Κεφάλαιο 4

Ο επεξεργαστής

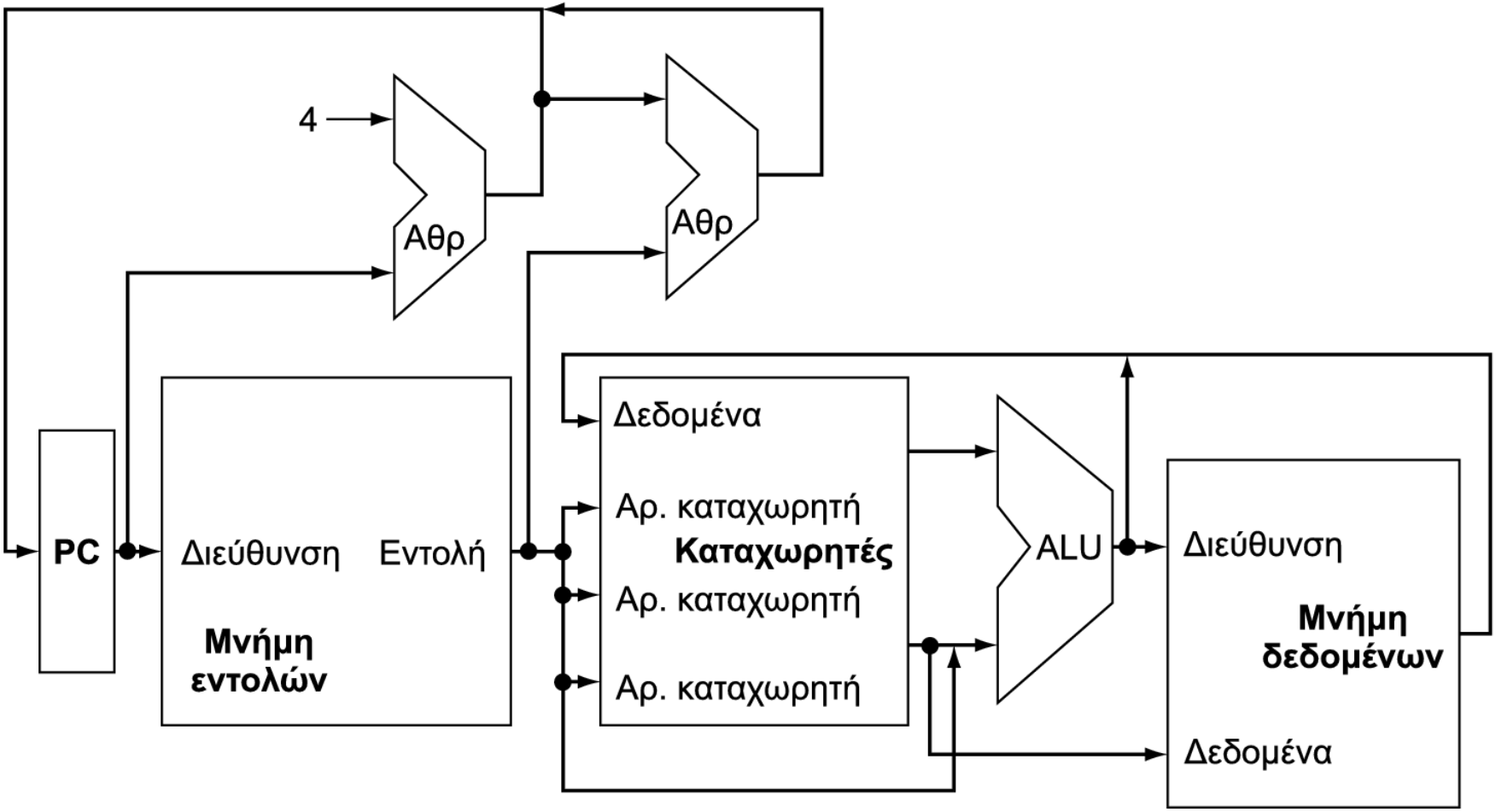
Εισαγωγή

- Παράγοντες απόδοσης της CPU
  - Πλήθος εντολών
    - Καθορίζεται από την αρχιτεκτονική συνόλου εντολών και το μεταγλωττιστή
  - CPI και Χρόνος κύκλου
    - Καθορίζεται από το υλικό της CPU
- Θα εξετάσουμε δύο υλοποιήσεις του MIPS
  - Μια απλουστευμένη έκδοση
  - Μια πιο ρεαλιστική έκδοση με διοχέτευση (pipeline)
- Απλό υποσύνολο, δείχνει τις περισσότερες πτυχές
  - Αναφορά μνήμης: `lw`, `sw`
  - Αριθμητικές/λογικές: `add`, `sub`, `and`, `or`, `slt`
  - Μεταφοράς ελέγχου: `beq`, `j`

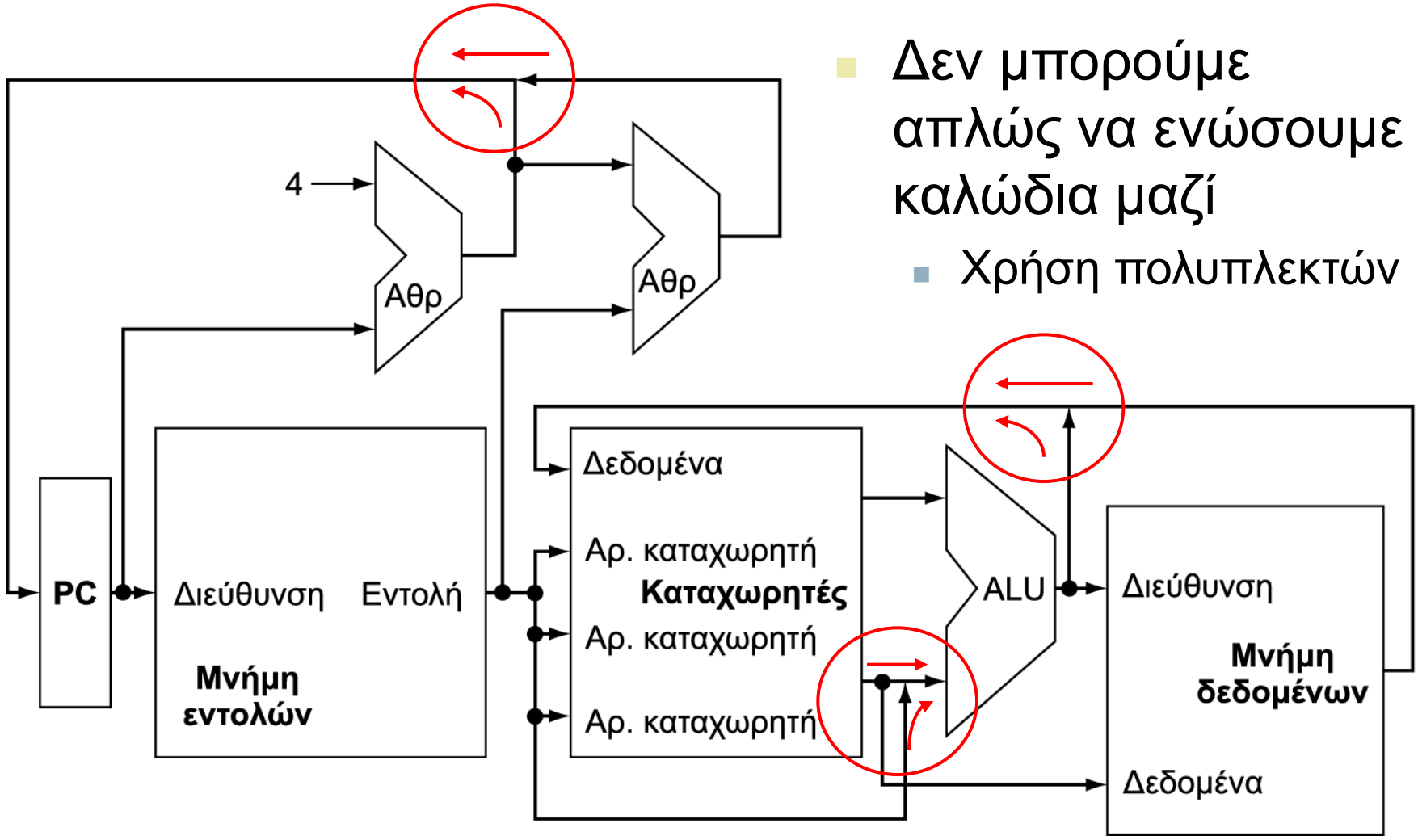
Εκτέλεση εντολής

- PC → μνήμη εντολών, προσκόμιση (fetch) εντολής
- Αριθμοί καταχωρητών → αρχείο καταχωρητών (register file), ανάγνωση καταχωρητών
- Ανάλογα με τη κατηγορία της εντολής
  - Χρήση της ALU για τον υπολογισμό
    - Αριθμητικού αποτελέσματος
    - Διεύθυνσης μνήμης για εντολές load/store
    - Διεύθυνση προορισμού διακλάδωσης
  - Προσπέλαση μνήμης δεδομένων για load/store
  - PC ← διεύθυνση προορισμού ή PC + 4

Επισκόπηση της CPU

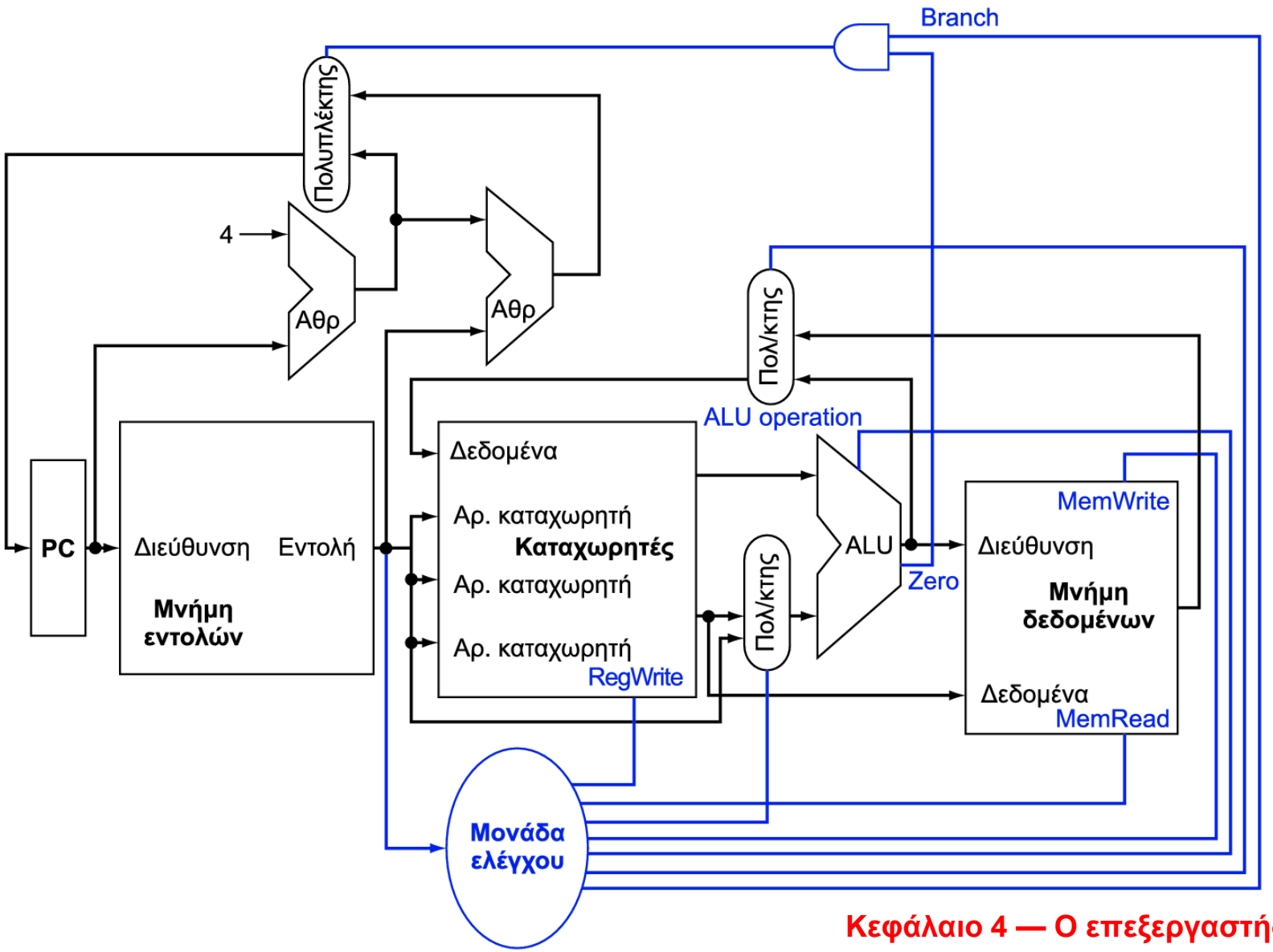


Πολυπλέκτες



- Δεν μπορούμε απλώς να ενώσουμε καλώδια μαζί
  - Χρήση πολυπλεκτών

Έλεγχος

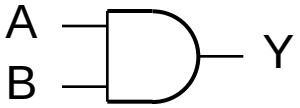

Branch
Πολυπλέκτης
4
Αθρ
Αθρ
Πολ/κτης
ALU operation
PC
Διεύθυνση
Εντολή
Μνήμη εντολών
Δεδομένα
Αρ. καταχωρητή
Καταχωρητές
Αρ. καταχωρητή
Αρ. καταχωρητή
RegWrite
Πολ/κτης
ALU
Zero
MemWrite
Διεύθυνση
Μνήμη δεδομένων
Δεδομένα
MemRead
Μονάδα ελέγχου

Βασικά λογικής σχεδίασης

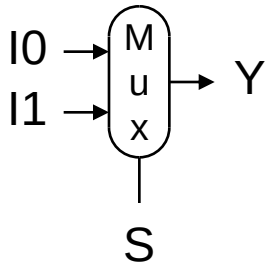
- Η πληροφορία κωδικοποιείται δυαδικά
  - Χαμηλή τάση = 0, Υψηλή τάση = 1
  - Ένα καλώδιο ανά bit
  - Δεδομένα πολλών bit κωδικοποιούνται με διαύλους πολλών καλωδίων
- Συνδυαστικό στοιχείο
  - Επενεργεί σε δεδομένα
  - Η έξοδος είναι συνάρτηση της εισόδου
- Στοιχεία κατάστασης (ακολουθιακά)
  - Αποθηκεύουν πληροφορίες

Συνδυαστικά στοιχεία

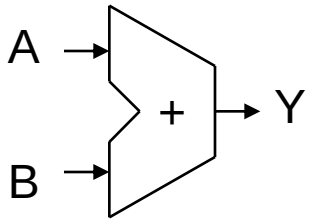
- Πύλη AND
  - Y = A & B



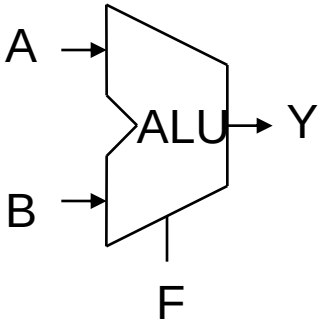
- Πολυπλέκτης
  - Y = S ? I1 : I0



- Αθροιστής
  - Y = A + B

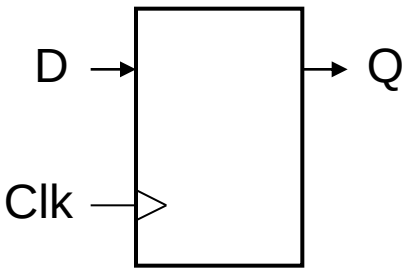


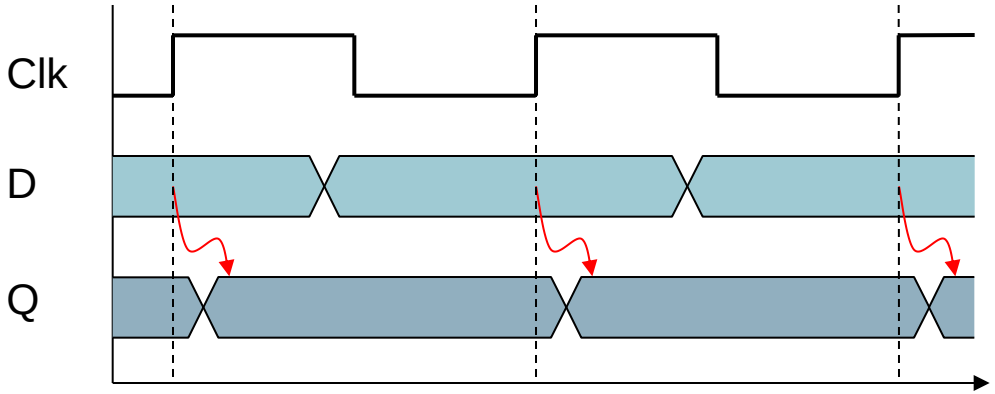
- Αριθμητική/Λογική Μονάδα
  - Y = F(A, B)



Ακολουθιακά στοιχεία

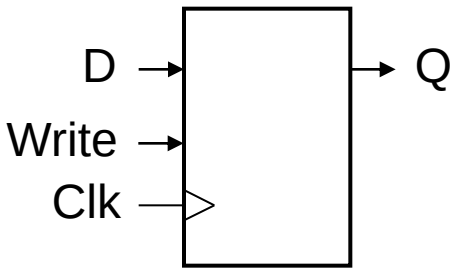
- Καταχωρητής: αποθηκεύει δεδομένα σε ένα κύκλωμα
  - Χρησιμοποιεί σήμα ρολογιού για να καθορίσει πότε ενημερώνεται η αποθηκευμένη τιμή
  - Ακμοπυροδοτούμενη: ενημέρωση όταν το Clk αλλάζει από 0 σε 1

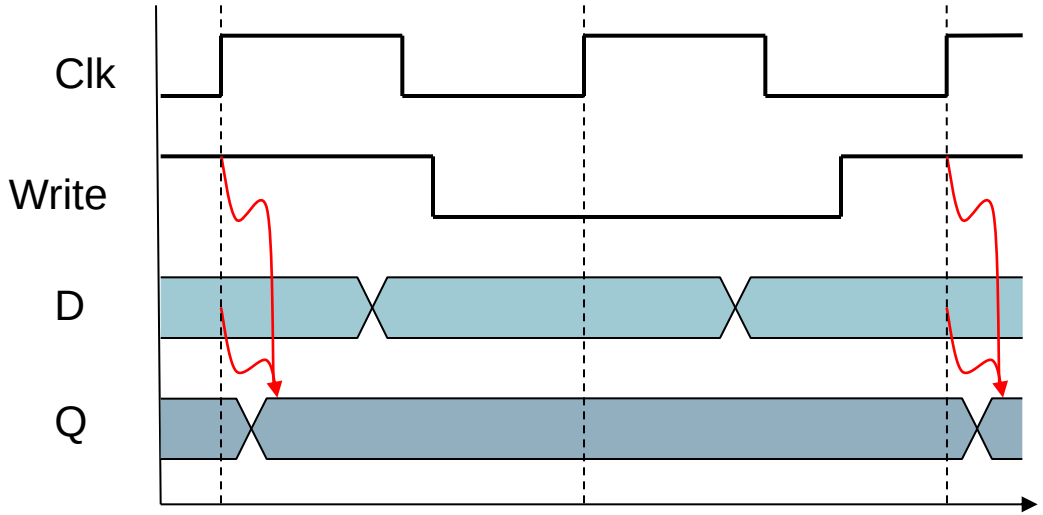




Ακολουθιακά στοιχεία

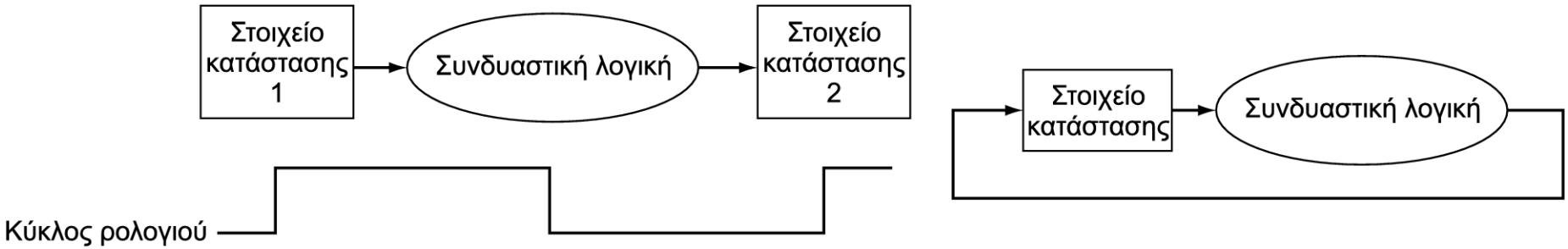
- Καταχωρητής με έλεγχο εγγραφής
  - Ενημερώνει στην ακμή του ρολογιού μόνο όταν η είσοδος ελέγχου εγγραφής είναι 1
  - Χρησιμοποιείται όταν η αποθηκευμένη τιμή απαιτείται αργότερα





Μεθοδολογία χρονισμού

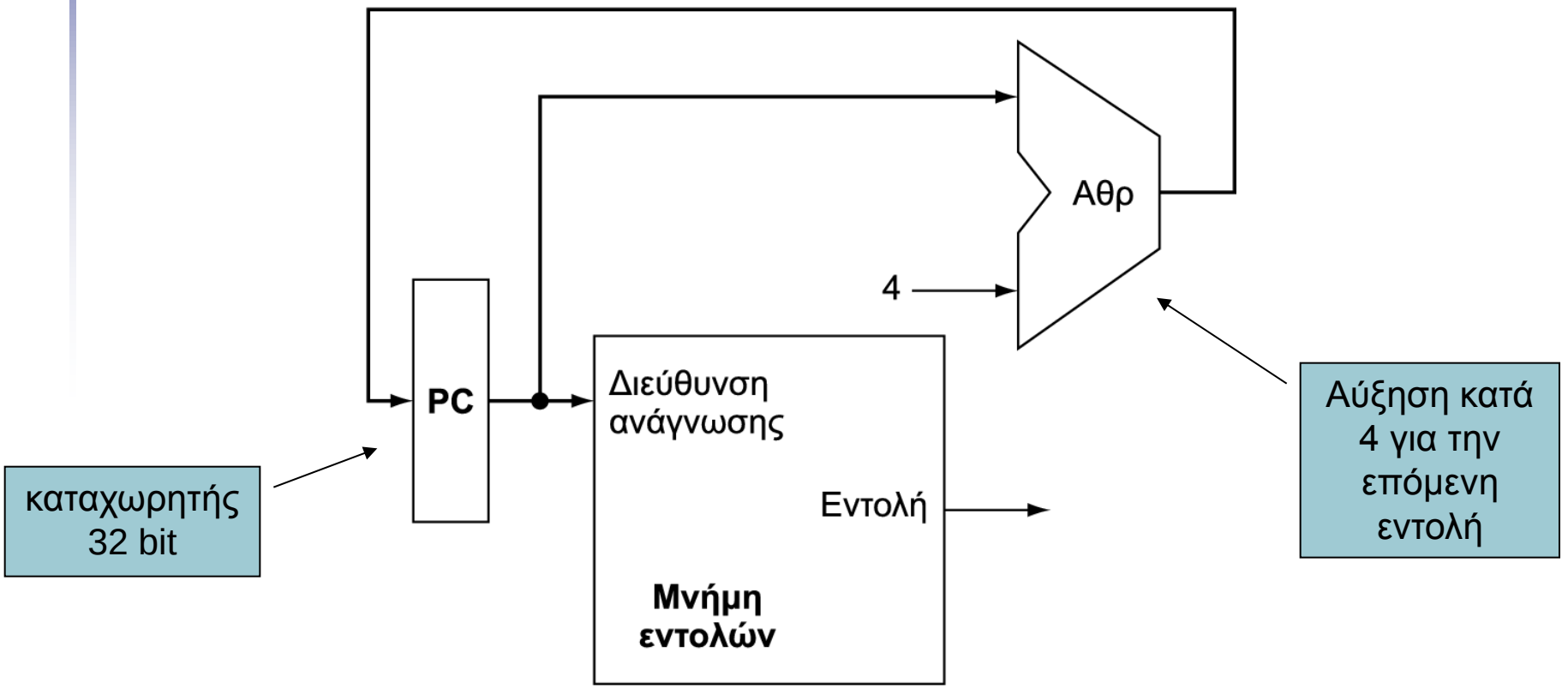
- Η συνδυαστική λογική μετασχηματίζει τα δεδομένα στη διάρκεια των κύκλων ρολογιού
  - Μεταξύ ακμών ρολογιού
  - Είσοδος από στοιχεία κατάστασης, έξοδος σε στοιχεία κατάστασης
  - Η μεγαλύτερη καθυστέρηση καθορίζει την περίοδο του ρολογιού



