

MicroBlazeMCS ソフト開発編用設計データの使い方

ファイル一覧

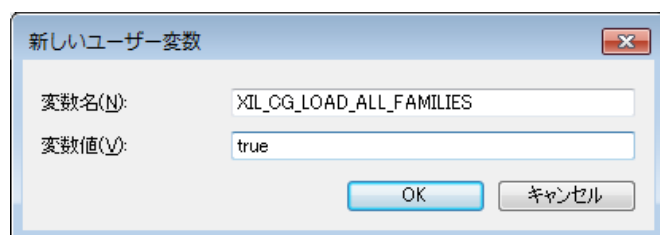
- ・ mb_mcs_sys.v トップ回路
- ・ iobus_reg.v 設定レジスタ回路
- ・ iobus_bram.v ブロック RAM インターフェース回路
- ・ user_module.v デバック回路
- ・ test_mcs_sys.v テストベンチ
- ・ mb_mcs_sys.ucf ピン配置指定
- ・ led_sw1.c IO モジュール用関数使用の LED 点灯 C ソースコード
- ・ led_sw2.c レジスタ直接書き込みの LED 点灯 C ソースコード
- ・ int_sw1.c 割り込み動作確認用 C ソースコード

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

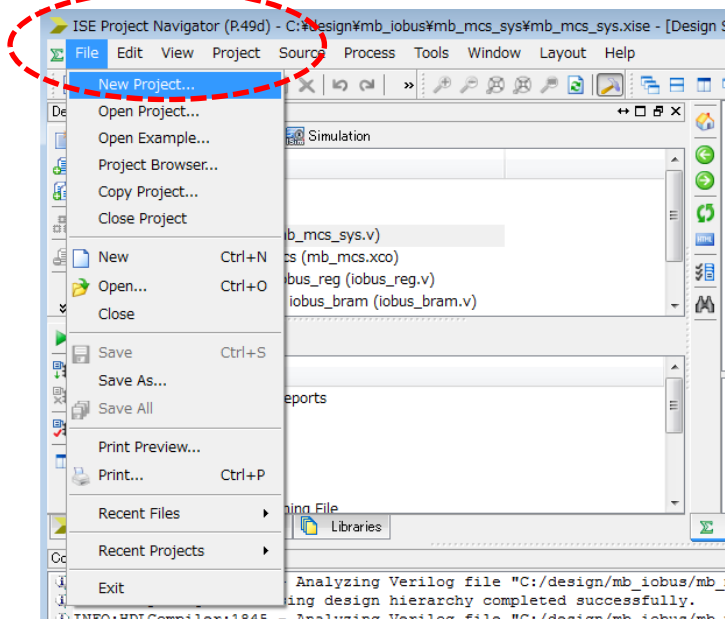
- ・ 設計用フォルダ mb_iobus/mb_mcs_sys を作成する
- ・ 記事のダウンロードデータ mcs_soft_data の下にある Verilog-HDL ソースと UCF ファイルを mb_iobus/mb_mcs_sys へコピー
- ・ 環境変数 XIL_CG_LOAD_ALL_FAMILIES が設定されてない場合は環境変数を追加する
コントロールパネル→システムとセキュリティ→システム→システムの詳細設定→環境変数→
ユーザー環境変数：新規をクリック



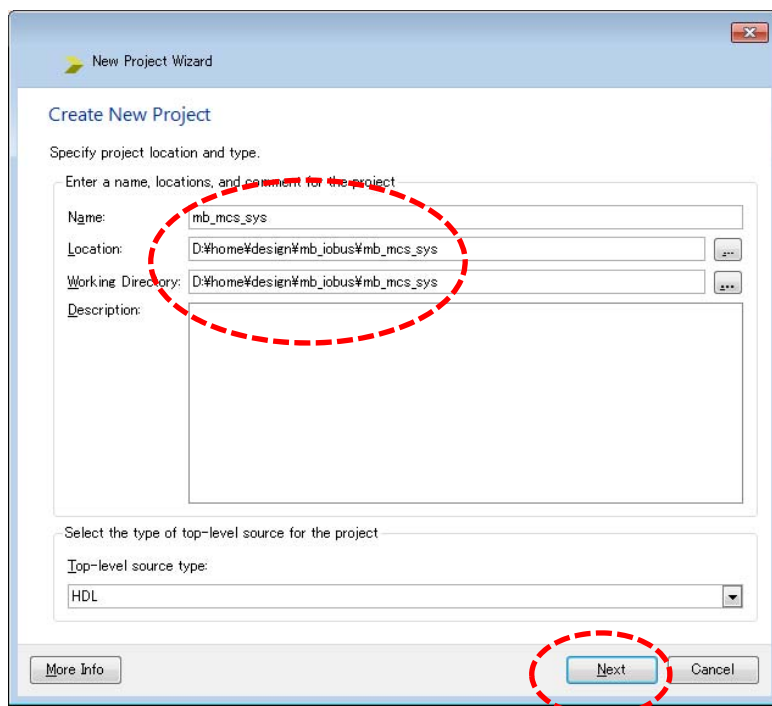
変数名：XIL_CG_LOAD_ALL_FAMILIES 値：true を入力して OK

- ・ 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」を起動します。

・新規設計プロジェクト作成



File→New Project 選択



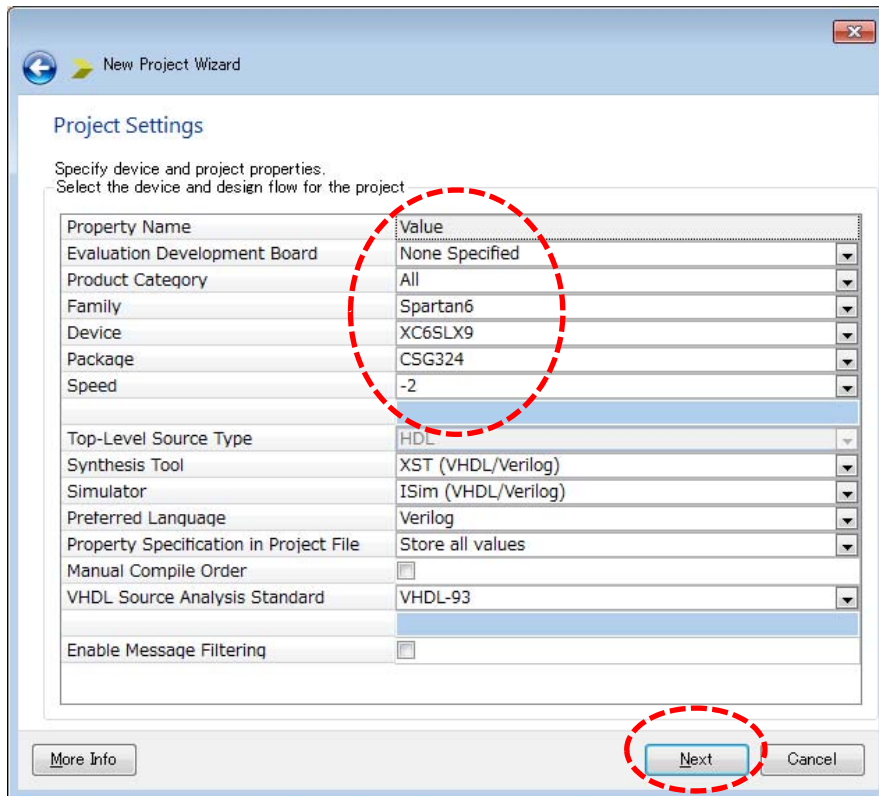
プロジェクト名と設計フォルダ指定

プロジェクト名 : mb_mcs_sys

設計フォルダ : 任意

次に Next をクリック



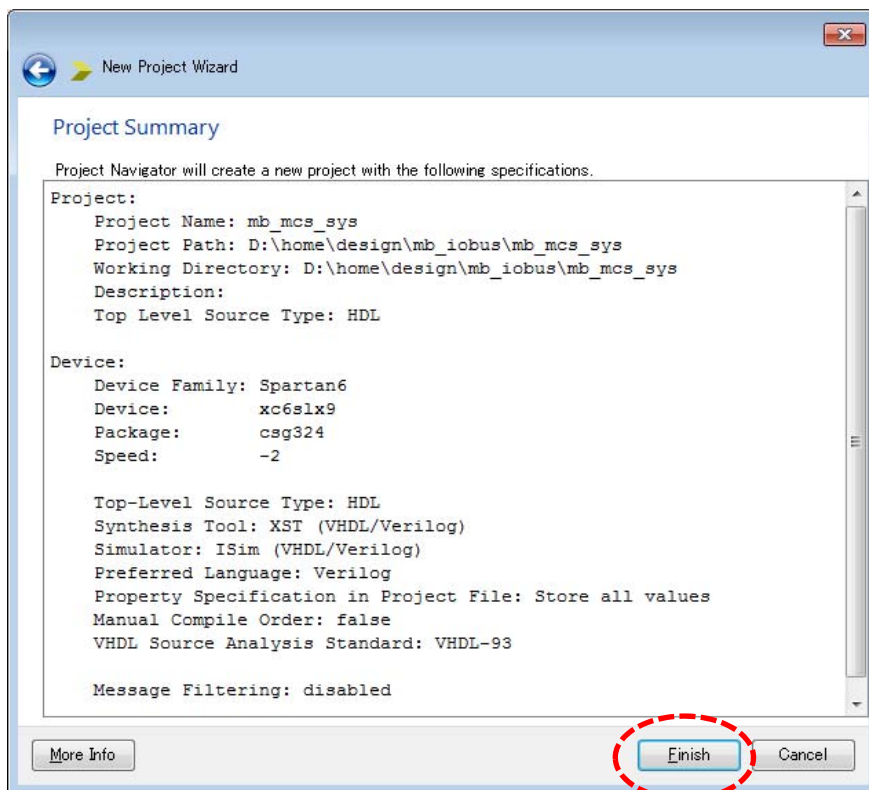


The 'New Project Wizard' dialog box is shown at the 'Project Settings' step. It contains a table of properties and their values. A red dashed circle highlights the 'Device' section, which includes 'Family' (Spartan6), 'Device' (XC6SLX9), and 'Package' (CSG324). Another red dashed circle highlights the '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: D:\home\design\mb_iobus\mb_mcs_sys
- Working Directory: D:\home\design\mb_iobus\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

