


# Gowin デバイス設計最適化・解析 ユーザーガイド

SUG113-1.2J, 2022-06-30

## 著作権について(2022)

著作権に関する全ての権利は、**Guangdong Gowin Semiconductor Corporation** に留保されています。

**GOWIN高云**、、Gowin、GowinSynthesis、及び GOWINSEMI は、当社により、中国、米国特許商標庁、及びその他の国において登録されています。商標又はサービスマークとして特定されたその他全ての文字やロゴは、それぞれの権利者に帰属しています。何れの団体及び個人も、当社の書面による許可を得ず、本文書の内容の一部もしくは全部を、いかなる視聴覚的、電子的、機械的、複写、録音等の手段によりもしくは形式により、伝搬又は複製をしてはなりません。

## 免責事項

当社は、GOWINSEMI Terms and Conditions of Sale (GOWINSEMI 取引条件) に規定されている内容を除き、(明示的か又は黙示的かに拘わらず) いかなる保証もせず、また、知的財産権や材料の使用によりあなたのハードウェア、ソフトウェア、データ、又は財産が被った損害についても責任を負いません。当社は、事前の通知なく、いつでも本文書の内容を変更することができます。本文書を参照する何れの団体及び個人も、最新の文書やエラッタ (不具合情報) については、当社に問い合わせる必要があります。

## バージョン履歴

日付	バージョン	説明
2016/12/20	1.0J	初版。
2017/09/11	1.1J	<ul style="list-style-type: none"><li>● GowinデバイスのDSPブロックサポートを更新</li><li>● タイミングレポート内容の説明を変更。</li></ul>
2022/06/07	1.2J	<ul style="list-style-type: none"><li>● サポートされるデバイスを削除。</li><li>● リソース共有の説明を削除。</li><li>● 完全条件と並行条件の説明を削除。</li><li>● 3.1.1設計の制約プロセスを更新。</li><li>● 3.2ピンの割り当ての説明を更新。</li><li>● 4タイミング収束の説明を更新。</li></ul>

# 目次

目次.....	i
図一覧.....	iii
表一覧.....	iv
<b>1 本マニュアルについて .....</b>	<b>1</b>
1.1 マニュアル内容 .....	1
1.2 関連ドキュメント.....	1
1.3 用語、略語.....	1
1.4 テクニカル・サポートとフィードバック.....	2
<b>2 HDL コーディングスタイル.....</b>	<b>3</b>
2.1 HDL コーディングスタイル(一般).....	3
2.1.1 階層型コーディング合成要件.....	3
2.1.2 パイプライン処理要件.....	4
2.1.3 If-Then-Else と Case の条件比較.....	4
2.1.4 ラッチ生成の回避.....	4
2.1.5 グローバルリセットとローカルリセット.....	4
2.1.6 クロックイネーブル .....	5
2.1.7 マルチプレクサ .....	5
2.1.8 双方向バッファ .....	5
2.1.9 クロックドメインクロッシングの処理.....	5
2.1.10 メモリのコーディングスタイル .....	5
2.1.11 DSP コーディングスタイル .....	6
2.2 FSM 状態機械のエンコーディング要件.....	6
2.2.1 状態機械の概要 .....	6
2.2.2 状態機械の状態のエンコーディング方法.....	7
2.2.3 セーフ状態機械の初期状態値とデフォルト状態値 .....	7
2.3 デバイスのハードウェア特性.....	8
2.3.1 IO ロジック .....	8
2.3.2 DSP .....	8
2.3.3 BSRAM.....	8

2.3.4 SSRAM.....	8
2.4 低消費電力コーディング .....	8
2.5 合成とシミュレーションが不一致のコーディングを避ける要件 .....	8
2.5.1 センシティビティリスト .....	9
2.5.2 ブロッキング代入及びノンブロッキング代入 .....	9
2.5.3 信号のファンアウト .....	9
<b>3 設計計画.....</b>	<b>10</b>
3.1 Gowin ソフトウェアでの設計計画プロセス .....	10
3.1.1 設計の制約プロセス .....	10
3.1.2 設計計画ツール .....	11
3.2 ピンの割り当て .....	11
3.2.1 ピン割り当て規則.....	12
3.2.2 ピン・マイグレーション .....	12
3.3 クロック割り当て.....	12
3.3.1 クロックリソース割り当て規則 .....	12
3.3.2 クロックリソース割り当て制約 .....	13
3.4 ロジックリソース制約.....	13
3.4.1 定義.....	13
3.4.2 制約の構文.....	13
3.4.3 制約ストラテジー .....	14
3.4.4 特別な注意.....	14
<b>4 タイミング収束 .....</b>	<b>15</b>
4.1 合成段階のタイミング収束ストラテジー.....	15
4.2 配置配線タイミング収束ストラテジー .....	16
4.2.1 タイミング制約 .....	17
4.2.2 オプションの設定.....	17
4.3 タイミングの問題の解決 .....	18
4.3.1 合成タイミングレポートの解析 .....	18
4.3.2 配置配線レポート .....	19
4.3.3 配置配線のタイミングレポートの解析 .....	20

## 図一覧

図 2-1 Gowin ソフトウェア IP Core Generator--メモリ .....	6
図 2-2 Gowin ソフトウェア IP Core Generator--DSP .....	6
図 3-1 FloorPlanner ツール GUI.....	11
図 3-2 Global Clock Constraints インターフェース .....	13
図 4-1 タイミングの問題の解決プロセス .....	18
図 4-2 Max Frequency Summary .....	19
図 4-3 デザインのリソース使用率 .....	20
図 4-4 クロックリソース使用率.....	20
図 4-5 タイミング最悪のパス .....	21

# 表一覧

表 1-1 用語、略語 .....	1
-------------------	---

# 1 本マニュアルについて

## 1.1 マニュアル内容

FPGA 設計の最適化は主にコーディングスタイル、設計計画、タイミング収束で構成され、これらの要素は FPGA 設計の成功の可否に直接関わります。

コーディングスタイルは FPGA 設計の実現に直接影響し、最終的には設計の性能を左右します。合成ツールが一連の最適化アルゴリズムを導入しているとはいえ、ユーザーはやはり一定のコーディングスタイルに従って合成ツールを使用し、特定の FPGA アーキテクチャにおいて最適な結果を達成しなければなりません。

