

まえがき

手のひらのケータイがインターネットにつながり、ビーチでスキーと面と向かって会話できる。ロボットのペットと遊ぶ。1日分のテレビ番組をまるまる録画でき、宇宙からの映像さえも見る事ができる。こんなことが「あたりまえ」と思えるようになりました。このようなことが実現できるようになった大きな要因の一つに、LSIの高性能化、高機能化が挙げられます。

新しいアイデアを形にするのはさまざまな方法で可能です。しかし、既存の製品より高機能、高性能が要求されるアイデアを実現するためには、「ハードウェア (LSI)」が必須です。

これまでLSIを作るということは、莫大な費用を必要としました。しかし、FPGA (field programmable gate array) やCPLD (complex programmable logic device) の登場により、LSIが身近なものになっています。近年、FPGA技術は飛躍的に伸び、ASIC (application specific IC) にとってかわるまでの勢いをみせています。メモリやマイクロプロセッサなどが取り込まれたFPGAもあります。この夢のハードウェアを使いこなせば、自分のアイデアをLSI化できる時代がきているのです。

LSI開発環境は、中小規模FPGA設計ツールを無償で利用できます。ちょっと試してみることができるようになっています。このことは教育という点においても大きな革命を起こしています。FPGAと無償設計ツールを用いて実験講義を行う大学も続々と増えつつあります。さらに、企業での教育という点でも、コストを削減し、十分な人材育成を図ることも可能になっています。

このような背景から本書では、FPGAボードを使いながらLSI設計技術の基礎を学んでいく構成を採りました。LSI設計の入門書と位置付けていますが、しごとでLSI設計を始める人はもちろん、趣味でおもしろいハードウェアを作る土台としても使えるものとしてまとめました。さらに、FPGA設計で避けては通れないハードウェア記述言語 (HDL ; hardware description language) の書きかたを実際の回路を設計しながら理解できるようにもなっています。LSI設計なんてまったくわからないという人でも本書を読めば簡単なゲームまで設計できるようになります。

ハードウェア設計技術は、書籍を読むだけでもある程度は身に付きます。しかしLSIを設計するのであれば、実際のチップとそれを動作させるためのボードがないと実感がわかないものです。さらに、正しく設計できたかどうかを実際に確認することが、理解の最短経路だと思えます。本書にはFPGAを搭載したボードとFPGA設計ツールも付属しています。LSI設計をはじめための道具をすべて用意しています。本書とともに、パソコンの前に座れば準備完了です。

本書が、将来のトップ・エンジニア、ひいては新しい技術を生み出す引き金となることを願います。

2003年4月 リスボンにて
山際伸一

本書の構成

● 目標 野球ゲームを作る

本書では、プログラマブル・デバイスで動作する回路を設計しながら、HDLによるLSI設計の基礎について解説していきます。

付属のFPGAボードで動作する「野球ゲーム」を作り上げることを最終目標にします。しかし、いきなり設計することはできません。最初はとても簡単な回路をHDLで記述しFPGAボードで動作させてみます。回路を少しずつ拡張し、それをFPGAボードで動作させながらHDL設計を身に付けていきます。

● 特集の構成

読者のみなさんは、まず「野球ゲームを作る」という目標をしっかりと持ってください。ただ書籍を読もうとしてもなかなか意欲がわかないものですが、明確な目標があれば、がんばろうと思えるものです。

まず、第1章では、ハードウェア設計における手順について、アイデアがLSIになるまでの過程でどのようなことを考え進めていくかについて解説します。プログラマブル・デバイスとASICの違いについてもふれます。

第2章では、LSI設計の流れを体験していただきます。付属のFPGAボードを活用するために使用する開発ツール「Quartus Web Edition」の操作を説明します。ひとつおりの操作を体験することにより、FPGA開発ツールの使いかたが身に付きます。ここでは、VHDLの詳細には触れません。あらかじめ設計済みの回路を用意します。FPGAボードが動作するのを見れば、きっと設計へのモチベーションも高まると思います。

第3章では、VHDLによる回路設計の基礎を固めます。非常に簡単な回路から、簡単なゲームを動作させるまで、少しずつステップアップしながら、実験を通して理解していきます。

