MOST TECHNICAL LITERATURE ON HIGH-SPEED DESIGN FOCUSES ON TERMINATION, RINGING, AND CROSSTALK. DESPITE SIGNAL INTEGRITY'S IMPORTANCE, INADEQUATE TIMING MARGINS CAUSE MANY MORE ERRORS IN TODAY'S 100-MHz DIGITAL DESIGNS.

Practical timing analysis for 100-MHz digital designs

As increasing chip complexity, high clock rates, and analog signal-integrity issues complicate digital design, time-to-market pressures continue to shorten development schedules. These factors present increasing challenges to digital-design engineers, who must spend more time understanding software and system-level issues and have less time for details such as timing analysis. Because you can't ignore board-level propagation delays in today's more-than-100-MHz designs, detailed timing analysis is essential.

This analysis develops general timing-margin equations, which include the effects of clock skew and propagation delay. A fast and effective procedure for analyzing timing margins uses a spreadsheet instead of traditional timing diagrams. This approach quickly identifies approximate timing margins early in the design. A little extra work can improve the margins, reduce the parts cost, or reduce engineering or pc-board-layout effort.

DIGITAL-TIMING REVIEW

A good starting point is a review of the timing requirements for a typical synchronous digital connection, such as the one in **Figure 1**. In this circuit, the driver, IC_1, produces valid data after a delay of T_{CO} from the rising clock edge. The data must arrive at IC_2 in time to meet the receiver's input setup-and-hold-time requirements.

Figure 2 illustrates the timing relationships among the driver clock-to-output delay, T_{CO}; the receiver setup time, T_{SU}; and the receiver hold time, T_H. Given these timing parameters, which the IC manufacturer normally specifies, you can establish the following relationships for maximum clock frequency, setup margin, and hold margin:

$$\text{MINIMUM CLOCK PERIOD} = T_{CO(MAX)} + T_{SU},$$

$$\text{SETUP MARGIN} = \text{CLOCK PERIOD} - T_{CO(MAX)} - T_{SU},$$

and

$$\text{HOLD MARGIN} = T_{CO(MIN)} - T_H,$$

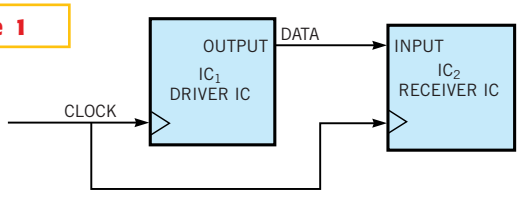
where $T_{CO(MIN)}$ and $T_{CO(MAX)}$ are the manufacturer-specified minimum and maximum clock-to-output propagation-delay values for IC_1's output, and T_{SU} and T_H are the manufacturer-specified minimum setup-and-hold times for IC_2's input. Note that although you can always increase the clock period to increase the setup margin, increasing the clock period does not affect the hold margin, leading to the following important result: Input-hold margin is independent of clock frequency.

This result indicates the importance of verifying hold-time requirements during a project's design phase, because you can't eliminate hold-time violations by simply lowering the clock frequency.

BOARD-LEVEL PROPAGATION DELAY

The simple example of **Figure 1** neglects data-propagation delay and clock skew between the transmitting and receiving ICs. In real digital systems in which clock signals reach frequencies of 100 MHz and higher, board-level-propagation or TOF (time-

Figure 1



This basic synchronous digital circuit includes only a driver and a receiver, both of which depend upon a common clock.

of-flight) delays are significant. In fact, at these high speeds, controlling clock skew is often necessary for producing adequate timing margins. To perform a realistic timing analysis, you must therefore modify the basic timing model to include clock skew and propagation delay. To model TOF and clock skew, the diagram in **Figure 3** introduces separate signals for clock and data at the driver and receiver. In this diagram, CLK_1 and $DATA_1$ represent signals at the driver, and CLK_2 and $DATA_2$ represent signals at the receiver.

