

Table of Contents

Introduction
Overview of SmartTime Timing Analysis
Timing Analysis for Generated Clocks
Inter-Clock Domain Analysis with Two Asynchronous Clocks
Inter-Clock Domain Analysis for Generated Clocks
Analyzing Source Synchronization
Analyzing Design with Jitter/Clock Uncertainty in SmartTime
Analyzing a Multicycle Path with Single Clock Domain
Analyzing a Multicycle Path with Inter-Clock Domain
Appendix A: Applying a Clock Constraint
Appendix B: Applying a Generated Clock Constraint
Appendix C: Enabling Inter-Clock Domains Analysis
Appendix D: Applying a Multicycle Clock Constraint

Introduction

Complex and sophisticated clocking schemes and exceptions are currently used in low power and highreliability Microsemi FPGA devices. Increasing complexity results in the need for more timing analysis capabilities that will be required for sign-off and validation. The SmartTime FPGA timing analysis tool, available in the Microsemi Libero[®] Integrated Design Environment (IDE) software suite, allows you to do the basic timing analysis for simple clocking schemes as well as the required analysis of complex clocking schemes. This application note describes advanced timing analysis with detailed steps using the Microsemi SmartTime FPGA timing analysis tool.

This document gives a quick overview of timing analysis using the SmartTime tool and then provides an example of advanced timing analysis as listed below:

- 1. Timing analysis for a generated clock
- 2. Inter-clock domain analysis with two asynchronous clocks
- 3. Inter-clock domain analysis for generated clocks
- 4. Analyzing source synchronization
- 5. Analyzing a design with jitter/clock uncertainty in SmartTime
- 6. Analyzing a multicycle path with a single clock domain
- 7. Analyzing a multicycle path with inter-clock domain
- 8. Analyzing clock gating
- 9. Four corner analysis



Overview of SmartTime Timing Analysis

SmartTime is the gate-level static timing analysis (STA) tool for SmartFusion[®] customizable system-onchip (cSoC), RTAX[™]-S/SL, Fusion[®], IGLOO[®], ProASIC[®]3, Axcelerator[®], eX, and SXA families. The SmartTime graphical user interface (GUI) provides the SmartTime Timing Analyzer for static timing analysis and SmartTime Constraints Editor for applying SDC constraints in the design.

The SmartTime Timing Analyzer has two timing analysis views: Maximum Delay Analysis $\boxed{2}$ and Minimum Delay Analysis $\boxed{2}$. The maximum delay analysis view checks the setup timing and the minimum delay analysis checks the hold timing. SmartTime constraints editor enables you to create, view, and edit the timing constraints of the selected scenario for use with SmartTime timing analysis.

The setup check in SmartTime involves comparing the latest data arrival time (longest data path delay) with the earliest required time (shortest clock path delay). The hold check in SmartTime involves comparing the earliest data arrival time (shortest data path delay) with the latest required time (longest clock path delay). Both setup and hold checks calculate the timing delay with respect to launched edge and captured edge, as shown in Figure 1 and Figure 2 on page 3. This is the base for all timing analysis and also used for all advanced timing analysis. Refer to the *SmartTime* Tutorial to understand basic timing analysis using the SmartTime tool.



• Required Time = Captured Edge (T) + Min. Clock to FF2 - Setup of FF2

Figure 1 • Setup Check Calculation





Hold Check Calculation

- Arrival Time = Launched Edge (0) + Min. Clock to FF1 + Min. Data Path
- Required Time = Captured Edge (0) + Max. Clock to FF2 + Hold of FF2

Figure 2 • Hold Check Calculation

The following section describes various methods of advanced timing analysis.

Timing Analysis for Generated Clocks

Many designs have clocks that are generated internally via phase-locked loop (PLL), clock divider, or other allowed methods. The SmartTime tool allows you to generate the clock constraints for the internally generated clocks and verifies their timing behavior. You need to apply a clock constraint on the main clock. For the clock generated via PLL, SmartTime creates the constraints for the generated clocks and applies it automatically during timing analysis. For the clock generated via clock divider, you need to manually apply the generated clock constraint.

Consider the design example shown in Figure 3. CLKA is the main clock, running at 50 MHz. PLL_50_20_0/Core:GLA and DFN1_0:Q are generated via PLL and clock divider. The following section describes the timing analysis for these two generated clocks.



Figure 3 • Design Example for Generated Clock



Analyzing Generated Clock Domain Timing with SmartTime

1. Specify the reference clock frequency and other attributes. Refer to "Appendix A: Applying a Clock Constraint" on page 35 for creating a clock constraint using the GUI.

File Ealt view Actions Tools window Help											
	വല	2	<u>_</u> _	2 3	n 🐜	🖏 🌬 🏅	<u>8</u> × >	₩ ۳	ſ <u>ŗ</u> 🏠	🏪 🕂	3
∃ Constraints ⊡ Bequirements		Syntax	Clock Name	Clock Source	Period (ns)	Frequency (MHz)	Dutycycle (%)	First Edge	Offset (ns)	Waveform	File
Clock		Click here	e to add a const	raint							
🕂 Generated Clock	1	٣	CLKA	CLKA	20.000	50.000	50.000	rising	0.000	010	GU
······ Input Delay											
Output Delay											
Exceptions Max Delay											
Exceptions Max Delay Min Delay											

Figure 4 • Clock Constraints Using Constraints Editor and SDC

On applying the reference clock constraint, the generated clock constraint for the PLL will be created by the SmartTime tool automatically. SmartTime reads the netlist that has the PLL divider setting and then automatically populates the divider ratio. However, you still need to identify other generated clocks and apply generated clock constraints.



Figure 5 • Automatically Generated PLL Clock Constraint in Constraints Editor



2. Identify the generated clock and apply the generated clock constraint. Refer to "Appendix B: Applying a Generated Clock Constraint" on page 37 for creating a generated clock constraint using the GUI.

The second								
	<u></u>	2	N 🖻 🔁 📄	🛂 📶 🦗 🖗 🧎	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	꼬 꼬 꼬		
Constraints		Syntax	Clock Name	Clock Pin	Reference Pin	Multiplier	Divider	Waveform
- Requirements		Click here	to add a constraint					
Clock	1	٣	PLL_50_20_0/Core:GLA	PLL_50_20_0/Core:GLA	PLL_50_20_0/Core:CLKA	8	20	synchronized
Generated Clock	2	٣	DFN1_0:Q	DFN1_0:Q	DFN1_0:CLK	1	2	synchronized
Input Delay								
Output Delay								
reate generated cl	ock	-n=	ma/PT.T. 50 20	$) 0/Core \cdot GLA $	-divide by 20	-multip	lv hv	8 -

Figure 6 • Generated Clock Constraint

The maximum delay analysis view displays the timing analysis for the reference clock, CLKA, and the two generated clocks, PLL_50_20_0/Core:GLA and DFN1_0:CLK.



Figure 7 • Maximum Delay Analysis Showing All Clocks



The timing analysis for the internally generated clocks is shown in Figure 8.

Summary for path From: Count16 1/DEN1C1 NUL 0-	CIK						
To: Count16 1/DFN1E1C1 NU 1	2/U1:D						
Data Required Time (ns) Data Arr	ival Time (ns) S	lack (ns)					
43.354 11.851	31	.503					
Path details							
Pin Name	Туре	Net Name	Cell Name	Op	Delay (ns)	Total (ns)	Fanout E
Data arrival time calculation							
DFN1_0:Q					0.000	0.000	
DFN1_0:Q	Clock source			+	0.000	0.000	r
	Clock generation			+	1.963	1.963	
DFN1_0_RNIKE25:A	net	DFN1_0_Q_i		+	0.759	2.722	1
DFN1_0_RNIKE25:Y	cell		ADLIB:CLKINT	+	0.558	3.280	17 r
Count16_1/DFN1C1_NU_0:CLK	net	DFN1_0_Q		+	0.502	3.782	1
Count16_1/DFN1C1_NU_0:Q	cell		ADLIB:DFN1C1	+	0.434	4.216	4 r
Count16_1/U_AND3_0_1_2:A	net	Count16_1/DFN1C1_NU_0		+	1.142	5.358	1
Count16_1/U_AND3_0_1_2:Y	cell		ADLIB:AND3	+	0.392	5.750	7 r
Count16_1/U_U_AND3_0_to_8:A	net	Count16_1/NU_0_1_2		+	2.749	8.499	1
Count16_1/U_U_AND3_0_to_8:Y	cell		ADLIB:AND3	+	0.392	8.891	1 8 r
Count16_1/DFN1E1C1_NU_12/U0:S	net	Count16_1/NU_0_to_8		+	2.441	11.332	1
Count16_1/DFN1E1C1_NU_12/U0:Y	cell		ADLIB:MX2	+	0.278	11.610	1 f
Count16_1/DFN1E1C1_NU_12/U1:D	net	Count16_1/DFN1E1C1_NU_12/Y		+	0.241	11.851	f
data arrival time						11.851	
Data required time calculation						,	
DFN1_0:Q	Clock Constraint				40.000	40.000	
DFN1_0:Q	Clock source			+	0.000	40.000	1
	Clock generation			+	1.963	41.963	
DFN1_0_RNIKE25:A	net	DFN1_0_Q_i		+	0.759	42.722	1
DFN1_0_RNIKE25:Y	cell		ADLIB:CLKINT	+	0.558	43.280	<u>17 r</u>
Count16_1/DFN1E1C1_NU_12/U1:CLK	net	DFN1_0_Q		+	0.502	43.782	1
Count16_1/DFN1E1C1_NU_12/U1:D	Library setup time		ADLIB:DFN1C1	-	0.428	43.354	
data required time						43.354	

Figure 8 • Setup Check for the Generated Clock

Note: SmartTime automatically calculates the clock generation delay. Figure 9 on page 7 shows the calculation of the delay from the CLKA port to the output pin of the clock divider.

