Quality Optimisation in Digital Integrated Circuit Design

Loek Thijssen

Delft University of Technology
Faculty of Information Technology and Systems
Department of Electrical Engineering
P.O. Box 5031, 2600 GA Delft, The Netherlands
E-mail A.P.Thijssen@et.tudelft.nl

Abstract

The quality of digital integrated circuits is usually expressed in parameters such as mean time between failure, life expectancy and in the percentage of guaranteed good circuits on delivery. There are other parameters as well like error handling, EMC hardness, error probability during life time, robustness to noise, built in margins for clock skew in flip-flops, timing margins for data transfer, etc. In this paper we will study some of these aspects, that significantly influence the quality of digital circuits.

Keywords: clock skew, flip-flop design, flip-flop timing, logic optimisation, noise effects, quality of ICs, reliability, state redundancy.

1 Introduction

For a clear understanding of quality aspects of ICs we must define what we understand under their quality. In this paper quality is to which extend the circuit meets the expectations of the user during the economical lifetime of the circuit. In general, for ICs this is a correct functioning of the circuit within the application in an electronic device, under predefined environmental conditions. No circuit is completely failure free. EMC noise and other disturbances, for example, switching phenomena in the power supply may cause short deviations of the expected environmental conditions, that can result in an erroneous behaviour during a certain interval of time. The probability of such an error is expressed in the mean time between failure parameter. The duration of a failure and its consequences must be taken into account as well.

Normally, a logic designer concentrates on the specification and implementation of the logic function of the circuit to be designed, and on its performance. Less attention is paid to error handling and error probability. The result may be a too long time to market because of intermittent errors in prototypes of the circuit. Error handling must be taken into account during the functional design level of the circuit. It is an essential part of the requirements specification. In many cases, specification of the error handling is a very tedious job, taking more design time than the functional specification of the circuit. This conclusion applies for software as well, as any computer user knows by experience.

On the implementation level, important aspects of error handling are redundant states in sequential circuits. In a redundant state the behaviour of the circuit has not been specified in advance. Redundant states are frequently used for the reduction of next-state equations of flip-flops. It has been shown [Thijssen, 1993] that state equivalence can solve state redundancy in such a way that error handling can be implemented without or with only a small increase of the number of gate equivalents for the circuit. Retiming techniques, in which the combinational logic is pipelined, introduce a large number of redundant states. An error handling for those states must be considered as well. If not, the behaviour of the circuit after an error has not completely been specified.
Intermittent errors are a more serious problem. They are hard to detect in a design. During production tests circuits with intermittent errors have a very high escape rate. Detection requires an uneconomical number of repetitions of a test procedure [Beenker, 1995]. A better approach is to avoid this type of errors as much as possible by a robust electronic design of the circuit. For a better electronic design many aspects are to be taken into account. Most of them deal with creating margins during signal transitions, the DC and AC noise margins on signal levels and margins for clock skew during signal transitions in synchronous circuits. Another aspect is flip-flop timing and the width of the intervals in which the data must be present. The effect of noise pulses on the internal states of flip-flops is another item.

2 Parameter specification

Parameters play a dominant role in digital circuit design and implementation. For example, timing parameters model the behaviour of the circuit during signal transitions. In digital circuits signals are in the logic high level H, in the logic low level L or the signal level is in between H and L. In the later case the signal is said to be logically undefined. The duration of the undefined interval is specified with two parameters $t_{P(min)}$ and $t_{P(max)}$. Figure 1 shows how both parameters are defined.

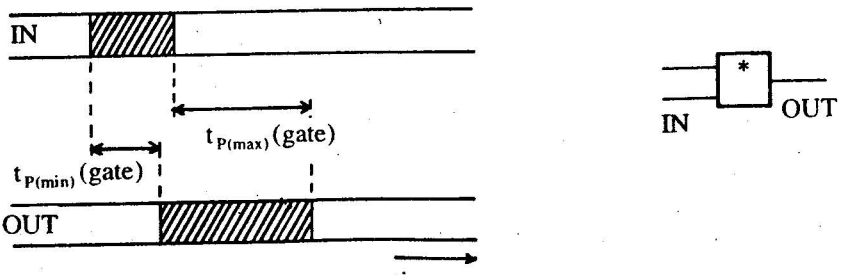

***Figure 1.** Timing model for a logic gate.*

The minimum propagation time has to be measured under 'best case' switching conditions and the maximum propagation time under 'worst case' conditions. In this way the designer knows when the logical value of a signal is available. The definition

