

FPGA マガジン NO.5

OpenCpres VGA core の MicroBlazeMCS 用設計データの使い方 (2014/4/23)

ファイル一覧

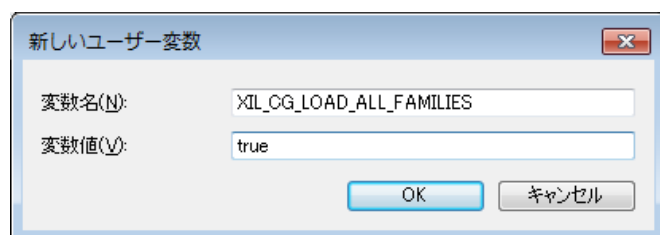
- ・ *.v Verilog-HDL の回路記述 (トップ回路は mbmcs_vga_ap.v)
- ・ mbmcs_vga_ap.ucf ピン配置指定
- ・ svga_sy_sw1.c サンプル C プログラム
- ・ mbmcs_vga_ap.bit bit ファイル、このファイルを FPGA へ書き込むことでも動作確認できます。

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

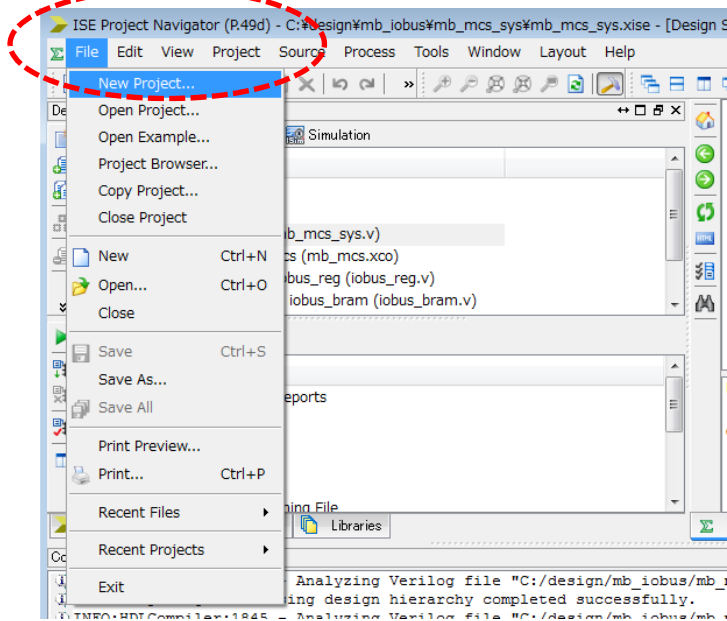
- ・ 設計用フォルダ mbmcs_vga_ap を作成する
- ・ 記事のダウンロードデータ vga_core_data_microboard の下にある ファイルを mbmcs_vga_ap へコピー
- ・ VGA core のデータを CopenCores のサイト (http://opencores.org/project,vga_lcd) からダウンロード
- ・ ダウンロードした vga_lcd_latest.tar.gz を解凍する
- ・ 解凍データの vga_lcd/trunk/rtl/verilog の下にある Verilog-HDL ソースを mbmcs_vga_ap へコピー
- ・ simple_gpio のデータを CopenCores のサイト (http://opencores.org/project,simple_gpio) からダウンロード
- ・ ダウンロードした simple_gpio_latest.tar.gz を解凍する
- ・ 解凍データの simple_gpio/trunk/rtl の下にある Verilog-HDL ソースを mbmcs_vag_ap 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



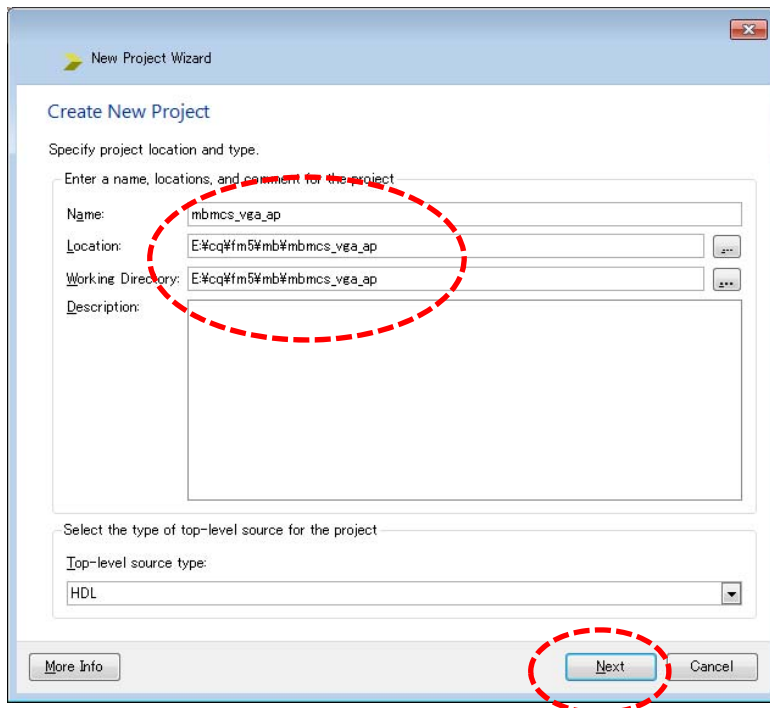
変数名 : XIL_CG_LOAD_ALL_FAMILIES 値 : true を入力して OK

- ・ ProjectNavigator を起動します。
スタートメニューから「Xilinx Design Tools」→「ISE Design Suite 14.7」→「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



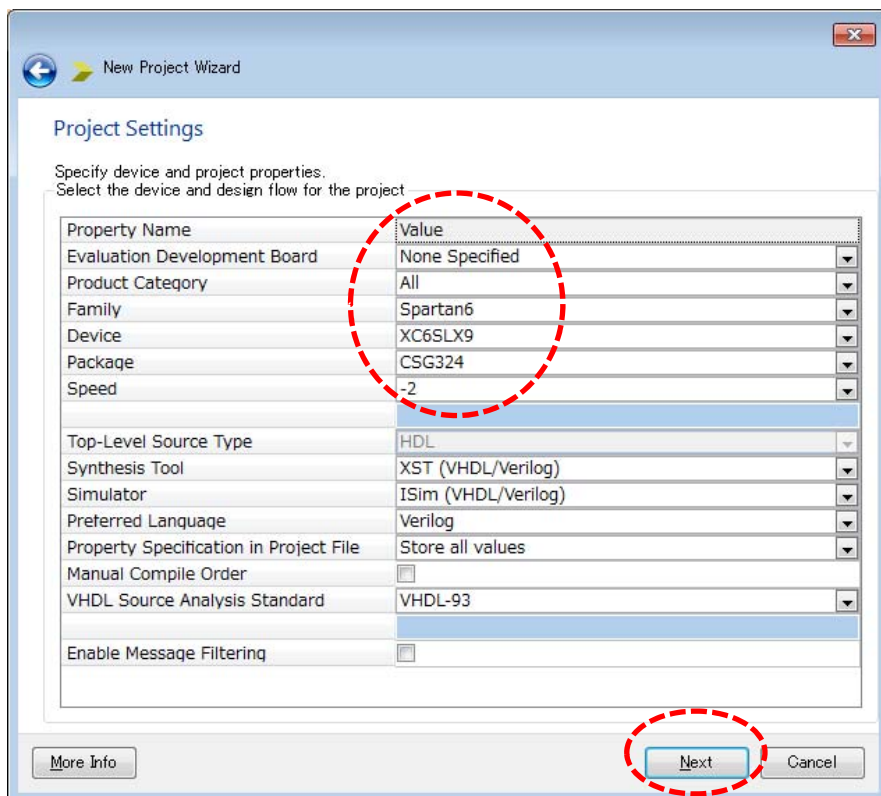
プロジェクト名と設計フォルダ指定

プロジェクト名 : mbmcs_vga_ap

設計フォルダ : 任意/mbmcs_vga_ap

次に Next をクリック

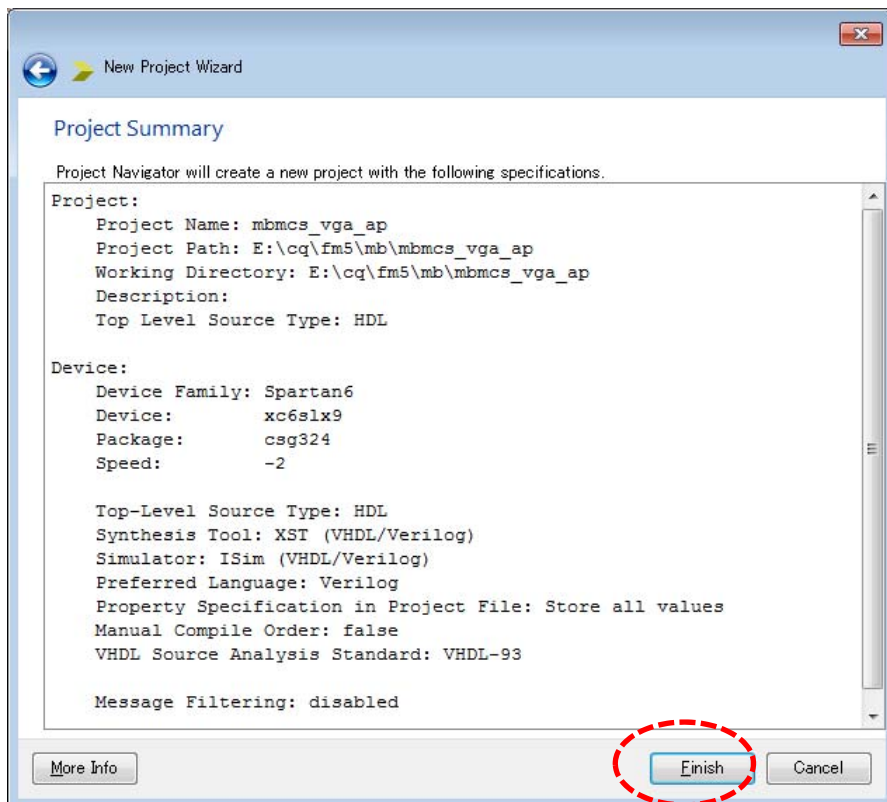




The 'New Project Wizard' dialog box is shown at the 'Project Settings' step. It contains a table of properties and their values. A red dashed circle highlights the 'Device' section, which includes 'Family' (Spartan6), 'Device' (XC6SLX9), and 'Package' (CSG324). Another red dashed circle highlights the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

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: mbmcs_vga_ap
- Project Path: E:\cq\fm5\mb\mbmcs_vga_ap
- Working Directory: E:\cq\fm5\mb\mbmcs_vga_ap
- 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

