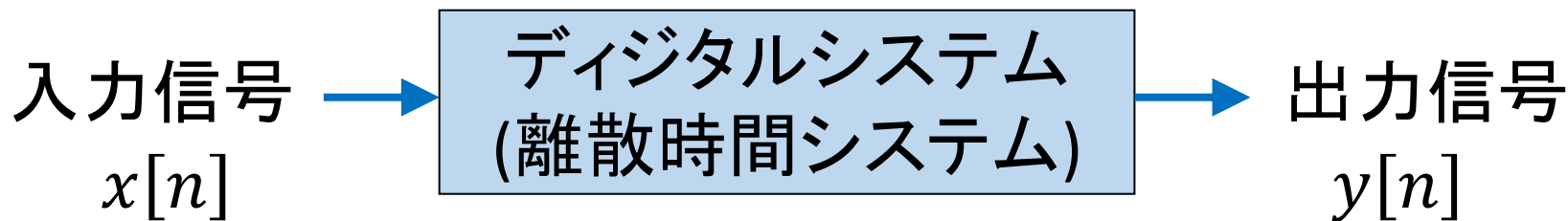


# 電気電子物理工学実験Ⅲ マイクロプロセッサ

課題: デジタルフィルタ

埼玉大学工学部電気電子物理工学科

## ■ デジタルフィルタと伝達関数



システムの入力と出力の比 = 伝達関数  $H(z)$

$$Y(z) = H(z)X(z)$$

伝達関数の一般形  $H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 - a_1z^{-1} - a_2z^{-2}}$  (二次の例)



$$(1 - a_1z^{-1} - a_2z^{-2})Y(z) = (b_0 + b_1z^{-1} + b_2z^{-2})X(z)$$

逆 $z$ 変換して

$$\begin{aligned} y[n] - a_1y[n-1] - a_2y[n-2] \\ = b_0x[n] + b_1x[n-1] + b_2x[n-2] \end{aligned}$$

## ■ 差分方程式

$$y[n] - a_1y[n - 1] - a_2y[n - 2] \\ = b_0x[n] + b_1x[n - 1] + b_2x[n - 2]$$



$$y[n] = b_0x[n] + b_1x[n - 1] + b_2x[n - 2] \\ + a_1y[n - 1] + a_2y[n - 2]$$

入力 $x$ の現在( $x[n]$ )、1つ前( $x[n - 1]$ )、2つ前( $x[n - 2]$ )、  
および出力 $y$ の1つ前( $y[n - 1]$ )、2つ前( $y[n - 2]$ )の値を  
用いて新しい出力 $y[n]$ を求める

➡  $n$ の値を1ずつ増やしながら繰り返す  
(課題プログラムでは $n = 0$ から49まで)

# ■ 作成プログラムにおける入力 $x[n]$ と出力 $y[n]$

入力データ: アドレス400~449 (50個)  
出力(50個)はアドレス450~499に保存

```

org 400
IDATA:
db 100 ←  $x[0]$  ( $n = 0$ )
db 143 ←  $x[1]$  ( $n = 1$ )
db 52  ←  $x[2]$  ( $n = 2$ )
db 163 ←  $x[3]$  ( $n = 3$ )
:
db 135 ←  $x[49]$  ( $n = 49$ )

```

```

org 450
ODATA:
db 0 ←  $y[0]$  ( $n = 0$ )
db 0 ←  $y[1]$  ( $n = 1$ )
db 0 ←  $y[2]$  ( $n = 2$ )
:
db 0 ←  $y[49]$  ( $n = 49$ )

```

## ● AC ← 入力 $x[n]$

```

LDA X, IDATA    X ← 400(IDATA)
LD  WA, X
LD  AC, N        AC ← n
ADD WA, AC
LD  X, WA        X ← 400 + n
LD  AC, (X)      AC ←  $x[n]$ 

```

N: db 0 ←  $n$

## ● $y[n]$ ← 出力Y

```

LDA X, ODATA    X ← 450(ODATA)
LD  WA, X
LD  AC, N        AC ← n
ADD WA, AC
LD  X, WA        X ← 450 + n
LD  AC, Y
LD  (X), AC       $y[n]$  ← 出力Y

```

Y: db 0 ← 差分方程式計算結果

# ■ 作成プログラムにおける入力 $x[n]$ と出力 $y[n]$

(別の方法)

```

org 400
IDATA:
db 100 ←  $x[0]$  ( $n = 0$ )
db 143 ←  $x[1]$  ( $n = 1$ )
db 52  ←  $x[2]$  ( $n = 2$ )
db 163 ←  $x[3]$  ( $n = 3$ )
:
db 135 ←  $x[49]$  ( $n = 49$ )