Figure 4 shows the timing diagram for this model, including the additional signals to represent clock and data timing at the receiver as well as at the transmitter. TOF_D and T_{SKEWC} represent propagation delay and clock skew in the equations for setup margin, hold margin, and minimum clock period:

$$\text{SETUP MARGIN} = \text{CLOCK PERIOD} - T_{CO(MAX)} - T_{SU} - TOF_D + T_{SKEWC}, \quad (1)$$

$$\text{HOLD MARGIN} = T_{CO(MIN)} - T_H + TOF_D - T_{SKEWC}, \text{ and} \quad (2)$$

$$\text{MINIMUM CLOCK PERIOD} = T_{CO(MAX)} - T_{CO(MIN)} + T_{SU} + T_H,$$

where TOF_D is the propagation delay, or TOF for the data path between IC_1 and IC_2. T_{SKEWC} is the clock skew from IC_1 and IC_2, defined as positive when IC_2 is clocked later in time than IC_1. You can establish a lower bound on the clock period by setting the setup-and-hold margins to 0 in **equations 1** and **2**, yielding:

$$\text{CLOCK PERIOD} = T_{CO(MAX)} + T_{SU} + TOF_D - T_{SKEWC}, \quad (3)$$

and

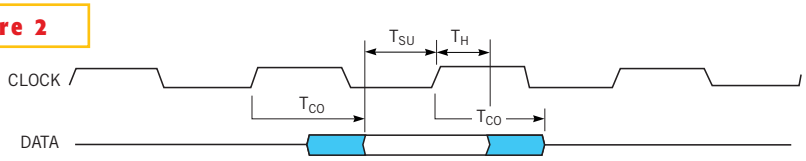
$$T_{CO(MIN)} = T_H = TOF_D + T_{SKEWC}. \quad (4)$$

Eliminating the term $(T_{SKEWC} - TOF_D)$ by substitution in **equations 3** and **4** leaves you with the following relationship for minimum clock period:

$$\text{MINIMUM CLOCK PERIOD} = T_{CO(MAX)} - T_{CO(MIN)} + T_{SU} + T_H.$$

This equation indicates that if you can control clock skew, the maximum achievable clock frequency for this circuit is independent of propagation delay, leading to the following important result: For unidirectional signaling, the uncertainty in driver propagation delay and the receiver setup-and-hold times limit the clock frequency, but the clock frequency is independent of the overall propagation-delay magnitude.

Figure 2



In the circuit of Figure 1, the driver, IC_1, produces valid data after a delay of T_{CO} from the rising clock edge. The data must arrive at IC_2 in time to meet the receiver's input setup-and-hold-time requirements.

POSITIVE SKEW

In this example, the clock skew is positive, which means that IC_2 is clocked after IC_1. In general, T_{SKEWC} can be either positive or negative. As you will see, the ability to adjust T_{SKEW} provides considerable flexibility in optimizing timing margins. With this timing model, you can introduce delays to the clock signals, the data signals, or both to improve timing margins, thereby increasing clock frequency.

You can make the several important observations in conjunction with this timing model. Delaying the clock signal by increasing the pc-board clock-trace

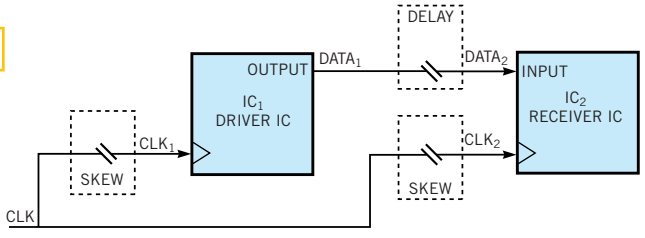
TABLE 1—TIMING-MARGIN-ANALYSIS SPREADSHEET

Timing equations: **Setup margin=clock period-$T_{CO(MAX)}$−TOF-T_{SU}+T_{SKEW}**				
Hold margin=$T_{CO(MIN)}$+TOF-T_H−T_{SKEW}				
T_{CO}−clock-to-output delay		**T_H--input-hold time**		
T_{TOF}−output-signal time-of-flight		**T_{SKEW}--output-to-input clock skew**		
T_{SU}−input-setup time				
Input Parameters:				
Clock frequency				
PC-board-trace delays				
IC_1 timing parameters				
IC_2 timing parameters				
IC_1 output to IC_2 input:			**IC_2 output to IC_1 input:**	
Clock frequency (MHz):	**100**		**Clock frequency (MHz):**	**100**
Clock skew (nsec):	**0.5**		**Clock skew (nsec):**	**−0.5**
TOF (nsec):	**0.25**		**TOF (nsec):**	**0.25**
T_{CO} output (nsec, minimum):	**1.5**		**T_{CO} output (nsec, minimum):**	**1**
T_{CO} output (nsec, maximum):	**7**		**T_{CO} output (nsec, maximum):**	**5.5**
T_{SU} input (nsec):	**1.5**		**T_{SU} input (nsec):**	**1**
T_H input (nsec):	**0.8**		**T_H input (nsec):**	**1**
Clock period (nsec):	**10**		**Clock period (nsec):**	**10**
Setup margin (nsec):	**1.75**		**Setup margin (nsec):**	**2.75**
Hold margin (nsec):	**0.45**		**Hold margin (nsec):**	**0.75**

