



SAPIENZA  
UNIVERSITÀ DI ROMA  
DIPARTIMENTO DI INFORMATICA

# Architettura degli Elaboratori 16 – Esercizi su MIPS e pipeline

Prof. Andrea Sterbini – [sterbini@di.uniroma1.it](mailto:sterbini@di.uniroma1.it)



# Argomenti

- Esercizi di esame
- ▶ **Esercizio (16-9-14)**

```
.data
vettore: # qui ci sono 70 pari e 30 dispari

.text
main:
    li    $t0, 0           # somma parz.
    li    $t1, 396        # offset ultimo

ciclo:
    lw    $t3, vettore($t1) #
    andi  $t2, $t3, 1     # è dispari?
    beqz  $t2, salta      # lo ignoro
    add   $t0, $t0, $t3    # o lo sommo

salta:  addi  $t1, $t1, -4  # prox. elem.
    bge   $t1, $zero, ciclo # ho finito?
    li    $v0, 1          # stampo
    move  $a0, $t0        # la somma

syscall #
```

# Argomenti

- Esercizi di esame
- ▶ **Esercizio (16-9-14)**
- Individuate i **data** e **control hazard**

```
.data
vettore: # qui ci sono 70 pari e 30 dispari

.text
main:
    li    $t0, 0           # somma parz.
    li    $t1, 396        # offset ultimo

ciclo:
    lw    $t3, vettore($t1) #
    andi  $t2, $t3, 1     # è dispari?
    beqz  $t2, salta      # lo ignoro
    add   $t0, $t0, $t3    # o lo sommo
salta:  addi $t1, $t1, -4  # prox. elem.
    bge  $t1, $zero, ciclo # ho finito?
    li   $v0, 1           # stampo
    move $a0, $t0         # la somma
    syscall                #
```

# Argomenti

- Esercizi di esame
- ▶ **Esercizio (16-9-14)**
- Individuate i **data** e **control hazard**
- Individuate i casi in cui il **forwarding** è applicabile

```
.data
vettore: # qui ci sono 70 pari e 30 dispari

.text
main:
    li    $t0, 0           # somma parz.
    li    $t1, 396         # offset ultimo

ciclo:
    lw    $t3, vettore($t1) #
    andi  $t2, $t3, 1      # è dispari?
    beqz  $t2, salta       # lo ignoro
    add   $t0, $t0, $t3    # o lo sommo

salta:  addi $t1, $t1, -4   # prox. elem.
    bge  $t1, $zero, ciclo # ho finito?
    li   $v0, 1           # stampo
    move $a0, $t0         # la somma

syscall #
```

# Argomenti

- Esercizi di esame
- ▶ **Esercizio (16-9-14)**
- Individuate i **data** e **control hazard**
- Individuate i casi in cui il **forwarding** è applicabile
- Individuate gli **stalli necessari**

```
.data
vettore: # qui ci sono 70 pari e 30 dispari

.text
main:
    li    $t0, 0           # somma parz.
    li    $t1, 396        # offset ultimo

ciclo:
    lw    $t3, vettore($t1) #
    andi  $t2, $t3, 1     # è dispari?
    beqz  $t2, salta     # lo ignoro
    add   $t0, $t0, $t3   # o lo sommo

salta:  addi  $t1, $t1, -4 # prox. elem.
    bge  $t1, $zero, ciclo # ho finito?
    li   $v0, 1          # stampo
    move $a0, $t0        # la somma

syscall #
```

# Argomenti

- Esercizi di esame
- ▶ **Esercizio (16-9-14)**
- Individuate i **data** e **control hazard**
- Individuate i casi in cui il **forwarding è applicabile**
- Individuate gli **stalli necessari**
- Calcolate la **durata totale** in cicli di clock

```
.data
vettore: # qui ci sono 70 pari e 30 dispari

.text
main:
    li    $t0, 0           # somma parz.
    li    $t1, 396         # offset ultimo

ciclo:
    lw    $t3, vettore($t1) #
    andi  $t2, $t3, 1      # è dispari?
    beqz  $t2, salta       # lo ignoro
    add   $t0, $t0, $t3    # o lo sommo

salta:  addi  $t1, $t1, -4  # prox. elem.
    bge  $t1, $zero, ciclo # ho finito?
    li   $v0, 1           # stampo
    move $a0, $t0         # la somma

syscall #
```

# Argomenti

- Esercizi di esame
- ▶ **Esercizio (16-9-14)**
- Individuate i **data** e **control hazard**
- Individuate i casi in cui il **forwarding** è applicabile
- Individuate gli **stalli necessari**
- Calcolate la **durata totale** in cicli di clock
- Calcolate la **durata senza forwarding**

```
.data
vettore: # qui ci sono 70 pari e 30 dispari

.text
main:
    li    $t0, 0           # somma parz.
    li    $t1, 396        # offset ultimo

ciclo:
    lw    $t3, vettore($t1) #
    andi  $t2, $t3, 1     # è dispari?
    beqz  $t2, salta     # lo ignoro
    add   $t0, $t0, $t3   # o lo sommo

salta:  addi  $t1, $t1, -4 # prox. elem.
    bge  $t1, $zero, ciclo # ho finito?
    li   $v0, 1           # stampo
    move $a0, $t0         # la somma

syscall #
```

# Argomenti

- Esercizi di esame
- ▶ **Esercizio (16-9-14)**
- Individuate i **data** e **control hazard**
- Individuate i casi in cui il **forwarding è applicabile**
- Individuate gli **stalli necessari**
- Calcolate la **durata totale** in cicli di clock
- Calcolate la **durata senza forwarding**
- **Riordinate le istruzioni** per ridurre gli stalli

```
.data
vettore: # qui ci sono 70 pari e 30 dispari

.text
main:
    li    $t0, 0           # somma parz.
    li    $t1, 396        # offset ultimo

ciclo:
    lw    $t3, vettore($t1) #
    andi  $t2, $t3, 1     # è dispari?
    beqz  $t2, salta     # lo ignoro
    add   $t0, $t0, $t3   # o lo sommo

salta:  addi  $t1, $t1, -4 # prox. elem.
    bge  $t1, $zero, ciclo # ho finito?
    li   $v0, 1           # stampo
    move $a0, $t0         # la somma

syscall #
```



# Argomenti

- Esercizi di esame
- ▶ **Esercizio (16-9-14)**
- Individuate i **data** e **control hazard**
- Individuate i casi in cui il **forwarding è applicabile**
- Individuate gli **stalli necessari**
- Calcolate la **durata totale** in cicli di clock
- Calcolate la **durata senza forwarding**
- **Riordinate le istruzioni** per ridurre gli stalli
- Calcolate la **durata** del prog. **ottimizzato**

```
.data
vettore: # qui ci sono 70 pari e 30 dispari

.text
main:
    li    $t0, 0           # somma parz.
    li    $t1, 396        # offset ultimo

ciclo:
    lw    $t3, vettore($t1) #
    andi  $t2, $t3, 1     # è dispari?
    beqz  $t2, salta     # lo ignoro
    add   $t0, $t0, $t3   # o lo sommo

salta:  addi  $t1, $t1, -4 # prox. elem.
    bge  $t1, $zero, ciclo # ho finito?
    li   $v0, 1           # stampo
    move $a0, $t0         # la somma

syscall #
```

# Argomenti

- Esercizi di esame
- ▶ **Esercizio (16-9-14)**
- Individuate i **data** e **control hazard**
- Individuate i casi in cui il **forwarding è applicabile**
- Individuate gli **stalli necessari**
- Calcolate la **durata totale** in cicli di clock
- Calcolate la **durata senza forwarding**
- **Riordinate le istruzioni** per ridurre gli stalli
- Calcolate la **durata** del prog. **ottimizzato**
- **Cosa c'è in pipeline a**

▶  $t=16$ ?

```
.data
vettore: # qui ci sono 70 pari e 30 dispari

.text
main:
    li    $t0, 0           # somma parz.
    li    $t1, 396        # offset ultimo

ciclo:
    lw    $t3, vettore($t1) #
    andi  $t2, $t3, 1     # è dispari?
    beqz  $t2, salta     # lo ignoro
    add   $t0, $t0, $t3   # o lo sommo

salta:  addi  $t1, $t1, -4 # prox. elem.
    bge  $t1, $zero, ciclo # ho finito?
    li   $v0, 1           # stampo
    move $a0, $t0        # la somma

syscall #
```

# Argomenti

- Esercizi di esame
- ▶ **Esercizio (16-9-14)**
- Individuate i **data** e **control hazard**
- Individuate i casi in cui il **forwarding è applicabile**
- Individuate gli **stalli necessari**
- Calcolate la **durata totale** in cicli di clock
- Calcolate la **durata senza forwarding**
- **Riordinate le istruzioni** per ridurre gli stalli
- Calcolate la **durata** del prog. **ottimizzato**
- **Cosa c'è in pipeline a**

▶  $t^2=16?$

```
.data
vettore: # qui ci sono 70 pari e 30 dispari

.text
main:
    li    $t0, 0           # somma parz.
    li    $t1, 396         # offset ultimo

ciclo:
    lw    $t3, vettore($t1) #
    andi  $t2, $t3, 1      # è dispari?
    beqz  $t2, salta       # lo ignoro
    add   $t0, $t0, $t3     # o lo sommo
salta:  addi $t1, $t1, -4   # prox. elem.
    bge   $t1, $zero, ciclo # ho finito?
    li    $v0, 1           # stampo
    move  $a0, $t0         # la somma
    syscall                #
```

# Forwarding e stalli

```

.data
vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

.text
main:
    li    $t0, 0                # F D E M W (5°)
    li    $t1, 396            #   F D E M W (6°)

ciclo:
    lw    $t3, vettore($t1) #   F D E M W (7°)           F D E M W 17°
    andi  $t2, $t3, 1        #           →F D E M W (9°)       →F D E M W 19°
    beqz  $t2, salta          #           →F D E M W (11°)      →F D E M W 21°
    add   $t0, $t0, $t3        #           F D E M W (12°)
salta:  addi $t1, $t1, -4     #           F D E M W (13°)      →F D E...23°
    bge   $t1, $zero, ciclo  #           →F D E M W (15°)      →F D .25°
    li    $v0, 1                #                                           1006°
    move  $a0, $t0              # ciclo = 30*(6+3+1)+70*(5+3+2)-2        1007°
    syscall                       # totale = 4 + 3 + 998 + 3 = 1008        1008°

```

# Forwarding e stalli

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

li \$t0, 0

li \$t1, 396

**ciclo:**

lw \$t3, vettore(\$t1)

andi \$t2, \$t3, 1

beqz \$t2, salta

add \$t0, \$t0, \$t3

**salta:** addi \$t1, \$t1, -4

bge \$t1, \$zero, ciclo

li \$v0, 1

move \$a0, \$t0

syscall

**4 (riempimento)**

# F D E M W (5°)

# F D **E** M W (6°)

# F D **E** M W (7°)

**F** D E M W 17°

# →F D **E** M W (9°)

→F D **E** M W 19°

# →F D **E** M W (11°)

→F D **E** M W 21°

# F D E M W (12°)

# F D **E** M W (13°)

→F D **E**...23°

# →F D **E** M W (15°)

→F D.25°

# ciclo = 30\*(6+3+1)+70\*(5+3+2)-2

1007°

# totale = 4 + 3 + 998 + 3 = 1008

1008°

# Forwarding e stalli

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

li \$t0, 0

li \$t1, 396

**ciclo:**

lw \$t3, vettore(\$t1)

andi \$t2, \$t3, 1

beqz \$t2, salta

add \$t0, \$t0, \$t3

**salta:** addi \$t1, \$t1, -4

bge \$t1, \$zero, ciclo

li \$v0, 1

move \$a0, \$t0

syscall

**4 (riempimento)**

# F D E M W (5°)	
# F D <b>E</b> M W (6°)	
# F D <b>E</b> M W (7°)	F D E M W 17°
# →F D <b>E</b> M W (9°)	→F D E M W 19°
# →F D E M W (11°)	→F D E M W 21°
# F D E M W (12°)	
# F D <b>E</b> M W (13°)	→F D <b>E</b> ...23°
# →F D E M W (15°)	→F D .25°
#	1006°
# ciclo = 30*(6+3+1)+70*(5+3+2)-2	1007°
# totale = 4 + 3 + 998 + 3 = 1008	1008°

