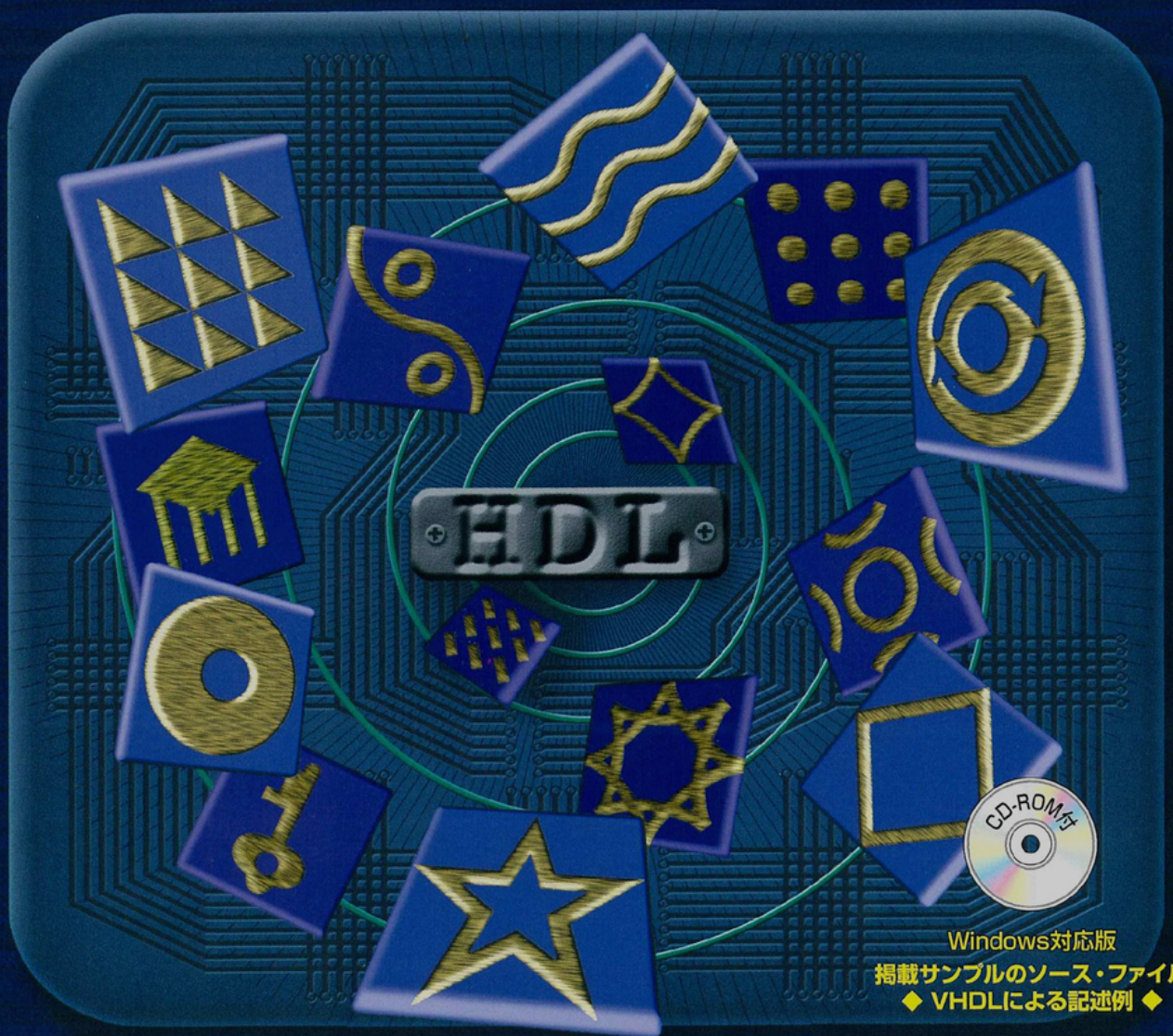
Design Wave BOOKS

HDLによる高性能ディジタル回路設計

ソフトウェア感覚を離れてハードウェアを意識する

森岡 澄夫 著



Windows対応版
掲載サンプルのソース・ファイル
◆ VHDLによる記述例 ◆

このＰＤＦは，ＣＱ出版社発売の「HDLによる高性能ディジタル回路設計ディジタル/CD-R版」の一部分の見本です．
内容・購入方法などにつきましては以下のホームページをご覧下さい．
http://shop.cqpub.co.jp/hanbai/books/I/I000025.htm

