

情報通信技術を支える高集積化システム LSI 設計技術者育成のための演習教材開発

情報電気電子工学科 久我 守弘

1. はじめに

集積度の向上により、システム全体をひとつのチップに実装するSoC (System on a Chip) が実現可能になるとともに、設計するシステム規模が増大しており、システムLSIを設計することができる人材が求められている。また、システムLSIをターゲットとする組込みシステム開発を行える技術者の育成も急務となっている。教育機関においてもシステムLSIを利用する組込みシステムの設計・開発を行うことができる技術者の育成に適した教材が開発されている。

一方、設計における生産性の向上のためにUML (Unified Modeling Language) を用いる手法が検討されている。UML を組込みシステム設計に用いる利点として、複雑なシステムの可視化や抽象化を行うことができ、設計者間の共通理解を深めることが可能であるという点が挙げられる。本研究では、システム設計技術者育成のために、UMLを用いた組込みシステム設計を行う教材の開発を行った。

2. UMLによる組込みシステム設計

2. 1 設計フロー

UMLはオブジェクト指向分析やオブジェクト設計モデリングのために標準化されたモデリング言語である。UMLによりシステムをプラットフォームに依存しないモデルで記述することができるため、ハードウェアとソフトウェア双方に依存せずにシステムの検討を行うことが可能である。UMLによる組込みシステム設計フロー

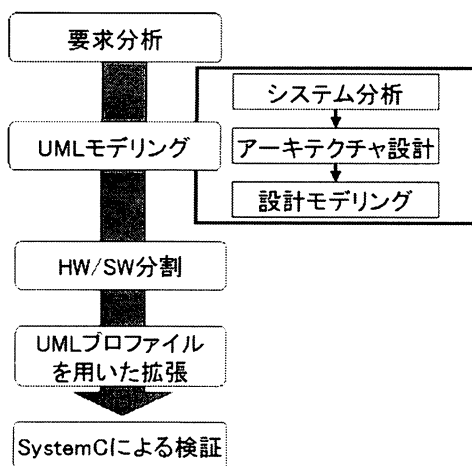


図 1: UML を用いた組込みシステム設計フロー

を図 1 に示す。本研究では、組込みシステム設計に MDA (Model-driven Architecture) の概念を適用している。MDAは要求分析モデル (CIM) の作成を行い、作成したCIMを用いプラットフォーム非依存モデル (PIM) を作成する。さらにPIMをプラットフォーム依存モデル (PSM) に変換し、PSMからソースコードの雛型であるスケルトンコードを生成する。図1でのCIMの作成は要求分析の工程で行い、PIMの作成はUMLモデリングの工程で行う。さらに、PSMへの変換はUMLプロファイルによる拡張の工程で行う。本設計フローは、要求分析、UMLモデリング、ハードウェア/ソフトウェア分割、UMLプロファイルによる拡張、SystemCによる検証、設計・実装の5つの工程に分類することができる。

2. 2 要求分析

要求分析の工程では、システムに要求されていることを調査・分析し特定する。要求分析の工程ではシステムの機能を分析する機能要求分析と性能や信頼性などを分析する非機能要求分析の2つの工程があり、一般的に機能要求はUMLを用い定義し、非機能要求はドキュメントとして定義している[1]。本設計フローでは機能要求を詳しく分析するために、ユースケース図を用いて抽出したユースケースごとにイベントフローを作成し詳細を検討する。さらに、非機能要求については組込みシステム設計向けに定義したガイドラインを用い分析を行う。

2. 3 UMLモデリング

UMLモデリングにはシステム分析、アーキテクチャ設計、設計モデリングの3つの工程が存在する。システム分析では、まず要求分析モデルを用いシステムの機能と構成を表す概念モデルを作成する。次に、システムがどのような機能要求で構成されているかを検討し、機能ごとに振る舞いを定義することでシステム全体の概要を決定する。アーキテクチャ設計では、要求分析とシステム分析で作成した仕様を満たすようにシステムの構成要素と構成要素間の関係と振る舞いを定義する。設計モデリングの工程では、作成した分析モデルを詳細化して実装を考慮にいれた静的構造モデルの作成を行う。これにより、システムの依存関係を明確にして操作や属性を定義し、PSMを意識したモデルの作成を行う。

