Estructura de Computadores

Tema 6. Memoria Cache

Tiempo de acceso a memoria

- El tiempo para servir un acceso a memoria incluye:
  - t_h: Comprobar etiquetas (hit/miss) y servir la referencia
  - t_p: Penalización debida al acceso a memoria principal
  - t_{am}: Tiempo medio de acceso a memoria

$$\boxed{t_{am} = t_h + t_p}$$

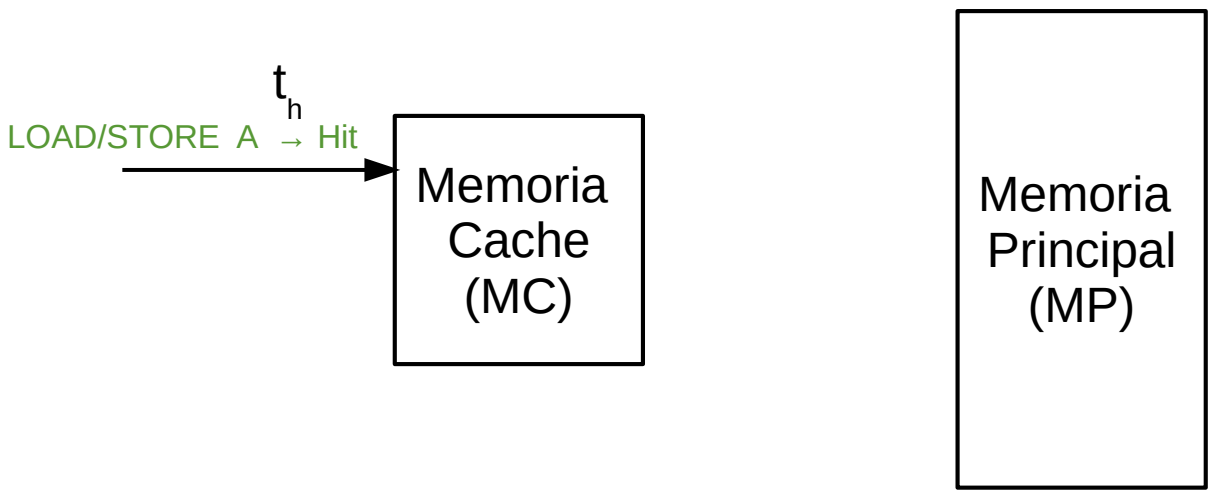
- En caso de escritura inmediata asumimos que:
  - Existe un buffer de escritura de tamaño ilimitado con todas las escrituras pendientes de llevar a MP
  - Ningún acceso posterior entra en conflicto con las escrituras pendientes
  - Las escrituras a MP se realizan en paralelo a la ejecución del procesador

Tiempo de acceso a memoria

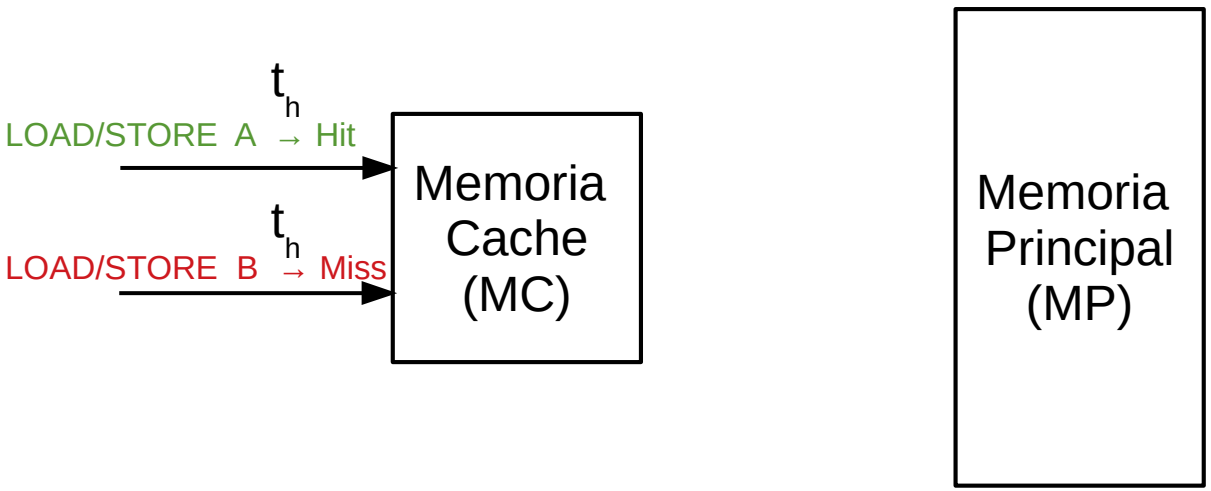
- $t_{am} = t_h + t_p$
- t_p depende de la política de escritura
- t_{bloque}: tiempo para mover el bloque entre MP y MC

t_p	Inmediata con asignación	Inmediata sin asignación	Retardada con asignación
Lectura - Acierto	0	0	0
Lectura - Fallo	$t_{bloque} + t_h$	$t_{bloque} + t_h$	Bloque modificado: $2 * t_{bloque} + t_h$ Bloque no modificado: $t_{bloque} + t_h$
Escritura - Acierto	0	0	0
Escritura - Fallo	$t_{bloque} + t_h$	0	Bloque modificado: $2 * t_{bloque} + t_h$ Bloque no modificado: $t_{bloque} + t_h$

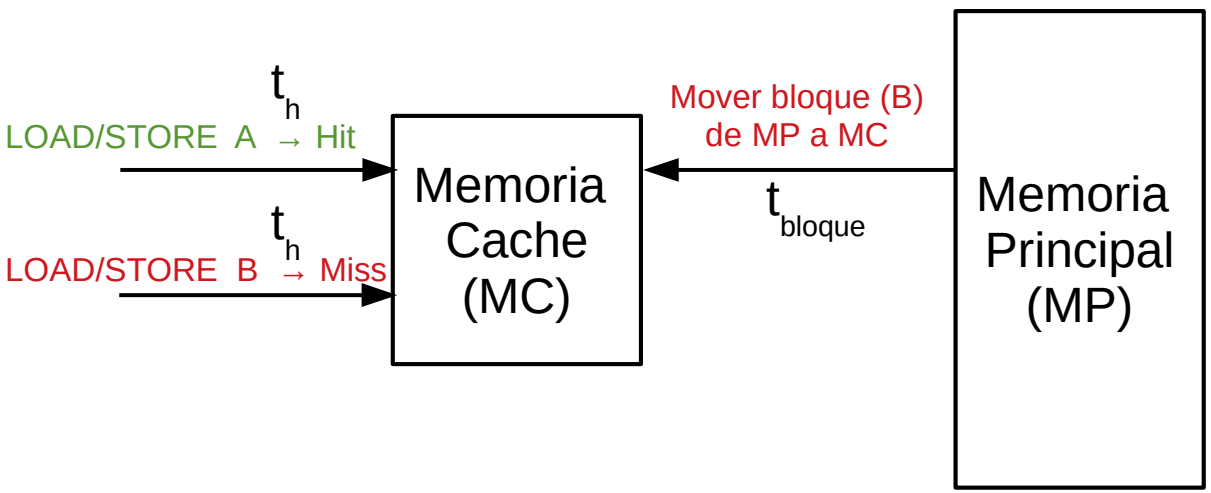
t_{am} - Escritura Inmediata con Asignación



