

### 3 J-1 省コードサイズマイコンに適用したコンパイラの開発

山本和司 富永宣輝 高山秀一 漆原誠一 春名修介 松下電器産業株式会社 情報通信研究所

#### 1 はじめに

最近の組み込みマイコンのプログラムサイズは増加の一途をたどり、アドレス空間64Kバイト以上を必要とする状況が発生している。このような場合、16ビットマイコンではなく、32ビットマイコンを使うのが一般的だが、これには次の3つの問題点がある。

第1に、外部変数を扱うときや関数呼び出しの際、変数や飛び先のアドレスを32ビットで扱うため、命令語長が長くなる。その結果コードサイズが大きくなり、コードサイズがコスト(必要なメモリ量)に影響する組み込み応用には不利である。

第2に、チップサイズ、消費電力が増加する。

第3に、現在、多くの組み込みマイコンのプログラムサイズは、32ビットマイコンで使用できるアドレス空間のたかだか数万分の一から数百分の一の数百K~数Mバイトにすぎない。また、扱うデータ長は、32ビットではなく16ビットですむ用途が多い。

そこで、我々は、アドレス演算、データ演算ともに24ビットで行い、24ビットのアドレスと、16ビットのデータを扱える最適資源型マイコン[1]とそのコンパイラを開発した。本論では、この最適資源型マイコンに適用したコンパイラの実現方法と、その比較モデルとなるマイコンとの評価を述べる。

#### 2 最適資源型マイコンのアーキテクチャ

ここでは最適資源型マイコンと比較モデルとなる32ビットマイコンのアーキテクチャについて述べる。

##### ● 最適資源型マイコン

我々の開発したマイコンは24ビットのデータレジスタ、アドレスレジスタをそれぞれ4本を持つ。

C言語のint型データは16ビットで扱い、アド

Development of compiler applied the resources-saving microcomputer

Kazushi Yamamoto Nobuki Tominaga

Syuichi Takayama Seiichi Urushiwara

Syusuke Haruna

Matsushita Electric Industrial Co.,Ltd

レスのデータは24ビットで扱う。

メモリ上のint型データはロードストア時に16ビット $\Rightarrow$ 24ビットのデータサイズ変換が行われ、メモリ上のアドレスデータはそのままのサイズ24ビットでロードストアされる。またデータレジスタ上の下位16ビットを24ビットに拡張することもできる。

データレジスタはデータ演算用に汎用的に使用されるレジスタで、汎用的な演算命令が用意されている。アドレスレジスタは、変数の値を参照する時やアドレス計算等に使用されるレジスタで、備えているアドレッシングモード(レジスタ直接、即値(24bit)、絶対(24bit)、レジスタ間接、レジスタ相対間接、インデックスレジスタ間接)のうち後者3つのアドレッシングモードを用いて間接参照する。また、アドレス計算用の演算命令(加減算、比較)が用意されている。

##### ● 32ビットモデルのマイコン

32ビットモデルのマイコンは最適資源型マイコンと比べて、レジスタと、即値、絶対アドレッシングのビット長が32ビットであり、レジスタメモリ間転送でデータサイズ変換がない点のみ異なる。

#### 3 C言語コンパイラの実現方法

ここでは、2章で述べたアーキテクチャに基づいたC言語コンパイラの実現方法を具体的なプログラムの例を用いて説明する。

図1はC言語で記述された、配列の代入式のプログラム例であり、図2は図1の配列の代入式に対してコンパイラが内部的に生成する解析木である。

図2の解析木では添字の型は整数型(以下(int)とする)、演算子+の型はポインタ型(以下(pointer)とする)、シフト演算子の型は、(int)である。

このとき32ビットモデルマイコンのコンパイラでは(int)、(pointer)は4バイトであり、次のような演算過程を経る。

- i(int)  $\rightarrow$  DR(int)
- DR(int) を2左シフト(int)

- aのアドレス値→AR (pointer)
- @(DR,AR) を用いて値3を格納  
ここでDR, ARはそれぞれデータレジスタ、アドレスレジスタ、→は転送である。

これに対し、最適資源型マイコンに図2の解析木を適用すると(int)は2バイト、(pointer)は3バイトとなり、演算過程は、

- i(int)→DR(int)
- DR(int)を1左シフト (int)
- DR(int) =>DR (pointer)
- aのアドレス値→AR (pointer)
- @(DR,AR)を用いて値3を格納

となる。ここで=>は拡張を意味する。

このとき左シフトしてから拡張されるために次の2点の問題点がある。

- アドレスの計算を24ビットで行う必要あるにもかかわらず値を16ビットしか保証していないために64Kバイト以上の配列領域を扱えない。
- (int)から(pointer)への拡張命令が冗長となる。

そこで、最適資源型マイコンのコンパイラが生成する解析木ではポインタと同じ長さの3バイト整数型(以下(int3)とする)を用意し、シフト演算子の型にこれを付加することで対応した。

この時、演算過程は次のようになる。

- i(int) =>DR (int3)
- DR(int3) を1左シフト(int3)
- aのアドレス値→AR (pointer)
- @(DR,AR) を用いて値3を格納

これにより、上記2点の問題が解決できた。

#### 4 評価

ここでは、3章で述べたC言語コンパイラの実現方法を適用した時の最適資源型マイコンマイコンと比較モデルのマイコンとの評価を行なう。

評価は、Dhrystone(Ver2.1)のプログラムのコードサイズについて行った。表1はこれらのマイコンの比較の表である。

これを見ると、最適資源型マイコンは、32ビットモデルと比べると、コードサイズが約8%減少してい

る。これは次の2点のためであると考えられる。

- アドレス計算を24ビットで行うために命令語長が短くなる。
- オフセットの型をポインタと同じ長さの3バイト整数型で生成するためアドレスの計算を効率良く行うことができる。

#### 5 まとめ

最適資源型マイコンに適用したコンパイラの開発により、コストパフォーマンスの良い、最適資源型マイコンの性能を引き出すことができ、32ビットマイコンに比べてコードサイズが約8%減少した。

```
int a[100];
int i;
f1()
{
    a[i] = 3;
}
```

図1 C言語でのプログラム例

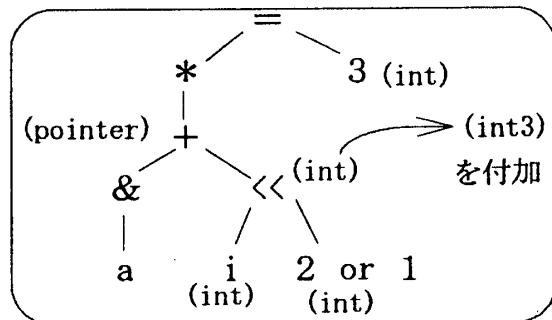


図2 図1に対する解析木

	コードサイズ (byte)	割合 (%)
32ビットモデル	1858	100.0
最適資源型マイコン	1703	91.7

表1 コードサイズ比較

参考文献[1]鈴木他：「カスタマイズアドレス空間マイコンのアーキテクチャ」電子情報通信学会コンピュータシステム研究会, November 16, 1993