

Cours 5

Superscalar
ARCH1

12/10/23

How to increase performance?

RISC

$$\text{CPI} = 1$$

CPI fijo, no es posible ser más rápido que un ciclo por instrucción

Pipeline

F 

Lo que se puede hacer para aumentar la frecuencia, es aumentar las etapas del Pipeline

F 


$$\text{Performance} = f / \text{CPI}$$

$$\text{Performance} \uparrow = f \uparrow / \text{CPI}$$

$$\text{CPI} = \# \text{ cycles} / \# \text{ instrucciones}$$

En 1987 realizaron un pipeline de 24 etapas permitiendo así una frecuencia de funcionamiento de 3.1 GHz

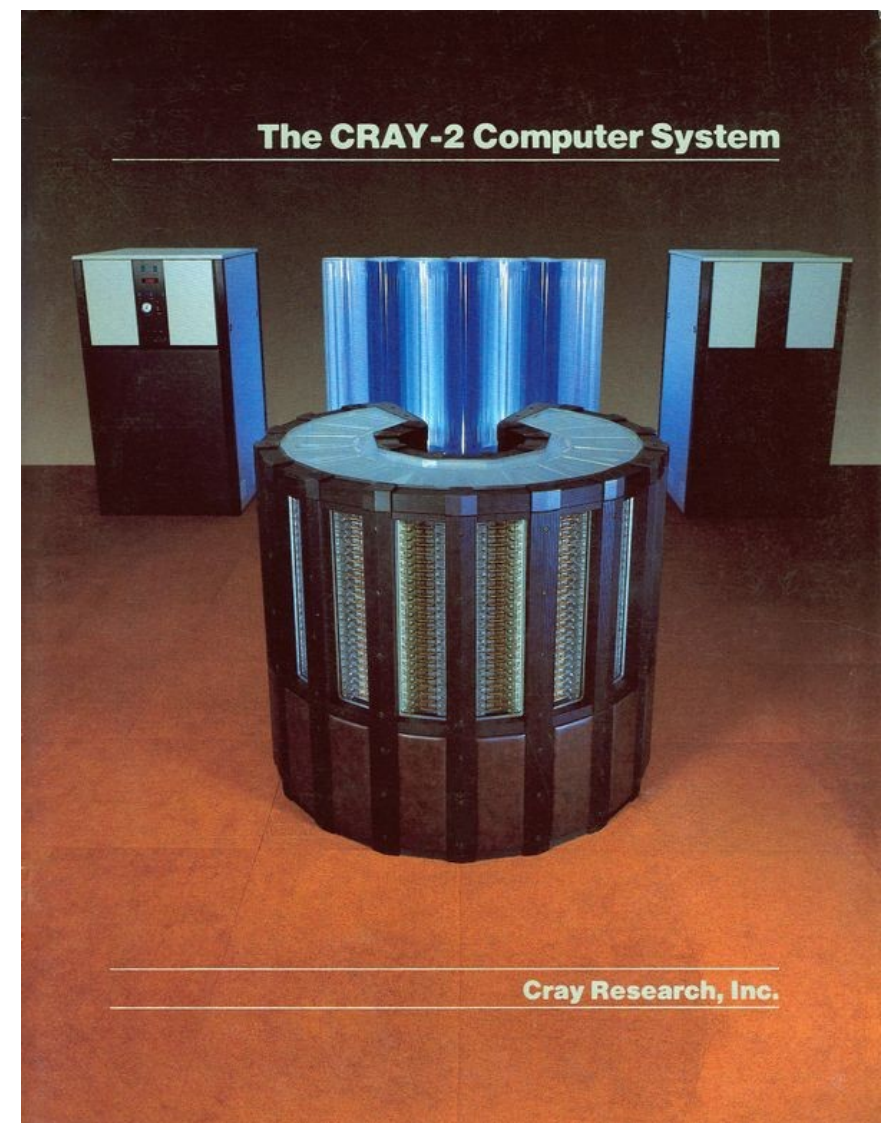
Super pipeline / Pipeline profond

Este procesador se llamaba CRAY nombrado por su inventor, una máquina especializada en cálculos científicos. Más específicamente, cálculos vectorial.

Habían dos de estas máquinas en Francia, utilizadas principalmente para cálculos del clima, debido a las operaciones enormes con matrices.

