

# Quartus II はじめてガイド TimeQuest によるタイミング制約の方法 ver.15



2016年2月 Rev.2

ELSENA,Inc.





# Quartus II はじめてガイド

TimeQuest によるタイミング制約の方法

# <u>目次</u>

1.	はじめに	3
2.	SDC ファイルの作成方法	5
2	2-1. Analysis & Synthesis(論理合成)の実行	. 5
2	2-2. TimeQuest Timing Analyzer を使って SDC ファイルの作成	. 6
	2-2-1. クロックの制約	.9
	2-2-2. I/O の制約	18
	2-2-3. フォルス・パスの制約	26
3.	SDC ファイルの登録とコンパイルの実行	30
3	-1. SDC ファイルの登録	30
3	3-2. コンパイルの実行	30
3	3-3. コンパイル・レポートの確認	31
改	版履歴	32

# 1. <u>はじめに</u>

この「Quartus II はじめてガイド」シリーズは、Quartus® II 開発ソフトウェアを初めてご利用になるユーザ向けの 資料です。



この資料は、FPGA/CPLD のデザイン(回路)に対して、タイミング制約を与える際に使用するタイミング制約用 ファイル(SDC ファイル)の作成方法について説明しています。上図の開発フローの中の「制約を設定する」で必要 となることが、主に書かれています。

元々は ASIC 業界の標準フォーマットになっている Synopsys Design Constraints (SDC) ファイルを FPGA/CPLD のタイミング制約に使用することで、Quartus II の Fitter(配置配線)で目標(ガイド)として参照する だけでなく、TimeQuest Timing Analyzer による高性能なタイミング解析にも使用されます。

なお、ユーザ・ロジック部分のタイミング制約は、設計者であるユーザが自分で制約する必要があります。IP (Intellectual Property) をデザイン内に使用している場合は、IP 部分に限って IP ベンダーが提供してくれるケー スが多いので、ベンダーに確認してください。

SDC ファイルを作成する方法は何通りかあります。SDC の文法を熟知しているユーザは テキスト・エディタで SDC ファイルを手動にて作成することもできますが、この資料では主に Quartus II の TimeQuest Timing Analyzer を使用して SDC ファイルを作成する方法について説明します。

この資料では、説明しているコマンドであっても、すべてのオプションについて説明をしていません。また、他に も様々なコマンドがあります。本資料は初めて使用する方向けの資料であるため、主に使用するオプションやコマ ンドに絞って説明します。

- 対象のツールおよびバージョン
  - Quartus II v15.0

この資料では、下記デザインを例に説明していきます。



# 2. <u>SDC ファイルの作成方法</u>

デザイン作成(回路設計)が終わったら、タイミング制約用の SDC ファイルを作成します。一般的な手順につい て説明します。

### 2-1. Analysis & Synthesis (論理合成)の実行

Quartus II の Processing メニュー ⇒ Start ⇒ Start Analysis & Synthesis または <sup>I</sup> で Analysis & Synthesis (論理合成)を実行します。エラーが出た場合は、エラー・メッセージを確認して、エラーを回避してください。もちろん、Fitter(配置配線)まで実行しても構いませんが、最低でも Analysis & Synthesis までは実行してください。

💱 Quartus II 64-Bit - C:/lab/TimeQue	st_In	tro/pipemult - pipemult				
File Edit View Project Assignments	Proc	essing Tools Window Help 🐬				
	<b>SP</b>	Stop Processing	Ctrl+Shift+C	2 <	) 💷 🕨 🛷 🖏 🖄 🖄 👘 👯 🖡	🚓 🕘 🕘 👗
Project Navigator		Start Compilation	Ctrl+L			
Q		Analyze Current File				
Entity		Start	÷	▶ <sub>a</sub>	Start Hierarchy Elaboration	
Cyclone V: 5CGXEC5C6E27C7		Update Memory Initialization File		2	Start Analysis & Elaboration	
Pipemult <sup>™</sup>	۹	Compilation Report	Ctrl+R	1	Start Analysis & Synthesis	Ctrl+K
	٩	Dynamic Synthesis Report		犞	Start Partition Merge	
	12	PowerPlay Power Analyzer Tool		×"	Start Fitter	
	27 111	SSN Apalyzer Tool		<b>1</b>	Start Assembler	
	T.M.	Solv Analyzer Tool		ð	Start TimeQuest Timing Analyzer	Ctrl+Shift+T
		Receive Compilation Status Notifications		M	Start EDA Netlist Writer	
<				5	Start Design Assistant	
Allerarchy 📄 Files 🖓 Design U	Inits			4	Start PowerPlay Power Analyzer	Ctrl+Shift+P
Tasks		₽₽×		N.	Start SSN Analyzer	0
Flow: Compilation	Cust	omize		1		
			くし	5	Start Rapid Recompile	
Task				1	Start SignalProbe Compilation	Ctrl+Shift+I
A Compile Design				≥io	Start I/O Assignment Analysis	
Analysis & Synthesis				3	Start Check & Save All Netlist Changes	
Fitter (Place & Route)				Ľ		
Assembler (Generate pro	gramm	ing files)			Start VQM Writer	
FDA Natiat Weiter	s				Start Equation Writer (Post-synthesis)	
Program Device (Open Brogram	mmer				Start Equation Writer (Post-fitting)	
riogram Device (Open Progra	niner)				Start Test Bench Template Writer	



2-2. TimeQuest Timing Analyzer を使って SDC ファイルの作成

Quartus II の Tools メニュー ⇒ TimeQuest Timing Analyzer または <sup>10</sup> で TimeQuest Timing Analyzer を起動します。

🚷 Quartus II 64-Bit - C:/lab/TimeQuest_Intro/pip	emult - pipemult		
File Edit View Project Assignments Processing	Tools Window Help 💎		
Project Navigator	Run Simulation Tool       >         Launch Simulation Library Compiler       >         Launch Design Space Explorer II       P         Flow Summary       Flow Status         TimeQuest Timing Analyzer       Flow Status	Successful - Tue Sep 06	
A Cyclone V: SCGXFC5C6F27C7     P      P pipemult ▲	Advisors     Quartus II 64-Bit Version       Quartus II 64-Bit Version     Revision Name       Top-level Entity Name     Top-level Entity Name       Top-level Entity Name     Family       Design Partition Planner     Device       Netlist Viewers     Logic utilization (in ALMs)	15.0.0 Build 145 04/22/ pipemult pipemult Cyclone V SCGXFCSC6F27C7 Final N/A	
✓     III       ▲ Hierarchy     Image: Files       J     Tasks       ₽     ₽       Φ     X	SignalTap II Logic Analyzer         Total registers           In-System Memory Content Editor         Total virtual pins           Iougic Analyzer Interface Editor         Total block memory bits           In-System Sources and Probes Editor         Total DSP Blocks           SignalProbe Pins         Total HSSI RX PCSs	48 46 0 512 1 0	
Flow: Compilation Customize Task Compile Design Analysis & Synthesis	<ul> <li>Programmer</li> <li>JTAG Chain Debugger</li> <li>Fault Injection Debugger</li> <li>System Debugging Tools</li> </ul>	0 0 1 0	
	<ul> <li>IP Catalog         <ul> <li>Nios II Software Build Tools for Eclipse</li> <li>Qsys</li> <li>Td Scripts</li> </ul> </li> </ul>		

