第3章 CPUと処理系ごとに変わる…整数の加減算と乗算，浮動小数点数演算も

オーバフロー時の挙動と確認方法

永井 健一

CPUの演算器やアドレス・バスは，特定のビット幅を持っています注1．基本的にこのビット幅によって扱える数の最大値が決まります．

メモリのアドレスも最大値の制約を受けるので，CPUのビット幅によって扱える最大メモリ量が決まります．例えば，32ビットでは約4Gバイトです．現在では，搭載メモリが4Gバイトを超えるコンピュータも増えており，Armの一部のCPUなどは64ビットでしか動作しないものも出てきています．

CPUが持つ演算器のビット幅は有限なので，無限桁数の計算はできません．大きな数を扱うとオーバフローやアンダフローが発生する可能性があります．

プログラミングの処理系や言語によっても異なりますが，C言語で低レイヤ・プログラミングする場合などは，それらに対処する必要があります．本章では，このときのCPUの振る舞いと，C言語での取り扱いについて説明します．

処理系ごとに異なるオーバフロー時の挙動

● C言語ではオーバフロー時の動作は未定義

C言語の整数には，符号付き整数(int)と符号なし整数(unsigned int)があります．C言語の規約上は，符号付き整数のオーバフローは未定義という扱いです．つまり，計算結果は実装依存であり，ハードウェアによって異なります．従って，特定のハードウェアに依存する計算結果を期待してC言語のプログラムを書くと，ポータビリティ(移植性)を損ないます．

実装依存とは言え，世の中にあるほとんどのCPUは，符号付き整数は最上位ビットが符号ビットであり，負の数は2の補数で表現されています．

符号なし整数は，そのビット幅によるWrap Around(そのビット幅の剰余)として定義されています．これはCPUのハードウェア的にはオーバフローした桁を無視しているだけです(**図1**)．

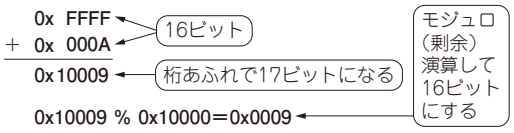

図1　符号なし整数がオーバフローするときの挙動(16ビットの場合)

● ステータス・レジスタで何が起きたか分かる

CPUには，演算用のオペランドを置くレジスタとは別に，CPUの動作を制御するために使うステータス・レジスタがあります．演算命令実行時に，計算結果に応じたフラグがセットされるレジスタです．分岐命令などは，そのフラグの値を見て，条件分岐するかどうかを決定しています．

オーバフロー時のCPUの挙動を知るには，ステータス・レジスタを見る必要があります．例えばArmの場合，CPSRというレジスタに情報が格納されます．フォーマットを**図2**に示します．

注1：Armやx86/x64のように，32ビットと64ビット・モードを切り替えて動作するCPUも多くあります．

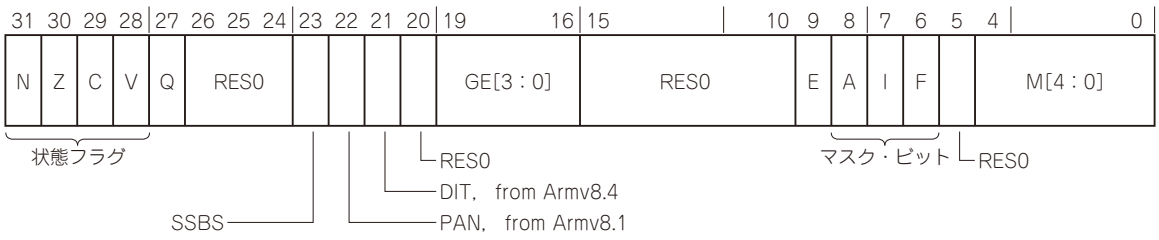

図2(1)　**Armのステータス・レジスタ(CPSR)の構成**