

[オンラインセミナー]

# インテル® Quartus® Prime を Tips で使いこなそう

<Standard Edition 編>

Rev.4

Altima Company | A Macnica Division Company

Co.Tomorrowing  
**MACNICA**

©Macnica,Inc.

# このセミナーで紹介する Tips

<p>FPGA 型番の 置き換え検討時に 便利な機能</p>	<p>自社デザインの ソースコードを 開示せずに納品先に 提供するワザ</p>	<p>ステートマシン回路 を作成時に 知っておきたい オプション</p>
<p>レジスターや 組み合わせノードを 保持する方法</p>	<p><b>Quartus Prime Tips</b> &lt;Standard Edition 編&gt;</p>	<p>QSF ファイルを カテゴリーで ソート表示させる方法</p>
<p>UI フォントサイズを 変更して 見やすくしよう</p>	<p>コンパイルレポートを HTML 出力して 見やすくしよう</p>	

<p><b>FPGA 型番の 置き換え検討時に 便利な機能</b></p>	<p>自社デザインの ソースコードを 開示せずに納品先に 提供するワザ</p>	<p>ステートマシン回路 を作成時に 知っておきたい オプション</p>
<p>レジスターや 組み合わせノードを 保持する方法</p>	<p><b>Quartus Prime Tips</b> &lt;Standard Edition 編&gt;</p>	<p>QSF ファイルを カテゴリーで ソート表示させる方法</p>
<p>UI フォントサイズを 変更して 見やすくしよう</p>	<p>コンパイルレポートを HTML 出力して 見やすくしよう</p>	

# FPGA 型番の置き換えを検討

- インテル® FPGA は、同一ファミリー&同一パッケージ内でデバイス・マイグレーションに対応

- 専用ピン, JTAG ピン, コンフィグレーション・ピンのレイアウトは同じ
- 電源ピンもほぼ同じレイアウト
  - ただし、FPGA の集積度に応じて電源のピン数は変わる
  - 該当するパッケージの計画最大集積度に対応してレイアウトする
- I/O ピンは、計画されるすべての集積度に対して照合が必要

- マイグレーション可能なラインナップをドキュメントでチェック

- 各デバイスの Device Overview

Figure 7. Vertical Migration Capability Across Cyclone V Device Packages and Densities

The arrows indicate the vertical migration paths. The devices included in each vertical migration path are shaded. You can also migrate your design across device densities in the same package option if the devices have the same dedicated pins, configuration pins, and power pins.

Variant	Member Code	Package										
		M301	M383	M484	F256	U324	U484	F484	U672	F672	F896	F1152
Cyclone V E	A2		↕		↕	↕	↕	↕				
	A4		↕		↕	↕	↕	↕				
	A5		↕									
	A7								↕	↕		
	A9								↕	↕	↕	
Cyclone V GX	C3						↕	↕				
	C4	↕	↕						↕			
	C5	↕	↕									
	C7										↕	
	C9										↕	↕
Cyclone V GT	D5						↕	↕		↕	↕	
	D7										↕	↕

# FPGA 型番の置き換え検討時の“つまずき”

どれ位の  
回路規模になるか  
わからないし、  
試作品は大きめの型番  
が良いよね。



量産のときは  
コスト削減のためにも  
規模を下げた型番だよなあ。

試作のボードを作り直さずに  
量産でもそのまま  
使いたいなあ。

## ● 一般的にやること

メーカーの Web サイトから  
各型番のピンアウトファイルをダウンロードして  
ユーザー自ら各ピン番号を比較



やるのは 正直ちょっと面倒



Bank Number	VREF	Pin Name/Function	Optional Function(s)	Configuration Function	Emulated LVDS Output Channel	FE84
B1	VREFB1N0	IO			DIFFIO_L1p	G4
B1	VREFB1N0	IO			DIFFIO_L1n	G3
B1	VREFB1N0	IO			DIFFIO_L2p	B2
B1	VREFB1N0	IO			DIFFIO_L2n	B1
B1	VREFB1N0	IO	VREFB1N0			G5
B1	VREFB1N0	IO			DIFFIO_L3p	E4
B1	VREFB1N0	IO			DIFFIO_L3n	E3
B1	VREFB1N0	IO			DIFFIO_L6p	D2
B1	VREFB1N0	IO		DATA1_ASD0	DIFFIO_L6n	D1

# FPGA 型番の置き換え検討時に “便利な機能”

- そんなときは？

Quartus Prime のマイグレーション機能を活用しよう！



# Quartus Prime の操作 [1/2]

## ステップ 1 Migration Devices ダイアログボックスで型番を指定

- ① Device ダイアログボックス内の Migration Devices ボタンをクリック

現在のプロジェクトで  
選択しているデバイス

マイグレーション  
候補のデバイス

マイグレーションに  
指定しているデバイス

- ② マイグレーションとしてピン照合したい型番を選択

## ステップ 2 Pin Migration Window で照合

- ① Pin Planner を起動
- ② View メニュー ⇒ Pin Migration Window

Pin Number	Migration Result			Migration Devices								
	Pin Function	I/O Bank	VREF Group	SCEFA5F23C7			SCEFA7F23C7			SCEFA9F23C7		
PIN_A13	Column I/O	7A	B7A_NO	Column I/O	7A	B7A_NO	Column I/O	7A	B7A_NO	Column I/O	7A	B7A_NO
PIN_A14	Column I/O	7A	B7A_NO	Column I/O	7A	B7A_NO	Column I/O	7A	B7A_NO	Column I/O	7A	B7A_NO
PIN_A15	Column I/O	7A	B7A_NO	Column I/O	7A	B7A_NO	Column I/O	7A	B7A_NO	Column I/O	7A	B7A_NO
PIN_A16	VCCIO7A	7A		VCCIO7A	7A		VCCIO7A	7A		VCCIO7A	7A	
PIN_A17	NC			Column I/O	7A	B7A_NO	Column I/O	7A	B7A_NO	NC		
PIN_A18	NC			Column I/O	7A	B7A_NO	Column I/O	7A	B7A_NO	NC		
PIN_A19	NC			Column I/O	7A	B7A_NO	Column I/O	7A	B7A_NO	NC		
PIN_A20	NC			Column I/O	7A	B7A_NO	Column I/O	7A	B7A_NO	NC		
PIN_A21	GND			GND			GND			GND		
PIN_A22	NC			Column I/O	7A	B7A_NO	Column I/O	7A	B7A_NO	NC		
PIN_AA1	NC			NC			NC			NC		
PIN_AA2	NC			NC			NC			NC		
PIN_AA3	GND			GND			GND			GND		
PIN_AA4	GND			GND			GND			GND		
PIN_AA5	Dedica...mming	3A		Dedica...mming	3A		Dedica...mming	3A		Dedica...mming	3A	
PIN_AA6	GND			GND			GND			GND		
PIN_AA7	Column I/O	3B	B3B_NO	Column I/O	3B	B3B_NO	Column I/O	3B	B3B_NO	Column I/O	3B	B3B_NO
PIN_AA8	Column I/O	3B	B3B_NO	Column I/O	3B	B3B_NO	Column I/O	3B	B3B_NO	Column I/O	3B	B3B_NO
PIN_AA9	Column I/O	3B	B3B_NO	Column I/O	3B	B3B_NO	Column I/O	3B	B3B_NO	Column I/O	3B	B3B_NO
PIN_AA10	Column I/O	3B	B3B_NO	Column I/O	3B	B3B_NO	Column I/O	3B	B3B_NO	Column I/O	3B	B3B_NO
PIN_AA11	GND			GND			GND			GND		

- ③ 各ピンの Pin Function を照合

# Quartus Prime の操作 [2/2]

ステップ  
3

## 結果を考慮してピンレイアウトを決定

### ● Pin Migration Window の Migration Result を参考に、ピンレイアウトを作成

- Migration Result は Pin Planner の Package View 上の表示にも反映
- 差動ピンのチェックは実施されないため、必要な場合は別途ユーザーが見比べる必要あり

### ● その他

- Current Device ではユーザー I/O ピンや NC であっても、Migration Result に応じて基板上で NC, VCC, GND 処理を行う
- Migration Result が Column I/O や Row I/O 表記で、論理回路上で未使用にしたピンは、Quartus Prime 上で Unused Pin としての処理が必要

Pin Migration View

Current Device: 10CL080YF484C8G

Pin Number	Migration Result			Migration Devices								
	Pin Function	I/O Bank	VREF Group	10CL040YF484C8G			10CL080YF484C8G			10CL120YF484C8G		
				Pin Function	I/O Bank	VREF Group	Pin Function	I/O Bank	VREF Group	Pin Function	I/O Bank	VREF Group
PIN_G4	VCCINT			Row I/O	1	B1_N0	Row I/O	1	B1_N0	VCCINT		
PIN_G5	Row I/...EFB1N0	1	B1_N0	Row I/...EFB1N0	1	B1_N0	Row I/...EFB1N0	1	B1_N0	Row I/...EFB1N0	1	B1_N0
PIN_G6	VCCA3			VCCA3			VCCA3			VCCA3		
PIN_G7	GND			Column I/O	8	B8_N3	Column I/O	8	B8_N2	GND		
PIN_G8	VCCINT			Column I/O	8	B8_N3	VCCINT			VCCINT		
PIN_G9	GND			Column I/O	8	B8_N3	GND			GND		

詳細は下記ドキュメント  
をご覧ください

「Quartus Prime –  
デバイス・マイグ  
レーション」





<p>FPGA 型番の 置き換え検討時に 便利な機能</p>	<p><b>自社デザインの ソースコードを 開示せずに納品先に 提供するワザ</b></p>	<p>ステートマシン回路 を作成時に 知っておきたい オプション</p>
<p>レジスタや 組み合わせノードを 保持する方法</p>	<p><b>Quartus Prime Tips</b> &lt;Standard Edition 編&gt;</p>	<p>QSF ファイルを カテゴリーで ソート表示させる方法</p>
<p>UI フォントサイズを 変更して 見やすくしよう</p>	<p>コンパイルレポートを HTML 出力して 見やすくしよう</p>	

