

まえがき

本書は HDL の文法書ではありません。大学水準の「デジタル・システム」の設計能力の習得を目指しています。ここで言う大学水準とは、米国のちゃんとした大学のように演習を課す大学という意味で、決して形だけの「デジタル」回路技術の講義や実験・演習でお茶を濁している水準を指してはいません。

その意味では、企業や研究の現場ですぐに実用に使える技術能力を得ることを目的としています。内容は通年の演習で使えるように構成されていますが、無理にそのとおりに学習する必要はありません。例題や課題をやっていく中で、自然と必要な事柄を身につけられるように考えられています。

本書では、この「デジタル・システム」の構築のための諸技術の解説とその設計演習を、HDL を使って単純な論理回路から複雑な「システム」まで順にやっていきます。基本的な「デジタル」技術や 2 値論理の理論、IC の制作方法などは、本書では簡単にしか説明しませんので、巻末に紹介する参考図書で学習してください。

なお本書に掲載されている例題は、論理回路の基本的なものを解説して、それがどのような論理式や Verilog-HDL の表現式になるかを理解してもらうためのものです。すべての設計は Altera 社の IC 上で動作を確かめてあります。また課題はそれを補足するためのものです。読み飛ばすだけでなく、ぜひとも自分でやってみてください。これだけでは技術の修得には不足ですし、自分の実力を評価してもらうこともできません。そのような方のために、下記の著者個人の Web site 上に実力検定ができるような「ホームページ」へ Link が張ってあります。本書を理解したら、ぜひ挑戦してみてください。

<http://www.ne.jp/asahi/yikai/class/index.htm>

またこの Web site には、本書では掲載しきれなかった課題の詳しい解説、および「デジタル」技術の基礎をなす部分としての半導体の技術や 2 値論理の理論的な側面の知識を補給するための参考図書や Web site についても、最新の情報を集めた一覧表やリンク集を用意してあります。

著者は大学を卒業して以来、40 年ほど「デジタル・システム」の設計にたずさわってきました。その間、大きく進歩したのは Digital IC の製造技術です。しかし残念ながら、回路設計技術は思ったほど進歩しませんでした。唯一の大きな進歩は、設計に計算機が自由に使えるようになり、図面や論理式を人手で書かなくてもよくなったことでした。

Hardware Description Language(HDL)が考え出されたことは、大きな進歩のようですが、そのような考えとやり方は、すでに 1960 年代から計算機の設計現場では個別に使われていました。ただ HDL

にいくつかの標準方式が決められたことで、HDL を誰もが使えるようになり、設計の可搬性が上がり再利用が可能になったことは、大きな進歩です。

「デジタル・システム」の記述法として、おもに論理回路を記述する MIL 記号法がかつて作られ、それによって記述された TI 社の 74 系列(series)の汎用論理 IC が提供されたことは、多くの技術者にとって学習や設計の指針となり、「デジタル」技術の普及に大いに貢献しました。しかし、この記号法に慣れたことが、「システム」を記述するには不利に作用しました。すなわち、この伝統的な MIL 記号法は、「システム」の記述について何も想定していないからです。

その後回路図の記述法は、より機能を明確に記述できる方法が開発され、JIS でも規定されましたが、IC はすでに LSI 化の時代に入ってしまったため、74 series の汎用論理 IC で「デジタル・システム」をすべて作ることはなくなりました。しかし、74 series の汎用論理 IC で書いた回路図から直接 LSI を作る、という変則的な逃げ道で回路設計を進める方法はよく使われました。

CPU に限らず、誤り訂正符号や Protocol の処理、Data 圧縮・伸長回路などを含んだりする、大きな規模の論理回路の設計を、MIL 記号法で書いていたのでは、設計用紙がいくらあっても足りません。論理式やさらに「システム」を記述できる HDL は必然となりました。また、論理回路の内部動作を実物で検証できなくなった LSI では、論理 Simulation をしないと実物を作ることさえできません。

汎用論理 IC の出現以前から、論理回路は汎用の離散系 Simulation 言語である GPSS などを使ったり、専用で作成した Program で動作検証をした例は多かったのですが、Logic analyzer などを利用した直接検証も有力な手法でした。直接目で見られるということは便利なもので、設計した論理回路を実物にした後に、基板上の配線を切ったり貼ったりして「虫取り」をしました。

歴史的経緯は以上のようなのですが、今では LSI の中を直接いじったり検査できないので、「デジタル・システム」を構築しようとするときには、どうしても Simulation の利用は避けられません。そのためには回路図よりは処理が簡単な HDL を使わないわけにはいきません。本書では、数式による RTL (Register Transfer Level)記述だけではなく、論理式による記述も適当に使い分けていきます。回路図による表現は Block 図として設計の意図を理解できるよう適宜加えます。

すべての設計を RTL の数式で記述していると、回路規模が膨らむという問題点が現実であり、回路の速度や規模を問題にする用途では、論理式や Gate 水準で回路設計をすることも必要になります。実際には数式による記述は、HDL 用の Compiler の組み込み関数によって、論理式や Gate 水準の Primitive による記述に展開されます。どのような回路になるかを知ること、設計技術者の実力の下支えになります。

