

Instituto Tecnológico de Costa Rica Área Académica de Ingeniería en Computadores Programa de Licenciatura en Ingeniería en Computadores Curso: CE-4301 Arquitectura de Computadores I Profesor: Ronald García Fernández Semestre: I, 2019	Examen Final Fecha: 10 de Junio, 2019 Puntos totales: 100 Puntos obtenidos: 59
--	---

Nombre: Alexis Gavriel Carné: 2016085662

Instrucciones Generales

- Trabaje individualmente en
- Utilice cuaderno de examen u hojas blancas numeradas para resolver la prueba.
- Escriba de manera legible y ordenada.
- Sea lo más detallado posible en sus respuestas no deje nada abierto a interpretaciones.
- Utilice bolígrafo para resolver la prueba. No se aceptarán reclamos sobre respuestas con lápiz
- El fraude se castiga según estipula el reglamento de enseñanza-aprendizaje del TEC.
- Tiempo para resolver la prueba es de 2 horas.
- No se permite el uso de celulares o algún otro tipo de dispositivo móvil.

Parte I Teoría [valor 56 puntos]

Responda las siguientes preguntas, sea detallado en sus respuestas, emplee diagramas para elaborar si es necesario, cada pregunta tiene el mismo puntaje (7 puntos)

- 1- Describa en qué consiste la calendarización dinámica, incluya en su descripción: ventajas y desventajas, los cambios requeridos para tener *OOE* en un pipeline tradicional de 5 etapas y relación respecto a excepciones.
- 2- Explique detalladamente cómo se manejan cada una las dependencias (*RAW, WAR, etc*) en una arquitectura *VLIW*.
- 3- Explique en qué consisten las siguientes técnicas para mejorar el desempeño de la memoria *DRAM*: *Row Buffer, Synchronous DRAM, DRAM Banking*. Incluya como se relacionan con los principios de localidad temporal y espacial en cada una.
- 4- Explique detalladamente 2 técnicas para mejorar el desempeño de la memoria caché.
- 5- Describa detalladamente el concepto de *Direct Memory Access*, indicando los modos de operación, y qué beneficios permite respecto al desempeño del sistema.
- 6- Explique cómo se relacionan los conceptos de Memoria Virtual, Multiprogramación, Protección de Memoria.
- 7- Describa que es paginación y una de las técnicas empleadas para el reemplazo de páginas, ¿Es la técnica descrita afectada por la anomalía de *Bélády* (Explique que es esta anomalía)?
- 8- Explique que es el *Working Set* en el contexto de memoria virtual y algún algoritmo para realizar el cálculo de su tamaño.

Parte II Desarrollo [valor 44 puntos]

Resuelva los siguientes ejercicios no omita ningún paso en sus cálculos

1- Se desea emplear la especialización de *write buffer* llamado *victim cache* para mejorar el desempeño de un sistema que tiene un caché L1 con mapeo directo de 16KB de capacidad, y con un caché L2 4 way set associative de 128KB con las siguientes características:

- L1 HIT time de 3 ciclos reloj.
- L1 Miss rate de 7.6%.
- L2 HIT rate 98%.
- L2 HIT time 10 ciclos de reloj.
- El 70% de misses son debido a conflictos y capacidad.
- El tiempo de acceso a memoria principal es de 50 ciclos de reloj.

De forma experimental se tienen los siguientes resultados para un código particular:

- El *victim cache* logra eliminar el 90% de los misses por conflicto para L1.
- El *victim cache* tiene una penalización de 5 ciclos de reloj en caso de HIT.

Respecto al sistema anterior se le pide:

- a- Calcule el AMAT sin uso de *victim cache* (5 puntos)
- b- Calcule el AMAT con el uso del *victim cache* (10 puntos)
- c- Mejora o no el AMAT el uso del *victim cache*, si su respuesta es NO justifique como podría mejorar el *victim cache* (5 puntos)

2- Se desea diseñar una memoria caché con las siguientes características

- El ancho de palabra de datos de 8 bytes.
- El tamaño de la línea de cache de 512 bits.
- 4-way set associative.
- Las direcciones son de 48 bits.
- La capacidad del caché es de 2048 blocks.
- Los datos son direccionables por media palabra.

- a- ¿Cuál es la capacidad del caché deseado en KB? (4 puntos)
- b- Indique mediante un diagrama la estructura del caché que se desea diseñar, señale los elementos básicos del caché y tamaños de los mismos. (20 puntos)

Pregunta Extra (7 puntos)

Escoja únicamente una de las siguientes preguntas:

- 1- Explique en qué consiste el fenómeno *rowhammer*.
- 2- Explique cómo se relaciona la ejecución especulativa con alguno de los siguientes problemas: *Spectre, Spoiler, Meltdown*.
- 3- Describa los diferentes niveles de privilegio descritos en Armv8.
- 4- Describa el algoritmo de manejo de interrupciones del Arm GIC.
- 5- Describa los tipos de especulación implementados en IA-64.