次に、TimeQuest で Create Timing Netlist を実行して、タイミング用ネットリストを作成します。この時、先ほどの Quartus II で Analysis & Synthesis まで実行しているか、Fitter まで実行しているかで操作が異なります。

# ■ <u>Analysis & Synthesis まで実行している場合</u>

Netlist メニュー ⇒ Create Timing Netlist を実行後、Input netlist で Post-map を選択して OK

# ■ <u>Fitter まで実行している場合</u>

Netlist メニュー ⇒ Create Timing Netlist を実行後、Input netlist で Post-fit を選択して OK

または

Task ペインの Create Timing Netlist をダブルクリック

TimeQuest Timing Analyzer - C:/lab/TimeQuest_Intro/pipemult - pipemult	
File View Netlist Constraints Reports Script Tools Window Help 🌱	
Report Create Timing Netlist 5 Getting Started	
Repor Set Operating Conditions Update Timing N	Welcome to the TimeQuest Timing Analyzer
Delete Timing M Netlist $\checkmark = = = \Rightarrow$ Create Timing Netlist	The TimeQuest Timing Analyzer is a powerful ASIC-style timing analysis tool that validates the timing performance of all logic in the design using industry standard constraint, analysis, and reporting methodology. You can use the TimeQuest analyzer to constrain, run, and view results for all timing paths in your design. The TimeQuest Timing Analyzer offers the following graphical user interface features:
Tasks ♀ 5 × ♥ 🐼 Open Project ▶ Create Timing Netlist ▶ Create Timing Netlist ■ Reset Design ■ Set Design ■ Set Design ■ Set Design	Report Pane Lists generated report panels Track Pane 113sks 71JV2/
Coperation     C	clicka a command to start a process in the flow.

🗓 Create Timin	🗓 Create Timing Netlist 🛛 🗾						
Input netlist	Delay model						
	Slow-corner						
Post-fit	Specify Speed Grade						
	Speed grade: 7_H6 💌						
Post-map	Fast-corner						
	V Zero IC delays						
Analysis & Synthesis まで実行してい	<mark>る場合</mark> netlist -post_map -model slow -zero_ic_delays						
	OK Cancel Help						

続いて、TimeQuest の File メニュー ⇒ New SDC File で SDC エディタを起動します。



この後、クロックや I/O、フォルス・パスなどのタイミング制約コマンドを SDC エディタ上に記述していきます。こ こで、まだ何も記述していませんが、File メニュー ⇒ Save As を選択し、ファイル名を指定して保存(拡張子 は .sdc)しておきましょう。ファイル名はプロジェクトのトップ階層と同じ名前にしておくことをお勧めします。

TimeQuest のテキスト・エディタ(SDC エディタ)では、コマンドを挿入したい行にカーソルを合せた状態で、Edit メニュー ⇒ Insert Constraint を選ぶと、代表的なタイミング制約コマンドが表示されます。各コマンドを選択すると 専用の設定ウィンドウが現れるので、必要な制約情報を設定します。詳細は、各コマンドの説明で触れたいと思い ます。





2-2-1. クロックの制約

最初に、クロックの制約をすることをお勧めします。クロックの制約には FPGA/CPLD 外部から供給される基本 クロック(Base Clock)と FPGA/CPLD 内部で生成した生成クロック(Generated Clock)があり、それぞれ決まったコ マンドで制約します。PLL で生成したクロックも生成クロックに含まれます。

ただし、PLL で生成したクロックを自動的に制約してくれる便利なコマンドも用意されています。このコマンドを 使用する場合は、生成クロック用のコマンドを使用する必要はありません。

あとで見やすくするために、コメント行を入れておきましょう。# に続く文字はコメント扱いとなり、Quartus II が配置配線する時や TimeQuest がタイミング解析する時などは無視されます。



■ 基本クロック(Base Clock) <コマンド:create\_clock>

FPGA/CPLD 外部から供給されるクロックは、基本クロック(Base Clock)用のコマンドを使用します。SDC エディタでコマンドを挿入したい行にカーソルを合せた状態で Edit メニュー ⇒ Insert Constraint ⇒ Create Clock を選択すると、Create Clock 用の設定ウィンドウが表示されます。

🌵 Create C	lock					23				
Clock name:	clock_in_50mhz									
Period:	20.000	ns								
Waveform	edges									
Rising:		ns								
Falling:		ns	0.00	10.00		20.00		Name Finder		
Targets:			0.00	10.00			~			
SDC comman	d: create_clock -name cl	ock_in_50	mhz -period 20.00	0						
			Insert	÷	ハンドウ内	で指定した	た内容	系の SDC 記:	述が表示され	る

Clock name

TimeQuest 上で表記させたい名称を指定します。デザイン上の信号名と異なる名称にしたい場合に入力 します。これはオプションなので空白でも良いですが、空白にした時はデザインで使用している信号名が TimeQuest や Quartus II で使用されます。

Period

クロックの周期を指定します。

Waveform edges

デューティ比が 50%以外の時に、立ち上がりエッジ(Rising)と立ち下がりエッジ(Falling)の絶対時間を指定します。空白にした場合は、デューティ比が 50%と認識されます。

Target

ターゲットとなるクロックのポートやピンを指定します。TimeQuast の検索機能である Name Finder を使用して指定することをお勧めします。

Name Finder は、ターゲットとなるポートやピン、クロックなどを検索する機能です。Quartus II の Node Finder と似ています。

Name Finder の説明をする前に、SDC 用語について触れておきます。

用語	定義
Cell	ロジックを構成するブロック(例:LUT やレジスタ、組込み乗算器、メモリ・ブロック、I/O、PLL など)
Pin	Cell の入力または出力
Net	Pin 間の接続
Port	

※ Port と Pin の定義の違いに注意してください。

また、Name Finder で指定する Collections とは、Port や Pin など一致する名前のリストをデザインのネットリストから検索して抽出するために使用します。

Collections	意味
get_ports	デザインの入出カピン(デバイスのピン)
get_pins	各セルの入力や出力信号
get_clocks	定義済みのクロック信号
all_clocks	すべてのクロック信号
all_registers	すべてのレジスタ
all_inputs	すべての入力
all_outputs	すべての出力

