

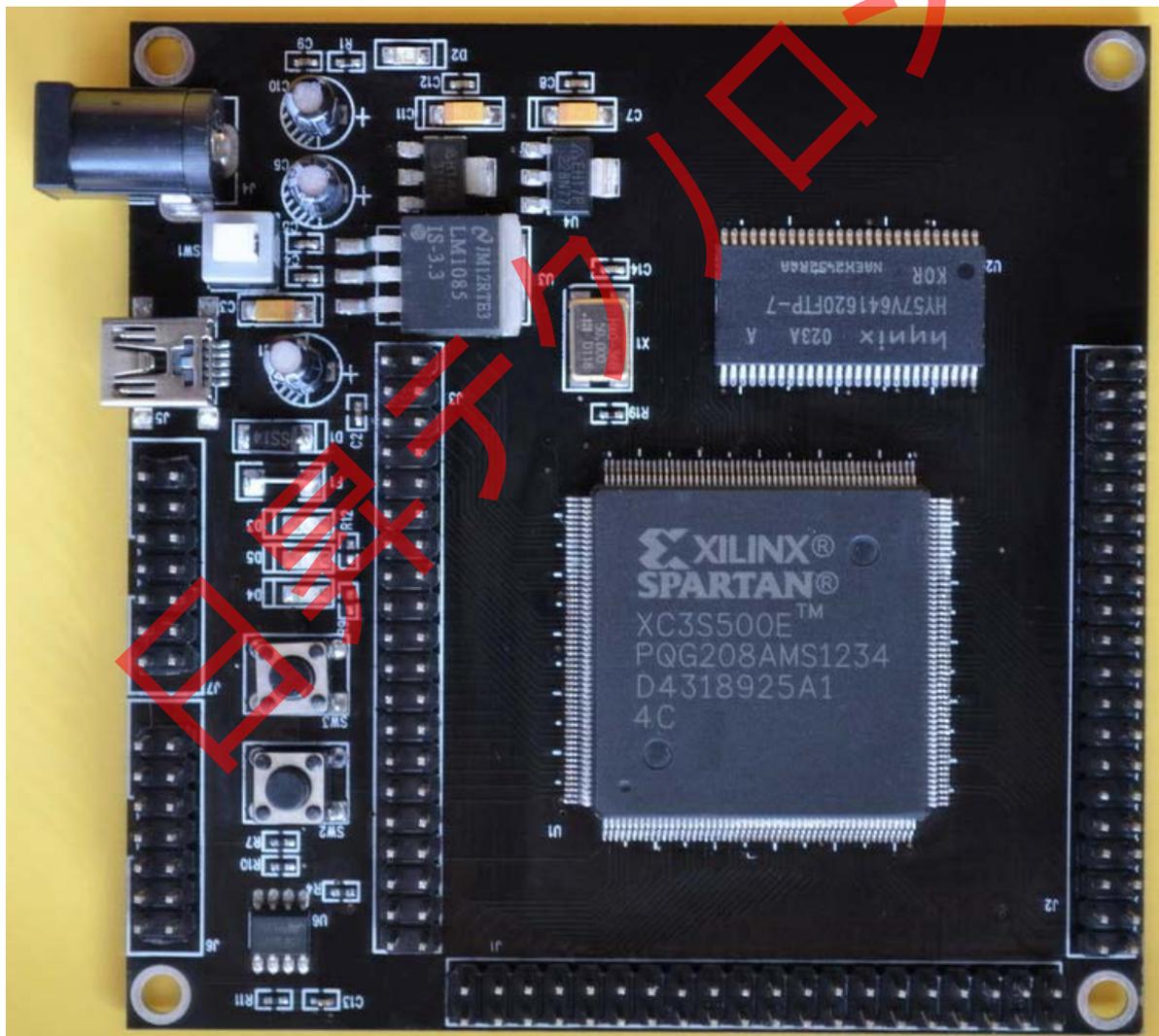
XILINX FPGA XC3S500E 開発ボードのマニュアル

株式会社日昇テクノロジー

<http://www.csun.co.jp>

info@csun.co.jp

作成日 2014/8/11



copyright@2014

・ 修正履歴

NO	バージョン	修正内容	修正日
1	Ver1.0	新規作成	2014/8/11

※ この文書の情報は、文書を改善するため、事前の通知なく変更されることがあります。
最新版は弊社ホームページからご参照ください。「<http://www.csun.co.jp>」

※ (株)日昇テクノロジーの書面による許可のない複製は、いかなる形態においても厳重に禁じられています。

日昇テクノロジー

目次

1、注意事項	4
2、開発ボードの主な部品及びハードウェアリソース	5
2.1 主な部品	5
2.2 ハードウェアリソース	5
2.3 PCB とサイズ	7
3、開発ボードの実装と使用	8
3.1 開発ボードの実装	8
3.2 開発ボードの使用	9
3.2.1 ロジック開発プロセスの紹介	9
3.2.2 ダウンロード	13
3.2.3 SPI Flash ダウンロード	15
4、開発ボード回路説明	25
4.1 電源回路	25
4.2 クロック回路	25
4.3 リセット回路	25
4.4 JTAG インタフェース回路	26
4.5 SPI FLASH インタフェース回路	26
4.6 LED 回路	27
4.7 メモリーSDRAM 回路	28
4.8 外部拡張 IO	29

1、注意事項

使用要求

- ケーブルを抜き差しする前に、開発ボードを使用中止し、電源を切る。
- 不明なもの又は液体が開発ボードと接触した場合に、すぐに開発ボードを使用中止し、電源を切って開発ボードに接続するケーブルを全部抜き出す。
- 異常状況が出る場合に、例えば、設備から煙が出る、異臭を放つ時に、すぐに開発ボードを使用中止し、電源を切る。
- 長時間に使用しない場合に、電源を切る。

静電気安全及び保護

静電気はしばしば開発ボードのチップを損傷する発生誘因となっている。そのため、開発ボードに触る前に十分な保護対策を行わなければならない。

ケーブルの挿抜安全

- ケーブルを挿抜する場合に、フラットケーブルの方向を間違い無い様に注意してください。間違った場合、ロジックチップ又はダウンロードケーブルを損傷する恐れがある。
- 電源を入れたまま JTAG ケーブルを挿抜操作禁止。通電状態操作すればロジックチップに内蔵された設定回路に致命的な損傷をもたなす。(コンピュータマザーボードのボードを挿抜と同じ)

外部電源の使用

開発ボードが DC 5V 電源入力をサポートする。DC 電源インタフェースを使う場合に、電圧が間違い又は極性が逆接するで開発ボードを損傷しないように、電源アダプタ出力が 5V、かつ内側が正極、外側が負極である事を確認する。

開発ボードは UBS 電源給電もサポートする。標準携帯電話の充電器及び Mini USB 線により開発ボードに給電する。

開発ボードを給電する場合に、DC インタフェースでも Mini USB インタフェースでも使える。(自由に一つを選ぶ)

開発ボードとインタフェースボードの接続

3列ソケットにより開発ボードとインタフェースボードを接続する。開発ボードにはピンヘッダ、インタフェースボードにはソケットがある。単独に本ボードを使わない場合は、出来るだけ抜き出さないでください。

本ボードを取り出す時に、片手でインタフェースボードを固定しながら本ボードを握り適当な力で取り出す。(ボードを握る力を適当に調整する)

本ボードを取り付ける時、ボードの方向を十分に注意してください。ピンヘッダとソケットが1対1対応するかを確認し、取り付ける。

2、開発ボードの主な部品及びハードウェアリソース

2.1 主な部品

開発ボードのコアチップはXILINX Spartan®-3E シリーズのFPGA XC3S500EPQ208を使用する。Spartan®-3E FPGAはロジック最適化、低コスト、完全な機能を持っているプラットフォームの一つである。100Kから1.6Mまでのシステムゲート、66から376までのI/Oインタフェースがある。詳細は以下のよう:

Table 1: Summary of Spartan-3E FPGA Attributes

Device	System Gates	Equivalent Logic Cells	CLB Array (One CLB = Four Slices)				Distributed RAM bits ⁽¹⁾	Block RAM bits ⁽¹⁾	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs	Total Slices						
XC3S100E	100K	2,160	22	16	240	960	15K	72K	4	2	108	40
XC3S250E	250K	5,508	34	26	612	2,448	38K	216K	12	4	172	68
XC3S500E	500K	10,476	46	34	1,164	4,656	73K	360K	20	4	232	92
XC3S1200E	1200K	19,512	60	46	2,168	8,672	136K	504K	28	8	304	124
XC3S1600E	1600K	33,192	76	58	3,688	14,752	231K	648K	36	8	376	156

Notes:

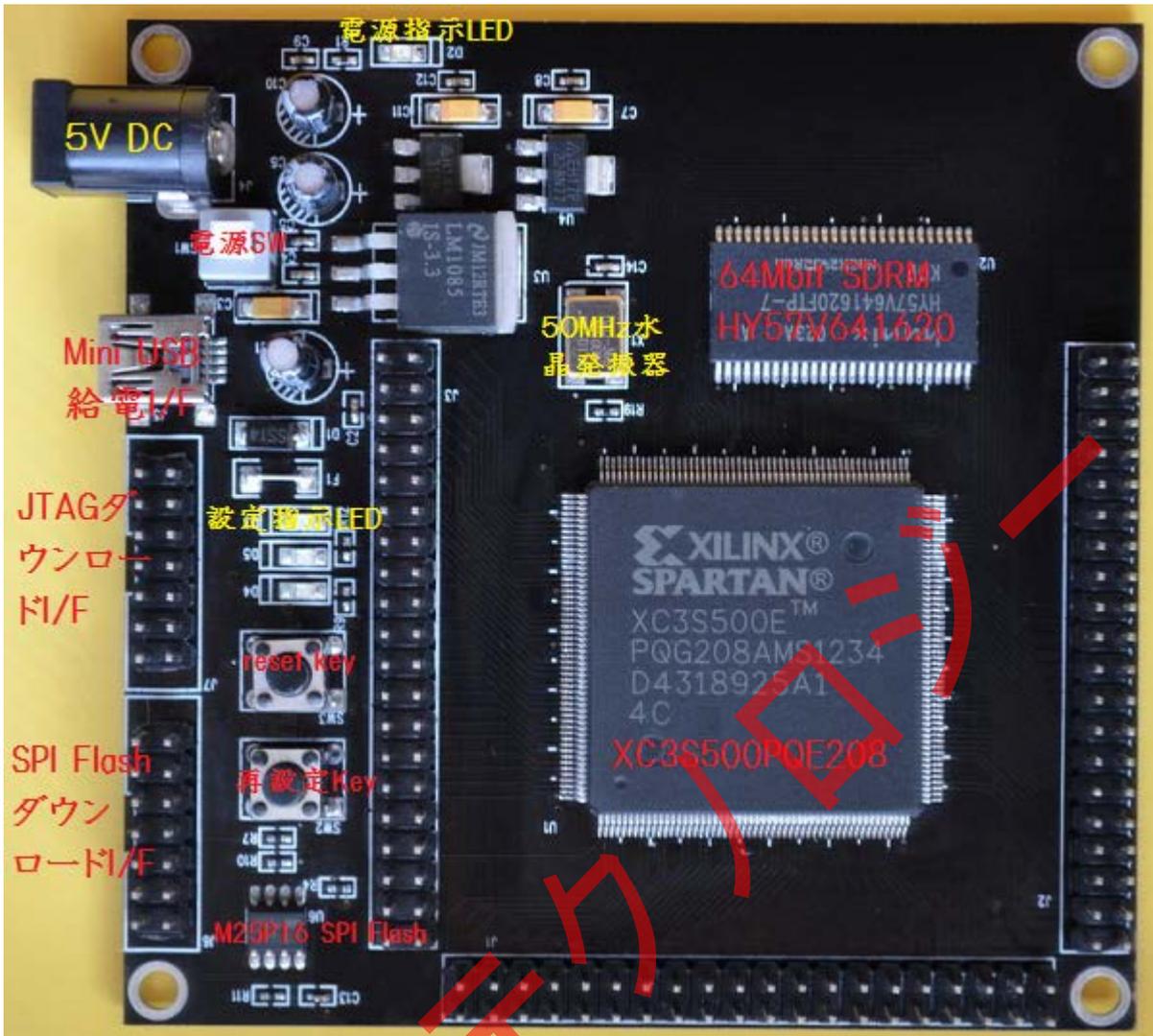
1. By convention, one Kb is equivalent to 1,024 bits.

FPGAにとってはロジックチップの論理ゲート数とRAM bit数はFPGAの重要な参考指標である。それに対して、I/O数はそれほど重要ではない。

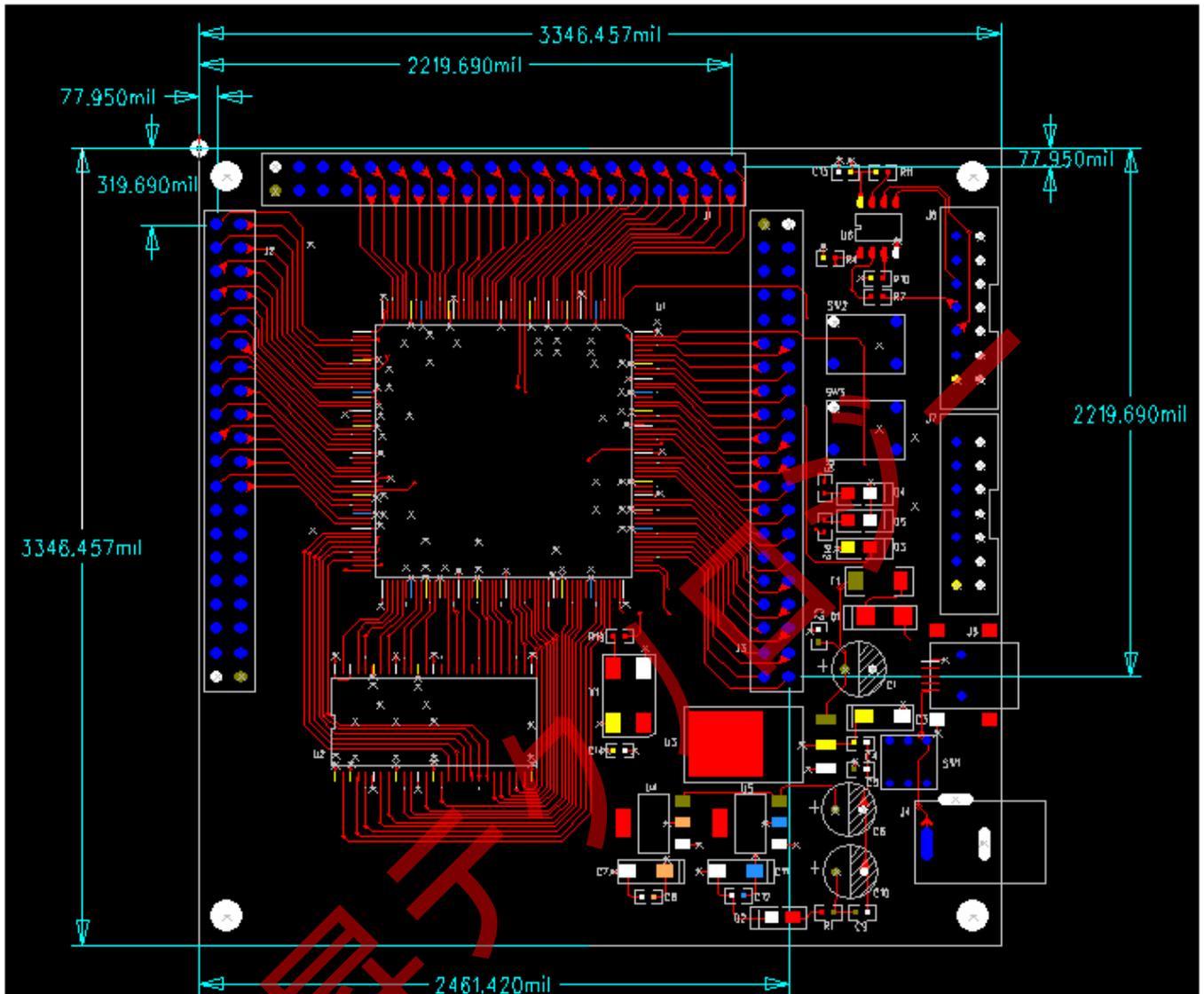
2.2 ハードウェアリソース

- メインチップはXILINX 会社のSpartan®-3E シリーズのFPGA XC3S500EPQ208を採用。
- 50MHz 水晶発振器、システムが動作するメインクロックを提供する。
- M25P16シリアル・コンフィギュレーション・チップを搭載する。FPGAロジックも、SOPCプログラムも保存できる。
- 64Mbit のSDRAM (HY57V64162) を搭載する。SOPC設計もできる。
- 大電力 LDO 電源管理チップ LM1085-3.3V を採用し、3A の 3.3v 電圧出力までサポートする。
- AMS1117-1.2V/2.5Vレギュレータチップを使用し、FPGAコアに電圧を提供する。
- IN5819 高速ショットキーダイオード、電源逆接の保護設計を持っている。
- Self-locking 電源スイッチ一つ
- 5V 直流電源コンセント、電源アダプタでも給電できる。
- MINI USB コネクタ、携帯電話の標準充電器でも給電できる。
- 赤 LED 一つ、電源表示灯とする。
- ブルーLED 二つ、LED 点滅試験とする。
- リセットキー一つ、ユーザーキーとする。
- 再設定キーがあり、FPGAのロジック再設定に利用する。
- JTAGインタフェースをサポートする。
- JTAGでM25P16のプログラミングをサポートする。
- デカップリング設計。数多くのデカップリングコンデンサーを採用する。
- 全ての I/O を 2.54mm ピッチ拡張ピンヘッダで引き出す。

