

山口啓太* 石田仁志**
山元浩幸*
楠原崇史*

大規模LSIの機能可視化手法

Visualization Technique in LSI Specification

Keita Yamaguchi, Hiroyuki Yamamoto, Takashi Kusuhara, Hitoshi Ishida

要旨

デジタル機器製品の中核機能の多くを支えるLSI(Large Scale Integration)は、半導体の微細加工技術の進歩に伴い、大規模化・高性能化が著しい。これに伴い、多くの機能がLSIに実装されることでLSIの開発難易度は高くなり、仕様や性能などの見積り誤りに起因する問題が顕在化し、後工程に不具合が流出するリスクが増加している。

そこで、仕様や見積り誤りに起因する不具合による設計手戻りや作り直しを防止する活動を行っている。具体的には、ソフトウェア開発で用いられている“仕様を図式表現で定義可能なUML^(注1)(Unified Modeling Language:統一モデリング言語)”をLSIの要求分析とアーキテクチャ検討に活用し、仕様を可視化する。

(1) LSI要求分析

UMLを用いて要求分析を実施し、要求を一覧にまとめ

る。図で仕様を明示することで、レビューの質を向上させ、要求の曖昧性をなくし、仕様に起因する不具合を防止する。

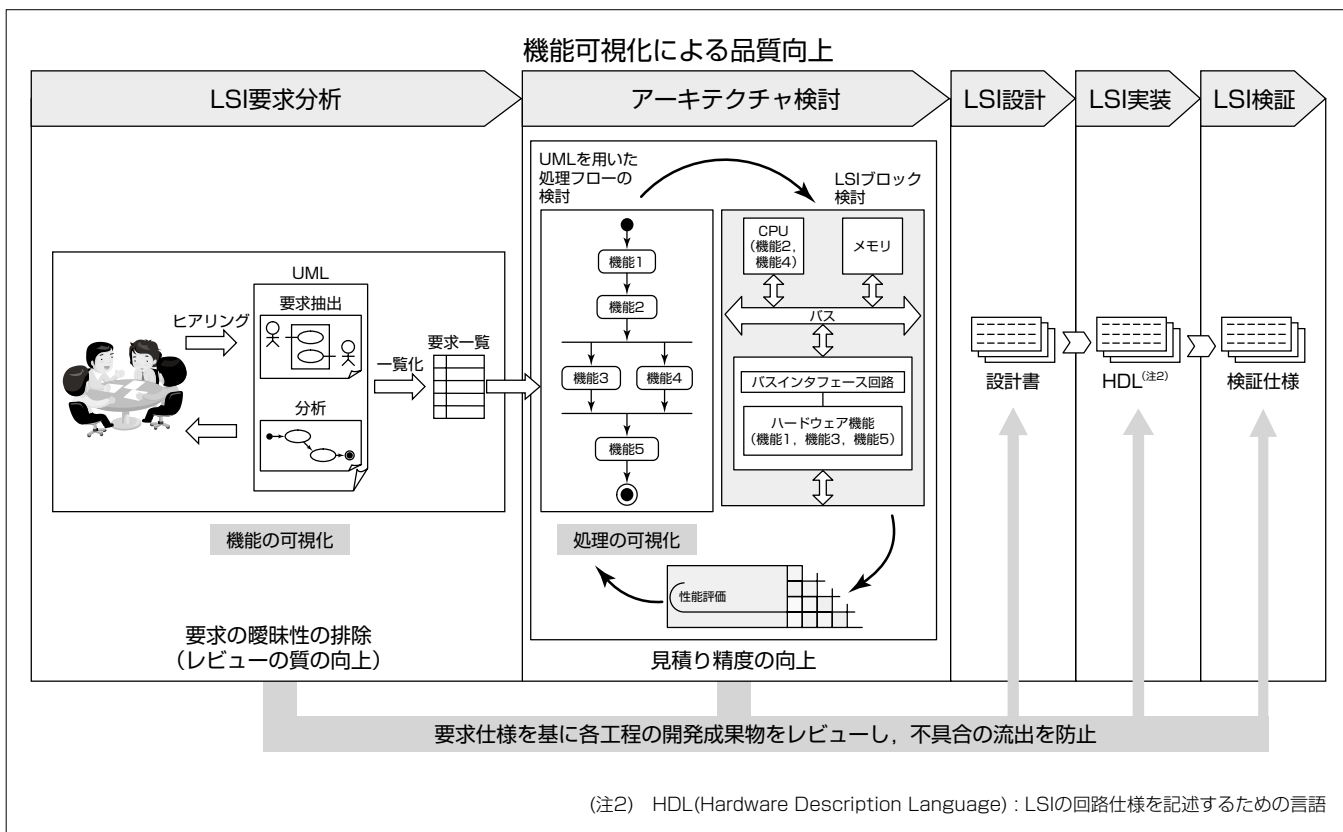
(2) アーキテクチャ検討

アーキテクチャ検討の段階で、回路規模・性能の見積り誤りによって不具合が入る可能性が高い。そこで、UMLを活用して処理フローを可視化し、見積りの精度を高める。

UMLは後工程で作成するLSI設計、実装、検証のドキュメントのレビューにも使用し、各工程のレビューの質を向上させる。

これらの取組みによって、UMLを活用した仕様検討が仕様や見積り誤りに起因する不具合を防止することが可能である。

(注1) UMLは、Object Management Group Inc.の登録商標である。



UMLを用いた大規模LSIの開発

LSI要求分析で、機能要件の抽出にUMLと要求一覧を活用する。UMLで機能と処理を見える化し、レビューの質を向上させて、仕様不具合を防止する。性能要件の分析で、LSIのアーキテクチャの検討にUMLを活用し、UMLで検討した処理フローを基にアーキテクチャを考案し、見積り精度を上げる。要求仕様を基に、開発後工程の開発成果物を品質確認し、不具合の流出を防止する。

