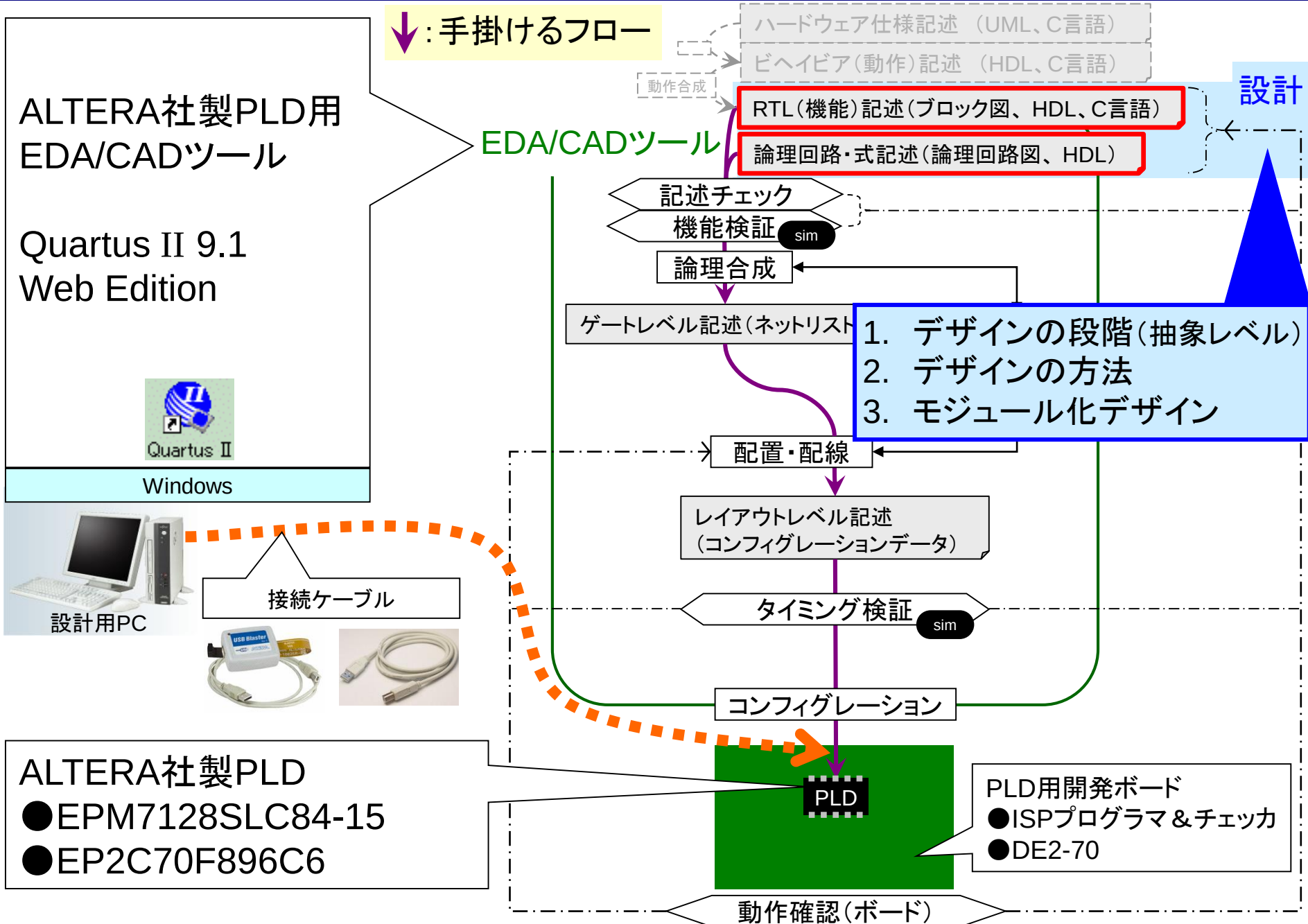
