

Quartus[®] Prime はじめてガイド TimeQuest によるタイミング制約の方法

Ver.17

Quartus® Prime はじめてガイド

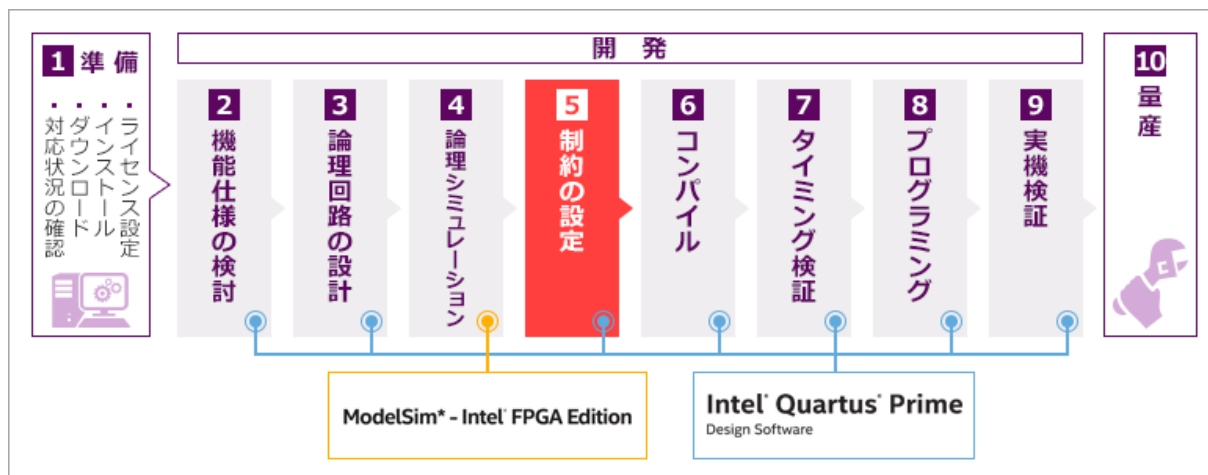
TimeQuest によるタイミング制約の方法

目次

1. はじめに	3
2. SDC ファイルの作成方法	5
2-1. Analysis & Synthesis(論理合成)の実行	5
2-2. TimeQuest Timing Analyzer を使って SDC ファイルを作成	5
2-2-1. クロックの制約	8
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
3-2. コンパイルの実行	30
3-3. コンパイル・レポートの確認	31
改版履歴	32

1. はじめに

この「Quartus® はじめてガイド」シリーズは、インテル® Quartus® Prime / Quartus® II 開発ソフトウェアを使用する初級ユーザ向けの資料です。



この資料は、FPGA 開発の「5. 制約の設定」フェーズで非常に参考になります。

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® Prime 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。

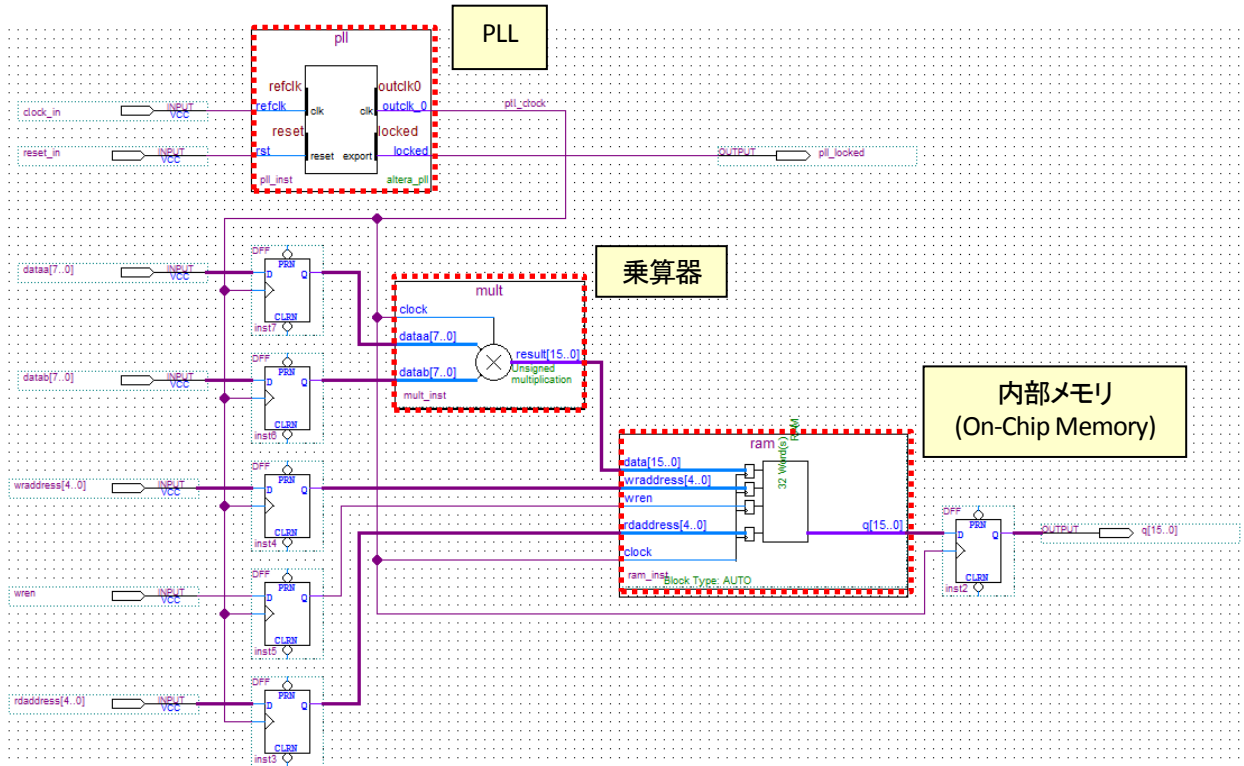
SDC ファイルを作成する方法は何通りかあります。SDC の文法を熟知しているユーザは テキスト・エディタで SDC ファイルを手動にて作成することもできますが、この資料では主に Quartus® Prime の TimeQuest Timing Analyzer を使用して SDC ファイルを作成する方法について説明します。

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。

本資料で使用しているツールおよびツール・バージョンは、以下の通りです。

- インテル® Quartus® Prime 開発ソフトウェア Standard Edition 17.1

この資料では、下記デザインを例に説明しています。(一部のコマンドでは、このデザイン例を使用していない箇所があります。)

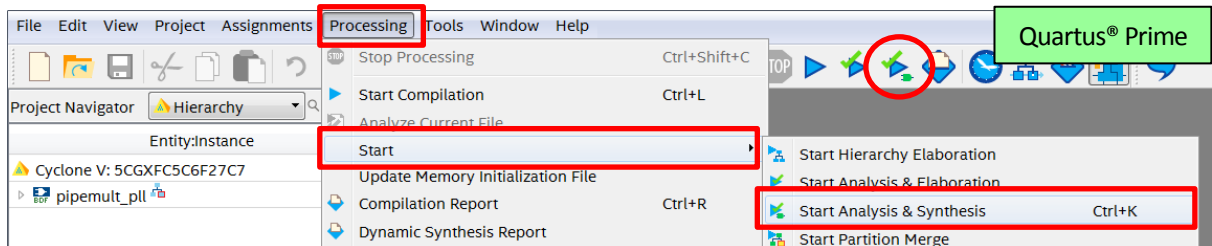


2. SDC ファイルの作成方法

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。

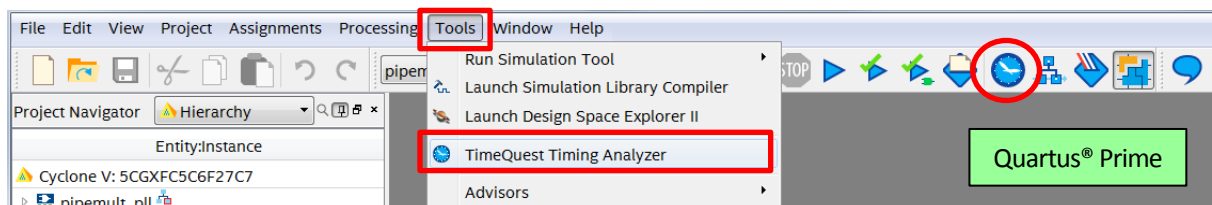
2-1. Analysis & Synthesis (論理合成) の実行

Quartus® Prime の Processing メニュー ⇒ Start ⇒ Start Analysis & Synthesis または ツールバーで 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。Fitter (配置配線) まで実行しても構いませんが、規模の大きなデザインだと時間が掛かるので、最低でも Analysis & Synthesis まで実行してください。



2-2. TimeQuest Timing Analyzer を使って SDC ファイルを作成

1. Quartus® Prime の Tools メニュー ⇒ TimeQuest Timing Analyzer または ツールバーで TimeQuest Timing Analyzer を起動します。



2. 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® Prime 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。

