Arquitectura de Computadoras — Parcial 2- Fecha 1 – 14/11/2023 — TEMA 3

Apellido: Nombre: DNI:

1. Analizar el siguiente programa, que se ejecuta con **forwarding, BTB y delay slot deshabilitados**:

```
        .data
tabla:  .word 10,18,22,45,63
        .code
        daddi  $t1, $zero, 5
        daddi  $t2, $zero, 0
        daddi  $t3, $zero, 0
loop:   sd     $t2, tabla($t3)
        daddi  $t1, $t1, -1
        daddi  $t2, $t2, 7
        daddi  $t3, $t3, 8
        bnez   $t1, loop
        halt
```

a) ¿Qué hace el programa anterior? _____ (5 ptos)

b) ¿Cuántas veces se ejecutará la instrucción **halt?** _____ (5 ptos)

c) ¿Cuántas veces se ejecutará la instrucción **halt** si el delay slot estuviera **habilitado?** _____ (5 ptos)

d) ¿Cómo modificaría el código para ejecutar correctamente el programa con delay slot habilitado? (5 ptos)

_____

_____

2. La instrucción **ld $t1, 5($t2)** se ejecuta **con forwarding activado.** ¿Cuándo estará disponible para una instrucción posterior el resultado de la operación? Indique luego de qué etapa estará disponible. _____ (5 ptos)

3. El siguiente programa pinta un cuadrado de color rojo de 5x5 píxeles en el extremo inferior derecho de la pantalla gráfica. Completar instrucciones faltantes (5 ptos c/instrucción)

```
        .data
X:          .byte 45
Y:          .byte 0
color:      .byte 255, 0, 0, 0
CONTROL:    .word32 0×10000
DATA:       .word32 0×10008
        .code
        lwu $s0, CONTROL($0)
        lwu $s1, DATA($0)
        lwu $t0, color($0)
        sw $t0, 0($s1)
        _____
        lbu $t2,X($0)
        daddi $t4,$0,50
        daddi $t5,$0,5
loop:   sb $t1,4($s1)
        _____
        daddi $t3, $0, 5
        sd $t3,0($s0)
        daddi $t2, $t2, 1
        bne $t4,$t2,loop
        _____
        daddi $t1,$t1,1
        bne $t5, $t1, loop
        halt
```

4. Indicar cual de las siguientes opciones es la correcta para el winMips (5 ptos).

- ☐ El winMips tiene flags para ver si el resultado fue cero o negativo.
- ☐ El winMips tiene un flag de punto flotante.
- ☐ El winMips tiene registros de flags.
- ☐ En el winMips no se verifica ningún flag porque todos los saltos son incondicionales.

5. Codificar una subrutina **PROCESAR_CADENA** que reciba como parámetro las direcciones de dos cadenas de caracteres ("cadena" y "digitos") y convierta cada dígito entre 0 y 8 de "cadena" en dígito + 1. Además, debe incorporar cada dígito cambiado a la cadena "digitos".

Ejemplo: Si cadena = "3a?7eN 95 en1234a", el resultado de PROCESAR_CADENA debe ser:

cadena = **"4a?8eN 96 en2345a"**

digitos = **"4862345"** (sólo contiene los dígitos que se tuvieron que convertir).

A su vez, para implementar **PROCESAR_CADENA** se deben generar e invocar a las siguientes subrutinas:

- **ES_DIGITO**, que recibe como parámetro un carácter y retorna 1 si es un dígito entre 0 y 8, caso contrario retorna 0.
- **OBTENER_DIGITO** que recibe un carácter dígito entre 0 y 8 y devuelve el dígito + 1 correspondiente.

Rango Ascii dígitos: entre 30h y 39h (48 a 57 en decimal)

Por último, generar un programa que invoque a **PROCESAR_CADENA** enviando los parámetros correspondientes y luego imprima cada cadena con un mensaje de encabezado mediante dos invocaciones a una subrutina llamada **IMPRIMIR**, que recibe la dirección del mensaje de encabezado y la dirección de la cadena a imprimir. Es decir, antes de imprimir la cadena se debe mostrar el mensaje "cad_msg" ("Cadena con reemplazos: ") y antes de imprimir la cadena de dígitos convertidos se debe mostrar el mensaje "digi_msg" ("Dígitos convertidos a Dígito + 1:"). **Se debe usar la convención para nombrar a los registros. (55 ptos.)**

```
.data
cad_msg:    .asciiz "Cadena con reemplazos: "
digi_msg:   .asciiz "Dígitos convertidos a Dígito + 1: "
cadena:     .asciiz "3a?7eN 95 en1234a"
digitos:    .asciiz ""
```