

### **PROBLEMA 3: (30%)**

(aprox.: 1 hora)

Durant l'execució de programes és molt habitual fer lectures seqüencials de dades col·locades de manera consecutiva a la memòria (arrays, etc.). Una de les estratègies per augmentar la velocitat global d'accés a dades consecutives és mitjançant el mode *burst* d'accés a memòria. Un  $\mu P$  amb capacitat de mode *burst* és el i80486DX de Intel.

La interfície d'aquest microprocessador és:

- A2-A31: Bus d'adreces
- ADS\*: Address Strobe
- D0-D31: Bus de dades (4 bytes)
- BE0\*-BE3\*: Data Strobes de cadascun dels bytes que conformen el bus de dades:  
(BE0: D0-D7, BE1: D8-15, BE2: D16-23, BE3: D24-D31)
- W/R\*: Senyal de escriptura-lectura

La durada d'un cicle de bus normal, sense cicles d'espera és de 2 cicles de rellotge. Si es vol accedir consecutivament a 8 bytes cal fer 2 cicles de lectura seguits, trigant en total 4 cicles de rellotge. En mode *burst* ho pot fer amb 3 cicles de rellotge de la manera següent:

- En el primer cicle de rellotge ordena la lectura dels 8 bytes al mateix temps, a dos bancs de memòria independents de 32 bits cadascun.
- En el segon cicle, llegeix els 4 bytes primers (32 bits del primer banc).
- En el tercer cicle, llegeix els 4 bytes últims (32 bits del segon banc).

El senyal que indica si el microprocessador vol llegir els primers 4 bytes (primer banc) o els segons (segon banc) és A2, tal com es pot veure al cronograma de lectura de la figura 1.

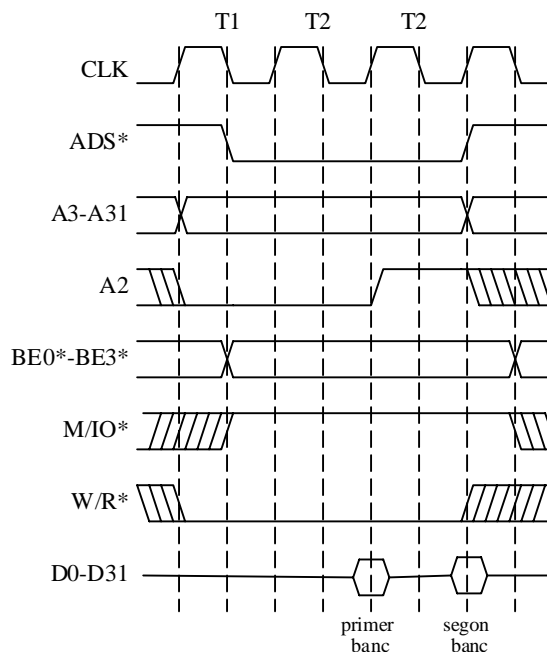


Fig. 1: Cronograma de lectura de 8 bytes en mode *Burst* del i80486.

En T1 el  $\mu P$  comença el cicle de lectura activant ADS\* i posant l'adreça que vol llegir, amb A2=0. En el primer T2, A2=0 i es llegeixen els primers 4 bytes. En el segon T2, A2=1 i es llegeixen els segons 4 bytes.

S'ha dissenyat un sistema basat en el  $\mu P$  i80486DX. Una versió simplificada de l'esquema en cicles de lectura es pot veure a la figura 2:

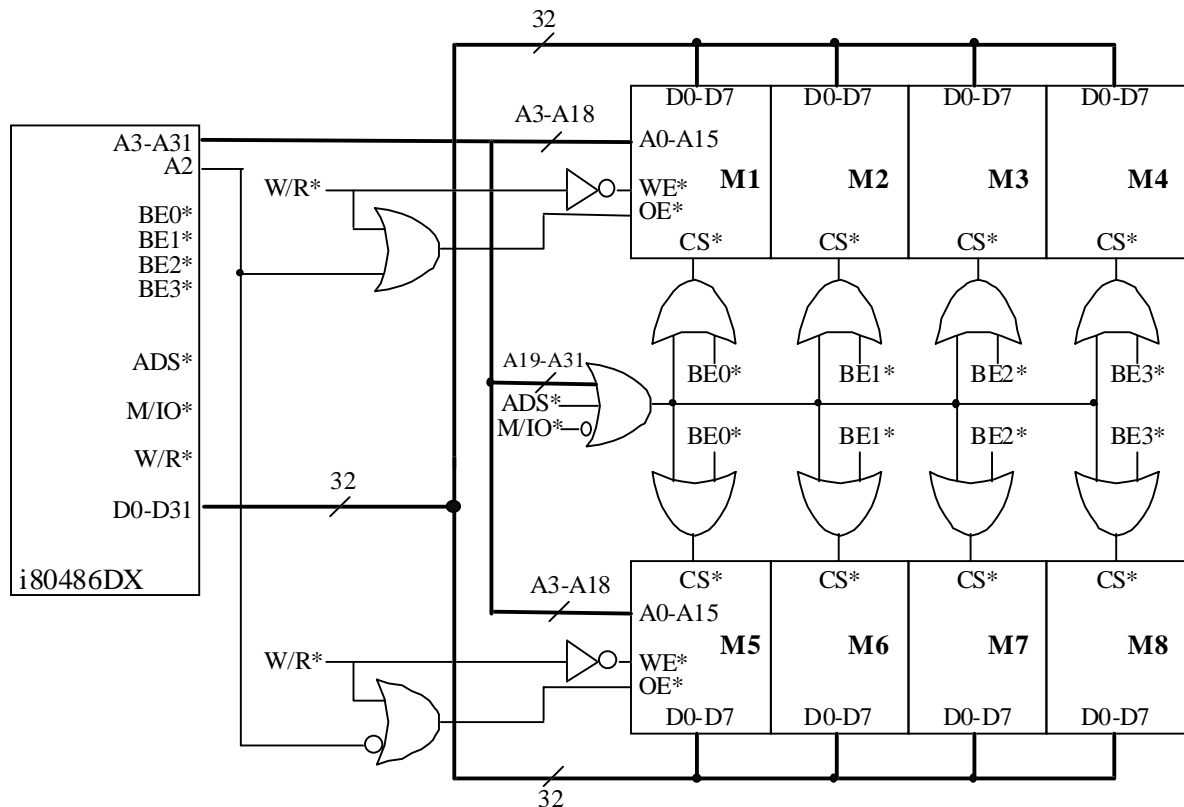


Fig. 2: Esquema del sistema.

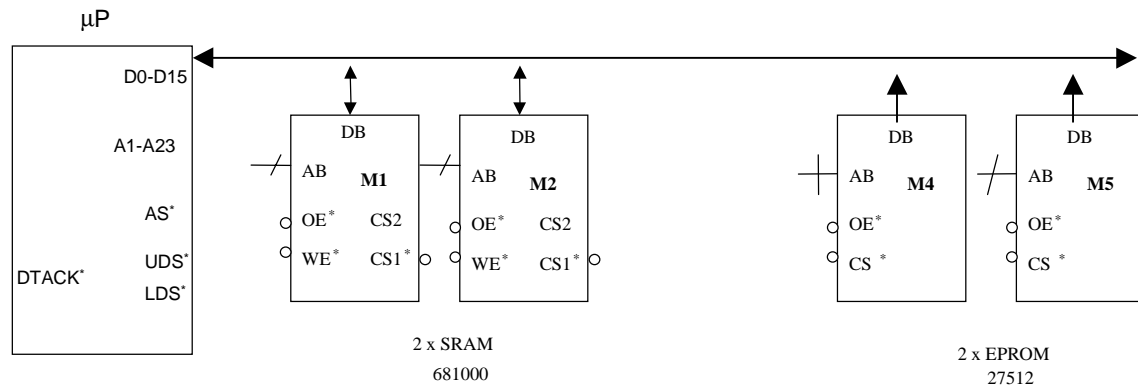
Es demana:

- (2 punts) Mapa de memòria del sistema, especificant clarament a quina memòria estan assignats els bytes. Hi ha zones imatge ?
- (3 punts) Avaluar si s'accedeix correctament a les memòries M1-M4 a nivell de temporització.
- (3 punts) Avaluar si s'accedeix correctament a les memòries M5-M8 a nivell de temporització.
- (2 punts) El mode *Burst* real del i80486DX funciona amb 4 bancs de memòria. Quants cicles de rellotge trigaria en transferir 32 bytes en mode *burst* ? I en mode normal ?

#### NOTA:

- Freq. de treball: 40MHz
- Temps d'accés de RAM des d'adreces ( $t_{AA}$ ) i CS\* ( $t_{CC}$ ) = 35ns
- Temps accés de RAM des de OE\* ( $t_{OE}$ ) = 10ns
- Retard de porta ( $t_P$ ) = 3 ns

Suposem que la memòria RAM es forma a partir de dos chips del tipus 681000 (128K x 8), i que volem situar a partir de la posició 20000<sub>H</sub> del mapa.