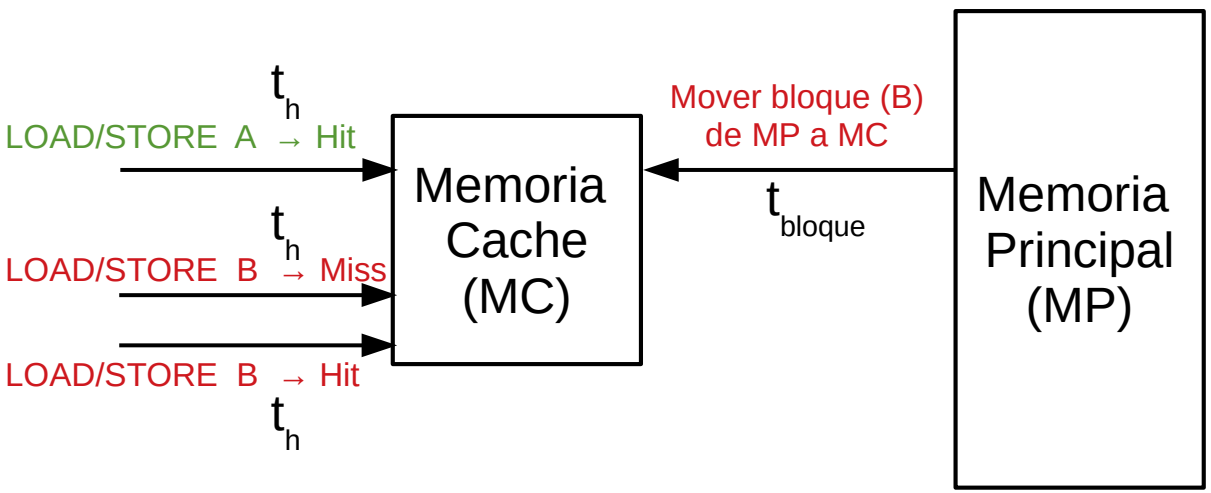
t_{am} - Escritura Inmediata con Asignación



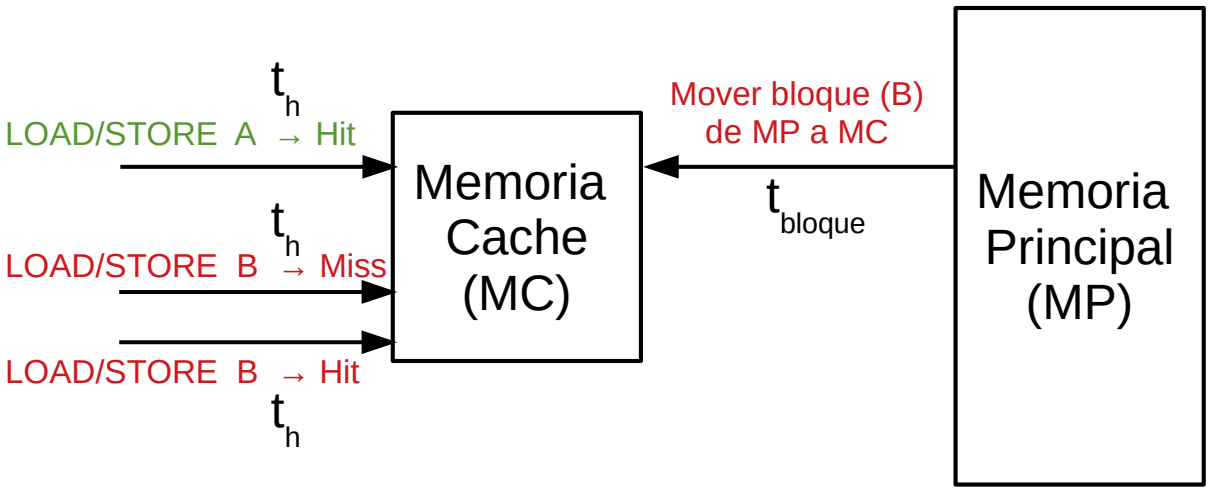
t_{am} - Escritura Inmediata con Asignación



