Vol.2011-SLDM-149 No.20
Vol.2011-EMB-20 No.20
2011/3/18

仕様記述言語によるＲＴＬ記述の生産性の試行評価

八田佳憲(*1)　泉知論(*1,*2,*3)　吉川寿広(*4)　荒木大(*4)

大規模化複雑化するシステムの開発には高い抽象度での設計による効率化が求められる一方，外部との接続部分や高性能を求められる部分では必要に応じて人手による詳細設計が行われる．高位設計から詳細設計まで一貫した設計言語・設計環境を使用することで，システム全体の開発の効率化を目指す．本稿では，仕様記述言語を用いて詳細設計を行った際の生産性や回路性能などについて，実際に設計の試行を行い従来のハードウェア記述言語による設計との比較評価を行った．記述法に習熟することで従来のハードウェア記述言語による詳細設計と比べて遜色の無い生産性と回路性能が得られる．

A Trial and its Productivity of Register-Transfer-Level Design in Specification Description Language

Yatsuda Yoshinori(*1)　Tomonori Izumi(*1,*2,*3)
Toshihiro Kikkawa(*4)　Dai Araki(*4)

Aiming to establish integrated design approach of abstracted specification description, behavioral description, and detailed structure description, we present a trial of register-transfer-level (RTL) design in a specification description language and discuss about the productivity of the design methodology compared to the conventional one in hardware description language. Our trial design demonstrates comparable productivity and circuit performance to the conventional can be achieved by getting accustomed to the new language and design tools.

1. はじめに

近年のＬＳＩの集積度向上に伴うシステムの大規模化・複雑化により，設計期間の長期化，開発コストの増大が大きな問題となっている．そのため，集積回路黎明期のトランジスタレベル設計，ゲートレベル設計から，現在のハードウェア記述言語によるレジスタ転送レベル設計，近年実用化されてきた動作レベル設計，さらにシステムレベル設計の研究開発と，設計の抽象度をあげての効率化が求められ進められてきている．そのような高い抽象での設計の為，システムを記述する言語と処理系が必要となる．SpecC [1] はシステムの仕様を記述するための言語として提案され，これによりシステムのモデリングと機能検証，システムレベルのタイミング仕様の記述と検証，コンポーネント分割や性能見積りを含むシステムアーキテクチャの検討，などの上流設計の効率化を狙っている [2].

一方で，高度な最適化のため，あるいは，外部回路とのインターフェースをとるため，サイクル精度のタイミングまで考慮した詳細な設計が求められる場面も依然存在する．一般には，システムレベルでの記述と検討，コンポーネント分割，ハードウェア化する部分については動作記述からの回路合成，そして必要に応じてサイクル精度記述による最適化，と設計を進めていくことになるが，段階が進むたびに記述言語や処理系が変わると，効率の低下をまねき，またミスが混入しやすくなる．そこで，SpecC では 2.0 版 [3]でレジスタ転送レベル記述のための言語仕様が追加され，回路における信号，レジスタ，状態遷移などが表現可能となった．

言語は仕様書ならびにリファレンスコンパイラにより規定されるが，実際に使用するにはシミュレータや合成ツールなどの処理系が必要である．SpecC の実使用環境の一例として，主に宇宙機器のための電子機器開発支援システム ELEGANT が挙げられる．ELEGANT システムでは SpecC が記述言語のひとつとして用いられ，設計詳細化支援ツール [4]やシミュレータ [5], 動作合成ツール [6]などの処理系が提供されている．さらに，このプラットホームを使用してのコンポーネント分割の研究開発 [7]などがすすんでいる．詳細設計については，ハードウェア記述言語への変換などの開発が進められており，一部試用段階に入りつつある [8] [9].

