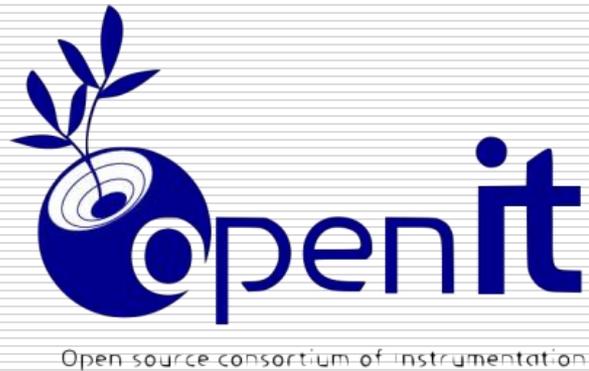


Open-It FPGAトレーニングコース(入門編)

7. IPの使用方法



第3.2版

2016年6月22日

IPとは？

- IPとはIntellectual Propertyの略で知的財産を意味します
- FPGA業界では既に設計された設計資産の意味で使用されることが多いです
- ユーザーから見た場合、既に設計された回路ライブラリの事だと思ってください
- Xilinxは有償/無償で様々なIPを提供しています
- ここでは頻繁に使用するクロック管理回路を実装する事でIPの実装方法を学びます

クロック管理回路の必要性

□ FPGA外部回路との同期

- 同期回路で設計する事が多いのでFPGA外部回路と内部回路で使用するクロックの位相を合わせなければいけない

□ FPGA内部で複数クロック使用

- 現在の回路は異なる周波数の複数クロックを使用する事が多い
- 例)ADCは50MHz, イーサネットは125MHzなど

クロック管理回路

以下のような専用クロック管理回路をFPGA内に用意する事で解決

高性能なアナログ回路なので専用回路を予め作り込む必要あり

- 外部クロックとの位相合わせ
 - フィードバック回路による遅延調整により実現
- 一つのクロック元から異なる周波数の複数クロック生成
 - クロック源のN/M倍の周波数を生成可能

7シリーズではこのIPをMixed-Mode Clock Manager (MMCM)と名付けている

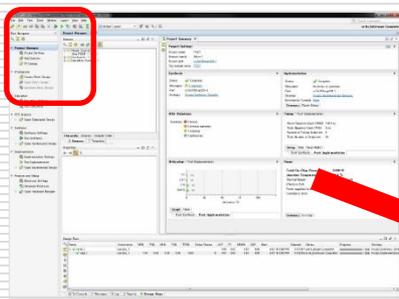
MMCM

- 頻繁に使用するXilinx FPGA機能
 - 多くの人が最初に使用するIP

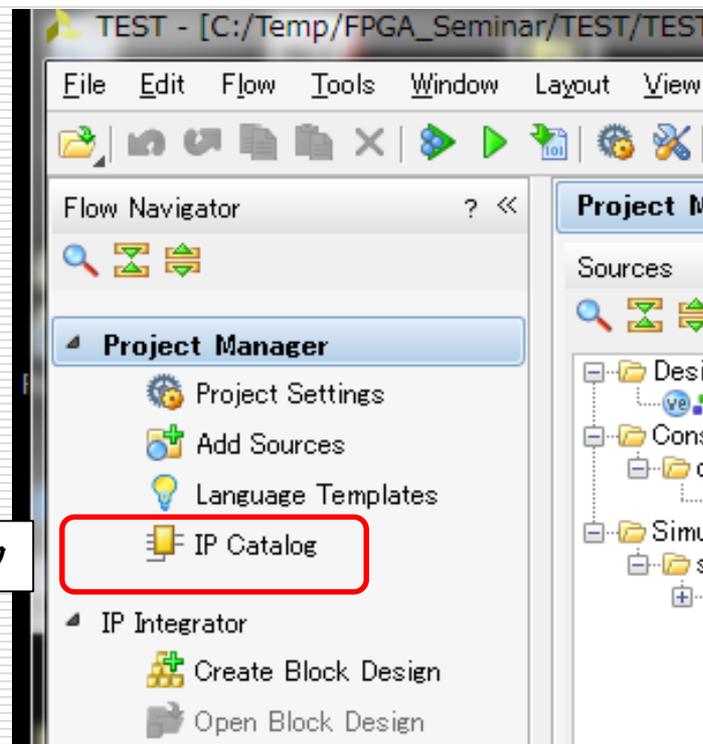
- クロック生成機能
 - 入力クロックのN/M倍のクロック生成可
 - 位相差も付ける事が出来る
 - 例) 100MHz入力→200MHzシステムクロック

