



東邦大学

いのち
生命の科学で未来をつなぐ

コンピュータは命令で動く

命令って？



東邦大学

コンピュータの動作を指示するもの



東邦大学

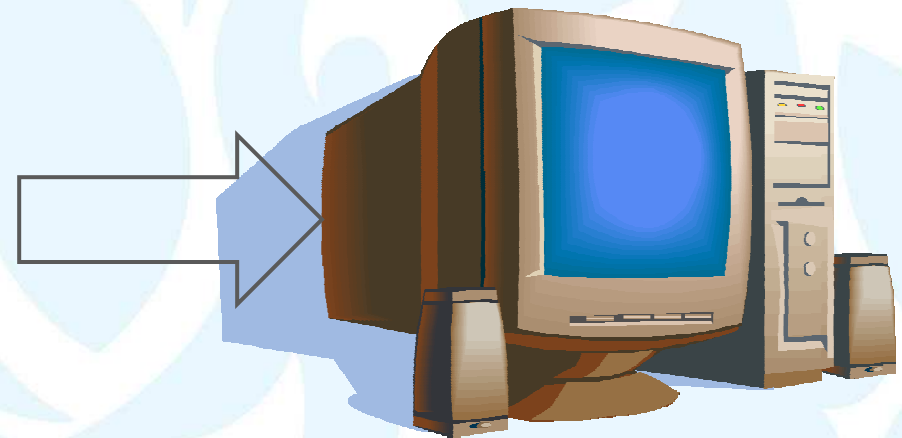
コンピュータの動作を指示するもの

これをせよ

これをせよ

これをせよ

これをせよ



コンピュータの動作を指示するもの

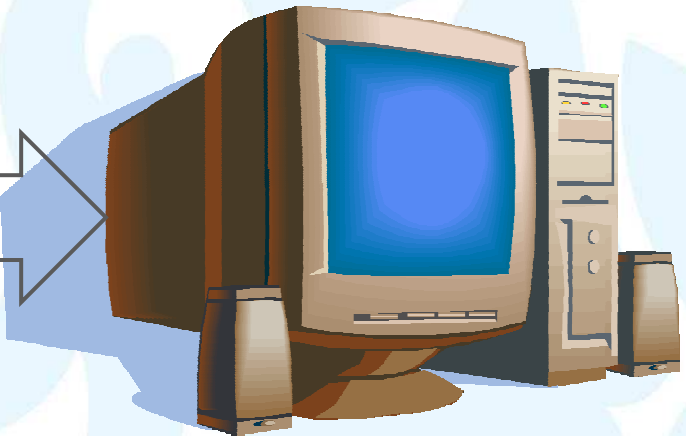
プログラム

これをせよ

これをせよ

これをせよ

これをせよ



東邦大学

じゃあ、プログラムの
部分要素？

If文とかForループとか？

じゃあ、プログラムの
部分要素？

~~If文とかForループとか？~~

ではなくて

じゃあ、プログラムの
部分要素？

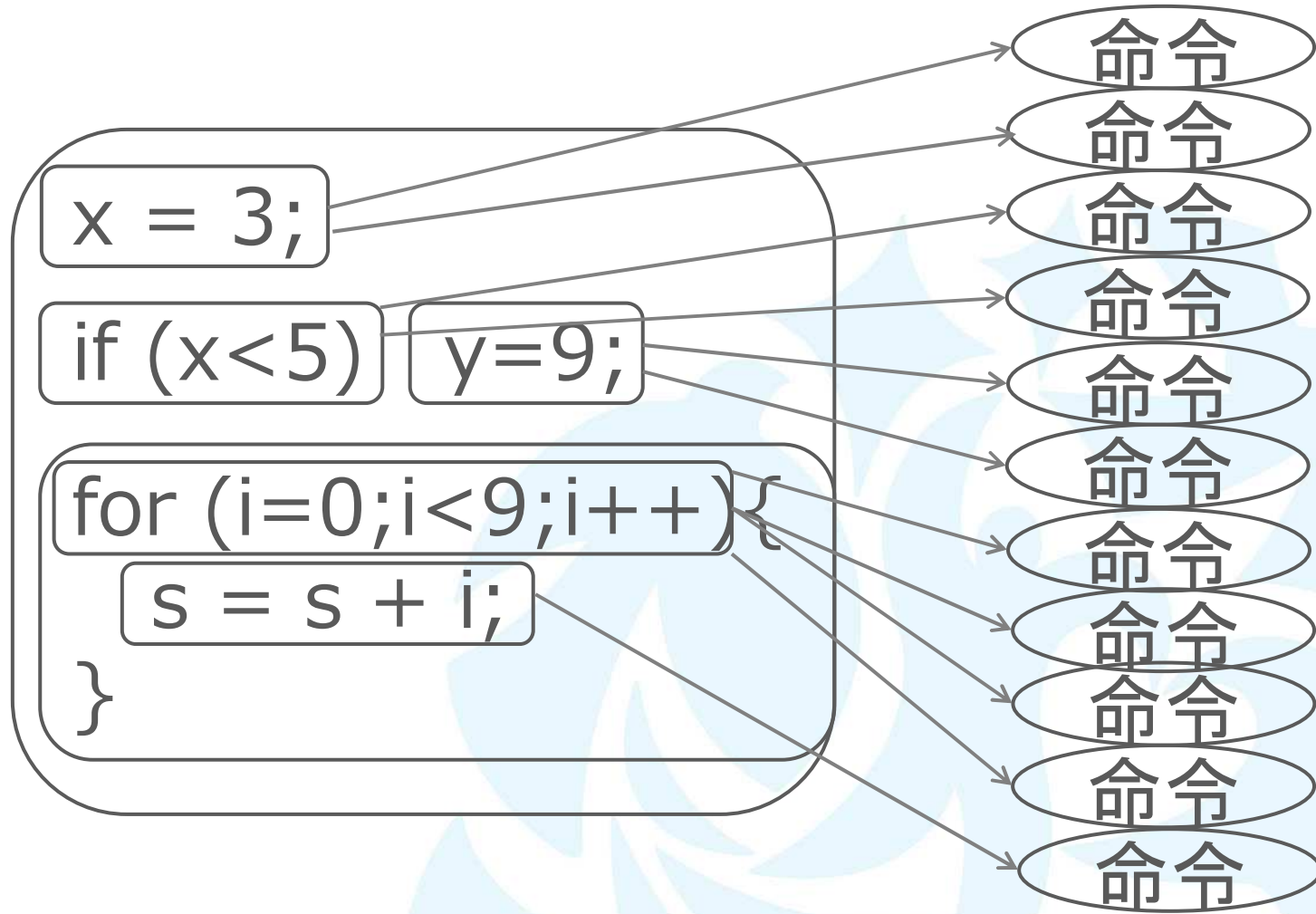
ではなくて

もっと細かい、内部で行われる操作

↑
=ハードウェア



プログラム



コンピュータ

プログラムが命令に分解されるイメージ



東邦大学

脱線！
「命令」って英語で何？

和英辞書を見てみよう



東邦大学

辞書では

order

オーダー？

command

コマンド？

ぐらい？？



実は、コンピュータでは
Instruction と言います

