

OpenCores I2C の LX9 MicroBoard 用設計データの使い方

ファイル一覧

- ・ mb_mcs_sys.v トップ回路
- ・ iobus2wb.v WishBone バスブリッジ
- ・ iobus_reg.v 設定レジスタ回路
- ・ iobus_bram.v ブロック RAM インターフェース回路
- ・ user_module.v デバック回路
- ・ test_mcs_sys.v テストベンチ
- ・ mb_mcs_sys.ucf ピン配置指定
- ・ i2c_sw1.c i2c 制御用 C ソースコード

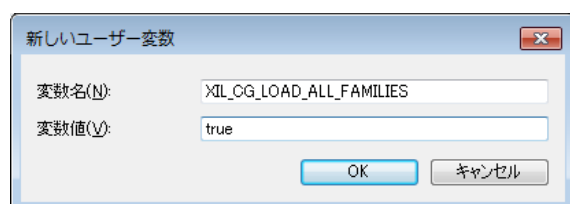
免責事項

本データの使用が原因として発生した損失や損害について、(有) ひまわり および 著作者は一切責任を負いません。著作者：横溝憲治 fpga@hmwr-lsi.co.jp

手順

- ・ 設計用フォル i2c/mb_mcs_sys を作成する
- ・ 記事のダウンロードデータを解凍して、i2c_microborad_data の下にある Verilog-HDL ソースと UCF ファイルを i2c/mb_mcs_sys へコピーする。
- ・ I²C のデータを CopenCores のサイト (<http://opencores.org/project,i2c>) からダウンロード
- ・ ダウンロードした i2c_latest.tar.gz を解凍する
- ・ 解凍データの i2c/trunk/rtl/verilog の下にある Verilog-HDL ソースを i2c/mb_mcs_sys へコピー
- ・ PWM のデータを CopenCores のサイト (<http://opencores.org/project,pwm>) からダウンロード
- ・ ダウンロードした pwm_latest.tar.gz を解凍する
- ・ 解凍データの pwm/trunk/RTL の下にある Verilog-HDL ソースを i2c/mb_mcs_sys へコピー
- ・ 環境変数 XIL_CG_LOAD_ALL_FAMILIES が設定されてない場合は環境変数を追加する

コントロールパネル→システムとセキュリティ→システム→システムの詳細設定→環境変数→ユーザー環境変数：新規をクリック



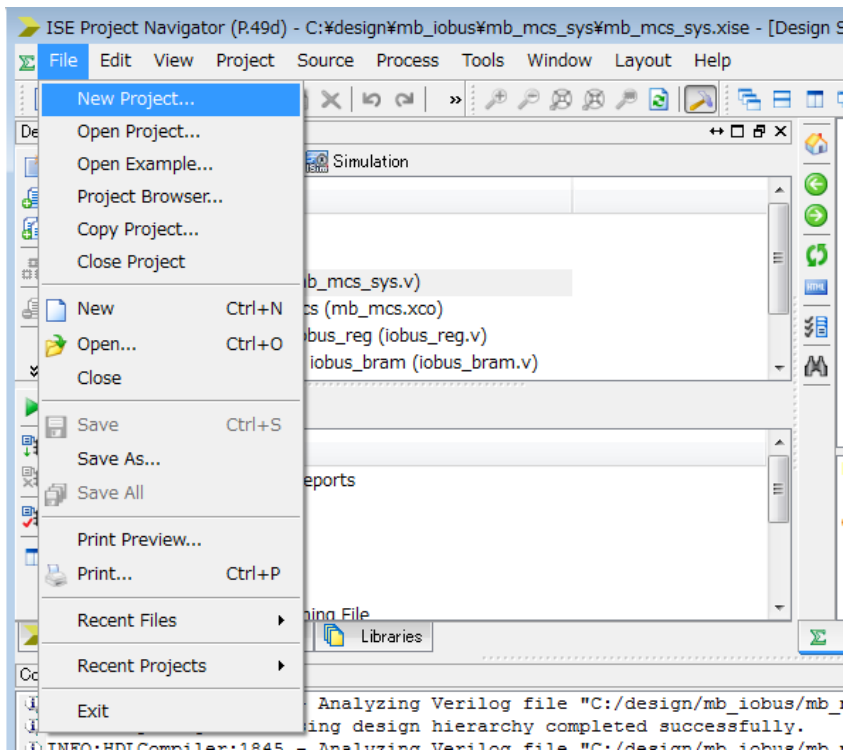
変数名：XIL_CG_LOAD_ALL_FAMILIES、値：true、と入力して OK

- ・ ProjectNavigator を起動します。

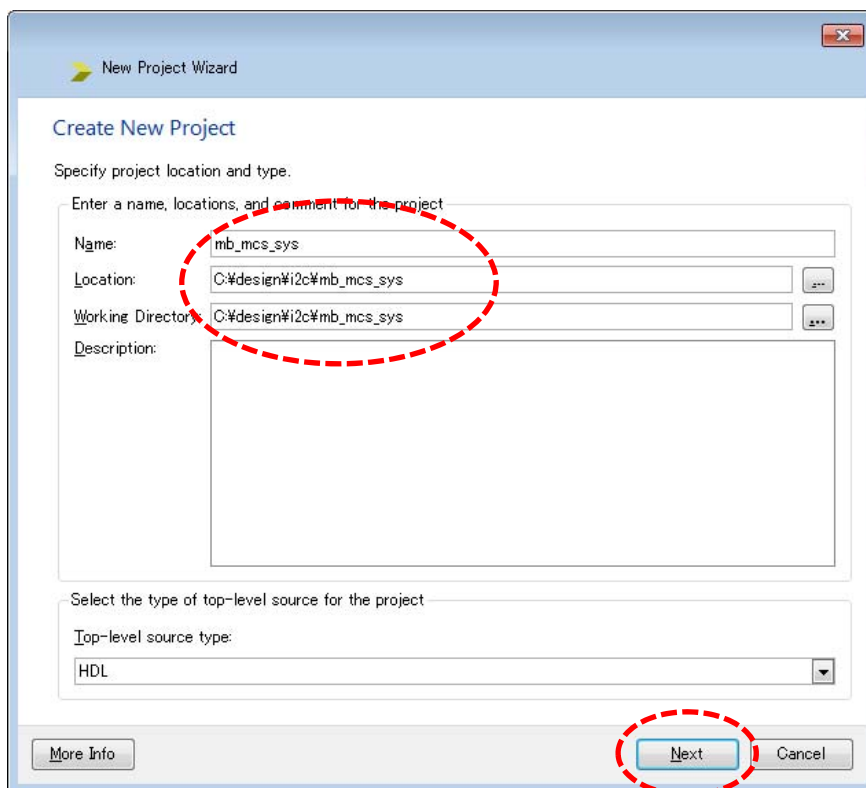
スタートメニューから「Xilinx Design Tools」→「ISE Design Suite 14.4」→「ISE Design Tools」→「Project Navigator」を起動してください。



・新規設計プロジェクト作成

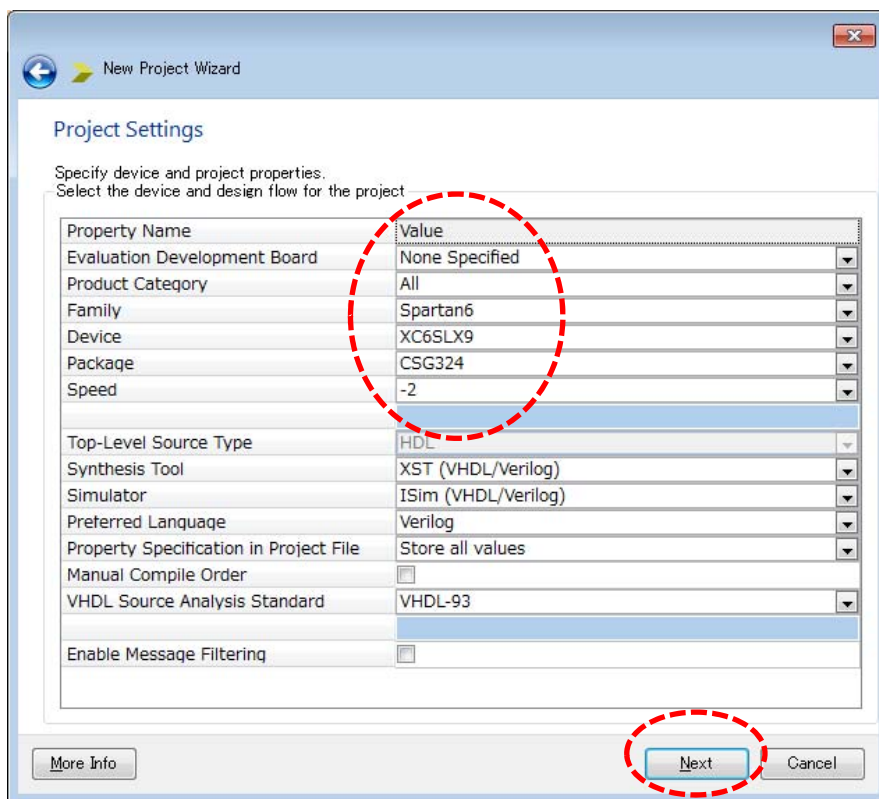


File→New Project 選択



プロジェクト名 : mb_mcs_sys、設計フォルダ : 任意、Next をクリック



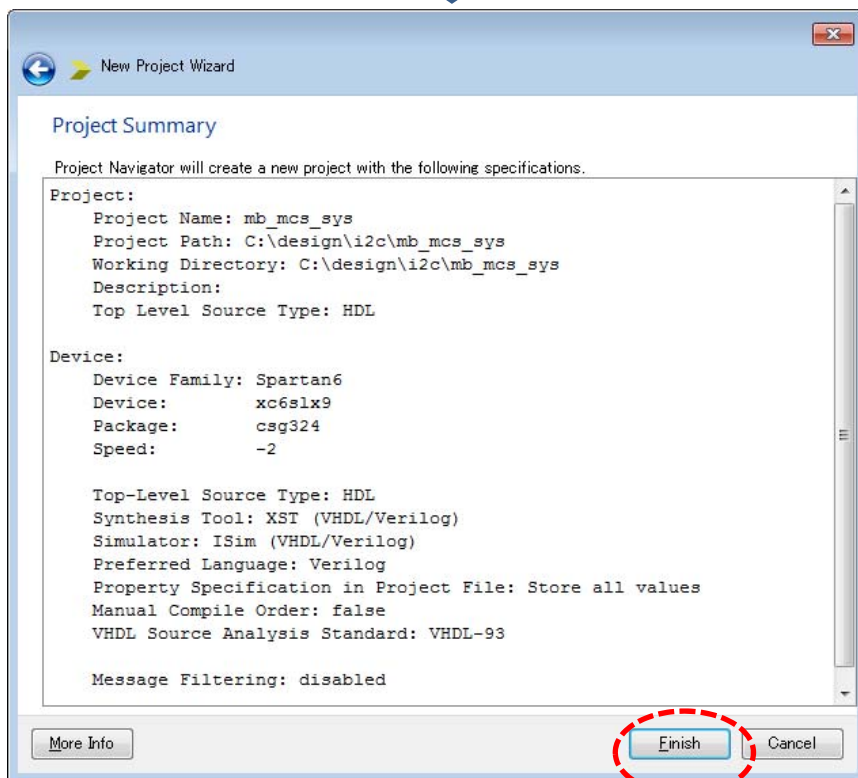


The 'New Project Wizard' dialog box is shown at the 'Project Settings' step. It contains a table of properties and their values. A red dashed circle highlights the 'Device' section, which includes 'Family' (Spartan6), 'Device' (XC6SLX9), and 'Package' (CSG324). Another red dashed circle highlights the 'Next' button at the bottom right.

Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Spartan6
Device	XC6SLX9
Package	CSG324
Speed	-2
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	Verilog
Property Specification in Project File	Store all values
Manual Compile Order	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	<input type="checkbox"/>

Buttons: More Info, Next, Cancel

デバイス指定 LX9 マイクロボードに合わせて、Next をクリック



The 'New Project Wizard' dialog box is shown at the 'Project Summary' step. It displays a summary of the project specifications. A red dashed circle highlights the 'Finish' button at the bottom right.

Project Navigator will create a new project with the following specifications.

Project:

- Project Name: mb_mcs_sys
- Project Path: C:\design\i2c\mb_mcs_sys
- Working Directory: C:\design\i2c\mb_mcs_sys
- Description:
- Top Level Source Type: HDL

Device:

- Device Family: Spartan6
- Device: xc6slx9
- Package: csg324
- Speed: -2

Top-Level Source Type: HDL

Synthesis Tool: XST (VHDL/Verilog)

Simulator: ISim (VHDL/Verilog)