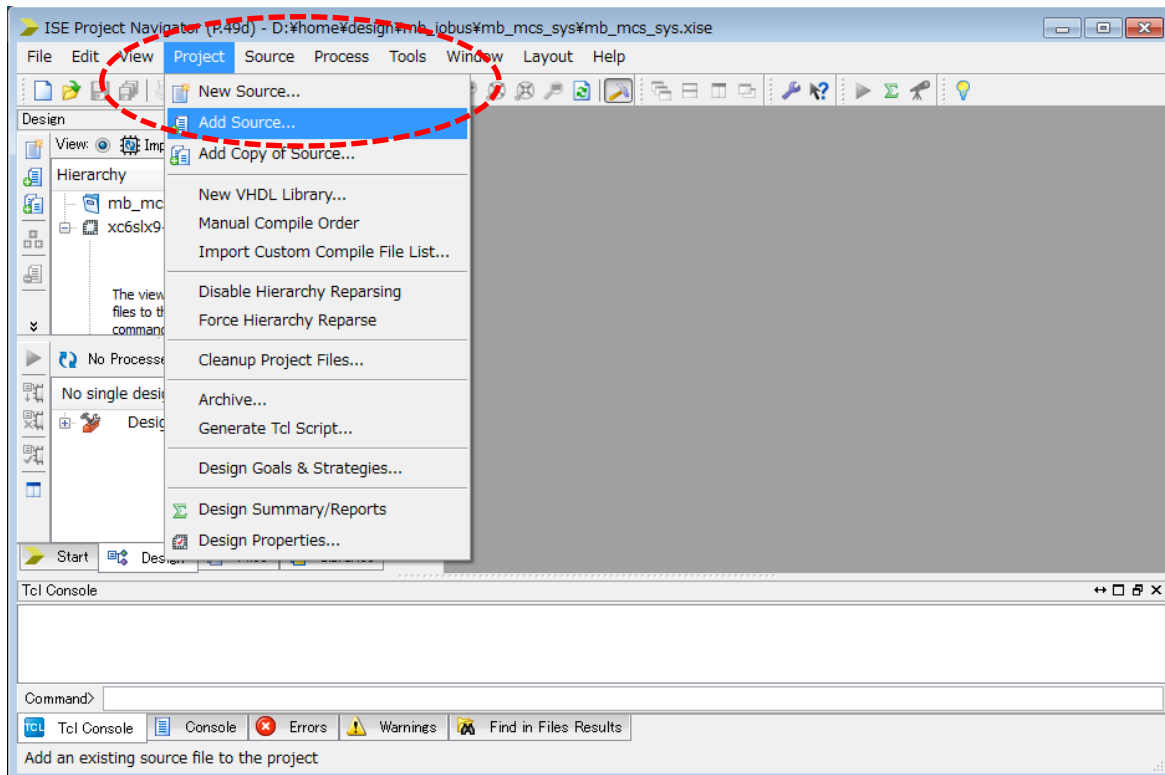
VHDL Source Analysis Standard: VHDL-93

Message Filtering: disabled

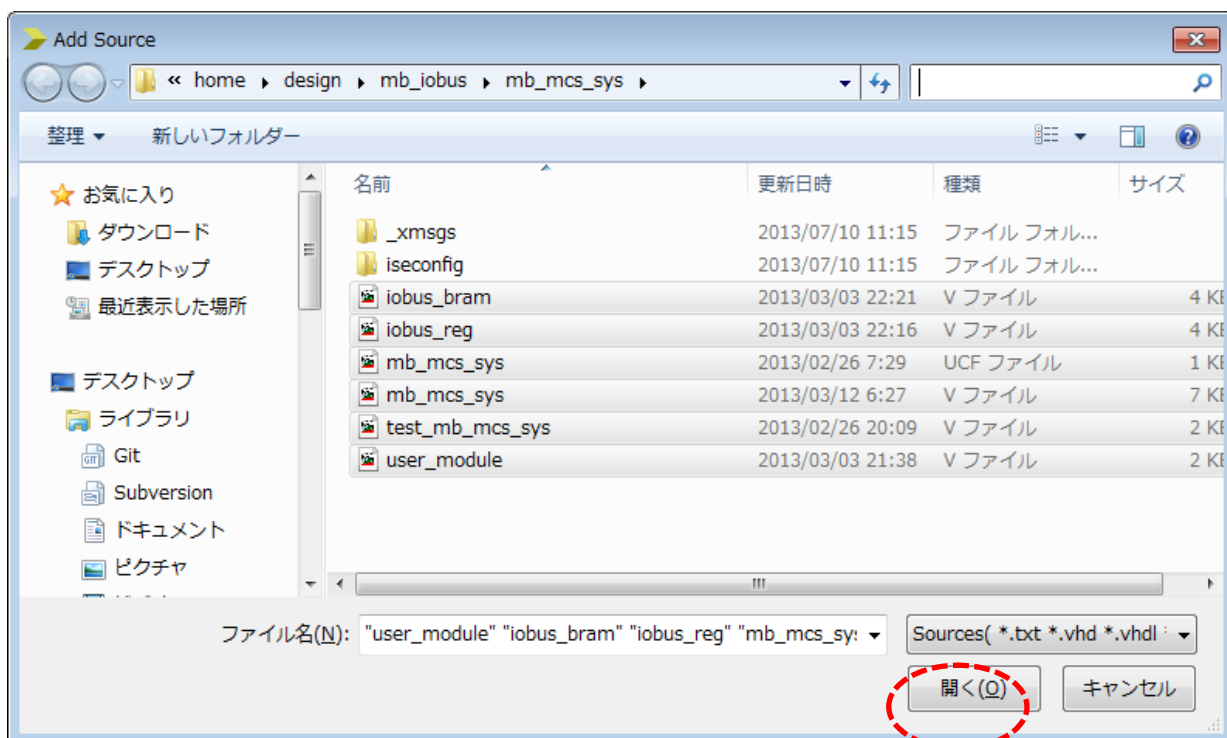
Buttons: More Info, Finish, Cancel

Finish をクリック



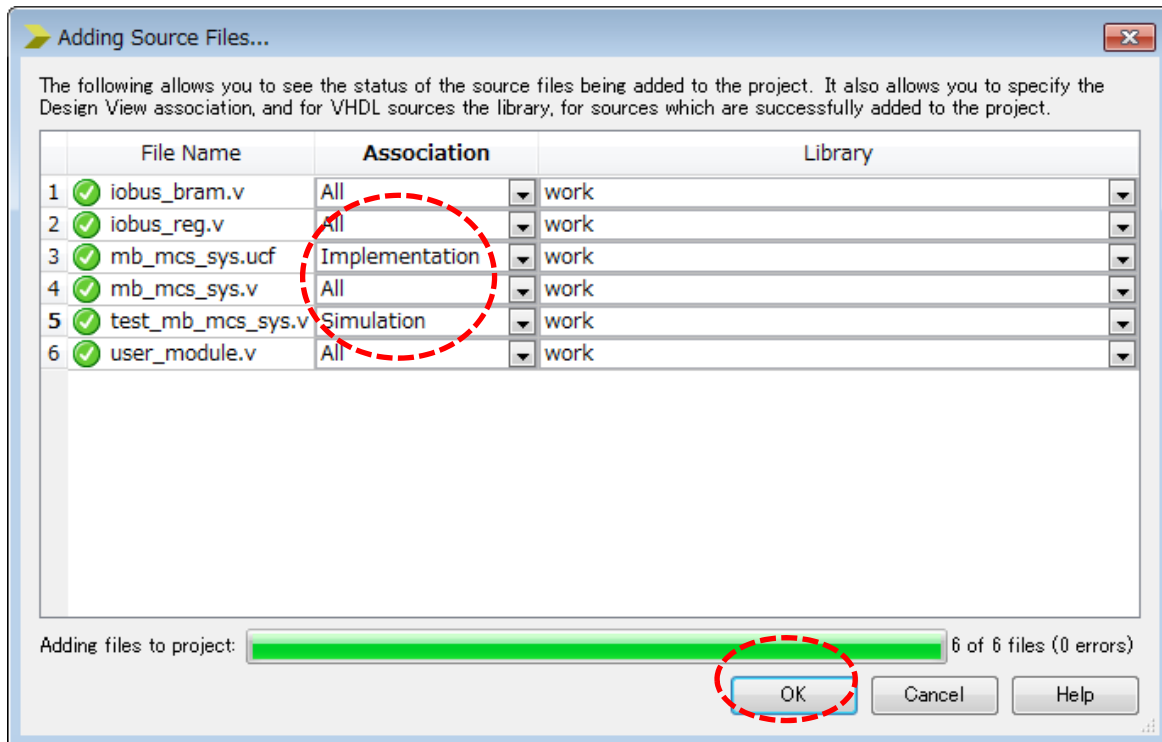


設計データを追加します。Project→Add Source を選択

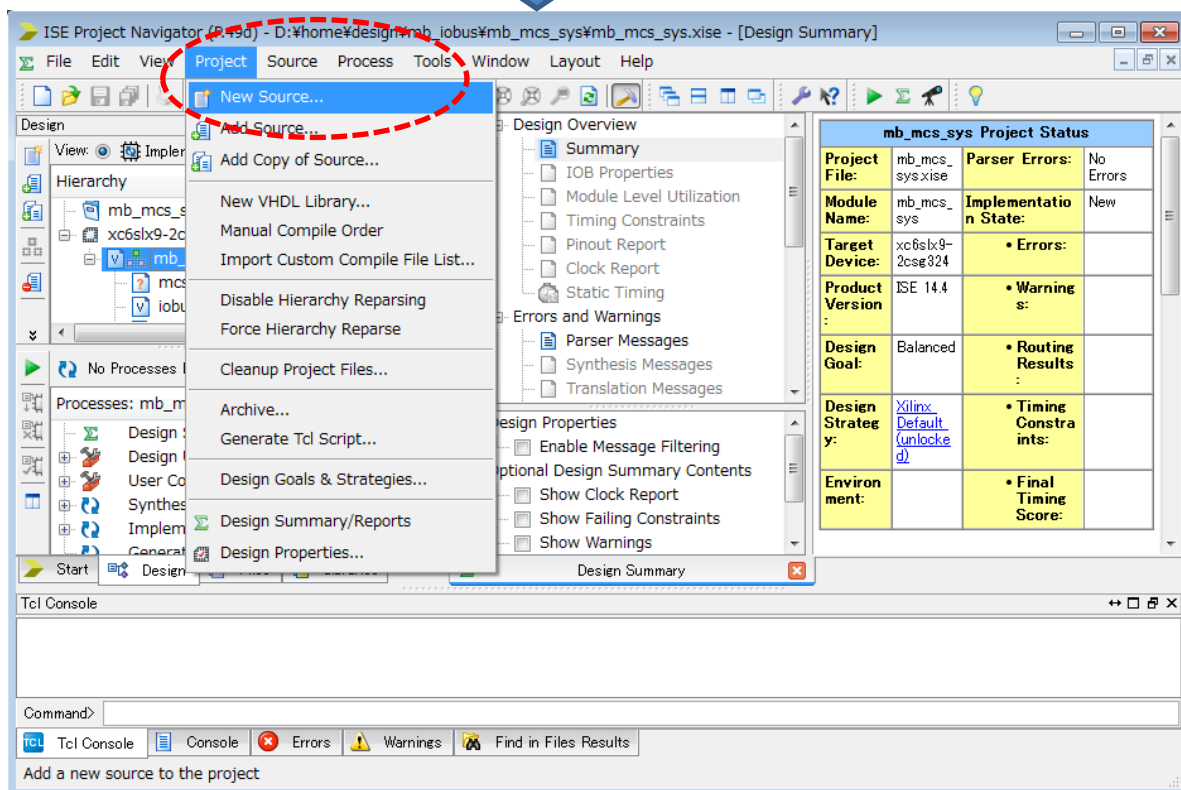


ファイルの指定、mb_mcs_sys の下にある Verilog-HDL ファイルと UCF ファイルを指定

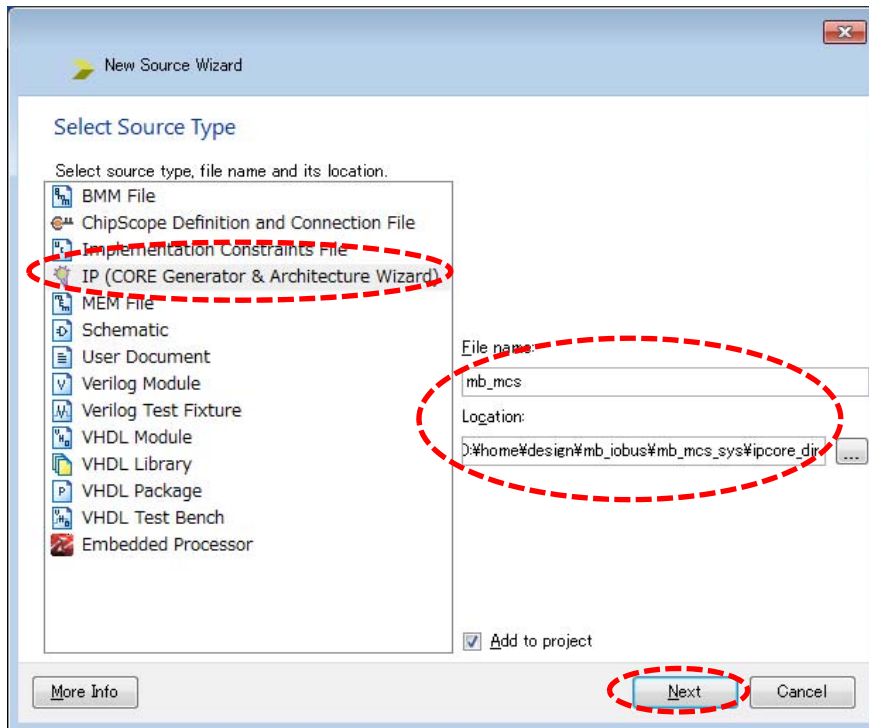




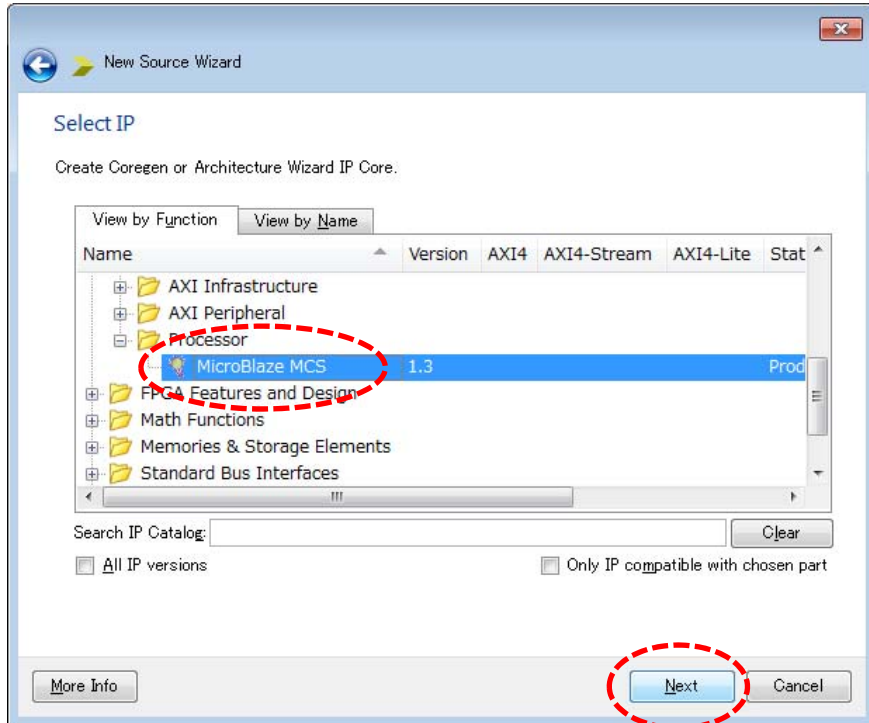
mb_mcs_sys.ucf はインプリメンテーションで使用するので Implementation を指定
test_mb_mcs_sys.v はテストベンチなので Simulation を指定、OK をクリック