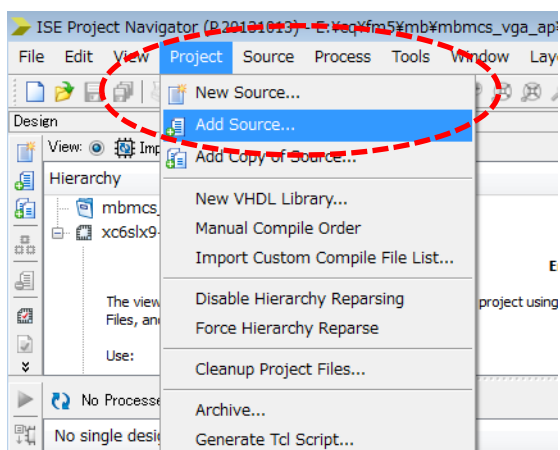
Manual Compile Order: false

VHDL Source Analysis Standard: VHDL-93

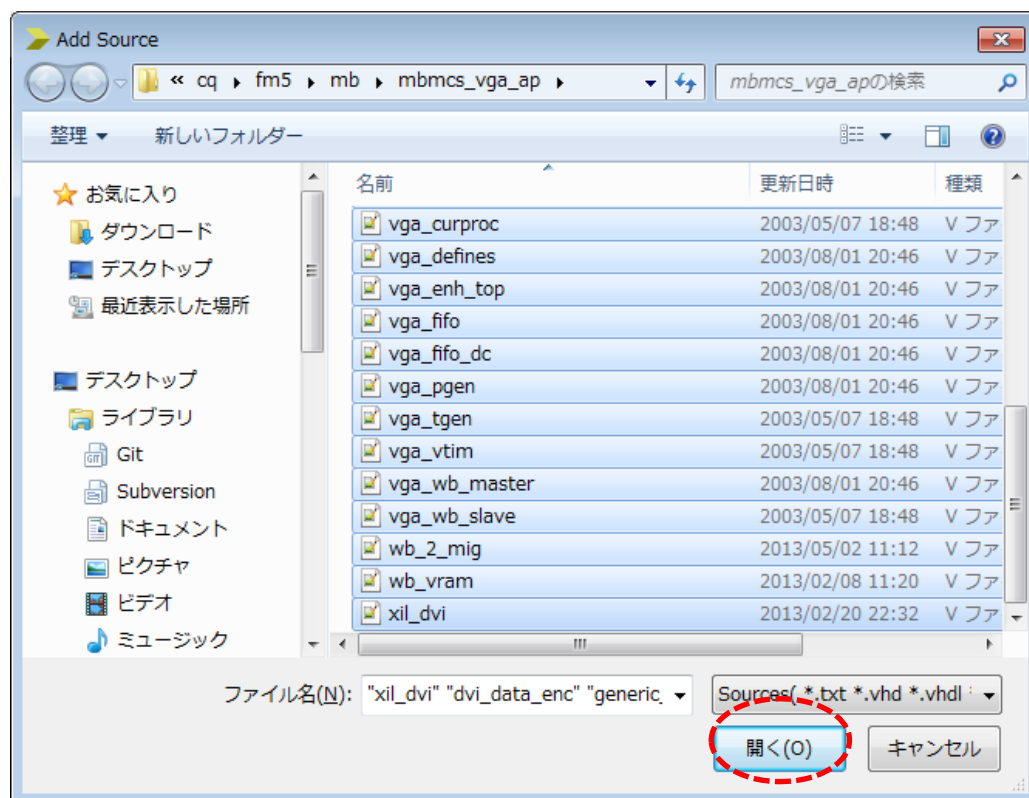
Message Filtering: disabled

Finish をクリック



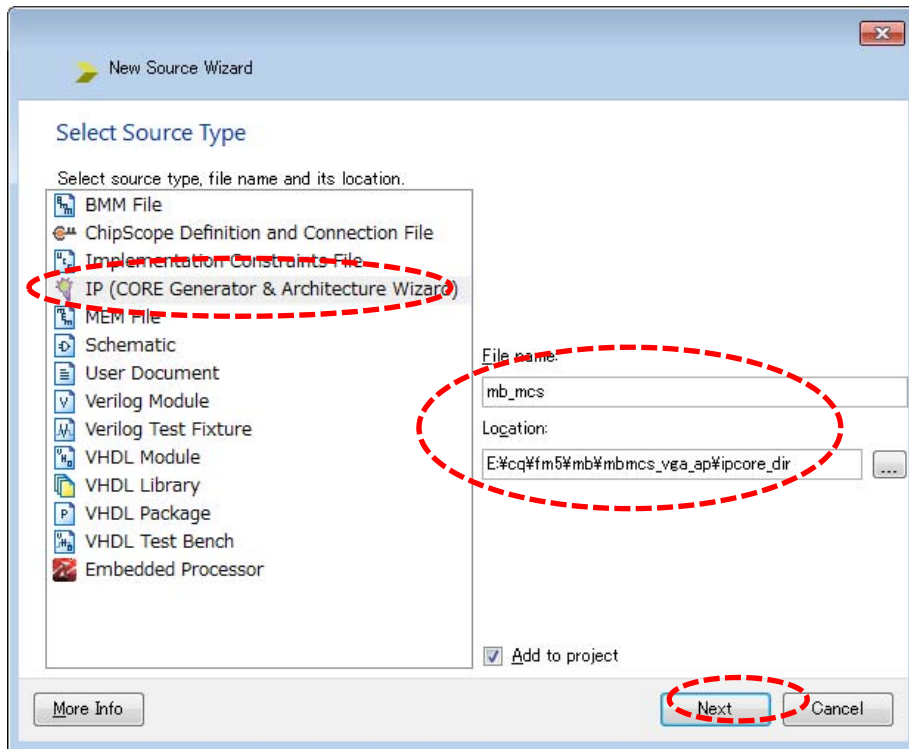


設計データを追加します。Project→Add Source を選択

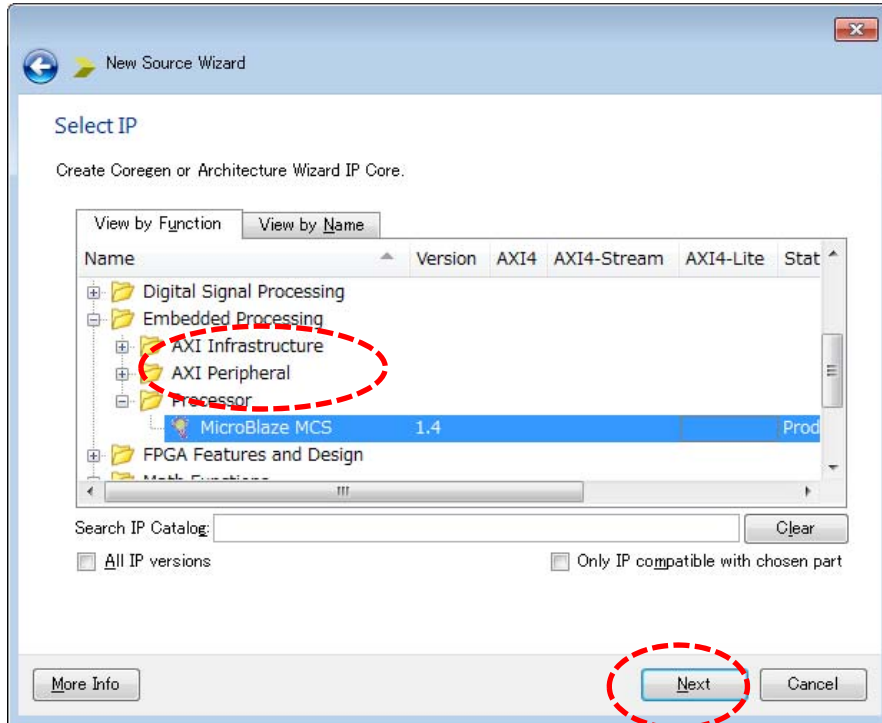


ファイルの指定、mb_mcs_sys の下にある Verilog-HDL ファイルと UCF ファイルを指定



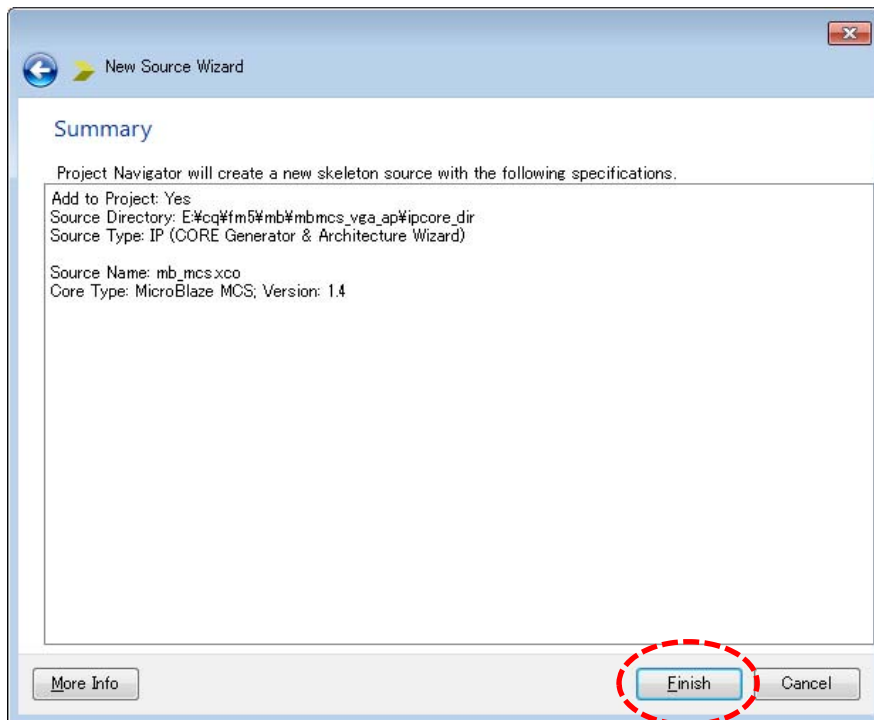


IP (CORE Gener...をクリックして選択、ファイル名を mb_mcs に指定、Next をクリック

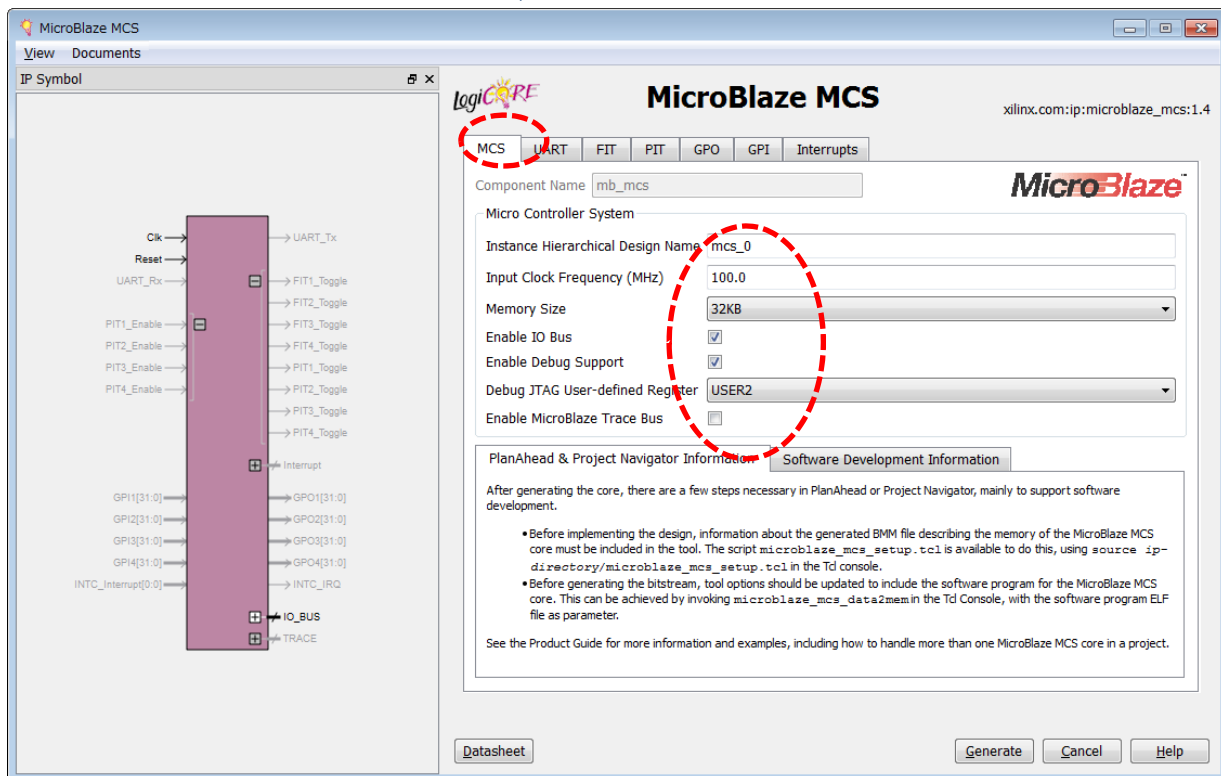


MicroBlaze MCS 選択して Next をクリック



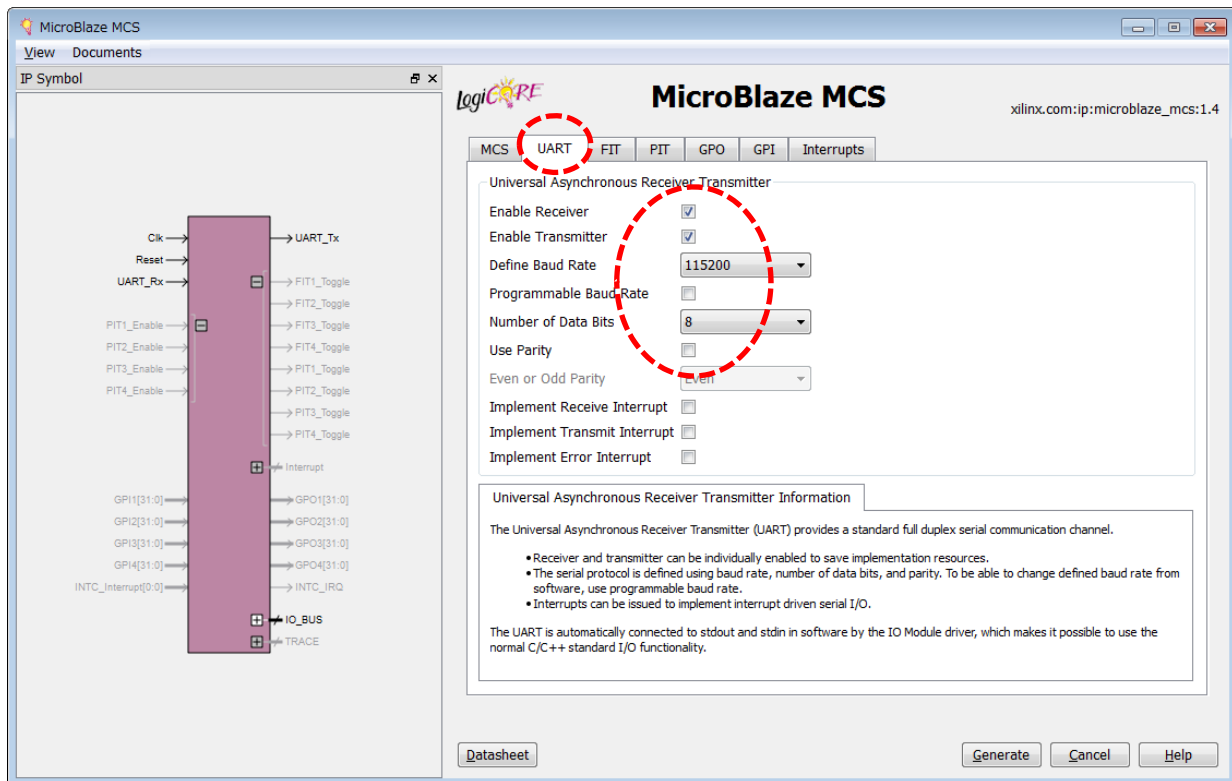


Finish をクリックで CORE generator が起動

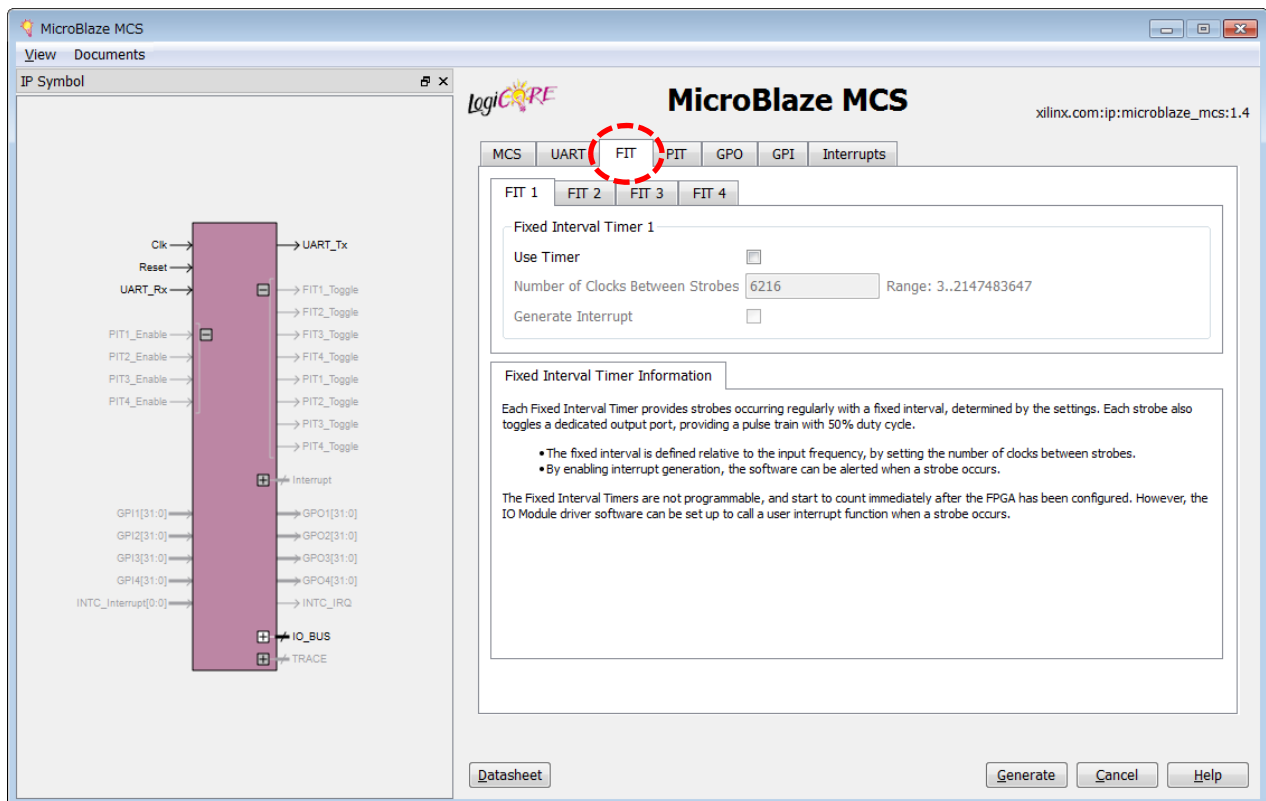


MicroBlaze MCS の基本設定



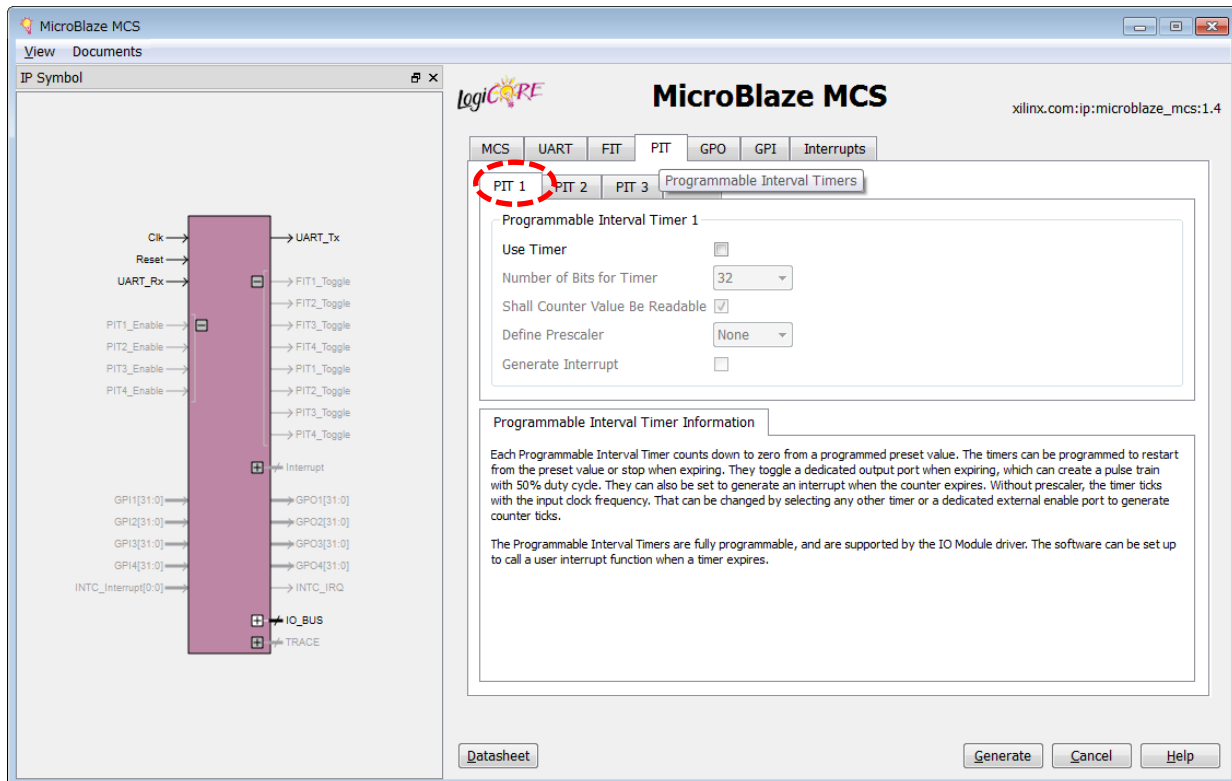


UART の設定

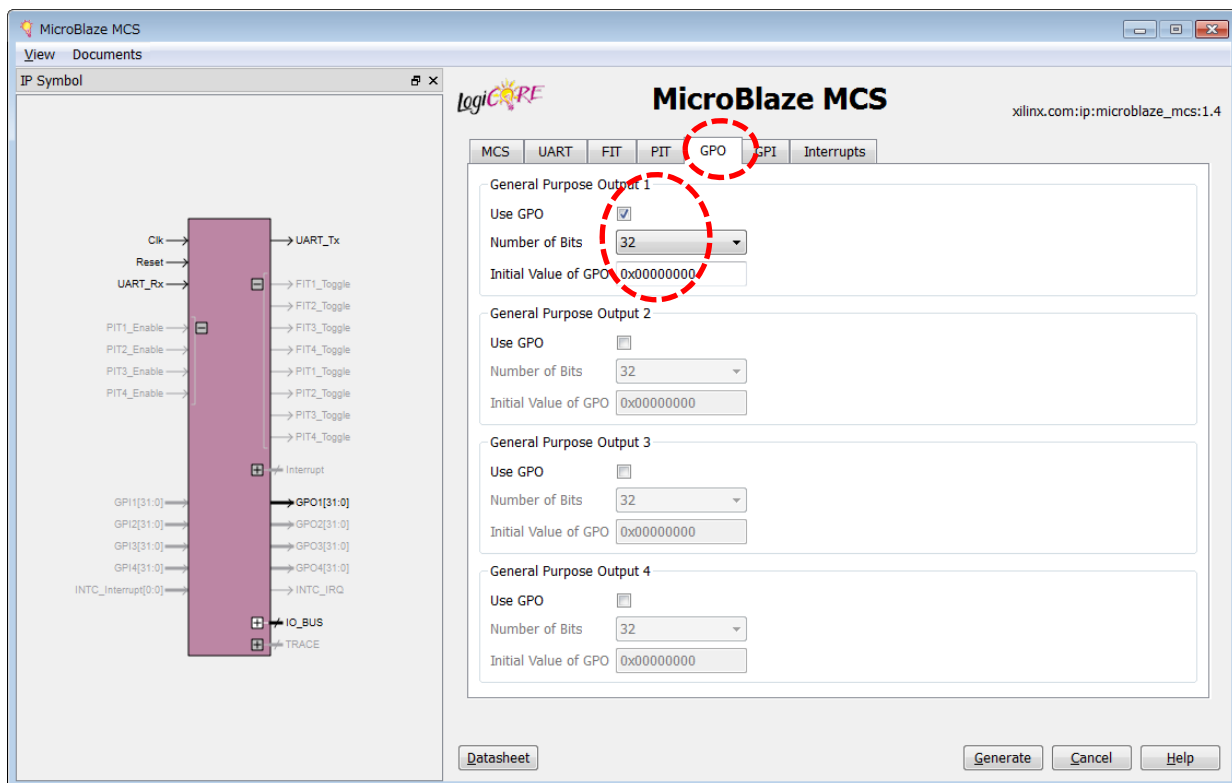


FIT の指定、何も設定しない



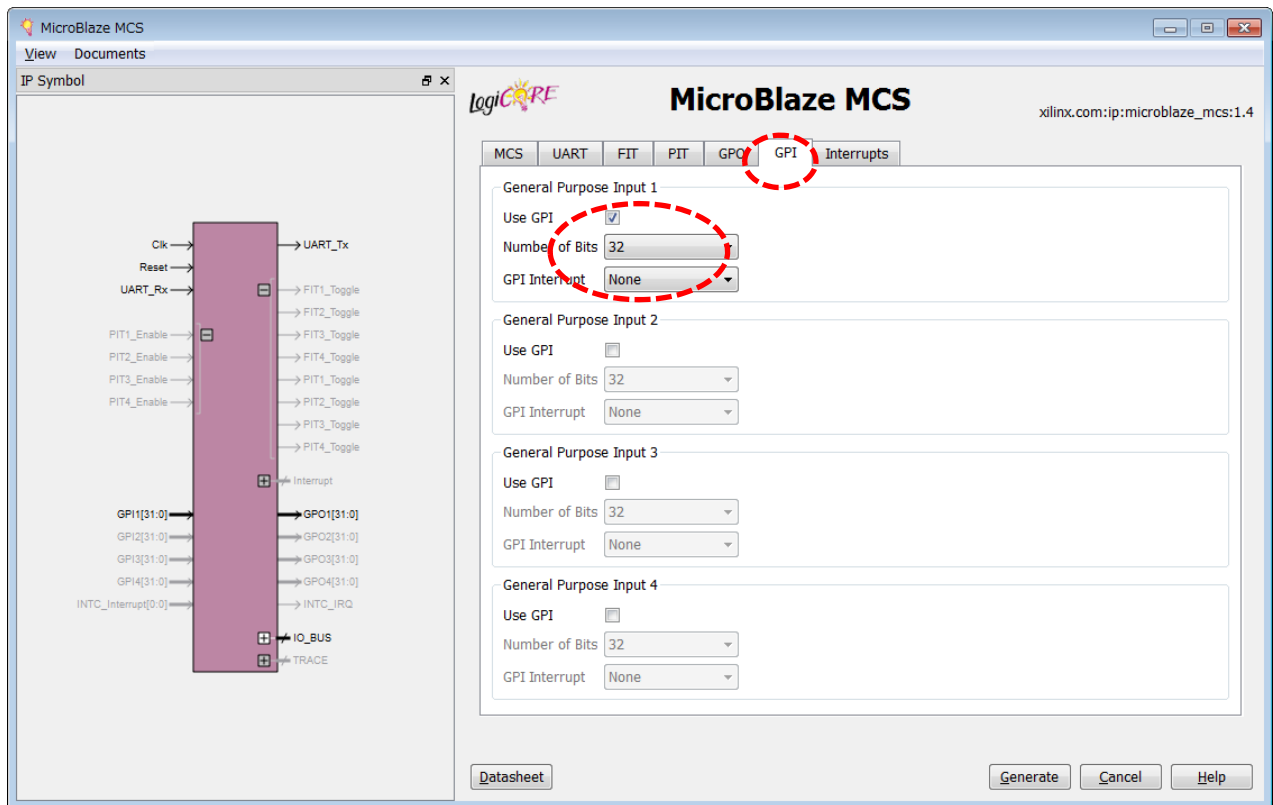


PIT の指定、何も設定しない

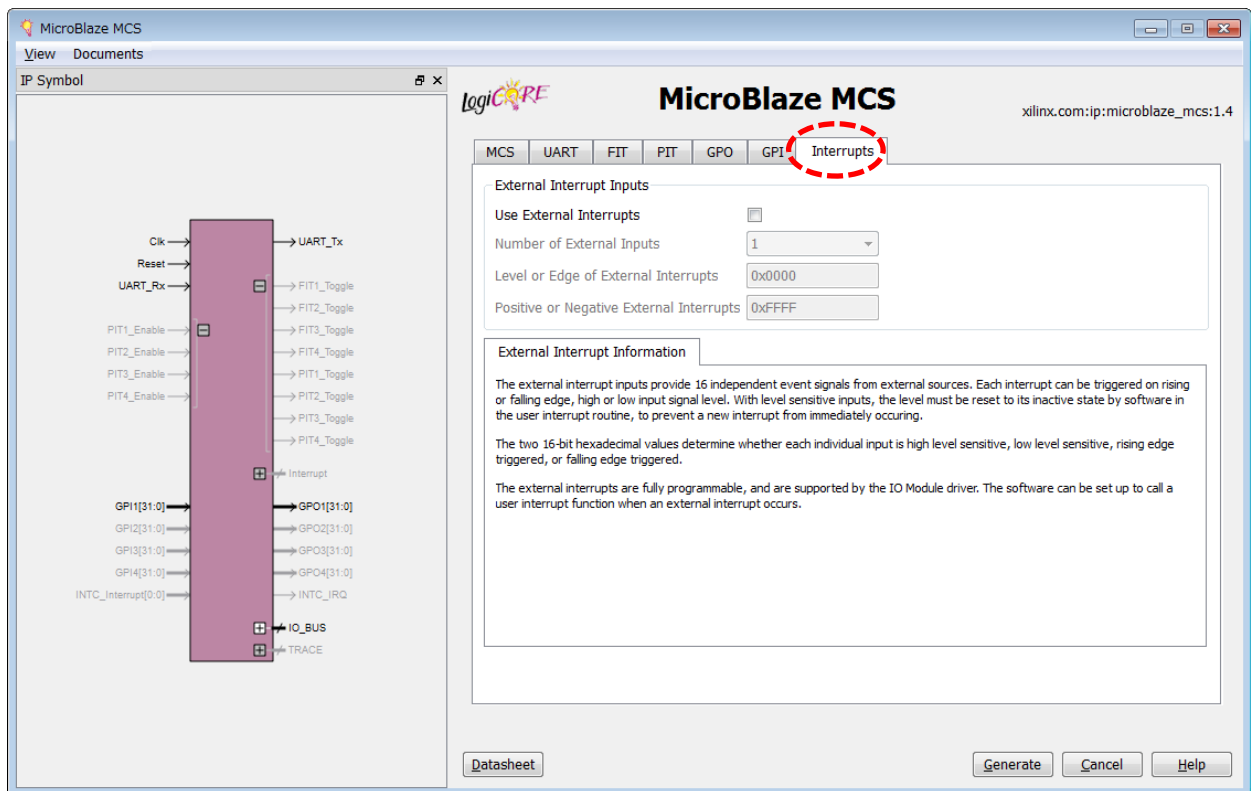


GPO の設定



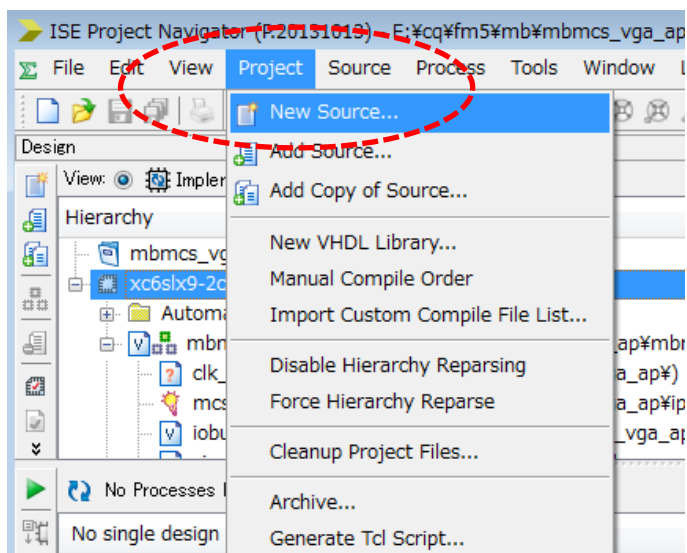


GPI の指定

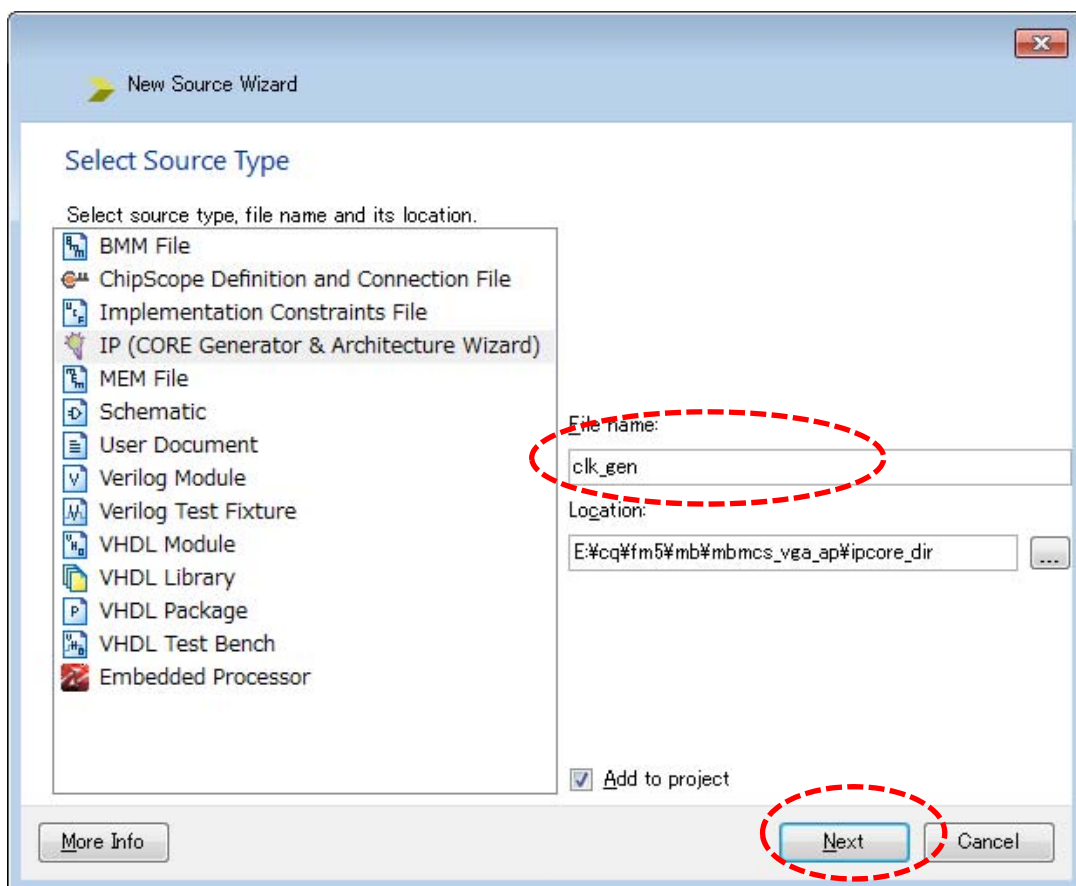


外部割り込みは設定なしで、Generate をクリックすると MicroBlaze MCS がプロジェクトに追加される





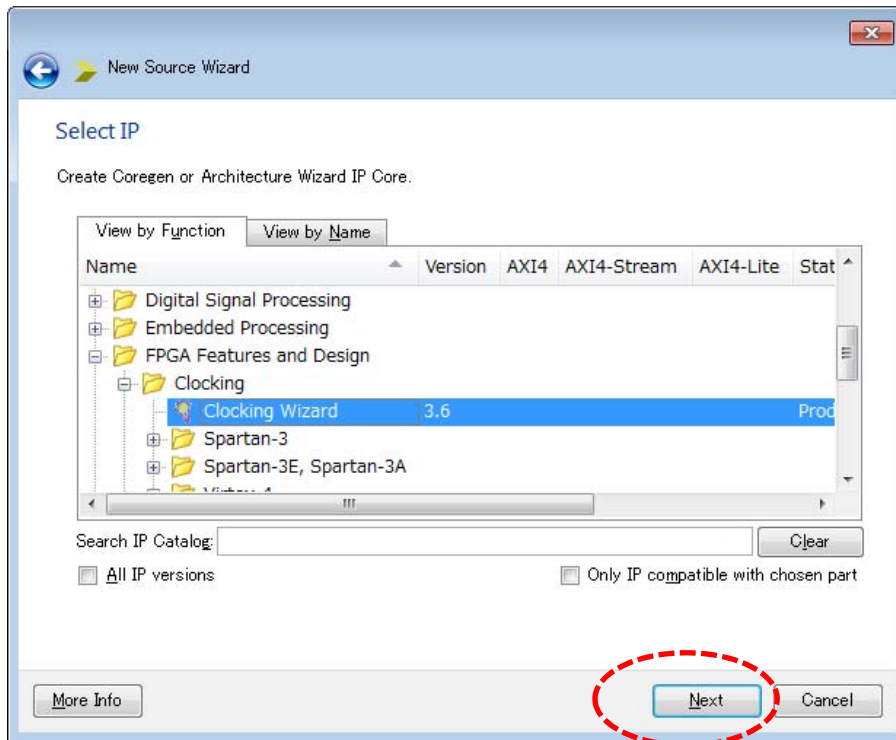
ブロック RAM の作成、Project→New Source を選択



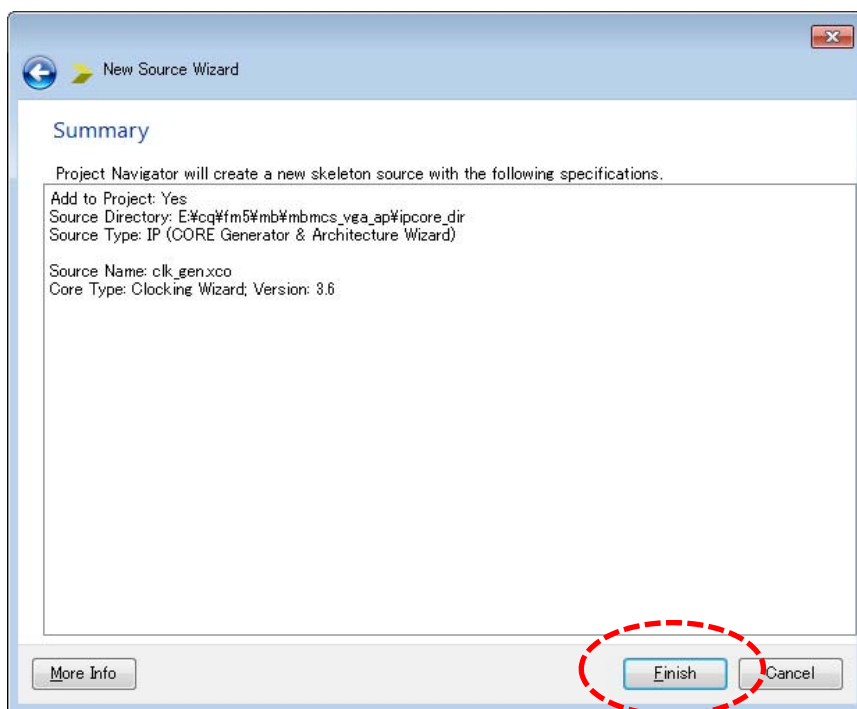
clk_gen の作成、Project→New Source を選択