Κατασκευή διαδρομής δεδομένων

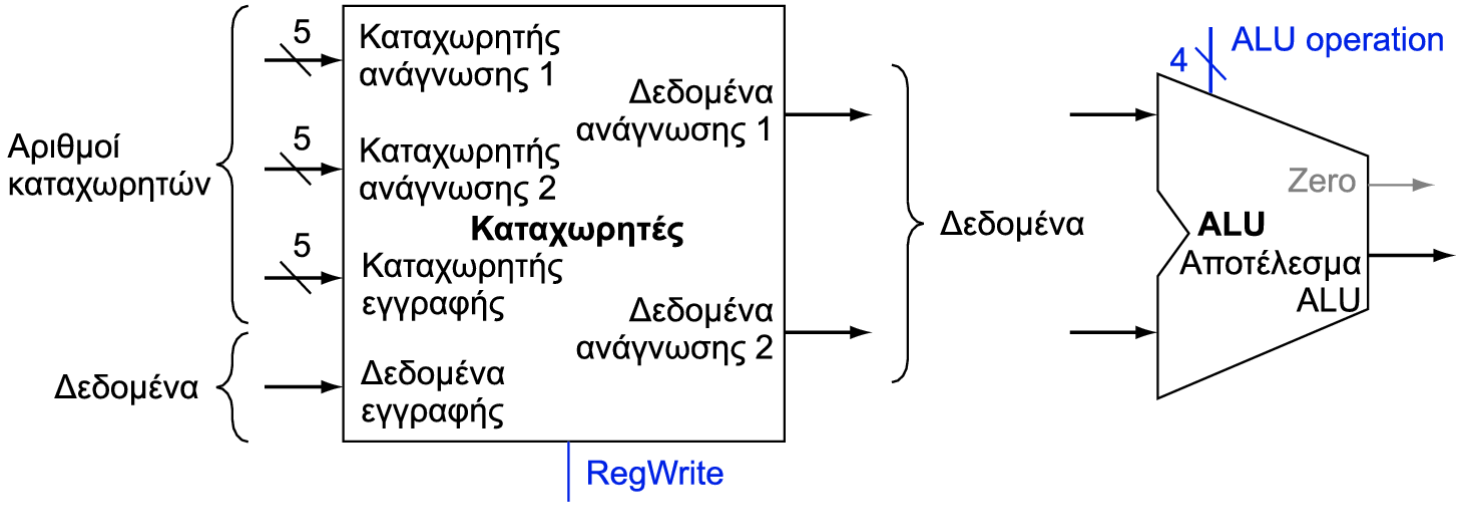
- Διαδρομή δεδομένων (datapath)
  - Στοιχεία που επεξεργάζονται δεδομένα και διευθύνσεις στη CPU
    - Καταχωρητές, ALU, πολυπλέκτες, μνήμες, …
- Θα κατασκευάσουμε μια διαδρομή δεδομένων MIPS με διαδοχικά βήματα
  - Θα κάνουμε πιο αναλυτικό το συνολικό σχέδιο

Προσκόμιση εντολής (Instruction Fetch)



Εντολές μορφής R

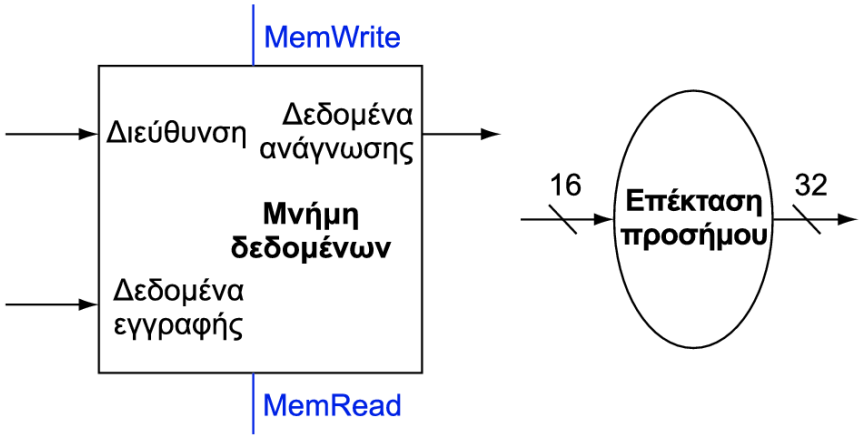
- Ανάγνωση δύο τελεστέων καταχωρητών
- Εκτέλεση αριθμητικής/λογικής λειτουργίας
- Εγγραφή αποτελέσματος σε καταχωρητή



α. Καταχωρητές β. ALU

Εντολές Load/Store

- Ανάγνωση τελεστέων καταχωρητών
- Υπολογισμός διεύθυνσης με χρήση της σχετικής απόστασης (offset) των 16 bit
  - Χρήση της ALU, αλλά με επέκταση προσήμου του offset
- Φόρτωση (Load): ανάγνωση μνήμης και ενημέρωση καταχωρητή
- Αποθήκευση (Store): εγγραφής τιμής καταχωρητή στη μνήμη



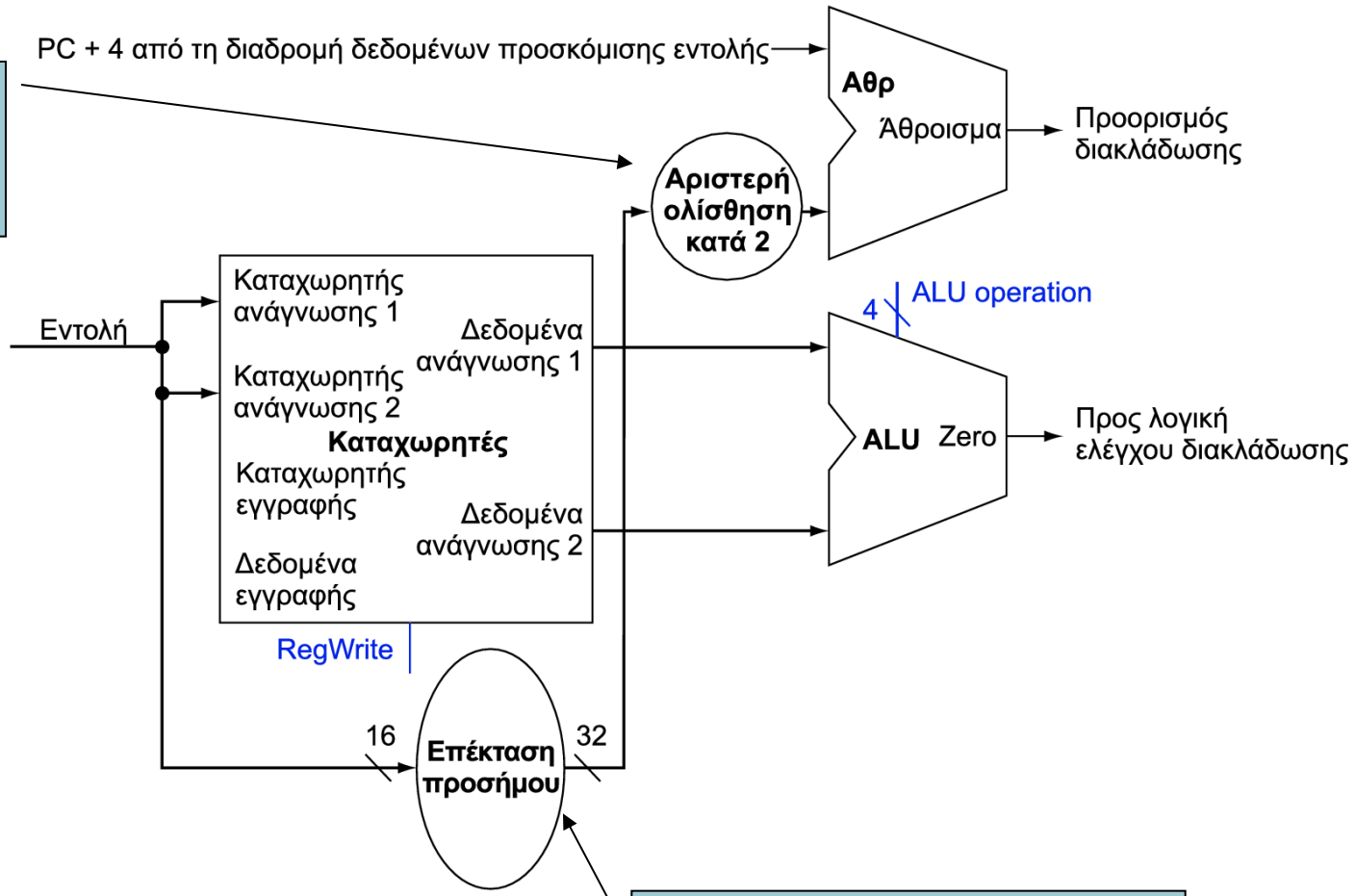
α. Μονάδα μνήμης δεδομένων β. Μονάδα επέκτασης προσήμου

Εντολές διακλάδωσης (branch)

- Ανάγνωση τελεστέων καταχωρητών
- Σύγκριση τελεστέων
  - Χρήση ALU, αφαίρεση και έλεγχος της εξόδου Zero
- Υπολογισμός διεύθυνσης προορισμού
  - Επέκταση προσήμου της μετατόπισης (displacement)
  - Αριστερή ολίσθηση κατά 2 θέσεις (μετατόπιση λέξης)
  - Πρόσθεση στο PC + 4
    - Έχει ήδη υπολογιστεί από την προσκόμιση εντολής

Εντολές διακλάδωσης

Απλώς επαναδρομολογεί τις εγγραφές

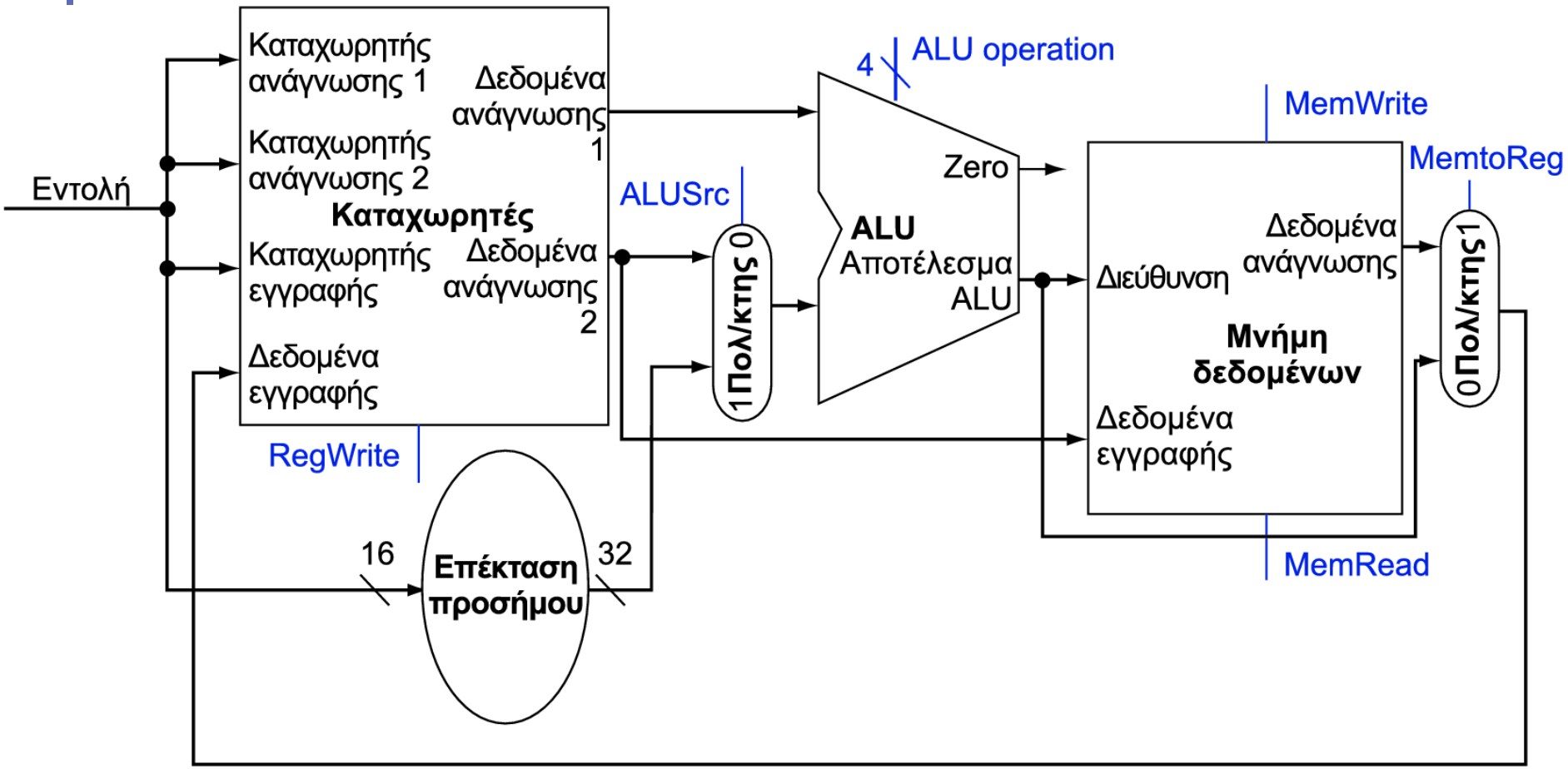


Επαναλαμβάνεται το καλώδιο του bit προσήμου

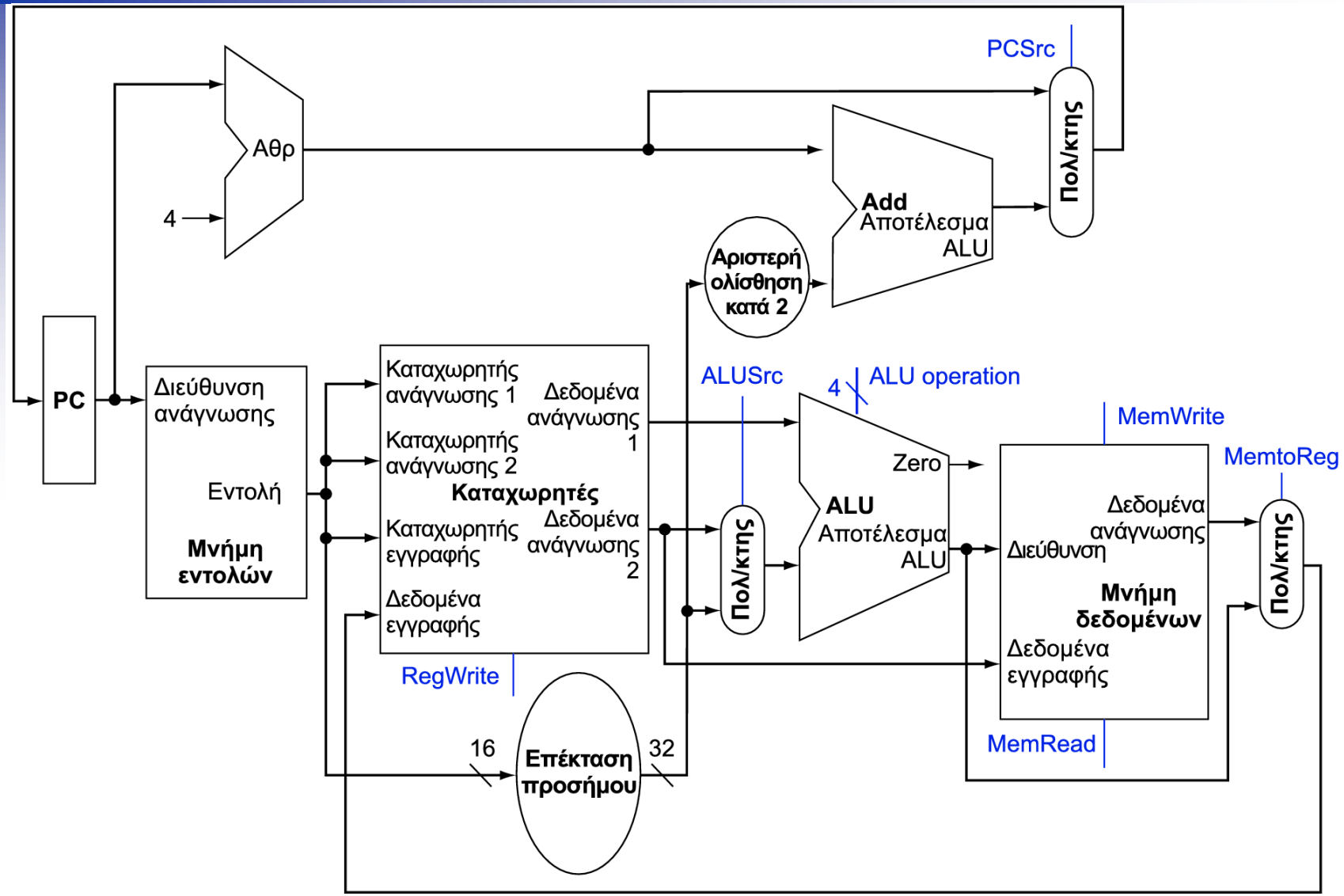
Δημιουργία των στοιχείων

- Μια πρώτη διαδρομή δεδομένων (data path) εκτελεί μία εντολή σε έναν κύκλο ρολογιού
  - Κάθε στοιχείο της διαδρομής δεδομένων κάνει μόνο μία συνάρτηση κάθε φορά
  - Έτσι, χρειαζόμαστε ξεχωριστές μνήμες εντολών και δεδομένων
- Χρήση πολυπλεκτών όταν χρησιμοποιούνται διαφορετικές προελεύσεις δεδομένων σε διαφορετικές εντολές

Διαδρομή δεδομένων για Τύπο R/Load/Store



Πλήρης διαδρομή δεδομένων



Έλεγχος ALU

- Η ALU χρησιμοποιείται για
  - Load/Store: λειτουργία = add
  - Branch: λειτουργία = subtract
  - Τύπου R: λειτουργία εξαρτάται από το πεδίο funct

Έλεγχος ALU	Λειτουργία
0000	AND
0001	OR
0010	add
0110	subtract
0111	set-on-less-than
1100	NOR

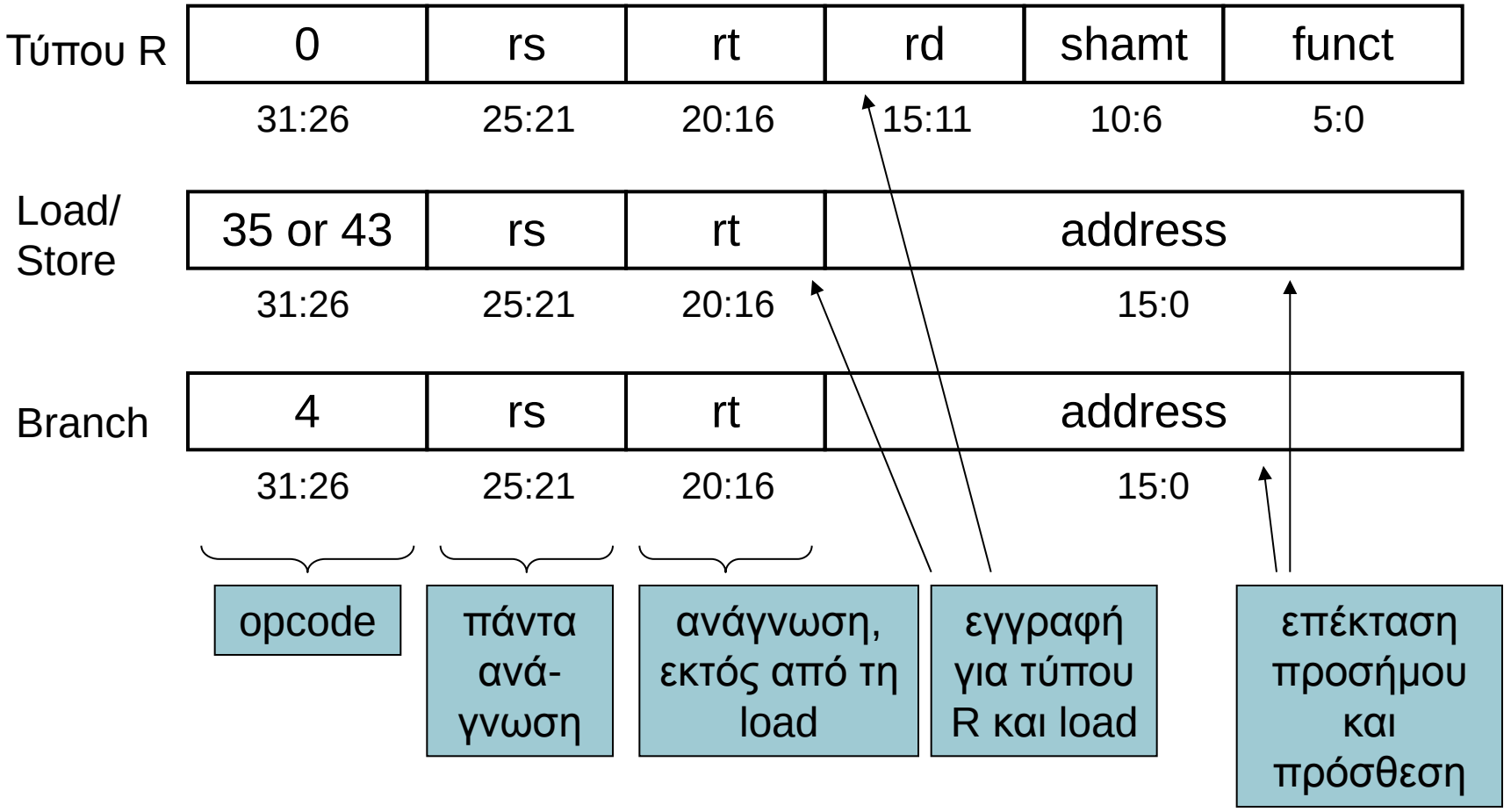
Έλεγχος ALU

- Υποθέτουμε ότι ένα πεδίο 2 bit ALUOp εξάγεται από το opcode
  - Συνδυαστική λογική εξάγει τον έλεγχο της ALU

