

第 10 章 タイミング制約設定ガイド

本章では、Lattice Diamond でタイミング・クロージャを効果的かつ効率良く成功させるために理解しておくべき、設計制約の考え方やツールの振る舞い、またタイミング検証を行う際に注意すべき点など、ガイドラインと共に記述します。

なお、具体的なタイミング制約の設定方法については、第 8 章「スプレッドシート・ビュー」の第 4 節をご参照ください。

10.1 設計制約

本節ではまず、タイミング・クロージャを達成するために与える設計制約 (Preference) の Lattice Diamond での扱いについて記載します。

10.1.1 基本的な設計制約の考え方

どのようなインプリメンテーションにおいても、適切な設計制約を与えることは期待する動作を実現するために重要です。考慮すべき制約事項と指定は、大きく分類して以下があります。

- ・ ピンに対する設計制約
- ・ タイミング制約
- ・ その他の設計制約

まず最初にピンに関する制約が適切であることを確実にします。そうでないと、内部タイミングが如何に満足されていようと本来達成すべきインプリメンテーションにならない可能性があります。IO タイミングの制約がフィッティング作業の後工程で追加・変更された場合、内部配置配線が大幅に異なる結果を引き起こす可能性があり、それまでに費やされた全ての処理に係わる時間が損失となります。

「その他の設計制約」は特定モジュールや IP などの実装に係わる配置指定を含む制約事項を意味します。特定の制約があるかどうかも含めて、付随するリファレンスガイドなどを参照します。

ネットやパス、或いはクロック信号に対して何らかのタイミングに係わる制約を与えた場合、配置配線プロセスは以下の内容を満たすように (指定回数分の) 繰り返し処理を実行します。

- ・ 最大動作速度 (周波数または周期)
- ・ 特定パス / ネットの最大遅延
- ・ レジスタ間のホールド時間
- ・ Set/Reset のリカバリ時間 (オプション設定によって解析範囲が変わります)

配置配線処理とタイミング解析 (Place & Route Trace) の両エンジンは連携しており、処理終了後に解析結果をレポートに書き出します。従って、タイミング制約を変更した際には、タイミング・データベースの参照する条件が変わるため、Lattice Diamond は配置配線処理から実行し直します。ターゲットデバイスのスピードグレードを変更した場合も同様です。

一方 I/O タイミング解析 (I/O Timing Analysis) プロセスはこれと異なり、配置配線処理後のネットリスト (.ncd) に対して、スピードグレードや PVT 条件の組み合わせを変えて、タイミング解析のみを繰り返し実行してレポートアウトします。

10.1.2 設計制約のタイプ

設計制約を分類すると以下のようになります。

1. 設計制約を（新たに）適用

FREQUENCY / PERIOD, INPUT_SETUP, CLOCK_TO_OUT, MAXDELAY, CLOCKSKEWDIFF など

2. 設計制約を緩和、適用除外

BLOCK, MULTICYCLE;

以下の二つについてはデフォルトで適用する事を推奨します。プロジェクトを生成した際に自動生成される LPF ファイルには必ず含まれています。

BLOCK RESETPATHS;

BLOCK ASYNCPATH;

なお、FREQUENCY や PERIOD 制約で与える遅延量を緩和させる場合、特定の単一クロックドメイン内では MULTICYCLE を用い、クロックドメイン間にまたがるパスについては MAXDELAY を用いるようにします。

BLOCK 制約は上述のように特定制約を緩和・適用除外する効果がありますが、もう一つの作用・意図としては、何もしないと未制約パスと認識される場合はタイミングカバレッジが高くなりませんので、これらを明示的に BLOCK 指定することで制約パスとして Diamond に解釈させることができ、従ってカバレッジが改善されます。また配置配線エンジンが、パスに対してタイミングを満たす為の高い優先度で処理することを不要にする効果もあります。

ただし、BLOCK 制約を特定ネットに適用した場合、そのネット自体についてはタイミングレポートには含まれませんので注意が必要です。これは Diamond 3.1 時点で既知の問題です（修整バージョン：TBD）。

10.1.3 重複する設計制約の適用ルール

Lattice Diamond において、設計制約が重複して与えられている場合は以下のような優劣度が適用されます（以下 ”>” 印は「(優先度高い) > (優先度低い)」を意味します）。