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



Clocking Wizard

Component name: clk_gen

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)

Input Clock Information

Input Clock	Input freq (MHz)		Input Jitter	Source
	Value	Valid range		
Primary	100.000	5.000 - 500.000	0.010	Single ended clock capable pin

The IP Symbol shown on the left describes the input and output ports provided from the clocking network. Note that the names of these ports do not necessarily match the pins of any specific primitive.

Next >

入力クロック周波数を 100MHz に設定、Next をクリック



Clocking Wizard

The phase is calculated relative to the active input clock.

Output Clock Settings

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	266.000	266.667	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"/>

Generate

CLK_OUT1 : 100MHz

CLK_OUT2 : 40MHz

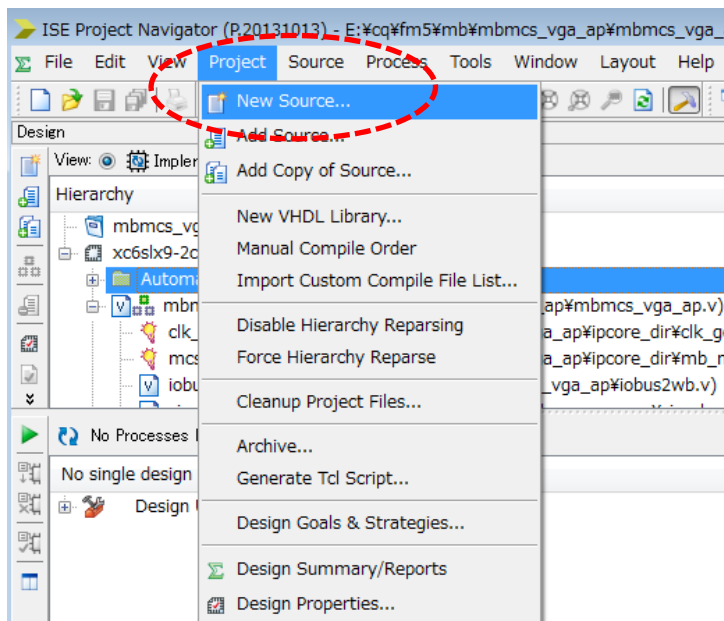
CLK_OUT3 : 200MHz

CLK_OUT4 : 80MHz

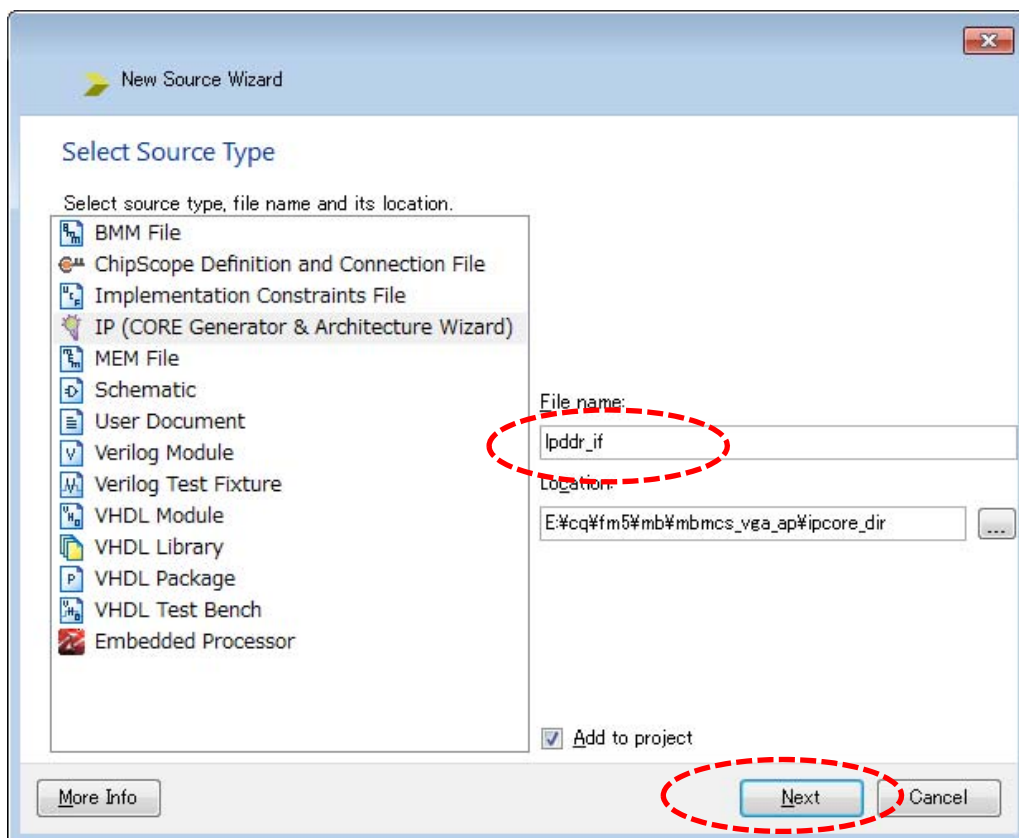
CLK_OUT5 : 200MHz [NoBuffer]

に指定して、Generate をクリック



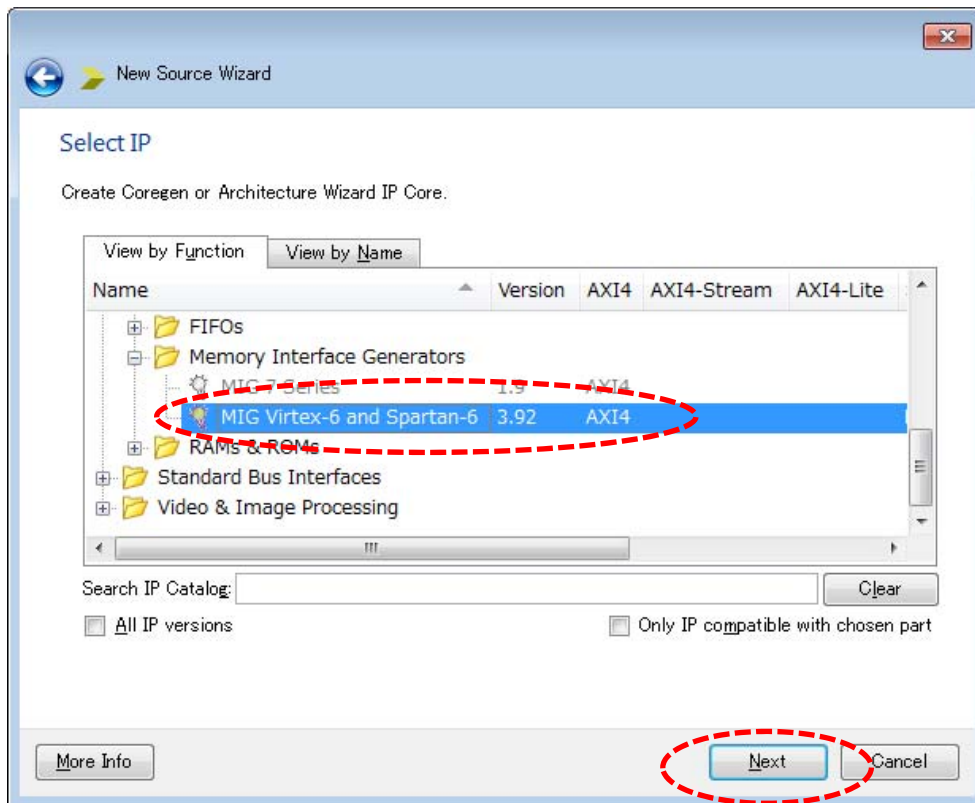


ISE Project Navigator→Project→New Source で新しいソースを追加する

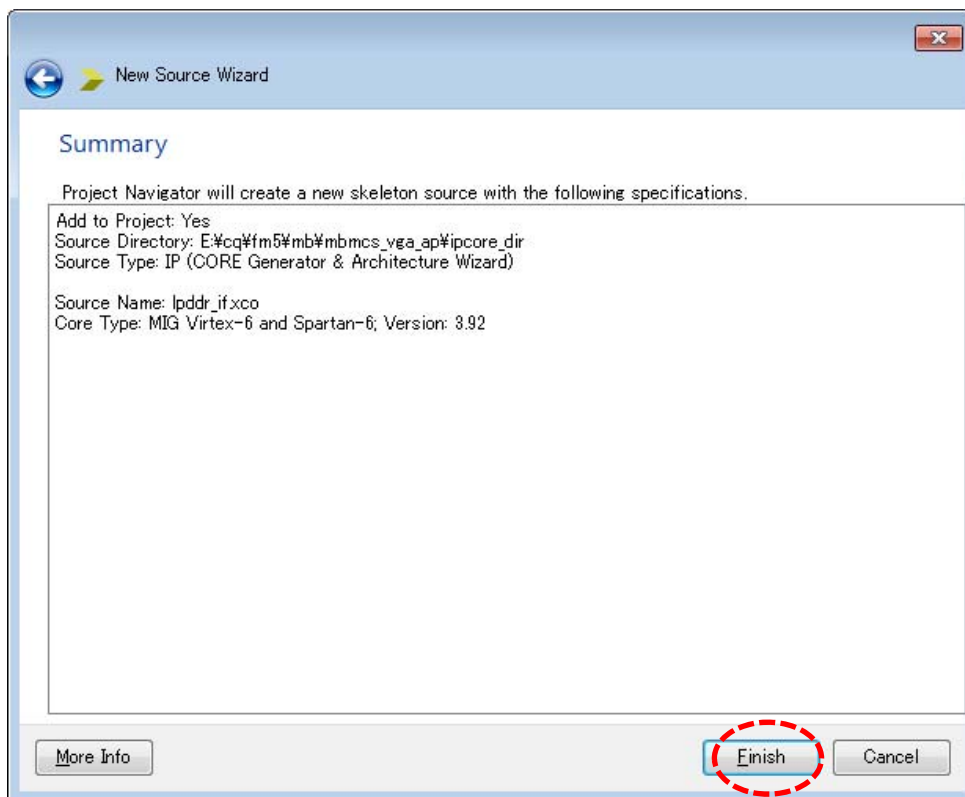


ファイル名に lpddr_if を入力して、IP (COREGenerator) を選択して、Next をクリック
作成するコンポーネント名は lpddr_if になる



