MAX IICPLD による SPI-I²C 変換

この資料は英語版を翻訳したもので、内容に相違が生じる場合には原文を優先します。こちらの日本語版は参考用としてご利用ください。設計の際には、最新の英語版で内容をご確認ください。

2007 年 12 月 ver 1.0 Application Note 486

はじめに

このデザインは、一般的に使用されている共有バス・アーキテクチャであるシリアル・ペリフェラル・インタフェース（SPI）と I²C バス間のプロトコル・コンバージェンスを提供します。アルテラの MAX® II CPLD は、SPI インタフェースを持つホストと、I²C バスを介して接続されたデバイスとの通信を可能にするブリッジとして機能します。

I²C と SPI

I²C は、2 線式、狭帯域幅の業界標準シリアル・プロトコルで、エンベデッド・システムにおいて各種の低速ペリフェラル・デバイスとの通信に使用されます。他方、SPI は、幅広く使用されている高速、4 線式の全二重シリアル通信インタフェースです。今日の多くのエンベデッド・システムが SPI インタフェースを備えているため、I²C 方式でペリフェラル・デバイスに接続するのが困難になっています。この接続はシステムを修正する方法で実施できますが、経済的に非効率です。この接続を行うのに最適な方法は、2 つのインタフェース間のブリッジとして機能する CPLD を使用することです。

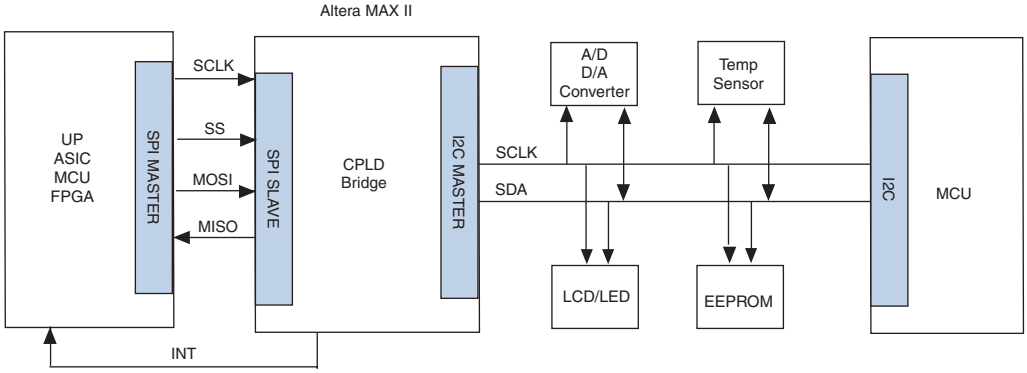
このデザインでは、MAX II CPLD を使用してこのようなブリッジを実装しています。MAX II CPLD を使用する理由は、高い柔軟性、低消費電力、およびエンベデッド・システムへの統合を経済的に行える点です。CPLD は、ホスト（SPI マスタ）に対してはSPIスレーブとなり、I²Cバスに対してはマスタとして機能します。

MAX IICPLD による SPI-I²C 変換

このデザインでは、SPI インタフェースを備えたホストは、A/D コンバータ、LED コントローラ、オーディオ・プロセッサなどの他のデバイスへのデータ・フローを制御して、I²C インタフェース上の温度センサ、ハードウェア・モニタ、および診断センサを読み取ることができます。

図 1 に、MAX II CPLD を使用した I²C インタフェースへの SPI の実装を示します。このブリッジは、4 線のうち SS 信号と SCLK 信号を制御用に、MISO 信号と MOSI 信号をデータ用に使用することで、SPI ホストに SPI スレーブとしてインタフェースします。I²C バスとのインタフェース側は 2 線式で、SCLK 信号および SDA 信号があります。

図 1. MAX II CPLD を使用した I²C インタフェースへの SPI の実装



SPI インタフェース

通常、SPI バスには 1 つのマスタとそれに接続される多数のスレーブがあります。CPLD ブリッジは、SPI マスタ・デバイスへのスレーブの 1 つとして機能します。図 2 に、SPI のタイミング図を示します。表 1 に、SPI インタフェース・ピンの説明を示します。