of these parameters should be done in a user-oriented way. A logic signal is said to be present or defined if all requirements on that signal are met. For a robust design these include the availability of predefined noise margins on the signal levels [Hill, 1986]. Relevant parameters are defined in Figure 2. Maximum noise margins are obtained when the signal levels are above $V_{out(min)}(H)$ or below $V_{out(max)}(L)$. As a consequence, these levels are the only relevant levels the timing parameters refer to. As a consequence, the transition time of signals in between these levels has a direct influence on the timing parameters. Databooks use $0.5V_{DD}$ as a reference level for the timing parameters. In using this level, timing calculations resulting in propagation delays are far too optimistic. The result may be a critical timing of the circuit.

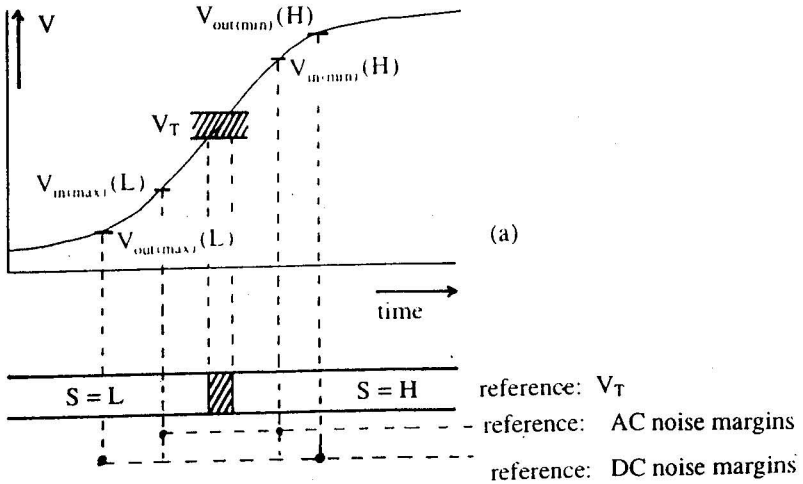

Figure 2. *Parameters for the definition of logic signals.*

When ICs from different logic families are used special attention must be given to the verification of the noise margins. In general, a mix of ICs results in smaller noise margins than within one family. The result may be a higher sensitivity to internal and external noise. So do not mix 5V and 3V circuits, even when they are said to be compatible.

Samsom [Samsom, 1996] has found that other mistakes may influence the timing specification as well. The setup and hold interval at the input of flip-flops are usually measured by shifting a rising or a falling edge through this interval. The setup interval starts at the point where the arrival time of a transition influences the behaviour of the flip-flop, e.g. the propagation delay and/or the dissipation. The same applies for measuring the hold time. Simulations have shown that measuring these parameters with a pulse of a minimum width extends the setup and hold interval with about 30%. In this case, the measurement method does not correspond to the predefined timing model, being an interval in which the input signal must be present.

Another example of an erroneous timing specification is when flip-flops and other components are characterised under so-called measurement conditions. With real logic circuits for fanout load other results are found, that deviate significantly from the specified timing parameters in component libraries [van Klaveren, 1997].
An accurate timing model is essential for the timing verification. In long paths in combinational logic a difference between the estimated parameters and the real signal timing may result in a critical timing of signals, with respect to setup and hold intervals of the flip-flops. This can result in intermittent errors, due to metastability [Kleeman. 1987]. In high-speed logic verification and testing of intermittent errors become more and more a problem. One way out is a better parameter verification. Using a lower performance level is another. Section 3 goes into the details of the minimum propagation delay