イメージは下図の通り :



2.3 PCB とサイズ



- PCB サイズ : 3346.457*3346.457 (単位 : mil)
 J1.1 位置 : (2219.690, -77.950) (単位 : mil)
 J2.1 位置 : (77.950, -319.690) (単位 : mil)
 J3.1 位置 : (77.950, -319.690) (単位 : mil)
 J4.1 位置 : (2461.420, -2219.690) (単位 : mil)

3、開発ボードの実装と使用

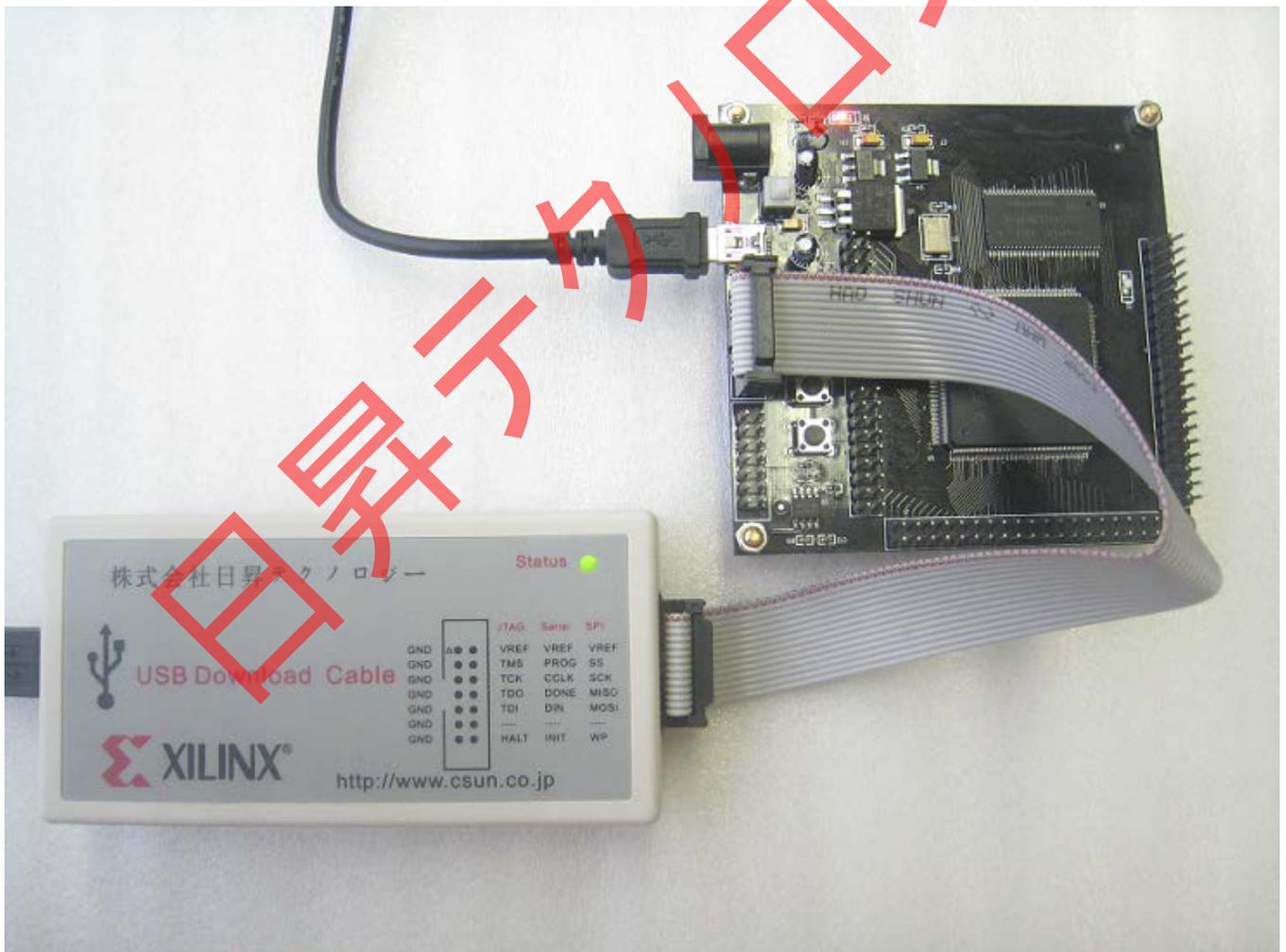
本開発ボードは以下の部品が含んでいる。

- 開発ボード 1個
- Mini USB ケーブル 1本
- スタッドとネジ 4本ずつ

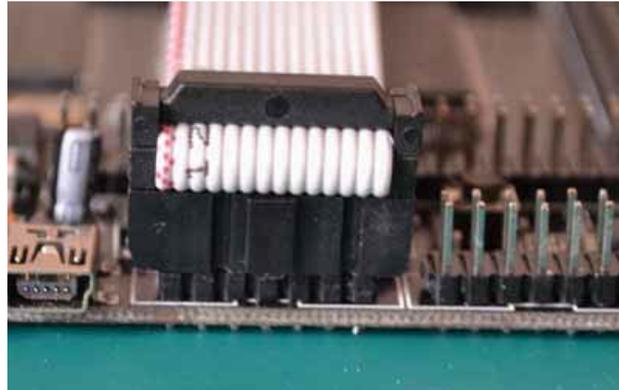
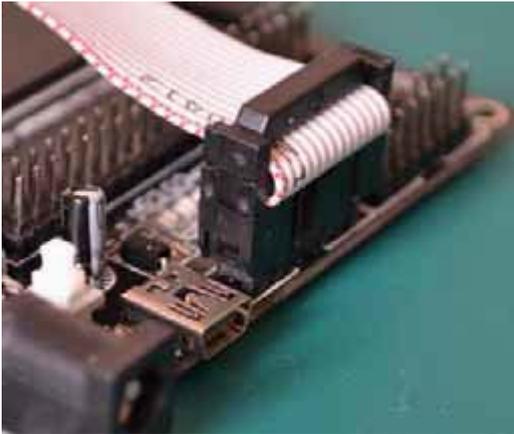
3.1 開発ボードの実装

- スタッドを開発ボードの穴に取り付ける。
- ダウンロードケーブルを開発ボードに接続する。(注意：JTAG フラットケーブルの赤い側が PCB のプライミング1ピンに接続する。)
- Mini USB を開発ボードに接続し、ボードに電源を提供する。
- 実装された開発ボードをデスクの上に平らに置く。
- パソコンに関連ソフトウェアをインストールする。例えば ISE、テキスト編集ソフトウェア UltraEdit、コーディングチェックソフトウェア Nlint など。

開発ボードの接続図：



下図がデータケーブルと開発ボードの接続詳細図：



注意：データケーブルを接続する時に、方向を注意してください。

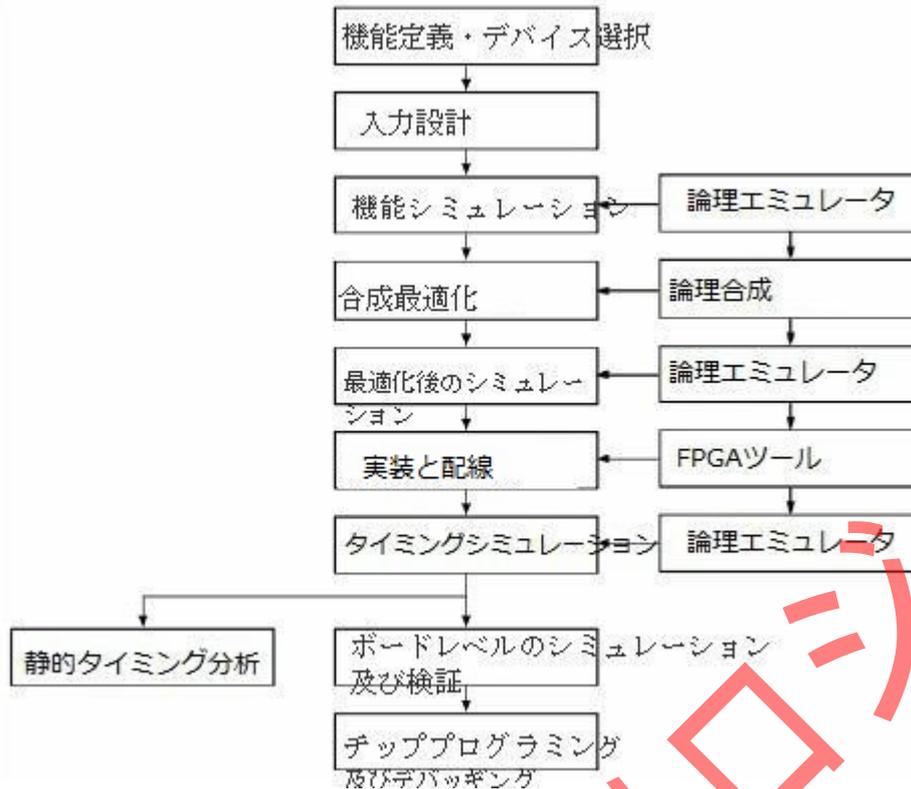
3.2 開発ボードの使用

開発ボードが正確に接続したあと、ボードに電源を入れる。

3.2.1 ロジック開発プロセスの紹介

ロジック開発は、ハードウェア設計とソフトウェア設計二つの部分が含まれている。ハードウェアはロジックチップ回路、JTAG、クロック、リセット、メモリ、入出力インタフェース回路及び他のインタフェースが含まれている。これは本ボードで提供している。ソフトウェアはHDLプロセスである。この部分はお客様の事情によって自分で設計する。ここでご参考までいくつかの例を提供する。

ロジック設計プロセスは EDA 開発ソフトと編集ツールを利用して、ロジックチップに対する開発する。代表的なロジック開発プロセスは機能定義・デバイス選択、入力設計、機能シミュレーション、合成最適化、最適化後のシミュレーション、実現及び配線後のシミュレーション、ボードレベルのシミュレーション、チッププログラミング及びデバッグなどのステップを含んでいる。下図の通り：



機能定義：実現したい機能を定義する。例えば、開発ボード上の二つのLEDの点滅。

入力設計：ハードウェア記述言語（HDL）で実現したい機能を記述する（つまり、テキスト編集ソフトウェア UltraEdit に HDL 言語で記述）。回路図モードを推奨していない（非効率的な方法、維持するのが困難、モジュール構造と再利用も難しい。かつ、回路図のポータビリティが悪い）。例えば、下図は一つのLED点滅の設計記述、つまりLEDが50000000回のclkずつ一回点滅する。

```

parameter TIME_COUNT      = 32'd100000000;

/***** ポート声明 *****/
//input
input          clk;
input          reset_n;

//output
output [1:0]   led_flash;      //LEDは1 s間隔で点灯する。1 : 点灯 ; 0 : 消灯

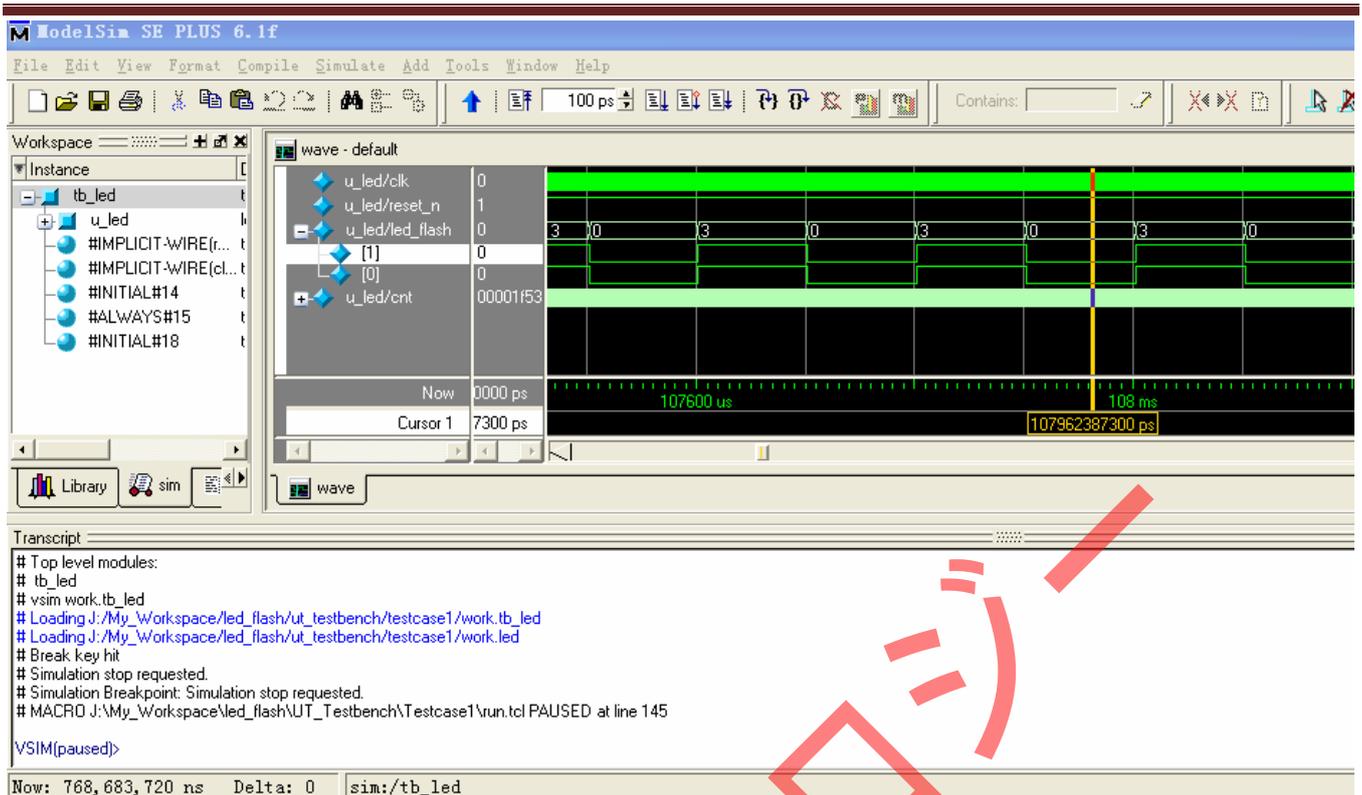
/***** ポート定義 *****/
reg [1:0]      led_flash;

/***** 内部信号定義 *****/
reg [31:0]     cnt;

/***** コード開始 *****/
always @(posedge clk or negedge reset_n)
begin
  if (reset_n == 1'b0)
    cnt <= 32'b0;
  else if (cnt >= TIME_COUNT - 1'b1)
    cnt <- #U_DLY 32'b0;
  else
    cnt <= #U_DLY cnt + 1'b1;
end

always @(posedge clk or negedge reset_n)
begin
  if (reset_n == 1'b0)
    led_flash <= 2'b0;
  else if (cnt == TIME_COUNT - 1'b1)
    led_flash <= #U_DLY 2'b11;
  else if (cnt == (TIME_COUNT >> 1) - 1'b1)
    led_flash <= #U_DLY 2'b0;
  else
    ;
end
  
