

①解説書(600頁) ②MAX 10®
③ライター ④電子部品 ⑤開発用DVD入り

この1冊で
入門から
実践まで



完全版FPGA電子工作 オールインワン・キット

圓山 宗智 著

1

600頁の解説書

MAX 10® 実験キットで学ぶ FPGA&コンピュータ

MAX 10®の動かし方から、論理設計の基礎、ソフトウェアIPプロセッサ NiosII®の組込み方などを実験解説。Linuxボードラズベリー・パイとの連携やインテル®の名作CPU 4004の製作にも挑戦

この解説書は単体(本体価格:6,000円+税)でも購入できます



2

ワンチップFPGA基板MAX10-FB



MAX 10®
(10M08SAE
144C8G
アルテラ製)

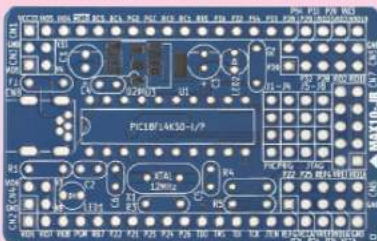
- ロジック・エレメント数 8000
- FLASHメモリ (コンフィグ 2496Kbit, ユーザ 1376Kbit)
- A-Dコンバータ (12bit, 1MSPS)

最大512Mbit
SDRAM搭載部
(256Mbit品が同梱)

3

ライター製作用プリント基板
MAX10-JB

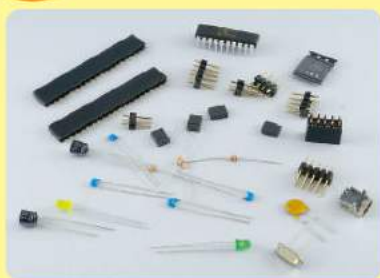
純正ライター USB-Blaster等価



本プリント基板に同梱の部品を搭載して初期状態のMAX10-FBを接続し電源を投入すると、MAX 10®からPICマイコンのFLASHメモリにファームウェアが自動的に書き込まれてライターとして機能する

4

製作用電子部品一式



* 写真はMAX10-JB用です

- 256Mbit SDRAM (AS4C16M16SA)
- PICマイコン(PIC18F14K50)
- 水晶発振器(SG8002DC48M)
- 3色カラーLED, ピン・ヘッダ ほか

5

FPGA開発用
DVD-ROM

[ソフトウェア/IP]

- Quartus® Prime Lite Edition/Qsys
- ModelSim Altera® Starter Edition
- NiosII® 32ビットCPUコア
- NiosII® EDS(統合開発環境)

[サンプル・プロジェクト]

- NiosII®用ファームウェア (A-D変換/SDRAM制御ほか)
- C言語混在シミュレーションとIP設計
- Linux I/Oボードラズベリー・パイ連携
- インテル® 4004システムの論理回路と電卓製作
- GPS/OLED/モジュールほか
- 3cm角モジュール用ファームウェアほか

ご購入はCQ出版社のホームページから

<http://shop.cqpub.co.jp/hanbai/books/48/48061.htm>

CQ出版社

見本

第16章 論理設計を知り、味わい、そして楽しむ

論理設計入門

●はじめに

自分の思い描くシステムを構築するには、ハードウェアとソフトウェアの知識と経験が必要です。

ソフトウェアについては入手しやすいマイコン評価基板があるので触れるチャンスも多く、多くの方が経験を積みながら腕を上げていると思います。

一方、ハードウェアの製作は、一般的には壁が高いです。アナログとデジタルが混在したミクスト・シグナル型のシステムの場合、アナログ回路なら多くのケースは比較的小規模であり、OP アンプや A-D 変換器など市販の IC を組み合わせて実現できる範囲にあるでしょう。

デジタル回路(論理回路)側は、汎用マイコンで済む範囲なら簡単ですが、特殊で高性能かつ高機能な論理回路が必要になるケースだと一般的に規模が大きくなります。これを TTL などの標準ロジック IC をたくさん組み合わせて大量の配線をする職人技で実現するのは、今の世の中では現実的ではありません。

MAX 10 は論理設計の経験を積む素晴らしいプラットフォームです。手配線は要りません。RTL コーディングでデジタル回路が完成します。この FPGA を使って思う存分論理設計して、現実動作するシステム開発を数多く経験して腕をメキメキ上げていきましょう。

そのためには、何はともあれ論理設計の基礎について学ぶ必要があります。本稿では必要最小限ではありますが、論理設計のキモになるところを解説します。本書で必要になる知識はすべて網羅していきますの

で、じっくり理解してください。すでに論理設計を十分知り尽くしている方は、読み飛ばしても構いません。

論理設計を始める前に

大きな流れ

●第1歩は入出力インターフェースと内部機能仕様の決定

まずは論理設計とはなんぞや、から説明します。

図1にそのイメージを示します。実現したいデジタル機能について、まず、入出力インターフェースを明確に定義します。また、内部機能仕様を明確に矛盾なく定義します。

論理機能モジュールの例を表1に示します。ロジカルに矛盾のない入出力インターフェースや内部仕様を、早い段階でしっかり規定することがとても重要で、ここの失敗による手戻りは全体の設計工数に多大な悪影響を及ぼします。

●仕様が決まった後の設計の流れ

次に、内部ハードウェアの実現方式(アルゴリズム)を規定します。ここが最も頭を使うところで、機能面、タイミング面、経済面(ハードウェア量、消費電力)を見渡しながら、最適な方式を決定します。