一般に，設計支援ツールは必ずしも言語仕様のすべてを扱えるわけではなく，言語仕様のサブセットのみを扱ったり特定の記述スタイルを要求したりすることが多い．また，複数のツールを複雑に組み合わせて使用する必要に迫られることもある．新しい言語や処理系が設計現場に受け入れられるには，試用とその知見の設計手法へのフ

(*1) 立命館大学 大学院 理工学研究科, Graduate School of Science and Engineering, Ritsumeikan University
(*2) 立命館大学 理工学部, College of Science and Engineering, Ritsumeikan University
(*3) 株式会社シンセシス, Synthesis Corporation
(*4) 東芝ソリューション株式会社, Toshiba Solutions Corporation

Vol.2011-SLDM-149 No.20
Vol.2011-EMB-20 No.20
2011/3/18

ィードバックを繰り返し，磨かれていかねばならない [9) 10)].

本稿では，SpecC による詳細記述設計の試行を行い，その生産性や結果として得られる回路の性能について評価を行う．設計例として 4 チャンネル可変周期信号発生器を取り上げ，SpecC によるサイクル精度の記述ならびにその処理系である VisualSpec 4 [9)]を用いた動作検証とハードウェア記述言語への変換，そして FPGA 向けの開発環境での論理合成を行う．併せて，同じ機能について，ハードウェア記述言語 Verilog-HDL による記述ならびに論理合成を行う．それぞれの試行について，設計に要する時間や記述量などの生産性，合成された回路の規模や最大動作周波数などの性能について比較評価を行う．

2. 高位言語による設計

2.1 システム設計の流れ

高位言語によるシステム設計の流れを図 1 に示す．設計対象のモデル化・表現の抽象度が高い状態から徐々に詳細で具体的な状態に設計を進めていく．作業工程は上流のシステム設計，中流の機能設計，下流の論理設計・レイアウト設計の大きくみっつに分けることができ，それぞれの段階ではそれぞれの抽象度にあった記述レベルの言語が用いられ，設計が行われる．

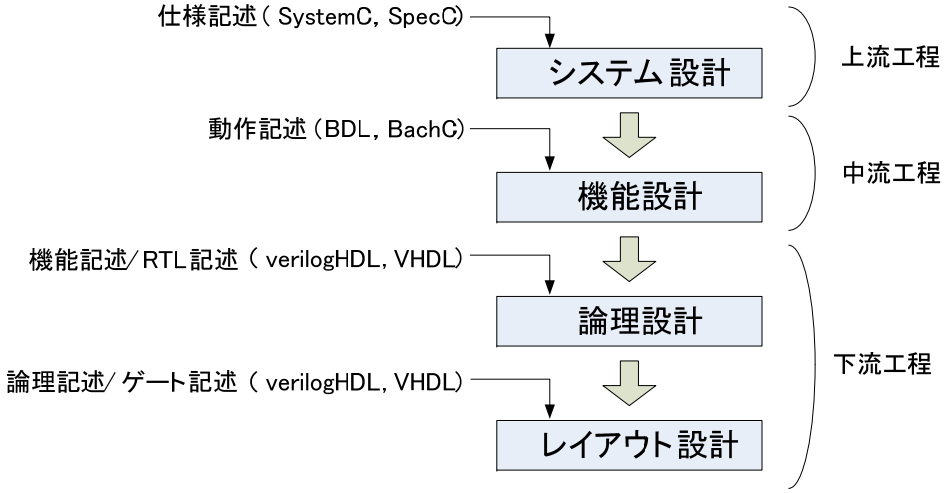


図 1　システム設計の流れ

上流工程では全体の機能やシステムの正当性の検証や，またハードウェアで実装する部分とソフトウェアで実装する部分の分割の検討などを行う．中流工程は回路設計に使用される段階で，動作レベルの記述を行い，これに性能や回路面積などの条件を付加し，専用ツールでコンパイルを行うことによって，機能記述（RTL）へ変換する．高位記述からのコンパイルで得られた回路で目標性能を達成することができれば，設計は完了する．ただし，動作記述から機能記述への変換は最適化の余地が大きく難しい問題であり，必ずしも人手で直接記述された機能記述ほどの性能が得られるとは限らない．数クロックずれても動作に支障がない，並列性がそれほど必要でない，設計の制限が少ない部分はこの手法で短期間で開発することができる．しかし，より高い性能が求められる部分は必要に応じて人知による詳細な機能記述を行う必要がある．下流工程では，論理回路としての構成を定める論理合成，部品の配置や配線など詳細な設計を行う．

