

6G-5

## 組込みマイコン用最適化コンパイラにおける レジスタ割付けの方式

田中 旭<sup>1</sup> 入交 旬子<sup>1</sup> 富永 宣輝<sup>2</sup>  
松下電子工業（株）<sup>1</sup> 松下電器産業（株）<sup>2</sup>

### 1. はじめに

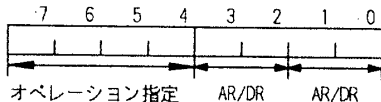
近年、マイコン制御の機器に組込まれるソフトウェアのサイズが大規模化しており、高級言語による開発が要望されている。しかし、高級言語による開発では、オブジェクトコードサイズの増大により、コストアップになることが組み込み分野では大きな問題となる。

そこで我々は、マイコンの設計方針として、組み込みマイコンに要請されるオブジェクトコードの高効率性と、大規模な組み込みソフトウェアを十分に処理可能な高速実行性を両立するために、コンパイラを含めたマイコン設計を行っている。

本稿では、この設計方針に基づいた組み込みマイコンMN10200<sup>[2]</sup>のレジスタ構成、MN10200用コンパイラにおける変数へのレジスタ割付け方式と、評価について報告する。

### 2. MN10200のレジスタ構成

C言語プログラムのオブジェクトコードサイズを圧縮するために、機械語命令のレジスタ指



【図1】基本命令のビット割付け

アドレスレジスタ	データレジスタ
AR0	DR0
AR1	DR1
AR2	DR2
AR3(SP)	DR3

【図2】MN10200のレジスタ構成

定フィールドに着目し、レジスタ間演算や転送命令等のコンパイラが生成する頻度の高い基本命令を1バイト化することを試みた。8本レジスタをデータ用とアドレス用に機能分離して各

4本構成にすると共に、レジスタ指定フィールドを2ビットにした。（図1、2）

### 3. 従来のレジスタ割付け方式

従来の大域的レジスタ割付け方式としては、グラフカラーリング方式<sup>[1]</sup>が知られている。この方式は、使用可能なレジスタのうち、変数への割付け専用レジスタに適用可能なものであり、関数呼び出しで使用される引数や戻り値用レジスタ、関数呼び出しの前後で値が保証されない破壊レジスタなど、予め用途が確定している特定用途レジスタを含めた大域的なレジスタ割付け方式としては不十分である。

そこで我々は、

- (1) 特定用途レジスタも変数への大域的な割付け用レジスタとして使用可能とし、
- (2) アドレス、データレジスタの相互割付けが可能な方式<sup>[3]</sup>

を考案した。ここでは、特に(1)について述べる。

### 4. MN10200におけるレジスタ割り付け方式

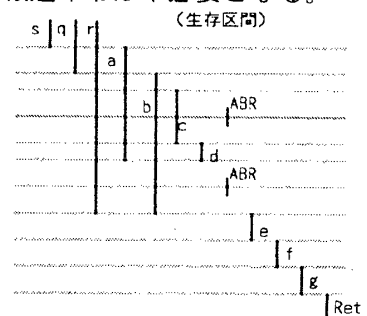
#### 4-1 考察

図3はプログラムと変数が保持している値の有効範囲（生存区間）を表わすものである。変数e, fのように生存区間の終わり始まりが一致するものを「終始一致関係」にあると定義する。変数Retとgは終始一致関係にあり、同じレジスタに割付けると転送命令は不必要となる。

(プログラム)

```

1: a = *s + 100;
2: b = *q - 100;
3: c = *r + 200;
4: f1(c);
5: d = c / b;
6: if(a == d)
7:     f2(b, *r);
8: e = b - *r;
9: f = e;
10: g = f * 15;
11: Ret = g;
    
```



ABR: 引数用と破壊レジスタ用のレジスタ(DR0, DR1)を割付けている便宜上の変数。  
Ret: 戻り値用のレジスタ(DR0)を割付けている便宜上の変数。

【図3】プログラムと生存区間の例

また、コンパイラのターゲットとするマイコンが2オペランド形式であり、例えば図の10番目の式の変数fが演算結果が格納されるデストネーションオペランドになるときは、変数f, gのように終始一致関係にある変数は、同じレジスタに割り付けると転送命令を削減できる。