図 2. SPI のタイミング図

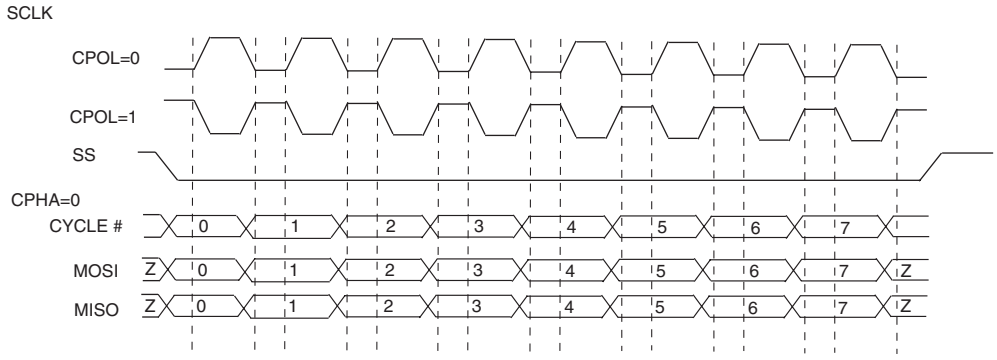


表 1. SPI インタフェース・ピンの説明		
信号	用途	入力 / 出力
SS	スレーブ選択	入力（アクティブ Low）
SCLK	SPI クロック	入力
MISO	マスタ入力、スレーブ出力	出力
MOSI	マスタ出力、スレーブ入力	出力

SPI ホストは以下を送信します。

- コマンド・レジスタ（8 ビット）
- データ・イン（8 ビット）

SPI ホストは以下を受信します。

- ステータス・レジスタ（8 ビット）
- データ・アウト（8 ビット）

SPI ワード長は 16 ビットに固定されています。図 2 に示すように、CPOL = 0、CPHA = 1 です。SPI ワードごとに、コマンド・レジスタは I²C バス上にファンクションを指令し、データ・インは I²C バスが送信するデータを保持します。同様に、ステータス・レジスタの最終ビットはアクノリッジ・ビットで、データ・アウトは前の I²C サイクルの I²C ライン上で受信されたデータです。

各 SPI バスの最後では、スレーブ選択ラインが High になってワードの完了を示し、その時点でのコマンド・レジスタの値に従って、I²C バス・サイクルが実行されます。I²C SCL の周波数に応じた一定の遅延後、別の SPI ワードを送信できます。2 つの SPI ワード間の最小遅延は、I²C SCL クロック周波数です。

I²C インタフェース

CPLD ブリッジは、I²C バスへのマスタとして機能します。このデザインは、SPI マスタと I²C スレーブ・デバイス間のインタフェースを提供することを目的としているため、I²C バスにマルチ・マスタ・サポートは提供されません。表 2 に、I²C インタフェース・ピンの説明を示します。

表 2. I²C インタフェース・ピンの説明

信号	用途	入力 / 出力
SCLK	I²C シリアル・クロック	出力
SDA	I²C データ・バス	双方向

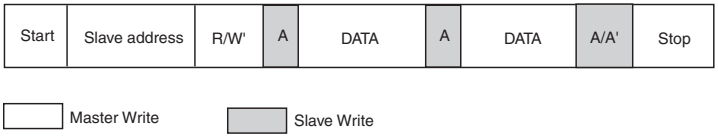
I²C ファンクションは、SPI サイドから受信したコマンド・レジスタの値に基づいて実行されます。表 3 に、コマンド・レジスタに格納された値の意味を示します。

表 3. I²C コマンド

コマンド・レジスタ	I²C ラインでの用途	データ・イン・レジスタ
10000000	開始 / 開始の反復	スレーブ・アドレス + R/W
01000000	バイトの書き込み	書き込まれるデータ
00100000	バイトの読み出し	Don't care
00010000	停止	Don't care
00000000	Null、ウェイト・ステート	Don't care

特定の I²C トランザクションで読み出されたデータは、データ・アウト・レジスタに格納され、次の SPI トランザクションで SPI マスタによって読み出されます。SPI マスタが I²C バス上で何も処理しないで、ステータス・レジスタとデータ・アウト・レジスタの値を読み出すには、最後のコマンド・ワードの 00000000（b）が必要です。図 3 に、I²C コマンド・フォーマットを示します。

図 3. I²C コマンド・フォーマット



実装

本デザインは、EPM240 またはその他の MAX II CPLD を使用して実装できます。実装では、このデザインのソース・コードを使用し、適切な信号ラインとコントロール・ラインを MAX II CPLD の汎用 I/O（GPIO）ラインに割り当てています。SPIマスタとI²Cスレーブは、この実装をデモするために必要な追加リソースです。

以降のステップでは、MDN-B2 ボード、Freescale の SPIGen ソフトウェア（およびハードウェアをインタフェースする適切なパラレル・ポート）を使用して作成された PC パラレル・ポート・ベースの SPI 環境、および I²C スレーブ・バッテリ・ゲージ・モジュール（MDN-B2 に付属）を使用して、詳細なデモを行います。

