

**Nombre:** \_\_\_\_\_ **NIU:** \_\_\_\_\_

**PROBLEMA 1 (4 puntos)** Tenemos un sistema como el utilizado en problemas (procesador RISC V, bus de datos y direcciones de 32 bits y una memoria RAM de 8Mbytes situada a partir de la dirección 0x4000) que contiene un controlador de E/S de un puerto USB. Las direcciones de sus registros de E/S son:

**0x00000600: Registro de Control**

**0x00000601: Registro de Estado**

**0x00000602-0x00000605: Registro de Datos (4 bytes)**

Se ha conectado un pendrive a un puerto USB, al cual se quiere realizar una escritura de **16 Kbytes (0x4000 bytes), de 4 bytes en 4 bytes**, desde un array situado la **memoria principal** cuya dirección de origen es **0x8000**,

**IMPORTANTE:** En los siguientes apartados donde haya que escribir código, se ha de utilizar el **lenguaje ensamblador visto en problemas**.

**a) (2 puntos) Escribir el código en ensamblador que realice la transferencia de 16 Kbytes mediante el método de E/S programada con espera de respuesta.** Tener en cuenta que:

- **El bit 7 del registro de Estado** se activa a 1 cuando el controlador está preparado para realizar una transferencia. Para desactivarlo (a **0**), se ha de escribir un **1** en dicho bit.

Suponer que ya está inicializado el controlador de E/S para realizar esta transferencia.

**Comentar todas las instrucciones**

**; R31 contiene dirección base del array y R27 contiene la dirección del último elemento del array**

LI R31, 0x00008000

LI R27, 0x0000C000 ó ADDI R27,R31,0x4000

**;R30, contiene la dirección inicio de los registros E/S del controlador del puerto USB**

LI R30, 0x00000600

bucle:

**; Esperar que el controlador USB esté preparado para enviar/recibir un dato. Utilizar el**

**; registro R29 para almacenar los datos intermedios de esta operación**

LB R29, 1(R30)

ANDI R29,R29,0x00000080

BEQ R29,X0, bucle

**; Enviar el dato. Utilizar el registro R28 para almacenar los datos intermedios**

LW R28,0(R31)

SW R28,2(R30)

**; Preparar el sistema para enviar/recibir el siguiente dato. Utilizar el registro R28 para almacenar los datos intermedios**

LI R28, 0x00000080

SB R28,1(R30)

ADDI R31, R31,0x4

**; Comprobar si se han transmitido todos los datos**

BGE R31,R27, final  
J bucle

**b) (2 puntos)** Escribir el **código de inicialización del controlador de E/S del puerto USB** y de una **rutina de servicio de interrupción (RSI)** para realizar la transferencia de **16 Kbytes (de 4 bytes en 4 bytes)**, mediante el método de **E/S dirigido por interrupción**. Tener en cuenta que:

- Para que el controlador active la **generación de peticiones local de interrupción**, hay que escribir un **1** en el **bit 3 del registro de control**. Si el **bit 1 del registro de control** tiene un valor de **0** implica una **operación de lectura** (disp. → registro datos) y si vale **1**, una **escritura** (registro de datos → disp.)
- La **desactivación del bit 7 del registro de estado**, provocará que se deje de generar la petición de interrupción (en caso que estas estén activas).

La variable global **dir\_array** contiene la dirección del índice (o elemento) del array donde se va a escribir el dato procedente del controlador de E/S.

;; \*\*\*\*\* Programa principal (main de C)

**;IMPORTANTE: Utilizar el registro R28 para almacenar los datos intermedios**

main:

**; R31, contiene la dirección inicio de los registros E/S del controlador del puerto USB**  
**; R30 contiene dirección base del array**  
**; R29 contiene la dirección de variable dir\_array**

LI R31, 0x00000600  
LI R30, 0x00008000  
LLA R29, dir\_array

**; Guardar la dirección base del array en dir\_array**  
SW R30, 0(R29)

**; Inicializar el controlador, utilizar el registro R30 para los datos intermedios**  
LB R30, 0(R31)  
ORI R30, R30, 0x0000000A  
SB R30 0(R31)

infinito:

WAI ; Wait An Interrupt.

