

# 計算機科学実験及演習 3A SIMPLEアーキテクチャ のプロセッサの実装

京都大学 工学部情報学科 計算機科学コース  
計算機科学実験及演習3  
HW 担当グループ

2017/04

## 実験3ハードウェア

2

## 実験3ハードウェア - 内容と目的 -

- 内容
  - マイクロプロセッサの方式設計、論理設計
  - FPGA上で応用プログラムを動作
- 目的
  - プロセッサの動作原理を理解する
  - 回路設計、最適化、動作テストの方法を知る
- 参考書
  - 富田真治、中島浩：コンピュータハードウェア
  - D.A.パターソン、J.L.ヘネシー著、成田光彰訳：コンピュータの構成と設計(上),(下) 第3or4版

3

## 課題とデモンストレーション

- 導入課題： 個人単位
    - 基本的な回路の設計
    - レポートの執筆
  - プロセッサ設計演習： グループ単位
    - 中間レポート
      - 各種仕様書、考察等
    - 中間デモ
      - 何らかの命令が動作しているところを見せる
    - 最終デモ
      - 設計したプロセッサの特徴の説明
        - 「SIMPLE/Bに比べてこんな点がすぐれている」というセールストーク
        - 応用プログラムの実行
    - 最終レポート
      - 最終成果物のユーザーズマニュアル
      - SIMPLE/B基本仕様からの拡張および性能評価
      - 各コンポーネントの仕様書(最終版)、性能評価
      - 考察、感想
- どう設計したか、何を使ったか、レポートに明記。
  - 単に動くだけではなく、スピード、回路規模、消費電力の最適化を目指す。

4

# SIMPLEプロセッサアーキテクチャ

5

## SIMPLEの概要

→ Sixteen-bit Microprocessor for Laboratory Experiment

- ☺ 簡単な命令セット
- ☺ 基本機能は1通り備えられている

### • 特徴

- 16bit固定長命令
- 8本の汎用レジスタ
- 16bit × 64K語の主記憶
- ロード/ストア・アーキテクチャ
- 2オペランド形式の命令セット(Rd op Rs -> Rd)

6

## アーキテクチャの説明

- アーキテクチャ
  - コンピュータ全体の構成
    - プロセッサ、メモリ、I/Oなど
  - 主記憶とレジスタの構成ここに含む
- 命令セット・アーキテクチャ
  - 命令の構成
  - 前述のロード/ストア・アーキテクチャは命令セットの形式の1つ
- マイクロ・アーキテクチャ
  - アーキテクチャの回路レベルでの実装

7

## 主記憶とレジスタ

コンピュータの状態を表すもの

1. 主記憶
  - 16bit × 64K語 (語アドレス方式)
  - ただし、実験で使用するFPGAでは約5K語(旧)、33K語(新)が最大
2. 汎用レジスタ
  - 16bit × 8語
3. プログラム・カウンタ (PC)
  - 16bit
4. 条件コード
  - S サイン
  - Z ゼロ
  - C キャリー
  - V オーバーフロー

8

# 命令セット

コンピュータの状態を変えるもの

## 1. 演算命令

- 算術論理演算命令
- シフト命令

## 2. ロード／ストア命令

## 3. 分岐命令

- 無条件分岐命令
- 条件分岐命令

## 4. その他

- 入出力命令
- 停止命令

9

# 演算命令

## • 算術論理演算命令

$$\bullet r[Rd] = r[Rd] \text{ op3 } r[Rs]$$

## • シフト命令

$$\bullet r[Rd] = \text{shift\_op3}(r[Rs], d)$$

- 注: 実行後に条件コードをセットする



10

# ロード／ストア命令(1)

## • ロード命令 (op1 : 00)

$$\bullet r[Ra] = *(r[Rb] + \text{sign\_ext}(d))$$

## • ストア命令 (op1 : 01)

$$\bullet *(r[Rb] + \text{sign\_ext}(d)) = r[Ra]$$



11

# ロード／ストア命令(2)

## • 即値ロード命令

$$\bullet r[Rb] = \text{sign\_ext}(d)$$

- 即値ロード命令2つとシフト命令で任意の16bitの値をレジスタ格納できる



12

## 分岐命令(1)

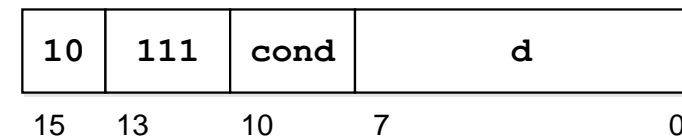
- 無条件分岐命令(B: Branch)
  - $PC = PC + 1 + \text{sign\_ext}(d)$



13

## 分岐命令(2)

- 条件分岐命令
  - $\text{if}(\text{cond}) PC = PC + 1 + \text{sign\_ext}(d)$
  - 条件コードの値に従って分岐
    - 条件コードは演算命令の実行時にセットされる



14

## その他の命令

- 停止命令 (op3: 1111)
- 入力命令 (op3: 1100)
  - $r[Rd] = \text{input}$
  - 入力先はボード上のスイッチ
- 出力命令 (op3: 1101)
  - $\text{output} = r[Rs]$
  - 出力先はボードのLED/7SEG LED

