

# HDL Designer Series 簡易チュートリアル ~DesignChecker 編~

Ver.2016.2



2017年10月 Rev.1

**ELSENA,Inc.** 





# HDL Designer Series 簡易チュートリアル ~DesignChecker 編~

# <u>目次</u>

1. はじめに	3
2. HDL Designer Series 概要	3
3. 解析方法	4
4. DesignChecker の起動	5
4-1. HDL Desinger ウィンドウより DesignCheker を起動	5
5. 解析ルールの設定	6
5-1. Ruleset フォルダに新規フォルダを作成	6
5-2. Essential ルールをコピー	6
5-3. ベースルールから追加したいルールをコピー	7
5-4. ルールのパラメーターを編集	8
5-5. ポリシーの作成	9
5-6. ポリシーにルールセットをコピー	9
5-7. 解析に使用するポリシーを選択1	0
6. 解析の実行1	1
6. 解析の実行1 6-1. Design Manager ウィンドウを開く1	1 1
<ol> <li>6. 解析の実行</li></ol>	1 1 1
<ul> <li>6. 解析の実行</li></ul>	1 1 1 2
<ul> <li>6. 解析の実行</li></ul>	1 1 2 3
<ul> <li>6. 解析の実行</li></ul>	1 1 2 3 3
<ul> <li>6. 解析の実行</li></ul>	1 1 2 3 3 4
<ul> <li>6. 解析の実行</li></ul>	1 1 2 3 3 4 4
<ul> <li>6. 解析の実行</li></ul>	1 1 2 3 4 4 5
<ul> <li>6. 解析の実行</li></ul>	1 1 2 3 4 4 5 8
<ul> <li>6. 解析の実行</li></ul>	1 1 2 3 4 4 5 8 9
<ul> <li>6. 解析の実行</li></ul>	1 1 2 3 4 4 5 8 9 9

## 1. <u>はじめに</u>

DesignChecker は、HDL Designer Series<sup>™</sup> に同梱されているリント・チェック機能です。DesignChecker ではパラメ ーター設定が可能な豊富なベースルールを持っており、それらを組み合わせた標準ルールセットが用意されてい ますので、はじめてリント・チェックを行う方でも簡単に基本的なチェックを実行することができます。また、ベース ルールの組み合わせを変更したり、パラメーターを変更したりすることによってカスタムルールを作成することも可 能です。この資料では、DesignChecker の基本的な使用方法を紹介します。

■ 対応バージョン

HDL Designer Series ver.2016.2

#### 2. HDL Designer Series 概要

HDL Designer Series<sup>™</sup> (HDS)は、HDL ベースの設計・管理を総合的に行うことができる開発環境です。

デザイン・エントリーのための各種グラフィカル・エディタを備えており、作成したデザインから HDL を生成する ことができます。テキスト・エディタも搭載されていますので、HDL 設計にも使用することが可能です。

また、HDL からブロック図やフローチャート等のグラフィカル・ビューを作成することができるため、デザインの動作を理解するのに役立ちます。その他、リント・チェック機能やソースファイルのバージョン管理機能等も備えています。

HDL Designer Series には2種類の製品があり、使用できる機能が異なります。

- HDL Designer : すべての機能を使用可能
- HDS Author : デザイン編集機能に特化

3	機能	HDL Designer	HDL Author	備考
	VHDL	0	0	VHDL 87, VHDL 93, VHDL 20202, VHDL 2008
言語	Verilog	0	0	Verilog 95, Verilog 2001, Verilog 2005
	SystemVerilog	0	0	SystemVerilog 2005 (*グラフィックは非対応。 テキストのみ)
	Windows	0	0	Windows <sup>®</sup> 7 (64bit), Windows 8.1 (64bit) (*32bit アプリで動
OS	Windows	0	0	Author         価券           〇         VHDL 87, VHDL 93, VHDL 20202, VHDL 2008           〇         Verilog 95, Verilog 2001, Verilog 2005           〇         SystemVerilog 2005 (*グラフィックは非対応。テキストのみ)           〇         SystemVerilog 2005 (*グラフィックは非対応。テキストのみ)           〇         Windows® 7 (64bit), Windows 8.1 (64bit) (*32bit アプリで動作)           〇         RHEL6 (64bit), RHEL 7 (64bit) (*32bit アプリとして動作)           〇         RTL コードをブロック回やフローチャート等)           HDL ファイルをグラフィカレ マケローチャート、ステートマシン等           のグラフィック に変換(編集は不可)           L THML ファイルで保存           リントチェッカー機能           〇         Subversion®, IBM Rational ClearCase®, CVS, RCS に対応           〇         FPGA ベンダ・ツールとの連携
	Linux	0	0	RHEL6 (64bit), RHEL 7 (64bit) (*32bit アプリとして動作)
デザインルエンル	11	0	0	テキスト・エディタ、各種グラフィカル・エディタ(ブロック図、
7942-121	·)—	0	0	ステートマシン、フローチャート等)
		0		HDL ファイルをグラフィカル・エディタで編集可能なファイル
		0		に変換
┏ҵ コードの司は	目化	0	0	RTL コードをブロック図やフローチャート、ステートマシン等
		U	U	のグラフィックに変換(編集は不可)
๛ –ะก		0		RTL コードを ブロック図等のグラフィックに変換(編集不可)
		U		し THML ファイルで保存
デザイン検証		0		リントチェッカー機能
バージョン管理		0	0	Subversion <sup>®</sup> 、IBM Rational ClearCase <sup>®</sup> 、CVS、RCS に対応
FPGA ベンダフロ	]—	0	0	FPGA ベンダ・ツールとの連携



