2A-4

C 言語ベースのシステムレベル設計ツールの試作

－設計詳細化フローにおける形式的等価性検証機能とその適用－

西原雄次　小池輝昌　山本徹也　辻政信
宇宙航空研究開発機構 情報・計算工学センター

1. はじめに

近年，宇宙機の高度化に伴い，宇宙用電子機器は，高機能化，複雑化の傾向にある．それらを高品質，短期開発の要求を満足して開発するためには，多くの課題がある．その為に，我々は，ハードウェア(HW)/ソフトウェア(SW)協調設計により，デジタル電子機器の仕様レベル記述から，段階的な設計詳細化を繰り返しながら，HW/SW 分割や協調検証，MPU 実装コード/RTL(Register Transter Level)コード生成までを一貫してサポートする C 言語ベースのシステムレベル設計ツール(ELEGANT[1])を試作した．ここで問題となるのが，設計詳細化前後の等価性や，同一レベルのモデルの書き換え前後の等価性をいかに保証するかである．そこで我々は，形式的な手法により，並列動作するハードウェアを含んだシステムの設計詳細化前後の等価性を検証する機能を試作した．本発表では，形式的等価性検証機能の概要と，デジタル電子機器設計に適用した結果について報告する．

2. ELEGANT システム概要と設計詳細化機能

2.1 ELEGANT システム概要

ELEGANT の設計対象は，MPU1 個と FPGA 数個から構成されるデジタル回路である．

ユーザから ELEGANT への入力は，仕様モデルである．仕様モデルとは，HW/SW に依存しない，設計対象のアルゴリズム・機能を SpecC 言語[2]で記述したモデルである．SpecC 言語とは，C 言語の拡張言語で，HW の並列動作などを記述可能な言語である．具体的には，ビヘイビア構成，ビヘイビアの動作，ビヘイビア間の通信，外部システムの動作モデル(テストベンチ)などを定義する．ビヘイビアとは，モデルの基本となる機能単位のことである．

そして，その仕様モデルをもとに，段階的な設計詳細化を繰り返しながら，HW/SW 分割や協調検証，MPU 実装コード/RTL コード生成までを行う．

2.2 ELEGANT 設計詳細化フロー

図 1 に設計詳細化フローの概要を示す．仕様モデルの各部分を，設計詳細化サブシステムのライブラリに登録された設計部品に置き換えることにより，アーキテクチャ構造と HW/SW の配分を決定したアーキテクチャモデルを作成する．ライブラリには現在，宇宙用 MPU，FPGA，メモリ等の設計部品が用意されている．

C-Based System Level Design Tool: Formal Verification for System Level Design.
Yuji Nishihara,Terumasa Koike,Tetsuya Yamamoto,and Masanobu Tsuji.
JAXA's Engineering Digital Innovation Center.

その次に，バスのアドレスおよびプロトコルを決定し，バスモデルを付加した通信モデルを，設計詳細化サブシステムの GUI を使って作成する．また，必要であれば，ユーザが各モデルの SpecC コードに手修正を加えることができる．

なお，今回は，仕様モデル作成，仕様モデル記述変更，設計詳細化によるアーキテクチャモデル生成という流れで設計を進め，適用例として，以下の組合せで等価性検証を行った．

・記述変更前仕様モデル－記述変更後仕様モデル
・仕様モデル－アーキテクチャモデル

※各モデルは全て SpecC 言語

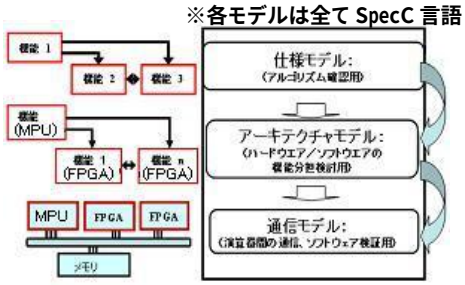

図 1　設計詳細化機能

3. 形式的等価性検証機能

ELEGANT では，設計詳細化前後のモデル間での等価性を保証する為に，設計詳細化機能を用いて生成された各モデル間での等価性の検証を行う形式的等価性検証機能を有する．具体的には，同一レベルの 2 つのモデルの等価性，および隣接するレベルの 2 つのモデルの等価性の検証が可能である．形式的等価性検証機能における，等価性の定義，等価性検証アルゴリズムを以下に示す．

3.1 等価性の定義

並列動作などを記述可能な SpecC 言語で記述されたモデルの等価性検証に当たっては，まず対象となる 2 つのモデルの入出力ポートの対応をとる．2 つのモデルの対応する入力ポートに同じタイミング，順序で任意の値やイベントが印加されたときに，2 つのモデルの対応する出力ポートから同じタイミング，順序で値やイベントが観測されるとき，2 つのモデルが等価であると定義する．なお，検証すべき出力ポートの値の比較を行うタイミングは，SpecC 言語で規定されているシミュレーションの抽象的アルゴリズムに基づいて，ELEGANT で定義した 4 つの方式から，ユーザ側で選択が可能となっている．タイミングの方式は，複数ステップから構成されるシミュレーション実行の比較対照ステップ(イベントによる同期ステップ，時刻による同期ステップ)，および比較条件(変化の順番，時間間隔)の組み合わせで分類される．