3 Timing of data transfer

A clock signal cannot be reproduced everywhere in a circuit at the same moment. Differences in propagation delays of clock buffers, propagation delays of clock signals and/or their transition times make that all flip-flops in a circuit do not recognise the clock pulse at the same moment. Flip-flops of a different internal structure may also have different trigger levels on their clock edge. All these results in so-called clock skew. Figure 3 describes the timing of a direct data transfer in between two flip-flops FF1 and FF2. From this figure we learn that there are two margins available to compensate for clock skew. Those margins must be greater than the maximum amount of clock skew in the circuit or board. The minimum margin necessary to compensate for clock skew is specified as $t_{skew(min)}$. It is important to define or estimate this parameter as exact as possible. Then

$$margin_1 = t_{P(min)}(FF1) - t_h(FF2) \geq t_{skew(min)}$$

$$margin_2 = T_{clock} - t_{P(max)}(FF1) - t_{su}(FF2) \geq t_{skew(min)}$$

In many modern flip-flops $margin_1$ becomes critical, what results in a hazardous data transfer. These flip-flops are too fast! Note that $margin_1$ is independent of the clock frequency. $Margin_1$, sometimes indicated as the parameter $t_{skew}(FF)$, defines the ability of a flip-flop design to compensate for clock skew! In designs with different types of flip-flops some of the available margins for clock skew are much smaller than in a design made in one type of flip-flops. Usually, a mix of flip-flops in a design is the finger print of an unskilled designer.
In a one-phase master-slave flip-flop the clock signal is directly applied to one of the latches. The inverted clock signal is applied to the other latch by an inverter. With the clock signal directly at the output latch the margin for clock skew, indicated as $margin_1$ becomes smaller than when the clock signal is applied to the input latch. When we indicate the flip-flop dependent margin for clock skew as $t_{skew}(FF)$, we find

$$t_{skew}(FF) = t_{P(min)}(\text{Z latch}) - t_{P(max)}(\text{INV}) - t_h(\text{Y latch})$$

with the external clock signal fed to the output latch Z and

$$t_{skew}(FF) = t_{P(min)}(INV) + t_{P(min)}(Z\ latch) - t_h(Y\ latch)$$

when the external clock signal is fed directly to the input latch. With the external clock signal directly on the input latch the margin for clock skew is significantly greater. This extra margin for clock skew is in conflict with the target of a high performance. The corresponding maximum toggle frequency of the flip-flop becomes lower. There is a conflict between performance and margins for clock skew. We conclude that in modern high-speed flip-flops the internal structure must be tested on margins for clock skew and, if necessary a balance must be sought between margins and performance.

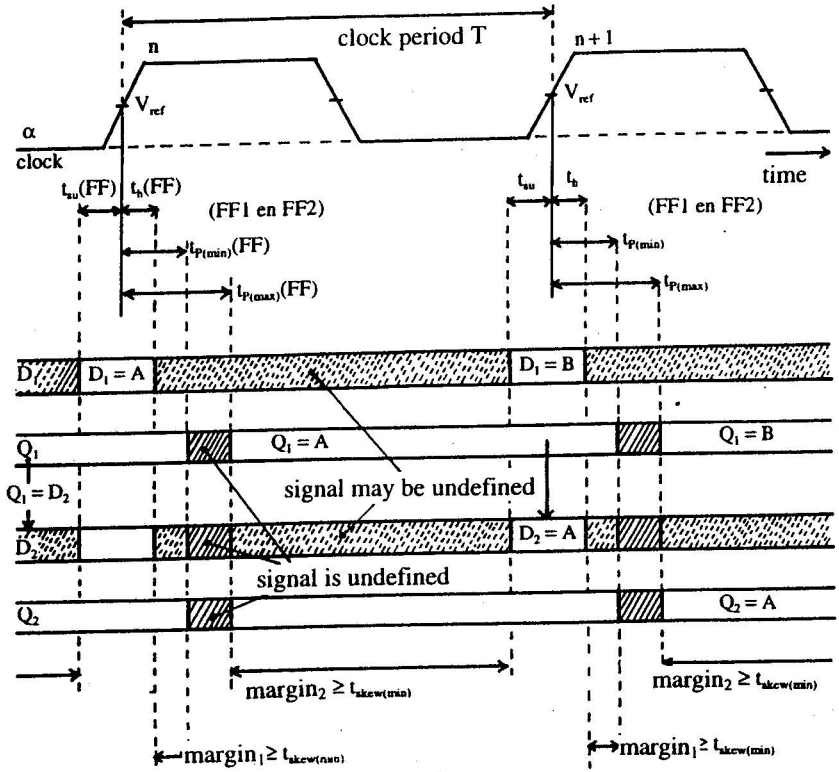

Figure 3. *Direct data transfer between flip-flops.*

Data lock out timing is an alternative. In a data lock out timing the flip-flop reads data at one clock edge and displays data on its output at the other edge. In direct data transfer both margins for clock skew become dependent of the clock period. As a consequence the minimum margin becomes greater. This principle has been used in the definition of the Boundary Scan Standard [Maunder, 1990]. Another solution is using a two-phase clock system, in which the two phases can be shifted

independently. In this way a designer can create any margin for clock skew. The price is, again, a lower performance.

