

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:

0x00000500: Registro de Control 0x00000501: Registro de Estado

0x00000502-0x00000505: 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 respueta**. Tener en cuenta que:
  - **El bit 3** 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

; R1 contiene dirección base del array y R6 contiene la dirección del último elemento del array

LI R1, 0x00008000 LI R6, 0x0000C000 ó ADDI R6,R1,0x4000

;R2, contiene la dirección inicio de los registros E/S del controlador del puerto USB

LI R2, 0x00000500

bucle:

- ; Esperar que el controlador USB esté preparado para enviar/recibir un dato. Utilizar el
- ; registro R3 para almacenar los datos intermedios de esta operación

LB R3, 1(R2) ANDI R3,R3,0x00000008 BEO R3 ,X0, bucle

; Enviar el dato. Utilizar el registro R4 para almacenar los datos intermedios

LW R4,0(R1) SW R4,2(R2)

; Preparar el sistema para enviar/recibir el siguiente dato. Utilizar el registro R4 para almacenar los datos intermedios

LI R4, 0x000000008 SB R4,1(R2)



ADDI R1, R1,0x4

; Comprobar si se han transmitido todos los datos

BGE R1,R6, final J bucle

#### final:

- 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 7 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 3 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 R4 para almacenar los datos intermedios

#### main:

- ; R1, contiene la dirección inicio de los registros E/S del controlador del puerto USB
- ; R2 contiene dirección base del array
- ; R3 contiene la dirección de variable dir\_array

LI R1, 0x00000500 LI R2, 0x00008000 LLA R3, dir\_array

; Guardar la dirección base del array en dir\_array

SW R2, 0(R3)

; Inicializar el controlador, utilizar el registro R2 para los datos intermedios

LB R2, 0(R1) ORI R2, R2, 0x00000082 SB R2 0(R1)



```
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.
- ; R1, contiene dirección base dir\_array, R2 su contenido y R3 contiene la dirección inicio de
- ; los registros E/S del controlador del puerto USB

LLA R1, dir\_array LW R2, 0(R1) LI R3, 0x00000500

; Comprobar que no se hayan transmitido todos los datos

LI R4, 0x0000C000 BGE R2,R4, final\_rsi

; Realizar la transferencia.

LW R4, 0(R2) SW R4,2(R3)

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

LI R4, 0x000000008 SB R4,1(R3)

; Actualizar la Dirección siguiente elemento del array

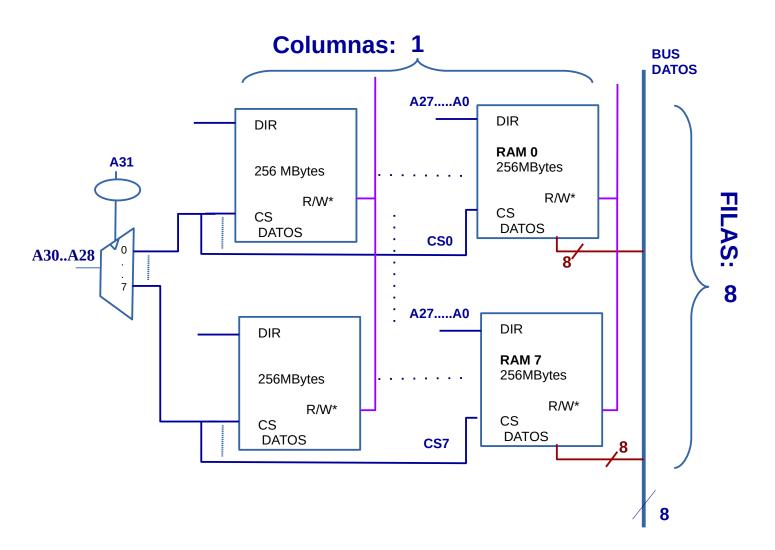
ADDI R2, R2,0x4 SW R2, 0(R1) final\_rsi: MRET



#### PROBLEMA 2 (2 puntos)

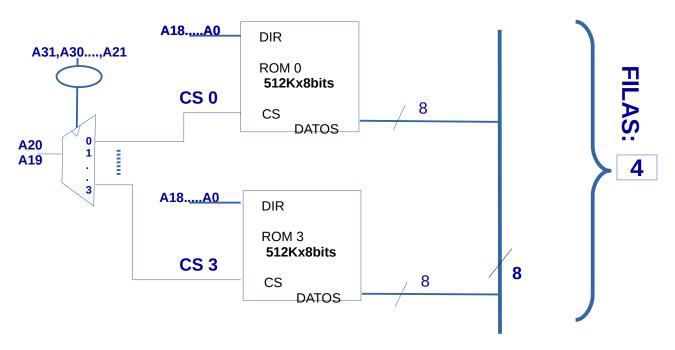
A un sistema (con un bus de **direcciones de 32 bits** y uno de **datos de 8 bits**) se le quieren añadir una **memoria RAM estática de 2 Gbytes** y una **memoria ROM de 2 Mbytes. La memoria RAM** está implementada con **chips de memoria RAM estática de 256 Mbytes** y está situada a partir de la dirección **0x80000000.** La **memoria ROM** está implementada con **chips de memoria ROM de 512 Kbytes** y situada a partir de la dirección **0x00400000.** 

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 CSO 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