第4章では、本書の最終目標である「野球ゲーム」の設計を行います。仕様の分析から開始し、機能分割、VHDLによる回路の記述を行い、FPGAボードでゲームを動作させるまで、機器の開発フローにしたがって



目標はしっかり持とう

解説します。実際に手を動かし、記述を真似することで、VHDLおよび、FPGA 上での電子設計を習得していただけることを期待します。

本書により、読者のみなさんがVHDLによる論理回路設計をマスタされ、すばらしい製品を生み出すことを願います。

使用する設計環境

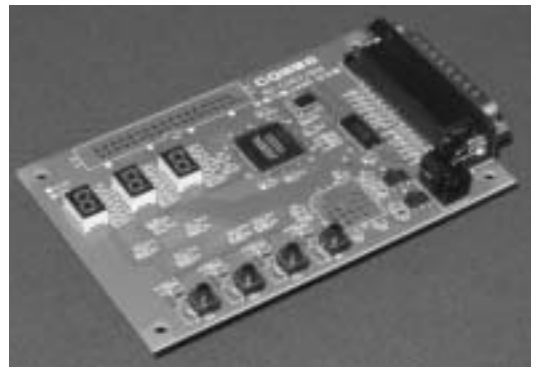
本書では、米国 Altera社のFPGAをターゲットにします。したがって設計環境については、このデバイスを用いる場合を想定して解説します。

ここでは使用する設計環境について、おおざっぱに説明しておきます。専門用語や固有名詞がいくつか出てきますので、HDL設計について予備知識がまったくない方は何を言っているのかわからないかもしれません。とりあえず図や写真だけ眺めて、こんなツールを使うという程度のイメージだけ持っておいてください。詳しい説明は、後の章で行います。

● FPGA ボード

本書では、付属のFPGAボードで回路を動作させるための回路を設計することを前提として解説を進めます。

本書に付属のFPGAボードには、Altera社の「ACEX 1K」ファミリのうち、1万ゲート規模の「EP1K10」を搭載しています。周辺機能には、単体LED、7セグメントLED、押しボタン・スイッチが用意されています。本書では、このボードのことを「FPGAボード」と呼ぶことにします。仕様の詳細は、付録Cを参照してください。



ACEX 1K搭載のFPGAボード

● 使用するソフトウェア・ツール

Altera社の中小規模デバイスの開発では、無償版ツールを使用することができます。本書に付属のFPGAボードも、無償版の設計ツール「Quartus Web Edition」を使用できます。この設計ツールは、本書に付属のCD-ROMに収録しています。また、最新版のツールは、Altera社のホームページからダウンロード可能です。インストール方法は、付録Aを参照してください。使いかたを第2章で説明しています。

目 次

まえがき	3
本書の構成	4
第1章 HDLによる回路設計を始めるにあたって	9
1.1 システム設計と仕様策定	9
1.1.1 システム・レベルの設計	10
1.1.2 ボード・レベルの設計	10
1.1.3 LSIレベルの設計	11
1.2 ASICとFPGA	11
1.2.1 ASICとは	12
1.2.2 プログラマブル・デバイスとは	12
1.2.3 ASICとPLDの使い分け	15
1.2.4 プログラマブル・デバイスの魅力	15
1.3 LSI設計の流れと開発環境	16
1.3.1 デザイン・エントリ	17
1.3.2 シミュレーション	18
1.3.3 論理合成と配置配線	18
1.3.4 設計ツールの選定	19
第2章 FPGA開発を体験する	21
2.1 Quartus によるFPGA開発の流れ	21
2.2 Quartus の基本操作	22
2.2.1 Quartus の起動	22
2.2.2 プロジェクトの作成	23
2.2.3 設計エントリ	27
2.2.4 コンパイル	29
2.2.5 ダウンロード	33
2.3 Quartus を使いこなす	34
2.3.1 既存のプロジェクトを修正する	34
2.3.2 既存の設計データを利用する	36