- **Analysis & Synthesis まで実行している場合**

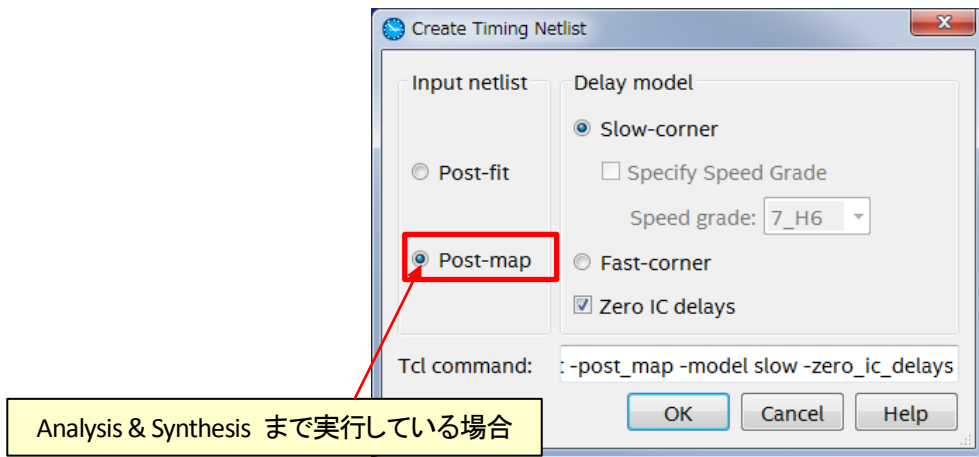
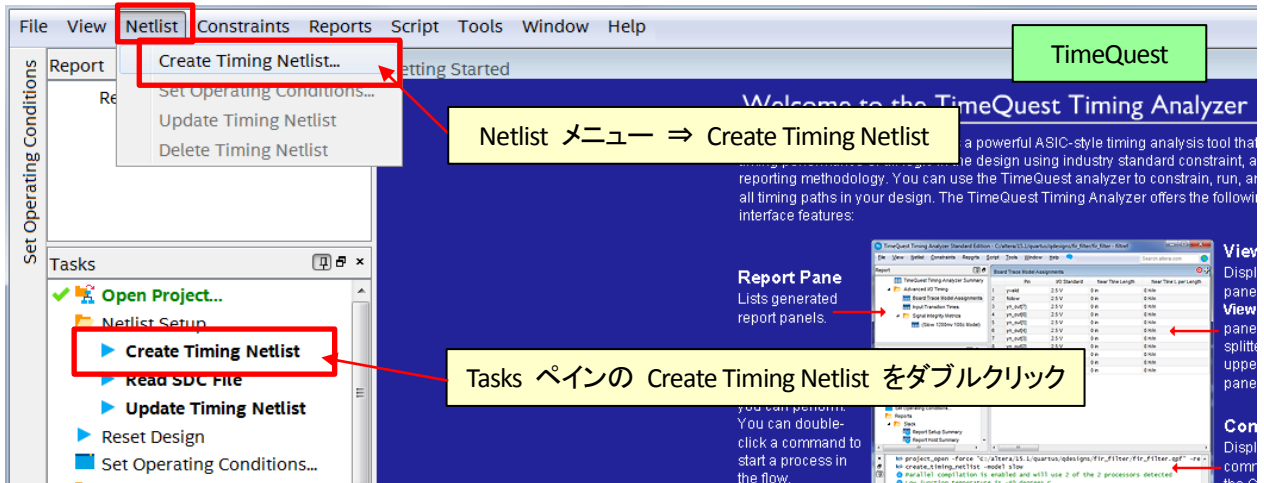
Netlist メニュー ⇒ Create Timing Netlist を実行後、Input netlist で Post-map を選択して OK

- **Fitter まで実行している場合**

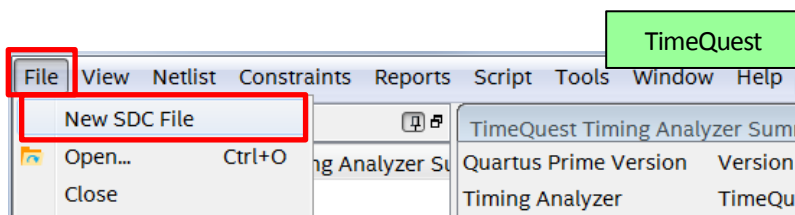
Netlist メニュー ⇒ Create Timing Netlist を実行後、Input netlist で Post-fit を選択して OK

または

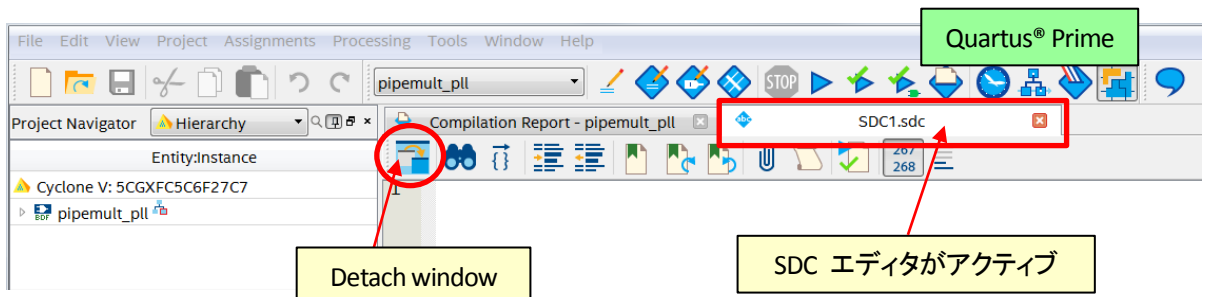
Task ペインの Create Timing Netlist をダブルクリック



- TimeQuest の File メニュー ⇒ New SDC File で SDC エディタを起動します。環境によって SDC エディタが別ウィンドウで起動したり、Quartus® Prime のトップ・ビュー・ウィンドウに起動したりします。別ウィンドウにしたい場合は、Windows® メニュー ⇒ Detach window を選択、または ツールバーをクリックします

