

# ALINT 2010.10 新機能概要



ALDEC



# ALINT 2010.06以前のバージョンからの アップデート時の注意

- ALINT 2010.06 以前のバージョンでコンパイルされたライブラリの情報、およびリント結果のデータベース (AVDB) は ALINT 2010.10 で使用不可



# 改善内容の概要

- フェーズベースリンティング（PBL）の改善
  - ◆ フローエディタ
  - ◆ フローウィザード
- チップレベルの解析ルールを追加
  - ◆ 階層設計ルール
  - ◆ 内部クロックとリセットの識別ルール
- 除外ルールの設定
  - ◆ 違反データベースとの統合
- その他の改善

# PBLーフロー管理機能の改善

- フェーズベースリンティングの操作性を改善

- ◆ フローエディタ

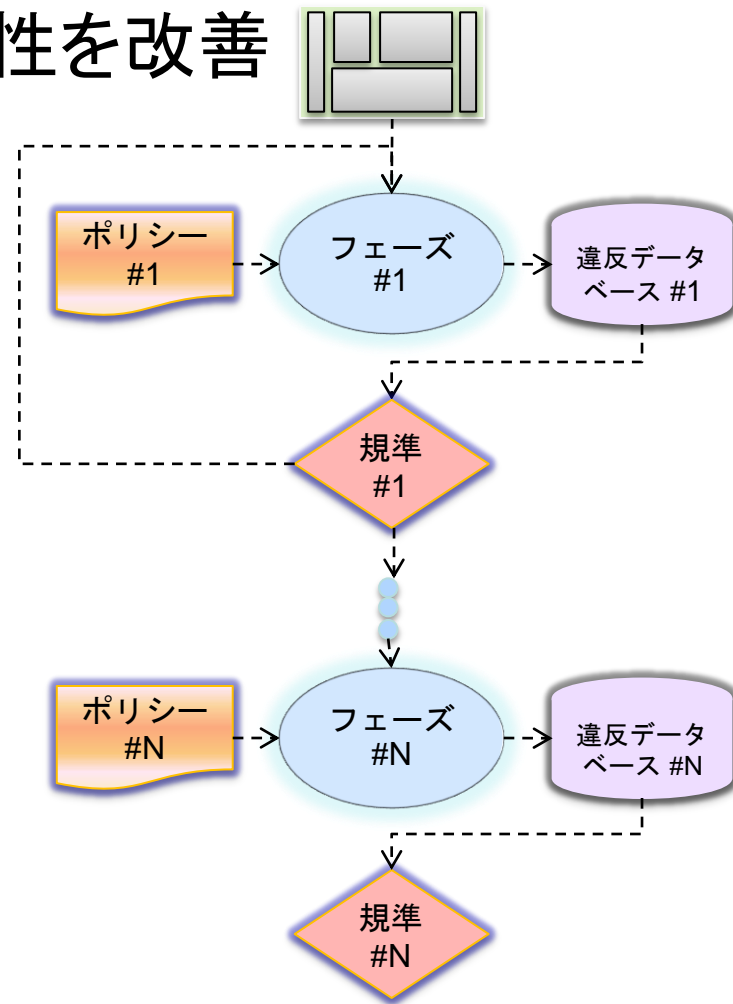
- カスタムフローを容易に作成

- ◆ 階層フロー

- フロー構造の表示と実行

- ◆ フローウィザード

- 定義済のルールセットに基づいてカスタムフローを迅速に作成



# PBL – フローエディタ 1

- 専用のフローエディタを追加
  - ◆ フローウィンドウとコンテキスト対応ウィンドウ

The screenshot shows the 'Clock and reset checking.alintflow\*' window. The left pane displays a tree view of the flow hierarchy: 'Clock and reset checking' > 'Clock and Reset' > 'Clock Domain Crossings'. The right pane shows a 'Flow summary' table with columns for phase names and counts.

