



計算機アーキテクチャ

第2回 ノイマン型コンピュータ

2014年 4月14日

電気情報工学科 田島 孝治



授業スケジュール(前期)

回	日付	タイトル
1	4/7	コンピュータ技術の歴史とコンピュータアーキテクチャ
2	4/14	ノイマン型コンピュータ
3	4/21	コンピュータのハードウェア
4	4/28	数と文字の表現
5	5/12	固定小数点数と浮動小数点表現
6	5/19	計算アーキテクチャ(ARU)
7	5/26	計算装置のハードウェア実装
8	6/2	文字コード
9	6/9?	中間試験(日程は仮)

回	日付	タイトル
10	6/16	主記憶装置とレジスタ
11	6/23	命令実行の流れ
12	6/30	命令形式とアセンブリ言語
13	7/7	命令セット
14	7/14	サブルーチンの実現
15	7/28	PCSpimによるアセンブリ言語プログラム
	8/4?	期末試験(日程は仮)
16	9/29?	フォローアップ(日程は仮)

※5/5はこどもの日、7/21は海の日のため休講

※急なスケジュール変更があった場合、掲示およびメールで連絡します



今日の授業の目的

ノイマン型コンピュータについての理解

ノイマン型コンピュータの

- **特徴**を理解する
- **構成要素**と関係性を理解する
- **計算手順**と**計算時間**について理解する



ノイマン型コンピュータとは何か

- 次の特徴を持つコンピュータ
 - プログラム(可変)内蔵方式
 - プログラムを内部のメモリで保持
 - プログラムは書き換え可能
 - デジタル処理
 - 数値を0/1の組み合わせで表す
- その他の特徴として
 - 逐次処理方式
 - プログラムは実行順にメモリに格納されている
 - 単一メモリ、線形メモリ
 - プログラムとデータは、一貫したアドレスを持つ同一のメモリ上に保持される

ノイマン型コンピュータのメモリ

- 全てが同一のメモリ空間
 - データもプログラムも書き換え可能
- メモリには0または1が格納できる
 - プログラムも数値で表す必要がある
- 数個の0/1をまとめて扱う
 - これをビット幅という
 - 8つまとめる: 8bit、64個まとめる: 64bit
- まとまり単位に場所を表す一意の番号を付ける
 - これをメモリ番地と呼ぶ
- プログラムは連続するメモリ番地に実行順に格納される

メモリ番地	ビット幅							
1	0	0	0	1	1	0	1	0
2	0	0	0	1	1	0	1	1
3	0	0	1	1	0	1	1	0
4	0	0	1	0	0	1	0	0
5	0	0	0	0	0	0	0	0
6	1	0	0	1	1	1	1	0
7	0	0	0	0	1	0	1	1
8	0	1	0	0	1	1	1	1
9	0	1	0	0	0	0	0	1
10	0	0	1	1	1	1	1	0
11	0	0	0	1	0	1	0	0
12	1	0	0	0	1	1	0	0

もう少し詳しくメモリを見てみると

メモリ番地								
1	0	0	0	1	1	0	1	0
2	0	0	0	1	1	0	1	1
3	0	0	1	1	0	1	1	0
4	0	0	1	0	0	1	0	0
5	0	0	0	0	0	0	0	0
6	1	0	0	1	1	1	1	0
7	0	0	0	0	1	0	1	1
8	0	1	0	0	1	1	1	1
9	0	1	0	0	0	0	0	1
10	0	0	1	1	1	1	1	0
11	0	0	0	1	0	1	0	0
12	1	0	0	0	1	1	0	0

解釈
10番地からデータを取得
11番地からデータを取得
取得中のデータを加算
12番地にデータを保存
プログラム終了
データ？
データ？
データ？
データ？
数値データ: 62
数値データ: 20
数値データ: -166