設計計画は、ユーザーが選択した FPGA に設計をより適合させ、エリアと速度の要件バランスを取るようガイドするものです。これによって、ユーザーは Gowin デバイスがサポートするすべての機能と特性を完全に使いこなせるようになります。

タイミング収束は、ユーザーの設計がある特定のタイミング要件を満たすことを保証します。このパートでは、主にタイミング要件、タイミング制約、タイミング最適化の方法について説明します。

## 1.2 関連ドキュメント

GOWIN セミコンダクターの Web サイト [www.gowinsemi.com/ja](http://www.gowinsemi.com/ja) から、以下の関連ドキュメントがダウンロード、参考できます：

- Gowin ソフトウェア ユーザーガイド([SUG100](#))
- Gowin タイミング制約ユーザーガイド([SUG940](#))
- Gowin 物理制約ユーザーガイド([SUG935](#))

## 1.3 用語、略語

下表に、本マニュアルで使用される用語、略語、及びその意味を示します。

表 1-1 用語、略語

用語、略語	正式名称	意味
FPGA	Field Programmable Gate Array	フィールド・プログラ



用語、略語	正式名称	意味
		マブル・ゲート・アレイ
HDL	Hardware Description Language	ハードウェア記述言語
FSM	Finite State Machine	有限状態機械
DSP	Digital Signal Processor	デジタル信号処理ブロック
BSRAM	Block Static Random Access Memory	ブロックSRAM
SSRAM	Shadow Static Random Access Memory	分散SRAM
RTL	Register Transfer Level	レジスタ転送レベル
CST	Physical Constraint	物理制約
SDC	Synopsys Design Constraint	タイミング制約
CFU	Configurable Function Unit	コンフィギュラブル機能ユニット
CLS	Configurable Logic Section	コンフィギュラブル論理セクション

## 1.4 テクニカル・サポートとフィードバック

GOWIN セミコンダクターは、包括的な技術サポートをご提供しています。使用に関するご質問、ご意見については、直接弊社までお問い合わせください。

Web サイト : [www.gowinsemi.com/ja](http://www.gowinsemi.com/ja)

E-mail : [support@gowinsemi.com](mailto:support@gowinsemi.com)

# 2 HDL コーディングスタイル

## 2.1 HDL コーディングスタイル(一般)

### 2.1.1 階層型コーディング合成要件

複雑なシステム設計を行う際は、階層型コーディングが必要となります。階層型コーディングの場合は、すべてのモジュールを一度に合成することも、階層構造に従って各モジュールを個別に合成することもできます。すべてのモジュールを一度に合成する時には、デザインは、階層構造のない1つのモジュールに、または階層構造を持つ複数のモジュールに合成できます。

2つのストラテジーにはそれぞれメリットとデメリットがあります。複雑なシステム設計では階層型コーディングが有利です。その理由は、階層型コーディングにより、モジュールの再利用性が向上し、開発サイクルが短縮されると同時に、問題の特定が容易になるためです。以下は、階層コーディング構造を作成する際の推奨事項です。

1. トップレベルのモジュールはサブモジュールのコールにのみ用い、制御ロジックができる限り各サブモジュールで実現されるようにします。
2. ピンの入出力バッファはトップレベルでインスタンス化します。
3. すべての入力、出力、双方向ピンはトップレベルでインスタンス化します。
4. トップレベルのみで双方向ピンのトライステートメントの使用を許可します。
5. できる限り、モジュールのすべての出力信号がレジスタを使用するようにします。これには次のようなメリットがあります。
  - 組み合わせロジックとレジスタを1つのモジュールに配置することで、モジュール間で合成最適化できないという欠点を克服。
  - 関連するロジックを1つのモジュールに入れ、リソースの共有を保証し、クリティカルパスを最適化できる。

- 関連性のないロジックを別々のモジュールに分けることで、異なる最適化ストラテジーを採用できる。例えば、速度優先かエリア優先か、など。

## 2.1.2 パイプライン処理要件

パイプライン処理は、複数のロジックレベルのデータパスを再構成し、クロックサイクルを増やしてロジックレベルを分割することで、設計性能向上の目的を果たします。パイプラインの構造は、データパスの高速化に効果的ですが、データパスの伝送遅延を増やすことに注意しなければなりません。

## 2.1.3 If-Then-Else と Case の条件比較

If-then-else 構造は、優先順位条件ロジックを生成し、case 構造はバランス条件ロジックを生成します。「if ステートメント」をネストする場合は、5 レベル未満にすることをお勧めします。

If-then-else 構造は複数の異なる条件式を含むことができますが、case 構造は 1 つの条件式しか含むことができません。条件が排他的状況では、if-then-else 構造と case 構造は同等であると考えられます。

## 2.1.4 ラッチ生成の回避

ユーザーは、FPGA 設計においてラッチの使用を避けるべきです。これは、合成ツールが組み合わせロジックループによってラッチを生成すると、リソースの無駄と性能の低下を引き起こし、組み合わせロジックループによって静的タイミング解析に大きな支障が生じるからです。

合成ツールは、ロジック条件式の組み合わせが不完全な場合にラッチを生成します。例えば、if-then-else 構造で else がない、または case 構造で default がないなどです。不要なラッチが生成されないよう、条件文においてすべての条件式をトラバースする必要があります。

ユーザーが case ステートメントを使用し、デフォルト条件の出力値を気にしない場合は、`/* synthesis full_case*/` 制約を追加して、ラッチの生成を回避できます。

## 2.1.5 グローバルリセットとローカルリセット

Gowin FPGA 製品には、コアロジックに直接接続されるグローバルセット/リセット(GSR)ネットワークが構築されています。グローバルセット/リセット(GSR)ネットワークは非同期/同期リセットまたは非同期/同期セットに使用できます。CFU と I/O のレジスタは、いずれも個別に構成できます。グローバルリセットのリソースは、一般の配線リソースを占有せず、チップの隅々に分布します。ローカルリセットは通常、ファンアウトより小さく、一般の配線をリセット信号に使用することを考慮できます。

