A New Synchronizer Design
Jacqueline Walker, Student Member, IEEE, and
Antonio Cantoni, Senior Member, IEEE

Abstract—A new synchronizer design is presented. Current synchronizer designs have certain disadvantages, both in characterization and in the tradeoff between settling time and sampling rate, which are overcome in the new design. Two possible implementations of the synchronizer are discussed.

Index Terms—Metastability, synchronizer, asynchronous, synchronization, synchronizer design, flip-flop, synchronous digital systems.

1 INTRODUCTION

In digital system design, the need to introduce an asynchronous signal into a synchronous system arises quite frequently. A common occurrence is in a digital system that samples a digital signal generated by an external autonomous device [6], [12]. An asynchronous input to a synchronous system may come from another synchronous system [3] or arise in a globally asynchronous, locally synchronous system when synchronous subsystems need to communicate [1]. An equivalent problem is the need to arbitrate between competing requests for a shared resource [2], [5] or to choose between competing commands which arise from independent sources [7].

It has become well known that asynchronous inputs in synchronous systems can lead to system failures due to bistable devices entering metastable states. Metastable states may occur in bistable devices when the timing constraints of the device are not observed [4], [6], [9], as can be the case with an asynchronous input which can change at any time with respect to the clock edges of the synchronous system. In the metastable state, the output of the device does not reach either of the valid logic levels but hovers between the two for a time that is long compared with the normal timing delays of the device [9] or may even oscillate [5], [10]. System failure occurs because of the inconsistent response of other devices within the system to the metastable output [6].

To overcome the problem of metastable failure within systems, synchronizers have been adopted as an interface between the asynchronous input and the synchronous system. The synchronizer samples the asynchronous input at the rate of the system clock. The output of the synchronizer is thus synchronous with the rest of the system. However, the use of a synchronizer does not eliminate the possibility of metastable failure, it is only possible to limit its occurrence to within the synchronizer and thereby minimize its effect on the system.

Modeling of metastability in bistable devices has shown that the probability of metastable failure, when the timing of data edge transitions is modeled by a Poisson process, decreases exponentially with the length of time allowed for the output to settle [6], [11], [13]. As a result the performance of a synchronizer can be substantially improved by increasing the delay before the output is accessed [6]. Thus when designing synchronizers the aim should be to provide an adequate settling time.

REFERENCES
The aperture model [6] of metastable behavior can be used to predict the mean time between failure (MTBF) for the synchronizer and verify that the settling time is sufficient. Consider a single flip-flop which is being used to synchronize an asynchronous signal to a clock of frequency $f = 1/T$. The aperture is a region of time associated with each clock event such that if a data edge occurs in the aperture, a metastable state will occur in the flip-flop. The probability of no metastable failure of the flip-flop in some time interval $[0, T_s]$, when the timing of data edge transitions is modeled by a Poisson process, is given approximately by

$$P(T_s) = e^{-\lambda T_s}$$

(1)

where $\lambda$ is the mean rate of arrival of the data edges, $\Delta t = T_0 e^{-T_{fl}/T}$ is the width of the aperture, $T_0$ and $T$ are parameters describing the metastability performance of the flip-flop, and $T_s$ is the settling time. Consistent with (1), synchronization failure is itself a Poisson process, hence the MTBF is given by

$$MTBF = \frac{T_0 e^{T_{fl}/T}}{\lambda T_0}$$

(2)

Equation (2) also shows that the MTBF of a synchronizer is inversely proportional to the sampling rate of the synchronizer [4], [5], [6].

In Section 2, we present existing high sampling rate synchronizer designs and discuss their performance and limitations. In Section 3, we present a new synchronizer design, briefly discuss some possible implementations, and analyze its performance and highlight the advantages of the new approach.

2 EXISTING SYNCHRONIZER DESIGNS

As noted above, the aim when designing synchronizers should be to provide an adequate settling time. In this section, we discuss existing synchronizer designs and the different approaches taken to achieve this aim. Modeling of the synchronizer's metastable failure probability is undertaken using the aperture model [6] and an expression for MTBF is obtained. Failure is considered to occur when the flip-flop at the output of the synchronizer samples a metastable state.

