

システムレベル設計支援ツール MEISSUT の開発

西岡 聡* 橋本 忠勝** 仁平 勝裕** 原田 晋至* 堤 利幸*,**

* 明治大学大学院 理工学研究科 基礎理工学専攻

** 明治大学 理工学部 情報科学科

{snishi, tsutsumi}@cs.meiji.ac.jp

キーワード: システムレベル設計 設計支援

1 はじめに

システムレベル言語 SystemC による設計を支援するツール MEISSUT(Meiji University System-level Design Support Tool)を開発している。本稿では、システムレベル設計を効率的に支援する方法を提案し、それを実現するために開発中のツールについて説明する。開発しているツールは、従来のコーディングによる設計と同様に、構造ブロックを直接用いた図設計が可能であるという特徴を持っている。コーディングによる設計と、図による設計が行えるので、直感的で見通しの良い設計を進めることができる。

2 システムレベル設計と設計支援ツール

システムレベル設計とは、ハードウェア、ソフトウェアの区別なしにシステム全体の設計仕様を満たす機能をトップダウン方式で構築していく設計手法である。システムレベル言語は C/C++言語などの抽象度が高いソフトウェア言語を、ハードウェアの概念を記述できるように拡張したものであり、ソフトウェア、ハードウェア両方の記述ができる。システムレベル言語を用いることで、設計の早い段階からシステム全体の統合シミュレーション及びテストが行える。

システムレベル言語を用いた設計は、最初は、ソースコードを記述し、シミュレーションするのみで進められていた。最近では、図によって設計を進められるツール(VisualSpec^[1]など)や、システムの性能解析ツール(ConvergenSC^[2]など)、言語ルールチェックを行うツール(AccurateC^[3]など)等、様々な設計支援ツールが開発されている。

3 システムレベル設計支援ツール MEISSUT

3.1 開発目的

システムレベル設計を進めるにあたっては、現在、コーディングによる設計と図による設計の 2 つの方法があるが、設計対象がアルゴリズムかハードウェア構造か、あるいは、設計フェーズがどの段階にあるかによって、適する設計方法が異なるため、随時適切な方法を自由に選択して設計を進められるツールが望まれる。そこで、我々は、現在最も広く使われているシステムレベル言語 SystemC^[4]を用いて、コーディングと図のどちらでも設計を進められる支援ツール MEISSUT(Meiji University System-level Design Support Tool)を開発している。

Development of a System-level Design Support Tool

* Satoshi NISHIOKA, Shinji HARADA, Toshiyuki TSUTSUMI
Graduate School of Science and Technology, Meiji University

** Tadakatsu HASHIMOTO, Katsuhiko NIHIRA, Toshiyuki TSUTSUMI
Department of Computer Science, Meiji University

3.2 MEISSUT のプログラム

MEISSUT は Windows 上で、Java 言語、Cygwin^[5]、SystemC ライブラリ^[4]を用いて開発されている。図 1 は MEISSUT のプログラム構造を簡単に示したものである。MEISSUT は、ソースコード編集による設計と、構造ブロック図編集による設計が可能である。SystemC 言語で記述されたソースコードから解析した構造ブロック情報はプログラム内部で構造ブロック図情報として保持され、構造ブロックの描画に利用される。また、構造ブロック図に対する操作は、対応したソースコードを生成する。ソースコードの編集は構造ブロック図に反映され、構造ブロック図の編集はソースコード情報に反映され、両者の編集は互いに等価な操作である。設計データは SystemC 言語ソースファイルとして保存されている。ソースファイルは、1 つのモジュールに対して 1 ヘッダファイル、1 インプリメンテーションファイルで構成され、構造ブロックとコンストラクタの記述はヘッダファイル、プロセスの処理内容はインプリメンテーションファイルに記述されている。ソースファイルはコンパイル、実行され、結果が出力される。

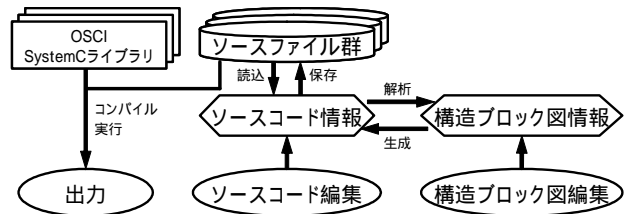


図 1 MEISSUT のプログラム構造

3.3 MEISSUT による設計

MEISSUT は、ソースコード編集による設計と構造ブロック図編集による設計を混在して進めることができ、随時望ましい設計方法を選択できる。MEISSUT のメインウィンドウを図 2 に示す。MEISSUT は階層表示ウィンドウ(図 2 の A)、構造ブロック図編集ウィンドウ(図 2 の B)、ソースコード編集ウィンドウ(図 2 の C)、出力表

表 1 構造ブロック名と説明

構造ブロック名	説明
モジュール	設計を階層化するための枠組み(図2ではA)
チャンネル	モジュール間、プロセス間の通信に利用されるデータの伝送媒体(図2ではB)
ポート	モジュールをチャンネルと接続するための端子(図2ではC)
プロセス	モジュールやチャンネルが持つ機能を記述(図2ではD)