t_{am} - Escritura Inmediata con Asignación

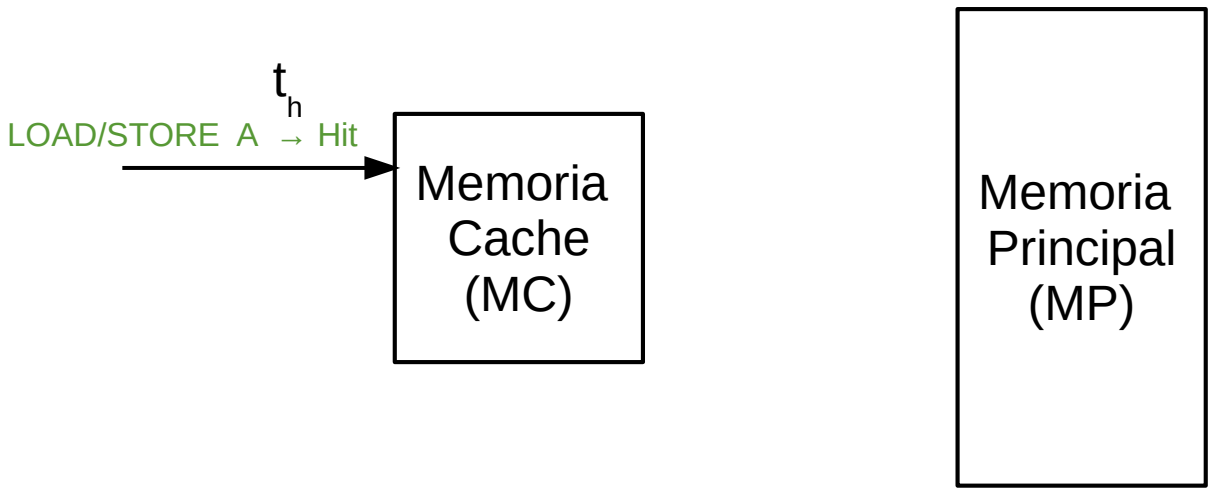


t_{am} - Escritura Inmediata con Asignación

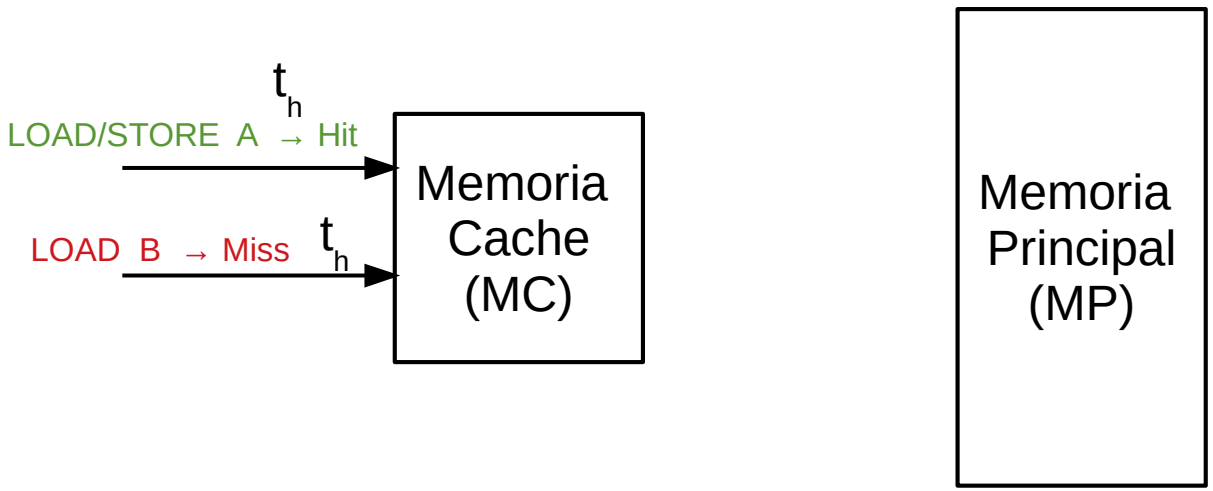


- m: miss ratio (tasa de fallos)
- $t_{am} = t_h + m \times (t_{bloque} + t_h)$

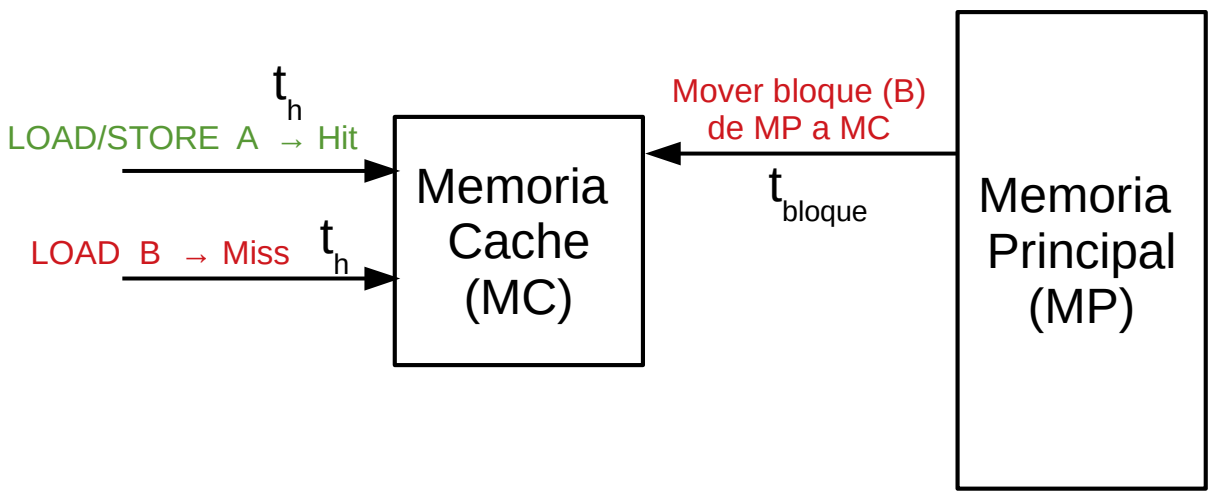
t_{am} - Escritura Inmediata sin Asignación



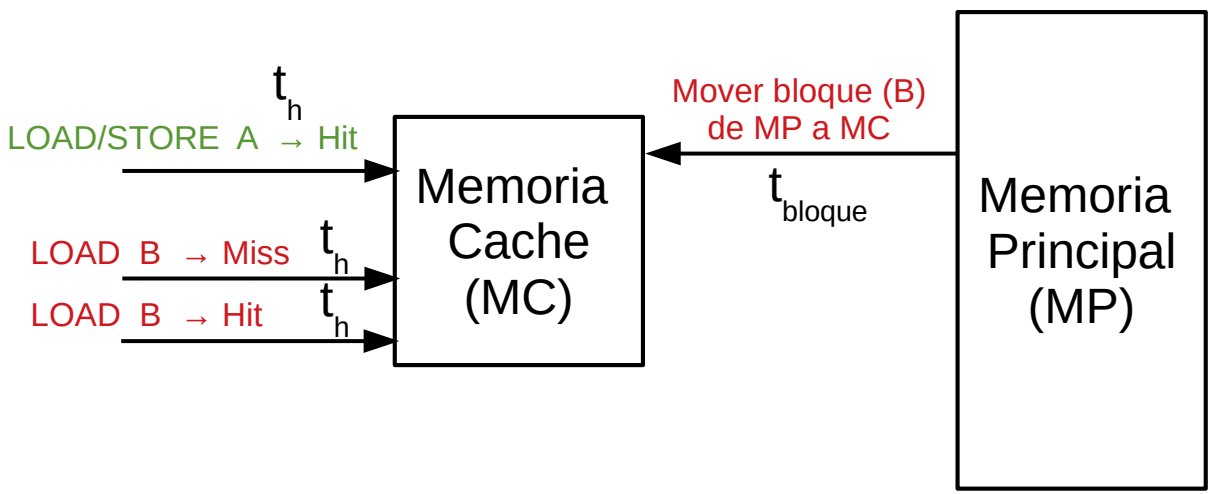
t_{am} - Escritura Inmediata sin Asignación



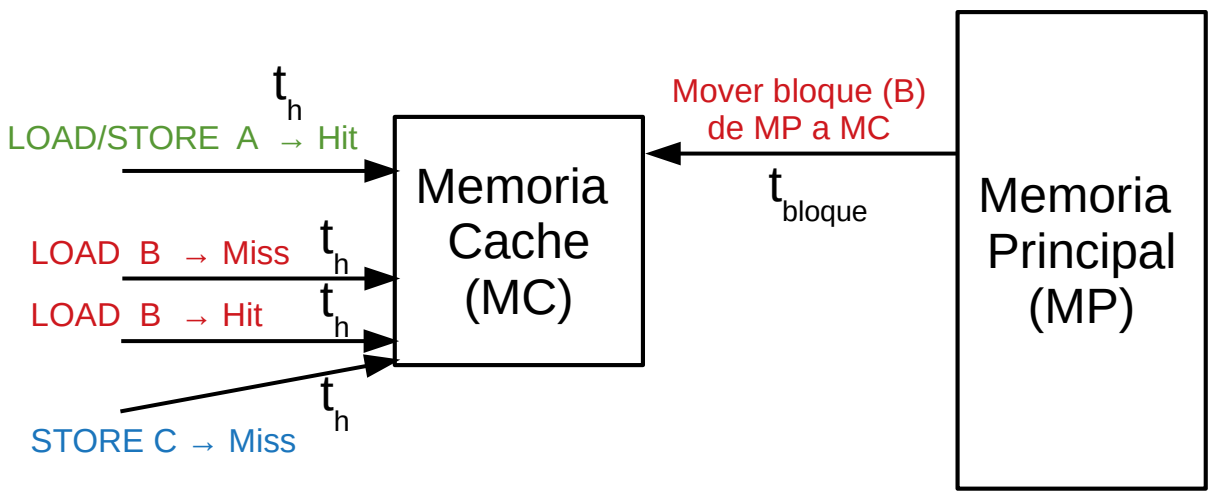
t_{am} - Escritura Inmediata sin Asignación



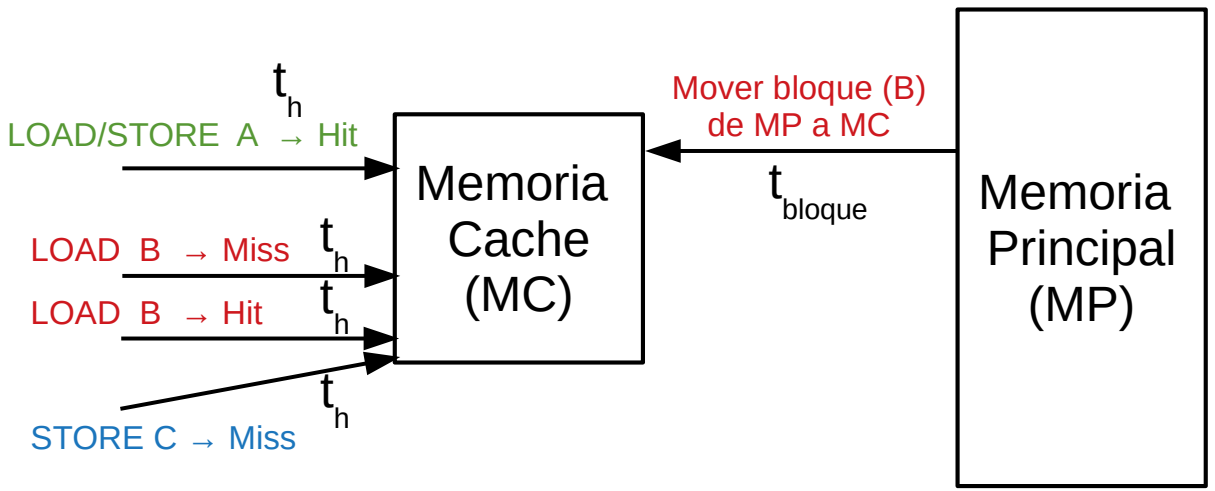
t_{am} - Escritura Inmediata sin Asignación