1. グローバル制約と特定の制約

特定 > グローバル

LPF で記述されている重複した特定の制約は**後ろの記載**が有効

2. 特定パスに対する制約

BLOCK > MULTICYCLE > MAXDELAY > FREQUENCY

3. ポートとそのポートに繋がるネット（たとえば FREQUENCY 制約）

ネット > ポート

1 の例にはファンアウト制約があります。論理合成ツールで与えるグローバル設定（Synplify Pro: Fanout Guide、LSE: Max Fanout Limit）は、HDL 内で与える「synthesis syn_maxfan=N（N はファンアウト数）」などの記述で特定ネットやポートに対する制約を適用することが可能です。

10.1.4 設計制約のレビュー

与えられた設計制約が適切かどうかを、タイミングレポート（Trace Report: .tw1 / .twr）を読んで以下の観点から慎重にレビューすることは重要です。

- ・ レポートされる全制約事項の適用対象アイテム数（”xxx items scored”）を確認する。アイテム数がゼロの場合、誤りであったり意図する制約になっていないことを示唆する。またエラーアイテム数が多すぎる場合、制約自体が過度であるか、RTL 記述を修正しないと根本解決ができない可能性がある
- ・ タイミングエラーのパスについて BLOCK できるかどうか。リセットなど非同期パスではないか
- ・ 逆に BLOCK すべきではないパスが BLOCK 指定されていないかどうか

- ・ タイミングエラーのパスについて **MULTICYCLE** 指定できないか（単一クロックドメイン内）
- ・ 逆に **MULTICYCLE** すべきではないパスが **MULTICYCLE** 指定されていないかどうか
- ・ 当該パスは同期・依存関係にない二つのクロックドメイン間（CDC: Clock Domain Crossing）かどうか。その場合、設計的にタイミング制約がなくても問題ない（Don't Care）回路構成になっているか
- ・ タイミングカバレッジは ”十分に” 高いかどうか（90～95%以上）
- ・ PAR ストラテジで「Check Unconstrained Paths」を TRUE にし（デフォルトは FALSE）、制約の与えられていないパスとしてレポートされてる中で、本来は制約されるべきパスはないかどうか

10.1.5 タイミング違反解消のための対処方法

タイミング違反を解消するために、上述のような設計制約自体の妥当性を見直すことが基本ですが、その次に考慮すべき対処方法の例を示します（順不同）。

まずレポートの着眼点から

- ・ 複数シード（PAR Iteration Starting Point）指定など、ストラテジオプションの考慮
- ・ クロックネット割り当てがプライマリやセカンダリで適切か
- ・ メモリや sysDSP などの出力で遅延が大きい場合、その出力レジスタをイネーブルすること
- ・ パス詳細レポートから、特に遅延が大きくついているエレメント、或いは想定以上にファンアウトが大きい場合はファンアウト制限の指定
- ・ シフトレジスタは FF ベースか分散メモリか（推論の場合、論理合成結果の確認も）
- ・ IO タイミングの場合、入出力 FF を IO セル内 FF とするか、ロジックスライス内 FF とするか。後者の場合、配置指定することも考慮
- ・ 論理合成ツールに対する制約を与えて「その実装にとってより望ましい」EDIF を生成させる
- ・ フロアプランニング：[1] PLL マクロなどの位置指定、[2] UGROUP、BBOX、Anchor などによる（ユーザ）モジュールレベルの配置指定

また、HDL の修正が不可避、或いは最善と判断される場合

- ・ クロックイネーブルの使用（含ストラテジオプション）
- ・ メモリはブロックメモリか分散メモリか（インスタンスする場合）
- ・ FF 間にさらに FF を挿入できないか（パイプライン化）、或いは FF 段数を増やせないか
- ・ パイプライン FF 間の論理バランスの見直し（リタイミング化。ストラテジ・オプションではなく HDL 記述による方法を推奨）
- ・ 論理合成ツールの推論ベースではなく、IPexpress 生成マクロによるインスタンス化