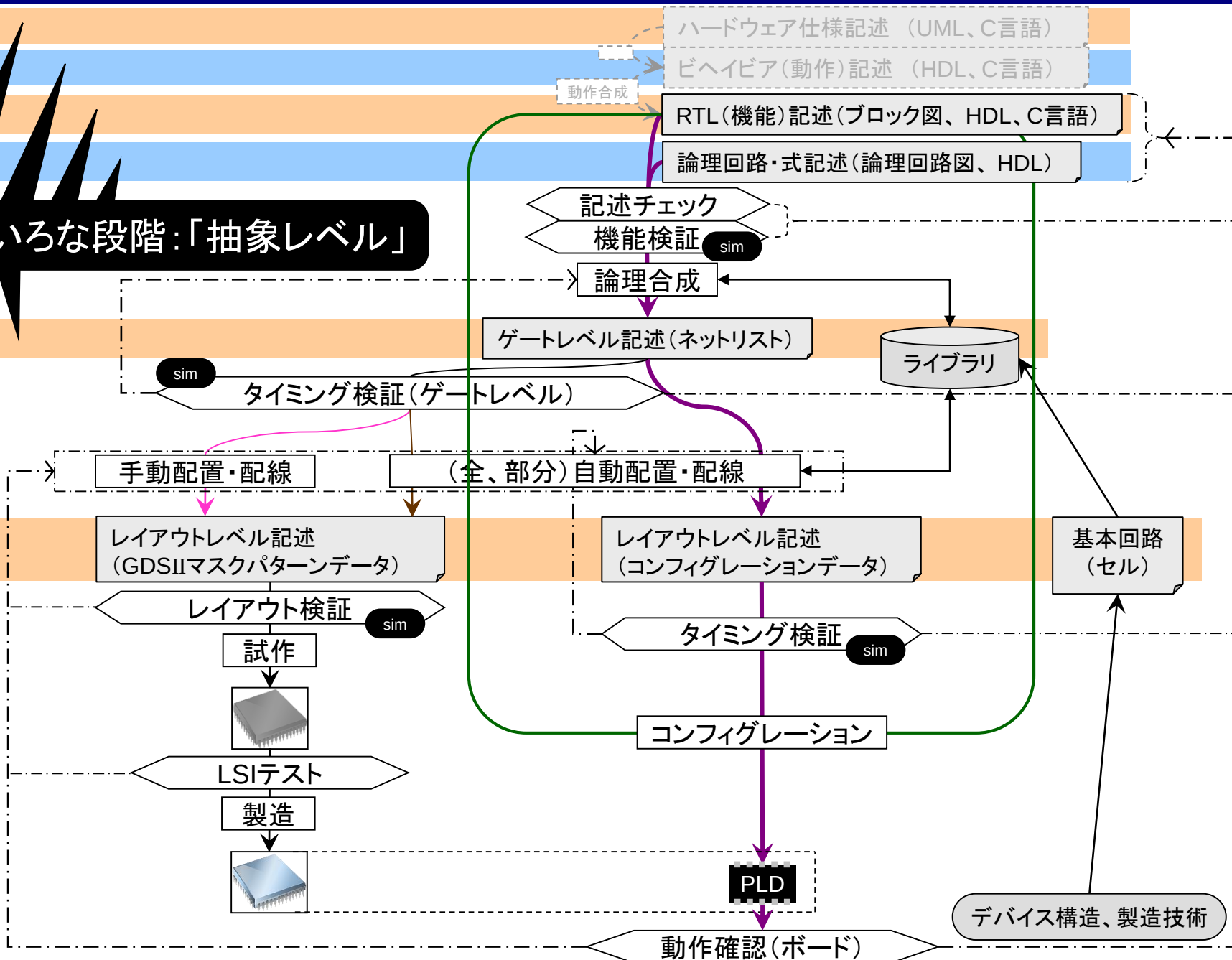


