

情報とコンピュータ

静岡大学工学部

安藤和敏

2005.01.17

9章

マシンアーキテクチャ

- コンピュータを組み立てよう
- アーキテクチャ例 P88計算機
- P88計算機のプログラミング
- まとめ

プログラムが実行されるまで

Pascalプログラム

Z := X + Y

コンパイル

アセンブリ言語の
プログラム

```
COPY AX, X  
ADD AX, Y  
COPY CN1, AX  
COPY AX, CN1  
COPY Z, AX
```

機械語プログラム

```
0010110100100  
0110101010001  
0011001000100  
1100100101101  
0110110100001
```

電気回路による実行

アセンブリ言語と機械語

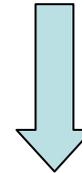
アセンブリ言語

機械語

COPY AX, X



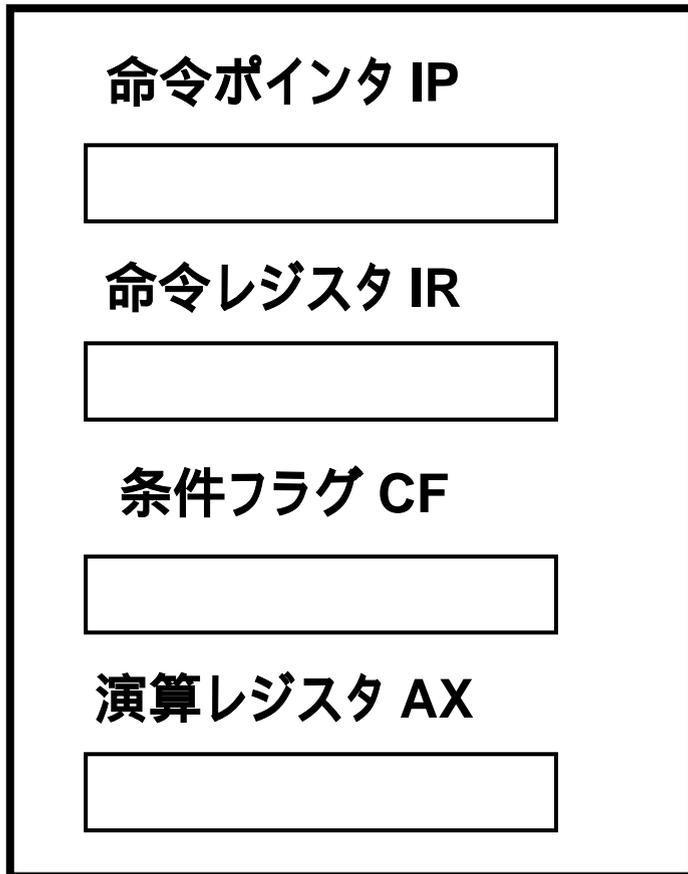
001011100010100



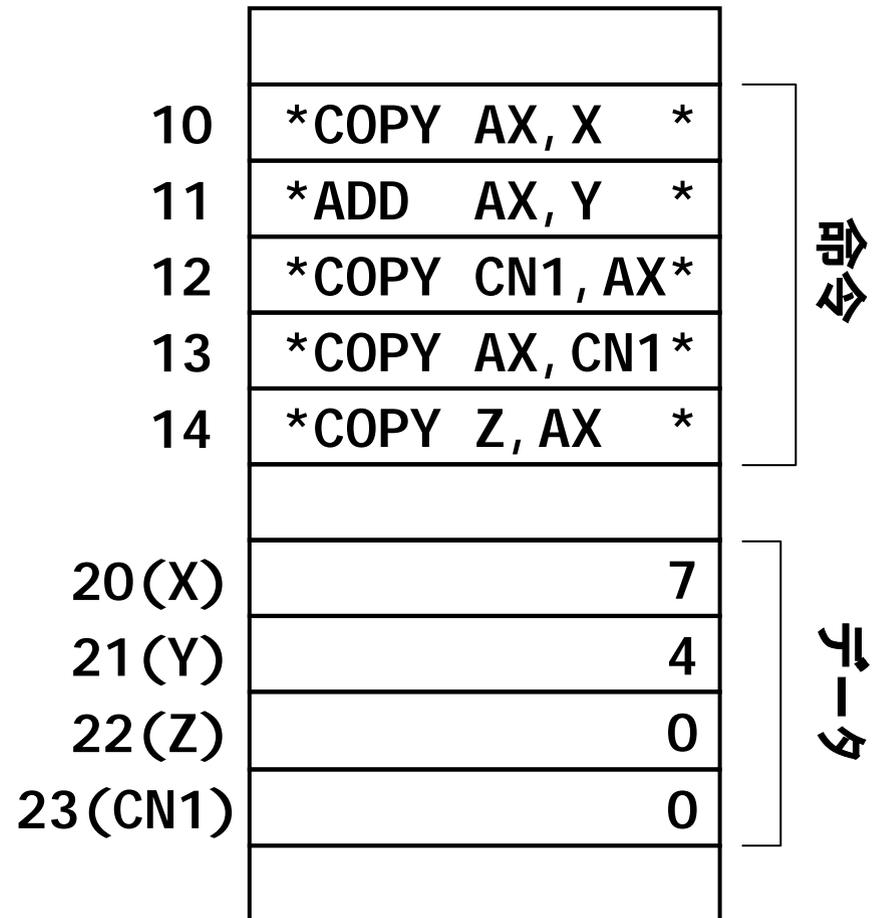
*COPY AX, X *

アセンブリ言語の命令文と機械語命令文は1対1に対応する。

コンピュータアーキテクチャ



中央処理装置 (CPU)



メモリ

命令取り出し-実行サイクル

コンピュータは、次の2つのステップを繰り返して実行する。

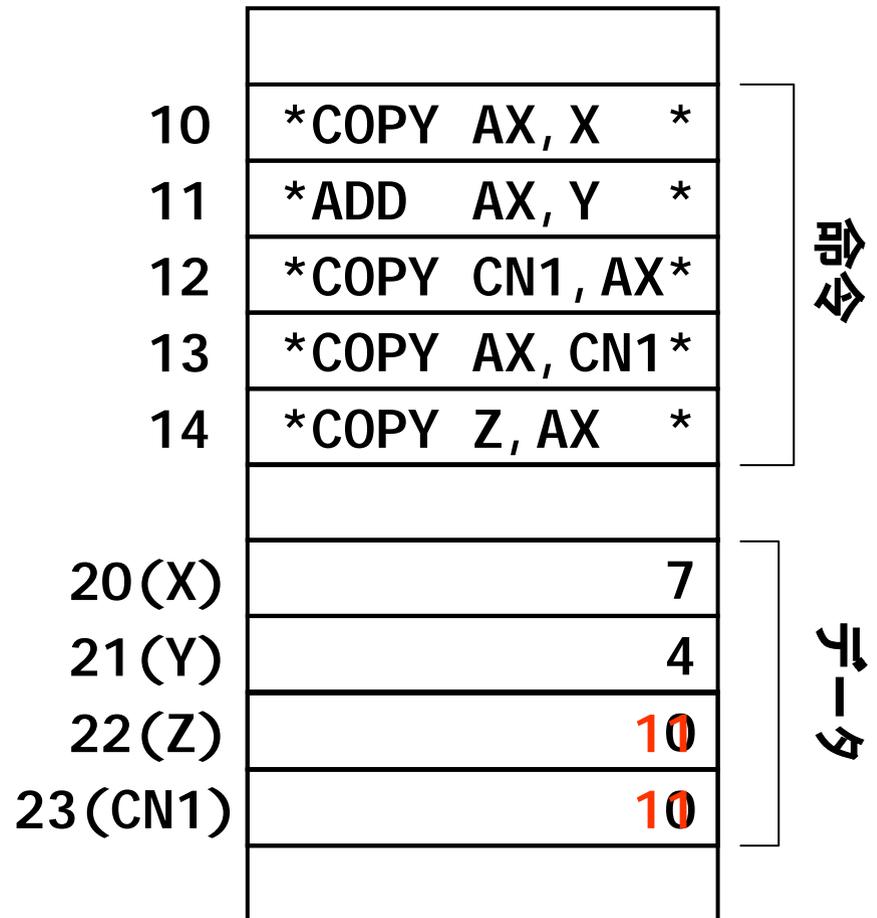
1. (命令取り出し) IPが指定するアドレスに記憶されている命令を取り出し、その命令をIRに入れる。IPの値を1つ増やす。
2. (命令実行) IRにある命令を実行する。