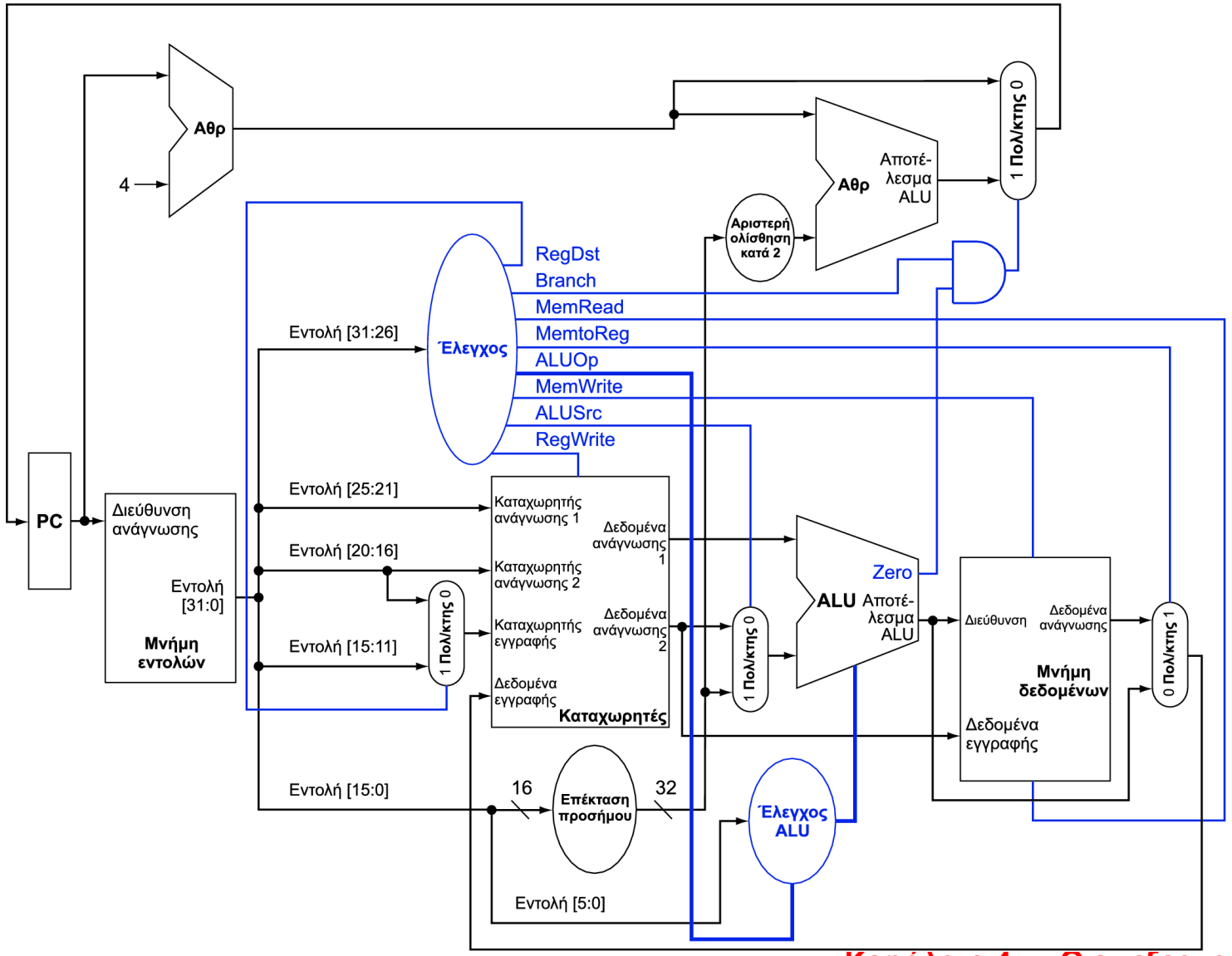
opcode	ALUOp	Λειτουργία	funct	ALU function	ALU control
lw	00	load word	XXXXXX	add	0010
sw	00	store word	XXXXXX	add	0010
beq	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
		subtract	100010	subtract	0110
		AND	100100	AND	0000
		OR	100101	OR	0001
		set-on-less-than	101010	set-on-less-than	0111

Η κύρια μονάδα ελέγχου

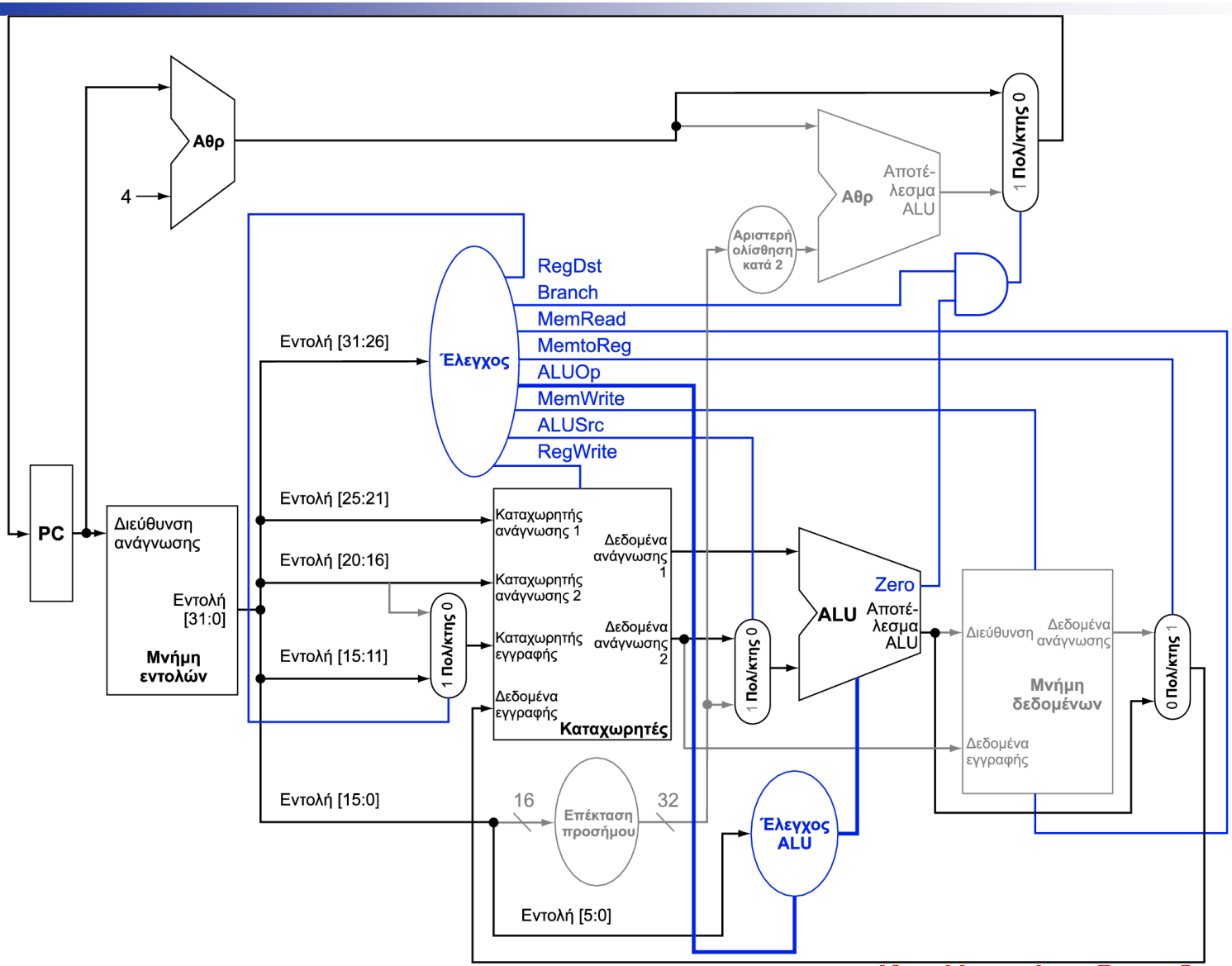
- Σήματα ελέγχου που εξάγονται από εντολή



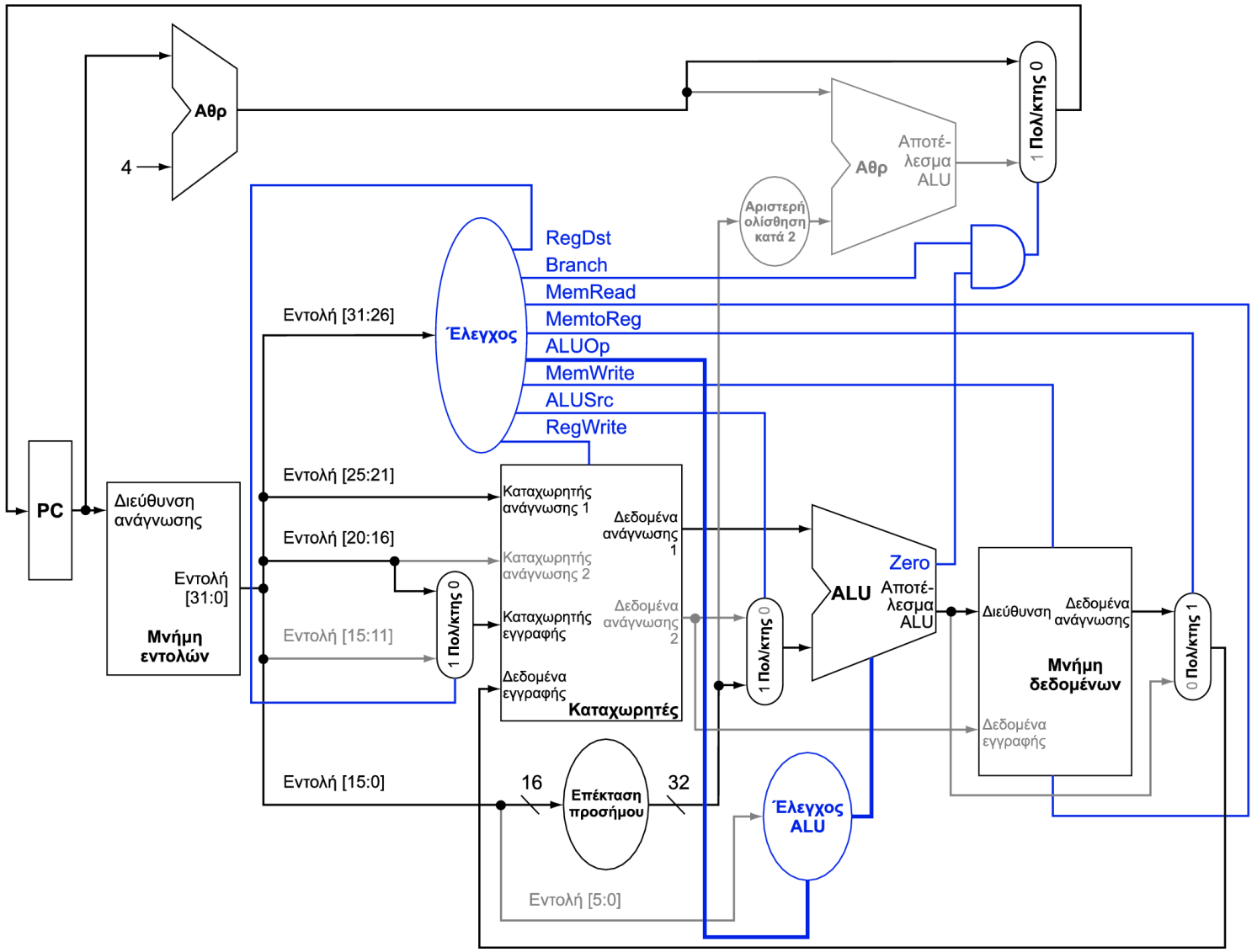
Διαδρομή δεδομένων και έλεγχος



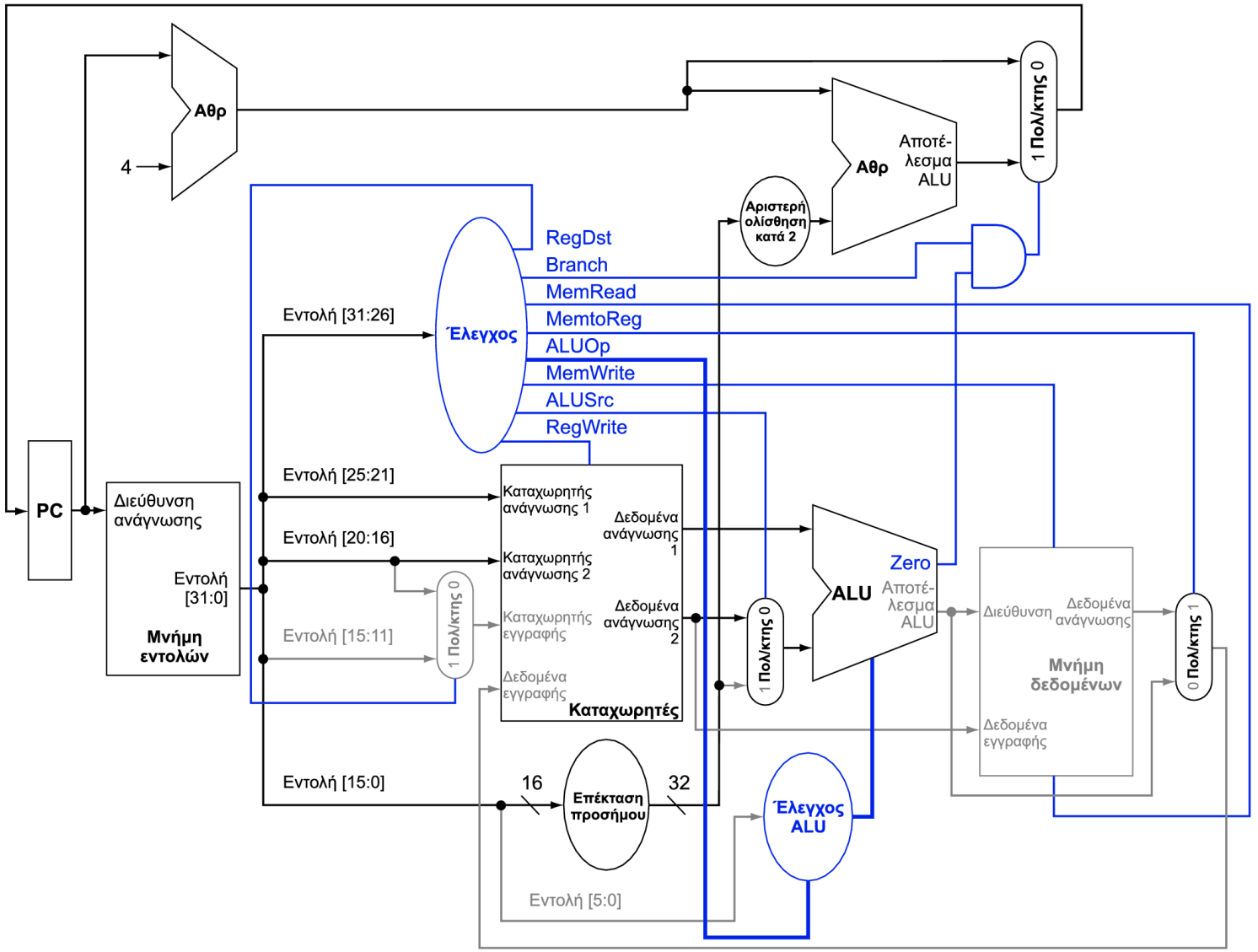
Εντολή τύπου R



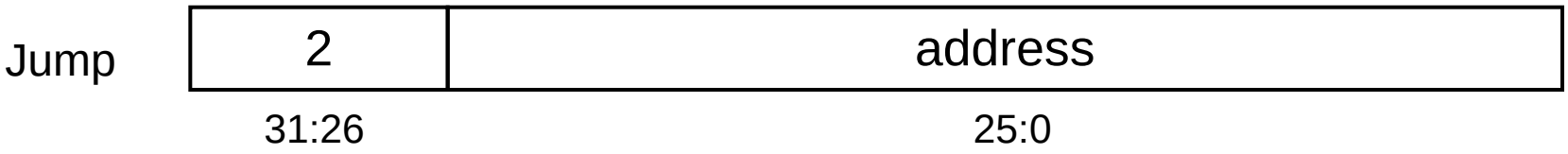
Εντολή Load


Αθρ
4
Αθρ
Αποτέ-
λεσμα
ALU
1 Πολ/κτης 0
Αριστερή
ολίσθηση
κατά 2
RegDst
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Έλεγχος
Εντολή [31:26]
PC
Διεύθυνση
ανάγνωσης
Εντολή
[31:0]
Μνήμη
εντολών
Εντολή [25:21]
Εντολή [20:16]
Εντολή [15:11]
Εντολή [15:0]
Καταχωρητής
ανάγνωσης 1
Καταχωρητής
ανάγνωσης 2
Καταχωρητής
εγγραφής
Δεδομένα
εγγραφής
Δεδομένα
ανάγνωσης
1
Δεδομένα
ανάγνωσης
2
Καταχωρητές
1 Πολ/κτης 0
1 Πολ/κτης 0
ALU
Zero
Αποτέ-
λεσμα
ALU
Διεύθυνση
Δεδομένα
ανάγνωσης
Μνήμη
δεδομένων
Δεδομένα
εγγραφής
0 Πολ/κτης 1
16
Επέκταση
προσήμου
32
Έλεγχος
ALU
Εντολή [5:0]

Εντολή Branch-on-Equal

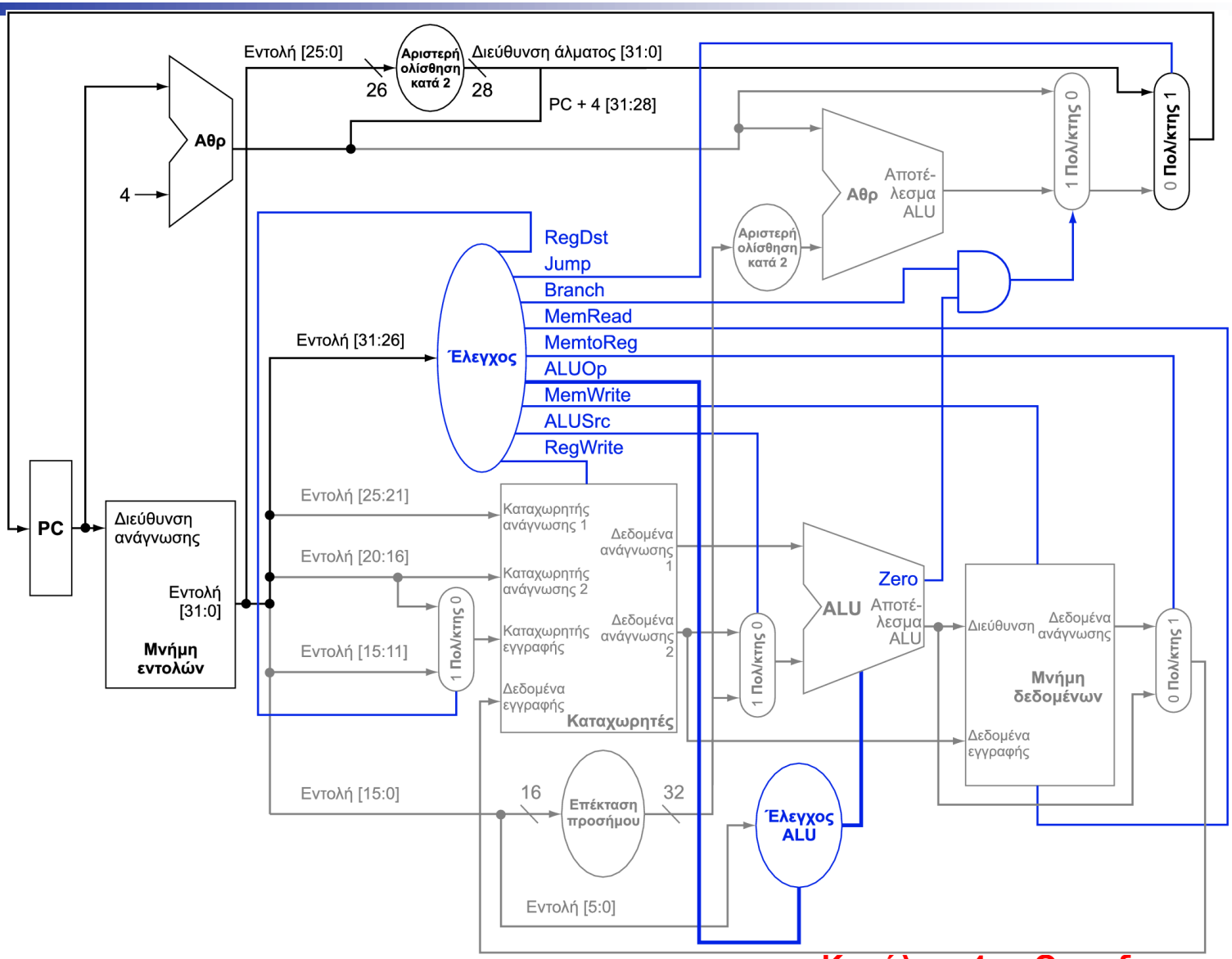


Υλοποίηση αλμάτων



- Η Jump χρησιμοποιεί διεύθυνση λέξης
- Ενημέρωση του PC με συνένωση των
  - Υψηλότερων 4 bit του παλιού PC
  - Διεύθυνσης άλματος των 26 bit
  - 00
- Χρειάζεται ένα επιπλέον σήμα ελέγχου από την αποκωδικοποίηση του opcode

Διαδρομή δεδ. με προσθήκη αλμάτων

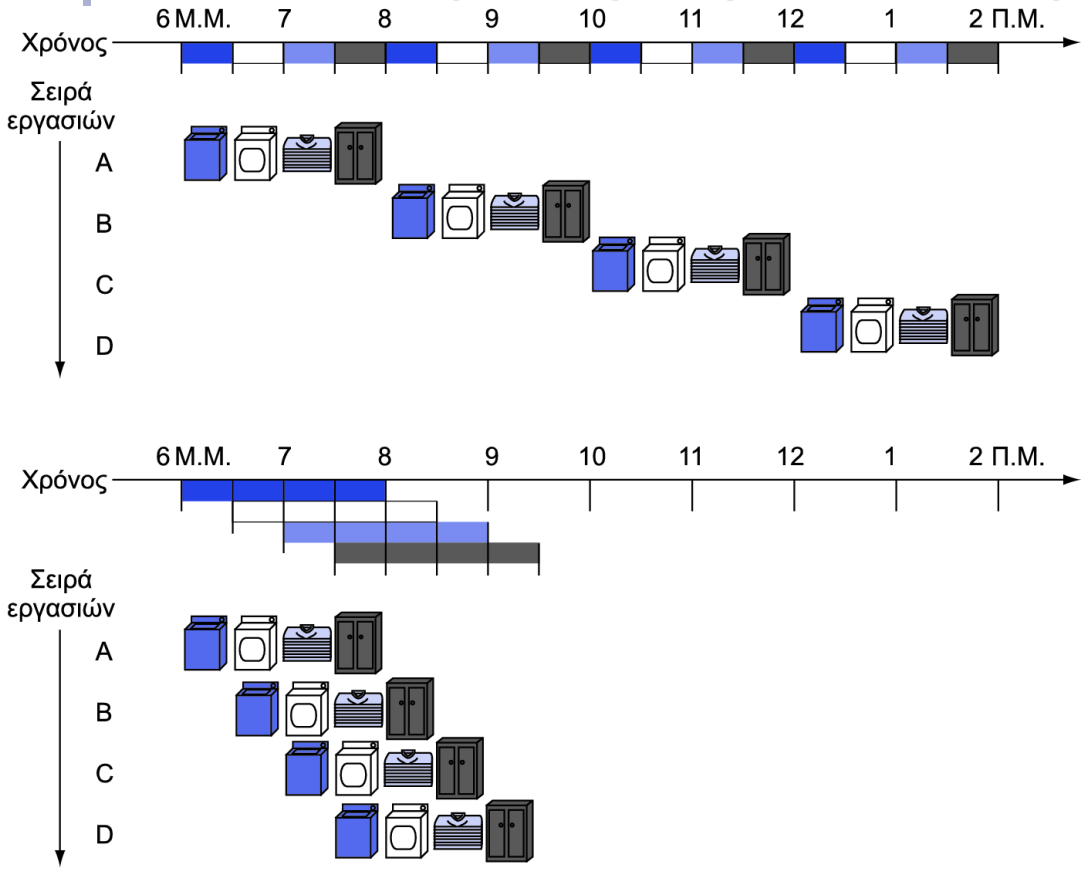


Ζητήματα απόδοσης

- Η μεγαλύτερη καθυστέρηση καθορίζει την περίοδο ρολογιού
  - Κρίσιμη διαδρομή (critical path): εντολή load
  - Μνήμη εντολών → αρχείο καταχωρητών → ALU → μνήμη δεδομένων → αρχείο καταχωρητών
- Δεν είναι εφικτή διαφορετική περίοδος για διαφορετικές εντολές
- Παραβιάζει τη σχεδιαστική αρχή
  - Κάνε τη συνηθισμένη περίπτωση γρήγορη
- Θα βελτιώσουμε την απόδοση με τη διοχέτευση (pipelining)