本科目の設計製作フローと、設計の記述(デザイン)

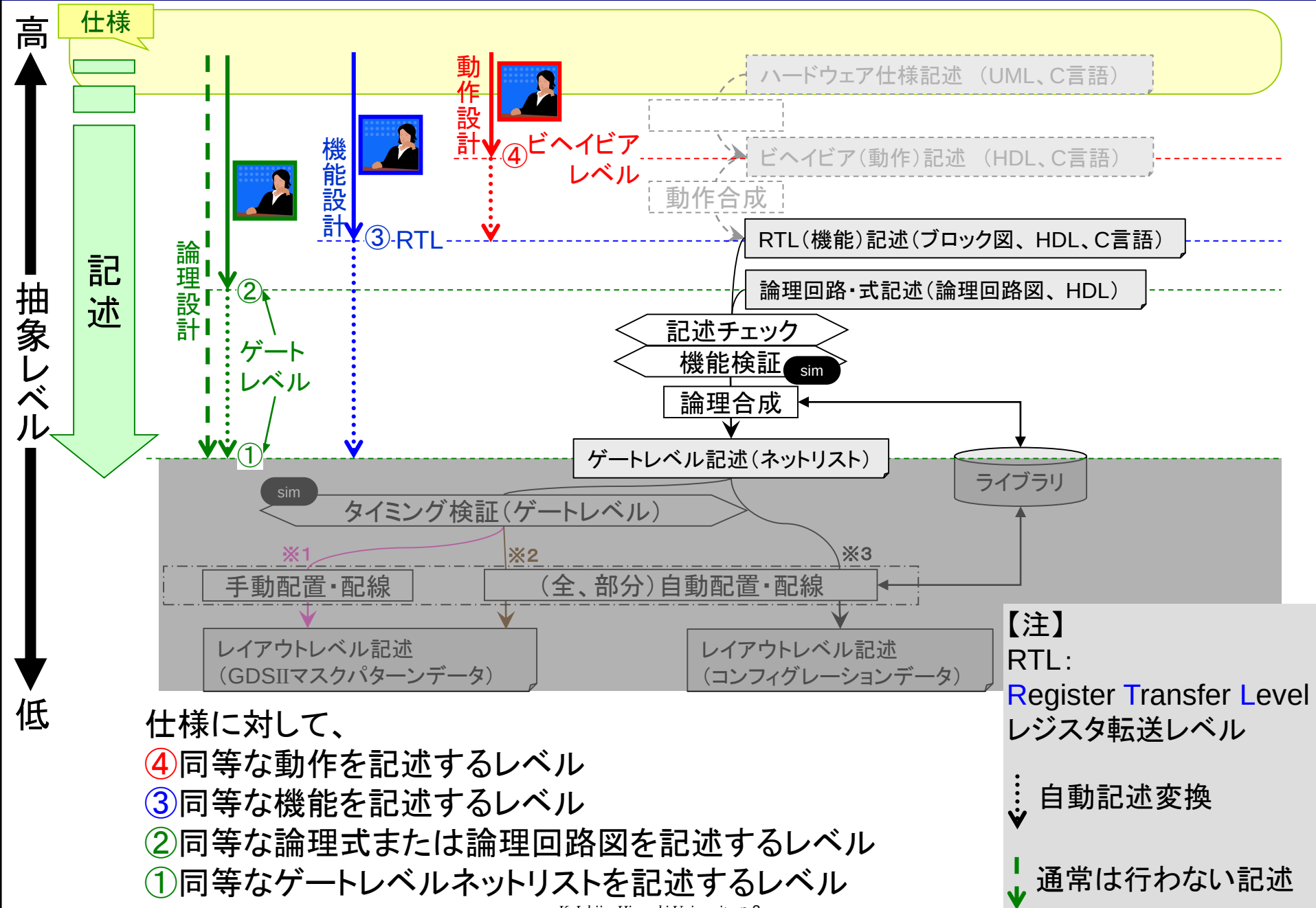


一般的なIC設計製作フローにおける、デザインのいろいろな段階

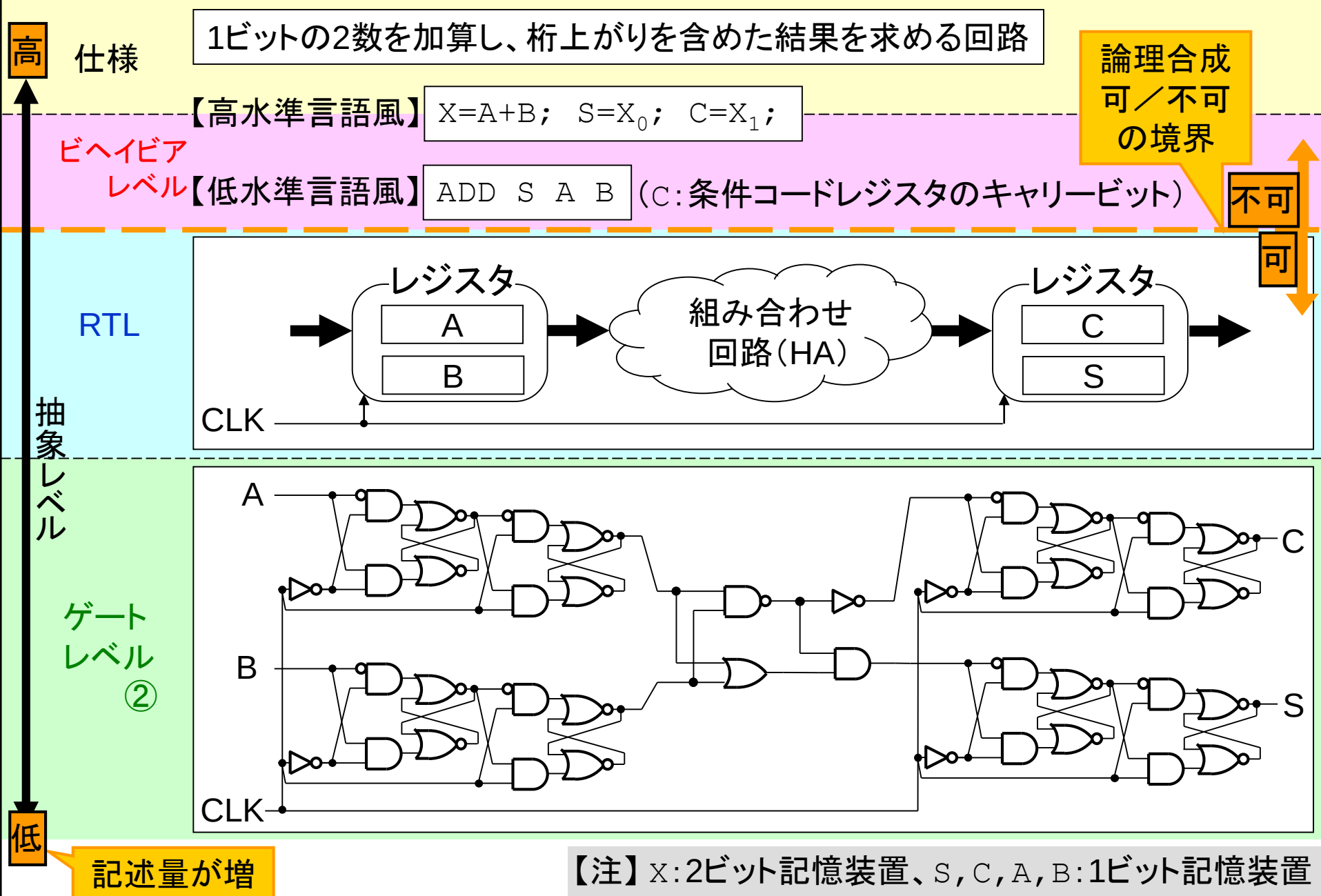