この2つのステップを、**命令取り出し-実行サイクル**という。

命令取り出し-実行サイクルの例



中央処理装置 (CPU)



メモリ

P88計算機の12種の命令(1)

書式

動作

COPY AX, mem

mem番地の内容をAXへコピー

COPY mem, AX

AXの内容をmem番地へコピー

P88計算機の12種の命令(2)

書式

動作

ADD AX, mem

AX := AX + mem番地の内容

SUB AX, mem

AX := AX - mem番地の内容

MUL AX, mem

AX := AX * mem番地の内容

DIV AX, mem

AX := AX / mem番地の内容

P88計算機の12種の命令(3)

書式

動作

CMP AX, mem

CF := AX - mem番地の内容

テキストの記述と異なることに注意

P88計算機の12種の命令(4)

書式

動作

JMP lab1

IP := lab1

JNB lab1

if CF >= 0 then IP := lab1

JB lab1

if CF < 0 then IP := lab1

P88計算機の12種の命令(5)

書式

動作

IN AX

整数を読み込んでAXへ代入する

OUT AX

AXの内容を出力する

これらの命令はそれぞれPascalのReadln, Writeln文に相当する。

簡単なP88プログラム(1)

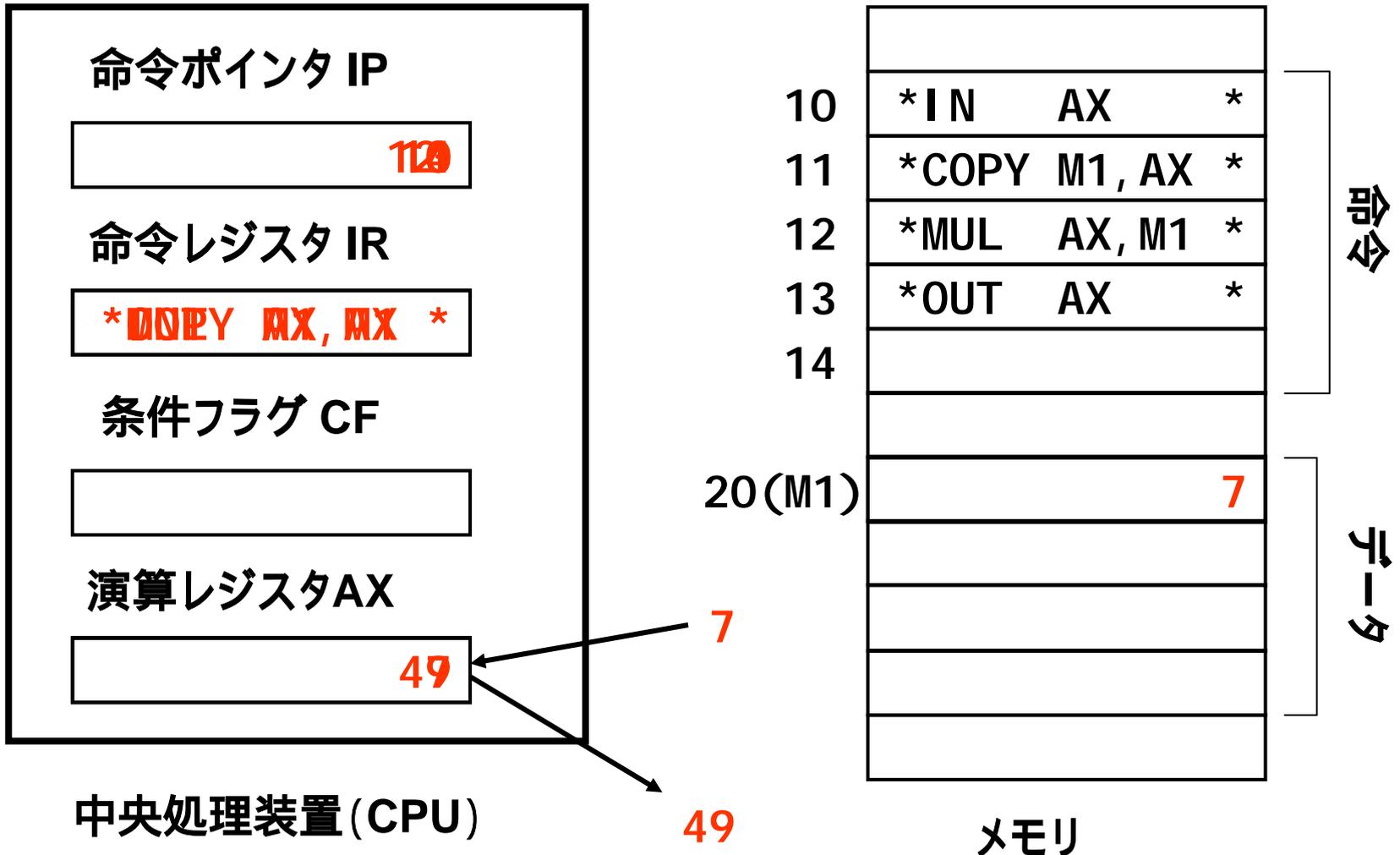
IN AX

OUT AX

簡単なP88プログラム(2)

```
IN    AX  
COPY  M1, AX  
MUL   AX, M1  
OUT   AX
```

簡単なP88プログラム(2)



簡単なP88プログラム(3)

```
IN    AX  
COPY  A, AX  
IN    AX  
COPY  B, AX  
COPY  AX, A  
DIV   AX, B  
OUT   AX
```

簡単なP88プログラム(3)



中央処理装置 (CPU)

15

3

5

10	*IN	AX	*
11	*COPY	A, AX	*
12	*IN	AX	*
13	*COPY	B, AX	*
14	*COPY	AX, A	*
15	*DIV	AX, B	*
16	*OUT	AX	*
20 (A)		15	
21 (B)		3	

命令

データ

メモリ

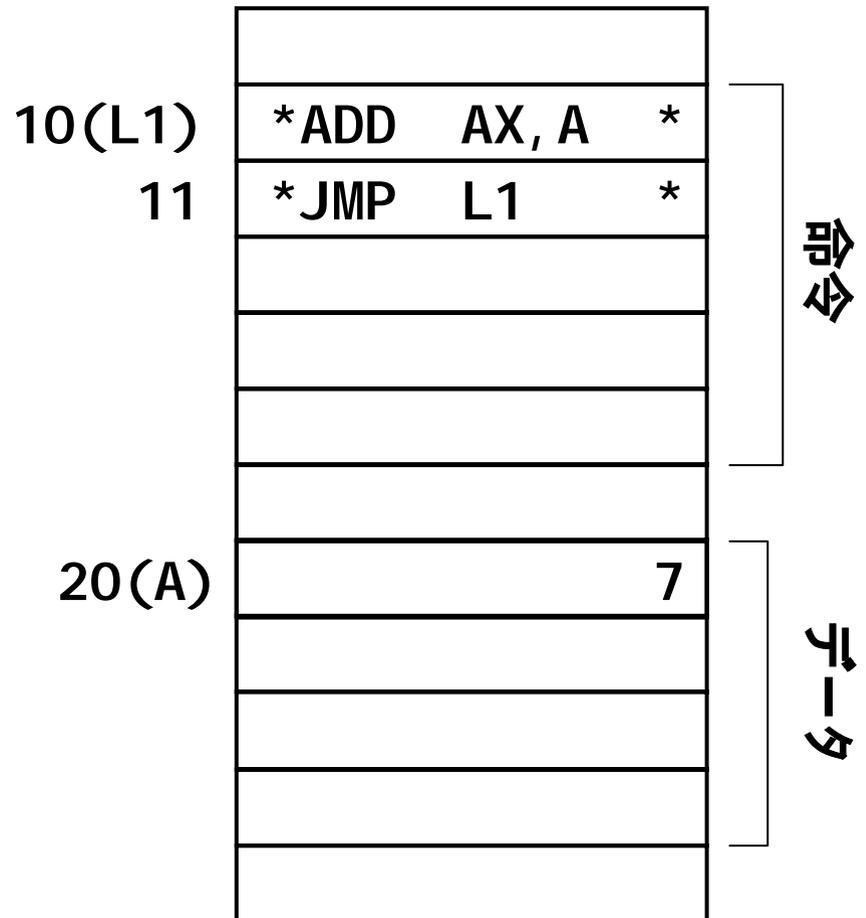
ジャンプ命令 (1)

```
L1    ADD    AX, A  
      JMP    L1
```

