E., G., "Support for Chaney's 'Comments on a Note on Synchronizer and Interlock Maloperation," IEEE Trans. on Computers, Vol. C-28, pp. 802-804, October 1979.

51. Xilinx, The Programmable Logic Data Book, Xilinx, Inc., San Jose, CA, 1994.

52. Zojer, B., Petschacher, R., Lusching, W., A., "A 6-bit/200-MHz full Nyquist A/D converter," IEEE J. Solid-State Circuits, Vol. SC-20, No. 3, pp. 780-786, June 1985.

Vita

Alan Cunningham was born on April 6, 1965 in Glasgow, Scotland to John and Maureen Cunningham. He received his Bachelor of Science Degree in Electrical Engineering from The Pennsylvania State University.

He is presently employed as a Member of Technical Staff by AT&T Microelectronics in Allentown, Pennsylvania. His current position is in the Field Programmable Gate Array Application Engineering Group.

END OF TITLE