# 3. <u>解析方法</u>

DesignChecker では下記の手順にて解析を実行します。



次章より基本的な手順をご紹介しますので、手順に沿って DesignChecker を操作し解析を実施してみましょう。

## 4. DesignChecker の起動

DesignChecker にてリント・チェックを行うには、まずチェック基準となるルールを設定する必要があります。 ルール設定を行うために、DesignChecker の設定ウィンドウを起動します。

※ プロジェクトの設定やデザインファイルのインポート方法については下記の資料をご参照ください。

「HDL Designer Series<sup>™</sup> - 簡易チュートリアル ~グラフィカルビュー変換編~」

http://www.altima.jp/members/p1-literature/1-software/2-mentor/078 hds tutorial visual.cfm

#### 4-1. HDL Desinger ウィンドウより DesignCheker を起動

Design Manager ウィンドウより「Tools メニュー」>「Manage Policies/Ruleset」を選択します。

Design Manag	jer - Project my_pro	oject					
<u>File E</u> dit <u>V</u> iew	v HDL Tasks <u>T</u> o	ools Options <u>W</u> indow <u>H</u> elp					
] 🏠 🔸 🚔   🎒	👌 + 👗 🗈 I	Find			- 28 - 🔎 🛛 🗖	🔊 🐼 - 🎹 😭	í - 🕅 - 😵
Main	Design Explore	Advanced Find		Dont Tou	ch Hidden ]		
	Design Unit	Viewpoint Manager		ds	Language	Time Stamp	Name RULET
New / Add		Component Browser DesignPad	F3	VHDL	VHDL VHDL	Tue Jun 10 2014 13:2 cor Tue Jun 10 2014 13:4 cou Tue Jun 10 2014 14:0 dec Tue Jun 10 2014 14:0 dec	control counter
		Manage Policies/Rulese			VHDL 2008		decoder filter
_ <b>_  _  _  _  _  _  _  _</b>		Registers			VHDL	Tue Jan 06 2015 21:2	inc
		Systemverliog Assistant	•		VHDL VHDL	Thu Jan 15 2015 17:5 Fri Jan 09 2015 11:13:	selector top
1		Mark Ton-Level Design Units					

別ウィンドウで DesignChecker が起動します。

🐼 DesignChecker (	(HDL Designer)		
File Run Edit	View Setup Exclusions Results Tools Options Help		
1 🖀 • 🖬 👿 •	∦ ๒ ६  ‡ ≠ ६  А 2  � ♣ &  № № 月 В		
Setup	Folders	Content of Policies	<u> </u>
1 Mary Dalian	Policies	Name ∠	
Rew Folicy	主 🕨 🛱 demo	🛱 demo	
😫 New RuleSet	My_Altera_Policy	My Altera Policy	
	My_Checklist_Policy	My Checklist Policy	
📮 Expand All	H My_DU-254_Policy	B My D0-254 Policy	
📥 Collapse All	The second secon	My Essentials Policy	
	My_Safety-Critical_Policy	B My BMM Policy	
Set As Default	+ 😰 My_Xilinx_Policy	Mu SafetuCritical Policu	
	E Verification_0VM_Policy	Mu Xiliov Policu	
	Verification_UVM_Policy	(aritication O)(M Policy	
	RuleSets	Verification LD/M Policy	
		se vehication_ovim_Policy	
	- A Essentials_demo	Parametere	
	Code Reuse	Parameter	Value
	Coding Practices		rate
	Downstream Checks		
	t + - ▲ 🖾 Essentials_original_do_not_modify		
	Verification_SV		
	▲ Verification_UVM		
	主 🛆 🧱 Xilinx		
	🕢 🔂 Base Rules		
<b>5</b> 1 1			
E xclusions			
Cheeked Files /Desig	Clocks & Besets		
Pula Dataila			
Viewpoints	Setup	1	
Ready			
( cody			



### 5. <u>解析ルールの設定</u>

DesignChecker にはデフォルトで下記のルールセットが設定されています。このルールセットを使用して解析を 行うことが可能です。ここでは、デフォルトのルールセット(Essentials)にルールを追加して使用する方法を例として ご紹介します。

DesignChecker 上のルールの構成については Appendix 9-1.DesignChecker ルールの構成をご参照ください。

- 5-1. Ruleset フォルダに新規フォルダを作成
  - ① 右側の Folders 欄より RuleSets フォルダを「右クリック」>「New Ruleset」を選択します。



- ② RuleSets フォルダに新しいフォルダが作成されますので、任意のフォルダ名をつけます。ここでは "test" と いう名前を付けています。(フォルダ名には日本語、スペースを含むファイル名は使用しないでください)
- ③ 新規フォルダを生成した際にフォルダアイコン上に表示される「!」マークは変更内容がまだ保存されて いないことを示しています。「Ctrl+S」もしくは「File メニュー」> 「Save All」より設定を保存します。
- 5-2. Essential ルールをコピー
  - 4-1 で作成したフォルダに既存の Essential ルールを下記の手順でコピーします。
  - ※ Essential ルールはその他の既存のルールセットと異なり、編集可能となっています。デフォルトのルール設定を 変更せずに残しておきたい場合は、コピーして使用することをお勧めします。
  - ① RuleSets フォルダ内の Essential フォルダを「右クリック」>「Copy」を選択します。
  - ② その後、5-1 で作成した test フォルダで「右クリック」>「Paste」を選択します。



test フォルダに Essential ルールがコピーされました。「!」マークがフォルダにつきますので「Ctrl+S」で保存します。



5-3. ベースルールから追加したいルールをコピー

ユーザが作成したルールセットでは、追加したいルールを Base Rules フォルダよりルールセットに追加して使用することができます。ここでは、例として "Port Mapping" というルール 5-1 で作成した test ルールセットフォ ルダにコピーしてみましょう。

