



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Departament d'Arquitectura de Computadors

Tema 7

Memòria Virtual

Estructura de Computadors (EC)

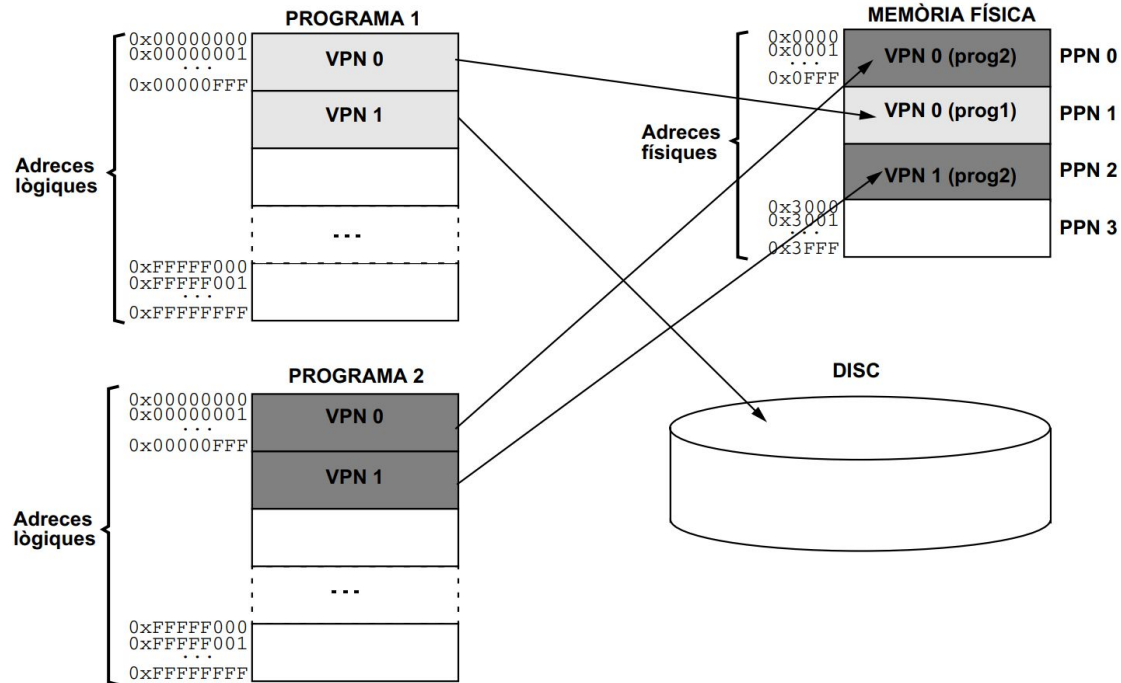
2023 - 2024 Q2

Adrià Armejach (adria.armejach@upc.edu)