4 Robustness of flip-flops

The input timing of flip-flops can be classified in positive or negative edge-triggered, positive or negative pulse-triggered, period-triggered and asynchronous. With an edge-triggered timing system the input data of a flip-flop must be present in a setup and hold interval, that lies around one of the edges of the clock pulse. In the remainder of the clock period T the input data may be undefined. (Note that in CMOS a reduction of the power dissipation may ask for additional requirements on the timing.)

In a pulse-triggered timing system the input data must be present during the pulse of the clock signal, or during the pause. A well-known example of a pulse-triggered timing is the classical master-and-slave J-K flip-flop with internal feedback. The feedback network disables the J or K input gate. The effect of a noise pulse on the active input cannot be compensated for by the signal on the other input, as this gate is disabled.

We classify a flip-flop timing system as period-triggered if for at least one of its input combinations the data must be present during the whole clock period. Figure 4 shows a construction that is found in some CMOS circuits. The edge-triggered CMOS flip-flop has a transmission gate in the input latch. This flip-flop can be set to read data from one out of two inputs. When both input transistors are switched off, the inhibit mode, the flip-flop reads the previous data. In fact, the load at the input node is every clock cycle refreshed by the flip-flop itself. In transistors, this is a cheap solution for an input selector with inhibit mode.

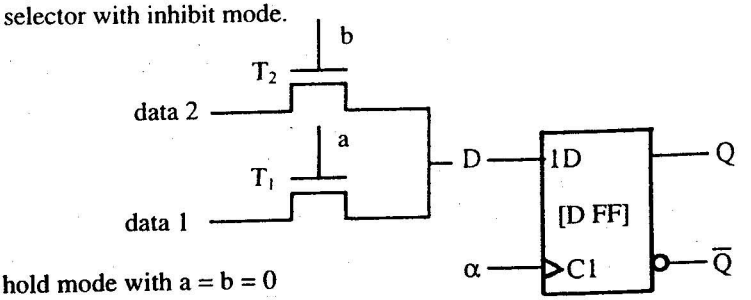

Figure 4. Example of a period-triggered timing system.

An asynchronous timing is a period-triggered timing, with the extra feature that the output of the flip-flop reacts directly when a so-called asynchronous input is active. When an asynchronous input must be disabled, the signal level for disable must be present during the entire clock period. Compare asynchronous resets on counters and registers.

Output timing is classified as direct acting (Q output reacts on the first or same clock edge as the data), as postponed (Q reacts on the next clock edge) or the output timing is asynchronous (Q reacts immediately and asynchronous with the clock signal). A

data lock-out timing, as is found in Boundary Scan for the scan line is called a postponed output timing. The input timing of a data lock-out flip-flop is always edge-triggered.
Note that the timing classification should be done worst-case. If for instance one of the input combinations has an edge-triggered timing and the others are pulse-triggered, then the entire circuit is said to have a pulse-triggered timing.

The choice for a timing system cannot be done without any consequences for the robustness of a circuit or system. Gito [Gito, 1981] has found that the error probability of asynchronous inputs on printed circuit boards is in the order of one error every 10^8 through 10^{12} clock cycles. He built a hardware monitor to detect those errors. In this way a great number of experiments could be done. All asynchronous inputs on his circuits (TTL logic) were detected as being tricky! The circuit of Figure 4 (CMOS logic) also proved to be susceptible to noise during testing [Beenker, 1994]. Evidently, a pulse-triggered timing is susceptible to noise as well, though the error probability may be somewhat lower. So there is absolutely no alternative for an edge-triggered input timing. The present-day practise still to copy the old TTL counters for example in CMOS, including their asynchronous inputs, cannot be understood taking into account the recently met EMC problems.
Any logic type of flip-flop can be realised with an edge-triggered timing or with an edge-triggered timing with data lock-out [Thijssen, 1998]. The only condition is that, during the enable phase, for all input combinations the input latch of a flip-flop directly follows the input data. So there is no need to use other timing systems. Sometimes this strategy requires more transistors for a flip-flop implementation. The price for a correct design. (By the way, nobody discusses the application of extra test hardware in digital circuits any longer.)