## 2.1.6 クロックイネーブル

FPGA 設計では、ゲーティングクロックの使用を避けるべきです。これは、ゲーティングクロックは予測不能なクロックグリッチなど、制御不能なタイミングの問題を引き起こすおそれがあるからです。Gowin デバイスの CLS アーキテクチャには専用のクロックイネーブル信号が含まれ、ユーザーはクロックイネーブルを使用してゲーティングクロックと同様の機能を実現でき、タイミング問題を懸念する必要はありません。以下は Gowin デバイスのクロックイネーブルを使用する時の注意事項です。

1. クロックイネーブルをサポートするのはレジスタのみで、ラッチはサポートしません。
2. 同じCFU内の各CLSは1つのクロックイネーブル信号を共有します。
3. すべてのレジスタのクロックイネーブルはアクティブ High です。

## 2.1.7 マルチプレクサ

CLS 内のルックアップテーブルはフレキシブルな構成が可能で、2 入力 1 出力、3 入力 1 出力、4 入力 1 出力、5 入力 1 出力のマルチプレクサなどを実現します。ユーザーは複数の 4 入力 LUT をカスケード接続することで、より大きなマルチプレクサを生成できます。

## 2.1.8 双方向バッファ

双方向バッファを使用すると、ピンの節約と出力イネーブルのコントロールができ、消費電力節減の目的を達成します。ユーザーは合成ツールの自動ピン入出力バッファの挿入オプションをオフにし、特定のピンの入出力バッファをインスタンス化できます。

## 2.1.9 クロックドメインクロッシングの処理

あるクロックドメインから別のクロックドメインにデータパスがまたがる時、発生するメタスタビリティがデータのセットアップ時間及びホールド時間に影響を与えないようにしなければなりません。シングルビット信号については、2 レジスタ構造または 3 レジスタ構造を利用してメタスタビリティを消去することを推奨します。マルチビット信号については非同期 FIFO の使用を推奨します。

## 2.1.10 メモリのコーディングスタイル

ランダムメモリのビヘイビア・レベル説明はカスタマイズ可能かつ非常に直感的ですが、異なるコーディングスタイルは異なる合成結果が生じる可能性があります。Gowin デバイスでは、Gowin ソフトウェア IP Core Generator を用いてブロックメモリ、分散メモリ、および FIFO を生成することを推奨しています。

Gowin デバイスは、デュアルポート RAM、シングルポート RAM、セミ・デュアルポート RAM、ROM、同期 FIFO、非同期 FIFO など、多くのメモリ構造をサポートしています(図 2-1)。

図 2-1 Gowin ソフトウェア IP Core Generator--メモリ

Name	Version
Hard Module	
Memory	
Block Memory	
DPB	1.0
SDPB	1.0
SP	1.0
pROM	1.0
Shadow Memory	
RAM16S	1.0
RAM16SDP	1.0
ROM16	1.0
Soft IP Core	
Memory Control	
FIFO	
FIFO	1.1
FIFO HS	1.0
FIFO SC	1.1
FIFO SC HS	1.1

### 2.1.11 DSP コーディングスタイル

DSP のビヘイビア・レベル説明はカスタマイズ可能かつ非常に直感的ですが、異なるコーディングスタイルは異なる合成結果が生じる可能性があります。Gowin デバイスでは、Gowin ソフトウェア IP Core Generator を用いて DSP を生成することを推奨しています。

Gowin デバイスは、ALU54、MULT、MULTALU、MULTADDALU、PADD など、多くの DSP 構造をサポートしています(図 2-2)。

図 2-2 Gowin ソフトウェア IP Core Generator--DSP

Name	Version
Hard Module	
DSP	
ALU54	1.0
MULT	1.0
MULTADDALU	1.0
MULTALU	1.0
PADD	1.0

## 2.2 FSM 状態機械のエンコーディング要件

有限状態機械はクロックエッジにおいて現在状態から次の状態への遷移を完了します。以下に、主に状態機械のエンコーディングの方法とストラテジーを説明します。

### 2.2.1 状態機械の概要

状態機械を実装するには、主に 3 つの方法があります。1 つ目は、状態遷移と状態出力を 1 つのプロセスで同時に処理することです。2 つ目は、

状態遷移を独立したプロセスで処理し、もう 1 つのプロセスは、状態遷移の法則と状態出力を処理することです。3 つ目は、3 つの独立したプロセスが、それぞれ状態遷移、状態遷移法則、および状態出力を処理することです。ここでは三つ目を推奨します。閲覧と変更がしやすいだけでなく、状態を直接出力する場合に余分な遅延が生じないためです。

## 2.2.2 状態機械の状態のエンコーディング方法

現在、状態機械のエンコーディング方法は、主にバイナリ、ワンホット、グレイコードの 3 種類があります。バイナリ及びグレイコードは少ないフリップフロップを使用しますが、多くの組み合わせロジックを使用し、ワンホットとは正反対です。

ワンホットエンコーディングの最大の利点は、状態比較中に 1 ビットのみを比較する必要があることです。これにより、デコードロジックがある程度簡素化されます。同じ数の状態を表す場合、ワンホットエンコーディングはより多くのビットを占有しますが(つまり、より多くのフリップフロップが使用される)、これらのフリップフロップが占める領域は、デコード回路の簡素化によって節約される領域を相殺できます。

このため、状態の数が 5 未満の状態機械については、バイナリ及びグレイコードを推奨します。状態の数が 5 以上の場合はワンホットコードを推奨します。

## 2.2.3 セーフ状態機械の初期状態値とデフォルト状態値

セーフ状態機械は、パワーアップ後に初期化され、初期有効状態となります。ユーザーは、パワーアップリセットまたはグローバルリセット操作によって初期有効状態にできます。同時に、状態機械が無効状態にならないように、デフォルトの状態値が必要です。コード内の状態の組み合わせがトラバースされない場合、無効状態が生成されます。

