

# Synchronization and Arbitration in GALS

David Kinniment<sup>1,2</sup>

*School of EECE  
Newcastle University  
Newcastle upon Tyne, UK*

## Abstract

Synchronizers and arbiters are important components of any Globally Asynchronous, Locally Synchronous network. They contribute to latency, because of the synchronization time required for reliability, and to metastability delay in the arbiters. Simple models of metastability have served us well up to now, but more recently work on the characterisation of deep metastability has demonstrated effects in commonly used components that may need to be taken into account.

*Keywords:* Synchronization, Arbitration, Metastability.

## 1 Introduction

Synchronization and arbitration are fundamental to the operation of a network of independently clocked processors because asynchronous data originating from outside a clocked processor has to be synchronized to the processor clock before it can be used. This usually involves a choice made on the basis of the arrival time of the data, which is a continuous variable, and the result of the decision, which clock edge to use, is discrete. In this situation it is possible to have very small input time differences between the data available signal and the clock. Such choices are difficult as the input energy to the synchronizer circuit falls towards zero when the time difference becomes small, and the response time of the circuit then approaches infinity. In fact there is always a point at which the synchronizer time response can become longer than the available time, and the synchronizer circuit fails because its output has not reached a stable state [1], [5]. Asynchronous systems have a similar problem where the system must decide which of two or more asynchronous requests for a common processing resource is to be granted first. In this case the circuit

<sup>1</sup> The author would like to thank the UK EPSRC for financial support, and those at Newcastle University, Sun Microsystems and Intel Corp who have taken part in many useful discussions.

<sup>2</sup> Email: [david.kinniment@ncl.ac.uk](mailto:david.kinniment@ncl.ac.uk)

that makes the decision is an arbiter, and the input energy to the arbiter is also derived from the time differences between the inputs. Again, this time difference is a continuous variable and can also approach zero, with the result that the arbitration time can approach infinity [5], [7]. In both cases the decision element, usually a flip flop, has become metastable, and the resolution time of metastability in a flip flop is important to the reliability of the system.

Typically networks of GALS processors are connected via routers, with network adapters connecting each processor to the network. If the network is asynchronous, and the processors synchronous, it will be necessary to synchronize the data in each network adapter, and arbitrate requests for common resources in the routers. Both reliability and communication times around the network are affected by the reliability of the synchronizers and the metastability time in the arbiters, and here we will examine the characteristics of some metastable circuits, and how they can affect the systems that use them in ways which are not always easily predictable. In section 2 we look at synchronizers reliability, and the effects of nanometer process, in section 3 we deal with the MUTEX, and effects which may alter its performance in an arbiter. Section 4 examines the effects of noise and non-uniform distributions of events in synchronizers and arbiters, and finally section 5 uncovers some anomalous responses in commonly used circuits.

## 2 Synchronizer reliability

Using a simple model of metastability, represented by the solution of the second order differential equation of cross coupled gates, the output voltage  $V_1$  can be described in terms of time  $t$ , and a time constant  $\tau$ , which is determined by the gain bandwidth product of the transistors  $\tau = \frac{C}{AG} = \frac{1}{\text{GainBandwidth}}$ . This leads to a closed form expression for the output voltage as a function of time [3] [4]

$$V_1 = K_a e^{-\frac{t}{\tau_a}} + K_b e^{\frac{t}{\tau_b}}$$

It is possible to estimate from this the input conditions necessary to produce a metastable response of a given time, and hence how often this time will be exceeded in a synchronizer. It is usual to simplify the expression to the second term where the output voltage  $V_1$  for a latch in metastability diverges exponentially from the metastable level so that at time  $t$  such that:

$$V_1 = K_b e^{\frac{t}{\tau}}$$

The initial condition,  $K_b$ , depends on the input time overlap between clock and data. If the data input changes to a high much earlier than the clock, will be positive, so the output voltage will reach the digital high value of  $+\frac{V_{dd}}{2}$  quickly. If it changes from low to high much later than the clock will be negative, so the output voltage will reach a low value quickly. In between, the value of will vary according to the relative clock data timing, and at some critical point  $K_b = 0$ , so the output voltage is stuck at zero. We will call the clock data timing that gives  $K_b = 0$ , the balance point, and we will measure input times relative to this point using the symbol  $\Delta t_{in}$ . If the synchronizer fails when the input overlap is shorter than  $\Delta t_{in}$  by giving an output response longer than the resolution time allowed,

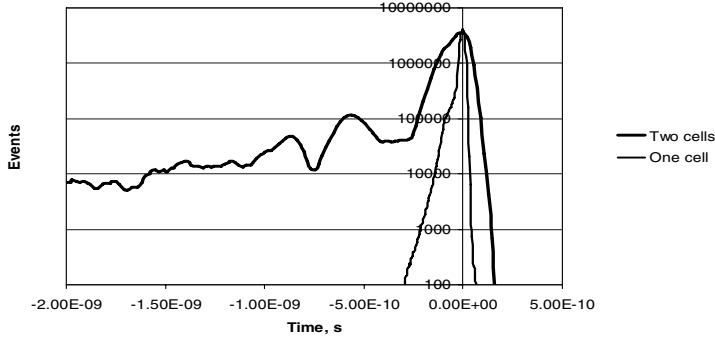


Fig. 1. Multiple FPGA cell latch

the mean time between failures will be:

$$MTBF = \frac{1}{f_d f_c \Delta t_{in}}$$

Failure occurs when  $\Delta t_{in} = T_w e^{-\frac{t}{\tau}}$ , so we can get the mean time between failures:

$$MTBF = \frac{e^{\frac{t}{\tau}}}{f_d f_c T_w}, [2]$$

where  $T_w$  is known as the metastability window. MTBF depends on  $f_d$  and  $f_c$ , which are system parameters rather than circuit parameters, so we can get a characterization of a synchronizer which is dependent only on the circuit parameters  $\tau$  and  $T_w$ :

$$\Delta t_{in} = T_w e^{-\frac{t}{\tau}}$$

This analysis applies to most simple latches, but may not hold for more complex devices, made from gates with more than one time constant in the feedback loop, or with long interconnections. It is very important for any flip-flop used for synchronization that it is characterized as a single cell with a fixed layout, and not assembled from individual gates or, for example, FPGA cells, because the feedback interconnection may have additional time constants, and the differential equation that describes the small signal behavior will be correspondingly complex. An example of multiple time constants is shown in figure 1, where a latch has been assembled out of two FPGA cells. In this figure the latch output change is being used to trigger the oscilloscope, and the figure shows a histogram of clock events. The time scale is 0.5ns/division. Because of the additional delay in the feedback loop the resolution time constant is nearly 1ns, and the histogram of clock to output times rises and falls as a result of oscillation in the output trajectory. For comparison the histogram from a standard latch on the same FPGA is also shown. Here, the resolution time constant is much faster at around 40ps and there is no sign of oscillation.

If the time allowed for metastability is short, we must also take into account the first term in the response equation. For many metastable events, the initial conditions are such that  $K_b$ , which is the voltage difference between output nodes at the start, is less than 10mV, and while  $K_a$ , the common offset at the start, may be as much as 0.5V. In these circumstances the  $K_a$  term is important, but it is only important in determining the response times for metastable events resolving early. This is because simple circuits using gates with only one time constant always

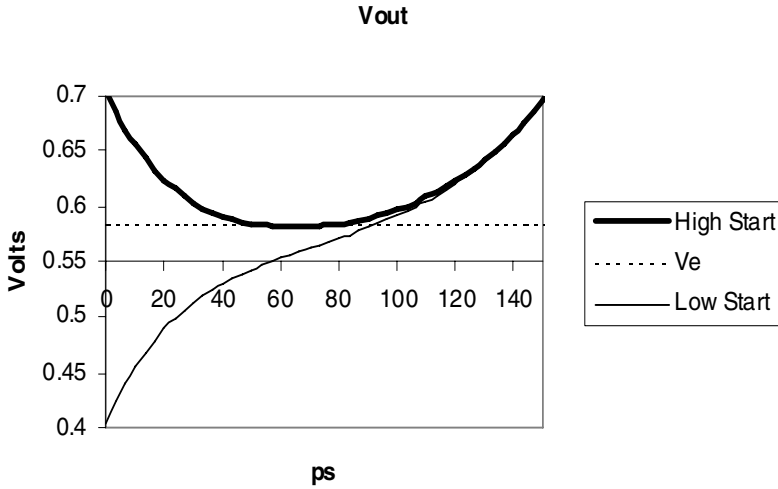


Fig. 2. Model response

reach the linear region quickly.  $\tau_a$  is small outside the linear region and so the  $K_a$  term becomes small very quickly. The metastable events that cause synchronizer failures last a long time, and will have trajectories that spend most of their time in the linear region and so we can usually still use the simplified second order model to predict the time it takes flip-flop circuits to escape from metastability. The difference between early and late trajectories, and how they are affected by the  $K_a$  term can be seen in figure 2.

This figure shows the response of the two outputs in a latch against time as predicted by the small signal model when both outputs start from a point higher than the metastable level and from a point lower. There is a common mode initial offset of both outputs given by  $K_a = +150\text{mV}$  and  $150\text{ mV}$  from the metastable level of  $0.55\text{V}$  for the high start and low start curves respectively. For both these trajectories  $K_b = 4\text{ mV}$ , representing the initial difference between the two outputs. We use  $\tau_a = 25\text{ ps}$  and  $\tau_b = 42.5\text{ ps}$  to represent a typical situation where A is about 5-10. It is common for the exit from metastability to be detected by an inverter with a slightly different threshold from the metastability level of the flip-flop. Thus when  $V_{out}$  exceeds that level the inverter output changes from a high to a low. The metastability level of the flip-flop here is  $0.55\text{V}$ , and the threshold level is  $0.58\text{V}$  (the dotted line).  $V_{out}$  exceeds  $0.58\text{V}$  at  $80\text{ps}$  for the high start of  $0.7\text{V}$  and at  $93\text{ps}$  for the low start of  $0.4\text{V}$ . Note that the time difference depends upon the threshold level, and that if the high start trajectory never goes below  $0.58\text{V}$ , that is if  $K_b > 4.7\text{ mV}$ , metastability is not detectable for output time delays between 0 and  $65\text{ps}$  because the detecting inverter output remains low all the time. From these curves it can be seen that the metastability resolution time depends on the value of  $K_a$  during the first  $100\text{ps}$ , but not beyond that.