実は、コンピュータでは
Instruction と言います

Instruction =

- ①教育・教えること
- ②指令・訓令・指図・命令

実は、コンピュータでは
Instruction と言います

というわけで、
命令といっても、
教えてあげる、感じね

じゃ命令ってどんなもの？

命令の具体的な例

レジスタGR4の内容と
メモリの33番地の内容を足して
レジスタGR4に書き戻せ

少し説明が要りますね

命令の具体的な例

レジスタGR4の内容と
メモリの33番地の内容を足して
レジスタGR4に書き戻せ

ここではイメージだけ説明します
細かい説明は後から改めて

命令のイメージ

レジスタGR4の内容と
メモリの33番地の内容を足して
レジスタGR4に書き戻せ

命令のイメージ

レジスタGR4の内容と

メモリの33番地の内容を足して
レジスタGR4に書き戻せ

レジスタ = CPU内にある高速・少量メモリ
この例ではGR0番～GR7番の8つある
演算はレジスタ内容に対して行う



命令のイメージ

レジスタGR4の内容と
メモリの33番地の内容を足して
レジスタGR4に書き戻せ

命令のイメージ

レジスタGR4の内容と
メモリの33番地の内容を足して
レジスタGR4に書き戻せ

メモリ = 前にも出てきた (引出し)

メモリは命令とデータを保持できる

引出し番号 = 番地 (アドレス)