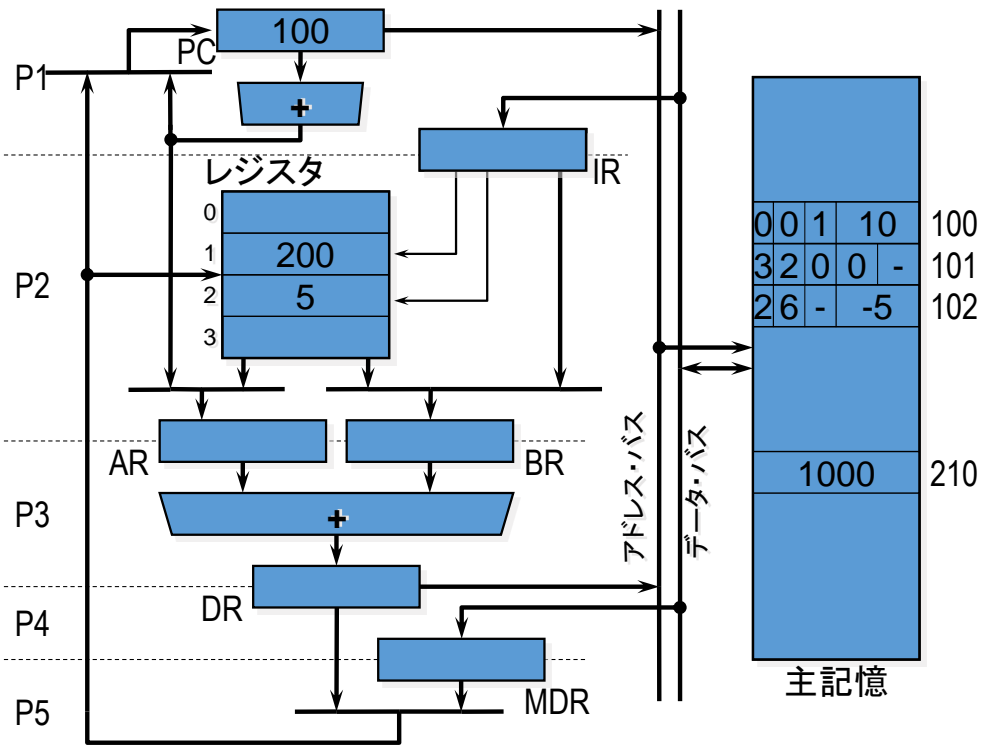


15

## 基本的な実装 SIMPLE/B

- 次スライドに示すように演算器／レジスタ／データ・パスを配置
- 5つのフェーズを逐次活性化: 実験2の順序回路と同じ
  - P1 命令フェッチ
  - P2 命令デコード、レジスタ読み出し
  - P3 演算
  - P4 主記憶アクセス
  - P5 レジスタ書き込み／PC更新
- フェーズの活性化: 制御部が担当
  - (フェーズへ入力されるデータを保持するレジスタを更新)
  - フェーズ内のセレクトを適切に切り替える
  - フェーズから出力されるデータを保持するレジスタを更新

16

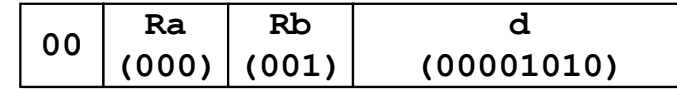


## 実行のサンプルの命令

- ロード命令: プログラム・カウンタ100

- LD R0, 10(R1)