## 2.3 デバイスのハードウェア特性

### 2.3.1 IO ロジック

入出力ロジックは、シリアル/パラレル変換、パラレル/シリアル変換、遅延制御及びバイトアライメント等の機能に集積され、主に高速データ送信時に使用されます。入出力ロジックは、ベーシックモード、シングルデータレート(SDR)、ダブルデータレート(DDR)など、多くのモードをサポート。ユーザーは、設計ニーズに応じて Gowin デバイスの入出力ロジックを使用できます。

### 2.3.2 DSP

LittleBee と Arora ファミリーのデバイスの DSP は 9bit、18bit の 2 種類の幅の符号付き数と符号なし数の乗算器、アキュムレータ、乗算アキュムレータ、54bit の ALU、バレルシフタをサポートします。また、入出力レジスタのパイプラインとバイパス機能もサポートします。

### 2.3.3 BSRAM

LittleBee と Arora ファミリーのデバイスの各 BSRAM は最大容量が 18K ビットで、そのデータ幅もアドレスの深さも構成可能です。各 BSRAM には 2 つのポートがあります。これらのポートは互いに独立し、独立したクロック、アドレスポート、データポート、制御ポートがありますが、メモリ領域を共有します。各 BSRAM は、シングルポートモード、デュアルポートモード、セミ・デュアルポートモード、及び読み出し専用モードという 4 つの操作モードに構成できます。また、BSRAM は出力レジスタのパイプラインおよびバイパス機能をサポートします。

### 2.3.4 SSRAM

LittleBee と Arora ファミリーのデバイスの SSRAM は、深さ 16、データ幅 1/2/4bit のシングルポート RAM、深さ 16、データ幅 1/2/4bit のセミデュアルポート RAM、または 16bit x 1 の ROM に構成できます。

## 2.4 低消費電力コーディング

エリアを最適化目標とすることで、ロジックと配線リソースの使用率を下げ、消費電力を削減することが可能です。ここでは、Gowin ソフトウェア IP Core Generator で Gowin デバイスの内部基本ユニットを呼び出すことを推奨します。これによって既知のグリッチを除去し、信号のトグルレートを下げ、適切な時刻にスリープ状態に入ることによって、システムの消費電力を低減させられます。

## 2.5 合成とシミュレーションが不一致のコーディングを避ける要件

特定のコーディングスタイルによって、合成とシミュレーションの結果が一致しない場合があります。これは、シミュレーションツールがエラー情報を無視し、これらのエラー情報が合成ツールによって検出されるためです。したがって、**Gowin** コーディングスタイルの使用を推奨します。

### 2.5.1 センシティビティリスト

組み合わせロジックの信号センシティビティリストには、合成とシミュレーション結果が一致するよう、すべての入力信号と出力信号が含まれている必要があります。

### 2.5.2 ブロッキング代入及びノンブロッキング代入

通常、組み合わせロジックにはブロッキング代入、タイミングロジックにはノンブロッキング代入の使用を推奨します。

### 2.5.3 信号のファンアウト

信号ファンアウト制御は、合成後のファンアウトが適切な範囲にあることを保証し、合成ツールは回路の複製によって信号のファンアウトを制御します。特定の信号に合成属性 **syn\_maxfan** を使用すると、より高いタイミング収束効果を得られます。

**Gowin** デバイスのアーキテクチャは、専用のクロックネットワークを使用することで大きなファンアウトのクロック信号を処理し、専用のグローバルリセットネットワークを使用することで大きなファンアウトのリセット信号を処理できます。ただし、合成ツールは大きなファンアウトのロジックに対して回路の複製を行う傾向があります。回路の複製は一連の副作用をもたらすため、ユーザーは、複製回路が多すぎないように、実際の状況に応じて合成属性 **syn\_maxfan** を柔軟に使用する必要があります。



# 3 設計計画

FPGA 設計計画には、主に 2 つの段階があります。

1. 設計前に、設計の機能と構造を定義し、適切な FPGA デバイスを選んで RTL コーディングを行う必要があります。
2. 設計中は、設計を最適な方法によって、選択したデバイスに適合させ、FPGA デバイスのリソースを十分に利用する必要があります。

この 2 つの段階は互いに影響し合っています。本章では主に第 2 段階を重点的に紹介し、ユーザーが Gowin ソフトウェアを使用して Gowin デバイスの機能と特性を活用できるようにします。

## 3.1 Gowin ソフトウェアでの設計計画プロセス

すべての設計に設計計画が必要なわけではありませんが、ほとんどの設計、特にリソース使用率が高く、タイミング要件が厳しい大規模設計には役立ちます。これらの大型設計の場合、設計計画は潜在的な配置配線及びタイミング収束の問題を軽減し、設計の再利用性と移植性を高めます。

Gowin ソフトウェアにおいて、設計計画は合成の完了後、配置配線の開始前に行われます。CST ファイルには、バックエンドの配置配線をガイドするためのすべての設計計画の制約が含まれています。いったん設計計画が変更されると、CST ファイルも変更され、全体の設計プロセスは合成完了後及び配置配線開始前の段階に戻ります。CST ファイルの制約と使用方法は、『Gowin 物理制約ユーザーガイド([SUG935](#))』を参照してください。