ここでは33番地 = 33番の引出し



動作は



東邦大学

命令のイメージ

レジスタGR4の内容と
メモリの33番地の内容を足して
レジスタGR4に書き戻せ

これ

GR3	9
GR4	5
GR5	3

+

32	2
33	1
34	7
35	4

メモリ

命令のイメージ

レジスタGR4の内容と
メモリの33番地の内容を足して
レジスタGR4に書き戻せ

GR3	9
GR4	5
GR5	3

+

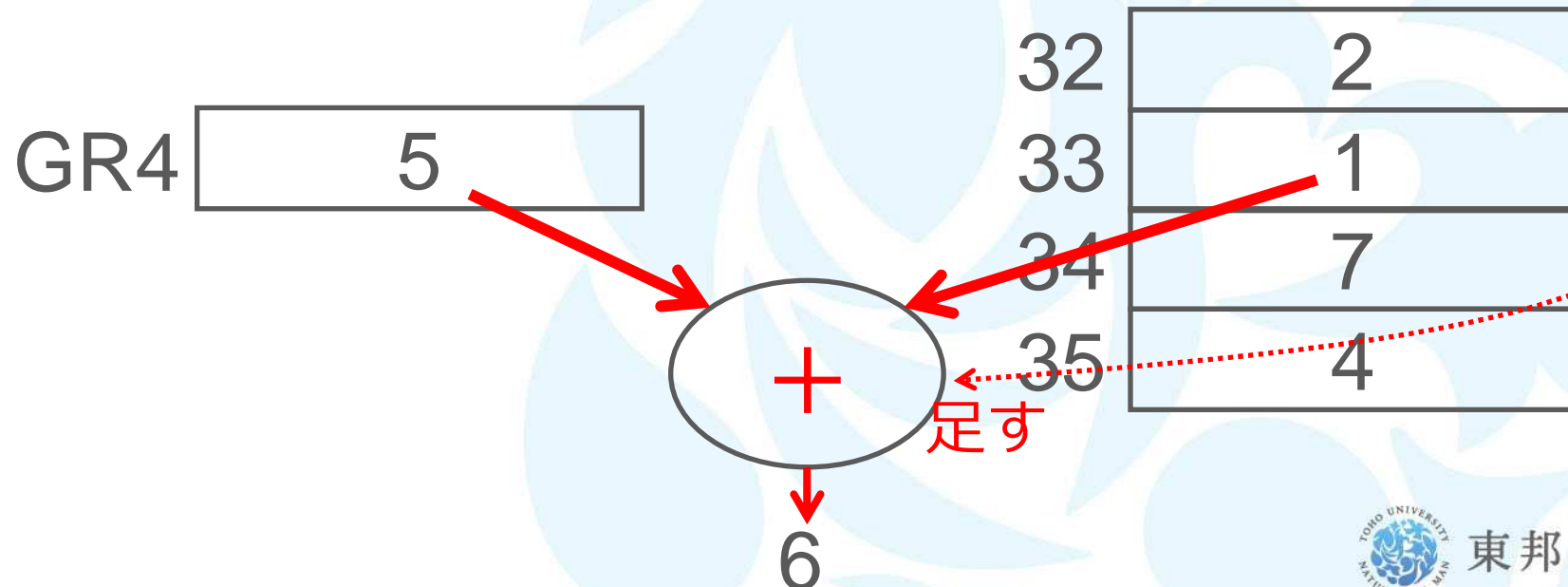
32	2
33	1
34	7
35	4

メモリ



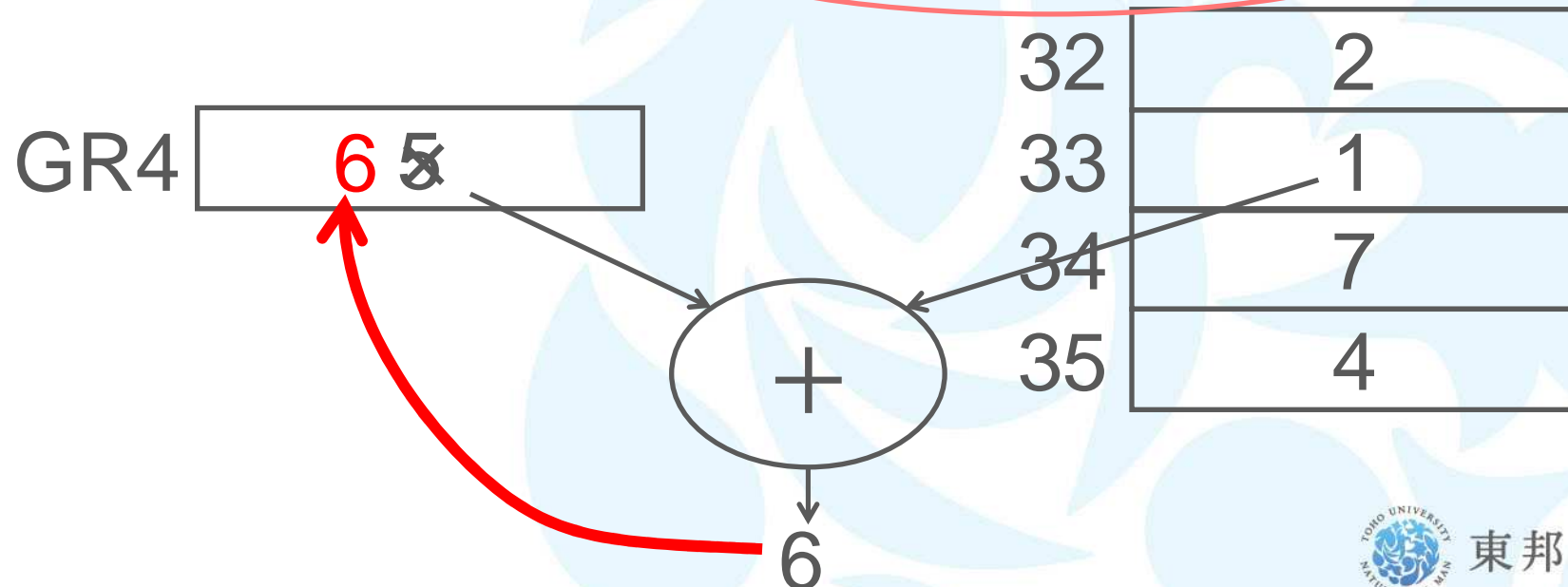
命令のイメージ

レジスタGR4の内容と
メモリの33番地の内容を足して
レジスタGR4に書き戻せ



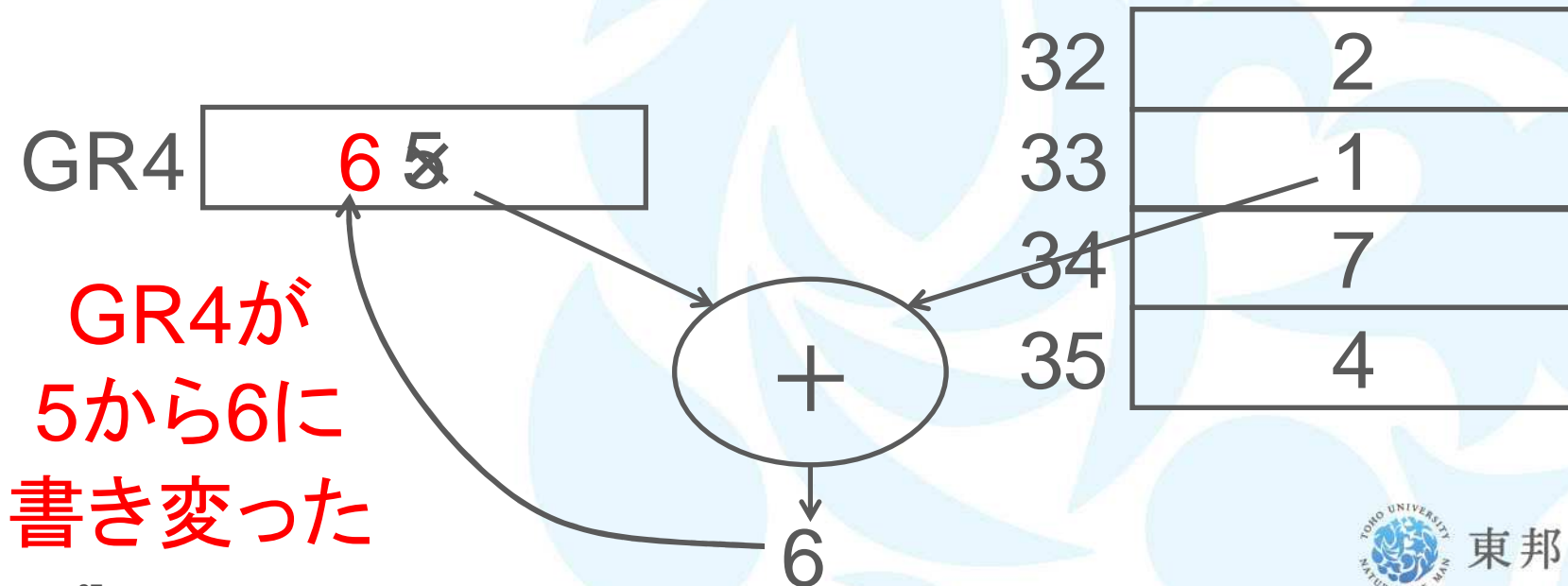
命令のイメージ

レジスタGR4の内容と
メモリの33番地の内容を足して
レジスタGR4に書き戻せ



命令のイメージ

レジスタGR4の内容と
メモリの33番地の内容を足して
レジスタGR4に書き戻せ



ところで
命令のいろいろ

命令のいろいろ どんな種類の命令があるのか

どんな種類の命令があるのか

2方面で
考えたい

1つのCPUの中で
どんな種類の命令？

いろいろなCPUの
中でどんな命令？

どんな種類の命令があるのか

2 方面で
考えたい

1 つのCPUの中で
どんな種類の命令？

いろいろな命令があるぞ

いろいろなCPUの
中でどんな命令？

どんな種類の命令があるのか

2 方面で
考えたい

1 つのCPUの中で
どんな種類の命令？

いろいろな命令があるぞ

いろいろなCPUの
中でどんな命令？

CPUによって命令の種類
や組合せが違ふぞ

1つのCPUの中でどんな命令？

>前に見たのは足し算

1つのCPUの中でどんな命令？

>前に見たのは足し算 $\Rightarrow + - \times \div$

