

第8章 ローカル処理中のレジスタ書き換えがトラブルの元！

ステップ解説！ ハマリポイント③…割り込み

永原 柁

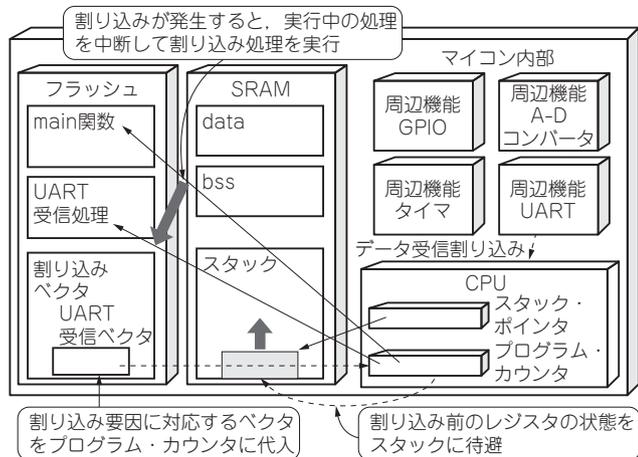


図1 割り込み発生時のマイコン内部の動作
割り込みが発生したとき、マイコン内部ではこのような動作は行われている

マイコンのプログラムでは、周辺機能を使用するために、割り込みを使えることがほぼ必須です。割り込みを使うことで、例えば通信を行いながら演算処理を行うといった、複数の処理を並行して進めることができます。一方、割り込みはトラブルを起こしやすい技術でもあり、マイコン内でどのような処理が行われているかを知っていると、トラブル対応が容易になると思います。

ここでは割り込みの概要をおさらいした後、ARMマイコン Stellaris LM4F120H5QR（以下、Stellarisマイコン）の内部動作について説明します（図1）。

おさらい…割り込み処理のキホン

関数呼び出しでは、呼び出し側はいつ関数を呼び出すかを知っています。しかし割り込みでは、いつ割り込みが発生するかわかりません。この点が大きく異なります。

図2に、第6章の代入文のプログラムを実行中に、割り込みが発生した例を示します。この例では、レジスタR0が示す領域にレジスタR1の値を格納しようとした直前に割り込みが発生したと想定しています。

割り込み処理も普通のプログラムなので、CPUのレジスタを使って実行します。割り込み処理が終わって、元の代入処理に復帰するとき、もしレジスタの値が割り込み処理で書き換えられたままの状態だと、予想できない結果になります。

図2ではレジスタR0の値が書き換えられた、と想定しています。レジスタR0は値を格納する領域を示すので、割り込み処理で書き換えられた値を、領域のアドレスとみなして、レジスタR1の値を書き込んでしまいます。

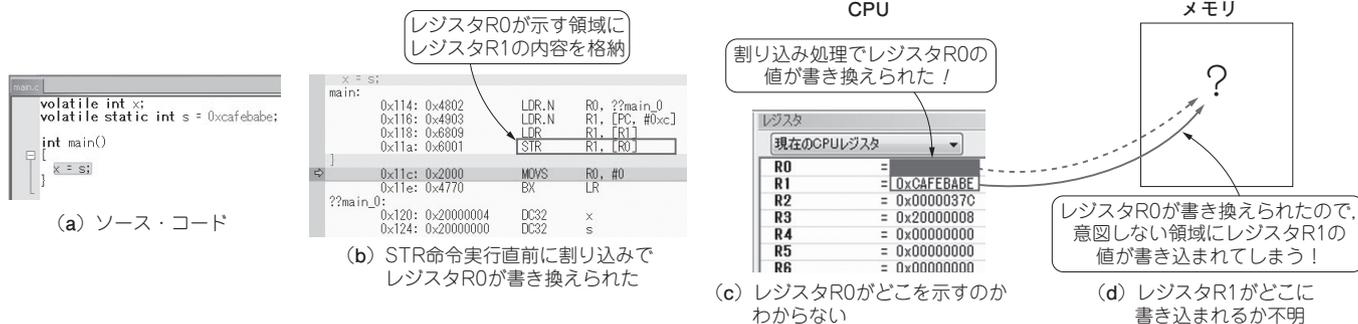


図2 割り込み処理でレジスタを書き換えると、代入さえ危険な処理になってしまう
代入直前に割り込みでレジスタR0が書き換えられた例。レジスタR0が示す意図しない領域にレジスタR1の値を書き込んでしまう