length relative to the data-trace length increases the setup margin at the expense of the hold margin, allowing the circuit to operate at a higher frequency. Delaying the data signal by increasing the pc-board data-trace length relative to the clock-trace length increases hold margin at the expense of setup margin. Additional hold margin may be necessary to accommodate devices with unusually long hold times. When T_{SKEWC} is equal to TOF_D, the clock skew exactly compensates for TOF, and **equations 1** and **2** give the same results as the previous model, which ignores TOF. For this example, which models the special case of unidirectional signaling, you can usually use a very fast clock, regardless of the propagation delay, as long as the data and clock delays are equal. In this case, $(T_{CO(MAX)} - T_{CO(MIN)})$ rather than $|T_{CO}|$ (the magnitude of T_{CO}) limits the clock frequency.

As you will see, you can't draw the same conclusion in the case of bidirectional signaling. In many cases, an IC with less T_{CO} uncertainty is easier to use in a high-speed design than one with a smaller overall T_{CO}. For this reason, IC manufacturers should always specify accurate $T_{CO(MIN)}$ values. As an example, consider a design that interfaces a device with 0 or unspecified $T_{CO(MIN)}$ to an SDRAM with a 1-nsec input-hold-time requirement. Whereas the SDRAM's $T_{CO(MIN)}$ is unlikely to ever be less then 1 nsec, strictly meeting the 1-nsec SDRAM hold time requires that you add 1 nsec (about 6 in.) of board-trace delay, thereby sacrificing 1 nsec of setup margin.

Figure 3



To model TOF and clock skew, this diagram introduces separate signals for the clock and the data at the driver and receiver.

TABLE 2–TIMING MARGINS FOR SELECTED TOF AND CLOCK-SKEW VALUES

TOF	Clock skew	IC_1 setup	IC_1 hold	IC_2 setup	IC_2 hold
0.25	−0.5	3.75	−0.25	0.75	1.45
0.25	−0.4	3.65	−0.15	0.85	1.35
0.25	−0.3	3.55	−0.05	0.95	1.25
0.25	−0.2	3.45	0.05	1.05	1.15
0.25	−0.1	3.35	0.15	1.15	1.05
0.25	0	3.25	0.25	1.25	0.95
0.25	0.1	3.15	0.35	1.35	0.85
0.25	0.2	3.05	0.45	1.45	0.75
0.25	0.3	2.95	0.55	1.55	0.65
0.25	0.4	2.85	0.65	1.65	0.55
0.25	0.5	2.75	0.75	1.75	0.45
0.25	0.6	2.65	0.85	1.85	0.35
0.25	0.7	2.55	0.95	1.95	0.25
0.25	0.8	2.45	1.05	2.05	0.15
0.68	−0.5	3.32	0.18	0.32	1.88
0.68	−0.4	3.22	0.28	0.42	1.78
0.68	−0.3	3.12	0.38	0.52	1.68
0.68	−0.2	3.02	0.48	0.62	1.58
0.68	−0.1	2.92	0.58	0.72	1.48
0.68	0	2.82	0.68	0.82	1.38
0.68	0.1	2.72	0.78	0.92	1.28
0.68	0.2	2.62	0.88	1.02	1.18
0.68	0.3	2.52	0.98	1.12	1.08
0.68	0.4	2.42	1.08	1.22	0.98
0.68	0.5	2.32	1.18	1.32	0.88
0.68	0.6	2.22	1.28	1.42	0.78
0.68	0.7	2.12	1.38	1.52	0.68
0.68	0.8	2.02	1.48	1.62	0.58

GENERALIZED TIMING MODEL

Because most digital connections involve inputs as well as outputs on each IC, the next step is to extend the timing model to include bidirectional signaling. **Figure 5** shows this straightforward extension, which adds a control-signal connection from IC_2 to IC_1. In this **figure**, you can think of IC_1 as an SDRAM and IC_2 as a microprocessor, The datapath represents the SDRAM data bus, and the CTRL (control) path represents the SDRAM address and control signals.

