

計算機方式論

第3章 命令形式と CISC/RISC

1

命令語の構成

命令コード	アドレス部
	AR 0,1
	AR 0 1

- **命令コード**
命令の動作の種類を指定
- **アドレス部 (オペランド)**
命令対象となるデータの指定
主記憶のアドレス
レジスタのアドレス
これ自身がデータ(即値, immediate address)
+
インデックス修飾指定
ベース修飾指定
間接修飾指定

S	2	512	3
S	2	512	3

2

アドレス部の数: オペランド方式

3オペランド方式 A op B → C

OP	A	B	C
----	---	---	---

4オペランド方式 A op B → C, D: 次の番地

OP	A	B	C	D
----	---	---	---	---

2オペランド方式 A op B → A

OP	A	B
----	---	---

1オペランド方式 A op Acc → Acc : Accumulator, 累算器

OP	A
----	---

0オペランド方式 専用のプッシュダウンスタック上の
データに対し演算 op を施す

OP

3

命令語の長さ

固定語長

すべての命令の長さが同じ
命令語の構成が簡単
命令語の種類でビットが余る

可変語長

各命令で長さが異なる
平均的に命令語長を短くできる
長さの解析にデコード時間がかかる

4

命令の種類

一般命令: 一般利用者が使える命令

単純命令

演算命令

四則演算... 加減乗除等

論理演算... ビットごとの演算

分岐命令

無条件分岐命令

条件分岐命令

データ転送命令

ロード・ストア命令

主記憶間データ転送命令

プロセス同期命令

スーパーバイザコール命令

複合命令

変換命令

特権命令: 制御プログラムが使う命令

入出力命令

構成制御命令

5

一般命令

単純命令

演算命令

四則演算... 加減乗除等

データの格納場所:

レジスタ

主記憶

AR 0 2

レジスタ加算命令

AR 0,2 (GR0)+(GR2)→GR0

S 2 512 3

主記憶読出(メモリ)減算命令

S 2,512(3) (GR2)-(512+(GR3))→GR2

6

一般命令

単純命令

演算命令

四則演算... 加減乗除等

データの種類(表現の種類):

固定小数点と浮動小数点

2進と10進

固定長と可変長

16ビットデータの解釈例

0001000110010011

指数部

仮数部

表す数

浮動小数点数

0001000110010011

$403 \times 2^2 = 1612$

固定小数点2進数

0001000110010011

4494

固定小数点10進数

0001000110010011

1193

7

一般命令

単純命令

演算命令

四則演算

論理演算

ビットごとのブール代数演算

0 1 0 1 → NOT → 1 0 1 0

OP) 0 0 1 1

AND 0 0 0 1

OR 0 1 1 1 包括的論理和

XOR 0 1 1 0 排他的論理和

8

一般命令

単純命令

演算命令

四則演算

論理演算

ビットごとのブール代数演算

シフト, ローテート演算

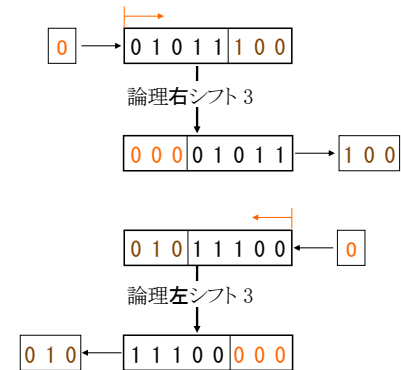
論理シフト…左と右

算術シフト…左と右

論理ローテート…左と右

9

• 論理シフト



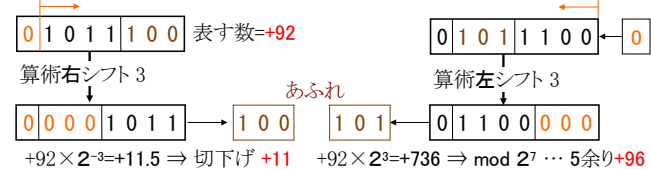
10

• 算術シフト … 符号を考慮したシフト

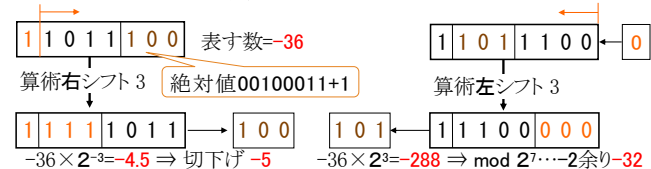
表す数を、あふれ分を考慮して

左シフトnならば 2^n 、右シフトnならば 2^{-n} 倍する

符号ビット: 0

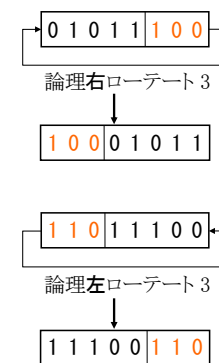


符号ビット: 1 … 2の補数表現で負数を表現



11

• 論理ローテート



12

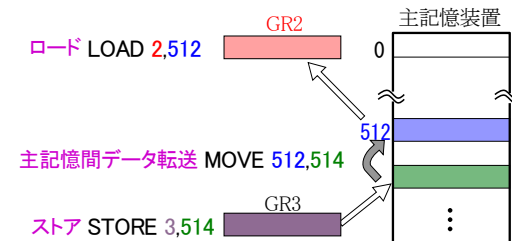
一般命令
単純命令
演算命令
分岐命令
無条件分岐
条件分岐

BZ 128 1 条件分岐命令
BZ 128(1) 直前に実行された命令の結果で、
ゼロフラグonならば128+(GR1)番地へ分岐
さもなければ、NOP(次の番地へ)