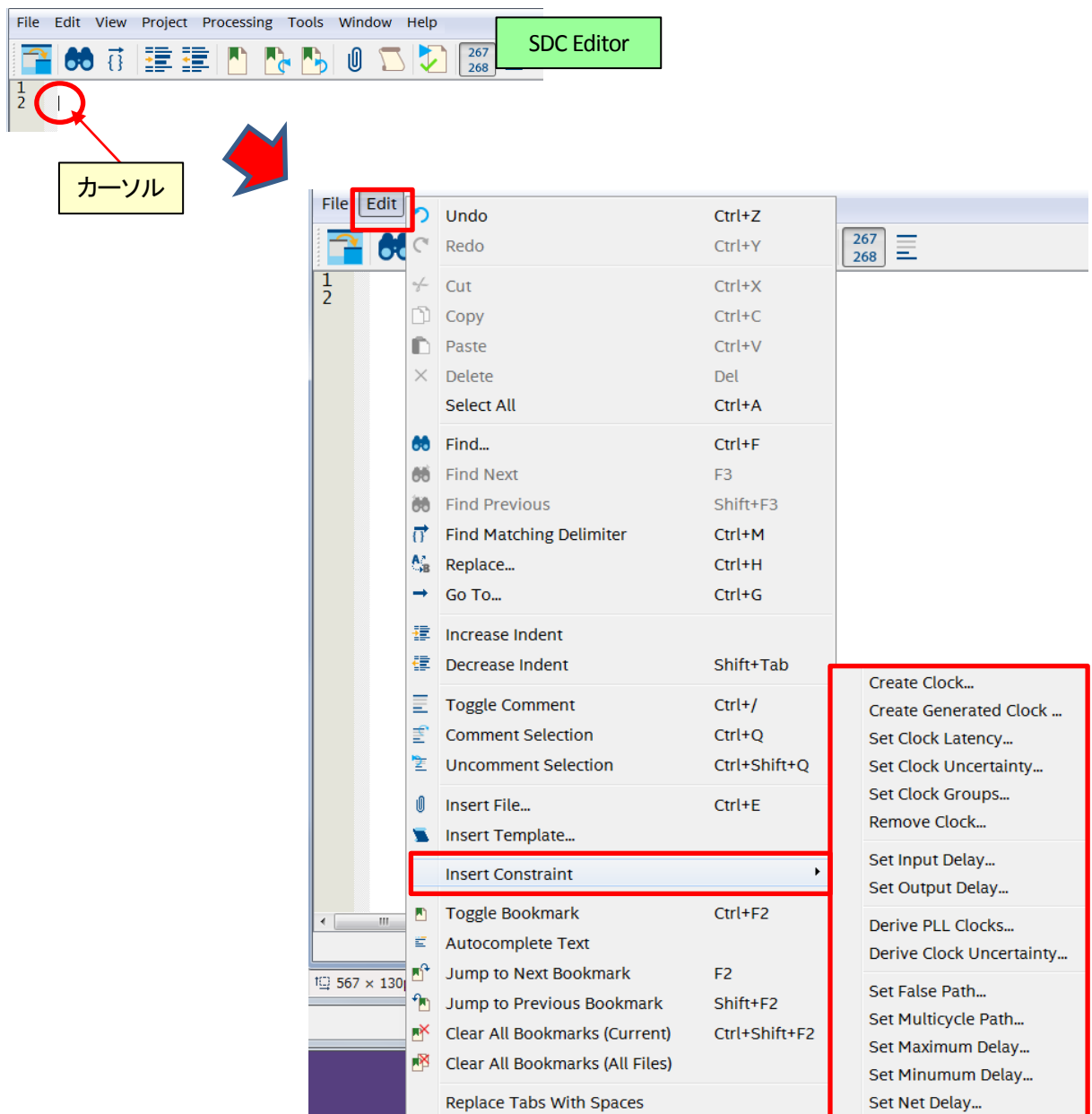


Quartus® Prime のトップ・ビュー・ウィンドウに起動した場合



- 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しておくことをお勧めします。(拡張子は、.sdc です。)

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ていきます。

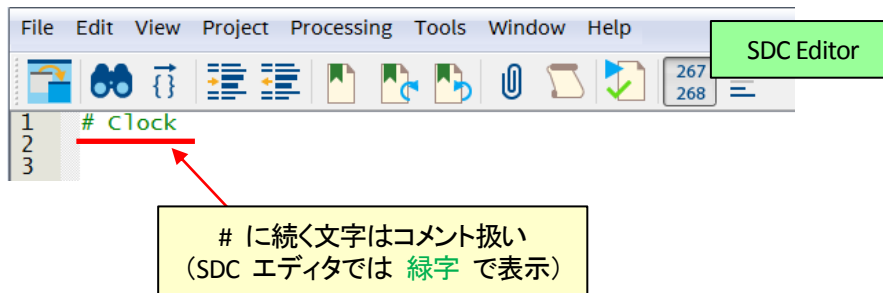


2-2-1. クロックの制約

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。

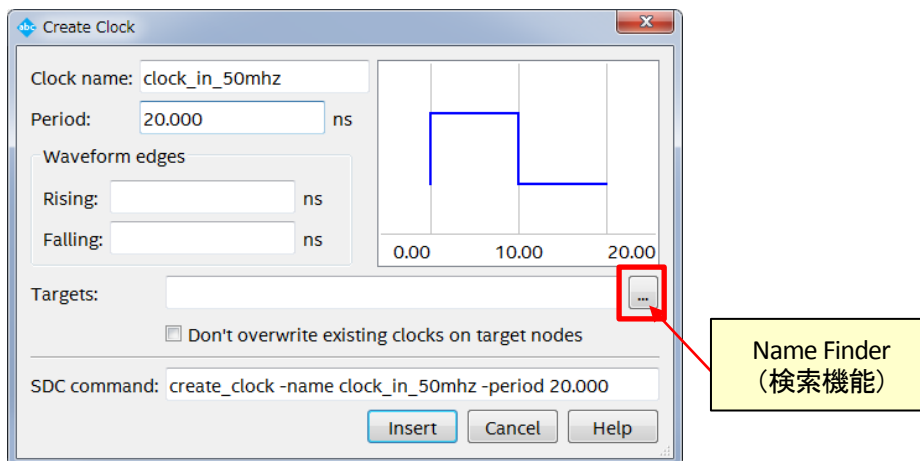
ただし、PLL で生成したクロックを自動的に制約してくれる便利なコマンドも用意されています。このコマンドを使用する場合は、生成クロック用のコマンドを使用する必要はありません。

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® Prime 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。



Clock name

TimeQuest や SDC 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® Prime 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。

Targets

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。TimeQuest 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® Prime 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	最上位階層の入力と出力(例:デバイス・ピン)

※ 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	デザインの入出力ピン(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

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*)

Options
 Case-insensitive
 Hierarchical
 Compatibility mode
 No duplicates

Case-insensitive: Filter 欄に入力したキーワードは、大文字と小文字を区別しない
 Hierarchical: Filter 欄に入力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
 Compatibility mode: 常に全階層を検索する

Matches
 List
 46 matches found
 1 selected name
 clock_in

リ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

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。

SDC command: [get_ports {clock_in}]

OK Cancel Help

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。

Clock name: clock_in_50mhz
 Period: 20.000 ns
 Waveform edges
 Rising: ns
 Falling: ns
 Targets: [get_ports {clock_in}]
 Don't overwrite existing clocks on target nodes

SDC command: me clock_in_50mhz -period 20.000 [get_ports {clock_in}]

この表示が SDC エディタに記載される。
 (手動修正可能)

Insert Cancel Help

File Edit View Project Processing Tools Window Help

SDC Editor

```

1 # clock
2 create_clock -name clock_in_50mhz -period 20.000 [get_ports {clock_in}]
3
    
```

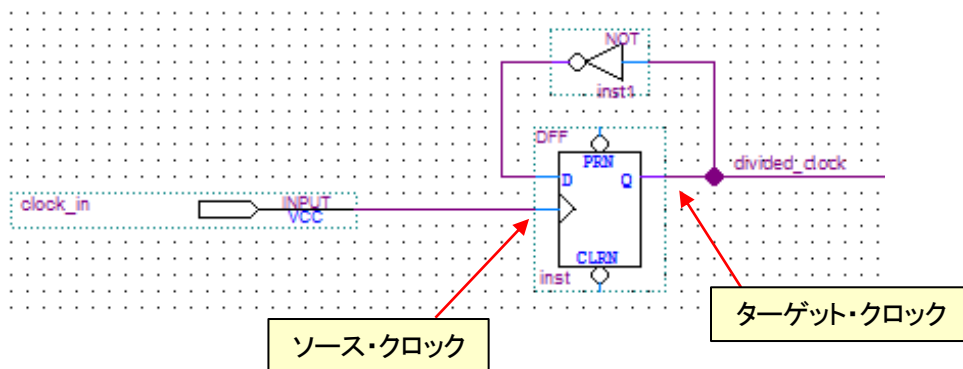
基本クロック(Base Clock)が複数ある場合は、この作業を繰り返します。

■ **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。)

なお、PLL で生成されたクロックは、後述の PLL 生成クロックの自動制約が簡単で便利です。もちろん、PLL 生成クロックも、このコマンドを使ってタイミング制約しても構いません。

このようなレジスタを使った分周クロックの場合、以下のように入力します。



Clock name

TimeQuest や SDC 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

ソース・クロックに対する分周比を指定します。

Multiply by

ソース・クロックに対する逡倍比を指定します。

Duty Cycle

デューティ比を指定します。単位は % です。50%(前半:High/後半:Low)のデューティ比の場合は、省略できます。

Phase

ソース・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

ソース・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。

Insert waveform

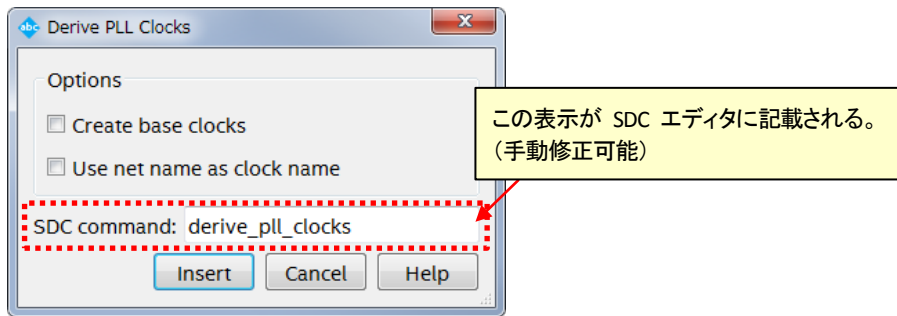
指定した条件の論理反転したクロックを制約したい時にチェックします。

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。このコマンドは、インテル FPGA の独自コマンドなので、サード・パーティのタイミング解析ツールでは認識されない可能性があります。



