

プロセッサと機械語

2年次前期（第1回）

中島 克人

情報メディア学科

nakajima@im.dendai.ac.jp

1

授業の概要(1/2)

- 目的
 - プロセッサの基本動作を理解する。
 - アセンブリ言語によって簡単な処理アルゴリズムを記述できるようになること。
- 達成目標
 - プロセッサの基本的な構成要素、各種情報の内部表現法、動作のメカニズム等を理解する。アセンブリ言語によるプログラミングを通じて、機械語やアセンブラの役割、および、アセンブラによる基本的な計算アルゴリズムの構成法を理解する。
- 教科書: 玉井浩著「詳解 COMET II & CASL II」
(発行SiB access社、発売 星雲社)
- 参考書: 坂井修一著「コンピュータアーキテクチャ」
(コロナ社)(後続科目の参考書としても利用)
- 評価方法
 - 3回程度の提出課題、および、期末試験を1:1で評価

2

授業の概要(2/2)

- 演習について
 - 講義中の演習は、紙と筆記用具で、もしくは、
各自のノートPCで、仮想計算機COMET IIのアセンブラであるCASL IIシミュレータを用いて行う
 - CASL IIシミュレータは下記、講義ホームページからインストールできる
- 講義ホームページ
<http://www.mlab.im.dendai.ac.jp/~nakajima/proc-ml/>
- 講義の一部を欠席すると、直ぐに付いて行けなくなる恐れが大きい
 - 遅刻・欠席はしないこと
 - 居眠りは欠席と同じ