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

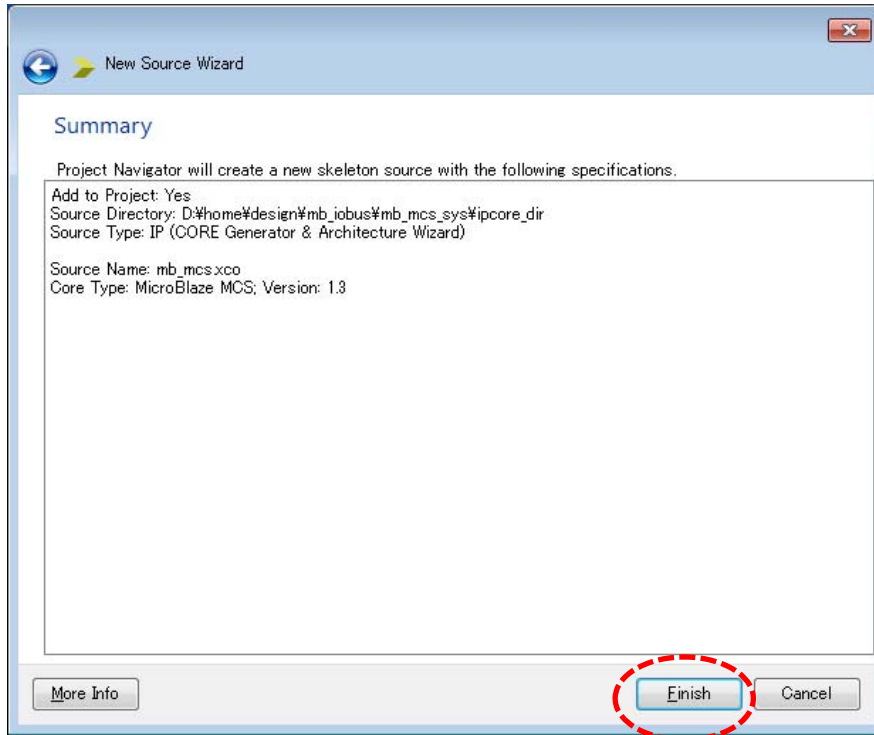


IP (CORE Gener...をクリックして選択、ファイル名を mb_mcs に指定、Next をクリック

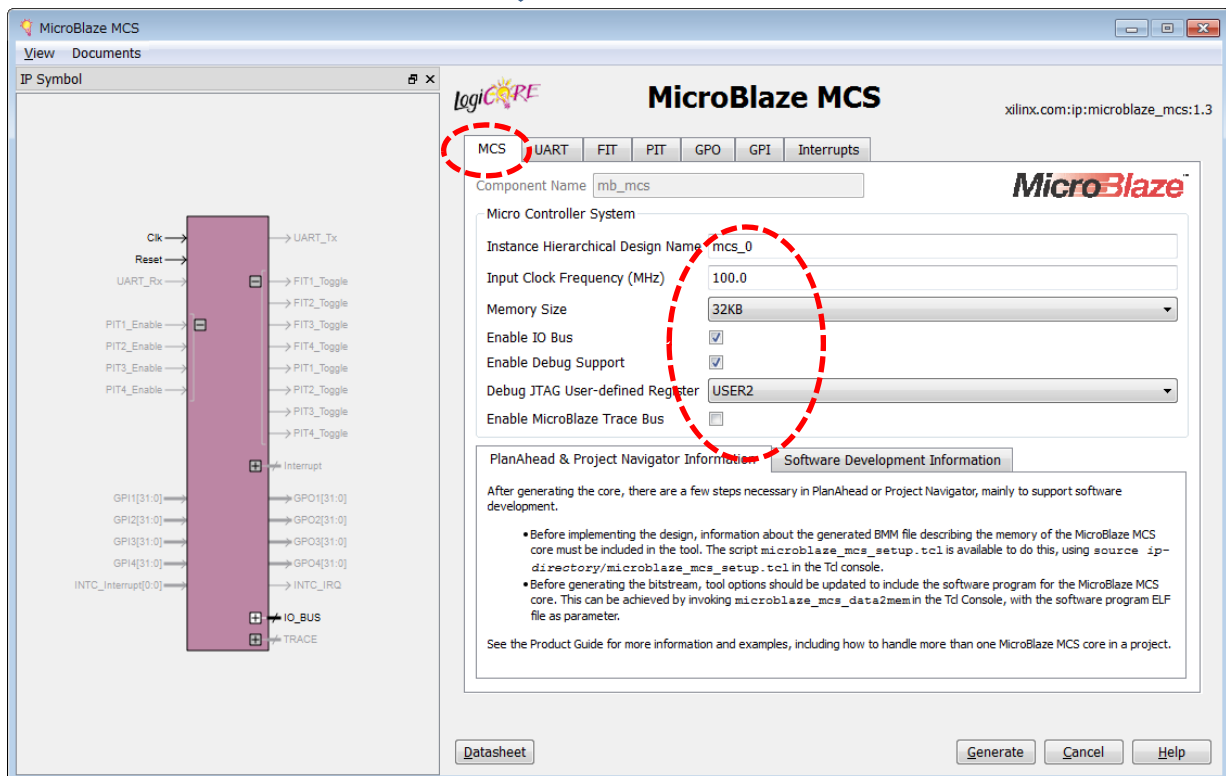


MicroBlaze MCS 選択して Next をクリック



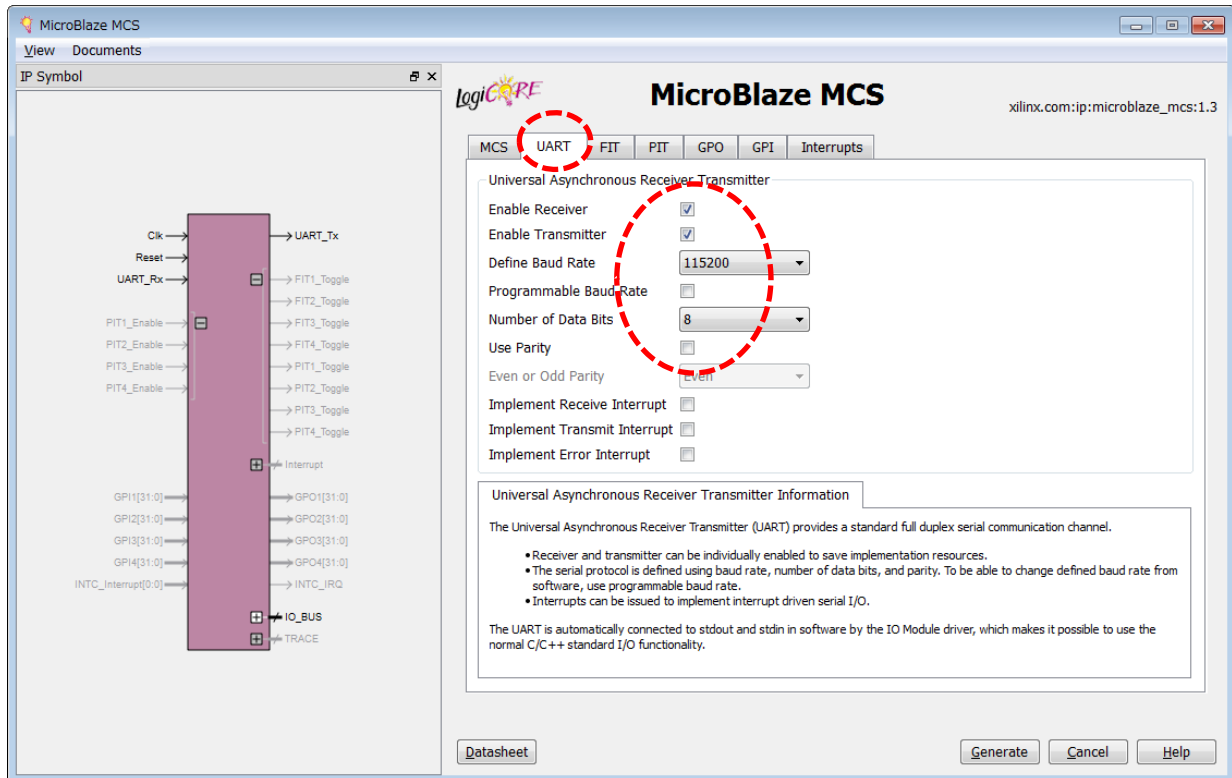


Finish をクリックで CORE generator が起動

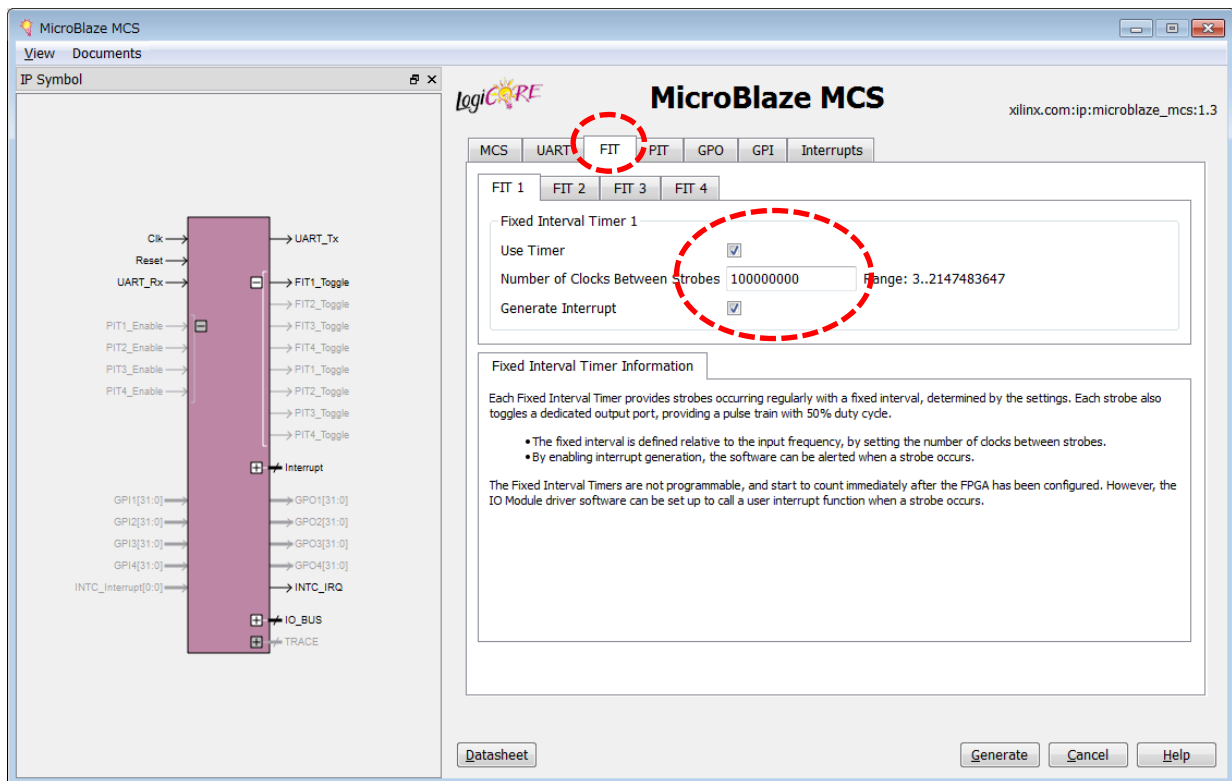


MicroBlaze MCS の基本設定



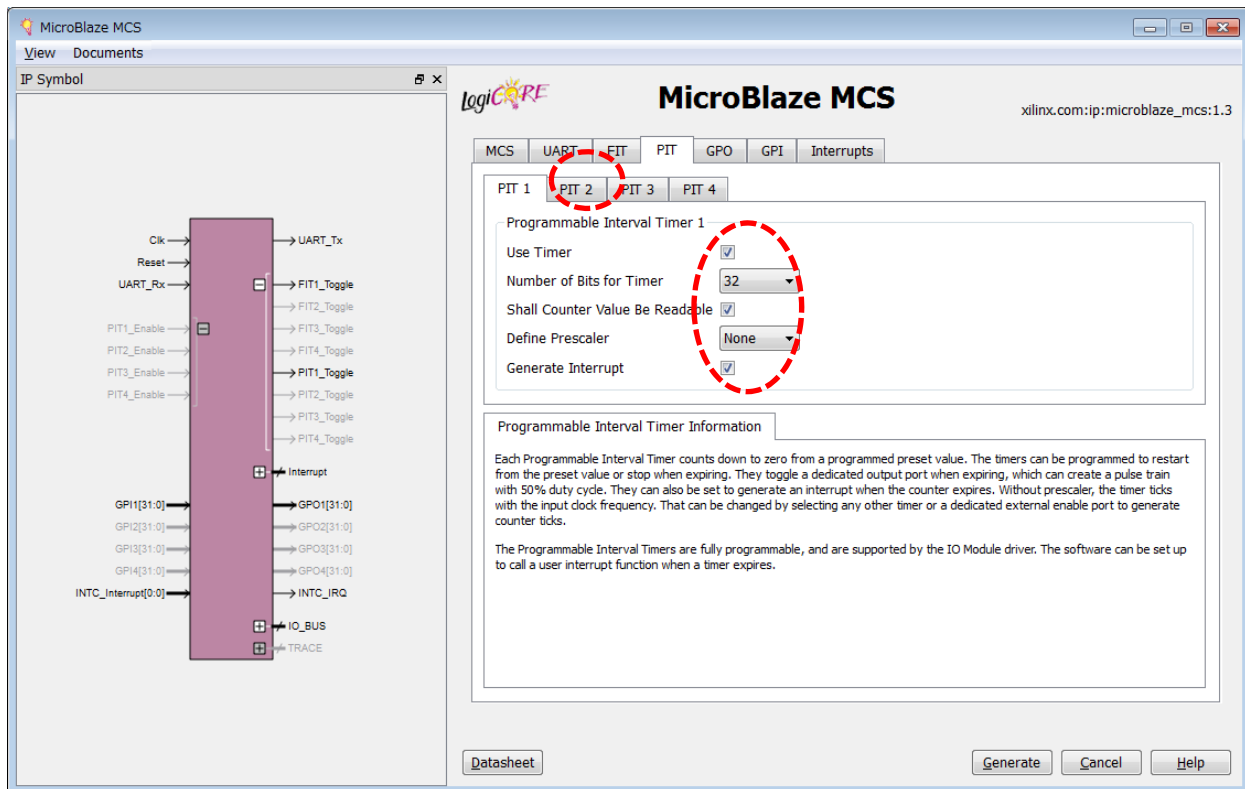


UART の設定

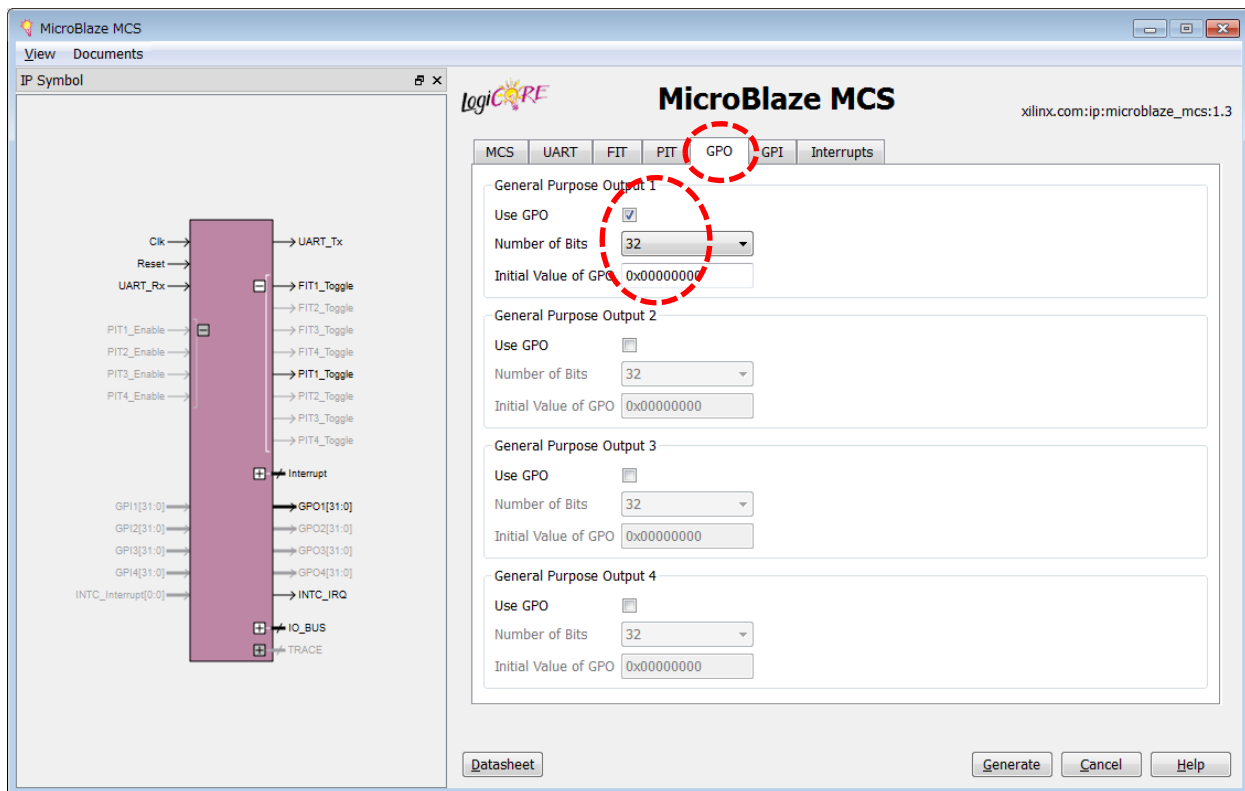


FIT の指定



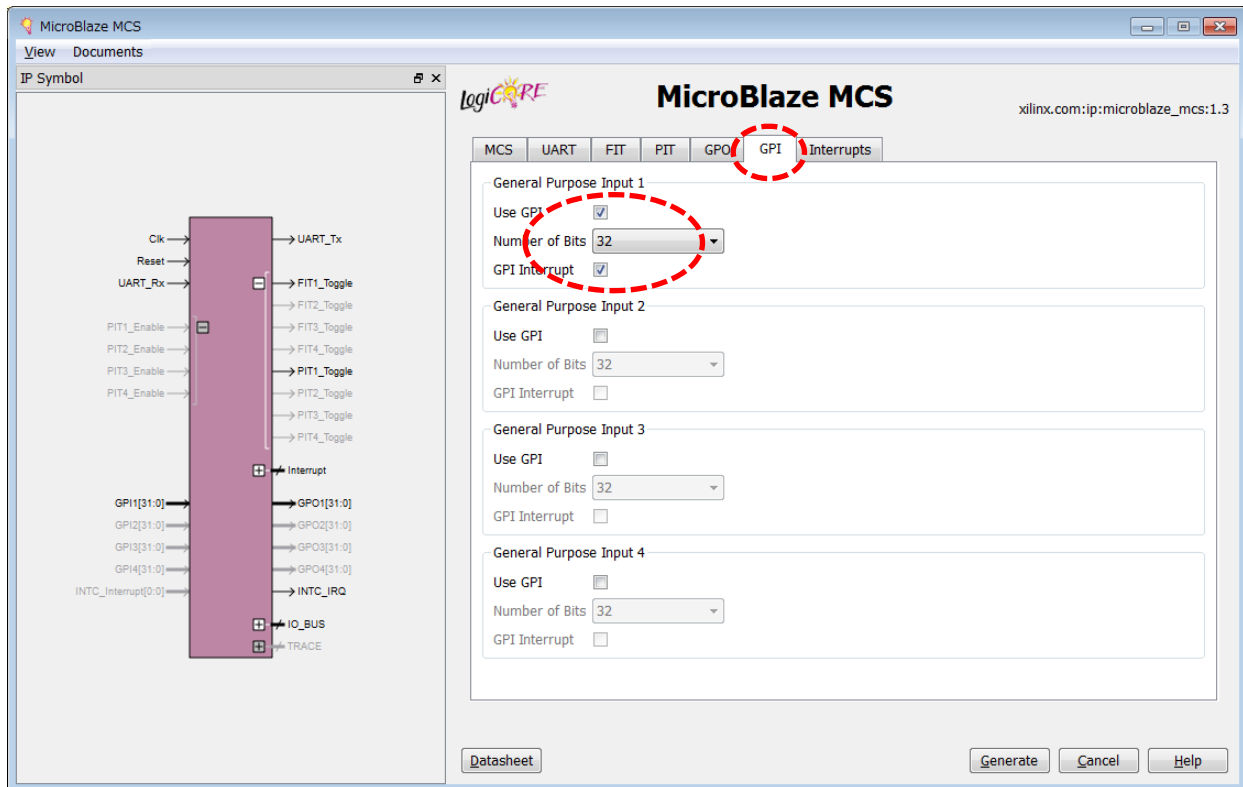


PIT の指定

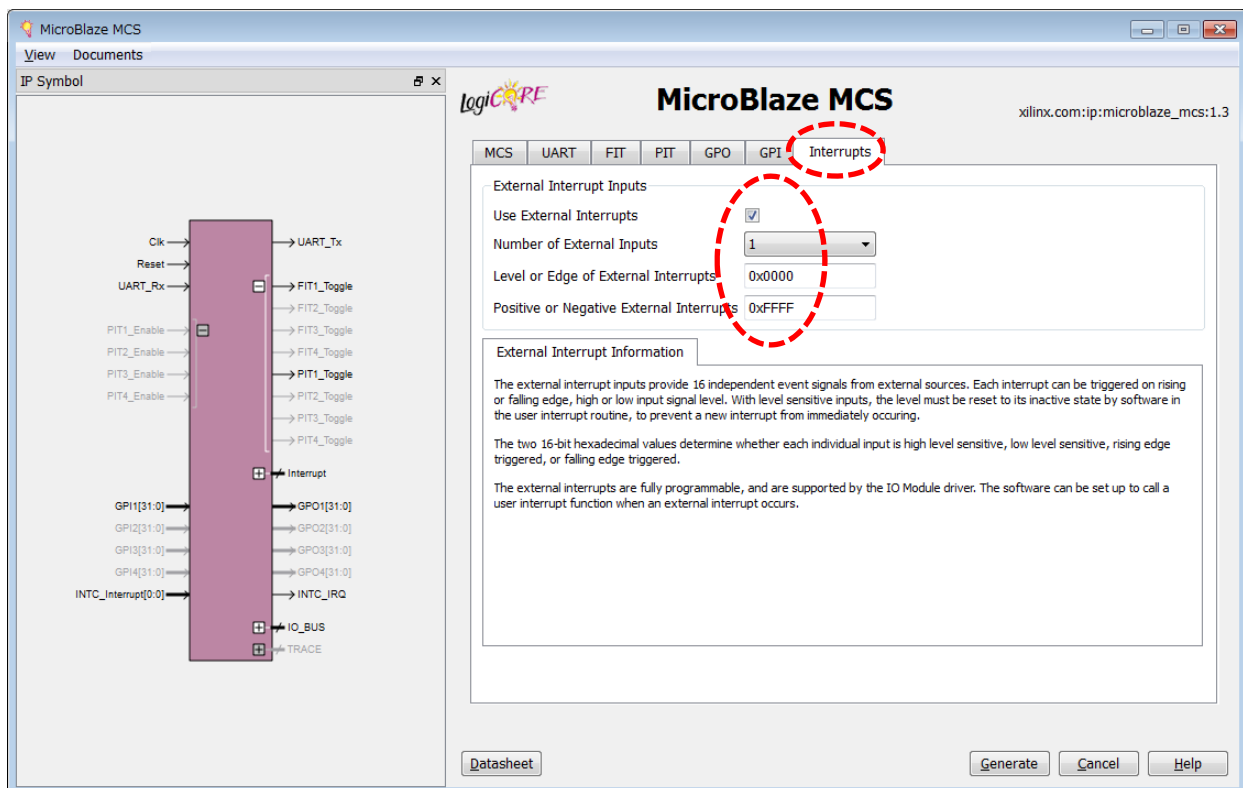


GPO の設定



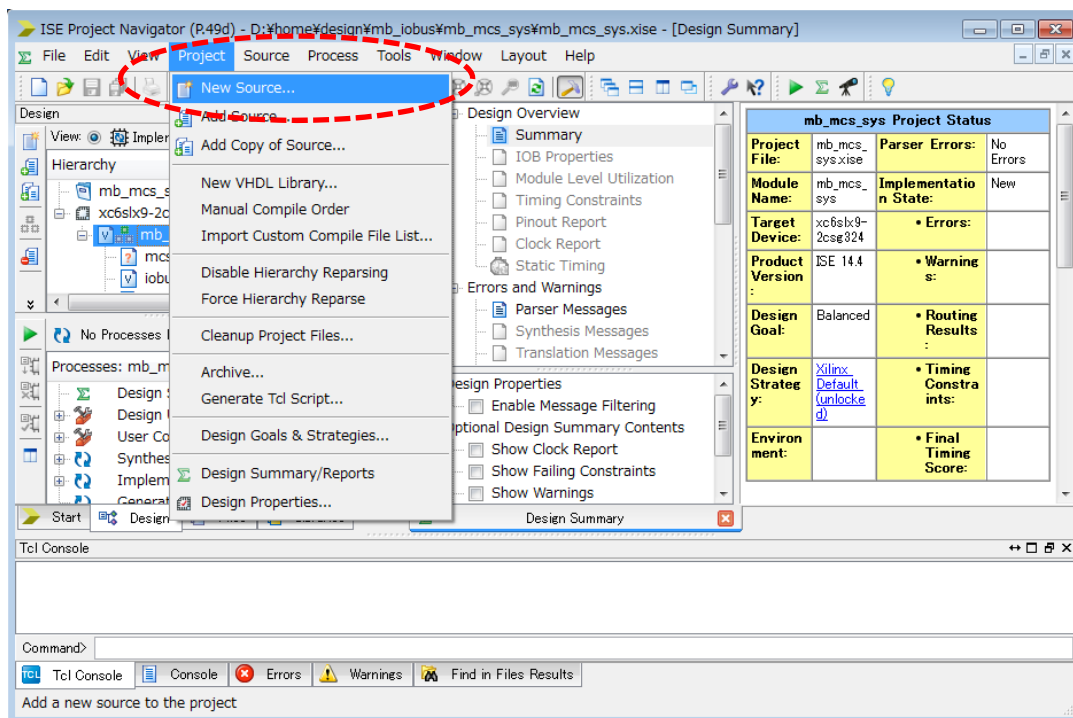


GPI の指定

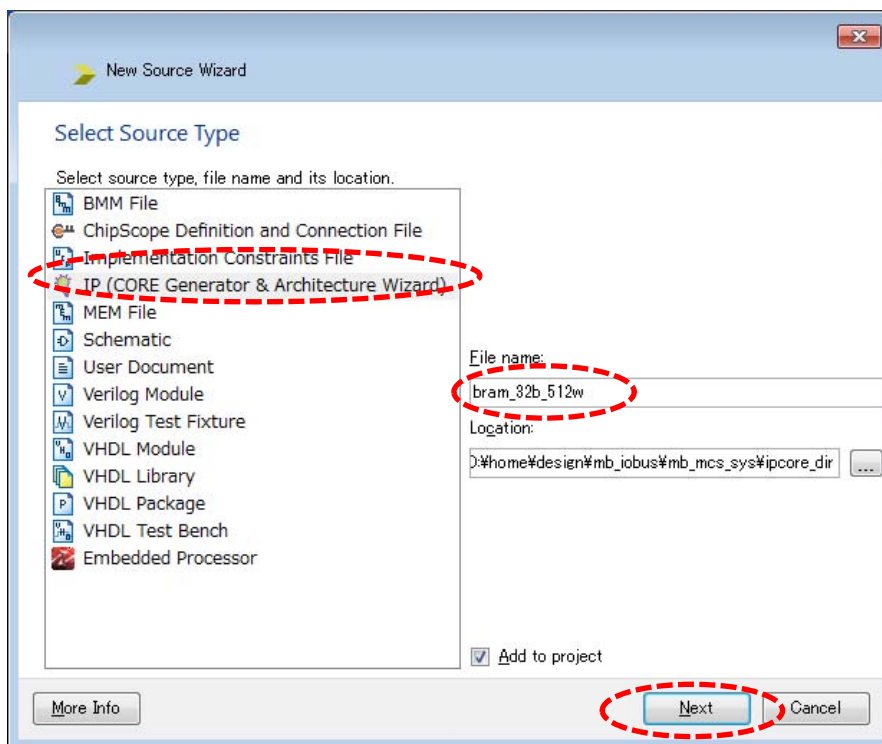


外部割り込みの設定後に Generate をクリックすると MicroBlaze MCS がプロジェクトに追加される



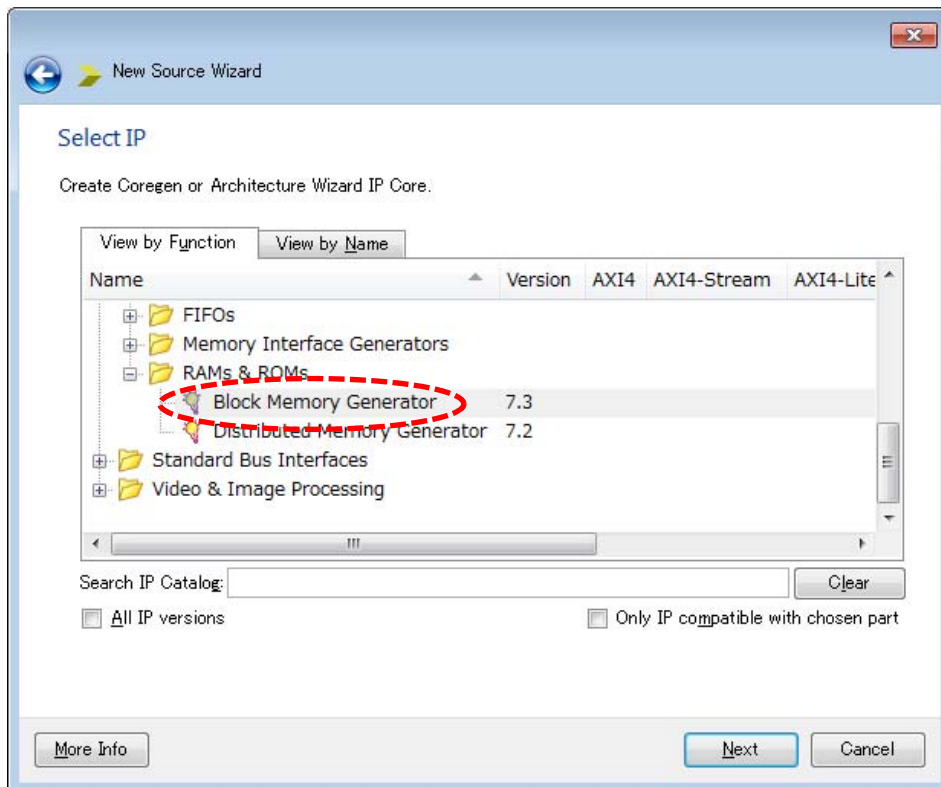


ブロック RAM の作成、Project→New Source を選択

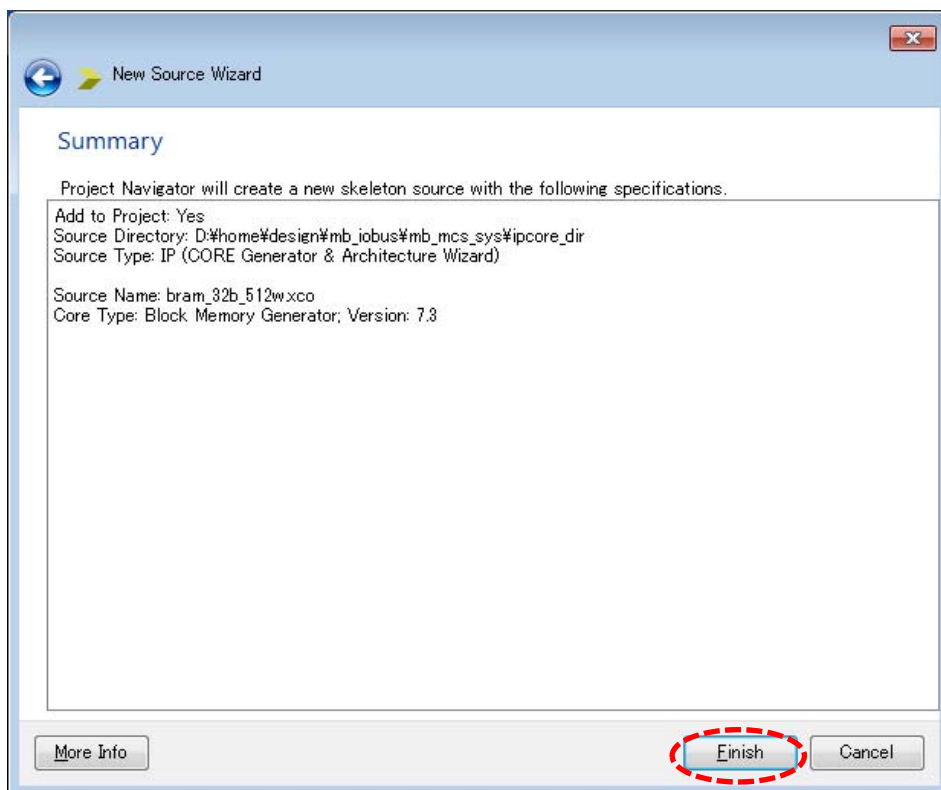


IP (CORE Gener...をクリックして選択、ファイル名に bram_32b_512w を指定、Next をクリック



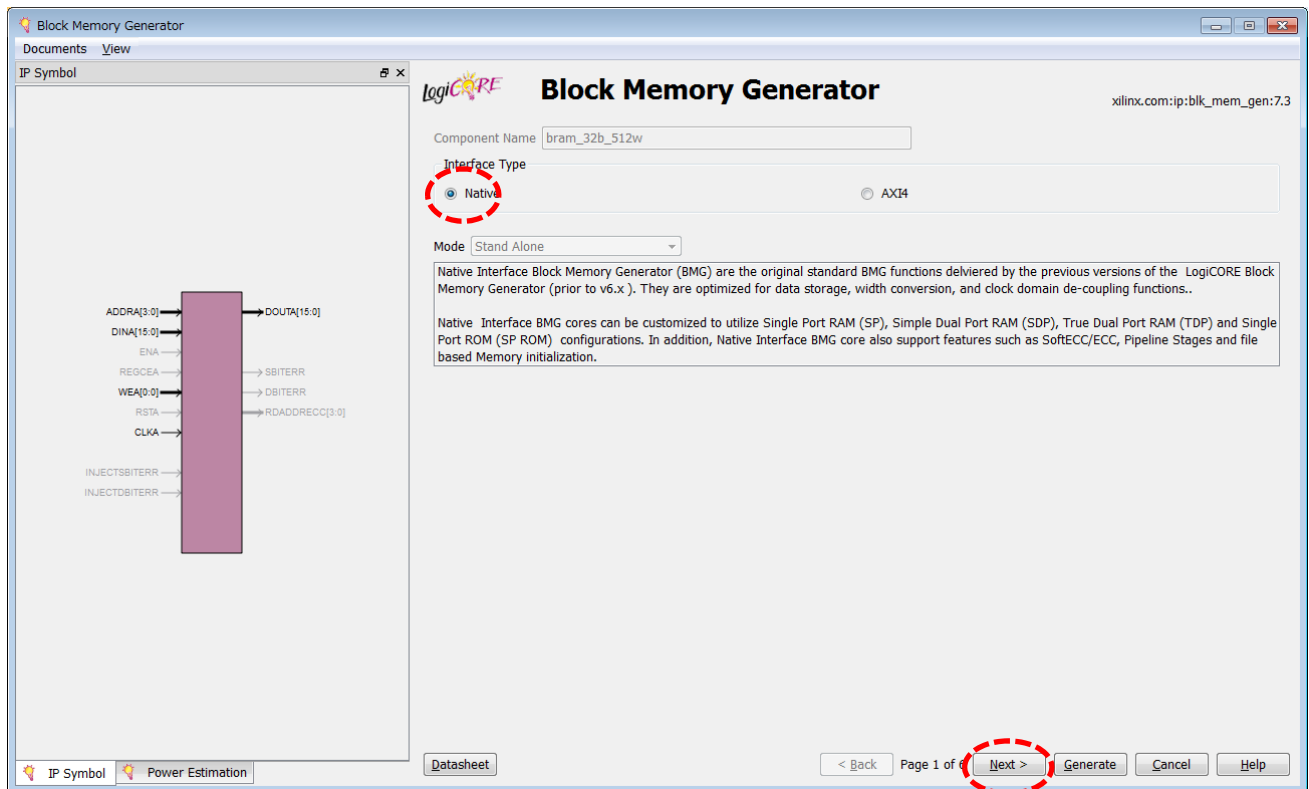


