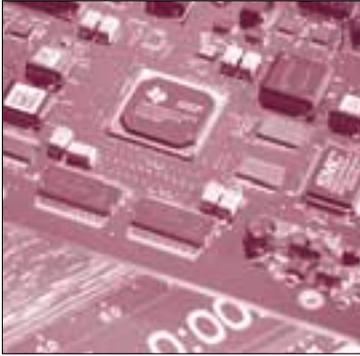


## 第 2 章



デジタル信号処理 IC を  
上手に使いこなそう！

## DSP の基礎知識と開発環境

山口 晶大  
Akio Yamaguchi

本章で取り上げるテーマは、

- DSP とは何か？
- DSP の処理性能と信号処理に必要な演算量
- DSP と FPGA や CPLD との使い分け
- DSP 応用システムの開発手順と開発環境
- DSP 開発の常識

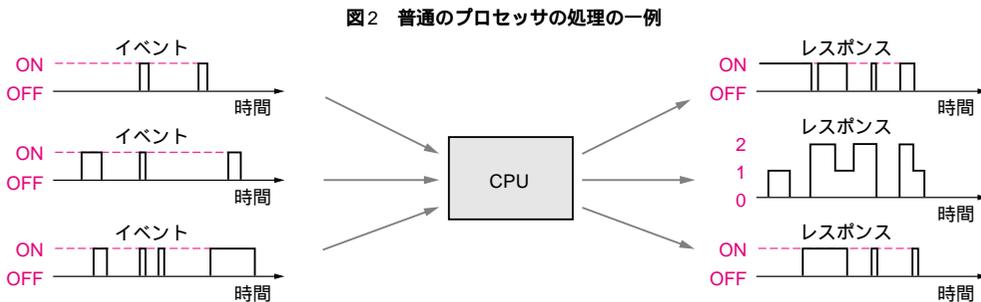
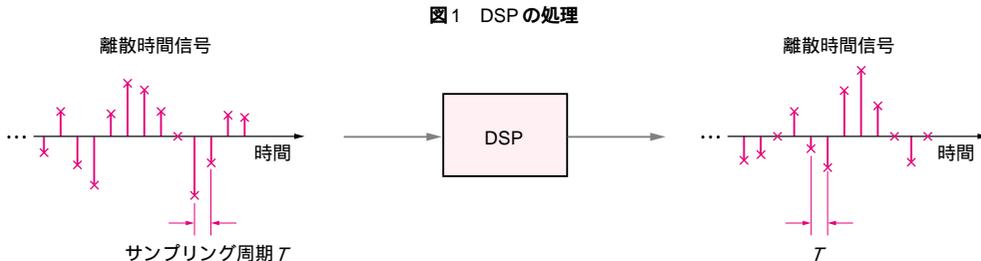
です。章末のコラムでは DSP とリアルタイム OS の関係について解説します。

### DSP とは何か？

#### DSP と普通のプロセッサの処理の違い

DSP のハードウェアの詳しい話をする前に、DSP と普通のプロセッサの処理の違いを比べてみましょう。DSP と普通のプロセッサの処理の違いを図式的に表すと図 1、図 2 のようになります。

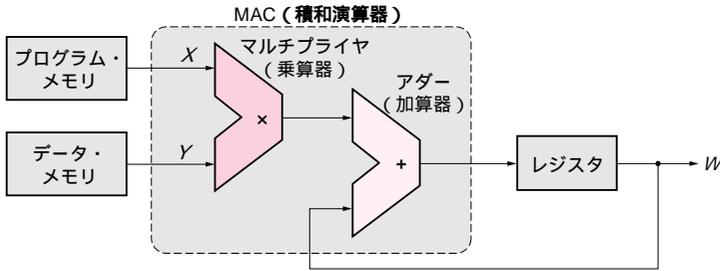
▶ DSP は離散時間信号入力を離散時間信号として出力  
DSP は、図 1 に示すように一定のサンプリング周期



### Keywords

リアルタイム OS, イベント処理, 積和演算, 差分演算, 累和演算, ハーバード・アーキテクチャ, プログラム・メモリ, データ・メモリ, ビット・リバース・アドレッシング機能, FFT 演算, サーキュラ・アドレッシング機能, FIR フィルタ, DMA, 積和演算性能, MAC, Multiply Accumulate, MIPS, FLOPS, Million Instructions Per Second, Floating Operations Per Second, ALU, PROM, ブート・ロード, 固定小数点 DSP, 浮動小数点 DSP, 倍精度乗算, 単精度乗算, FFT, スーパーコンピュータ, x86 系プロセッサ, CPI, Clock cycles per Instruction, FPGA, CPLD, 16 ビット乗算器マクロ, MATLAB, Octave, Scilab, INRIA, Hypersignal/VAB, SystemView, CoolEdit Pro, 信号処理ライブラリ, 画像処理ライブラリ, 単精度ランタイム・ライブラリ, バッチ処理, アセンブラ。

