

## 31.3. メモリマップド I/O

入出力機器の制御を行う場合において、メモリのリード/ライトのための CPU 命令にて入出力機器の制御を行う手法を「メモリマップド I/O」と呼びます。(入出力用の特別な命令があり、その命令を使用する手法を「ポートマップド I/O」と呼びます)

メモリマップド I/O は、アドレス空間上にメモリと入出力機器が共存するようにマッピングされます。

入出力機器は CPU のバスを監視し、自身のレジスタがマッピングされた空間への CPU からのアクセスにตอบสนองします。

### 特徴

メモリマップド I/O は、入出力を特別扱いしないため、CPU の内部回路が簡略化され、高速化や低価格化が容易です。

マイクロプロセッサの規模が 16 ビットから 32 ビット、さらには 64 ビットとなるに従って、メモリマップド I/O のためにアドレス空間を用意することはほとんど問題にならなくなってきました。例えば、16 ビットの場合はアドレス空間が 64K( $2^{16}$ ) バイト、32 ビットの場合は 4G( $2^{32}$ ) バイトとなるためです。

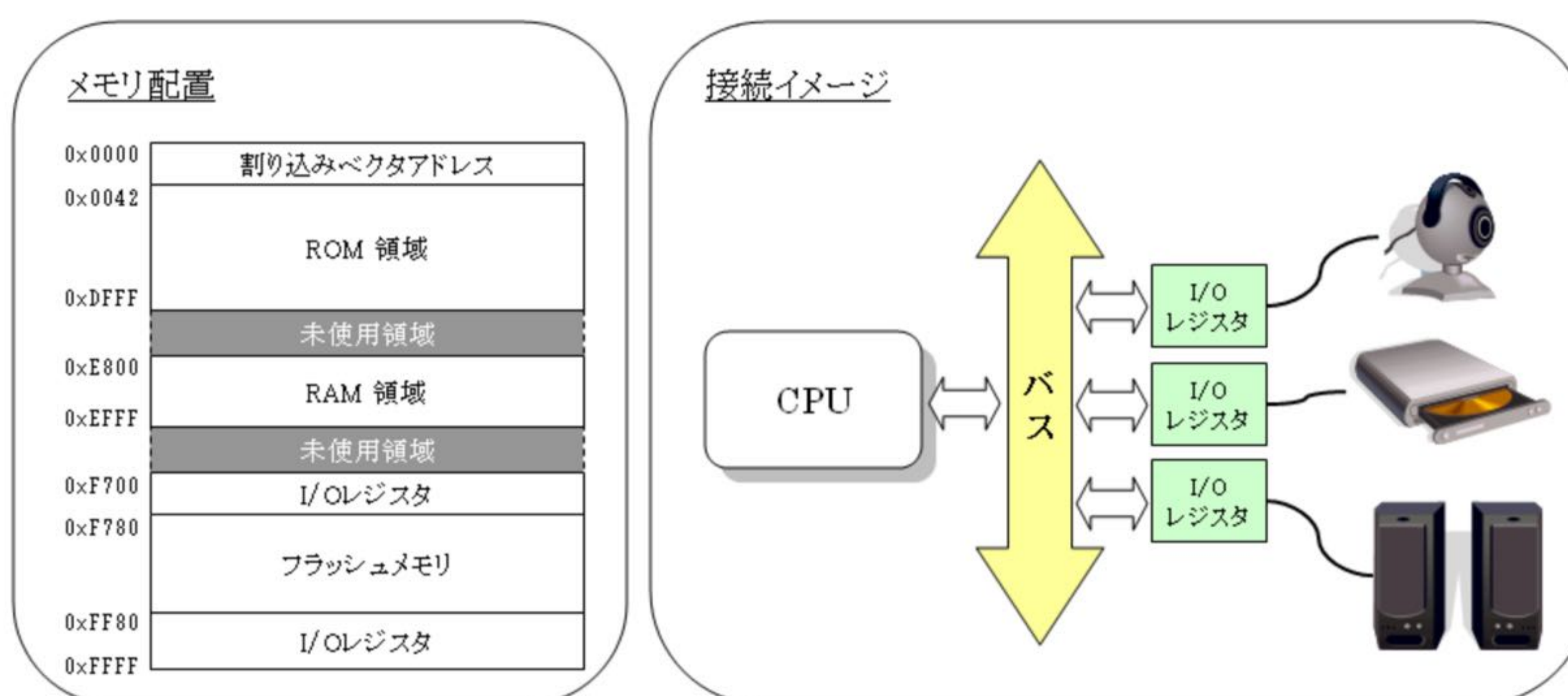
### 注意点

メモリマップド I/O はメモリと入出力機器が同じアドレス空間上に並存しますが、全く同じに扱うことはできません。

例えば、キャッシュメモリ(ライトバック方式)を使用しているシステムでは、読み書きをしてもバス上にそれが出て行かないことがあるため、I/O アクセスはキャッシュを通して行うことはできません。

### メモリ配置

以下にメモリマップド I/O のメモリ配置と接続イメージの例を示します。



メモリマップド I/O では、メモリ上に I/O レジスタが配置されています。プログラムで I/O レジスタのアドレスを参照し、レジスタの値を参照・変更することができます。入出力機器は I/O レジスタと接続し、バス経由で CPU と接続されます。I/O レジスタの状態を入出力機器は監視し、レジスタの状態に応じた動作をします。

メモリ上に配置されている要素のいくつかを説明します。

### I/O レジスタ

入出力機器を制御するためのレジスタがマッピングされています。I/O レジスタの値を読み書きすることで入出力機器の制御を行います。

### 割り込みベクタアドレス

割り込み要因ごとのジャンプ先アドレスを定義しています。(割り込みベクタテーブル)

割り込み発生時に呼び出される割り込みハンドラのアドレスを格納しています。(割り込みについては後の節で解説します)

### ROM 領域

ROM は、読み込み専用のメモリで、プログラムや定数データを格納します。一般的に使用している ROM には、以下の種類があります。

- マスク ROM  
製造工程で書き込んだ後、内容の変更ができない
- フラッシュ ROM  
特定の回路操作をすることで書き込みや内容の更新が可能となる  
ただし、書き込み回数に制限がある

ROM の読み込みの速度は、通常 RAM よりも遅いです。実行速度向上のためにスタートアップルーチンの中で、ROM から RAM 領域にプログラムをコピーし、RAM 上のプログラムを実行するといったテクニックがよく用いられます。(スタートアップルーチンについては後の節で解説します)

### RAM 領域

RAM は、読み書き可能なメモリです。一般的に使用している RAM には、以下の種類があります。

- ダイナミック RAM (DRAM)  
安価に大容量を実現できるが、放置しておくと記憶内容が消えてしまうため、定期的に記憶内容の書き直し(リフレッシュ)を必要とする
- スタティック RAM (SRAM)  
同じ記憶容量だと DRAM よりも高価だが、低消費電流/高速アクセスで、リフレッシュを必要としない