計算機方式論

第4章 演算方式

演算方式

①アキュムレータ方式

②多数レジスタ方式

③スタック方式

④主記憶間直接演算方式

⑤ベクトル演算−並列プロセッサ方式

⑥ベクトル演算−パイプライン方式

①アキュムレータ方式

- 算術論理演算ユニットALUへの入出力…レジスタ
- ひとつの演算専用レジスタ
  アキュムレータ（accumulator，累算器）
- 1オペランド方式
- アキュムレータと主記憶とのデータの引渡しが頻繁になり、演算効率が悪い
  ・・・ノイマンの隘路

〔例〕a*b+c*d

```
LOAD  b    (b)→ACC
MULT  a    (a)*(ACC)→ACC
STORE t    (ACC)→t      a*bを一時t番地に待避
LOAD  d    (d)→ACC
MULT  c    (c)*(ACC)→ACC
ADD   t    (t)+(ACC)→ACC
```

主記憶アクセス12回 命令数6個 一時的記憶領域1個

②多数レジスタ方式

- アキュムレータ方式の拡張
- 多数の演算用レジスタを設けた
- 2～3オペランド方式
- アキュムレータ方式に比べ、演算効率がよい
- レジスタの集まりをレジスタファイルとよび、アドレス部でどのレジスタを使うかを指定

〔例〕a*b+c*d　レジスタ−メモリ方式の演算命令の場合

```
LOAD  R1,a   (a)→R1
MULT  R1,b   (R1)*(b)→R1    メモリ乗算
LOAD  R2,c   (c)→R2
MULT  R2,d   (R2)*(d)→R2    メモリ乗算
AR    R1,R2  (R1)+(R2)→R1  レジスタ加算
```

主記憶アクセス9回　命令数5個

②多数レジスタ方式-2

- 主記憶アクセスは、LOAD/STOREだけで行う
- レジスタどうしの演算だけを用いる

〔例〕a*b+c*d　レジスタ-レジスタ方式の演算命令の場合

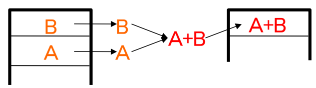
```
LOAD  R1,a    (a)→R1
LOAD  R2,b    (b)→R2
LOAD  R3,c    (c)→R3
LOAD  R4,d    (d)→R4
MR    R1,R2   (R1)*(R2)→R1  レジスタ乗算
MR    R3,R4   (R3)*(R4)→R3  レジスタ乗算
AR    R1,R3   (R1)+(R3)→R1  レジスタ加算
```

主記憶アクセス11回　命令数7個

③スタック方式

- 0オペランド方式
- オペランドが専用のプッシュダウンスタック上にとられる（後出）

ADD命令の実行



プッシュダウンスタック

④主記憶間直接演算方式

- 直接、主記憶上のデータに演算を施し、結果も主記憶上の番地に格納する

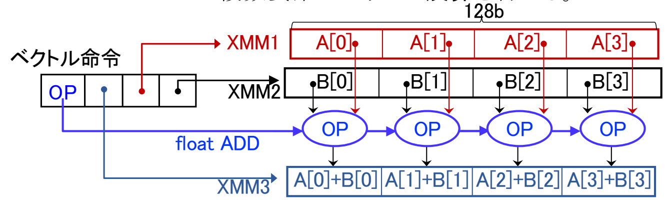
　MULT　a,b　(a)*(b)→a

　4回の主記憶アクセス

⇒ ノイマンの隘路

⑤ベクトル演算-並列演算方式
（ベクトルプロセッサ方式）

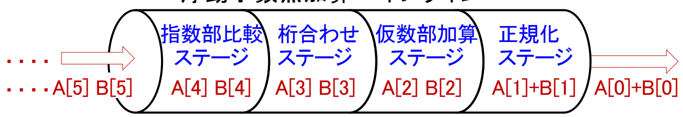
- ベクトル演算：配列どうしの演算。対応する各配列要素に同じ演算を施す（配列A,Bの加算：A[0]+B[0],…, A[n−1]+B[n−1]）。
- SIMD命令として、多くのプロセッサに設けられている。（SIMD:Single Instruction stream Multi Data stream）
- 配列A,Bの各要素どうしの演算を、対応する演算装置で同時に（並列）処理する。
- インテルX86以上では、128bitsのレジスタファイルXMM0～XMM15で複数要素のベクトル演算を行える。



⑥ベクトル演算–パイプライン演算方式
（ベクトルプロセッサ方式, アレイ計算機）

- 配列要素どうしのパイプライン方式の演算で、Crayなどのスーパーコンピュータで導入された。
- 演算を複数のステージに分けて、配列の各要素を先頭から順番にこのステージに送る。
- 理想的には、ステージ数倍高速化される。

浮動小数点加算パイプライン



9

浮動小数点加算のステージ分割

浮動小数点数：

仮数部M,指数部E,基数R

S	E	M

数値 $(-1)^S M \cdot R^E$ を表わす。

2つの浮動小数点数F1,F2の和を考える(共に正とする)：

$F1=M1 \cdot R^{E1}$　　$F2=M2 \cdot R^{E2}$　（E1>E2とする）

$F1+F2 = M1 \cdot R^{E1}+M2 \cdot R^{E2}$

$=(M1+M2 \cdot R^{E2-E1}) \cdot R^{E1}$

浮動小数点加算を4ステージに分割：

1. 指数部比較　　E1とE2との比較
2. 桁合わせ　　$M2 \cdot R^{E2-E1}$の計算
3. 仮数部加算　　$M1+M2 \cdot R^{E2-E1}$
4. 正規化

10

7.25+1.125 の浮動小数点加算

$7.25 = (111.01)_2$ ==正規化==⇒ 1.1101×2^2

0	0	1	0	1	1	1	0	1	0	0

基数2

$1.125 = (1.001)_2$ ==正規化==⇒ 1.001×2^0

0	0	0	0	1	0	0	1	0	0	0

1.指数部比較

3.仮数部加算

1.1101×2^2 ⇔ 1.001×2^0

+) 0.01001×2^2　　2.桁合わせ

10.00011×2^2

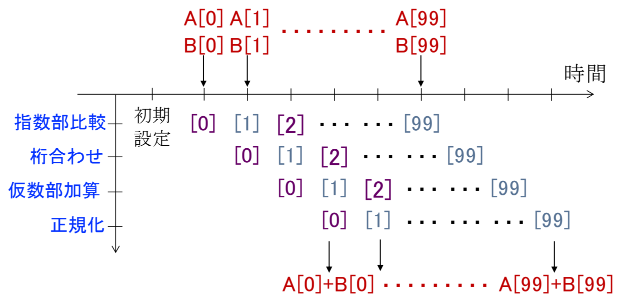
⇩ 4.正規化

1.000011×2^3

0	0	1	1	1	0	0	0	0	1	1

11

浮動小数点配列A,Bの和の演算パイプライン



103クロック強でベクトル和が求まる！
(c.f. 逐次演算だと400クロック)

12