

## デジタル回路は電子回路でありハードウェアである

藤平 雄二  
Yuji Fujihira

### ● ハードウェアを作る力は今なお重要！

私はかつて制御系のソフトウェア開発の仕事をしていました。制御系のソフトウェア開発では、ターゲットと呼ばれるデバッグ用の装置が必要です。このターゲットに簡単なトランジスタ回路を付け足す必要が生じたとします。この作業を、どれだけのソフトウェア担当者がこなせるのでしょうか。少なくとも私の経験では、ほとんどのソフトウェア担当者はお手上げでした。

ハードウェアの設計でも同じようなことが起ころうとしています。現在、デジタル回路にはFPGA (Field Programmable Gate Array) や CPLD (Complex Programmable Logic Device) といったプログラマブル論理素子が使用されています。7~8年前はこれらの設計には、まだまだ回路図入力を使用していました。HDL (Hardware Description Language) による論理合成システムが非常に高価だったからです。

しかし、HDLによる論理合成システムが安価になった現在、回路図入力で設計している人はまずいないでしょう。こうなってくると、デジタル回路の設計をしていながら「はんだ付けをしたことがない」という人が出てきても、決して不思議なことではなくなるでしょう。

以上のような状況ですが、だからといって**電子回路というハードウェアの設計技術を習得する必要がなくなったというわけではありません。**

### ■ ハードウェア技術を マスタすることの価値

ソフトウェアのほうが取っ付きやすいのでしょうか。

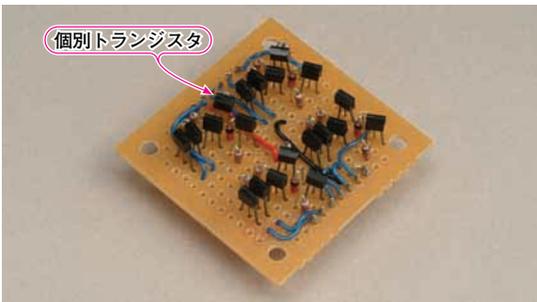


写真1 個別のトランジスタで製作した NAND ユニット ▶ 第2章  
ユニバーサル基板に手配線で製作した。ピン配列は7400に準拠している

今後ますます、ハードウェア技術を目指す人は少なくなると思います。特にアナログ回路を目指す人ともなれば、近い将来、絶滅寸前の状態になるおそれもあります。しかし、このような状態だからこそ、ハードウェアを習得したエンジニアは重宝されるはずで

### ● 開発効率が上がる

制御系のソフトウェア技術者にとっては、**ターゲットを自分で改良/修理できるようになり、効率の良い仕事ができるようになります。**さらに周囲から尊敬の目で見られること請け合いです。これは職場での地位向上を意味します。

もちろん、ソフトウェア作成にもハードウェアの知識が役に立ちます。そもそもハードウェアの知識なしに制御系のソフトウェアを作成するのは無理があるように思われます。

### ● トラブルに対処できる

設計がHDLになったとしても、ハードウェアの知識は必要です。特に、**トラブルを解決したり、より効率の良い設計を行うためには、どうしてもハードウェアの知識が必要になります。**

自分の専門に直接関係がなくても、習得しておくべき基本的な技術があると思います。デジタル回路の基本は、まさにこのような技術ではないでしょうか。このような基本的な技術は若いうちに身に付けておくと、一生役に立つでしょう。

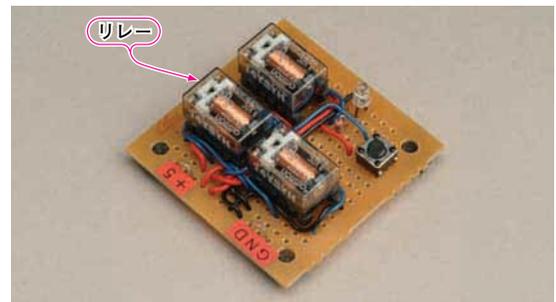


写真3 リレーを使って手作りしたフリップフロップ回路 ▶ 第4章 Appendix

押しボタンを押すごとにLEDが点灯/消灯を繰り返す

## ■ NAND ゲートから始めよう！

### ● NANDゲートはデジタル回路の基本中の基本

デジタル回路習得の第一歩は、**NANDゲートやフリップフロップなどの基本機能を習得すること**です。本特集の第2章と第3章では、これらの基本機能について電子回路レベルから説明しています。

