
Clocked Elements

James Warnock

IBM T.J. Watson Research Center

3.1 Introduction

One of the most critical design considerations during the planning of any large VLSI structure is the definition and implementation of the clocked storage elements (CSEs) and the circuitry which drives the local clocks to these elements [1–4]. The nature of the solutions adopted will affect almost every aspect of the design, including its manufacturability, testability, reliability, power consumption, and operating frequency, while the complexity and style of latches and flip-flops employed will affect almost every design automation tool, from high level logic simulation methodology and logic synthesis engines, to circuit tools for detailed device tuning, timing, and testability analyses. A modern microprocessor chip may contain from 0.75 to 1.5M latches and flip-flops [5, 6], and clocked elements may account for 30–40% of the total chip AC power dissipation [7, 8]. Furthermore, the delay overhead or latency of these elements is typically in the range of 2–3 FO4 for modern high-speed designs [5, 9] which may account for 10–25% of the design cycle time for designs spanning the range from 10 FO4 (performance-only optimization) up to about 30 FO4, typically the upper end of the range for power performance optimized designs [10]. Thus the CSE definition is of fundamental importance to any VLSI project; the correct selection, optimization, and implementation will be a basic part of the global design strategy. The goal of this chapter is first to provide a high level overview of the design space of these elements covering the basic design metrics, issues, and trade offs, and second to look at several families of CSEs. This will be followed by a more detailed discussion on aspects of test and testability, design robustness against variability, reliability and soft error rate (SER) considerations.

3.2 CSE Design Issues

In this section some of the basic CSE design issues are discussed, including latency, hold time, power, and scan testing. This foundation will serve as a useful reference when describing the design space covered by the various CSE families.

3.2.1 Latency

One of the primary considerations of any digital design element is its delay, ideally characterized and compared at a constant, fixed power dissipation. For CSEs, the definition of latency is a little more complicated, due to the interaction of the data with the clock edge. If the data arrives at the CSE well before the activating clock edge, the time delay between when the data arrives at the input (labeled for reference as “ d ”) to when it is propagated to the output (labeled for reference as “ q ”), will be very long since the data has to wait for the arrival of the clock before it may propagate to the output of the CSE. The total time from d to q will thus decrease linearly as d arrives later (as the wait time is reduced). However, a point will be reached where, if the data arrives too late, the time from clock arrival to the output transition (q) starts to increase, as the point is approached where the data arrives too late to be captured or propagated to q . Therefore, it can be seen that, when measuring the latency of the CSE there will generally be some optimum specification for the data arrival time relative to the clock edge, which produces the minimum overall delay [11], and this optimum point may be used for design comparisons. Since the clock arrival time typically marks the cycle timing boundary, the total $d - q$ delay is often broken into two segments describing the behavior on either side of the boundary; the setup time (amount of time that the data needs to arrive in advance of the clock), and clock-to- q delay (time measured from arrival of the clock and the appearance of the new data at “ q ”).

Having described the conditions under which CSE latencies may be compared, it should be noted that the shape of the latency curves described above is also an important design consideration. This effect is illustrated in Fig. 3.1, for two hypothetical CSEs, plotting the time from d to q vs. the d arrival time relative to the clock. It can be seen from this example that “design A” has a lower latency than “design B,” since it offers the minimum latency (d to q time). However, this design (A) exhibits a “hard” cycle boundary behavior. Data cannot arrive much later than at the optimal setup time point before the element will fail, and getting the data to the latch early will provide no benefit to the timing of the downstream logic on the next cycle. Design B exhibits a “soft” cycle boundary. The data arrival time can be targeted to fall in the middle of the “window” where the latency is minimal. Provided that the data arrives somewhere in the window, earlier arriving data will show up at q earlier (since the $d - q$ latency is about constant), to the benefit of the downstream logic on the next cycle. Later arriving data will not cause a failure until the data falls outside this “transparent” window, or until the timing impact on the downstream logic causes a failure.

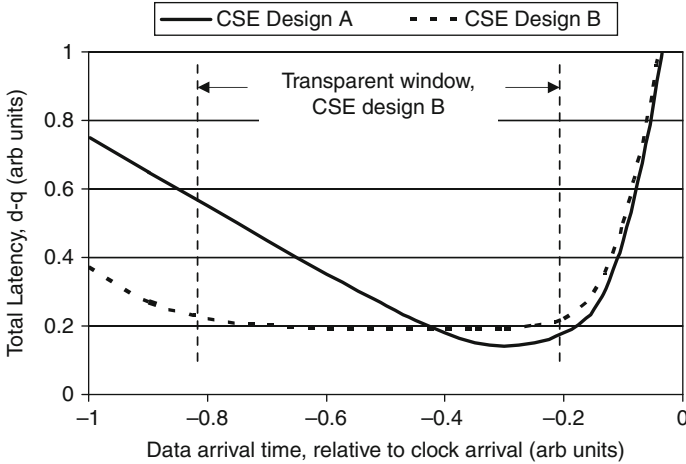


Fig. 3.1. Latency ($d - q$ time) vs. data arrival time for two hypothetical CSE designs

This concept of “soft” vs “hard” cycle boundaries is very important for CSE design, and particularly as technology variability increases, a simple latency metric is insufficient to characterize a particular CSE style. In many situations, with all other things equal, a design like “B” may be preferred over “A,” and may actually give better results in hardware, even though the absolute latency is worse.

3.2.2 Hold Time

Just as the data arriving at the CSE must arrive early enough to get properly written, the data must also hold its value for some amount of time (relative to the clock), if the proper value is to be written. Hold time errors may result, for example, from a situation where new data arrives very quickly at the beginning of the cycle, overwriting the data that was supposed to be captured and latched at the end of the previous cycle.

Hold time specifications are not typically considered when comparing power performance characteristics of CSEs. However, long hold times may drive the addition of a substantial number of delay-padding elements to eliminate any possible early transition at the CSE input. Thus indirectly, long hold times may cost both area and power. In addition, the amount of effort devoted to detection, analysis, and the fixing of hold time misses reduces resources available for other forms of design optimization. Finally, design solutions with longer hold times are inherently riskier, and will be more prone to unexpected hardware issues. In many designs there is a natural trade off between the hold time and the size of the transparent window at the cycle boundary. As the window is widened (making the timing on cycle-limiting speed paths less

sensitive to clock skew and process variation), the hold time also increases (making the design more prone to clock-skew-induced or variation-induced functional failure through hold time misses).

3.2.3 Power

Power is often considered in conjunction with latency, since the two metrics typically have an inverse dependency on each other. There have been many comparative power performance studies of various latch and flip-flop families [3, 9, 11–14], but the analysis is generally not as straight-forward as it may seem at first. In the same way that the action of the clock signal complicates the latency considerations, it also complicates power comparisons as well. In a given design, the total AC power dissipation is the sum of the power dissipated by clock nets, and other nodes regularly charged and discharged by the clock, and the nodes that switch whenever the input (and output) data change. Thus, when considering a particular latch or flip-flop solution for a specific design application, it is important to characterize the design with the correct weighting between clock switch events and data switch events.

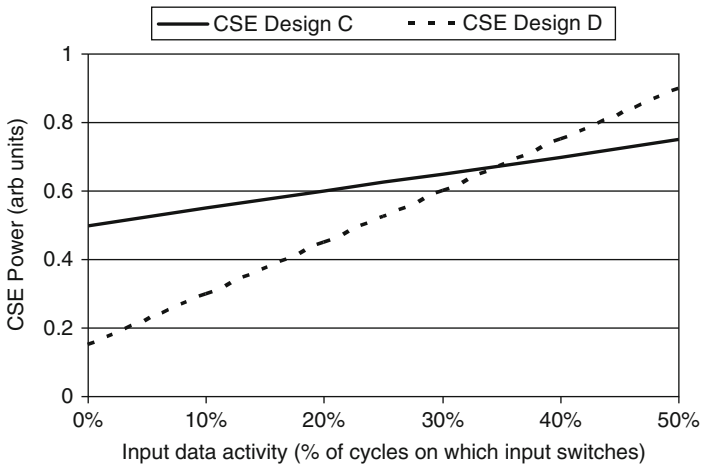


Fig. 3.2. Power vs. input data switching factor for two hypothetical CSE designs

This is illustrated in Fig. 3.2, for two hypothetical CSE designs “C” and “D.” Design “C” has a relatively high capacitance on clock nets and/or nodes which charge and discharge every cycle independent of the input data, but uses this dynamic action to limit the amount of capacitance switched by changing data inputs. This sort of behavior is typical of many dynamic CSE designs. Design “D” has much less clock capacitance, but has more capacitance switched when the data changes state. Power comparisons at high data switching factors, for example at 50%, would tend to favor the dynamic design “C,” while comparisons at lower switching factors, for example 10–20%, would tend to favor design “D.” Complicating the situation further is the

fact that in modern microprocessors, the local clocks are gated off whenever it is known enough ahead of time that a group of CSEs does not need to be updated on a given cycle. Thus for power comparisons, it is necessary to understand the average data switching factor on cycles when the clock is active. Depending on the type of CSE, it may also be necessary to consider the power dissipated by data switching activity when the clock is gated off as well.

At this point, it is worth mentioning that the ideal situation, with regard to power consumption, would be to activate the clocks to a given CSE only when the new input data is different from the old data, and there have been several schemes proposed along these lines [12, 15]. However, the power, area, and delay overhead associated with determining whether or not to fire the clocks may be significant, and these techniques have never been widely deployed.

3.2.4 Scan Design for CSEs

Given the complexity of modern VLSI systems, the ability to directly observe the contents of critical system storage elements during design test and debug, as well as the ability to apply arbitrary initialization patterns to these elements, is another important design consideration. Typically this is done by linking the CSEs together in a serial fashion, using a secondary input port as shown schematically in Fig. 3.3. Level sensitive scan design (LSSD) [16] is one methodology which accomplishes this goal, although there are other more general ways of providing scan capability in a design.

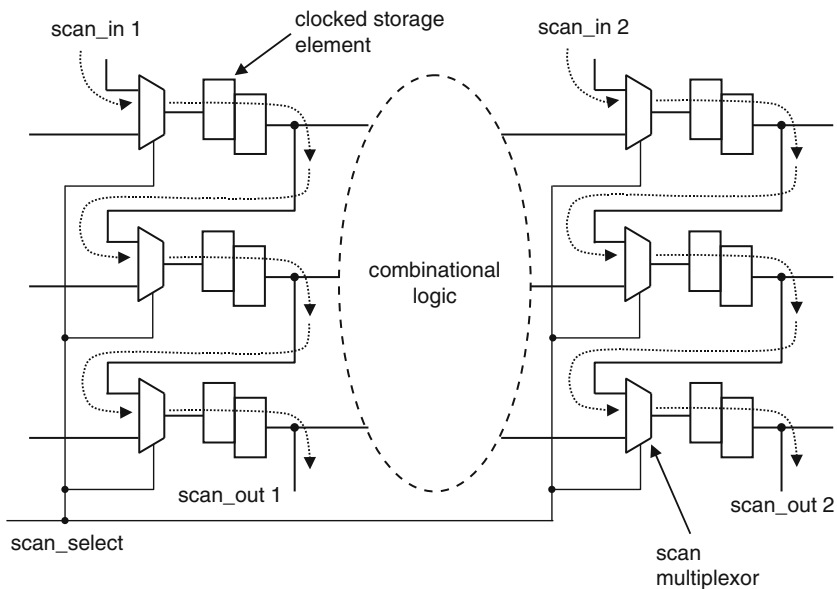


Fig. 3.3. Basic scan design. Scan data flow is indicated by the *dotted lines*

The key features of a scannable design include the scan multiplexor, to allow choosing between scan and normal functional data to load into the CSE, and the scan chain itself which needs to be configured in the style of a shift register to allow movement of the data in and out of the CSEs in an orderly fashion. In practice, the goal is usually to minimize the overhead of the extra logic needed for scan and test functionality, without sacrificing the desired test features; for example the multiplexor in Fig. 3.3 is often integrated directly into the CSE itself. Some of the detailed requirements, and ideal test features are described later in this chapter. In the CSE discussion that follows, for each design style, the implications and overhead associated with scan functionality will be described. In modern VLSI designs, it should be assumed that a significant fraction of CSEs will need to be scannable, and therefore this is a significant design consideration.

3.3 Static Latch Designs

In this section, examples of several basic styles and classes of static latches will be presented and discussed, including a discussion of the local clock generation aspects, comparative merits and general features of each of the families in light of some of the design considerations described earlier.

