

Apellidos _____

Nombre _____

DNI _____



Examen de Fundamentos de los computadores. Área de Arquitectura y Tecnología de Computadores

1er parcial. O: 13-03-2004

A

Instrucciones generales para la realización de este examen

La respuesta debe escribirse en el hueco existente a continuación de cada pregunta **con letra clara**.

Cada respuesta correcta suma un punto. Cada respuesta incorrecta, ilegible o vacía no suma ni resta. El total de puntos se dividirá entre el total de preguntas y se multiplicará por 10 para obtener la nota del examen.

Se sabe que las entradas de la ALU de 16 bits vista en clase tienen los siguientes valores:

- En el **operando A** se encuentran los 16 bits más altos de la codificación del número 2^{-128} en formato IEEE-754 simple.

- En el **operando B** se encuentra la combinación de bits que representa el cero en formato exceso a Z con 16 bits y exceso central.

- CarryIn = 0.
- Resta = 0.
- Op1 = 1 y Op0 = 0.

Se recuerda que las ALUs elementales que componen la ALU de 16 bits tienen conectadas las salidas de una puerta AND, de una puerta OR, de una puerta XOR y el resultado de un sumador elemental, a las entradas e_0 , e_1 , e_2 y e_3 del multiplexador, respectivamente.

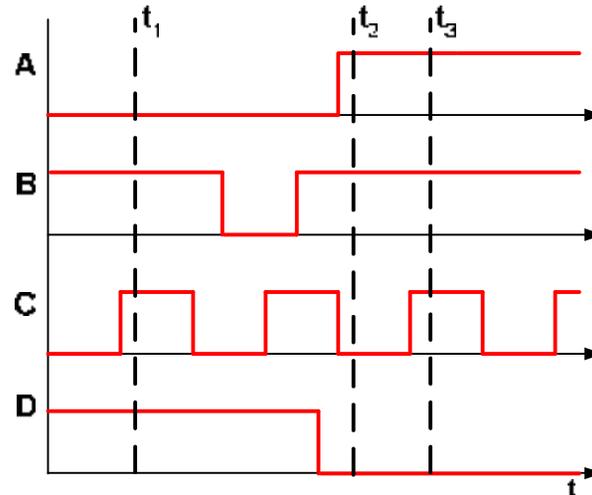
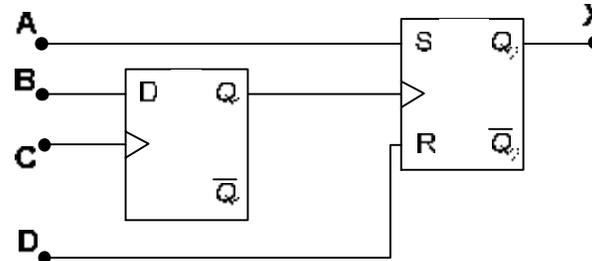
- ¿Cuál es el resultado obtenido en la salida? Expresar el resultado con **cuatro dígitos hexadecimales**.

8020h

- ¿Cuál es el error cometido al representar en formato IEEE-754 simple el número $2^{21} + 0,625$? Expresar el resultado **en decimal**.

0,125

- Dado el sistema digital mostrado a continuación, siendo sus entradas A, B, C y D y su salida X, calcular el valor de la salida X en los instantes t1, t2 y t3. Se sabe que A, B, C y D evolucionan tal y como muestra el cronograma y que en el instante inicial ambos biestables tienen sus salidas (Q_1 y Q_2) a cero.



t1: 0 t2: 0 t3: 1

- En un formato de coma flotante de 5 bits, se utilizan 3 bits para representar el exponente. La mantisa se representa en formato signo magnitud, normalizada a todo fracción con bit implícito, y el exponente en exceso a Z central. **Representa el rango de representación del formato en decimal.**

Rango: [-6 , 6]

- Se ejecuta en la CPU teórica la instrucción SUB R0, R1, R2. Se sabe además que, justo antes de la ejecución de la instrucción, los bytes más significativos de R1 y R2 estaban a cero; para R1, el byte menos significativo contenía el resultado de un XOR entre los códigos ASCII de los caracteres 'v' y 'V'; y, para R2, contenía el resultado de un XOR entre los códigos ASCII de los caracteres 'W' y 'w'. ¿Cuál es el valor obtenido en R0 (en hexadecimal) y cuál es el valor de los bits del registro de estado tras ejecutar la instrucción?

Resultado: 0h Z: 1 C: 0 O: 0 S: 0

- Se desea ampliar la ALU para que, además de las 4 operaciones habituales (las lógicas AND, OR y XOR y la aritmética de suma -o resta, si está activada dicha línea-), permita efectuar otras 39 operaciones más. ¿Cuántas líneas de selección de operación debería como mínimo tener esta ALU? Contestar en decimal.

