



**Universitat
Autònoma
de Barcelona**

ARQUITECTURA DE COMPUTADORS I PERIFÈRICS

PROBLEMES

Entrada/Sortida

Es té un sistema computador amb un únic bus compartit que gestiona, entre d'altres, una pantalla alfanumèrica (per mostrar resultats) i un disc dur (per emmagatzemar dades). Per controlar aquests dispositius, aquest sistema inclou els següents components:

- Una CPU amb el mateix repertori d'instruccions vist als problemes del tema 1 del processador, que executa 200 MIPS (milions d'instruccions per segon) i amb un sistema d'interrupcions vectoritzades. Els **registres d'aquesta CPU son de 32 bits**, però permet carregar-li's dades de **8 i 16 bits (halfword)** des de la memòria, situant **0** en els **bits més alts (o significatius)** del registre (del 31 al 8, per les 8 bits i del 31 al 16 per les de 16 bits). Per operar amb aquestes dades , s'han afegit les següents instruccions:
 - **byte**: Implica la lectura/escriptura de la dada (1 byte) de 1 adreça de memòria :
 - LB Rd, desplaçament(Rm)**: Llegir una dada (tipus byte) de la memòria.
 $Rd_{(8..0)} \leftarrow [mem(Rm + desplaçament)]$
 - SB Rd, desplaçament(Rm)** Escriure una dada (tipus byte) a memòria.
 $Rd_{(8..0)} \rightarrow [mem(Rm + desplaçament)]$
 - **halfword (16 bits)**: Implica la lectura de les dades (d'un byte cadascuna) de 2 adreces de memòria consecutives.
 - LH Rd, desplaçament(Rm)**: Llegir una dada (tipus halfword) de la memòria.
 $Rd_{(15..8)} \leftarrow [mem(Rm + desplaçament+1)]$ i $Rd_{(7..0)} \leftarrow [mem(Rm + desplaçament)]$
 - SH Rd, desplaçament(Rm)** Escriure una dada (tipus halfword) a memòria.
 $Rd_{(15..8)} \rightarrow [mem(Rm + desplaçament+1)]$ i $Rd_{(7..0)} \rightarrow [mem(Rm + desplaçament)]$

Exemples:

Si **R3** conté **0x0000F100** (es una adreça de memòria)

La instrucció **LB R1, 0(R3)** situa el contingut de l'adreça de memòria (**0x0000F100 + 0**) (1 byte) en els **8 bits de la part baixa de registre R1** (i 0 als 24 bits de la part alta).

La instrucció **SH R2, 4(R3)** situa els 16 bits de la part baixa de R2 al contingut de les adreces de memòria (**0x0000F100+4**) i (**0x0000F100+4+1**). Això es:

R2_(15..8) -> [0x0000F105] i R2_(7..0) -> [0x0000F104]