3.3.1 Master–Slave Latches

For general logic design, the master–slave latch (MSL) is probably the most widely used CSE, with many topological variants and clocking styles described over the years. A typical transmission-gate master–slave latch [17] is shown in Fig. 3.4, using the style and naming conventions of reference [5]. The two clocks, “dclk” (or “data capture clock”) and “lclk” (or “launch clock”) are roughly 180° out of phase. At the beginning of a cycle (cycle boundary), dclk falls while lclk rises, simultaneously blocking any writes to the master latch while writing the slave latch and propagating

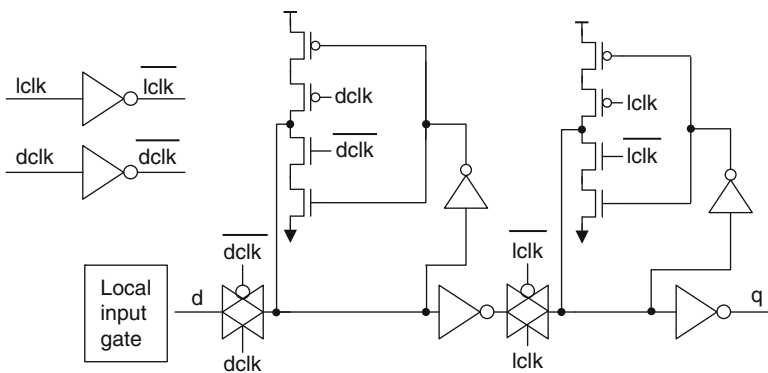


Fig. 3.4. Master–slave latch. Reproduced with permission from [5], ©2006 IEEE

new data to the latch output q . At the middle of the cycle, $dclk$ rises while $lclk$ falls, allowing the master latch to accept new data until the end of the cycle, while blocking any update of the data in the slave latch so that the data is held until the beginning of the next cycle.

In high-speed designs, the $dclk$ and $lclk$ waveforms are designed to overlap (both clocks high) by some amount at the cycle boundary in order to allow late-arriving data to propagate through the MSL with minimal delay by avoiding interference from the clock edges. This technique not only minimizes latency [5], but also provides for soft cycle boundaries; if the data is timed to arrive in the middle of the window when both clocks are high, then later-arriving data can still propagate through, stealing time from the next cycle. Also, the MSL becomes tolerant to some amount of local clock skew depending on the width of this window, and to the extent that the data arrives in the middle of the window, away from the clock edges. The trade off though is that as the falling $dclk$ is delayed to improve latency and widen the window of transparency, the hold time increases, creating race hazards, and requiring delay padding on short latch-to-latch paths. This trade off is a fundamental consideration for MSL operation, and is shown in Fig.3.5 [5]. Here the delay unit “FO4” refers to a technology independent delay metric, the delay of an inverter with a fan out of 4 [18]. It should be noted here that since the latency of the MSL is highly dependent on the relative timing of the master and slave clocks, any power performance comparisons including MSL elements (often cited as a standard benchmark in comparisons), must provide detailed information on the clocking assumptions made, since otherwise any latency quoted is not very meaningful.

Delaying $dclk$ with respect to $lclk$ also improves the situation at the middle of the cycle. Here, the desire is for the two clocks to be well un-overlapped, with $dclk$ rising well after $lclk$ falls in order to avoid potential race-through of the wrong data at mid-cycle.

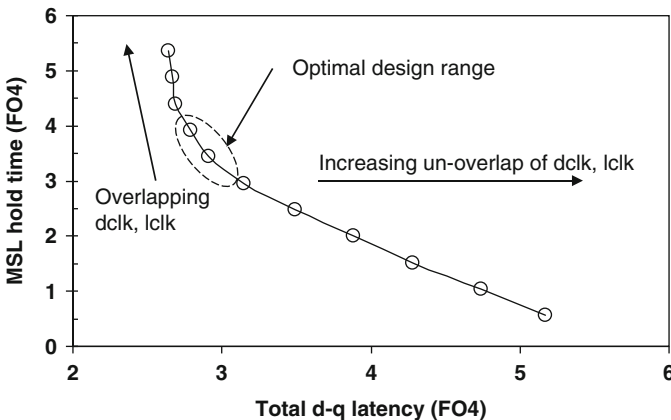


Fig. 3.5. MSL hold time vs. total MSL latency as the cycle boundary overlap of $dclk$ and $lclk$ is varied. Reproduced with permission from [5], ©2006 IEEE

It should be noted here that there are many other MSL circuit topologies, some of which are inherently more robust against race or min-path issues [19]. However, these will generally be higher-latency solutions, and by their very nature will not possess the soft cycle boundary properties which provide protection against delay variability and clock skew. For designs with long cycle times (for example, cycle times larger than 40–50 FO4), it may be more important to have a simple, robust solution, than to get the ultimate latency, or optimal power performance. It is expected though, that modern high-speed microprocessors will continue to need the more difficult design solutions, especially as technology variability (and the benefit of soft cycle boundaries) continues to increase. Increasing sophistication of design analysis tools and techniques will help designers to deal with these more complicated solutions.

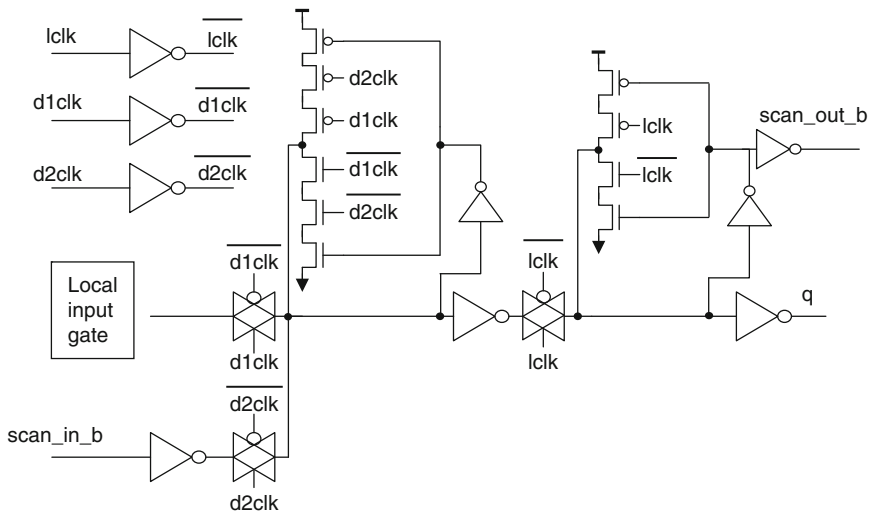


Fig. 3.6. Scannable MSL. Reproduced with permission from [5], ©2006 IEEE

The MSL shown in Fig.3.4 has no provision for scan testing. One of the most common ways of adding this functionality is to add another port to the master latch, controlled by its own clock. This second port may be controlled by a symmetric twin of the first dclk, as shown in Fig.3.6. In this case the MSL design can support scan shifts and test sequences at frequencies up to the functional design frequency, although other considerations (such as power, and noise) may often impose practical limits on the scan shift frequency. A set of clock waveforms is shown in Fig.3.7, illustrating the shifting of data through the scan chains, and functional operation. This type of design solution supports a seamless transition from scan shifting to functional operation, and vice versa.

Other scannable MSL topologies have also been used. One alternate version uses a multiplexor on the slave latch feedback path [20] instead of directly at the input to

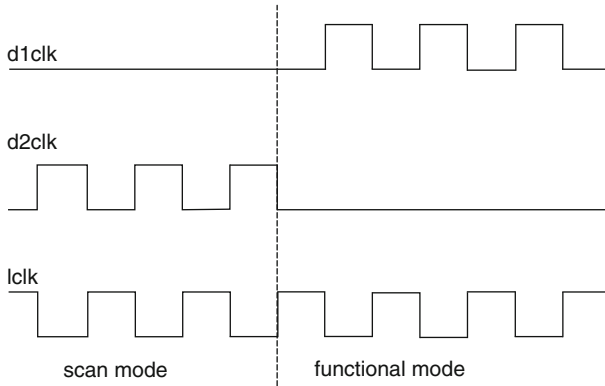


Fig. 3.7. Sample set of clock waveforms for scan shifting and for functional operation, for the MSL of Fig.3.6

the latch node, as shown in Fig.3.8. In LSSD methodologies, the second port to the slave latch is generally a low-speed scan clock with its own dedicated distribution network. Even with a low-speed scan clock, it is still possible to do full-frequency AC testing on the part. After scanning the desired data in to all scannable latches and appropriately initializing any nonscan elements, the system can be started synchronously in functional mode at speed, with the first at-speed lclk launching out either the data from the last scan clock, or the functional data waiting at the data port input, depending on the type of test desired. Such LSSD schemes may provide very flexible and robust scan design solutions [21], even if the rate of data shifting through the scan strings is limited by the speed of the secondary LSSD clock distributions.

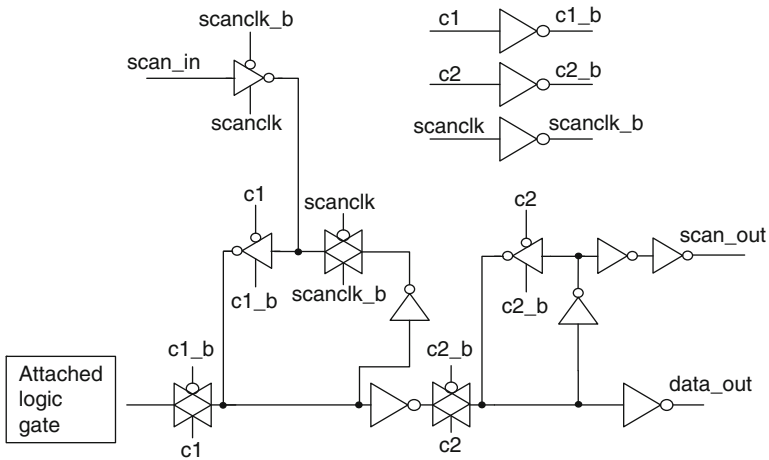


Fig. 3.8. MSL with scan MUX in feedback path. Reproduced with permission from [20], ©2002 IBM

Many VLSI designs have been implemented with MSLs as the main “workhorse” latch, often with sophisticated clock edge controls [22]. However, the variation-tolerant soft cycle boundary concept can only be taken so far because of the hold time increase associated with the transparent overlapped clocks at the cycle boundary. In addition, designers have looked for lower power solutions, or higher speed solutions for specific critical applications. These considerations have led to some of the styles described in the following sections.

3.3.2 Two-Phase Level-Sensitive Latches

It has long been recognized that the use of level sensitive latches, as shown in Fig. 3.9, will improve the clock-skew and delay-variability tolerance of the design, while simultaneously offering more flexibility for balancing logic delays across multiple cycles. This feature, or technique has been referred to as “cycle stealing” [23] since it allows circuits on either side of the latch to “steal” time from the previous or next cycle. In comparison with the MSL style described above, this technique can be conceptualized as the result of splitting apart some, or all of the master and slave latches, with logic interspersed between master and slave, as well as between the slave and master. Any path through the logic will pass alternately through level sensitive latches clocked with opposite phases. It can be shown that, with about one-half of a cycle of logic between each latch, the data arrival can be timed such that data arrives at each latch in the middle of its active clock window. Therefore, this scheme can be made robust to clock skew and local delay variability. It can also be made very robust against hold time issues, by shrinking the active phase of each clock, with no latency penalty as long as the data is timed to propagate through the latches near the middle of each active clock phase.

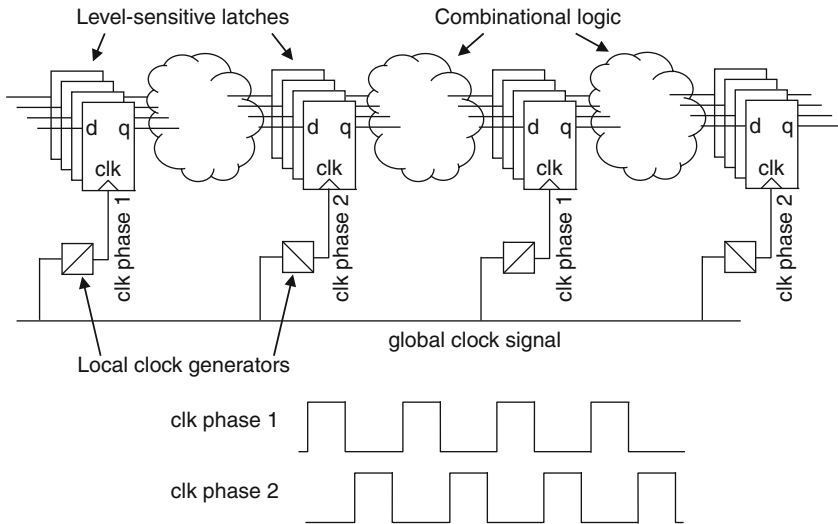


Fig. 3.9. Basic two-phase level-sensitive latch scheme, with sample local clock waveforms

This scheme would seem to offer the most tolerance to process variability and clock skew, and in fact these techniques have been used in some large microprocessor designs [20, 24]. From a power perspective, it would seem that the power dissipated should be comparable to, or somewhat less than that of a design using an MSL methodology, since the extra flexibility offered might help to cut down the overall number of latches needed.