検索する Collections を選択 Wame Finder Collection: get_ports ・ Filter: *	検索したい信号の一部(キーワード)に * を使って入力すると、 より詳細な検索が可能 (例:data*)
© Case-insensitive : Filter 欄に Hierarchical ⑦ Compatibility mode No duplicates	入力したキーワードは、大文字/小文字を区別しない りしたキーワードで一致したものを下位階層の信号を含めて表示させる 階層を検索する
Matches List 46 matches found atable dock_in datable Uストアップされた結果から、該当の ものを選択するために > をクリック	
Gtaba[0]       dataa[7]       dataa[7]       datab[0]       datab[1]       datab[2]       datab[2]       datab[3]       SDC command: [get_ports {dock_in}]	上記で検索した情報が記載される。 また、ここで最終確定するコマンドを手動修正できる。

最終確定させるには、Insert をクリックします。すると、SDC エディタに SDC コマンドが反映されます。

🔶 Create C	lock				×	
Clock name:	clock_in_50mhz					
Period:	20.000	ns				
Waveform	edges					
Rising:		ns				
Falling:		ns				
			0.00	10.00	20.00	
Targets:	[get_ports {clock_i	n}]				
Text Editor - C	C:/lab/Tin est	_Intro/pi	pemult - pip	emult - [pipemul	t.sdc]*	
e Edit View	Project ssir	ng Tools	Window H	lelp 🐬		
a 🖓 🖓	1 🛊 🤹 🚺	° 🛈 💽	2 0 🕅	267 ab/	🔿 🗆 🖾 🖻	
# Clock create_	<b>clock</b> -name (	clock_i	n_50mhz -	period 20.00	0 [get_ports	{clock_in}]

# ■ 生成クロック(Generated Clock) <コマンド:create\_generated\_clock>

分周クロックや PLL 生成クロックなどの FPGA/CPLD 内部で生成したクロックは、生成クロック (Generated Clock)用のコマンドを使用します。SDC エディタでコマンドを挿入したい行にカーソルを合せた 状態で Edit メニュー ⇒ Insert Constraint ⇒ Create Generated Clock を選択すると、Create Generated Clock 用の設定ウィンドウが表示されます。必要な指定を行ったら、Insert をクリックして SDC エディタに 反映させてください。(良く使用する設定内容のみ説明します。)

🔶 Create G	Generated Clock					×	
Clock name:	pll_clock_150mhz						Name Finder
Source:	[get_ports {clock_i	1}]					
Relationshi	p to source						
Based	on frequency						
Divide	by:	Phase:					
Multipl	y by: 3	Offset:					
Duty c	yde:						
Based	on waveform						
Edge li	st:						
Edge s	hift list:	ns		ns		ns	
Invert	waveform						
Targets:	s {pll_inst pll_ins	t altera_pll_i g	eneral[0].gpll/	-PLL_OUTF	PUT_COUNTER	divdk}]	Name Finder
SDC comman	ıd: ;et_pins {pll_inst	pll_inst altera_	_pll_i general[i	0].gpll~PLL		NTER  divclk}]	
			Inse	ert	9		
					ウィンド	ウ内で指定した	-内容の SDC 記述が表示される

# Clock name

TimeQuest 上で表記させたい名称を指定します。デザイン上の信号名と異なる名称にしたい場合に入力 します。これはオプションなので空白でも良いですが、空白にした時はクロックが割り当てられる最初のノ ード名となります。

# Source

生成クロックのリファレンス・クロックを指定します。分周クロックを生成する場合は分周前のクロックを指定し、PLL 生成クロックであれば PLL へ入力しているクロックを指定します。

# Divide by

リファレンス・クロックに対する分周比を指定します。

# Multiply by

リファレンス・クロックに対する逓倍比を指定します。

# Duty Cycle

デューティ比を指定します。単位は % です。

# Phase

リファレンス・クロックの立ち上がりエッジに対する位相シフト量を指定します。単位は 。(度)です。

Offset

リファレンス・クロックに対するオフセットを指定します。単位は ns です。

Invert waveform

指定した条件の論理反転したクロックを制約したい時にチェックします。

Target

ターゲットとなるクロックのポートやピンを指定します。

# ■ PLL 生成クロックの自動制約 <コマンド:derive\_pll\_clocks>

PLL で生成したクロックを自動的に制約してくれる便利なコマンドです。このコマンドを使用する場合は、 PLL で生成されたクロックに限って、Create Generated Clock コマンドを使用する必要はありません。SDC エディタでコマンドを挿入したい行にカーソルを合せた状態で Edit メニュー ⇒ Insert Constraint ⇒ Derive PLL Clocks を選択すると、Derive PLL Clocks 用の設定ウィンドウが表示されます。なお、複数の PLL を 使用していても、このコマンドは1度実行(記載)すれば大丈夫です。

🐳 Derive PLL Clocks 📃						
Options						
Create base clocks						
☑ Use net name as dock name						
SDC command: derive_pll_clocks -use_net_name						
Insert Cancel Help						

#### Create base clocks

基本クロック(Base Clock)も同時に制約してくれます。ただし、基本クロックは Create Clock コマンドを使用して自分で制約することをお勧めします。

Use net name as clock name

PLL 生成クロックの名前としてネット名を使用します。

# ■ FPGA/CPLD 内クロックのばらつきの自動制約 <コマンド:derive\_clock\_uncertainty>

FPGA/CPLD 内部のクロックのばらつき(スキューやガードバンドなど)を自動で制約して、タイミング解析 に反映してくれるコマンドです。SDC エディタでコマンドを挿入したい行にカーソルを合せた状態で Edit メ ニュー ⇒ Insert Constraint ⇒ Derive Clocks Uncertainty を選択すると、Derive Clocks Uncertainty 用の設 定ウィンドウが表示されます。



Add clock uncertainty assignment

Set Clock Uncertainty コマンドで既にクロックのばらつきの制約をしている場合は、Set Clock Uncertainty コマンドの制約値と Derive Clock Uncertainty コマンドの制約値が合算されます。

Overwrite existing clock uncertainty assignment

Set Clock Uncertainty コマンドで既にクロックのばらつきの制約をしている場合は、それを上書きします。

ここまで説明したコマンドを使うと、クロック関連の SDC 制約ができます。PLL を組み込んだデザインの場合、 以下のような SDC 記述になっていると思います。(この例では、PLL 生成クロックに derive\_pll\_clocks を使用し ています。PLL 生成クロックに create\_generated\_clock を使用した記述も書かれていますが、重複を避けるために コメント扱いしています。)

(例)



クロックの制約ができたら、クロックの制約を TimeQuest に読み込ませてみて、正しく TimeQuest が認識でき るかを確認してみましょう。TimeQuest の Tasks ペインにある Read SDC File をダブルクリックするか、 Constraints メニュー ⇒ Read SDC File で読み込ませる SDC ファイルを選択して読み込ませます。文法やスペ ルのミスなどで読み込みに失敗した場合は、Status が NG と表示されます。その時は SDC ファイルを確認して、 文法やスペルなどを確認して、修正して再度読み込ませてください。



次に Tasks ペインの Update Timing Netlist をダブルクリックして、タイミング解析用ネットリストにタイミング制約をアップデートします。



ここまでできたら、次に未制約クロックの有無を確認するために、Tasks ペインの Diagnostic  $\Rightarrow$  Report Unconstrained Paths をダブルクリックします。Unconstrained Clocks が 0 ならば、未制約クロックがないことを意味 します。