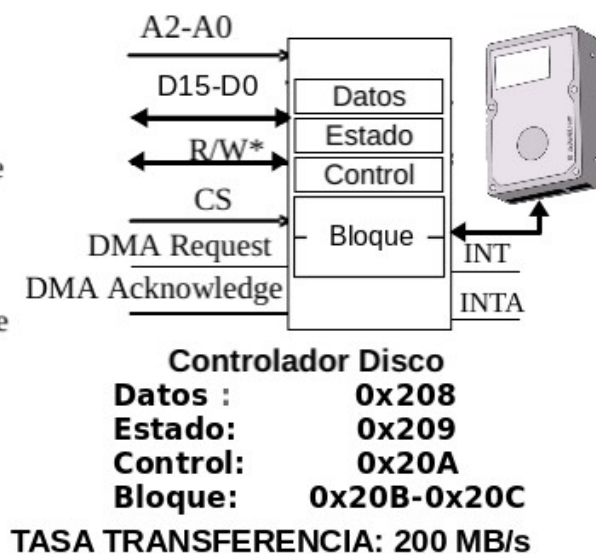
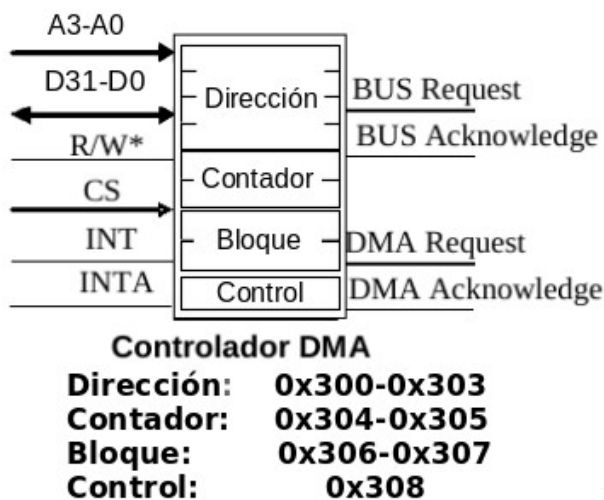
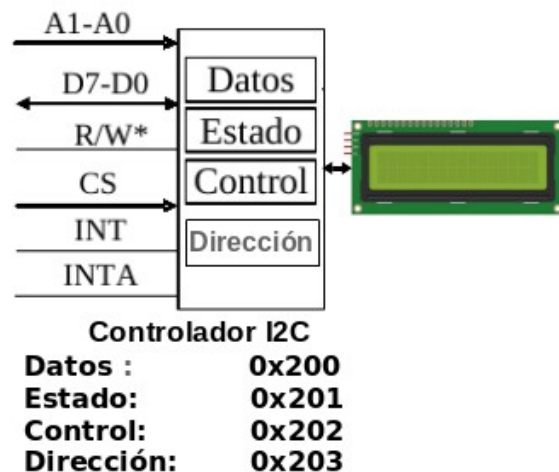
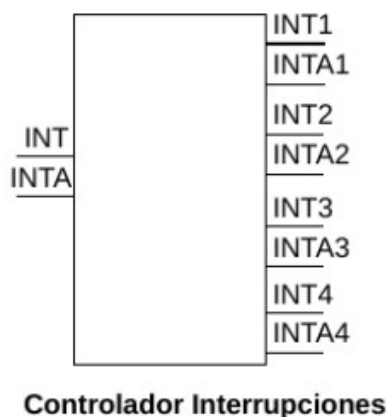
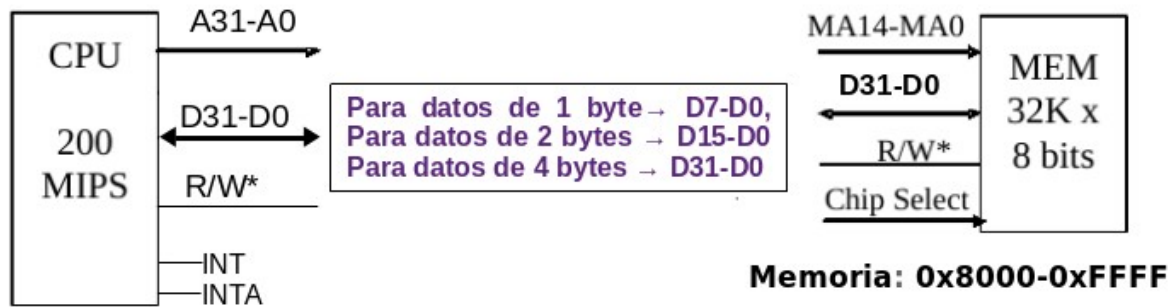
- El **espai d'adreces d' E/S** del sistema **NO està separat del de memòria (es memory mapped)**.
- Un mòdul de memòria de 32K x 8bits
- Un mòdul **controlador I2C** per dispositius connectats a través d'un bus I2C (serie síncron) amb capacitat d'interrupció al qual està connectat la pantalla alfanumèrica. El seu **identificador d'interrupció es el 11**.
- Un mòdul **controlador d'Interrupcions** amb capacitat per a gestionar fins a 4 dispositius i de generar els **vectors d'interrupció de 8 bits** corresponents a cada dispositiu.
- Un mòdul **controlador de disc** amb capacitat d'interrupció i capaç de comunicar-se amb un controlador de DMA. La taxa de transferència del disc és de 200 MB/s. El seu **identificador d'interrupció es el 13**.
- Un mòdul **controlador de DMA**. El seu **identificador d'interrupció es el 15**.
- La gestió de les interrupcions **es apropiativa**, això es, els sistema d'interrupcions **no queda inhibit** mentre s'atén a una interrupció.

La figura de la pàgina següent indica les adreces en les que es pot accedir a cadascun del ports d' E/S dels controladors.

