

MicroBlazeMCS_I0 バス用設計データの使い方

ファイル一覧

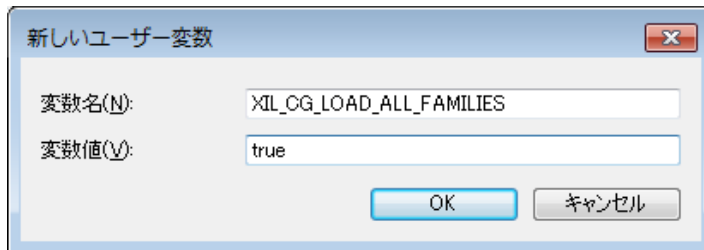
- ・ mb_mcs_sys.v トップ回路
- ・ iobus_reg.v 設定レジスタ回路
- ・ iobus_bram.v ブロック RAM インターフェース回路
- ・ user_module.v デバック回路
- ・ test_mcs_sys.v テストベンチ
- ・ mb_mcs_sys.ucf ピン配置指定
- ・ helloworld.c C ソースコード

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の microblaze_mcs/mb_iobus 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 が設定されてい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 を起動して以下の手順で設計プロジェクト作成

New Project Wizard

Create New Project

Specify project location and type.

Enter a name, locations, and comment for the project

Name:

Location: ...

Working Directory: ...

Description:

Select the type of top-level source for the project

Top-level source type:

プロジェクト名と設計フォルダ指定



New Project Wizard

Project Settings

Specify device and project properties.
Select the device and design flow for the project.

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"/>

[More Info](#) [Next](#) [Cancel](#)

デバイス指定 LX9 マイクロボードに合わせる



New Project Wizard

Project Summary

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

Project:

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

Device:

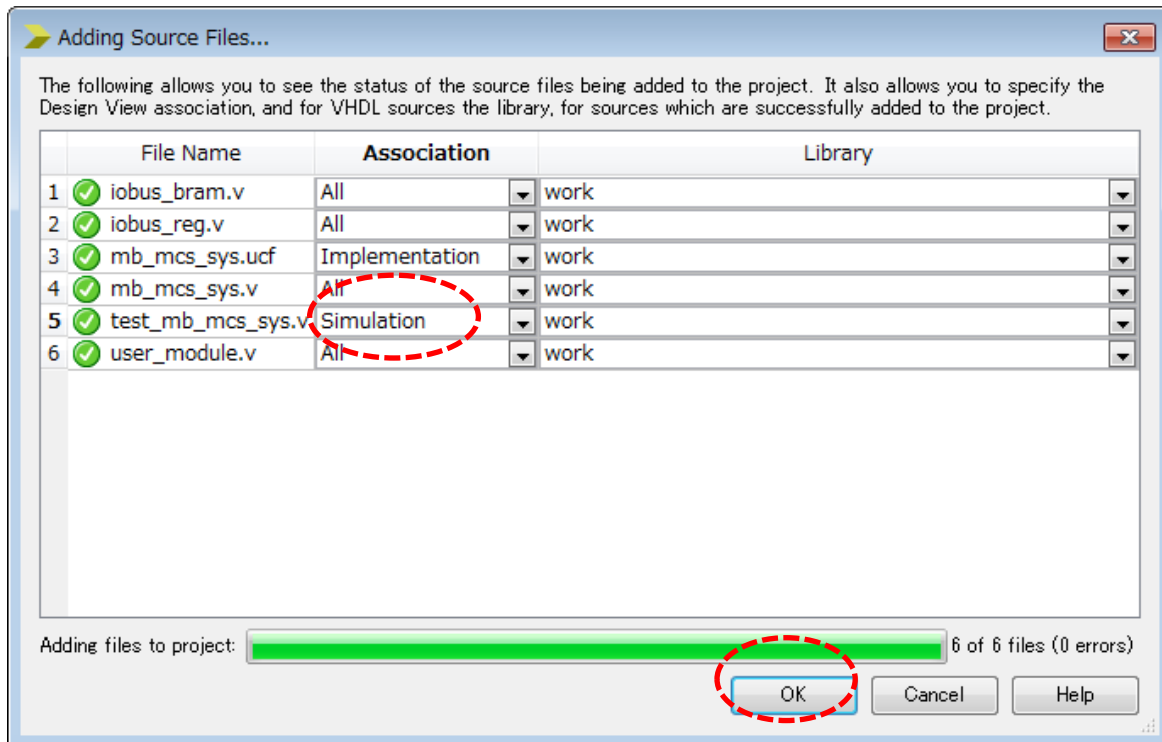
- Device Family: Spartan6
- Device: xc6slx9
- Package: csg324
- Speed: -2

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

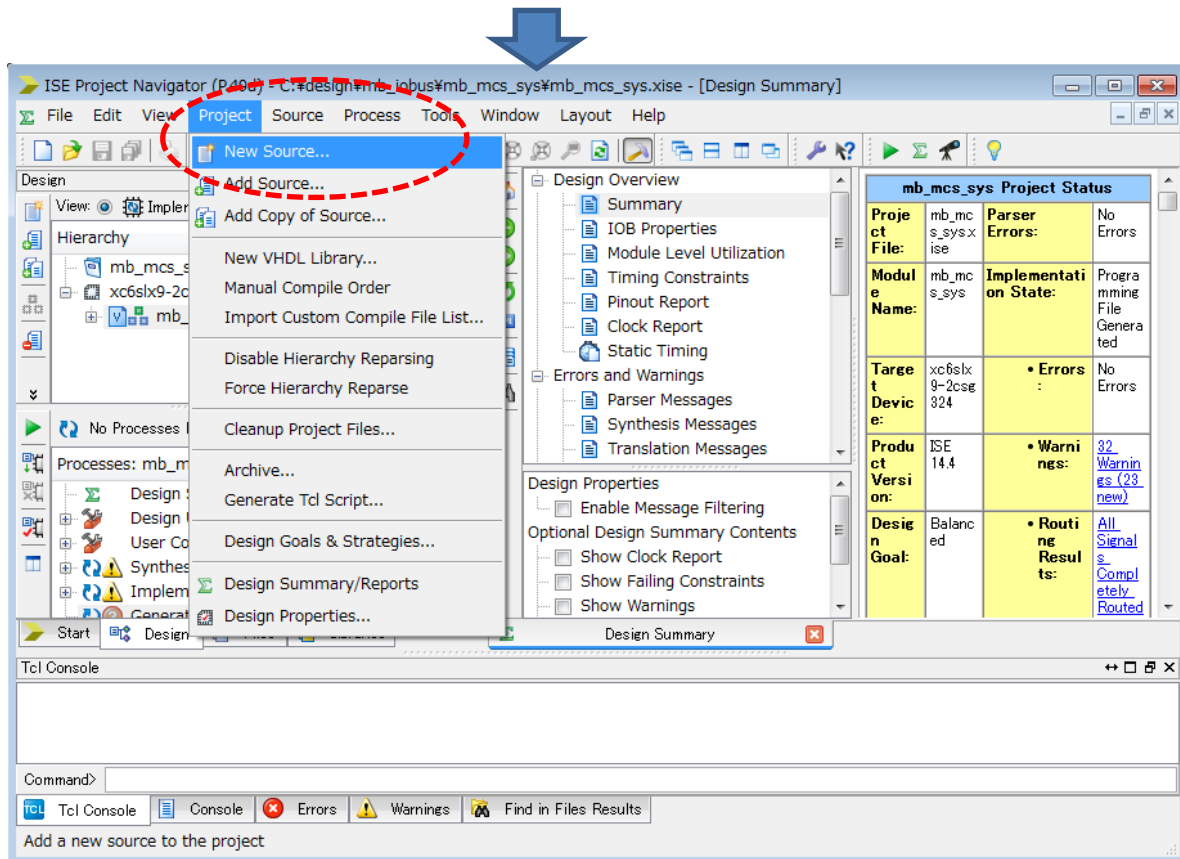
Message Filtering: disabled

[More Info](#) [Finish](#) [Cancel](#)

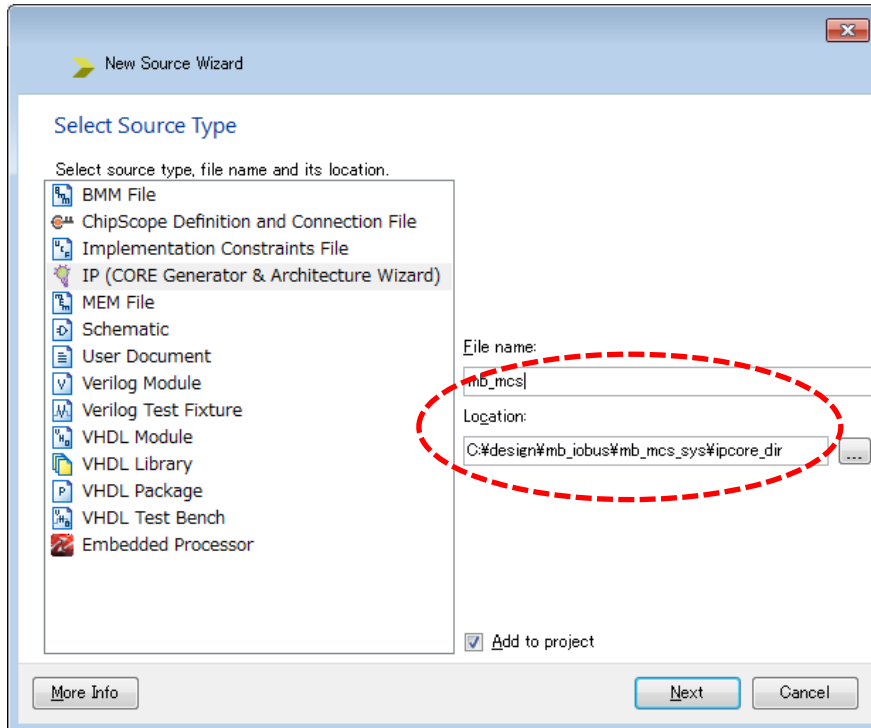




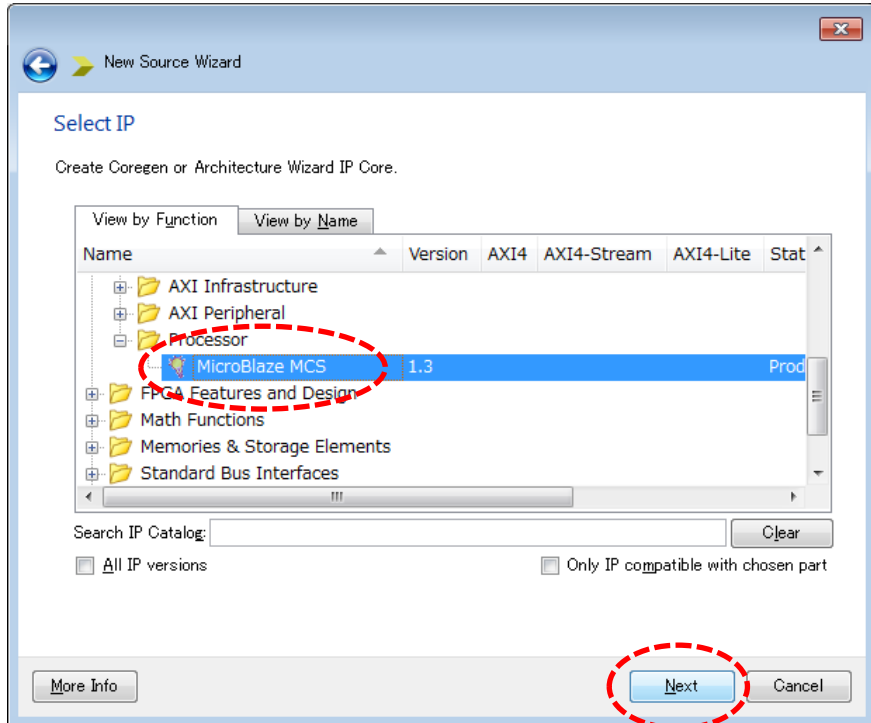
mb_mcs_sys.ucf はインプリメンテーションで使用するので Implementation を指定
test_mb_mcs_sys.v はテストベンチなので Simulation を指定