Phase Name	Count
Pl	3
Tr	3
R	3
C	0
R	615
R	45
P	0
used in optional phases:	0

**1** フローウィンドウ

1. フェーズの追加、削除、名前の変更
1. フェーズの実行順序や階層構造の変更

**2** コンテキスト対応ウィンドウ

1. フェーズをパスする  
規準のカスタマイズ
2. フェーズポリシーの編集
3. オプションの設定

# PBL - フローエディタ 2

- フェーズのパス規準のカスタマイズが可能

1 パスするルール割合で設計品質規準を指定

Name	Quality	Critical...
Clock and reset checking		
Clock and Reset		
Clock Domain Crossings	Q - 80%	R - 1
Policy		
Preferences		
General Clock Guidelines	Q - 100%	
General Reset Guidelines	Q - 100%	

Quality: 80

2 クリティカル・ルールを選択

3 必須またはオプションの指定

# PBL – フローエディタ 3

- フェーズポリシーの編集やカスタマイズが可能

Name	Quality	Critical...
Clock and reset checking		
Clock and Reset		
Clock Domain Crossings	Q - 80%	R - 1
Policy		
Preferences		
General Clock Guidelines	Q - 100%	
General Reset Guidelines	Q - 100%	

1 ポリシーエディタにて、フェーズポリシーの構成内容を修正

2 パラメータエディタを使って、ルール固有の設定が可能

# PBL - フローエディタ 4

- フェーズのオプション設定が可能

1 @alint プラグマの扱い方法

2 ワーニング数の制限

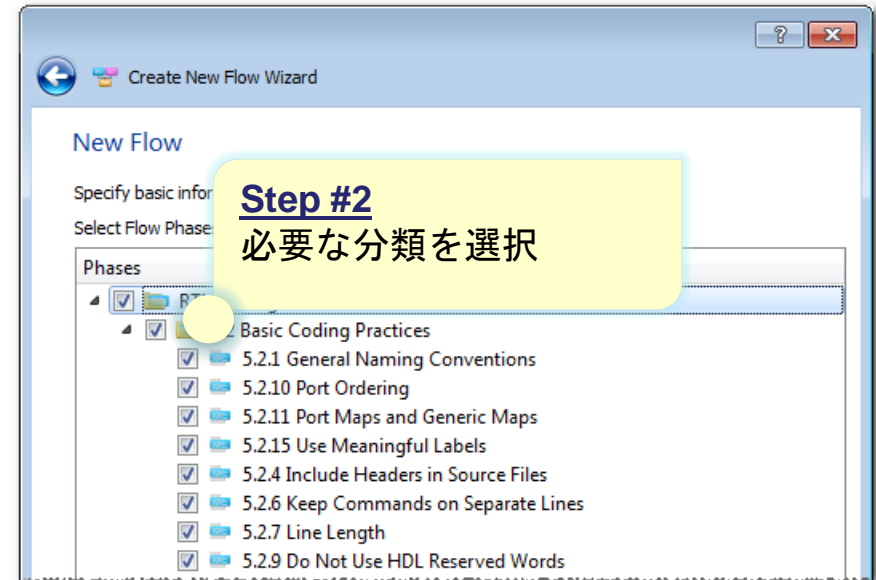
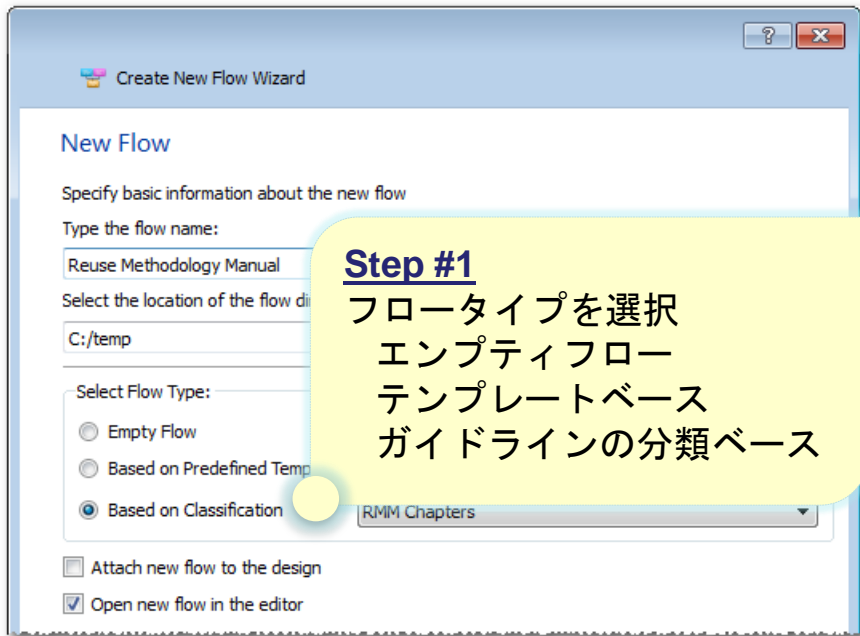
3 違反情報レポートの設定