Preferred Language: Verilog

Property Specification in Project File: Store all values

Manual Compile Order: false

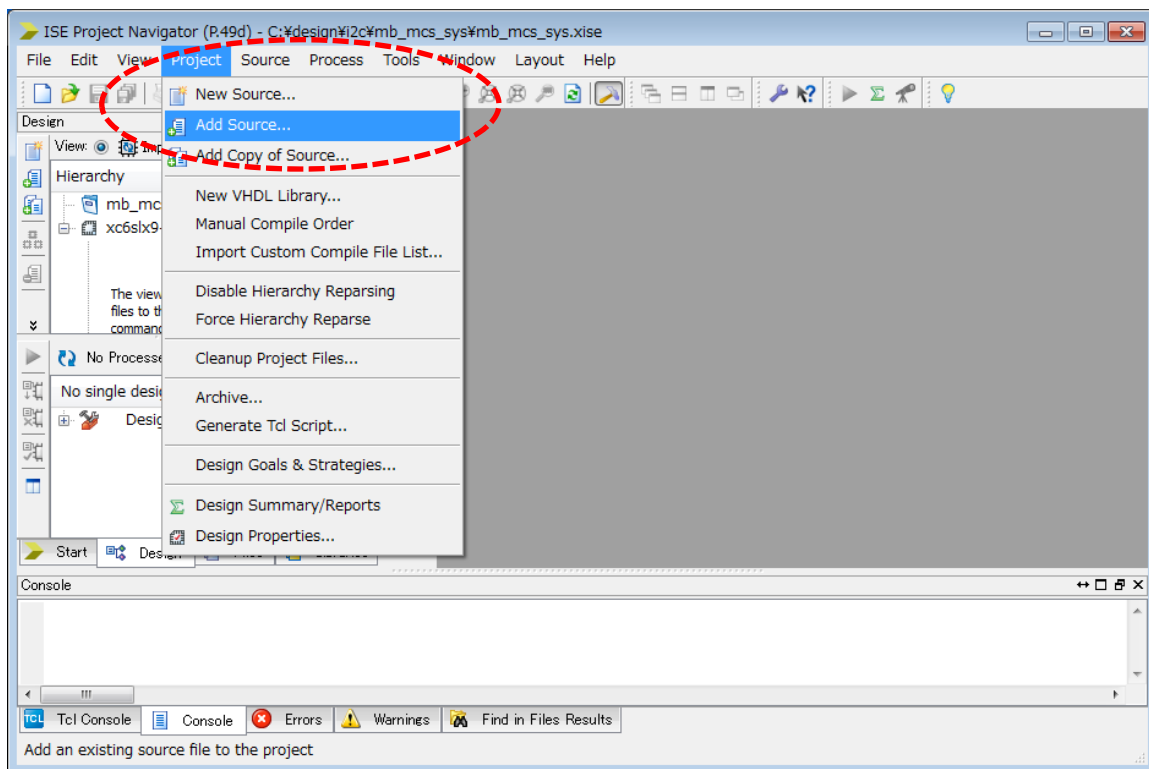
VHDL Source Analysis Standard: VHDL-93

Message Filtering: disabled

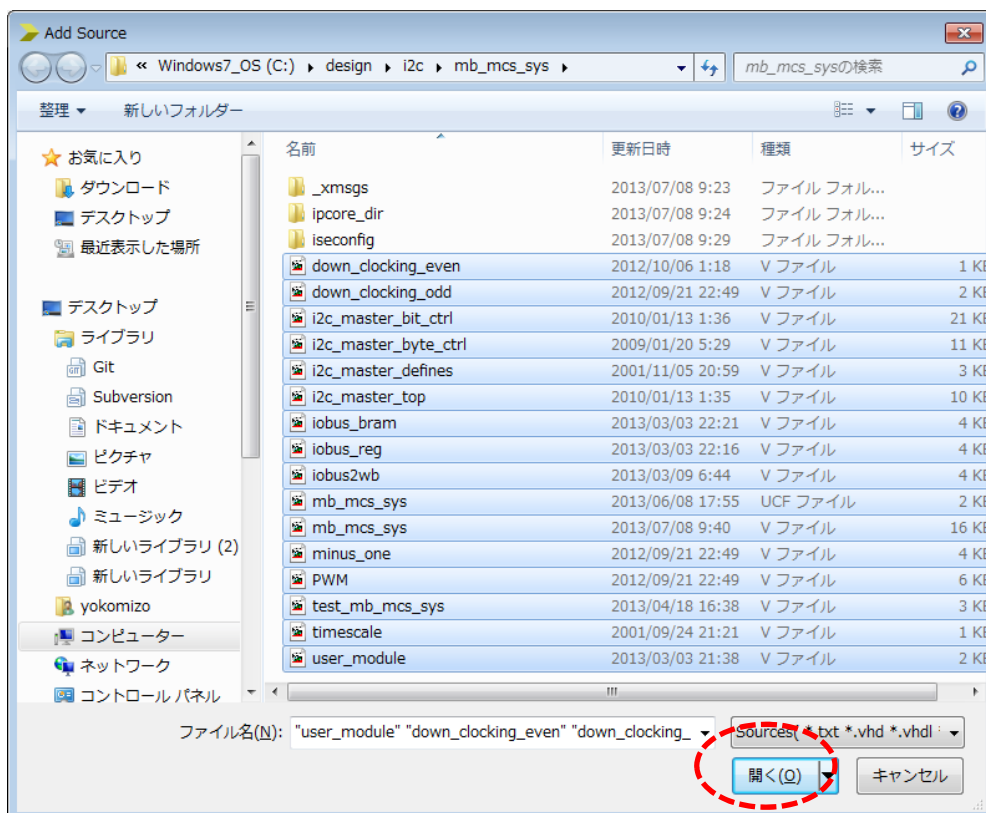
Buttons: More Info, Finish, Cancel

Finish をクリック



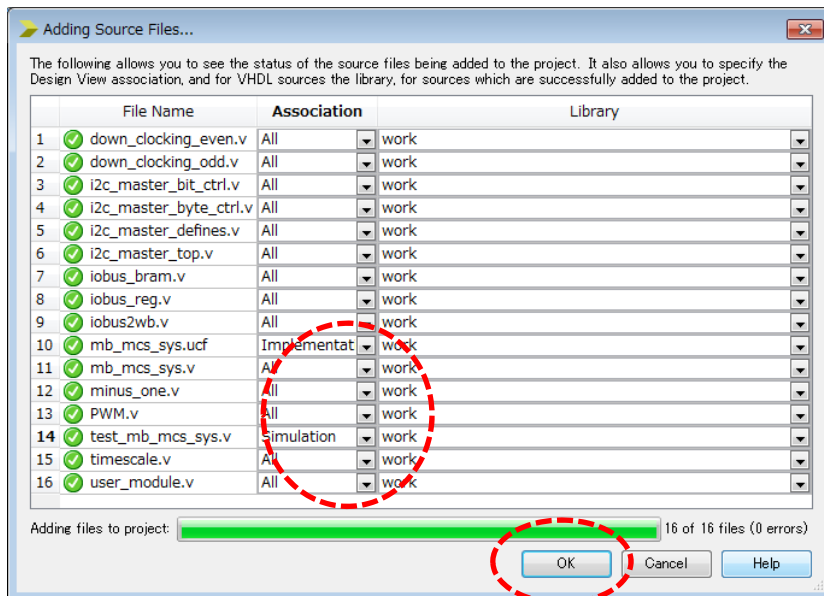


設計データを追加します。Project→Add Source を選択

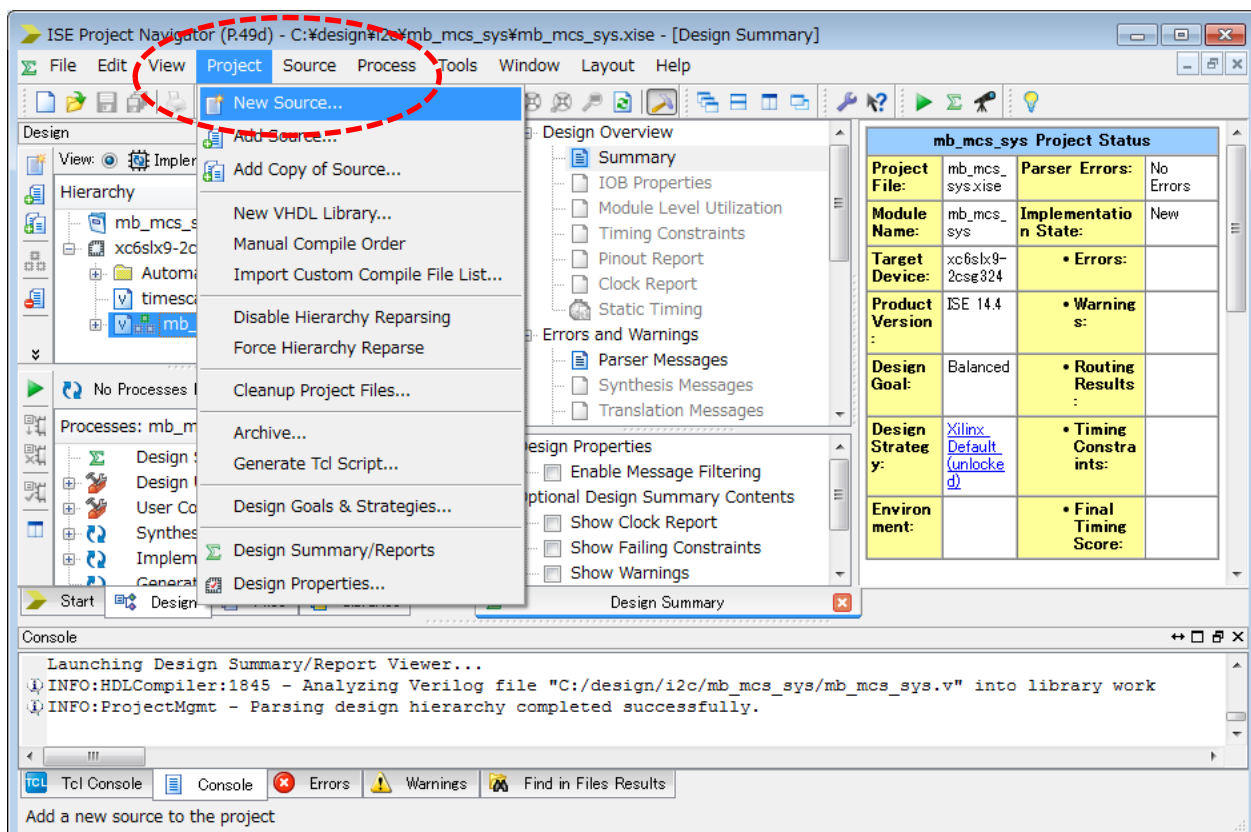


ファイルの指定、mb_mcs_sys の下にある Verilog-HDL ファイルと UCF ファイルを指定



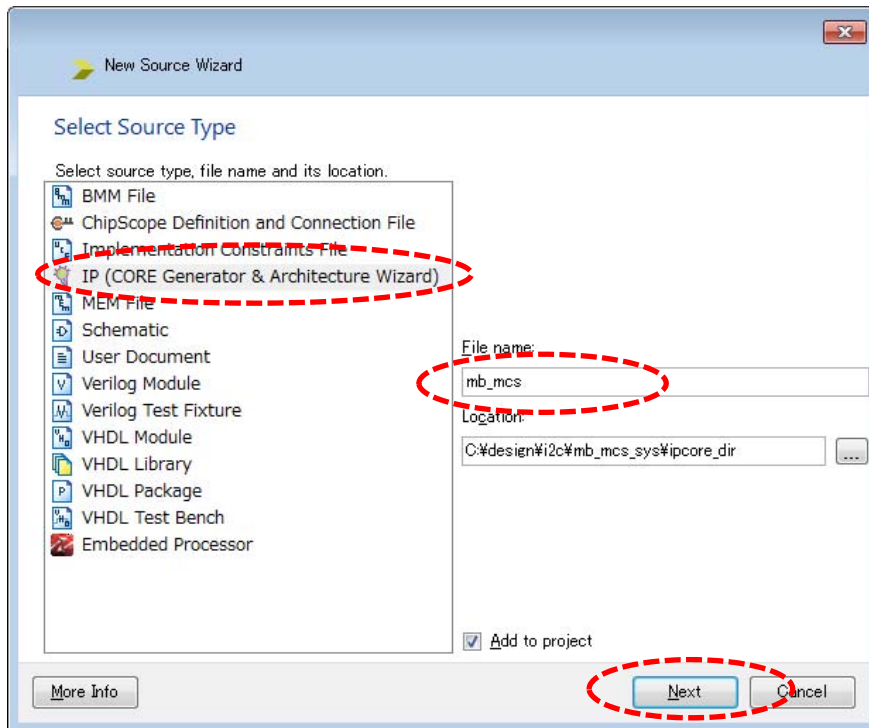


mb_mcs_sys.ucf はインプリメンテーションで使用するので Implementation を指定
 test_mb_mcs_sys.v はテストベンチなので Simulation を指定

