■6 群（コンピュータ･基礎理論とハードウェア）- 4 編（コンピュータアーキテクチャ（I）標準的）

# 6 章 入出力

**【本章の構成】**

本章では，I/O（6-1 節），について述べる．

なお，I/O インタフェース（6-2 節）については，8 群 1 編「センシングとインタラクション」，3 群「コンピュータネットワーク」，及び，4 群 4 編「無線 LAN，無線アクセス，近距離ワイヤレス」を参照して頂きたい．

■6 群 - 4 編 - 6 章

# 6-1 I/O

（執筆者：佐藤　充）[2018 年 12 月 受領]

## 6-1-1 I/O とメモリマップ I/O

計算機システムを構成する要素として入出力デバイスは欠かせないものである．チューリングマシンとしての計算機の構成要素は演算装置とメモリ（主記憶）があれば十分であるが，計算機システムを実世界に適用しようとすると，実行の命令や結果の出力など，外界とのやり取りが必要不可欠となる．この外界とのやり取りをプログラムから制御することを「入出力（Input/Output）」あるいは「I/O」と呼ぶ．I/O には例えば以下のようなものが挙げられる．

実行するプログラムの指示
命令列あるいはデータの記憶装置（テープ，ハードディスクなど）からの読み込み
演算結果の出力
外部装置（モータなど）の制御
キーボードからのキー入力
ディスプレイモニタの表示

プログラムから I/O を実行する方法は様々であり，各プロセッサの実装に依存する．例えば Intel 社の 8086 シリーズでは，専用の I/O 命令があり（IN 命令/OUT 命令），I/O 専用に割り当てられた「I/O ポート」と情報のやり取りができる[1)]．これらの命令を使うことで，プロセッサの外部から 1 あるいは 0 の「ビット」の形で状態を読み込む，あるいはプロセッサから外部にビットとして情報を出力することができる．I/O ポートは複数存在するため，IN 命令や OUT 命令では，どのポートとデータのやり取りを行うかを別途指定する．IN 命令によって取り込まれた情報はプロセッサ内部のデータと等価であり，プロセッサ内部の演算に利用することができる．また，内部での演算結果であるデータは OUT 命令によって外部に出力することができるので，演算結果をプロセッサ外に通知することが可能となる．このような IN 命令，OUT 命令を命令列に組み込むことで，プログラム内部からプロセッサの外部とデータのやり取りを行うことができるようになる．

IN 命令と OUT 命令は I/O 専用命令であるが，プロセッサが実際に行っていることは I/O ポートを指定してのデータの読み込みと書き出しである．I/O ポートの指定は番号で行うので，I/O ポート番号をアドレスとみなすと，プロセッサの動作はメモリの読み書きと等価である．この性質を利用して，メモリの読み書き動作で I/O を行う方式がメモリマップ I/O である．

メモリマップ I/O では，メモリ空間の一部を I/O に割り当てる．プロセッサがメモリの読み書きを行った際，メモリアドレスを判別し，アドレスが I/O に割り当てられた領域であれば，アドレスを I/O ポート番号とみなし，データの読み書きを外部に対する情報の入出力とする．メモリマップ I/O におけるメモリアクセス（MOV 命令など）は，アドレスのみでメモリか I/O かを判別するので，プログラム上ではメモリアクセスと I/O との見分けがつかなくなる．つまり，メモリマップ I/O を用いると，I/O 専用命令をそろえる必要がなくなり，命令

セットがシンプルになるという利点がある．一方で，メモリアクセスのたびにメモリか I/O かを判別しなくてはならないため，アドレスのデコードが複雑になるというデメリットも存在する．

メモリマップ I/O の最大の利点は，メモリ向けのプロテクション機能がそのまま利用可能となることである．通常 I/O は外部に影響を与える操作となるため，IN 命令や OUT 命令は特権命令とされる．しかし，プロセッサに接続する外部デバイスが高速化してきた現代では，通常ユーザからも高速な I/O を直接操作したいという要求が出てきている．メモリマップ I/O では，通常のメモリを割り当てるのと同様に，メモリのプロテクション機構を用いて I/O アクセスを伴う空間を一般ユーザに割り当てることが可能となる．割り当てられた空間は通常のメモリと同様に操作できるため，一般ユーザがメモリアクセス命令を用いて直接 I/O を行うことが可能となる．

### 6-1-2 I/O デバイスと DMA

計算機で行われる処理が複雑になるに従い，計算機の入出力処理も複雑さを増してきた．このなかには定型的な処理も含まれており，それらを含めてすべてプロセッサのプログラムで実行するのは効率が悪い．そのため，プロセッサを中央演算装置と位置づけ，その周辺に様々な定形処理を実施するモジュールを配置し，全体で一つの計算機を構成するモジュール構成が，大型計算機を中心として実現されるようになってきた．このような構成では，周辺モジュールは中央演算装置であるプロセッサからの I/O により指示を受け実行を行う．中央演算装置が全体を指揮しながら全体の処理を進めていく形になる．

現代では大型計算機だけでなく，より小さなシステム（例えば PC など）でも同様のモジュール構成をとっている．小さなシステムでは，周辺モジュールは「I/O デバイス」と呼ばれることが多い．I/O デバイスはプロセッサからの I/O 指示を受け，プロセッサから独立して動作する．

I/O デバイスで処理する定形処理のうち典型的なものがデータ転送である．プロセッサがメモリ内の異なるアドレス領域の間，あるいはメモリと I/O 空間との間でデータ転送を行おうとしたとき，プロセッサ単体で実行しようとするとデータの読み出しと書き込みを繰り返すことになる．プロセッサでのデータ読み書きはレジスタが基本なので，一度にレジスタサイズでしかデータ転送ができない．小容量のデータであればこれでも問題ないが，データ量が大きくなるとプロセッサのみを用いたデータ転送は効率が悪い．

そこで，データ転送を専門に行う I/O デバイスが広く用いられている．プロセッサに頼らず，外部の I/O デバイスを用いてデータ転送を行うことを DMA（Direct Memory Access）と呼ぶ．通常プロセッサがアクセスするメモリを，プロセッサを介在せず，外部デバイスが直接アクセスすることからこの呼名がつけられている．DMA を実行する際には，プロセッサは転送元アドレス，転送先アドレス，長さなどを指定した後，実行を指示する必要があり，その分は実行までのセットアップ・オーバーヘッドとなる．プロセッサが直接データを転送するのに比べて DMA が高速かどうかは，このセットアップ・オーバーヘッド，データ転送の単位やサイズ，外部 I/O デバイスの実行速度などに依存するため，常に DMA を用いた方が効率的であるとは限らないことに注意が必要である．

**■参考文献**

1) Intel 社：“IA-32 インテル (R) アーキテクチャ ソフトウェア・デベロッパーズ・マニュアル, 中巻 A: 命令セット・リファレンス A-M,” https://www.intel.co.jp/content/dam/www/public/ijkk/jp/ja/documents/developer/IA32_Arh_Dev_Man_Vol2A_i.pdf

■6 群 - 4 編 - 6 章

# 6-2 I/O インタフェース

8 群 1 編「センシングとインタラクション」 参照.
3 群「コンピュータネットワーク」 参照.
4 群 4 編「無線 LAN，無線アクセス，近距離ワイヤレス」 参照.