- ① Base Rules フォルダ より Instances フォルダを選択します。
- ② 右上の Content of Base Rule Category 欄に Instances フォルダ内のベースルール一覧が表示されます。
- ③ "Port Mapping"ルールを「右クリック」>「Copy」を選択します。





④ RuleSets > test フォルダを「右クリック」>「Paste」を選択します。

右上の Content of RuseSet 欄に "Port Mapping" ルールが追加されます。

A		a particular second second	19 944	1964	
Folders			•	Content of RuleSet : test	
🕂 🕀 🔁 My_RMM	_Policy		*	Name 🛆	
🕂 🕂 🙀 My_Safet	y-Critical_Policy			🛈 Essentials	
🕂 🕀 😥 My_Xilinx	_Policy		ſ	🕲 Port Mapping	
🕂 🕂 🔛 🔁 Verificatio	n_OVM_Policy		Ļ		
🕂 🐨 🚼 Verificatio	n_UVM_Policy	4	٨		
🔽 🛄 RuleSets					
🕂 🛆 📴 Altera					
🕂 🛆 🙀 Checklist					
🕂 🛆 📴 DO-254					
🕂 🛆 😰 Essentials	\$				
🕂 🛆 😰 Essentials	s_demo				
🕂 🛆 🕄 Essentials	s_original_do_not_mo	odify			
			F		
T T T	Expand All				
+-∆🙀 BM	Collapse All			Parameters	
+ ∆ 🔙 Saf				Parameter	Value
∆ 🔄 Ver	New RuleSet				
∆ 🔙 Ver	Cut	Ctrl+V			
	Cut	Cul+x			
	Сору	Ctrl+C			
Base Rue	Paste	Ctrl+V			
		W5			

#### 5-4. ルールのパラメーターを編集

ルールセットに追加したベースルールのパラメーターを編集することが可能です。

 RuleSets の test フォルダをハイライトして、Content of RuleSet 欄に test フォルダを表示させ、Port Mapping ルールを選択します。

右下の Parameter of Configured Rule 欄に、Port Mapping ルールのパラメーターが表示されます。

- ② パラメーターを編集する場合には、各パラメーターの Value 欄をクリックし編集します。
- ③ ルールの編集が完了しましたら、「Ctrl+S」で編集した内容を保存します。

Parameter	Value
ରହି Name	Port Mapping
🐼 Severity	Error
🗞 Score	5
🐼 Weight	2
🐼 Language	VHDL Any, Verilog Any
🐼 Hint	Port Maps should use named association with explicit widths for vectors. Avoid use of
🐼 Short Description	Checks for a consistent port mapping style.
🐼 Keywords	port, mapping, named, positional, style, logical, expressions, width, vector, non-explicit
🐼 Mapping Style	Positional
🐼 Threshold	Named Resident
생 Disallow	

※ 各ルールのパラメーターには、すべてのルールに共通の項目と、ルール毎の項目があります。



※ ルールのパラメーターの詳細を確認するには、ルールを「右クリック」>「Base Rule Details」を選択すると、ルールの内容やパラメーターの説明が記載された HTML 形式のページが表示されます。こちらには、説明とともにエラーになる記述サンプル等も合わせて記載されています。

			Port Mapping x		
			← → C ① file:///C:/MentorGraphics/HDS_2016.2/docs/htmldocs/baserules/PortMapping.htm	🔤 🕁 🚺 😣	1
			🔢 アプリ G google 📒 mac 📙 IntelFPGA 📒 Mentor 📒 Sutdy 📒 Sleep ★ Bookmarks 🗅 新しいタブ 🏟 設定 🚾 COSMO 🔤 Box		
			Instance Base Rules Port Mapping TOC < RRG	V NEXT >	h.
Contract of	J.		Port Mapping		
Lontent of	HuleSet : test		Cherks for a consistent Port Mannion style		
Name			Description:		
			This rule checks the style of port mapping associations (named or positional/ordered). It also checks for the use of expressions and non-explicit widths in port map	is, and checks for	
🗓 Port M			signals connected to more than one input of the same instance.		
	pen				
	Cut	Chrly V	Parameters:		
	Cut	Cul+X	See also Parameters Common to all Base Rules		
	Сору	Ctrl+C	Mapping Style		
	Paste	Ctrl+V	Specify the required port map association style – Named or Positional/Ordered.		
			Values: Named, Positional		
	Delete	Del	Default: Named Threshold		
			Specify the minimum number of port associations that trigger the check. This is used to control the mapping style check only, and has no effect on the disallowed	parameter.	
	Find In Here		The mapping style check is not done if the number of port associations is below the threshold.		
	Find Where Used		Values: <user entered=""></user>		
			• Default: 5		
Paramete	Chow PacoPulo		Unsarrow Specify which constructs are not allowed within Port Mans		
	Base Rule Details		Values: Intercontected ports and signals with different names     Logical Expressions		÷