1つのCPUの中でどんな命令？

>前に見たのは足し算 $\Rightarrow + - \times \div$

>レジスタ \Leftrightarrow メモリ間の転送

レジスタに値をセットできないと困るし

1つのCPUの中でどんな命令？

>前に見たのは足し算 $\Rightarrow + - \times \div$

>レジスタ \Leftrightarrow メモリ間の転送

あとは？

1つのCPUの中でどんな命令？

>前に見たのは足し算 $\Rightarrow + - \times \div$

>レジスタ \Leftrightarrow メモリ間の転送

あとは？

if 文とかは？

1つのCPUの中でどんな命令？

>前に見たのは足し算 $\Rightarrow + - \times \div$

>レジスタ \Leftrightarrow メモリ間の転送

>条件分岐 \leftarrow if 文, for文の判定に

1つのCPUの中でどんな命令？

>前に見たのは足し算 $\Rightarrow + - \times \div$

>レジスタ \Leftrightarrow メモリ間の転送

>条件分岐 \leftarrow if 文, for文の判定に

>入出力の制御 \leftarrow 読んだり書いたり

1つのCPUの中でどんな命令？

>前に見たのは足し算 $\Rightarrow + - \times \div$

>レジスタ \Leftrightarrow メモリ間の転送

>条件分岐 \leftarrow if 文, for文の判定に

>入出力の制御 \leftarrow 読んだり書いたり

その他いろいろ \sim CPU種類によって



CPU 「COMET II」 での例



東邦大学

CPU 「COMET II」 での例

聞いたこと[↑]無い！

実は



東邦大学

CPU「COMET II」での例

聞いたこと無い！

