

# アドレッシング・モード

(教科書3.8節)

# アドレッシング・モードとは？

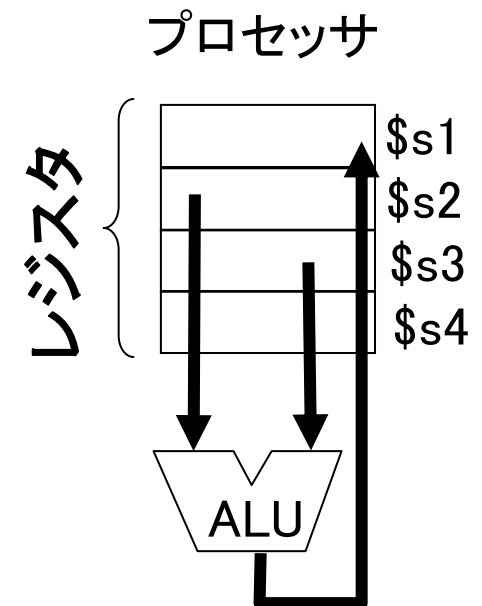
アクセスしたいオブジェクトに対するアドレスの指定方法  
(定数, レジスタ, メモリ番地)

例えば, MIPSの加算命令の場合

**add \$s1, \$s2, \$s3**

レジスタ\$s2の値と, レジスタ\$s3の値を  
加算して、レジスタ\$s1に格納する

**レジスタ・アドレッシング**



# MIPSのアドレッシング・モード

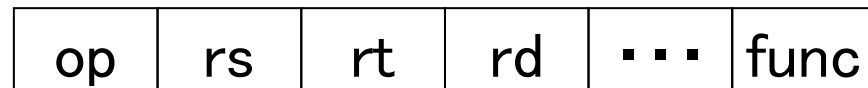
- レジスタ・アドレッシング
  - レジスタをオペランドとする
- ベース相対アドレッシング
  - 命令中に指定した定数とレジスタの和によって、オペランドが記憶されているデータのメモリ番地を表す
- 即値アドレッシング
  - 命令中に指定された定数をオペランドとする
- PC相対アドレッシング
  - PCと命令中に指定した定数との和によって、フェッチする命令のメモリ番地を表す
- 擬似直接アドレッシング
  - 命令中の26ビットとPCの上位ビットを連結することによって、フェッチする命令のメモリ番地を表す