Gavriel Gómez

516 085 662

Arquitectura de Computadores I

2019, I

Ronald Guerra

Parte I

1. Calendarización dinámica.

Busca utilizar al máximo a cada componente del hardware, para ello se aplican técnicas como reordenamiento de instrucciones para que se reduzca el tiempo en idle del procesador. Además logra controlar los distintos módulos del hardware al mismo tiempo.

ventajas: ~~se reduce~~ se incrementa el ILP, más instrucciones en menor tiempo.

Es invisible para el programador, ya que serializa por hardware.

desventajas: mayor consumo de potencia (más trabajo por módulo)
se incrementa la latencia de algunas instrucciones.
Es dependiente del hardware donde se ejecute.

Se debe implementar la unidad de despacho de instrucciones, los registros o módulos que controlan ~~los~~ los módulos para cada instrucción y la unidad de reordenamiento de datos.

Se debe ser consistente ante excepciones. A pesar de utilizar ejecución fuera de orden en instrucciones, si se produce una excepción todas las instrucciones anteriores a la instrucción que produjo la excepción deben haber finalizado y todas las instrucciones siguientes ya realizadas se deben invalidar.

2) VLIW:

RAW: se colocan stalls para que se lea después de escribir, ~~write after read~~

WAR: se debe validar que el dato leído sea el correcto

WAW: se asegura que se escriba en orden correcto y validar el valor final

reg renombrar

- 3) Row Buffer: Se cargan ~~diversas~~ filas de datos al mismo tiempo. espacial: se traen los datos contiguos al requerido temporal: se guardan en cache por si vuelven a ser requeridos.

Synchronous DRAM:

DRAM Banking:

1/7

Relacion
con Localidad

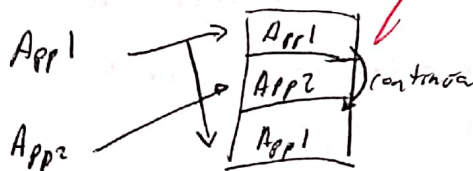
- 4) Incrementar el tamaño del cache: si se incrementa el tamaño se reduce el miss rate, hay ~~mayor~~ más capacidad para ~~guardar~~ almacenar datos.

~~7/7~~ Asociatividad: modificando la asociatividad se puede reunir el tiempo de escritura (aumentando asociatividad) pero disminuyendo el tiempo de lectura (disminuyendo asociatividad).

- 5) Direct Memory Access: Las direcciones de memoria se mapean en otro lugar. mientras que ~~el~~ el programa direcciona sus datos en memoria desde $0-n$, estos valores son mapeados, se agrega una dirección K para que las instrucciones vayan de $K \rightarrow n+K$

0/7

- 6) Para la multiprogramación, ejecución de varios programas se utiliza la memoria virtual para asignar la memoria a un programa o varios. Cada proceso tiene su memoria protegida de otro, para evitar modificaciones.



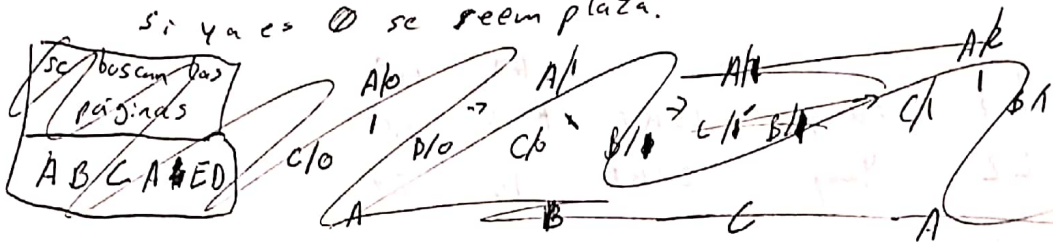
Pudo elaborarse más

Paginación: bloques de datos o instrucciones se separan en páginas. el TLB administra cuáles páginas están cargadas, si se busca un dato que no está ahí, se trae la página que contiene el dato buscado.

Cuando no hay espacio para cargar una página, se debe reemplazar una ya cargada.

