

OpenCores SD CARD の LX9 MicroBoard 用設計データの使い方 (2014/01/23)

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
- ・ sd_card_sw1.c 制御用 C ソースコード

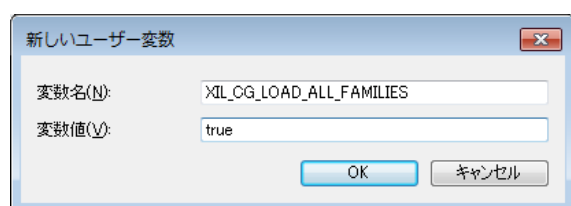
免責事項

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 sd_card/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、sd_card_microborad_data の下にある Verilog-HDL ソースと UCF ファイルを sd_card/mb_mcs_sys へコピーする。
- ・ SD_CARD のデータを OpenCores のサイト (<http://opencores.org/project,spimaster>) からダウンロード
- ・ ダウンロードした spimaster_latest.tar.gz を解凍する
- ・ 解凍データの spimaster/trunk/rtl/の下にある Verilog-HDL ソースを sd_card/mb_mcs_sys へコピー
- ・ PWM のデータを OpenCores 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 sd_card/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



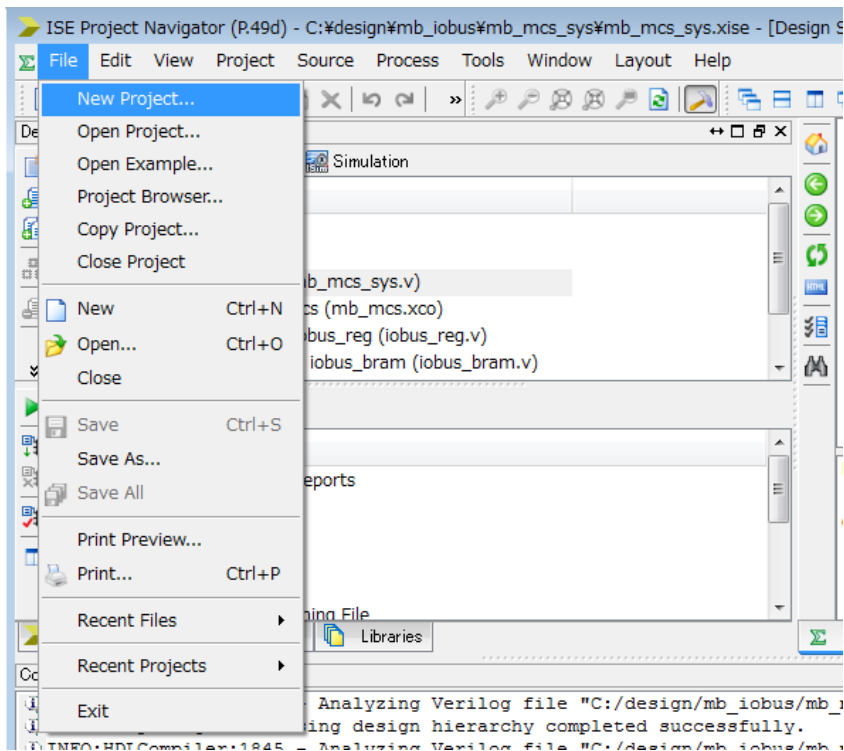
変数名：XIL_CG_LOAD_ALL_FAMILIES、値：true、と入力して OK

- ・ ProjectNavigator を起動します。

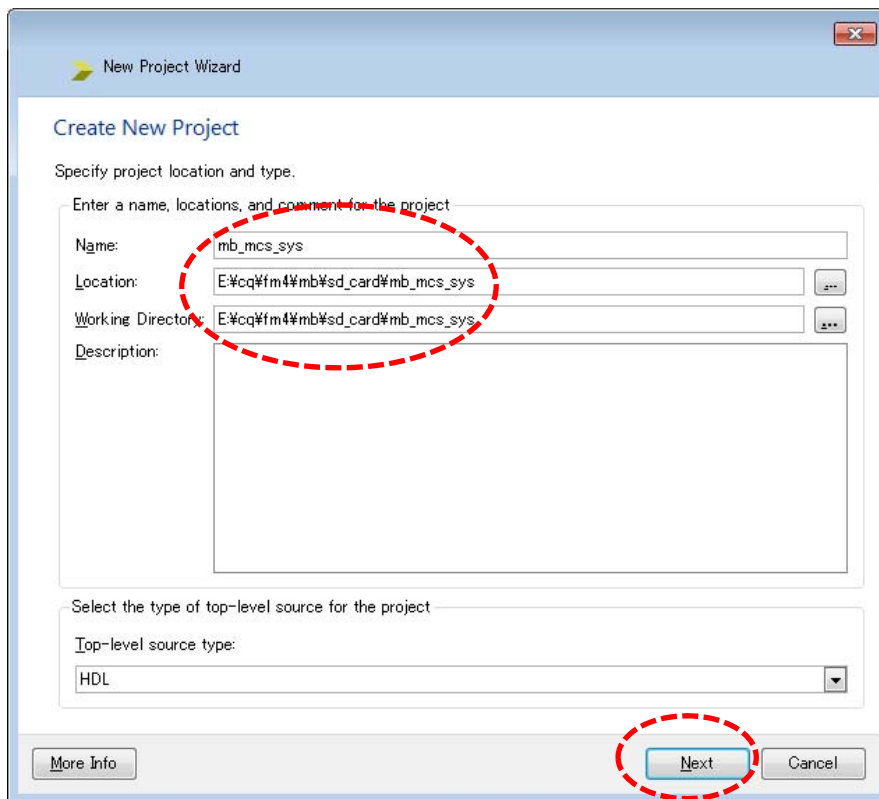
スタートメニューから「Xilinx Design Tools」→「ISE Design Suite 14.4」→「ISE Design Tools」→Project Navigator」を起動してください。



・新規設計プロジェクト作成

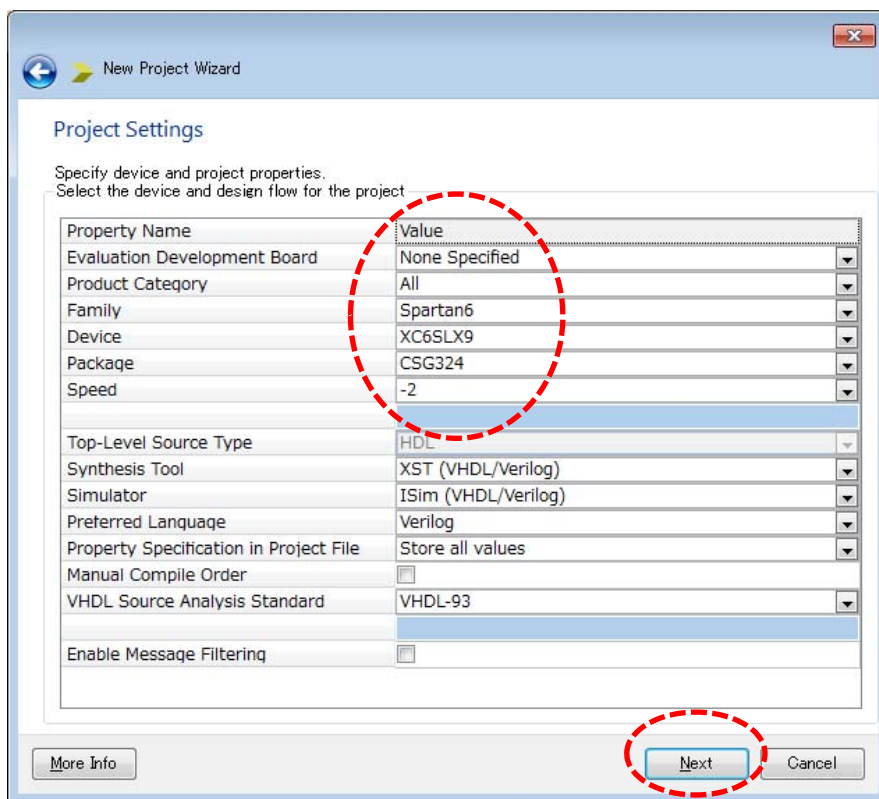


File→New Project 選択



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



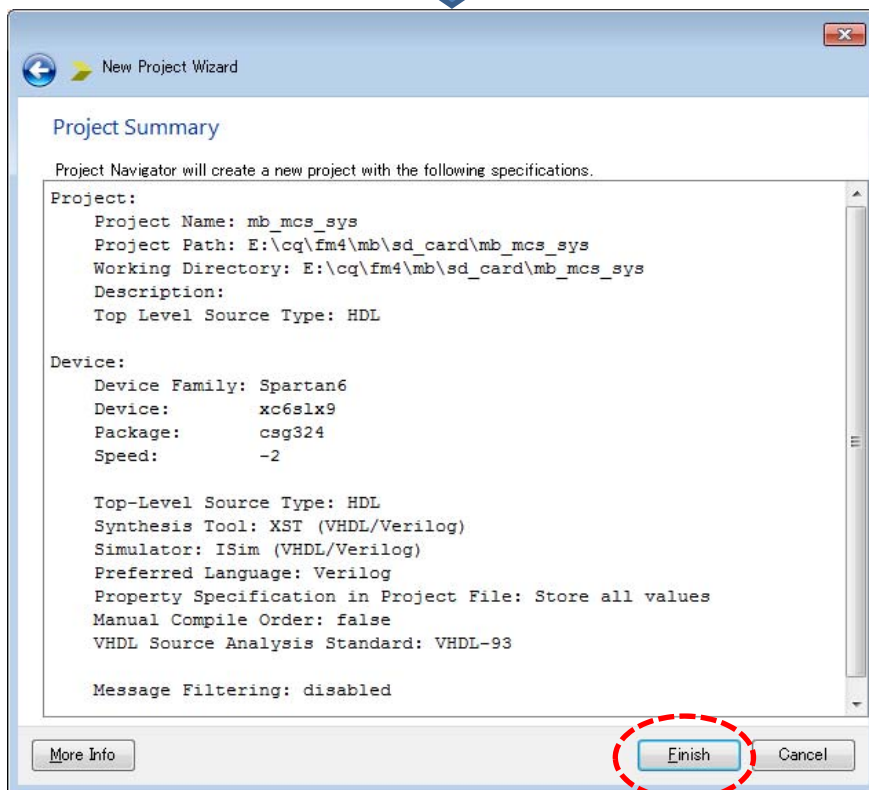


The 'New Project Wizard' dialog box is shown at the 'Project Settings' step. It contains a table of properties and their values. A red dashed circle highlights the 'Device' section, which includes 'Family' (Spartan6), 'Device' (XC6SLX9), and 'Package' (CSG324). Another red dashed circle highlights the '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: E:\cq\fm4\mb\sd_card\mb_mcs_sys
- Working Directory: E:\cq\fm4\mb\sd_card\mb_mcs_sys
- Description:
- Top Level Source Type: HDL

Device:

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

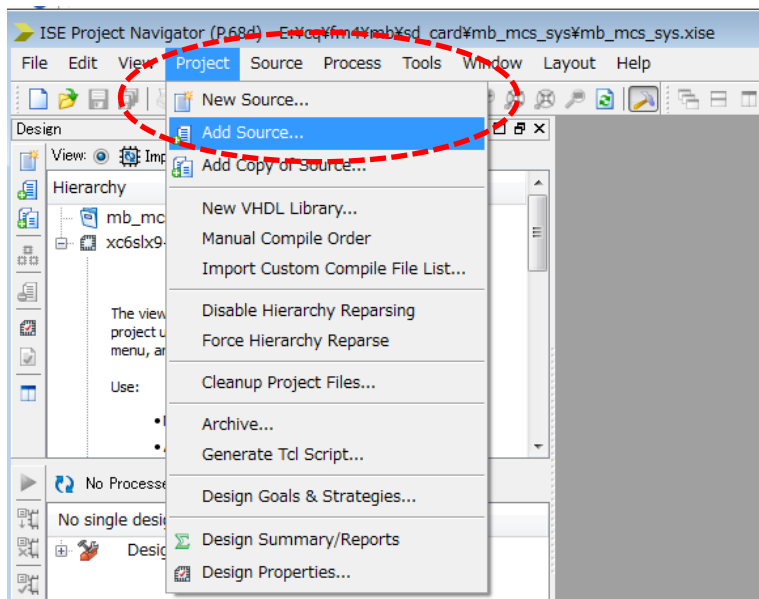
Top-Level Source Type: HDL
Synthesis Tool: XST (VHDL/Verilog)
Simulator: ISim (VHDL/Verilog)
Preferred Language: Verilog
Property Specification in Project File: Store all values
Manual Compile Order: false
VHDL Source Analysis Standard: VHDL-93