Unconstrained Clocks が 0 でなければ未制約クロックがあるということになるので、Report ペインの Unconstrained Paths ⇒ Clock Status Summary を見ると未制約クロックがレポートされているので確認してください。 未制約クロックの制約を SDC ファイルに追加して保存したら、Tasks ペインの Reset Design 実行後、Report Unconstrained Paths を再度実行して、未制約クロックがなくなっていることを確認してください。(この時、Read SDC File と Update Timing Netlist は自動で実行されます。)



SDC 記述内のターゲット指定やオプション記述が正しくない場合は、SDC の記述ミスがあっても、Read SDC File でエラーにならないことがあります。その場合は制約が無視されているので、Tasks ペインの Diagnostic ⇒ Report Ignored Constraints で無視された制約の有無を確認してください。No constraints were ignored. と表示されて いれば、無視された制約はありません。無視された制約がレポートされていたら SDC 記述を見直して、再度正し く反映されているかを確認してください。



# ALTIMA

次に、クロックの制約が正しく認識されているかを確認します。Tasks ペインの Diagnostic ⇒ Report Clocks を ダブルクリックすると、クロックの制約情報が表示されます。名前や Type、周期、周波数などが正しくレポートされ ているかを確認してください。

もし基本クロックの周波数が 1000MHz (1GHz) になっていたら制約されていないことが考えられますので、再 度見直してください。

🗓 TimeQuest Timing Analyzer - C:/lab/Ti	imeQu	est_Intro/pipemult - pipemult						
File View Netlist Constraints Reports	Script	Tools Window Help 💎						
Report	4 <del>8</del> (	Clocks Summary						
TimeQuest Timing Analyzer Summary		Clock Name	Туре	Period	Frequency	Rise	Fall	
SDC File List		1 dock_in_50mhz	Base	20.000	50.0 MHz	0.000	10.000	
Clocks Summary		2 pll:pll_inst pll_0002:pll_inst altera_pll:altera_pll_i general[0].gpll~FRACTIONAL_PLL_O_VCOPH0	Generated	3.333	300.03 MHz	0.000	1.666	50
		3 pll:pll_inst pll_0002:pll_inst altera_pll:altera_pll_i outdk_wire[0]	Generated	6.666	150.02 MHz	0.000	3.333	50
Tasks 7	₽× ^							
Report Metastability Summary     Diagnostic     Report Clocks	^							
1 Report Clock Transfers ダブ	ブルク	Uv/						
Report Unconstrained Paths								
Report SDC								
Report Partitions								

# 🔥 ALTIMA

2-2-2. I/O の制約

クロックの制約の次は、I/O の制約を行いましょう。ここでは、前段のデバイスや後段のデバイスのタイミング・ パラメータ値などから、ターゲットとなる FPGA/CPLD の I/O タイミングを制約する方法を紹介します。

■ 入力 I/O の制約 <コマンド:set\_input\_delay>

下記のようなデバイスを跨いでいる信号であっても、同じクロック源で動作しているので同期動作として扱えます。

入力 I/O を制約するには、前段のデバイスの Clock-to-Output 時間(Tco)の値を予めデータシートなどで 調べておく必要があります。また、前段のデバイスからターゲットとなる FPGA/CPLD までのボード上の信 号伝搬遅延(Tdata\_PCB)やオシレータ(OSC)からデバイスまでのクロック伝搬遅延(Tclk1 や Tclk2ext)もタ イミング制約に盛り込むと、実機に近いタイミング解析ができます。この set\_input\_delay コマンドは、最大値 (-max オプション)と最小値(-min オプション)のセットで制約します。前段のデバイスの Tco は、最大値と 最小値を調べておいてください。



入力/出力 I/O の制約値を算出するには、まず用語と考え方を理解する必要があります。本資料では、用語と考え方の説明を省略しています。詳細は、本資料を入手したサイト内から以下の資料を入手してご覧ください。

TimeQuest ~タイミング解析の用語と考え方~

# <入力セットアップ時間の制約>

入力セットアップ時間の制約における Data Arrival Time (Max) と Data Required Time (Setup) は、以下のようになります。

Data Arrival Time (Max) =  $Tclk1 + Tco(Max) + Tdata_PCB$ 

Data Required Time (Setup) = (Tclk2ext + Tclk2int) - uTsu

Data Arrival Time (Max) はデータ到達時間であり、データが FPGA/CPLD の入力ポートに到達する時間 を表します。Data Required Time (Setup) はセットアップ側のデータ要求時間であり、ここまでにデータが確定 (到達)している必要がある時間を表します。そして、セットアップ・スラックは、以下の式で表すことができま す。

```
Setup Slack = Data Required Time (Setup) - Data Arrival Time (Max)
```

セットアップ・スラックが正の値であれば、タイミング制約(要求)を満たしているということができ、その値が 大きければ大きいほどスラック(余裕度)が大きいということになります。それに対して、セットアップ・スラック が負の値であれば、タイミング制約(要求)を満たしていないということになります。

入力セットアップ時間は、set\_input\_delay コマンドの -max オプションを使用して制約します。制約の値は、 以下の式で算出します。なお、あまり話を難しくしないために、ボード上の伝搬遅延(Tdata\_PCB と Tclk1、 Tclk2ext)は一定とします。

```
最大入力遅延 = ボード上の遅延 - ボード上のクロック・スキュー + Tco(max)
= Tdata_PCB - (Tclk2ext - Tclk1) + Tco(max) …式①
```

最大入力遅延が求まったら、SDC エディタに制約を記述してみましょう。クロックの時と同じように、コメント 行を入れておきましょう。

7 8 # I/O	
9	# に続く文字はコメント扱い
	(SDC エディタでは 緑字 で表示)

SDC エディタでコマンドを挿入したい行にカーソルを合せた状態で Edit メニュー ⇒ Insert Constraint ⇒ Set Input Delay を選択すると、Set Input Delay 用の設定ウィンドウが表示されます。(良く使用する設定内容のみ説明します。)

🚸 Set Input I	Delay			
Clock name:	clock_in_50mhz			
	Use falling clock edge			
Input delay o	ptions			
Minimur	n	Rise		
Maximu	ım	🔘 Fall		
🔘 Both		O Both		
		\		
Delay value:	5		ns 🔲 /	Add delay Name Finder
Targets:	[get_ports {dataa[0]}]			
SDC command:	set_input_delay -clock { clock_in	_50mhz } -max 5 [	get_ports {da	dataa[0]}]
		Insert	Cancel	Help
				バスの提合 1ビット分だけ Name Finder で選択して こ
				こで * を使って編集することができます
				例)
				修正前:dataa[0]
				修正後:dataa[*]

# Clock name

基準となるクロックを選択します。なお、事前にクロックの制約を読み込ませて Update Timing Netlist を 実行していないと選択できません。クロックの立ち下がりエッジを基準にしたい場合は、Use falling clock edge にチェックを入れます。

