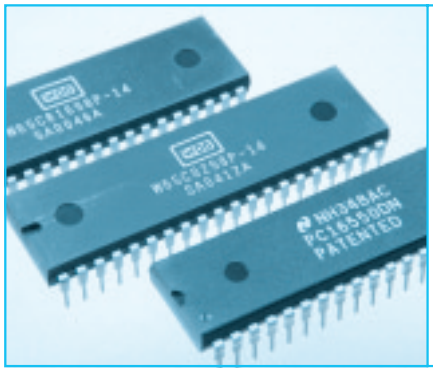

マイコン・システムのしくみを基礎から理解する

6502マイコン・ボード製作記

〈第6回〉6502の信号とバス・タイミング(後編)

～制御信号の役割と詳細～

桑野　雅彦
Masahiko Kuwano

前回(第5回，2006年9月号)は，A_0～A_{15}，BE，D_0～D_7の各信号について解説しました．今回は，そのほかの信号とバス・タイミングについて詳しく解説します．

CPUの信号とその役割 ②

● IRQB(Interrupt Request)

マスク可能割り込み要求信号です．IRQBがアサート("L")されたとき，プロセッサ・ステータス・レジスタ(Pレジスタ：フラグ・レジスタ)のIビット(ビット2)が'0'になっていれば割り込みが受け付けられます．'1'になっていると割り込み要求は受け付けられず，'0'になった時点で受け付けられます．

▶ 6502の割り込み処理

6502は割り込みを受け付けると，プログラム・カウンタの値(割り込み復帰後に実行すべき命令のあるアドレス)の上位8ビット，下位8ビット，Pレジスタをこの順でスタックに待避します．続いてIビットを'1'にセットして立て続けに割り込みが入ってしまうことを防ぎます．

この後，割り込みベクタ領域($FFFE,$FFFF)に書かれている割り込み処理ルーチン(マニュアルでは「割り込みハンドラ」)の先頭アドレスを読み出してジ

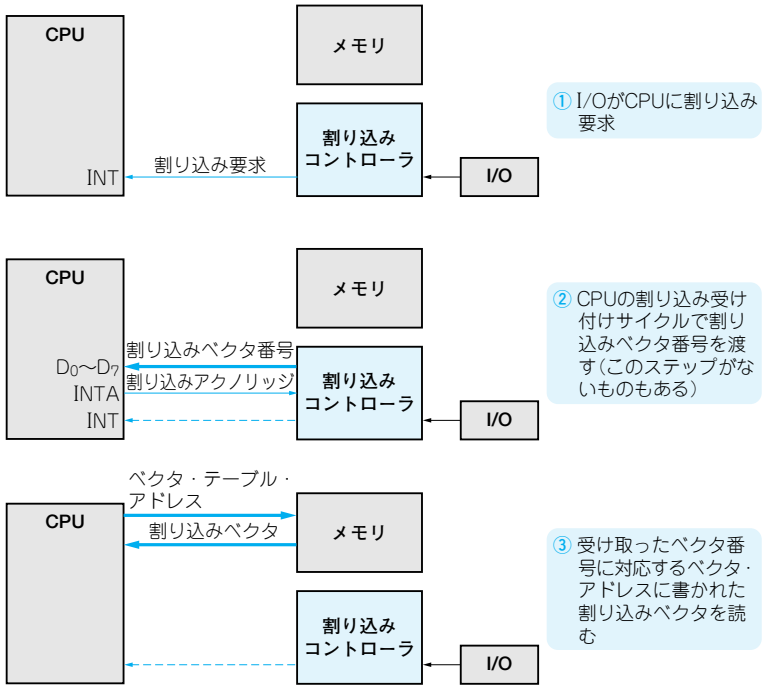

図6－1
割り込みベクタ番号を持つCPUの割り込み応答動作
6502は②のステップがない

Keywords
IRQB，MLB，NMIB，PHI2，PHI2O，PHI1O，RWB，RDY，RESB，SOB，SYNC，VPB，割り込みのネスティング，リード・モディファイ・ライト・サイクル，マルチプロセッサ・システム，ウェイト，セットアップ・タイム，ホールド・タイム

ャンプし，割り込み処理を開始します．このときのバスの動きは，VPB信号がアサートされる以外は通常のメモリ・リード・サイクルと同様です．

IRQBは，割り込み処理ルーチンのなかで，割り込み要因がクリアされるまでアサートしておきます．

割り込み処理の最後では，RTI（ReTurn from Interrupt）命令を実行します．この命令では，スタックからPレジスタ，プログラム・カウンタの下位8ビット，上位8ビットを順次取り出し，元のプログラムの実行を再開します．