```

```

org 450
ODATA:
db 0 ←  $y[0]$  ( $n = 0$ )
db 0 ←  $y[1]$  ( $n = 1$ )
db 0 ←  $y[2]$  ( $n = 2$ )
:
db 0 ←  $y[49]$  ( $n = 49$ )

```

```

LDA X, IDATA
LD  IADDR, X      IADDR ← 400
LDA X, ODATA
LD  OADDR, X      OADDR ← 450
:
LD  X, IADDR     } AC ←  $x[n]$ 
LD  AC, (X)      }
INC X            } 次の $x$ 読み出し
LD  IADDR, X     } に備える
:
LD  X, OADDR     }  $y[n] ← AC$ 
LD  (X), AC      }
INC X            } 次の $y$ 読み出し
LD  OADDR, X     } に備える

```

```

IADDR: dw 0 ← Input data address
OADDR: dw 0 ← Output data address

```

# ■差分方程式の計算 $y[n] = b_0x[n] + b_1x[n - 1] + b_2x[n - 2] + a_1y[n - 1] + a_2y[n - 2]$

```

LD AC, 50
LD DCOUNT, AC
L0:
X0 ← x[n]
Y ← X0 * b0

AC ← X1 * b1
Y ← Y + AC

AC ← X2 * b2
Y ← Y + AC

AC ← Y1 * a1
Y ← Y + AC

AC ← Y2 * (-a2)
Y ← Y - AC

y[n] ← Y

X2 ← X1
X1 ← X0
Y2 ← Y1
Y1 ← Y
    
```

$a_2$ が負の場合

```

LD AC, DCOUNT
DEC AC
LD DCOUNT, AC
JNZ L0
HLT

DCOUNT: db 0
X0: db 0 ← x[n]
X1: db 0 ← x[n - 1]
X2: db 0 ← x[n - 2]
Y: db 0 ← y[n]
Y1: db 0 ← y[n - 1]
Y2: db 0 ← y[n - 2]
B0: db xx ← b0
B1: db xx ← b1
B2: db xx ← b2
A1: db xx ← a1
A2: db xx ← -a2
    
```

$n = 0 \sim 49$ について  
繰り返し

$x[-1], x[-2],$   
 $y[-1], y[-2]$ は  
全て0とする

あらかじめ計算して  
xxの部分に記入しておく

## ■課題プログラム

$$(b_0 = b_2, b_1 = 2b_0)$$

$$H(z) = \frac{0.13608 + 0.27216z^{-1} + 0.13608z^{-2}}{1 - 0.72057z^{-1} - (-0.26490)z^{-2}}$$



$$y[n] = 0.13608x[n] + 0.27216x[n-1] + 0.13608x[n-2] \\ + 0.72057y[n-1] - 0.26490y[n-2]$$