4 コマンド用追加オプション

- フロー設定ファイル .alintflow を自動生成

# PBL - フローウィザード

- メニューの File -> New -> Flow
  - ◆ Based on Classification オプションを追加
    - ガイドラインの定義に対応した分類を提供
    - フローに含むべきルール項目を選択するだけで、カスタムフローを迅速に作成することが可能

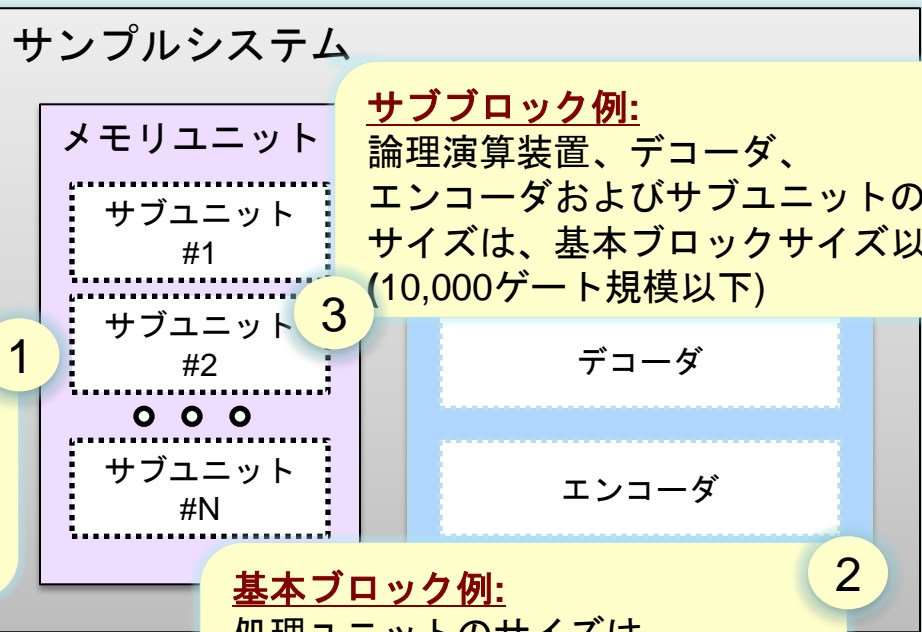


# 階層設計ルール追加 1

- デザインのインスタンスを分類
  - ◆ デザインを独立した階層セグメントに分割
  - ◆ ゲート規模に応じて、いずれかのカテゴリにデザインユニットを関連付ける
    - 上位レベルブロック
    - 基本ブロック
    - サブブロック

## 上位レベルブロック例:

メモリユニットのサイズは、基本ブロックサイズ以上 (デフォルトでは40,000 ゲート規模以上)



## サブブロック例:

論理演算装置、デコーダ、エンコーダおよびサブユニットのサイズは、基本ブロックサイズ以下 (10,000ゲート規模以下)

## 基本ブロック例:

処理ユニットのサイズは、基本ブロックサイズの範囲内 (10,000~40,000ゲート規模)

