

FPGAの定量的開発プロセス管理手法

古茂田典夫*
城崎雅美**
星直之*

Quantitative Process Management Method of FPGA Development

Norio Komoda, Masami Shiroasaki, Naoyuki Hoshi

要旨

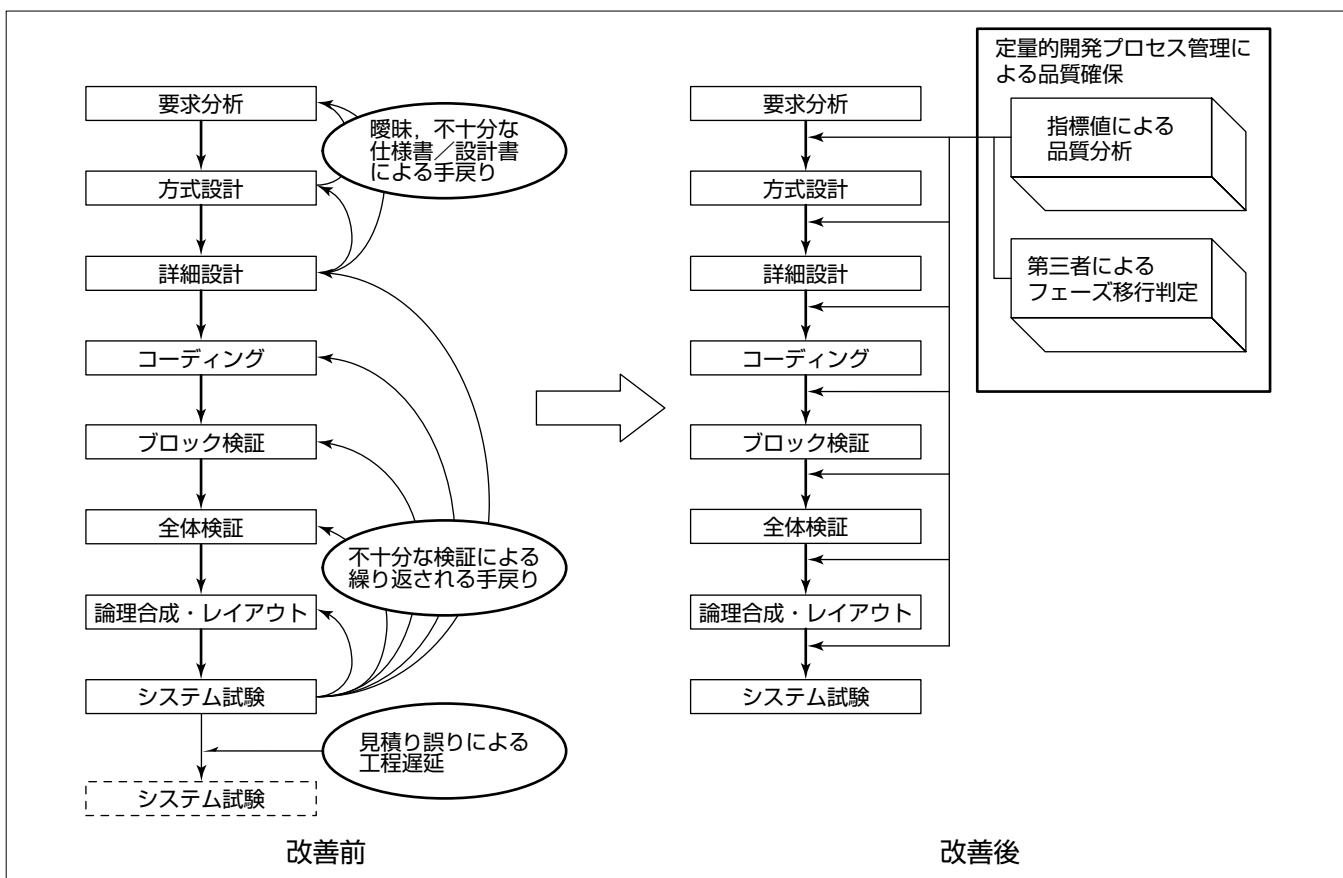
FPGA(Field Programmable Gate Array)開発における回路設計は、言語ベース設計が主流である。半導体の微細加工技術の進歩とデジタル製品機能の複雑化から、設計回路規模は指数関数的に増大し、言語記述のライン数も数十万行に膨れ上がっている。これに伴い、設計検証品質や開発期間を確保するための開発プロセス管理が不十分で、不具合流出や工程遅延の問題が顕在化しつつある。そこで、言語ベース設計の点で類似しているソフトウェア開発で体系化されている管理手法に着目した。