プログラム
領域

未使用領域

データ領域



演習問題1

- 次の2台のPCについてメモリ番地がいくつまであるかを計算せよ。また、その数値を表現するのに必要なビット数も求めよ
 - 2000年ごろの一般的なPC
 - ビット幅: 32bit (CPU: Pentium III-733MHz)
 - メモリ容量: SDRAM 128Mbytes
 - 現在の一般的なPC
 - ビット幅: 64bit (Core-i5-4440 3.1GHz)
 - メモリ容量: DDR3 8GByte
- 諸注意
 - 1byte = 8bit
 - 1kbyte(KB) = 1024byte = 2^{10} byte
 - M(メガ)は 2^{20} 、G(ギガ)は 2^{30} と



計算用紙

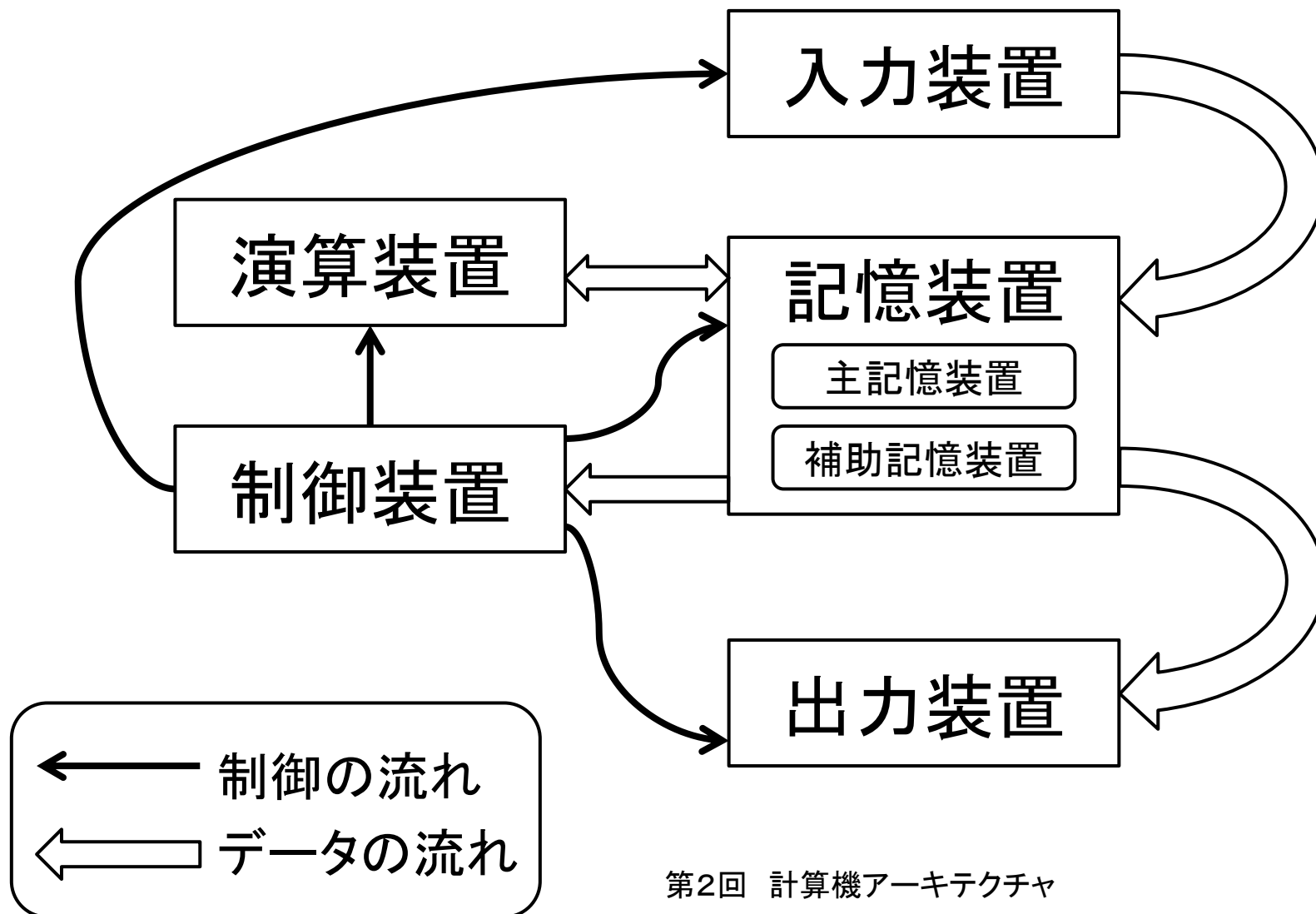
数値列の意味とアーキテクチャ

- 0,1の意味付け: 計算機的设计そのもの
- どうやって数値化するか: 2進数の考え方
- 数字以外のデータをどう扱うか
 - 文字、画像はどうする?
- 命令をどう扱うか
 - 文字の集合にする?
 - 数列を命令に置き換える?
 - 何種類の命令数を用意する?

数値表現
文字コード

命令セット
アーキテクチャ

ノイマン型コンピュータの構成





それぞれの装置の特徴

- 演算装置
 - データの算術演算や論理演算を行う装置
- 制御装置
 - 全ての装置をコントロールする装置
 - ハードウェアとしては演算装置と統合されることが多い
 - 統合した装置は中央処理装置(CPU)と呼ばれる
- 記憶装置
 - データやプログラムを記憶する装置
 - 高速なやり取りを行うための主記憶装置と、長期的にデータを記憶しておくための補助記憶装置に分けて考えることが多い
- 入力装置
 - プログラムやデータを主記憶装置に入力する装置
- 出力装置
 - 計算機が処理したデータを出力する装置