ProjectNavigator で MicroBlaze MCS を追加する

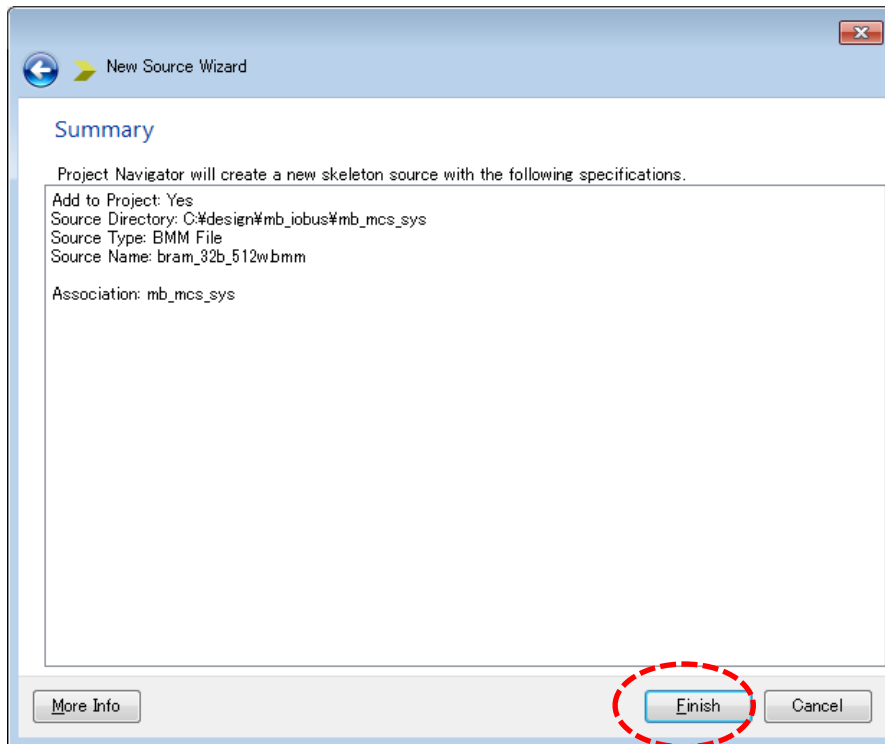


ファイル名を mb_mcs に指定

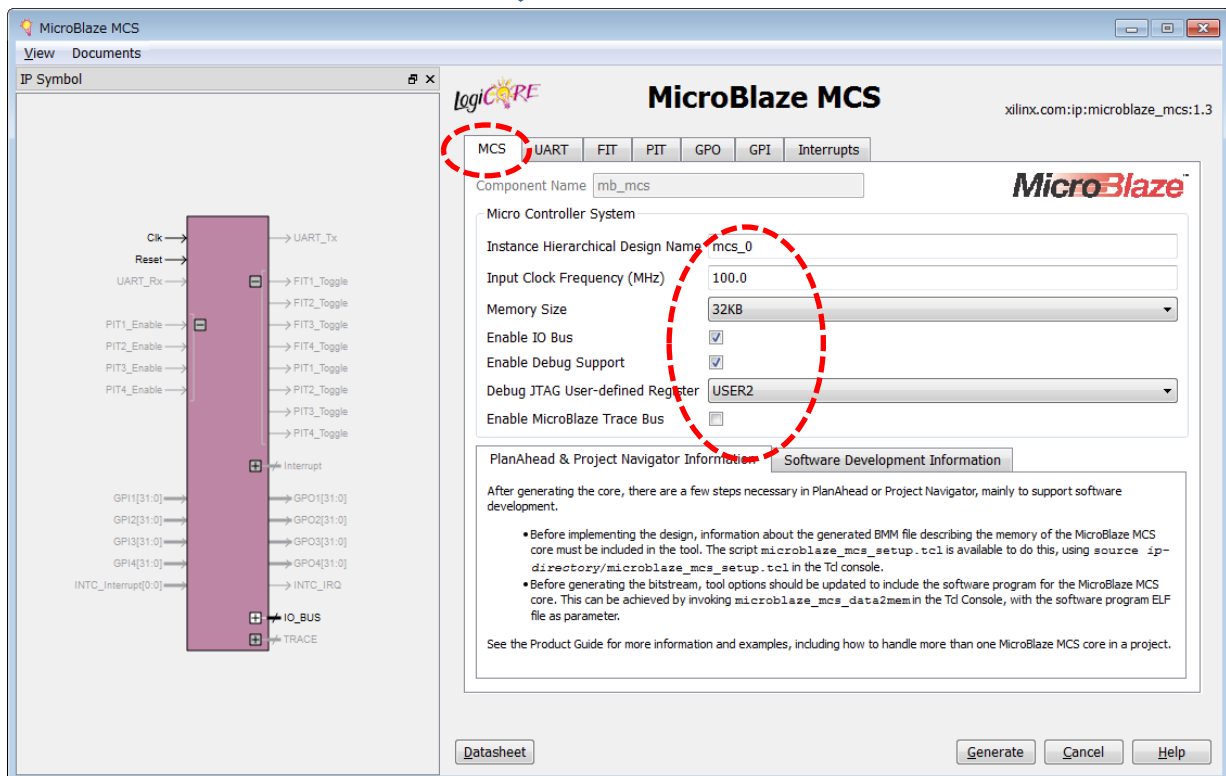


MicroBlaze MCS 選択



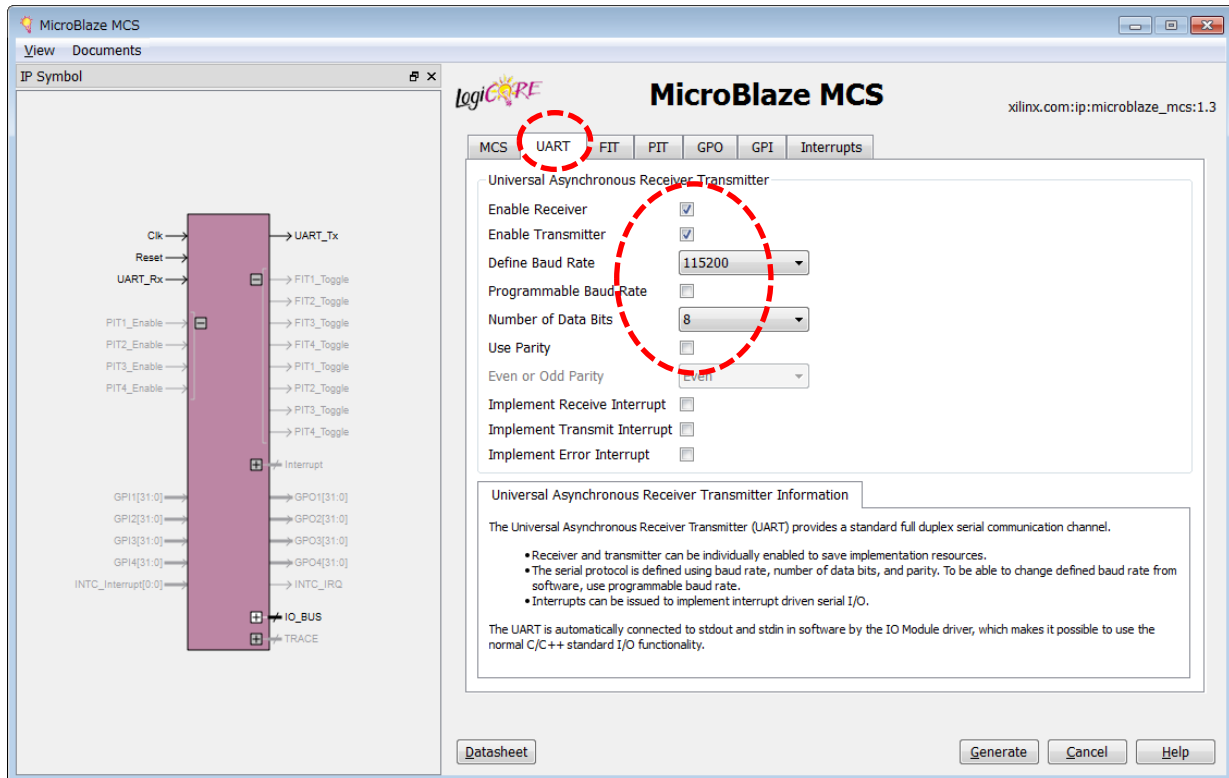


Finish で COREgenerator 起動

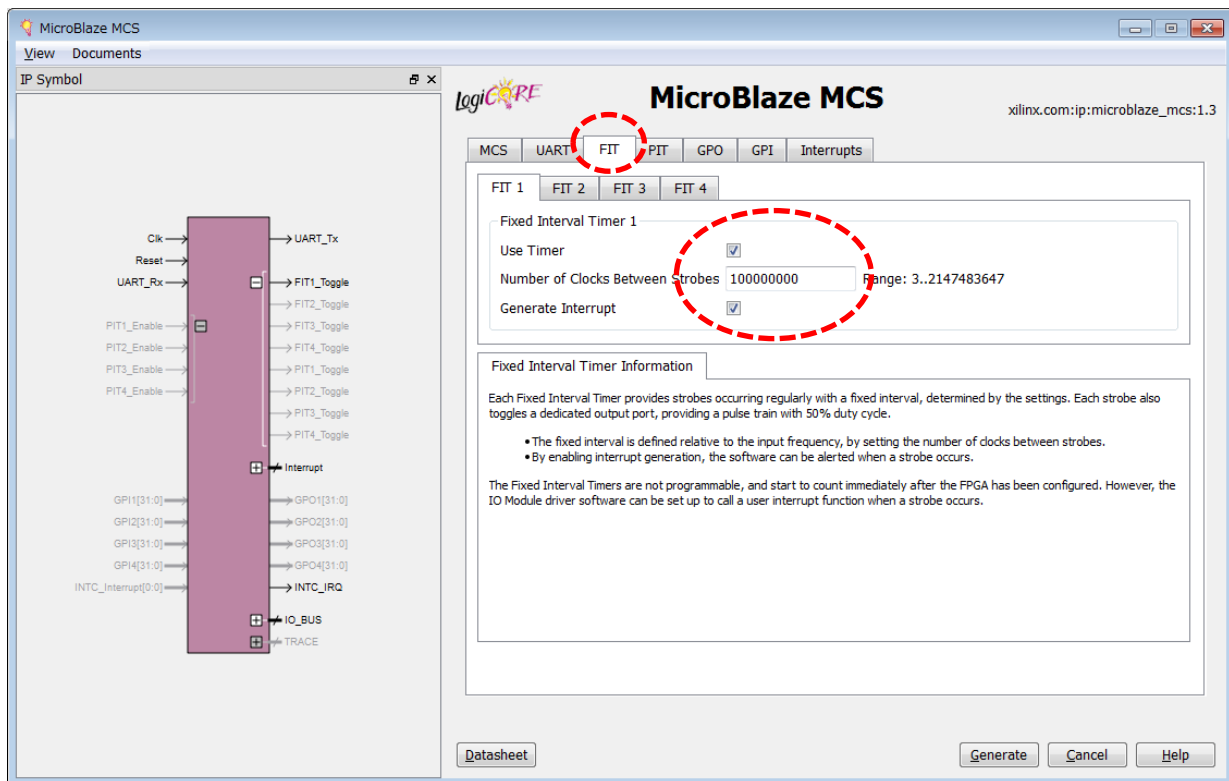


MicroBlaze MCS の基本設定



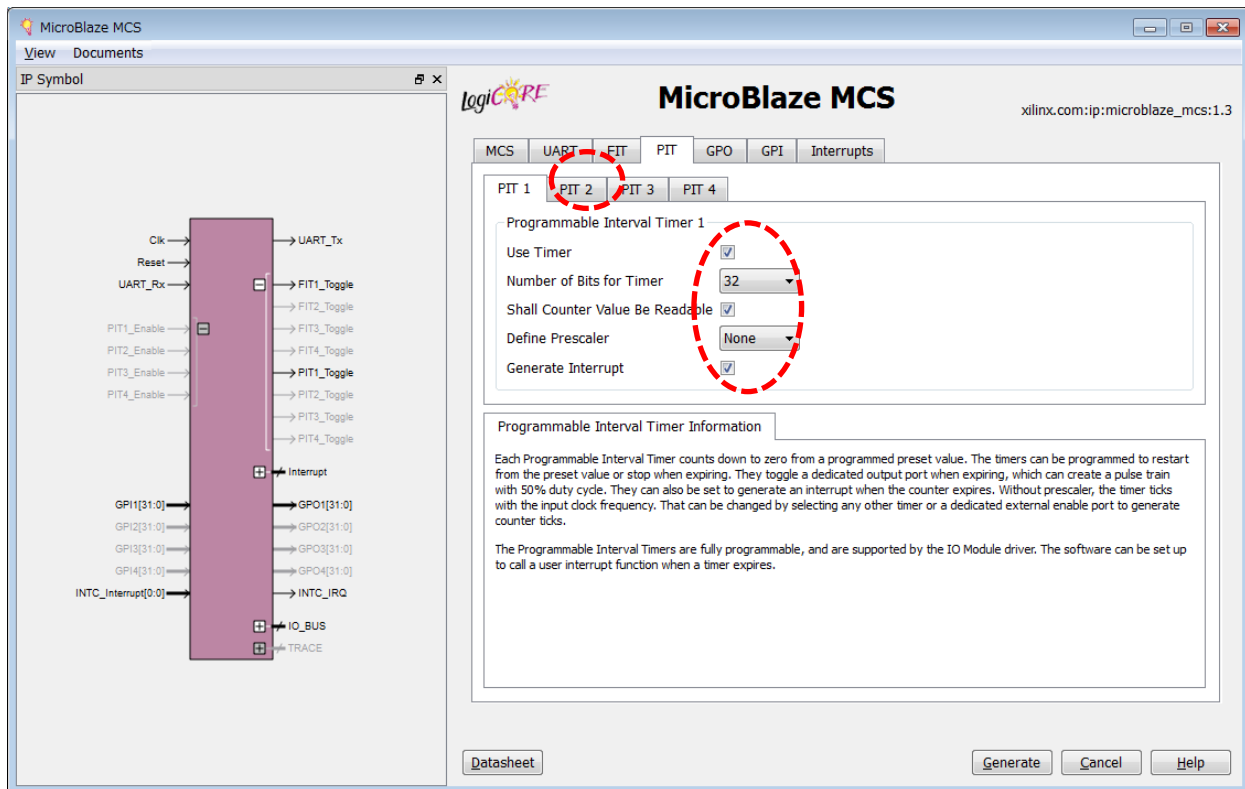


UART の設定

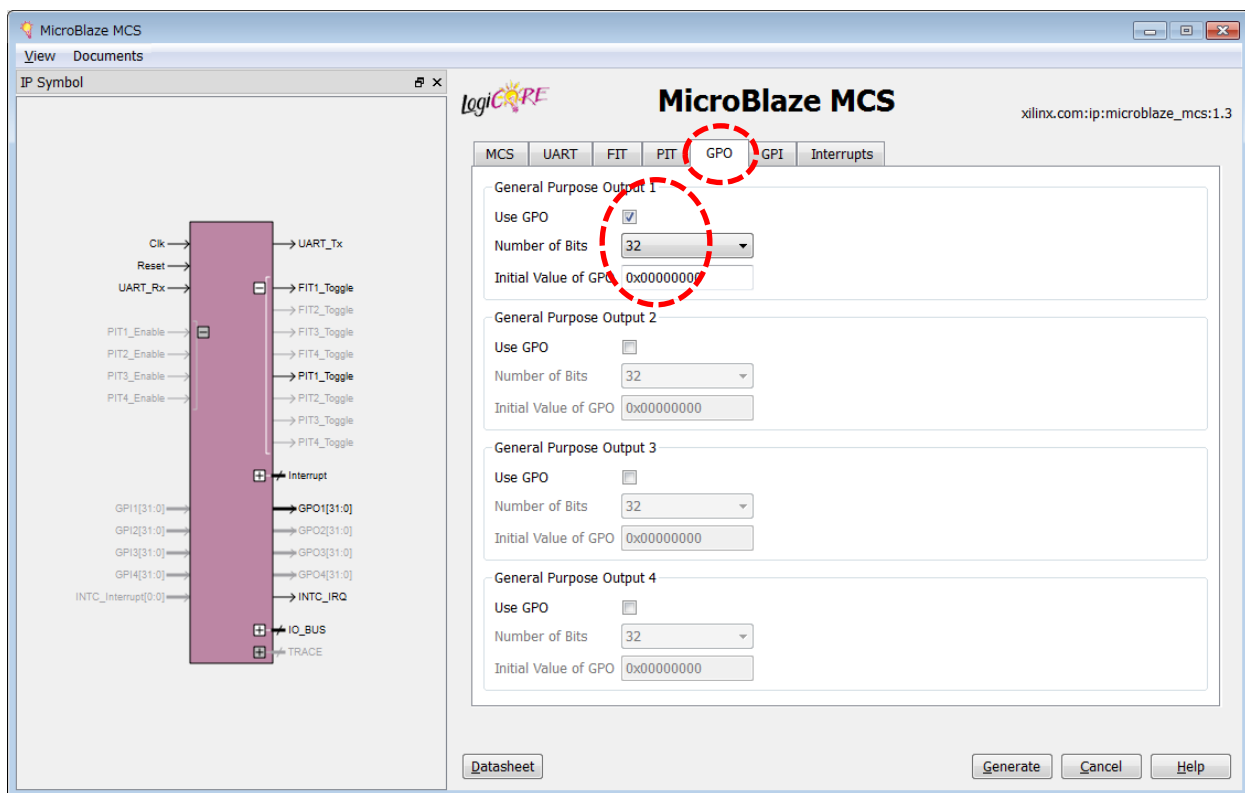


