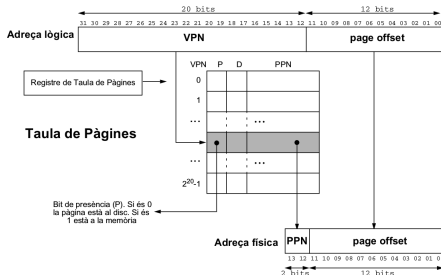


# Estructura de Computadores

## Tema 7. Memoria Virtual

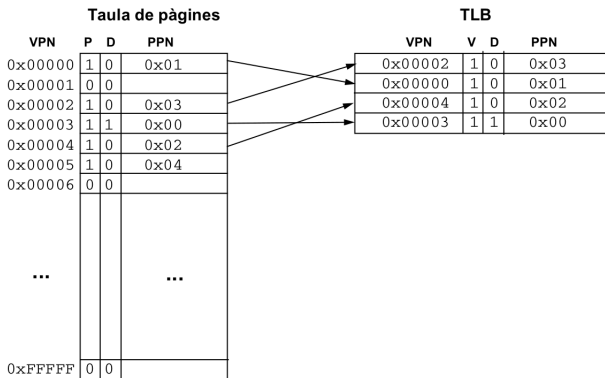
## Traducción con tabla de páginas

- ▶ Cada load/store requiere **dos accesos** a memoria
  - ▶ Un acceso a la tabla de páginas para traducir la dirección
  - ▶ Un acceso para leer/escribir el dato
- ▶ La tabla de páginas está en memoria principal
  - ▶ Traducir la dirección lógica a física introduce una latencia muy elevada en el acceso a memoria



# Traducción rápida con TLB

- ▶ Translation-Lookaside Buffer (TLB)
  - ▶ Cache de traducciones: almacena las entradas más recientemente accedidas de la tabla de páginas
  - ▶ Forma parte de la MMU (hardware)



# TLB

- ▶ Al buscar un VPN en el TLB:
  - ▶ Se compara el VPN que buscamos con los VPNs de todas las entradas del TLB (emplazamiento completamente asociativo)
  - ▶ Si se encuentra tenemos un acierto de TLB
    - ▶ Incluso si el bit de válido (V) es 0, en cuyo caso se produce un **fallo de página**
  - ▶ Si no lo encontramos se produce un **fallo de TLB** (TLB miss)
- ▶ En MIPS los fallos de TLB y los fallos de página los gestiona el sistema operativo (por software)

## Fallo de TLB

- ▶ Para resolver un fallo de TLB:
  - ▶ Hay que leer la entrada correspondiente de la tabla de páginas y copiarla en una entrada del TLB
  - ▶ Si hay una entrada libre ( $V=0$ ), se utilizará dicha entrada
  - ▶ Si no hay entradas libres, se producirá un reemplazo (**LRU**)
  - ▶ Si el bit de presencia (P) de la tabla de páginas es 0, el bit V de la entrada de la TLB se queda a 0

## Fallo de TLB

- ▶ Para resolver un fallo de TLB:
  - ▶ Hay que leer la entrada correspondiente de la tabla de páginas y copiarla en una entrada del TLB
  - ▶ Si hay una entrada libre ( $V=0$ ), se utilizará dicha entrada
  - ▶ Si no hay entradas libres, se producirá un reemplazo (**LRU**)
  - ▶ Si el bit de presencia ( $P$ ) de la tabla de páginas es 0, el bit  $V$  de la entrada de la TLB se queda a 0
- ▶ El bit  $V$  de una entrada del TLB puede valer 0 por dos motivos:
  - ▶ La entrada no está inicializada
  - ▶ La entrada se acaba de copiar de la tabla de páginas y el bit de presencia era 0 (todavía hay que resolver el fallo de página)

## Bit D en la entrada del TLB

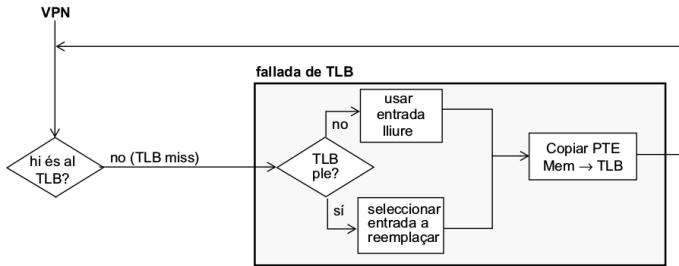
- ▶ El bit D es el único que puede ser modificado por la ejecución del programa
  - ▶ Se pone a 1 cuando el programa realiza una escritura (store)
- ▶ Se utiliza una política de escritura inmediata
  - ▶ El bit D de la tabla de páginas se actualiza de forma inmediata
- ▶ No es necesario acceder a la tabla de páginas para todas las escrituras
  - ▶ Solo la primera vez que se escribe una página, cuando el bit D vale 0 en el TLB