ジャンプ命令 (1)



中央処理装置 (CPU)



メモリ

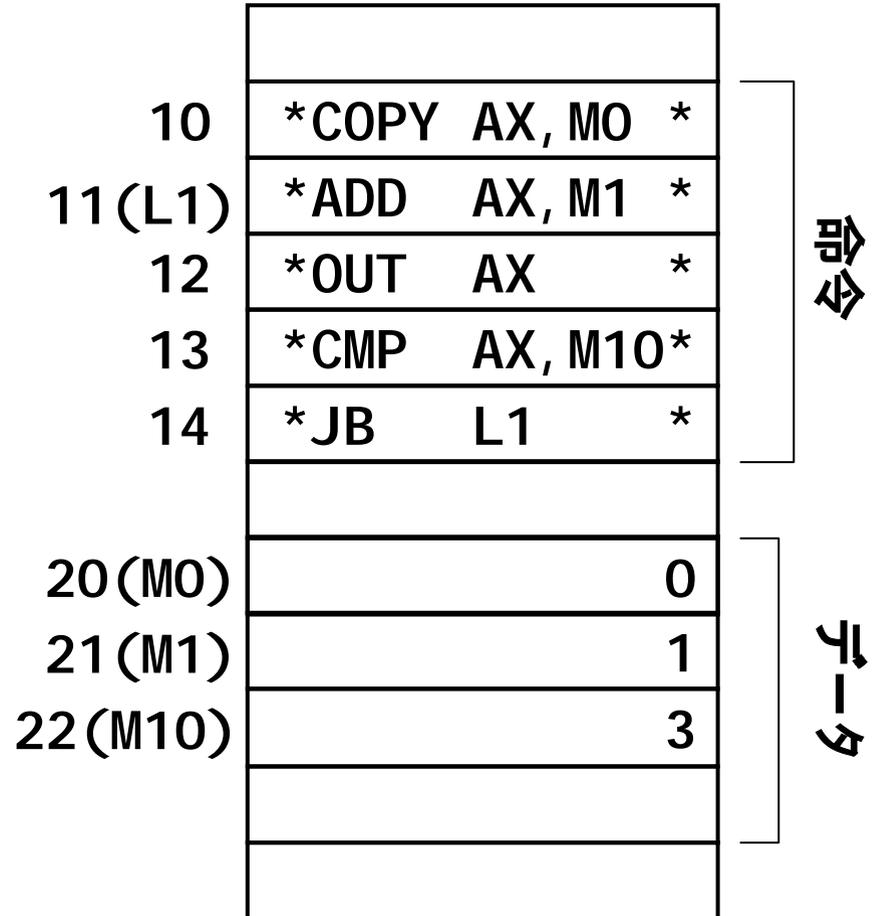
ジャンプ命令(2)

```
L1    COPY AX, M0  
      ADD  AX, M1  
      OUT  AX  
      CMP  AX, M10  
      JB  L1
```

ジャンプ命令 (2)