Message Filtering: disabled

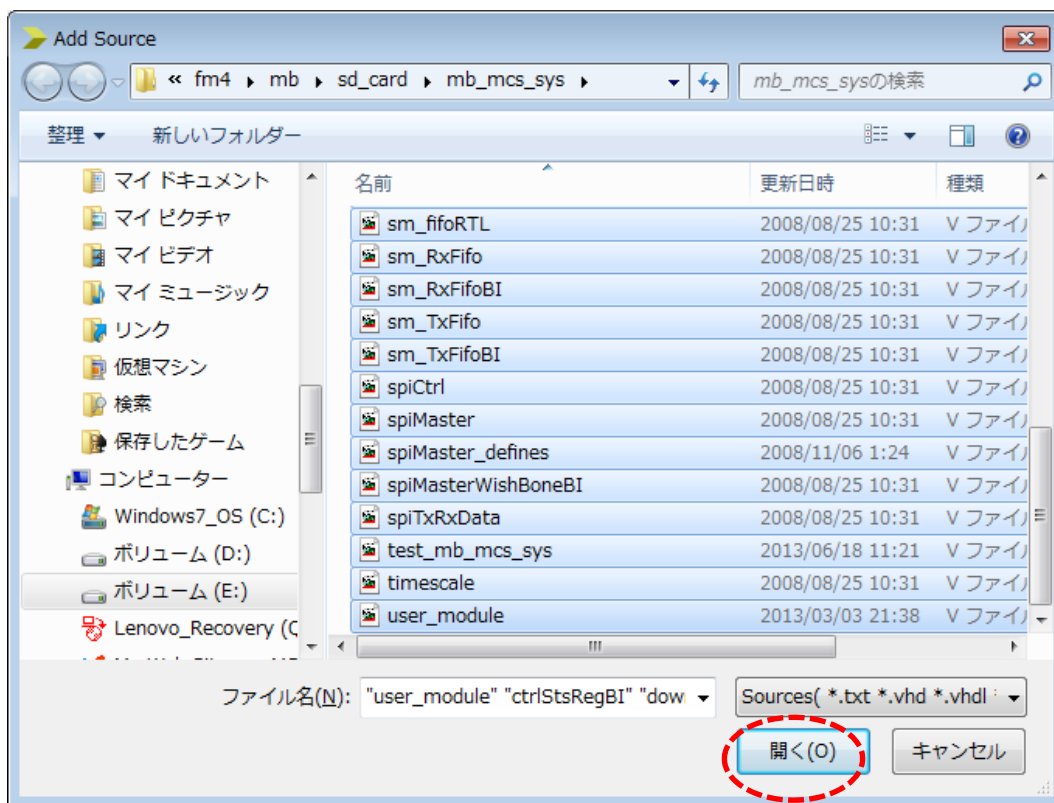
Buttons: More Info, Finish, Cancel

Finish をクリック



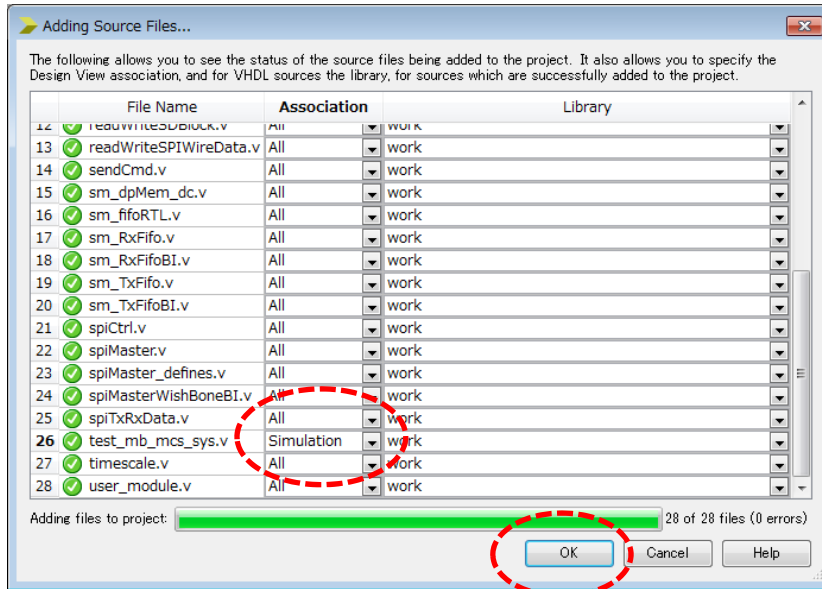


設計データを追加します。Project→Add Source を選択

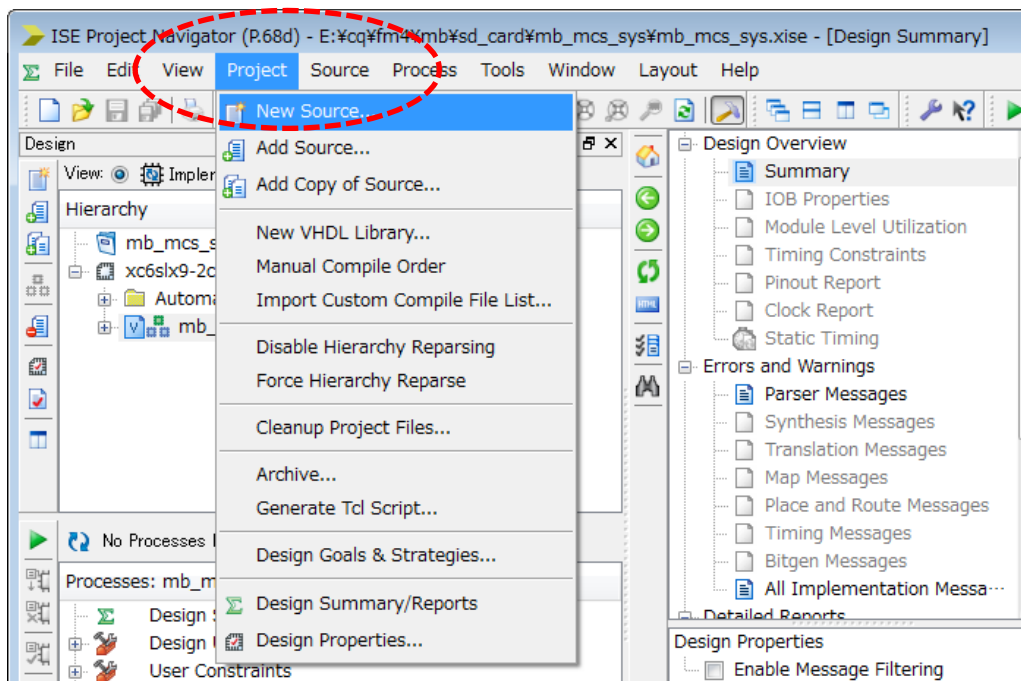


ファイルの指定、mb_mcs_sys の下にある Verilog-HDL ファイルと UCF ファイルを指定



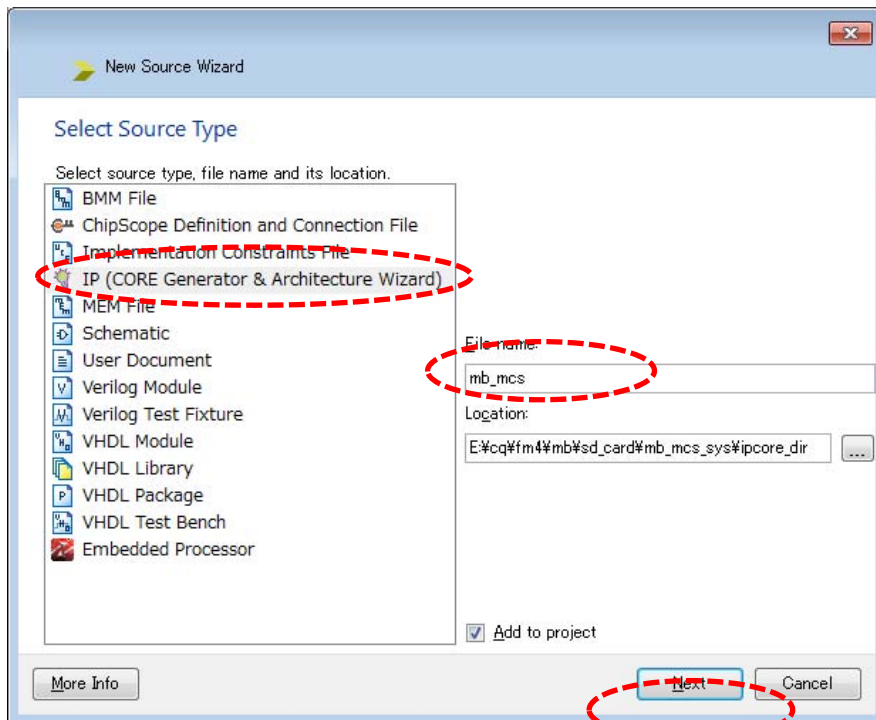


mb_mcs_sys.ucf はインプリメンテーションで使用するので Implementation を指定
 test_mb_mcs_sys.v はテストベンチなので Simulation を指定

