MIPS: Εισαγωγή στο Binary System

5-Απρ-2021

Δυαδικοί αριθμοί Μη-προσημασμένοι

- Δυαδικός αριθμός n-bit:

$$x = x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \cdots + x_1 2^1 + x_0 2^0$$

- Αριθμητικό διάστημα: 0 to $+2^n - 1$
- Παράδειγμα:
  - $0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 1011_2$
    $= 0 + \ldots + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
    $= 0 + \ldots + 8 + 0 + 2 + 1 = 11_{10}$
- Διάστημα για 32 bits:
  - 0 to +4,294,967,295

Προσημασμένοι Ακέραιοι: 2s-Complement

- Δυαδικός αριθμός n-bit:

$$x = -x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \cdots + x_1 2^1 + x_0 2^0$$

- Διάστημα: -2^{n-1} to $+2^{n-1} - 1$
- Παράδειγμα:
  - $1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1100_2$
    $= -1 \times 2^{31} + 1 \times 2^{30} + \ldots + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$
    $= -2{,}147{,}483{,}648 + 2{,}147{,}483{,}644 = -4_{10}$
- Διάστημα για 32 bits
  - –2,147,483,648 to +2,147,483,647

Προσημασμένοι Ακέραιοι: 2s-Complement

- Bit 31 είναι το bit προσήμου
  - 1 για αρνητικούς αριθμούς
  - 0 για θετικούς αριθμούς
- $-(-2^{n-1})$ δεν μπορεί να αναπαρασταθεί

- Παραδείγματα:
  - 0: 0000 0000 … 0000
  - –1: 1111 1111 … 1111
  - Μικρότερος αρνητικός: 1000 0000 … 0000
  - Μεγαλύτερος θετικός: 0111 1111 … 1111

Προσημασμένη Άρνηση

- Complement and add 1
  - Συμπλήρωμα σημαίνει: 1 → 0, 0 → 1

$$x + \bar{x} = 1111\ldots111_2 = -1$$

$$\bar{x} + 1 = -x$$

- Παράδειγμα: negate +2
  - +2 = 0000 0000 … 0010_2
  - –2 = 1111 1111 … 1101_2 + 1
    = 1111 1111 … 1110_2

Επέκταση Προσήμου

- Αναπαράσταση ενός αριθμού με περισσότερα bits
  - Διατήρηση της αριθμητικής τιμής
- Με χρήση του σετ εντολών του MIPS
  - `addi`: extend immediate value
  - `lb`, `lh`: extend loaded byte/halfword
  - `beq`, `bne`: extend the displacement
- Αντιγραφή του bit προσήμου προς τα αριστερά:
  - Μη προσημασμένες τιμές (unsigned): επέκταση με 0
- Παραδείγματα: 8-bit σε 16-bit
  - +2: 0000 0010 => 0000 0000 0000 0010
  - –2: 1111 1110 => 1111 1111 1111 1110

Hexadecimal

- Base 16
  - Αναπαράσταση ακολουθιών από bit
  - 4 bits ανά hex ψηφίο

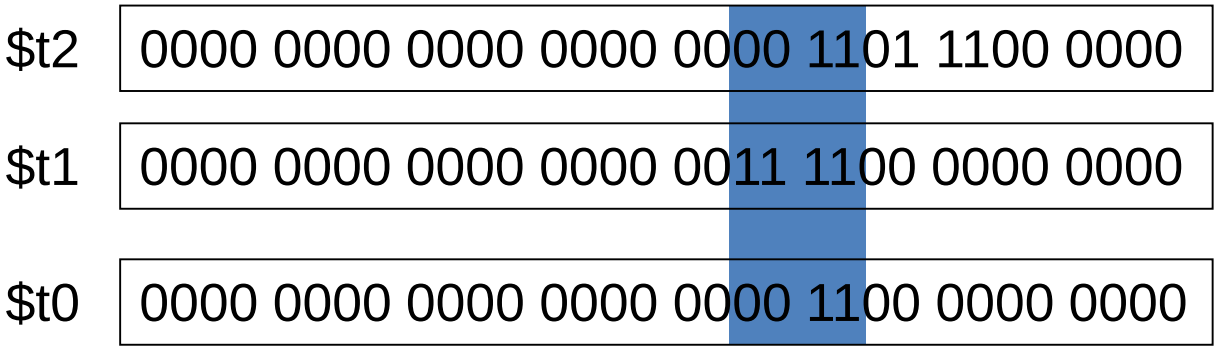
0	0000	4	0100	8	1000	c	1100
1	0001	5	0101	9	1001	d	1101
2	0010	6	0110	a	1010	e	1110
3	0011	7	0111	b	1011	f	1111

