

# Exercicis 1.6 i 1.7

## 1.6

Afegiu tantes instruccions “NOOP” com creieu al codi següent per tal que pugui funcionar en un pipeline de 5 etapes del processador RISC-V sense “Forward Unit”.

Recordeu que tenim 32 registres que van des de l’x0 fins a l’x31.

## 1.6: Codi

|                   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------------------|---|---|---|---|---|---|---|---|
| addi x11, x12, 5  |   |   |   |   |   |   |   |   |
| add x13, x11, x12 |   |   |   |   |   |   |   |   |
| addi x14, x11, 15 |   |   |   |   |   |   |   |   |
| add x15, x11, x11 |   |   |   |   |   |   |   |   |

## 1.6: Codi

|                   | 1  | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------------------|----|---|---|---|---|---|---|---|
| addi x11, x12, 5  | IF |   |   |   |   |   |   |   |
| add x13, x11, x12 |    |   |   |   |   |   |   |   |
| addi x14, x11, 15 |    |   |   |   |   |   |   |   |
| add x15, x11, x11 |    |   |   |   |   |   |   |   |

## 1.6: Codi

|                   | 1  | 2  | 3 | 4 | 5 | 6 | 7 | 8 |
|-------------------|----|----|---|---|---|---|---|---|
| addi x11, x12, 5  | IF | ID |   |   |   |   |   |   |
| add x13, x11, x12 |    | IF |   |   |   |   |   |   |
| addi x14, x11, 15 |    |    |   |   |   |   |   |   |
| add x15, x11, x11 |    |    |   |   |   |   |   |   |

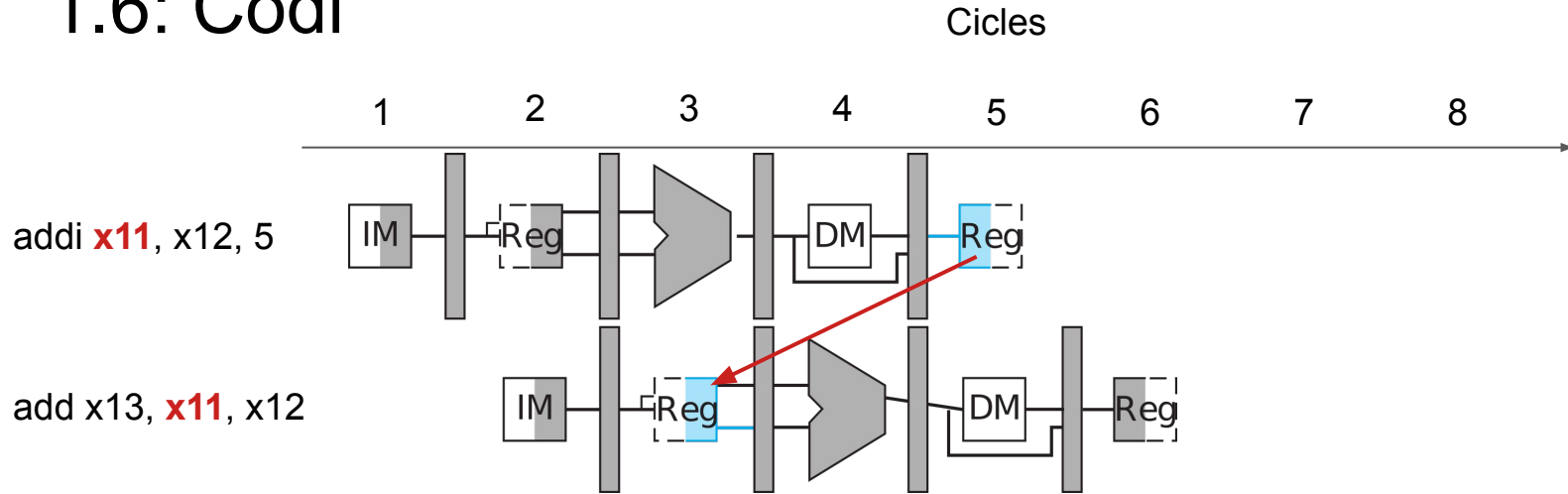
## 1.6: Codi

|                   | 1  | 2  | 3  | 4 | 5 | 6 | 7 | 8 |
|-------------------|----|----|----|---|---|---|---|---|
| addi x11, x12, 5  | IF | ID | EX |   |   |   |   |   |
| add x13, x11, x12 |    | IF | ID |   |   |   |   |   |
| addi x14, x11, 15 |    |    | IF |   |   |   |   |   |
| add x15, x11, x11 |    |    |    |   |   |   |   |   |

