

# 計算機方式論

## 第4章 情報の記憶方式

1

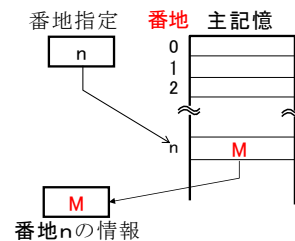
## 情報の記憶方式

- ① 番地指定方式 (アドレッシング方式)
- ② 連想記憶方式
- ③ スタック方式

2

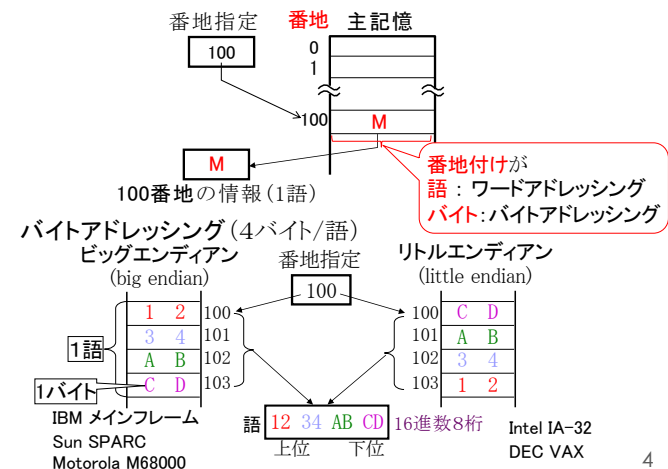
### ① 番地指定方式 (アドレッシング方式)

- 主記憶装置の個々の場所に**固有の番地**(アドレス)を与え、格納されている情報はその**番地**で指定する



3

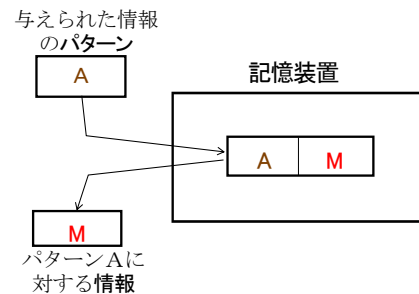
### ワードアドレッシングとバイトアドレッシング



4

## ②連想記憶方式

- 情報の**パターン**で情報の**記憶場所**を指定する。
- 仮想記憶方式の番地変換に使われている



5

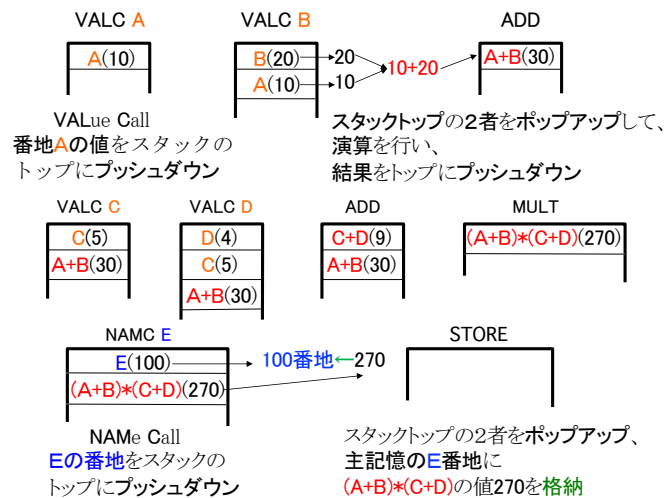
## ③スタック方式

- 命令対象は専用の**プッシュダウンス** **タック**上のデータ。
- ポーランド記法に基づく演算に適する

〔例〕BURROUGHS B5500 等に採用  
主記憶+専用スタックで構成 (100番地)  
数式  $E = (A + B) * (C + D)$   
ポーランド記法  $AB + CD + * E =$   
スタック上のデータに対し、加減乗除を施す