ARQUITECTURA DE COMPUTADORS I PERIFÈRICS

Escola d'Enginyeria
Enginyeria Telecomunicacions

PROBLEMES
ENTRADA/SORTIDA



Registres de la CPU relacionats amb la gestió d'interrupcions:

Apart dels registres vistos de la CPU en el tema 1, aquesta també té els següents **registres de control i estat (CSR)** per la gestió de les interrupcions (i les excepcions):

- **Registre d'estat de la màquina (MSTATUS):** El bit 3 d'aquest registre **habilita i deshabilita globalment les interrupcions**. Si val 1 les interrupcions estan habilitades i la CPU les acceptarà, si val 0, estan deshabilitades i la CPU no acceptarà cap interrupció.
- **Registre MIE:** **Habilita o deshabilita una interrupció determinada (habilitació local d'interrupcions)**. si el **bit i de MIE** està a **1**, la interrupció associada a aquest bit està **habilitada** i la CPU l'acceptarà (sempre que les interrupcions estiguin habilitades globalment). Si **val 0** està **deshabilitada** i la CPU no l'acceptarà. Els bit del MIE associat a la interrupció del **controlador I2C** es el **11**, el del **controlador de disc** es el **13** i el del **DMA** es el **15**.
- **Registre MTVEC:** Conte l'adreça base (a on comença) de la **taula del vector d'interrupcions**. En aquest sistema la taula de vectors d'interrupcions està situada a les primeres posicions de la memòria principal, per tant el valor d'aquest registre serà **0x8000**
- **Registre MCAUSE:** Conte el identificador de la font d'interrupció. Per exemple, si ha interromput el **controlador I2C**, aquest registre tindrà el valor de **11**.

IMPORTANT: Mentre no s'indiqui el contrari, es considerarà que les interrupcions estan habilitades, tant globalment com localment.

Controlador I2C

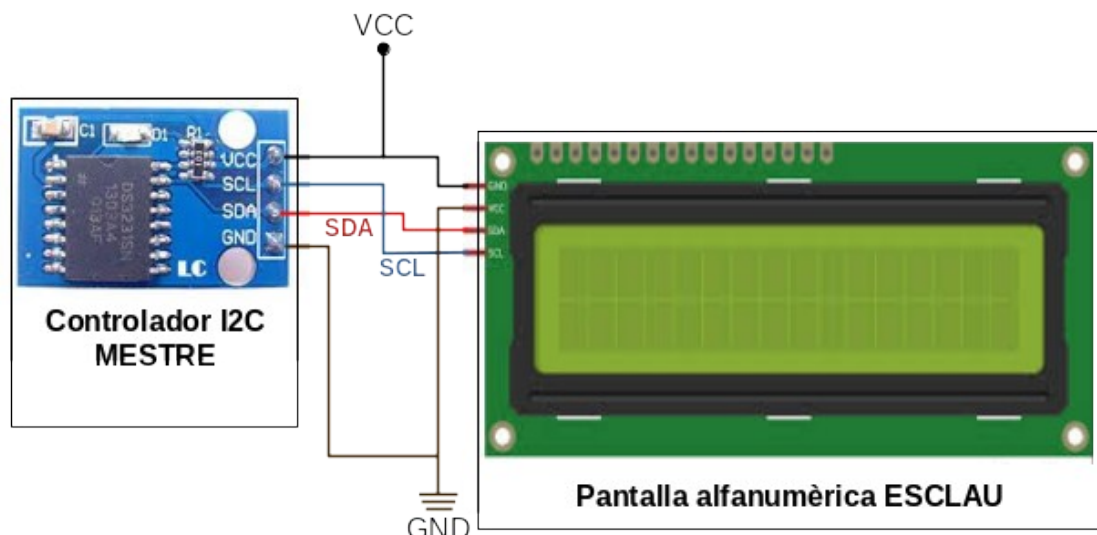
El bus I2C, que utilitzen per intercanviar informació el controlador I2C i la pantalla alfanumèrica, es un bus de comunicacions **serie síncron**.

Les característiques més importants d'aquest bus son:

