

目次

1 はじめに

- 1-1 HDLとは
- 1-2 Verilog HDLの歴史
- 1-3 3つの記述レベル — 設計の抽象度
- 1-4 HDLによるASIC開発手順
- 1-5 HDL設計 (RTL設計) の手順
- 1-6 RTL設計
- 1-7 RTL検証
- 1-8 論理合成
- 1-9 ゲートレベル検証
- 1-10 Verilog HDLの文法体系

2 簡単な組み合わせ回路の記述

- 2-1 加算演算子による加算回路
- 2-2 フルアダーによる加算回路
- 2-3 フルアダーの動作
- 2-4 フルアダーの記述
- 2-5 フルアダーによる加算回路の記述 1
- 2-6 フルアダーによる加算回路の記述 2
- 2-7 ポート接続について 1
- 2-8 記述に関する規則

3 簡単な順序回路の記述

- 3-1 バイナリカウンタの動作
- 3-2 バイナリカウンタの記述 (1)
- 3-3 バイナリカウンタの記述 (2)
- 3-4 順序回路記述のポイント

4 回路の検証

- 4-1 回路動作確認のためのテストパターン1
- 4-2 回路動作確認のためのテストパターン2
- 4-3 回路動作確認のためのテストパターン3
- 4-4 回路シミュレーション
- 4-5 テストベンチ
- 4-6 テストベンチの記述構造 (1)
- 4-7 テストベンチの記述構造 (2)
- 4-8 シミュレーション時のファイル指定
- 4-9 加算回路のシミュレーション例
- 4-10 クロックの記述
- 4-11 カウンタのテストベンチ
- 4-12 カウンタのシミュレーション例
- 4-13 テストベンチのHDL記述構造

5 基本文法1

- 5-1 モジュールの構造 (1)
- 5-2 モジュールの構造 (2)
- 5-3 ポート宣言
- 5-4 信号の宣言と型
- 5-5 ネット型と変数型の使い分け (1)
- 5-6 ネット型と変数型の使い分け (2)
- 5-7 ネット型と変数型の使い分け (3)
- 5-8 ネット型と変数型の使い分け (4)
- 5-9 ビット幅
- 5-10 複数ビット信号 (1)
- 5-11 複数ビット信号 (2)
- 5-12 パラメータ宣言
- 5-13 assign文
- 5-14 function
- 5-15 always文
- 5-16 initial文
- 5-17 if文
- 5-18 case文
- 5-19 begin~end
- 5-20 モジュール文でのポート宣言

6 基本文法2

- 6-1 論理値と数値表現
- 6-2 演算子 1
- 6-3 演算子 2
- 6-4 演算子 3
- 6-5 ビット演算と論理演算
- 6-6 リダクション演算
- 6-7 条件演算
- 6-8 接続演算
- 6-9 演算の優先順位 1
- 6-10 演算の優先順位 2
- 6-11 ビット幅の不一致

7 基本記述スタイル1

- 7-1 4つの記述スタイル
- 7-2 assign文による組み合わせ回路
- 7-3 functionによる組み合わせ回路
- 7-4 always文による組み合わせ回路
- 7-5 組み合わせ回路記述上の留意点
- 7-6 always文による順序回路-1
- 7-7 always文による順序回路-2
- 7-8 順序回路記述上の留意点
- 7-9 記述と並行動作
- 7-10 defaultと代入信号

目次

8 基本記述スタイル2

- 8-1 ポートリストの記述スタイル
- 8-2 下位モジュール接続 (1)
- 8-3 下位モジュール接続 (2)
- 8-4 ポート接続の注意点 (1)
- 8-5 ポート接続の注意点 (2)
- 8-6 可読性を考慮した記述 (1)
- 8-7 可読性を考慮した記述 (2)
- 8-8 可読性を考慮した記述 (3)

9 セレクタの記述

- 9-1 if文によるセレクタ
- 9-2 case文によるセレクタ
- 9-3 ビット選択によるセレクタ
- 9-4 条件演算子によるセレクタ

10 デコーダとエンコーダの記述

- 10-1 デコーダとエンコーダ
- 10-2 if文によるデコーダ
- 10-3 case文によるデコーダ
- 10-4 casex文, casez文によるデコーダ
- 10-5 if文によるエンコーダ
- 10-6 case文によるエンコーダ
- 10-7 プライオリティエンコーダとは
- 10-8 if文によるプライオリティエンコーダ
- 10-9 casex文によるプライオリティエンコーダ
- 10-10 if文とcasex文の動作の違い