t_{am} - Escritura Inmediata sin Asignación

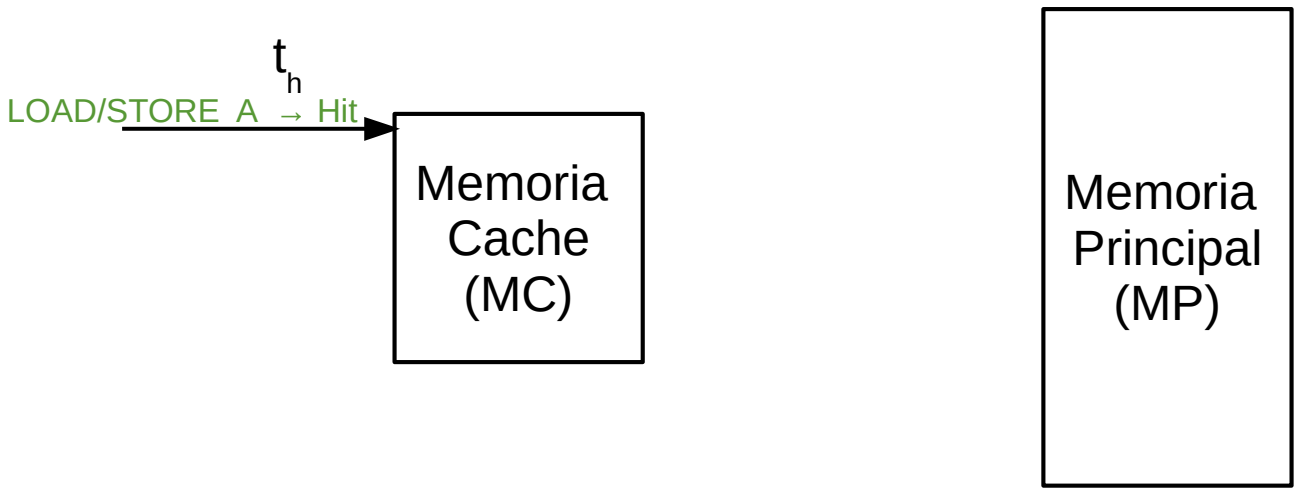


t_{am} - Escritura Inmediata sin Asignación

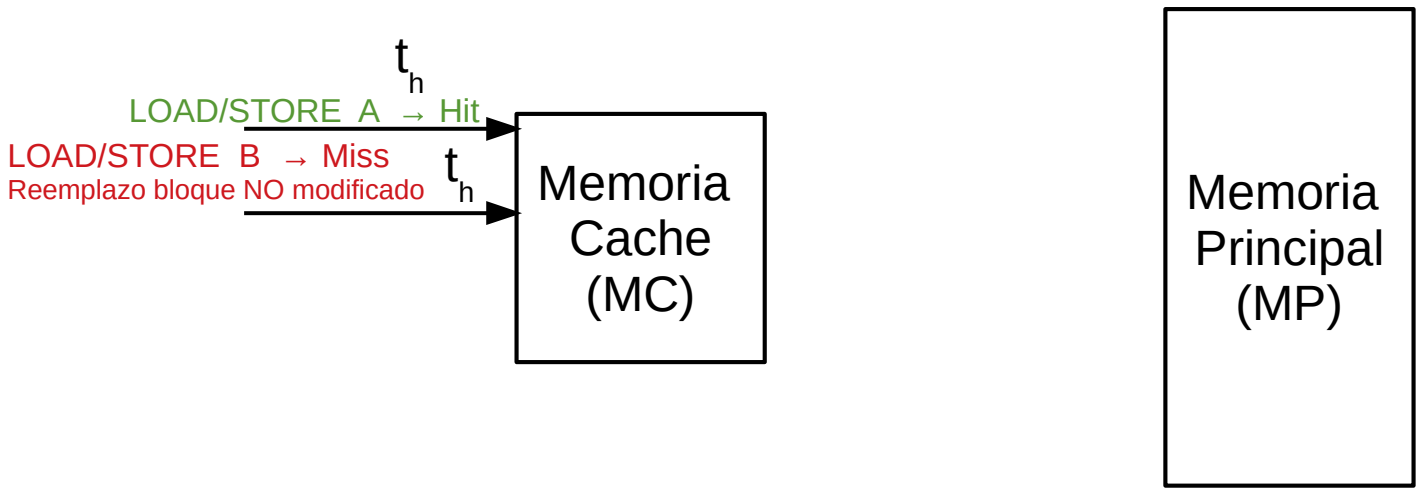


- m: miss ratio (tasa de fallos)
- l: porcentaje de lecturas
- $t_{am} = t_h + m \times l \times (t_{bloque} + t_h)$

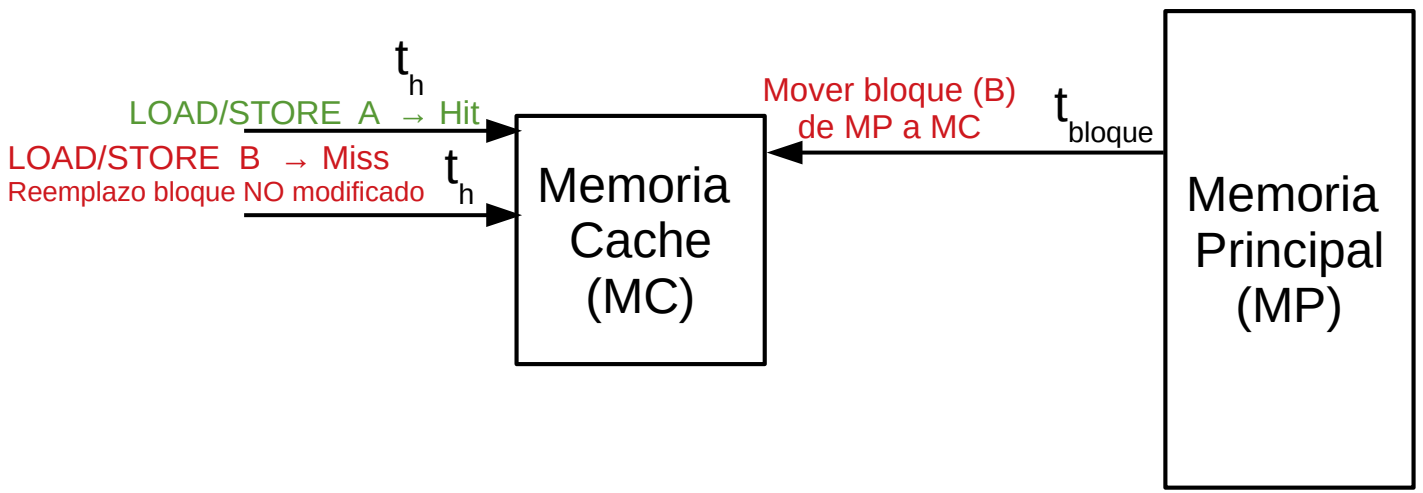
t_{am} - Escritura Retardada con Asignación



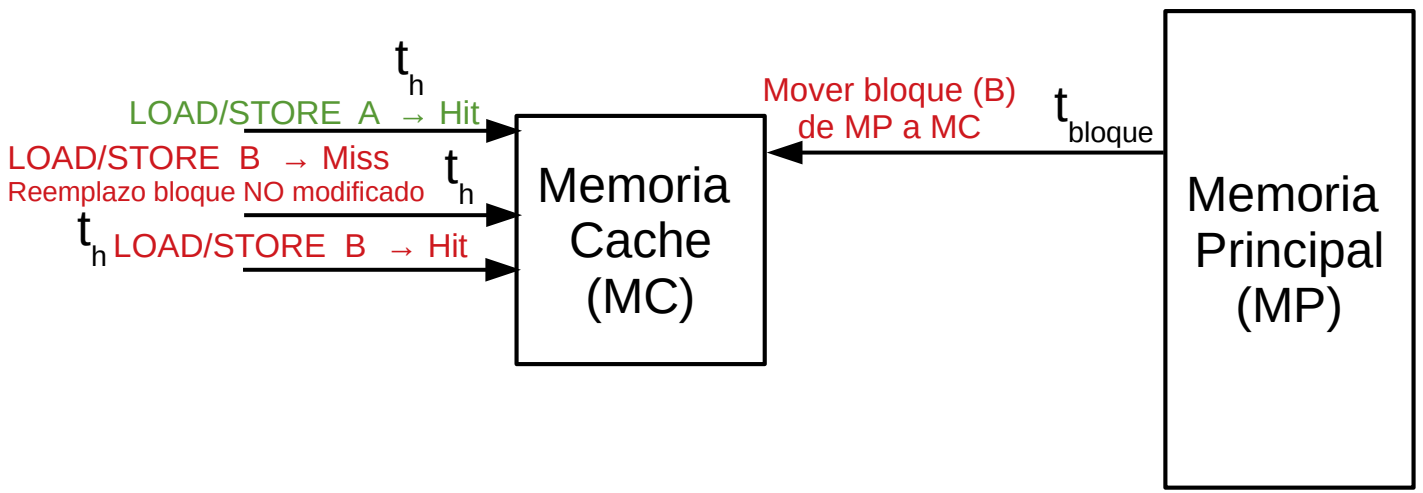
t_{am} - Escritura Retardada con Asignación



