HW/SW 協調動作に対するブロックダイアグラム環境利用に関する一評価

Evaluation of the System Design Environment Utilizing Block Diagram Tools for FPGA/DSP-Based System

C-4

橋本 耕太郎† 櫻木 誠† 田中 康一郎‡ 佐藤 寿倫†‡ 有田五次郎†
Kohtaro Hashimoto Makoto Sakuragi Koichiro Tanaka Toshinori Sato Itsujiro Arita

1．はじめに

我々は，SoC の研究を行うためのハードウェアプラットフォームとして，FPGA/DSP/DIMM を搭載した PCI カードである RYUOH[1]を開発し，それに対するシステム設計を行ってきた．RYUOH によるシステム設計では，FPGA と DSP に対し，HDL と C 言語によりそれぞれ個別に設計/検証が必要となる．

一方，大規模集積回路やそれらを組み合わせたシステムのための新たなシステム開発環境が切望されている．従来のハードウェア(HW)とソフトウェア(SW)を個別に設計する手法では，それらを短期間に実現することは難しく，双方を同時に行えるコデザイン環境が今後必要となると予想される．そこで我々は，RYUOH に対する HW/SW 協調動作システムの開発環境として，ブロックダイアグラムによるコデザイン環境に着目した．

本稿では，コデザイン環境の実現に向けた調査結果について報告する．ブロックダイアグラムによるシステム開発の概要と，従来の手法との相違点及び問題点，また，JPEG エンコードの設計事例を基にモデル設計と自動生成コードの検証について述べる．

2．ブロックダイアグラムによるシステム開発

ブロックダイアグラムによるシステム開発の概要，従来の手法との相違点について述べる．また，その問題点について説明する．

2.1 システム開発環境及び概要

ブロックダイアグラムによるシステム開発環境として MathWorks 社の MATLAB 及び Simulink[2]を用いた．Simulink ではブロックライブラリを用いた階層的なモデル設計が可能であり，さらにシステムレベル設計からシミュレーションまでを容易に行うことができる．要求するブロックが存在しない場合は，C 言語により記述されたコードを S-Function ブロック[2]で読み込むことにより，特定の動作を行うブロックとして使用する．また，Simulink により設計されたモデルは，コード生成ツール Real-Time Workshop[2]によりターゲットに対応したコードを自動生成する．

2.2 従来の開発手法との相違点

従来の手法では，開発するシステムを HW で処理する部分と SW で処理する部分とに分割し，それぞれ個別に設計した後にシミュレーションを行ってきた．しかし，この手法を用いると，システム全体のシミュレーションは HW/SW に実装した後に行うことになり，フィードバックに必要な期間が増加する．それに対し，ブロックダイアグラムによるシステム開発では，システムレベルでモデルを設計することにより，コード生成を行う前に実際の動作をシミュレートすることができる．

2.3 システム開発の問題点

Simulink では double 型の処理を基本としており，信号処理用のブロックライブラリの大半が double 型の入出力を扱っている．整数演算による処理は Fixed-Point Blockset[2]を使う必要があるが，用意されているブロックは double 型と比較して不足しており，整数演算による信号処理を用意されているブロックだけで設計することは困難である．整数演算による信号処理を行う他の方法のとして，必要な整数処理を C 言語記述して S-Function ブロックに組み込むといった方法があるが，この方法ではブロックダイアグラムによるシステム開発の利点が損なわれる．これらの問題点を次の章で検証する．

3．JPEG エンコーダにおける設計事例

ブロックダイアグラムによるシステム開発により，設計は容易となるが，自動生成されたコードの品質が問題となってくる．そこで，JPEG エンコーダ[1]を設計し，自動生成されたコードと手書きによる C コードとの比較を行った．

3.1 JPEG エンコーダ

RYUOH に対して設計した JPEG エンコーダの処理の一部として，以下に示す 3 つの処理を設計した．エンコードには処理方式が 4 つ存在しているが，最も多く利用されている基本 DCT 方式を使用した．

色空間変換

入力された RGB データを式(1)により YCbCr データに変換する．

$$\begin{cases} Y=(76\times R+150\times G+28\times B-32768)>>8 \\ Cb=(-43\times R-84\times G+128\times B)>>8 \\ Cr=(128\times R-107\times G-20\times B)>>8 \end{cases} \quad (1)$$

2 次元離散コサイン変換(2D-DCT)

画像を 8x8 のブロックに分割し，そのブロックに対して 2D-DCT を適用する．

量子化

2D-DCT 後のデータに対して，用意されている量子化テーブルとの除算を行う．

3.2 Simulink によるモデル設計

Simulink により JPEG エンコードの 3 つの処理に対してモデルを設計した．ブロックは基本的に Fixed-Point Blockset を用いて整数演算となるようにした．2D-DCT のブロックは存在しないが，2D-DCT は 1D-DCT を 2 回適用することと等価であることから，1D-DCT のブロックを用いることにした．1D-DCT は，double 型ブロックしか存在しないため，double 型 1D-DCT を利用した．なお，ブロッ