```

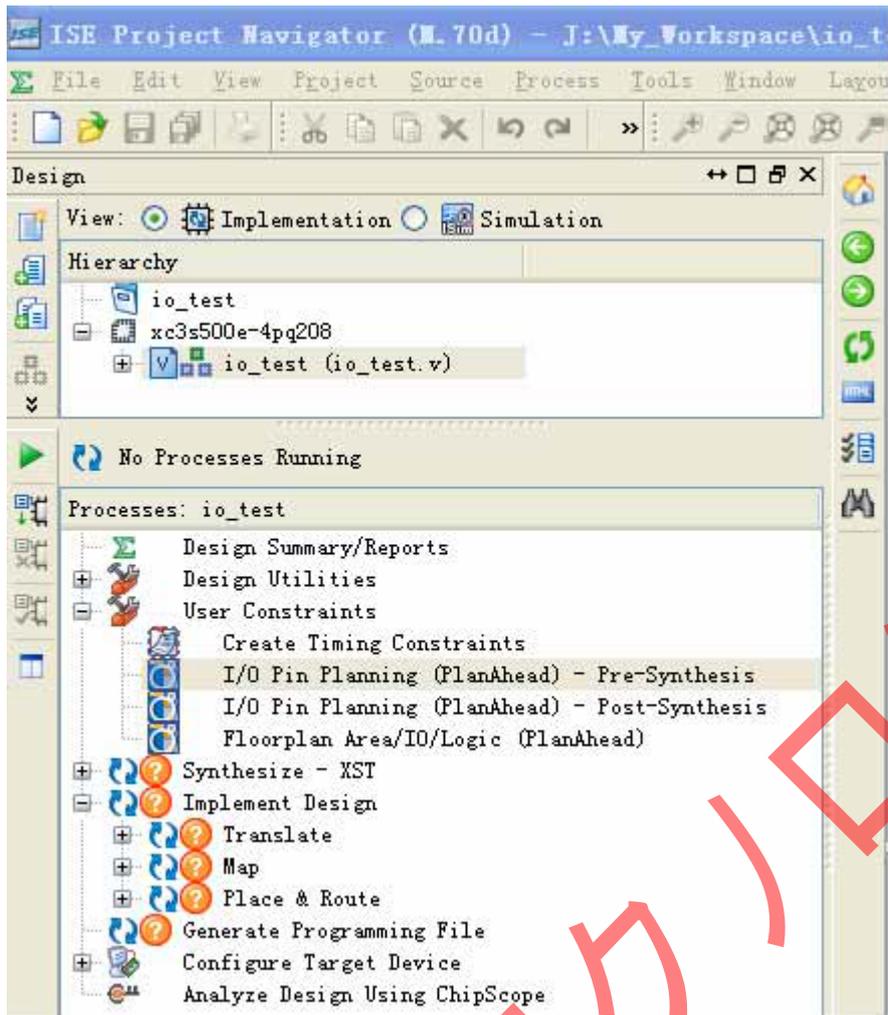
機能シミュレーション：前シミュレーションとも呼ばれる。コンパイルの前にユーザーにより設計された回路のロジック機能を検証する。一般的には Modesim 又は Questa でシミュレーションする。下図が LED 点滅のシミュレーション波形である。Led_flash の値が一定な時間を間隔に 1 になり、LED を点灯する。



合成最適化：設計入力を、実際のゲート回路ではなく、AND ゲート、OR ゲート、NAND ゲート、RAM、トリガーなどの基本ロジックユニットから組み立てられた論理接続ネットリストにコンパイルされる。実際のゲート回路はメーカーのレイアウトソフトウェアを利用して、合成に生成された標準ゲートレベル構造に応じて生成する。XILINX 会社の部品は、設計を ISE に導入し、ISE ソフトウェアで合成最適化する。

合成後のシミュレーション：合成後の結果と当初の設計が同じかどうかを確認する。シミュレーションの時、合成に生成された標準遅延ファイルを合成のシミュレーションモデルにマークし、ゲート遅延の影響を推定できる。

配線レイアウトと実装：つまり、ロジックメーカーのソフトウェアを利用して、ロジックを目標デバイス構造のリソースにマッピングし、ロジックのベストレイアウトを決定する。ロジックと入出力機能の接続配線チャンネルを選択する上に、接続し、対応的なファイル（例えば設定ファイル、関連報告）を生成し、合成に生成されたロジックネットリストを具体的なロジックチップに配置する。XILINX 会社の部品は、設計を ISE に導入し、ISE ソフトウェアで配線レイアウトを行って、設定ファイルを生成する。下図は LED 点滅で配線レイアウトの例である。



タイミングシミュレーション：後シミュレーションとも呼ばれる。配線レイアウトの遅延情報を設計ネットリストにマークし、タイミングの違反（即、タイミング約束条件又はデバイス固有のタイミング規則に満足しない、例えば、セットアップ時間、ホールド時間など）を検出する。タイミングシミュレーションの遅延情報は最も正確、最も完全である。

ボードレベルのシミュレーション：主に高速回路設計に使用される。高速システムの信号完全性、電磁妨害などを分析し、一般的には第三者によりシミュレーションと検証が行われる。

チッププロフラミングとデバッグ：チッププロフラミングは配線レイアウトと実現後に生成された bit ファイルをロジックチップ又は設定チップにダウンロードし、定義された機能を実現させる。デバッグは FPGA チップのオンラインエンベデッドロジックアナライザで（例えば、XILINX ISE における chipScope, ALTERA Quartus における SignalTap）波形を採集し、かつ分析することである。

3.2.2 ダウンロード

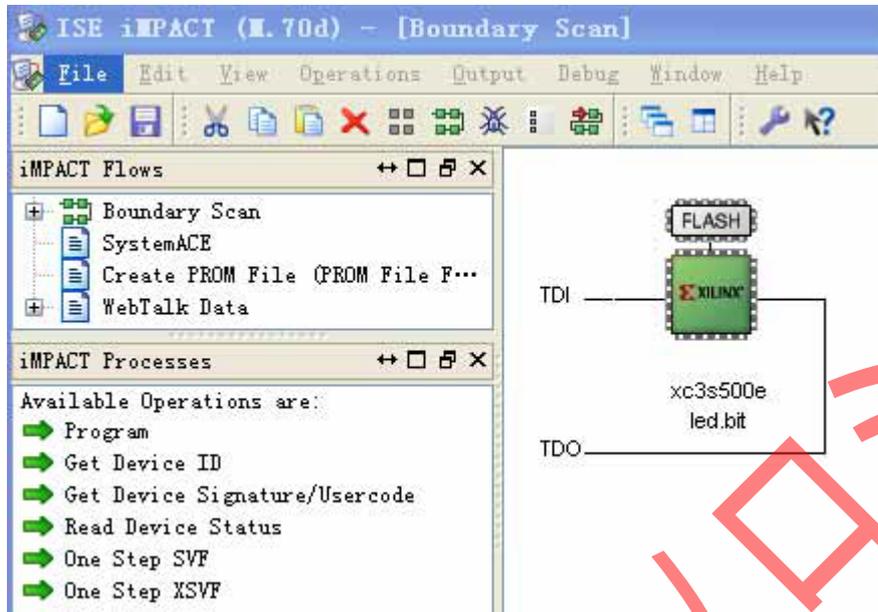
本ボードは JTAG モードと SPI FLASH モードをサポートする。デバッグでは、JTAG モードを使用する。次に、この二種類のモードについて紹介する。

A JTAG モードで FPGA を設定

ダウンロードするファイルは bit ファイルである（ダウンロードスピードが速い）。JTAG インタフェースでロジックを JTAG の内蔵 SRAM にダウンロードする。しかし、電源切れの場合に、SRAM の内容を失って、電源を入れ再設定される。勉強とデバッグの時に、JTAG モードを推奨する。デバッグ成功後、SPI FLASH モード

でロジックを設定チップにダウンロードする。

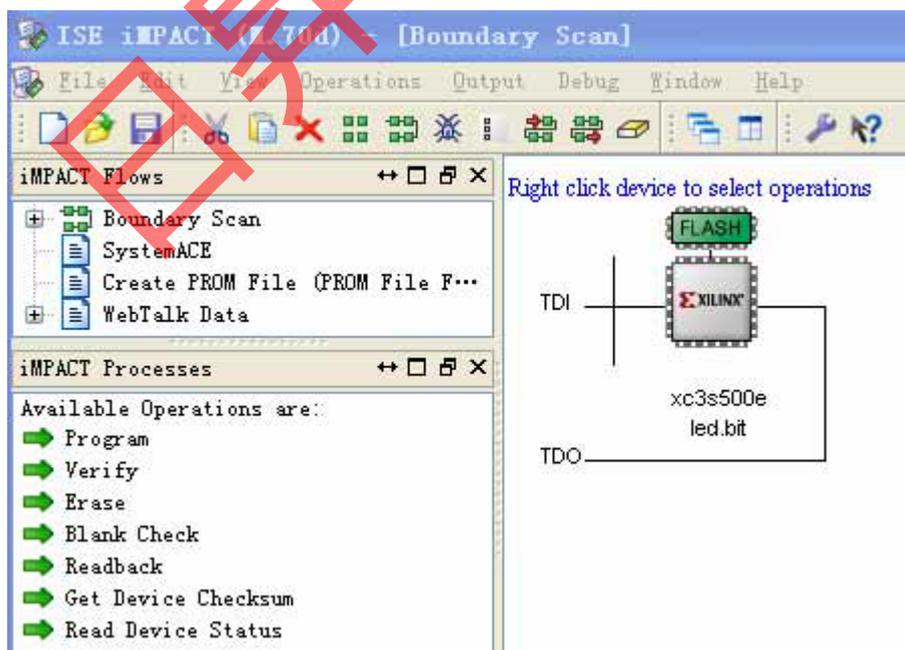
- 1) ISEソフトウェアでbit設定ファイルを生成する。(bit接尾辞)
- 2) ダウンロードケーブルを開発ボードのJTAGコネクタJ7に接続する。
- 3) 開発ボードに電源を入れる。
- 4) ISEの「Tools」に「iMPACT」をクリックし、JEDファイルを選択してから右側のXILINXアイコンをクリックし、次に「program」をダブルクリックしダウンロードする。



B SPI FLASHモードでFPGAを設定

ダウンロードするファイルはmcsファイルである(ダウンロードスピードが遅い)。SPI FLASHモードでロジックを設定チップにダウンロードする(本ボードはM25P16を使用している)。電源切れの場合に、設定チップの内容が失わない。電源を入れ、FPGAが自動的に設定チップのロジックをFPGAにロードする。

ISE 10.1を初め、シリアルチップに対する直接のプログラミングをサポートしないが、JTAGで間接にプログラミングする。(上図と下図の右側のアイコンの色を注意してください)



Mcsファイルを生成する方法と、spi flashプログラムプロジェクトを構築する方法は次を参照ください。

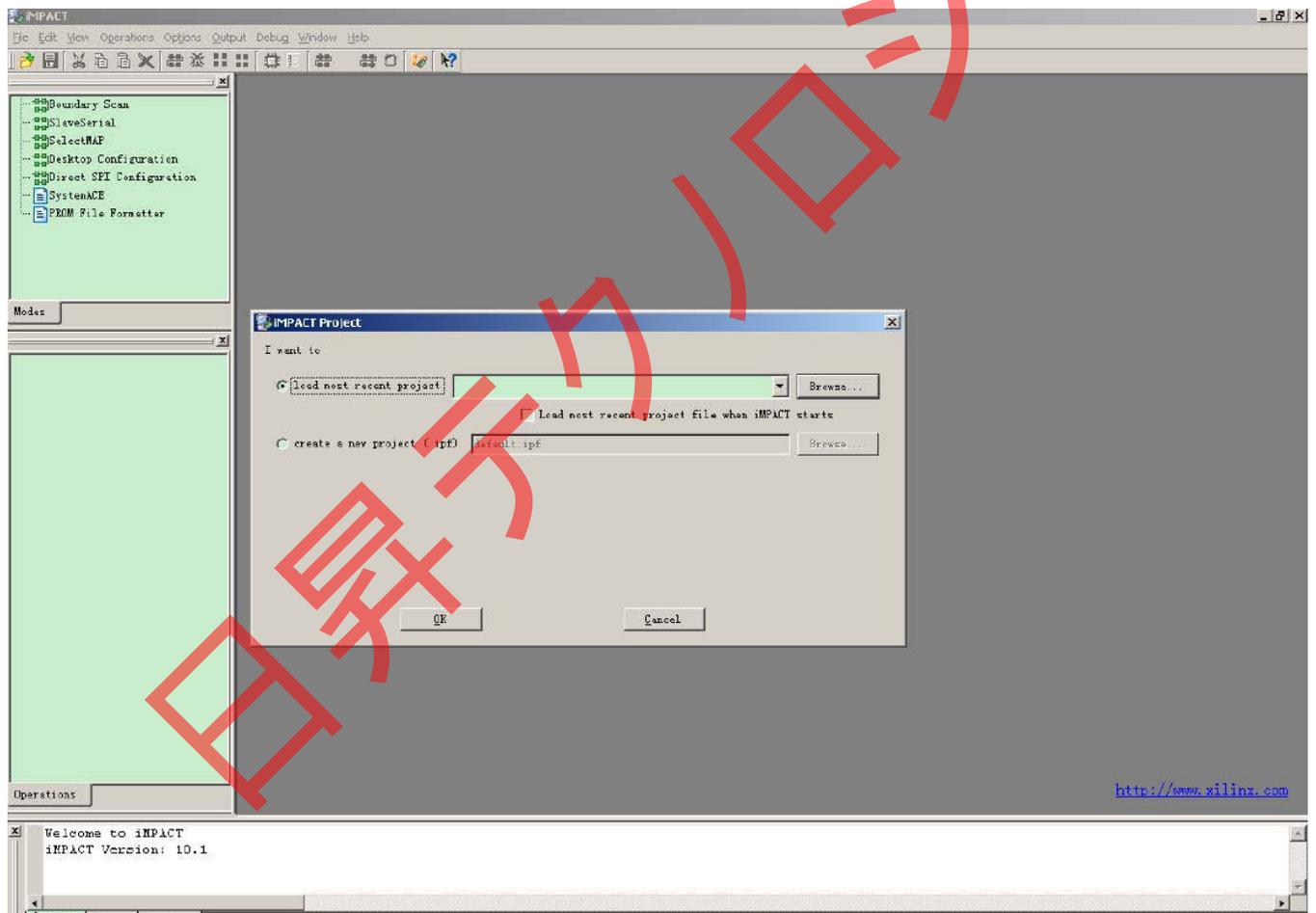
3.2.3 SPI Flash ダウンロード

環境：

- ISE10.1バージョン以上。
- ダウンロードケーブル
- Windows XP
- 開発ボード
- M2:M0=001、MASTER SPI MODE(本ボードではハードウェア回路上設定済み)
- VS2:VS0=111(本ボードではハードウェア回路上設定済み)