Figure 6 shows the timing diagram for the bidirectional-signaling model, including signals CTL_1 and CTL_2, which represent the control signals from IC_2 to IC_1. The setup-and-hold analysis is also a straightforward extension of the unidirectional analysis:

$$IC_2\ SETUP\ MARGIN = CLOCK\ PERIOD - T_{COU1(MAX)} - T_{SUU2} - TOF + T_{SKEWC};$$

$$IC_2\ HOLD\ MARGIN = T_{COU1(MIN)} - T_{HU2} + TOF - T_{SKEWC};$$

$$IC_1\ SETUP\ MARGIN = CLOCK\ PERIOD - T_{COU2(MAX)} - T_{SUU1} - TOF - T_{SKEWC};$$

and

$$IC_1\ HOLD\ MARGIN = T_{COU2(MIN)} - T_{HU1} + TOF + T_{SKEWC}.$$

The timing diagram in **Figure 6** is too complicated for everyday use. Moreover, there is no longer a closed expression for minimum clock period because there are too many parameters. Fortunately, you can easily perform timing analysis for this model with a spreadsheet. Before beginning the spreadsheet analysis, observe the following about the bidirectional-signaling model:

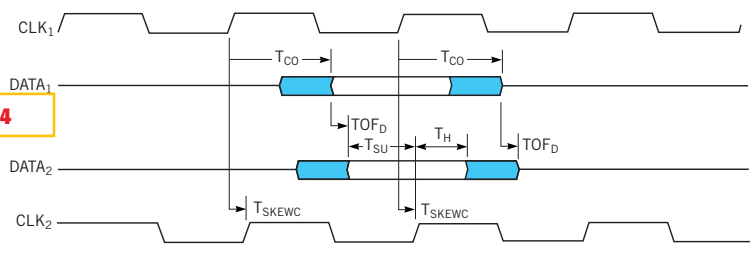
- Delaying the clock signal to IC_2 by increasing the pc-board clock-trace

length *increases* data-setup margin at IC_2 and *increases* CTL-hold margin at IC_1 at the expense of *decreased* data-hold margin at IC_2 and *decreased* CTL-setup margin at IC_1.

- Delaying the clock signal to IC_1 by increasing the pc-board clock-trace length *increases* CTL-setup margin at IC_1 and *increases* data-hold margin at IC_2 at the expense of *decreased* CTL-hold margin at IC_1 and *decreased* data-setup margin at IC_2.
- Increasing the TOF for a signal increases its hold-time margin at the expense of setup margin.

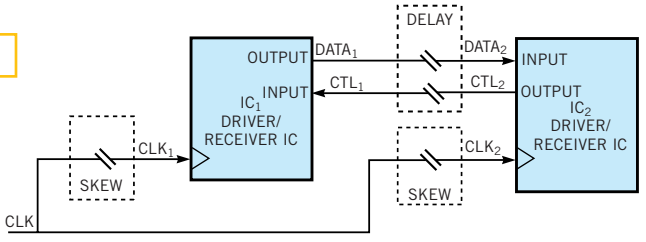
In general, when both devices' T_{CO}, T_{SU}, and T_H specifications are the same, setting T_{SKEWC} to 0 yields optimal timing margins. When the devices have different timing requirements, you can usually introduce clock skew to improve the timing margins. Note also that when the $T_{CO(MIN)}$ and TOF are both small, meeting input-hold-time requirements can be difficult. This observation suggests that closer is not necessarily better from a timing standpoint. In any case, when timing margins are tight, you should always do a detailed analysis to verify timing and to establish detailed layout instructions.

Figure 4



The timing diagram for Figure 3 includes additional signals to represent clock and data timing at the receiver as well as at the transmitter.

Figure 5



This straightforward extension of Figure 4 connects a control signal from IC_2 to IC_1 to enable bidirectional signaling.