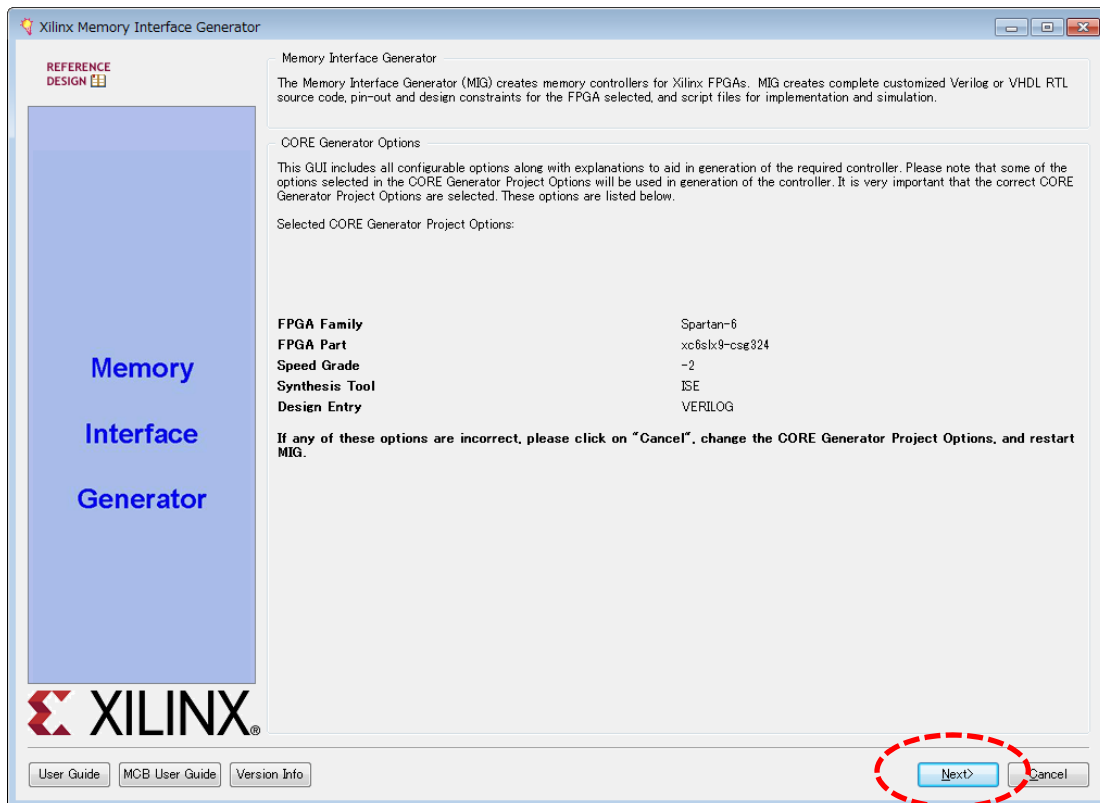


MIG Virtex-6 and Spartan-6 を選択して、Next をクリック

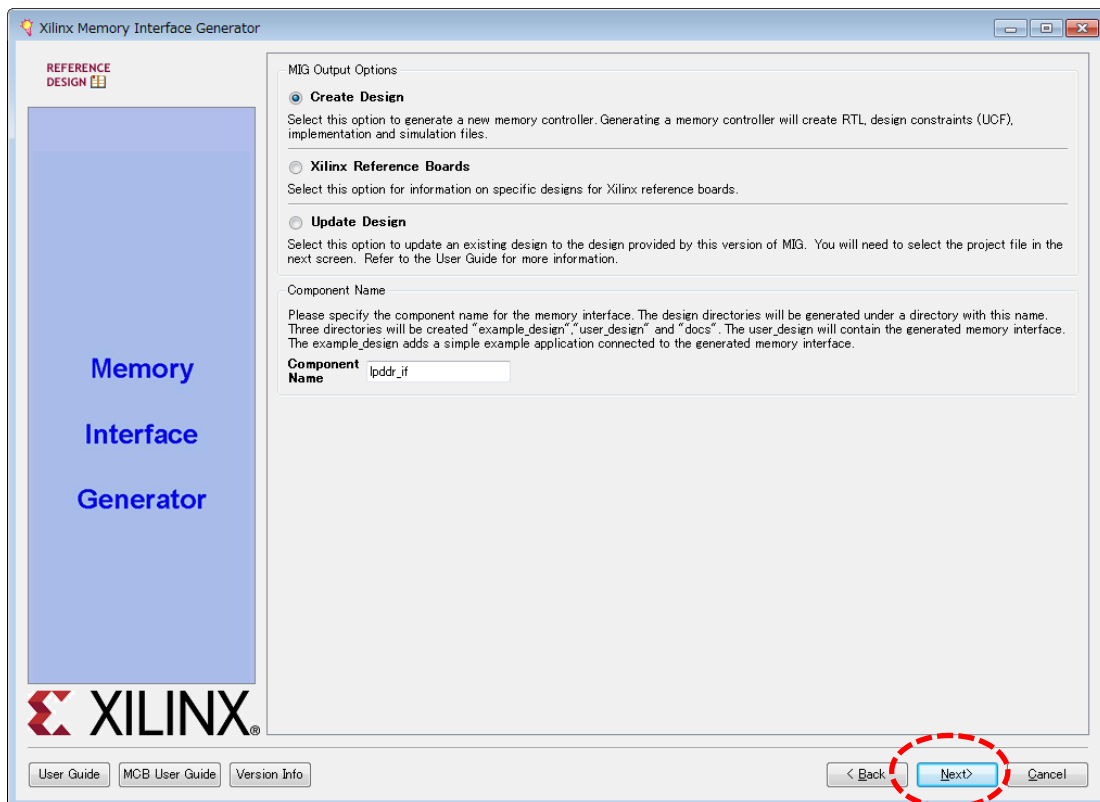


Finshi をクリック



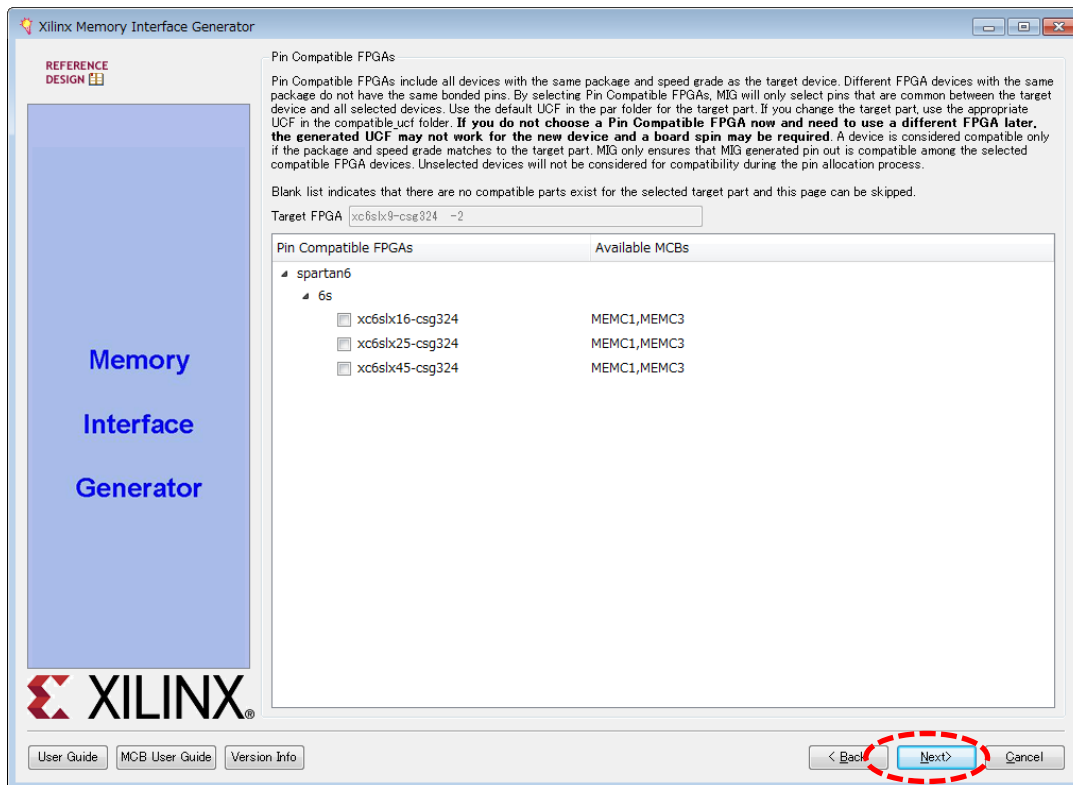


Next をクリック

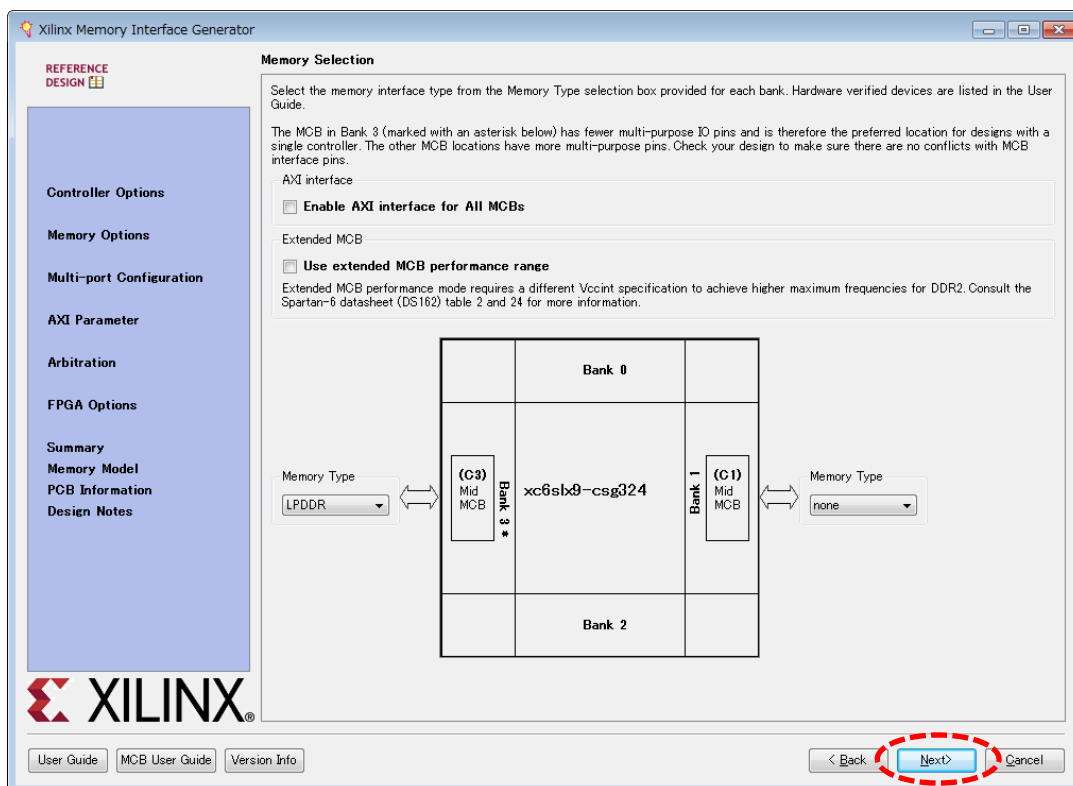


変更せずに Next をクリック



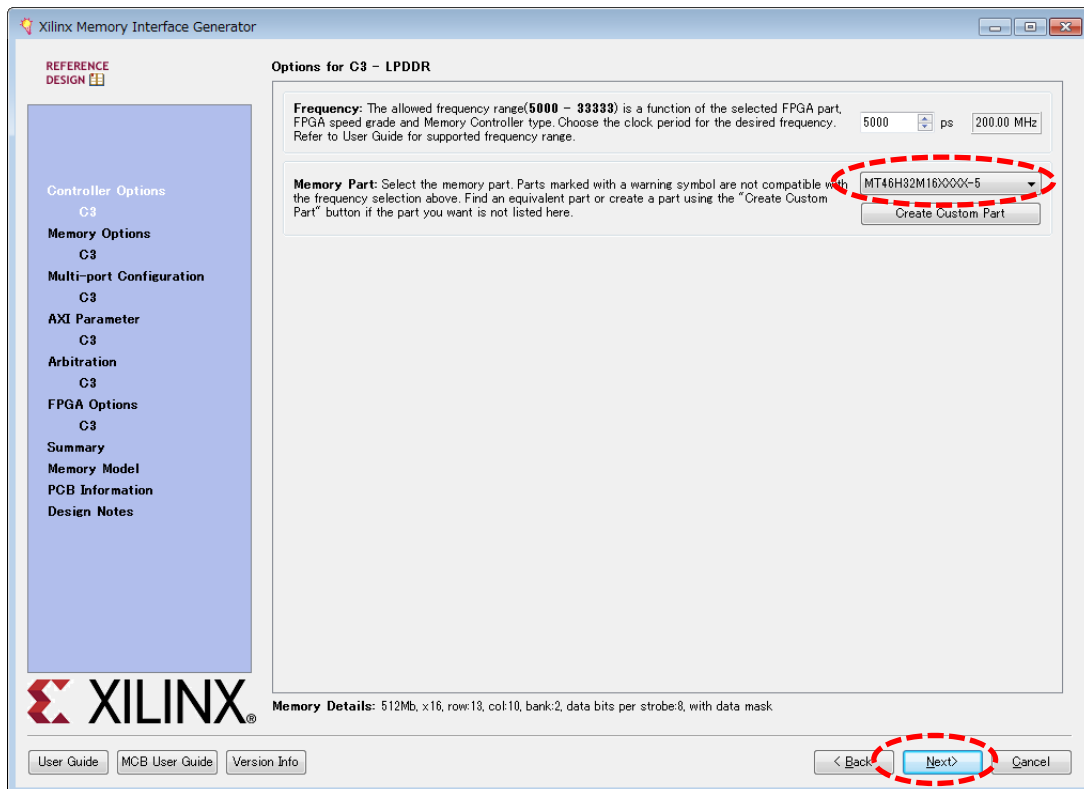


変更せずに Next をクリック

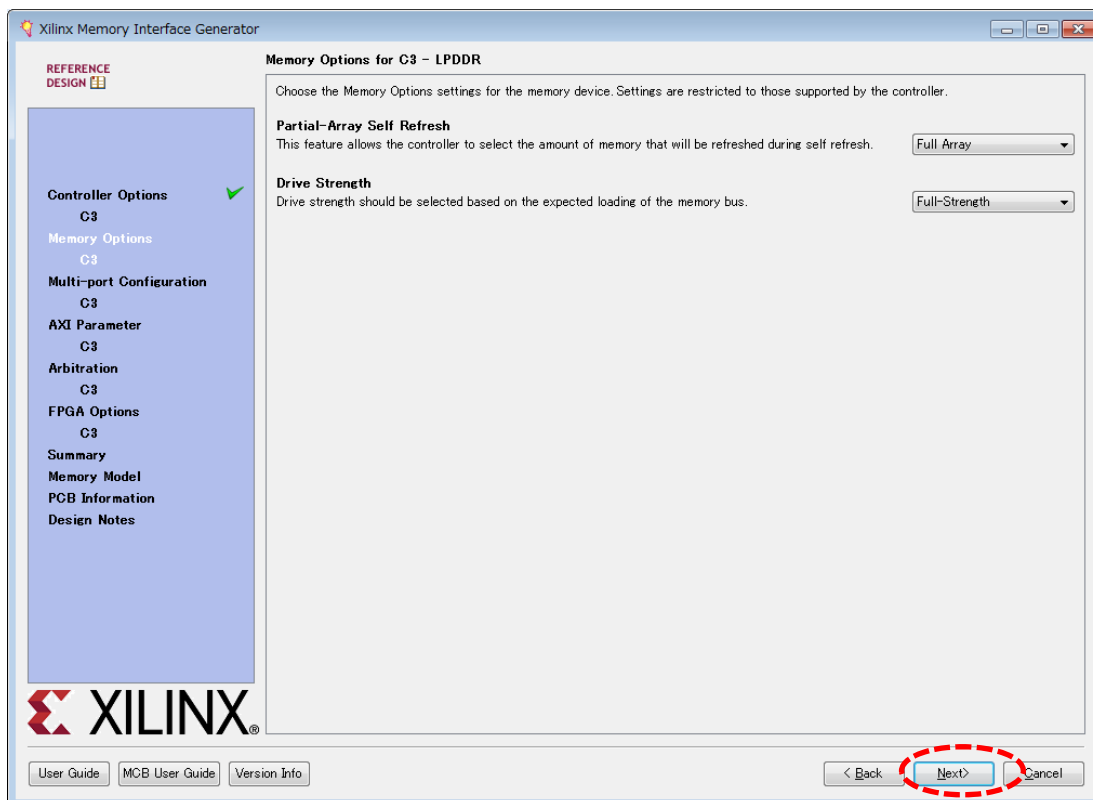


bank3 を LPDDR を指定



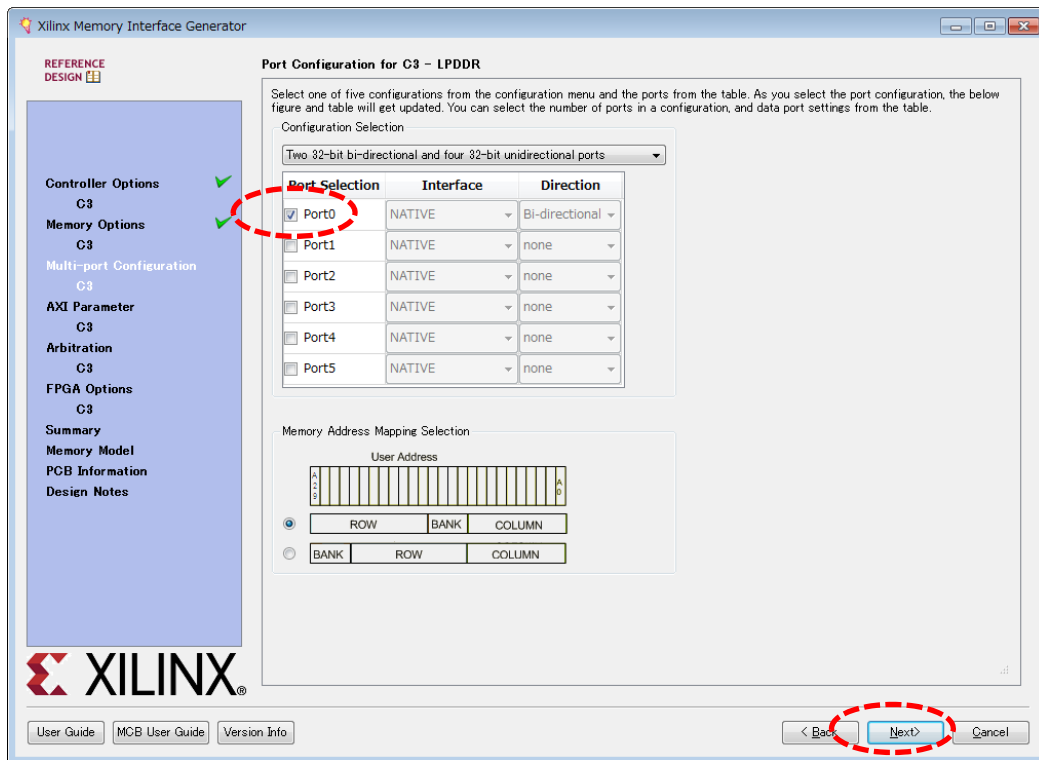


MicroBoard の実装メモリに合わせて MT46H32M16XXX-5 を選択する

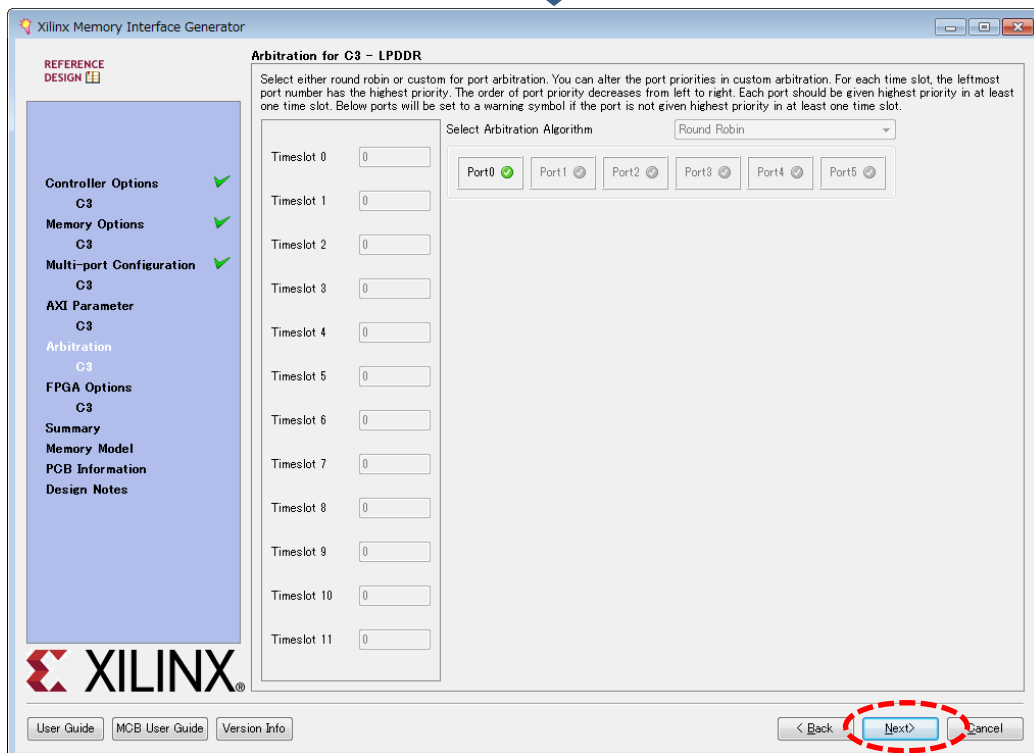


変更せずに Next をクリック



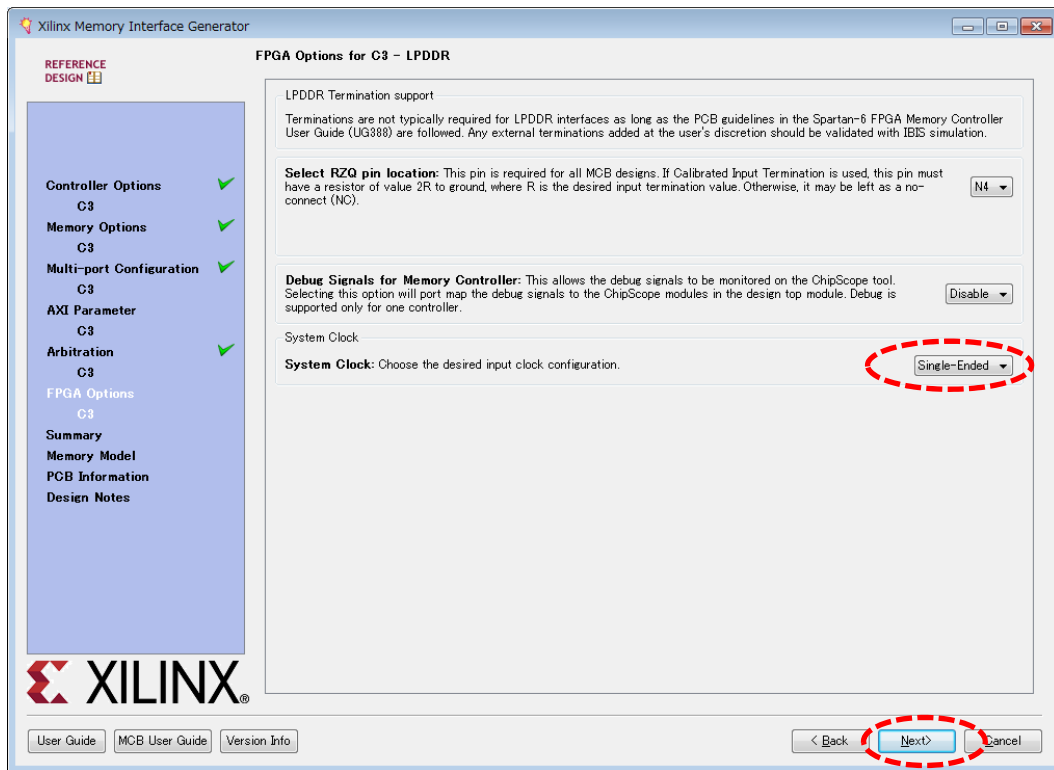