- Explicar breument quin és el procés que es posa en marxa en un sistema com el que s'ha presentat per arribar a localitzar la subrutina de servei d'una interrupció d'usuari. (1 punt)
- Dissenyar l'esquema de descodificació més simple possible, a partir de les dades del sistema i del diagrama de la figura. (4 punts)
- Es ben conegut que els µP similar al 68000 utilitzen el senyal DTACK\* per tal de gestionar el seu bus asíncron. Quina és la missió del senyal DTACK\*? Com es podria generar dins del sistema anterior? (1 punt)

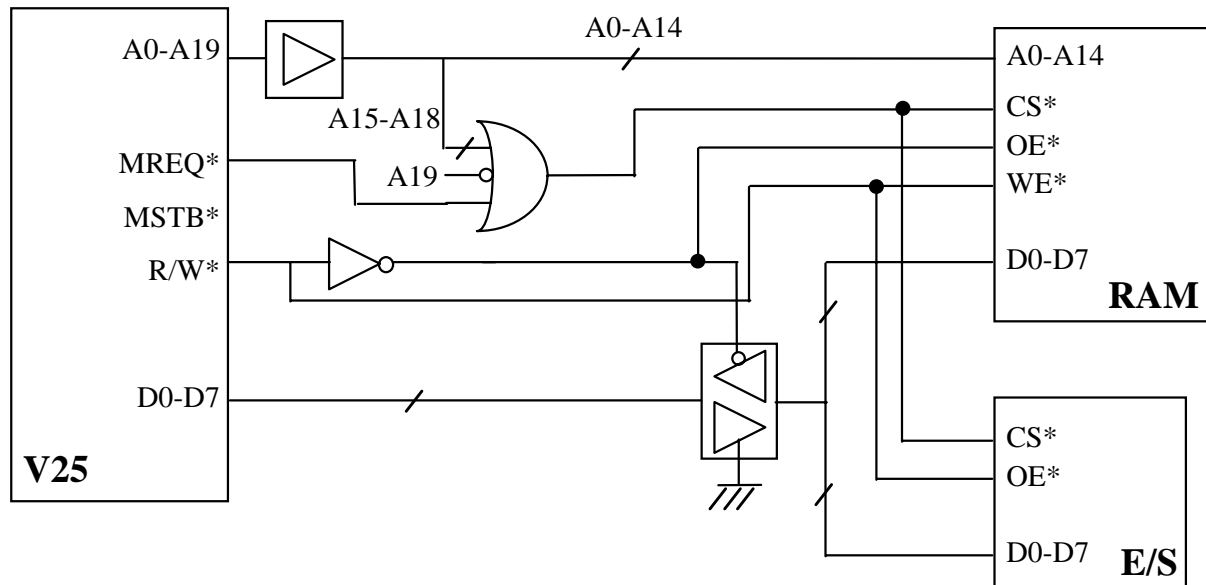
## Problema 2

Es vol dissenyar un sistema basat en el µC V25 que permeti realitzar transferències de dades d'un perifèric d'E/S a una RAM de la manera més optimitzada possible.

Normalment el procés de lectura de dades d'un perifèric té la següent forma:

```
in al,[@ port E/S]      /* Transf. de la dada del perifèric d'E/S a un registre (p.e. al)*/
mov [@ ram],al          /* Transf. de la dada del registre al a la RAM */
```

Aquest procés implica que per a transferir una dada es necessiten dues instruccions: una de lectura del perifèric d'E/S i una altra per escriure la dada a la memòria RAM. Per optimitzar la velocitat de transferència s'ha optat per la solució de la figura 1. L'objectiu d'aquest esquema és que cada cop que el microcontrolador executa una instrucció per a escriure una dada a la RAM, s'escriui de fet una dada provinent del perifèric, en comptes de l'especificada per la instrucció.



**Fig. 1:** Esquema del disseny.

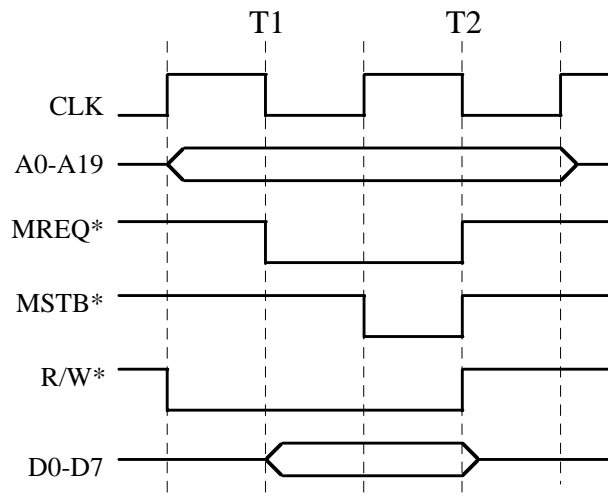
Aquest disseny permet realitzar una transferència completa entre el perifèric i la memòria amb una única instrucció del tipus:

*mov [@ RAM],00 /\* a l'adreça @ de la memòria, s'escriu una dada X del perifèric \*/*

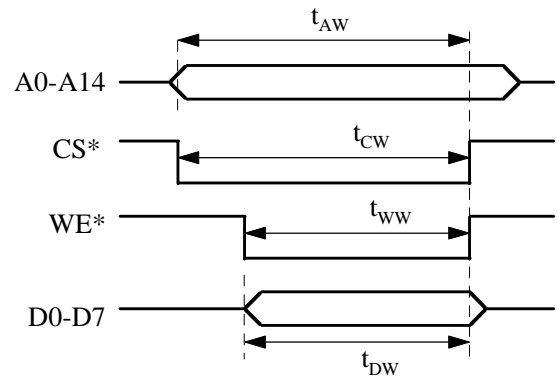
Es demana:

- Marge d'adreces ocupades per la RAM al mapa de memòria. És una descodificació completa? Per què? (1 punt)
- Pot el  $\mu C$  llegir una dada de la RAM? Pot escriure una dada a la RAM? (1 punt)
- El  $\mu C$ , pot llegir una dada directament del perifèric? (transferència de dada des de perifèric a registre intern  $\mu C$  amb una única instrucció). (1 punt)
- Quins son els resultats de l'execució de les següents instruccions ? (1 punt)  
*mov [@ RAM], 00                      mov [@ RAM], 10*
- Es demana avaluar si una transferència entre el dispositiu d'E/S i la memòria té una temporització correcta.

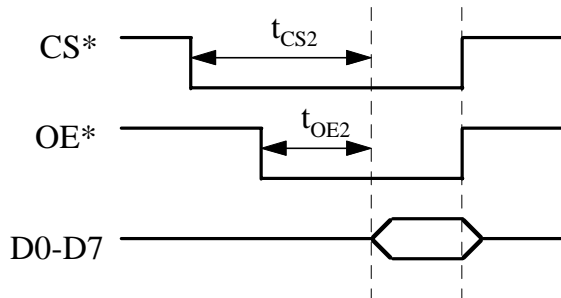
A la figura 2 es pot veure el cronograma d'un cicle d'escriptura del  $\mu C$  V25, a la figura 3 el cronograma d'un cicle d'escriptura a la memòria RAM i a la figura 4 el cronograma de lectura del perifèric. Considerar a la figura 3 que la finalització de l'escriptura es dona quan es desactiva  $CS^*$  o  $WE^*$ .



**Fig. 2:** Cicle d'escriptura del V25.



**Fig. 3:** Cicle d'escriptura RAM.



**Fig. 4:** Cicle de lectura del perifèric.

Temps	Max	Min
$t_{AW}$	-	200 ns
$t_{CW}$	-	200 ns
$t_{WW}$	-	150 ns
$t_{DW}$	-	150 ns
$t_{CS2}$ (perifèric)	100 ns	-
$t_{OE2}$ (perifèric)	75 ns	-

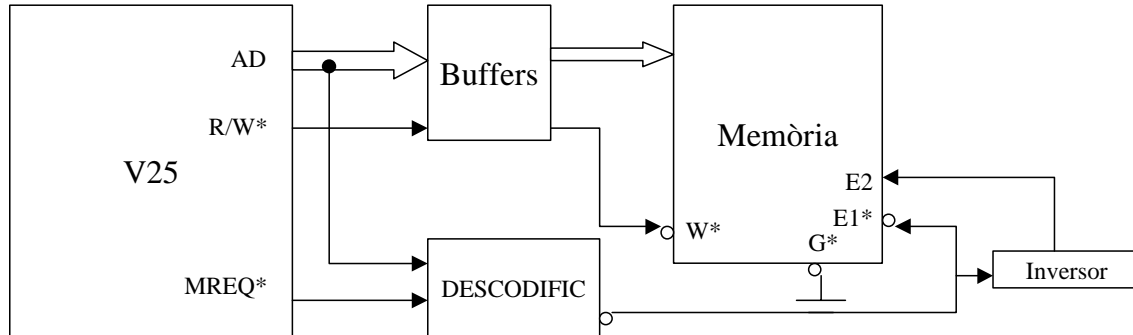
**Taula 1:** Temporitzacions RAM i E/S.