† 九州工業大学 情報工学部 知能情報工学科
‡ 九州工業大学 マイクロ化総合技術センター

クが存在しない int 型に対するシフト処理などに対しては，S-Function ブロックを用いて C 言語により処理を行った．

例として，Simulink による色空間変換のモデルを図 1 に示す．このモデルによって式(1)の処理を行っている．

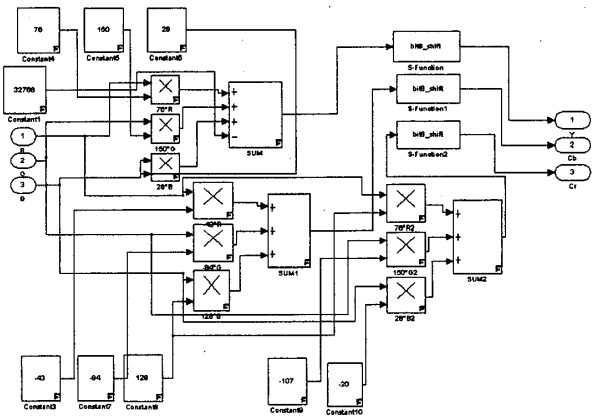


図 1. Simulink による色空間変換のモデル

3.3 自動生成コードの検証

Simulink により設計した各モデルに対して，Real-Time Workshop を使い TI 社製 DSP の C67x[3]向けのコードを自動生成し，手書きのコードとの実行サイクルの比較を行った．比較結果を表 1 に示す．なお，入力するデータはサイズ 8x8 の short 型とした．

C67x は VLIW アーキテクチャを採用しており，8 個の演算ユニットを持つ．高い処理性能を実現するには，この演算ユニットを効率よく利用することが重要となる．手書きコードは，2D-DCT および量子化に TI 社のライブラリを使用した．各処理は演算ユニットを効率よく利用する手法であるソフトウェアパイプラインが可能なように設計した．

表 1. Simulink と手書きコードによる実行サイクル数の比較

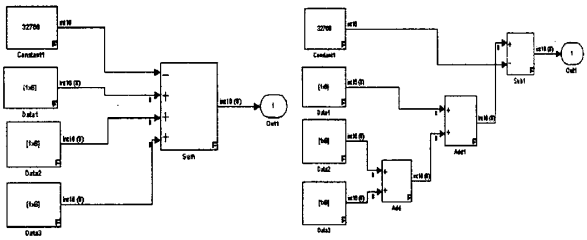
	色空間変換	2D-DCT	量子化
Simulink	1,505	178,865	761
手書きコード	301	528	234

結果から Simulink モデルからの自動生成コードでは，手書きのコードよりも基本処理性能が低いことがわかる．2D-DCT については，手書きのコードでは固定小数演算で実現されるライブラリを使用しているのに対して，Simulink では double 型の 1D-DCT を 2 回使用し，データの入れ替えも行っているため処理性能が大幅に低い結果となっている．

基本処理性能が大きく低下している原因の一つとして，どのモデルにおいてもソフトウェアパイプラインが効率よく適用できていないことが確認できた．そこで，ブロック数による処理性能の検証を行い、演算ユニットがどれくらい利用されているかを検証した．

3.4 ブロック数による処理性能の検証

ブロック数の違いによる自動生成コードの処理性能の検証として，short 型サイズ 8 の 3 加算 1 減算を 1 ブロックで行うモデルと，3 つのブロックで行うモデルでの実行サイクル数の比較を行った．各モデルを図 2 に示に，比較結果を表 2 に示す．



(a) 1 ブロックによる演算　　(b) 3 ブロックによる演算

図 2.　3 加算 1 減算のモデル

表 2. ブロック数による実行サイクル数の比較

	実行サイクル数
1 ブロック処理	42
3 ブロック処理	77

結果から，ブロック数が増すと処理性能が低下することがわかった．生成された C コードとそのアセンブリコードを比較した結果，複数のブロックにまたがった最適化が行われていないことが判明した．これは，ブロックごとの処理を関数レベルで行っているためである．

3.5 考察

今回の検証結果から，Simulink により処理性能の高いシステムを開発していく上で，整数演算可能なブロックライブラリ数の不足，ブロックレベルでの最適化しか行われていないため，全体的な最適化が可能な手書きのコードと比較して処理性能が低下するといった問題が確認できた．システム開発環境として利用するためには，整数演算ブロックライブラリを増加し，また，同時に展開実行できるブロックを自動で検出し，まとめて最適化を行うようなシステムを取り入れるといった方法が考えられる．

4. おわりに

本稿では，HW/SW 協調処理システムを行うためのシステム開発環境として，Simulink についてブロックダイアグラムによるシステム開発の概要について述べた．また，JPEG エンコーダの設計事例を基に，モデル設計および自動生成されるコードについて検証を行った．その結果，手書きのコードと比較して処理性能が大幅に低下することがわかった．その主な原因として，ブロックにまたがった最適化が行われないことが判明した．

今後は，効率的なモデル設計方法の検討及び Simulink による FPGA 設計ツール/ブロックライブラリである Xilinx System Generator[4]での FPGA 設計の検証を行っていく．

参考文献

[1]林悠平 他："FPGA/DSP による協調処理環境の構築"，信学技報，CAS2002-45(2002).

[2]MathWorks Inc. : MATLAB, Simulink, Real-time Workshop, Fixed-Point Blockset, S-Function, http://www.mathworks.com/.

[3]TEXAS INSTRUMENTS Inc. : DSP C67x, http://www.tij.co.jp/.

[4]Xilinx Inc. : Xilinx System Generator http://www.xilinx.com/.