port0 を使用に設定

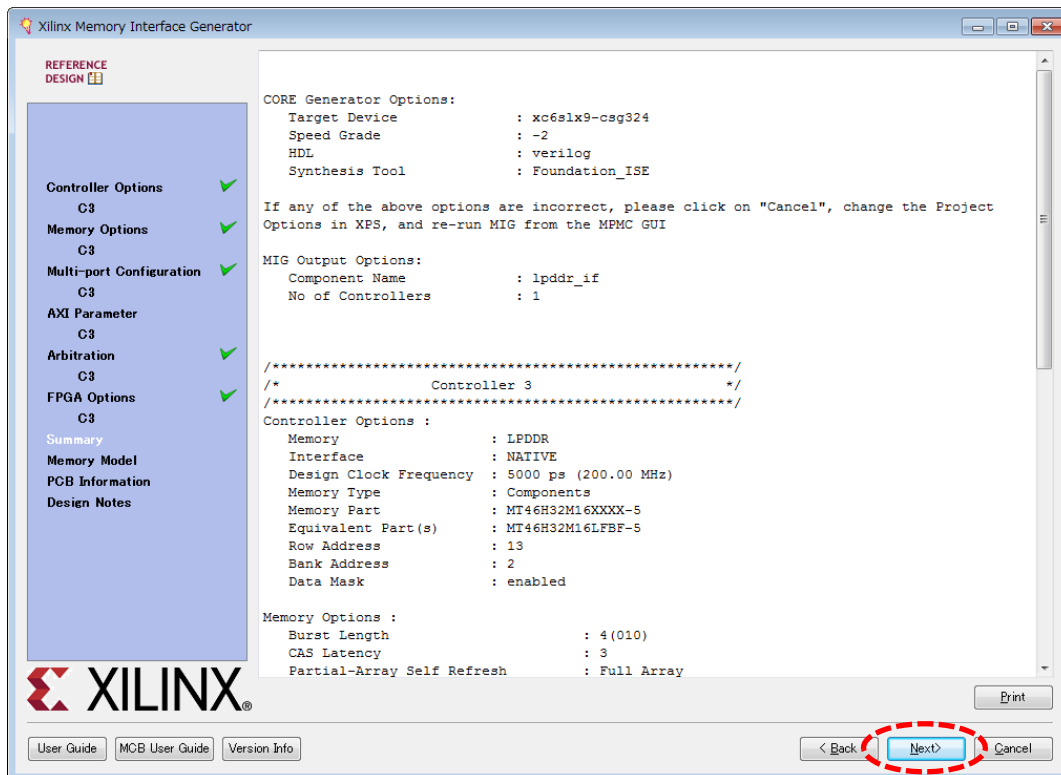


アクセスのアービトレーションの選択、変更せずに Next をクリック



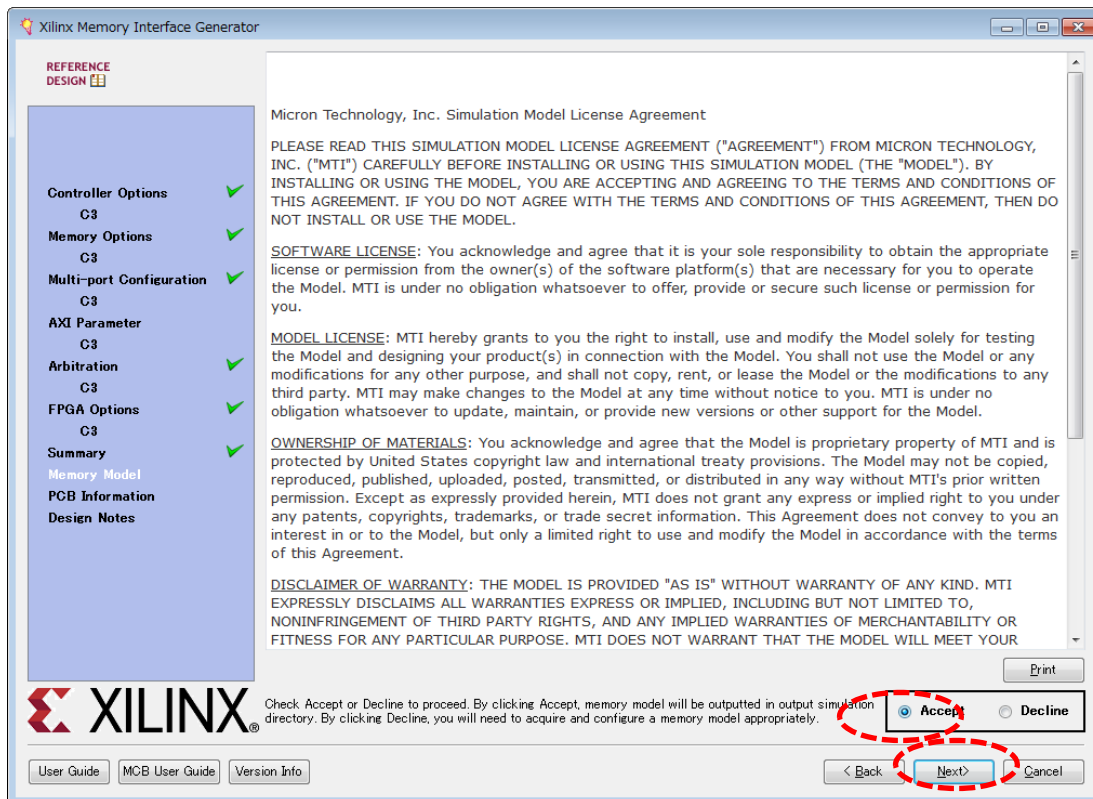


System Clock は Single-Ended を選択



作成時のサマリーが表示される。Next をクリック



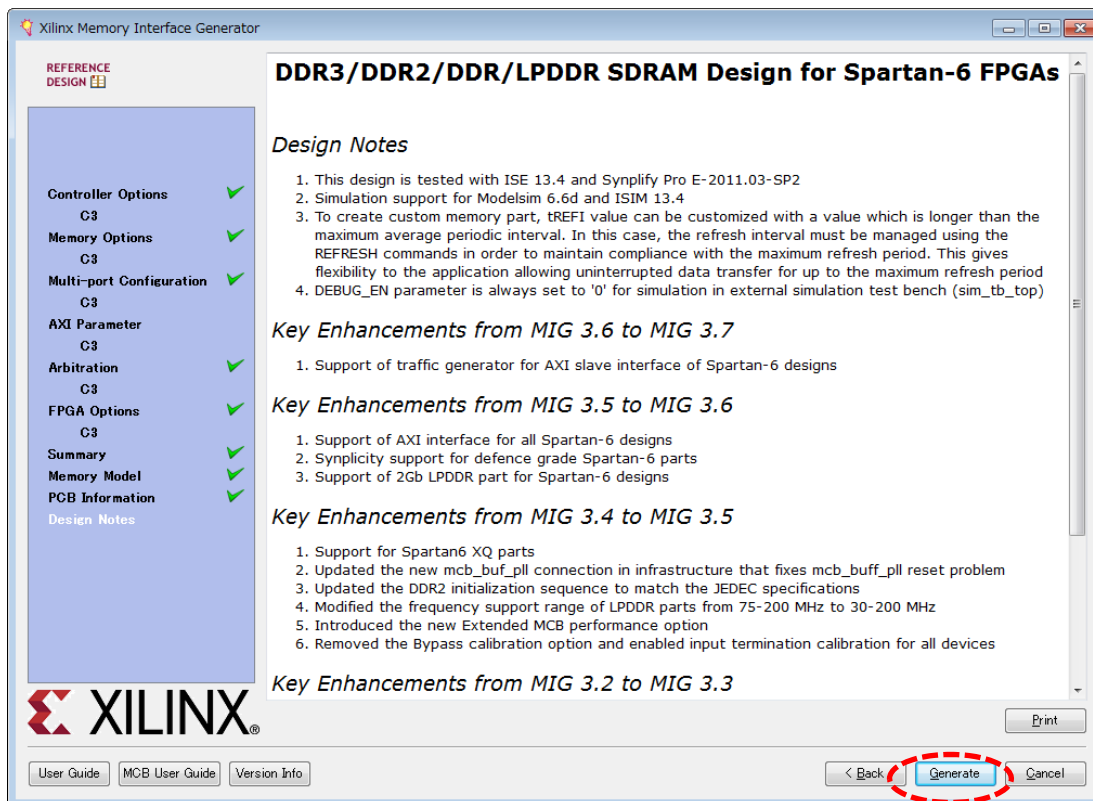


シミュレーションモデルのライセンスに Accept を選択、Next をクリック

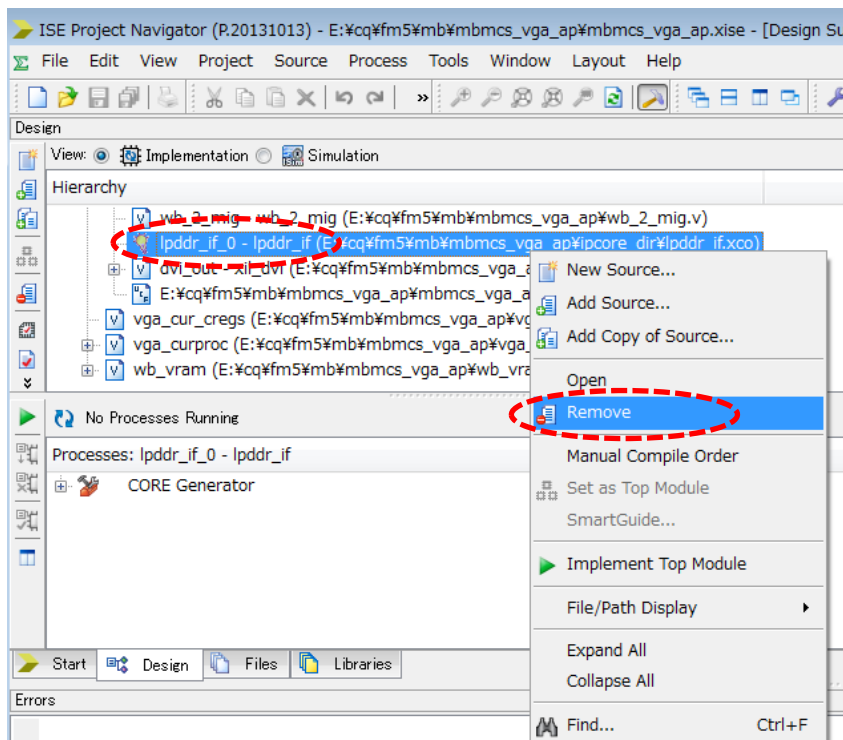


Next をクリック



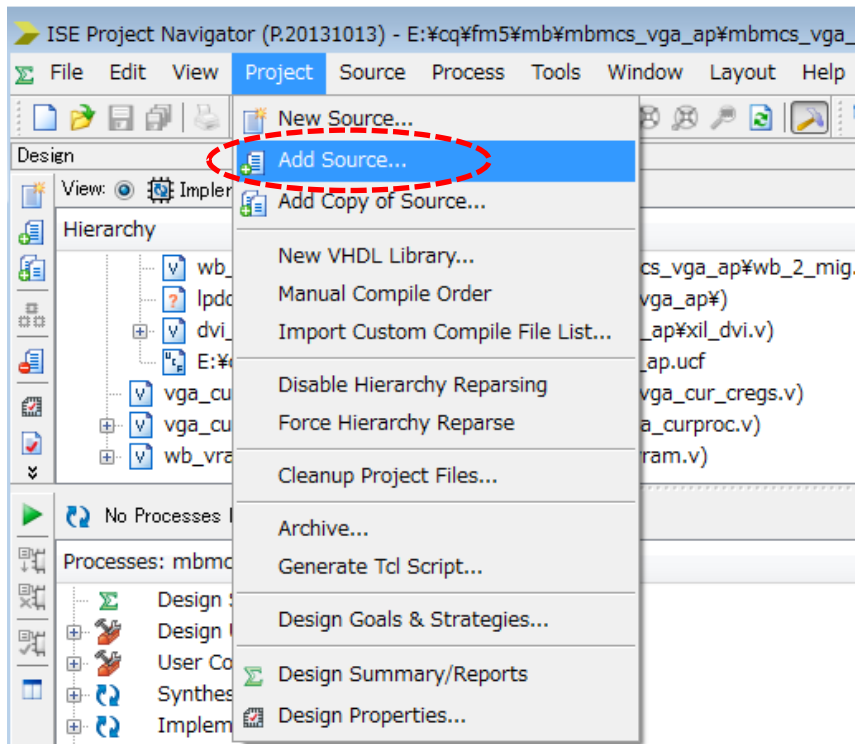


Generate をクリックすると lpddr_if の作成が開始

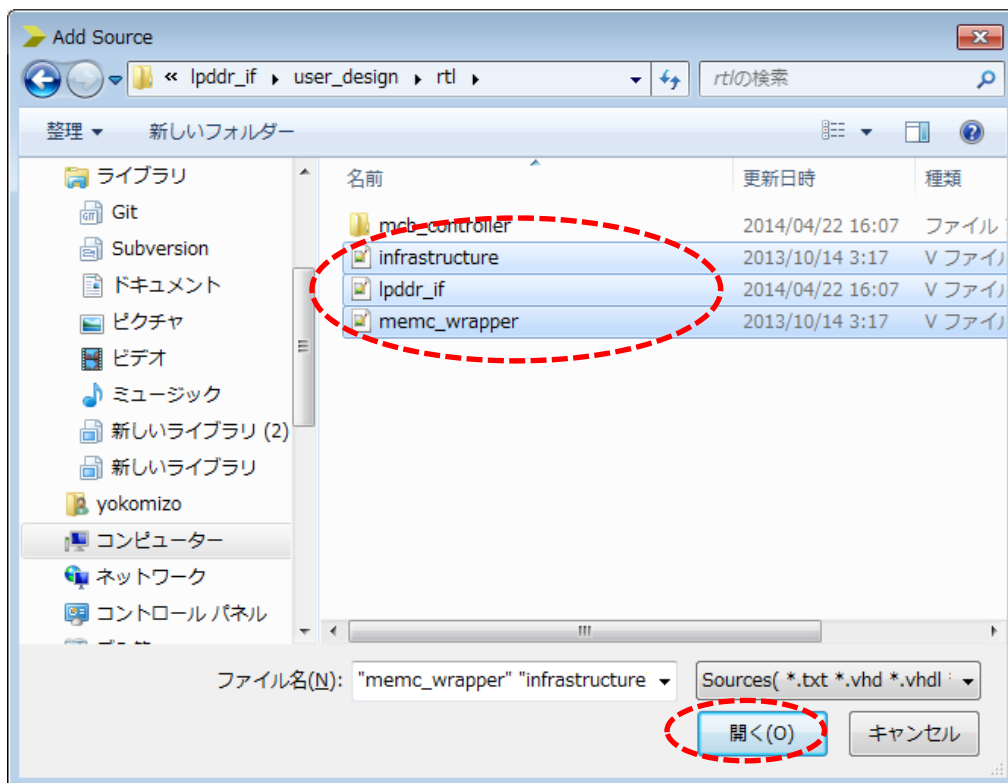


論理合成を実行すると MIG. xco が組み込まれるが、Verilog-HDL モデルに変更するので、lpddr_if. xco を削除する。lpddr_if 上でマウス右ボタン→Remove