主記憶装置	
A	10
B	20
C	5
D	4
E	270
(100番地)	
VALC A	
VALC B	
ADD	
VALC C	
VALC D	
ADD	
MULT	
NAMC E	
STORE	

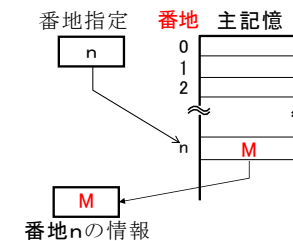
6



7

## 番地指定方式 再掲 (アドレッシング方式)

- 主記憶装置の個々の場所に**固有の番地** (アドレス)を与え、格納されている情報はその**番地**で指定する



8

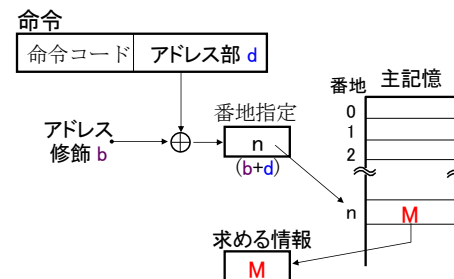
## 番地指定方式-番地の指定法

- 直接番地指定方式
  - 間接番地指定方式
- 絶対番地指定方式
  - 相対番地指定方式

9

## 直接番地指定方式 (direct addressing)

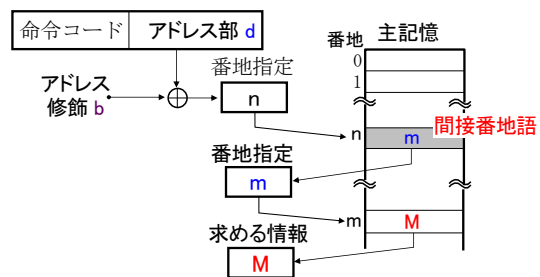
アドレス部で与える番地が、求める情報の記憶場所になる。



10

## 間接番地指定方式 (indirect addressing)

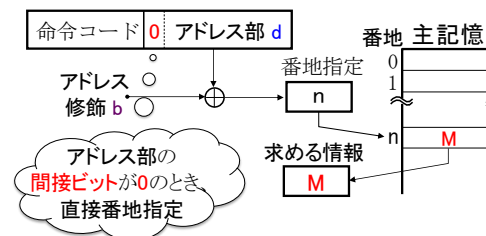
アドレス部で与える番地に格納されている情報が、再び、番地を表わす(間接番地語)。この番地に格納されている情報が、求める情報である場合もあるし、再び、間接番地語として、番地を表わす場合もある。



11

## 間接修飾指定

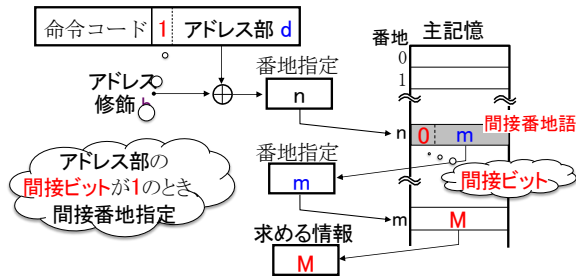
アドレス部や間接番地語に間接修飾指定ビットを設ける。それが0ならば、番地指定された語は求める情報、1ならば、間接番地語を表す。



12

## 間接修飾指定

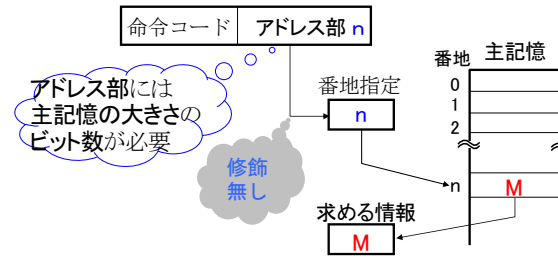
アドレス部や間接番地語に**間接修飾**指定ビットを付加する。  
それが**0**ならば番地指定された語は求める情報、  
**1**ならば番地指定された語は**間接番地語**を表す。



13

## 絶対番地指定方式

命令の**アドレス部**で指定された番地が、そのまま何の修飾も受けずに、主記憶装置の番地を表わす。



14

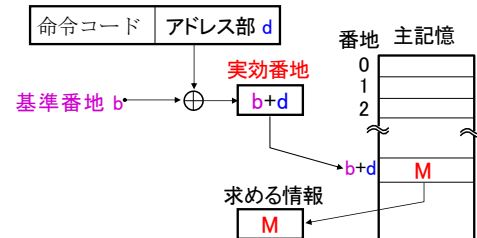
## 相対番地指定方式

アドレス部…**基準番地**からの**相対的位置**を表す。

変位, displacement

基準番地 + アドレス部 … 実際の番地

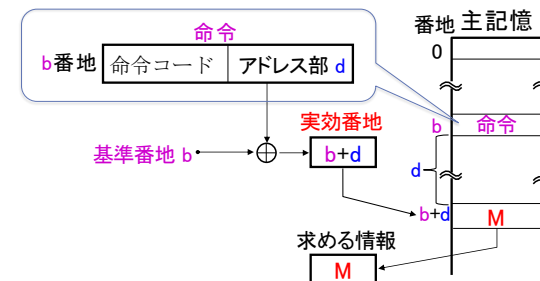
**実効番地**(effective address)