Another serious bug has been found in the output latch of many CMOS flip-flops. Most flip-flop designs stem from a period that elementary insight in timing and robustness did not exist. It is not permitted that a pulse at the output of a flip-flop may change the internal state of that flip-flop. See Figure 5. Whenever the memory loop of the output latch can be set or reset with an external pulse, the resulting circuit is susceptible to noise. The realisation of Figure 5.b does not have this problem.
When, for example, flip-flops are embedded deep in a register array, the probability of noise pulses in between two flip-flops is extremely low. The above problem does not exist. For flip-flops driving an output pin of an IC you are absolutely sure to have problems. The real truth is always in between. But, with a circuit designer having a poor electrical background, types of flip-flops having this bug should not be available in a library!

Conclusions

In a modern high-speed and/or high-performance CMOS environment a logic designer should pay more and more attention to the mapping of a logic design into silicon. Most problems, but not all can be avoided by clearing libraries in CAD systems and a better

characterisation of the timing parameters. Experiments have shown that with a good design style the yield of complex ICs can be improved dramatically. A very short time-to-market is another advantage. This compensates very much the price of a few extra transistors.

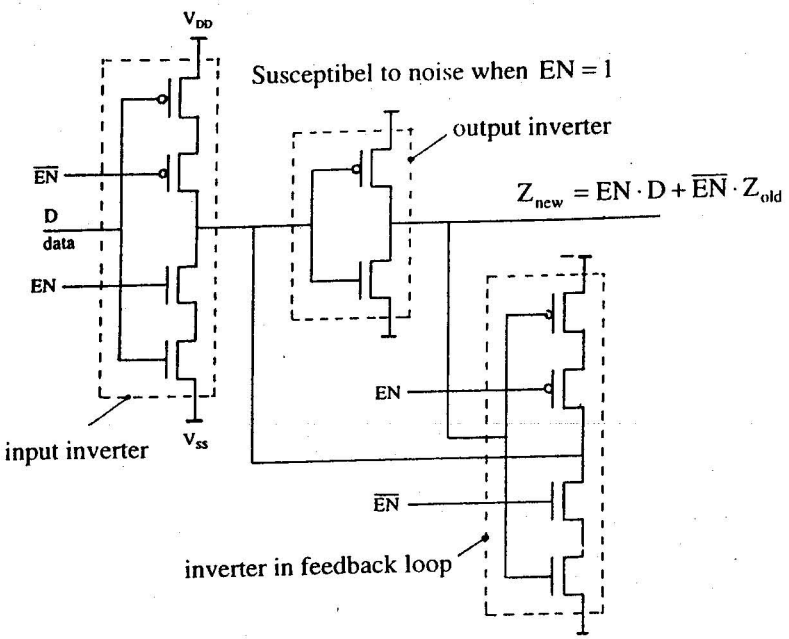

Figure 5. Flip-flop with an output problem.

References

F.P.M. Beenker, R.G. Bennetts, A.P. Thijssen, Testability Concepts for Digital ICs, Kluwer Academic Publishers, Dordrecht, 1995.

F.P.M. Beenker, Philips Research, The Netherlands, private communication.

J. Gito, Design of a clock generator, Master's Thesis Delft University of Technology, 051560-27(1981)05, Delft, The Netherlands

C.F. Hill, Noise Margins and Noise Immunity in Logic Circuits, Microelectronics, 1986, pp.16-21.

J. van Klaveren and A.M. Noordam, private communication, 1997.

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

C.M. Maunder and R.E. Tullos, The test Access Port and Baundary-Scan Architecture, IEEE Press, Los Alamitos, 1990.

S. Samsom, Clocking of Digital Systems, Master's Thesis Delft University of Technology 1-68340-28(1993)24, Delft, The Netherlands.

Loek Thijssen and Frank Bouwman, Redundant States in Test Control Block Design, IEEE European Test Conference ETC93, Rotterdam, April 19-22, 1993.

A.P. Thijssen e.a., Digital Circuit Design, to be published in 1998.