# 自社で作成したデザインを納品先に提供

## ● 一般的にやること

HDL のソースコードをそのまま納品



会社の設計資産を保護したい！  
でも Standard Edition では  
Pro Edition のように暗号化できないし…



## ● そんなときは？

Quartus Prime で QXP ファイルを生成しよう！

※ Lite Edition は非サポート



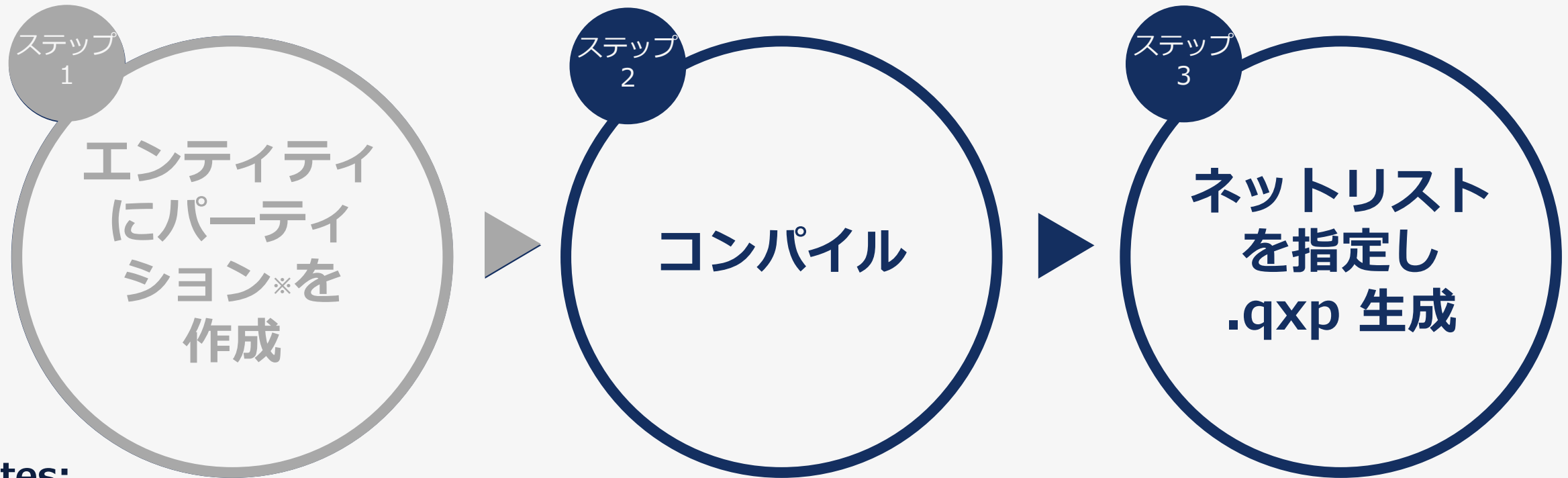
[QXP (Quartus Prime Exported Partition) ファイルとは？]

- バイナリ化されたネットリスト
- 本来は インクリメンタル・コンパイルで使用されるファイル

インクリメンタル・コンパイルって何？



# 自社デザインのソースコードを開示せずに 納品先に提供するワザ



## Notes:

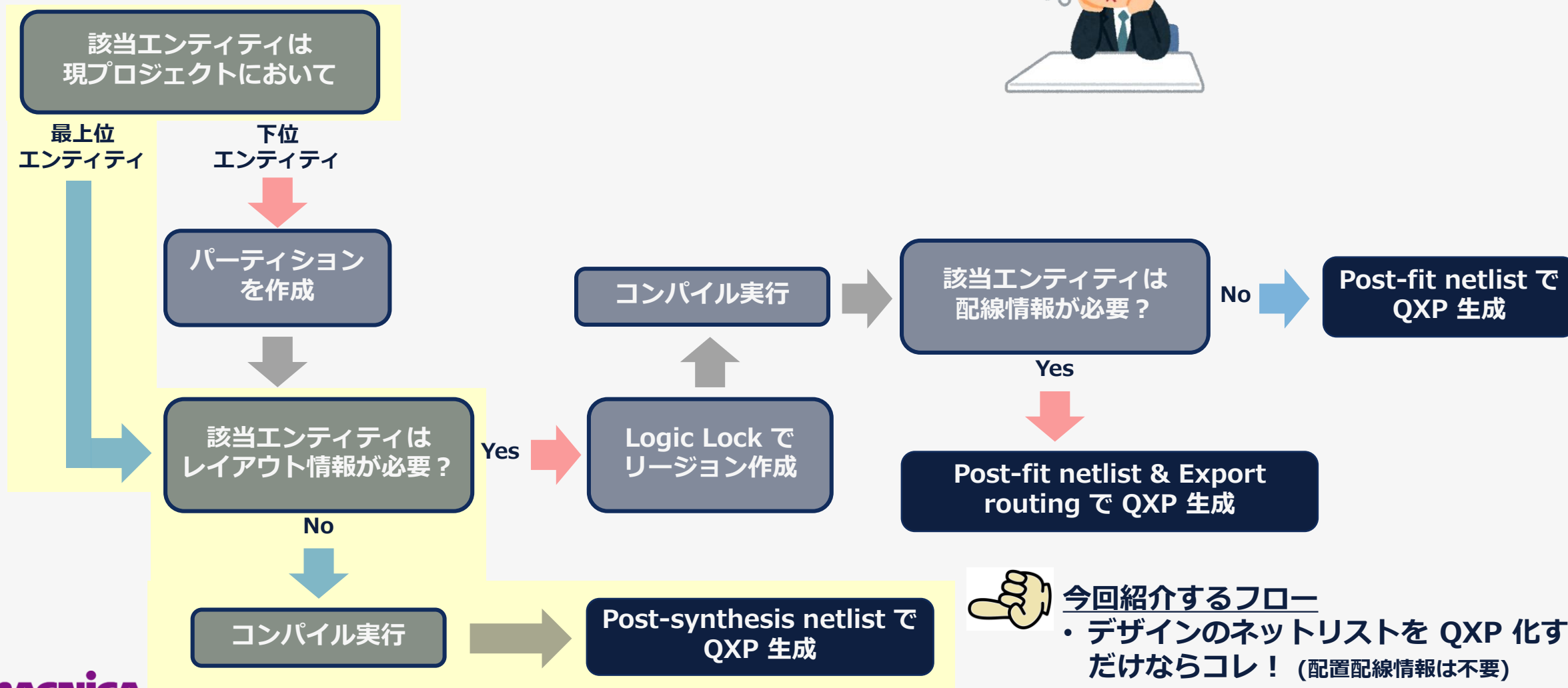
- プロジェクトの最上位エンティティを QXP 化する場合は、ステップ1 は不要
- Export 側と Import 側のプロジェクト環境を統一
  - FPGA は同じ型番
  - Quartus Prime は同じバージョン (ビルド番号を含む)

※ パーティションとは? : デザインのインスタンスに適用できる論理的な名前付きの階層境界割り当てのこと。

# QXP ファイルを作成する際のポイント

- どのような情報を QXP に含ませるか？

- QXP ファイルの生成フローが異なる



今回紹介するフロー

- ・ デザインのネットリストを QXP 化する  
 だけならコレ！ (配置配線情報は不要)

# Quartus Prime の操作 <エクスポート側>

## ステップ 3 ネットリストを指定し、.qxp 生成

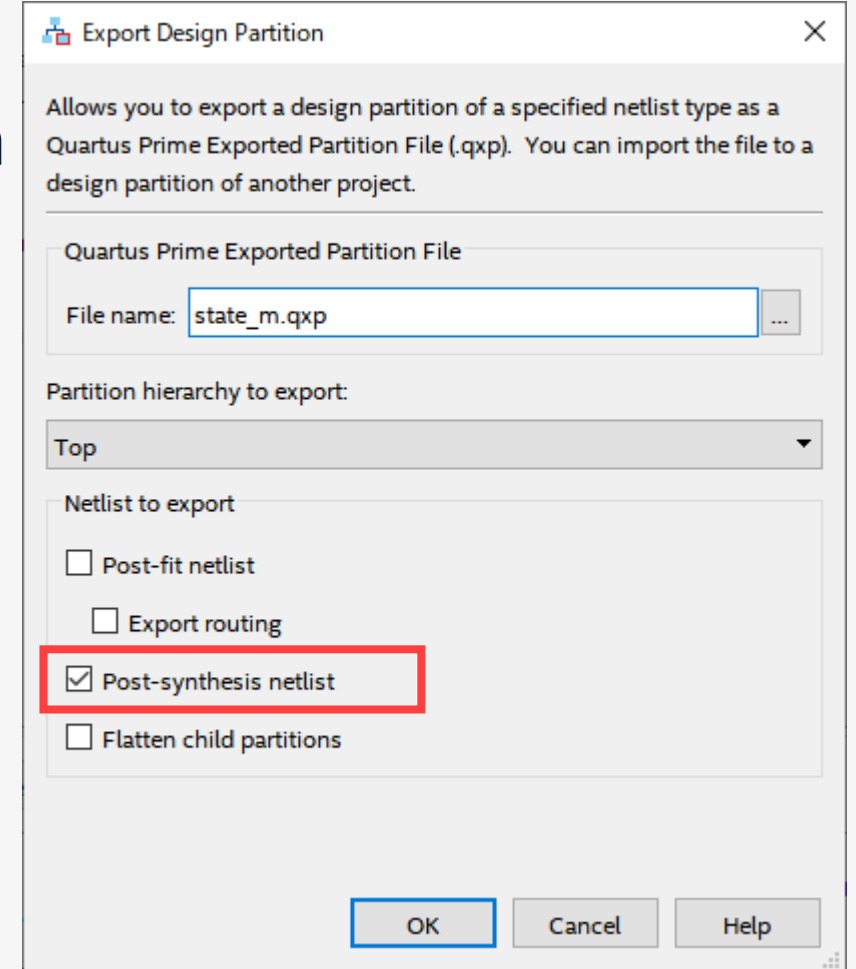
① Project メニュー ⇒ Export Design Partition