実は

情報処理技術者試験でのモデルCPU
単純なので、理解がしやすい

CPU「COMET II」での例

聞いたこと無い！

実は

情報処理技術者試験でのモデルCPU
単純なので、理解がしやすい

実用されているIntel Core i3/5/7などは
命令数が非常に多くて複雑 ⇒ 分かりにくい
COMET II は単純すぎて実用化には不足



CPU「COMET II」での例

ロード・ストア命令

ロード	LD
ストア	ST
ロードアドレス	LAD

演算命令

算術加算	ADDA
論理加算	ADDL
算術減算	SUBA
論議減算	SUBL
論理積	AND
論理和	OR
排他的論理和	XOR

比較命令

算術比較	CPA
論理比較	CPL

シフト命令

算術左シフト	SLA
算術右シフト	SRA
論理左シフト	SLL
論理右シフト	SRL

スタック命令

プッシュ	PUSH
ポップ	POP

分岐命令

正分岐	JPL
負分岐	JMI
非零分岐	JNZ
零分岐	JZE
オーバー フロー分岐	JOV
無条件分岐	JUMP

その他

コール	CALL
リターン	RET
SVC	SVC
NOP	NOP

参照 [情報処理技術者試験試験で使用する情報技術に関する用語・プログラム言語など](#)  東邦大学

では
いろいろなCPUの中でどんな命令？



東邦大学

いろいろなCPUの中でどんな命令？

- > 前に見たのは COMET II
モデル（実際は存在しない）CPUで
命令数が少なく、それぞれ簡単な動作

いろいろなCPUの中でどんな命令？

> 前に見たのは COMET II

> 周りのパソコンで使われる

Core 3i/5i/7i など (通称 x86系)

実用規模。非常に複雑な命令体系で
命令数も多く、複雑な動作の命令がある

いろいろなCPUの中でどんな命令？

> 前に見たのは COMET II

> 周りのパソコンで使われる

Core 3i/5i/7i など (通称 x86系)