いろいろな段階:「抽象レベル」



デザインの抽象レベル



デザインの抽象レベルの例



【注】 X:2ビット記憶装置、S,C,A,B:1ビット記憶装置

RTL(レジスタ転送レベル)の特徴

高 仕様

ビヘイビア
レベル

RTLの特徴

1. 論理合成可能である。
2. 抽象レベルが高く、記述量が少なく済む。

論理合成
可／不可
の境界

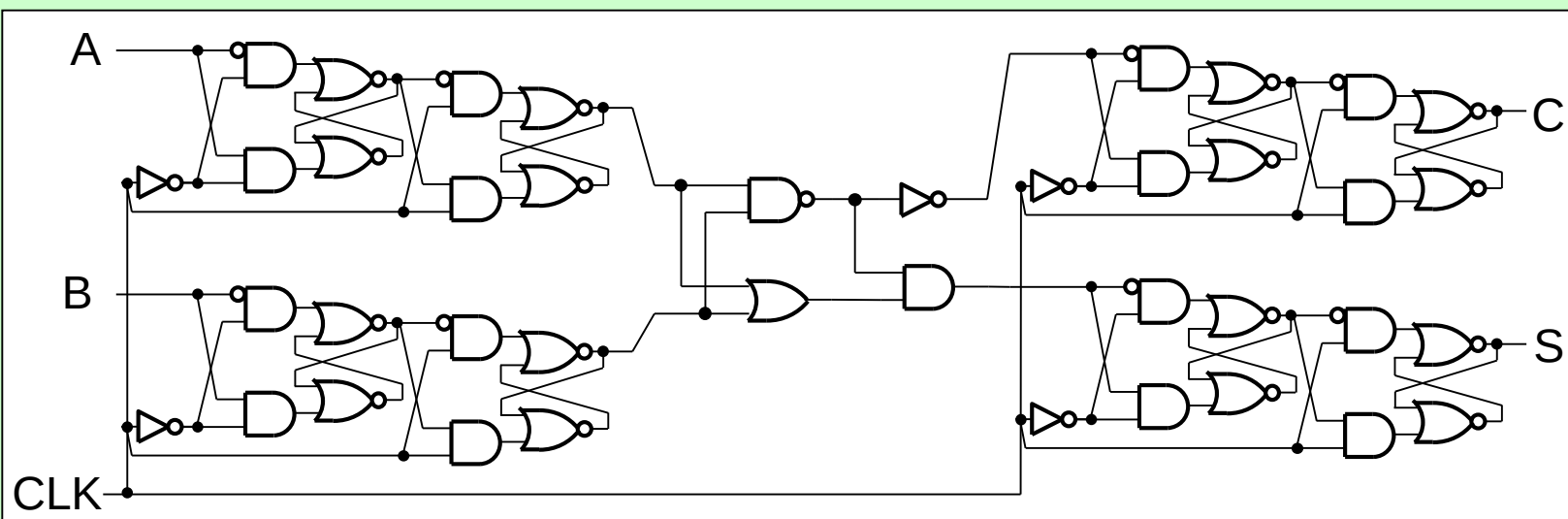
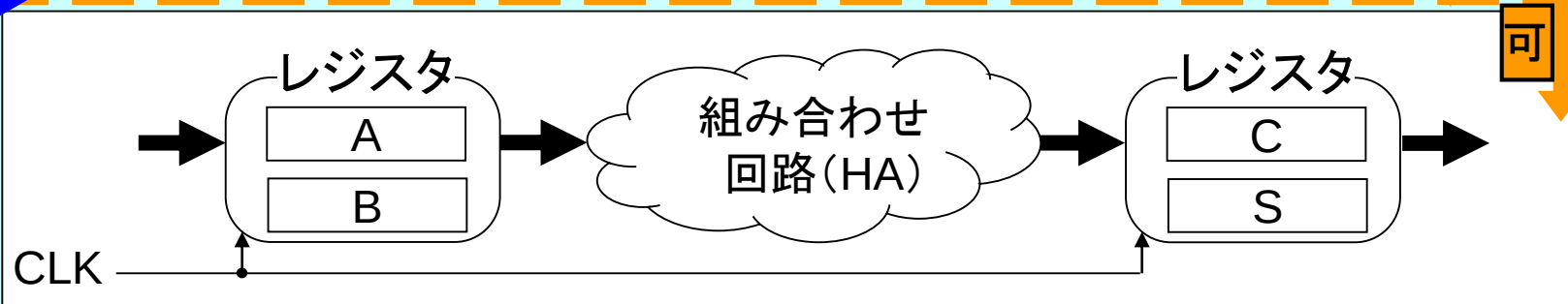
不可