第3章 VHDLによる論理回路設計の基本テクニック	37
3.1 practice1 : LEDの点灯と消灯	37
3.1.1 ライブラリ宣言	39
3.1.2 パッケージ呼び出し	39
3.1.3 エンティティ宣言	39
3.1.4 アーキテクチャ宣言	39
3.2 practice2 : 複数のLEDでバイナリ値を表示する	40
3.3 practice3 : 7セグメントLEDに16進数を表示する	41
3.3.1 デコーダ	41
3.3.2 process文	42
3.3.3 case文	44
3.3.4 ハードウェアの基本は同時処理	44
3.4 practice4 : 7セグメントLEDに表示する数字を変化させる	45
3.4.1 クロック	45
3.4.2 if文	45
3.4.3 バイナリ・カウンタ	47
3.4.4 算術パッケージ	48
3.5 practice5 : スイッチのON/OFFに合わせてLEDを点灯/消灯する	48
3.5.1 チャタリング除去回路	50
3.5.2 generic文	51
3.6 practice6 : ルーレットを作る	51
3.6.1 ステート・マシンによる設計	55
3.6.2 ミーリ型とムーア型	55
3.7 practice7 : スロットマシンを作る	57
3.7.1 階層設計	62
3.7.2 component宣言	62
3.7.3 ポートのマッピング	63
第4章 野球ゲームの設計	65
4.1 野球ゲームの仕様策定	65
4.1.1 動作仕様を考える	65

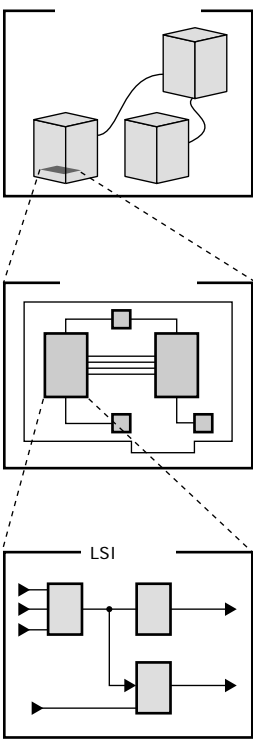
4.1.2	ボードの仕様を考える	67
4.2	機能分割	68
4.3	各モジュールの仕様策定	70
4.3.1	トップ・モジュール(baseball_top)	70
4.3.2	ルーレット管理モジュール(batting)	70
4.3.3	ルーレットの出目パルス発生モジュール(batlatch)	72
4.3.4	ルーレット状態出力モジュール(batter_led_dec)	72
4.3.5	アウト・カウント管理モジュール(outcount)	73
4.3.6	出塁状態管理モジュール(base)	73
4.3.7	加点パルス発生モジュール(score)	73
4.3.8	得点/出塁状態管理モジュール(baseball_led_out)	73
4.4	再利用性の高い機能ブロックの関数化	74
4.5	得点/出塁状態管理モジュールの設計	77
4.6	ルーレット状態出力モジュールの設計	80
4.7	アウト・カウント管理モジュールの設計	81
4.8	ルーレット管理モジュールの設計	83
4.9	ルーレットの出目パルス発生モジュールの設計	88
4.10	加点パルス発生モジュールと出塁状態管理モジュールの設計	91
4.11	クロック分周モジュールの設計	99
4.12	トップ・モジュールの設計	100
4.13	FPGA ボードで動作させる	106
4.14	機能の拡張について	107
付録 A	FPGA 設計環境のセットアップ	109
付録 B	トラブル・シューティング	116
付録 C	付属FPGA ボードの概要	120

第 1 章

HDL による回路設計を始めるにあたって

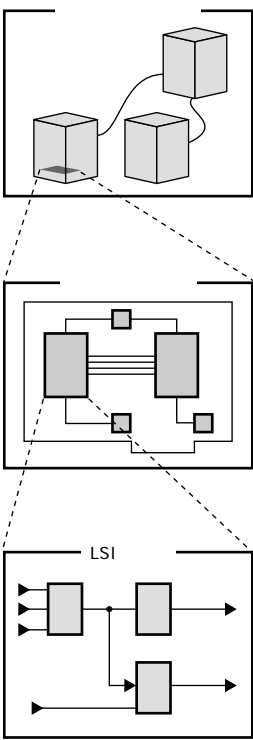
この章では、ハードウェアの基本的な設計手順について説明します。まず仕様策定時に考える事項をまとめます。専用LSI開発の二つの選択肢であるASICとプログラマブル・ロジックの違いを比較します。また、LSI設計に必要な設計ツールについても説明します。

