

作る MiniOS の CPU 動作と仕様

菅原 政義

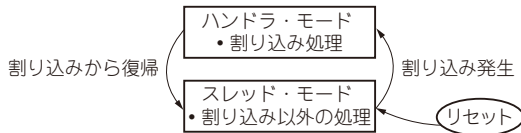


図1 Cortex-M4の動作モード

表1 動作モードと特権レベル

動作モード	CONTROLレジスタの設定	特権レベル
ハンドラ・モード	—	特権(変更できない)
スレッド・モード	nPRIV = 0	特権
スレッド・モード	nPRIV = 1	非特権

表2 動作モードとスタック・ポインタ

動作モード	CONTROLレジスタの設定	スタック・ポインタ
ハンドラ・モード	—	MSP(変更できない)
スレッド・モード	SPSEL = 0	MSP
スレッド・モード	SPSEL = 1	PSP

● CPUの動作モード

▶動作モード

MiniOSで使用するCortex-M4の機能について説明します。

Cortex-M4には、ハンドラ・モードとスレッド・モードという2つの動作モードがあります(図1)。ハンドラ・モードは、割り込み中の動作モードで、割り込み処理(および例外処理)を行います。スレッド・モードは、割り込み以外の動作モードで、割り込み以外の通常の処理を行います。

CPUがリセットされると、スレッド・モードで起動します。割り込みが発生すると、自動的にプロセッサによってハンドラ・モードに切り替わります。割り込み処理から復帰する際に、元の動作モードに戻ります。

▶特権レベル

Cortex-M4には、動作モードとは別に、特権レベルという設定があります。特権レベルでは、特別なレジスタ(CONTROLレジスタなど)の操作が可能ですが、

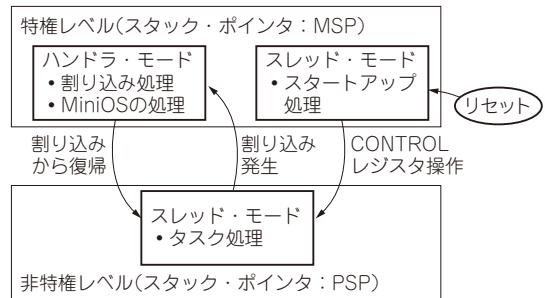


図2 MiniOSの動作モード、特権レベル、スタック・ポインタ

非特権レベルではそれらのレジスタを操作することができません。ハンドラ・モードは常に特権レベルで動作し、スレッド・モードはCONTROLレジスタの設定により特権レベルを指定できます(表1)。

▶スタック・ポインタ

Cortex-M4には、MSP(Main Stack Pointer)とPSP(Process Stack Pointer)という2つのスタック・ポインタが用意されています。ハンドラ・モードでは、常にMSPを使用します。スレッド・モードでは、CONTROLレジスタの設定によりMSPとPSPのどちらを使用するか指定できます(表2)。

● MiniOSでの仕様

MiniOSでは、上記を図2に示す仕様とします。

▶目的

OSとタスクで使用するスタックを分けることで、OS側のスタックがタスク側から意図せず壊されるのを防ぎます。タスクを非特権レベルとすることで、タスクから特別なレジスタへのアクセスを制限し、システムを保護します。

▶仕様

OSの処理はハンドラ・モード、特権レベル、MSPで動作し、タスクの処理はスレッド・モード、非特権レベル、PSPで動作します。

CPUリセット時は、Cortex-M4の仕様でスレッド・モード、特権レベル、MSPで起動するので、スター