5-5. ポリシーの作成

チェックを実行する単位は、ルールセットではなく、ポリシー単位で行います。作成した test ルールセットを使用 するために、新規でポリシーを作成します。

- ① Folders 欄の Policies フォルダを「右クリック」>「New Policy」を選択します。
- ② 新規フォルダが作成されるので、任意の名前を付けます。ここでは "test\_Policy" という名前にしています。



5-6. ポリシーにルールセットをコピー

作成したポリシーフォルダに使用するルールセットをコピーします。

- ① Rulesets の test フォルダを Policies の test\_Policy フォルダにコピー & ペースト、もしくはドラッグ & ドロップでコピーすることができます。
- ② コピーしたら「Ctrl+S」で保存します。



#### 5-7. 解析に使用するポリシーを選択

Policies フォルダには複数のポリシーを保存しておくことができます。これらの中でどのポリシーを解析に使用するのかを設定しておく必要があります。

① 使用するポリシー(今回は test\_Policy)をハイライトして「右クリック」>「Set As Default」を選択します。

![](_page_9_Picture_6.jpeg)

Test\_Policy が太字になります。これで解析ルールの設定が完了しました。

# 6. <u>解析の実行</u>

- 5. 章で設定したルールを使用して解析を実行します。
- 6-1. Design Manager ウィンドウを開く

Design Manager のウィンドウに戻ります。

Design Manag	ger - Project hds_seminar_demo				
<u>File Edit Viev</u>	v HDL Tasks <u>T</u> ools <u>O</u> ptions	Window Help			
1 🐴 🗸 🚔 🖽	🔕 •   X 🖣 🛍 🗙 🕼 🛛 🏯	/ 🖬 🖽 🗛 🖪		- <b>ħ ₽ @</b>	
<b>₩</b> ₩ • <b>Ш</b>	■ • ◎ • M • ፵ • <b>24</b> • %	· •			
Main	Design Explorer [Using viewpoin	t : Default Viewpoint ]		? 4 ×	Tasks 🤶 🛃
	Design Unit	Туре	Design Hierarchy	Design Unit	My lasks
HDL	- Uart_sample	Madula	□ Part to	uart_tb	M V DesignLhecker
New / Add	+ clock divider	Module		tester	DesignUnecker Flow
	+ control operation	Module	F 🛱 U 1	uart top	Desister Assister
	+ cpu_interface	Module			Tregister Assistant
	🛨 🔚 serial_interface	Module			T as ModelSim Compile
Check	+ status_registers	Module			ModelSim Simulate
	+ tester	Module			RuestaSim Compile
		Module			QuestaSim Simulate
	+ smit rev control	Component			+ Precision Synthesis
Simulate					+ LeonardoSpectrum
					Altera MegaWizard
	< III	4			Altera SOPC Builder
	Files	Tune			FPGA Technology Setup
Synthesize	+ W DesignChecker	1300			FPGA Library Compile
	Visualiz				🛨 🦻 🕏 🛛 Quartus II Synthesis
	HTML				🛛 🚱 Quartus Place and Route
	🖃 🔂 Visualization	E			Quartus Programmer
Visualize	🖃 🛄 uart_sample				C/C++ Wrapper Generator
	+ 🛃 address_decode	Module			M ▼ ModelSim Flow
	主 📷 clock_divider	Module			UuestaSim Flow
	+ 🔂 control_operation	Module			teonardoSpectrum Flow
Registers	🛨 🔂 cpu_interface	Module			+ recision Synthesis Flow
	+ serial_interface	Module			
<b>•</b>	+ status_registers	Module			
Explore	+ koster	Module +			
Tasks	· · · · · · · · · · · · · · · · · · ·	•	<	+	
Viewpoints	Project hds_seminar_demo_lib ua	art_sample			Lasks Templates

#### 6-2. 解析の実行

- ① Design Unit 欄から解析を実行するモジュールを選択してハイライトします。
- メニューバーにある「Design Checker」のプルダウンを開きます。どの範囲で解析を行うかを選択して実行します。

V	Single :	選択したモジュールのみ解析
1	Through Block :	選択したブロック以下すべてのモジュールを解析
₩	Through Component :	選択したコンポーネント以下のモジュールを解析
<b>\</b>	Through Design :	Root 以下すべてのモジュールを解析*

(\*) あらかじめデザインルートを設定しておく必要があります。
 Design Unit 欄よりモジュールを選択して右クリックから設定できます。

6-3. 解析結果のレポート表示

解析が完了すると下記のように DesignChecker の Result 画面が表示されます。

📝 DesignChecke	r (HDL Designer)			
File Run Edit	View Setup Exclusions Results Tools Options Help			
🏠 • 🖬 👹	• ] ※ 凾 @   章 幸  色 ] ぬ ② ] 🧇 🍰 🍰 🖺 🖳 🖽			
Setup	Results (Using viewpoint: Severity & Ruleset)		G Summary	4 ×
Exclusions	Message	∠ Line Number Filename	🖃 Settings	
Results			Policy:	test_Policy
茸 Expand All	🗉 🎱 Warning - 5 items, 30 violations. (28 primary, 2 associated)		Library:	uart_sample
÷ C-II AII	O Error - 3 items, 28 violations. (22 primary, 6 associated)		Primary:	uart_top
Collapse All			Secondary:	uart_top
Dpen Source			Master Clocks:	clk
Jourse			Master Resets:	rst
Open HDL			Depth:	ThroughDesignRoot
😫 Show Rule			Check Level:	Design Unit Level
📐 Disable Rule			Exclusions	
🛌 Disable			Number of exclusions in the se	ittings:
RuleSet			Policy Disabled Rules	0
Show			Code/Rule Exclusions	0
Violations			Black Boxed Files	0
			Don't Touch Files	0
			Exclusion Pragmas	0
			Pragma Code Excluded	0
			Missing Masters	0
			Unbound Instances	U
			⊞ Design Quality: 166/207 (80%)	
			Rules: (Using policy test_Policy	)
			🛨 Design Units:	
Checked Files/De				
Rule Details	•		×	۱. ۲
Viewpoints	Setup Exclusions Rule Details Checked Files/Design Units Results			
Ready				//