Create base clocks

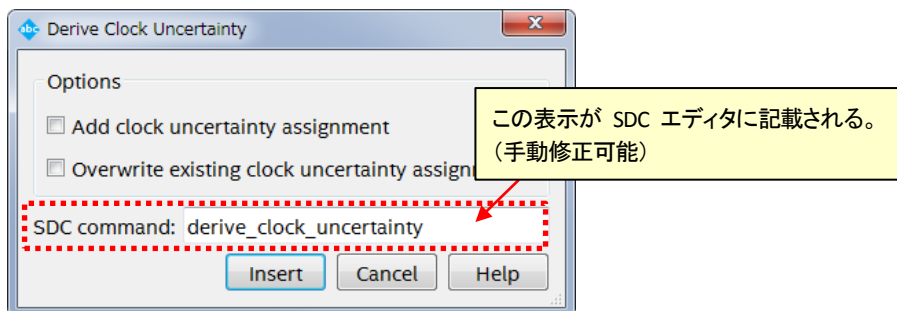
基本クロック(Base Clock)も同時に制約してくれます。ただし、基本クロックは Create Clock コマンドを使用して自分で制約することをお勧めします。

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 Clock Uncertainty を選択すると、Derive Clock Uncertainty 用の設定ウィンドウが表示されます。このコマンドは、インテル FPGA の独自コマンドなので、サード・パーティのタイミング解析ツールでは認識されない可能性があります。



Add clock uncertainty assignment

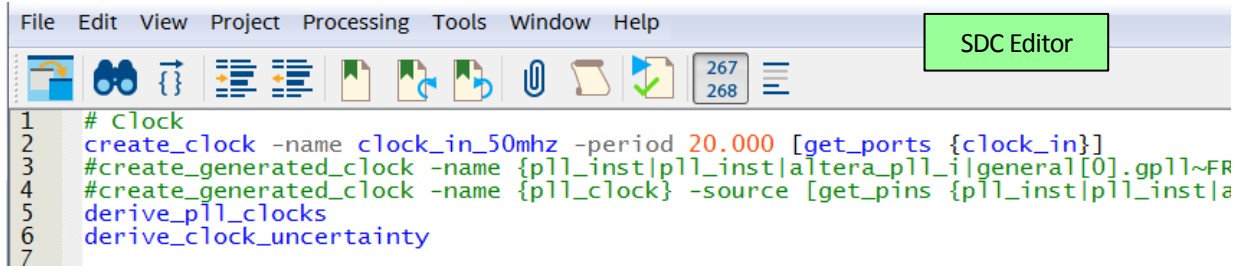
Set Clock Uncertainty コマンドで既にクロックのばらつきの制約をしている場合は、Set Clock Uncertainty コマンドの制約値と Derive Clock Uncertainty コマンドの制約値が合算されます。

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。)

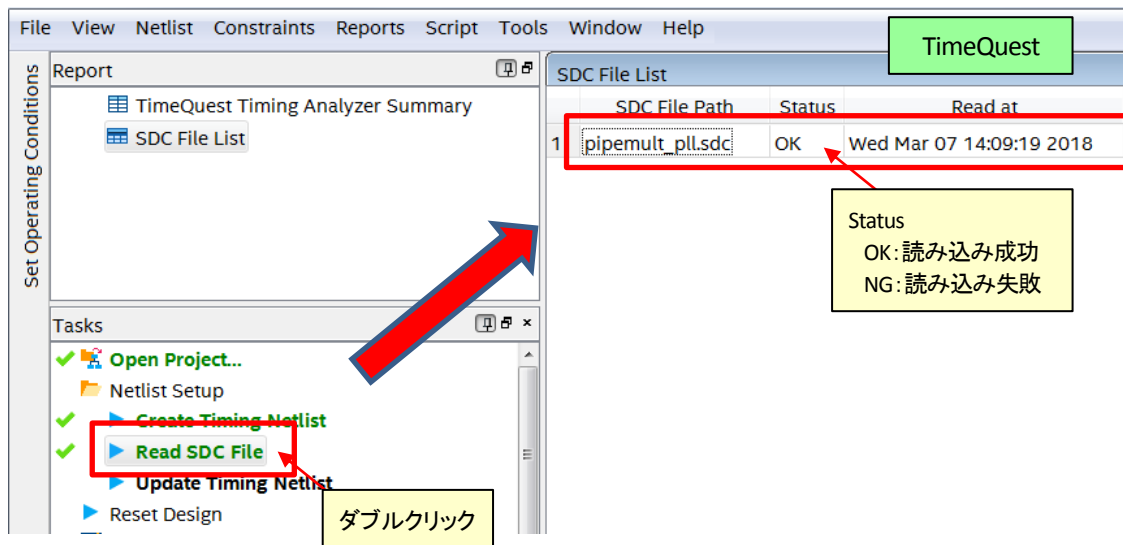
<例>



```

1 # clock
2 create_clock -name clock_in_50mhz -period 20.000 [get_ports {clock_in}]
3 #create_generated_clock -name {pll_inst|pll_inst|altera_pll_i|general[0].gp11~FR
4 #create_generated_clock -name {pll_clock} -source [get_pins {pll_inst|pll_inst|a
5 derive_pll_clocks
6 derive_clock_uncertainty
7
    
```

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 SDC ファイルの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。



The screenshot shows the TimeQuest interface. In the 'Tasks' pane, 'Read SDC File' is highlighted with a red box and a red arrow pointing to it, with a label 'ダブルクリック' (Double-click) below it. In the 'SDC File List' table, the first row is highlighted with a red box:

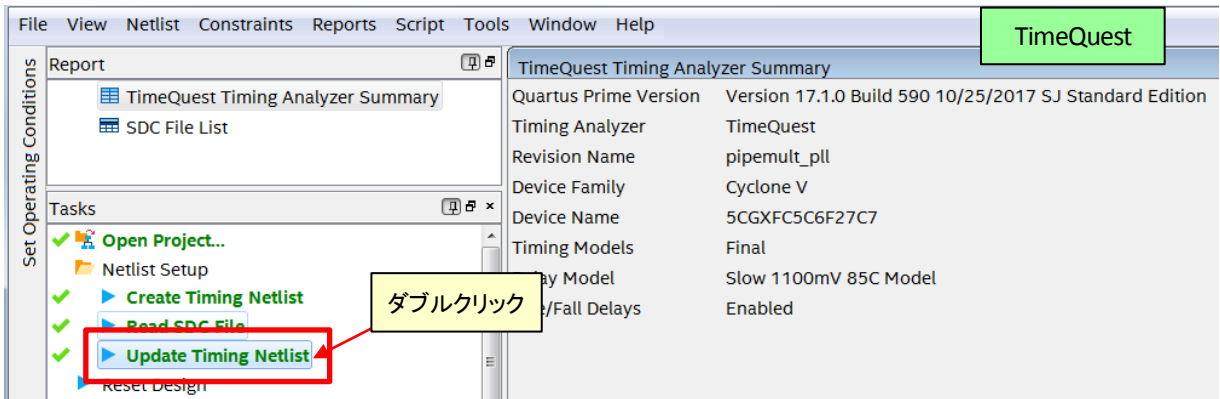
SDC File Path	Status	Read at
pipemult_pll.sdc	OK	Wed Mar 07 14:09:19 2018

A yellow callout box next to the 'Status' column contains the following text:

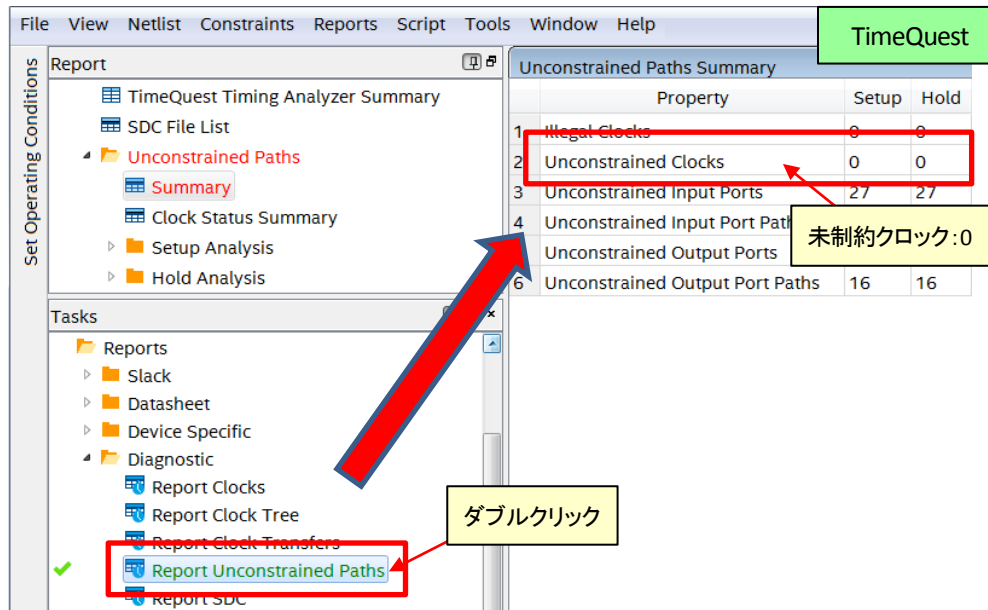
```

Status
OK:読み込み成功
NG:読み込み失敗
    
```