However, there are some negative aspects of this design style which must be weighed in the balance. The timing analysis for split-latch designs can be very complicated, with long multicycle (or multihalf-cycle) critical paths, timing loops, and a general inability to define any clear cycle boundaries [25–27]. In fact, the more optimal the design (in terms of lining up data arrival at the latch with the midpoint of the appropriate clock phase), the more complicated the timing analysis becomes. In addition, the split master and slave latches cannot generally share any of the overhead for test control and clock gating in the generation of the local clocks, and will usually need separate global clock taps. The split latches themselves may impose a certain overhead on the design, especially if small devices are used in the latches in order to save clock power. Each latch may become a “bottleneck,” requiring stages of gain afterwards to drive the surrounding logic, or, to avoid this situation, clocked devices may have to be increased in size relative to that of an MSL design.

Finally, there are some test issues that need to be considered. Each scannable split-latch requires an additional scan-latch in order to build a robust scan chain, as shown in the example in Fig. 3.10 [20]. Also, AC test sequences need to consider the

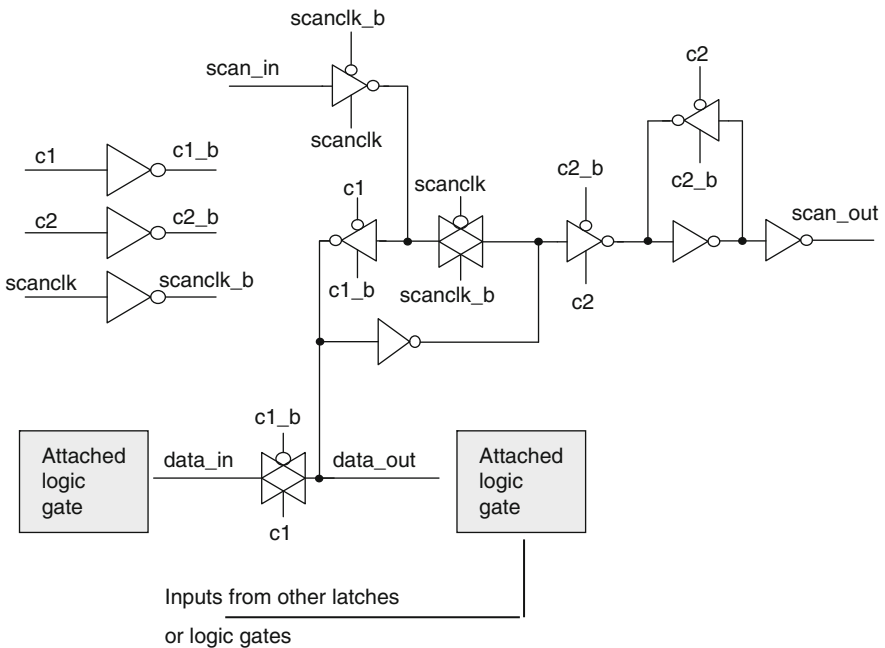


Fig. 3.10. Scannable level-sensitive latch. Reproduced with permission from [20], ©2002 IBM

implications of having the scan data propagating into the downstream logic as soon as the clock fires. This means that the initial timing of signals launched from the level-sensitive latch during test is likely to be different from that during functional operation, when transitions may flush through later in the cycle. Therefore AC test sequences may become more complicated, with several functional clock cycles required.

The above design and test issues probably explain why this design style has never gained widespread acceptance for large, high-speed microprocessor designs, even as design for variability tolerance becomes an increasingly significant consideration.

3.3.3 Pulsed-Clock Static Level-Sensitive Latches

The use of static level-sensitive latches with single-phase pulsed-clocks appears to be gaining favor for use in large microprocessor designs [5, 8, 22], even as device variability and presumably the hazards associated with hold time issues have been increasing. It is worthwhile to study this particular CSE implementation in order to understand the benefits and hazards associated with it.

An example of a very simple static pulsed level-sensitive latch is shown in Fig. 3.11 [5]. This style has minimal overhead, whether measured in terms of latency, power dissipation, or area. The latch itself is very simple, but the clock design is more delicate. A locally-generated, self-timed clock pulse is generated from either a global or local clock at the cycle boundary, and is used to write the latch (or, more generally, a group of latches). The width of the pulse must be wide enough to reliably write the latch across the full process/voltage/temperature (PVT) space, including all margins for local variability, noise, and any modeling uncertainties. However, the wider the pulse, the longer the hold time and the more difficult it becomes to protect against min-path or hold time failures. Therefore, careful design of the pulse-generating circuitry is of utmost importance, and the pulse width is a critical design parameter.

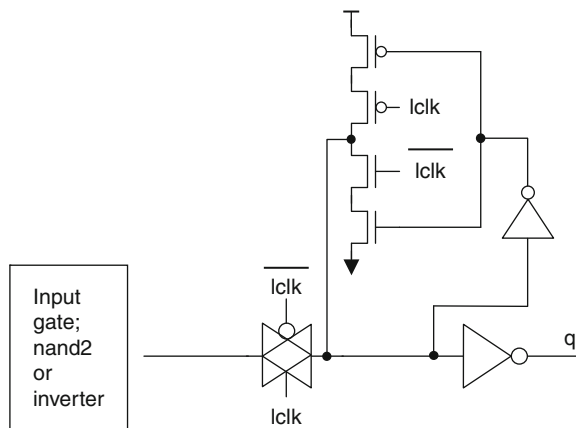


Fig. 3.11. Simple nonscan pulsed-clock latch. Reproduced with permission from [5], ©2006 IEEE

Despite the issues associated with pulsed-clock operation of these latches, there are several significant advantages with this design style. Reduced power consumption is one important benefit. In contrast to the conventional MSL solution, or the 2-phase level-sensitive scheme, only one active clock is needed, cutting clock power by about a factor of 2. In fact, this may be the most important benefit of pulsed clocking. One power-saving strategy along these lines is to use regular MSLs, but force the master clock to stay high all the time, while pulsing the slave clock [5, 22, 28]. This approach delivers significant power savings while preserving a fall-back strategy to conventional two-phase clocking in case problems are seen in the hardware. However, this approach does not provide the benefits of reduced area and latency offered by dedicated pulsed-clock latch solutions.

Another important feature of the pulsed latch is its ability to provide a soft cycle boundary, similar to that of the MSLs with overlapped clocks. Unlike the MSL situation though, the amount of cycle boundary transparency cannot be tuned below a certain value (set by the minimum pulse width needed). Thus, even as the benefit of such a pulsed-clock soft cycle boundary will increase as the technology variability increases, this benefit will be offset by the difficulty of simultaneously ensuring writeability and maintaining margin against data races. The design cycle time figures prominently in this trade off. The tolerable upper limit for pulse width would be expected to scale with the overall logic depth, with a practical maximum pulse width of probably about 1/4 to 1/3 of the design cycle time. With wider pulses, the difficulty of padding all the potential short data paths, while avoiding any impact to the cycle-limiting paths, is likely to become unmanageable. At a constant design cycle time (as measured in FO4) increasing technology variability may push pulse-widths up to their practical limit, necessitating larger devices (with faster write-speed but more power dissipation) or other design restrictions for continued use of pulsed-clock latches.

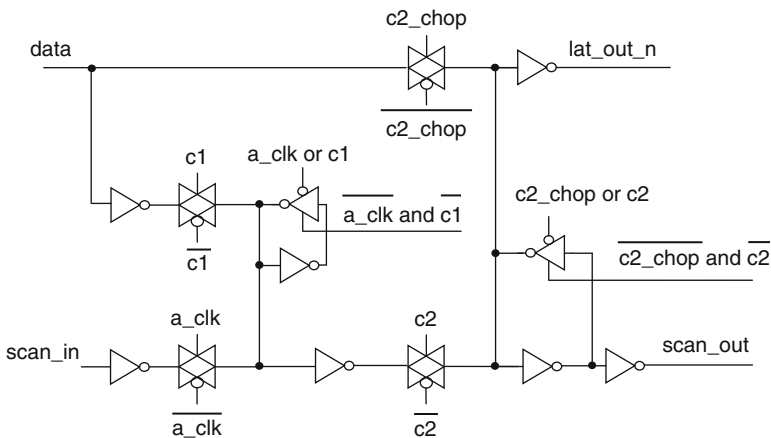


Fig. 3.12. Scannable pulsed-clock ($c2_chop$) latch with built-in MSL-mode fallback (using $c1$ and $c2$ clocks). Reproduced with permission from [28], ©2007 IBM

There are some special test issues for pulsed-latch clocking. In general, it would not be desirable to have to pad the scan path, so usually an additional scan latch is needed. This makes the serial shift operation along the scan chain look like a series of MSLs, whose clocks may be unoverlapped as desired to avoid any possibility of race issues during scan. Thus most of the area advantage of the pulsed-clock solution is likely to be lost in situations where a scannable design is needed. An example of a scannable pulsed latch is shown in Fig. 3.12 [28]. This particular design achieves a low latency typical of dedicated pulsed-clock latches, but still maintains the ability to revert back to an MSL mode of operation if problems develop, although at a reduced performance level. In this sense, the extra scan latch added not only converts the scan chain into a series of MSLs, but also can provide a similar function on the data path as well, if conditions warrant.

3.4 Flip-Flop Designs

Flip-flops are inherently edge-triggered designs, as opposed to the latches described in the previous section, which are inherently level-sensitive structures. There are a large variety of flip-flop designs which have been employed over the years, and it would be impossible to cover all of them here. This section will describe some of the more common styles that have been used, comparing some of the design issues and the merits of each.

3.4.1 Sense-Amp Style Flip-Flop

The proto typical sense-amp flip-flop (SAFF) is shown in Fig. 3.13 [29–31]. This type of design consists of a sense-amplifier coupled with a slave latch to hold the output data when the sense-amp is being reset. This style is more naturally suited for use with dynamic logic on the input side, where data values will hold until they are reset, and where this data reset will occur at about the same time that the flip-flop itself is being reset. Furthermore, with dual-rail dynamic logic, if both true and complement inputs are precharged low, then the activating transition may arrive somewhat later than the clock edge, effectively borrowing time from the next cycle. Since there are internal nodes floating when both inputs are low, and the clock is high, the amount of time borrowing may be limited, depending on the process, noise, and environmental conditions. Also, with dual-rail logic the extra overhead for deriving the other data phase may not be an issue.

Given the generally high power dissipation of dynamic circuit styles, especially dual-rail, sensitivity to noise and process variations, and the general difficulties associated with using such circuits on a large scale in a modern microprocessor, it is unlikely that the industry will see a widespread resurgence in the use of dynamic circuits for general purpose processors. Therefore, this type of design should really be considered in the context of the static circuits which would usually surround it. In a static design, it can be seen that the critical path through the flip-flop traverses 4 stages of logic (the local inverter to generate the complement data, the pull down

of one side and subsequent pull-up of the output, and then the pull down of the other side), and so it would not be expected to have a very low latency. Also, power dissipation is likely to be an issue unless very small (and therefore slow, and more variable) devices are used, since one side or the other will need to be charged and discharged in each cycle that the clock is active, independent of the data input pattern. Furthermore, with static input signals, this design does not provide for soft cycle boundaries; whatever data is present when the clock edge arrives is written into the latch. The corollary though is that the hold time for this type of CSE is generally small.

Finally, it can be seen that if the input data were to change before the clock were to fall, internal nodes would be left floating, potentially resulting in a failure to retain the proper state at the output. This particular issue is relatively easy to remedy, by the addition of a single additional small NMOS transistor between nodes A and A' in Fig. 3.13, with gate tied to the supply [30]. This fix will, however, have some impact on the power/performance characteristics. The additional transistor not only adds device and interconnect capacitance to both sides of the sense amplifier (one side of which must be charged and discharged every cycle), but also partially discharges the nonactive side of the sense-amp on every cycle. In addition, there will be a slight degradation of the differential resolution due to the weak coupling between the opposite arms of the sense-amp structure, leading to a corresponding increase in setup time.

