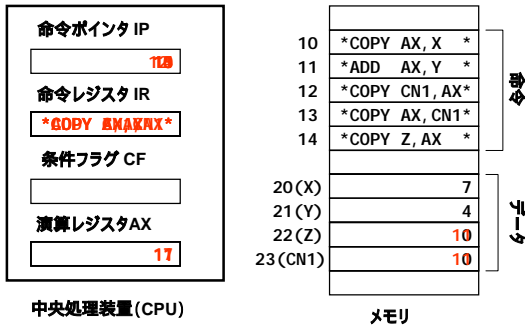


命令取り出し-実行サイクルの例



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。

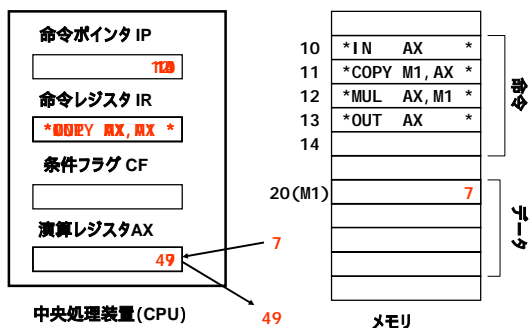
簡単なP88プログラム(1)

```
IN AX
OUT AX
```

簡単なP88プログラム(2)

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

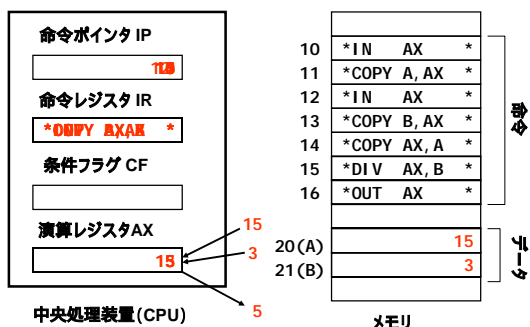
簡単なP88プログラム(2)



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
```

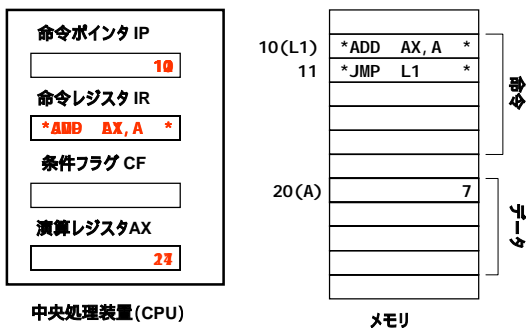
簡単なP88プログラム(3)



ジャンプ命令(1)

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

ジャンプ命令 (1)

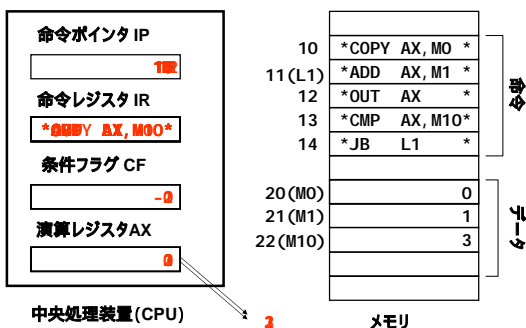


ジャンプ命令 (2)

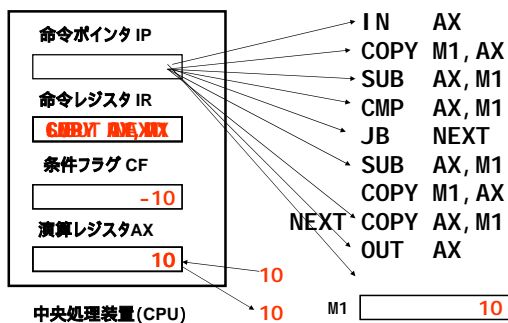
```

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

ジャンプ命令 (2)



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。