② Post-synthesis netlist = On

③ OK ボタンをクリックして QXP ファイル生成



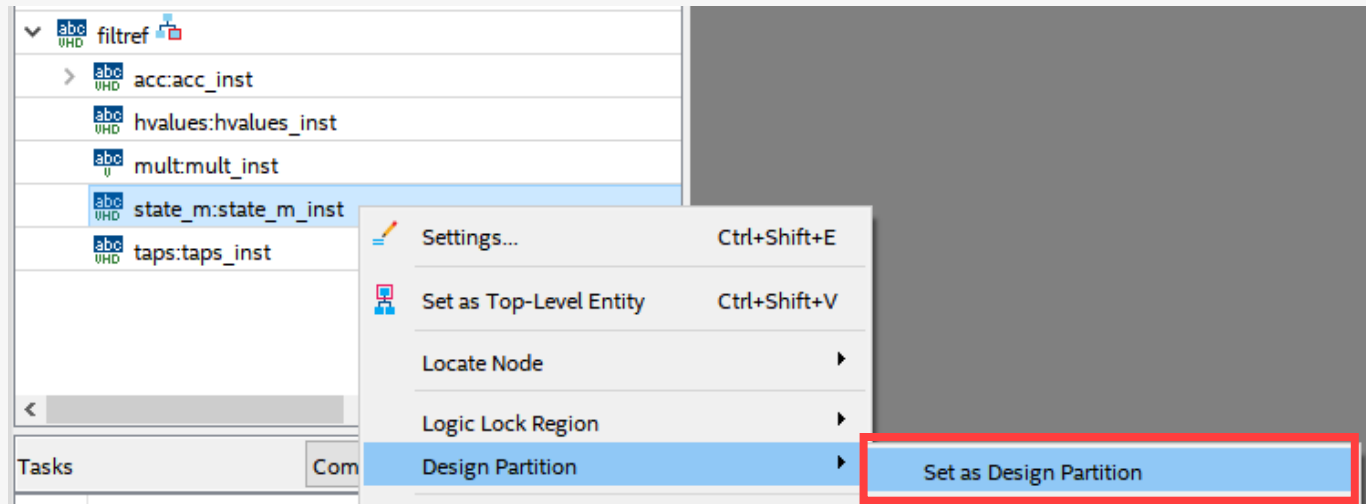
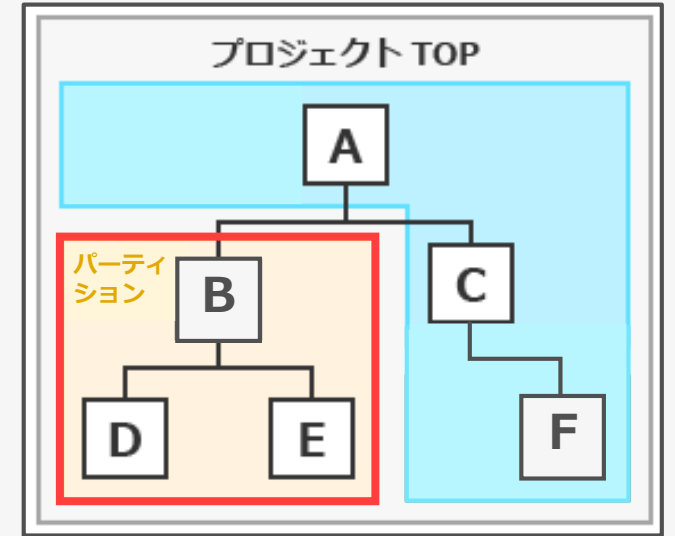
納品先へ提供



# Quartus Prime の操作 <インポート側 [1/2]>

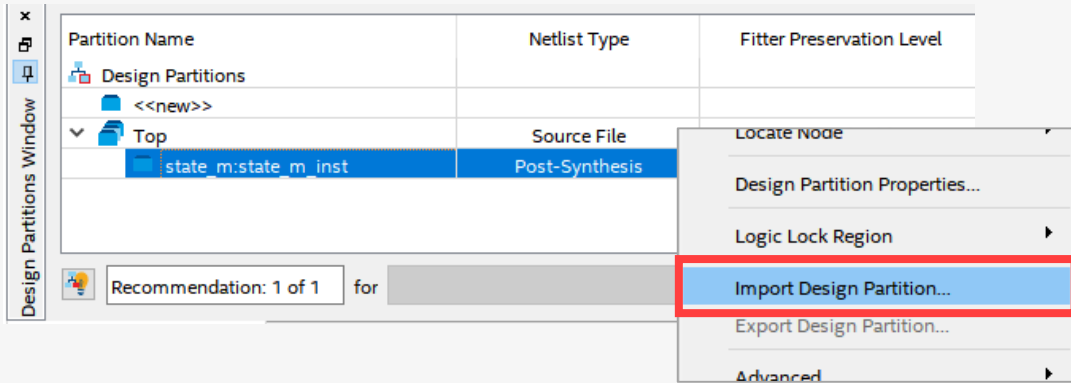
例: インポートプロジェクトにおいて  
QXP エンティティが下位エンティティの場合

- ① 該当エンティティ用のラッパーファイルを作成、プロジェクトに登録
- ② Analysis & Elaboration を実行
- ③ QXP エンティティ用のパーティションを作成

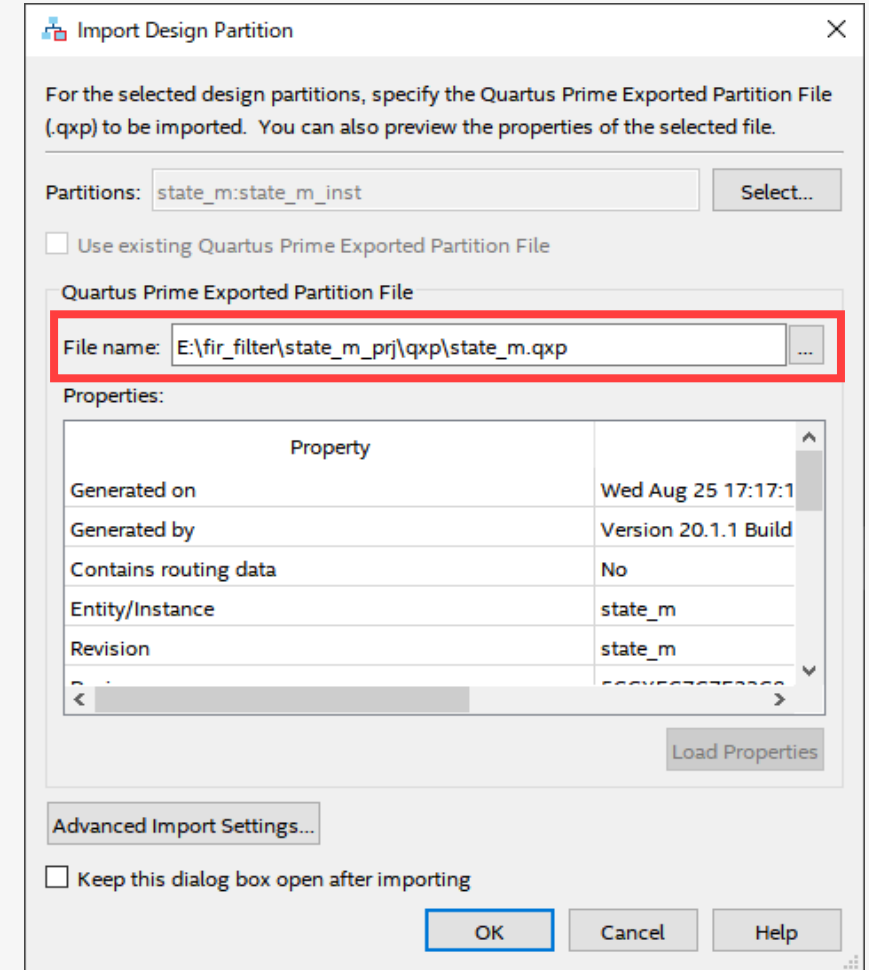


# Quartus Prime の操作 <インポート側 [2/2]>

## ④ Import Design Partition を選択



## ⑤ QXP ファイルを指定



### Notes:

- 対象エンティティ(.qxp)と周辺のロジック間は、パーティションをまたいだ最適化が実施されない。
  - リソースやタイミングの最適化の質が低下する可能性も

下記 Web コンテンツは関連した内容です。参考としてご覧ください。

配置制約した下位エンティティを別プロジェクトへ適用させる方法 (Standard Edition)



<p>FPGA 型番の 置き換え検討時に 便利な機能</p>	<p>自社デザインの ソースコードを 開示せずに納品先に 提供するワザ</p>	<p><b>ステートマシン回路 を作成時に 知っておきたい オプション</b></p>
<p>レジスタや 組み合わせノードを 保持する方法</p>	<p><b>Quartus Prime Tips</b> &lt;Standard Edition 編&gt;</p>	<p>QSF ファイルを カテゴリーで ソート表示させる方法</p>
<p>UI フォントサイズを 変更して 見やすくしよう</p>	<p>コンパイルレポートを HTML 出力して 見やすくしよう</p>	



# ステートマシンを作成

- 論理合成ツール(Quartus Prime を含む)は、速度と面積に応じてステートマシンのロジックを最適化する

## ◆ イリーガルステート

- 一般的な HDL 記述での対応
  - VHDL : others
  - Verilog HDL : default
- 論理合成ツールが最適化を実施
  - イリーガルステート用の回路は実装されない



もし、イリーガルステートに入ってしまうとリセットされるまで正常に機能しない可能性がある

## ◆ エンコード方式

- ステートレジスタ値の割り当て
- ユーザーが HDL でエンコード方法を記述しても、ツールが最適化
  - 主に One-Hot にエンコード
    - FPGA はレジスタが多い



One-Hot は“存在する”ステート数が多くなる

使わないステートが多いとイリーガルステートに入る可能性も出てくる



# ステートマシン回路を作成時に知っておきたいオプション

- そんなときは？

## Safe State Machine / State Machine Processing オプションを活用しよう！



### Safe State Machine

無効なステートから抜け出すためのロジックをインプリメントするオプション

On / Off

### State Machine Processing

ステートマシンをどのようなエンコード形式でコンパイルするかを指定するオプション

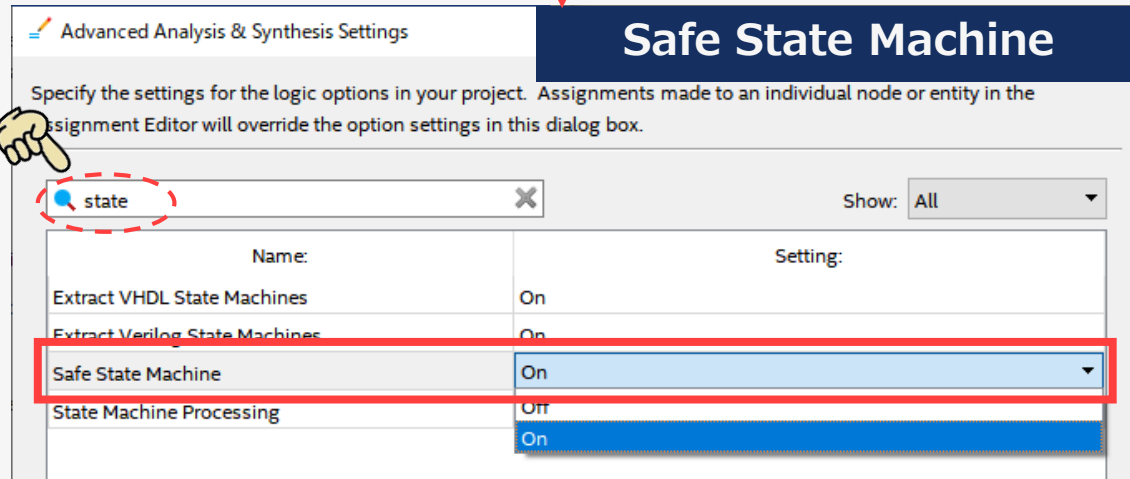
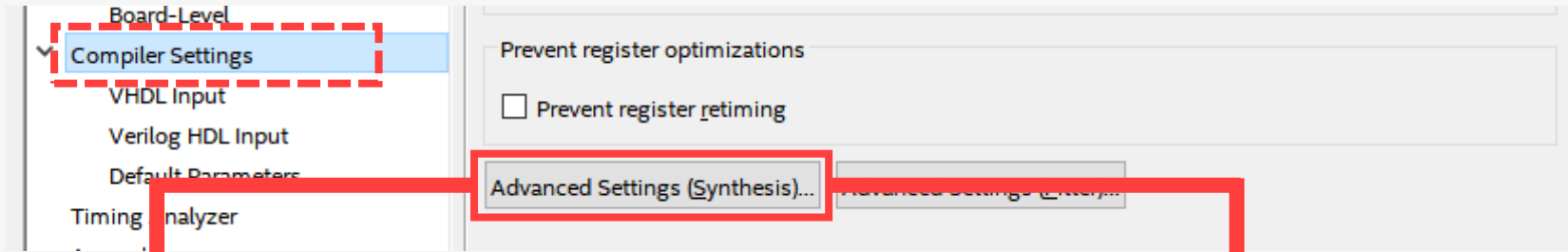
Auto / Gray / Johnson /  
**Minimal Bits** / One-Hot /  
**User-Encoded**