The simplest method, illustrated in Fig. 1, is to use a chain of $(N + 1)$ flip-flops. It is assumed that a metastable state is transferred along the chain of flip-flops by simple sampling [6], thus allowing a settling time of $T_s = NT - (N - 1)T_p$ where $T_p$ is the propagation delay of a flip-flop and $T$ is the clock period. Using the aperture model summarized in Section 1, the MTBF is given by

$$MTBF = \frac{(N-1)T \cdot e^{-T_{fl}/T}}{\lambda T_0}$$

(3)

Note the assumption that all flip-flops in the chain have identical parameters.

![Fig. 1. A shift register synchronizer.](image)

A second type, illustrated in Fig. 2, uses three flip-flops and a divided clock. Using the same assumption of simple sampling, the settling time is given by $T_s = NT - T_p - T_f$ where $T_f$ is the propagation delay of a flip-flop, $T_p$ is the delay in the clock divider circuit, and $T$ is the clock period. The application of the aperture model presented in Section 1 shows that the MTBF is given by

$$MTBF = \frac{(N-1)T \cdot e^{-T_{fl}/T}}{\lambda T_0}$$

(4)

Again, an assumption is necessary that the first two flip-flops have identical parameters. For a given $T$ the mean time between failures of the divided clock synchronizer is $(N-1)$ times that of the shift register synchronizer, but the trade-off is the $(N-1)$ times lower sampling rate.

3 A NEW SYNCHRONIZER DESIGN

This section discusses a new synchronizer design and presents possible implementations of the design. As shown in Fig. 3, the new synchronizer consists of four blocks: a sequential sampling register, a multiplexer, an output flip-flop, and a timing generator. The sequential sampling register has $N$ independent flip-flops which all see the same asynchronous input, but are clocked cyclically. Each flip-flop within the sequential sampling register is clocked at $1/T$ the rate of the clock. Viewed sequentially, each sample is taken at the clock frequency, but each successive sample is taken at a different flip-flop.

![Fig. 2. A divided clock synchronizer.](image)

The shift register synchronizer is simple, but characterization of a practical realization is difficult. Unlike the theoretical analysis that assumes the transfer of the memory states from device to device by "simple sampling" [6], real systems may introduce additional dynamics between the devices and also the devices may not be well matched. The divided clock synchronizer is compact but is not an ideal replacement because of the lower sampling rate.

![Fig. 3. A possible implementation of the new synchronizer.](image)

The $N$ outputs of the flip-flops are connected to the multiplexer together with $N$ corresponding select signals. The select signals are
phased with respect to the flip-flop strobe by the timing generator, so that a given flip-flop's output is not selected until it has had \((N-1)\) clock periods to settle. Thus, the multiplexer presents the delayed samples to the output flip-flop at the same rate as the samples are taken. This is a key difference with respect to the synchronizer of Fig. 2 which only samples the input every \((N-1)\) clock periods. The output flip-flop is clocked synchronously with the clocking of the flip-flops in the sequential sampling register.

3.1 A Possible Implementation

Fig. 4 shows the timing diagram for the new synchronizer implementation shown in Fig. 3.

Every flip-flop within the sequential sampling register sees the asynchronous input. The \(i\)th flip-flop of the sequential sampling register is clocked by the \(i\)th strobe signal from the timing generator. The multiplexer function is implemented by and-ing the output of the \(i\)th flip-flop from the sequential sampling register with the \(i\)th select signal from the timing generator and then or-ing the resulting signals. Thus, the output of one flip-flop at a time is routed to the output flip-flop.

Fig. 5 shows the detail of a possible implementation of the timing generator. The shift register and nand gate combination is self-starting and circulates a zero through the shift register thus generating the cycle of \(N\) successive pulses for the strobe and select signals.