FIT の指定



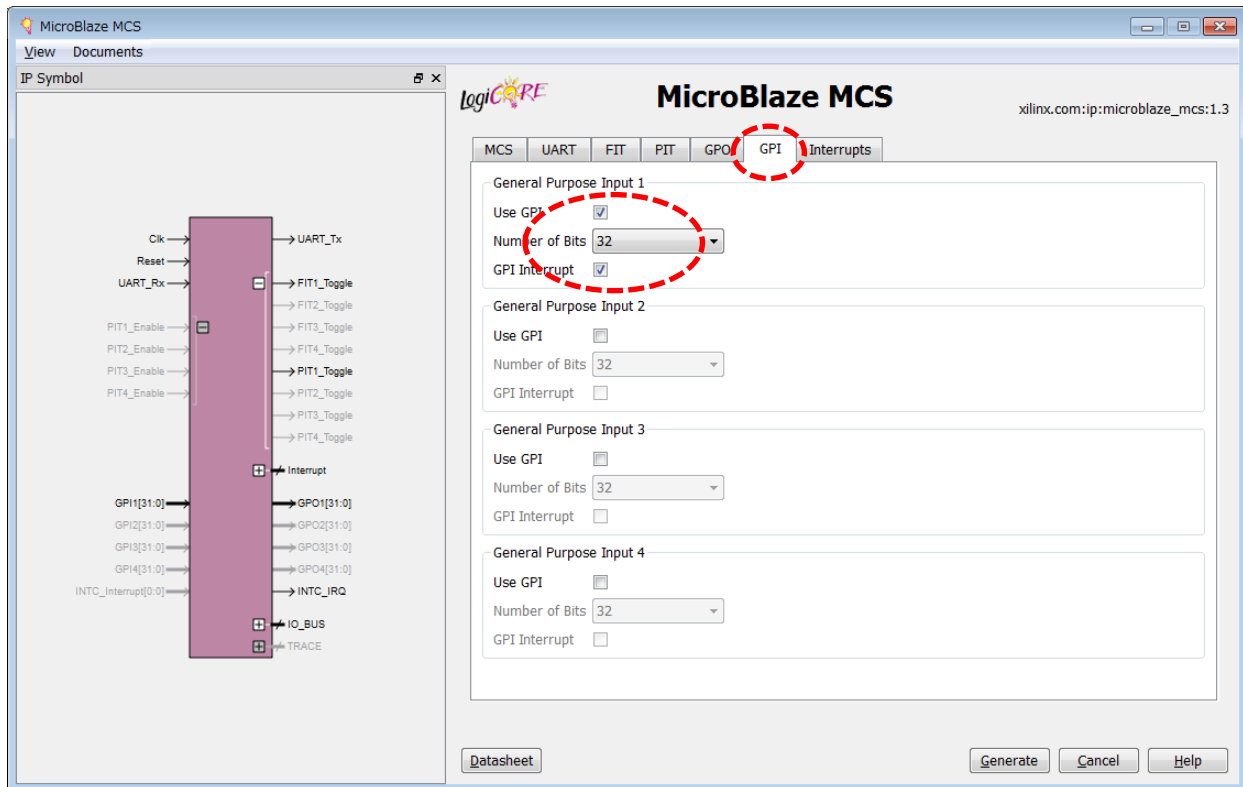


PIT の指定

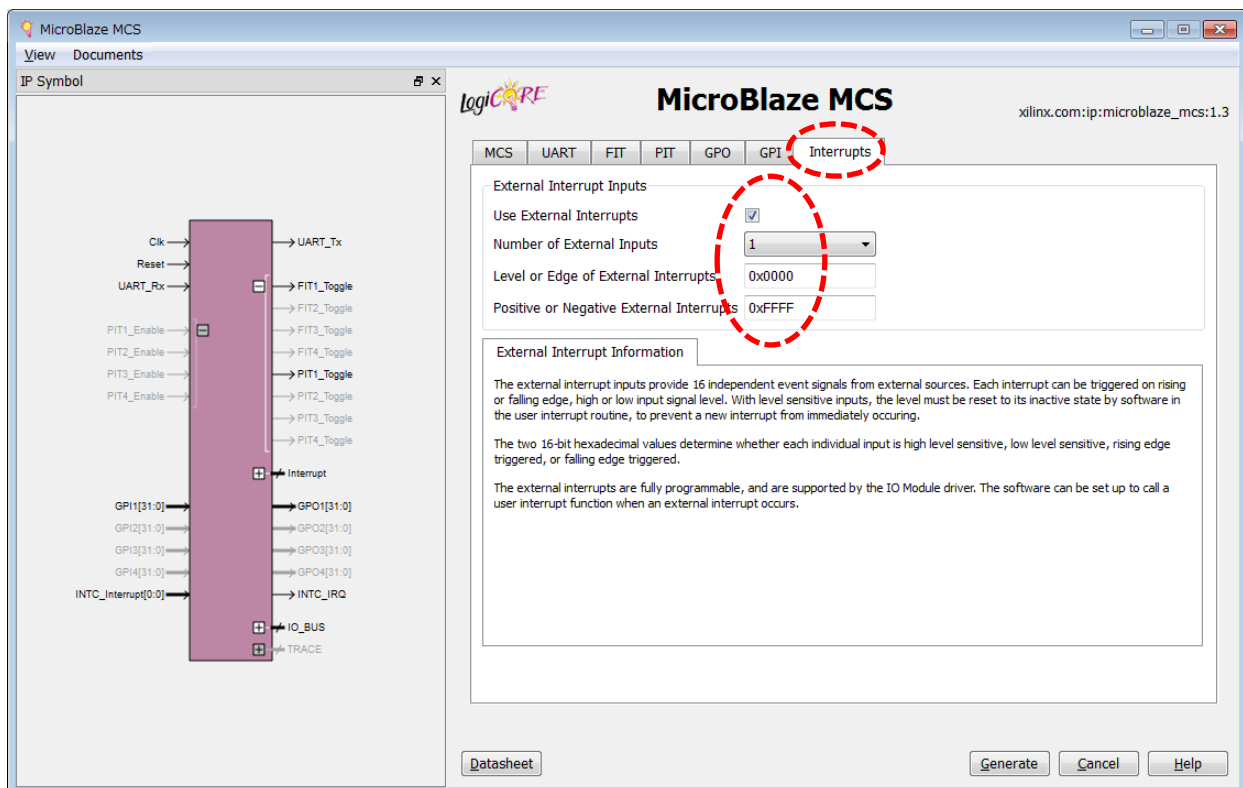


GPO の設定



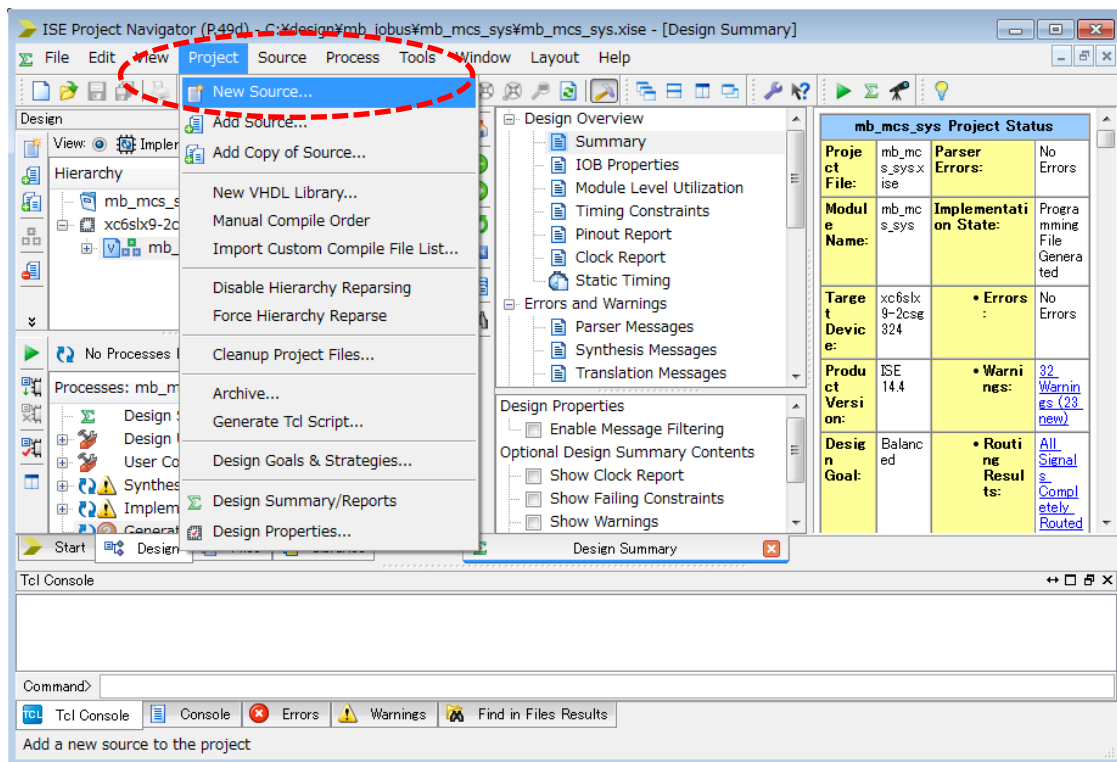


GPI の指定

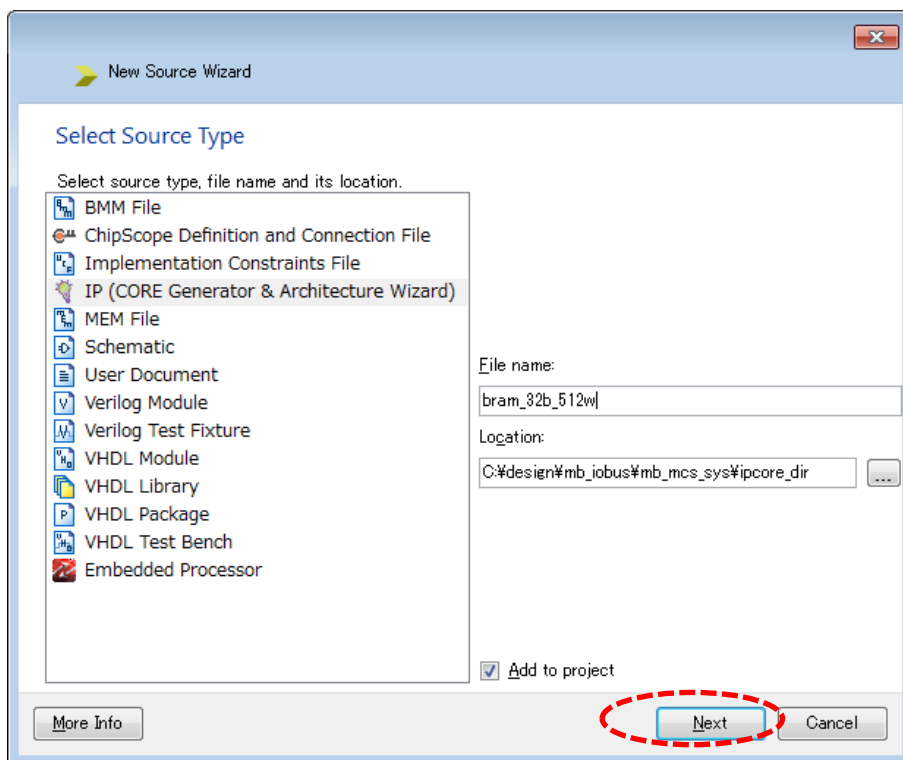


外部割り込みの設定後に Genrate をクリックすると MicroBlaze MCS がプロジェクトに追加される



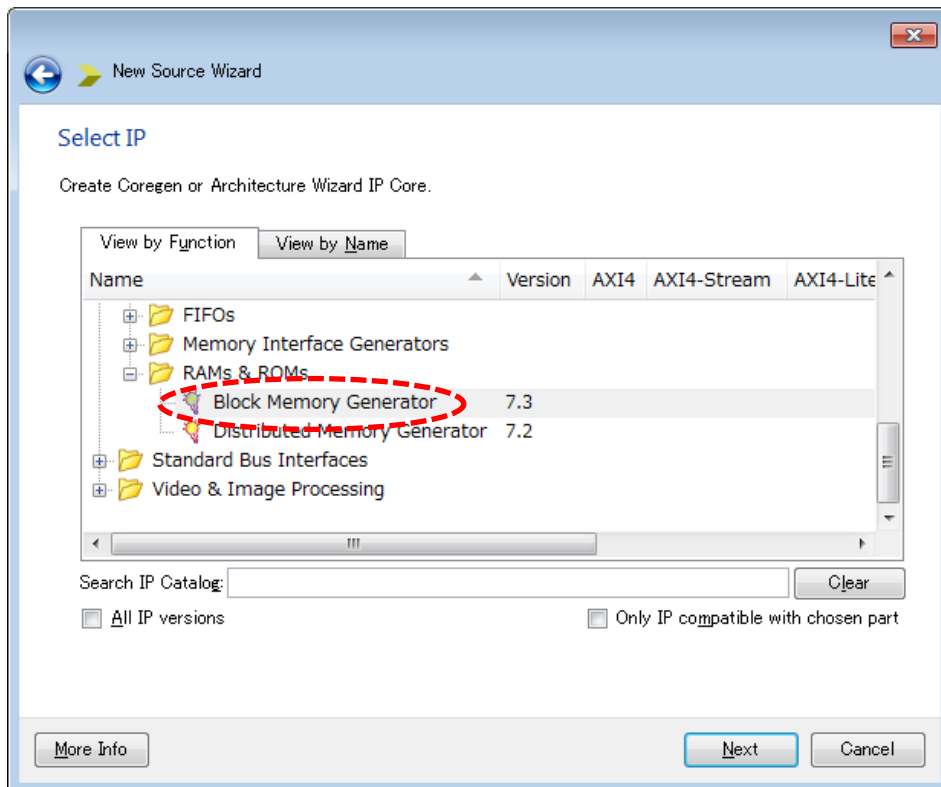


ブロック RAM の作成

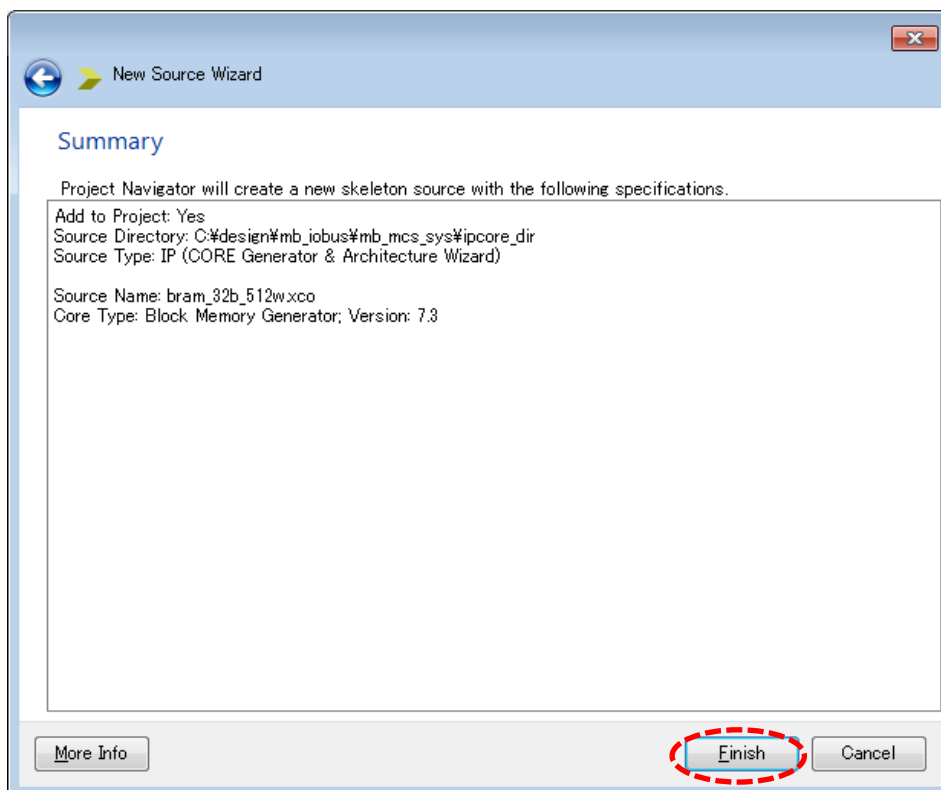


ファイル名指定 bram_32b_512w