Block Memory Generator を選択して Next をクリック

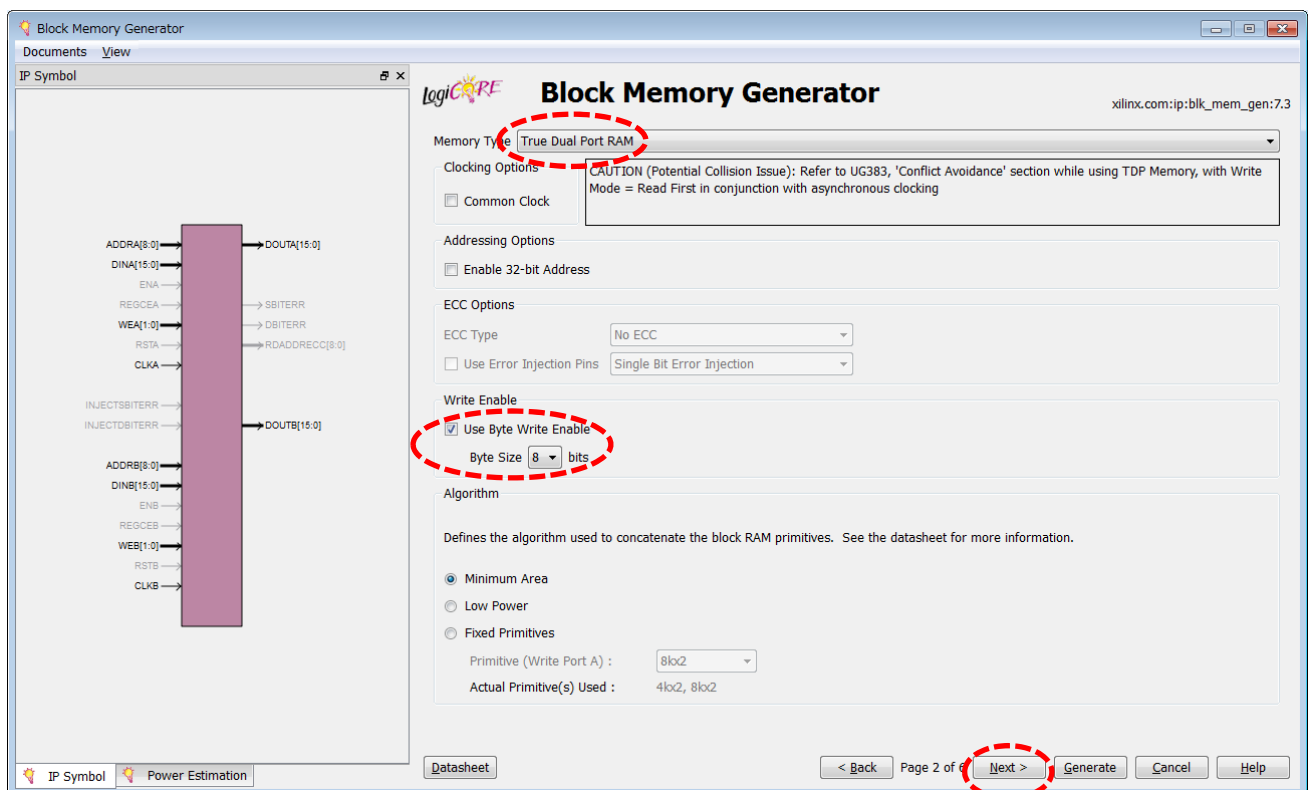


Finish をクリックすると CORE generator が起動



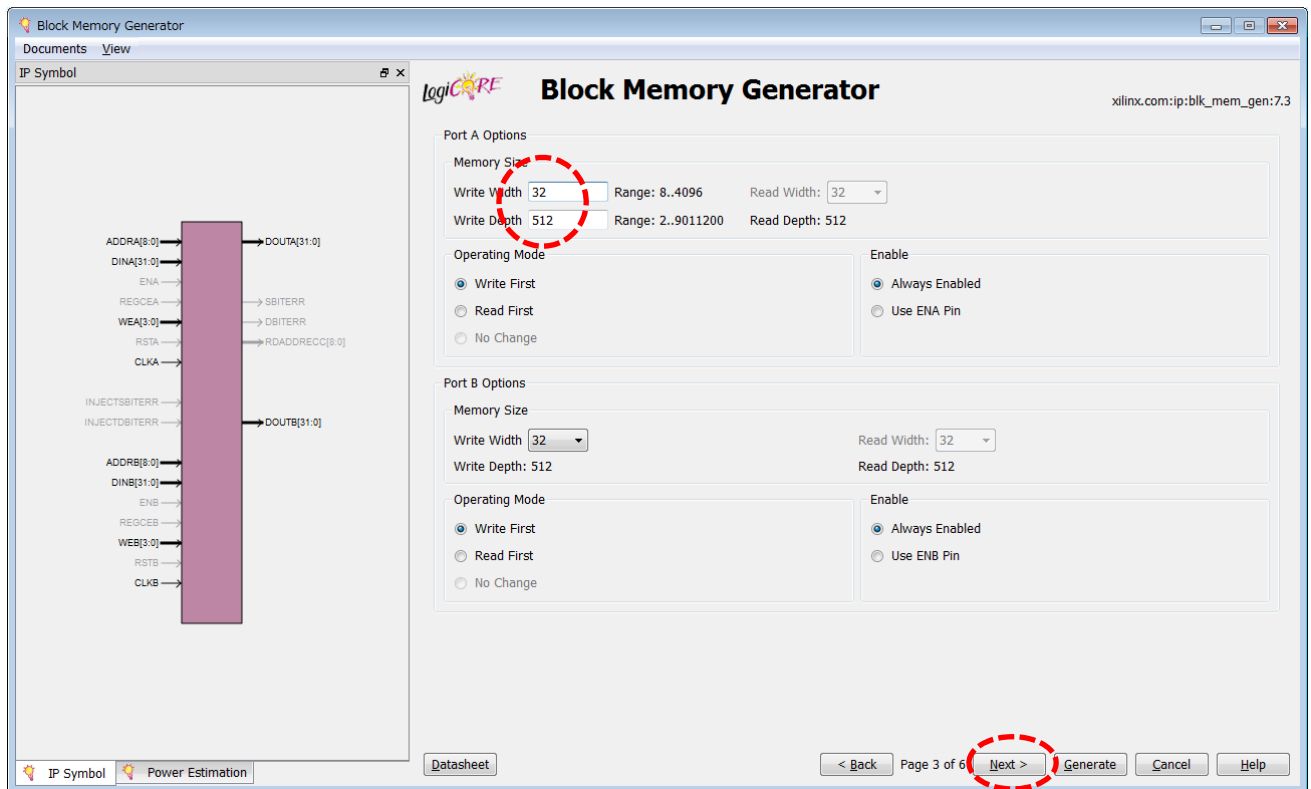


ブロック RAM インターフェースの設定

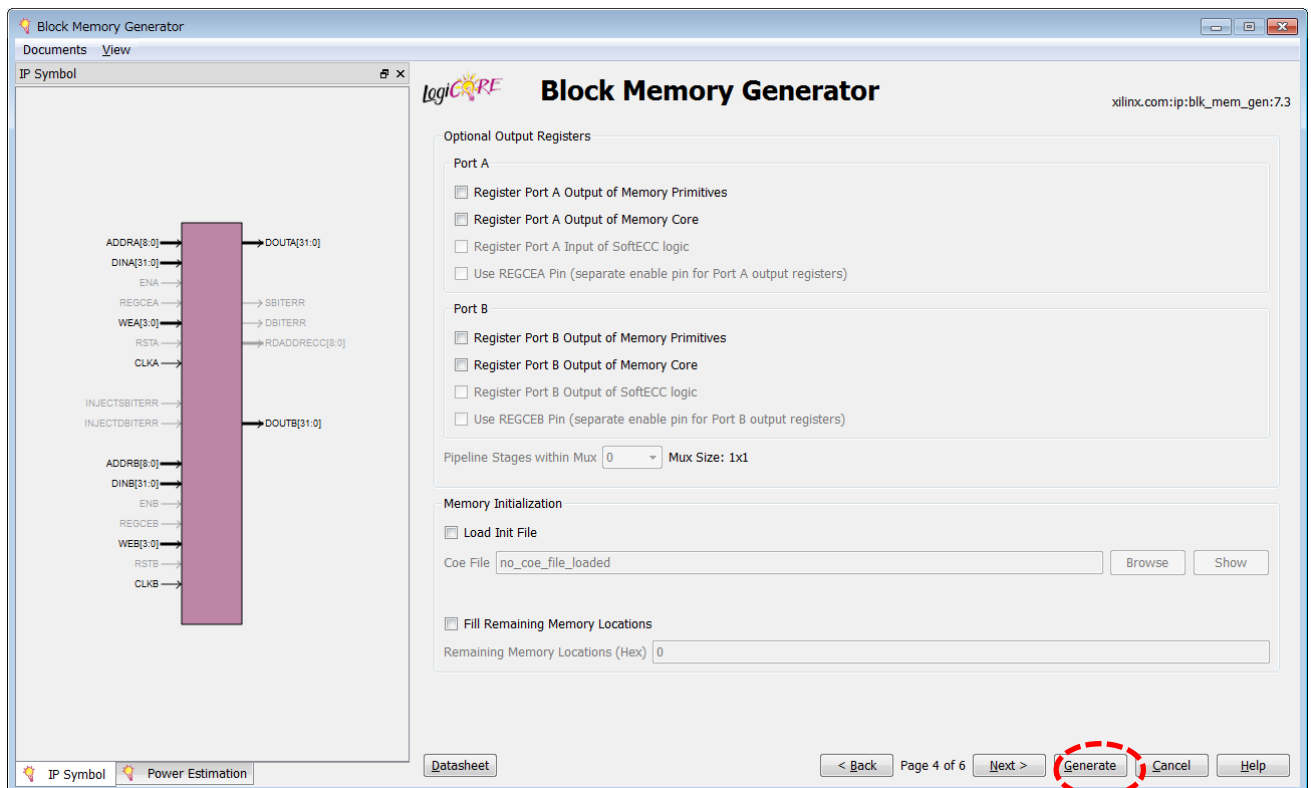


メモリのタイプ指定



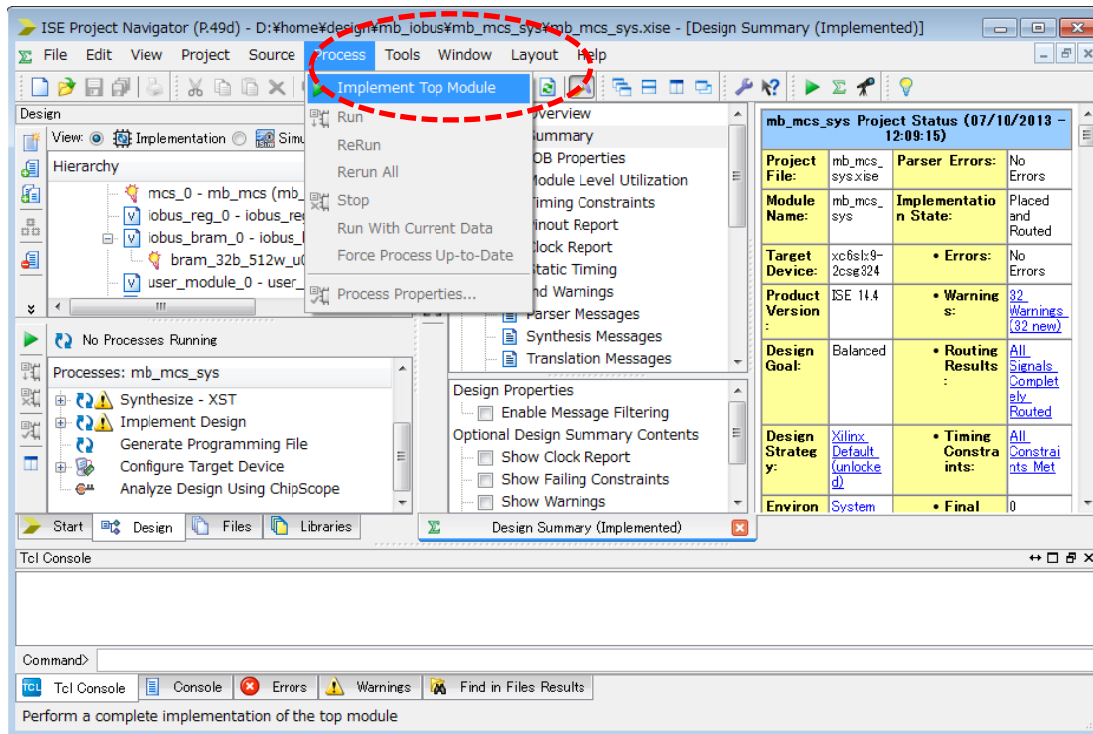


データ幅、データ量の指定



Generate をクリックで作成開始





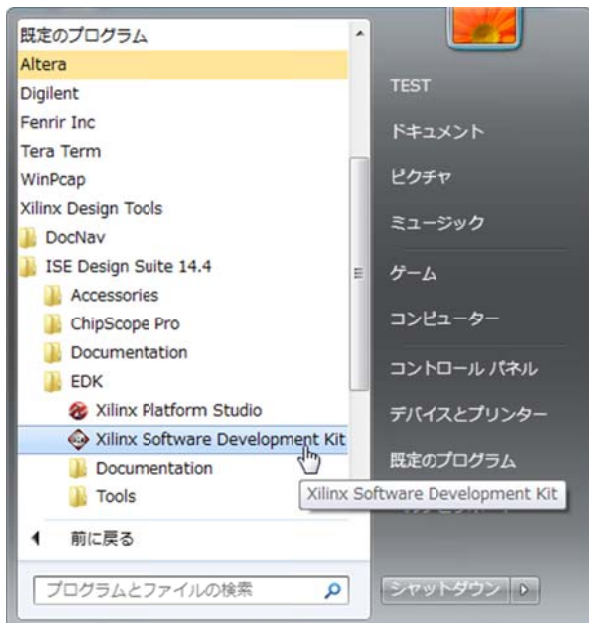
インプリメンテーションの実行、Process→Implement Top Module をクリック

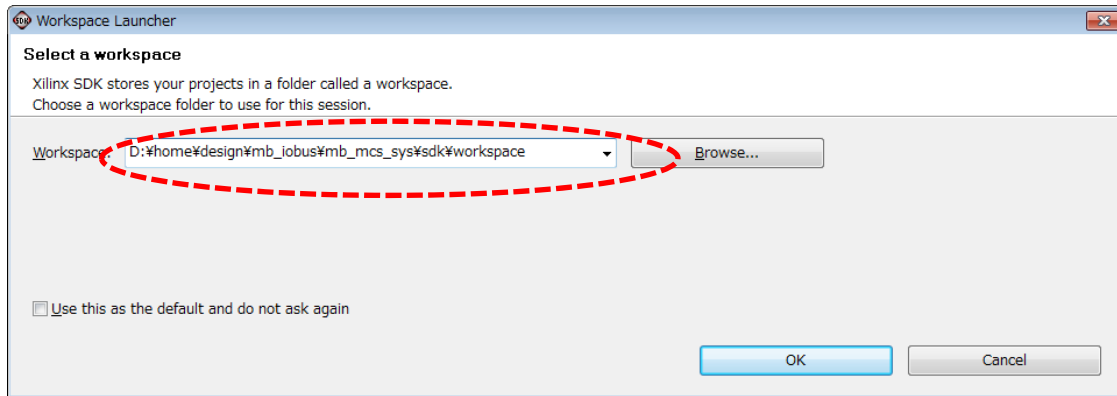


次に Xilinx Software Development Kit(以降 SDK)でソフトウェアを作ります。

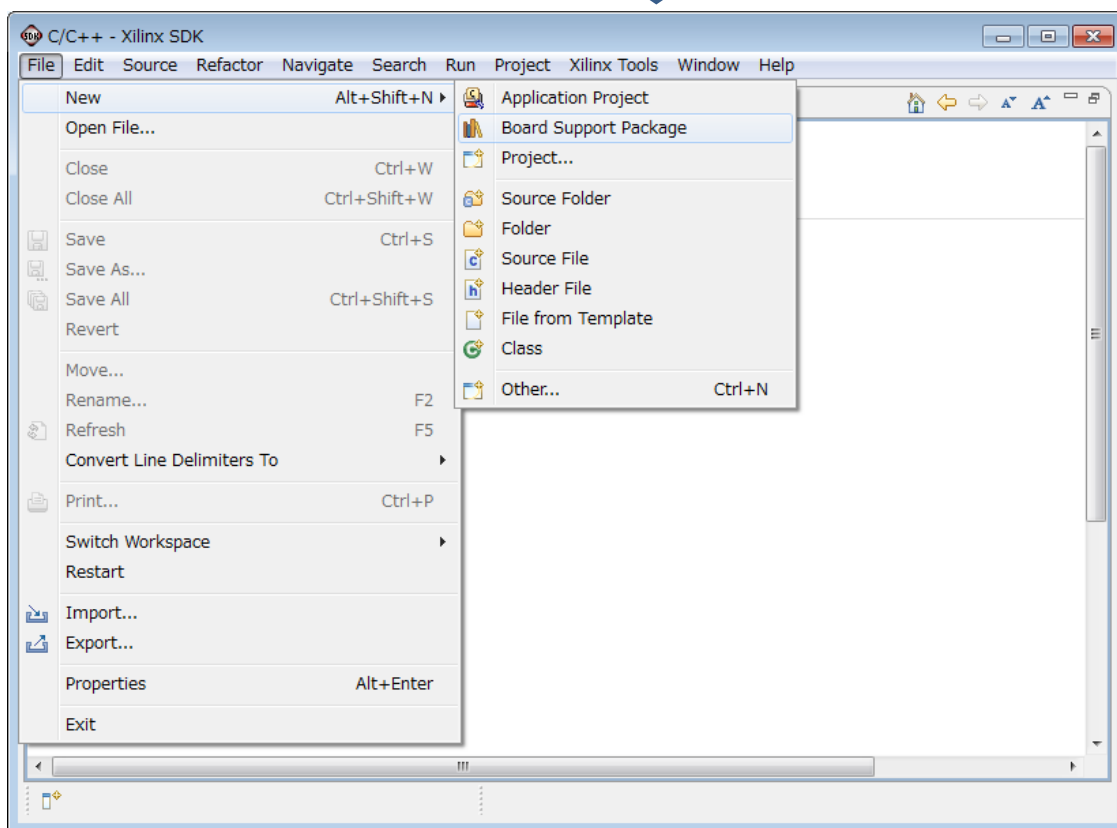
・ SDK の起動

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。

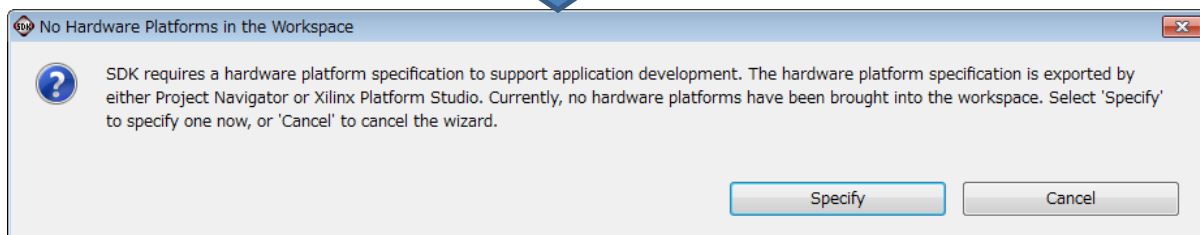




SDK を起動するとワークスペースを指定が要求されます。設計フォルダ/sdk/workspace を設定



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



New Hardware Project
Create a new Hardware Project.

Project name:

☒ Use default location

Location:

Choose file system:

Provide the path to the hardware specification file exported from Project Navigator or XPS.