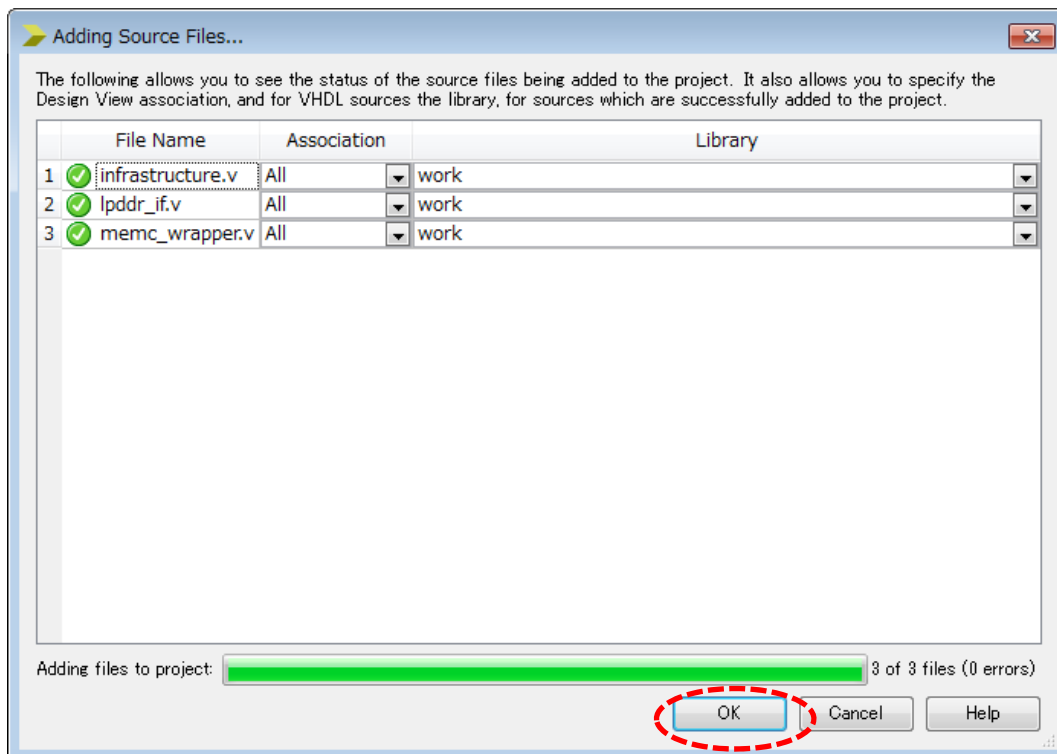


Add Source で lpddr_if の Verilog-HDL 記述をプロジェクトに追加する

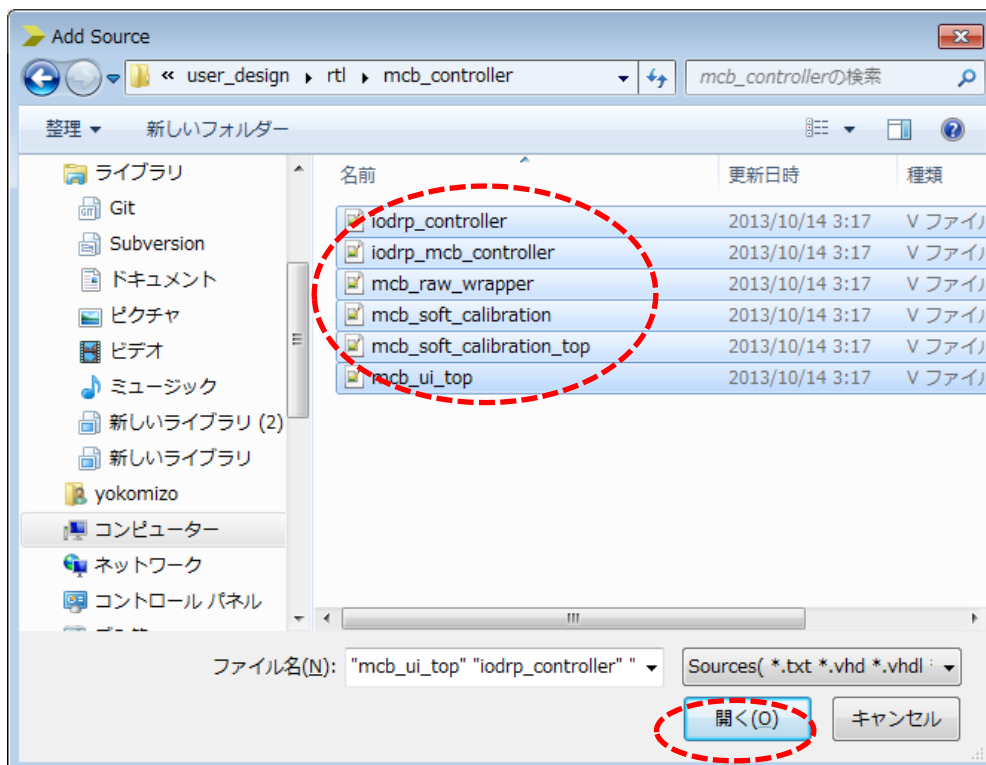


ipcore_dir/lpddr_if/uadr_design/rtl の下の Verilog-HDL ソースファイルを選択



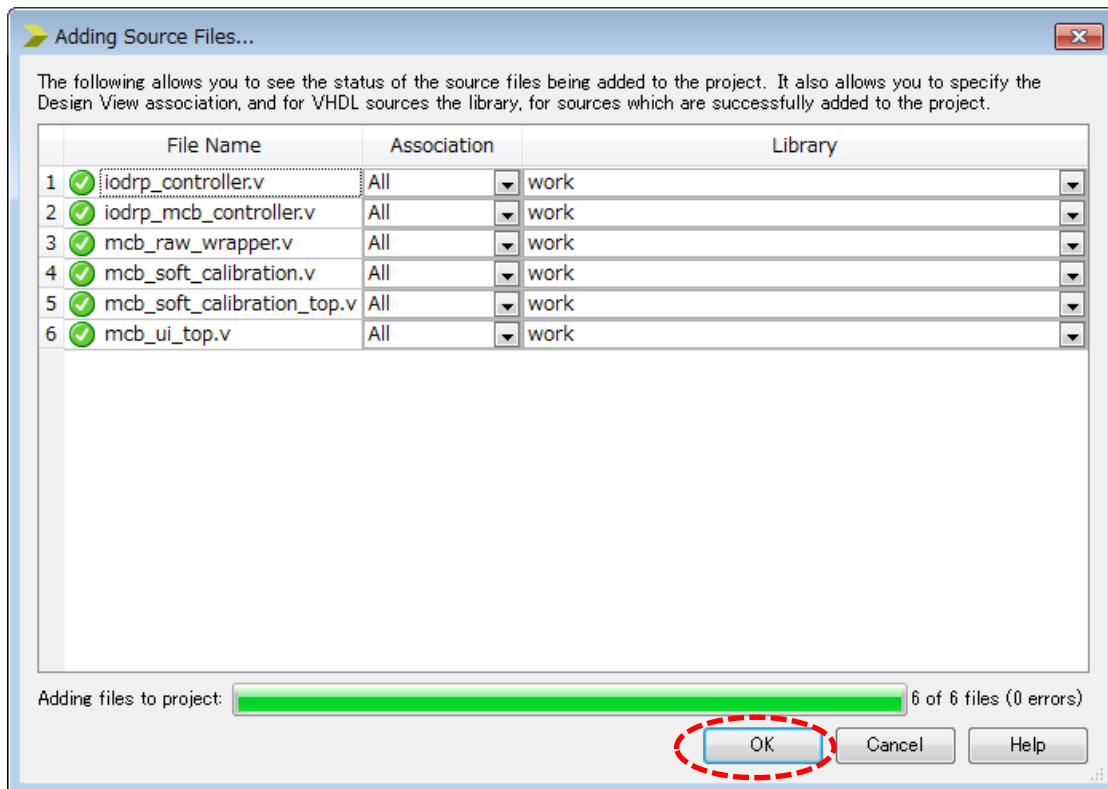


ソースファイルを追加

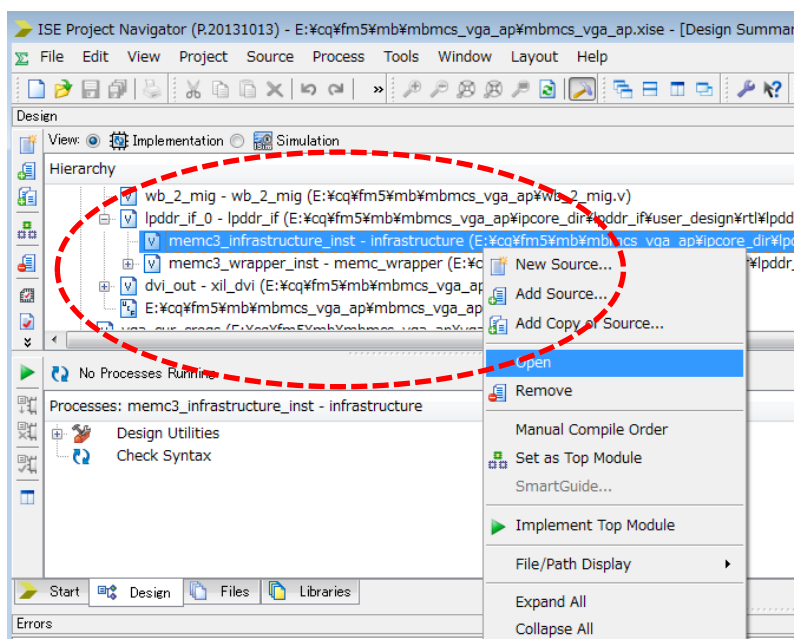


さらに ipcore_dir/lpddr_if/uadr_design/rtl/mcb_controller の下の Veri-g-HDL ソースファイルを選択



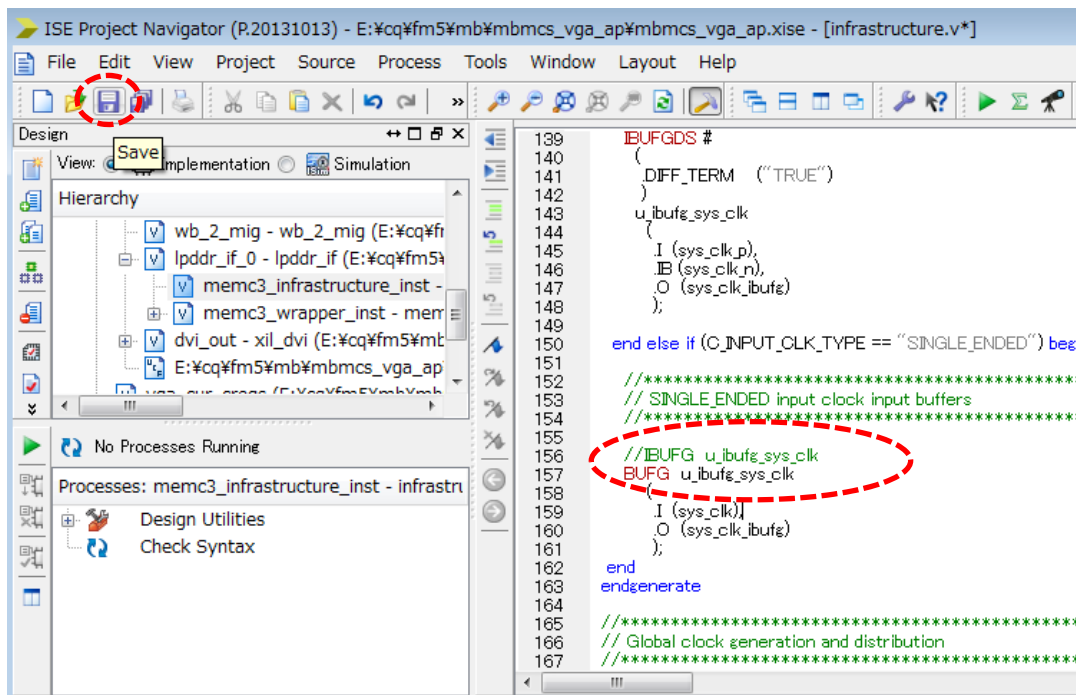


ソースファイルを追加すると、Verilog-HDL モデルの MIG が組み込まれる。

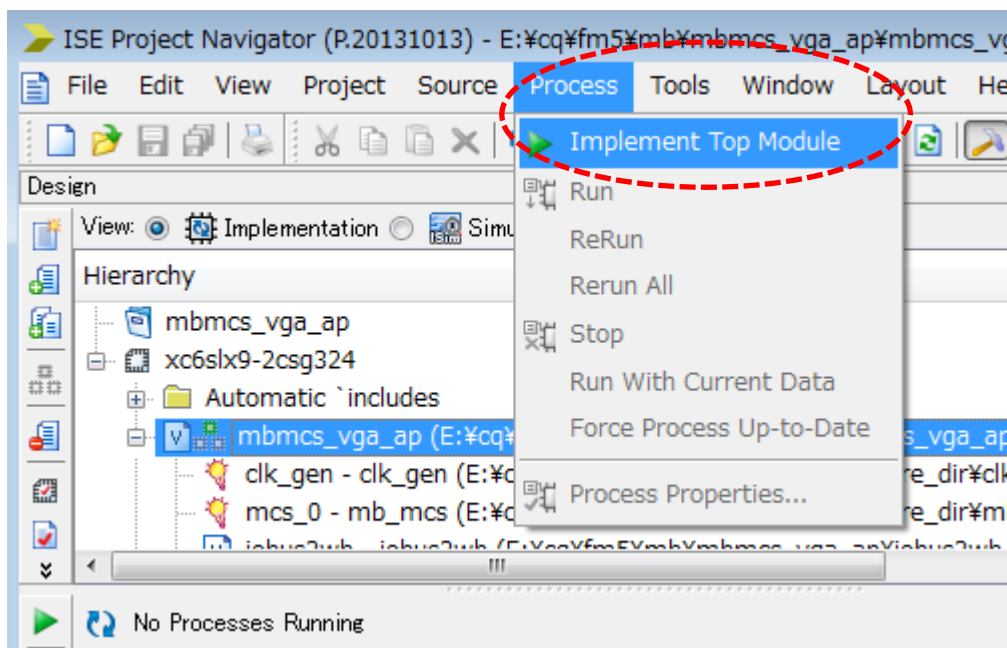


infrastructure を選択→open でファイルを開く





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] へ変更しセーブ。



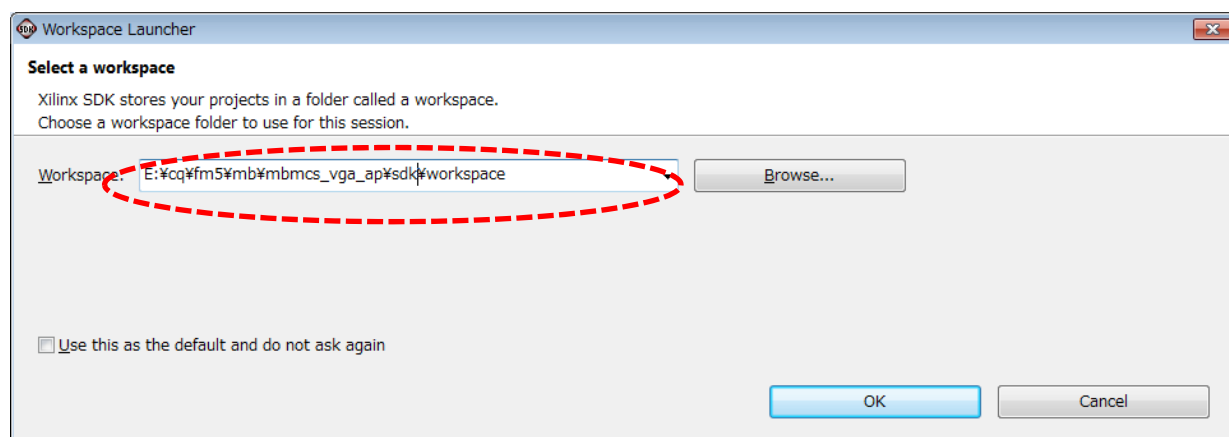
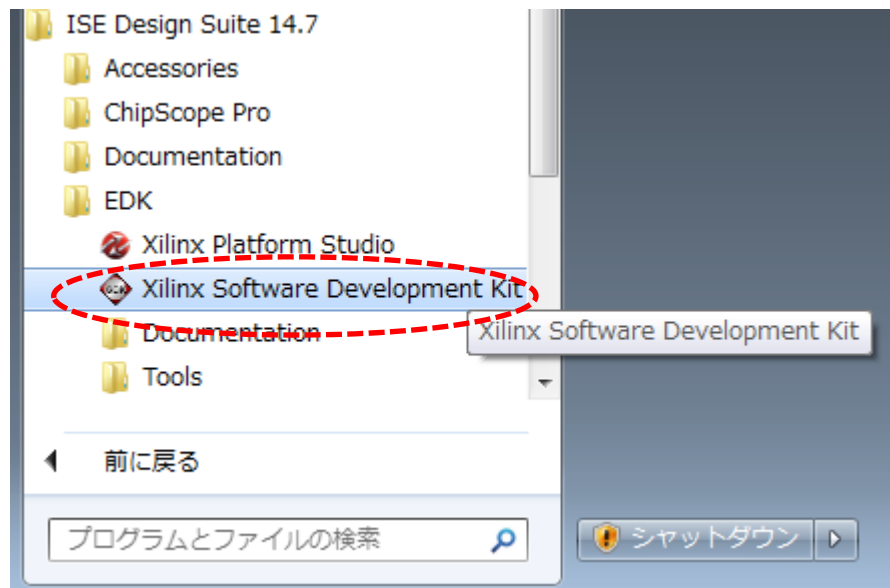
インプリメンテーションの実行、Process→Implement Top Module をクリック



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

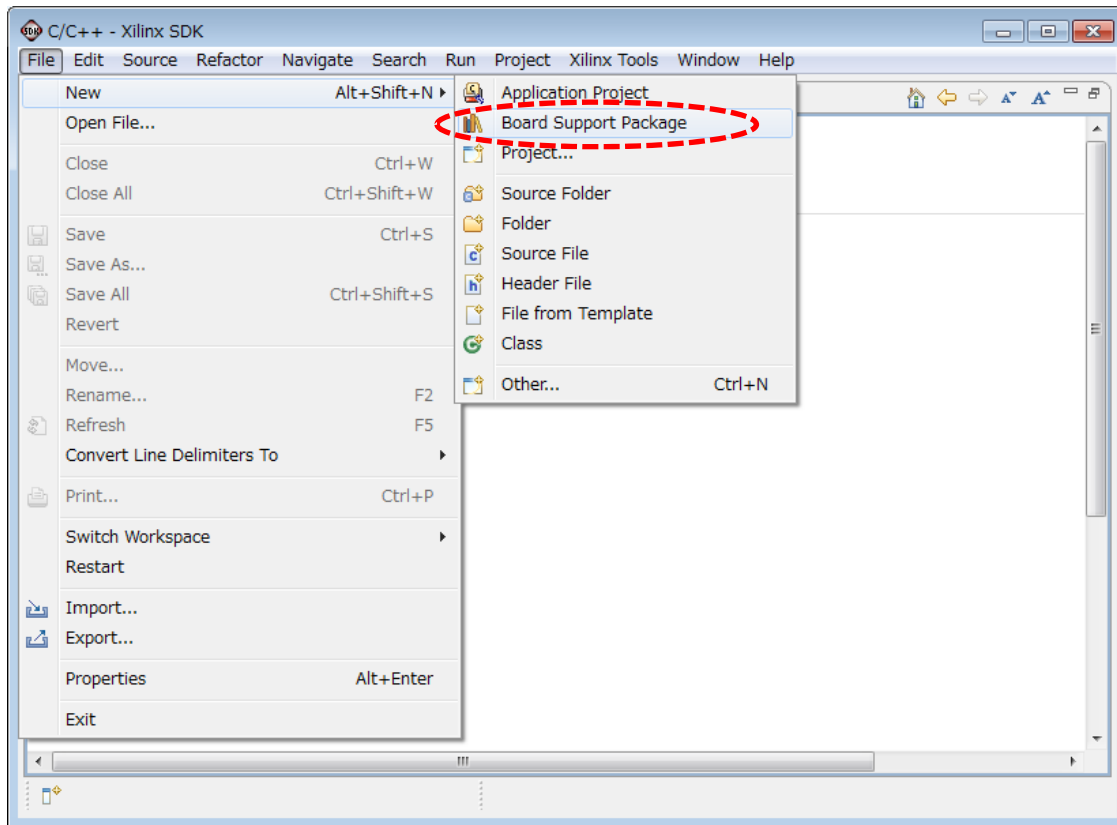
・ SDK の起動

スタートメニューから「Xilinx Design Tools」→「ISE Design Suite 14.7」→「EDK」→「Xilinx Software Development Kit」を起動してください。

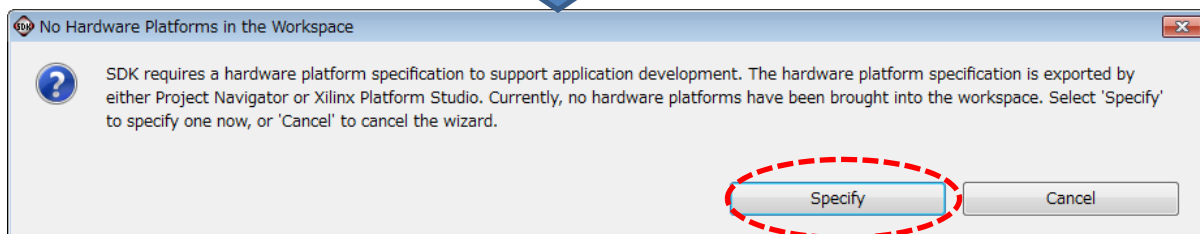


SDK を起動するとワークスペースを指定が要求されます。設計フォルダ/sdk/workspace を設定

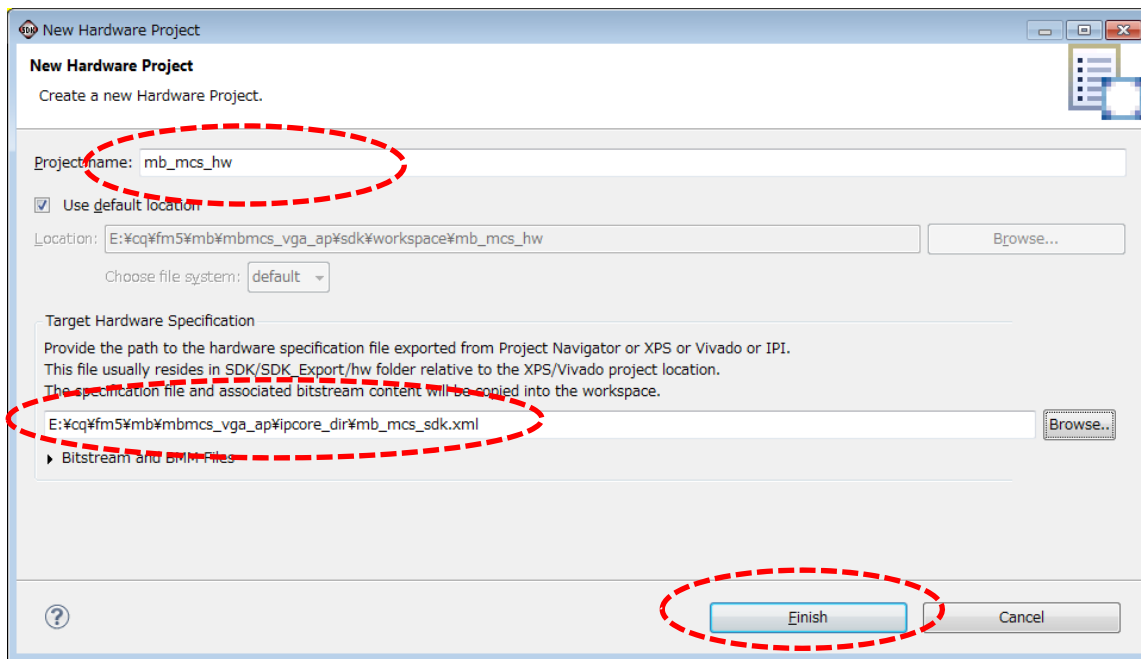




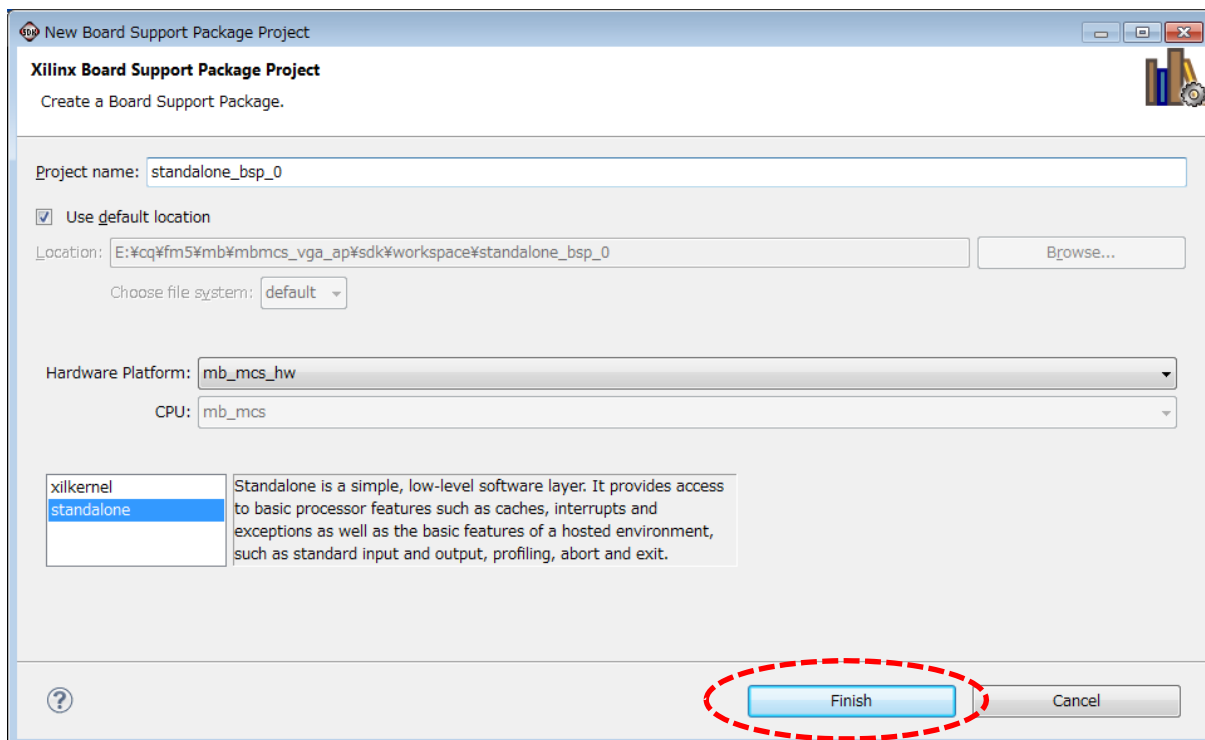
新規のボードサポートパッケージ作成、File→New→Board Support Package



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

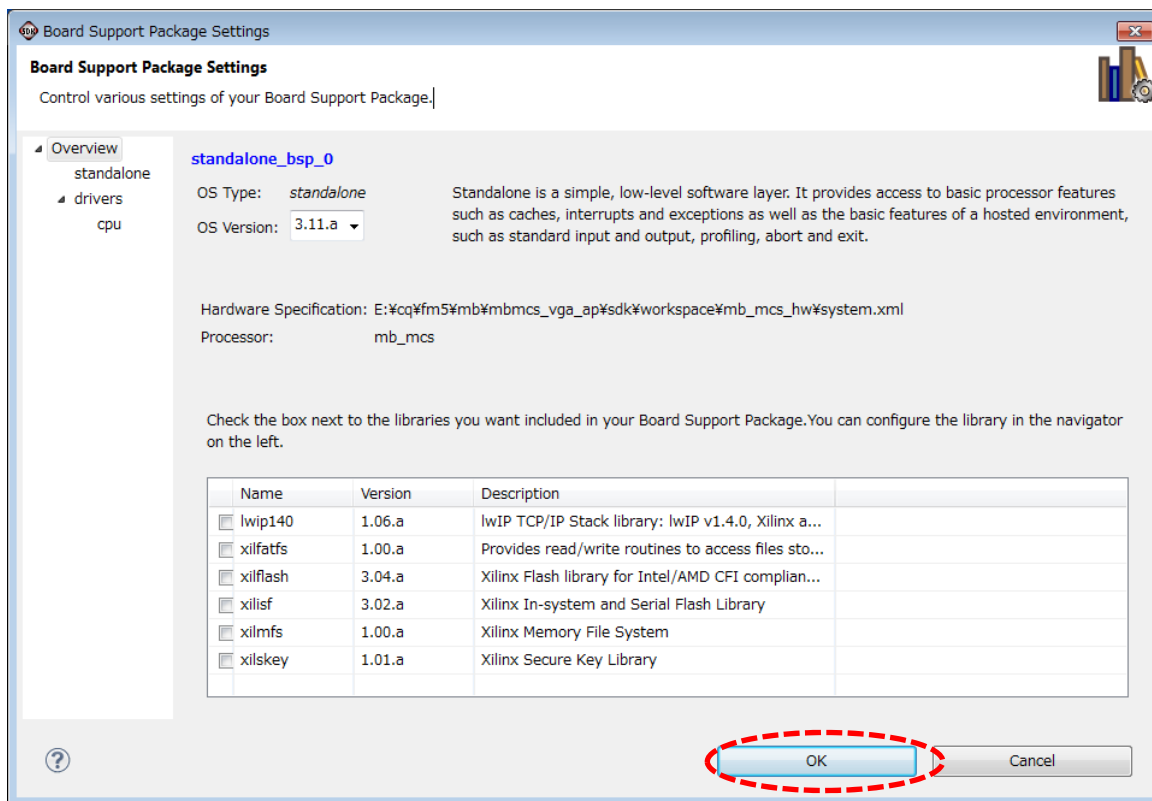


SDK の管理するハードウェアプロジェクト名指定 (mb_mcs_hw)
xml ファイル (設計ファルダ / ipcore_dir / mb_msc. sdk. xml) 指定
Finish をクリック

