

## ソフトウェア資産のハードウェア化技術 C言語ハードウェア設計ツール eXCite

### ソフトウェア処理をハードウェア処理に

#### 製品の特徴

- ソフトウェアのハードウェア化を実現
- C言語によるハードウェア設計
- ハードウェア設計効率大幅アップ
- ソフト、ハードの性能を比較検討
- 自動テストベンチ生成機能搭載
- 高位合成で最適なハードウェア生成
- 通信合成で楽にインターフェース設計
- IP再利用で多彩な部品を再利用
- FPGAボードと連動させて実機検証

組み込みシステムの複雑化に伴い、ソフトウェアだけでは仕様を満足することが難しくなっています。性能が不十分な場合は、ソフトウェア処理の一部または全てをカスタムハードウェアとして実装することで、処理速度の高速化や消費電力の削減が可能になります。

しかしながら、ソフトウェア処理をハードウェア化する場合は、どの部分をハードウェア化すれば良いかの選択や、HDLでの再設計、HDLでのテスト環境構築など作業項目は少なくありません。更にHDLは並列や同期など考慮しなければならず、C言語で設計しているものに比べて、情報量も増え、設計そのものの難易度も上がります。

図1のC言語とRTL HDL行数比較から分かる通り、HDLの設計はC言語設計に比べて記述量が大幅に増えることが分かります。平均するとおおよそ10倍程度の差になるといわれています。

ソリトンシステムズでは、ハードウェアをより簡単に設計するためのソリューションとして、ソフトウェアで設計に使用しているC言語から直接ハードウェア設計する開発環境「eXCite (エキサイト)」をご提供しています。

eXCiteを導入することで、図2にあるように、従来のハードウェア設計工数を大幅に削減することが可能になります。これにより、ソフトウェア処理をハードウェア処理に容易に変更することが可能になります。

ソフトウェアからのハードウェア設計時間が短縮されることで、今まで十分に検討できていなかった、ハードウェア、ソフトウェアの分割検討や、アルゴリズムそのものの品質向上などに時間を使うことが出来るようになります。結果として、システム全体の製品性能の向上も検討できるようになります。

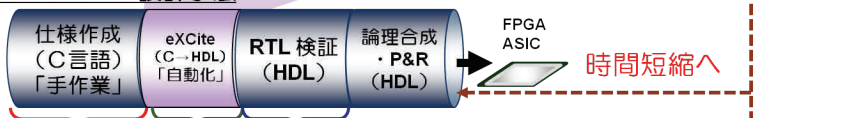
アルゴリズム名	C言語 行数	RTL HDL 行数	対比率
FIR フィルタ	140	400	2.9
Median フィルタ	50	2300	46.0
ラブラシアンフィルタ	60	2100	35.0
RSA	300	2400	8.0
ガウス乗算	45	3000	66.0
経路探索	1200	7000	5.8

図1: C言語とRTL HDLの行数比較

#### ● 旧来の設計手法



#### ● eXCite HW 設計手法



#### ● eXCite によるメリット

- 複数のアルゴリズムを検討
- HW/SW 分割の方針検討
- テスト & 性能測定環境生成
- プラグアンドプレイ
- 短期に設計
- 複数のアーキテクチャを検討

『決め打ち設計』からの脱却!!

図2: 設計工数比較

## ソリトンシステムズのC言語設計ソリューション

ソリトンシステムズでは、弊社開発製品であるC言語ハードウェア設計ツールeXCiteを用いた次の様なビジネスモデルをご提供しております。

- ・ C言語設計環境の販売
- ・ カスタムHW/SW分割検討環境構築
- ・ プロトタイプボードへの実装支援
- ・ 顧客アルゴリズムのハードウェア化

例えば、お客様がハードウェア化を検討しているC言語で記述されたアルゴリズムをお持ちの場合には、製品を販売するだけでなく、最適なハードウェアを設計するための設計支援もご提供しております。

また、弊社パートナー企業の各種FPGAボード向けの専用ライブラリの提供や設計支援もご提供可能です。

さらに、お客様がハードウェア、ソフトウェアの分割検討ご検討されている場合は、アルゴリズムの性能解析からハードウェアへの実装までを支援するプログラムもご用意しております。

