

## 2.6. La memòria d'un ordinador és adreçable per bytes, i la longitud de la paraula és de 32 bits. Un programa consisteix en dos bucles nidats (nested-loops): un petit bucle interior i un bucle exterior molt més gran.

L'estructura general del programa es mostra a la figura 1. Les adreces de memòria decimal mostren la localització dels dos bucles i el principi i final del programa total.

Totes les ubicacions de memòria a les diverses seccions del programa, 8-52, 56-136, 140-240, etc., contenen instruccions per executar-les en seqüències. El programa s'executarà en una computadora que tingui una memòria cau d'instruccions organitzada amb mapejat directe amb els següents paràmetres:

- Mida de la memòria cau 1KiB
- Mida del bloc de 128 bytes

El “miss penalty” en el caché d'instruccions és  $80\tau$ , on  $\tau$  és el temps d'accés de la memòria cau. Calculeu el temps total necessari per a obtenir instruccions durant l'execució del programa a la Figura 1. Supposeu que, després de portar un bloc de MP, triguem  $\tau$  en portar la dada de la caché.

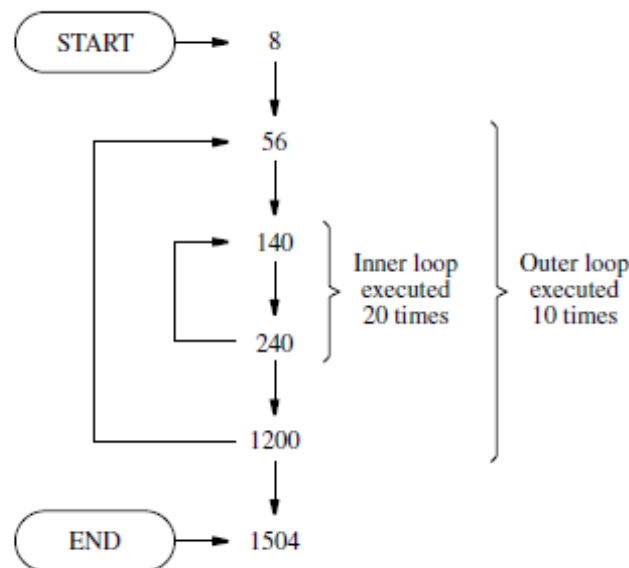


Figura 1.

1) Cap el programa a la caché ?

La caché és d'1 KiB. Quantes adreces té el bucle ?  $1504 - 8 = 1496$  adreces, cada adreça ens retorna 1 B => el programa no cap a la memòria cau, haurem de fer substitucions.


2) Quantes línies tenim ?

$C = \text{Mida Caché} / \text{Mida bloc} = 2^{10} / 2^7 = 8$  línies (cada línia de 128 B)

3) A quin bloc de la memòria caché va cada direcció ?

Adreça MP	# Bloc
0 – 127	0
128 – 255	1
256 – 383	2
384 – 511	3
512 – 639	4
640 – 767	5
768 – 895	6
896 – 1023	7
1024 – 1151	0'
1152 – 1279	1'
1280 – 1407	2'
1408 - 1535	3'

4) Blocs necessaris segons etapa del bucle:



Adreces	# Blocs
8 – 52	0
56 – 136 (comença bucle extern)	0,1
140 – 240 (bucle interior) x20	1
244 – 1200 (fi bucle extern)	1, 2, 3, 4, 5, 6, 7, 0', 1'
1204 – 1508	1', 2', 3'

Aquí ens adonem que al principi del bucle exterior necessitem els blocs 0 i 1, i que al final del bucle exterior eliminem aquests dos blocs per portar el 0' i 1'.

5) Accessos a MP:

Calculem quants cops accedim a MP durant l'execució:

- 1 cop per dur bloc 0 (primera fila taula anterior)
- 1 cop per dur bloc 1 (segona fila taula anterior)
- 0 cops pel bucle interior (tercera fila taula anterior)
- 8 cops per dur blocs que falten per acabar bucle exterior (4a fila taula anterior)
- Just acabem 1er bucle exterior, hem de tornar a començar (queden 9). Sabem que falten blocs 0 i 1 (els hem substituït). Així doncs, treiem bucle 0' i 1' per dur 0 i 1. Al final del bucle necessitem 0' i 1' i haurem d'expulsar els blocs 0 i 1 => 4 accessos a MP, com tenim 9 iteracions a fer => 9x4 accessos a MP
- 2 accessos per acabar d'executar el programa (5ena fila taula anterior)

En total son 48 accessos a MP. El temps per accedir és de  $80\tau \times 48 = 3840 \tau$

Calculem accessos amb hit (a caché):

Start section:  $(56 - 8) * \tau = 48\tau$

Inner loop:  $(244-140) * 20 * 10 * \tau = 20800 \tau$

Outter loop – Inner loop:  $10 * [(1204-56)-(244 - 140)] * \tau = 10440 \tau$

End section:  $(1508 - 1204) * \tau = 304 \tau$

Si sumem el temps que hem necessitat per fer accessos a MP i a caché, el temps total per accedir a instruccions és de 35432  $\tau$