TEST.vにMMCM機能を追加して
システムクロックを200MHzへ変更します

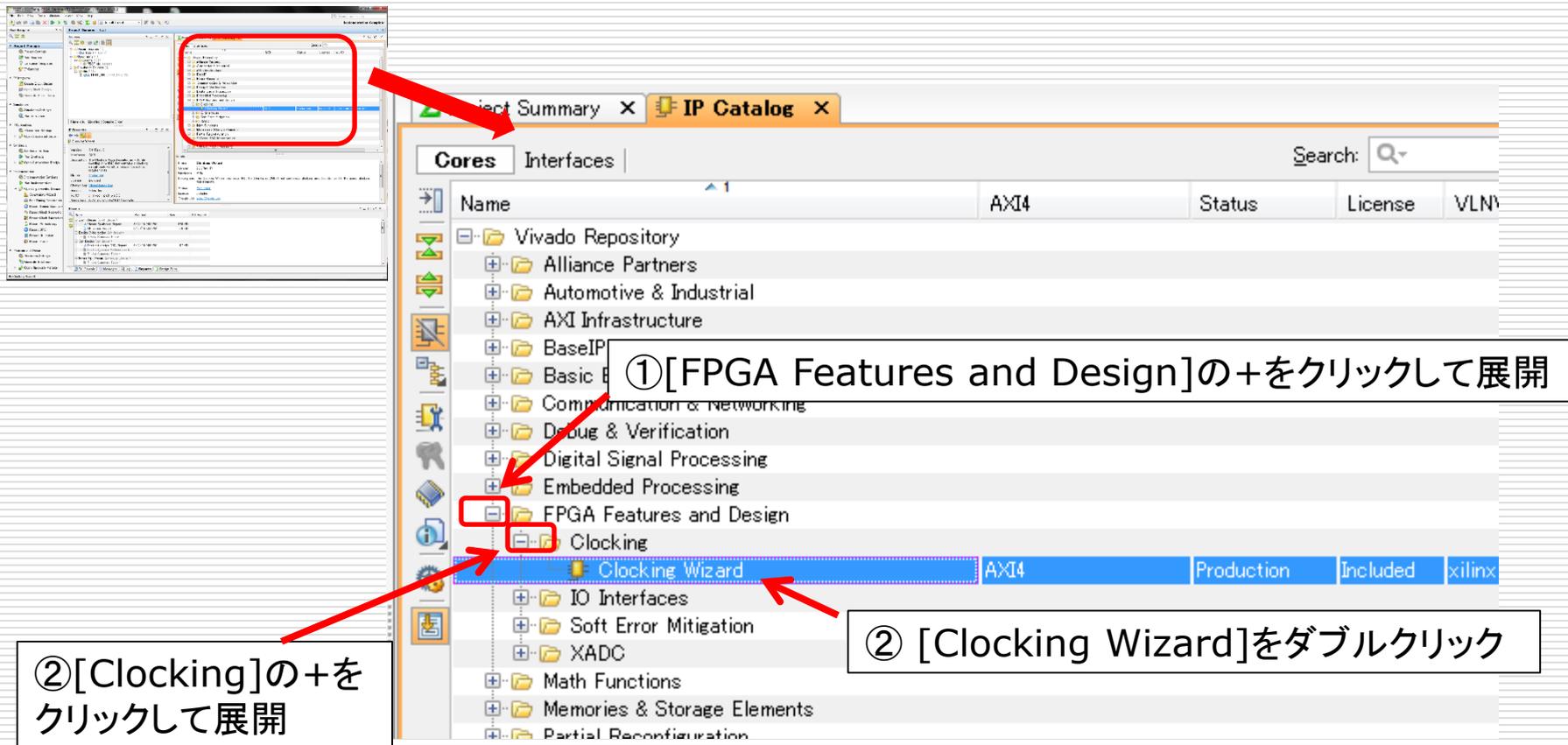
IP Catalogを開く



[IP Catalog]をクリック



Clocking Wizardの起動



Project Summary x IP Catalog x

Cores Interfaces Search: Q

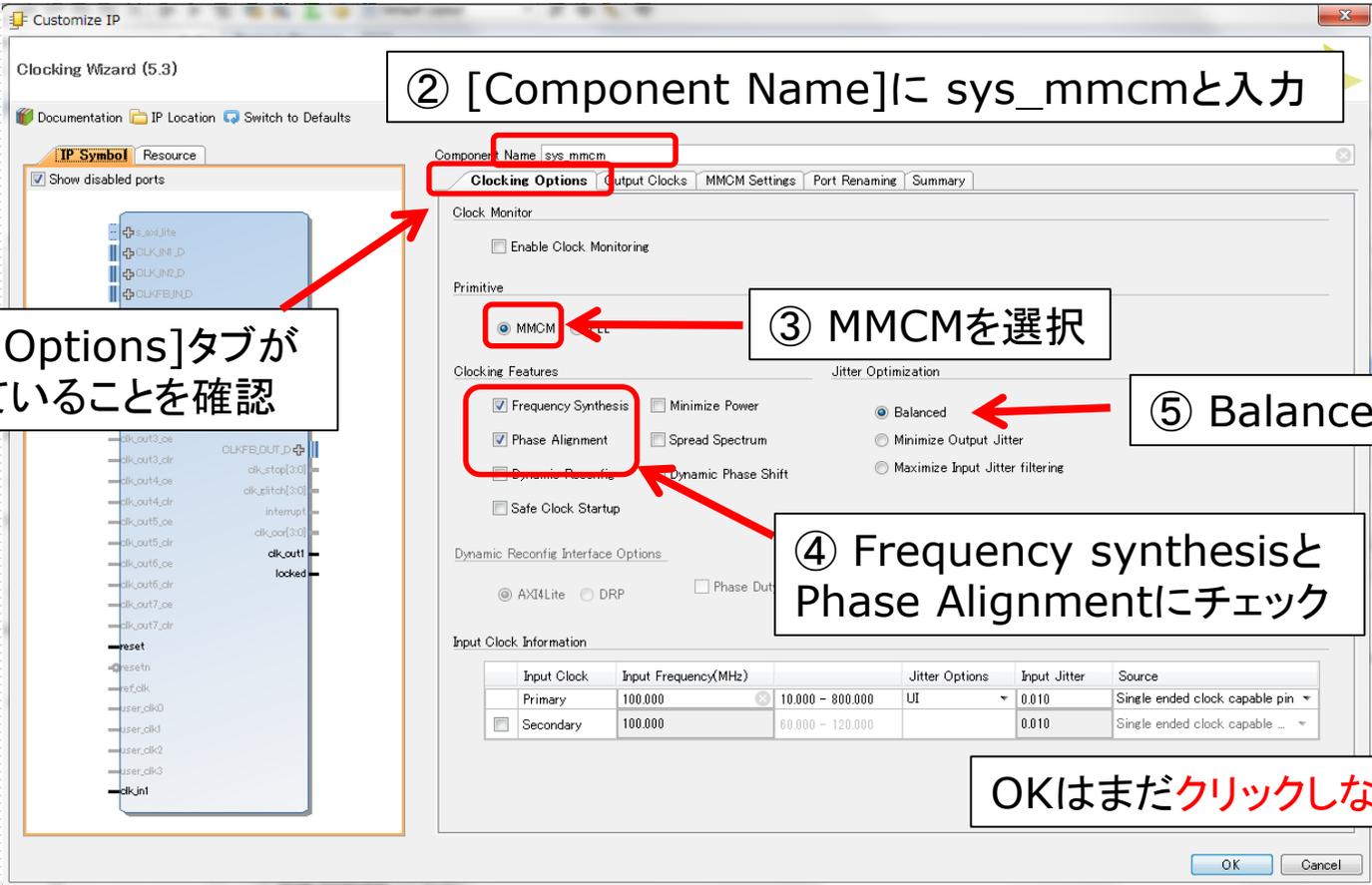
Name	AXI4	Status	License	VLIN
Vivado Repository				
Alliance Partners				
Automotive & Industrial				
AXI Infrastructure				
BaseIP				
Basic				
Communication & Networking				
Debug & Verification				
Digital Signal Processing				
Embedded Processing				
FPGA Features and Design				
Cloning				
Cloning Wizard	AXI4	Production	Included	xilinx
IO Interfaces				
Soft Error Mitigation				
XADC				
Math Functions				
Memories & Storage Elements				
Partial Reconfiguration				

① [FPGA Features and Design]の+をクリックして展開

② [Cloning]の+をクリックして展開

③ [Cloning Wizard]をダブルクリック

Clock options 設定画面



② [Component Name]に sys_mmcmと入力

① [Clock Options]タブが選択されていることを確認

③ MMCMを選択

④ Frequency synthesisと Phase Alignmentにチェック

⑤ Balancedを選択

OKはまだクリックしない！！

Input Clock	Input Frequency(MHz)	Jitter Options	Input Jitter	Source	
Primary	100.000	10.000 - 800.000	UI	0.010	Single ended clock capable pin
Secondary	100.000	60.000 - 120.000		0.010	Single ended clock capable ...

Output Clocks設定画面

Customize IP
Clocking Wizard (5.3)

Documentation IP Location Switch to Defaults

IP Symbol Resource

Component Name: xilinx_mmcmm

Clocking Options **Output Clocks** MMCM Settings Port Renaming Summary

The phase is calculated relative to the active input clock.

Output Clock	Output Freq (MHz)		Phase (degrees)		Duty Cycle (%)	
	Requested	Actual	Requested	Actual	Requested	Actual
<input checked="" type="checkbox"/> clk_out1	100.000	100.000	0.000	0.000	50.000	50.0
<input checked="" type="checkbox"/> clk_out2	200	200.000	0.000	0.000	50.000	50.0
<input type="checkbox"/> clk_out3	100.000	N/A	0.000	N/A	50.000	N/A
<input type="checkbox"/> clk_out4	100.000	N/A	0.000	N/A	50.000	N/A
<input type="checkbox"/> clk_out5	100.000	N/A	0.000	N/A	50.000	N/A
<input type="checkbox"/> clk_out6	100.000	N/A	0.000	N/A	50.000	N/A
<input type="checkbox"/> clk_out7	100.000	N/A	0.000	N/A	50.000	N/A

USE CLOCK SEQUENCING

Output Clock Sequence Number

Output Clock	Sequence Number
clk_out1	1
clk_out2	1
clk_out3	1
clk_out4	1
clk_out5	1
clk_out6	1
clk_out7	1

Enable Optional Inputs / Outputs

reset power_down input_clk_stopped

Reset Type

Automatic Control On-Chip Automatic Control Off-Chip User-Controlled On-Chip User-Controlled Off-Chip

Clocking Feedback

Source Signaling

Single-ended Differential

① [Output Clocks]タブを選択

② [clk_out2]にチェックを入れる

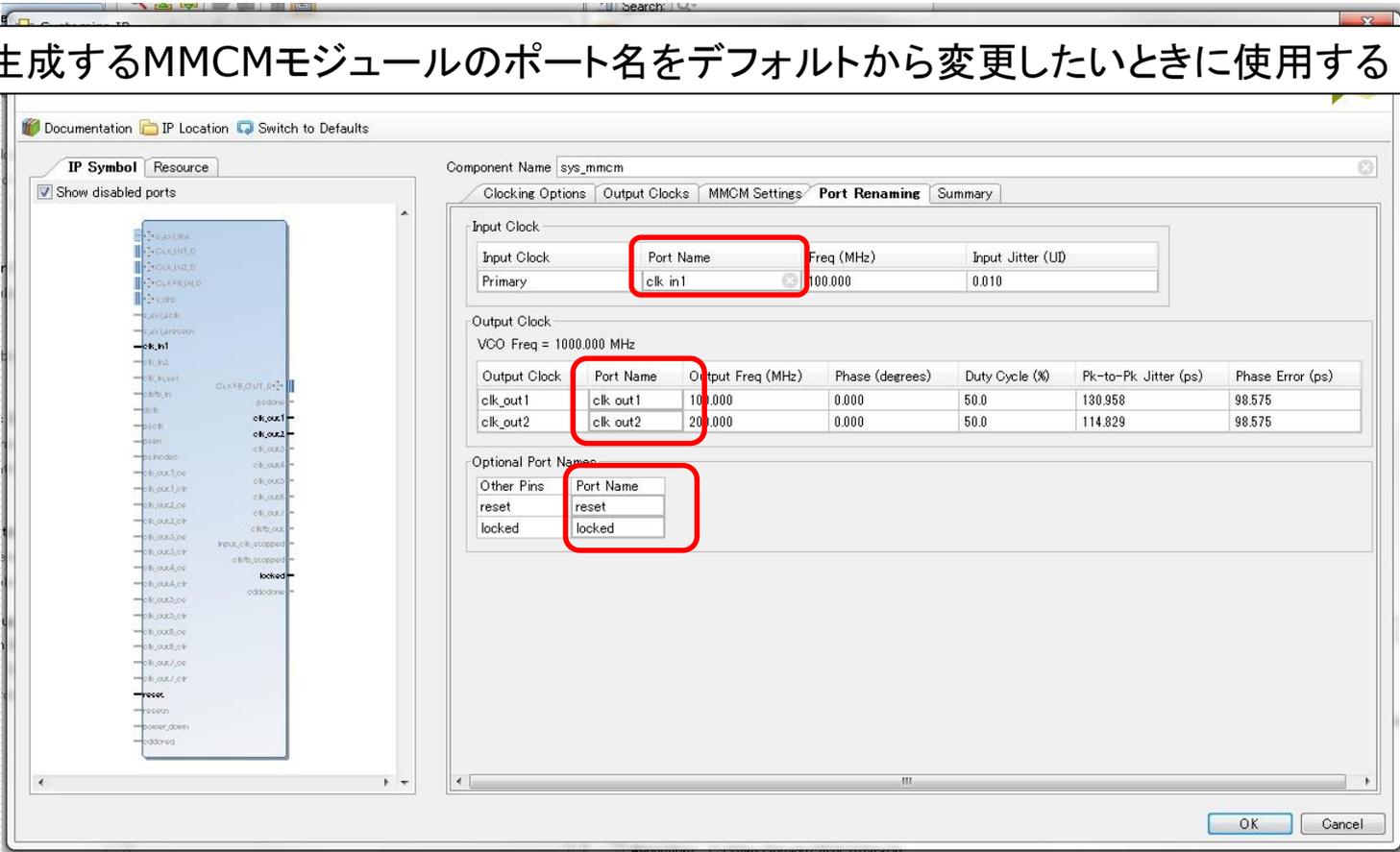
③ 200MHzを生成するので200と入力

④ OKをクリック

OK Cancel

補足: Port Renaming設定

生成するMMCMモジュールのポート名をデフォルトから変更したいときに使用する



Component Name: sys_mmcm

Port Renaming Configuration:

Input Clock

Input Clock	Port Name	Freq (MHz)	Input Jitter (UD)
Primary	clk_in1	100.000	0.010

Output Clock

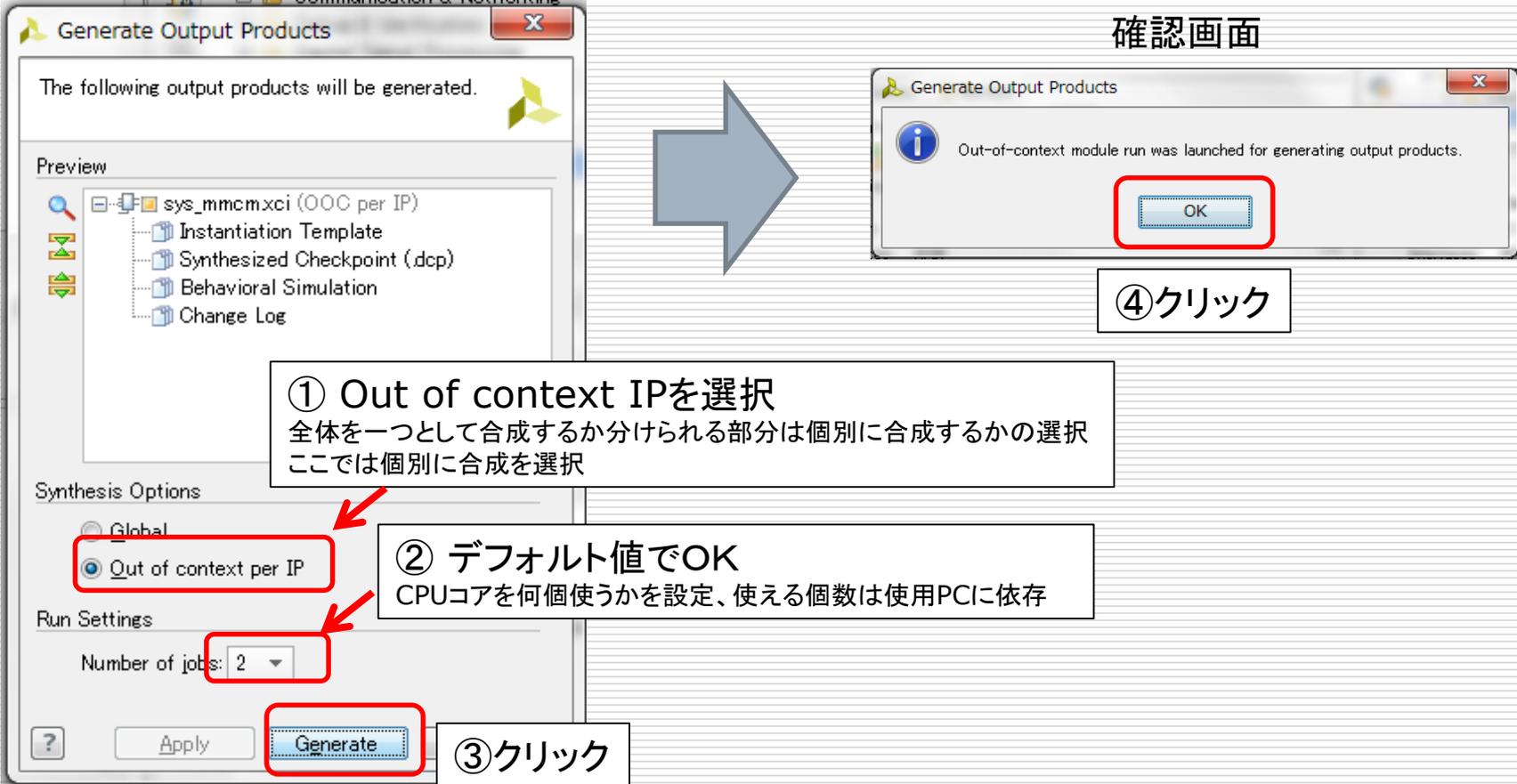
VCO Freq = 1000.000 MHz

Output Clock	Port Name	Output Freq (MHz)	Phase (degrees)	Duty Cycle (%)	Pk-to-Pk Jitter (ps)	Phase Error (ps)
clk_out1	clk_out1	100.000	0.000	50.0	130.958	98.575
clk_out2	clk_out2	200.000	0.000	50.0	114.829	98.575

Optional Port Names

Other Pins	Port Name
reset	reset
locked	locked

確認画面



The following output products will be generated.

Preview

- sys_mmcmxc1 (OOC per IP)
 - Instantiation Template
 - Synthesized Checkpoint (.dcp)
 - Behavioral Simulation
 - Change Log

Synthesis Options

Global

Out of context per IP

Run Settings

Number of jobs: 2

Generate

確認画面

Out-of-context module run was launched for generating output products.

OK

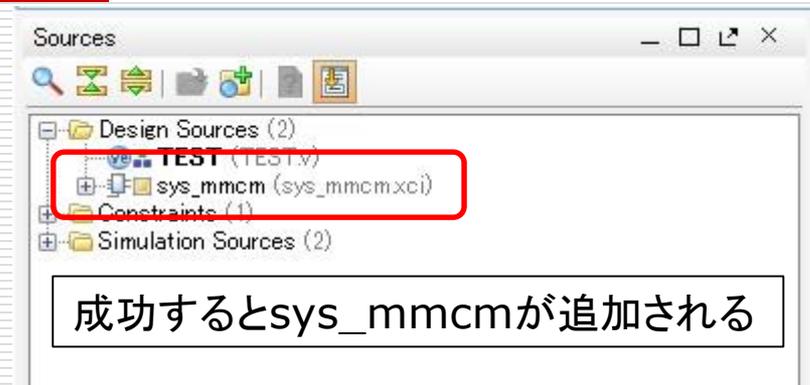
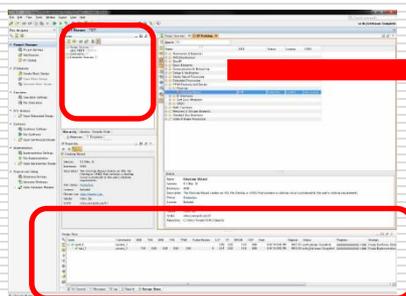
① Out of context IPを選択
全体を一つとして合成するか分けられる部分は個別に合成するかを選択
ここでは個別に合成を選択

② デフォルト値でOK
CPUコアを何個使うかを設定、使える個数は使用PCIに依存

③ クリック

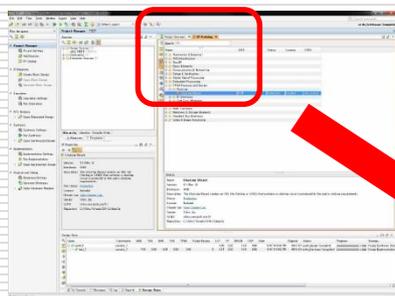
④ クリック

生成の進捗と結果



Sys_mmcm_syhth1の左に✓が表示されたら生成成功(○が回っている時は生成中)