**NOTA:** Freqüència de rellotge (CLK) 4 Mhz. Retard portes i/o buffers: 10 ns

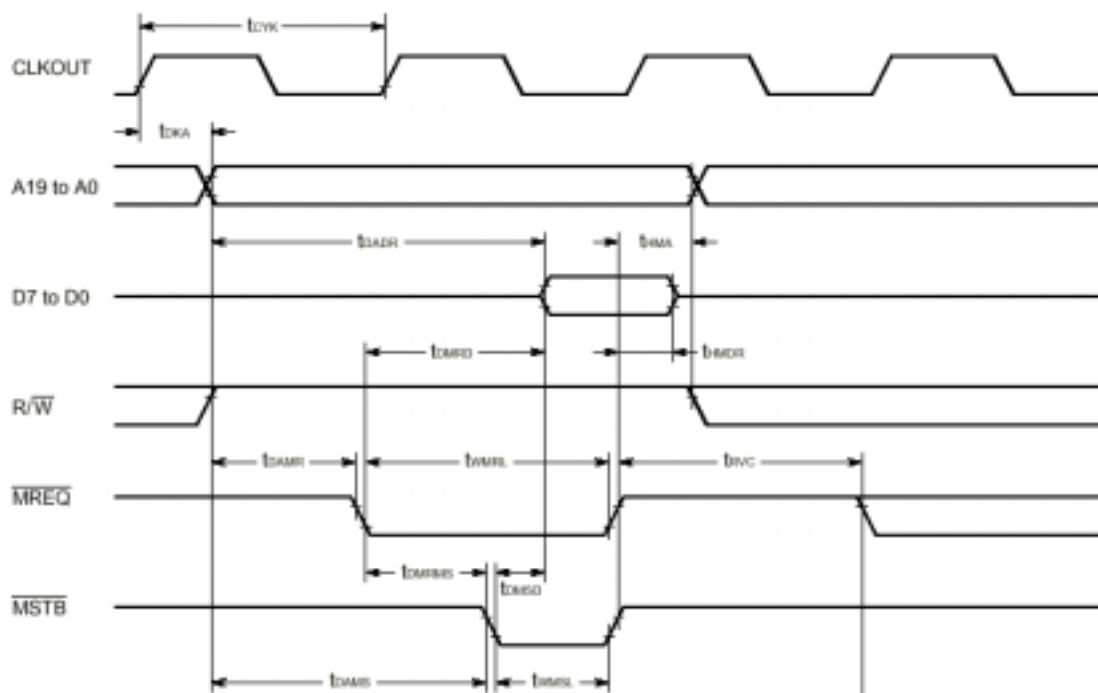
**Nom i Cognoms:** \_\_\_\_\_

### PROBLEMA 2 (20%)

Suposeu que tenim el microprocessador V25 connectat amb una memòria SRAM, segons un esquema tradicional de descodificació, que permet accessos de lectura i d'escriptura.



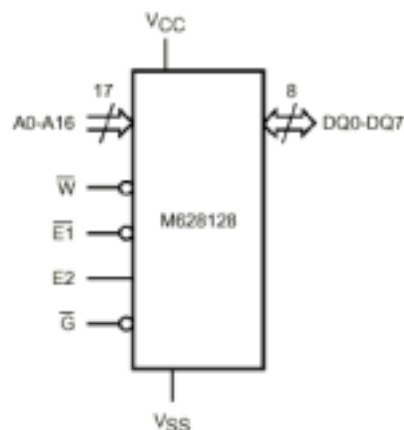
Podem suposar que el sistema de buffers indicat introdueix un retard de 10 ns als senyals que passen a través seu, que l'inversor introdueix també 10 ns i que el descodificador n'afegeix 15 ns. Tindrem en compte que no utilitzarem el senyal MSTB\* del V25, que el diagrama de temps per la operació de lectura i corresponent als principals senyals implicats apareix a continuació, junt a les dades corresponents als principals espais temporals que es poden tenir en compte.



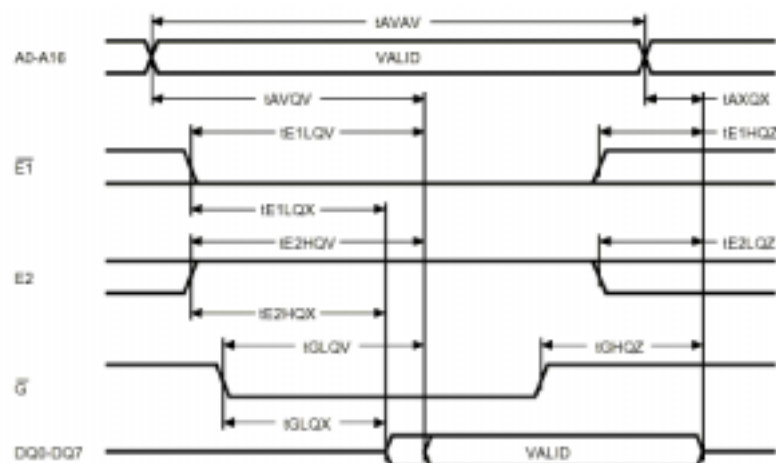
Parameter	Symbol	Test Conditions	MIN.	MAX.	Unit
Address Delay Time from CLKOUT	$t_{DCA}$			90	ns
Data Input Delay Time from Address	$t_{DAOI}$			$(n + 1.5)T - 90$	ns
Data Delay Time from MREQ ↓	$t_{DMRD}$			$(n + 1)T - 75$	ns
Data Delay Time from MSTB ↓	$t_{DMSD}$			$(n + 0.5)T - 75$	ns
MSTB ↓ Delay Time from MREQ ↓	$t_{DMRMS}$		$0.5T - 35$	$0.5T + 35$	ns
MREQ Low-Level Width	$t_{MWL}$		$(n + 1)T - 30$	$(n + 1)T + 30$	ns
Address Hold Time (from MREQ ↑)	$t_{MAH}$		$0.5T - 30$		ns
Data Input Hold Time (from MREQ ↑)	$t_{MDI}$		0		ns
Control Signal Recovery Time	$t_{RVC}$		$T - 25$		ns
Data Output Delay Time from Address	$t_{DAOW}$			$0.5T + 50$	ns
Address Setup Time (to MREQ ↓)	$t_{DAMS}$		$0.5T - 30$		ns
Address Setup Time (to MSTB ↓)	$t_{DAMS}$		$T - 30$		ns
MSTB Low-Level Width	$t_{MWL}$		$(n + 0.5)T - 30$	$(n + 0.5)T + 30$	ns

A la taula n significa el nombre d'estats d'espera introduïts.

Suposem que volem connectar un chip de memòria del tipus M628128, amb una capacitat total de 1 Mbit, i amb un diagrama de blocs tal com el que apareix a continuació:



El diagrama de temps i els valors associats apareixen a continuació:



Nom i Cognoms:\_\_\_\_\_

Symbol	Parameter	M628128				Unit
		-15		-20		
		Min	Max	Min	Max	
$t_{OAV}$	Read Cycle Time	15		20		ns
$t_{WVQZ}^{(1)}$	Address Valid to Output Valid		15		20	ns
$t_{E1LOV}^{(1)}$	Chip Enable 1 Low to Output Valid		15		20	ns
$t_{E2H0V}^{(1)}$	Chip Enable 2 High to Output Valid		15		20	ns
$t_{E1LOV}^{(1)}$	Output Enable Low to Output Valid		7		8	ns
$t_{E1LOZ}^{(2)}$	Chip Enable 1 Low to Output Transition	2		2		ns
$t_{E2H0Z}^{(2)}$	Chip Enable 2 High to Output Transition	2		2		ns
$t_{E1LOZ}^{(2)}$	Output Enable Low to Output Transition	0		0		ns
$t_{E1HZZ}^{(2)}$	Chip Enable 1 High to Output Hi-Z	0	8	0	10	ns
$t_{E2L0Z}^{(2)}$	Chip Enable 2 Low to Output Hi-Z	0	8	0	10	ns

Suposant que s'ha connectat el microprocessador V25 a un circuit de memòria del tipus M628128-20, responeu a les qüestions següents (**pels càlculs següents podeu considerar despreciables els temps de pujada i baixada dels senyals**):

- Quina es la organització interna del chip de memòria?
- Feu una estimació del temps de SETUP del microprocessador.
- Suposant que en el procés de lectura, tal i com s'ha indicat, es deixa actiu permanentment el terminal G\* (Output enable) de la memòria, determineu la freqüència correcta de treball perquè no s'hagi d'afegir cap estat d'espera.



## Problema 2 (35%)

Es vol dissenyar un sistema basat en el  $\mu\text{C}$  V25 de NEC, que contingui RAM dinàmica. Aquest tipus de RAM es caracteritza per:

- 1) la informació es guarda en capacitats: si carreguem un condensador a una certa tensió, aquest mantindrà la tensió només durant un cert temps, depenent de les pèrdues. S'haurà de refrescar la informació periòdicament.
- 2) Per reduir el número de pins de l'encapsulat es multiplexen les adreces:

Tots els bytes de la RAM es distribueixen en una matriu (files x columnes). El que es fa és primer enviar l'adreça de files (Row) i després la de columnes. Això fa que el número total de pins d'adreces del xip sigui la meitat del necessari.

L'esquema intern, simplificat, d'una RAM dinàmica es pot veure a la Figura 1. Com es pot veure, la quantitat total de memòria és 64Kbytes: distribuïda en una matriu de 256x256 bytes. Els bits d'adreces són només 8 (A0-A7), en comptes de les 16 que a priori es necessitarien.