ESTIMATING TIME OF FLIGHT

To determine and optimize timing margins, you need a reliable method for estimating TOF. Initially apply this method to estimate preliminary timing margins from parts-placement data and then later to create pc-board-layout instructions. The two primary components in TOF are propagation delay in the pc-board traces and capacitive-loading delay. Trace delay varies only slightly because of variations in the pc-board dielectric and characteristic impedance.

You can usually get results accurate to within about 10% by multiplying 170 psec/in. (about 6 in./nsec) by the trace length in inches. Accuracy is important, because you use trace length to control clock skew and TOF delay when you optimize timing margins.

Capacitive loading, however, wreaks havoc in designs by causing waveform distortion and introducing delays that depend on bus topology and driver characteristics. Loading delays are less predictable than propagation delays because of variations in capacitance, but these delays always increase TOF, leading to another important point: TOF delays caused by capacitive loading always decrease setup margin and increase hold margin.

In TOF calculations, don't combine loading delays with propagation delays; estimate them separately and then provide sufficient setup margin in the final design to compensate for the delays. For complicated bus structures that have multiple loads and tight timing margins or when driving SDRAMs and DIMMs, a transmission-line simulator is the best way to ensure accurate TOF calculations. However, for simple bus structures with light loads, you can simply allow 50 psec of additional setup margin for each picofarad of load capacitance. The following analysis uses this rule of thumb and defers detailed TOF calculations.

SPREADSHEET ANALYSIS

You now have a model that you can use to perform detailed timing analyses on most high-speed digital designs, but the associated timing diagrams are confusing, and creating them is time-consuming. The equations for timing margins can save a lot of time because you can use them in a spreadsheet analysis, such as the one in **Table 1**. You can download the Excel spreadsheet in the **table** from the web version of this article at www.edn.com.

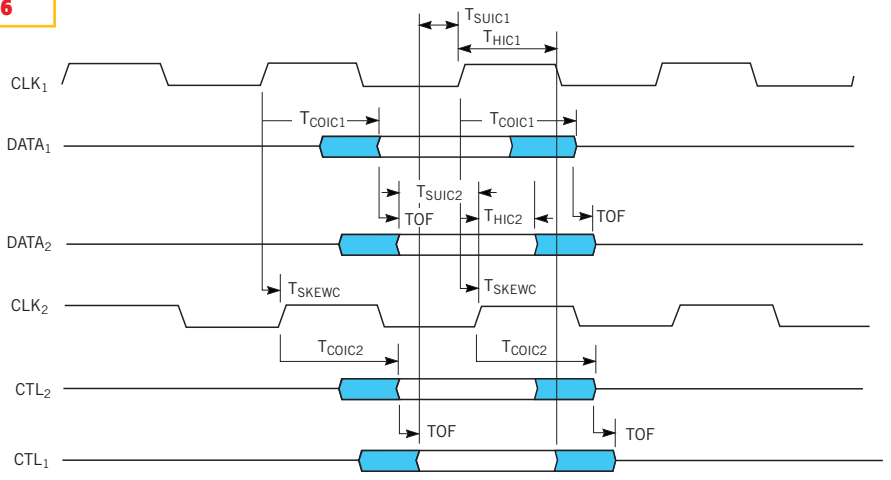
In this spreadsheet, the equations for setup-and-hold margin are located in the cells at the bottom of the page. These equations reference T_{SU}, T_H, T_{SKEW}, TOF, and clock period from the user input area (colored cells). You can easily perform a what-if analysis by entering different values of TOF and clock skew and observing the resulting setup-and-hold margins.

Using this spreadsheet, enter the manufacturer's timing parameters, estimate TOF using preliminary placement data, and then adjust the clock skew to optimize the timing margins. You can summarize the spreadsheet design procedure as follows:

1. Enter the clock frequency and each device's manufacturer-specified values of $T_{CO(MIN)}$, $T_{CO(MAX)}$, T_{SU}, and T_H.
2. Using the preliminary parts-placement and pc-board-constraint data, estimate the trace length and TOF for the signals of interest. Enter the TOF into the spreadsheet.
3. Adjust the clock skew (and possibly the TOF) to optimize the timing margins for both devices.
4. Create pc-board-layout instructions to implement the required clock skew and TOF delays.

