

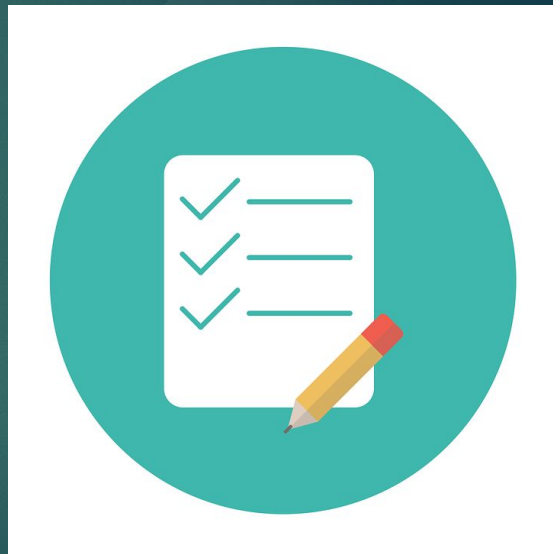


UNIVERSIDAD
DE SANTIAGO
DE CHILE

Ayudantía 4: Caché parte 1

Índice

- ❑ **Conceptos**
- ❑ **Hit-Miss en caché**
- ❑ **Rendimiento**
- ❑ **Direccionamiento**
- ❑ **Maapeo directo**

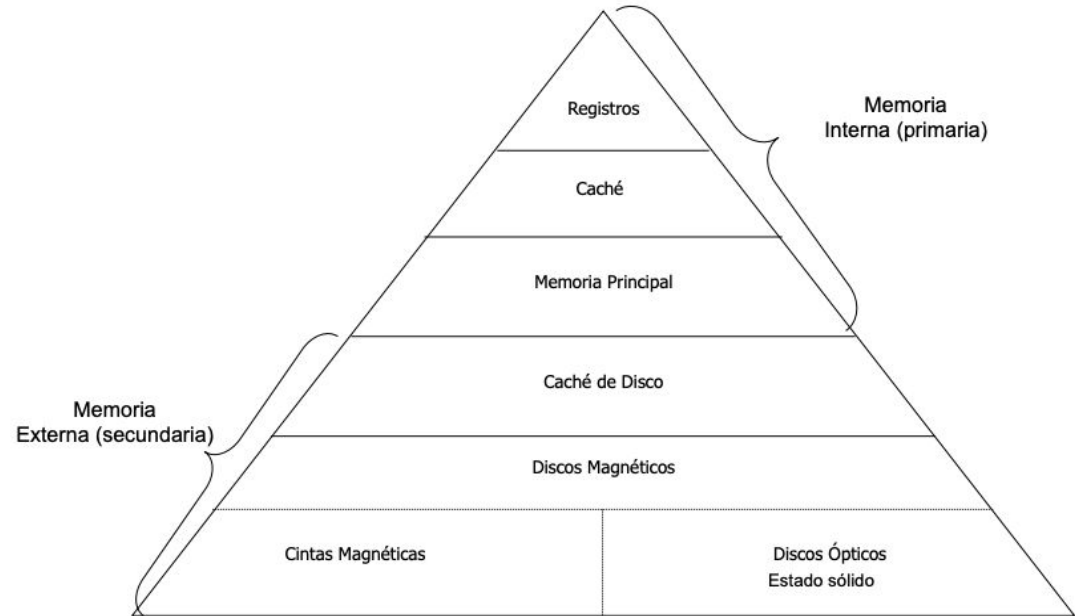


Niveles de memoria

Memoria principal: RAM

Memoria secundaria: HDD, SSD

L1, L2, L3,.. : caché



Implementación caché

Una línea o entrada de caché contiene:

Index	V	Tag	Data
000	Y	10	Mem[10000]
001	N		
010	Y	11	Mem[11010]
011	Y	00	Mem[00011]
100	N		
101	N		
110	Y	10	Mem[10110]
111	N		

