

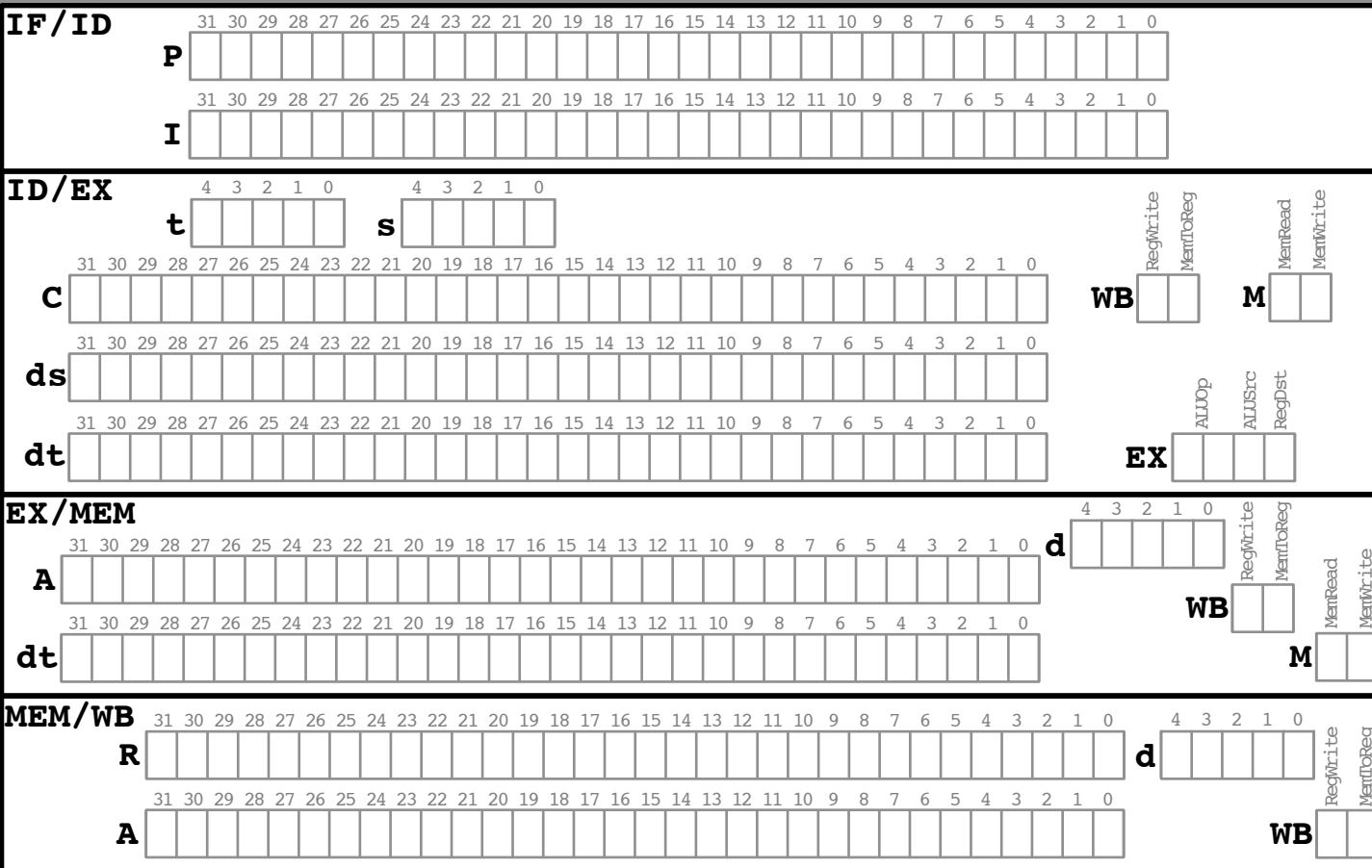
ESERCIZIO 1 Determinare il numero di cicli di clock richiesti per eseguire il seguente programma secondo lo schema di implementazione in pipeline:

```
      addi $s0, $zero, 20
loopA: beq $s0, $zero, end
      addi $t1, $t0, 16
loopB: lw $s1, arr($t1)      #arr è l'indirizzo di un array di words
      add $s2, $s2, $s1
      addi $t1, $t1, -4
      bne $t1, $t0, loopB
      addi $t0, $t0, 1024
      add $s0, $s0, -1
      j loopA
end:   sub $s2, $s2, $s3
```

