

FPGA マガジン NO.3

MicroBlazeMCS システム開発編用設計データの使い方 (2013/10/25)

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
- ・ svga_ctrl.v グラフィック処理
- ・ xil_dvi.v DVI 信号生成
- ・ dvi_data_enc.v DVI 用エンコーダ
- ・ test_mcs_sys.v テストベンチ
- ・ mb_mcs_sys.ucf ピン配置指定
- ・ svga_sys_sw1.c サンプルプログラム

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

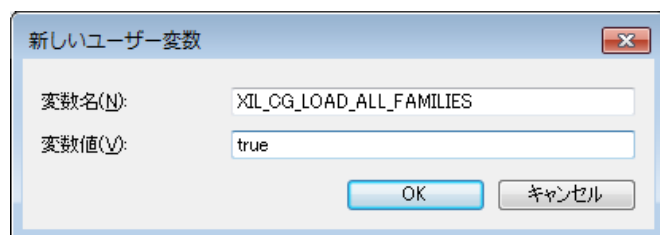
手順

- ・ 設計用フォルダ svga_sys/mb_mcs_sys を作成する
- ・ 記事のダウンロードデータ mb_mcs_svga の下にある Verilog-HDL ソースと UCF ファイルを svga_sys/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

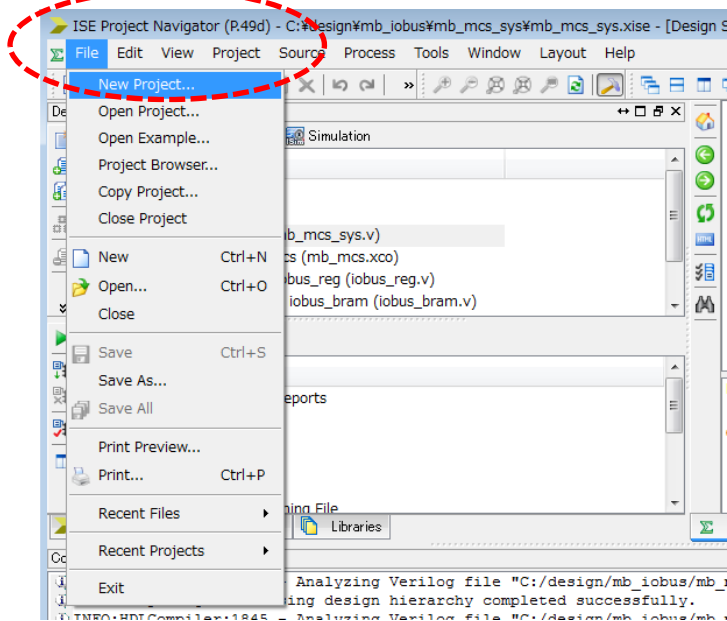


変数名：XIL_CG_LOAD_ALL_FAMILIES 値：true を入力して OK

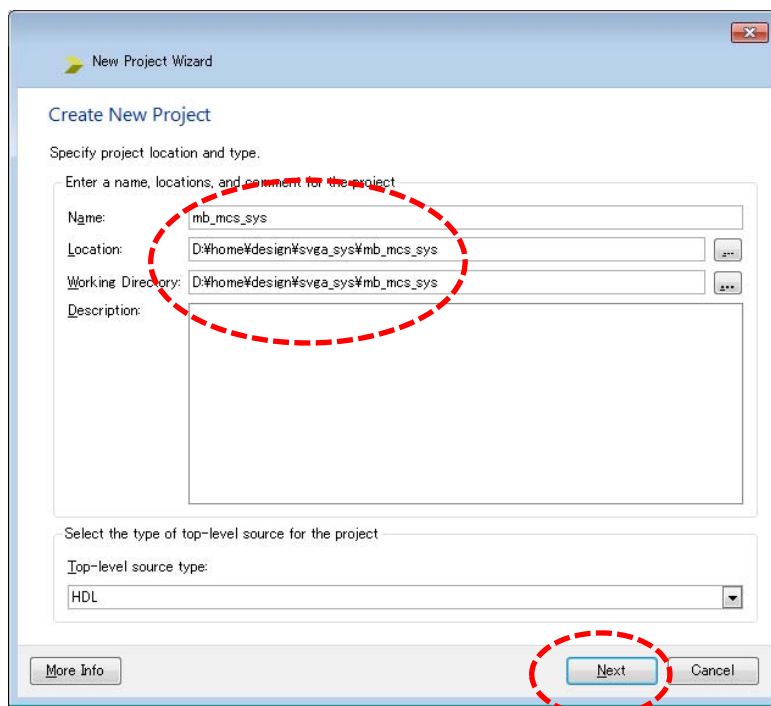
- ・ ProjectNavigator を起動します。

スタートメニューから「Xilinx Design Tools」→「ISE Design Suite 14.6」→「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



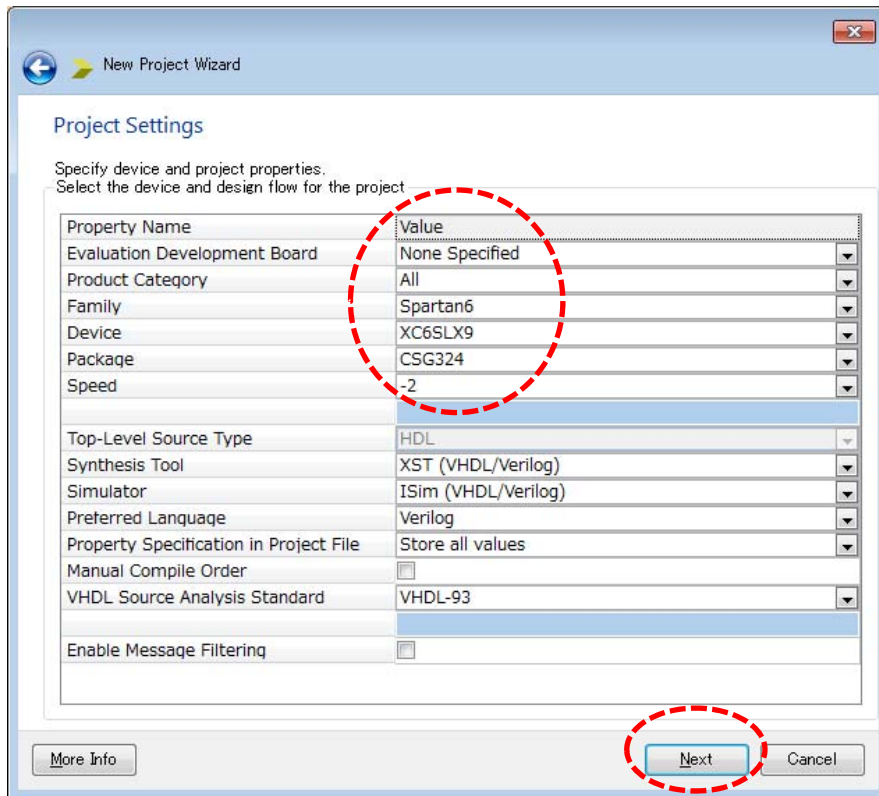
プロジェクト名と設計フォルダ指定

プロジェクト名 : mb_mcs_sys

設計フォルダ : 任意/svga_sys/mb_mcs_sys

次に Next をクリック



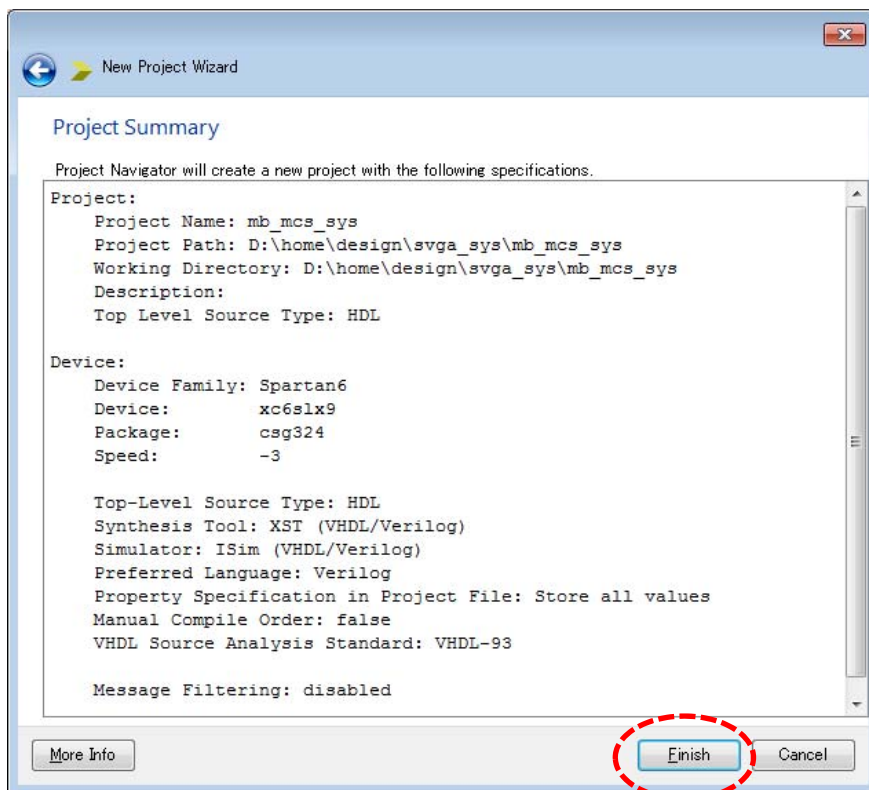


The 'Project Settings' dialog box in the New Project Wizard. It contains a table of properties and their values. A red dashed circle highlights the 'Device' section, which includes 'Family' (Spartan6), 'Device' (XC6SLX9), and 'Package' (CSG324). Another red dashed circle highlights the 'Next' button 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 'Project Summary' dialog box in the New Project Wizard. It displays the project specifications in a text area. 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\svga_sys\mb_mcs_sys
- Working Directory: D:\home\design\svga_sys\mb_mcs_sys
- Description:
- Top Level Source Type: HDL

Device:

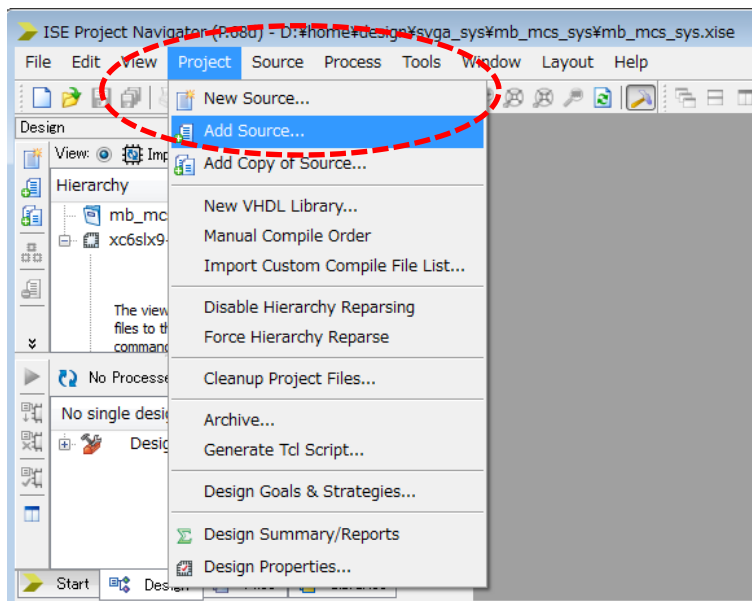
- Device Family: Spartan6
- Device: xc6slx9
- Package: csg324
- Speed: -3

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

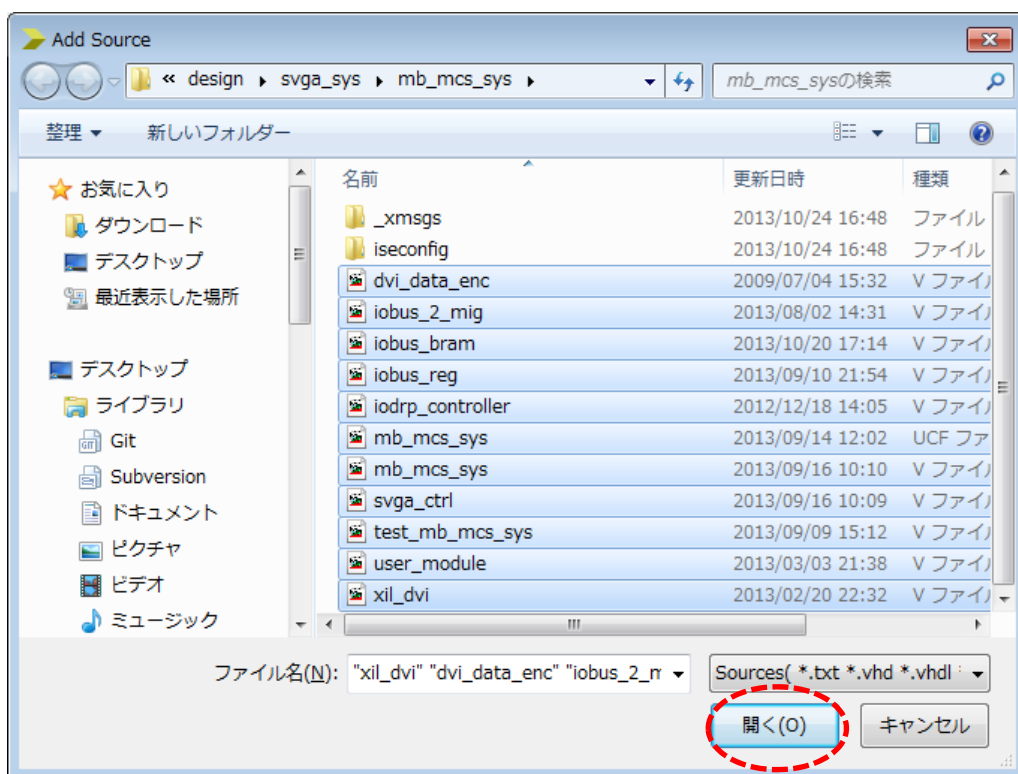
Buttons: More Info, Finish, Cancel

Finish をクリック



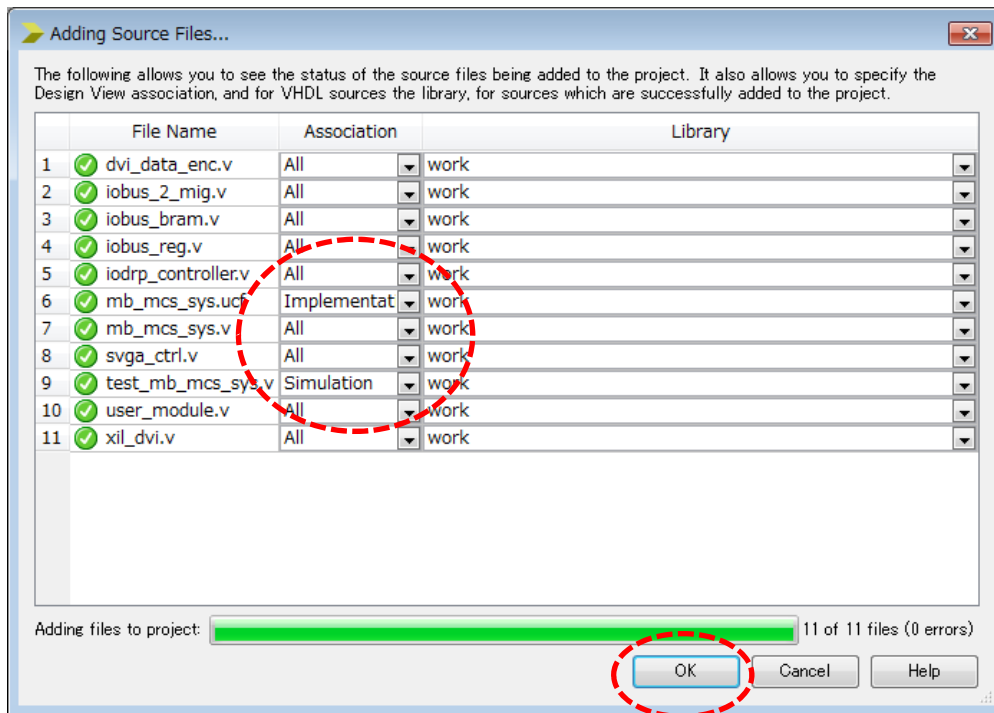


設計データを追加します。Project→Add Source を選択

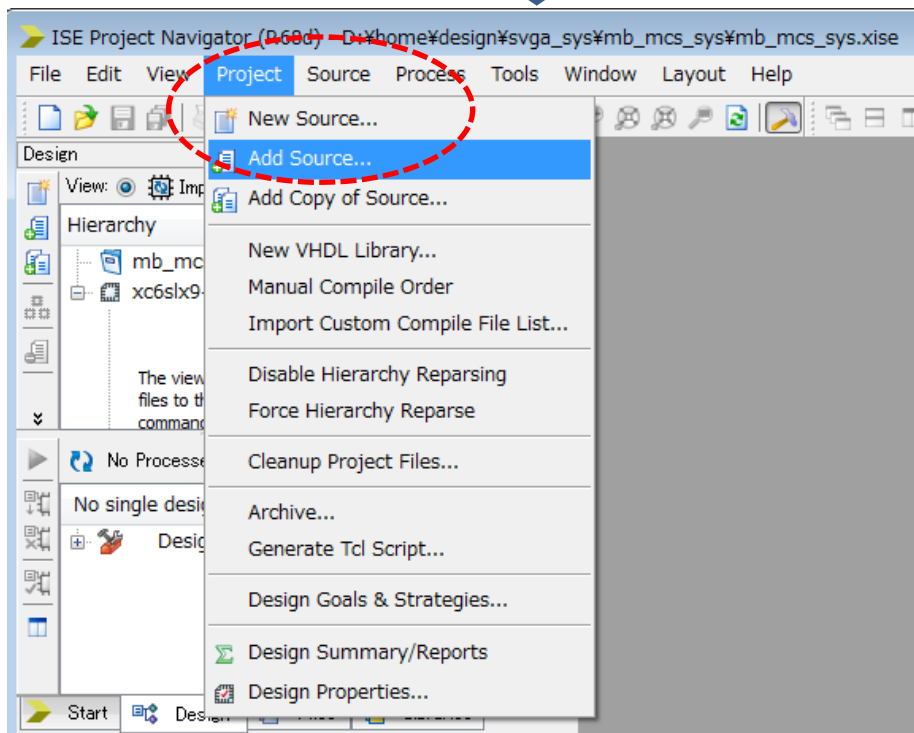


ファイルの指定、mb_mcs_sys の下にある Verilog-HDL ファイルと UCF ファイルを指定



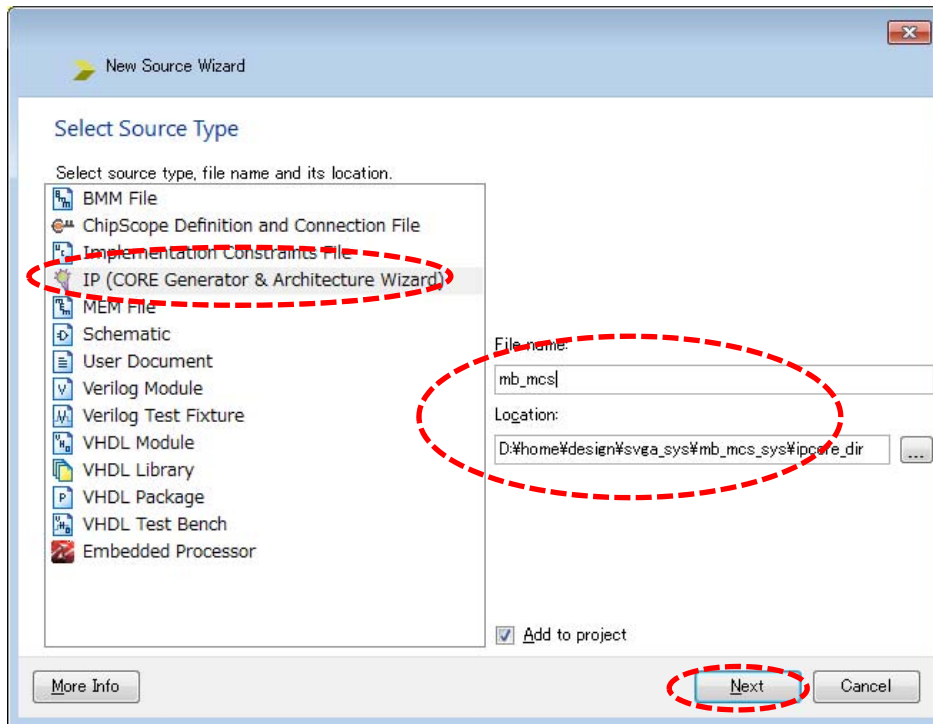


mb_mcs_sys.ucf はインプリメンテーションで使用するので Implementation を指定
test_mb_mcs_sys.v はテストベンチなので Simulation を指定、OK をクリック