Αναλογία διοχέτευσης

- Μπουγάδα με διοχέτευση: επικάλυψη εκτέλεσης
  - Η παραλληλία βελτιώνει την απόδοση



- Τέσσερα φορτία:
  - Επιτάχυνση = 8/3.5 = 2.3
- Ασταμάτητα:
  - Επιτάχυνση = 2n/0.5n + 1.5 ≈ 4 = αριθμός σταδίων

Διοχέτευση του MIPS

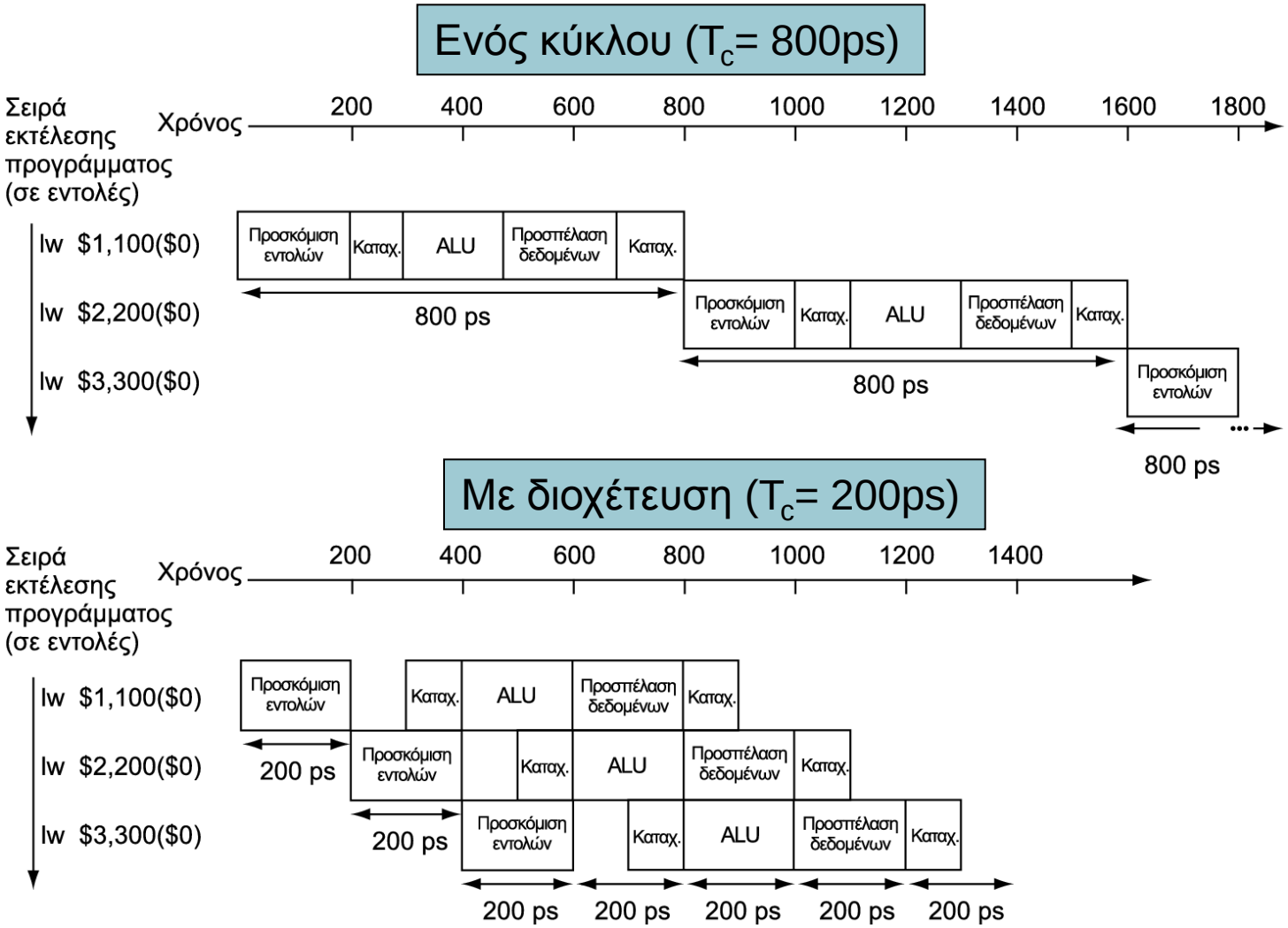
- Πέντε στάδια (stages), ένα βήμα σε κάθε στάδιο
  1. IF: Instruction fetch from memory (προσκόμιση εντολής από τη μνήμη)
  2. ID: Instruction decode & register read (αποκωδικοποίηση εντολής & ανάγνωση καταχωρητών)
  3. EX: Execute operation or calculate address (εκτέλεση λειτουργίας ή υπολογισμός δ/νσης)
  4. MEM: Access memory operand (προσπέλαση τελεστέου μνήμης)
  5. WB: Write result back to register (επανεγγραφή αποτελέσματος σε καταχωρητή)

Απόδοση διοχέτευσης

- Υποθέστε ότι ο χρόνος των σταδίων είναι
  - 100ps για ανάγνωση ή εγγραφή καταχωρητή
  - 200ps για τα άλλα στάδια
- Σύγκριση της διαδρομής δεδομένων με διοχέτευση με τη διαδρομή δεδομένων ενός κύκλου

Εντολή	Instr fetch	Register read	ALU op	Memory access	Register write	Συνολικός χρόνος
lw	200ps	100 ps	200ps	200ps	100 ps	800ps
sw	200ps	100 ps	200ps	200ps		700ps
R-format	200ps	100 ps	200ps		100 ps	600ps
beq	200ps	100 ps	200ps			500ps

Απόδοση διοχέτευσης



Επιτάχυνση λόγω διοχέτευσης

- Αν είναι ισορροπημένα όλα τα στάδια
  - Δηλαδή, όλα διαρκούν τον ίδιο χρόνο
  - $\text{Χρόνος μεταξύ εντολών}_{\text{με διοχέτευση}} = \frac{\text{Χρόνος μεταξύ εντολών}_{\text{χωρίς διοχέτευση}}}{\text{Αριθμός σταδίων}}$
- Αν δεν είναι ισορροπημένα, η επιτάχυνση είναι μικρότερη
- Επιτάχυνση λόγω αυξημένης διεκπεραιωτικής ικανότητας (throughput)
  - Λανθάνων χρόνος – latency (χρόνος για κάθε εντολή) δεν μειώνεται

Διοχέτευση και σχεδίαση συνόλου εντολών

- Το σύνολο εντολών του MIPS είναι σχεδιασμένο για διοχέτευση
  - Όλες οι εντολές είναι των 32 bit
    - Ευκολότερη προσκόμιση και αποκωδικοποίηση σε έναν κύκλο
    - σύγκριση με x86: εντολές 1 έως 17 byte
  - Λίγες και κανονικές μορφές εντολών
    - Μπορεί να αποκωδικοποιήσει και να διαβάσει καταχωρητές σε ένα βήμα
  - Διευθυνσιοδότηση Load/store
    - Μπορεί να υπολογίσει τη δ/νση στο τρίτο στάδιο, και να προσπελάσει τη μνήμη στο τέταρτο στάδιο
  - Ευθυγράμμιση των τελεστέων μνήμης
    - Προσπέλαση μνήμης διαρκεί μόνο έναν κύκλο

Κίνδυνοι (hazards)

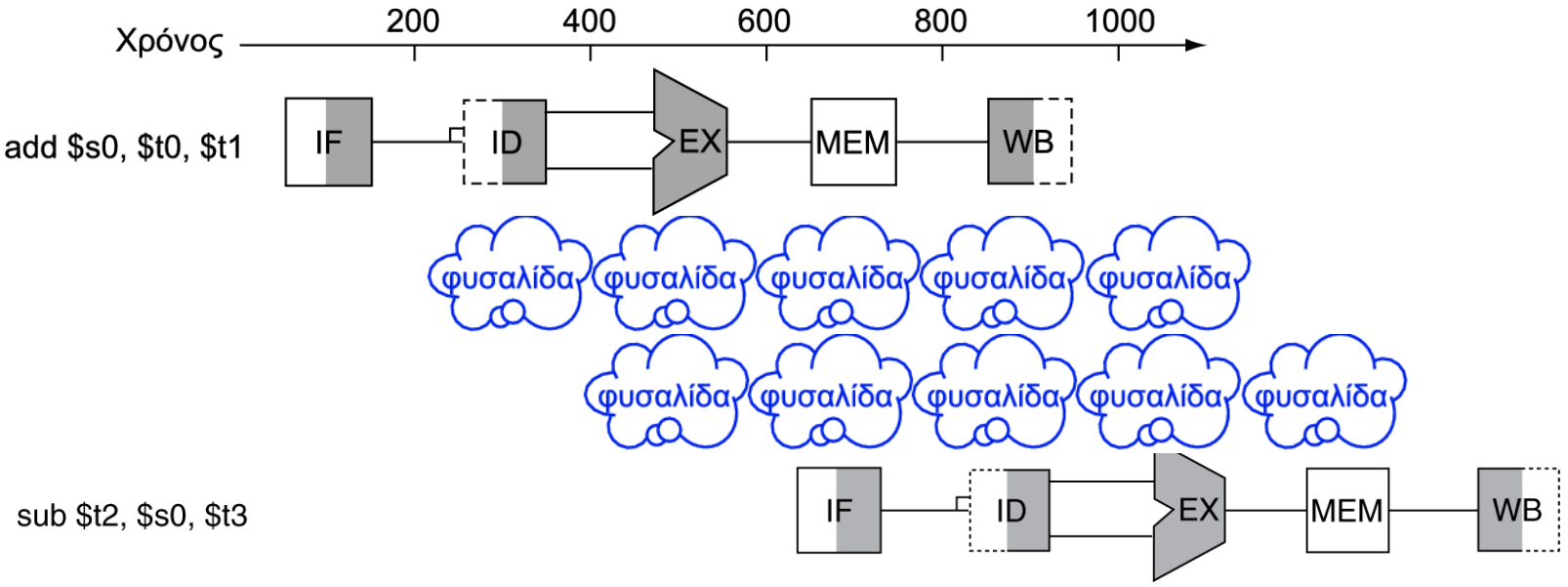
- Καταστάσεις που αποτρέπουν την εκκίνηση της επόμενης εντολής στον επόμενη κύκλο
- Κίνδυνος **δομής** (structure hazards)
  - Ένας απαιτούμενος πόρος είναι απασχολημένος
- Κίνδυνος **δεδομένων** (data hazard)
  - Πρέπει να περιμένει την προηγούμενη εντολή να ολοκληρώσει την ανάγνωση/εγγραφή δεδομένων της
- Κίνδυνος **ελέγχου** (control hazard)
  - Η απόφαση σε μια ενέργεια ελέγχου εξαρτάται από προηγούμενη εντολή

Κίνδυνοι δομής

- Διένεξη στη χρήση ενός πόρου
- Στη διοχέτευση του MIPS με μία μοναδική μνήμη
  - Οι εντολές load/store απαιτούν προσπέλαση μνήμης
  - Η προσκόμιση εντολής πρέπει να *καθυστερήσει* (*stall*) σε εκείνο το κύκλο
    - Θα προκαλούσε «φυσαλίδα» της διοχέτευσης (pipeline “bubble”)
- Έτσι, οι διαδρομές δεδομένων με διοχέτευση απαιτούν ξεχωριστές μνήμες εντολών/δεδομένων
  - Ή ξεχωριστές κρυφές μνήμες (cache memories) εντολών/δεδομένων

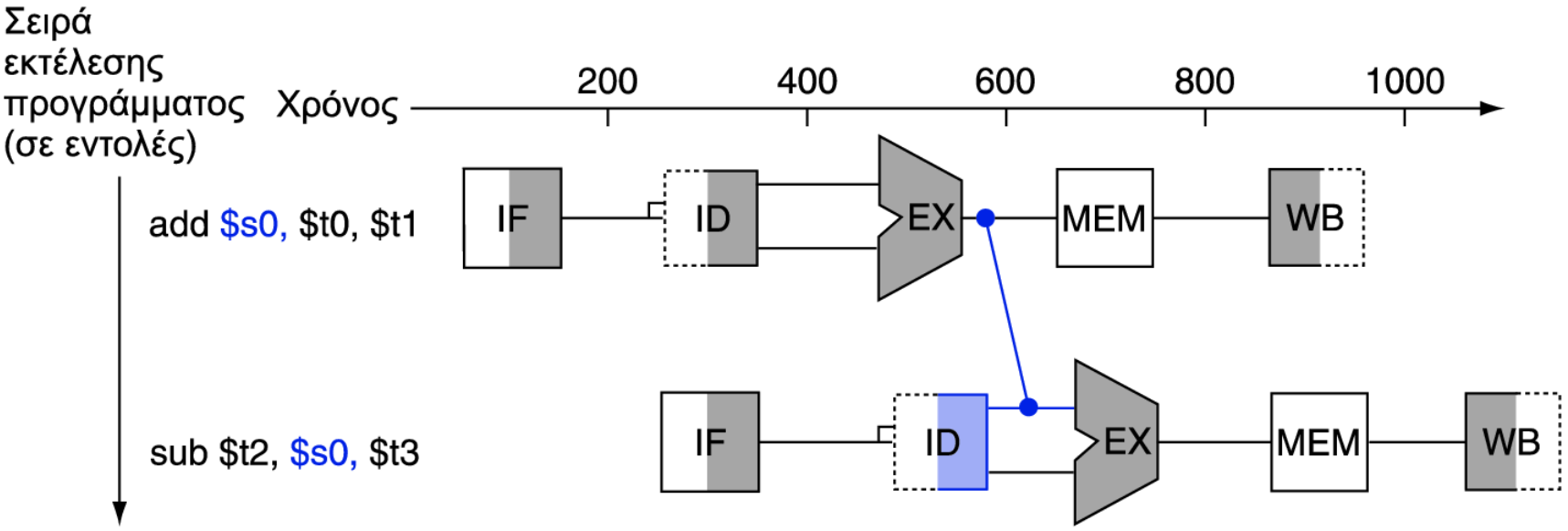
Κίνδυνοι δεδομένων

- Μια εντολή εξαρτάται από την ολοκλήρωση μιας προσπέλασης δεδομένων μιας προηγούμενης εντολής
  - `add $s0, $t0, $t1`
    `sub $t2, $s0, $t3`



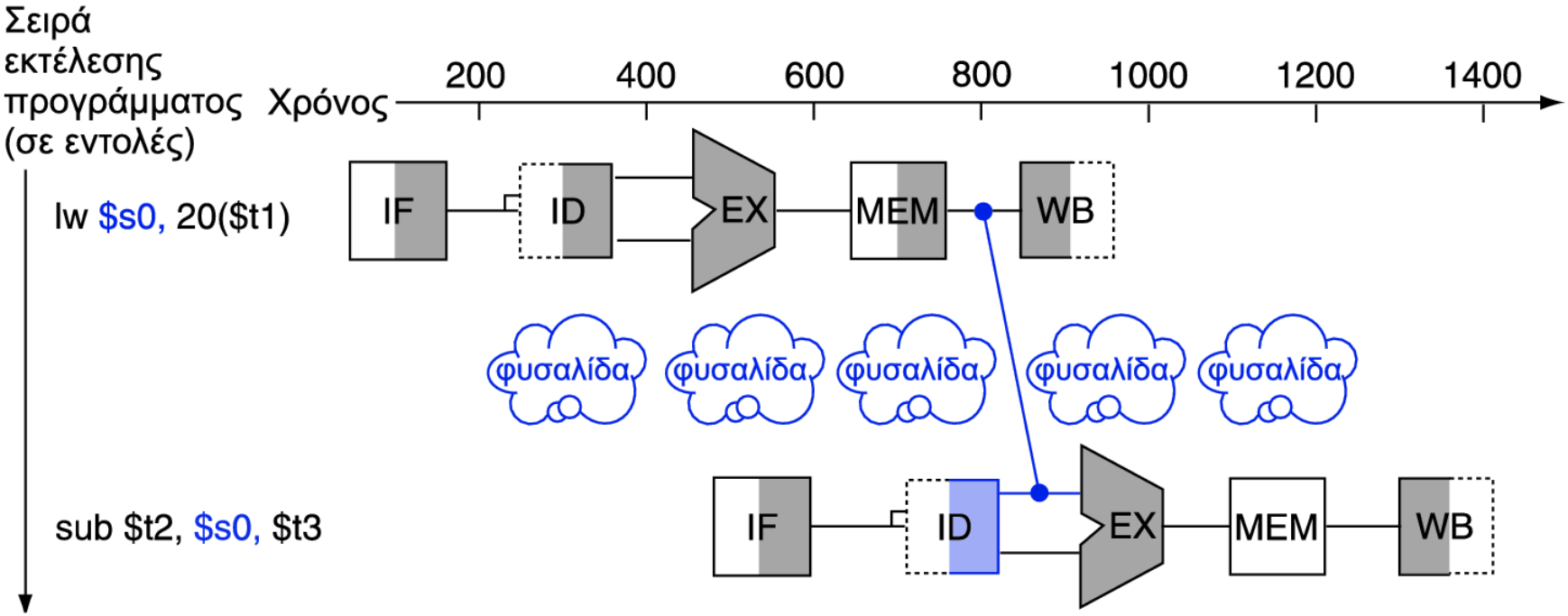
Προώθηση (Forwarding)

- Λέγεται επίσης Παράκαμψη (Bypassing)
- Χρήση του αποτελέσματός όταν δημιουργηθεί
  - Μην περιμένεις να αποθηκευτεί σε καταχωρητή
  - Απαιτεί επιπλέον συνδέσεις στη διαδρομή δεδομένων



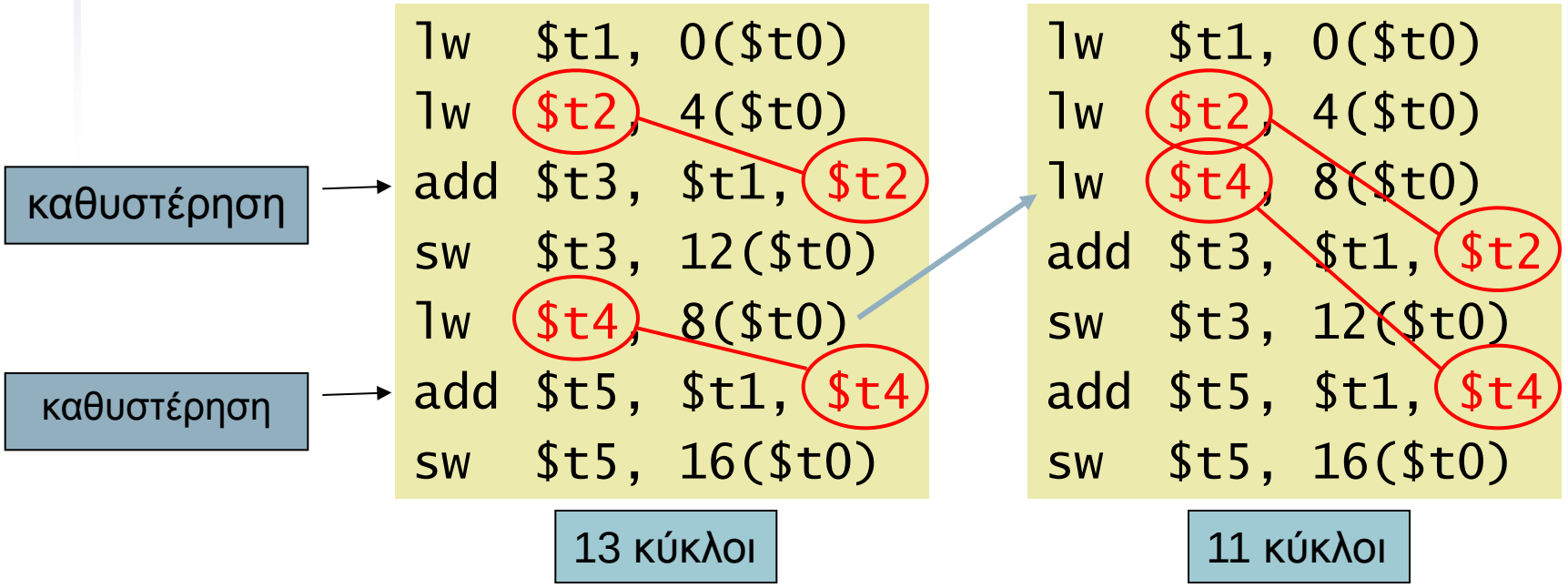
Κίνδυνος δεδομένων Φόρτωσης/Χρήσης

- Φόρτωση/Χρήση (Load/Use)
- Δεν μπορούμε να αποφύγουμε πάντα τις καθυστερήσεις με την προώθηση
  - Αν η τιμή δεν έχει υπολογιστεί όταν χρειάζεται
  - Δεν μπορεί να γίνει προώθηση προς τα πίσω στο χρόνο!