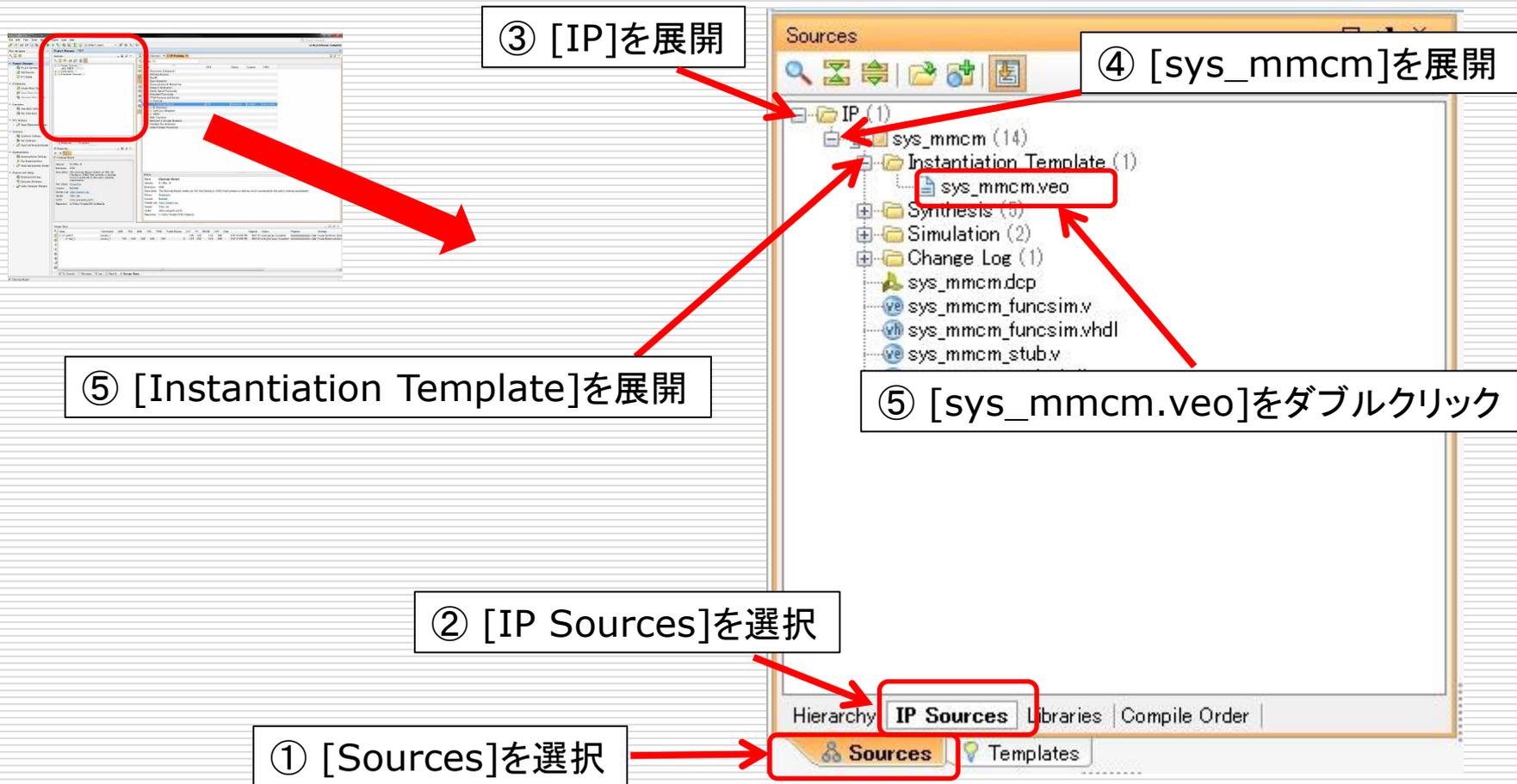
IP Catalogを閉じる



TEST.vへの組み込み

- 自動生成せれるMMCM(sys_mmcm)のテンプレートを用いて組み込んでみます

テンプレートを開く



① [Sources]を選択

② [IP Sources]を選択

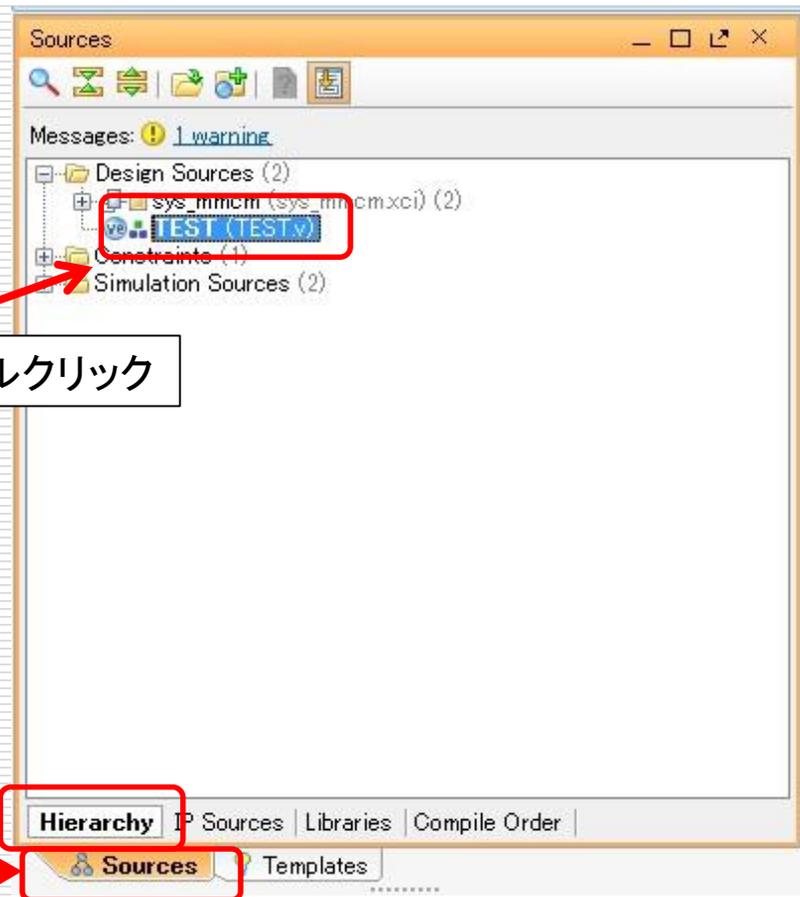
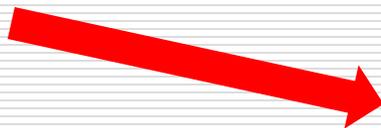
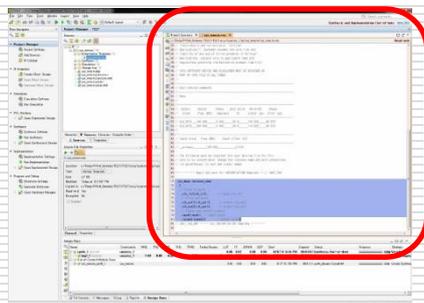
③ [IP]を展開

④ [sys_mmcm]を展開

⑤ [sys_mmcm.veo]をダブルクリック

⑤ [Instantiation Template]を展開

TEST.vを開く

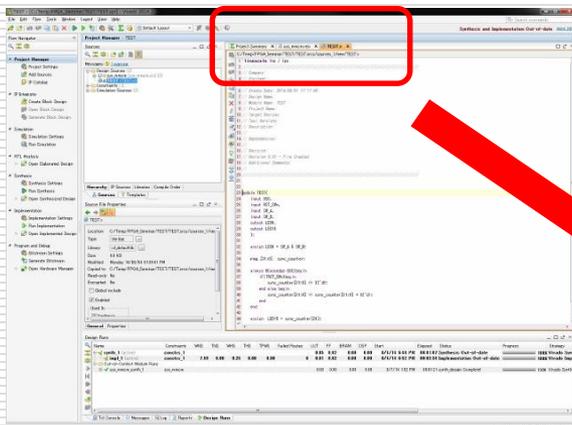


③ [TEST]をダブルクリック

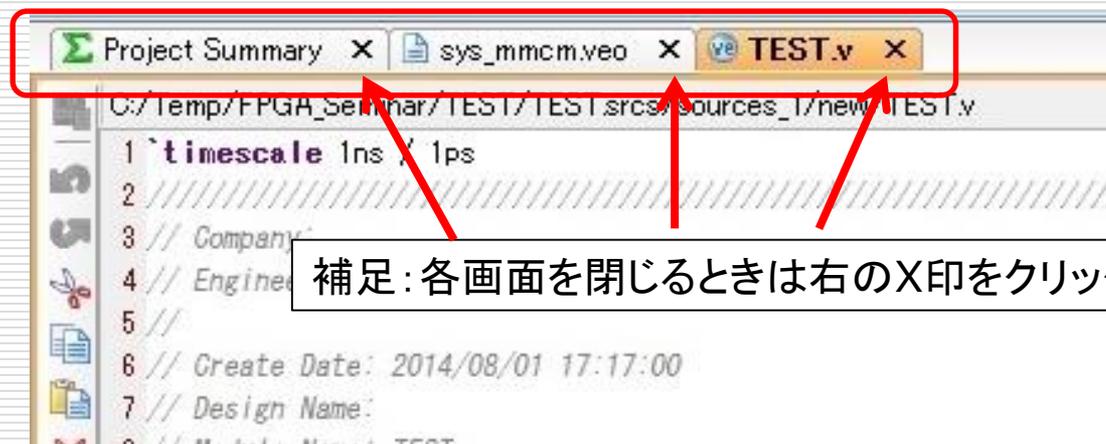
② [Hierarchy]を選択

① [Sources]を選択

画面の切り替え

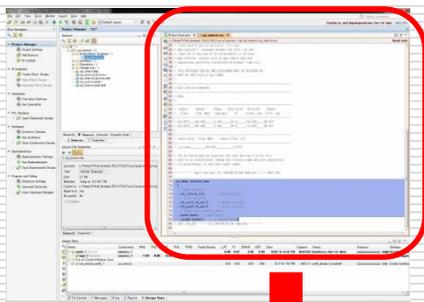


このタブで画面切り替え



補足: 各画面を閉じるときは右のX印をクリック

テンプレートの表示とコピー



① タブを選択しsys_mmcm.veoを表示させてください

② Begin Cut here for INSTANTIATION templateの下から
INST_TAG_END ----- End INSTANTIATION Templateの前まで
選択コピー



```

69 //----- Begin Cut here for INSTANTIATION Template -----// INST_TAG
70
71 sys_mmcm instance_name
72 (
73     // Clock in ports
74     .clk_in1(clk_in1),      // input clk_in1
75     // Clock out ports
76     .clk_out1(clk_out1),   // output clk_out1
77     .clk_out2(clk_out2),   // output clk_out2
78     // Status and control signals
79     .reset(reset), // input reset
80     .locked(locked);      // output locked
81 // INST_TAG_END ----- End INSTANTIATION Template -----
82