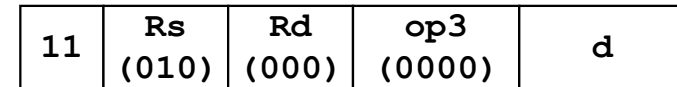
略記 **00110**



- 加算命令: プログラム・カウンタ101

- ADD R0, R2

略記 **3200-**



15 13 10 7 3 0

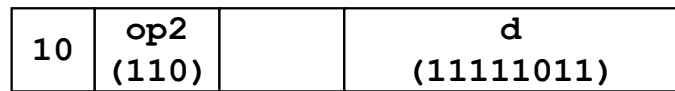
18

## 実行のサンプルの命令

- 無条件分岐命令: プログラム・カウンタ102

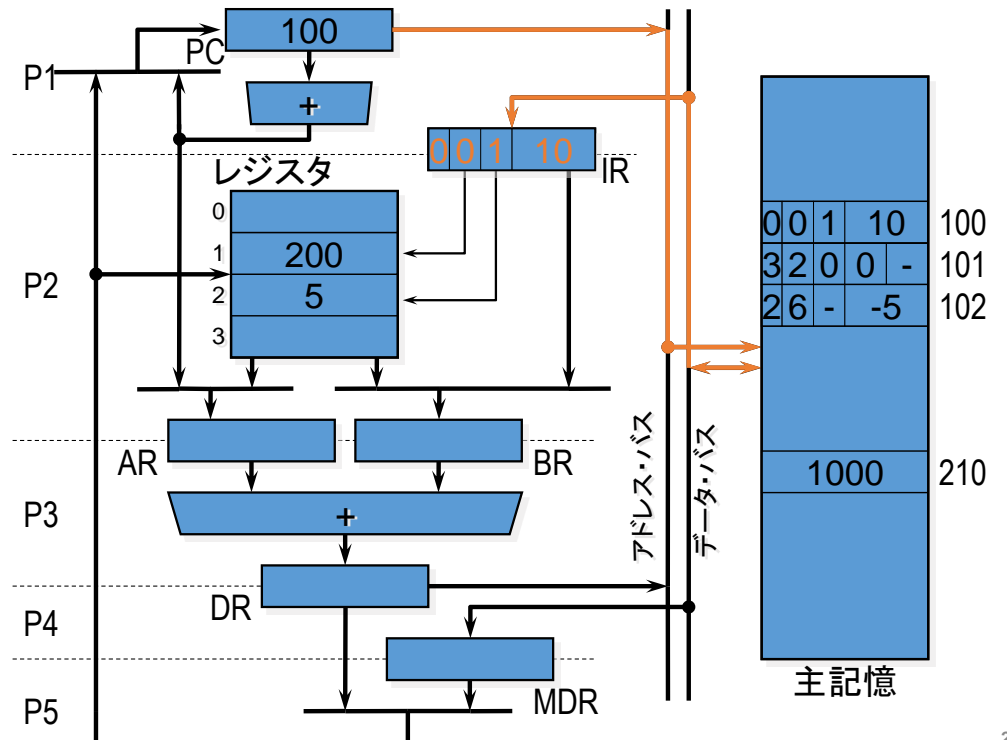
- B-5

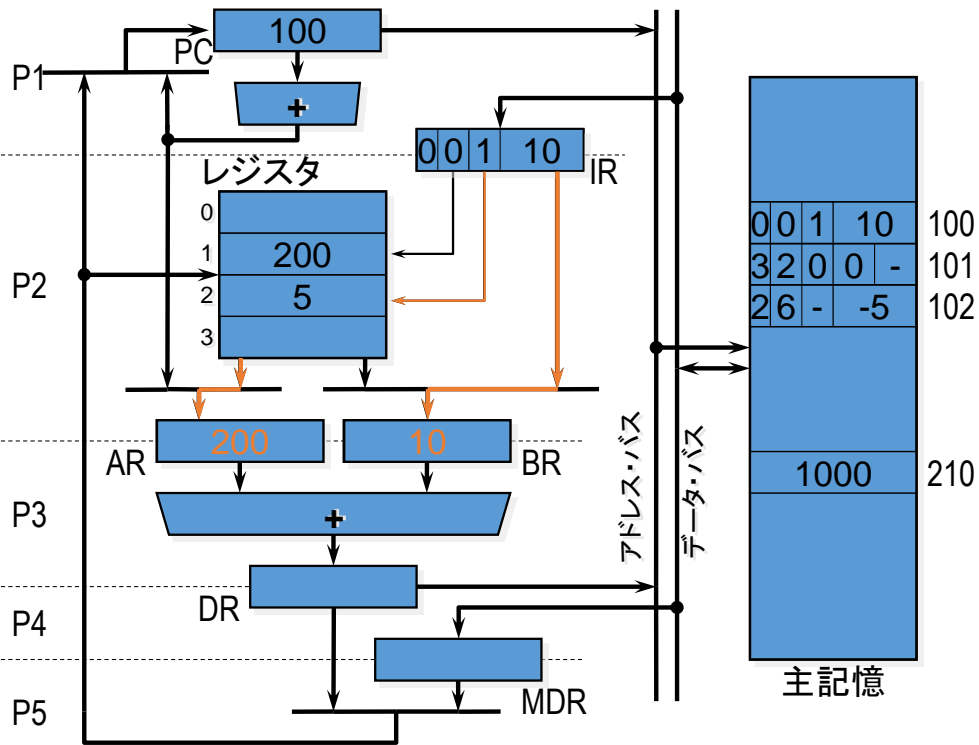