ソフトウェア開発では定量的開発プロセス管理(以下“定量的開発管理”という。)手法によって品質や工程の測定が行われており⁽¹⁾⁽²⁾、その手法を通信機器用FPGA開発へ適用した。その結果、進捗の定量化による工程遅延発生時

の早期対策が可能となり、システム試験への流出不具合数は品種あたり半減以下となった。また、実際の工数に対する設計初期段階での見積り誤差も半減した。

こうした成果によって、開発全体の品質を示すフロントローディング率^(註1)は75%から88%に向上した。このことは、不具合の早期検出率が高まり、手戻りが削減されたことによって、早期の品質改善につながったことを意味する。なお、今回のFPGAの定量的開発管理手法はASIC(Application Specific Integrated Circuits)開発でも同様の効果を得ている。

(注1) フロントローディング率: 回路設計完了までの検出不具合数÷総不具合数。この数値が高いほど検証工程以降の流出不具合数が低く、設計品質が高いことを示す。



定量的開発プロセス管理による品質確保

従来は、品質が不十分なまま次の開発プロセスに移行する場合があったため、品質確保できず手戻りが発生していた。定量的開発プロセス管理を導入することで、指標値から品質を確認し、不十分な場合は現行の開発プロセスで改善してから次の開発プロセスに移行することで、品質確保と工程遵守が可能となった。

1. ま え が き

1990年代半ばから、ASICやFPGA開発で、言語ベースの回路設計手法が急速に普及した。この設計手法は、HDL(Hardware Description Language)と呼ばれる言語で記述されるため、設計の容易性、半導体ベンダーに依存しないメリットがあり、設計生産性の向上に大きく寄与した。また、半導体の微細加工技術の進歩とデジタル製品機能の複雑化から、図1に示すように、設計回路規模は指数関数的に増大し、現在では、設計回路規模で数千万ゲート、HDLのライン数で数十万行となる開発も珍しくない。このような設計回路規模の増大に伴い、各開発プロセスで行うべき作業や品質作りこみの実施漏れを見逃したり、分業化された作業分担の割当てや作業間の連携ができないまま作業を進めてしまうことがある。その結果、工程の計画値と実績値との乖離(かいり)や特定回路ブロックの設計品質悪化を見逃し、工程遅延を招くという問題が発生している。同様の問題はソフトウェア開発では既に顕著に現れており、その対策法として定量的開発管理がある。

本稿では、通信機器用FPGA開発に対する定量的開発管理の適用と、その効果について述べる。

2. 定量的開発管理の狙いとV字モデル

大規模化したソフトウェア開発では、検証工程で全ての項目を検証し妥当性を確認することは現実的に不可能である。この前提に立ち、検証工程以前の早い工程で品質を管理し作りこんでいくことがソフトウェアの定量的開発管理の目的である。FPGA開発でも言語ベース設計という点で類似しているため、ソフトウェア開発における対策法を応用できると考えられる。

対策を検討するにあたり、V字モデルを活用する。V字モデルとは、開発工程を設計、検証工程に分け、詳細レベルを同じ高さにしてV字型に記載した開発モデルである。V字モデルを見れば、開発工程の各フェーズで何を設計し、その成果物に対して何を検証すれば良いかがわかり、そこ