さらに、変数Ret, fのように間接的に終始一致関係(間接終始一致関係と定義する)にあるものも同じレジスタに割り付けると、全体的に転送命令を減少させることが可能となる。

#### 4-2 局所・大域割付有効度による割り付け方式

我々は以上の考察を踏まえて、プログラム中の変数xに、各レジスタを割り付けた時に、生成コードのサイズ縮小度と実行速度の増加度である、割付有効度を推定する方法をとった。このとき、変数xと終始一致関係にある変数yに割り付けているレジスタと、変数xが使用されている演算で使用している他の変数の割り付けレジスタを元に、割付有効度を算出した局所割付有効度を求め、さらに、間接一致関係にある変数zに割り付けているレジスタを元に、割付有効度を算出した大域割付有効度から、最も割付有効度の高いレジスタをxに割り付ける。

以下は具体的なレジスタ割り付けのアルゴリズムである。

- s1. 変数の使用頻度や生存区間の長さ等から変数の割付優先度を求める。
- s2. 未割付けでかつ割付優先度が最大の変数xがなくなるまで、以下のs3からs8までを行う。
- s3. xと生存区間が重なる変数に割り付けていないレジスタの集合Reg1を求める。
- s4. Reg1に属する全てのレジスタr1について以下の局所割付有効度を求める。
  - 1) 演算のデストネーションとなる終始一致変数yに、r1を割り付けていたなら、yの優先度分だけr1の局所割付有効度を上げる。
  - 2) xが非可換演算の結果となっており、デストネーションとならない変数zに割り付けているレジスタがr1のとき、局所割付有効度をzの優先度分だけ下げる。逆にxが非可換演算のデストネーションとならない変数であるとき、演算結果である変数zに割り付けているレジスタがr1のとき局所割付有効度をzの優先度分だけ下げる。
- s5. 求めた局所割付有効度が最大のレジスタの集合をReg2とする。

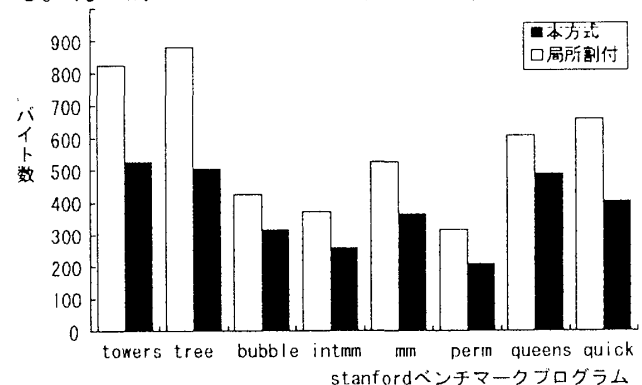
s6. Reg2の要素が1個のときは、そのレジスタをxに割り付けs2へ戻る。複数個の場合はs7, s8を行う。

s7. Reg2に属する全てのレジスタr2について以下の大域割付有効度を求める。

- 1) 演算のデストネーションとなる間接終始一致変数yに、r2を割り付けていたなら、yの優先度分だけr2の大域割付有効度を上げる。
  - 2) 1)と同様の変数yと生存区間が重なる変数にr2を割り付けていたならyの優先度分だけrの大域割付有効度を下げる。
- s8. 求めた大域割付有効度が最大のレジスタをxに割り付け、s2へ戻る。

### 5. 評価

図4は、この方式と、一般的なラベリング方式<sup>[1]</sup>による局所的なレジスタ割り付けを行った場合との生成コードサイズの比較を示すものである。約30%、コードサイズが縮小できた。



【図4】生成コードサイズの比較

### 6. 結び

今後、レジスタ割り付け方式とレジスタ特性(本数、機能)との関係を定量的に評価し、よりオブジェクト効率の高いマイコンシステムの構築を行っていく予定である。

#### 【参考文献】

- [1] A. Aho, R. Sethi, J. Ulman, "Compilers Principles, Techniques, and Tools", Addison Wesley, 1988
- [2] 桧垣他, "機器組込み用小型16ビットマイコンMN10200のパイプライン制御方式" 電情信秋季大会, D-57, 1992
- [3] 田中他, "最適化コンパイラにおけるレジスタ割り付けの一方式" 情処秋季大会, Oct. 1991