

Nombre y Apellidos: Carlos Javier Hellín Asensio

Grupo: GII Tarde

### **Cómo determinar el acarreo en números naturales en máquinas sin registros de estado**

Para entender cómo determinar el acarreo se ha usado un software llamado SPIM que simula la arquitectura MIPS de 32-bit que no tiene registros de estado y permite ejecutar programas en ensamblador.

Primero se ha inspeccionado el ISA del MIPS para encontrar las instrucciones de suma para números naturales (sin signo) y son las siguientes:

`addu rd, rs, rt` ; Cuya operación es  $rd = rs + rt$

`addiu rd, rs, inm` ; Cuya operación es  $rd = rs + inm$

Siendo los operandos `rd` como registro de destino, `rs` registro fuente, `rt` el segundo registro fuente e `inm` valor inmediato.

Se programa el siguiente código en ensamblador y se carga en el simulador SPIM:

```
.data
    numero: .word 0xFFFFFFFF

.text

.globl main

main:
    lw $t1, numero
    addiu $t0, $t1, 0xFFFFFFFF
```

En datos tenemos número que es una palabra que contiene el valor 0xFFFFFFFF en hexadecimal. El programa principal lo que hace es cargar esa misma palabra al registro `t1` y usando la instrucción de suma `addiu`, sumamos el contenido del registro `t1` y el valor inmediato de 0xFFFFFFFF para guardar dicho resultado en el registro `t0`. El resultado debería ser 0x1FFFFFFFE pero al observar el registro `t0` del simulador se obtiene 0xFFFFFFFEE y no nos ha avisado del acarreo al no tener ningún tipo de registros de

estado. Por lo tanto, es trabajo del programador comprobar si hay acarreo o no. Una solución posible sería la siguiente:

```
.data
```

```
    numero: .word 0xFFFFFFFF
```

```
.text
```

```
.globl main
```

```
main:
```

```
    lw $t1, numero
```

```
    addiu $t0, $t1, 0xFFFFFFFF
```

```
    sltu $t2, $t0, $t1
```

Las instrucciones de comparación pueden usarse para generar el flag de acarreo. Después de sumar dos valores, si la suma es menor que cualquiera de las entradas, entonces hay acarreo en el bit más significativo.

Por ello, con la instrucción `sltu` estamos comprobando si el contenido del registro `t0` (el resultado de la suma) es menor que el contenido del registro `t1` (el número que hay en datos) y si es así el registro `t2` valdría 1 o 0 en el caso contrario. De esta forma se tendría un propio flag de acarreo en un registro y el programador podría comprobarlo para realizar un salto o lo que vea conveniente.