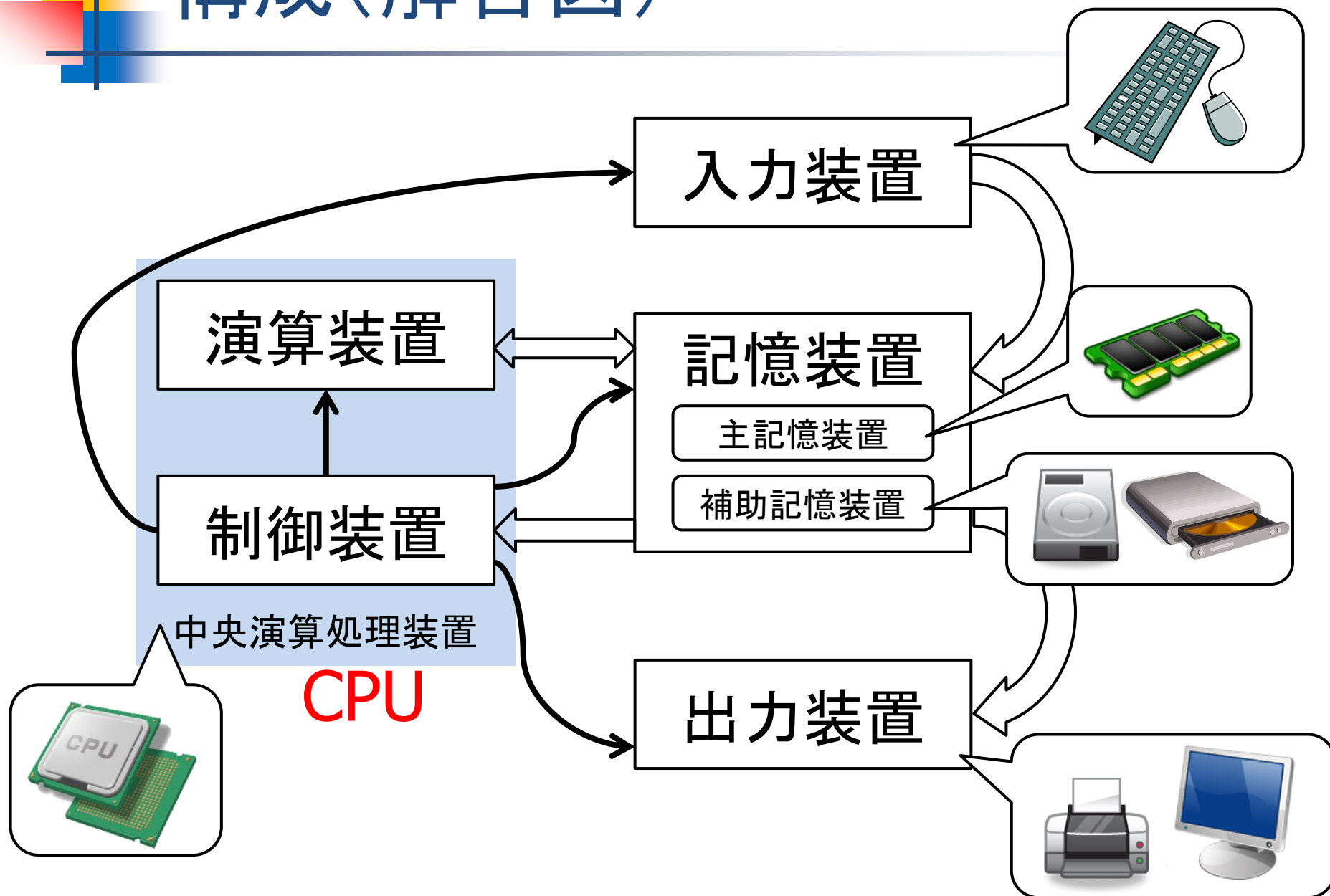
演習問題2

■ 次の装置を分類せよ

- (A)演算装置、(B)制御装置、(C)主記憶装置、
(D)補助記憶装置、(E)入力装置、(F)出力装置
から選択すること