# 階層設計ルールを追加 2

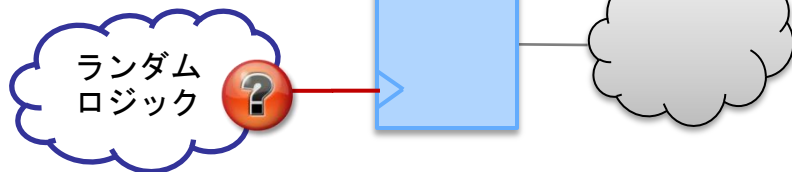
- デザインユニットのカテゴリ別分類は、レイアウトの最適化に有効
- STARC\_VHDL / STARC\_VLOG の階層設計ルールのサポート
  - ◆ STARC\_VHDL.1.6.1.4 / STARC\_VLOG.1.6.1.4
  - ◆ STARC\_VHDL.1.6.2.1 / STARC\_VLOG.1.6.2.1
  - ◆ STARC\_VHDL.1.6.4.1 / STARC\_VLOG.1.6.4.1

# 内部クロックとリセットの識別ルールを追加 1

- 内部クロックとリセットを識別するための専用メカニズムを実装
  - クロックとリセットの生成スキームの正確性を検証
  - 内部生成された信号と他のデザインエレメントの接続を確認

## 2010.06以前:

外部ポート接続のみを  
クロックとして認識 →  
疑似エラーが検出される  
ケースが発生



内部  
クロック  
の生成



## 2010.10:

クロックとリセットの生成  
スキームを、専用ルールで  
詳細に分析



# 内部クロックとリセットの識別ルールの追加 2

- 内部クロック識別ルール例

- ◆ ALDEC.3305

- 内部クロックが存在 → 次のガイドラインに従って設計すること
      - クロック生成論理は、別のデザインユニットに分離
      - クロック制御以外で使用するいかなる論理とも混在させない

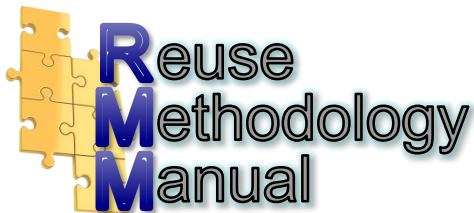
# 追加ルール



- ALDEC\_VHDL : 2 ルール追加 (合計 24)
- ALDEC\_VLOG : 2 ルール追加 (合計 48)

## STARC

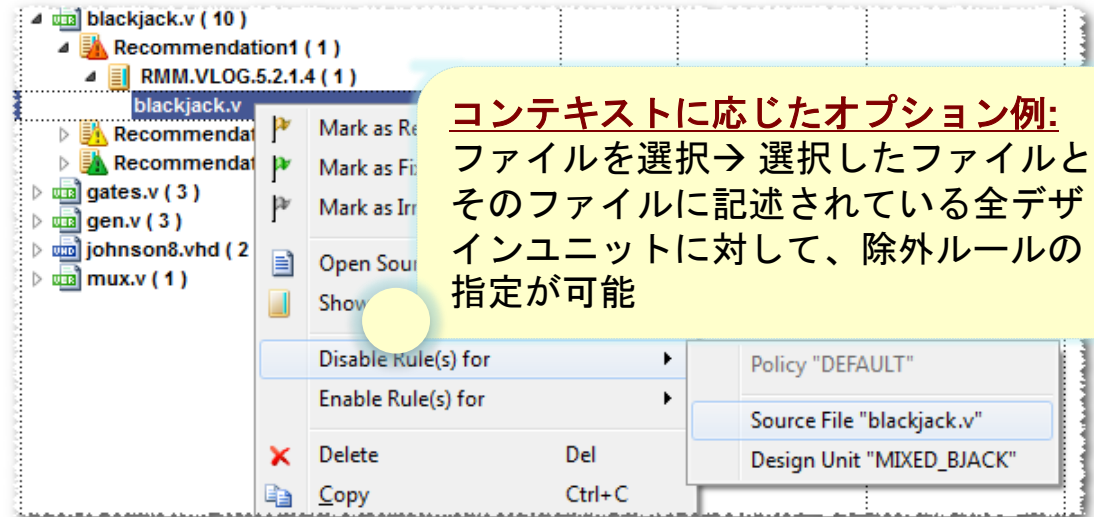
- STARC\_VHDL : 4 ルール追加 (合計160)
- STARC\_VLOG : 4 ルール追加 (合計202)