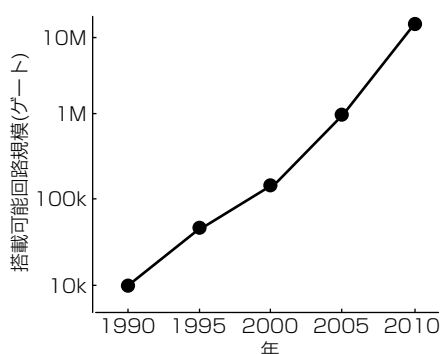


図1. 半導体プロセス進化に伴うFPGA搭載可能回路規模

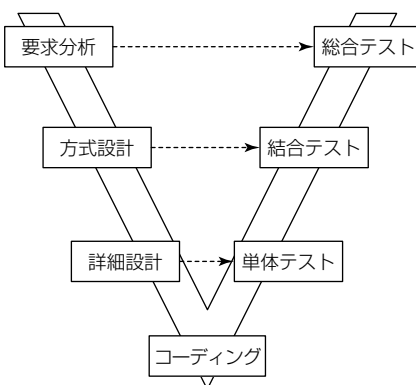


図2. ソフトウェア開発におけるV字モデル

から成果物、管理すべき指標値を抽出できる。図2は、ソフトウェア開発工程を規定したもので、V字の左側がシステムの仕様を詳細化していく工程、右側が詳細な検証からシステム検証に向かってシステムを構築していく工程である。左右の工程は関連付けられており、開発の詳細さのレベルを合わせている。このモデルと同様の考え方で、FPGA開発の差異を考慮しながら作成したV字モデルを図3に示す。設計プロセスについては、ソフトウェアと同じ開発プロセスとなった。検証プロセスについては、FPGA開発の場合、ブロック検証→全体検証の順となる。これは、ソフトウェア開発の単体テスト→結合テストに類似しているため、そのまま置き換えた。FPGA開発特有の開発プロセスである論理合成・レイアウトについては、対応するプロセスがないため、独立した開発プロセスとして追加した。完成したV字モデルから各開発プロセスの成果物を定義し、各成果物に対して表1のような指標値を定量的管理すべき値として抽出した。

V字モデルと各プロセスにおける成果物、指標値の関係をまとめたものを図4に示す。各開発プロセスでは第三者が指標値を分析し、その結果を踏まえて次の開発プロセスに移行してよいかの判断材料とした。

3. FPGA開発における定量的開発管理手法

3.1 要求分析段階での定量的開発管理手法

大規模FPGA開発では、工程計画を精度よく見積もり、遅延が発生した場合でも早期に挽回(ばんかい)する必要がある。そのためには、FPGA開発における作業を細分化し、細分化された各作業に対してリスクを評価し、作業の進捗状況を定量化する必要がある。そこで、次の施策を行った。

- (1) 品質計画書によるリスク評価と事前対策
- (2) 最小作業単位のWBS(Work Breakdown Structure) 入力による進捗定量化と工程遅れの早期対策

品質計画書とは、開発開始前に、開発対象のFPGAに関して、どのような設計手法、検証手法、指標を用いて品質確保するかを示した文書である。今回、新たに開発に対す

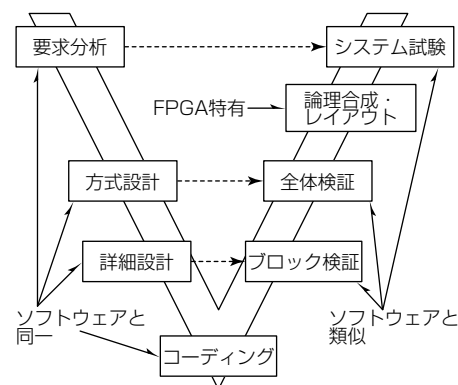


図3. FPGA開発におけるV字モデル

るリスク評価を行う項目を設けた。これによって懸念点を事前に洗い出した上で、そのリスク対策を検討し、開発中にフォローすることでリスクを最小限に抑えることが可能となった。

工程管理は進捗状況をWBSに入力し管理する。この作業単位を一週間以上にすると、進捗遅れの早期把握が困難

になる。そこで、最小作業単位を5日以内とし、週単位の進捗定量化を行った(図5)。これによって、週単位での進捗遅れの把握ができ、遅れが発生した場合の早期対策が可能となった。これらの対策によって、計画値と実績値の誤差が50%削減され、工数見積り誤りによる開発遅延防止が可能となった。

表1. 定量化すべき指標値

フェーズ	指標値	ソフトウェアとの類似性
要求分析	リスク評価	×
	工程表入力による進捗の計画値との差異	×
方式設計	ページ数	○
	レビュー工数	○
	レビュー指摘数	○
詳細設計	ページ数	○
	レビュー工数	○
	レビュー指摘数	○
	DRBFMによる変更点管理	×
コーディング	レビュー指摘数	○
	構文チェック修正数	×
ブロック検証	検証項目数	○
	HDLコード改訂数	×
	レビュー工数	○
	レビュー指摘数	○
	カバレッジ	○
全体検証	不具合検出数	○
	検証項目数	○
	HDLコード改訂数	×
	レビュー工数	○
	レビュー指摘数	○
論理合成・レイアウト	カバレッジ	○
	不具合検出数	○
	データ作成回数	×
システム試験	検証項目数	○
	不具合検出数	○

3.2 設計段階での定量的開発管理手法

V字モデルの左側に位置する設計の詳細化では、次の2種類の管理手法を用いた。

- (1) 設計レビュー工数密度と指摘密度を用いたゾーン分析による品質管理
- (2) DRBFM(Design Review Based on Failure Modes)による変更影響範囲の管理