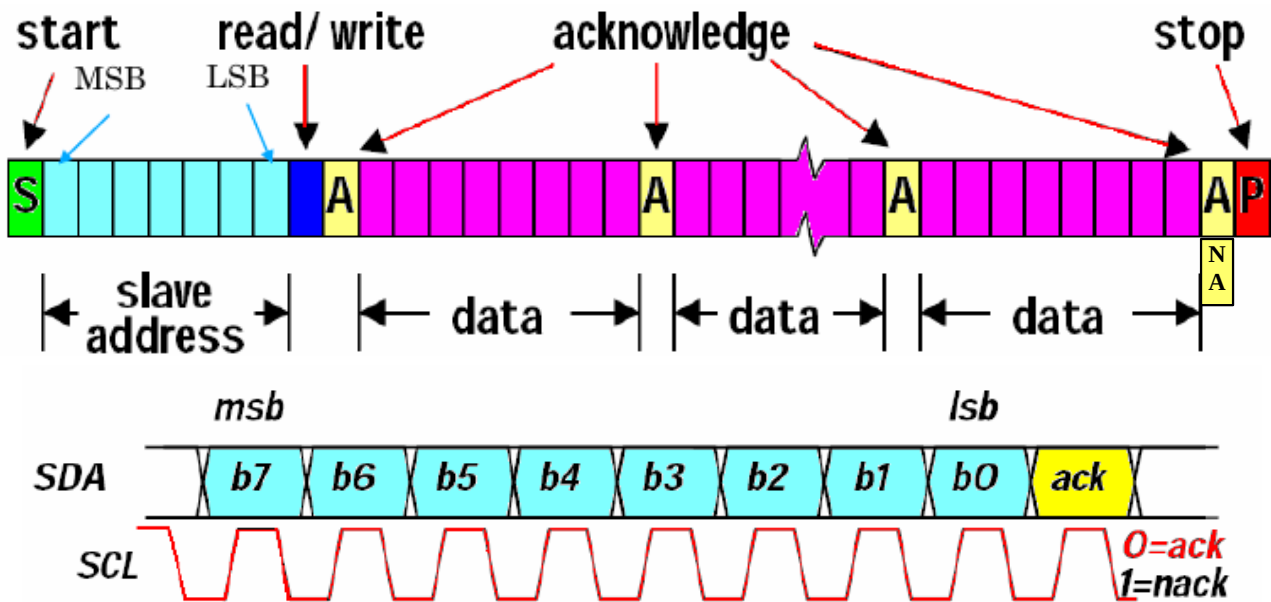
- Només necessita dues línies per transmetre dades, la del rellotge (**SCL**) i la de dades (**SDA**).
- Tots els dispositius connectats a aquest bus poden ser **mestres** o **esclaus**, però sempre es el dispositiu **mestre** (en el nostre cas es el controlador I2C) el que inicia una transmissió.
- Independentment que el dispositiu sigui mestre o esclau, pot enviar o rebre informació (es el mestre qui decideix si vol rebre o enviar informació a l'esclau).
- Cada dispositiu es reconegut amb **adreça única de 7 bits** (el **controlador I2C te la 0x28** i **pantalla te la 0x2C**).
- Les dades es transmeten en blocs de 8 bits i darrere de cada un d'aquest blocs, s'ha de rebre una senyal de reconeixement (handshake)
- Velocitat de transmissió: 3,4 Mbits /s

Esquema de connexió i format d'un missatge

L'esquema de connexió entre el controlador I2C i la pantalla alfanumèrica:



El format dels missatges enviats pel bus de dades (SDA) i la seva sincronització amb la senyal de rellotge (SCL) es les següents:



msb: Bit més significatiu (Most significant bit)
lsb: Bit menys significatiu (low significant bit)

Transmissió d'un missatge

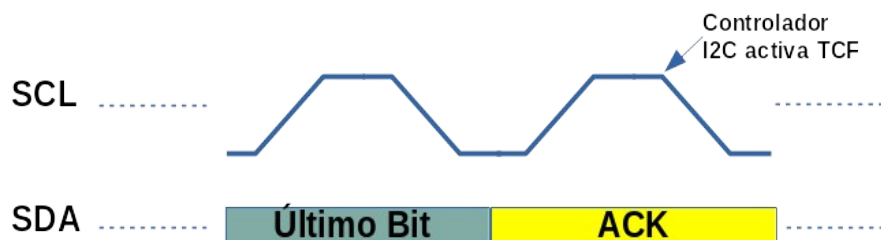
La transmissió d'un missatge pel bus **SDA** segueix els següents principal passos:

- 1) Sempre es el dispositiu **Mestre** que inicia la comunicació i això ho fa posant la senyal de **START** i el **primer byte del missatge**, en el qual situa l'**adreça del dispositiu Esclau (bits 7 al 1)** i al bit 0 si es una operació de **lectura (Esclau -> Mestre)** o **escriptura (Mestre -> Esclau)**.
- 2) Durant el següent cycle del rellotge del **SCL**, el dispositiu **Esclau** envia un **Acknowledge (ACK, té un valor de 0)** per confirmar que ha rebut la petició del **Mestre**.
- 3) A continuació es transmeten **totes les dades de 1 byte (1 bit per cycle del SCL)** fins l'última. Si es una **lectura**, el **Mestre situa les senyals de ACK** després de cada byte per confirmar que ha rebut les dades. Si es un **escriptura** es l'**Esclau que situa les senyal de ACK** després de cada dada per confirmar la recepció de les dades. En cas d'una **lectura**, després de l'**últim byte de dades**, el **Mestre envia**, una senyal de **NACK (te el valor de 1)** enlloc d'un **ACK**. Per informar que el missatge ha finalitzat, el **Mestre envia**, després de l'**últim ACK o NACK**, la senyal de **STOP** al **Esclau**.

