

命令の形式



復習 「命令とは？」

- コンピュータの動作を指示する
 - プログラムより細かい単位の指示
- メインメモリの中に置かれる
- 「読出し・解釈・実行・次へ」の4ステップで実行される
 - 命令を1つずつ、このステップで実行
 - この4ステップはハードウェアで実現



命令はメインメモリ中に置かれる

どのような形で？

⇒

2進数で書かれている

そうでないとメモリに置けない

いくつかの部分(フィールド)から成る

(1つ1つの) 命令の形

命令コード	オペランド
-------	-------

(1つ1つの) 命令の形

命令コード

オペランド

何を (どんな処理を)
行うのか

(1つ1つの) 命令の形

命令コード

オペランド

何を (どんな処理を)
行うのかを指定

演算をする

ロード・ストアする

ジャンプする など

(1つ1つの) 命令の形

命令コード

オペランド

どこのデータに対して
処理を行うのかの指定

(1つ1つの) 命令の形

命令コード

オペランド

どこのデータに対して
処理を行うのかの指定

レジスタRに対して
メモリM番地に対して
定数Cに対して

(1つ1つの) 命令の形

命令コード | オペランド1 | オペランド2

オペランドは
複数指定することもある

① Reg R と ② メモリM番地を
足して Reg R へ入れる



8

命令の形のまとめ

1つの命令は、 と
から成っている

は、
かを指定する

は、
かを指定する



9

命令の形のまとめ

1つの命令は、命令コードとオペランドから成っている

命令コードは、

かを指定する

オペランドは、

かを指定する

命令の形のまとめ

1つの命令は、命令コードとオペランドから成っている

命令コードは、どのような処理をするかを指定する

オペランドは、処理の対象が何であることを指定する

いろいろなオペランド数

3-オペランド方式

2-オペランド方式

1-オペランド方式

0-オペランド方式

12

3-オペランド方式

$X \oplus Y \Rightarrow Z$ のうち

X, Y, Z をすべて別々に
指定する

命令コード	オペランド1	オペランド2	オペランド3
-------	--------	--------	--------

← 長くなる →

⇒ メモリを食う

13

2 – オペランド方式

$X \oplus Y \Rightarrow X$ のように
入力と出力を重ねる
(X を同じところにする)

X に元々あった値は消えてしまう

1 – オペランド方式

$A \oplus Y \Rightarrow A$ とし
 A は特定の決まったところにする
 A : アキュムレータ


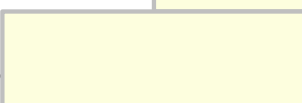
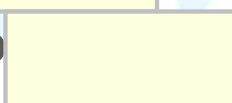
A の内容を毎度メモリに出入れ
する必要がある


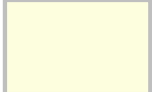
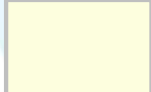


