

Problema 1: (2 Ptos)

Hallar el CPI para un procesador en cuyo sistema se tienen 4 niveles de caché (L1\$, L2\$, L3\$, L4\$), con un CPI ideal de 1 ciclo, con un tiempo de penalización de 110 ciclos de acceso a MP. Se tienen un 34% de instrucciones de tipo load/stores, con una tasa de aciertos de 98% a cache L1 de instrucciones y de 96,5% a cache L1 de datos. Incluido en el procesador se tiene, además, una cache L2 \$ con tasa de aciertos del 98,5% y un tiempo de penalización de 30 ciclos, y una cache L3 \$ con tasa de fallos del 0,8% y un tiempo de penalización de 37 ciclos. Externamente, y muy cercana al procesador se tiene una cache L4 \$, con tasa de fallos del 0,4% y un tiempo de penalización de 47 ciclos.

$$CPI = CPI_{ideal} + Memory\ Cycles$$

$$= 1 + (Mi1 * (1-M2) * 30 + Mi1 * M2 * (1-M3) * 37 + Mi1 * M2 * M3 * (1-M4) * 47 + Mi1 * M2 * M3 * M4 * 110) + 0.34 * (Md1 * (1-M2) * 30 + Md1 * M2 * (1-M3) * 37 + Md1 * M2 * M3 * (1-M4) * 47 + Md1 * M2 * M3 * M4 * 110)$$

$$1 + (0.02 * (1-0.015) * 30 + 0.02 * 0.015 * (1-0.008) * 37 + 0.02 * 0.015 * 0.008 * (1-0.004) * 47 + 0.02 * 0.015 * 0.008 * 0.004 * 110) + 0.34 * (0.035 * (1-0.015) * 30 + 0.035 * 0.015 * (1-0.008) * 37 + 0.035 * 0.015 * 0.008 * (1-0.004) * 47 + 0.035 * 0.015 * 0.008 * 0.004 * 110)$$

$$= 1.96\ cicles$$

Problema 2: (1 Ptos)

Calcula el tiempo medio de acceso (AMAT, en segundos) para el procesador del problema anterior (considerando únicamente la presencia de la cache L1), si el Tciclo = 50 psec, el tiempo de acceso a cache es de 1 ciclo y se tiene una tasa de fallos de 2% por instrucción.

$$Penanlizacion\ de\ acceso\ a\ memoria = 110\ ciclos$$

$$Tma = 1 + 0.02 * 110 = 3.2\ cicles$$

$$Tma = 3.2\ cicles * Tciclo = 3.2 * 50 = 160\ psec = 160 \times 10^{-12}\ sec = 1.6 \times 10^{-10}\ sec$$

Problema 3 (3 Ptos)

Consideremos un sistema de memoria virtual con páginas de 8 KBytes. El TLB tiene una capacidad limitada a la traducción de 4 páginas. Tenemos las siguientes referencias a direcciones virtuales: 68536, 20514, 82150, 50112, 33415, 6754, 44960, 55. En caso que la página se tenga que traer desde disco, se asignará al siguiente mayor número de página en la PT. El LRU del TLB se implementa con 2 bits (00, 01, 10, 11). Cada vez que se acierta una entrada del TLB se incrementa el valor de los bits REF. Cada 4 accesos se ponen otra vez a 00. Para elegir una entrada del TLB para escribir una nueva traducción, se elige aquella con el valor de REF más bajo (y se escribe en esa posición con un 01 como REF). Si varias tienen el mismo valor, se elige la que tenga el TAG más bajo. El estado inicial de la tabla de páginas y TLB es el siguiente:

Valido	MARco de pag o Disco
--------	----------------------

0	Disco	
1	35	
0	Disco	
1	36	
1	37	
1	38	
1	39	
0	Disco	
1	40	
0	Disco	
1	41	
Ref	TAg	Num de marco
01	3	36
10	8	40
00	10	41
00	1	35

68536:

$68536 / 8192 = 8$: hit de TLB : pag 40. incrementem el ref del tag donat

Els bits de OFFSET seran els 13 de menos pes: 0 1011 1011 1000

Ref	TAg	Num de marco
01	3	36
11	8	40
00	10	41
00	1	35

20514:

$20514 / 8192 = 2$: miss de TLB i miss també de TP: ho portem de disc :pag 42:

Offset: 1 0000 0010 0010

Valido	MARco de pag o Disco
0	Disco
1	35
1	42
1	36
1	37
1	38
1	39
0	Disco
1	40
0	Disco
1	41

Actualitzem Tlb treient el ref més baix, en cas d'empat el tag més baix:

Ref	TAg	Num de marco
01	3	36
11	8	40
00	10	41
01	2	42

82150:

82150 / 8192 = 10 : hit de TLB : pag 41 . Icrementem ref:

Offset: 1 0110 1010 1010

Ref	TAg	Num de marco
01	3	36
11	8	40
01	10	41
01	2	42

50112:

50112 / 8192 = 6 : miss de TLB, hit de TP : pag 39 : actualitzem tlb

Offset = 0 0011 1100 0000

Ref	TAg	Num de marco
01	3	36
11	8	40
01	10	41
01	6	39

Toca posar els ref de la tlb a 0, ja portem 4 accessos:

Ref	TAg	Num de marco
00	3	36
00	8	40
00	10	41
00	6	39

33415:

33415 / 8192 = 4 : miss de tlb, hit de TP : pag 37 . Actualitzem TLB:

Offset: 0 0010 1000 0111

Ref	TAg	Num de marco
01	4	37
00	8	40
00	10	41
00	6	39

6754 :

6754 / 8192 = 0: miss de TLB , miss de TP : el portem de disc, pag 43, actualitzem TP i TLB

Offset: 1 1010 0110 0010

Valido	MArco de pag o Disco
1	43
1	35
1	42
1	36
1	37
1	38
1	39
0	Disco
1	40
0	Disco
1	41

Ref	TAg	Num de marco
01	4	37
00	8	40
00	10	41
01	0	43

44960:

44960 /8192 = 5 : miss de TLB , hit de TP : pag 38

Offset: 0 1111 1010 0000

Ref	TAg	Num de marco
01	4	37
01	5	38
00	10	41
01	0	43

55:

55 / 8192 = 0 : hit de TLB : pag 43 , incrementem ref

Offset: 0 0000 0011 0111

Ref	TAg	Num de marco
01	4	37
01	5	38
00	10	41
10	0	43

Tornem a portar 4 accessos així que toca tornar a posar a 0:

Ref	TAg	Num de marco
00	4	37
00	5	38
00	10	41
00	0	43

Problema 4: (1 Ptos)

Tenemos un sistema de memoria virtual con páginas de 8 KB. El espacio de direcciones físicas es de 1 GB y el de direcciones lógicas es de 2 GB.

a) Representa todos los campos de la dirección física y de la dirección lógica.

-> les pages de 8KB, necessiten 13 bits per expressar l'offset

-> Per les 2GB de direccions llògiques es necessiten $\log_2 2GB = 31$ bits

-> Per 1GB de direccions físiques es necessiten 30bits

Marco de pagina	offset
30-13 = 17 bits	13 bits
#de pagina	offset
31-13 = 18 bits	13 bits

b) ¿Cuál sería el tamaño de la tabla de páginas en este sistema de memoria virtual para cada aplicación? Ten en cuenta que en cada entrada de la PT se guardan los bits del marco de página y 4 bits (bits de validez, de dirty y 2 bits de referencia para la LRU).

Mida PT = 2^{18} entrades * (17 bits de marc de pag. + 4 bits extres) = 5505024 bits = 672 KB

Problema 5: (1,5 ptos)

A) Indique el contenido de las direcciones de la ROM_Q+:

a) 0x31 : 11 0001 -> complementamos hasta 10 bits : 00 0011 0001

Estat actual = 00 001 = 1 = D

Estat següent: 1000 1 = 8 + 1 = bnz amb codi d'estat 12

-> contingut ROM Q+ = 0x0C

b) 0x28 10 1000 -> complementamos hasta 10 bits : 00 0010 1000

Estat actual = 00 001 = 1 = D

Estat següent: 0100 0 = 4 + 0 = st amb codi d'estat 7,

Pero per anar a ldb des de D hem de passar abans per l'estat Addr,

Llavors, l'estat següent es: Addr amb codi d'estat 5

-> contingut ROM Q+ = 0x05

B) Qué direcció(es) de la ROM Q+ conté(n) el estado:

a) Ldb

Per anar al Ldb (codi d'estat = 8) hi hem d'accedir des de l'estat Addr (estat actual = 00101)

, no ens cal bit extra d'operacions, llavors les direccions amb aquest estat són:

00101 1000 i 00101 1001

b) Addi

Per anar a l'estat Addi (codi d'estat = 4) hi hem d'accedir des de l'estat D

(estat actual = 0 0001), no ens cal bit extra d'operacions, llavors les direccions amb aquest estat son:

00001 00100 i 00001 00101

Problema 6: (1,5 ptos)

Dado el estado actual de la Unidad de Control (UC) y el contenido del registro IR, indique la palabra de control y el estado siguiente de la UC (asuma que antes de ejecutar "d" el registro R6 vale 0x0001)

A) -> estado siguiente = F

B) estado siguiente = St

C) estado siguiente = D

D) estado siguiente = bnz

Apartado	@A	@B	Pc/Rx	Ry/N	OP	F	P//L/A	@D	WrD	Wr-Out	Rd-In	Wr-Mem	Ldlr	LdPc	Byte	Alu/R@	R@/Pc	N (hexa)	ADDR-IO (hexa)
----------	----	----	-------	------	----	---	--------	----	-----	--------	-------	--------	------	------	------	--------	-------	-------------	-------------------

a	011	xxx	x	0	xx	xxx	xx	xx	0	1	0	0	x	0	x	x	x	xxxx	02
b	xxx	101	0	0	00	100	xx	xxx	0	0	0	0	0	0	x	x	x	FFFa	xx
c	xxx	xxx	1	0	00	100	xx	xxx	0	0	0	0	1	1	0	1	0	0002	xx
d	110	xxx	1	0	00	100	xx	xxx	0	0	0	0	0	0	x	x	x	FFE8	xx