可

RTL

抽象
レベル

ゲート
レベル
②

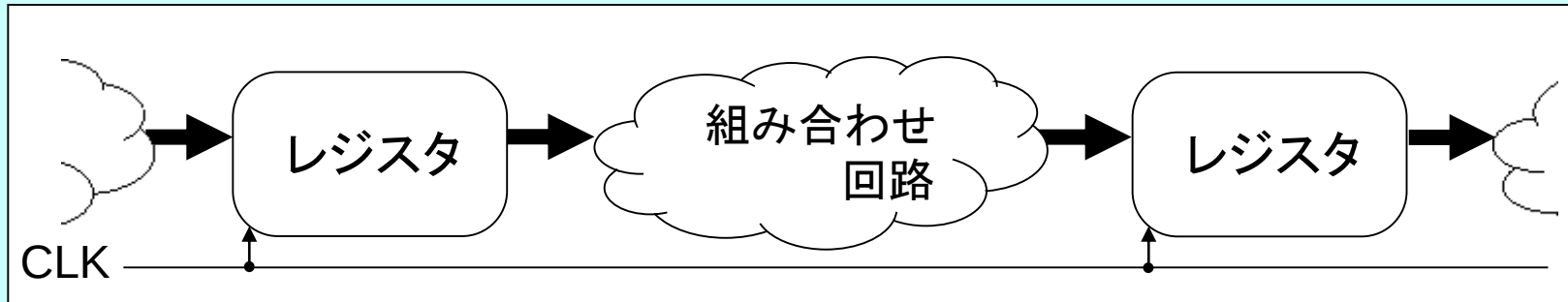


記述量が増

【注】 X:2ビット記憶装置、S,C,A,B:1ビット記憶装置

RTL(レジスタ転送レベル)だけでよい？

RTL



RTL(機能)記述(ブロック図、HDL、C言語)

論理回路・式記述(論理回路図、HDL)

ゲートレベル②

	組み合わせ回路の設計	レジスタの設計
ブロック図	組み合わせ回路機能の ブロック図用シンボル	レジスタ機能の ブロック図用シンボル
HDL	組み合わせ回路機能の記述	レジスタ機能の記述

RTLの特徴

+

ゲートレベル②が有効な場合



対象レベル RTLをメイン、ゲートレベル②をサブ、
の抽象レベルに据えてデザインする

○相応しいシンボルがない
・論理が単純
↓
論理ゲートシンボルで記述

・論理が単純
↓
論理式記述の方が
記述量減になることも

EDAツールの能力や
ライブラリの豊富さ

例 `if (a==0 || b==0) {x=0;}
else {x=1;}`

【注】

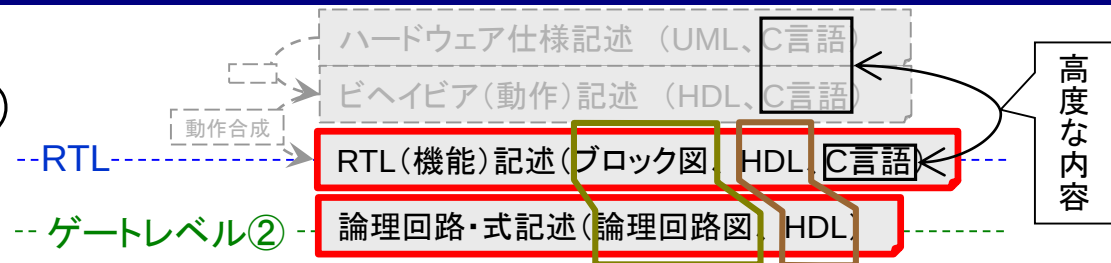
- ・a, b, x: 1ビットデータ
- ・例はC言語風(≠HDL風)の記述

対象レベルでの デザイン方法

主な2つのデザイン方法

1. 作図(ブロック図、論理回路図)
2. HDL

ファイルとして保存する
(デザインファイル)



