

ステート・マシンとモジュール化、階層設計を取り入れる

HDLによるプログラミングは、合成してFPGAに書き込むとハードウェアになるが、記述する上ではプログラムである。本章では、一般的なプログラムと同じように逐次的な処理を記述するステート・マシンを説明する。また、既存の資産を有効に活用する方法として、各機能ごとにモジュール化を行い、トップ階層でつなぎ合わせる階層設計の記述方法も解説する。
(編集部)

ソフトウェアのプログラミングでは、行う処理や必要なデータを抽象化し既存の資産を有効に活用したり、再利用性を高めたりすることが求められます。また、よくあるパターンとして定式化することでソフトウェアとして綺麗な記述を目指します。これらの取り組みにより、ソフトウェアを記述する時間やデバッグ時間を削減できます。同じように、HDLなどによるハードウェアのプログラミングも、データ構造の抽象化やモジュールの再利用などにより、記述やデバッグの負荷を削減できます。

本章では、決められた手順で処理を実行するモジュールを設計するときの有効となるステート・マシンや、すでに実装したモジュールをサブモジュールとして利用する方法を説明します。

に必要不可欠な概念であるステート・マシンと、その実装方法について説明します。

ハードウェアでは処理を並列に実行できますが、世の中には順番にしか実行できない物事もたくさんあります。たとえば、そうめんをゆでるとき、鍋の水がまだ湯になる前に並行してそうめんを鍋に入れても、とても食べられるものはできあがりません。

● 料理は逐次的な処理

何かを実行し、その次に何かを実行し、その次に…、という決まった手順に従った処理の実装は、ソフトウェアによく見られます。たとえば、そうめんをゆでるとき(図1)には、

- ① 鍋に水を入れる
- ② 鍋を火にかける
- ③ 沸騰するまで待つ
- ④ そうめんを入れる
- ⑤ 1分くらい待つ

1. ステート・マシンを実装する

ハードウェア・プログラミングで逐次的に処理するため

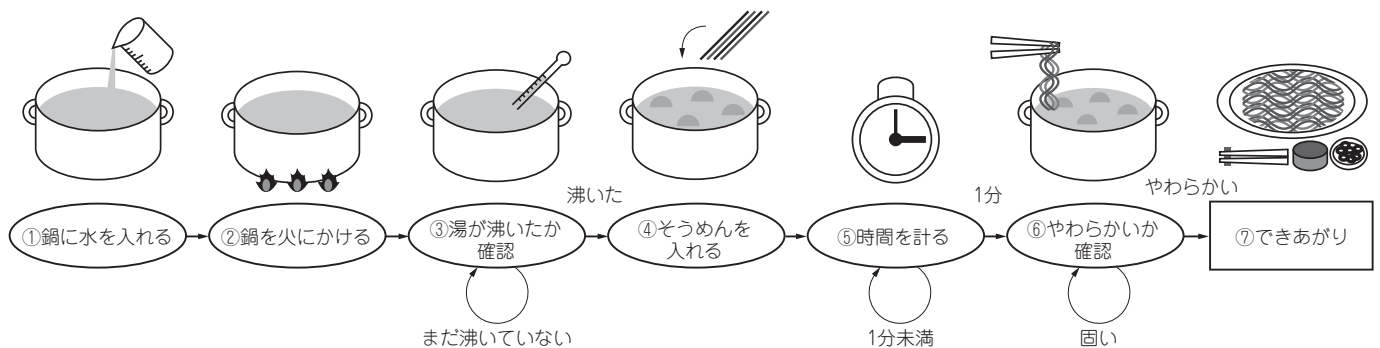


図1 そうめんをゆでる手順をステート・マシンで表す