2. 4 ハードウェア/ソフトウェア分割

UMLモデリングで作成した設計モデルを検討してハードウェアとソフトウェアの分割を行う。分割したハードウェア部はSystemCを用い実現し、ソフトウェア部はUMLのオブジェクト性を表現することができるC++で実現する。本稿では、分割したハードウェアに対しての検討方法を中心に議論を展開する。

2. 5 UMLプロファイルによる拡張

UMLプロファイルは、特定の用途向けのステレオタイプをまとめて定義し、UMLの記述を拡張する仕組みである。プロファイルを用いた拡張を行うことで、UMLの記述を厳密に定義することができる。MDAのプロセスであるPIMをPSMへ変換する工程もプロファイルを用いて行うことができる。SoC 設計向けUMLプロファイルは様々な団体や企業、個人によって定義されており、UMLを管理するObject Management GroupからはUML Profile for System on a Chipが提供されている[2]。さらに、SystemCの記述に注目して作成しているUML 2.0 profile for SystemCなどが存在する[3]。

本稿では、上記2つのUMLプロファイルをもとにシステムの入出力関係を明確に定義し、システムの構造を把握することで、スケルトンコードの生成を行うことを目標とするプロファイルを作成する。表1にSystemCの構造を表すステレオタイプを、図2に作成したUMLプロファイル構成を示す。SystemCのモジュール構造の

表1: SystemCの構造を表すステレオタイプ

図の名称	要素	定義
<<sc_module>>	クラス	システムの構造を表現
<<sc_channel>>	インスタンス	階層構造を記述する際のインスタンス化を表現
<<sc_prim_channel>>	クラス	内部にプロセスを持つチャネルを表現
<<sc_clock>>	ポート	クロック信号を表現
<<sc_time>>	ポート	時間変数の宣言を表現
<<sc_port>>	ポート	入力や出力、入出力兼用のポートを表現
<<sc_interface>>	インタフェース	ポートに接続するインタフェースを表現
<<sc_connector>>	コネクタ	ポートとチャネル間の接続を表現
<<sc_process>>	ステートチャート	スレッドやメソッドなどのプロセスを表現

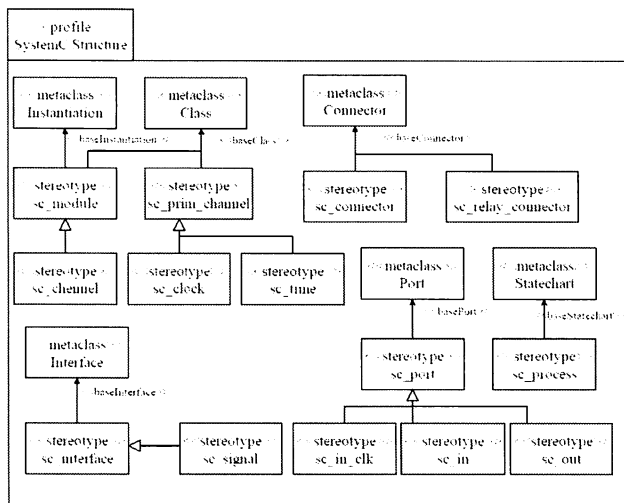


図2: UMLプロファイル

中心となる<<sc_module>>をクラス図で定義し、SystemCのコードに対応した要素をステレオタイプとして様々なダイアグラムを用いて定義した。設計モデリングで作成したモデルをもとに、プロファイルを用いた拡張を行う。プロファイルを用いた拡張を行うことにより、PIMをPSMへと変換しソースコードの生成を意識したモデルとなる。