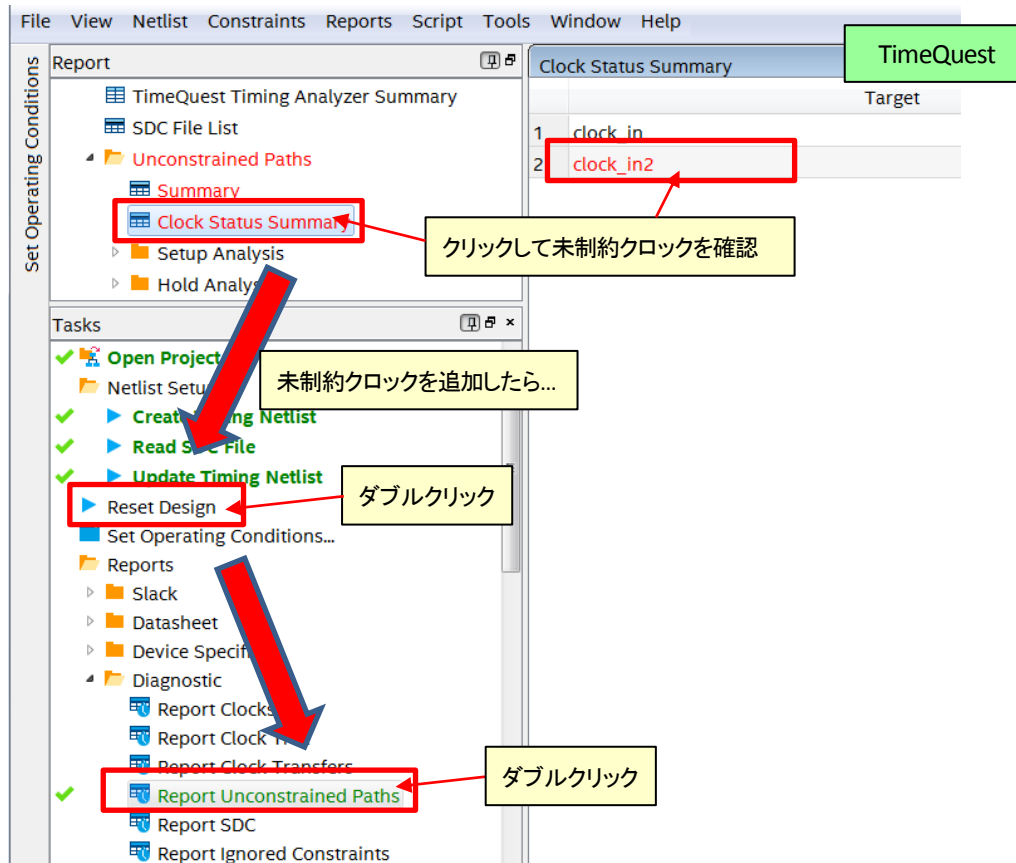
次に Tasks ペインの Update Timing Netlist をダブルクリックして、タイミング解析用ネットリストにタイミング制約をアップデートします。



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 ⇒ 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 を実行して、未制約クロックを確認してください。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。

SDC Command	Name	Period	Waveform	Targets
1 create_clock	clock_in_50mhz	20.000	{0.000 10.000}	[get_ports {clock_i}]

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 1,000MHz (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。(仕様で、未制約の基本クロックは 1,000MHz (1GHz) で自動制約してしまいます。)

	Clock Name	Type	Period	Frequency	Rise	Fall	Duty Cycle
1	clock_in_50mhz	Base	20.000	50.0 MHz	0.000	10.000	
2	pll_inst pll_inst alt...ACTIONAL_PLL vcoph[0]	Generated	3.333	300.03 MHz	0.000	1.666	50.00
3	pll_inst pll_inst al...UTPUT_COUNTER divclk	Generated	10.000	100.0 MHz	0.000	5.000	50.00

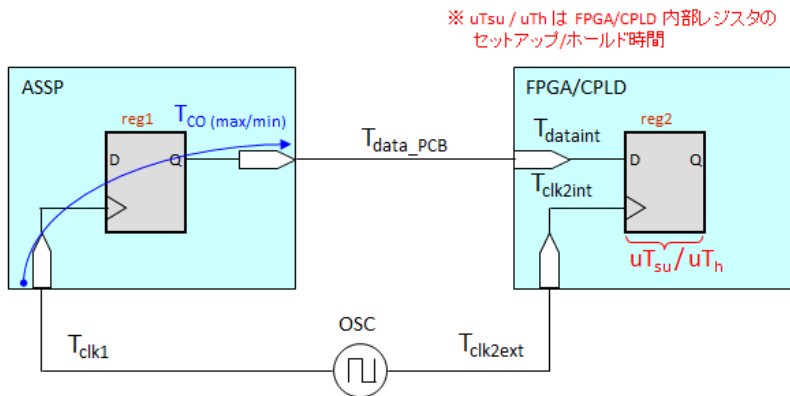
2-2-2. I/O の制約

クロックの制約の次は、I/O の制約を行きましょう。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、下記資料をご覧ください。

『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。

$$\text{Data Arrival Time (Max)} = \text{Tclk1} + \text{Tco(Max)} + \text{Tdata_PCB}$$

$$\text{Data Required Time (Setup)} = (\text{Tclk2ext} + \text{Tclk2int}) - \text{uTsu}$$

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。

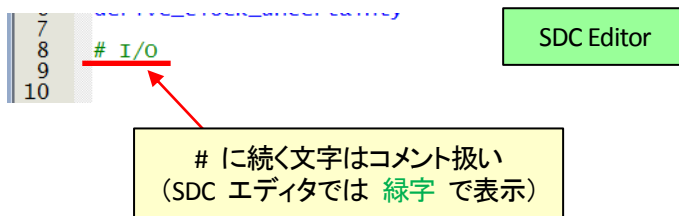
$$\text{Setup Slack} = \text{Data Required Time (Setup)} - \text{Data Arrival Time (Max)}$$

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。

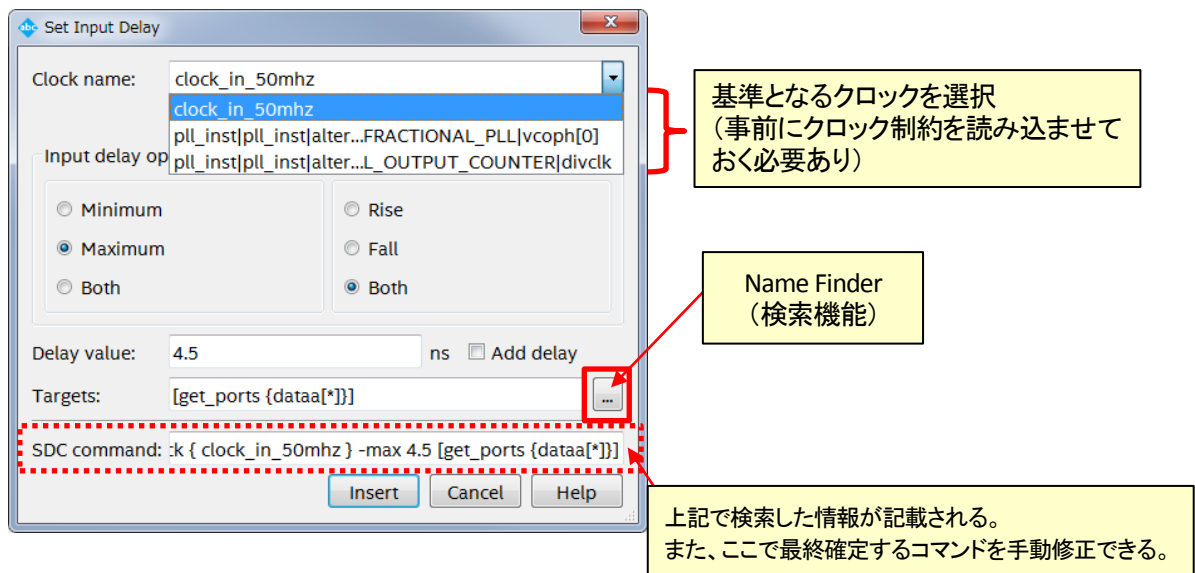
入力セットアップ時間は、set_input_delay コマンドの -max オプションを使用して制約します。制約の値は、以下の式で算出します。なお、ここではあまり話を難しくしないために、ボード上の伝搬遅延(Tdata_PCB と Tclk1、Tclk2ext)は一定とします。

$$\begin{aligned} \text{最大入力遅延} &= \text{ボード上の遅延} - \text{ボード上のクロック・スキュー} + T_{co}(\text{max}) \\ &= T_{\text{data_PCB}} - (T_{\text{clk2ext}} - T_{\text{clk1}}) + T_{co}(\text{max}) \quad \dots \text{式①} \end{aligned}$$