# Quartus Prime の操作

- Assignments メニュー ⇒ Settings  
⇒ Compiler Settings カテゴリー内 ⇒ Advanced Settings (Synthesis)

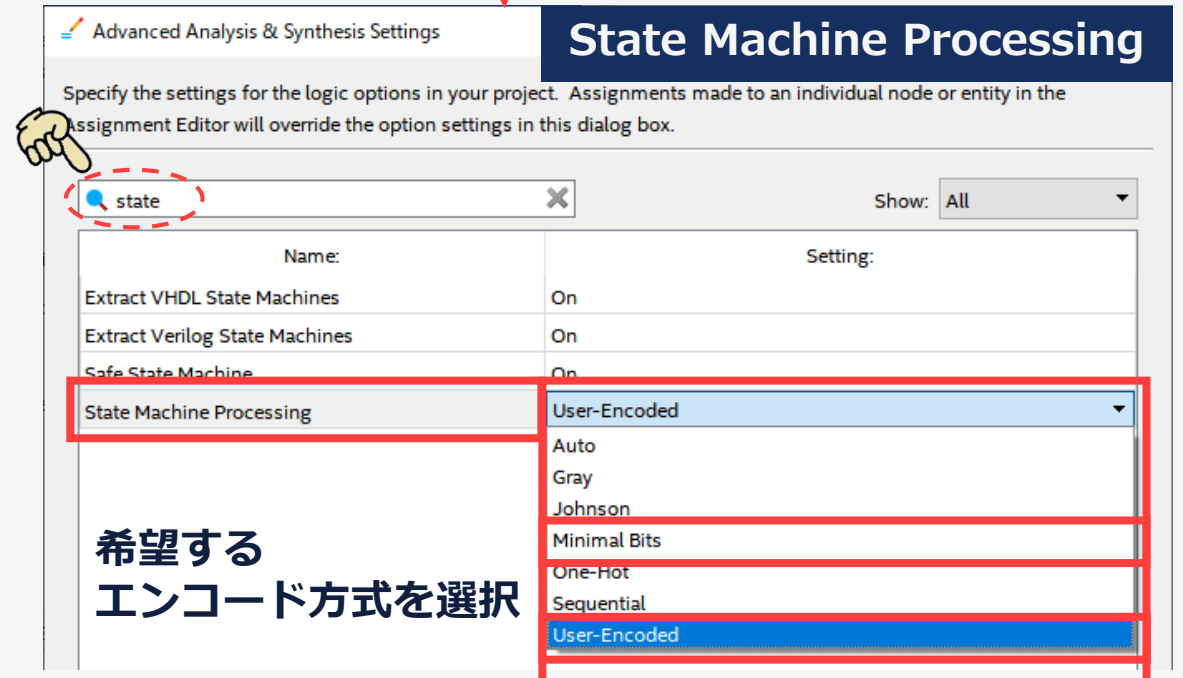
詳細は下記をご覧ください

「ステートマシン回路の設定方法」



イリーガルステートから脱出するためのロジックが追加

➔ ロジックの消費は増加



希望する  
エンコード方式を選択

<p>FPGA 型番の 置き換え検討時に 便利な機能</p>	<p>自社デザインの ソースコードを 開示せずに納品先に 提供するワザ</p>	<p>ステートマシン回路 を作成時に 知っておきたい オプション</p>
<p>レジスターや 組み合わせノードを 保持する方法</p>	<p><b>Quartus Prime Tips</b> &lt;Standard Edition 編&gt;</p>	<p>QSF ファイルを カテゴリーで ソート表示させる方法</p>
<p>UI フォントサイズを 変更して 見やすくしよう</p>	<p>コンパイルレポートを HTML 出力して 見やすくしよう</p>	

# レジスタや組み合わせがコンパイルでいなくなっただ！

- Quartus Prime のコンパイラがデザインを最適化

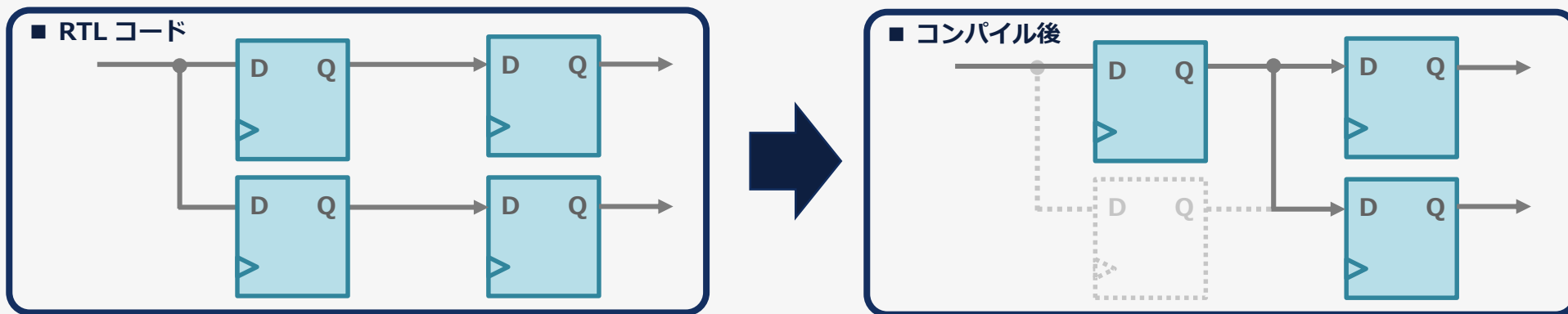
  - 無駄なリソースを削減 ⇒ 基本的には良いこと

- 設計者によっては “あえて” 構築したノード ⇒ 無くなると困る



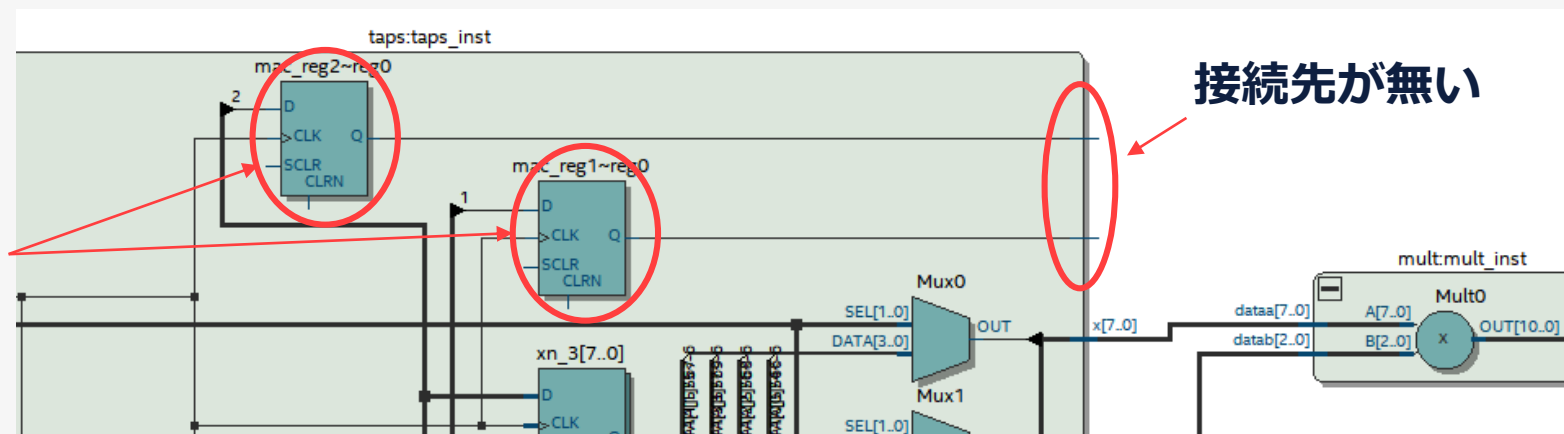
※ クロック信号の配線は省略。全て同じクロックドメイン。

例1)



例2)

論理圧縮される



# レジスターや組み合わせがコンパイルでいなくなった！

- そんなときは？

ロジック・オプションを活用しよう！



レジスターを保持する

Disable Register Merging

Preserve Registers

Preserve Fan-out Free Register Node

組み合わせノードを保持する

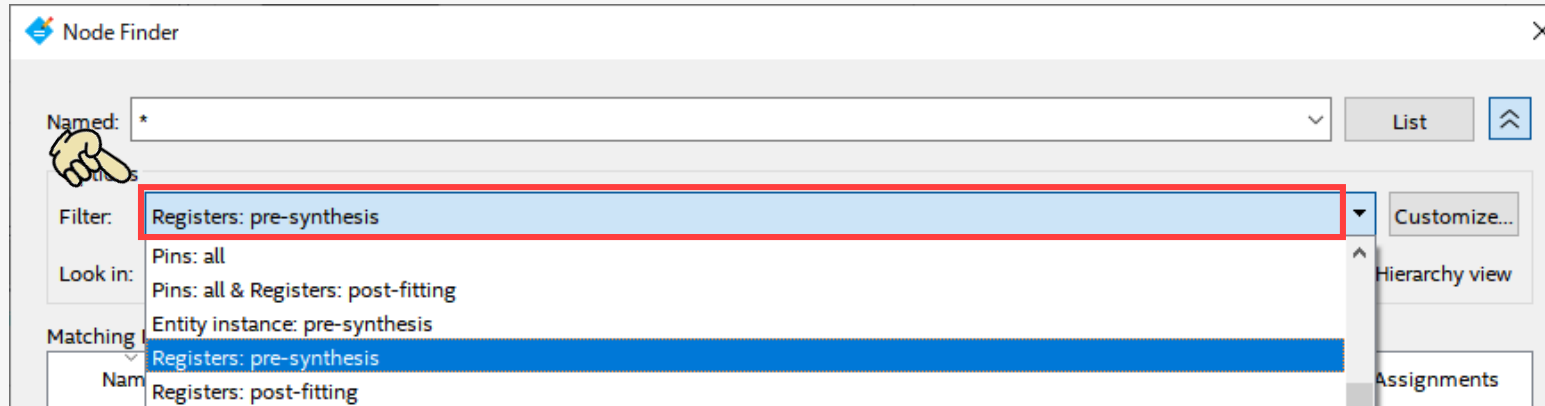
Implement as Output of Logic Cell

- 対象ノードに対して Assignment Editor で制約
  - ソースコードに直接 アトリビュート/プラグマ を記述しても OK！

# レジスターを保持する方法

## ● Assignments メニュー ⇒ Assignment Editor

1. Node Finder で保持したいレジスターを選択
  - Filter で “Registers: pre-synthesis” を使用して検出



2. 適切なオプションを選択し、Value = On にセット

Assignment Editor dialog box showing the 'Filter on node names' checkbox checked. The table below shows the assignments for the selected filter.