2. 6 SystemCを用いた検証

前節で作成したプロファイルを用い、SystemCのスケルトンコードの生成する。スケルトンコードを基にしてシステムのソースコードを作成し、シミュレーションを行うことによりアーキテクチャレベルでの検証を行う。シミュレーションにはCadence社のSystemCシミュレータであるNC-SCを用いて行い、シミュレーションを行った結果はSimvisionで波形表示することができる。この工程では機能レベルでのシミュレーションを行い、作成したシステムが動作しているかを確認する。また、ソフトウェア部で実現したC++との協調検証も行うことができる。

2. 7 設計・実装

機能分割を行うためにUML によるアーキテクチャ設計モデルに立ち戻り、システム構成を明確にする。これにより、従来のシステムレベル設計よりも明確に機能に対してハードウェアとソフトウェアの割り当てを行うことができ、設計時の誤りを減少させることができる。本来、この工程ではSystemCからの論理合成を行いたいが、RT (Register Transfer) レベルのSystemC記述に記述レベルを具体化しなければ論理合成することができない。そこで、ハードウェアで実現する部分をVerilog HDL、ソフトウェアで実現する部分をC++言語で作成したIPとして提供する方法で実装を行う。システムの実装を行った後、実際に実機での動作確認を行い、要求仕様を満たしているか確認を行う。

3. UMLを用いた組み込みシステム設計教材の開発

3. 1 教材の方針

本研究ではUMLやシステムレベル設計における利点を活かした教材を開発する。教材の開発方針として、要求分析の重要性の理解、システムのハードウェアとソフトウェアの協調設計に対する理解の深化、様々な組み込みシステム開発に応用できる教材の3点を挙げて開発を行った。

要求分析の重要性の理解: 学生レベルでの設計では、要求された仕様を全て洗い出すことなく場当たり的に設計を行うことが多い。そこで、要求分析を一連のフローの一つとして行うことによって、システム設計に対する理解を深めることが可能だと考える。要求分析の重要性を理解することで、大規模なシステム設計を

に対しても要求仕様を満たすことが可能な設計者を育成することができる教材の作成を目指す。

ハードウェア/ソフトウェア協調設計に対する理解：

システムレベル設計の重要な工程の1つであるハードウェアとソフトウェアの切り分けを検討することができる教材の作成を行う。SystemCは、システムレベル記述言語としてハードウェアとソフトウェアの協調設計に用いられる言語の中で主流となっている言語の1つであり、C++言語をベースとした記述が可能である。SystemCによる協調検証を行うことで、近年重要視されているソフトウェアとハードウェア協調設計の考え方を身につけた技術者を育成することができる教材の作成を目指す。

様々な組み込みシステム開発に応用できる教材：

開発した教材は、設計事例を与えテキストに記述した組み込みシステム設計フローの手順を追いながら学習者が設計を行う。教材で与えているフローはどのような組み込みシステム設計においても適用できるような設計フローであり、学習者がこのフローを学ぶことで様々なシステムに対応可能になるような教材の作成を目指す。

3. 2 教材の概要

開発を行った教材はHTML言語で記述したWeb教材となっており、提供する情報を閲覧しながら設計例題となるシステムの設計を行えるよう考慮している。提供する内容は、組み込みシステム設計フローにおける各工程の意図を解説するためのシステムレベル設計の概要、UMLに対する理解を深めUMLの記述方法を解説するためのUMLの概要、組み込みシステムの設計フローにおける各ステージの詳細な解説、設計を行った後で行う理解度テストや設計物のドキュメント管理などの説明である。また、要求分析の重要性の理解を深めるために、イベントフローの定義を行うための表や組み込みシステム向けの非機能要求分析ガイドラインの提供を行っている。

4. 設計例題：簡易カラオケシステム

設計例題として簡易カラオケシステムを考える。本システムは表2の機能を有すると定義する。

要求分析：要求分析を行い、抽出したユースケースである「曲を流す」について定義したイベントフローを表

表2: カラオケシステムを実現する機能

機能
カラオケの音楽 (WAV) を呼び出す
カラオケの音楽 (WAV) を解析し再生を行う
マイクから歌の入力を行う
入力音声に遅延音声を付加しエコー処理をする
カラオケの音楽と音声の出力