簡単な回路や組み合わせ論理回路の説明では MIL 記号法による回路図も適当に加えて、さらに波形図や真理値表、Karnaugh 図も利用して、設計技術の理解を深めるようにします。組み合わせ論理回路では論理式による設計にも重点を置き、Flipflop を含む順序回路では数式で論理を記述した RTL 水準の設計を主に使います。

RTL 水準の論理回路を表記する HDL 言語としては VHDL と Verilog-HDL が現在もっとも普及していますが、本書でその双方を平行して記載するのはむだが多いのと、論理式との親和性から、表記法としては厳密さを欠く嫌いはありますが、記述が短くなる Verilog-HDL を使用します。

たとえばよくないのですが、Programming 言語とくらべると、VHDL は Pascal 的で Ada の流れを汲んでいます。Verilog-HDL は C 言語からの発展形です。Verilog-HDL の多くの記述規則は C 言語からの流用です。もちろん論理式や Gate 水準の表現は Assembler 水準というわけですが、さらに C 言語のままでも、論理回路の表現は SpecC などのように工夫を加えればできますが、本書では言及しません。

本書では Verilog-HDL の Tool 類には、Altera 社から無償試用版が提供されている Max+plus II とその付属 Tool 類を使うことを前提にします。当然ながら論理合成の対象も、同社の FLEX や Max といった CPLD(Complexed Programable Logic Device) ですが、他の Device や Simulator へも EDIF File を出力できますので、汎用的な設計 Tool として使うことが可能です。もちろん Verilog-HDL や VHDL の Source text のまま他の Design tool 類に渡すことも可能なので、どの IC にでも設計した回路を実装できます。なお、Tool 類の使い方を本書に載せると分量が多くなりすぎるので、前述の著者個人の Web site で Support しています。

論理式での表現には、同社の AHDL の表記法も用います。なおこの AHDL の表記法は、PLD の論理回路の Assembler として広く使われた PALASM や CUPL の表記法を直接受け継いでいます。これらは多くの参考文献や資料でも親しまれており、AHDL による論理式の表記法は、誰でも容易に理解することができると思います。

また AHDL による論理式水準の記述は、簡単に Verilog-HDL での論理式水準の記述に変換できます。AHDL は使わないという人にも十分役に立ちます。

Verilog-HDL での設計は数式などで動作状態を表現する RTL 水準を主とします。論理素子 Module を直接使った Gate 水準の Primitive での表現や論理式での記述は、練習や慣れのため以外はあまり使いません。もし必要なら、上述のように AHDL での記述から変換してください。

Altera 社の Tool は本来有料ですが、試用のための機能限定版は Internet からや技術雑誌の付録 CD-

ROM として無償で提供されています。これは試用版といっても、AHDL で記述するなら本書の執筆時点で、3 万 Gate 相当の IC chip まで使えるので、本書の課題の練習をする程度や 74 系列 IC 100 個程度の回路規模なら、無料の範囲で十分実用的な設計ができます。もちろん Verilog-HDL や VHDL で記述することもできますが、Flipflop 数で 500 個程度までに限定されます。ほかのもっと大きな回路が設計できる Verilog-HDL 用の Tool で EDIF 形式の出力を得て、Max+plus II で配置・配線させることもできます。

この試用 Program を使用する際には License key が必要です。BaseLine(AHDL 用)あるいは E+Max(VHDL , Verilog-HDL 用)の License key は各自で Internet 上の Altera 社の Web site (<http://www.altera.com/cgi-bin/indexhtml.pl>)から入手してください。

なお、本書にも Altera 社提供の Tool 類が CD-ROM として付いています。使い方などは著者の個人ホームページを参照してください。

また本書の中の Simulation 波形の図版には、本書の設計を Simulation した Max+plus II の画面から直接引用しました。同社に感謝いたします。

本格的に量産化される LSI を設計する必要がある方以外で、研究用や試作・多品種少量生産にたずさわの方は、この Altera 社の CPLD 類や他社の Field Programable Gate Array(FPGA)類を使って論理回路を作らざるを得ません。74 系列の汎用論理 IC を基板上に配置して論理回路を組むことは、IC の供給や速度、論理規模の点からも、もはや実用的ではないのです。

現実には、論理回路の実用化の件数においては、これらの Programable 論理素子の利用例は、Gate array などの LSI の設計件数より絶対的に多いわけですから、これらの Tool で論理回路の構築の練習することは決して無意味ではありません。

最後に本書では、人名や信号名のほかにも回路や技術の用語も極力英文表記のまま使っています。漢語を用いた日本語表現ができるものはそれを利用します。たとえば「論理」は使いますが「ロジック」、 「Logic」は使わない、「Enable 信号」を使い「イネーブル・シグナル」を避ける、というふうです。

その理由は、とくにカタカナ語が日本語として定着していない用語は、へたにカタカナ化するとその意味や原語の綴りが不明になるからです。国際化の中で外国人と技術の話をしたときに、英語が通じない中で単語だけでも正しく綴ると話が簡単に通じた、という筆者のみっともない経験からの手法です。またこの方法は、日本語を習ったばかりの外国人留学生や技術者にとっても、技術を早く正確に理解するためには有効な方法であると、著者の少ない技術指導・教育経験からも感じています。