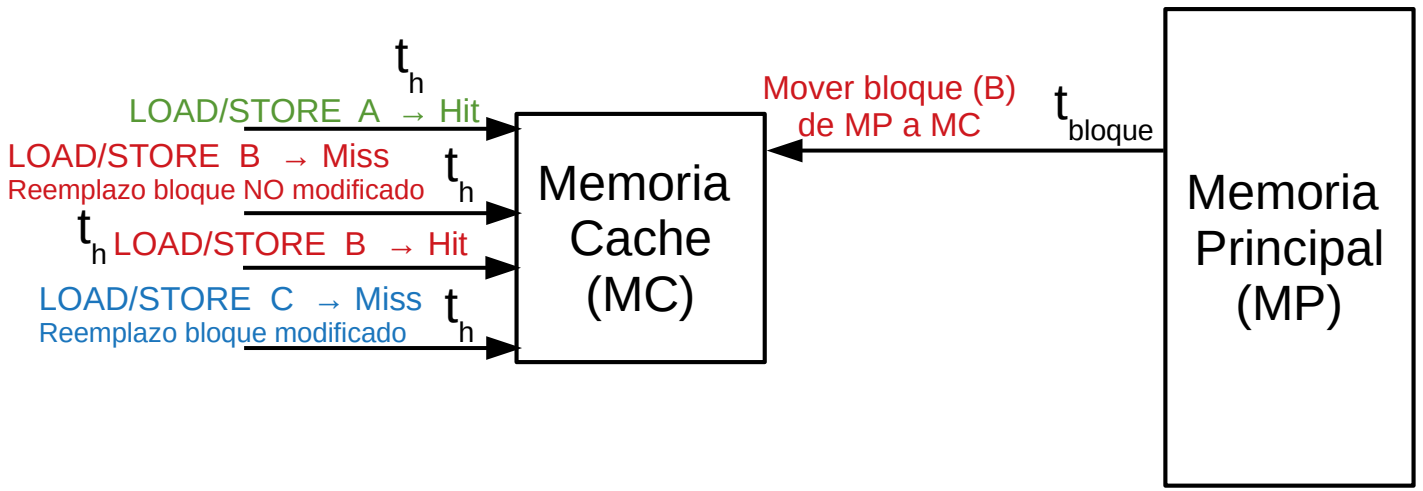
t_{am} - Escritura Retardada con Asignación



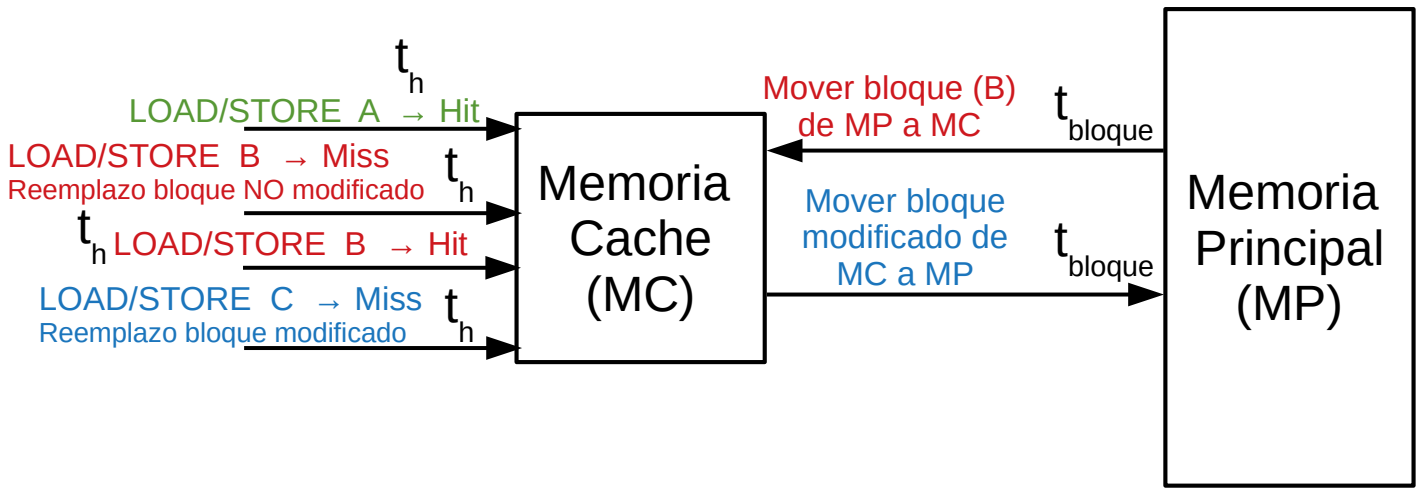
t_{am} - Escritura Retardada con Asignación



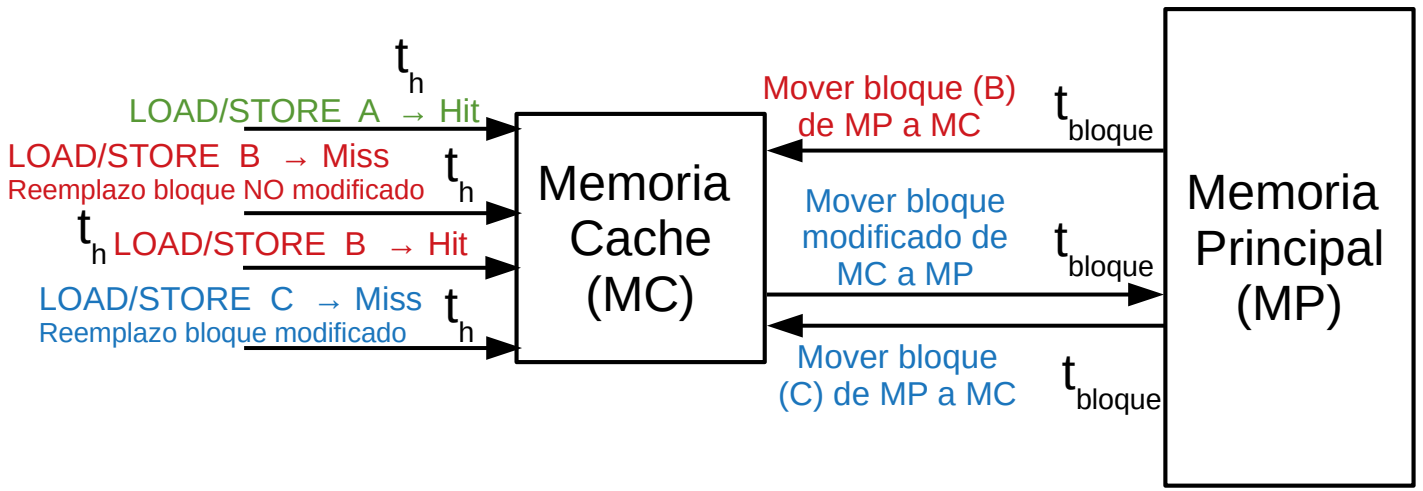
t_{am} - Escritura Retardada con Asignación