La multiplexació d'adreces consisteix en guardar l'adreça total en dues operacions. Primer es guarda la meitat de l'adreça a un registre (de files) i després l'altra meitat a un altre (de columnes). Aquests registres de files i columnes guarden les adreces (A0-A7) en els flancs de baixada dels senyals RAS\* (Row Address Strobe) i CAS\* (Column Address Strobe), respectivament. Aquest darrer senyal també fa les funcions de CS\*.

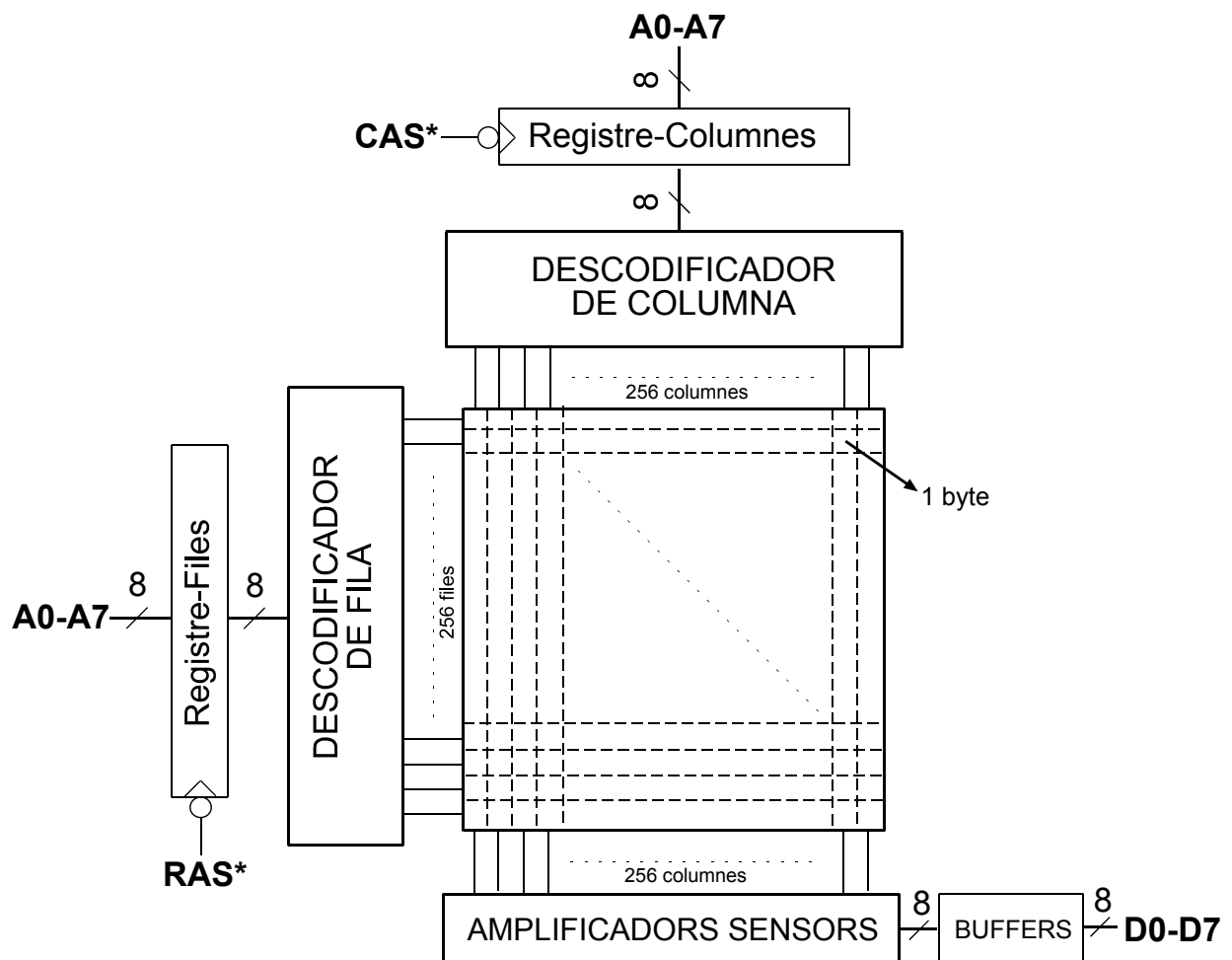


Figura 1: Esquema intern de la RAM dinàmica.

L'operació temporal d'una lectura es pot veure a la figura 2:

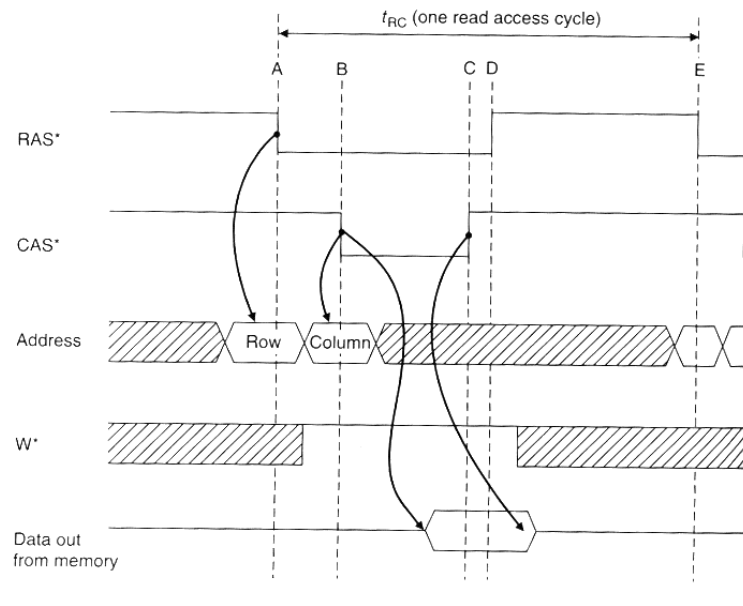


Figura 2: Operació de lectura d'una RAM dinàmica.

- Instant A: amb el flanc de baixada del senyal RAS\* la memòria guarda l'adreça de files.
- Instant B: amb el flanc de baixada del senyal CAS\* la memòria guarda l'adreça de columnes. Ara la memòria ja té l'adreça completa i comença la descodificació. Amb el seu corresponent retard surten les dades pel bus de dades.
- Instant C: amb la desactivació del senyal CAS\*, la memòria deixa, al cap d'un cert retard, el bus de dades en alta impedància.

El sistema dissenyat es pot veure a la següent figura 3:

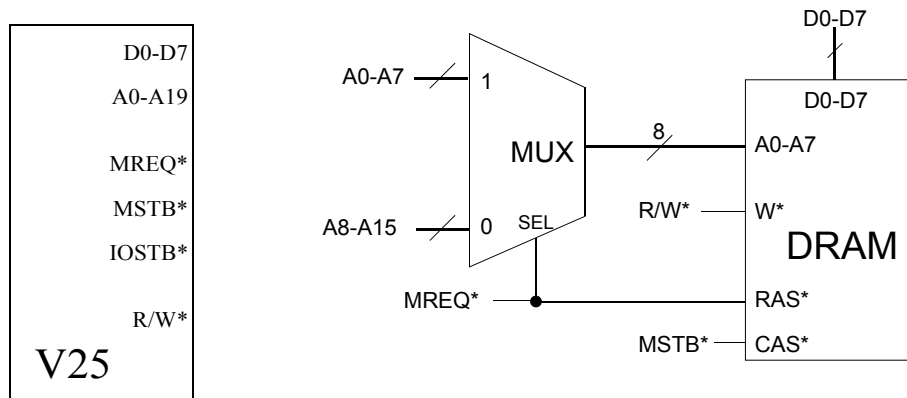


Figura 3: Sistema dissenyat.

En general la temporització requerida entre els senyals RAS\* i CAS\* demana una circuiteria externa d'una certa complexitat. En aquest cas i com que fem servir el V25, aprofitem el retard entre els senyals MREQ\* i MSTB\* per a generar aquest retard i poder fer còmodament la multiplexació d'adreces.

**El retard del multiplexor entre un canvi del senyal SEL i l'actualització de la sortida és de 10 ns (retard fix, sempre el mateix) i la freqüència de treball del V25 és de 10MHz.**

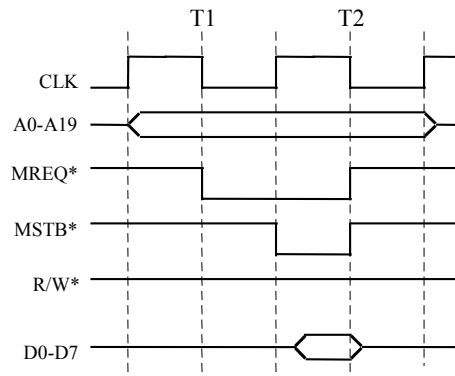


Figura 4: Cronograma lectura V25.

Es demana:

- a) (1 punt) Rang d'adreces ocupat per la RAM al sistema i n° de zones imatge (si n'hi ha).