表3: 「曲を流す」のイベントフロー

ユースケース	曲を流す
アクタ	ユーザ, RAM
説明	ユーザはシステムに対して再生の入力を行う システムは RAM から WAV データを呼び出し再生
事前条件	停止状態
事後条件	再生開始
ステップ	1. ユーザが再生開始ボタンを押す 2. システムが再生開始を受け取る 3. システムは RAM より WAV データを呼び出す 4. RAM からシステムにデータ転送 5. 転送されてきたデータを読み出し再生

3に示す。このようにユースケースごとにイベントフローを作成し、システムの機能を詳しく分析する。

UMLモデリング：システム分析を行い、アーキテクチャ設計を行った後に設計モデリングを行う。図3に設計モデリングの工程で作成したWAVデータ取得とWAV再生の静的構造モデルを示す。図3のように各処理ごとに静的構造モデルを作成する。

UMLプロファイルを用いた拡張：設計モデリングで作成したモデルをもとに、プロファイルを用いた拡張を行う。プロファイルを用いた拡張を行うことで、これまでの工程でPIMであったモデルをPSMのモデルへと変換し、ソースコードの生成を意識したモデルとなる。図4に、図3に対してUML プロファイルを用いた拡張を行ったクラス図を示す。SystemCに適した拡張を行うと、内部クロックを定義するなど実装を考慮に入れたモデルに変換することができる。またステレオタイプでポートなどを定義しており、SystemCの構造に非常に近い記述となっている。

SystemCによる検証：UML プロファイルを用いた拡張モデルから生成したスケルトンコードをもとに

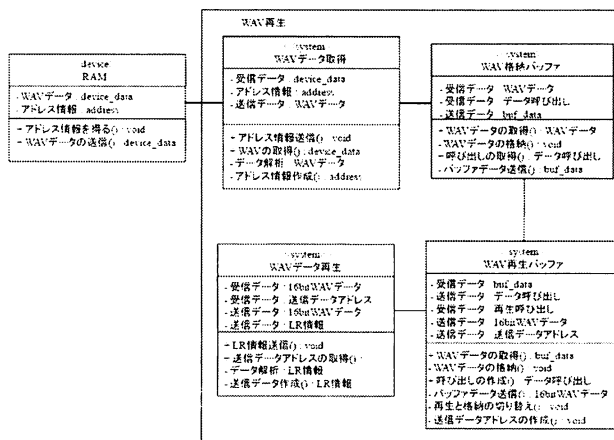


図3: WAV データに関する静的構造モデル

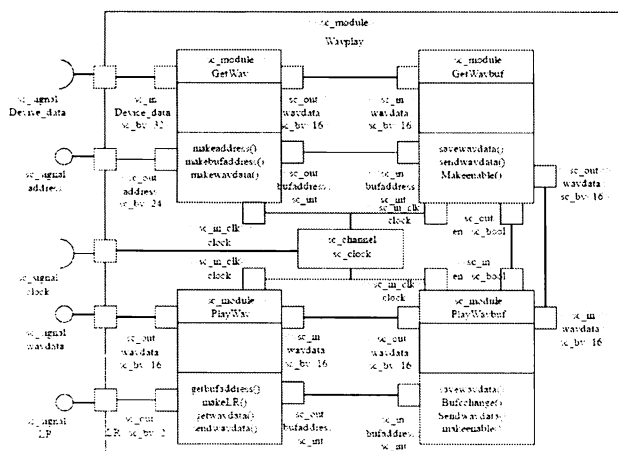


図 4: UML プロファイルを用いた拡張モデル

SystemCコードを作成し検証を行う。SystemCによるシミュレーションを行うことで作成したモデルが実行可能であるかを確認し、アーキテクチャレベルでのシステムの検討を行う。設計教材では提供する検証モジュールを用いて検証を行い、モデルで作成したシステムが動作しているか検証することができる。