HDLによる高性能ディジタル回路設計

ソフトウェア感覚を離れてハードウェアを意識する

森岡　澄夫 著

CQ出版社

見本

●**本書記載の社名，製品名について** ― 本書に記載されている社名および製品名は，一般に開発メーカーの登録商標です．なお，本文中では™，®，©の各表示を明記していません．
●**本書掲載記事の利用についてのご注意** ― 本書掲載記事は著作権法により保護され，また工業所有権が確立されている場合があります．したがって，記事として掲載された技術情報をもとに製品化をするには，著作権者および工業所有権者の許可が必要です．また，掲載された技術情報を利用することにより発生した損害などに関して，CQ出版社および著作権者ならびに工業所有権者は責任を負いかねますのでご了承ください．
●**本書付属のCD-ROMについてのご注意** ― 本書付属のCD-ROMに収録したプログラムやデータなどは著作権法により保護されています．したがって，特別の表記がない限り，本書付属のCD-ROMの貸与または改変，個人で使用する場合を除いて複写複製(コピー)はできません．また，本書付属のCD-ROMに収録したプログラムやデータなどを利用することにより発生した損害などに関して，CQ出版社および著作権者は責任を負いかねますのでご了承ください．
●**本書に関するご質問について** ― 文章，数式などの記述上の不明点についてのご質問は，必ず往復はがきか返信用封筒を同封した封書でお願いいたします．ご質問は著者に回送し直接回答していただきますので，多少時間がかかります．また，本書の記載範囲を越えるご質問には応じられませんので，ご了承ください．

まえがき

本書では，これまでディジタル回路設計の入門書ではほとんど触れられてこなかった二つの重要なテーマを主題として扱います．

一つ目は，C言語などで書かれた複雑な処理を，どのような考えかたと手順で回路化するかを説明することです．

二つ目は，優れた性能の回路を設計するためには何に着目して，どんな工夫をすればよいか，ソフトウェアの工夫とは何が違うかを説明することです．

なぜこれらのテーマが重要なのでしょうか？

一つ目のテーマについては，現在の現場で作る回路が，入門書でよく説明されるカウンタやアダーなどとは，もうまったく比較にならないほど複雑な処理を行うからです．たとえば，画像処理，音声処理，暗号処理，あるいはプロトコル処理やパケット処理といったように，ソフトウェアでもするような処理を回路で作ることが多くなっています．そのような回路は，カウンタやアダーといった細かい部品を組み合わせるという発想では，とても作ることができません．ではどうやって作るのかは，現場で活躍している技術者達はもちろん理解しています．しかし，それが入門者向けに説明されることは，従来はほとんどありませんでした．

二つ目のテーマについては，ハードウェア記述言語(HDL)と回路自動設計ツール(EDAツール)が普及した結果，回路を知らずにソフトウェアのセンスでHDLを使ってしまい，非常に性能の悪い回路を作ったり，あるいはまったく動作しない回路を作ってしまう場合が増えているからです．1990年代にHDLが広く普及しはじめた頃，「回路の構造(回路図)を考えることはもう必要ない．HDLを使えば人は回路動作だけを考えればよい」と言われたものです．しかし，実際にはそううまくいきませんでした．結局，良い回路をどう作るかを知らなければ，HDLもツールも使いこなすことができないし，実用に耐える回路を作ることもできないのです．

それほど遠くない将来，C/C＋＋言語を使った設計へ移る可能性が高いでしょう．現在，「Cベース設計へ移行すれば誰でも早く回路を設計できるようになる」とも言われていますが，それはHDLでも同じでした．Cベース設計があまり普及していない段階では，HDLのときと同じく，開発期間の劇的な短縮が宣伝されもてはやされることでしょう．しかし，本当に誰もがCベース設計をするようになったとき，いかにハードウェアを意識した設計をするかで決定的な差がつくようになるはずです．その折に，本書で説明する知識が，今以上に役立つであろうと筆者は確信しています．

本書は，月刊「トランジスタ技術」誌の1998年8月号～1999年12月号に「HDLによる大規模ディジタル回路設計入門」として連載された記事をもとに，構成の見直しを行って加筆したものです．本書の構

成は次のようになっています．

第1章では回路設計の全体的な流れについて紹介します．

第2章～第5章で，ソフトウェア的に考えた処理を回路へ落とし込む基本手順を説明します．そこでの一番のポイントは，処理と回路との対応をつかみ，なぜ回路がうまく動くのかを知ることです(これは一般に想像されるほど難しくありません)．

第6章～第9章では，回路最適化の一般的な方法とその考えかたについて紹介します．

第10章では応用設計事例を示します．実用回路と比べれば簡単なものですが，設計センスは同じです．

各章を順番に読み進める必要はなく，気が向いたところから進んでください．回路設計現場の雰囲気を感じとっていただければ幸いです．また，本書の内容でわかりにくい箇所や誤りがありましたら，ぜひご一報いただきたく思います．