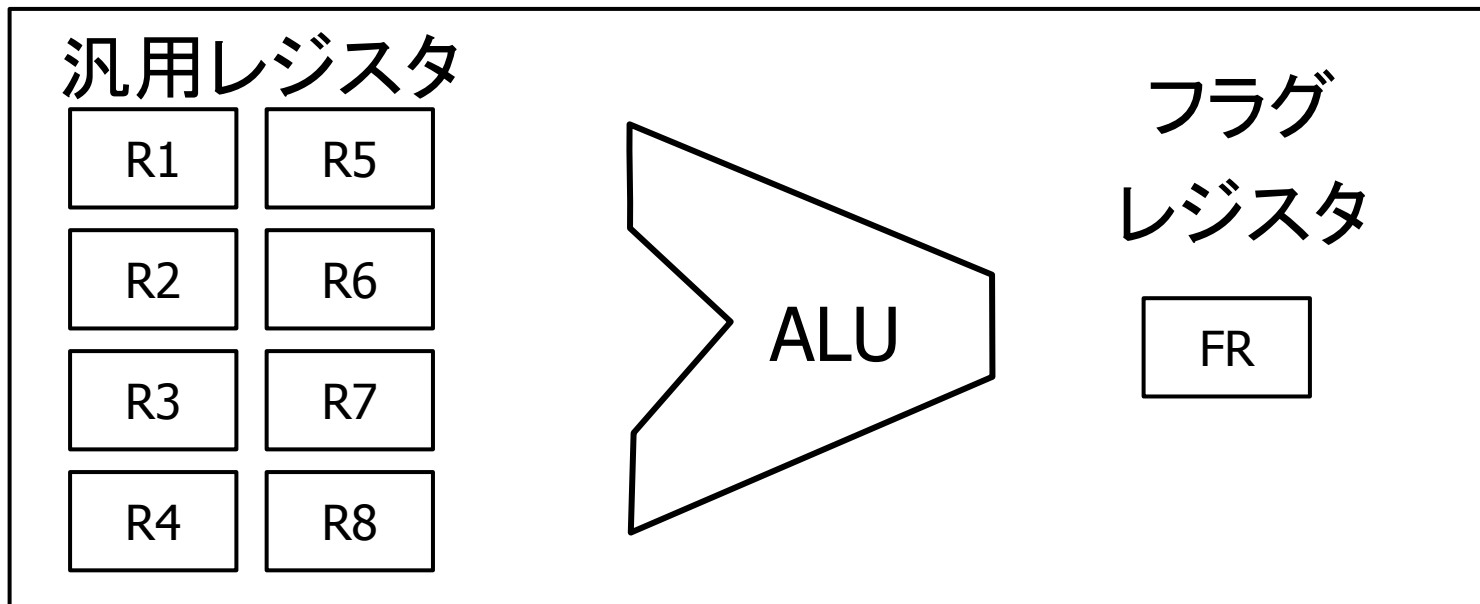
ハードウェア	分類	ハードウェア	分類
フロッピードライブ		USBメモリ	
液晶ディスプレイ		内臓メモリ	
マウス		キーボード	
ネットワークHDD		DVDドライブ	
内臓HDD		グラフィックボード	
CPU		プリンタ	

