

主力の設計手法は10年ごとに変化、次はプラットフォーム・ベース

今井 正治



今井正治(いまい・まさはる)
大阪大学大学院 情報科学研究科 教授
エイシップ・ソリューションズ(株)
代表取締役 CTO

1979年、名古屋大学大学院 工学研究科情報工学専攻 博士後期課程修了(工学博士)。豊橋技術科学大学を経て1996年に大阪大学に異動。現在、大学院情報科学研究科情報システム工学専攻教授。2005年4月に大学発ベンチャーであるエイシップ・ソリューションズを設立し、代表取締役CTOに就任。多忙ながら、充実した毎日を送っている。

Design Wave Magazine 編集部のみなさま、創刊100号おめでとうございます。貴誌の読者のひとりとして、お祝い申し上げます。継続とは力なりと言いますが、1995年12月の創刊以来、10年にわたるたゆまぬ努力に敬服いたします。本誌は日本で初めて「電子システムの設計技術」を専門に取り扱った雑誌だと思いますが、日本の電子システムの設計者にとって本誌は新しい設計手法や応用技術を平易に紹介するという非常に重要な役割を果たして来られたと思います。

記念すべき創刊100号に寄稿させていただくという栄誉を与えていただき、ありがとうございます。この機会に、将来の設計技術や電子産業に関して日ごろ考えていることを述べさせていただきたいと思っております。

● 10年単位で設計手法が大きく変化

ご存じのように、半導体集積回路(メモリおよびロジック)の集積度は、18ヵ月ごとに2倍に増えてきました。この傾向は、半導体集積回路が製造され始めた1960年代以来、40年以上にわたって続いています。18ヵ月で2倍という増加率は、10年で100倍の増加率に相当します。過去40年の半導体集積回路の設計手法の変遷を調べると、**図1**、**表1**に示すように、ほぼ10年ごとに大きな変化が起きていることがわかります。とくに1980年代から1990年代に移ったときに、大きな変化(パラダイム・シフト)が起きました。すなわち、1980年代まではトランジスタや論理ゲートの回路図(図形)を直接編集して集積回路を設計していたのですが、1990年代に入って論理合成技術が成熟するにつれて、テキスト(HDL)を用いた設計手法が実用化されたのです(編集部注)。

1990年代後半から2000年代にかけての上流設計手法の変化は、高位合成(high level synthesis)の実用化です。高位合成技術はビヘイビア合成(behavioral synthesis)とも呼ばれ、1980年代から研究が行われていました。しかし、最適化アルゴリズムの計算量が多いこともあり、実用化に至るまでには長い期間がかかりました。その後、コンピュータの処理能力の向上とアルゴリズムのくふうにより、実用的な規模の設計にも適用可能となりました。いわゆる「C言語からの合成」と呼ばれている技術は、この高位合成技術に基づいています。

編集部注：本誌はこのころ、HDLへのパラダイム・シフトの波に乗って創刊された。創刊当初は隔月刊だった。本記事の筆者である今井正治氏には、創刊号の特集「キーパーソンに聞く電子技術の最前線」のインタビュー記事「第4章 ハードウェア・ソフトウェア・コデザインの実現」に登場していただいた。

表1 設計記述の変遷

年代	1960年代	1970年代	1980年代	1990年代	2000年代
集積度 (搭載トランジスタ数)	1 ~ 10 ²	10 ² ~ 10 ⁴	10 ⁴ ~ 10 ⁶	10 ⁶ ~ 10 ⁸	10 ⁸ ~ 10 ¹⁰
集積回路の呼びかた	IC	SSI, MSI	LSI, VLSI	VLSI, ULSI, SOC, SiP	VLSI, ULSI, SOC, SiP, MPSOC
設計抽象度	マスク・パターン	トランジスタ	論理ゲート	レジスタ・トランスファ(RT)	ビヘイビア
設計技術 (EDA ツール)の キーワード	座標値入力	レイアウト・ エディタ	回路図エディタ, 論理シミュレータ, 自動レイアウト	RTL記述(HDL), 論理合成, ハードウェア・ソフトウェア 協調設計, 設計検証	仕様記述(UML), ビヘイビア記述(C, C++, SystemC), 高位合成, 設計検証, 構成可変プロセッサ, 低消費電力化, DFM
設計業務	物理(レイアウト) 設計中心	物理設計中心	論理設計と物理 設計の分離	論理設計と物理設計の分離 から融合へ	システム設計と物理設計を考慮した 論理設計