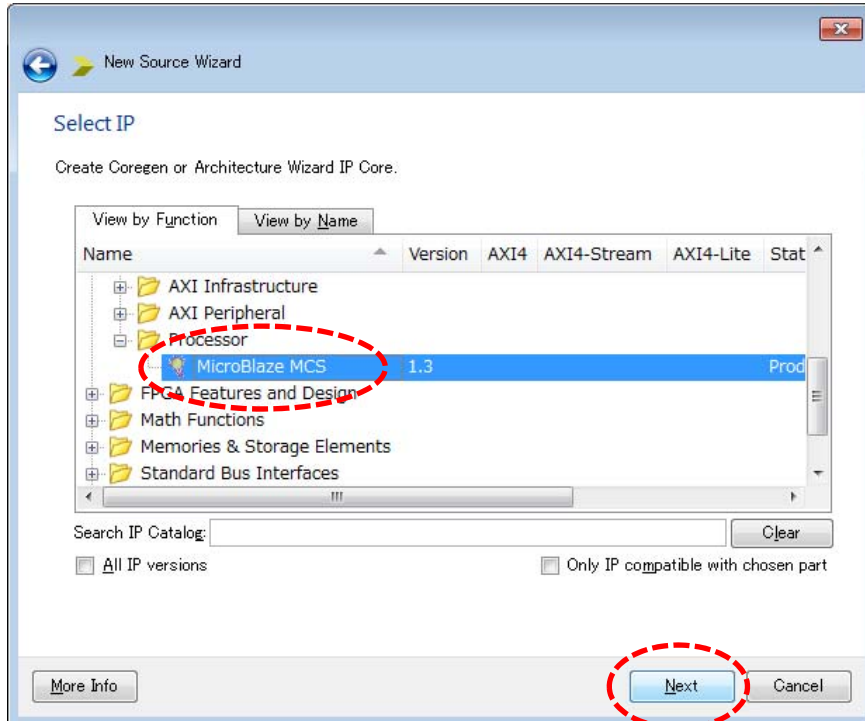


Project Navigator で MicroBlaze MCS を追加する。Project→New Source を選択



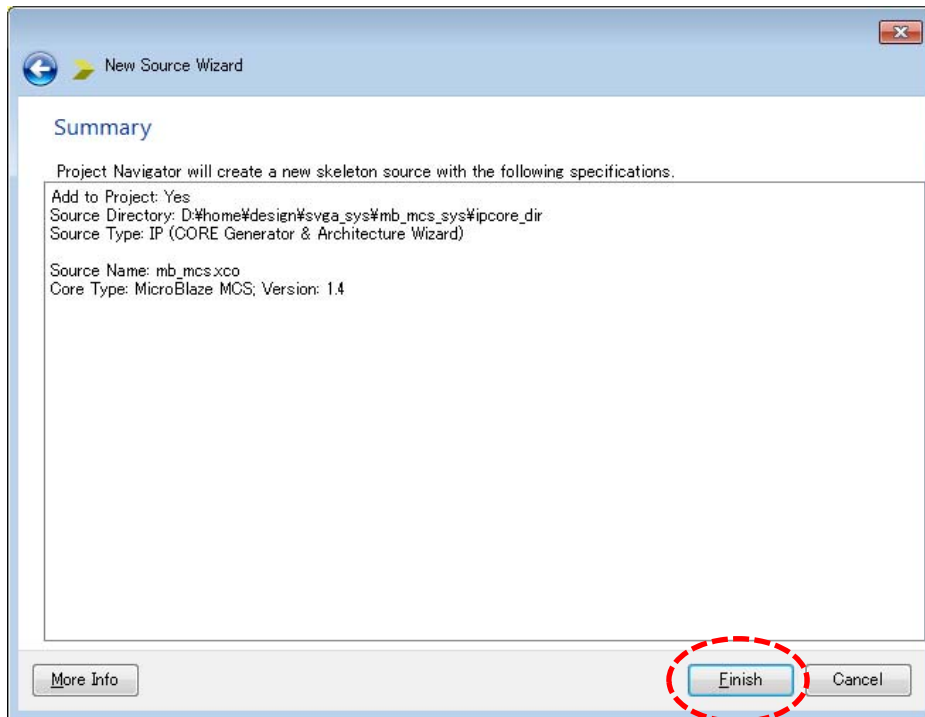


IP (CORE Gener…をクリックして選択、ファイル名を mb_mcs に指定、Next をクリック

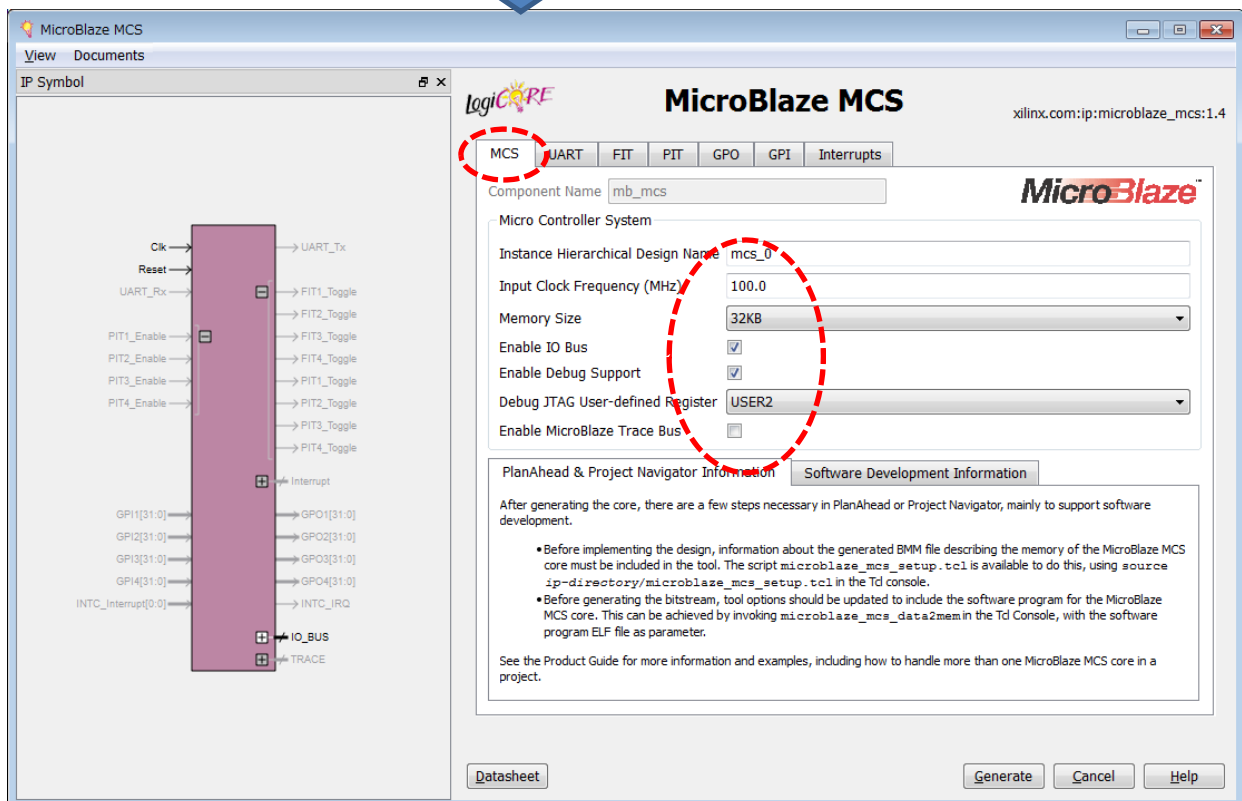


MicroBlaze MCS 選択して Next をクリック

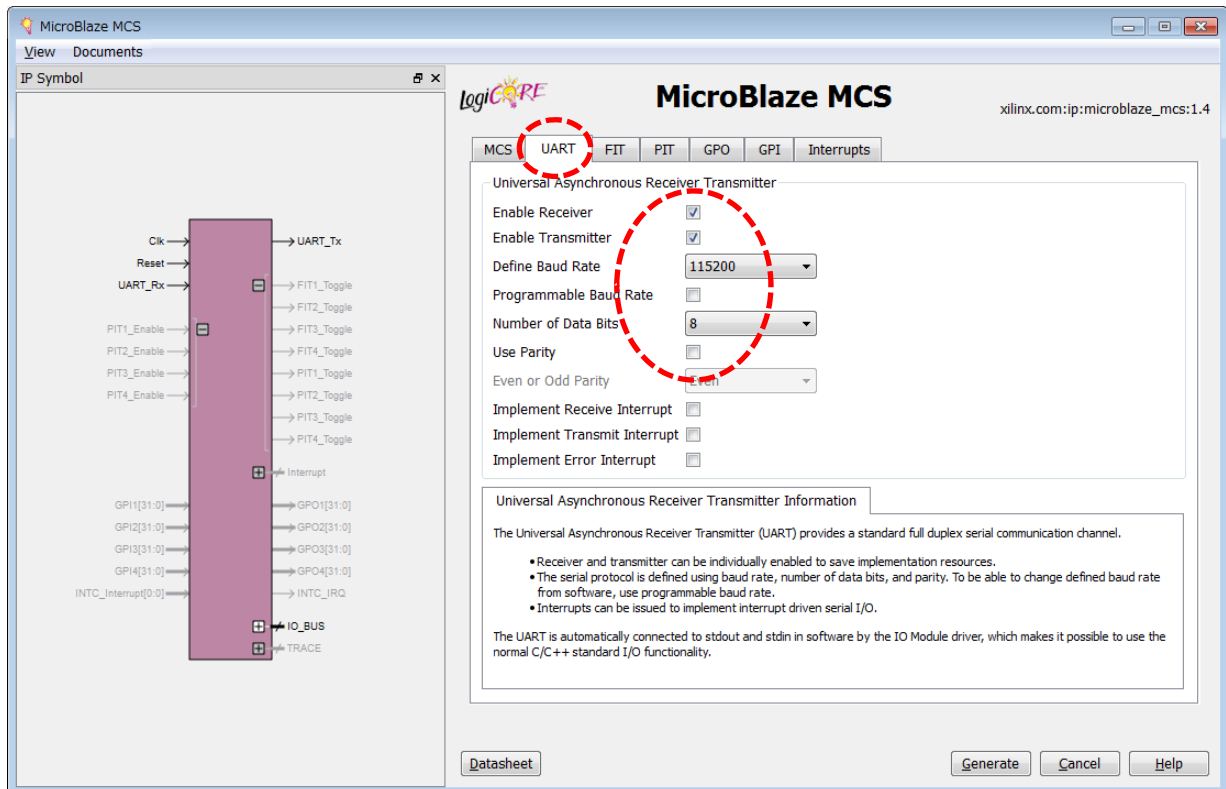




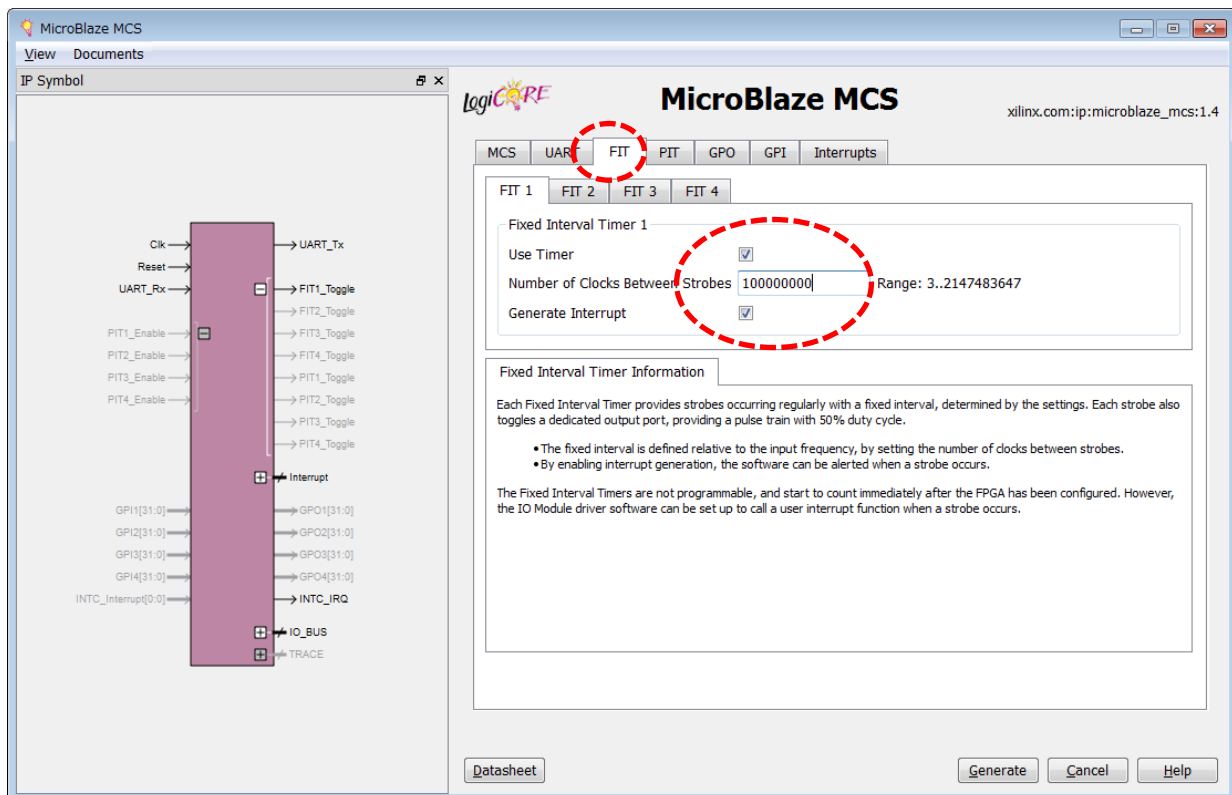
Finish をクリックで CORE generator が起動



MicroBlaze MCS の基本設定

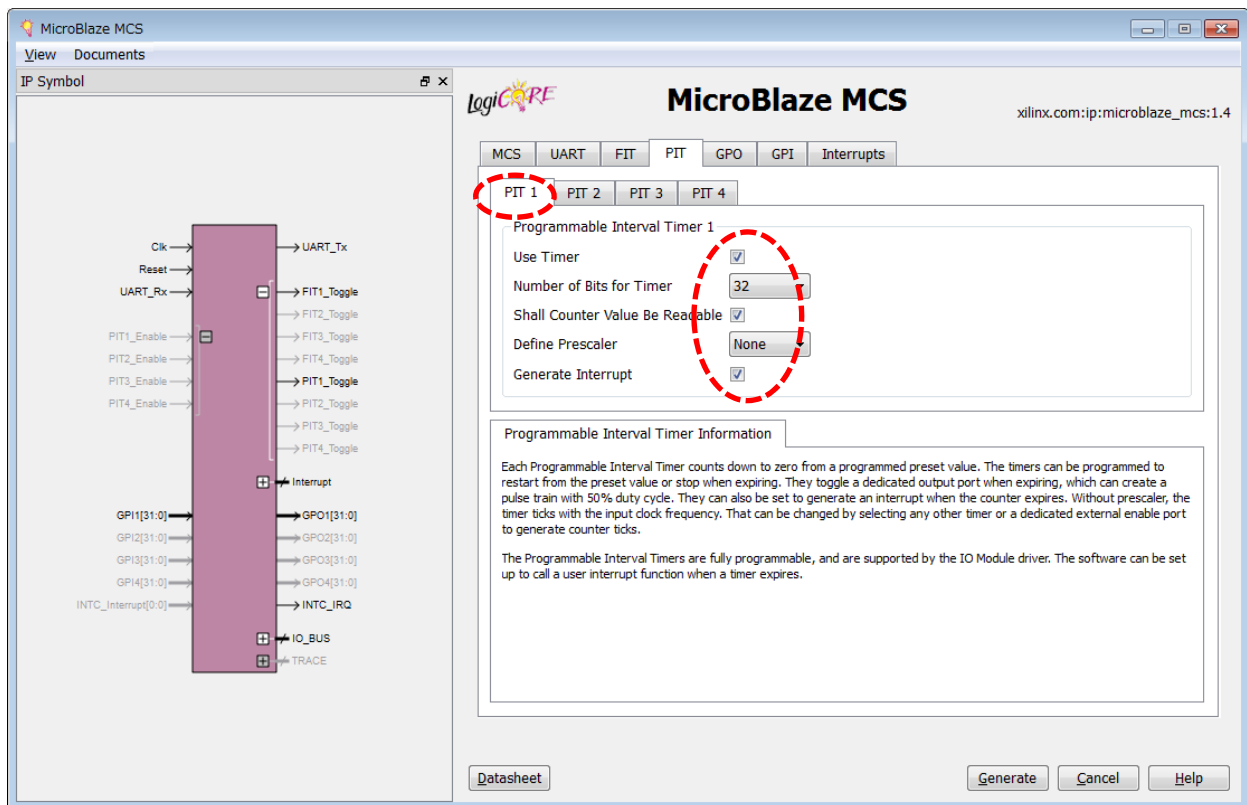


UART の設定

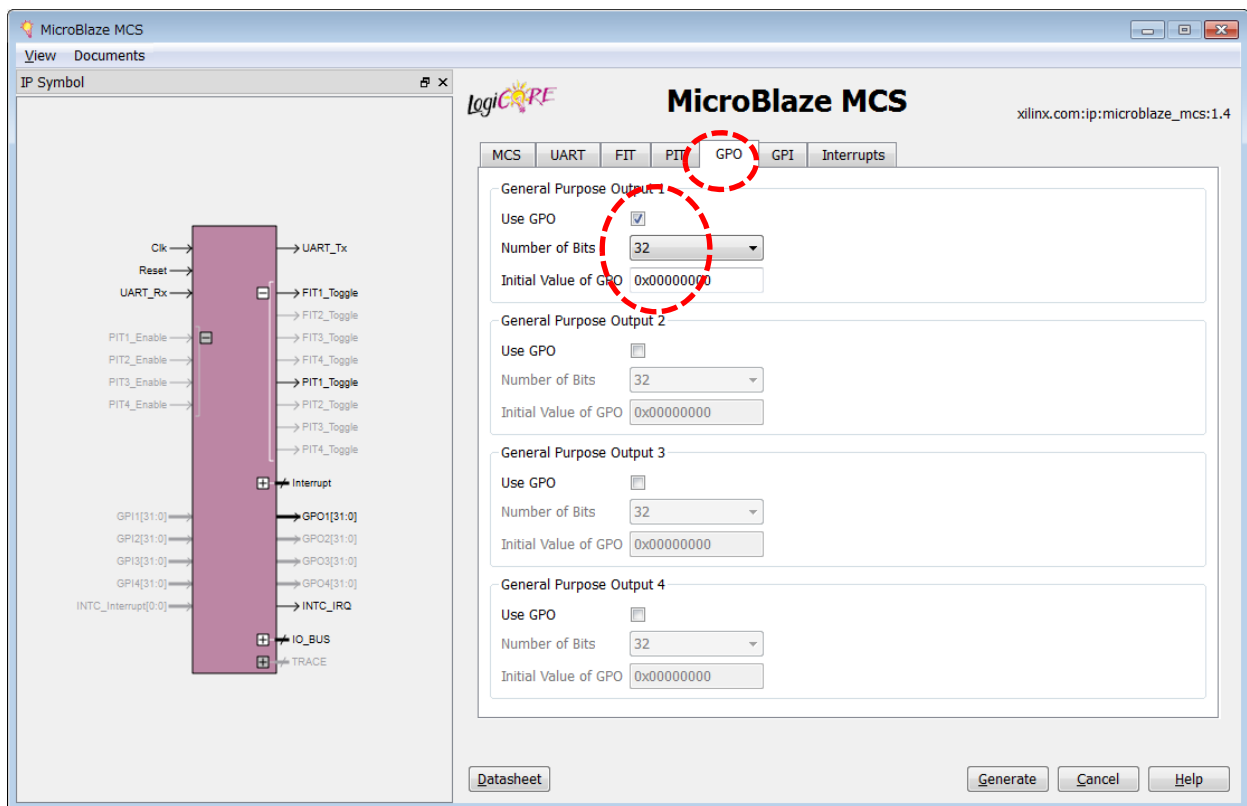


FIT の指定



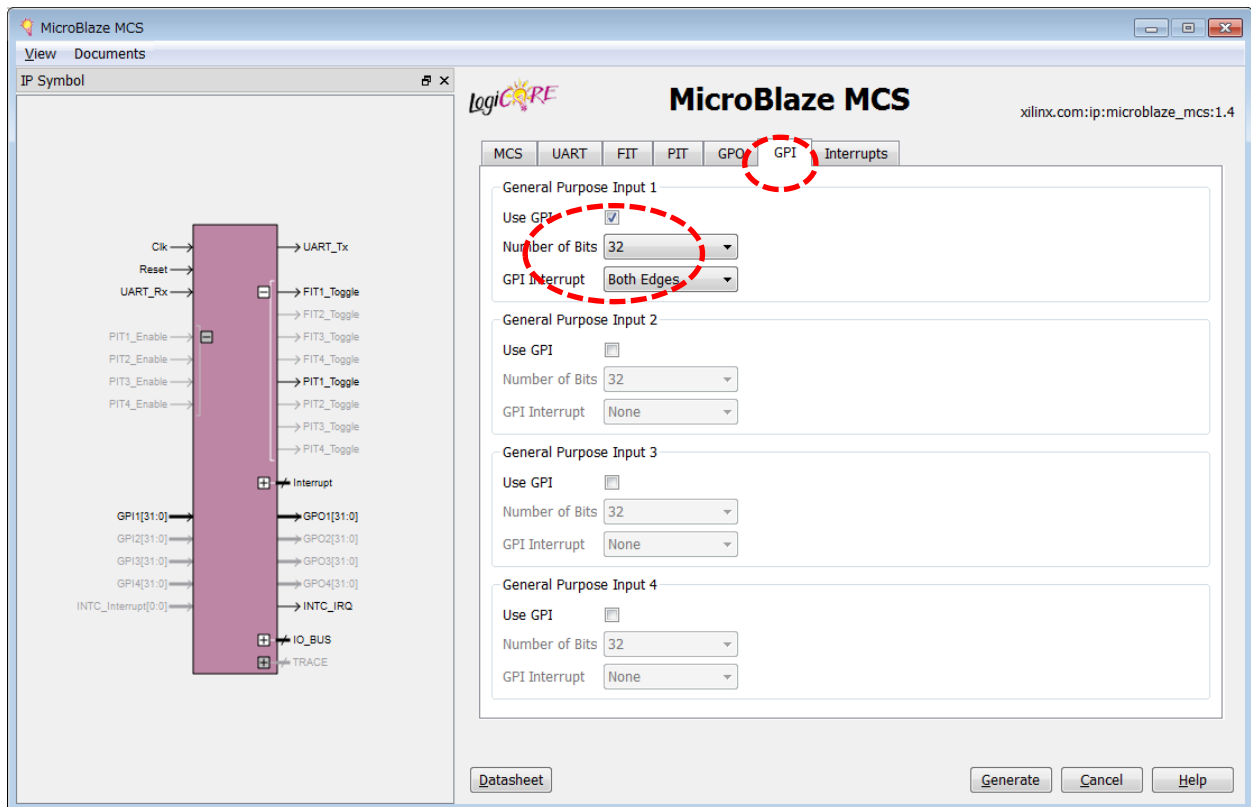


PIT の指定

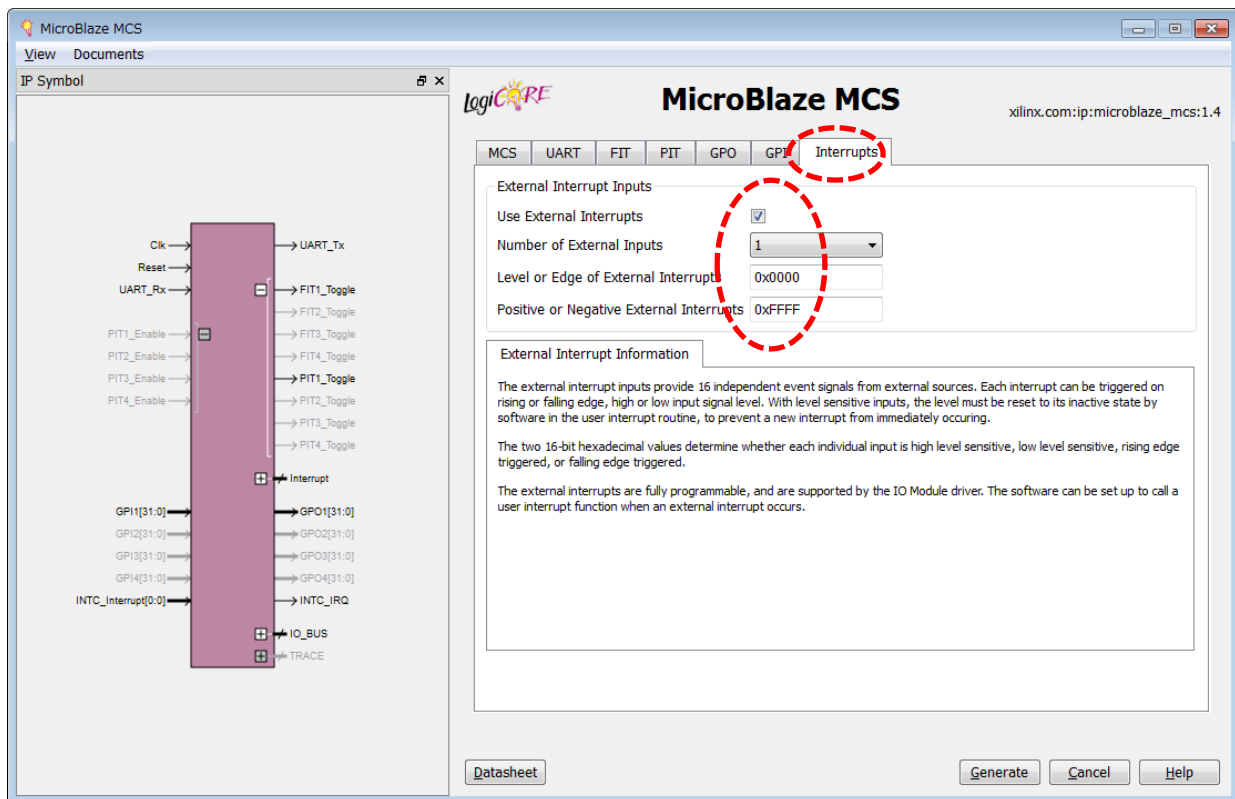


GPO の設定



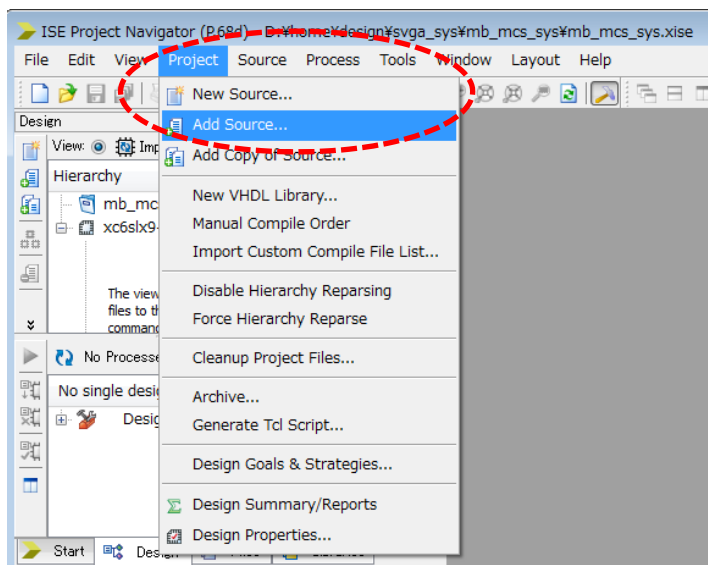


GPI の指定

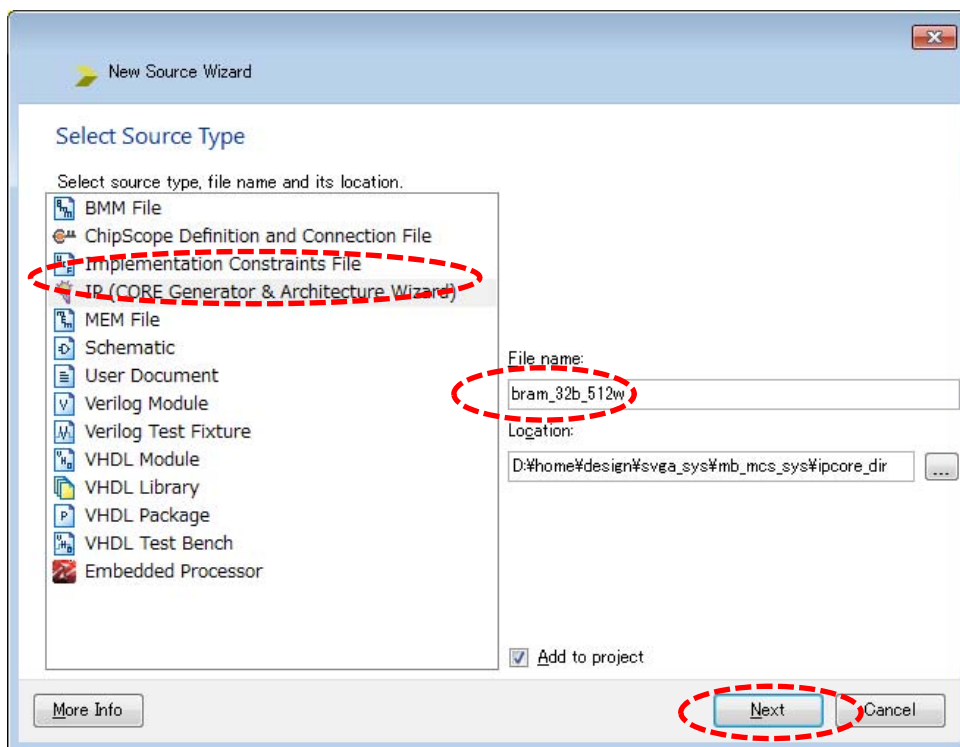