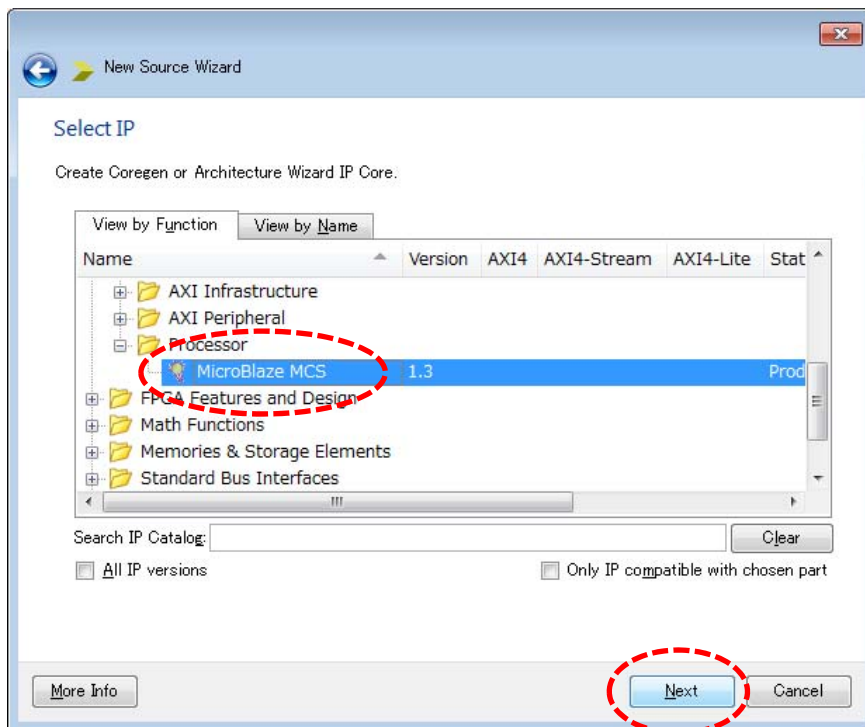


ProjectNavigator で MicroBlaze MCS を追加する。Project→New Source を選択



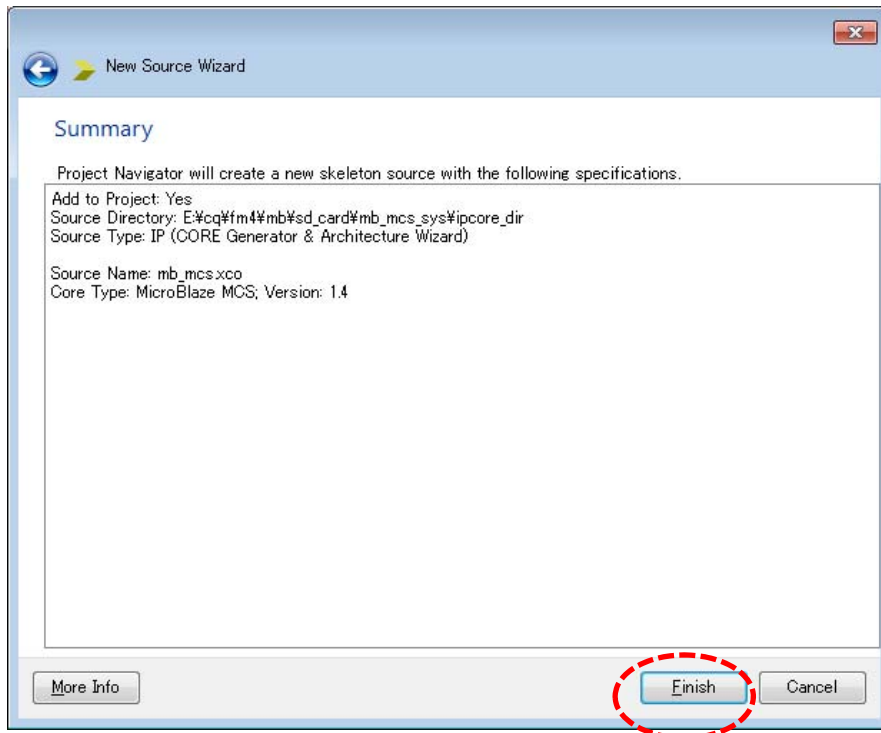


IP (CORE Gener…をクリックして選択、ファイル名に mb_mcs を指定、Next をクリック

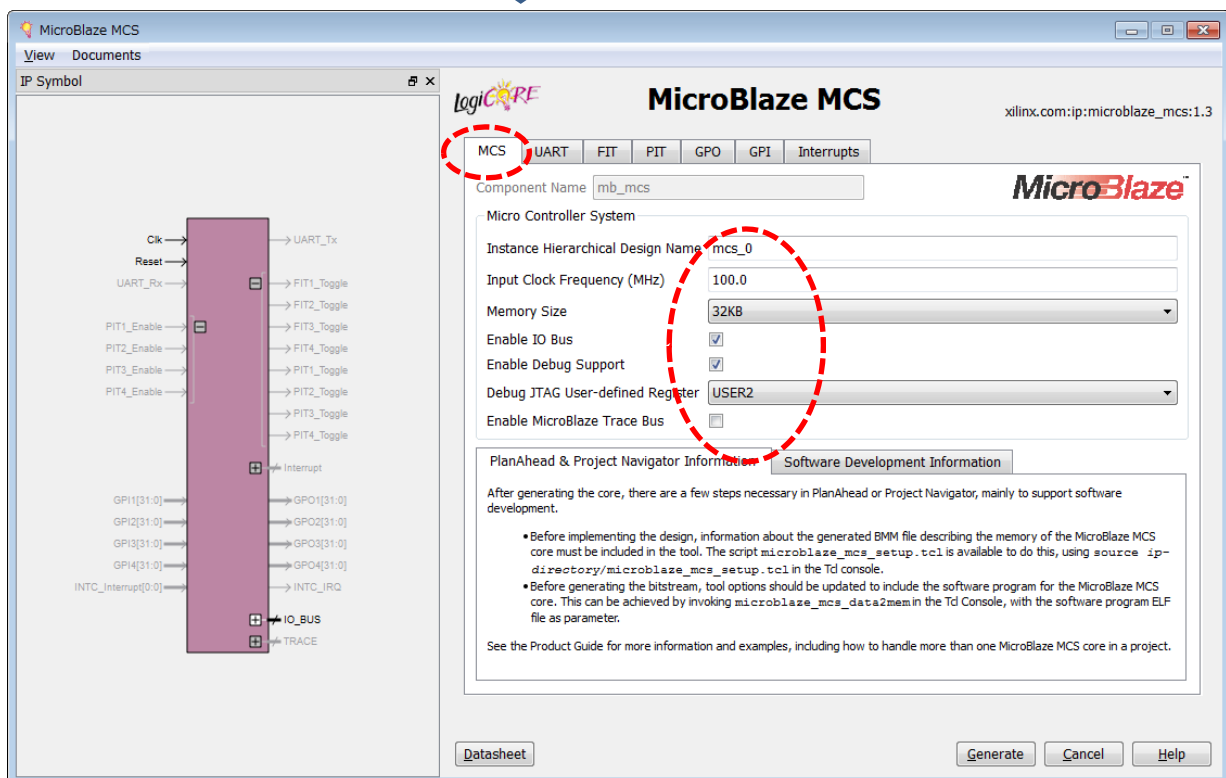


MicroBlaze MCS 選択して Next をクリック



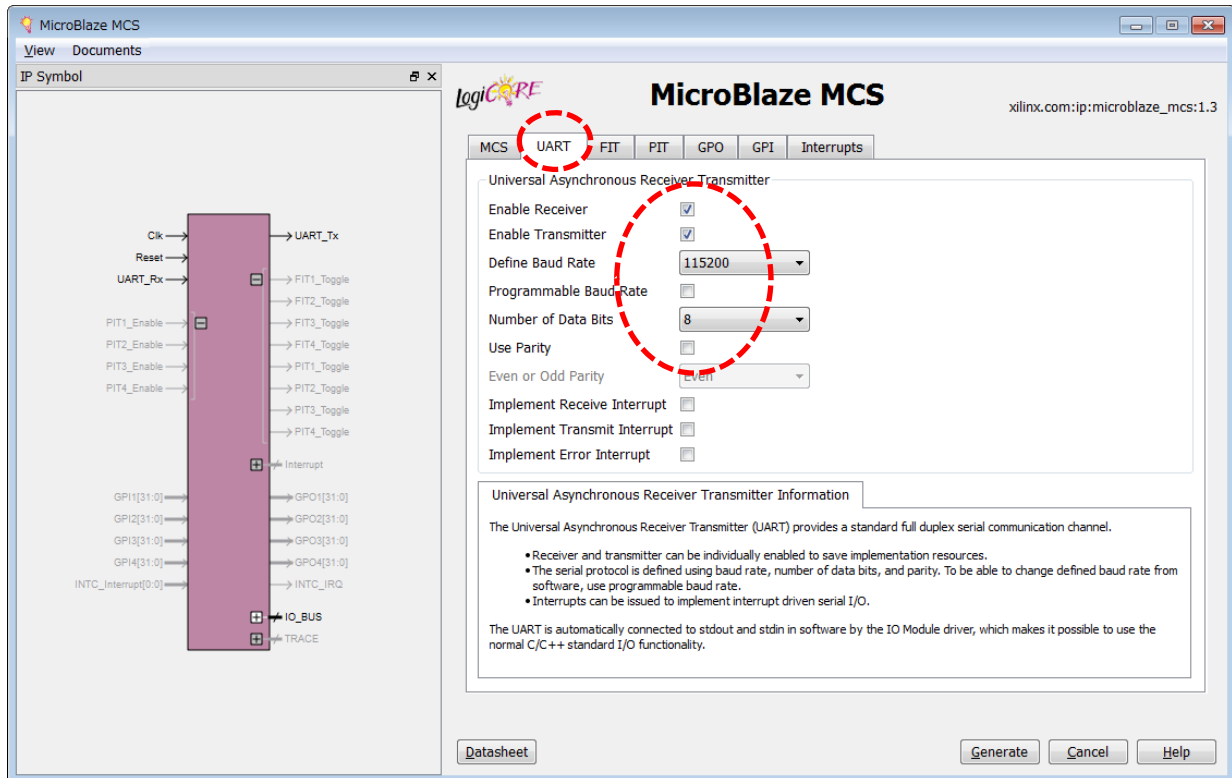


Finish で CORE generator 起動

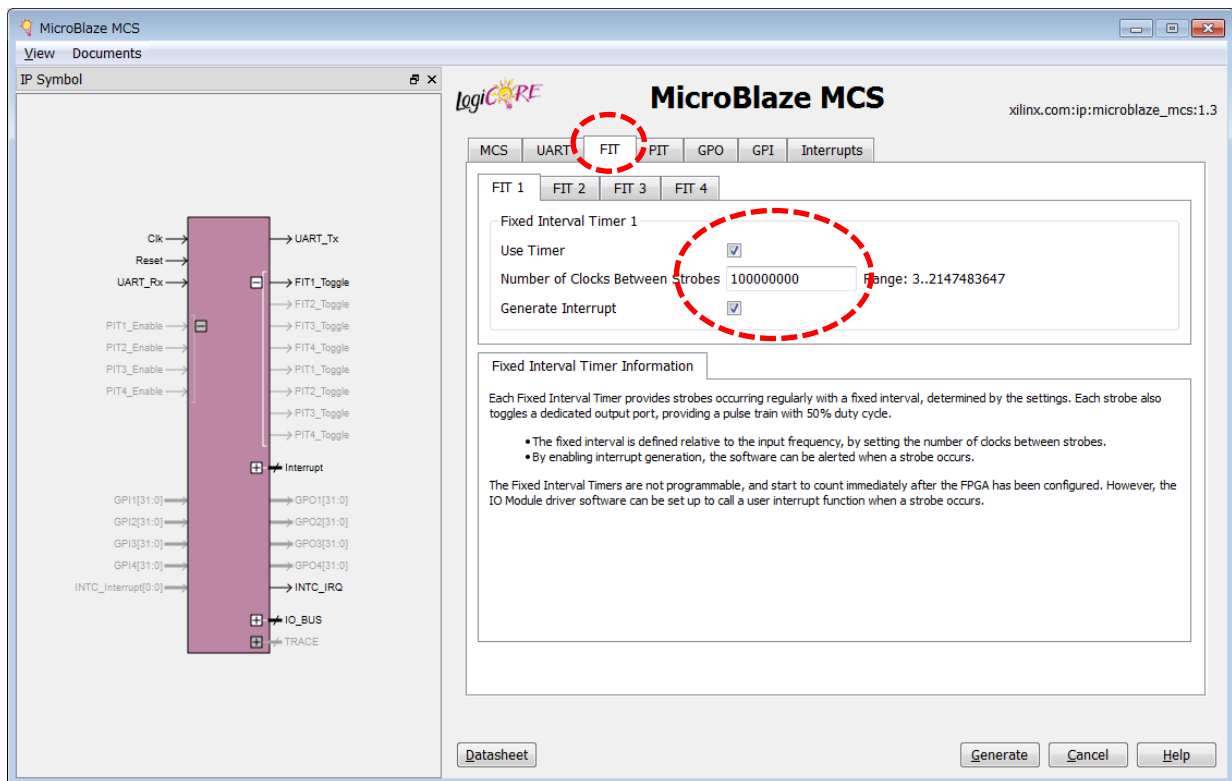


MicroBlaze MCS の基本設定



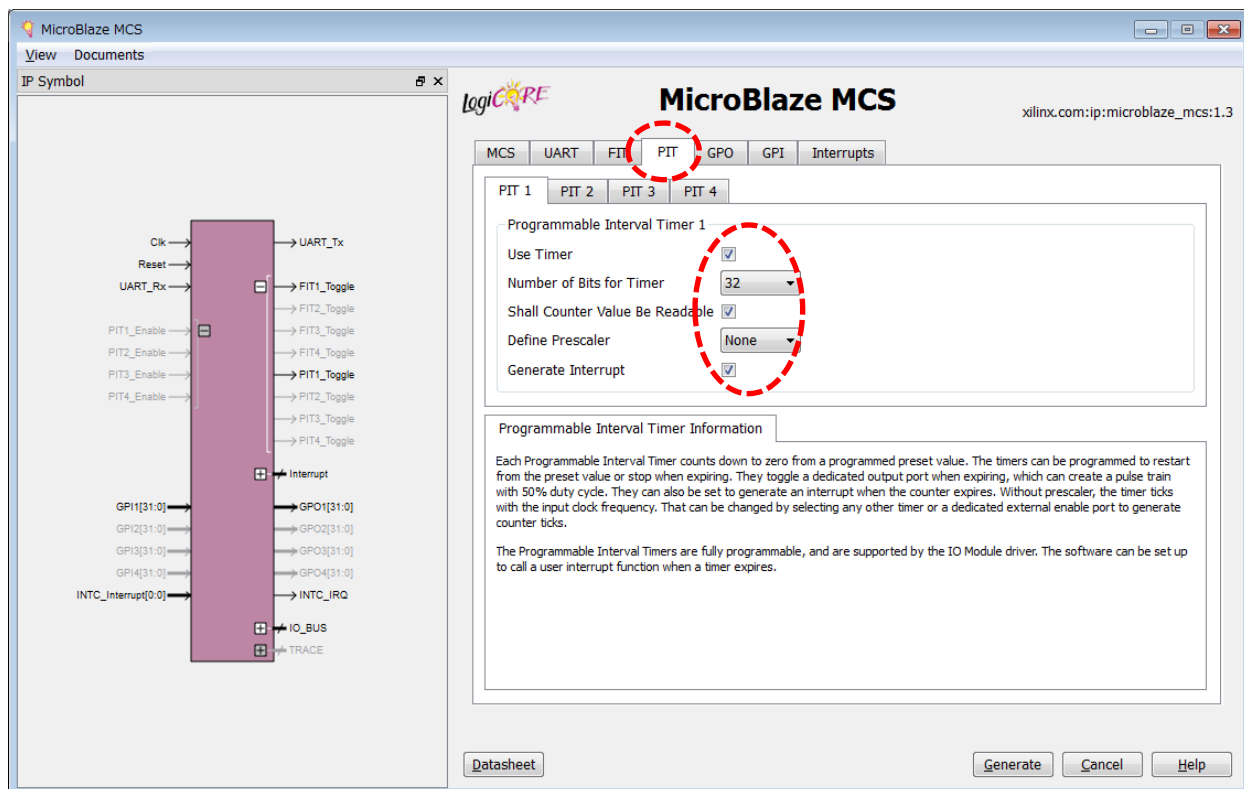


UART の設定

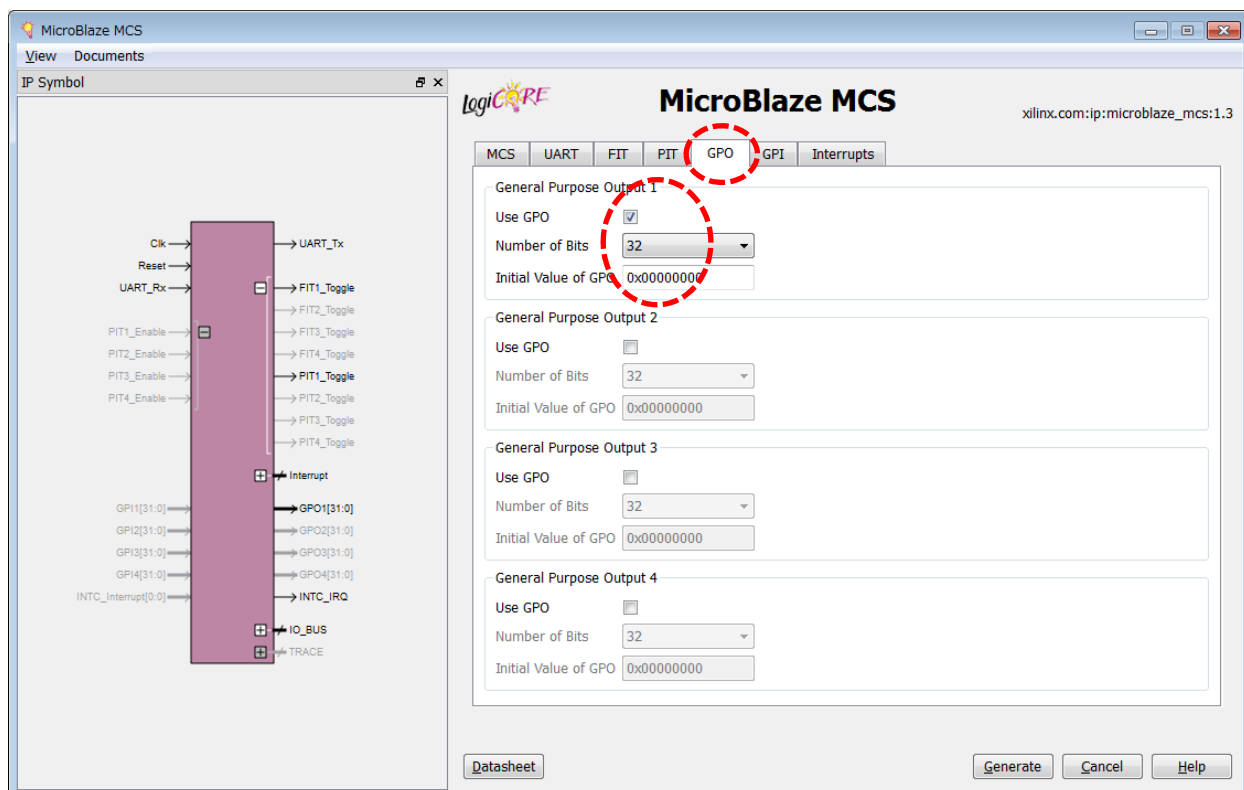


FIT の指定



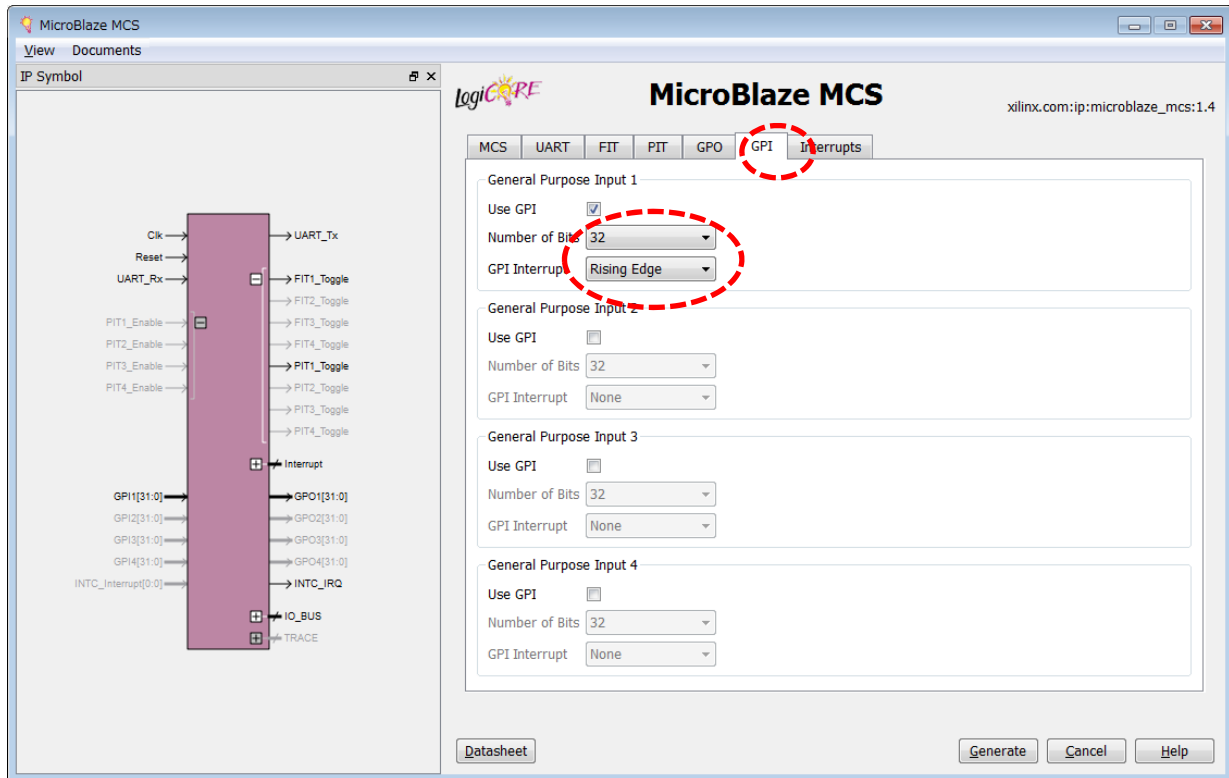


PIT の指定

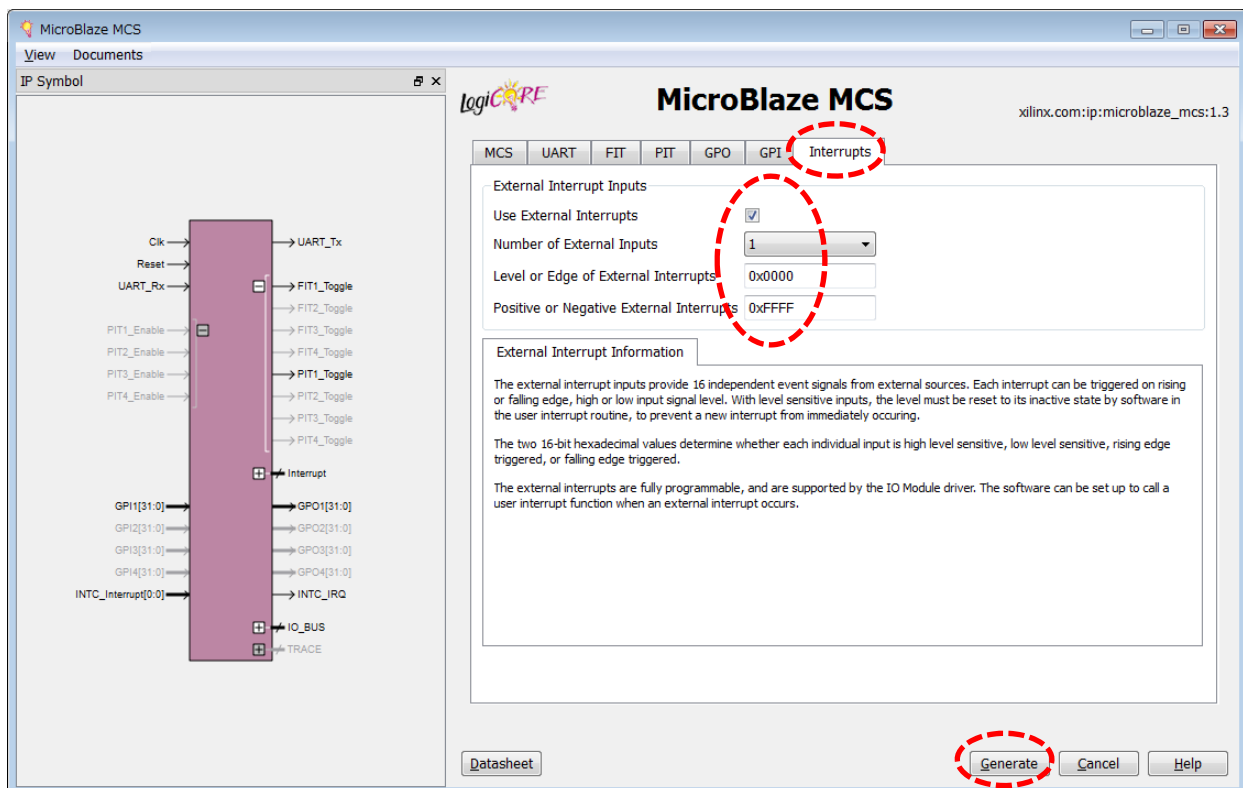


GPO の設定



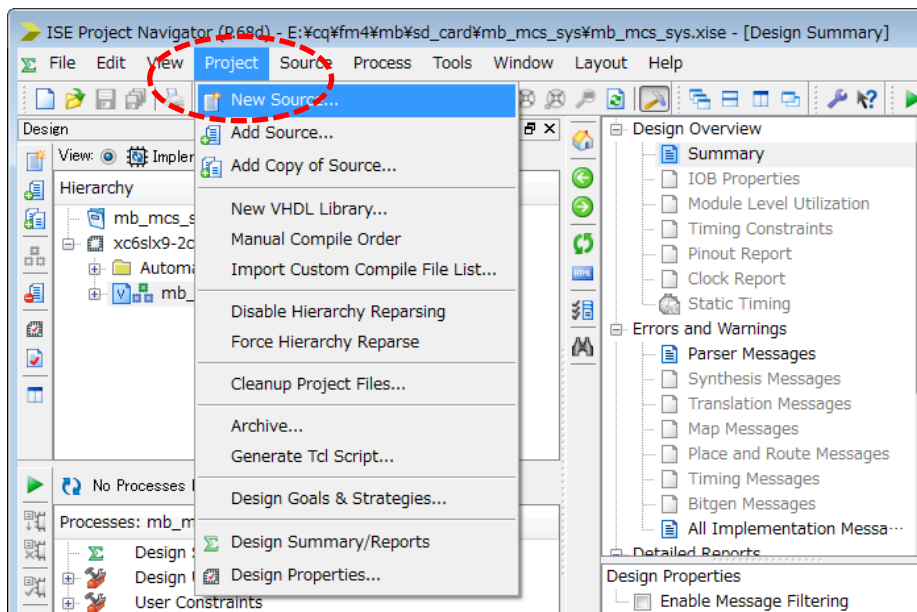


GPI の指定

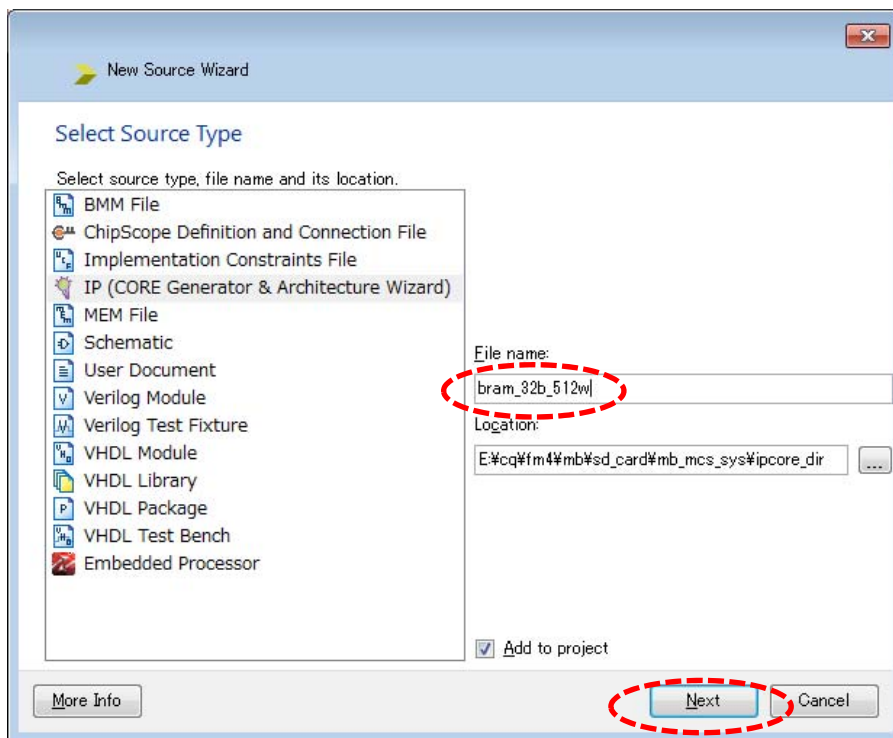


外部割り込みの設定後に Generate をクリックすると MicroBlaze MCS がプロジェクトに追加される



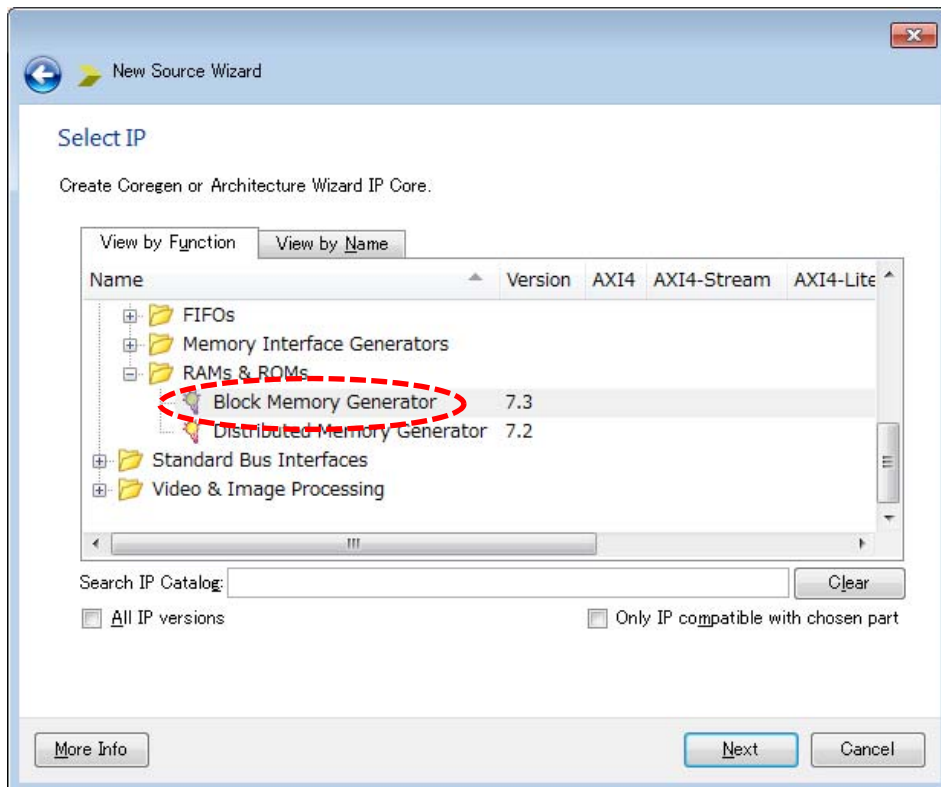


ブロック RAM の作成、Project→New Source を選択

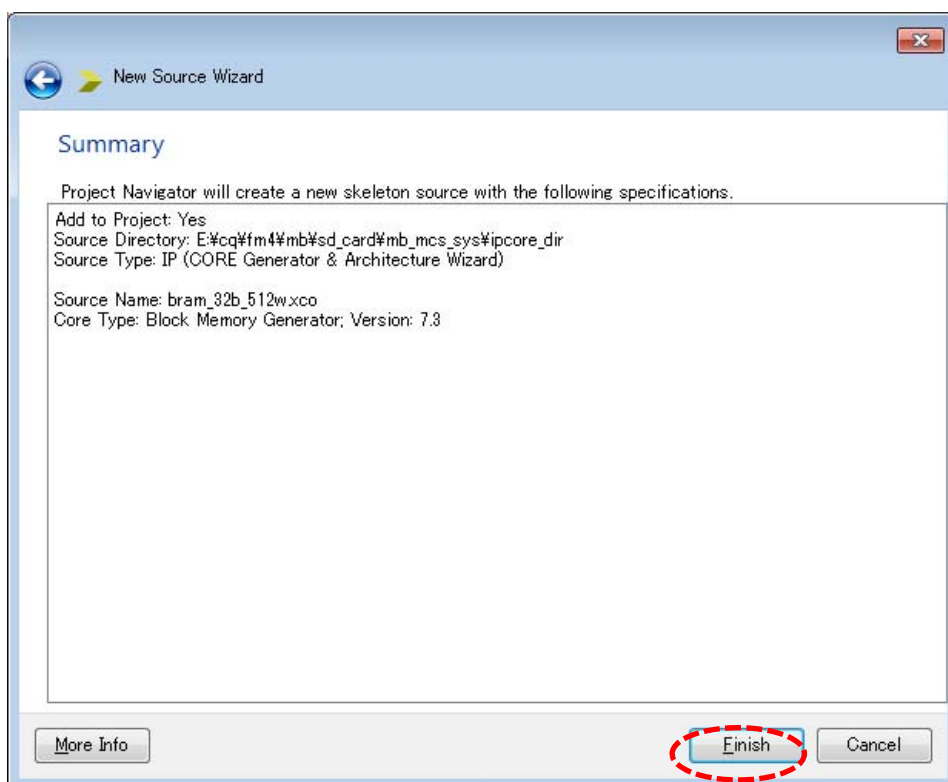


IP (CORE Gener...をクリックして選択、ファイル名に bram_32b_512w を指定、Next をクリック



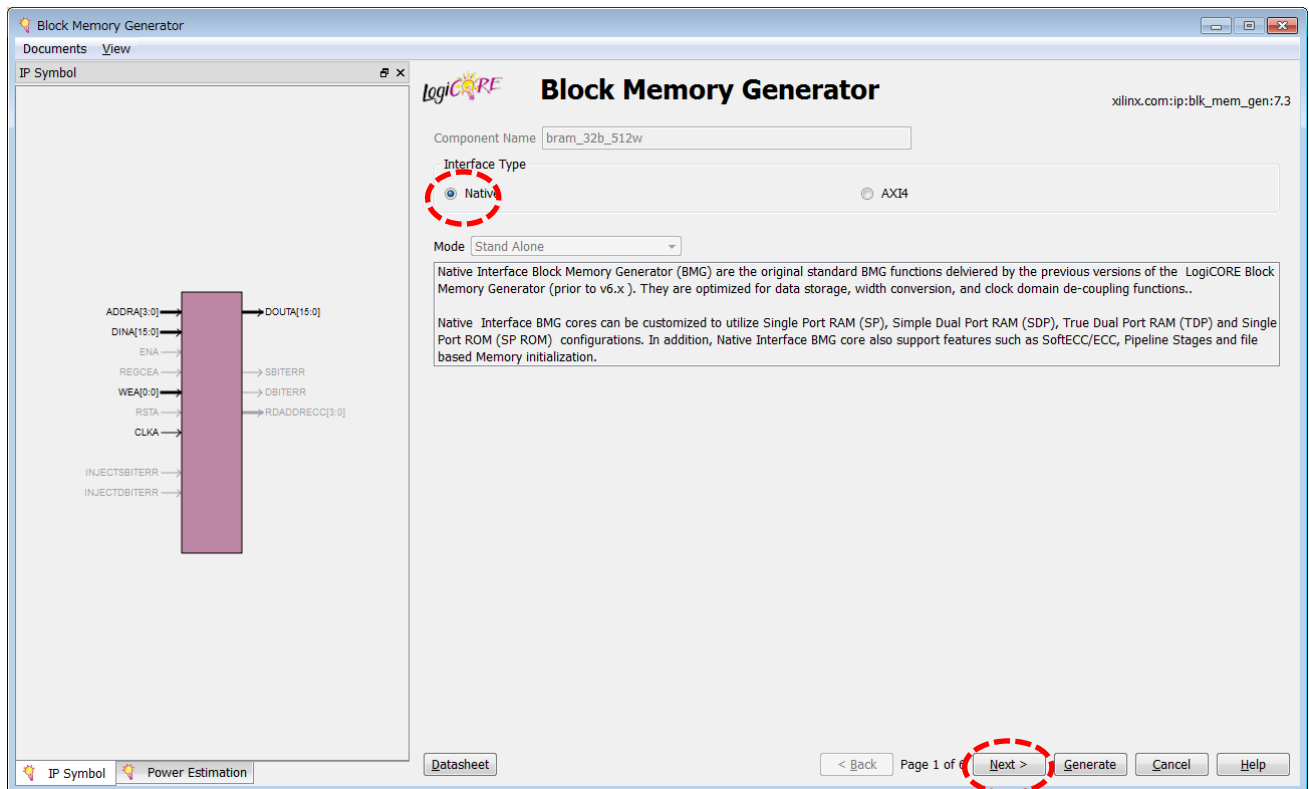


Block Memory Generator を指定

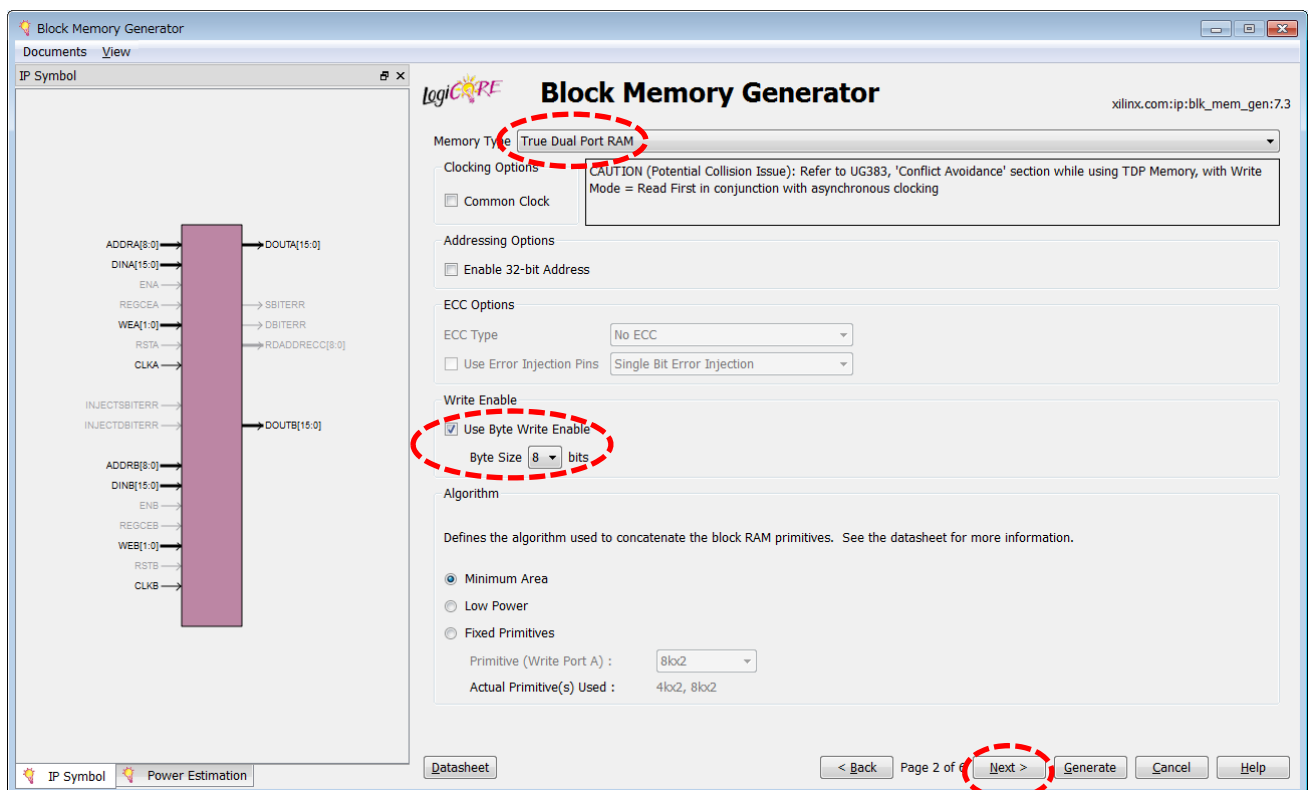


Finish をクリックすると CORE generator が起動



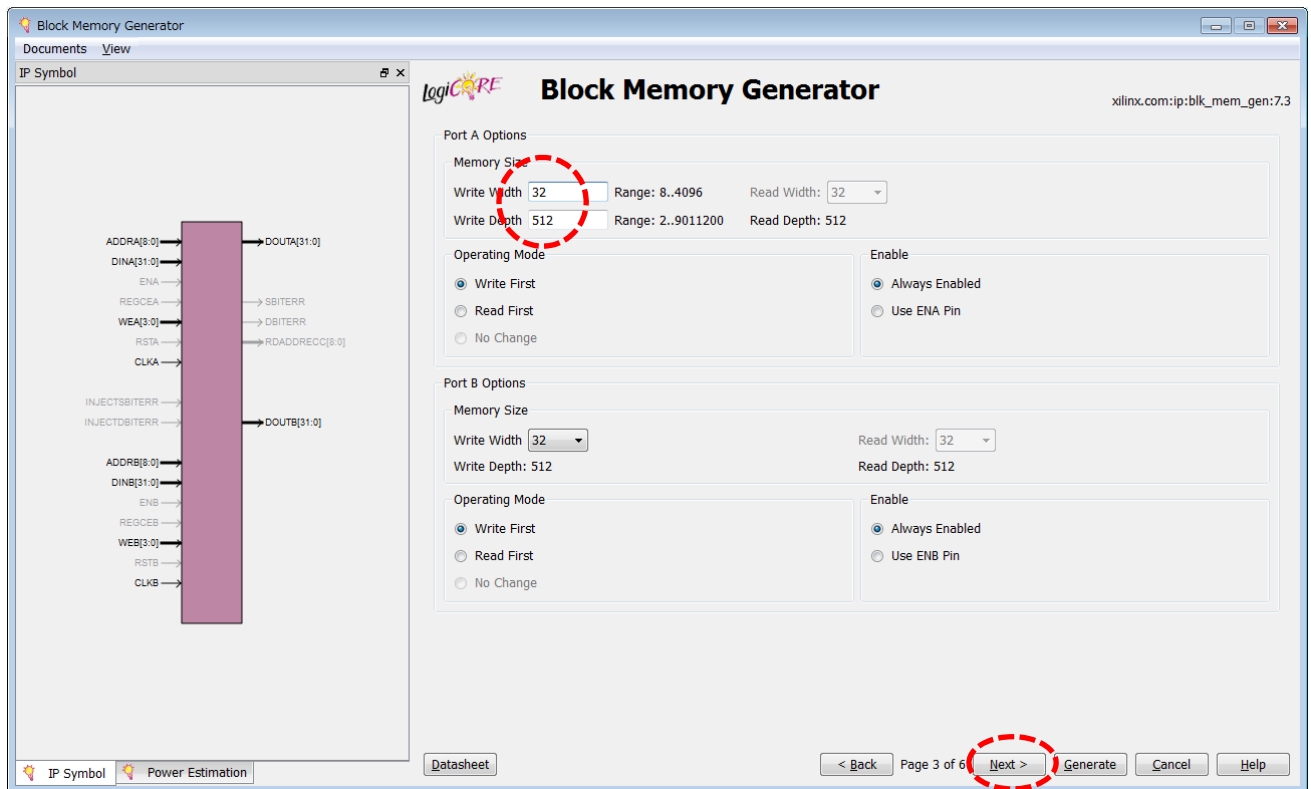


ブロック RAM インターフェースの設定

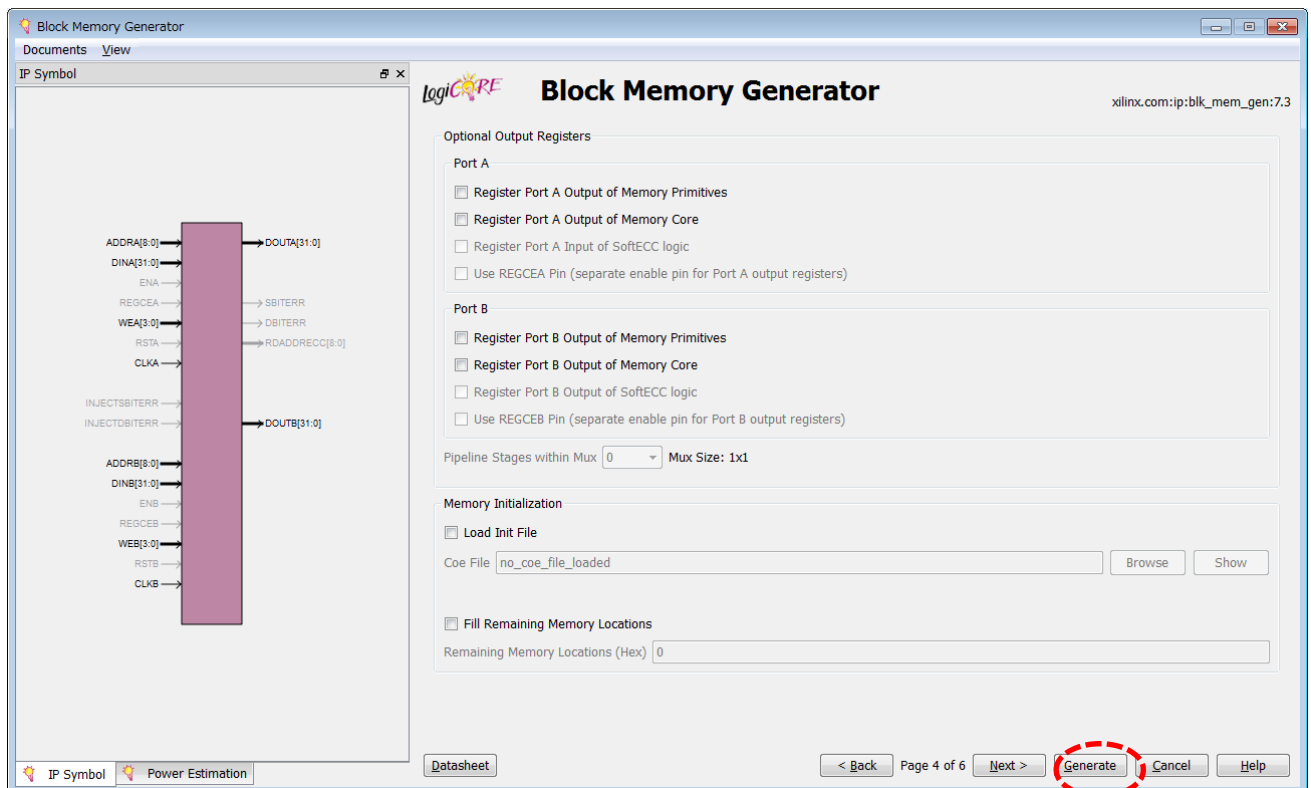


メモリのタイプ指定



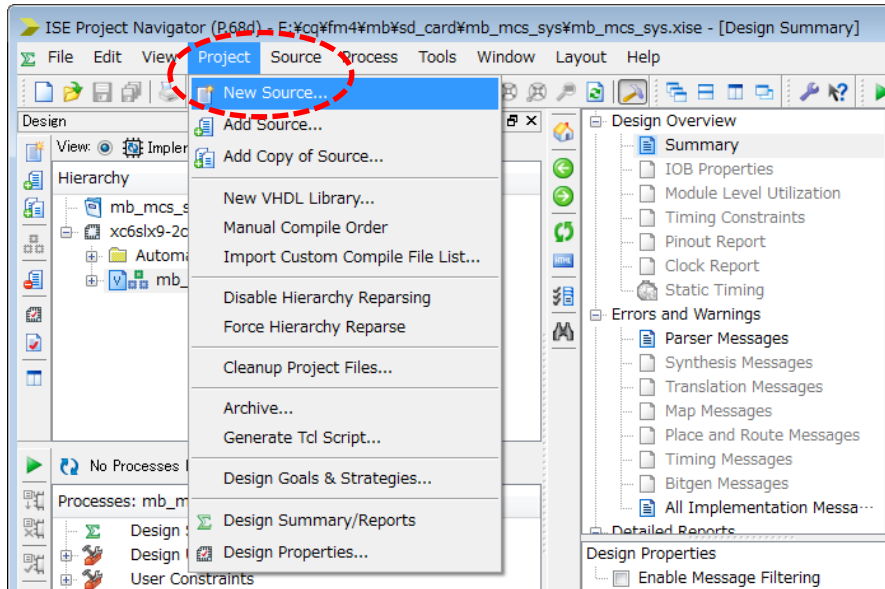