Estas máquinas eran superenfriadas debido a que la velocidad de los electrones aumentan de manera cuadrática con respecto a la temperatura.

Es decir que si la temperatura bajaba a la mitad, la velocidad aumentaba en cuatro.



Desventajas Super Pipeline

Dependance de données

Stall cycles (Cycles de gel) aumentan

CPI aumenta 

Performance Perf disminuye 

Número de Bypass aumenta

Se agrega más HW (en este caso, multiplexores)

Frecuencia F disminuye 

Desventajas Super Pipeline

Branching

Delayed slots después del branch debido a la profundidad del pipeline

Loi d'Amdahl : $\text{Gain_eff} = \text{Gain_theo} * (\text{taux d'utilisation})$

CPIutil 

BNE

NOP (75% de posibilidad que el compilador lo remplace poir una instrucción útil)

NOP (5%)

...

Franglaispagnol

Dérouler cycles **for**

Minimiza el aumento de stall cycles

3 fuentes de branchements

Boucle **for** (on peut le dérouler)

Boucle **while** (Essayer de les supprimer)

Condition **if** (Essayer de les supprimer)

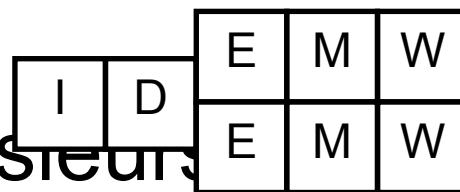
El compilador no va a decir que está mal usar **for**, **while** e **if** debido a que forman parte del language de programación.

Sin embargo, el programador debe saber que es mejor evitar usar estas fuentes de branching debido a que alentan la ejecución del programa.

Podemos mejorar el CPI al aumentar el número de instrucciones ejecutadas por ciclo

Esto se puede lograr al aumentar el número de pipelines (ojo, esto no es un verdadero multinúcleo)

Conocidos como procesadores **Super Scalare** (plusieurs scales = plusieurs pipelines)

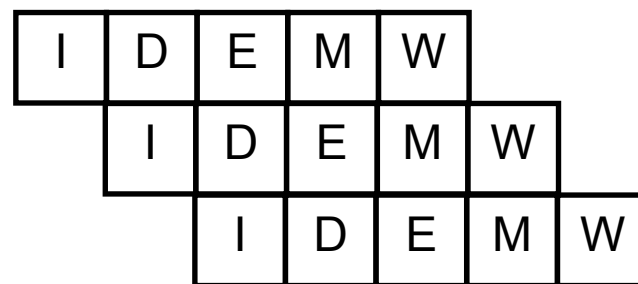


Plusieurs coeurs > Plusieurs Scalaires > Plusieurs Pipelines > Plusieurs étages > Plusieurs instructions

Multiplicaciones

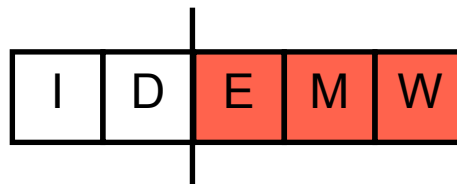
Las multiplicaciones no pueden ser ejecutadas porque son mucho más complejas que simples sumas

MULT Rs,
Rt



32
additions

En la arquitectura MIPS el pipeline de ejecución es de 3 ciclos



Multiplicaciones

Para las multiplicaciones existe un pipeline exclusivo a este tipo de operaciones



Para las multiplicaciones existe un pipeline dedicado

```
for(int i=0; i < size; i++){
```

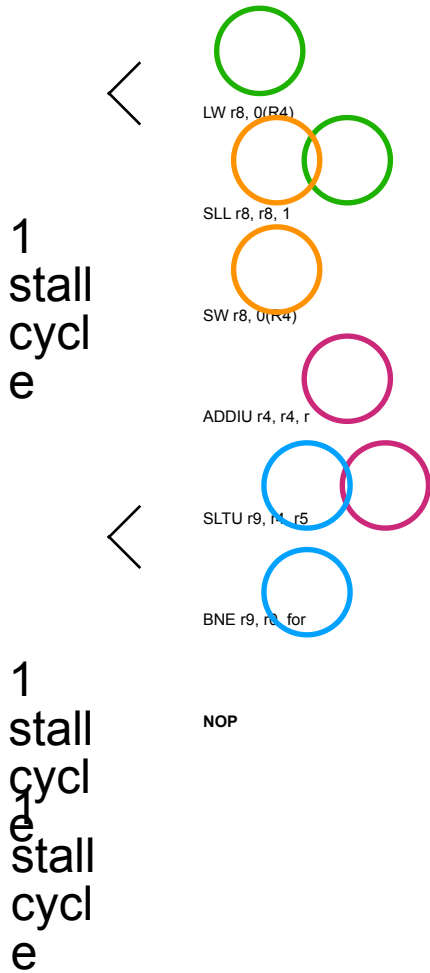
```
    v[i] = 2*v[i]
```

```
}
```

```
size = @fin
```

```
v = R4
```

for:

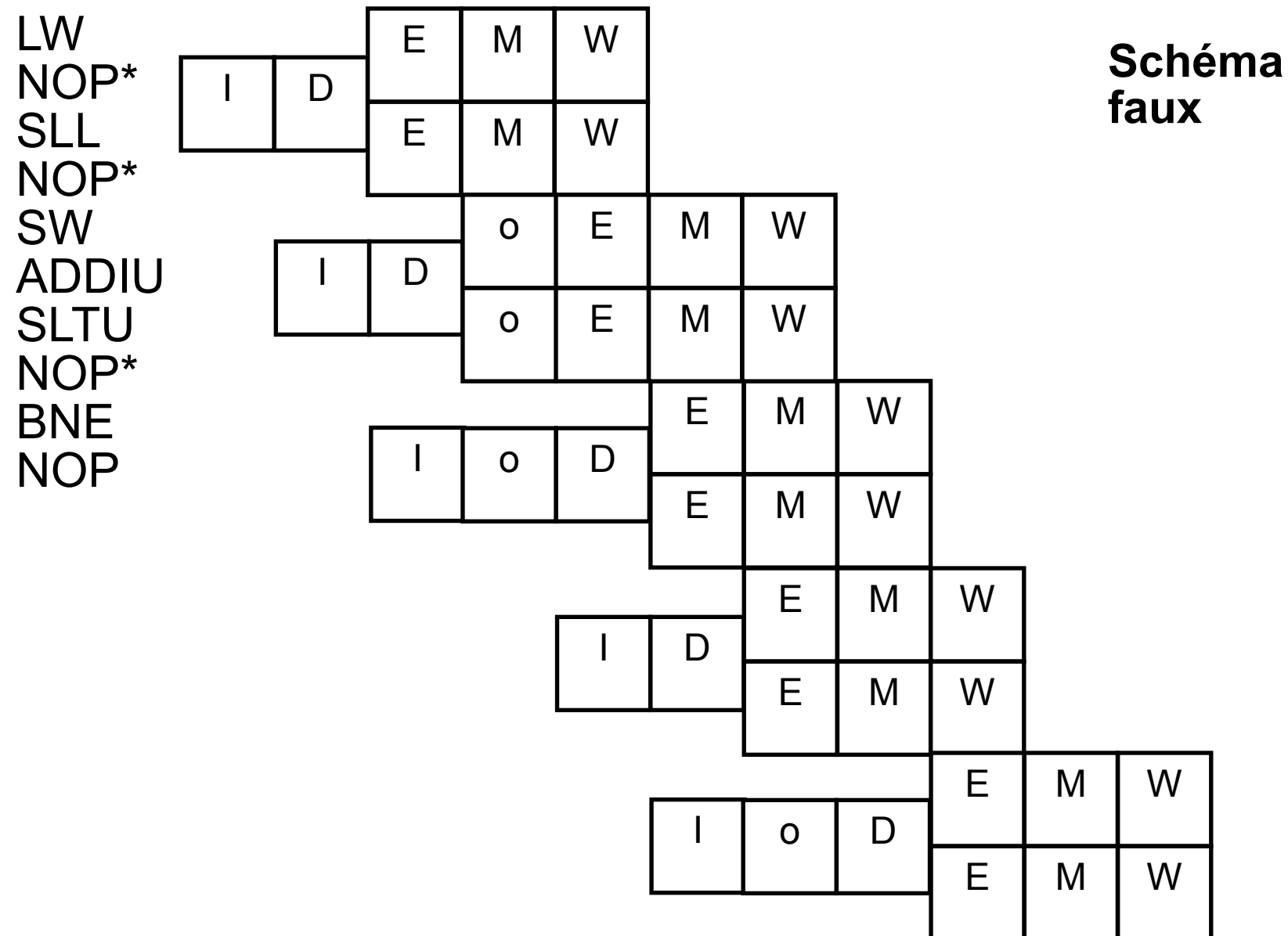


CPIutil = 9 cycles / 6 instructions

CPIutil = 1.5