Rang[@]:

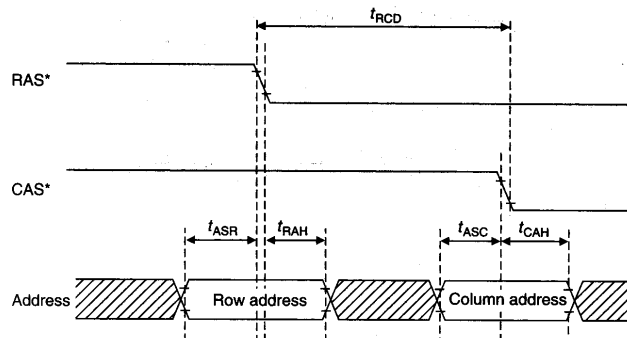
N° Z.I.:

- b) (2 punts) Quan executem la instrucció `aux=*(BYTE far*)MK_FP(0x1234,0x5678)`, quina serà l'adreça, en hexadecimal, de columna i quina la de fila a l'interior de la RAM ?

Fila:

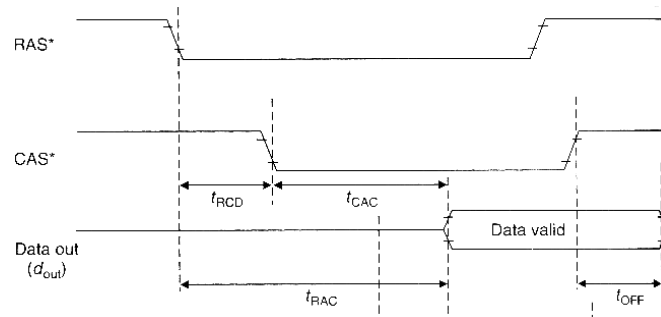
Columna:

- c) (2.5 punts) A la següent figura tenim la temporització requerida entre els senyals RAS\* i CAS\* i les adreces. El nostre sistema compleix tots els cinc requisits temporals ?



$t_{ASR}$ : Temps de setup d'adreces de fila:	5 ns	(mín.)
$t_{RAH}$ : Temps de hold d'adreces de fila:	5 ns	(mín.)
$t_{ASC}$ : Temps de setup d'adreces de columna:	7 ns	(mín.)
$t_{CAH}$ : Temps de hold d'adreces de columna:	7 ns	(mín.)
$t_{RCD}$ : Temps entre activació RAS* i activació CAS*	25 ns (mín.) – 75 ns (max.)	

- d) (2.5 punts) Els temps d'accés de la memòria ( $t_{RAC}$  i  $t_{CAC}$ ) es visualitzen a la següent figura. Tenint en compte el sistema que hem dissenyat i la temporització del V25 en cicle de lectura, funcionarà el sistema? (independentment de la resposta a l'apartat anterior).



$t_{RAC}$ : Temps màxim des d'activació RAS\* fins dada vàlida: 100 ns (temps d'accés)  
 $t_{CAC}$ : Temps màxim des d'activació CAS\* fins dada vàlida: 25 ns (temps d'accés)

Temps setup bus de dades del V25 en lectura: 10ns.

- e) (2 punts) Les memòries dinàmiques necessiten ser refrescades periòdicament. Estan dissenyades de tal manera que quan es fa la lectura d'una dada situada a una fila concreta, es fa automàticament el refresc de tota la fila. Sabent això s'han fet els següents programes:

#### **PROGRAMA A**

```
WORD p=0x0000; // variable global
void interrupt RSI_Timer0 ()
{
    BYTE aux;

    // lectura dada: refresc de la seva fila
    aux=*(BYTE far *)MK_FP(0x0000,p);

    p++;
    FINT;
}
```

#### **PROGRAMA B**

```
void interrupt RSI_Timer0 ()
{
    BYTE aux;
    WORD i,p;

    p=0x0000;
    for (i=0;i<256;i++) {
        aux=*(BYTE far *)MK_FP(0x0000,p);
        p++;
    }
    FINT;
}
```

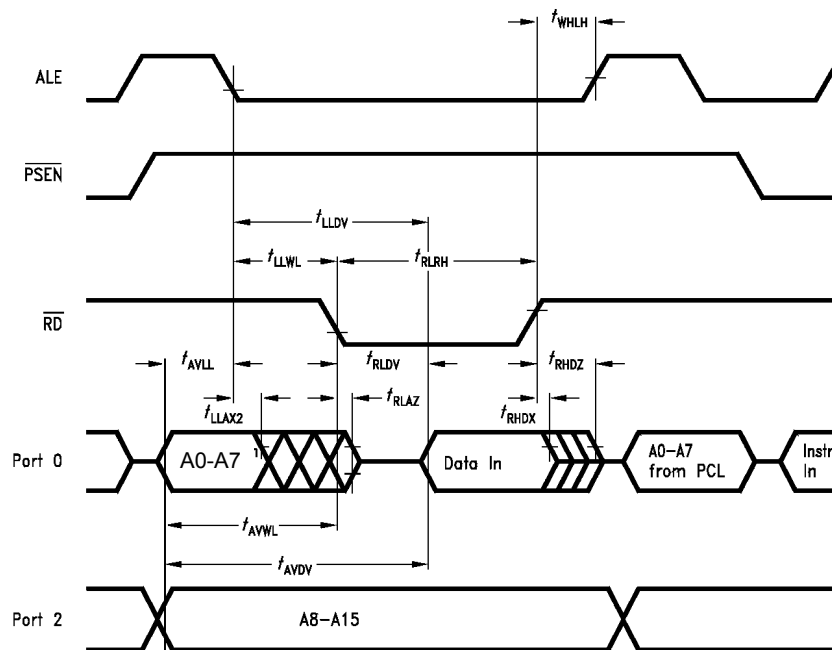
Un byte qualsevol de la memòria s'ha de refrescar cada 50ms com a màxim (i per tant cada 50 ms s'ha d'haver refrescat tota la RAM). Indiqueu la freqüència mínima d'interrupció del Timer0 que s'hauria de programar per cadascun dels dos programes. Quin dels dos triaríeu? Justifiqueu breument la resposta.

Freqüència mínima prog. A:

Freqüència mínima prog. B:

**PROBLEMA 2 (30%)**

A la següent figura tenim el cronograma de lectura de dades del  $\mu\text{C}$  8051.



*Figura 1*

Parameter	Symbol	Limit Values				Unit
		18 MHz Clock		Variable Clock 1/ <i>t</i> <sub>CLCL</sub> = 3.5 MHz to 18 MHz		
		min.	max.	min.	max.	
Address setup to ALE	<i>t</i> <sub>AVLL</sub>	26	—	<i>t</i> <sub>CLCL</sub> − 30	—	ns
$\overline{\text{RD}}$ pulse width	<i>t</i> <sub>RLRH</sub>	233	—	6 <i>t</i> <sub>CLCL</sub> − 100	—	ns
$\overline{\text{WR}}$ pulse width	<i>t</i> <sub>WLWH</sub>	233	—	6 <i>t</i> <sub>CLCL</sub> − 100	—	ns
Address hold after ALE	<i>t</i> <sub>LLAX2</sub>	81	—	2 <i>t</i> <sub>CLCL</sub> − 30	—	ns
$\overline{\text{RD}}$ to valid data in	<i>t</i> <sub>RLDV</sub>	—	128	—	5 <i>t</i> <sub>CLCL</sub> − 150	ns
Data hold after $\overline{\text{RD}}$	<i>t</i> <sub>RHDX</sub>	0	—	0	—	ns
Data float after $\overline{\text{RD}}$	<i>t</i> <sub>RHDZ</sub>	—	51	—	2 <i>t</i> <sub>CLCL</sub> − 60	ns
ALE to valid data in	<i>t</i> <sub>LLDV</sub>	—	294	—	8 <i>t</i> <sub>CLCL</sub> − 150	ns
Address to valid data in	<i>t</i> <sub>AVDV</sub>	—	335	—	9 <i>t</i> <sub>CLCL</sub> − 165	ns
ALE to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	<i>t</i> <sub>LLWL</sub>	117	217	3 <i>t</i> <sub>CLCL</sub> − 50	3 <i>t</i> <sub>CLCL</sub> +50	ns
$\overline{\text{WR}}$ or $\overline{\text{RD}}$ high to ALE high	<i>t</i> <sub>WHLH</sub>	16	96	<i>t</i> <sub>CLCL</sub> − 40	<i>t</i> <sub>CLCL</sub> +40	ns
Address valid to $\overline{\text{WR}}$	<i>t</i> <sub>AVWL</sub>	92	—	4 <i>t</i> <sub>CLCL</sub> − 130	—	ns
Data valid to $\overline{\text{WR}}$ transition	<i>t</i> <sub>QVWX</sub>	11	—	<i>t</i> <sub>CLCL</sub> − 45	—	ns
Data setup before $\overline{\text{WR}}$	<i>t</i> <sub>QVWH</sub>	239	—	7 <i>t</i> <sub>CLCL</sub> − 150	—	ns
Data hold after $\overline{\text{WR}}$	<i>t</i> <sub>WHQX</sub>	16	—	<i>t</i> <sub>CLCL</sub> − 40	—	ns
Address float after $\overline{\text{RD}}$	<i>t</i> <sub>RLAZ</sub>	—	0	—	0	ns

*Taula 1*

Aquest microcontrolador té un bus d'adreces de 16 bits, distribuït en dos ports (Port 0: [A0-A7/Bus de dades] i Port 2: [A8-A15]). El Port 0 és un port multiplexat: al principi del cicle de bus és un bus de sortida on el microcontrolador posa la part baixa del bus d'adreces (A0-A7) i després és el bus bidireccional de dades.