1. ま え が き

大規模・高性能・高機能のLSIの開発難易度は高くなり、要求仕様の曖昧性による不具合や性能などの見積り誤りに起因する問題が顕在化している。そのため、後工程に不具合が流出するリスクが増加している。

三菱電機設計システム技術センターでは、その根本原因である仕様の品質を向上させるために、ソフトウェア開発で利用されている“仕様を図式表現で定義可能なUML”をLSI設計に活用し、LSI仕様策定フェーズから徹底的な品質の作りこみを行う設計フロントローディングを推進している⁽¹⁾。

本稿では、通信ネットワーク装置のようにLSIとソフトウェアが密に連携するシステムで、UMLを活用して、仕様の品質を向上させる手法について述べる。

2. LSI仕様策定フェーズにおける課題

当センターでは、定量的開発プロセスの管理⁽²⁾によって、開発後工程に設計不具合の流出を防止する活動を行っている。設計不具合流出の原因としては、図1に示すように、仕様や性能に起因する不具合が過半数を占めている。

この主な原因は、2つある。

(1) 仕様の不具合に起因するケース

LSIの機能や性能を定義する仕様は、自然言語(日本語や英語)ベースの仕様書で記載されるのが一般的である。しかし、自然言語による記述は仕様の曖昧性や、機能の相互関係を厳密に表現できないために機能間の不整合が生じる問題がある。

(2) 見積り精度に起因するケース

要求仕様を実装設計に落とし込む時の検討が不十分なために、性能などの見積りの精度が悪く、所望の性能を達成できない。

したがって、要求仕様の曖昧性をなくし、仕様見積り誤りに起因する不具合を防止することがLSI開発の課題である。

3. LSI設計へのUMLの適用

ソフトウェア開発では、ISO/IEC19501:2005規格とOMG (Object Management Group)で標準化されているUMLで仕様を図式表現することによって仕様を見える化し、レビュー性を向上させている。

LSIの開発でも、LSI仕様策定フェーズで仕様や見積り誤り起因の不具合を防止するために、図2に示すように、UMLを要求分析とアーキテクチャの検討に活用し、要求仕様の完成度と回路規模・性能の見積り精度の向上を図っている。

要求分析にUMLを用い、要件管理のために要求事項を整理した要求一覧を作成し、レビューの質を向上させることによって、要求の抽出漏れを防止する。LSI要求分析における仕様不具合への取組みについては、4章で述べる。