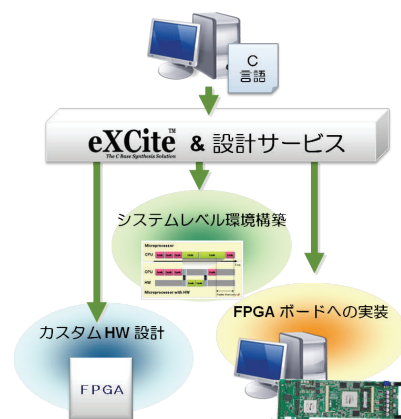


図3：ソリトンシステムズのビジネスモデル

## C言語ハードウェア設計環境eXCite

C言語ハードウェア設計環境eXCiteは、順次処理のC記述から並列性を自動抽出するために「高位合成」エンジンを搭載しております。また、C言語の変数やポインタによる値のやり取りに対して、バスやメモリ、FIFOなどのハードウェアとしてのインターフェースを割り付けることができる「通信合成」技術や、浮動小数点演算器や各種演算部品を選択し利用することができる「IP再利用」機能を搭載しております。（図4参照）

さらに、HDLシミュレータやFPGAベンダ提供の論理合成ツール等と連携するツール連携も搭載しております。

また、出力結果がソフトウェアと同じ処理をするか、またどの様な性能かを確認することのできる「自動テストベンチ生成機能」なども搭載されており、出力HDLの為に手設計で検証環境を用意するなどの手間を省くことができます。

なお、eXCiteの入力はANSI-Cで出力はRTL VHDL/Verilogです。アルテラ社、ザイリンクス社の各種デバイスに対応しております。また、アルテラ社Avalonインターフェース等のFPGAデバイス特有のインターフェースに接続できます。



図4：eXCiteの主な機能

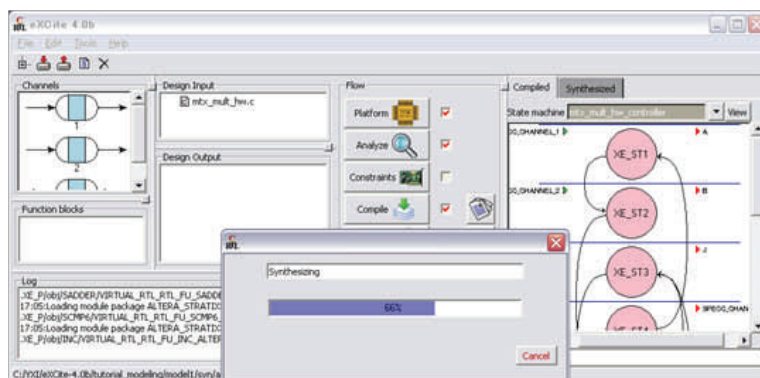


図5：C言語高位合成ツールeXCite画面

## eXCiteの主な用途

C言語ハードウェア設計ツールは、ハードウェア設計者だけのためも者と考えられがちですが、標準C言語を入力とするeXCiteは多くのお客様に様々な局面でご利用いただいております。主な分野は以下の通りです。

- ・システム設計  
(アーキテクチャ探索)
- ・ハードウェア設計  
(短期HW&IP設計)
- ・ソフトウェア設計  
(プロトタイプ設計)

■ システム設計において  
複雑化している組み込み設計において、HW化によるシステム全体のパフォーマンスアップは魅力のあるものです。しかしながら、マイコンを使うのと違いHWを簡単に実装し、性能まで把握するものは容易ではありません。HWを実装するにはRTL HDLでの設計が必須となるのは勿論、SW上で動作しているものをHW化するには膨大な工数とノウハウが必要とされます。eXCiteは、マイコン向けに用意しているC言語アルゴリズムを殆ど変えることなく

HWへと実装することが可能です。また、eXCiteが提供するC言語用通信関数により、HW/SWの分割時の通信ボトルネック等をC言語上である程度確認することができます。

- システム設計での主な効果
  - ・ SW・HW 処理性能を直接比較検討
  - ・ 決め打ちのHW・SW分割からの脱却
  - ・ 低コスト、短期間でのHW設計
  - ・ 大幅な手戻り防止
  - ・ RTL 設計コストの削減