略記 **26--5**



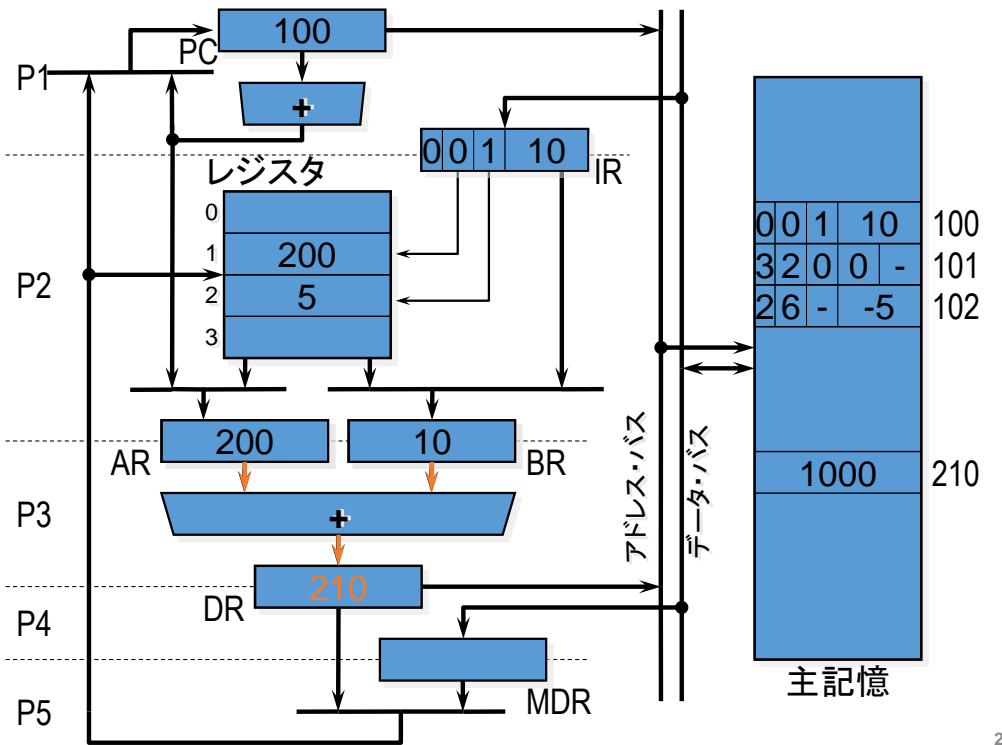
15 13 10 7 0

19

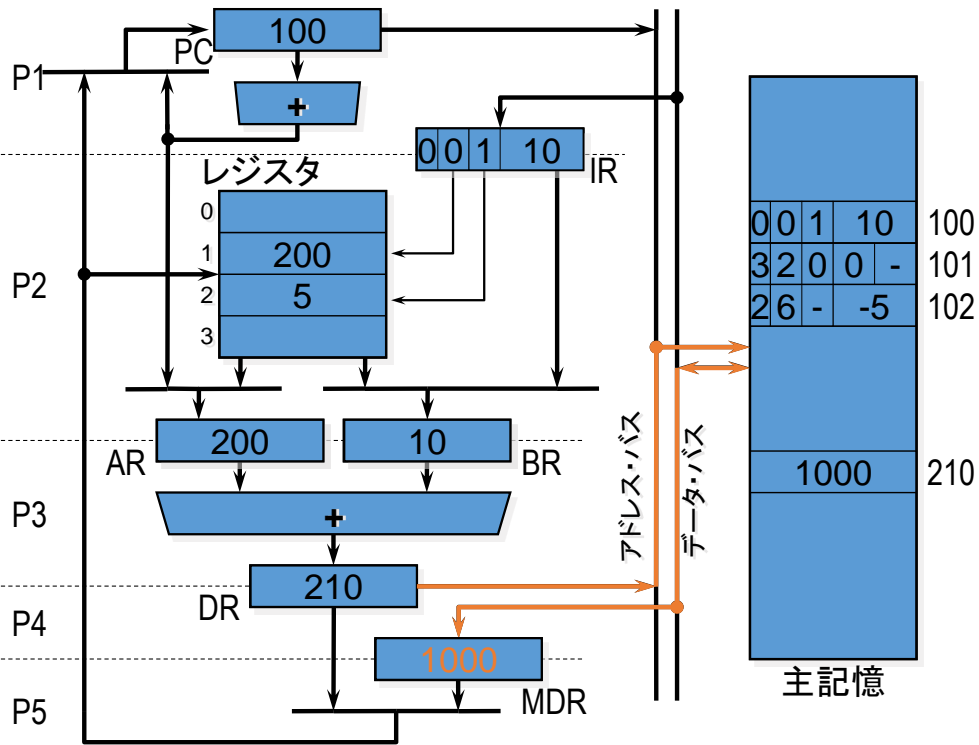




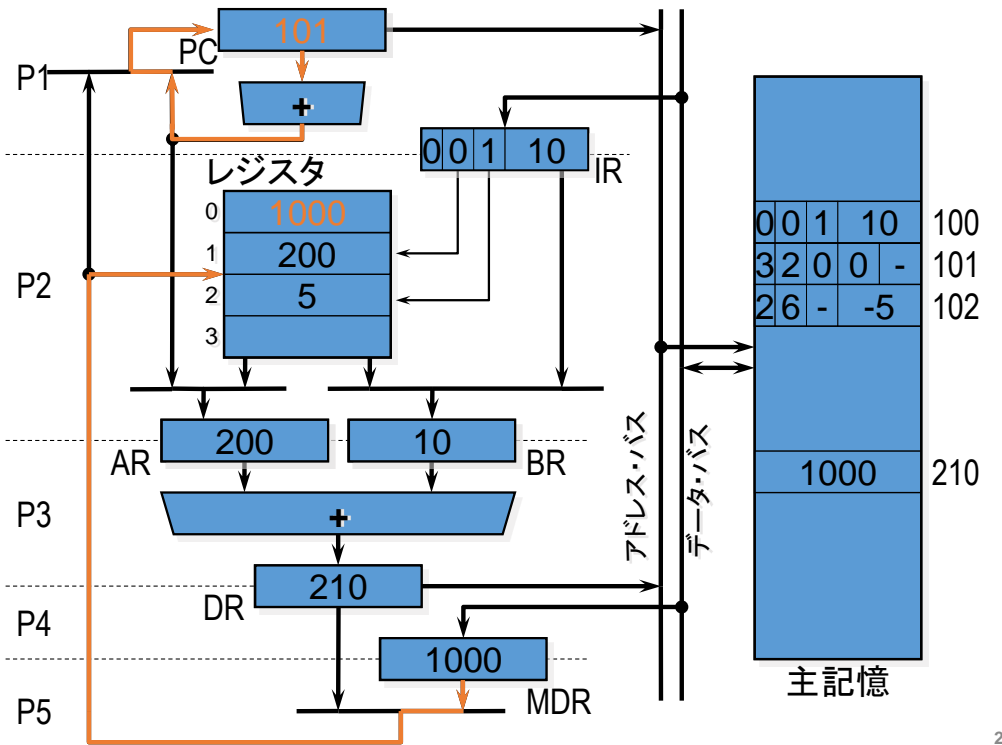
21



22



23



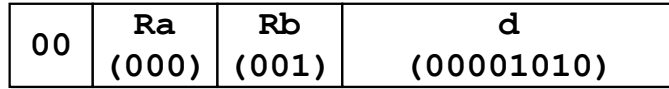
24

# 実行のサンプルの命令

• ロード命令: プログラム・カウンタ100

• LD R0, 10(R1)

