

第1章

ブート・ローダの理解に欠かせない

CPU 14のコモンセンス

柳川 誠介

本章では、CPUのデータシートを読む際のポイントについて述べます。データシートは1,000ページ近くになることがあります。これを最初から1ページずつ読んでいく必要はありません。また、その時間もありません。かといって、問題が起きてからここを読んでおけばよかったと後悔するようでは開発の効率が上がリません。

ハードウェアの設計にはデータシートが欠かせませんが、コンパイラを選択する際にも読んでおかなければならない箇所があります。コンパイラとCPUの間に相性の問題があるからです。高級言語はCPUに依存しないのが建前ですが、アドレス空間の問題からハードウェアに絡む問題まで、実際にはCPUごとに解決しなければならない問題がたくさんあります。

ここではCPU別の細かな仕様に触れるのではなく、データシートを読んでいないと、後々どのようなトラブルが発生するのかについて紹介します。本章が「転ばぬ先のつえ」となれば幸いです。

どのCPUのデータシートでも、初めの部分に概要が書かれています。どのCPUでも解説されている内容は似たようなものなので、読み飛ばすこともあるかもしれませんが、しかしこれは基礎となるところですから、確認が必要なところはしっかりと読みましょう。

① 機能が増えてきたマイコン

「マイコン」という呼び方が広く一般的に使われていますが、炊飯器などに組み込まれている制御用のものは「マイクロコントローラ」、パソコンのCPUは「マイクロコンピュータ」と呼ぶのがより正しいのかもしれませんが、初めて世に出たのは1971年にIntel社が電卓向けに開発した4004で、これは「マイクロプロセッサ」と呼ばれました。

その後ICの集積度が上がり、基板上で接続されていたインターフェース回路などをCPUチップの中に

取り込むようになりました。そういう経緯があるので、同一チップ上にありながら、タイマやシリアル入出力などの部分は「ペリフェラル(peripheral；周辺という意味)」と呼ばれています。

図1に4004の内部ブロック図を示します。コンピュータとして動作させるにはクロック発生回路やROMなどを接続する必要があります。歴史的には、ここに含まれていない回路はすべてペリフェラルと見なすのが正しいでしょう。図2のルネサス テクノロジのH8マイクロコンピュータ(1994年に発表)の内部ブロック図と比較してみてください。H8のブロック図で「CPU」と書かれている部分が4004を拡張した機能を持っています。ペリフェラルはもとよりROMもRAMも内蔵しているので、このチップ単独でコンピュータとして動作します(いわゆる1チップ・モード)。内蔵のROMは128Kバイト、RAMは4Kバイトの容量しかないので、より大きなシステムに用いるときは外部メモリ接続用バスを外部に引き出す拡張モードを使います。

② データ幅によって一括処理できる量が変わる

4004の内部で扱うデータは4ビットでした。電卓用には4ビットあれば、BCD(Binary Coded Decimal)によって0～9の数を表すことができるからです。データは4本のラインでやり取りされます。この本数(バス幅)によって「何ビットのCPUである」というのが普通です。今日のパソコンでは、64ビットや128ビットのCPUが使われています。一度のデータのやり取りで、ビット数にして4004の16倍、32倍の量を一括処理できるようになったわけです。

内部のデータ・バスの幅とレジスタのビット幅やアドレス数は異なるのが普通です。H8シリーズは16ビットのマイクロコンピュータですが、レジスタのデータ幅は32ビットあります(図3)。このため16

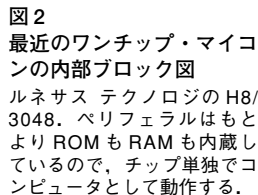
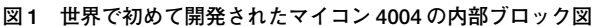


図3 16ビットCPUでも32ビットの内部レジスタを持っている例

Intel社の86系やルネサス テクノロジーのH8シリーズでは、図のように分割使用できる。一方、RISCアーキテクチャであるルネサス テクノロジーのSHシリーズやFreescale Semiconductor社のPowerPCでは分割使用できない。このように、CPUによって構成が異なる。

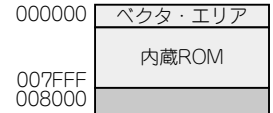
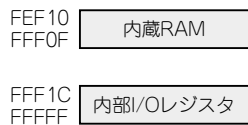
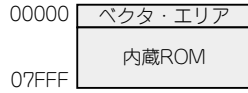
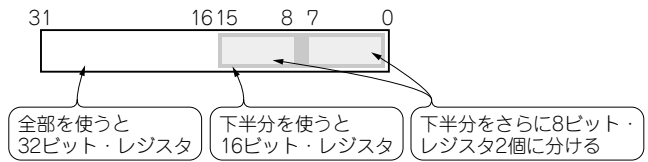


図4
H8/3048のメモリ・マップの例
全部で5通りのメモリ・マップがある。

(a) シングル・チップ、アドバンスド・モード

(b) 内蔵ROM有効拡張、1Mバイト・モード

(c) 内蔵ROM有効拡張、16Mバイト・モード

ビット数と16ビット数のかけ算の結果を一つのレジスタに収めることができます。

③ メモリ・マップを参考に型を選択

アドレス空間はCPUのプログラム・カウンタのビット幅で決まります。同じ16ビットのCPUでもアドレス空間は同じではありません。さらに、初期設定でアドレス空間の広さを決めることができるタイプもあります。

図4にH8シリーズのメモリ・マップの例を示します。プログラム・カウンタの幅が16ビットならば、アドレス空間は64Kバイトです。扱う変数がすべてこの領域に収まれば、C言語におけるポインタは符号なしの16ビットの数で済みます。このような型を採用することでCPUの演算が速くなり、コード・サイズが小さくなるのであれば、これに越したことはありません。CPUとコンパイラの可能性を見計らって、効率が良く、拡張性の高い方法を選びます。

