WHITE PAPER

cadence

DESIGNING CLOCK SIGNALS—PREDICTABILITY COUNTS

TABLE OF CONTENTS

TABLE OF FIGURES

OVERVIEW

Ultra-deep submicron IC clock signals and clocking schemes are designed carefully to achieve optimum performance. A poorly designed clock can cost a company millions of dollars in profits. Simply put, the faster the clock, the more money a company will make.

Clock design involves a continuous trade-off among design time, power, clock speed, clock skew, and performance. The less predictable these items are, the more you have to overdesign to ensure a certain clock speed. But if you overdesign too much, you can increase the risk of chip failure due to IR drop, signal integrity (SI), or electromigration (EM).

CLOCK DESIGN IS OVER-CONSTRAINED

The requirements of clock design are severely constraining. You have to minimize clock skew, clock delay, clock area and power, and noise while maximizing clock reliability. In addition, you have to drive an unbalanced load, which won't be known until the chip is nearly finished. And in the end, you can't compromise the schedule.

So can you deliver all of the above? No. However, with the right tools you can find the point of maximum profitability for your design.

THE CLOCK AFFECTS THE WHOLE CHIP

Clock design is tightly coupled with various core components of a chip—library design, floorplanning, scan insertion, timing verification, and critical path networks.

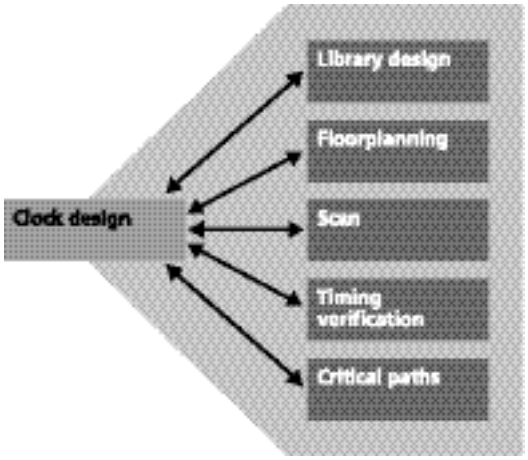


Figure 1: Clock design has global impact

- Latches and flip-flops are dependent on the clock design
- Depending on how the clock and latches are designed, you might need to perform a minimum delay for a path check
  - If a data signal arrives too early, it can cause a functional failure. But if the clock is designed differently, that condition may never arise, so you don't have to check for it. Interdependency exists among how the clock is designed, what checks you have to perform, and which constraints must be verified
- The floorplanning tool interacts with the clock design
  - For example, a big memory doesn't require a clock, but datapaths depend heavily on the clock. How you design the clock to be distributed over the chip depends on where all these other things are, and what kind of load they present to the clock
- Implementing scan in your chip has an impact on the clock because scan can introduce timing problems
- Hold time violations can be fatal and slowing down the clock does not alleviate them
- Clock skew adds directly to your minimum cycle time

THE FOUR RULES OF CLOCK DESIGN

If you follow four basic rules while designing your clock, you'll increase your chances of beating the competition in the speed game:

RULE #1: KNOW YOUR DESIGN REQUIREMENTS

While designing the clock, keep your design goals in mind:

- Deliver performance
- Minimize risk
- Maximize profits

These goals appear to be simple; however, historically, designers have achieved them only by over-designing the clock. Microprocessor designers have employed such techniques as huge drivers to ensure short delays and small skew. But the power consumed by the drivers alone can cause the chip to fail.

Other secondary problems caused by deep submicron process requirements may appear, such as IR drop, SI, and EM. It's like fighting a fire—more hoses may not help if the water pressure drops to near zero because too many hoses are attached to one water system. Just over-designing is no longer a solution because it leads to IR, SI, and EM failures.

With either approach, you're making tradeoffs between clock performance and company profit. Your design efforts may result in a clock with inadequate performance, or one that fails altogether due to secondary problems caused by deep submicron effects.

The clock's job

The clock's function is to synchronize all the latches and flip-flops, and to synchronize the data I/O with the internal operation. Depending on how it's designed, the clock may or may not succeed. The clock and the critical path are interdependent. The amount of time available for the critical path delay is derived from the clock design so if the clock is not well designed, your critical path delay is going to be too small.

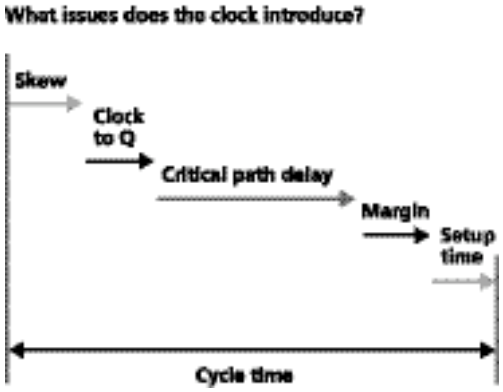


Figure 2: Critical path and clock skew interact

The entire clock cycle consists of the skew, clock to Q, critical path delay, some amount of margin, and set-up time. Clock skew can introduce new issues into the equation—for example, if skew is too large, a hold time violation can occur.

