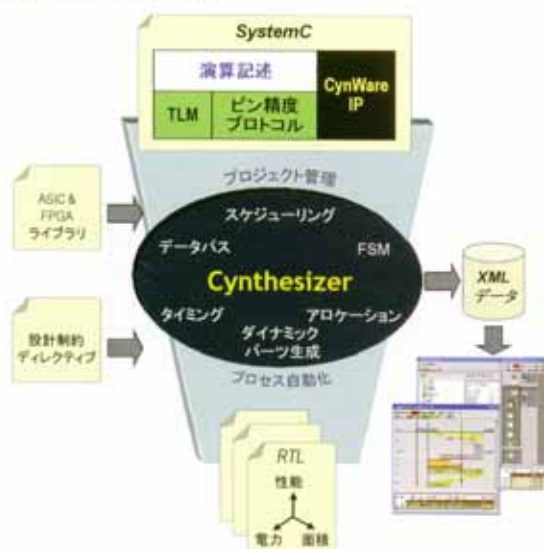


# Cynthesizer™

## 動作合成ツール Cynthesizer

高性能な電子機器に必要な半導体集積回路LSIは、ますます複雑で大規模になってきています。製品サイクルも短期化し、限られた開発期間で、高機能な製品を開発しなければなりません。設計チームは、LSI設計の大規模化と高機能化に対応して、先進的なアルゴリズムの採用、既存設計資産の利用、カスタム設計などを組み合わせながら開発努力を進めています。しかしながら、同じ人員で、より多くの機能を、短期間で設計しなければならない、従来の設計手法で設計できる規模を越えつつあります。

そのような状況の中、C++言語を拡張し、ハードウェア/ソフトウェア設計の概念から実装までをカバーするSystemC言語が提案され、IEEE 1666-2005として標準化されました。SystemCを用いることで、階層構造やプロセスによる同時性の記述が可能となり、従来、困難だったシステム検証やソフトウェアとハードウェアの協調検証、アーキテクチャのトレードオフなどが容易になりました。さらに、動作合成を用いることで抽象度の高いアルゴリズム記述からRTL記述を生成できるようになりました。動作合成を使えば、詳細な構造を記述するRTL設計に比べ、設計生産性を大幅に向上させることができます。



## 製品の特長と機能

### ■ 多くの製品やシリコンでの実証済みの実績

- ・ 演算系や制御系などの幅広い大規模デザインに対応
- ・ 手設計でのRTL記述に匹敵する優れたデザイン品質

### ■ 高品質なRTLデザインの生成

- ・ 複数RTLを生成させ、面積と性能のトレードオフにより最適ポイントを探る
- ・ カスタム・データバス・コンポーネントの生成により、高性能/小面積なデザインを実現
- ・ タイミング精度およびバックエンドのタイミング収束性を考慮した高品質なRTLの生成、

### ■ 自由度の高い記述スタイル

- ・ アンタイムなC/C++コードの利用
- ・ 階層構造の定義や同時的な複数プロセス

### ■ 幅広いデザインに対応

- ・ 一時停止可能な高性能なパイプライン構造
- ・ 配列からのメモリモデルの推定、複雑なメモリアccessを含めたスケジューリング
- ・ FPGAとLSIに対する共通設計資産

### ■ 汎用性の高い動作IP

- ・ 固定小数点、浮動小数点、ストリーム・インターフェイス、メモリ・インターフェイスなどの動作IPの利用

### ■ インターフェイス設計の容易化

- ・ モジュラ・インターフェイスによるインターフェイスのライブラリ化と再利用性の向上
- ・ インターフェイス・ジェネレータによる汎用的なモデルの自動生成
- ・ TLMとPINレベルのサポート

### ■ サードパーティツールとのリンク

- ・ 論理合成、FPGA配置配線、消費電力解析などの各種サードパーティツールとの統合

### ■ 解析と実行のためのグラフィック環境

- ・ グラフィック・インターフェイスを用いた検証、動作合成の実行、ジョブ管理および解析の容易化
- ・ 生成されたRTLに対するデータベースやステートマシン構造の表示、ソースコードとのクロスリファレンスなどによる解析機能の提供

### ■ 先進的なCynthesizer Ultra

- ・ CynthesizerとCMD(CellMath Designer)を統合
- ・ CMDのデータベース最適化機能によってデータベース部の性能を改善
- ・ 浮動小数点IPにより、浮動小数点を用いたデザインの検証と実現化が容易に

