

## 計算機アーキテクチャ 第二 (O)

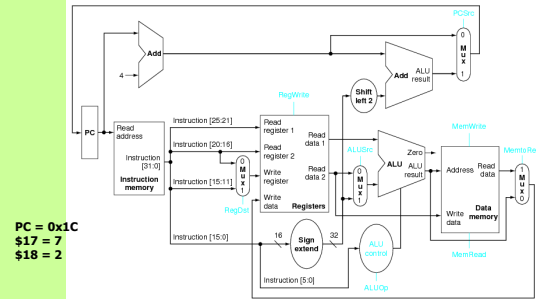
### 4. パイプライン処理

大学院情報理工学専攻 計算工学専攻  
吉瀬謙二 kise\_at\_cs.titech.ac.jp  
S321講義室 月曜日 5, 6時限 13:20-14:50

1

## プロセッサのデータパス(シングル・サイクル)

op rs rt rd shamt funct  
add \$t0, \$s1, \$s2 [ add \$8, \$17, \$18 ]



Adapted from Computer Organization and Design, Patterson &amp; Hennessy, © 2005

2

## MIPSの基本的な5つのステップ(ステージ)

- **IF (Instruction fetch)ステージ**  
メモリから命令をフェッチする。
- **ID (Instruction decode and register file read) ステージ**  
命令をデコードしながら、レジスタを読み出す。
- **EX (Execution or address calculation) ステージ**  
命令操作の実行またはアドレスの生成を行う。
- **MEM (Data memory access) ステージ**  
データ・メモリ中のオペランドにアクセスする。
- **WB (Write back) ステージ**  
結果をレジスタに書き込む。

Adapted from Computer Organization and Design, Patterson &amp; Hennessy, © 2005

3

## 機能レベルのプロセッサシミュレータ

```

//xxx SimCore/MIPS since 2006-10-06 Kenji KISE, Tokyo Tech xxx//
//=====
#include <stdio.h>
#include <stdlib.h>

#define L_NAME "SimCore/MIPS Functional Simulator since 2006-11"
#define L_VER "V.0.6.4-simple 2012-10-07"

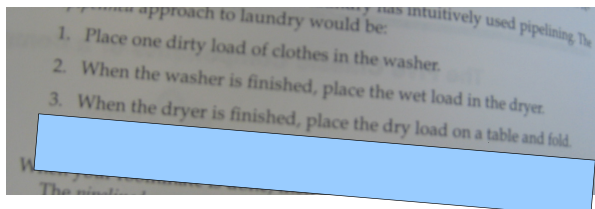
//=====
#define ID_NOP 0x000
#define ID_SLL 0x001
#define ID_SRL 0x002
#define ID_SHL 0x003
#define ID_ADD 0x004
#define ID_ADDI 0x005
#define ID_SUB 0x006
#define ID_SUBI 0x007
#define ID_AND 0x008
#define ID_OR 0x009
#define ID_XOR 0x00a
#define ID_NOR 0x00b
#define ID_SLT 0x00c
#define ID_L 0x00d
#define ID_BEQ 0x00e
#define ID_BNE 0x00f
#define ID_ADDI 0x010
#define ID_ADDIU 0x011
#define ID_SLLI 0x012
#define ID_LUI 0x013
#define ID_SW 0x014
#define ID_RCEP 0x015
#define ID_NOTIMP 0xfff /* not-implemented instruction */

```

Adapted from Computer Organization and Design, Patterson &amp; Hennessy, © 2005

4

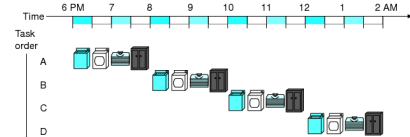
## パイプライン処理 (pipelining)