tat	From	To	Assignment Name	Value	Enabled
1	✓	R simple_c...cnica[1]	Preserve Registers	On	Yes
2	✓	R simple_c...cnica[0]	Preserve Fan-out Free Register Node	On	Yes
3	✓	R simple_c...t]cnt[2]	Disable Register Merging	On	Yes
4	✓	in RESET	I/O Standard	3.3-V LVTTTL	Yes

# レジスターを保持する 各ロジック・オプションの概要

## ● どう使い分けるか？



詳細は下記 Web コンテンツをご覧ください

「コンパイルで削除されたレジスタを保持する方法」

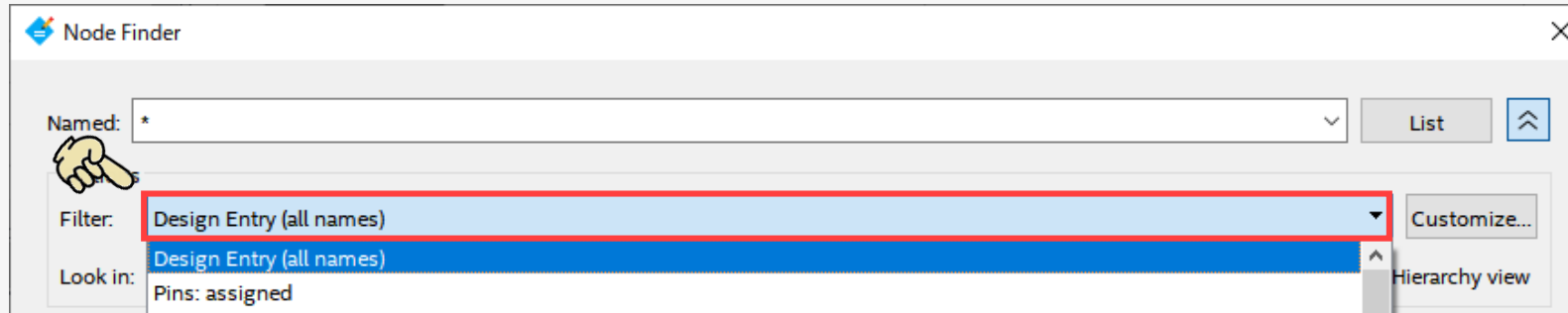
ロジック・オプション編	アトリビュート編



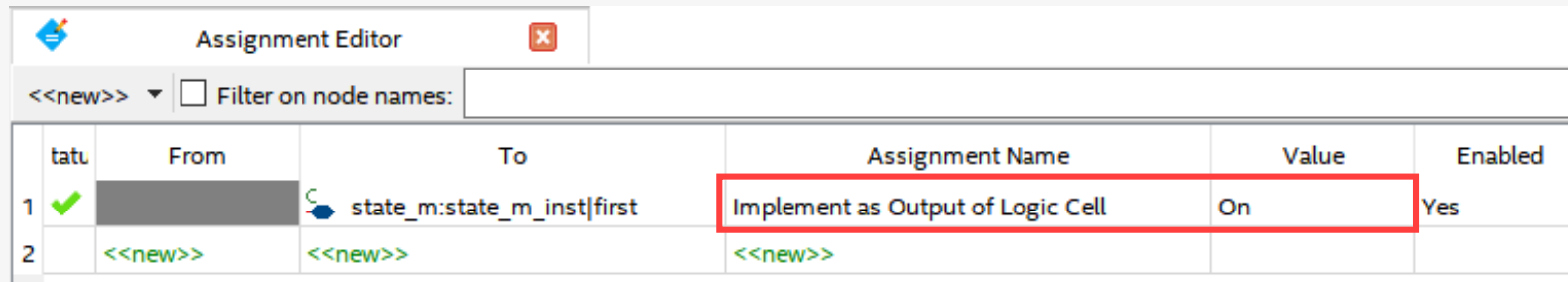
# 組み合わせノードを保持する方法 [1/2]

## ● Assignments メニュー ⇒ Assignment Editor

1. Node Finder で保持したい組み合わせノードを選択
  - Filter で “**Design Entry (all names)**” を使用して検出



2. Implement as Output of Logic Cell オプションを選択し、Value = On にセット
  - 必要に応じて Ignore LCELL Buffers = Off を併用



	tatu	From	To	Assignment Name	Value	Enabled
1	✓		state_m:state_m_inst first	Implement as Output of Logic Cell	On	Yes
2		<<new>>	<<new>>	<<new>>		

# 組み合わせノードを保持する方法 [2/2]

- Implement as Output of Logic Cell オプションが効かないときは？



## アトリビュート/プラグマをソースコードに挿入



### VHDL

```
signal my_wire: std_logic;  
attribute keep: boolean;  
attribute keep of my_wire: signal is true;
```

### Verilog HDL ※

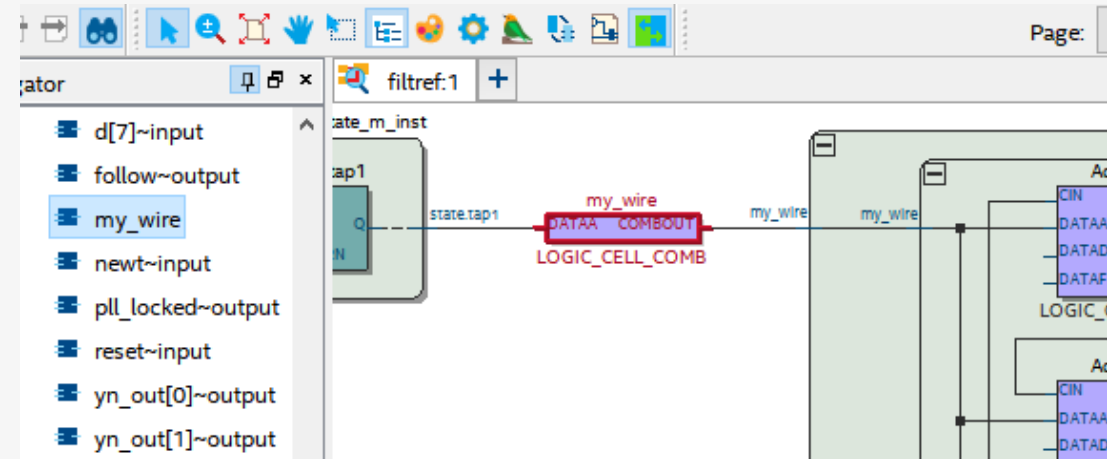
```
wire my_wire /* synthesis keep = 1 */
```

※ synthesis keep 後ろの “= 1” は省略することも可能

### Verilog-2001 ※

```
(* keep = 1 *) wire my_wire;
```

※ keep 後ろの “= 1” は省略することも可能



この内容は、Web コンテンツとして近日公開予定！

「コンパイルで削除された組み合わせ信号を保持する方法」

- ・ロジック・オプション編
- ・アトリビュート編

<p>FPGA 型番の 置き換え検討時に 便利な機能</p>	<p>自社デザインの ソースコードを 開示せずに納品先に 提供するワザ</p>	<p>ステートマシン回路 を作成時に 知っておきたい オプション</p>
<p>レジスタや 組み合わせノードを 保持する方法</p>	<p><b>Quartus Prime Tips</b> &lt;Standard Edition 編&gt;</p>	<p><b>QSF ファイルを カテゴリーで ソート表示させる方法</b></p>
<p>UI フォントサイズを 変更して 見やすくしよう</p>	<p>コンパイルレポートを HTML 出力して 見やすくしよう</p>	

# QSF ファイルをカテゴリーでソート表示させる方法

FAQ で公開中



## ● QSF ファイルとは？

- プロジェクトに設定された制約内容を記録したもの
  - Quartus Prime Settings File

## ● GUI でデザインに制約を設定する

- <プロジェクトのリビジョン名>.qsf に記録される
- 追加/変更された順でランダムに記録



QSF ファイルを  
直接参照または編集する  
際に見づらい



## ● そんなときは？

**Organize Quartus Prime Settings File**  
**を実行しよう！**



Project メニュー

⇒ Organize Quartus Prime Settings File

- 制約がカテゴリーごとにソートされる



```
40
41 # Project-wide Assignments
42 # =====
43 set_global_assignment -name SMART_RECOMPILE OFF
44 set_global_assignment -name LAST_QUARTUS_VERSION "20.1.1 SP
45 set_global_assignment -name PROJECT_OUTPUT_DIRECTORY output
46 set_global_assignment -name SDC_FILE filtref.sdc
   set_global_assignment -name                                /ans.vhd

57 # Pin & Location Assignments
58 # =====
59 set_location_assignment PIN_AA7 -to yvalid
60 set_location_assignment PIN_P8 -to follow
61 set_location_assignment PIN_AB7 -to yn_out[0]
   set_location_assignment PIN_A10 -to yn_out[1]

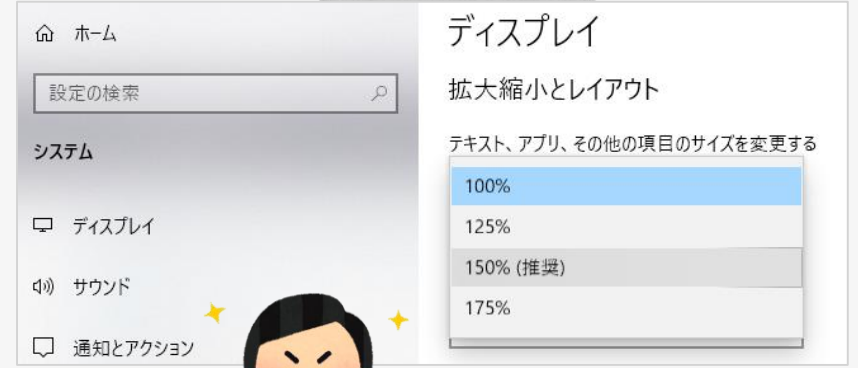
82 # Analysis & Synthesis Assignments
83 # =====
84
85 set_global_assignment -name FAMILY "Cyclone V"
86 set_global_assignment -name MUX_RESTRUCTURE ON
87 set_global_assignment -name OPTIMIZATION_TECHNIQUE AREA
88 set_global_assignment -name ASSIGNMENT_SET_OPT SYNTH_WYSIWYG_R
```