```

テンプレートの挿入

① タブを選択し画面をTEST.vに切り替えてください

```
22
23 module TEST(
24     input OSC,
25     input RST_SWn,
26     input SW_A,
27     input SW_B,
28     output LED0,
29     output LED15
30 );
31
32 assign LED0 = SW_A & SW_B;
33
34
35     .CLK      (OSC),
36     .RST     (RST_SWn),
37     .LED     (LED15)
38 );
39
40 endmodule
```

② ポートリストの下をクリックして、カーソルを移動させる

③ ペーストすると先ほどのテンプレートが挿入される

変更箇所

```
23 module TEST(  
24     input OSC,  
25     input RST_SWn,  
26     input SW_A,  
27     input SW_B,  
28     output LED0,  
29     output LED15  
30 );  
31 sys_mmcm instance_name  
32 (  
33     // Clock in ports  
34     .clk_in1(clk_in1),      // input clk_in1  
35     // Clock out ports  
36     .clk_out1(clk_out1),   // output clk_out1  
37     .clk_out2(clk_out2),   // output clk_out2  
38     // Status and control signals  
39     .reset(reset), // input reset  
40     .locked(locked)); // output locked  
41 assign LED0 = SW_A & SW_B;  
42
```

これから以下の修正を順に説明します

1. sys_mmcmに入カクロックとリセット信号を接続
2. 新しく生成した200MHzクロックをカウンターに接続
3. カウンターのリセットにMMCMのlockedを接続
 - 出カクロック周期が補償されないことを回避する。

locked信号:

出カクロック(200MHzクロック)が正しく生成されたら1(High)になる信号

TEST.vの修正 1/2

```

29  output LED15
30  );
31
32  wire mmcm_reset;
33  assign mmcm_reset = ~RST_SWn;
34
35  wire CLK200M;
36  wire sys_reset;
37
38  sys_mmcm mmcm_1
39  (
40  // Clock in ports
41  .clk_in1(OSC), // input clk_in1
42  // Clock out ports
43  .clk_out1(), // output clk_out1
44  .clk_out2(CLK200M) // output clk_out2
45  // Status and control signals
46  .reset(mmcm_reset), // input reset
47  .locked(sys_reset)); // output locked
48

```

MMCMのリセット入力は正論理のため反転した信号を作る

RST_SW_nは負論理(リセットの時にLになる)信号

新しく使用する信号を定義

インスタンス名をmmcm_1に書き換える

clk_in1からOSCに書き換え

clk_out1は使用しないので削除

clk_out2からCLK200Mに書き換え

resetからmmcm_resetに書き換え

lockedからsys_resetに書き換え

TEST.vの修正 2/2

OSC → CLK200M

```
50 TEST_SYNC_COUNTER_U1(  
51     .CLK      (CLK200M),  
52     .RSTn    (sys_reset),  
53     .LED     (LED15)  
54 );
```

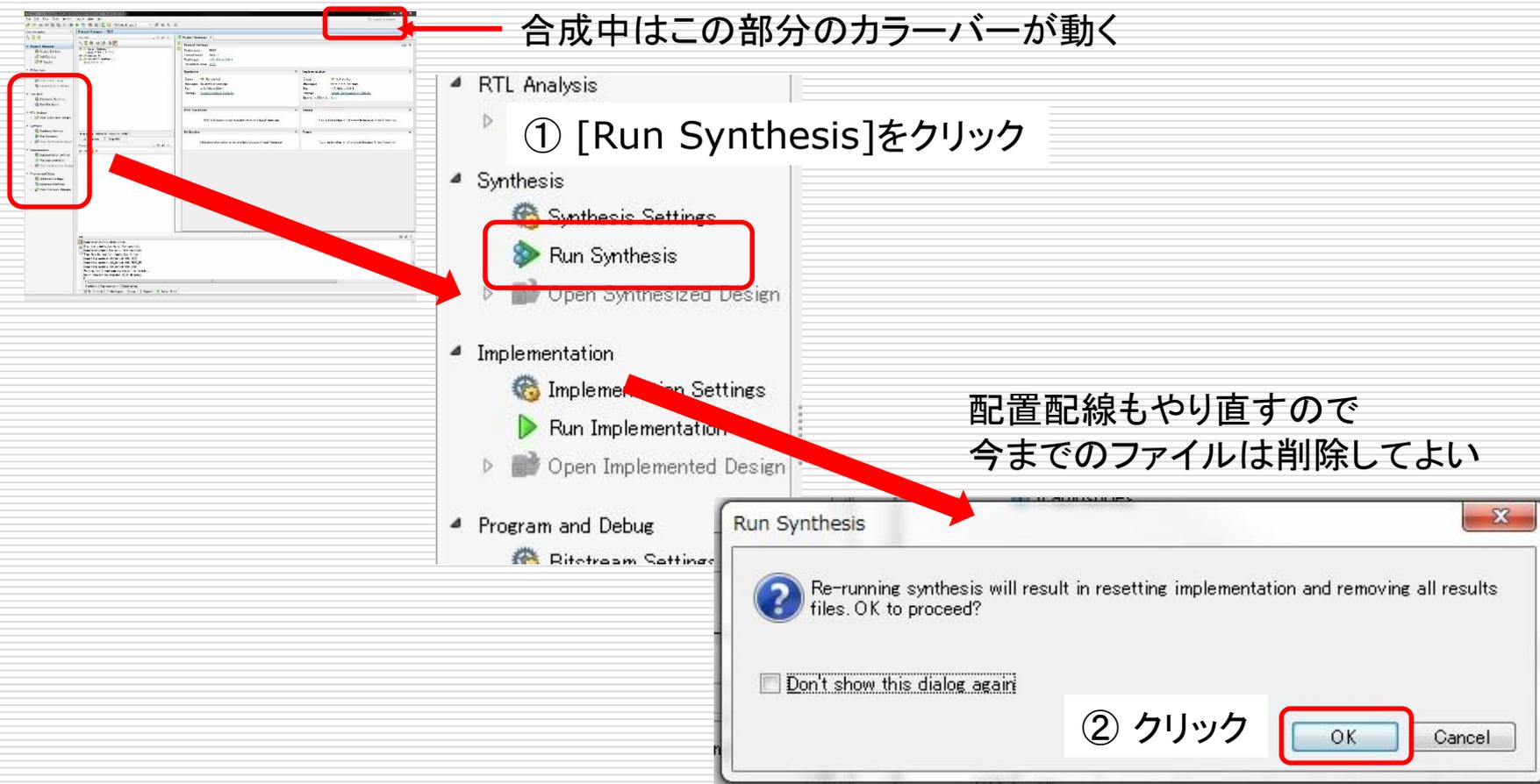
RST_SWn → sys_reset

```
55
```

```
56 endmodule
```