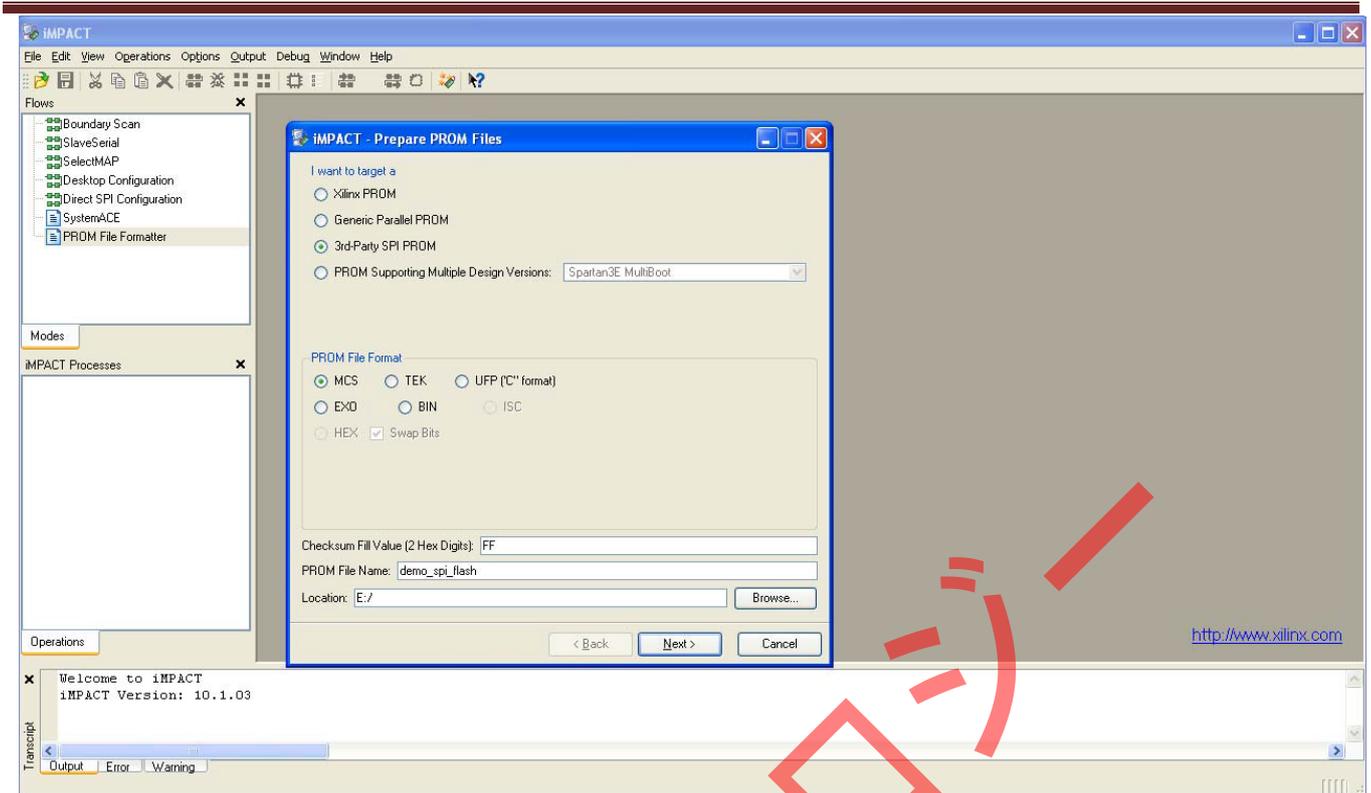
ステップ1、iMPACTを起動

Xilinx ISE Design Suite 10.1->ISE->Accessories->iMPACT、起動後の画面：

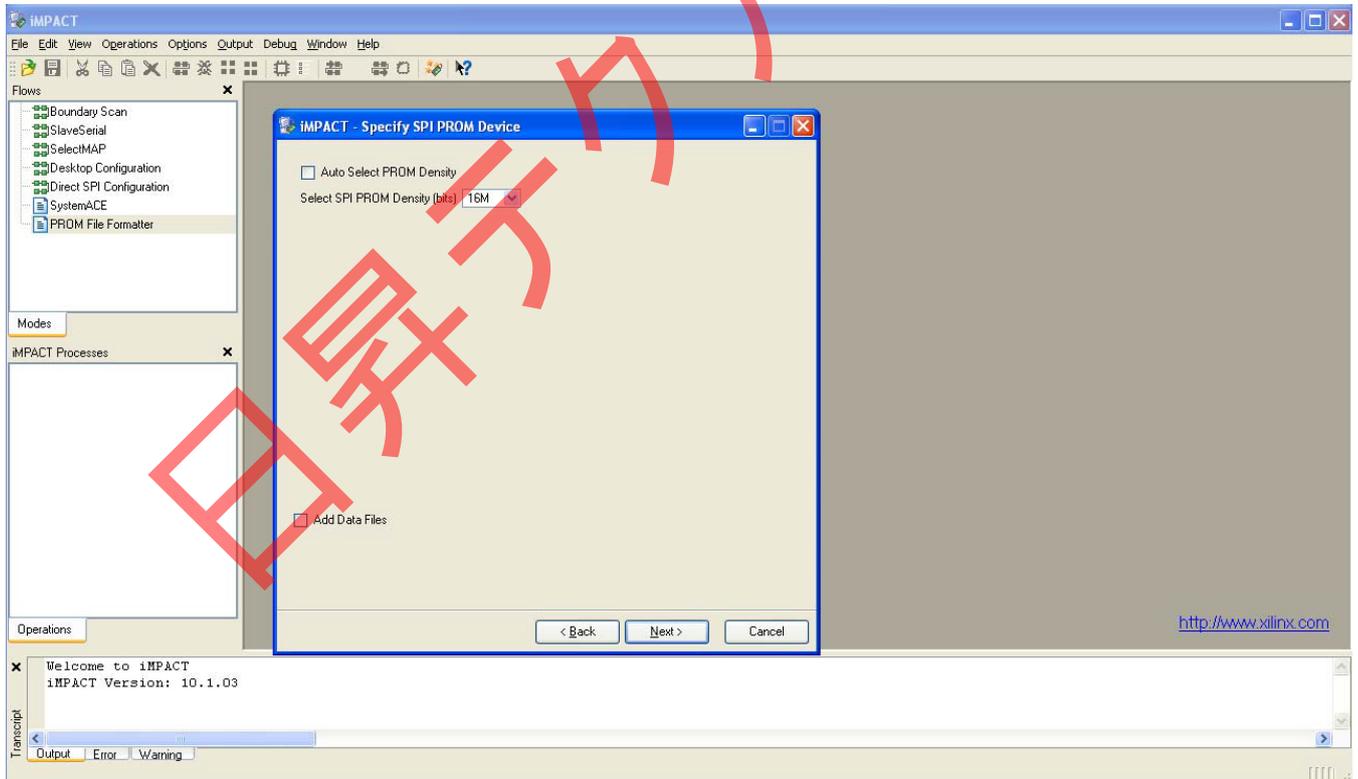


ステップ2、mcsファイルを生成

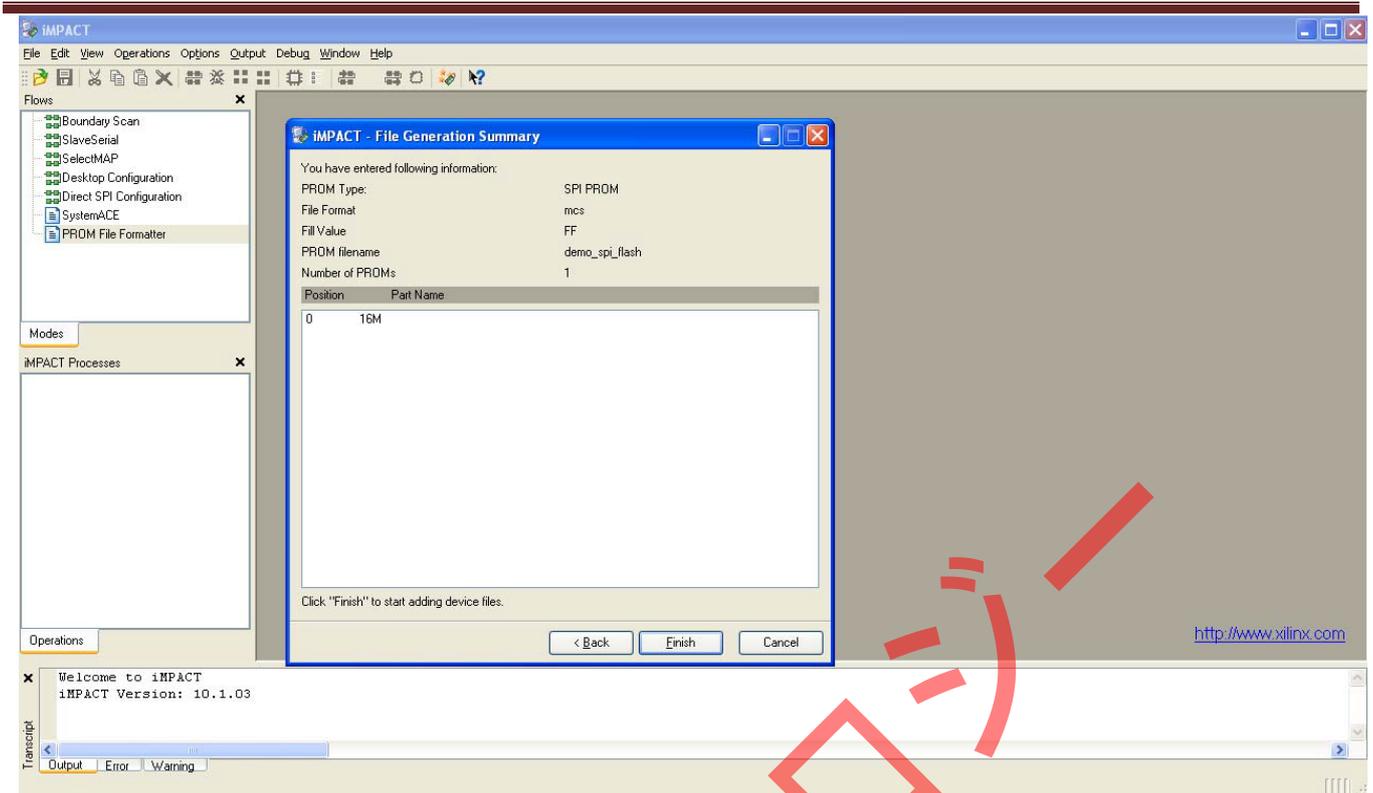
上記ウィンドウの左側「Flows」の一番下にある「PROM File Formatter」をダブルクリックして、下記の様に設定する：



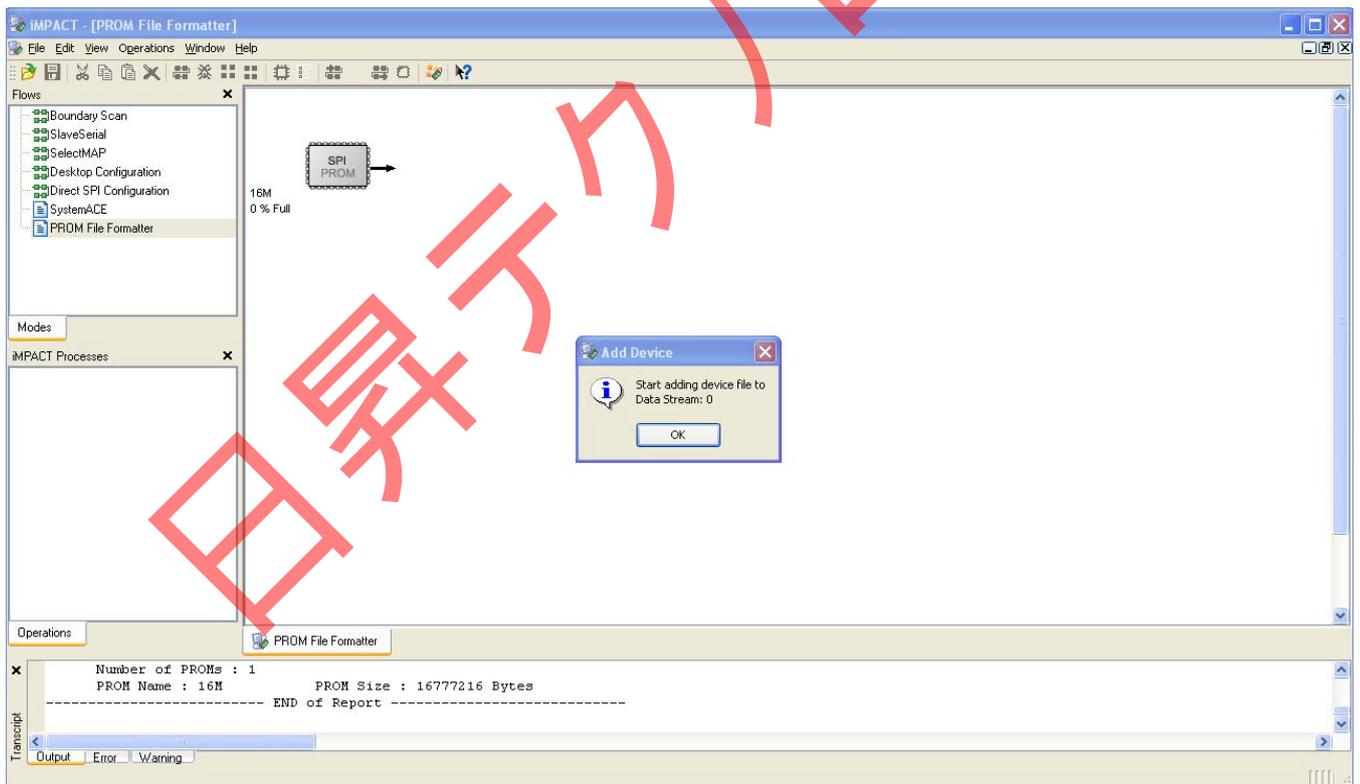
「Next」をクリックして、提示された画面で「Select SPI PROM Density(bits)」の所のリストから「16M」を選択する。本開発ボードで搭載しているSPI FlashはM25P16で容量は16M bitsである。