■ ハードウェア設計において  
C言語からHWを設計するのは手間の掛かるものです。一方で、RTL 記述においては、設計の仕様変更による修正は簡単なものではありません。設計のゴールデンモデルをC言語で持ち、C言語高位合成ソリューションを用いることで、様々な設計仕様の変更や性能の改善などを容易かつ短期に行うことができます。また、多くの算術演算を整数演算は勿論、浮動小数点演算も、パイプライン動作をするIPとして実装することが可能です。

- ハードウェア設計での主な効果
  - ・ 手設計の数十倍早く、HWを設計
  - ・ 面倒な仕様変更にも容易に対応
  - ・ 面積、スピードのトレードオフを考慮
  - ・ C言語からIP設計

■ ソフトウェア設計において  
今までの組み込み設計では、SWデバッグのためのプロトタイプの完成に膨大な時間を割いていました。C言語でHWを設計することが可能になることで、プロトタイプを早期に手に入れることが出来るだけでなく、SW実装を検討していたタスクをHW化するのも容易に行えます。これにより、今まで以上に早期に本格的なSWデバッグが可能になります。

- ソフトウェア設計での主な効果
  - ・ プロトタイプボード待ちを短縮
  - ・ システム仕様変更を柔軟に対応
  - ・ ボトルネックを容易にHWへ

## マイコン側から見たハードウェアの魅力

組み込みシステムにおいてマイコンだけで処理ができればコスト面、開発期間など様々な面において有利ではありますが、実際の組み込みシステムでは、いくつかの処理（タスク）がボトルネックとなることがあります。そのような場合にハードウェアを導入することで、全体のパフォーマンスアップを行うことができます。多くの場合、ハードウェア対象となるものは、繰り返し多くの演算を行う処理や、連続的に外部からの信号をモニタし、その内容を計算し続ける様なものが対象となることが多いとされています。例えば、カメラから入ってくる画像情報を演算し、その結果に基づいてアクションを起こす場合などは、画像入力から計算結果までをハードウェア化することで、マイコンの処理を大幅に軽減することができます。C言語によるハードウェア設計技術はこのようなソフトウェア処理のボトルネック解消に有効なソリューションです。

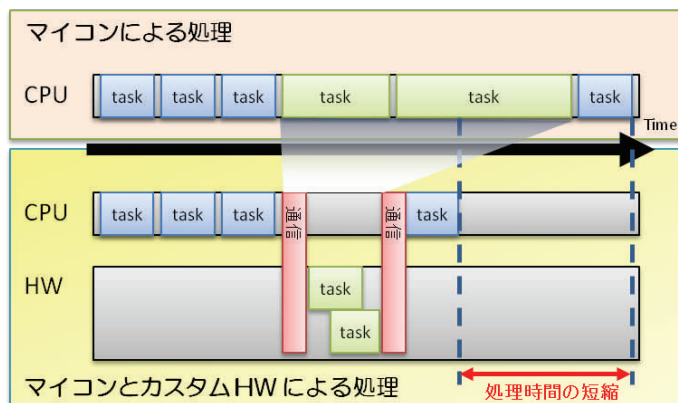


図6：HW導入の効果

## eXCite のデザインフロー

eXCiteのデザインフローは図7ようになります。eXCiteを用いた設計では、C言語レベル、高位合成レベル、実装レベルに分けて設計を行います。これらの各ステップでeXCiteは、便利な開発環境を提供しています。これらにより、設計者は必要十分な性能のHWを簡単に設計することができます。

### ■ C言語レベル

ハードウェア設計のスタートは、Cコンパイラ (Microsoft社 Visual C++ や gcc) でコンパイルし実行できる標準C言語が準備できたところからスタートします。設計者は、ハードウェア化を検討する部分をeXCiteが提供する通信関数 (API) を用いて切り分けを行います。これにより、ハードウェア設計対象部とそれ以外の部分を明示的に分離して取り扱うことができます。なお、通信関数は後のeXCiteでの合成により、各種バス、メモリ、FIFO、レジスタなどのインターフェースに割りつけることができます。さらに、C言語レベルで通信量の確認を行うこともできます。なお、分割したC検証において、高位合成後に行うRTL HDL 検証で使用されるテストベクタを作成することができます。C言語上で必要十分なテストベクタを作成することで、合成結果を十分に検証することができます。