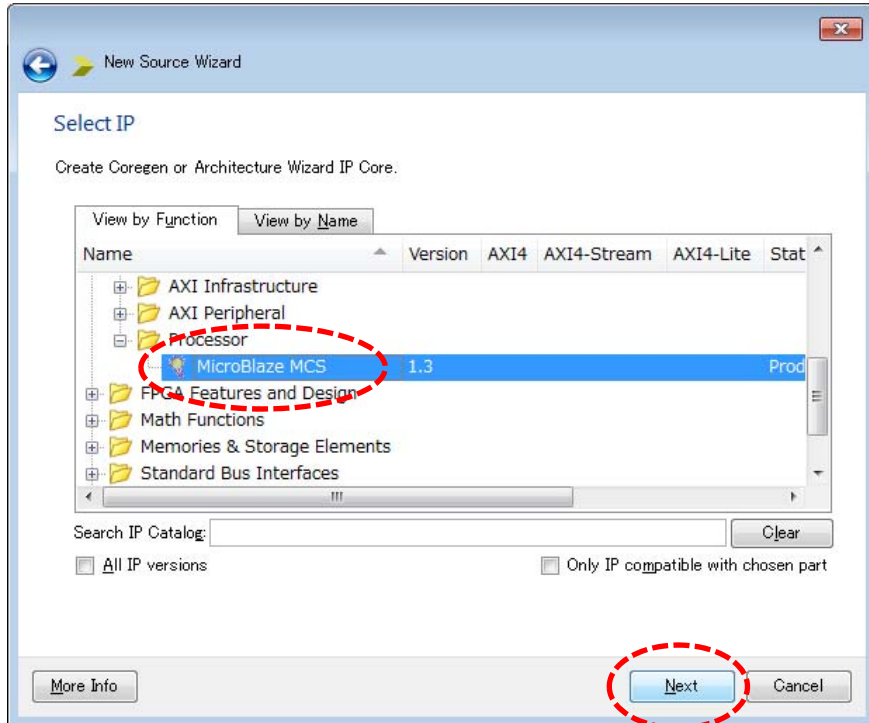


ProjectNavigator で MicroBlaze MCS を追加する。Project→New Source を選択



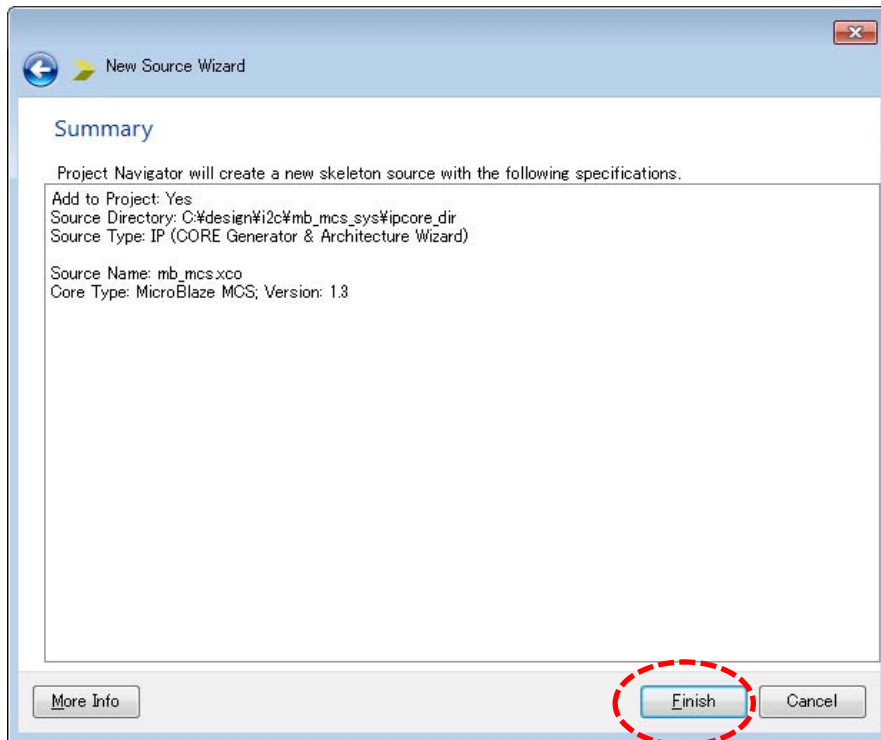


IP (CORE Gener…をクリックして選択、ファイル名に mb_mcs を指定、Next をクリック

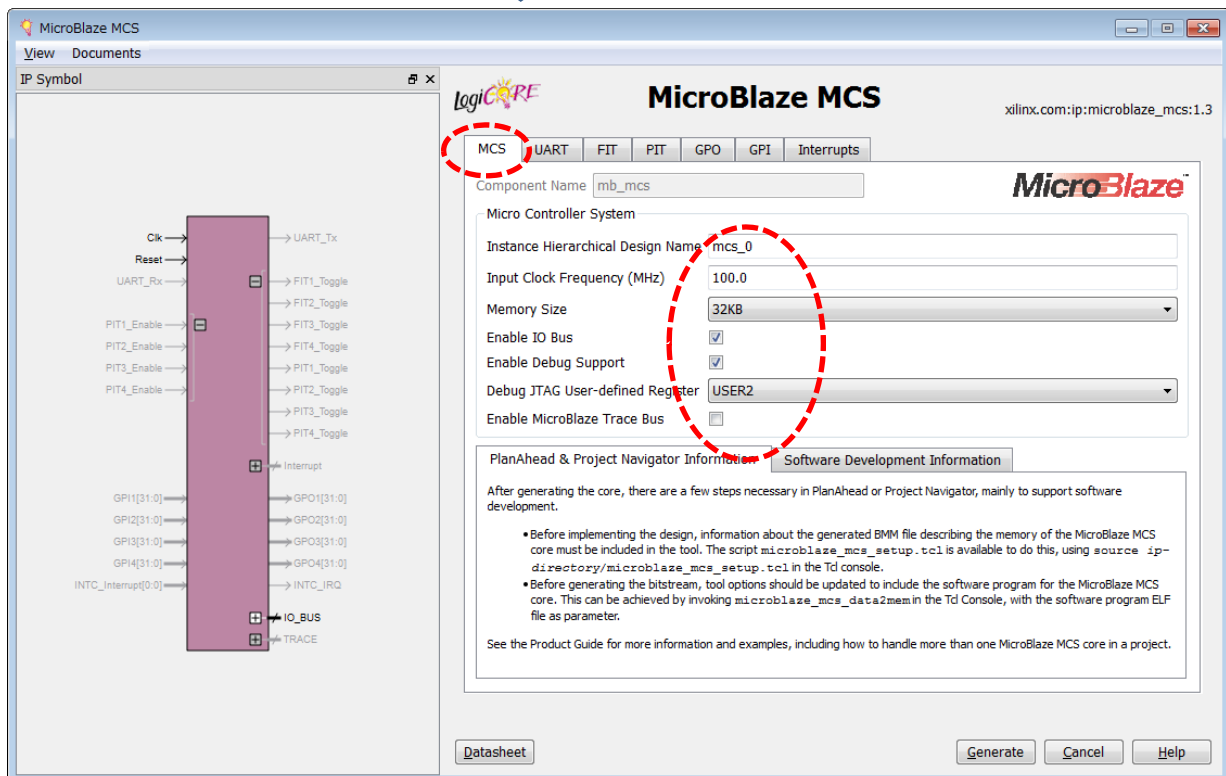


MicroBlaze MCS 選択して Next をクリック



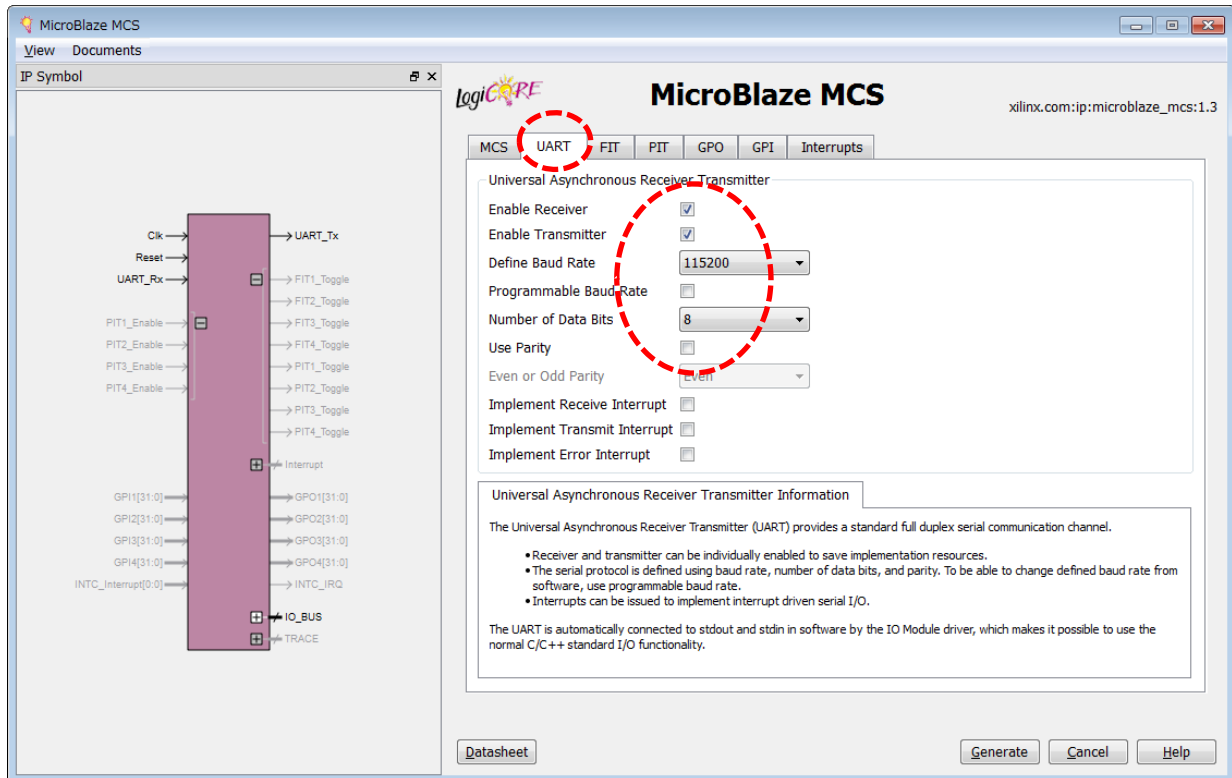


FinishでCORE generator 起動

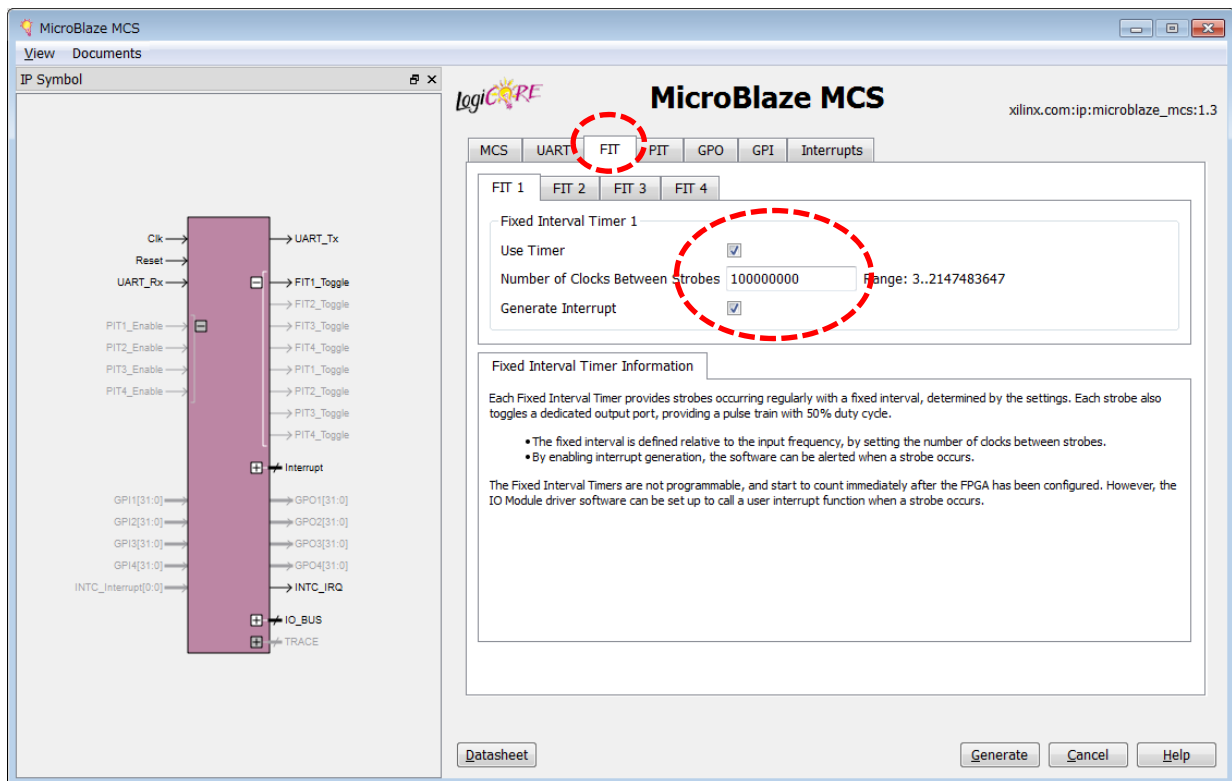


MicroBlaze MCS の基本設定



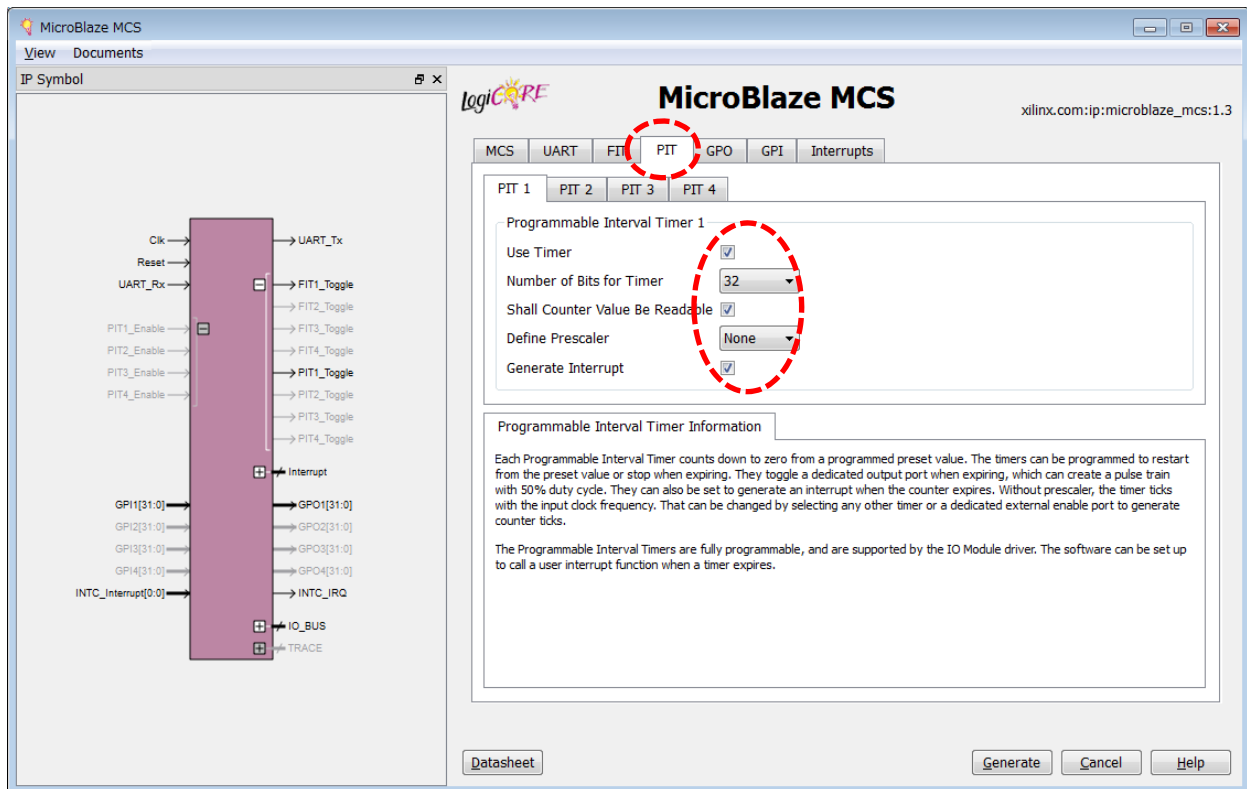


UART の設定

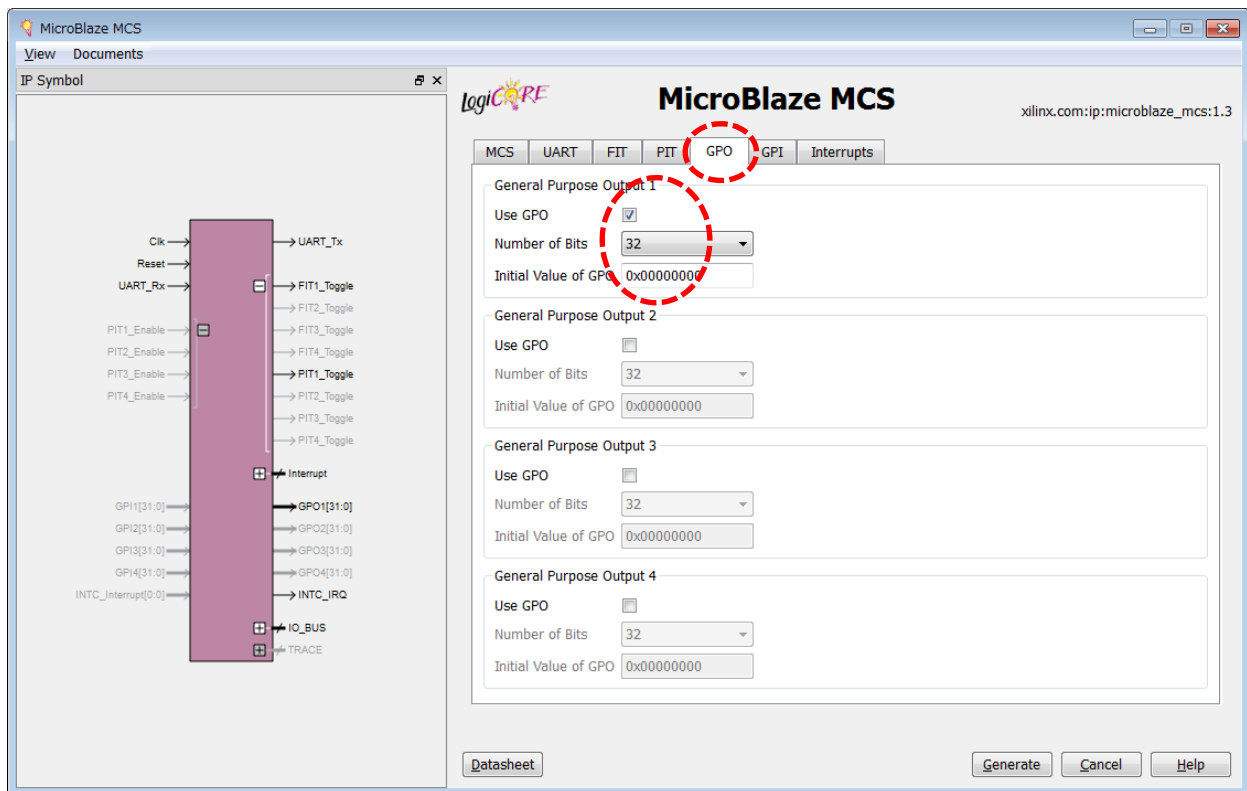


FIT の指定



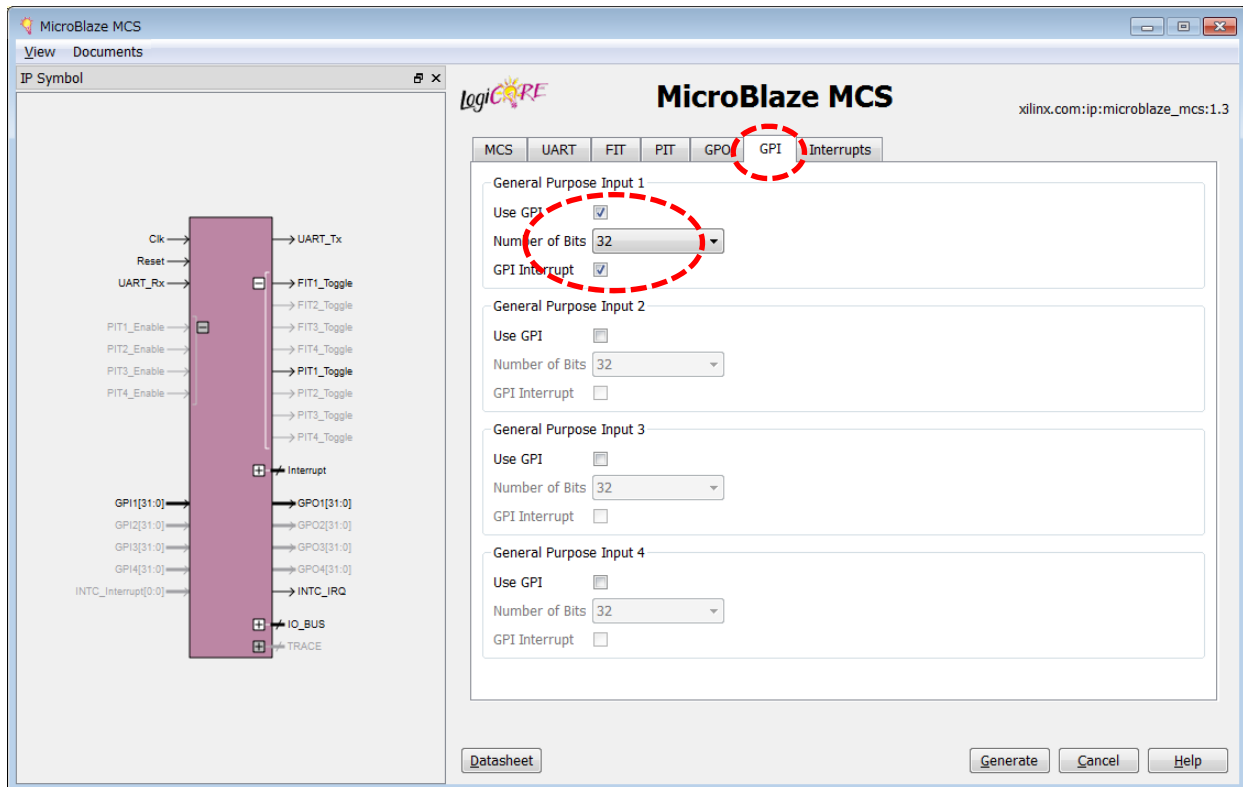


PIT の指定

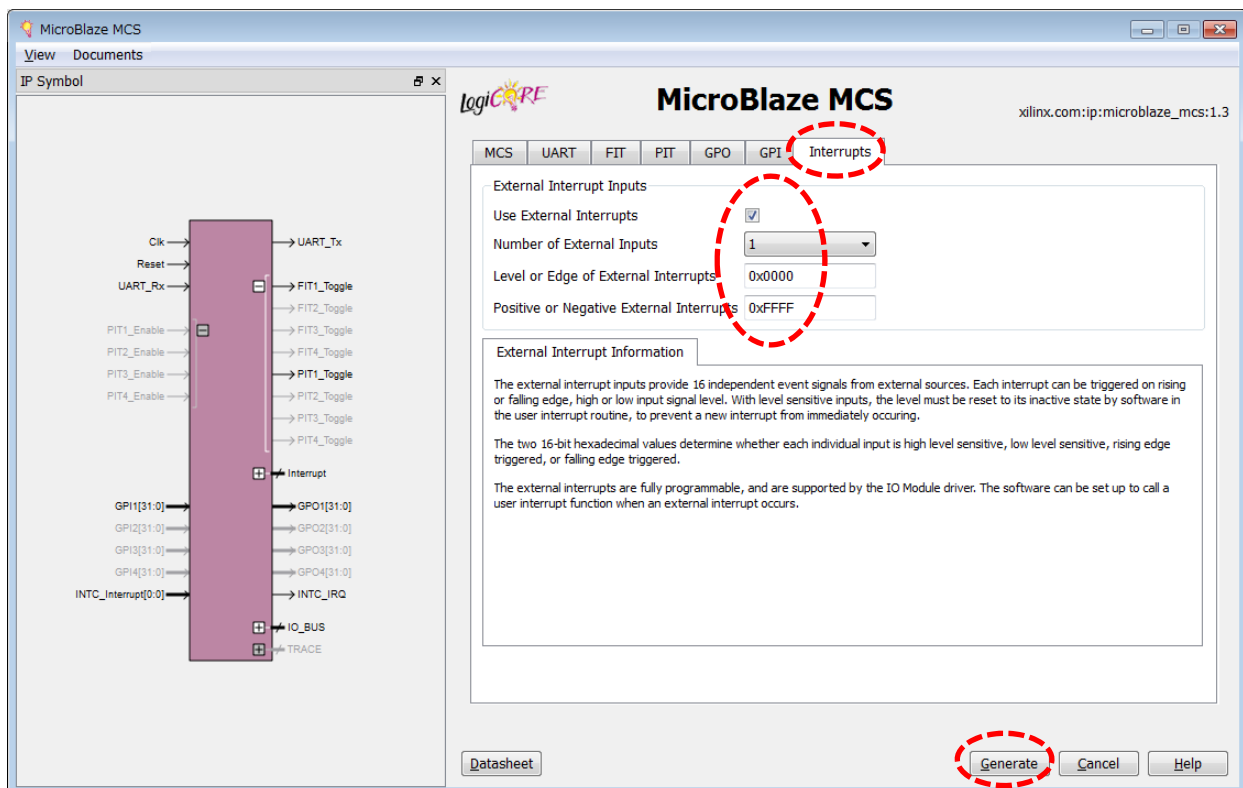


GPO の設定



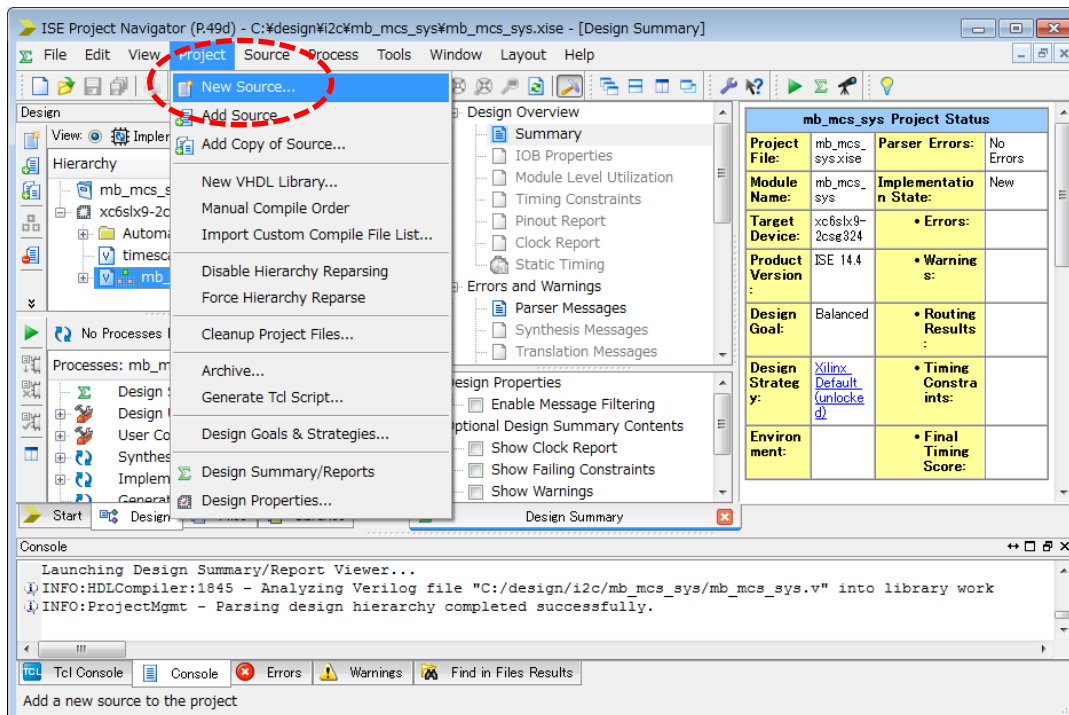


GPI の指定