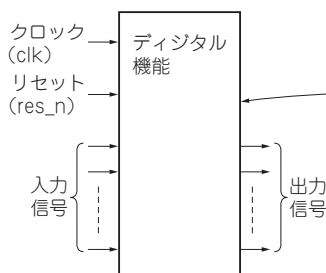


図1 論理設計とは

論理設計とは

- 入出力インターフェースを明確に定義
- 内部機能仕様を明確に矛盾なく定義
- 内部ハードウェア方式(アルゴリズム)の規定
— 機能面/タイミング面/経済面を見渡して
- 組み合わせ回路/順序回路による論理回路の組み立て
- 網羅的な機能検証
- 論理合成とタイミング検証

第17章 論理設計の道具を自分のものにしよう

Verilog HDLによるRTL記述入門

●はじめに

本章ではハードウェア記述言語の代表格、Verilog HDLによるRTL記述方法の基礎を説明します。本稿の内容が理解できれば、ほとんどの論理回路は問題なく記述できると思います。

ハードウェア記述言語 「Verilog HDL」とは

●Verilog HDLとは

論理回路を抽象的に記述するためのハードウェア記述言語としては、Verilog HDLとVHDLが有名です。歴史的には、まずVHDLが米国国防総省によって開発されました。VHDLは記述量が多いのですが、明確に機能仕様を定義できる特徴があります。その後、論理シミュレータと一体化した言語としてVerilogが開発されました。Verilogは記述量がVHDLよりも少なく手軽にコーディングできますが、論理シミュレータ側の視点で定義された言語であり、物理的な論理回路を確実に記述するには多少の注意が必要です。

Verilogは歴史的に複数のバージョンが定義されました。本稿ではVerilog 1995とVerilog 2001を説明します。Verilog 2001はVerilog 1995の改良版です。現在では、さらに改良され多くの機能が加えられたSystem Verilogも普及しています。System Verilogを使った機能検証方法については、本書内で改めて触れる予定です。

●Verilog 1995とVerilog 2001

基本的に、アルテラ社が提供する開発環境Quartus PrimeやModelSim-AlteraではVerilog 1995もVerilog 2001も共に使えます。もちろんSystem Verilogもサポートしています。

本稿で解説するVerilog 1995とVerilog 2001の間は、さほど大きな差はありませんが、Verilog 2001のほうが組み合わせ回路の記述方法が楽になる利点があります。Verilog 1995型の記述とVerilog 2001型の記述は、一つのRTL記述の中で混在可能です。Verilog

1995とVerilog 2001の記述方法で意識すべき差異は、この後に説明します。

論理機能のモジュール構造記述

●モジュール構造と階層構造の記述

論理機能はモジュールという単位で記述します。リスト1(a)のように、論理機能はmodule文とendmodule文で挟み、入出力信号と内部機能を記述します。

モジュールは一般的に階層構造をもちます。例えば、一つのLSIの中にCPUやDMACが内蔵されている場合、LSIという階層の下にCPUとDMACがあります。さらにCPUの下にも、例えばDATAPATHとCONTROLという階層が置かれるのが通常でしょう。

●インスタンス化

あるモジュールの下で階層として、別のモジュールを配置する場合の記述方法をリスト1(b)に示します。

ここで重要な概念があります。一つのモジュールは、あちこちで使いまわされることが一般的です。例えば、デュアル・コアをもつLSIの場合は同じCPUが2個あるでしょうし、2入力NANDゲートをモジュールとして定義すれば、LSI内で同じものが大量に使われます。

このとき、ある階層内に置く個々の下位階層モジュールを区別するため、その階層内で固有となるインスタンス名を付けて、その下位階層モジュールを置きます。これをモジュールのインスタンス化(instantiation)といいます。インスタンス化というのは「実体化・具体化」という意味です。module文に記述するモジュール名は、定義名にすぎないことに注意してください。

原則として各モジュールは必ずその上位階層でインスタンス化されます。それではLSIやFPGAの外部入出力信号が置かれる階層(LSIやFPGA)によっての

※ 本稿は、『ARM PSoCで作るMyスペシャル・マイコン(開発編)』(2014年1月、CQ出版社)の第19章「論理設計入門」を参考に、内容を大幅に改訂・加筆したものです。

第18章 単純なレジスタの動作からCPUまで、 実際の設計と論理シミュレーションにトライ

論理設計の具体例とシミュレーション

本書付属DVD-ROM収録関連データ		
DVD-ROM格納場所	内 容	備 考
CQ-MAX10¥Verilog_Samples¥simple_register	レジスタの設計ファイル一式	ModelSim-Altera用、Icarus Verilog用 がそれぞれ格納されている
CQ-MAX10¥Verilog_Samples¥simple_counter	カウンタの設計ファイル一式	
CQ-MAX10¥Verilog_Samples¥simple_debouncer	チャタリング除去回路の設計 ファイル一式	
CQ-MAX10¥Verilog_Samples¥simple_statemachine	ステート・マシンの設計ファ イル一式	
CQ-MAX10¥Verilog_Samples¥simple_cpu	簡易型8ビットCPUの設計フ ァイル一式	