<p>FPGA 型番の 置き換え検討時に 便利な機能</p>	<p>自社デザインの ソースコードを 開示せずに納品先に 提供するワザ</p>	<p>ステートマシン回路 を作成時に 知っておきたい オプション</p>
<p>レジスタや 組み合わせノードを 保持する方法</p>	<p><b>Quartus Prime Tips</b> &lt;Standard Edition 編&gt;</p>	<p>QSF ファイルを カテゴリーで ソート表示させる方法</p>
<p><b>UI フォントサイズを 変更して 見やすくしよう</b></p>	<p>コンパイルレポートを HTML 出力して 見やすくしよう</p>	

# UI フォントサイズを変更して見やすくしよう



- 最近のノート PC のディスプレイやモニターは高解像度！
  - その反面 文字が小さくなって目が疲れる
- Windows 10 の場合、ディスプレイ設定で表示サイズを変更可能
  - すべてのアプリ表示も拡大される

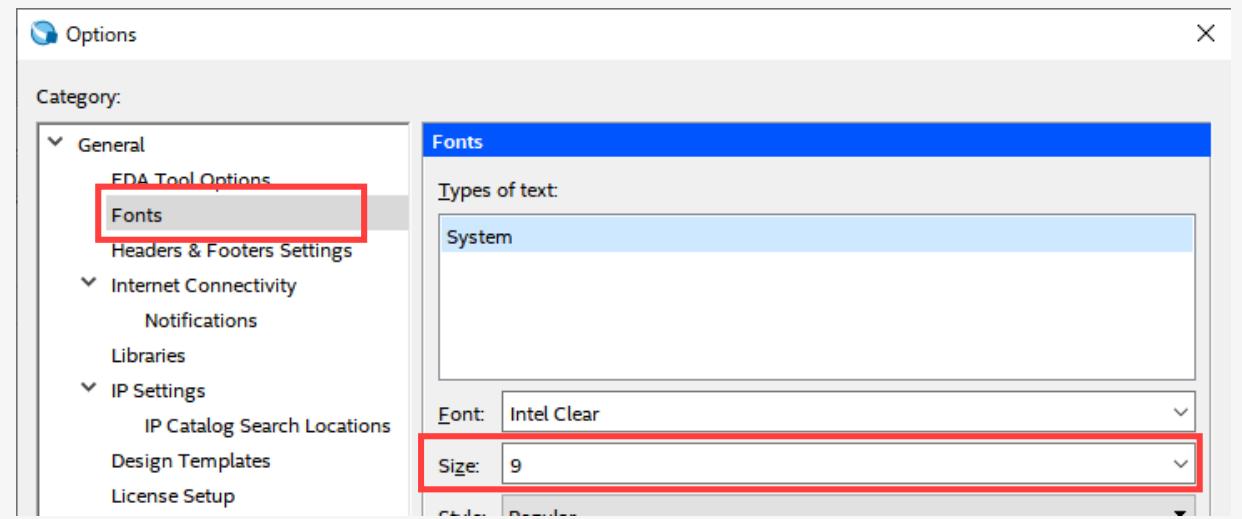


- そんなときは？

## Quartus Prime の UI フォントサイズを調整しよう！



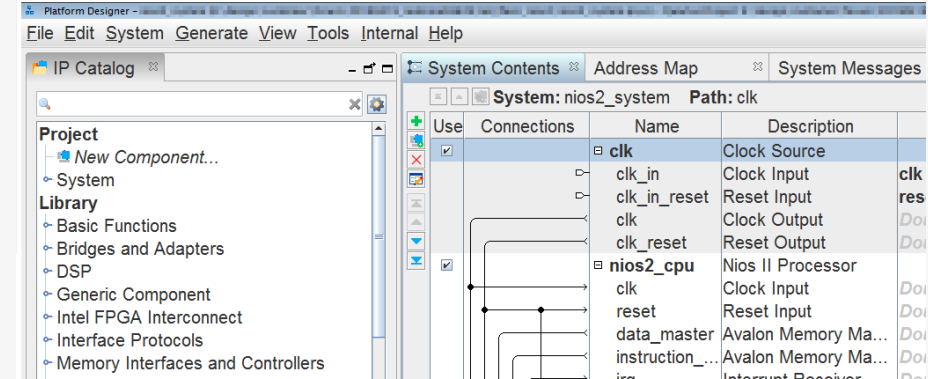
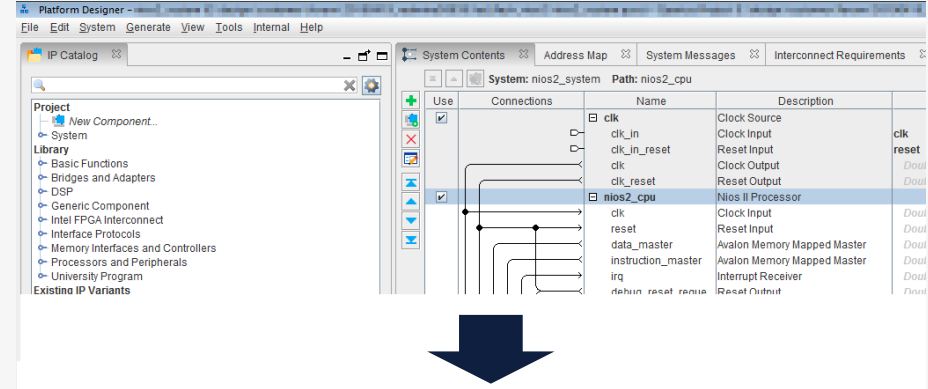
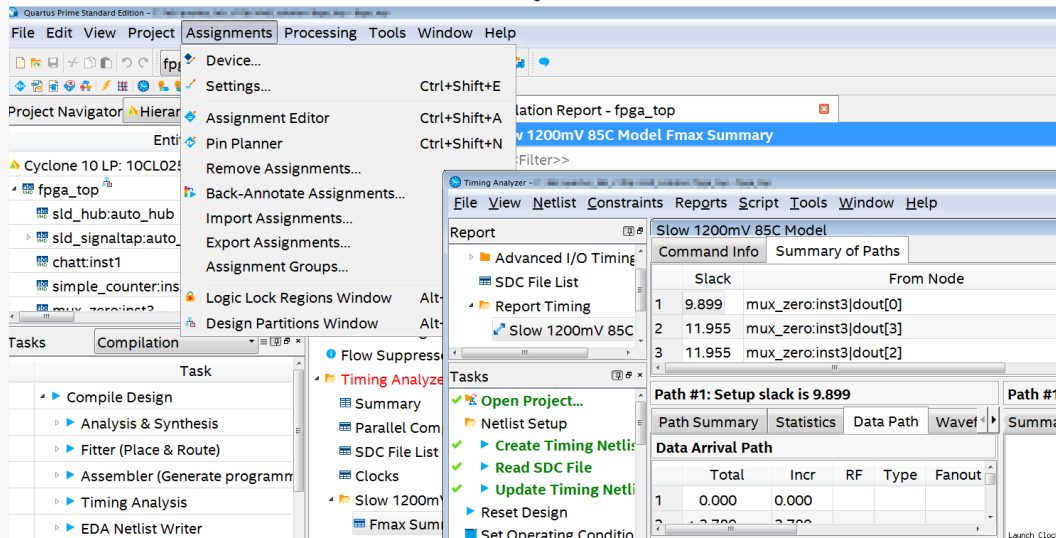
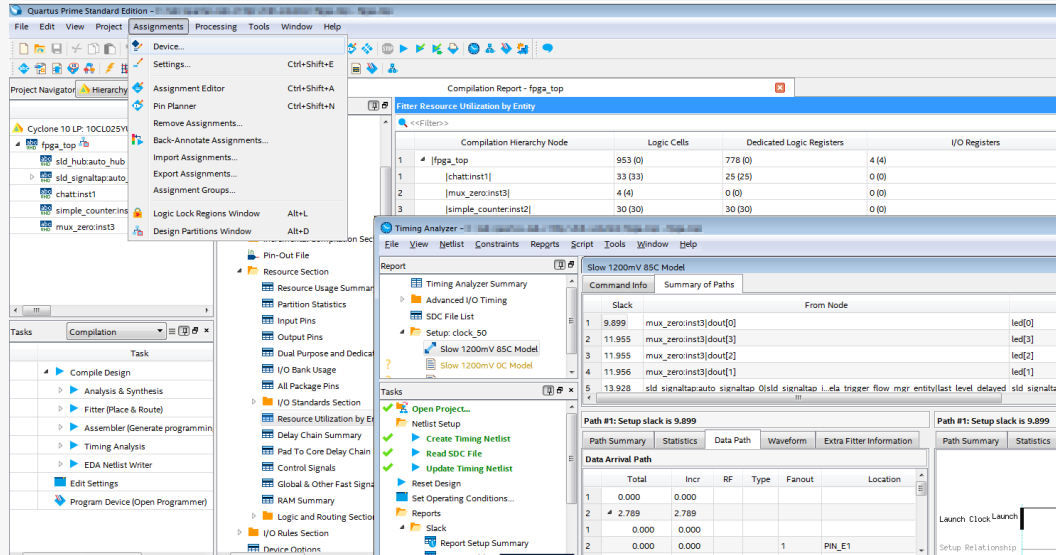
Tools メニュー ⇒ Options  
⇒ General カテゴリー内 Fonts  
⇒ System の Size で値を変更



# UI フォントサイズを変更してみました！

## Platform Designer の UI フォントサイズ

番外編



- OS の環境変数で設定
  - Windows
    - 環境変数名: QSYS\_FONTSIZE
    - 変数値: 希望のポイントサイズに設定
  - Linux
    - QSYS\_FONTSIZE=20