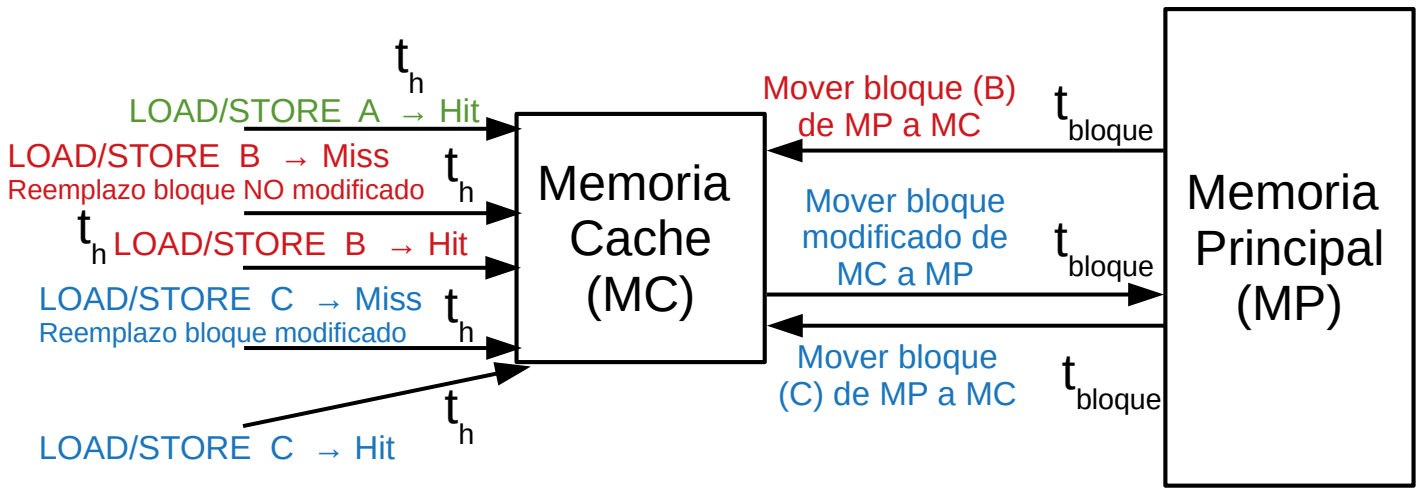
t_{am} - Escritura Retardada con Asignación



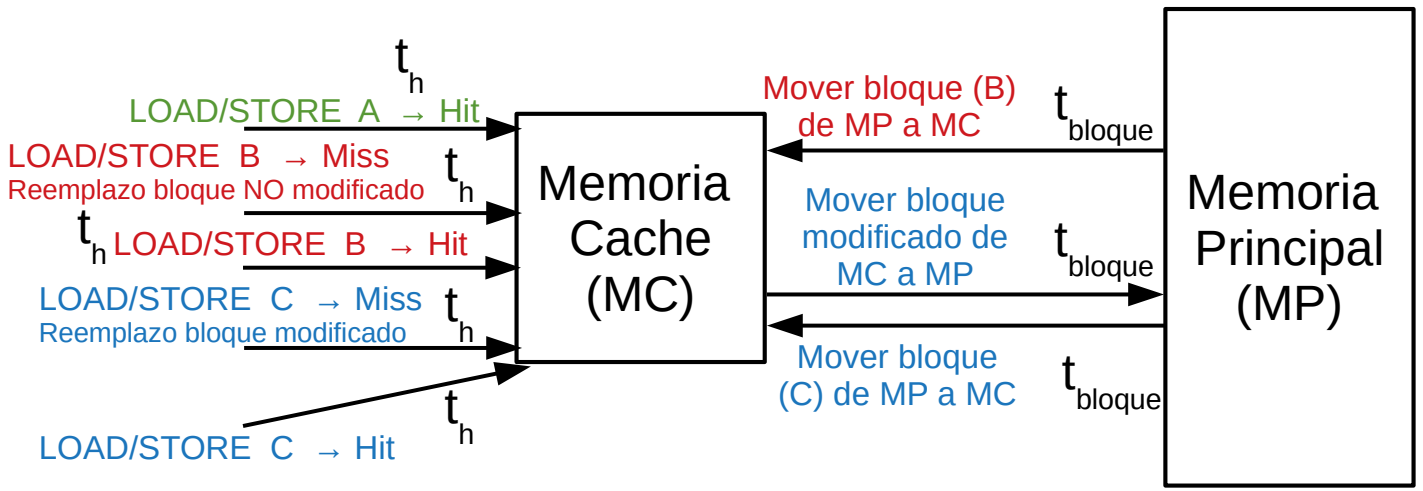
t_{am} - Escritura Retardada con Asignación



t_{am} - Escritura Retardada con Asignación



t_{am} - Escritura Retardada con Asignación



- m: miss ratio (tasa de fallos)
- bnm: porcentaje de reemplazos a bloque no modificado
- bm: porcentaje de reemplazos a bloque modificado
- $t_{am} = t_h + m \times (bnm \times (t_{bloque} + t_h) + bm \times (2 \times t_{bloque} + t_h))$

Es vol definir la política d'escriptura de la memòria cache d'un determinat processador. Es consideren les alternatives: (1) escriptura immediata sense assignació i (2) escriptura retardada amb assignació.

Mitjançant simulació s'han obtingut les següents mesures:

- percentatge d'escriptures (pe): 20%
- percentatge de blocs modificats sobre el total de blocs reemplaçats (pm): 33.33%
- taxa d'encerts cas (1): 0.9
- taxa d'encerts cas (2): 0.85

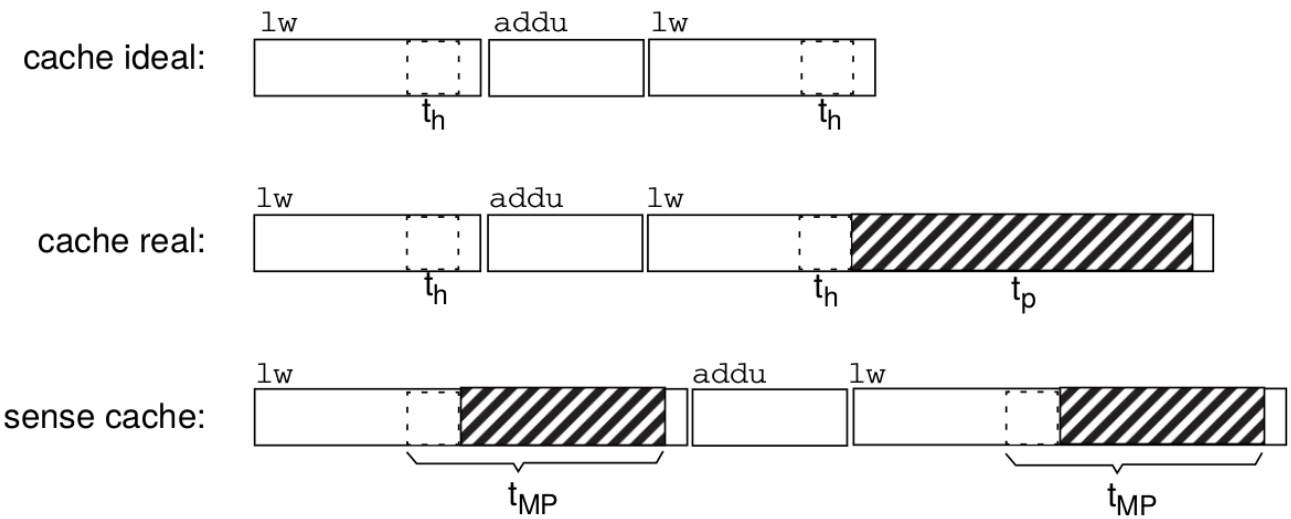
El temps d'accés a memòria cache en cas d'encert (t_h) és de 10 ns. La lectura o escriptura d'un bloc de memòria principal (t_{block}) requereix 100 ns.