外部割り込みの設定後に Generate をクリックすると MicroBlaze MCS がプロジェクトに追加される



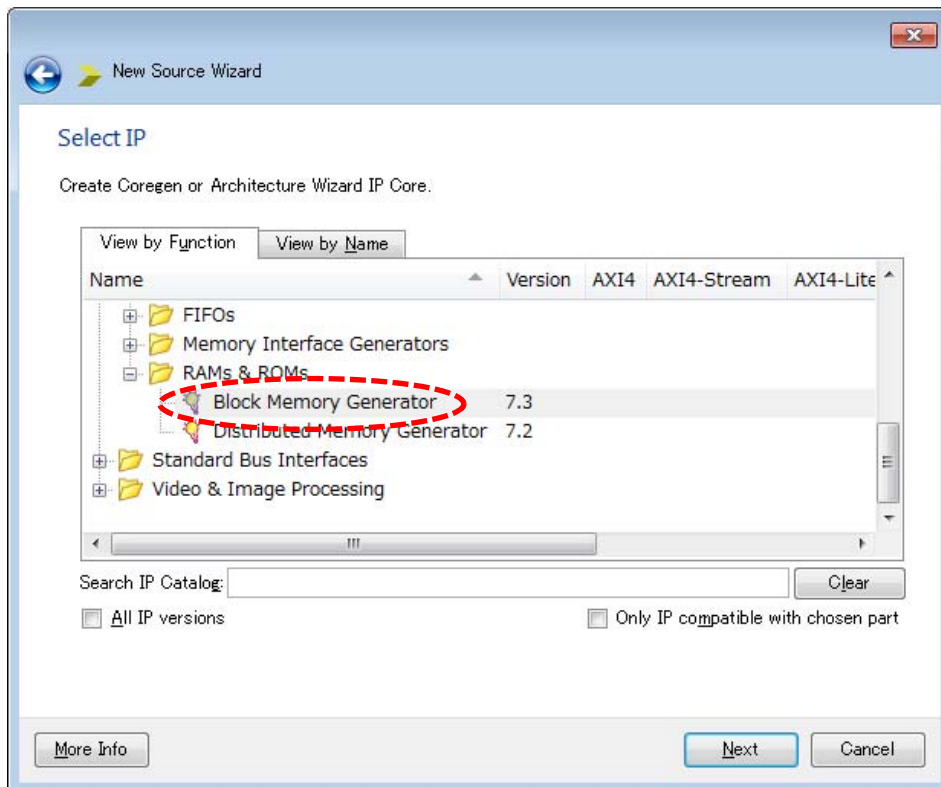


ブロック RAM の作成、Project→New Source を選択

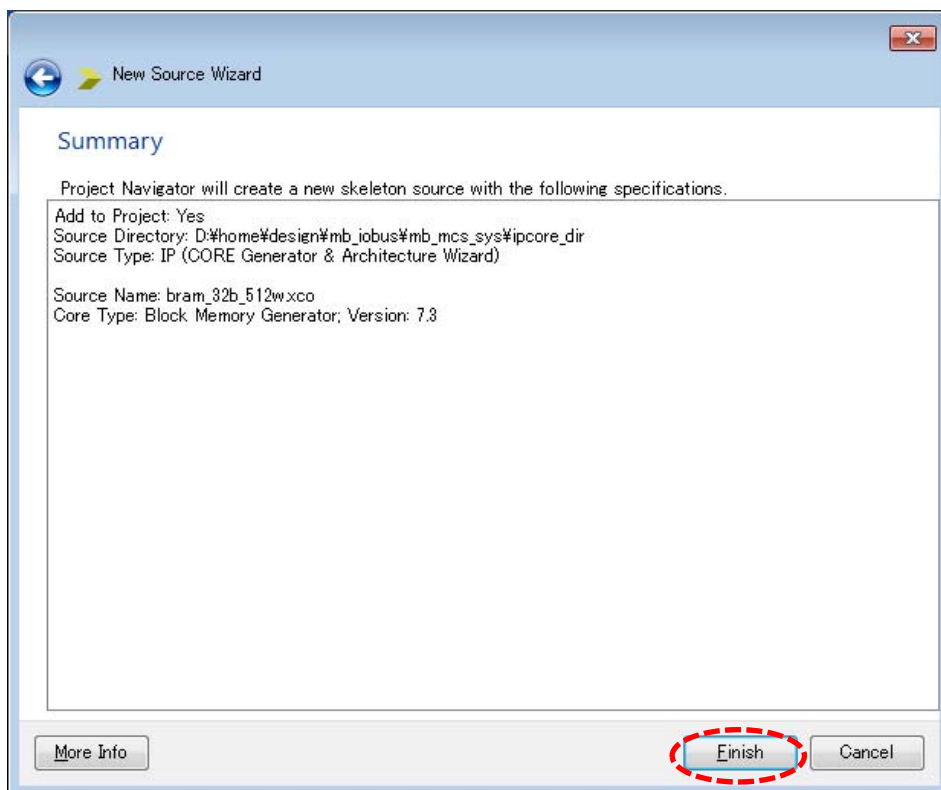


IP (CORE Gener...をクリックして選択、ファイル名に bram_32b_512w を指定、Next をクリック



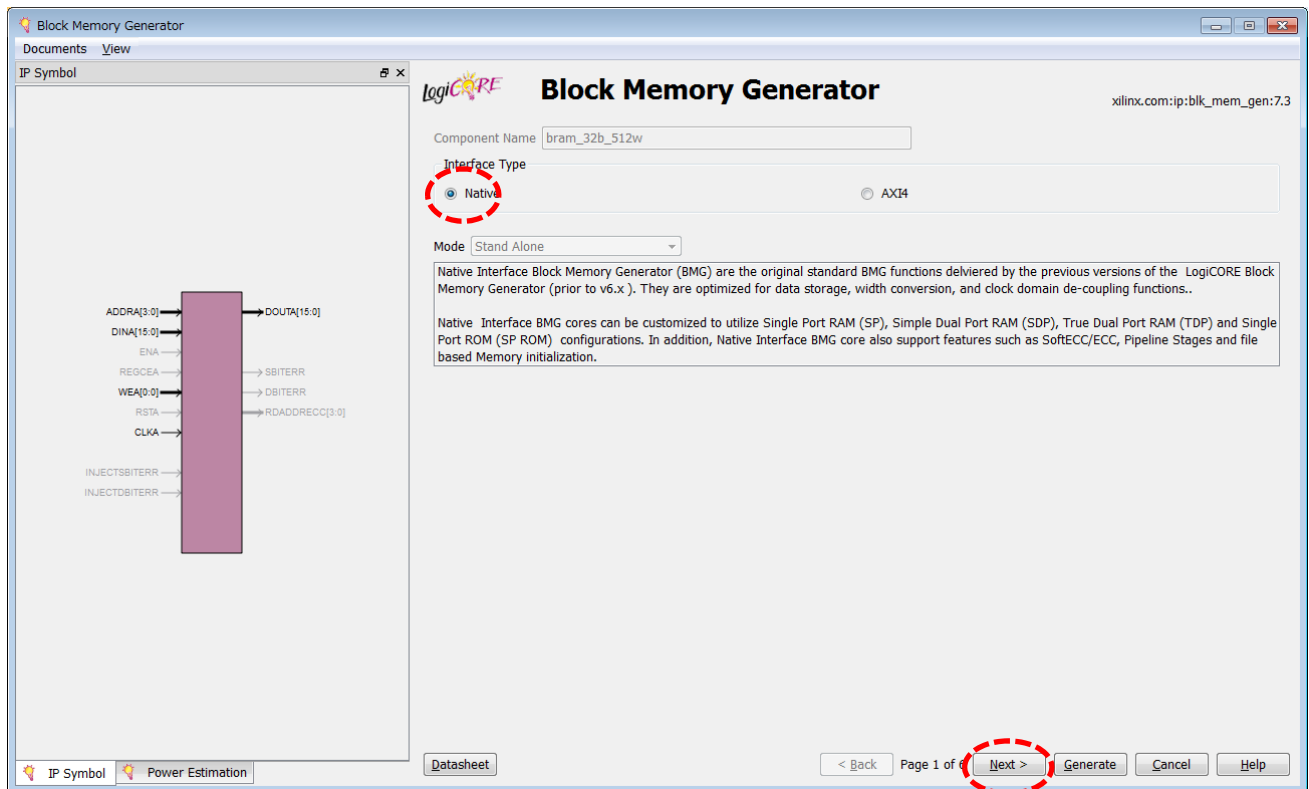


Block Memory Generator を選択して Next をクリック

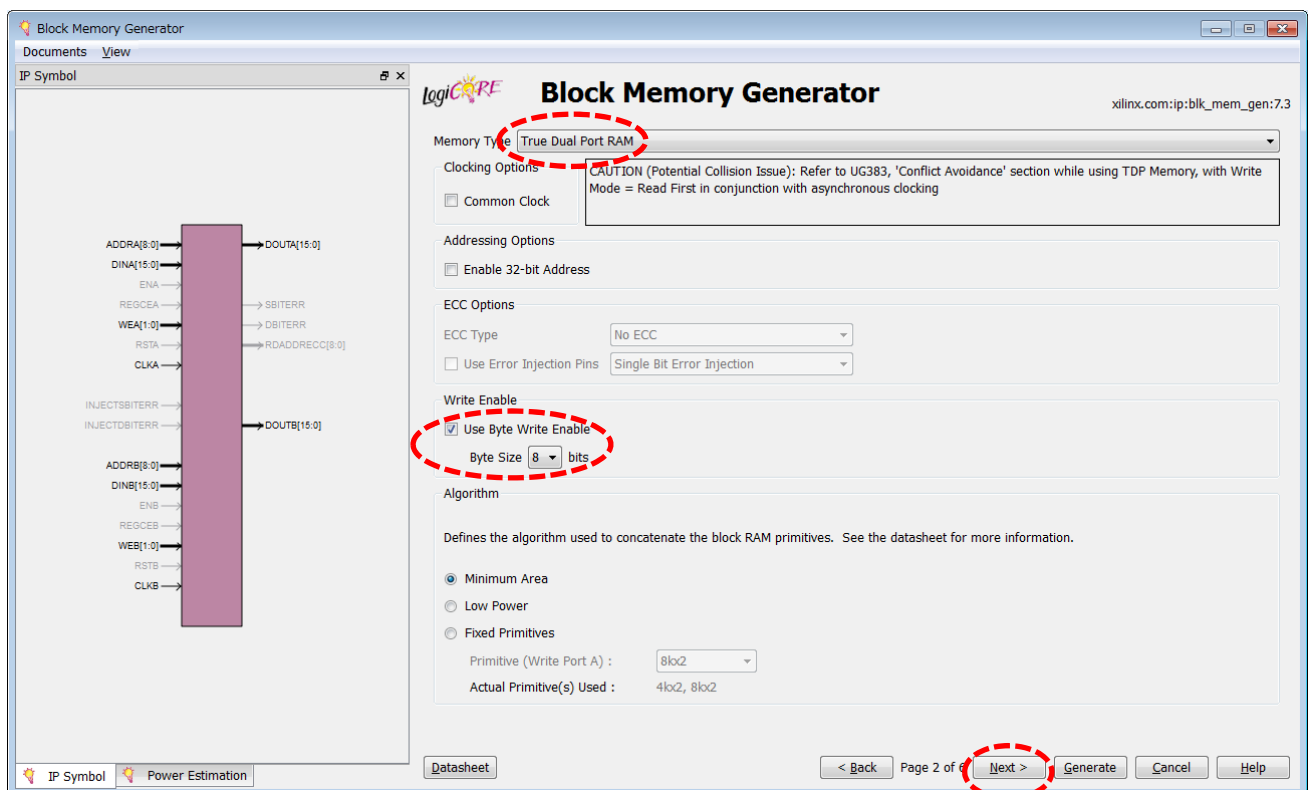


Finish をクリックすると CORE generator が起動



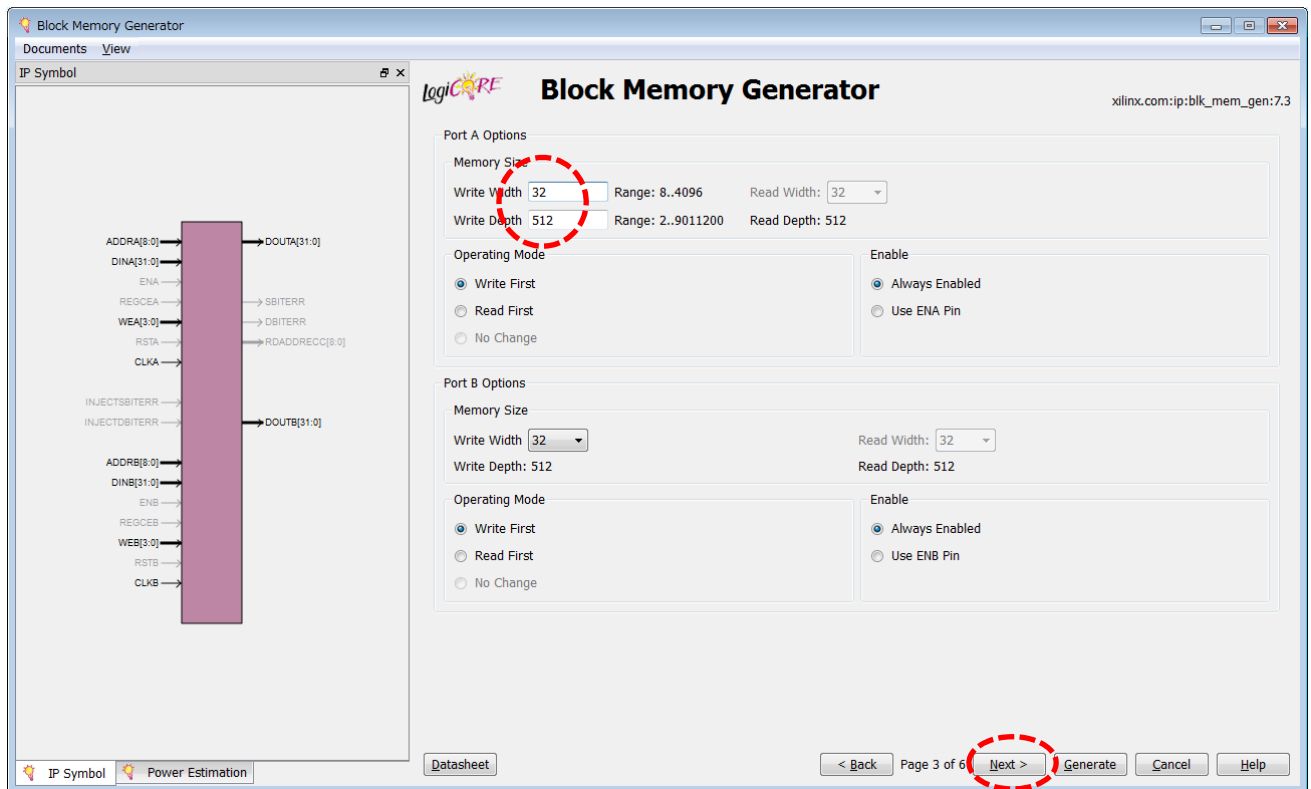


ブロック RAM インターフェースの設定

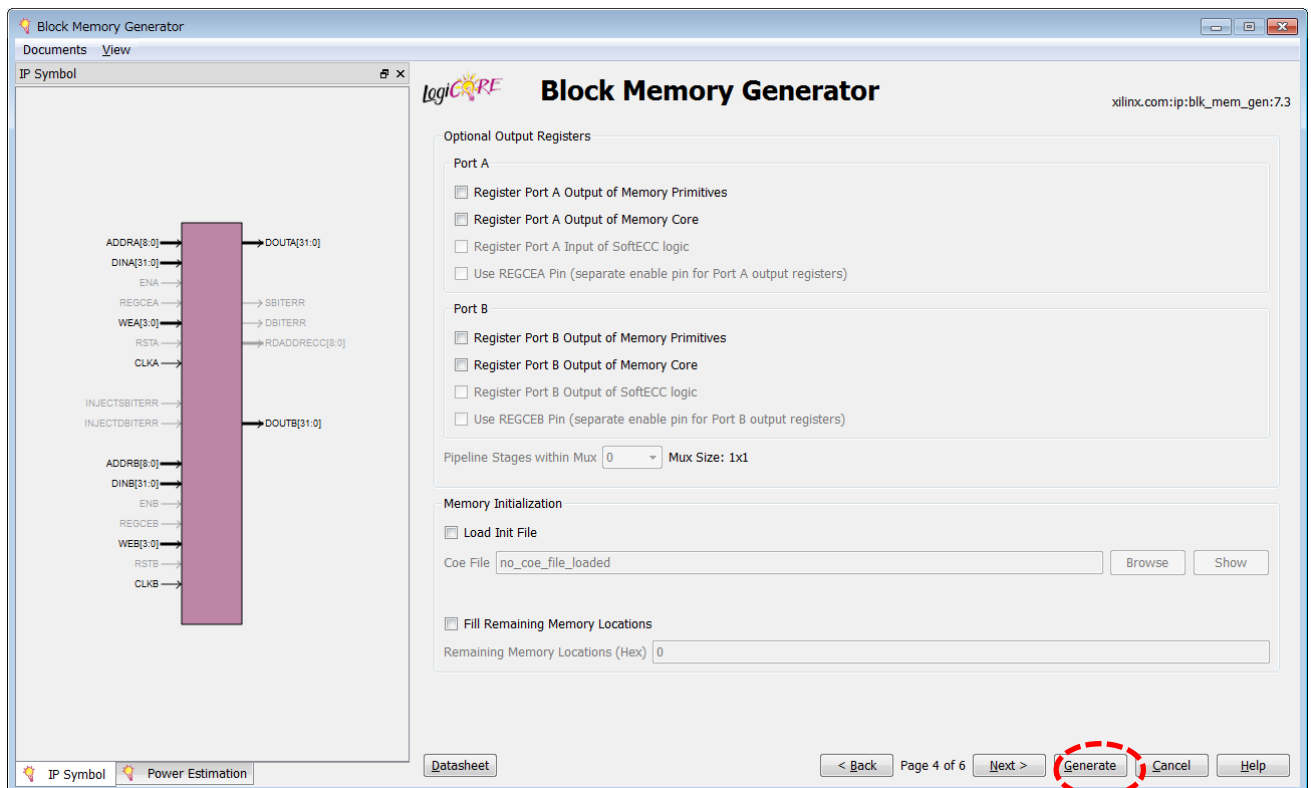


メモリのタイプ指定



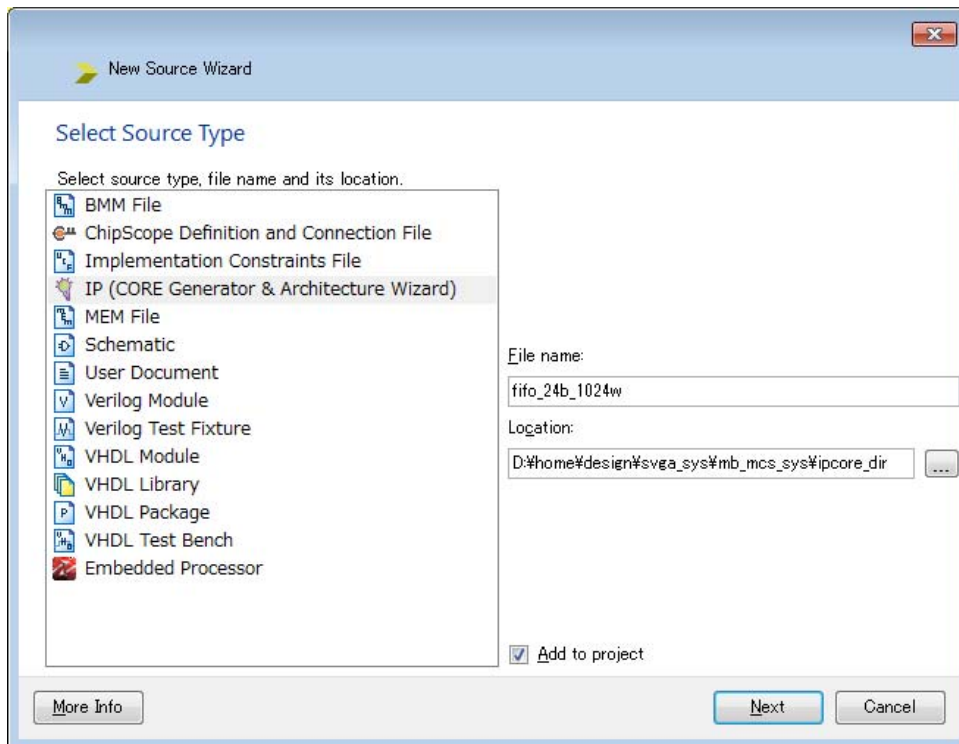


データ幅、データ量の指定



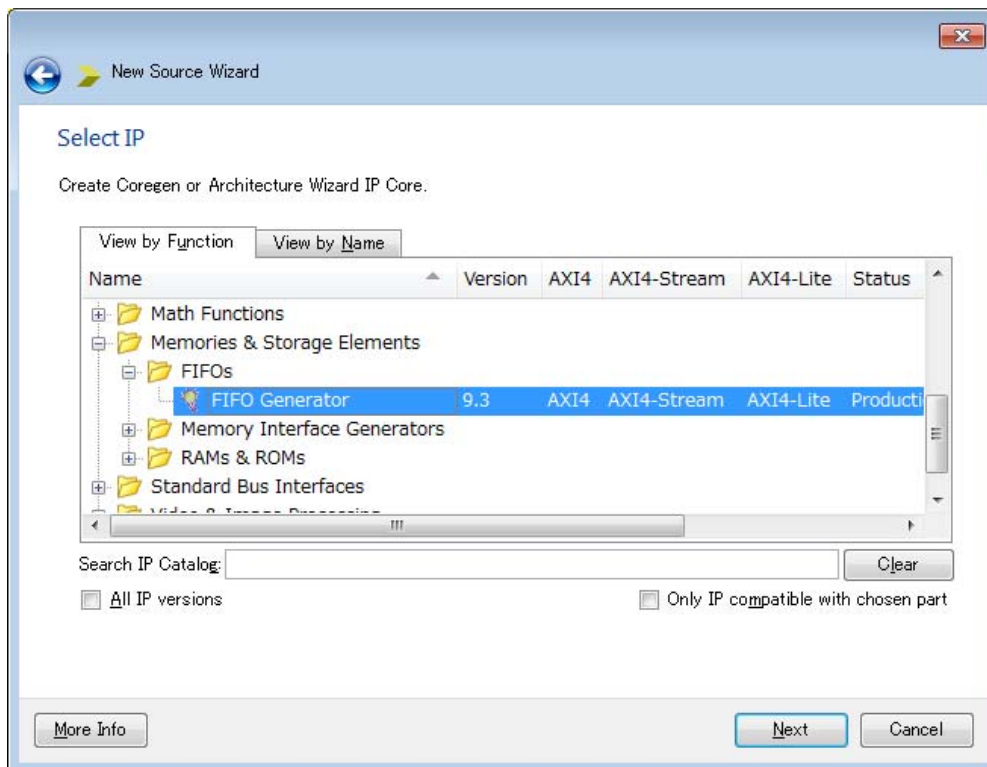
Generate をクリックで作成開始





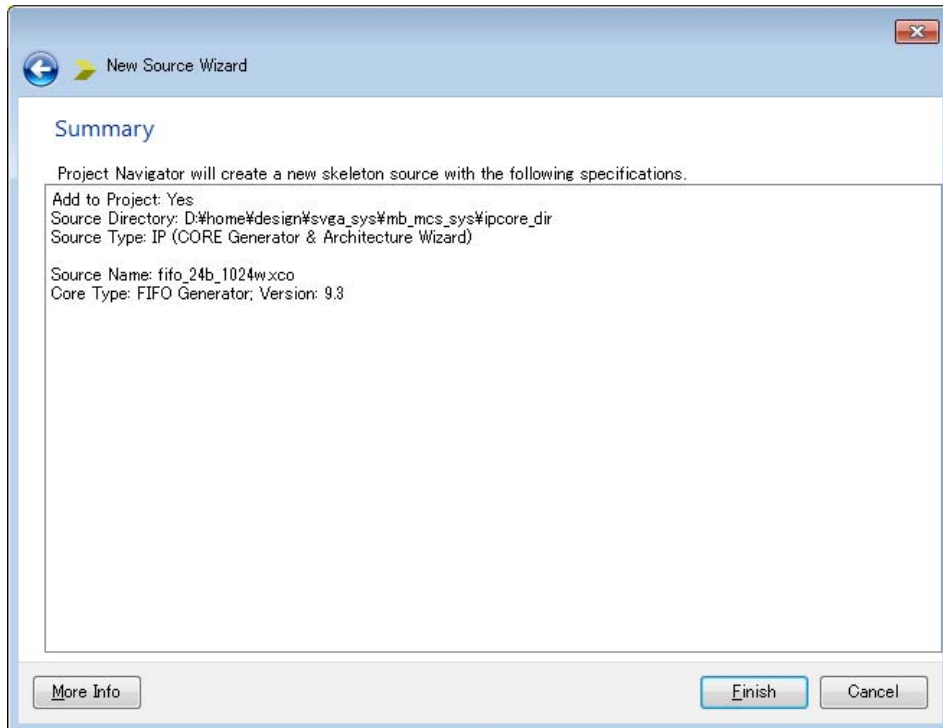
fifo_24b_1024w の作成、Project→New Source を選択