Adapted from Computer Organization and Design, Patterson &amp; Hennessy, © 2005

5

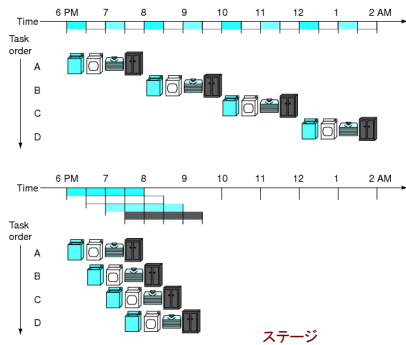
## パイプライン処理 (pipelining)



Adapted from Computer Organization and Design, Patterson &amp; Hennessy, © 2005

6

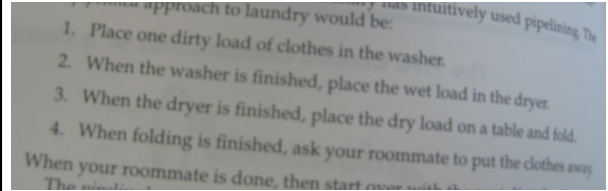
## パイプライン処理 (pipelining)



Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

7

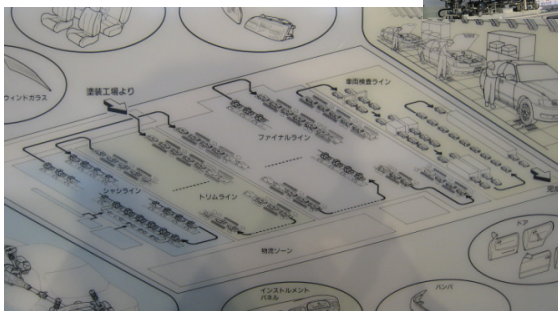
## パイプライン処理 (pipelining)



Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

8

## パイプライン処理 (pipelining)



Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

9

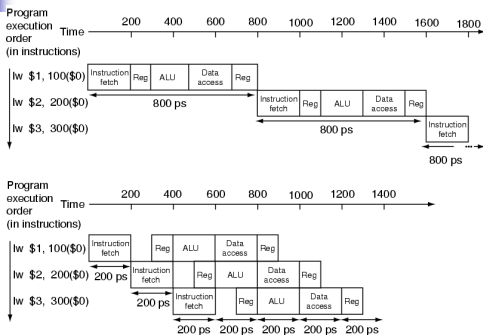
## MIPSの基本的な5つのステップ(ステージ)

- **IF (Instruction fetch)ステージ**  
メモリから命令をフェッチする。
- **ID (Instruction decode and register file read) ステージ**  
命令をデコードしながら、レジスタを読み出す。
- **EX (Execution or address calculation) ステージ**  
命令操作の実行またはアドレスの生成を行う。
- **MEM (Data memory access) ステージ**  
データ・メモリ中のオペランドにアクセスする。
- **WB (Write back) ステージ**  
結果をレジスタに書き込む。

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

10

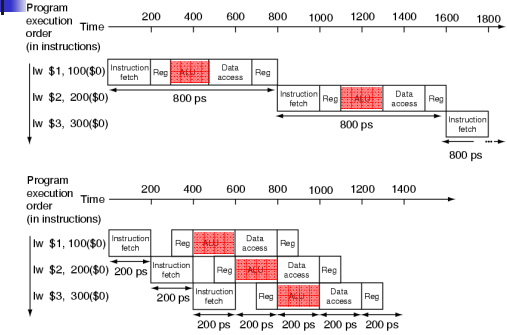
## パイプライン処理 (pipelining)



Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

11

## パイプライン処理 (pipelining)



Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

12

## パイプラインによる速度向上

- パイプラインステージの数(段数):  $n$
- 実行する命令の数:  $s$
- パイプライン化されたプロセッサのクロックを単位時間とする.
- 全命令が終了するまでの理想的なサイクル数
  - $n + s - 1$
- パイプラインを利用しないシングルサイクルのプロセッサ
  - $n * s$