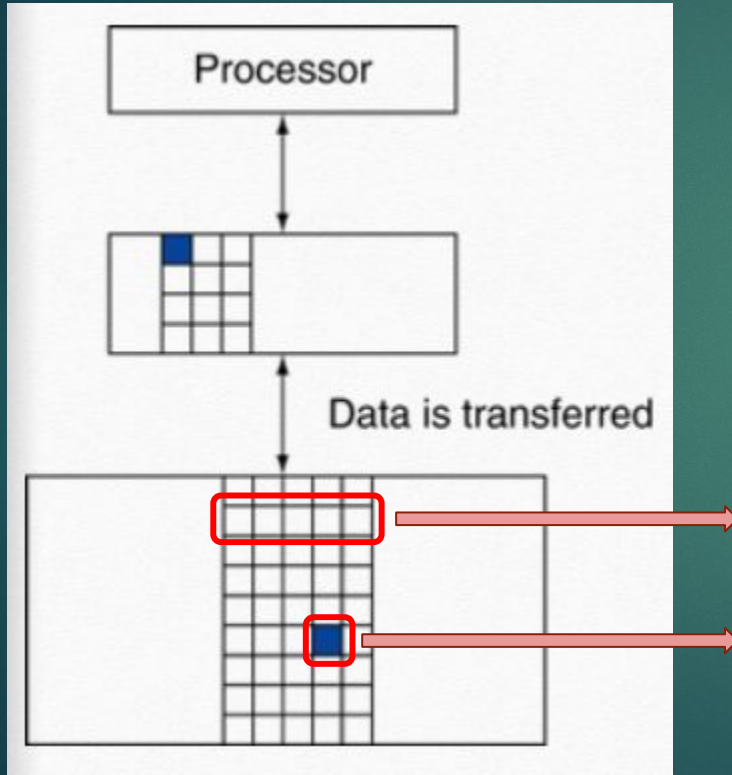
V: valid bit

Tag: etiqueta de identificación

Data: datos cargados desde memoria

línea de caché
(entrada)

bit (0 ó 1) **!= byte** (8 bits)



Hit: acierto en un **bloque**

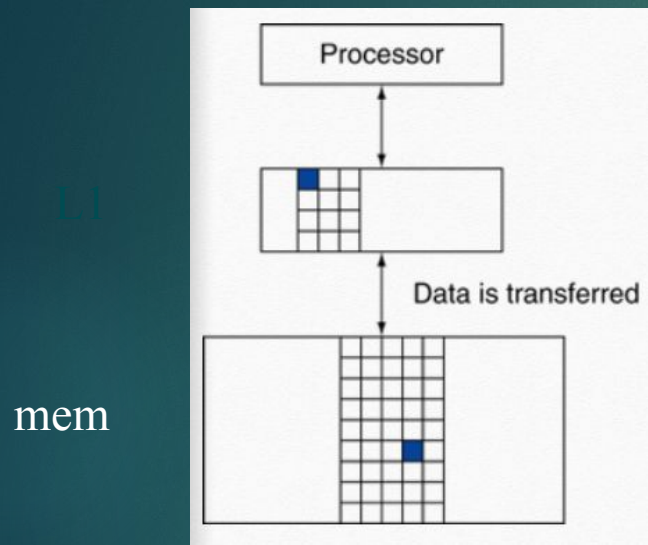
Miss: falla en un **bloque**

Rate: tasa (%) de probabilidad de
ocurrencia

bloque = conjunto de palabras

word = palabra = 4 bytes (default)

Rendimiento y multinivel



cpi base = cpi ejecución

Las probabilidades simultáneas se multiplican
(se cumplen ambas condiciones)

$$\text{AMAT} = \text{cpi base} * (1 - \% \text{mem access}) +$$
$$(\text{hit rate} * \text{hit penalty} + \text{miss rate} * \text{miss penalty}) * \% \text{mem access}$$

Ejemplo rendimiento en caché

Considere un procesador con 1 nivel de caché (L1) que tiene un **CPI base** de **1.5** (sin considerar accesos a memoria). La **frecuencia** de accesos a **memoria** es de **30%**. La **tasa de acierto** para la caché es de **80%** y tiene un tiempo de acceso de **2 [ciclos]**. El tiempo de **acceso a memoria** principal es de **300 [ciclos]**.