IP (CORE Gener…をクリックして選択、ファイル名に fifo_24b_1024w を指定、Next をクリック

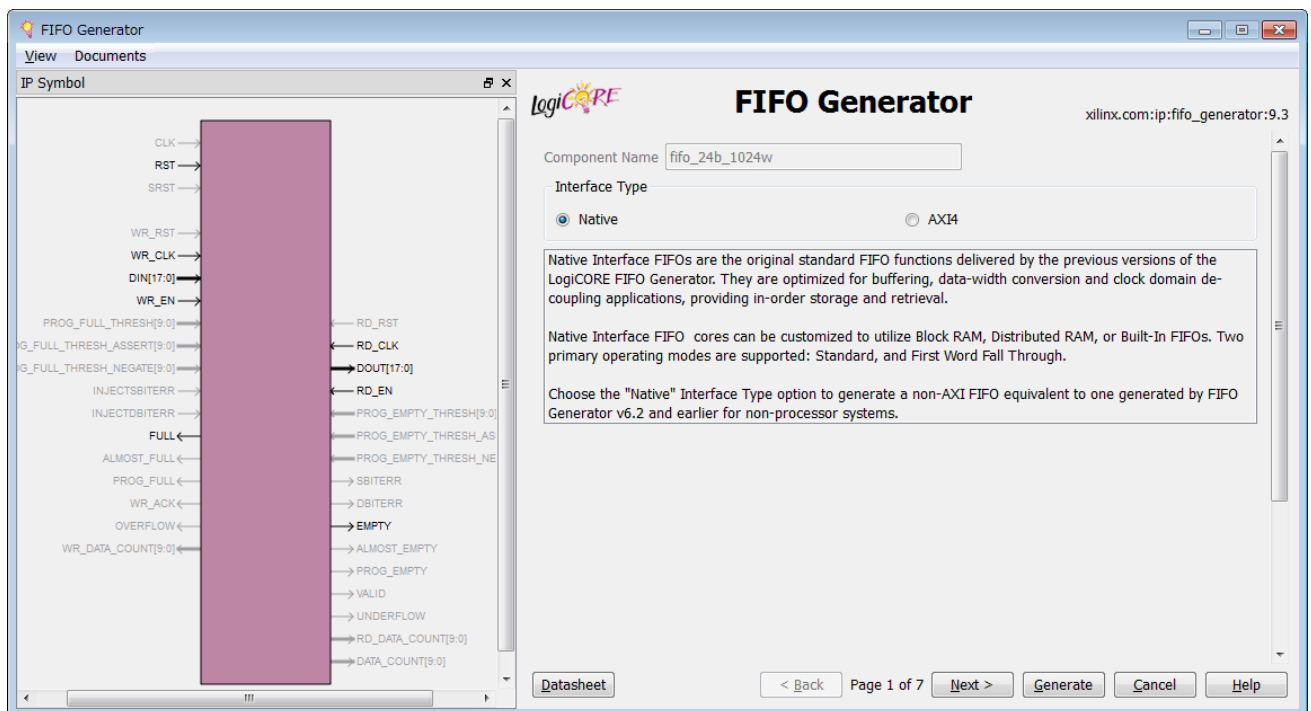


FIFO Generator 選択、Next をクリック



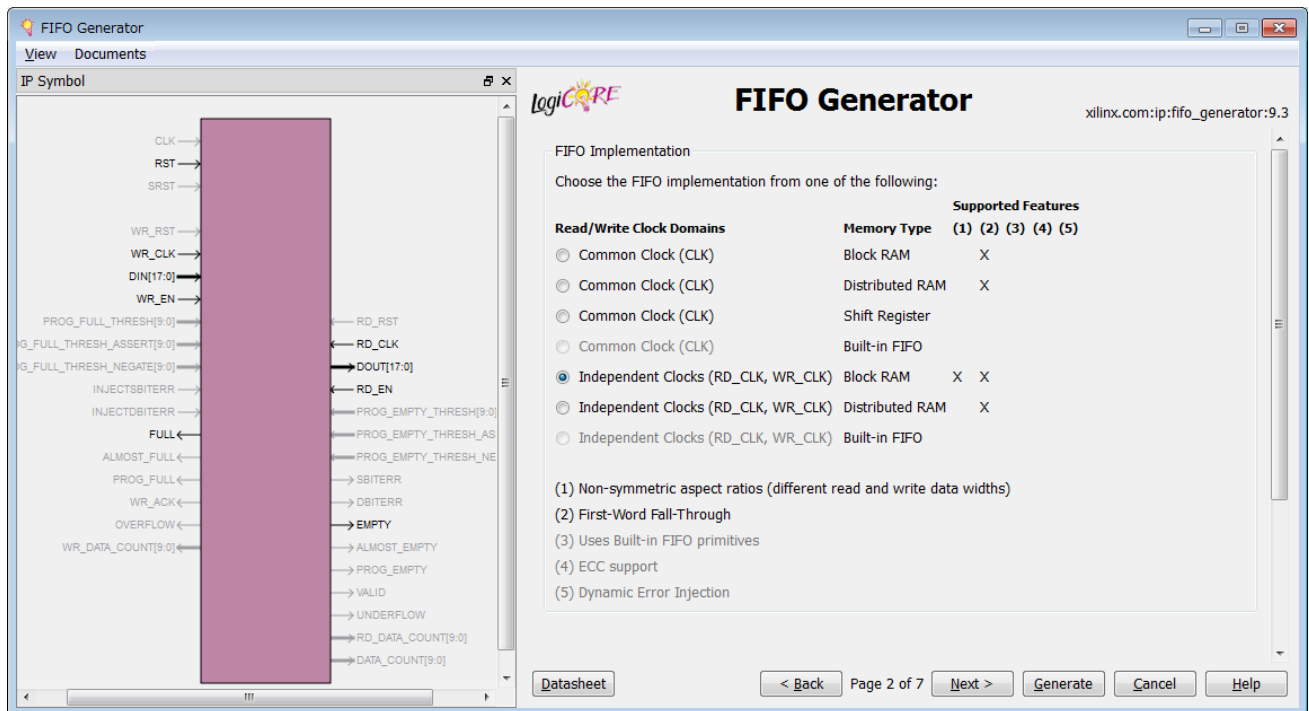


Finish をクリックすると CORE generator が起動

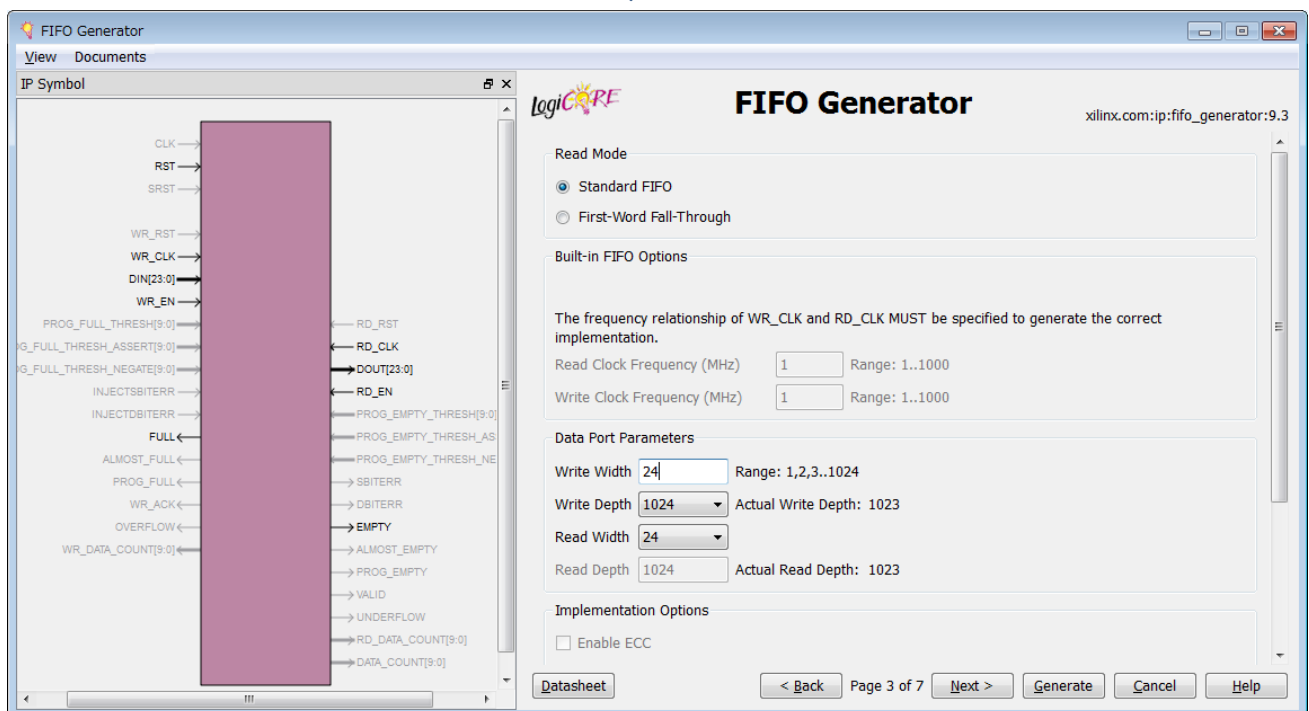


Next をクリック



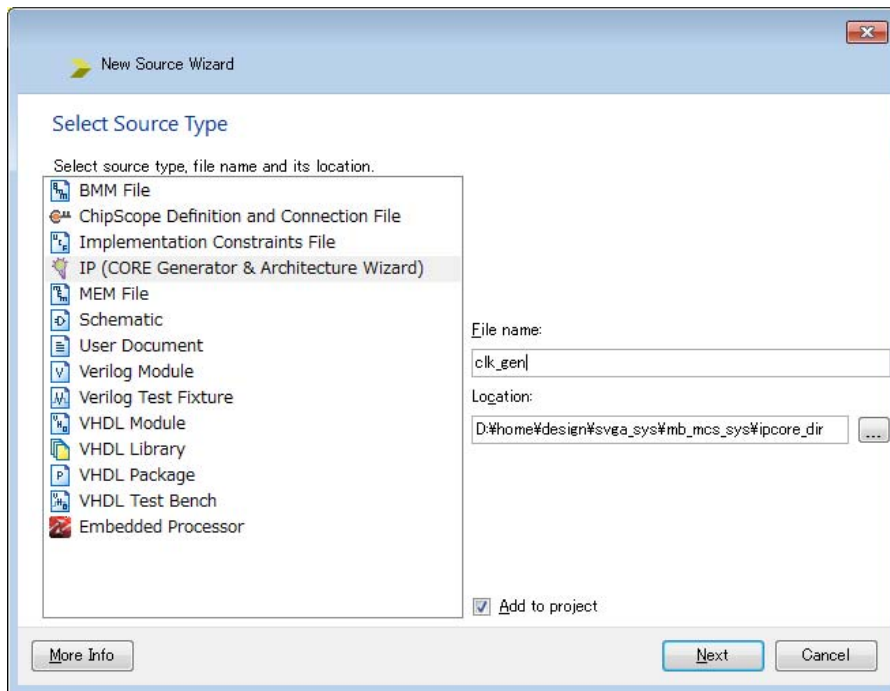


Independent Clock (RD_CLK, WR_CLK) Block RAM をチェック、Next をクリック



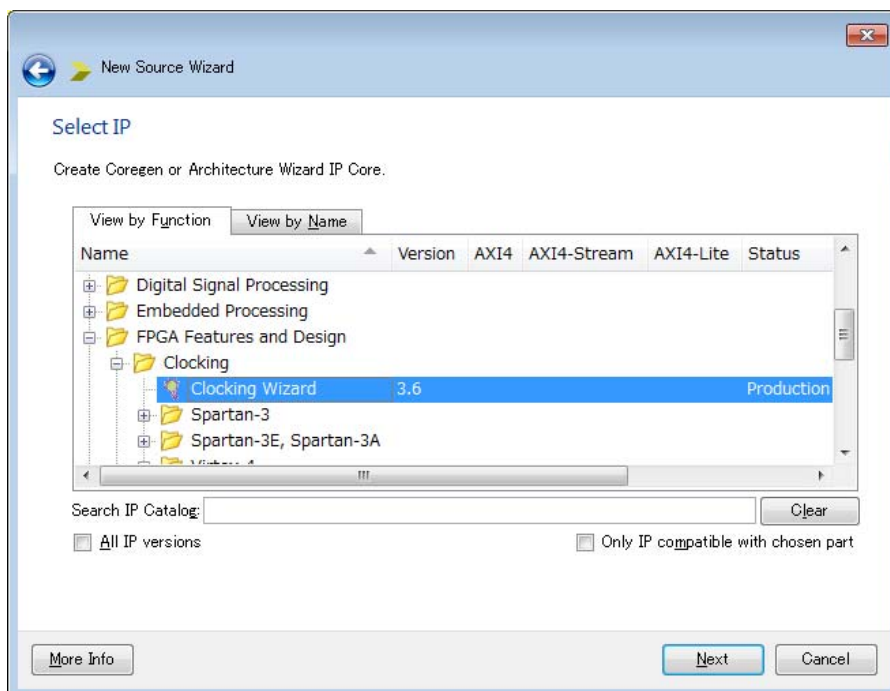
Write Width:24、Write Depth:1024 に設定、Generate をクリック





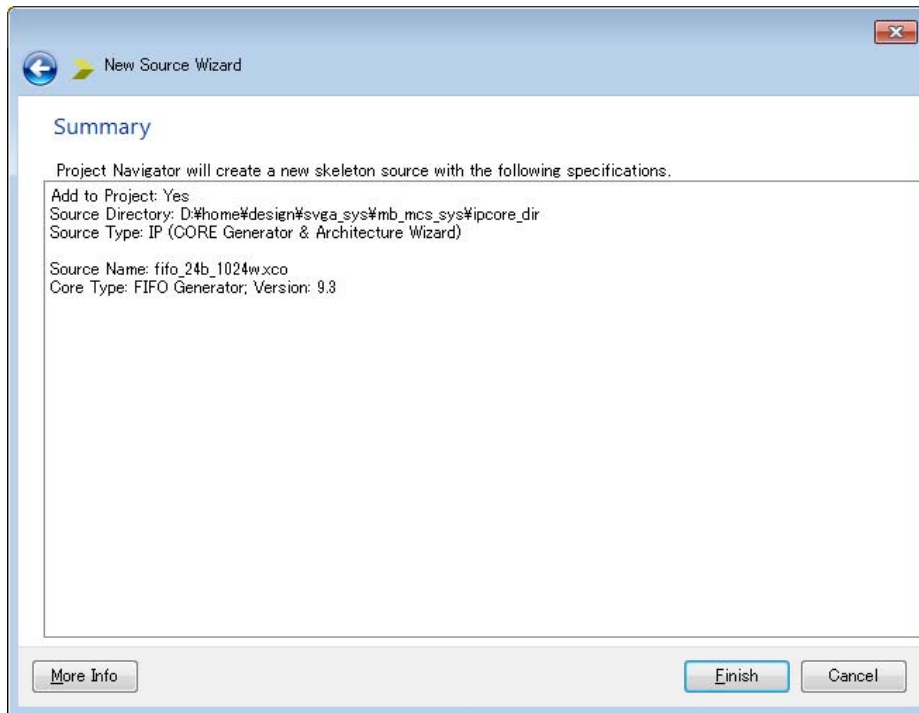
clk_gen の作成、Project→New Source を選択

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

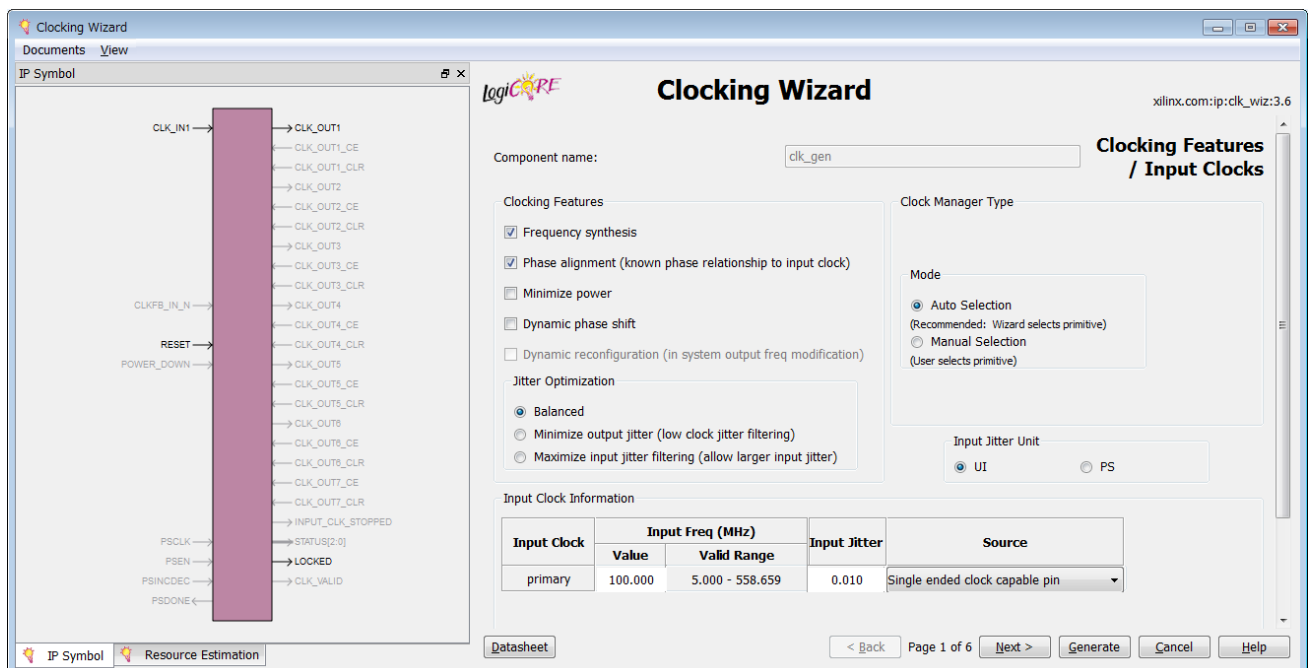


Clocking Wizard を選択、Next をクリック





Finish をクリックすると CORE generator が起動



入力クロック周波数を 100MHz に設定、Next をクリック



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		
CLK_OUT1	100.000	100.000	0.000	0.000	50.000	50.0	BUFG	<input type="checkbox"/>
<input checked="" type="checkbox"/> CLK_OUT2	40.000	40.000	0.000	0.000	50.000	50.0	BUFG	<input type="checkbox"/>
<input checked="" type="checkbox"/> CLK_OUT3	200.000	200.000	0.000	0.000	50.000	50.0	BUFG	<input type="checkbox"/>
<input checked="" type="checkbox"/> CLK_OUT4	80.000	80.000	0.000	0.000	50.000	50.0	BUFG	<input type="checkbox"/>
<input checked="" type="checkbox"/> CLK_OUT5	200.000	200.000	0.000	0.000	50.000	50.0	No buffer	<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

CLK_OUT1:100MHz

CLK_OUT2:40MHz

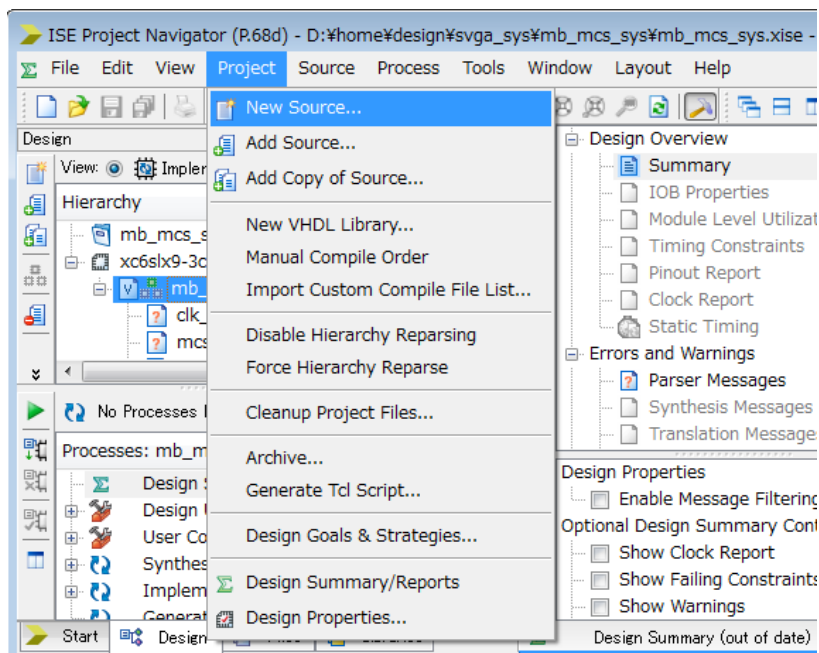
CLK_OUT3:200MHz

CLK_OUT4:80MHz

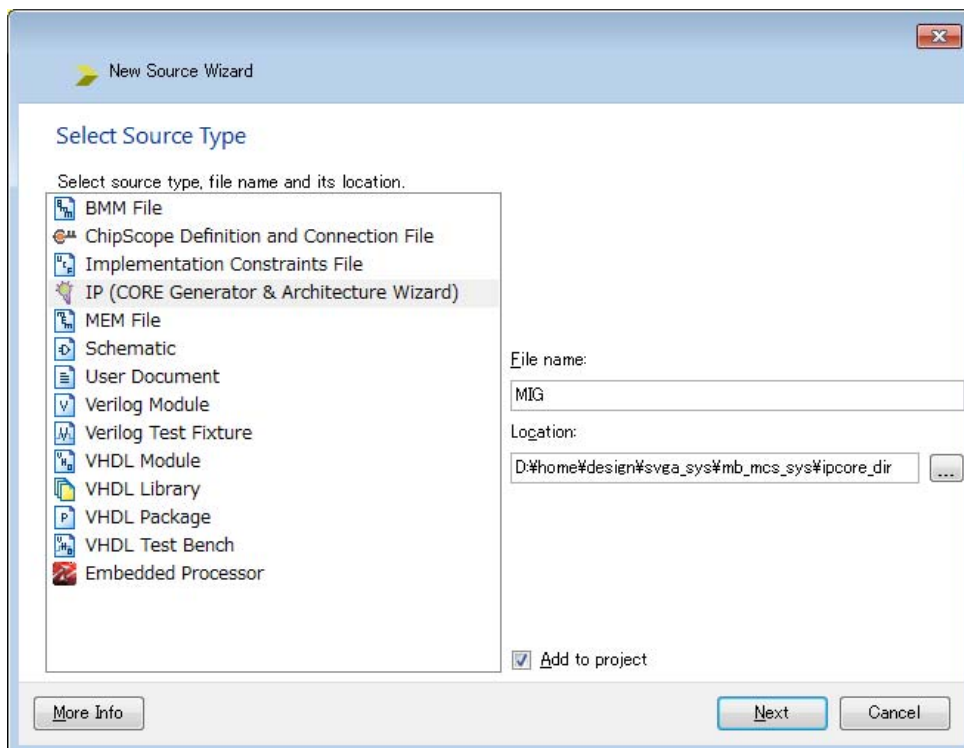
CLK_OUT5:200MHz [NoBuffer]