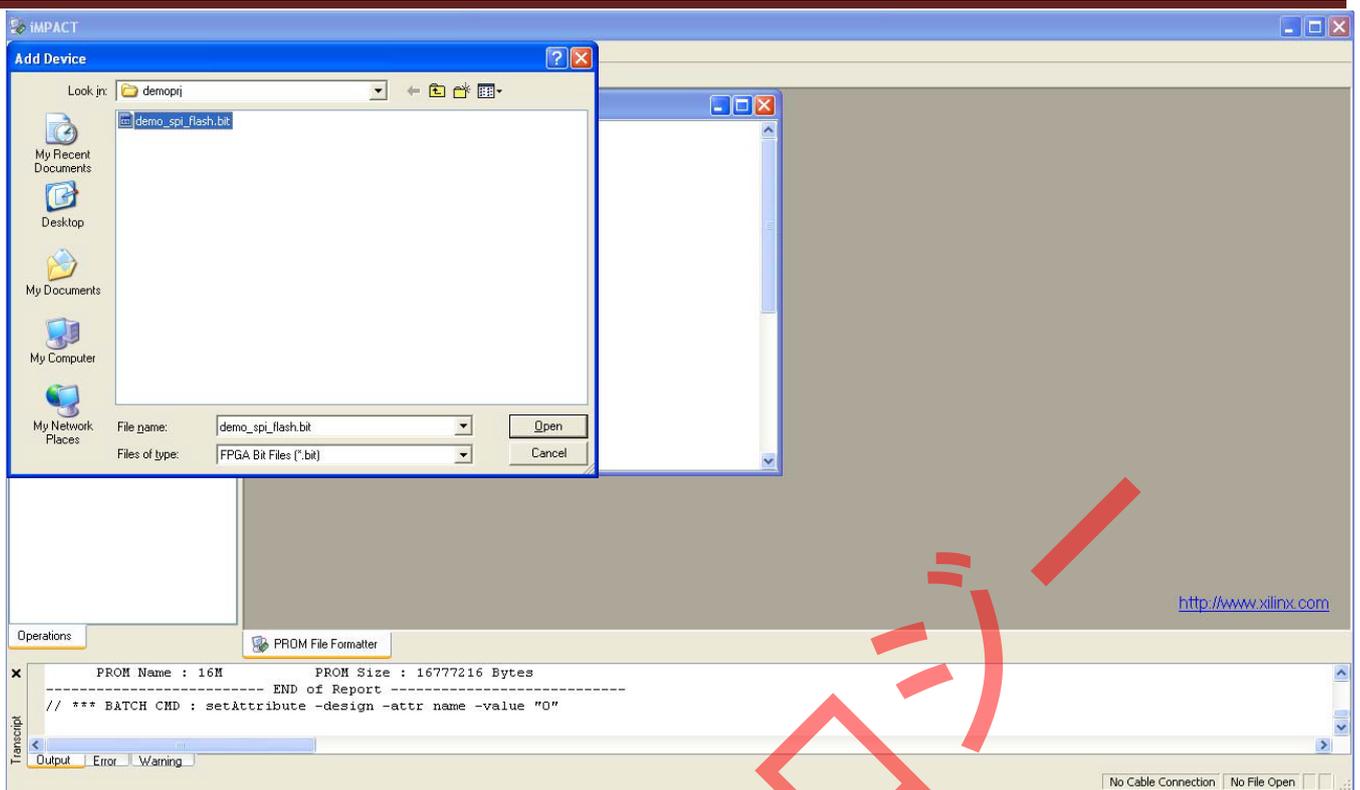
「Next」をクリックすると：



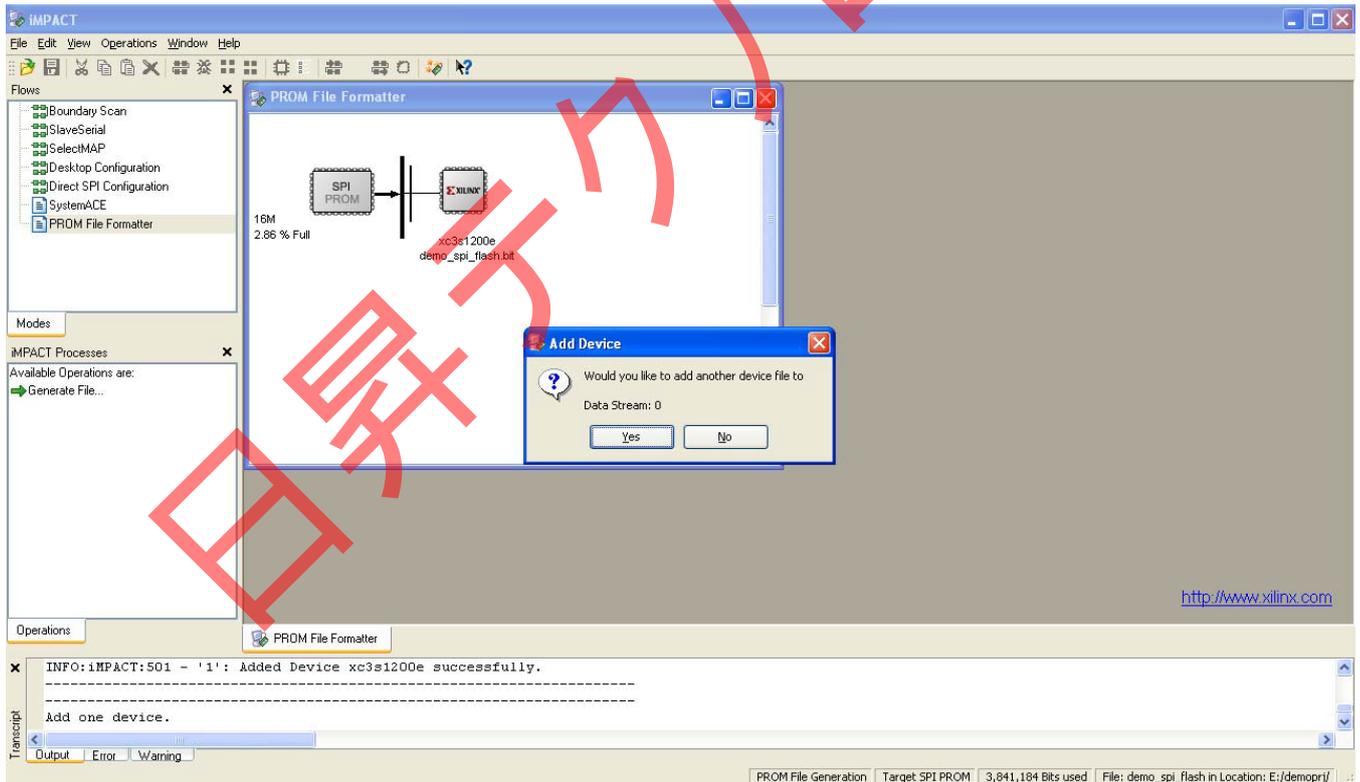
「Finish」をクリックする：



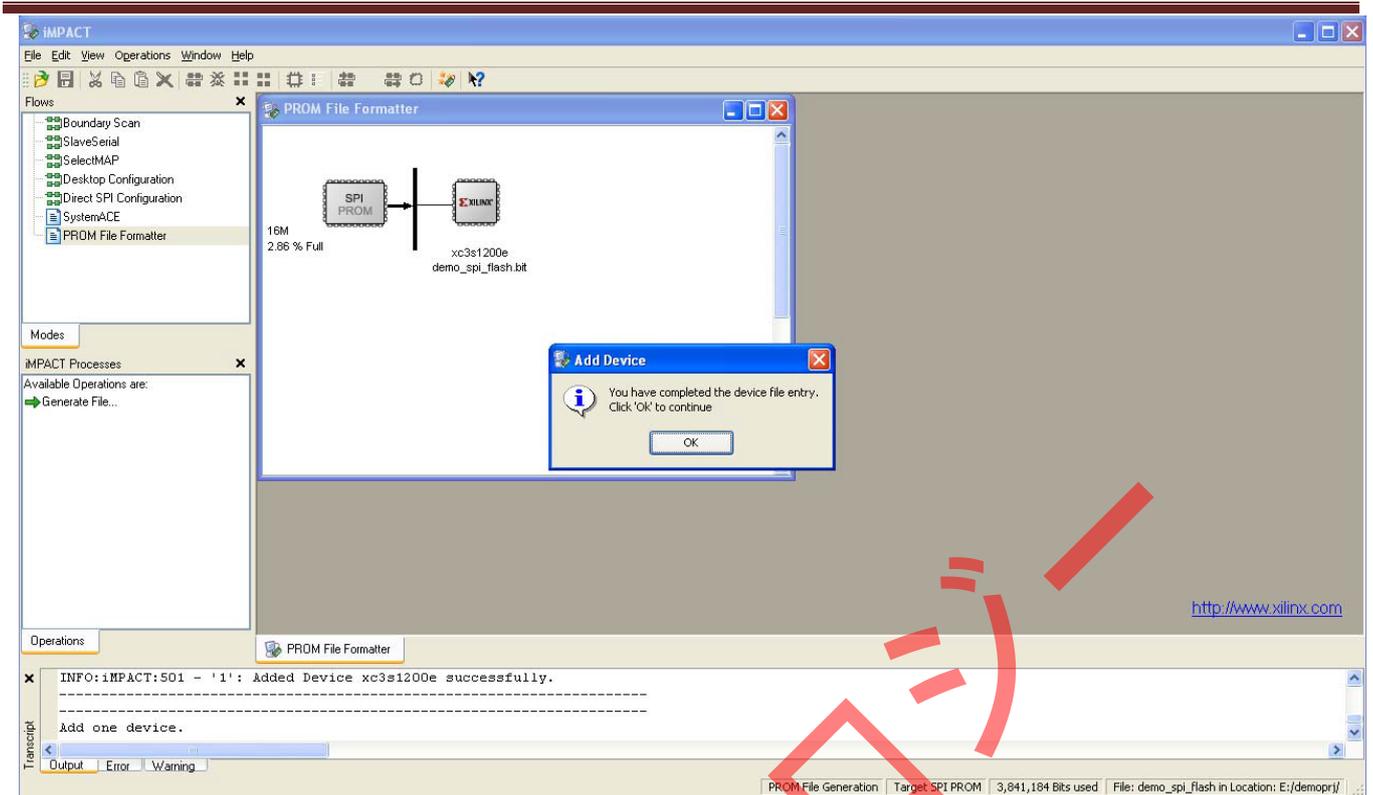
「OK」をクリックする：



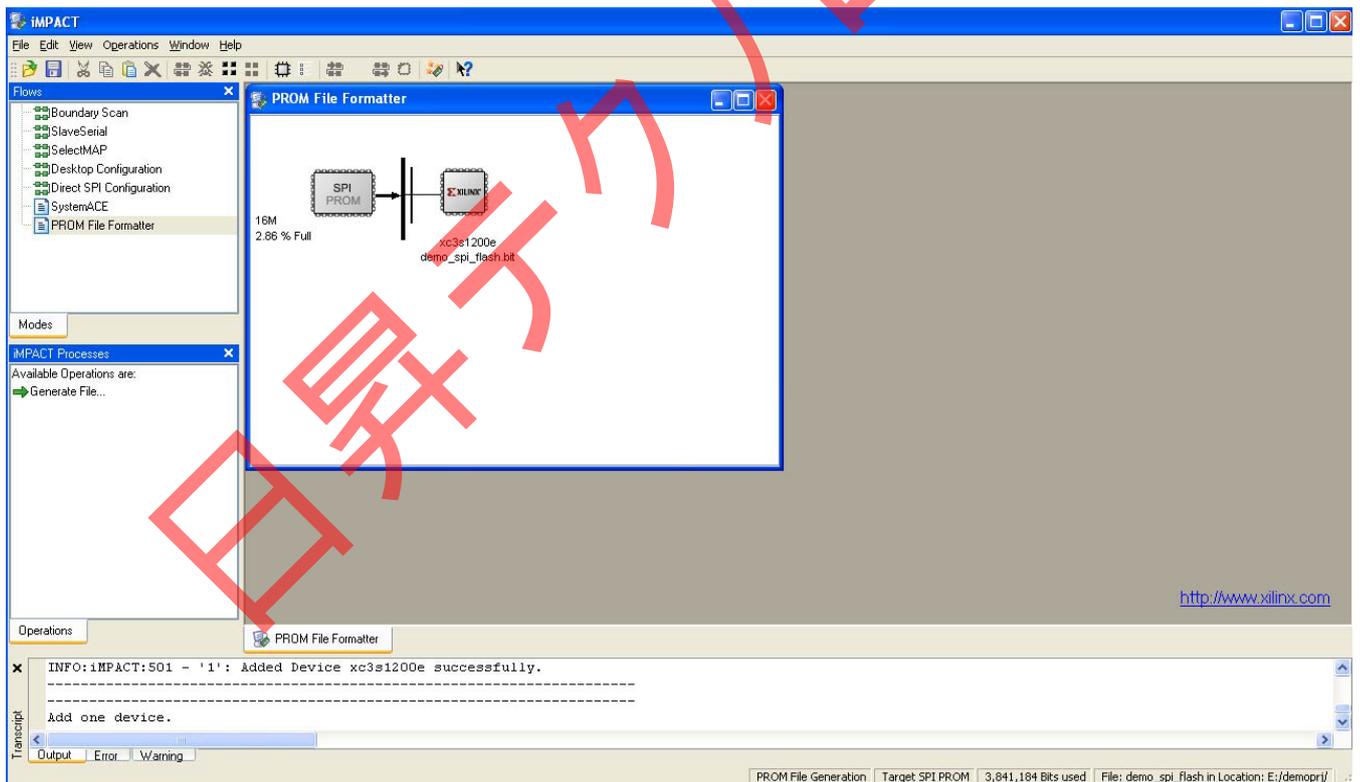
bitファイルを選択して「Open」をクリックする：



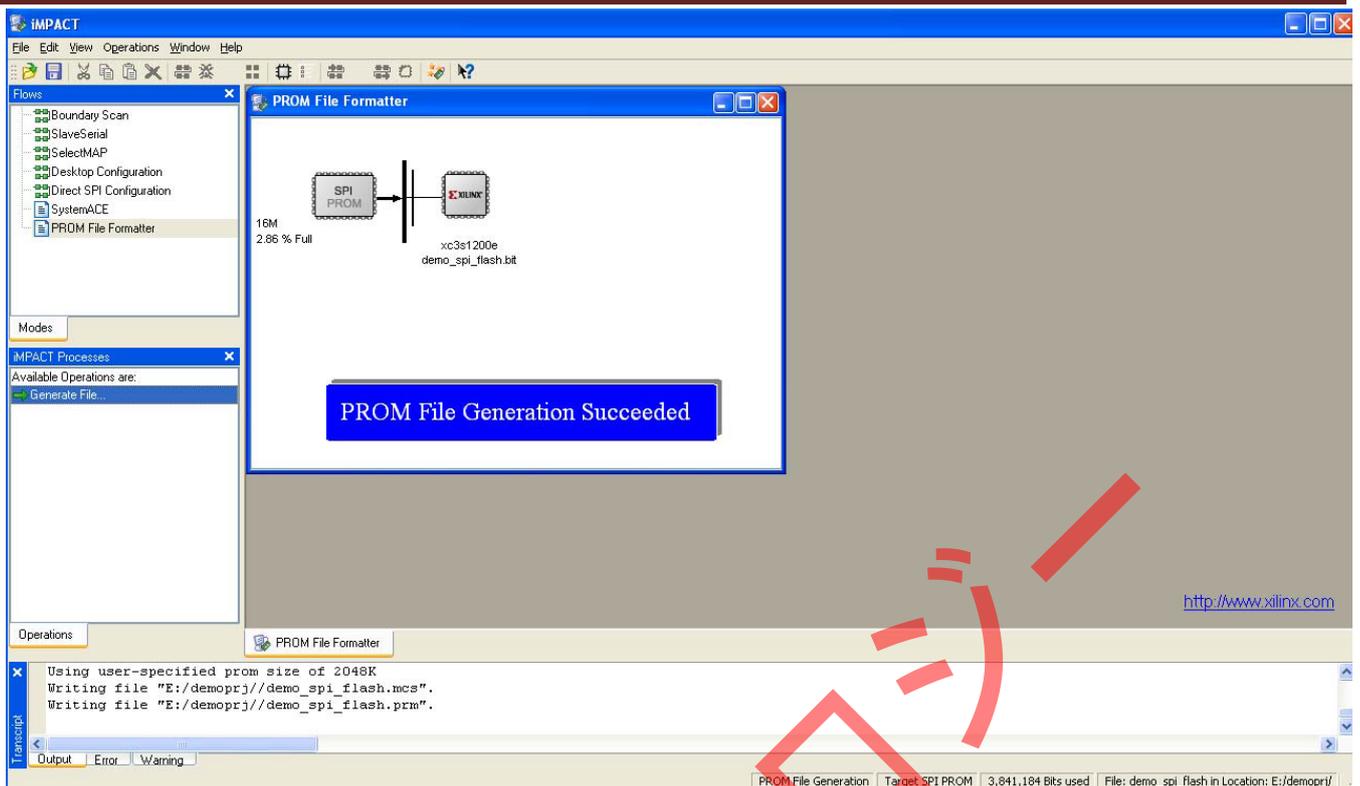
「Would you like to add another device file to Data Stream:0」の提示画面で「No」をクリックする：