Calcule el CPI para este procesador.

Ejemplo rendimiento en caché

$$\text{CPI} = \text{cpi base} * (1 - \% \text{mem access}) + (\text{hit rate} * \text{hit penalty} + \text{miss rate} * \text{miss penalty}) * \% \text{mem access}$$

Datos:

cpi base = 1.5


mem access = 30% = 0,3

hit rate L1 = 80% = 0,8


hit penalty = 2 [ciclos]

hit penalty mem = 300 [ciclos]


$$\text{CPI} = 1.5 * 0,7 + (0,8 * 2 + 0,2 * 300) * 0,3 = \mathbf{19.53 \text{ [ciclos]}}$$



instrucciones
sin acceso a
memoria



si el dato no
está en caché,
se encuentra en
memoria



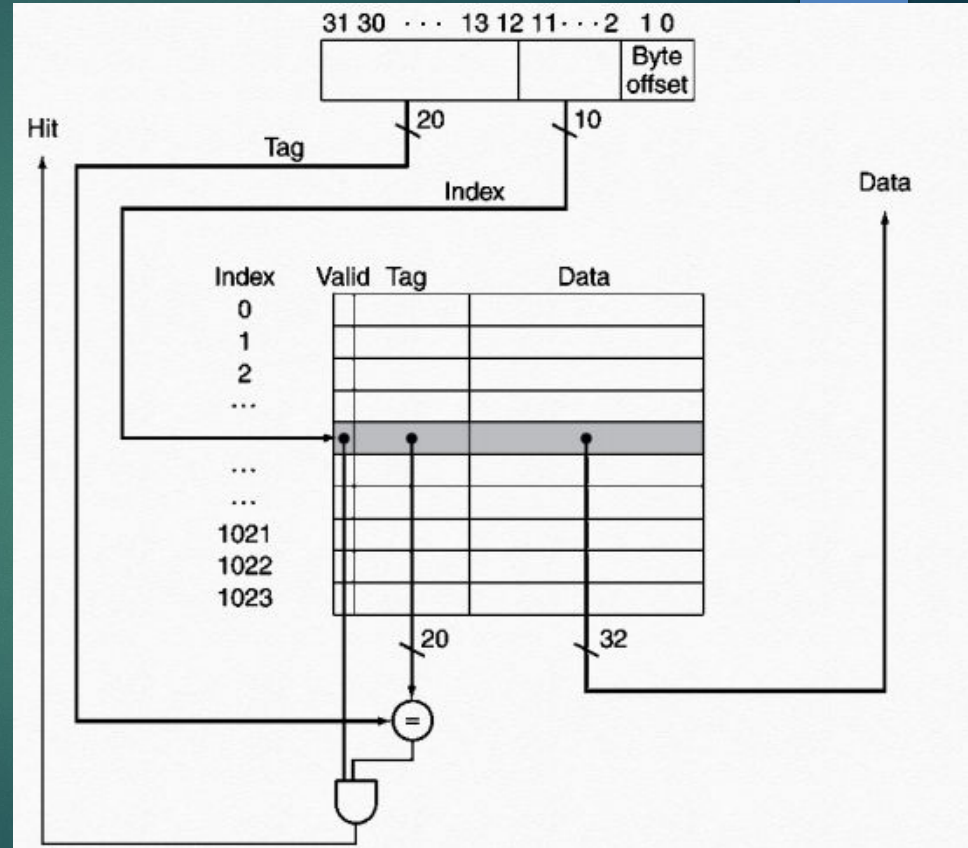
instrucciones
con acceso a
memoria

Direccionamiento

Los bits de direccionamiento están determinados por el tamaño **máximo** de la memoria principal (**RAM**)

TiB	2^{40}
GiB	2^{30}
MiB	2^{20}
KiB	2^{10}
-	2^0

El exponente determina la cantidad de bits.