Es demana:

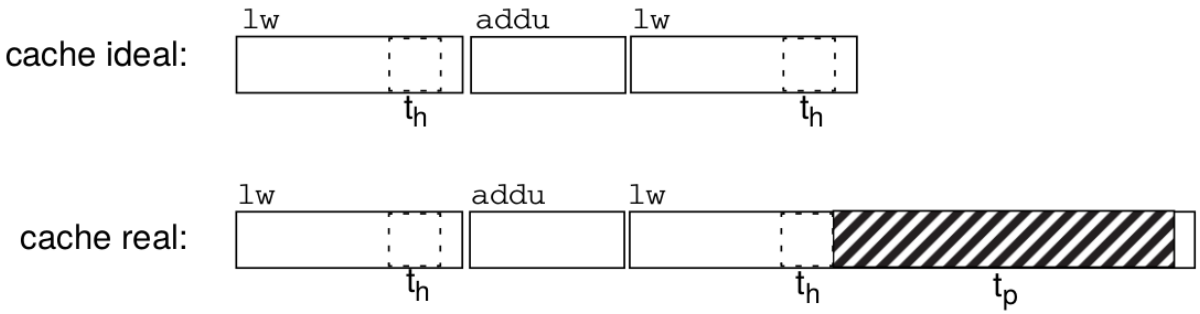
a) Calculeu el temps mitjà d'accés a memòria (t_{am}) en ambdues alternatives.

Número de ciclos

- $n_{ciclos} = n_{ins} \cdot CPI$
  - n_{ins}: Número de instrucciones ejecutadas
  - CPI: CPI promedio
- El CPI varía en función del número de fallos de cache



CPI



- n_{ciclos_ideal}: Tiempo que tarda la ejecución ideal
- n_{ciclos_penal}: Total de ciclos de penalización extra

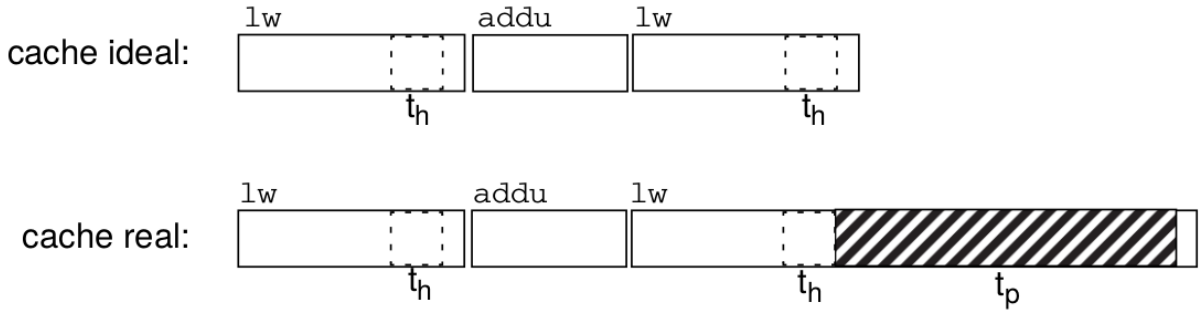
$$CPI_{ideal} = n_{ciclos\_ideal}/n_{ins}$$

$$CPI_{real} = (n_{ciclos\_ideal} + n_{ciclos\_penal})/n_{ins}$$

$$CPI_{real} = CPI_{ideal} + (n_{ciclos\_penal}/n_{ins})$$

$$CPI_{real} = CPI_{ideal} + (n_{fallos} \cdot t_p)/n_{ins}$$

Tiempo de ejecución

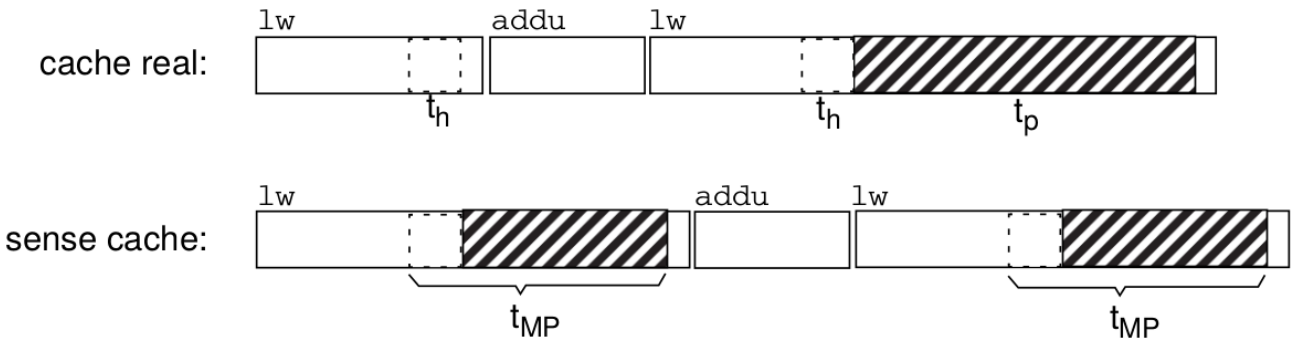


$$CPI_{real} = CPI_{ideal} + (n_{fallos} \cdot t_p)/n_{ins}$$

$$t_{eje} = t_c \cdot n_{ins} \cdot CPI_{real}$$

$$\boxed{t_{eje} = t_c \cdot (n_{ins} \cdot CPI_{ideal} + n_{fallos} \cdot t_p)}$$

Tiempo de ejecución

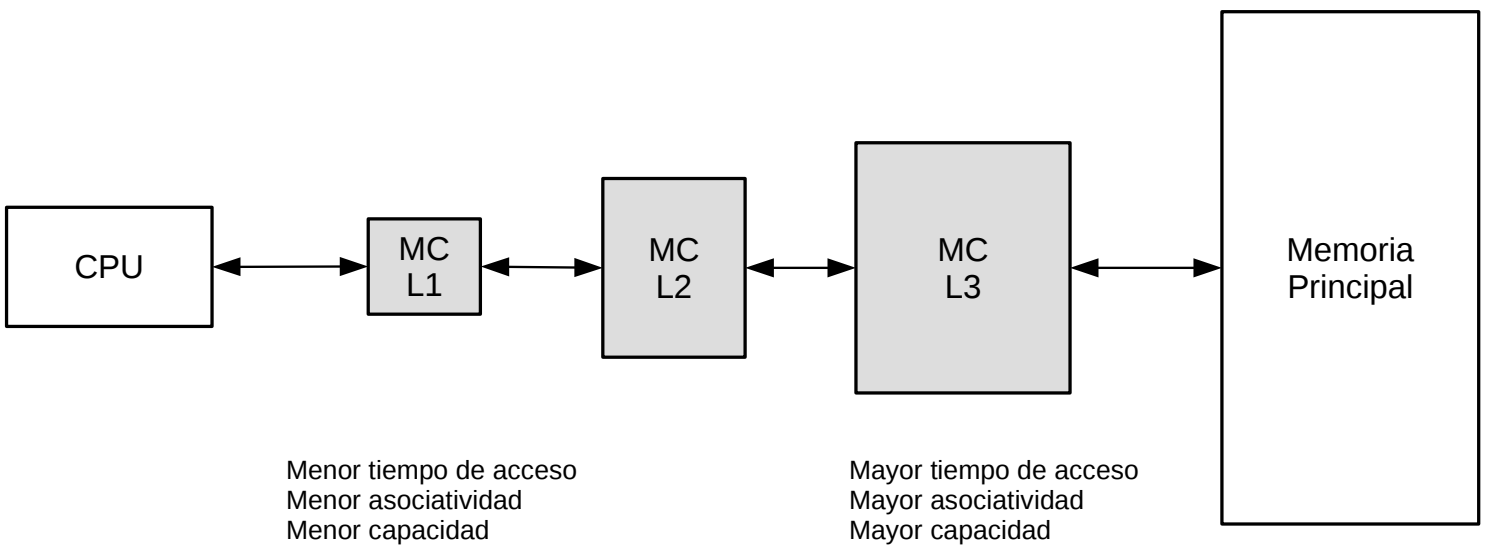


$$t_{eje} = t_c \cdot (n_{ins} \cdot CPI_{ideal} + n_{fallos} \cdot t_p)$$

- La misma ecuación se puede aplicar a un sistema sin memoria cache:
  - n_{fallos} es igual al número total de loads
  - El tiempo de penalización es $t_p = t_{MP} - t_h$
  - t_{MP}: tiempo de acceso a memoria principal
  - Todos los `lw` tardan t_{MP}, pero el CPI ideal ya incluye t_h