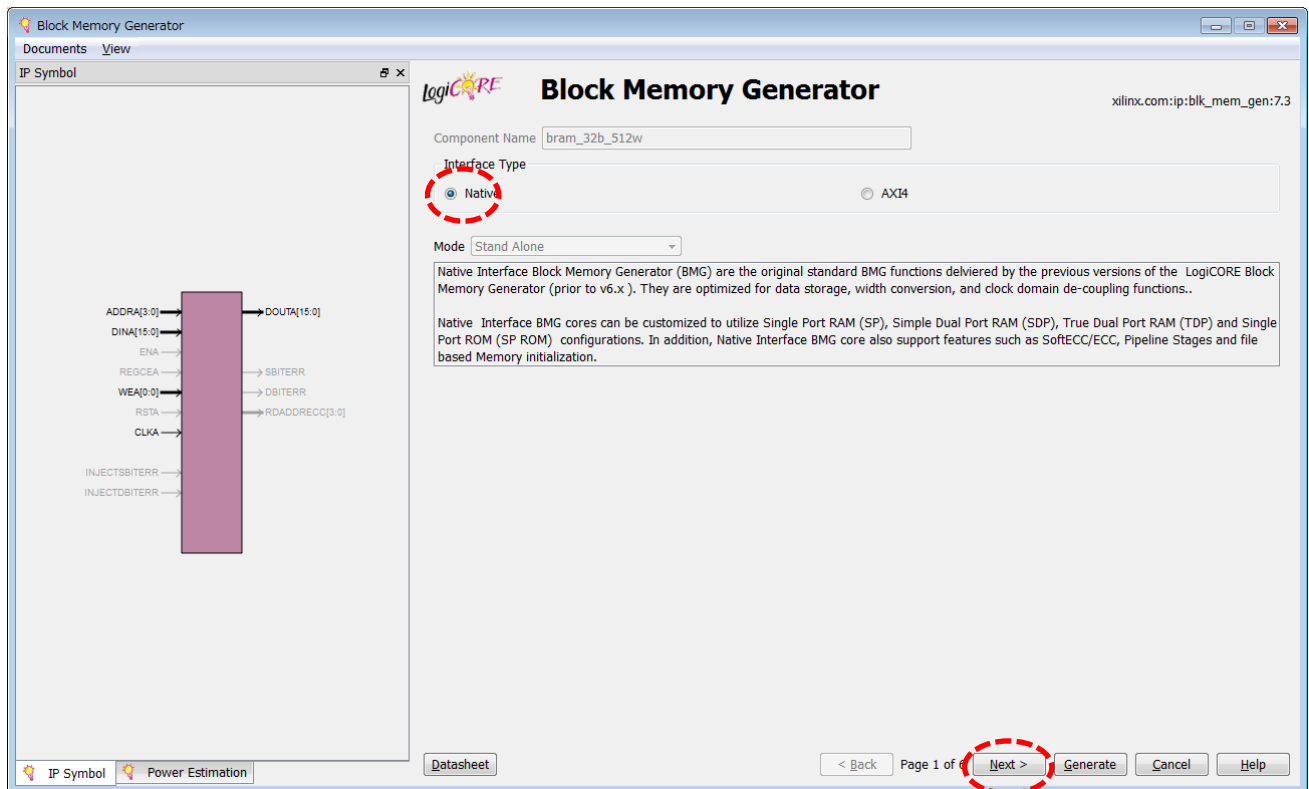


Block Memory Genetator を指定

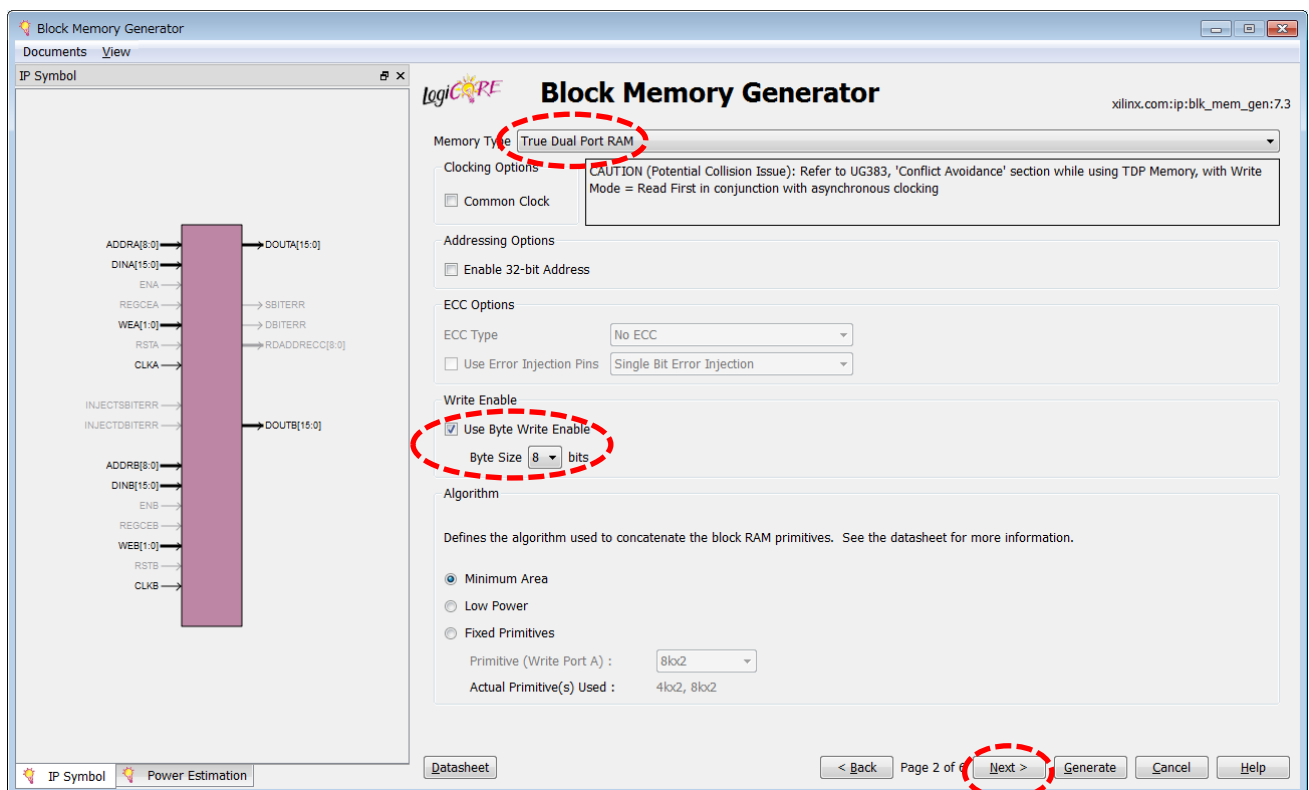


Finish をクリックすると COREgenetator が起動



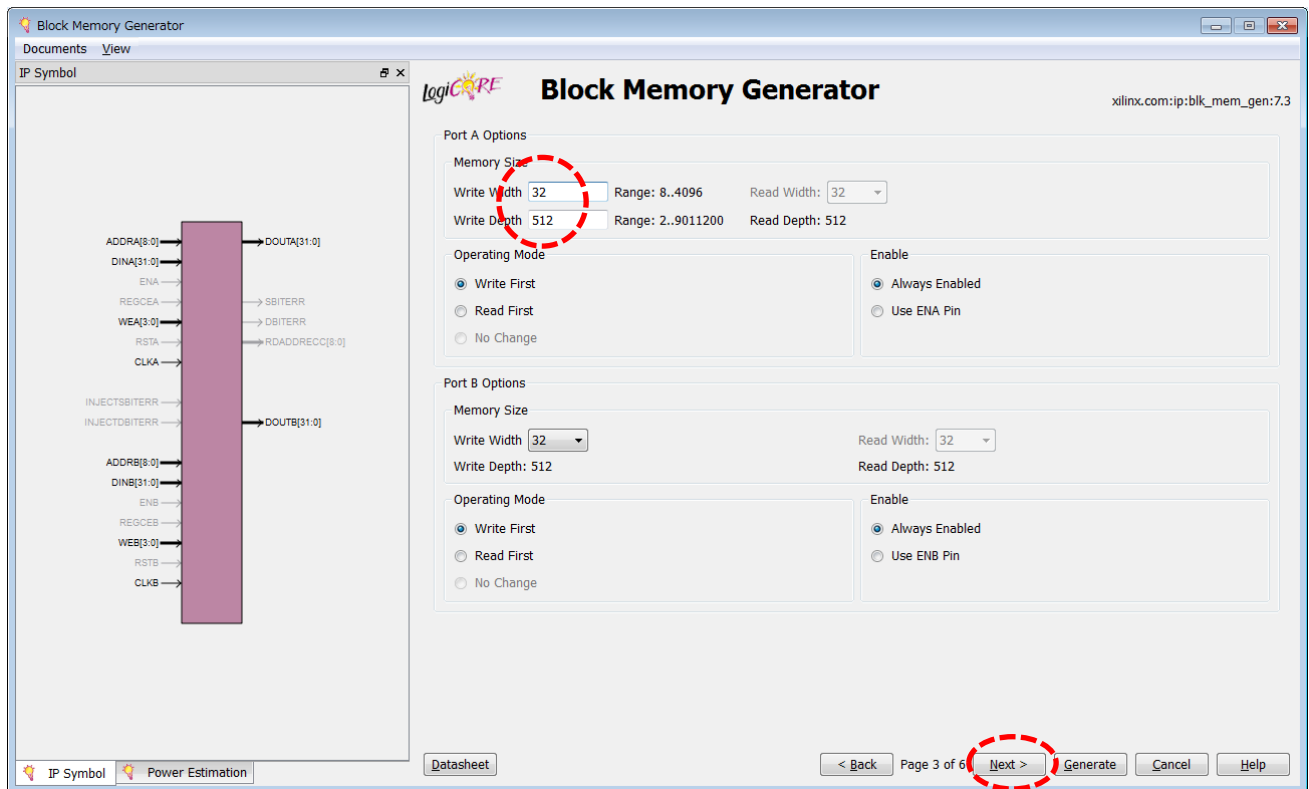


ブロック RAM インターフェースの設定

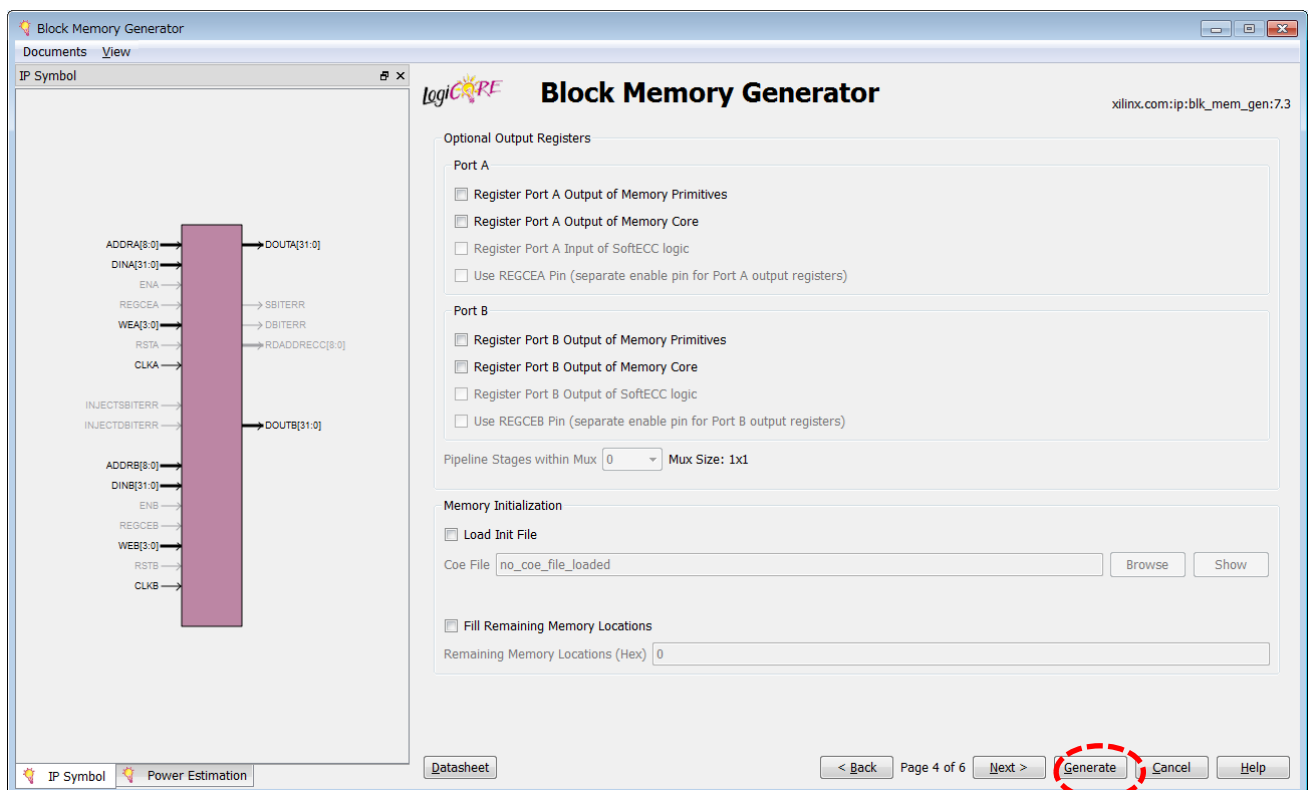


メモリのタイプ指定



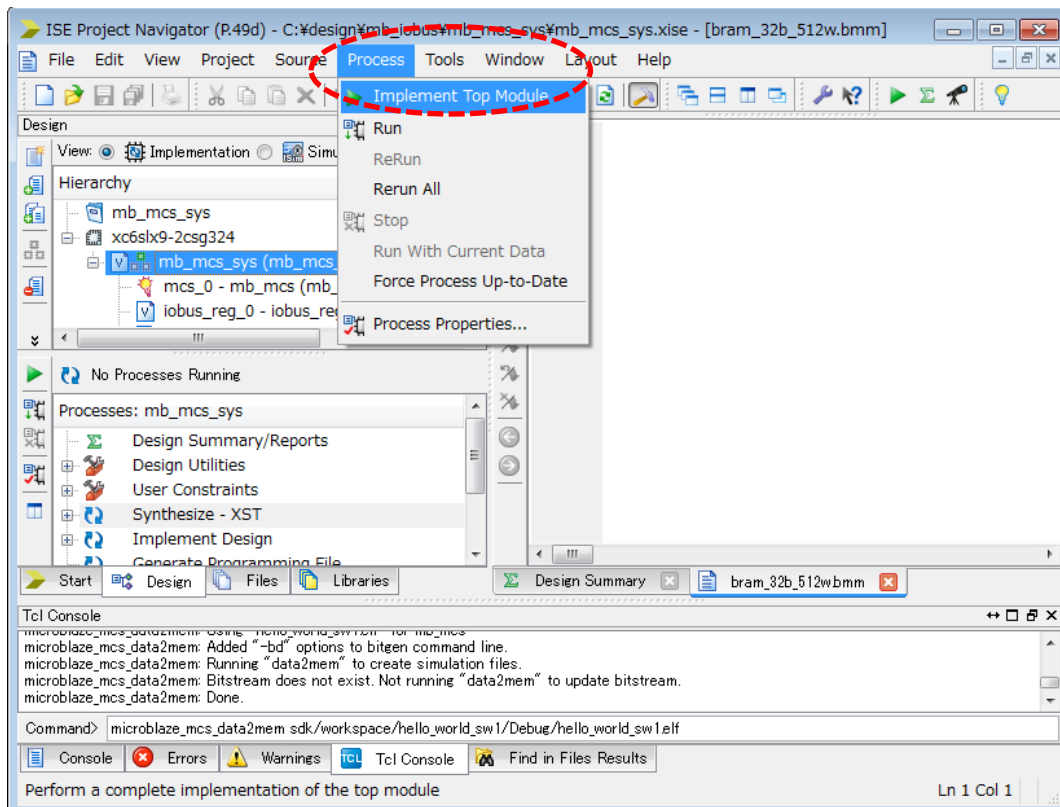


データ幅、データ量の指定



Generate をクリックで作成開始

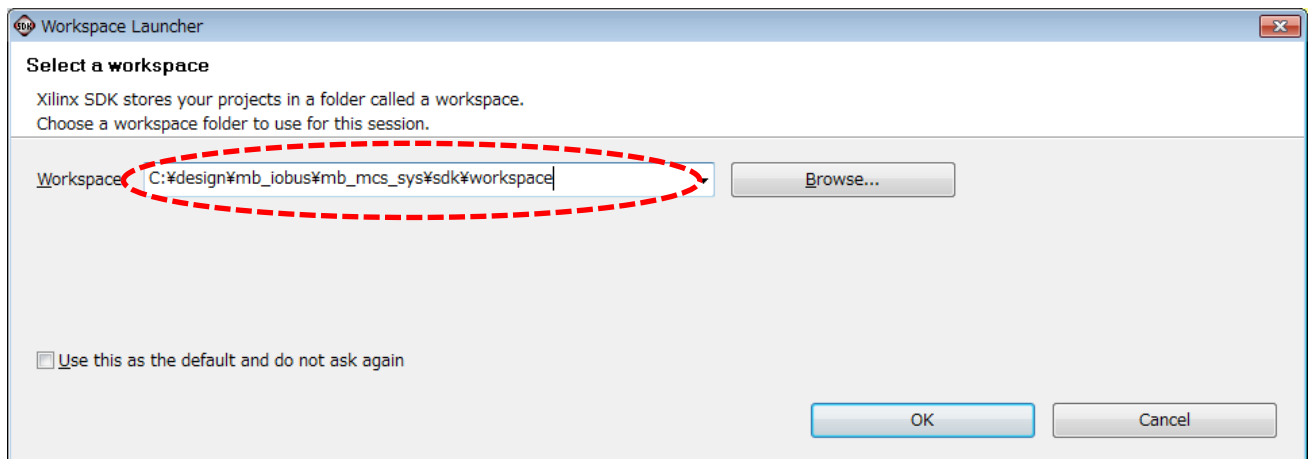