3

プロセッサと機械語

計算機のしくみと 計算機の内部表現

- プロセッサの中には？
- 計算機の構成例
- プロセッサの基本動作
- プログラムと機械命令
- 計算機の内部表現
- 計算機の構成

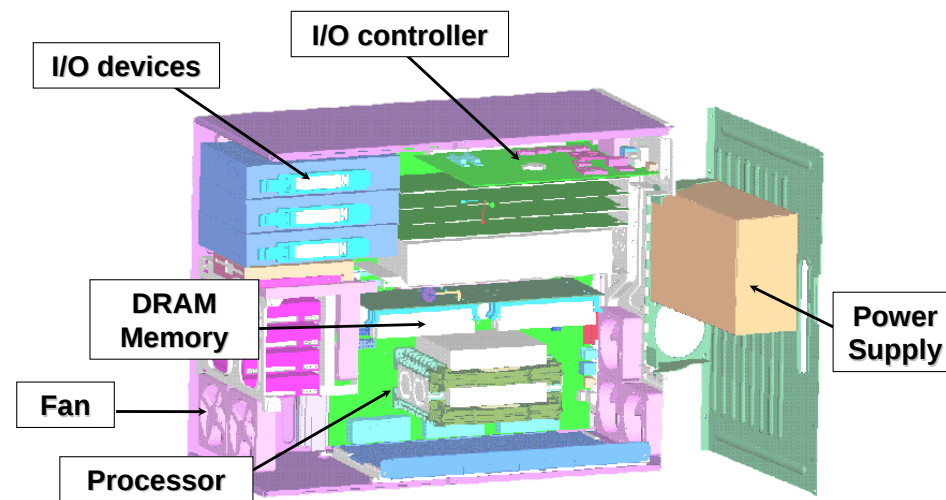
4

まずはコンピュータ(計算機)を
分解してみよう！



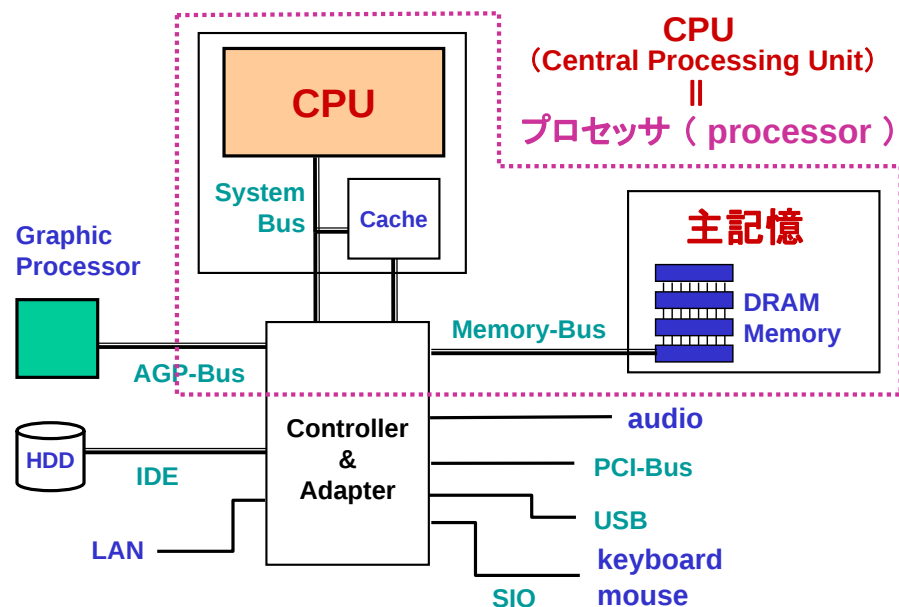
5

本体Cabinetの中を覗くと・・・
・・ 計算機の (物理) 構成例



6

計算機の (論理) 構成例

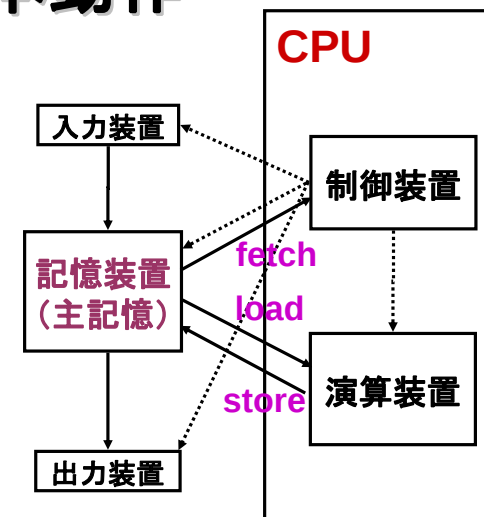


7

プロセッサの基本動作

●プログラム格納型 (フォン・ノイマン型)

- 主記憶にプログラム (機械命令、データ) を置いておく
- CPUが機械命令を主記憶からfetchし解釈・実行
- 演算の入力データは主記憶からloadし、結果は主記憶にstore



..... : 制御の流れ
—— : データの流れ

8

プログラムと機械命令

高水準プログラミング言語 (C言語)

```
swap (int v[ ], int k)
{
    int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}
```

アセンブリ言語

```
swap:
    muli $2, $5, 4
    add  $2, $4, $2
    lw   $15, 0($2)
    lw   $16, 4($2)
    sw   $16, 0($2)
    sw   $15, 4($2)
    jr   $31
```

assembler

機械命令(機械語)

```
0000000010100001000000...
0000000010001110000110...
1000110001100010000000...
1000110011110010000000...
1010110011110010000000...
1010110001100010000000...
0000001111100000000000 ...
```

compiler

9

機械命令(machine instruction)

- ## ● 機械命令(機械語)

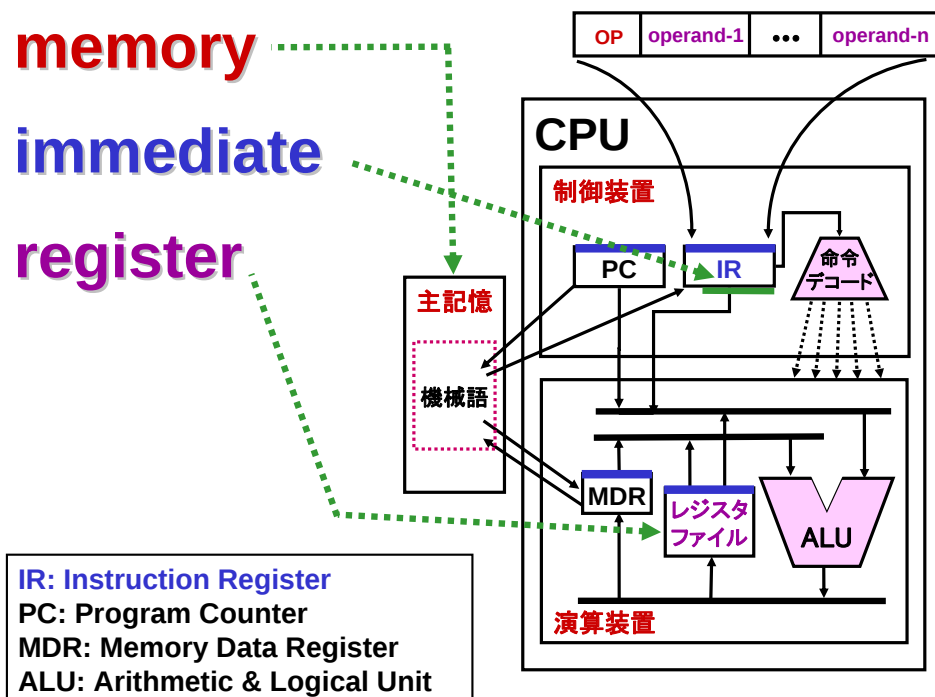
1010110011110010000000....