10.2 クロックドメイン内の制約設定と解析

各クロックに周期（PERIOD）または周波数（FREQUENCY）制約を設定すれば、クロックドメイン内は勿論、可能な限りクロックドメインをまたぐパスでも自動的に適切な制約が設定されます。しかし、以下のようなクロックの構成の場合は、周期／周波数制約だけでは期待通りの制約が適用されないこともあるので注意してください。

レジスタ間で使用しているクロックエッジが異なるパスでは、基本的にはクロック周期の 1/2 が制約として適用されます。ただし、PERIOD 制約でデューティ比を設定していればその値が適用されます。

10.2.1 反転クロックを使用した場合の注意事項

Lattice Diamond のタイミング解析では、各スライス内で選択されているクロックの極性設定をみて適用される制約を計算します。

Lattice Diamond 日本語ユーザガイド

HDL ソースに反転クロックを記述した場合、ほとんどの場合においてエッジの反転はインプリした際にロジックスライス内のクロック反転機能で実現されます。このため反転／非反転クロックエッジの混在しているパスでも意図するタイミング解析が行われます。このようなインプリが行われた場合、エッジを問わずクロックネット名が共通になります（立下りエッジ用のクロックネットが生成されない）。

図 10-1. エッジの異なるクロック間のパス（スライスの反転機能を使用）

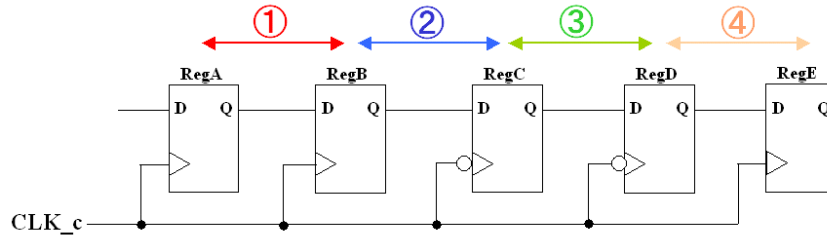


図 10-1 を例にとると、“RegC” および “RegD” のクロック反転はスライス内のクロック反転機能を使用して実現されるため、使用しているエッジが異なるパス（②④）では両エッジ間のタイミングが計算され制約として適用されます。

しかし、ゲートド・クロックで反転論理を実現した場合や、回路図入力でインバータを使用してクロックの反転論理を実現した場合は、インプリの際にスライス内のロジック反転機能を使用されず図 10-2 のように LUT を使用してクロックの反転が実現されます。このような場合、Lattice Diamond のタイミング解析ではゲーティングの内容（LUT 内の論理）を考慮せずに解析を行うので、期待通りのデューティ比とクロックエッジでの解析が行われません。

図 10-2. エッジの異なるレジスタ間のパス（LUT で反転）

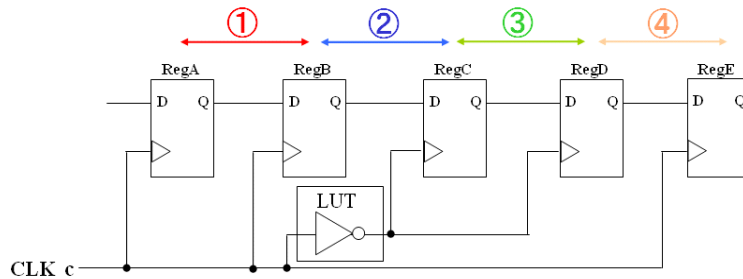
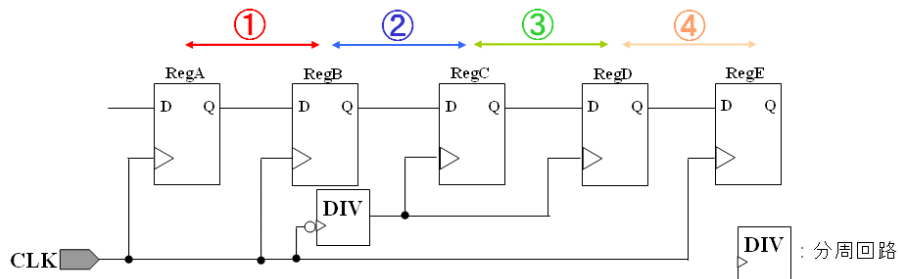