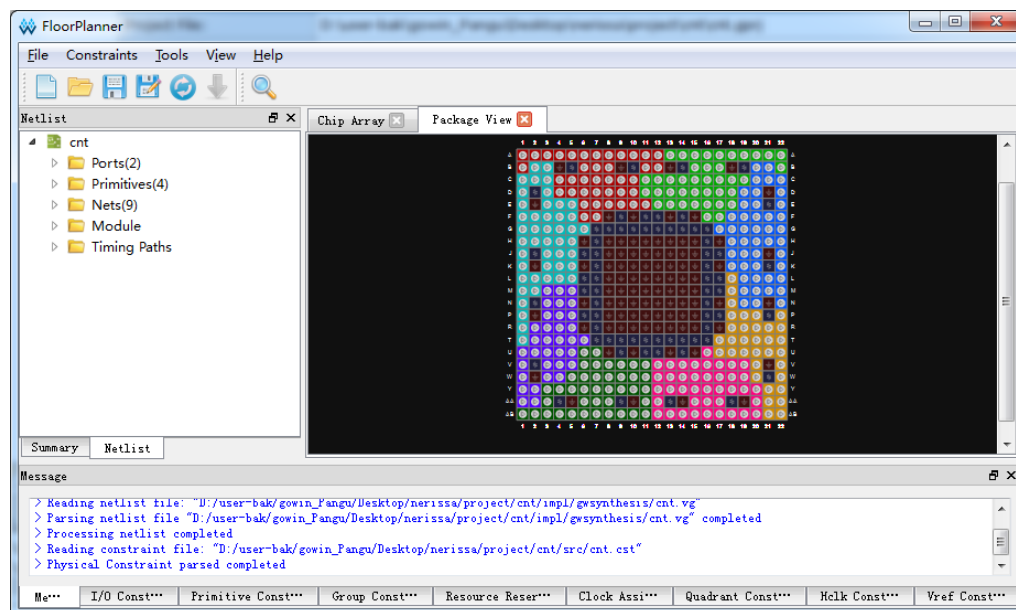
### 3.1.1 設計の制約プロセス

Gowin ソフトウェアの制約では、ユーザーによる、合成後のポート、ネットリスト、レジスタ、インスタンスの制約が可能です。CST ファイルは、ユーザーが設計計画の制約を定義するための主な入口です。

バックエンドの配置配線ソフトウェアは、CST ファイルを読み取ることでユーザーの制約情報を取得します。CST ファイルに構文エラーまたは無効な制約のエラーが生じた場合、バックエンドの配置配線ソフトウェアは直接終了します。CST ファイルはテキスト編集でき、FloorPlanner ツール

で生成できます(図 3-1)。詳しくは、『Gowin 物理制約ユーザーガイド (SUG935)』を参照してください。

図 3-1 FloorPlanner ツール GUI



### 3.1.2 設計計画ツール

Gowin ソフトウェアは、設計計画のために、次の機能を持つ FloorPlanner ツールを提供します。

- 設計でユーザーが処理可能なすべての基本論理ユニットをプレビュー。
- デバイスでサポートされるすべてのハードウェアリソースをプレビュー。
- 特定の論理ユニットを特定のハードウェアリソースに割り当てる。
- ドラッグ操作によるデバイス制約をサポート。
- 制約情報の有効性チェックをサポート。
- タイミングパスの手動調整・最適化をサポート。

## 3.2 ピンの割り当て

ピン割り当て計画は、ピンの標準と位置を定義するプロセスであり、ユーザーの実際の設計と選択したデバイスに基づきます。以下は、ピン割り当ての手順です：

1. 設計のポートを特定のピンに割り当てます。
2. ピンのレベル規格とピンの特徴パラメータを定義します。
3. 割り当ての有効性をチェックします。
4. 必要に応じて、PCB と回路図に基づきピンの割り当てと構成を変更します。

### 3.2.1 ピン割り当て規則

- まず、クロック入力、位相同期回路入力、DDR などの専用ピンを割り当てます。
- 特定ピンではなくコモンピンを専用の **BANK** に割り当てると、バックエンドの配置配線ツールが最適な結果を得ることができます。
- 多重化ピンの場合、デバイスのプログラミングモードと競合していないかどうかを確認する必要があります。

### 3.2.2 ピン・マイグレーション

同じパッケージのデバイスの場合、ユーザーは、将来の機能拡張のために低密度デバイスから高密度デバイスに、またはコストを削減するために高密度デバイスから低密度デバイスに設計を移行したい場合があります。しかし、ピン割り当ては **PCB** の再設計を避けるため、変更せず現状を保持するか、わずかな調整を行います。**Gowin** ソフトウェアは、ピン・マイグレーションの特性を提供します。**FloorPlanner** では非対応のピンを確認でき、また **LOC\_RESERVE** でこれらのピンを非表示にできます。

## 3.3 クロック割り当て

### 3.3.1 クロックリソース割り当て規則

クロックリソース計画を検討する時、その二大要素としてクロックのメイン周波数とファンアウトが挙げられます。このほか、**Gowin** デバイス自体に含まれるクロックリソースの制約も考慮する必要があります。

通常、専用のクロックリソースにより、より適切なタイミング結果を得られます。これは、専用のクロックリソースの遅延が相対的に小さく、節約できた配線リソースはより適切に、密集レベルが比較的高いシーンで使うことができるからです。ごくまれに、ユーザーはクロックを一般の配線に割り当てる必要があります。一般の配線を使用するとクロックの比較的大きな遅延が発生します。したがって、一般の配線リソースは、低速で小さなファンアウトのアプリケーションシナリオのみで使用されます。以下は、一般的なクロックリソース割り当て規則を説明です。

- ユーザーデザインにおけるクロック数と各クロックのファンアウトを決定します。
- ターゲットデバイスの提供するクロックリソースを決定します。
- 各クロックの速度要件を明確にします。
- 高速で大きなファンアウトのクロックは、通常グローバルクロックに割り当てられます。
- グローバルクロック数が設計のクロック数より少ない場合、象限クロックの使用を検討する必要があります。

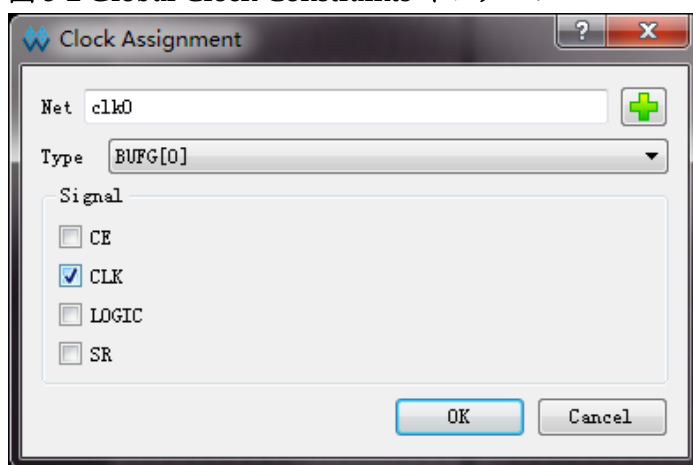
- 高スループットの高速インターフェースには、高速クロックリソースを使用します。

