

専用プロセッサ設計支援システム (SYARDS)

におけるパイプライン処理の考慮

9M-3

吉田 裕

雨坪 孝尚

白井 克彦

早稲田大学 理工学部

1 はじめに

LSIの製造技術の進歩に伴い、様々なシステムのLSI化が実現されている。そしてVLSIの規模が増大し、その応用分野も急激に広がりつつある。今後、LSIの多様な分野への応用が進む中で、一層、カスタムVLSIへの要求が増大していくと考えられる。

その反面、システムが大規模になると、様々な最適化やテストが複雑で困難な問題となり、設計期間の長期化とコストの増加が生じる。今後の専用LSIプロセッサの要求の増大に応えるためには、より容易に短期間でLSIを設計することのできる設計支援システムが必要不可欠となる。

そこで、我々は高位仕様記述(Pascal)を入力とした設計支援システムの研究、開発を行っている。本報告では、パイプライン処理の考慮による速度性能の向上を目的とし、その方法と具体例を示す。

2 システムの概要

本システムは、中間情報生成系、解析系、合成系の3つの系からなる(図1)。Userは、ほぼPascalの仕様にしたがって、ハードウェアとして実現したいアルゴリズムの仕様記述を記述し、実行するサンプルデータと共に入力する。

- 中間情報生成系

仕様記述に対して、字句・構文解析、冗長なコードを削除する等の種々の解析を行い、中間情報として3番地文によるフロー情報を出力する。

- 解析系

サンプルデータに基づいてシミュレーション、ブロック・命令ごとの頻度解析を行い、機能ブロックとして合成できる部分をグラフ構造として取り出し、Userに結果を出力する。またここで基本ブロックごとにアルゴリズムを変えることなくパイプライン処理に最適なように、命令を並び換えていく。

- 合成系

Userより与えられた機能ブロック・命令・レジスタの諸制限の中で割付を行い、システムの出力として使用した命令・機能ブロック・レジスタの情報とそのフロー情報をハードウェア情報として出力する。

Userは、解析系での結果と合成系より得られたコードの再度のシミュレーションにより、速度と資源の制約を満たしたハードウェアを合成して行く。

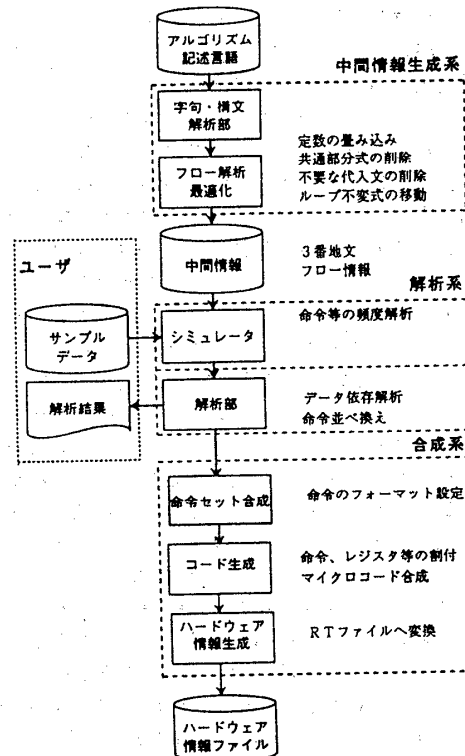


図1: システム構成図

3 パイプライン処理の考慮

本研究では、パイプライン処理を考慮し速度性能を上げることが目的とする。ここで取り上げているパイプライン処理では、一つの命令を3段に分割し並列に処理できる部分は並列に実行する。各ステージの機能はまず第1ステージでは命令の読みだしとデコード、第2ステージではオペランドの読み出しと演算の実行、また第3ステージでは結果の書き込みを主な機能とする。パイプライン処理においては、パイプラインの乱れをできるだけ少なくすることで速度の向上を図る。

パイプラインに乱れを生じるのは一般に、分岐命令を実行した場合及び連続する命令間にデータの依存関係がある場合などに発生する。分岐命令については、分岐命令と分岐先命令の間に挿入命令をいれることが可能な場合には挿入し、またデータ依存のある命令が連続するような場合には、プログラムの実行に影響を与えない範囲で命令の順番