Input delay options

最大入力遅延を制約する場合は、Maximum を選択します。

Delay value

上記の式①で算出した値を入力します。

Target

ターゲットとなる入力 I/O ポートを指定します。Name Finder を使用して指定することをお勧めします。

#### <入力ホールド時間の制約>

入力ホールド時間の制約における Data Arrival Time (Min) と Data Required Time (Hold) は、以下のよう になります。

Data Arrival Time (Min) =  $Tclk1 + Tco(min) + Tdata_PCB$ 

Data Required Time (Hold) = (Tclk2ext + Tclk2int) + uTh

Data Arrival Time (Min) はデータ到達時間であり、データが FPGA/CPLD の入力ポートに到達する時間 を表します。Data Required Time (Hold) はホールド側のデータ要求時間であり、ここまでデータが保持され ている必要がある時間を表します。そして、ホールド・スラックは、以下の式で表すことができます。

Hold Slack = Data Arrival Time (Min) - Data Required Time (Hold)

ホールド・スラックが正の値であれば、タイミング制約(要求)を満たしているということができ、その値が大きければ大きいほどスラック(余裕度)が大きいということになります。それに対して、ホールド・スラックが負の値であれば、タイミング制約(要求)を満たしていないということになります。

入力ホールド時間は、set\_input\_delay コマンドの -min オプションを使用して制約します。制約の値は、以下の式で算出します。なお、あまり話を難しくしないために、ボード上の伝搬遅延(Tdata\_PCB と Tclk1、Tclk2ext)は一定とします。

最小入力遅延 = ボード上の遅延 - ボード上のクロック・スキュー + Tco(min) = Tdata\_PCB - (Tclk2ext - Tclk1) + Tco(min) …式②

SDC エディタでコマンドを挿入したい行にカーソルを合せた状態で Edit メニュー ⇒ Insert Constraint ⇒ Set Input Delay を選択すると、Set Input Delay 用の設定ウィンドウが表示されます。(良く使用する設定内容のみ説明します。)

Clock name

基準となるクロックを選択します。なお、事前にクロックの制約を読み込ませて Update Timing Netlist を 実行していないと選択できません。クロックの立ち下がりエッジを基準にしたい場合は、Use falling clock edge にチェックを入れます。

Input delay options

最小入力遅延を制約する場合は、Minimum を選択します。

Delay value

上記の式②で算出した値を入力します。

<u>Target</u>

ターゲットとなる入力 I/O ポートを指定します。Name Finder を使用して指定することをお勧めします。

# ■ 出力 I/O の制約 <コマンド:set\_output\_delay>

下記のようなデバイスを跨いでいる信号であっても、同じクロック源で動作しているので同期動作として扱えます。

出力 I/O を制約するには、後段のデバイスのセットアップ時間(Tsu)とホールド時間(Th)の値を予めデー タシートなどで調べておく必要があります。また、ターゲットとなる FPGA/CPLD から後段のデバイスまでの 信号伝搬遅延(Tdata\_PCB)やオシレータ(OSC)からデバイスまでのクロック伝搬遅延(Tclk1ext や Tclk2) もタイミング制約に盛り込むと、実機に近いタイミング解析ができます。この set\_output\_delay コマンドは、最 大値(-max オプション)と最小値(-min オプション)のセットで制約します。



# <br /> **<出力最大** Clock-to-Output 時間の制約>

出力最大 Clock-to-Output 時間の制約における Data Arrival Time (Max) と Data Required Time (Setup) は、以下のようになります。

Data Arrival Time (Max) = (Tclk1ext + Tclk1int) + uTco(max) + (Tdataint + Tdata\_PCB)

Data Required Time (Setup) = Tclk2 - Tsu

Data Arrival Time (Max) はデータ到達時間であり、データが後段のデバイスの入力ポートに到達する時間 を表します。Data Required Time (Setup) はセットアップ側のデータ要求時間であり、ここまでにデータが確定 (到達)している必要がある時間を表します。そして、セットアップ・スラックは、以下の式で表すことができま す。

```
Setup Slack = Data Required Time (Setup) - Data Arrival Time (Max)
```

セットアップ・スラックが正の値であれば、タイミング制約(要求)を満たしているということができ、その値が 大きければ大きいほどスラック(余裕度)が大きいということになります。それに対して、セットアップ・スラック が負の値であれば、タイミング制約(要求)を満たしていないということになります。

出力最大 Clock-to-Output 時間は、set\_output\_delay コマンドの -max オプションを使用して制約します。 制約の値は、以下の式で算出します。なお、あまり話を難しくしないために、ボード上の伝搬遅延 (Tdata\_PCB と Tclk1ext、Tclk2)は一定とします。

```
最大出力遅延 = ボード上の遅延 - ボード上のクロック・スキュー + Tsu
= Tdata PCB - (Tclk2 - Tclk1ext) + Tsu ・・・式3
```

SDC エディタでコマンドを挿入したい行にカーソルを合せた状態で Edit メニュー ⇒ Insert Constraint ⇒ Set Output Delay を選択すると、Set Output Delay 用の設定ウィンドウが表示されます。(良く使用する設定 内容のみ説明します。)

※ Set Output Delay 用の設定ウィンドウは Set Input Delay 用の設定ウィンドウとほぼ同じです。

#### Clock name

基準となるクロックを選択します。なお、事前にクロックの制約を読み込ませて Update Timing Netlist を 実行していないと選択できません。クロックの立ち下がりエッジを基準にしたい場合は、Use falling clock edge にチェックを入れます。

Output delay options

最大出力遅延を制約する場合は、Maximum を選択します。

Delay value

上記の式③で算出した値を入力します。

Target

ターゲットとなる出力 I/O ポートを指定します。Name Finder を使用して指定することをお勧めします。

#### <u><出力最小 Clock-to-Output 時間の制約></u>

出力最小 Clock-to-Output 時間の制約における Data Arrival Time (Min) と Data Required Time (Hold) は、以下のようになります。

Data Arrival Time (Min) =  $(Tclk1ext + Tclk1int) + uTco(min) + (Tdataint + Tdata_PCB)$ 

Data Required Time (Hold) = Tclk2 + Th

Data Arrival Time (Min) はデータ到達時間であり、データが後段のデバイスの入力ポートに到達する時間 を表します。Data Required Time (Hold) はホールド側のデータ要求時間であり、ここまでデータが保持され ている必要がある時間を表します。そして、ホールド・スラックは、以下の式で表すことができます。

Hold Slack = Data Arrival Time (Min) - Data Required Time (Hold)

ホールド・スラックが正の値であれば、タイミング制約(要求)を満たしているということができ、その値が大きければ大きいほどスラック(余裕度)が大きいということになります。それに対して、ホールド・スラックが負の値であれば、タイミング制約(要求)を満たしていないということになります。