opcode (OP)	operand-1	...	operand-n
----------------	-----------	-----	-----------

- **opcode / OP** : Operation Code (操作コード、オペコード、命令コード)
 - データ読み書き・移動: load, store, move
 - 算術演算: add, sub, multiply, divide
 - 論理演算: and, or, xor, logical shift, arithmetic shift , ...
 - 分岐: branch, branch if not equal, branch if less than, ...
- **operand** : 操作(演算)対象
 - **memory** (主記憶の内容)
 - **immediate** (即値=機械語の一部をそのままデータとする)
 - **register** (レジスタ)

10

memory
immediate
register



11

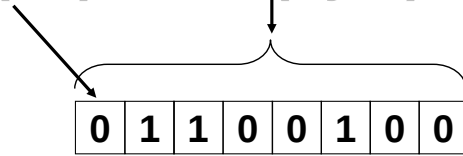
プロセッサと機械語

計算機の内部表現

12

データの計算機内部表現

ビット(bit), バイト(byte)=8bit



ワード／語 (word)= 32bit (多くの場合)

ハーフワード／半語 (half word)= 16bit

ダブルワード／倍長語 (double word)= 64bit

13

データの計算機内部表現

●これは何と読める？

opcode (OP)								operand-1	...	operand-n
0	1	1	0	0	1	0	0			

- 解釈1: $01100100_2 = 64_{10} + 32_{10} + 4_{10} = 100_{10}$
- 解釈2: $0110_2 \times 10 + 0100_2 = 60_{10} + 4_{10} = 64_{10}$
- 解釈3: ASCIIコード “64”₁₆ = 文字 “d”
- 解釈4: OPコード “64”₁₆ = add命令
- 所詮、約束事 ⇒ 代表的なものは覚える
⇒ 自分達で決める場合もある

14

データの計算機内部表現

- 0, 1の2値
- 文字
 - ASCII 7ビットコード
 - JIS 16ビットコード
- 整数(固定小数点数)
 - 符号なし
 - 符号付き: 2の補数表現
- 10進数
 - 0000~1001 (BCD: Binary Coded Decimal)
- 浮動小数点数
 - 仮数 × 基底^{指数}
 - 単精度: 32ビット
 - 倍精度: 64ビット
 - 拡張倍精度: 128ビット

15

データの計算機内部表現

● ASCIIコード

“b”

“i”

“t”

という文字列は
ASCIIコードで

01100010

01101001

01110100

と表す

	b6~b4							
b3~b0	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	'	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAC	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF/NL	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

16

2進数表現と16進数表現 (binary) (hexa-decimal)

10進数	2進数	16進数	10進数	2進数	16進数
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

2進数表現と16進数表現(例)

10進数	2進数	16進数
100		
= 64	1000000	
+ 32	+ 100000	
+ 4	+ 100	
	= 1100100	= 64

表記法

10進数: 100, 100₁₀

2進数: 1100100₂, b'1100100', 1100100B

16進数: #64, #'64', 64₁₆

演習：2進数表現と16進数表現を求めよ

10進数	2進数	16進数
200		
=		
	=	=

宿題：2進数表現と16進数表現を求めよ

- 授業HP「第1回宿題」のPDFを各自印刷し、次回授業の冒頭に副手に提出すること