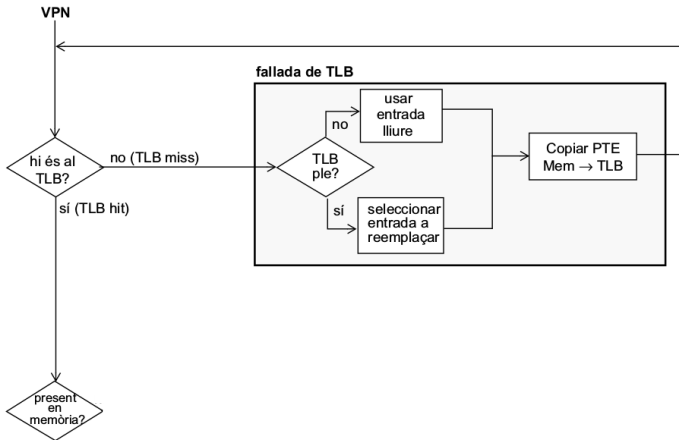


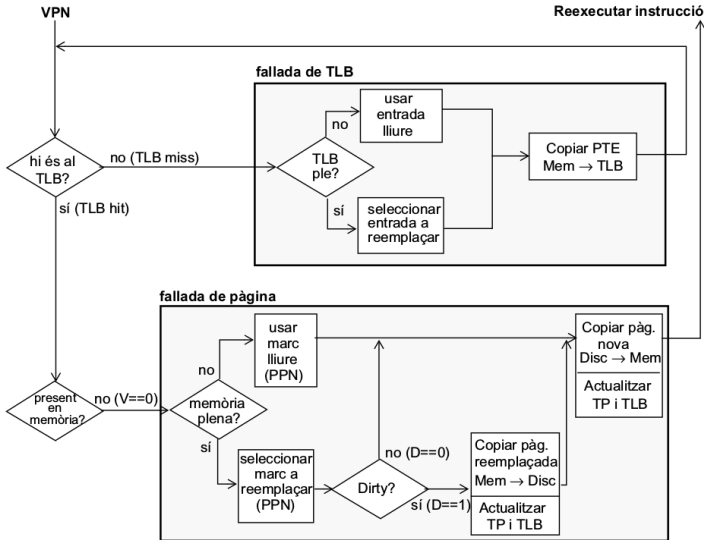
VPN

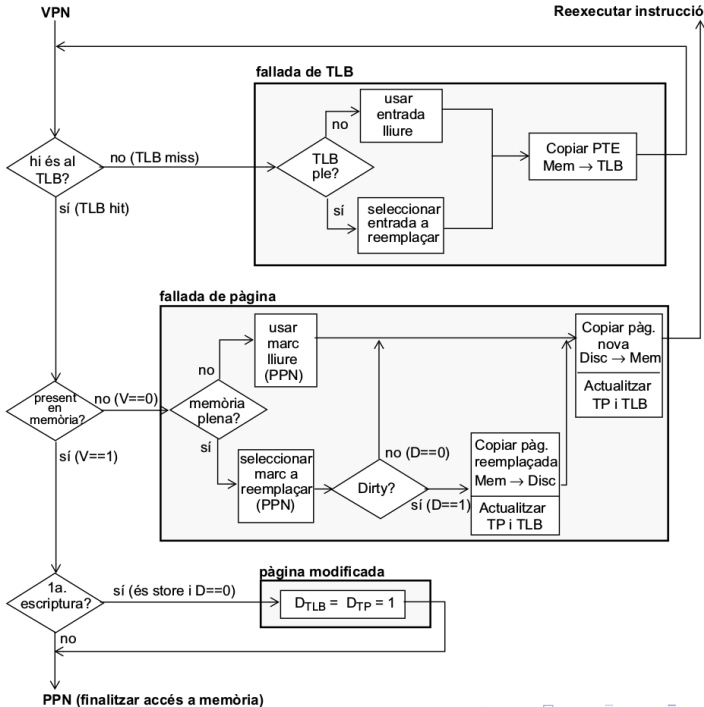


hi és al  
TLB?









## Protección con memoria virtual

- ▶ Cada página física está asignada a un único proceso, y no aparece en las tablas de páginas de ningún otro proceso
  - ▶ El mecanismo de traducción hace imposible que un proceso acceda a páginas físicas de otro proceso

## Protección con memoria virtual

- ▶ Cada página física está asignada a un único proceso, y no aparece en las tablas de páginas de ningún otro proceso
  - ▶ El mecanismo de traducción hace imposible que un proceso acceda a páginas físicas de otro proceso
- ▶ ¿Qué sucede si un proceso trata de modificar su propia tabla de páginas?

