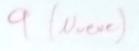
Arquitectura de computadoras 2022 - Parcial 2B



Nombre: Ivan Renison

Ejercicio 1

Considere una caché de 1Mbyte, ASOCIATIVA por conjuntos de 4 vías, dispuesta en un procesador de 64 bits con una capacidad de direccionamiento de 4Gbytes (cada byte es directamente direccionable en memoria). Se sabe que la memoria principal posee 16M bloques Se pide

a) Completar cada casillero con el número de bits de cada campo del formato de dirección de memoria principal:

Tag	Set	Word	Offset byte(*)
1.4	7 0	5	3

^{*} Llenar con 0 en el caso que no corresponda

b) Suponga que cada LÍNEA de la caché contiene además un bit de validación (V) y 4 bits de contador de accesos. Cual es el tamaño completo de una LÍNEA (expresado en bits) de la caché, considerando datos, tags y los bits de status antes mencionados?

Respuesta: 1067 bits.

c) Cuál es el tamaño total del área de TAG de la caché completa, expresada en bits?

Respuesta: 5 7 3 44 bits.

d) Si la memoria caché y la memoria principal tienen tiempos de acceso de 3ns y 250ns respectivamente, ¿qué hit rate (tasa de acierto) se necesitaria para obtener un tiempo promedio de acceso a memoria (AMAT) de 40.5ns?

Respuesta: 85 % s

Ejercicio 2

Considere la ejecución del siguiente segmento de código LEGv8 para N > 0, donde N→ X3, y X6 contiene la dirección base del arreglo A[] del tipo uint64_t.

ADDRESS	<label></label>	OPCODE	ASSEM	BLY
0x03394834 0x03394838	<count_z>: <loop_for>:</loop_for></count_z>	0x8b1f03e0 0x8b0303e9 0xb40000e9 0xf84000ca 0xb500004a	add add cbz ldur cbnz	x0, xzr, xzr x9, xzr, x3 x9, NEXT x10, [x6] x10, NOT Z
0x03394848	<not_z>:</not_z>	0x9410b5bc 0xd1000529 0x910020c6 0x17fffffa	bl subi addi b	ZERO x9, x9, #0x1 x6, x6, #0x8 LOOP_FOR
0x03394854	<next>:</next>	0x8b1f03ff	add	xzr, xzr, xzr
0x037c1f38	<zero>:</zero>	0x91000400 0xd61f03c0	addi br	x0, x0, #0x1 x30

Nombre: Ivan Renison

El sistema posee una CACHE exclusiva para INSTRUCCIONES de correspondencia ASOCIATIVA POR CONJUNTOS de 2 VÍAS de 128 bytes y 2 palabras de 32 bits por linea, sobre un procesador de 64bits, CPI = 1, que resuelve todos los data y control hazard sin necesidad de stalls, tiene una memoria principal de 4G palabras de 1 byte cada una. Considerar que la CACHÉ contiene los datos mostrados a continuación al inicio de la ejecución del segmento:

	Jain.	Via #0			Via #1
Tag	V	Data	Tag	V	Data
00ce520	1	8b1f03ff_8b1f03ff	00ce521	0	8b1f03ff_8b1f03ff
3ffffff	0	fffffff_fffffff	3ffffff	0	fffffff_fffffff
01f5580	1	8b1f03ff_17fffffa	00ce581	0	00000000_000000000
3ffffff	1	301f030f_87f00ffa	3ffffff	1	9b1f03ff_570ffff8
3 <i>ffffff</i>	0		3ffffff	0	fffffff_fffffff
	0		3ffffff	0	fffffff_ffffff
	-		01f5580	1	9b1f0300_870f65f8
	-		3ffffff	0	00000000_000000000
	00ce520 3ffffff 01f5580	00ce520 1 3ffffff 0 01f5580 1 3ffffff 1 3ffffff 0 3ffffff 0	Tag V Data 00ce520 1 8b1f03ff_8b1f03ff 3ffffff 0 ffffffffffffffffffffffff 01f5580 1 8b1f03ff_17fffffa 3ffffff 1 301f030f_87f00ffa 3ffffff 0 ffffffffffffffffffffffffffffffffffff	Tag V Data Tag 00ce520 1 8b1f03ff_8b1f03ff 00ce521 3ffffff 0 ffffffffffffffffffffffffffffffffffff	Tag V Data Tag V 00ce520 1 8b1f03ff_8b1f03ff 00ce521 0 3ffffff 0 fffffffffffffffffffffff 3ffffff 0 01f5580 1 8b1f03ff_17fffffa 00ce581 0 3ffffff 1 301f030f_87f00ffa 3ffffff 1 3ffffff 0 ffffffffffffffff 3ffffff 0 3ffffff 0 00000000_0000000 3ffffff 0 3ffffff 0 ffffffffffffffffffff 01f5580 1

a) Determinar el estado de la CACHÉ al final de la ejecución del segmento para N = 100. Para esto completar la siguiente tabla SOLO con el contenido de las líneas de CACHÉ que hayan sufrido modificaciones. El tamaño de la tabla no representa la cantidad de líneas necesarias.