構成(解答図)



演算装置をもう少し細かく見てみよう

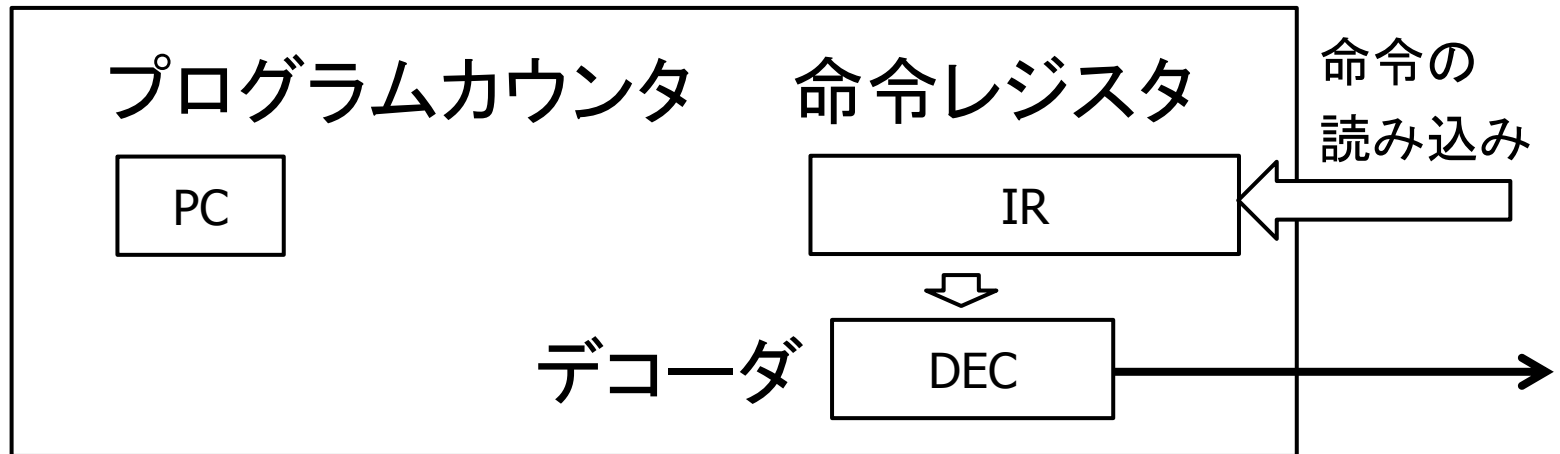
■ 演算装置の構成



- レジスタ: 演算装置内にある一次的な記憶装置
 - メモリに比べ非常に高速にアクセスできる
- ALU: 算術演算、論理演算を行う電子回路

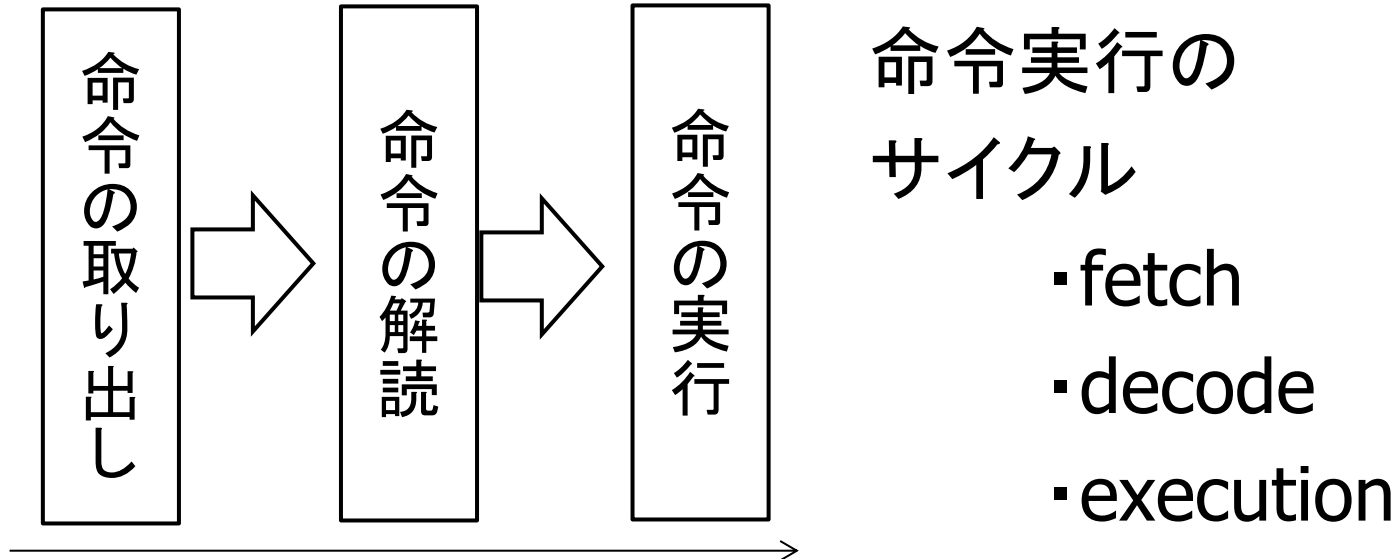
制御装置をもう少し細かく見てみよう

■ 制御装置の構成



- **プログラムカウンタ**: 次に実行する**命令の番地**を記録
 - 自動で1つつ加算されていくようになっている
- **命令レジスタ**: メモリから取り出した命令を一時的に記録
- **デコーダ**: 命令レジスタの内容を解釈し制御信号を出力

ノイマン型コンピュータの計算過程



- どんな**命令もメモリ上**に記録されている
 - 実行時に**必ず1回はメモリを読み込む**
- 命令の内容が「**メモリのデータを取り出す**」場合には再びメモリにアクセスする

⇒ メモリへのアクセス回数は非常に多い

なぜノイマン型？



- Von Neumann (1903-1957)
 - ハンガリー出身のアメリカの数学者
- First Draft of a Report on the EDVAC
 - (EDVACに関する報告書)
 - 1945年にJohn von Neumann が執筆
 - EDVACの設計に対する概念をまとめた報告書
- ノイマンは開発に対して客観的な立場から書いたものであり、実際のアイデアを提案したのは本人だけではないという話もある

演習問題3

- 次のコンピュータで3つの数の加算を行った場合の処理時間を求めよ
 - CPUの演算速度: 1処理あたり6秒
 - メモリの読み書き速度(バス速度): 6秒
 - プログラムの読み込みにもこの時間が必要
- 実行上の注意
 - CPUメモリ上の000番地からの内容をプログラムとして読み込み実行する
 - メモリ上に結果を格納する

メモリ

番地	内容
000	100番地の値を取得
001	101番地の値を取得
002	レジスタの内容を加算
003	結果を103番地へ保存
004	102番地の値を取得
005	103番地の値を取得
006	レジスタの内容を加算
007	結果を103番地へ保存
...	...
100	データA
101	データB
102	データC
103	結果保存用