- A l'inici del cicle de bus l'adreça completa (A0-A15) surt per Port 0 i Port 2.
- Externament, s'utilitza un *latch* per guardar la part baixa del bus d'adreces (A0-A7).
- El *latch* és un element de memòria que és transparent ( $Q=D_n$ ) quan el seu *enable* (LE) és actiu ('1') i manté l'última dada ( $Q_{futura}=Q_{passat}$ ) quan *enable* (LE) és inactiu ('0').
- Com a senyal de control d'aquest latch, el microcontrolador disposa d'un senyal específic (ALE: *Address Latch Enable*) que és utilitzat com a senyal habilitador del latch.
- En el moment en que el senyal ALE fa una transició de '1' a '0', el latch emmagatzema A0-A7.

El sistema dissenyat es pot veure a la següent figura:

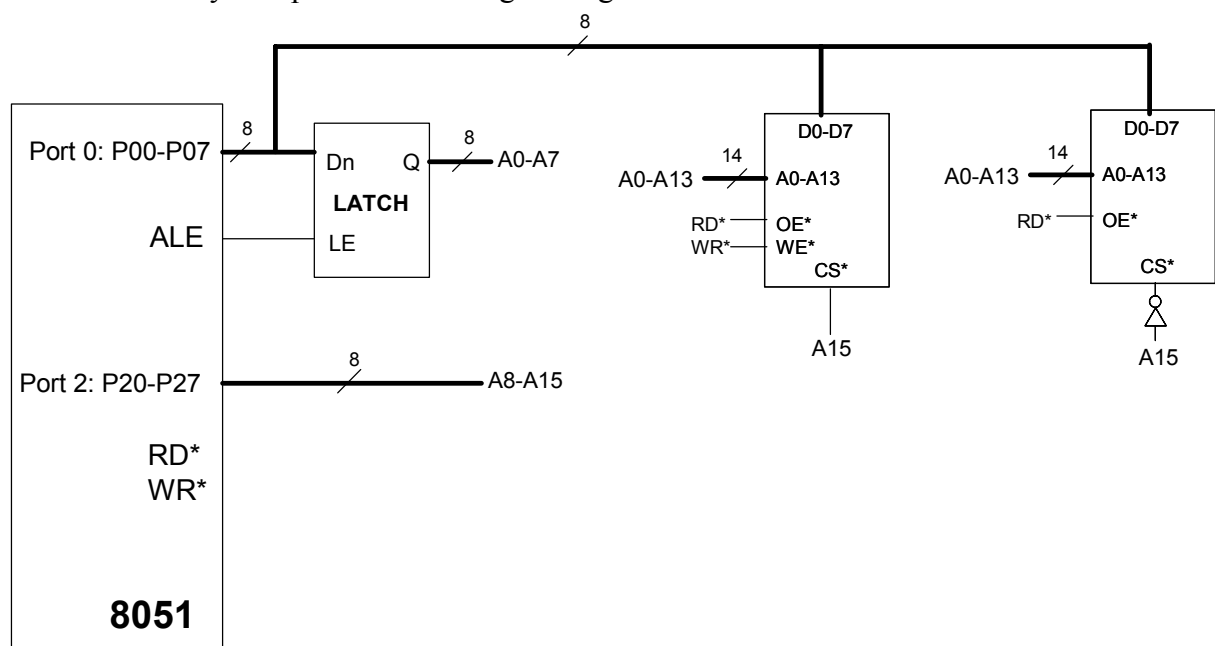


Figura 2

Les característiques del latch que fem servir són les següents:

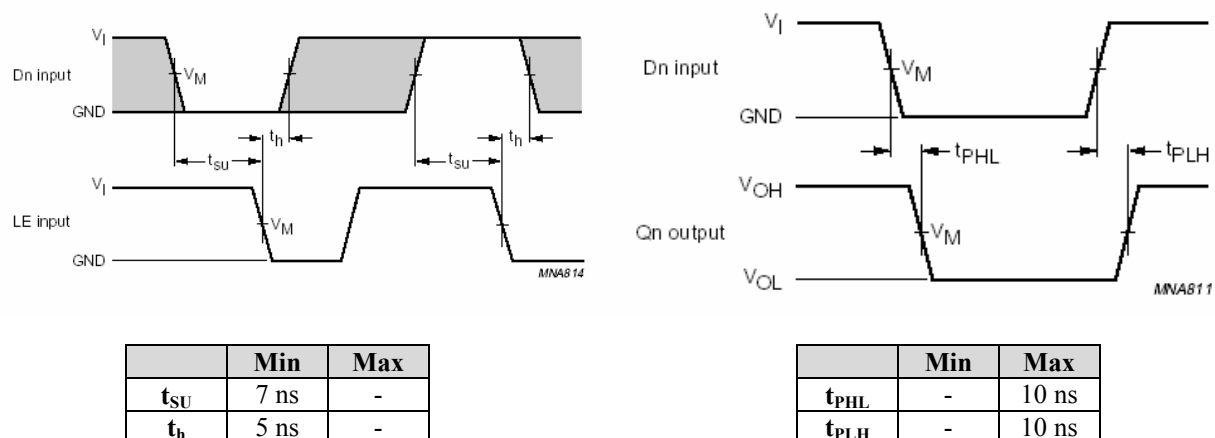
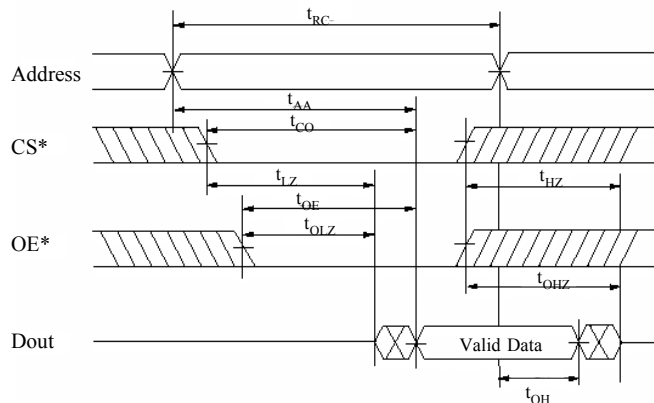


Figura 3

Les característiques temporals en lectura de la RAM utilitzada es poden veure a la següent figura.



	Min.	Max.
$t_{AA}$	-	300 ns
$t_{CO}$	-	300 ns
$t_{OE}$	-	250 ns

Figura 4

Es demana:

1. Rang d'adreces ocupat per la memòria ROM. Hi ha zones imatge ? (10%)

Rang @:

Nº Z.I.:

2. Ens garanteix el microcontrolador que quan activa el senyal RD\* ha deixat prèviament en alta impedància el Port 0 ? Per què ? (20%)
3. Si la freqüència de treball del microcontrolador és de 18MHz, està complint el sistema les especificacions de *setup* i *hold* del latch ? (20%)
4. Quins 3 temps màxims del microcontrolador estan relacionats amb l'aparició de dada vàlida al bus de dades ? Identifiquen els seus noms (Taula 1). (10%)
5. **Tenint en compte la resposta de l'apartat anterior**, quina és la freqüència màxima del microcontrolador que permet accedir correctament a la RAM ? (40%)

## CISE IV – Control (Quadrimestre de Primavera 2008)

**PROBLEMA 1:** (6 punts): Hem connectat una RAM IS61LV25616-10 a un  $\mu\text{C}$  Philips LPC2292, basat en el  $\mu\text{P}$  ARM7TDMI. Els senyals del microcontrolador són:

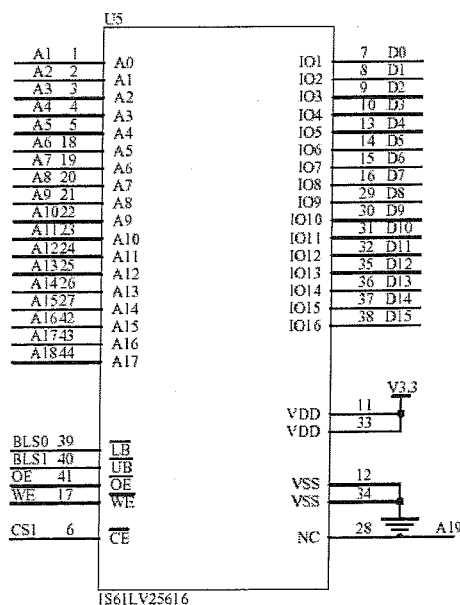
Pin Name	Type	Pin Description
D[31:0]	Input/Output	External memory data lines.
A[23:0]	Output	External memory address lines.
OE	Output	Low-active Output Enable signal.
BLS[3:0]	Output	Low-active Byte Lane Select signals.
WE	Output	Low-active Write Enable signal.
CS[3:0]	Output	Low-active Chip Select signals.