に指定して、Generate をクリック



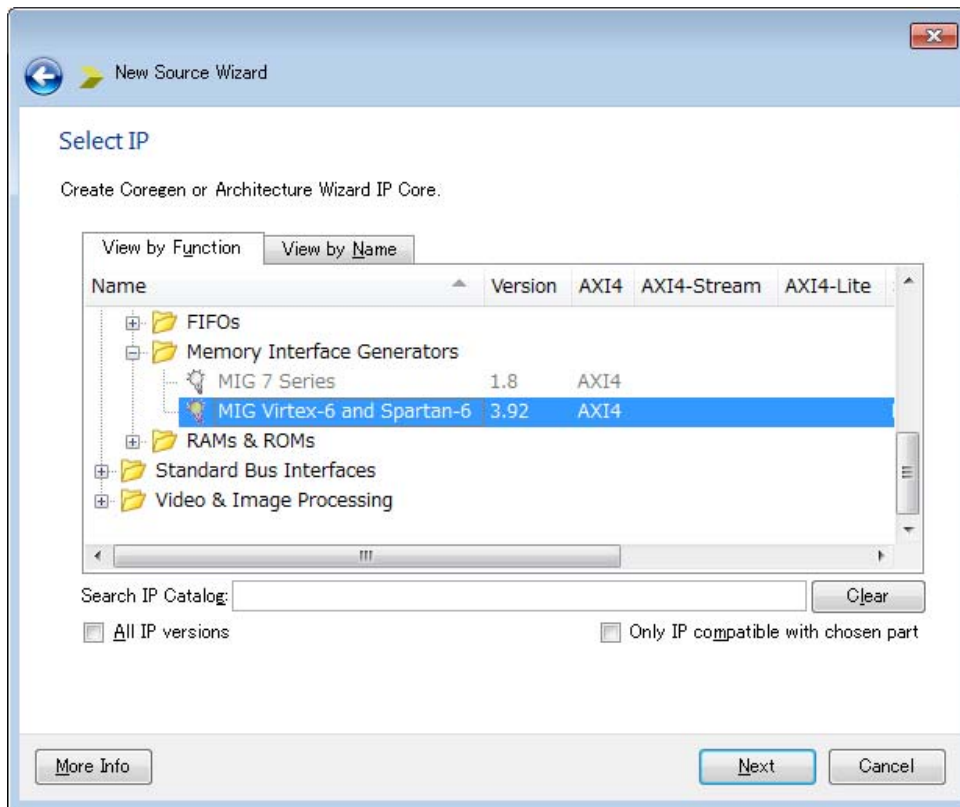


ISE Project Navigator→Project→New Source で新しいソースを追加する

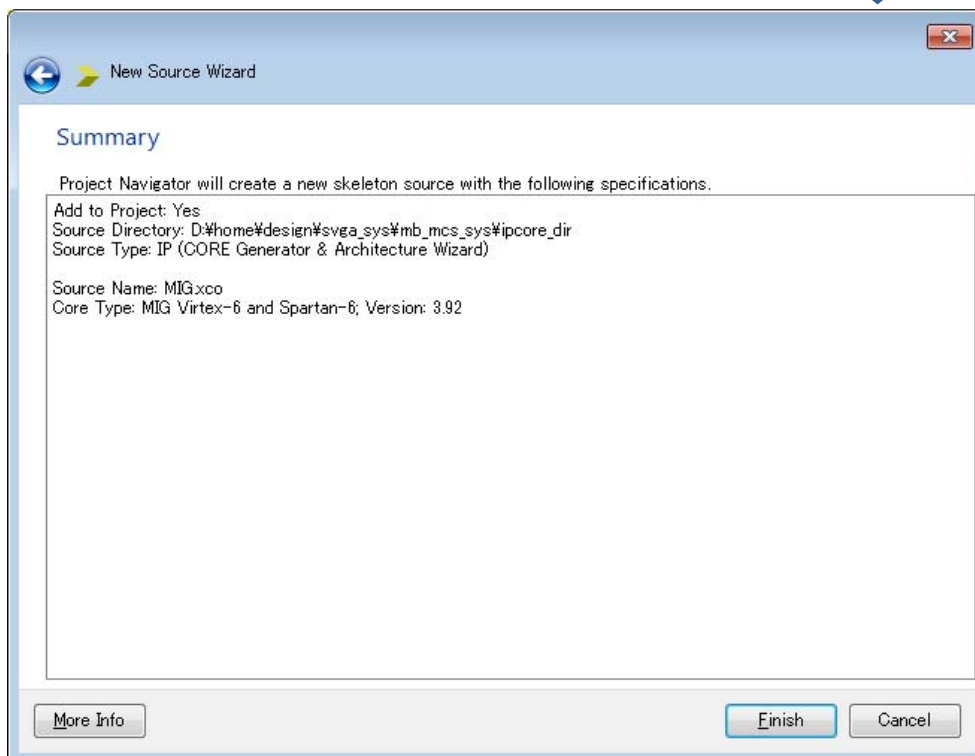


ファイル名に MIG を入力して、IP (COREGenerator) を選択して、Next をクリック
作成するコンポーネント名は MIG になる



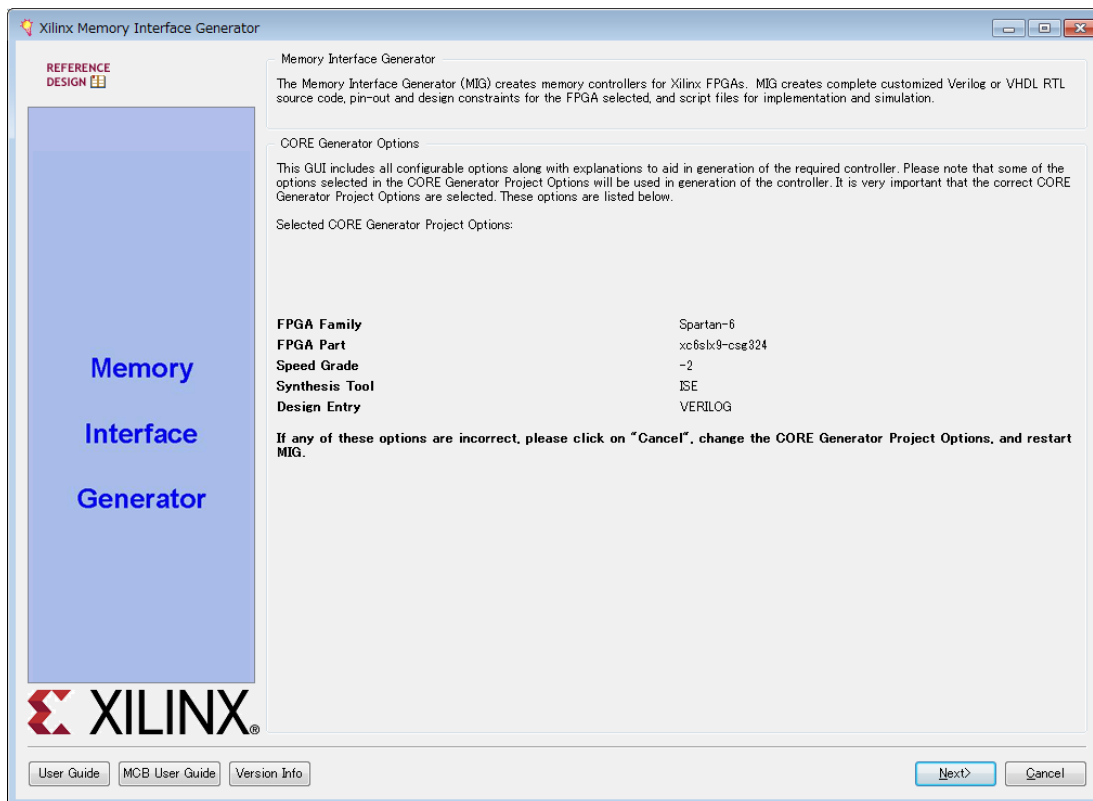


MIG Virtex-6 and Spartan-6 を選択して、Next をクリック

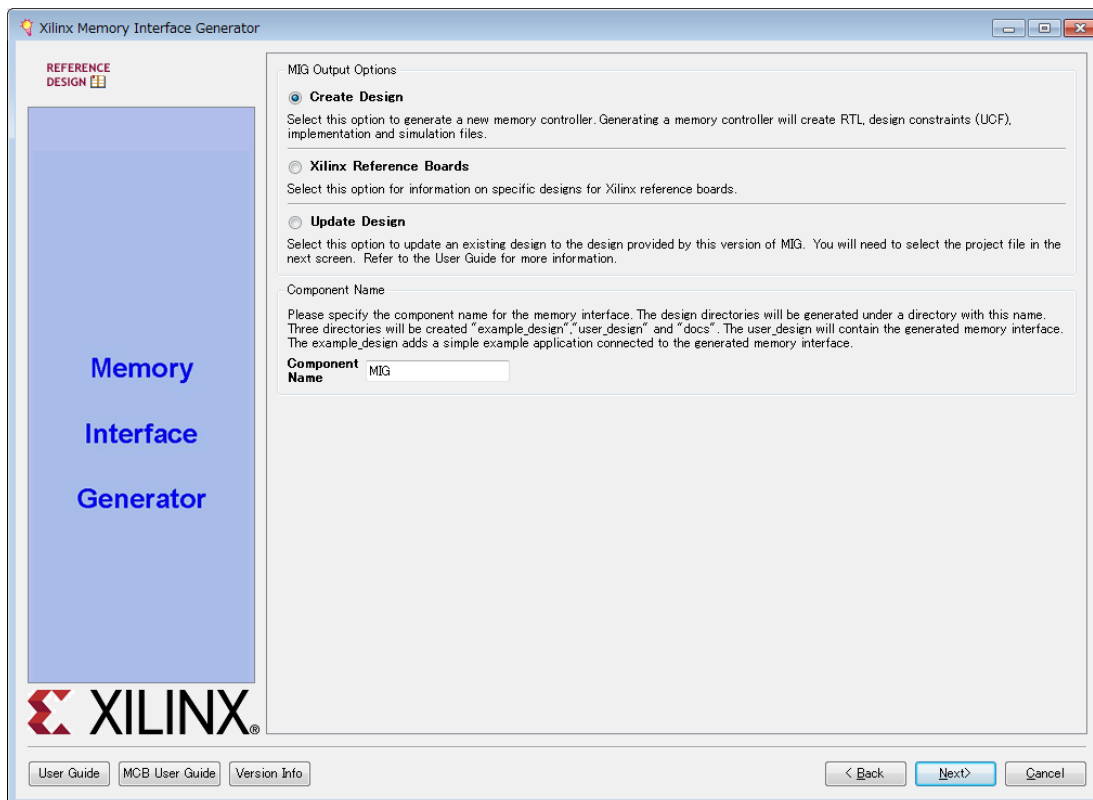


Next をクリック



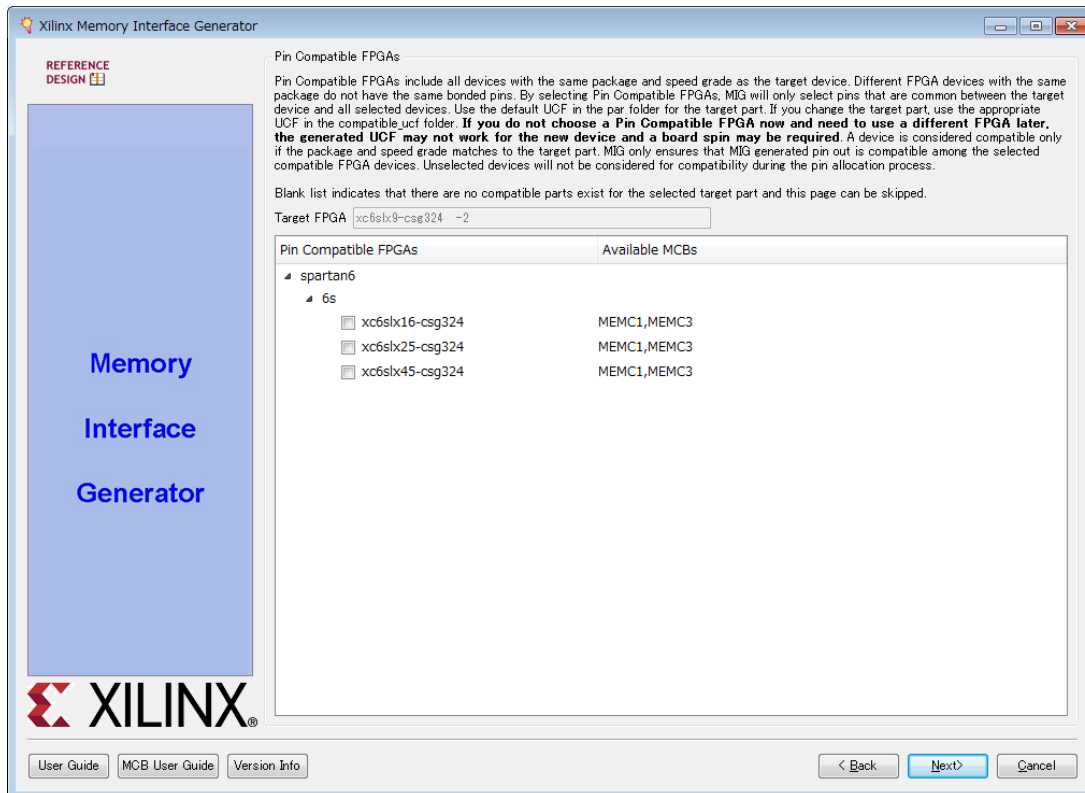


Next をクリック

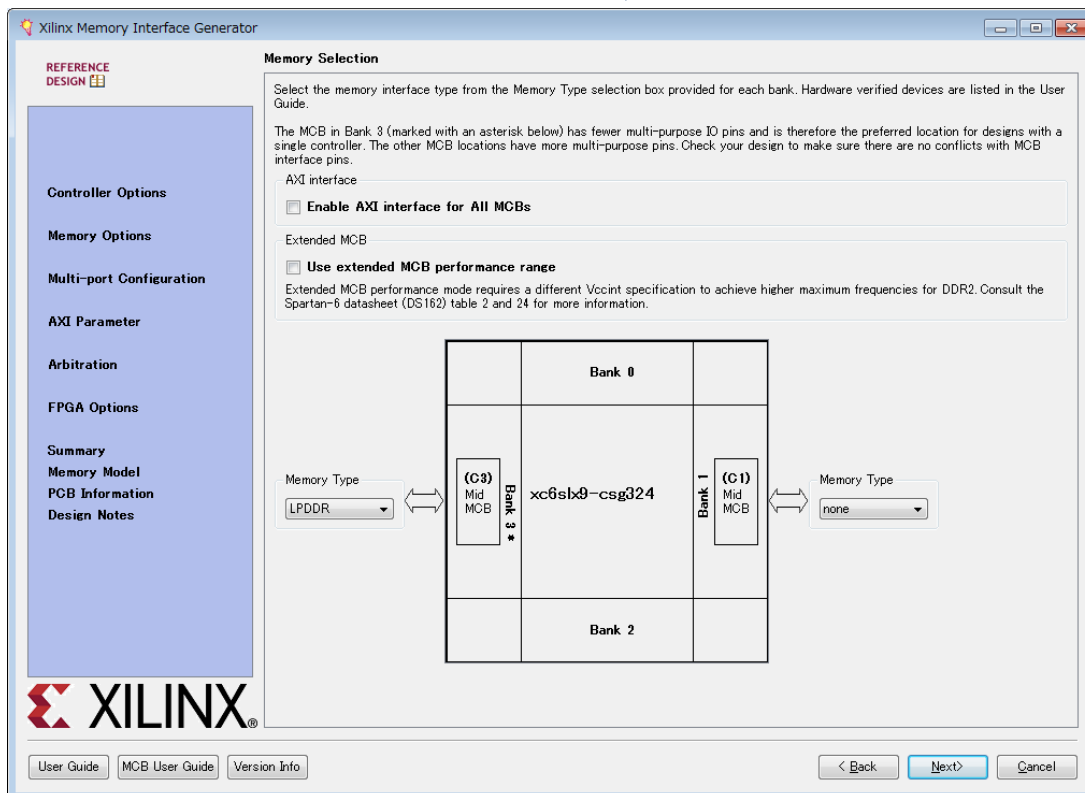


変更せずに Next をクリック



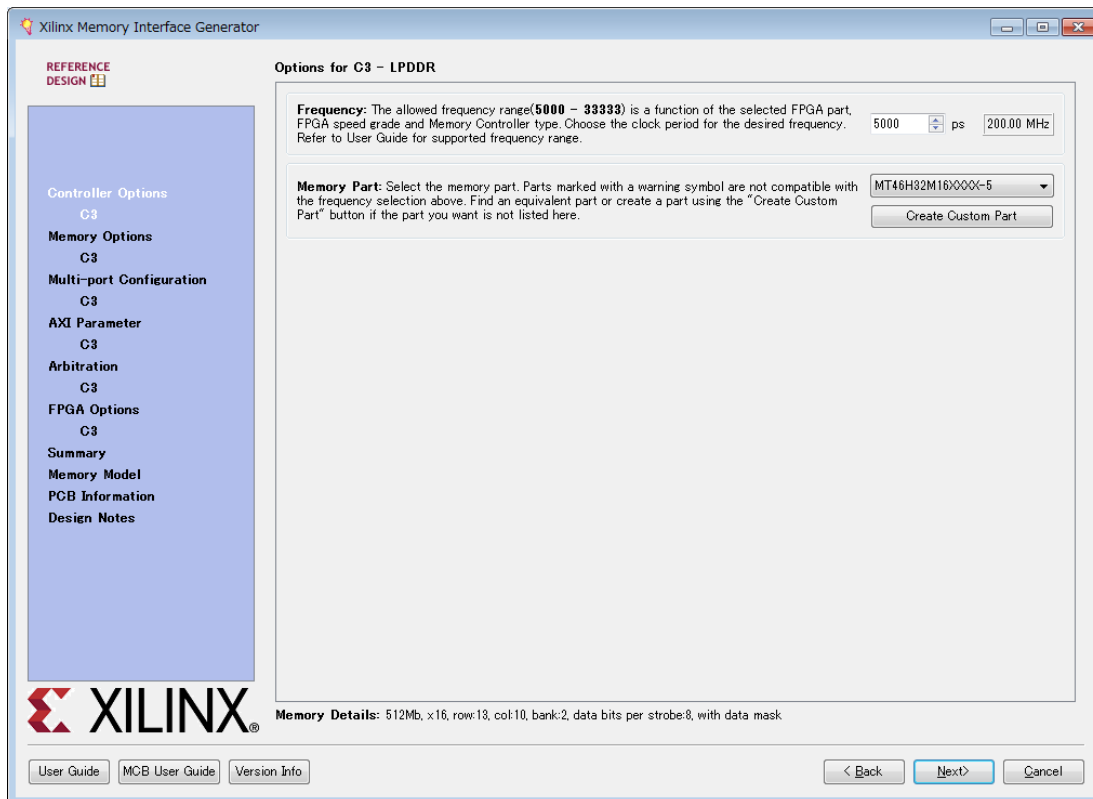


変更せずに Next をクリック

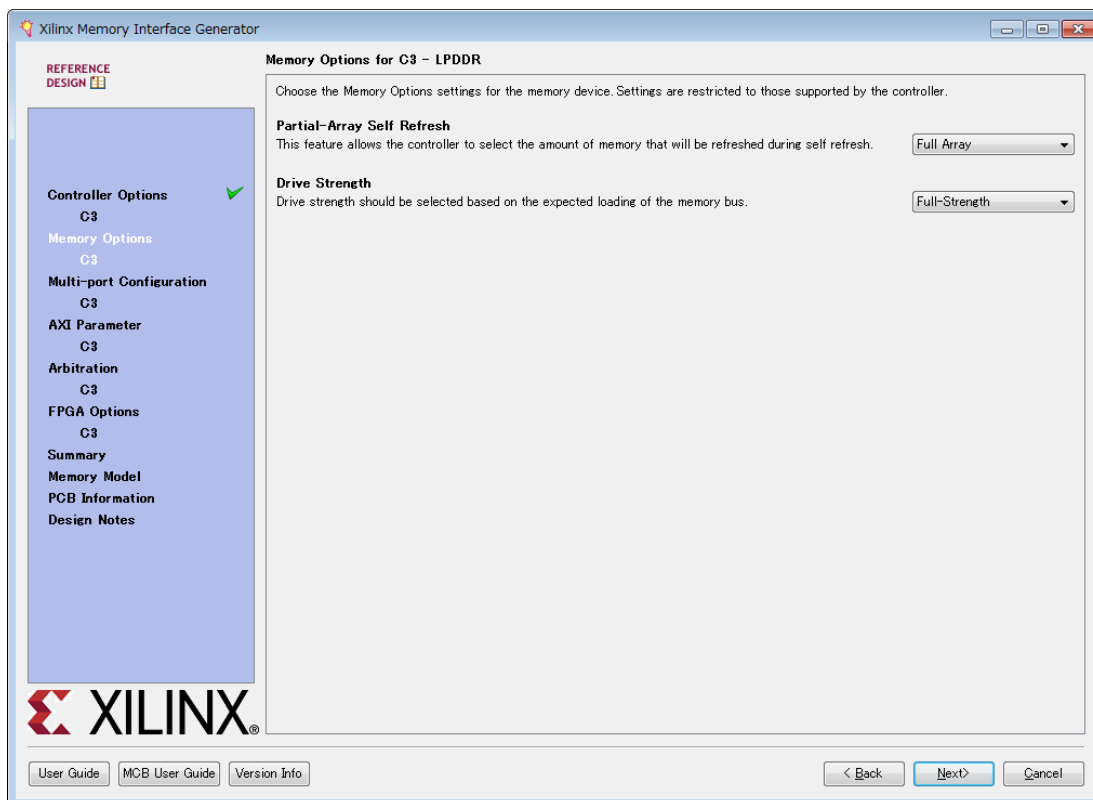


bank3 を LPDDR を指定





MicroBoard の実装メモリに合わせて MT46H32M16. . を選択する



変更せずに Next をクリック



Xilinx Memory Interface Generator

REFERENCE DESIGN

Controller Options ✓
C3
Memory Options ✓
C3
Multi-port Configuration
C3
AXI Parameter
C3
Arbitration
C3
FPGA Options
C3
Summary
Memory Model
PCB Information
Design Notes

Port Configuration for C3 - LPDDR

Select one of five configurations from the configuration menu and the ports from the table. As you select the port configuration, the below figure and table will get updated. You can select the number of ports in a configuration, and data port settings from the table.

Configuration Selection
Two 32-bit bi-directional and four 32-bit unidirectional ports

Port Selection	Interface	Direction
<input checked="" type="checkbox"/> Port0	NATIVE	Bi-directional
<input type="checkbox"/> Port1	NATIVE	none
<input checked="" type="checkbox"/> Port2	NATIVE	Read
<input type="checkbox"/> Port3	NATIVE	none
<input type="checkbox"/> Port4	NATIVE	none
<input type="checkbox"/> Port5	NATIVE	none

Memory Address Mapping Selection

User Address

☒ ROW BANK COLUMN
☐ BANK ROW COLUMN

User Guide MCB User Guide Version Info

< Back Next> Cancel

port0 と port2 を使用、port2 は SVGA_CTRL と接続のためリードのみに設定



Xilinx Memory Interface Generator

REFERENCE DESIGN

Controller Options ✓
C3
Memory Options ✓
C3
Multi-port Configuration ✓
C3
AXI Parameter
C3
Arbitration
C3
FPGA Options
C3
Summary
Memory Model
PCB Information
Design Notes

Arbitration for C3 - LPDDR

Select either round robin or custom for port arbitration. You can alter the port priorities in custom arbitration. For each time slot, the leftmost port number has the highest priority. The order of port priority decreases from left to right. Each port should be given highest priority in at least one time slot. Below ports will be set to a warning symbol if the port is not given highest priority in at least one time slot.

Select Arbitration Algorithm Round Robin

Timeslot 0 02
Timeslot 1 20
Timeslot 2 02
Timeslot 3 20
Timeslot 4 02
Timeslot 5 20
Timeslot 6 02
Timeslot 7 20
Timeslot 8 02
Timeslot 9 20
Timeslot 10 02
Timeslot 11 20

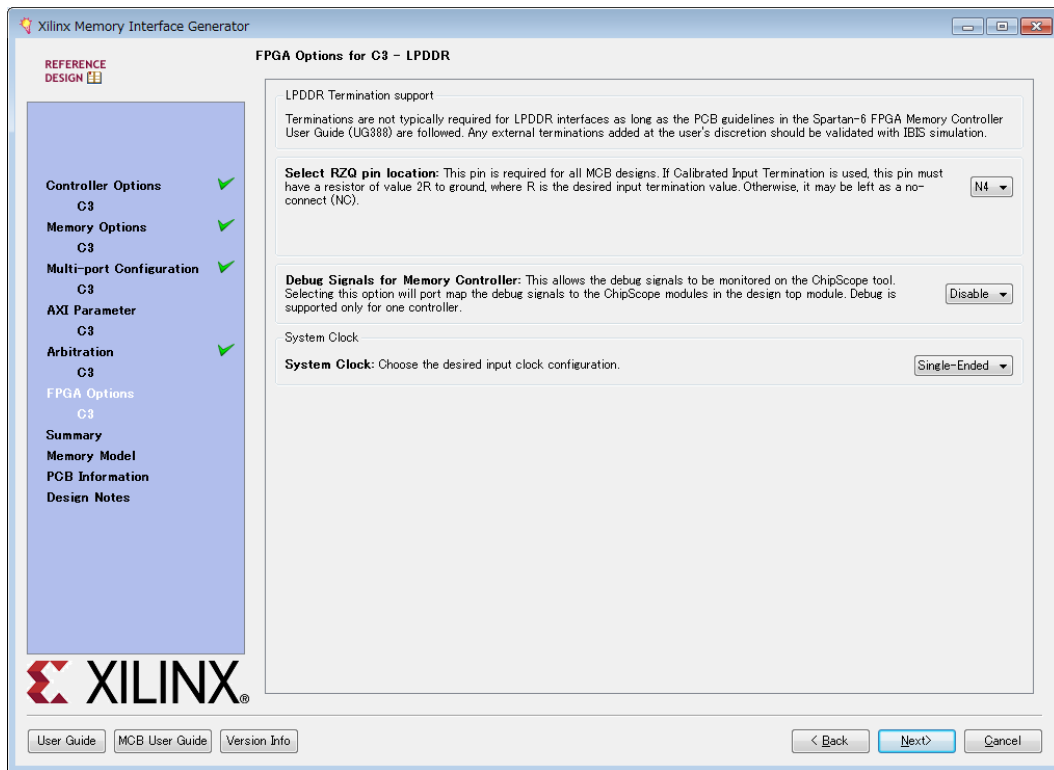
Port0 ✓ Port1 ⚠ Port2 ✓ Port3 ⚠ Port4 ⚠ Port5 ⚠

