

大規模 SoC の高位設計への取組み R-CUBE

R-CUBE: Toward a High-Level Design Methodology for Large-Scale SoCs

西尾 誠一 相原 雅己 田中 利一

■ NISHIO Seichi ■ AIHARA Masami ■ TANAKA Toshikazu

大規模 SoC (System on a Chip) の開発効率を抜本的に向上させ、開発の期間と工数をともに 1/2 以下とするため、C 言語をベースとした高位設計・検証環境を構築し、実際の製品開発に適用するプロジェクト R-CUBE を推進している。

C 言語で記述したハードウェア (HW) のモデルを開発することにより、SoC のシミュレーション速度を従来の RTL (Register Transfer Level) と比較して約 1,000 倍高速化することができ、試作チップ完成前からソフトウェア (SW) の検証とデバッグを行うことが可能となった。この結果、試作チップ完成後の SW 検証期間を従来の 1/3 に短縮することができた。

Toshiba is actively engaged in the R-CUBE project, the target of which is to dramatically improve the work efficiency of system on chip (SoC) development and reduce the time and human resources required by half through a paradigm shift from register transfer level (RTL)-based design to C language-based design.

We have already constructed a C language-based design methodology and applied it to actual SoC design. We have successfully developed C language-based models of a SoC for digital TVs and demonstrated a simulation speed 1,000 times faster than that of conventional RTL-based models. This has made it possible for SoC software designers to verify and debug their software before an engineering sample chip is fabricated. Consequently, the software verification period after completion of the sample chip has been reduced to one-third of the conventionally required period.

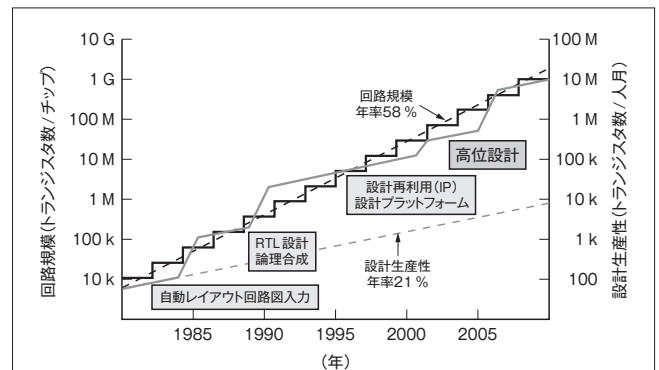
1 まえがき

東芝は、大規模 SoC の開発効率を抜本的に向上させ、開発の期間と工数をともに 1/2 以下とすることを目標として、R-CUBE と呼ぶプロジェクトを 2005 年度から推進している。

R-CUBE では、C 言語レベルの高位設計・検証に関する設計環境の構築とその製品開発への適用、設計コンテンツのプラットフォーム化 (IP (Intellectual Property) 再利用) による効率向上、SW 開発効率の向上、ドキュメント提供効率の向上など、大規模 SoC の開発効率向上にかかわる広範な領域をカバーしている。ここでは、HW 開発効率を向上させる高位設計・検証環境と SW 開発効率向上の施策に焦点を当てて述べる。

2 R-CUBE の基本コンセプト

CAD ツールの改善やその実行環境としての計算機の能力向上による設計生産性の向上は、年率 21 % 程度と言われている。一方、半導体の微細加工技術の進歩により、設計対象となる SoC の回路規模や複雑度は、年率 58 % で増加している。このため、集積可能な回路規模と設計可能な回路規模の間には、大きなギャップが生じる (図 1)。このギャップを埋めるためには、設計手法を大きく改革することが必要となる。過去においては、自動レイアウトによる設計下流工程の自動化や RTL 設計の実現などにより、このギャップを埋める努力がな



出典：SEMATEC 予測

図 1. 設計パラダイムシフトによる設計生産性の向上 — 集積可能な回路規模と設計可能な回路規模の間のギャップを埋めるために、高位設計の実現が望まれている。

Improvement of design efficiency by design paradigm shift

されてきた。しかし、近年の SoC 開発では再度ギャップの拡大が顕著となってきており、開発期間やコストの増大、検証不足による作り直しの発生などの課題に直面している。これらの課題を解決する新たな設計改革として、RTL より抽象度の高い C 言語レベルから設計を開始する高位設計の実現が期待されている。

R-CUBE の基本コンセプトは、次の三つの R により大規模 SoC の開発実行力を強化することである。

- (1) Revolution 高位設計の実現による HW/SW 設計メ

ソドロロジーや設計プロセスの改革

- (2) Reuse 設計資産のC言語モデル化に加え、SWの標準化やHWとプラットフォームの統一などにより、設計資産の再利用を容易化
- (3) Reduce 上記(1), (2)を実際のSoC開発に活用することにより、開発の期間と工数を削減