Registres d'Entrada/Sortida del controlador del I2C:

- **Dades:** Conte el byte a enviar o rebut.
- **Control:** Controla el funcionament del controlador I2C, els seus bits son (el reste son indiferents):
 - **7 (IICEN):** Activa el el controlador (1: actiu, 0: inactiu)
 - **6 (IICIE):** Activa la generació d'interrupcions (1:actiu, 0:Inactiu)
 - **5 (MST):** Selecciona si actua com a Mestre o Esclau (1:Mestre, 0:Esclau)
 - **4 (TX):** Selecciona si el Mestre envia o rep dades (1:Envia, 0:Reb)
 - **3 (TXAC):** Activa la generació de senyals de reconeixement (1:Actiu, 0:inactiu)
- **Estat:** Conte d'informació sobre el funcionament del controlador, els seus bits son (el reste son indiferents):

- **7 (TCF):** Informa si s'ha completat una transferència (1: Completada, 0: No completada). En cas de lectura indica que ja hi és el byte al registre de dades i si es fa una escriptura, el registre de dades ja està preparat per rebre la dada a enviar. Aquest **bit s'esborra al fer una lectura o escriptura del registre de dades**. Perquè es pugui comprovar si la senyal ACK ha estat col·locada per l'esclau, aquest flag s'activa al pols de baixada del cicle del SCL corresponent al bit d'aquesta senyal ACK (transmesa pel bus SDA).



- **2 (SRW):** Direcció de transmissió indicada pel Mestre a l'Esclau (1:Envia, 0:Reb)
- **0 (RXAC):** Recepció de les senyals de reconeixement (1:Rebut ACK, 0:No Rebut ACK)
- **Adreça:** Conté l'adreça de 7 bits (bits 7 al 1) del dispositiu en mode **Esclau**.

Controlador disc dur

Els dispositius d'emmagatzematge de dades, com els discs durs, DVD's, memòries USB, etc, utilitzen diferents **formats físics i protocols** (suport magnètic, òptic, interfície SATA, M.2, etc) per a emmagatzemar les dades dels diferents fitxers del sistema (executables, documents, llibreries, imatges, etc). Per a aïllar al programador (i al microprocessador) dels diferents formats físics dels dispositius d'emmagatzematge, aquests són definits com uns **dispositius d'emmagatzematge lògic**, els quals estan formats per una **llista de N blocs** (bloc 0, bloc 1,..., bloc N-1, per aquest controlador N és 2^{16}) **d'una mida de M bytes cadascun**, pels exercicis d'aquest tema, **M és 1 Kbyte**. Per tant, pel programador cada fitxer estarà emmagatzemat en un determinat nombre d'aquests **blocs lògics** (i no necessitarà preocupar-se de com s'emmagatzemen físicament als diferents dispositius reals). Per exemple, un fitxer de **2000 bytes** ocuparà **2 blocs lògics** ($1 \text{ Kbyte} < 2000 \text{ bytes} < 2 \text{ Kbytes}$), que podrien ser el **200** i el **201** (si estan els dos lliures).

El Controlador, per a dur a terme les peticions dels programes dels usuaris (o en sistemes reals, del Sistema Operatiu), s'encarrega de comunicar-se, tant a **nivell elèctric** com de **protocol**, amb el dispositiu físic i informar-li a on es **situen físicament** els bytes dels **blocs lògics** sol·licitats. Suposem que el **controlador de disc** té connectat un **disc dur del tipus electromecànic** (suport magnètic). Aquest disc, físicament, està dividit en un conjunt de **C cilindres (numerats 0 al C-1)** de **P plats** cadascun (**numerats de la 0 a la P-1**) i amb **S sectors** per plats (**numerats 0 al S-1**). Cada **sector** té una mida de **512 bytes** i és a on s'emmagatzemen (magnèticament) les dades (mida del disc: $(C \cdot P \cdot S) \cdot 512 \text{ bytes}$). Per tant, el controlador al conèixer com s'emmagatzemen físicament les dades al disc, li pot informar (a través de missatges de control) en quins **cilindres, plats i sectors** estan situades les dades dels **blocs lògics** sol·licitats. Si, a més a més, el sistema implementa una interfície tipus SATA (Seriata), llavors el connector entre el controlador i el disc dur és un bus sèrie síncron d'alta velocitat amb 4 línies de dades (2 anada i 2 de tornada), compartint els missatges de dades i control aquestes línies de dades.