図 10-3. 反転クロックで分周クロックを生成した場合



また、反転クロックで分周クロックを生成した場合（図 10-3）においても、分周回路でのエッジを考慮しないタイミング解析が行われるため、期待している制約が適用されません。

これらのパスのように、クロック反転論理の実現がパスの始点・終点レジスタ（スライス内）で行われなない場合は、別途 MULTICYCLE 制約等で適切な制約を設定する必要があります。

10.2.2 PLL の位相シフト出力を使用した場合

PLL を使用した場合、PLL の各出力にはモジュール生成時に指定した周波数が制約として自動的に設定されます。ユーザは他のクロックと同様の方法で、各 PLL 出力に対して生成時の設定とは違う制約を設定することができますが、その場合には配置配線およびタイミング解析実行時に以下のようなメッセージが出力されます。

```
WARNING - par: PLL: 「PLL インスタンス名」 [ 出力ポート ] frequency [ 指定し直した制約 ]
doesn't match divider settings
```

このメッセージが出力されても、処理や解析は新たに設定した制約が有効になります。

タイミング解析では、位相シフトクロックを使用している場合の位相シフト分はスキューとして処理されます。この際、位相シフトしているクロックと位相シフトしていないクロック間のパスの解析内容が、ユーザの意図していないものになることがあります。

図 10-4. PLL の各出力間のパス

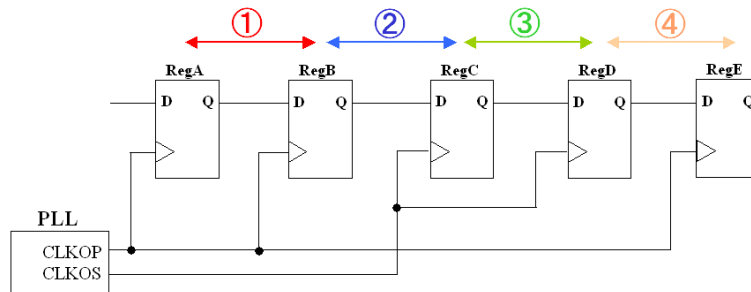
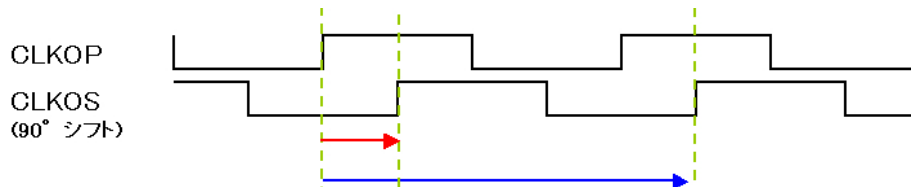


図 10-4 のような構成の回路で CLKOS を 90° シフト（遅延）させて使用した場合、デフォルト設定ではパス②に許容される遅延が図 10-5 のように [クロック 1 サイクル + 位相シフト分] となります。パスの始点のクロックが CLKOP、終点のクロックが CLKOS となるパスでは、位相シフト量にかかわらず必ずこのルールが適用されます。

以下のように MULTICYCLE 制約を設定することで、このクロック間の制約を図 10-6 のように位相シフト分だけにできます。

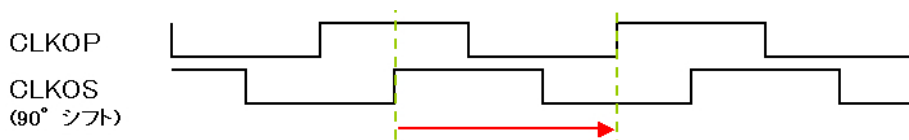
図 10-5. PLL の出力間パスの適用される制約 (1)



```
MULTICYCLE FROM CLKNET "CLKOP 出力名" TO CLKNET "CLKOS 出力名" 0 X ;
```

なお、図 10-4 ④のパスのように始点のクロックが CLKOS、終点のクロックが CLKOP となるパスでは、制約として図 10-6 のように [クロック 1 サイクル - 位相シフト分] が設定されます。