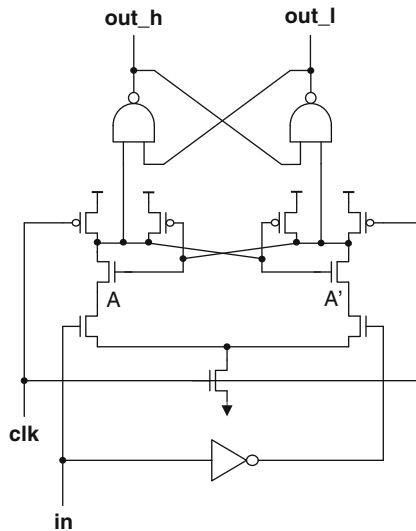


Fig. 3.13. Sense-Amp flip-flop. Reproduced with permission from [31], ©2006 IEEE

For improved testability, modifications have been described to make these designs scannable [32], an example of which is shown in Fig. 3.14. This technique can be optimized for AC test, since the paths from input to output are similar for both the functional data input and the scan input. The scan data input is implemented

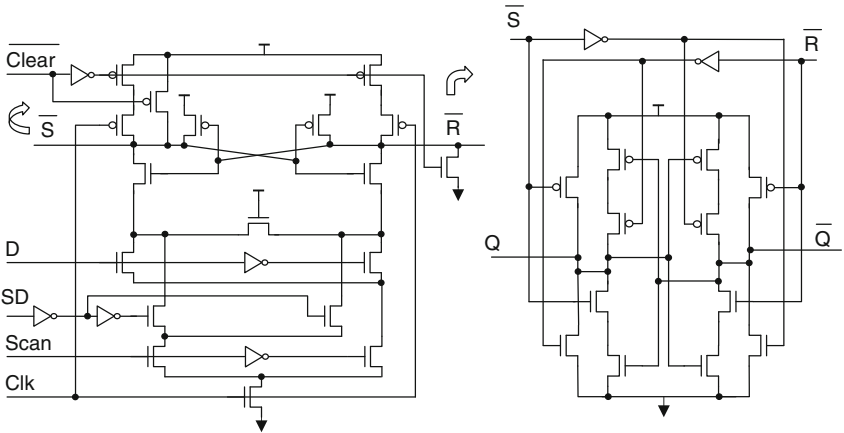


Fig. 3.14. Improved scannable sense-amp flip-flop, with asynchronous reset. Sense-amp “pulse-generator” on the *left* drives the latching stage on the *right*. Reproduced with permission from [32], ©2000 IEEE

as a second port in the first “pulse-generator” stage of the flip-flop. However, this will add capacitance to nodes which need to be precharged every cycle, thereby increasing power dissipation. Other methods [13] may avoid this power cost, but may be less well suited for AC test. In addition, there have been many modifications proposed to this basic design including techniques for “conditional precharge” to try to avoid charging and discharging of internal nodes when input data is not changing from cycle to cycle [33, 34] as well as other variations aimed at improving the power performance characteristics [32, 35, 36]. The resulting designs tend to be more complicated, and may still suffer from one or several of the drawbacks mentioned above. Therefore it seems likely that these SAFF will remain confined to only special-purpose applications in the future.

3.4.2 Hybrid Latch Flip-Flop

The goal of the hybrid latch flip-flop (HLFF) [37] was to try to combine some of the best features of flip-flops, including edge-triggered sensing, low latency, and low input clock load, with some of the best features of latches, including a soft cycle boundary. The result, shown in Fig.3.15 is a flip-flop-style front end, activated by what is effectively a locally generated clock pulse, with a static capture latch back-end.

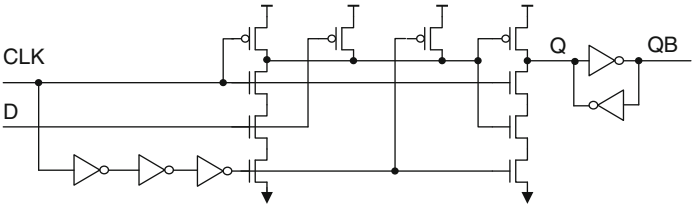


Fig. 3.15. Hybrid latch flip-flop. Reproduced with permission from [37], © 1996 IEEE

With only two gate delays from input to output, low latency can be achieved. Also, the local delay chain sets up a transparent window at the cycle boundary, allowing some amount of clock skew/delay variability tolerance. This particular design suffers from the fact that the static capture latch must be small enough to be quickly overwritten in either direction, meaning that the output will be very sensitive to any noise when only the capture latch holds the state of the output. In realistic applications, an additional local output gate would probably be needed. Also, this type of design will consume significant power, due to the precharge/discharge which occurs every cycle whenever the input data is high, and also the switching in the local delay chain inverters. Finally, the output is subject to glitching when the input data is high, since the output stage will begin to discharge when the clock edge is received, pulling back high only after the first stage output transitions low. Various improvements on this design have been reported [38, 39], but in general microprocessor designers have not found this type of flip-flop to offer any compelling benefits, at least in its original form.

3.4.3 Semi-Dynamic Flip-Flop

The term semi-dynamic flip-flop (SDFF) [40] was coined to refer to a design style which includes a dynamic front end followed by a static latch [5, 28, 41, 42]. In some ways, this is very similar to the HLLFF approach discussed above, but now expanded to provide a means of incorporating a stage of high-speed dynamic logic at each cycle boundary. This technique still provides an easy interface to surrounding static logic, accepting normal static inputs, and providing static outputs. A typical design is shown in Fig.3.16. These designs rely on some form of a pulsed clock to limit the hold time at the input dynamic stage. This may be accomplished by providing an explicit pulsed clock [28, 40, 41], or by ANDing in a delayed complement clock either in the dynamic pull down tree [40], or somewhere in the cone of logic for all the data inputs [5]. The dynamic stage is followed by a static set–reset latch (SRL), which holds the output while the dynamic stage is being precharged for the next cycle. For situations when the dynamic stage pull down path is cut off while the clock input is still high, a full keeper is usually used on the dynamic node in order to ensure that the dynamic node is held solidly low after being discharged. This keeper can be gated with the clock input to avoid drive fights during the precharge operation.

The advantage of this design is that it provides a way to incorporate a stage of dynamic logic into an otherwise fully-static design. It can also be extended to add additional dynamic logic stages after the first stage. In this case, only the first stage, with static inputs, needs a footer device and a pulsed evaluation clock. The SRL is moved downstream to the final dynamic stage, providing dynamic-to-static signal conversion. This technique can be applied at both clock edges [43], so that in such a two-phase system, pulsed clocks are no longer necessary.

The SDFF is a very useful construct for extremely critical paths in a design, where a wide OR, or wide multiplexor is required at the cycle boundary. However, there are some design costs associated with this solution. There is a relatively large capacitance which may be charged and discharged every cycle, even when there is

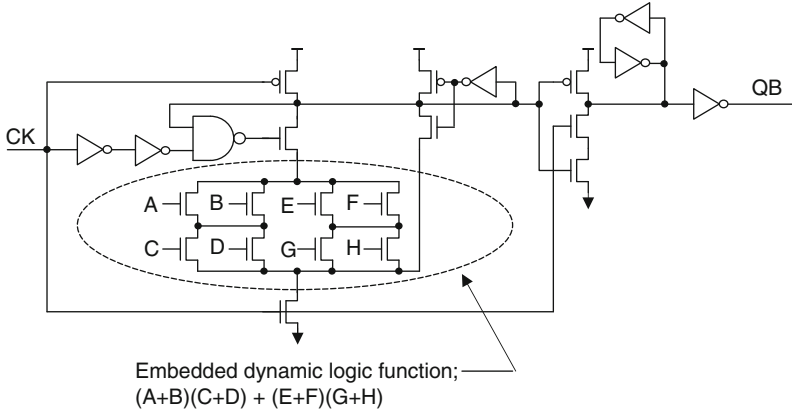


Fig. 3.16. Semi-dynamic flip-flop with embedded dynamic logic. Reproduced with permission from [40], ©1999 IEEE

no switching activity present on the data inputs. Thus, power consumption tends to be high for these designs, especially when used in contexts which do not see particularly high data switching factors, or when efficient clock gating is difficult. Power consumption may also be increased by glitching at the output when holding a 1. This glitch is similar in nature to that observed in the HLFF, occurring since the path to pull the output down (via the reset action of the static latch) is usually faster than the path to force the output high (via the dynamic pull down). The flip side of this glitch though, is that the latch will generally write a 0 faster than a 1, and downstream static logic can be skewed to take advantage of this fact (and also absorb the glitch).

The cycle boundary also is not easily softened to absorb timing variability, in contrast to the HLFF, due to the dynamic input stage. In principle, late rising input transitions may arrive at the dynamic gate after the clock rises, provided that they still have enough time to discharge the dynamic node, but falling transitions must meet a strict setup criterion. Similarly, the hold times for these designs tend to be relatively long, and are also asymmetric due to the action of the dynamic gate. Inputs only need stay high long enough to be able to discharge the dynamic node, but inputs initially low must stay low until near the end of the clock pulse. Just as for the pulsed level-sensitive latches, the clock pulse width must be wide enough to reliably write the latch across the process and application space of interest, but not so wide that the length of the hold time becomes an issue. For the SDFF, it is necessary to consider the range of dynamic input gates used in order to determine the minimum pulse width required.

For test, there are several strategies for making the design scannable [5, 28, 41]. One method involves adding a port into the static latch, and also adding in an additional scan latch, as was required for the static pulsed designs [28, 41]. An example of this approach is shown in Fig.3.17. With this scheme it should be noted that for AC test, the launch of scan data from the static latch bypasses the input dynamic stage, and so if the test sequence involves a launch from the scan clock,

the timing characteristics may differ from that in functional operation. It is possible to more closely match the functional path with the scan path by using an additional input into the front-end dynamic stage [5]. In this case, all the pull down paths through the functional data ports need to be disabled to avoid interference with the scan path.

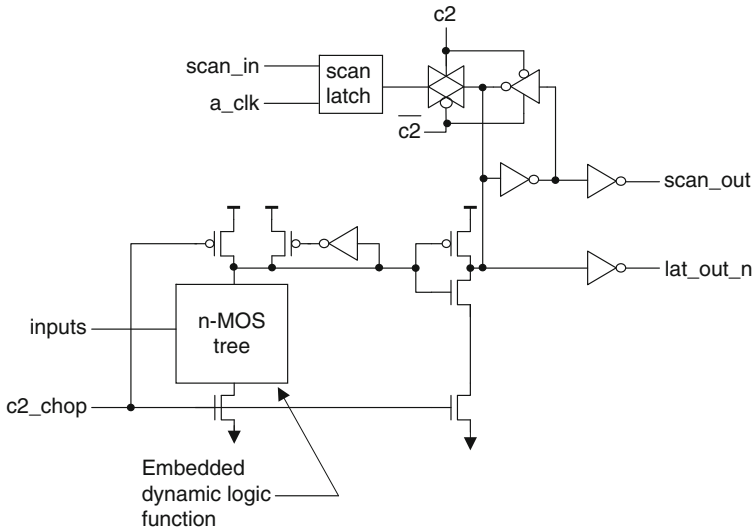


Fig. 3.17. Scannable semi-dynamic flip-flop. “c2_chop” is a pulsed clock input. Reproduced with permission from [28], ©2007 IBM

3.5 Test and Debug Considerations

As microprocessor frequency scaling has run up against severe thermal, physical and electrical constraints, the industry has turned to multicore architectures in order to allow continued performance gains from generation to generation [44]. As a result, microprocessor chips are now being fabricated with over 2 billion transistors [45], and with density scaling still expected to continue for at least several more generations in the future [46], observability and testability, including both DC and AC test coverage, have become of paramount importance. For this reason, large microprocessor design projects have by and large adopted scan design methodologies [5, 28, 41, 47, 48] and it is expected that scan design and test/debug methodologies will continue to need careful consideration in future designs. It will be important to minimize the overhead of these scan methodologies, but it will not be tolerable to make significant sacrifices to the testability of the design.

For DC test coverage, it may not be important exactly how the scan data is written into the latch, since the timing of the launch into the downstream logic is unimportant. Also, in the downstream logic, it is unimportant how each node is

switched; the activating transition is unimportant, as long as a pattern can be found which tests the node in each state. However, for high-speed AC test, there are a number of key features which are desirable for a robust and flexible test methodology. The highest test coverage is obtained if each CSE is designed with an extra scan-only storage element, as shown in Fig. 3.18a, used only for test purposes, which can store an independent data value to be used to launch a transition into the downstream logic [49]. In this way, each scannable latch can be independently configured to launch an arbitrary transition into the downstream logic. Referring to the initial input vector as “V1” and the transition vector as “V2,” it can be seen in this case that “V1” and “V2” are independent and unconstrained. This enhanced scan design provides the maximum test coverage, but the overhead is generally quite high. Although this technique can be confined to be used only on critical CSEs [50], and selective deployment has been reported in critical areas of some microprocessors [41], the cost is too high for techniques such as these to be used commonly in the industry.