# Forwarding e stalli

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

li \$t0, 0

li \$t1, 396

**ciclo:**

lw \$t3, vettore(\$t1)

andi \$t2, \$t3, 1

beqz \$t2, salta

add \$t0, \$t0, \$t3

**salta:** addi \$t1, \$t1, -4

bge \$t1, \$zero, ciclo

li \$v0, 1

move \$a0, \$t0

syscall

**4 (riempimento)**

# F D E M W (5°)

# F D **E** M W (6°)

# F D **E** **M** W (7°)

# F D E M W 17°

# →F D **E** M W (9°)

# →F D E M W 19°

# →F **D** E M W (11°)

# →F D E M W 21°

# F D E M W (12°)

# F D **E** M W (13°)

# →F D **E**...23°

# →F **D** E M W (15°)

# →F D.25°

# ciclo = 30\*(6+3+1)+70\*(5+3+2)-2

1007°

# totale = 4 + 3 + 998 + 3 = 1008

1008°

# Forwarding e stalli

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

li \$t0, 0

li \$t1, 396

**ciclo:**

lw \$t3, vettore(\$t1)

andi \$t2, \$t3, 1

beqz \$t2, salta

add \$t0, \$t0, \$t3

**salta:** addi \$t1, \$t1, -4

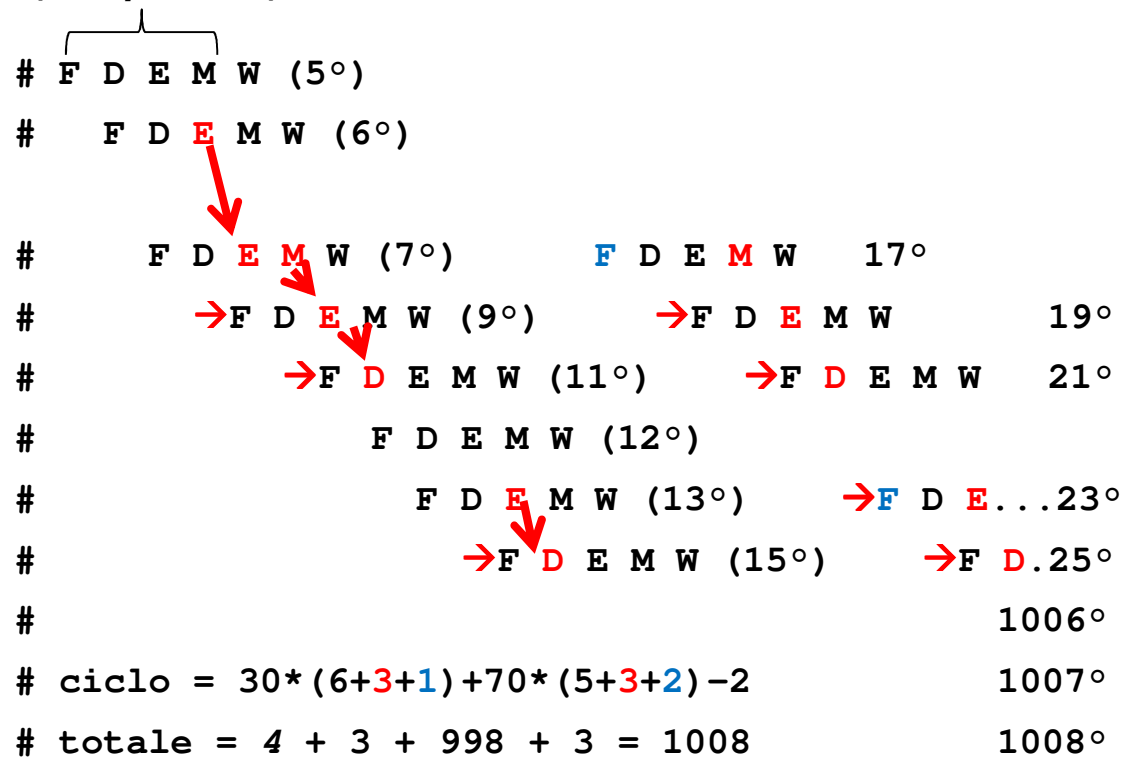
bge \$t1, \$zero, ciclo

li \$v0, 1

move \$a0, \$t0

syscall

**4 (riempimento)**





# Forwarding e stalli

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

li \$t0, 0

li \$t1, 396

**ciclo:**

lw \$t3, vettore(\$t1)

andi \$t2, \$t3, 1

beqz \$t2, salta

add \$t0, \$t0, \$t3

**salta:** addi \$t1, \$t1, -4

bge \$t1, \$zero, ciclo

li \$v0, 1

move \$a0, \$t0

syscall

**4 (riempimento)**

# F D E M W (5°)

# F D **E** M W (6°)

# F D **E** M W (7°)

→ F D **E** M W (9°)

→ F D **E** M W (11°)

F D E M W (12°)

F D **E** M W (13°)

→ F D **E** M W (15°)

# ciclo = 30\*(6+3+1)+70\*(5+3+2)-2

# totale = 4 + 3 + 998 + 3 = 1008

F D E M W 17°

→ F D **E** M W 19°

→ F D **E** M W 21°

→ F D **E**...23°

→ F D .25°

1006°

1007°

1008°

# Forwarding e stalli

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

li \$t0, 0

li \$t1, 396

**ciclo:**

lw \$t3, vettore(\$t1)

andi \$t2, \$t3, 1

beqz \$t2, salta

add \$t0, \$t0, \$t3

**salta:** addi \$t1, \$t1, -4

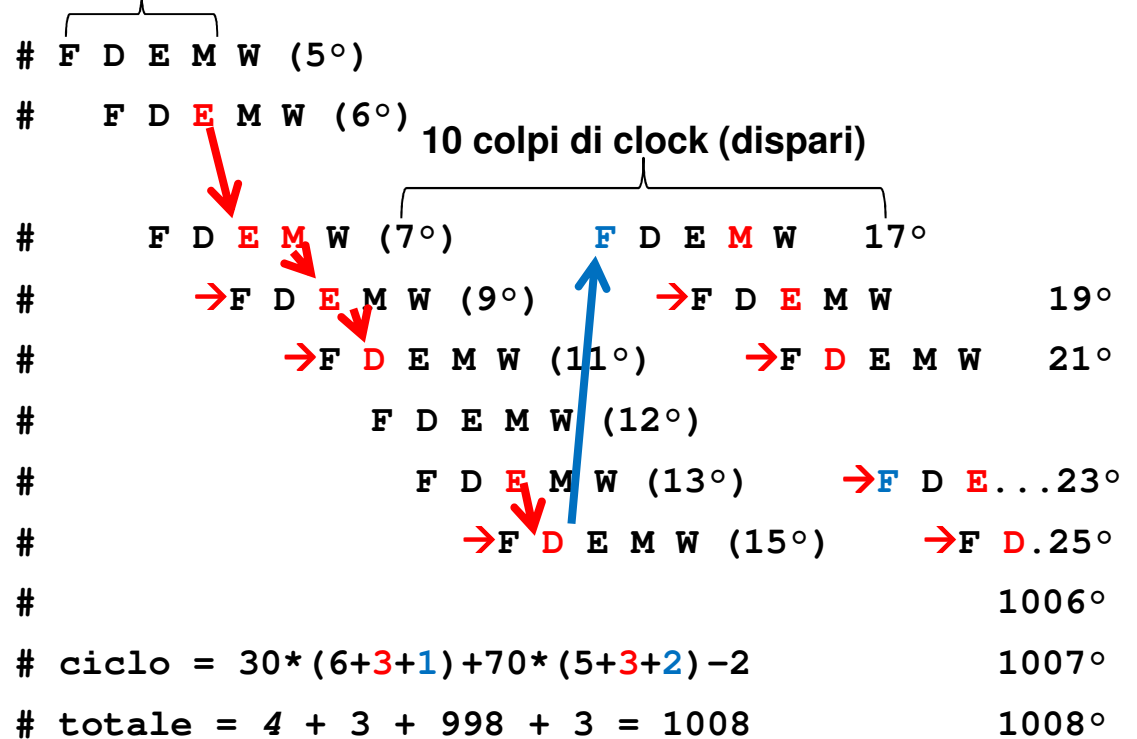
bge \$t1, \$zero, ciclo

li \$v0, 1

move \$a0, \$t0

syscall

**4 (riempimento)**



# Forwarding e stalli

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

li \$t0, 0

li \$t1, 396

**ciclo:**

lw \$t3, vettore(\$t1)

andi \$t2, \$t3, 1

beqz \$t2, salta

add \$t0, \$t0, \$t3

**salta:** addi \$t1, \$t1, -4

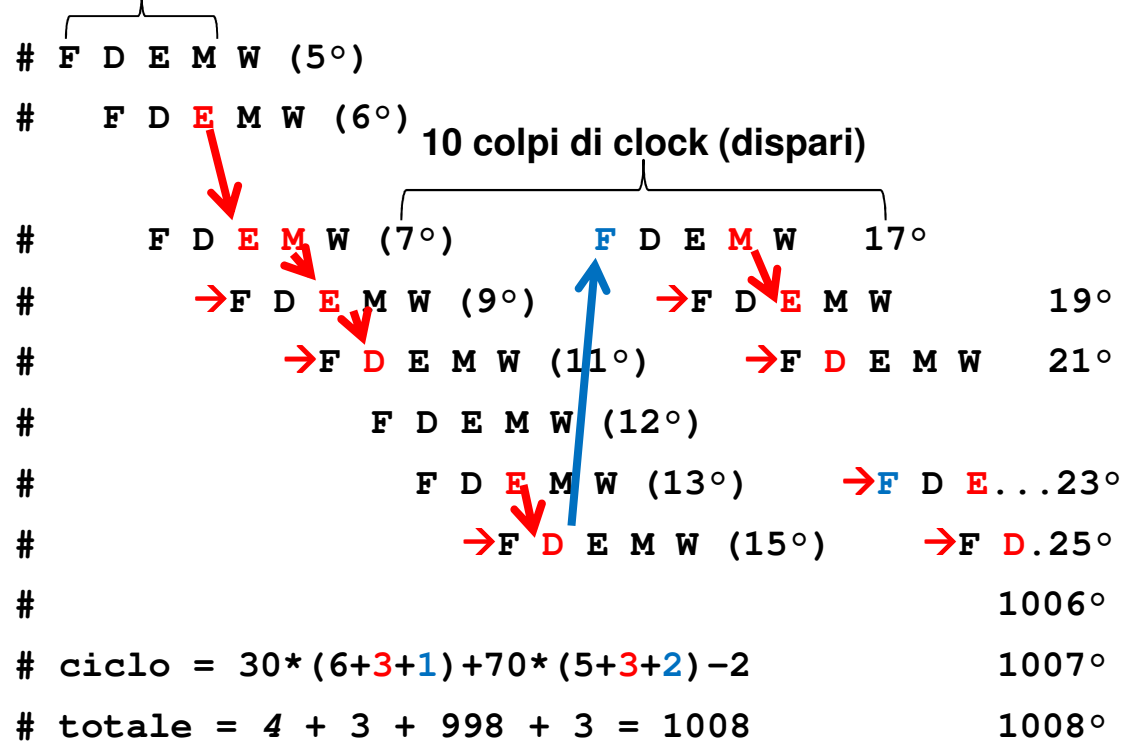
bge \$t1, \$zero, ciclo

li \$v0, 1

move \$a0, \$t0

syscall

**4 (riempimento)**



# Forwarding e stalli

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

li \$t0, 0

li \$t1, 396

**ciclo:**

lw \$t3, vettore(\$t1)