This file usually resides in SDK/SDK_Export/hw folder relative to the XPS project location.
The specification file and associated bitstream content will be copied into the workspace.

▶ Bitstream and BMM Files

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



New Board Support Package Project
Create a Board Support Package.

Project name:

☒ Use default location

Location:

Choose file system:

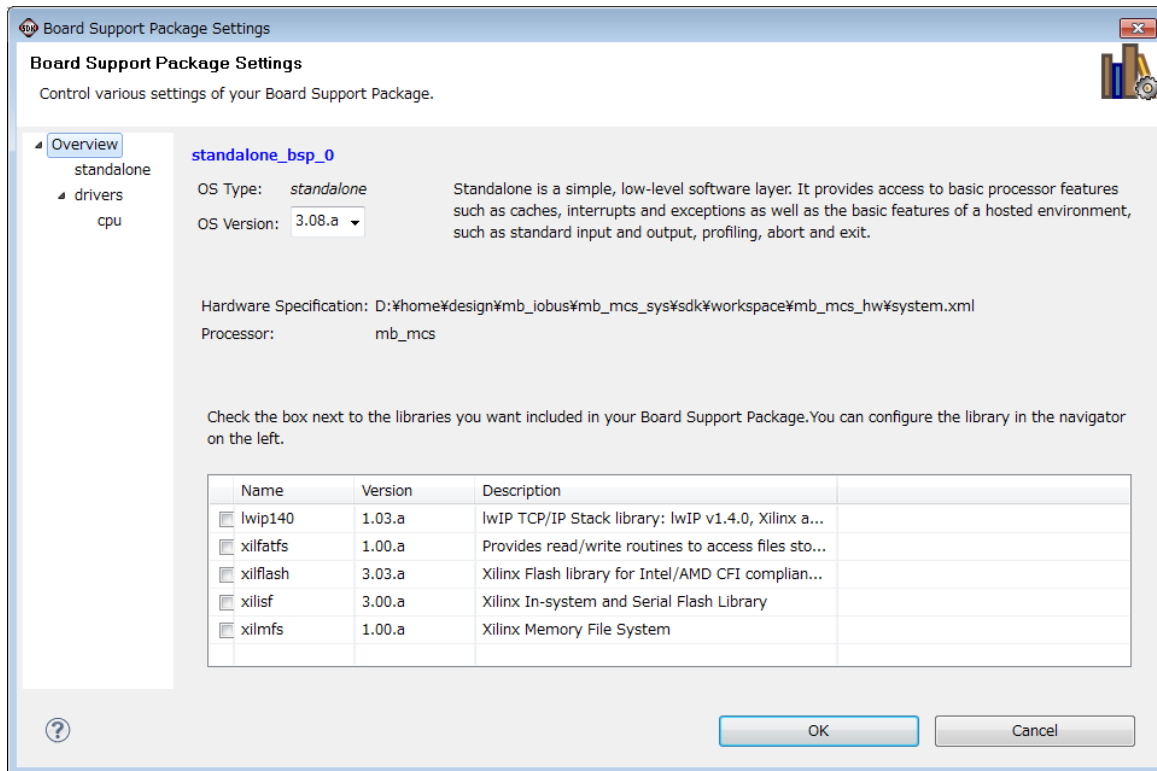
Hardware Platform:

CPU:

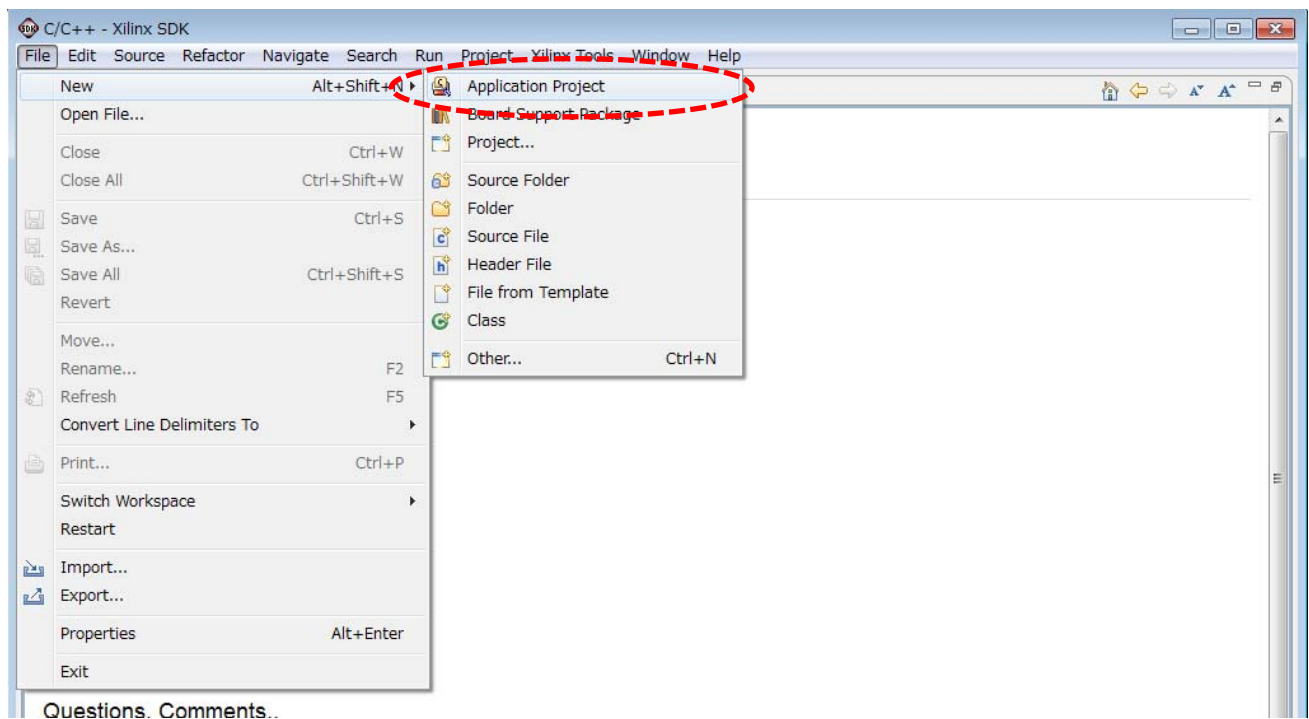
xilkernel	Standalone is a simple, low-level software layer. It provides access to basic processor features such as caches, interrupts and exceptions as well as the basic features of a hosted environment, such as standard input and output, profiling, abort and exit.
standalone	

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作成、File→New→Application Project



New Project

Application Project
Create a managed make application project.