# レジスタ・アドレッシング

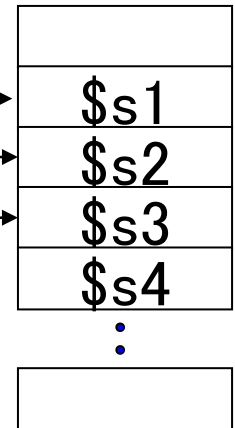
レジスタをオペランドとする

例) add \$s1, \$s2, \$s3

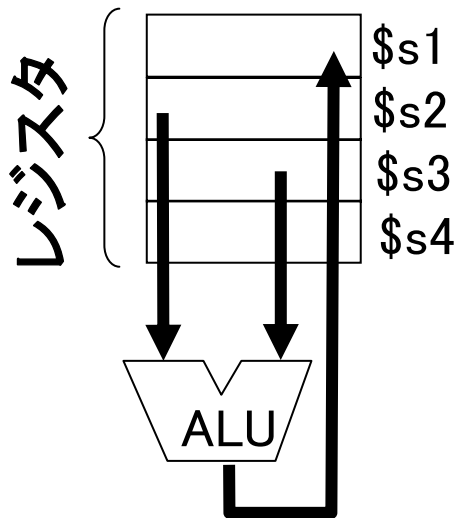
R形式命令



レジスタ



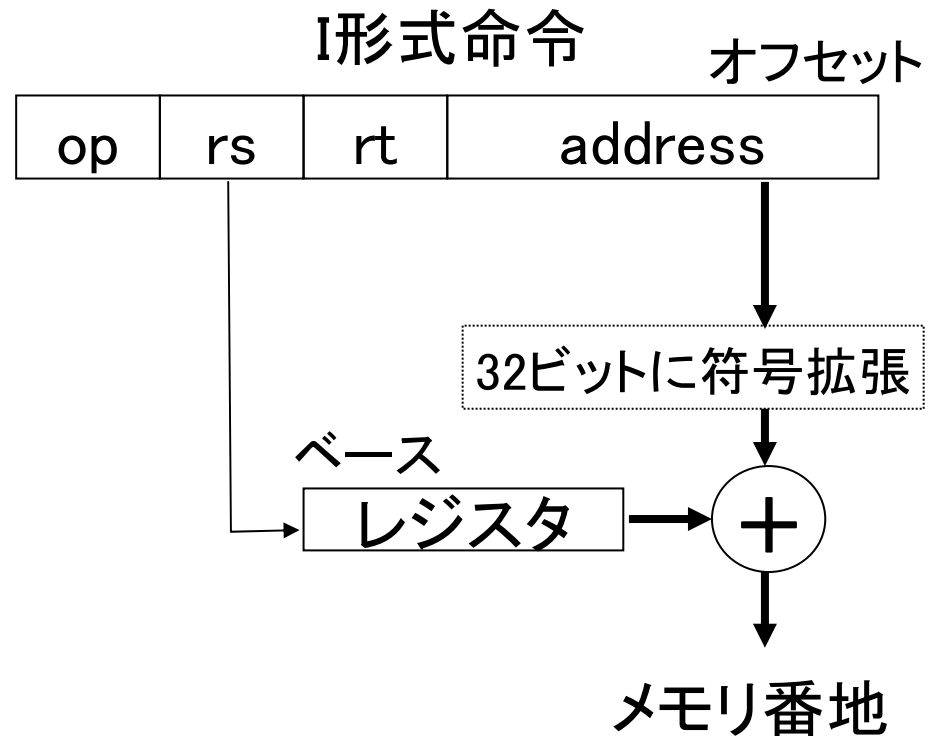
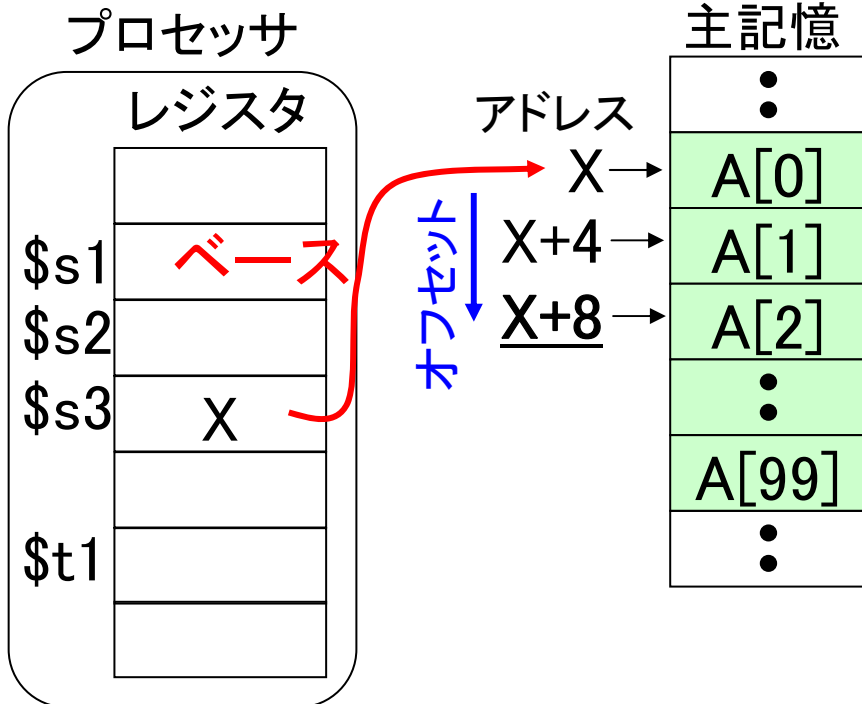
プロセッサ