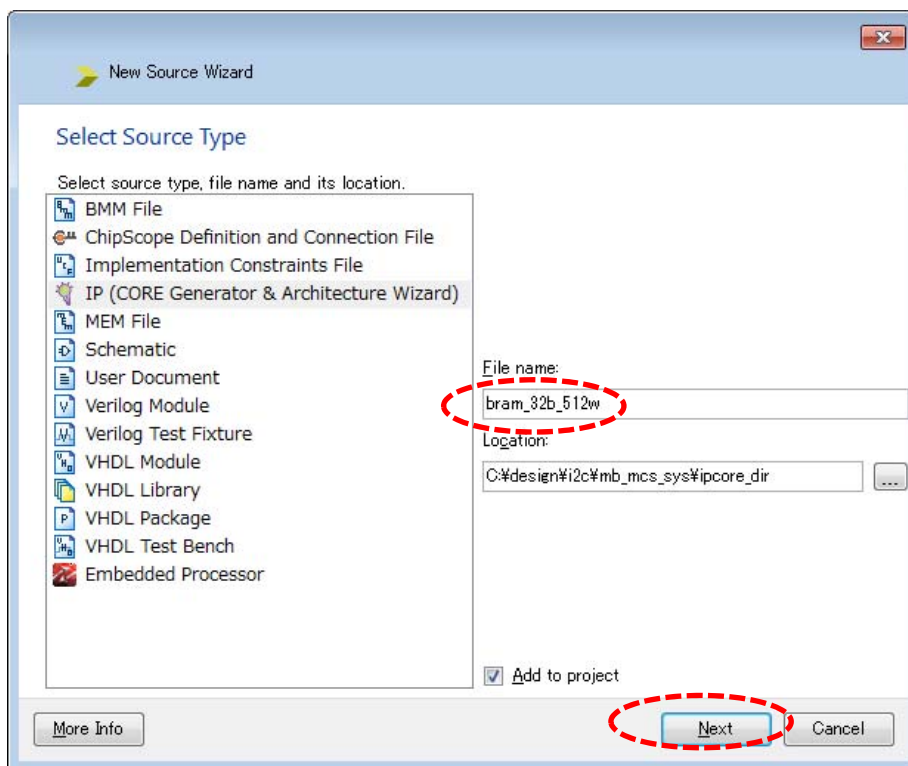


外部割り込みの設定後に Generate をクリックすると MicroBlaze MCS がプロジェクトに追加される



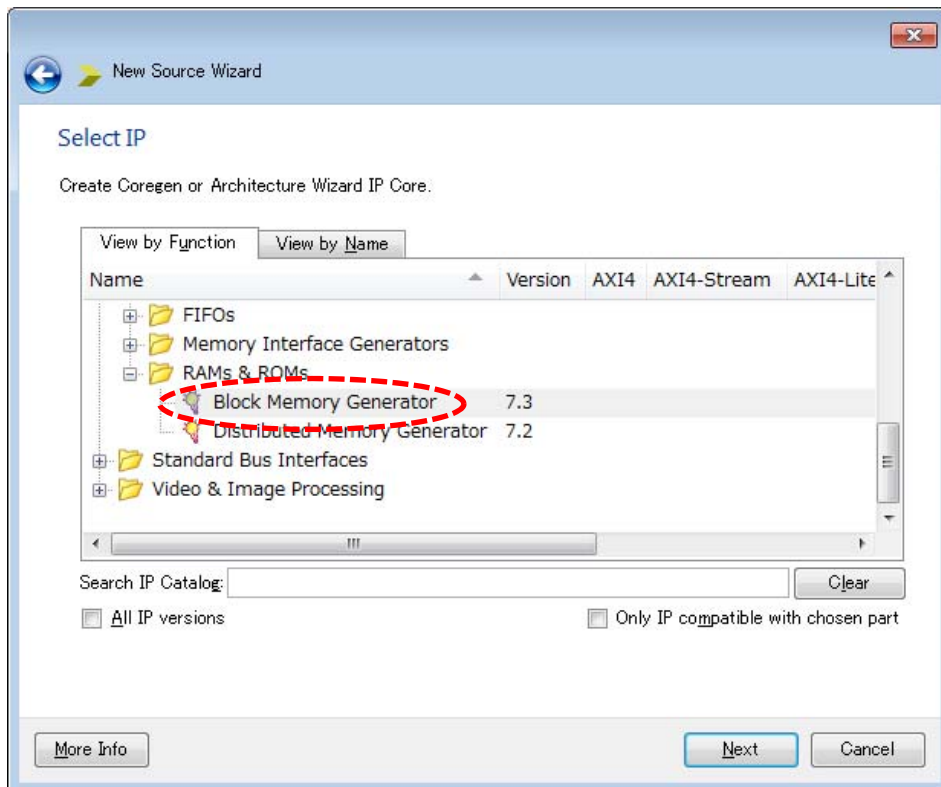


ブロック RAM の作成、Project→New Source を選択

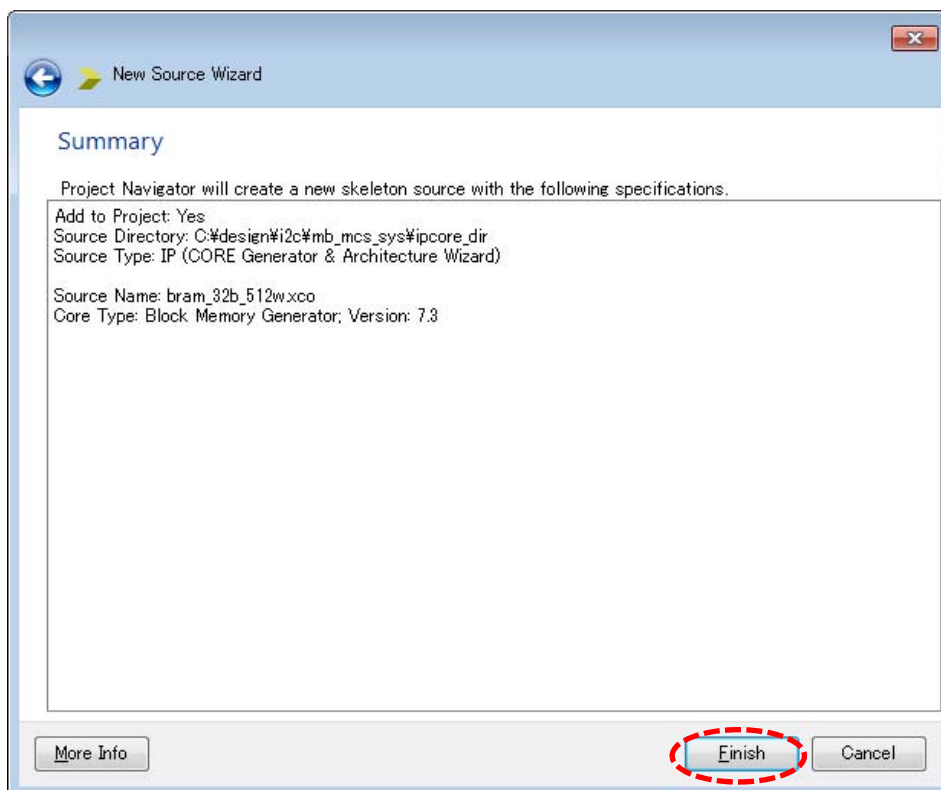


IP (CORE Gener...をクリックして選択、ファイル名に bram_32b_512w を指定、Next をクリック



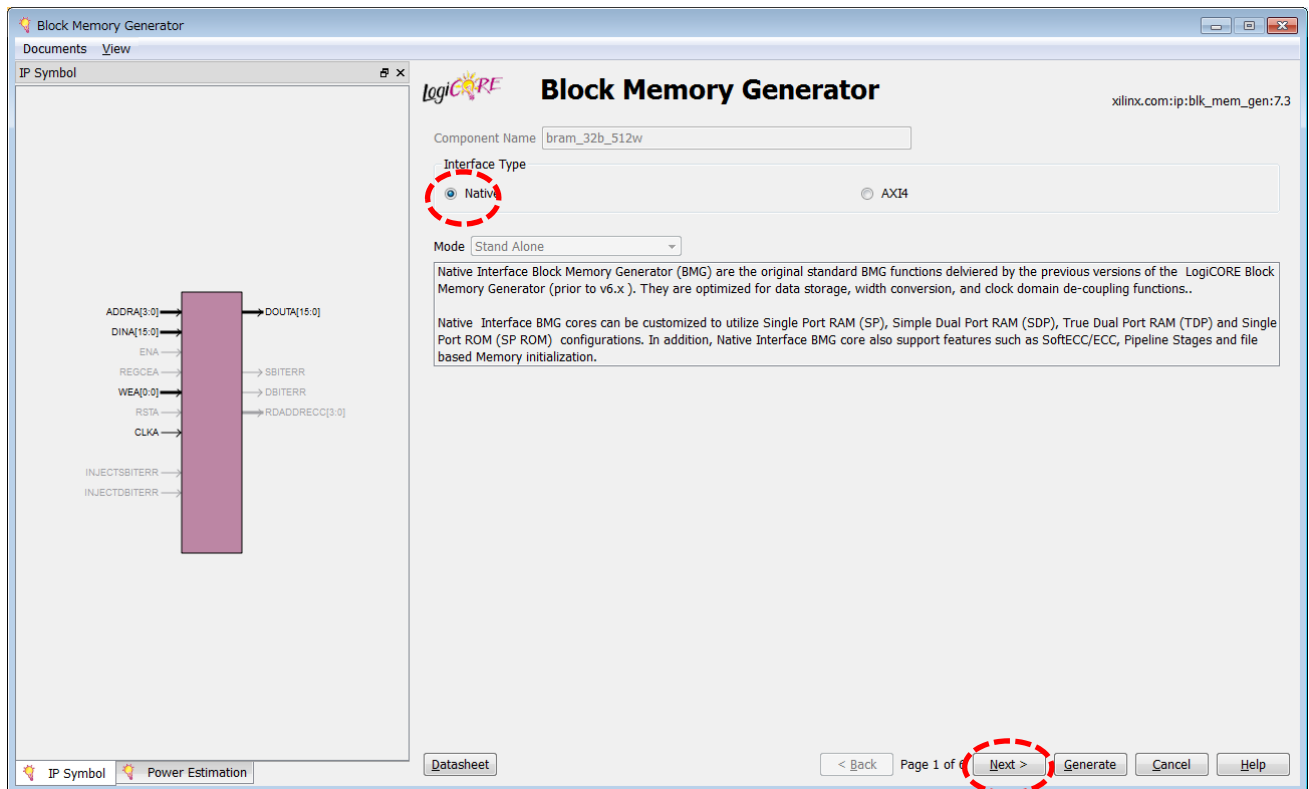


Block Memory Generator を指定

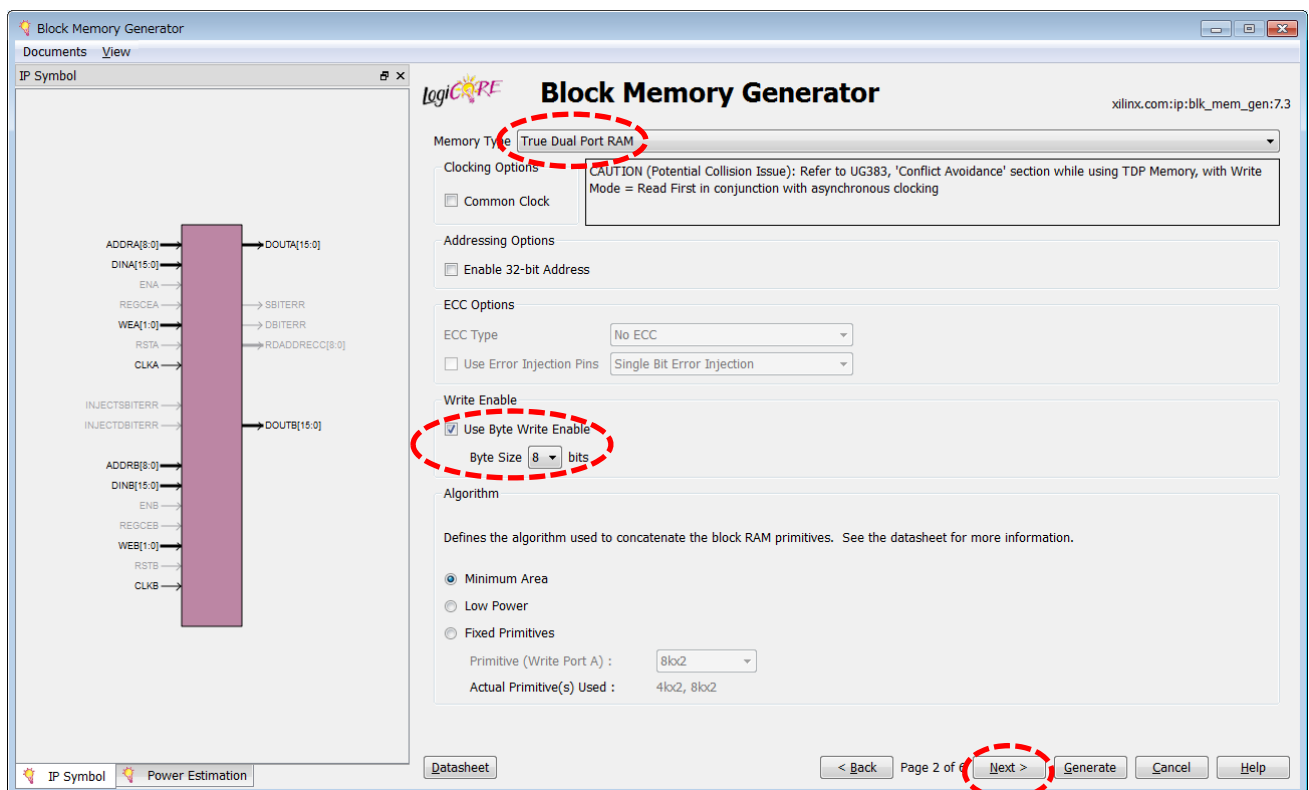


Finish をクリックすると CORE generator が起動



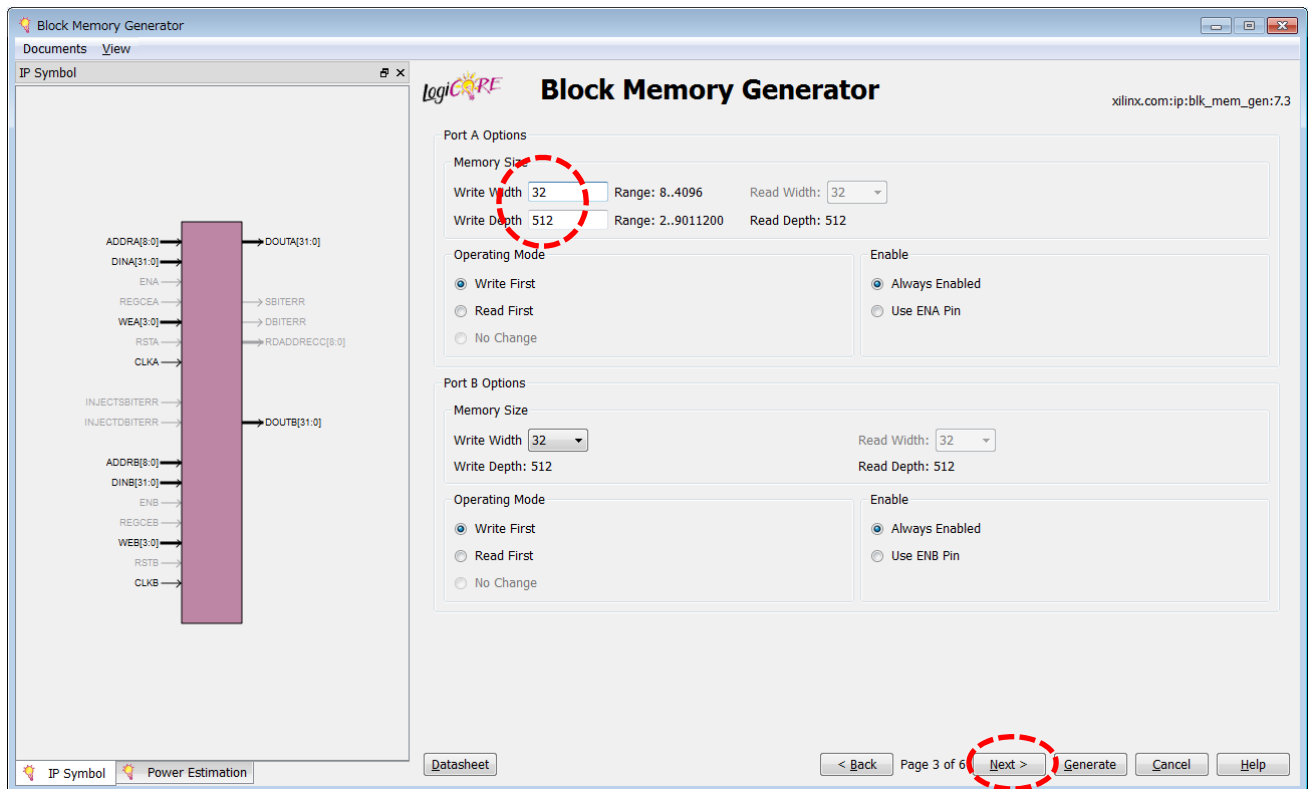


ブロック RAM インターフェースの設定

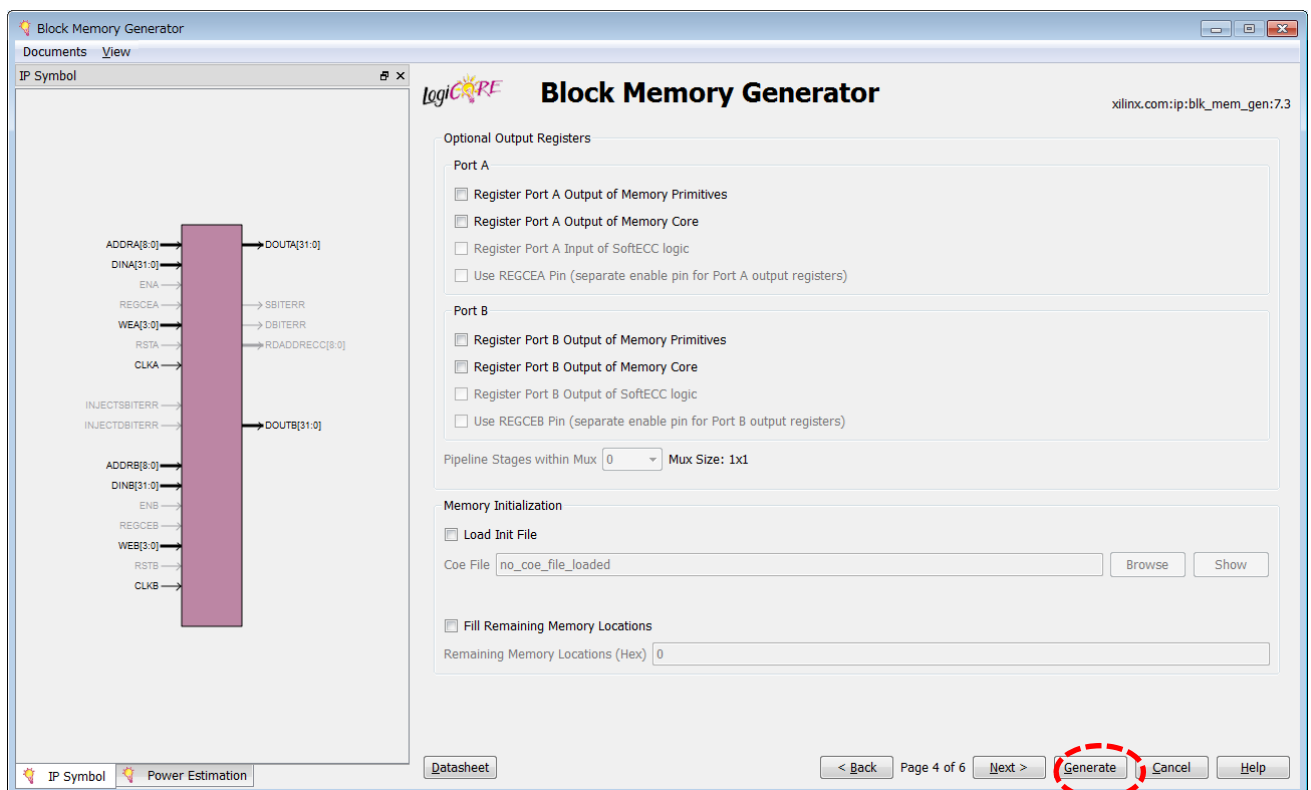


メモリのタイプ指定



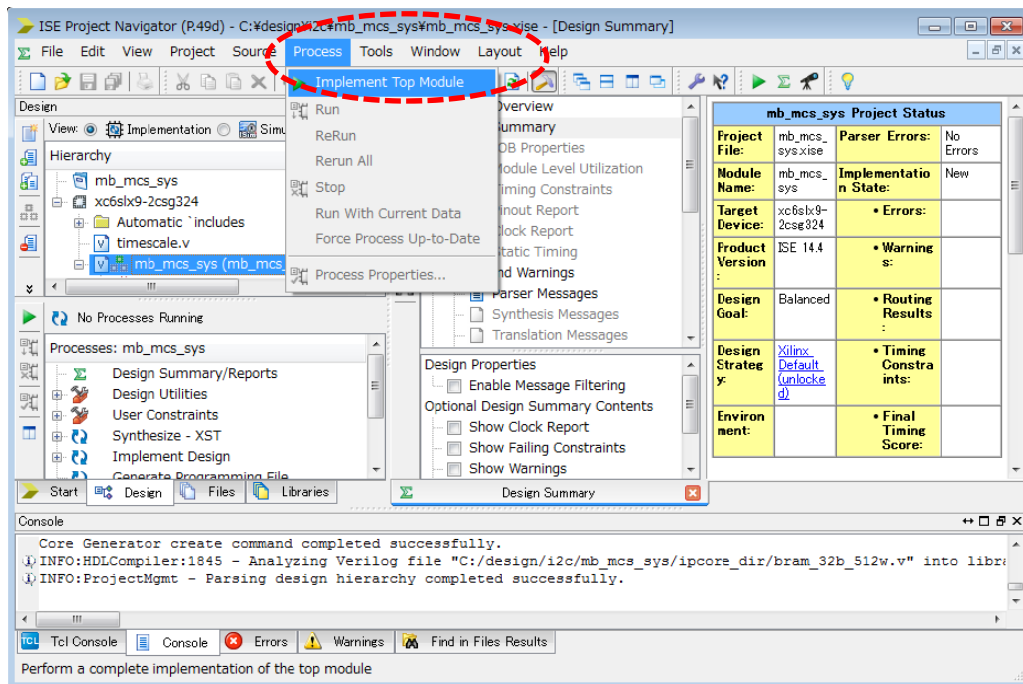


データ幅、データ量の指定



Generate をクリックで作成開始





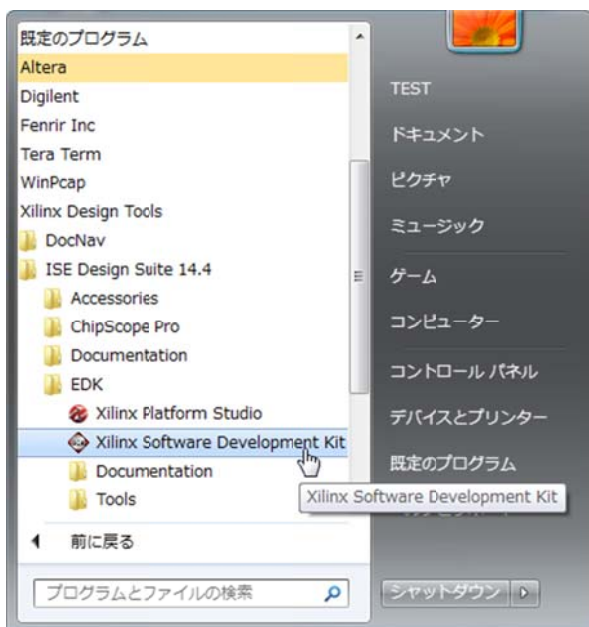