andi \$t2, \$t3, 1

beqz \$t2, salta

add \$t0, \$t0, \$t3

**salta:** addi \$t1, \$t1, -4

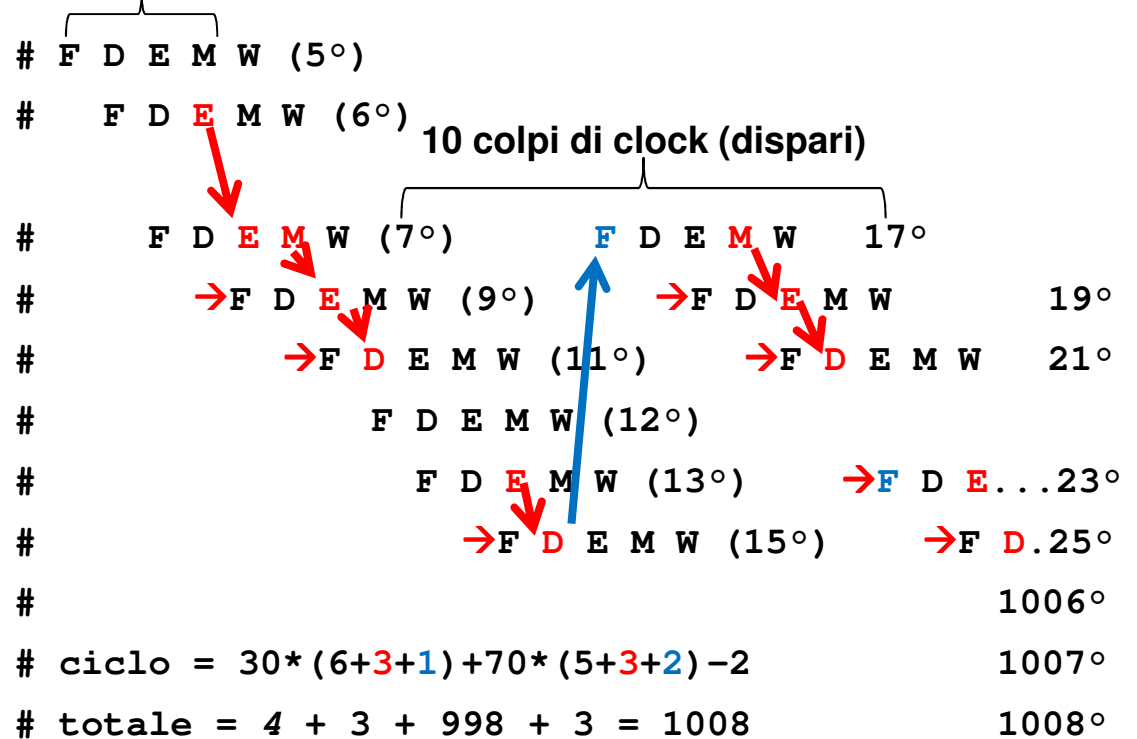
bge \$t1, \$zero, ciclo

li \$v0, 1

move \$a0, \$t0

syscall

**4 (riempimento)**



# Forwarding e stalli

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

li \$t0, 0

li \$t1, 396

**ciclo:**

lw \$t3, vettore(\$t1)

andi \$t2, \$t3, 1

beqz \$t2, salta

add \$t0, \$t0, \$t3

**salta:** addi \$t1, \$t1, -4

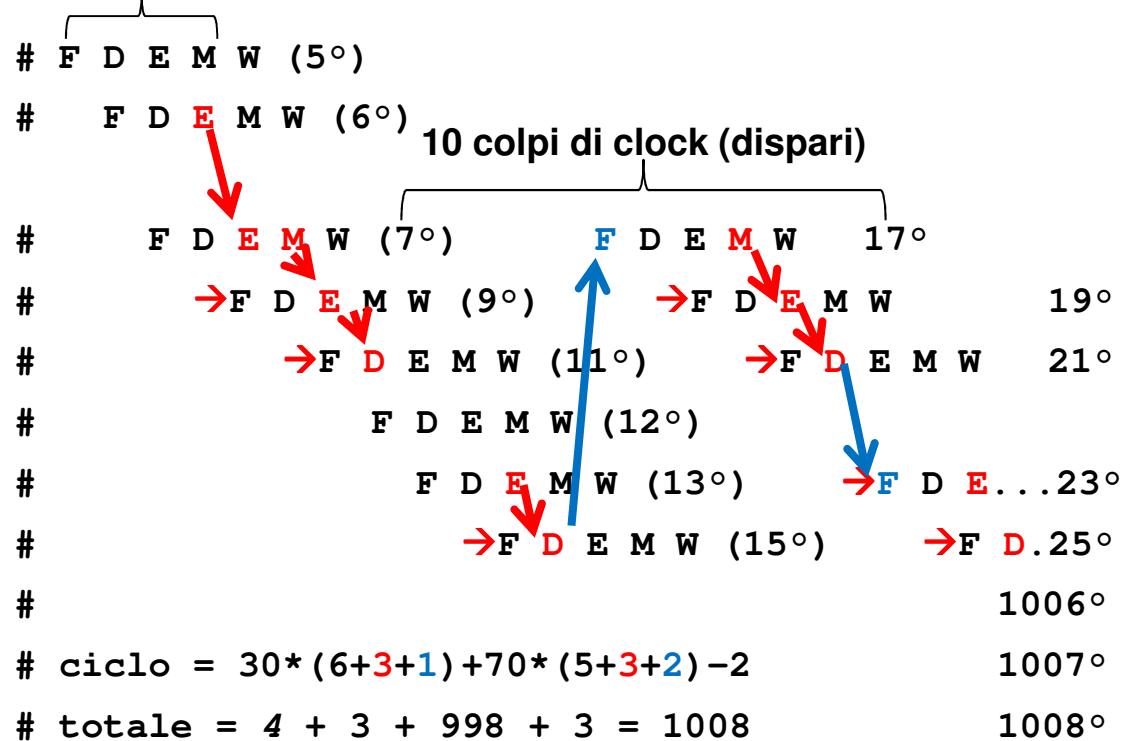
bge \$t1, \$zero, ciclo

li \$v0, 1

move \$a0, \$t0

syscall

**4 (riempimento)**



# Forwarding e stalli

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

li \$t0, 0

li \$t1, 396

**ciclo:**

lw \$t3, vettore(\$t1)

andi \$t2, \$t3, 1

beqz \$t2, salta

add \$t0, \$t0, \$t3

**salta:** addi \$t1, \$t1, -4

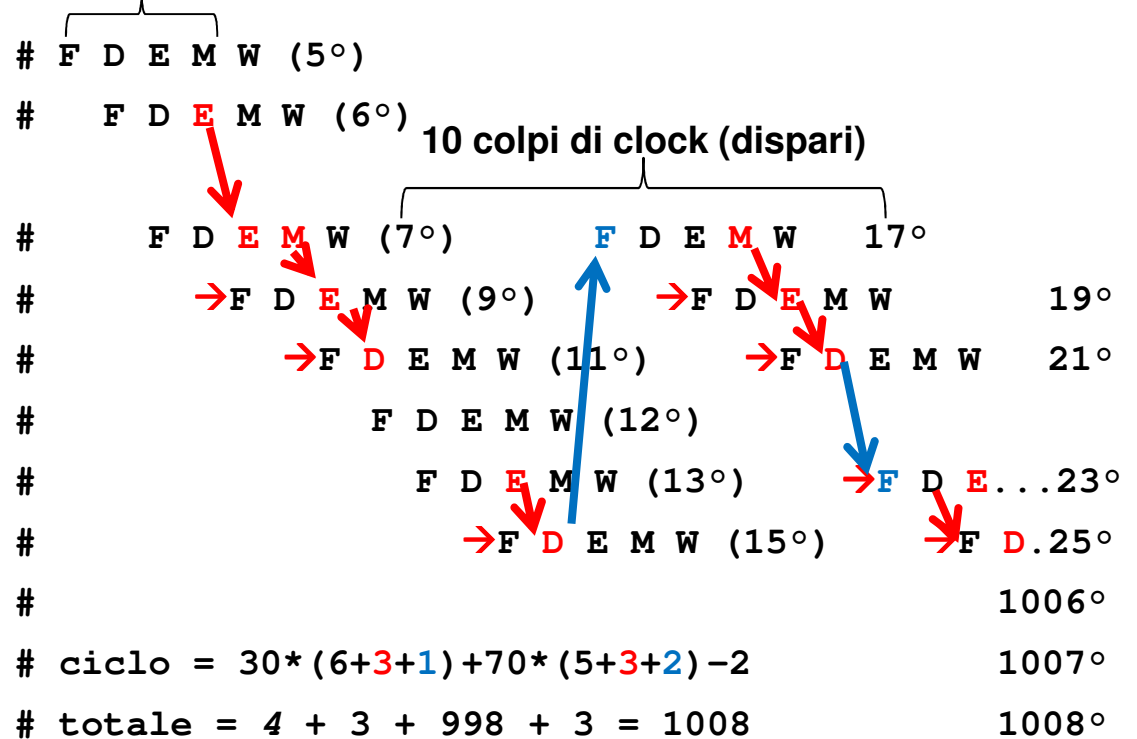
bge \$t1, \$zero, ciclo

li \$v0, 1

move \$a0, \$t0

syscall

**4 (riempimento)**



# Forwarding e stalli

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

li \$t0, 0

li \$t1, 396

**ciclo:**

lw \$t3, vettore(\$t1)