Per exemple, si el disc dur té una capacitat de **1 Mbyte**, el programador el veu com un **dispositiu lògic** de **1024 blocs** d'un **Kbyte** cadascun ($1024 \text{ blocs} \cdot 1 \text{ kbyte} = 1 \text{ Mbyte}$). Però físicament, aquest disc dur està format per **8 cilindres** amb **8 plats** per cilindre i amb **32 sectors** per plat ($(8 \text{ cilindres} \cdot 8 \text{ plats} \cdot 32 \text{ sectors}) \cdot 512 \text{ bytes} = 1 \text{ Mbyte}$). Per tant, el **controlador del disc** al

conèixer aquesta informació, pot saber **quins sectors del disc dur** ocupen cada **bloc lògic**, en aquest cas:

Blocs lògics	Cilindre	plats	Sectors
0 al 15	0	0	0,1 → bloc 0 ;...; 30,31 → bloc 15
16 al 31	0	1	0,1 → bloc 16 ;...; 30,31 → bloc 31
.....			
128 al 143	1	0	0,1 → bloc 128 ;...; 30,31 → bloc 143
.....			
704 al 719	5	4	0,1 → bloc 704 ;...; 30,31 → bloc 719
.....			
1008 al 1023	7	7	0,1 → bloc 1008 ;...; 30,31 → bloc 1023

Descripció dels principals bits dels Registres d'Entrada/Sortida del controlador del disc dur:

El **bit 2** del **registre d'estat** del controlador de disc dur es posa a **1** quan **hi ha disponible un byte al registre de dades** (lectura de dades del disc dur) o quan aquest registre **està preparat per rebre un byte** (escriptura de dades al disc dur). Per **esborrar aquest bit**, **hi ha que escriure un 1 en el bit 2 del registre d'estat**. Per **activar localment la interrupció** del disc dur (per que el controlador la generi) hi ha que **escriure un 1 al bit 7 del registre de control**. Els **bits 6 i 5 del registre de control** indiquen les següents accions: **00** → No hi ha **cap operació per realitzar-se**, **01** → Petició per fer una **escriptura al disc**, **10** → petició per fer una **lectura del disc**.

PROBLEMA 1.

Els principals passos que es fan amb una escriptura de dades (controlador I2C -> pantalla) son:

- 1) Configurar el controlador I2C (com a Mestre): Definir-lo com a dispositiu Mestre, (bit **MST** del reg. de control), si envia o rep dades (bit **TX** del reg. de control), si s'activen les senyals **ACK** (bit **TXAC** del reg. de control) i l'activació de la generació d'interrupcions (bit **IICIE** del reg. de control)
- 2) Activar el controlador I2C: bit **IICEN** del reg. de control. **Comença la transmissió del missatge** amb la generació de la senyal **START** al bus **SDA** pel Mestre
- 3) Escriure en el registre de dades (bits 7 al 1) l'adreça de l'esclau (pantalla) i al bit 0 d'aquest registre, si es una operació de lectura (o escriptura).
- 4) Esperar l'activació del bit (**TCF**) del registre d'estat per assegurar-se que s'ha transmet la dada
- 5) Testejar el bit (**RXAC**) per assegura-se que s'ha rebut la senyal de **ACK** per part del esclau
- 6) Enviar les dades, escrivint-les en el registre de dades. Per cada dada enviada hi ha que repetir els **passos 4** (si es fa enquesta programada) i 5.
- 7) Genera la condició de parada (**STOP**) desactivant el bit (**MST**) del registre de control

Escriure un fragment en llenguatge ensamblador que primer inicialitzi el **controlador I2C** i que després envii **200 caràcters** (ASCII de 8 bits) a la pantalla alfanumèrica. Aquests caràcters estan

emmagatzemats en un **array de caràcters** situat a partir de l'adreça **0x9000** de la **memòria principal**.

- a) Utilitzant la tècnica d'E/S programada amb espera de resposta.
- b) Utilitzant la tècnica de E/S per interrupcions
- c) Si volguéssim que mentre s'executa una RSI no es pugues atendre cap altra interrupció (sistema **no apropiatiu** o **sistema d'interrupcions inhibít**). Quins canvis s'haurien de fer a la RSI anterior?

PROBLEMA 2

Si els 4 primers caràcters del array, situat en memòria, son **'T', 'i', 'm' i 'e'**, mostreu el missatge (fins als 4 primers caràcters i el final d'aquest missatge) que enviarà el controlador I2C, pel cable **SDA** a la pantalla alfanumèrica.

PROBLEMA 3.

Suposar que el temps des de que el controlador I2C activa la línia d'interrupció fins que es comença a la RSI és de 50 ns. Calcular el temps que dedica la CPU a transferir els 200 caràcters del array de caràcters a la pantalla alfanumèrica.