データ幅、データ量の指定

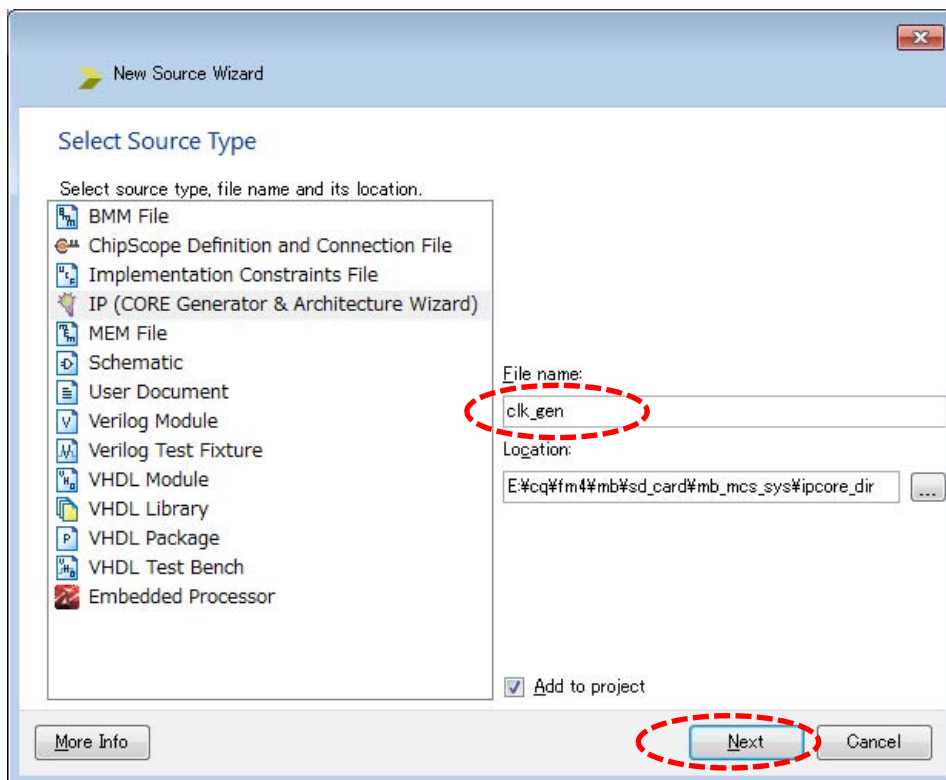


Generate をクリックで作成開始



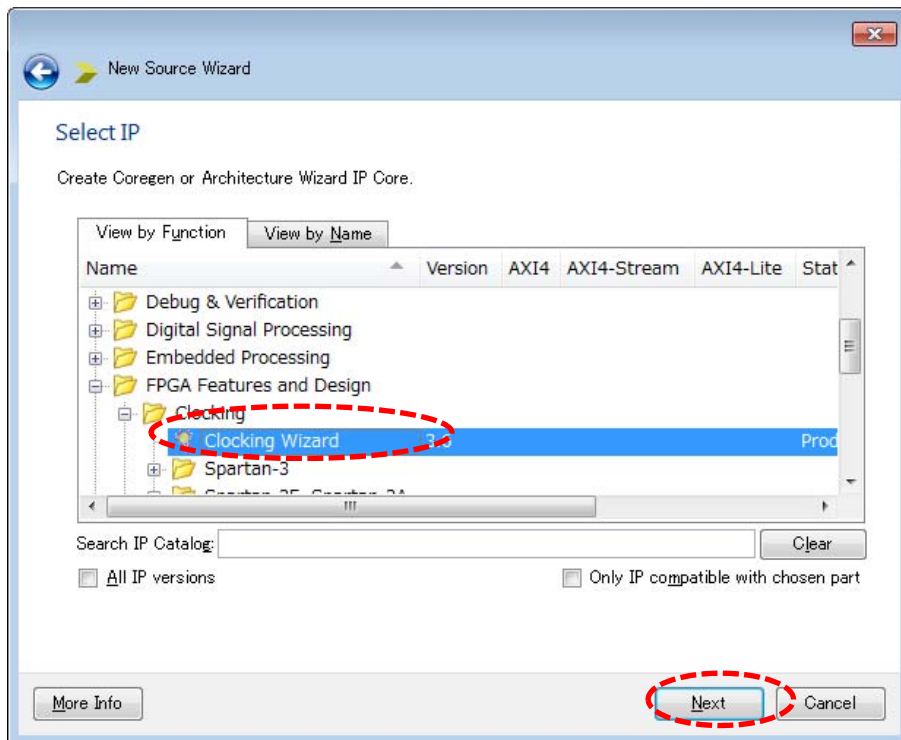


ブロック RAM の作成、Project→New Source を選択

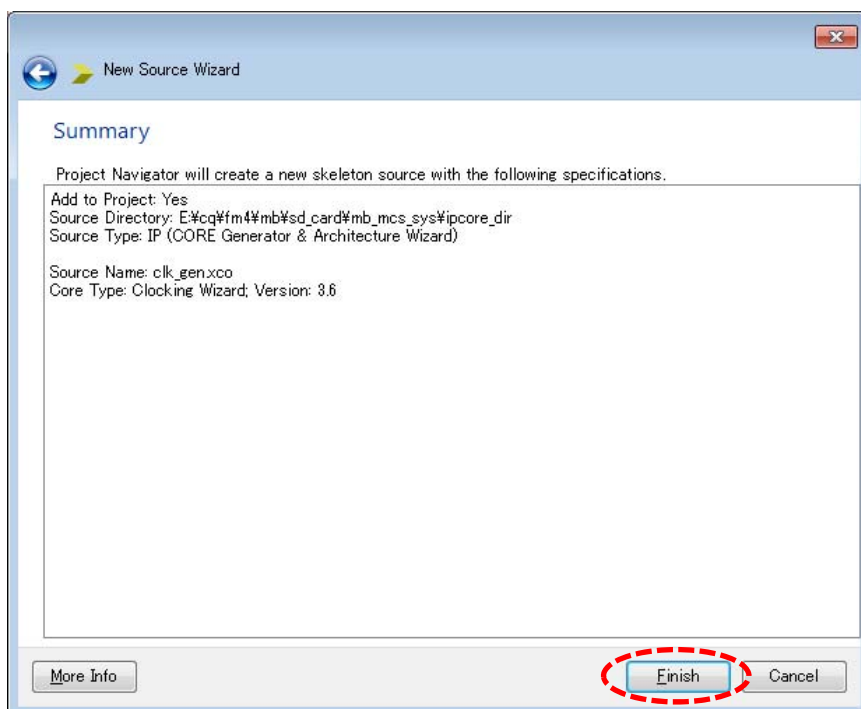


IP (CORE Gener...をクリックして選択、ファイル名に clk_gen を指定、Next をクリック





Clocking Wizard 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



Clocking Wizard

Documents View

IP Symbol

logiCORE

Clocking Wizard

xilinx.com:ip:clk_wiz:3.6

Component name:

Clocking Features

- ☒ Frequency synthesis
- ☒ Phase alignment (known phase relationship to input clock)
- ☐ Minimize power
- ☐ Dynamic phase shift
- ☐ Dynamic reconfiguration (in system output freq modification)

Jitter Optimization

- ☒ Balanced
- ☐ Minimize output jitter (low clock jitter filtering)
- ☐ Maximize input jitter filtering (allow larger input jitter)

Clock Manager Type

Mode

- ☒ Auto Selection (Recommended: Wizard selects primitive)
- ☐ Manual Selection (User selects primitive)

Input Jitter Unit

- ☒ UI
- ☐ PS

Input Clock Information

Input Clock	Input Freq (MHz)	Input Jitter	Source
primary	Value: 100.000 Valid Range: 5.000 - 558.659	0.010	Single ended clock capable pin

IP Symbol Resource Estimation

Datasheet < Back Page 1 of 6 Next > Generate Cancel Help

入力クロック周波数：100MHz の指定



Clocking Wizard

Documents View