The most common representation of the reliability of a synchronizer is to count

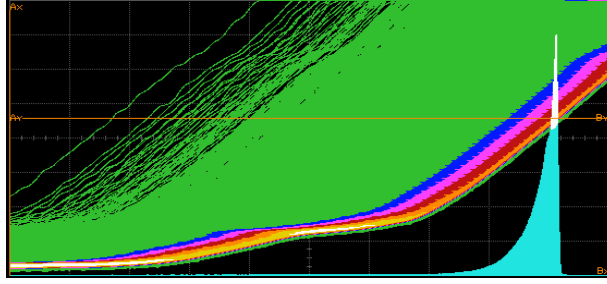


Fig. 3. Output escapes against time

the number of times the output escapes from metastability within  $\delta t$  of  $t$ .  $T$  is the period over which events are measured, and the number of events that resolve within  $t$  is:

$$-\frac{d}{dt}[f_d f_c T T_w e^{\frac{-t}{\tau}}] \delta t, \text{ or } -\frac{\delta t}{\tau} [f_d f_c T T_w e^{\frac{-t}{\tau}}]$$

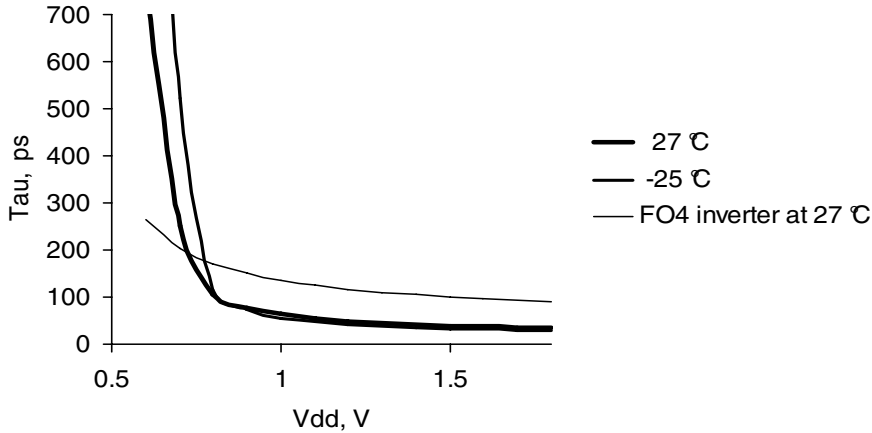
Oscilloscope traces showing metastable escapes against time are shown in figure

3.

### 2.1 Future Processes

One of the problems of synchronizers in submicron technology is that latches using cross coupled inverters do not perform well at low voltages and low temperatures. Since the value of  $\tau$  depends on the small signal parameters of the inverters in the latch it is more sensitive to power supply variations than the large signal propagation delay expected when an inverter is used normally. This is because the conductance of both the p and n type devices can become very low when the gate voltage approaches the transistor threshold voltage  $V_T$ , and consequently time constants can become very high. As  $V_{dd}$  reduces in submicron processes, and  $V_T$  increases, the problem of increased  $\tau$  and therefore greatly increased synchronization time gets worse. Typical plots of  $\tau$  against  $V_{dd}$  for a  $0.18\mu$  process is shown in figure 4. It can be observed from this figure that  $\tau$  increases with  $V_{dd}$  decreasing and the reduction in speed becomes quite rapid where  $V_{dd}$  approaches the sum of thresholds of p and n-type transistors so that the value of  $\tau$  is more than doubled at a  $V_{dd}$  of 0.9V, and more than an order of magnitude higher at 0.7V,  $-25^\circ\text{C}$ . For comparison the typical large signal inverter delay with a fan out of four (FO4) in this technology is shown. This demonstrates  $\tau$  is likely to track the processor logic delay rather poorly, making design difficult. The increase in  $\tau$  can have a very important effect on reliability. For example, a synchronizer in a system where a time equivalent to  $30\tau$  has been allowed might give a synchronizer MTBF of 10,000 years. A 33 % increase for  $\tau$ , in this synchronizer will cause that time fall to an equivalent of only  $20\tau$ . As a result the MTBF drops by a factor of  $e^{-10}$  from 10,000 years, to less than 6 months.

It is therefore very important, that worst case variations of all parameters, such as process fluctuations, temperature, and power supply are taken into account in any estimate of  $\tau$  to ensure that the reliability of the system under all operating conditions is as expected, and circuits are needed that are robust to variations of process, voltage and temperature.

Fig. 4. Jamb latch  $\tau$  vs  $V_{dd}$ 

One way of improving the value of  $\tau$  is to increase the current in the transistors by increasing all transistor widths, but this will also increase power dissipation. In order to estimate the average energy used during metastability, we will assume that the average metastability time is  $\tau$ . As the transistor width increases, the total switching energy increases in proportion but  $\tau$  only decreases slowly as transistor sizes increase, and reaches a limit at around 31ps. While  $\tau$  can be optimized for conventional circuits, sensitivity to PVT variation remains a problem. An improved synchronizer circuit [11] that is much less sensitive to power supply variations is shown in figure 5.

This circuit is essentially a modified Jamb latch where two  $0.8\mu$  p-type load transistors are switched on when the latch is metastable so as to maintain sufficient current to keep the total transconductance high even at supply voltages less than the sum of thresholds of the p and n-type transistors. Two  $0.5\mu$  feedback p-types are added in order to maintain the state of the latch when the main  $0.8\mu$  p-type loads are turned off. Because of these additional feedback p-types, the main p-types need only to be switched on during metastability, and the total power consumption is not excessive. In the implementation of figure 5 a metastability filter is used to produce the synchronizer output signals, which can only go low if the two nodes have a significantly different voltage. The outputs from the metastability filter are both high immediately after switching, and are then fed into a NAND gate to produce the control signal for the gates of two main p-types. In this circuit, main p-types are off when the circuit is not switching, operating like a conventional Jamb latch, but at lower power, then when the circuit enters metastability the p-types are switched on to allow fast switching. The main output is taken from the metastability filter, again to avoid any metastable levels being presented to following circuits. Now there is no need for the feedback p-types to be large, so set and reset can also be small. The optimum transistor sizes for the improved synchronizer are shown in