設計レビュー工数密度と指摘密度を用いたゾーン分析による品質管理は、設計レビュー工数密度(レビュー工数/ページ)を横軸に、指摘密度(指摘件数/ページ)を縦軸にとり、実績値をグラフ上にプロットして、設計品質を定量化する。

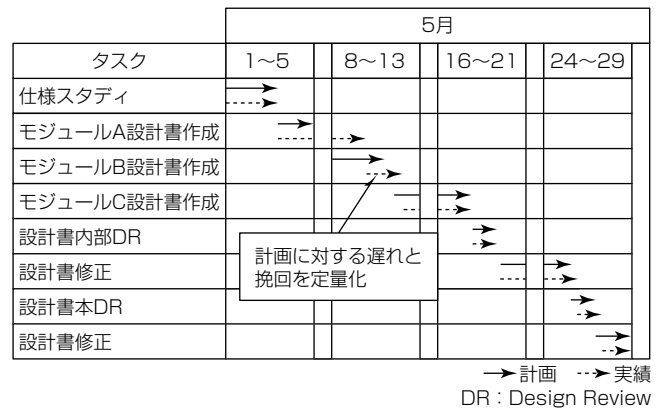


図5. WBS進捗入力例

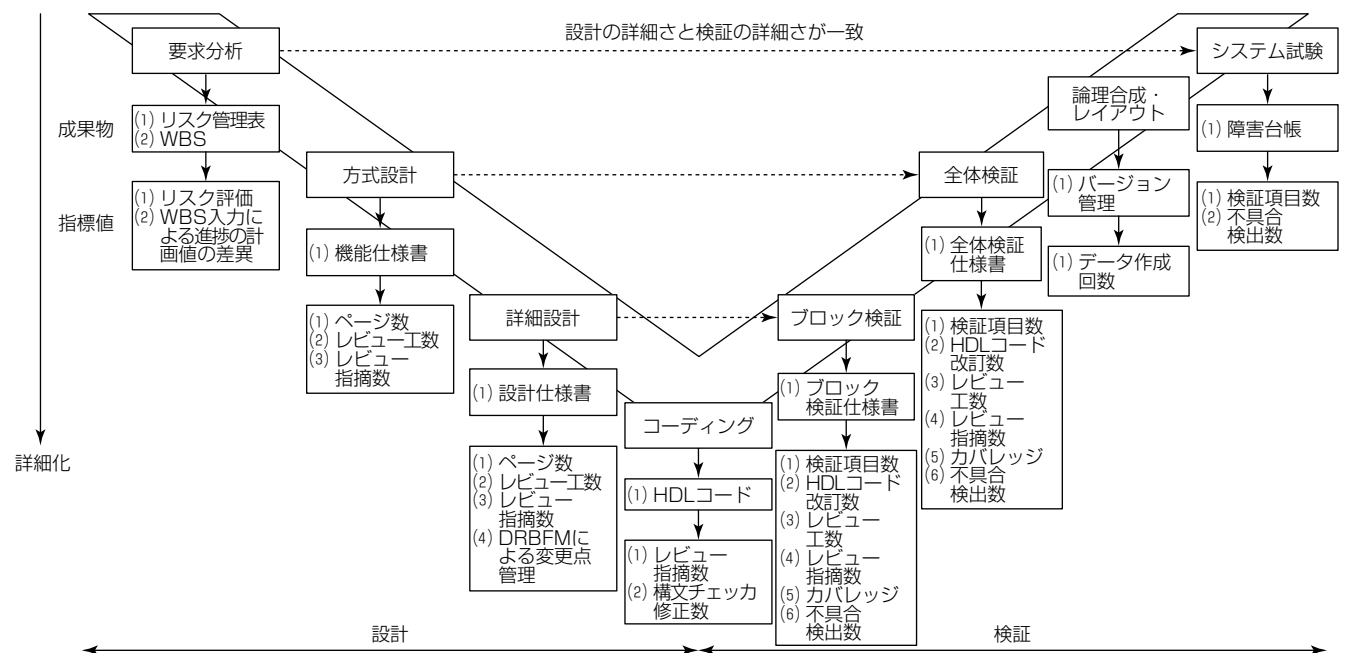


図4. FPGA開発におけるV字モデルと成果物、指標値の関係

図6に例を示す。この位置があらかじめ設定した指標値内であれば適正な設計レビューが行われ、不具合除去も適正と判断する。指標値の範囲外の場合は次のように判断する。設計レビュー工数に比べて指摘件数が少ないときは、設計品質が元々高かったか、設計の詳細にまで踏み込んだレビューができていないと判断しレビューが十分かを調査する。逆に設計レビュー工数に比べて指摘件数が多いときは、設計品質が低いと判断する。この場合は、設計品質の低さが今後問題となる可能性があるため、再レビューを行う。これらの対策によって、早期の品質向上が可能となった。