出力最小 Clock-to-Output 時間は、set\_output\_delay コマンドの -min オプションを使用して制約します。 制約の値は、以下の式で算出します。なお、あまり話を難しくしないために、ボード上の伝搬遅延 (Tdata\_PCB と Tclk1ext、Tclk2)は一定とします。

最小入力遅延 = ボード上の遅延 - ボード上のクロック・スキュー - Th = Tdata\_PCB - (Tclk2 - Tclk1ext) - Th ····式④ SDC エディタでコマンドを挿入したい行にカーソルを合せた状態で Edit メニュー ⇒ Insert Constraint ⇒ Set Output Delay を選択すると、Set Output Delay 用の設定ウィンドウが表示されます。(良く使用する設定 内容のみ説明します。)

Clock name

基準となるクロックを選択します。なお、事前にクロックの制約を読み込ませて Update Timing Netlist を 実行していないと選択できません。クロックの立ち下がりエッジを基準にしたい場合は、Use falling clock edge にチェックを入れます。

Output delay options

最小出力遅延を制約する場合は、Minimum を選択します。

Delay value

上記の式④で算出した値を入力します。

Target

7

ターゲットとなる出力 I/O ポートを指定します。Name Finder を使用して指定することをお勧めします。

ここまで説明したコマンドを使うと、I/O 関連の SDC 制約ができます。

(例)

8	# I/O
9	set input delay -clock { clock in 50mhz } -max 5 [get ports {dataa[*]}]
10	set input delay -clock { clock in 50mhz } -min 1 [get ports {dataa[*]}]
11	set input delay -clock { clock in 50mhz } -max 5 [get ports {datab[*]}]
12	<pre>set input delay -clock { clock in 50mhz } -min 1 [get ports {datab[*]}]</pre>
13	set input delay -clock { clock in 50mhz } -max 5 [get ports {rdaddress[*]}]
14	set input delay -clock { clock in 50mhz } -min 1 [get ports {rdaddress[*]}]
15	set input delay -clock { clock in 50mhz } -max 5 [get ports {wraddress[*]}]
16	set input delay -clock { clock in 50mhz } -min 1 [get ports {wraddress[*]}]
17	set input delay -clock { clock in 50mhz } -max 5 [get ports {wren}]
18	set input delay -clock { clock in 50mhz } -min 1 [get ports {wren}]
19	
20	set output delay -clock { clock in 50mhz } -max 1 [get ports {g[*]}]
21	set output delay -clock { clock in 50mhz } -min 0.5 [get ports {g[*]}]
22	

I/O の制約ができたら、クロックの制約のところで説明したのと同じように、I/O の制約も TimeQuest に読み込ませてみて、正しく TimeQuest が認識できるかを確認してみましょう。TimeQuest の Tasks ペインにある Reset Design をダブルクリックして、タイミング解析用ネットリストを生成した状態に戻します。その後、Read SDC File をダブルクリックするか、Constraints メニュー ⇒ Read SDC File で読み込ませる SDC ファイルを選択して読み込ませます。文法やスペルのミスなどで読み込みに失敗した場合は、Status が NG と表示されます。その時は SDC ファイルを確認して、文法やスペルなどを確認して、修正して再度読み込ませてください。

次に Tasks ペインの Update Timing Netlist をダブルクリックして、タイミング解析用ネットリストにタイミング制約をアップデートします。

ここまでできたら、次に未制約ポートの有無を確認するために、Tasks ペインの Diagnostic ⇒ Report Unconstrained Paths をダブルクリックします。Unconstrained Input Port や Unconstrained Output Port が 0 ならば、 未制約 I/O がないことを意味します。

Unconstrained Input Port や Unconstrained Output Port が 0 でなければ未制約ポートがあるということになるの で、Report ペインの Unconstrained Paths ⇒ Setup Analysis や Hold Analysis ⇒ Unconstrained Input Port や Unconstrained Output Port を見ると、未制約ポートがレポートされているので確認してください。未制約ポートの制 約を SDC ファイルに追加して保存したら、Tasks ペインの Reset Design 実行後、Report Unconstrained Paths を 再度実行して、未制約ポートがなくなっていることを確認してください。(この時、Read SDC File と Update Timing Netlist は自動で実行されます。)



SDC 記述内のターゲット指定やオプション記述が正しくない場合は、SDC の記述ミスがあっても、Read SDC File でエラーにならないことがあります。その場合は制約が無視されているので、Tasks ペインの Diagnostic ⇒ Report Ignored Constraints で無視された制約の有無を確認してください。No constraints were ignored. と表示されて いれば、無視された制約はありません。無視された制約がレポートされていたら SDC 記述を見直して、再度正し く反映されているかを確認してください。

次に、I/O の制約が正しく認識されているかを確認します。Tasks ペインの Diagnostic ⇒ Report SDC をダブ ルクリックすると、クロックや I/O ポートの制約情報が表示されます。名前や制約情報が正しくレポートされている かを確認してください。