5. 教材の有効性に関する考察

本稿で提案した設計は、MDAの概念に基づきUMLのモデリングを行った。設計フローの下流の工程であるほどプログラム言語でシステムを記述するかモデリング言語でシステムを記述するかのみ違いとなっており、UMLによるモデリングを行うことによる教育的な効果は上流工程になるほど高くなると考えられる。特に、要求分析の工程では、これまで与えられた仕様に対して場当たり的に設計を行ってきた学習者に対し、効果的な要求分析方法を提供していると考えられる。

今回、UMLを用いた組込みシステム設計教材を作成するに当たり、要求分析の重要性の理解、システムのハードウェアとソフトウェアの協調設計に対する理解の深化、様々な組込みシステム開発に応用できる教材の3つの方針を持って開発を行った。

要求分析の重要性の理解については、設計フローで機能要求分析と非機能要求分析は設計モデリングまで影響を及ぼし、非常に重要な工程であることを理解することができると考えられる。また、様々な組込みシステム開発に応用できる教材という点については、提案したフローは様々な組込みシステム設計に応用することができる。さらに、作成したUMLプロファイルも他の設計モデルに対して提供できることから、様々な組込みシステム設計に応用可能であると言える。しかしながら、システムのハードウェアとソフトウェアの協調設計に対する理解の深化に対しては、現状では従来のシステムレベル言語と同様の検証を行うため、検証に関しては他の

教材と変わらない。また、UMLから抽出することができるXMLを用いたSystemCの構造を抽出する研究を実現する必要がある。モデルからSystemCソースへの変換を手作業で行っており、学習者の負担となる恐れがある。今後は協調設計のための検証環境やモデル変換ツールの開発を行う必要があると考えられる。

6. まとめと今後の課題

本稿では、UMLを用いた組込みシステム設計教材の開発を行った。教材開発ではまずUMLを用いた組込みシステム設計フローの提案を行った。提案した設計フローは、MDAの概念に基づきモデル変換を行うことで、仕様から一貫性のある設計を行うことができる。また、PIMからPSMに変換するために、SystemC向けのUMLプロファイルの定義を行った。UMLプロファイルを定義することで、実装を考慮に入れたモデルに変換を行うことができる。この教材を用いることで、要求分析の重要性に対する理解や様々な組込みシステム設計開発に応用可能であるという点に関して満足のいく教材開発を行うことができた。

今後の課題として、ハードウェアとソフトウェアの協調設計に対する理解の深化について、協調設計のための検証環境やモデル変換ツールの開発を行う必要がある。検証環境や変換ツールを開発することで、組込みシステム設計に対して有効性の高い教材を提供することができると考えている。

参考文献

- [1] 佐川博樹, 図解入門よくわかる最新システム開発者のための要求定義の基本と仕組み, 秀和システム, 2005.
- [2] "UML Profile for System on a Chip (SoC), v1.0.1," <http://www.omg.org/docs/formal/06-08-01.pdf>, 2006.
- [3] E. Riccobene, P. Scandurra, A. Rosti, and S. Bocchio, "A UML 2.0 Profile for SystemC: Toward High level SoC Design," EMSOFT'05, pp.138-141, Sep. 2005.

本研究に関連する学会発表一覧

- [1] 濱崎慎一郎, 久我守弘, 末吉敏則, "UMLを用いた組込みシステム設計教材," 火の国情報シンポジウム 2009 論文集, B-3-4, Mar. 2009.
- [2] 近藤裕樹, 浦野裕貴, 濱崎慎一郎, 濱田亮, 久我守弘, 末吉敏則, "UMLに基づく組込みシステム設計手法の検討," 情報・システムソサイエティ誌 2009 年総合大会特別号, p.134, Mar. 2009.
- [3] 濱田亮, 久我守弘, 末吉敏則, "UMLを用いたシステム LSI 設計手法に関する研究," 火の国情報シンポジウム 2009 論文集, B-3-3, Mar. 2009.