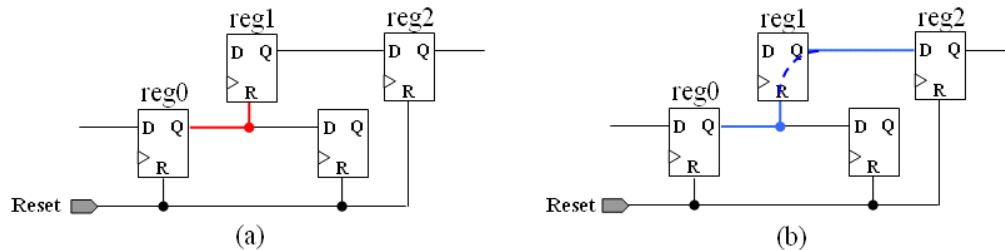
図 10-6. PLL の出力間パスの適用される制約 (2)



10.2.3 非同期リセットの解析

クロックのタイミング制約が設定されたパスでは、非同期リセット（およびセット）のパスも解析されます。解析対象となるのは、FPGA 内部のレジスタで生成された非同期リセットです。

図 10-7. リセット解析パス



ツールのデフォルト設定では、図 10-7 (a) のようにリセットを生成したレジスタ "reg0" から、"reg1" のレジスタのリセット入力までが解析の対象となります。更に "reg0" 出力から "reg1" の Q 出力を介して "reg2" の D 入力までのパスを解析する必要がある場合は、制約ファイル内の **BLOCK RESETPATHS** 制約記述を削除してください。

10.3 クロックドメイン間の制約設定と解析

10.3.1 基本的な制約方針

任意の 2 クロックドメインを跨ぐ (CDC: Clock Domain Crossing) パスに対する制約設定と解析については、例えば始点/終点レジスタのクロック信号名が異なるパスのタイミング解析では、クロック構成により解析可否が異なります。幾つかの具体例について次節以降に見ていきますが、まず制約設定の基本的な考え方や Diamond の振る舞いをまとめます。

解析のデフォルト条件

周波数が整数倍の関係 ~ 高速クロックの一周期相当を適用

それ以外 ~ 当該 2 クロック間の「最小となるエッジ~エッジ間隔」を適用。何も対処しないと制約がきつくなりすぎるので、要注意

高速クロックから低速クロックへ

どちらかという、より適切なのは **MULTICYCLE** 制約。ただし、高速クロックに伴うデータが毎周期変化する場合は適用できない

低速クロックから高速クロックへ

適切な制約があるとすれば **BLOCK** (解析対象外とするべき) と考えられる

位相や周波数として特定の依存関係がない

BLOCK 制約で解析対象外であることを明示的にツールに指示する。何も設定しないとタイミング・カバレッジが満足するレベルを満たせない可能性が大きくなる

図 10-8 の例を元に振る舞いを示します。図中左下のボックスは派生クロック (分周クロックやゲーテッド・クロック) の生成、或いは PLL をカスケード接続した場合等を想定しています。クロックツリーを辿って共通のクロックソースにたどり着くクロック間であれば同じような解析対象となります。

図 10-8. CDC (クロックドメイン間) パスの極端な例

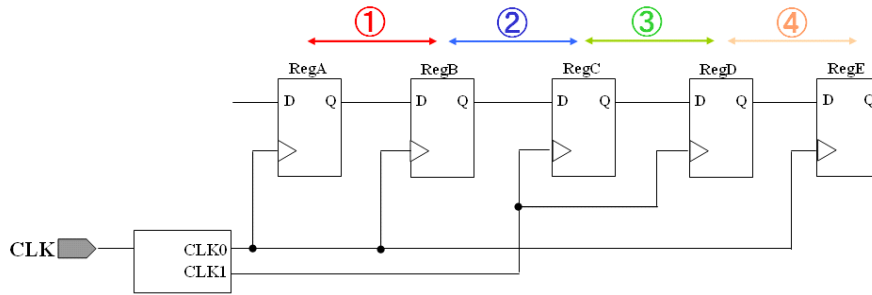


図 10-9 は CLK0 周波数が CLK1 周波数の 2.5 倍の場合に、図 10-8 のパス②に適用される制約について記述しています。終点レジスタのクロック CLK1 の方が低速のため、両クロックエッジ間の最小時間を制約として適用します。一方、同じ周波数関係でパス④では、逆に終点レジスタのクロック CLK0 の方が高速のため、CLK0 の一周期相当が制約として適用されます。