- a) Utilitzant la tècnica de E/S programada amb espera de resposta vista al exercici 1 a)
- b) Utilitzant la tècnica de E/S per interrupcions vista al exercici 1 b)
- c) Amb la tècnica de E/S per interrupcions. Quin percentatge del temps d'execució de la CPU te per dedicar a altres tasques mentre dura la transmissió dels 200 caràcters?

PROBLEMA 4.

Escriure un fragment en llenguatge ensamblador que primer inicialitzi el **controlador del disc dur** i que després envii **1024 bytes, emmagatzemats** en un buffer situat a partir de l'adreça 0x3000 de la memòria (i que emmagatzema el contingut d'un fitxer), al **bloc 100 del disc dur**.

- a) Utilitzant la tècnica d'E/S programada amb espera de resposta.
- b) Utilitzant la tècnica de E/S per interrupcions, escriure sols la RSI i la inicialització del controlador d'E/S del disc (no el codi que instal·la la RSI associada al disc).

PROBLEMA 5.

a) Suposar que el temps des de que el **controlador de disc dur** activa la línia d'interrupció fins que es comença a la RSI és de 50 ns. i que la velocitat de transmissió (anomenada mitjana) del disc dur es de 5 MBytes/s (la de 200 MBytes/s es la màxima, aconseguida en modes especials com el DMA). Calcular el temps que dedica la CPU a realitzar la transferència de 1024 bytes del exercici anterior per la tècnica de E/S d'interrupcions.

b) Podríem realitzar aquesta transferència utilitzant E/S per interrupcions? (pista, penseu en quan triga el controlador del disc dur en estar preparat per rebre un byte i el temps total d'execució de la RSI)

c) Quina seria la funció lògica que generaria la senyal CS del controlador del disc dur?

PROBLEMA 6.

a) Escriure un fragment de codi en llenguatge ensamblador per a programar el controlador de DMA i que transfereixi **8192 bytes** situats a partir del **bloc 1000** del disc (ocupen 8 blocs, del 1000 al 1007) a un **buffer de memòria que comença a l'adreça 0xA000** . Suposar que la transferència es posa en marxa al escriure el valor **0Fh** en el registre de control del DMA.

b) Si la CPU tingues l' **espai d'adreces de E/S separat del d'adreces de memòria principal**. Que canviaria respecte al codi anterior?

PROBLEMA 7.

a) Cada vegada que el controlador DMA demana el bus a la CPU, aquesta li dona prioritat immediatament (mecanisme de cicle robat). Per a la transferència d'una dada (1 byte a la velocitat de transferència màxima del disc de **200 MBytes/s**) entre el controlador del disc i la memòria, s'utilitza el **bus de memòria durant 3 ns**, i es torna a cedir l'ús del bus a la CPU. Aquests 3 ns inclouen el temps de cessió del bus, el temps de la transferència pel bus, i el temps de recuperació del bus.

Calcular el percentatge del temps que dedica la CPU a l'operació de transferència per DMA dels **8 Kbytes** del exercici anterior. Suposar que la **CPU dedica 500 ns per a programar el DMA** (codi d'inicialització del DMA més coordinació amb el controlador del disc) i **500 ns per a reconèixer la finalització de la tasca del DMA** i que la **CPU no pot fer cap tasca** durant tot el temps en que el bus està ocupat per part del controlador DMA.

b) Per la transacció del **exercici 4**, **1024 bytes** emmagatzemats a partir de l'**adreça 0x3000** de la memòria **al bloc 100** del disc dur (velocitat de transmissió del disc dur es de 5 MBytes/s). Quin % del temps d'execució dedicaria la CPU si s'utilitzés el mètode de transferència per **DMA**. Com en el apartat **a**, la CPU dedica 500 ns per a programar el DMA i 500 ns per a reconèixer la finalització de la tasca del DMA i la CPU no pot fer cap tasca durant tot el temps en que el bus està ocupat per part del controlador DMA.

c) Comparar % del temps d'execució dedicat per la CPU a la transferència dels **1024 bytes del exercici 4**, al utilitzar el mètode de transferència per **DMA** o el d'**interrupcions**.

PROBLEMA 8.

a) Completar el diagrama de blocs del sistema computador utilitzant els elements descrits en la figura, afegint els senyals necessaris i les seves interconnexions, i indicant el seu sentit (entrada, sortida, o entrada/sortida). **En aquest apartat NO considerar la generació dels senyals CS.**

b) Generar la funció lògica de la línia CS del controlador DMA (sortida 0 Address decoder) suposant que:

- Els ports es troben a l'**espai compartit d'adreces de memòria** (descodificació completa).
- Els ports es troben a l'**espai separat d'adreces de E/S** (descodificació completa).