「OK」をクリックする：



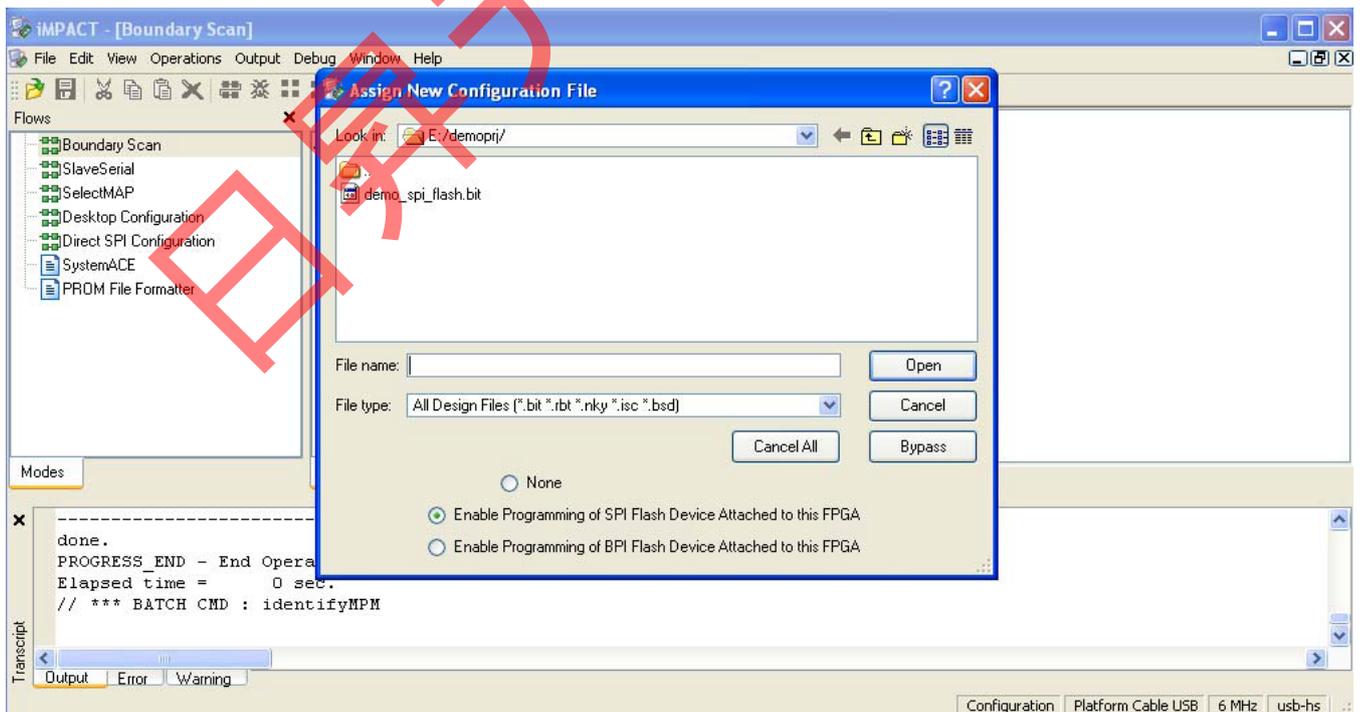
左側の「iMPACT Process」の「Generate File...」をダブルクリックして、mcsファイル生成画面が出る：



これで、SPI FLASHにダウンロード用のmcsファイルが生成される。

ステップ3、bitファイル及びmcsファイルをSPI FLASHにダウンロード

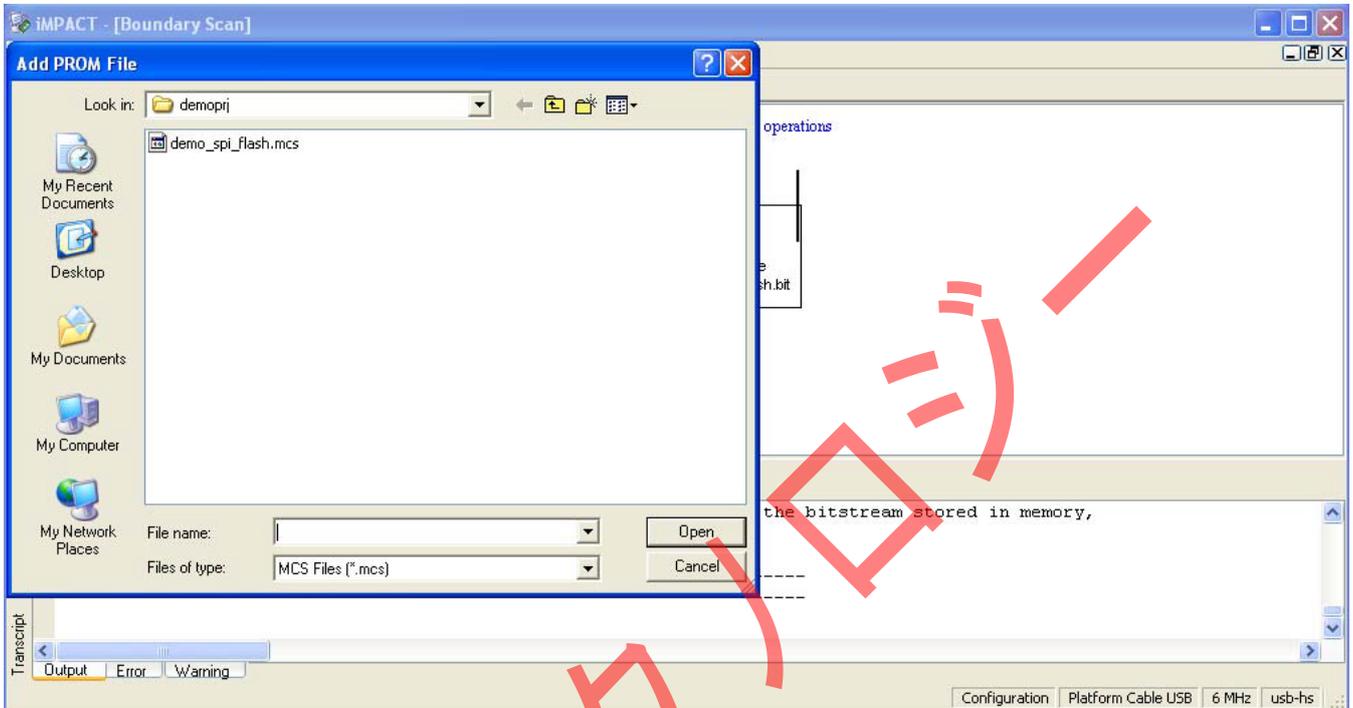
左側の「Flows」の「Boundary Scan」をダブルクリックする。  アイコンが緑色になる。このアイコンをクリックする。



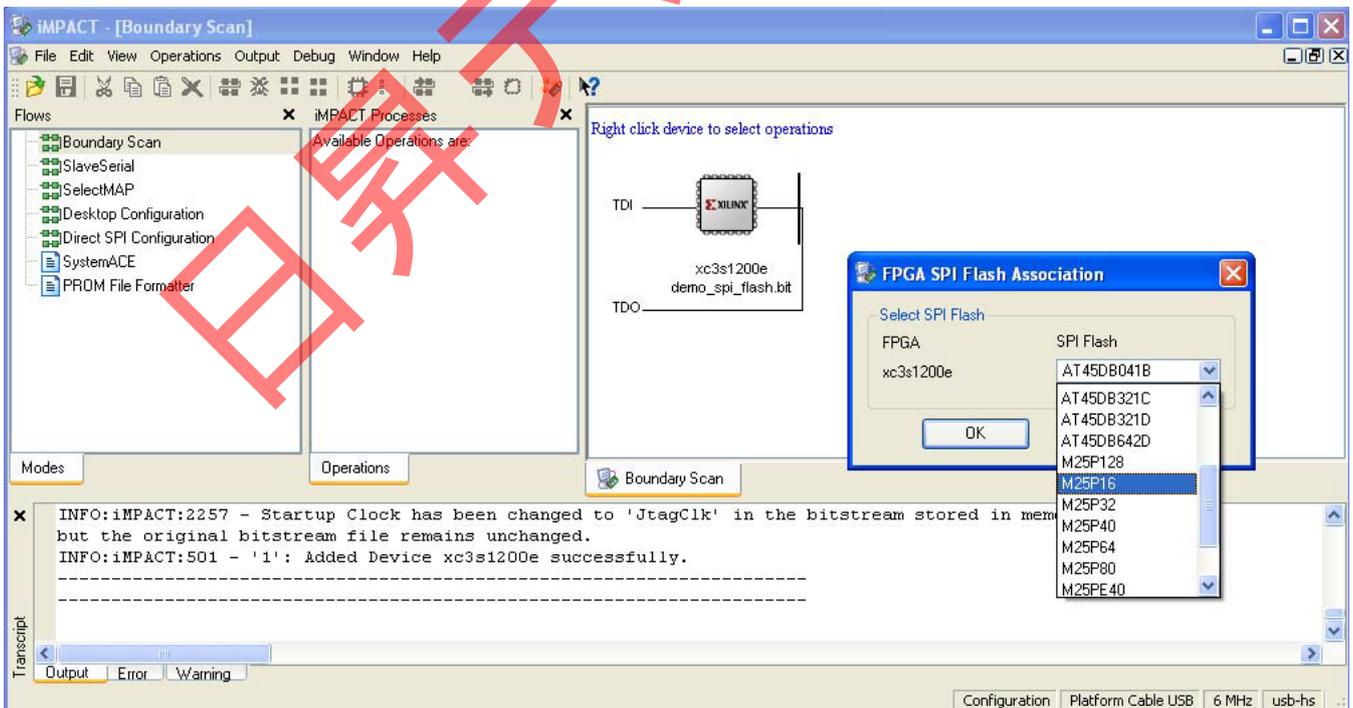
bitファイルを選択する。また下側にある

- None
- Enable Programming of SPI Flash Device Attached to this FPGA
- Enable Programming of BPI Flash Device Attached to this FPGA

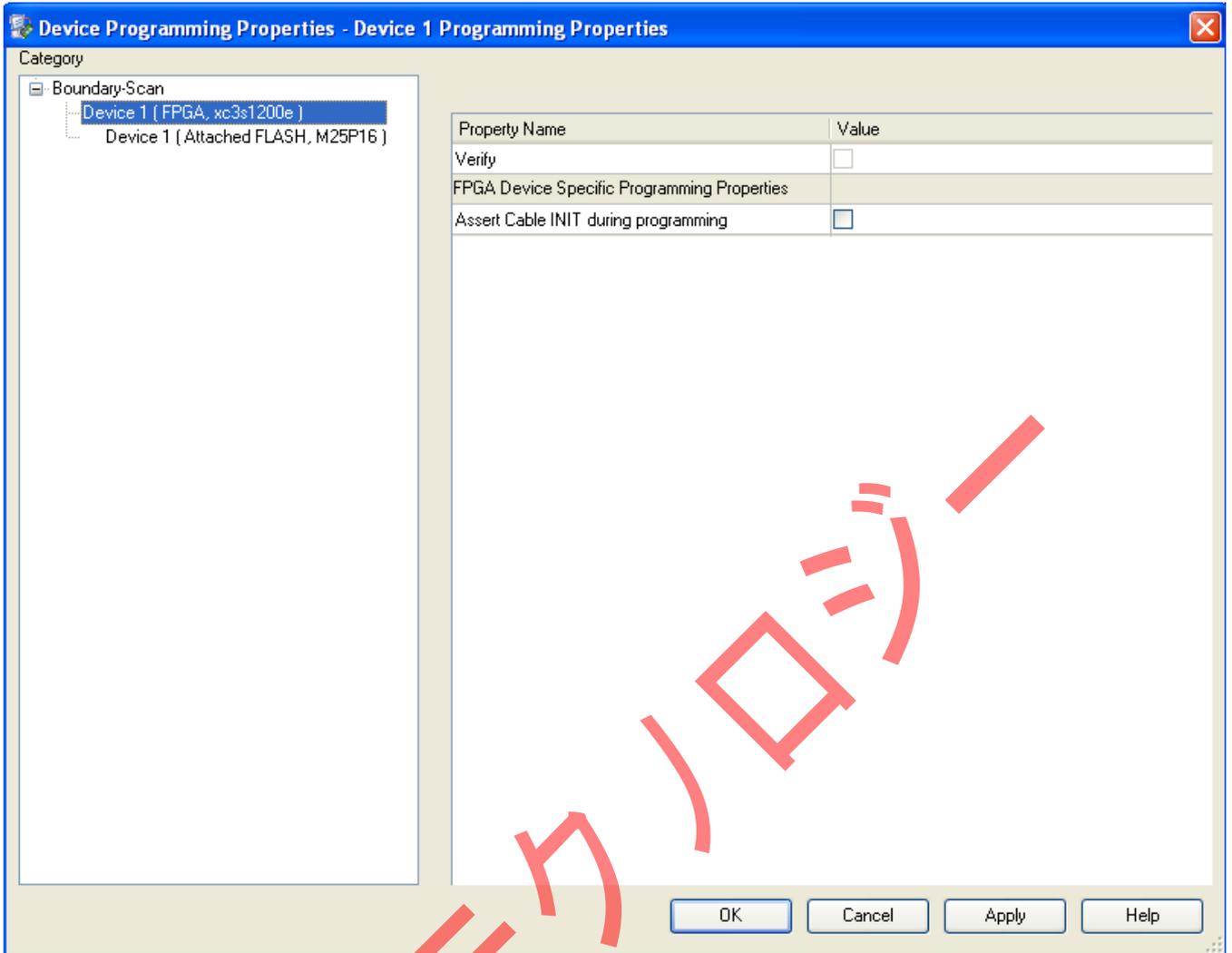
デフォルトでは「None」が選択されているが、SPI FLASHにダウンロードするので、必ず「Enable Programming of SPI Flash Device Attached to this FPGA」を選択必要。「Open」をクリックする：