Χρονοπρογρ/μός κώδικα για αποφυγή καθυστερήσεων

- Αναδιάταξη κώδικα για αποφυγή χρήσης του αποτελέσματος της load στην επόμενη εντολή
- Κώδικας C για το `A = B + E; C = B + F;`

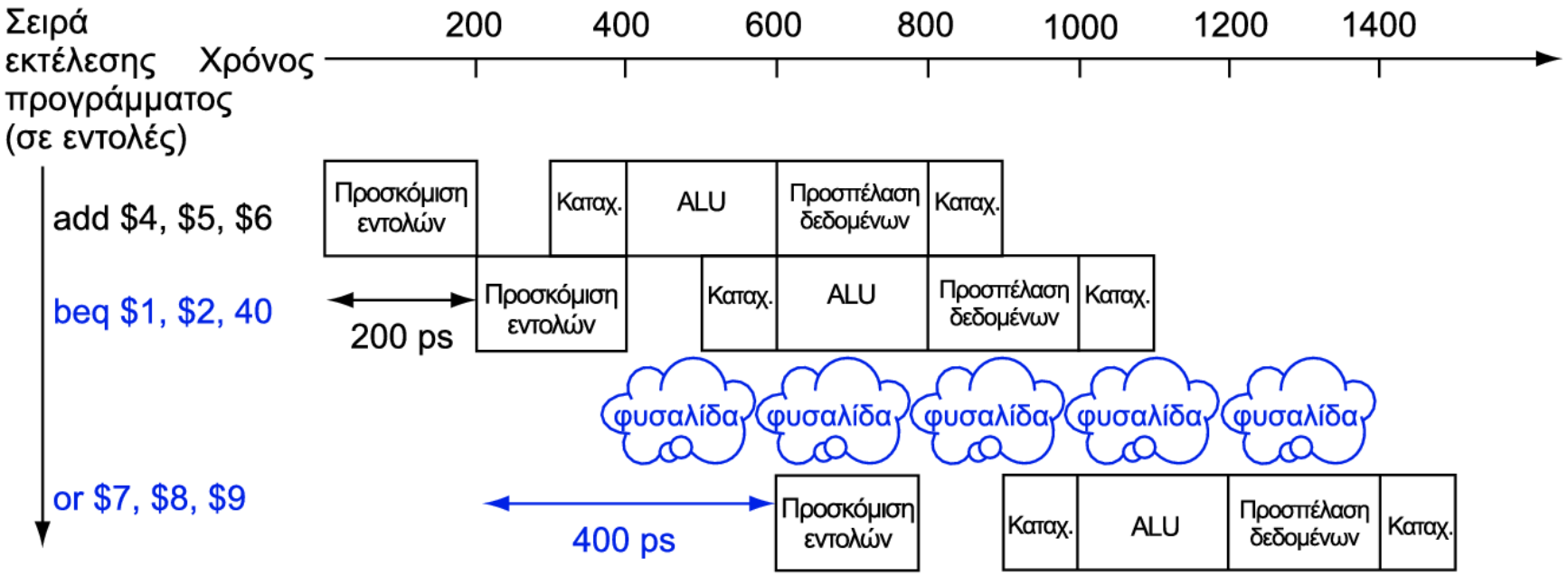


Κίνδυνοι ελέγχου

- Η διακλάδωση καθορίζει τη ροή του ελέγχου (flow of control)
  - Η προσκόμιση της επόμενης εντολής εξαρτάται από το αποτέλεσμα της διακλάδωσης
  - Η διοχέτευση δεν μπορεί να προσκομίσει πάντα τη σωστή εντολή
    - Ακόμη δουλεύει στο στάδιο ID της διακλάδωσης
- Στη διοχέτευση του MIPS
  - Πρέπει να συγκρίνει καταχωρητές και να υπολογίσει τη δ/νση προορισμού νωρίς στη διοχέτευση
  - Προσθήκη υλικού για να γίνουν στο στάδιο ID

Καθυστέρηση σε διακλάδωση

- Stall on branch
- Περίμενε μέχρι να καθοριστεί το αποτέλεσμα της διακλάδωσης πριν προσκομίσεις την επόμενη εντολή

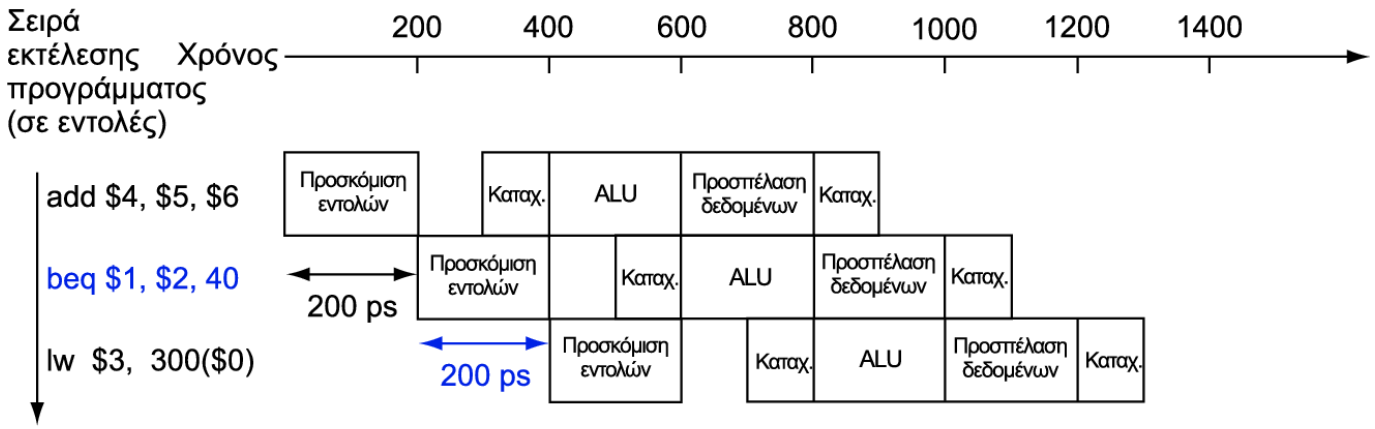


Πρόβλεψη διακλάδωσης

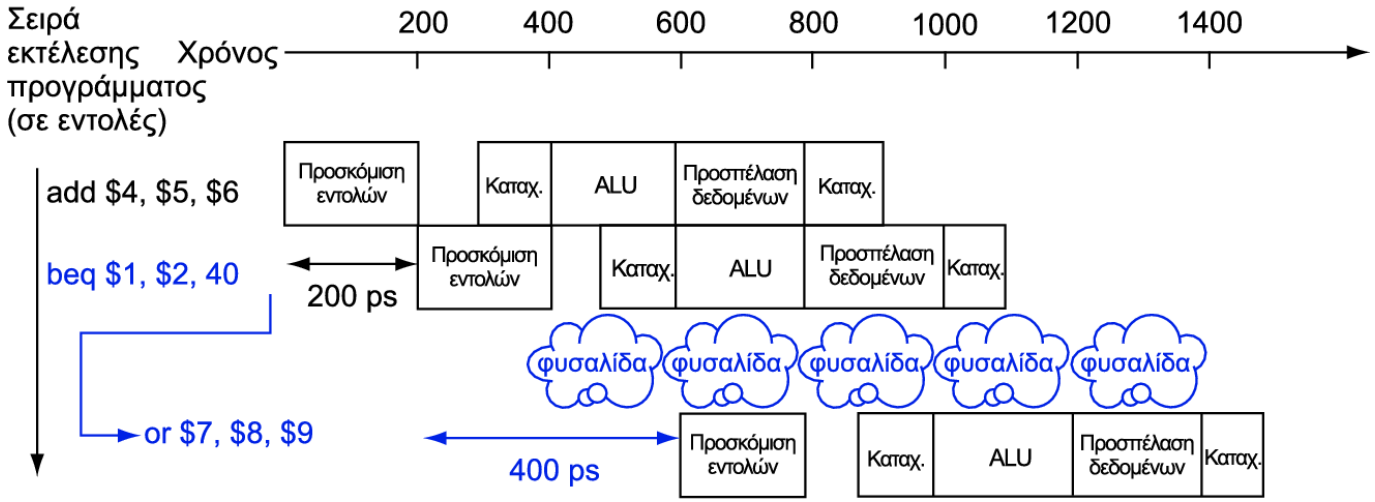
- Branch prediction
- Οι μεγαλύτερες διοχετεύσεις δεν μπορούν να καθορίσουν σύντομα το αποτέλεσμα της διακλάδωσης
  - Η ποινή καθυστέρησης (stall penalty) γίνεται υπερβολικά μεγάλη
- Πρόβλεψη (predict) του αποτελέσματος της διακλάδωσης
  - Καθυστέρηση μόνο αν η πρόβλεψη είναι λανθασμένη
- Στη διοχέτευση του MIPS
  - Μπορεί να γίνει πρόβλεψη μη λήψης της διακλάδωσης (predict not taken)
  - Προσκόμιση της εντολής μετά τη διακλάδωση, χωρίς καθόλου καθυστέρηση

MIPS με πρόβλεψη μη λήψης

Σωστή πρόβλεψη



Λανθασμένη πρόβλεψη



Πιο ρεαλιστική πρόβλεψη διακλάδωσης

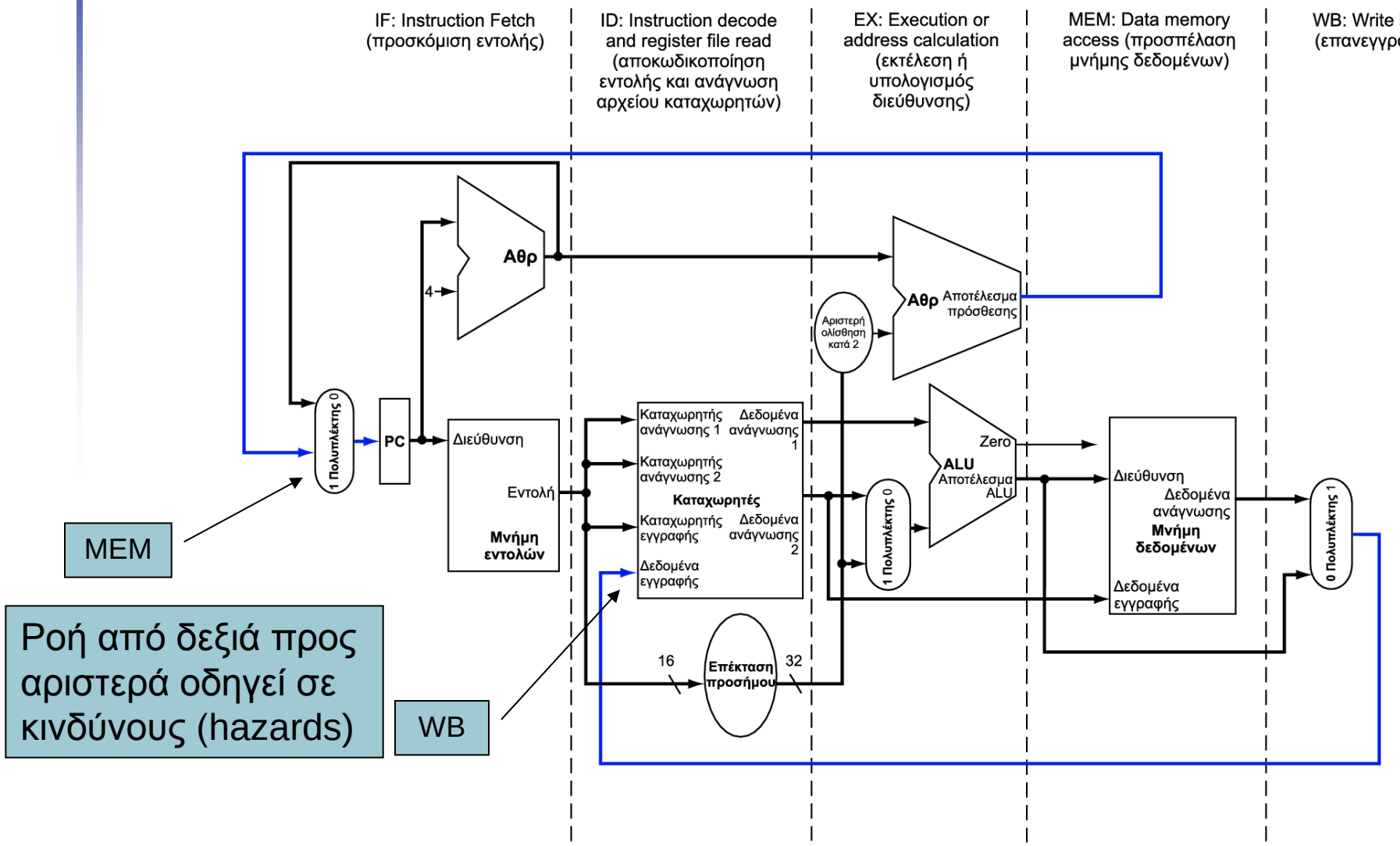
- Στατική πρόβλεψη διακλάδωσης
  - Βασίζεται στην τυπική συμπεριφορά των διακλαδώσεων
  - Παράδειγμα: διακλαδώσεις σε βρόχους και εντολές if
    - Πρόβλεψη διακλαδώσεων προς τα πίσω (backward branches) ως λαμβανόμενες
    - Πρόβλεψη διακλαδώσεων προς τα εμπρός (forward branches) ως μη λαμβανόμενες
- Δυναμική πρόβλεψη διακλάδωσης
  - Το υλικό μετράει την πραγματική συμπεριφορά διακλαδώσεων
    - π.χ., καταγράφει την πρόσφατη ιστορία κάθε διακλάδωσης
  - Υποθέτει ότι η μελλοντική συμπεριφορά θα συνεχίσει την τάση
    - Σε περίπτωση λάθους, γίνεται καθυστέρηση κατά την επαναπροσκόμιση, και ενημέρωση του ιστορικού

Περίληψη διοχέτευσης

ΓΕΝΙΚΗ εικόνα

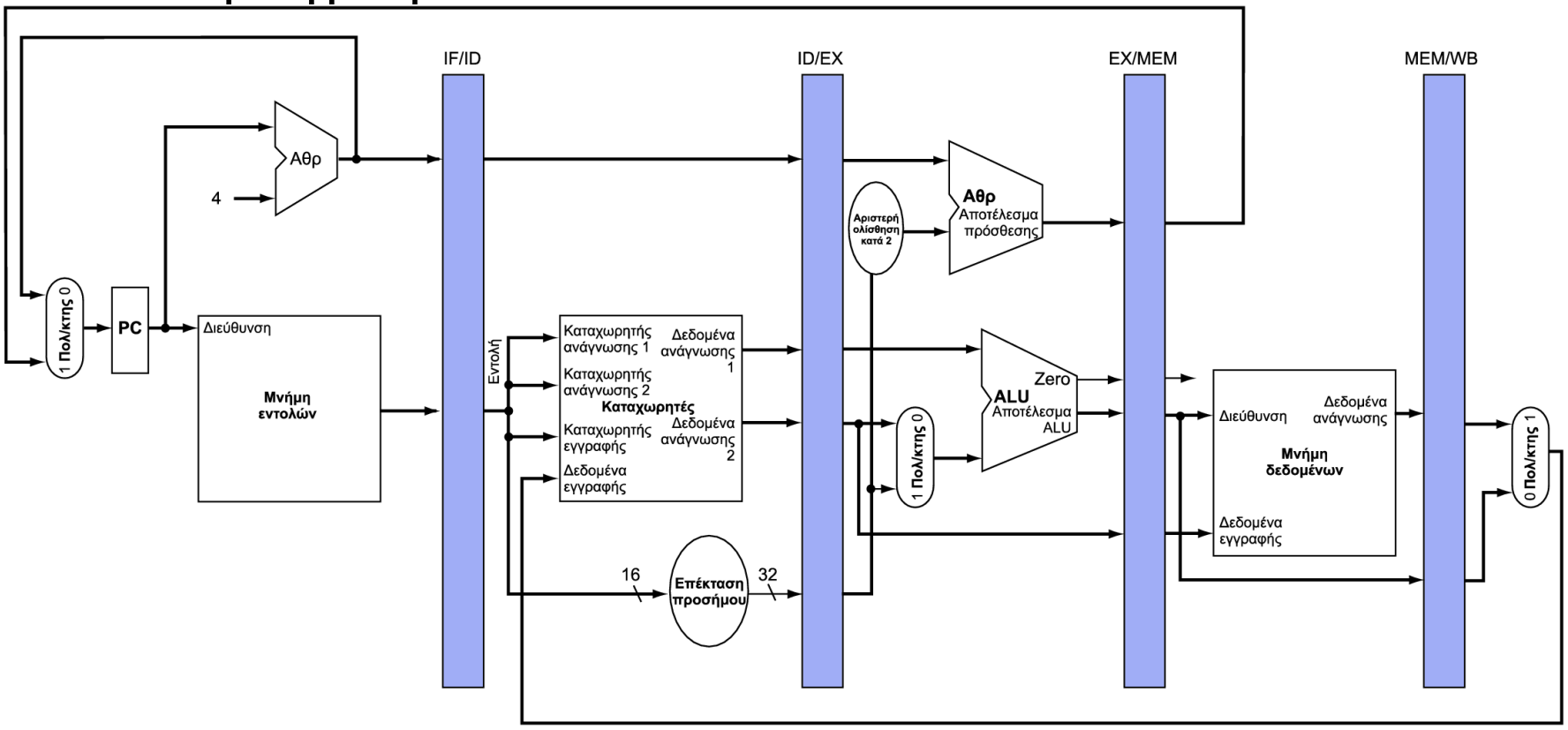
- Η διοχέτευση βελτιώνει την απόδοση με την αύξηση της διεκπεραιωτικής ικανότητας σε εντολές
  - Εκτελεί πολλές εντολές παράλληλα
  - Κάθε εντολή έχει τον ίδιο λανθάνοντα χρόνο
- Υπόκειται σε κινδύνους
  - Δομής, δεδομένων, ελέγχου
- Η σχεδίαση του συνόλου εντολών επιδρά στην πολυπλοκότητα της υλοποίησης της διοχέτευσης

Διαδρομή δεδομένων MIPS με διοχέτευση



Καταχωρητές διοχέτευσης

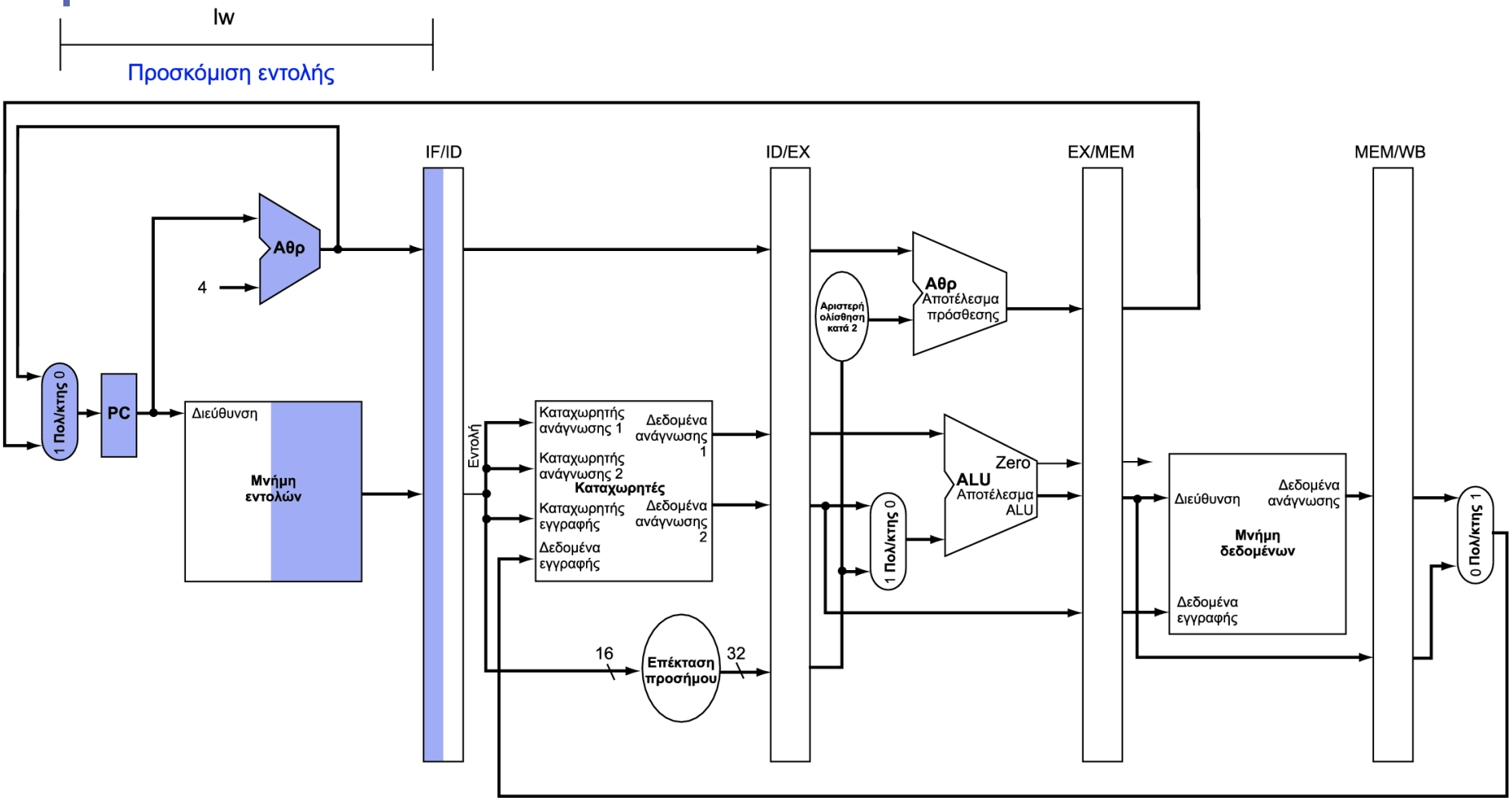
- Χρειάζονται καταχωρητές ανάμεσα στα στάδια
  - Για να κρατήσουν πληροφορίες που παράγονται στον προηγούμενο κύκλο



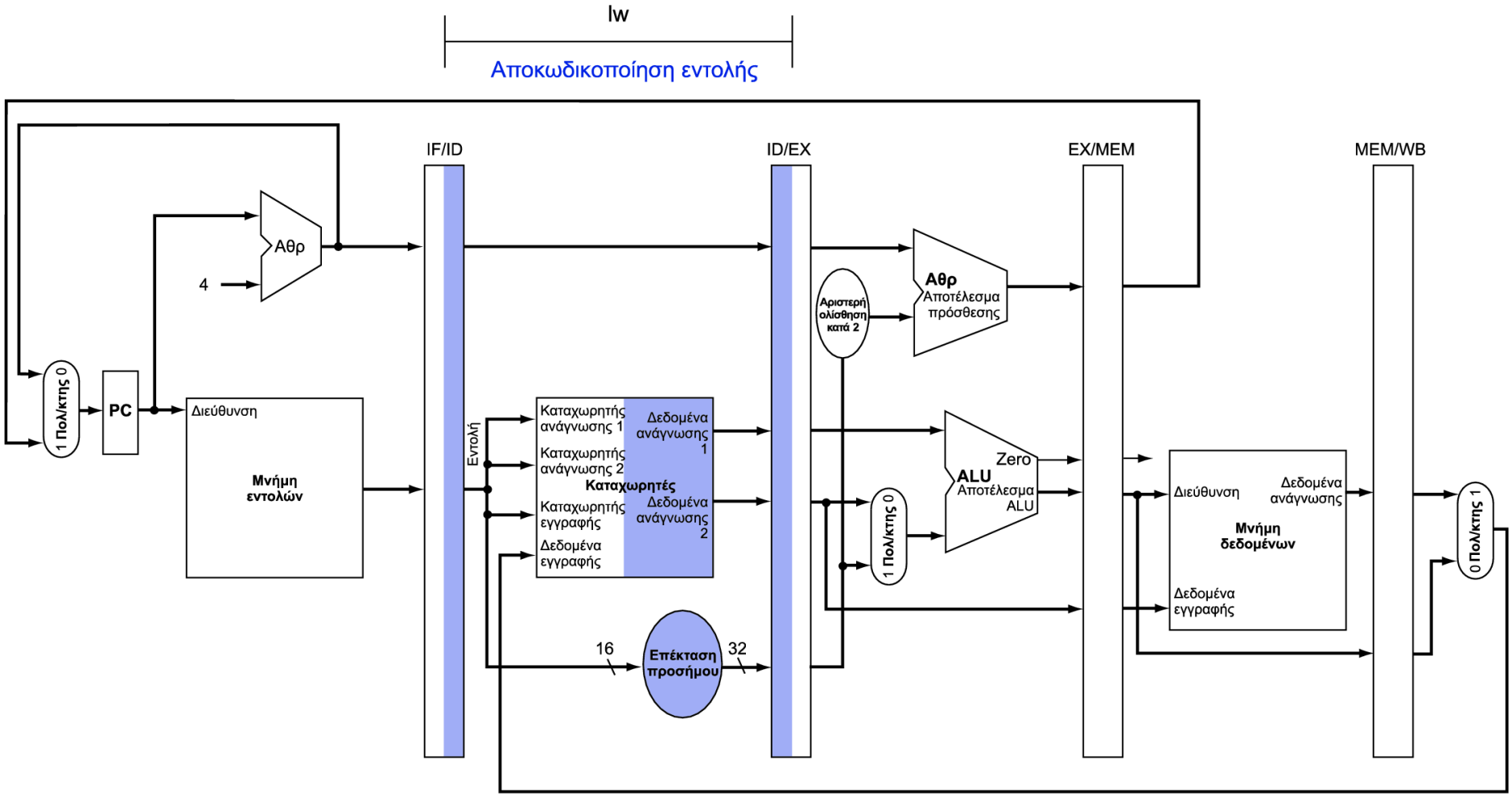
Λειτουργία διοχέτευσης

- Ανά κύκλο, ροή των εντολών μέσα από τη διαδρομή δεδομένων με διοχέτευση
  - Διάγραμμα διοχέτευσης «ενός κύκλου ρολογιού» (“single-clock-cycle”)
    - Δείχνει τη χρήση της διοχέτευσης σε ένα μόνο κύκλο
    - Τονίζει τους πόρους που χρησιμοποιούνται
  - Σύγκριση με διάγραμμα «πολλών κύκλων ρολογιού» (“multi-clock-cycle”)
    - Γράφημα της λειτουργίας στο χρόνο
- Θα δούμε διαγράμματα «ενός κύκλου ρολογιού» για εντολές load και store