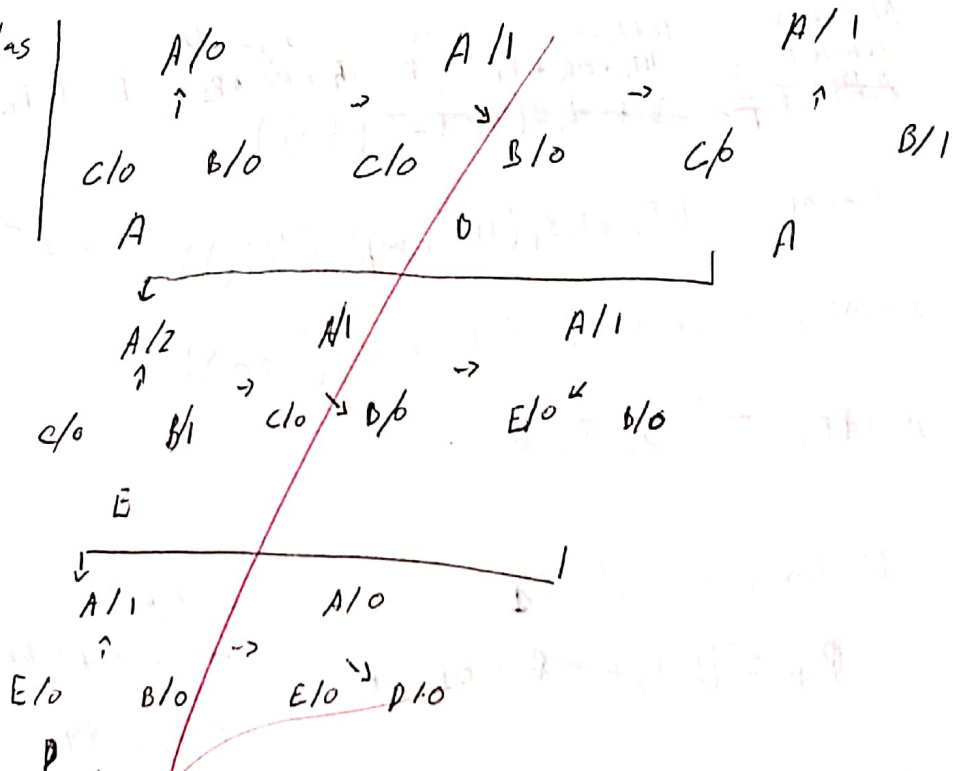
A Clock: se reemplaza la menos usada, se lleva un contador al no tener la página deseada, se reduce el contador, al tenerla se incrementa.

Si ya es 0 se reemplaza.



se buscan las páginas:

~~717~~ A B A E P



Belady: incrementar la cantidad de páginas puede no mejorar el tiempo de respuesta.

La técnica es afectada ya que si hay más páginas se deben realizar más comparaciones, se lleva más tiempo completar un ciclo del "reloj".

Se supone de solo afecta FIFO

8) Se refiere a las páginas a utilizar entre las ~~4~~ ^{2/7} cargadas

$$VP / PA + VP \times PA$$

virtual page
page amount (cantidad de
páginas cargadas)

Parte II

			HT	MR
L1	1way	16KB	16 3	7,6%
L2	4way	128KB	10	2%
			50	
mem				

a) $AMAT = \text{Hit time} + \text{miss rate} \cdot \text{miss penalty}$
 $AMAT_1 = HT_1 + MR_1 \cdot P_1$; $P_1 = HT_2 + MR_2 \cdot P_2$; $P_2 = HT_R$
 ~~$AMAT_1 = 3 + 7,6\% (10 + 2\% (50))$~~

$$AMAT_1 = HT_1 + MR_1 (HT_2 + MR_2 (HT_R))$$

$$AMAT_1 = 3 + 7,6\% (10 + 2\% (50))$$

$$AMAT_1 = 3,836$$

5/5

b) Victim cache ll L2

$$P_1 = HT_2 + MR_2 \cdot MR_2 +$$

7,6%
10% L2 0,76% $\cdot P_1$
90% L2 6,84% $\cdot HT_V$

Victim Hit = 5

$$AMAT_2 = HT_1 + MR_{L1} \cdot P_1 + MR_{L2} \cdot HT_V$$

$$3 + 0,76\% (10 + 2\% (50)) + 6,84\% \cdot 5$$

$$AMAT_2 = 3,4256$$

5/10

No entiendo
su lógica

Se mejoró el AMAT con el cache víctima

7/5

word = 8B

lines = bloque = 512 = 64B / 8 words

4-way

capacidad 2048 bloques

direccionable 486

direccionable por media palabra (4B)

a) Tamaño

$T_{size} = \text{capacidad de bloques} \times \text{tamaño por bloque}$

$$T_{size} = 2048 \times 64B$$

$$T_{size} = 128KB$$

4/4

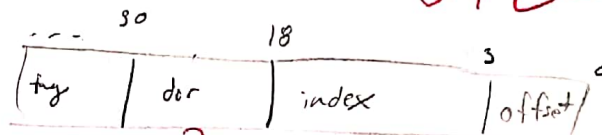
b)

words-per-block = 64B / 8B = 8

offset = $\log_2(w-p-b)$

$$= 3$$

[2:0]



5/20

sets = $T_{size} / \text{num-way}$

$$2^{17} / 4 = 2^{15}$$

index = 15 [15:3]

direccionable 4B = 32B

[49:18]

tag

dir