IP Symbol

logiCORE

Clocking Wizard

xilinx.com:ip:clk_wiz:3.6

Output Clock Settings

The phase is calculated relative to the active input clock.

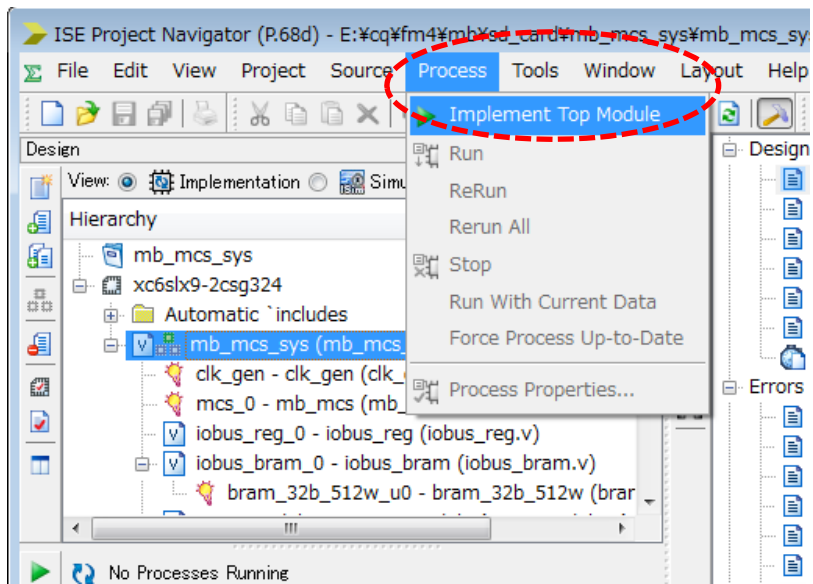
Output Clock	Output Freq (MHz)		Phase (degrees)		Duty Cycle (%)		Drives	Use Fine Ps
	Requested	Actual	Requested	Actual	Requested	Actual		
<input checked="" type="checkbox"/> CLK_OUT1	100.000	100.000	0.000	0.000	50.000	50.0	BUFG	<input type="checkbox"/>
<input checked="" type="checkbox"/> CLK_OUT2	50.000	50.000	0.000	0.000	50.000	50.0	BUFG	<input type="checkbox"/>
<input checked="" type="checkbox"/> CLK_OUT3	20.000	20.000	0.000	0.000	50.000	50.0	BUFG	<input type="checkbox"/>
<input checked="" type="checkbox"/> CLK_OUT4	10.000	10.000	0.000	0.000	50.000	50.0	BUFG	<input type="checkbox"/>
<input type="checkbox"/> CLK_OUT5	100.000	N/A	0.000	N/A	50.000	N/A	BUFG	<input type="checkbox"/>
<input type="checkbox"/> CLK_OUT6	100.000	N/A	0.000	N/A	50.000	N/A	BUFG	<input type="checkbox"/>

IP Symbol Resource Estimation

Datasheet < Back Page 2 of 6 Next > Generate Cancel Help

出力周波数（CLK_OUT1=100MHz，CLK_OUT1=50MHz，CLK_OUT2=20MHz，CLK_OUT3=10MHz）の設定、Generate をクリックで作成





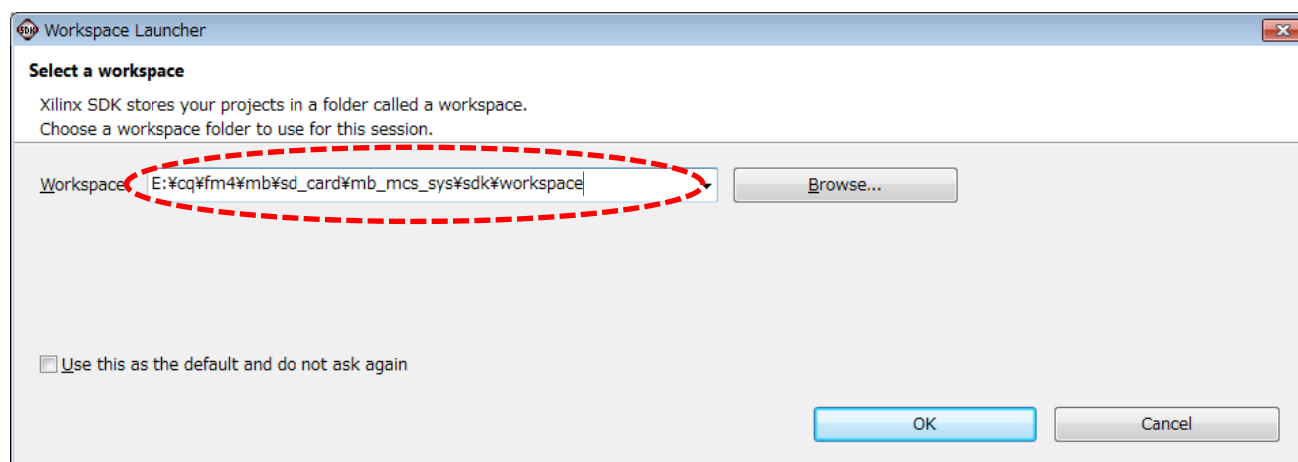
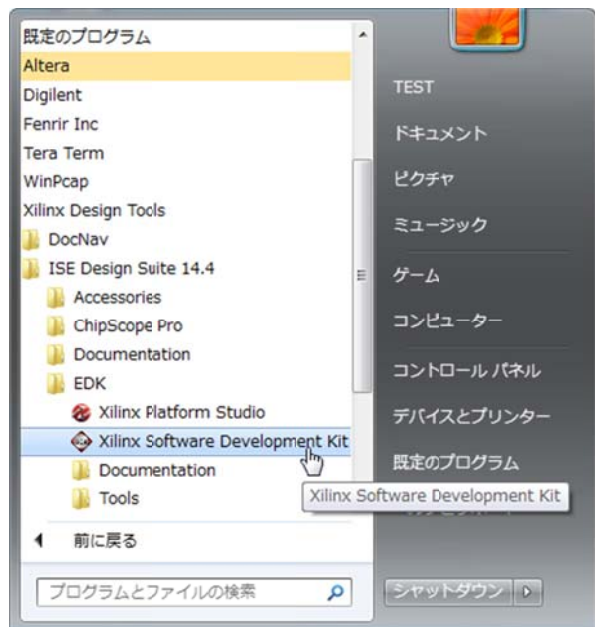
インプリメンテーションの実行、Process→Implement Top Module をクリック