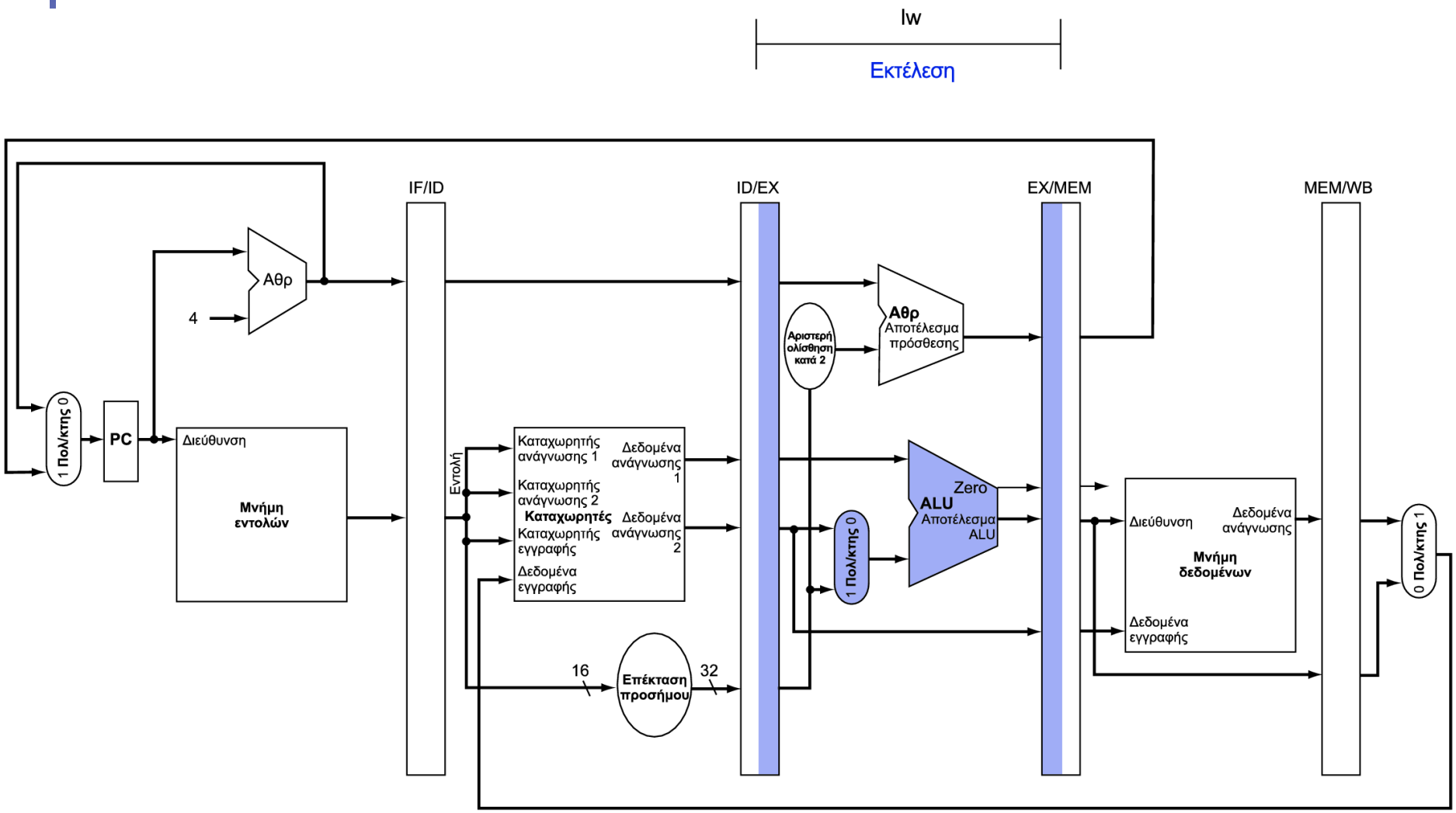
Στάδιο IF για Load, Store, …



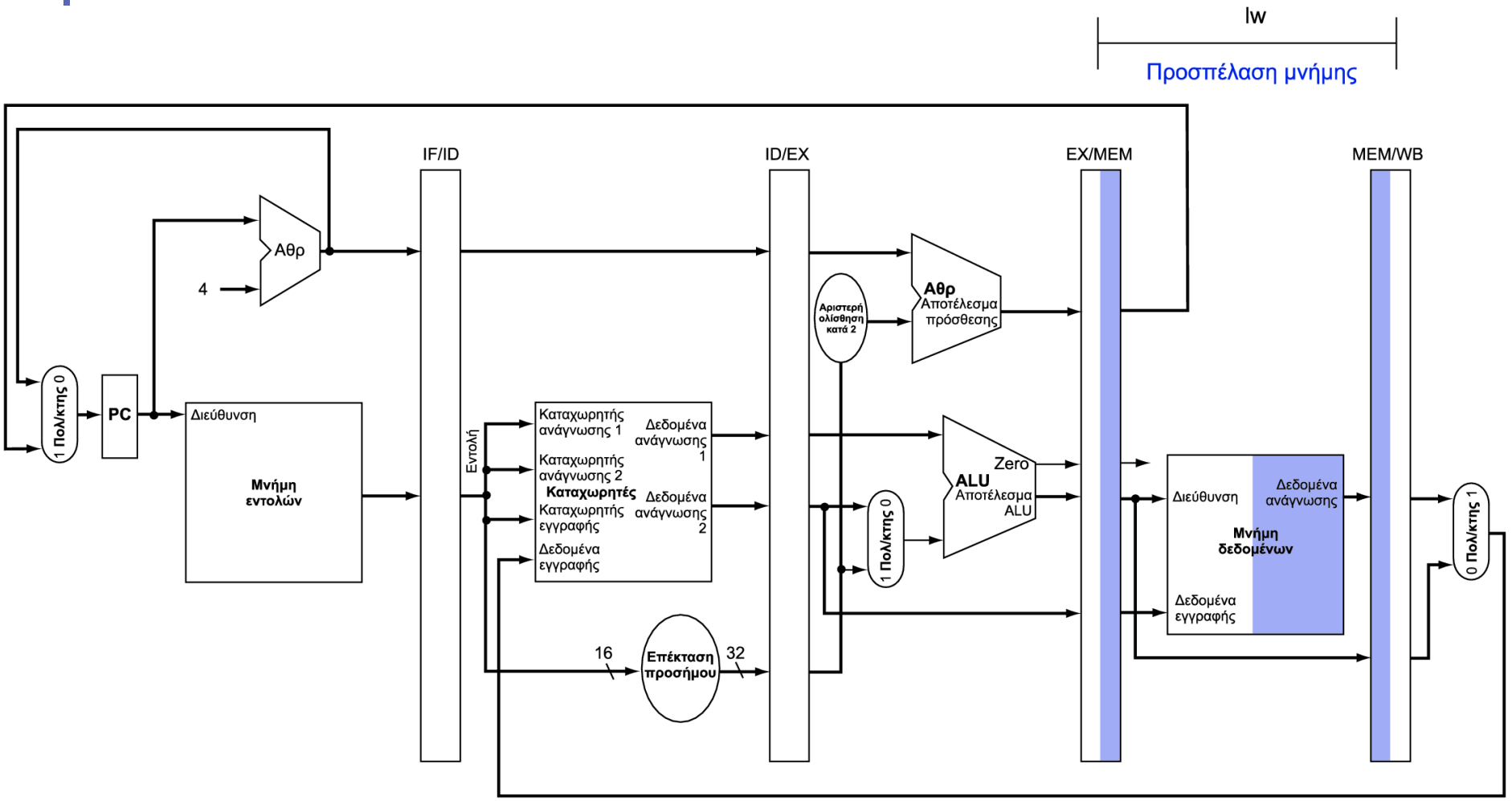
Στάδιο ID για Load, Store, …



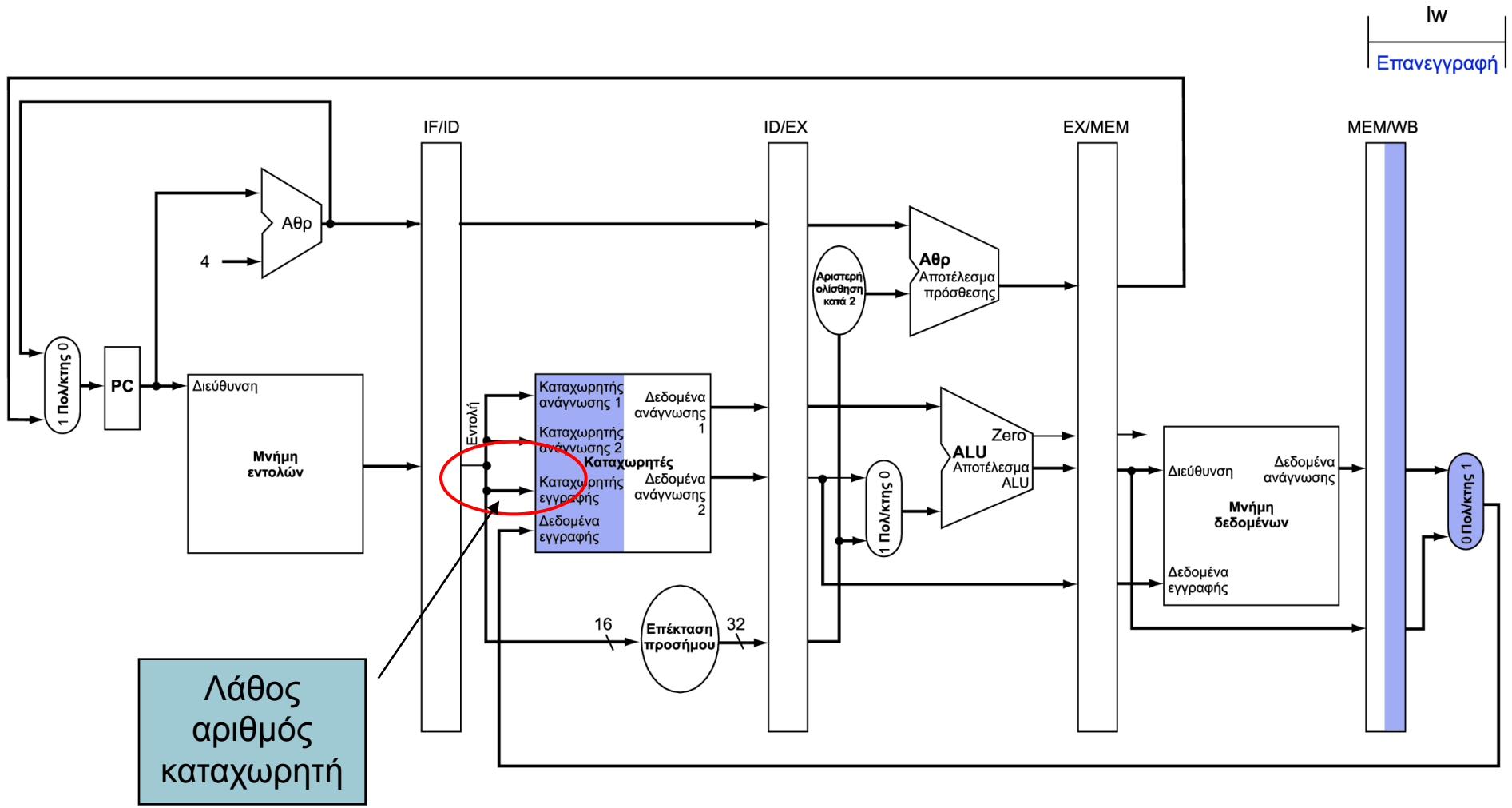
Στάδιο EX για Load



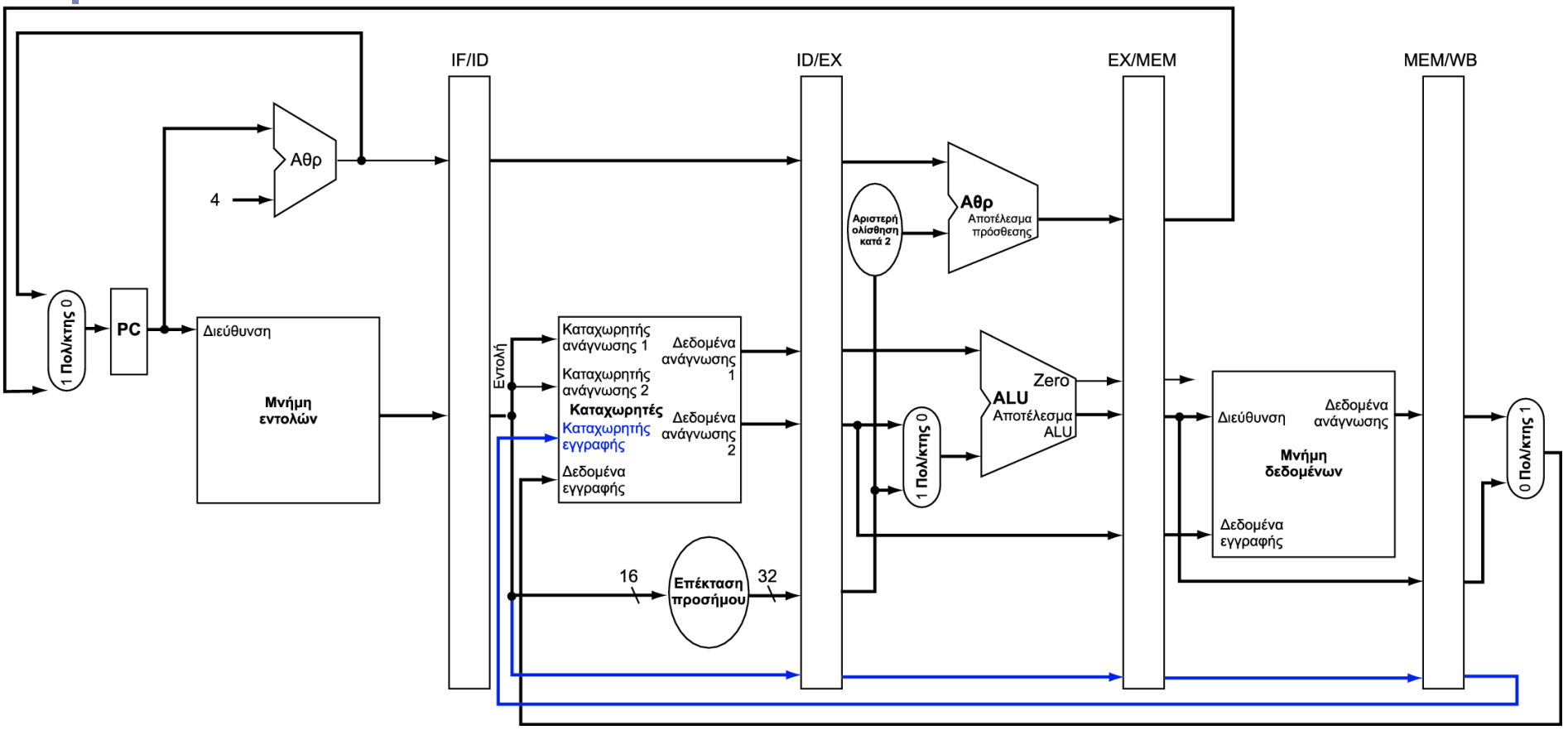
Στάδιο MEM για Load



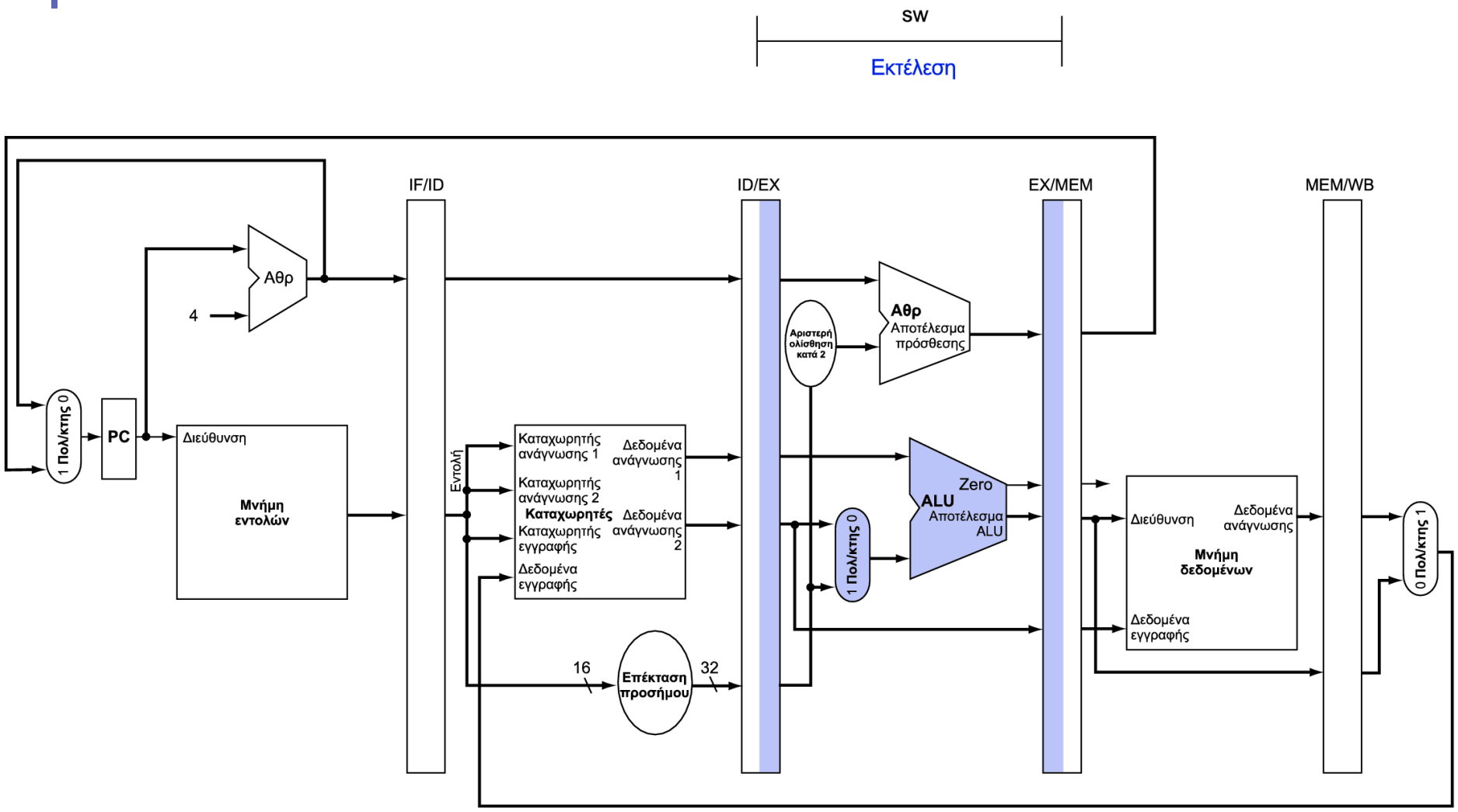
Στάδιο WB για Load



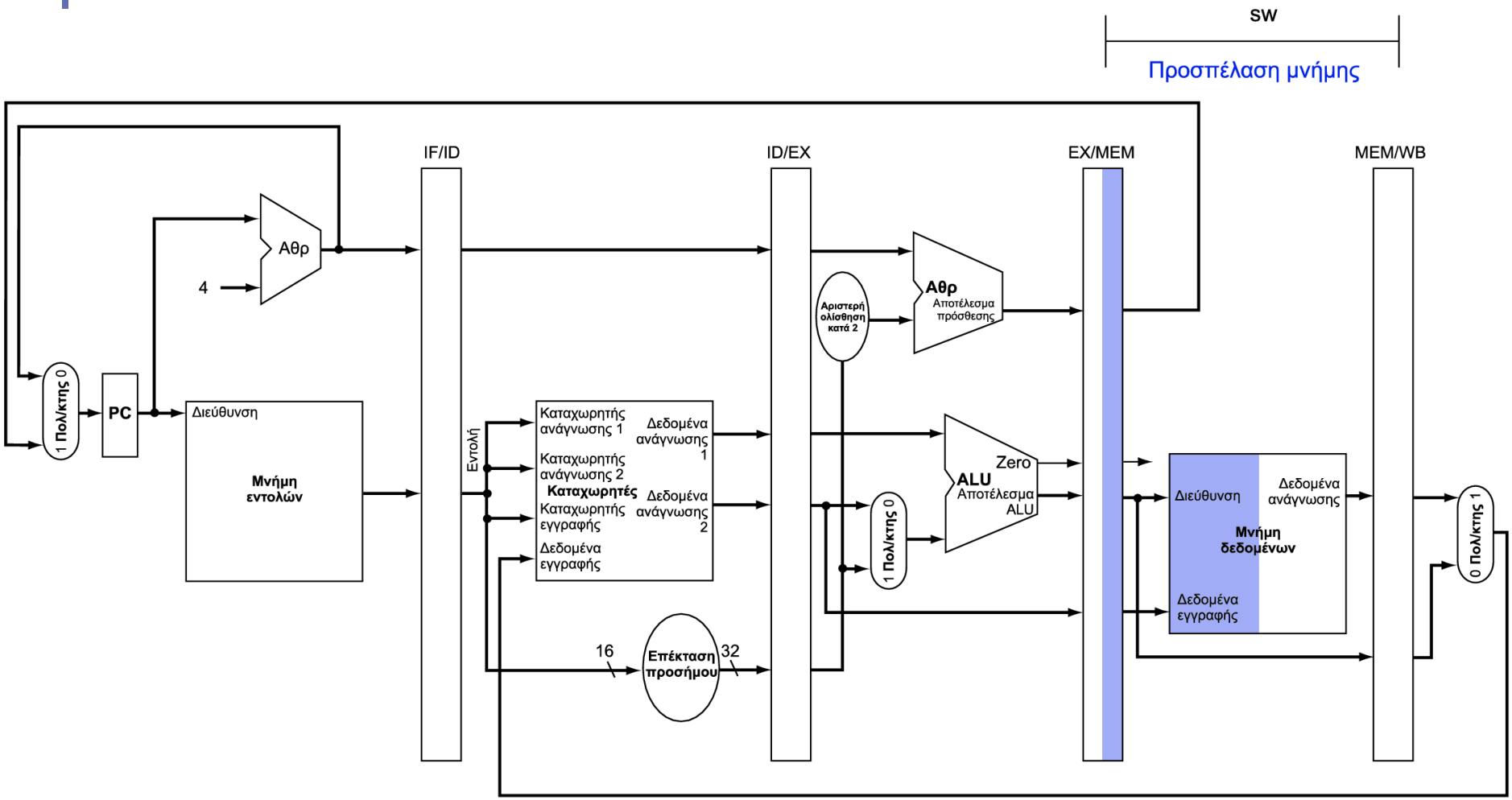
Διορθωμένη διαδρομή Load



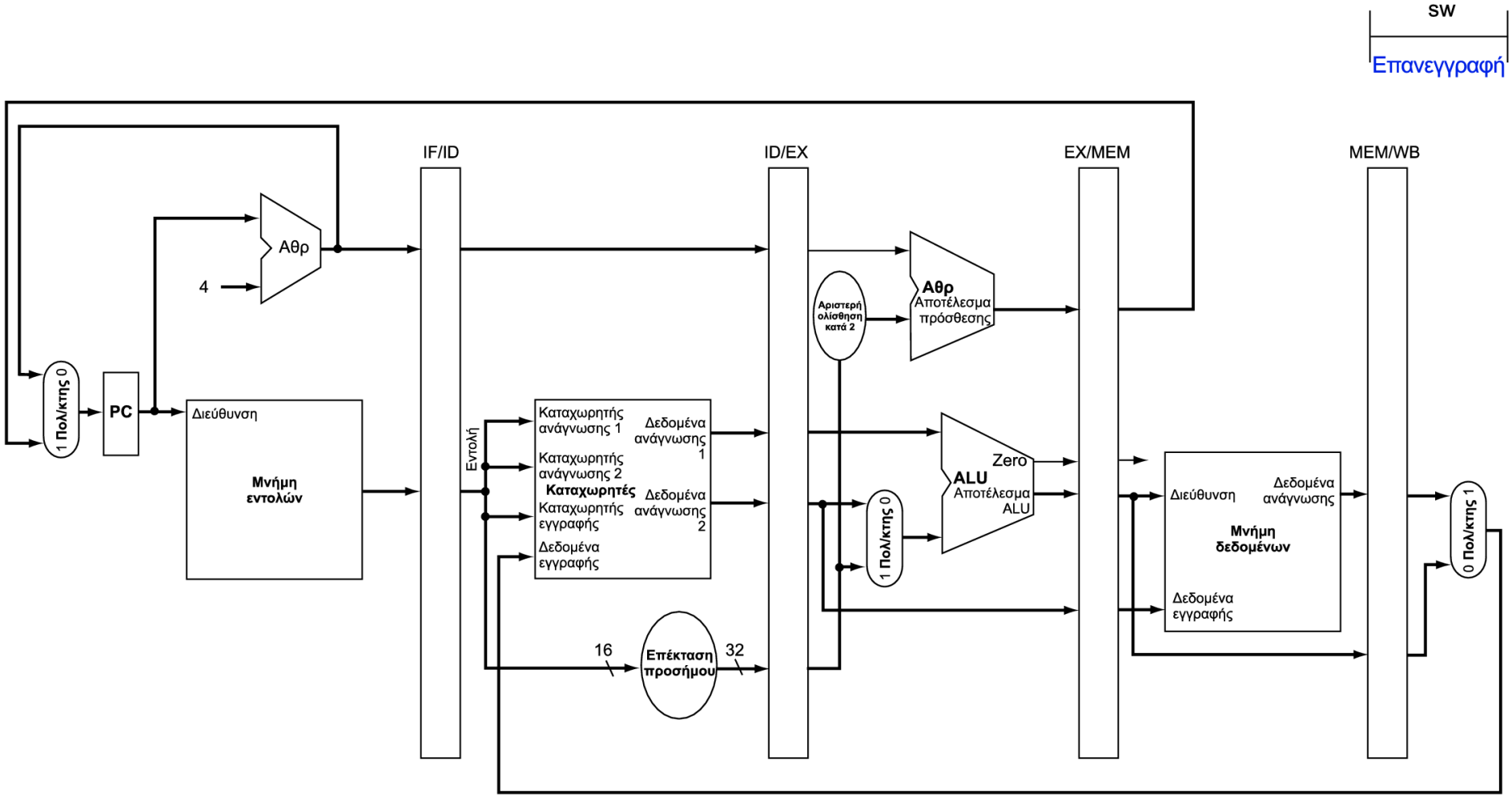
Στάδιο EX για Store



Στάδιο MEM για Store

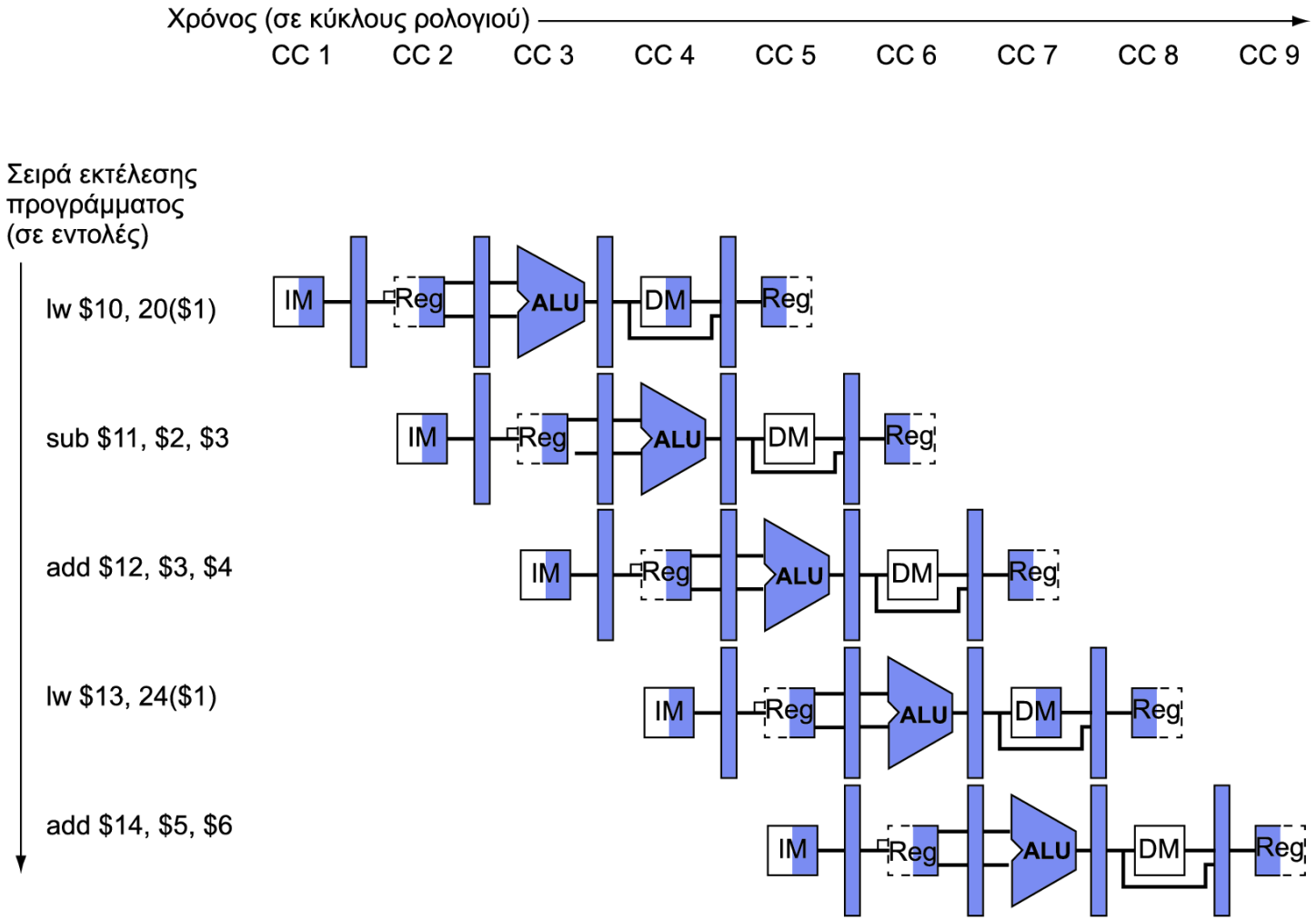


Στάδιο WB για Store