6

- Completa los caracteres que faltan en el interior de las casillas, teniendo en cuenta los caracteres que ya hay y sus códigos ASCII.

| | | | | | |
|-----|-----|-----|-----|-----|-----|
| 41h | 51h | 6Ch | 6Ah | 30h | 34h |
| A | | | j | | 4 |



- Describe las dos salidas de un sumador elemental de 1 bit como suma de minterms. El orden de las entradas que se supondrá para evaluar ambas sumas de minterms será A, B y Cin. Ejemplo: $S = m_1 + m_9$ y $Cout = m_0 + m_2 + m_4$.

$$S = m_1 + m_2 + m_4 + m_7 \quad Cout = m_3 + m_5 + m_6 + m_7$$

Se pretende crear una nueva instrucción para la CPU teórica que permita sumar un registro de la CPU con el contenido de una posición de memoria y almacenar el resultado en la memoria. El mnemónico de la nueva instrucción es:

ADDM [Ri₂], Rs, [Ri₁] ; [Ri₂] ← Rs + [Ri₁]

Completar la siguiente tabla con las señales de control necesarias en cada paso para implementar esta instrucción.

| | |
|---|--|
| 4 | Ri ₁ -IB, IB-MAR, LEER |
| 5 | Rs-IB, IB-TMPE |
| 6 | ADD, MDR-IB, ALU-TMPS, ALU-SR |
| 7 | TMPS-IB, IB-MDR |
| 8 | IB-MAR, Escribir, Ri ₂ -IB, |
| 9 | FIN |

En una CPU se sabe que el siguiente fragmento de código ha tardado 24 microsegundos en ejecutarse:

```
atrás: MOVL R6, 0FEh
      MOVH R6, 0FFh
      INC R6
      BRC alli
      JMP atrás
alli: XOR R6, R6, R6
```

- ¿Cuál es la frecuencia, en MHz, del reloj de la CPU?
2 MHz

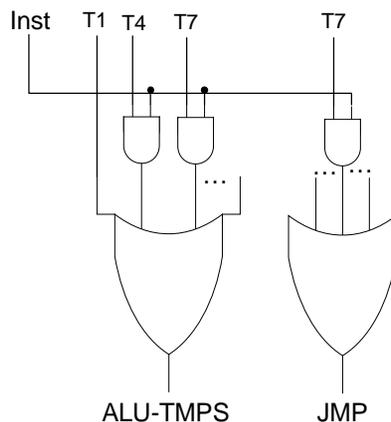


— ¿Cuáles de las siguientes afirmaciones son FALSAS? (Puedes responder "ninguna" si así lo consideras.)

- En las normas ISO de representación de caracteres, los caracteres asignados a los códigos iguales o superiores a 128 son siempre los mismos.
- Con la norma IEEE-754 de representación de números reales en precisión simple, la cantidad total de números que se pueden representar es igual a $2^{32}-1$.
- El bit de overflow activo en una operación de suma de números enteros en complemento a 2 significa que el resultado ha sido incorrecto.
- Los chips de RAM estática reciben ese nombre porque mantienen la información que almacenan aunque se retire la alimentación.

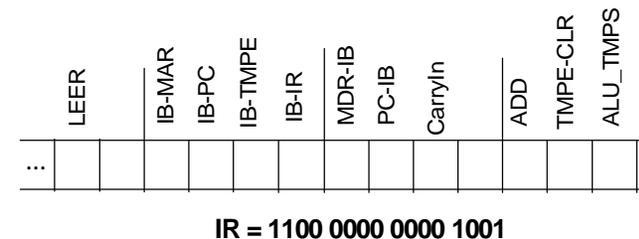
A, B, D

El siguiente circuito muestra la parte de la UC cableada encargada de activar las señales JUMP y ALU-TMPS para una determinada instrucción.



- ¿Cuál es el valor de la señal indicada como Inst?
call inm_8

Se dispone de una unidad de control microprogramada para la CPU teórica que se encarga de generar las palabras de control. Las palabras de control se interpretan como se indica en la figura (se muestran sólo los 14 bits menos significativos de la palabra de control).



Para el valor del Registro de Instrucción que se muestra, la unidad de control genera las siguientes palabras de control:

```
...10 1000 0110 1110
...00 0100 0001 0000
...00 0001 1000 0000
...00 0010 0100 0000
...01 0000 0000 1010
...00 0100 0001 0001
```

- Teniendo en cuenta que la señal de control FIN se encuentra en el bit cero de la palabra de control, ¿qué señales de control corresponderán a los bits 4 y 12 de la palabra de control?

bit 4= TMPS-IB bit 12= JUMP

- Si sabemos que la instrucción anterior estaba almacenada en la posición de memoria 3A59. ¿Cuál será el valor del registro contador de programa al terminar la ejecución de dicha instrucción?

3A63

- Se quiere construir un chip de memoria que tendrá 10 líneas de dirección y 8 de datos. ¿Cuántas celdas unitarias deberíamos utilizar? Contestar en decimal.

8192

Apellidos _____

Nombre _____

DNI _____



Examen de Fundamentos de los computadores. Área de Arquitectura y Tecnología de Computadores

1er parcial. O: 13-03-2004

Se ha escrito un programa en ensamblador que maneja listas de NIFs (Números de Identificación Fiscal). El NIF es igual al DNI más una letra que se obtiene a partir de los números del DNI y sirve para evitar errores.