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。



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が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。

Targets

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。

$$\text{Data Arrival Time (Min)} = \text{Tclk1} + \text{Tco(min)} + \text{Tdata_PCB}$$

$$\text{Data Required Time (Hold)} = (\text{Tclk2ext} + \text{Tclk2int}) + \text{uTh}$$

Data Arrival Time 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。

$$\text{Hold Slack} = \text{Data Arrival Time (Min)} - \text{Data Required Time (Hold)}$$

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、ここでは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。

$$\begin{aligned} \text{最小入力遅延} &= \text{ボード上の遅延} - \text{ボード上のクロック・スキュー} + \text{Tco(min)} \\ &= \text{Tdata_PCB} - (\text{Tclk2ext} - \text{Tclk1}) + \text{Tco(min)} \quad \dots \text{式②} \end{aligned}$$

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が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。

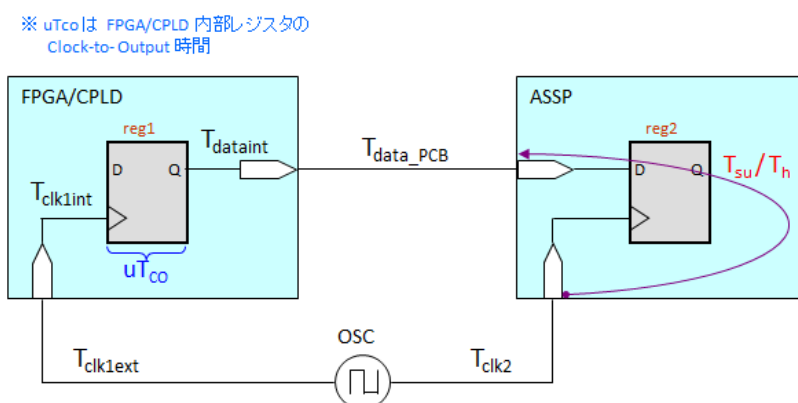
Targets

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。



<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 と Data Required Time (Setup) は、以下のようになります。

$$\text{Data Arrival Time (Max)} = (\text{Tclk1ext} + \text{Tclk1int}) + \text{uTco(max)} + (\text{Tdataint} + \text{Tdata_PCB})$$

$$\text{Data Required Time (Setup)} = \text{Tclk2} - \text{Tsu}$$

Data Arrival Time 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。

$$\text{Setup Slack} = \text{Data Required Time (Setup)} - \text{Data Arrival Time (Max)}$$

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、ここでは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。

$$\begin{aligned} \text{最大出力遅延} &= \text{ボード上の遅延} - \text{ボード上のクロック・スキュー} + T_{su} \\ &= T_{data_PCB} - (T_{clk2} - T_{clk1ext}) + T_{su} \quad \text{…式③} \end{aligned}$$

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。

Input 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。

Targets

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。

<出力最小 Clock-to-Output 時間の制約>

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

$$\text{Data Arrival Time (Min)} = (T_{clk1ext} + T_{clk1int}) + uT_{co(min)} + (T_{dataint} + T_{data_PCB})$$

$$\text{Data Required Time (Hold)} = T_{clk2} + T_h$$

Data Arrival Time 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。

$$\text{Hold Slack} = \text{Data Arrival Time (Min)} - \text{Data Required Time (Hold)}$$

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、ここでは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。

$$\begin{aligned} \text{最小出力遅延} &= \text{ボード上の遅延} - \text{ボード上のクロック・スキュー} - Th \\ &= Tdata_PCB - (Tclk2 - Tclk1ext) - Th \dots \text{式④} \end{aligned}$$

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がリ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

最小出力遅延を制約する場合は、Mimumum 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。

Targets

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。

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>

```

1 # clock
2 create_clock -name clock_in_50mhz -period 20.000 [get_ports {clock_in}]
3 #create_generated_clock -name {pll_inst|pll_inst|altera_pll_i|general[0].gp11~FRACTIONAL_PLL|vcoph[0]}
4 #create_generated_clock -name {pll_clock} -source [get_pins {pll_inst|pll_inst|altera_pll_i|general[0].
5 derive_pll_clocks
6 derive_clock_uncertainty
7
8 # I/O
9 set_input_delay -clock { clock_in_50mhz } -max 4.5 [get_ports data*]
10 set_input_delay -clock { clock_in_50mhz } -min 2.5 [get_ports data*]
11 set_input_delay -clock { clock_in_50mhz } -max 4.5 [get_ports {rdaddress* wraddress*}]
12 set_input_delay -clock { clock_in_50mhz } -min 2.5 [get_ports {rdaddress* wraddress*}]
13 set_input_delay -clock { clock_in_50mhz } -max 4.5 [get_ports {wren}]
14 set_input_delay -clock { clock_in_50mhz } -min 2.5 [get_ports {wren}]
15 set_output_delay -clock { pll_inst|pll_inst|altera_pll_i|general[0].gp11~PLL_OUTPUT_COUNTER|divclk }
16 -max 1.5 [get_ports q*]
17 set_output_delay -clock { pll_inst|pll_inst|altera_pll_i|general[0].gp11~PLL_OUTPUT_COUNTER|divclk }
18 -min 0.5 [get_ports q*]
19
    
```

I/O の制約ができたなら、クロックの制約のところでも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。

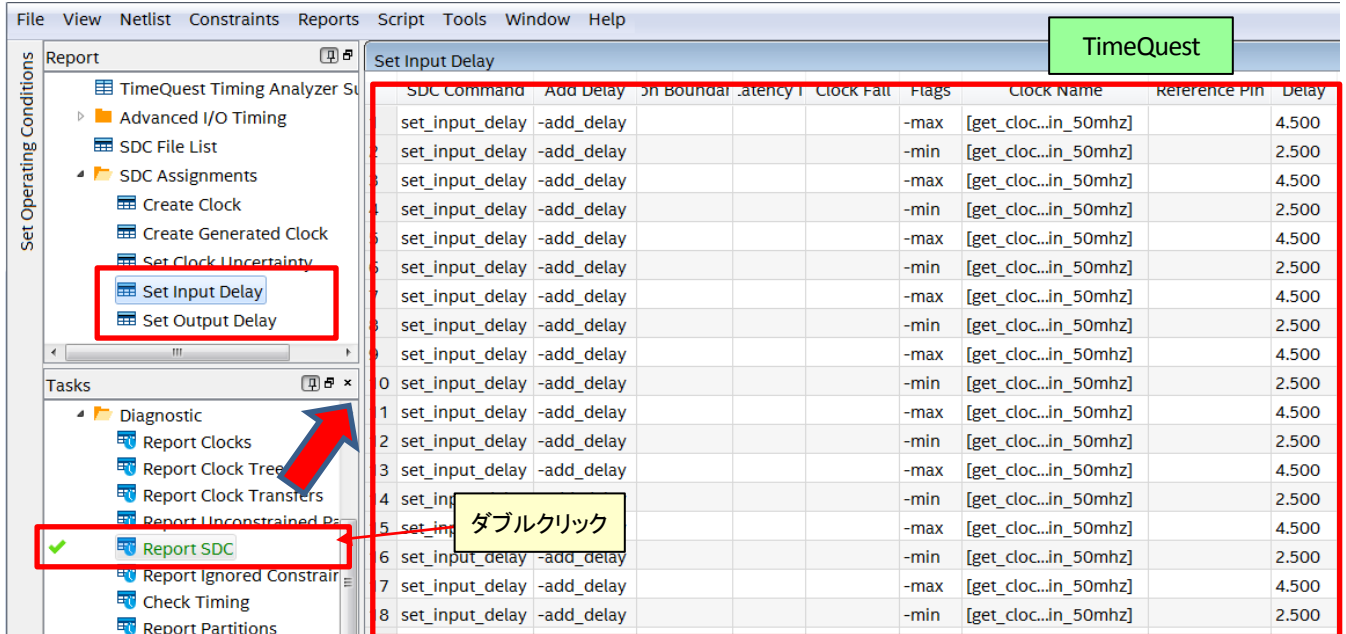
ここまでできたら、次に未制約ポートの有無を確認するために、Tasks ペインの Diagnostic ⇒ Report Unconstrained Paths をダブルクリックします。Unconstrained Input Port や Unconstrained Output Port が 0 ならば、未制約 I/O がないことを意味します。

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。)

※ **クロックのところでも説明した内容と同じです。**

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。



The screenshot shows the Quartus Prime interface with the TimeQuest report window open. The 'Tasks' pane on the left shows 'Report SDC' selected and highlighted with a red box. A red arrow points to the 'Report SDC' task with the text 'ダブルクリック' (Double-click). The main report window displays a table of SDC commands for 'Set Input Delay'.