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

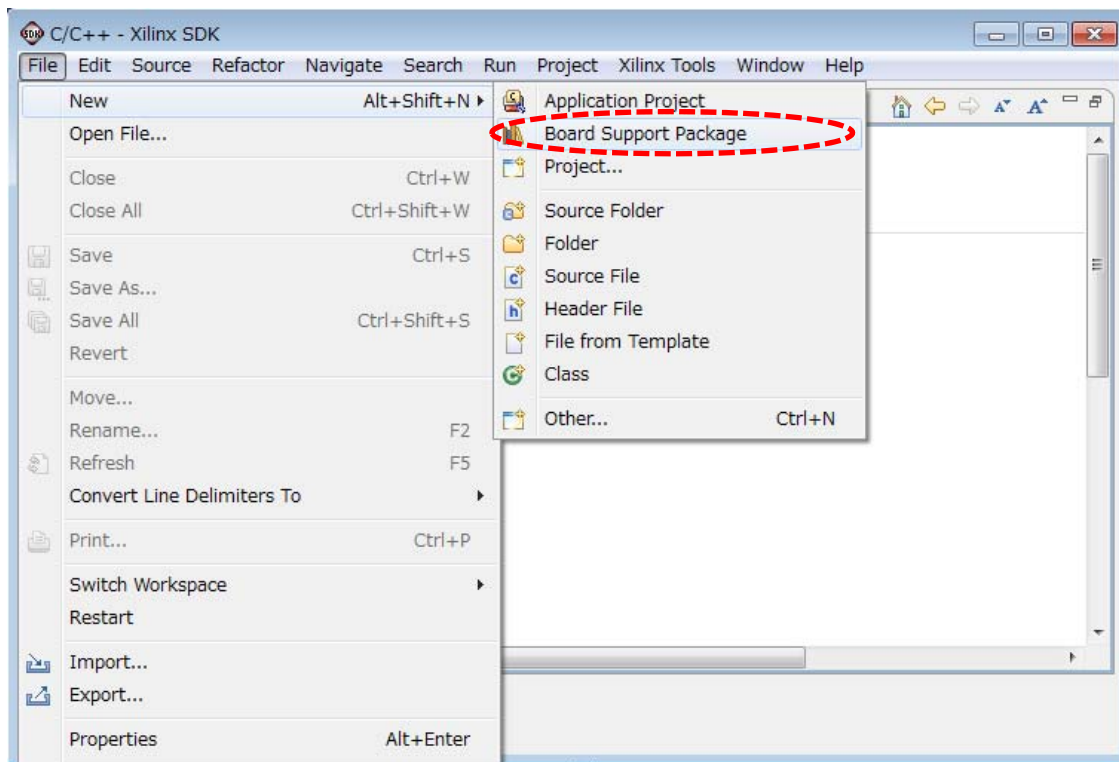
・ SDK の起動

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



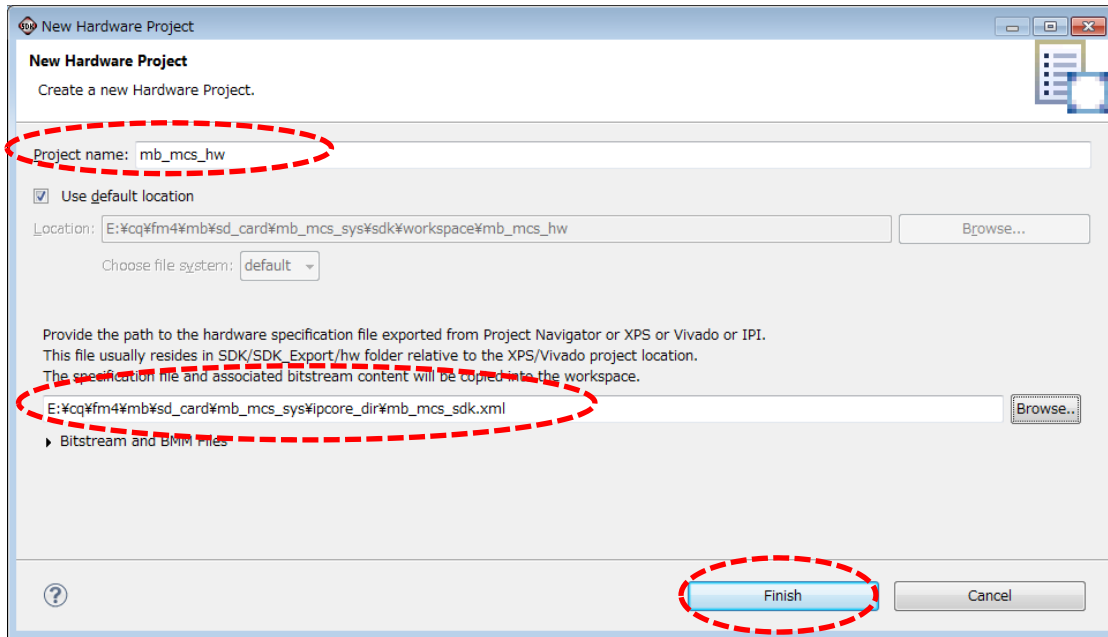


新規のボードサポートパッケージ作成、File→New→Board Support Package



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

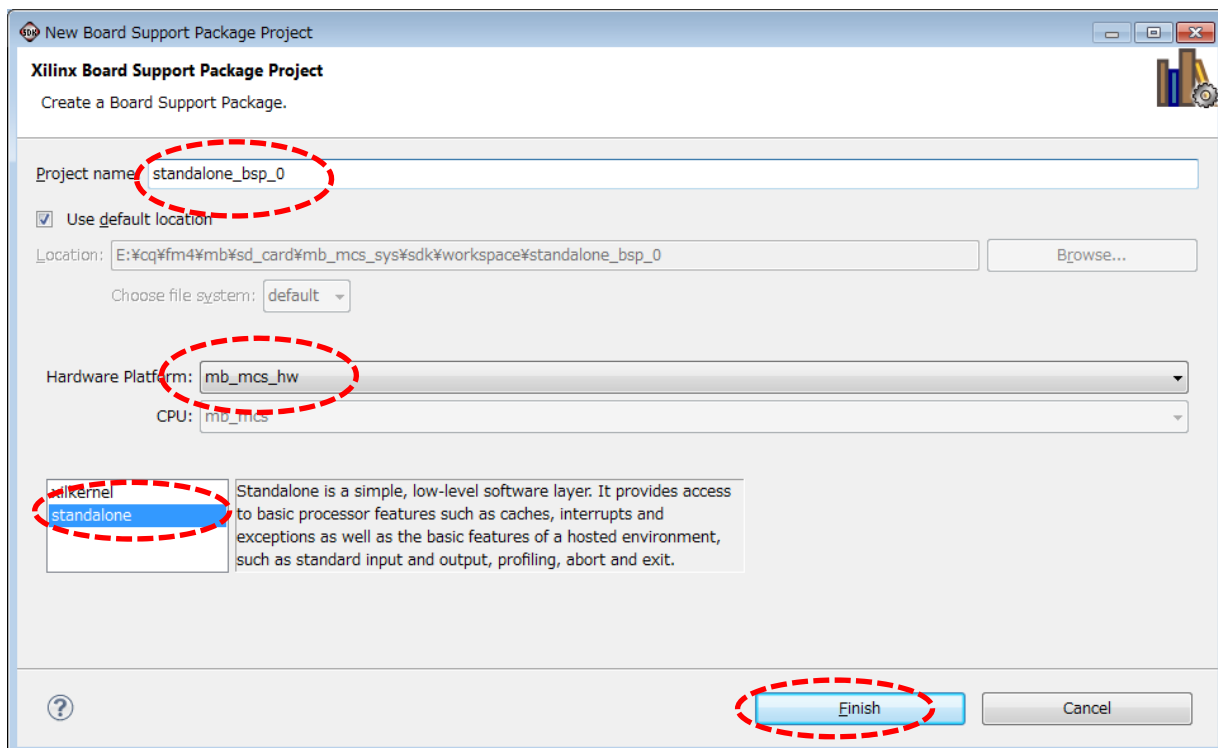




SDK の管理するハードウェアプロジェクト名指定

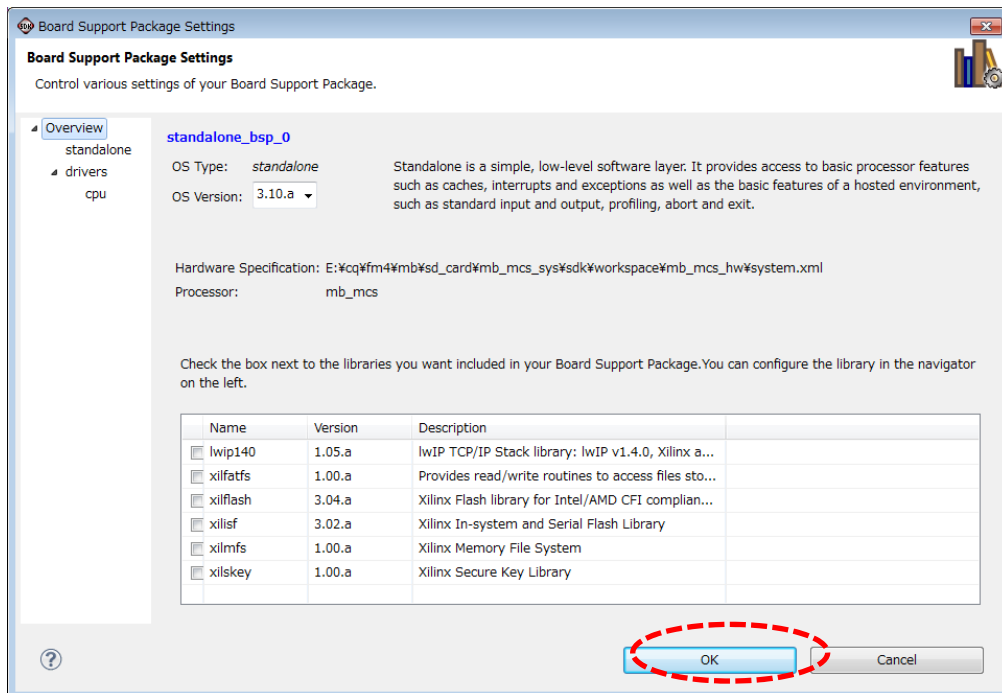
xml ファイル (設計ファルダ/ipcore_dir/mb_msc. sdk. xml) 指定

Finish をクリック

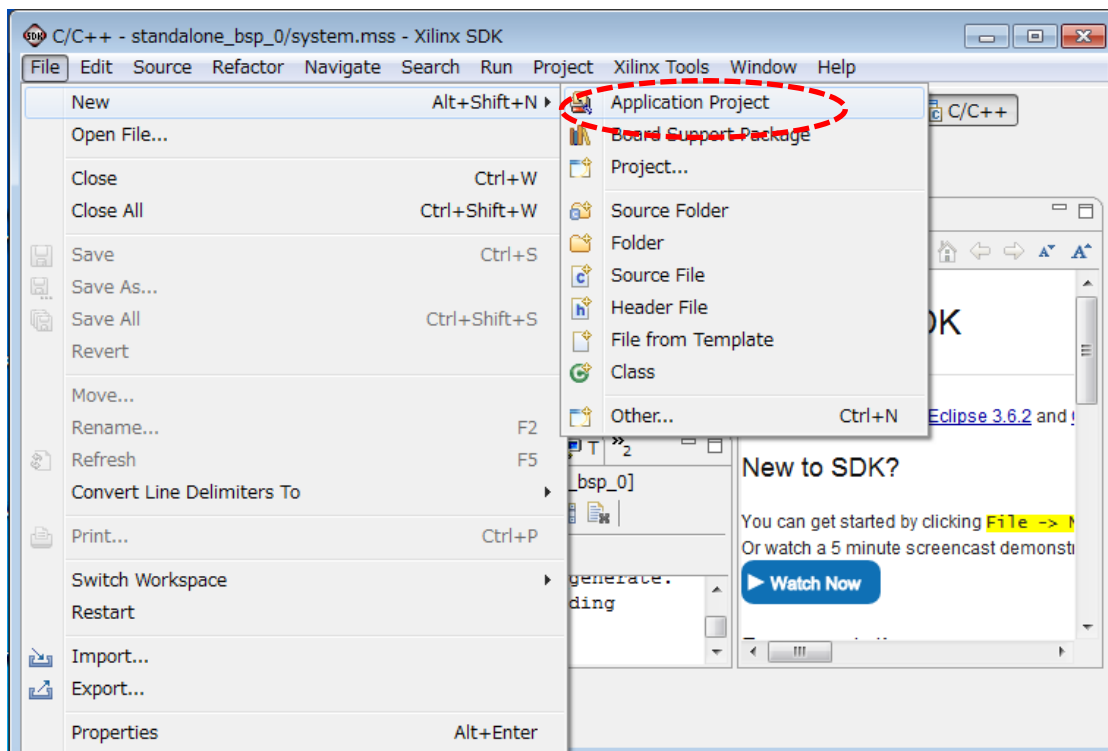


ボードサポートパッケージの定義



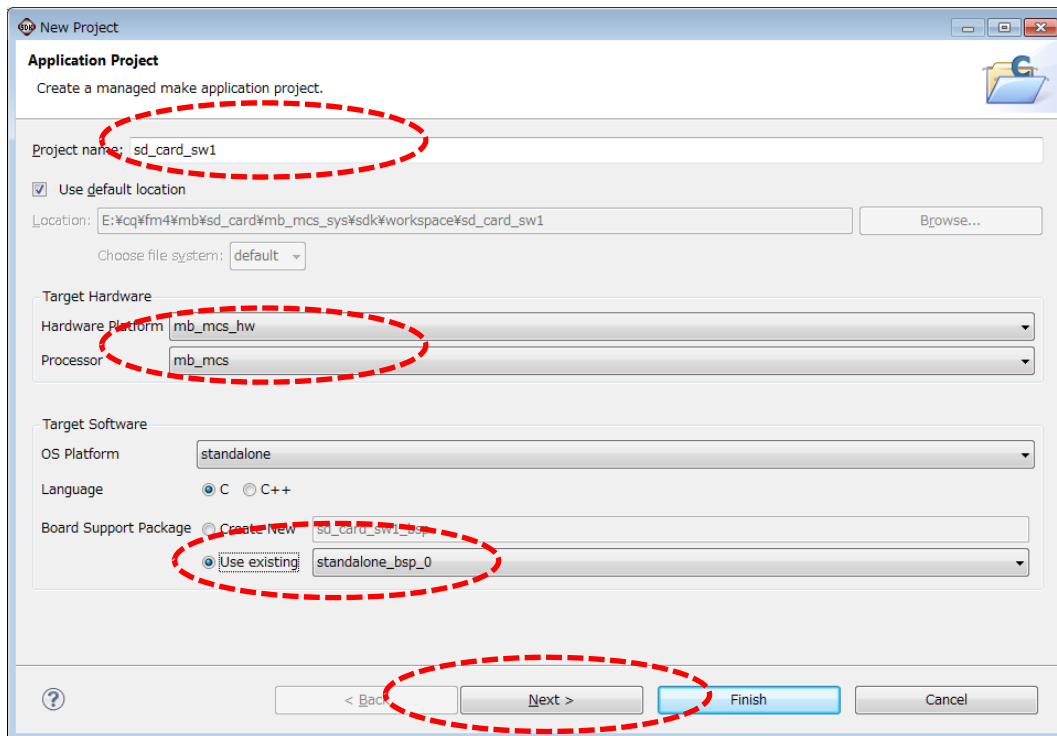


ボードサポートパッケージのオプション定義, OK をクリック



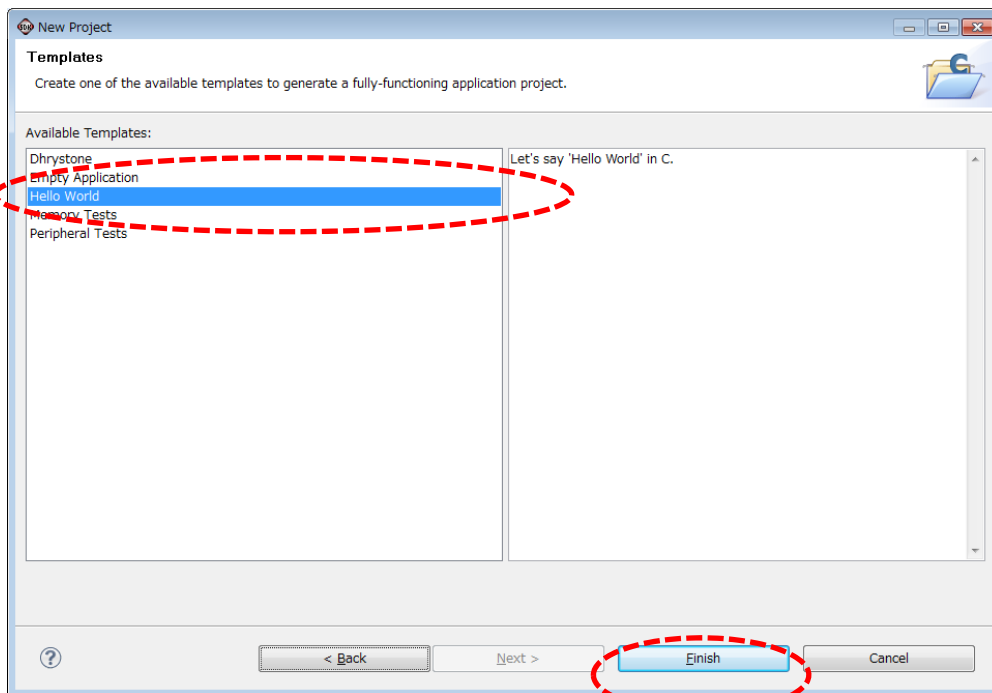
新規のソフトウェアプロジェクトの作成





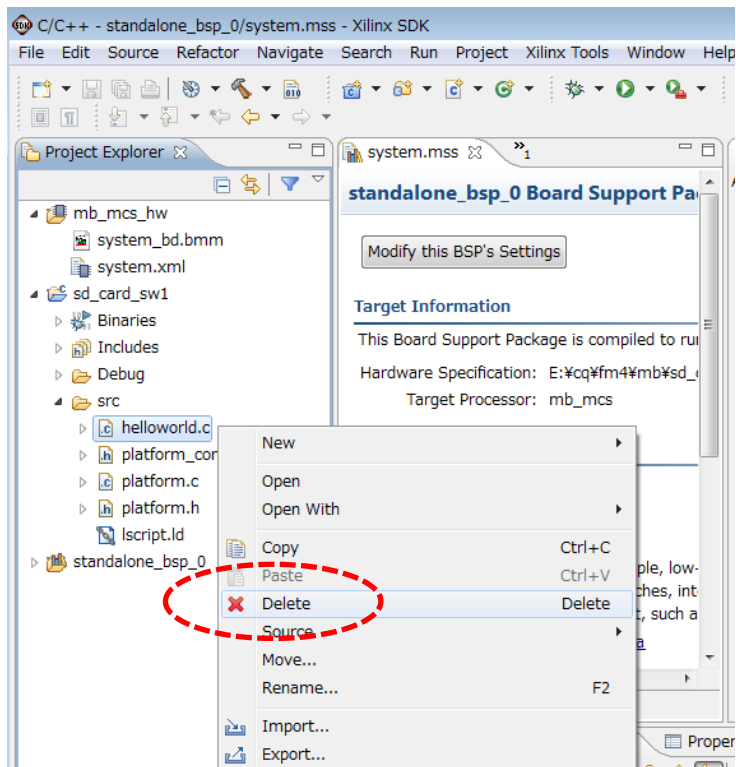
ソフトウェアプロジェクト名指定:sd_card_sw1

Next をクリック

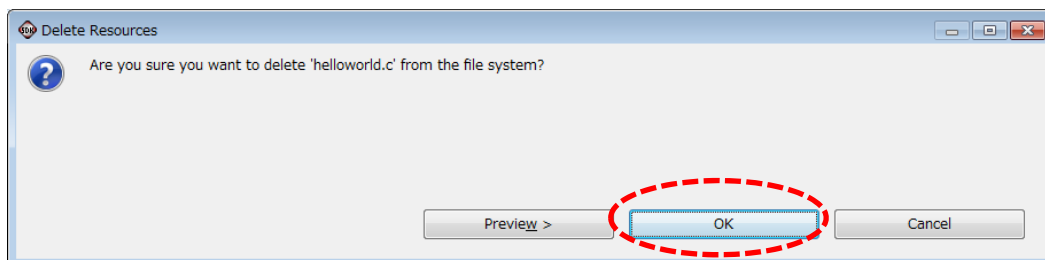


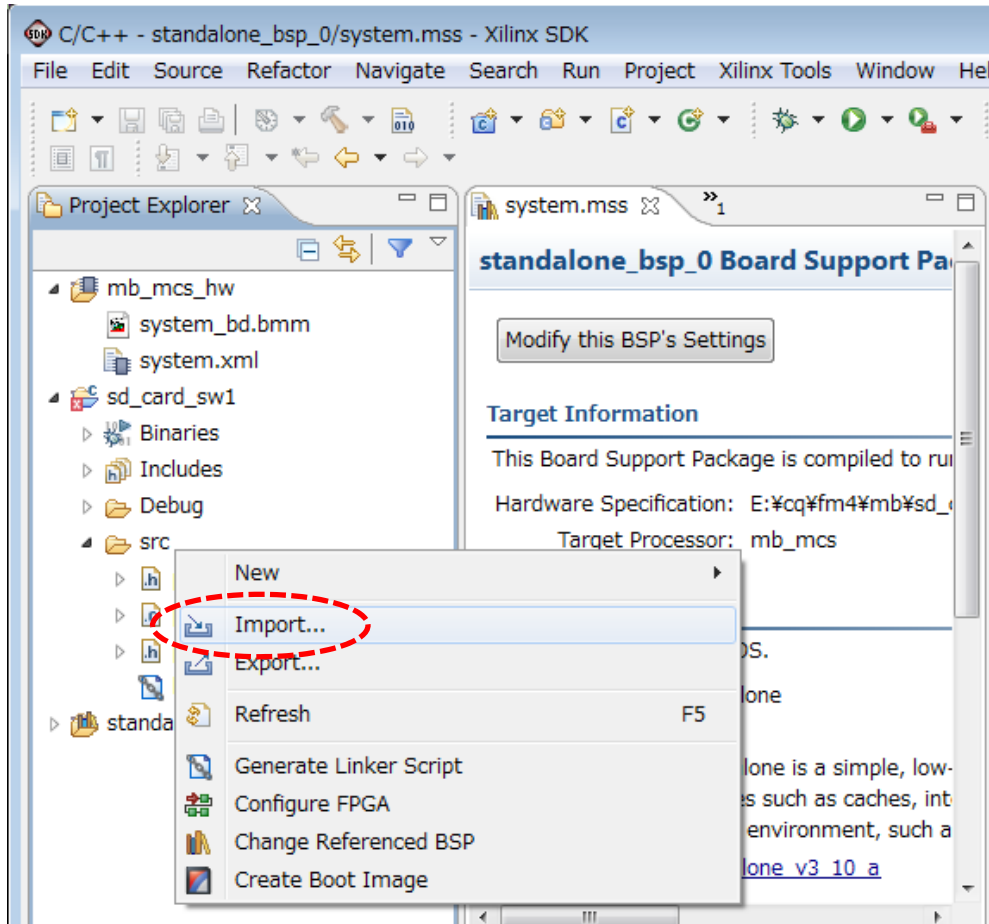
テンプレートに Hello World 選択後、Finish をクリックでソフトウェアプロジェクトが作成される