Project name:

☒ Use default location
Location:

Choose file system:

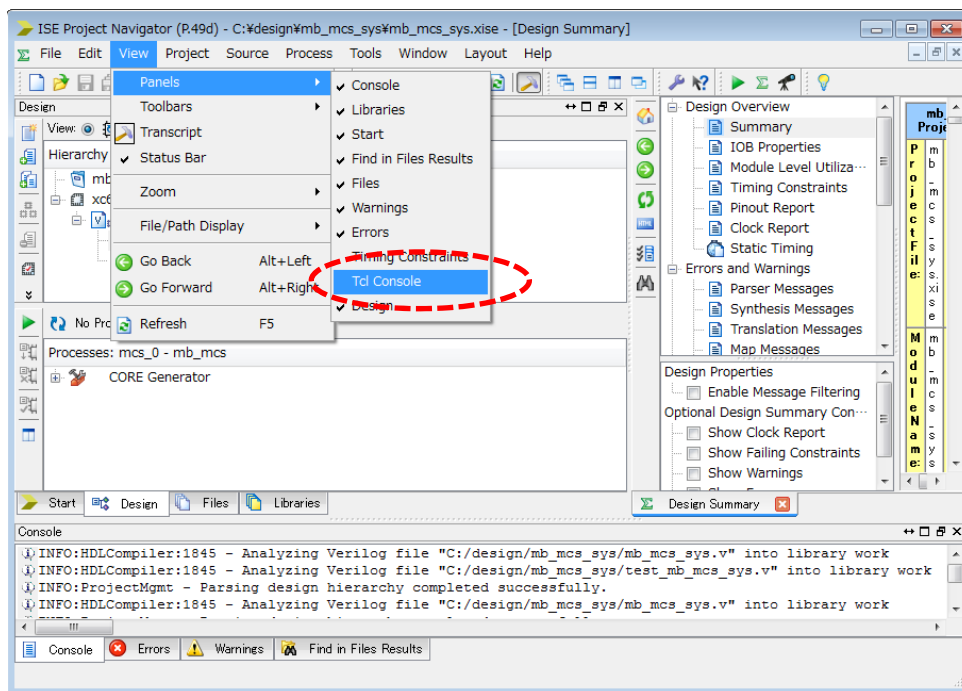
Target Hardware
Hardware Platform:
Processor:

Target Software
OS Platform:
Language: ☒ C ☐ C++
Board Support Package: ☒ Create New
☐ Use existing

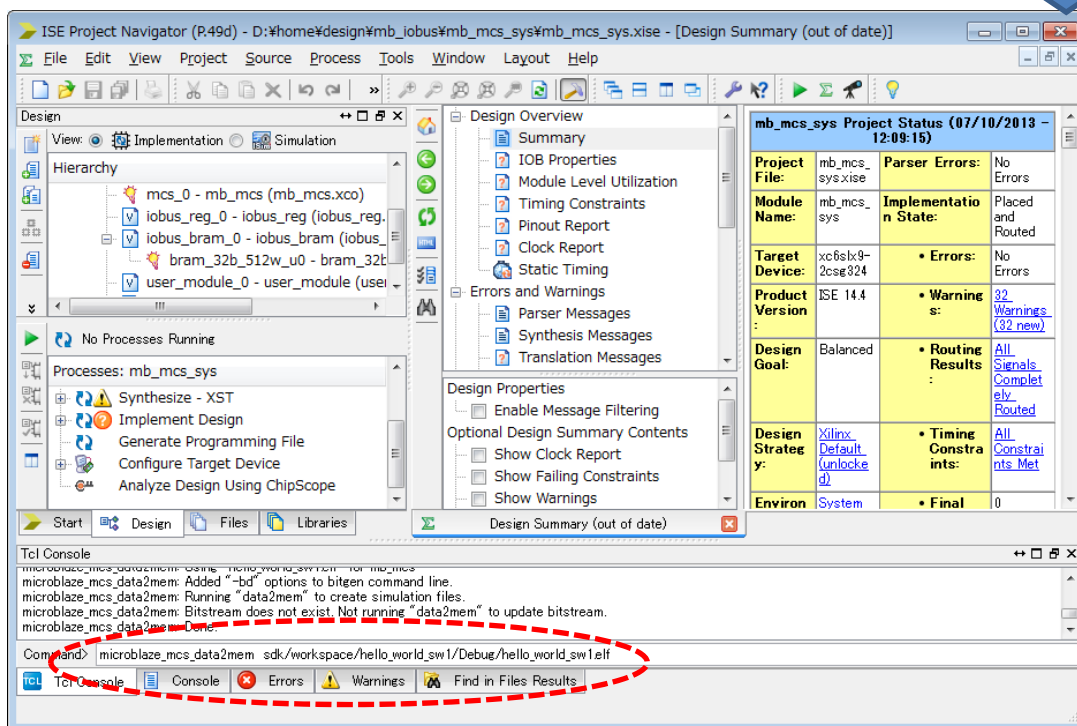
ソフトウェアプロジェクト名指定:hello_world_sw1

Next をクリック





Project Navigator に戻って、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

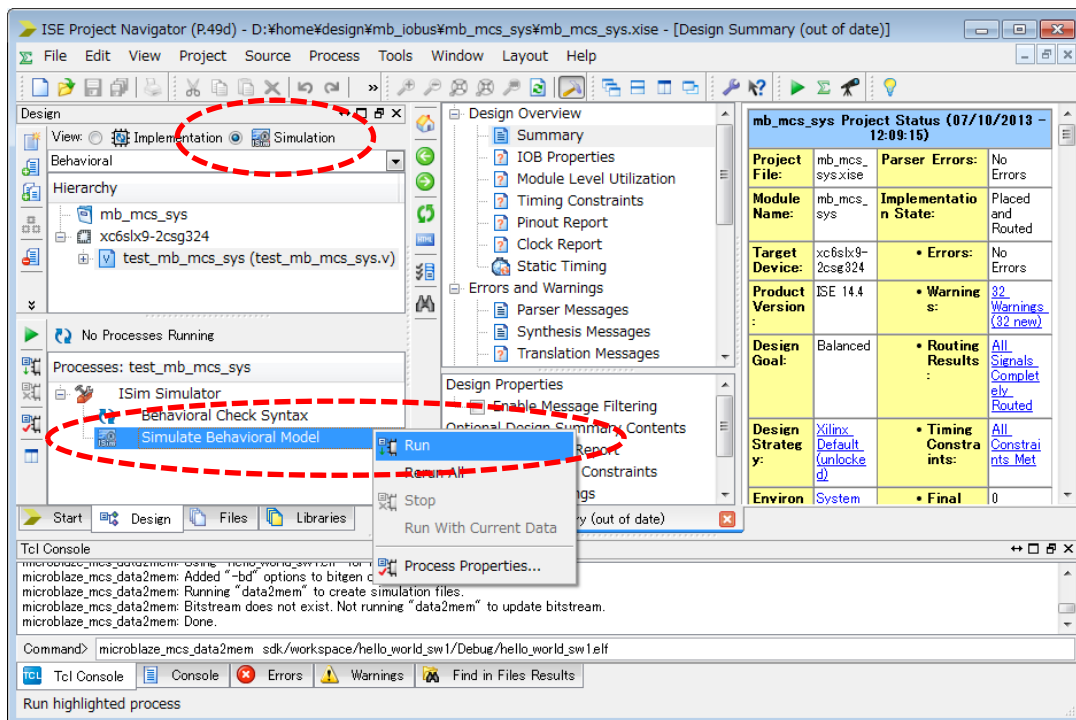


Tcl コマンドを使ってソフトウェア実行ファイルを MicroBlazeMCS のメモリの初期値定義ファイルに変換する。

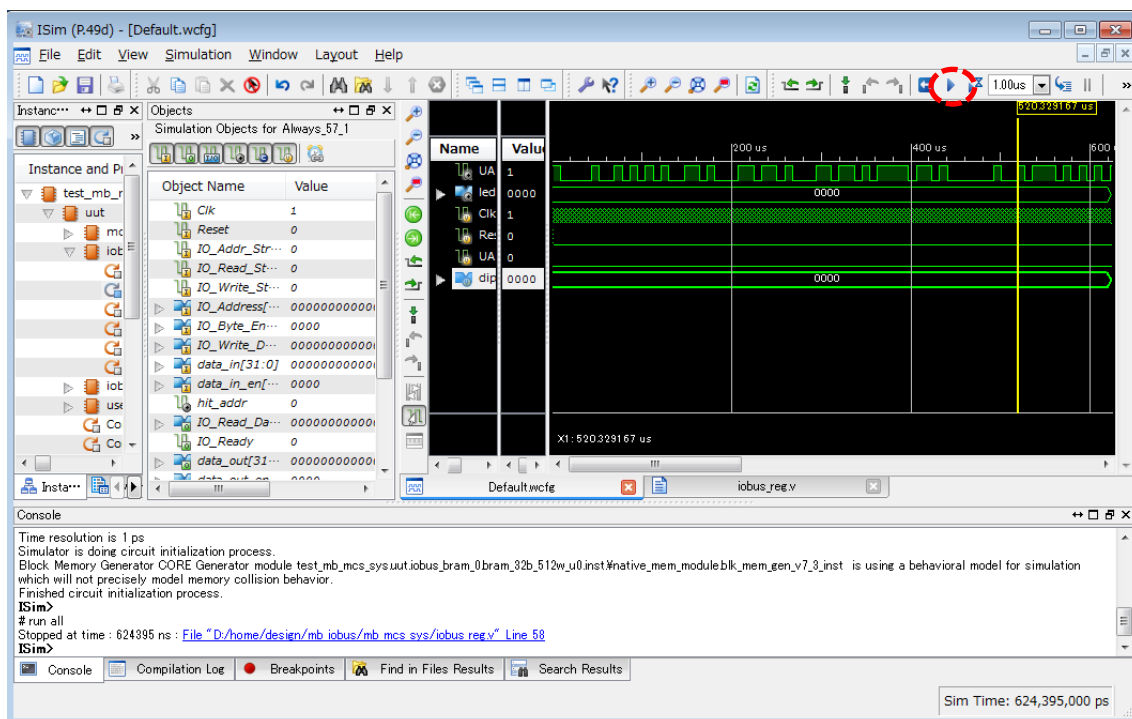
Tcl コマンド

source ipcore_dir/microblaze_mcs_setup.tcl [Enter]

microblaze_mcs_data2mem sdk/workspace/hello_world_sw1/Debug/hello_world_sw1.elf [Enter]



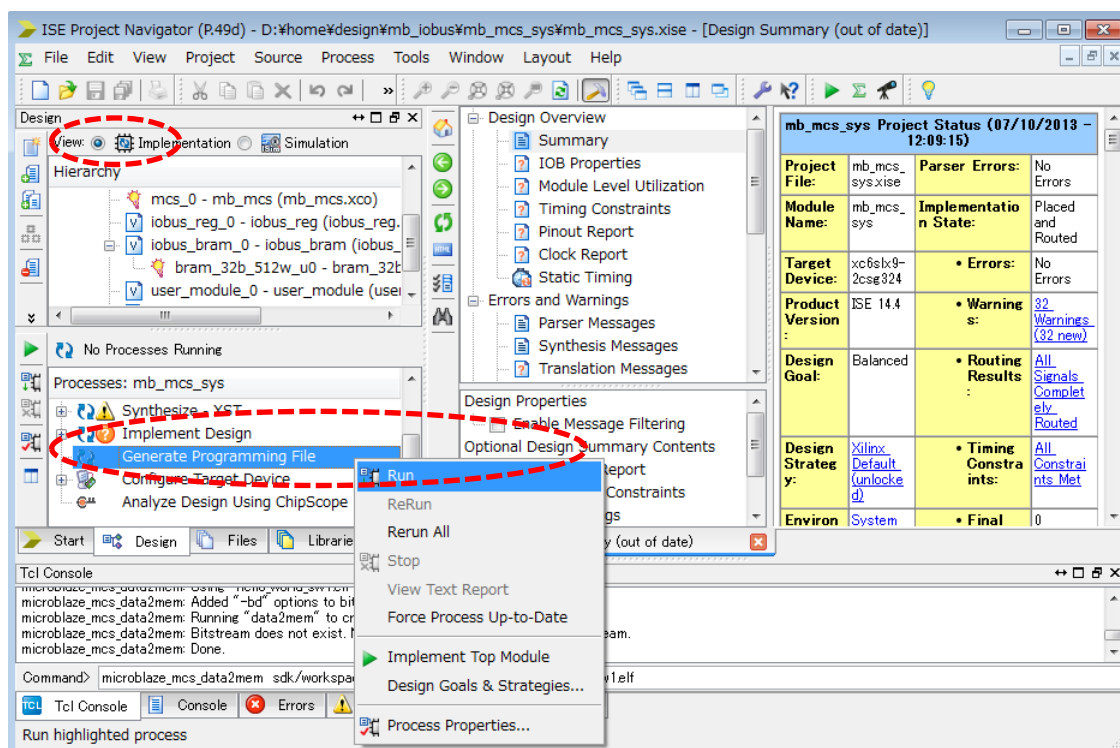
Isimの実行、Design→Simulationクリック、Hierarchy→test_mb_mcs_sys 選択、
ISim Simulator→Simulate Beahv... →Run 選択



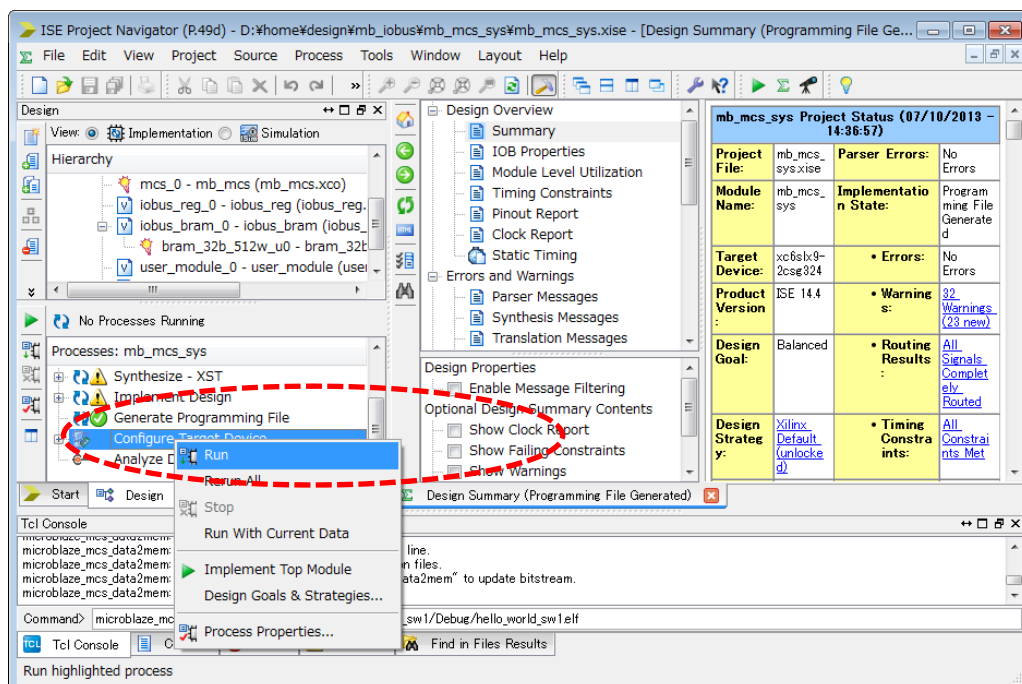
シミュレーションを最後まで実行



- ・ シミュレーションで波形確認したら FPGA で動作確認します。
- ・ 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 ポートに接続します。
- ・ PC でターミナルソフト立ち上げ、UART 用 USB に割り当てられたポート番号に 115200bps で接続します。