演習問題の計算用紙

演習問題4

- もっと早く計算がしたい！
- 高速なCPUを買ってみた！
 - CPUの演算速度：1処理あたり2秒(3倍速)
 - メモリの読み書き速度(バス速度)：6秒
 - プログラムの読み込みにもこの時間が必要

メモリ

番地	内容
000	100番地の値を取得
001	101番地の値を取得
002	レジスタの内容を加算
003	結果を103番地へ保存
004	102番地の値を取得
005	103番地の値を取得
006	レジスタの内容を加算
007	結果を103番地へ保存
...	...
100	データA
101	データB
102	データC
103	結果保存用



計算時間の計算用紙



演習問題5

■ さらに高速なCPUでは？

- CPUの演算速度：1処理あたり0.6秒
- メモリの読み書き速度（バス速度）：6秒
 - プログラムの読み込みにもこの時間が必要

メモリ

番地	内容
000	100番地の値を取得
001	101番地の値を取得
002	レジスタの内容を加算
003	結果を103番地へ保存
004	102番地の値を取得
005	103番地の値を取得
006	レジスタの内容を加算
007	結果を103番地へ保存
...	...
100	データA
101	データB
102	データC
103	結果保存用



計算時間の計算用紙

なぜあまり早くないのか

■ 処理時間の内訳

	計算処理	メモリへの読み書き	合計時間
初期状態	秒	秒	秒
3倍速	秒	秒	秒
10倍速	秒	秒	秒

■ 処理時間の大半メモリの読み書きが占める

- フォンノイマンボトルネック
(Von Neumann bottleneck)

bottleneck



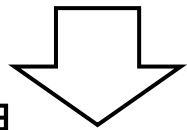
■ メモリバスの取り合いにより高速に計算を行うことが難しい

⇒ 今のコンピュータでも起こっている？

演習問題6

田島の使っている計算機の場合

- CPU: Core i5 3.2GHz
- メモリの種類: PC3-12800
 - データ転送: 12.8GB/s
 - 転送速度としては・・・800MHz



1回あたり何秒？

- 加算処理
- メモリからのデータ取得

※結果は概算です(加算が1クロックで終わるとは限らない)

メモリ

番地	内容
000	100番地の値を取得
001	101番地の値を取得
002	レジスタの内容を加算
003	結果を103番地へ保存
004	102番地の値を取得
005	103番地の値を取得
006	レジスタの内容を加算
007	結果を103番地へ保存
...	...
100	データA
101	データB
102	データC
103	結果保存用



計算時間の計算用紙



計算時間の計算用紙



対策技術

- メモリを早くすれば良いのですが...
 - 性能的な限界よりも制御的な限界があります
 - 半導体への読み書きよりも通信路と、CPUからのリクエスト処理(割り当て)が問題
- マルチチャネルアーキテクチャ
 - 一般的にはデュアルチャネル(Dual-channel)
 - 複数のメモリを同時に読み書き可能にする
 - 1回の通信で2データ分をやり取りできるイメージ
 - 仮想的に通信路を拡大することができる



演習問題7

- 次のコンピュータで3つの数の加算を行った場合の処理時間を求めよ
 - CPUの演算速度: 1処理あたり6秒
 - メモリの読み書き速度(バス速度): 6秒
 - プログラムの読み込みにもこの時間が必要
 - デュアルチャネル
 - プログラムは2行まとめて読み込める
 - 読み込んだ2つの命令が両方とも読み込み命令の場合には、データも2つ同時に読み込める
- 実行上の注意
 - CPUメモリ上の000番地からの内容をプログラムとして読み込み実行する
 - メモリ上に結果を格納する

メモリ

番地	内容
000	100番地の値を取得
001	101番地の値を取得
002	レジスタの内容を加算
003	結果を103番地へ保存
004	102番地の値を取得
005	103番地の値を取得
006	レジスタの内容を加算
007	結果を103番地へ保存
...	...
100	データA
101	データB
102	データC
103	結果保存用



計算時間の計算用紙