Each pathway within the new synchronizer design, from a single sampling flip-flop to the output flip-flop, is equivalent to the divided clock synchronizer of Fig. 2. However, for a given clock period \(T\), the overall device containing \(N\) of these units samples the asynchronous input at a rate which is \((N-1)\) times higher than the sampling rate of Fig. 2.

Using the same definition as in Section 2 that failure is considered to occur when the flip-flop at the output of the synchronizer samples a metastable state, the settling time for the new design is given by 

\[
T_s = (N-1)T + T/2 + t_{p1} + t_{p2} + t_{pmux}
\]

where \(t_{p1}\) is the propagation delay for the inverted clock signal, \(t_{p2}\) is the propagation delay for the strobe and select signals produced by the timing generator, and \(t_{pmux}\) is the propagation delay of the multiplexer. Applying the aperture model, the MTBF for the new design for the synchronizer is given by

3.2 A Synchronous Front End for the New Synchronizer

For ease of testing, it is preferable to use purely synchronous designs, for example, this simplifies the use of the scan path methodology. A possible alternative implementation of the synchronizer front end is in Fig. 6 and the corresponding timing diagram is shown in Fig. 7.
settling time determined by the number of flip-flops used and increases reliance on restrictive assumptions of flip-flop behavior in samples at the system clock rate. When characterization is required it reduces to consideration of two flip-flops which determine viability of this correspondence.

The enable takes the place of the strobe in the previous implementation and is activated cyclically at \( \frac{1}{2} \) the clock rate, routing the asynchronous input to the input of each flip-flop in turn. The flip-flops are clocked at the clock rate, thus although each flip-flop only samples the asynchronous input once every \( N \) clock cycles, a sample is taken of the asynchronous input on every clock edge. When the enable input of a particular flip-flop is not activated, the output of that flip-flop is routed to its input. Thus, at every clock edge while its enable is not active, the flip-flop resamples its own output. For testing purposes the front end can be converted to a scan register. The timing of the data input can be forced to satisfy the necessary set-up and hold time requirements during testing.

The implementation of the new synchronizer design incorporating a synchronous front end is analogous to a shift register synchronizer with the first \( N - 1 \) flip-flops in the chain being exactly the same flip-flop. The settling time for the new design with a synchronous front end is given by \( T_S = (N - 1)T - (N - 2)t_p - t_{pmux} \) where \( t_{pmux} \) is the propagation delay through the multiplexer, \( t_p \) is the propagation delay through a sampling flip-flop and its multiplexed input, and \( T \) is the clock period. Applying the aperture model, the MTBF for the new synchronizer with a synchronous front end is given by

\[
\frac{T_e}{AT_0} = \frac{T}{T_0} (N - 1)T - (N - 2)t_p - t_{pmux}
\]

The assumptions for characterization of a practical realization of this design can still be relaxed compared with characterization of a shift register synchronizer because the first \( N - 1 \) flip-flops in the chain are the same flip-flop, hence the assumption that the flip-flop parameters are identical is valid. Characterization is not as straightforward as in the previous implementation because it still relies on characterizing how metastability propagates in a device which is being clocked while settling is presumed to occur.

It may be possible to further improve the performance of the new synchronizer with known techniques such as using asymmetric threshold devices interspersed between the appropriate flip-flops. However, the study of these techniques is beyond the scope of this correspondence.

4 Conclusion

When synchronous systems have asynchronous inputs, the problem of metastability has to be faced. One way to reduce the probability of metastable failure is to use a synchronizer with an adequate settling time. This paper has presented a new method and two possible implementations for a synchronizer which provides a settling time determined by the number of flip-flops used and samples at the system clock rate. When characterization is required it reduces to consideration of two flip-flops which decreases reliance on restrictive assumptions of flip-flop behavior in the shift register synchronizer with the same fast sampling and settling times.

Acknowledgment

Jaccaline Walker is supported in part by a grant from the Australian Telecommunications and Electronics Research Board.

5 References