Ejemplo direccionamiento

Suponga un procesador que está en una placa que soporta un máximo de **8 GiB de memoria RAM**. El procesador tiene sólo 1 nivel de **caché** con un tamaño de **64 KiB**. Además cada **bloque** en memoria almacena **8 palabras de 32 bits**.

Muestre los **bits de la dirección** de memoria usados para acceder a la caché. Indique la **cantidad de entradas** que tiene la caché y la **cantidad de palabras** que puede almacenar la caché. Indique la cantidad de **bits** necesarios para **implementar** la caché

Ejemplo direccionamiento

Tamaño RAM = $2^3 * 2^{30} = 2^{33}$ [bytes]

Tamaño caché = $2^6 * 2^{10} = 2^{16}$ [bytes]

Tamaño word = $32/8 = 4 = 2^2$ [bytes]

Tamaño bloque = $8 * 2^2 = 2^5$ [bytes]

Cantidad de bits necesarios para
direccionar dentro de un bloque
(incluyendo las palabras)

Cantidad de bits
necesarios para
direccionar la caché

Cantidad de bits
necesarios para
direccionar dentro de
una palabra

cantidad de bloques en
caché:

$$2^{16} / 2^5 = 2^{11}$$

Cantidad de bits
necesarios para
direccionar los
bloques dentro de la
caché

32 - 16 15 - 5 4 - 2 1 - 0



Ejemplo direccionamiento

VB	Tag	Data							
1	10110110011010000	34	12	67	98	12	90	121	54
1	10110110011010111	56	80	123	567	4	12	65	67
1	10000110011011010	78	121	465	797	65	79	09	12
0	10000110011010111	0	0	0	0	0	0	0	0
...
1	10110001011011101	456	5	7	23	465	68	12	3

1 bit

17 bits

4 bytes

4 bytes

4 bytes

4 bytes

...

