

# 命令セット

## 機械語とアセンブリ言語

# 命令・命令セット

- 命令 (instruction)
  - コンピュータの動作を指示するもの。コンピュータは命令にしたがって実行する(命令の実行)
- 命令セット (instruction set)
  - コンピュータのすべての命令の集まり

# 命令の種類

- 算術論理演算命令
  - 加算、減算、乗算、除算、剰余、絶対値など
  - 論理積、論理和、否定など
- データ転送命令
  - レジスタ間、メモリとレジスタ間、メモリと入出力機器間
- 分岐命令
  - 無条件分岐命令、条件分岐命令

# 命令・機械語・アセンブリ言語

- 命令は2進数(ビット列)で符号化され、1つの命令は1語の命令メモリに格納されている
  - 1語はCPUが扱うデータの最小単位。通常32ビット(4バイト)か64ビット(8バイト)。CPUに依存
- 1命令を1命令語とも呼ぶ
- このようなビット列で「命令」(「命令語」)を表す言語を「**機械語**」と呼ぶ
  - CPUという機械が理解できる言語(CPUという機械向けの言語) = 機械語
- 機械語は人間が理解できない⇒アセンブリ言語
  - 人間が読める、英語に近い表記
  - 機械語と1対1対応(疑似命令などで例外はある)

# MIPS32アーキテクチャ(1/2)

- ミップス・コンピュータシステムズ(現ミップス・テクノロジーズ)が開発した32ビットマイクロプロセッサの命令セットアーキテクチャである
- 命令セットが非常にきれいなので、米国ではコンピュータ・アーキテクチャを学校で教えるときに教材としてMIPSアーキテクチャを使うことが多い
- 対して、x86, x64アーキテクチャ(インテル社CPUなど)の命令セットは複雑で、教材としては適さない

# MIPS32アーキテクチャ(2/2)

- ワークステーション: MIPSプロセッサを使ったワークステーションシステムを製造していた企業として、SGI、DEC、NEC、ソニーなどがあつた
- OS分野: MIPSアーキテクチャ上に移植されたオペレーティングシステムとして、Windows CE、Linux、BSDなどがあつた
- 機器組み込み分野: コンピュータネットワーク(CISCOのルーター)、ゲーム機(NINTENDO64・ソニーのPlayStation)、プリンター、デジタルテレビ、DVDレコーダ、携帯情報端末などに広く採用されている

# 言語間の関係

Cなど高水準言語

```
y=x;
```

コンパイラ

命令セット

機械語 (低水準言語、CPUへの命令)

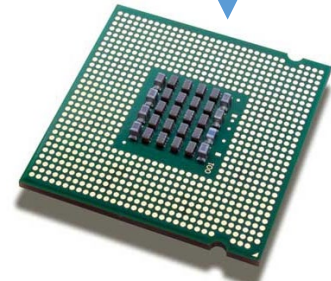
```
00111011000010000001000000000001  
10001100000100000000000000000000  
00111011000000010001000000000001  
00110100001010010000000000000100  
10101101001100000000000000000000
```

cc -S \*\*\*.c

アセンブラ

アセンブリ言語  
(低水準言語)

```
la $t0, x  
lw $s0, 0($t0)  
la $t1, y  
sw $s0, 0($t0)
```



# C言語・アセンブリ言語・機械語 の具体例

C言語	アセンブリ言語(疑似命令あり)	実際の命令	機械語(16進数で表記。実際は二進数)
<pre>y=x; //変数代入</pre>	<pre>la \$t0, x      #アドレスをロード lw \$s0, 0(\$t0) #語をロード la \$t1, y      #アドレスをロード sw \$s0, 0(\$t1) #語をストア</pre>	<pre>lui \$8, 4097 lw \$16, 0(\$8) lui \$1, 4097 ori \$9, \$1, 4 sw \$16, 0(\$9)</pre>	<pre>0x3C081001 0x8D100000 0x3C011001 0x34290004 0xAD300000</pre>

lui \$8, 4097

00111100000001000000100000000001

司令部 レジスタレジスタ 即値(定数)

\$zero \$t0(\$8)

オペランド

命令語 \$zero+即値4097→\$t0 (の上位16ビットに)



# 命令形式 (Instruction Format)

- 命令語は32ビットの固定長
- Rタイプ, Iタイプ, Jタイプの3種類に分類

type	フォーマット(ビット数)					
R	命令部 (6)	レジスタ (5)	レジスタ (5)	レジスタ (5)	シフト量 (5)	機能(6)
I	命令部 (6)	レジスタ (5)	レジスタ (5)	即値(オフセット)(16)		
J	命令部 (6)	アドレス(26)				

# 命令部と機能部

- 命令部は命令の大分類:たとえば演算命令
- 機能部はRタイプの命令の詳細分類:たとえば「加算」「減算」など
- 機能部のない命令は命令部で細かく分類され、機能部のある命令は命令部は大まかでまたは極端な場合1つのみで結構

# ビット数の一般的な決め方

- $n$ ビットが $2^n$ 通り表現できる
- 命令セット中の命令数で命令部(+機能部)の最小ビット数を決める
- レジスタの数でレジスタの最小ビット数を決める
- 残りのフィールドのビット数も必要に応じてその最小ビット数を決める
- 命令語のビット数は固定なので、上記すべてを満足するように、設計を行う

# manaba小テスト:演習13-1

- 10分
- 5点

# 補足

- 命令数やレジスタの数は必ず2のべき乗でない場合も十分ありうる。この場合は余裕をもって設計するしかない
  - たとえば命令が34個なら、6ビット以上を設ける必要がある
  - 上記演習のような、最大最小だけ考えればよい、というわけではない
- 命令は複数のフィールド(以下の例)からなるので、ビット数の設計時、総合的に考慮する必要がある

命令部 (?bit)	レジスタ (?bit)	レジスタ (?bit)	即値(オフセット) (?bit)
---------------	----------------	----------------	---------------------

# 期末定期試験について

- 内容的には、manaba小テスト、manabaレポート課題、授業中の演習・質問(Question)が中心。ただし、問題の出し方はこれらと一致しない(つまりこれらをそのまま使わない)ので、注意してください
- 以下の内容からは出題しない
  - 順序回路(ラッチ、フリップフロップ、カウンタ)