修正が終わったらセーブしてTEST.vとsys_mmcm.vcoを閉じてください。
終わった人は論理合成→配置配線とFPGAデータ書き込みまで進んでください

論理合成開始



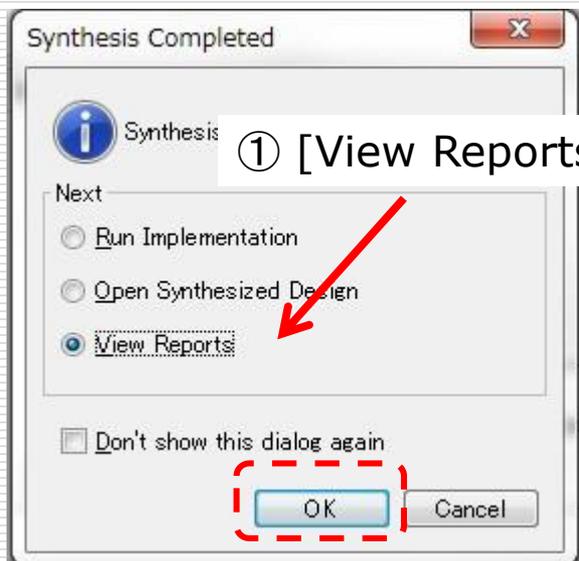
合成中はこの部分のカラーバーが動く

① [Run Synthesis]をクリック

配置配線もやり直すので
今までのファイルは削除してよい

② クリック

論理合成結果の確認



① [View Reports]を選択

②クリック

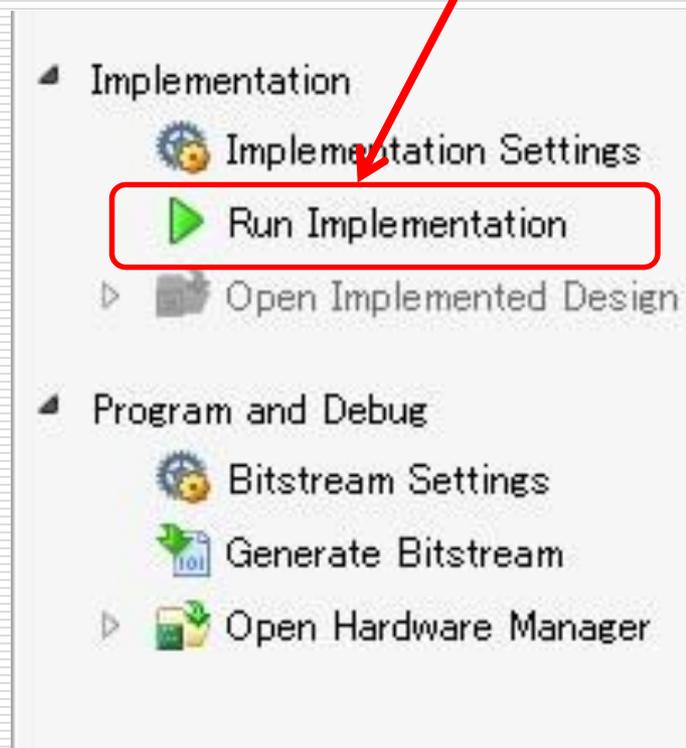
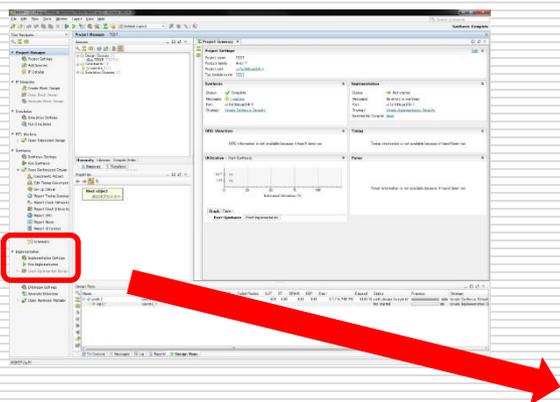
③エラーが無いことを確認



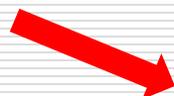
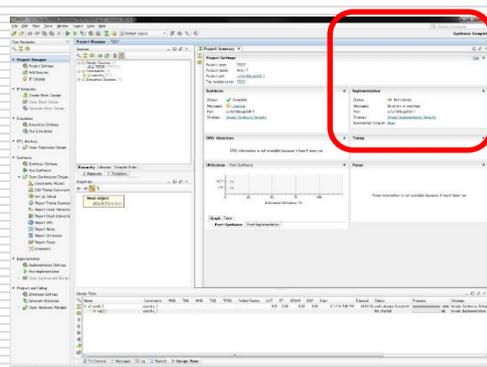
Project summaryページのSynthesis項目

配置配線の実行

[Run Implementation]をクリック



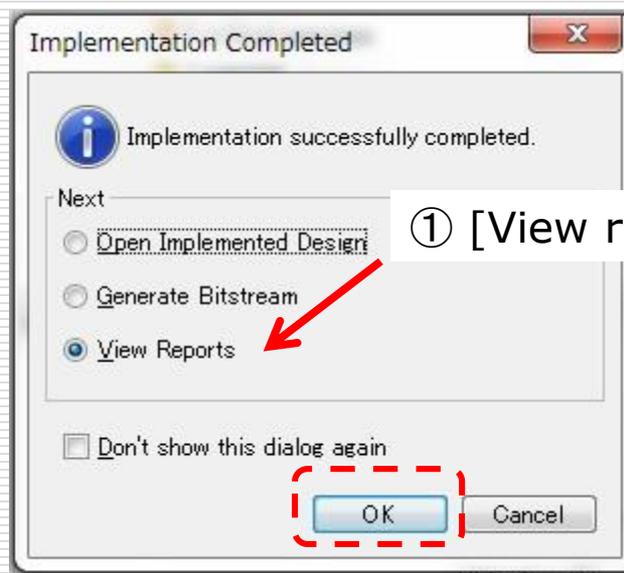
配置配線中の画面



StatusがRunningになる

配置配線結果の確認

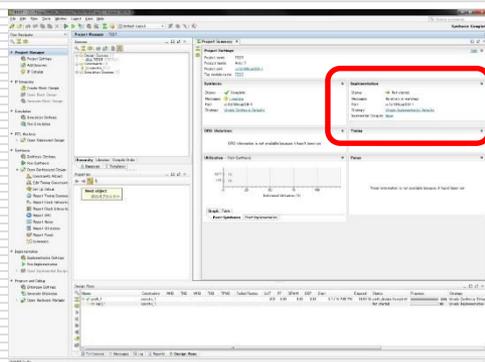
配置配線が終わると表示される



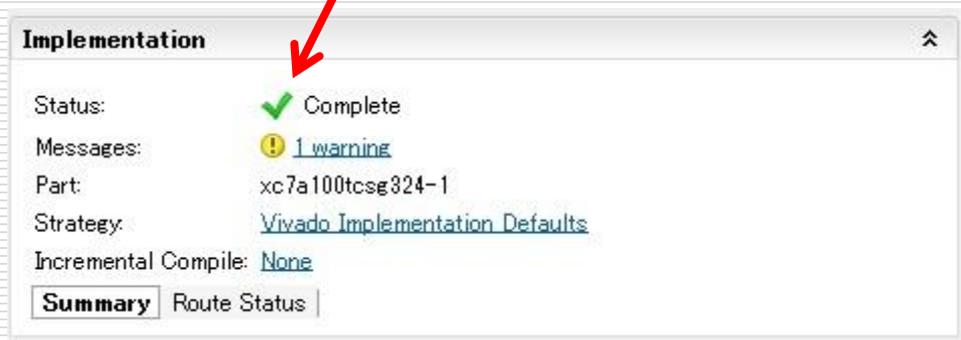
① [View reports]を選択

②クリック

配置配線結果を確認



Completeになっていることを確認



Implementation

Status:  Complete

Messages:  [1 warning](#)

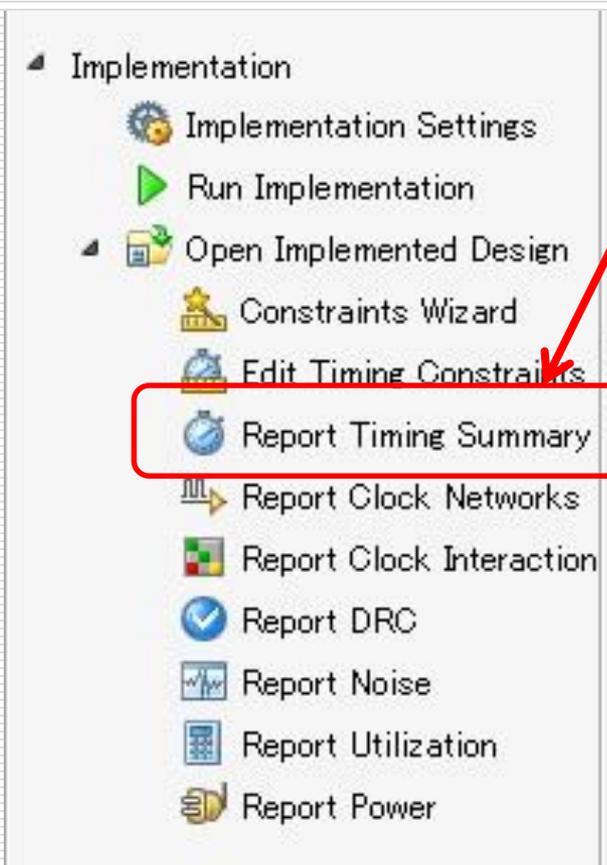
Part: [xc7a100tcsq324-1](#)

Strategy: [Vivado Implementation Defaults](#)

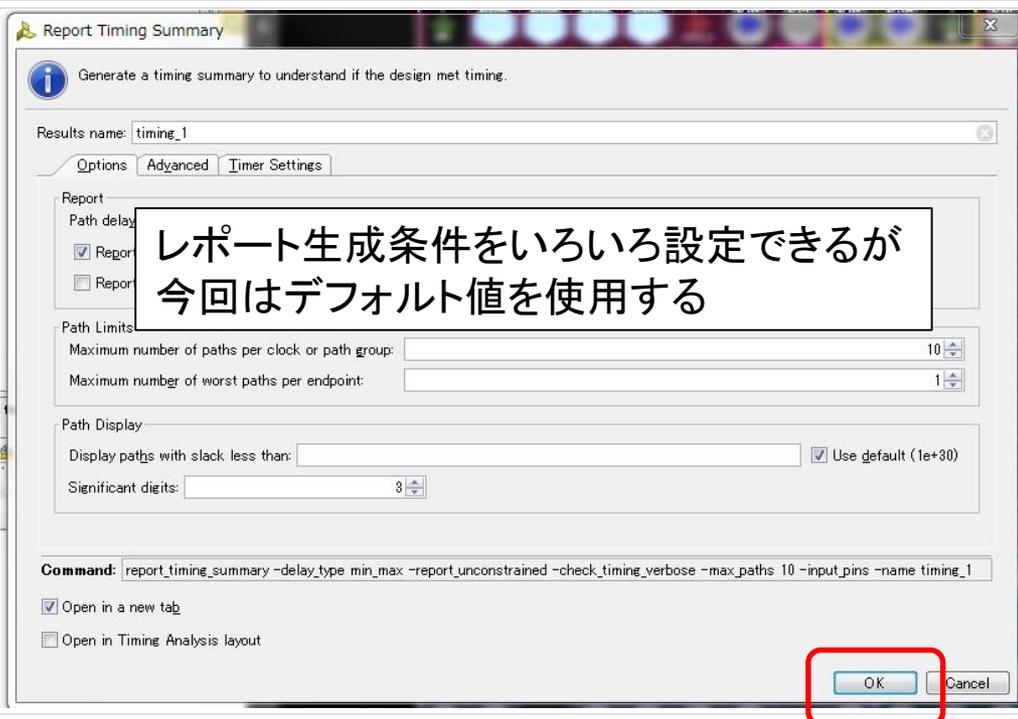
Incremental Compile: [None](#)