表 4 に、EPM240G アサインメントを示します。

表 4. EPM240G のピン・アサインメント

ピン・アサインメント	
信号	**ピン**
I2C_scl:	ピン 39
I2C_sda	ピン 40
SPI_cs	ピン 95
SPI_miso	ピン 91
SPI_mosi	ピン 92
SPI_clk	ピン 96

未使用ピンは、Quartus® II ソフトウェアのデバイスおよびピン・オプション設定で **As input tri-stated** に割り当ててあります。SPI_sclk には、ピン・アサインメント時に、Quartus II ソフトウェアの I/O 規格カラムで 2.5 V シュミット・トリガ入力が割り当てられます。Quartus II ソフトウェアの Assignment Editor は、I2C_scl および I2C_sda ピンで **Auto Open Drain** をイネーブルするのに使用されます。適切な設定をオンにした後、デザインをコンパイルします。

デザイン・ノート

MDN-B2デモ・ボードで本デザインのデモを行うには、以下のステップを実行します。

1. PC とそのパラレル・ポートを使用して SPI 環境をセットアップするには、Freescale から SPIGen などの SPI エミュレーション・ソフトウェアをダウンロードします。このソフトウェアは、Freescale の利用規約と無償登録を同意した後、無償でダウンロードできます。

以下のウェブサイトにアクセスして、SPI エミュレーション・ソフトウェアをダウンロードしてください。

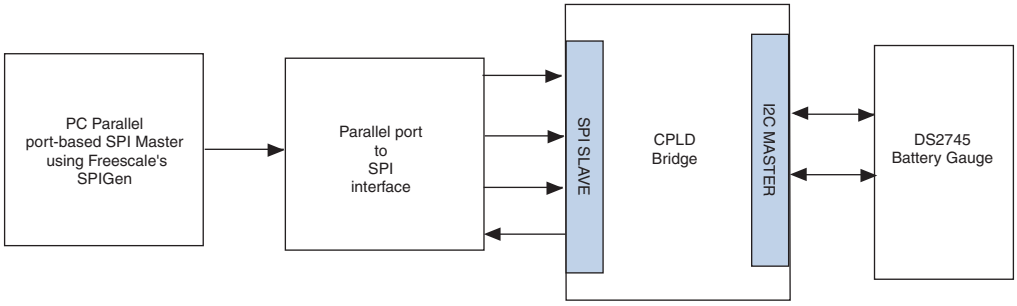
www.freescale.com/files/soft_dev_tools/software/device_drivers/SPIGen.html

2. 登録後、SPIGen をダウンロードしてインストールします。アプリケーションに合わせて SPIGen をコンフィギュレーションし、MDN-B2 デモ・ボードに付属している Parallelport-to-SPI ドングルを装着して作業する必要があります。

3. SPIGen ソフトウェアを起動し、**Configure** メニューをクリックします。**Edit Configuration** を選択します。**General** タブで、デフォルトの **Port Address** が、使用する PC のパラレル・ポートの正しいポート・アドレスであることを確認します。正しくない場合は、正しいオプションを選択します。PC のパラレル・ポート・アドレスは、コントロール パネル > システム > ハードウェア > デバイス マネージャ > ポート > ECP プリンタ ポート (LPT1) > リソースを選択して設定を調べるとわかります。

   図 4 に、デモの構成を示します。

図 4. デモの構成



4. 16 ビット・フォーマットを選択し、SPI 値を 16 進（Intel フォーマット）ファイル（**.hex**）形式で表示するように選択します。

5. 以下の操作を実行して、**SPI Pins** タブをコンフィギュレーションします。

   a. ピン・アサインメントをデフォルト設定のままにします（Chip Select: 2、Data In: 3、Data Out: 12、および Clock: 4）。

   b. **Bit Order** をデフォルトの **MSB is sent first** 設定のままにします。

   c. **Chip Select** を **High when asserted** に変更します。

   d. **Data In** と **Data out** を **Low = 1** に変更します。

   e. **SPI Type** を **Type 4** に変更します。

   f. **Apply** と **OK** をクリックして、メイン SPIGen 画面に戻ります。

6. Parallelport-to-SPI ドングルを、PC のパラレル・ポートに接続します。パラレル・ポート延長ケーブルを使用すると、ドングルとデモ・ボードを容易に接続できます。これで、パラレル・ポート・ベースの SPI 環境を使用できるようになりました。