最後に，本書の編集に多大なるご尽力をいただいたCQ出版(株)の清水当氏，寺前裕司氏，蒲生良治氏の各氏と，連日仕事に熱中している筆者をサポートし続けてくれた妻の和香子に，深く感謝いたします．

2002年晩秋　著者

目　次

本文イラスト：神崎真理子

CQ出版社

第1章
ディジタル回路設計の全体的な流れ
HDLによる設計の概要と本書の構成

この章では，HDLを使ったディジタル回路設計の全体像をつかんでいただきながら，本書の構成を紹介しておきたいと思います．

1-1 本書のねらい

● ねらいは現在の回路設計のセンスをつかむこと

本書のゴールは「プロの卵として第一歩を踏み出せる」というレベルです．難しいと思われるかもしれませんが，適切な内容を選んで学習を進めれば，さほどでもありません．

本書の文章を第1章から読むだけでは退屈なはずですので，出てくる回路を実際にCAD(メーカのWebサイトから入手できる)にかけたりして体で覚えていけば，きっと飽きることなくスムーズに学習が進むでしょう．もしわからない箇所にぶつかったときは，そこで止まらずに，飛ばしてどんどん先に進んでください．

● 従来の入門書と本書とのスタンスの違い

本書は，従来のディジタル回路の入門書と比べて，次の①②で大きくスタンスを変えました．

▶ その①：ゲート・レベルでの回路作成法の説明は必要最低限に抑えました．

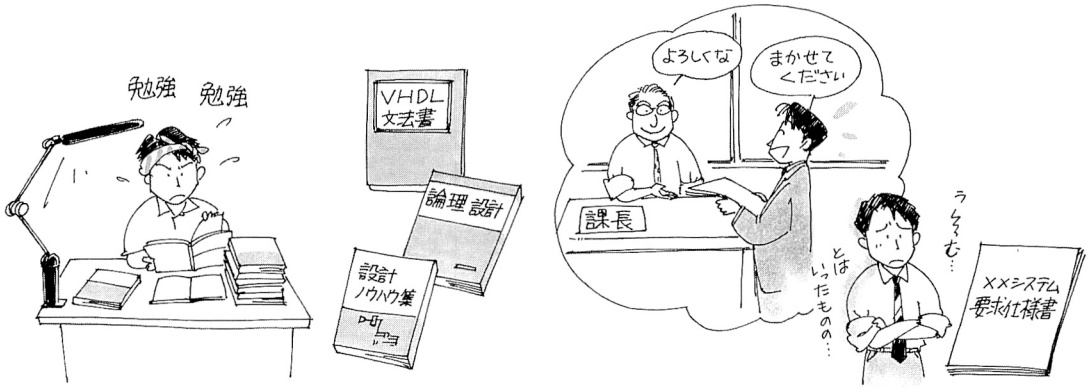
具体的には，カルノー・マップやクワイン・マクラスキー法などの論理圧縮法や，ステート・マシンをゲート・レベルの回路に直す方法などを省略し，その代わりハザードや非同期入力の扱いなどにページを割きました．

また，初学者がゲート・レベル設計から学習を始めてしまうと，現在実際に行われている回路設計の説明に到達するまで非常に長い道のりとなってしまいます．本書のスタンスは，たとえばJavaプログラミングをマスタしようとしたとき，2進数→アセンブラ→FORTRAN→C→…というように歴史に従って下から勉強を進めるのではなく，まずJavaから入って必要に応じて下へ降りる，というものです．

▶ その②：良いハードウェアをどのような考えかたで作るかを説明します．

HDL(ハードウェア記述言語，詳しくは1-2節)を使って，どのような考えかたをして回路を作っていくのか，良い回路に仕上げるためにはどのようなアイデアを用いるのか，といった回路設計のセンスを説明します(**図1-1**)．

〈図1-1〉ディジタル回路設計で使う考えかたをつかもう



(a) 細かいノウハウはあれば
あるほど良いのだが…

(b) ノウハウの根本にあるものの考えかたを
知らないと実設計には手がつけられない

1-2　現在のディジタル回路の設計現場では

● 回路で行う処理が高度になってきた

設計者に要求されるのは，性能の優れた回路，つまり速くて小規模で消費電力の少ない回路を，スピーディにバグなしで作ることです．これは昔も今もまったく変わりません．

もっとも変わったのは，作らなければならない処理の内容が昔とは段違いに複雑になり，多様化してきたことです．とくに，ちょっとしたソフトウェアなみのデータ処理を回路で実現することが，ごく普通になってきました．たとえば，CPUコア，画像/音声処理，プロトコルの上位レイヤ処理，誤り訂正符号処理，暗号処理などです．FPGAメーカのホームページなどを見ると多くのIPコア(いわゆる回路部品のこと)が販売されていますが，それらの多くはこうしたデータ処理を行う回路です．いろいろな機械の制御シーケンサなど，昔から作られているような回路も多くありますが，それらもけっこう高度な処理をするようになってきています．