andi \$t2, \$t3, 1

beqz \$t2, salta

add \$t0, \$t0, \$t3

**salta:** addi \$t1, \$t1, -4

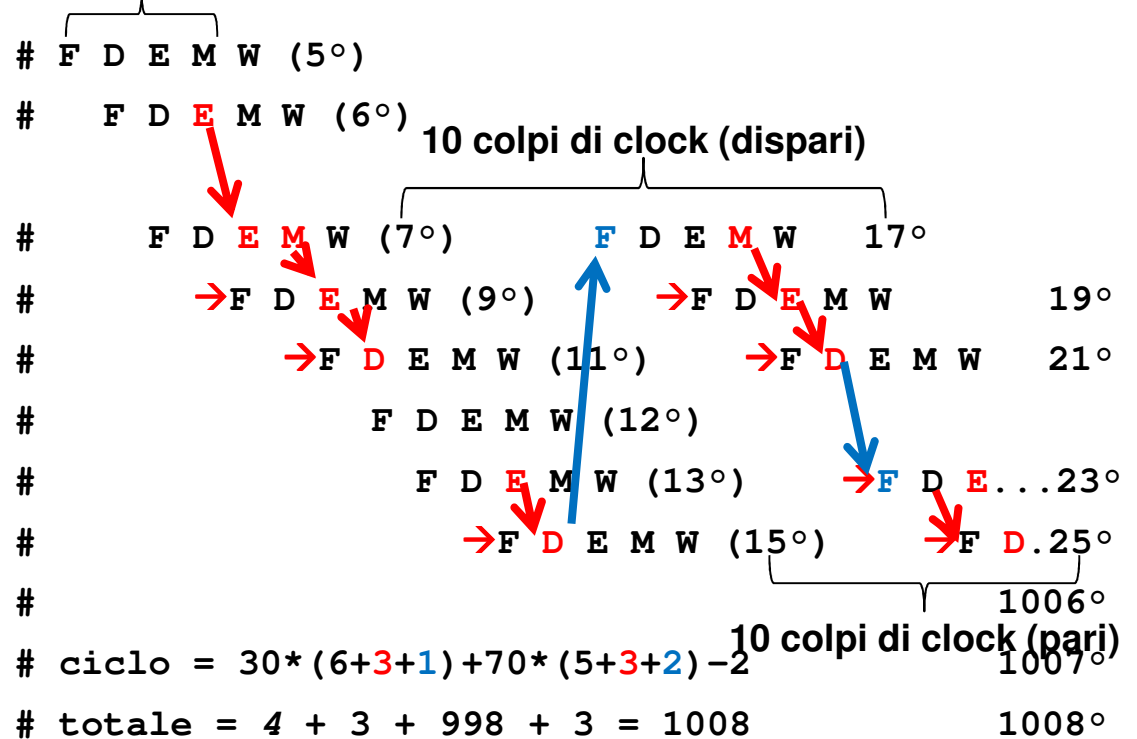
bge \$t1, \$zero, ciclo

li \$v0, 1

move \$a0, \$t0

syscall

**4 (riempimento)**



# Senza forwarding si attende il WB

```

.data
vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

.text
main:
    li    $t0, 0
    li    $t1, 396

ciclo:
    lw    $t3, vettore($t1)
    andi  $t2, $t3, 1
    beqz  $t2, salta
    add   $t0, $t0, $t3

salta:
    addi  $t1, $t1, -4
    bge   $t1, $zero, ciclo
    li    $v0, 1
    move  $a0, $t0
    syscall

```

**4 (riempimento)**

	# F D E M W (5°)	
	# F D E M W (6°)	
		<b>13 colpi di clock (dispari)</b>
	→ → F D E M W (9°)	22°
	→ → F D E M W (12°)	25°
	→ → F D E M W (15°)	28°
	F D E M W (16°)	
	F D E M W (17°)	30°
	→ → F D E M W (20°)	33°
		<b>13 colpi di clock (pari)</b>
		1308°
	# ciclo = 30*(6+6+1)+70*(5+6+2)-2	1309°
	# totale = 4+3+2+ 1298 + 3 = 1310	1310°



# Senza forwarding si attende il WB

```

.data
vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

.text
main:
    li    $t0, 0
    li    $t1, 396

ciclo:
    lw    $t3, vettore($t1)
    andi  $t2, $t3, 1
    beqz  $t2, salta
    add   $t0, $t0, $t3

salta:
    addi  $t1, $t1, -4
    bge   $t1, $zero, ciclo
    li    $v0, 1
    move  $a0, $t0
    syscall
    
```

**4 (riempimento)**

	# F D E M W (5°)	
	# F D E M <b>W</b> (6°)	
	↓	
	→ → F D E M W (9°)	13 colpi di clock (dispari)
	→ → F D E M W (12°)	22°
	→ → F D E M W (15°)	25°
	F D E M W (16°)	28°
	F D E M <b>W</b> (17°)	30°
	→ → F D E M W (20°)	33°
		<b>13 colpi di clock (pari)</b> 1308°
	# ciclo = 30*(6+6+1)+70*(5+6+2)-2	1309°
	# totale = 4+3+2+ 1298 + 3 = 1310	1310°

# Senza forwarding si attende il WB

```

.data
vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

.text
main:
    li    $t0, 0
    li    $t1, 396

ciclo:
    lw    $t3, vettore($t1)
    andi  $t2, $t3, 1
    beqz  $t2, salta
    add   $t0, $t0, $t3

salta:
    addi  $t1, $t1, -4
    bge   $t1, $zero, ciclo
    li    $v0, 1
    move  $a0, $t0
    syscall
    
```

**4 (riempimento)**

#	F D E M W	(5°)	
#	F D E M <b>W</b>	(6°)	
			<b>13 colpi di clock (dispari)</b>
#	→ → F D E M <b>W</b>	(9°)	22°
#	→ → F <b>D</b> E M W	(12°)	25°
#	→ → F D E M W	(15°)	28°
#	F D E M W	(16°)	
#	F D E M <b>W</b>	(17°)	30°
#	→ → F <b>D</b> E M W	(20°)	33°
			<b>13 colpi di clock (pari)</b>
			1308°
#	ciclo = 30*(6+6+1)+70*(5+6+2)-2		1309°
#	totale = 4+3+2+ 1298 + 3 = 1310		1310°

# Senza forwarding si attende il WB

```

.data
vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

.text
main:
    li    $t0, 0
    li    $t1, 396

ciclo:
    lw    $t3, vettore($t1)
    andi  $t2, $t3, 1
    beqz  $t2, salta
    add   $t0, $t0, $t3

salta:
    addi  $t1, $t1, -4
    bge   $t1, $zero, ciclo
    li    $v0, 1
    move  $a0, $t0
    syscall
    
```

**4 (riempimento)**

	# F D E M W (5°)	
	# F D E M <b>W</b> (6°)	
	↓	
	→ → F D E M <b>W</b> (9°)	22°
	→ → F <b>D</b> E M <b>W</b> (12°)	25°
	→ → F <b>D</b> E M W (15°)	28°
	F D E M W (16°)	
	F D E M <b>W</b> (17°)	30°
	→ → F <b>D</b> E M W (20°)	33°
		<b>13 colpi di clock (pari)</b> 1308°
	# ciclo = 30*(6+6+1)+70*(5+6+2)-2	1309°
	# totale = 4+3+2+ 1298 + 3 = 1310	1310°

**13 colpi di clock (dispari)**

# Senza forwarding si attende il WB

```

.data
vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

.text
main:
    li    $t0, 0
    li    $t1, 396

ciclo:
    lw    $t3, vettore($t1)
    andi  $t2, $t3, 1
    beqz  $t2, salta
    add   $t0, $t0, $t3

salta:
    addi  $t1, $t1, -4
    bge  $t1, $zero, ciclo
    li   $v0, 1
    move $a0, $t0
    syscall
    
```

**4 (riempimento)**

	# F D E M W (5°)	
	# F D E M <b>W</b> (6°)	
	↓	
	→ → F D E M <b>W</b> (9°)	22°
	→ → F <b>D</b> E M <b>W</b> (12°)	25°
	→ → F <b>D</b> E M <b>W</b> (15°)	28°
	F D E M <b>W</b> (16°)	
	F D E M <b>W</b> (17°)	30°
	→ → F <b>D</b> E M <b>W</b> (20°)	33°
		<b>13 colpi di clock (pari)</b> 1308°
	# ciclo = 30*(6+6+1)+70*(5+6+2)-2	1309°
	# totale = 4+3+2+ 1298 + 3 = 1310	1310°

**13 colpi di clock (dispari)**

# Senza forwarding si attende il WB

```

.data
vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

.text
main:
    li    $t0, 0
    li    $t1, 396

ciclo:
    lw    $t3, vettore($t1)
    andi  $t2, $t3, 1
    beqz  $t2, salta
    add   $t0, $t0, $t3

salta:
    addi  $t1, $t1, -4
    bge   $t1, $zero, ciclo
    li    $v0, 1
    move  $a0, $t0
    syscall
    
```

**4 (riempimento)**

# F D E M W (5°)

# F D E M **W** (6°)

# → → F D E M **W** (9°)

# → → F **D** E M **W** (12°)

# → → F **D** E M W (15°)

# F D E M W (16°)

# F D E M **W** (17°)

# → → F **D** E M W (20°)

# ciclo = 30\*(6+6+1)+70\*(5+6+2)-2

# totale = 4+3+2+ 1298 + 3 = 1310

**13 colpi di clock (dispari)**

22°

25°

28°

30°

33°

**13 colpi di clock (pari)**

1308°

1309°

1310°

# Senza forwarding si attende il WB

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

li \$t0, 0

li \$t1, 396

**ciclo:**

lw \$t3, vettore(\$t1)

andi \$t2, \$t3, 1

beqz \$t2, salta

add \$t0, \$t0, \$t3

**salta:**

addi \$t1, \$t1, -4

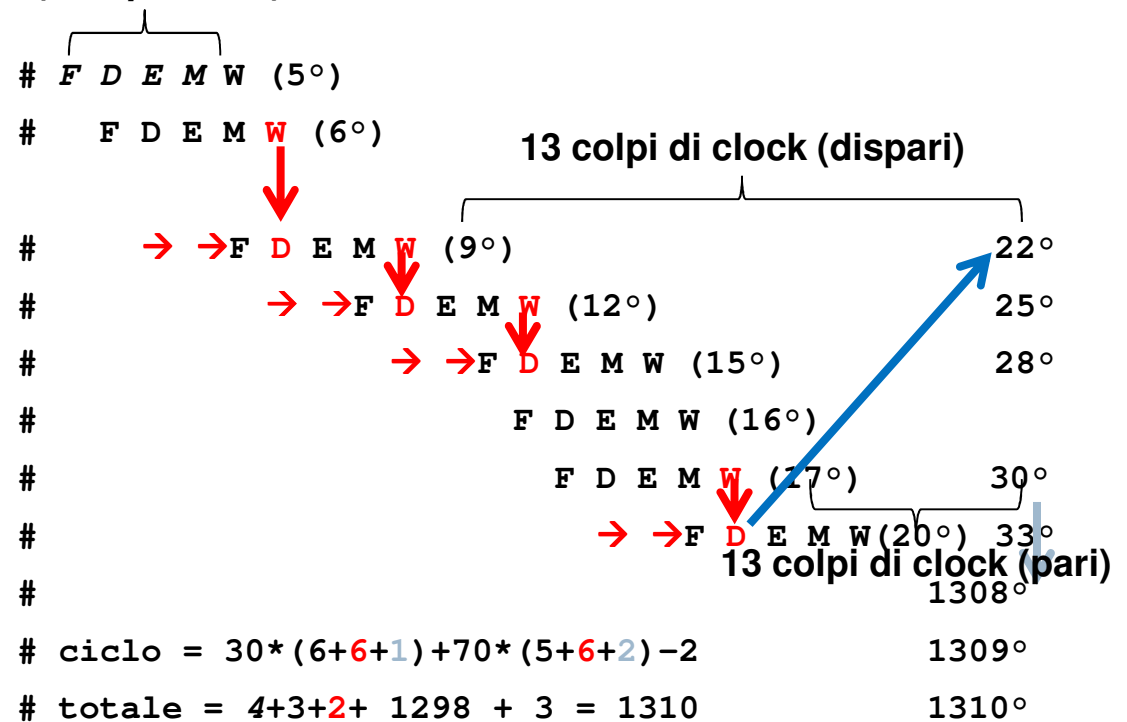
bge \$t1, \$zero, ciclo

li \$v0, 1

move \$a0, \$t0

syscall

4 (riempimento)