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

13

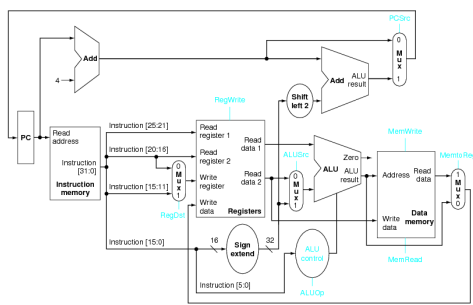
## パイプライン処理のポイント

- 処理の分割
  - ある処理(命令の実行)をできるだけ均等なステージに分割
  - 例えば, 均等な10段のステージに分割
- 処理のためのハードウェア資源の投入
  - それぞれのステージを処理するためのハードウェアを準備
  - 例えば, 10段のステージに分割した場合には, それぞれのステージの処理を担当する10個のハードウェアユニットを準備
- 処理の進行
  - それぞれのハードウェアがある命令のステージを処理する.
  - それぞれのハードウェアが独立に作業できるように, 入力と出力にレジスタを挿入.

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

14

## プロセッサのデータパス(シングル・サイクル)

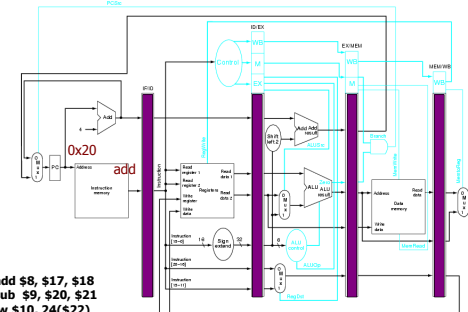


Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

15

## プロセッサのデータパス(パイプライン処理)

Clock 1:



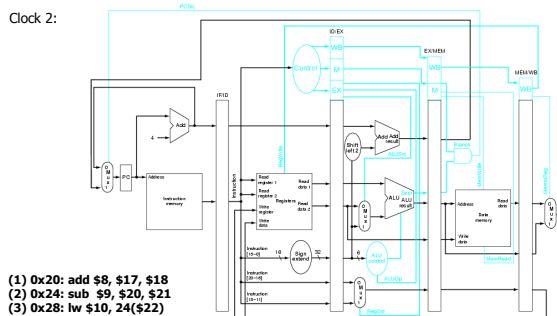
(1) 0x20: add \$8, \$17, \$18  
(2) 0x24: sub \$9, \$20, \$21  
(3) 0x28: lw \$10, 24(\$22)

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

16

## プロセッサのデータパス(パイプライン処理)

Clock 2:



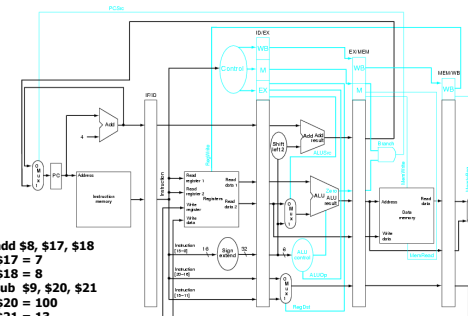
(1) 0x20: add \$8, \$17, \$18  
(2) 0x24: sub \$9, \$20, \$21  
(3) 0x28: lw \$10, 24(\$22)

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

17

## プロセッサのデータパス(パイプライン処理)

Clock 3:



(1) 0x20: add \$8, \$17, \$18  
\$17 = 7  
\$18 = 8  
(2) 0x24: sub \$9, \$20, \$21  
\$20 = 100  
\$21 = 13  
(3) 0x28: lw \$10, 24(\$22)  
\$22 = 128

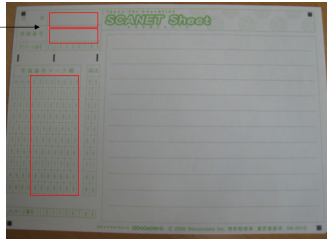
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

18

## Exercise

Clock 3 における各パイプラインステージの様子を示せ。

氏名, 学籍番号,  
学籍番号マーク欄(右詰で)



Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

19

## アナウンス

- 講義スライド, 講義スケジュール
- [www.arch.cs.titech.ac.jp](http://www.arch.cs.titech.ac.jp)

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

20