11 演算回路の記述

- 11-1 組み合わせ回路によるシフト回路
- 11-2 乗算回路
- 11-3 大小比較回路と一致比較回路
- 11-4 ROMで作る演算回路

12 順序回路の基礎とFF

- 12-1 2つの代入
- 12-2 2つの代入による回路 (1)
- 12-3 2つの代入による回路 (2)
- 12-4 非同期リセットと同期リセット
- 12-5 初期化には非同期リセット
- 12-6 論理合成できない順序回路記述
- 12-7 ラッチ
- 12-8 イネーブル付きFF

13 カウンタとシフトレジスタの記述

- 13-1 N進カウンタ (1)
- 13-2 N進カウンタ (2)
- 13-3 アップダウンカウンタ
- 13-4 任意のカウント値をもつカウンタ (1)
- 13-5 任意のカウント値をもつカウンタ (2)
- 13-6 分周回路
- 13-7 シリアルパラレル変換回路
- 13-8 立ち上がり検出回路
- 13-9 レジスタ回路

14 パラメタライズ

- 14-1 回路のパラメータ化
- 14-2 モジュールのパラメータ化
- 14-3 パラメータ化モジュールの呼び出し
- 14-4 複数パラメータの利用
- 14-5 特定パラメータの指定

15 ステートマシン記述例

- 15-1 ステートマシン
- 15-2 ステートマシンの用途
- 15-3 ステートマシンの回路
- 15-4 ステートマシンの動作 1
- 15-5 ステートマシンの動作 2
- 15-6 ステートマシンの動作 3
- 15-7 ステートマシンの動作 4
- 15-8 トレイ制御回路の入出力とステート名
- 15-9 ステートマシンの記述
- 15-10 トレイ制御回路の記述 1
- 15-11 トレイ制御回路の記述 2
- 15-12 ステートレジスタの構成
- 15-13 ワンホット方式の記述

目次

16 テストベンチ向き文法

- 16-1 テストベンチのHDL記述構造
- 16-2 データタイプ
- 16-3 タイミング制御
- 16-4 評価と代入の遅延
- 16-5 forループ
- 16-6 whileループ
- 16-7 repeatループ
- 16-8 begin～endとfork～join
- 16-9 階層アクセス

演習

実習・演習問題リスト
演習1～20

17 システムタスク

- 17-1 システムタスクとは
- 17-2 システムタスクの呼び出し
- 17-3 画面表示のシステムタスク (1)
- 17-4 画面表示のシステムタスク (2)
- 17-5 画面表示のシステムタスク (3)
- 17-6 メモリ内容を読み込むシステムタスク
- 17-7 メモリ内容を読み込むシステムタスクの記述例
- 17-8 ファイル出力のシステムタスク (1)
- 17-9 ファイル出力のシステムタスク (2)
- 17-10 値を返すシステムタスク
- 17-11 中断／終了するシステムタスク

18 テストベンチの各種テクニック

- 18-1 入力信号の印加遅延
- 18-2 ループ制御
- 18-3 ループ制御の記述例
- 18-4 期待値の生成と比較
- 18-5 出力値の保存
- 18-6 出力値保存の記述例
- 18-7 文字列
- 18-8 シミュレーションモデルとは
- 18-9 ROMシミュレーションモデル
- 18-10 RAMシミュレーションモデル

19 taskを用いたテストベンチ

- 19-1 taskとは
- 19-2 taskの定義と呼び出し
- 19-3 taskによるテストベンチ記述例
- 19-4 taskの引き数
- 19-5 taskの効果的な利用
- 19-6 taskによるバス動作の記述
- 19-7 taskとfunctionの相違点

20 コンパイラ指示子

- 20-1 コンパイラ指示子とは
- 20-2 define
- 20-3 include (1)
- 20-4 include (2)
- 20-5 timescale
- 20-6 defineとparameterの違い