# Riordiniamo + forwarding

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

li \$t0, 0 # F D E M W (5°)

li \$t1, 396 # F D E M W (6°)

**ciclo:**

lw \$t3, vettore(\$t1) # F D E M W (7°) F D E M W 17°

andi \$t2, \$t3, 1 # →F D E M W (9°) →F D E M W 19°

beqz \$t2, salta # →F D E M W (11°) →F D E M W 21°

add \$t0, \$t0, \$t3 # F D E M W (12°)

salta: addi \$t1, \$t1, -4 # F D E M W (13°) →F D E...23°

bge \$t1, \$zero, ciclo # →F D E M W (15°) →F D...25°

li \$v0, 1 # 1006°

move \$a0, \$t0 # ciclo = 30\*(6+3+1)+70\*(5+3+2)-2 1007°

syscall # totale = 4 + 3 + 998 + 3 = 1008 1008°

4 (riempimento)

10 colpi di clock (dispari)

10 colpi di clock (pari)

# Riordiniamo + forwarding

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

li \$t0, 0 # F D E M W (5°)

li \$t1, 396 # F D E M W (6°)

**ciclo:**

lw \$t3, vettore(\$t1) # F D E M W (7°) F D E M W 17°

→ andi \$t2, \$t3, 1 # → F D E M W (9°) → F D E M W 19°

beqz \$t2, salta # → F D E M W (11°) → F D E M W 21°

add \$t0, \$t0, \$t3 # F D E M W (12°)

salta: addi \$t1, \$t1, -4 # F D E M W (13°) → F D E...23°

bge \$t1, \$zero, ciclo # → F D E M W (15°) → F D...25°

li \$v0, 1 # 1006°

move \$a0, \$t0 # ciclo = 30\*(6+3+1)+70\*(5+3+2)-2 1007°

syscall # totale = 4 + 3 + 998 + 3 = 1008 1008°

4 (riempimento)

10 colpi di clock (dispari)

10 colpi di clock (pari)



# Riordiniamo + forwarding

```

.data
vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

.text
main:
    li    $t0, 0           # F D E M W (5°)
    li    $t1, 396        # F D E M W (6°)

ciclo:
    lw    $t3, vettore($t1) # F D E M W (7°)
    andi  $t2, $t3, 1     # →F D E M W (9°)
    beqz  $t2, salta      # →F D E M W (11°)
    add   $t0, $t0, $t3   # F D E M W (12°)
salta:
    addi  $t1, $t1, -4    # F D E M W (13°)
    bge   $t1, $zero, ciclo # →F D E M W (15°)
    li    $v0, 1          #
    move  $a0, $t0        # ciclo = 30*(6+3+1)+70*(5+3+2)-2
    syscall               # totale = 4 + 3 + 998 + 3 = 1008

```

**4 (riempimento)**  
 F D E M W (5°)  
 F D E M W (6°)  
**10 colpi di clock (dispari)**  
 F D E M W (7°) F D E M W 17°  
 →F D E M W (9°) →F D E M W 19°  
 →F D E M W (11°) →F D E M W 21°  
 F D E M W (12°)  
 F D E M W (13°) →F D E...23°  
 →F D E M W (15°) →F D...25°  
 1006°  
**10 colpi di clock (pari)**  
 1007°  
 1008°

# Riordiniamo + forwarding

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

li \$t0, 0 # F D E M W (5°)

li \$t1, 396 # F D E M W (6°)

**ciclo:**

lw \$t3, vettore(\$t1) # F D E M W (7°) F D E M W 17°

→ andi \$t2, \$t3, 1 # → F D E M W (9°) → F D E M W 19°

→ beqz \$t2, salta # → F D E M W (11°) → F D E M W 21°

add \$t0, \$t0, \$t3 # F D E M W (12°)

salta: addi \$t1, \$t1, -4 # F D E M W (13°) → F D E...23°

→ bge \$t1, \$zero, ciclo # → F D E M W (15°) → F D...25°

li \$v0, 1 # 1006°

move \$a0, \$t0 # ciclo = 30\*(6+3+1)+70\*(5+3+2)-2 1007°

syscall # totale = 4 + 3 + 998 + 3 = 1008 1008°

4 (riempimento)

10 colpi di clock (dispari)

10 colpi di clock (pari)

# Riordiniamo + forwarding

```

.data
vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

.text
main:
    li    $t0, 0
    li    $t1, 396

ciclo:
    lw    $t3, vettore($t1)
    andi  $t2, $t3, 1
    beqz  $t2, salta
    add   $t0, $t0, $t3
salta:  addi $t1, $t1, -4
    bge  $t1, $zero, ciclo
    li   $v0, 1
    move $a0, $t0
    syscall

```

**4 (riempimento)**

# F D E M W (5°)

# F D E M W (6°)

**10 colpi di clock (dispari)**

# F D E M W (7°) F D E M W 17°

# →F D E M W (9°) →F D E M W 19°

# →F D E M W (11°) →F D E M W 21°

# F D E M W (12°)

# F D E M W (13°) →F D E...23°

# →F D E M W (15°) →F D...25°

**10 colpi di clock (pari)**

# ciclo = 30\*(6+3+1)+70\*(5+3+2)-2 1006°

# totale = 4 + 3 + 998 + 3 = 1008 1007°

1008°

# Riordiniamo + forwarding

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

li \$t0, 0 # F D E M W (5°)

li \$t1, 396 # F D E M W (6°)

**ciclo:**

lw \$t3, vettore(\$t1) # F D E M W (7°) F D E M W 17°

→ andi \$t2, **addi \$t1, \$t1, -4** → F D E M W (9°) → F D E M W 19°

→ beqz \$t2, # F D E M W (11°) → F D E M W 21°

add \$t0, \$t0, \$t3 # F D E M W (12°)

**salta:**

→ bge \$t1, \$zero, ciclo # F D E M W (13°) → F D E...23°

li \$v0, 1 # → F D E M W (15°) → F D...25°

move \$a0, \$t0 # ciclo = 30\*(6+3+1)+70\*(5+3+2)-2 1006°

syscall # totale = 4 + 3 + 998 + 3 = 1008 1007° 1008°

4 (riempimento)

10 colpi di clock (dispari)

10 colpi di clock (pari)

# Riordiniamo + forwarding

```

.data
vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

.text
main:
    li    $t0, 0           # F D E M W (5°)
    li    $t1, 396         # F D E M W (6°)

ciclo:
    lw    $t3, vettore($t1) # F D E M W (7°)
    andi  $t2, $t3, $t3     # F D E M W (9°)
    beqz  $t2, $t1, $t1, -4 # F D E M W (11°)
    add   $t0, $t0, $t3     # F D E M W (12°)

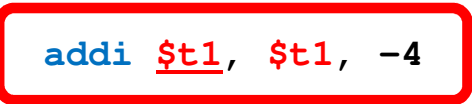
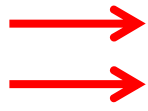
salta:
    bge   $t1, $zero, ciclo # F D E M W (13°)
    li    $v0, 1            # F D E M W (15°)
    move  $a0, $t0         # ciclo = 30*(6+3+1)+70*(5+3+2)-2
    syscall                # totale = 4 + 3 + 998 + 3 = 1008

```

4 (riempimento)

10 colpi di clock (dispari)

10 colpi di clock (pari)



# Riordiniamo + forwarding

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

li \$t0, 0 # F D E M W (5°)

li \$t1, 396 # F D E M W (6°)

4 (riempimento)

**ciclo:**

lw \$t3, vettore(\$t1) # F D E M W (7°) F D E M W 17°



andi \$t2, \$t3, \$t3 → F D E M W (9°) → F D E M W 19°

beqz \$t2, \$t3, \$t3 → F D E M W (11°) → F D E M W 21°

add \$t0, \$t0, \$t3 # F D E M W (12°)

**salta:**

bge \$t1, \$zero, ciclo # F D E M W (13°) → F D E...23°

bge \$t1, \$zero, ciclo # → F D E M W (15°) → F D...25°

li \$v0, 1 # 1006°

move \$a0, \$t0 # ciclo = 30\*(6+3+1)+70\*(5+3+2)-2 1007°

syscall # totale = 4 + 3 + 998 + 3 = 1008 1008°

10 colpi di clock (dispari)

10 colpi di clock (pari)

# Eliminati 2 stalli \* 100 = 200 ck

```

.data
vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

.text
main:

    li    $t0, 0        # F D E M W (5°)
    li    $t1, 396     #  F D E M W (6°)

ciclo:

    lw    $t3, vettore($t1)    #    F D E M W (7°)          F D E M W          15°
    andi  $t2, $t3, 1 #    →F D E M W (9°)          →F D E M W          17°
    addi  $t1, $t1, -4 #          F D E M W (10°)          F D E M W          18°
    beqz  $t2, salta #          F D E M W (11°)          F D E M W          19°
    add   $t0, $t0, $t3 #          F D E M W (12°)

salta: bge  $t1, $zero, ciclo #          F D E M W (13°)          F D E...
21°

    li    $v0, 1        #
    move  $a0, $t0     # ciclo = 30*(6+1+1)+70*(5+1+2)-2          807°
    syscall # totale = 4 + 3 + 798 + 3 = 808          808°

```

# Eliminati 2 stalli \* 100 = 200 ck

```

.data
vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

.text
main:

    li    $t0, 0        # F D E M W (5°)
    li    $t1, 396     #   F D E M W (6°)

ciclo:
    lw    $t3, vettore($t1) #   F D E M W (7°)           F D E M W           15°
    andi  $t2, $t3, 1 #   →F D E M W (9°)           →F D E M W           17°
    addi  $t1, $t1, -4 #           F D E M W (10°)           F D E M W           18°
    beqz  $t2, salta #           F D E M W (11°)           F D E M W           19°
    add   $t0, $t0, $t3 #           F D E M W (12°)

salta: bge  $t1, $zero, ciclo #           F D E M W (13°)           F D E...
21°

    li    $v0, 1        #
    move  $a0, $t0      # ciclo = 30*(6+1+1)+70*(5+1+2)-2           807°
    syscall # totale = 4 + 3 + 798 + 3 = 808           808°

```



# Eliminati 2 stalli \* 100 = 200 ck

**.data**

vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

**.text**

**main:**

```
li    $t0, 0      # F D E M W (5°)
li    $t1, 396    #  F D E M W (6°)
```

**ciclo:**

```
→ lw    $t3, vettore($t1) #  F D E M W (7°)           F D E M W           15°
andi  $t2, $t3, 1 #      → F D E M W (9°)           → F D E M W           17°
addi  $t1, $t1, -4 #      F D E M W (10°)           F D E M W           18°
beqz  $t2, salta #      F D E M W (11°)           F D E M W           19°
add   $t0, $t0, $t3 #      F D E M W (12°)
salta: bge $t1, $zero, ciclo #      F D E M W (13°)           F D E...
21°

li    $v0, 1      #
move  $a0, $t0    # ciclo = 30*(6+1+1)+70*(5+1+2)-2
syscall # totale = 4 + 3 + 798 + 3 = 808
```

