

# Puzzle modos de direccionamiento

## Motivando los modos de direccionamiento relativos e indirecto

*R0 ... R15*

Tenemos un procesador con 16 registros de propósito general (R0 ... R15) (el registro R0 contiene la constante 0 y no se puede cambiar), y en su conjunto de instrucciones encontramos las siguientes:

- **ADDi Rd, Rf, #imm**
  - Instrucción de suma:  $Rd \leftarrow Rf + imm$
  - Los operandos Rd y Rf tienen modo de direccionamiento directo a registro.
  - El operando imm tiene modo de direccionamiento inmediato y representa un número entero.
- **ADD Rd, Rf1, Rf2**
  - Instrucción de suma:  $Rd \leftarrow Rf1 + Rf2$
  - Los tres operandos tienen modo de direccionamiento directo a registro.
- **LD Rd, Addr**
  - Instrucción de carga (load):  $Rd \leftarrow Mem[Addr]$
  - El operando destino (Rd) tiene modo de direccionamiento directo a registro.
  - El operando fuente (Addr) tiene modo de direccionamiento directo a memoria.
- **BEQ Rf1, Rf2, Addr**
  - Instrucción de salto condicional (branch): if ( $Rf1 = Rf2$ )  $PC \leftarrow Addr$
  - Los operandos Rf1 y Rf2 tienen modo de direccionamiento directo a registro.
  - El operando Addr tiene modo de direccionamiento inmediato.
- **J Addr**
  - Instrucción de salto incondicional (jump):  $PC \leftarrow Addr$
  - El operando Addr tiene modo de direccionamiento inmediato.

En la memoria principal de dicho procesador se encuentran almacenadas las notas de los 5 parciales que has realizado de la asignatura "Adoro los Computadores (ALC)". La nota del primer parcial se encuentra almacenada en la posición 20 de memoria, y las del resto de parciales en posiciones de memoria consecutivas a esta (21, 22, 23 y 24).

La nota final de la asignatura (entre 0 y 100) se calcula sumando la nota obtenida en cada uno de los 5 parciales (entre 0 y 20 cada uno).

Tenéis que hacer un programa, usando el ensamblador de dicho procesador, que tras ejecutarse deje en el registro R10 vuestra nota total obtenida por parciales en ALC.

El profesor os anima a que pongáis especial cuidado en hacer correctamente el programa, ya que en ello va vuestra nota ;)

nota 1 = 5    nota 5 = 20

nota 2 = 10

nota 3 = 8

nota 4 = 15

## Programa ensamblador: Cálculo de nota ALC

```
ADD r10, r0, #0
```

```
LD r1, #20
```

```
LD r2, #21
```

```
LD r3, #22
```

```
LD r4, #23
```

```
LD r5, #24
```

```
ADD r1, r1, r1
```

```
ADD r1, r1, r2
```

```
ADD r1, r1, r3
```

```
ADD r1, r1, r4
```

```
ADD r1, r1, r5
```

```
ADD r10, r1, r0
```

Una vez terminado y repasado el programa, intercambiarlo con vuestro grupo compañero, estudiad la solución propuesta por el grupo compañero, anotando preguntas, errores, mejoras, etc. que se os ocurran para posteriormente ponerlo en común los dos grupos juntos.

Aparte de debatir estos aspectos que habéis anotado del código del grupo compañero, aquí os dejamos un par de preguntas para debatir entre todos y llegar a alguna conclusión consensuada.

1. ¿Ha sido fácil calcular correctamente vuestra nota? ¿Sería igual de fácil hacer el programa si en vez de 5 parciales hubiera 50? (Ojo, que el profesor de ALC está “chapado a la antigua” y no os deja usar `ctrl-c` `ctrl-v` ni `cmd-c` `cmd-v` a los maqueros) ¿Qué cambios tendríais que introducir en el código?

Con 50 eoad distinto

2. Si sois de los que tenéis fe y habéis contestado que “sí” a la pregunta anterior, a ver qué os parece esta ... ¿y si en vez de 50 parciales, hubiera un número de parciales no conocido a priori, sino que el profesor almacena dicho número en la posición de memoria 19? (se supone que podéis hacer parciales sin ni siquiera daros cuenta, de ahí que el número de parciales sea desconocido para vosotros).

Un bucle.