Set	Tag	٧	Data	Tag	V	Data
AND STREET	onces20/	1	867 + 03 90 _ 86030389	0145580/	1	9 6 1 4 0300 _ 8 7 0 + 6515
1	00(8520/	1	64000009-18400000	3ffffff	0	00000000_00000000
	00 (95) 1					961F03FF_961F0#F
	00(8521/	1	11000529_ 7100206	3fffffff/	0	ttttt-tttt
	00(8521	1	1744446 2 36140344/	00 (858%	0	0000000_00000000

b) Determine la cantidad de ciclos de clk necesarios para su ejecución en las mismas condiciones del punto a), considerando que cada MISS de caché tiene un tiempo de ejecución de 10 ciclos. Suponer que el pipeline ya se encuentra en régimen y que la CACHE de DATOS solo produce aciertos en los accesos, por lo que no se tiene penalidades por acceso a datos.

Expresión: 5*(10+1) + 100 1 6 Respuesta: 655 ciclos de clk. X

Manual Train Town Co.

Zimennanin 3

The second of arquitement EDVS 2-issue, que predice los saltos perfectamente, as muse que los maneros son manejados por hardware, con una modificación que municipal de control son manejados por hardware, con una modificación que municipal de control son propier una instrucción pueda ser cualquier tipo y la otra deba ser una instrucción pueda ser una cualquier tipo y la otra deba ser una tipo de control deba ser una cualquier tipo y la otra deba ser una tipo deba ser una cualquier tipo y la otra deba ser una tipo deba ser una cualquier tipo y la otra deba ser una cualq

25. **(\$15175). St. **(\$1518)*

100 STER NO. [VII.,48]

Instrucción de Icológuier Spo	Instrucción aritmésica/lógica
1	2
3	4
5	hop
6	10 P
7	9
710	
11	12
13	nop

LOOP END:.... COME

a) Sin alteria e orden de las instrucciones, mostrar en la tabla de arriba cómo organizaria los asse passess para ejecutar el programa en la menor cantidad posible de ciclos de clock (tasta instrucción sólo puede agruparse con la inmediata anterior, la inmediata posterior o una rupi). Usar los números correspondientes para referirse a las instrucciones del código.

- b) Wosstrar el orden de ejecución del codigo del punto "a" en el procesador 2-issue (sólo hasta completar una iteración del bucle "LOOP"). Indicar los caminos de forwarding political.
- c) El siguiente conigni resulto de aplicar las técnicas de loop unrolling y register renaming al fragmento de conigni dado. Sin alterar los resultados obtenidos tras la ejecución del mismo y asumiendo que siempre en la instrucción nº1 se asigna el número 40 al registro X0, eliminar (tactitat) la mayor cantidad de instrucciones posible y completar los espacios vacios en las restames.

Numbre: Iván Renison

Se cuenta con un predictor por torneos que está compuesto por dos predictores más simples un predictor global con un GR de 4 bits y un predictor de dos bits clásico. Considerando el siguiente segmento de código en LEGv8 y que los registros implicados estân inicializados con los siguientes valores: X0=0, X1=99, X3=138, GR=0011;

están inic	ializados con los de 11. vo. x0
1 - 1 :	addts 40, 40 addis x0, x0,
21	b.eq E1 add ×1, ×zr, ×zr
4× E11	empi X1, #0 b.eq E2
6 >	addi X1, X1, #1 cmpi X1, 100
7 x F2:	b.neq L subis x3, X3, #1

#10	PHT	redictor	global				
0	Dirección		Contenido				
	0017		0.7	(7)			
			00	(11)			
	0111		01 0	(7)			
	7710		010	CNI			
	1700 x		00	1(111)			
				Chan.			