8 colpi di clock (dispari)

8 colpi di clock (pari)

807°

808°

# Cosa c'è in pipeline a t=16?

```

.data
vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

.text
main:
    4 (riempimento)
    li    $t0, 0                # F D E M W (5°)
    li    $t1, 396            # F D E M W (6°)

ciclo:
    lw    $t3, vettore($t1) # F D E M W (7°)          F D E M W          17°
    andi  $t2, $t3, 1        # →F D E M W (9°)          →F D E M W          19°
    beqz  $t2, salta          # →F D E M W (11°)         →F D E M W          21°
    add   $t0, $t0, $t3        # F D E M W (12°)
salta:  addi $t1, $t1, -4      # F D E M W (13°)          →F D E...23°
    bge  $t1, $zero, ciclo    # →F D E M W (15°)          →F D...25°
    li   $v0, 1                # 1006°
    move $a0, $t0              # ciclo = 30*(6+3+1)+70*(5+3+2)-2    1007°
    syscall                    # totale = 4 + 3 + 998 + 3 = 1008
1008°

```

# Cosa c'è in pipeline a t=16?

```

.data
vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

.text
main:
    4 (riempimento)
    li    $t0, 0                # F D E M W (5°)
    li    $t1, 396              # F D E M W (6°)

ciclo:
    lw    $t3, vettore($t1)     # F D E M W (7°)          F D E M W
    andi  $t2, $t3, 1           # →F D E M W (9°)      →F D E M W
    beqz  $t2, salta            # →F D E M W (11°)    →F D E M W
    add   $t0, $t0, $t3         # F D E M W (12°)
salta:  addi $t1, $t1, -4       # F D E M W (13°)      →F D E...23°
    bge  $t1, $zero, ciclo     # →F D E M W (15°)    →F D...25°
    li   $v0, 1                #
    move $a0, $t0              # ciclo = 30*(6+3+1)+70*(5+3+2)-2
    syscall                     # totale = 4 + 3 + 998 + 3 = 1008

1008°

```

# Cosa c'è in pipeline a t=16?

```

.data
vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

.text
main:
    4 (riempimento)
    li    $t0, 0                # F D E M W (5°)
    li    $t1, 396              # F D E M W (6°)

ciclo:
    lw    $t3, vettore($t1)    # F D E M W (7°)          F D E M W          17°
    andi  $t2, $t3, 1          # →F D E M W (9°)      →F D E M W          19°
    beqz  $t2, salta           # →F D E M W (11°)     →F D E M W          21°
    add   $t0, $t0, $t3        # F D E M W (12°)
salta:  addi $t1, $t1, -4      # F D E M W (13°)     →F D E...23°
    bge   $t1, $zero, ciclo    # →F D E M W (15°)     →F D...25°
    li    $v0, 1                # 1006°
    move  $a0, $t0              # ciclo = 30*(6+3+1)+70*(5+3+2)-2 1007°
    syscall                     # totale = 4 + 3 + 998 + 3 = 1008
1008°

```

# Cosa c'è in pipeline a t=16?

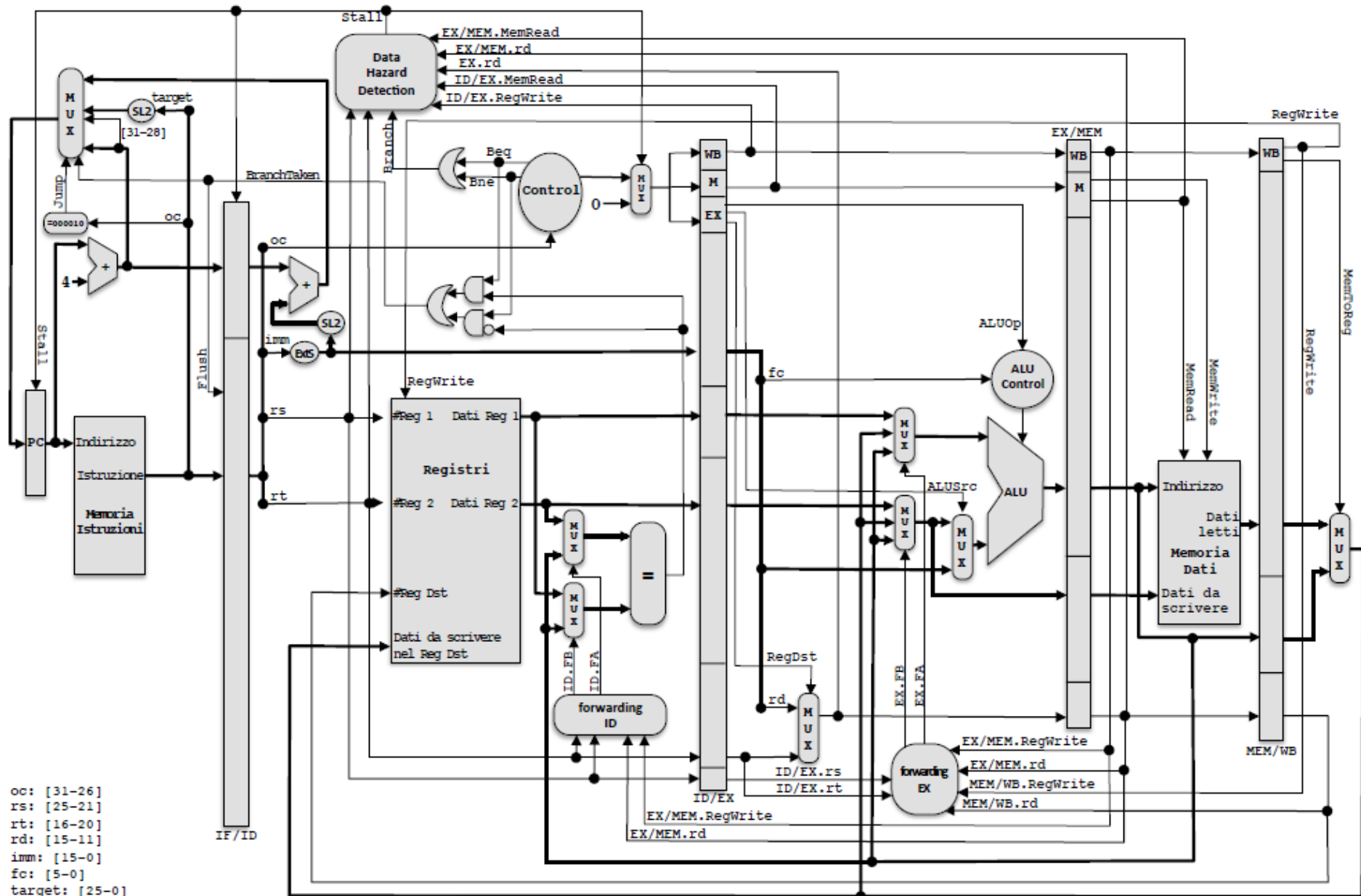
```

.data
vettore: # qui 70 pari e 30 dispari (p.es il 1° è dispari e il 2° pari)

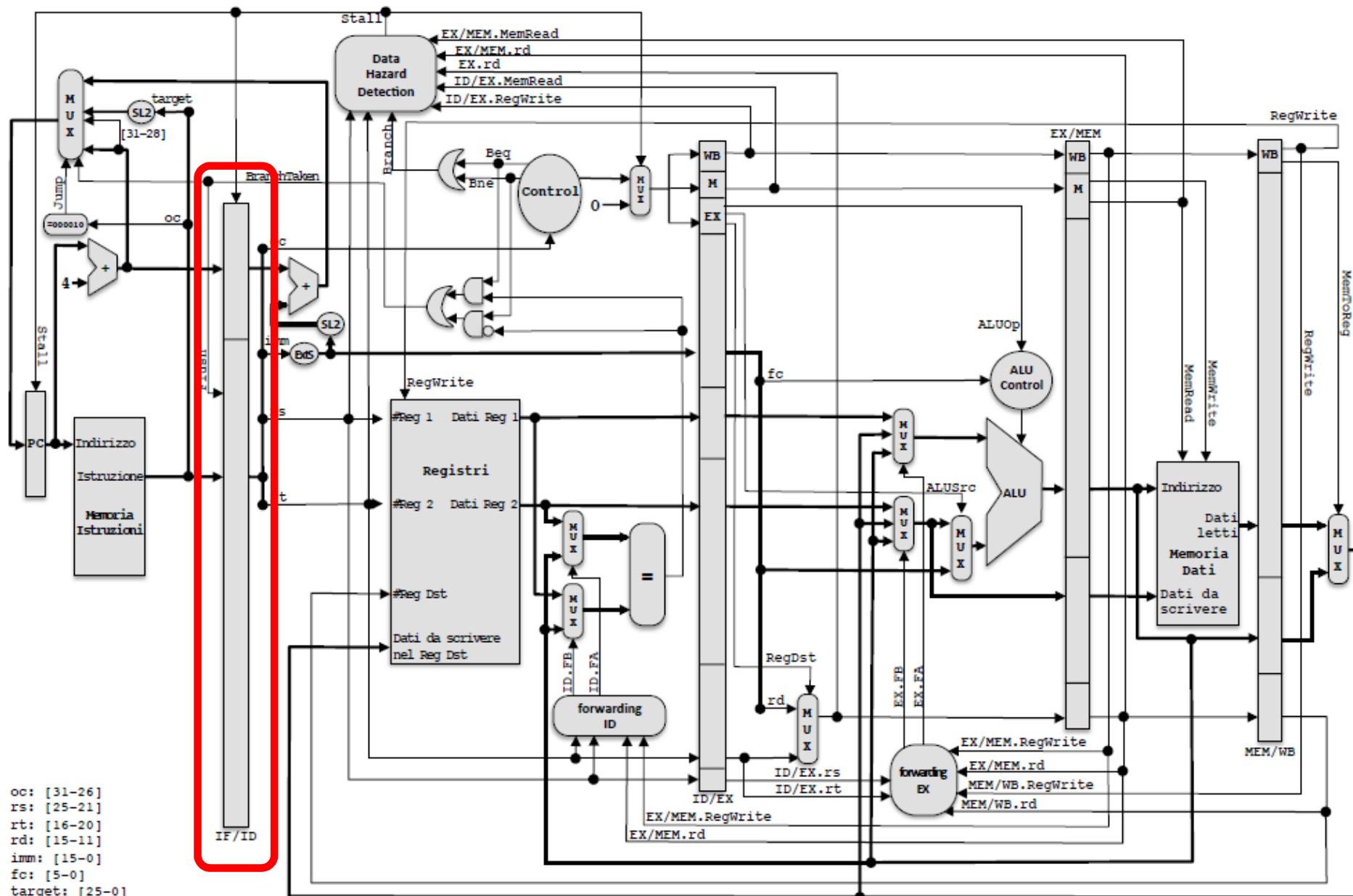
.text
main:
    4 (riempimento)
    li    $t0, 0           # F D E M W (5°)
    li    $t1, 396        # F D E M W (6°)

ciclo:
    lw    $t3, vettore($t1) # F D E M W (7°)           F D E M W 17°
    andi  $t2, $t3, 1     # →F D E M W (9°)           →F D E M W 19°
    beqz  $t2, salta      # →F D E M W (11°)          →F D E M W 21°
    add   $t0, $t0, $t3   # F D E M W (12°)
salta:
    addi  $t1, $t1, -4    # F D E M W (13°)          →F D E...23°
    bge   $t1, $zero, ciclo # →F D E M W (15°)        →F D...25°
    li    $v0, 1          # 1006°
    move  $a0, $t0        # ciclo = 30*(6+3+1)+70*(5+3+2)-2 1007°
    syscall               # totale = 4 + 3 + 998 + 3 = 1008
1008°