## Protección con memoria virtual

- ▶ Cada página física está asignada a un único proceso, y no aparece en las tablas de páginas de ningún otro proceso
  - ▶ El mecanismo de traducción hace imposible que un proceso acceda a páginas físicas de otro proceso
- ▶ ¿Qué sucede si un proceso trata de modificar su propia tabla de páginas?
- ▶ Las tablas de páginas se almacenan en un espacio de direcciones reservado al sistema operativo
  - ▶ En MIPS, direcciones lógicas con bit 31=1



## Protección con memoria virtual

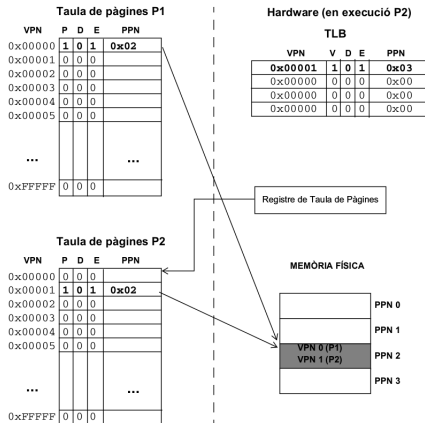
- ▶ Cada página física está asignada a un único proceso, y no aparece en las tablas de páginas de ningún otro proceso
  - ▶ El mecanismo de traducción hace imposible que un proceso acceda a páginas física de otro proceso
- ▶ ¿Qué sucede si un proceso trata de modificar su propia tabla de páginas?
- ▶ Las tablas de páginas se almacenan en un espacio de direcciones reservado al sistema operativo
  - ▶ En MIPS, direcciones lógicas con bit 31=1
- ▶ El procesador tiene dos modos de funcionamiento: usuario y sistema
  - ▶ Solo cuando el procesador está en modo sistema será posible modificar la TLB y las tablas de páginas

## Protección contra escritura

- ▶ Resulta conveniente prohibir la escritura en determinadas páginas
- ▶ Se incluye un bit de permiso de escritura (E) en cada entrada de la tabla de páginas y del TLB
- ▶ Si un proceso trata de escribir en una página con bit  $E=0$ , el sistema operativo aborta la ejecución

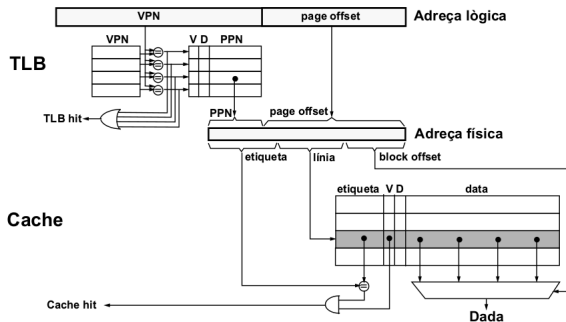
# Compartición de memoria entre procesos

- El sistema operativo puede habilitar la compartición mediante las tablas de páginas



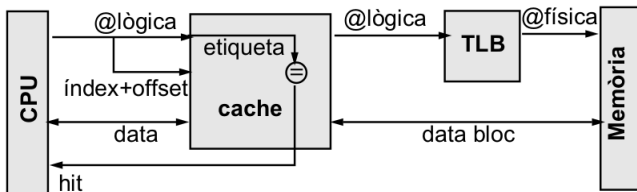
# Integración del TLB y la memoria cache

- ▶ Memoria cache indexada físicamente
  - ▶ Simple pero con tiempo de acceso elevado: se accede secuencialmente a la TLB y a la cache



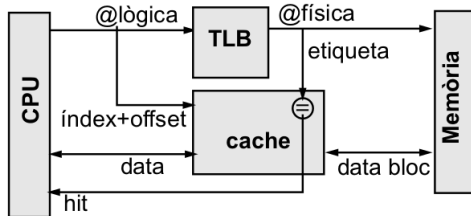
## Integración del TLB y la memoria cache

- ▶ Memoria cache indexada virtualmente
  - ▶ Tiempo de acceso más reducido: no hay que traducir en caso de acierto en la cache
  - ▶ Se pierde el mecanismo de protección



## Integración del TLB y la memoria cache

- ▶ Memoria cache indexada virtualmente y etiquetada físicamente
  - ▶ Se inicia el acceso con la dirección virtual: el índice/conjunto de la cache se obtiene del page offset
  - ▶ En paralelo, se accede a la TLB
  - ▶ Finalmente, se comprueban las etiquetas usando la dirección física



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         |     |     |             |       |       |                  |     |

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 |  |



- 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?

VPNs =