Forte Design Systems社の動作合成ツールCynthesizerは、SystemCによる抽象度の高い動作記述から高品質なRTLデザインを生成します。Cynthesizerの高いスケジューリング能力と最適化能力によって、目標性能を満足するRTLデザインを生成することができ、ハードウェア化を含めた設計期間の大幅な短縮が期待できます。動作合成可能な設計資産IPは、従来のRTLで記述された設計資産よりも設計変更が容易で汎用性が高いため、異なる仕様に対して柔軟に対応できます。

Cynthesizerは、さらに、検証や動作合成の実行を容易にするグラフィック・インターフェイスを提供しています。生成したRTLを可視化する各種機能によるRTLやソースコードの解析が可能です。Cynthesizerは動作合成だけでなく統合的な設計環境を提供しています。


Cynthesizerは、現在までに多くのASICやFPGA設計に対して実績を挙げています。

## 高品質なRTLデザインの生成

Cynthesizerは、要求するクロック周期やクロック・サイクル数、入出力遅延などの設計制約と、演算リソースのタイミングに基づいてスケジューリングを実行します。Cynthesizerの優れたタイミング解析機能と、スケジュール・アルゴリズムによって、加算器や乗算器といった演算器の使用を最小化し、演算器を最大限に活用できます。Cynthesizerは、自動的に最適なステート・マシン(FSM)を生成し、演算器で構成されたデータバス回路を制御します。

豊富なディレクティブやコマンド・オプション、たとえば、配列展開、ループ展開、パイプライン化、データバス演算のコンポーネント化などを指定することで、生成するRTLの構造を詳細に制御できます。ターゲットのASICまたはFPGAテクノロジーに対応した複数のRTL候補を短時間に生成することで、面積、性能、消費電力などの要求を満たす最適なRTLデザインを選択できます。

**SystemC 動作合成のみが  
抽象化とハードウェア設計に  
必要な要素を提供できる**

		C++	ANSI C	System Verilog
複雑性をマネージするオブジェクト指向対応	Yes	Yes	No	No
カスタムインターフェイス・合成とシミュレーション	Yes	No	No	Yes
ビット精度データ型	Yes	No	No	Yes
固定小数点データ型	Yes	No	No	No
マルチプロセスのモデル化	Yes	No	No	Yes
非同期バス	Yes	No	No	Yes
構造的な階層表現	Yes	No	No	Yes
アルゴリズム言語との親和性	Yes	Yes	Yes	No
シミュレーションと合成で一貫したセマンティクス	Yes	No	No	Yes
複数の抽象度	Yes	No	No	No

## 自由度の高い記述スタイル

SystemCは、C++言語を基にしてハードウェア記述に必要な階層、クロック、ビット精度データ型、プロセスなどの概念を追加し、標準化された設計言語です。SystemCの強力な記述能力によって、アンタイムド(Untimed)なC++アルゴリズムと、タイミングを明示的にモデル化したタイムド(Timed)なデザインを混在して記述することができます。SystemCを用いて上位システム環境の中で動作を検証した後に、入出力プロトコルを維持して動作合成を実行できます。そのため、同一のテストベンチを用いてRTLを検証できます。C++/SystemCモデルを設計の初期段階で使用すれば、従来のRTLシミュレーションと比べてシミュレーションを高速化できます。

さらに、SystemCはC++に基づいているため、C++の特長であるオブジェクト指向をハードウェア設計に適用できます。たとえば、ハードウェア設計で用いる画像データのデータ型とその操作関数、各種インターフェイスなどをライブラリ化すれば、RTLで作成したIPに比べ、設計再利用性が格段に向上します。

## 幅広いデザインに対応

Cynthesizerは、データバス処理が主体の演算系デザイン、タイミング処理が主体の制御デザインなど幅広いデザインに対応しています。

演算系では主に制約に基づいてスケジューリングを実行し、データバス構造を最適化します。画像処理ではパイプライン構造が中心となりますが、パイプライン・ストール(一時停止)や、ストールに対応するバッファリング、メモリ・アクセスを含めたパイプライン化が可能です。

制御系では、主にサイクル精度に基づいた記述と最適化を行い、また、デザインにRTL記述を含めることもできます。SystemCを用いて複雑な仕様の実現と動作合成が可能であるため、システム全体に対してSystemCの適用が可能です。