技術というのは実際にやってみて始めて理解が深まるものです。C言語などのプログラミングを習得するには、実際に試行錯誤を繰り返す以外、良い方法がないのは万人の認めるところでしょう。デジタル回路の習得もまったく同じです。実際に製作せずに理解を深めることは、まず不可能でしょう。そこで、第2章では**トランジスタを使って実際にNANDゲートを製作することにより、デジタル回路の最も基本であるNANDゲートのふるまいを確実に理解できるようにしました。**

さらに、実際に製作したNANDゲートで、組み合わせ回路やフリップフロップを製作しました。自分で製作したNANDゲートを使うことにより、これらの理解をより完全なものにしようというわけです。以上により、デジタル回路習得の第一歩は確実に踏み出すことができると思います。

### ● TTLもCMOSも考え方は同じ

NANDは個別トランジスタによるTTL構成で製作しています。現代ではTTLはほとんど使われなくなっており、CMOSがほとんどです。本当はCMOSで製作したほうがよいのかもしれませんが、しかし、CMOSの個別部品を入手するのが困難なため、TTL方式でNANDゲートを製作しました。

今さらTTLを学習して一体何の役に立つのかと思われる方もいらっしゃるでしょう。でもご安心ください

い。TTLでもCMOSでも考えかたは共通です。TTLは論理素子の基本中の基本であり、TTLを完全に理解しておく**とCMOSを理解するのは容易**です。喩えれば、ある一つのマイコンを完全に理解して使いこなせるようになると、ほかのマイコンに移行するのも非常に容易になるのと同じです。

## ■ デジタル回路マスタへの道は同期式順序回路の理解から始まる

デジタル回路習得のその次の一步は、**同期式の順序回路を確実に理解すること**です。

ある画像のフレーム・メモリを設計したときの事です。それまでは場当たりのフリップフロップを使用していましたが、同期式の順序回路の考え方に改心しました。同期式の順序回路を採用することにより、「ある状態Aでは垂直同期信号を待ち、垂直同期信号を検出したら状態Bに遷移し、状態Bでは水平同期信号を何本か待ち、それを検出したら状態Cに…」と整然と設計することができました。同期式順序回路の考え方によると、なんと簡単に、なんと楽しく回路が設計できるのだらうと感激したものです。

この特集だけで、同期式順序回路を完全に自分のものにするのは難しいかもしれません。しかしその大切さを知り、今後、同期式順序回路を完全に自分のものにするための第一歩として、本特集を役に立てただけならと思います。

\* \* \*

基本の習得がなければ、見聞する知識は素通りしてしまい、技術が上がっていくことは決してありません。一方、基本を習得しておく**と、その後に見聞する技術を確実に吸収でき、今までの技術の上に積み上げていくことができるのです。**

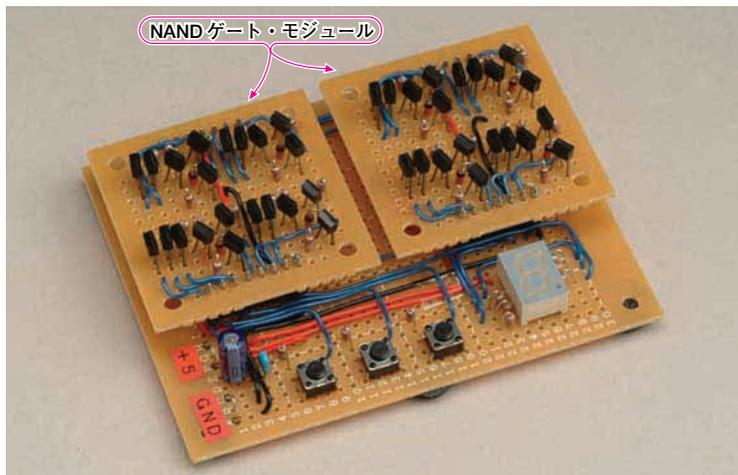


写真2

7セグメントLEDのエンコード回路▶第3章  
写真1のNANDユニットを2枚使用して、1、2、3の数字を表示する組み合わせ回路を構成した