3 HWの高位設計・検証環境

従来の設計フローでは、HW設計者は、仕様からRTLによる設計を行い、機能を検証する。検証完了後、論理合成ツールによりRTL記述からネットリストを生成し、レイアウト設計及びSoC試作へと進む。そして、試作されたSoCを組み込んだ評価用ボードが完成して、ようやくSWの検証が開始できる。このように、従来の設計フローではHW開発とSW検証は逐次的に行わざるをえないため、SoCの大規模・複雑化に伴って設計期間が長期化している。また、SWの検証が始まって、初めてHWとSWの間の不整合が見つかり、最悪の場合、HWを作り直さなければならないこともある。これに対して高位設計では、設計者は、仕様からRTLよりも抽象度が高いC言語ベースの記述言語を用いてHWの動作を記述（以下、C記述と略記）する。C記述を用いることで、RTLに比べシミュレーション速度を約1,000倍高速化できるため、HW/SW協調シミュレーションにより、SoCの試作チップが完成する前からSWの検証を開始できる。従来、設計者が作成していたRTL記述は、高位合成ツールを用いることによってC記述から自動合成できる。その際、制約条件として与える動作周波数、回路サイズ、及び実行サイクル数を変更することにより、設計に応じた最適なRTLを、C記述を書き換えることなく合成できる。そのため設計者は、仕様どおりの機能を設計するという、本質的な作業に能力を集中させることが可能となる（図2）。

R-CUBEで構築しているC言語ベースの設計フローは高位検証環境、HW/SW協調検証環境、高位合成環境、及び全体検証環境

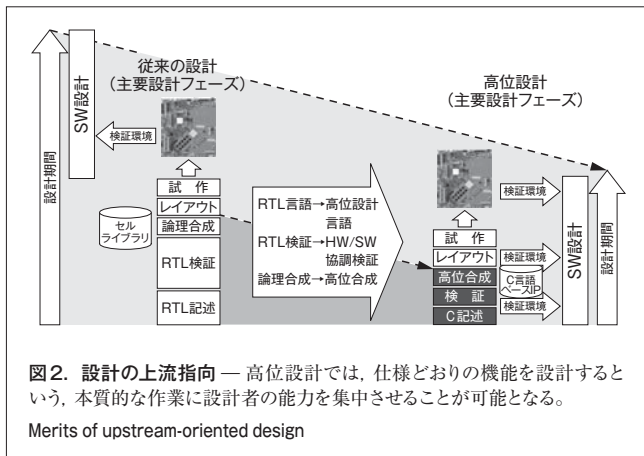


図2. 設計の上流指向 — 高位設計では、仕様どおりの機能を設計するという、本質的な作業に設計者の能力を集中させることが可能となる。
Merits of upstream-oriented design

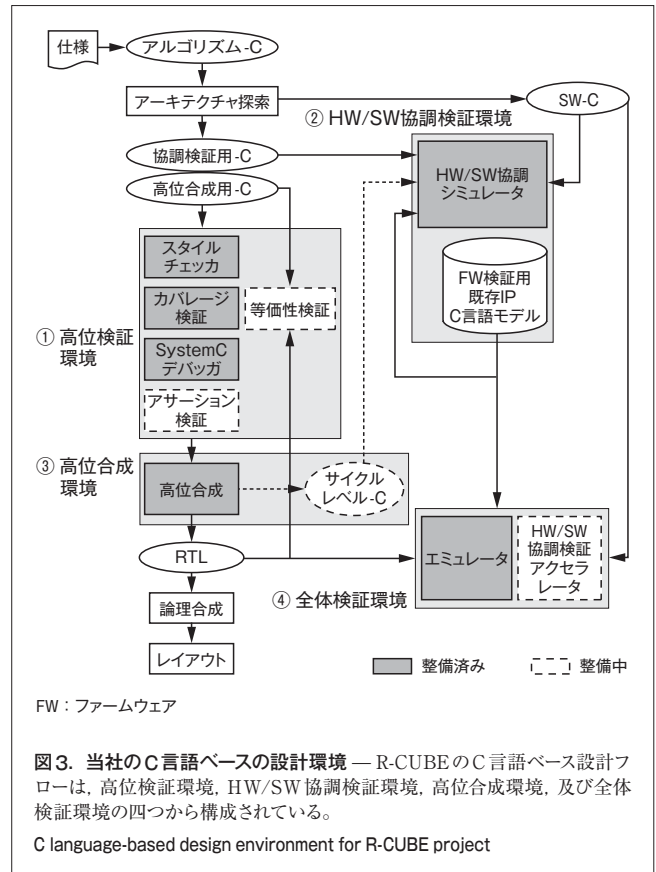


図3. 当社のC言語ベースの設計環境 — R-CUBEのC言語ベース設計フローは、高位検証環境、HW/SW協調検証環境、高位合成環境、及び全体検証環境の四つから構成されている。