ターゲットのハードウェアはシステムLSIを含めたASICだけでなく、FPGAもサポートしているため、システムのプロトタイプ開発にも効果を発揮します。

## 汎用性の高い動作IP

CynthesizerがサポートするSystemC言語の範囲は広く、SystemCを活用すれば、設計資産の再利用が容易です。そのため、現在の設計プロジェクトだけでなく、将来の設計プロジェクトに対しても、設計の効率化が可能です。SystemCはC++に基づいているため、その特徴であるオブジェクト指向をハードウェア設計に適用し、機能やインターフェイスをライブラリ化できます。SystemCモデルは、抽象度の高いレベルで記述され、プロセス・テクノロジーとは完全に独立しています。

Cynthesizerでは、動作合成可能な動作IPをCynWareとして提供しています。CynWareには、固定小数点、浮動小数点<sup>※1</sup>、ストリーム・インターフェイス、メモリ・インターフェイスなどのIPがあります。また、各種記述サンプルも提供しています。

動作合成可能なIPは、RTL記述されたIPより汎用性や柔軟性に優れ、機能の理解や変更も容易です。設計規模の大規模化に伴い設計資産の動作IP化が進んでいます。

## インターフェイス設計の容易化

Cynthesizerのモジュラ・インターフェイスの記述スタイルによって、インターフェイスと機能本体の設計を分離することができ、両者を組み合わせたトレードオフによる最適なアーキテクチャを容易に得ることができます。また、同名のアクセス関数を持つTLMインターフェイスを使用することにより、TLMシミュレーションを実行することが可能です。多くの場合、モジュラ・インターフェイスのスタイルは、動作合成結果においてもピンレベルのスタイルより良い結果を得られています。生成されたRTLは、オリジナルのテストベンチを使って容易に検証でき、検証、合成共に設計期間を大幅に短縮できます。

さらに、Cynthesizerが提供するインターフェイス・ジェネレータ<sup>※2</sup>によって、各種のインターフェイス・モデルを生成させることができます。生成可能なイ

ンターフェイスには、trig-done、シングル/ダブル・バッファ、サーキュラ・バッファ、ラインバッファなどのインターフェイスがサポートされ、メモリ構成などのパラメータを入力してモデルを生成させます。

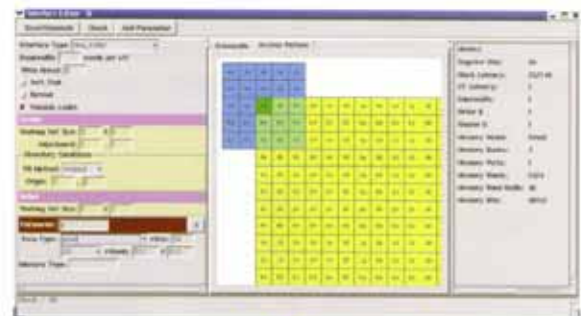
たとえば、ラインバッファ・インターフェイスでは、画像ストリーム・データから2次元データを構築するラインバッファを自動的に生成します。そのため、ユーザは、2次元データに基づいた演算処理を記述するだけで済みます。これらのモデルは動作合成可能SystemCコードとして容易にデザイン内に取り込むことができます。

インターフェイス・モデルはピンレベルでシミュレーション可能であり、動作シミュレーションと同じテストベンチを用いてCynthesizerが生成したRTLとともに検証ができます。

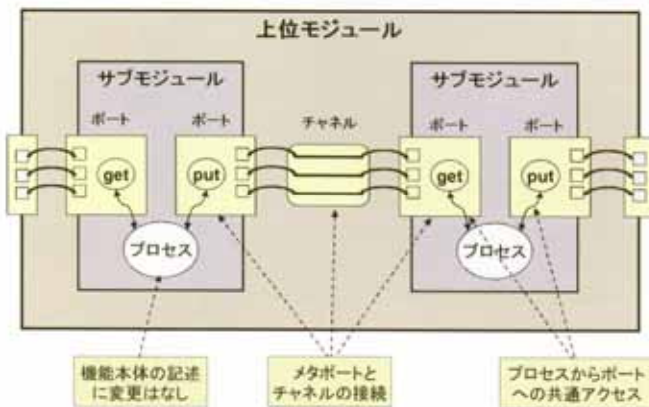
## サードパーティ・ツールとのリンク

Cynthesizerでは、動作合成の実行だけでなく検証や合成に関連したサードパーティ・ツールを同一環境から実行できます。各ツールには、論理合成、Verilogシミュレータ、SystemCデバッガ、リントチェッカ、消費電力解析、FPGA配置配線、等価検証などのツールがあります。