●はじめに

前章までの座学だけでは、実際にどのようにRTL記述を書いていけばよいのかわかりにくいかもしれません。そのため本章では具体的に動作するRTL記述を紹介し、実際に論理シミュレーションして動作確認してみます。本章の内容が理解できれば、さらに複雑な論理機能も自由に設計することができるようになるでしょう。

●サンプル・ファイルをインストール

まず、付属のDVD-ROMに格納したサンプル・ファイルをインストールしてください。Cドライブの最上位階層にディレクトリ「C:¥CQ-MAX10」を作成して、DVD-ROMのトップ階層以下の「¥CQ-MAX10¥Verilog_Samples」を、その「C:¥CQ-MAX10」以下にコピーしてください。

ディレクトリ「Verilog_Samples」が、「C:¥CQ-MAX10¥Verilog_Samples」の位置にあればOKです。本書の解説では、この位置にサンプル・ファイル類が置かれていることを前提としています。

論理シミュレータModelSim-Altera を単体で使う方法

●論理シミュレータ ModelSim-Altera を単体で 使う

前章までは、ModelSim-AlteraをQuartus PrimeやNios II EDSから起動する方法を説明しました。本章ではModelSim-Alteraを単体で使う方法について

説明します。

●設計ファイルのディレクトリ階層構造

論理シミュレーション対象の設計ファイルのディレクトリ階層構造を統一化しておくと、同じシミュレーション環境を使い回せるので便利です。本稿で使う設計サンプルのディレクトリ階層構造を図1に示します。各ファイルの内容などは、個々のサンプルのところで説明します。

●立ち上げとプロジェクト作成

本稿のサンプルを論理シミュレーションする場合のModelSimの使い方を図2に示します。

図2(a)のようにModelSim単体を起動して、図2(b)、図2(c)のように新規プロジェクトを作成します。サンプル・ファイルをそのまま使う場合は、プロジェクト・ファイルをオープンしてください。

●プロジェクトにファイルを登録

図2(d)～(h)に示す方法でプロジェクトに関連ファイルを登録します。独自の論理設計を行う場合は、新規の空ファイルを登録してください。サンプル・ファイルをそのまま使う場合は、既存ファイルの登録を行ってください。

●ファイルを編集

図2(i)に示すように各ファイルを編集してセーブします。実行スクリプトなどにディレクトリの相対パスを記述する場合は、カレント・ディレクトリ(相対

第19章 タイミング解析の基礎を学び、SDCファイルを自在に書けるようになる

TimeQuest Timing Analyzer によるタイミング解析とSDCファイル

● はじめに

論理設計においては、その動作周波数や入出力遅延時間などのタイミング設計が不可欠です。アルテラ社のFPGA開発ツールQuartus Primeには、設計した回路のタイミング解析ツールとしてTimeQuest Timing Analyzer(以下、TimeQuestと略す)が組み込まれており、簡単にその強力な機能を使うことができるようになっていました。本書のこれまでの章でもTimeQuestの使い方をいくつか紹介してきました。

本章ではあらためてTimeQuestによるタイミング解析の基本と、タイミング制約SDC(Synopsys Design Constraint)の書き方を詳しく解説します。本章を理解することで、安定に動作するロバストな(堅牢な)論理回路を設計できるようになるでしょう。さらに、SDCの書き方は業界標準であり、他のFPGAやSoC設計にも適用できる内容がほとんどです。ここで学んでも損はしません。

TimeQuestによるタイミング解析の基本概念

● TimeQuestによるタイミング解析は静的かつ網羅的

TimeQuestは、設計対象の下記のタイミングを網羅的に解析します。

- レジスタ(D-F/F)とレジスタ(D-F/F)の間
- 入力信号経路
- 出力信号経路
- 非同期リセット信号経路

このためにTimeQuestは、データ必要時刻(Data Required Time)、データ到達時刻(Data Arrival Time)、クロック到達時刻(Clock Arrival Time)を使って、設計対象のタイミング違反や性能を、静的(static)に解析します。静的解析という意味は、論理シミュレーションのように信号を動的(dynamic)に動作させずに、回路の接続構造(トポロジ)だけを見て解析することです。動的解析よりも網羅的な解析ができる特長があります。静的なタイミング解析のことをSTA(Static Timing Analysis)と呼びます。

このタイミング解析には、FPGA内のロジック・アレイ、メモリ、配線などの物理的な回路要素の遅延情報をもったライブラリを、内部的に参照します。こうしたタイミング解析用のライブラリは、FPGAベンダやSoCベンダ、あるいは半導体製造のファウンダリから提供されています。

● TimeQuest 解析の用語

TimeQuest解析で使われる用語が、いくつか定義されています。表1にその内容をまとめておきます。

表1 TimeQuest解析の用語

用語		定義内容
和文	英文	
ノード	node	タイミング・ネットリストの基本単位。ポート、ピン、レジスタを表す
セル	cell	LUT(Look Up Table)、レジスタ、DSP(Digital Signal Processor)、メモリ・ブロック、入出力要素など
ピン	pin	セルの入力または出力
ネット	net	ピン間の結線
ポート	port	モジュールの最上位階層の入力または出力(例: デバイスの端子)
クロック	clock	設計対象の内部または外部に置かれるクロック・ドメイン(領域)を表す抽象オブジェクト
経路	path	信号が伝搬する経路
送信エッジ	launch edge	レジスタ出力を変化させるクロック・エッジ
受信エッジ	latch edge	レジスタ入力をラッチするクロック・エッジ