The values in this spreadsheet example come from a design that uses a Texas Instruments TMS320C6211 DSP with PC100 SDRAM operating at 100 MHz. For this design, the microprocessor is close to the SDRAM, so you can use a minimum value of 0.25 nsec for TOF, which corresponds to a minimum trace length of about 1.5 in. Because capacitive loading increases the hold margin and decreases the setup margin, the strategy is to maximize the setup margin and then estimate the additional capacitive loading and trace delay that the design supports. Ideally, you want enough margin to use an automatic router on the SDRAM-data and -control signals.

Figure 6



The timing diagram for the bidirectional-signaling model of Figure 5 includes signals CTL_1 and CTL_2, which represent the control signals from IC_2 to IC_1.

Suppose that instead of requiring the pc-board designer to match data- and control-signal lengths (a time-consuming task), you constrain these lengths to fall between 1.5 and 4 in. (You might expect this type of variation in trace length from the output of an automatic routing program.) To analyze the timing for these constraints, you can examine the worst-case margins for the longest and shortest traces with various values of clock skew. You can then choose a clock skew that gives acceptable overall margins.

Table 2 lists the setup-and-hold margins for several values of clock skew for TOF values of 0.25 nsec (1.5 in.) and 0.68 nsec (4 in.). Recall that you chose these two TOF values as minimum and maximum trace lengths for the SDRAM-data and -control signals. The **table** shows that IC_2's setup-and-hold margins are the limiting parameters. The highlighted rows show the setup-and-hold margins for the 1.5- and 4-in. traces with 0.5 nsec of positive clock skew. If you introduce 0.5 nsec of positive clock skew between IC_1 and IC_2, you have worst-case margins of 1.32 and 0.45 nsec for setup-and-hold time, respectively.

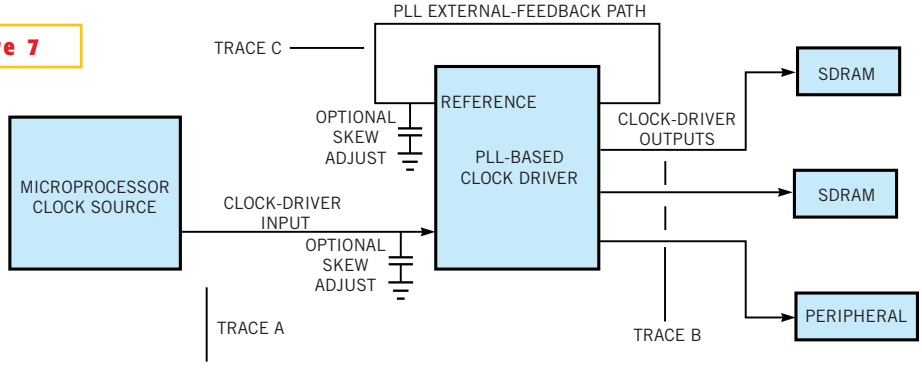
CLOCK SKEW AND TOF

After determining timing margins and the required implementation delays, your next task is to design the clock architecture and create pc-board-layout instructions. In general, the best way to make clock-skew and TOF adjustments is to control the length of the associated signal traces during pc-board layout. Controlling trace lengths is a painful task for pc-board designers. In addition, one of the primary design goals is to minimize pc-board-layout constraints. Careful planning during parts placement is particularly important to achieving this goal.

Figure 7 shows the clock architecture for a typical 100-MHz-microprocessor SDRAM interface. The microprocessor supplies a bus clock, which serves as the timing reference for all data and control signals. A PLL clock driver distributes this clock to the SDRAMs and other peripherals. In this architecture, the PLL contains an external feedback path, which is the key to controlling clock skew between the microprocessor and its peripherals.

The PLL clock driver's output clocks advance in time by an amount equal to the feedback-path trace delay. You can compensate for the propagation delay from

Figure 7



In this clock architecture for a typical 100-MHz-microprocessor SDRAM interface, the microprocessor supplies a bus clock, which serves as the timing reference for all data and control signals.

the microprocessor to the peripherals to achieve zero clock skew, or you can introduce positive or negative clock skew by controlling the length of traces A, B, and C as follows:

Case 1: C=A+B; skew=0. In Case 1, the feedback path exactly compensates for the clock delay from the microprocessor to the peripherals. The clock driver's outputs advance so that the clock signals arrive at the SDRAM at the same time they leave the microprocessor.