User Guide MCB User Guide Version Info

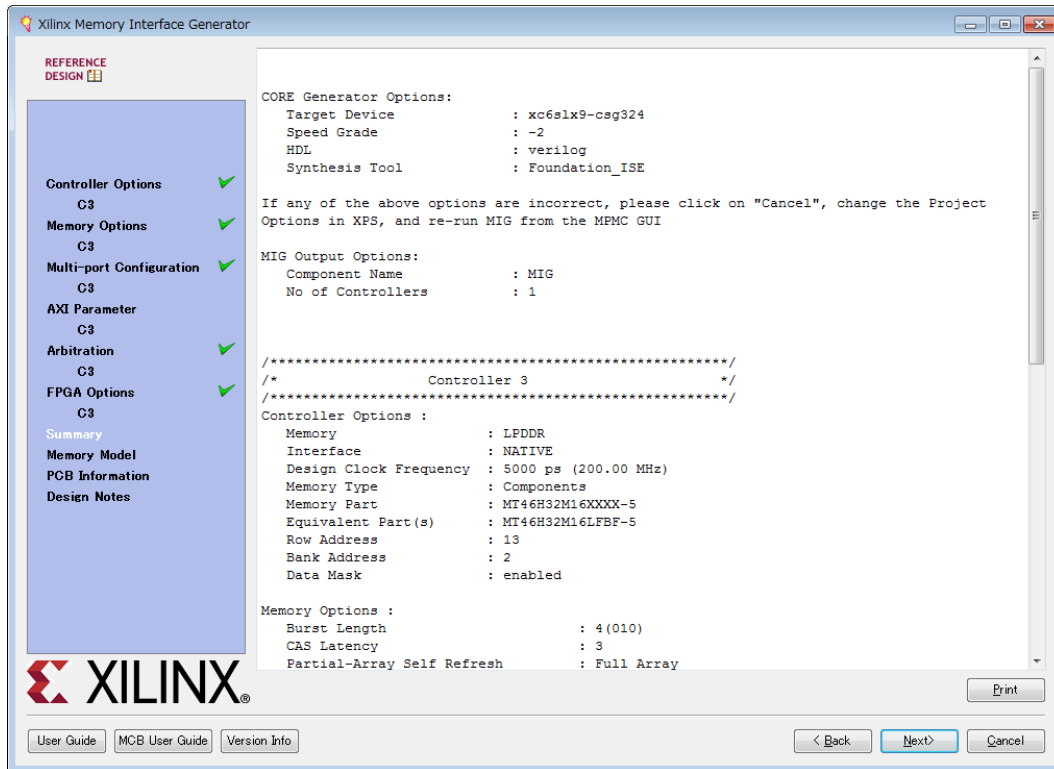
< Back Next> Cancel

アクセスのアービトレーションの選択、変更せずに Next をクリック



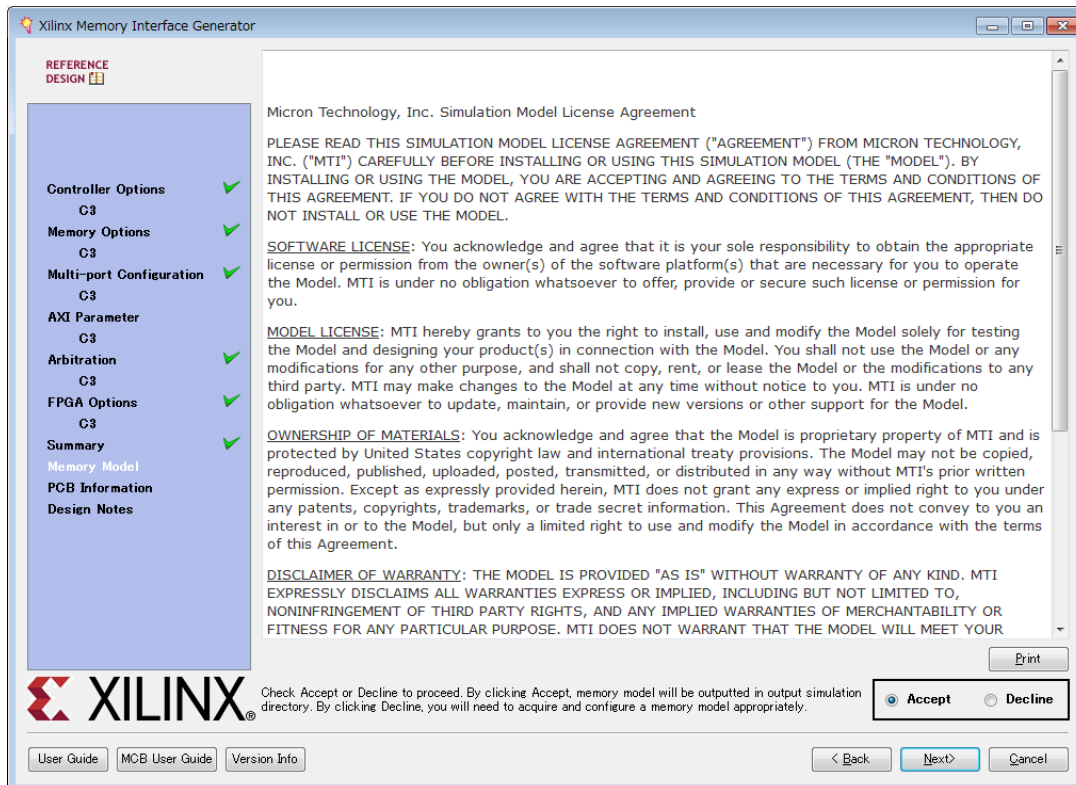


System Clock は Single-Ended を選択

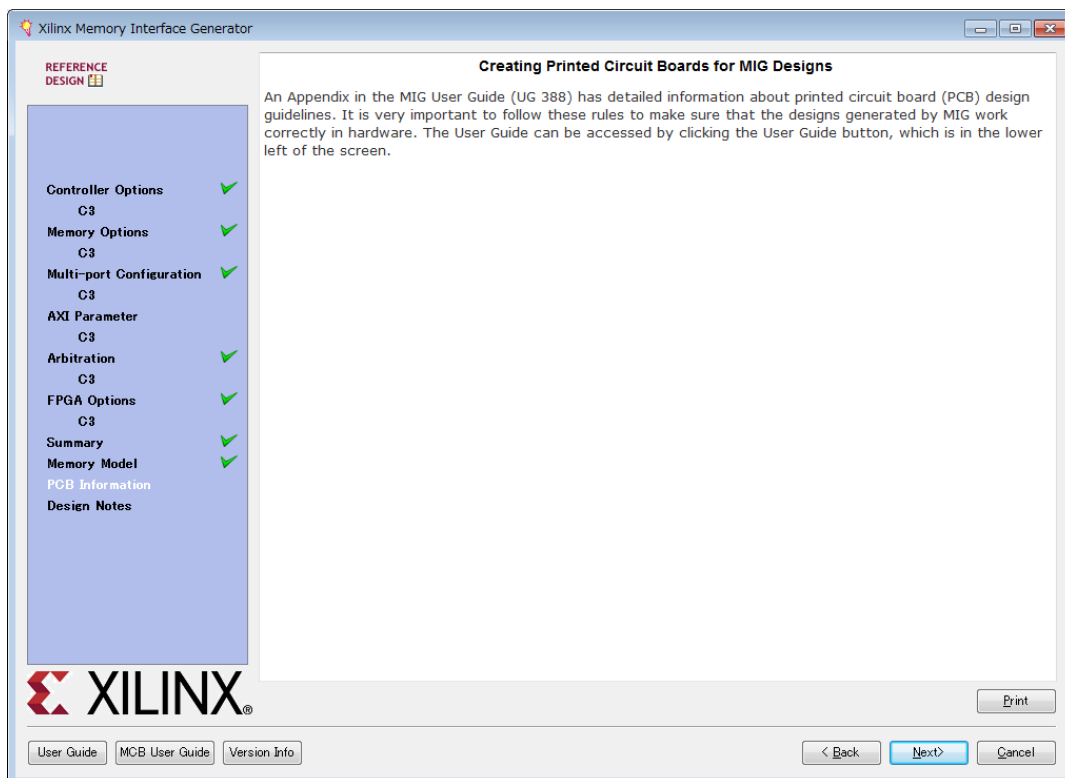


作成時のサマリーが表示される。Next をクリック



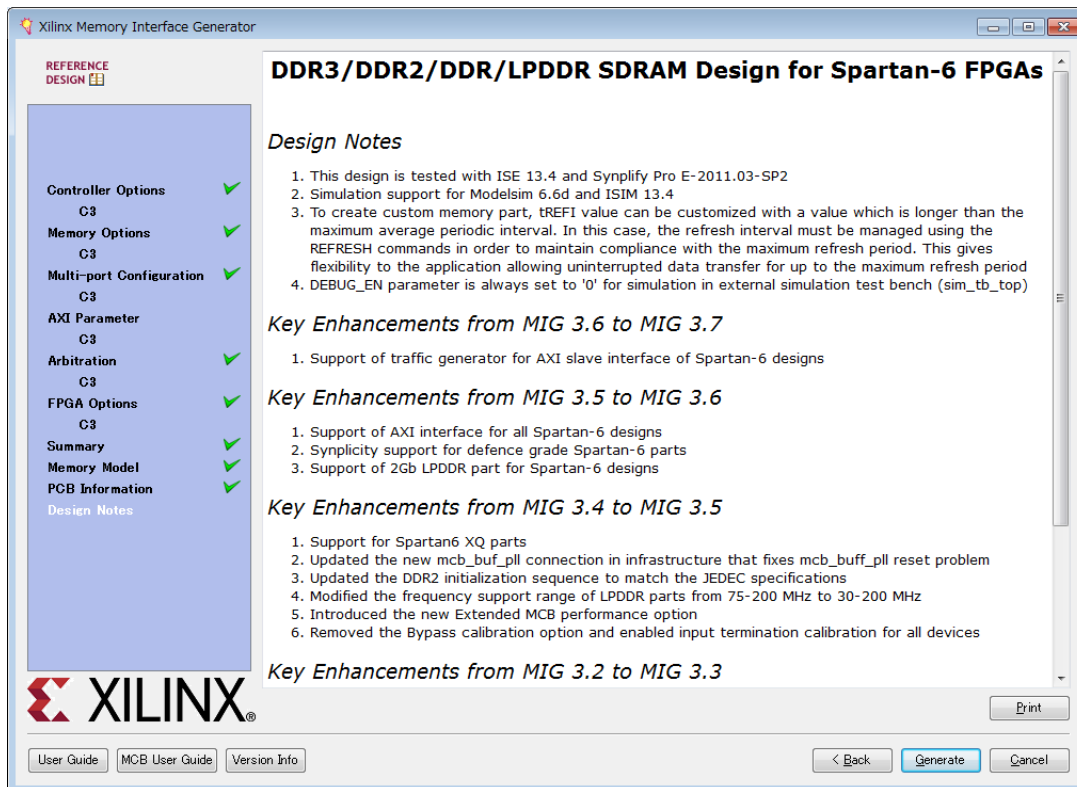


シミュレーションモデルのライセンスに Accept を選択、Next をクリック

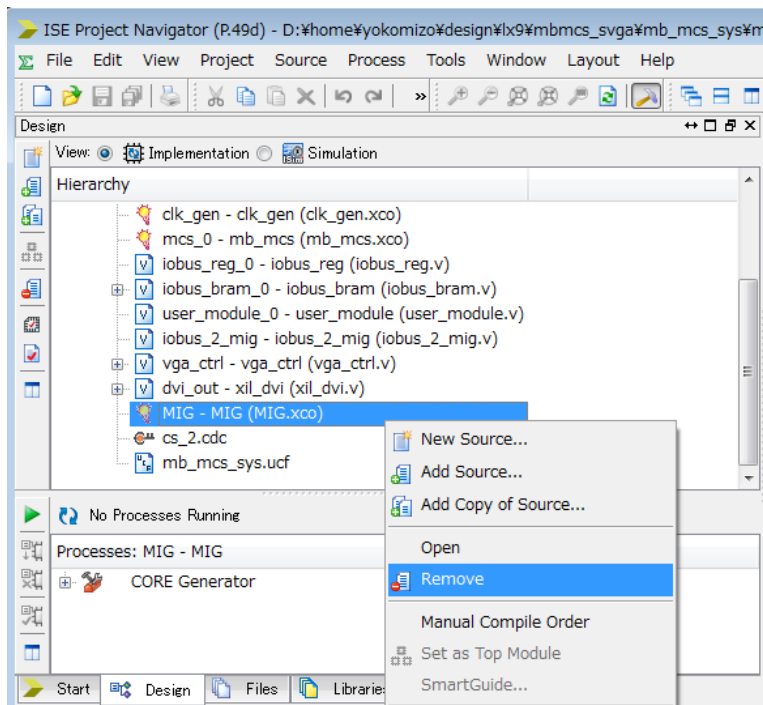


Next をクリック



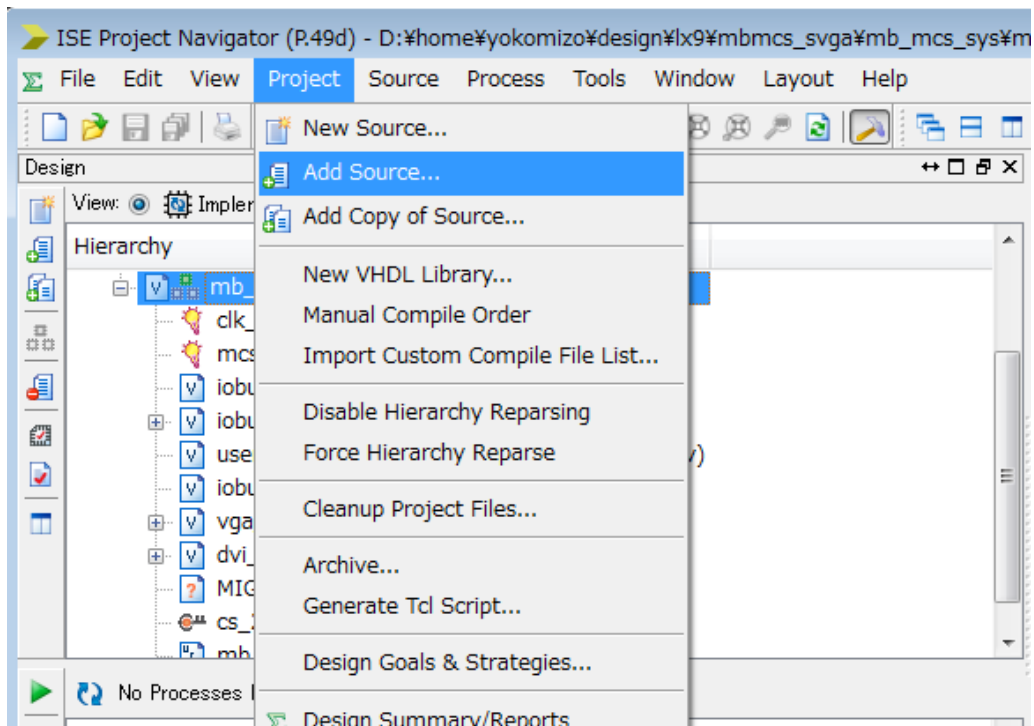


Generate をクリックすると MIG の作成が開始

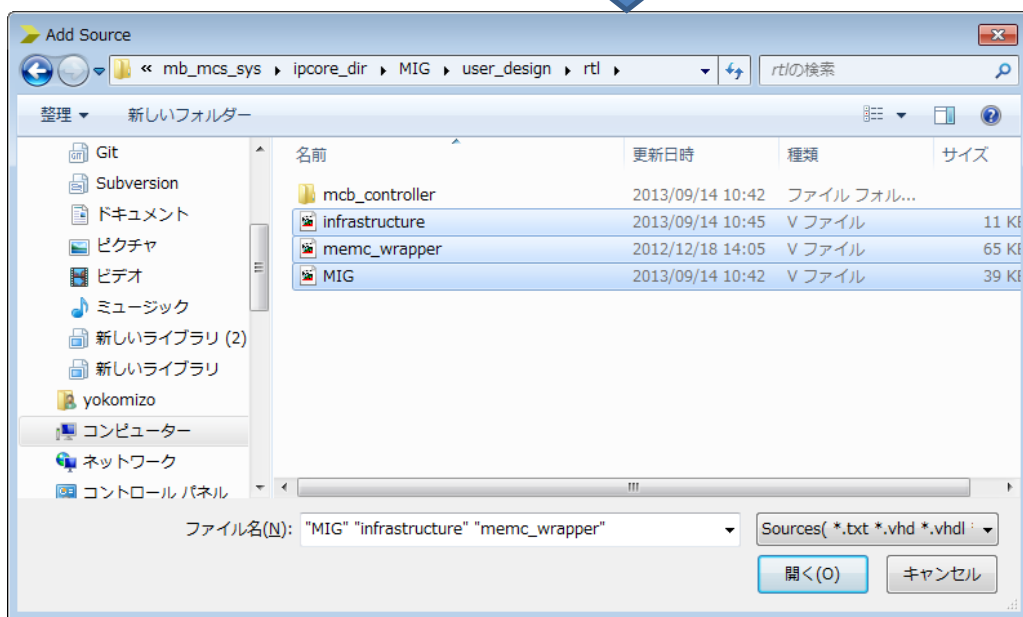


論理合成を実行すると MIG.xco が組み込まれるが、Verilog-HDL モデルに変更するので、MIG.xco を削除する。



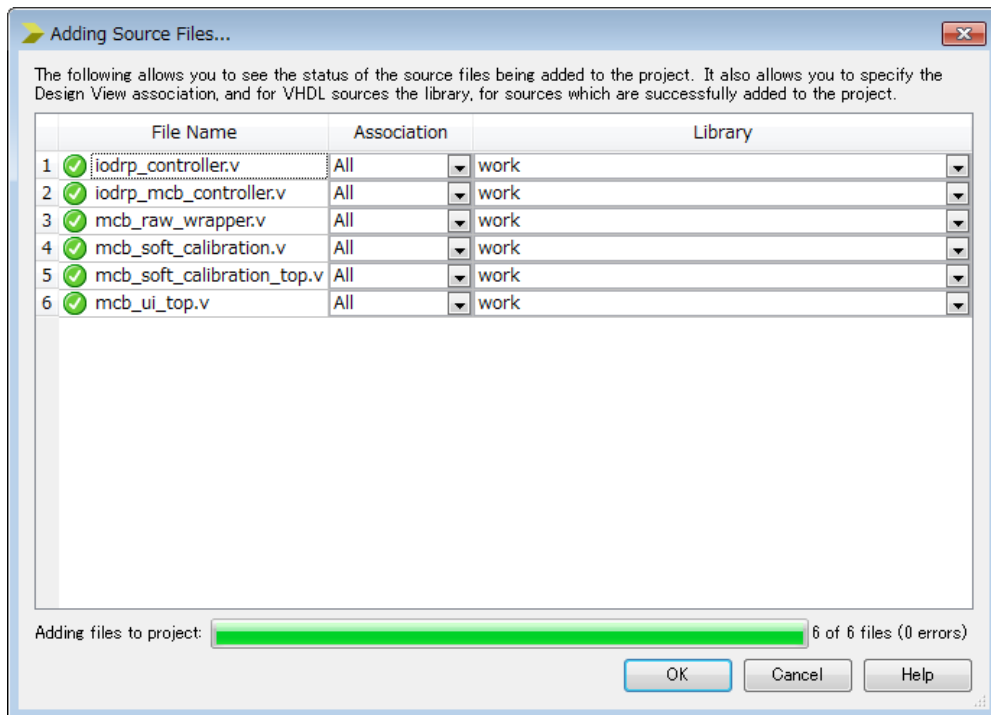


Add Source で MIG の Verilog-HDL 記述をプロジェクトに追加する

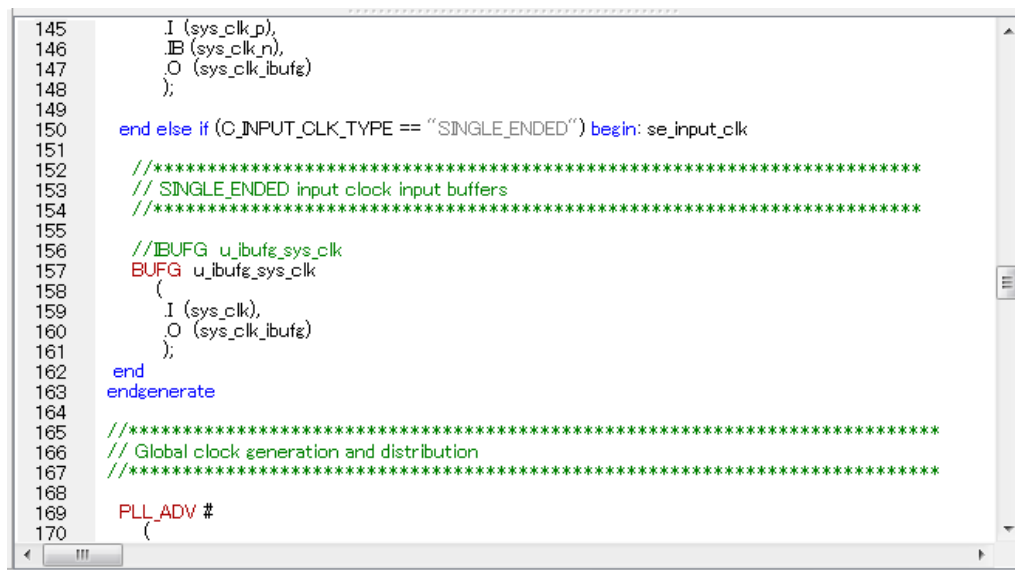


ipcor_dir/MIG/uadr_design/rtl の下の Verig-HDL ソースファイルを選択

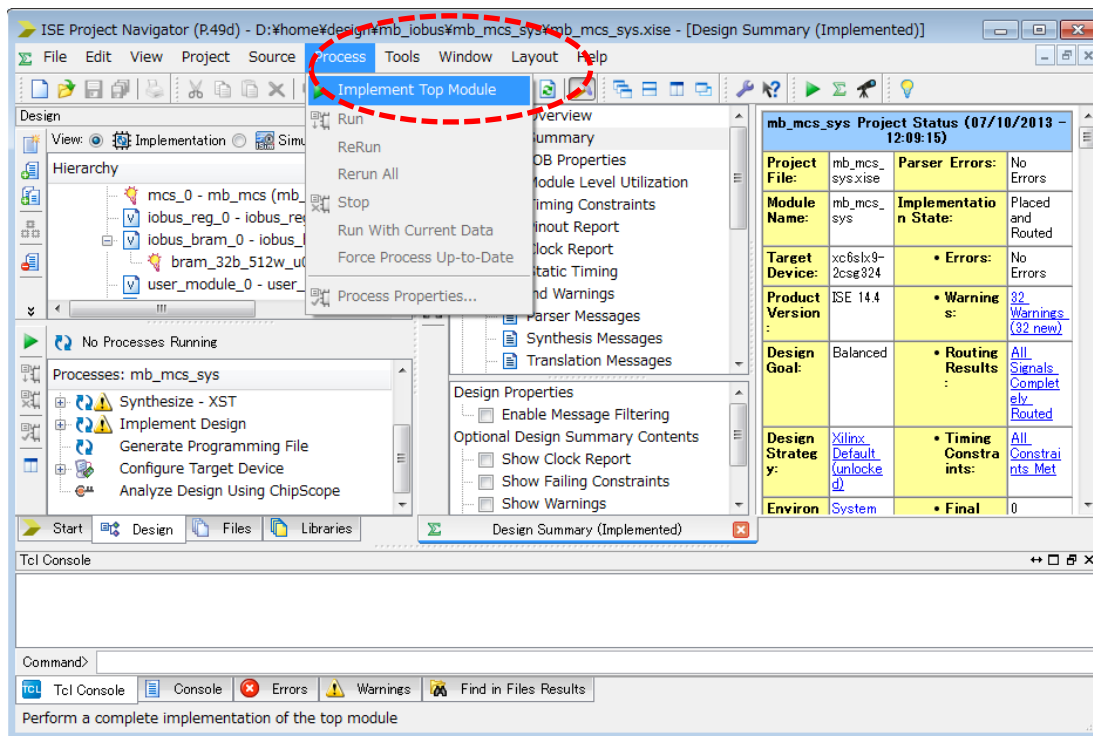




ソースファイルを追加すると、Verilog-HDL モデルの MIG が組み込まれる。



infrastructure.v を開き_ibufg_sys_clk に使用される入力用バッファ[IBUF]を通常バッファ[BUF]へ変更する。



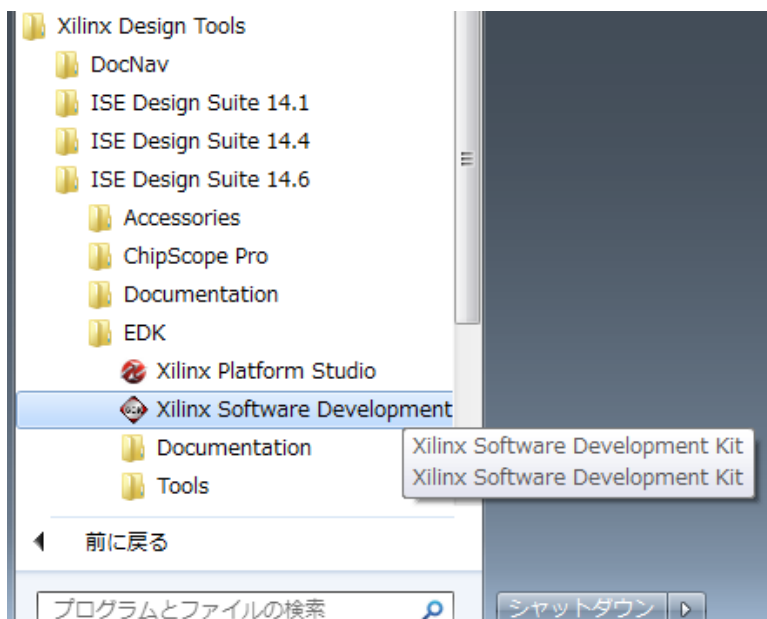
インプリメンテーションの実行、Process→Implement Top Module をクリック

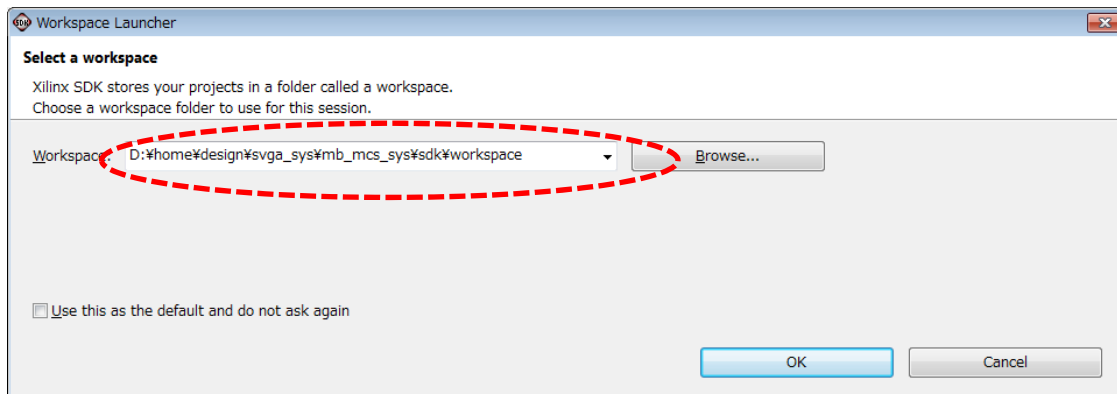


次にXilinx Software Development Kit(以降 SDK)でソフトウェアを作ります。

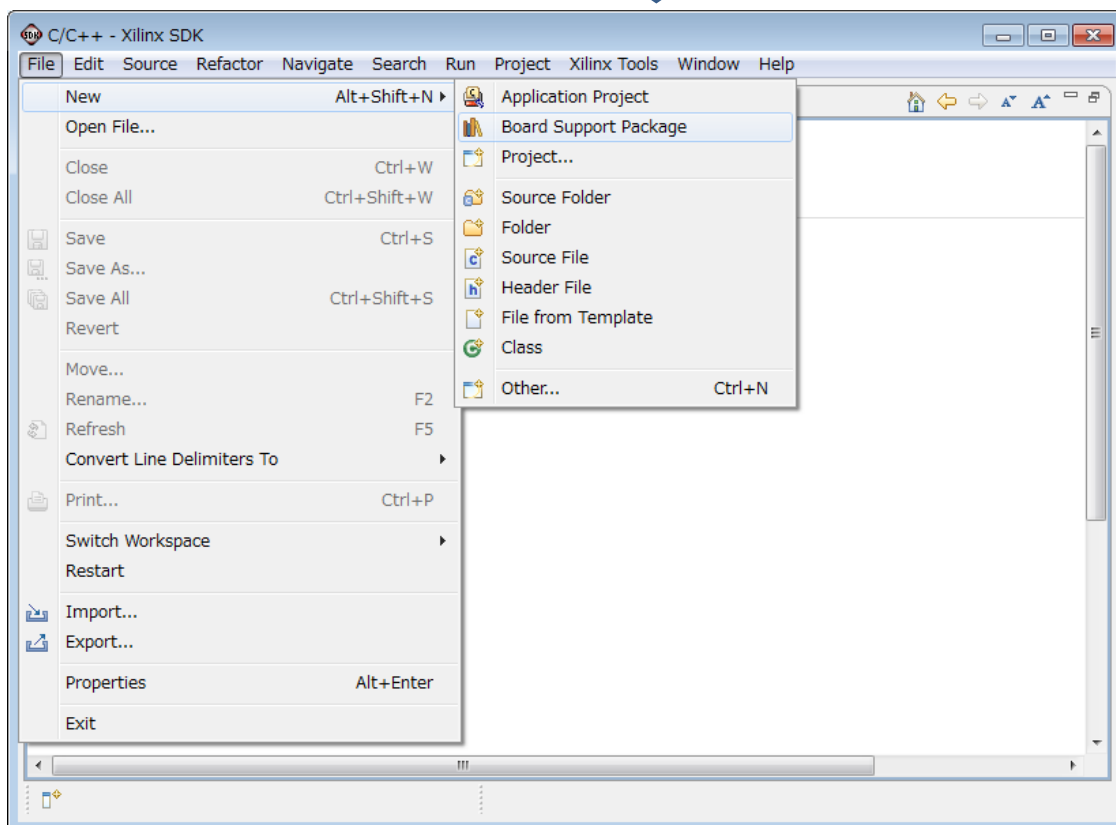
・SDK の起動

スタートメニューから「Xilinx Design Tools」→「ISE Design Suite 14.6」→「EDK」→「Xilinx Software Development Kit 」を起動してください。

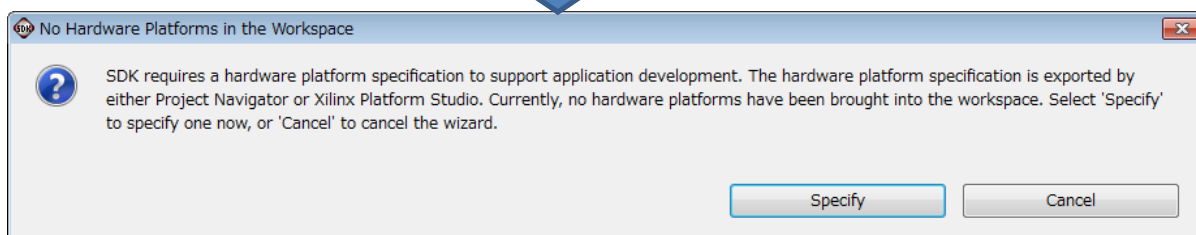




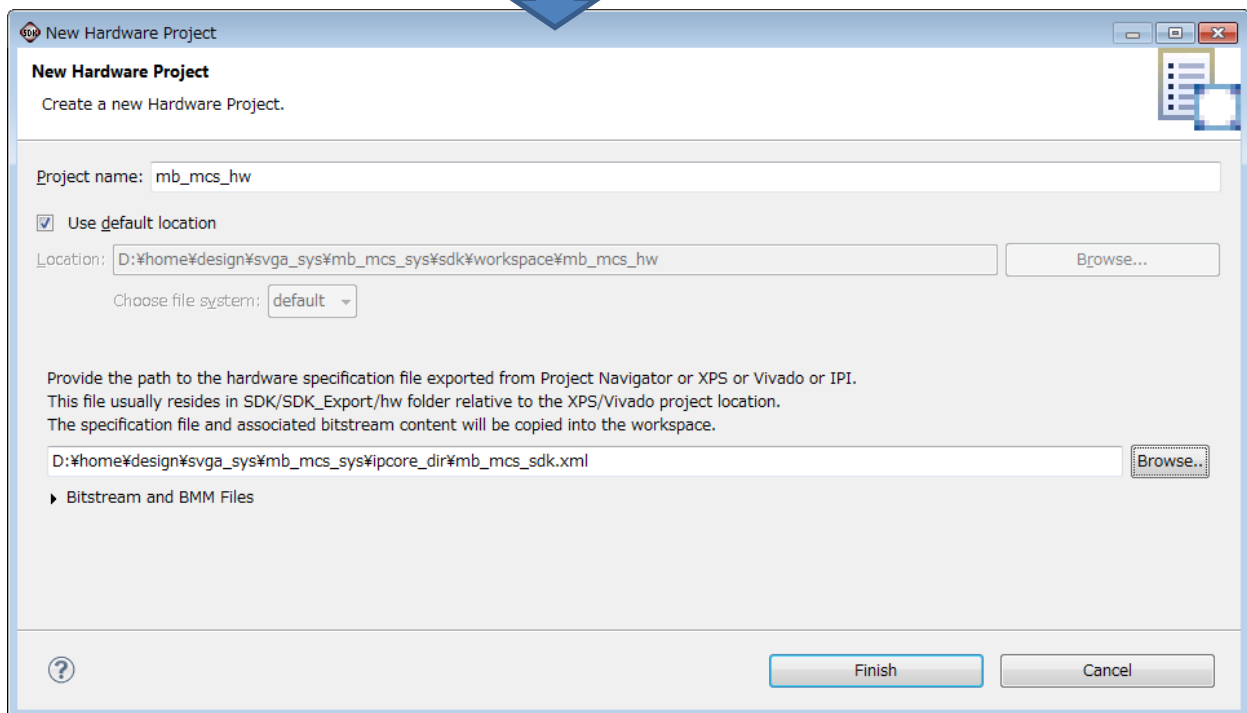
SDK を起動するとワークスペースを指定が要求されます。設計フォルダ/sdk/workspace を設定



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



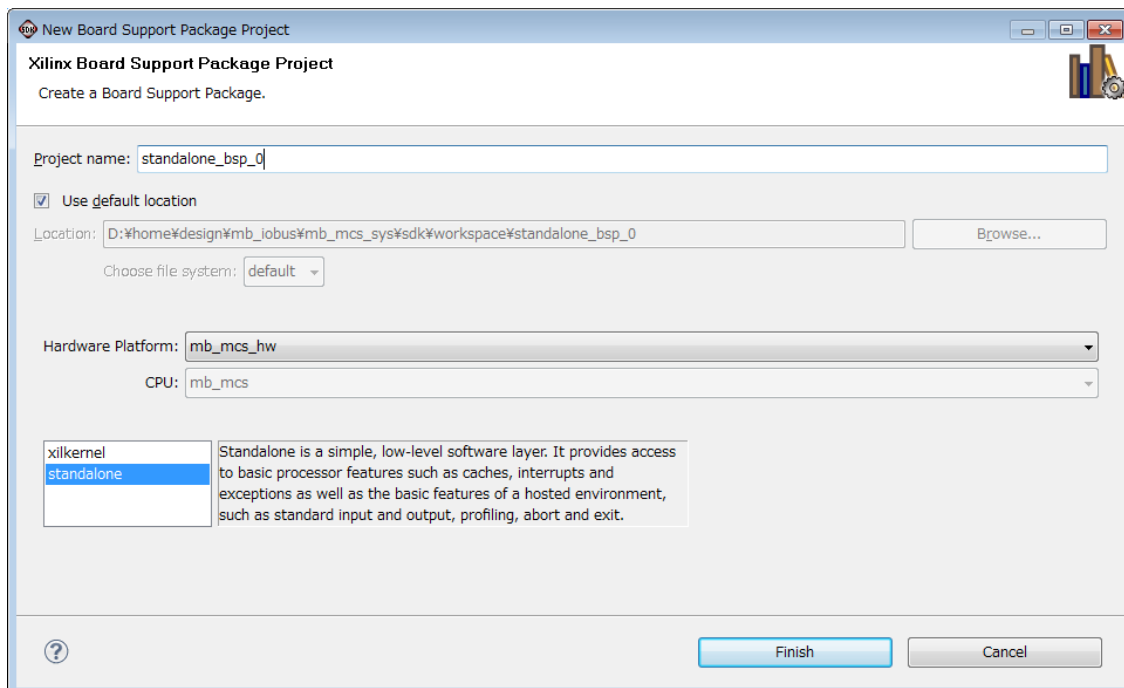
Hardware Platform を定義するか聞かれるので、specify を選択して定義する