見本

第20章 C言語とVerilog HDLの混在シミュレーションを使ってAvalon-MMスレーブIPを設計しよう

C言語混在シミュレーションとIP設計

本書付属DVD-ROM収録関連データ

DVD-ROM 格納場所	内 容	備 考
CQ-MAX10¥Projects¥PROJ_DPI	C言語とVerilog HDLの混在シミュレーションの基本 (Quartus Prime, ModelSim用)	
CQ-MAX10¥Projects¥PROJ_MM-Slave	Avalon-MMスレーブIPの設計事例: pic_programmer用SPI機能 (Quartus Prime, ModelSim用)	このプロジェクトは、バス・マスタの機能モデルにSRAMだけ接続する検証にも使うが、DVDに格納したのはpic_programmerまで接続した完成形である。FPGA_completedフォルダは無視してよい

●はじめに

本章では、Nios IIなどのCPUシステム内の周辺機能、すなわちスレーブIP(Intellectual Property)を設計するための手法を学びます。

まず、C言語とVerilogの混在シミュレーション手法DPI(Direct Programming Interface)を説明します。論理シミュレーションのテストベンチにC言語記述を組み込むことができるので、最終的なアプリケーションと同じプログラムを使って論理検証できるようになり、作業効率が向上します。

次に、Nios IIのスレーブIPを設計するのに必要な知識として、Avalon-MMインターフェース(内部バス)の信号とタイミング仕様を説明します。

そして、C言語から制御してAvalon-MMマスタにバス・アクセスさせてRAMのリード/ライト動作をさせてみます。この時点で、Quartus Prime, QSys, ModelSim ASE(Altera Starter Edition)を総合的に連携させて、Avalon-MMスレーブIPの設計と検証を進める手法を学べます。

最後に、本書の第5章で説明した、PICマイコンのFLASHメモリ書き込み用の特殊なSPI通信モジュールpic_programmerを設計して、C言語混在シミュレーションにより機能検証してみます。

C言語とVerilogの混在シミュレーション技法「DPI」

●実はもうやっていたC言語混在シミュレーション

本書の読者の方は、実はすでにC言語とVerilogの混在シミュレーションを行っていました。第12章の

Nios IIによるシステム設計の中でです。

Nios IIのソフトウェア統合開発環境Nios II EDSの中から論理シミュレータModelSim ASEを起動して、ソフトウェアをコンパイルしたバイナリ・ファイルによりシステム内のプログラム・メモリの内容を初期化することで、Nios II CPUコアを含めたシステム全体の論理シミュレーションを実行していました。

この手法はシステム全体をまるごと検証できるのでとても有効なのですが、C言語のスタートアップ・ルーチン(メモリ内の変数領域の初期化など)も実行されるためサイクル数が長く、シミュレーション時間が長くなる問題があります。

さらに、CPUコアを含めた大規模論理をシミュレーションするので、無償版のModelSim ASEだと実行速度が遅く、PCのメモリや波形記憶用のストレージ容量も多く消費してしまいます。

もちろん、最終的にシステムが完成してその全体を検証する段階に至れば、上記の手法はとても有効なのですが、検証対象のIPを設計しながらシステム全体を検証する設計工程の前段階では、繰り返して論理シミュレーションしたいので、時間がかかる上記の手法は実用的ではありません。

●C言語混在シミュレーションの意義

IPを設計しながら効率的に論理シミュレーションするには、Verilogだけでテストベンチを作ればよい話です。

しかし、そのIPの機能を動作させるためにシーケンス処理やループ処理(フラグがセットされるまで待つなど)が必要なケースでは、テストベンチをVerilog

見本

第21章 Raspberry Piのハードウェア機能拡張と、MAX 10のコネクティビティ強化を両立する

MAX 10とRaspberry Piを接続する 拡張用MAX10-EB基板のハードウェア詳説

本書付属DVD-ROM収録関連データ

DVD-ROM格納場所	内 容
CQ-MAX10¥Board¥MAX10-EB	- MAX10-EB基板のガーバ・データ - 関連ドキュメント

●はじめに

Raspberry PiにMAX 10 FPGAを搭載したMAX10-FB基板を接続して、さまざまな実験やシステム構築をするための拡張基板MAX10-EB(EBは、Expansion Boardの略)を、別売りで提供します。本章では、このMAX10-EB基板のハードウェアについて詳しく解説します。

MAX10-FB基板に接続できるRaspberry Piとしては、Raspberry Pi 2 Model BおよびRaspberry Pi 3 Model Bのいずれにも対応しています。本書内でRaspberry Piと記述があれば、Raspberry Pi 2 Model BまたはRaspberry Pi 3 Model Bのいずれかのことを指します。