インプリメンテーションの実行、Process→Implement Top Module をクリック

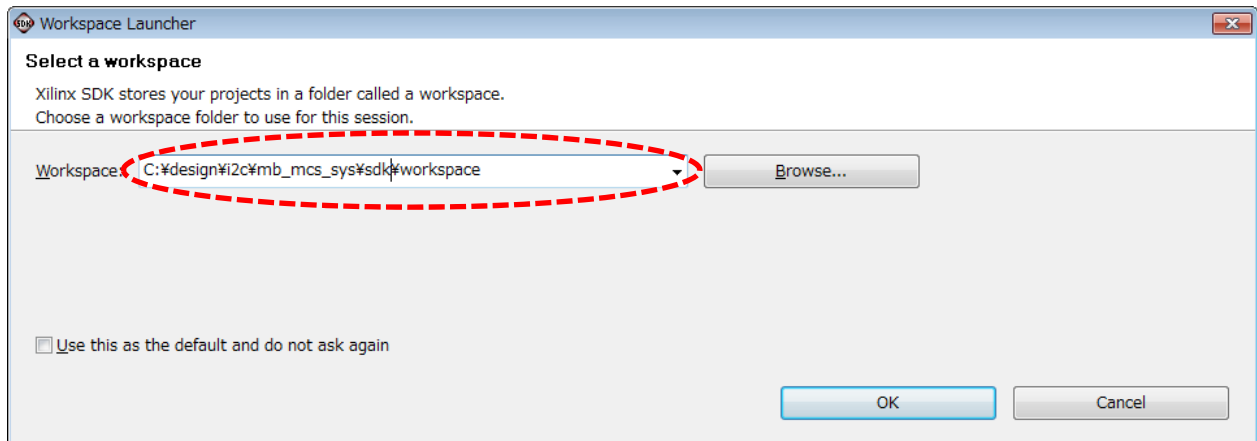


次に Xilinx Software Development Kit (以降 SDK) でソフトウェアを作ります。

・ SDK の起動

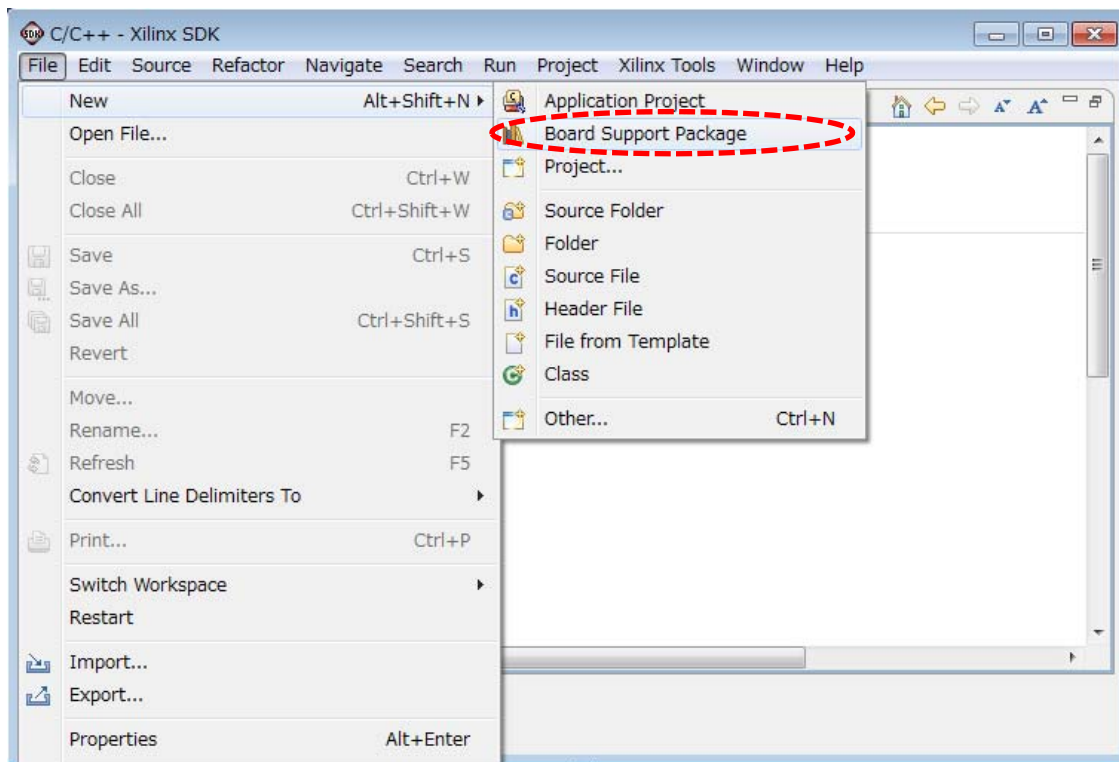
スタートメニューから「Xilinx Design Tools」→「ISE Design Suite 14.4」→「EDK」→「Xilinx Software Development Kit」を起動してください。





SDK を起動するとワークスペースを指定が要求されます。設計フォルダ/sdk/workspace を設定



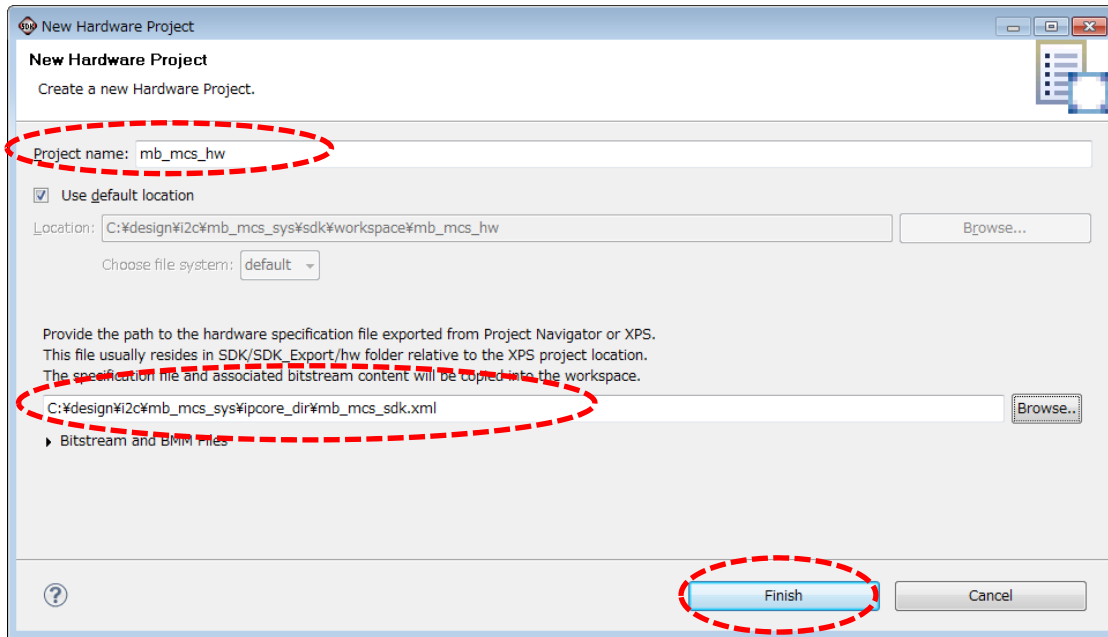


新規のボードサポートパッケージ作成、File→New→Board Support Package



Hardware Platform を定義するか聞かれるので、specify を選択して定義する

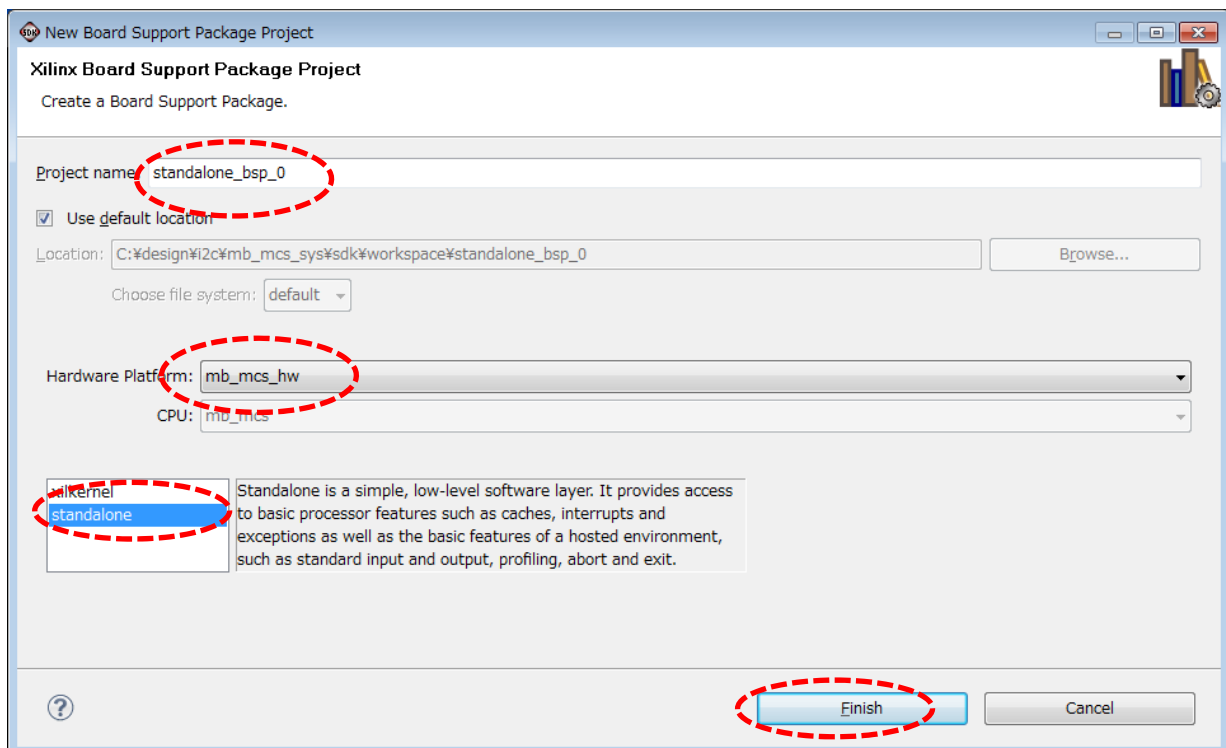




SDK の管理するハードウェアプロジェクト名指定

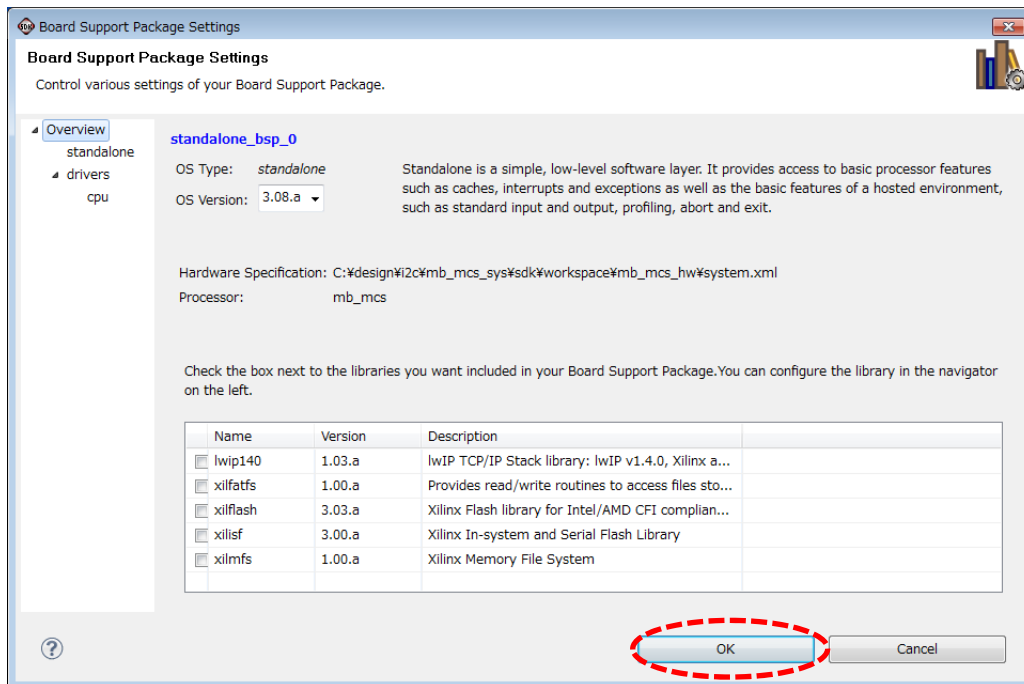
xml ファイル (設計フォルダ/ipcore_dir/mb_msc. sdk. xml) 指定