### 3.3.2 クロックリソース割り当て制約

NET\_LOC "xxx" BUFG0 = CLK | CE | SR | LOGIC

クロック信号を専用のグローバルクロックネットワークに割り当てます。BUFG0~BUFG7 はそれぞれ Gowin デバイスでサポートされる 8 つのグローバルクロックを示します。そのうち CLK は制約対象がクロックであること、CE は制約対象がクロックイネーブルであること、SR は制約対象が同期リセットであること、LOGIC は制約対象がロジックデバイスであることを表します(図 3-5)。図 3-2 に示す通りです。

図 3-2 Global Clock Constraints インターフェース



## 3.4 ロジックリソース制約

### 3.4.1 定義

ロジック制約は、論理的に設計基本ユニットを分割するもので、これによって設計の物理配置と実現方法が変更されます。ロジック制約は主に FPGA 内部リソースの位置をロックすることで実現します。

Gowin デバイスにとって、ロジック制約は設計性能の改善、特に階層構造の最適化に役立ちます。ロジック制約では、自動化とユーザー制御を組み合わせたコンパイル方法を提供し、設計のモジュール化と再利用性を高めます。これは、インクリメンタルコンパイルの基礎です。

### 3.4.2 制約の構文

INS\_LOC "cnt[5]" R2C2

特定の対象を特定の CFU 位置に制約します。

GROUP hh = { "cnt\_Z[1]" "cnt\_Z[2]" "cnt\_Z[3]" "cnt\_Z[4]" "cnt\_Z[5]"  
"cnt\_Z[6]" "cnt\_Z[7]" }

**GRP\_LOC hh R[3:6]C[4:6]**

特定対象をグループ化し、特定のエリアに制約します。

**3.4.3 制約ストラテジー**

- ロジックモジュールの分割は階層構造に基づきます。
- ロジックモジュールの分割はクリティカルパスに基づきます。
- ロジックモジュールの分割は大きなファンアウトの入出力信号に基づきます。
- ロジックモジュールを分割し、各モジュールに異なる最適化ストラテジーを使用します。

**3.4.4 特別な注意**

- ブロックメモリは単独の位置制約が可能で、グループ化の必要はありません。
- 大規模なロジックモジュールの位置制約では、開始位置と相対サイズを指定する必要があります。
- キャリーチェーンとバスはグループ化の必要がありません。
- 相互接続ロジックはグループ化の必要がありません。

# 4 タイミング収束

それぞれの設計は、一般に特定の速度で実行することが必要です。FPGA 設計には主にタイミング、帯域幅、遅延の 3 つの要件があります。帯域幅と遅延は排他的な関係です。帯域幅が広い場合、通常は多くのパイプラインを必要とし、これは遅延を増やします。逆に遅延を小さくするには複雑な組み合わせロジックパスが必要となり、パイプライン数と帯域幅が減少することになります。このため、設計構造を定義する時、まず帯域幅と遅延のバランスを保たなければなりません。

一般に、タイミング収束は、デザインが特定のレート要件を満たすことができるかどうかを決定する上で重要な要素です。タイミング収束を達成するには多くの時間と労力が必要です。本章では、主にタイミング収束に焦点を合わせ、設計の中でタイミングを収束させる方法について説明します。

## 4.1 合成段階のタイミング収束ストラテジー

GowinSynthesis で合成する場合、タイミング収束をよりよく達成するために従うべきいくつかの一般的なガイドラインがあります。

1. Gowin の RTL コーディングスタイルに従います。
2. 適切な属性制約を使用して、合成をガイドします。
3. 入出力レジスタを使用し、ピンタイミングを改善します。入力ピンレジスタを使用すると、入力セットアップ時間を改善できます。出力ピンレジスタを使用すると、クロックから出力時間を改善できます。
4. IO 遅延ユニットを使用し、入力ホールド時間を改善します。入力ピンレジスタを使用すると、ホールド時間の問題が生じる場合があります。これは、データレーンの遅延が短いことが原因です。したがって、**IODELAY** をデータパスに追加して、入力ホールド時間を補正することができます。プリミティブの詳細については、『Gowin プログラマブル汎用 IO(GPIO)ユーザーガイド([UG289](#))』のセクション 4.4 を参照してください。
5. **HCLK** を使用して、入力レジスタのデータとクロックの間の関係を調整し、セットアップ時間とホールド時間の条件を同時に満たせます。

6. 専用の **GSR** リソースを使用します。つまり、RTL デザインで **Gowin** プリミティブ **GSR** をインスタンス化します。設計に大きなファンアウトのリセットまたはセット信号がある場合、専用の **GSR** リソースを使用することを推奨します。これによって配線の混雑(密集)を低減し、配線効率を高めます。
7. メイン周波数を高めるため、ファンアウトを減らします。クリティカルパスに対して **syn\_maxfan** 属性をオプションで使ってファンアウトを減らします。その代償として、レジスタの複製が必要となります。
8. ワンホット状態機械エンコーディングを使用します。高速設計については、ワンホット状態機械エンコーディングの使用を推奨しますが、リソース利用率と消費電力の増加を招きます。
9. リソースを再利用します。リソースを再利用すると、ロジックレベルが増加し、大きな遅延のパスを生成します。通常、合成ツールは非クリティカルパスに対してリソース再利用を行います。しかし例外もあり、ユーザーはタイミングの問題が生じないようにクリティカルパスをチェックすることが必要です。
10. 合成レポートをチェックし、高いファンアウトパス、クリティカルパス、大きな遅延パスを解析します。
11. 合成タイミングレポートを読みます。合成タイミングレポートには配置配線情報が含まれていないため、レポート結果は実際の結果より良くなる傾向にあります。一般的に、実際の結果は **1/3** から **1/2** 低くなります。