を入れ換えて、データ依存関係のある命令が連続しないようにする。このようにしてパイプラインの乱れを出来るだけ少なくする。

4 システムの適用例と評価

例として Parcor 格子型フィルターのアルゴリズムについて評価を行った結果を示す。ブロック 14(B0014) について見てみると(図2)、このブロック中で S31 と S32、S34 から S37、S39 から S42 とデータに依存関係のある命令が連続している。これを自動的に命令の順序を入れ換えることによって、連続する命令間にデータの依存関係をなくしている(図3)。ブロック 14 の場合ブロックの中だけで元来 23 ステージ(段) かかっていた処理が、16 ステージに減少する。16 ステージはこのブロックの実行に必要な最小限のステージ数であることから、このブロックに連続するデータの依存関係は完全に除去されたことになる。このようにしてプログラム全体にたいして変換を施してやり、各ブロックに対してステージ数をカウントする。また、あるデータに対して解析系でのシミュレートによるブロックの移動情報をもとにこのアルゴリズムが必要とする全ステージ数をカウントして比較したものが表1である。表1を見てわかるように、垂直方式と比較すると、全体では著しくステージが減る。パイプライン処理を行なうと全体で 67,747 ステージに減少し、このシステムの適用によりさらに 18,048 ステージ減少することになる。一つの命令が 3 ステージであり、各命令の各ステージは、全て同時で実行できると仮定していることから、この数値を 3 で割った 6,016 個分の命令を実行する時間だけ速度の向上が期待できることになる。

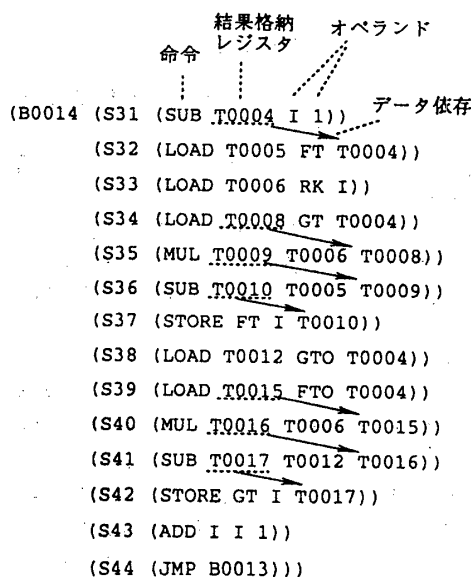


図 2: B0014 の中間情報 (変更前)

データ
依存関係解消

```

(B0014 (S31 (SUB T0004 I 1))
      (S32 (LOAD T0006 RK I))
      (S33 (LOAD T0005 FT T0004))
      (S34 (LOAD T0008 GT T0004))
      (S35 (LOAD T0012 GTO T0004))
      (S36 (LOAD T0015 FTO T0004))
      (S37 (MUL T0009 T0006 T0008))
      (S38 (MUL T0016 T0006 T0015))
      (S39 (SUB T0010 T0005 T0009))
      (S40 (SUB T0017 T0012 T0016))
      (S41 (STORE FT I T0010))
      (S42 (STORE GT I T0017))
      (S43 (ADD I I 1))
      (S44 (JMP B0013)))
  
```

図 3: B0014 の中間情報 (変更後)

表 1: 実行ステージ減少状況

	垂直方式	パイプライン方式	
		変更前	変更後
ブロック 14	42	23	16
全ステージ数	139,545	67,747	49,699

5 むすび

高位の仕様記述から専用プロセッサを設計するシステム(SYARDS)におけるパイプライン処理の考慮について報告した。本システムにより、ユーザはソフトウェアの記述と同じ感覚で最適な専用プロセッサの設計が可能となる。今後は、パイプライン処理のステージ数の制限や命令の時間的な制約などを考慮してより実際的なシステムの実現と評価を試みたい。

参考文献

- [1] 田丸啓吉:「専用プロセッサの現状と動向」, 情報処理, Vol.31, No.4, (1990)
- [2] 池永 剛, 白井克彦:「高級言語によるアルゴリズム記述を入力とする専用プロセッサ設計支援システム」, 情報処理学会論文誌, Vol.32, No.11, (1991)
- [3] 北畠宏信, 白井克彦:「高位仕様記述からの専用プロセッサ設計支援システムにおける機能合成について」, 電子情報通信学会 VLSI 設計技術研究会, Vol.91, VLD91-92, (1991).
- [4] Hironebu KITABATAKE, Katsuhiko SHIRAI:「Functional Design of a Special Purpose Processor Based on High Level Specification Description」, IEICE TRANS FUNDAMENTALS, VOL.E75-A, NO 10, (1992)