特徴比較項目	作図	HDL
記述要素	シンボル図形と配線ライン	テキスト
記述用ツール	EDAツール専用作図エディタ (Quartus IIの場合、bdfエディタ)	テキストエディタ (一般の／EDAツール内蔵の)
ファイル形式	EDAツール専用形式 (Quartus IIの場合、bdf)	テキストファイル
機能記述の自由度	低い(シンボル依存)	高い(論理合成可ならOK)
移植・再利用性<EDAツール依存性>	低い<高い>	高い<低い>
機能記述の変更・拡張	困難	容易
機能記述の複製方法	Copy&Pasteのみ	Copy&Pasteとfor-generate
ブロック図としての視認性	高い	低い
回路規模増大に伴う配線の絡まり	生じやすい	生じない

【注】シンボル: 作図の素材となる図形
bdf: block design file

モジュール化デザインの容易性

デザインファイルとモジュール化デザイン(1/2)

モジュール化デザイン

目的回路を、複数のモジュールに分割して捉え、
各モジュールのデザインファイルを予め用意し、
それらを組み合わせて、目的回路全体のデザインファイルを作成すること。

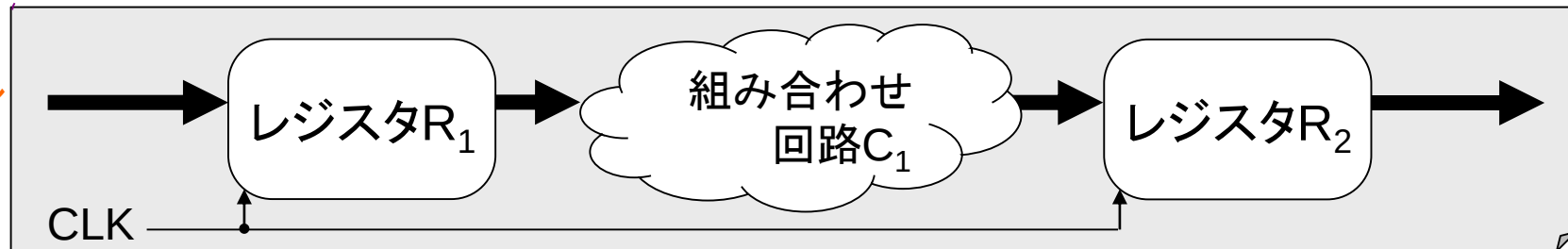
▼モジュール化デザインを行わない場合



(最上位モジュール: 目的回路全体を表すモジュール)

複数モジュールに分割せずデザイン

最上位
モジュール
デザイン
ファイル

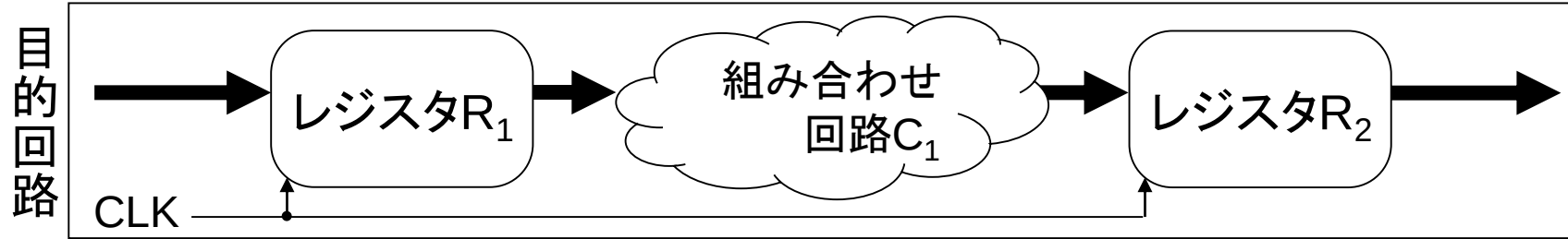


下位
モジュール
デザイン
ファイル

なし

デザインファイルとモジュール化デザイン(2/2)

