

1 はじめに

現在、一般に使用されている CPU は、多種多様なアプリケーション（以降 AP）を実行できるように設計されている。したがって、必ずしも使用頻度の高い命令順に高速処理されるような回路構成でない上、APによってはまったく使用しない命令を含んでいる場合もある。そこで、APの実行に際し命令の出現頻度に応じたクロック数の割当てを各命令に対して行えば、さらなる高速化が期待できる。ここでは、APの実行に必要な命令の出現頻度と命令の実行に必要な最小限のクロック数とのトレードオフについての考え方を示し、APに対応した最適な CPU 構成の実現性について考察する。

2 APの命令実行回数と出現頻度の関係

APを高速動作させる1つの方法としては、使用頻度の高い命令をできるだけ少ないクロック数で動作させることが考えられる。そのことは、APのホットスポットを分析することで、各APが実行される際にどの関数が多く繰返し実行され、どの命令が数多く実行されているのかが分かり、同時にそれらの命令を実行するのに必要なクロック数と比較することで高速動作のボトルネックを抽出することができる。

このような観点で、APを解析する場合、その解析対象としては、OSの実行を含めた全体的な場合と、OSの実行を含めないAP固有の実行のみを考慮した場合とが考えられるが、現在のAPの多くはOS上で実行させることが通常であることから、今回の解析については、前者を対象とした。今回対象としたAPは通信系、画像処理系、解凍圧縮系の3つであり、それらの解析の結果から、以下のことが判明した。

1. APにより、使用される関数の種類、出現頻度が異なる。(図1)
2. APにより、使用される命令の種類、出現頻度が異なる。(図2)
3. 命令の出現頻度とクロック数とは必ずしも相補関係にない。(図3)

APが実行される際、使用される関数の多くは、他の関数を互いに参照し連携してAPを動作させていた。ただし、実際に使用された関数は、特定の関数に集中していた。出現頻度の多い関数の上位1%が使用された関数全体の60%の回数を占めていた。さらに、これら上位1%の関数で使用された命令の種類、出現頻度の分布が、実行された関数全体の命令の種類、出現頻度の分布とほぼ同じであることが分かった。

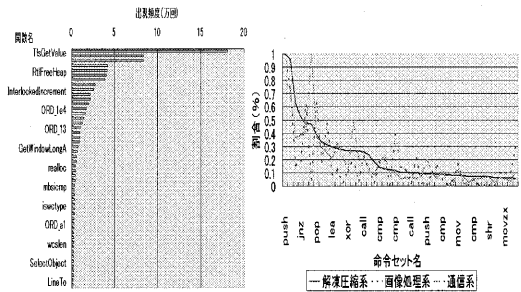


図1 APの実行に必要な関数の出現頻度

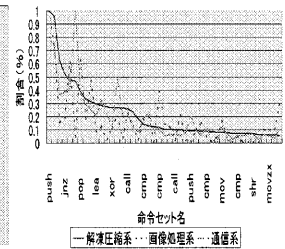


図2 APによる命令の出現頻度

APを高速動作させるためには、出現頻度の高い関数を高速に動作させる必要があるが、今回の検証結果から、APの実行に使用された命令の出現頻度の分布と命令実行に必要なクロック数とが多くの場合に一致していないことがわかった。このことから、現時点の汎用CPUでは、APが最適な環境で動作しているとは言い切れず、改善の余地があると言える。

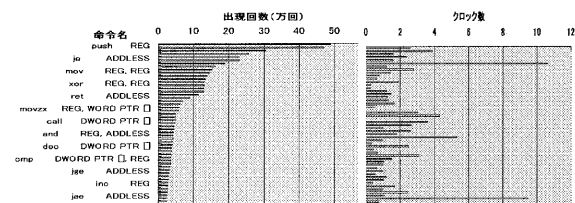


図3 命令の出現頻度と実行に必要なクロック数

3 AP対応型CPU

出現頻度の高い命令を高速（少クロック）に動かすことにより、APをより高速に動作させることができる。このことから本稿では、APを実行させる際に多く出現する命令のクロック数をできるだけ少なくするように、APに対応して最適なCPU構成を採る方式を提案するものである。