### ■ 高位合成レベル

C言語からハードウェア設計者の殆どが利用しているRTL HDL に変換 (合成) するのに、eXCiteを利用します。eXCite の入力は標準C言語であるため、SystemC やその他特殊なC言語のようにC記述そのものを大きく書き換える必要は一切ありません。また、ハードウェア的な表現を行いたい場合でも、GUI上のオプション設定やC言語記述中にコメントとして合成指示子を設定することができます。

eXCite の出力は、RTL VHDL または Verilog になります。また、合成したRTLを

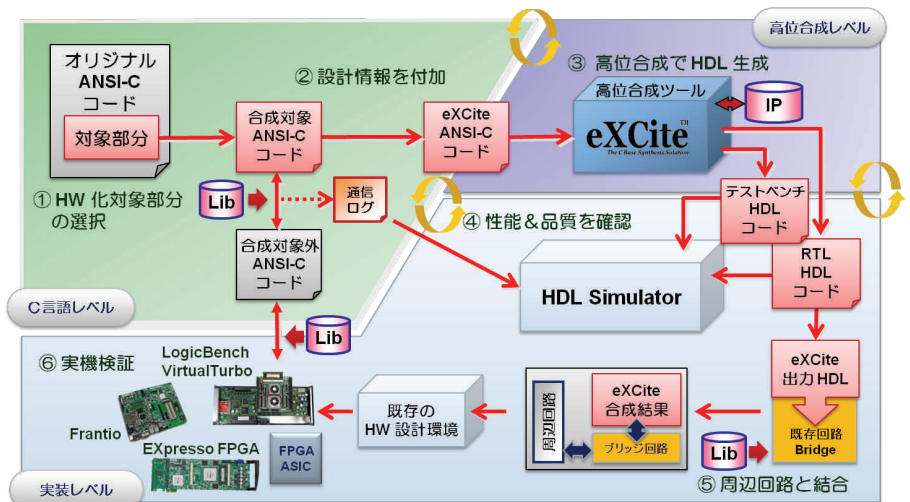


図7: eXCite のデザインフロー

検証するためのテストベンチも同時に自動生成します。これらのファイルは、市販のHDLシミュレータで利用できます。さらに、合成結果の情報をまとめたレポートファイルも生成されますので、Cアルゴリズムがどのようなアーキテクチャとして実装されたかも確認できます。

高位合成の制約としては、使用するテクノロジー、クロック周波数は勿論、演算器数やその種類、さらにその部品の共有化設定を細かく指示することが可能です。また、使用するライブラリの変更で、FPGAベンダの各種IPも利用可能になります。

高位合成レベルでは、高位合成処理の他に通信合成やテスト環境の生成等も同時に行われます。設計者は合成された結果を確認の上、必要に応じて様々なアーキテクチャを生成し検証することが可能になります。

通信合成と高位合成を同時に行うことで、実装する先のFPGAボードなど、各種アーキテクチャへ容易に接続することができる出力を行えます。

### ■ 実装レベル

実装レベルでは、合成されたRTL HDLおよびテストベンチ(HDL)を用いて、市販のHDLシミュレータで性能確認や動作検証を行うことができます。また、合成の対象が浮動小数点であれば、PC (Windows) 上ではIEEE754 企画以上のビット幅で計算との誤差を確認することができます。

eXCite がサポートしているいくつかのFPGAボードを用いれば、PC (Windows) 上で動作しているソフトウェア部とeXCiteの出力結果を連動させることができます。現時点でのサポートFPGAボードは、日立情報通信エンジニアリング株式会社製VirtualTurbo/LogicBench、オムロン株式会社製Frantio、ソリトンシステムズ社製EXpresso FPGAシリーズがあります。また、ソリトンシステムズの設計支援サービスにより、お客様がご指定のFPGAボードでも、同様の環境を構築することが可能です。



