

- Duración del examen: 1 hora 45 minutos.
- La solución a cada ejercicio debe escribirse en el espacio reservado para ello en el propio enunciado.
- No podéis utilizar calculadora, móvil, apuntes, ...
- La solución al examen se publicará mañana en Atenea y las notas se publicarán en una semana

Ejercicio 1 (1,5 puntos)

Cada uno de los apartados pregunta sobre un ciclo concreto de la ejecución de una instrucción en el SISC Von Neumann. Escribid el valor de los bits de la palabra de control que genera el bloque SISC CONTROL UNIT durante el ciclo a que hace referencia cada apartado. Poned x siempre que no se pueda saber el valor de un bit (ya que no sabemos cómo se han implementado las x en la ROM_OUT). Para cada apartado/fila se indica el nodo/estado de la UC en ese ciclo y la instrucción (en ensamblador) que está almacenada en el IR en ese ciclo.

| Apartado | Nodo/Estado (Mnemo Salida) | Instrucción en IR (en ensamblador) | Palabra de Control | | | | | | | | | | | | | | | | N (hexa) | ADDR-IO (hexa) |
|----------|-------------------------------|---------------------------------------|--------------------|----|---------------|----|---|---------|----|-----|--------|-------|--------|------|------|------|--------|-------|-------------|-------------------|
| | | | @A | @B | Pc/Rx Ry/N | OP | F | P/I/L/A | @D | WrD | Wr-Out | Rd-In | Wr-Mem | LdIr | LdPc | Byte | Alu/R@ | R@/Pc | | |
| a | D | JALR R5, R3 | | | | | | | | | | | | | | | | | | |
| b | Movhi | MOVHI R2, 0x85 | | | | | | | | | | | | | | | | | | |
| c | Addr | LDB R1, 3 (R1) | | | | | | | | | | | | | | | | | | |

Ejercicio 2 (2 puntos)

Indicad qué cambios se producen en el estado en el SISC Von Neumann después de ejecutar cada una de las instrucciones de la tabla suponiendo que antes de ejecutarse cada una de ellas PC=0x1234, Ri=0x7788 y que el contenido de todas las direcciones pares de la memoria es 0x85 y el de las impares es 0xAA. Utilizad el mnemotécnico $MEM_b[...]$ = ... y/o $MEM_w[...]$ = ... para indicar cambios en la memoria.

| Instrucción a ejecutar | Cambios en el estado del computador |
|------------------------|-------------------------------------|
| LD R1, 10 (R6) | |
| MOVI R1, 0xFA | |
| BNZ R3, 8 | |
| STB 3 (R5), R3 | |

Ejercicio 3 (1 punto)

Completad las filas y columnas de la siguiente tabla que representa un subconjunto de la ROM_OUT de la unidad de control del SISC Von Neumann. Poned x siempre que un bit pueda valer tanto 0 como 1.

| @ROM | Bz | MxF | WrD | R@/PC | Pc/Rx | LdIR | Ry/N | Estado |
|------|----|-----|-----|-------|-------|------|------|--------|
| 3 | | | | | | | | Cmp |
| 5 | | | | | | | | Addr |
| 9 | | | | | | | | Stb |
| 12 | | | | | | | | Bnz |

Ejercicio 4 (3 puntos)

El programa ensamblador de la derecha se ha traducido a lenguaje máquina para ser ejecutado en el SISC Von Neumann, situando la sección `.text` a partir de la dirección `0x1000` de memoria y justo a continuación la sección `.data`.

a) Una vez cargado el programa en memoria:

- ¿A qué dirección de memoria corresponde la etiqueta, o dirección simbólica, `V1`? (0,25 puntos)

`V1 = 0x`

- ¿Cuál es el word almacenado en las siguientes direcciones de memoria? (0,5 puntos)

`Memw[0x1002] = 0x`

`Memw[0x100E] = 0x`

b) Una vez ejecutado el programa en el computador SISC Von Neumann ¿Cuál es la dirección de memoria escrita por la instrucción `ST`? ¿Cuál es el valor escrito? (1 punto)

`Memw[0x] = 0x`

```

        FLAG = -1
.data
MAGIC: .word 0x89FF
V1:     .word 1, 2, 4, 8, 16
        .word 32, 64, 128, 256
        .word FLAG
V2:     .word 0

.text
        MOVI    R0, lo(V1)
        MOVHI   R0, hi(V1)
        MOVI    R1, lo(FLAG)
        MOVHI   R1, hi(FLAG)
        MOVI    R2, 0
L1:     LD      R3, 0(R0)
        CMPEQ   R4, R3, R1
        BNZ     R4, L2
        ADD     R2, R2, R3
        ADDI    R0, R0, 2
        BZ      R4, L1
L2:     MOVI    R7, lo(V2)
        MOVHI   R7, hi(V2)
        ST      0(R7), R2

.end

```

c) Indicad el número total de instrucciones ejecutadas por el código de la sección `.text` y desglosadlo en instrucciones lentas y rápidas. Calculad el tiempo de ejecución para el computador Harvard unicycle y el Von Neumann suponiendo que sus tiempos de ciclo son 2.500 y 1.000 u.t. respectivamente. Finalmente, ¿cuánto vale x para que sea cierta la afirmación: *el computador Harvard unicycle tarda un $x\%$ menos que el Von Neumann en ejecutar dicho código.* ? (1 punto). **Indicad tanto las operaciones como el resultado.**