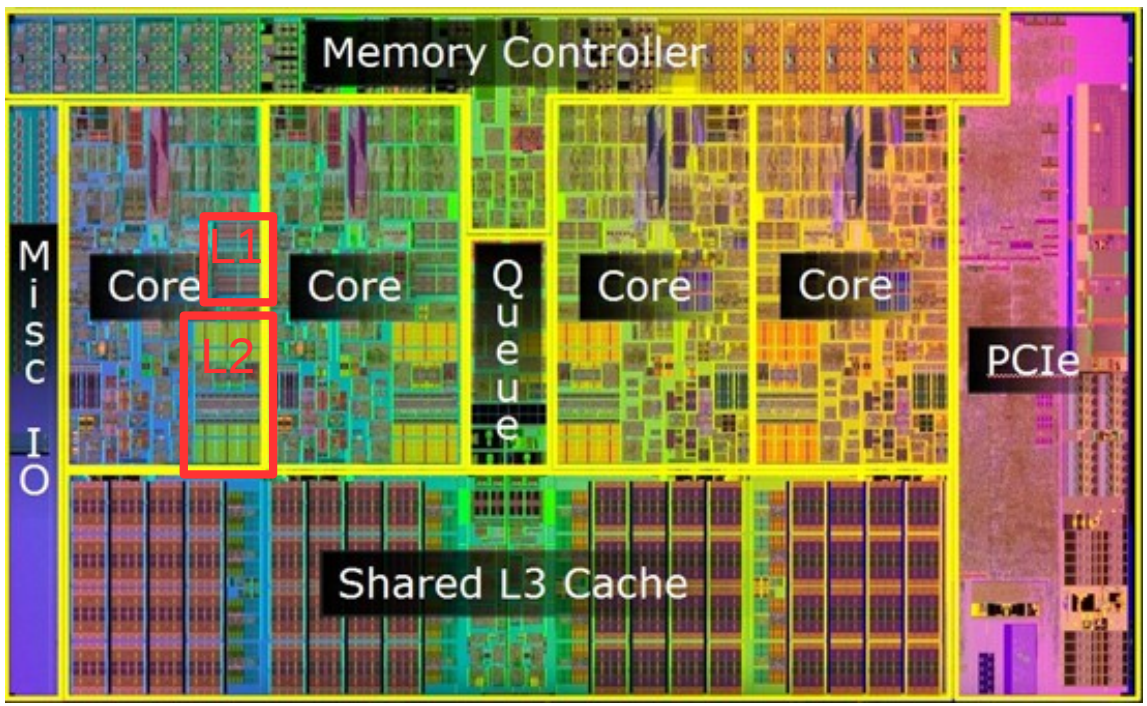
Múltiples niveles de memoria cache

- Los procesadores comerciales incluyen una jerarquía de memoria con múltiples niveles de memoria cache
- Ayuda a reducir el tiempo de acceso a memoria
  - Ej: En caso de fallo en L1, podemos encontrar el bloque en L2, sin tener que acceder a memoria principal



Intel i5-750 (Nehalem)

- L1 de 32 KB (por core)
- L2 de 256 KB (por core)
- L3 de 8 MB (compartida)



http://www.pcstats.com/articleview.cfm?articleid=2581&page=4

Un sistema disposa d'un processador de 20 bits d'adreces, i una memòria cache (MC) de 8 Kbytes amb la següent organització:

- Correspondència associativa per conjunts, de grau 2 (2 blocs per conjunt)
- Blocs de 256 bytes
- Reemplaçament LRU
- Escriptura retardada amb assignació.

Estant la cache inicialment buida, un programa produeix una seqüència de referències a memòria segons s'indica a la següent taula, on apareixen les adreces en hexadecimal i si són lectures o escriptures (L/E). Completa les columnes que falten indicant, per a cada referència: el número de conjunt de MC; si és encert (e) o fallada (f); i el nombre de bytes de Memòria Principal (MP) llegits i/o escrits.

L/E	adreça (hex)	núm. de conjunt	encert (e)/ fallada (f)	bytes de MP	
				llegits	escrits
L	01200				
E	082A0				
L	083F0				
E	01204				
L	04204				
E	083F4				
L	01208				
E	082A0				

Suposem que tenim un processador de 32 bits amb una memòria cache de dades de 512 bytes, on cada bloc té 16 bytes. Suposem que executem els següents programes.

```
//programa A

int M[4][128];

void main() {
int i, j; //en registres
  for (i=0;i<4;i++)
    for (j=0;j<128;j++)
      M[i][j]= 0;
}
```

```
//programa B

int M[4][128];

void main() {
int i, j; //en registres
  for (j=0;j<128;j++)
    for (i=0;i<4;i++)
      M[i][j]= M[i][j]+1;
}
```

Calcula el nombre de fallades de la cache suposant que la memòria cache és inicialment buida. L'adreça base de la matriu `M` és 0.

a) Suposant que la cache és de correspondència directa i té la política d'escriptura retardada amb assignació.

Fallades A = ____ Fallades B = ____

b) Suposant que la cache és associativa per conjunts de 4 vies (algorisme de reemplaçament LRU), i que té la política d'escriptura immediata sense assignació.

Fallades A = ____ Fallades B = ____

¿Verdadero o falso?

1. En una memòria cache amb política d'escriptura immediata sense assignació, un accés a la memòria cache pot implicar dos accesos a memòria principal.

¿Verdadero o falso?

1. En una memòria cache amb política d'escriptura immediata sense assignació, un accés a la memòria cache pot implicar dos accesos a memòria principal.
2. Si en una cache canviem la política d'escriptura immediata amb assignació a retardada amb assignació, sense cap més canvi, el nombre total de fallades no canvia

¿Verdadero o falso?

1. En una memòria cache amb política d'escriptura immediata sense assignació, un accés a la memòria cache pot implicar dos accesos a memòria principal.
2. Si en una cache canviem la política d'escriptura immediata amb assignació a retardada amb assignació, sense cap més canvi, el nombre total de fallades no canvia
3. En un processador amb adreces de 32 bits, una cache associativa de 4 vies, de 32KB i blocs de 32 bytes, s'han de dedicar 19 bits a etiqueta (TAG), 8 al número d'entrada (conjunt) i 5 al desplaçament.

¿Verdadero o falso?

1. Un sistema que tingués una memòria cache més gran que la memòria principal mai tindria fallades.

¿Verdadero o falso?

1. Un sistema que tingués una memòria cache més gran que la memòria principal mai tindria fallades.
2. En les memòries cache que utilitzen escriptura immediata s'ha de posar el dirty bit a 1 només quan hi ha un encert d'escriptura.

¿Verdadero o falso?

1. Un sistema que tingués una memòria cache més gran que la memòria principal mai tindria fallades.
2. En les memòries cache que utilitzen escriptura immediata s'ha de posar el dirty bit a 1 només quan hi ha un encert d'escriptura.
3. El temps de penalització en cas de fallada d'escriptura d'una memòria cache d'escriptura immediata sense assignació és zero segons el model estudiat.

¿Verdadero o falso?

1. Un sistema que tingués una memòria cache més gran que la memòria principal mai tindria fallades.
2. En les memòries cache que utilitzen escriptura immediata s'ha de posar el dirty bit a 1 només quan hi ha un encert d'escriptura.
3. El temps de penalització en cas de fallada d'escriptura d'una memòria cache d'escriptura immediata sense assignació és zero segons el model estudiat.
4. Si s'executa un mateix programa en dues memòries cache diferents, però les dues de correspondència directa, sempre tindrà una major taxa d'encerts la que tingui els blocs de mida més gran.