Case 2: C<A+B; Skew=(A+B−C)• Delay_per_unit_distance. In Case 2, the feedback path undercompensates for the clock delay from the microprocessor to the peripherals, and the clock signals arrive at the SDRAM after they leave the microprocessor.

Case 3: C>A+B; Skew=(A+B−C)• Delay_per_unit_distance. In Case 3, the feedback path overcompensates for the clock delay from the microprocessor to the peripherals, and the clock signals arrive at the SDRAM before they leave the microprocessor. Assuming that you use this clock architecture to implement the timing shown in **Table 1**, you can use a propagation delay of 170 psec/in. to produce the following layout instructions:

1. Match trace B to within 0.5 in. and route A, B, and C so that (A+B−C)= 2.75 to 3.25 in.
2. Route the bus's data and control signals so that each is 1.5 to 4 in. long.

This implementation results in clock skew of 0.47 to 0.55 nsec and TOF of 0.25 to 0.68 nsec.

Assuming clock jitter of 200 psec, in the worst case, you still have an approximately 1.75−(0.68−0.25)−0.2=1.1-nsec setup margin and 0.45−0.2=0.25-nsec hold margin. Using a value of 50 psec/pF to estimate the maximum delay from capacitive loading allows enough setup margin for bus loads of 22 pF or less. If you expect bus loading to exceed this value, you must perform a more detailed analysis, increase layout constraints, or do both. For bus loads of less than 22 pF, manually routing the clock signals should allow you to automatically route the data and control signals.

HOW MUCH MARGIN?

To determine the amount of setup-and-hold margin required for reliable operation, you need to consider three primary sources of uncertainty: cycle-to-cycle clock jitter, trace-length delay, and capacitive-loading delay.

All clock sources introduce cycle-to-cycle clock jitter, which causes clock-period variation. You must allow sufficient margin to prevent this variation from causing setup-and-hold-time violations. Cycle-to-cycle clock jitter affects both setup and hold margins.

Signal-trace lengths are an important component of TOF delay. You can control this uncertainty when you create pc-board-layout instructions. You generally want to allow for as much layout uncertainty as possible to minimize layout constraints, particularly if you plan to use automatic routing.

The second component of TOF delay is capacitive-loading delay. You should provide sufficient setup margin to account for the worst-case capacitive-loading delay. For complicated or heavily loaded bus structures, you should use a simulator to estimate capacitive-loading delay.

If you identify and optimize margins early in the design, you can allocate them to minimize cost, time to market, or both. As development progresses, this optimization enables you to make intelligent trade-offs between parts cost, design time, and pc-board-layout time.□

AUTHOR'S BIOGRAPHY

Bob Kirstein is president of Stratus Engineering (www.stratusengineering.com), San Diego, an electronic design-consulting company, where he has worked for the past eight years. He holds a BS in computer engineering from the University of California—San Diego. His 18 years' experience includes designing analog and digital boards and firmware and implementing logic in ASICs and FPGAs. Among the products he has developed are several high-performance, microprocessor-based networking switches, a robotics motion controller, an automotive collision-warning system, and an FFT-based vibration analyzer. He enjoys biking, swimming, tennis, and spending time with his wife, Janice, and eight-year-old daughter, Jacquelyn. You can reach him at rkirstein@stratusengineering.com.

射 频 和 天 线 设 计 培 训 课 程 推 荐

易迪拓培训(www.edatop.com)由数名来自于研发第一线的资深工程师发起成立，致力并专注于微波、射频、天线设计研发人才的培养；我们于 2006 年整合合并微波 EDA 网(www.mweda.com)，现已发展成为国内最大的微波射频和天线设计人才培养基地，成功推出多套微波射频以及天线设计经典培训课程和 ADS、HFSS 等专业软件使用培训课程，广受客户好评；并先后与人民邮电出版社、电子工业出版社合作出版了多本专业图书，帮助数万名工程师提升了专业技术能力。客户遍布中兴通讯、研通高频、埃威航电、国人通信等多家国内知名公司，以及台湾工业技术研究院、永业科技、全一电子等多家台湾地区企业。