1.1 システム設計と仕様策定

一つの独立した機器(システム)を設計することを考えてみましょう。設計者は、 1-1に示すように、

- 1) システム・レベル
- 2) ボード・レベル
- 3) LSIレベル

という順序で設計を進めていきます。ある過程で行き詰まった場合、最上位のシステム・レベルで検討し直すことも必要です。

LSI設計者にとっては、仕様に沿ったLSIを作り上げることがゴールですが、だからといってシステム・レベルのことを考えなくてよいというものではありません。設計するLSIは何らかの機器の一部として使われる部品だからです。つまり、 1-1に示すような順序で、さまざまな制約事項を考慮して決められたものなのです。

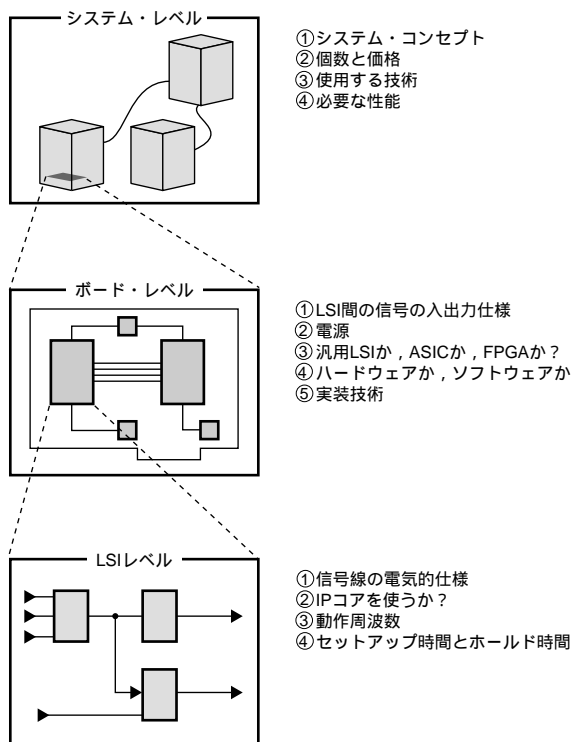


図 1-1 仕様策定時の検討事項の例

とはいえシステム・レベルやボード・レベルの設計に必要な技術は非常に幅広く、また奥が深いものです。HDL設計は始めるにあたって、すべてを理解しなければならないというものではありません。ここではこれからHDL設計を始める方であっても知っておいていただきたい基本的な解説だけを行います。より深い知識は、本書の内容をクリアしてから、さまざまな文献を参考にしながら身に付けてください。

●1.1.1 システム・レベルの設計

システム設計では、何をさせるものを作るか、そのターゲットを絞り込み、プロジェクトの終点を見据えることが重要です。また、製造台数と価格を見積もる必要もあります。システム・コンセプト、個数、価格まで決定すれば、あとは、どのような技術を使うかを考えなければなりません。使用する技術が独自技術であれば、設計期間に研究や技術開発の時間を含める必要があります。すでに標準化された技術を使う場合でも、ライセンス形態などを調査し、利用するための条件を十分に調べないと特許侵害などの問題に発展する場合があります。標準技術で設計資産があれば、設計期間を短縮できる可能性もあります。

システム・レベルの仕様策定で重要な事項は性能です。例えば、システムに何らかの信号を与えたときの応答時間はどれくらいかといった内容です。可能な限り高速なものでなければならないのか、それとも、人がストレスを感じない程度の応答速度でよいのかによって実現方法が変わってきます。これらさまざまな要素を考慮してシステム・レベルのブロック図を作りあげていきます。

この段階で、どの機能をどのように実現するかのイメージがあれば、後の工程をスムーズに進められるでしょう。すなわち、システム・レベルのブロック図をもとに、ボード・レベルの仕様策定を行うこととなります。