El programa utiliza unas listas de NIFs con la siguiente estructura:

- El número de NIFs de la lista.
- La lista de NIFs propiamente dicha, representando cada NIF como una cadena de 9 caracteres.

Uno de los procedimientos del programa (no mostrado) se llama **comprobar_NIF**. Este procedimiento recibe la dirección de un NIF y comprueba que la letra se corresponde con el número. Se sabe que dentro de este procedimiento se introducen 5 elementos en la pila y se retiran de la misma antes de salir.

Otro de los procedimientos del programa es **comprobar_lista**. Este procedimiento recibe la dirección de una lista (que incluye como primer elemento el número de NIFs) y la comprueba haciendo uso del procedimiento **comprobar_NIF**.

Otro procedimiento (no mostrado) que usa el programa es **multiplica**. Recibe dos números por la pila, los multiplica y devuelve el resultado en R0. Se sabe que dentro de este procedimiento se introducen 3 elementos en la pila y se retiran de la misma antes de salir.

Tenemos los siguientes fragmentos del programa:

```
ORIGEN 0A050h
INICIO ini
.PILA ???? ; Eliminado intencionadamente
.DATOS
lista1 VALOR 3, "01923842I", "74992840J"
        VALOR "32000012R"
lista2 VALOR 2, "15887623A", "98301934P"
; ...Más datos no mostrados
```

```
.CODIGO
; ...Instrucciones no mostradas

; Programa principal: Comprobar NIFs
ini:
    MOVL R5, 9
    MOVH R5, 0 ; R5 = longitud de un NIF
    MOVL R3, BYTEBAJO DIRECCION lista1
    MOVH R3, BYTEALTO DIRECCION lista1
```

```
MOVL R1, 12
MOVH R1, 0 ; Comprobar 12 listas
bucle1:
    PUSH R3 ; Meter la dir. de la lista
    CALL comprobar_lista
    INC R7 ; Destruir parámetro

; Calcular la longitud de la lista
MOV R2, [R3] ;R2= N° elementos de
lista
    PUSH R2
    PUSH R5 ; R5 = 9 (long. de un NIF)
    CALL multiplica
    INC R7 ; Destruir parámetros
    INC R7

; Poner R3 al principio de la
; siguiente lista
    INC R3
    ADD R3, R3, R0

    DEC R1
    BRNZ bucle1
; ...Más código no mostrado
```

```
PROCEDIMIENTO comprobar_lista
    PUSH R6
    MOV R6, R7
    PUSH R0
    PUSH R1
    PUSH R5

; Copiar el primer parámetro en R0
--1--

    MOV R1, [R0] ; R1 = n° elementos lista
    INC R0 ; R0 apunta al 1er NIF en lista

    MOVL R5, 9
    MOVH R5, 0 ; R5 = longitud de un NIF

bucle2:
    PUSH R0 ; Meter la dirección del NIF
    CALL comprobar_NIF
    INC R7 ; Destruir el parámetro
    ADD R0, R0, R5 ; Pasar al siguiente
NIF
    DEC R1
    BRNZ bucle2
```

```
POP R5
POP R1
POP R0
--2--
```

```
FINP
; ...Más código no mostrado
```

- ¿Qué falta en --1--?

INC R6

INC R6

MOV R0, [R6]

- ¿Qué falta en --2--?

POP R6

RET

- Sabiendo que el código ASCII del '0' es 30h, ¿qué valor habrá en la posición de memoria A053h? Contestar en **hexadecimal**.

0039h

- A la vista del código mostrado y los datos de la introducción, ¿qué tamaño mínimo debería tener la pila? Contestar en **decimal**.

13



- Sabiendo que TMPS vale A128h durante el segundo paso de la primera ejecución de la instrucción **BRNZ bucle2**, ¿en qué dirección está almacenada la instrucción **CALL comprobar_NIF**? Contestar en **hexadecimal**.

A123h

- Si en ejecutar desde la instrucción marcada por la etiqueta **ini** hasta la instrucción siguiente a **BRNZ bucle1** se tardó 100 ms utilizando una frecuencia de procesador de 150 MHz, ¿cuánto se tardará utilizando una frecuencia de 300 MHz?

50 ms

- ¿Cuál es el valor del registro de estado después de ejecutarse la instrucción **BRNZ bucle1** la primera vez?

Z: 0 C: 1 O: 0 S: 0

- Dentro del código del programa principal mostrado, ¿cuántas instrucciones acceden a memoria en sus pasos 4 o siguientes?

6



- ¿Cuáles de las siguientes afirmaciones son **FALSAS**? (puedes responder "ninguna" si así lo consideras)

- a) El Registro de Instrucción contiene el código de la instrucción que se está ejecutando pero **NO** durante todos los pasos de ejecución de la misma.
- b) El número de líneas del bus de control del sistema depende directamente del número de bits del Registro de Instrucción y del número de señales de control que puede generar la U.C.
- c) En los programas que usan la pila, las únicas instrucciones que pueden utilizar R7 son PUSH, POP, CALL y RET.
- d) En la CPU de ejemplo, se pueden crear nuevas instrucciones sin necesidad de aumentar el juego de registros ni el número de señales de control.

B, C