![](_page_12_Picture_0.jpeg)

## 7. <u>解析結果の確認</u>

解析結果は DesignChecker ウィンドウに複数のタブに分かれて表示されます。この章では DesignChecker で表示される解析結果の各項目について説明します。

😵 DesignChecker	r (HDL Designer)			_ <b>D</b> X
File Run Edit	View Setup Exclusions Results Tools Options Help			
🖀 🕶 🖬   🔐 -	· ] & 🖻 🛍   🏚 📫 🗈 ] M 🛛 ] 🏈 🍰 🖓 📭 🖳 🗒			
Setup	Results (Using viewpoint: Severity & Ruleset)		Summary	<u> </u>
Exclusions	Message	∠ Line Number Filename	🖃 Settings	
Results	E Solution Note - 1 item, 2 violations. (1 primary, 1 associated)		Policy:	test_Policy
韋 Expand All	🗉 🔮 Warning - 5 items, 30 violations. (28 primary, 2 associated)		Library:	uart_sample
÷ C.II	e Error - 3 items, 28 violations. (22 primary, 6 associated)		Primary:	uart_top
📮 Cullapse All			Secondary:	uart_top
Den Open			Master Clocks:	clk
The second			Master Resets:	rst
Dpen HDL			Depth:	ThroughDesignRoot
Eta Show Rule			Check Level:	Design Unit Level
🖹 Disable Rule			Exclusions	
🛌 Disable			Number of exclusions in the set	tings:
RuleSet			Policy Disabled Rules	0
Show			Code/Rule Exclusions	0
Violations			Black Boxed Files	0
			Don't Touch Files	0
			Exclusion Pragmas	0
			Pragma Code Excluded	U
			Missing Masters	U
			Unbound Instances	U
			E Design Quality: 166/207 (80%)	
			Rules: (Using policy test_Policy)	
			🗈 Design Units:	
Checked Files/De				
Rule Details		•	۰ III	÷.
Viewpoints	Setup Exclusions Rule Details Checked Files/Design Units Results			
Ready				11.

#### 7-1. Exceptions タブ

解析から除外されたファイル・デザインユニットの一覧です。

ブラック・ボックス、Don't Touch、プラグマ除外により解析対象でないデザインやファイルが表示されます。

DesignCheck	er										) 🗙
File Run Edi	t View Setup Exclus	sions Results '	Tools Option	ns Help							
🖆 + 🖃 👹	•   X 🖻 🛍   🏚	≑ ☎ 鍋(	2   🔮 🛻	品牌科用	12						
Setup	Code/Rule Exclusions	3							Summary		
Exclusions	Object Name	Library	Rule Name		Object Type	Description			Policy Disabled Rules		
Add In 1									Policy:	test	
Exclusion									Disabled Rules:	0	
Edit											
Exclusion	Black Boxed Files	1		1			1		Code/Rule Exclusions		
	File Name	Library	Dialect	File Path			Justification		Libraries:	RULET	
Show Hule									Exclusions:	0	
🖹 🔔 Open File									Exclusions In Play:	0	
- Bemove	•							4			
Exclusion	Don't Touch Files	Library	Dislast	Eile Date			Lucificanticus	≜	Black Boxed Files		
👷 Remove All	riie ivaliie	Libidiy	Dialect	riteraut			Justilication	_	Exclusions:	0	
Exclusions											
									Don't Touch Files		
	Euclusion Program							-	Exclusions:	0	
	Pragma Classes		9	itart Pragma	End Pragma			-			
	🖹 hds, pragma, synthesis	;		hecking_off	checking_o	n		*	Exclusion Pragmas		
	🖹 hds, pragma, synthesis	, exemplar	b	anslate_off	translate_or			E	Excluded Blocks:	23	
	🖹 pragma, synthesis,	exemplar	3	ynthesis_off	synthesis_	on		-	Exclusion Pragmas:	1	
	Pragma Code Excluded	5.10	Le.		le ui	In ur	151 D.4		Unbound Component/Instance	es	
	Start Pragma	End Fragma	1	ile Name Discussed for oddal	Start Line	End Line	File Path		Unbound Instances:	0	
	synthesis_off	synthesis_on	8	control_isit.vnu	70	70	E:/work/HDS/my_project/HDLET/hdf/	ы	Missing Masters:	0	
	synthesis_on	synalesis_on					E. Work and a way project a rock i a daw	*			
				III			4	<u> </u>			
	Unbound Component/Inst Instance Name	Master Name	I F	ile Name	Line	File Path		1			
		index in calle		ine i carrie	010	THO F GAT					
Results											
Checked Files/De											_
Rule Details	Setup Exclusions B	tula Dataile Charl	rad Files /Dasign	Unite Basulte					•		•
viewpoints	Settip Excitisions In		con new pesign	onia nesuis							
neady								_			/