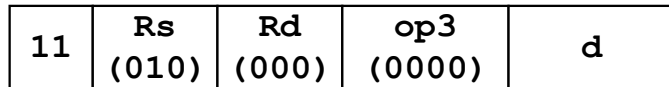
略記 00110



• 加算命令: プログラム・カウンタ101

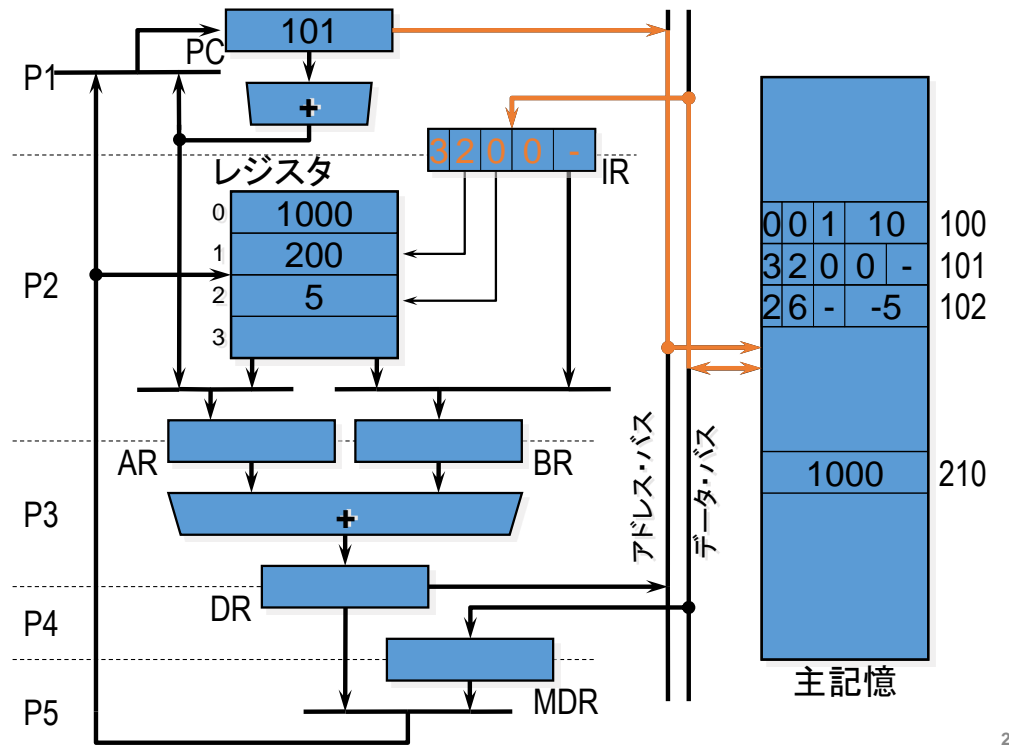
• ADD R0, R2

略記 3200-

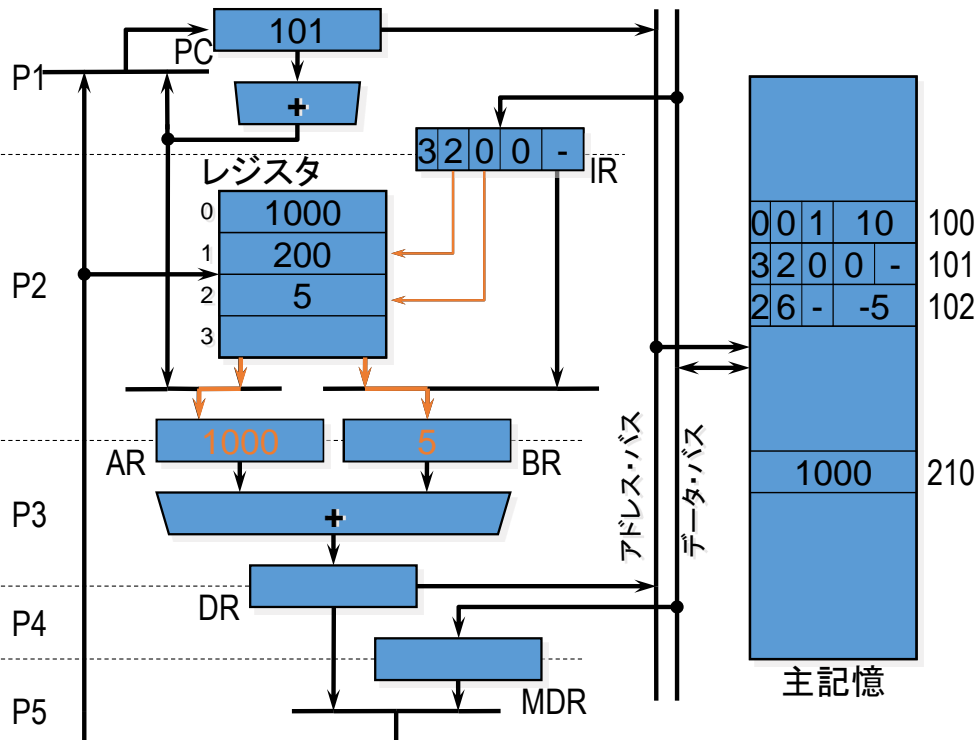


