

第4章

PIO専用アセンブラを使ってPWMパルス生成とUART通信

噂のプログラマブルI/Oはこう使う

中森 章

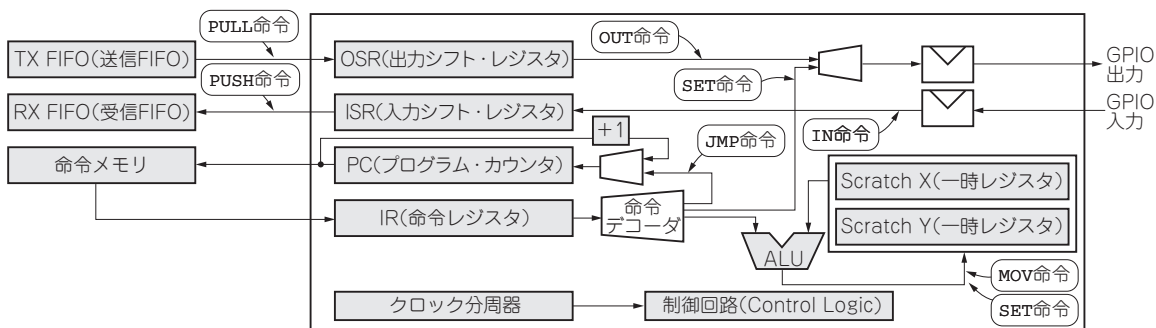


図1 (1) PIOを構成するステート・マシン(SM)ブロック図
プログラム・カウンタやControl LogicなどCPUコアに準ずる機能を有する。なお、図1は文献(1) RP2040 Datasheetの図と解説から筆者が制作した図です

● PIOは小さなCPUコア

プログラマブルI/O (PIO) はラズベリー・パイ Pico (以降、Pico) の特徴の一つです。本書11ページの図1と本書12ページの図2で示したように、PIO0とPIO1という2種類のGPIOを独立に操作する4つのステート・マシン(SM)から構成されます。それぞれのステート・マシンは並列に動作することが可能で、PIO0、PIO1ごとに用意された32ワードの命令メモリから、ステート・マシン独自の命令を、フェッチし、デコードし、実行します。これは、まさにCPUコアの動作です。大げさに例えるなら「Picoは10個[2×(Cortex-M0+) + 2×(4×ステート・マシン)]のCPUコアを持っている」と言っても過言ではありません。

表1 (1) プログラマブルI/O (PIO) を制御するステート・マシン(SM)が実行できる命令一覧

ビット	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
JMP	0	0	0	遅延/ サイドセット	Condition			Address									
WAIT	0	0	1		Pol			Source			Index						
IN	0	1	0		Source						Bit count						
OUT	0	1	1		Destination						Bit count						
PUSH	1	0	0		0	IFF	Blk	0	0	0	0	0	0	0	0	0	
PULL	1	0	0		1	IFE	Blk	0	0	0	0	0	0	0	0	0	
MOV	1	0	1		Destination			Op	Source								
IRQ	1	1	0		0	Clr	Wait	Index									
SET	1	1	1		Destination						Data						

図1にプログラマブルI/Oを構成するステート・マシンのブロック図を示します。プログラマブルI/Oのブロック図はp.16の図2を参照してください。プログラム・カウンタ、制御回路、一時レジスタX、一時レジスタYなどを見るとCPUコアに準じる機能を持っていることが分かります。

PIO0、PIO1ごとに、それぞれ、32ワードの命令メモリしかありませんので、小型CPUコアとも言える各ステート・マシンは、あまり複雑な処理ができません。できるのは、GPIOをプログラムされた間隔で“H”に駆動したり“L”に駆動したりすることです。また、GPIOから入力されるシリアルな信号を一時メモリに格納することくらいです。現実的に、SPIやI²CやUARTの動作を行わせるためにはCPUコアの助けが必要です。しかし、各ステート・マシンはある程度自律して動作するので、実際のCPUコアの行うべき手間を削減できます。

● トライすること

もっとも、一定周期の矩形波やPWM波形はCPUコアの助けを借りなくても、32ワード内の命令メモリに格納されたプログラムだけで実行可能です。ここでは、初めに、CPUコアの補助なしで、自律して動作するプログラマブルI/Oのプログラム例を示し、最後にCPUコアの補助で動くPIO版の「Hello World!」