```

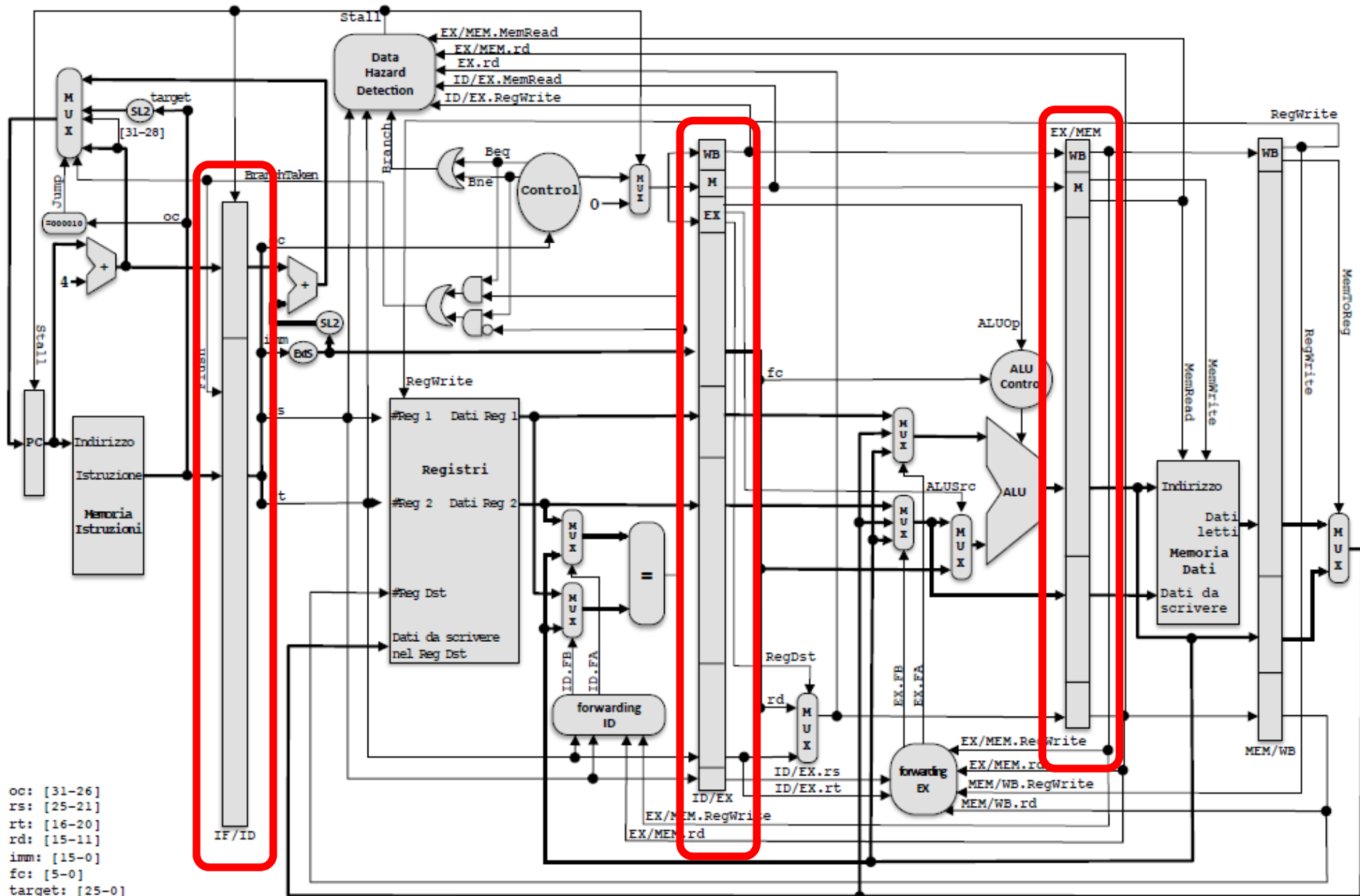


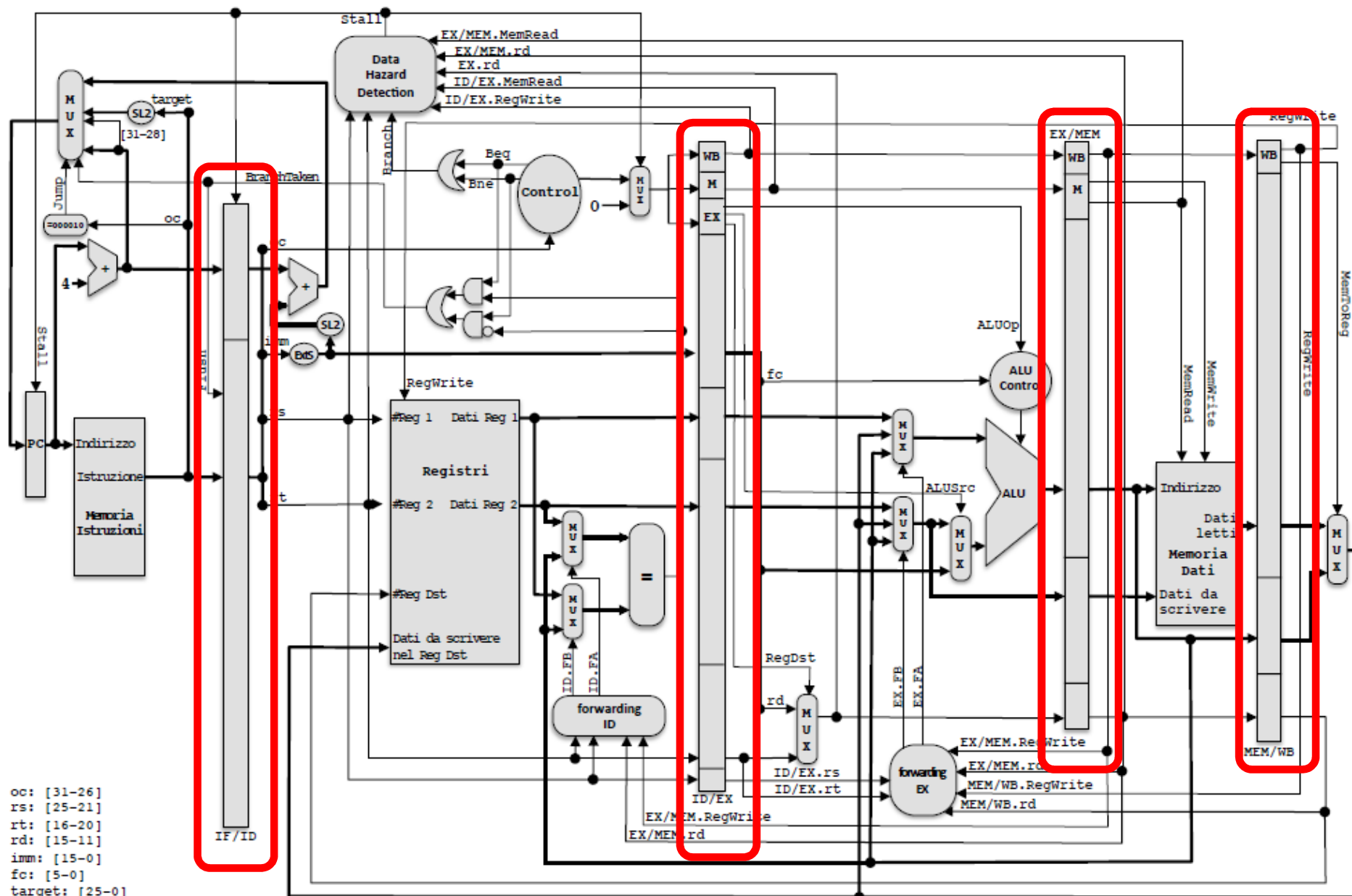
oc: [31-26]  
 rs: [25-21]  
 rt: [16-20]  
 rd: [15-11]  
 imm: [15-0]  
 fc: [5-0]  
 target: [25-0]











oc: [31-26]  
 rs: [25-21]  
 rt: [16-20]  
 rd: [15-11]  
 imm: [15-0]  
 fc: [5-0]  
 target: [25-0]

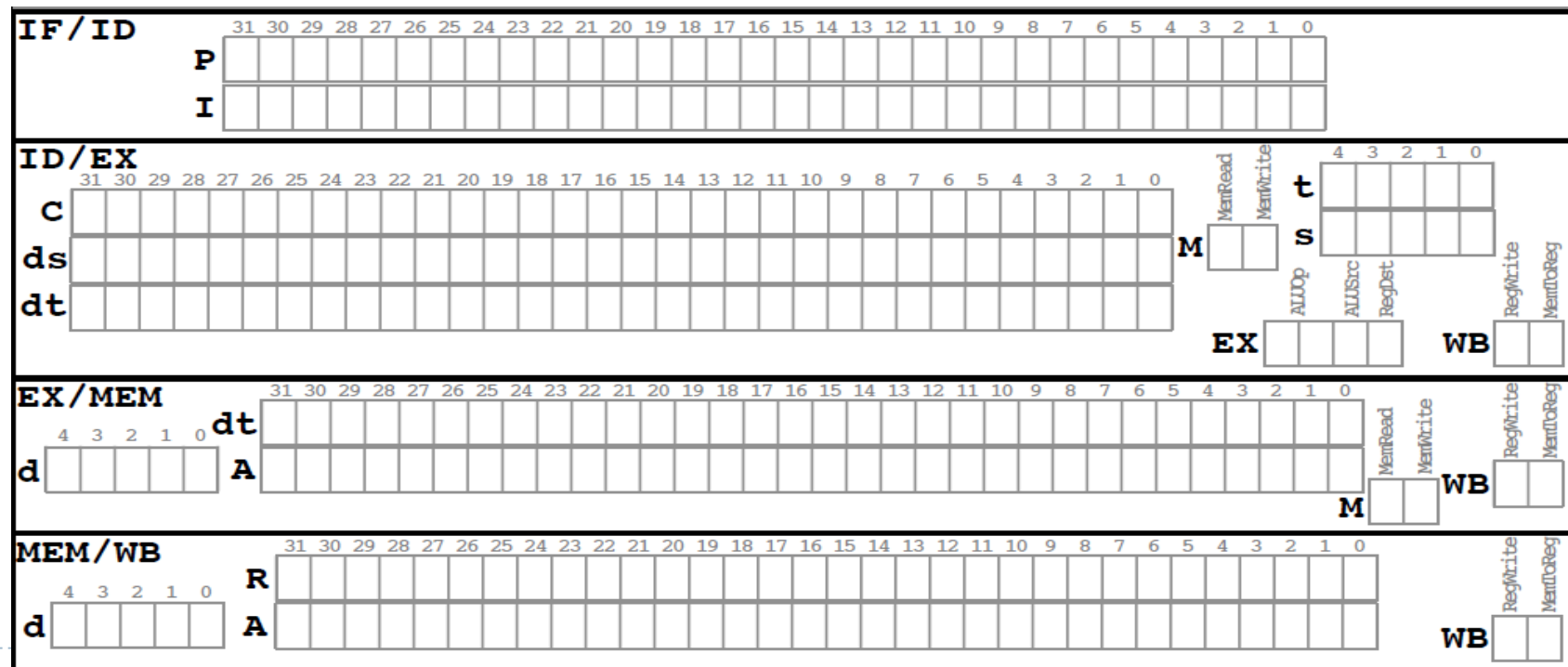
# Cosa c'è in pipeline a t=16?