	CLKA					0.000	0.000		
	CLKA	Clock source			+	0.000	0.000		r
1	CLKA_pad/U0/U0:PAD	net	CLKA		+	0.000	0.000		r
1	CLKA_pad/U0/U0:Y	cell		ADLIB:IOPAD_IN	+	0.758	0.758	1	r
1	CLKA_pad/U0/U1:A	net	CLKA_pad/U0/NET1		+	0.000	0.758		r
1	CLKA_pad/U0/U1:Y	cell		ADLIB:CLKIO	+	0.260	1.018	2	r
1	DFN1_0:CLK	net	CLKA_c		+	0.511	1.529		r
	 DEN1 0.0	cell		ADLIB-DEN1	+	0.434	1 963	1	r



Figure 9 • Delay Calculation for Clock Generation

Inter-Clock Domain Analysis with Two Asynchronous Clocks

SmartTime enables inter-clock domain timing checks for designs containing functional paths that exist across two clock domains (the register launching the data and the register capturing the data are clocked by two asynchronous clock sources). Accurate specifications of both clocks are required to allow a valid inter-clock domain timing check.

Note: The default SmartTime setting does not show inter-clock domain analysis. You need to change the setting (see "Appendix C: Enabling Inter-Clock Domains Analysis" on page 40) to enable the interclock domain analysis. Depending on the design, some of the inter-clock domain paths are valid timing paths and some are false paths. It is the designer's responsibility to identify these paths and apply the timing exception as needed.

For an inter-clock domain path, SmartTime analyzes the relationship between all the active clock edges over a common period equal to the least common multiple of the two clock periods. For a setup check, the tightest relation of launch to capture is considered to ensure that the data arrives before the capture edge.



The hold check verifies that a setup relationship is not overwritten by a following data launch. The clock edge used for setup and hold analysis is shown in Figure 10.



Figure 10 • Clock Relationship for Inter-Domain Clocks

Consider the inter-domain design example shown in Figure 11. Note the path from the CLK1 domain to the CLK2 domain, which is a valid inter-clock domain path. Assume that CLK1 is 100 MHz and CLK2 is 75 MHz and both have zero offset. The "Analyzing Inter-Clock Domain Timing with SmartTime" section on page 9 shows how to analyze this cross-clock domain path.



Figure 11 • Inter-Clock Domain Example

Analyzing Inter-Clock Domain Timing with SmartTime

1. Specify the clock frequency and other attributes for both reference clocks. Refer to "Appendix A: Applying a Clock Constraint" on page 35 for creating a generated clock constraint using the GUI.

Constraints		Syntax	Clock Name	Clock Source	Period (ns)	Frequency (MHz)	Dutycycle (%)	First Edge	Offset (ns)	Waveform	File
		Click here	e to add a const	raint							
Generated Clock	1	٣	CLK1	CLK1	10.000	100.000	50.000	rising	0.000	05	GUI
Input Delay	2	٣	CLK2	CLK2	13.333	75.000	50.000	rising	0.000	0 6.66665	GUI
Output Delay											
create_clock -name	{CL]	K1}-p	eriod 1	0.000-w	avef	orm{0.	000 5.	000	}{CI	-K1}	
create_clock -name	{CL]	K2}-p	period 1	3.333-w	avef	orm{0.	000 6.	667	}{CI	JK2 }	

Figure 12 • Clock Constraint Using Constraints Editor and SDC

2. Enable inter-clock domain analysis. Refer to "Appendix C: Enabling Inter-Clock Domains Analysis" on page 40.

The maximum delay analysis view displays the timing analysis for CLK1 to CLK2 under CLK2 domain analysis, as shown in Figure 13.

CLK1	Source Pin	Sin	k Pin Del	ay Slack /	(ns)	Required	Setup	
Register to Register 🛛 🖪	na:CLK	ac:D	1	769 1 1 32	3 272	2 4 40	4 0 428	
Setup	2 ab:CLK	ac:D	1	412 1.489	2.915	5 4.40	4 0.428	
sk to Output	4=· - =· ·	14						
to Asynchron Recovery	Details for path From: qa:CLK To: qc:D							
to negi	Pin Name	Туре	Net Name	Cell Name	Op	Delay (ns)	Total (ns) F	anc
to Register								
	data required time						4.404	
	data arrival time				-		3.272	
o Asynchron	slack						1.132	
sto Regi	Data arrival time ca	lculation				0.000	0.005	
	CLK1	-			_	0.000	0.000	
		Clock source			+	0.000	0.000	
put –	CLK1_pad/UU/UU:PAD	net	ULK1		+	0.000	0.000	
	CLK1_pad/U0/U0:Y	cell		ADLIB:IOPAD_I	N +	0.747	0.747	
	CLK1_pad/U0/U1:A	net	CLK1_pad/U0/NET		+	0.000	0.747	
	CLK1_pad/U0/U1:Y	cell		ADLIB:CLKIO	+	0.260	1.007	
	qa:CLK	net	CLK1_c		+	0.496	1.503	
	qa:Q	cell		ADLIB:DFN1	+	0.550	2.053	
	qc_1:B	net	qa		+	0.241	2.294	
	qc_1:Y	cell		ADLIB:XOR2	+	0.737	3.031	
	qc:D	net	qc_1		+	0.241	3.272	
	data arrival time						3.272	
	Data required time o	alculation						
	CLK2	Clock Constraint				3.333	3.333	
	CLK2	Clock source			+	0.000	3.333	
	CLK2_pad/U0/U0:PAD	net	CLK2		+	0.000	3.333	
	CLK2_pad/U0/U0:Y	cell		ADLIB:IOPAD_I	V +	0.747	4.080	
	CLK2_pad/U0/U1:A	net	CLK2_pad/U0/NET		+	0.000	4.080	
	CLK2_pad/U0/U1:Y	cell		ADLIB:CLKIO	+	0.260	4.340	
	gc:CLK	net	CLK2_c	1	+	0.492	4.832	
	gc:D	Library setup time		ADLIB:DFN1	•	0.428	4.404	
	data required time						4 404	

Figure 13 • Inter-Clock Domain Timing Analysis in SmartTime Maximum Delay Analysis View



The clock edges and data path (longest data path is from qa to qc register) used in the setup calculation are shown in Figure 14.



Figure 14 • Clock Edges and Data Path Used in Intra-Clock Domain Setup Calculation

Similarly, the minimum delay analysis view displays the hold analysis from CLK1 to CLK2.



⊡~ 🖏 Summary								
🚽 🖧 Datasheet			Delen Cleak			1.11-14		
🖻 🗹 🐵 CLK1	Source Pin	Sink Pin	(ne) (ne)	Amvai r	nequirea (no)			
Register to Register	1 Jab:CLK	ne:D		585 1 309	0.72			
External Hold		40.0 7c:D	0.708 0.6	92 1.416	0.72	4 0.000		
Clock to Output		40.0	0.100 0.0	1.410	0.12	- 0.000		
Register to Asynchron	Details for path							
External Removal	To: go:D							
Asynchronous to Regi	Pin Name	Тире	Net Name	Cell Name		elau (ns)	Total (ns) F	anout Edge
⊢✓ 😡 CLK2		Турс				sidy (lis)	rotar (najj r	anoug Euge
Register to Register								
External Hold	data arrival time						1.309	
Clock to Output	data required time						0.724	
Begister to Asunchron	slack			1			0.585	
External Removal		1		1				
Asupchropous to Begi	Data arrival time calculat	ion						
CLK1 to CLK2	CLK1					0.000	0.000	
"S" Din to Din	CLK1	Clock source			+	0.000	0.000	1
Inside Colored	CLK1_pad/U0/U0:PAD	net	CLK1		+	0.000	0.000	r
"S" Uses Cata	CLK1_pad/U0/U0:Y	cell		ADLIB:IOPAD_IN	1 +	0.319	0.319	1 r
	CLK1_pad/U0/U1:A	net	CLK1_pad/U0/NET1		+	0.000	0.319	1
	CLK1_pad/U0/U1:Y	cell		ADLIB:CLKIO	+	0.123	0.442	2 r
	qb:CLK	net	CLK1_c		+	0.266	0.708	1
	qb:Q	cell		ADLIB:DFN1	+	0.206	0.914	1 r
	qc_1:A	net	qb		+	0.119	1.033	1
	qc_1:Y	cell		ADLIB:XOR2	+	0.158	1.191	1 f
	qc:D	net	qc_1		+	0.118	1.309	f
	data arrival time						1.309	
	Data required time calcul	ation						
	CLK2	Clock Constraint				0.000	0.000	
	CLK2	Clock source			+	0.000	0.000	r
	CLK2_pad/UU/UU:PAD	net	ULK2		+	0.000	0.000	1
	LLK2_pad/UU/UU:Y	cell		AULIB:IUPAD_IN	1 +	0.319	0.319	11
	ULK2_pad/UU/U1:A	net	ULK2_pad/UU/NET1		+	0.000	0.319	1
1 I I I I	LLK2_pad/UU/UT:Y	cell	CLK2 -	ADLIB:ULKIU	+	0.123	0.442	21
		net	ULKZ_C		+	0.282	0.724	1
2	qc:U	Library hold time		ADLIB:DENT	+	0.000	0.724	
<u></u>	uata required time						0.724	

The hold check from the CLK1 to CLK2 domain is shown in Figure 15.



The clock edges and data path (shortest data path is from qb to qc register) used in the hold calculation are shown in Figure 16.



Figure 16 • Data Path for Hold Check



Inter-Clock Domain Analysis for Generated Clocks

Designs with internally generated clocks can also have a cross-clock domain path. SmartTime enables you to specify the generated clock constraint for the internally generated clocks and then apply crossclock domain analysis. As mentioned in the previous section, it is the designer's responsibility to identify a path as valid or false and apply timing exceptions as required.

Consider the design example shown in Figure 17, where CLKA is a reference clock and DFN1_0:CLK is generated via clock divider. There is a path where data is launched from CLKA to be captured in the DFN1_0:CLK domain. The "Constraints Using GUI" section shows how to analyze the cross-clock domain between the main clock and internally generated clock.



Figure 17 • Design Example for Inter-Clock Domain Analysis Using Generated Clocks

Constraints Using GUI

1. Specify the reference clock frequency and other attributes. Refer to "Appendix A: Applying a Clock Constraint" on page 35 for creating a generated clock constraint using the GUI.

