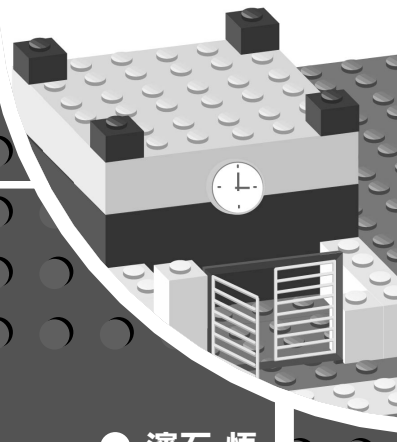


3 無償IPコアの活用事例
—— I²C ROMライタの製作

● 濱石 悟

本稿では，無償で利用できるIPコアの活用事例を紹介する．製作するアプリケーションはI²Cインターフェースのシリアル ROMライタである．無償のI²Cインターフェース・コアと評価版のCPUコアを活用した．本稿を参考に，IPコアを利用した設計を体験してみてほしい．（編集部）

FPGAのゲート数が大きくなるにつれて，さまざまな機能を1チップで実現できるようになっています．これに伴い，LSI設計者のしごとの担当範囲が広くなっています．製品の開発サイクルも短くなってきており，設計者の負担は増えるばかりです．

このような状況の中，「再利用できる機能モジュールがあれば効率が上がるのになぁ」と感じているのは筆者だけでしょうか？ 市販のIPコアを活用できれば非常にありがたいと感じるときがあります．IPコアをじょうずに活用し，開発期間を短縮したり開発リスクを低減できれば，設計者にとっては喜ばしいかぎりです．しかし，実際には，IPコアは非常に高価なイメージ（あくまでも筆者のイメージだが…）があり，購入費用の問題や，LSI化したときに所望の性能を引き出せるのか，ほんとうに開発リスクを回避できるのかという不安があります．また，IPコアをどのように活用すれば効果的なのか，悩んでいる方もおられるかもしれません．

本稿では，無償のIPコアを活用したROMライタの設計事例を通じて，IPコアを使ううえでの注意点などを解説します．読者のみなさんも，ぜひ実際にIPコアを使ってみてください．

I²Cインターフェースについて

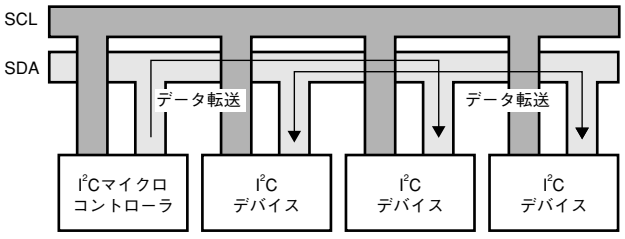
本稿ではI²Cインターフェースを持つシリアルROM向けのROMライタを製作します．まず，基本となるI²Cバス規格の概要を説明します．

●I²C仕様の概要

I²Cは，Inter IC Busの略称であり，オランダPhilips社が提唱し業界標準になったシリアル・バス規格の一つです．I²Cバスは非常にシンプルな構造で，シリアル・クロック（SCL）とシリアル・データ（SDA）の2線だけで双方向通信を行います（**図1**）．SDAとSCLは，いずれも双方向信号であるため，オープン・ドレイン出力にし，プルアップしなければなりません．

I²Cバスの特徴を以下にまとめます．

- シリアル・データ・ライン（SDA）とシリアル・クロック・ライン（SCL）の2本のバス・ラインを使う
- 各デバイスは，それぞれ固有のアドレス（スレーブ・アドレス）を持ち，アドレス指定を行える
- データ転送はマスタとスレーブの間で行う
- バスの衝突検出機能およびアービトレーション機能を備えたマルチマスタ・バスである



〔図1〕I²Cバスの構成

●複数の通信速度に対応している（標準モード：100kbps，ファースト・モード：400kbps，ハイスピード・モード：3.4Mbps）
●バス上にいくつものデバイスを接続できる（ただし，バスの静電容量は400pF以下）