---

- ▶ A t=16° si completa lo stallo dopo la **bge**, quindi in pipeline ci sono le istruzioni seguenti:
- ▶ **WB:** stallo per il salto dopo `bge $t1, $zero, ciclo` (control hazard)
- ▶ **MEM:** `lw $t3, vettore($t1)`
- ▶ **EXE:** stallo di `andi $t2, $t3, 1` che attende lw (data hazard)
- ▶ **ID:** `andi $t2, $t3, 1` che riceve da lw tramite forwarding
- ▶ **IF:** `beqz $t2, salta` che andrà in stallo in fase ID per attendere andi (data haz.)

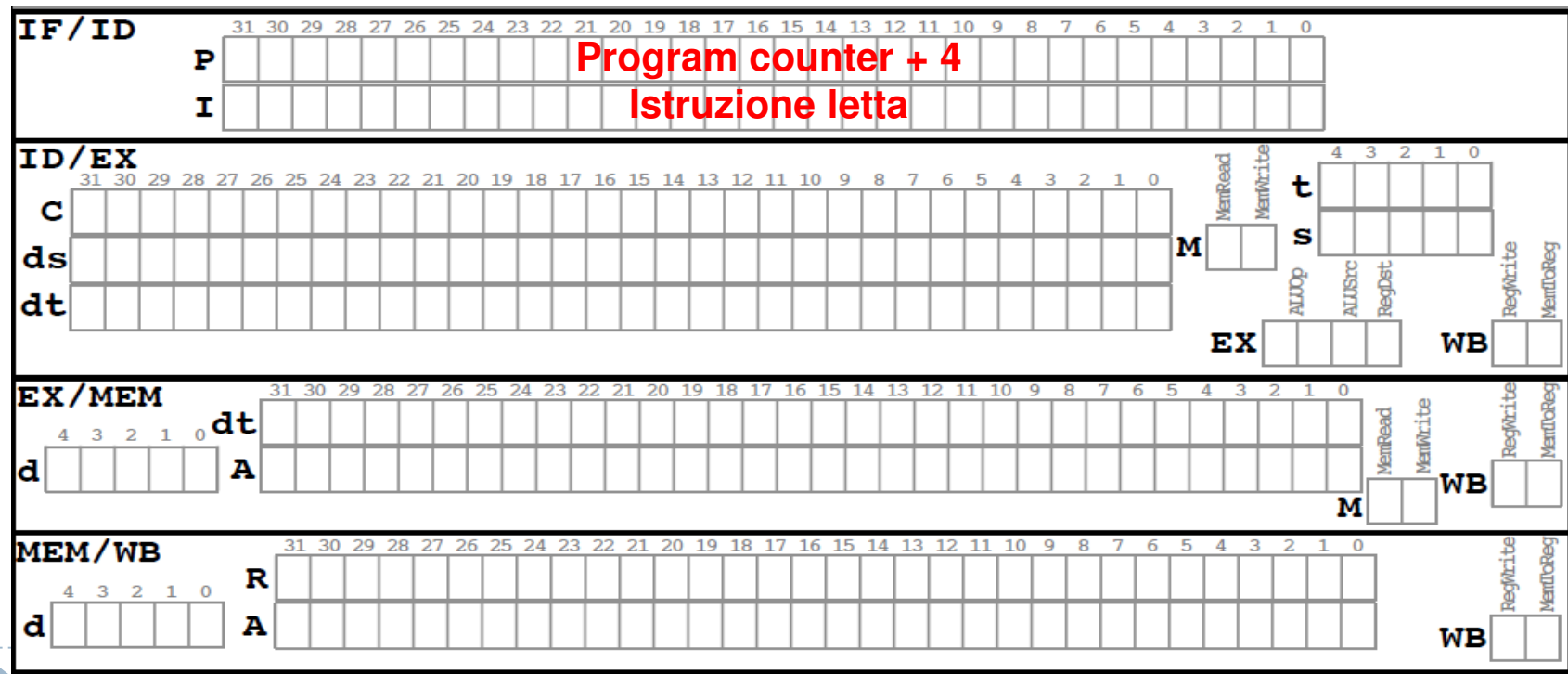
# Cosa c'è in pipeline a t=16?

- ▶ A t=16° si completa lo stallo dopo la **bge**, quindi in pipeline ci sono le istruzioni seguenti:
- ▶ **WB:** stallo per il salto dopo **bge \$t1, \$zero, ciclo** (control hazard)
- ▶ **MEM:** **lw \$t3, vettore(\$t1)**
- ▶ **EXE:** stallo di **andi \$t2, \$t3, 1** che attende lw (data hazard)
- ▶ **ID:** **andi \$t2, \$t3, 1** che riceve da lw tramite forwarding
- ▶ **IF:** **beqz \$t2, salta** che andrà in stallo in fase ID per attendere andi (data haz.)



# Cosa c'è in pipeline a t=16?

- ▶ A t=16° si completa lo stallo dopo la **bge**, quindi in pipeline ci sono le istruzioni seguenti:
- ▶ **WB:** stallo per il salto dopo **bge \$t1, \$zero, ciclo** (control hazard)
- ▶ **MEM:** **lw \$t3, vettore(\$t1)**
- ▶ **EXE:** stallo di **andi \$t2, \$t3, 1** che attende lw (data hazard)
- ▶ **ID:** **andi \$t2, \$t3, 1** che riceve da lw tramite forwarding
- ▶ **IF:** **beqz \$t2, salta** che andrà in stallo in fase ID per attendere andi (data haz.)



# Cosa c'è in pipeline a t=16?

- ▶ A t=16° si completa lo stallo dopo la **bge**, quindi in pipeline ci sono le istruzioni seguenti:
- ▶ **WB:** stallo per il salto dopo **bge \$t1, \$zero, ciclo** (control hazard)
- ▶ **MEM:** **lw \$t3, vettore(\$t1)**
- ▶ **EXE:** stallo di **andi \$t2, \$t3, 1** che attende lw (data hazard)
- ▶ **ID:** **andi \$t2, \$t3, 1** che riceve da lw tramite forwarding
- ▶ **IF:** **beqz \$t2, salta** che andrà in stallo in fase ID per attendere andi (data haz.)



# Cosa c'è in pipeline a t=16?

- ▶ A t=16° si completa lo stallo dopo la **bge**, quindi in pipeline ci sono le istruzioni seguenti:
- ▶ **WB:** stallo per il salto dopo **bge \$t1, \$zero, ciclo** (control hazard)
- ▶ **MEM:** **lw \$t3, vettore(\$t1)**
- ▶ **EXE:** stallo di **andi \$t2, \$t3, 1** che attende lw (data hazard)
- ▶ **ID:** **andi \$t2, \$t3, 1** che riceve da lw tramite forwarding
- ▶ **IF:** **beqz \$t2, salta** che andrà in stallo in fase ID per attendere andi (data haz.)



# Cosa c'è in pipeline a t=16?

- ▶ A t=16° si completa lo stallo dopo la **bge**, quindi in pipeline ci sono le istruzioni seguenti:
- ▶ **WB:** stallo per il salto dopo **bge \$t1, \$zero, ciclo** (control hazard)
- ▶ **MEM:** **lw \$t3, vettore(\$t1)**
- ▶ **EXE:** stallo di **andi \$t2, \$t3, 1** che attende lw (data hazard)
- ▶ **ID:** **andi \$t2, \$t3, 1** che riceve da lw tramite forwarding
- ▶ **IF:** **beqz \$t2, salta** che andrà in stallo in fase ID per attendere andi (data haz.)





# Cosa c'è in pipeline a t=16?

- ▶ A t=16° si completa lo stallo dopo la **bge**, quindi in pipeline ci sono le istruzioni seguenti:
- ▶ **WB:** stallo per il salto dopo **bge \$t1, \$zero, ciclo** (control hazard)
- ▶ **MEM:** **lw \$t3, vettore(\$t1)**
- ▶ **EXE:** stallo di **andi \$t2, \$t3, 1** che attende lw (data hazard)
- ▶ **ID:** **andi \$t2, \$t3, 1** che riceve da lw tramite forwarding
- ▶ **IF:** **beqz \$t2, salta** che andrà in stallo in fase ID per attendere andi (data haz.)



# Cosa c'è in pipeline a t=16?

- ▶ A t=16° si completa lo stallo dopo la **bge**, quindi in pipeline ci sono le istruzioni seguenti:
- ▶ **WB:** stallo per il salto dopo **bge \$t1, \$zero, ciclo** (control hazard)
- ▶ **MEM:** **lw \$t3, vettore(\$t1)**
- ▶ **EXE:** stallo di **andi \$t2, \$t3, 1** che attende lw (data hazard)
- ▶ **ID:** **andi \$t2, \$t3, 1** che riceve da lw tramite forwarding
- ▶ **IF:** **beqz \$t2, salta** che andrà in stallo in fase ID per attendere andi (data haz.)



# Cosa c'è in pipeline a t=16?

- ▶ A t=16° si completa lo stallo dopo la **bge**, quindi in pipeline ci sono le istruzioni seguenti:
- ▶ **WB:** stallo per il salto dopo **bge \$t1, \$zero, ciclo** (control hazard)
- ▶ **MEM:** **lw \$t3, vettore(\$t1)**
- ▶ **EXE:** stallo di **andi \$t2, \$t3, 1** che attende lw (data hazard)
- ▶ **ID:** **andi \$t2, \$t3, 1** che riceve da lw tramite forwarding
- ▶ **IF:** **beqz \$t2, salta** che andrà in stallo in fase ID per attendere andi (data haz.)



# Cosa c'è in pipeline a t=16?

- ▶ A t=16° si completa lo stallo dopo la **bge**, quindi in pipeline ci sono le istruzioni seguenti:
- ▶ **WB:** stallo per il salto dopo **bge \$t1, \$zero, ciclo** (control hazard)
- ▶ **MEM:** **lw \$t3, vettore(\$t1)**
- ▶ **EXE:** stallo di **andi \$t2, \$t3, 1** che attende lw (data hazard)
- ▶ **ID:** **andi \$t2, \$t3, 1** che riceve da lw tramite forwarding
- ▶ **IF:** **beqz \$t2, salta** che andrà in stallo in fase ID per attendere andi (data haz.)



# Cosa c'è in pipeline a t=16?

- ▶ A t=16° si completa lo stallo dopo la **bge**, quindi in pipeline ci sono le istruzioni seguenti:
- ▶ **WB:** stallo per il salto dopo **bge \$t1, \$zero, ciclo** (control hazard)
- ▶ **MEM:** **lw \$t3, vettore(\$t1)**
- ▶ **EXE:** stallo di **andi \$t2, \$t3, 1** che attende lw (data hazard)
- ▶ **ID:** **andi \$t2, \$t3, 1** che riceve da lw tramite forwarding
- ▶ **IF:** **beqz \$t2, salta** che andrà in stallo in fase ID per attendere andi (data haz.)