インプリメンテーションの実行

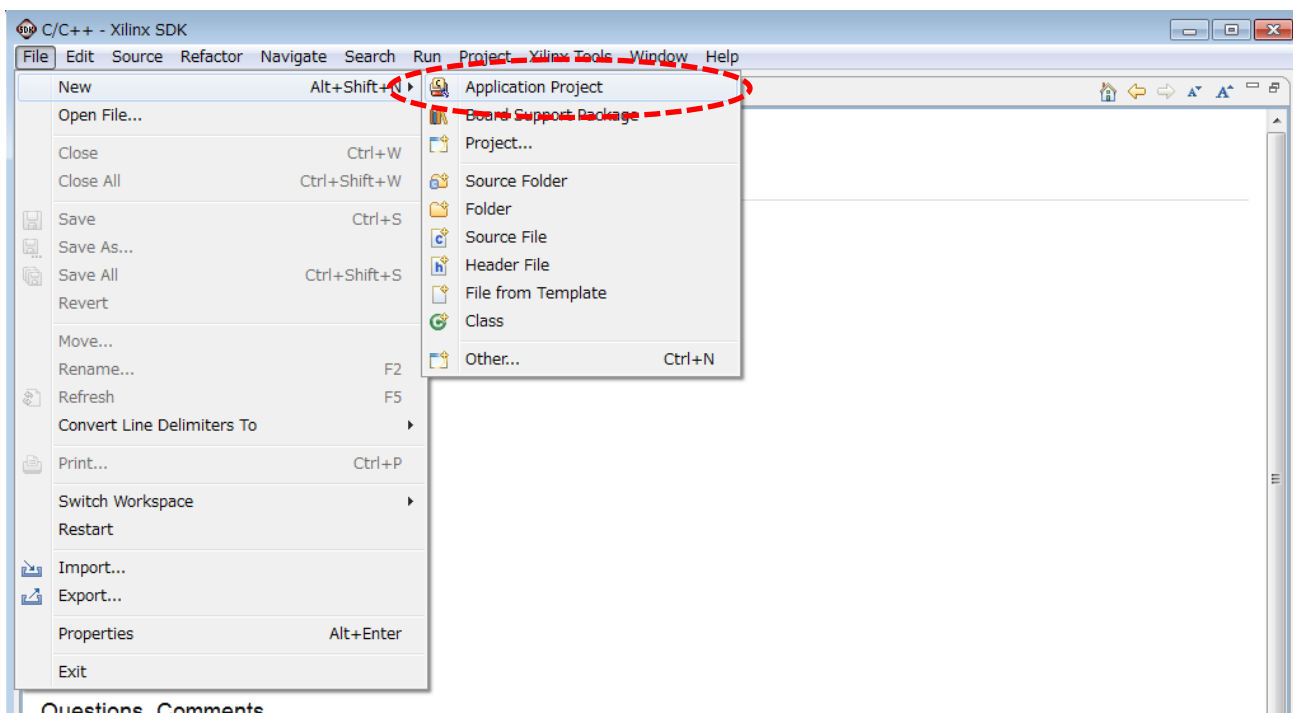


次に Xilinx Software Development Kit (以降 SDK) でソフトウェアを作ります。

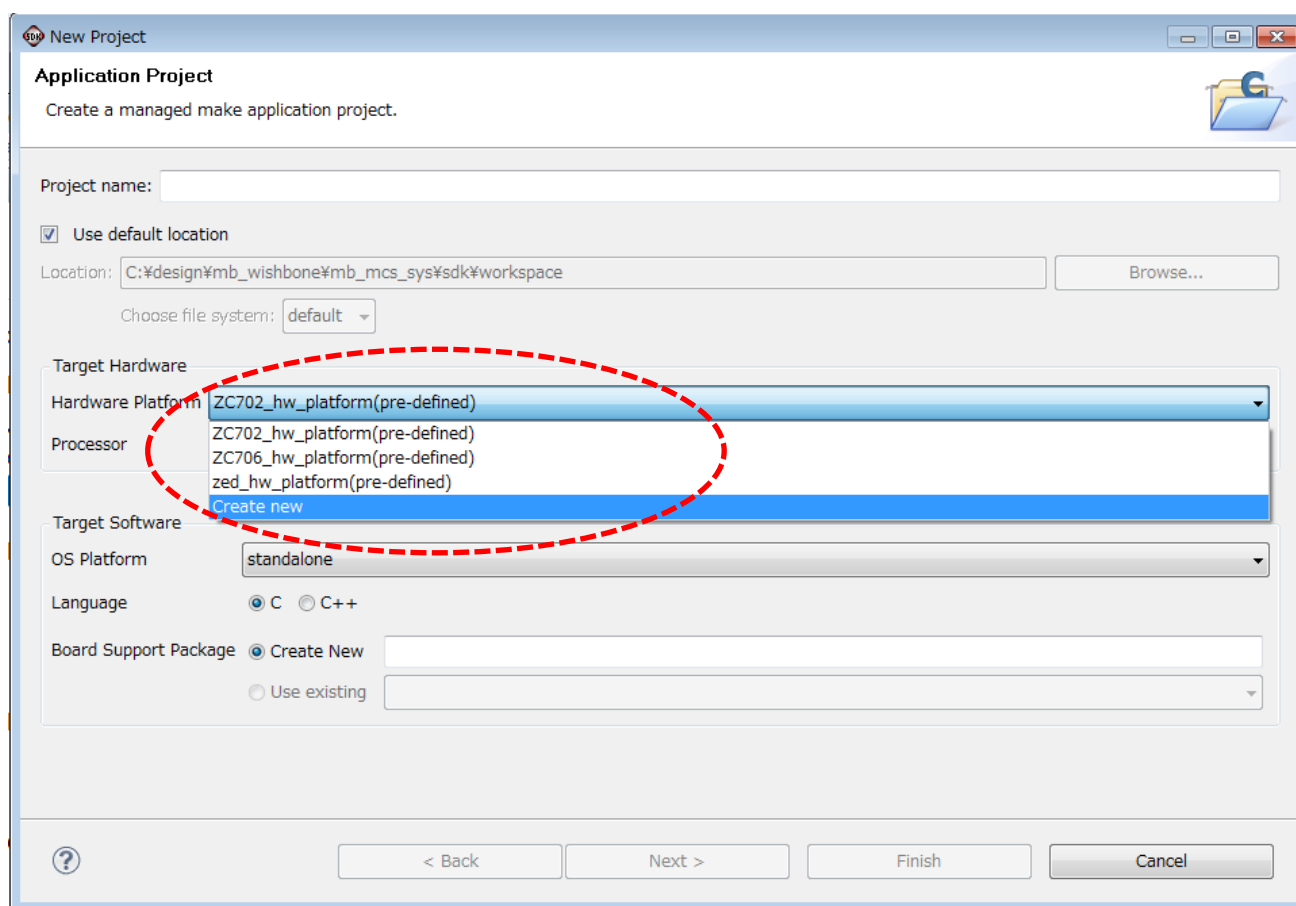


SDK を起動するとワークスペースを指定が要求されます。設計フォルダ/sdk/workspace を設定



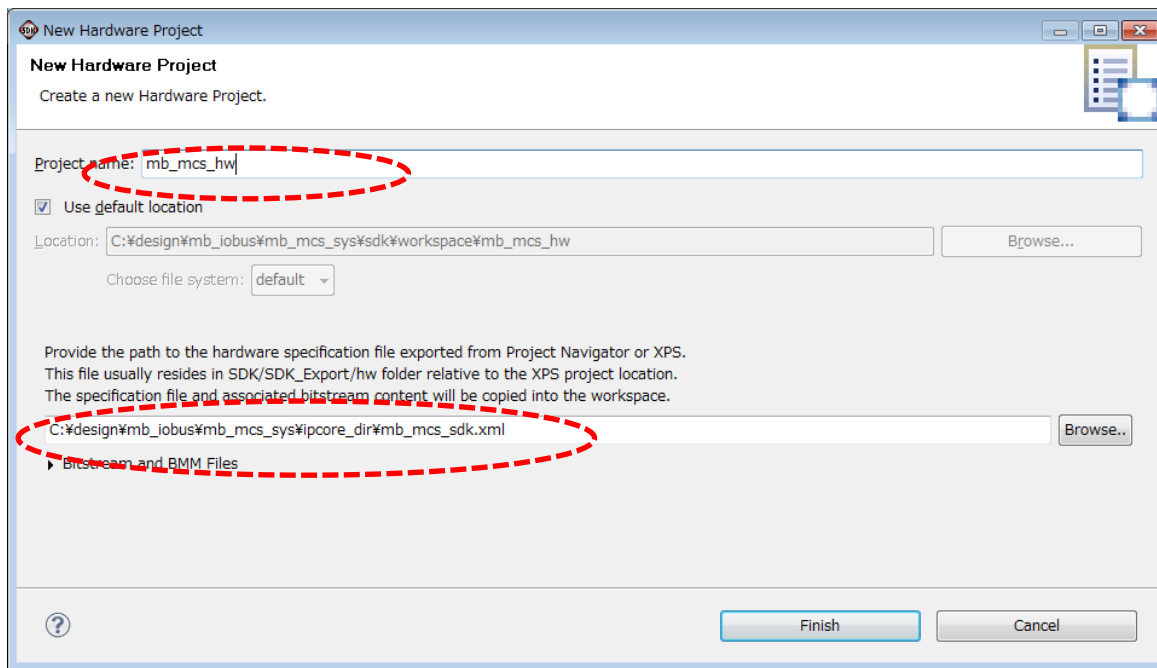


新規のソフトウェアプロジェクト作成



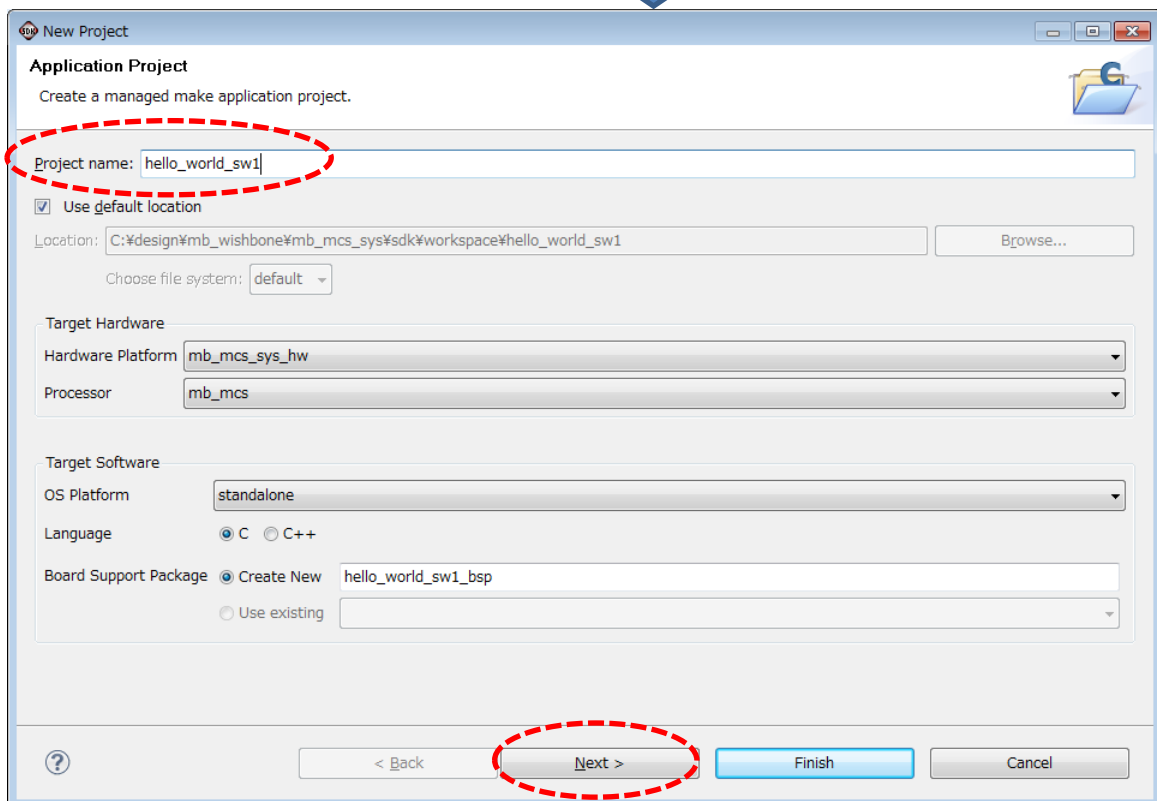
始めに TargetHardware を新規で定義する





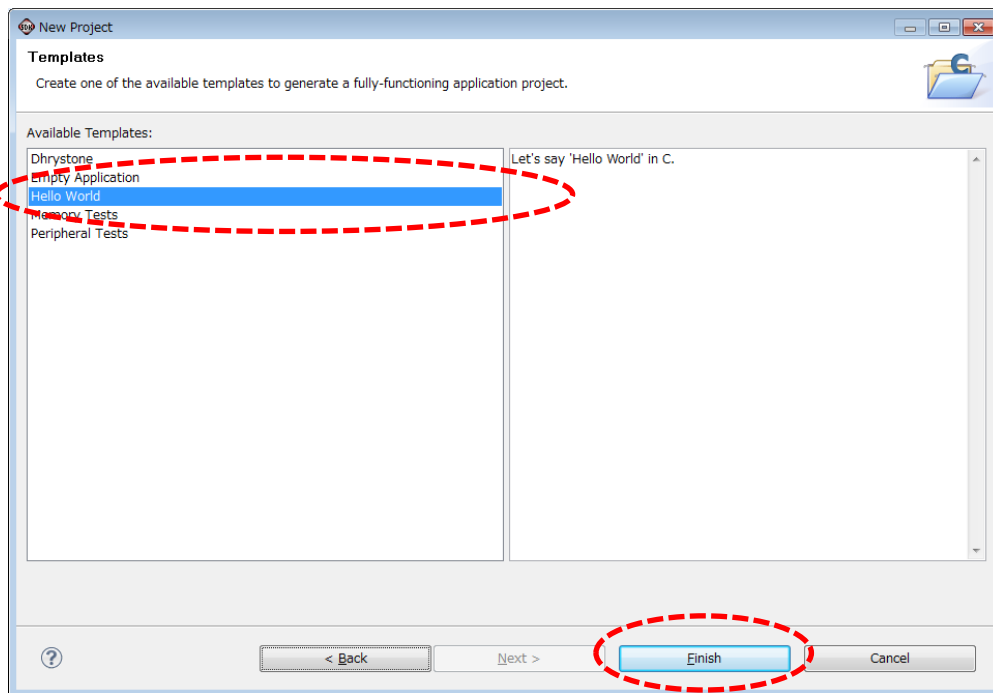