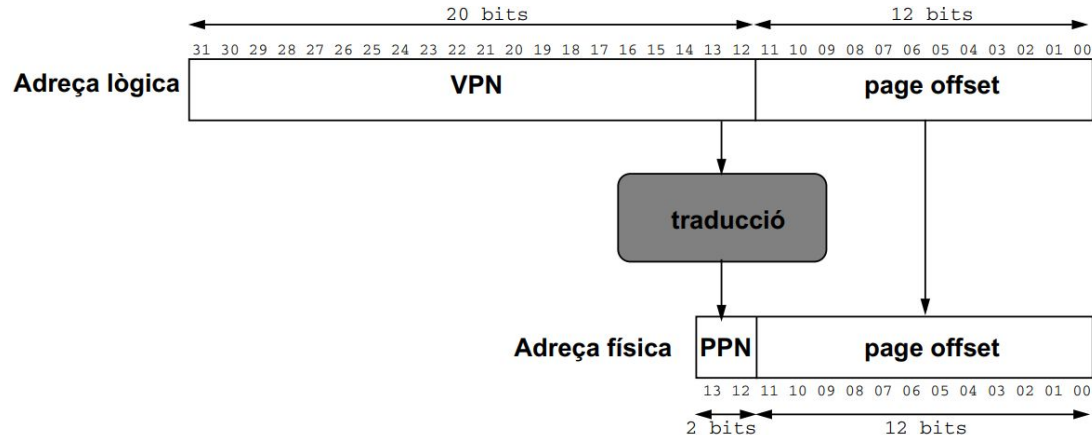
Exemple

- Adreces de 32 bits, pàgines de 4KB, memòria física de 16KB

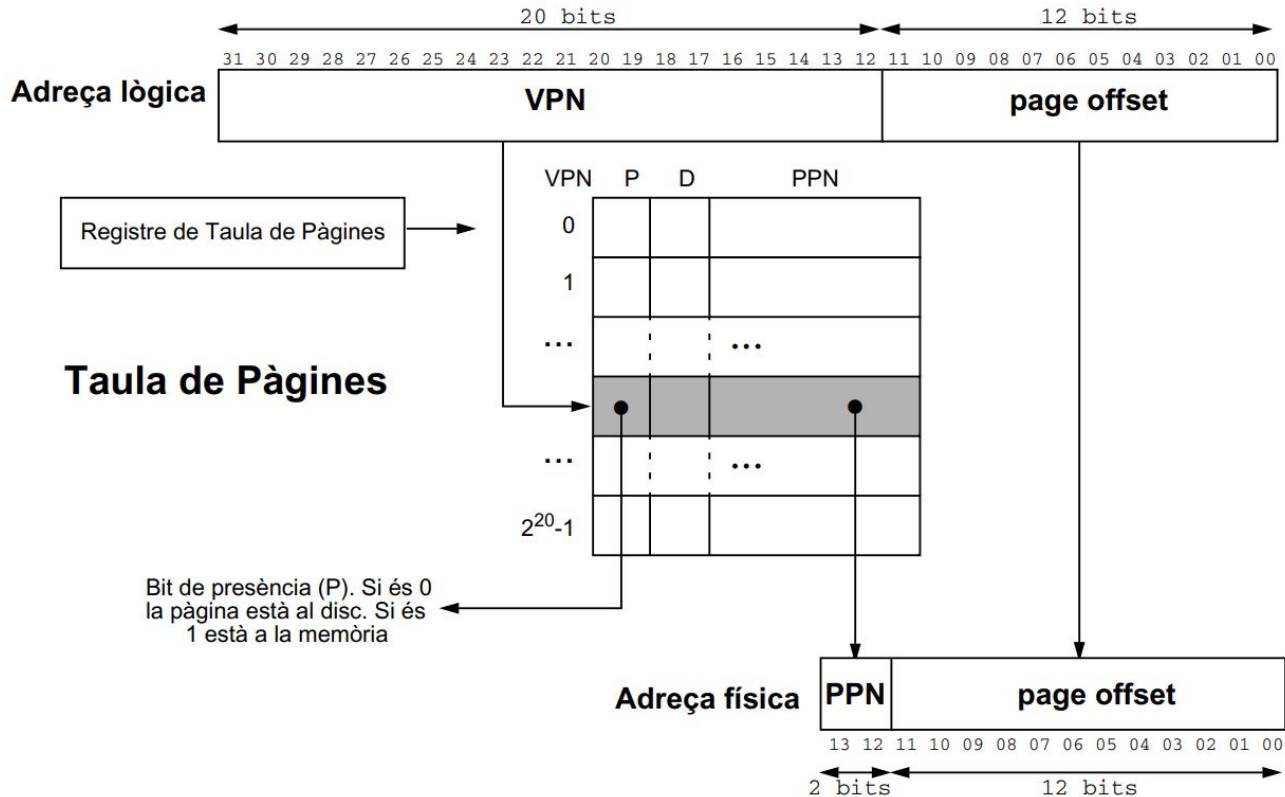


Traducció d'adreces

- El processador treballa amb adreces lògiques
- Per cada load/store s'ha de traduir l'adreça lògica a una adreça física
- MMU (Memory Management Unit):
 - Tradueix el VPN al corresponent PPN
 - L'offset dintre de la pàgina no canvia