7. デモ・ボードの電源をオンにします（スライド・スイッチ SW1 を使用）。デモ・ボード上の JTAG ヘッダ JP5 とプログラミング・ケーブル（ByteBlaster™ II または USB-Blaster™）を使用してデザインを MAX II CPLD へダウンロードします。

8. プログラミング・プロセスの起動前と起動中、デモ・ボードの SW4 を押し続けます。完了したら、電源をオフにして JTAG コネクタを取り外します。

9. DS2745 I²C バッテリ・ゲージ・モジュール（MDN-B2 デモ・ボードに付属）をデモ・ボードの JP3 に取り付けます。モジュールの赤マークが JP3 のピン #1 と一致していることを確認してください。

10. Parallelport-to-SPI ドングルをデモ・ボードに接続します。

11. ドングルのピグテール線の 6 ピン・コネクタをデモ・ボードの JP8 に接続します。この 6 ピン・コネクタの赤い線が JP8 のピン #2 と一致していることを確認します。

JP8 は 10 × 2 ピン・ヘッダであることに注意してください。この 6 ピン・コネクタは JP8 の部分のみ占有します。

12. スライド・スイッチ SW1 を使用して、デモ・ボードの電源をオンにします。

13. 表 5 に、SPIGen ソフトウェアの **Word to Send (DI)** フィールドを使用して送信する **.hex** データの順序を示します。

表 5.　SPIGen を使用した 16 進での SPI “Data to Send”	
信号の方向	コマンド / データ
送信（DI）	80 90（開始 + スレーブ・アドレス）
送信（DI）	40 0C（書き込み + メモリ・アドレス）
受信（DO）	01 00 (ack)
送信（DI）	80 91（開始 + スレーブ・アドレス）
受信（DO）	01 00 (ack)
送信（DI）	20 00（リード）
受信（DO）	01 ??（ack + MSB データ）
送信（DI）	00 00（何も実行しない）
受信（DO）	01 ??（ack + LSB データ）

14. 表 5 の **.hex** データを使用し、SPI データを送信するたびに **Send Once** をクリックします。アクノリッジ・データが受信されることを確認します。

15. 受信したバッテリ・ゲージ・データを確認します。このデータは 2 つの部分に分けて受信され、最初に MSB 8 ビット・データ、次に LSB 3 ビット・データ（最後の行）が供給されます。

16. バッテリ・ゲージ・モジュールの黄色のプリセットを変更し、変化したデータが受信されることを確認します。この黄色のプリセットにより、I²C インタフェースを介して11ビットの2の補数形式で電圧の読取値を提供するバッテリ・ゲージ・チップ（Maxim DS2745）への入力電圧が変化します。

詳細は以下を参照してください。

www.maxim-ic.com/quick_view2.cfm/qv_pk/4994

ソース・コード

このデザインは Verilog を使用して作成しており、MDN-B2 デモ・ボードを使用したデモとなっています。ソース・コード、テストベンチ、および完成した Quartus II プロジェクトは、以下から入手可能です。

www.altera.co.jp/literature/an/an486_design_example.zip

まとめ

このデザインで示すとおり、MAX II CPLD は、SPI や I²C などの業界標準インタフェースを実装するための有力な選択肢となります。MAX II CPLD は、低消費電力、使いやすいパワー・シーケンス、および内部オシレータを特長としており、SPI-I²C などのインタフェース・コンバータ・アプリケーションを実装するための最適なプログラマブル・ロジック・デバイスです。

参考資料

このアプリケーション・ノートでは、以下のドキュメントを参照しています。

- http://www.altera.com/literature/an/an486_design_example.zip
- http://www.freescale.com/files/soft_dev_tools/software/device_drivers/SPIGen.html
- http://www.maxim-ic.com/quick_view2.cfm/qv_pk/4994

関連情報

以下に、このアプリケーション・ノートの関連情報を示します。

- MAX II CPLD ホームページ :
  http://www.altera.com/products/devices/cpld/max2/mx2-index.jsp
- MAX II デバイスの資料ページ :
  http://www.altera.com/literature/lit-max2.jsp
- MAX II パワーダウン・デザイン :
  http://www.altera.com/support/examples/max/exm-power-down.html
- MAX II アプリケーション・ノート :
  「AN 428: MAX II CPLDのデザイン・ガイドライン」
  「AN 422: MAX II CPLD を使用したポータブル・システムにおける消費電力の管理」

改訂履歴

表 6 に、このアプリケーション・ノートの改訂履歴を示します。

表 6. 改訂履歴

日付 & ドキュメント・バージョン	変更内容	概要
2007 年 12 月 v1.0	初版	—

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Literature Services:
literature@altera.com