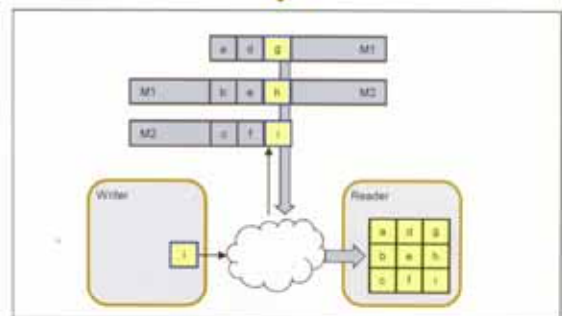
動作合成だけでなく、動作合成を含めたトータル設計フローをサポートしているため、既存の設計フローとの統合や実行が容易です。



インターフェイス・ジェネレータ



モジュール・インターフェイスによるインターフェイス設計の容易化



ラインバッファ・インターフェイス・モデル

※1 Cynthesizer Ultraにてサポート

※2 オプション製品

## 解析と実行のための グラフィック・ワークベンチ

Cynthesizerでは、設計データの管理、検証や動作合成の実行、結果の解析などを実行するワークベンチと呼ばれるグラフィック・インターフェイスを提供しています。

ワークベンチの解析ビューには、ソースコード、コントロールフロー、データフロー、レジスタユース、RTL構造の各ビューやハードウェア詳細レポートなどがあり、動作合成した結果のRTL構造をグラフィカルに解析し、ソース記述とRTLとのクロス・リンクによる追跡が可能です。

また、ワークベンチでは、シミュレーションや動作合成の実行と、それらのジョブを管理することができます。このようにCynthesizerワークベンチを用いることで、動作合成を含めた実行や解析が容易に行えます。

## 先進的なCynthesizer Ultra

Cynthesizer Ultraは、CynthesizerとCMD(CellMath Designer)を統合したツールです。CynthesizerからCMDを連携して実行することができます。

Cynthesizer Ultraでは、CMDのデータバス最適化機能を使用することで、より高性能で面積の小さい

データバス部の合成結果を得る事ができます。それにより演算が中心のデザインに効果を発揮します。

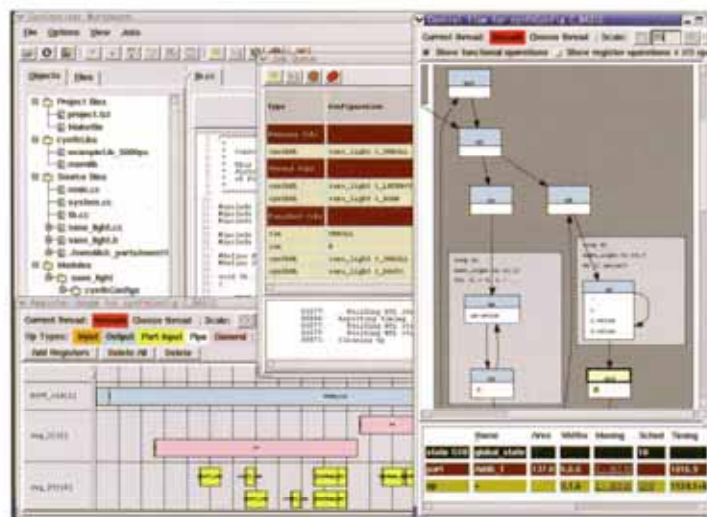
また、浮動小数点データタイプとCMDの持つ浮動小数点ライブラリの利用により、SystemC環境で浮動小数点を用いたデザインの検証と動作合成が可能になります。ライブラリには、加減乗除演算、 $\exp()$ 、 $\log()$ 、 $\sqrt{}$ 、 $\sin()$ 、 $\cos()$ などの関数が含まれ、要求する仕様に合わせ、IEEE754準拠での実現と性能を重視した実現の使い分けが可能です。Cynthesizer Ultraを用いることで、浮動小数点を用いた記述の検証と動作合成が容易になり、アルゴリズム記述からRTLに至る設計期間の短縮を可能とします。

## まとめ

現在の大規模設計システムでは、従来と同じRTL設計手法を用いてシステムLSI/FPGA設計を行うことが困難になってきています。製品開発の短縮や設計人員に関する厳しいプレッシャーに対応するため、新しい設計手法が必要とされています。今後、さらに高度化する要求に対応するためには、SystemCを用いた設計フローの構築と、動作合成ツールCynthesizerが必要です。

## 実行環境

Redhat Enterprise Linux 4.x, 5.x



CynthWB : Cynthesizerワークベンチ

フォルテ・デザイン・システムズ株式会社  
〒222-0033 神奈川県横浜市港北区新横浜2-5-10  
Tel:045-478-2268 Fax:045-472-5517 <http://www.ForteDS.com>