Traducció amb taula de pàgines



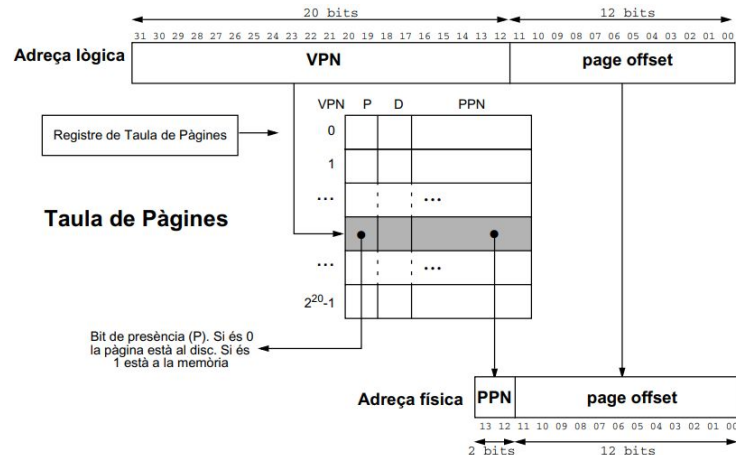


Traducció ràpida amb TLB



Traducció amb la taula de pàgines

- La taula de pàgines està en memòria principal
- Cada load/store requereix **dos accessos** a memòria
 - Un accés a la taula de pàgines per traduir l'adreça
 - Un accés per llegir/escriure la dada
 - Traduir adreces lògiques a físiques introdueix una latència molt elevada



Traducció ràpida amb TLB

- Translation-Lookaside Buffer (TLB)
 - Cache de traduccions
 - Emmagatzema les darreres entrades utilitzades de la taula de pàgines
 - Forma part de la MMU (hardware)

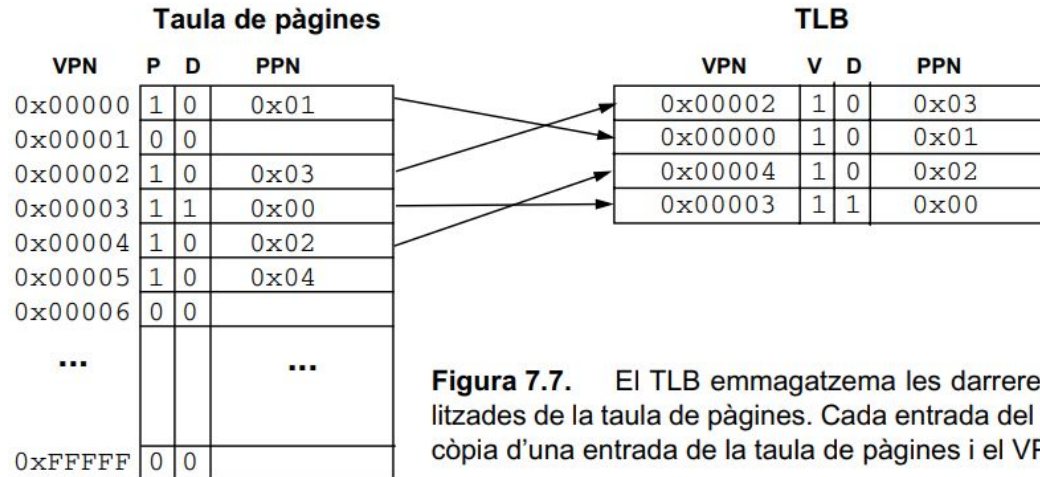


Figura 7.7. El TLB emmagatzema les darreres entrades utilitzades de la taula de pàgines. Cada entrada del TLB conté una còpia d'una entrada de la taula de pàgines i el VPN d'aquesta.

Accessos al TLB

- Al buscar un VPN al TLB:
 - Es compara el VPN que busquem amb els VPNs de totes les entrades del TLB - emplaçament completament associatiu
- Si es troba, tenim un encert de TLB (TLB hit)
 - Fins i tot si el bit de valid (V) val 0 => es produeix una **fallada de pàgina**
 - Un cop resolta s'actualitza la taula de pàgines i l'entrada del TLB
- Si no el trobem es produeix una fallada de TLB (TLB miss)
- A MIPS les fallades de TLB i les fallades de pàgina els gestiona el sistema operatiu (per software)

Fallada de TLB

- Per resoldre una fallada de TLB:
 - S'ha de llegir l'entrada corresponent de la taula de pàgines i copiar-la a una entrada del TLB
 - Si hi ha una entrada lliure al TLB ($V = 0$), s'utilitzarà aquesta entrada
 - Si no hi han entrades lliures al TLB, es produeix un reemplaçament (LRU)
 - Si el bit de presència (P) de la taula de pàgines val 0, el bit V de la entrada del TLB es queda a 0

