

# リファレンス・ボードで ハードウェアを知ろう(基礎編)

—DIPスイッチの値を読み取り、LEDを制御する

岩橋正実

プログラムの開発は仕様書に基づいて行いますが、仕様書だけに頼るのではなく、回路図などのハードウェアそのものの情報から、ソフトウェア実装に必要な仕様を読み取ることが重要です。ここでは実践例として、ルネサス テクノロジのSH-3プロセッサ(SH7727)を搭載したリファレンス・ボードを使い、OSを利用せずにハードウェアを制御する典型的な組み込みプログラムを開発します。(編集部)

本稿では、CPUのリファレンス・ボード(評価用ボード)を使い、OSを利用せずにハードウェアを制御するソフトウェアを開発します。まず基礎編として、DIPスイッチの情報を取得し、それをLEDに表示するプログラムを作成します。

注1: ROM( read only memory )は読み取り専用の記憶装置(メモリ)であり、RAM( random access memory )は読み書きができる記憶装置である。本稿で使用したリファレンス・ボードは、ROMとしてEPROM(erasable programmable read only memory)とフラッシュ・メモリを、RAMとしてSRAM( static random access memory )とSDRAM( synchronous dynamic random access memory )を搭載している。

ハードウェアの動作原理を理解して、メカをソフトウェアで制御する感動を少しでも感じていただければと思います。そして、すべての仕様書には基準となる事実情報があり、それを明確にすることの重要性も感じ取っていただければ幸いです。

## 1 開発の準備

実際の開発に入る前に、知っておいていただきたい前提知識について簡単に説明します。

### ● CPUの動きを理解しよう

#### 1) CPUの基本アーキテクチャ

まず、一般的なCPUの基本アーキテクチャについて解説します。CPUは、アドレス・バスとデータ・バスによって外部周辺デバイス(ROM, RAM<sup>注1</sup>など)と接続されています。CPUはアドレス・バスを使ってアクセスする周辺デバイスを指定し、データ・バスを使って外部周辺デバイスに

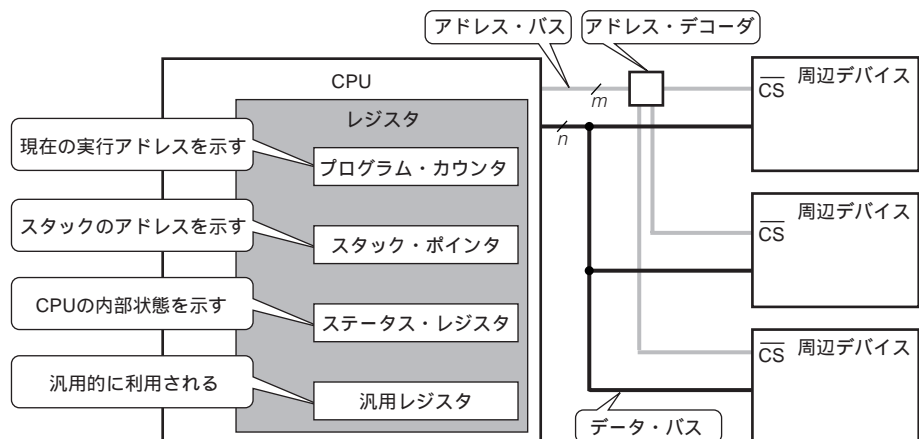
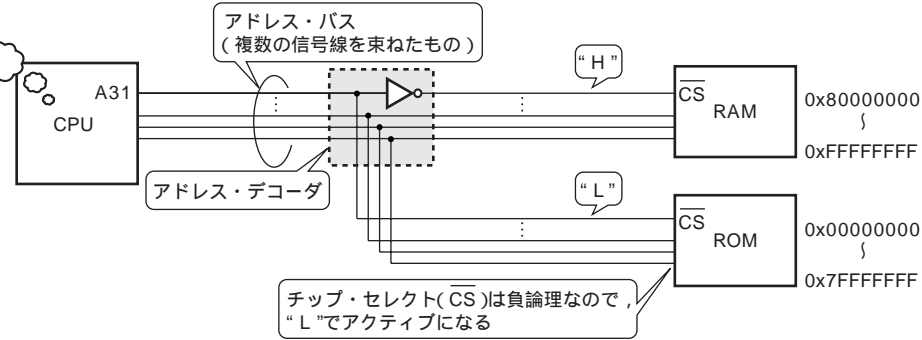


図1 CPUはアドレス・バスとデータ・バスによって外部周辺デバイスと接続されている  
例えば、CPUがデバイスAにデータを書き込みたい場合、アドレス・バスにデバイスAのアドレスを出力し、データ・バスに書き込みたいデータを出力する。アドレス・デコーダがアドレスを解釈し、デバイスAのチップ・セレクトをアクティブにすることによって、デバイスAがアクティブになり、CPUがデバイスAにアクセスできるようになる。

0x00000000番地にアクセスしたい!  
(A31~A0とも'0'. アドレス線は  
正論理なので" L "が出力される)

図2  
チップ・セレクトをアクティブにしてデバ  
イスを選択する

CPUが出力したA31が'0'ならROMが,'1'  
ならRAMが選択される. ここでは,CPUがア  
ドレス・バスに0x00000000を乗せた場合を示  
している(この例ではA31がアドレス・バスの  
最上位のけたに,A0がアドレス・バスの最下  
位のけたに相当する).