2.2 SpecC 言語

SpecC 言語は当初ハードウェアとソフトウェアを含むデジタル組み込みシステムの仕様と設計を形式的に記述する言語として開発された．ソフトウェア記述言語として一般的な C 言語を基に構成されており，C 言語の標準規格である ANSI-C プログラム言語の最上位層に組み込まれている．当初 SpecC 言語は上流工程での設計のみを扱うものであったが，システム設計の全工程を扱うため，後の SpecC 2.0 で，中流・下流工程の設計向けの言語仕様が追加された．これまでは各工程によって用いられる記述言語が異なることから，より詳細な設計に進める際に別の言語で書きなおす必要があり，無駄な手間を強いられていた．しかしひとつの言語で統一して扱うことができれば，それぞれの工程間にある壁が取り除かれ，効率的に設計が進められる．また，ソフトウェア記述言語である C 言語をベースとしており，その動作を実行形式にコンパイルしてシミュレーションを行うため，ゲート単位でのイベント駆動型の動作をするハードウェア記述言語ベースのシミュレーションよりも高速に実行できる．さらに，ソフトウェア開発者には敬遠されがちな Verilog HDL よりも，馴染みやすいといった利点も挙げられる．

2.3 SpecC 言語による設計フロー

SpecC 言語による設計フローを図 2 に示す．上流・中流工程で抽象度の高い設計を行い，システムレベル設計支援ツール SER による設計の詳細化，動作合成ツールによ

Vol.2011-SLDM-149 No.20
Vol.2011-EMB-20 No.20
2011/3/18

る回路合成，さらに必要に応じて一部分を取り出して人手による詳細設計を行う．詳細記述では，SpecC 2.0 で拡張されたサブセットである SpecC RTL を用いる．各工程では設計支援ツールや合成ツールが利用できるほか，人手による詳細化に関しても一から書き直すのではなく，統一言語を用いていることにより，前工程の記述を利用して書き換えていくことで次の工程の設計を進めることができる．このように全工程を一貫して設計する為の言語やツール環境は整いつつあり，これにあわせて設計法や記述法，ツール群の扱い方など，実運用のための知見の集約と整理を進めていく必要がある．