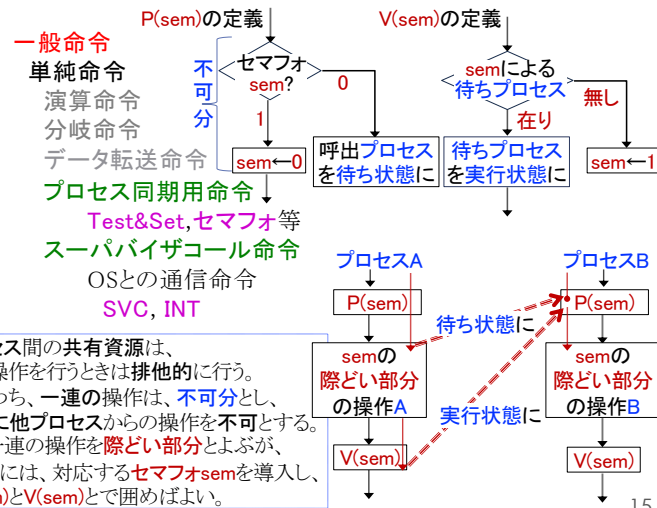
B 128 無条件分岐命令
B 128 128番地へ分岐

13

一般命令
単純命令
演算命令
分岐命令
データ転送命令
ロード・ストア命令
主記憶間データ転送命令



14



15

一般命令
単純命令
演算命令
分岐命令
データ転送命令
複合命令
単純命令をいくつか組み合わせた命令
フェッチサイクルの省略
変換命令 …EBCDICコードをJISコードに変換

【参考】
より進んだアーキテクチャに
VLIW (Very Long Instruction Word)がある。
1命令語中に、ロード・ストア、演算、分岐などの同時に実
行可能な複数の命令をコンパイラレベルで納める。
複数のプロセッサによってパイプラインで並列実行する。
Intel Itaniumで採用 (Intel IA-64アーキテクチャ)。

16

一般命令
単純命令
複合命令

特権命令

入出力命令: 入出力装置は共有資源のため

入出力開始命令 SIO

構成制御命令

停止命令 HALT

PSWロード命令 LPSW

オペレーティングシステムが使う命令
直接に一般利用者プログラムが使えない

17

CISCとRISC

命令という観点から
見たコンピュータ

- 命令セットアーキテクチャの複雑化… Intel, AMD : CISC
コンピュータのファミリー化による上位互換性 ARM, A, M : RISC
プログラミング言語レベルに命令を高機能化



CISC(Complex Instruction Set Computer)

18

CISCの欠点

- 複雑になった命令セット:
実際に使われるものがそれ程多くない
高機能命令も要求される機能と一致しない
- 命令セットの複雑化
⇒ CPUの制御部が複雑⇒ デコードに時間がかかる
- CISCは布線論理で実現すると、開発コストがかかる
のでマイクロプログラムで実現する ⇒ 速度が遅い

[例] モトローラのM68000, M68020, M68030, M68040
M68020は、M68000に比べて、命令数が2倍以上、アドレッシング法が11多い

[例] インテルの8086, 80286 (16ビットの命令セット),
80386, Pentium (32ビットのIA-32(X86)命令セット)

19

RISC

- 命令セットの単純化



RISC(Reduced Instruction Set Computer)

[例] IBMの801
カリフォルニア大のRISC I, RISC II
スタンフォード大のMIPS
Apple, IBM, モトローラのPowerPC
サン・マイクロシステムズのSPARC
ARM, Apple A, M

20

RISCの特長

- 特長

単一サイクル操作 (1演算/クロックのためパイプライン処理に適す)

ロード／ストア設計 (ロード/ストア以外メモリアクセス命令がない)

少数命令 (命令数が少ない)

固定長命令形式

布線論理(wired logic) (高速制御)

[例] RISC I : 1981年 31命令 2アドレッシング法
M68000: 1980年 61命令 14アドレッシング法

21

その後

- CISC

インテルは、IA-32 (32ビット拡張X86) 命令セットの80386, 80486, Pentiumや、Intel 64 (X86-64) 命令セットのCore 2, Core i7等では、マシン命令をプリフェッチし、制御記憶からマイクロ命令(μOPs)を予め読み出すことで、マイクロ命令の実質的フェッチ時間を無視でき、効率を上げている。

- RISC

ARMもファミリー化が進み、命令の追加等で、その長所が薄れてきている。

22