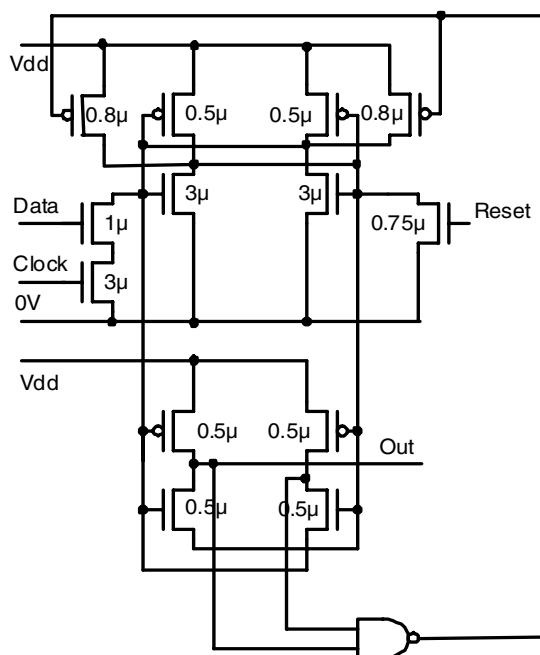


Fig. 5. Robust synchronizer

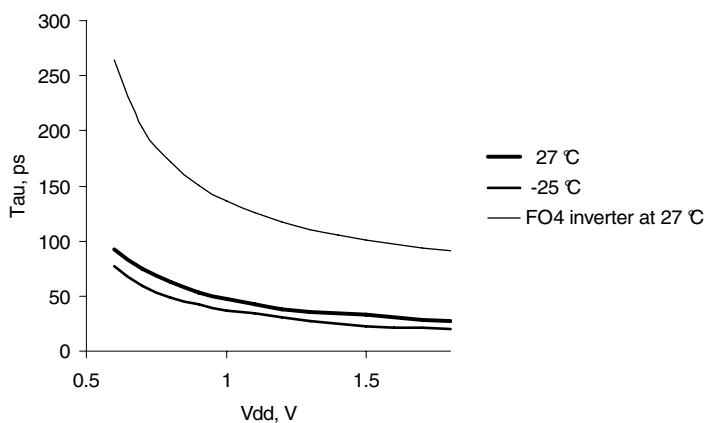


Fig. 6. Robust synchronizer  $\tau$  vs  $V_{dd}$

figure 5, and the resultant  $\tau$  at  $V_{dd}$  of  $1.8v$  is as low as  $27.1ps$  because the main transconductance is provided by large n-type devices and because there are two additional p-types contributing to the gain. It also operates well at  $0.6V$   $V_{dd}$  and  $-25^{\circ}C$ , because it does not rely on any series p and n-type transistors being both switched on by the same gate voltage. The relationship between  $\tau$  and  $V_{dd}$  for the improved synchronizer is shown below in figure 6.

The switching energy for this circuit is 20 % higher than a conventional Jamb latch. At the same time as maintaining a low value of  $\tau$ , the ratio between  $\tau$  and FO4 is much more constant at around 1:3 over a wide range of  $V_{dd}$  and temperature.

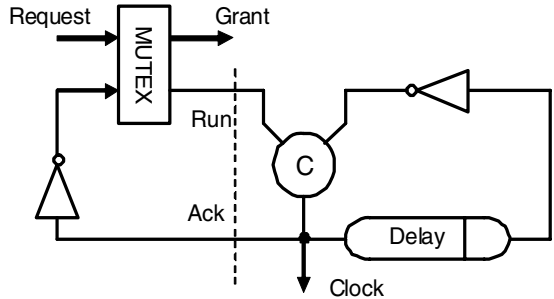


Fig. 7. Pausible clock

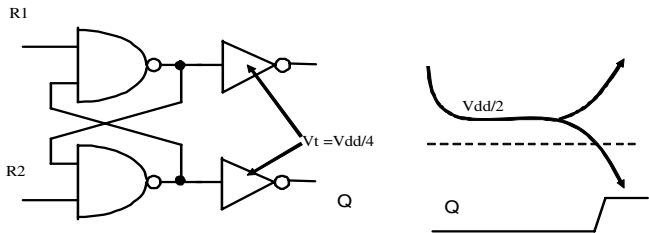


Fig. 8. MUX with low threshold inverters

### 3 Arbitration and the MUX

Mutual exclusion circuits are the core component of arbiters and of pausable GALS clock generators. A typical MUX is used to arbitrate between an input request and the clock as shown in figure 7.

When the request signal goes high the run signal stays low if it is already low (clock is high), or goes low the next time the clock goes high. The clock cycle is completed by the other input to the C gate going low after the delay time. The clock then goes low, and a high going edge arrives at the C gate after the delay. Because run remains low a new cycle starts. When the request goes low again, the clock is low, run goes high and a new cycle of the clock is initiated. The characteristics of MUX circuits vary depending on how metastable levels are filtered out. There are two basic ways of providing such filters, either gates with a threshold level slightly different from the metastable level, figure 8, or a more sophisticated filter figure 9

Figure 8 shows a simple MUX made up of two NAND gates. When, both R1 and R2 go high and the MUX outputs may become metastable. Without the two inverters on the outputs a metastable level of  $\frac{V_{dd}}{2}$  could cause any following digital circuits to malfunction. The inverters on the outputs prevent the half level appearing at the output because they have a lower than normal threshold level. If the latch is in a metastable state, both inverter outputs are low because the output level when the circuit is metastable is higher than the output inverter threshold. Only when one latch output move lower than the inverter threshold can the corresponding inverter output go high. An alternative metastability filter arrangement, [8] is shown in figure 10 where a high output only appears when there is sufficient