Fallada de TLB

- Per resoldre una fallada de TLB:
 - S'ha de llegir l'entrada corresponent de la taula de pàgines i copiar-la a una entrada del TLB
 - Si hi ha una entrada lliure al TLB ($V = 0$), s'utilitzarà aquesta entrada
 - Si no hi han entrades lliures al TLB, es produeix un reemplaçament (LRU)
 - Si el bit de presència (P) de la taula de pàgines val 0, el bit V de la entrada del TLB es queda a 0
- El bit V de una entrada del TLB pot valer 0 per dos motius:
 - L'entrada no està inicialitzada (lliure)
 - L'entrada s'acaba de copiar de la taula de pàgines i el bit de presència era 0 (s'està resolent la fallada de pàgina corresponent)

Consistència del bit D del TLB

- El bit D és l'únic que pot ser modificat per l'execució del programa
 - Es posa a 1 quan el programa fa una escriptura (store)
- TLB és una cache
 - S'utilitza política d'escriptura immediata
 - El bit D es posa a 1 simultaneament al TLB i a la taula de pàgines
- No és necessari accedir a la taula de pàgines per totes les escriptures
 - Només la primera vegada que s'escriu a una pàgina
 - Quan el bit D val 0 al TLB

Diagrama del procés de traducció d'adreces amb TLB

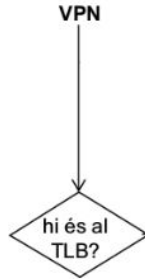


Diagrama del procés de traducció d'adreces amb TLB

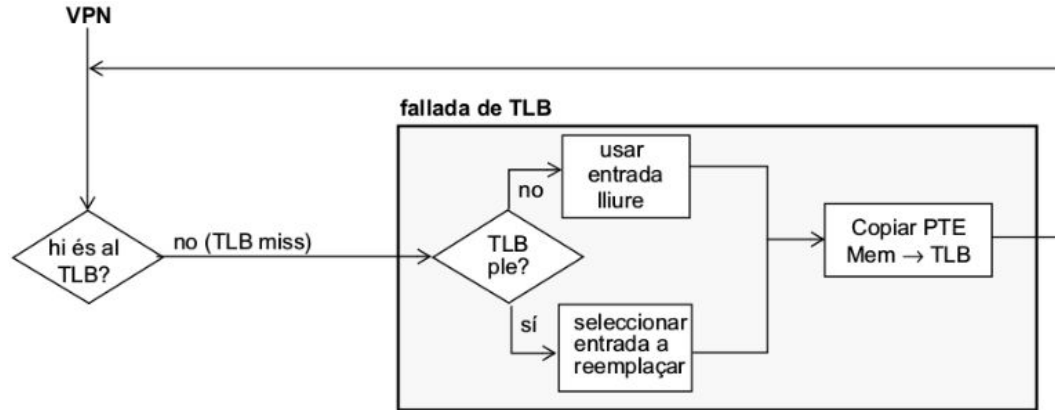


Diagrama del procés de traducció d'adreces amb TLB

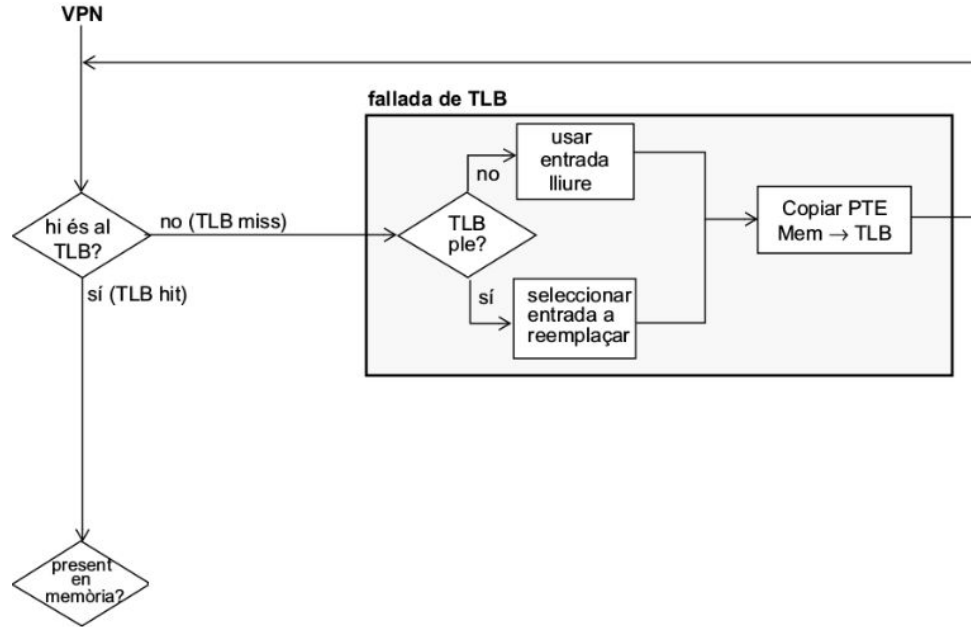


Diagrama del procés de traducció d'adreces amb TLB

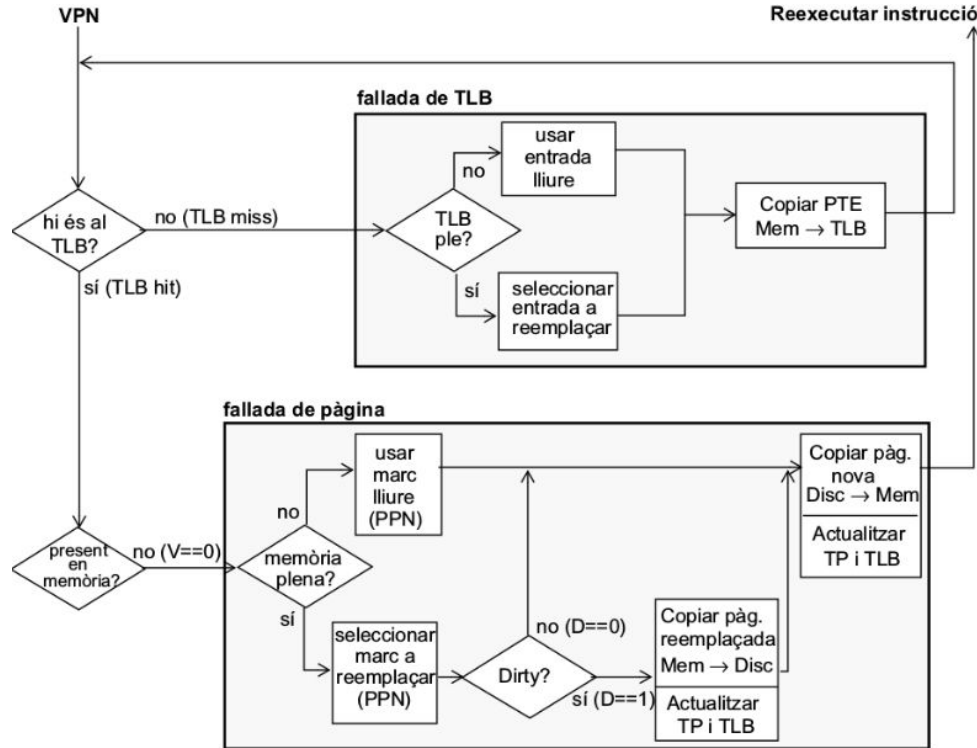
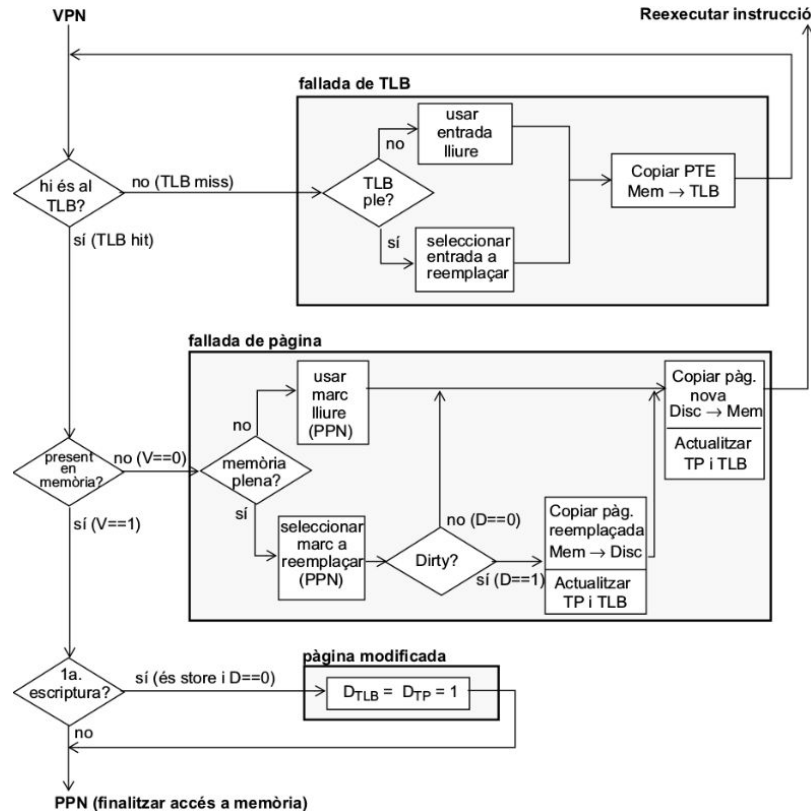