データを書き込んだり(Write),読み出したり(Read)しま  
す(図1).

また,CPUは内部にレジスタと呼ばれる記憶領域を持っ  
ています.非常に高速にアクセスでき,演算などに利用す  
る領域です.内部状態を示すステータス・レジスタ,現在  
の実行アドレスを示すプログラム・カウンタ,スタックの  
アドレスを示すスタック・ポインタ,汎用的に利用される  
汎用レジスタなどから構成されます.

アクセスするデバイスを選択するには,デバイスのチッ  
プ・セレクト(CS)信号をアクティブ(有効)にします.例  
えばアドレス・バスが32本の信号線で構成されているCPU  
で,アドレス0x00000000~0x7FFFFFFFをROMに,ア  
ドレス0x80000000~0xFFFFFFFFをRAMに割り当てる  
場合を考えます.アドレス・バスの信号線(A31~A0)の  
うち,A31の信号が'0'("L")<sup>注2</sup>のとき,ROMデバイスの  
チップ・セレクトがアクティブ(ON)になって<sup>注3</sup>,ROMデ  
バイ스에アクセスできます.また,A31の信号が'1'("H")  
のとき,RAMデバイスのチップ・セレクトがアクティブに  
なって,RAMデバイスにアクセスできます(図2).なお,信  
号や論理回路についてなじみのない方は,p.65のコラム「簡  
単な論理回路を作ってみよう!」を参照していただきたい).

また,CPUは,書き込みの対象となるデバイスとは書き  
込み(Write)信号線で,読み出しの対象となるデバイスと  
は読み出し(Read)信号線で接続されています.CPUがデ  
バイスに対して読み書きを行う際には,アドレス・バスと  
データ・バス以外に,Write信号線もしくはRead信号線  
を利用します.

CPU上で動作するソフトウェアから見ると,外部周辺デ  
バイスや内蔵する制御回路(アナログ情報を入出力する  
A-DコンバータやD-Aコンバータ,割り込みコントローラ,  
タイマ,シリアル・コントローラ,USBコントローラ,

LCDコントローラなど)はすべてメモリ空間内にマッピ  
ングされています<sup>注4</sup>.CPUはプログラム・カウンタが指し  
しているアドレス(メモリ空間の中の番地)に書かれた命令  
を読み,それに従って動作します.このように,メモリ空  
間内の特定のアドレスにデータを入出力することにより,  
周辺デバイスとのやりとりを実現しています.

起動アーキテクチャはCPUによって異なりますが,一般  
的な例を図3に示します.CPUは電源リセット時(つまり,  
CPUの動作開始時に)プログラム・カウンタに0x00000000  
を設定して,アドレス・バスに0x00000000を出力し(今回  
使用するSH7727の場合はプログラム・カウンタに  
0xA0000000を設定するが,32ビットのアドレスのうち上  
位3ビットは無視されるので<sup>注5</sup>,アドレス・バスに出力す  
るのはやはり0x00000000番地となる),Read信号をアク  
ティブ("L")にします.ROMのチップ・セレクトがアクテ  
ィブになり,ROMがデータ・バス上にデータを出力しま  
す.CPUはROMからデータを読み出し,Read信号を"H"  
に戻します.

CPUは読み出したデータ(命令)を解析し,命令の種類  
に応じて処理を行います.例えば,メモリにデータを書き

注2: 論理回路の信号は'1'と'0'で表現する.正論理の場合は'1'が"H"  
レベルに,'0'が"L"レベルに対応し,負論理の場合は'0'が"H"  
に,'1'が"L"に対応する."H"と"L"は特定の電位をしきい値として  
決定する.信号の反転とは,"H"を"L"に,"L"を"H"にすることを  
意味する.

注3: チップ・セレクトは,負論理で設計されていることが多いので,"L"  
でアクティブ(有効)になる.なお,信号名の上の横線は,その信号が  
負論理であることを示している.

注4: メモリ空間とは仮想的な記憶領域である.8ビット(1バイト)単位に区  
切られており,区切りごとに番地(アドレス)が振られている.外部周  
辺デバイスや制御回路がどのアドレスにマッピングされているかを示  
す資料が「メモリ・マップ」である.

注5: SH7727はMMU(memory management unit;メモリ管理ユニット)  
機能を持っている.MMU機能を利用すれば,物理アドレスよりも大  
きなメモリ空間(仮想メモリ空間)を確保することができる.SH7727  
のアドレスの上位3ビットはMMU機能を利用する場合のみ有効な情報  
であり,今回はMMU機能を利用していない(設定をOFFにしている).