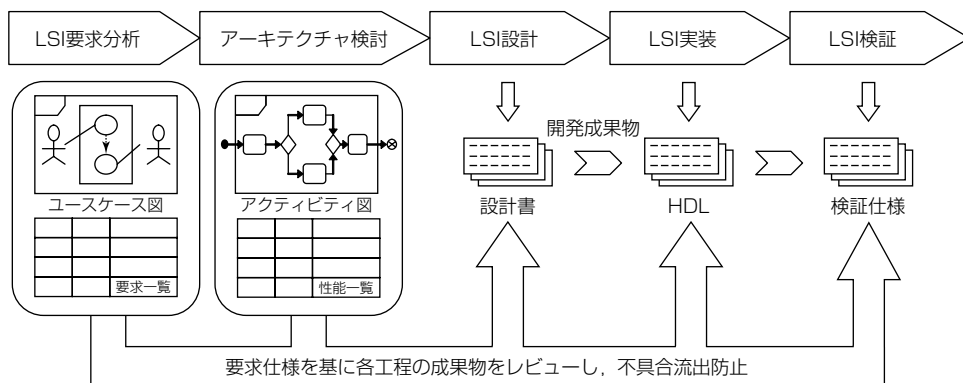


図2. UMLを活用したLSIの開発

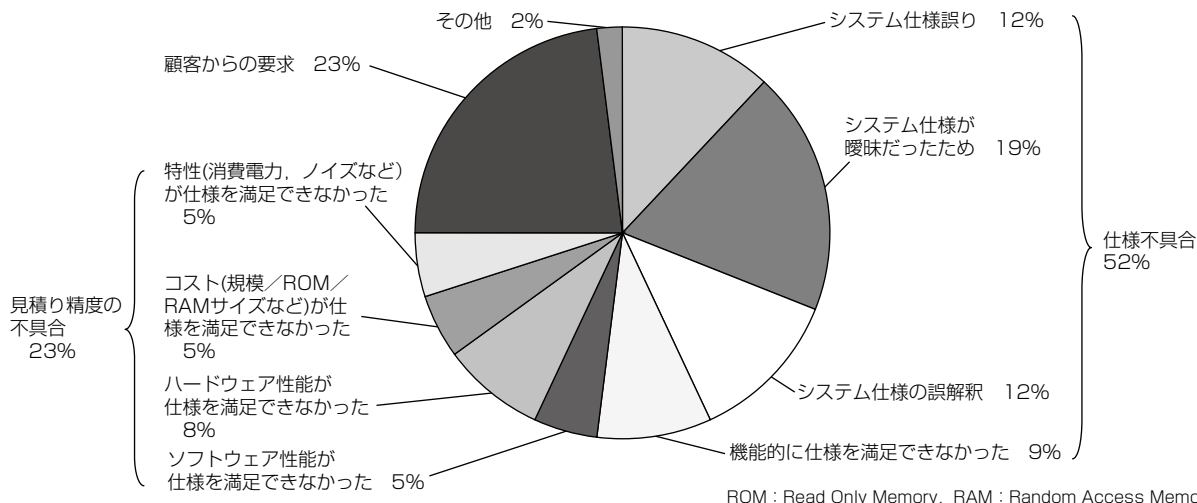


図1. システム仕様変更原因⁽³⁾

また、アーキテクチャの検討に、UML図と性能などの項目を記載した見積表を作成し、見積りに基づいたソフトウェアとハードウェアの構成を検討して、見積り誤りに起因する回路肥大化や性能不足の不具合を防止する。見積り誤りに起因する不具合に対する取組みについては、5章で述べる。

なお、開発後工程の成果物である設計書、HDL (Hardware Description Language)、検証仕様書のレビューで、要求分析の開発成果物である要求仕様(UML図、要求一覧、各種見積り)が期待通りに作成されていることを確認することによって、設計と検証の品質を向上させ、不具合の流出を防止する。

4. LSI要求分析

要求抽出の善し悪(あ)しによって、仕様に起因する不具合が発生するか否かが決まると言っても過言ではない。

要求仕様の品質を向上させるために、要求仕様に次に示す事項を盛り込むことを必須条件とする。

- (1) 要求事項を全て網羅していること
- (2) 何通りにも解釈できる書き方をしないこと
- (3) 簡潔に書かれていること
- (4) 要求事項間で矛盾がないこと
- (5) 制約条件(開発費用、開発期間)のもとで実現性のある要求であること
- (6) 要求を検証可能であること

図3に示すように、要求分析で作成する要求仕様書は自然言語の文章などで記述することが一般的である。そのため、要求を何通りにも解釈することができ、要求間の矛盾や過不足に気づきにくく、曖昧性の残った要求仕様となる可能性がある。