係数前の符号(加算/減算)に注意

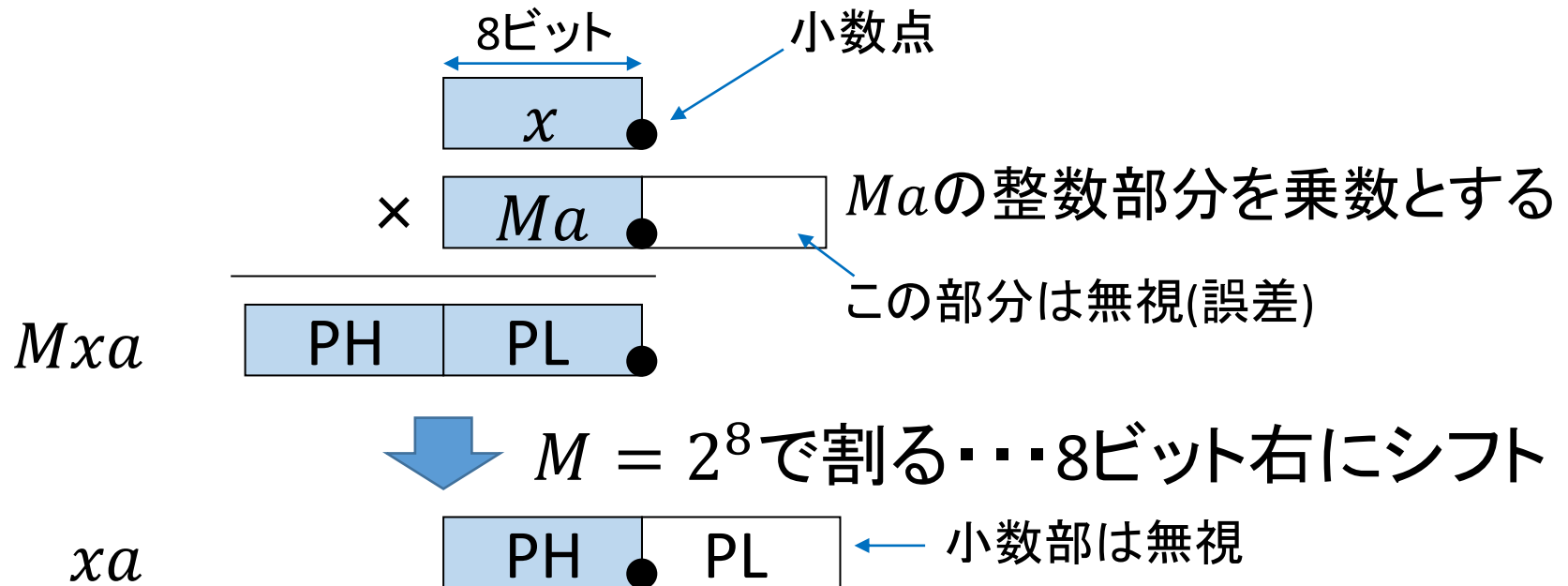


- $Y \leftarrow 0.13608 * X_0 + 0.27216 * X_1 + 0.13608 * X_2 \\ + 0.72057 * Y_1 - 0.26490 * Y_2$
  - $X_2 \leftarrow X_1, X_1 \leftarrow X_0, Y_2 \leftarrow Y_1, Y_1 \leftarrow Y$
- } 繰り返す

## ■ 小数の乗算

$$x \times a = (x \times (Ma)) / M$$

➡  $M$ として $256 = 2^8$ を用いる



➡ つまり、 $x$ (8ビット)と $Ma$ (8ビット)の積の上位8ビット(PH)が求める乗算結果となる(PL部分は無視→計算省略可)

➡ デジタルフィルタにおいて $a$ は定数であり、予め $Ma$ を計算しておけばよい (プログラム中に"db"で記載)



## ■乗算をサブルーチンとして呼び出す

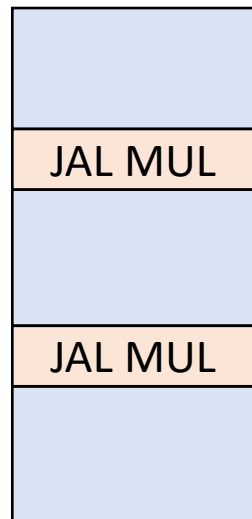
出力Yの計算過程で、 $0.13608 * X0$ 、 $0.82057 * Y1$ など乗算を複数回実行

➡ 乗算をサブルーチンとして必要なところで呼び出す

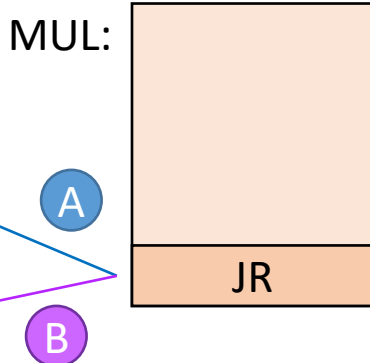
### サブルーチンとは

- ・プログラム中の複数の箇所からジャンプして来る
- ・処理が完了したら元の場所(の次のアドレス)へ戻る

本体プログラム



乗算実行  
サブルーチン



・JAL (Jump And Link)

戻りアドレスを保存(Link)してサブルーチンへジャンプ

・JR (Jump for Return)

保存している戻りアドレスへジャンプ

## ■ サブルーチン関連の命令定義

JAL arg

nbyte 3

opcode xxxxxxxx

0: MAR <- PC, PC <- inc

1: MDR <- mem

2: IR <- MDR

3: MAR <- PC, PC <- inc

4: MDR <- mem, MAR <- PC, PC <- inc

5: MDRH <- MDR, MDR <- mem, X <- PC

6: PC <- MDRW

JAL label

alias JAL arg

JR

alias JMP X

JR命令はJMP Xと同じ

戻りアドレス保存に  
レジスタXを利用



**注意**

・サブルーチンから  
さらにサブルーチン  
を呼び出すことは  
できない

・サブルーチン内で  
レジスタXを書き換え  
てはならない

## ■乗算サブルーチンの使用方法

LD	AC, X1	← 被乗数(MPCD)、乗数(MPLY)を設定
LD	MPCD, AC	(ここではX1 * B1を計算)
LD	AC, B1	
LD	MPLY, AC	
JAL	MUL	← JALでサブルーチン呼び出し
LD	AC, PH	← ここに戻ってきたとき、積はPHに入っている
:		
MUL:		
XOR	AC, AC	
LD	PH, AC	← PHを0に初期化
LD	AC, 8	
LD	COUNT, AC	← COUNT(繰り返し回数)を8に初期化
:		
JR		← 乗算が終了したらJRでサブルーチンから戻る
COUNT:	db 0	
WMPLY:	db 0	← サブルーチン中で使用するメモリ領域
MPCD:	db 0	
MPLY:	db 0	
PH:	db 0	
PL:	db 0	