<p>FPGA 型番の 置き換え検討時に 便利な機能</p>	<p>自社デザインの ソースコードを 開示せずに納品先に 提供するワザ</p>	<p>ステートマシン回路 を作成時に 知っておきたい オプション</p>
<p>レジスタや 組み合わせノードを 保持する方法</p>	<p><b>Quartus Prime Tips</b> &lt;Standard Edition 編&gt;</p>	<p>QSF ファイルを カテゴリーで ソート表示させる方法</p>
<p>UI フォントサイズを 変更して 見やすくしよう</p>	<p><b>コンパイルレポートを HTML 出力して 見やすくしよう</b></p>	



# コンパイルレポートを設計グループで共有

- Quartus Prime で表示されるコンパイルレポート GUI は見やすい！  
でも、設計者以外の人とレポート情報を共有する場合は…



- 通常は

プロジェクトフォルダー内の \*.\*.rpt (テキストファイル) を共有



回路規模が大きいと、階層が深くノード名が長くなり 情報過多で見づらい

欲しい情報が見つけにくい! (こともある)



# Quartus Prime のコンパイルレポート

## ● Quartus Prime レポートファイル : 例) Resource Utilization by Entity

Table of Contents

- Pin-Out File
- Resource Section
  - Resource Usage Summary
  - Partition Statistics
  - Input Pins
  - Output Pins
  - Bidir Pins
  - I/O Bank Usage
  - All Package Pins
  - GXB Reports
  - Optimized GXB Elements
  - I/O Standards Section
  - PLL Usage Summary
  - Resource Utilization by Entity**
  - Delay Chain Summary
  - Pad To Core Delay Chain Fanout
  - Control Signals
  - Global & Other Fast Signals
  - Non-Global High Fan-Out Signals

Compilation Report - C5G\_HSMC\_XCVR\_LOOPBACK\_TEST

### Fitter Resource Utilization by Entity

Compilation Hierarchy Node	ALMs needed [=A+B+C]	[A] ALMs used in final placement	[B] Estimate of AL
1   C5G_HSMC_XCVR_LOOPBACK_TEST	5773.0 (0.3)	7206.5 (0.5)	1498.0 (0.2)
1     c5g_xcvr_qsys:u0	5704.7 (0.0)	7121.5 (0.0)	1481.3 (0.0)
1       alt_xcvr_reconfig:alt_xcvr_reconfig	1047.1 (44.9)	1179.9 (48.5)	134.1 (3.7)
1         alt_xcvr_arbiter:arbiter	3.1 (3.1)	3.1 (3.1)	0.0 (0.0)
2         alt_xcvr_reconfig_analog:analog_sc_analog	236.7 (0.0)	280.5 (0.0)	43.9 (0.0)
1           alt_xcvr_reconfig_analog_av:reconfig_analog_cv	236.7 (0.0)	280.5 (0.0)	43.9 (0.0)
1             alt_xreconf_analog_datactrl_av:inst_analog_datactrl	82.3 (39.6)	92.4 (43.3)	10.1 (3.7)
1               alt_xreconf_analog_ctrlsm:inst_analog_ctrlsm	9.4 (9.4)	11.2 (11.2)	1.7 (1.7)

GUI

3236 ; Fitter Resource Utilization by Entity

3237 -----

3238 ; Compilation Hierarchy Node ; ALMs needed [=A+B+C] ; [A] ALMs used in final placement ; [B] Estimate of ALMs recoverable by dense packing ; [C] Estimate of ALMs unavailable ; ALMs used for memory ; Combinational ALUTs ; Deduplication ;

3239 -----

Compilation Hierarchy Node	ALMs needed [=A+B+C]	[A] ALMs used in final placement	[B] Estimate of ALMs recoverable by dense packing	[C] Estimate of ALMs unavailable	ALMs used for memory	Combinational ALUTs	Deduplication	...				
3240   C5G_HSMC_XCVR_LOOPBACK_TEST	5773.0 (0.3)	7206.5 (0.5)	1498.0 (0.2)	64.5 (0.0)	0.0 (0.0)	9265 (1)	8622 (0)	0 (0)	1688960	213 ; 3	277 ; 0	...
3241     c5g_xcvr_qsys:u0	5704.7 (0.0)	7121.5 (0.0)	1481.3 (0.0)	64.5 (0.0)	0.0 (0.0)	9151 (0)	8534 (0)	0 (0)	1688960	213 ; 3	0 ; 0	...
3242       alt_xcvr_reconfig:alt_xcvr_reconfig	1047.1 (44.9)	1179.9 (48.5)	134.1 (3.7)	1.2 (0.2)	0.0 (0.0)	1804 (82)	1084 (5)	0 (0)	4096	2 ; 0	0 ; 0	...
3243         alt_xcvr_arbiter:arbiter	3.1 (3.1)	3.1 (3.1)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	6 (6)	7 (7)	0 (0)	0	0 ; 0	0 ; 0	...
3244           alt_xcvr_reconfig_analog:analog_sc_analog	236.7 (0.0)	280.5 (0.0)	43.9 (0.0)	0.1 (0.0)	0.0 (0.0)	359 (0)	299 (0)	0 (0)	0	0 ; 0	0 ; 0	...
3245             alt_xcvr_reconfig_analog_av:reconfig_analog_cv	236.7 (0.0)	280.5 (0.0)	43.9 (0.0)	0.1 (0.0)	0.0 (0.0)	359 (0)	299 (0)	0 (0)	0	0 ; 0	0 ; 0	...
3246               alt_xreconf_analog_datactrl_av:inst_analog_datactrl	82.3 (39.6)	92.4 (43.3)	10.1 (3.7)	0.0 (0.0)	0.0 (0.0)	145 (67)	55 (7)	0 (0)	0	0 ; 0	0 ; 0	...
3247                 alt_xreconf_analog_ctrlsm:inst_analog_ctrlsm	9.4 (9.4)	11.2 (11.2)	1.7 (1.7)	0.0 (0.0)	0.0 (0.0)	17 (17)	16 (16)	0 (0)	0	0 ; 0	0 ; 0	...
3248                   alt_xreconf_analog_rmw_av:inst_rmw_sm	33.3 (33.3)	37.9 (37.9)	4.6 (4.6)	0.0 (0.0)	0.0 (0.0)	61 (61)	32 (32)	0 (0)	0	0 ; 0	0 ; 0	...
3249                     alt_xreconf_cif:inst_xreconf_cif	81.5 (0.0)	96.1 (0.0)	14.6 (0.0)	0.0 (0.0)	0.0 (0.0)	95 (0)	149 (0)	0 (0)	0	0 ; 0	0 ; 0	...
3250                       alt_xreconf_basic_acq:inst_basic_acq	0.3 (0.3)	0.3 (0.3)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	1 (1)	0 (0)	0 (0)	0	0 ; 0	0 ; 0	...
3251                         alt_xreconf_basic_acq:inst_basic_acq	81.2 (81.2)	95.8 (95.8)	14.6 (14.6)	0.0 (0.0)	0.0 (0.0)	94 (94)	149 (149)	0 (0)	0	0 ; 0	0 ; 0	...
3252                           alt_xreconf_uif:inst_xreconf_uif	72.9 (72.4)	91.9 (90.7)	19.2 (18.4)	0.1 (0.1)	0.0 (0.0)	119 (117)	95 (91)	0 (0)	0	0 ; 0	0 ; 0	...
3253                             alt_xreconf_wait_generator:wait_gen	0.5 (0.5)	1.3 (0.5)	0.8 (0.0)	0.0 (0.0)	0.0 (0.0)	2 (1)	4 (2)	0 (0)	0	0 ; 0	0 ; 0	...
3254                               alt_xcvr_resync:inst_sync	0.0 (0.0)	0.8 (0.8)	0.8 (0.8)	0.0 (0.0)	0.0 (0.0)	1 (1)	2 (2)	0 (0)	0	0 ; 0	0 ; 0	...
3255                                 alt_xcvr_reconfig_basic:basic	357.2 (10.4)	376.7 (11.0)	20.5 (0.6)	1.0 (0.0)	0.0 (0.0)	633 (0)	192 (0)	0 (0)	4096	2 ; 0	0 ; 0	...
3256                                   alt_xcvr_reconfig_basic:a5	357.2 (10.4)	376.7 (11.0)	20.5 (0.6)	1.0 (0.0)	0.0 (0.0)	633 (22)	192 (15)	0 (0)	4096	2 ; 0	0 ; 0	...
3257                                     alt_xcvr_arbiter:pif[0]:pif_arb	0.5 (0.5)	0.7 (0.7)	0.2 (0.2)	0.0 (0.0)	0.0 (0.0)	1 (1)	1 (1)	0 (0)	0	0 ; 0	0 ; 0	...
3258                                       alt_xcvr_arbiter:pif[1]:pif_arb	0.7 (0.7)	0.7 (0.7)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	1 (1)	1 (1)	0 (0)	0	0 ; 0	0 ; 0	...
3259                                       alt_xcvr_arbiter:pif[2]:pif_arb	0.7 (0.7)	0.8 (0.8)	0.1 (0.1)	0.0 (0.0)	0.0 (0.0)	1 (1)	1 (1)	0 (0)	0	0 ; 0	0 ; 0	...
3260                                       alt_xcvr_arbiter:pif[3]:pif_arb	0.5 (0.5)	0.6 (0.6)	0.1 (0.1)	0.0 (0.0)	0.0 (0.0)	1 (1)	1 (1)	0 (0)	0	0 ; 0	0 ; 0	...
3261                                       alt_xcvr_arbiter:pif[4]:pif_arb	0.4 (0.4)	0.8 (0.8)	0.5 (0.5)	0.1 (0.1)	0.0 (0.0)	1 (1)	1 (1)	0 (0)	0	0 ; 0	0 ; 0	...
3262                                       alt_xcvr_arbiter:pif[5]:pif_arb	0.5 (0.5)	0.5 (0.5)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	1 (1)	1 (1)	0 (0)	0	0 ; 0	0 ; 0	...
3263                                       alt_xcvr_arbiter:pif[6]:pif_arb	0.5 (0.5)	0.5 (0.5)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	1 (1)	1 (1)	0 (0)	0	0 ; 0	0 ; 0	...
3264                                       alt_xcvr_arbiter:pif[7]:pif_arb	0.5 (0.5)	0.5 (0.5)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	1 (1)	1 (1)	0 (0)	0	0 ; 0	0 ; 0	...
3265                                       alt_xcvr_arbiter:pif[8]:pif_arb	0.7 (0.7)	0.7 (0.7)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	1 (1)	1 (1)	0 (0)	0	0 ; 0	0 ; 0	...
3266                                       alt_xcvr_arbiter:pif[9]:pif_arb	0.7 (0.7)	0.7 (0.7)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	1 (1)	1 (1)	0 (0)	0	0 ; 0	0 ; 0	...
3267   alt_xcvr_reconfig_bundle_to_basic:bundle	0.3 (0.3)	0.3 (0.3)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	1 (1)	0 (0)	0 (0)	0	0 ; 0	0 ; 0	...