Diagrama del procés de traducció d'adreces amb TLB





Protecció i Compartició



Protecció amb memòria virtual

- Cada pàgina física està assignada a un únic procés, i no pot aparèixer a la taula de pàgines de cap altre procés
 - El mecanisme de traducció fa impossible que un procés pugi accedir a pàgines físiques d'un altre procés

Protecció amb memòria virtual

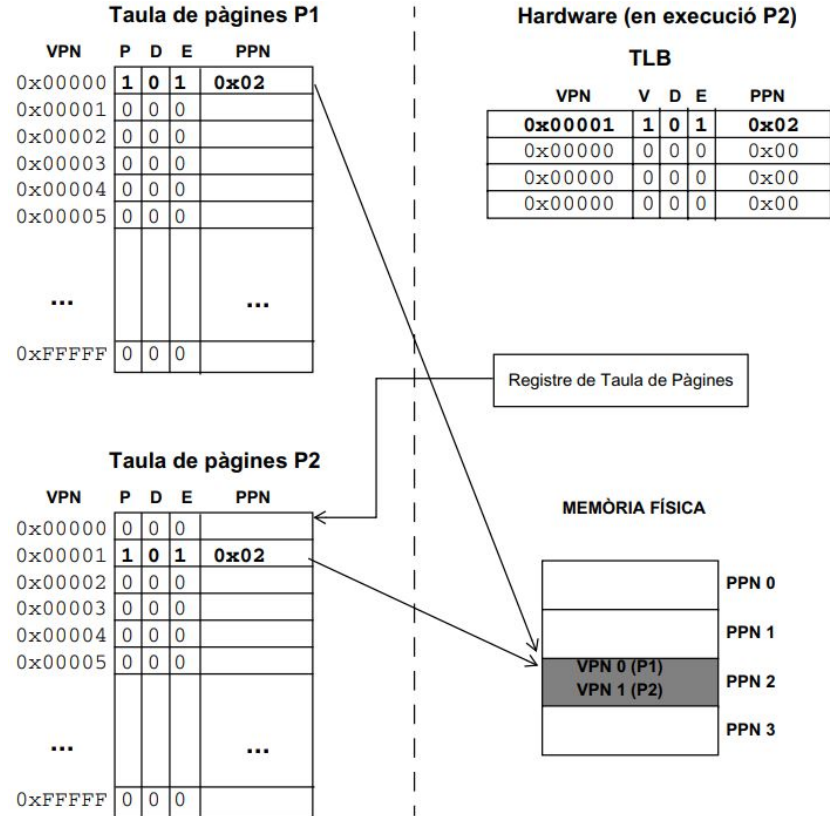
- Cada pàgina física està assignada a un únic procés, i no pot aparèixer a la taula de pàgines de cap altre procés
 - El mecanisme de traducció fa impossible que un procés pugi accedit a pàgines físiques d'un altre procés
- I si un procés intenta modificar la seva pròpia taula de pàgines?
 - La taula de pàgines s'emmagatzema a un espai d'adreces reservat al SO
 - En MIPS adreces lògiques amb el bit 31 = 1
 - El processador té dos modes de funcionament: usuari i sistema
 - Només quan el processador està en mode sistema es pot modificar el TLB i les taules de pàgines

Protecció contra escriptura

- Resulta convenient prohibir l'escriptura en determinades pàgines
 - Per exemple les de codi
- S'inclou un bit de permís d'escriptura (E) a cada entrada de la taula de pàgines i del TLB
- Si un procés intenta escriure a una pàgina amb el bit $E = 0$, el SO aborta l'execució