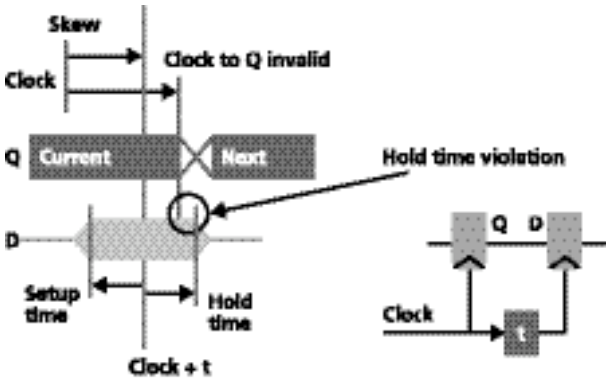


Figure 3: Hold time failures are independent of frequency

RULE #2: KNOW YOUR TECHNOLOGY UNKNOWNS

Technology is changing fast and design problems are definitely keeping pace with the rate of change! The rate of increase in transistors per chip is about the same as the rate of decrease in clock cycle time—and typically the smaller your technology, the more transistors you have, and the more delay is due to interconnect. The verification tools you used in the past are focused on gate delay and not interconnect delay. The design margins forced by the inaccuracy of these older tools means it is impossible to verify whether you meet your design requirements.

Interconnect verification reveals the unknowns

A physical verification strategy known as "interconnect verification" is required to reveal and pinpoint the unknowns of deep submicron. Interconnect verification consists of full-chip 3-D parasitic extraction combined with physical design analysis. To achieve the accuracy demanded by state-of-the-art process technology, 3-D extraction accuracy is critical. Depending on the clock frequency as well as design style RC extraction may be sufficient. However with higher clock frequencies and/or a design methodology without shielding, it becomes important to take inductive effects into account. Using the extracted data, analysis and visualization of such phenomena as IR drop, SI, timing, and EM reveal previously unknown design problems. Power grid IR drop affects skew and jitter. For example, a 5% IR drop can increase stage delay by 5% to 15%.

Plan for change

Because technology will continue to take you into unknown territory, you need to plan for evolutionary changes such as:

- Process migration and shrinks
- Reduced voltage operation
- Future device model "refinements"
- Random and systemic process variations
- Environment changes

For certain, the assumptions you made at the beginning of your design project will have to change before the end of the project, or at least during the lifetime of the chip. For example, the manufacturing technology will be different and the characteristics of the design will change. Your clock design must be robust enough to handle constantly changing processes and technologies. Most designs are shrunk — some even before first tapeout.

This leads to the advice: If you (don't) like your process today, (don't) worry, it will change!

RULE #3: DETERMINE YOUR CLOCK METHODOLOGY

Clocks are designed today automatically by synthesis or custom-crafted by hand. Many designers use a hybrid design approach where some parts of the chip are synthesized and others are done by hand. Table 1 shows the pros and cons of each approach.

SYNTHESIS	HAND DESIGN
You get what you get	Performance first
"Correct by construction"	Tedious
Simplistic	Complex
Based on estimation	Based on extraction
Some tool support	Little tool support

Table 1: Hand design vs. synthesis

Although "correct by construction," clock synthesis uses estimations of wire delay and many of the assumptions made are not valid. The only way to obtain accurate delays is to use extraction and analysis.

Clock implementation styles

Clock implementation styles vary from H-tree, custom buffered tree, distributed driven gridded clock, and various hybrids. Table 2 shows the advantages and disadvantages of each style.

STYLE	ADVANTAGES	DISADVANTAGES
H-tree	• Balanced by construction • Fixed routing (wire placement is easy)	• Rigid floor plan • Fixed routing (rigid)
Custom buffered tree	• Distributes drivers and power • Various sized drivers • Maximum flexibility	• Custom designed • Skew is more difficult to manage • Narrow routes are more susceptible to noise
Distributed driven gridded	• Reduced local clock skew • Distributes drivers and power • Clock routing a local issue, not global • Conservative	• Custom buffer placement • Inefficient use of area
Single driver	• Easy	• High IR drop potential • Strong noise source • Doesn't scale
Distributed	• Alleviates IR drop • Distributes power and IR effects to reduce noise • Complicated	• Complicated

Table 2: Advantages and disadvantages of each style

H-Tree

Custom buffered tree

Distributed driven gridded

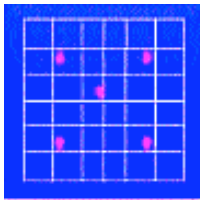


Figure 4: Clock implementation styles

RULE #4: NEVER DESIGN WHAT YOU CAN'T VERIFY

The goal of clock verification is to reduce the uncertainty about how your clock behaves in operation, and enable you to achieve maximum profitability. The more predictable your clock is, the less uncertainty you have that the clock will work. Ultra-deep submicron requires tight margins and high confidence in physical verification tools and methodologies.