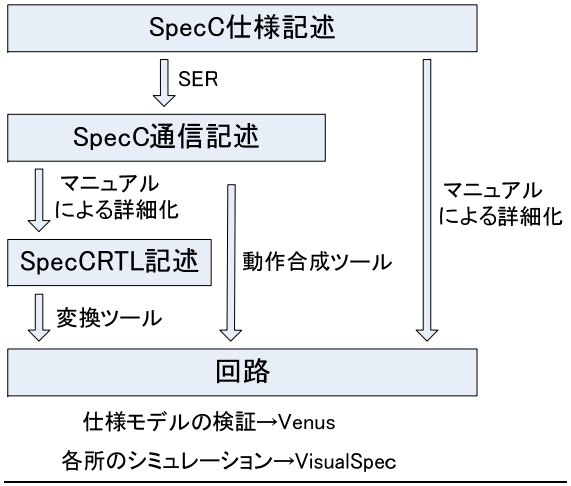


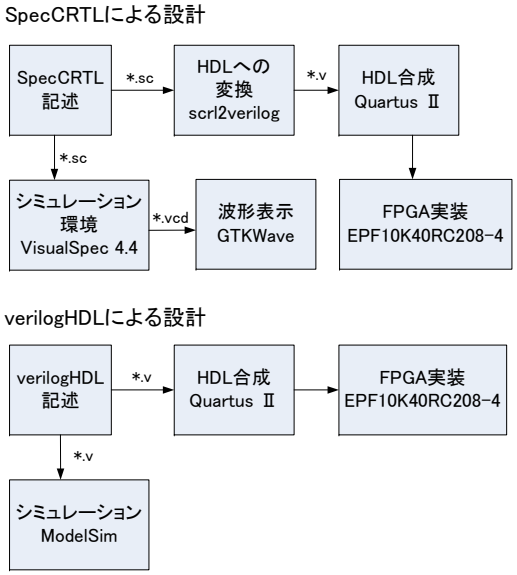
図 2　SpecC 言語による設計フロー

3. 設計試行実験～4ch 可変周期信号発生器

3.1 試行の概要

我々は，SpecC 言語を用いた全工程一貫設計により高位設計と詳細設計を融合した効果的な設計方法論の検討を進めている．そのため，まず SpecC RTL による詳細設計に関して，設計の試行による記述法や設計手法の検討と評価を行う．ここでは４チャンネル可変周期信号発生器を設計対象とし，SpecC RTL を用いた記述からハードウェア記述言語 Verilog HDL に変換し，合成を行って実際にＦＰＧＡボード上で動作させる．また，SpecC 言語による一貫設計が全体の生産性向上に寄与することは期待しつつも，詳細設計に限れば長年ハードウェア記述言語により行われてきた実績がある．SpecC RTL による詳細設計と従来の Verilog HDL による設計を比較し，生産性や設計結果の性能についての評価を行う．

今回の試行に用いた設計環境の概要を図 3 に示す．SpecC RTL 設計では，インターデザイン・テクノロジー社の SpecC 設計環境である VisualSpec 4.4 でシミュレーションを行い，結果を VCD ファイルとして出力して，BSI の波形ビューアである GTKWave を用いて波形により動作を確認した．その後，インターデザイン・テクノロジー社の scrtl2verilog によって SpecC RTL から Verilog HDL へと変換し，ALTERA 社の開発環境 Quartus-II で同社 FPGA の EPF10K40RC208-4 をターゲットとして，論理合成と実装を行った．また，Verilog HDL 設計では Mentor Graphics 社の ModelSim でシミュレーションを行い，その後は SpecC RTL 設計と同様に Quartus-II を用いて論理合成と実装を行った．



Vol.2011-SLDM-149 No.20
Vol.2011-EMB-20 No.20
2011/3/18

図 3　設計環境

3.2 回路仕様

図 4 に設計した 4 チャンネル可変周期信号発生回路のブロック図を示す．可変周期信号発生器（図中④）を 4 つ搭載しており，入力である選択信号 0～7 は入力検知モジュール（図中①）により検知され，アービター（図中②）により発生器を選択して，それぞれの入力に対応する周期を設定（図中③）する．発生器は設定された周期の方形波を発生し，それら 4 チャンネルの信号はミキサー（図中⑤）により合成され，出力される．