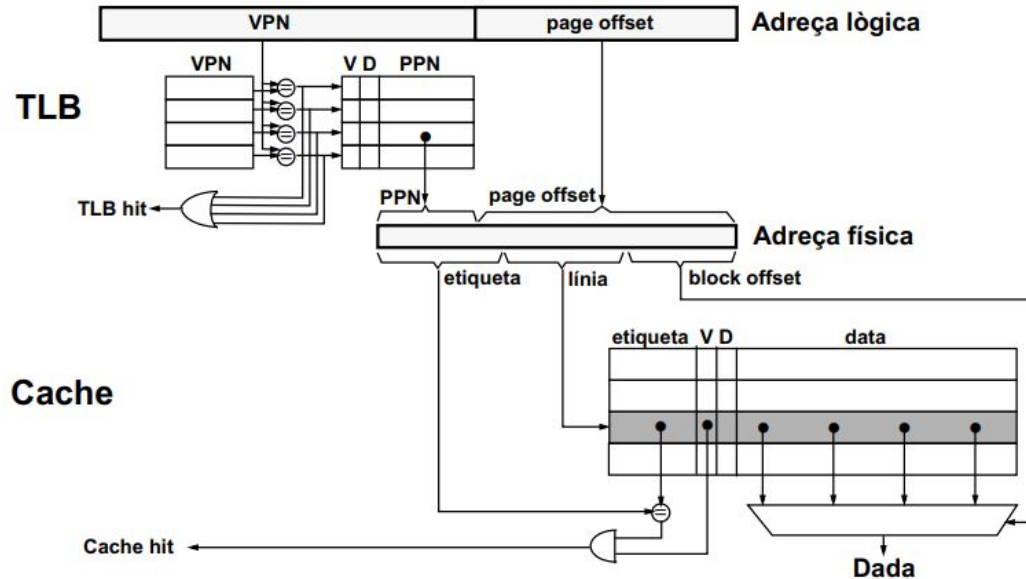
Compartició de memòria entre processos

- Hi pot haver situacions on un procés (P1) vulgui permetre a un altre (P2) accedir al seu espai d'adreçament
- El SO pot habilitar la compartició mitjançant la taula de pàgines



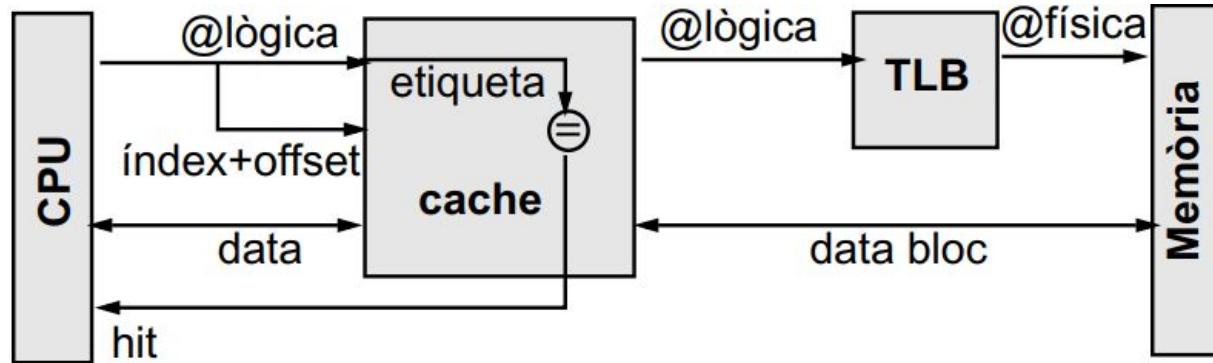
Integració del TLB i la memòria cache

- Memòria **cache indexada físicament**
 - Primer fer traducció amb TLB i a continuació indexar la cache amb l'adreça física
 - Simple però temps d'accés elevat: accedeix seqüencialment al TLB i a la cache