C language-based design environment for R-CUBE project

検証環境の四つから構成されている（図3）。高位検証環境では、仕様に基づいて作成したC記述を検証し、誤りのないC記述とする。ここでは、現在普及しているRTL記述の検証環境と同様に充実した検証環境が望まれる。しかし、現時点では等価性検証とアサーション検証^(注1)の技術がまだ確立されていないため、シミュレーションを主体とした検証環境となっている。カバレッジ検証^(注2)ツールに関しては、シミュレーションによる検証のカバレッジを測定することで検証漏れを防止できる環境を自社開発した。検証が完了したC記述は、高位合成環境により要求される回路サイズと性能を満たすRTL記述に変換される。上述したように、合成時に与える制約条件を変えることで所望のRTL記述が合成できるのが理想であるが、現実的には、まだ高位合成技術が発展途上であるため、C記述自体の書き直しが必要な場合もある。高位合成と並行してHW/SW協調検証を行い、HWとSWの整合性の確認とSW検証を行う。C記述とプロセッサのモデルを統合してシミュレーションモデルを作成し、このモデルにSWをロードしてシミュレーションを実行することで協調検証を実現してい

(注1) アサーション (assertion) 検証とは、本来論理回路が成立すべき条件 (アサーション) を表現した記述を用いて、回路記述の正当性を検証する手法のこと。

(注2) カバレッジ (coverage) とは、テストの網羅率を示したもので、ソースコードや条件分岐などのテスト網羅率を示す。カバレッジ検証とは、記述の検証網羅率を確認し、網羅率が不足している場合はテストパターンを増やして検証網羅率を上げる検証手法のこと。

る。最後に、すべてのRTL記述によるSoC全体としての検証をエミュレータなどを用いて実施し、C言語ベース設計が完了する。これ以降のフローは従来と同様である。

4 SW開発改革

4.1 背景とアプローチ

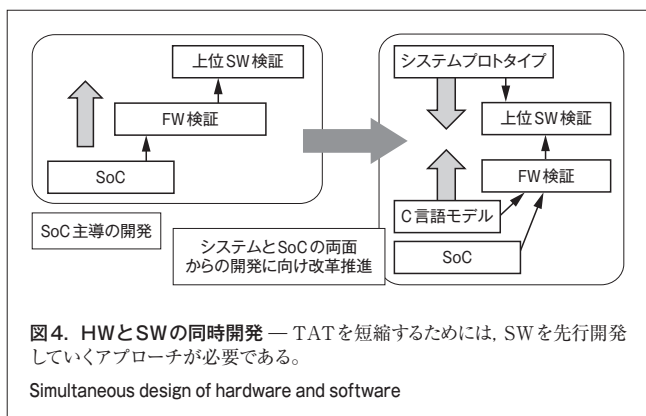
当社のSoC製品の搭載対象は、デジタルテレビ(TV)、携帯電話、オーディオプレーヤなど多岐にわたるが、一方で、携帯電話でTV放送を受信する、TVでハードディスクに番組を録画するなど、機能統合が進展してきている。また、高品質と短納期の両立が要求されるようになってきている。