## 4.2 配置配線タイミング収束ストラテジー

**Gowin** ソフトウェアを使用して配置配線する時、タイミング収束をより効果的に実現するために、一般的な規則に従う必要があります。

1. 適切な物理制約およびタイミング制約を使用して、配置配線をガイドします。
2. ツールによって出力されるいくつかの警告とエラー情報に注意します。
3. 適切な配置配線のオプションを選択します。
4. クロック制約なし、非同期クロック解析なしなどの制約を避けてください。適切なタイミング制約を追加します。
5. クロック周波数が実際の要件の制約よりも高い、**IO** のマルチサイクルパスなどの過度の制約を回避します。**multicycle** または **maxdelay** を使用してタイミングを緩和できます。
6. クロックイネーブルはできる限り **GCLK** リソースを使用します。クロックイネーブルは通常、大量のレジスタを使用する大きなファンアウト信号です。一般の配線リソースを使用すると大きな遅延が生じることがあるため、**GCLK** リソースに置くことを推奨します。

### 4.2.1 タイミング制約

Gowin ソフトウェアの **Timing Constraints Editor** は、一般的なタイミング制約をサポートしています。タイミング制約構文仕様については、『**Gowin タイミング制約ユーザーガイド(SUG940)**』付録 A タイミング制約構文仕様を参照してください。クロック制約が追加されていない場合、ソフトウェアはデフォルトのクロック周波数でデザインのタイミングを解析します。

1. **LittleBee®**ファミリーのクロックはデフォルトで **50MHz** で解析されます。
2. **Arora®**ファミリーのクロックはデフォルトで **100MHz** で解析されます。
3. **setup** タイミング解析条件は、高温低圧です。
4. **hold** タイミング解析条件は、低温高圧です。

### 4.2.2 オプションの設定

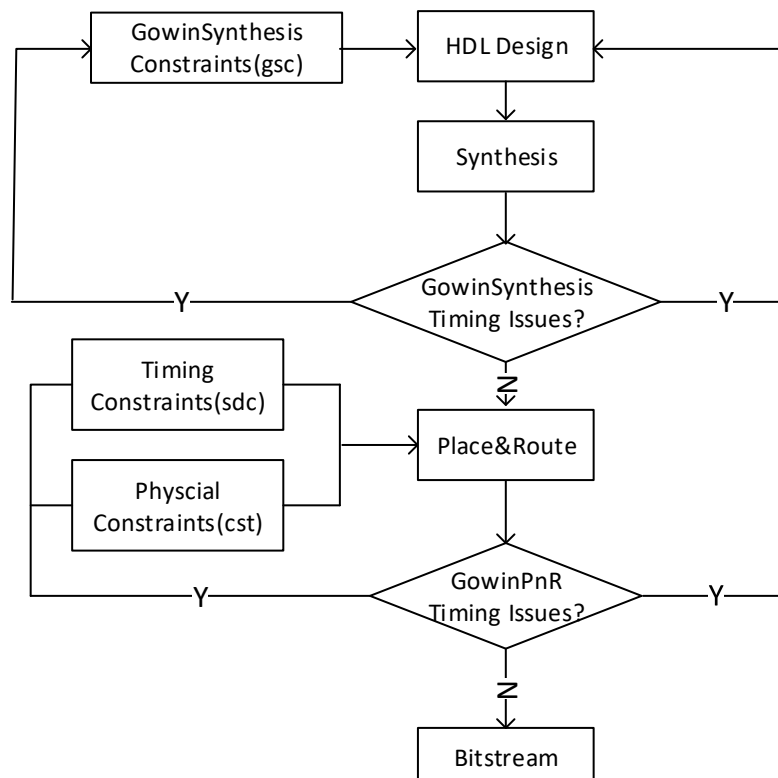
1. **Run Timing Driven** : デフォルトでオンになっています。タイミングドライバ配線であり、タイミング要件が高くない場合は、実行時間を節約するために、このオプションをオフにすることができます。
2. **Place Option** : 配置アルゴリズムのオプション(0 または 1)です。デフォルトは 0 です。
  - 0 の場合、デフォルトの配置アルゴリズムを使用します。
  - 1 の場合、アルゴリズム 0 に比べ、より良い配置結果を見つけるためにより多くの時間がかかります。
3. **Route Option** : 配線アルゴリズムのオプション(0、1、または 2)です。デフォルトは 0 です。
  - 0 の場合、デフォルトの配線アルゴリズムが採用され、密集レベルに従って配線します。
  - 1 の場合、タイミングデータに従って配線します。
  - 2 の場合、配線速度は比較的に速いです。
4. **Place input register to IOB** : 入力バッファにより駆動されるレジスタを IOB に配置して、IO ロジックのタイミングを改善します。デフォルトは True です。
5. **Place output register to IOB** : 出力/トライステートバッファにより駆動されるレジスタを IOB に配置して、IO ロジックのタイミングを改善します。デフォルトは True です。
6. **Place inout register to IOB** : 双方向バッファにより駆動されるレジスタを IOB に配置して、IO ロジックのタイミングを改善します。デフォルトは True です。



## 4.3 タイミングの問題の解決

Gowin ソフトウェアのコンパイルプロセスでは、合成最適化段階と配置配線段階で出力されるタイミング解析レポートを通じて、タイミング要件の予備確認を実行できます。タイミングの問題をチェックして解決するプロセスを図 4-1 に示します。

図 4-1 タイミングの問題の解決プロセス



### 4.3.1 合成タイミングレポートの解析

合成する前に、まず RTL デザインが Gowin のコーディングスタイルに準拠しているかどうかを確認します。たとえば、DSP の出力がレジスタを通過するかどうか、BSRAM の出力がレジスタを通過するかどうか、状態機械のエンコード方法などです。

合成後、合成レポートのタイミング解析結果を読みます。

1. 図 4-2 に示すように、“Max Frequency Summary” の内容を通じて、最大動作周波数が設計の周波数要件を満たしているかどうかを確認します。

## 図 4-2 Max Frequency Summary

Max Frequency Summary:

No.	Clock Name	Constraint	Actual Fmax	Logic Level	Entity
1	clk	50.0(MHz)	136.9(MHz)	6	TOP
2	ddk	50.0(MHz)	144.8(MHz)	5	TOP
3	clk_01ms	50.0(MHz)	73.3(MHz)	9	TOP
4	clk_100ms	50.0(MHz)	85.4(MHz)	9	TOP
5	clk_game	50.0(MHz)	81.2(MHz)	9	TOP
6	clk_5ms	50.0(MHz)	53.0(MHz)	13	TOP
7	clk_1ms	50.0(MHz)	189.2(MHz)	4	TOP

### 2. “Detail Timing Paths Information”の内容を通じて最悪パスの推定タイミングを確認します。

- 最悪パスのロジックレベル数が多い場合は、ロジックレベル数を減らすためにレジスタを挿入する必要があります。LittleBee®ファミリーC6 スピードグレードのチップを 100MHz まで実行させたい場合は、ロジックレベル数を 4 以下にする必要があります。Arora ファミリーC8 スピードグレードのチップを 100MHz まで実行させたい場合は、ロジックレベル数を 8 以下にする必要があります。
- 最悪のパスでの主な遅延が BSRAM によって引き起こされている場合は、BSRAM の最終出力を 1 クロックサイクル遅延させることができます。
- 最悪のパスでの主な遅延が DSP によって引き起こされている場合は、DSP の最終出力を 1 クロックサイクル遅延させることができます。
- 最悪のパスでの主な遅延が SSRAM によって引き起こされている場合は、合成属性 syn\_srlstyle を使用して、SSRAM をシフトレジスタに合成できます。

## 4.3.2 配置配線レポート

配置配線レポートのリソース使用率を解析することにより、リソースが適切に使用されていることを確認します。使用率が高すぎる場合は、RTL デザインを更新するか、合成属性の制約を追加する必要があります。

### デザインのリソース使用率

タイミングレポートを解析する前に、配置配線のレポートをチェックして、デザインのリソース使用率が高すぎるかどうかを確認する必要があります。一般的に、CLS の使用率が 85%を超える場合、または BSRAM/DSP のリソース使用率が 80%を超える場合、リソース使用率が高すぎるデザインであると判断できます。これは、タイミング収束の問題を引き起こすおそれがあります(図 4-3)。

図 4-3 デザインのリソース使用率

## Resource Usage Summary:

Resource	Usage	Utilization
Logic	7057/8640	81%
--LUT,ALU,ROM16	7009(1415 LUT, 5594 ALU, 0 ROM16)	-
--SSRAM(RAM16)	8	-
Register	5828/6867	84%
--Logic Register as Latch	0/6480	0%
--Logic Register as FF	5827/6480	89%
--I/O Register as Latch	0/387	0%
--I/O Register as FF	1/387	1%
CLS	3781/4320	87%
I/O Port	38	-
I/O Buf	38	-
--Input Buf	2	-
--Output Buf	36	-
--Inout Buf	0	-
IOLOGIC	0	0%
BSRAM	24 SDP	92%
DSP	18 MULT18X18	90%
PLL	1/2	50%
DCS	0/8	0%
DQCE	0/24	0%

## クロックリソース使用率

理論的には、デザインのクロックリソースは、ターゲットデバイスのクロックの総数を超えてはなりません。そうしないと、一部のクロックラインが通常の配線リソースに配線し、セットアップ違反またはホールド違反が発生する可能性があります。図 4-4 に示すように、配置配線レポートでクロックリソースの使用率を確認できます。

図 4-4 クロックリソース使用率

## Global Clock Usage Summary:

Global Clock	Usage
PRIMARY	1/8(12%)
SECONDARY	1/8(12%)
GCLK_PIN	2/7(28%)
PLL	1/2(50%)
CLKDIV	0/8(0%)
DLLDLY	0/8(0%)

## 4.3.3 配置配線のタイミングレポートの解析

## レジスタのファンアウトの減少

タイミングレポートの「Setup Analysis Report」で最悪のパスを特定し、タイミングパスの開始レジスタのファンアウト数を確認します。ファンアウトが大きすぎる場合は、レジスタをコピーすることにより、レジスタのファンアウトを減らすことができます(図 4-5)。

図 4-5 タイミング最悪のパス

Data Arrival Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					clk_usbif
0.000	0.000	tCL	RR	1	IOB48[A]	usb_ifclk_ibuf/I
0.683	0.683	tINS	RR	7392	IOB48[A]	usb_ifclk_ibuf/O
0.926	0.243	tNET	RR	1	R28C62[0][A]	usb_rf32/mov_sig_src_o_s0/CLK
1.158	0.232	tC2Q	RF	21	R28C62[0][A]	usb_rf32/mov_sig_src_o_s0/Q
2.766	1.609	tNET	FF	1	R32C33[0][B]	n5914_s2/I2
3.219	0.453	tINS	FF	3	R32C33[0][B]	n5914_s2/F

上図に示す最悪のパスの開始レジスタ “usb\_rf32/mov\_sig\_src\_o\_s0” のファンアウトは 21 であり、1.609ns の net 遅延はパス全体の遅延の 42% を占め、タイミングの収束に深刻な影響を及ぼします。 よって、RTL デザインで、次のような net 属性をこのレジスタに追加します。

```
reg mov_sig_src_o_s0 /* synthesis syn_maxfan = 10 */
```

したがって、RTL はレジスタの実際のファンアウトに従って設計する必要があります。しかし、ファンアウトが小さいほど良いというわけではありません。ファンアウトが小さいほど、レジスタソースのファンアウトが増加し、タイミング収束にも影響します。

#### BUFS リソース使用の最適化

一般的に、クロックイネーブル CE は高ファンアウト net であり、BUFS 配線リソースは配置配線段階で優先的に使用されます。最悪のタイミングパスでの遅延が主に CE が BUFS 配線リソースを使用することによって引き起こされる場合は、次のように BUFS の使用を回避するために物理的な制約を追加できます。

```
CLOCK_LOC "ce_0" LOCAL_CLOCK
```