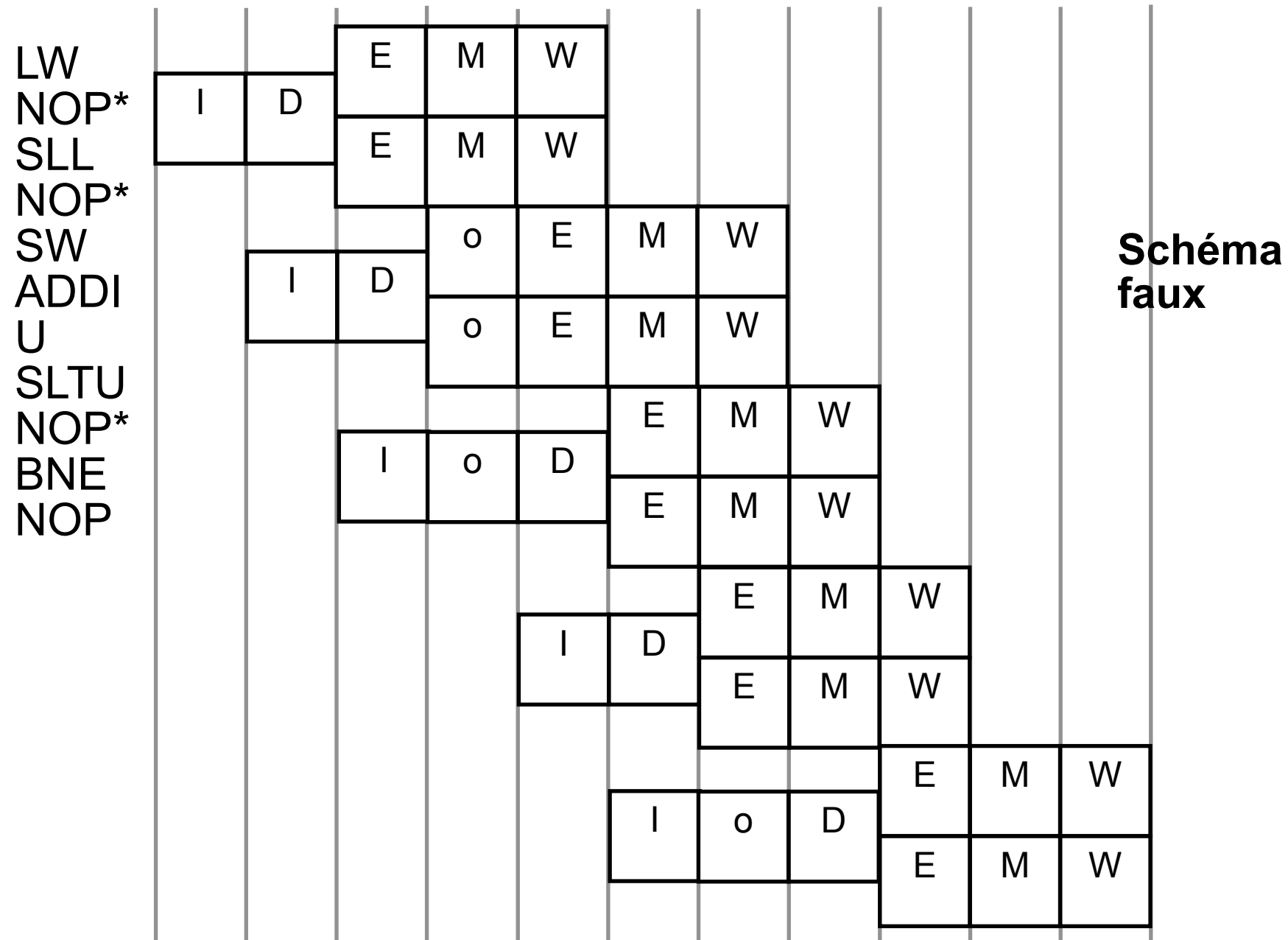
Super Scalaire

Execution dans l'ordre (simple)