DFM : design for manufacturability MSI : medium scale integrated circuit SSI : small scale integrated circuit
 HDL : hardware description language RTL : register transfer level ULSI : ultra large scale integrated circuit
 IC : integrated circuit SiP : system in package UML : unified modeling language
 LSI : large scale integrated circuit SOC : system on a chip VLSI : very large scale integrated circuit
 MPSOC : multi-processor SOC

● C言語からのハードウェア生成が今後の主流に

C言語は、本来は逐次的アルゴリズム(プログラム)を記述するための言語なので「システム設計者やソフトウェア設計者にも理解しやすい」、広く普及している言語なので「習得に要する期間が短い」、などの利点があります。しかし、次のような限界もあります。

- C言語の記述は逐次的に実行されることを仮定しているので、ハードウェアの持つ並列性や経過時間を表現できない
- ハードウェア設計で使われる階層構造を記述できない。また、部品のインスタンス化(instantiation)を明示的に記述できない
- C言語の特徴であるポインタや再帰呼び出し(recursive call)を用いた記述から効率の良いハードウェアを生成することが困難(ここで言う「効率の良いハードウェア」とは、面積が小さく、実行時間が短いハードウェア回路のこと)

これらの問題を解決するためには、さまざまなくふうが必要です。現在のC言語からの合成技術では、次のような方法で対処しています。いずれの場合も、合成を可能にするためのコーディング・ガイドラインを設定しています。

- 1) 言語に制約と拡張を加える。この手法は、SpecC(開発元は米国 University of California, Irvine)やBDL(NEC), Bach C(シャープ), Handel-C(英国 University of Oxford, 英国 Celoxica 社)などの言語で採用されている。
- 2) 拡張するべき機能を、C++言語のクラス・ライブラリ

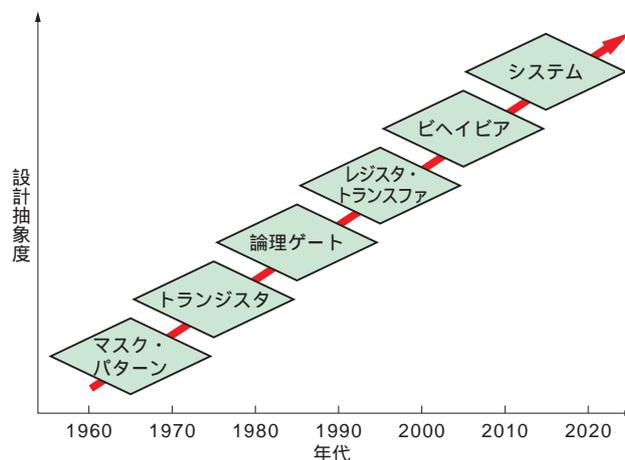


図1 設計記述の抽象度の変遷と将来予測

1960年代に集積回路が製造され始めてから、およそ10年ごとに設計手法が変遷を遂げてきた。なぜ10年ごとなのかは不明だが、設計者が細部まで把握して取り扱うことができる設計記述の量と関係がありそうだ。なお、2010年代に主流となる設計手法は、筆者の予想である。

として実装する。この方法は、SystemCで採用されている。

- 3) 処理系に対する指示(ディレクティブ)をコメントの形で記述する。

上記の1)や3)の方法を採用した場合、高位合成可能な言語のサブセットを用いて記述することになります。したがって、C言語で書かれた逐次的なアルゴリズムからハードウェアを生成すると考えるよりも、C言語の文法を用いて高位合成系の入力となるデータフローを記述していると考えたほうがわかりやすいかもしれません。高位合成技術を用いたビヘイビア記述からのハードウェアの生成は、今