👽 File Edit View Actions Tools	Windo	w Help								
	2	2	<u>s</u> 🕺	8	👽 🖄	L 🐜 🏝	a 🔊 🎘	s (25) 🕅	[<u></u> []	<u>پر</u>
Constraints		Syntax	Clock Name	Clock Source	Period	Frequency [)utycycle	First C)ffset (ns) Way	eform File
Requirements		Click here	e to add a consi	traint	_ [II3]	[[1112]]	[10]	Lage		
Generated Clock	1	٣	CLKA	CLKA	6.667	150.000	50.000	rising I	0.000 0 3.33	333 GUI
Input Delay										
Eucoptions										
Max Delay										
Min Delay										

Figure 18 • Clock Constraint Using Constraints Editor and SDC



2. Identify the generated clock and then apply the generated clock constraint. Refer to "Appendix B: Applying a Generated Clock Constraint" on page 37 for creating a generated clock constraint using the GUI.

😲 File Edit View Actions Tools	Windo	w Help							
	29		<u> @</u> ㅈ	2 3	n 🦗 📴 🍋 🥻	<u> </u>	ı ‱ ‱		
⊡ Constraints		Syntax	Clock Name	Clock Pin	Reference Pin	Multiplier	Divider	Waveform	File
⊟ Requirements		Click her	e to add a const	raint					
🕈 Clock	1	٣	DFN1_0:Q	DFN1_0:Q	DFN1_0:CLK	1	2	synchronized	GUI
······ Generated Clock ······ Input Delay ······ Output Delay								1	
reate_generated_c	lock	-na	me{DFN1	_0:Q}-d	ivide_by2 -s	ource{DF1	N1_0:	CLK}	

Figure 19 • Generated Clock Constraint

3. Enable inter-clock domain analysis. Refer to "Appendix C: Enabling Inter-Clock Domains Analysis" on page 40.

The maximum delay analysis view displays the timing analysis from CLKA to DFN1_0:Q under DFN1_0:Q domain analysis, as shown in Figure 20 on page 14. SmartTime calculates clock generation delays and clock constraints using clock edges between the clocks automatically.





Figure 20 • Setup Check for Inter-Clock Domain Clock Using Generated Clock



Analyzing Source Synchronization

This section describes the techniques for constraining and analyzing source-synchronization. Sourcesynchronous clocking refers to the technique of sourcing a clock along with the data. The timing of unidirectional data signals is referred to a clock sourced by the same device that generates the signals. Constraining source-synchronous interfaces can be complex. In addition to using the reference clock constraint, you need to constrain the source synchronous outputs by specifying the output delay relative to the reference clock.

Refer to the Source-Synchronous Clock Designs: Timing Constraints and Analysis application note to understand source-synchronous clock design timing constraints and analysis in detail.

Analyzing Design with Jitter/Clock Uncertainty in SmartTime

SmartTime uses the relationship between launched clock edge and captured clock edge during interclock domain timing analysis. However, the non-idealities of the clock generation and clock distribution system, also called jitter, manifest themselves as uncertainties of the clock edge arrivals. The clock-toclock uncertainty constraint in SmartTime enables you to specify these uncertainties between different clocks. Clock-to-clock uncertainty defines the timing uncertainty between two clock waveforms or maximum clock skew.



Figure 21 • Clock-to-Clock Uncertainty

A design example with two external clocks, CLK1 and CLK2, is shown in Figure 22. Assume that these two clocks have a tracking jitter of 2 ns. During timing analysis, this tracking jitter can be added as clock_uncertainty.



Figure 22 • Example of Inter-Clock Uncertainty for Rise-Rise Setup Check



Analyzing a Design with Clock Uncertainty

1. Specify the clock frequency and other attributes for both reference clocks. Refer to "Appendix A: Applying a Clock Constraint" on page 35 for creating a generated clock constraint using the GUI.

SmartTime [Top] - [Constraints E	ditor]								
💱 File Edit View Actions Tools Wind	low Help								
		<u>s @</u> z	2 3	n he	<u>⊗</u> * <u>+</u> ⊠	<u> </u>	<mark>۳</mark>	Şr 🏠	🏪 🖑
Constraints Bequirements	Syntax	Clock Name	Clock Source	Period (ns)	Frequency (MHz)	Dutycycle (%)	First Edge	Offset (ns)	Waveform
Clock	Click here	e to add a const	raint						
Generated Clock	٣	CLK2	CLK2	10.000	100.000	50.000	rising	0.000	05
Input Delay 2	*	CLK1	CLK1	10.000	100.000	50.000	rising	0.000	05
Output Delay									
Max Delay									
Multicucle									
False Path									
create clock -name{CI	.K2}-r	period 1	10 000 - w	avef	orm{0	000 5	000	} { CT	K2}
areato aloak name (CI	121) r	oriod 1	10.000 **	aver	orm (0	000 5) (CT	1721)
create_clock -name{Cl	'VT } – È	period -	LU.UUU-W	avei	orm(0.	000 5.	.000	} { CL	12/13

Figure 23 • Clock Constraint Using Constraints Editor and SDC

2. Add the clock-to-clock uncertainty constraint by clicking the 📰 button on the toolbar and applying 2 ns of clock uncertainty between CLK1 and CLK2.

Set Clock-to-clock	Uncertainty Const	raint	×
From Clock: CLK1			•
Edge	C rising	C falling	both
To Clock: CLK2			▼
Edge	C rising	C falling	 both
Uncertainty: 2	ns		
Use uncertainty for:	C setup checks	C hold checks	I all checks
Comment:			
1			
Help			OK Cancel
et_clock_u	ncertainty	l-from{CLK	<pre>K1}-to{CLK2}</pre>

Figure 24 • Applying Clock-to-Clock Uncertainty Constraint



3. Enable inter-clock domain analysis. Refer to "Appendix C: Enabling Inter-Clock Domains Analysis" on page 40.

SmartTime timing analysis view uses the clock-to-clock uncertainty constraint for timing checks. Figure 25 shows how the clock-to-clock uncertainty constraint is used in a setup check.

	Source Pin	Sink Pin	Delay Slack	Arrival F	Required	1 Setup		
ister 🗖			(ns) (ns)	(ns)	[ns]	(ns)		
etun	DFN1_U:CLK D	FN1_1:D	2.776 4.6	56 5.044	9.7	00 0.71	1	
nchron	Details for path							
	From: DFN1_0:CLK							
eni 🗕	Din Name	Tuno	NotName			Dolay (no)	Total (no)	anout Fr
		Туре	<u>net name</u>			Delay (risj	i utai (risjj r	anouger
er i i								
	data required time			1			9 700	
	data arrival time			1			5.044	
n -	slack			1			4.656	
			1	1				
, Begi	Data arrival time calculation	on						
	CLK1					0.000	0.000	
	CLK1	Clock source			+	0.000	0.000	1
	CLK1_pad/U0/U0:PAD	net	CLK1		+	0.000	0.000	1
	CLK1_pad/U0/U0:Y	cell		ADLIB:IOPAD_I	N +	1.001	1.001	1 r
	CLK1_pad/U0/U1:YIN	net	CLK1_pad/U0/NET1		+	0.000	1.001	1
	CLK1_pad/U0/U1:Y	cell		ADLIB:IOIN_IB	+	0.043	1.044	1 r
	DFN1_0:CLK	net	CLK1_c		+	1.224	2.268	r
	DFN1_0:Q	cell		ADLIB:DFN1	+	0.654	2.922	1 f
	BUFD_0:A	net	DFN1_0_Q		+	0.308	3.230	f
	BUFD_0:Y	cell		ADLIB:BUFF	+	0.537	3.767	1 f
	AND2_0:B	net	BUFD_0_Y		+	0.329	4.096	f
	AND2_0:Y	cell		ADLIB:AND2	+	0.631	4.727	1 f
	DFN1_1:D	net	AND2_0_Y	ļ	+	0.317	5.044	f
	data arrival time						5.044	
		•						
	Data required time calcula	Cleak Canabaint	1	1		10.000	10.000	
		Clock Constraint			+. +	10.000	10.000	
	l olook to olook uncertainty	LIUCK SOUICE			+	2,000	9 000	
	CLK2_pad/U0/U0:PAD	net	CLK2		-	2.000	8,000	
		net	ULNZ		T N +	1.001	9.000	1
	CLK2_pad/00/00.1	uell net	CLK2_pad/U0/NET1	AULID.IUFAU_I	1N +	0.000	9.001	1 1
	CLK2_pad/00/01.11M	cell			-	0.000	9.001	1.
		net	CLK2 c			1 369	10 412	
	DEN1 1:D	Libraru setup time	OENZ_0	ADLIB-DEN1	- T	0.713	9 700	
		Library setup time		ACCOUNT NO		0.713	0.700	

Figure 25 • Setup and Hold Check

Note: If you use a PLL in the design, SmartTime automatically adds the clock uncertainty between the PLL reference clock and the PLL output clock. However, SmartTime will not add clock uncertainty between the output clocks (the output clocks are generated from the same VCO clock). Figure 26 on page 18 shows a design example with data paths from the PLL reference clock and PLL output clocks and also on the PLL output clocks. When adding the reference clock constraint, the generated clock and clock uncertainty constraint are added by the tool automatically.





create_generated_clock -name{PLL_40_50_20_0/Core:GLA}-divide_by40 multiply_by20 -source{PLL_40_50_20_0/Core:CLKA}{PLL_40_50_20_0/Core:GLA}

create_generated_clock -name{PLL_40_50_20_0/Core:GLB}-divide_by16 multiply_by20 -source{PLL_40_50_20_0/Core:CLKA}{PLL_40_50_20_0/Core:GLB}

Figure 26 • Design Example and Constraint for Clock-to-Clock Uncertainty Using a PLL Design

Analyzing a Multicycle Path with Single Clock Domain

Multicycle paths are data paths that may need more than one clock cycle to latch data at the captured register. The multicycle path constraint enables you to move the captured clock edge forward or the launched clock edge backward. When a multicycle constraint is applied to setup, it modifies the setup relationship by moving the captured (destination) clock edge to the right.

Similarly, when a multicycle constraint is applied to hold, it modifies the hold relationship, changing the launched (source) clock edge to the left. Applying the multicycle path constraint requires design knowledge.



Figure 27 • Setup and Hold Check on Multicycle Path

Figure 28 shows a design example where you assume that the path from DFN1_1 to DFN1_2 is a multicycle path. The "Analyzing a Multicycle Path with SmartTime" section on page 20 shows how to apply and analyze the multicycle constraint on this design.