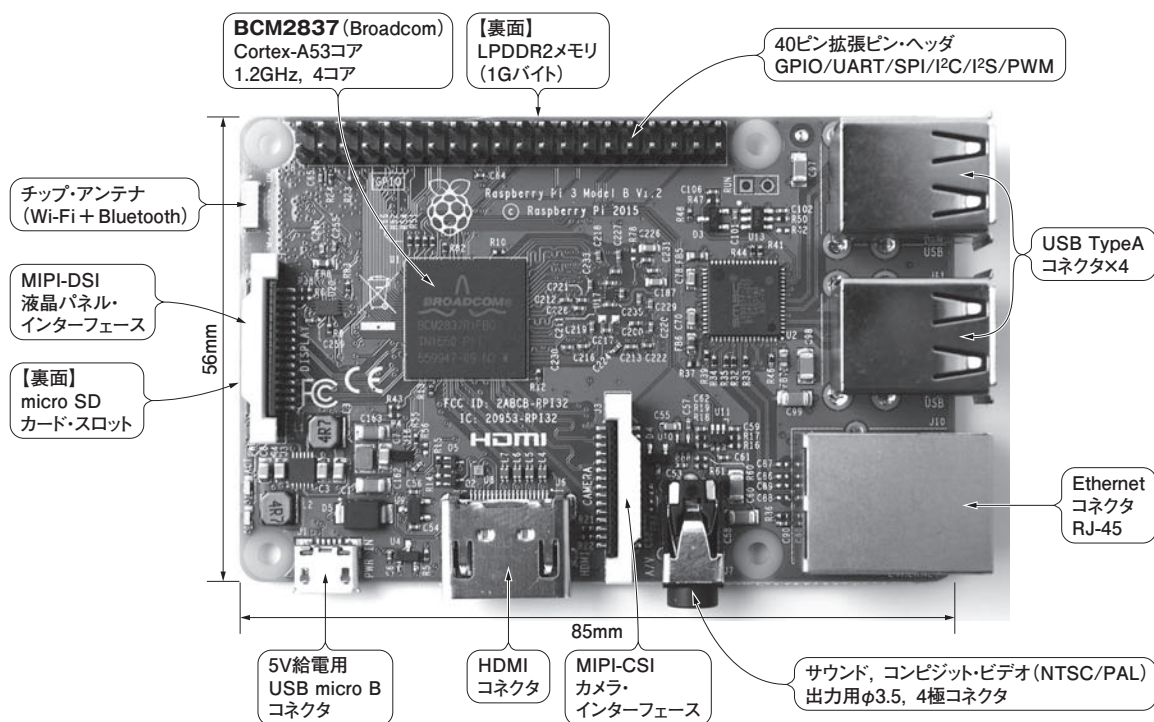


写真1 Raspberry Pi 3 Model Bの外形

Raspberry Pi 3は、基板上に無線機能(Wi-FiとBluetooth)が搭載されて使い勝手が向上した

見本

第22章 高速SPI通信によるインターフェースと、Qt CreatorによるGUIアプリの作成

MAX 10とRaspberry Piの連携方法

本書付属DVD-ROM関連データ		
DVD-ROM格納場所	内容	備考
CQ-MAX10¥Projects¥PROJ_NIOSII_RASPI	Raspberry Piと通信しながら連携動作するMAX 10のプロジェクトのひな形(Quartus Prime用, Nios II EDS用)	MAX 10側
CQ-MAX10¥RaspberryPi¥CQ-MAX10.tar.gz	① Raspberry Piのホーム・ディレクトリ下に作業用ディレクトリ/home/pi/tempを作成し、本ファイルをその下に置いて解凍(tar xvzf CQ-MAX10.tar.gz)する。その後、解凍してできたディレクトリのうち/home/pi/temp/CQ-MAX10/MAX10を、/home/pi/CQ-MAX10の下にコピー ② /home/pi/CQ-MAX10/MAX10が、MAX 10とのSPI通信テスト用GUIアプリケーションのプロジェクトのひな形(Raspberry Pi上のQt Creator用)	Raspberry Pi側(本章では、このプログラムを作成する具体的な手順を解説する。本データは参考用の完成版)

●はじめに

Raspberry PiとMAX 10(MAX10-FB基板)を組み合わせて動作させることができるMAX10-EB基板のハードウェアについて、前章で解説しました。本章では、Raspberry PiとMAX 10の間の具体的なインターフェース方法について説明します。

さらに、Raspberry Pi側のアプリケーション・プログラムの作成方法と、MAX 10側のFPGAハードウェア、およびMAX 10内のNios II用プログラムの構成例についても説明します。本章の技術情報をマスタすれば、オリジナルのシステムを構築できるようになるでしょう。

本章で説明するRaspberry Piとしては、Raspberry Pi 3 Model Bを使用することを前提としています。その立ち上げと設定方法はAppendix 3を参照してください。なお、基本的に本章の内容はRaspberry Pi 2 Model Bにも適用可能です。

Raspberry Pi 3とMAX 10間のインターフェース方法

●Raspberry PiとMAX 10のインターフェースはSPI通信

Raspberry PiとMAX 10の間のインターフェース信号としては、Raspberry Piの拡張基板を組み合わせることも考えると、それら拡張基板類があまり使っていない信号を、なるべく少ない本数だけ使うべきと考えました。

ここではインターフェース信号として、図1に示す4線式SPI(Serial Peripheral Interface)通信信号だけを使うことにします。

Raspberry Pi側をSPIマスタ、MAX 10側をSPIスレーブとします。

SPI1_CE2_Nはスレーブ選択信号(負論理)、SPI1_SCLKはSPIクロック、SPI1_MOSIはマスタ側送信データおよびスレーブ側受信データ、SPI1_MISOはスレーブ側送信データおよびマスタ側受信データです。