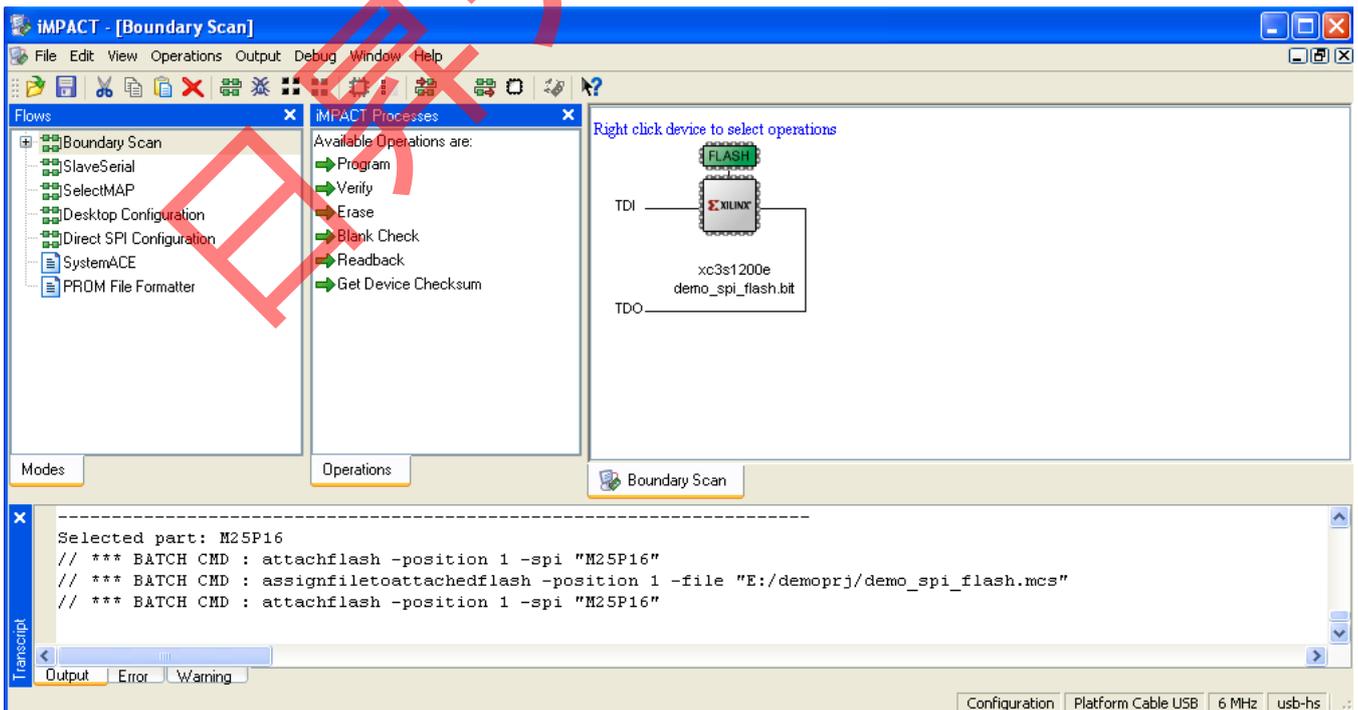
生成されたmcsファイルを選択する。「Open」をクリックする：



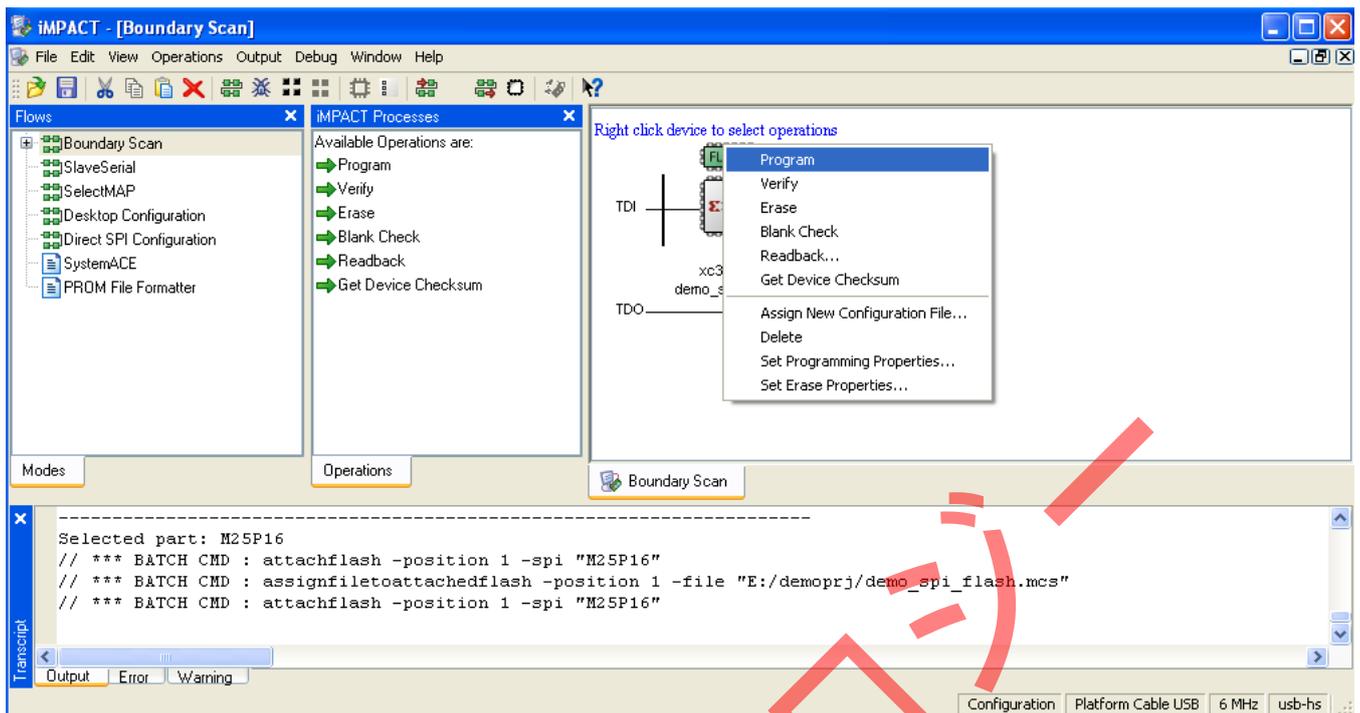
「FPGA SPI Flash Association」の提示画面でボードに搭載されているSPI FLASH型番を選択する。ここでは「M25P16」を選択して、「OK」をクリックする：



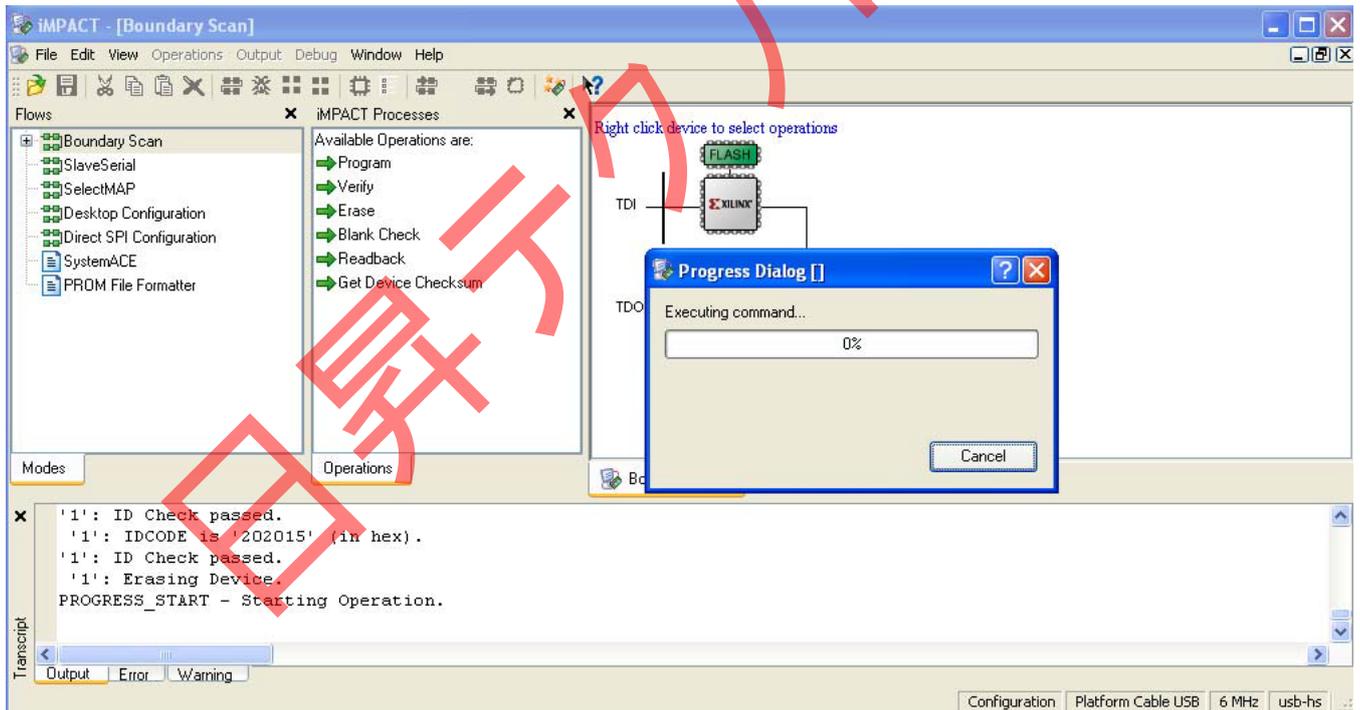
「OK」をクリックする：



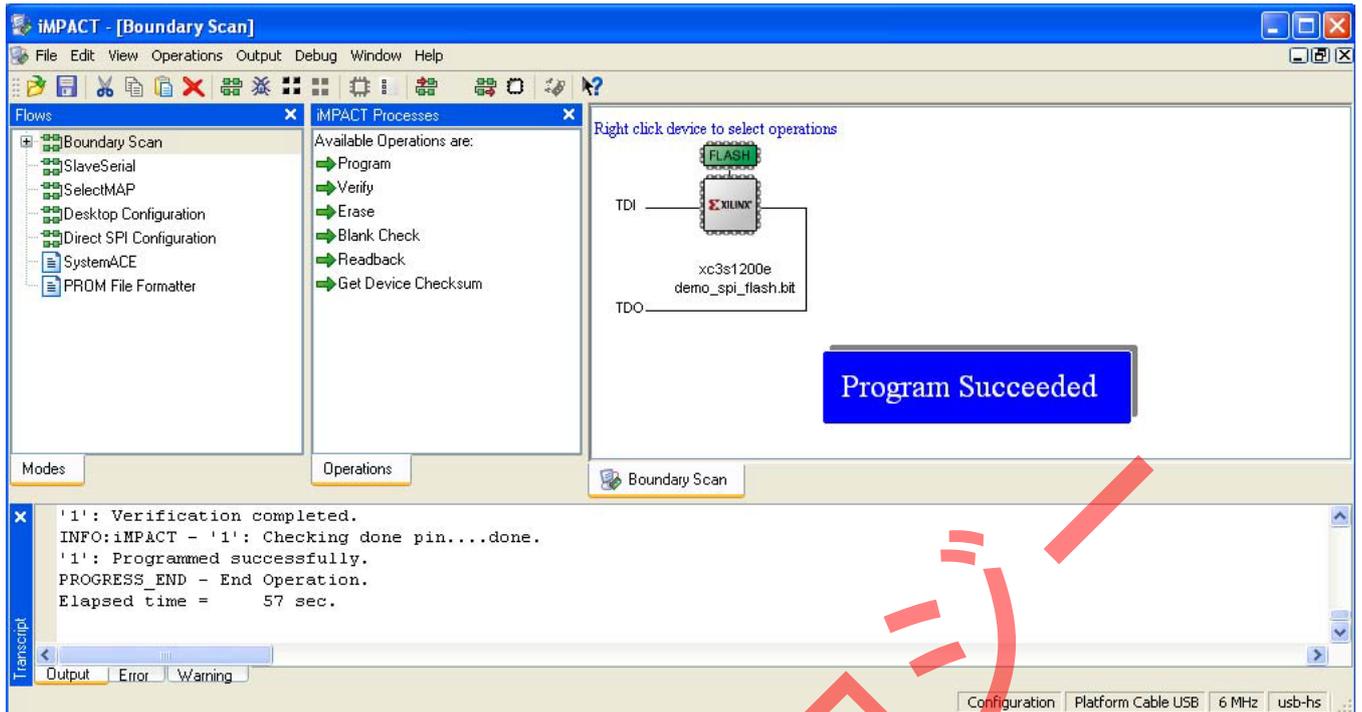
「FLASH」アイコンをクリックして緑色になる。右クリックして下記画面が表示する：



「Program」をクリックする。



ダウンロードの進捗が表示される。
 ダウンロード完了したら、下記画面が表示する：



これまで、SPI FLASHダウンロード完了。ボードを再起動すると、SPI FLASHからデータを読んでFPGAを設定する。

4、開発ボード回路説明

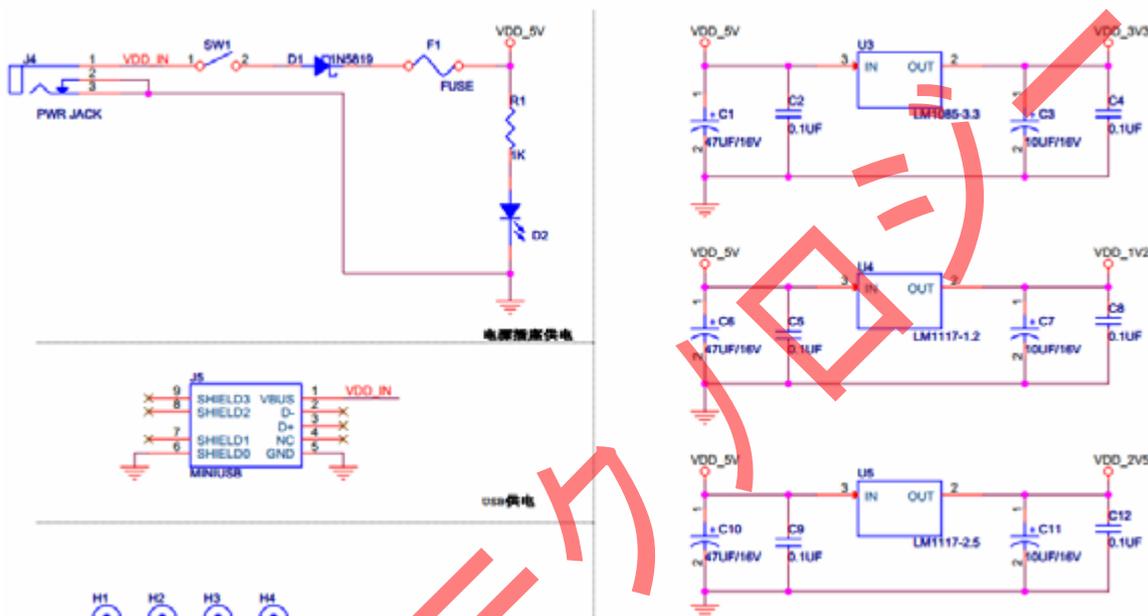
4.1 電源回路

本開発ボードは外部から 5V の電源を投入し、LM1085-3.3V レギュレーターを通じて 3.3V 電源を出力する。主に FPGA IO インタフェース、メモリーSDRAM、シリアル設定デバイス、リセット回路用の電源とする。

また 5V は AMS1117-2.5/1.2V のレギュレータに与え、出力電源は FPGA コアとフェーズロック・ループに使用される。

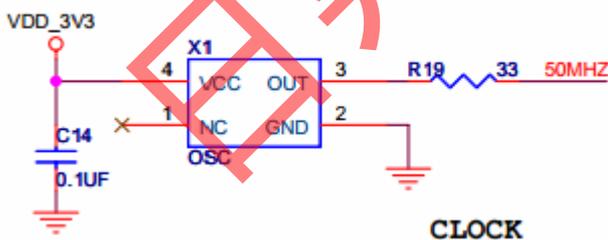
SW2 はセルフロック電源スイッチ、D2 は電源表示灯、D1 は電力の逆接続を防ぐ指示灯である。