●1.1.2 ボード・レベルの設計

次に、システム・レベルのブロック図をもとに、一つのボードに搭載する機能について検討していきます。まず、ハードウェアで実現する部分とソフトウェアで実現する部分を決定する必要があります。非常に複雑な処理、例えば、浮動小数点演算を含む処理などはソフトウェアで実現することのほうが容易で、安価に実現できることが多いようです。ハードウェアですべてを実現するのではなく、性能に見合う範囲でソフトウェアによる実現も考慮に入れておく必要があります。そして、機能の切り分けを行い、部品に汎用LSIを使うのか、ASICを設計するのか、FPGAを用いるのかといった選択を行います。

ボード・レベルにおける機能の切り分けが終わったら、切り分けた機能の依存関係を明らかにしておきます。例えば、LSI間でデータを交換する場合、そのバス構成をどうするか、プロトコル(データ交換の手順)はどうかといった入出力仕様を考えます。

ボード・レベルでは、プリント基板実装技術の知識も必要となります。つまり、ボードの層数やおよそのレイアウトを考えておく必要があります。

第 2 章

FPGA 開発を体験する

米国 Altera 社の FPGA 設計は「Quartus Web Edition(以降,単に「Quartus」と呼ぶ)」ソフトウェアを使って進めていきます。ここではサンプル回路を使って FPGA 設計の一連の流れを体験していただきます。サンプル回路の詳細については,第3章で説明しますので,ここでは何も考えずに,Quartus の使いかたをマスタすることだけに集中してください。

この章を読み始める前に,あらかじめ Quartus をインストールしておいてください。インストール方法は付録 A で説明してあります。

2.1 Quartus による FPGA 開発の流れ

Quartus による FPGA 設計フローを図2-1に示します。

最初に,プロジェクトの作成を行います。これからコンパイルを行う設計データ,ターゲット・デバイスなどを指定します。次に,論理合成と配置配線を行い,FPGAのための回路データ(コンフィ

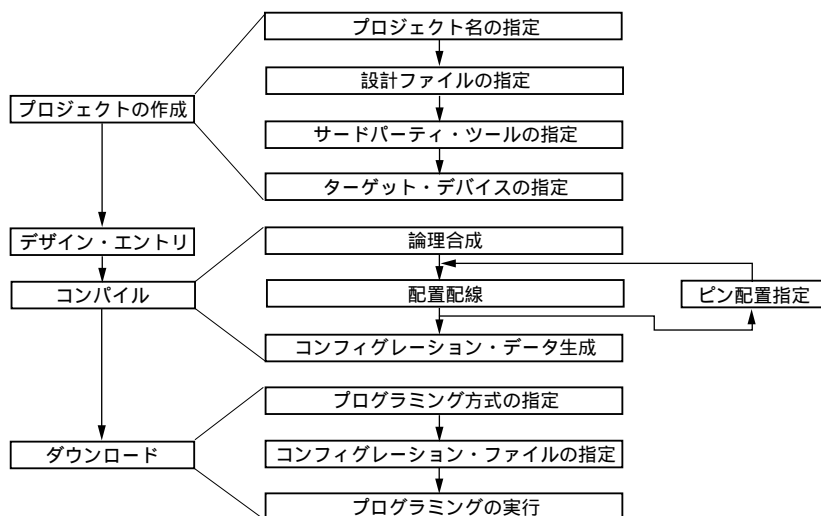


図2-1
Quartus による FPGA
設計フロー

レーション・データ)をFPGAに書き込む(ダウンロードする)ことで、実機による動作確認を行うことができます。

1) プロジェクトの作成

プロジェクトとは、作成する回路を統括する設定のことです。Quartus は、この設定をもとにコンフィグレーション・データを生成します。

プロジェクトには、HDLで記述した回路のソース・コード・ファイルとターゲット・デバイス、ピン・アサインなどのさまざまな情報が記録されます。設計をプロジェクトとして管理することで、設計変更や拡張の際に、細かな設定をやり直さずに済みます。