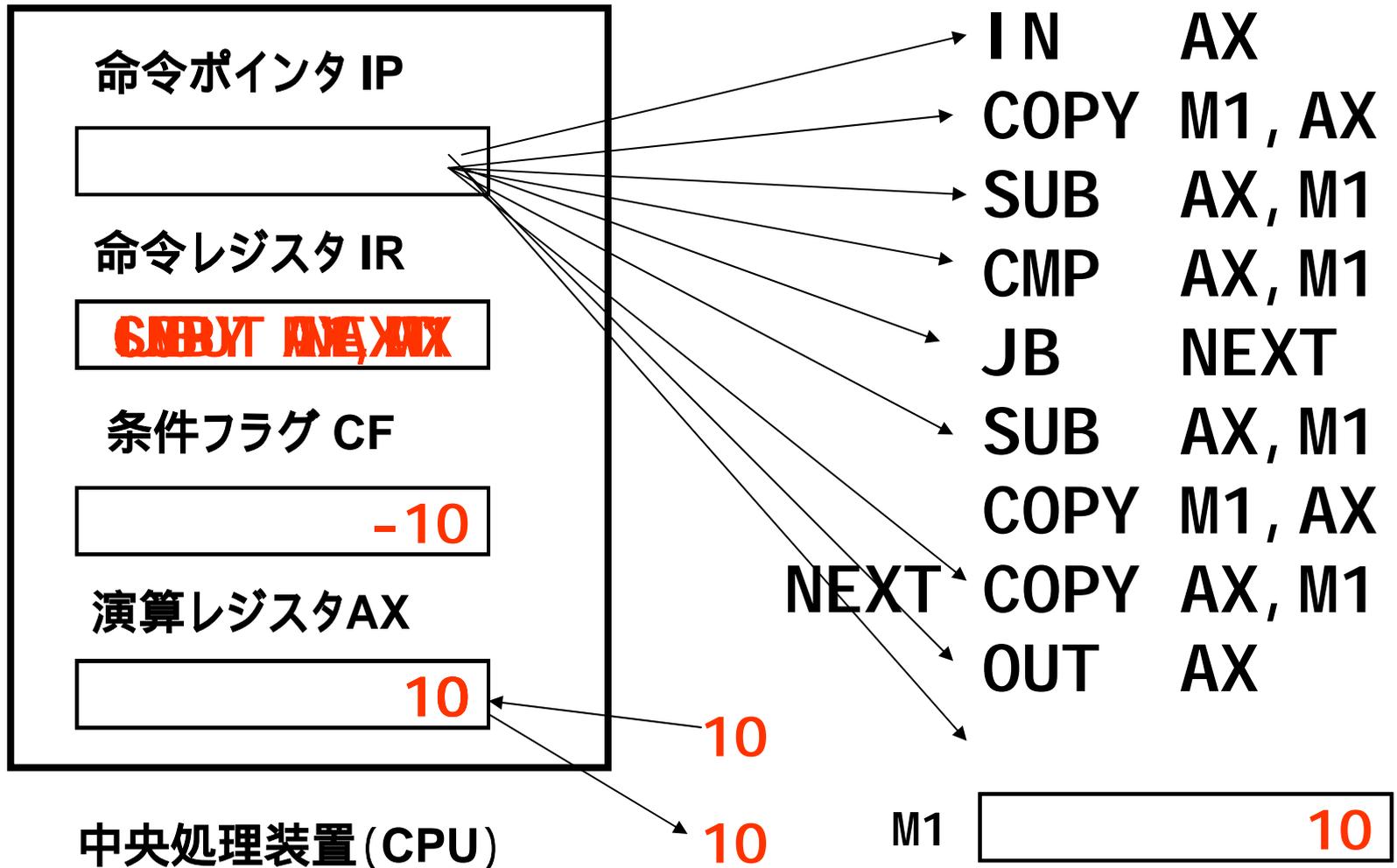
中央処理装置 (CPU)



メモリ

3

ジャンプ命令 (3)



アセンブリ言語

COPY, ADD, CMP などの記号化された形で書かれた機械語命令の言語をアセンブリ言語という。

1940年代, 1950年代のコンピュータ・プログラムはアセンブリ言語で書かれていた。

その後, PascalやCなどのより人間の言葉に近い高水準言語が登場した。現在においては, 大規模はプログラムをアセンブリ言語で書く人はいない。

フォン・ノイマン型アーキテクチャ(1)

命令が計算データとともにメモリに記憶され、命令取り出し-実行サイクルを繰り返すCPUを持つコンピュータ(・アーキテクチャ)は、**フォン・ノイマン型コンピュータ**(アーキテクチャ)と呼ばれる。

このアーキテクチャは、数学、物理学、経済学など、様々な分野においてその才能を発揮した天才 John von Neuman (1903-1957) によって、1940年代に考案された。

60年後の現在においても、どんなコンピュータ(パソコンも、スパコン)もノイマン型アーキテクチャに基づいている。

フォン・ノイマン型アーキテクチャ(2)

フォン・ノイマン型アーキテクチャが登場する前のコンピュータは、1つの決まった計算(例えば、円周率の計算とか図形の面積の計算など)しかできなかった。

なぜならば、プログラムはハードウェアで実現されていたから。

ノイマン型アーキテクチャの画期的な点は、プログラムをハードウェアから独立させて、メモリの中に内蔵したことにある。(プログラム内蔵方式)