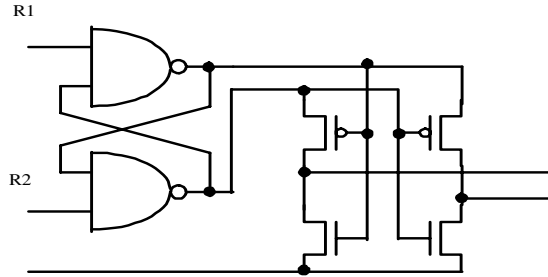


Fig. 9. with metastability filter

difference between the two latch outputs. The advantage of this is that it will filter out metastable outputs where both output voltages have the same value irrespective of whether the output voltage is, high, low, or half level. Thus it can also be used to remove the effects of metastability in circuits where both outputs go up and down in phase. Only when they diverge can an output appear from the filter. In figure 9 one of R1 and R2 may go high just before the other causing the latch to start to change state, but if the overlap is short the latch may be left in metastability. The NAND gate outputs both start high, but the filter outputs are low because the two p-type transistors are non-conducting

When both gate outputs go to the same metastable level, the filter outputs remain low, and as the metastability resolves, the latch outputs diverge. Only when there is a difference of at least  $V_t$  between the gate outputs can the filter output start to rise, so that one output rises when the high output gate is at about:  $\frac{V_{dd}+V_t}{2}$ , and the low output gate is at about  $\frac{V_{dd}+V_t}{2}$ .

### 3.1 Effects of Filtering

The event histograms of latches with metastability filters can be affected by the nature of the filter. By SPICE simulations and measurements on sample circuits it is possible to find the output times for a range of input time differences between R1, and R2. Typical results are shown in the event histograms of figure 10 and figure 11.

Figure 10 shows the effect of the circuit with a filter of the type shown in figure 9, where the gate outputs must go more than  $\frac{V_t}{2}$  in order to escape from metastability. Output times in this figure are measured as the elapsed time in multiples of  $\tau$  after the last input goes high. In figure 11 the initial slope is significantly faster than the final slope, because the outputs are taken from low threshold inverters with a threshold only about  $\frac{V_{dd}}{30}$  below the metastable level. Early trajectories therefore escape more quickly than the later ones even though the transistor sizes are similar in the two filters.

In figure 10 the slope of the trend line is slower because the loading on the MUTEX outputs is greater, but the increase in delay applies equally to early and late events, so there is little difference between initial and final slope. In fact loading effects often dominate the time delays in a MUTEX, particularly if the low threshold inverters are made by paralleling the inputs on a multi-input OR gate. The resulting

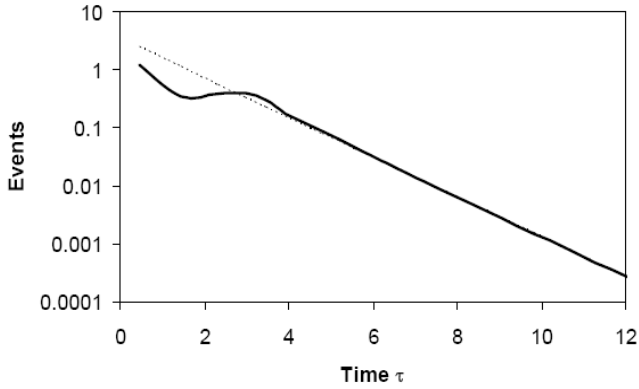


Fig. 10. Histogram for metastability filter

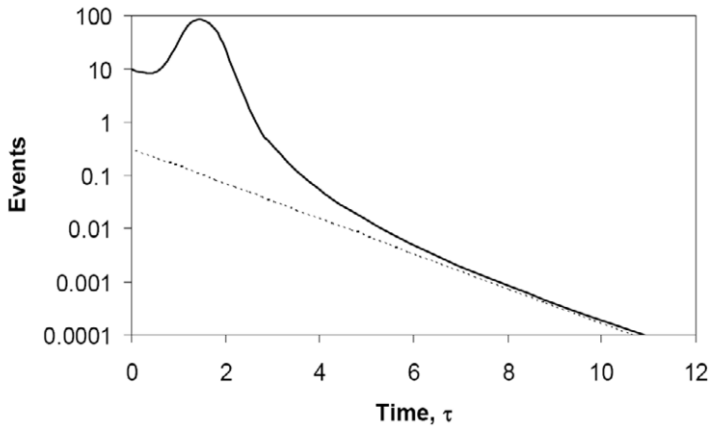


Fig. 11. Histogram for MUTEX with low threshold output inverters

large increase in capacitance loading may easily double both the delay time and the metastability time constant  $\tau$ .

## 4 Noise and Non-uniform distributions

In a synchronizer application all input time distributions are equally probable and noise does not affect the probability of a particular outcome. In order to show this, let us assume a normal distribution of noise voltages such that the probability of a noise voltage being within  $dv_1$  of  $v_1$  is

$$P_1(v_1) = \frac{1}{e_n \sqrt{2\pi}} e^{\frac{-v_1^2}{2e_n^2}} . dv_1$$

Here the RMS noise voltage is  $e_n$ , and the probability of the noise voltage being somewhere between  $-\infty$  and  $+\infty$  is:

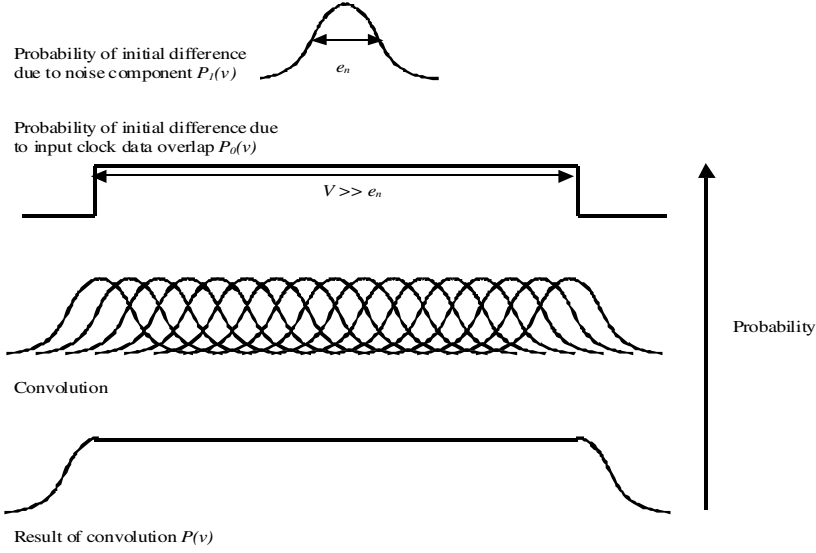


Fig. 12. Convolution with noise voltage

$$\int_{-\infty}^{\infty} \frac{1}{e_n \sqrt{2\pi}} e^{-\frac{v_1^2}{2e_n^2}} dv_1 = 1$$

Further, we will assume for a normal synchronizer that the probability of a particular initial voltage difference,  $K_b$ , being within  $dv$  of  $v$  is

$$P_0(v) = \frac{dv}{V}$$

where  $V$  is the range of voltages over which the probability is constant. If the probability density of the initial voltage difference is constant, then the probability density of the trajectories at any time is also constant. We convolve these two to get the resulting probability density of initial differences at the latch nodes.

$$P(v) = \int_{-\infty}^{\infty} P_1(v_1) \cdot P_0(v - v_1) \cdot dv_1 \cdot dv$$

In a normal synchronizer  $P_0(v - v_1)$  is a constant over a range much wider than the noise voltage, ( $V \gg e_n$ ), so it can be taken outside the integral

$$P(v) = \frac{1}{V} \int_{-V/2}^{V/2} \frac{1}{e_n \sqrt{2\pi}} e^{-\frac{v_1^2}{2e_n^2}} dv_1, \text{ and } \int_{-\infty}^{\infty} \frac{1}{e_n \sqrt{2\pi}} e^{-\frac{v_1^2}{2e_n^2}} dv_1 = 1.$$

The final probability density is also constant,

$$P(v) = \frac{dv}{V}$$

and the result is nearly the same as the one we started with.

The process of convolution is illustrated in figure 12, where the noise distribution  $P_1(v)$ , is convolved with the initial difference distribution  $P_0(v)$ , to produce the result  $P(v)$ . In fact it does not matter much that we used a Gaussian distribution of noise voltages, any other distribution with negligible probability outside  $V/2$  and  $+V/2$  would give a similar result.

Sometimes the two inputs can become intentionally or unintentionally locked together so that the data available signal always changes at exactly the same relative time. If this time the balance point of the latch or MUTEX, the input is called a malicious input, and metastability times could become very long were it not for the

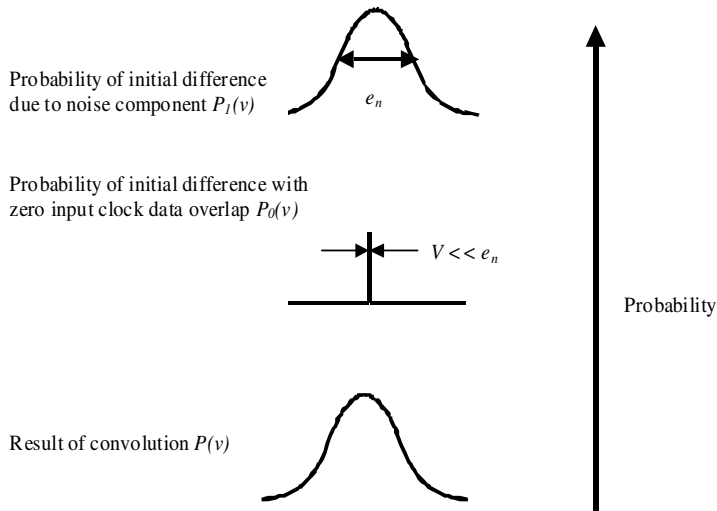


Fig. 13. Malicious input

presence of noise. For a malicious input the initial difference voltage between the nodes is always 0V, and the situation is very different from a normal synchronizer because  $V \ll e_n$ , so that if  $v = 0$ ,  $P_0(0) = 1$ , and if  $v \neq 0$   $P_0(v) = 0$ . Looking at the two cases separately,

when  $v_1 = v$ ,

$$P_1(v_1).P_0(v - v_1) = P_1(v), \text{ Otherwise } P_1(v_1).P_0(v - v_1) = 0$$

This means that

$$\int_{-\infty}^{\infty} P_1(v_1).P_0(v - v_1).dv_1 = P_1(v), \text{ or } P(v) = \frac{1}{e_n\sqrt{2\pi}} e^{-\frac{v^2}{2e_n^2}}.dv.$$

So the result of adding noise to an initial time difference between clock and data difference that is always equal to the balance point, is to produce a distribution of initial differences equal to the noise distribution. This is the intuitively obvious result shown in figure 13, in which the noise has the effect of knocking the metastability off balance so that it resolves quicker.