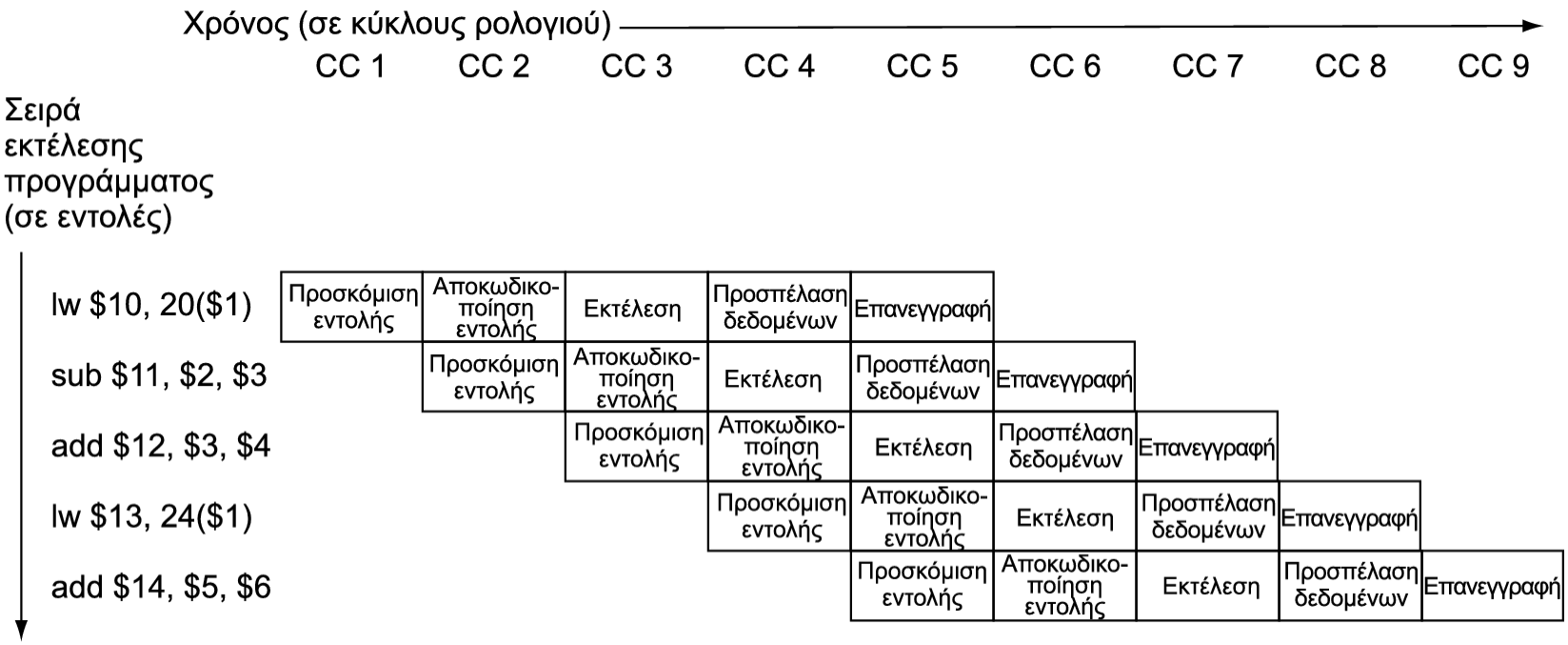
Διάγραμμα διοχέτευσης πολλών κύκλων

- Μορφή που δείχνει τη χρήση των πόρων



Διάγραμμα διοχέτευσης πολλών κύκλων

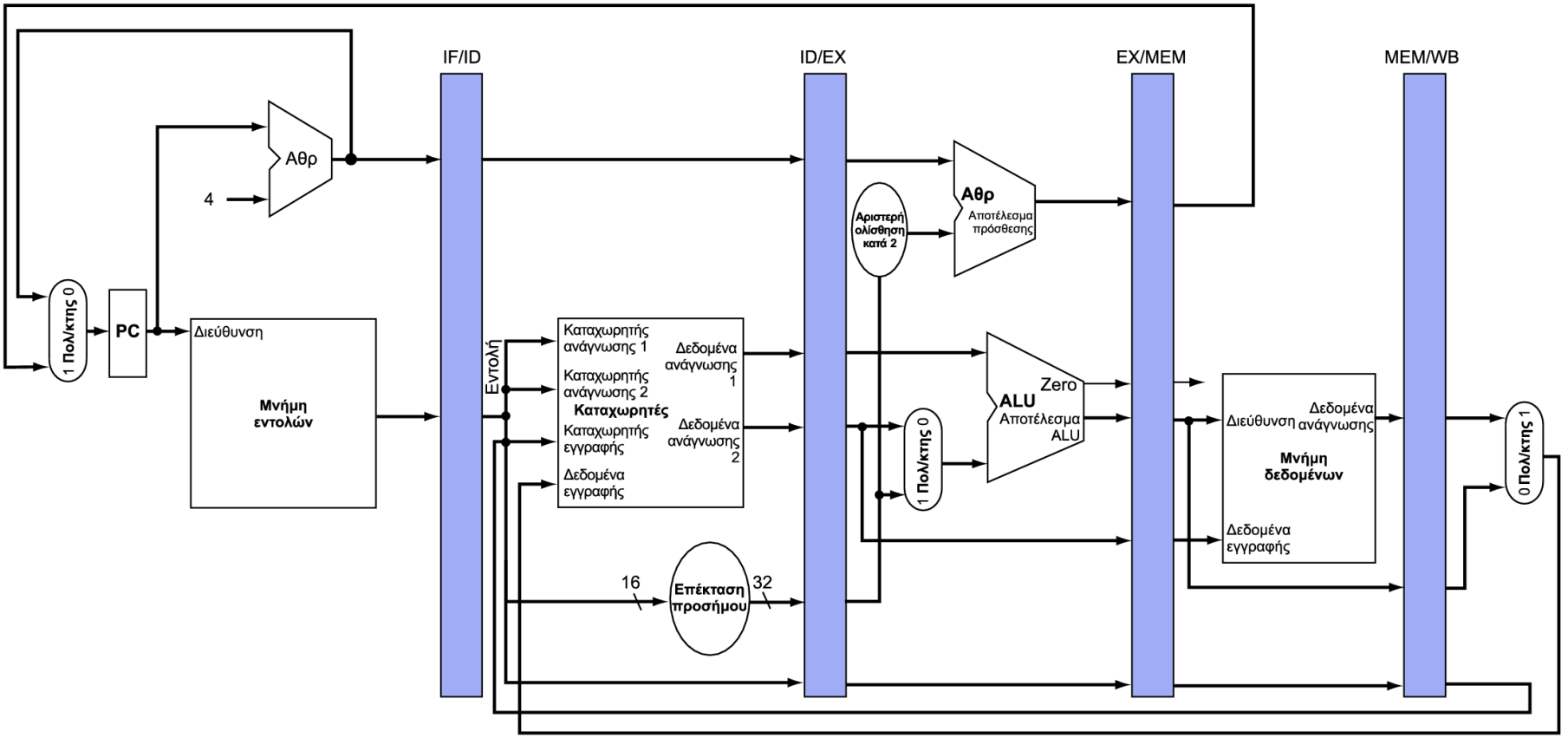
- Παραδοσιακή μορφή



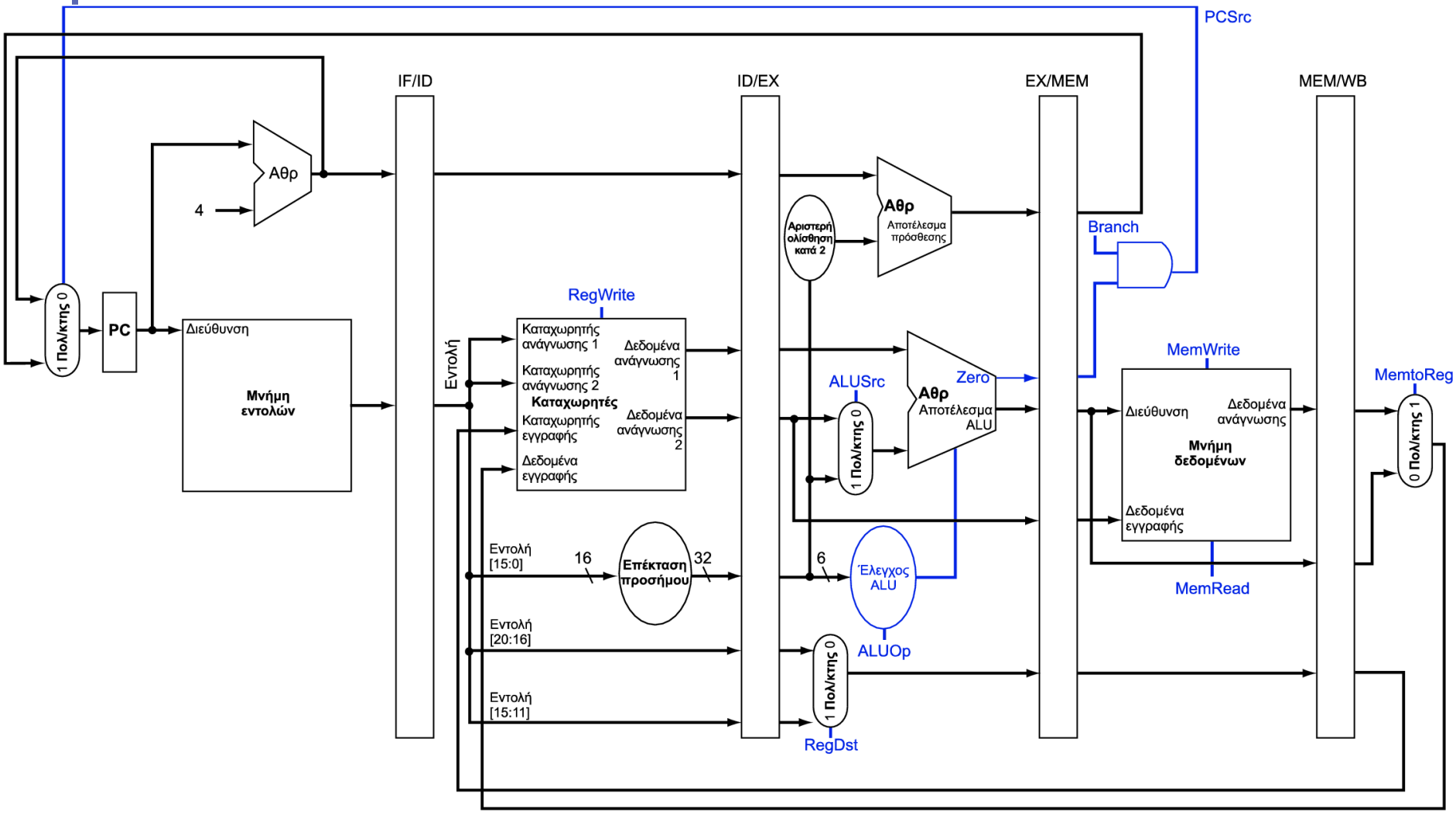
Διάγραμμα διοχέτευσης ενός κύκλου

- Κατάσταση της διοχέτευσης σε δεδομένο κύκλο

add $14, $5, $6	lw $13, 24 ($1)	add $12, $3, $4	sub $11, $2, $3	lw$10, 20($1)
Προσκόμιση εντολής	Αποκωδικοποίηση εντολής	Εκτέλεση	Προσπέλαση μνήμης	Επανεγγραφή

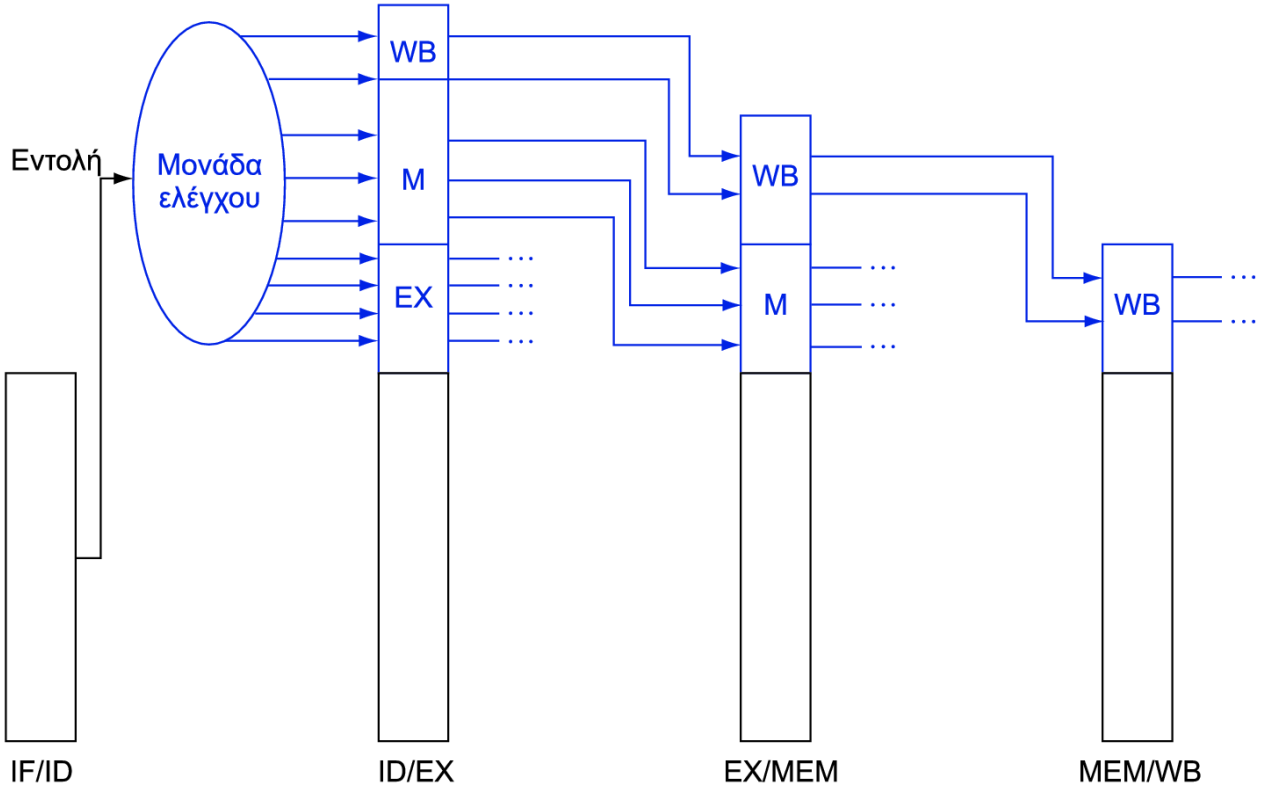


Έλεγχος διοχέτευσης (απλουσ.)