- RMM [VHDL and Verilog] : 4 ルール追加 (合計 80)

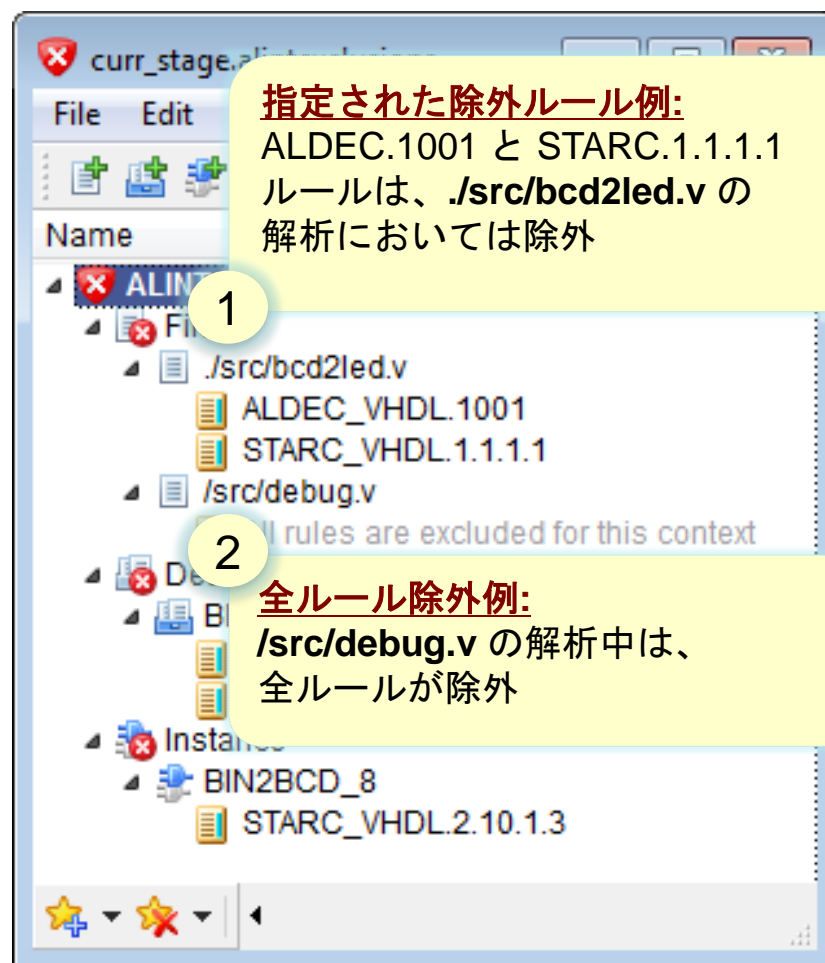
# エクスクルージョンエディタの改善 1

- バイオレーションビューワと統合
  - ◆ コンテキストメニューにより、除外ルールの指定が容易
  - ◆ 解析中のルールについても、解析の対象から指定を外すことが可能
    - ソースファイル
    - デザインユニット
    - インスタンス



# エクスクルージョンエディタの改善 2

- 専用のエクスクルージョンエディタの使用によりカレントデザインに対する除外オプションの指定が容易
  - ◆ 選択したエレメントに対して指定した除外ルールを表示
  - ◆ 全ルールの除外も指定可能



# その他の改善

- データベース (AVDB) 比較ダイアログボックスの改善
- セベリティレベルに Critical Warning を追加
- マクロコマンドオプションの変更
  - ◆ -alint\_cdclog → -alint\_synthesislog
    - チップレベルのデザイン解析情報をテキストフォーマットで出力
      - 検出された外部・内部クロックおよびリセットの情報
      - クロックドメインの情報
      - レジスタ間のドライブ情報
      - デザインのゲート規模の見積り
  - ◆ -alint\_blackbox → -alint\_skipunit
    - ALINT のチェックから除外するデザインユニットの指定