Figure 28 • Design with Multicycle Path



Analyzing a Multicycle Path with SmartTime

1. Specify the clock frequency and other attributes for the reference clock. Refer to "Appendix A: Applying a Clock Constraint" on page 35 for creating a generated clock constraint using the GUI.

SmartTime [Top *] - [Constrai	nts Editor]
🕎 File Edit View Actions Tools '	Window Help
	○ >≤ @ > 2 >>
Constraints	Syntax Clock Name Clock Source Period (ns) Frequency (MHz) Dutycycle (%) First Edge Offset (ns) Waveform File Click here to add a constraint Click here to add a constraint
Generated Clock 	1 🚩 CLK CLK 5.000 200.000 50.000 irising 0.000 0 2.5 GUI
create_clock -name{	CLK}-period 5.000-waveform{0.000 2.500}{CLK}

Figure 29 • Clock Constraint Using Constraints Editor and SDC

2. Identify a through pin for Multicycle and apply a multicycle constraint. Refer to "Appendix D: Applying a Multicycle Clock Constraint" on page 41 for creating a multicycle clock constraint using the GUI.



Figure 30 • Multicycle Path Constraints in Constraint Editor

SmartTime timing analysis view uses the above multicycle constraint for a timing check. Figure 31 shows how the multicycle constraint is used in a setup check.

Pin Name	Туре	Net Name	Cell Name	Op	Delay (ns)	Total (ns)	Fanout Edge
data arrival time						7.304	
slack						4.194	
Data arrival time calculat	ion	1					
					0.000	0.000	
	Clock source			+	0.000	0.000	1
CLK_pad/U0/U0:PAD	net	СГК		+	0.000	0.000	r
	cell	02.1	ADI IB·IOPAD IN	+	1.016	1 016	1 r
CLK_pad/U0/U1:A	net	CLK_pad/U0/NET1		+	0.000	1 016	r .
	cell		ADLIB:CLKIO	+	0.348	1.364	3 r
DFN1 0:CLK	net	CLK c		+	0.678	2.042	1
DFN1 0:0	cell		ADLIB:DFN1	+	0.737	2.779	2 f
AND2 0:A	net	DFN1 0 Q		+	0.308	3.087	f
AND2 0:Y	cell		ADLIB:AND2	+	0.386	3.473	1 f
BUFD 0:A	net	AND2 0 Y		+	0.308	3.781	f
BUFD 0:Y	cell		ADLIB:BUFF	+	0.537	4.318	1 f
BUFD 1:A	net	BUFD 0 Y		+	0.308	4.626	f
BUFD 1:Y	cell		ADLIB:BUFF	+	0.537	5.163	1 f
BUFD_2:A	net	BUFD_1_Y		+	0.308	5.471	f
BUFD 2:Y	cell		ADLIB:BUFF	+	0.537	6.008	1 f
MX2 0:B	net	BUFD 2 Y		+	0.323	6.331	f
MX2_0:Y	cell		ADLIB:0R2A	+	0.650	6.981	1 f
DFN1_1:D	net	MX2_0_Y	İ	+	0.323	7.304	f
data arrival time						7.304	
Data required time calcul	ation	,					
CLK	Multicyle Constraint				10.000	10.000	
CLK	Clock source			+	0.000	10.000	r
CLK_pad/U0/U0:PAD	net	CLK		+	0.000	10.000	r
CLK_pad/U0/U0:Y	cell		ADLIB:IOPAD_IN	+	1.016	11.016	1 r
CLK_pad/U0/U1:A	net	CLK_pad/U0/NET1		+	0.000	11.016	r
CLK_pad/U0/U1:Y	cell		ADLIB:CLKIO	+	0.348	11.364	3 г
DFN1_1:CLK	net	CLK_c		+	0.673	12.037	r
DFN1_1:D	Library setup time		ADLIB:DFN1		0.539	11.498	
data required time						11.498	

Figure 31 • Setup for Multicycle Path

Analyzing a Multicycle Path with Inter-Clock Domain

The analysis of a multicycle path in a cross-clock domain is complex. If the captured clock is generated from the launched (source) clock and also runs slower than the launched clock, then moving the launched clock one cycle forward is not equal to moving the end clock one cycle backward. The different options give totally different timing windows. So you need to be careful when applying multicycle setup and multicycle hold for this condition.

A design example with a reference clock (CLK) and generated clock (DFN1_3:CLK) is shown in Figure 32 on page 22. Assume that the path through the AND gate is a multicycle path and the designer wants to apply a multicycle constraint for this path.





Figure 32 • Design for Multicycle Path with Inter-Clock Domain



The setup and hold analysis under various conditions are shown in Figure 33. Due to the offset between CLK and DFN1_3:Q, SmartTime by default uses setup check 1 (SC1) for the setup check. However, you should use setup check 2 (SC2) for setup checks. For hold check, SmartTime uses hold check 2 (HC2) by default. If you use the wrong edge, hold check 1 (HC1), you may see a timing violation.



Figure 33 • Launched and Captured Edges During Multicycle Analysis

Analyzing a Multicycle Path on a Generated Clock with SmartTime

1. Specify the reference clock frequency and other attributes. Refer to "Appendix A: Applying a Clock Constraint" on page 35 for creating a generated clock constraint using the GUI.

SmartTime [Top *] - [Constraints Editor]											
💱 File Edit View Actions Tool:	🖵 File Edit View Actions Tools Window Help										
	വല		<u>s 🗠 x</u>	8	2	m 🐜 💈	* × ×	8	5 M	<u>r</u> 2	<u>n</u>
E Constraints		Syntax	Clock Name	Clock Source	Period (ns)	Frequency (MHz)	Dutycycle (%)	First Edge	Offset (ns)	Waveform	File
		Click her	e to add a const	raint							
Generated Clock	1	٣	CLK	CLK	5.714	175.000	50.000	rising	0.000	0 2.85714	GUI
Input Delay Output Delay											
create_clock -name	e{CL	K}-pe	eriod 5	.714-wav	efor	m{0.00	0 2.85	57}{	CLK}		

Figure 34 • Clock Constraint Using Constraints Editor and SDC



2. Identify the generated clock constraint and apply the generated clock constraint. Refer to "Appendix B: Applying a Generated Clock Constraint" on page 37 for creating a generated clock constraint using the GUI.

SmartTime [Top] - [Constrain [] File Edit View Actions Tools	nts Ed Windo	itor] w Help							
o s he rx :	2		<u>s @</u> z	8	📝 n 🦗 📴	م 🔊 🖾	< 🏷 🧌	<u>n 19 19 19</u>	r 🖓
E Constraints		Syntax	Clock Name	Clock Pin	Reference Pin	Multiplier	Divider	Waveform	File
- Beguirements		Click here	e to add a const	raint					
Clock	1	٣	DFN1_3:Q	DFN1_3:Q	DFN1_3:CLK	1	2	synchronized	GUI
Generated Clock									
Input Delay									
Output Delay									
Eucoptions									
create_generated_c.	loci	k −na	ume { DFN1	3:Q}-a	divide_by2 -s	source	{DFN1	1_3:CLK}	
{DFN1 3:0}									
(21111_0,0)									

Figure 35 • Generated Clock Constraint

- 3. Enable inter-clock domain analysis. Refer to "Appendix C: Enabling Inter-Clock Domains Analysis" on page 40.
- 4. Identify a through pin for multicycle and apply a multicycle constraint. Refer to "Appendix D: Applying a Multicycle Clock Constraint" on page 41 for creating a generated clock constraint using the GUI. Ensure that the **Setup Check only** option is selected, since multicycle constraint is applied to setup check only in this design example. For hold check, the default edge is used.

🕑 SmartTime [Top *] - [Constrai	ints I	Editor]							
😳 File Edit View Actions Tools '	🖓 File Edit View Actions Tools Window Help								
8 8 1X 12 x 0 x 2 y ***** ** *** **									
E Constraints		Syntax	From	Through	То	Setup Cycles	Hold Cycles	Register Sensitivity	File
Clock	Click here to add a constraint								
Generated Clock	1	٣		[get_pins { MX2_0:Y }]		2		source register	GUI
Input Delay									
Output Delay									
Exceptions									
Max Delay									
Min Delay									
False Path									
set_multicycle_path	i-s	etup2	2-through[ge	et_pins{MX2_	_0:Y}]				

Figure 36 • Multicycle Path Constraints in Constraint Editor



SmartTime timing analysis view uses this multicycle path constraint for setup checks. Figure 37 shows a setup check and a hold check.