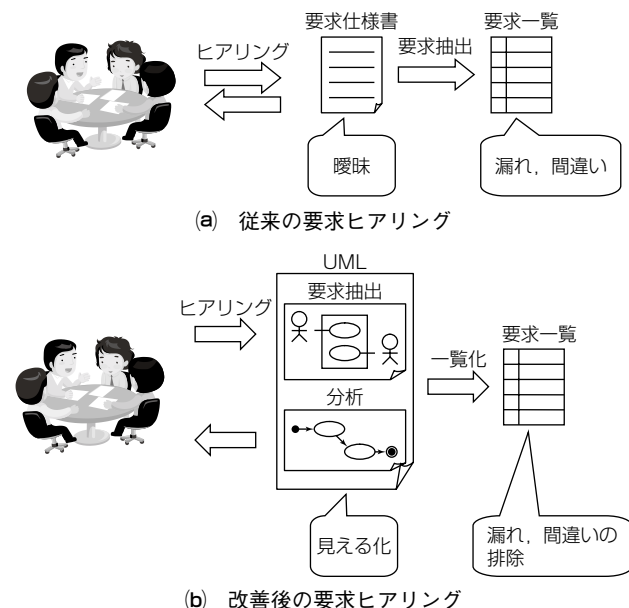


図3. 要求ヒアリングと要求分析

それに対し、UMLで規定される表記法で要求を記述することで曖昧性を排除できる。また、図的表現で要求の相互関係を分析・把握することが可能になり、機能要件や性能要件を過不足なく抽出できる。

UMLは、ソフトウェア設計の現場に浸透してきており、LSI設計者とソフトウェア設計者とのコミュニケーションが円滑にできるというメリットもある。このため、LSIの要求仕様をソフトウェア設計者でレビューすることが容易でLSI設計者だけで気づかない問題を発見することができる。

UMLで検討した要求は一覧表にまとめ、仕様設計から検証にいたる各工程の成果物が要求に合致しているか確認するために活用する。

5. アーキテクチャ検討

5.1 回路規模と性能

要求分析に基づき、回路規模と性能に対する要求を満たすように、LSIのアーキテクチャ検討を行う。このアーキテクチャ検討の段階では、回路規模や性能の見積り誤りが混入する可能性が高い。そこで、UMLを活用して処理フローを可視化し、見積りの精度を上げる。

5.2 アーキテクチャ検討例

UMLを活用したアーキテクチャ検討例として、機能の共有化による回路規模削減の検討例と、従来LSIのハードウェアだけで制御していた機能をソフトウェアによって代替する検討例を示す。

(1) 機能の共有化による回路規模の削減

要求分析に基づき、要求を実現する機能を抽出し、UMLを活用して、処理フローを分析する。図4は、共通の機能を共有する複数の処理フローを1つの処理フローで表記した例である。複数の処理フローで共有する機能から、処理フローごとの個別機能に分岐するポイントを、UML図を