図 10-9. CLK0 => CLK1 CDC パスの適用例 1 (CLK0 周波数が CLK1 周波数の 2.5 倍)

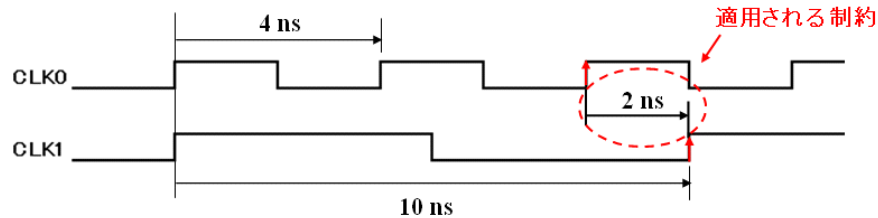
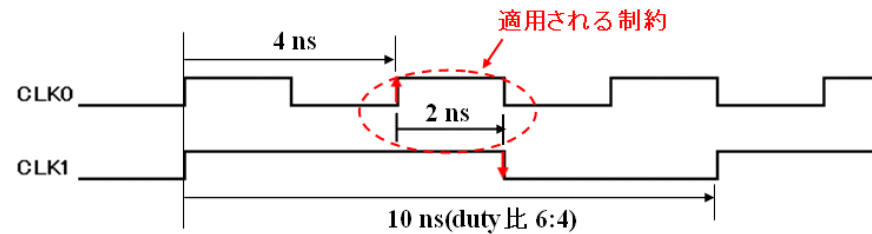


図 10-10 は CLK1 のデューティ比が 6:4 の例で、CLK1 が立ち下がりエッジ有効の場合です。これもエッジ間が最小となる時間相当が制約として適用されます。始点/終点レジスタで使用しているクロックエッジが異なる場合でも、エッジおよびデューティ比を考慮した解析が行われます。

図 10-10. CLK0 => CLK1 CDC パスの適用例 2 (例 1 と同様で CLK1 は立ち下りかつデューティ 6:4)



解析の際には両クロックの (PLL の位相シフトも含む) スキューを考慮した解析が行われます。各パスの解析結果は終点側クロックの解析レポートに出力されます。

10.3.2 パスの始点/終点のクロックソースが同じ場合の解析

CDC パスの 2 レジスタ間で、始点と終点のクロック信号名が異なる場合でも、それぞれの元のクロック入力ピンが同じ場合があります。例えば 同一 PLL の出力クロック相互や、派生クロックとオリジナルのクロック間等がこれに該当します。このようなケースでは以下の解析が可能です。

- ・最大動作速度 (周波数または周期) ※1
- ・レジスタ間のホールド時間 ※1
- ・Set/Reset のリカバリ時間 ※1 ※2

※1 : 適切な解析を行うためには周波数/周期制約だけでなく、MULTICYCLE 制約等が必要になる場合もあります

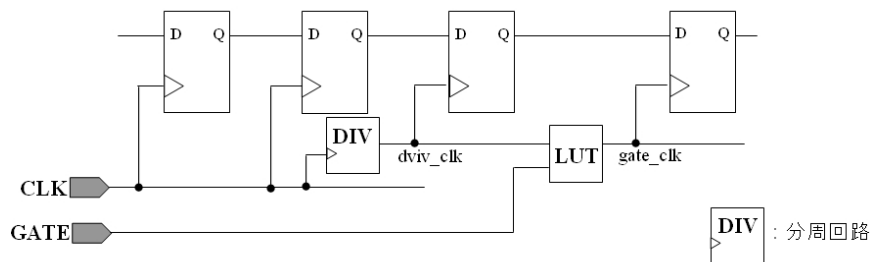
※2 : オプション設定による解析範囲が変わります

10.3.3 派生クロックへのタイミング制約の伝播

クロックのタイミング制約は、制約設定を与えられたクロックから生成された派生クロック（分周クロックやゲートド・クロック）に伝播します。例えば、図 10-11 のように外部からの入力クロックを分周したりゲートングしたりしている場合、入力ポート "CLK" に制約を設定すれば、そこから生成される分周クロック "div_clk" およびそのゲートド・クロック "gate_clk" にも制約が伝播し、"CLK" と同じ制約が適用されます。