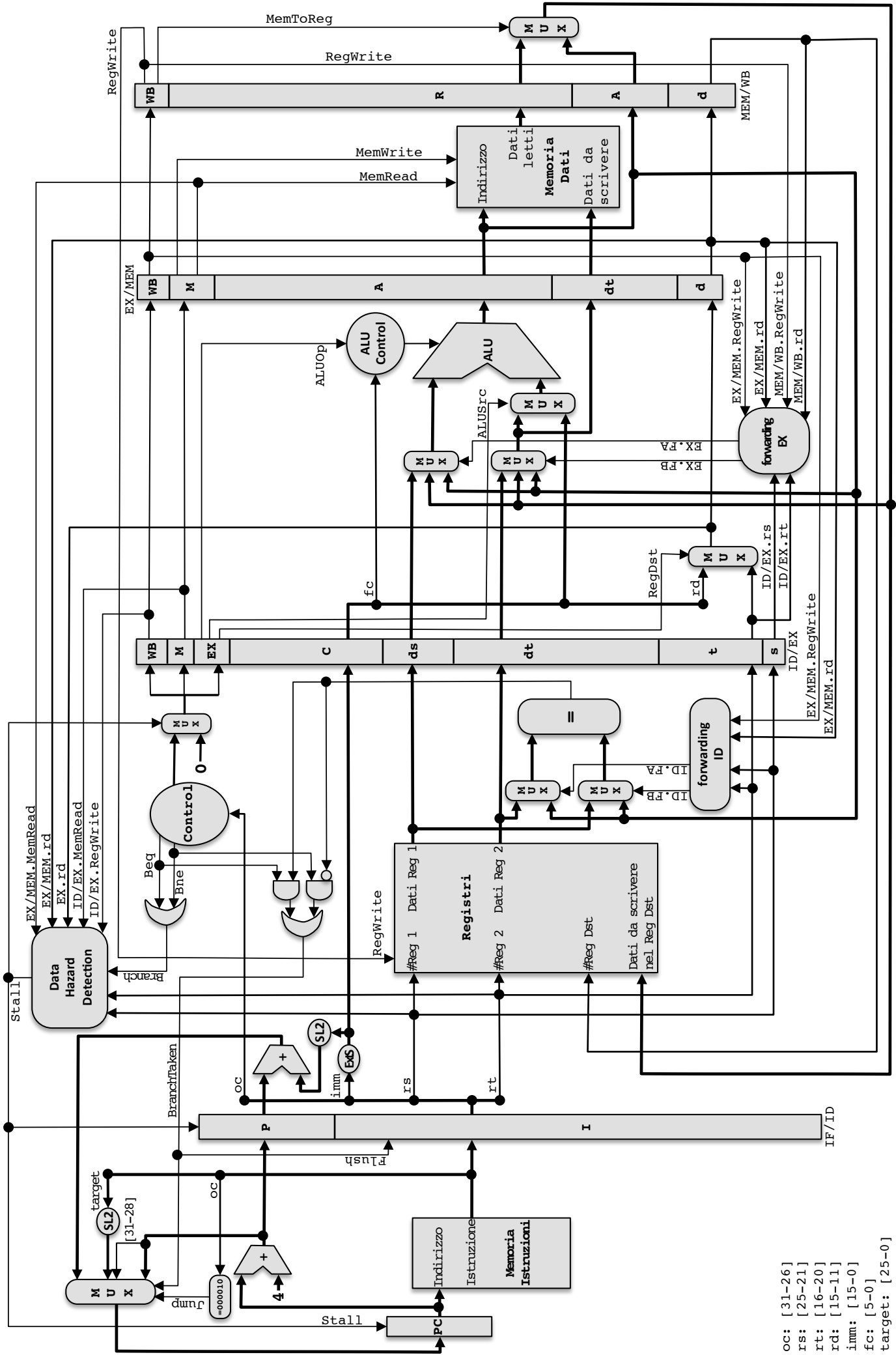
ESERCIZIO 2 Si consideri il seguente programma (accanto ad ogni istruzione è riportata la rappresentazione in binario):

0x00400000	lw \$s0, (\$s1)	100011 10001 10000 0000000000000000
	add \$s1, \$s1, \$s0	000000 10001 10000 10001 00000 100000
	addi \$s2, \$s1, 4	001000 10001 10010 0000000000000100
	lw \$s0, (\$s1)	100011 10001 10000 0000000000000000

Assumendo un'esecuzione in pipeline secondo lo schema riportato nella pagina seguente (è quello visto a lezione) e che quando inizia l'esecuzione, al ciclo di clock 1, i contenuti dei registri sono $s0 = 0x00000000$, $s1 = 0x10000000$, $s2 = 0x00000000$ e la word in memoria all'indirizzo $0x10000000$ è $0x00000010$, determinare il contenuto dei registri di pipeline alla fine del ciclo 5 riempiendo lo schema qui sotto riportato.



Implementazione pipeline di MIPS (solamente le istruzioni: add, addi, sub, and, andi, or, ori, xor, xori, nor, slt, slti, lw, sw, beq, bne, j).



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