MAX10-EB基板の上のはんだジャンパを使って、図1に示す4本の信号が、Raspberry PiとMAX 10の間に接続されるようにしてください。

●Raspberry Pi上のSPI機能モジュールはAUX SPI0を使う

図1に示す信号線を使用する場合、Raspberry Pi側のSPI機能モジュールは、メインのSPI機能モジュールではなく、補助(Auxiliary)周辺機能の中のSPI機能になります。補助周辺機能の中には1組のMini UARTと2組のSPI(AUX SPI0とAUX SPI1)が入っていますが、ここではAUX SPI0だけを使います。

Raspberry PiのGPIO端子上の名称としては、AUX SPI0とAUX SPI1は、それぞれSPI1とSPI2に対応付けられているので混同しないようにしてください。ここで使用するRaspberry Pi上のSPIモジュールは、SPI1すなわちAUX SPI0になります。

見本

第23章 インテルMCS-4アーキテクチャをじっくりと堪能して、先人の知恵の深さを感じとろう

4004 CPUアーキテクチャとMCS-4システム

本書付属DVD-ROM収録関連データ	
DVD-ROM格納場所	内容
CQ-MAX10¥RaspberryPi¥ CQ-MAX10.tar.gz	Raspberry Piのホーム・ディレクトリ下に作業用ディレクトリ/home/pi/tempを作成し、本ファイルをその下に置いて解凍(tar xvfz CQ-MAX10.tar.gz)する。その後、解凍してできたディレクトリのうち/home/pi/temp/CQ-MAX10/MCS4を、/home/pi/CQ-MAX10/の下にコピー /home/pi/CQ-MAX10/MCS4/ADS4004が、4004 CPUの2パス・アセンブラ、逆アセンブラ、シミュレータ(標準Cで記述)

●はじめに

これまでいろいろなことを長々と解説してきましたが、その集大成として、世界最初のマイクロコンピュータであるインテル社4ビット・マイクロコンピュータ・システムMCS-4の4004(CPU)および、そのチップ・セット一式をVerilog HDLで論理設計し、MAX 10に実装したいと思います。

その前にまず本章では、4004をはじめとするMCS-4システムのアーキテクチャ全体を詳細に解説します。さらに4004のプログラム開発用アセンブラやシミュレータを自作したので、それについても説明します。これで今日からあなたも、4004の名プログラマになります。

●4ビットに慣れよう

4004は4ビットCPUなので、扱うデータは基本的には4ビット単位です。よって、データのアドレスは4ビット単位でアサインされます。今どきのマイコン少年～中年の皆さんは、8ビット単位のデータとアドレスには慣れていると思いますが、4ビット単位になると少し違和感を感じるかもしれません。ぜひ、その違和感を快感として楽しんでみてください。なお、命令コードは8ビット単位なので、そのアドレス・アサインは8ビット単位になっています。

MCS-4 チップ・セット

●MCS-4 チップ・セット

4ビット・マイクロコンピュータ・システムを組むためのチップとしては、4004(CPU)、4001(ROM)、

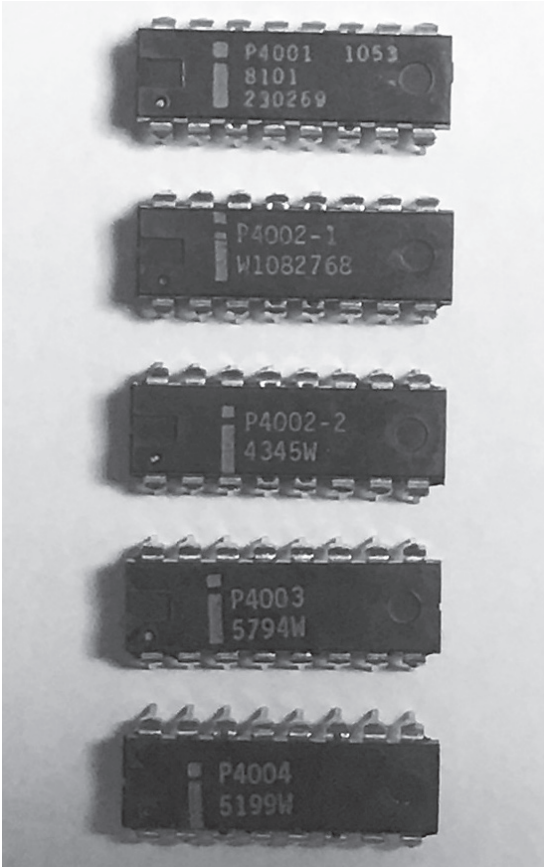


写真1 筆者所蔵のインテル社MCS-4チップ・セット
上から、4001(マスクROMおよび入出力ポート)、4002-1/4002-2(RAMおよび出力ポート)、4003(出力ポート拡張用シフトレジスタ)、4004(4ビットCPU)。4002には4002-1と4002-2の2種類があるが、RAMのバンク内チップ番号の割り当て方法によって使い分けるために用意されている

