C-001

ブロックベース設計を用いた統合環境による効率的なハードウェア開発手法の評価

The Evaluation of the efficient Hardware Development Method by Integrated Environment with Block-Based Design

松田 昭信　　**石原 亨**
Akitoshi Matsuda　　Tohru Ishihara

1．はじめに

現在，デジタル家電等における高機能化及び大規模化に従い，これらを構成するシステムLSIにおいては，開発期間・開発コストの増大や品質問題が多発している．これらの各設計フェーズにおいて，設計期間短縮と設計品質向上のためには，ハードウェアのシステムレベルの構想及び設計段階からの対策が必要である[1]．その対策の一つとして，ハードウェア設計にブロックベース設計手法の導入が検討されている．ここでのブロックベース設計とは，図 1 のように，ハードウェアを構成するブロック(加算・乗算器，マルチプレクサからフィルタなどの大規模機能まで)を組み合わせてハードウェア回路を実現することと定義する．これらのブロックからは，ハードウェアを構成するHDLのみならず，ソフトウェアを構成するC言語も自動生成可能である[2]．このC言語を入力とする動作合成技術を適用する設計手法[3],[4]へも適用でき，ハードウェア設計において非常に効率的である．そこで，このブロックベース設計手法を用いて，アルゴリズムレベルの演算処理部を，ハードウェアブロックに変換した．これにより，回路構成の全体構造を可視化すると共に，不要処理の削除などのハードウェア開発の最適化が容易となる．今回は，斬新かつ効率的なハードウェア開発手法を考案した．本稿では，その評価事例について述べる．

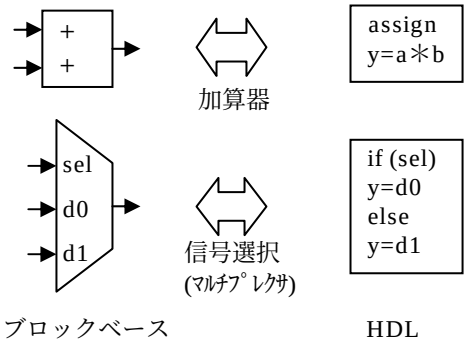

図 1．ハードウェアブロック

2．ブロックベース設計でのカウンタ記述例

ブロックベース設計手法を用いて簡単なハードウェアブロックであるカウンタを作成する方法を示す．ハードウェアを実現するため，図 2 に示すようなイベントの発生毎に出力値を 1 ずつカウンアップさせるブロックが完成する．

九州大学大学院　統合新領域学府, Graduate School of Integrated Frontier Science, Kyushu University

これをハードウェア様式に適合させるため，入力及び出力信号のビット数やリセット信号の条件などを設定することにより，条件を満たした HDL が自動生成するブロックが完成する．このように，ブロック作成により簡単に図 3 のような HDL が自動生成できる．

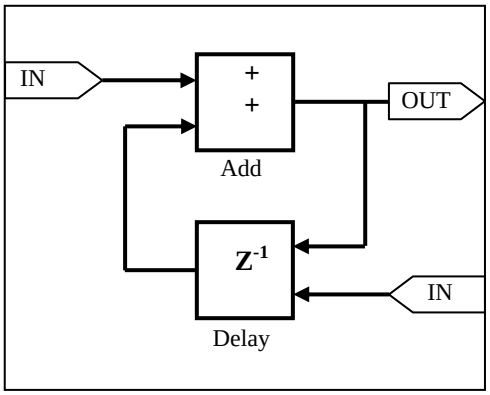

図 2．ハードウェアブロック

```
module counter (clk, reset, dout);

  input   clk, reset;
  output [3:0] dout;
  reg    [3:0] dout;

always @(posedge clk or negedge reset)
     begin
      if(!reset)
   dout <= 4'h0;
 else
        dout <= dout + 4'h1;
end

endmodule#
```

図 3．HDL（Verilog-HDL）コード

3．開発事例

今回は，デジタル画像処理システムを構成する DCT 変換アルゴリズムをモチーフとして選択した．そのアルゴリズムをハードウェア化するにあたって，まず全体構造や処理フローを理解する必要がある．しかし，アルゴリズムそのものを理解するには，人的ミスや理解不足が発生する可