15 13 10 7 3 0

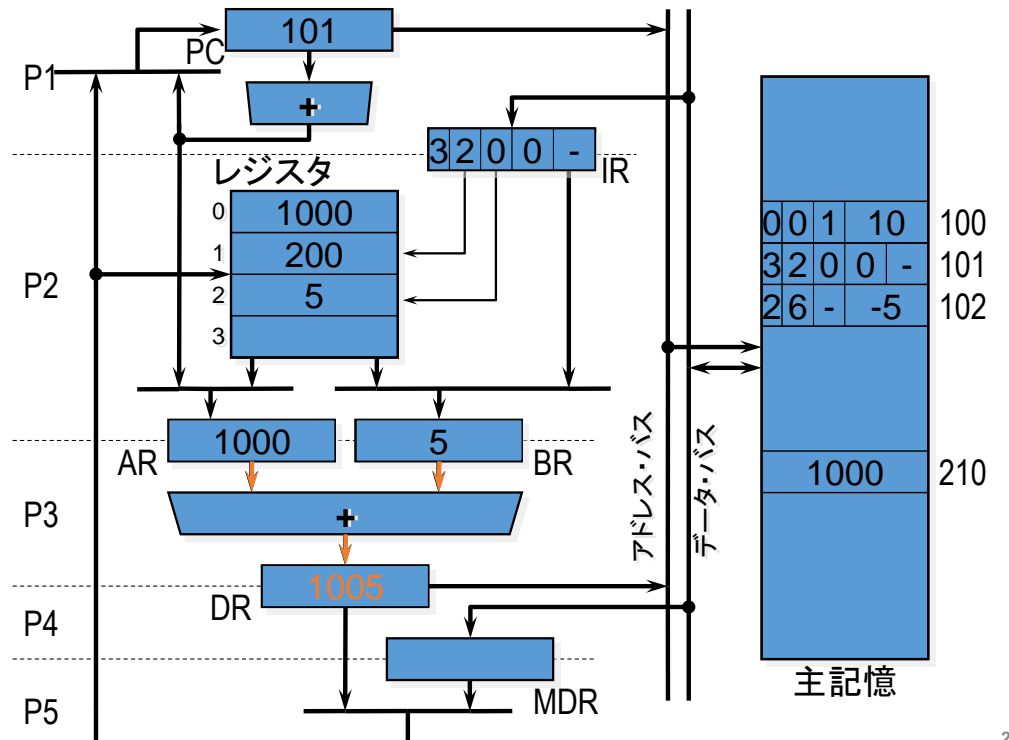
25



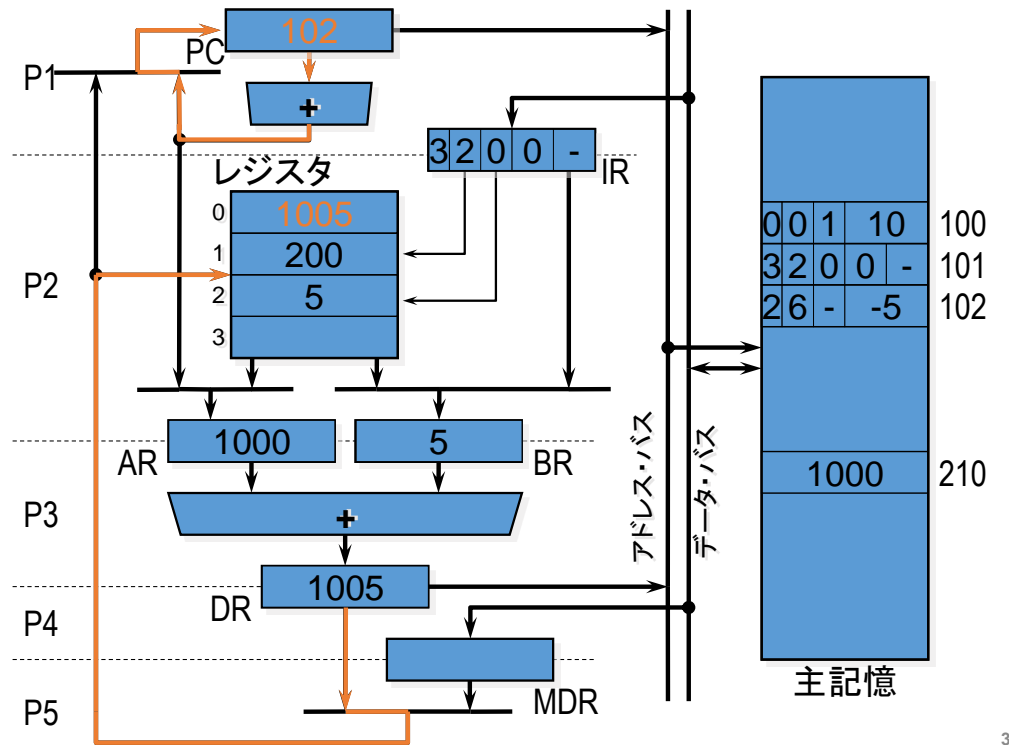
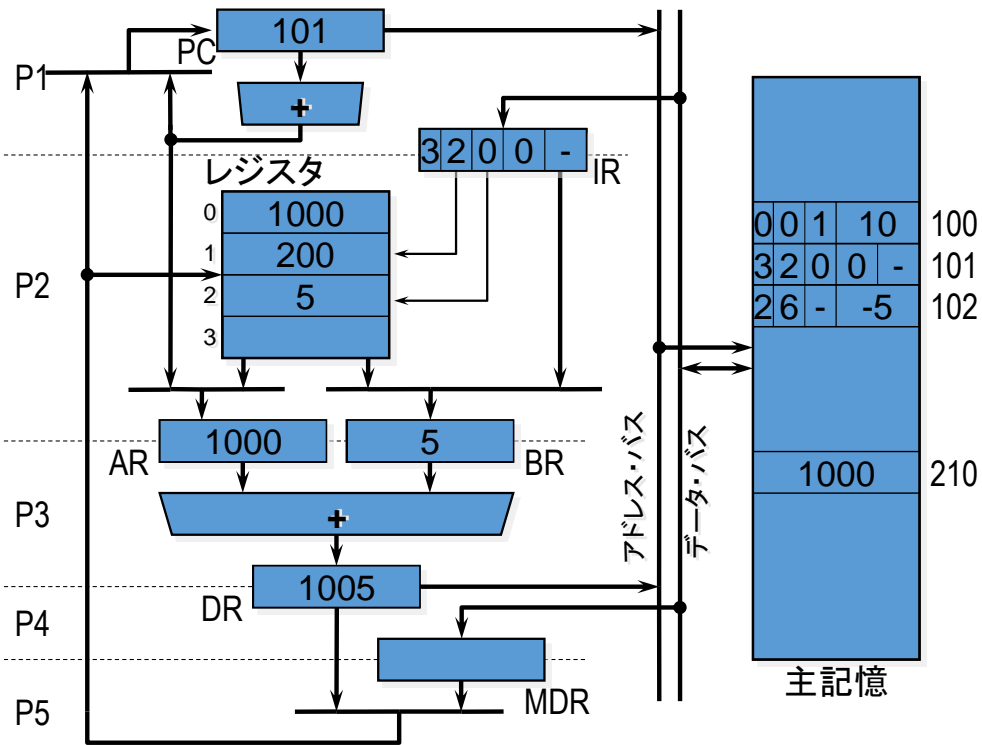
26