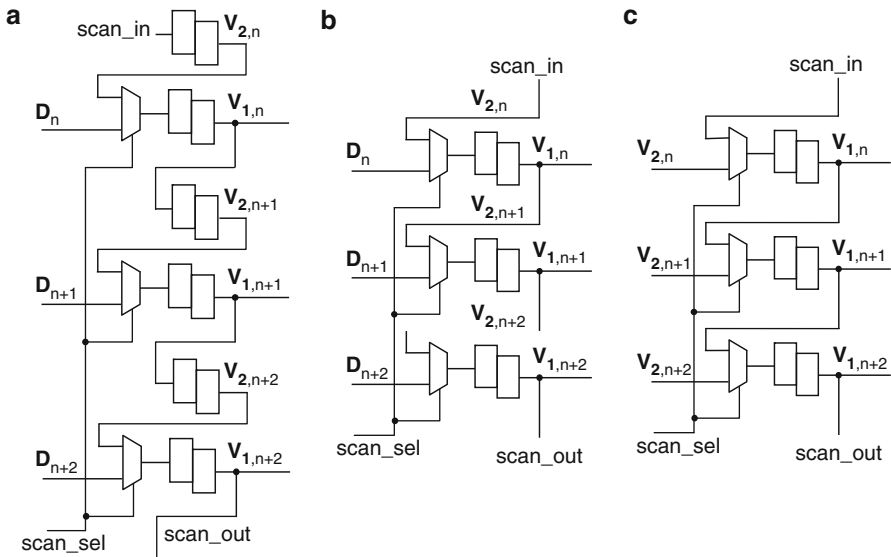


Fig. 3.18. Scan test configurations (a) enhanced scan, (b) skewed load, (c) broadside test

At the other extreme, the simplest AC test sequence relies on loading scan data into all scannable latches, then switching to functional mode to launch and capture data for test. In this case, V1 is unconstrained, but V2 is the next state determined from combinational logic response to the V1 vector, i.e., the functional data at the inputs of the CSEs given the V1 values at their outputs, as shown in Fig. 3.18c. This has been called a “broadside test” [51] sequence, and places minimal constraints on the design of the scannable CSEs. The speed at which the scan data can be written into the CSEs is unimportant, as are the details of the switch from scan mode to functional mode. The downside of this method is that AC test coverage can be low,

with few, or no options for improvement through DFT-related design changes. In addition, analysis and debug are more complicated, since V2 depends on the response of all the upstream combinational logic and the V1 state of all CSEs in the cone of logic, increasing simulation time for test analysis and making it harder to change V2 in a systematic way.

A compromise between the two above approaches is the concept of a “skewed load” sequence [52]. In this type of sequence, V1 is again scanned into the CSEs, but now V2 comes from the upstream data in the scan string as shown in Fig. 3.18b. During the high-speed test sequence, data must be loaded into each CSE from the scan port, and launched at speed into the downstream logic. On the next clock cycle, data is captured in each CSE through the functional data port. Sample clock waveforms for such a sequence are shown in Fig. 3.19a for the scannable MSL shown in Fig. 3.6, contrasting with those for the broadside load sequence (Fig. 3.19b). To make this sequence work properly, all CSEs must be designed such that the clock-Q delay from the scan port is a close match to that from the data port, otherwise the AC characteristics of the test will not match that of the functional operation. Also, the CSE clock control system has to be capable of switching from “scan mode” to “functional mode” within a single cycle, in order to launch scan data, and then to capture data from the functional port. This will generally require an accurate pipelining and distribution scheme for at least this one high-speed global test control signal.

One advantage of the skewed load test sequence is that it provides enhanced AC test coverage compared to the broadside test, although there are still coverage limitations which may arise from the adjacency of V1 and V2 in the scan string. Furthermore, the test patterns applied are very flexible, easy to modify in a systematic

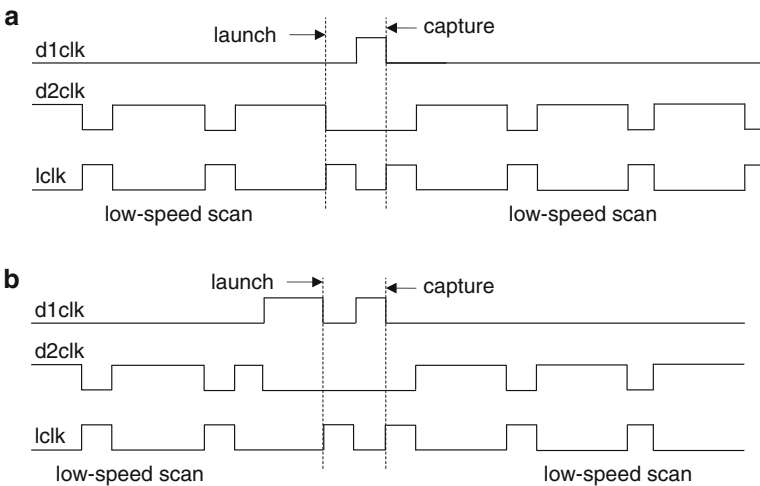


Fig. 3.19. Sample clock waveforms for AC test using the MSL from Fig. 3.6. (a) skewed load test. Note that data from scan port (d2clk) is launched. (b) Broadside test. Note that data from the functional port (d1clk) is launched

fashion, and failure analysis needs only to consider a single cycle's worth of logic instead of two as in the broadside case.

Finally, even though much of the discussion above has focused on single-cycle test sequences, which offer an appealing simplicity from a test point of view, various other CSE design factors may result in the requirement for longer at speed test sequences. Inclusion of nonscan elements (which may help to reduce chip area, latency, and power dissipation) will force the use of longer sequences to test all paths in and out of these components. Also, the use of soft cycle boundaries may also affect AC test operation, since certain critical paths may effectively become more than one cycle long.

3.6 CSE Design for Variability

Design for variability has become an increasingly important consideration as the relative level of parameter and device variability has been increasing recently and is expected to continue to increase rapidly with future technology scaling [53]. The importance of this fact is magnified by the expected continuation of device density scaling; this increased variability is manifested in ever larger collections of devices on a single chip. There are many ways in which variability may degrade the quality of a design, and this topic is addressed in greater detail in Chap. 7 of this book. The discussion here will focus on two particular aspects of variability-induced degradation which are relevant to CSE design, namely operating frequency degradation, which involves the cycle limiting paths in the design, and functional failures, which may result from racing paths with insufficient margin, or other design vulnerabilities.

3.6.1 Variability-Induced Frequency Degradation

The key aspect of cycle limiting paths in any given design is that there is usually a relatively large number of logic gates involved, including not only the logic between the launching and capturing CSEs, but also the CSEs themselves, and the circuitry driving the local clock signals to these CSEs. Thus local, uncorrelated, random fluctuations in device parameters are less likely to have a large impact on the timings of these paths, since these will average over the large numbers of devices involved. Problems are more likely to arise from more global, systematic variations which apply to all the circuits in a given region of the design. Some examples here might include PFET to NFET strength ratios, wire speed to device speed ratios, design pattern-factor-induced effects, chip mean device threshold voltage variability, etc. In principle, given an accurate knowledge of all the parameter distributions, and a complete statistical timing methodology [54], it would be possible to predict more accurately which speed paths are most likely to limit the design, and in fact it is expected that this sort of analysis will become more prevalent in the future. However, the underlying variabilities and uncertainties will not go away, even as the ability to accommodate them improves, and so it is important to understand how specific CSE design techniques might improve the situation.

The concept of soft cycle boundaries has already been discussed at some length, and this is an important method of providing some protection against variability or unknown factors. The 2-phase level-sensitive latch methodology probably provides the most benefit in this regard, but in light of the issues described earlier, it is unlikely that the usage of this style will become widespread as a solution for improved variability tolerance. Rather, it is expected that efforts to soften the cycle boundaries are more likely to be concentrated on MSFF and pulsed level-sensitive designs. Furthermore, given that an MSFF with overlapped clocks and pulsed level-sensitive latches have similar hold time issues for a given transparent window size, but unequal power and latency characteristics [5], it seems that pulsed clocking techniques would be the preferred method of providing a soft cycle boundary to protect against variability-induced frequency degradation. However, the local clock edges for the MSFF solution can be tuned to optimize the trade off between hold time and transparent window size without the minimum pulse-width constraints of pulsed-clock designs, and widespread use of pulsed-clock latches will make the design more prone to variability-induced functional failure, as described in the next section.

In addition to the use of soft cycle boundaries, many chips are being designed with programmable clock edges which can also be used to mitigate the effects of variability and uncertainty [28]. Delaying specific clock edges can mitigate the longest timing paths found in the hardware. It is expected that techniques such as this will become more prevalent in the future, as variability continues to increase, and as the improvement in technology decreases from generation to generation. One might expect to see more automated techniques used to optimize clock control settings on a chip-specific basis, or the use of adaptive or autonomous techniques, moving from the more global adaptive deskew techniques in use today [55] (also described in Chaps. 2 and 7), to more fine-grained adjustments in the future.

3.6.2 Variability-Induced Functional Failures

In contrast to variability-induced frequency degradation, a race path or other functional hazard may involve only a small number of devices, so that local random variations may play a big role in determining the margins required to protect against such failures. For specific circuit configurations within certain CSE topologies, it will be necessary to carry out a complete statistical analysis at the desired conditions for use, and at various process corners, to make sure that the design is robust enough against any probable statistical variation [6]. However, in the general case, it will be impossible to analyze all race conditions between all CSEs to that level of detail without a global statistical timing methodology. Also, since the amount of variability will depend on the detailed circuit parameters (for example channel widths involved, device types, the types and amounts of parasitic resistance and capacitances, etc.) it will become extremely difficult to guardband every potential race path against the worst-case combination of statistical fluctuations. Thus it is likely that advanced CSE designs and the need to protect against statistical-variability-induced failures will drive the need for an increasingly sophisticated true statistical timing methodology.

In fact, the need for such a methodology is arguably even greater here than for the operating frequency analysis discussed in the preceding section.

Although there are many sources of variability, any one of which may lead to failure, it is expected that future designs will continue to push towards lower supply voltages, while device dimensions continue to shrink. This means that factors leading to threshold voltage variability are likely to play an increasingly significant role [56]. For this reason, statistical analyses of race conditions will be needed at the minimum operating voltage, and this low-voltage corner will often impose more stringent restrictions on the design than will the high-voltage corner, especially if the write of the latch must overcome a “weak” keeper device. Recent effort has been reported on measurement of hold time variability, using structures designed to look at race conditions under realistic design conditions, showing directly, in a hardware-based measurement, the increase in margins needed as the technology is scaled to finer dimensions [57], and/or as the operating voltage is lowered [58].

Pulsed static, or pulsed semi-dynamic designs may be particularly susceptible to fluctuation-induced writeability failures. For this reason, designers have added local pulse-width controls to the pulse generators used for these latches [6, 28], for test purposes and also as an emergency option in case of unexpected hardware problems. Such an example is shown in Fig. 3.20. Another technique mentioned earlier in the context of power-reduction features is to configure the clock drivers for MSFF latches such that they can provide either regular clocks, or a pulsed clock to the slave with the master clock held at a constant high value [5, 22, 28]. This scheme maintains a fail-safe option in case of either writeability issues or data races. In principle, with techniques like this it should be possible to cut down on the margins normally needed for guaranteeing race-free operation; registers containing latches which are seen to fail in the hardware with pulsed-mode clocking could be configured to run in MSFF mode.

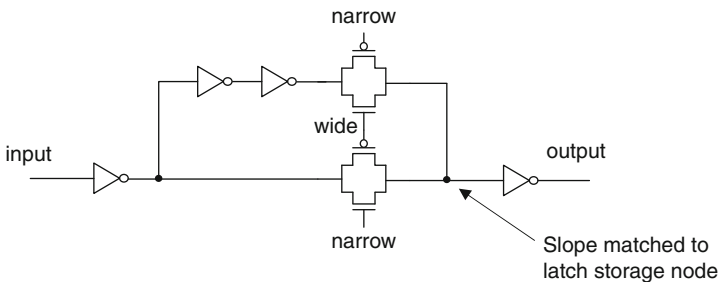


Fig. 3.20. Programmable delay line defining the trailing edge of a local clock pulse. Transmission gate structure is designed to match that in the latch. Reproduced with permission from [6], ©2008 IEEE

Techniques like these may be used not only to guard against unexpected problems, but may also help with test and debug, and one may imagine that in the future, designs may be automatically configured with the optimal pulse-width settings for simultaneously maximizing cycle stealing across soft cycle boundaries while maintaining a safe margin against race failures.