![](_page_13_Picture_0.jpeg)

7-2. Rule Details タブ

解析に使用したルールの一覧を確認することができます。

ルールの詳細ウィンドウには、ポリシー、ルールセットと設定されたルールの名前が表示されます。

DesignCheck	TARBORIS IS IN A RAY-RYN-R-Y-					
File Run Edit	View Setup Exclusions Results Tools Options Help					
📔 • 🖃 🞯	・  3 階 間   車 手 白   み ②   � 鳥 島 陰 陸 陸 周 国					
Setup	Rule Details					
Exclusions	Name	Enabled	Run	Failed	Justification	
Results	test					^
Checked Files/De	test					
Rule Details	Style					
Show	Assignment Alignment	Enabled	Full	Fail		
Rule/RuleSe	Declaration Style	Enabled	Full	Pass		
	Enclosed Block Style	Enabled	Full	Pass		
	File References	Enabled	Full	Pass		-
	Indentation	Enabled	Full	Fail		
	Indentation Style	Enabled	Full	Fail		
	Length	Enabled	Full	Fail		
	Statement Style	Enabled	Full	Pass		
	Mixed Combinational Sequential Code	Enabled	Full	Pass		-
	VHDL Statement Order	Enabled	Full	Fail		
	Verlog Statement Order	Enabled	Full	Pass		
	Essentials					
	Coding Practices					
	Unused Declarations	Enabled	Full	Fail		
	Unassigned Objects	Enabled	Full	Pass		
	Unconnected Input Ports	Enabled	Full	Pass		
	Gated Clocks	Enabled	Full	Pass		
	Internally Generated Clocks	Enabled	Full	Pass		
	Internally Generated Resets	Enabled	Full	Pass		
	Mixed Clocks Resets	Enabled	Full	Pass		
	Consistent Resets	Enabled	Full	Pass		
	Multiple Drivers	Enabled	Full	Pass		
	Matching Range	Enabled	Full	Pass		
	FSM Transitions	Enabled	Full	Pass		
	Logical and Bitwise Operators	Enabled	Full	Pass		
	Sub-Program Body	Enabled	Full	Pass		*
Viewpoints	Setup Exclusions Rule Details Checked Files/Design Units Results					
Ready						

7-3. Checked Files / Design Unit タブ

解析対象となったファイルやデザインユニットの一覧を確認することができます。

📝 DesignChecke	r	-	-			-			-		
File Run Edit	View Setup Exclusions	Results	Tools C	Options Help							
🖀 • 🖬 😡	-  X 🖻 🛍   🏚 🚔   (	白人的	2 🛛 🎱	1 <b>8 8</b> 12 24							
Setup	Checked Files		1.5								
Exclusions	Name	Order	Language	Syntax Error(s)	Code-Rule 8	Excluded Black B	oxed Library	Path			
Results	E filter_rtl.vhd	1	VHDL	No	No	No	RULET	E:\work	\HDS\my_projec	t\RULET\hdl	
Checked Files/De	inc_struct.vhd	2	VHDL	No	No	No	RULET	E:\work	HDS\my_projec	t\RULET\hdl	
🗗 🕰 Open	control_fsm.vhd	3	VHDL	No	No	No	RULET	E:\work	\HDS\my_projec	t\RULET\hdl	
T = Source	counter_flow.vhd	4	VHDL	No	No	No	RULET	E:\work	\HDS\my_projec	t\RULET\hdl	
	decoder_tbl.vhd	5	VHDL	No	No	No	RULET	E:\work	HDS/my_projec	t\RULET\hdl	
	E selector_struct.vhd	6	VHDL	No	No	No	RULET	E:\work	HDS/my_projec	t\RULET\hdl	
	top_struct.vhd	7	VHDL	No	No	No	RULET	E:\work	HDS\my_projec	t\RULET\hdl	
	Checked Design Units Primary	Design U	Init Type	Secondary	BTL	Code-Rule Excluded	Black Boxed	Language	Library	File Name	 File Path
	Control	Entity			RTL	No	No	VHDL	RULET	control fsm.vhd	E:\work\HDS\mv_pro
	Control	Architect	ure	fsm	RTL	No	No	VHDL	RULET	_ control_fsm.vhd	E:\work\HDS\mv_pro
	词 counter	Entity			RTL	No	No	VHDL	RULET	_ counter_flow.vhd	E:\work\HDS\my_pro
	💹 counter	Architect	ure	flow	RTL	No	No	VHDL	RULET	counter_flow.vhd	E:\work\HDS\my_pro
	decoder	Entity			RTL	No	No	VHDL	RULET	decoder_tbl.vhd	E:\work\HDS\my_pro
	decoder	Architect	ure	tbl	RTL	No	No	VHDL	RULET	decoder_tbl.vhd	E:\work\HDS\my_pro
	💹 filter	Entity			RTL	No	No	VHDL	RULET	filter_rtl.vhd	E:\work\HDS\my_pro
	📜 filter	Architect	ure	rti	RTL	No	No	VHDL	RULET	filter_rtl.vhd	E:\work\HDS\my_pro
	📜 inc	Entity			RTL	No	No	VHDL	RULET	inc_struct.vhd	E:\work\HDS\my_pro
	📜 inc	Architect	ure	struct	RTL	No	No	VHDL	RULET	inc_struct.vhd	E:\work\HDS\my_pro
	🔚 selector	Entity			RTL	No	No	VHDL	RULET	selector_struct.vhd	E:\work\HDS\my_pro
	selector	Architect	ure	struct	RTL	No	No	VHDL	RULET	selector_struct.vhd	E:\work\HDS\my_pro
	📜 top	Entity			RTL	No	No	VHDL	RULET	top_struct.vhd	E:\work\HDS\my_pro
	📜 top	Architect	ure	struct	RTL	No	No	VHDL	RULET	top_struct.vhd	E:\work\HDS\my_pro
Rule Details	•					m					•
Viewpoints	Setup Exclusions Rule Del	tails Ch	ecked Files/D	Design Units Results							
Ready											1.