- Παράδειγμα: eca8 6420
  - 1110 1100 1010 1000 0110 0100 0010 0000

Λειτουργίες AND

- Χρήσιμη για την εφαρμογή mask σε συγκεκριμένα bits σε μία λέξη
  - Επιλογή μερικών bits, μηδενισμός των άλλων σε 0

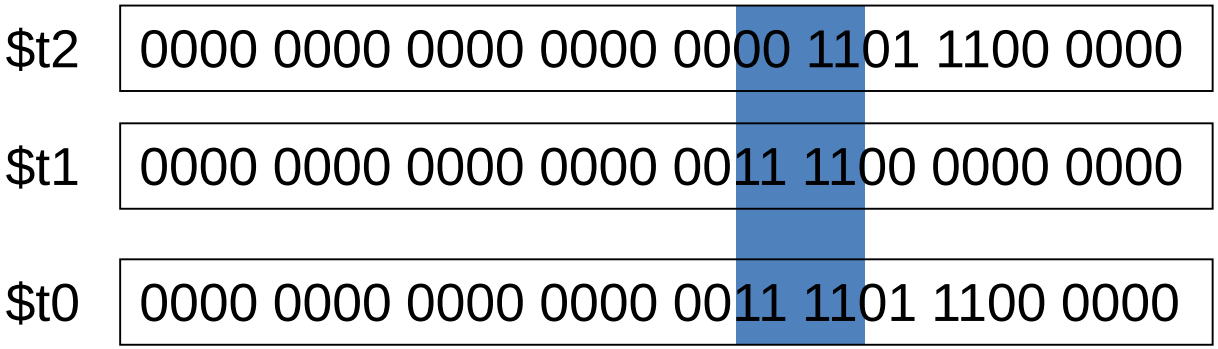
```
and $t0, $t1, $t2
```



Λειτουργίες OR

- Χρήσιμη για να εισάγουμε bits σε ένα word
  - Εξαναγκάζουμε κάποια bits σε 1, ενώ αφήνουμε τα υπόλοια ανεπηρέαστα

```
or $t0, $t1, $t2
```



Λειτουργίες NOT

- Χρήσιμη για να αντιστραφούν bits σε ένα word
  - αλλαγή 0 to 1, και 1 to 0
- Ο MIPS υποστηρίζει εντολή NOR 3-operand
  - a NOR b == NOT (a OR b)

```
nor $t0, $t1, $zero
```

Register 0: always read as zero

$t1	0000 0000 0000 0000 0011 1100 0000 0000
$t0	1111 1111 1111 1111 1100 0011 1111 1111

Δεδομένα Character

- Σετ χαρακτήρων Byte-encoded
  - ASCII: 128 characters
    - 95 graphic, 33 control
  - Latin-1: 256 characters
    - ASCII, +96 more graphic characters
- Unicode: 32-bit σετ χαρακτήρων
  - Used in Java, C++ wide characters, …
  - Most of the world's alphabets, plus symbols
  - UTF-8, UTF-16: variable-length encodings

Λειτουργίες Byte/Halfword

- Μπορούν να χρησιμοποιηθούν λειτουργίες bitwise
- MIPS byte/halfword load/store
  - String processing: συνήθης διαδικασία

`lb rt, offset(rs)` `lh rt, offset(rs)`

  - Επέκταση προσήμου (Sign extend to 32 bits in rt)

`lbu rt, offset(rs)` `lhu rt, offset(rs)`

  - Επέκταση με 0 (Zero extend to 32 bits in rt)

`sb rt, offset(rs)` `sh rt, offset(rs)`

  - Αποθήκευση μόνο το λιγότερο σημαντικό byte (rightmost byte/halfword)

Σταθερές 32-bit

- Οι περισσότερες σταθερές είναι μικρές
  - 16-bit immediate είναι συνήθως αρκετό
- Για (περιστασιακές) σταθερές 32-bit:

  `lui rt, constant`
  - Αντιγράφει τη σταθερά 16-bit στα αριστερά 16 bits του rt
  - Μηδενίζει τα 16 bits του rt σε 0