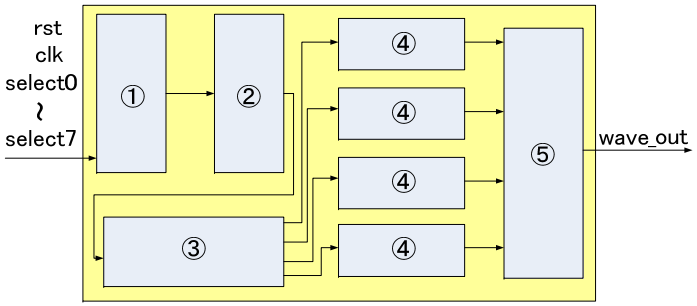


図 4　4 チャンネル可変周期信号発生器

3.3 生産性と合成結果の比較評価

3.3.1 設計時間

時間は，生産性を評価する上での重要なパラメタであるが，設計者の言語習熟度の差が結果に大きな影響を与える．厳密な比較は難しいが，いち試行の結果としてデータを示す．今回試行した設計者は，電子情報系の研究室に所属する大学院生であり，大学の講義で Verilog HDL を学び，設計演習や研究室での課題等で 4 年間使用している．設計回数は 40 回程度，ソースコード行数は最大 200 行程度である．SpecC RTL は研究室にて独学で学び，その後 1 年間使用している．設計回数は 10 回程度，ソースコード行数は最大 150 行程度である．設計者の習熟度の差により Verilog HDL の方が時間が短くなる傾向があると思われる．また，同一のモジュールを同一設計者が 2 種類の言語で設計した場合，先の設計でモジュールそのものの仕様や機能の理解が深まり，実装のノウハウなどが得られるため，後からの設計の方が時間が短くなる傾向があると考えられる．そこで，設計工程を分け，工程毎に Verilog HDL と SpecC RTL とで順番を替えて設計を進めた．表 1 に，Verilog HDL と SpecC RTL で設計に要した時間，Verilog HDL に対する SpecC RTL の時間の比を示す．各工程での作業として，工程 1 で入力スイッチなしの固定周波数信号発生器，工程 2 で固定周波数信号発生器，工程 3 で 3 チャンネル固定周波数信号発生器，工程 4 で 8 チャンネル固定周波数信号発生器，工程 5 で 4 チャンネル可変周波数信号発生器と徐々に拡張しながら，それぞれ詳細仕様の検討，記述，シミュレーションによる動作確認を行った．

表 1　設計に要した時間の比較

	Verilog HDL	SpecC RTL	比
工程 1	5.0 時間（先）	15.0 時間（後）	300%
工程 2	4.5 時間（後）	17.5 時間（先）	389%
工程 3	7.5 時間（先）	9.0 時間（後）	120%
工程 4	7.5 時間（後）	16.5 時間（先）	220%
工程 5	13.5 時間（先）	12.5 時間（後）	93%

今回の時間比較においては，やはり SpecC RTL による設計に長い時間がかかっている．特に初期の工程では顕著で，これは SpecC RTL 言語を用いた設計の経験が浅く，記述方法やツールの使い方についての試行錯誤も発生してしまったためである．しかし，設計が進むにつれ，徐々にその差は小さくなっている．2 つの言語に対する設計者の言語習熟度に差があることを考慮すれば，時間差はそれ程大きくないと思われる．

3.3.2 記述量

生産性の評価指標のひとつとして記述量が挙げられる．記述量が少なければ，記述にかかる時間が短くなり，ミスが混入する可能が減り，全体像を掌握しやすく，保守性が向上するといった傾向があると考えられる．表 2 に Verilog HDL と SpecC RTL それぞれによるソースコードの行数と Verilog HDL に対する SpecC RTL の比を示す．SpecC RTL による記述が Verilog HDL による記述を上回っている．SpecC RTL では，並列動作を起動する部分やクロック同期のための記述など，本来はシーケンシャルな動作を記述するための文法のもとで並列動作を表現し実現するための技巧的な記述を要し，そのために形式的な記述量が増える傾向にある．また，モジュール（ビヘイビア）の宣

Vol.2011-SLDM-149 No.20
Vol.2011-EMB-20 No.20
2011/3/18

言部分で Verilog HDL では複数の入出力をまとめて記述できるのに対し，SpecC RTL では図 5 のように一行ずつ記述するため，記述量が増える傾向にある．ただし，Verilog HDL と SpecC RTL では設計対象の抽象度はほぼ同じであり，設計対象の機能に関する記述についてはほぼ同等の記述量となっている．

表 2　記述量の比較

	Verilog HDL	SpecC RTL	比率
工程 1	22 行	35 行	159.10%
工程 2	39 行	113 行	289.74%
工程 3	117 行	183 行	156.41%
工程 4	152 行	308 行	202.63%
工程 5	236 行	291 行	123.31%

```
behavior ctrl_bh(in  signal unsigned bit[8] button,
                 out signal unsigned bit[16] change1,
                 out signal unsigned bit[16] change2,
                 out signal unsigned bit[16] change3,
                 out signal unsigned bit[16] change4,
                 out signal unsigned bit[1] trig1,
                 out signal unsigned bit[1] trig2,
                 out signal unsigned bit[1] trig3,
                 out signal unsigned bit[1] trig4){
};
```

図 5　SpecC RTL の記述例

3.3.3 回路規模

設計の結果として得られた回路の性能について評価を行う．回路規模に関して，表 3 に Verilog HDL と SpecC RTL それぞれによる設計により得られた回路のロジックエレメント数と Verilog HDL に対する SpecC RTL の比率を示す．全体としてはほぼ同程度の回路規模となっている．

