2ZB−02

DSPにおける変数の用途を考慮したワークRAM割り付け手法*

松澤 亮一　塩見 彰睦†
静岡大学大学院情報学研究科‡

1　はじめに

近年、通信網の整備による高速化、プロセッサ処理性能の向上、記憶装置の大容量化にともない、動画・画像・音声といったマルチメディアコンテンツが一般的に利用されてきている。また、携帯電話の爆発的な普及と共に、機器の携帯化が進み、移動体通信環境や組み込み用途での、高機能なマルチメディアコンテンツ処理の安価な実現に対する要望が高まっている。

移動体通信環境や組み込み用途では面積、熱容量、消費電力、コスト等のさまざまな制限が課せられる。その制限の中で、高度なマルチメディアコンテンツの処理に不可欠な画像・音声・信号などの圧縮伸長には、主に組み込みのデジタル信号処理専用プロセッサ（DSP）が用いられる。

しかし面積やコストの制限により、DSPの設計では演算器とメモリの間のインタフェースが制約されることがある。マルチメディアコンテンツのような巨大なデータを処理するにあたり、制約のあるインタフェースでも効率良くデータを演算器に送り、処理性能の向上をはかる必要がある。

本研究では、演算器とメモリの間のインタフェースに制約のあるアーキテクチャを持ったDSPを対象とし、処理性能を向上させるために、アプリケーションが利用する変数を適切な作業メモリ(ワークRAM)に割り付ける手法を提案する。

2　目的

本研究は、アプリケーションでの変数の用途を調べ、変数を用途に適したワークRAMに割り付けることを目的とする。適切なワークRAMに変数を割り付けることで、変数をワークRAM間で転送することによる性能低下を防ぐことができる。

本手法では以下の2つのステップで変数の割り付けを行う。

1. **変数の用途の解析：** アプリケーションの動作を解析し、データフローから変数がどの演算に使われているか、第何番目の引数に使われているかを調べる。
2. **変数の割り付け：** 変数の用途とDSPのアーキテクチャの制約をもとに、変数の転送が少なくなるよう、適切なワークRAMへ割り付ける。

3　仮想ＤＳＰ

DSPの対象アーキテクチャとしては、仮想DSPであるKDSPを用いる。KDSPのデータパス部のブロック図を図1に示す。KDSPは2段階の実行ステージを持った5段パイプラインプロセッサである。1段階目の実行ステージではMUL演算器、2段階目の実行ステージではALU演算器による演算をそれぞれ行う。ワークRAMとしてはX-Mem、Y-Memの2種類を持つ。

KDSPの演算器に対する入力は次のような制約を持つ。MUL演算器への入力は第1引数がX-Mem、第2引数がY-Memに固定であり、ALU演算器への入力は第1引数がX-Memに固定され、第2引数はレジスタReg、X-Mem、Y-Memから選択できる。それぞれワークRAMに対する入力はALU演算器からの出力のみであるため、ワークRAM間での変数の移動やコピーも演算器を通して行う必要がある。

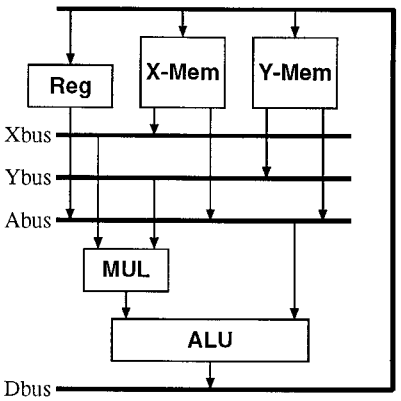

図1: 仮想DSPデータパス部

対象となるアプリケーションは、MPEG2-AACのデコーダである。MPEGデコーダの処理はDSPの処理として一般的であり、積和演算処理が多く含まれてい

*The work RAM allotment technique taking into account of the use of the variables on DSP
†Ryoichi Matsuzawa, Akichika Shiomi
‡Shizuoka University graduate school of information

る [1]。本研究では ISO の MPEG-4 Audio Reference Software [2] を対象とした。

4 変数の用途の解析手法

変数の用途を解析するには、アプリケーションの動作を解析して、データフローを求める必要がある。変数と演算の依存関係を調べるために、アプリケーションを GNU Compiler Collection (GCC) の内部表現である Register Transfer Language (RTL) に変換する。この変換には GCC の RTL ダンプ出力を使用する。出力された RTL を解析し、関数毎の入力とローカル変数、出力に対してデータフローを求める。