Pin Name	Туре	Net Name	Cell Name	Op	Delay (ns)	Total (ns)	Fanout
data required time	1		1	-		22 /10	
data arrival time						11 894	
slack		-				11.524	
					1		
Data arrival time calculatio	n				0.000	0.000	
	Clock source				0.000	0.000	
	net	CI K		+	0.000	0.000	
	cell	CER		+	1 560	1.560	1
CLK pad/U0/U1:A	net	CLK pad/U0/NET	1	+	0.000	1.560	
CLK_pad/U0/U1:Y	cell		ADLIB:CLKIO	+	1.016	2.576	3
DFN1_1:CLK	net	CLK_c		+	0.784	3.360	
DFN1_1:Q	cell		ADLIB:DFN1	+	1.399	4.759	1
AND2_0:B	net	DFN1_1_Q		+	0.931	5.690	
AND2_0:Y	cell		ADLIB:AND2	+	1.517	7.207	1
3UFD_0:A	net	AND2_0_Y		+	1.797	9.004	
3UFU_U:Y	cell		ADLIB:BUFF	+	1.177	10.181	1
	net	BOFD_0_Y	ADUB/HV2	+	0.286	10.467	
MAZ_0.1	Cell net	MX2.0.V	AULID.MAZ	+	0.000	11.608	
data arrival time	nou	MA2_0_1		-	0.205	11.034	
						11.004	
Data required time calculat	tion						
DFN1_3:Q	Multicyle Constrai	nt			17.142	17.142	
DFN1_3:Q	Clock source			+	0.000	17.142	
	Clock generation			+	4.300	21.442	
DEN1_3_RNINE25:A	net	DFN1_3_Q_i		+	0.365	21.807	
JEN1_3_RNINE25:Y	cell		ADLIB:CLKINT	+	1.982	23.789	3
JENT_2:ULK	net	DENT_3_Q		+	0.974	24.763	
JFN1_2.D	Library setup time		ADLIB:DENT	-	1.343	23.410	
)etails for path rom: DFN1_0:CLK							
)etails for path 'rom: DFN1_0:CLK 'o: DFN1_2:D	Turne	Mat Nama	C-1111	0-1		T-1-1 ()	F 1
Details for path 'rom: DFN1_0:CLK 'o: DFN1_2:D Pin Name	Туре	Net Name	Cell Name	Op	Delay (ns)	Total (ns)	Fanout
Details for path Trom: DFN1_0:CLK <u>o: DFN1_2:D</u> Pin Name	Туре	Net Name	Cell Name	Op	Delay (ns)	Total (ns)	Fanout
Details for path rom: DFN1_0:CLK o: DFN1_2:D Pin Name ata arrival time	ј Туре	Net Name	Cell Name	Op	Delay (ns))	Total (ns)) 7.103	Fanout
Details for path rom: DFN1_0:CLK o: DFN1_2:D Pin Name ata arrival time ata required time	Туре	Net Name	Cell Name	<u>Op</u>	Delay (ns)j	Total (ns) 7.103 5.076	Fanout
Details for path rom: DFN1_0:CLK o: DFN1_2:D Pin Name lata arrival time ata required time lack	Туре	Net Name	Cell Name	<u>Op</u>	Delay (ns)	Total (ns) 7.103 5.076 2.027	Fanout
Details for path rom: DFN1_0:CLK o: DFN1_2:D Pin Name lata arrival time lata required time lack bata arrival time calculatio	Type	Net Name	Cell Name	O p -	Delay (ns)	Total (ns) 7.103 5.076 2.027	Fanout
Details for path rom: DFN1_0:CLK o: DFN1_2:D Pin Name lata arrival time lata required time lack Pata arrival time calculatio LK	Type	Net Name	Cell Name	<u>Ор</u> -	Delay (ns)	Total (ns) 7.103 5.076 2.027 0.000	Fanout
Details for path rom: DFN1_0:CLK o: DFN1_2:D Pin Name lata arrival time lata required time lack Pata arrival time calculatio LK LK	n Clock source	Net Name	Cell Name	Ор - +	Delay (ns)	Total (ns) 7.103 5.076 2.027 0.000 0.000	Fanout
Details for path rom: DFN1_0:CLK ro DFN1_2:D Pin Name lata arrival time lata required time lata required time lack Data arrival time calculatio LK LK LK LK LK_pad/U0/U0:PAD	n Clock source net	Net Name	Cell Name	Op - - + +	Delay (ns)	Total (ns) 7.103 5.076 2.027 0.000 0.000 0.000	Fanout
Details for path rom: DFN1_0:CLK ro: DFN1_2:D Pin Name lata arrival time lata required time lack Data arrival time calculatio LK LK_pad/U0/U0:PAD LK_pad/U0/U0:Y	n Clock source net cell	Net Name	Cell Name	Op 	Delay (ns)	Total (ns) 7.103 5.076 2.027 0.000 0.000 0.000 1.039 1.039	Fanout
Details for path rom: DFN1_0:CLK o: DFN1_2:D Pin Name lata arrival time lata required time lack Data arrival time calculatio LK LK LK_pad/U0/U0:PAD LK_pad/U0/U0:Y LK_pad/U0/U1:A 1K_C_PADUB/EV DATA	n Clock source net cell net	Net Name	Cell Name	Op - - + + + + +	Delay (ns)	Total (ns) 7.103 5.076 2.027 0.000 0.000 0.000 1.039 1.039	Fanout
Details for path rom: DFN1_0:CLK o: DFN1_2:D Pin Name lata arrival time lata required time lack Data arrival time calculatio LK LK_pad/U0/U0:PAD LK_pad/U0/U0:Y LK_pad/U0/U1:A LK_pad/U0/U1:Y UK_pad/U0/U1:Y	n Clock source net cell net cell	Net Name CLK CLK_pad/U0/NET1	Cell Name	Op - - + + + + + + + + + +	Delay [ns]	Total (ns) 7.103 5.076 2.027 0.000 0.000 0.000 1.039 1.039 1.715 2.234	Fanout
Details for path rom: DFN1_0:CLK o: DFN1_2:D Pin Name lata arrival time lata required time lack Data arrival time calculatio LK DAta arrival time calculatio LK LK_pad/U0/U0:PAD LK_pad/U0/U0:Y LK_pad/U0/U1:A LK_pad/U0/U1:Y IK_pad/U1:X IK_pad/U0/U1:Y IK_pad/U0/U1:Y IK_p	n Clock source net cell net cell net cell	Net Name	Cell Name	Op 	Delay (ns) 0.000 0.000 0.000 1.039 0.000 0.676 0.556 0.562	Total (ns) 7.103 5.076 2.027 0.000 0.000 1.039 1.039 1.715 2.271 2.260	Fanout 1
Details for path rom: DFN1_0:CLK ro: DFN1_2:D Pin Name lata arrival time lata required time lata required time lack Data arrival time calculatio LK LK LK LK_pad/U0/U0:PAD LK_pad/U0/U0:Y LK_pad/U0/U1:A LK_pad/U0/U1:A LK_pad/U0/U1:Y iFN1_0:CLK iFN1_0:Q ND2 0:4	n Clock source net cell net cell net cell	Net Name	Cell Name	Op 	0.000 0.000 0.000 0.000 1.039 0.000 0.676 0.556 0.556 0.557 0.245	Total (ns) 7.103 5.076 2.027 0.000 0.000 1.039 1.039 1.715 2.271 2.2688 3.112	Fanout
Details for path rom: DFN1_0:CLK ro DFN1_2:D Pin Name lata arrival time lata required time lack Data arrival time calculatio LK LK_pad/U0/U0:PAD LK_pad/U0/U0:PAD LK_pad/U0/U1:Y LK_pad/U0/U1:Y IK_PAD/U1:Y IK_PAD/U1:Y IK_PAD/U1:Y IK_PAD/U1:Y IK_PAD/U1:Y IK_PAD/U1:Y IK_PAD/U1:Y IK_PAD/U1:Y IK_PAD/U1:Y IK_PAD/U1:Y IK_PAD/U1:Y IK_PAD/U1:Y IK_PAD/U1:Y IK_PAD/U1:Y IK_PAD/U1:Y IK_PAD/U1:Y IK_PAD/U1:Y IK_X	n Clock source net cell net cell net cell net cell net	Net Name	Cell Name	Op ·	Delay (ns) 0.000 0.000 0.000 1.033 0.000 0.676 0.556 0.557 0.245 0.504	Total (ns) 7.103 5.076 2.027 0.000 0.000 1.039 1.039 1.039 1.715 2.271 2.868 3.113 3.617	Fanout 1 3 1
Details for path rom: DFN1_0:CLK ro: DFN1_2:D Pin Name lata arrival time lata required time lack Data arrival time calculatio LK LK_pad/U0/U0:PAD LK_pad/U0/U0:PAD LK_pad/U0/U1:A LK_pad/U0/U1:A LK_pad/U0/U1:Y FN1_0:CLK FN1_0:CLK FN1_0:Q ND2_0:A ND2_0:A UFD_0:A	n Clock source net cell	Net Name	Cell Name	0p 	Delay (ns) 0.000 0.000 0.000 1.039 0.000 0.676 0.556 0.556 0.597 0.245 0.504 1.416	Total (ns) 7.103 5.076 2.027 0.000 0.000 0.000 1.039 1.039 1.039 1.039 1.039 1.039 1.039 1.031 1.715 2.271 2.868 3.113 3.617 5.033	Fanout 1 3 1
Details for path from: DFN1_0:CLK for: DFN1_2:D Pin Name lata arrival time lata required time lack Jata arrival time calculatio LK LK_pad/U0/U0:PAD LK_pad/U0/U1:A LK_pad/U0/U1:A LK_pad/U0/U1:Y FN1_0:CLK FN1_0:Q ND2_0:A ND2_0:A UFD_0:A UFD 0:Y	n Clock source net cell	Net Name Net Name CLK CLK CLK_pad/U0/NET1 CLK_c DFN1_0_Q AND2_0_Y	Cell Name	Op -	Delay [ns]	Total (ns) 7.103 5.076 2.027 2.027 0.000 0.000 0.000 1.039 1.039 1.715 2.271 2.868 3.113 3.617 5.033 5.638	Fanout 1 3 1 1 1
Details for path rom: DFN1_0: CLK rom: DFN1_2:D Pin Name lata arrival time lata required time lata arrival time calculatio Jata arrival time calculatio LK UK_pad/U0/U0:PAD LK_pad/U0/U0:Y LK_pad/U0/U1:A LK_pad/U0/U1:A LK_pad/U0/U1:Y IFN1_0:CLK ND2_0:A ND2_0:Y UFD_0:A UFD_0:Y K2_0:A	n Clock source net cell net cell net cell net cell net cell net cell net cell net	Net Name	Cell Name	Op - - + + + + + + + + + + + + + + + + +	Delay [ns] 0.000 0.000 0.000 0.000 0.000 0.676 0.556 0.556 0.557 0.245 0.504 1.416 0.605 0.605 0.267	Total (ns) 7.103 5.076 2.027 0.000 0.000 0.000 1.039 1.039 1.715 2.271 2.868 3.113 3.617 5.033 5.638 5.905	Fanout 1 3 1 1 1
