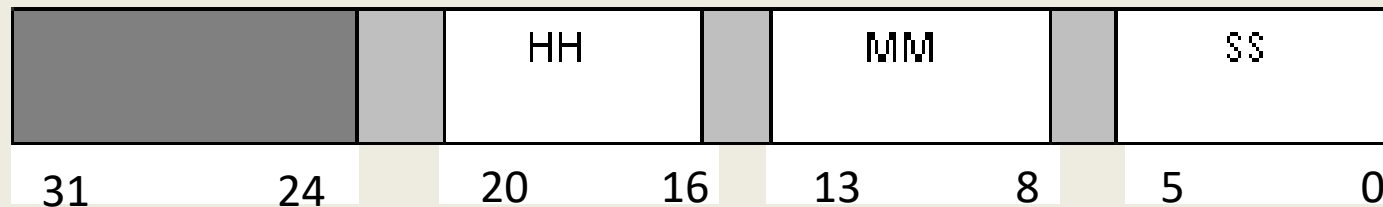


Lab Session 6

INTEGER ARITHMETIC

ADDITION, SUBTRACTIONS AND SHIFT

Multiplication using shifts, addition and subtraction instructions



Method to convert time in format HH:MM:SS into the equivalent seconds

$$\text{seconds} = \text{HH} * 3600 + \text{MM} * 60 + \text{SS}$$

$$3600 = 2^{11} + 2^{10} + 2^9 + 2^4$$

$$60 = 2^5 + 2^4 + 2^3 + 2^2$$

In both cases 4 shifts and
3 additions are needed

DESPLAZAMIENTO

sll rd, rt, shamt Desplazamiento logico a la izquierda R
Desplaza el registro rt a la izquierda tantos bits como indica shamt

sll \$t0, \$t1, 16 # \$t0 ← \$t1 << 16

srl rd, rt, shamt Desplazamiento logico a la derecha R
Desplaza el registro rt a la derecha tantos bits como indica shamt.

srl \$s0, \$t1, 4 # \$s0 ← \$t1 >> 4

Multiplication using shifts, addition and subtraction instructions

- Second option: Booth algorithm
- Using shift, addition and subtraction instructions
- 3600 and 60 can be recoded as:

$$3600 = 2^{12} - 2^9 + 2^5 - 2^4$$

$$60 = 2^6 - 2^2$$

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

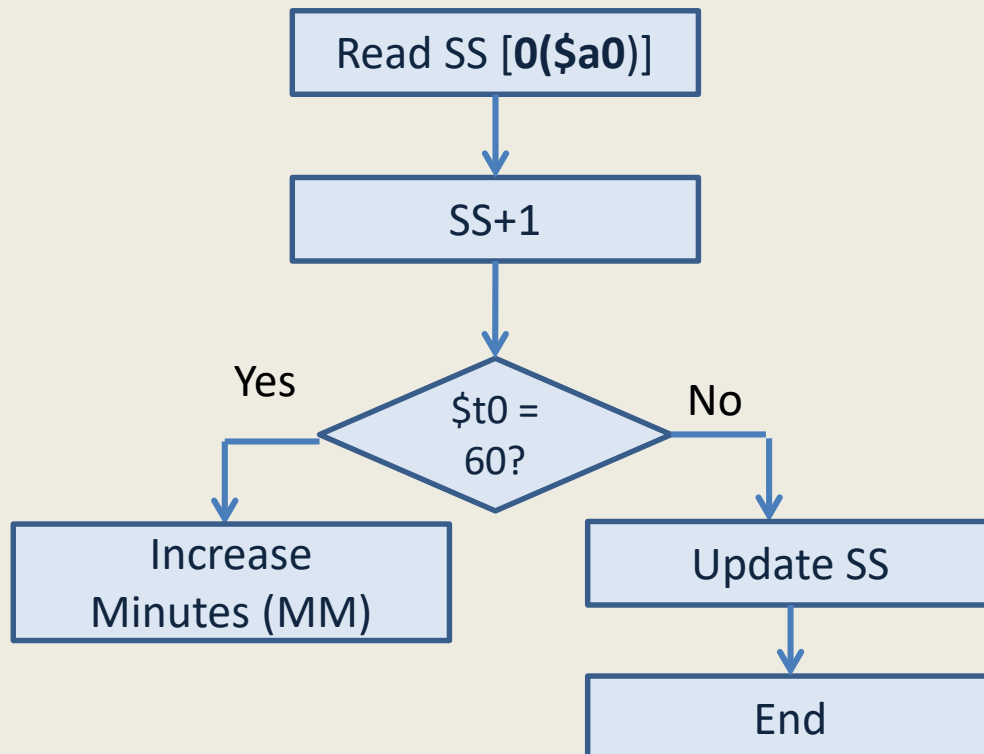
Redesign Session 5 routine using Booth algorithm for multiplication:

1. Select an accumulator register. Initialize it to 0
2. Get HH (byte access) and multiply by 3600
 - Use shift/addition/subtraction instructions
 - $3600 = 2^{12} - 2^9 + 2^5 - 2^4$
3. Accumulate into the selected register
4. Get MM (byte access) and multiply by 60
 - Use shift/addition/subtraction instructions
 - $60 = 2^6 - 2^2$
5. Accumulate into the selected register
6. Get SS (byte access) and accumulate
7. Return result in \$v0

Exercise 2: Increase time

\$a0 = memory address of reloj

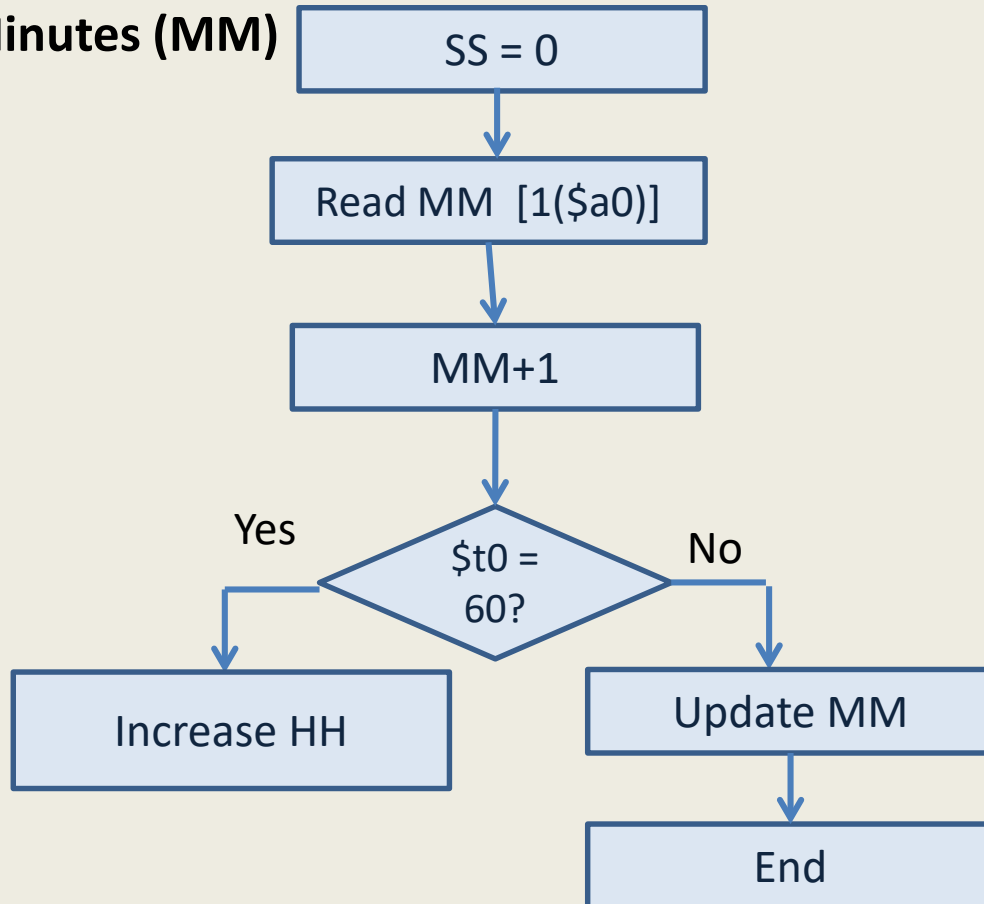
Increase Seconds (SS)



Exercise 2: Increase time

\$a0 = memory address of reloj

Increase Minutes (MM)



Exercise 2: Increase time

\$a0 = memory address of reloj

Increase Hours (HH)