## 多彩な最適化技術

eXCite には様々な高位合成技術が搭載されています。これにより、単純にスケジューリングした場合に比べて性能を大幅に上げることが可能です。例えば、符号ビットが存在しない変数の乗算であれば、演算コストの削減最適化により、シフトと加算に変換されます。このような基本最適化技術やパイプライン合成、ビット最適化、Loop 最適化なども搭載されています。設計者は、これらの技術を自動または、ツールオプションを指定することで利用することができます。

```
unsigned int y, a;  
...  
y = a * 5;
```

➡  $y = a \ll 2 + a;$

**強度の低減**

図 8：乗算の最適化例

## 通信合成

eXCiteの通信合成は、合成C記述中で合成指示子（C言語としてはコメント扱いの記述がeXCiteではコマンドとして認識される特別なコメント記述）によって指定された変数に対して、eXCite合成ステップの合成制約で割り付けた様々なインターフェースを割り付けることが可能です。

図9ではアルテラ社Avalonインターフェースを割り付けた例です。設計者は、ツール上で指定した変数に対してAvalonインターフェースを割り付けることができます。バスの割り付けの際は、バスの仕様などを理解出来なくてもAvalonを持った回路を生成することができます。なお、Avalonインターフェースを持った合成回路はアルテラ社SOPC BuilderやQsys Toolを用いて周辺回路と接続することができます。これによりチップ全体を設計する際にも簡単に行うことが可能になります。

eXCiteが対応している主なインターフェースは、各種バス、FIFO、レジスタ、メモリなどがあり、ハードウェアにおける殆どのインターフェースをメニューから選ぶだけで割り付けることが可能です。また、特定のFPGAボードへHDL設計レスで実装するための専用周辺回路などを受託設計として承っております。

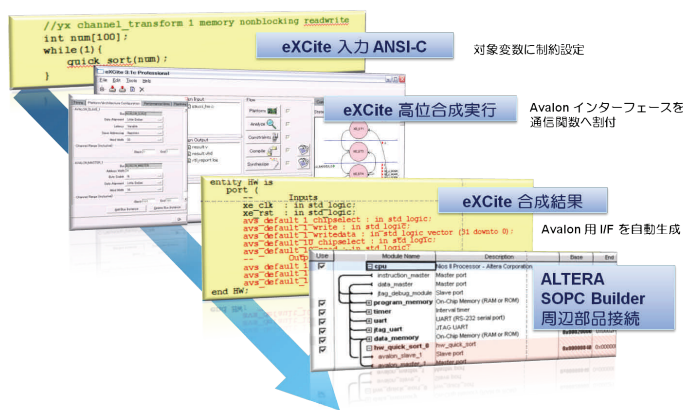


図 9：eXCite通信合成イメージ

## IP再利用

eXCiteは算術演算器、浮動小数点演算器やパイプライン演算器、ストレージのIPを標準で搭載しています。また、お客様のお持ちのIP（演算系、ストレージ系、バスなど）を有償にて登録するサービスもあります。eXCiteのIPデータベースは、C言語の関数やオペレータ等により再利用できます。

## ASIC利用

eXCiteは、FPGAは勿論 ASICへも対応が可能です。ASIC 向けに合成する場合は、各社ASICベンダのテクノロジファイル（.db や .lib ファイル）と利用する論理合成ツール（Synopsys Design Compiler、Cadence BuildGates/RTL Compiler など）があれば、eXCite 向きのテクノロジライブラリを自動生成することができます。また、.dbファイルをご提供頂ければ有償にて専用ライブラリをご提供することができます。

## テストベンチ自動生成と性能比較

eXCiteが自動生成するテストベンチは図8にあるように、ソフトウェアで動作させた内容でHDL検証を行うことができます。結果としてソフトウェア側の性能とハードウェア側の性能を比較することができます。

図9はラベリング処理をインテル社Xeonプロセッサ2.8GHzとアルテラ社Stratix IIIを70MHzで動作させた場合とで比較した例です。ソフトウェア処理はRDTSCコマンドを用いることである程度正確に処理サイクル数を測ることができます。また、ハードウェア側はeXCiteが生成するテストベンチにサイクル数を測る仕掛けが入っており、HDLシミュレータで実行するだけで処理サイクル数を測定できます。処理時間を比較したい場合は、サイクル数と動作周波数の逆数を掛けることでそれぞれの時間を算出することができます。