一方で、特定の命令の高速化を計るあまりに回路規模の巨大化や、他の命令との組合せの結果として高速化が図れない場合も考えられる。そのため、全体に与える影響を最小限にして、最大の効果が得られるような調整が必要となる。

CPUの実行時間に要する時間は下記の式で求めることができる。

$$\text{実行時間} = (\text{実行に要する総命令} \times \text{CPI}) / (\text{クロック周波数}) [1]$$

したがって、「実行に要する総命令×CPI」の値を少なくするために、APの実行許容時間を総命令実行回数で割った値を目安値として設定する(図4)。この目安値を大きく下回る命令は、そのクロック数を

命令名	0	50	100	150	200	250
push REG						
je ADDRESS						
mov REG, REG						
xor REG, REG						
ret ADDRESS						
movzx REG, WORD PTR []						
call DWORD PTR []						
and REG, ADDRESS						
dec DWORD PTR []						
cmp DWORD PTR [], REG						
jge ADDRESS						
inc REG						

4 FPGAを用いたCPU構成

現在の再構成可能素子は最大200MHzまで動作可能で集積度も高く、システムゲート数は50,000~1,000,000であり、汎用システムとも容易に接続できるべくP C I仕様と完全に準拠している。以前の68000CPUが68000トランジスタ、約2万ゲート規模であることを考えれば、CPUだけでなく周辺機器としてA Pに特化した専用回路さえも1チップに収めることが可能である。またその設計においても、現在デジタル回路設計の主流となっているハードウェア記述言語(HDL)を利用することで、回路の変更にも柔軟に対応できる。

HDLで記述したデータベース幅が8, 16, 32ビットを持つCPU回路構成をXilinx社のXCV300に実装し、標準の命令セットを搭載した場合とAPに対応した命令セットを持つ場合の回路規模について検証を行った。その結果、命令セットを19種削除した場合、このCPUに限れば平均4割の回路規模の減少が見られた(図5)。次に、XCV50を使った回路のベンチマークテストを参照すれば、ロジック使用率が低ければ低いほど、最大内部クロック周波数は高速に動作することが示された。(図6) 必要最小限の回路構成にすることは高速動作実現には不可欠であることが示された。

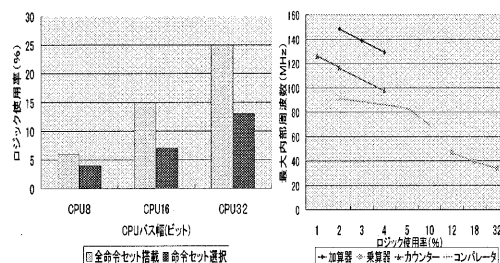


図5 CPUの回路規模

[illegible]

図7 アプリケーション対応型CPU構成

5 まとめ

このことを受け、本稿では、APを高速動作させるために、その実行に必要な関数の頻度順に高速動作させる方式を提案した。つまり使用されている命令の出現頻度の高いものに対してクロック数の少ない割当てを可能にするダイナミックなCPU構成法を示した。それには、汎用CPUよりも回路構成を柔軟に変更できるFPGAなどの再構成可能素子を利用することが有効である。

実行するAPに合わせてCPU構成を最適化できれば、物理的には1つのCPUでありながら、CPUの内部構成を柔軟に変えることができ、同時にAPにとっては必要最小限の機能を有する専用のCPU構成となる。つまりハードウェアとソフトウェアの境界をなくして、APにとって最適な環境上での実行が可能になる。

参考文献

- [1] 神保進一:“最新マイクロプロセッサテクノロジー”
日経バイト, pp288-309 (1999).