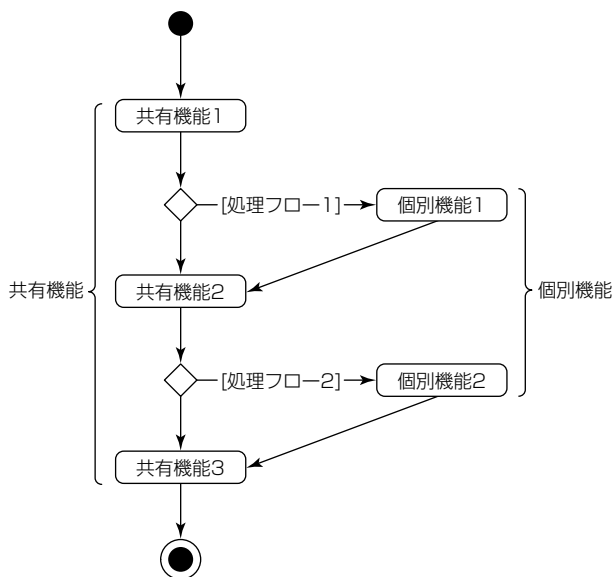


図4. 機能の共有化

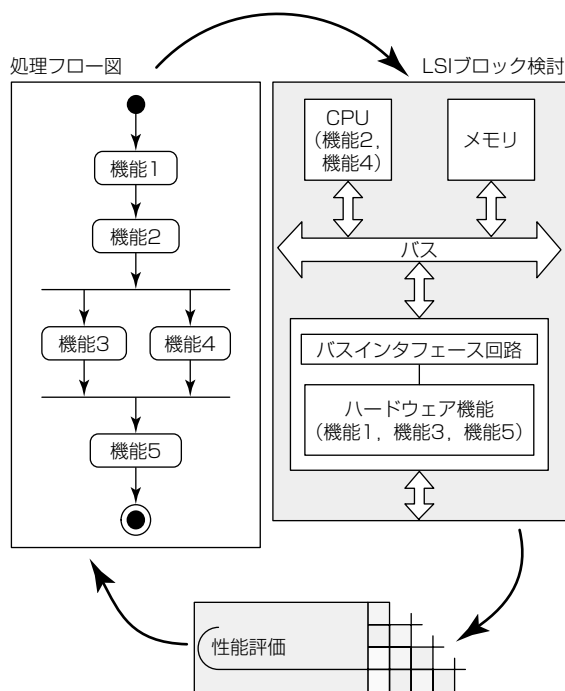


図5. ハードウェア機能のソフトウェアへの代替

利用して分析することで処理フローごとの個別機能を削減し、回路規模を抑制する。

(2) ハードウェア機能のソフトウェアへの代替

CPUの高性能化に伴い、従来ハードウェアで実現していた機能をソフトウェアに置き換えることが可能になってきた。

ハードウェア機能をソフトウェアへ代替する例を図5に示す。この例では、ソフトウェアを処理するCPU、演算処理データや通信データを保持するメモリ、CPUとハードウェアのインタフェースを接続するバス、及びCPUインタフェース回路を持つハードウェアで構成されるアーキテクチャで検討する。

UMLを活用して分析した処理フローの機能に対し、ソフトウェアとハードウェアで実現する機能の複数の組合せを検討し、各組合せの処理性能の見積りを行う。機能をソフトウェアに振り分けることによって、ハードウェアの回路規模は小さくなるため、回路規模の削減目標が決まっている場合には、回路規模も評価の項目として加える。性能低下の一番の要因は、バスインタフェースとメモリインタフェースの遅延であり、バス幅、バスプロトコル、及びメモリアクセス方法による性能の違いを評価項目として挙げ、

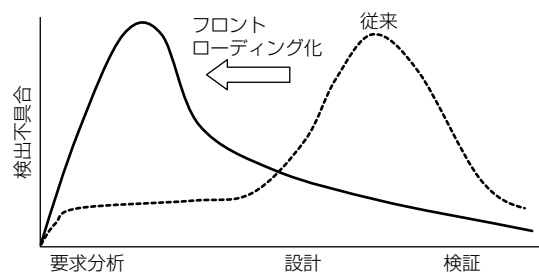


図6. 不具合検出傾向の変化

複数のアーキテクチャを比較検討する。

この性能評価の結果から、性能、回路規模を満たさない場合は、処理フローの再検討やソフトウェアへ代替する機能の再検討を行う。ソフトウェアとハードウェアとを同じ環境でシミュレーションし、性能評価の精度を上げることに加えて、アーキテクチャの検討時点で様々なアーキテクチャで性能を評価することによって、性能品質をより向上させることができる。

6. む す び

ソフトウェア開発で利用されているUMLをLSI開発へ適用することで、開発の初期段階で仕様に関する不具合の検出が可能になる。これによって、図6に示すように上流設計での仕様品質や性能品質が向上し、設計下流工程への不具合流出を防止できる。そして、設計手戻り削減と作り直しの抑止につながる。今後は性能見積りの容易化や機能分割方法の指針を整理することが必要である。

参考文献

- (1) 竹垣盛一, ほか: フロントローディング型開発設計への取り組み, 三菱電機技報, 80, No.10, 636~638 (2006)
- (2) 古茂田典夫, ほか: FPGAの定量的開発プロセス管理手法, 三菱電機技報, 87, No. 4, 232~235 (2013)
- (3) JEITA EDA技術専門委員会 SLD研究会: ニーズ/シーズ調査に基づくシステム設計手法の提案とシステム技術言語の適用化検討, 第4回 システムLSI琵琶湖ワークショップ発表資料 (2000)
http://www.jeita-edatc.com/old/project/sld/public-j2000/bws_slides_final.pdf