この例ではFPGA側が約8.7倍高速に動作することが確認できています。このように動作周波数がCPUの1/40と低速のFPGAでも並列化によりソフトウェアよりも高速化出来ていることが確認できます。

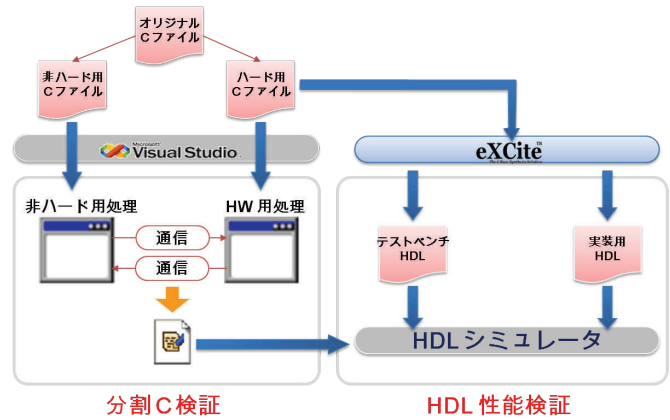


図8：テストベンチ利用イメージ



処理の内容：  
ノイズ除去、二値化、鋭鋭化、ラベル処理、ファイルフォーマット処理

■ PC上ソフトウェア処理時間を測定 (RDTSCコマンドで測定)  
プロセッサ名: Intel 社 Xeon 2.8GHz

```
Number of labels : 4
Number of labels : 4
Number of labels : 4
消費 CPU サイクル数 == 946600214
output file : out.bmp
```

946600214 (clock) x 1000/2400 (ns / clock)  
= 394416756 (ns) ≒ 349 (ms)

■ C言語合成結果の処理時間を計算 (自動生成されたテスト環境で測定)  
FPGA名: ALTERA社 Stratix III 70MHz

```
# Time:
# Time: Requested Clock
# Time: Estimated Cycle
# End c
# Cycle time : 100 ns
# Run time of dut : 317791100 ns
# Run cycle of dut : 3177911
# 0 differences found. ( 49154 tested. )
```

3177911 (clock) x 1000/110 (ns / clock)  
= 45398729 (ns) ≒ 45.4 (ms)

**8.7倍 FPGA側が高速**

図9：性能比較例

## eXCite製品構成

eXCiteにはProfessional版とFPGA版があります。Professional版はeXCiteの持っている全機能を搭載し、FPGA版は最適化機能、ライセンス形態、動作環境、ASICへの応用等に制限を設けて、その分価格をお安くしたものです。

販売形態としては、永久ライセンスやタームライセンスなどがあります。また、教育用にはアカデミック特価もご用意しております。

さらに、コンサルテーションや設計支援のプログラムもご用意しております。

製品名	eXCite Professional	eXCite FPGA
入力言語	ISO/ANSI-C	
出力言語	VHDL、Verilog (1995、2001)、SystemC (RTL)	
最適化処理	各種最適化エンジン、ビット最適化エンジン、バイブライシオン、最適化古風エンジン、変数2ポート処理	デザインバイブライシオンを除く各種最適化処理
対応テクノロジー	FPGA、ASIC ※2	FPGA
テスト環境生成	テストベンチ生成機能搭載	
対応 OS ※1	Windows、Linux	Windows
対応 FPGA ボード	LogicBench/VirtualTurbo、EXpresso シリーズ、Frantio	
ライブラリ	標準演算パッケージ、標準通信パッケージ、IPT パッケージ Avalon パッケージ、LogicBench パッケージ、EXpresso FPGA パッケージ	
必要コンパイラ	gcc2.8.1以降 (Linux 版)、 Visual Studio [2003/2005/2008/2010] (Windows 版)	Visual Studio [2003/2005/2008/2010] (Windows 版)
ライセンス	フローティング	ノードロック
推奨作業メモリ・HDD	500MByte以上の作業メモリ、150MByte インストールスペース	

図10：eXCite製品構成

株式会社ソリトンシステムズ  
エコデバイス事業本部 組み込みシステム部  
〒160-0022 東京都新宿区新宿 2-4-3

電話 03-5360-3851 FAX 03-5360-3888  
Email: at@soliton.co.jp