Table 6: External Memory Controller Pin Description

Hem configurat el  $\mu\text{C}$  per tal que faci accessos de 8 o 16 bits a aquesta memòria. És a dir, quan el microcontrolador faci un accés només activarà:

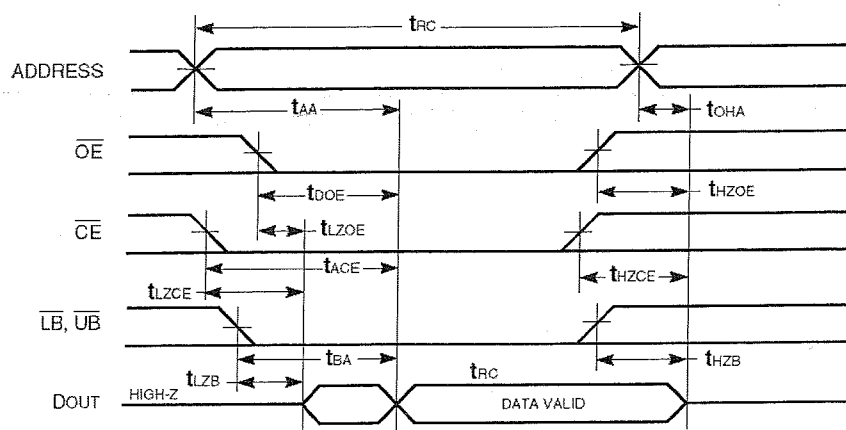
- A1-A23: Bus d'adreces (A0 no pren cap valor significatiu)
- D0:15: Bus de dades de 16 bits (la part D16:D31 NO es fa servir)
- OE\*: Output enable
- WE\*: Write enable
- BLS0\*: Senyal de validació dels bits D0-D7 del bus de dades
- BLS1\*: Senyal de validació dels bits D8-D15 del bus de dades
- CS1\*: Senyal d'habilitació de memòria, generat pel LPC2292, per connectar directament al CS\* d'una RAM. S'activarà per qualsevol accés al rang d'adreces: 0x8100 0000-0x81FF FFFF.

Els senyals amb "\*" són actius a nivell baix. L'esquema de connexions que hem fet és:



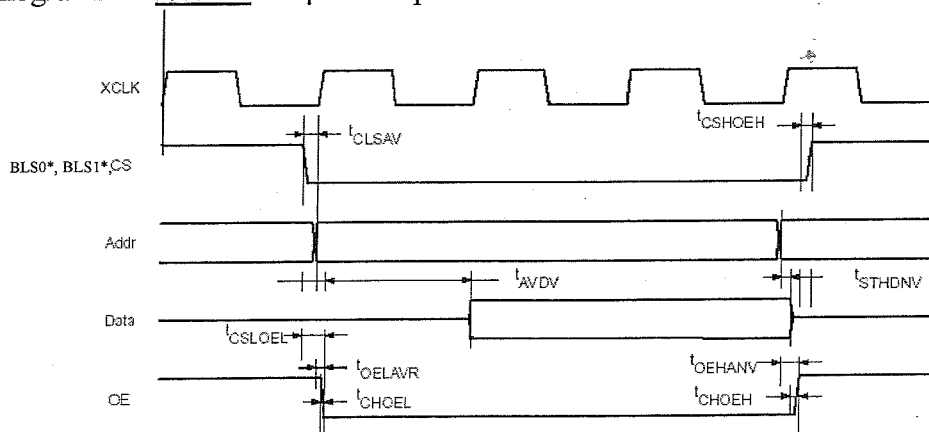
Els senyals LB\* i UB\* de la RAM són senyals d'habilitació de la part baixa i alta respectivament del bus de dades (D0:7 i D8:15). La temporització de la RAM és:





Symbol	Parameter	Min.	Max.
t <sub>RC</sub>	Read Cycle Time	10	—
t <sub>AA</sub>	Address Access Time	—	10
t <sub>OHA</sub>	Output Hold Time	3	—
t <sub>ACE</sub>	CE Access Time	—	10
t <sub>DOE</sub>	OE Access Time	—	4
t <sub>HZOE</sub> <sup>(2)</sup>	OE to High-Z Output	—	4
t <sub>LZOE</sub> <sup>(2)</sup>	OE to Low-Z Output	0	—
t <sub>HZCE</sub> <sup>(2)</sup>	CE to High-Z Output	0	4
t <sub>LZCE</sub> <sup>(2)</sup>	CE to Low-Z Output	3	—
t <sub>BA</sub>	LB, UB Access Time	—	4

El cronograma de **lectura** del µC Philips LPC2292 és:



El datasheet del microcontrolador ens diu que entre que s'ha activat l'últim d'aquests senyals:

Adreces  
CS1\*,  
BLS0\* i/o BLS1\*  
OE\*

i hem de posar la dada al bus de dades, disposem d'un temps màxim  $t_{AVDV} = t_{CYC} * (2 + WST1) - 20$  ns, en lectura.  $t_{CYC} = 20\text{ns} = (50\text{MHz})^{-1}$ , i WST1 és el nombre d'estats d'espera que volem.

Es demana:

- a) (1 punt) Quina capacitat, en bytes, té la memòria ?

$$2^{18} \times 2 = 512 \text{ Kbytes}$$

- b) (2 punts) Tenint en compte que el senyal CS1\*, connectat al CS\* de la RAM, s'activarà per qualsevol accés al rang d'adreces: 0x8100 0000 – 0x81FF FFFF, digueu el nombre de zones imatge, si n'hi ha, de la memòria.

A19, A20, A21, A22, A23

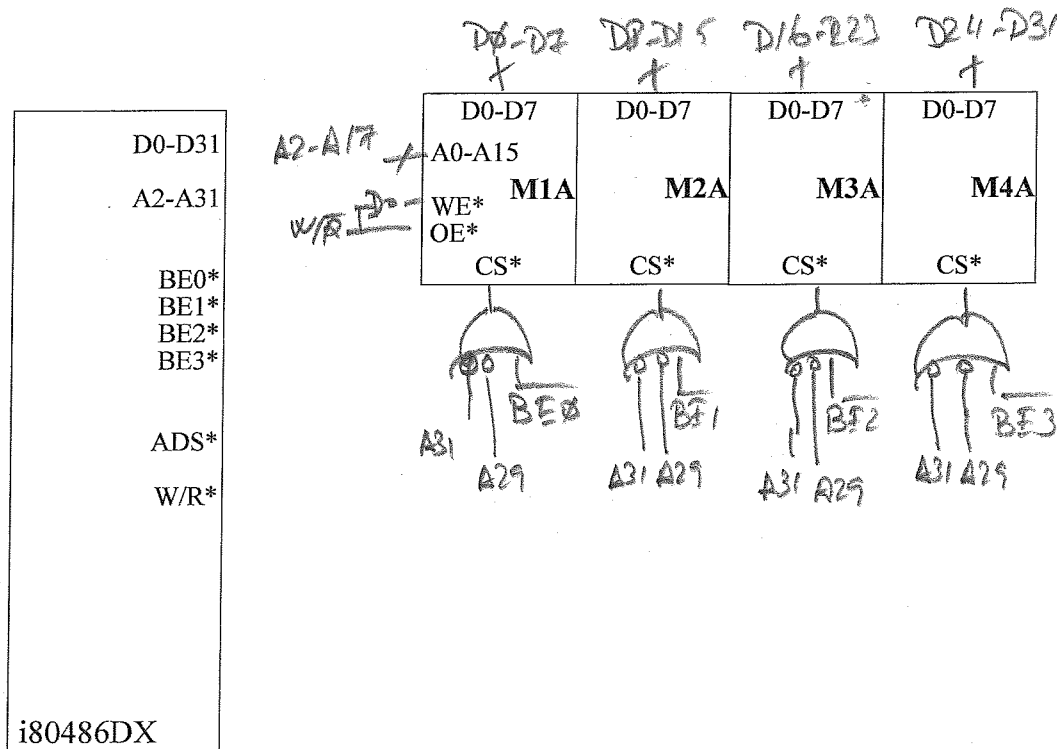
A23 A9

$$2^5 - 32 = 1$$

- c) (3 punts) Calculeu el nombre mínim de cicles d'espera per accedir a la RAM en lectura.

$$t_{AVDV} = T_{CYC} \cdot (2 + WST1) - 20 \text{ ns} \stackrel{T_{CYC} = 20 \text{ ns}}{=} 20 \text{ ns} + WST1 \cdot 20 \text{ ns} \geq 10 \text{ ns} \Rightarrow \boxed{WST1 = 0}$$

**PROBLEMA 2:** (4 punts) Dissenyeu un sistema basat en el  $\mu\text{P}$  i80486, que tingui situats els xips de RAM de la figura en un sol bloc a partir de l'adreça: A0000000. Poden haver-hi zones imatge situades a adreces més altes, però sempre garantint que el rang 00000000-9FFFFFFF no estigui ocupat per aquests xips.



## CISE IV – Control (Quadrimestre de Tardor 2008-2009)

**PROBLEMA 1 (6 punts)** Hem connectat una RAM IS61LV25616-10 a un  $\mu$ C Winbond W90P710CD, basat en el  $\mu$ P ARM7TDMI. Els senyals del microcontrolador són:

- nECS[3:0]: Senyals d'habilitació de memòries, generats pel microcontrolador, per connectar directament al CS\* de les memòries.
- A[21:0]: Bus d'adreces extern.
- D[31:0]: Bus de dades
- nOE: Output enable (actiu a nivell baix)
- nWBE[3:0]: Write enable de cadascun dels bytes del bus de dades (D0:7, D8:15, D16:23, D24:31).
- nWAIT: Senyal d'entrada que serveix per intercalar cicles d'espera. Actiu a nivell baix. Cada cop que el microcontrolador llegeix un '0' afegeix un cicle d'espera.