27



28

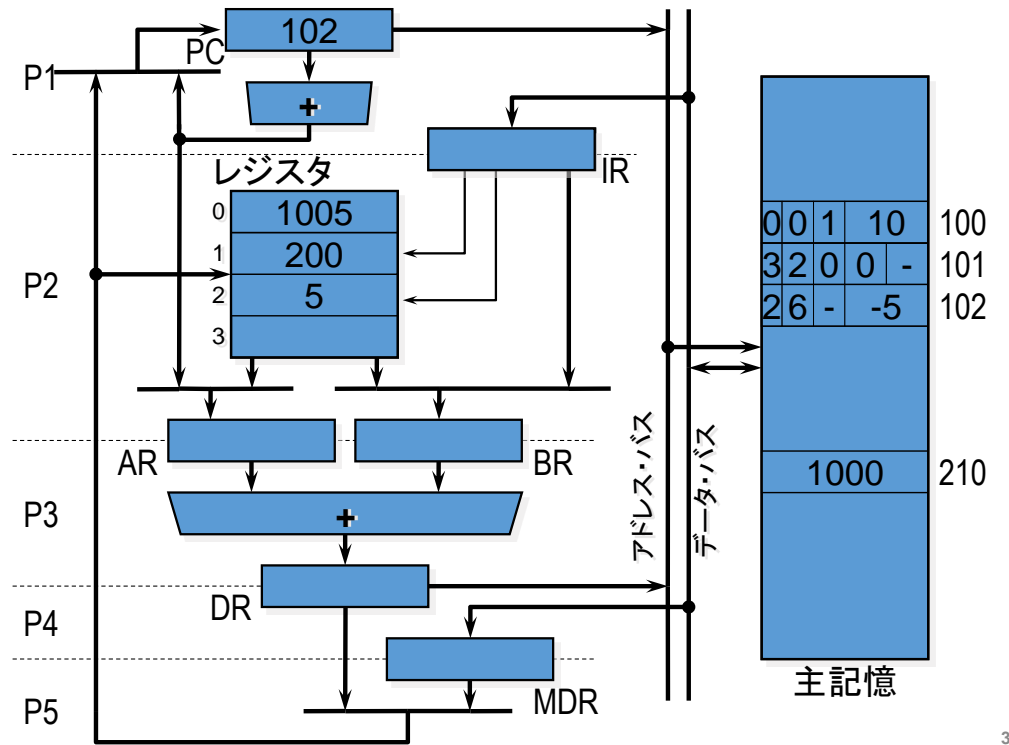


## 実行のサンプルの命令

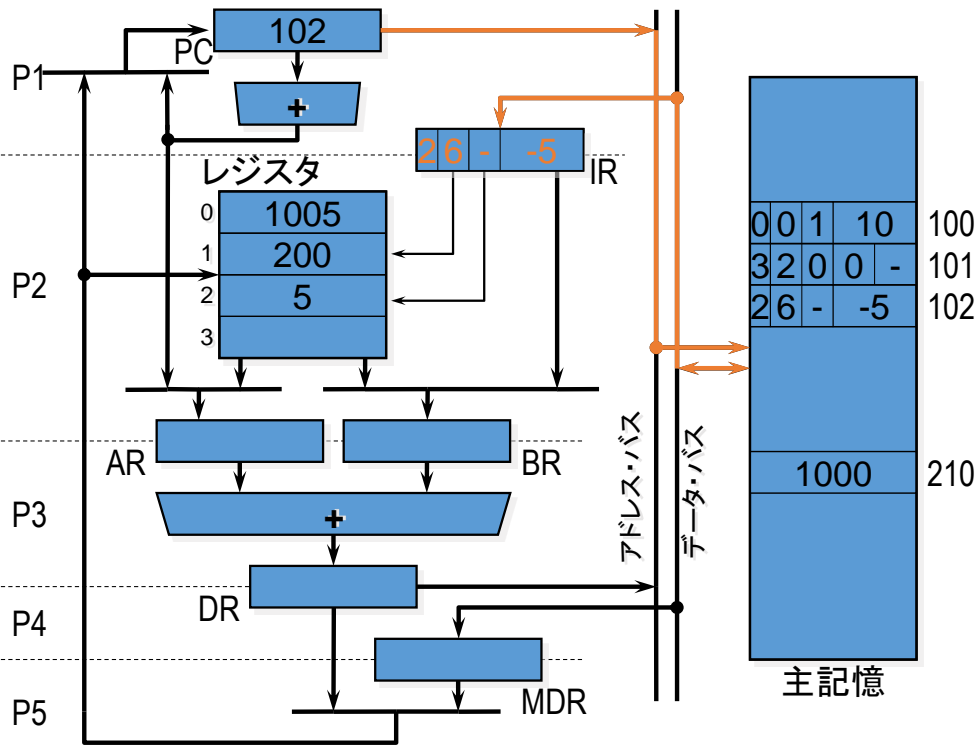
- 無条件分岐命令: プログラム・カウンタ102

• B-5

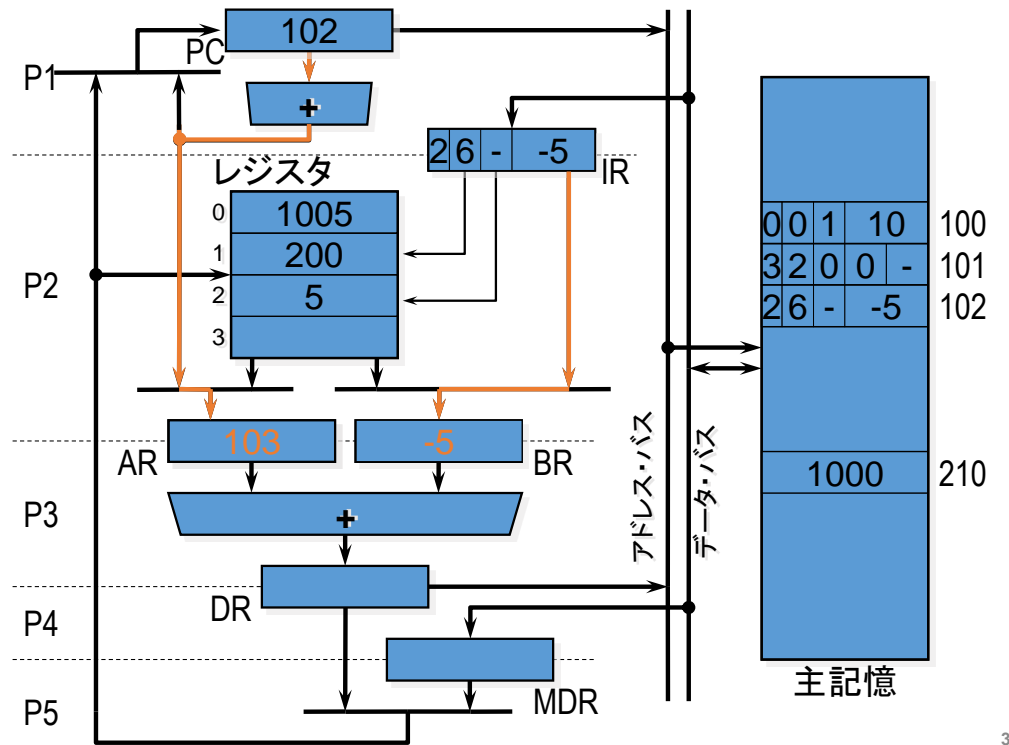
略記 26 - -5



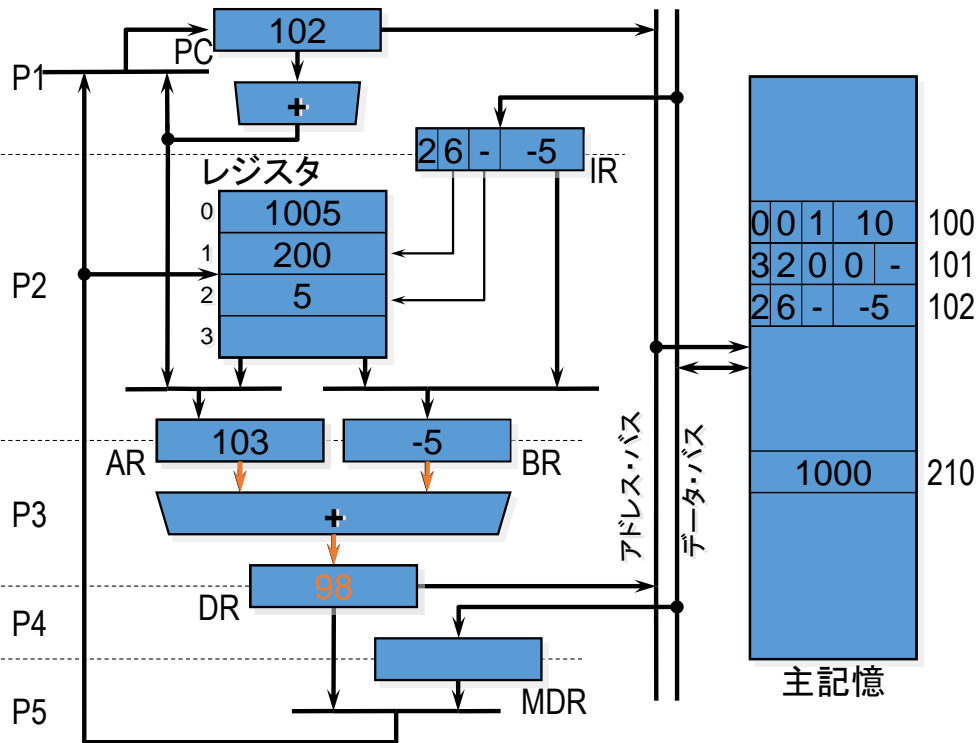




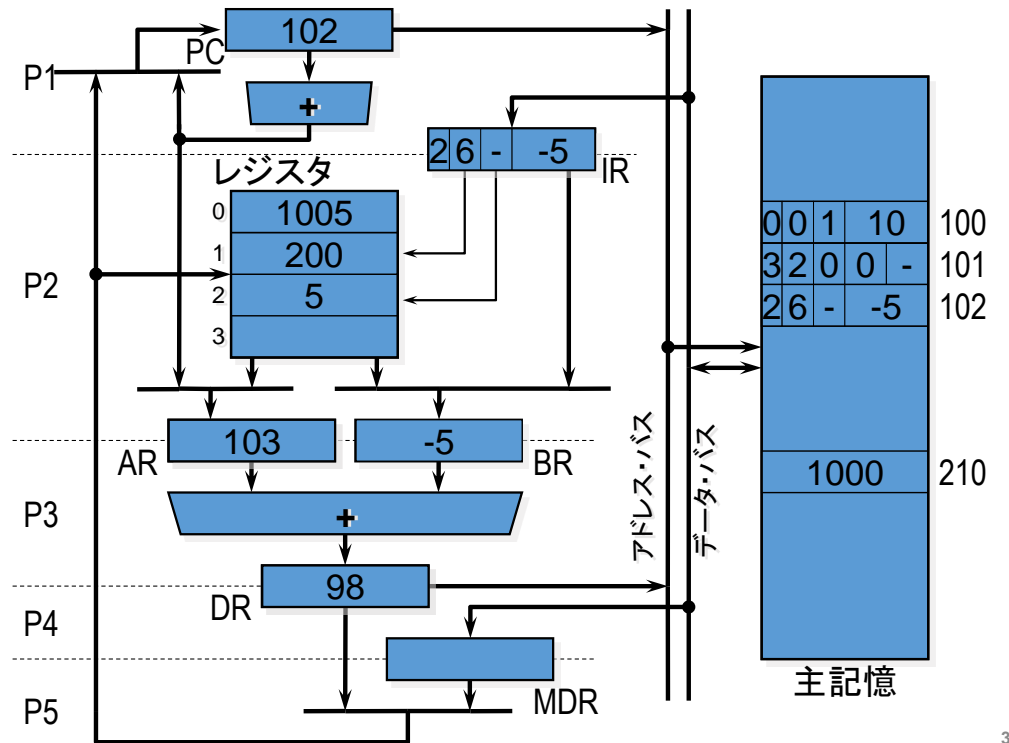
33



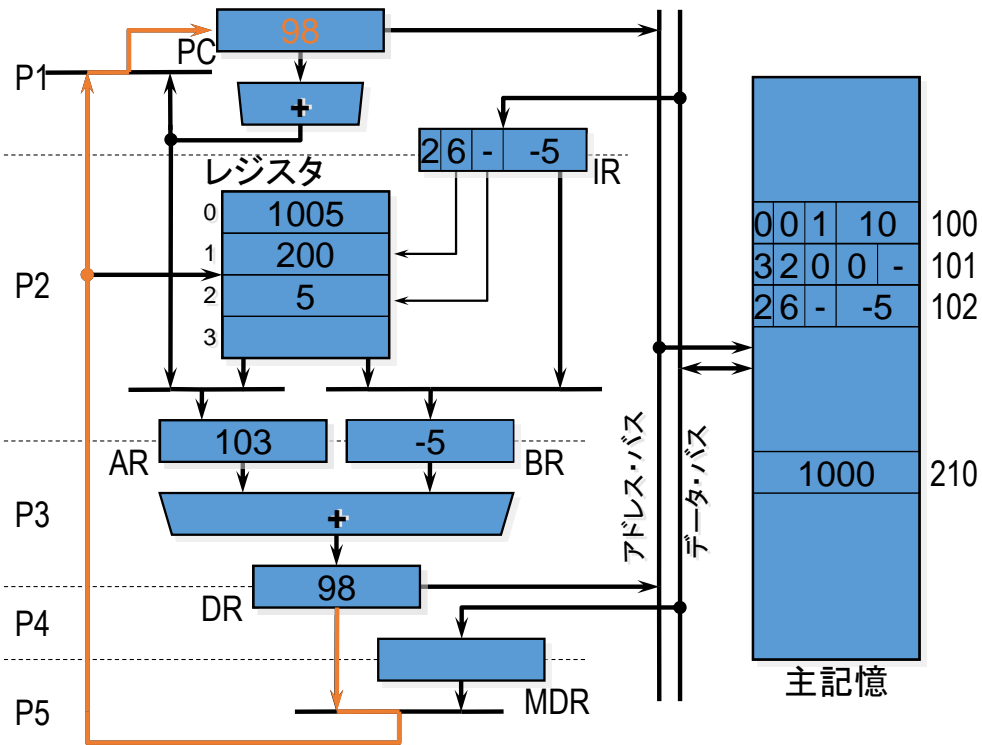
34



35



36



## 設計のヒント

38

## モジュール構成

- 全体をサブデザインに分割
  - どの論理を1つの単位にするか
  - Verilogのモジュール? ブロック?
  - 各レジスタはどの単位に属するか
- 分担
  - 制御系とデータパス系?
  - サブデザインとトップデザイン & インタフェース?
  - 基本機能と拡張機能?
  - 同じ機能ブロックの違うバージョンをそれぞれ設計?

39

## 検証環境

- シミュレーションテストベンチ
  - シミュレーション毎の手作業が減るように自動化
- 実機検証
  - ボードのSW/LEDを利用して内部信号をプローブ
  - プロセッサ本体の外側にテスト用回路(プローブ & スイッチ・表示系ドライバ)を構成

40

# 工程管理・スケジュール

- トップダウン？ボトムアップ？
  - 部品から作るか、トップデザイン(部品はダミー)をまず用意するか
- プロトタイピング、マイルストーン、線表
  - 「何らかの命令が動作」の時期と機能をどう設定するか
  - 最も単純な機能・構成から始めるか、機能拡張に備えた構成をまず考えるか
  - 最終成果物の仕様をいつ決めるか、いつ見直すか

41

# 課題とデモンストレーション

42

