

RTL設計中級 論理合成 … 目次

第0章 論理合成とは

1. 論理合成とは
2. 従来の論理設計のフロー
3. HDLを使った論理設計のフロー
4. 論理合成のメリット
5. 論理合成ツールの限界
6. 論理合成可能な記述
7. ASICとFPGAの合成ツール
8. ASICとFPGAの合成手法の違い
9. ツールやバージョンによる結果の違い

第1章 論理合成を前提としたRTL記述

1. 第1章 アジェンダ
2. 論理合成の基本フロー
3. RTL解説
4. 算術演算回路
5. 算術演算の記述
6. 演算子使用時の注意点
7. 加算演算子の回路構成
8. 演算子の共有
9. case文から生成されるセレクタ
10. if文から生成されるセレクタ
11. default項に不定値を代入
12. default項に固定値を代入
13. ラッチを生成する危険を避ける(1)
14. ラッチを生成する危険を避ける(2)
15. ラッチを生成する危険を避ける(3)
16. ラッチを生成する危険を避ける(4)
17. for文記述
18. for文による組み合わせ回路の記述
19. FFの記述
20. FF記述の制限
21. 代入式右辺の遅延値
22. 10進カウンタ記述
23. ブロッキング代入とノンブロッキング代入
24. ラッチの記述
25. ゲート vs. RTLシミュレーション不一致問題(1)
26. ゲート vs. RTLシミュレーション不一致問題(2)
27. ゲート vs. RTLシミュレーション不一致問題(3)
28. ゲート vs. RTLシミュレーション不一致問題(4)

第2章 論理合成の基本とスクリプト

1. 第2章 アジェンダ
2. 論理合成の対象(1)
3. 論理合成の対象(2)
4. 論理合成の対象(3)
5. 論理合成の基本フロー
6. 論理合成のアルゴリズム
7. dc_shell と Design Vision(1)
8. dc_shell と Design Vision(2)
9. dc_shellスクリプト
10. 標準スクリプト
11. 論理合成基本コマンド(1)
12. 論理合成基本コマンド(2)
13. 論理合成基本コマンド(3)
14. オブジェクト
15. タイミングのパス
16. report_area
17. report_reference
18. report_timing
19. report_timing 2
20. 論理合成基本コマンド(4)
21. 論理合成の制約
22. 論理合成基本コマンド(5)
23. 制約条件 - set_operating_conditions
24. 制約条件 - set_wire_load_model
25. 制約条件 - ポーと名の指定
26. 制約条件 - set_driving_cell
27. 制約条件 - set_load
28. 制約条件 - create_clock
29. 制約条件 - set_input_delay, set_output_delay
30. 制約条件 - set_max_area
31. 制約条件 - set_ideal_network
32. report_lib
33. allコマンド
34. getコマンド
35. DCセットアップファイル - .synopsys_dc.setup
36. クロックとリセット
37. 組み合わせ回路の制約

第3章 RTL記述スタイル

1. 第3章 アジェンダ
2. 回路構造を意識した記述
3. always文による組み合わせ回路記述の注意点
4. always文内に複数の信号を記述しない
5. if文のスタイル1
6. if文のスタイル2
7. case文のスタイル
8. リソースシェアリング
9. リソースシェアリングされない記述
10. リソースシェアリングの注意点(1)
11. リソースシェアリングの注意点(2)
12. ステートマシン
13. ミーリ型ステートマシン
14. ムーア型ステートマシン
15. 基本はムーア型
16. ステートマシンの記述例
17. ステートマシンの記述例(つづき)
18. 何故、FFとcase文を分離するのか？
19. ステートマシン(バイナリー型)
20. ステートマシン(One Hot型)
21. パラメタライズ記述

付録 Design CompilerでのTcl

1. 簡単なコマンド例
2. 変数の設定と参照
3. 特殊文字
4. リストとコレクション
5. コレクション
6. get_object_name
7. 文字列の作成
8. リストの操作
9. 算術演算
10. スクリプトファイルの読み込み
11. 出力先の変更
12. フロー制御
13. ループ foreach
14. ループ foreach_in_collection
15. ループ while
16. ループ for
17. 条件分岐 if
18. ループの終了

DesignVisionクイックリファレンス

第4章 論理合成の実際

1. 第4章 アジェンダ
2. 基本階層合成
3. 基本階層合成スクリプト例(1)
4. 基本階層合成スクリプト例(2)
5. 基本階層は一括合成
6. 余計な階層は破壊する(ungroup)
7. 階層合成 - ungroup
8. 階層合成 - dont_touch
9. 階層合成 - Design Ware
10. 階層合成 - 基本階層より上は最適化
11. Design Compiler Ultra
12. タイミング解析
13. 最大遅延解析(1)
14. 最大遅延解析(2)
15. 最大遅延解析(3)
16. 最大遅延解析(4)
17. 最大遅延解析(5)
18. タイミング解析のパス
19. タイミング解析レポート - in2out
20. タイミング解析レポート - in2ff
21. タイミング解析レポート - ff2ff
22. タイミング解析レポート - ff2out
23. クロックの遅延
24. クロックスキューによる誤動作
25. ホールドタイム保証
26. クロックツリーシンセシス(CTS)
27. set_fix_multiple_port_netsコマンド
28. change_namesコマンド
29. ログファイルとレポートファイルの確認
30. Tcl - システムコマンド
31. Tcl - foreach文
32. Tcl - if文
33. helpとman