SDK の管理するハードウェアプロジェクト名指定

xlm ファイル(設計ファルダ/ipcore_dir/mb_msc. sdk. xml) 指定

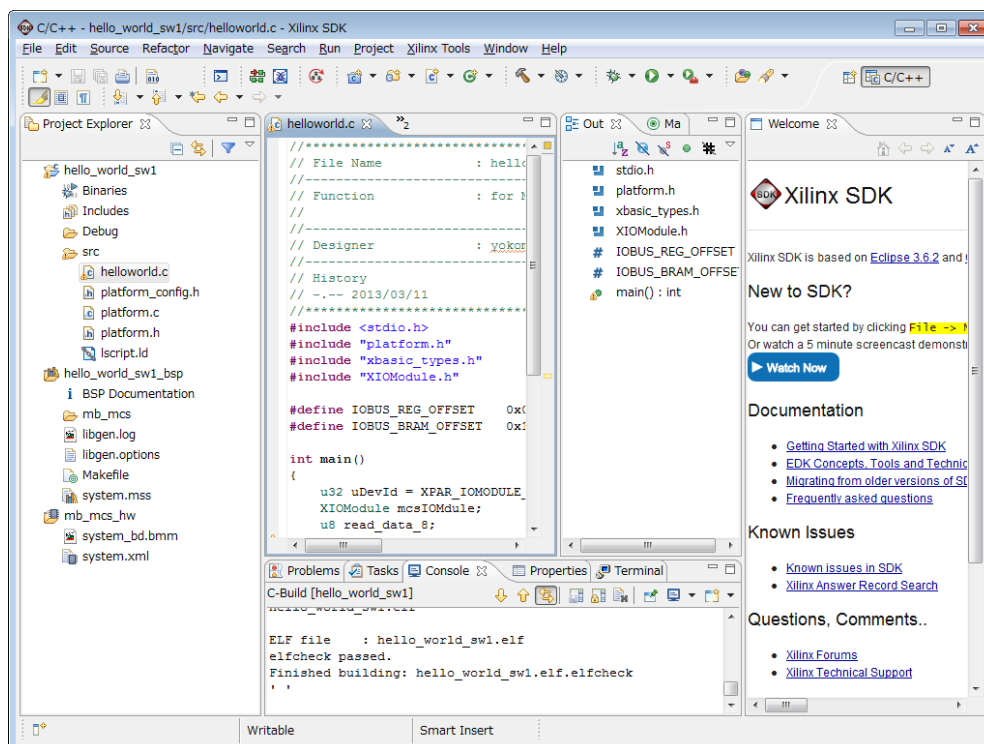


ソフトウェアプロジェクト名指定



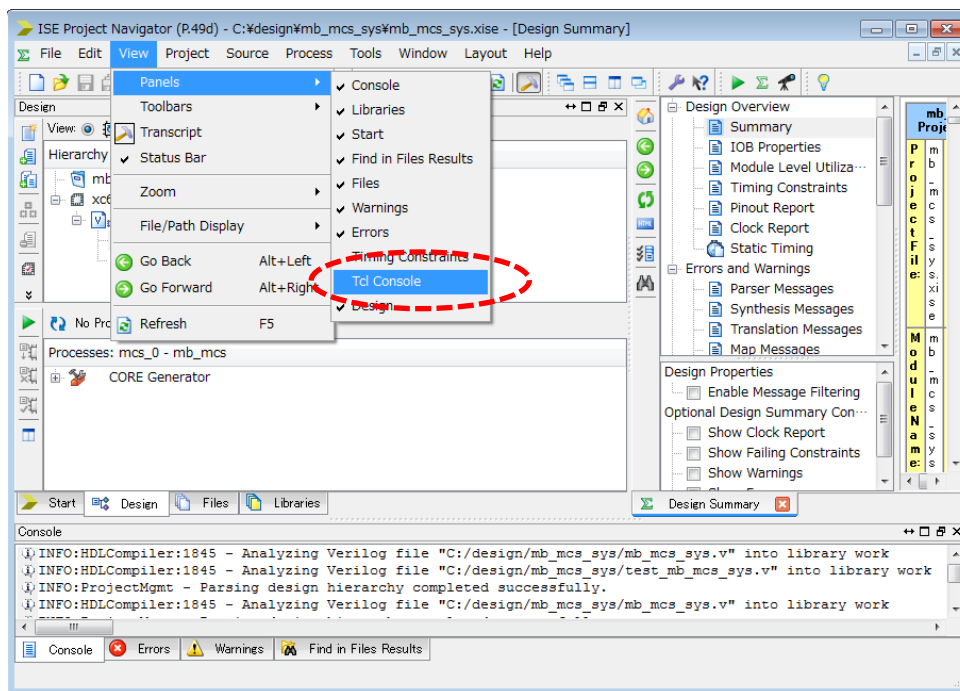


テンプレートに Hello World 選択後、Finish をクリックでソフトウェアプロジェクトが作成される

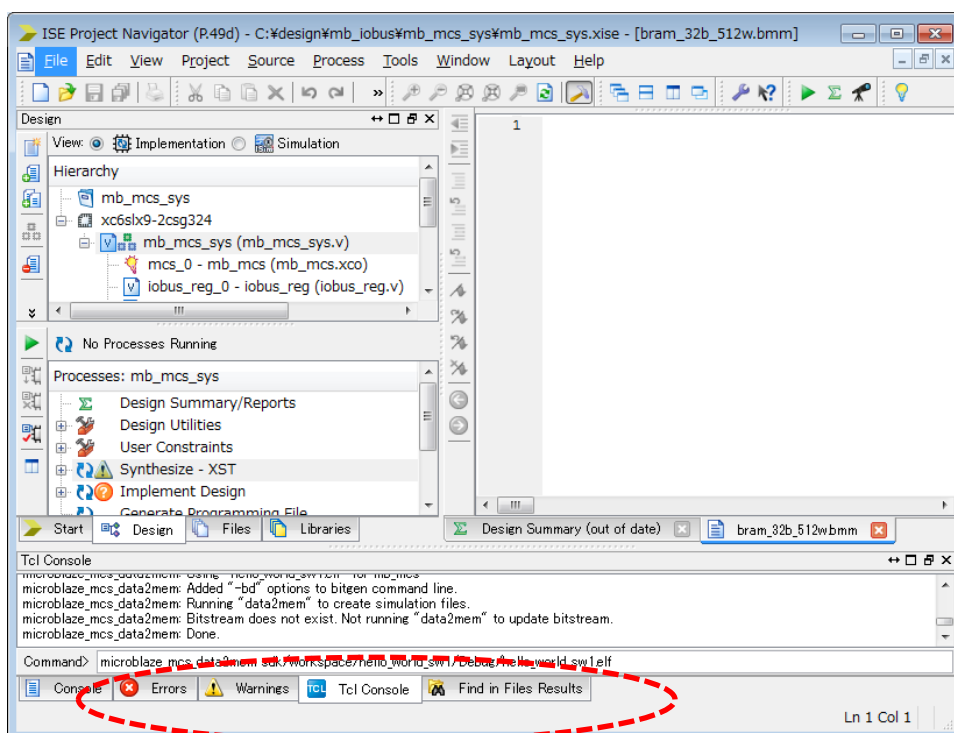


hello_world_sw1 の Helloworld.c にダウンロードデータの mb_iobus/ Helloworld.c の内容を書き写しセーブする。ビルドが実行され、
