



## EXAMEN DE PRUEBA - FUNDAMENTOS DE COMPUTADORES II 3 DE MAYO DE 2023

Los conversores analógico digitales (ADCs) son ampliamente utilizados en la industria electrónica. Los parámetros físicos (temperatura, humedad, luz, etc) no tienen valores discretos, sino que ocurren en un continuo. Para entenderlo un ordenador, se debe *digitalizar* esta información.

Un ejemplo de este tipo de proceso lo podemos ver en las básculas de comida, que incorporan los ADC HX711. El sensor de peso es una resistencia deformable. El ADC lee su valor, que transforma a gramos.

El funcionamiento es sencillo: El dato resultante (peso) será un entero sin signo de 24 bits. Para leerlo, el sensor comparte la memoria con el procesador. Debemos enviar una secuencia de control (0 seguido de un 1) a la dirección de control 0x1000, y leeremos a continuación un bit del dato (de más a menos significativo) desde la dirección 0x1004. Tras repetir este proceso 24 veces, tenemos el dato completo.

Se proporciona un pseudocódigo en RISC-V con una función `lectura_sensor` que implementa esta funcionalidad:

```
.equ DIR_CTRL, 0x1000
.equ DIR_DATA, 0x1004

.text
lectura_sensor:
    //Prólogo de la función
    //...
    ///////////////////////////////////

    li s0, DIR_CTRL
    li s1, DIR_DATA
    li s2, 24          //número de iteraciones

    li t0, 0          //iterador
    li t1, 1          //constante 1 para utilizar

    li a0, 0          //valor resultado
bucle:
    beq t0, s2, fin_bucle
    sw zero, 0(s0)    //mando un 0 a control
    sw t1, 0(s0)      //mando un 1 a control
    slli a0, a0, 1    //desplazo a0 un bit a la izquierda y hago hueco al nuevo
    lw a1, 0(s1)      //leo el bit desde la dirección de datos
    or a0, a0, a1     //añado el bit al número que ya tengo
    add t0, t0, 1     //incremento iterador
    j bucle           //reinicio bucle

fin_bucle:
    //Epílogo de la función
    //...
    ///////////////////////////////////
```

Responda a las siguientes preguntas:

1. **ASM [1pt]** Escriba el prólogo y el epílogo de la función proporcionada.
2. **ASM [0.5pt]** ¿En qué cambiaría el apartado anterior, si la función `lectura_sensor` llamara a su vez a otra?
3. **ASM [0.5pt]** Si se llama a la función, ¿de qué registro se recogería el resultado? ¿Respeto esto el estándar de llamadas RISC-V?
4. **CPU [1.5pt]** Supongamos que se ejecuta una iteración del bucle (8 instrucciones) en el procesador segmentado RISC-V. Realice el cronograma indicando qué riesgos de datos y control existen.
5. **CPU [0.5pt]** ¿Puedo eliminar alguno de los riesgos reordenando el código? ¿Cuál/es?
6. **CPU [1pt]** Asuma que ejecutamos el bucle completo (con todas sus iteraciones) en el procesador monociclo, multiciclo y segmentado. ¿Cuántos ciclos tarda en cada caso en ejecutarse?
7. **CPU [1pt]** Calcule el CPI de los tres procesadores. **NOTA:** Si no averiguó el número de ciclos, puede asumir que son 187 para el monociclo, 765 para el multiciclo, y 271 para el segmentado.
8. **CPU [1pt]** Si partimos de que el ciclo de reloj es de  $27,6ns$  para el monociclo,  $9,8ns$  para el multiciclo y  $10,5ns$  en el segmentado, ¿Cuánto tarda cada uno en ejecutar el bucle completo?
9. **MEM [1pt]** Asumamos que para nuestro procesador tenemos una memoria principal de 1MB, y una caché de 128B con bloques de 16B. Indique el formato de las direcciones para memoria caché.
10. **MEM [1pt]** En el procesador RISC-V con la caché indicada anteriormente ejecutamos el bucle completo con todas sus iteraciones. Si la instrucción `beq` está en la dirección `0x50`, ¿Cuál es la tasa de fallos de caché?
11. **MEM [1pt]** Comenta qué ocurriría si la instrucción `beq` estuviera en la dirección `0x80`. ¿Habrá más o menos fallos de caché? Explique por qué.