Finish をクリック

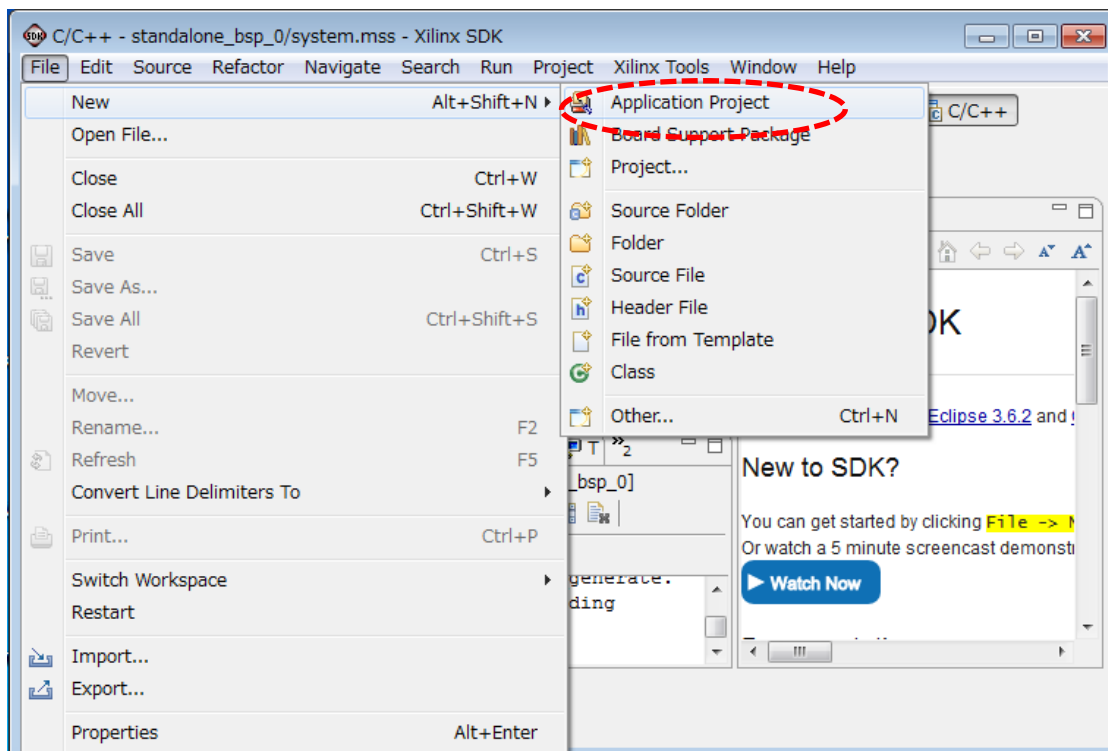


ボードサポートパッケージの定義



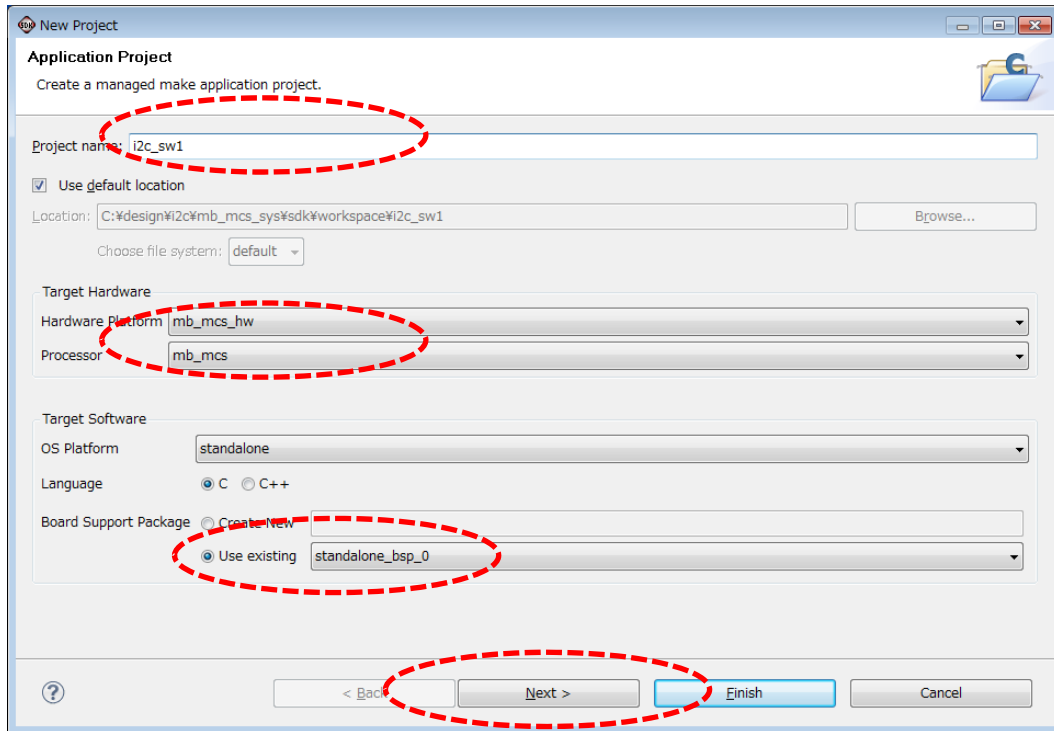


ボードサポートパッケージのオプション定義



新規のソフトウェアプロジェクトの作成





The 'New Project' dialog box is shown with the 'Application Project' tab selected. The 'Project name' field is set to 'i2c_sw1'. The 'Use default location' checkbox is checked. The 'Location' field shows a path starting with 'C:\design\i2c\mb_mcs_sys\sdk\workspace\i2c_sw1'. The 'Choose file system' dropdown is set to 'default'. Under 'Target Hardware', 'Hardware Platform' is 'mb_mcs_hw' and 'Processor' is 'mb_mcs'. Under 'Target Software', 'OS Platform' is 'standalone', 'Language' is 'C', and 'Board Support Package' is 'standalone_bsp_0'. The 'Finish' button is highlighted.

New Project
Create a managed make application project.

Project name: i2c_sw1

☒ Use default location
Location: C:\design\i2c\mb_mcs_sys\sdk\workspace\i2c_sw1
Choose file system: default

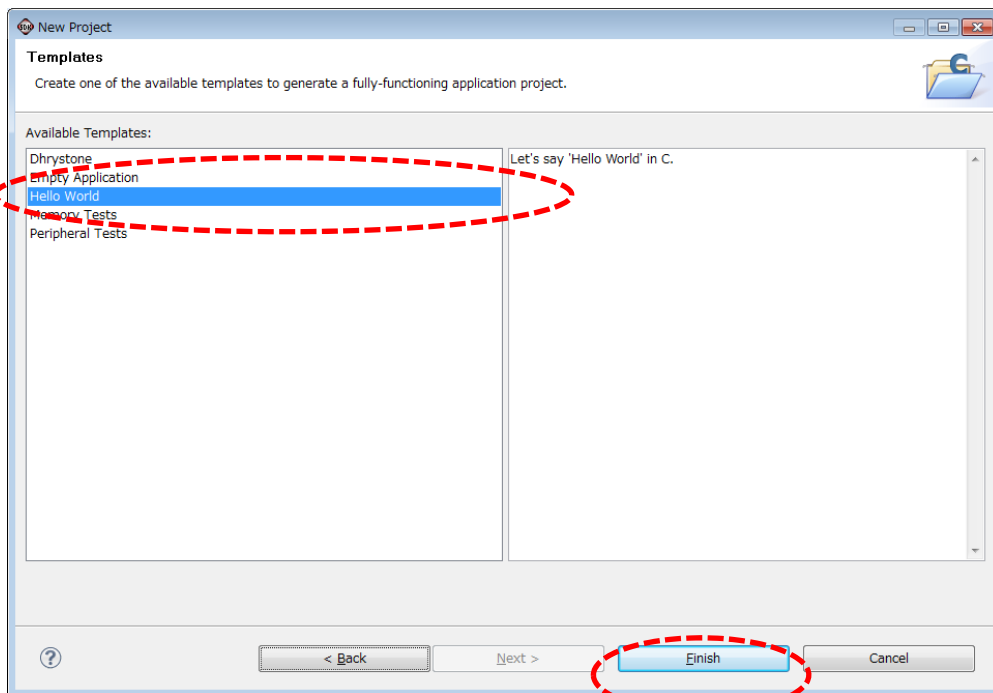
Target Hardware
Hardware Platform: mb_mcs_hw
Processor: mb_mcs

Target Software
OS Platform: standalone
Language: ☒ C ☐ C++
Board Support Package: ☒ Use existing standalone_bsp_0

< Back Next > Finish Cancel

ソフトウェアプロジェクト名指定: i2c_sw1

Next をクリック



The 'New Project' dialog box is shown with the 'Templates' tab selected. The 'Available Templates' list on the left includes 'Dhrystone', 'Empty Application', 'Hello World', 'Memory Tests', and 'Peripheral Tests'. 'Hello World' is selected and highlighted in blue. The 'Finish' button is highlighted.

New Project
Create one of the available templates to generate a fully-functioning application project.

Available Templates:

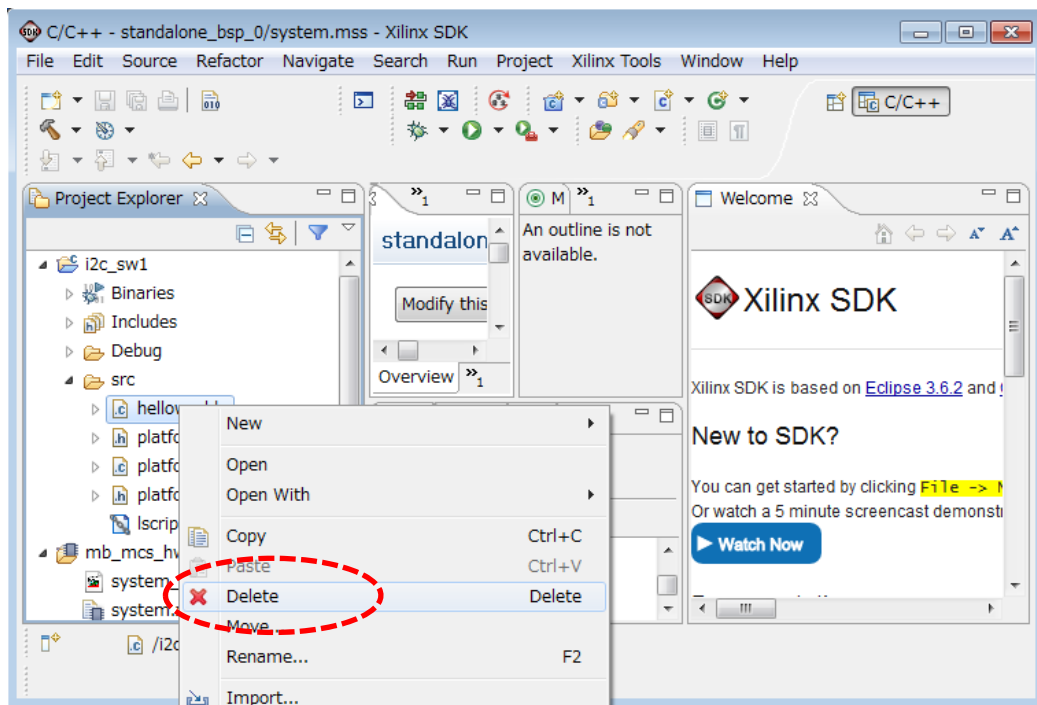
- Dhrystone
- Empty Application
- Hello World
- Memory Tests
- Peripheral Tests

Let's say 'Hello World' in C.

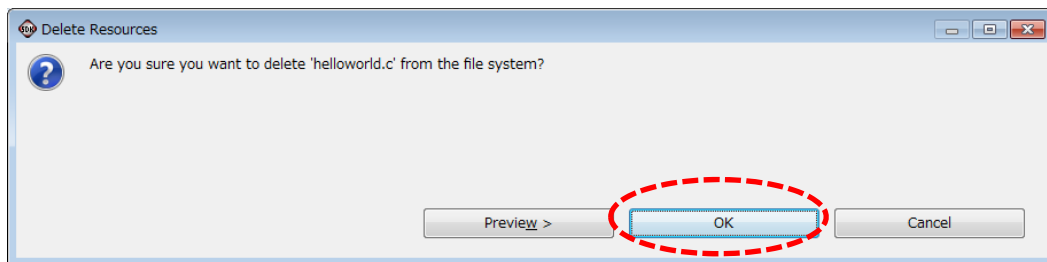
< Back Next > Finish Cancel

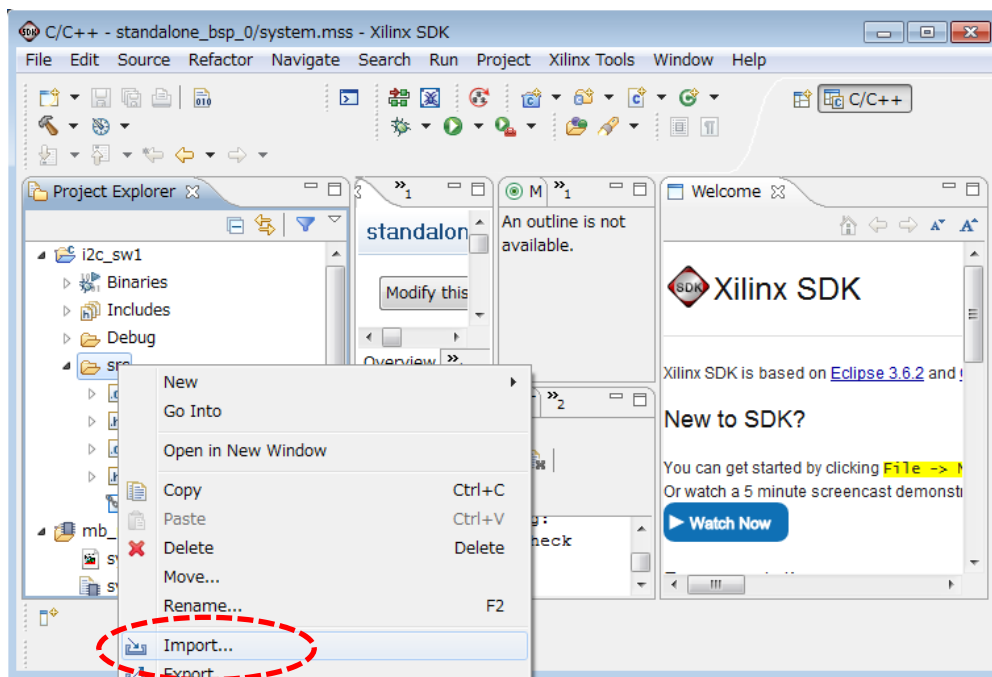
テンプレートに Hello World 選択後、Finish をクリックでソフトウェアプロジェクトが作成される