実行ソフトウェアのファイル (sdk/workspace/hello_world_sw1/Debug/hello_world_sw1.elf) が作られる。





ProjectNavigator でTcl コマンドを使用できるように Tcl Console を表示する



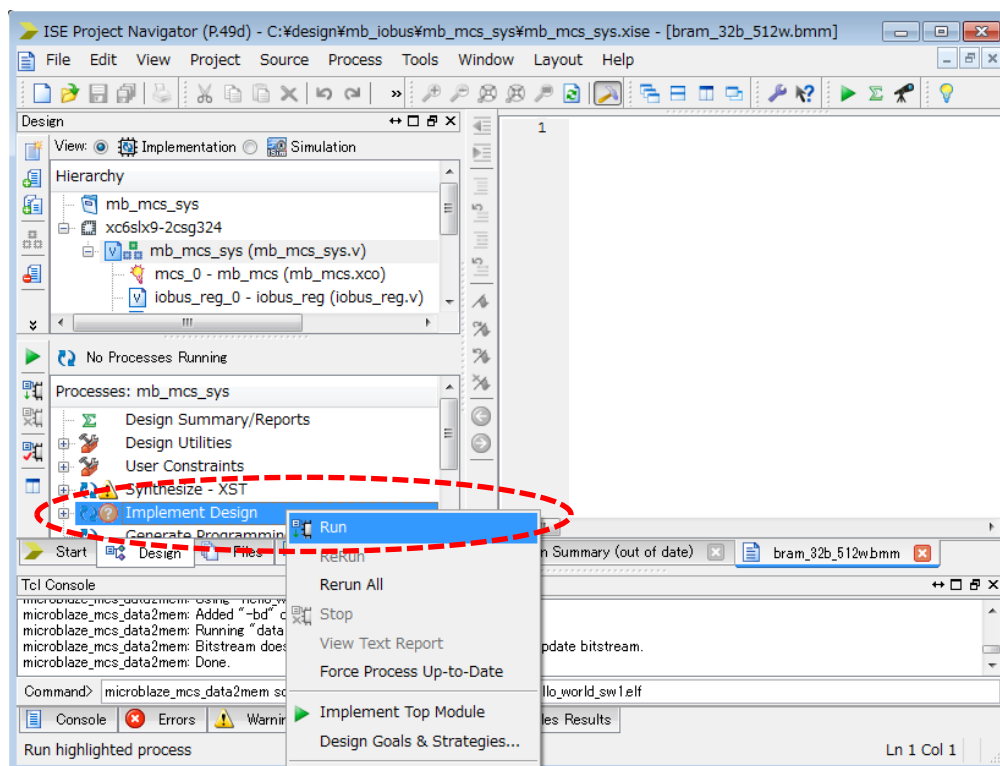
ProjectNavigator でTCL コマンドを使って実行ソフトウェアのファイルを MicroBlazeMCS のメモリの初期値定義ファイルに変換する。

source ipcore_dir/microblaze_mcs_setup.tcl

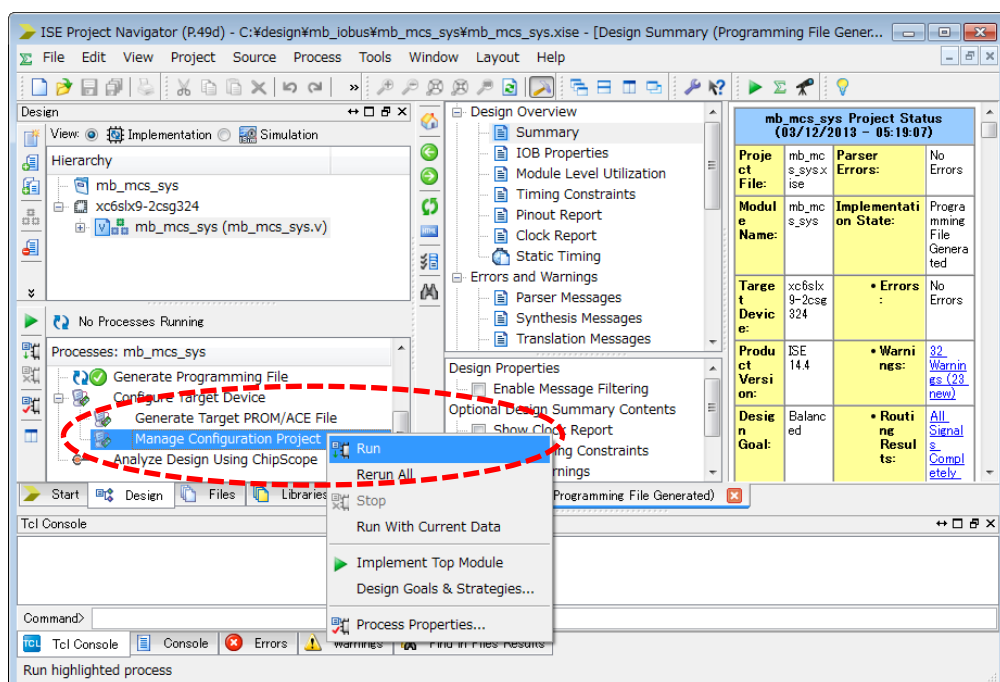
microblaze_mcs_data2mem sdk/workspace/hello_world_sw1/Debug/hello_world_sw1.elf