![](_page_14_Picture_0.jpeg)

7-4. Results タブ

ルールをもとに解析した結果(エラー / ワーニング / ノート) が表示されます。

下記のようにエラーやワーニングの種類、対象となったルール、対象のファイル・行数という風にカテゴリー分け されてツリー表示されます。

Error - 10 items, 518 violations. (508 primary, 10 associated)					
🖃 🔄 ERROR : Essentials\Naming - Unique Names - 1 item, 2 violations. (1 primary, 1 associated)					
RULET, selector, Architecture - 1 item, 2 violations. (1 primary, 1 associated)					
□ Rule Name: Unique Names - 1 item, 2 violations. (1 primary, 1 associated)					
🖂 🚍 selector - 1 item, 2 violations. (1 primary, 1 associated)					
🖃 \Theta selector_struct.vhd (130) : [ERROR] Naming, Unique Names - Instance name "U_3" inside architectu	re 'struct' is not unique thro	ughout the desig			
🖃 🥥 selector_struct.vhd (130) : [ERROR] Essentials\Naming, Unique Names - Instance name 'U_3' insi	de architecture 'struct' is no	t unique through			
RulePath: Essentials/Naming - 1 item, 2 violations. (1 primary, 1 associated)					
Instance name 'U_3' inside architecture 'struct' is not unique throughout the design.	130				
127 sum ⇒ sum,					
128 valid => valid					
129 );		ゟ゙゚゙ヺ゙ゕゟ			
130 U_3 : inc ーマンフレク					
131 PORT MAP (					
Use unique FSM states & instance names throughout the design.					
① Associated Violations - 1 item 1 associated violation					

ッリーを開くとソースが表示されます。こちらをダブルクリックすることによって、テキストエディタにて対象のファ イルを開くことができます。ファイルを開くと、下記のようにエラー / ワーニングとなっている箇所がハイライトされ、 ウィンドウ下部にメッセージの内容が表示され、ソースファイル上でどこがエラーになっているのかを確認しながら 修正することも可能です。

![](_page_14_Figure_7.jpeg)

![](_page_15_Picture_0.jpeg)

また、Results タブには Summary の項目があり、解析結果のまとめが表示されています。

1 Settings

解析に使用したポリシー名や、デザイン名等が表示されます。

Summary 🔳							
🖃 Settings							
Policy:	test						
Library:	RULET						
Primary:	top						
Secondary:	struct						
Master Clocks:	clk						
Master Resets:	rst_n						
Depth:	ThroughDesignRoot						
Check Level:	Design Unit Level						

② Exclusions ペイン

最新の分析結果に影響を与える除外設定だけに関する情報が表示されます。最新の実行に関与している ポリシー、無効のルール、コード/ルールの除外、ブラックボックスファイル、および除外プラグマの数などです。

Exclusions						
Number of exclusions in the settings:						
Policy Disabled Rules	0					
Code/Rule Exclusions	0					
Black Boxed Files	0					
Don't Touch Files	0					
Exclusion Pragmas	1					
Pragma Code Excluded	23					
Missing Masters	0					
Unbound Instances	0					

③ Rules ペイン

#### 使用したポリシー内の各ルールに対して、違反となったルールの個数を表示します。

iles Setting, grouped by Rule	set:						
RuleSet	Failed	Total	Failed %	Disabled	Not-Run	Partially-Run	Fully-Run
All	12	65	18.46%	0	0	0	65
test	5	11	45.45%	0	0	0	11
Style	5	11	45.45%	0	0	0	11
Essentials	7	54	12.96%	0	0	0	54
Coding Practices	2	19	10.53%	0	0	0	19
Downstream Checks	1	22	4.55%	0	0	0	22
Code Reuse	0	6	0.00%	0	0	0	6
Naming	4	7	57.14%	0	0	0	7

#### ④ Design Units ペイン

各デザインユニットのルール違反を表示します。

Failed Design Unit types:				
Design Unit	Failed	Total	Failed %	Excluded
All	12	21	57.14%	0
File	0	7	0.00%	0
Module	0	0	0.00%	0
Interface	0	0	0.00%	0
Program Block	0	0	0.00%	0
Configuration	0	0	0.00%	0
SV Package	0	0	0.00%	0
Class	0	0	0.00%	0
Package Header	0	0	0.00%	0
Package Body	0	0	0.00%	0
Architecture	5	7	71.43%	0
Entity	7	7	100.00%	0