## 1.6: Codi

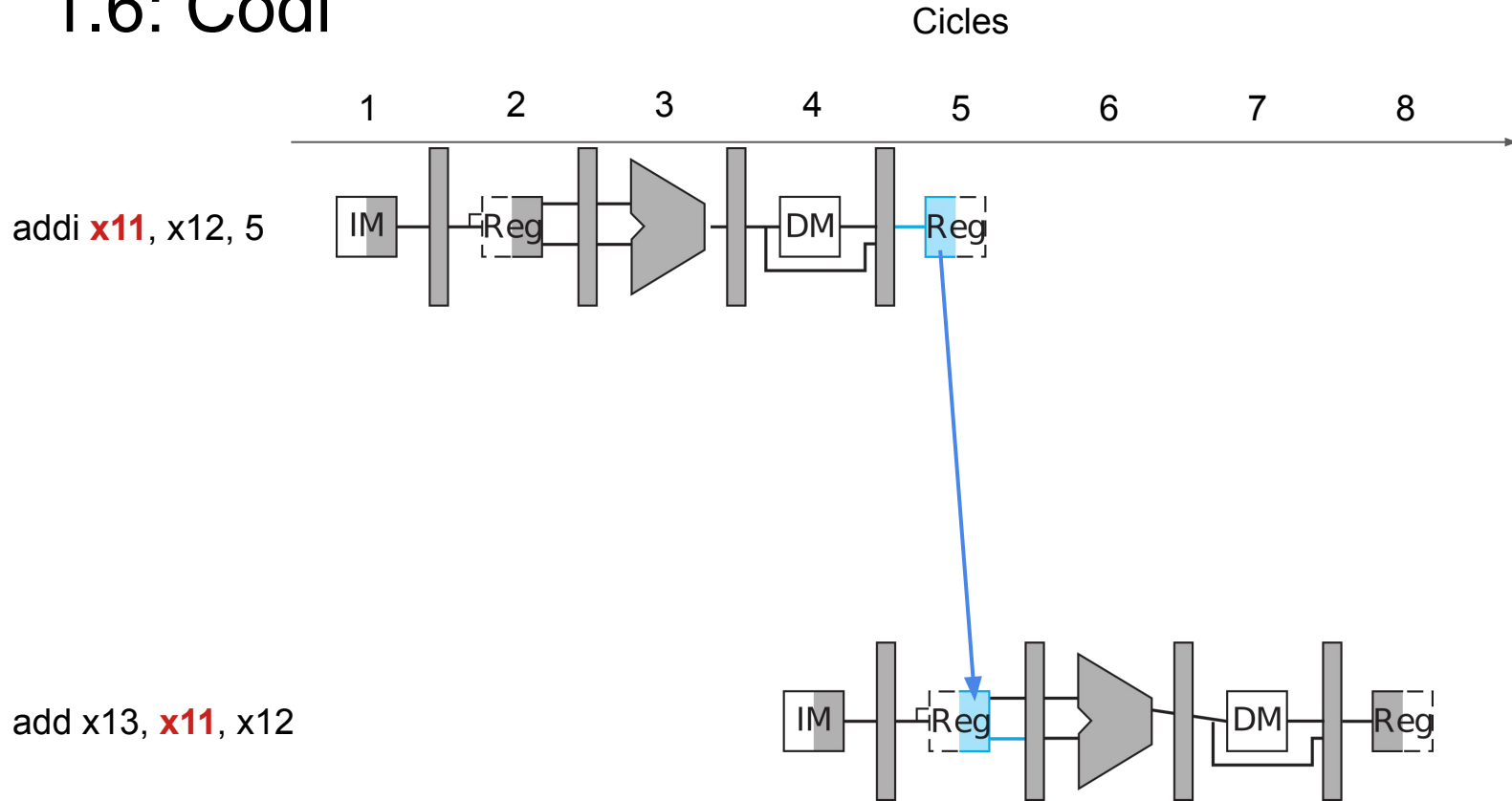
|                           | 1  | 2  | 3  | 4                            | 5  | 6 | 7 | 8 |
|---------------------------|----|----|----|------------------------------|----|---|---|---|
| addi <b>x11</b> , x12, 5  | IF | ID | EX | MEM                          | WB |   |   |   |
| add x13, <b>x11</b> , x12 |    | IF | ID | <b>RAW: Read After Write</b> |    |   |   |   |
| addi x14, x11, 15         |    |    | IF |                              |    |   |   |   |
| add x15, x11, x11         |    |    |    |                              |    |   |   |   |