Details for path rom: DFN1_0:CLK rom: DFN1_2:D Pin Name lata arrival time lata required time lata rrival time calculatio lata trival time lata required time lata trival time calculatio LK LK_pad/U0/U0:PAD LK_pad/U0/U1:A LK_pad/U0/U1:A LK_pad/U0/U1:Y IFN1_0:CLK IFN1_0:CA ND2_0:Y UFD_0:A UFD_0:A UFD_0:Y X2_0:Y	nt Clock source net cell net cell net cell net cell net cell net cell net cell net cell net cell net cell	Net Name Net Name CLK CLK_pad/U0/NET1 CLK_c DFN1_0_Q AND2_0_Y BUFD_0_Y	Cell Name	Op - - - - - - - - - +	Delay (ns) 0.000 0.000 0.000 0.000 0.000 0.676 0.556 0.556 0.557 0.245 0.504 1.416 0.605 0.267 0.267 0.233	Total (ns) 7.103 5.076 2.027 0.000 0.000 1.039 1.039 1.715 2.271 2.868 3.113 3.617 5.033 5.638 5.905 6.838	Fanout 1 3 1 1 1 1 1
Details for path rom: DFN1_0:CLK rom: DFN1_0:CLK Pin Name Iata arrival time Iata required time Iata required time Iack Data arrival time calculatio LK LK_pad/U0/U0:PAD LK_pad/U0/U0:PAD LK_pad/U0/U0:Y LK_pad/U0/U1:Y VFN1_0:CLK VFN1_0:Q ND2_0:A UFD_0:A UFD_0:Y VX2_0:A VX2_0:Y	net cell net cell net cell net cell net cell net cell net cell net cell net	Net Name	Cell Name	Op -	Delay (ns) 0.000 0.000 0.000 0.000 0.000 0.000 0.676 0.556 0.556 0.557 0.245 0.504 1.416 0.605 0.265 0.933 0.285	Total (ns) 7.103 5.076 2.027 0.000 0.000 1.039 1.715 2.271 2.868 3.113 3.617 5.033 5.638 5.905 6.838 8.7103	Fanout 1 3 1 1 1 1 1
Details for path rom: DFN1_0:CLK rom: DFN1_2:D Pin Name Iata arrival time Iata required time Iack Data arrival time calculatio LK LK_pad/U0/U0:PAD LK_pad/U0/U0:Y LK_pad/U0/U1:Y JFN1_0:Q ND2_0:A UFD_0:A UFD_0:Y UFD_0:Y IVE_DO:A UFD_0:Y VX2_0:Y UFD_0:D VX2_0:Y UFD_0:D IX	n Clock source net cell cell net cell net cell net cell cell net cell net cell cell cell net cell net cell cell cell cell cell cell cell ce	Net Name	Cell Name	Op -	Delay (ns) 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.676 0.556 0.556 0.557 0.245 0.554 1.416 0.605 0.267 0.933 0.285 0.285	Total (ns) 7.103 5.076 2.027 0.000 0.000 0.000 1.039 1.715 2.271 2.868 3.113 3.161 3.163 5.638 5.905 6.838 5.905 6.838 7.103 7.103 7.103	Fanout 1 3 1 1 1 1 1
Details for path rom: DFN1_0:CLK rom: DFN1_2:D Pin Name Iata arrival time Iata arrival time Iata required time Iack Data arrival time calculatio LK LK_pad/U0/U0:PAD LK_pad/U0/U0:Y LK_pad/U0/U1:Y VFN1_0:CLK VFN1_0:CLK VFN1_0:Q ND2_0:A UFD_0:A UFD_0:Y VK2_0:Y UFD_0:A UFD_0:Y VX2_0:Y UFD_0:A UFD_0:Y VX2_0:Y	n Clock source net cell net ce	Net Name	Cell Name	Op - - - - +	Delay (ns) 0.000 0.000 0.000 0.000 0.000 0.000 0.676 0.556 0.556 0.557 0.245 0.504 1.416 0.605 0.267 0.933 0.285 0.285	Total (ns) 7.103 5.076 2.027 0.000 0.000 1.039 1.715 2.271 2.868 3.113 3.161 5.033 5.638 5.905 6.838 5.905 6.838 5.7103 7.103	Fanout 1 3 1 1 1 1 1
Details for path rom: DFN1_0:CLK rom: DFN1_0:CLK Pin Name Iata arrival time Iata required time Iata required time Iack Data arrival time calculatio LK LK_pad/U0/U0:PAD LK_pad/U0/U1:PAD LK_pad/U0/U1:PAD LK_pad/U0/U1:PAD LK_pad/U0/U1:PAD UFL_pad/U0/U1:PAD UK_D_DCLK IFN1_0:Q ND2_0:P UIFD_0:A UFD_0:Y K2_0:A K2_0:Y IFN1_3:Q ata arrival time	n Clock source net cell cell net cell cell cell cell cell cell cell ce	Net Name	Cell Name	Dp	Delay (ns) 0.000 0.000 0.000 1.033 0.000 0.676 0.556 0.557 0.245 0.554 1.416 0.605 0.267 0.333 0.265 0.287 0.333 0.265	Total (ns) 7.103 5.076 2.027 0.000 0.000 1.039 1.039 1.715 2.271 2.868 3.113 3.617 5.033 5.638 5.905 6.838 5.905 6.838 7.103 7.103 7.103	Fanout 1 3 1 1 1 1 1
Details for path rom: DFN1_0:CLK for DFN1_2:D Pin Name Iata arrival time Iata required time Iata arrival time calculatio Iata arrival time calculatio LK LK_pad/U0/U0:PAD LK_pad/U0/U0:PAD LK_pad/U0/U1:A LK_pad/U0/U1:A UK_pad/U0/U1:A UK_pad/U0/U1:Y FN1_0:CLK IFN1_0:Q VID2_0:A UFD_0:Y UFD_0:A UFD_0:A UFD_0:A UFD_0:A VX2_0:A X2_0:A X2_0:Y FN1_3:Q Pata arrival time Pata arrival time		Net Name Net Name CLK CLK_pad/U0/NET1 CLK_c DFN1_0_Q AND2_0_Y BUFD_0_Y MX2_0_Y	Cell Name	Op - - - + - - - + + + + + + +	Delay [ns]	Total [ns] 7.103 5.076 2.027 0.000 0.000 0.000 1.039 1.039 1.039 1.039 1.039 1.715 2.271 2.868 3.113 3.617 5.033 5.638 5.905 6.838 5.905 6.838 7.103 7.103 7.103 7.103	Fanout 1 3 1 1 1 1
Details for path rom: DFN1_0:CLK o: DFN1_2:D Pin Name lata arrival time lata required time lack Data arrival time calculatio LK LK_pad/U0/U0:PAD LK_pad/U0/U0:PAD LK_pad/U0/U1:A LK_PAD/U0/U1:A LK_PAD/U0/U1:A LK_PAD/U0/U1:A L		Net Name CLK CLK_pad/U0/NET1 CLK_c DFN1_0_Q AND2_0_Y BUFD_0_Y MX2_0_Y	Cell Name	Op -	Delay [ns] 0.000 0.000 0.000 0.000 0.076 0.556 0.597 0.245 0.504 1.416 0.605 0.267 0.933 0.265 0.265 0.267 0.933 0.265 0.200 0.265 0.200 0.200 0.200 0.200 0.200 0.200 0.200 0.200 0.200 0.000 0.000 0.000 0.000 0.000 0.000 0.268 0.265 0.26	Total (ns) 7.103 5.076 2.027 0.000 0.000 0.000 1.039 1.03	Fanout 1 3 1 1 1 1
Details for path from: DFN1_0: CLK for: DFN1_2:D Pin Name Iata arrival time Iata required time Iata arrival time calculatio Jata arrival time calculatio Jata arrival time calculatio JK UK_pad/U0/U0:PAD LK_pad/U0/U1:A LK_pad/U0/U1:A LK_pad/U0/U1:A UK_pad/U0/U1:A UFD_0:A UFD_0:A UFD_0:A UFD_0:A UFD_0:A		Net Name	Cell Name		Delay (ns) 0.000 0.000 0.000 0.000 1.039 0.000 0.676 0.556 0.557 0.245 0.504 1.416 0.605 0.267 0.245 0.504 1.416 0.605 0.267 0.933 0.265 0.265 0.000 0.000 0.2868 0.269	Total (ns) 7.103 5.076 2.027 0.000 0.000 1.039 1.039 1.715 2.271 2.868 3.113 3.617 5.033 5.638 5.5905 6.838 7.103 7.103 7.103 0.0000 0.000 0.0000 0.000 0.000 0.000 0.000 0.000 0	Fanout 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Details for path rom: DFN1_0:CLK rom: DFN1_2:D Pin Name Iata arrival time lata required time lata required time lx LK LK_pad/U0/U0:PAD LK_pad/U0/U1:A LK_pad/U0/U1:A LK_pad/U0/U1:A LK_pad/U0/U1:A UFD_0CA WD2_0:A WD2_0:A WD2_0:A WED_0:Y VFN1_2:D lata arrival time Pata required time calculat VFN1_3:Q VFN1_3:RNINE25:A VFN1_3_RNINE25:Y		Net Name	Cell Name Cell Name Cell Name ADUB:IOPAD_IN ADUB:IOPAD_IN ADUB:CLKIO ADUB:AND2 ADUB:BUFF ADUB:BUFF ADUB:BWX2 ADUB:CLKINT		Delay (rs) 0.000 0.000 0.000 0.000 0.000 0.000 0.676 0.556 0.556 0.557 0.504 1.416 0.605 0.265 0.259 0.25	Total [ns] 7.103 5.076 2.027 0.000 0.000 1.039 1.039 1.715 2.271 2.868 3.113 3.617 5.033 5.638 5.905 6.838 7.103 7.103 7.103 0.000 0.000 0.000 0.000 2.868 3.127 4.386	Fanout
Details for path rom: DFN1_0:CLK rom: DFN1_0:CLK rom: DFN1_2:D Pin Name Iata arrival time Iata required time Iack Data arrival time calculatio LK LK LK_pad/U0/U0:PAD LK_pad/U0/U0:PAD LK_pad/U0/U0:PAD LK_pad/U0/U1:Y VFN1_0:CLK VFN1_0:Q ND2_0:A VUFD_0:A VUFD_0:A VUFD_0:Y iuFD_0:Y iuFN1_3:Q iFN1_3:RNINE25:A iFN1_2:CLK	net Cell net cell cell net cell cell net	Net Name	Cell Name Cell N	Op - - - - - +	Delay (rs) 0.000 0.000 0.000 0.000 0.000 0.000 0.676 0.556 0.557 0.554 1.416 0.605 0.265 0.25	Total (ns) 7.103 5.076 2.027 0.000 0.000 1.039 1.039 1.715 2.271 2.868 3.113 3.617 5.033 5.638 5.905 6.838 7.103 7.10	Fanout 1 3 1 1 1 1 1 3 3 3