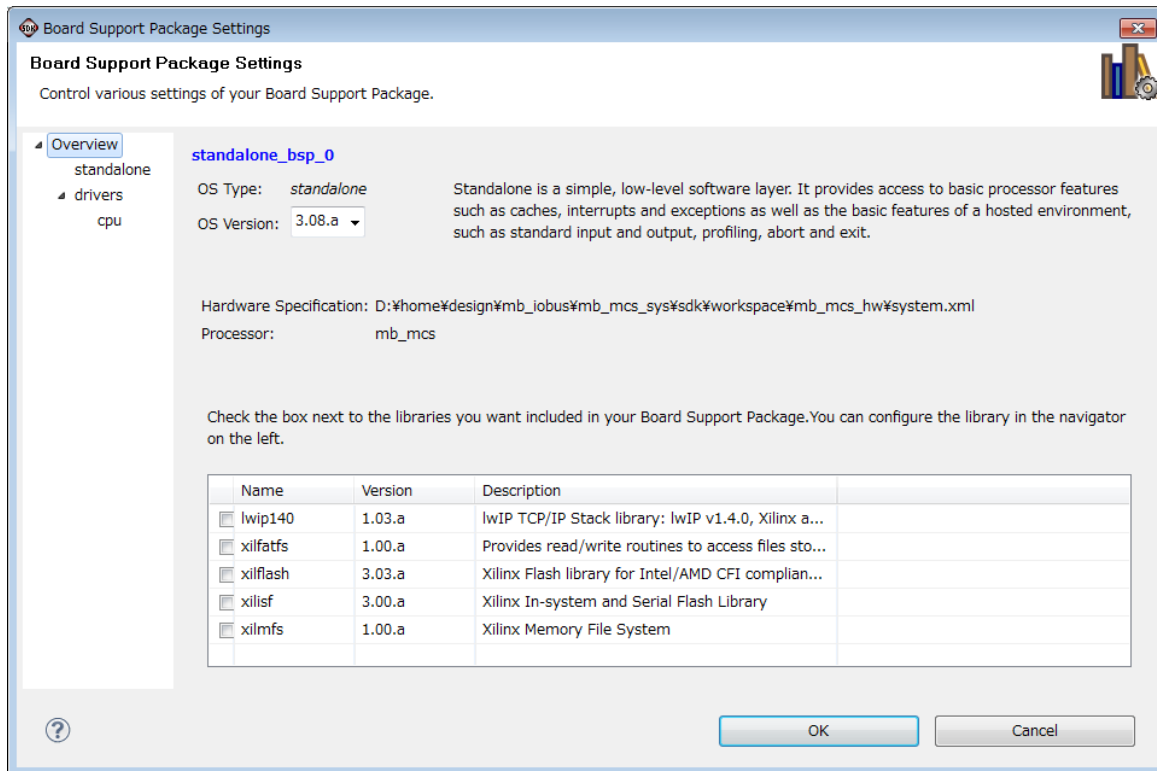
SDK の管理するハードウェアプロジェクト名指定

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

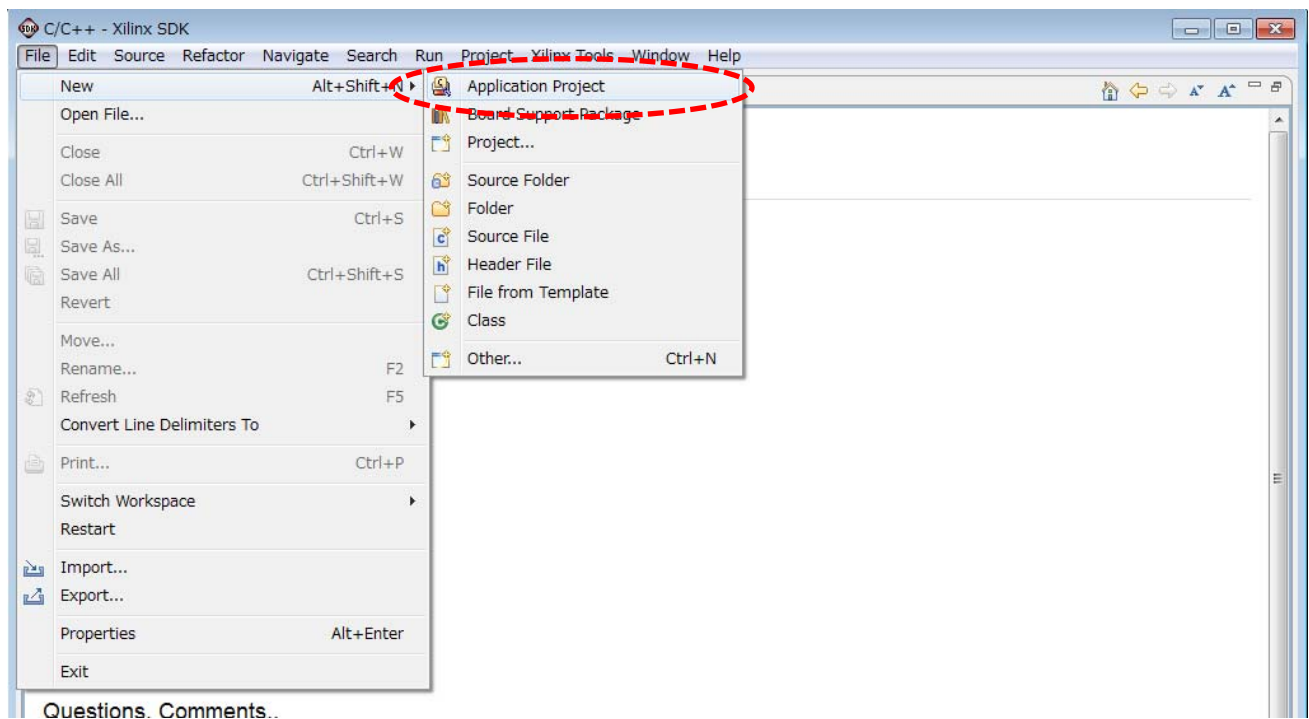
Finish をクリック



ボードサポートパッケージの定義

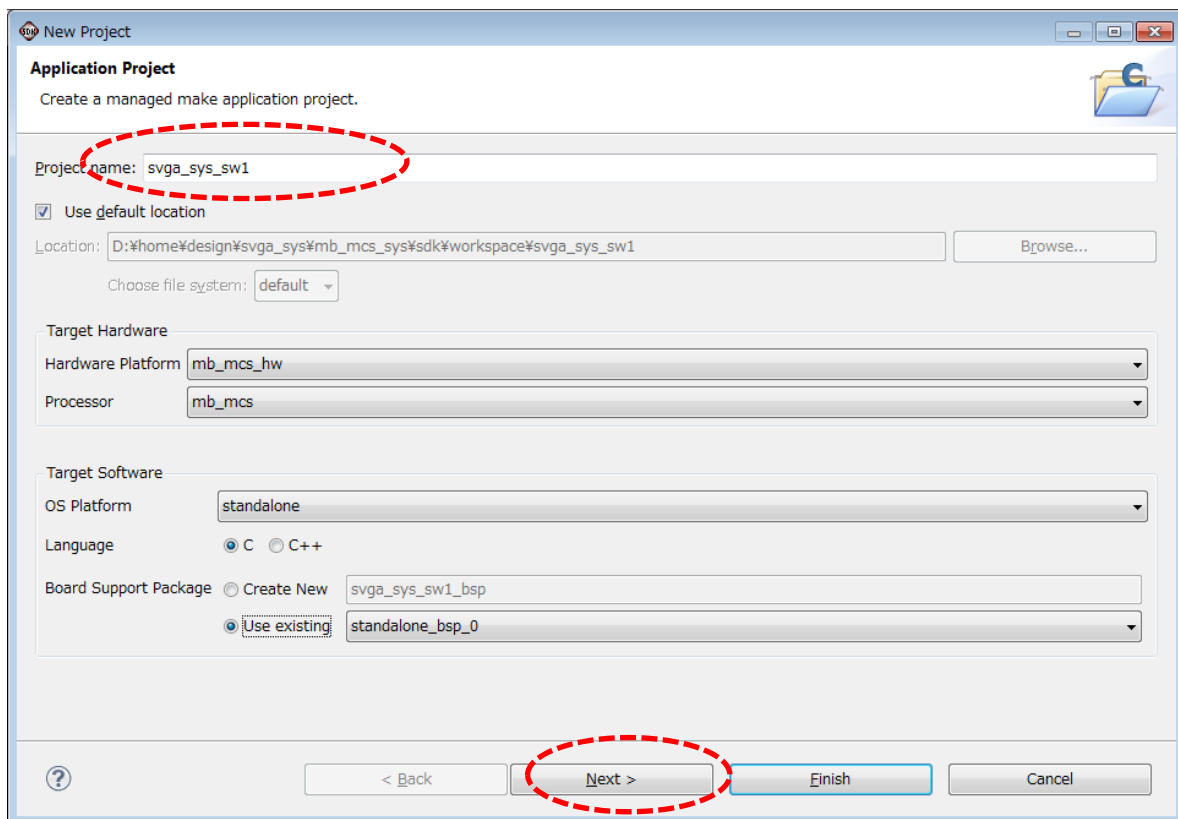


ボードサポートパッケージのオプション定義



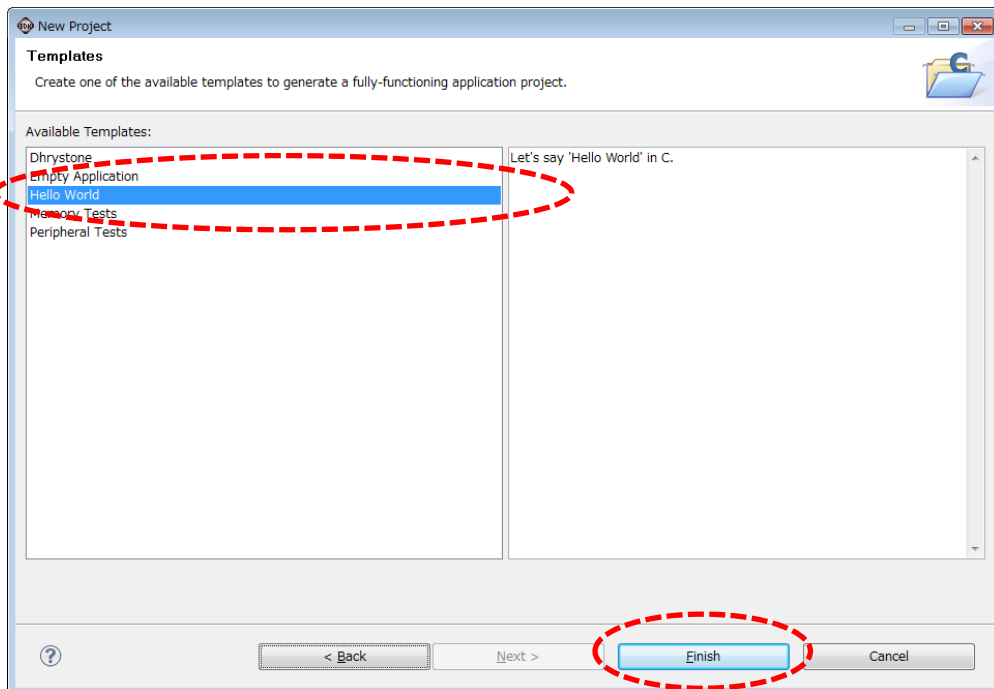
新規のソフトウェアプロジェクト作成、File→New→Application Project





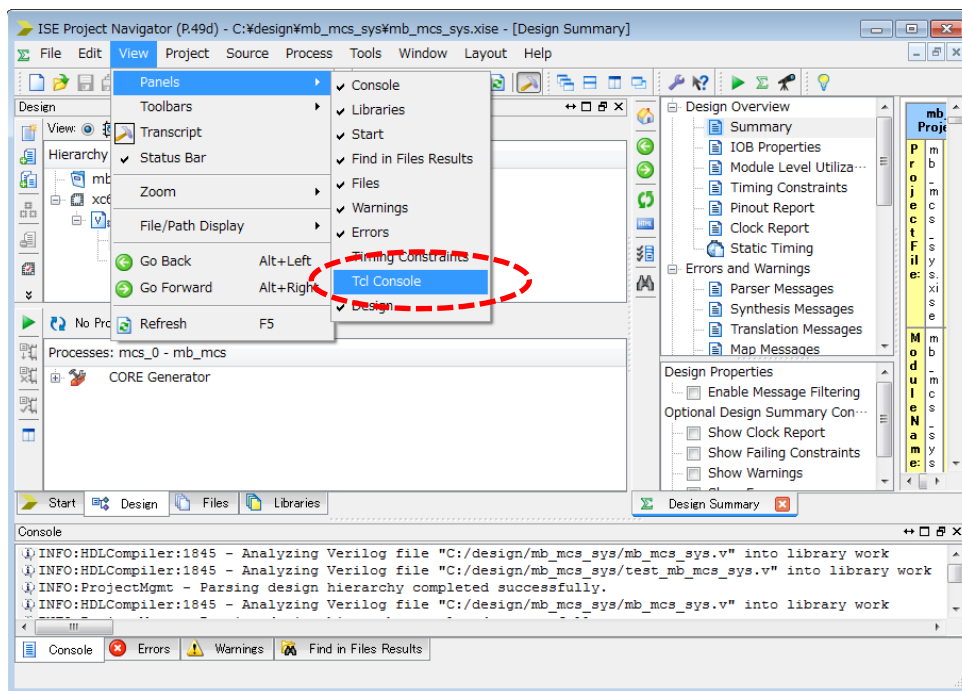
ソフトウェアプロジェクト名指定:hello_world_sw1

Next をクリック

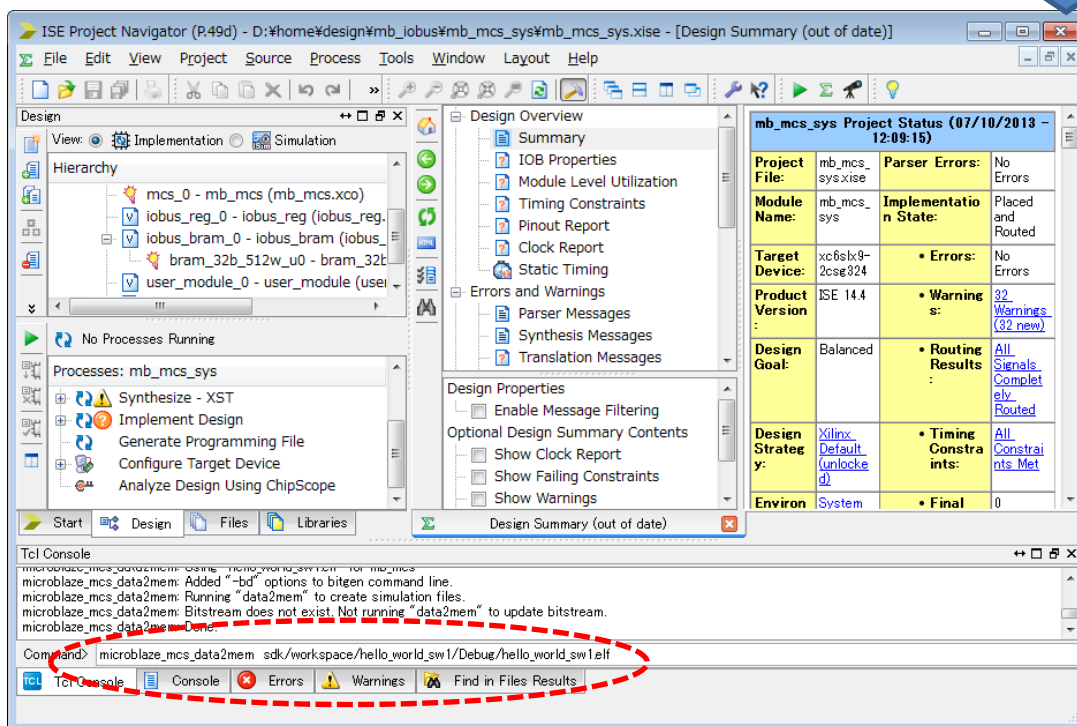


テンプレートに Hello World 選択後、Finish をクリックでソフトウェアプロジェクトが作成される。
ソフトウェアの実行ファイル (sdk/workspace/svga_sys_sw1/Debug/svga_sys_sw1.elf) が作られる。





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/svga_sys_sw1/Debug/svga_sys_sw1.elf [Enter]

FPGA で動作確認します。

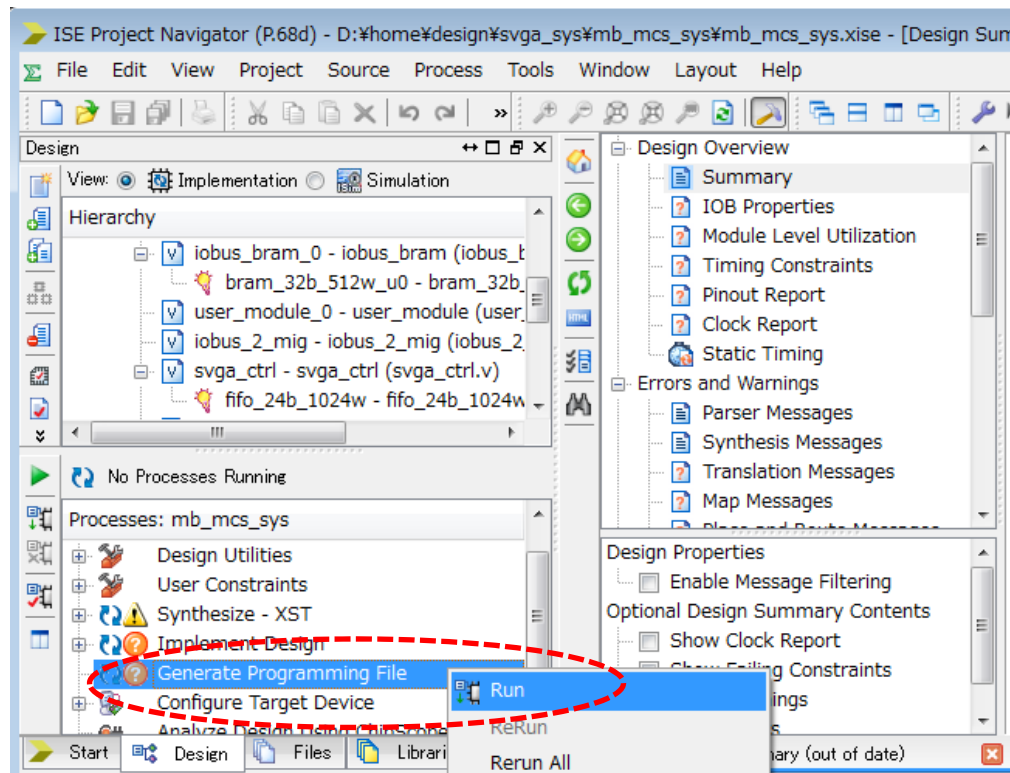
LX9 マイクロボードを拡張ボードに搭載し、プログラミング用 USB と UART 用 USB をそれぞれ PC の USB ポートに接続します。拡張ボードの HDMI 端子とディスプレイを接続します。

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。

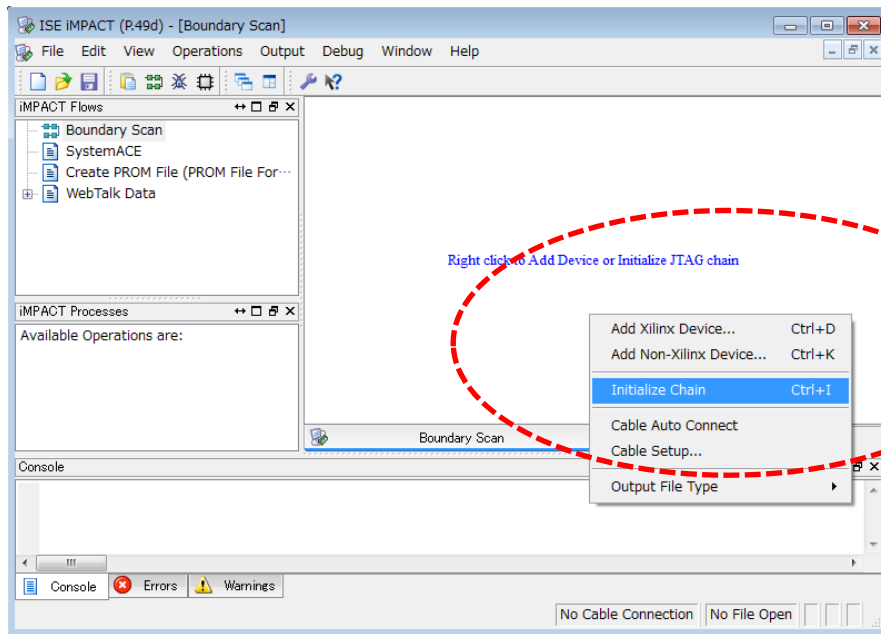
ターミナルソフトの送信改行コードを [CR] に設定



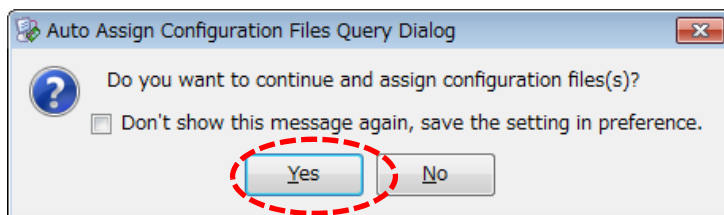
ISE に戻る



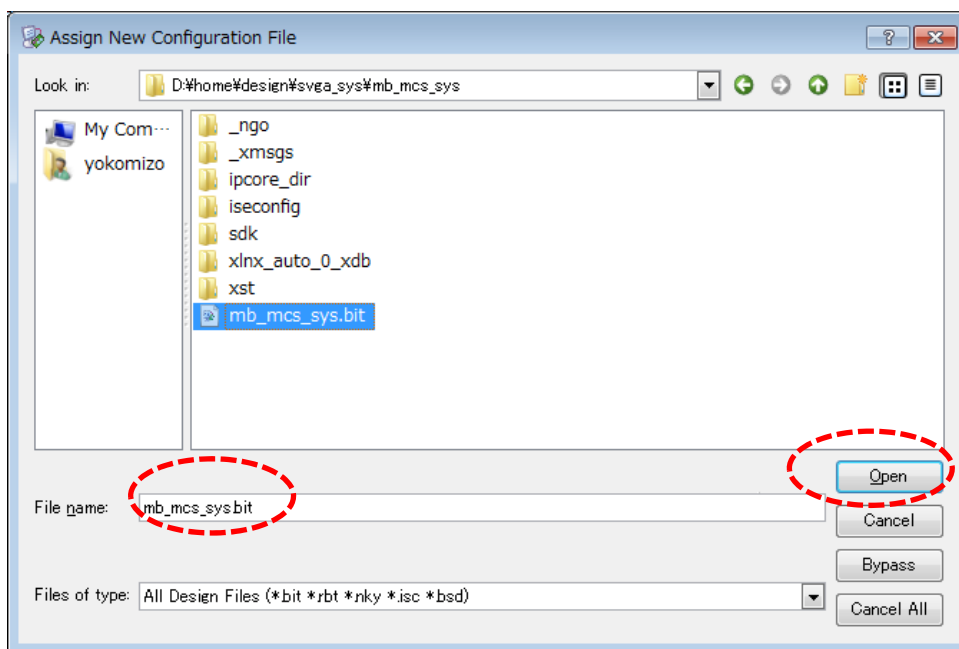
FPGA へ書き込む mb_mcs_sys.bit 作成



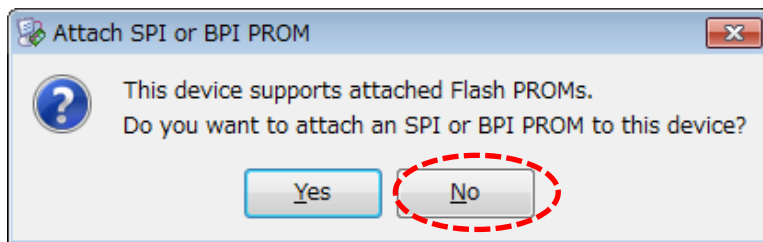
BoundaryScan のウィンドをマウス右ボタン押して、Initialize Chain 選択して FPAG を検出する