ただし、各クロックに個別に制約が設定されている場合は、伝播した制約よりも明示的な個別の制約が優先されます。図 10-11 を例に挙げると、"CLK" と "div_clk" に制約が設定されている場合、"CLK" の制約は "div_clk" 以降には伝播せず、"gate_clk" には "div_clk" の制約が伝播し同じ制約が適用されます。

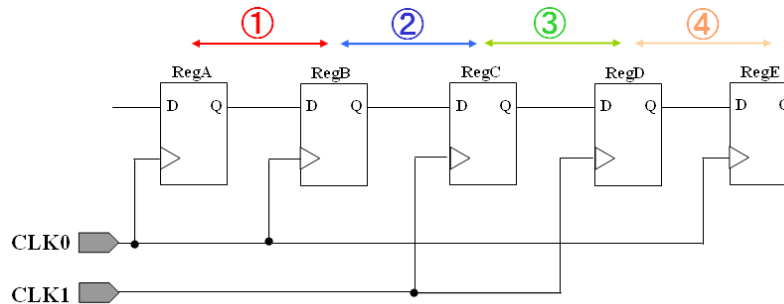
図 10-11. 制約が伝播するクロックツリーの例



なお、CLKDIV マクロなど、その出力がクロックとして用いられることを想定してデバイスに組み込まれているものを除き、**分周クロックやゲートドクロックの使用は推奨しません**。一般配線を使用するクロックとなる場合はスキュー管理できません。クロックネットワークに割り当てられるとしても、インジェクション遅延が大きくなり、バイオレーションのない安定した配置配線結果が得られるとは限りません。

10.3.4 パスの始点/終点のクロックソースが異なる場合の解析

図 10-12. クロックソースが異なるクロックドメイン間パス



それぞれのクロックソース（元のクロック入力ピン）が異なる場合は、デフォルトでは解析が行われません。図 10-12 を例にとると、"CLK0" および "CLK1" に **FREQUENCY** または **PERIOD** 制約が設定されていても、更には **MULTICYCLE** 制約や **MAXDELAY** 制約を設定してもパス②と④は解析されません。

このようなパスに関しては入力ポートでのクロック間スキューを定義することで解析させることも可能です。これにより依存関係のない2クロックを関係があるものとしてツールに認識させられます。

CLKSKEDIFF CLKPORT <port_name> CLKPORT <port_name> <max_time_unit> [MIN <time_unit>];

例えば、

```
CLKSKEDIFF CLKPORT "clk0" CLKPORT "clk1" 1.4ns Min 0.4ns;
```

などです。クロック 2 系統間の関係が周波数的 (例: 整数倍比の関係)、或いは位相的に分かり易い単純な関係であれば懸念はありませんが、そのような関係にない場合に解析が必要となる設計とすることは推奨しません。通常はデュアルポートメモリなど、タイミング要件的に分離できるような回路構造とします。

なお、CLKSKEWDIFF 制約はクロックポート間にのみ有効です。PLL 生成のクロックとの間とかは TRACE エンジンが自動的に解析しますので、指定する必要はありません。ただし、PLL の位相シフト出力についてはシフト分を自動で取りこみませんので、10.2.2 節に記述したような方法で制約します。

10.3.5 BLOCK INTERCLOCKDOMAIN PATHS 制約

Lattice Diamond には、**BLOCK INTERCLOCKDOMAIN PATHS** という、異なるクロックドメイン間のタイミング制約/解析を無効 (フォルスパス) にしてしまうオプションがあります。このオプションを使用すると、解析が可能としているパスでも全てフォルスパスに設定されます。

この制約は、最優先される制約の 1 つです。これが記述されていると、個別にクロック間に設定 MULTICYCLE 制約等を設定していても全てフォルスパスになります。

期待しないクロックドメイン間の制約/解析の無効化を防ぐため、**BLOCK INTERCLOCKDOMAIN PATHS 制約の使用は推奨しません**。クロックドメイン間のパスをフォルスパスに設定する場合は、個別に **BLOCK** 制約を設定することを推奨します。