割り込みマスク用のフラグ・ビットであるIビットは，割り込み処理に入ったときに自動的にセットされますが，これをCLI（CLear Interrupt disable bit）命令などでクリアすると，新たな割り込み要求を受け付けられるようになります．このように，割り込み処理のなかでさらに割り込みを受け付けることを割り込みのネスティングと言います．

▶割り込みベクタを持つCPUの割り込み処理

6502の場合，IRQBに対応する割り込みベクタは一つだけです．しかし，数多くの割り込み要因に配慮したCPUでは，割り込みを受け付けたときに，対応する割り込みベクタ番号を受け取るようになっています．なお，複数の割り込み要因のどれを先に受け付けるかという判定やベクタ番号の生成などは少々面倒なため，割り込みコントローラという専用ICが用意されています（現在はマイコン内部に組み込まれることが多い）．

図6－1に，割り込みベクタ番号をもつCPUの割り込み応答動作を示します．

① I/Oから割り込み要求が発生すると，割り込みコントローラはCPUに対して割り込み要求を行う
② CPUは割り込みを受け付けると，バス上に割り込み応答サイクルを生成する（ここではINTA信号で通知するように書いている）．割り込みコントローラは割り込み応答サイクルを検出すると，割り込みベクタ番号をデータ・バス上に出力し，CPUがこれを読む
③ CPUは割り込みベクタ番号に対応したメモリ上の割り込みベクタ・テーブルをアクセスする．CPUはスタックにフラグや戻り先の番地などを待避し，割り込み禁止状態にしてからベクタ・テーブルに書かれていた割り込み処理ルーチンのアドレスに制御を移す

なお，8080の割り込み応答サイクルは，3バイトの命令フェッチ・サイクルになっていました．このため，8080用の割り込みコントローラ8259（8259Aを8080モードで動かしたときも同じ）では，割り込み応答サイクルでCD XX XX（CDはCALL命令，XX XXはアドレス情報）という3バイトのコードをCPUに送るという仕様になっていました．

現在ではこのような，命令コードをフェッチする，というものはまれで，ここで説明したようなベクタ番号を読むという方法か，6502と同様に直接割り込みベクタ領域を読むという方法が使われています．

6502の場合には②の段階がありませんが，ベクタ領域をリードしているということはVPB信号を見ているとわかります．もし，割り込み要求元に応じた先に分岐させたいときは，このサイクルではメモリ・アクセスをさせずに，I/O側からベクタ・アドレスを出力したり下位の何ビットかだけをI/Oのデータとすり替えるという方法も考えられます．

今回は，I/Oは16550一つだけなので，このような細工はせず，単純にIRQBに16550の割り込み要求信号端子を接続しています．

● MLB（Memory Lock）

メモリ・ロック信号です．

▶リード・モディファイ・ライト・サイクル

CPUのメモリ・アクセスのなかには，1命令の処理のなかでメモリのデータを読み出して変更したものを書き戻すという，リード・モディファイ・ライト・サイクルを実行するものがあります．

6502の場合には，例えばSMB0（Set Memory Bit 0）命令などがこれに相当します．SMB0は指定されたアドレスのビット0を‘1’にする命令ですが，CPUとメモリの間は8ビット単位でのデータ転送しかできないので，CPUはいったん8ビットぶんを読み取って，ビット0を‘1’にしたデータを作成しメモリに書き込むという動作を行います（**図6－2**）．

▶マルチプロセッサ・システムで生じる問題

CPUが一つであればよいのですが，マルチプロセッサ・システムなどで，複数のCPUが同一のメモリ領域をアクセスするようなときには問題が生じます．

図6－3のように，CPUが二つあり，CPU#1側で，

```
SMB0 $80
```

CPU#2側で，

```
SMB1 $80
```

を実行したとします．メモリの初期値は$00であったとしましょう．このときCPU#1とCPU#2による命令実行が交互に行われていれば，メモリの内容はビット0とビット1の両方がセットされて$03になるはずです．

ところが，図のようにちょうどタイミングが一致して，リード・モディファイ・ライト・サイクルの間に相手のアクセスが挟まったとしたらどうなるでしょう．

①CPU#2がメモリの内容を読む（$00が読める）