Super Scalaire

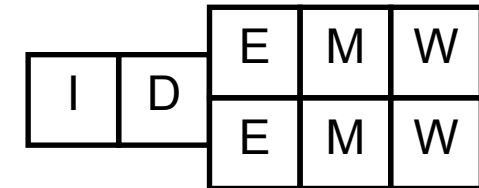
Execution dans l'ordre (simple)



Super Scalaire

Bypasses

Bypasses en **pipeline normal**



2 oper (Rs, Rt)

2 producteur (produisent les donnés) (EXE - MEM)

2 consommateur (consomment les donnés) (DEC, EXE)

$2 \times 2 \times 2 = 8$ **bypasses**

Bypasses en **pipeline super scalaire**

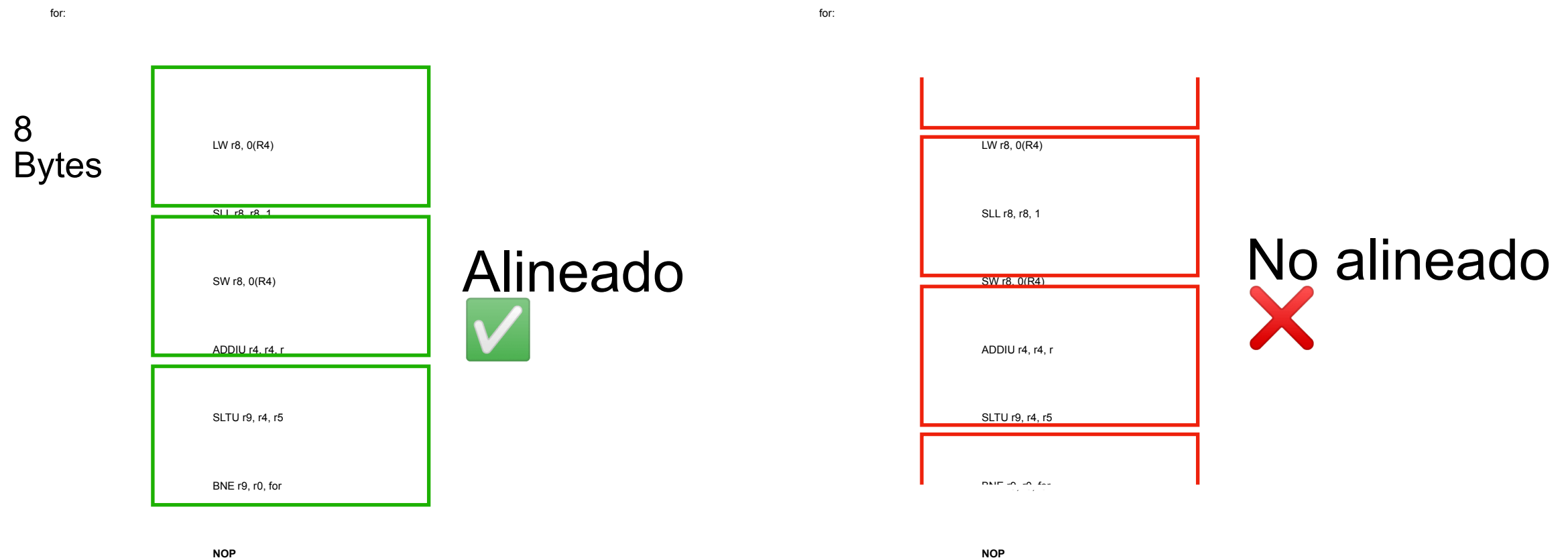
2 oper (Rs, Rt)

2 x 2 producteur (EXE, EXE, MEM, MEM)

2 x 2 consommateur (DEC, DEC, EXE, EXE)

$2 \times (2 \times 2) \times (2 \times 2) = 32$ **bypasses**

Alignment: L'adresse doit être **multiple de la taille**



Las instrucciones no ejecutadas deben de ser guardadas en el **buffer de instrucciones** para que no se pierdan

Buffer d'Instructions

Le nombre de places minimum dans le buffer d'instructions doit être minimum le **2 Nb de pipeline**

Ejemplo de comida

IF est aller faire les courses

En el refri DEC no puede comer la instrucción hasta que esté metida en el refri (buffer d'instruction)

La comida sólo puede ser organizada de 2 en 2 en el refri

No podemos ir de compras hasta que el refri esté vacío

Execution dans l'ordre (simple)