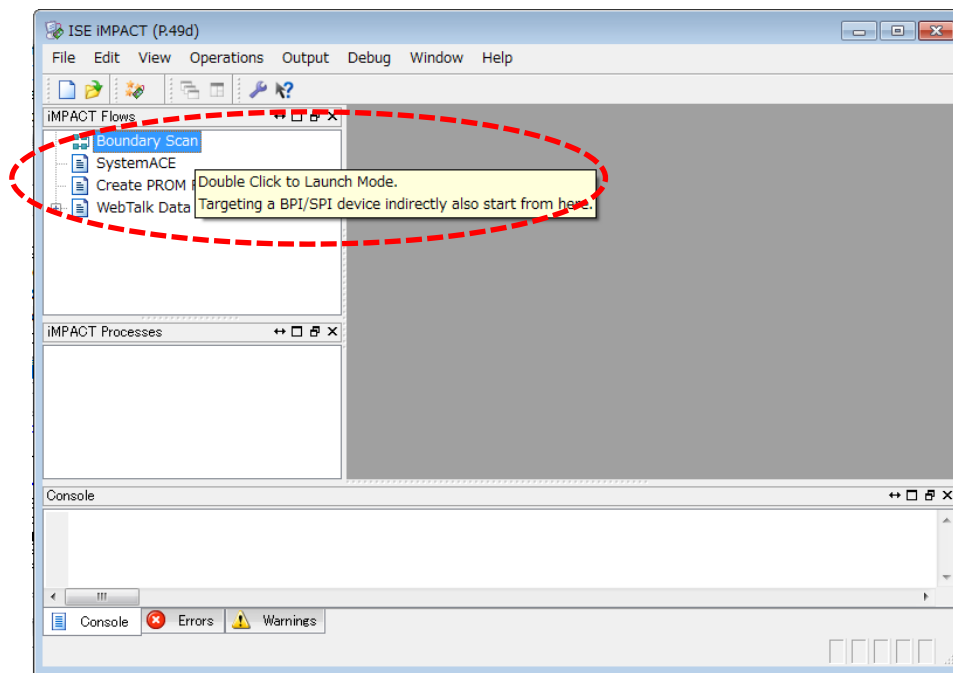


FPGA へ書き込む mb_mcs_sys.bit 作成

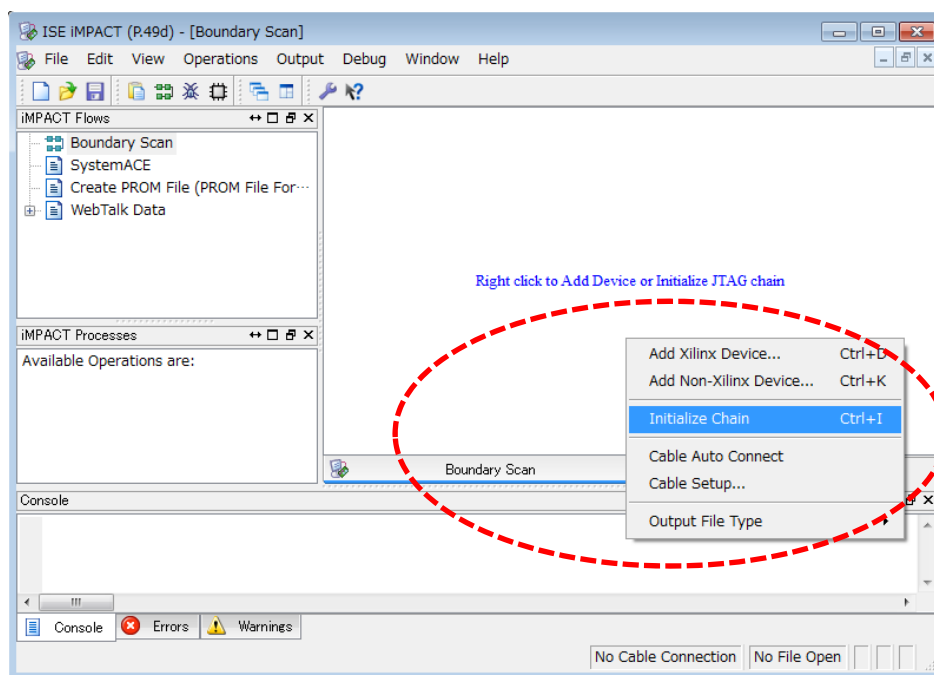


Project Navigator で iMPACT を起動



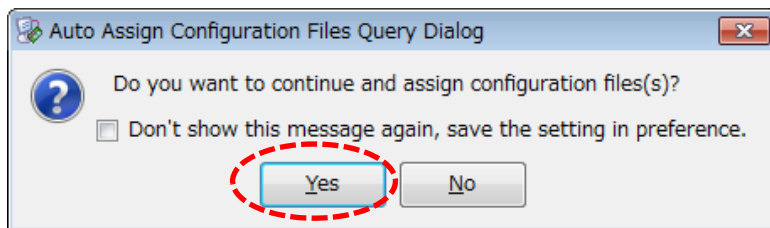


BoundaryScan モードにする

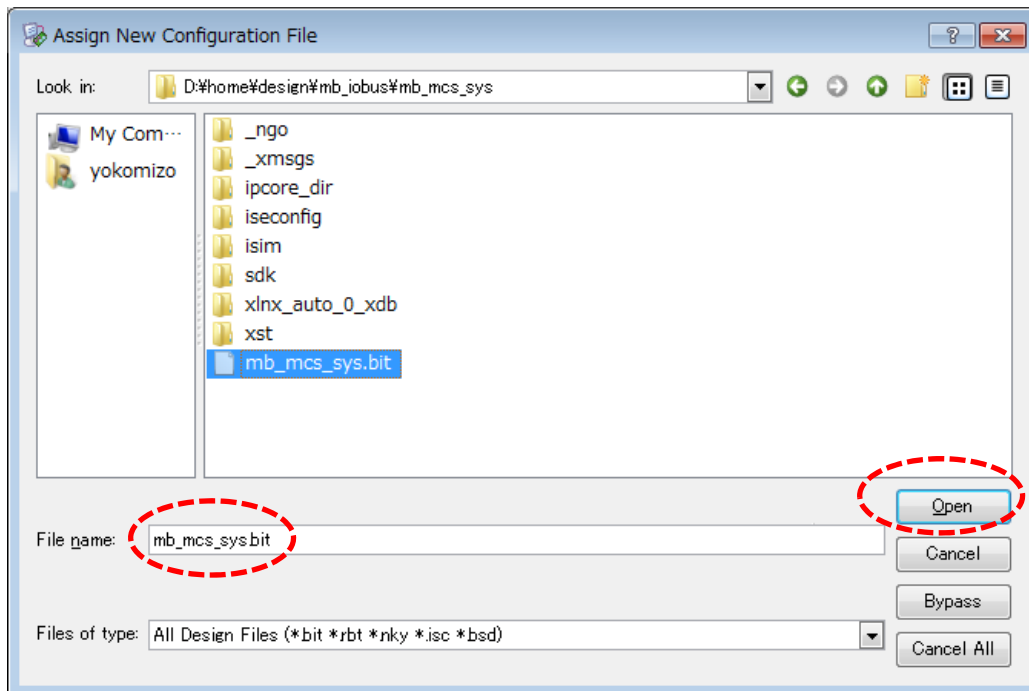


BoundaryScan のウィンドをマウス右ボタン押して、Initialize Chain 選択して FPAG を検出する

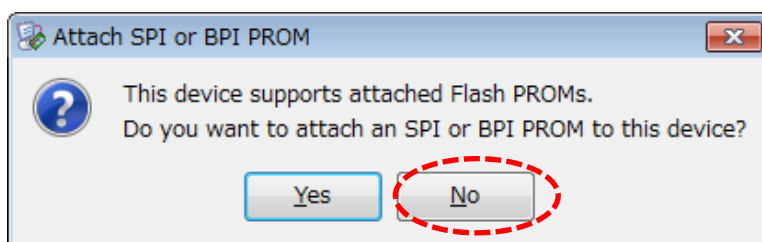




Yes で FPGA に書き込むファイルを指定する

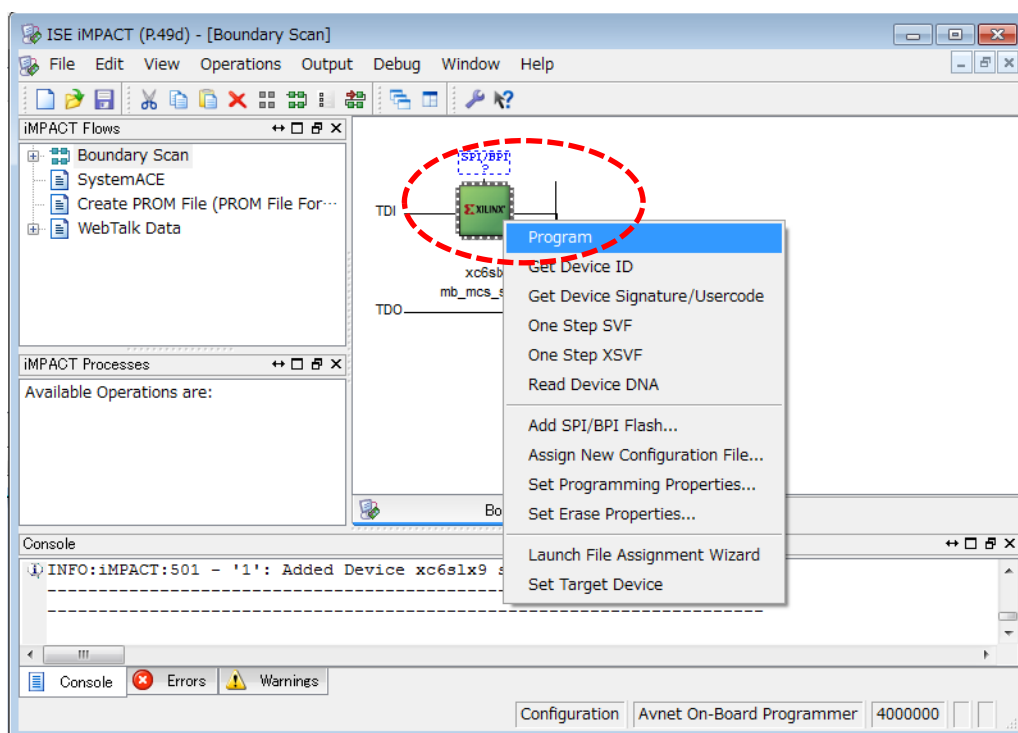
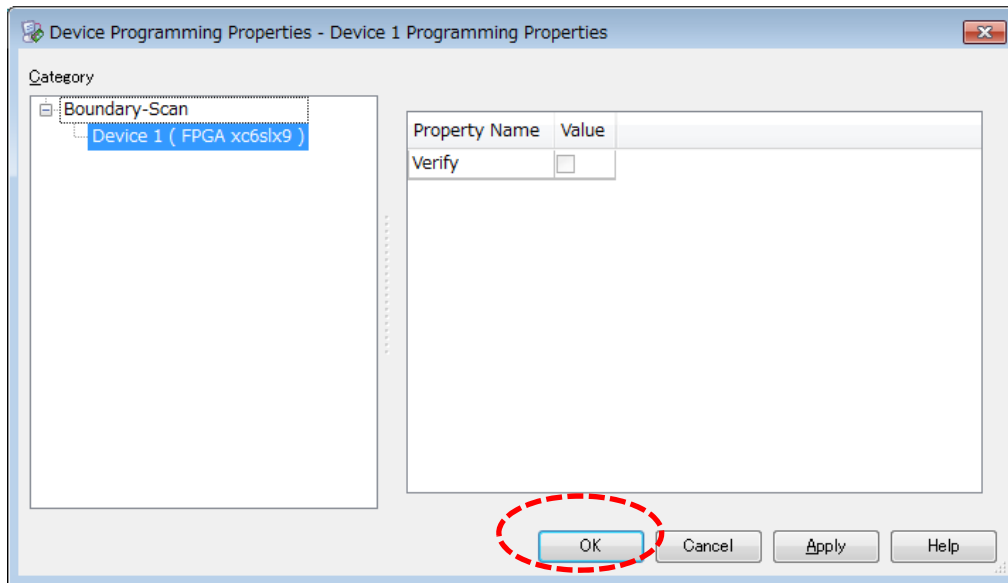


mb_mcs_sys.bit を指定



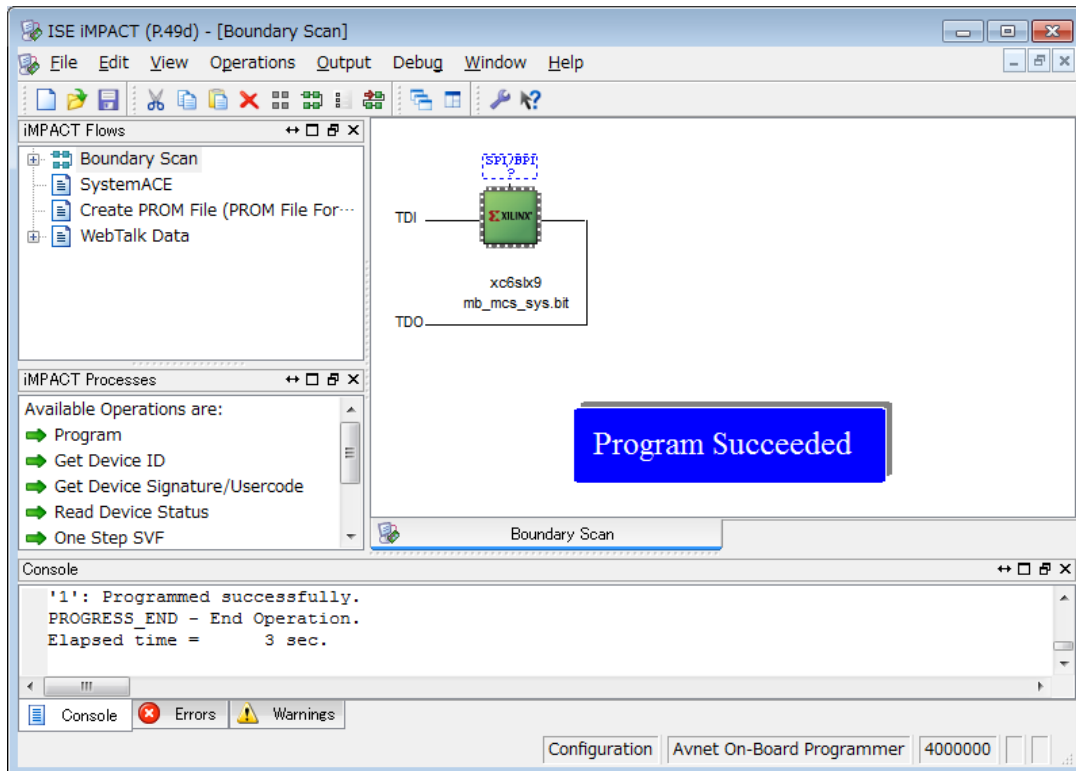
PROM データは使わないので No を選択



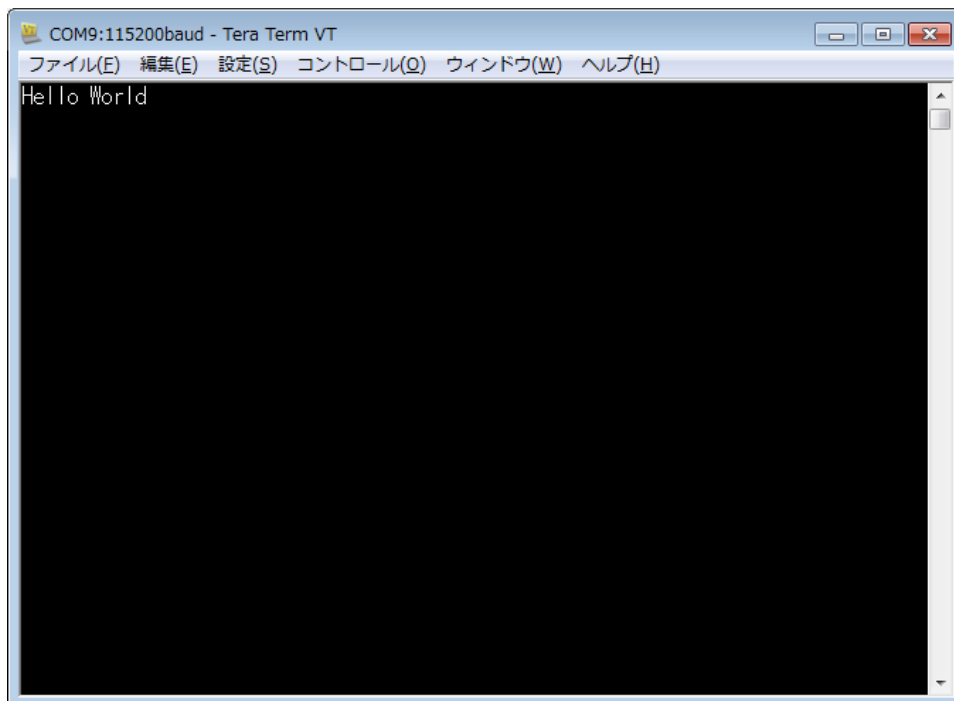


FPGA へのプログラミング実行、デバイス上でマウス右ボタンを押して Program 選択



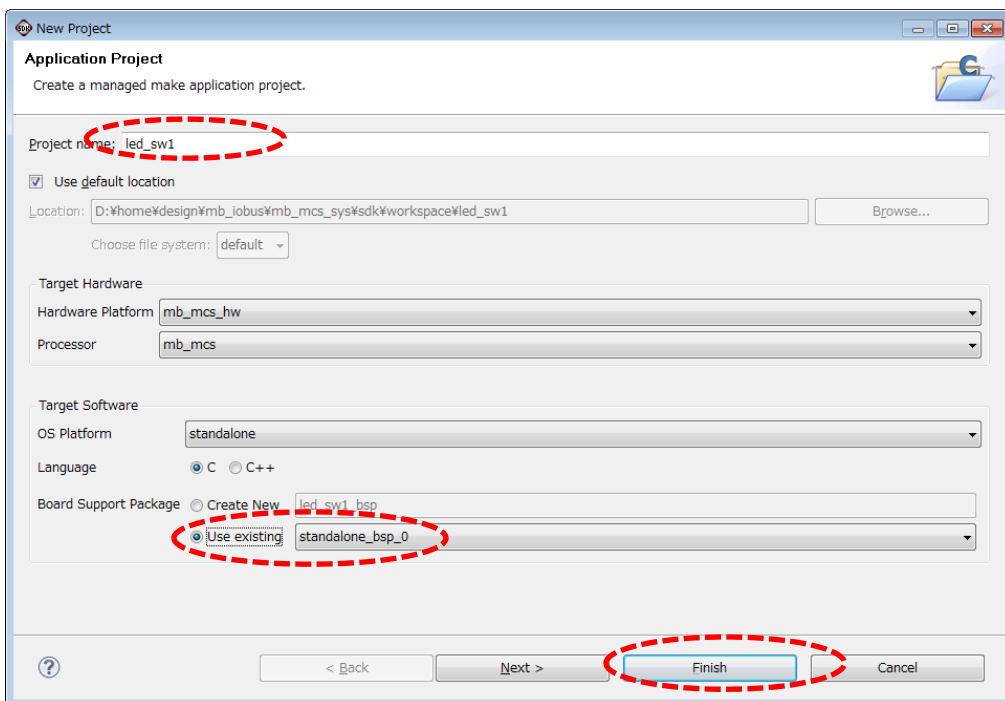
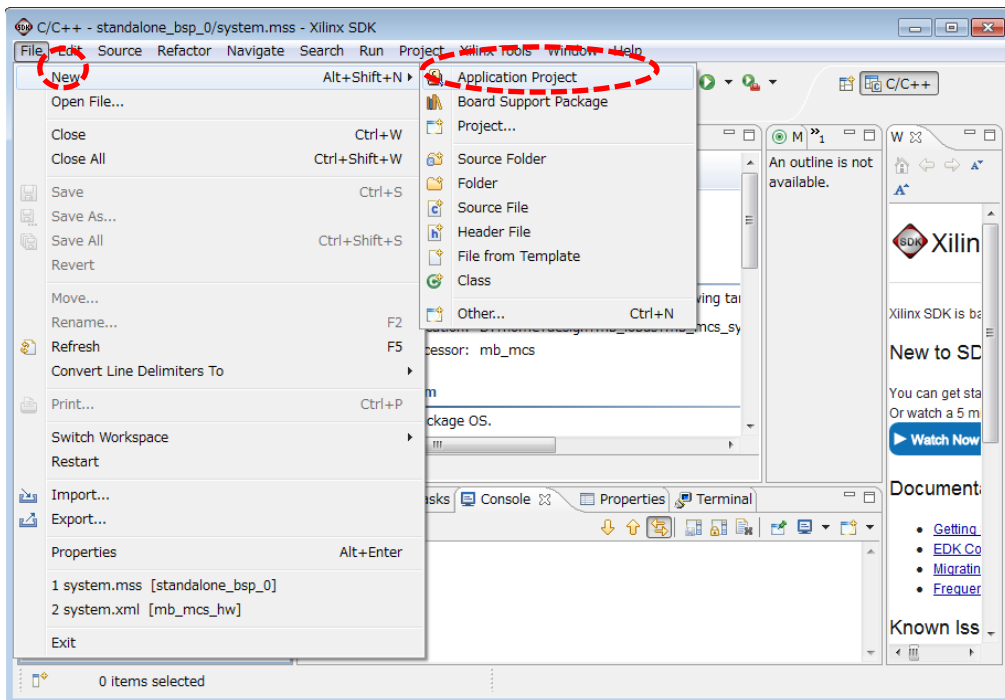