見本

第24章 歴史的4004アーキテクチャをFPGAの中に実現し、 ビンテージ電卓を再現する

MCS-4システムの論理設計と 電卓の製作

本書付属DVD-ROM関連データ		
DVD-ROM格納場所	内 容	備 考
CQ-MAX10¥Projects¥ PROJ_MCS4	- ディレクトリFPGA：4004(CPU)を中心としたMCS-4システム全体のRTLコードと、そのMAX 10用FPGAプロジェクトとNios IIのファームウェア(Quartus Prime用、Nios II EDS用) - ディレクトリverification：MCS-4システムの論理検証環境(ModelSim用)	MAX 10側
CQ-MAX10¥RaspberryPi¥ CQ-MAX10.tar.gz	- Raspberry Piのホーム・ディレクトリ下に作業用ディレクトリ/home/pi/tempを作成し、本ファイルをその下に置いて解凍(tar xvfz CQ-MAX10.tar.gz)する。その後、解凍してできたディレクトリのうち/home/pi/temp/CQ-MAX10/MCS4を、/home/pi/CQ-MAX10/の下にコピー(第22章の操作と同じ) /home/pi/CQ-MAX10/MCS4/MCS4_Panel_320×240が、画面サイズ320×240のMCS-4システム用デバッグと電卓GUIインターフェース /home/pi/CQ-MAX10/MCS4/MCS4_Panel_800×480が、画面サイズ800×480のMCS-4システム用デバッグと電卓GUIインターフェース /home/pi/CQ-MAX10/MCS4/toolsが、インターネット上にある電卓用バイナリ・コードをMCS-4システム用デバッグに組み込むための変換ツール	Raspberry Pi側

●はじめに

前章では、4004(CPU)を中心としたMCS-4システムのアーキテクチャを詳細に解説しました。本章ではいよいよ、そのアーキテクチャをVerilog HDLで設計し、MAX 10 FPGAの中に実装してみたいと思います。

ここでは、4004(CPU)の他に、周辺チップとしての4001(ROM)、4002(RAM)、4003(シフトレジスタ)と同等な機能も設計し、システムに組み込みます。さらに、CPU(4004)には独自のオン・チップ・デバッグを搭載して、Raspberry Piからプログラムをダウンロードしたりデバッグしたりできるようにします。

MCS-4システムのアプリケーションとしては、博物館に展示されている往年の歴史的電卓(ビジコン社141-PF)も再現し、今でも十分実用的に使える機能とその仕組みをじっくりと味わいたいと思います。

MCS-4 システム全体構成

●Raspberry Pi をユーザ・インターフェースとして使用したシステム

今回製作するMCS-4システムの全体構成を図1に、全体仕様を表1に示します。MAX 10 FPGAを搭載

したMAX10-FB基板(およびFPGA コンフィグレーション用MAX10-JB基板)を、拡張基板MAX10-EB基板に載せ、それをさらにRaspberry Piに載せたシステムを使用します。

●Raspberry Piはユーザ・インターフェース用

図1の一番左側にはRaspberry Piがあります。これはユーザ・インターフェースを受け持ちます。MAX 10 FPGA側とSPI通信により連携します(第22章で解説した手法)。Raspberry Pi上のアプリケーションは、FPGAの中に実装する4004(CPU)のオン・チップ・デバッグと、電卓の入出力部(キーボードとプリンタ)をエミュレーションするハードウェアの両方を制御します。Qt Creatorで開発しました。

●Raspberry PiとMAX 10 FPGAの接続

Raspberry PiとMAX 10を接続するために、MAX10-EB基板を使用します。

Raspberry Piのモニタとして、HDMI接続モニタを使用するか、または7インチ公式タッチ・ディスプレイを使用する場合は、編成A-1または編成A-2(第21章)で互いに接続してください。

4D Systems社のタッチLCDパネル基板4DPiシリーズなど、Raspberry Piの上に重ねるLCD基板

見本

第25章 各種周辺機能をもつ小型MARY基板をMAX10から制御する

MARY基板とMAX10の連携方法

本書付属DVD-ROM関連データ	
DVD-ROM格納場所	内 容
CQ-MAX10¥Projects¥PROJ_MARY	Nios II システムにSDRAMを接続して、MARY 基板(MARY-OB、MARY-GB)を制御するプロジェクト一式(Quartus Prime 用、Nios II EDS 用)
CQ-MAX10¥MARY	MARY 基板の回路図、部品表、基板パターン

●はじめに

MARY 基板とは、Cortex-M0 をコアにもつ NXP 社のマイコン LPC1114 を搭載した約 3 cm 角の超小型基板 MARY-MB(MCU Board)と、それに搭載できる各種周辺基板群から構成されるシリーズで、参考文献(1)で詳しく紹介されています。

MARY 基板は 2011 年に登場した年季ものですが、その周辺機能基板が汎用的に使えるので、今でも流通しています。MARY 基板の回路図、部品表、基板パターンについては、本書の付属 DVD-ROM にも収録したので、参考にしてください。

MAX10-FB 基板を Raspberry Pi に接続するための MAX10-EB 基板を第 21 章で紹介しましたが、この基板には MARY 基板も搭載可能です。MAX10-EB 基板に搭載できる MARY 基板を、表 1 に示します。

本章では、MAX10-EB 基板に、MARY-OB(カラー OLED 表示モジュール + 3 軸加速度センサ)と、MARY-GB(GPS モジュール + リアルタイム・クロック)を搭載する例を紹介します。