3.3.4 最大動作周波数

回路の速度に関して，表 4 に Verilog HDL と SpecC RTL それぞれによる設計により得られた回路の最大動作周波数と Verilog HDL に対する SpecC RTL の比率を示す．全体としてはほぼ同程度のものとなっている．

表 3　合成結果の回路規模の比較

	Verilog HDL	SpecC RTL	比率
工程 1	6	7	116.67%
工程 2	11	10	90.91%
工程 3	95	93	97.89%
工程 4	220	228	103.64%
工程 5	357	331	92.72%

表 4　合成結果の最大動作周波数の比較

	Verilog HDL	SpecC RTL	比率
工程 1	125.00MHz	104.17MHz	83.34%
工程 2	107.53MHz	125.00MHz	116.25%
工程 3	45.87MHz	47.39MHz	103.31%
工程 4	39.53MHz	39.22MHz	99.22%
工程 5	26.88MHz	25.77MHz	95.87%

4. おわりに

高位記述言語による詳細記述設計の試行を行い，その生産性を評価した．試行者は従来のハードウェア記述言語による設計経験があり，高位記述言語による設計の経験は浅かったが，速やかに新しい方法に移行することができ，次第に習熟していった．いち設計者によるいち試行ではあるが、最終的には，生産性も合成された回路の性能も従来手法に比べて遜色の無い結果を示している．試行者らは，さらに習熟すれば，一般的にサイクル精度記述に用いられるハードウェア記述言語よりもむしろ高位記述言語のほうが，生産性が高いだろうという感触を得ている．高位記述言語により，仕様記述からサイクル精度記述まで言語やツールを切り替えることなく，一貫した効率的な設計が可能となる．そのためには，さらに，動作合成向けの記述とサイクル精度の記述の混在など，様々な設計事例を積み重ね，記述法や設計手法を確立していく必要がある．

参考文献

1)　Domer, R., Gerstlauer, A. and D. Gajski, D.: SpecC Language Reference Manual, Ver.2.0, SpecC Technology Open Consortium (2002). http://www.specc.gr.jp/tech/SpecC_LRM_20.pdf

Vol.2011-SLDM-149 No.20
Vol.2011-EMB-20 No.20
2011/3/18

2) Gajski, D. et.al 著，木下常雄，冨山宏之訳: SpecC 仕様記述言語と方法論，ＣＱ出版 (2000).
3) Fujita, M.: SpecC Language Version 2.0: C-based SoC Design from System level down to RTL, tutorial, Asia and South Pacific Design Automation Conference (ASPDAC) (2003).
4) Gerstlauer, A., Peng, J., Shin, D., Gajski, D., Nakamura, A., Araki, D., Nishihara, Y.: Specify-explore-refine (SER): from specification to implementation, in Proc. Design Automation Conference (DAC) pp.586-591 (2008).
5) 荒木大：ELEGANT の SpecC シミュレーションと設計詳細化について，第三回先端宇宙情報技術ワークショップ講演集 (2007).
6) 若林一敏：ELEGANT で使う動作合成について，第三回先端宇宙情報技術ワークショップ講演集 (2007).
7) 松永惇弥，村岡道明，荒木大：ソフトウェア並列化を考慮したハードウェア/ソフトウェア分割手法，電子情報通信学会技術研究報告 VLD2009-71, CPSY2009-53, RECONF2009-56 (2010).
8) VisualSpec Version 4 利用ガイド, InterDesign Technologies, Inc. (2008).
9) 泉知論，吉川寿広，荒木大：仕様記述言語 SpecC によるサイクル精度記述の一試行，情報処理学会 研究報告 2010-SLDM-144(17), 2010-EMB-16(17), 2010-MBL-53(17), 2010-UBI-25(17) (2010)
10) 井上諭，橋詰大毅，泉知論，福井正博：高位記述言語を用いた画像処理向けディジタルシステム設計の効率化手法，電子情報通信学会技術研究報告 VLD2006-123, ICD2006-214 (2007).