3.7 Reliability Issues

While issues related to process variability or random parameter fluctuations can, in principle, be screened or protected against by proper testing of the part, reliability issues can be much more difficult to protect against. It will be no surprise to the reader that adequate reliability of CSEs is becoming increasingly difficult to guarantee, as the number of CSEs on a single chip continues to increase and as the technology feature size continues to shrink. This section will examine two types of reliability issues, soft error upsets which can disturb the data in the CSEs, and wear out mechanisms which may cause failure after a prolonged period of use.

3.7.1 Soft Error Rate Considerations

Several factors combine to make SER robustness an increasingly important design concern in future systems. Although many designers may consider SER to be important only for situations where extremely high reliability is required, or a problem only for large, high-density SRAMs, recent work has shown that this assumption is no longer true, and as a result, soft errors in CSEs have been getting an increasing amount of attention in recent years.

As CMOS technology has continued to scale well into the sub-100nm dimensions, the combination of feature size reduction, and especially power supply voltage reduction has resulted in a steadily increasing susceptibility of individual latches and flip-flops to soft error upset [59–61], and by some accounts the SER rate in a typical latch or flip-flop has already eclipsed the SRAM SER rate in unprotected rate-per-bit comparisons [62]. Furthermore, error correction codes may be used to help improve the SER in SRAMs, with a relatively low overhead, but no such low-overhead solution is currently available for the collection of CSEs in a typical microprocessor. The voltage factor is especially important, and as systems become ever more adept at lowering chip voltage whenever possible to cut back on power dissipation, there will be a price to be paid in terms of SER. At the same time, the microprocessor industry has seen a shift away from steady clock frequency increases for system performance improvements, towards increased parallelism, increased functionality, and increased integration levels. Therefore, it is expected that the number of CSEs per chip will continue to grow rapidly. At the same time, the SER per element is also growing, leading to a looming “SER crisis” if no action is taken.

To improve the SER situation, in addition to specific CSE design techniques, there are various system, error checking/detection, and technology options which have been discussed in the literature [41, 60, 63–66]. Improvements in processes and materials may also help to reduce contaminants and/or provide increased immunity. However, this section will focus on local CSE design options, and methods therein to improve SER robustness. Perhaps the first approach to try might be to target improvements in SER by careful layout optimization. However, studies of the SER as a function of the CSE layout details have not shown any significant dependencies [62], and there appears to be little success reported in this regard. Another possibility is to improve the SER robustness of individual latches or flip-flops by simply adding

additional capacitance to sensitive nodes [67], or by selectively increasing the size of devices holding sensitive nodes [68]. These techniques have been shown to help, and overall area cost may not be too excessive in some situations [69]. However, these techniques are more useful for specific design areas where high reliability is needed, and operating frequency/power is not a big issue. These hardening techniques are not very suitable for general use in microprocessor design. Moreover, these SER mitigation strategies will not scale very well to future technologies.

Future strategies for improving the soft error characteristics of CSEs are more likely to involve topological modifications to the circuits themselves. It is expected that the industry is likely to make more use of circuit related hardening techniques such as the Dual Interlocked storage Cell (DICE) latch [70]. The DICE latch uses redundant and interconnected storage nodes such that the cell cannot be flipped by the upset of a single device. An example [71] is shown in Fig.3.21. This design is resistant to single device upsets, although it can be seen that certain nodes may be left floating for a short period of time after a strike occurs. Depending on the operating voltage, experimental data on this cell showed a factor of 30× to 100× improvement in SER robustness with the observed errors likely caused by multiple upsets. Since the failure mechanism for this type of CSE involves charge collection on multiple nodes, care should be taken during layout to separate critical devices in order to reduce the probability of an error caused in this fashion [71].

DICE latches have been employed in a recent high-end microprocessor design [6], illustrating the combination of pulsed-clock techniques for power reduction and soft cycle boundaries, scan capability for test and debug, and special design tech-

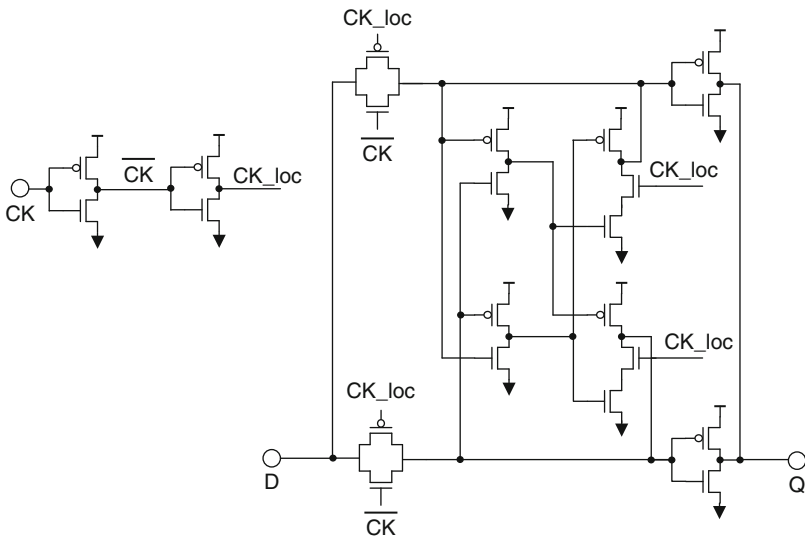


Fig. 3.21. DICE latch topology. Reproduced with permission in a form similar to that in [71], ©2004 IEEE

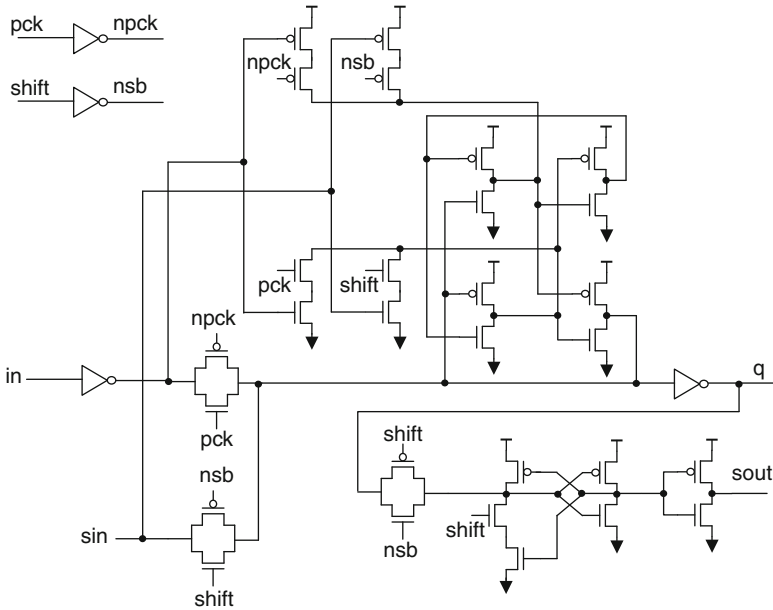


Fig. 3.22. Scannable pulsed-clock DICE latch. Reproduced with permission in a form similar to that in [6], ©2008 IEEE

niques for SER robustness. This scannable, pulsed-clock DICE latch is shown in Fig. 3.22. The DICE method is expensive; an overall latch area increase of about 35% and power increase of about 25% has been reported [6]. However, there will be many situations where error checking may not be practical, or the overhead may be too high (much higher than that of switching to DICE elements), and it is expected that techniques such as these will become more widespread in the future as designers are forced to turn their attention to this problem.

3.7.2 End of Life Considerations for CSE Design

There are a number of wear out mechanisms which affect the operation of silicon CMOS circuits, some of which may result in relatively sudden catastrophic failures, while others result in a more gradual parametric device degradation over time, leading eventually to failure. The former type of problem is difficult to deal with at the single element level; usually higher level checking or redundancy schemes are needed, which are beyond the scope of this chapter. However, the second type of problem is more amenable to local circuit solutions, and some of these solutions will be described in this section.

Two major phenomena which may lead to gradual circuit degradation are hot carrier injection (HCI) and negative bias temperature instability (NBTI) [72]. HCI has long been recognized as a potential problem for CMOS circuits [73, 74], but as modern CMOS technology has moved to lower voltages, degradation due to HCI

has tended to diminish in terms of its importance. This is especially true for digital CMOS circuits where the current flow through the devices is very transient in nature, and devices are not biased in a way that would subject them to hot carrier stress for long periods of time. Thus for CSE designs with reasonable signal slews, HCI is unlikely to cause significant degradation, although with the advent of new materials this an area that will bear watching for the future [75]. Both HCI and NBTI are addressed in more detail in Sect. 8.3.5

For CSE designs in today’s technologies, NBTI is a more important concern. NBTI is specific to PFETs, increasing the magnitude of the threshold voltage, and decreasing carrier mobility over time, depending on the stress conditions. The customary manner of treating this issue is to design test sequences such that all parts which pass are guaranteed to have adequate margin under all operating conditions against the impact of any future degradation. Random collections of parts may be stressed over periods of time to assess their reliability, given a particular set of screening tests at time zero. However, given the intrinsic statistical nature of the phenomenon involved, the shrinking of the device geometries with the resultant growing variability, and the increasing numbers of devices integrated on a single chip, it may be necessary to ask whether the margins needed during test will remain within reasonable bounds in the future, and, given the power/performance cost of maintaining such margins, whether there are CSE design techniques which could be used to reduce this overhead.

Most of the work on this subject is focused on the issue of frequency degradation, or the slow down of components which occurs over time. One method that has been studied is the so called Razor technique, originally proposed [76] to allow aggressive dynamic voltage scaling, but also applicable to wearout-induced circuit degradation as well. A Razor MSL is shown in Fig. 3.23. The clock to the shadow latch is delayed enough to guarantee that the incoming data is successfully latched, even when the main latch fails to capture the correct data. In the event that the two latches contain different data, an error signal is registered, and it is possible for the

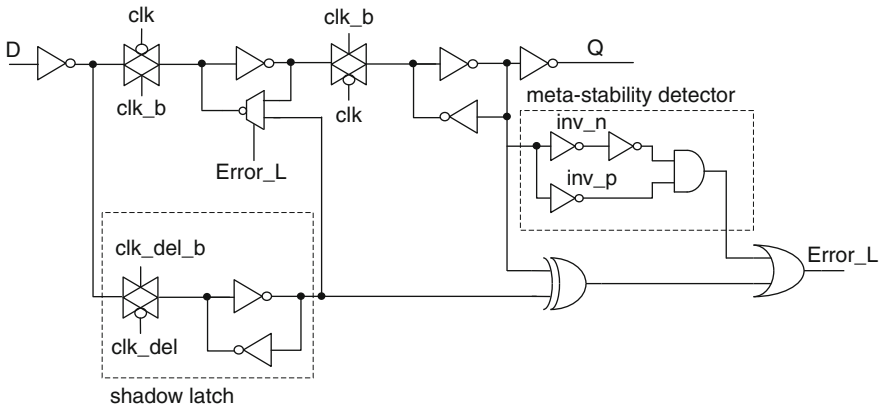


Fig. 3.23. Razor master–slave latch. Reproduced with permission from [76], ©2003 IEEE

system to recover by later swapping in the correct data from the shadow latch (at the cost of a few cycles, depending on the exact recovery scheme used). An error monitor could take appropriate action to maintain a reasonably low error rate, for example by adjusting the processor voltage or frequency, thereby avoiding excessive performance loss. The optimal error rate could then represent a trade off between the performance degradation caused by the overhead of error correction and the benefits of a higher frequency or lower operating voltage. The Razor technique applies considerable overhead to a typical MSL (not to mention the recovery logic overhead) but would not be needed on all CSEs. The Razor MSFF is itself vulnerable to hold time issues, and in fact the delayed clock to the shadow latch significantly increases the overall hold time for the CSE. Another issue of such a scheme would be that either the power or performance of the system could change over time, as adjustments were made for NBTI degradation (or simply in response to environmental changes), perhaps requiring re-instatement of some of the guardbands that were to be avoided in the first place. Sections 7.4.1–7.4.3 describe the Razor methodology in more detail from the viewpoint of addressing process variation through resiliency.