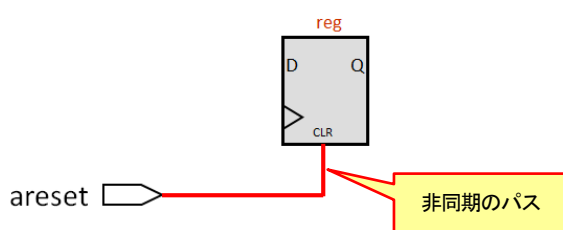
SDC Command	Add Delay	In Boundar	atency I	Clock Fall	Flags	Clock Name	Reference Pin	Delay
set_input_delay	-add_delay				-max	[get_cloc...in_50mhz]		4.500
set_input_delay	-add_delay				-min	[get_cloc...in_50mhz]		2.500
set_input_delay	-add_delay				-max	[get_cloc...in_50mhz]		4.500
set_input_delay	-add_delay				-min	[get_cloc...in_50mhz]		2.500
set_input_delay	-add_delay				-max	[get_cloc...in_50mhz]		4.500
set_input_delay	-add_delay				-min	[get_cloc...in_50mhz]		2.500
set_input_delay	-add_delay				-max	[get_cloc...in_50mhz]		4.500
set_input_delay	-add_delay				-min	[get_cloc...in_50mhz]		2.500
set_input_delay	-add_delay				-max	[get_cloc...in_50mhz]		4.500
set_input_delay	-add_delay				-min	[get_cloc...in_50mhz]		2.500
set_input_delay	-add_delay				-max	[get_cloc...in_50mhz]		4.500
set_input_delay	-add_delay				-min	[get_cloc...in_50mhz]		2.500
set_input_delay	-add_delay				-max	[get_cloc...in_50mhz]		4.500
set_input_delay	-add_delay				-min	[get_cloc...in_50mhz]		2.500
set_input_delay	-add_delay				-max	[get_cloc...in_50mhz]		4.500
set_input_delay	-add_delay				-min	[get_cloc...in_50mhz]		2.500
set_input_delay	-add_delay				-max	[get_cloc...in_50mhz]		4.500
set_input_delay	-add_delay				-min	[get_cloc...in_50mhz]		2.500
set_input_delay	-add_delay				-max	[get_cloc...in_50mhz]		4.500
set_input_delay	-add_delay				-min	[get_cloc...in_50mhz]		2.500

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。

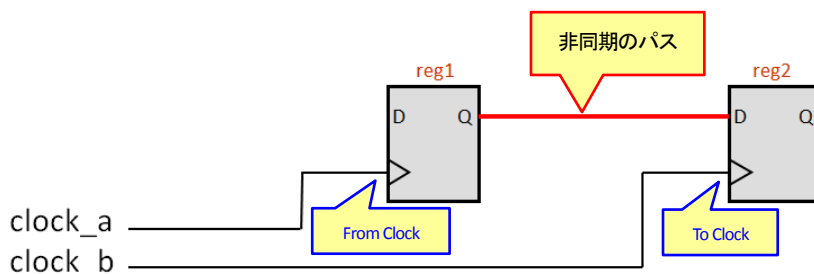
下図のような非同期の入力ポート／出力ポートのパスや異なるクロック・ドメイン間の乗せ換えのパス、クロック・マルチプレクサは、フォルス・パスに制約できる代表例になります。

<例 1: 非同期の入力ポート>



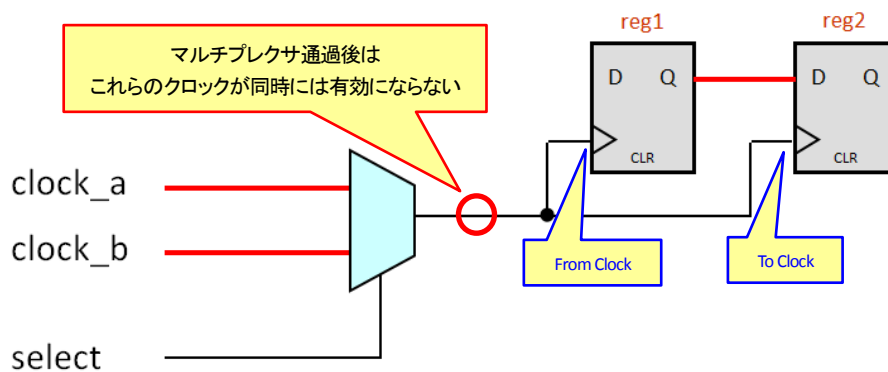
例1のケースは、areset 信号の生成元が完全に非同期である場合、set_false_path コマンドで制約できます。フォルス・パスの制約を掛けない場合は、Recovery/Removal の解析対象となります。(Recovery/Removal については、本資料では説明を省略しています。)

<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くことになります。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	pll altpll_component auto_generated pll1 clk[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	

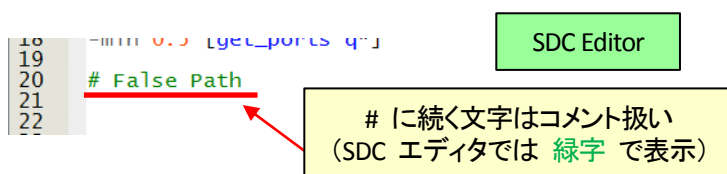
RR Paths : 送信レジスタ(reg1)が立ち上がりエッジで、受信レジスタ(reg2)が立ち上がりエッジのパス数

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

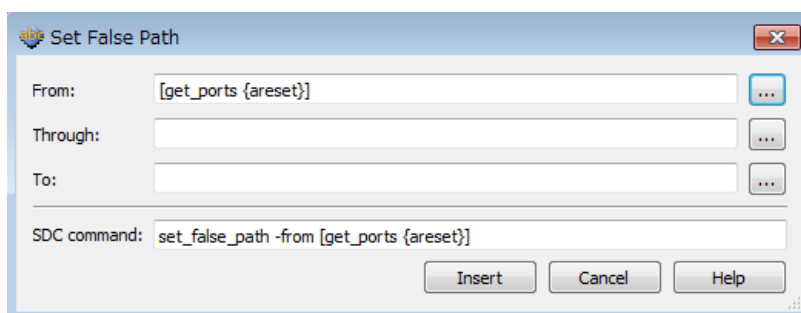
フォルス・パス制約するパスが決まったら、SDC エディタに制約を記述してみましょう。クロックや I/O の時と同じように、コメント行を入れておきましょう。



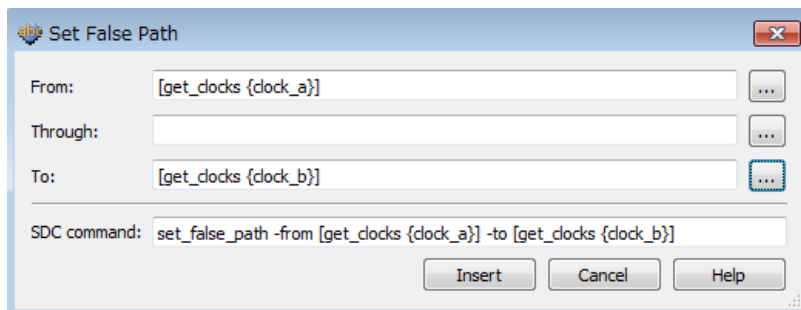
■ 特定のノードに指定するフォルス・パスの制約 <コマンド:set_false_path>

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。)

<①:ポイントで指定>



<②:クロックで指定>



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

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。

To

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。

■ 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>

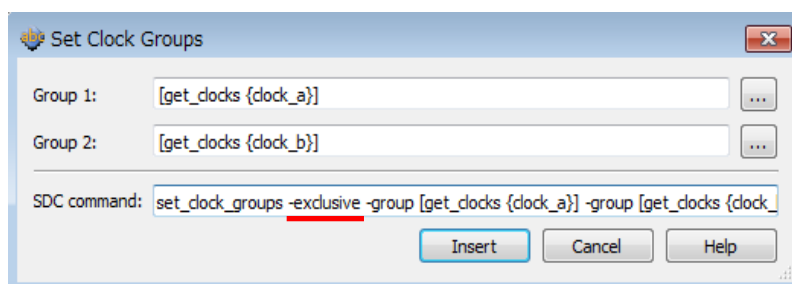
SDC エディタでコマンドを挿入したい行にカーソルを合せた状態で Edit メニュー ⇒ Insert Constraint ⇒ Set Clock Groups を選択すると、設定ウィンドウが表示されます。(良く使用する設定内容のみ説明します。)