$N_{total} =$

$N_{rapidas} =$

$N_{lentas} =$

$T_{ejec}(Harv.Uni.) =$

$T_{ejec}(VonNeu.) =$

$x =$

d) Una vez ejecutada la instrucción `ST`, ¿qué hará el computador? (0,25 puntos)

Ejercicio 5 (2,5 puntos)

Completad el diseño del SISC Von Neumann para que ejecute, **además** de las instrucciones originales, 8 nuevas instrucciones LDESCk (k es un entero entre +3 y -4, es decir, LDESC+3, LDESC+2, ..., LDESC-3 y LDESC-4):

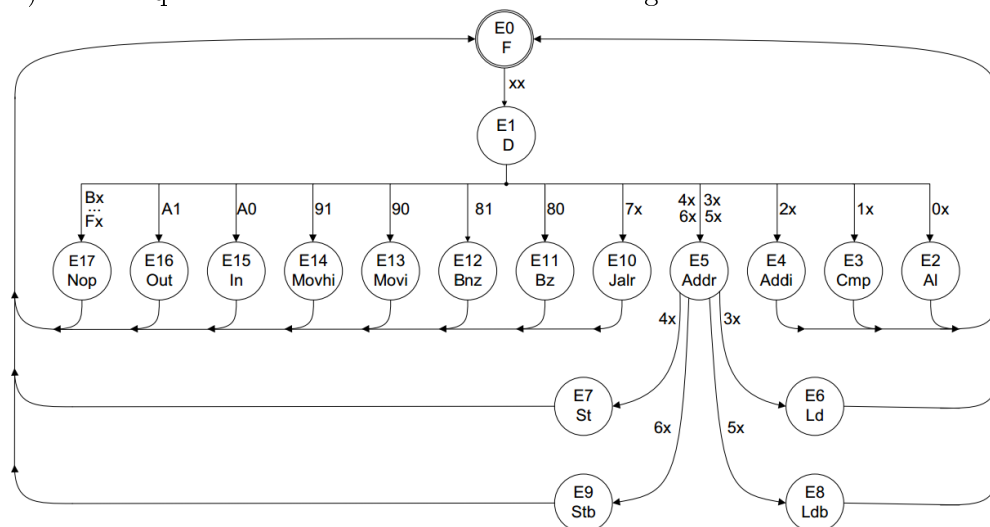
- Codificación: 1111 aaa bbb ddd kkk
- Sintaxis: LDESCk Rd, Ra, Rb
- Semántica: $Rd = Mem_W[Ra + Rb \times 2^k]$; donde k es la interpretación de los bits 2, 1 y 0 de la instrucción como un entero codificado en complemento a 2.

Para poder ejecutar las nuevas instrucciones, os proponemos que modifiquéis únicamente la ALU y el contenido tanto de la ROM_OUT como el de la ROM_Q+.

a) Indicad cómo modificaríais la tabla de funcionalidades de la ALU. (0,5 puntos)

| F | | | OP | | | |
|-------|-------|-------|----|-----------------|---------------|------------|
| b_2 | b_1 | b_0 | 11 | 10 | 01 | 00 |
| 0 | 0 | 0 | | X | CMPLT (X, Y) | AND (X, Y) |
| 0 | 0 | 1 | | Y | CMPLE (X, Y) | OR (X, Y) |
| 0 | 1 | 0 | | MOVHI (X, Y) | | XOR (X, Y) |
| 0 | 1 | 1 | | $X \& (\sim 1)$ | CMPEQ (X, Y) | NOT (X) |
| 1 | 0 | 0 | | | CMPLTU (X, Y) | ADD (X, Y) |
| 1 | 0 | 1 | | | CMPLEU (X, Y) | SUB (X, Y) |
| 1 | 1 | 0 | | | | SHA (X, Y) |
| 1 | 1 | 1 | | | | SHL (X, Y) |

b) Indicad qué modificaciones introduciríais en el grafo de estados de la unidad de control. (0,5 puntos)



c) Indicad la(s) dirección(es) de la ROM_Q+ (en binario) y su contenido (en hexadecimal) para implementar la transición que hayáis añadido desde el estado E1 (D). (0,5 puntos)

| |
|--|
| |
|--|

d) Indicad el contenido de las filas de la ROM_OUT que sea preciso modificar así como las acciones a realizar (la tabla adjunta tiene el número suficiente de filas, incluso es posible que no sean necesarias todas) (1 punto)

| @ROM | Bnz | Bz | WrMem | RdIn | WrOut | WrD | Ldlr | Byte | R@/Pc | Alu/R@ | Pc/Rx | Ry/N | P//L/A1 | P//L/A0 | OP1 | OP0 | MxN1 | MxN0 | MxF | F2 | F1 | F0 | Mx@D1 | Mx@D0 |
|------|-----|----|-------|------|-------|-----|------|------|-------|--------|-------|------|---------|---------|-----|-----|------|------|-----|----|----|----|-------|-------|
| | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | |

Acciones asociadas al estado
(en lenguaje de transferencia
de registros)

| |
|--|
| |
| |
| |
| |