能性が大きい．よって，このアルゴリズム（C ソースコード）を図 4 のようなブロック化されたモデルに変換すれば，全体構造が容易に理解することができる．

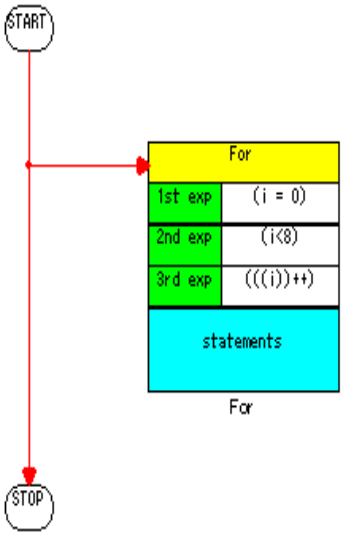

図 4．ブロック図変換手法の概要

このように，今回 C ソースコードから処理モジュールごとに分けられたブロックに変換すると同じタイミングで，C ソースコードの不要変数の削減，if 文の結合，for 文の結合などの C ソースコードの最適化を施すことにより，ハードウェア化するにあたっても，一定の効果が得られた．また，このブロックベース設計手法を用いることにより，設計期間の短縮や設計工数の削減にも効果を挙げることができた．

今回は，ブロックベース設計環境で検証したアルゴリズムから自動的に HDL へ変換する機能を使用して生成した回路と，仕様書レベルからハンドコーディング（手設計）によって HDL を生成した回路との比較方法をおこなった．ブロックベース設計手法により，ハードウェアブロックから HDL の書換え作業が不要となり，さらに，仕様検討が終了すると同時に，ハードウェア・ソフトウェア設計が並列に検証できる．これにより，イタレーション及びリスピンが減り，全体的な設計工数も短縮される．今回の設計事例では，図 5 にあるように，設計工数が 52(h)時間から 31(h)時間に短縮しており，全体の 40%工数削減が実現できた．また，この C ソースをモデル変換してブロック図にすると同時に，最適化を実施した．この C ソースにおいて最

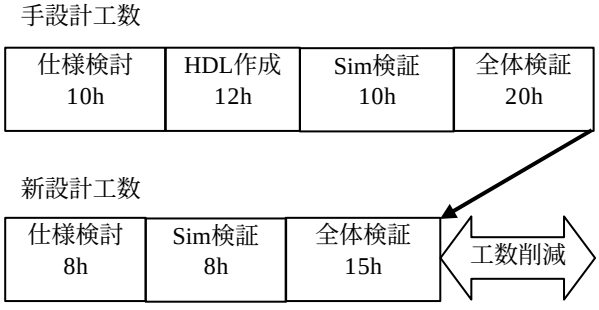

図 5．設計工数の比較

表 1．回路性能比較

評価項目	最適化後	最適化前
最大周波数(MHz)	40.17	40.28
面積(Gate)	4,104	6,480
FF使用数(個)	180	630
Slice使用数(個)	351	627
LUT使用数(個)	610	601

適化前後で動作合成を実施後の回路比較結果を表 1 に示す．これによると，パフォーマンスは大きな変化が見られなかったが，面積は 37%縮小されている．これらから，今回のブロックベース設計手法の適用により，設計工数及び回路性能ともによい結果が得られたことがわかる．

4．まとめ

今回のブロックベース設計手法により，アルゴリズム検証から FPGA のシステム記述，詳細設計まで一貫してほぼ自動化することによって，システムレベル設計が高品質かつスピーディに実行できることがわかった．また，C ベース設計よりもう一段階抽象度を上げたブロックベース設計手法の効果を確認できた．

今回の結果は，あらゆるケースにおいて，このような評価結果は出ないのかもしれないが，着実に設計期間を短縮することは見てとれる．また，動作合成ツールなどの様々なツールに過度に依存することなく，ベンダーフリーなブロックをデータベースに蓄積することも可能となった．また，これらの可視化（見える化）されることにより，設計資産の再利用にも貢献できると考える．

5．今後の課題

現状では，生成された HDL コードの等価性検証及びプロパティ検証などが十分に実施できる環境が整っていない．C ソースのブロックと HDL のブロック間において接続関係を設定するだけで，コ・シミュレーション及びコ・ベリフィケーションが実現できれば，検証時間短縮にも貢献できると考える．これらの課題についても，今後さらなる検討を進めていく予定である．

参考文献

[1]Semiconductor Industry Association, “ The National Technology Roadmap for Semiconductors-Technology Needs , ” 3rd Edition, 1998.

[2] J.Staunstrup, W.Wolf : “Hardware/Software Co-Design: Principles and Practice”, Kluwer Academic Publishers, 1997.

[3]松田昭信,”高位合成手法を用いた C ベース設計による LSI 開発事例”,情報処理学会,第 67 回全国大会, 1-99,100 ,2005.

[4]L.Semeria, K.Sato, and G.De Miceli: “Synthesis of Hardware Models in C with Pointers and Complex Data Structures,” in IEEE Trans. VLSI Systems, vol. 9, no.6, pp.743-756(Dec. 2001).