> サーバーで使われるPower CPU

> スマホでよく使われるARM CPU

いろいろなCPUの中でどんな命令？

命令体系の比較

いろいろなCPUの中でどんな命令？

命令体系の比較

⇒ どんな組合せになっているか

⇒ CISCとRISCへ続く

まとめ

命令 = instruction 教える・指示する

コンピュータの命令のイメージ：

レジスタGR4の内容と

メモリ33番地の内容を足してGR4に書き戻せ

コンピュータの命令のイメージ：

加減乗除

レジスタ \leftrightarrow メモリ間の転送

条件分岐

入出力の制御

その他いろいろ

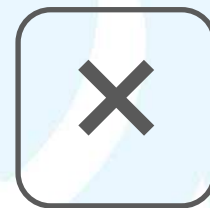
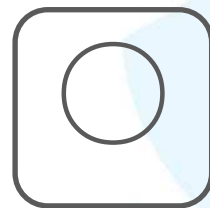
まとめ

どんな命令があるかは
CPUの種類によって違う
命令の種類数もCPUの種類による

でも
命令のイメージはまあ似ている

加減乗除のような演算命令
レジスタやメモリ間の転送命令
条件判定とそれによる分岐 (if文に対応)
入出力の制御
など

命令の考え方について
分かりましたか？



次へ

(ノイマン型コンピュータ)

ノイマン型コンピュータとは

ノイマン型コンピュータとは
コンピュータのタイプの1つなのだが

ノイマン型コンピュータとは
コンピュータのタイプの1つなのだが

現代のデジタルコンピュータは
ほとんど全部ノイマン型

ノイマン型コンピュータとは
コンピュータのタイプの1つなのだが

現代のデジタルコンピュータは
ほとんど全部ノイマン型

つまり

(デジタル)コンピュータ = ノイマン型

で
ノイマン型コンピュータとは
どんなものか？

ノイマン型コンピュータとは

特徴（≈条件）

1. プログラム(可変)内蔵方式
プログラムを内部のメモリに記憶させる

ノイマン型コンピュータとは

特徴（≈条件）

1. プログラム(可変)内蔵方式

プログラムを内部のメモリに記憶させる

2. 逐次処理方式

プログラム内の命令を上から順に実行する

ノイマン型コンピュータとは

特徴（≈条件）

1. プログラム(可変)内蔵方式

プログラムを内部のメモリに記憶させる

2. 逐次処理方式

プログラム内の命令を上から順に実行する

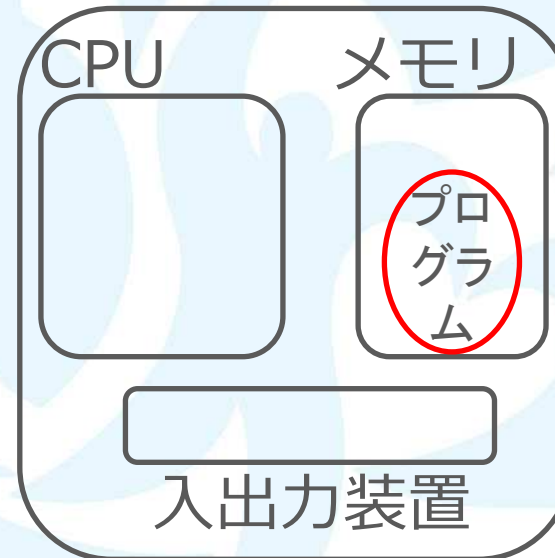
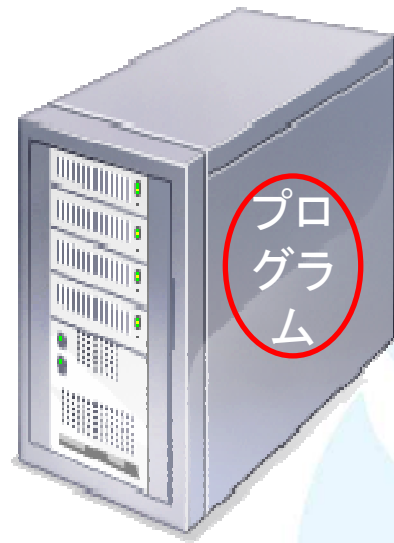
3. 単一メモリ方式