メモリの内容を書き換えれば、1つのコンピュータでいろいろな計算を行うことができる。

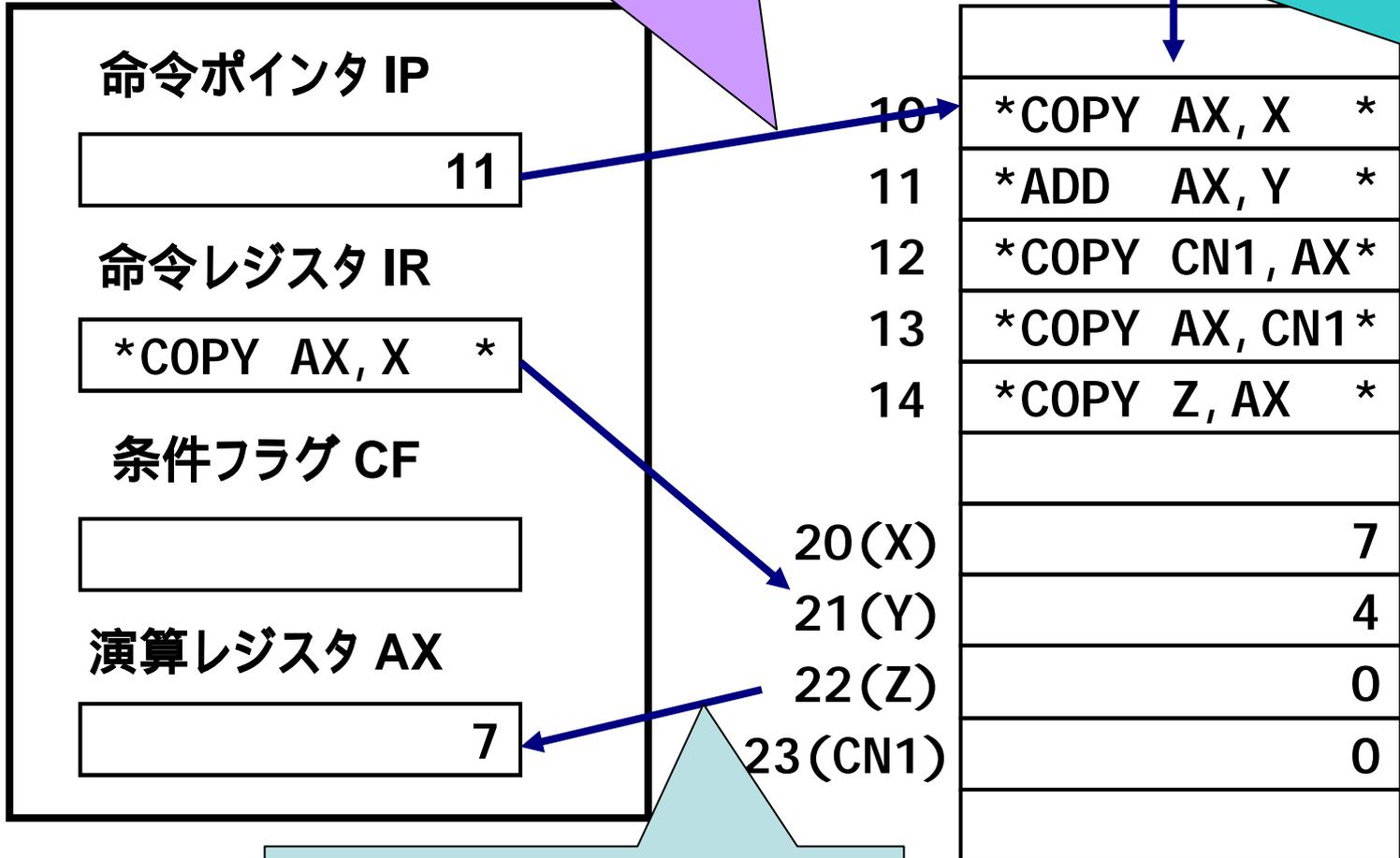
まとめ

Pascal (第1 ~ 4章)

命令取り出し-実行サイクル(第9章)

Z := (X + Y)

翻訳(第10章)



命令解読と実行(第6章)

アセンブリ言語(第9章)