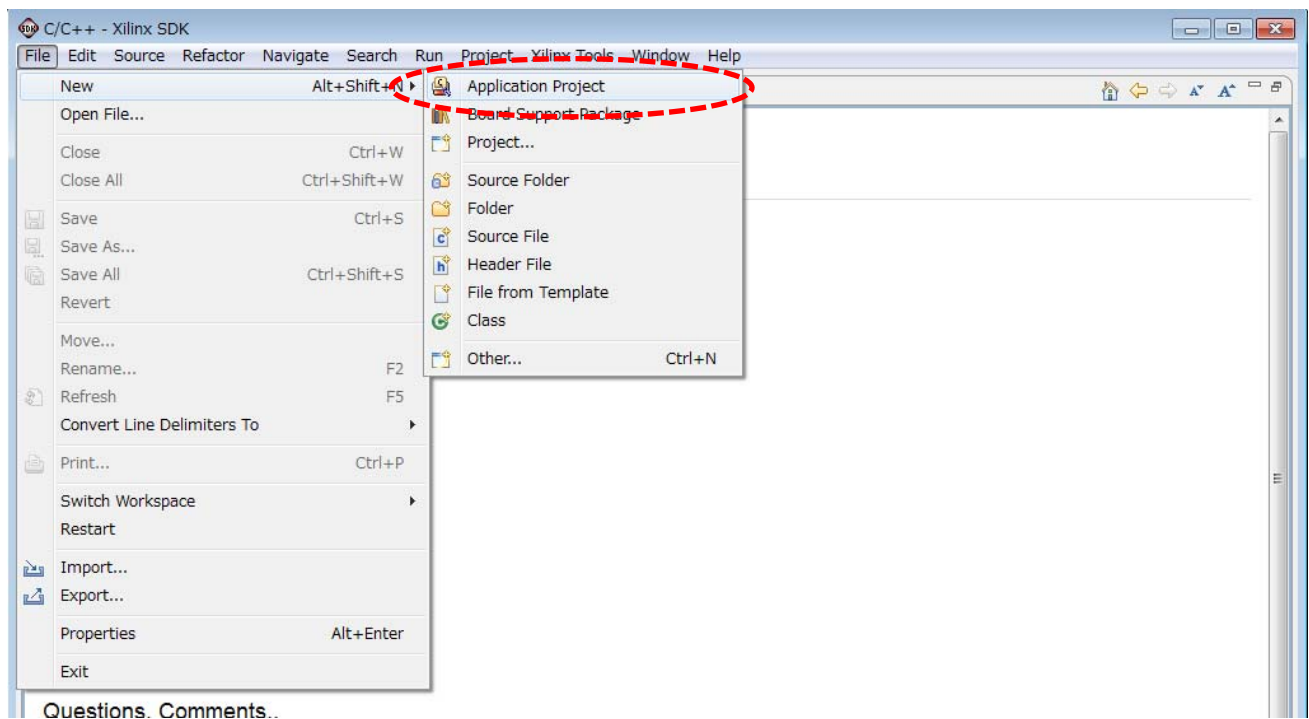


ボードサポートパッケージの定義
Finish をクリック



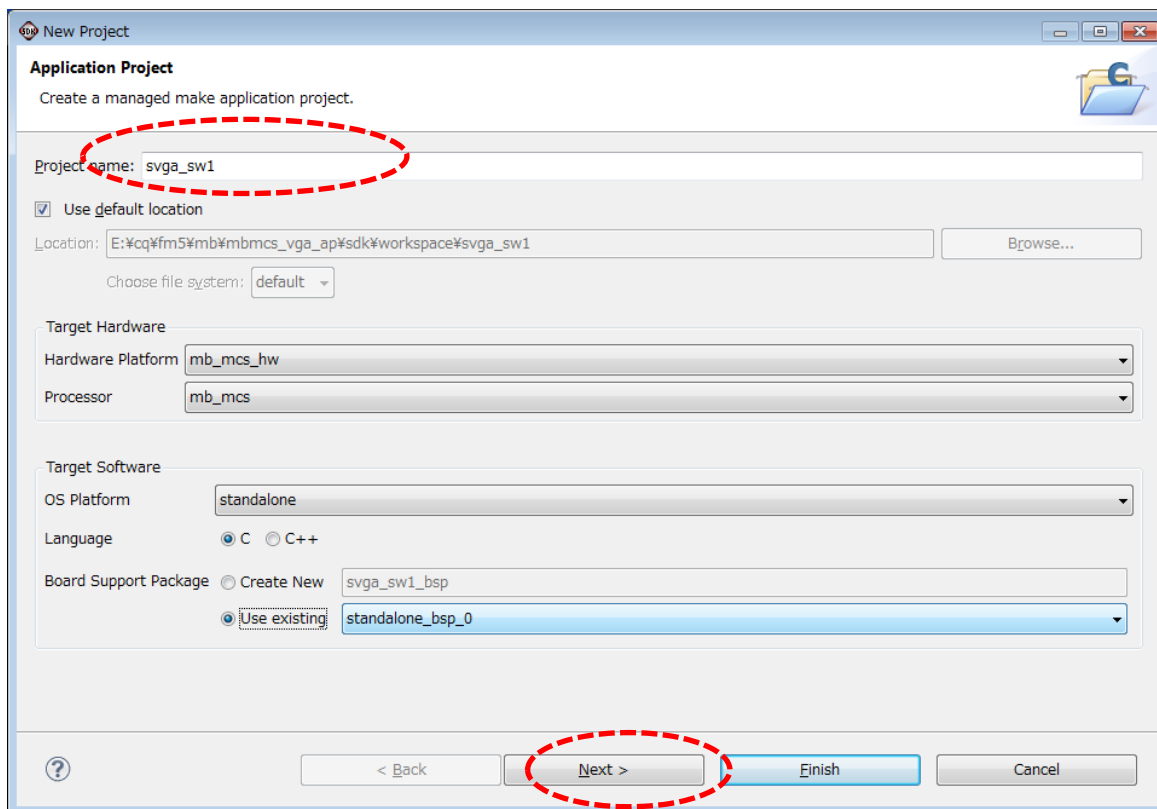


ボードサポートパッケージのオプション定義



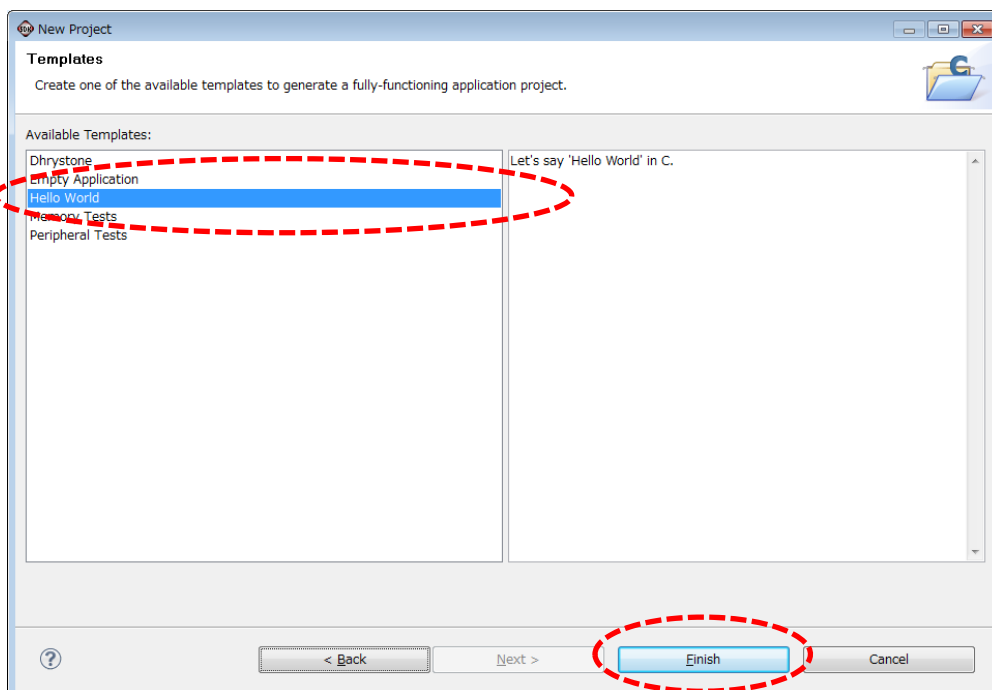
新規のソフトウェアプロジェクト作成、File→New→Application Project





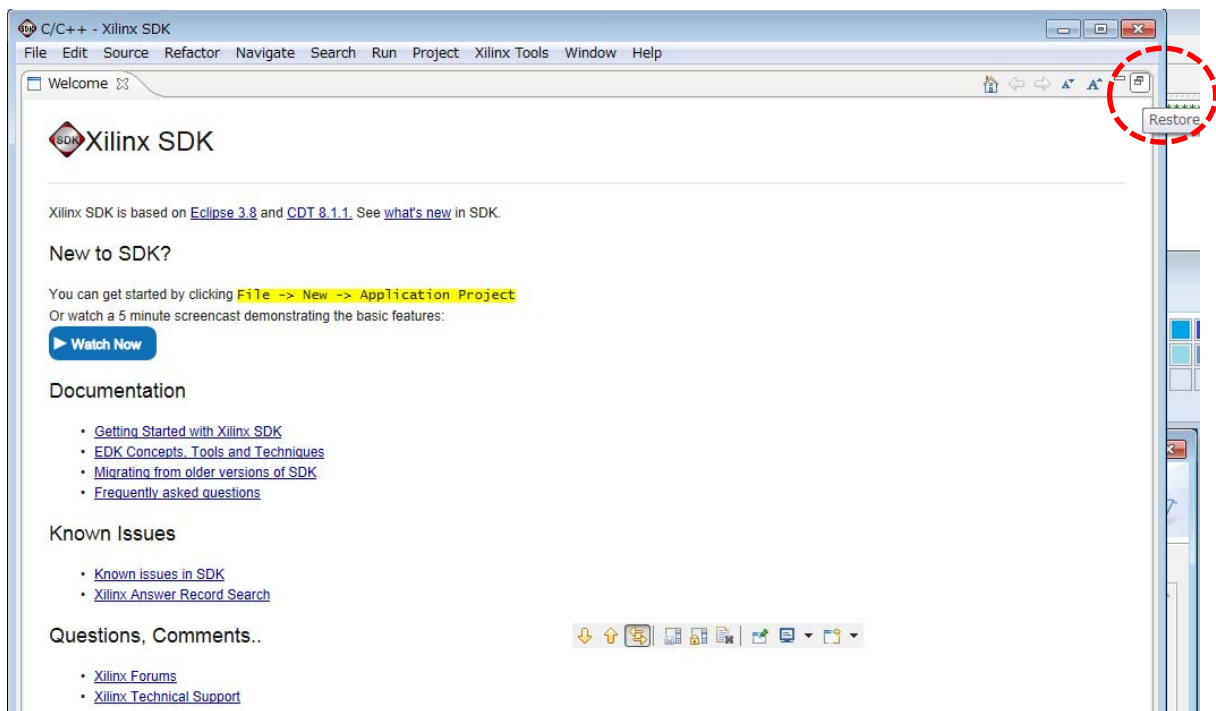
ソフトウェアプロジェクト名指定:svga_sw1

Next をクリック

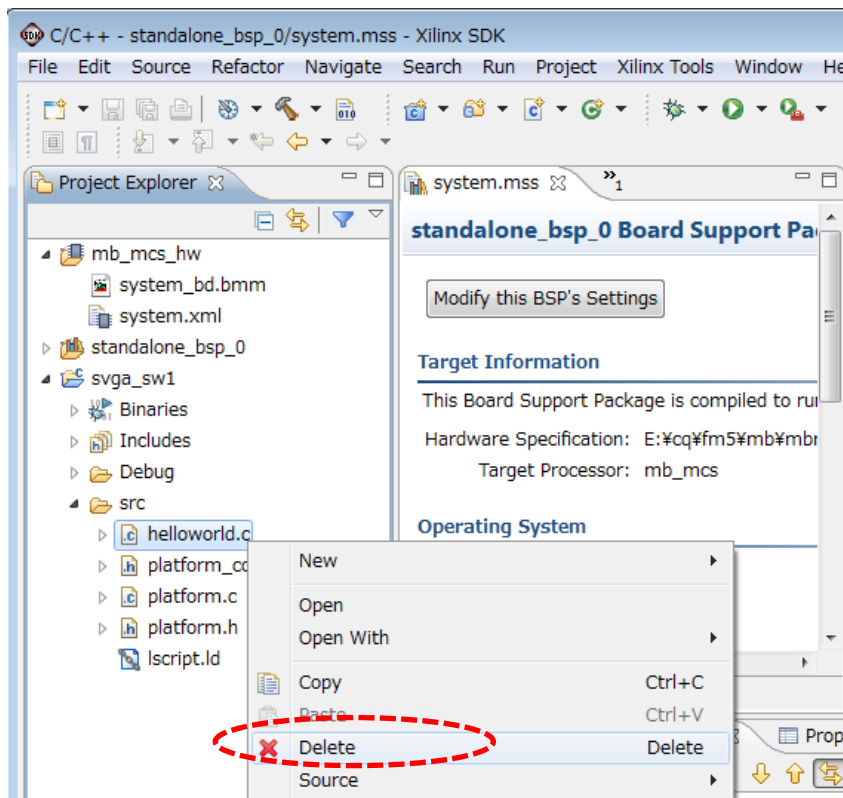


テンプレートに Hello World 選択後、Finish をクリックでソフトウェアプロジェクトが作成される。



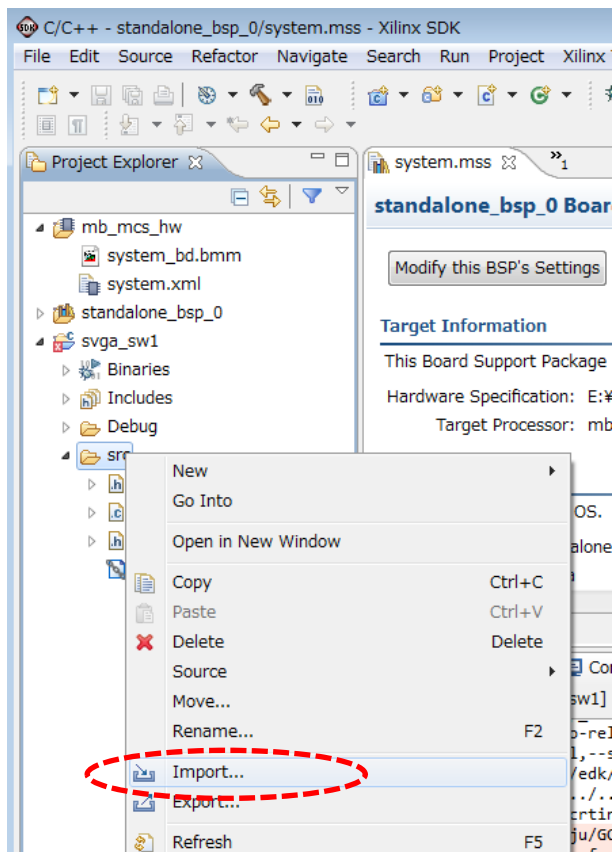


ウィンドの右上の Restore をクリックするとプロジェクトが表示される

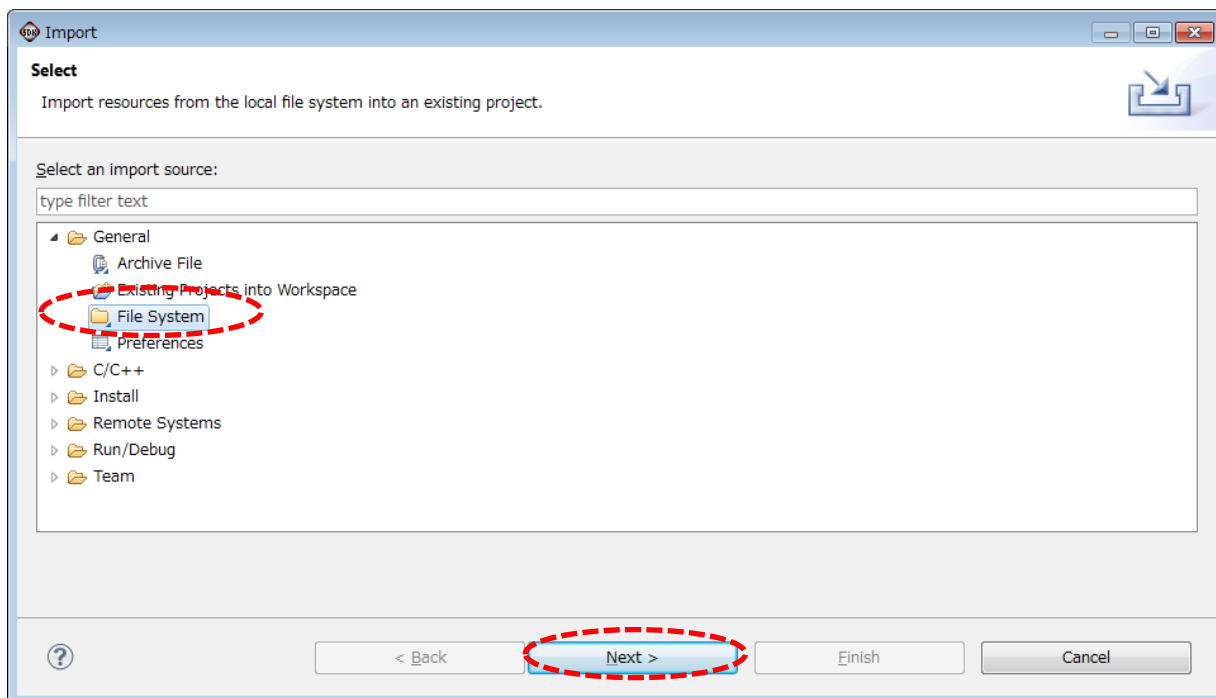


svga_sw1→src の下にある helloworld.c を削除



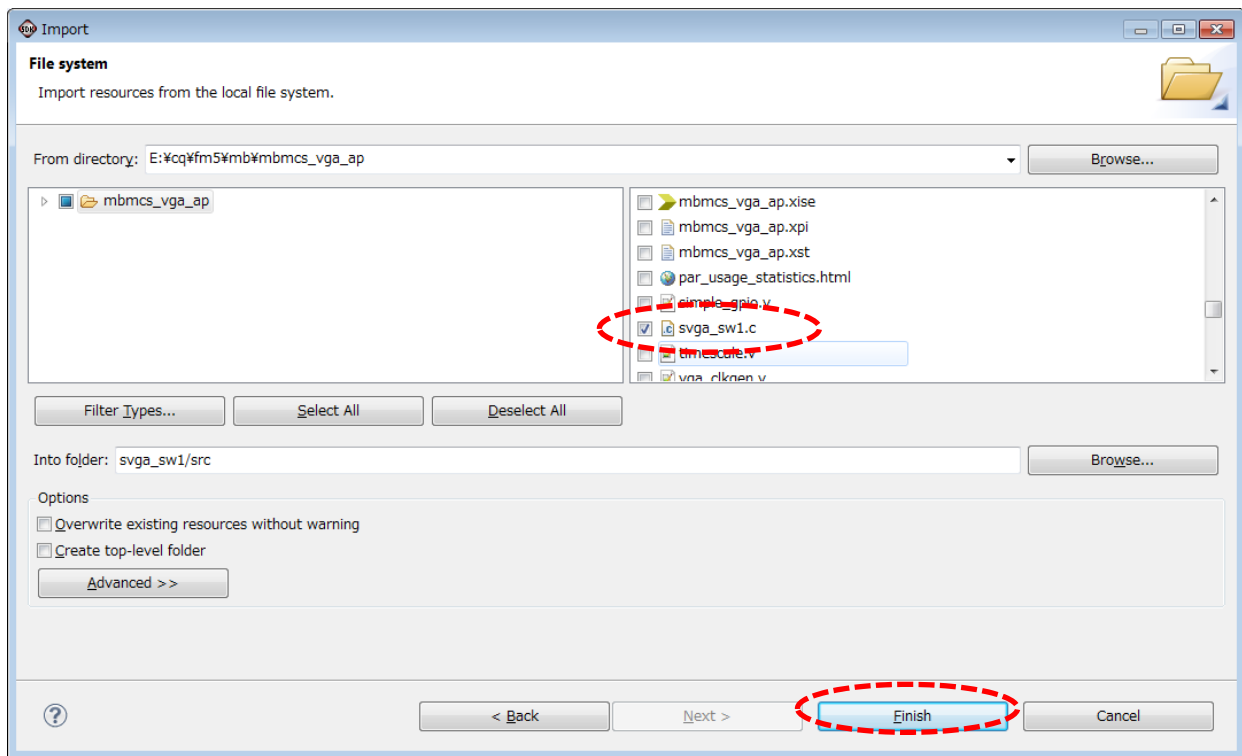


svga_sw1→src 上でマウス右ボタンを押して Import 選択

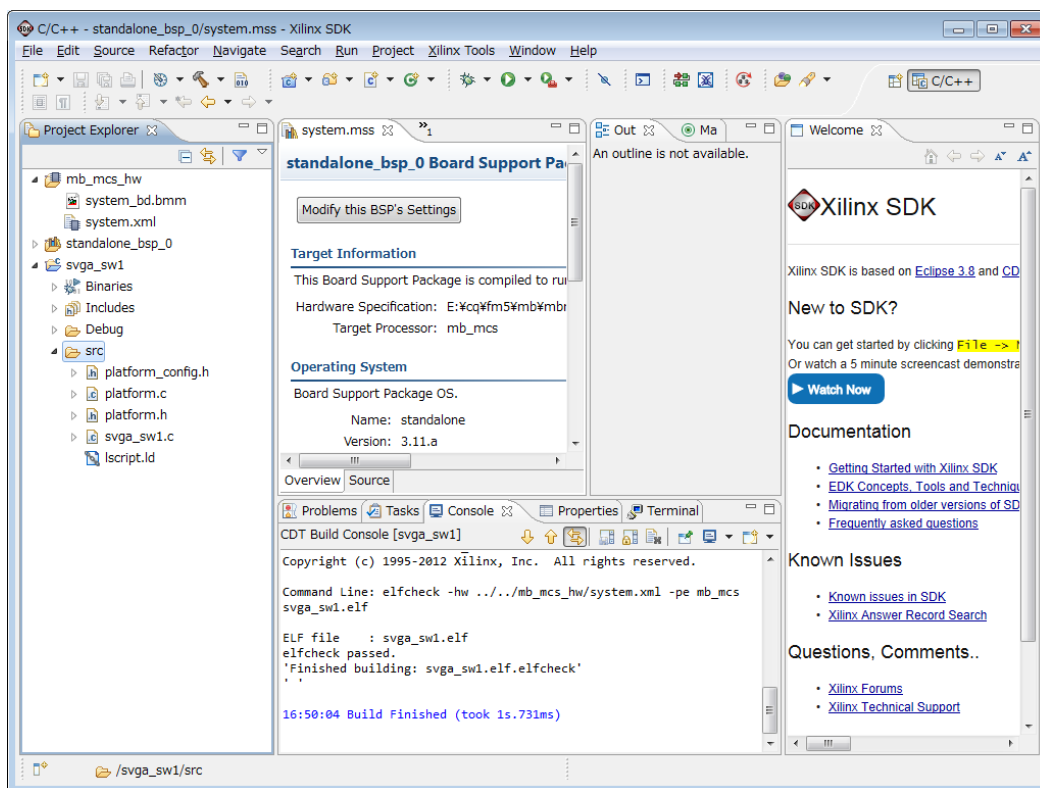


File System を選択して、Next





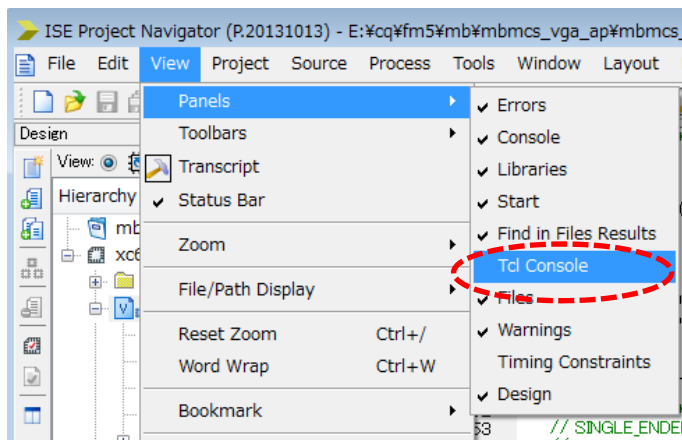
mbmcs_vga_ap の下にある svga_sw1.c を選択して、Finish