プログラムとデータを同じメモリに格納する



プログラム(可変)内蔵方式

プログラムを内部のメモリに記憶させる



プログラム(可変)内蔵方式

プログラムを内部のメモリに記憶させる

1. プログラムが取替えられる

2. 処理手順を高速に読出して処理

プログラム(可変)内蔵方式

プログラムを内部のメモリに記憶させる

1. プログラムが取替えられる

ある時はワープロ、ある時は表計算のように、
プログラムを入換えて、万能の箱として使える

単能(専用)の箱ではなく、万能の箱

プログラム(可変)内蔵方式

プログラムを内部のメモリに記憶させる

1. プログラムが取替えられる

ある時はワープロ、ある時は表計算のように、
プログラムを入換えて、万能の箱として使える

単能(専用)の箱ではなく、万能の箱

その代わりに、プログラム(ソフト)が無ければ
全く無能な「ただの箱」になってしまう



プログラム(可変)内蔵方式

プログラムを内部のメモリに記憶させる

1. プログラムが取替えられる

ある時はワープロ、ある時は表計算のように
プログラムを入換えて、万能の箱として使える

2. 処理手順を高速に読出して処理

手順(=プログラム)を高速に読出して、
高速に処理できる

プログラム(可変)内蔵方式

プログラムを内部のメモリに記憶させる

1. プログラムが取替えられる

ある時はワープロ、ある時は表計算のように
プログラムを入換えて、万能の箱として使える

2. 処理手順を高速に読出して処理

手順(=プログラム)を高速に読出して、
高速に処理できる

電卓 (=手で処理手順を入力) だと遅いが
コンピュータは自動的に手順を読んで早い



東邦大学

プログラム(可変)内蔵方式

プログラムを内部のメモリに記憶させる

1. プログラムが取替えられる

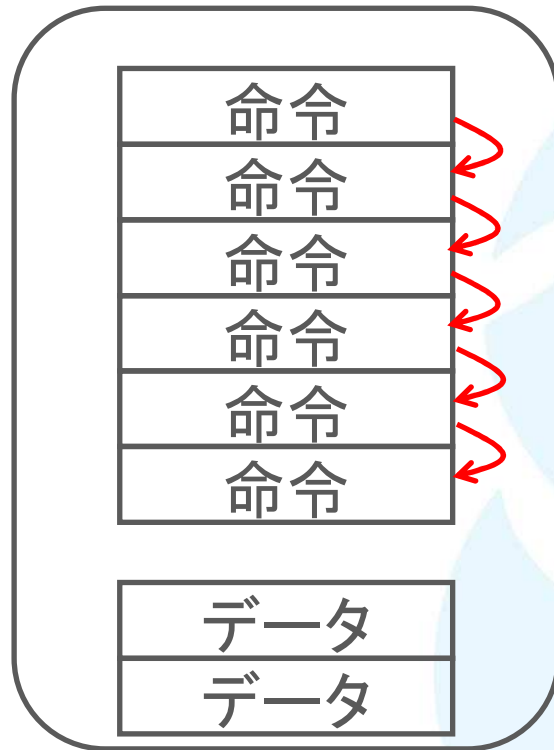
ある時はワープロ、ある時は表計算のように、プログラムを入換えて、万能の箱として使える