なお、この制約はクロックドメイン内のパスの解析には影響を与えません。

10.3.6 各クロックに対する制約設定の確認

各クロックでどのような制約が適用されているかは、タイミング解析レポートで確認することができます。

タイミング解析レポートの最後には、クロックごとにそのクロックドメイン内で適用された制約 (図 10-131 ①)、クロックドメインをまたぐパスがある場合は対象となるクロックとそのドメイン間パスで適用された制約 (図 10-13 ②) についてレポートされます。

図 10-13. クロックドメイン解析レポート

<pre>Clock Domain: sr_clk_div_inferred_clock_2 Source: SLICE_288.Q0 Loads: 138 Covered under: FREQUENCY NET "sr_clk_div_inferred_clock_2" 13.000000 MHz ;</pre>	①
<pre>Data transfers from: Clock Domain: s_gclk Source: SLICE_409.F0 Not reported because source and destination domains are unrelated.</pre>	②
<pre>Clock Domain: HOLDN_c Source: SPIAA/SLICE_71.Q0 Covered under: FREQUENCY NET "sr_clk_div_inferred_clock_2" 13.000000 MHz ; Transfers: 1</pre>	
<pre>Clock Domain: SPMCLK_c Source: SPMCLK.PAD Covered under: FREQUENCY NET "sr_clk_div_inferred_clock_2" 13.000000 MHz ; Transfers: 3</pre>	

この解析レポートをレビューして、必要な制約が設定されていること、不要な制約が削除されていることを確認してください。

10.4 I/O 信号のタイミング解析

10.4.1 出力負荷設定

Lattice Diamond で適切な出力信号の遅延を見積もるためには、出力ピンごとの出力負荷を設定する必要があります。出力負荷は SSO 解析でも必須ですので、必ず適切な値を設定するようにします。

10.4.2 出力信号のスキュー

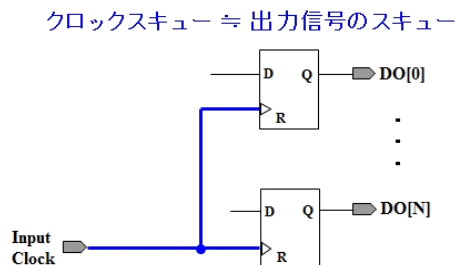
バスやソースシンクロナスなどの出力信号間でスキュー値が必要な場合、I/O タイミング解析レポートで示されるスキュー値は、2つのパラメータが考慮されていないため必ずしも適切でない場合があります。その1つはクロックスキュー、もう一つは出力バッファの立ち上がり／立ち下り遅延の差分です。

10.4.2.1 クロックスキューについて

Lattice Diamond のタイミング解析では、クロック専用配線の遅延がレジスタの位置にかかわらず全て同じ値になっています（スキューなし）。エッジクロックも同様です。これは「スキュー値がない」ということではなく、タイミングモデルとして、FF などのセットアップ時間 / ホールド時間要件の中に組み込まれているということです。

このため、出力信号の最大遅延および最小遅延の解析結果自体は正しく計算されますが、この解析から算出される信号間のスキューは、クロックスキューが考慮されていない値になっています。データシートにはクロック専用配線のスキューがパラメータとして記載されていますので、「出力信号間のスキュー」が必要な場合は、レポートから算出した値にクロックスキューの値を加算します。

図 10-14. スキューを計算できる回路構成



10.4.2.2 出力バッファの立ち上がり／立ち下り遅延

Lattice Diamond のタイミング解析では、出力バッファ遅延はバッファタイプや出力負荷に応じた最大値／最小値が適用されます。出力信号の最大遅延および最小遅延の解析結果自体は正しく計算されますが、これらの値は立ち上がり／立ち下りどちらか遅延が大きな方が適用されています。立ち上がり／立ち下りの遅延差はデータシートにも記載されていません。

この値を知るためには、例えば適切な負荷条件での IBIS シミュレーションを行い、その結果から遅延差を求めることになります。

10.5 改訂履歴

Ver.	Date	page	内容
3.1v2.0	2014, July		章タイトルを「タイミング制約設定ガイド」に変更し、構成を大幅変更 ・クロックドメイン内と間に分けて記述、項番含めて見直し ・MAXDELAY、BLOCK 等、種々について振る舞いとガイドラインを追記
3.3	Mar.2015	10-9	・10.3.4 節、CLKSKEWDIFF に関して追記

--- *** ---