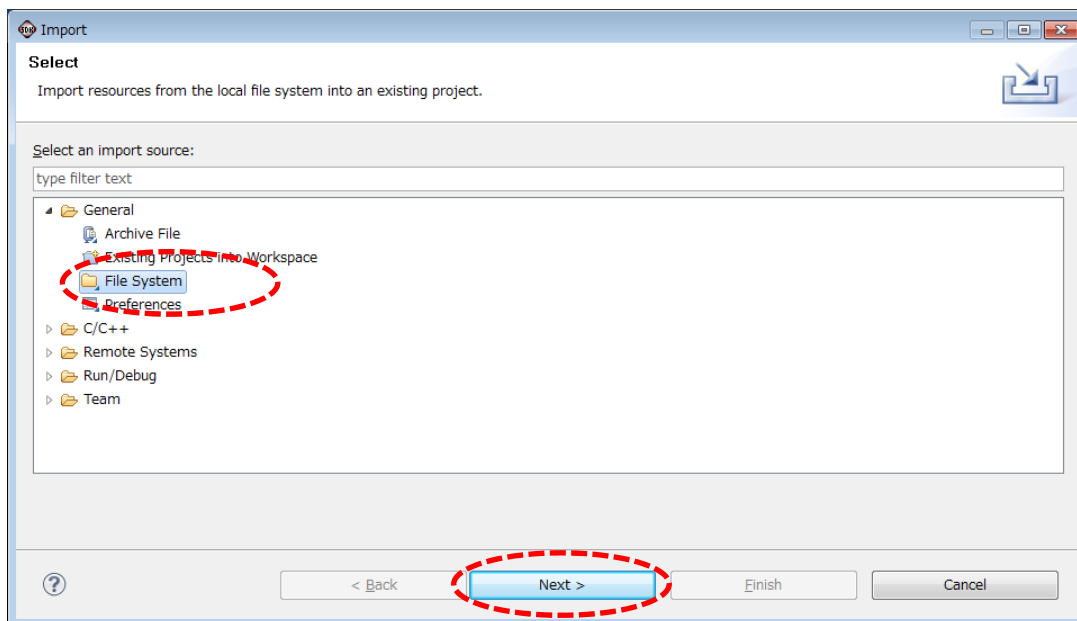


helloworld.c を削除



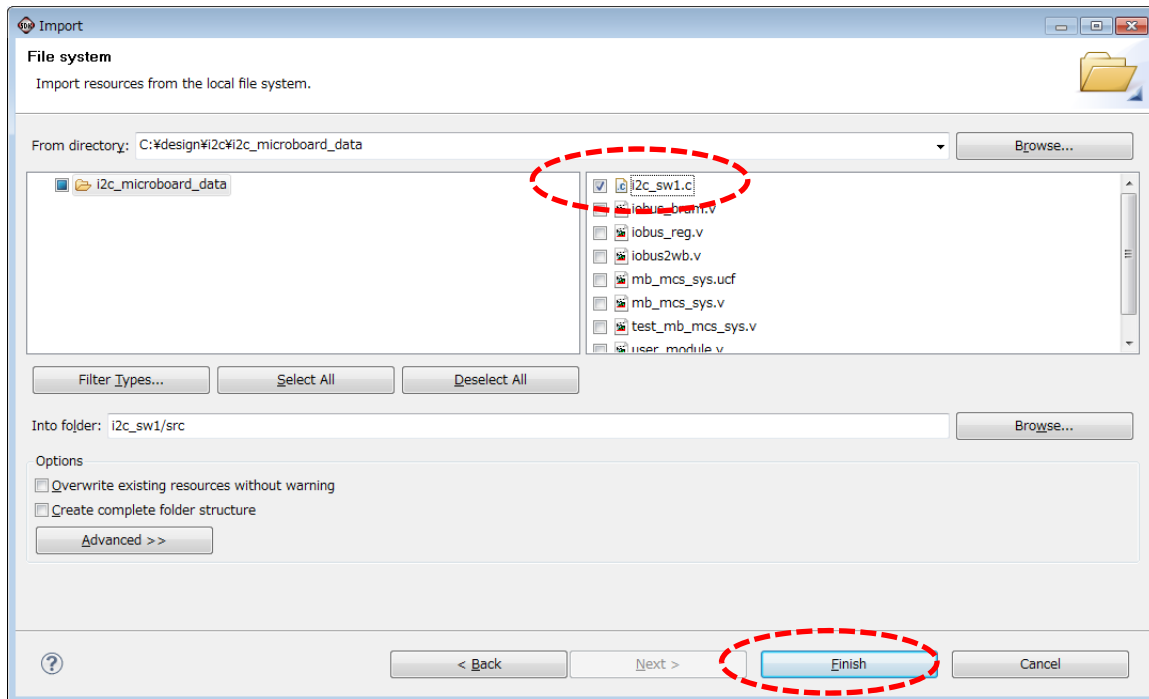


C ソースコードのインポート、i2c_sw1 の src の上で右クリック、import 選択



General→File System 選択



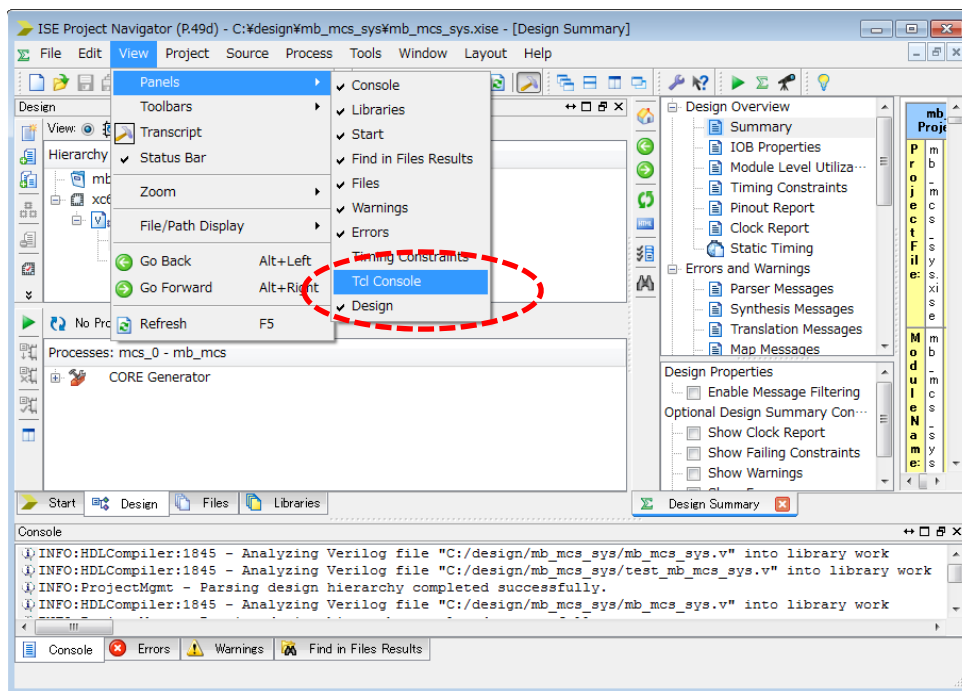


解凍データの i2c_sw1.c を選択

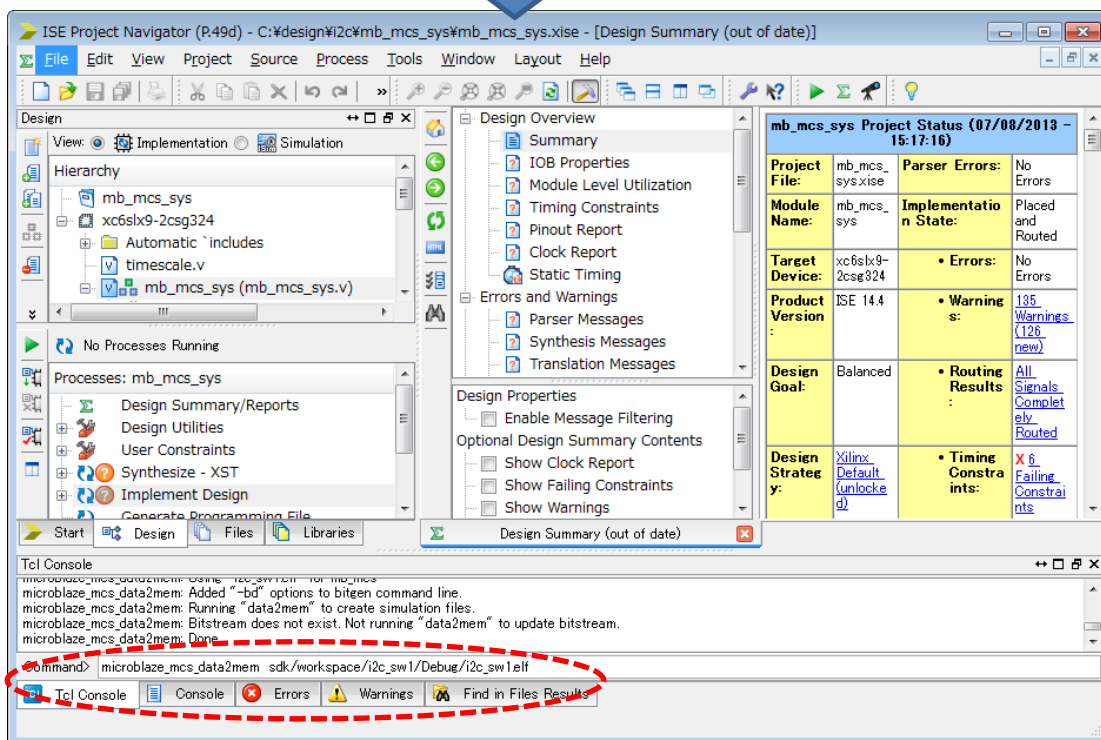
ビルドが実行され、

実行ソフトウェアのファイル（sdk/workspace/i2c_sw1/Debug/i2c_sw1.elf）が作られる。





Project Navigator に戻り、Tcl コマンドを使用できるように Tcl Console を表示する
View→Panels→Tcl Console



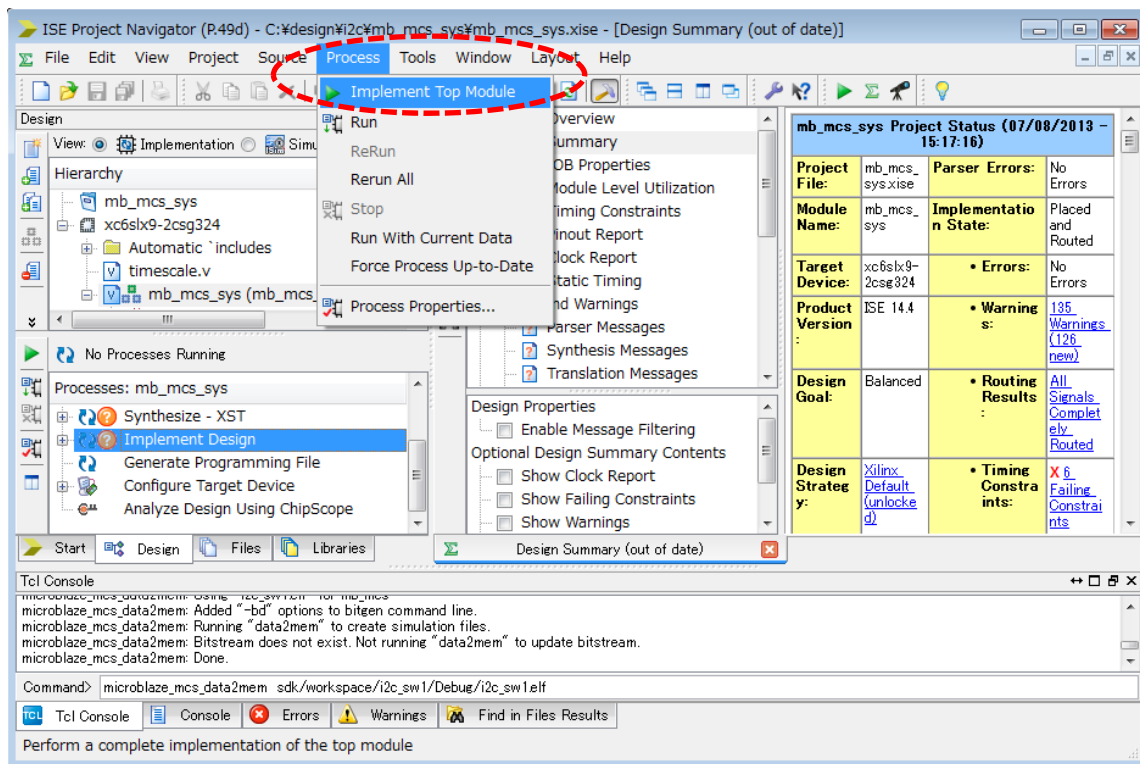
Tcl コマンドを使って実行ソフトウェアのファイルをMicroBlazeMCSのメモリの初期値定義ファイルに変換する。

Tcl コマンド

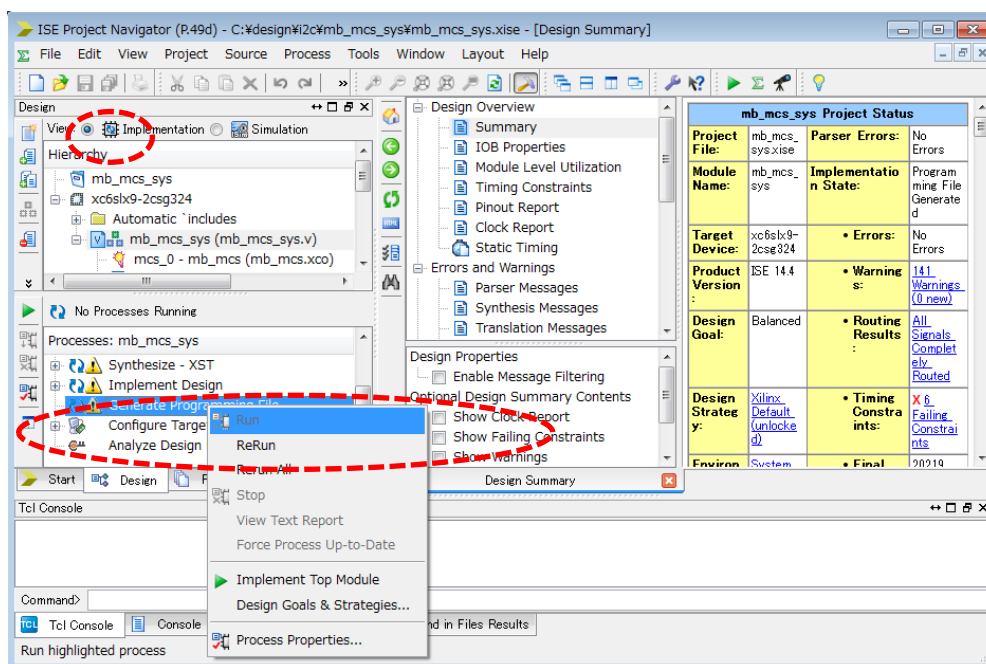
source ipcore_dir/microblaze_mcs_setup.tcl[Enter]

microblaze_mcs_data2mem sdk/workspace/i2c_sw1/Debug/i2c_sw1.elf[Enter]