2. 処理手順を高速に読出して処理

手順(=プログラム)を高速に読出して、高速に処理できる

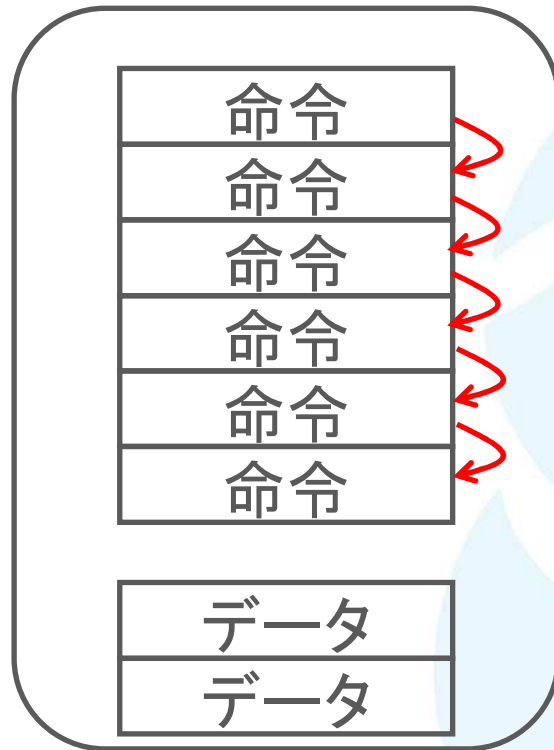
逐次処理方式

プログラム内の命令を上から順に実行する
プログラム



逐次処理方式

プログラム内の命令を上から順に実行する
プログラム



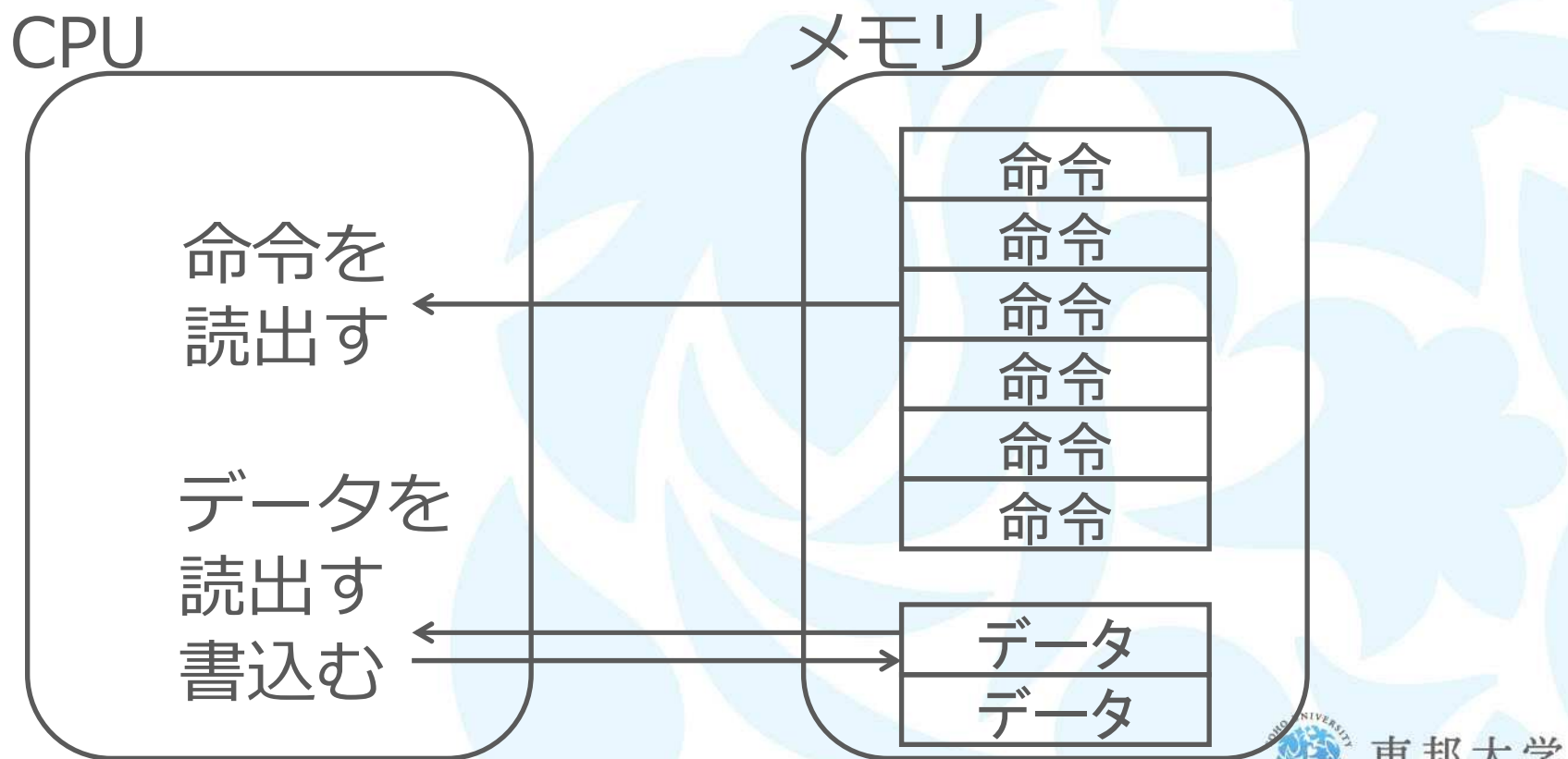
「現在実行している命令」
(を示すレジスタ PC)
を +1 して次の命令へ進む



次がどの命令かを、
命令中に明示しなくて済む

単一メモリ方式

命令とデータを同じメモリに格納する



単一メモリ方式

プログラムとデータを同じメモリに格納

メモリ領域を効率的に利用できる

命令部分とデータ部分の量の大小に依らず
全体がメモリに収まればOK

まとめると ノイマン型コンピュータとは 特徴（≈条件）

1. プログラム(可変)内蔵方式

プログラムを内部のメモリに記憶させる

2. 逐次処理方式

プログラム内の命令を上から順に実行する

3. 単一メモリ方式

プログラムとデータを同じメモリに格納する



確認の問題です

ノイマン式コンピュータの特徴は

1.

2.

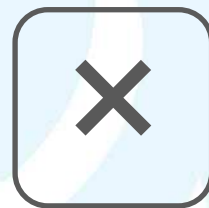
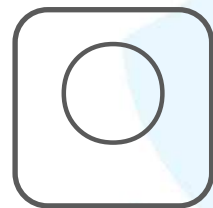
3.

確認の問題です

ノイマン式コンピュータの特徴は

1. プログラム(可変)内蔵方式
2. 逐次処理方式
3. 単一メモリ方式

「ノイマン型」の考え方について
分かりましたか？



次へ
(CISCとRISCへ)