④ メモリ空間の中に存在する特別な領域

CPUごとに異なる特別な番地の筆頭として挙げられるのは、パワーONしてまず実行する番地です。リセット・ベクタやブート番地と呼ばれています。0番地のものが大勢を占めますが、86系のようにそうでないものもあります。本来、コンパイラ本体では絶対

番地は指定できません。初期化プログラムが用意されている場合は、どのようなしくみになっているかを把握しておくべきでしょう。

ブート番地の次に重要なのは割り込みベクタです。これは、割り込みによって実行されるルーチンの先頭アドレスのテーブルです。内蔵の割り込みコントローラにより任意の場所に配置できるタイプと、ブート番地とともに決まった場所に配置されているタイプがあります。いずれにせよ絶対番地の指定が必要で、割り込みルーチンの先頭番地を示す値が書き込まれます。

絶対番地の指定が必要な領域はまだあります。スタック・エリアです。普通、RAMエリアの上限に配置します。メインのプログラムが起動した直後はスタック・ポインタはこの上限に近いところを指していますが、サブルーチンと呼ぶごとに、あるいはルーチン内で変数(auto変数)を宣言するごとに、ポインタは若い番地へ移動していきます。一瞬のことですが、割り込み発生時も同じです。これを「スタックが深くなる」と表現することもあります(図5)。

スタックが用意されたエリアをはみ出して深くなると、通常の変数を上書きしてしまったり、サブルーチンの戻り番地が変数の操作で変わったりして、プログラムが暴走します。スタック・エリアがどれほど必要

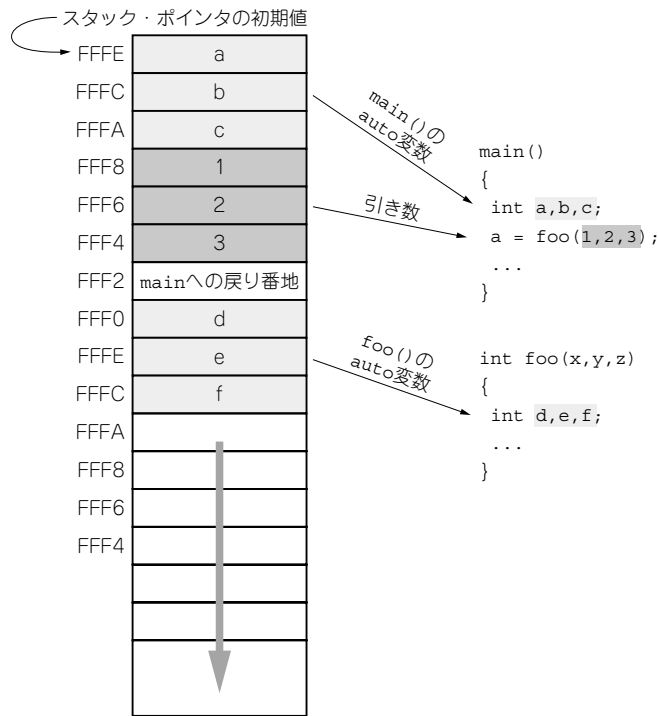


図5 変数の宣言や関数呼び出しによってスタックは深くなっていく
具体的な割り当て方法は処理系によってまちまちである。

かは、プログラムの書き方に依存します。極端な場合、再帰呼び出しを行う関数を実行し続けると、スタックが深くなります。

スタックの状態はICE (In-circuit Emulator) などのデバッガがあれば分かりますが、そのような機材を使わなくても、スタック・ポインタの値を変数としてどこかに保存するルーチンを用意すると、スタック・エリアの内容を表示できるようにするなどの方法によって知ることができます。スタックに余裕があることが確かめられたら、スタイルが大幅に変わらない限り、その経験は後のプログラミングでも生かれます。

ペリフェラルは86系ではメモリとは別の空間に配置されますが、そのほかのCPUではメモリと同じ空間に配置されます(いわゆるメモリ・マップドI/O)。この領域は、大半が8ビットまたは16ビットの変数として扱われます。

C言語では、8ビットの変数の場合はchar、16ビットの場合はint(コンパイラによっては32ビットになることもある。そのときの16ビットの指定はshort)と宣言します。ただし、符号なしでないと演算中に符号拡張するようなコードをコンパイラが生成するなど、思わぬトラブルを招くことがあります。デ

フォルトが符号付きになっているコンパイラもあるので、unsigned(符号なしの指定)も付けておいた方が無難です。

もし、charと宣言すべきところをintまたはlongと宣言するとどうなるかという、上位バイト(unsignedで256未満なら0)が隣の番地に書き込まれてしまいます。コード効率や処理速度を考えると16ビットのCPUが得意な演算は16ビットの数なのですが、処理の途中はともかく、結果がどう表されるかも意識しておくことが大切です。

⑤ フラッシュROMとEEPROMを使い分ける

フラッシュROMもEEPROMも、どちらも書き換え可能なROMです。データをブロックごとに消去するフラッシュROMにはプログラムを、バイト単位で書き換えられるEEPROMには製品情報などを保存します。開発時にはフラッシュROM搭載のマイコンを使い、そこに書き込んだプログラムにバグがないことを確かめたら量産に向けたマスクROMバージョンのマイコンに乗り換えられるように、マイコン製品のラインナップが構成されています。

最近では量産機にもフラッシュROMを用いる場合があります。ユーザの手元でプログラムを更新できる