図3 DSPのハードウェア構成 ハーバード・アーキテクチャと積和演算器



の離散時間信号を入力して、その処理結果を同じく離散時間信号として出力するのが最も基本的な処理です。入力および出力はともに途切れなく連続的に続く離散的データ列となります。

まさに、離散時間信号処理のための算術演算(加減算や乗算)が主体のプロセッサがDSPです。なぜDSPが特殊なハードウェア構成をもつのかというと、このようなデータ・ストリームの処理を一般的なプロセッサよりも高速で処理するためです。

▶ 一般的なプロセッサはイベント処理型

一方、図2に示すように一般的に普通のプロセッサの入力となるのは、外部からの複数の非同期のイベント(event)となります。入力したイベントに応じて相異なる論理演算主体の処理を行うのがプロセッサの役割です。DSPの場合は、入力信号のサンプリング周期は一定でしたが、普通のプロセッサでは図2のように各イベントの発生するタイミングは一定していないことも大きな違いです。

DSPの定義とハードウェア構成

DSPとは離散時間信号処理を行うためのプロセッサであることを説明しましたが、そのための特別なハードウェア構成とはどのようなものでしょうか？最も単純なDSPの定義は次のようなものです。

- 1クロックで積和演算ができるプロセッサ

積和演算とは次式に示す形の演算です。

$$w = w + xy \dots \dots \dots (1)$$

ただし、xおよびy：入力、w：出力

乗算(積)と加算(和)を組み合わせた演算なので積和演算と呼ばれます。実は、信号処理の演算の多くはこの積和演算の形を取っているのです。差分や累和演算、さらにはデジタル・フィルタや各種の相関演算も積和演算の一種として処理可能です。

▶ FIR デジタル・フィルタの演算例

一例を挙げると、FIR デジタル・フィルタの演算は次式のように定義されますが、これは式(1)の積和演算をN回繰り返した形になっていることがわかります。

図4 最も基本的なDSPの利用形態

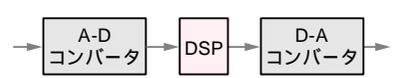
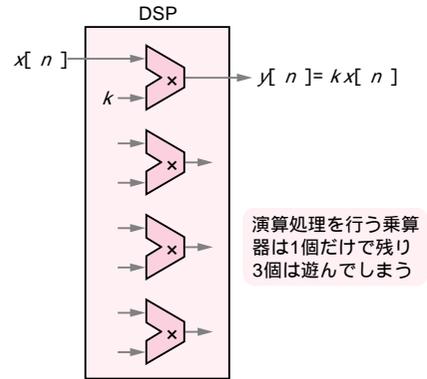


図5 乗算器複数(4個)搭載したDSPでの図4の処理のようす(x[n]=ky[n]の演算の処理例)



$$w[n] = \sum_{i=0}^{i < N} x[n-i]k[i] \\ = x[n]k[0] + x[n-1]k[1] + x[n-2]k[2] + \dots + x[n-(N-1)]k[N-1] \dots (2)$$

ただし、N：フィルタ次数(タップ長)、x[n]：入力信号、y[n]：フィルタ係数、w[n]：出力信号  
積和演算を高速で処理できるということは、言い換えれば離散時間信号処理を高速処理できることになるのです。

▶ 積和演算を1クロックで処理するハードウェア構成  
積和演算そのものは普通のプロセッサでも実行可能なのですが、DSPはこれを1クロックで高速処理するために、次に示すような特徴のあるハードウェア構成を取っています。

- クロックで乗算が可能な乗算器(積和演算器)を内蔵
- ハーバード・アーキテクチャをもつ

このDSPのハードウェアの特徴は、図3のように積和演算を高速で処理するために、1クロックで処理が可能な乗算器と加算器を内蔵している点です。乗算器と加算器を一体化した構成になっているので、二つをまとめて積和演算器と呼びます。

積和演算器には二つの入力XとYがありますが、1クロックで積和演算を行うためには、二つの入力データの読み込みも1クロックで同時に行わなければなりません。そのために、プログラム・メモリとデータ・メモリおよびそれぞれのバスが分離、独立したハーバード・アーキテクチャと呼ばれる構成をDSPはもつ