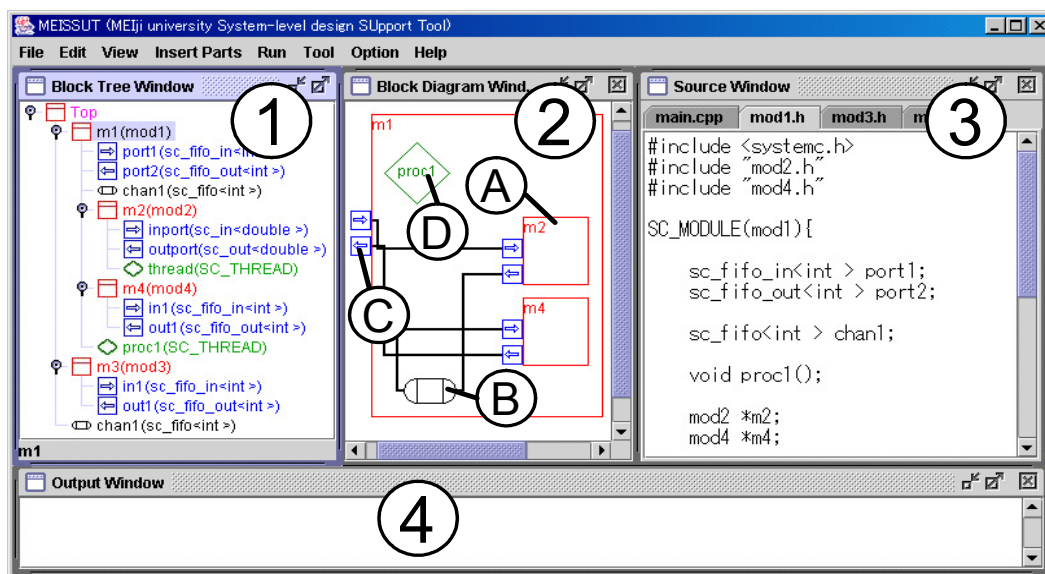


図2 MEISSUT のメインウィンドウ

示ウィンドウ(図2の)の4つのウィンドウで構成される。

SystemC 言語では、システムを設計していく上での基本的な構造ブロック^[6]として、表1の4つが挙げられる。これらの構造ブロックをソースコードで記述したり、図で編集することによってシステムレベルの設計を進めることができる。階層表示ウィンドウでは、現在設計中の構成全体を、構造ブロックのツリー形式で表示する。構造ブロック図編集ウィンドウでは、構造ブロックを追加したり削除したり、インスタンス名やそのブロックが持つ固有の情報(例：チャンネルの種類、そのチャンネルを通るデータ型など)を変更することができる。また、ポートとポート、ポートとチャンネルを接続する場合、接続したいポートとポートまたはチャンネルを線で結ぶようにドラッグ&ドロップすると接続できる。ソースコード編集ウィンドウでは、ソースコードによる設計を進めることができる。出力表示ウィンドウには、ソースコード情報をコンパイル、実行した結果が表示される。

図編集ウィンドウで操作した内容は、それに対応するソースコードに反映され、ソースコードを直接編集した結果も、構造ブロック図編集ウィンドウに反映され、互いの設計方法が補完し合い、その操作は等価である。

MEISSUT ではソースコード編集と構造ブロック図編集が併用できることから、以下の利点を得ることができる。

- (1)正しくソースコードを記述できたかを構造ブロック図として確認できる。
- (2)自動的に誤りのないソースコードを、構造ブロック図を編集することによって得られる。
- (3)構造ブロック図編集ウィンドウから、設計中のブロック構成や接続関係が一目でわかる。

3.4 設計補助機能

MEISSUT の設計支援補助機能として、キーワードや特定の単語のテキスト色を変える機能、ソースコードと構造ブロックを検索する機能、コンパイルの結果出力されたエラー部分を表示する機能がある。また、ソースコ

ード編集ウィンドウのタブ幅を変える設定などのカスタマイズ機能を備えている。

4 実験結果

実際に MEISSUT を用いて、ガロア体演算回路^[7]の設計を試みた。今回設計したガロア体演算回路は2階層6モジュールほどの規模の回路である。MEISSUT を用いた設計と比較するため、一般的なテキストエディタのみを用いた設計も行った。実験の結果、SystemC 言語の初心者だが、MEISSUT を用いてガロア体演算回路を設計できることを確認できた。また、テキストエディタのみを用いた設計と比べ、構造ブロック図編集機能を利用することで、接続関係の記述部分の効率化や記述の誤りを低減でき、より短期間に回路を完成させることができた。MEISSUT を用いた回路設計例を増やし、設計効率化効果を定量的に調査していく予定である。

5 結論

MEISSUT を使用することで、従来の設計方法よりも、正確かつ短期間で、設計することができることを実験で確認した。MEISSUT の最大の特徴は、ソースコード編集と構造ブロック図編集、どちらの設計方法にも随時変更できることである。今後の MEISSUT の課題として、SystemC マクロを用いない複雑なモジュール記述、ユーザ定義のチャンネル記述、テンプレート記述、配列記述への対応、さらに動作速度の向上が挙げられる。

参考文献

- [1] VisualSpec 株式会社インターデザイン・テクノロジー <http://www.interdesigntech.co.jp/>
- [2] ConvergenSC コーウェア株式会社 <http://www.coware.co.jp/>
- [3] AccurateC 株式会社プライムゲート <http://www.prime-gate.com/>
- [4] SystemC OSCI ホームページ <http://www.systemc.org/>
- [5] Cygwin レッドハット株式会社 <http://cygwin.com/>
- [6] SystemC の薦め <http://www.testbuilder-jp.com/inno/tu/sysc/>
- [7] 「エラー訂正や暗号処理で使われる演算回路を極めるーガロア体を回路で実現するには...」
Design Wave Magazine 2003.7 CQ 出版株式会社