2) デザイン・エントリ

Quartus には、HDLソース・コードを入力するためのテキスト・エディタの機能があります。それを用いて、HDLデザインを入力していくことが可能です。

3) コンパイル

HDLソース・コードの入力が完了したら、論理合成と配置配線を行います。この一連の処理を「コンパイル」と呼びます。

コンパイルでは、HDLソース・コードの文法、回路規模などのチェックを行い、最終的にコンフィグレーション・データを生成します。

FPGAはI/Oピンの機能のある程度自由に決めることができます。Quartus はピン配置を自動的に決めてくれますが、実際にはこれではFPGAボードの回路と合わないので所望の動作は期待できません。そこで、設計した回路の入出力をFPGAのどのピンに割り当てるかを指定します。

4) ダウンロード

コンフィグレーション・データを作成できたらFPGAに書き込みます。これを「ダウンロード」といいます。

2.2 Quartus の基本操作

それでは、Quartus を使って、FPGA設計の流れを体験してみましょう。

● 2.2.1 Quartus の起動

Quartus がインストールされていれば、デスクトップにアイコンが生成されているはずです。これをダブルクリックするとQuartus が起動し、図2-2のような初期画面が開きます。デスクトップにアイコンがない場合は、Windowsのスタート・メニューから「プログラム」 「Altera」

第 3 章

VHDL による論理回路設計の基本テクニック

本章では、VHDL による論理回路設計の基本を、付属FPGA ボードを使いながら学びます。

ここで説明する内容は、最終目標である野球ゲームを設計するための基本技術要素でもあります。野球ゲームではLED、7セグメントLED、スイッチの三つのキーとなる素子をFPGAで制御します。

3.1 practice1 : LEDの点灯と消灯

LEDは光半導体素子の一つです。アノードとカソードの二つの端子を持ちます。ここに規定値以上の電圧がかかり、アノードからカソードに電流が流れると発光します。

図3-1に、付属FPGAボードのLED周辺回路を示します。LEDのアノードには、FPGAのI/O電圧(V_{ccio})と同じ電圧がかかっています。したがって、FPGAのI/Oピンが“L”のときにLEDが点灯します。“H”のときには消灯します。すなわち、LEDの点灯と消灯を、デジタル信号の“L”と“H”の2値で表現できることとなります。

FPGAボード上の一つのLEDだけを点灯する回路のVHDLソース・コードをリスト3-1(a)に示し

ます。また、FPGAのピン配置を表3-1に示します。FPGAの30番ピンに接続されているLED(D9)を点灯させます。リスト3-1(b)のように、出力する論理値を‘0’から‘1’に変更すると、消灯になります。

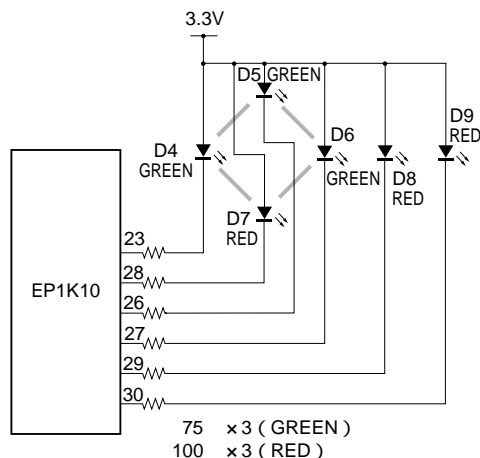


図3-1 付属FPGAボードのLED周辺回路

表3-1 LEDを点灯または消灯する回路のピン配置

ピンの名まえ	ピン番号
led_out	30

リスト3-1 LEDを点灯または消灯する回路のVHDLソース・コード

```

library ieee;
use ieee.std_logic_1164.all;

entity practice1 is
    port (
        led_out : out std_logic);
end practice1;

architecture rtl of practice1 is
begin
    -- LEDはLOWで点灯するように回路が構成され
    -- ているためstd_logic型の0を出力する。
    -- これを1にすると消灯する。
    led_out <= '0';
end rtl;
    
```

(a) LEDを点灯する

```

library ieee;
use ieee.std_logic_1164.all;

entity practice1 is
    port (
        led_out : out std_logic);
end practice1;

architecture rtl of practice1 is
begin
    -- LEDはLOWで点灯するように回路が構成され
    -- ているためstd_logic型の0を出力する。
    -- これを1にすると消灯する。
    led_out <= '1';
end rtl;
    
```