このような動向に対して、R-CUBEにおけるSW開発改革活動として以下のアプローチをとっている。

- (1) SWをSoC試作チップに先立って開発
- (2) 製品間にまたがったSW体系を準備

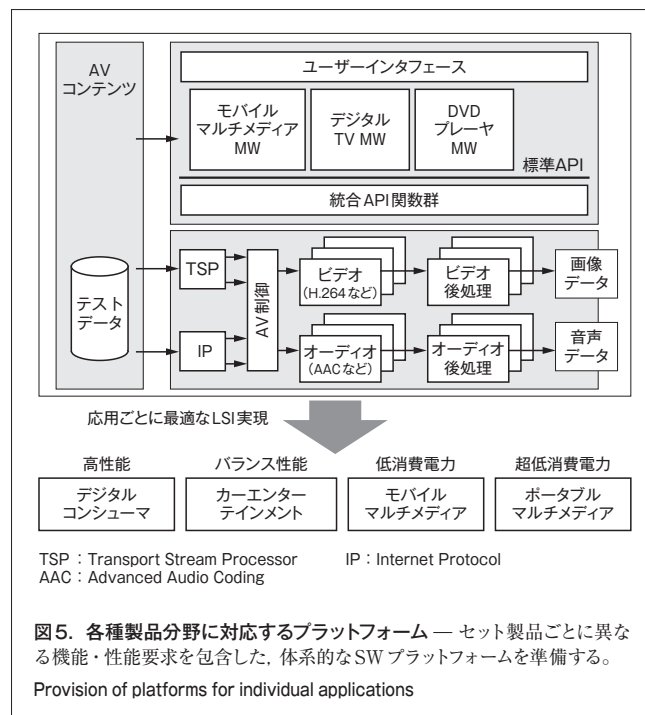
従来、SWの検証及びデバッグは、試作チップの完成後に、これを搭載した評価用ボードを用いて行われていた。C記述による協調検証技術により、SoCの実機が完成する前に、C記述とSWの組合せ検証が可能になり、開発期間(TAT:ターンアラウンドタイム)の大幅短縮を達成した。

しかし、更にTATを短縮するには、SoCのHW仕様が決まってから、それに合わせて後工程としてSWを開発するのではなく、SWを先行開発していくアプローチが必要である(図4)。



そのためには、セット製品ごとに異なる機能・性能要求を包含した、体系的なSWプラットフォームをあらかじめ準備しておく必要がある。その構成として、チップによるオーディオ及びビデオ(AV)の処理モジュール群と、製品の基本機能を実現するミドルウェア(MW)層の間に共通API(Application Programming Interface: プログラム開発のためのインタフェース)を設ける構成を提案する(図5)。

このような構成により、MW層ではセット製品間でプログラムの共通化や再利用を促進でき、開発工数の削減、TATの短縮、更には高品質を実現できる。一方、SoCに直結したモ



ジュール群は、消費電力や液晶画面の精細度、サイズなどの違いを反映した多様なチップ製品展開が可能となり、この中に半導体製造技術、信号処理技術などの面で差異化技術を投入していくことが可能となる。

現在までに、特定のSoCを意識しないSWプラットフォームをパソコン上で構築し、共通APIの妥当性を確認した。

4.2 SW開発プロセス・手法の改革

前述のSW主体のアプローチを実現させるには、SW開発人員の確保、開発プロジェクト管理力のアップ、及び開発プロセスの整備が必要である。また、複数の製品分野や複数の顧客に対応できる設計、数年先を見越した設計手法、及び品質確保と試験工数削減を両立させるためのSW検証技術など、エンジニアリング面の手法開発が必要になる。

開発人員の確保に向けては、グローバルな顧客対応とコストダウンを狙ってインドのTESI (Toshiba Embedded Software India) をはじめ開発拠点を展開している⁽¹⁾。

R-CUBE活動としては、海外開発拠点を含めた開発プロセスの整備、技術者の育成・評価・活用の体系化などを推進している。

エンジニアリング面では、検証工程の効率化のため、実用化目前となってきた形式検証技術の適用に積極的に取り組んでいる。当社は、組込みオペレーティングシステム(OS)の開発過程で発生したバグが、形式検証手法により除去できるかどうかを、実際に検証用モデルを構築してフィージビリティスタディを行い、形式検証手法の有効性を確認した。