Summary Route Status

タイミングレポートの生成

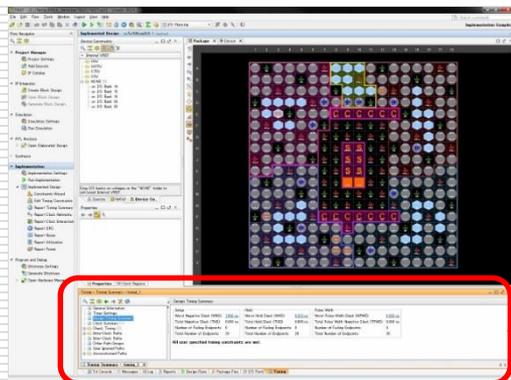


① [Report Timing Summary]をクリック



② [OK]をクリック

Timing summaryの確認



Timing - Timing Summary - timing_1

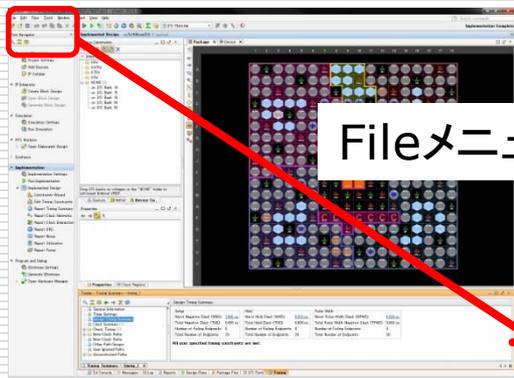
- General Information
- Timer Settings
- Design Timing Summary**
- Clock Summary (3)
- Check Timing (1)
- Intra-Clock Paths
- Inter-Clock Paths
- Other Path Groups
- User Ignored Paths

Design Timing Summary		
Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 2.383 ns	Worst Hold Slack (WHS): 0.254 ns	Worst Pulse Width Slack (WPWS): 2.000 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 20	Total Number of Endpoints: 29	Total Number of Endpoints: 35

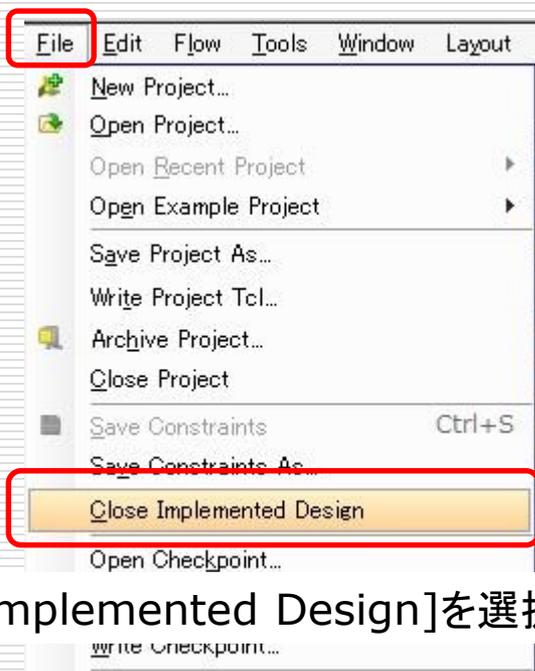
All user specified timing constraints are met.

全ての制約が満たされていることを確認

Implemented designを閉じる



Fileメニュー

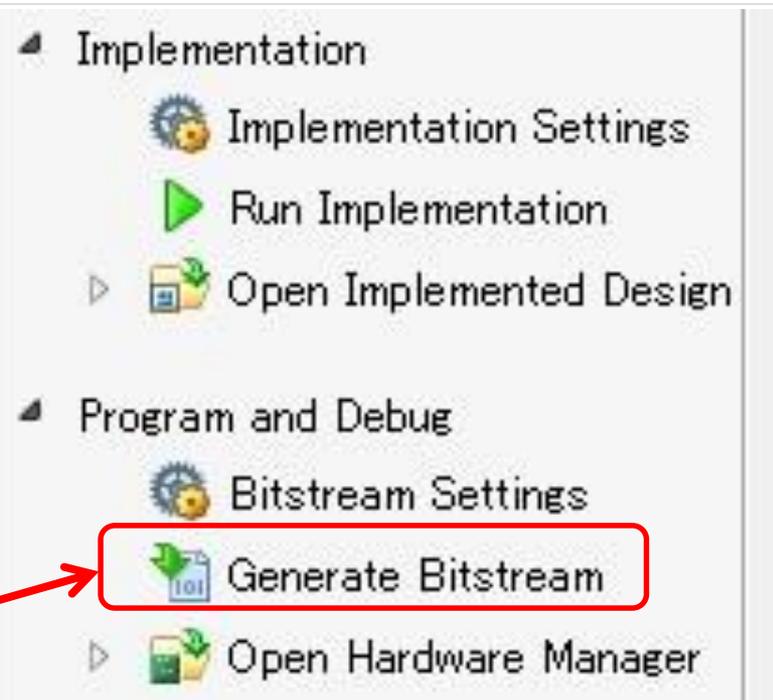
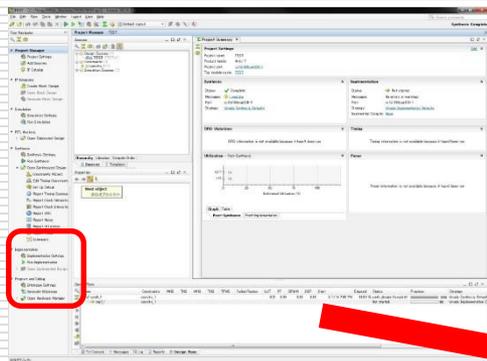