0 – オペランド方式

メモリの場所指定を、
アドレスでなく、スタック
の頂上と指定する

スタック頂上から2要素を取出し
それらを演算して
結果をスタックの頂上に積む
かなり特殊なケース




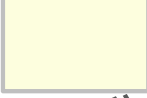

オペランドをまとめると

オペランドとは  のことで
具体的には  や  がある

オペランドの  はいろいろあって
 オペランド、  オペランド、
 オペランド、  オペランド
などの方式がある


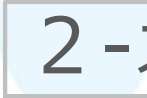


オペランドをまとめると

オペランドとは処理の対象のことで
具体的にはレジスタやメモリがある

オペランドの  はいろいろあって
 オペランド、  オペランド、
 オペランド、  オペランド
などの方式がある

オペランドをまとめると

オペランドとは処理の対象のことで
具体的にはレジスタやメモリがある

オペランドの個数はいろいろあって
 3 - オペランド、  2 - オペランド、
 1 - オペランド、  0 - オペランド
などの方式がある

オペランド指定のビット数？

20

オペランド指定のビット数？

オペランドとして
レジスタ（の番号）を指定

メモリ（のアドレス）を指定

21

オペランド指定のビット数？

オペランドとして

レジスタの番号を指定するには
 $\log_2(\text{レジスタ個数})$ ビット

22

オペランド指定のビット数？

オペランドとして

レジスタの番号を指定するには
 $\log_2(\text{レジスタ個数})$ ビット

たとえば、レジスタが8個あれば

$\log_2(8) = 3$ ビット必要

000番, 001番, ..., 111番 の8個

23

オペランド指定のビット数？

オペランドとして

レジスタの番号を指定するには
 $\log_2(\text{レジスタ個数})$ ビット

メモリのアドレスを指定するには
 $\log_2(\text{メモリの量})$ ビット

オペランド指定のビット数？

オペランドとして

レジスタの番号を指定するには
 $\log_2(\text{レジスタ個数})$ ビット

メモリのアドレスを指定するには
 $\log_2(\text{メモリの量})$ ビット

たとえば、メモリが 2^{16} 個であれば
 $\log_2(2^{16}) = 16$ ビット必要

オペランド指定のビット数？

COMET II の場合

メモリ指定 \Rightarrow 容量は 2^{16} 個なので

$$\log_2(2^{16}) = 16 \text{ビット必要}$$

レジスタ指定 \Rightarrow レジスタ数は8 個で

$$\log_2(8) = 3 \text{ビット必要}$$

2 オペランド $\text{Reg} \oplus \text{メモリ} \Rightarrow \text{Reg}$
ならば、 $3 + 16 = 19$ ビット必要

26



命令の形とオペランドの個数について
分かりましたか？



↓
次へ

27



話変わって 命令のエンコード

28



命令は2進数で書く

29



命令は2進数で書く

メモリに蓄えるのに
2進数でなければ置けない

30

命令は2進数で書く

メモリに蓄えるのに
2進数でなければ置けない

ロード, ストア, 加算, 減算,
などすべて2進数に置換える

31

命令は2進数で書く

メモリに蓄えるのに
2進数でなければ置けない

ロード, ストア, 加算, 減算,
などすべて2進数に置換える

たとえば (COMET IIの場合)

00010000 ロード 00010001 ストア

00100000 加算 00100001 減算

01100001 マイナスの時ジャンプ

01100101 プラスの時ジャンプ



命令は2進数で書く

2進数だと覚えづらい



命令は2進数で書く

2進数だと覚えづらい
名前(略号)で書く

34



命令は2進数で書く

2進数だと覚えづらい
名前(略号)で書く

たとえば (COMET IIの場合)

ロード 00010000 ⇒ LD

ストア 00010001 ⇒ ST

加算 00100000 ⇒ ADDA

減算 00100001 ⇒ SUBA

マイナスの時ジャンプ 01100001 ⇒ JMI

プラスの時ジャンプ 01100101 ⇒ JPL

35



命令は2進数で書く

2進数だと覚えづらい
名前(略号)で書く

たとえば (COMET IIの場合)

ロード	00010000	⇒	LD	この略号を ニーモニック コードと呼ぶ
ストア	00010001	⇒	ST	
加算	00100000	⇒	ADDA	
減算	00100001	⇒	SUBA	
マイナスの時ジャンプ	01100001	⇒	JMI	
プラスの時ジャンプ	01100101	⇒	JPL	

36



東邦大学

命令のエンコーディングをまとめると

命令は に置かれ、 で書かれている

読みにくいので、LD(ロード)やST(ストア)などの で書くことが多い

37



東邦大学

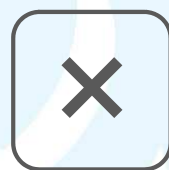
命令のエンコーディングをまとめると

命令はメモリ上に置かれ、2進数で書かれている

読みにくいので、LD(ロード)やST(ストア)などのニーモニックコードで書くことが多い

38

命令のエンコーディングが分かりましたか？



↓
次へ

39