I²Cバス規格の資料は，Philips社のホームページからダウンロードすることができます．英文の仕様書というと敬遠される方もいらっしゃるかもしれません．しかし日本語の解説書も用意されているので安心してください（http://semicon.philips.co.jp/products/pamphlet/pdf/i2c/i2cbus.pdf）．

I²CバスはPhilips社がライセンスを保有しています．本稿執筆にあたり，ライセンスに関して問い合わせてみました．結論から言えば，今回の事例についてはライセンスは不要です．しかしI²Cバスのライセンスはかなり複雑なようで，日本フィリップスのことばを借りれば，「そのつど，ライセンスが必要かどうか確認していただきたい」とのことです（下掲のコラム「I²Cのライセンスについて」を参照）．

●通信手順

I²Cバスのマスタとスレーブの間で通信する手順を説明します．タイミング・チャートを図2に示します．

1）バス使用権の取得（START信号の生成）

I²Cバス上でデータを転送するときは，最初にSTART信号を生成します．SCLが“H”のときにSDAを“H”から“L”にするとSTART信号になります．バスに接続されているデバイスがSCLを使用していないとき（つまり“H”），SDAを変化させることで，バス使用権を獲得します．

2）スレーブ・アドレスの設定

I²Cバス上には複数のスレーブ・デバイスを接続することができます．これらのデバイスを指定するために，バス使用権を獲得したデバイス（マスタ）は，第1バイト内の7ビットを用いて通信相手（スレーブ・デバイス）のアドレスを指定します．また，残りの1ビットはデータ転送の方向（リードかライト）を指定します．

すべてのスレーブ・デバイスは，I²Cバス上に流れてきたスレーブ・アドレスと自分のアドレスを比較し，もし一致すればACK（アクノリッジ・クロック・パルス）信号で応答します．

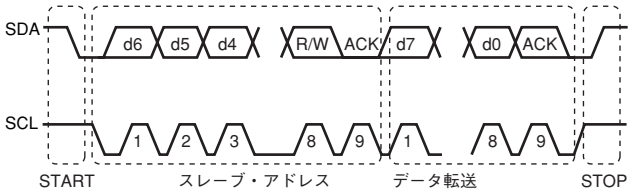
3）データの転送

スレーブ・デバイスがACK信号で応答すれば，データ転送フェーズに移ります．

データ転送時においては，各バイトごとに正常に受信できればACK信号で応答し，異常ならNACK信号で応答します．このデータ転送フェーズを繰り返すことにより連続したデータ転送を行うことが可能です．

4）終了（STOP信号の生成）

I²Cバス上のデータ転送を終了するときは，STOP信号を生成します．SCLが“H”の状態のとき，SDAを“L”から“H”にするとSTOP信号になります．



〔図2〕I²Cバスのタイミング図

column　I²Cバスのライセンスについて

筆者は，I²Cバスのライセンスについて，スレーブ機能を実現する場合は，スレーブ・アドレスをPhilips社から入手しないといけないのでライセンスが必要であり，マスタ機能のみを実現する場合はライセンスはいらないと思っていました．しかし，Philips社にライセンスの問い合わせを行ったところ，マスタ機能のみを実現する場合でもライセンスが必要になるケースがあることがわかり，認識の甘さに恥ずかしい思いをしました．

I²Cバスのライセンスについてまとめると，以下のようになります．

1）I²C対応のシリアルROMやI²C対応のデバイスを制御する際，FPGAでI²Cのマスタ機能（シングル・マスタ機能のみ）を実現する場合はライセンスは必要ない．
2）I²C対応のIPコアを使用してFPGAにマスタ機能を組み込む場合も，ライセンスは必要ない．

結論として，本稿のケースではライセンスは必要ないということでした．しかしライセンスの条件は複雑であり，基本的にはケース・バイ・ケースで判断しているそうです．I²Cバスを使うときにはPhilips社にライセンスの確認をしたほうがよいでしょう．