DEPARTAMENT DE ARQUITECTURA DE COMPUTADORS

EPSEVG 340375 - ESTRUCTURA DE COMPUTADORS II

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

PROVE D'AVALUACIÓ 2º PARCIAL - 2020/2021-Q1 15-**Gener-2021**

Cognoms i Nom: ***DNI:***

Problemas de Rendimiento **(3 Ptos)**

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 instruccionesde 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 = CPIIdeal + (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)

1-0.98=0.02 = mI1

1-0.965 = 0.035 = md1

1-0.985 = 0.015 = m2

0.08 = m3

0.004 = m4

CPI = 1 + (0.02 · (1 – 0.015) · 30 + 0.02 · 0.015 · (1-0.08)· 37 + 0.02 · 0.015 · 0.08 · (1-0.004) · 47 + 0.02 · 0.015 · 0.08 ·0.004 ·110) + 0.34 · (0.035 · (1 – 0.015) · 30 + 0.035· 0.015 · (1-0.08)· 37 + 0.035· 0.015 · 0.08 · (1-0.004) · 47 + 0.035· 0.015 · 0.08 · 0.004 ·110) = 1.96074

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.

Tma(ciclos) = 1+ 0.02*110 = 3,2 ciclos
Tma(s)= 3.2 ciclos * 50*10^-12 s/ciclo = 1.6x10^-10 s

Problemas de Memoria Virtual **(4 Ptos)**

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:

PT

Válido	Marco de Página
0	Disco
1	35
0	Disco
1	36
1	37
1	38
1	39
0	Disco
1	40
0	Disco
1	41

TLB

Ref	Tag	Número de Marco
01	3	36
10	8	40
00	10	41
00	1	35

Dados los estados iniciales mostrados, indica **paso a paso** el estado del sistema para las secuencia de accessos y responde a lo siguiente:

a) Si es un acierto o fallo en la TLB
b) Un acierto en la Tabla de Páginas o un fallo de pàgina
c) Si se lleva una traducción al TLB, ¿qué entrada de las 4 es seleccionada?
d) Obtener el valor de **offset** para cada acceso.

	Válido	Marco de Página o en Disco
0	0	Disco
1	1	35
2	0	Disco
3	1	36
4	1	37
5	1	38
6	1	39
7	0	Disco
8	1	40
9	0	Disco
10	1	41

REF	TAG	NUM MARC
01	3	36
10	8	40
00	10	41
00	1	35

8KB = 8192B

2^13

Offset = 13 bits

68536,
20514,
82150,
50112,
33415,
6754,
44960,
55

<u>68536</u>

68536/8192 = 8.3662109375

10000101110111000

TAG= 8
TLB Hit
PT: No
hace falta
interaccion

Offset = 0101110111000

REF	TAG	NUM MARC
01	3	36
11	8	40
00	10	41
00	1	35

20514
20514/8192 = 2.50415039063
101000000100010
TAG= 2
TLB Miss
PT: fallo de pagina
Offset = 1000000100010

	Válido	Marco de Página o en Disco
0	0	Disco
1	1	35
2	1	42
3	1	36
4	1	37
5	1	38
6	1	39
7	0	Disco
8	1	40
9	0	Disco
10	1	41

REF	TAG	NUM MARC
01	3	36
10	8	40
00	10	41
01	2	42

82150
82150/8192 = 10.0280761719
10100000011100100
TAG = 10
TLB Hit

PT: no hace falta interaccion
Offset = 0000011100110

REF	TAG	NUM MARC
01	3	36
10	8	40
01	10	41
01	2	42

50112
50112/8192 = 6.1171875
1100001111000000
TAG = 6
TLB Miss
PT: Hit
Offset = 0001111000000

REF	TAG	NUM MARC
01	3	36
10	8	40
01	10	41
01	6	39

33415
33415/8192 = 4.07897949219
1000001010000110
TAG = 4
TLB Miss
PT: Hit
Offset = 0001010000110
Se han reiniciado todos los refs a 00 despues de 4 accesos

REF	TAG	NUM MARC
01	4	37
00	8	40
00	10	41
00	6	39

6754
6754/8192=0
1101001100010
TAG = 0
TLB Miss
PT: fallo de pagina
Offset = 1101001100010

	Válido	Marco de Página o en Disco
0	1	43
1	1	35
2	1	42
3	1	36
4	1	37
5	1	38
6	1	39
7	0	Disco
8	1	40
9	0	Disco
10	1	41

REF	TAG	NUM MARC
01	4	37
00	8	40
00	10	41
01	0	43

<u>44960</u>
44960/8192 = 5.48828125
1010111110100000
TAG = 5
TLB Miss
PT: Hit
Offset = 0111110100000

REF	TAG	NUM MARC
01	4	37
01	5	38
00	10	41
01	0	43

<u>55</u>
55/8192 = 0
110111
TAG = 0
TLB Hit
PT: no hace falta operación
Offset = 0000000110111

REF	TAG	NUM MARC
01	4	37
01	5	38
00	10	41
10	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.
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).

Paginas de 8KB se necesitan 13 bits de offset

VIRTUAL
2GB = 2^31
31-13 = 18

NUMERO PAGINA VIRTUAL	OFFSET
18 bits	13 bits

FISICA
1GB = 2^30

MARCO DE PAGINA	OFFSET
17 bits	13 bits

Bits marco pagina = 17 bits
Bit validez = 1 bit
Bit de dirty = 1 bit
Bit de referencia = 2 bits

18 bits para direccionar la Tabla de Páginas por lo tanto la PT tendría 2^18 entradas, cada una con: 17 + 4 = 21 bits/entrada

En conclusión el tamaño de tabla de paginas es = 2^18 entradas * 21 bits/entrada = 5505024bits que pasados a bytes son = 688128 bits que entre 1014 para pasar a KB da 672KB

Problemas de Microprogramación **(3 ptos)**

Problema 5: **(1,5 ptos)**

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

a) 0x31 = 00110001
Complementamos a 10 bits
00 0011 0001
00001 = 1 = D (Estado actual)
10001 = E12 = BNZ

12 = 1100
Se guardara en la Rom Q 0x0C

b) 0x28 = 101000
Complementamos a 10 bits
00 0010 1000
00001 = 1 = D (Estado actual)
01000 = E7 = ST

Mirant el graf d'estats de la UC, si estem a l'estat D NO podem anar a ST directament Llavors:
Aquesta entrada NO guarda ningú contingut a la ROM_Q+

B) Qué direccion(es) de la ROM Q+ contiene(n) el estado:

a) Ldb
Arribem a Ldb (0101x) desde Addr (0011x 0100x 0101x 0110x) i amb codi operacio + extensio
0011x0101x
0100x0101x
0101x0101x
0110x0101x

b) Addi
Arribem a Addi desde D (00001) i amb codi operacio + extensió 0010x
000010010x
0000100100 =0x24
0000100101 = 0x25

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)

Apartado	Nodo/Estado	Instrucción en el IR	Nodo/Estado Siguiente

a	**OUT**	**OUT 0x0002, R3**	**F**
b	**Addr**	**ST -6(R2), R5**	**ST**
c	**F**	**X**	**D**
d	**D**	**BNZ R6, -12**	**BNZ**

Apartado	@A	@B	Pc/Rx	Ry/N	OP	F	P/I/L/A	@D	WrD	Wr-Out	Rd-In	Wr-Mem	LdIr	LdPc	Byte	Alu/R@	R@/Pc	N (hexa)	ADDR-IO (hexa)
a																			xx
b																			xx
c																			xx
d																		HEX(-12*2) = FFE8	xx