Figure 37 • Setup and Hold Checks for Multicycle Path



Figure 38 shows that if you use the wrong value for hold, it displays the incorrect hold violation.

To: DFN1_2:D							
Pin Name	Туре	Net Name	Cell Name	Op	Delay (ns)	Total (ns)	Fanout
data arrival time						7.103	
data required time				-		10.790	
slack						-3.687	
Data arrival time calculat	tion						
CLK					0.000	0.000	
CLK	Clock source			+	0.000	0.000	
CLK_pad/U0/U0:PAD	net	CLK		+	0.000	0.000	
CLK_pad/U0/U0:Y	cell		ADLIB:IOPAD_IN	+	1.039	1.039	1
CLK_pad/U0/U1:A	net	CLK_pad/U0/NET1		+	0.000	1.039	
CLK_pad/U0/U1:Y	cell		ADLIB:CLKIO	+	0.676	1.715	3
DFN1_0:CLK	net	CLK_c		+	0.556	2.271	
DFN1_0:Q	cell		ADLIB:DFN1	+	0.597	2.868	1
AND2_0:A	net	DFN1_0_Q		+	0.245	3.113	
AND2_0:Y	cell		ADLIB:AND2	+	0.504	3.617	1
BUFD_0:A	net	AND2_0_Y		+	1.416	5.033	
BUFD_0:Y	cell		ADLIB:BUFF	+	0.605	5.638	1
MX2_0:A	net	BUFD_0_Y		+	0.267	5.905	
MX2_0:Y	cell		ADLIB:MX2	+	0.933	6.838	1
DFN1_2:D	net	MX2_0_Y		+	0.265	7.103	
data arrival time						7.103	
Data required time calcul	lation						
DFN1_3:Q	Multicyle Constraint	t			5.714	5.714	
DFN1_3:Q	Clock source			+	0.000	5.714	
	Clock generation			+	2.868	8.582	
DFN1_3_RNINE25:A	net	DFN1_3_Q_i		+	0.259	8.841	
DFN1_3_RNINE25:Y	cell		ADLIB:CLKINT	+	1.259	10.100	3
DFN1_2:CLK	net	DFN1_3_Q		+	0.690	10.790	
DEN1 2D	Library hold time		ADLIB:DEN1	+	0.000	10 790	

Figure 38 • Wrong Clock Edge Used in Hold Check

Analyzing Clock Gating

The gated clock signal occurs when a clock path contains logic other than inverters or buffers. The default setting in SmartTime timing analysis view enables setup and hold analysis for the reference clock. However, it does not do timing analysis on the gating cells between the gating signal and clock. It is possible for the gated signal to have transitions while clock pulses are passing through the gating cells and this can lead to both clipped and spurious clock pulses. This section provides detailed information on doing this timing analysis manually.

Figure 39 on page 27 shows the most generalized circuit for the gated clock and the timing waveform. The GATED_CLK signal propagates through AND2_0 to the downstream flip-flops only when CLK_EN is high. In order to be glitch free, the output from DFN0_0 should arrive at input B of AND2_0 after the falling edge of CLK arrives at input A of AND2_0 and before the next rising CLK arrives at input A of AND2_0. The setup check analysis should use the following timing calculation:

- Launched edge: The data path starts at CLK, goes through DFN0_0 (D->Q), and then ends at the AND2_0:B pin. Both rising edge and falling edge timing must be calculated and the larger result will be used.
- Captured edge: The clock path starts at CLK and ends at the AND2_0:A pin. The timing must be checked for rising edge only.



The hold check analysis should use the following timing:

- Launched edge: The data path starts at CLK, goes through DFN0_0 (D->Q), and ends at the AND2_0:B pin. The timing must be for the falling edge of the clock to the DFN1_0 FF and then for either rising or falling edge to the AND2_0 gate, whichever is shorter.
- Captured edge: The clock path starts at CLK and ends at the AND2_0:A pin. The timing must be for delays of the negative edge of the clock.



Figure 39 • Gated Clock Example





Figure 40 • Timing Waveform



Analyzing a Gated Clock

1. Specify the reference clock frequency and other attributes. Refer to "Appendix A: Applying a Clock Constraint" on page 35 for creating a generated clock constraint using the GUI.

SmartTime [Top *] - [Consti	aints Edito	r]							
😳 File Edit View Actions Tools	Window H	elp							
	ael B	<u>× @</u> x	2	32	n ha	× 😽 🖄	* *	<u>m</u>	\$
E Constraints	Syn	ax Clock Name	Clock Source	Period (ns)	Frequency (MHz)	Dutycycle (%)	First 0 Edge	ffset (ns) Wav	eform
Generated Clock	1	CLK	CLK	10.000	100.000	50.000	rising (0 5 0.000	
Output Delay									
create clock -name	{CLK}-	period 10).000-wa	vefo	rm{0.0	00 5.0)00}{	CLK}	

Figure 41 • Clock Constraint Using Constraints Editor and SDC

2. Identify AND2_0:A and AND2_0:B as generated clocks and apply the generated clock constraint. Refer to "Appendix B: Applying a Generated Clock Constraint" on page 37 for creating a generated clock constraint using the GUI.

SmartTime [Top] - [Constrai] File Edit View Actions Tools	nts Editor] Window Helj	D						
	e 🔉	<u> </u>	8	n 🐜 📴 🍋	× & %	(<u></u>	2 🗐 🔊	3
⊡ Constraints	Synta	x Clock Name	Clock Pin	Reference Pin	Multiplier	Divider	Waveform	File
🗄 Requirements	Click he	ere to add a const	raint					
🕈 Clock	1	AND2_0:A	AND2_0:A	CLK	1	1	synchronized	GUI
🔨 Generated Clock	2 🚩	AND2_0:B	AND2_0:B	DFN0_0:CLK	1	1	inverted	GUI
Output Delay			'					
create generated c	lock -n	ame{AND2	$2 \ 0:A \} - c$	livide byl -s	ource	{CLK	}{AND2 0):A}
<pre>create_generated_c source{DFN0_0:CLK}</pre>	lock -n {AND2_0	ame{AND2 :B}	2_0:B}-c	livide_by1 -i	nvert	-	_	

Figure 42 • Generated Clock Constraint

3. Enable inter-clock domain analysis. Refer to "Appendix C: Enabling Inter-Clock Domains Analysis" on page 40.



SmartTime maximum delay analysis view shows the reference clock and AND2_0:A to AND2_0:B clock domain. You need to get the delay values from the expanded data path and do setup and hold calculation on the gated cell. Figure 43 shows one of the expanded paths and the setup and hold calculation on the gated cell.

⊡~ 🖏 Summary									
🔄 🖓 Datasheet				Dalan		•	Demuired	L Cabur	1
🖃 🖌 🚾 CLK		Source Pin	Sink Pin	(ns)		(ns)	(ns)	fins)	
Register to Register	1	DFN1 1:CLK	DFN1 2:D	1.30	9 2.936	4.834	7,770	0.71	a
External Setup			_						-
Clock to Output		Details for path							
Register to Asynchron		From: DFN1_1:CLK							
External Recovery		IO: UFNI_2:U	Turn	Net News				hal (= a) [an and Edge
Asynchronous to Regi			Туре				relay (nsj_ i o	tai (nsjj r	anoug Euge
AND2_0:A to CLK									
AND2_0:B to CLK		data required time		1	1		1	7 770	
⊡ 🖌 💮 AND2_0:A		data arrival time			1			4 834	
Register to Register		slack						2.936	
External Setup									
Clock to Output		Data arrival time calcula	tion						
Register to Asynchron		AND2_0:B					0.000	0.000	
External Recovery		AND2_0:B	Clock source			+	0.000	0.000	T
Asynchronous to Regi			Clock generation	1		+	2.600	2.600	
CLK to AND2_0:A		AND2_0:Y	cell		ADLIB:AND2	+	0.591	3.191	2 r
AND2_0:B to AND2_		DFN1_1:CLK	net	AND2_0_Y		+	0.334	3.525	1
🖻 🗹 🐨 AND2_0:B		DFN1_1:Q	cell		ADLIB:DFN1	+	0.654	4.179	1 1
Register to Register		DFN1_2:D	net	DEN1_1_Q		+	0.655	4.834	1
External Setup		data arrival time						4.834	
Clock to Output		Data and in different sector	1-1 ⁻						
Register to Asynchron			Clock Constraint				5.000	5 000	
External Recovery		AND2_0.A	Clock constraint	-		+	0.000	5.000	
Asynchronous to Regi			Clock generation			+	2.032	7.032	
CLK to AND2_0:B		AND2 0:Y	cell		ADLIB:AND2	+	0.488	7.520	2 r
AND2_0:A to AND2_		DFN1 2:CLK	net	AND2 0 Y		+	0.963	8.483	
😑 🛼 Pin to Pin		DFN1_2:D	Library setup time		ADLIB:DFN1		0.713	7.770	
Input to Output		data required time						7.770	

Figure 43 • Inter-Clock Domain AND2_0:A to AND2_0:B Path

Setup check = Capture edge - launch edge = (10 + 2.032) - (5 + 2.6) = 4.432 ns.

Hold check = Capture edge - launch edge = (5 + 2.6) - (5 + 2.032) = 0.56 ns.

Note: You need to do similar calculations using the delay numbers under the minimum delay analysis view.

Four Corner Analysis

The delay of a path or gate depends on factors such as voltage, temperature, process, and loading. Figure 44 on page 31 shows timing delay under various conditions. In SmartTime, the default maximum delay analysis checks the setup timing under worst case scenario and the minimum delay analysis checks the hold timing under best case scenario. However, for some designs these scenarios do not always cover all the corner case scenarios.





Figure 44 • Timing Delay under Various Conditions

The combination of temperature, voltage and process allow timing variation across various corners. This is why designers want to qualify their design across many conditions. Although multiple corner cases exist where a design can be analyzed, a designer normally uses the following four corners for timing analysis: Min-BEST, Min-WORST, Max-BEST, and Max-WORST. The most extreme timing numbers are found at these corners.

SmartTime performs analysis for Max-WORST and Min-BEST scenarios by default. In general, this is correct for most of the designs. However, if you have very tight slack, the analysis for the other two cases should be done by changing the SmartTime default setting. The "Timing Analysis for Min-WORST or Max-BEST Scenario" section on page 32 shows how to perform analysis for all four corner cases using SmartTime.



Timing Analysis for Min-WORST or Max-BEST Scenario

1. Open the SmartTime Options dialog box (Figure 45) by selecting **Tools** > **Options** from the SmartTime menu bar. You can see that the maximum delay analysis is based on BEST and the minimum delay analysis is based on WORST condition.