インプリメンテーションの実施、Process→Implement Top Module



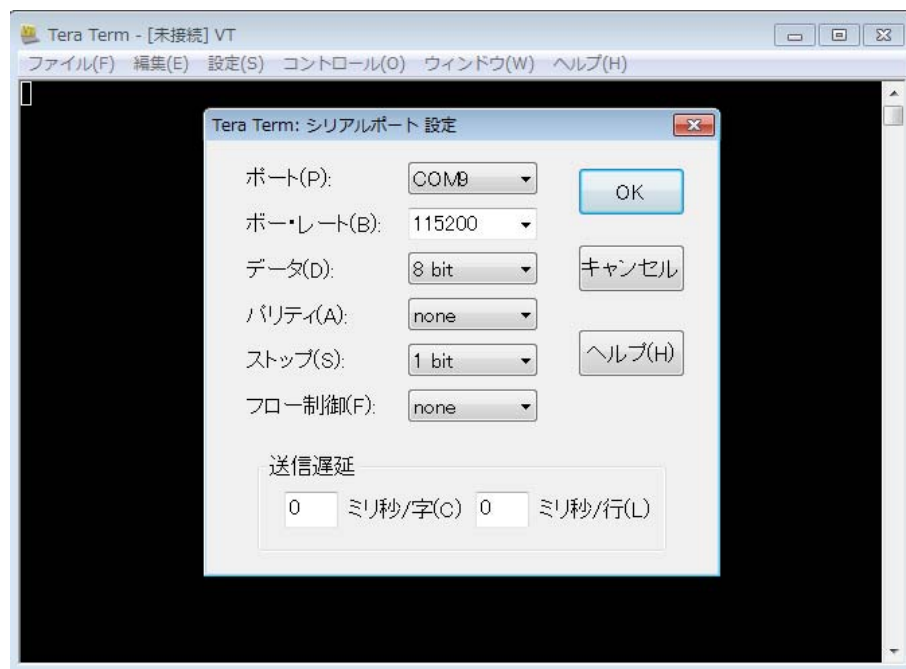
FPGA へ書き込む mb_mcs_sys.bit 作成



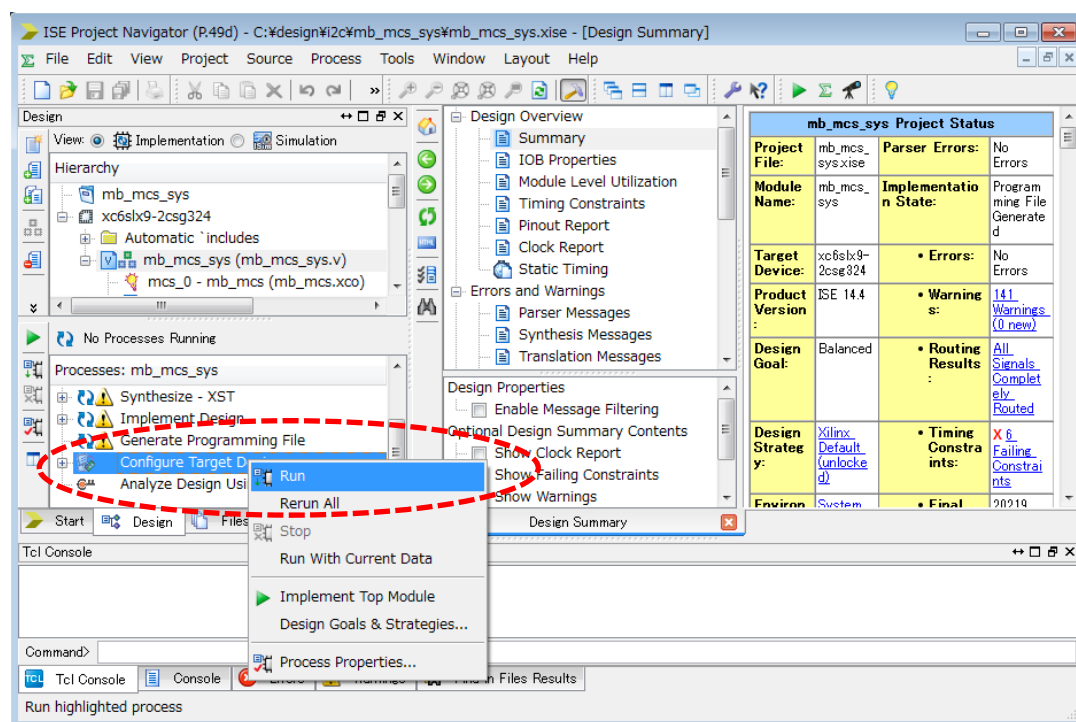
実機の動作確認をします。

LX9 マイクロボードのプログラミング用 USB と UART 用 USB をそれぞれ PC の USB ポートに接続で接続します。

- ・FPGA マガジンの記事と同様に I²C デバイスと MicroBoard を接続する

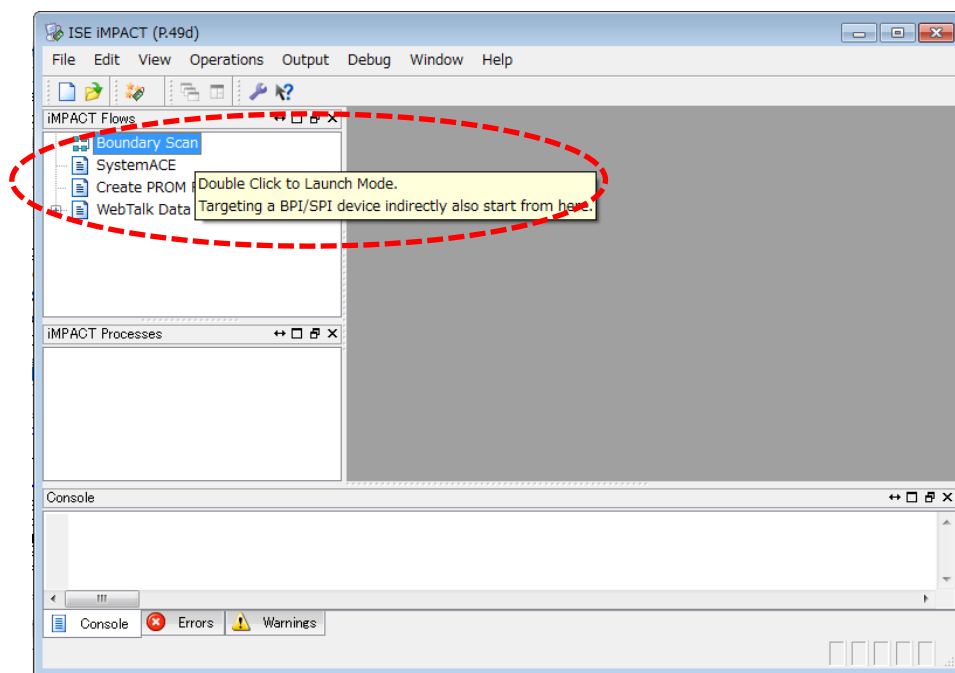


ターミナルソフトを立ち上げる。ポート番号は自分 PC で割り当てられて番号を使う、

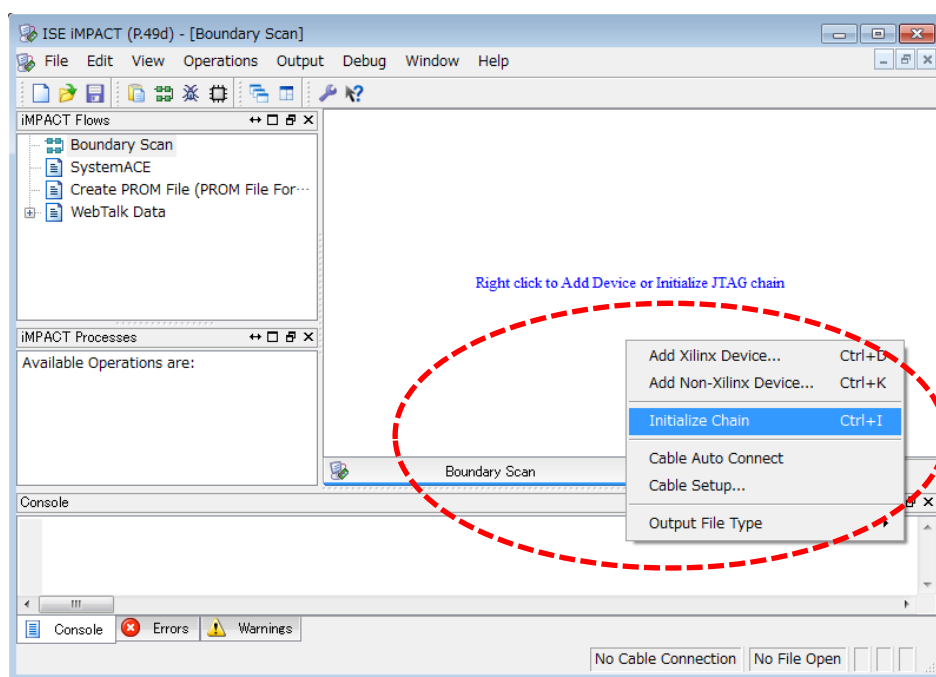


projectNavigator で iMPACT を起動



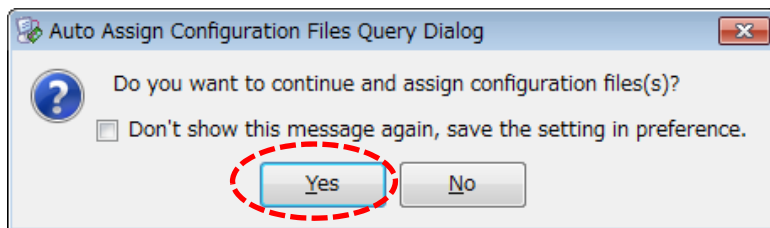


BoundaryScan モードにする

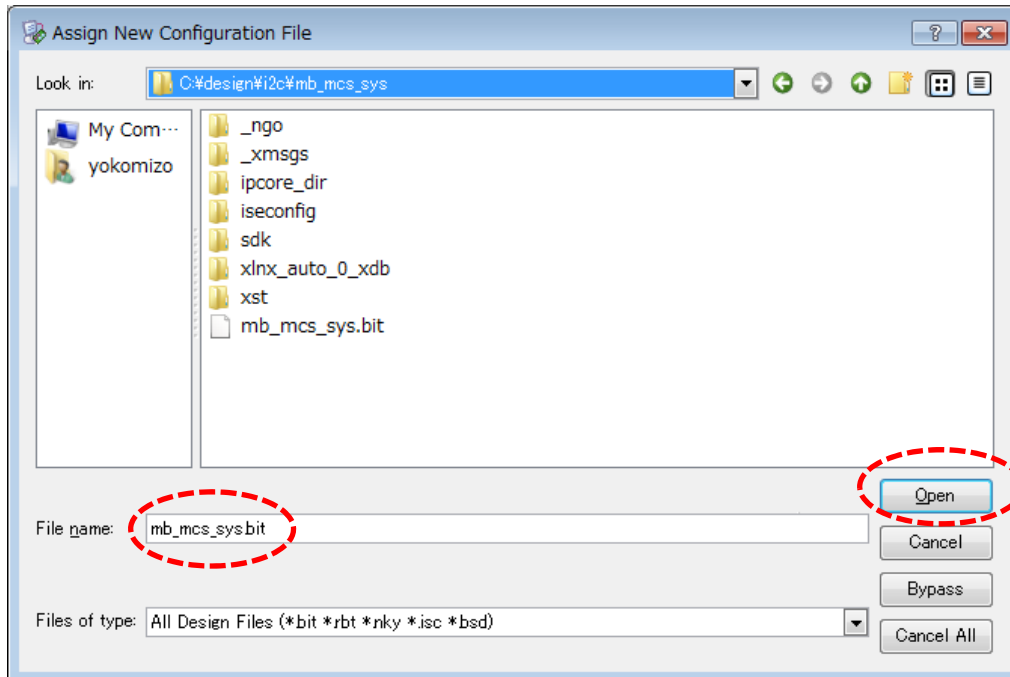


BoundaryScan のウィンドをマウス右ボタン押して、Initialize Chain 選択して FPAG を検出する

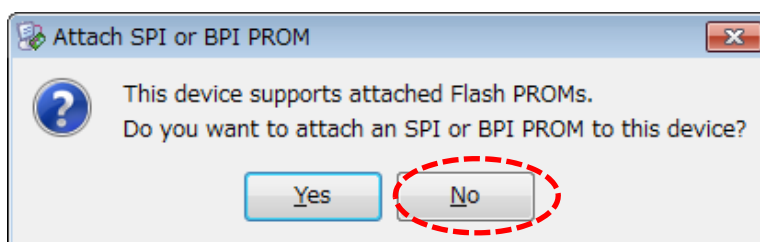




Yes で FPGA に書き込むファイルを指定する

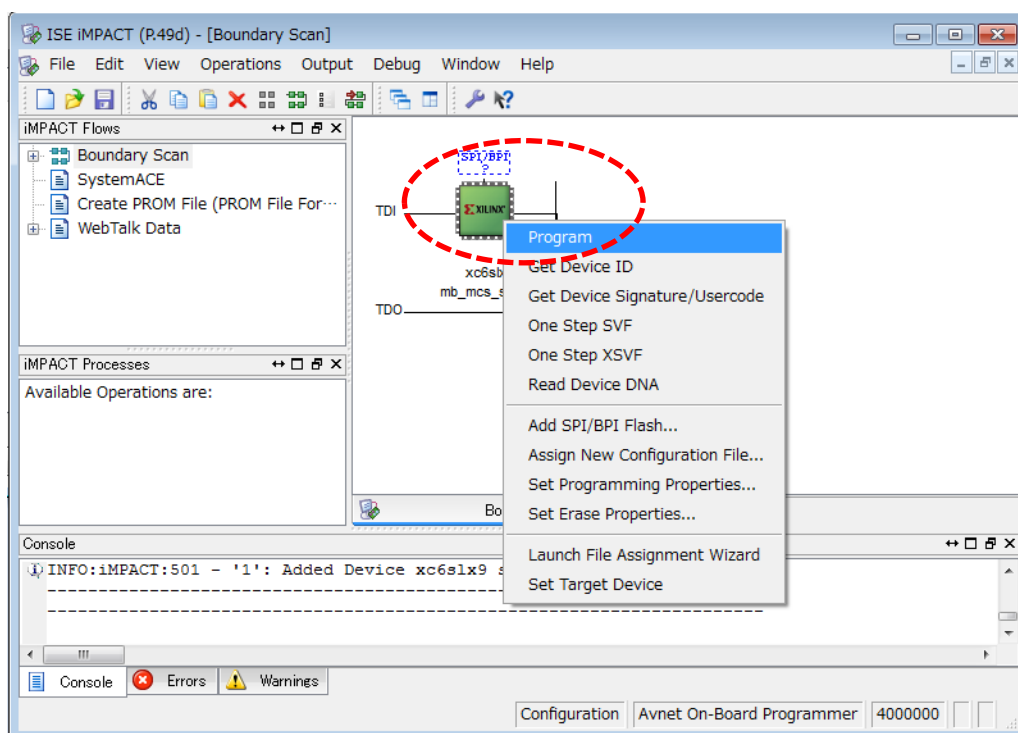
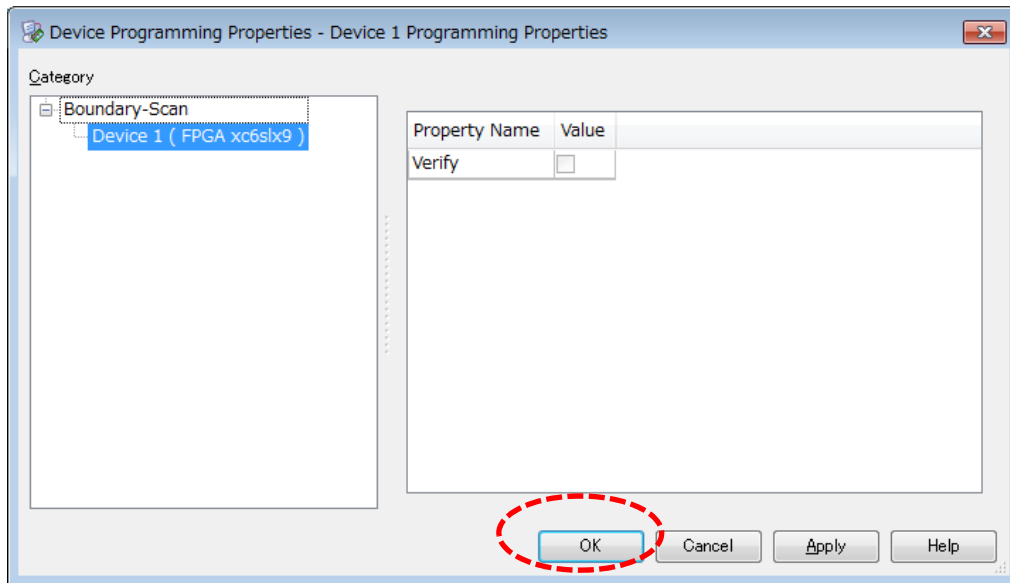


mb_mcs_sys.bit を指定



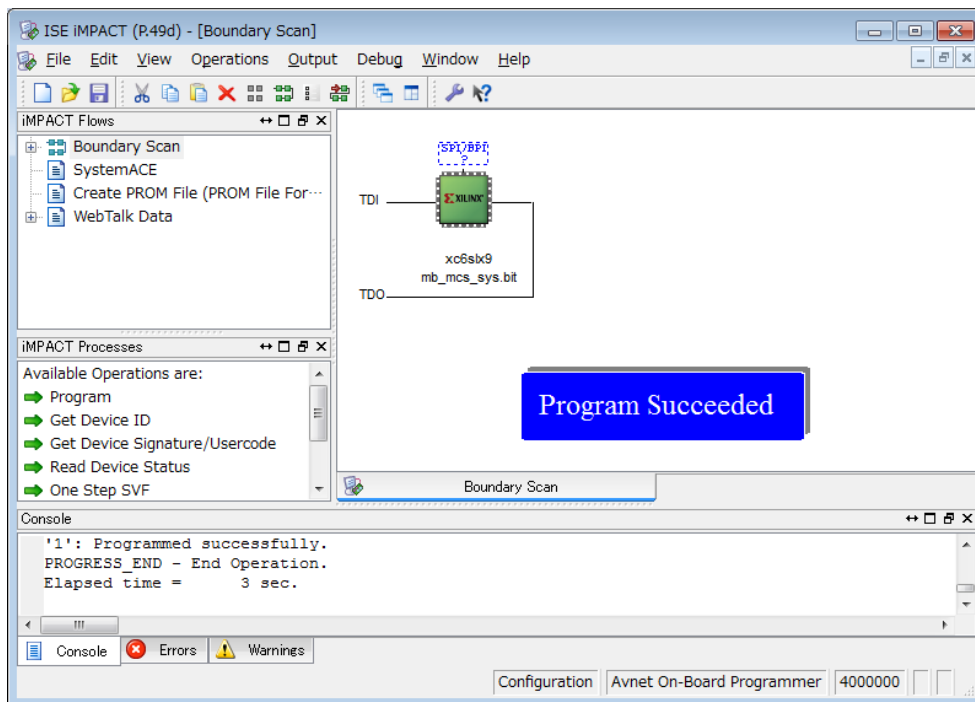
PROM データは使わないので No を選択





FPGA へのプログラミング実行、デバイス上でマウス右ボタンを押して Program 選択

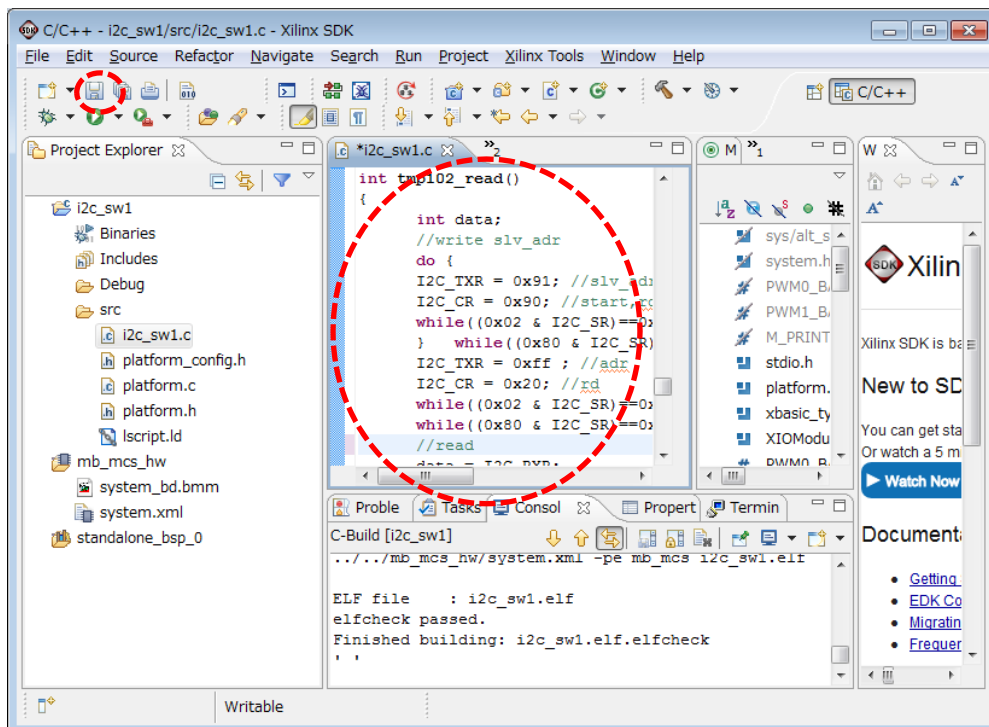




Program Succeeded と表示されればプログラミング完了、LX9 マイクロボードの動作を確認する。
LCD モジュールに温度が表示されていれば動作 OK です。

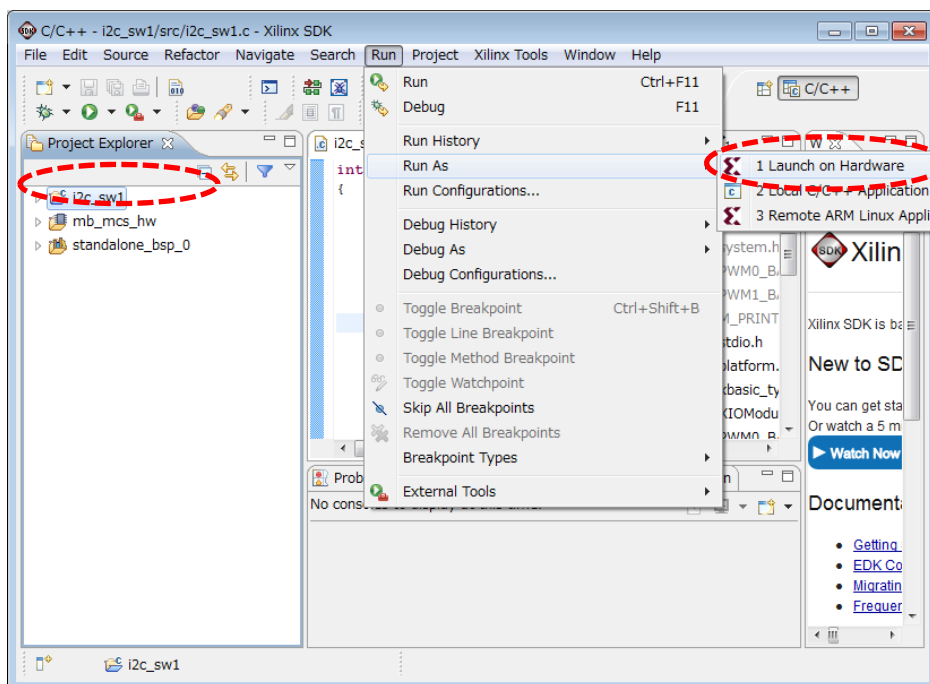


- ・ソフトウェアのみを変更した場合は SDK から elf ファイルを FPGA にダウンロードできます。



C ソースコードを変更して、save するとビルドが実行されて新しい elf ファイルが作成さる。





i2c_sw1 をクリックして選択

Run→Run As →Launch on Hardware で新しい elf が FPGA へダウンロードされて実行される
ダウンロードに失敗する場合は SDK を終了させて再起動してください。