Program Succeeded と表示されればプログラミング完了、LX9 マイクロボードの動作を確認する。



ターミナルソフトに Hello World と表示されれば OK

- ・他のCソースを使って動作確認する方法を紹介します。
- ・新しいソフトウェアプロジェクトを作ります。

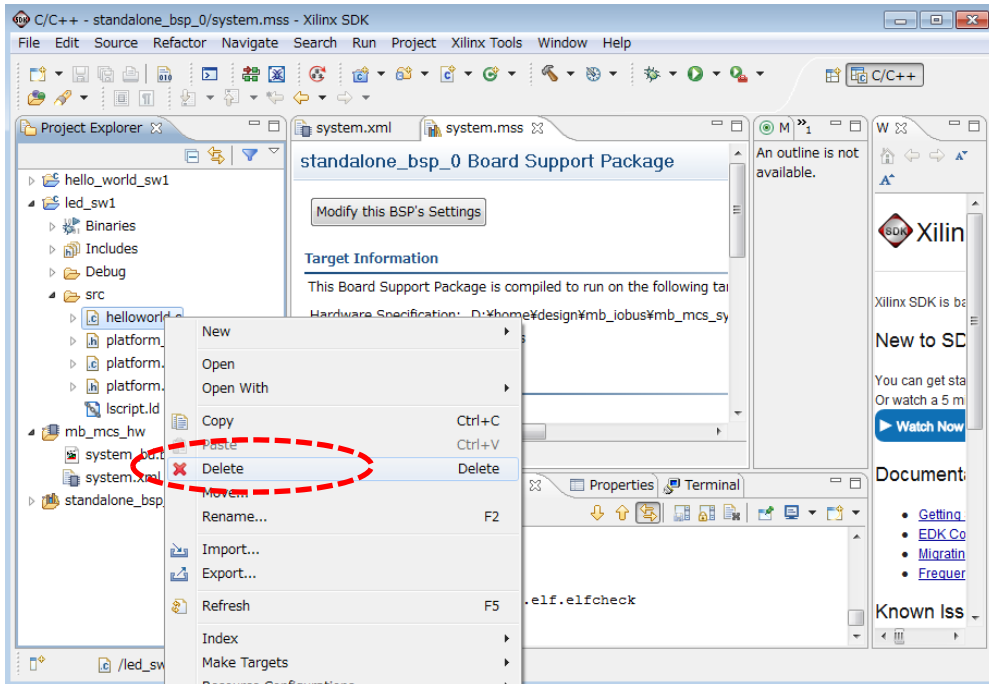


新しいソフトウェアプロジェクト名を設定(例.led_sw1)

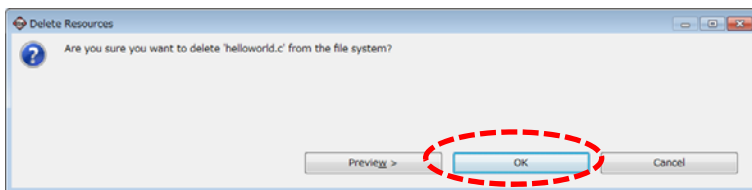
Board Support Package に standalone_bsp_0 を設定

Finsh をクリックする

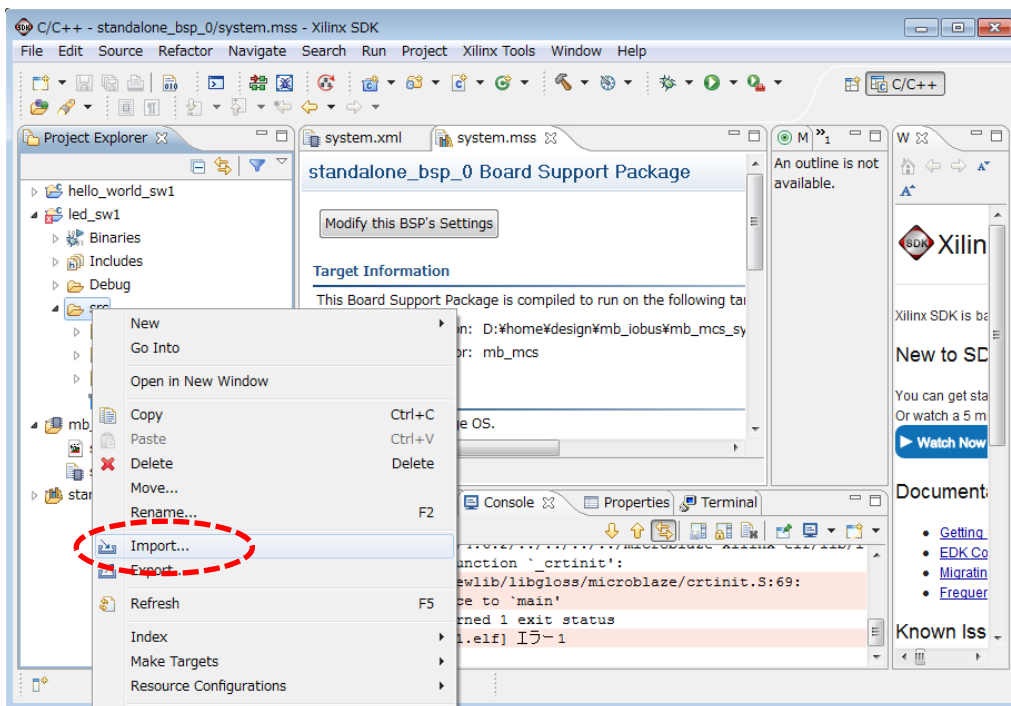




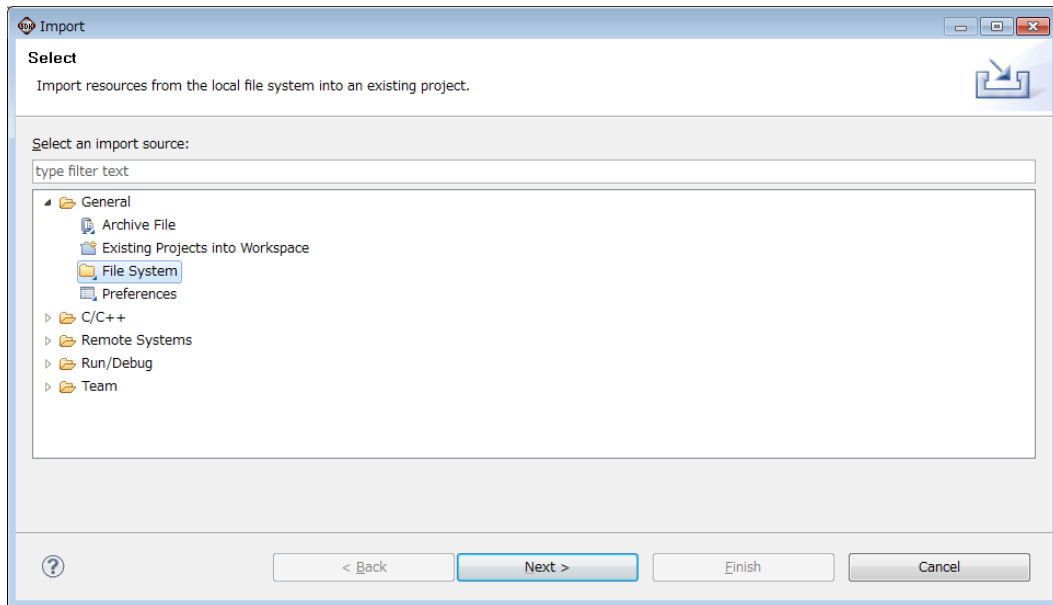
helloworld.c の上にカーソルを合わせ右ボタンで delete を選択して、削除



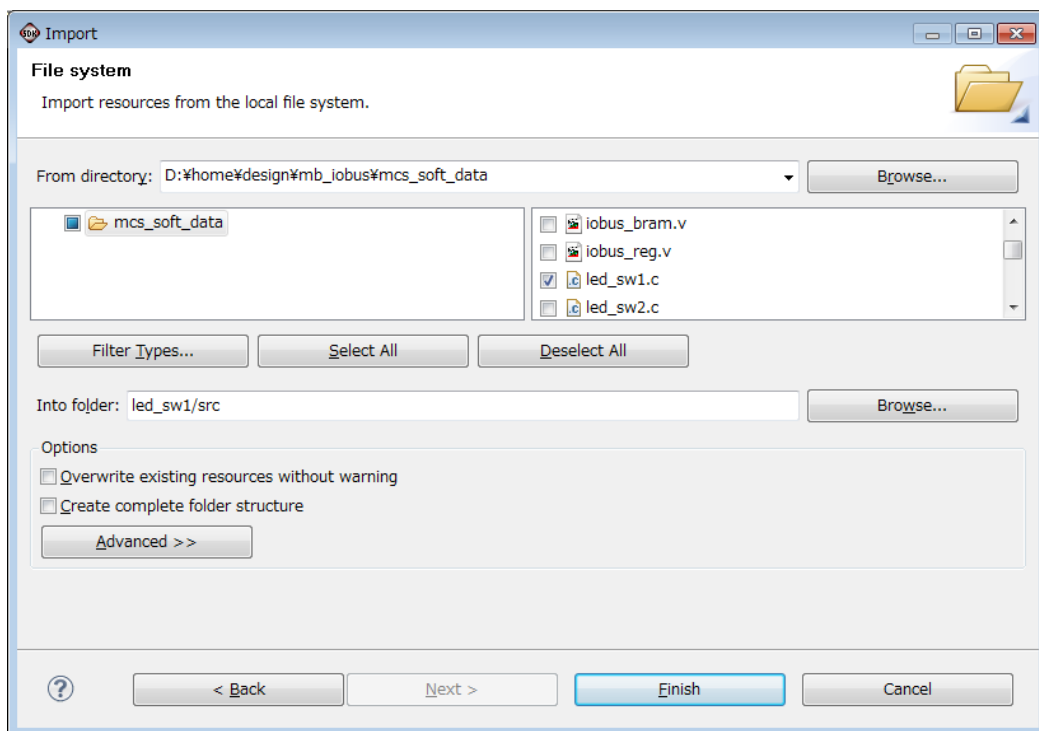
OK をクリック



src の上にカーソルを合わせて、右ボタンを押して Import を選択

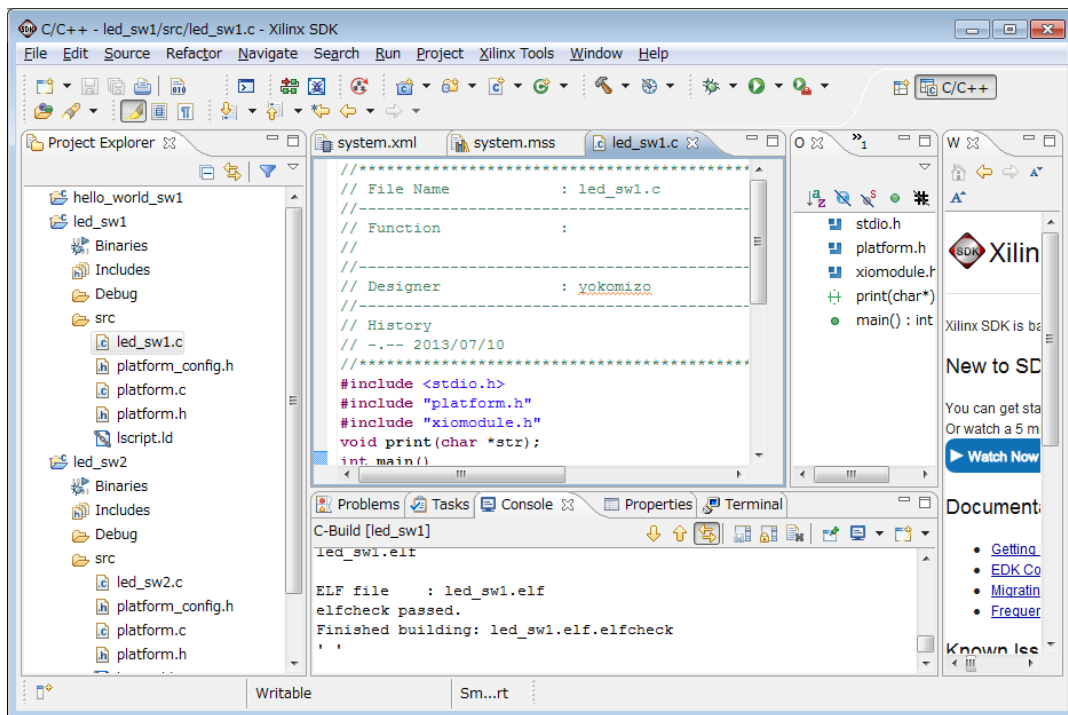


General→File System 選択

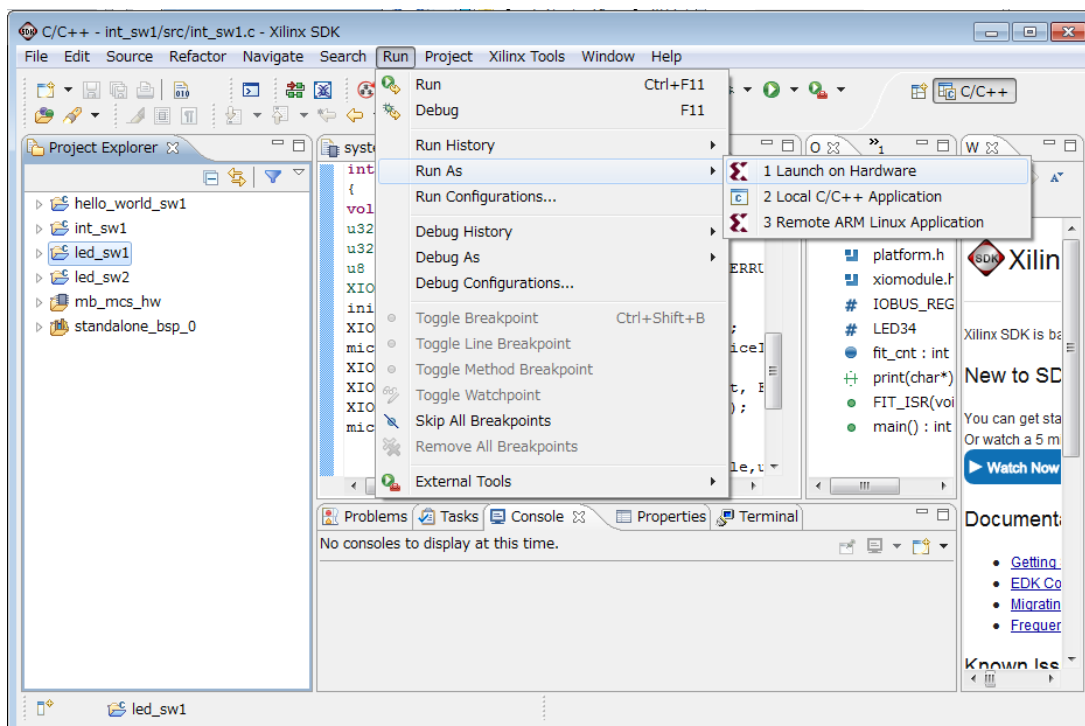


解凍データにある led_sw1.c を選択して、Finish





実行ファイル(led_sw1.elf)が作成されます。



led_sw1 を選択して、Run→Run As → Launch on Hardware を選択すると、実行される。
LED が点灯していれば動作 OK です。