同時に、電力システムの安定性を向上させるために、電源入出力の部分に高品質の電解及びタンタルコンデンサを設計した。下図は回路図である。



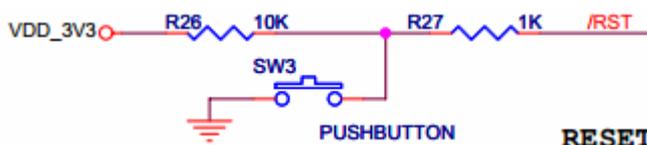
4.2 クロック回路

50MHz水晶発振器でシステムに動作クロックを提供する。下図が回路図である。



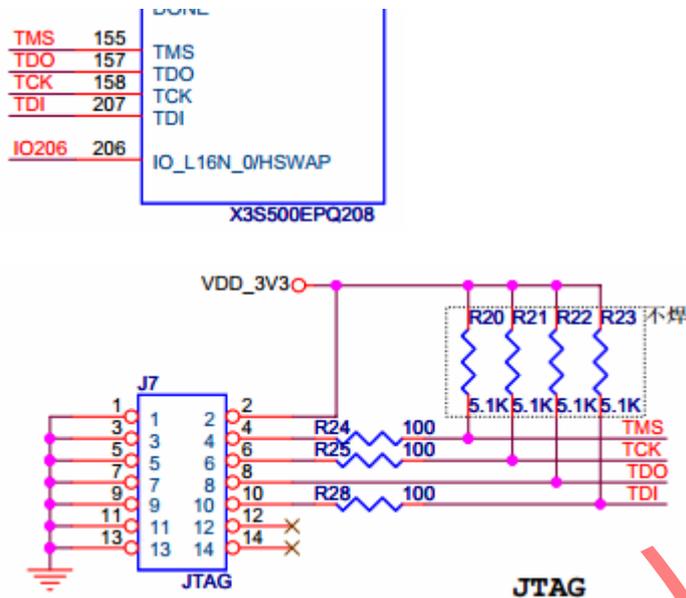
4.3 リセット回路

ローレベルリセット、普通のボタンとしても使える。XC3S500EPQ208 の Pin184 に接続する。下図が回路図である。



4.4 JTAG インタフェース回路

FPGA をダウンロードし、デバッグする時使用される。ダウンロードスピードは早くて、Chipscope サンプルングもできる。電源切れの場合に、FPGA のロジックが失う。デバッグの時は、JTAG モードで Platform Cable と一緒にダウンロードしデバッグすることを推奨する。

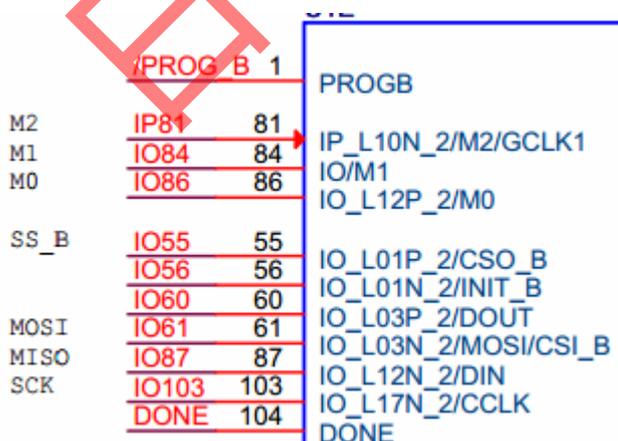


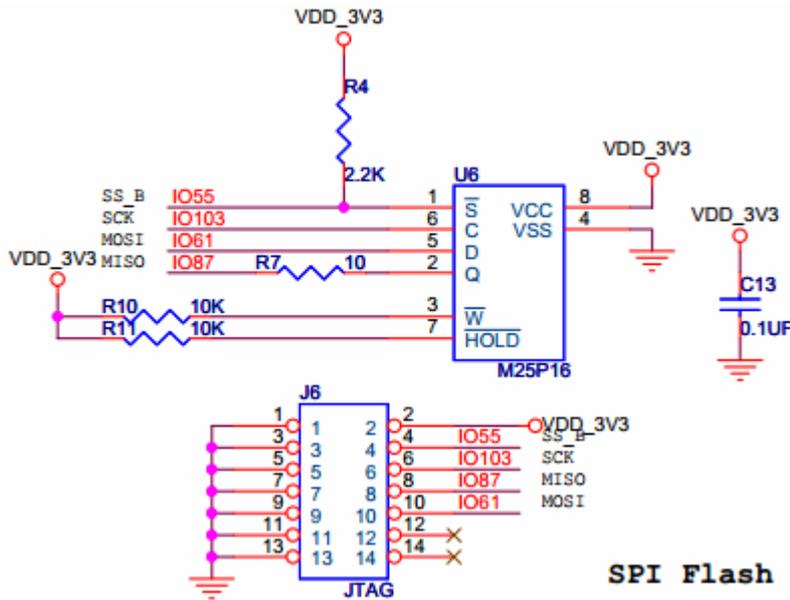
注意：ロジック内部がプルアップされたため、TMS と TDI はプルアップ抵抗を使用しない。

4.5 SPI FLASH インタフェース回路

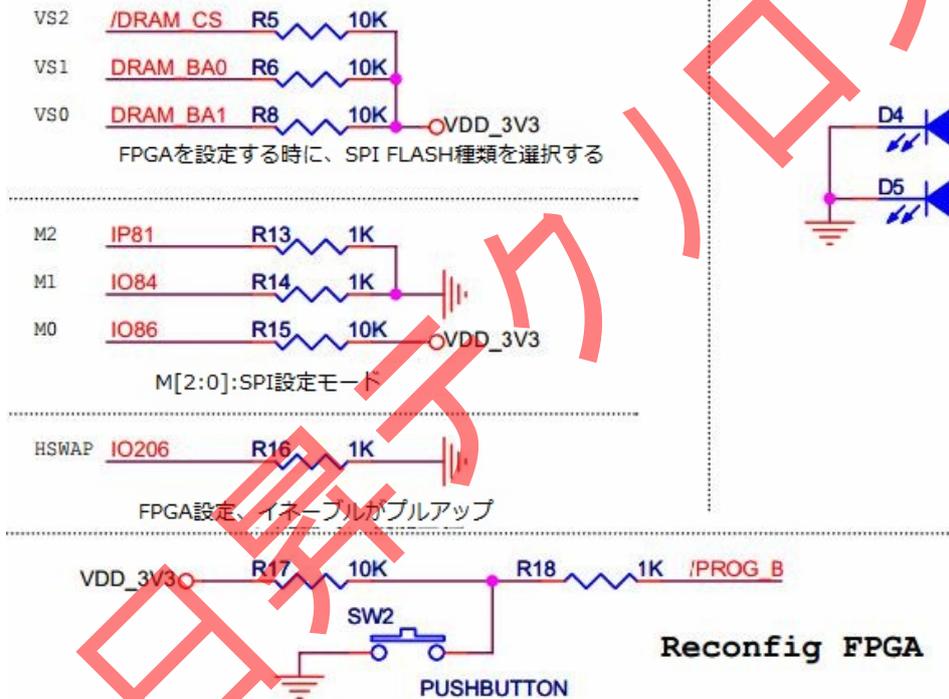
SPI FLASH インタフェース、M25P16 設定チップ、M[2:0]設定モードの選択、VS[2:0] SPI Flash 種類の選択、再設定キーなどがある。ボードに電源を入れ、または再設定キー SW2 を押すと、FPGA が自動的に M25P16 のロジックをロードし FPGA を再設定する。

SPI FLASH ダウンロードモードでは、M25P16 チップにプログラミングする。本ボードは 16Mbit チップを設定し、FPGA ロジックと SOPC プログラムを設定チップに格納する。詳細：a JTAG インタフェースで間接に M25P16 をプログラミングする。b 電源を入れ、又は再設定した後、FPGA が自動的に M25P16 のロジックをロードし FPGA を再設定する。c FPGA がロジックを実行し、動作する。以下は回路図である。



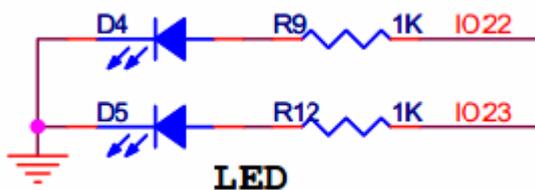


SPI Flash



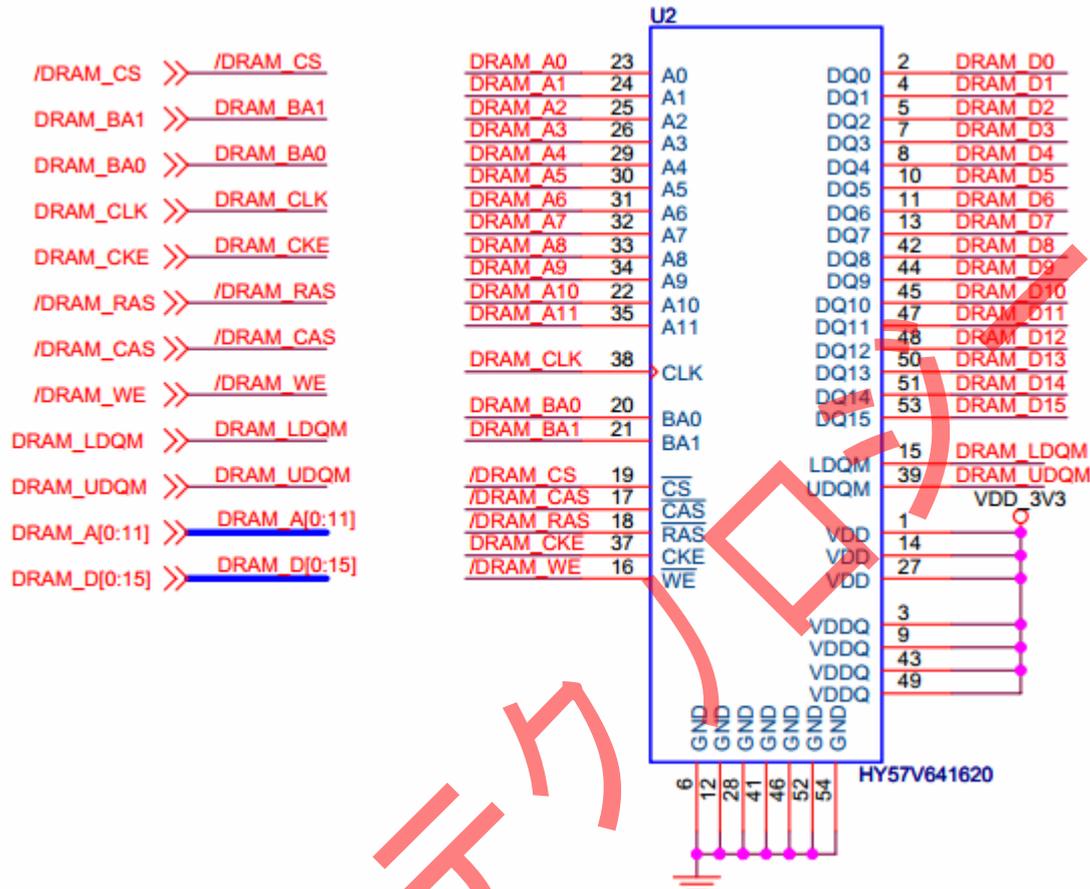
4.6 LED 回路

LED 二つ、ハイレベルで点灯する。LED 試験に利用する。例えば、点灯、点滅など。下図が回路図である。



4.7 メモリー-SDRAM 回路

開発ボードのメモリーはSDRAM、タイプはHY57V641620FTP、64Mbitである。SDRAMアドレス線はA0～A11、データワイドビットは16ビット、0.1μFのコンデンサでフィルタ処理を行って、チップが安定に動作する。



SDRAM と XC3S500EPQ208 の接続と制約の関係は、以下のピン制約をご参照ください。

```

NET "sdr_am_ba[0]" LOC = P103;
NET "sdr_am_ba[1]" LOC = P104;
NET "sdr_am_cas_n" LOC = P99 ;
NET "sdr_am_cke" LOC = P117;
NET "sdr_am_clk" LOC = P116;
NET "sdr_am_dq[0]" LOC = P151;
NET "sdr_am_dq[1]" LOC = P150;
NET "sdr_am_dq[2]" LOC = P149;
NET "sdr_am_dq[3]" LOC = P147;
NET "sdr_am_dq[4]" LOC = P146;
NET "sdr_am_dq[5]" LOC = P145;
NET "sdr_am_dq[6]" LOC = P144;
NET "sdr_am_dq[7]" LOC = P143;
NET "sdr_am_dq[8]" LOC = P114;
NET "sdr_am_dq[9]" LOC = P113;
NET "sdr_am_dq[10]" LOC = P112;
NET "sdr_am_dq[11]" LOC = P110;
  
```

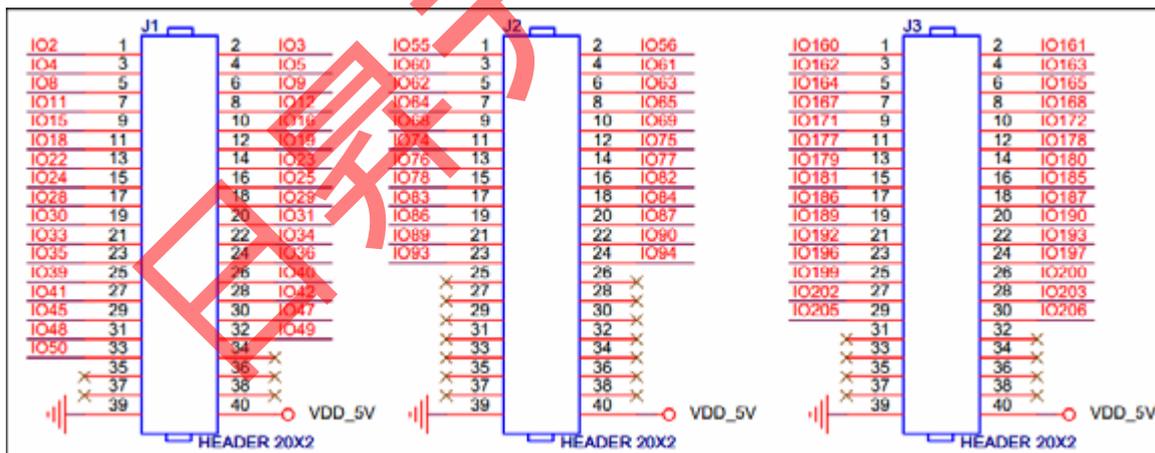
```

NET "sdram_dq[12]" LOC = P108;
NET "sdram_dq[13]" LOC = P107;
NET "sdram_dq[14]" LOC = P106;
NET "sdram_dq[15]" LOC = P105;
NET "sdram_dqm[0]" LOC = P142;
NET "sdram_dqm[1]" LOC = P115;
NET "sdram_ras_n" LOC = P101;
NET "sdram_sa[0]" LOC = P139;
NET "sdram_sa[1]" LOC = P138;
NET "sdram_sa[2]" LOC = P137;
NET "sdram_sa[3]" LOC = P135;
NET "sdram_sa[4]" LOC = P134;
NET "sdram_sa[5]" LOC = P133;
NET "sdram_sa[6]" LOC = P128;
NET "sdram_sa[7]" LOC = P127;
NET "sdram_sa[8]" LOC = P120;
NET "sdram_sa[9]" LOC = P119;
NET "sdram_sa[10]" LOC = P141;
NET "sdram_sa[11]" LOC = P118;
NET "sdram_we_n" LOC = P97 ;
NET "sdram_cs_n" LOC = P102;

```

4.8 外部拡張 IO

開発ボードは三つのピン配列で外部に接続する。連続ピンはダブル列、40pin、ピンの間隔は 2.54mm ピッチ、ピンの定義は下図の通り：



以上。