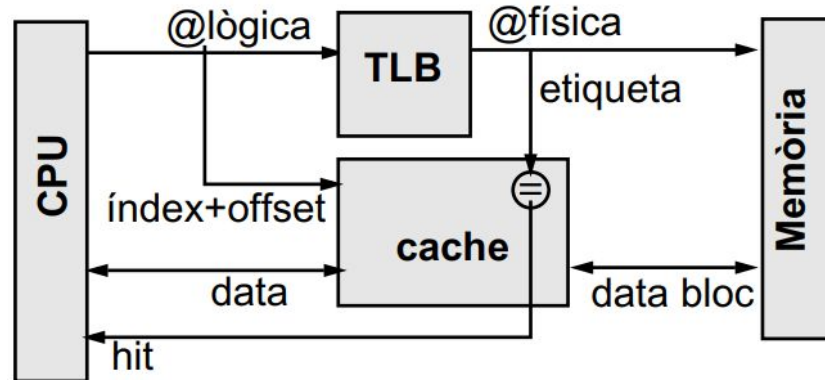
Integració del TLB i la memòria cache

- Memòria **cache indexada virtualment**
 - Temps d'accés més reduït: no s'ha de traduir en cas d'encert
 - Es perd el mecanisme de protecció - problema: aliasing



Integració del TLB i la memòria cache

- Memòria **cache indexada virtualment i etiquetada físicament**
 - S'inicia l'accés a la cache amb l'adreça virtual (lògica)
 - L'index o conjunt de la cache s'obté del page offset
 - En paral·lel s'accedeix al TLB
 - Finalment, es comproven les etiquetes utilitzant l'adreça física (bits del PPN)





Exercicis



Exercici 1

Sigui un processador amb memòria virtual basada en paginació amb les següents característiques:

- 16 bits d'adreça lògica, i 15 bits d'adreça física
- mida de pàgines: 8 KB
- reemplaçament LRU

El contingut inicial de la taula de pàgines es mostra a continuació, on P és el bit de presència, D és el bit de pàgina modificada i PPN el número de marc de pàgina.

	P	D	PPN
0	0	0	-
1	0	0	-
2	1	0	3
3	1	1	0
4	0	0	-
5	1	1	1
6	0	0	-
7	0	0	-

Tots els marcs de pàgina poden ser usats pel programa. Inicialment, la memòria física només conté les pàgines lògiques 2, 3 i 5, les quals han estat accedides en aquest mateix ordre (la pàgina 5 és la més recent). La següent taula mostra una seqüència de referències a memòria (E:escriptura/L:lectura). Emplena la taula indicant, per cada referència, el número de pàgina lògica (VPN) i el número de marc de pàgina resultant de la traducció (PPN). Indica també si es produeix una fallada de pàgina, si es llegeix del disc, i si s'escriu al disc. Indica també, en cas de reemplaçar una pàgina, el VPN i PPN de la pàgina reemplaçada.

L/E	adreça (hex)	VPN	PPN	fallada s/n	Disc		Pàg. reemplaçada	
					L s/n	E s/n	VPN	PPN
E	4000							
E	5C44							
L	1BBF							
E	8665							
L	6600							
L	F458							

Exercici 2

Considera un computador MIPS que gestiona memòria virtual paginada amb les pàgines de 4 KB i amb la restricció que el sistema operatiu admet fins a 6 pàgines com a màxim carregades a memòria física per programa. L'algorisme de reemplaçament de pàgines dins de memòria física és LRU.

La gestió de memòria virtual es realitza amb un TLB de 4 entrades, totalment associatiu i amb reemplaçament LRU.

El computador està executant el següent programa:

```
int V[102400];
main() {
    int i, sum=0;
    for (i=0; i<102400; i++) sum += V[i];
}
```

Considera que les variables globals s'ubiquen a partir de l'adreça 0x10010000 i que el codi s'ubica a partir de l'adreça 0x00400000.

El contingut del TLB i de memòria física (MF) en un moment donat de l'execució d'aquest programa, en què s'acaba de tractar una fallada de pàgina, és el següent.

TLB					MF	
entrada TLB	V	D	VPN	PPN	PPN	VPN
0	1	0	0x00400	0	0	0x00400
1	1	0	0x10024	5	1	0x10020
2	1	0	0x10022	3	2	0x10021
3	1	0	0x10023	4	3	0x10022
					4	0x10023
					5	0x10024

Exercici 2

- a) (0,30p) Quin és el VPN (en hexadecimal) de la pàgina que s'acaba de carregar a MF?

VPN =

- b) (0,40p) Quantes pàgines ocupa el vector v?

nombre de pàgines =

- c) (0,40p) Quantes fallades de pàgina es produiran en tota l'execució del programa?

fallades de pàgina =

- d) (0,10p) Quantes fallades de TLB es produiran en tota l'execució del programa?

fallades de TLB =

- e) (0,40p) Quins 4 VPN (en hexadecimal) contindrà el TLB quan s'acabi d'executar tot el programa?

VPN_s =