PROBLEMA 9.

a) Escriure un fragment en llenguatge ensamblador que realitzi les següents opcions:

- Inicialitzi l'adreça inicial de la taula de vectors d'interrupció
- Activi les interrupcions globalment i les de tots els controladors del sistema (activació local de les interrupcions pel controlador I2C, el del disc i el del DMA)

b) Completar la taula (omplint les caselles ombrejades) que indica les passes a realitzar per a la gestió d'una interrupció del controlador I2C. Indicar quan estan actives i desactives els senyals INT e INTA.

MÒDUL	BUS Adreces	BUS DADES	INT	INTA	INT2	INTA2	Comentari
I2C	-----	-----	----	----	ACT	DES	Es pot enviar un nou caràcter pel bus SDA i el controlador I2C activa el senyal INT
Cont. INT.							
CPU							
CPU	-----	-----	ACT	ACT	ACT	DES	CPU activa INTA i dóna inici al cicle de reconeixement d'interrupció

ARQUITECTURA DE COMPUTADORS I PERIFÈRICS

Escola d'Enginyeria
Enginyeria Telecomunicacions

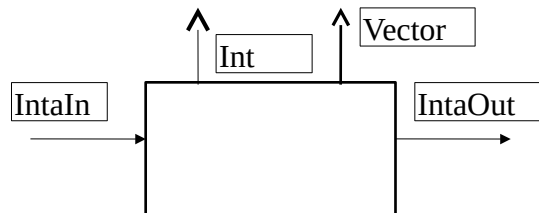
PROBLEMES
ENTRADA/SORTIDA



Cont. INT.							
I2C							
Cont. INT.							
CPU	-----	VECTOR	DES	DES	DES	ACT	CPU llegeix l'identificador del vector d'interrupció del bus de dades, el situa en el reg. MCAUSE i desactiva INTA
Cont. INT.							
CPU							
Memòria	-----	DIR. RSI²	DES	DES	DES	DES	Memòria contesta amb l'adreça de la RSI associada al I2C
CPU							
CPU	-----	-----	DES	DES	DES	DES	Modifica el Comptador de Programa per a què contengui l'adreça de la primera instrucció de la RSI

PROBLEMA 10.

Tenim tres mòduls d'entrada/sortida amb capacitat d'interrupció com el següent:



a) Mostrar la connexió dels mòduls amb la CPU, considerant que la CPU té una única línia de petició d'interrupció i una única línia de reconeixement.

b) Si cada una de les rutines d'atenció als mòduls d'E/S triguen 20 ms (considerar que el temps de reconeixement de la interrupció està inclòs), mostrar l'evolució de la pila si la seqüència d'arribada de les interrupcions es la següent: Mòdul 2 (als 5 ms), Mòdul 3 (als 12 ms) i Mòdul 1 (als 20 ms). Considerar també que quan s'atén una interrupció el sistema d'interrupcions queda inhibit.

Problema 11.

Per transmetre **rafegues** de fins a **64 bytes**, s'ha afegit un buffer de 64 bytes al controlador de DMA. Cada cop que el controlador DMA demana el bus a la CPU, aquesta li dona prioritat immediatament. Per a la transferència d'**una ràfega o bloc de dades de 64 bytes (de 4 bytes en 4 bytes, pel màxim ample de banda del bus de dades)** entre el controlador de disc i la memòria, s'utilitza el bus de memòria durant **18 ns** i es torna a cedir l'ús del bus a la CPU. Aquests 18 ns

inclouen el temps de cessió del bus (1 ns) , el temps de la transferència pel bus (1 ns per cada dada de 4 bytes), i el temps de recuperació del bus (1 ns). La programació i la finalització de la transmissió per DMA (amb una RSI) consumeixen un total (entre les dues tasques) de 1 us de temps d'execució de la CPU.

a) Suposant que la CPU no pot fer cap tasca durant tot el temps en que el bus està ocupat per part del controlador DMA. Per la transferència de l'exercici 6, de 8 Kbytes entre el controlador del disc dur i un buffer de memòria. Quin percentatge de temps està ocupada la CPU sense poder executar codi efectiu d'altres programes durant la transferència per DMA amb rafegues de 64 bytes?

b) Les mateixes condicions que l'apartat anterior, però la CPU pot seguir executant instruccions durant el 25% del temps en que el bus està ocupat per part del controlador DMA. Quin percentatge de temps està ocupada la CPU sense poder executar codi efectiu d'altres programes durant la transferència?