3.2 等価性検証アルゴリズム

図 2 に等価性検証アルゴリズムを示す．

まず，前処理として，検証対象である 2 つのモデル間でのビヘイビアおよびポートの対応をとり，次に，PEN の生成を行う．PEN とは Potential Equivalent Node pair の略で，2 つのモデル内部において等価である可能性がある変数の組である．そして，PEN を利用し，モデルから部分回路に相当する演算式を切り出し，切り出された 2 つの演算式が等価であるかどうかを記号シミュレーションにより検証する．以後，その処理を繰り返す．

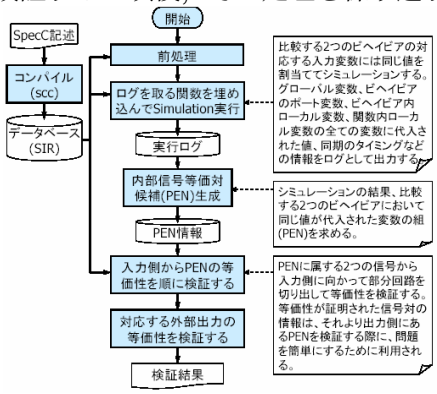

図 2　等価性検証アルゴリズム

4. 適用例

以下に形式的検証機能を用いた等価性検証の適用例とその結果を示す．

4.1 記述変更前仕様モデル-記述変更後仕様モデル

設計対象のアルゴリズム・機能を SpecC 言語で記述した仕様モデル(図 3)の記述変更前後の等価性検証を行った．なお，このモデルでは，ビヘイビア B1，B2，B3 は並列に動作しており，それぞれは，チャネルと呼ばれる通信部品でデータを受け渡すタイミングの同期をとっている．

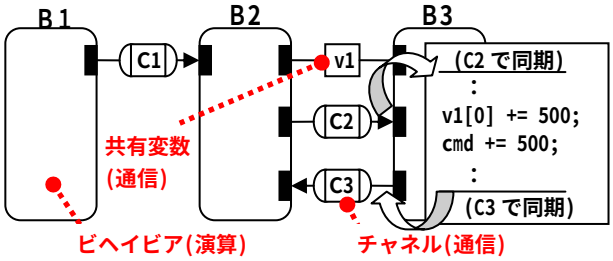

図 3　仕様モデル

記述変更では，図 4 のように，順次実行している独立した処理を並列実行処理にするために，B3 ビヘイビアを B3_1・B3_2 ビヘイビアへ分割した．

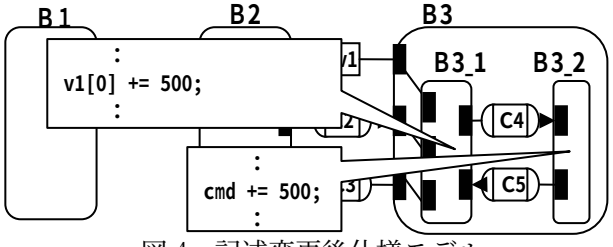

図 4　記述変更後仕様モデル

イベントによる同期ステップで，検証モデル間で対応する各ビヘイビアの出力ポートの値のサンプリングを行う方式で等価性検証を行った．その結果，仕様モデルに前述の記述変更を加えた前後での等価性が確認できた．

図 5 に，検証モデルを指定し，処理を実行する操作画面と検証結果を示す．

図 5　操作画面と検証結果

4.2 仕様モデル-アーキテクチャモデル

仕様モデルと，設計詳細化で作成した，アーキテクチャ構造と HW/SW の配分を決定したアーキテクチャモデル(図 6)の等価性検証を行った．

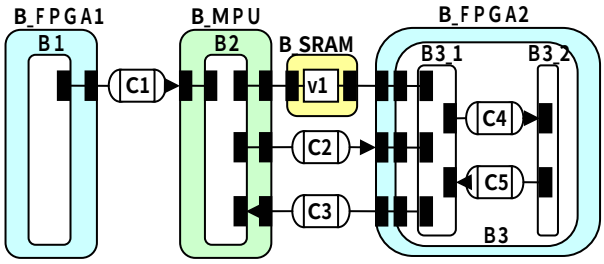

図 6　アーキテクチャモデル

図 7 の検証結果に示すように，仕様モデルと，設計詳細化により，アーキテクチャ構造に関する記述が追加されたアーキテクチャモデルとの間での等価性が確認できた．

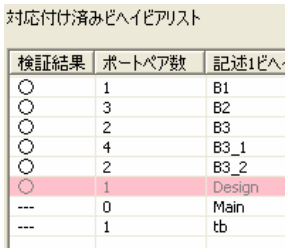

図 7　検証結果

5. まとめと今後の課題

C 言語ベースのシステムレベル設計ツールと，その設計詳細化機能で生成された各モデル間での等価性を保証する為の形式的等価性検証機能について報告した．また試作したツールが，HW/SW 協調のシステムレベル設計において，並列言語で記述されたモデルの等価性検証に活用できることを示した．今後の課題として，処理時間の高速化があり，内部処理やアルゴリズムの最適化を行う．

参考文献

[1] 山本徹也，西原雄次，小池輝昌，辻政信: C 言語ベースのシステムレベル設計ツールの試作－宇宙用デジタル電子機器設計への適用－，情報処理学会第 69 回全国大会，2007 年 3 月

[2] Daniel D.Gajski et al.著 木下常雄 他訳:SpecC 仕様記述言語と方法論,CQ 出版社