テキスト

# コンパイルレポートを HTML 出力して見やすくしよう

- そんなときは？

**HTML で出力したレポートを共有しよう！**



Resource Utilization by Entity report for C5G_HSMC_XCVR_LOOPBACK_TEST													
Table of Contents	Fitter Resource Utilization by Entity												
<ul style="list-style-type: none"> <li>Legal Notice</li> <li>Fitter Resource Utilization by Entity</li> </ul>	Compilation Hierarchy Node	ALMs needed [=A- B+C]	[A] ALMs used in final placement	[B] Estimate of ALMs recoverable by dense packing	[C] Estimate of ALMs unavailable	ALMs used for memory	Combinational ALUTs	Dedicated Logic Registers	I/O Registers	Block Memory Bits	M10Ks	DSP Blocks	Pip
	C5G_HSMC_XCVR_LOOPBACK_TEST	5773.0 (0.3)	7206.5 (0.5)	1498.0 (0.2)	64.5 (0.0)	0.0 (0.0)	9265 (1)	8622 (0)	0 (0)	1688960	213	3	27
	c5g_xcvr_qsys:u0	5704.7 (0.0)	7121.5 (0.0)	1481.3 (0.0)	64.5 (0.0)	0.0 (0.0)	9151 (0)	8534 (0)	0 (0)	1688960	213	3	0
	alt_xcvr_reconfig:alt_xcvr_reconfig	1047.1 (44.9)	1179.9 (48.5)	134.1 (3.7)	1.2 (0.2)	0.0 (0.0)	1804 (82)	1084 (5)	0 (0)	4096	2	0	0
	alt_xcvr_arbiter:arbiter	3.1 (3.1)	3.1 (3.1)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	6 (6)	7 (7)	0 (0)	0	0	0	0
	alt_xcvr_reconfig_analog:analog_sc_analog	236.7 (0.0)	280.5 (0.0)	43.9 (0.0)	0.1 (0.0)	0.0 (0.0)	359 (0)	299 (0)	0 (0)	0	0	0	0
	alt_xcvr_reconfig_analog_av:reconfig_analog_cv	236.7 (0.0)	280.5 (0.0)	43.9 (0.0)	0.1 (0.0)	0.0 (0.0)	359 (0)	299 (0)	0 (0)	0	0	0	0
	alt_xreconf_analog_datactrl_av:inst_analog_datactrl	82.3 (39.6)	92.4 (43.3)	10.1 (3.7)	0.0 (0.0)	0.0 (0.0)	145 (67)	55 (7)	0 (0)	0	0	0	0
	alt_xreconf_analog_ctrlsm:inst_analog_ctrlsm	9.4 (9.4)	11.2 (11.2)	1.7 (1.7)	0.0 (0.0)	0.0 (0.0)	17 (17)	16 (16)	0 (0)	0	0	0	0
	alt_xreconf_analog_rmw_av:inst_rmw_sm	33.3 (33.3)	37.9 (37.9)	4.6 (4.6)	0.0 (0.0)	0.0 (0.0)	61 (61)	32 (32)	0 (0)	0	0	0	0
	alt_xreconf_cif:inst_xreconf_cif	81.5 (0.0)	96.1 (0.0)	14.6 (0.0)	0.0 (0.0)	0.0 (0.0)	95 (0)	149 (0)	0 (0)	0	0	0	0
	alt_arbiter_acq:mutex_inst	0.3 (0.3)	0.3 (0.3)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	1 (1)	0 (0)	0 (0)	0	0	0	0
	alt_xreconf_basic_acq:inst_basic_acq	81.2 (81.2)	95.8 (95.8)	14.6 (14.6)	0.0 (0.0)	0.0 (0.0)	94 (94)	149 (149)	0 (0)	0	0	0	0
	alt_xreconf_uif:inst_xreconf_uif	72.9 (72.4)	91.9 (90.7)	19.2 (18.4)	0.1 (0.1)	0.0 (0.0)	119 (117)	95 (91)	0 (0)	0	0	0	0
	altera_wait_generate:wait_gen	0.5 (0.5)	1.3 (0.5)	0.8 (0.0)	0.0 (0.0)	0.0 (0.0)	2 (1)	4 (2)	0 (0)	0	0	0	0

# レポートファイルを HTML ファイル出力方法

## ● Quartus Prime 上でコンパイルレポートを表示

- Processing メニュー ⇒ Compilation Report

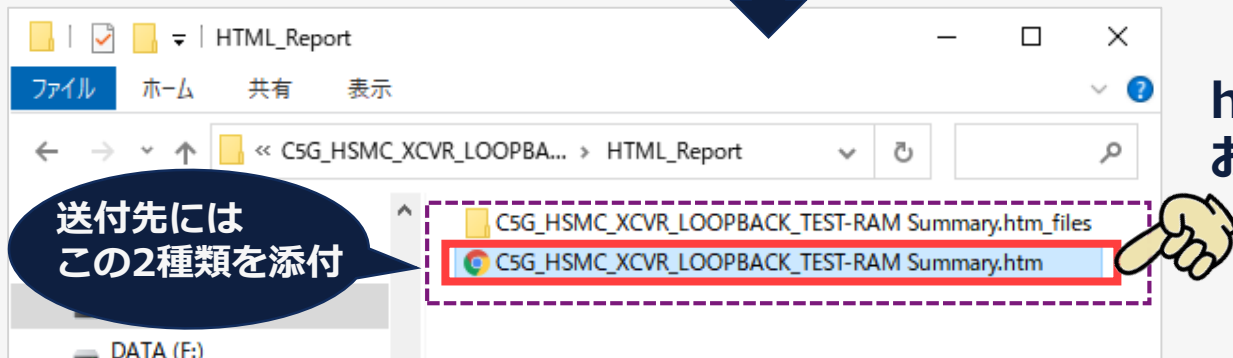
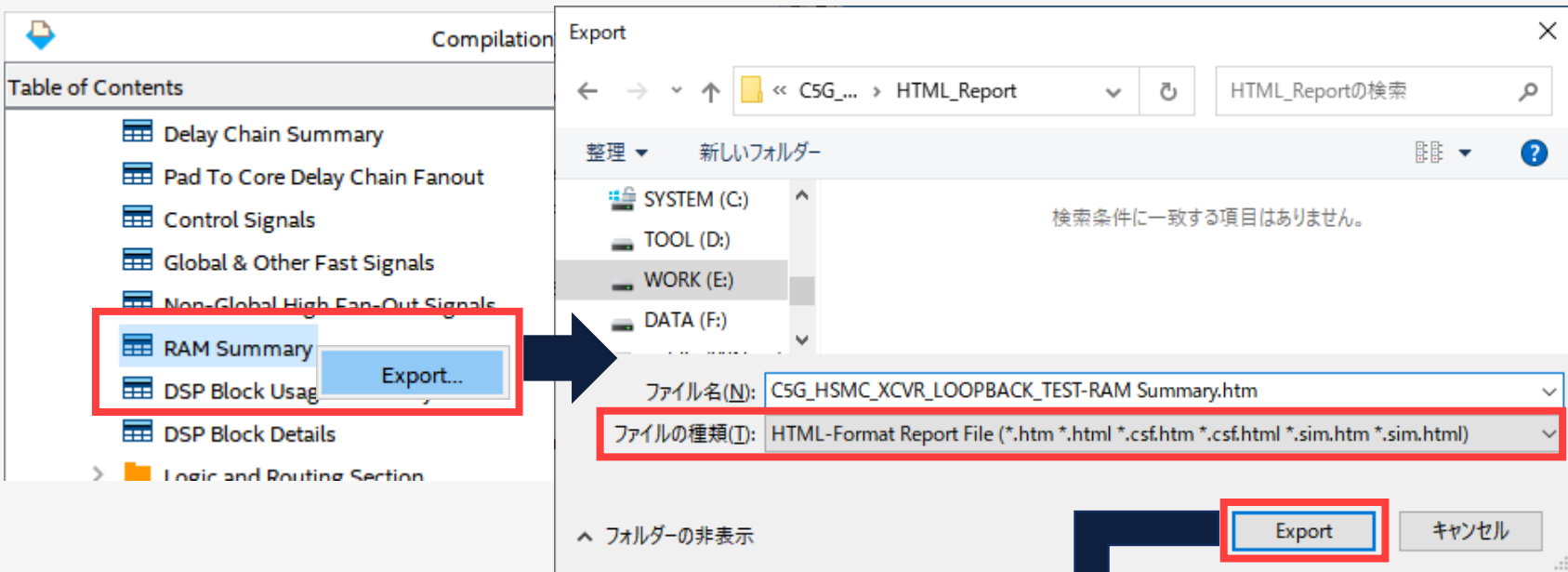
① 対象のレポート項目を右クリック ⇒ Export

② 生成先フォルダーを指定し、ファイルの種類に“HTML-Format Report File”を選択


③ Export ボタンをクリックして HTML ファイル生成

htm ファイルをダブルクリック！  
お好みの Web ブラウザーで閲覧

CSV ファイルで生成することも可能



# まとめ

<p>FPGA 型番の置き換え検討時に便利な機能</p> <p>デバイスマイグレーション機能</p>	<p>自社デザインのソースコードを開示せずに納品先に提供するワザ</p> <p>QXP ファイル生成</p>	<p>ステートマシン回路を作成時に知っておきたいオプション</p> <p>State Machine Processing Safe State Machine</p>
<p>レジスタや組み合わせノードを保持する方法</p> <p>ロジック・オプション アトリビュート/プラグマ</p>	<p><b>Quartus Prime Tips</b> &lt;Standard Edition 編&gt;</p>	<p>QSF ファイルをカテゴリーでソート表示させる方法</p> <p>Organize Quartus Prime Settings File</p>
<p>UI フォントサイズを変更して見やすくしよう</p> <p>Tools メニュー ⇒ Options ⇒ General ⇒ Fonts</p>	<p>コンパイルレポートをHTML 出力して見やすくしよう</p> <p>Export ⇒ HTML Format Report File</p>	<p> より便利に使って作業の効率をUP させましょう!</p> <p>intel.</p>

Co.Tomorrowing  
**MACNICA**

- ・本資料に記載されている会社名、商品またはサービス名等は各社の商標または登録商標です。なお、本資料中では、「™」、「®」は明記していません。
- ・本資料のすべての著作権は、第三者または株式会社マクニカに属しており、(著作権法で許諾される範囲を超えて) 無断で本資料の全部または一部を複製・転載等することを禁じます。
- ・本資料は作成日現在における情報を元に作成されておりますが、その正確性、完全性を保証するものではありません。