# 1.6: Codi

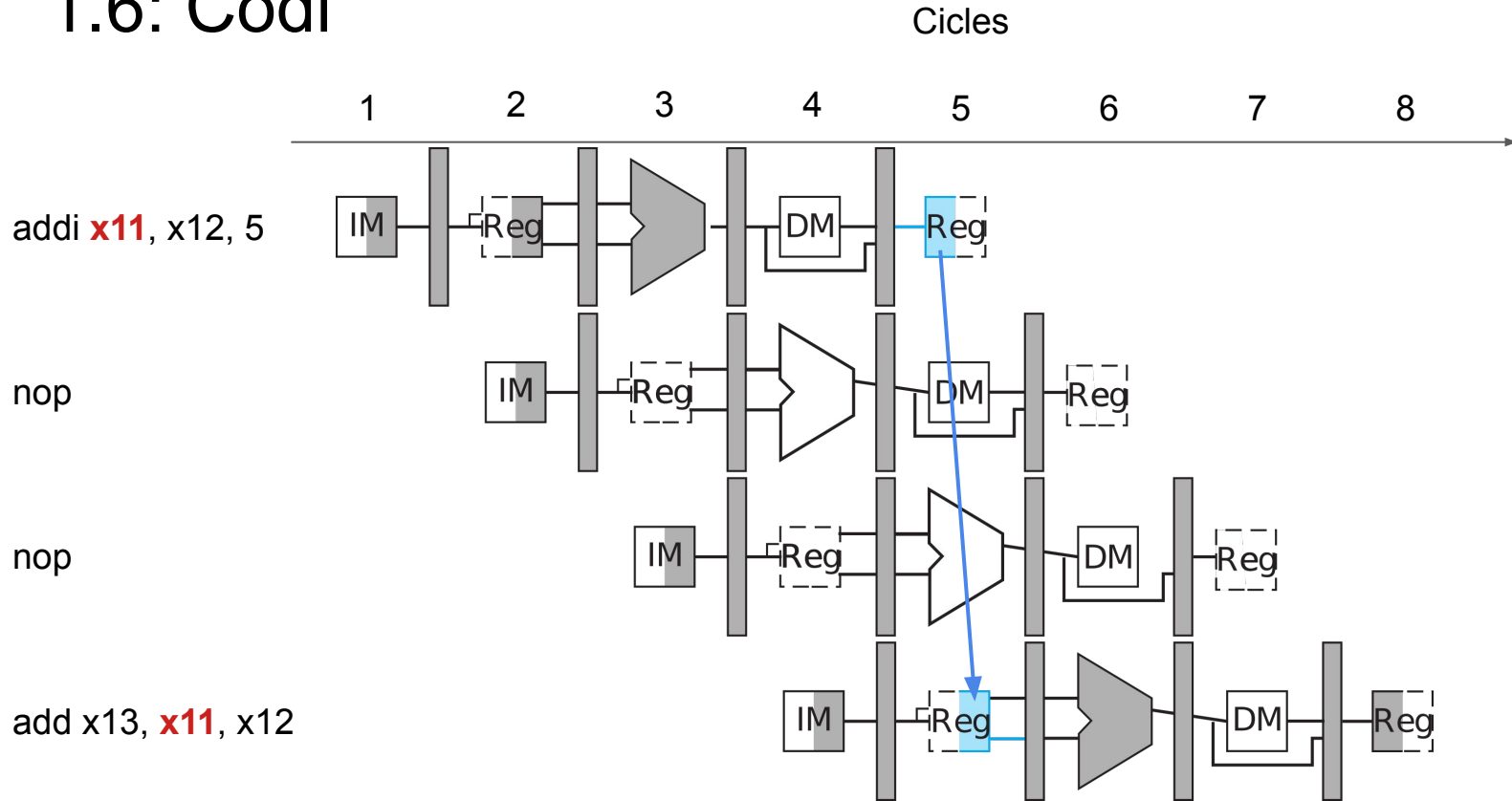




# 1.6: Codi

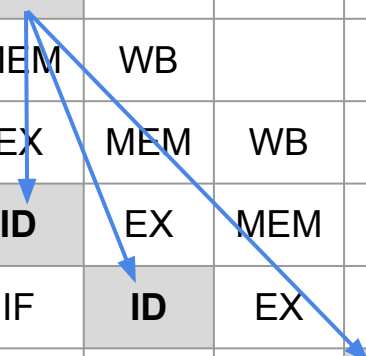


# 1.6: Codi



## 1.6: Codi

|                                  | 1  | 2  | 3  | 4   | 5         | 6         | 7   | 8         | 9  | 10  | 11 |
|----------------------------------|----|----|----|-----|-----------|-----------|-----|-----------|----|-----|----|
| addi <b>x11</b> , x12, 5         | IF | ID | EX | MEM | <b>WB</b> |           |     |           |    |     |    |
| <b>nop</b>                       |    | IF | ID | EX  | MEM       | WB        |     |           |    |     |    |
| <b>nop</b>                       |    |    | IF | ID  | EX        | MEM       | WB  |           |    |     |    |
| add x13, <b>x11</b> , x12        |    |    |    | IF  | <b>ID</b> | EX        | MEM | WB        |    |     |    |
| addi x14, <b>x11</b> , 15        |    |    |    |     | IF        | <b>ID</b> | EX  | MEM       | WB |     |    |
| add x15, <b>x11</b> , <b>x11</b> |    |    |    |     |           |           | IF  | <b>ID</b> | EX | MEM | WB |



## 1.7

En un processador RISC-V de 5 etapes examinem com afecta el pipeline al temps de cicle del rellotge del processador. Les qüestions d'aquest exercici suposen que cada etapa triga un temps diferent en executar-se, en particular els temps d'execució de cada etapa són els següents:

| <b>IF</b> | <b>ID</b> | <b>EX</b> | <b>MEM</b> | <b>WB</b> |
|-----------|-----------|-----------|------------|-----------|
| 250 ps    | 350 ps    | 150 ps    | 300 ps     | 200 ps    |

Suposeu també que les instruccions executades pel processador es desglossen de la manera següent:

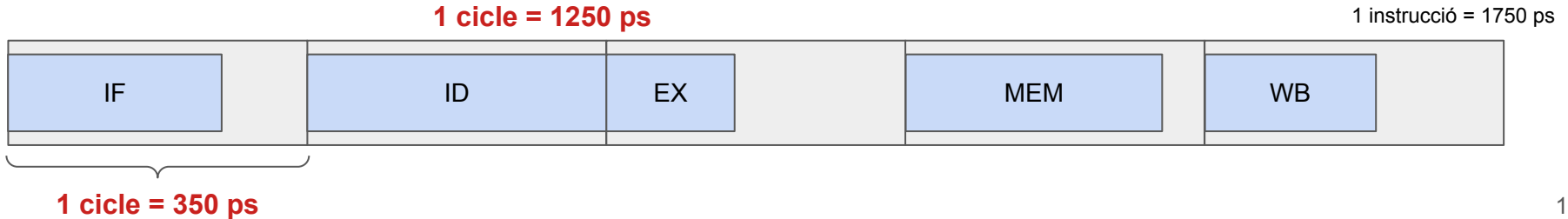
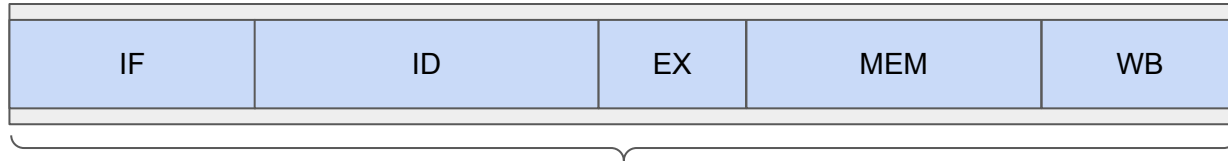
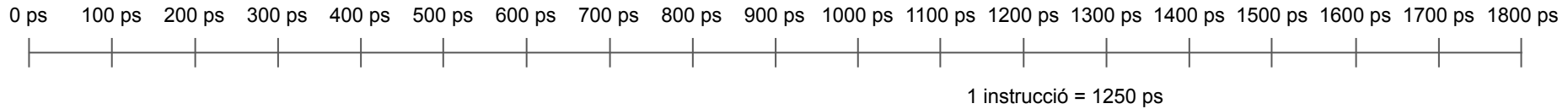
| <b>ALU/Logic</b> | <b>Jump/Branch</b> | <b>Load</b> | <b>Store</b> |
|------------------|--------------------|-------------|--------------|
| 45%              | 20%                | 20%         | 15%          |

## 1.7. Pregunta 1: Mida del cicle de rellotge

Quin és el temps de cicle de rellotge mínim (en ps) en un processador amb pipeline i en un sense pipeline?

# 1.7. Pregunta 1: Mida del cicle de rellotge

Quin és el temps de cicle de rellotge mínim (en ps) en un processador amb pipeline i en un sense pipeline?

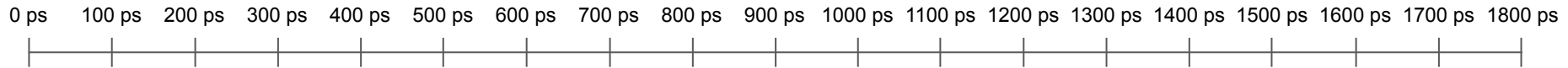


## 1.7. Pregunta 2: temps d'execució d'un load

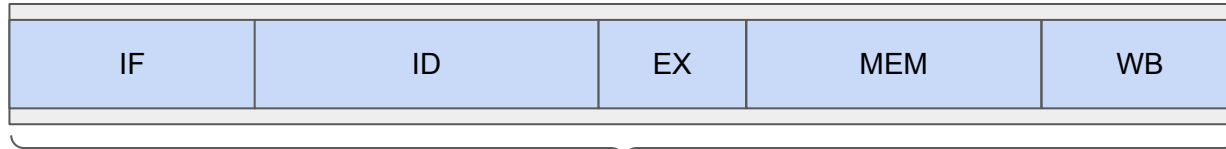
Quant temps triga en executar-se una instrucció de tipus Load en un processador amb pipeline i en un sense pipeline?

## 1.7. Pregunta 2: temps d'execució d'un load

Quant temps triga en executar-se una instrucció de tipus Load en un processador amb pipeline i en un sense pipeline?