なお、形式検証のために、本来の設計作業とは別にモデル構築作業が発生してしまうのは二度手間であり、工数削減の

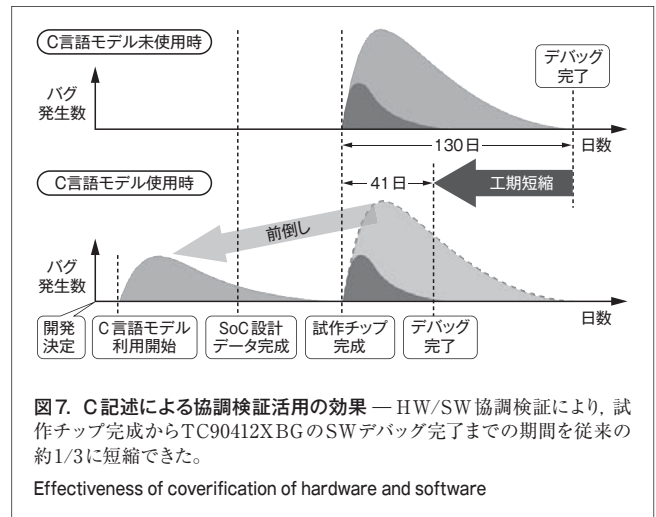
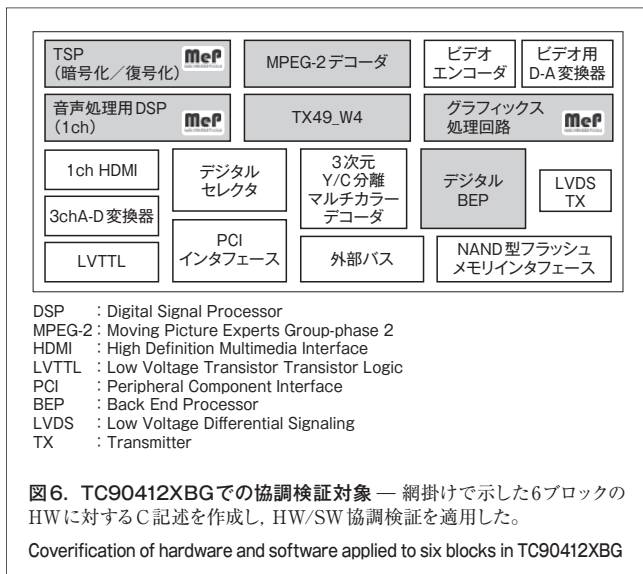
効果が半減してしまう。そこで、設計の記法としてUML (Unified Modeling Language) を採用し、UML 記述からのモデル生成手法を確立する方向で検討を行っている。また、統一的な記法を用いることで、仕様の不整合や設計の不備の発見が容易になると期待できる。

5 製品適用事例

これまで述べたR-CUBEの成果を、フラットパネルデジタルTV向けの大規模SoC (TC90412XBG) の開発に適用した。具体的には、C記述を用いてSoCの性能を見積もることによる内部接続バスの選定とアーキテクチャの決定、HW/SW協調検証環境によるチップ完成前からのSW検証、高位合成ツールによる新アルゴリズム回路のC記述からのRTL合成などの部分に適用した。以下に、チップ完成前からのSW検証について述べる。

従来、SWの検証及びデバッグは、試作チップを使って行われていた。その主な理由は、RTL設計で作成されるRTL記述を用いたシミュレーションでは、SWを効率よく検証するために必要なシミュレーション速度(実機速度の1/10~1/100程度)が得られないためである。C記述でモデル化することによりHW部分の記述の抽象度を上げることができ、シミュレーションを高速化できるため、HW/SW協調検証環境によるSWの検証及びデバッグが可能となった。

TC90412XBGでは、図6の網掛けで示したブロックのHWに対するC記述を作成し、そのブロック中に含まれるMeP (Media embedded Processor) コア²⁾のSWの検証を行った。その結果、図7に示すように、試作チップ完成からTC90412XBGのSWのデバッグ完了までの期間を従来の約1/3に大きく短縮できた。これは、試作チップ完成前に上記C言語モデルを使っ



て多数のバグを発見でき、試作チップ完成時にSWの完成度が高まっていたためである。

6 あとがき

当社が2005年度から推進しているR-CUBEについて、概要と適用事例とを述べた。

3章で述べたように、高位合成を用いて人手によるRTL設計に近い高品質の回路を合成するためには、現状では、HW/SW協調検証に用いるC記述とは別に、高位合成向けのC記述を別途作成しなければならない場合が多い。開発効率の観点からは、これらのC記述を一本化するのが望ましい。一つのC記述で高位合成と協調検証の両方に対応できるようにすることが今後の課題である。

文献

- (1) 篠原潤一, ほか. システムLSI用ソフトウェア開発のグローバル体制構築. 東芝レビュー. 61, 1, 2006, p.36-39.
- (2) 松井正貴. システムオンチップの普及とMeP. 東芝レビュー. 58, 5, 2003, p.2-8.



西尾 誠一 NISHIO Seichi

セミコンダクター社 システムLSI事業部 システムLSI設計技術部主幹。システムLSIの設計メソドロジー技術の開発に従事。電子情報通信学会会員。
System LSI Div.



相原 雅己 AIHARA Masami

セミコンダクター社 システムLSI事業部 システムLSI設計技術部参事。システムLSIの設計メソドロジー技術の開発に従事。
System LSI Div.



田中 利一 TANAKA Toshikazu

セミコンダクター社 システムLSI事業部 システム・ソフトウェア技術部グループ長。組込みSWの開発管理に従事。情報処理学会, 人工知能学会会員。
System LSI Div.