Issues that increase uncertainty include:

- Process variations
- Extraction accuracy
- Analysis accuracy
- Analysis capacity
- Your ability to interpret results

As the need for predictability increases, the capabilities of your tools must increase as well. If the predictability you require is not being provided by your physical verification tools, you need new tools.

THE SEVEN STEPS OF CLOCK DESIGN

The seven steps of clock design are listed below along with some items to consider during each step.

1. **Design your clock phasing scheme.** The scheme you select will be dictated by your flop and latch design. Select whether you will design with:
   - Single or multiple phases
   - Individually distributed or encoded clocks
2. **Identify the sources of clocks:** external, internal, PLL, or scan
3. **Evaluate proposed flop/latch styles.** Consider the following:
   - Set-up/hold/clock to Q trade-offs
   - Area/power/performance trade-offs
   - Scan
   - Positive vs. negative edge
   - Load per flop
   - Constant load
   - Total number of flops
   - Other loads/special loads
   - Total “end-user” clock load

4. Simulate inverters and buffers. Consider the following:

- Drive strength and load
- Transfer curves (PVT corners)
- IR drop interaction
- Package L dl/dt, resonance effects, and thermal effects
- Electromigration (EM)
- Determine standard finger
- Very consistent layout (dummy fingers)
- Same physical orientation and context

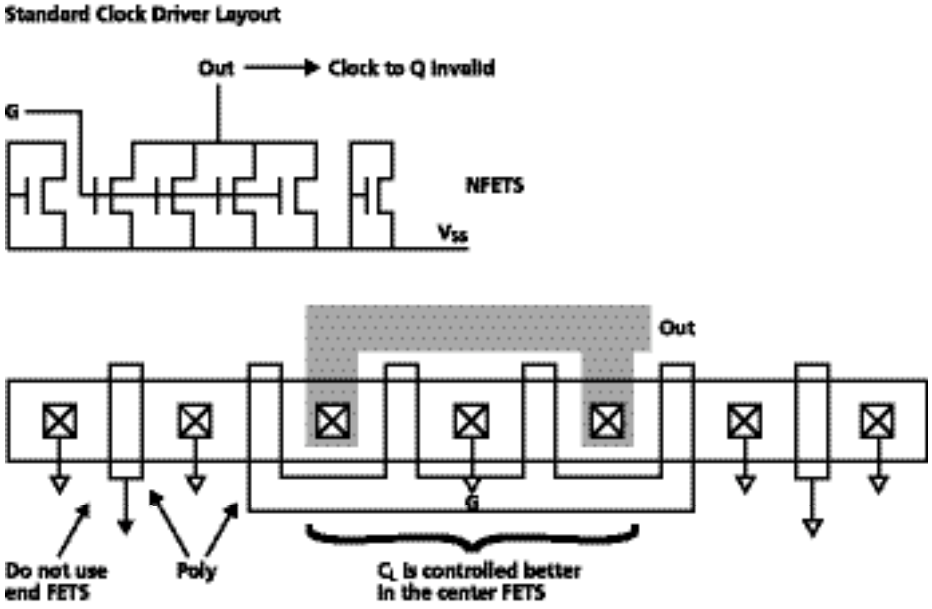


Figure 5: All clock drivers should be oriented the same way to help minimize the effects that would lead to channel length variations

5. Study the wires. Consider the following:

- Repeater study
- Gain stage spacing for clocks
- Test layouts, varying width, spacing, layer, shielding, and neighboring layer density
- Skin effects (cases to avoid it)
- Verify last twig non-shielding effect
- EM
- Via resistance variance

6. Select your topology. Consider the following:

- Load uniform and predictable?
- Load fixed or changes late in schedule?
- Tools verify changes quickly?
- Synthesis vs. custom (both need verification)
- Tree vs. grid

7. Design for verification. Consider the following:

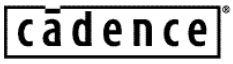
- Tool capabilities
- Tool limitations
- Building block approach
- Test cases and capacity verification
- Data reduction
- Data presentation

THOROUGH CLOCK DESIGN

Thorough clock verification requires the following:

- 3-D parasitic extraction accuracy
- Simulation accuracy
- Chip-level capacity
- Analysis of SI effects such as crosstalk noise and delay and power IR drop effects

If you're not getting all of this from your verification tools, you can’t do a good enough job to ensure predictability of clock design success. The Cadence interconnect verification tools can help you achieve your goals, improving the predictability of your clock performance — and maximize profitability for your company.



Cadence Design Systems, Inc.

Corporate Headquarters
555 River Oaks Parkway
San Jose, CA 95134
800.746.6223
408.943.1234
www.cadence.com

Stock #4107 08/02