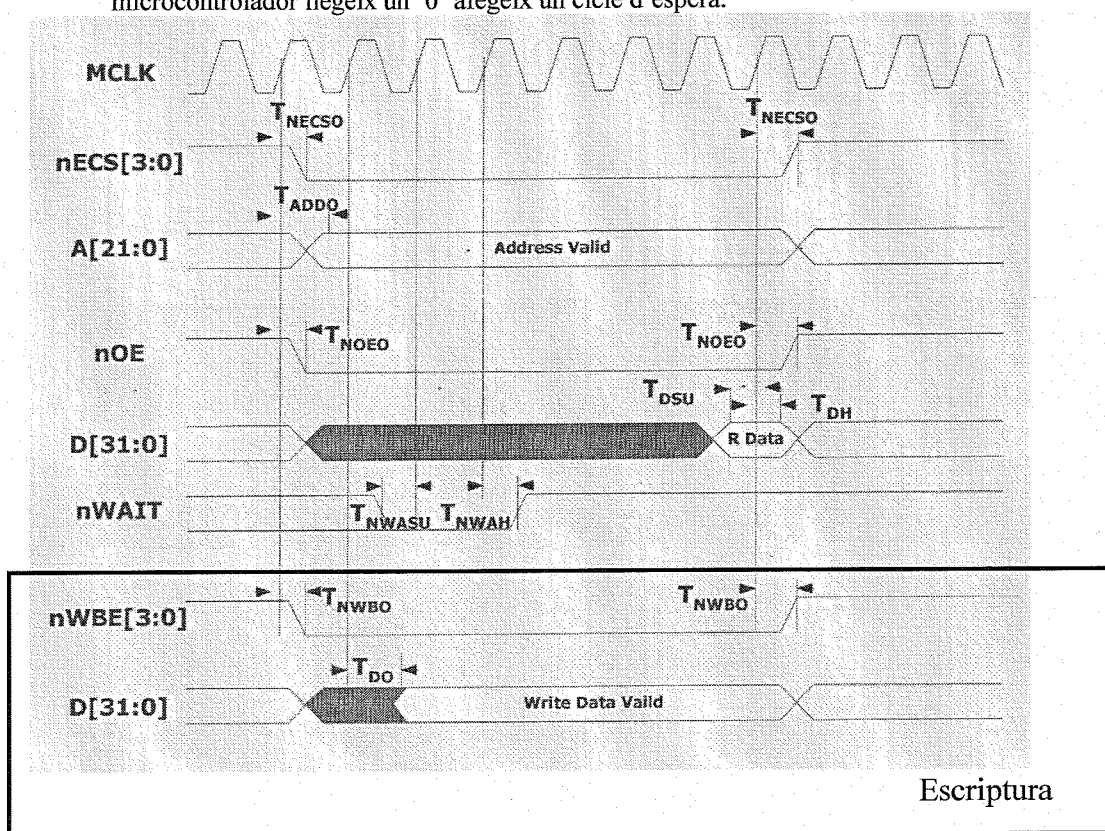
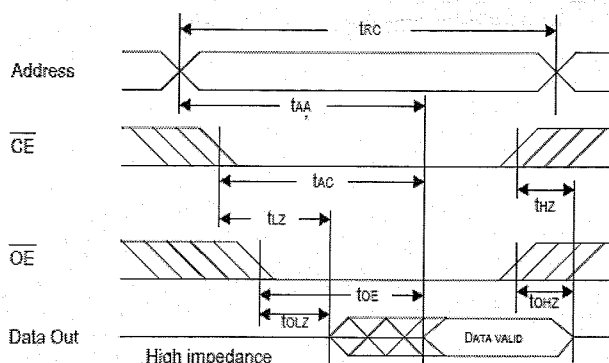


Fig. 1: Cicle de lectura/escriptura del microcontrolador on s'han afegit 2 cicles d'espera amb el senyal nWAIT.

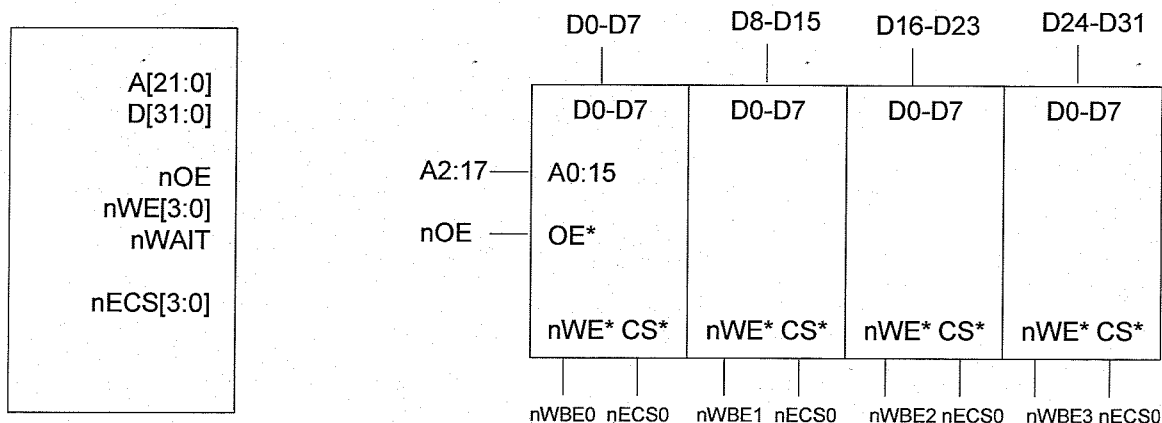
SYMBOL	DESCRIPTION	MIN	MAX	UNIT
T_ADDO	Address Output Delay Time	2	7	ns
T_NECSO	ROM/SRAM/Flash or External I/O Chip Select Delay Time	2	7	ns
T_NOEO	ROM/SRAM or External I/O Bank Output Enable Delay	2	7	ns
T_NWBO	ROM/SRAM or External I/O Bank Write Byte Enable Delay	2	7	ns
T_DH	Read Data Hold Time	7		ns
T_DSU	Read Data Setup Time	0		ns
T_DO	Write Data Output Delay Time (SRAM or External I/O)	2	7	ns
T_NWASU	External Wait Setup Time	3		ns
T_NWAH	External Wait Hold Time	1		ns

La informació del cicle de lectura de la RAM és:



Parameter	Symbol	-7	
		Min	Max
Read cycle time	t <sub>RC</sub>	7	—
Address access time	t <sub>AA</sub>	—	7
Chip enable access time ( $\overline{CE}$ )	t <sub>AC</sub>	—	7
Output enable to output valid ( $\overline{OE}$ )	t <sub>OE</sub>	—	3
Output hold from address change	t <sub>OH</sub>	3	—
Chip enable to output in low Z ( $\overline{CE}$ )	t <sub>LZ</sub> *	3	—
Output enable to output in low Z ( $\overline{OE}$ )	t <sub>OLZ</sub> *	0	—
Chip disable to output in High Z ( $\overline{CE}$ )	t <sub>HZ</sub> *	—	3.5
Output disable to output in High Z ( $\overline{OE}$ )	t <sub>OHZ</sub> *	—	3

El connexionat que s'ha fet és el següent:



Es demana:

a) (1 punt) En el cronograma de la Figura 1, el senyal nWAIT s'ha fet servir per introduir dos cicles d'espera en el cicle de bus. Quants cicles de rellotge hi haurà entre que el microcontrolador treu les adreces i l'instant de captura en lectura, sense cicles d'espera ?

5T

b) (1 punt) Quants cicles de rellotge hi ha entre que el microcontrolador treu les adreces i es llegeix per primera vegada el senyal nWAIT ? Quins els temps de setup i hold associats a la lectura d'aquest senyal ?

2T

T<sub>NWASU</sub> = 3ms

T<sub>NWAH</sub> = 1μs

b) (4 punts) El sistema pot funcionar a la màxima freqüència de rellotge del microcontrolador (80MHz), sense cicles d'espera ? Justifiqueu la resposta.

$$5T - 7\text{ms} \geq 7\text{ms} + t_{DSU} \Rightarrow 5T \geq 14\text{ms}$$

$\uparrow$   $T_{NECSO}$   $\downarrow$   $t_{AA}$   $\uparrow$   $t_{CS}$   
 $T_{ADDO}$   $t_{CS}$

62.5ms  $\geq$  14ms

OK. (M=0)

**PROBLEMA 2:** (4 punts) Dissenyeu un sistema basat en el  $\mu P$  i80486, que tingui situats els xips de RAM a partir de l'adreça 0x0000 0000 i els xips de ROM que acabin a l'adreça 0xFFFF FFFF. Es demana el disseny més simple possible, permetent zones imatge, que deixi lliure el rang d'adreces 0x4000 0000 – 0xEFFF FFFF.