SmartTime Options	
Option Categories	General
⊡ Select a category: General Analysis View Advanced	Operating Conditions Perform maximum delay analysis based on Perform minimum delay analysis based on BEST Clock Domains Clock Domains Include inter-clock domains in calculations for timing analysis. Chable recovery and removal checks. Restore Defaults
Help	OK Cancel

Figure 45 • SmartTime Options Dialog Box

2. Under Operating Conditions, change **Perform maximum delay analysis based on** to **BEST** and **Perform minimum delay analysis based on** to **WORST**.

Option Categories	General
Select a category: General Analysis View Advanced	Operating Conditions Perform maximum delay analysis based on Perform minimum delay analysis based on WORST Clock Domains Include inter-clock domains in calculations for timing analysis. Enable recovery and removal checks. Restore Defaults

Figure 46 • SmartTime Options Dialog Box for Max-BEST and Min-WORST Analysis

With this setup, maximum delay analysis view shows setup check under Min-WORST condition and minimum delay analysis view shows hold check for Max-BEST condition.

Consider the design example shown in Figure 47. The clock network has some buffers which add skew on the clock network.



Figure 47 • Design Example for Min-BEST and Min-WORST Analysis



The default minimum delay analysis view shows a slack of -1.32 ns for the register-to-register path. However, changing the minimum delay analysis view to WORST shows slack of -1.877 ns. You can see that the Min-BEST condition does not always have the worst case scenario for hold check.

SmartTime [top *] - [Minimu	ım Delay	Analysis View]							
≲ File Edit View Actions Tools	Window	Help							
		<u>></u>	3 🖸 🕺 📶 🐜 📴	🗞 🔯	×8	<u>m r</u>	2 🚛 🕂	<u>B</u>	
min ⊡- & Summary	From	*					То	*	
⊡ Statesheet ⊡ × @ CLKIN		Source Pin	Sink Pin	Delay (ns)	Slack (ns)	Arrival (ns)	Required (ns)	Hold (ns)	Skew (ns)
External Hold	1 DF1	_1:CLK	DF1_0:D	0.884	-1.321	2.563	3.884	0.000	-2.205

Figure 48 • Register-to-Register Path for Min-BEST Condition



Figure 49 • Register-to-Register Path for Min-WORST Condition

In summary, the SmartTime timing analyzer default setting only checks for Max-WORST and Min-BEST conditions. You need to change the settings to check for Min-WORST or Max-BEST condition if you have tight margin in your design.



Appendix A: Applying a Clock Constraint

 Open the SmartTime constraints editor by clicking the Constraints Editor button in the Designer GUI. The clock constraint is displayed in the SmartTime Constraints Editor, as shown in Figure 50.

Constraints	Syntax Clock Name Clock Source Period (ns) Frequency (MHz) Dutycycle (%) First Edge Offset (ns) Waveform File Commer
🖻 – Requirements	Click here to add a constraint
Clock	
Generated Clock	
Input Delay	
Output Delay	
Exceptions	
Max Delay	
Multicycle	
False Path	
⊡ Advanced	
Clock Source Latency	

Figure 50 • SmartTime Constraints Editor

 Add a clock constraint by clicking the new clock constraint button in the SmartTime toolbar, or by selecting Actions > Constraint > Clock from the SmartTime Menu bar. The Create Clock Constraint dialog box is displayed.



Figure 51 • Create Clock Constraint Dialog Box

 Select the clock source pin from the Clock Sources drop-down list or by clicking the browse button. Select the pin CLK as the clock source. Click OK to close the Clock Source Pin dialog box.



4. Enter 150 as the **Frequency** in the Create Clock Constraint box and accept all other default values. Click **OK** to create the clock constraint.



Figure 52 • Entering a Clock Constraint in the Create Clock Constraint Dialog Box

The clock constraint is visible in the SmartTime Constraints Editor.

Constraints		Syntax	Clock Name	Clock Source	Period (ns)	Frequency (MHz)	Dutycycle (%)	First Edge	Offset (ns)	Waveform	File C
- Requirements		Click here	e to add a const	raint							10.1
* Clock	1	1	CLK	CLK	6.667	150.000	50.000	rising	0.000	0 3.33333	GUI
- Exceptions											

Figure 53 • SmartTime Constraints Editor with Clock Constraint

Appendix B: Applying a Generated Clock Constraint

1. Open the SmartTime constraints editor by clicking the **Constraints Editor** button in the Designer GUI. The clock constraint is visible in the SmartTime Constraints Editor, as shown in Figure 54.

- Constraints	Syntax Clock Name Clock Source Period (ns) Frequency (MHz) Dutycycle (%) First Edge Offset (ns) Waveform File Comme
- Requirements	Click here to add a constraint
Clock	
Generated Clock	
Input Delay	
Output Delay	
- Exceptions	
Max Delay	
Multicycle	
False Path	
- Advanced	
Clock Source Latencu	

Figure 54 • SmartTime Constraints Editor

2. Right-click **Generated Clock** in the Constraints Editor window. The Create Generated Clock Constraint dialog box is displayed.

Create Generated Clock Constraint
Clock Reference:
Generated Clock Name The generated frequency is such as $f(clock) = f(reference) \times 1 \qquad (1)$ Get Pre-Computed Factors
The generated waveform is the same as The reference waveform
Help OK Cancel

Figure 55 • Generated Clock in the Constraint Window



3. Browse to select a **Clock Pin**. The Select Generated Clock Source dialog box displays with the list of available generated clock source pins, as shown in Figure 56.

DFN1_0:Q	
Filter available	objects:
Туре:	Explicit clocks
Filter:	
*	Filter

Figure 56 • Select Generated Clock Source Dialog Box

4. Select the DFN1_0:Q pin and click **OK** to save the clock constraint details. In some cases, the generated clock pins are not defined as Explicit clocks. You need to change the filter type and add the generated clock source pin.



5. Browse to select a Reference Pin. The Select Generated Clock Reference dialog box displays the list of available clock reference pins, as shown in Figure 57.

Select a pin:			
DFN1_0:CLK DFN1_0:Q			
Filter available Type:	objects:	ock network 💌	
Filter:	,	Filter	

Figure 57 • Select Generated Clock Reference Dialog Box

- 6. Select the DFN1_0:CLK pin and click **OK** to save the clock constraint details. Note that DFN1_0:CLK is actually CLKA.
- 7. Enter the division factor of 2, since DFN1_0:Q is a "divided by 2" clock of DFN1_0: CLK.
- 8. Enter the first edge of the generated waveform as "the same as" with respect to the reference waveform.
- 9. Click OK. The new constraint appears in the Constraints List.



Figure 58 • Constraints List



Appendix C: Enabling Inter-Clock Domains Analysis

- 1. Select **Tools > Options** from the SmartTime menu bar.
- 2. Select the Include Inter-clock domains in calculations for timing analysis check box in the SmartTime Options dialog box to select inter-clock domain analysis, as shown in Figure 59. Click **OK**.

Option Categories	General
 Select a category: General Analysis View Advanced 	Operating Conditions Perform maximum delay analysis based on Perform minimum delay analysis based on BEST case Clock Domains
	 Include inter-clock domains in calculations for timing analysis. Enable recovery and removal checks. Restore Defaults

Figure 59 • Enabling Inter-Clock Domain Analysis

Appendix D: Applying a Multicycle Clock Constraint

1. Right-click **Multicycle** under exception in the SmartTime constraints editor. The Set Multicycle Constraint dialog box is displayed.

et Multicycle Constraint	×
Specify multiplier(s) for: Setup Check only Setup and Hold Check 	;
Setup Path Multiplier:	
Default setup edge Hold edge	New setup edge
From:	
Through:	
2	
To:	
Comment:	
Help	Cancel

Figure 60 • Set Multicycle Constraint Dialog Box

- 2. Select the **Setup Check only** radio button as the multicycle constraint. It is applied to setup only for this design. For hold check, the default edge is used.
- 3. Enter Setup Path Multiplier (2, for example).



4. Click the browse button at the **Through** text box. The Select Through Pins for Multicycle Constraint dialog box is displayed with the list of available pins in the design, as shown in Figure 61.

Specify pins 💿 by explicit list	🔘 by keyword and wild	card	
Vailable Pins:		Assigned Pins:	
AND2_0:A AND2_0:B AND2_0:Y	Add >		
BUFD_0:A BUFD_0:Y BUFD_1:A	Add All >		
BUFD_1:Y BUFD_2:A BUFD_2:Y Bin1_and/uo/uo/BAD	< Remove		
Bin1_pad/U0/U0:Y Bin1_pad/U0/U1:Y Bin1_pad/U0/U1:YIN	< Remove All		
CLK pad/U0/U0:PAD		1	
Filter available pins:			
Pin Type: All Pins	•		
*	Filter		

Figure 61 • Adding Through Pins for Multicycle Path

5. Select a through pin (MX2_0:Y, for example) and then click **OK** to save this dialog box setting. On applying the multicycle path constraint, it will be shown in the constraint Editor.

🐙 File Edit View Actions Tools	Windo	w Help							
	2	2	< ®¤ 2	📑 📝 🦌 🦌 🦛	<u>>>></u>	<u>Fr</u> 😓	<u>n</u>	2 🕥	
E Constraints		Syntax	From	Through	To	Setup Cycles	Hold Cycles	Register Sensitivity	F
Clock		Click here	e to add a constraint		-				
Generated Clock	1	٣		[get_pins { MX2_0:Y }]		2		source register	6
Input Delay									
Output Delay									
Exceptions									
Max Delay									
Min Delay									

Figure 62 • Multicycle Path Constraints in Constraint Editor



Microsemi Corporate Headquarters One Enterprise Drive, Aliso Viejo CA 92656 Within the USA: (800) 713-4113 Outside the USA: (949) 221-7100 Fax: (949) 756-0308 · www.microsemi.com Microsemi Corporation (NASDAQ: MSCC) offers a comprehensive portfolio of semiconductor solutions for: aerospace, defense and security; enterprise and communications; and industrial and alternative energy markets. Products include high-performance, high-reliability analog and RF devices, mixed signal and RF integrated circuits, customizable SoCs, FPGAs, and complete subsystems. Microsemi is headquartered in Aliso Viejo, Calif. Learn more at **www.microsemi.com**.

© 2011 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.