📋 TimeQuest Timing Analyzer - C:/lab/TimeQue	st_Intro/pipemult/p	ipemult - pipem	nult						
File View Netlist Constraints Reports Script	Tools Window Help								Se
Report	Set Input Delay								
SDC File List	SDC Command	Add Delay	Source Latency Included	Clock Fall	Flags	Clock Name	Reference Pin	Delay	Ports
4 🦾 SDC Assignments	1 set input delay	-add delav			-max	faet clocks clock in 50mhz]		5.000	foet ports dataa[0]]
Treate Clock	2 set input delay	-add delay			-min	[get clocks clock in 50mhz]		1.000	[get_ports dataa[0]]
Create Generated Clock	3 set_input_delay	-add_delay			-max	[get_clocks clock_in_50mhz]		5.000	[get_ports dataa[1]]
	4 set_input_delay	-add_delay			-min	[get_clocks clock_in_50mhz]		1.000	[get_ports dataa[1]]
	5 set_input_delay	-add_delay			-max	[get_clocks clock_in_50mhz]		5.000	[get_ports dataa[2]]
E Set Input Delay	5 set_input_delay	-add_delay			-min	[get_clocks clock_in_50mhz]		1.000	[get_ports dataa[2]]
Set Output Delay	7 set_input_delay	-add_delay			-max	[get_clocks clock_in_50mhz]		5.000	[get_ports dataa[3]]
Tasks 🕂 🗗 🗙 8	8 set_input_delay	-add_delay			-min	[get_clocks clock_in_50mhz]		1.000	[get_ports dataa[3]]
V 😼 Open Project	<pre>set_input_delay</pre>	-add_delay			-max	[get_clocks clock_in_50mhz]		5.000	[get_ports dataa[4]]
Netlist Setun	10 set_input_delay	-add_delay			-min	[get_clocks clock_in_50mhz]		1.000	[get_ports dataa[4]]
Create Timing Netlist	11 set_input_delay	-add_delay			-max	[get_clocks clock_in_50mhz]		5.000	[get_ports dataa[5]]
Pand SDC File	12 set_input_delay	-add_delay			-min	[get_clocks clock_in_50mhz]		1.000	[get_ports dataa[5]]
V Read SDCT lie	13 set_input_delay	-add_delay			-max	[get_clocks clock_in_50mhz]		5.000	[get_ports dataa[6]]
Populate finning webist     E 1	14 set_input_delay	-add_delay			-min	[get_clocks clock_in_50mhz]		1.000	[get_ports dataa[6]]
Reset Design	15 set_input_delay	-add_delay			-max	[get_clocks clock_in_50mhz]		5.000	[get_ports dataa[7]]
set Operating Conditions	16 set_input_delay	-add_delay			-min	[get_clocks clock_in_50mhz]		1.000	[get_ports dataa[7]]
Reports 1	17 set_input_delay	-add_delay			-max	[get_clocks clock_in_50mhz]		5.000	[get_ports datab[0]]
Slack	18 set_input_delay	-add_delay			-min	[get_clocks clock_in_50mhz]		1.000	[get_ports datab[0]]
Datasheet	19 set_input_delay	-add_delay			-max	[get_clocks clock_in_50mhz]		5.000	[get_ports datab[1]]
Device Specification (1998)	20 set_input_delay	-add_delay			-min	[get_clocks clock_in_50mhz]		1.000	[get_ports datab[1]]
🌯 🗁 Diagnostic ダブルクレック	21 set_input_delay	-add_delay			-max	[get_clocks clock_in_50mhz]		5.000	[get_ports datab[2]]
Report 777777	22 set_input_delay	-add_delay			-min	[get_clocks clock_in_50mhz]		1.000	[get_ports datab[2]]
📆 Report 🔄 🚽	23 set_input_delay	-add_delay			-max	[get_clocks clock_in_50mhz]		5.000	[get_ports datab[3]]
Report Clock rers	24 set_input_delay	-add_delay			-min	[get_clocks clock_in_50mhz]		1.000	[get_ports datab[3]]
Report Onco ined Paths	25 set_input_delay	-add_delay			-max	[get_clocks clock_in_50mhz]		5.000	[get_ports datab[4]]
V Report SDC 2	26 set input delay	-add_delay			-min	[aet_clocks.clock_in_50mbz]		1.000	[get_ports_datab[4]]

# 2-2-3. フォルス・パスの制約

クロックと I/O のタイミング制約を掛けた状態で Report Unconstrained Path に未制約のパスが残っている場合 は、非同期ポートである可能性があります。デザイン(回路)の動作に関係しないパスや設計者が意図的にタイミ ング解析をしなくても良いと考えているパスであっても、TimeQuest はデフォルトでこれらのパスもタイミング解析 の対象とします。タイミング解析をする必要がないパスはフォルス・パスの制約を掛けることで、タイミング解析の 対象から除外することができます。

下図のような非同期の入力ポート/出力ポートのパスや異なるクロック・ドメイン間の乗せ換えのパス、クロッ ク・マルチプレクサは、フォルス・パスに制約できる代表例になります。

(例1)非同期の入力ポート



例1のケースは、areset 信号の生成元が完全に非同期である場合、set\_false\_path コマンドで制約できます。フォルス・パスの制約を掛けない場合は、Recovery/Removal の解析対象となります。 (Recovery/Removal については、本資料では説明を省略しています。)

# (例2)異なるクロック・ドメイン間の乗せ換えのパス



例2のケースは、set\_false\_path コマンドか set\_clock\_groups コマンドの -asynchronous オプションのどち らかで制約できます。フォルス・パスの制約を掛けない場合は、Setup/Hold の解析対象となります。

(例3)クロック・マルチプレクサ



例3のケースは、set\_false\_path コマンドか set\_clock\_groups の -exclusive オプションのどちらかで制約で きます。フォルス・パスの制約を掛けない場合は、あり得ないパターンの解析も行ってしまうので、フォルス・ パスの制約を掛けて、余計なパスの解析をしないようにします。

(あり得るパターン): 通常の同期パスとして解析する必要あり

From Clock : clock\_a — To Clock : clock\_a

From Clock : clock\_b — To Clock : clock\_b

(あり得ないパターン):存在しないのでフォルス・パスの制約で除外

From Clock : clock\_a — To Clock : clock\_b

From Clock : clock\_b — To Clock : clock\_a

Tasks ペインの Diagnostic ⇒ Report Clock Transfer を実行すると、From Clock と To Clock それぞれク ロック・ドメイン間のパス数をレポートしてくれます。同一クロック・ドメイン間のパスであれば、クロック制約で すでにタイミング制約ができていることになります。もし異なるクロック・ドメイン間のパスがあれば、必要に 応じてフォルス・パスの制約を行います。(ここでいう必要に応じてとは、異なるクロック・ドメイン間のデータ 伝送でタイミング解析が必要でないパスやあり得ないパターンのパスを指します。)

Setup Transfers								
	From Clock	To Clock	RR Paths	FR Paths	RF Paths	FF Paths		
1	pll altpll_component auto_generated pll1 clk[0]	pll/altpll_component/auto_generated/pll1/clk[0]	33	0	0	0		
2	plialtpli_component auto_generated pli1 ck[1]	pll/altpll_component/auto_generated/pll1/clk[0]	16	0	16	0		
3	pll altpll_component auto_generated pll1 clk[0]	pll/altpll_component/auto_generated/pll1/clk[1]	48	16	0	0		
4	pll altpll_component auto_generated pll1 clk[1]	pll/altpll_component/auto_generated/pll1/clk[1]	272	0	0	0		

RR Paths : 送信レジスタ(reg1)が立ち上がりエッジで、受信レジスタ(reg2)が立ち上がりエッジのパス数 FR Paths : 送信レジスタ(reg1)が立ち下がりエッジで、受信レジスタ(reg2)が立ち上がりエッジのパス数 RF Paths : 送信レジスタ(reg1)が立ち上がりエッジで、受信レジスタ(reg2)が立ち下がりエッジのパス数 FF Paths : 送信レジスタ(reg1)が立ち下がりエッジで、受信レジスタ(reg2)が立ち下がりエッジのパス数

フォルス・パス制約するパスが決まったら、SDC エディタに制約を記述してみましょう。クロックや I/O の時と同じように、コメント行を入れておきましょう。

22 23	# Fals	- e Path	
24			# に続く文字はコメント扱い
			(SDC エディタでは 緑字 で表示)

# ■ 特定のノードに指定するフォルス・パスの制約 <コマンド:set\_false\_path>

SDC エディタでコマンドを挿入したい行にカーソルを合せた状態で Edit メニュー ⇒ Insert Constraint ⇒ Set False Path を選択すると、設定ウィンドウが表示されます。(良く使用する設定内容のみ説明します。)

#### From

ポイントでフォルス・パス指定する場合は、始点のポート(get\_ports)またはピン(get\_pins)で指定します。 クロック・ドメインで指定したい場合は、From Clock 側のクロック(get\_clocks)で指定します。その際、事 前にクロックの制約を読み込ませて Update Timing Netlist を実行していないと、Name Finder では選択 できません。