The effect of noise on the average metastability time is now determined by the probability of the initial difference being near the balance point, so it is possible to compute the MTBF for a synchronizer in a typical application where all values of data to clock times are equally probable. This was shown to be  $MTBF = \frac{e^{\frac{t}{\tau}}}{f_d f_c T_w}$  earlier. For the case where the initial difference is determined solely by noise, the probability of a long metastable time where  $K_b$  is less than  $V_x$  is given by the number of clocks in time T, multiplied by the probability of a noise voltage less than  $V_x$ , or  $[f_c T] \left[ \frac{V_x}{e_n\sqrt{2\pi}} \right]$ .

This gives  $MTBF = \frac{e_n\sqrt{2\pi}.e^{\frac{t}{\tau}}}{f_c.V_e}$ . Comparing the two formulae, it can be seen that  $f_d$  no longer appears in the formula for a malicious input because the data is assumed to change at exactly the same time as the clock.  $T_w$  is associated with input conditions, which do not now determine the input time, and the noise allows the flip-flop to resolve quicker, larger noise voltages giving longer MTBF, and no

noise meaning zero MTBF. As an example, if  $T_w = 10$  ps,  $e_n = 0.8\text{mV}$ , and  $f_c, f_d = 100\text{MHz}$ , a normal synchronizer would have an MTBF of  $10^{-5} \cdot e^{t/\tau}$ , where one with a malicious input would have an MTBF of  $2 \cdot 10^{-10} \cdot e^{t/\tau}$ . To get the reliability for a synchronizer with a malicious input to be the same as the normal synchronizer reliability with a uniform distribution of clock-data overlaps we would need an  $11\tau$  longer synchronization time. A figure of  $e_n = 0.8\text{mV}$ , or  $0.1\text{ps}$  time variation, would represent the effects of internal thermal noise only. In practice jitter on the clock or data of a small system might be  $4\text{ps}$ , and in a large system  $20\text{ps}$  clock jitter would not be unusual. If this jitter has a Gaussian distribution it would be sufficient to add  $7.1\tau$  or  $5.5\tau$  to the synchronization time in order to get the same reliability.

#### 4.1 Asynchronous systems

In an asynchronous system arbitration is subject to delay due to metastability, but if the relative timing of the competing requests is uniformly distributed, the average additional delay from this cause is only  $\tau$ . In many systems the requests are not uniformly distributed, for example in a pipelined processor where both instructions and operands generate requests from different parts of the pipeline to the same cache memory. In this case the dynamics of the system may cause both requests to collide frequently, and in the worst case, always at similar times. In this worst case situation jitter and noise could spread the request spacing over a range of, say,  $5\text{ps}$  rather than exactly zero every time. Even a system with nominally constant probability density for input time spacing can have 3:1 or 4:1 variations as in figure 14

The time  $t$  taken by the MUTEX to resolve can be found from averaging the time over all possible inputs

$$t = \tau \cdot \ln \left[ \frac{T_w}{\Delta t_{in}} \right], \text{AverageTime} = \tau \cdot \int_0^\infty \ln \left( \frac{T_w}{\Delta t_{in}} \right) \cdot d\Delta t_{in}$$

If the time variation is restricted either because of a malicious input and noise or a smaller than normal range of possible input times to  $T_n$ , we need to average the response time over the range 0 to  $T_n$ .

$$\text{AverageTime} = \tau \cdot \int_0^{T_n} \ln \left( \frac{T_w}{\Delta t_{in}} \right) \cdot d\Delta t_{in}$$

The result of averaging is:

$$\text{AverageTime} = \tau \cdot [1 + \ln \left( \frac{T_w}{T_n} \right)]$$

So if the noise is as wide or wider than the metastability window, the extra time is still only  $\tau$ , but if it is much less, for example if  $T_w$  is  $100\text{ps}$ , and  $T_n$  as low as  $5\text{ps}$ , the average time might be  $4\tau$  rather than  $\tau$

## 5 Measuring Effects in deep metastability

The traditional measurement method uses two oscillators with a similar frequency to provide data and clock for the synchronizer. In this way the overlap time between data and clock is evenly distributed over a range of times equal to the difference between the clock and data periods. The drawbacks of this method include the relatively few deep metastability events close to the balance point as these events



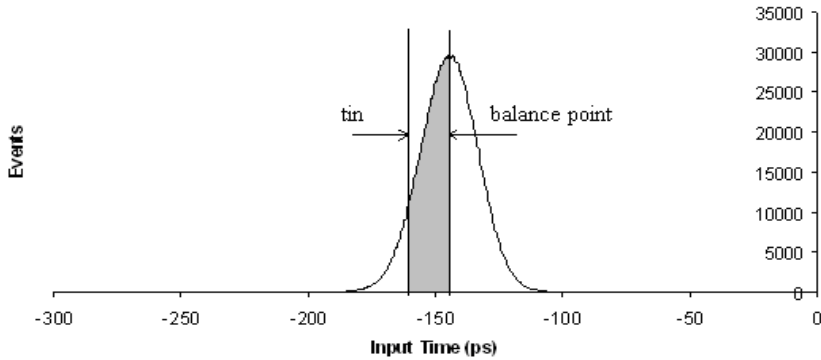


Fig. 16. Input histogram of Jamb latch

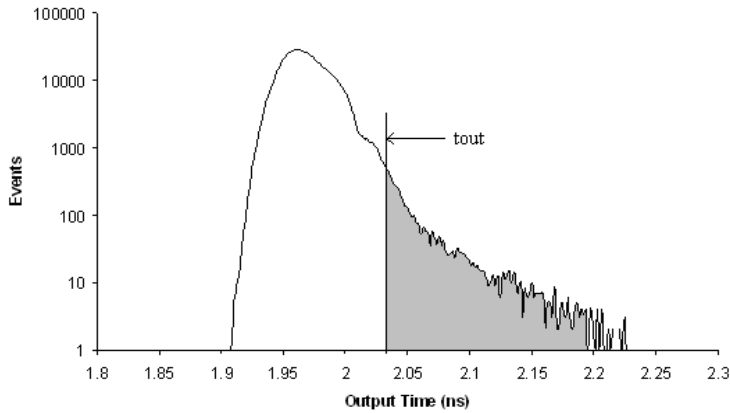


Fig. 17. Output histogram of Jamb latch

count as the input count given by  $t_{in}$  establishes the correspondence between  $t_{in}$  and  $t_{out}$  as shown in figures 16 and 17.