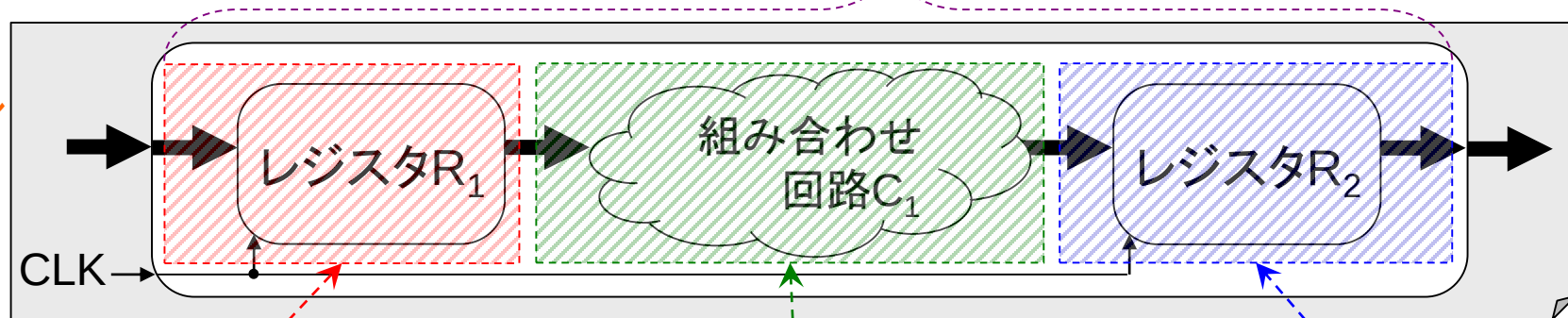
▼モジュール化デザインを行った場合



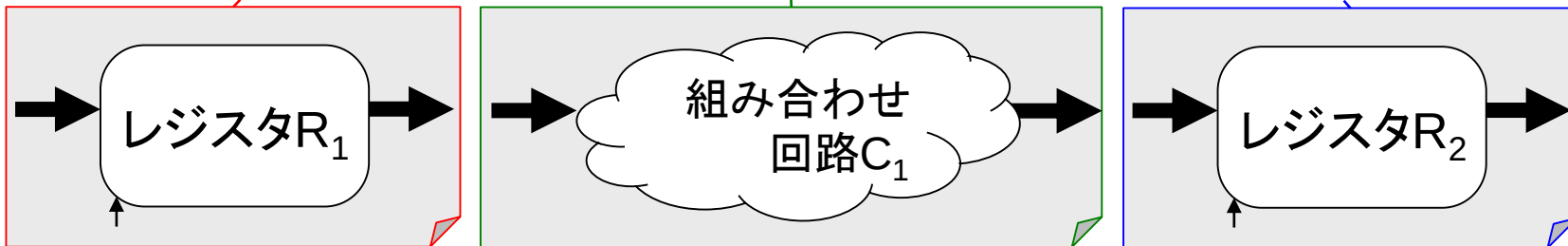
(最上位モジュール: 目的回路全体を表すモジュール)

複数モジュールに分割してデザイン

最上位
モジュール
デザイン
ファイル



下位
モジュール
デザイン
ファイル



本科目の設計製作フローと、本科目の目的

ALTERA社製PLD用
EDA/CADツール

Quartus II 9.1
Web Edition



Windows



設計用PC

接続ケーブル



ALTERA社製PLD

- EPM7128SLC84-15
- EP2C70F896

↓: 手掛けるフロー

EDA/CADツール

ハードウェア仕様記述 (UML、C言語)

ビヘイビア(動作)記述 (HDL、C言語)

動作合成

RTL(機能)記述(ブロック図、HDL、C言語)

論理回路・式記述(論理回路図、HDL)

設計

記述チェック

機能検証

sim

論理合成

ゲートレベル記述(ネットリスト)

配置・配線

レイアウトレベル
(コンフィグレーション)

タイミング検証

コンフィグレーション

VHDL(プログラミング)演習
『デジタル回路を設計する』

1. デザインの段階(抽象レベル)
 - ・メイン: RTL
 - ・サブ: ゲートレベル②
2. デザインの方法
HDL → VHDL
3. モジュール化デザイン
容易

プログラマブルデバイス演習
『設計したデジタル回路を動かす』

PLD

- PLD用開発ボード
- ISPプログラマ & チェッカ
 - DE2-70

動作確認(ボード)