a) Indicar que posiciones de la PHT del predictor global se modifican y considerando que esta completamente inicializada en cero, que valores quedarian almacenados. (hasta el primer branch a

b) Si este segmento de código se ejecuta muchas veces, indicar cuál de los dos predictores

obtendrá mejores resultados para cada uno de los saltos:

obtend	ra mejores	
2 = D2	4031	-
5= 13	1	

Se muestra a continuación el estado de los registros y las

reservation station en un determinado momento de un procesador out-of-order. Deducir el segmento de código que se está ejecutando

Issue = 4 instrucciones Load = 3 RS / 2 clk Store = 3 RS / 2 clk ALU entero = 3 RS / 1 clk Multiplicación enteros = 3 RS / 2 clk ALU punto flotante = 3 RS / 2 clk Multiplicación punto flotante = 3 RS / 4 clk

		Reservation stations Qk					
			V	Vk	Q)	0	#0
Hame	BUSY	Ob	V1		alu int 1		#8
		load			alu int 1	0	#O
sort I		load		[X0]	atu fp 1	0	
said #		store		[Au]			
work I				- 10	0	0	
sore if	F	subi	[X2]	16	alu int 1	0	
su int 1		sub		[KA]	and an a		
hi tell 2		0.00					
uni trat 3	1 1					mult to 1	
salt Ini I		add			load 4 2	This is a	
in to 1		200				-	
to to 1		- Aller		[D4]	foarf 1.	0	
uni lp 1		(liv					-
not be 8		- 11	-		alu int 2	0	
rarefi		cb2					

_				Register Status				07
		01	1 00	0.3	D4	05	06	D/
	0.0	lead i	load 2	mult to 1				
1/1	aiu tp x	1000 1	V is	×3	Nil	X5	X6	X7
	pho mi d	×I	alu int 1	-				

Se quenta con un predictor por tomeos que está compuesto por dos predictores más simples: un predictor global con un GR de 4 bits y un predictor de dos bits dásico. Considerando el siguiente segmento de código en LEGi8 y que los registros implicados están inicializados con los siguientes valores: X0=0, X1=99, X3=138, GR=0011;

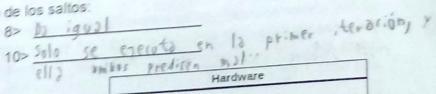
están inic	ISHZSON ON THE
10 63	addis xo, we addis xo, xo
2)	b.eq E1
3>	add x1, xzr, xzr
4> E1:	cmp1 X1, #8
59	b.eq E2
62	addi X1, X1, #1
7> E2:	cmpi X1, 188
82	b.neq L

## PHIT predic	PHT predictor global						
Dirección	Contenido						
0011 /	0.7 (7)						
0111 /	00 (11)						
7110	018 (7)						
1100 x	00 100						
	25:43.						

subis x3, X3, #1 92 a) Indicar que posiciones de la PHT del predictor global se modifican y considerando que esta completamente inicializada en cero, que valores quedarian almacenados. (১৯৮३ el primér branch à L) b) Si este segmento de código se ejecuta muchas veces, indicar cuál de los dos predictores

obtendrá mejores resultados para cada uno de los saltos:

abtend	ra mejores resultan	
2002	9021	-
5> 03	1993	_



Ejercicio 5

Se muestra a continuación el estado de los registros y las reservation station en un determinado momento de un procesador out-of-order. Deducir el segmento de código que se está ejecutando

Hardware	-
Issue = 4 instrucciones	
Load = 3 RS / 2 clk	
Store = 3 RS / 2 clk	
ALLLentero = 3 RS / 1 clk	
Multiplicación enteros = 3 RS / 2 CIK	
ALLE GUIDTO flotante = 3 RS / 2 CIX	
Multiplicación punto flotante = 3 RS / 4 clk	

	Reservation stations						A
Name				Vk	Qi	Qk	#0
	Busy	Op	Vj		alu int 1	0	
		foad	-		alu int 1	0	#8
iced 1		load		Direct.	alu fp 1	0	#0
inad 2	1	store	-	[X0]	and ip 2		
store 1	1 0				0	0	
store 2		subi	[X2]	16		0	
siu int 1		sub		[X1]	alu int 1		
alu int 2							
mult int 1						462	
mult int 2		add			load 4 2	mult fp 1	
ally fp 1		BUG					
situ fp 2		div		[D4]	load 1	0	
mult fp 1		drv					
mult fp 2		-6-2		-	alu int 2	0	2
Branch		cbz					

				Register Status				
	T 00	01	D2	D3	D4	D5	Dő	07
-	alu fp 1	load 1	load 2	mult fp 1			A SECTION	
Q)	and the r	VI.	X2	×3	Xd	X5	X6	X7
	alu int 2	~ ~	alu int 1					