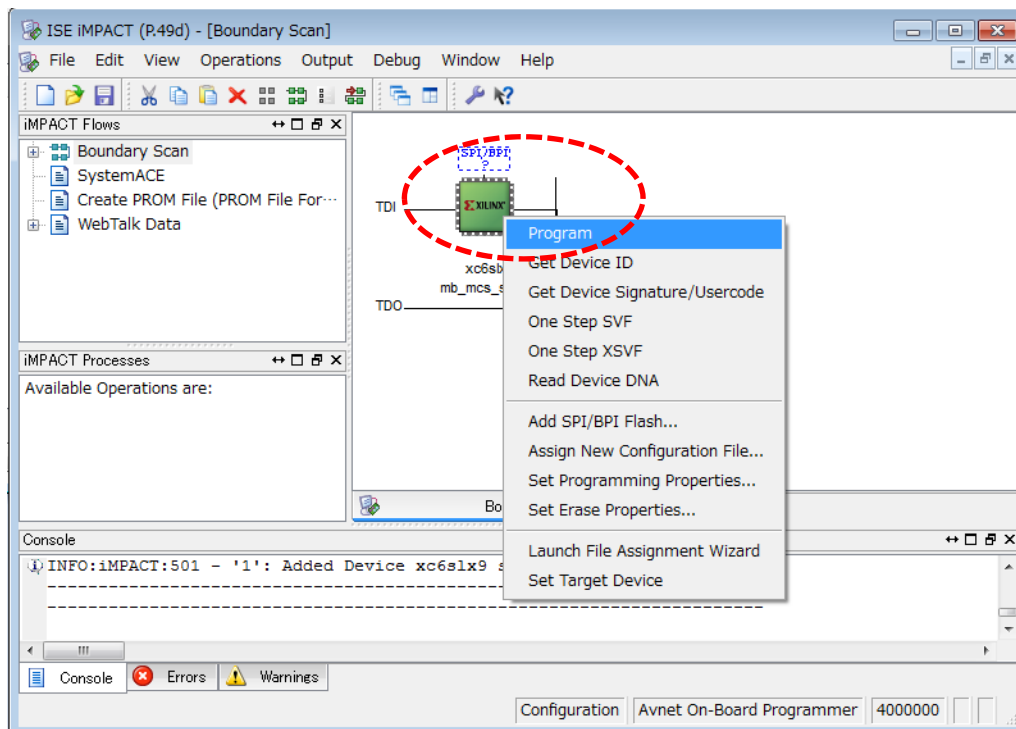
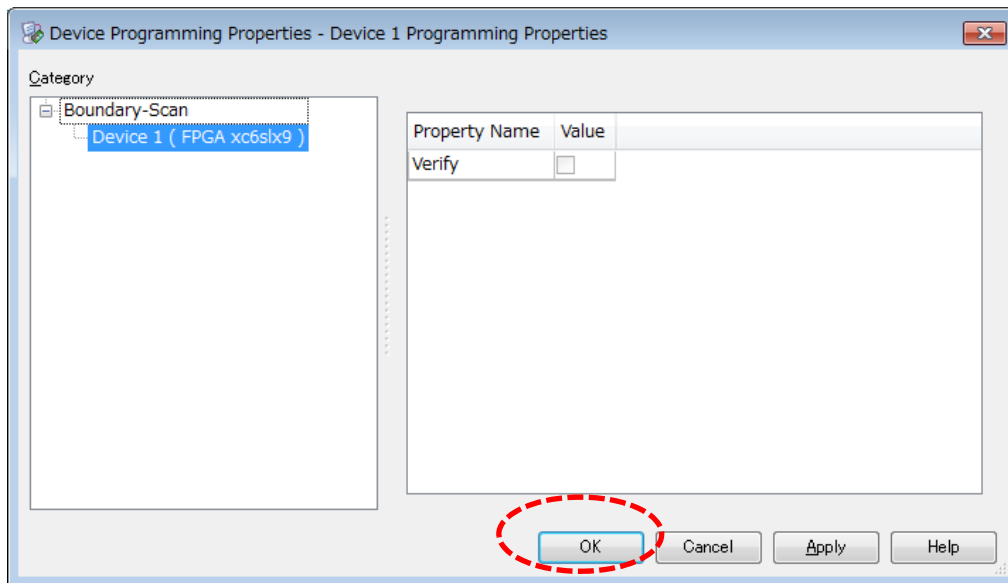
Yes で FPGA に書き込むファイルを指定する



mb_mcs_sys.bit を指定

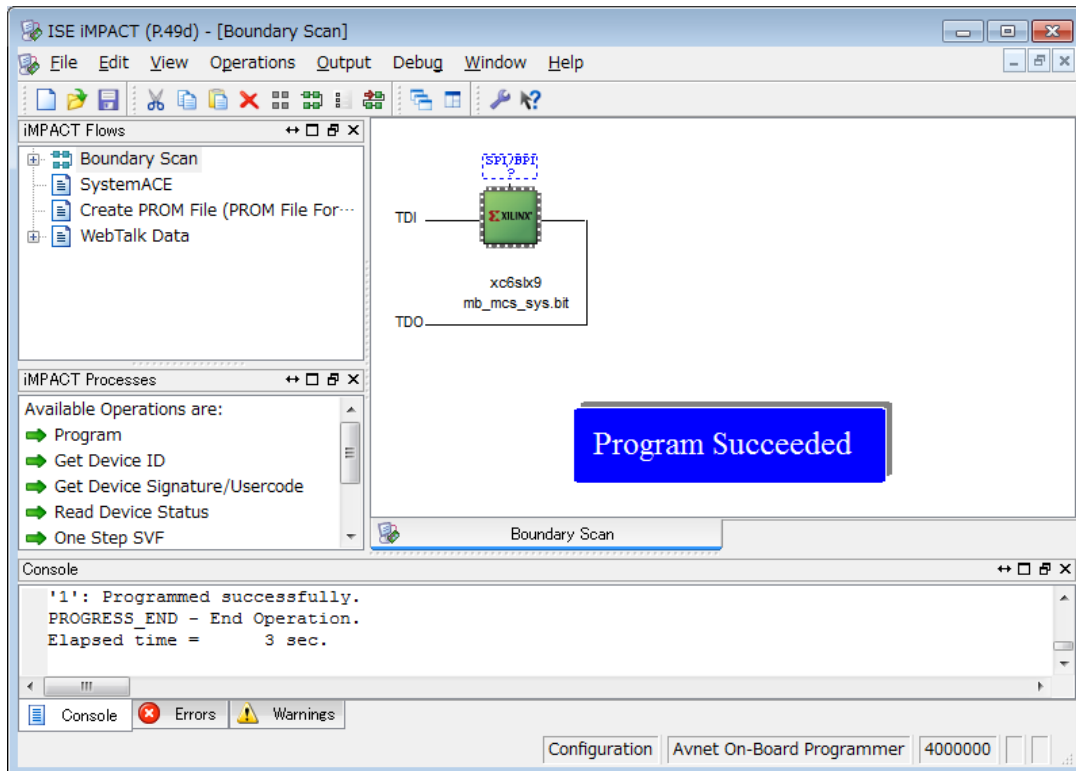


PROM データは使わないので No を選択



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

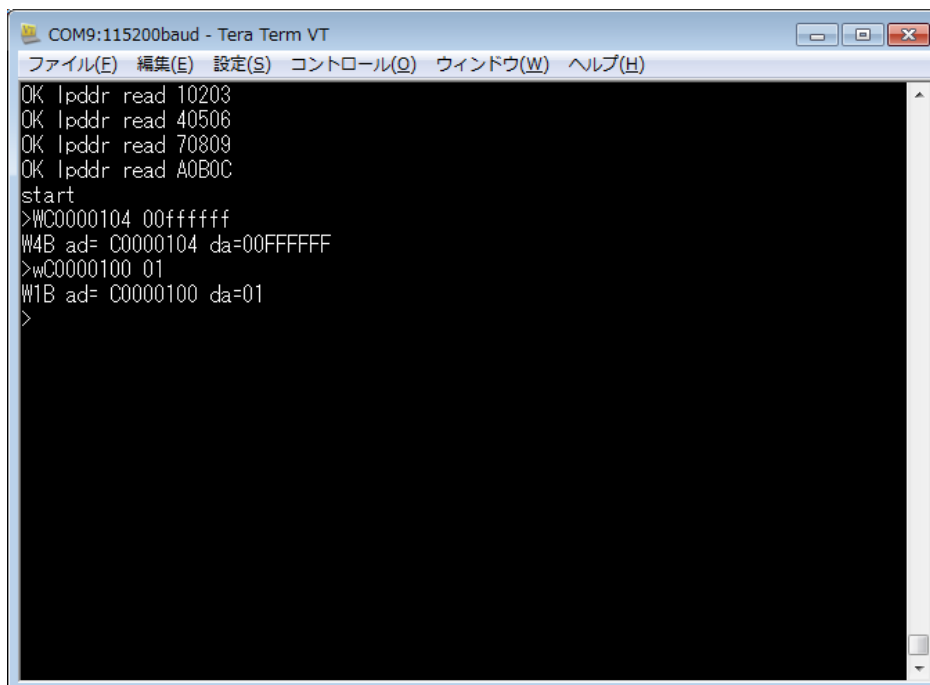




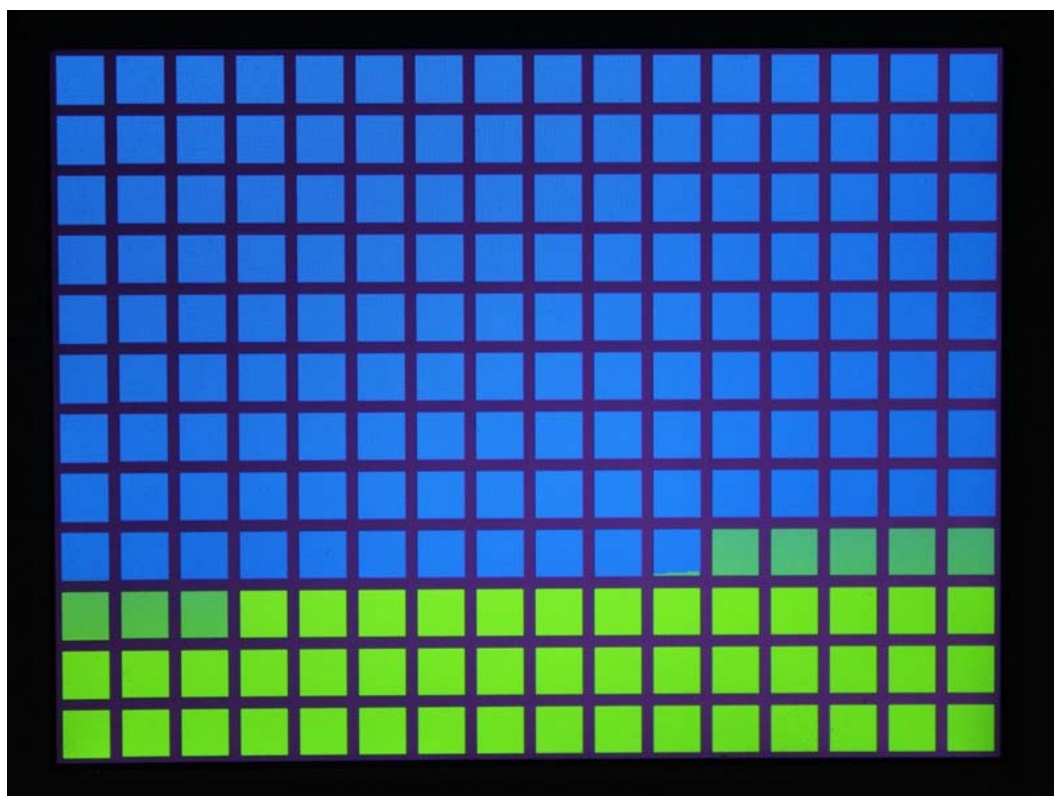
Program Succeeded と表示されればプログラミング完了、LX9 マイクロボードの動作を確認する。



ターミナルソフトの表示

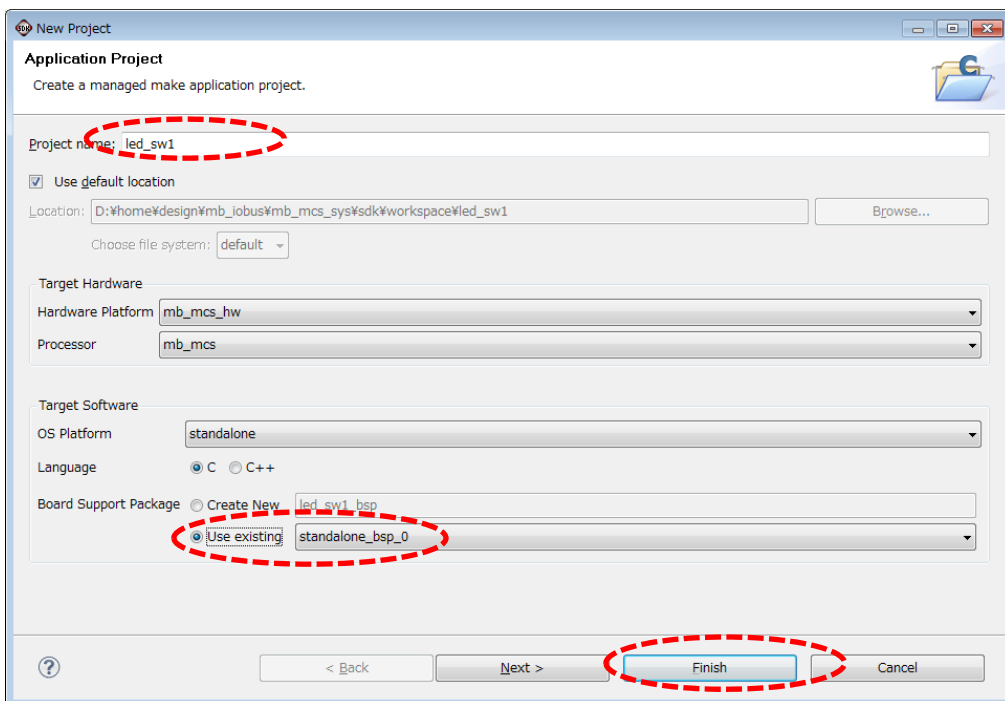
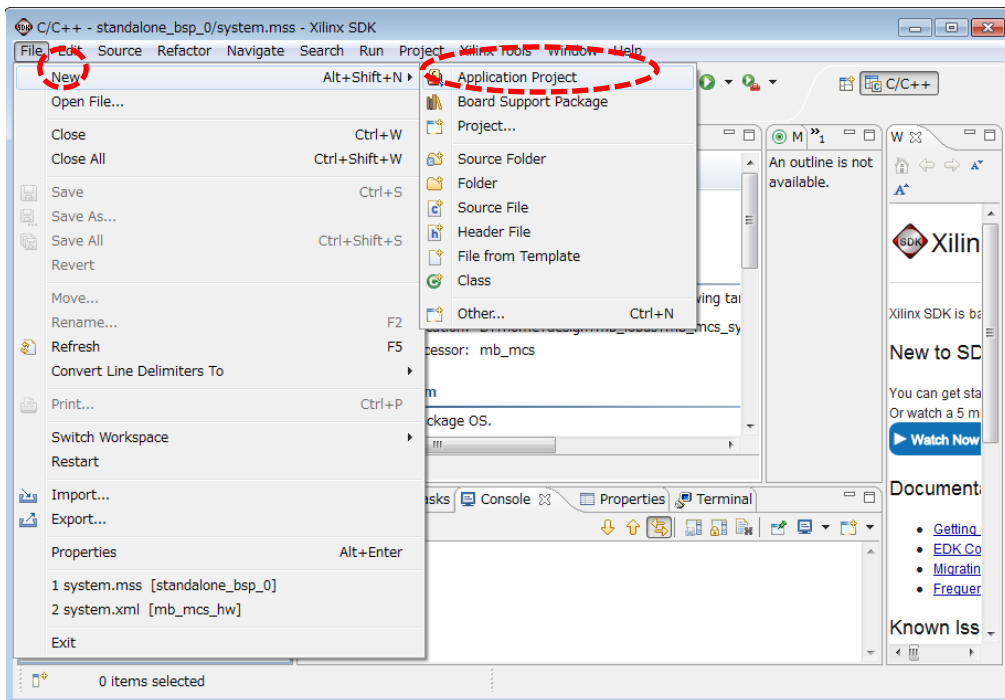


strat と表示後に、“WC0000104 00ffffff[リターン]” と入力するとディスプレイの表示が灰色から白に変わります。さらに “wC0000100 01[リターン]” と入力するとプログラムで作成したマス目模様がディスプレイに表示されます。



“wC0000100 01[リターン]” と入力後にディスプレイにマス目が表示されれば OK です。
説明は以上になります。

- ・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。

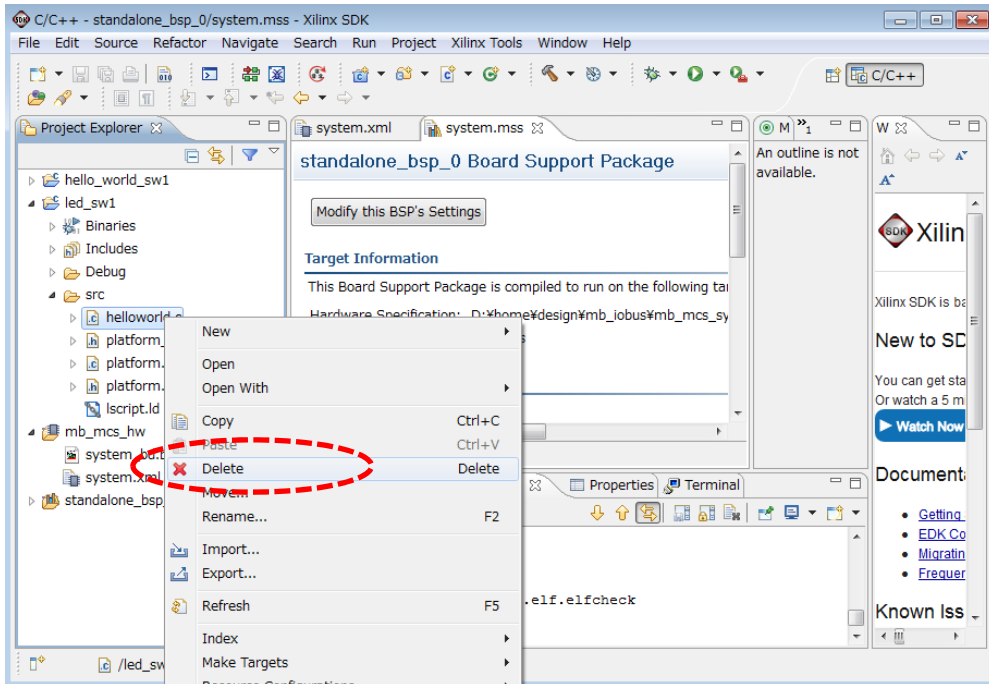


新しいソフトウェアプロジェクト名を設定(例.led_sw1)

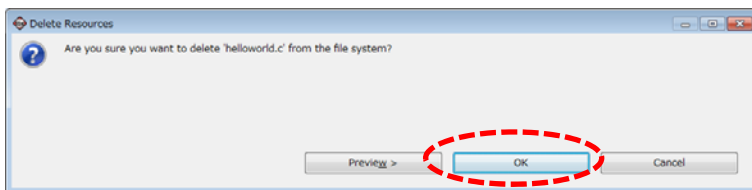
Board Support Package に standalone_bsp_0 を設定

Finsh をクリックする

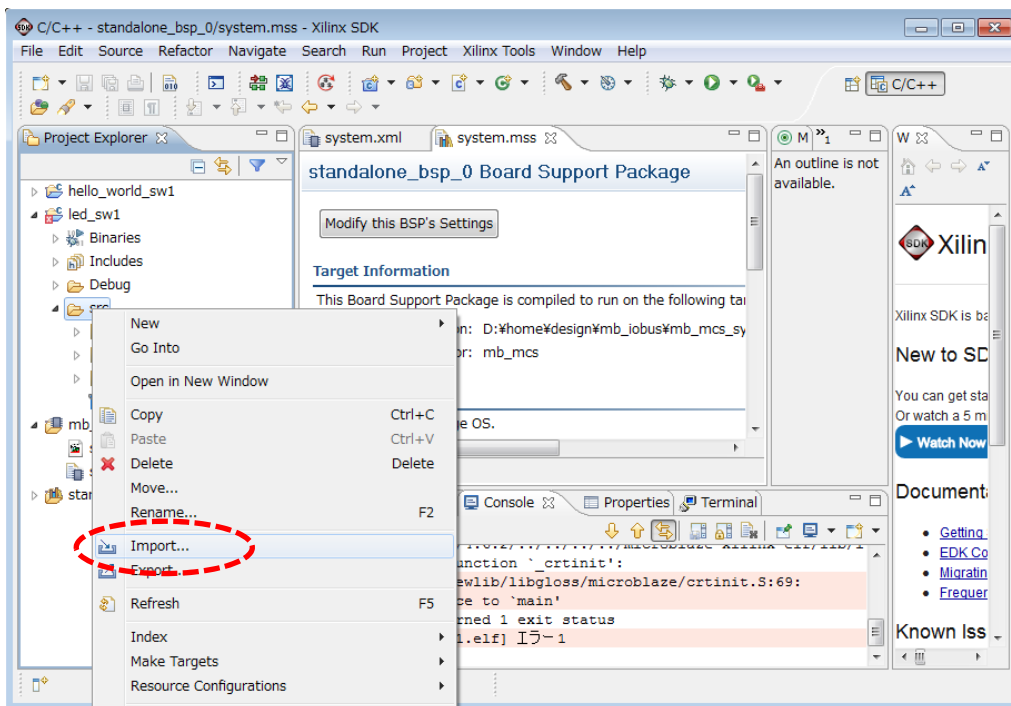




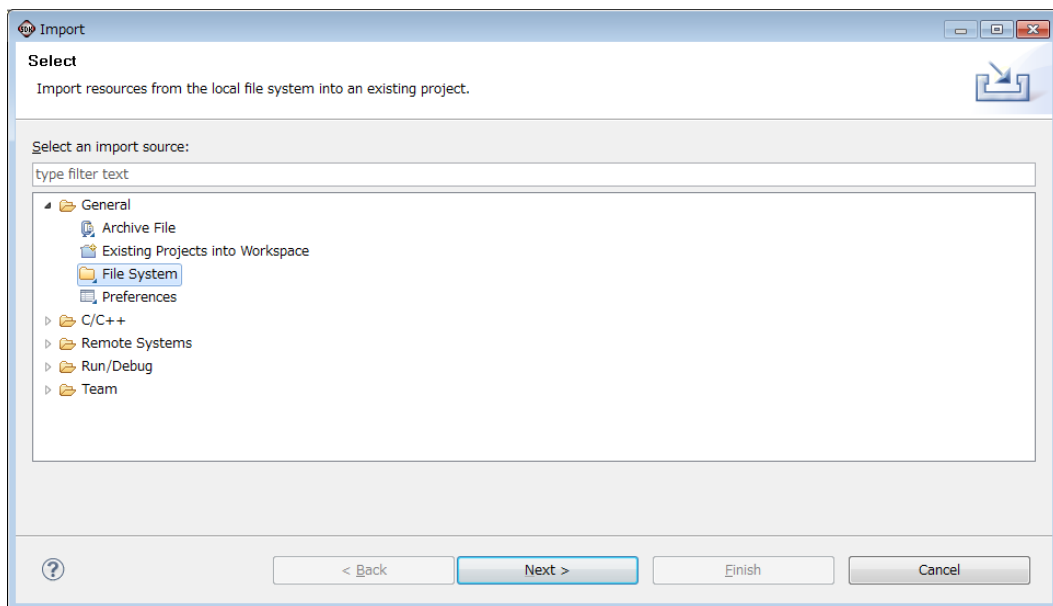
helloworld.c の上にカーソルを合わせ右ボタンで delete を選択して、削除



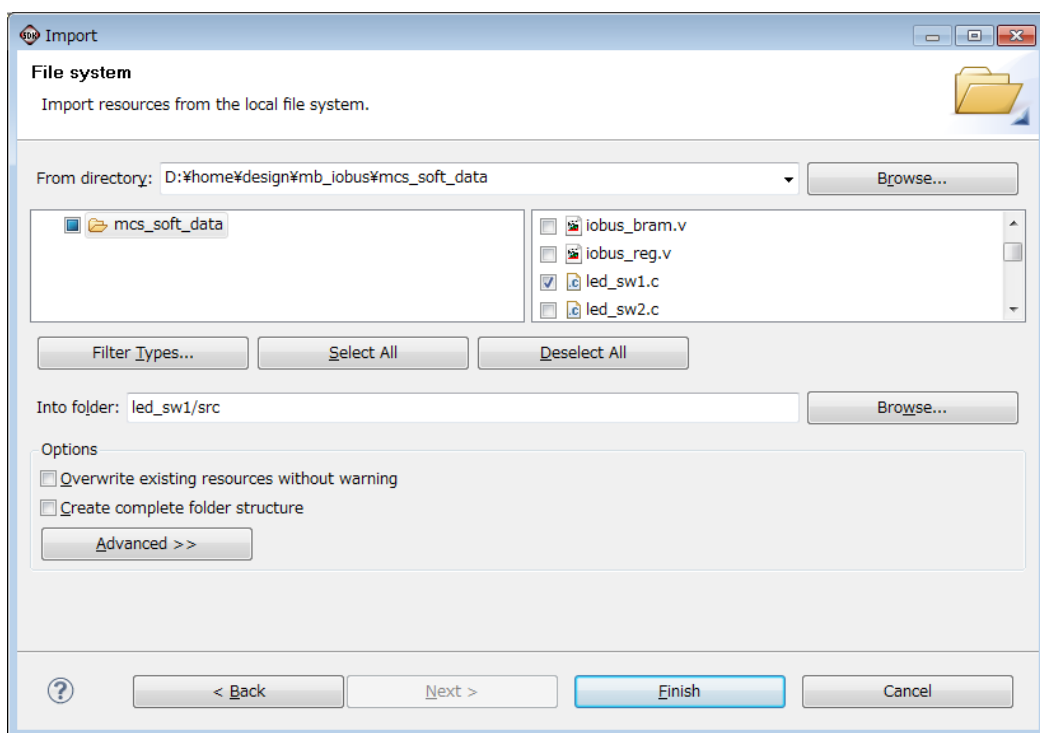
OK をクリック



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

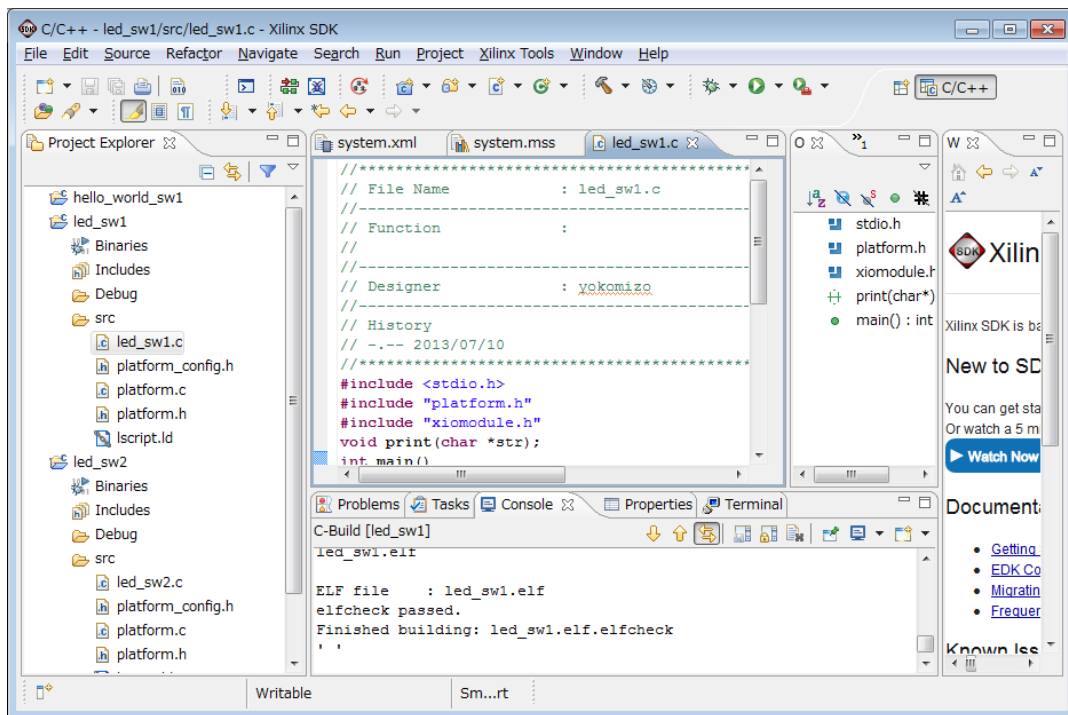


General→File System 選択

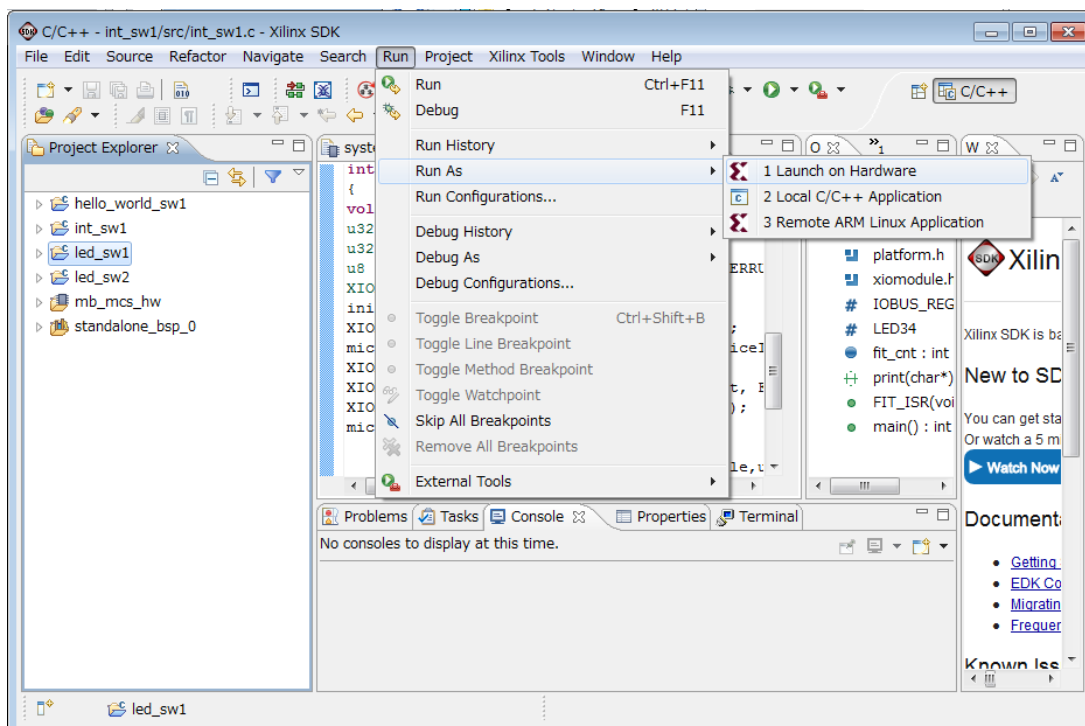


解凍データにある led_sw1.c を選択して、Finish





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。