DRBFMは、流用設計変更時や設計途中の仕様変更時に適用する。DRBFMとは、既存の設計を変更する際、変更による心配点を抽出し、それが持つ影響範囲を検討し、変更設計する上での対策や第三者の視点でのレビューを実施し、未然に不具合を防止する手法である。一覧表化された変更点に対して一つ一つ管理・対策することで、設計者自身が気づけなかった不具合を検出することができた。

これらの施策によって、設計段階での早期不具合検出が可能となり、フロントローディング率が向上した。

3.3 検証段階での定量的開発管理手法

V字モデルの右側に位置する検証では、次の2種類の管理手法を用いる。

- (1) 不具合カウント基準の統一による不具合混入段階の見える化と分析、対策の実施
- (2) 検証レビュー工数と指摘件数を用いたゾーン分析による品質管理

設計データは、HDLバージョン管理ツールに更新データと更新履歴を登録することで管理する。不具合カウントにはHDLバージョン管理ツールが出力する更新履歴ログを用いるが、更新履歴の記載レベルが設計者ごとに異なり、不具合と指標値の対応が不明確であったため、十分な分析ができなかった。そこで、HDLバージョン管理ツールに登録すべきバージョン更新履歴情報を定型化し、カウント基準を統一化した。さらに、HDLバージョン管理ツールのログから不具合発生ブロックや混入原因などのデータを自動取得するツールや、プロセス別検出欠陥数を自動集計するツールを作成した。これらによって、検証品質の分析精度向上と効率化が可能となり、真に必要な対策を実施することができた。

検証レビュー工数密度と指摘密度を用いたゾーン分析による品質管理は、設計段階のゾーン分析と同様に行い、必要に応じて追加レビューなどの対策を実施することで、検証漏れを防止することができた。

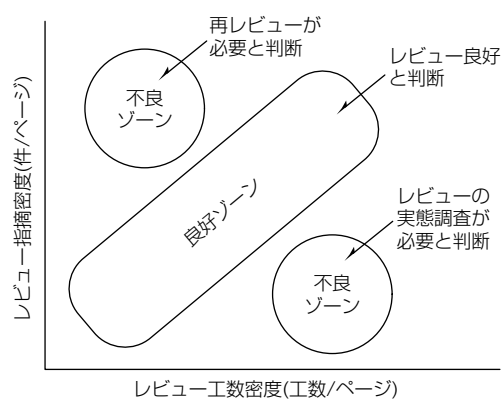


図6. 設計レビューにおけるゾーン分析

3.4 定量的開発管理による効果

この手法を通信機器用FPGA開発に用いることで、システム試験への流出不具合は半減以下となった。設計段階の定量的管理によって、仕様の誤解やインタフェース仕様不備による設計段階での不具合が減少し、検証段階の定量的管理によって検証漏れによる検証段階での不具合が減少した。特に、設計段階での不具合が減少しており、フロントローディング率は75%から88%に向上して、早期の品質改善につながった。

4. む す び

FPGA開発に対してソフトウェアの定量的開発管理手法を適用することは、工数見積り誤りの削減、システム試験への流出不具合削減、開発工程遅延防止に効果があることを述べた。流出不具合削減では、フロントローディング率が向上し不具合の早期検出率が高まり、手戻り削減と品質改善につながった。なお、今回のFPGAの定量的開発プロセス管理手法はASIC開発でも同様の効果を得ている。

今後は、総合的な検証終了判断を行うために、検証消化計画と累積欠陥数をグラフ化したX管理図などの手法を新たに適用して最終設計品質を向上させるとともに、FPGA開発工数に対する定量的開発管理工数の占める割合が高い中規模以下の開発を行う際の設計者の負担を抑える工夫を行っていく。

参考文献

- (1) (独)情報処理推進機構 ソフトウェア・エンジニアリング・センター編：改訂版 組込みソフトウェア向け開発プロセスガイド，翔泳社（2007）
- (2) (独)情報処理推進機構 ソフトウェア・エンジニアリング・センター編：定量的品質予測のススケーターITシステム開発における品質予測の実践的アプローチ，オーム社（2008）