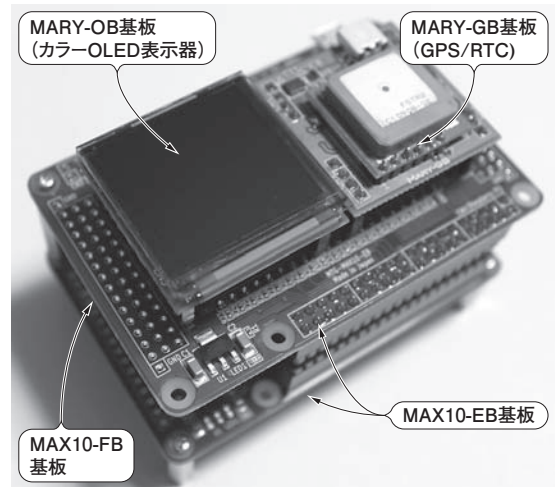


写真1 MARY-OBとMARY-GBをMAX10-EB基板に搭載
MAX 10 から MARY 基板を直接制御する。この写真は編成 D-2 の場合

表1 MARY 基板の種類
MAX10-EB 基板に搭載できるもの

MARY 基板の種類	搭載機能			接続 インターフェース
	種 類	デバイス	メーカ	
MARY-OB	カラー OLED 表示モジュール	UG-2828GDEDF-11	UNIVISION	SPI + GPIO
	3 軸加速度センサ	LIS33DE	STMicro	I ² C + GPIO
MARY-LB	2 色 LED アレイ	A3880EG	Linkman	SPI + GPIO
MARY-XB	XBee 無線モジュール	XB24-ACI-001	Digi	UART
	micro SD カード	—	—	SPI + GPIO
MARY-GB	GPS モジュール	UP501	Fastrax	UART + GPIO
	RTC (リアルタイム・クロック)	RX-8564LC	SEIKO Epson	I ² C + GPIO

見本

ここがスゴイ!

● 開発から学習まで! 手軽に使える大容量FLASH メモリ内蔵ワンチップFPGA誕生

多くのFPGAは、電源投入時、内部のSRAMに外付けのEEPROMから回路データを書き込む「コンフィグレーション処理」が必要です。大容量FLASHメモリを内蔵したMAX 10[®]は、これらの準備作業がいっさい不要なばかりか、ソフトCPUコアNios II[®]も実装できます。A-D変換器も内蔵しており、マイコンの自作も可能です。

同梱のMAX 10 (10M08SAE144C8GES) のスペック

項目	内容
ベンダ	アルテラ [®]
製品シリーズ	MAX 10 FPGA
製品型名	10M08SAE144C8G
プロセス	TSMC 55nm Embedded FLASH プロセス技術
ロジック・エレメント	8K
M9Kメモリ	378Kビット
FLASHメモリ	2496Kビット(コンフィグレーション 用+ユーザ用合計) 1376Kビット(ユーザ用最大値)
18×18乗算器	24個
PLL	2個
LVDS	専用Rx/Txチャネルまたはエミュレーション出力チャネル×41ペア
コンフィグレーション数	最大2コンフィグレーション・イメージを記憶可能
A-D変換器	12ビット×1ユニット、変換レート 1MSPS _{max} 、アナログ入力端子×9本
クロック発振器	リング・オシレータ内蔵 (55M ~ 116MHz)
I/O本数	101本
電源	3.3V単一電源
パッケージ	EQFP-144(20mm×20mm、ピン・ ピッチ0.5mm)

ISBN978-4-7898-4806-0

C3855 ¥15600E

CQ出版社

定価：本体15,600円(税別)



● ホビー用Linux I/Oコンピュータ 「ラズベリー・パイ」でIoTの実験

イーサネットやWi-Fi, USB, Bluetooth, カメラ, LCDなど、多彩なコネクティビティをもつコンピュータ「ラズベリー・パイ」とMAX 10[®]を組み合わせると、強力な計算能力をもつI/Oコンピュータに生まれ変わります。



● 温故知新! ビンテージ・マイコン インテル[®]4004の実験

世界初の4ビット・マイクロコンピュータ MCS-4の4004とそのチップ・セットをHDLで設計してMAX 10[®]に実装します。



姉妹書のご案内

① MAX 10[®] ライタ DVD付き! FPGA電子工作スーパーキット



B5判, 272頁, 4,500円+税

本書の第1部と基板2枚、開発用DVDを収めた体験版

MAX 10[®]実験キットで学ぶ FPGA&コンピュータ



B5判, 600頁, 6,000円+税

本書同梱の解説書そのもの。「キットを買う前によく調べたい」「MAX 10のことがわかればOK!」という方向け

「はんだ付けはパス!」「もう1枚!」そんな方へ

おすすめキット①



MTG-MAX10-FB-F
(マルツエレクトロニクス)
5,980円+税

本書同梱のMAX 10[®]基板のフル実装版。512bit SDRAMなど搭載済み。参考書は「MAX 10[®]実験キットで学ぶFPGA&コンピュータ」

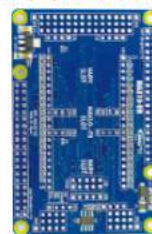
おすすめキット②



MTG-MAX10-JB-F
(マルツエレクトロニクス)
3,480円+税

本書同梱のライタ製作用プリント基板のフル実装版。PICマイコンなど搭載済み。参考書は「MAX 10[®]実験キットで学ぶFPGA&コンピュータ」

おすすめキット③



MTG-MAX10-EB
(マルツエレクトロニクス)
3,980円+税

「MAX 10[®]実験キットで学ぶFPGA&コンピュータ」のラズベリー・パイ連携製作や3cm角のジャンプボードに使うインテリジェントボード

見本