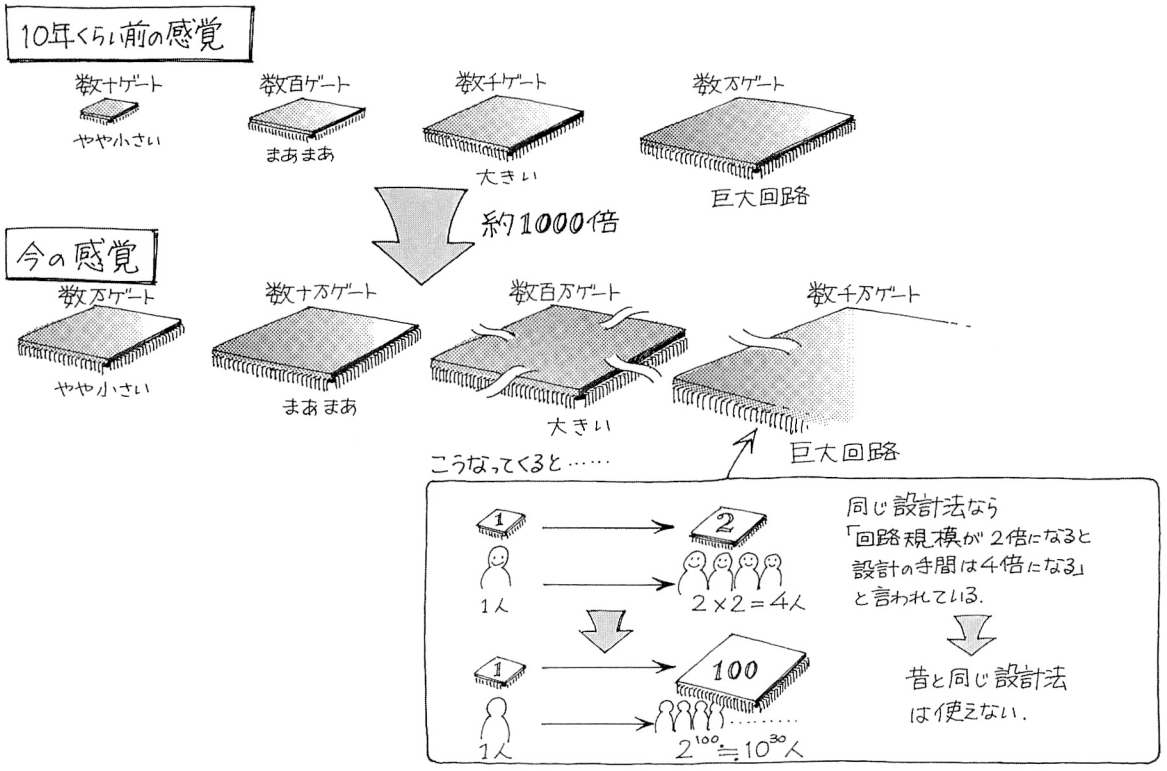
このような変化が起こったのは，デバイスの規模と速度が格段に進歩したからです．データ処理回路は，一つの処理を行うだけでも数万～数十万ゲートをあっという間に消費してしまうので，昔は作りたくても作れませんでした．今ではそれが可能になったわけです．10年くらい昔と比べると，ゲート数の感覚が1000倍くらい変わりました(**図1-2**)．デバイスのスピードも速くなって，ASICなら数百MHz～1 GHz，FPGAなどでも数十MHz～400 MHzは珍しくありません．

● HDLと自動論理合成CADの進歩

このような複雑な処理を行う回路を設計できるようにするために，HDLと設計自動化CAD(正確にはEDAツールという)が開発されてきました．多少の細かい無駄には目をつぶり，回路の大部分はコンピュータに設計させてしまうのです．人は概略設計だけをすればよく，あとはコンピュータが苦手な箇所(1-4節も参照)だけに手を入れるようにします．

今でもアセンブラがなくならないのと同じく，人が直接ゲートをいじる機会も残っています(将来も残るでしょう)．しかし，すべてをゲート・レベルで組まなくてもよくなったということは大きな進歩です．