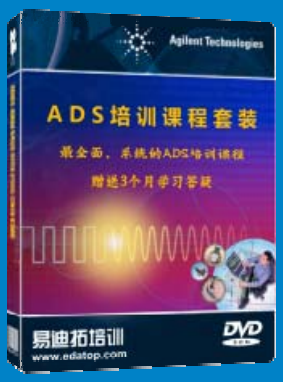
易迪拓培训课程列表：http://www.edatop.com/peixun/rfe/129.html

射频工程师养成培训课程套装

该套装精选了射频专业基础培训课程、射频仿真设计培训课程和射频电路测量培训课程三个类别共 30 门视频培训课程和 3 本图书教材；旨在引领学员全面学习一个射频工程师需要熟悉、理解和掌握的专业知识和研发设计能力。通过套装的学习，能够让学员完全达到和胜任一个合格的射频工程师的要求…

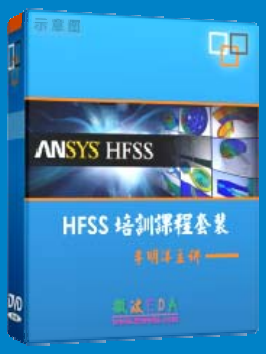
课程网址：http://www.edatop.com/peixun/rfe/110.html

ADS 学习培训课程套装



该套装是迄今国内最全面、最权威的 ADS 培训教程，共包含 10 门 ADS 学习培训课程。课程是由具有多年 ADS 使用经验的微波射频与通信系统设计领域资深专家讲解，并多结合设计实例，由浅入深、详细而又全面地讲解了 ADS 在微波射频电路设计、通信系统设计和电磁仿真设计方面的内容。能让您在最短的时间内学会使用 ADS，迅速提升个人技术能力，把 ADS 真正应用到实际研发工作中去，成为 ADS 设计专家...

课程网址： http://www.edatop.com/peixun/ads/13.html



HFSS 学习培训课程套装

该套课程套装包含了本站全部 HFSS 培训课程，是迄今国内最全面、最专业的 HFSS 培训教程套装，可以帮助您从零开始，全面深入学习 HFSS 的各项功能和在多个方面的工程应用。购买套装，更可超值赠送 3 个月免费学习答疑，随时解答您学习过程中遇到的棘手问题，让您的 HFSS 学习更加轻松顺畅…

课程网址：http://www.edatop.com/peixun/hfss/11.html

CST 学习培训课程套装

该培训套装由易迪拓培训联合微波 EDA 网共同推出，是最全面、系统、专业的 CST 微波工作室培训课程套装，所有课程都由经验丰富的专家授课，视频教学，可以帮助您从零开始，全面系统地学习 CST 微波工作的各项功能及其在微波射频、天线设计等领域的设计应用。且购买该套装，还可超值赠送 3 个月免费学习答疑…

课程网址：http://www.edatop.com/peixun/cst/24.html

HFSS 天线设计培训课程套装

套装包含 6 门视频课程和 1 本图书，课程从基础讲起，内容由浅入深，理论介绍和实际操作讲解相结合，全面系统的讲解了 HFSS 天线设计的全过程。是国内最全面、最专业的 HFSS 天线设计课程，可以帮助您快速学习掌握如何使用 HFSS 设计天线，让天线设计不再难…

课程网址：http://www.edatop.com/peixun/hfss/122.html

13.56MHz NFC/RFID 线圈天线设计培训课程套装

套装包含 4 门视频培训课程，培训将 13.56MHz 线圈天线设计原理和仿真设计实践相结合，全面系统地讲解了 13.56MHz 线圈天线的工作原理、设计方法、设计考量以及使用 HFSS 和 CST 仿真分析线圈天线的具体操作，同时还介绍了 13.56MHz 线圈天线匹配电路的设计和调试。通过该套课程的学习，可以帮助您快速学习掌握 13.56MHz 线圈天线及其匹配电路的原理、设计和调试…

详情浏览：http://www.edatop.com/peixun/antenna/116.html

我们的课程优势：

※ 成立于 2004 年，10 多年丰富的行业经验，

※ 一直致力并专注于微波射频和天线设计工程师的培养，更了解该行业对人才的要求

※ 经验丰富的一线资深工程师讲授，结合实际工程案例，直观、实用、易学

联系我们：

※ 易迪拓培训官网：http://www.edatop.com

※ 微波 EDA 网：http://www.mweda.com

※ 官方淘宝店：http://shop36920890.taobao.com