# 課題とデモンストレーション

- 導入課題： 個人単位
    - 基本的な回路の設計
    - レポートの執筆
  - プロセッサ設計演習： グループ単位
    - 中間レポート
      - 各種仕様書、考察等
    - 中間デモ
      - 何らかの命令が動作しているところを見せる
    - 最終デモ
      - 設計したプロセッサの特徴の説明
        - 「SIMPLE/Bに比べてこんな点がすぐれている」というセールストーク
        - 応用プログラムの実行
    - 最終レポート
      - 最終成果物のユーザーズマニュアル
      - SIMPLE/B基本仕様からの拡張および性能評価
      - 各コンポーネントの仕様書(最終版)、性能評価
      - 考察、感想
- どう設計したか、何を使ったか、レポートに明記。
  - 単に動くだけではなく、スピード、回路規模、消費電力の最適化を目指す。

# 機能拡張と性能評価

- 命令セット・アーキテクチャの改良
  - 新命令の追加、有用性の評価
- マイクロ・アーキテクチャの改良
  - フェーズの並列実行(パイプライン化)
- 何らかの拡張を行って、拡張前と比較する
  - プログラムの実行がどれだけ高速化したか？
    - 最高クロック周波数、実行命令数、実行サイクル数
  - 追加で必要となったハードウェアは？
    - ゲート数(LUT数)

44

# コンテスト

- 「俺のプロセッサはすごいで！」ということを証明したいあなたに...
- データをソートする時間を競うコンテスト(改定予定?)
  - データ
    - 16bitの符号付整数1024個
    - ランダム、昇順ソート済み、降順ソート済みの3種類
  - 時間の定義
    - 完了までのクロック・サイクル数×クロック周波数
    - 3種類のデータ各々の処理時間の平均値
- **ぜひ参加して、昨年度の記録を破ってください。**  
(旧ボードでも動作すれば、過去の最高記録とも比較可能。)