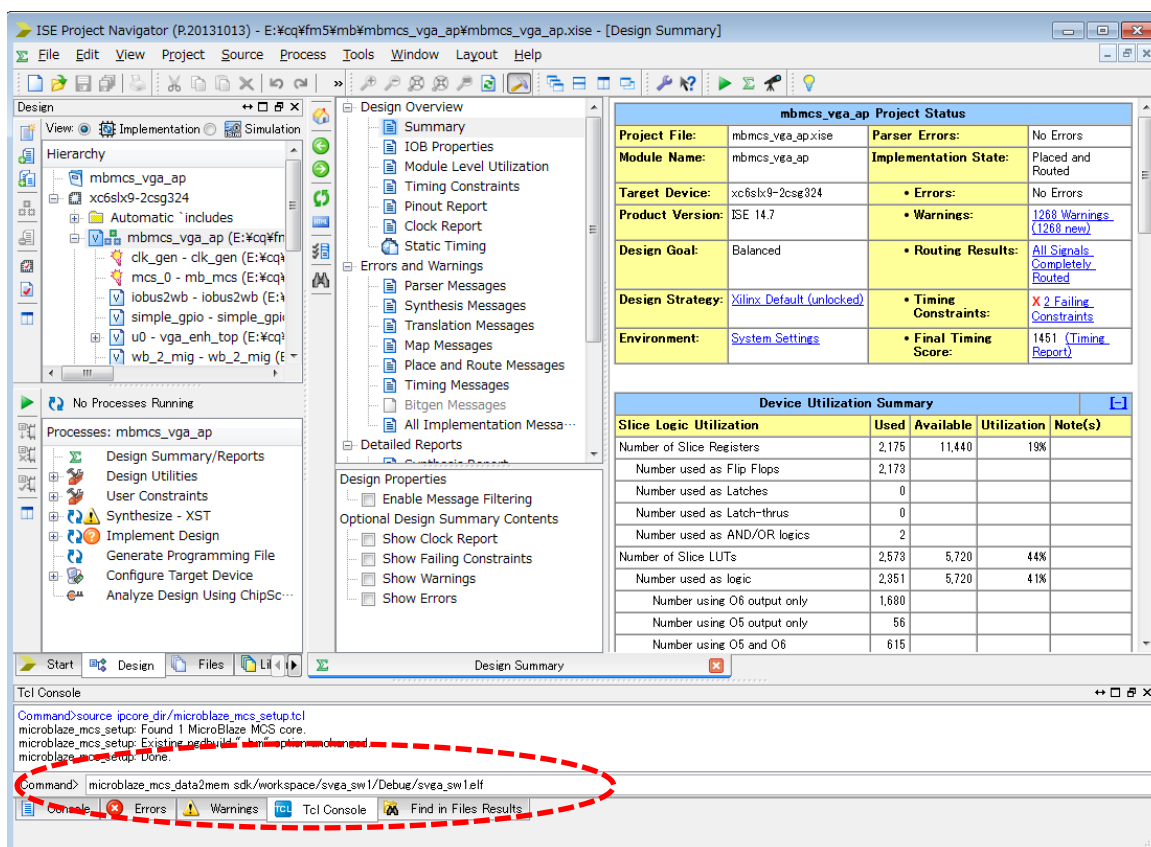
自動でビルドが実行される



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



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]
```

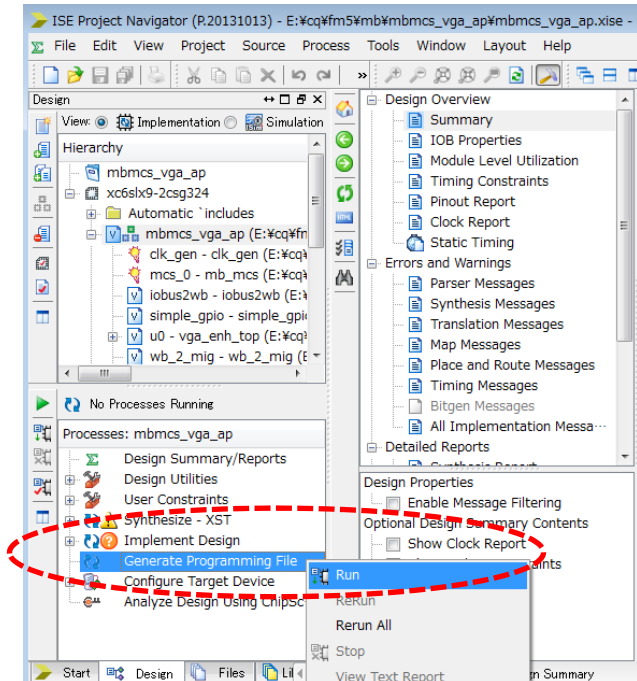
```
microblaze_mcs_data2mem sdk/workspace/svga_sys_sw1/Debug/svga_sys_sw1.elf [Enter]
```



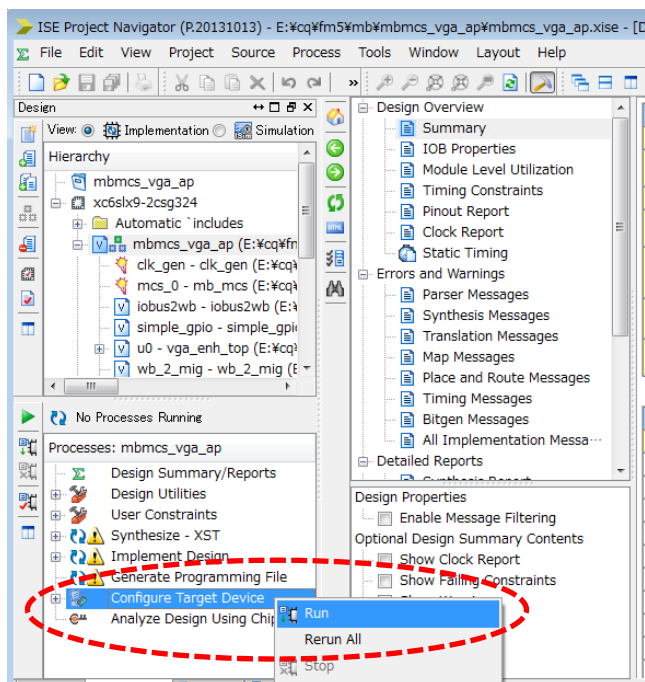
FPGA で動作確認します。

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。

ISE に戻る

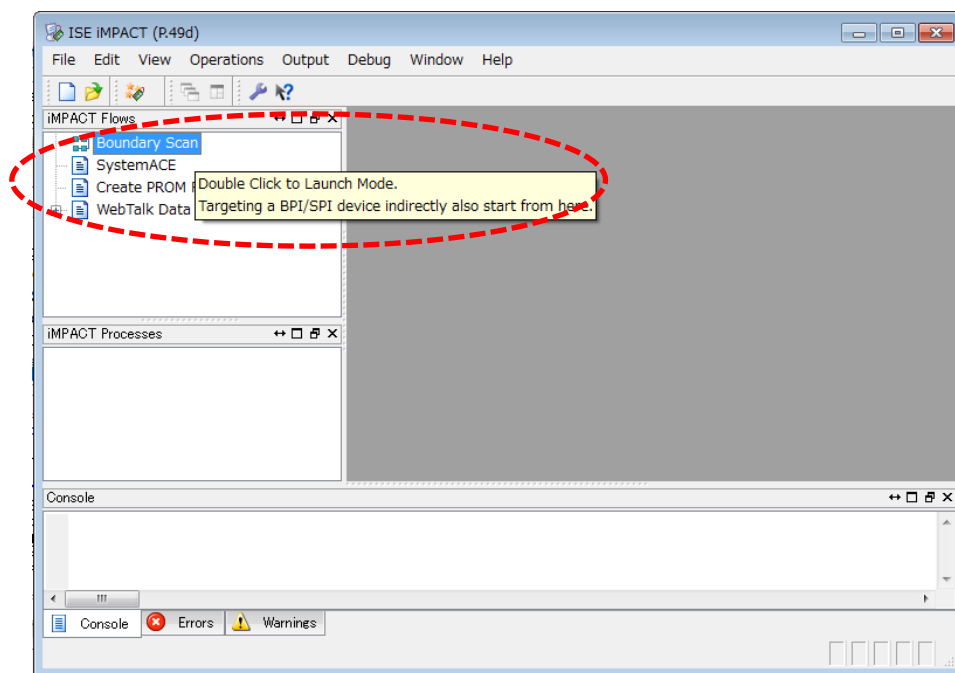


FPGA へ書き込む mb_mcs_sys.bit 作成

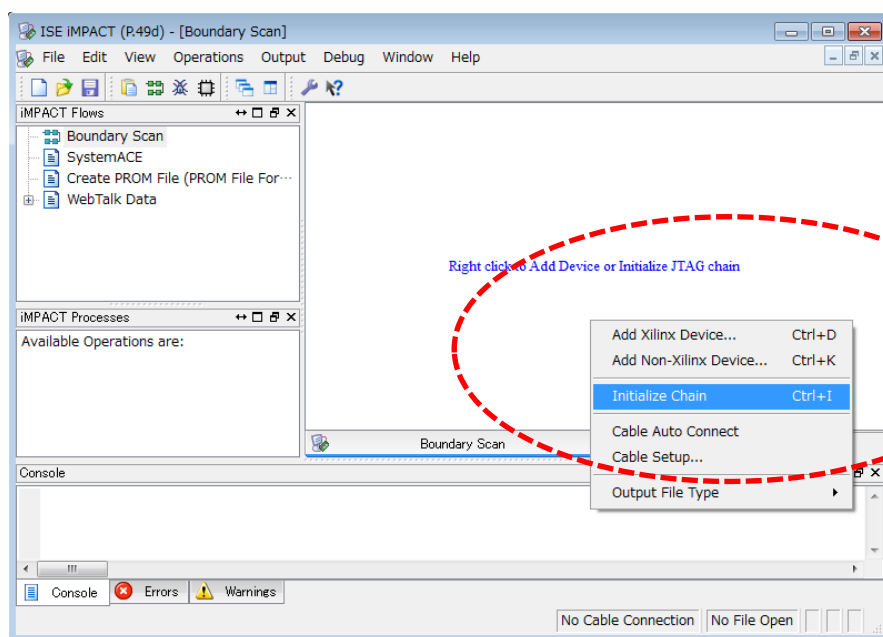


Project Navigator で iMPACT を起動

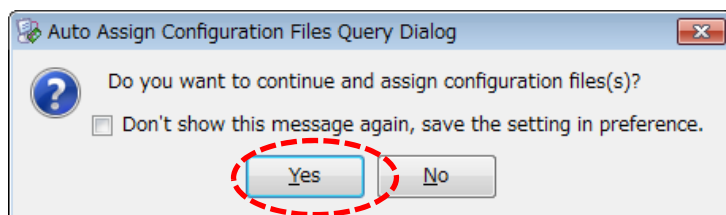




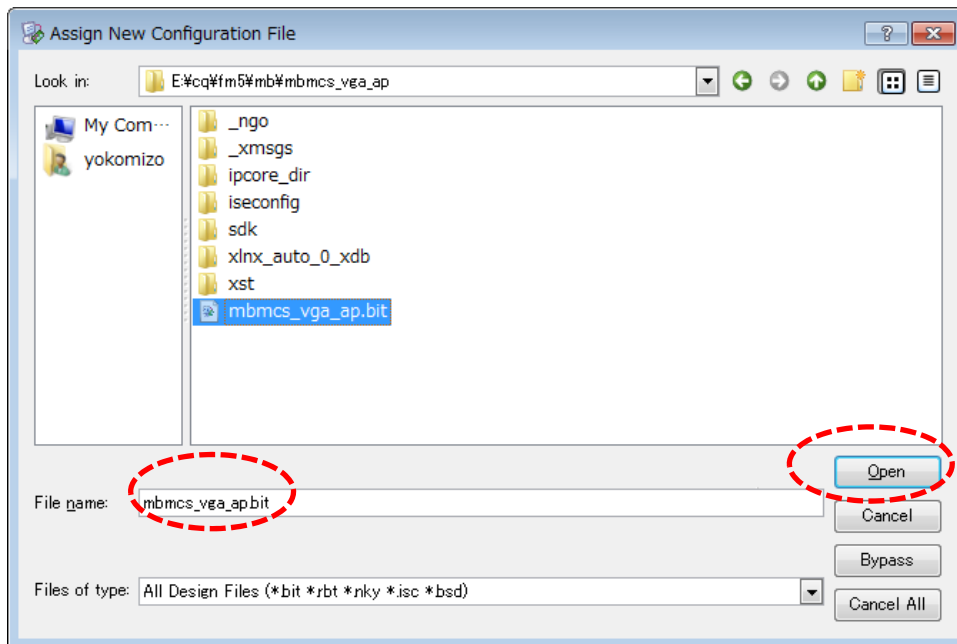
BoundaryScan モードにする



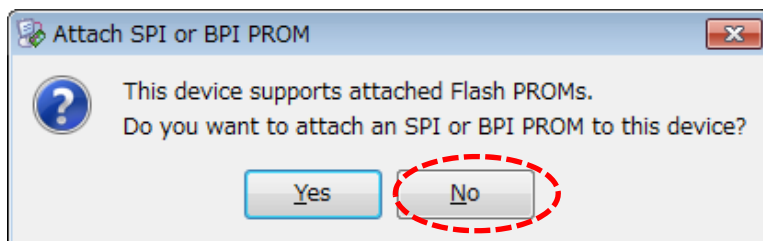
BoundaryScan のウィンドをマウス右ボタン押して、Initialize Chain 選択して FPAG を検出する



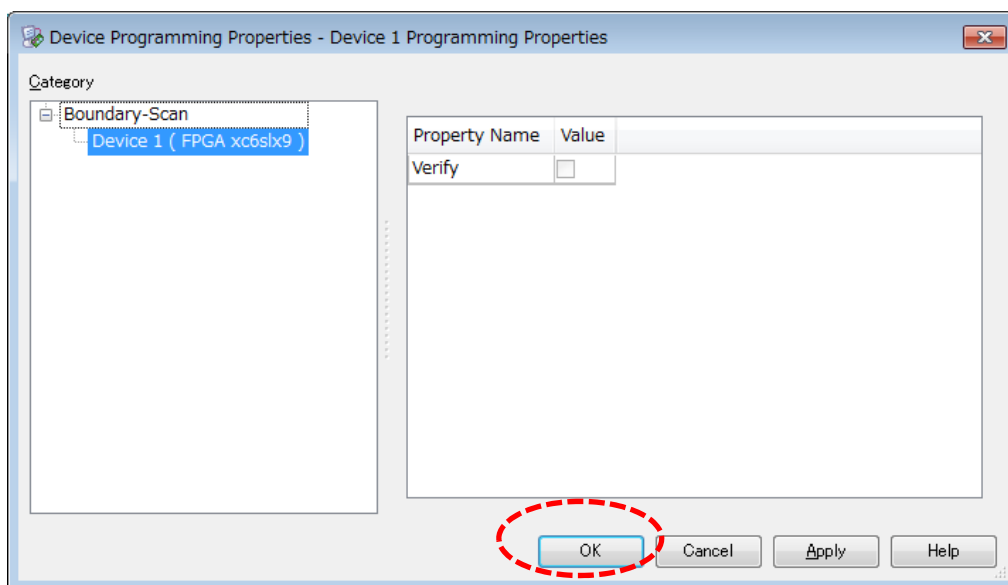
Yes で FPGA に書き込むファイルを指定する

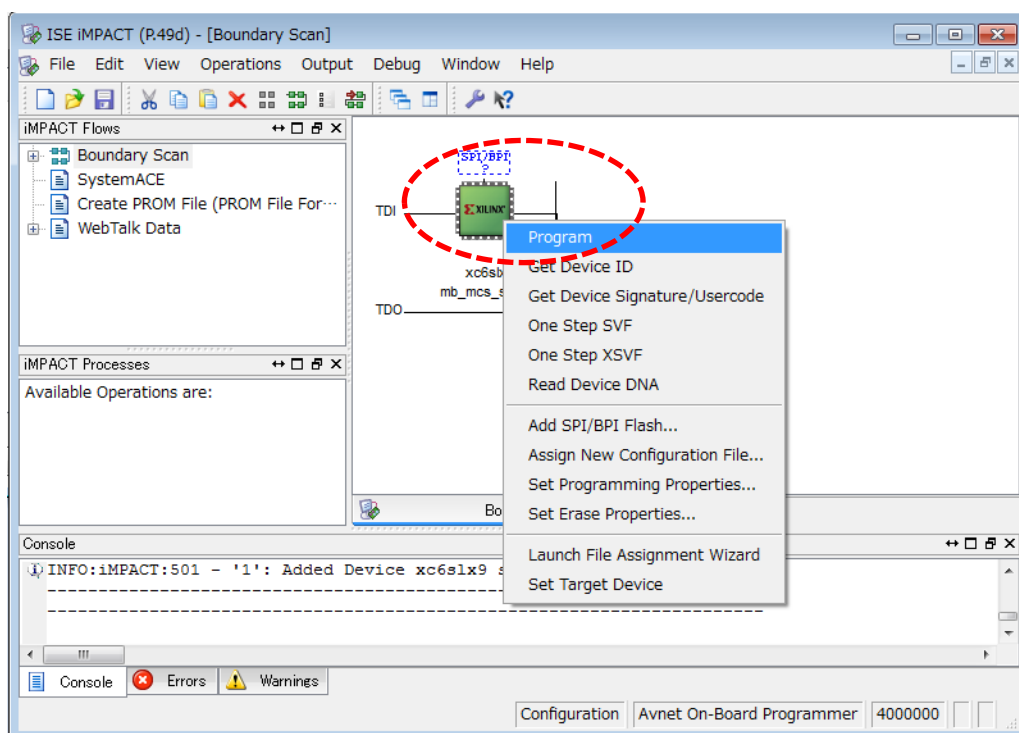


mb_mcs_sys.bit を指定

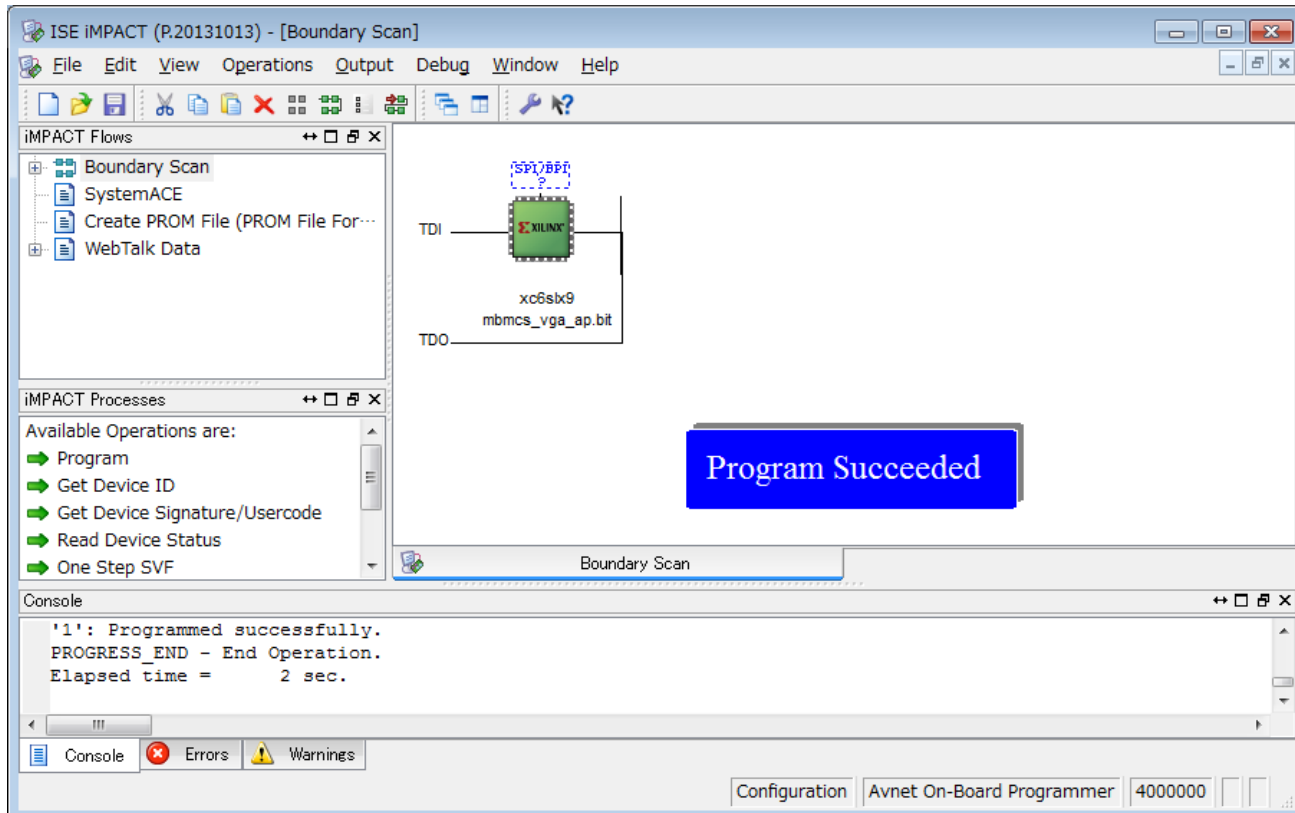


PROM データは使わないので No を選択





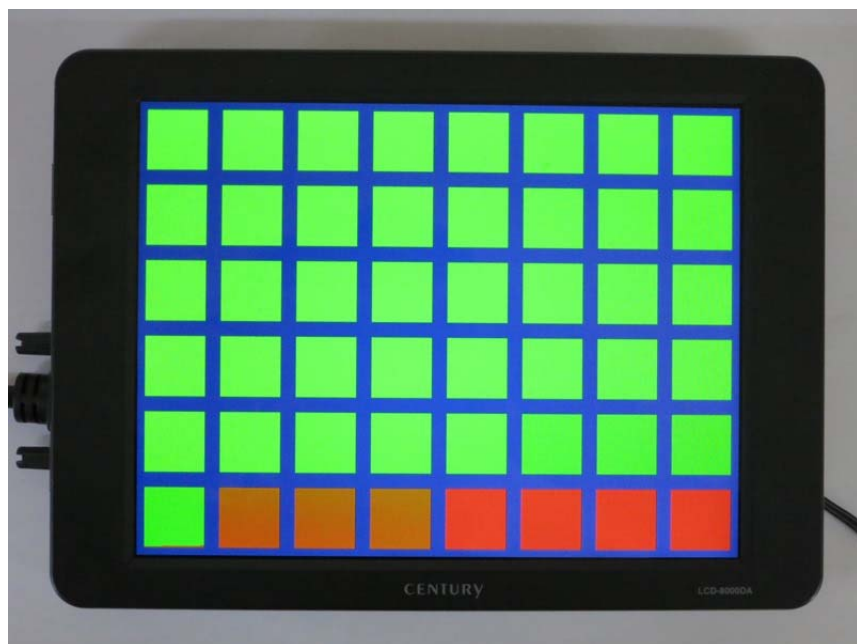
FPGA へのプログラミング実行、デバイス上でマウス右ボタンを押して Program 選択



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。



プログラムで作成したマス目模様が HDMI で接続したモニタに表示されます。



以上で動作確認終了です。