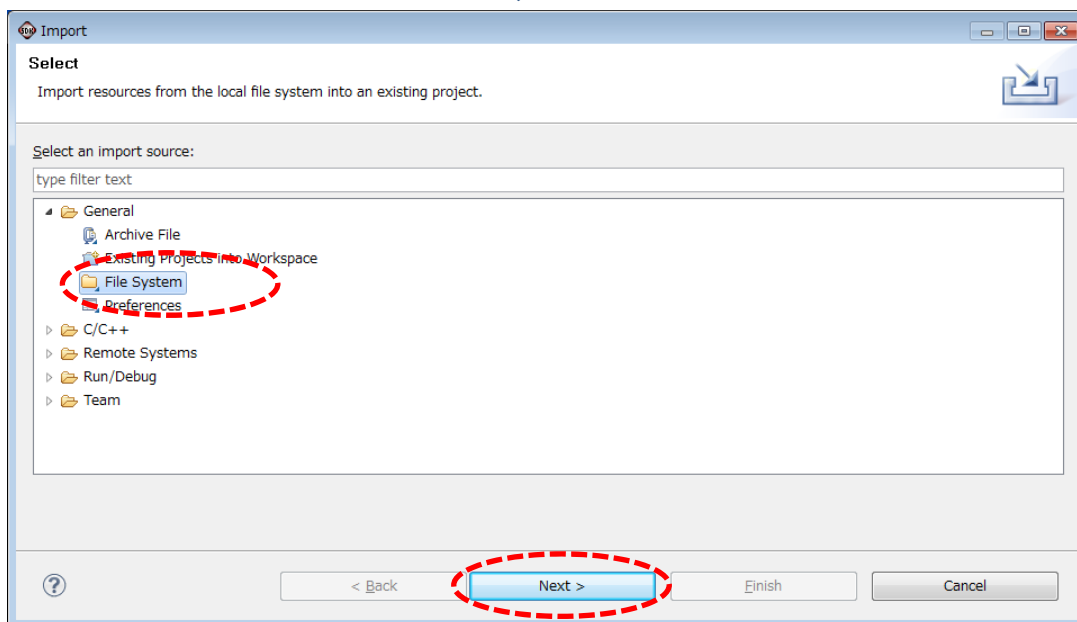


helloworld.c を削除



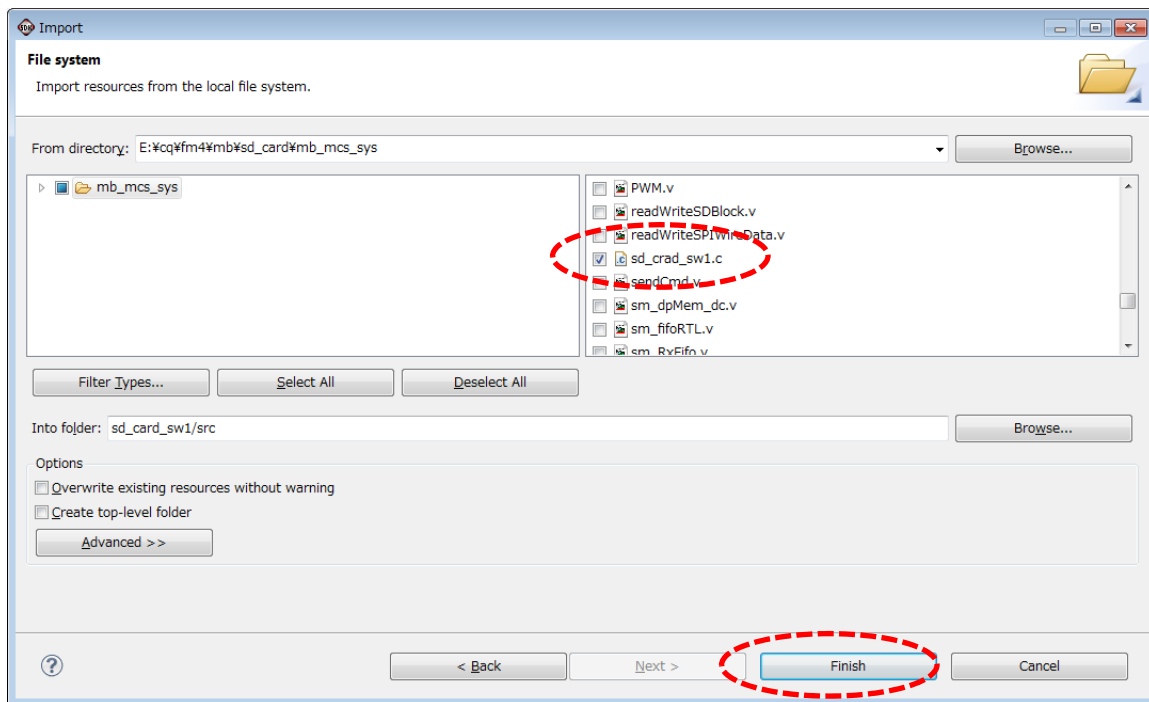


C ソースコードのインポート、sd_card_sw1 の src の上で右クリック、import 選択



General→File System 選択



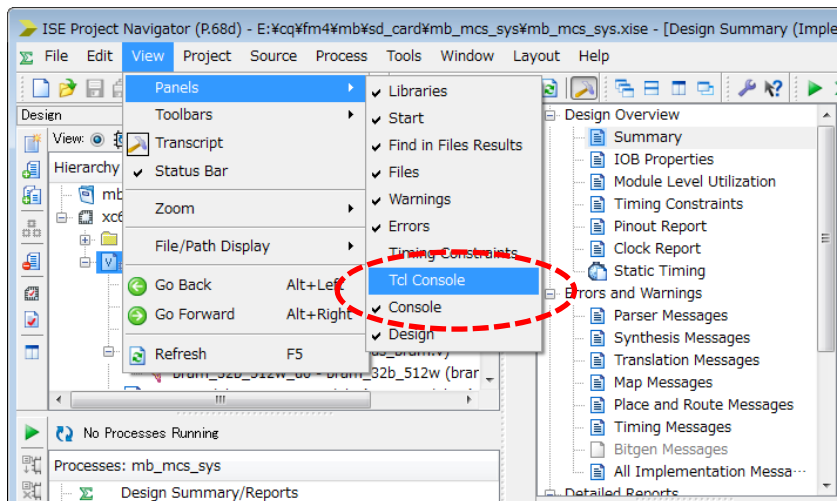


解凍データの sd_card_sw1.c を選択

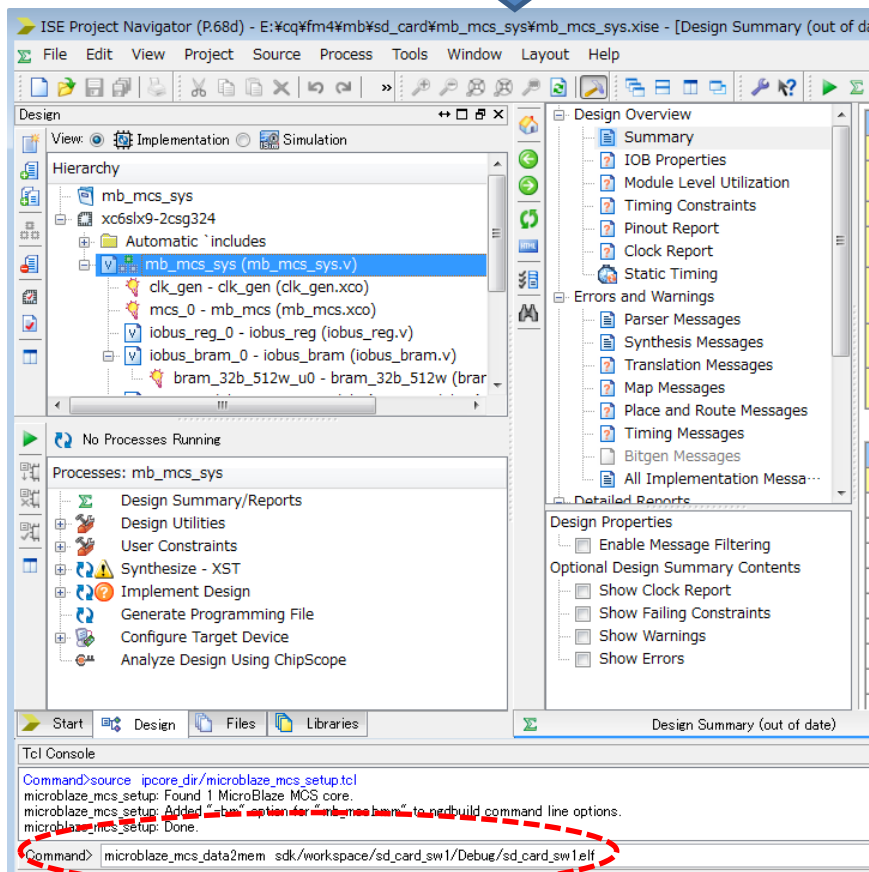
sd_card_sw1.c が登録されて、ビルドが実行されます。

実行ソフトウェアのファイル（sdk/workspace/sd_card_sw1/Debug/sd_card_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



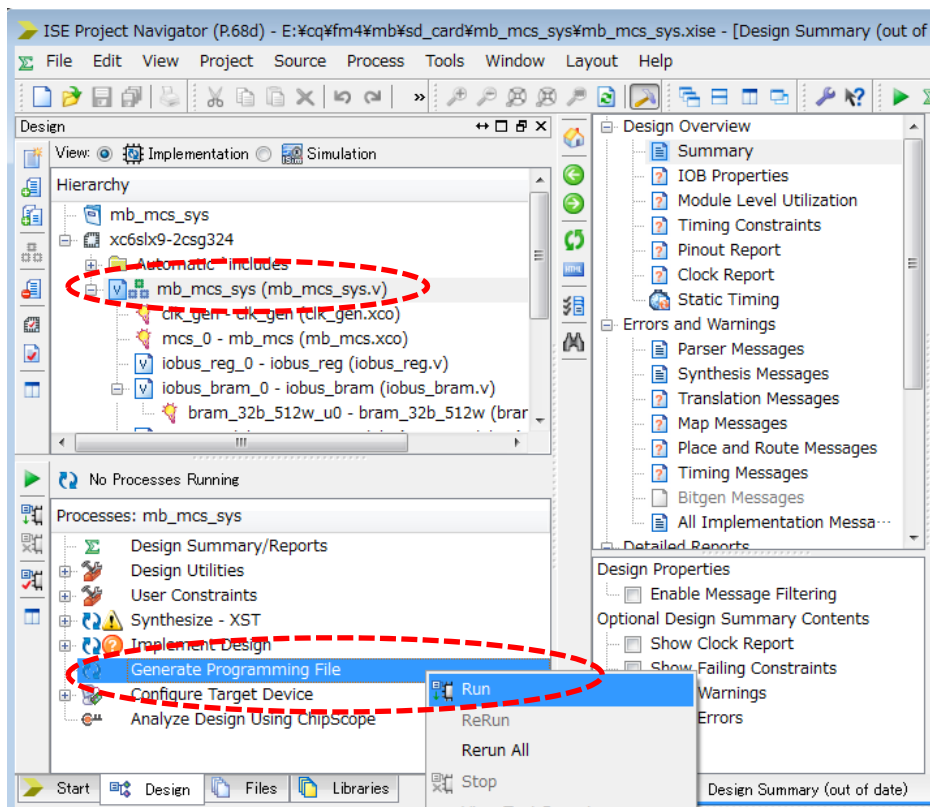
Tcl コマンドを使って実行ソフトウェアのファイルをMicroBlazeMCSのメモリの初期値定義ファイルに変換する。

Tcl コマンド

```
source ipcore_dir/microblaze_mcs_setup.tcl[Enter]
```

```
microblaze_mcs_data2mem sdk/workspace/sd_card_sw1/Debug/sd_card_sw1.elf[Enter]
```





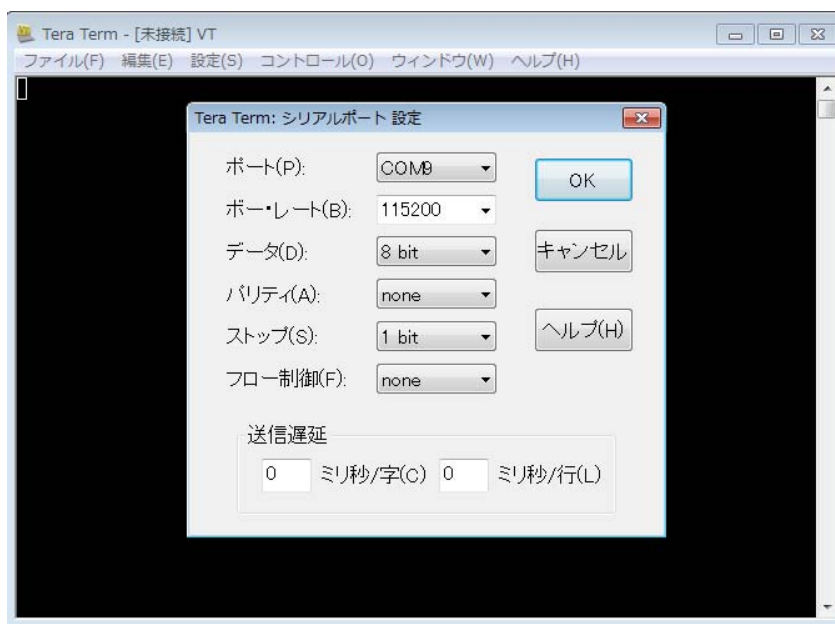
FPGA へ書き込む mb_mcs_sys.bit 作成



実機の動作確認をします。

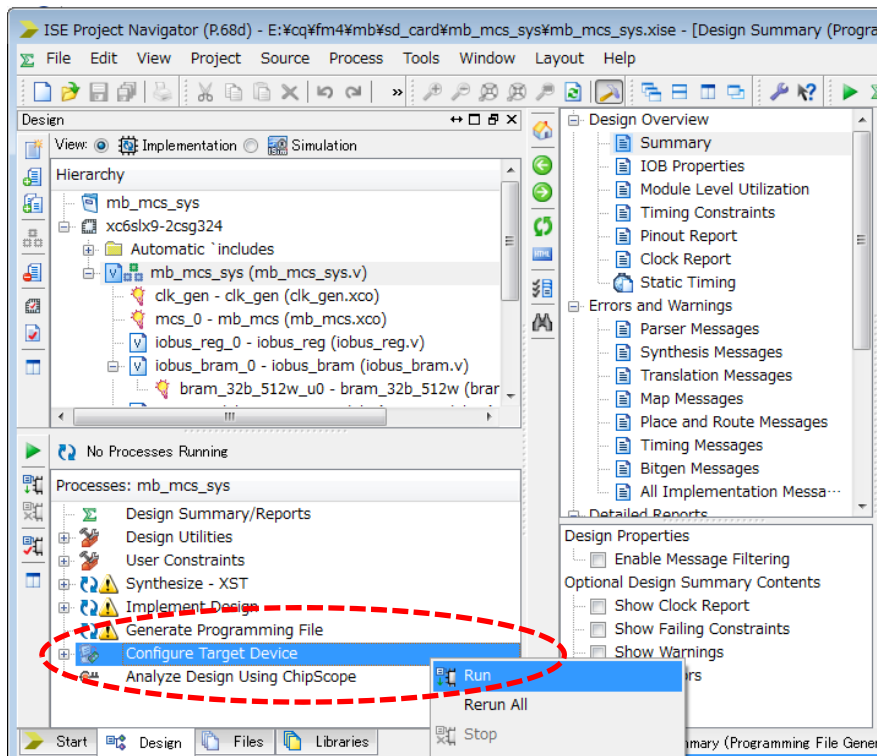
LX9 マイクロボードのプログラミング用 USB と UART 用 USB をそれぞれ PC の USB ポートに接続で接続します。