#### ⑤ Violations ペイン

#### 設計対象物の種類ごとエラー、ワーニング、ノートの数が表示されます。

∃ Violations: 510			
Number of primary violations for e	ach severit	y:	
Syntax Error	0		
Elaboration Error	0		
Synthesis Error	0		
Error	508 from 1	0 Rules	
Warning	1 from 1 R	ule	
Note	1 from 1 R	ule	
Number of primary violations for e	ach scope:		
Scope	Violations	%	
File	0	0.00%	
Unknown	0	0.00%	
Configuration	0	0.00%	
Package Header	0	0.00%	
Package Body	0	0.00%	
Module	0	0.00%	
Interface	0	0.00%	
SV Package	0	0.00%	
Program Block	0	0.00%	
Class	0	0.00%	
Architecture	411	80.59%	
Entity	99	19.41%	•

![](_page_17_Picture_0.jpeg)

テーブル内の行をダブルクリックすると、各スコープの違反の詳細を表示することができます。

DesignChecke	er de la companya de	A local or management of \$1.000					
File <u>R</u> un Edit	View <u>S</u> etup Exclusions Res <u>u</u> lts <u>T</u> ools <u>O</u> ptions <u>H</u> elp						
🏠 🕶 🖬 🕼	•   ※ 🖻 🛍   章 🌲 🗈   州 ②   🥝 🏭 🏭 🏪 🗏 🗒						
Setup	Results (Using viewpoint: Violations for scope Entity) (Filtered) - Some objects have be	een hidden 🔳					
Exclusions	Message	Line Number File and Line					
Results	🗏 🖻 🖀 Entity - 3 items, 104 violations. (99 primary, 5 associated)						
韋 Expand All	Rule Name: Avoid Feedthroughs - 1 item, 3 violations. (1 primary, 2 associated)						
≑ Collapse All	e All O Peedthrough detected from input port 'result' to output port 'valid' in design unit 'selector'. 17 selector_struct.vhd (17) 14 PDRT(						
<b>Ça</b> ≜ Open I≣ Source	15 clk : IN std_logic; 16 cnt : IN std_logic_vector (2.DDW/NTO 0);						
🕒 Open HDL	17 result : IN std_logic; 18 rst n : IN std_logic;						
🐏 Show Rule	Avoid coding style that implies feedthroughs. ⊕ Associated Violations - 2 items, 2 associated violations.						
🖹 Disable Rule	Rule Name: Avoid Asynchronous Reset Release → 1 item, 4 violations. (1 primary, 3 associated)						
Disable RuleSet	Rule Name: Capitalization - 97 items, 97 violations. (97 primary, 0 associated)						
Remove							

## 8. 解析レポートの出力

DesignChecker のレポートを CSV、TSV、HTML 形式で出力することができます。HTML など汎用的なファイル形式にて出力することによって、HDL Designer を起動せずに解析結果のレポートを参照することが可能となります。

- ① DesignChecker ウィンドウより「File メニュー」>「Export Results..」を選択します。
- ② Export Results ウィンドウにて出力形式等を選択し「OK」をクリックすると各指定したファイル形式で出力 されます。

![](_page_17_Figure_8.jpeg)

![](_page_18_Picture_0.jpeg)

#### 9. Appendix

9-1. DesignChecker ルールの構成

DesignChecker で使用するルールは、ベースルール、ルールセット、ポリシーの 3 つの階層で構成されています。

• ベースルール

パラメーター設定可能な基本ルールで、このベースルールをもとにパラメーターをカスタマイズすることに よってオリジナルルールを作成することが可能です。130以上の豊富なルールがそろっています。

ルールの一覧は下記をご参照ください。

<PDF>

<インストールディレクトリ>¥docs¥htmldocs¥baserules¥ base\_rules.pdf

<u><HTML></u> ※Web ブラウザで一覧を表示することができます

<インストールディレクトリ>/docs/htmldocs/baserules/BaseRuleTOC.htm

ルールセット

ベースルールを組み合わせて作成されたセットです。あらかじめ基本的なルールをまとめた標準ルールセットが用意されています。独自のルールセットを作成することも可能です。

<標準ルールセット>

- ♦ Essentials
- ♦ RMM
- ♦ Safety-Critical
- ♦ Checklist
- ♦ DO-254 など
- ポリシー

ポリシー単位で解析が行われます。ルールセットを組み合わせてポリシーを作成することができ、標準ル ールセットに対応するポリシーはあらかじめ設定されています。

![](_page_18_Figure_22.jpeg)

![](_page_19_Picture_0.jpeg)

### <u> 改版履歴</u>

Revision	年月	概要
1	2017 年 10 月	初版

#### 免責およびご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

- 1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
- 2. 本資料は予告なく変更することがあります。
- 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、本資料を入手されました下記代理店までご一報いただければ幸いです。
   株式会社マクニカ アルティマ カンパニー <a href="https://www.alt.macnica.co.ip/">https://www.alt.macnica.co.ip/</a> 技術情報サイト アルティマ技術データベース <a href="https://www.alt.main.ip/members/">http://www.alt.macnica.co.ip/</a> 技術情報サイト アルティマ技術データベース <a href="https://www.alt.main.ip/members/">http://www.alt.macnica.co.ip/</a> 技術情報サイト ETS <a href="https://www.alt.main.ip/members/">https://www.alt.macnica.co.ip/</a> 技術情報サイト ETS <a href="https://www.alt.main.ip/members/">https://www.alt.macnica.co.ip/</a>
- 4. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。
- 5. 本資料は製品を利用する際の補助的な資料です。製品をご使用になる際は、各メーカ発行の英語版の資料もあわせてご利用ください。