15

## 相対番地指定方式-命令格納番地

〔例1〕 **命令格納番地** + **アドレス部** … **実効番地**  
基準番地 変位



16

## 相対番地指定方式-バンクの番地

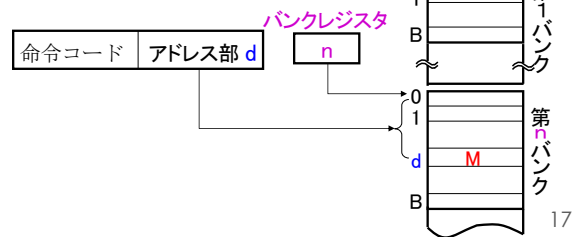
〔例2〕 バンクの番地 + アドレス部 … 実効番地  
基準番地 変位

主記憶をバンクに分割

同じ長さのひと固まりの領域

バンクの指定 … バンクレジスタ

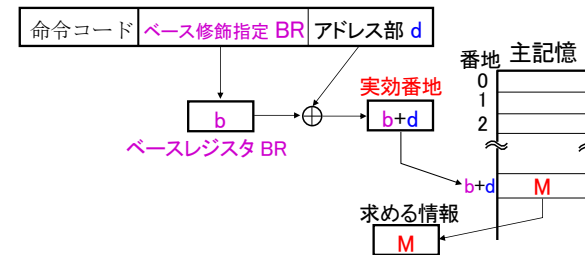
アドレス部 … バンク内の相対番地



17

## 相対番地指定方式-ベース修飾

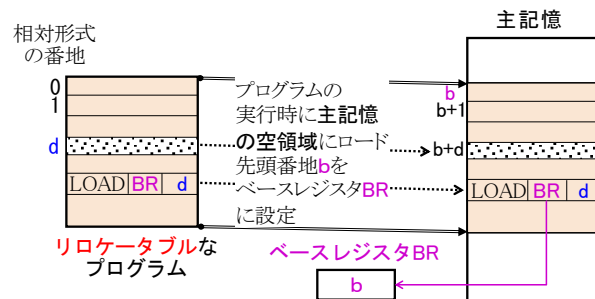
〔例3〕 ベースレジスタ + アドレス部 … 実効番地  
基準番地 変位



18

## ベース修飾の使われ方

プログラムを主記憶の任意の位置にロード(リローケーション,再配置)して実行するために使われる

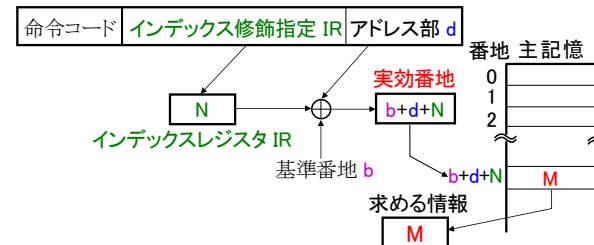


19

## インデックス修飾

基準番地+アドレス部  
+ インデックスレジスタの内容 … 実効番地

インデックスレジスタの内容を規則的に変更することで、命令に変更を加えずに、その命令の番地指定を規則的に変更するプログラミング上の手段

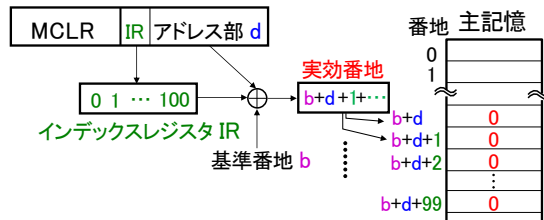


20

## インデックス修飾の例

「IRを0,1,2,...,99と変えていき、d,d+1,d+2,...,d+99番地の内容をクリアする」プログラム

```
CLR  IR      0 → IR
loop: MCLR d(IR)  0 → d+(IR)番地
      INC  IR      (IR)+1 → IR
      BLT  IR,100,loop (IR)<100ならloopへ飛ぶ
```



21

## 演習—命令形式

- 演算パイプライン方式を述べよ。
- 番地指定方式には、絶対番地指定方式と相対番地指定方式がある。それぞれを説明せよ。
- RISCプロセッサの特徴を3つ挙げよ。

22