(b) LEDを消灯する

ここで、VHDL記述の基本的事項について説明しておきます。

VHDLの言語構造は図3-2のように、ライブラリ宣言とパッケージ宣言で始まり、エンティティ宣言で入出力ポートを定義し、アーキテクチャ宣言に動作を記述していきます。

“--”に続く文字列はコメントです。何を意図した記述かをコメントとして記述することで、保守性、再利用性が高くなります。

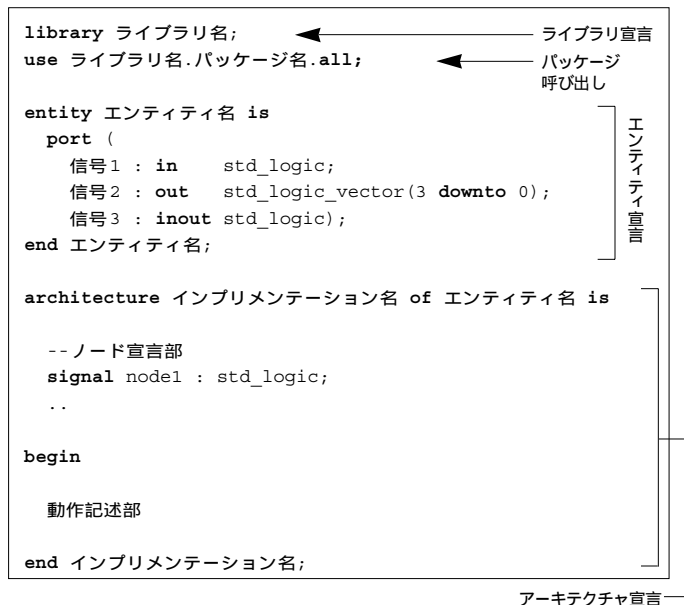


図3-2 VHDLの言語構造

第4章

野球ゲームの設計

いよいよ本書の最終目標である「野球ゲーム」の設計に入ります。野球ゲームの仕様を決め、その仕様を分析し、いくつかの機能モジュールに分割し、VHDLで記述していくようすを説明します。

VHDLによる基本的な記述法は第3章で説明しました。しかしより大規模な回路になる野球ゲームでは、これまで説明しなかった記述法を使うところがあります。これらについては、順次説明していきます。

4.1 野球ゲームの仕様策定

まずは、設計しようとする野球ゲームの仕様を策定します。仕様を作るためには、作ろうとするモノに対するイメージをしっかりと持つ必要があります。具体的にユーザが想定されているときには、お互いが持つ製品イメージに違いがないかといったすりあわせも必要です。

● 4.1.1 動作仕様を考える

はじめに野球ゲームの動作を考えます。

電源投入後、リセット・スイッチを押すことでゲームが始まります。最初の状態は、チーム0の攻撃になります。当然、ノー・アウト、ランナーなしです。

打撃スイッチを1回押すと、打撃ルーレットが回り始めます。もう一度、打撃スイッチを押すことで、打撃ルーレットが止まります。このときの出目に従い、アウト、ヒット、2塁打、3塁打、ホームランのいずれかが決まります。

打撃ルーレットの出目によってランナーの進塁を行います。盗塁やタッチアップなど複雑なルールはなしとします。ランナーは塁打どおりに進塁すると仮定します。例えば、1塁打が出たときは、ランナーは1塁となります。そのあと2塁打が出たときは、ランナーは1塁と3塁にいる状態にします。イチローのような強肩野手、または、俊足ランナーは存在しないことにします。得点が入ると、チーム0の得点がホームインしたランナー数によって加算されます。

出目がアウトだった場合、アウト表示が1アウトの状態になります。3アウトになると、チーム0の攻撃は終了し、チーム1の攻撃となり、先ほど説明したチーム0の例のように、攻撃が始まります。

表示はLEDを使用する。ランナーの出塁状態を表すLEDを三つ、アウト・カウントを表すLEDを二つ、打撃ルーレットのための7セグメントLEDを一つ、点数表示のための7セグメントLEDを二つ使用する。