0
1
2
3
...
2ⁿ⁻¹
n

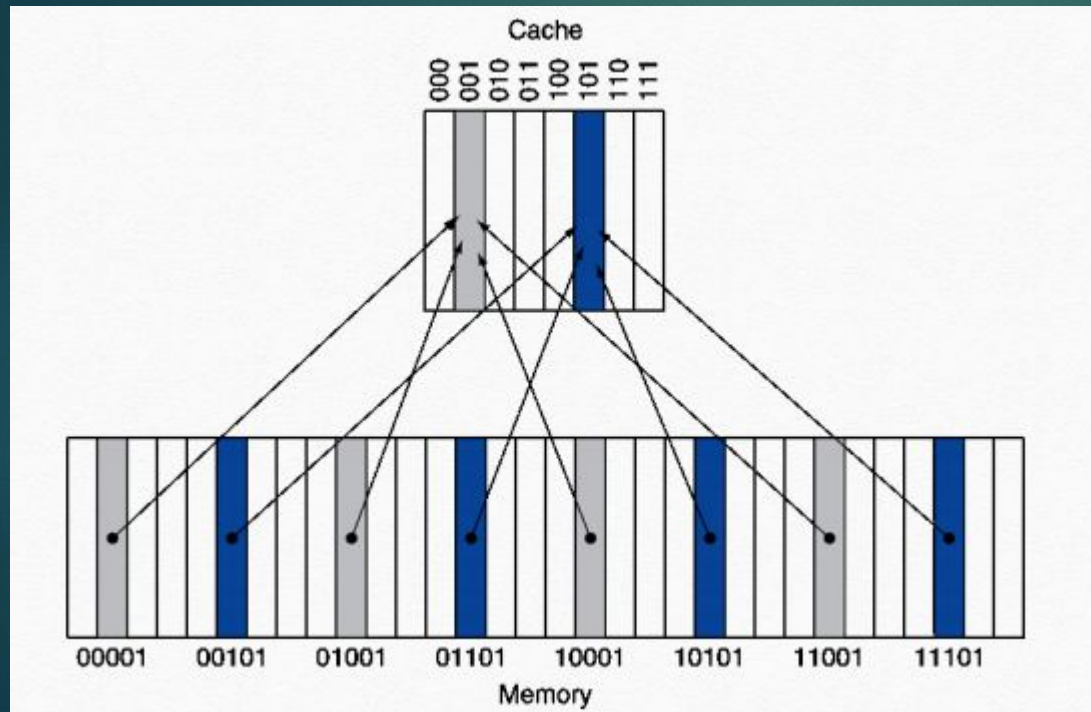
Mapeo directo

$$\text{IdBloq} = \text{bloqMem} \% \# \text{bloqC}$$

13

bloque en caché

0 1 2 3 4 5 6 7 8



$$\text{bloqMem} = [\text{addr} / \text{tamañoBloq}]$$

$$1 \% 8 = 1$$

Parte entera

$$5 \% 8 = 5$$

$$9 \% 8 = 1$$

$$13 \% 8 = 5$$

$$\text{bloqMem} = \text{addr}$$

$$17 \% 8 = 1$$

$$\# \text{bloqC} = 8$$

$$21 \% 8 = 5$$

$$25 \% 8 = 1$$

$$\text{idBloq} = \text{addr} \% 8$$

$$29 \% 8 = 5$$

Mapeo directo

$$\text{idBloq} = [\text{addr}/3] \% 2$$

bloqMem:

0	1	2	3	4	5	6	7	8	9	10	11	12	
13													
0	3	6	9	12	15	18	21	24	27	30	33	36	39
1	4	7	10	13	16	19	22	25	28	31	34	37	40
2	5	8	11	14	17	20	23	26	29	32	35	38	41

7 11 12 13 23

tamañoBloq = 3 palabras

#bloqC = 2

B0	w0	6	6	12	12	12
	w1	7	7	13	13	13
	w2	8	8	14	14	14
B1	w0		9	9	9	21
	w1		10	10	10	22
	w2		11	11	11	23

Referencias a memoria:

7, 11, 12, 13, 23

Ejemplo mapeo directo

Considere la siguiente secuencia de referencias a memoria, entregadas como direcciones de *palabra*:

16, 88, 18, 16, 28, 19, 89, 30, 88, 70, 68, 12

Muestre el contenido final de una caché con mapeo directo con **2 bloques** de **4 palabras**, especificando aciertos (*hit*) y desaciertos (*miss*). Asuma que la caché se encuentra **inicialmente vacía**.

Mapeo directo

$$\text{bloqMem} = [\text{addr}/4]$$

0	1	2	3	4	5	6	7	8	9	10	11	12	
0	4	8	12	16	20	24	28	32	36	40	44	48	52
1	5	9	13	17	21	25	29	33	37	41	45	49	53
2	6	10	14	18	22	26	30	34	38	42	46	50	54
3	7	11	15	19	23	27	31	35	39	43	47	51	55
14	15	16	17	18	19	20	21	22	23	24	25	26	27
56	60	64	68	72	76	80	84	88	92	96	100	104	108
57	61	65	69	73	77	81	85	89	93	97	101	105	109
58	62	66	70	74	78	82	86	90	94	98	102	106	110
59	63	67	71	75	79	83	87	91	95	99	103	107	111

Ejemplo mapeo directo

idBloq = [addr/4] % 2

16 88 18 16 28 19 89 30 88 70 68 12

B0	w0	16	88	16	16	16	16	88	88	88	88	88	88	
	w1	17	89	17	17	17	17	89	89	89	89	89	89	
	w2	18	90	18	18	18	18	90	90	90	90	90	90	
	w3	19	91	19	19	19	19	91	91	91	91	91	91	
B1	w0					28	28	28	28	28	68	68	12	
	w1					29	29	29	29	29	69	69	13	
	w2					30	30	30	30	30	70	70	14	
	w3					31	31	31	31	31	71	71	15	