さらに、インタフェースによって、変数を演算の第何番目の引数として使用するか、という点で制約を受けることがある。この制約は変数の割り付けに影響する。そこで、可換則が成り立つ演算においても、変数が第何番目の引数として使用されているか、という情報をデータフローに組み込む。このデータフローにより、演算と変数の依存関係を求める。

5 変数の割り付け手法

データフローの解析結果から変数の割り付けを決定する。基本的には入力制約にしたがって、変数を X-Mem と Y-Mem に割り付ける。しかし、変数が全ての演算に関して同一の入力制約を持つわけではなく、またメモリ容量の制約によって割り付けることができない場合がある。その場合、演算前に変数をもう一方のワーク RAM に転送する必要がある。変数を適切なワーク RAM に割り付けるには、この転送が少なくなる割り付けを探索すればよい。

変数 v_i に対して $v_i(X)$ は v_i が X-Mem に割り付けられているものを指すこととする。。また、文献 [1] の手法より基本ブロック b_j の実行回数 num_{b_j} を求める。ここで、変数 $v_i(X)$ が基本ブロック b_j 内で転送される回数を $T(v_i(X), b_j)$ とすると、式 (1) よりアプリケーション実行時に基本ブロック b_j 内で変数 $v_i(X)$ が引き起こす全転送数 $ET(v_i(X), b_j)$ を求めることができる。

$$ET(v_i(X), b_j) = T(v_i(X), b_j) \times num_{b_j} \quad (1)$$

ここで、基本ブロック全体を B とすると、式 (2) より変数 $v_i(X)$ の全体でのエラー数 $VT(v_i(X))$ を求めることができる。

$$VT(v_i(X)) = \sum_{b_j \in B} ET(v_i(X), b_j) \times num_{b_j} \quad (2)$$

同様にして $VT(v_i(Y))$ も求めることができるため、変数 v_i の割り付けを X-Mem から Y-Mem に変更した場合の転送数の変化は $VT(v_i(X)) - VT(v_i(Y))$ で求めることとことができる。つまりこの値が大きな変数 v_i を優先的に Y-Mem に割り付けることで、全体として転送数を少なくすることができる。

ここで、変数 v_i の割り付け先メモリを m_i 、変数全体の集合を V とすると式 (3) より全体としての転送数 AT を求めることができる。

$$AT = \sum_{v_i \in V} \sum_{b_j \in B} ET(v_i(m_i), b_j) \times num_{b_j} \quad (3)$$

KDSP は MUL 命令に特に強い制約があり、MPEG-4 Audio Reference Software は MUL 命令を多用することから、変数を適切なワーク RAM に割り付けることで効率良く演算器へのデータ転送を行うことができ、処理性能の向上が期待できる。

また、X-Mem に割り付けられた変数の集合を V_x とし、変数 v_i のサイズを $sizeof(v_i)$ とすることで、式 (4) より X-Mem に必要な領域 MEM_x を求めることができる。Y-Mem についても同様に MEM_y を求めることができる。

$$MEM_x = \sum_{v_i \in V_x} sizeof(v_i) \quad (4)$$

これら MEM_x, MEM_x がアーキテクチャに規定された制限を越えないように割り付ける必要がある。

6 おわりに

アプリケーションからデータフローを求め、データフローを解析する手法について述べた。また、データフロー解析結果から変数の用途を調べ、KDSP のインタフェースの制約と照らし合わせて、変数の割り付けを決定する手法を提案した。

データの依存性と共に命令のスケジューリングと併せたエラーの見積りをすることで、より適切な変数割り付けを行うことができると考えられる。命令のスケジューリングとの連係は今後の課題である。

謝辞

本研究を行うにあたり討論して頂いた（株）ヤマハの木村繁樹氏、静岡大学塩見研究室の諸氏に感謝いたします。なお、本研究の一部は（株）ヤマハとの共同研究によるものである。

参考文献

[1] 松澤亮一, “システム機能分割のための応用プログラム解析手法の検討”, 卒業論文, 2000.

[2] “MPEG-4 Reference Software (Committee Draft 14496-5)”, ISO/IEC JTC1/SC29/WG11 N1905, 1997.