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

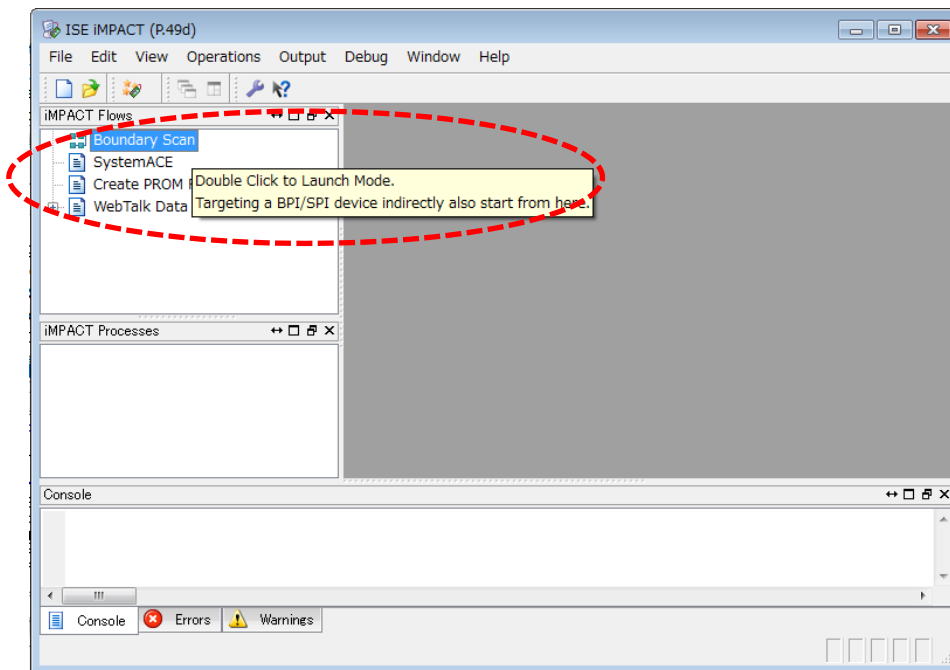


ターミナルソフトを立ち上げる。ポート番号は自分 PC で割り当てられて番号を使う、



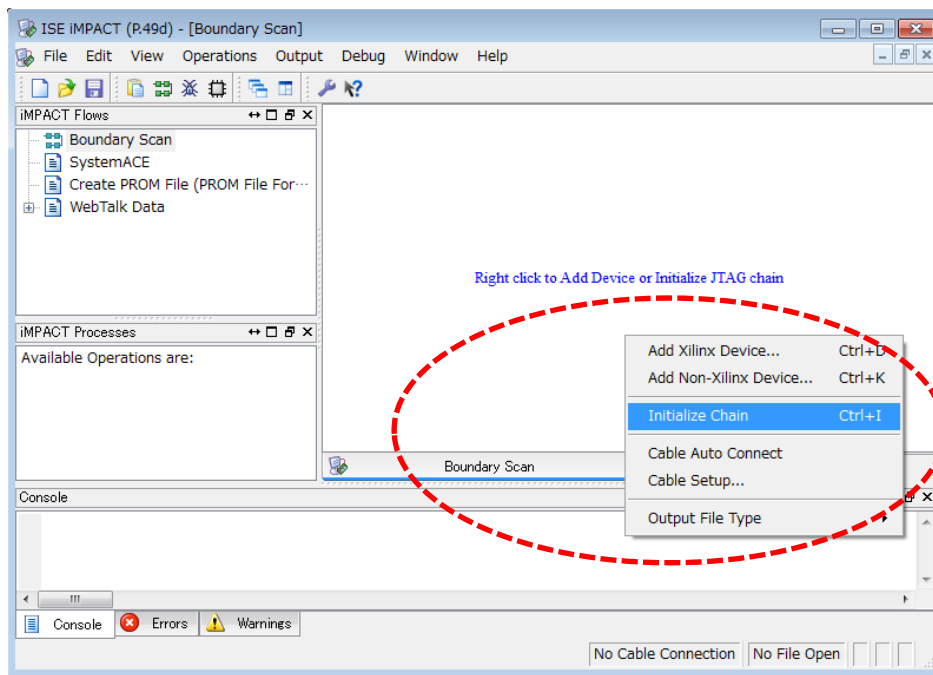


projectNavigator で iMPACT を起動

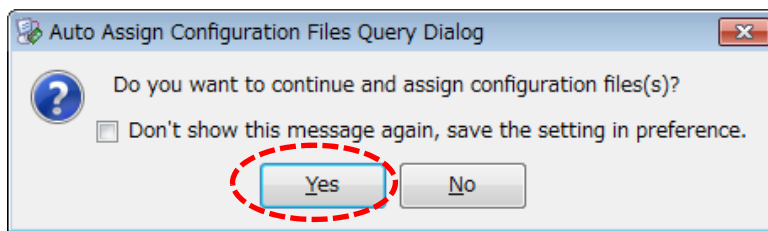


BoundaryScan モードにする



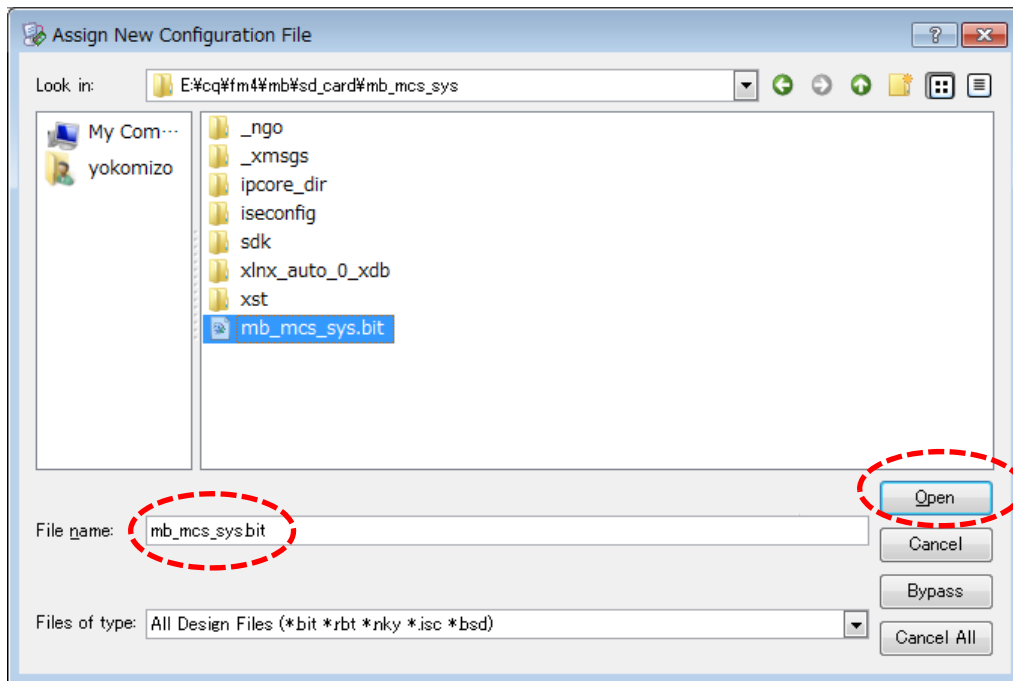


BoundaryScan のウィンドをマウス右ボタン押して、Initialize Chain 選択して FPAG を検出する

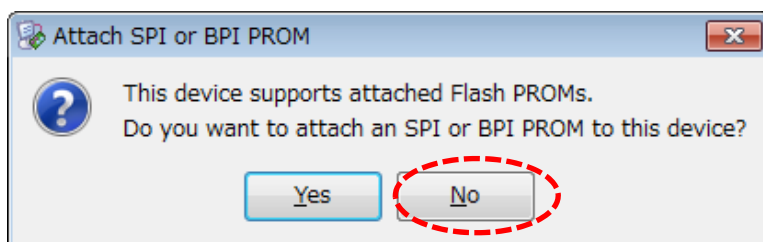


Yes で FPGA に書き込むファイルを指定する



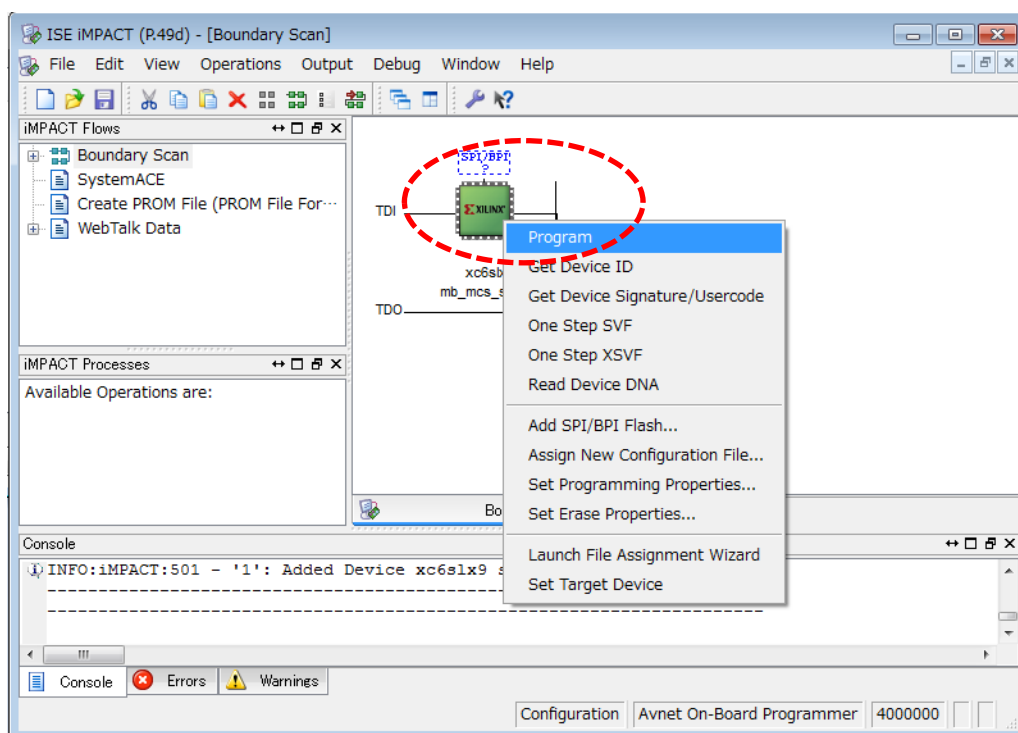
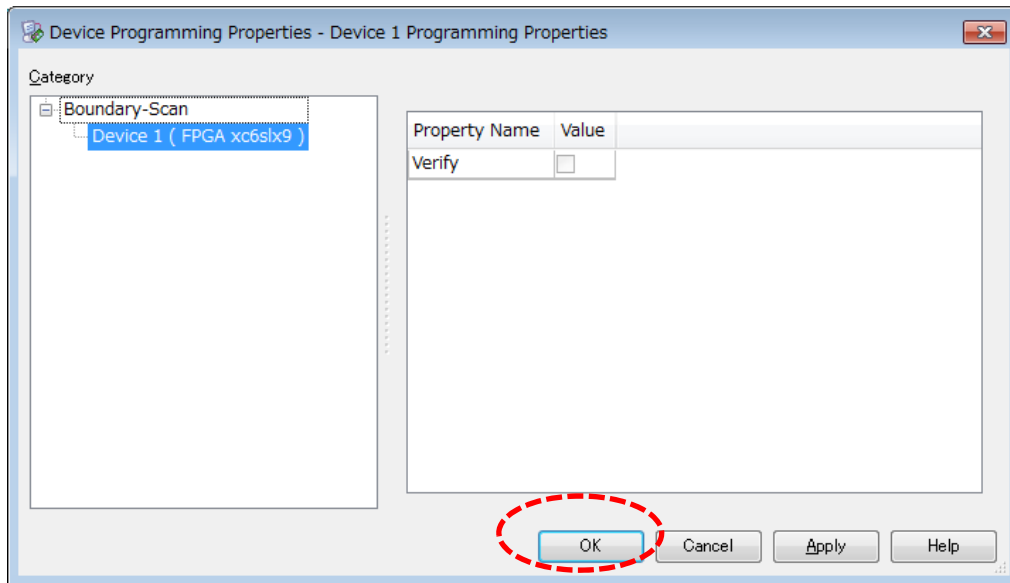


mb_mcs_sys.bit を指定



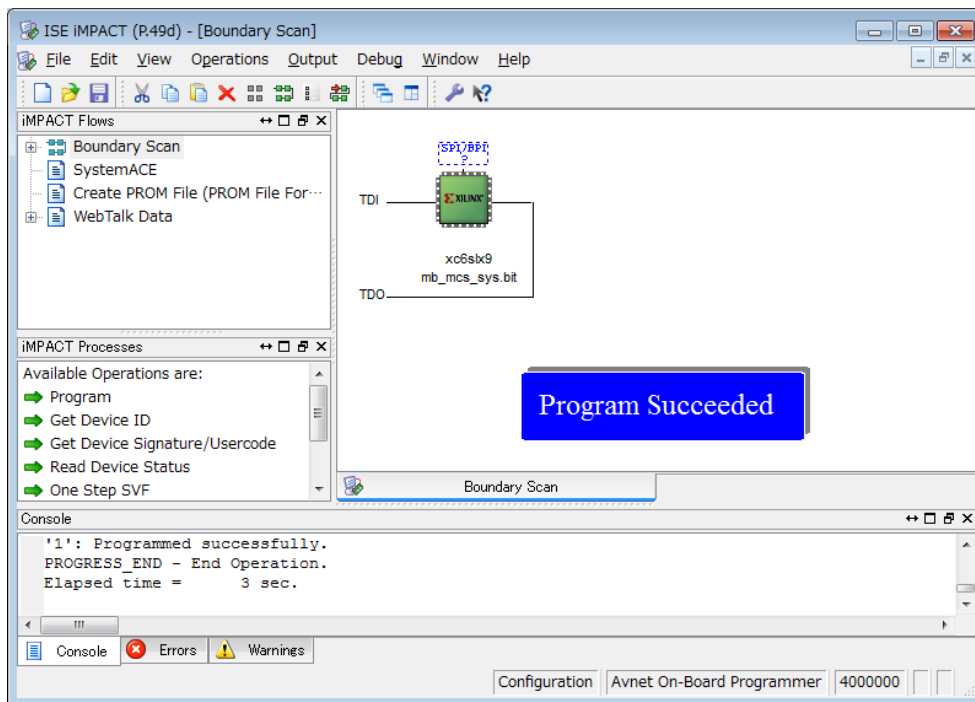
PROM データは使わないので No を選択



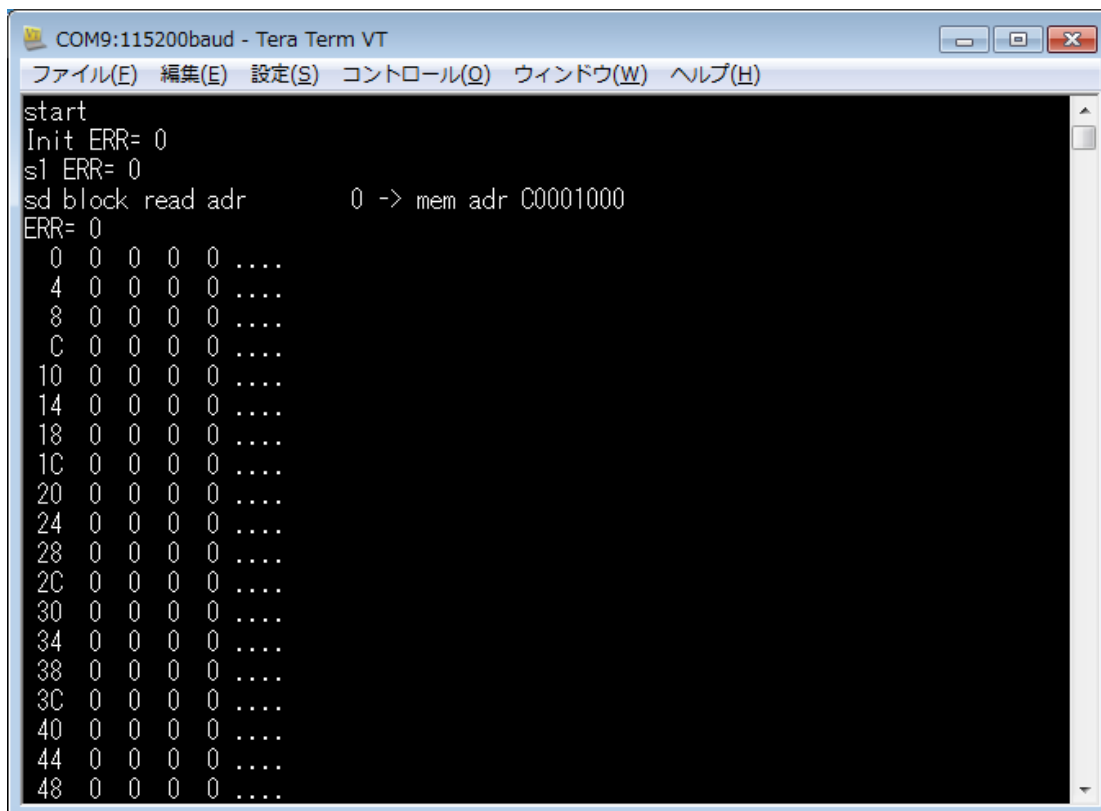


FPGA へのプログラミング実行、デバイス上でマウス右ボタンを押して Program 選択





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。



ターミナルソフトに表示された実行ログ、FPGA マガジン NO. 4 の内容と同じであれば動作 OK です。