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。

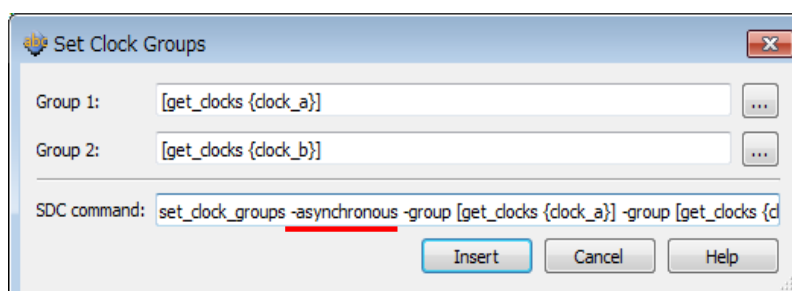
<①:-exclusive オプションで指定>

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。



<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。



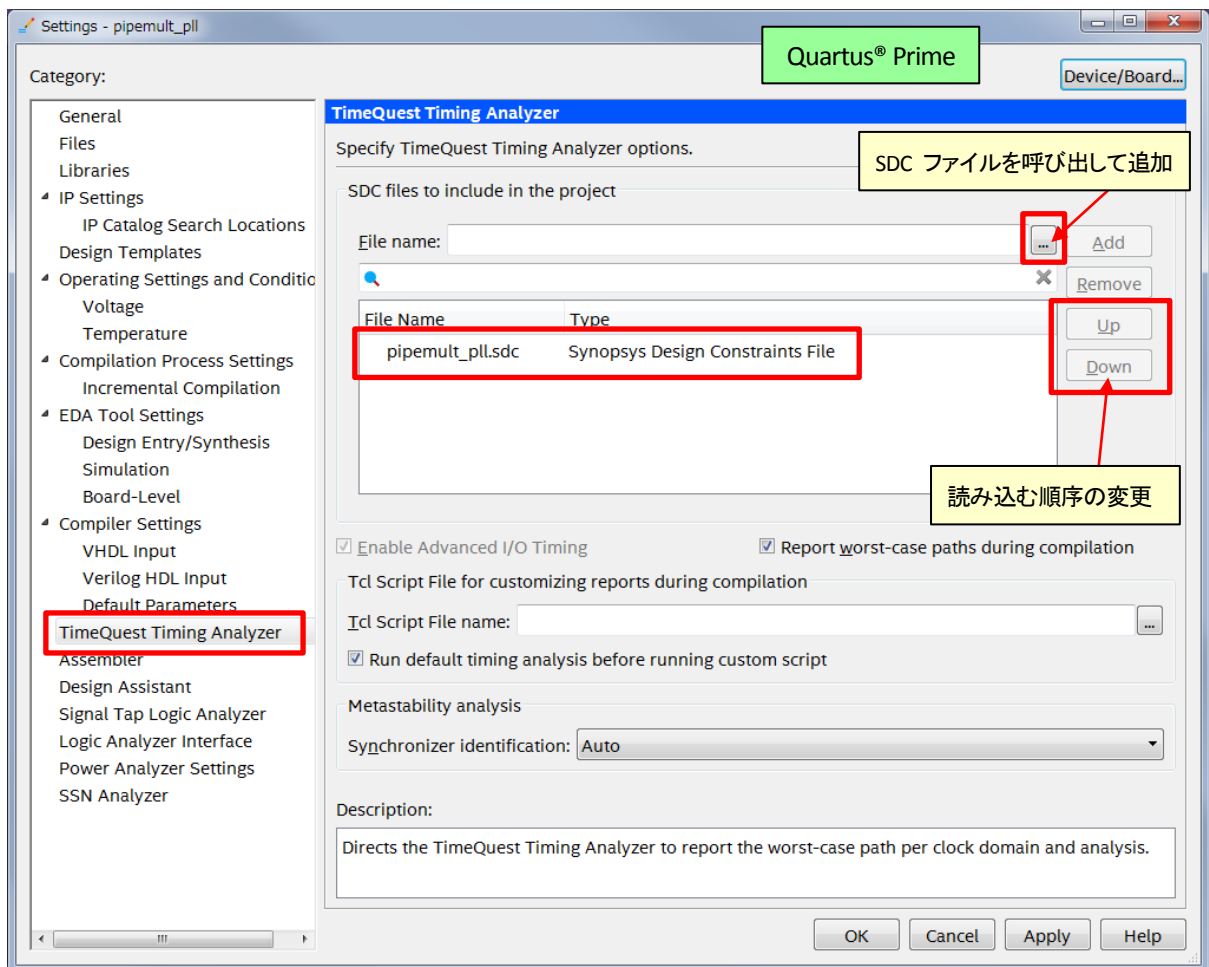
ここまでで説明したコマンドの中には、説明していないオプションが存在しているものがあります。また、非同期パスの制約やマルチ・サイクル・パスの制約などもありますが、本資料は初心者向けの資料であるため、これらは割愛しています。

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® Prime 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、Quartus® Prime の Assignments メニュー ⇒ Settings ⇒ TimeQuest Timing Analyzer で SDC ファイルが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。

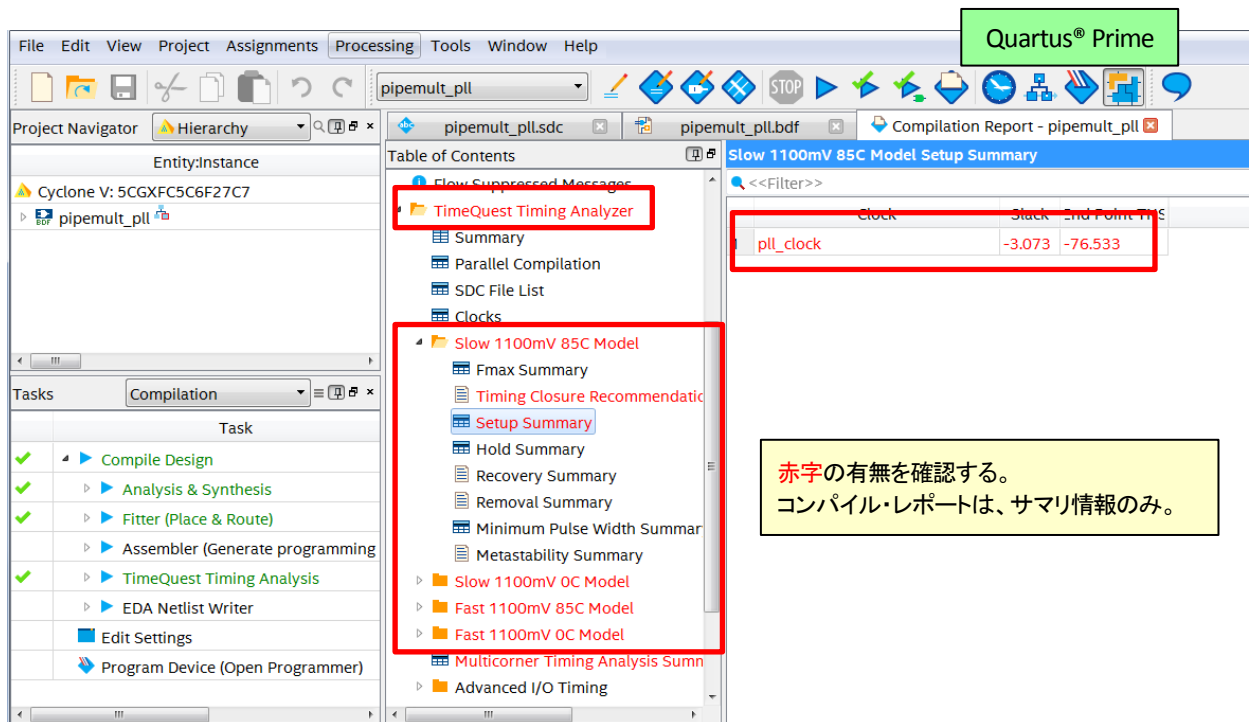


3-2. コンパイルの実行

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。



The screenshot shows the Quartus Prime interface. The 'Table of Contents' pane on the left has two red boxes highlighting the 'TimeQuest Timing Analyzer' folder and the 'Slow 1100mV 85C Model' folder. The 'Slow 1100mV 85C Model Setup Summary' pane on the right shows a table with a red entry for 'pll_clock'.

Clock	Slack	End Point	Units
pll_clock	-3.073	-76.533	

A yellow box contains the text: 赤字の有無を確認する。コンパイル・レポートは、サマリー情報のみ。

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下記資料をご覧ください。

『Quartus® Prime はじめてガイド - TimeQuest によるタイミング解析の方法』

改版履歴

Revision	年月	概要
1	2018 年 3 月	初版
2	2018 年 3 月	誤記訂正 P.22 最大入力遅延 → 最大出力遅延 P.23 最小入力遅延 → 最小出力遅延

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。
3. 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。

株式会社マクニカ アルティマ カンパニー <https://www.alt.macnica.co.jp/> 技術情報サイト アルティマ技術データベース <http://www.altima.jp/members/>

株式会社エルセナ <http://www.elsena.co.jp> 技術情報サイト ETS <https://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、各メーカー発行の英語版の資料もあわせてご利用ください。