# Through 1997

経由するポイントを指定したい場合は、ここに指定します。

<u>To</u>

ポイントでフォルス・パス指定する場合は、終点のポート(get\_ports)またはピン(get\_pins)で指定します。 クロック・ドメインで指定したい場合は、To Clock 側のクロック(get\_clocks)で指定します。その際、事前に クロックの制約を読み込ませて Update Timing Netlist を実行していないと、Name Finder では選択でき ません。

# ① ポイントで指定

👾 Set False P	ath	×
From:	[get_ports {areset}]	
Through:		
To:		
SDC command:	set_false_path -from [get_ports {areset}]	
	Insert Cancel Help	

# ② クロックで指定

From: [get docks {dock a}]
Through:
To: [get_docks {dock_b}]
SDC command: set_false_path -from [get_clocks {clock_a}] -to [get_clocks {clock_b}]
Insert Cancel Help

# ■ クロック・ドメインで指定するフォルス・パスの制約 <コマンド:set\_clock\_groups>

SDC エディタでコマンドを挿入したい行にカーソルを合せた状態で Edit メニュー ⇒ Insert Constraint ⇒ Set Clock Groups を選択すると、設定ウィンドウが表示されます。(良く使用する設定内容のみ説明します。)

Group

クロック・ドメインをクロック(get\_clocks)で指定します。設定ウィンドウでは 2 つのグループ(クロック・ドメ イン)までしか指定できませんが、3 つ以上のグループを制約したい場合は、SDC エディタに反映させた 後に手動で編集します。

# ① -exclusive オプションで指定

Group で指定したクロック同士が同時に有効にならない時に使用するオプションです。SDC command 欄に -exclusive の記述がない場合は、手動で追加記述してください。

👾 Set Clock 🛛	Groups	
Group 1:	[get_clocks {clock_a}]	]
Group 2:	[get_docks {dock_b}]	]
SDC command:	set_dock_groups -exclusive -group [get_docks {dock_a}] -group [get_docks {dock_	]
	Libert Cancel Help	) i

# ② -asynchronous オプションで指定

Group で指定したクロック・ドメイン間のデータ伝送をタイミング解析の対象から除外したい時に使用する オプションです。SDC command 欄に -asynchronous の記述がない場合は、手動で追加記述してください。

🔶 Set Clock (	Groups	×
Group 1:	[get_docks {dock_a}]	
Group 2:	[get_docks {dock_b}]	
SDC command:	set_dock_groups -asynchronous -group [get_docks {dock_a}] -group [get_dock	:ks {d
	Insert Cancel Help	

ここまでで説明したコマンドの中には、説明していないオプションが存在しているものがあります。また、非同期 パスの制約やマルチ・サイクル・パスの制約などの説明していないコマンドも多々ありますが、本資料は初心者向 けの資料であるため、これらは割愛しています。

# 3. SDC ファイルの登録とコンパイルの実行

SDC ファイルが完成したら、Quartus II で SDC ファイルの登録とコンパイルを実行します。

## 3-1. SDC ファイルの登録

コンパイル実行前に、Assignments メニュー  $\Rightarrow$  Settings  $\Rightarrow$  TimeQuest Timing Analyzer で SDC ファイルが Quartus II に登録されていることを確認します。登録されていなかったら、作成した SDC ファイルを忘れずに登録 してください。なお、SDC ファイルは、ユーザ・ロジックと IP 部分などのように複数登録することができます。その 際は、上から順に読み込まれます。SDC 記述の仕方によっては読み込む順番が重要になるので、注意してくださ い。読み込む順番は、Up や Down で変更できます。

🖌 Settings - pipemult		
Category:		Device
General Files	TimeQuest Timing Analyzer	-
Libraries	Specify TimeQuest Timing Analyzer options.	選択して追加
<ul> <li>IP Settings         IP Catalog Search Locations     </li> <li>Design Templates</li> </ul>	SDC files to include in the project	
<ul> <li>Operating Settings and Conditions</li> <li>Voltage</li> </ul>	File name:	Add
Temperature	File Name Type	Remove
Compilation Process Settings Incremental Compilation     EDA Tool Settings Design Entry/Synthesis Simulation Formal Verification Board-Level     Compiler Settings VHDL Input Verilog HDL Input Default Parameters	pipemult.sdc Synopsys Design Constraints File 読み込む	Up Down 順番の変更
TimeQuest Timing Analyzer Assembler Design Assistant SignalTap II Logic Analyzer Logic Analyzer Interface PowerPlay Power Analyzer Settings SSN Analyzer	Enable Advanced I/O Timing     Report worst-case paths during of     Td Script File for customizing reports during compilation     Td Script File name:     IV     Run default timing analysis before running custom script	ompilation
	Metastability analysis Synchronizer identification: Off	•

### 3-2. コンパイルの実行

フル・コンパイルを実行します。エラーが出たら、エラー・メッセージを確認して、エラーを回避してください。

なお、フル・コンパイル中にタイミング解析が行われますが、配置配線した結果がタイミング制約を満たせない場合も、エラーで止まることなくコンパイルは最後まで実行されます。

# 3-3. コンパイル・レポートの確認

コンパイルが終わったら、コンパイル・レポートを確認してください。コンパイル・レポートではフェーズ毎に詳細 なコンパイル結果を確認することができますが、タイミング解析についてはサマリー情報のみ確認できます。配置 配線した結果がタイミング制約を満たせないなどのタイミング違反があった場合は、赤字で表示されます。コンパ イル後には、まず赤字の有無を確認してください。



詳細なタイミング解析結果は、TimeQuest を起動して確認することになります。TimeQuest を使用したタイミング 解析結果の確認方法は、本資料を入手したサイト内から以下の資料を入手してご覧ください。

Quartus II はじめてガイド – TimeQuest によるタイミング解析の方法



# <u> 改版履歴</u>

Revision	年月	概要
1	2015年9月	初版
2	2016年2月	誤記訂正
		・「2-2-2 I/O の制約」の「<出力最大 Clock-to-Output 時間の制約>」と 「<出力最小 Clock-to-Output 時間の制約>」の Target の説明
		誤)ターゲットとなる入力 I/O ポートを指定します。
		正)ターゲットとなる出力 I/O ポートを指定します。

#### 免責およびご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

- 1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
- 2. 本資料は予告なく変更することがあります。
- 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、本資料を入手されました下記代理店までご一報いただければ幸いです。
   株式会社アルティマ ホームページ: http://www.altima.co.jp
   技術情報サイト EDISON: http://www.altima.jp/members/index.cfm
   株式会社エルセナ ホームページ: http://www.elsena.co.jp
   技術情報サイト ETS : http://www.elsena.co.jp/elspear/members/index.cfm
- 4. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。

5. 本資料は製品を利用する際の補助的な資料です。製品をご使用になる際は、各メーカ発行の英語版の資料もあわせてご利用ください。