

# Chapter 4

## 性能指向のステート・マシン設計

本章では、ステート・マシンについて解説します。ステート・マシンは、システムの中枢神経にあたる重要な働きをします。そのため、ステート・マシンの設計技術は、システム全体の性能を決定する大きな要素になります。

### 4-1 ステート・マシンとは

ステート・マシンは、順序制御を行うための回路です。ステート・マシンの中には複数のラッチが含まれていて、それらのラッチの‘0’、‘1’の組み合わせでステート(状態)を表します。状態を表すラッチのことを「状態ビット(state bit)」と言います。

例えば、3個の状態ビットが含まれていて、これらがすべて‘0’となる“000”は一つの状態です。これらの状態ビットが変化して例えば“100”になると、これは別の状態です。3個の状態ビットが“000” “100” “110” “111” “011” “001” “000”というように、決められた順序で変化すると、これがステート・マシンの動作ということになります。このステート・マシンの変化に応じて、順序制御を行うことができます。ステート・マシンはなんらかの外部信号によって起動し、一連の動作を終えると初期状態に戻ります。

ステート・マシンには、状態ビットのほかにもいくつかの回路が含まれます。タイミングをとるためのタイマや数を数えるためのカウンタ、データを記憶するためのレジスタ、比較回路、演算回路、外部との通信をするためのインターフェース回路などが必要に応じて付加されます。では、ステート・マシンの最小構成はというと、それは状態を表すための状態ビットとインターフェース回路になります。ステート・マシンにインターフェース回路は欠かせません。それは、ステート・マシンが順序制御のためにはならず外部信号を見る必要があるからです。

なお、ステート・マシンの機能は順序制御だけではありません。このほかに、同期、協調、排他制御といったものがあります。

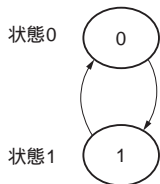
#### もっともシンプルなステート・マシン

いちばん簡単なステート・マシンは、状態ビットが一つだけのステート・マシンです。このステート・マシンでは、状態ビットが‘0’のときを状態0、状態ビットが‘1’のときを状態1とすれば、二つの

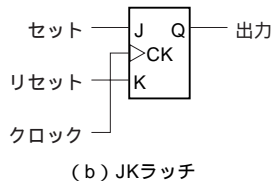
図4-1

## シンプルなステート・マシン

もっともシンプルなステート・マシンとして、状態ビットが一つのものが考えられる。これは、JKラッチのとして知られている基本的な回路部品である。



(a) 状態ビットが一つのステート・マシン



(b) JKラッチ

状態を表すことができます(図4-1)。このステート・マシンは、JKラッチとして知られている基本的な回路部品です。

二つの状態は、次に示す例のように、制御信号で互いに移動することができます。

- 状態0 外部信号Jが'1'になったら、状態1に移動する
- 状態1 外部信号Kが'1'になったら、状態0に移動する

状態ビットの数を増やしていけば、状態の数も増やすことができます。

ステート・マシンの変化の条件には外部信号を使いますが、変化するタイミングそのものはクロックという別のタイミングを用います。

### システムを共通のクロックで動作させる「同期式设计」

回路は、システム・クロックという共通のクロックで動作しています。つまり、複数のラッチが含まれているシステムでは、それぞれのラッチは次に変化する先が'0'であるか、または'1'であるかを外部条件や内部条件を用いて決めますが、そのタイミングをとるには次に来るシステム・クロックの立ち上がりを使います。

これは複数のラッチを協調動作させるために非常に重要なことです。そもそもステート・マシンでは複数の状態ビットが同時に変化しないと、状態の意味がありません。またデータについても、データのビットごとにばらばらに変化するようでは、いつ正しいデータがそろうのか、タイミングをとるのが難しくなります。そこで、クロックを使って、あるタイミングでデータや状態を確定させ、次の状態に遷移するというのを繰り返しています。

システム全体を共通クロックで動作させることを「同期式设计」と言います。

### ラッチごとに違うタイミングで動作する「非同期式设计」

同期式设计に対して、それぞれのラッチがばらばらのタイミングで動作してもかまわないという設計法もあります。これは、「非同期式设计」と言います。非同期式设计の場合、最終的にむだな動作のないコンパクトな回路を得られる可能性があります。

ところが、大規模な回路を効率良く設計し、テストするという観点から見ると、この設計法は適しているとはいえません。これは、同期式回路の設計ツールやテスト・ツールは比較的開発が容易であるのに対して、非同期式回路の同種のツールの開発が困難とされているからです。同期式回路では、ラッチの変化するタイミングをクロックの立ち上がりという一点に統一することで、回路動作を単純化することができるのです。そして、大規模の回路設計に対応できるようになっています。ASICや

FPGAの設計においては、同期式设计を採用することにより、効率の良い設計とテストが可能になります。

## 4-2 ステート・マシン設計の基本

ステート・マシンの設計の基本は、必要な状態数を数えるという地道な作業から始まります。例えば、外部信号によってパルスを二つ出すという順序制御を考えてみましょう。この場合、以下に示すように四つの状態が必要とわかります。

- 状態0      アイドル状態。外部信号によって状態1へ移動
- 状態1      一つ目のパルスを出す。状態2へ移動
- 状態2      状態3へ移動
- 状態3      二つ目のパルスを出す。状態0へ移動

状態の数が決まれば、ここで状態ビットを割り当てます。

### 状態の流れを決める

状態は区別さえできればよいので、いくつもの状態の決めかたが考えられます。例えば、最初の状態を“00”と決めたとします。この場合、その後の状態の流れは図4-2に示す3通りが考えられます。これは組み合わせ問題です。つまり、ビットの交換とビットの反転が同じであると考え、初期値は“00”でかまわないということになり、その後は並べ替えで3通りに決まってしまう。ところで、図4-2の状態の流れを見ていると、図4-2(a)はジョンソン・カウンタ、図4-2(b)はバイナリ・カウンタ、図4-2(c)はその逆順になっていることがわかります。

状態の決めかたが自由だからといってばらばらに決めていくと、設計の自由度が大きくなりすぎて、かえって收拾がつかなくなります。状態の流れを制御するにはなんらかのベース・カウンタを使うと便利です。よく使われるベース・カウンタの例としては、次のものが挙げられます。

- バイナリ・カウンタ(図4-3(a))
- グレイコード・カウンタ(図4-3(b))
- リング・カウンタ(図4-3(c))
- ジョンソン・カウンタ(図4-3(d))

ベース・カウンタの選びかたによって、ステート・マシンの性質が変わってきます。

状態の変化に伴って状態ビットは変化する(あるいは変化しない)のですが、二つの状態を比較したときに一致していない状態ビットの数を「ハミング距離」と言います。ハミング距離が2以上の場合、

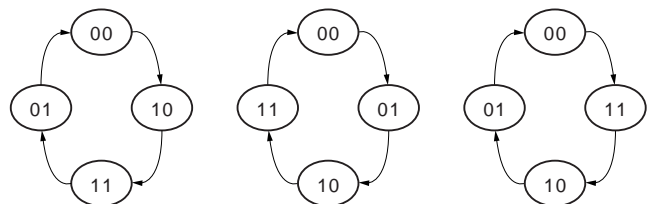


図4-2

状態の決めかた

初期状態を“00”とすると、(a)~(c)の3通りの状態のとりかたが考えられる。

(a) 状態の流れ(その1) (b) 状態の流れ(その2) (c) 状態の流れ(その3)

見本