Other research has focused on trying to predict impending errors, and take action before the error actually occurs [78]. In this case, a transition detector watches for transitions which are arriving very late at the capturing CSE, and triggers if these transitions fall within the defined danger window. Chip monitoring hardware or software may then take action before any errors occur. This eliminates the logic complexity and overhead associated with the error recovery mechanisms. Such transition detectors may also be integrated into pulsed latches and used as error detectors [77, 79] in a similar fashion to the original Razor design, but with less overhead inside the CSE. An example of such a scheme is shown in Fig. 3.24. A more detailed description of these techniques appears in Sect. 7.4. One drawback of these transition detector schemes is that, in order to reliably signal the presence of an error (or an impending error), such circuits will need enough built-in margin to work reliably; it has to be guaranteed that the transition is always detected before an error actually occurs. This built-in margin will tend to lower the achievable operating frequency.

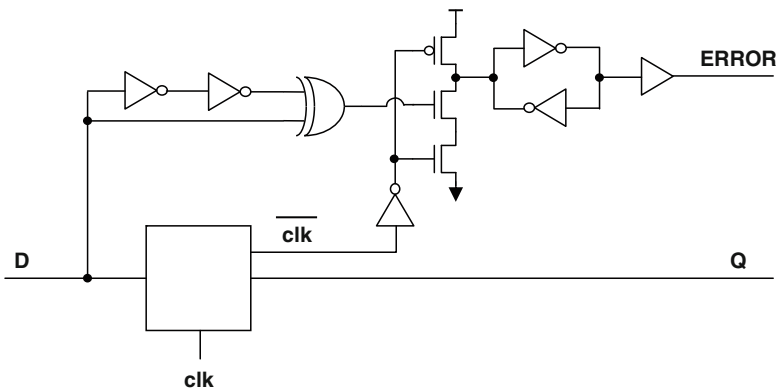


Fig. 3.24. Transition detection scheme. Reproduced with permission from [77], ©2008 IEEE

As a “last resort,” redundant logic techniques have also been studied, where the CSE has the ability to automatically swap out a whole block of logic on sensing an impending fail [80], replacing it with an equivalent set of logic gates. Of course, the overhead here is extremely high. Regardless of whether or not techniques like these ever become adopted in a widespread fashion by the industry, it is likely that future microprocessors will require more advanced techniques to guard against wear out-induced reliability failures, either locally at the CSE-level, as described above, or by using more global monitoring and checking algorithms.

Finally, throughout all of the above, it has been assumed that race conditions, pulse-width and/or latch writeability margins can be ensured through a combination of design margin and test conditions, without too much overhead. While this latter assumption may still hold true for some time in the future, usage of large numbers of pulsed-clock components in future technologies are likely to drive the need for more advanced testing techniques, including both race path and pulse-width stressing using some of the special clock pulse width/and/or clock edge control features described earlier. In addition, that fact that race path or hold time failures generally involve a small number of logic gates means that they will tend to be more sensitive to variabilities inherent in the various device degradation mechanisms. As this variability increases, it may no longer be possible to ensure adequate margin through specific test voltage and temperature conditions alone.

3.8 Conclusion

In light of the ongoing power crisis in modern microprocessors, tomorrow’s high-performance processors are likely to continue the push towards aggressive use of pulsed-static latches, which require only a single clock and provide for a soft cycle boundary. Improved analysis tools will be needed to guarantee robust operation of a large collection of such circuits across the full PVT space, and especially to be able to handle the ever-increasing impact of random local fluctuations. To ensure the highest quality, reliability, and system performance, future designs will use an increasingly sophisticated collection of special features for test, debug, and chip optimization. Finally, SER-related reliability will become a key issue for CSE design in the future. Tomorrow’s latch and flip-flop designers will need to consider not only the usual power/performance aspects of their solutions, but will also need to design for enhanced testability, robustness against statistical variations, and high levels of reliability, and SER immunity.

Acknowledgements

The author would like to acknowledge the feedback and comments from Leon Sigal and Thucydidis Xanthopoulos.

References

- [1] S. Unger and C.-J. Tan, "Clocking schemes for high-speed digital systems," *IEEE Trans. Comput.*, vol. 35, no. 10, pp. 880–895, Oct. 1986.
- [2] K. Wagner, "Clock system design," *IEEE Des. Test Comput.*, vol. 5, no. 5, pp. 9–27, Oct. 1988.
- [3] V. Oklobdzija, V. Stojanovic, D. Markovic, and N. Nedovic, *Digital System Clocking*. Wiley-IEEE Press, New York, 2003.
- [4] V. Oklobdzija, "Clocking and clocked storage elements in a multi-gigahertz environment," *IBM J. Res. Dev.*, vol. 47, no. 5/6, pp. 567–583, September/November 2003.
- [5] J. Warnock, D. Wendel, T. Aipperspach, E. Behnen, R. Cordes, S. Dhong, K. Hirairi, H. Murakami, S. Onishi, D. Pham, J. Pille, S. Posluszny, O. Takahashi, and H. Wen, "Circuit design techniques for a first-generation Cell broadband engine processor," *IEEE J. Solid-State Circuits*, vol. 41, no. 8, pp. 1692–1706, Aug. 2006.
- [6] D. Krueger, E. Francom, and J. Langsdorf, "Circuit design for voltage scaling and SER immunity on a quad-core Itanium® processor," in *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC 2008)*, 2008, pp. 94–95.
- [7] J. Friedrich, B. McCredie, N. James, B. Huott, B. Curran, E. Fluhr, G. Mittal, E. Chan, Y. Chan, D. Plass, S. Chu, H. Le, L. Clark, J. Ripley, S. Taylor, J. Dilullo, and M. Lanzerotti, "Design of the POWER6 microprocessor," in *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC 2007)*, 2007, pp. 96–97.
- [8] S. Naffziger, G. Colon-Bonet, T. Fischer, R. Riedlinger, T. Sullivan, and T. Grutkowski, "The implementation of the Itanium 2 microprocessor," *IEEE J. Solid-State Circuits*, vol. 37, no. 11, pp. 1448–1460, Nov. 2002.
- [9] C. Giacomotto, N. Nedovic, and V. Oklobdzija, "The effect of the system specification on the optimal selection of clocked storage elements," *IEEE J. Solid-State Circuits*, vol. 42, no. 6, pp. 1392–1404, June 2007.
- [10] V. Zyuban, D. Brooks, V. Srinivasan, M. Gschwind, P. Bose, P. Strenski, and P. Emma, "Integrated analysis of power and performance for pipelined microprocessors," *IEEE Trans. Comput.*, vol. 53, no. 8, pp. 1004–1016, Aug. 2004.
- [11] V. Stojanovic and V. Oklobdzija, "Comparative analysis of master-slave latches and flip-flops for high-performance and low-power systems," *IEEE J. Solid-State Circuits*, vol. 34, no. 4, pp. 536–548, April 1999.
- [12] D. Markovic, B. Nikolic, and R. Brodersen, "Analysis and design of low-energy flip-flops," in *Proceedings of the Low Power Electronics and Design, International Symposium*, 6–7 Aug. 2001, pp. 52–55.
- [13] V. Zyuban, "Optimization of scannable latches for low energy," *IEEE Trans. VLSI Syst.*, vol. 11, no. 5, pp. 778–788, Oct. 2003.
- [14] J. Tschanz, S. Narendra, Z. Chen, S. Borkar, M. Sachdev, and V. De, "Comparative delay and energy of single edge-triggered and dual edge-triggered pulsed flip-flops for high-performance microprocessors," in *Proceedings of the Low*

- Power Electronics and Design, International Symposium*, 6–7 Aug. 2001, pp. 147–152.
- [15] M. Hamada, H. Hara, T. Fujita, C. K. Teh, T. Shimazawa, N. Kawabe, T. Kitahara, Y. Kikuchi, T. Nishikawa, M. Takahashi, and Y. Oowaki, “A conditional clocking flip-flop for low power H.264/MPEG-4 audio/visual codec LSI,” in *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC 2005)*, 18–21 Sept. 2005, pp. 527–530.
- [16] S. DasGupta, E. Eichelberger, and T. Williams, “LSI chip design for testability,” in *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC 1978)*, 1978, pp. 216–217.
- [17] G. Gerosa, S. Gary, C. Dietz, D. Pham, K. Hoover, J. Alvarez, H. Sanchez, P. Ippolito, T. Ngo, S. Litch, J. Eno, J. Golab, N. Vanderschaaf, and J. Kahle, “A 2.2 W, 80 MHz superscalar RISC microprocessor,” *IEEE J. Solid-State Circuits*, vol. 29, no. 12, pp. 1440–1454, Dec. 1994.
- [18] R. Ho, K. Mai, and M. Horowitz, “The future of wires,” *Proceedings of the IEEE*, vol. 89, no. 4, pp. 490–504, April 2001.
- [19] Y. Suzuki, K. Odagawa, and T. Abe, “Clocked CMOS calculator circuitry,” *IEEE J. Solid-State Circuits*, vol. 8, no. 6, pp. 462–469, Dec 1973.
- [20] J. Warnock, J. Keaty, J. Petrovick, J. Clabes, C. Kircher, B. Krauter, P. Restle, B. Zoric, and C. Anderson, “The circuit and physical design of the POWER4 microprocessor,” *IBM J. Res. Dev.*, vol. 46, no. 1, pp. 27–51, January 2002.
- [21] D. Lackey, “Efficient latch and clock structures for system-on-chip test flexibility,” in *Proceedings of the IEEE International Test Conference ITC '06*, Oct. 2006, pp. 1–7.
- [22] B. Stolt, Y. Mittlefehldt, S. Dubey, G. Mittal, M. Lee, J. Friedrich, and E. Fluhr, “Design and implementation of the POWER6 microprocessor,” *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 21–28, Jan. 2008.
- [23] I. Lin, J. Ludwig, and K. Eng, “Analyzing cycle stealing on synchronous circuits with level-sensitive latches,” in *Proceedings of the ACM/IEEE Design Automation Conference*, 8–12 June 1992, pp. 393–398.
- [24] E. Shriver, D. Hall, N. Nassif, N. Rahman, N. Rethman, G. Watt, and J. Farrell, “Timing verification of the 21264: A 600 MHz full-custom microprocessor,” in *Proceedings of the International Conference on Computer Design: VLSI in Computers and Processors ICCD '98*, 5–7 Oct. 1998, pp. 96–103.
- [25] K. Sakallah, T. Mudge, and O. Olukotun, “ $checkT_c$ and $minT_c$: Timing verification and optimal clocking of synchronous digital circuits,” in *Proceedings of the IEEE International Conference on Computer-Aided Design ICCAD-90. Digest of Technical Papers*, 11–15 Nov. 1990, pp. 552–555.
- [26] T. Szymanski and N. Shenoy, “Verifying clock schedules,” in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design ICCAD-92. Digest of Technical Papers*, 8–12 Nov. 1992, pp. 124–131.
- [27] T. Burks, K. Sakallah, and T. Mudge, “Identification of critical paths in circuits with level-sensitive latches,” in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design ICCAD-92. Digest of Technical Papers*, 8–12 Nov. 1992, pp. 137–141.