- ・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 ポートに接続で接続します。

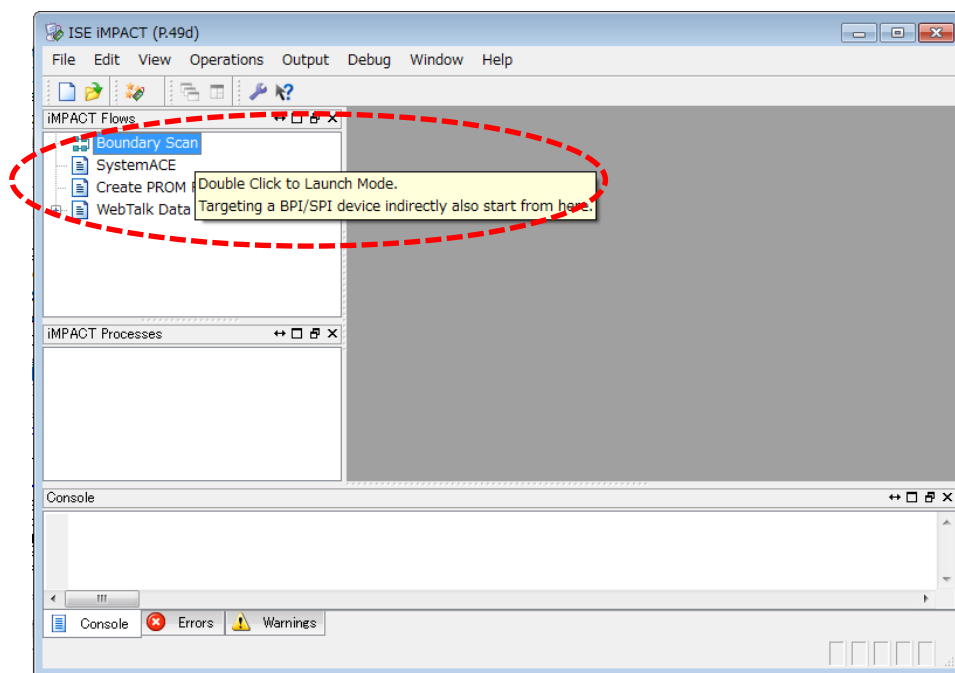


FPGA へ書き込む mb_mcs_sys.bit 作成

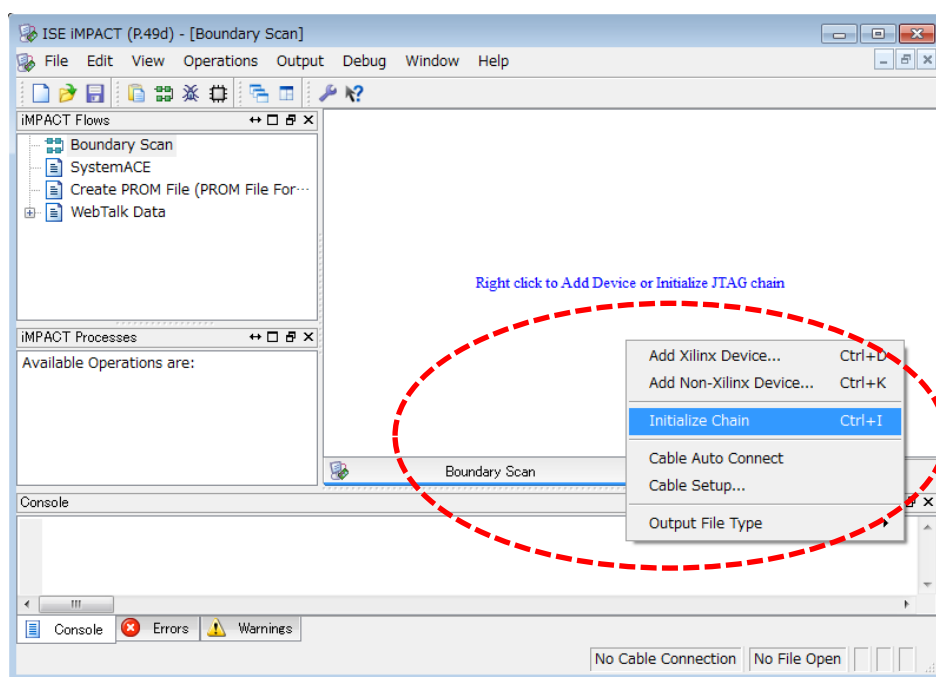


projectNavigator で iMPACT を起動



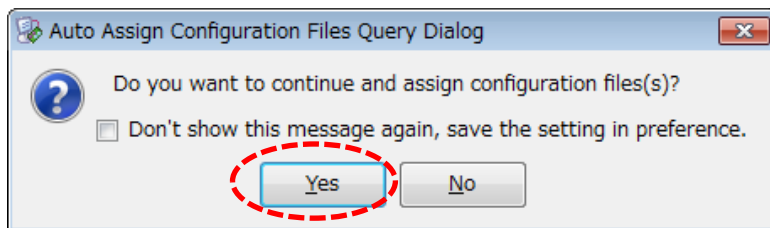


BoundaryScan モードにする

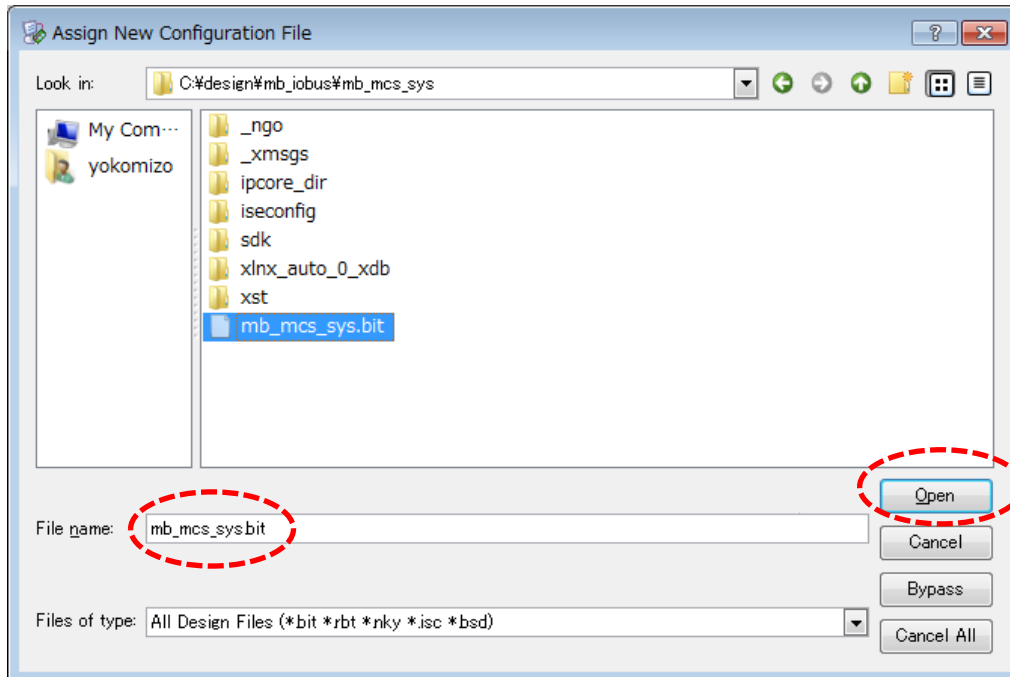


BoundaryScan のウィンドをダブルクリックで FPG を検出する

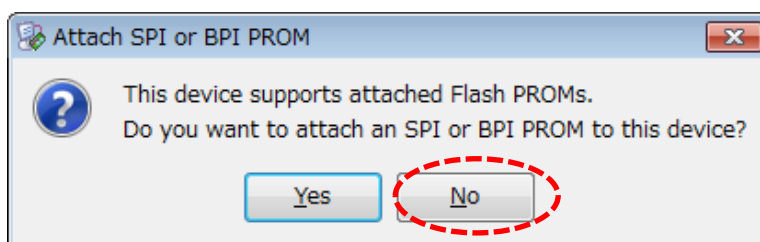




Yes で FPGA に書き込むファイルを指定する

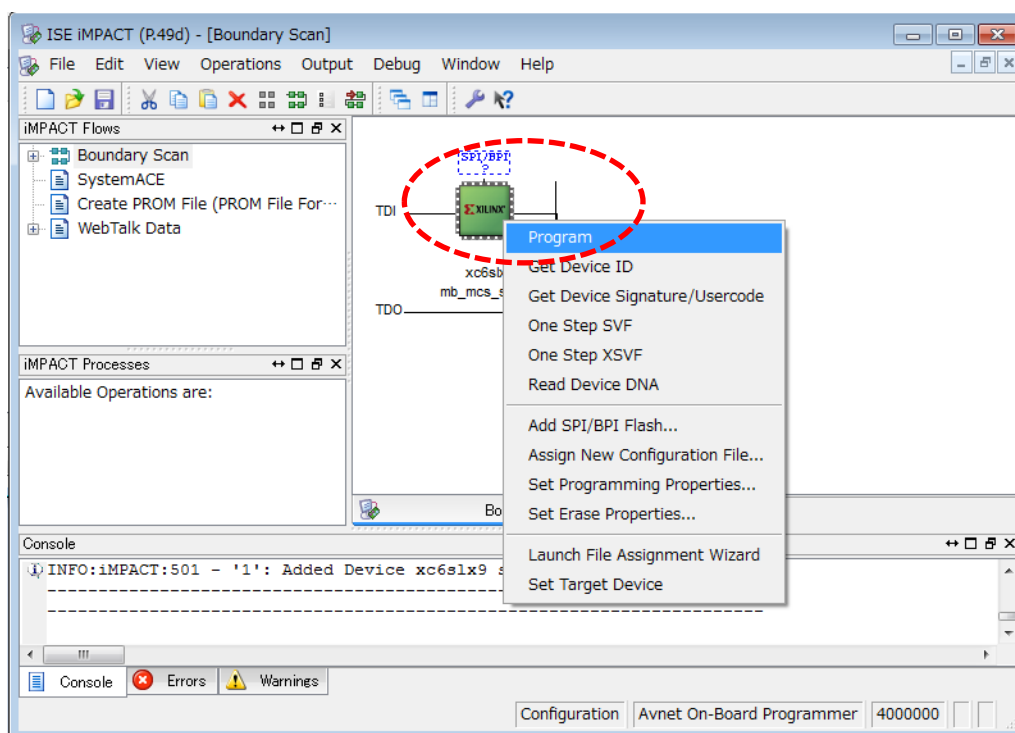
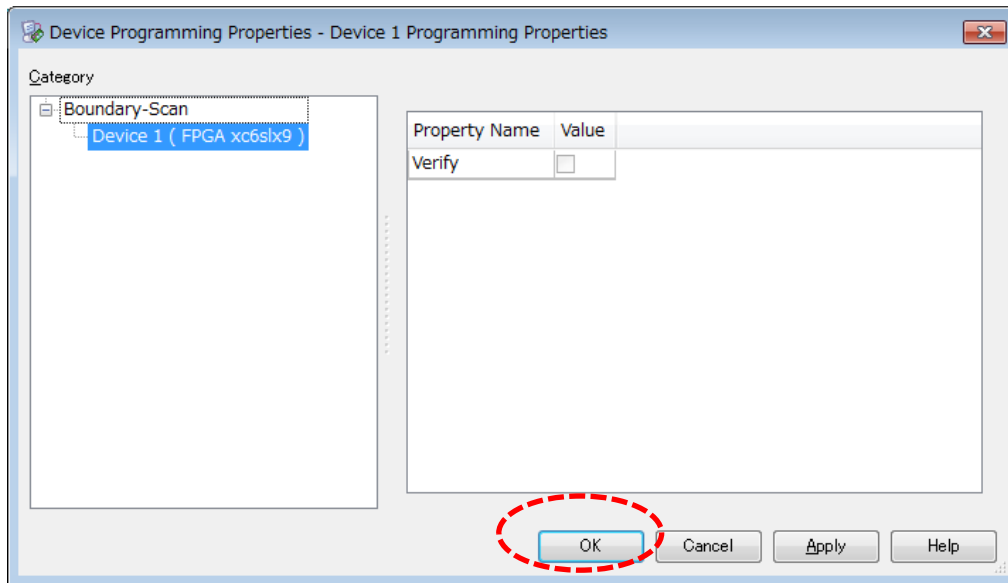


mb_mcs_sys.bit を指定



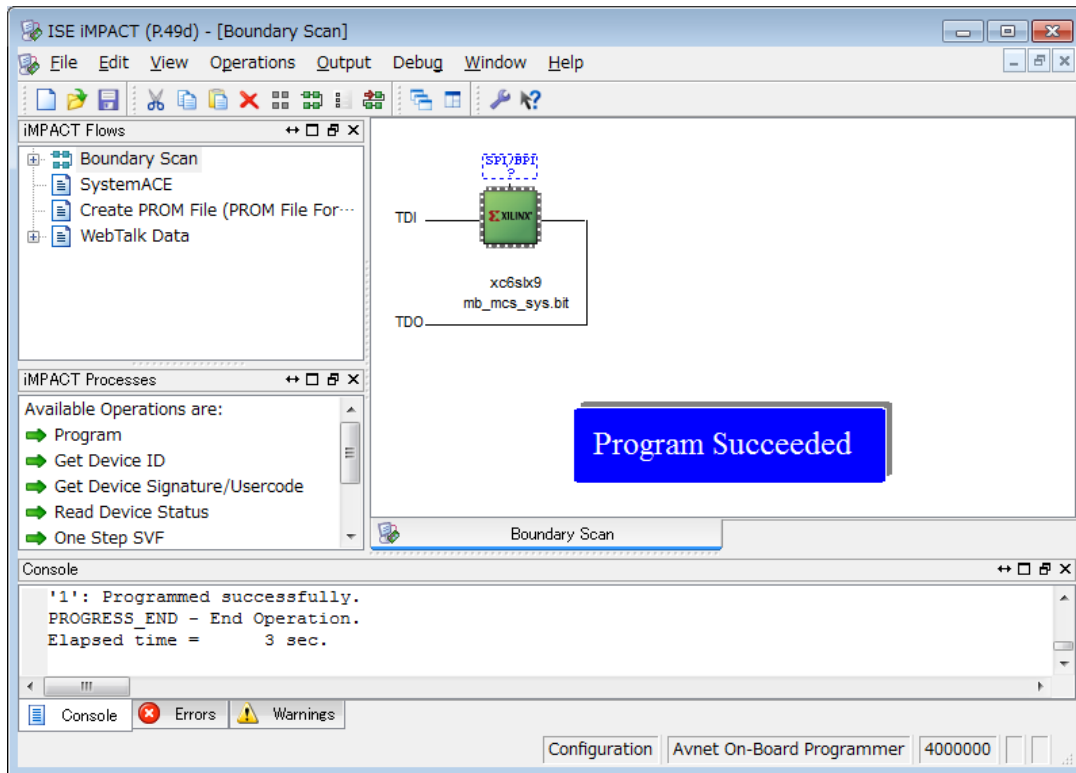
PROM データは使わないので No を選択





FPGA へのプログラミング実行

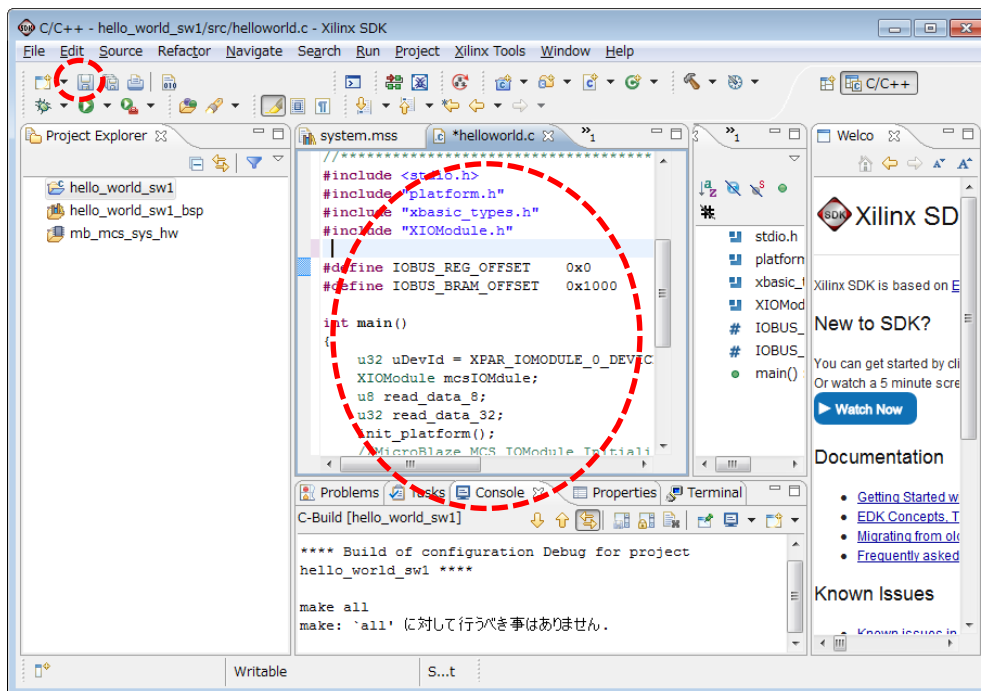




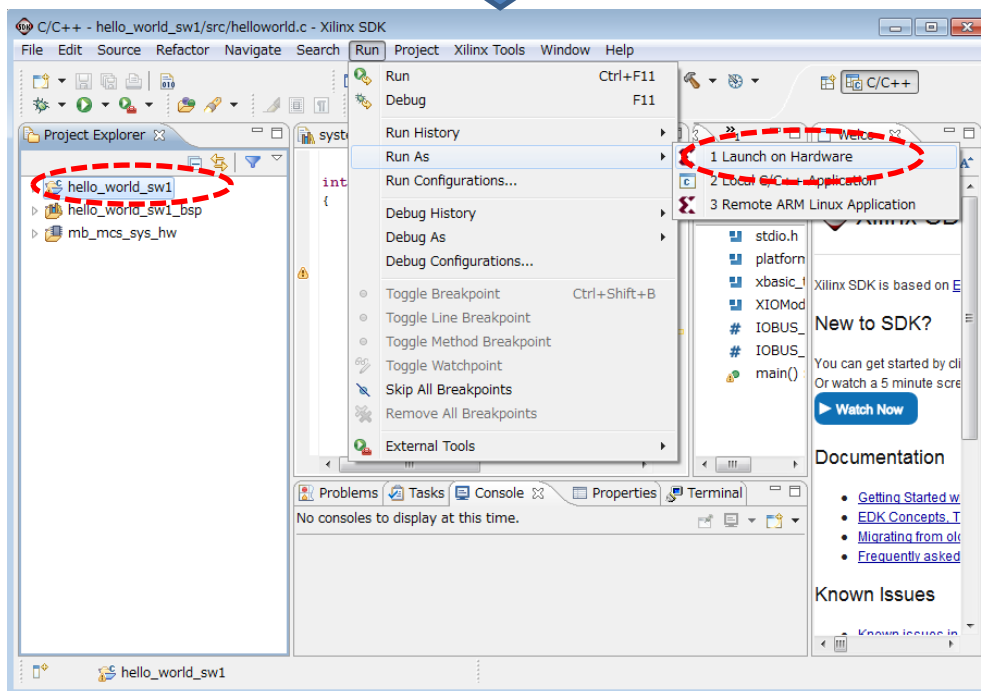
ProgramSucceeded と表示されればプログラミング完了、LX9 マイクロボードの動作を確認する。



- ・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。



hello_world_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。