RX62Nマイコンを100%使いこなすために RXファミリ用C/C++ コンパイラ活用技法~最適化編~

鹿取 祐二

ルネサス エレクトロニクス製RXファミリ用C/C++コンパイラは、コンパイル時にさまざまな最適化を行うことによって効率の高いコードを出力する。これにより、速度を高速化したりコード・サイズを最小にすることができるが、実現するには適切なコンパイル・オプションを設定する必要がある。その具体的な方法と最適化結果を見ていこう。
(編集部)

1. 外部変数アクセス

ここでは外部変数の操作に対するルネサス エレクトロニクス製RXファミリ用C/C++コンパイラ(以下、ルネサス製コンパイラと呼ぶ)の最適化オプションを紹介します。

● 外部変数アクセスの問題点

RXファミリに限らず、CPUは外部変数(大域変数)の操作が苦手です。その理由はC言語の文法上、外部変数は必ずメモリに割り付けられるからです。アクセスするためには目的の変数のアドレス情報が命令中のどこかに必要となります。RXファミリは4Gバイトのメモリ空間をサポートしているので、アドレス情報は32ビットです。結果、外部変数をアクセスするたびに32ビットのアドレス情報を汎用レジスタにロードしなければなりません。

リスト1のソースをコンパイルした結果がリスト2(a)です。小文字のrの部分外部変数のアドレス情報です。非常に長いことが理解できると思います。

これを削減するにはプログラマ自らが外部変数を構造体化する必要があります。構造体にすれば、構造体の先頭アドレスだけを汎用レジスタにロードし、先頭以外のメンバは先頭アドレスからのオフセットで操作することにより、

リスト1 外部変数アクセスのソース

```
1: short a, b;
2: int c;
3: extern short x, y;
4: extern int z;
5:
6: void main(void)
7: {
8:     c = a + b;
9:     z = x + y;
10: }
```

リスト2
外部変数アクセスのコンパイル
結果

アドレスのロードの回数を削減できます。

● 「外部変数アクセス最適化」オプション

外部変数の構造体化をプログラマに代わって行うのが「外部変数アクセス最適化」オプションです。これを使えば外部変数の宣言を変更する必要はなく、ルネサス製コンパイラが自動的に外部変数を構造体化し、アドレス・ロードの回数を削減します。

目的のオプションは図1に示すコンパイラのタブ、カテゴリは最適化、外部変数アクセス最適化のドロップ・ダウン・メニューで指定します。なお、設定には「モジュール内」と「モジュール間」の二つがあり、「モジュール間」の場合は図2に示すダイアログが表示されますが、かまわず[OK]を押してください。このダイアログの意味は後述します。

● 「外部変数アクセス最適化」の「モジュール内」の効果

「外部変数アクセス最適化」オプションを「モジュール内」に設定した場合、ソース・ファイル内に変数定義のある外部変数の構造体化を行います。

リスト2(b)では、変数a, b, cが構造体化の対象であり、externにより外部参照している変数x, y, zは対象外となります。また、デフォルトでは「変数の配置」オプショ

1:	FB42rrrrrrrr	_main:	MOV.L	#_a,R4
2:	FB32rrrrrrrr		MOV.L	#_b,R3
3:	DC45		MOV.W	[R4],R5
4:	DC34		MOV.W	[R3],R4
5:	FB32rrrrrrrr		MOV.L	#_y,R3
6:	4B54		ADD	R5,R4
7:	FB52rrrrrrrr		MOV.L	#_c,R5
8:	E354		MOV.L	R4,[R5]
9:	FB42rrrrrrrr		MOV.L	#_x,R4
10:	DC45		MOV.W	[R4],R5
11:	DC34		MOV.W	[R3],R4
12:	4B54		ADD	R5,R4
13:	FB52rrrrrrrr		MOV.L	#_z,R5
14:	E354		MOV.L	R4,[R5]
15:	02		RTS	

(a) 最適化しない場合