**1 instrucció = 1250 ps**



1 cycle = 1250 ps

**1 instrucció = 1750 ps**



1 cycle = 350 ps

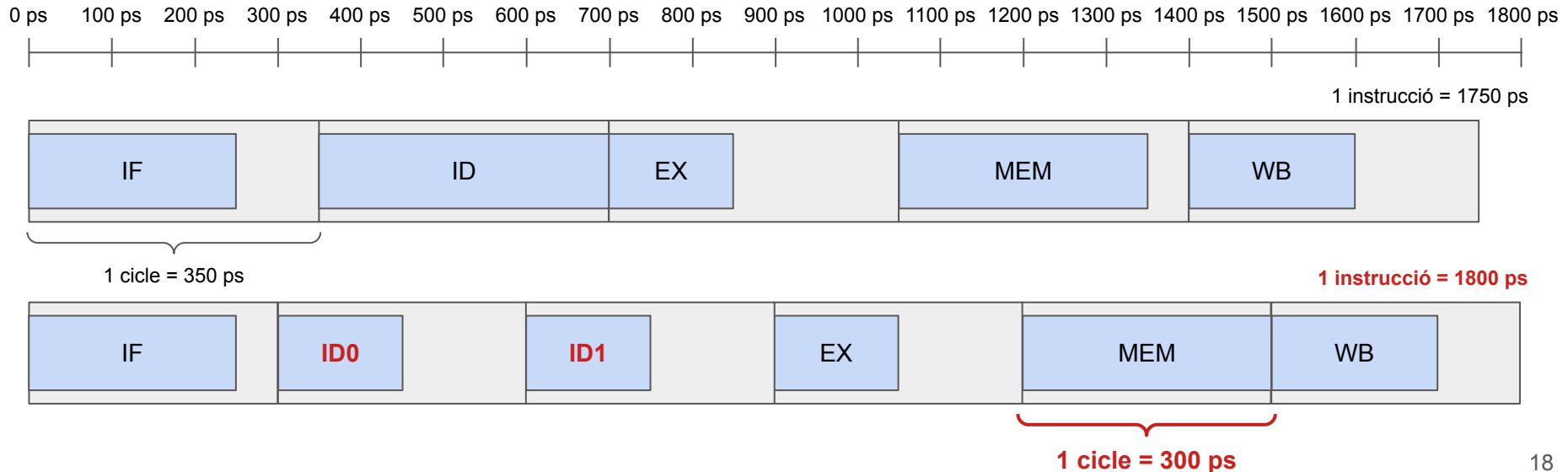


## 1.7. Pregunta 3: subdividir una etapa

Si podem dividir una etapa del pipeline en dues noves etapes, cadascuna trigarà la meitat de temps en executar-se que l'etapa original, quina etapa dividiríeu i quin és el nou temps de cicle del rellotge del processador amb pipeline?

## 1.7. Pregunta 3: subdividir una etapa

Si podem dividir una etapa del pipeline en dues noves etapes, cadascuna trigarà la meitat de temps en executar-se que l'etapa original, quina etapa dividiríeu i quin és el nou temps de cicle del rellotge del processador amb pipeline?



## 1.7. Pregunta 4: temps d'execució d'un load

Suposant que no hi ha “stalls” ni “hazards”, quina és percentatge d'utilització de la memòria de dades per a les instruccions?

## 1.7. Pregunta 4: temps d'execució d'un load

Suposant que no hi ha “stalls” ni “hazards”, quina és percentatge d'utilització de la memòria de dades per a les instruccions?

La memòria de dades s'utilitza en les instruccions de load i store:

- Load: 20% de les instruccions
- Store: 15% de les instruccions

Quan no s'executa cap d'aquestes instruccions, la memòria no es fa servir. Per tant, només s'usa durant el 35% del temps.

## 1.7. Pregunta 5: temps d'execució d'un load

Suposant que no hi ha “stalls” ni “hazards”, quina és percentatge d'utilització del port d'escriptura dels registres per a les instruccions executades pel processador amb pipeline?

## 1.7. Pregunta 5: temps d'execució d'un load

Suposant que no hi ha “stalls” ni “hazards”, quina és percentatge d'utilització del port d'escriptura dels registres per a les instruccions executades pel processador amb pipeline?

Les instruccions que desen el resultat en un registre són:

- ALU/Logic: 45% de les instruccions
- Load: 20% de les instruccions

Quan no s'executa cap d'aquestes instruccions, el port d'escriptura dels registres s'ignora. Per tant, s'usa durant el 65% del temps.