- [28] B. Curran, "Power-constrained high-frequency circuits for the IBM POWER6 microprocessor," *IBM J. Res. Dev.*, vol. 51, no. 6, pp. 715–731, November 2007.
- [29] M. Matsui, H. Hara, Y. Uetani, L.-S. Kim, T. Nagamatsu, Y. Watanabe, A. Chiba, K. Matsuda, and T. Sakurai, "A 200 MHz 13 mm^2 -2D DCT macrocell using sense-amplifying pipeline flip-flop scheme," *IEEE J. Solid-State Circuits*, vol. 29, no. 12, pp. 1482–1490, Dec. 1994.
- [30] J. Montanaro, R. Witek, K. Anne, A. Black, E. Cooper, D. Dobberpuhl, P. Donahue, J. Eno, W. Hoepfner, D. Kruckemyer, T. Lee, P. Lin, L. Madden, D. Murray, M. Pearce, S. Santhanam, K. Snyder, R. Stehpany, and S. Thierauf, "A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor," *IEEE J. Solid-State Circuits*, vol. 31, no. 11, pp. 1703–1714, Nov. 1996.
- [31] P. Gronowski, W. Bowhill, R. Preston, M. Gowan, and R. Allmon, "High-performance microprocessor design," *IEEE J. Solid-State Circuits*, vol. 33, no. 5, pp. 676–686, May 1998.
- [32] B. Nikolic, V. Oklobdzija, V. Stojanovic, W. Jia, J. K.-S. Chiu, and M. Ming-Tak Leung, "Improved sense-amplifier-based flip-flop: design and measurements," *IEEE J. Solid-State Circuits*, vol. 35, no. 6, pp. 876–884, June 2000.
- [33] Y. Zhang, H. Yang, and H. Wang, "Low clock-swing conditional-precharge flip-flop for more than 30% power reduction," *Electron. Lett.*, vol. 36, no. 9, pp. 785–786, 2000.
- [34] T. Darwish and M. Bayoumi, "Reducing the switching activity of modified SAFF flip-flop for low power applications," in *Proceedings of the 14th International Conference on 2002 – ICM Microelectronics*, 11–13 Dec. 2002, pp. 96–99.
- [35] J.-C. Kim, Y.-C. Jang, and H.-J. Park, "CMOS sense amplifier-based flip-flop with two $N - C^2$ MOS output latches," *Electron. Lett.*, vol. 36, no. 6, pp. 498–500, 16 March 2000.
- [36] A. Strollo, D. De Caro, E. Napoli, and N. Petra, "A novel high-speed sense-amplifier-based flip-flop," *IEEE Trans. VLSI Syst.*, vol. 13, no. 11, pp. 1266–1274, Nov. 2005.
- [37] H. Partovi, R. Burd, U. Salim, F. Weber, L. DiGregorio, and D. Draper, "Flow-through latch and edge-triggered flip-flop hybrid elements," in *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC 1996)*, 1996, pp. 138–139.
- [38] N. Nedovic and V. Oklobdzija, "Dynamic flip-flop with improved power," in *Proceedings of the 26th European ESSCIRC Solid-State Circuits Conference '00*, 19–21 Sept. 2000, pp. 376–379.
- [39] N. Nedovic and V. Oklobdzija, "Hybrid latch flip-flop with improved power efficiency," in *Proceedings of the 13th Symposium on Integrated Circuits and Systems Design*, 18–24 Sept. 2000, pp. 211–215.
- [40] F. Klass, C. Amir, A. Das, K. Aingaran, C. Truong, R. Wang, A. Mehta, R. Heald, and G. Yee, "A new family of semidynamic and dynamic flip-flops with embedded logic for high-performance processors," *IEEE J. Solid-State Circuits*, vol. 34, no. 5, pp. 712–716, May 1999.

- [41] C. Webb, C. Anderson, L. Sigal, K. Shepard, J. Liptay, J. Warnock, B. Curran, B. Krumm, M. Mayo, P. Camporese, E. Schwarz, M. Farrell, P. Restle, I. Averill, R.M., T. Slegel, W. Houtt, Y. Chan, B. Wile, T. Nguyen, P. Emma, D. Beece, C.-T. Chuang, and C. Price, "A 400-MHz S/390 microprocessor," *IEEE J. Solid-State Circuits*, vol. 32, no. 11, pp. 1665–1675, Nov. 1997.
- [42] R. Heald, K. Aingaran, C. Amir, M. Ang, M. Boland, P. Dixit, G. Gouldsberry, D. Greenley, J. Grinberg, J. Hart, T. Horel, W.-J. Hsu, J. Kaku, C. Kim, S. Kim, F. Klass, H. Kwan, G. Lauterbach, R. Lo, H. McIntyre, A. Mehta, D. Murata, S. Nguyen, Y.-P. Pai, S. Patel, K. Shin, K. Tam, S. Vishwanthaiiah, J. Wu, G. Yee, and E. You, "A third-generation SPARC V9 64-b microprocessor," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1526–1538, Nov. 2000.
- [43] W. Belluomini, D. Jamsek, A. Martin, C. McDowell, R. Montoye, T. Nguyen, H. Ngo, J. Sawada, I. Vo, and R. Datta, "An 8 GHz floating-point multiply," in *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC 2005)*, 2005, pp. 374–375, 604.
- [44] J. Parkhurst, J. Darringer, and B. Grundmann, "From single core to multi-core: Preparing for a new exponential," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design ICCAD '06*, 2006, pp. 67–72.
- [45] B. Stackhouse, B. Cherkauer, M. Gowan, P. Gronowski, and C. Lyles, "A 65nm 2-billion-transistor quad-core Itanium® processor," in *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC 2008)*, 2008, pp. 92–93, 598.
- [46] C.-G. Hwang, "New paradigms in the silicon industry," in *Proceedings of the International Electron Devices Meeting IEDM '06*, 11–13 Dec. 2006, pp. 1–8.
- [47] D. Josephson, S. Poehlman, V. Govan, and C. Mumford, "Test methodology for the McKinley processor," in *Proceedings of the International Test Conference*, 30 Oct.–1 Nov. 2001, pp. 578–585.
- [48] R. Molyneaux, T. Ziaja, H. Kim, S. Aryani, S. Hwang, and A. Hsieh, "Design for testability features of the SUN Microsystems Niagara2 CMP/CMT SPARC chip," in *Proceedings of the IEEE International Test Conference ITC 2007*, 21–26 Oct. 2007, pp. 1–8.
- [49] S. DasGupta, R. Walther, T. Williams, and E. Eichelberger, "An enhancement to LSSD and some applications of LSSD in reliability, availability, and serviceability," in *Proceedings of the Twenty-Fifth International Symposium on Fault-Tolerant Computing, ' Highlights from Twenty-Five Years'*, 1995, p. 289.
- [50] K.-T. Cheng, S. Devadas, and K. Keutzer, "A partial enhanced-scan approach to robust delay-fault test generation for sequential circuits," in *Proceedings of the International Test Conference*, 26–30 Oct 1991, p. 403.
- [51] J. Savir and S. Patil, "On broad-side delay test," *IEEE Trans. VLSI Syst.*, vol. 2, no. 3, pp. 368–372, Sept. 1994.
- [52] J. Savir and S. Patil, "Scan-based transition test," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 12, no. 8, pp. 1232–1241, Aug. 1993.

- [53] A. Asenov, "Simulation of statistical variability in nano MOSFETs," in *Proceedings of the IEEE Symposium on VLSI Technology*, 12–14 June 2007, pp. 86–87.
- [54] C. Visweswariah, "Death, taxes and failing chips," in *Proceedings of the Design Automation Conference*, 2–6 June 2003, pp. 343–347.
- [55] E. Fetzer, "Using adaptive circuits to mitigate process variations in a microprocessor design," *IEEE Design & Test Comput.*, vol. 23, no. 6, pp. 476–483, June 2006.
- [56] G. Roy, A. Brown, F. Adamu-Lema, S. Roy, and A. Asenov, "Simulation study of individual and combined sources of intrinsic parameter fluctuations in conventional Nano-MOSFETs," *IEEE Trans. Electron. Devices*, vol. 53, no. 12, pp. 3063–3070, Dec. 2006.
- [57] G. Neuberger, F. Kastensmidt, R. Reis, G. Wirth, R. Brederlow, and C. Pacha, "Statistical analysis of systematic and random variability of flip-flop race immunity in 130nm and 90nm CMOS technologies," in *Proceedings of the IFIP International Conference on Very Large Scale Integration VLSI - SoC 2007*, 15–17 Oct. 2007, pp. 78–83.
- [58] F. Klass, A. Jain, G. Hess, and B. Park, "An all-digital on-chip process-control monitor for process-variability measurements," in *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC 2008)*, 3–7 Feb. 2008, pp. 408–409, 623.
- [59] T. Karnik and P. Hazucha, "Characterization of soft errors caused by single event upsets in CMOS processes," *IEEE Trans. Dependable Secure Comput.*, vol. 1, no. 2, pp. 128–143, April–June 2004.
- [60] R. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Trans. Device Mater. Rel.*, vol. 5, no. 3, pp. 305–316, Sept. 2005.
- [61] H. Fukui, M. Hamaguchi, H. Yoshimura, H. Oyamatsu, F. Matsuoka, T. Noguchi, T. Hirao, H. Abe, S. Onoda, T. Yamakawa, T. Wakasa, and T. Kamiya, "Comprehensive study on layout dependence of soft errors in CMOS latch circuits and its scaling trend for 65 nm technology node and beyond," in *Proceedings of the Digest of Technical Papers VLSI Technology 2005 Symposium*, 14–16 June 2005, pp. 222–223.
- [62] T. Heijmen, P. Roche, G. Gasiot, K. Forbes, and D. Giot, "A comprehensive study on the soft-error rate of flip-flops from 90-nm production libraries," *IEEE Trans. Device Mater. Rel.*, vol. 7, no. 1, pp. 84–96, March 2007.
- [63] F. Wang and V. D. Agrawal, "Single event upset: An embedded tutorial," in *Proceedings of the 21st International Conference on VLSI Design VLSID 2008*, 4–8 Jan. 2008, pp. 429–434.
- [64] M. Nicolaidis, "Design for soft error mitigation," *IEEE Trans. Device Mater. Rel.*, vol. 5, no. 3, pp. 405–418, Sept. 2005.
- [65] P. Meaney, S. Swaney, P. Sanda, and L. Spainhower, "IBM z990 soft error detection and recovery," *IEEE Trans. Device Mater. Rel.*, vol. 5, no. 3, pp. 419–427, Sept. 2005.

- [66] S. Mitra, M. Zhang, N. Seifert, T. Mak, and K. S. Kim, "Built-in soft error resilience for robust system design," in *Proceedings of the IEEE International Conference on Integrated Circuit Design and Technology ICICDT '07*, May 30 2007–June 1 2007, pp. 1–6.
- [67] Y. Dhillon, A. Diril, A. Chatterjee, and A. Singh, "Analysis and optimization of nanometer CMOS circuits for soft-error tolerance," *IEEE Trans. VLSI Syst.*, vol. 14, no. 5, pp. 514–524, May 2006.
- [68] Q. Zhou and K. Mohanram, "Gate sizing to radiation harden combinational logic," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 1, pp. 155–166, Jan. 2006.
- [69] A. KleinOowski, E. Cannon, M. Gordon, D. Heidel, P. Oldiges, C. Plettner, K. Rodbell, R. Rose, and H. Tang, "Latch design techniques for mitigating single event upsets in 65 nm SOI device technology," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 6, pp. 2021–2027, Dec. 2007.
- [70] T. Calin, M. Nicolaidis, and R. Velazco, "Upset hardened memory design for submicron CMOS technology," *IEEE Trans. Nucl. Sci.*, vol. 43, no. 6, pp. 2874–2878, Dec. 1996.
- [71] P. Hazucha, T. Karnik, S. Walstra, B. Bloechel, J. Tschanz, J. Maiz, K. Soumyanath, G. Dermer, S. Narendra, V. De, and S. Borkar, "Measurements and analysis of SER-tolerant latch in a 90-nm dual-Vt CMOS process," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1536–1543, Sept. 2004.
- [72] D. Schroder and J. Babcock, "Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing," *J. Appl. Phys.*, vol. 94, p. 1, 2003.
- [73] T. Ning, P. Cook, R. Dennard, C. Osburn, S. Schuster, and H. Yu, "1 μm MOSFET VLSI technology: Part IV – hot-electron design constraints," *IEEE Trans. Electron. Devices*, vol. 26, no. 4, pp. 346–353, Apr 1979.
- [74] C. Hu, S. C. Tam, F.-C. Hsu, P.-K. Ko, T.-Y. Chan, and K. Terrill, "Hot-electron-induced MOSFET degradation – model, monitor, and improvement," *IEEE J. Solid-State Circuits*, vol. 20, no. 1, pp. 295–305, Feb 1985.
- [75] J. McPherson, "Reliability trends with advanced CMOS scaling and the implications for design," in *Proceedings of the IEEE Custom Integrated Circuits Conference CICC '07*, 16–19 Sept. 2007, pp. 405–412.
- [76] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: a low-power pipeline based on circuit-level timing speculation," in *Proceedings of the 36th Annual IEEE/ACM International Symposium on MICRO-36 Microarchitecture*, 2003, pp. 7–18.
- [77] K. Bowman, J. Tschanz, N. S. Kim, J. Lee, C. Wilkerson, S.-L. Lu, T. Karnik, and V. De, "Energy-efficient and metastability-immune timing-error detection and instruction-replay-based recovery circuits for dynamic-variation tolerance," in *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC 2008)*, 3–7 Feb. 2008, pp. 402–403, 623.
- [78] M. Agarwal, B. Paul, M. Zhang, and S. Mitra, "Circuit failure prediction and its application to transistor aging," in *Proceedings of the 25th IEEE VLSI Test Symposium*, 6–10 May 2007, pp. 277–286.

- [79] D. Blaauw, S. Kalaiselvan, K. Lai, W.-H. Ma, S. Pant, C. Tokunaga, S. Das, and D. Bull, “Razor II: In situ error detection and correction for PVT and SER tolerance,” in *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC 2008)*, 2008, pp. 400–401, 622.
- [80] T. Nakura, K. Nose, and M. Mizuno, “Fine-grain redundant logic using defect-prediction flip-flops,” in *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC 2007)*, 2007, pp. 402–403, 611.