得点表示は16進数1けたで“0”から“F”までとする。

スイッチとして、リセット・スイッチと打撃スイッチをそれぞれ一つずつ使用する。

チームは2チームとする。点数表示用7セグメントLEDのドット表示で表現する。

ルーレットの出目の種類と表示は、アウトが“0”，1塁打が“1”，2塁打が“2”，3塁打が“3”，ホームランが“H”とする。

ルーレットは打撃スイッチを1回押すことでスタートし、もう1回押すことでストップする。

リセット・スイッチを押すと、得点表示は両チームとも0，ルーレットはストップ状態，チームは0の攻撃状態，アウト表示用LEDはすべてOFF（ノー・アウト），出塁状態LEDはすべてOFF（ランナーなし）とする。

ルーレットが停止した際にアウトが出たら，アウト・カウントを+1し，アウト表示用LEDを1アウトの場合は一つ，2アウトの場合は二つ表示する。

アウト表示用LEDが二つともONのとき（2アウトのとき）にアウトが出たら，チームを変更し，アウト表示用LED，出塁状態を表すLEDをすべてOFFにする。

出塁状態LEDは進塁状態によりON/OFFする。例えば，ランナーが1塁，3塁の場合は，1塁と3塁のLEDをONにし，2塁のLEDをOFFにする。

ランナーは塁打数どおりに自然に変更する。例えば，1塁にランナーがいる際に2塁打が出ると，出塁状態は2塁，3塁となる。

ホームインしたランナーの数に応じて，点数を加算する。例えば，2塁，3塁にランナーがいる場合に2塁打が出たときは2点が加算され，ランナー2塁となる。

図4-1 野球ゲームの概略仕様

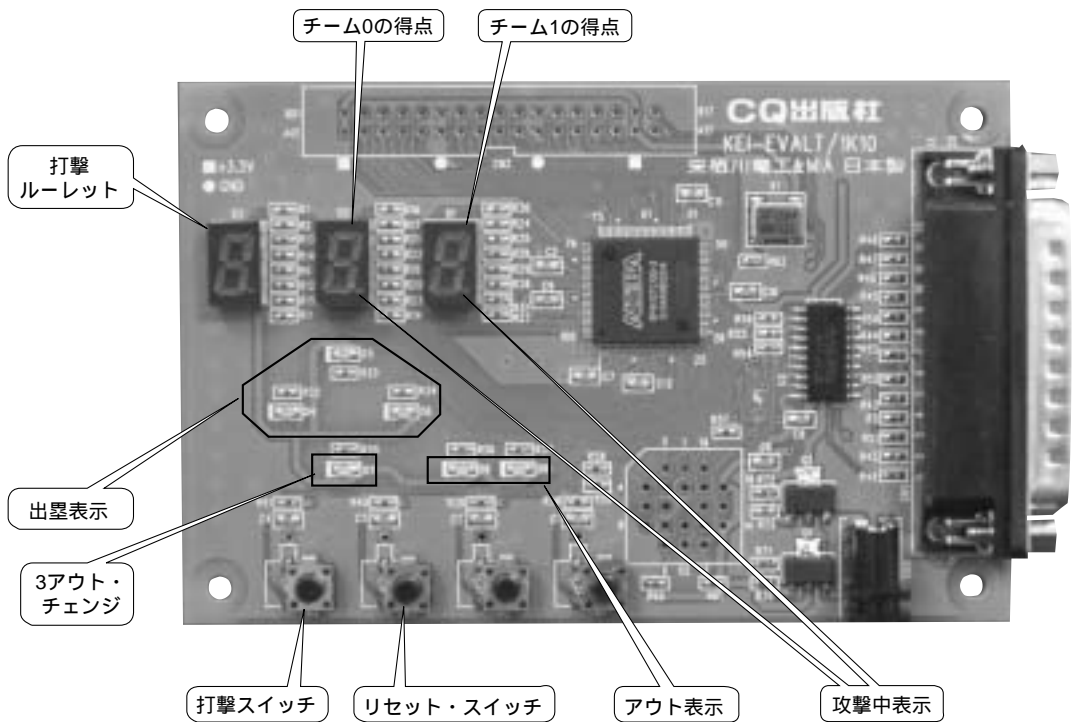


写真4-1 野球ゲームの動作イメージ