[Close Implemented Design]を選択



OKをクリック

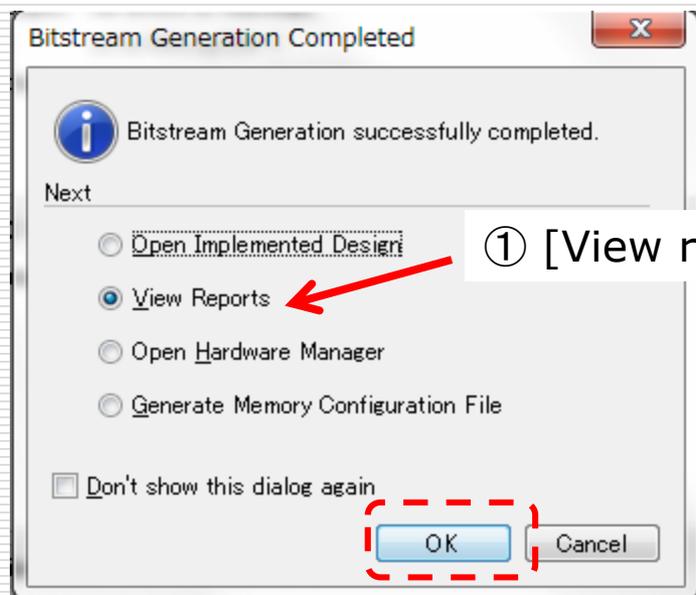
デーファイル生成



[Generate Bitstream]をクリック

結果確認

データ生成が終わると表示される



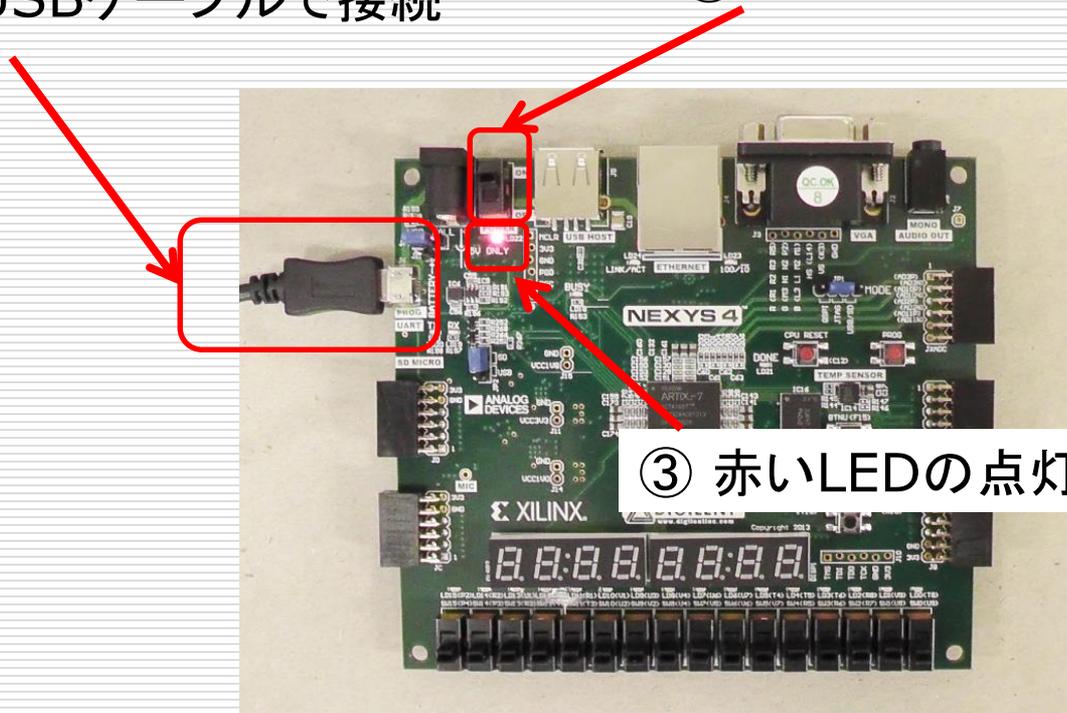
① [View reports]を選択

②クリック

ハードウェアの準備

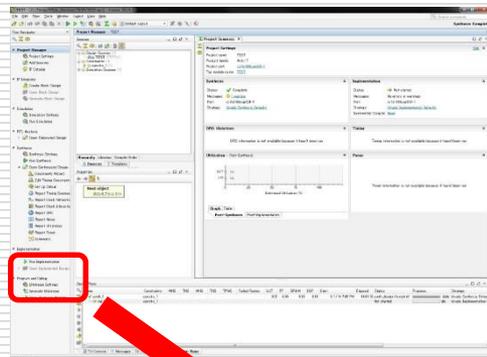
① PCとUSBケーブルで接続

② POWER SWをON

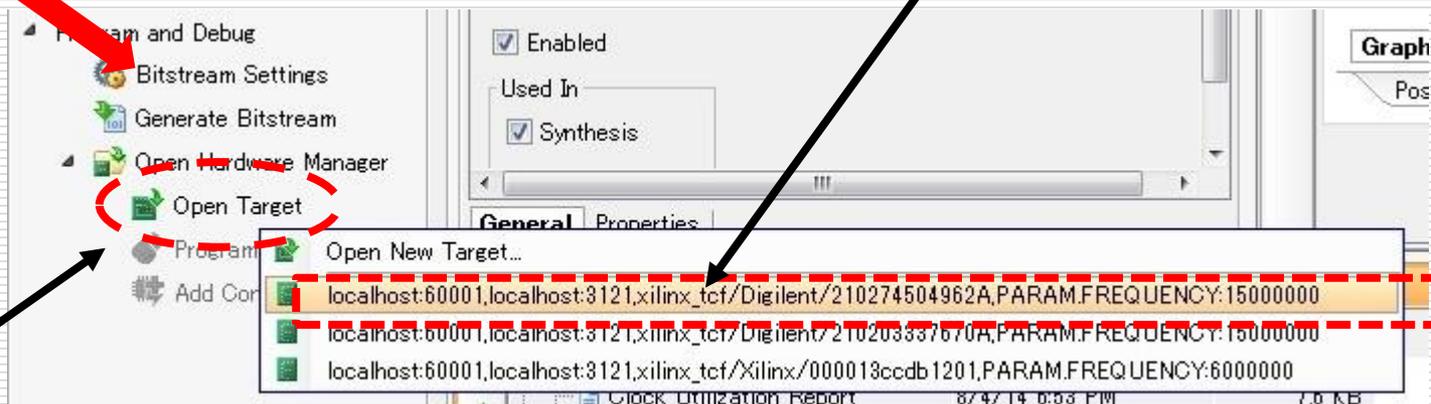


③ 赤いLEDの点灯確認

ダウンロードツールの起動

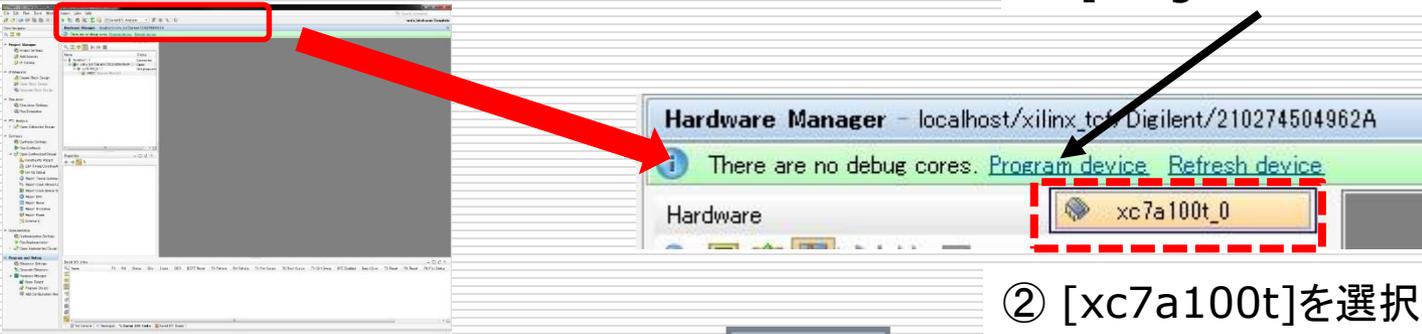


② [*/Digilent/*]を選択



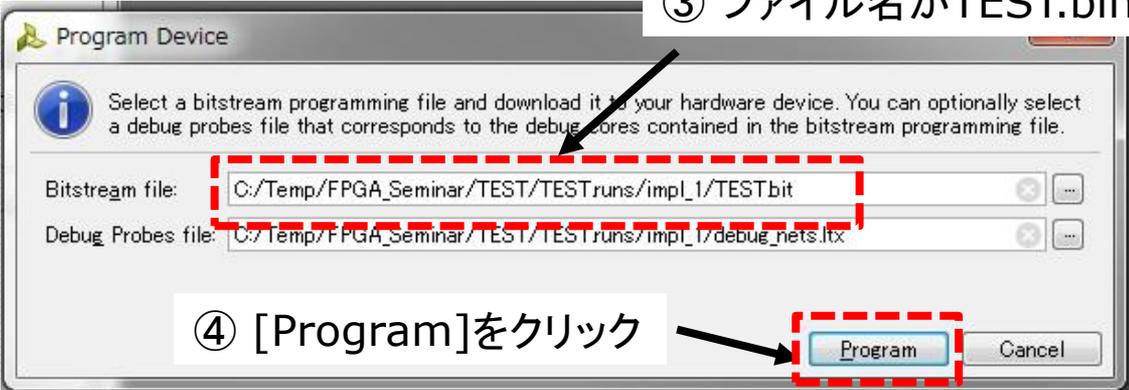
① [Open Target]を左クリック

書き込み



① [Program device]をクリック

② [xc7a100t]を選択

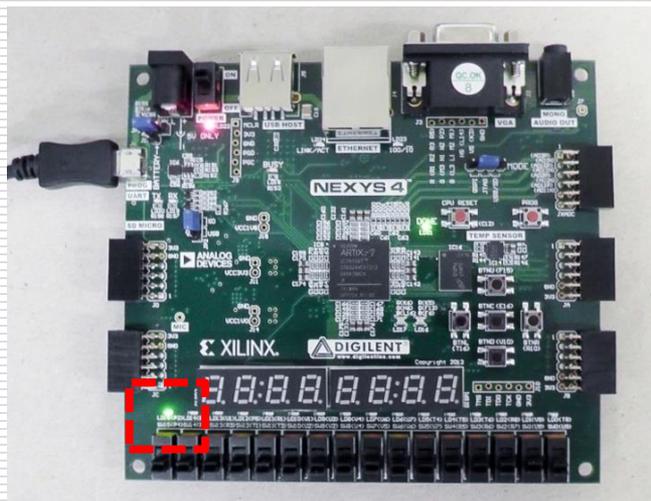
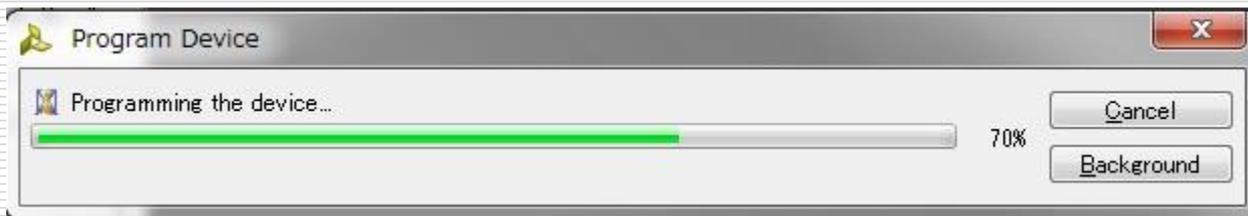


③ ファイル名がTEST.binであることを確認

④ [Program]をクリック

データ書き込みと動作確認

書き込み中の表示



緑色LEDの点滅周期が半分になりましたか？

履歴

- 2012/5/17 第1.0版 ISE13.4対応 内田智久(Esys, KEK/総研大), 林達也(大阪大学)
- 2014/8/7 第2.0版 Vivado2014対応、章構成変更 内田智久(Esys, KEK/総研大)
- 2015/7/31 第3.0版 Vivado2015対応、章構成変更 内田智久(Esys, KEK/総研大)
- 2015/12/05 第3.1版 不要ページ削除 内田智久(Esys, KEK/総研大)
- 2016/6/22 第3.2版 Vivado2016.2対応 内田智久(Esys, KEK/総研大)