

Repaso

Arquitecturas + Cache

Alexandra Ibarra
Rodrigo Urrea
CC3301 - Otoño 2024

Arquitecturas

- **Microprogramada**

- 1 instrucción toma varios ciclos del reloj
- Simple de implementar

- **En pipeline**

- 1 instrucción puede tomar 1 ciclo del reloj.
- Se procesan varias instrucciones en distintas fases a la vez.
- Una instrucción que lenta bloquea el pipeline (ej. división, multiplicación, etc.)
- Al saltar se debe anular trabajo hecho.

Arquitecturas

- **Superescalares (n pipelines)**

- n instrucciones por ciclo del reloj.
- Instrucciones lentas siguen bloqueando el pipeline.
- También se puede bloquear el pipeline en caso de dependencias (operando no listo)

- **Ejecución especulativa y fuera de orden (OOO)**

- n instrucciones por ciclo del reloj.
- En caso de instrucciones lentas o si falta calcular operandos el pipeline no se para, se ejecutan instrucciones posteriores.
- Se predicen los saltos para no tener que esperar a calcularlos.
- Si se equivoca la predicción se pierde el cómputo.
- Muy difíciles de implementar.

Ejercicio Arquitecturas (C3 Primavera 2023 P2-b)

La tabla de la derecha muestra las instrucciones Risc-V ejecutadas por un programa.

Haga un diagrama que muestre el ciclo en que se ejecuta cada etapa de las instrucciones, considerando una arquitectura (i) en **pipeline** con etapas *fetch*, *decode* y *execute*, y (ii) **superescalar**, con 2 pipelines con las mismas etapas de (i). Suponga que el salto en **E** no ocurre y el salto en **F** sí ocurre (no hay ningún tipo de predicción de saltos).

A	add	a1, a2, a3
B	andi	t4, a1, 7
C	addi	t5, t4, 1
D	add	t6, t4, 1
E	blt	a1, t4, l
F	beq	t5, a1, P
G	add	...
H	sub	...
I	xor	...
J	andi	...
	...	
P	addi	a3, a2, 1
Q	sub	t7, a4, t5

Solución

Pipeline			
	Fetch	Decode	Exec
1	A		
2	B	A	
3	C	B	A
4	D	C	B
5	E	D	C
6	F	E	D
7	G	F	E
8	H	G	F
9	P		
10	Q	P	
11		Q	P
12			Q

Superescalar - 2 pipelines			
	Fetch	Decode	Exec
1	A B		
2	C D	A B	
3	C D	B	A
4	E F	C D	B
5	G H	E F	C D
6	G H	F	E
7	I J	G H	F
8	P Q		
9		P Q	
10			P Q
11			
12			

Memorias Caché

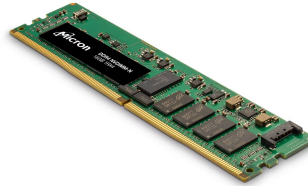
Disco duro / SSD

- Baratas y densas.
- Muy lentas.



Memoria DRAM

- Más caras y menos densas.
- Más rápidas pero no para CPU.



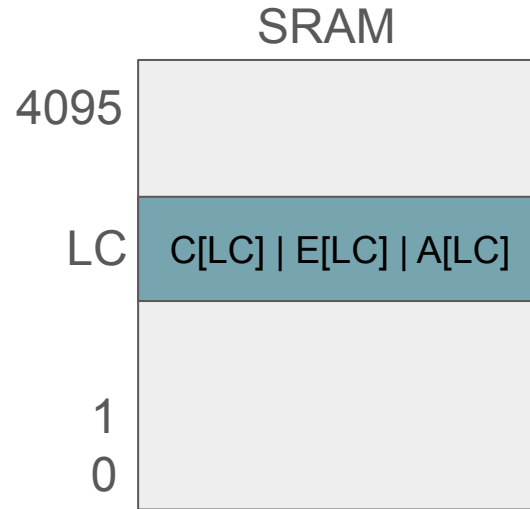
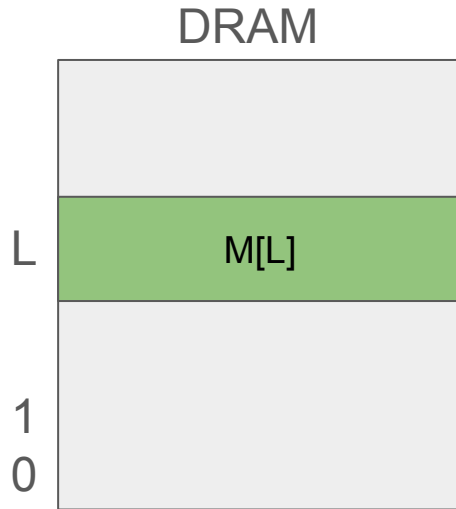
Caché (SRAM)

- Mucho más caras y menos densas.
- Más rápidas, lo suficiente para CPU.



Memorias Cache

Caché 64 de KB, 1 grado de asociatividad con líneas de 16 bytes.



L y LC etiquetas

$$D = M[L] \leftrightarrow L = D/16$$

$$D = C[LC] \leftrightarrow LC = L \% C$$

$$A[LC] = 1 \leftrightarrow C[LC] \text{ alterada}$$

$$\text{acierto} \leftrightarrow E[LC] = L$$

Ejercicio Cache (C3 Primavera 2023 P2-a)

La figura muestra un extracto del estado actual de un caché de *4 KB* (2^{12} bytes) de *1 grado* de asociatividad con *256 líneas* de *16 bytes*.

Por ejemplo en la línea **6b** del caché (en hexadecimal) se almacena la línea de memoria que tiene como etiqueta **86b** (es decir, la línea que va de la dirección **86b0** a la dirección **86bf**).

Línea caché	Etiqueta	Contenido
9c	79c	
6b	86b	
34	a34	

Un programa accede a las siguientes direcciones de memoria (en hexadecimal): **79c0**, **86b4**, **a34c**, **56b0**, **79c4**, **a348**, **d340**, **79c8**, **86b8**. Indique qué accesos a la memoria son aciertos en el caché, cuáles son desaciertos y rehaga la figura mostrando el estado final del caché. Por ejemplo el acceso **79c0** es un acierto.

Solución

Aciertos:

79c0, 86b4, a34c, 79c4, a348, 79c8

Desaciertos:

56b0, d340, 86b8

Línea caché	Etiqueta	Contenido
9c	79c	
6b	86b	
34	d34	