〈図1-2〉10年前と現在のディジタル・システムの規模に対する感覚の相違



● 動作が複雑な回路を作るならHDLやCADは必須

このような動向のなかで，現在，HDLやCADをまったく使わずに回路設計をするのは，現場では考えられない状況になっています．

先述のような処理内容の複雑な回路を作るときは，たとえその回路規模が大きくなくても，適切な抽象度でCADを使って設計をしないと，とてもデバッグしきれません．もし複雑なプログラミングで(たとえば「Linuxを作る」とき)，コンパイラがなくてアセンブラしかなかったら，いつバグのないプログラムが完成するでしょうか?　回路でも同じです．

また，同じ処理でも速度などの性能が異なる回路をいろいろ取り揃えたい，少し機能を追加/削除したい，ざっと性能の見積もりをとりたい，といった局面も多くあります．CADを使わずに毎回毎回ゲート・レベルで回路を手設計していては，こうした要求にまったく応えられません．

● HDLと実際にできる回路との対応をつかむのが最大のポイント

その一方で，現場の回路設計では，なんでもいいから回路を作れば終わりということはまずなくて，回路速度や回路サイズに対する要求が必ずついてまわります．また，他の回路モジュールやチップとうまくつながるように調整する必要などもあります．実際に，CADの自動論理合成で得られる回路では性能が足りず，苦労させられる場合が多々あります．

回路の性能や信頼性を上げる作業は，単にHDLの文法やCADの使いかたを知っているだけではまったく不足です．「性能を上げるためには回路をどういじるべきか」「どうHDLを書くとどんな回路になるか」

というイメージが，自分の頭の中にないとできません．HDL設計を学ぶうえでは，これが最大のポイントです．おもに第3章と第6章で説明します．

● C言語とRTL

残念ですが，現在のところ，C言語のようなソフトウェアの記述から回路を自動で作るのは一般的ではありません．1-3節で説明しますが，RTL(Register Transfer Level)といって，もう少しハードウェアに近いHDLコードの書きかたをする必要があります．

近い将来，C/C＋＋言語の拡張版がHDLの代わりに使われるようになるかもしれません．実際に，現在System CやSpec Cなどの策定が進んでいます．ただし，これらはC/C＋＋と同じ文法で回路動作やRTLを書けるということであって，C＋＋/Cプログラミングとまったく同じセンスでハードウェアを設計できるようになるわけではありません．良い設計をしたり確実に動く設計をするにはハードウェアのセンスが必要なままなので，そこは誤解しないでおく必要があります．むしろ，ソフトウェアで良いアルゴリズムほど，ハードウェアに向かない傾向があります．

1-3　回路設計作業の全体的な流れ

それでは，以下，現在の回路設計の作業の流れと注意点について，順を追って紹介していきます．

各項目が本書の何章で詳しく説明されるかも示します．本書の内容は，「とりあえず動く回路を作るための知識」(第5章まで)と「回路性能を上げるための知識」(それ以降)に分けられます．

● いきなり回路を作り始めない

たとえばディジタル時計を作る場合，「60進カウンタを使って秒や分を数えればいいかな」と処理の進めかたを考えつくまでに，時間はかからないと思います．カウンタを組み合わせながら回路を考えていく，という方もいるでしょう．しかし，もう少し仕様(スペック)が複雑になってくると，ゲート・レベルで回路をいじりながら仕様を実現していく方法は，すぐに通用しなくなります．

そこで，以下のような基本手順に従い，処理アルゴリズムといった大枠の決定から始めて，必要に応じて細部を詰めていきます．

● 作業1…実現したい処理の回路仕様を決める

どのような処理をする回路を設計するのか，回路の仕様をきちんと決めます．ここをいい加減にすませると，後でいろいろな誤解やトラブルが起きるもとになります．

多くの場合，回路入出力のタイムチャートや状態遷移図，要求性能，回路内部のブロック図などをあわせたものが仕様です．

● 作業2…モジュールの構成を考える

通常，複雑な回路の設計では，全体の処理をいくつかの機能単位で分割し，それぞれを独立した回路モジュール(いわゆるIPコア)として設計/テストしてから，最後に各モジュールを結合して全体のテストをします．

このとき，各モジュールの性能をどの程度にして，各モジュールをどうつなぎ(全体のバス・アーキテクチャをどうするか)，どのような手順でデータの交換をするか，ということを検討します．それによっ

HDLによる高性能ディジタル回路設計
ディジタル/CD-R版

SD33951

価格：本体1,714円（税別）

価格：1,800円（税込）5%

CQ出版社

HDLによる高性能ディジタル回路設計