# ベース相対アドレッシング

命令中に指定した定数とレジスタの和によって  
オペランドが記憶されているデータのメモリ番地を表す

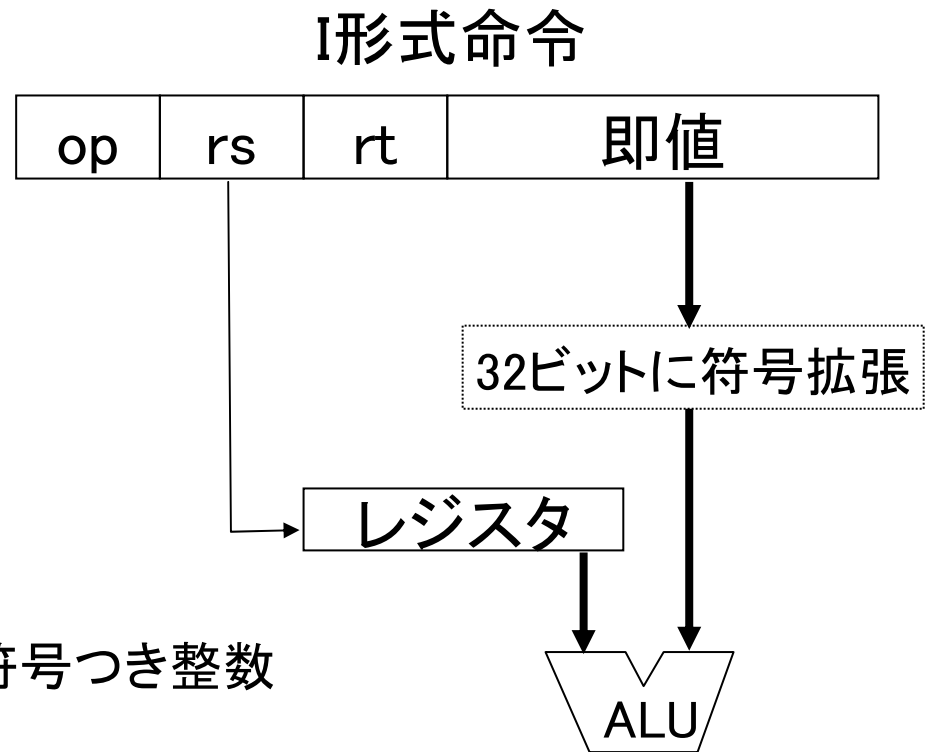
例) `lw $t1, 8($s3)`



# 即値アドレッシング

命令中に指定された定数をオペランドとする

例) `addi $s1, $s2, n`

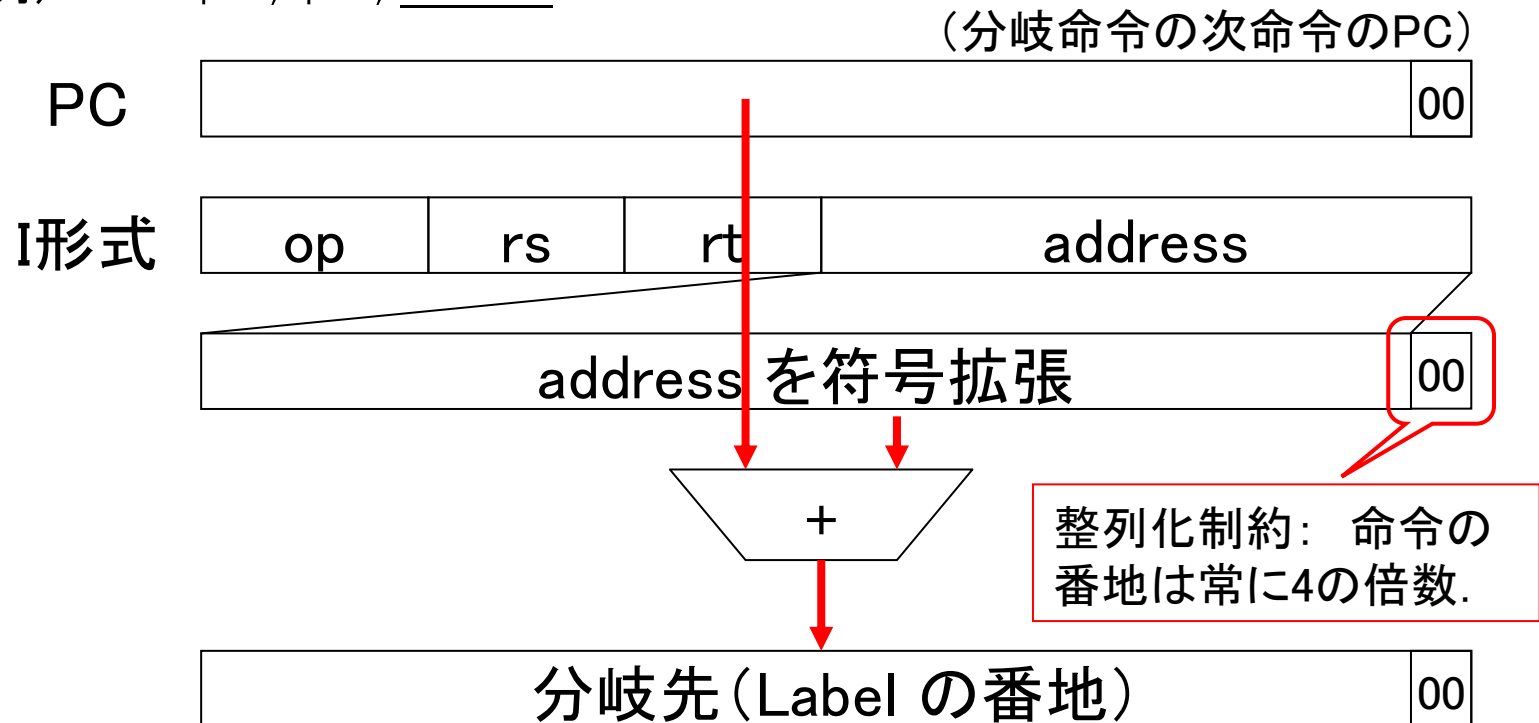


\* ) 定数nは16bit の2の補数表現の符号つき整数

# PC相対アドレッシング

PCと命令中に指定した定数との和によって  
フェッチする命令のメモリ番地を表す

例) `bne $s1, $s2, LableA`



# 擬似直接アドレッシング

命令中の26ビットとPCの上位ビットを連結することによって、フェッチする命令のメモリ番地を表す

例) j LableA

