

## *Lab Session 5*

### **INTEGER ARITHMETIC.**

### **USING INSTRUCTIONS FOR MULTIPLICATION AND DIVISION**

# Multiplication and division instructions in MIPS

**mult**    **\$2, \$3:**     $\text{HI-LO} = \$2 * \$3$ ; Signed numbers

**multu** **\$2, \$3:**     $\text{HI-LO} = \$2 * \$3$ ; Unsigned numbers

**div**     **\$2, \$3:**     $\text{LO} \leftarrow \$2 / \$3$  quotient;     $\text{HI} \leftarrow \$2 \bmod \$3$ ; Signed num.

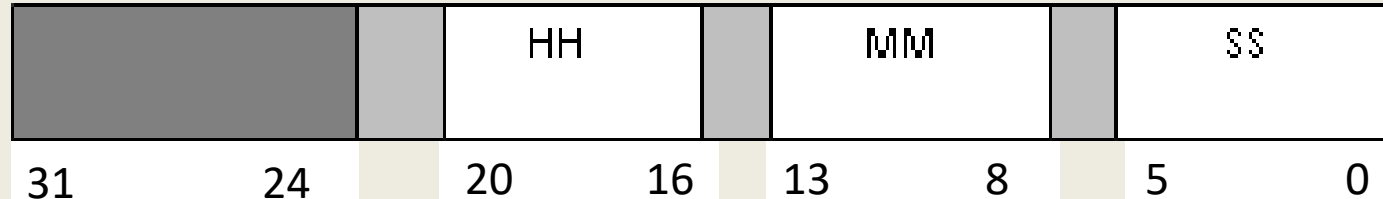
**divu**    **\$2, \$3:**     $\text{LO} \leftarrow \$2 / \$3$  quotient;     $\text{HI} \leftarrow \$2 \bmod \$3$ ; Unsigned num.

## Transferring results

**mfhi** **\$2**    :     $\$2 = \text{HI}$

**mflo** **\$2**    :     $\$2 = \text{LO}$

# Time format: HH:MM:SS



Giving the variable *reloj* implement two routines to initialize it:

1. Using a single input parameter:

NAME	INPUT PARAMETERS	Ex...
inicializa_reloj	\$a0: memory address for variable reloj \$a1: HH:MM:SS	reloj = HH:MM:SS

Passing parameters by reference

2. Using 3 input parameters:

NAME	INPUT PARAMETERS	OUTPUT
inicializa_reloj_alt	\$a0: memory address for variable reloj \$a1: HH \$a2: MM \$a3: SS	reloj =

Passing parameters by value

Condition: Force to 0 those bits not used in each field

Exercise 1: Routine that receives as input parameter the variable *reloj* (HH:MM:SS) and returns its value in seconds

1. Select a temporary register (e.g. \$t0 = 0)
2. Get HH and multiply\*3600 (byte access)
3. Add to \$t0
4. Get MM and multiply\*60
5. Add to \$t0
6. Get SS and add to \$t0
7. Return result in \$v0

## Exercise 2: Routine that receives as input parameter a value in seconds and returns the time in HH:MM:SS

- Method:

*value\_in\_seconds / 60*

- quotient = *value\_in\_minutes*

- mod = **SS**

*value\_in\_minutes / 60*

- quotient = **HH**

- mod = **MM**

# Exercise 2

- Procedure:
  - $\$a1 = \text{value\_in\_seconds}$
  - Divide:  $\$a1 / 60$
  - Store mod (HI) in SS (byte access)
  - Move quotient (LO) to a temporary register (Eg.  $\$t1$ )
  - Divide  $\$t1 / 60$
  - Store mod (HI) in MM (byte access)
  - Move quotient (LO) to HH
  - `j r ra`

Extra: Detect division by 0