The method is described in detail in [6], and allows us to construct the input time vs output time from the input distribution and output distributions recorded by the oscilloscope. One problem is that the input time distribution is obscured by measurement noise. [6] also shows how this noise can be removed by adjusting the ratio of high output events and low output events. Many synchronizer circuits have a slightly worse reliability in the second half of the clock cycle than might be expected from projecting the trends measured in the first half of the cycle, and some are significantly worse. This is often because attention has been given to minimizing the metastability time constant of the master flip flop, but not the slave. There is also a hand-over effect when the clock goes low [10], [6] which increases the delay further. While these effects can be estimated theoretically, or by simulation [10], they are not easy to measure using conventional methods. The techniques described in [6] can allow measurement of synchronizer reliability into the region after the back edge of the clock. In some flip flops, the low going clock transition is associated with a 1.5 - 2ns increase in output time for events after the back edge. This additional delay is clearly observable in the back edge measurements of figure

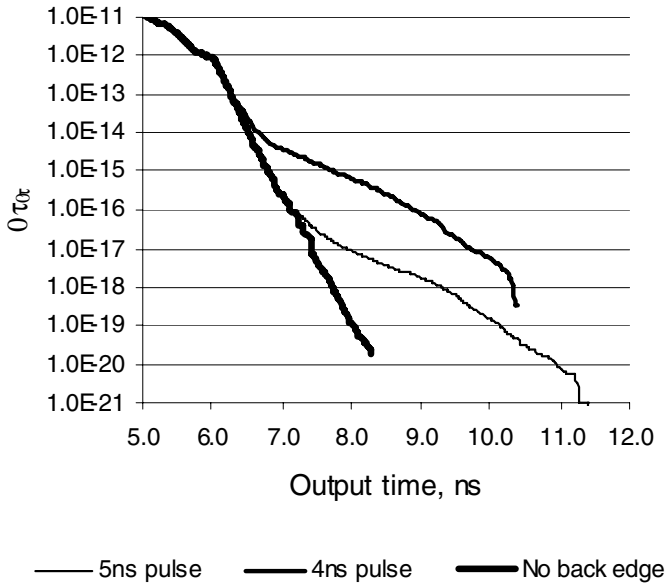


Fig. 18. flip flop Input time vs. Output time for different back edge times

18 where the input output time characteristics for a long clock high pulse (50ns) are compared with those for shorter pulses of 4ns and 5 ns. The long pulse has little effect on the metastability resolution time, but the short ones add up to 2ns to metastability resolution times after the back edge.

## 6 Conclusions

Synchronization and arbitration components can comprise a n important part of the latency involved in communication paths in a GALS system. Calculating the performance of these components is not necessarily straight forward, and the effect of noise is to make the computation paths non-deterministic. Noise increases as dimensions reduce, so non-determinism also increases making probabilistic estimations of performance necessary Until recently tools to enable the simulation of reliability in the second half of the clock cycle have not been available. This has resulted in the poor design of some synchronizer library elements which have a longer metastability time constant in the second half of the clock cycle as compared with the first half. Taking this together with the additional delay involved in the hand over of metastability from master to slave, the reliability may be a several orders of magnitude worse than expected

## References

- [1] Chaney, T.J., and C.E.Molnar, *Anomalous behavior of synchronizer and arbiter circuits*, IEEE Transactions on Computers, **C-22**(4), (1973), 412-422.



- [2] Dike, C.E., and E Burton, *Miller and Noise Effects in a Synchronizing Flip-Flop* IEEE Journal of Solid State Circuits **34**(6), (1999), 849-855,
- [3] Jeppson, K.O., *Comments on the Metastable Behavior of Mismatched CMOS Latches* IEEE Journal of Solid State Circuits **31**(2), (1996), 275-277
- [4] Kinniment, D.J., A. Bystrov, A.V. Yakovlev, *Synchronization Circuit Performance*, IEEE Journal of Solid-State Circuits, **37**(2), (2002), 202-209.
- [5] Kinniment, D.J., and D. B. G. Edwards, *Circuit Technology in a large computer System* Proc. conference on Computers-Systems and Technology London, (1972), 441-449
- [6] Kinniment, D.J., K Heron, and G Russell *Measuring Deep Metastability*, Proc. ASYNC'06, Grenoble, France, (2006), 2-11
- [7] Kinniment, D.J., and J.V.Woods, *Synchronization and arbitration circuits in digital systems*, Proc. IEE **123**(10), (1976) 961-966
- [8] Seitz, C.L., *Ideas about arbiters*. Lambda,**1** (First Quarter)(1980) 10-14.
- [9] Veendrick, H.J.M., *The behavior of flip-flops used as synchronizers and prediction of their failure rate*, IEEE Journal of Solid-State Circuits, **SC-15**(2), (1980), 169-176.
- [10] Yang, S., and M Greenstreet, *Computing Synchronizer Failure Probabilities* Proc. DATE 07, (2007), 1-6.
- [11] Zhou, Z., D.J.Kinniment, G. Russell, and A. Yakovlev, *A Robust Synchronizer Circuit*, Proc. ISVLSI06, (2006), 442-443.