**; u otras instrucciones de otras tareas**  
J infinito

rsi\_PORT\_USB:

**; Inicializar los registros del procesador para gestionar la transacción.**  
**; R31, contiene dirección base dir\_array, R30 su contenido y R29 contiene la dirección inicio**  
**; de los registros E/S del controlador del puerto USB**  
LLA R31, dir\_array  
LW R30, 0(R31)

LI R29, 0x00000600

**; Comprobar que no se hayan transmitido todos los datos**

LI R28, 0x0000C000

BGE R30,R28, final\_rsi

**; Realizar la transferencia.**

LW R28, 0(R30)

SW R28,2(R29)

**; Desactivar el bit que indica que el controlador USB esté preparado para enviar/recibir un  
;dato**

LI R28, 0x000000080

SB R28,1(R29)

**; Actualizar la Dirección siguiente elemento del array**

ADDI R30, R30,0x4

SW R30, 0(R31)

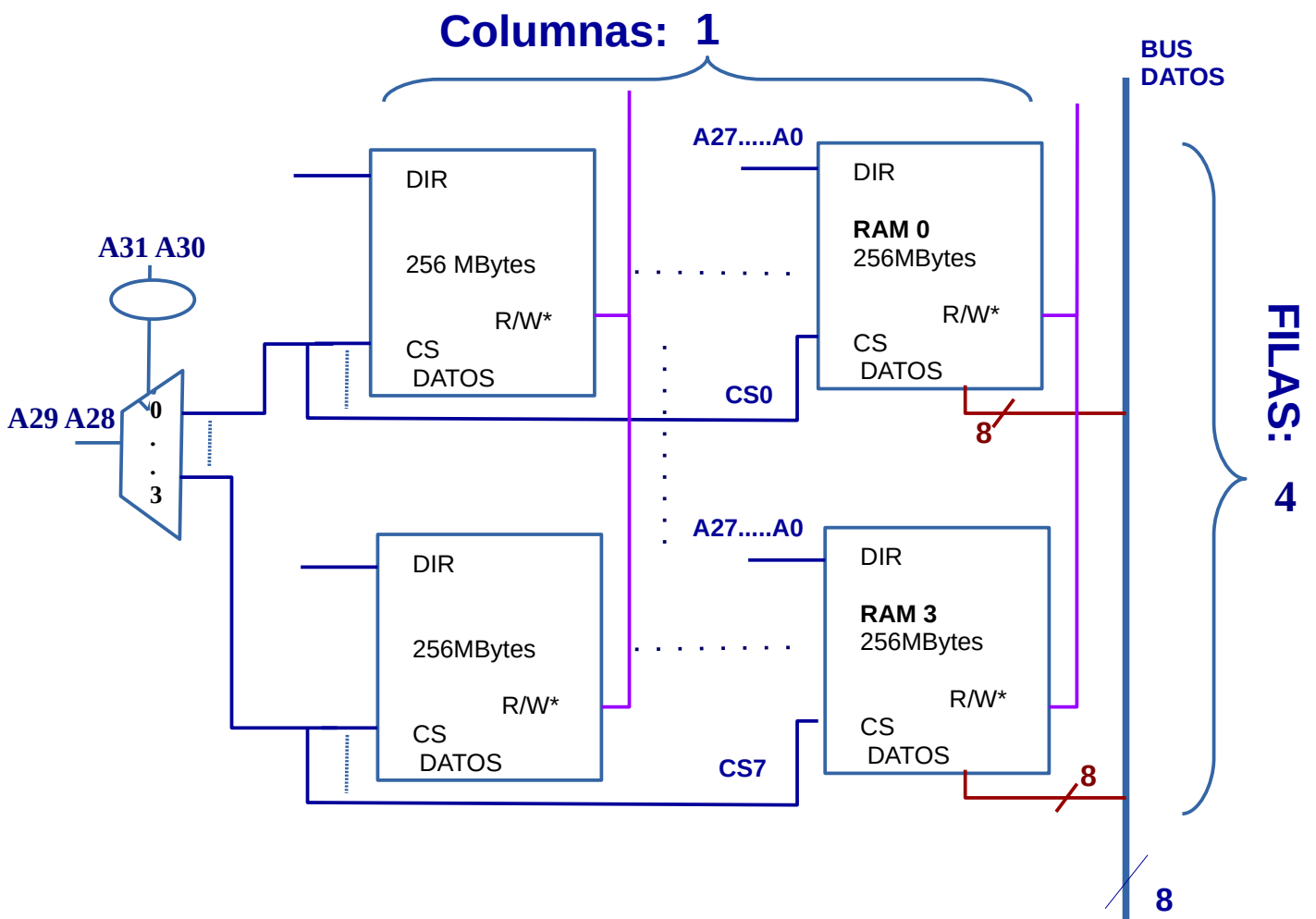
final\_rsi:

**MRET**

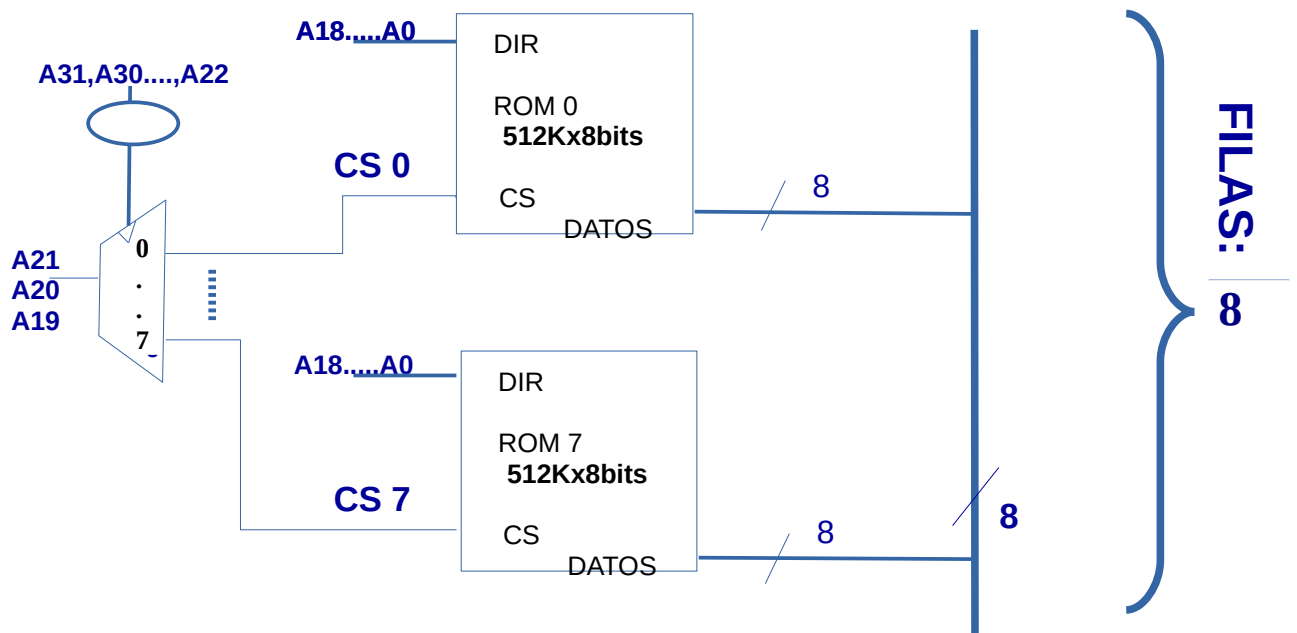
**PROBLEMA 2 (2 puntos)**

A un sistema (con un bus de **direcciones de 32 bits** y uno de **datos de 8 bits**) se le quieren añadir una **memoria RAM estática de 1 Gbytes** y una **memoria ROM de 4 Mbytes**. La **memoria RAM** está implementada con **chips de memoria RAM estática de 256 Mbytes** y está situada a partir de la dirección **0x40000000**. La **memoria ROM** está implementada con **chips de memoria ROM de 512 Kbytes** y situada a partir de la dirección **0x00800000**.

- a **(1,5 puntos)** Completar esquema de conexión de la CPU con los chips **de memoria RAM estática de 256 Mbytes** y con los **de memoria ROM de 512 Kbytes**. En los recuadros de los esquemas se deben indicar que líneas de los buses de direcciones y datos reciben los diferentes componentes de dichos esquemas, el tamaño del decodificador, e identificar numéricamente cada chip y cada señal CS, por ejemplo, RAM 0, CS0, etc.



## Columnas: 1



b) (0,5 puntos) Especificar las funciones lógicas de los CS0 de los chips de la memoria RAM y ROM

**RAM**

$$CS0 = A31 * A30 * A29 * A28 *$$

**ROM**

$$CS0 = A31 * A24 * A23 * A22 * A21 * A20 * A19 *$$

**AYUDA:** Sintaxis instrucciones del lenguaje ensamblador utilizado en problemas

- **LLA** Rd, símbolo, (LB/SB LH/SH | LW/SW) Rd, inm (Rm)
- **LI** Rd, imm
- **MV** Rd, Rm
- **ADD / SUB / MUL / DIV / SRA / SLL / SRL / AND / OR / XOR** Rd, Rm, Rs
- **ADDI / SUBI / MULI / DIVI / SRAI / SLLI / SRLI / ANDI / ORI / XORI** Rd, Rm, inm
- **Enteros con signo:** B{cond} etiqueta . **Enteros sin signo:** B{cond}U etiqueta, donde cond puede ser: EQ: Igual, NE: No igual, GE: mayor o igual, LT: menor, GT: mayor, LE: mas pequeño o igual.
- **J** etiqueta
- **JAL** Rd, etiqueta, RET
- **MRET**