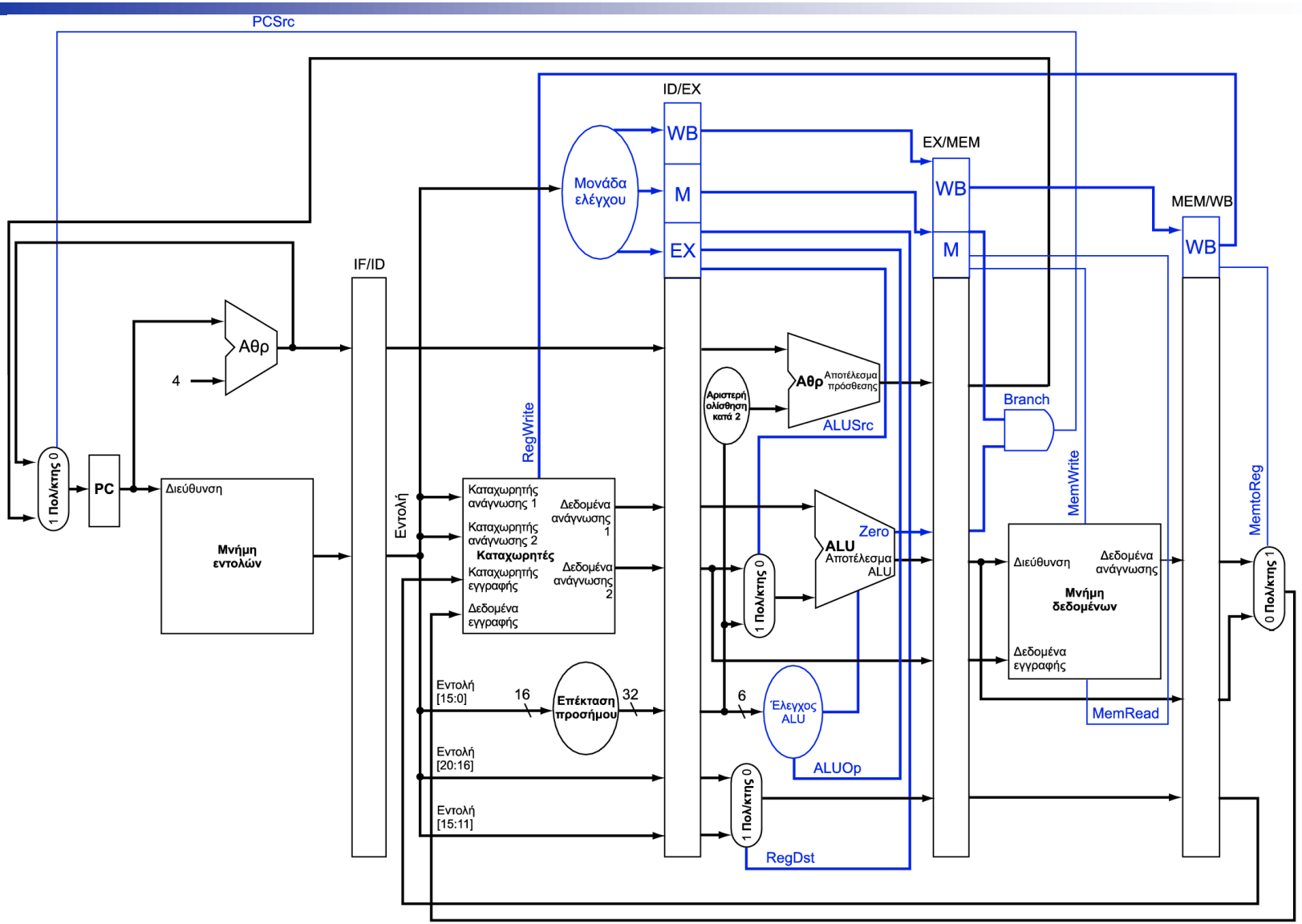


Έλεγχος διοχέτευσης

- Σήματα ελέγχου εξάγονται από την εντολή
  - Όπως και στην υλοποίηση ενός κύκλου



Έλεγχος διοχέτευσης



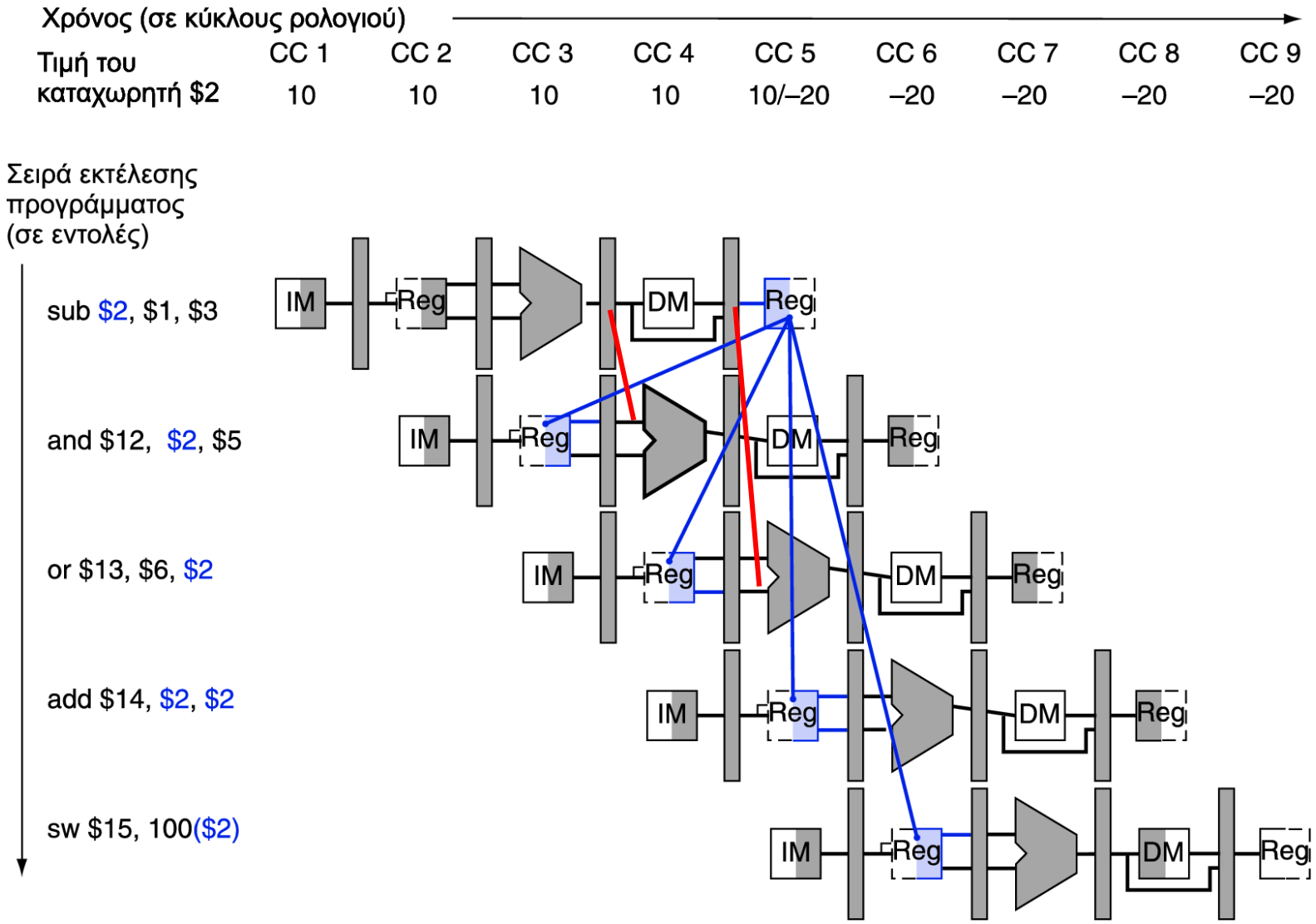
Κίνδυνοι δεδομένων σε εντολές ALU

- Θεωρήστε την ακολουθία:

```
sub $2, $1,$3
and $12,$2,$5
or  $13,$6,$2
add $14,$2,$2
sw  $15,100($2)
```

- Μπορούμε να επιλύσουμε τους κινδύνους με προώθηση (forwarding)
  - Πώς ανιχνεύουμε πότε πρέπει να γίνει προώθηση;

Εξαρτήσεις και προώθηση



Ανίχνευση ανάγκης για προώθηση

- Μεταβίβαση αριθμών καταχωρητών μέσα στη διοχέτευση
  - π.χ., ID/EX.RegisterRs = αριθμός καταχωρητή για το Rs που βρίσκεται στον καταχωρητή διοχέτευσης ID/EX
- Οι αριθμοί καταχωρητών των τελεστέων της ALU στο στάδιο EX δίνονται από τα
  - ID/EX.RegisterRs, ID/EX.RegisterRt
- Κίνδυνοι δεδομένων όταν
  1α. EX/MEM.RegisterRd = ID/EX.RegisterRs
  1β. EX/MEM.RegisterRd = ID/EX.RegisterRt
  2α. MEM/WB.RegisterRd = ID/EX.RegisterRs
  2β. MEM/WB.RegisterRd = ID/EX.RegisterRt

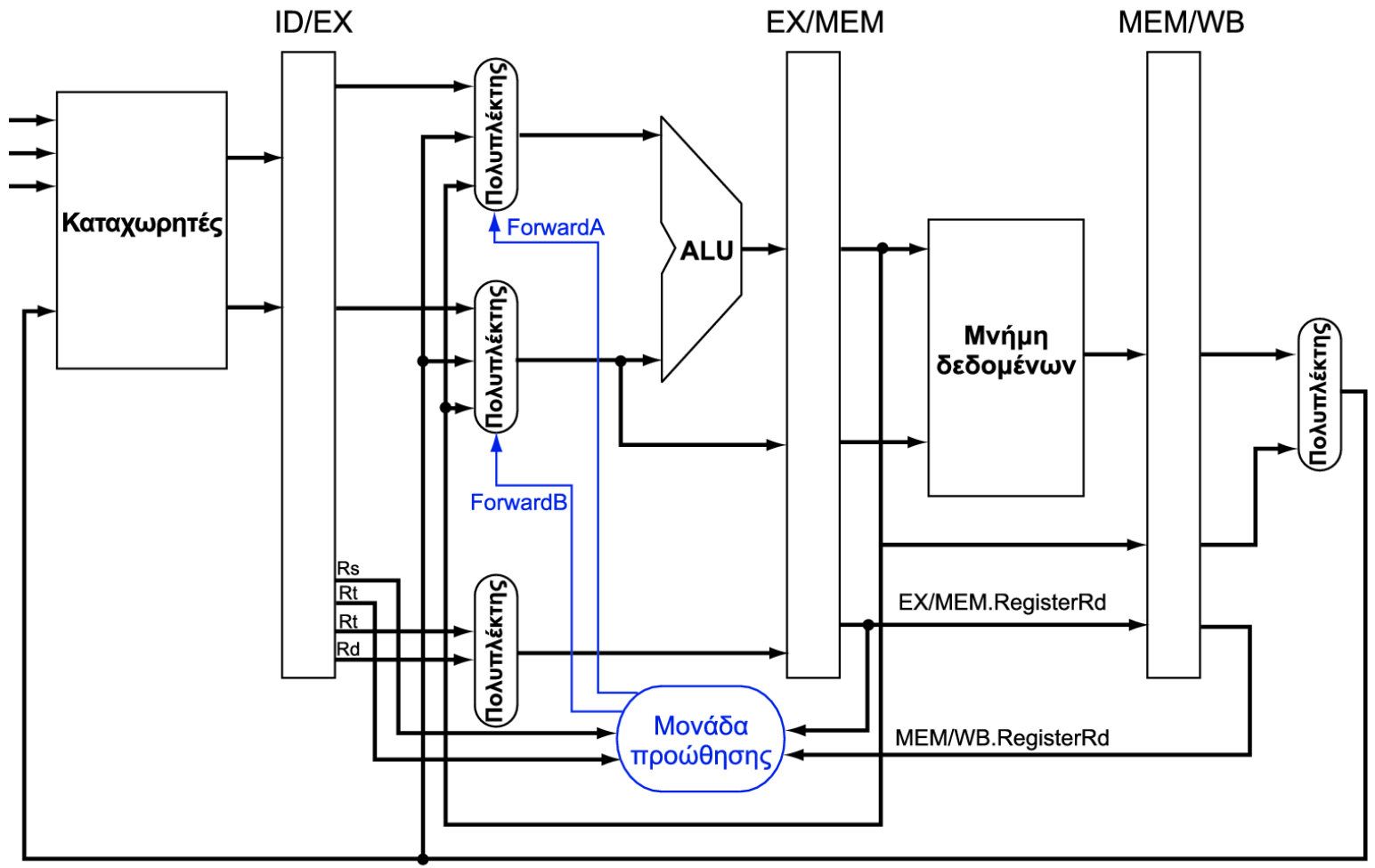
Προωθ. από EX/MEM καταχ. διοχ.

Προώθ. από MEM/WB καταχ. διοχ.

Ανίχνευση ανάγκης για προώθηση

- Αλλά μόνο αν η εντολή που προωθεί πρόκειται να γράψει σε κάποιο καταχωρητή!
  - EX/MEM.RegWrite, MEM/WB.RegWrite
- Και μόνο αν το Rd της εντολής δεν είναι $zero
  - EX/MEM.RegisterRd ≠ 0, MEM/WB.RegisterRd ≠ 0

Διαδρομές προώθησης



β. Με προώθηση

Συνθήκες προώθησης

- Κίνδυνος στο στάδιο EX
  - if (EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
    and (EX/MEM.RegisterRd = ID/EX.RegisterRs))
    ForwardA = 10
  - if (EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
    and (EX/MEM.RegisterRd = ID/EX.RegisterRt))
    ForwardB = 10
- Κίνδυνος στο στάδιο MEM
  - if (MEM/WB.RegWrite and (MEM/WB.RegisterRd ≠ 0)
    and (MEM/WB.RegisterRd = ID/EX.RegisterRs))
    ForwardA = 01
  - if (MEM/WB.RegWrite and (MEM/WB.RegisterRd ≠ 0)
    and (MEM/WB.RegisterRd = ID/EX.RegisterRt))
    ForwardB = 01

Διπλός κίνδυνος δεδομένων

- Θεωρήστε την ακολουθία:

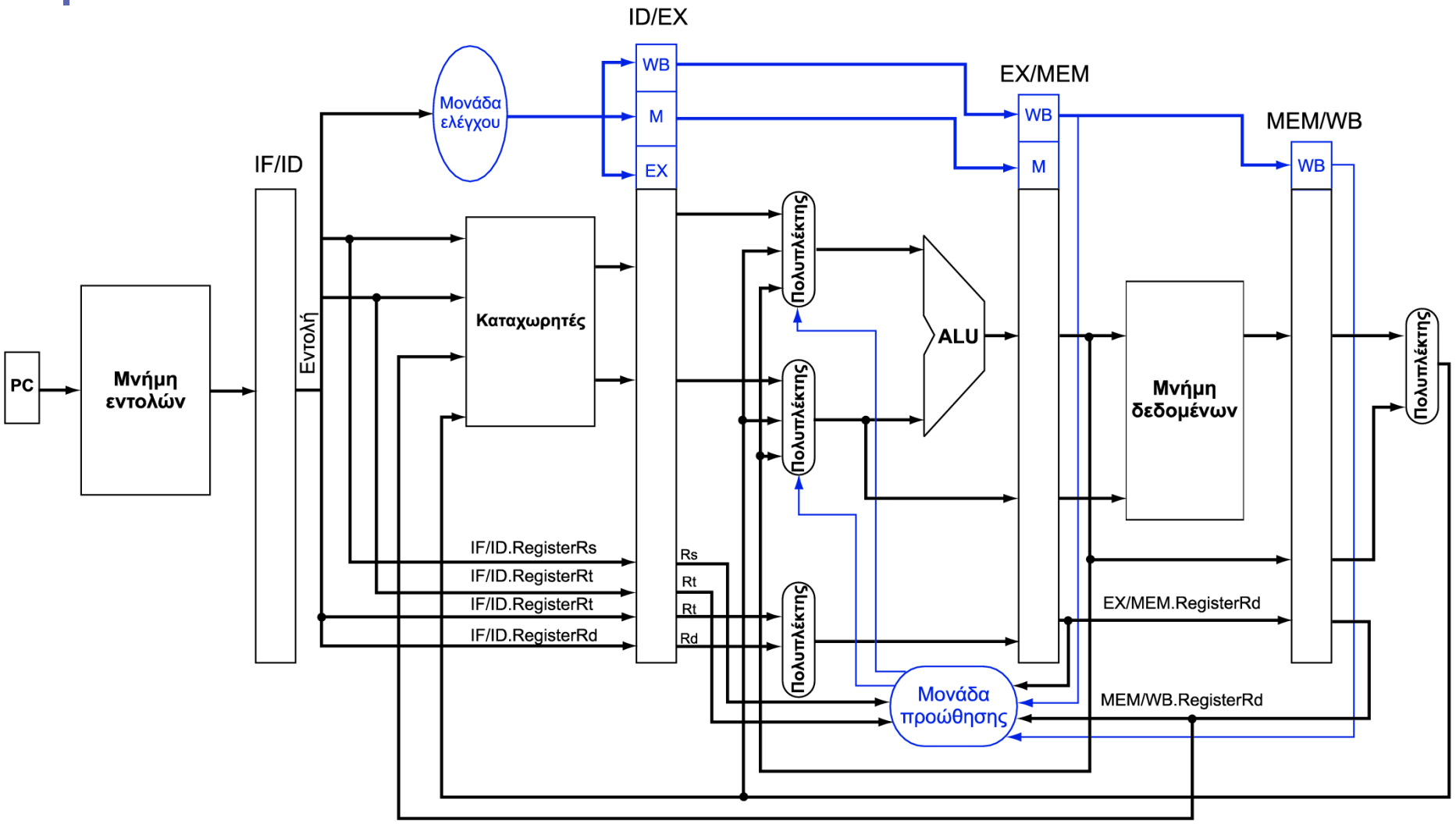
```
add $1,$1,$2
add $1,$1,$3
add $1,$1,$4
```

- Συμβαίνουν και οι δύο κίνδυνοι
  - Θέλουμε να χρησιμοποιήσουμε τον πιο πρόσφατο
- Αναθεώρηση της συνθήκης προώθησης του MEM
  - Προώθηση μόνο αν η συνθήκη κινδύνου του σταδίου EX δεν είναι αληθής

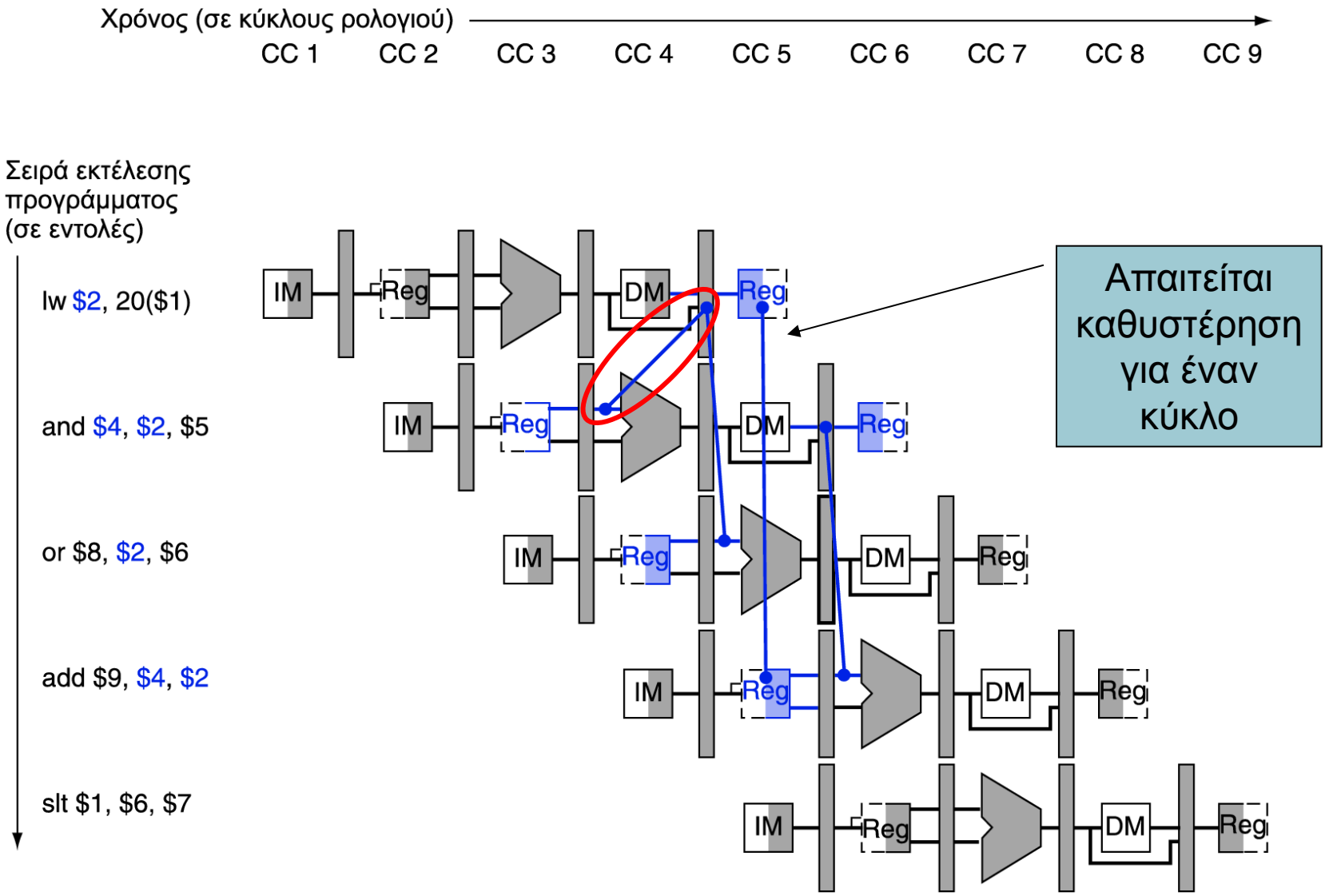
Αναθεωρημένη συνθήκη προώθησης

- Κίνδυνος στο στάδιο MEM
  - if (MEM/WB.RegWrite and (MEM/WB.RegisterRd ≠ 0)
    and not (EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
    and (EX/MEM.RegisterRd = ID/EX.RegisterRs))
    and (MEM/WB.RegisterRd = ID/EX.RegisterRs))
    ForwardA = 01
  - if (MEM/WB.RegWrite and (MEM/WB.RegisterRd ≠ 0)
    and not (EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
    and (EX/MEM.RegisterRd = ID/EX.RegisterRt))
    and (MEM/WB.RegisterRd = ID/EX.RegisterRt))
    ForwardB = 01

Διαδρομή δεδομένων με προώθηση



Κίνδυνος δεδομένων φόρτωσης-χρήσης



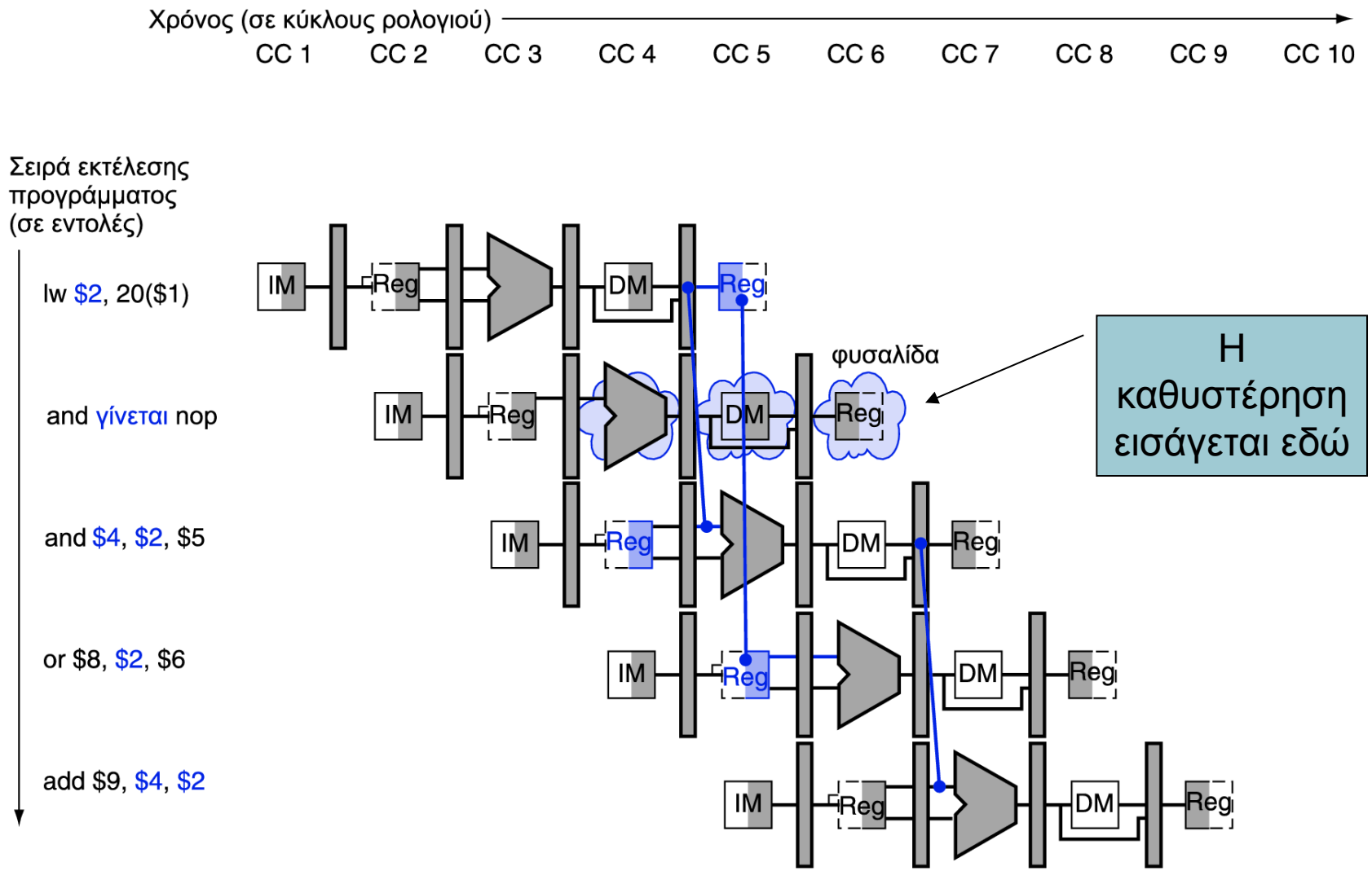
Ανίχνευση κινδύνου φόρτωσης-χρήσης

- Έλεγχος όταν η εντολή που κάνει τη χρήση αποκωδικοποιείται στο στάδιο ID
- Οι αριθμοί των καταχωρητών τελεστέων της ALU στο στάδιο ID δίνονται από τα
  - IF/ID.RegisterRs, IF/ID.RegisterRt
- Κίνδυνος φόρτωσης-χρήσης όταν
  - ID/EX.MemRead and
    ((ID/EX.RegisterRt = IF/ID.RegisterRs) or
    (ID/EX.RegisterRt = IF/ID.RegisterRt))
- Αν ανιχνευθεί, γίνεται καθυστέρηση της διοχέτευσης και εισάγεται φυσαλίδα

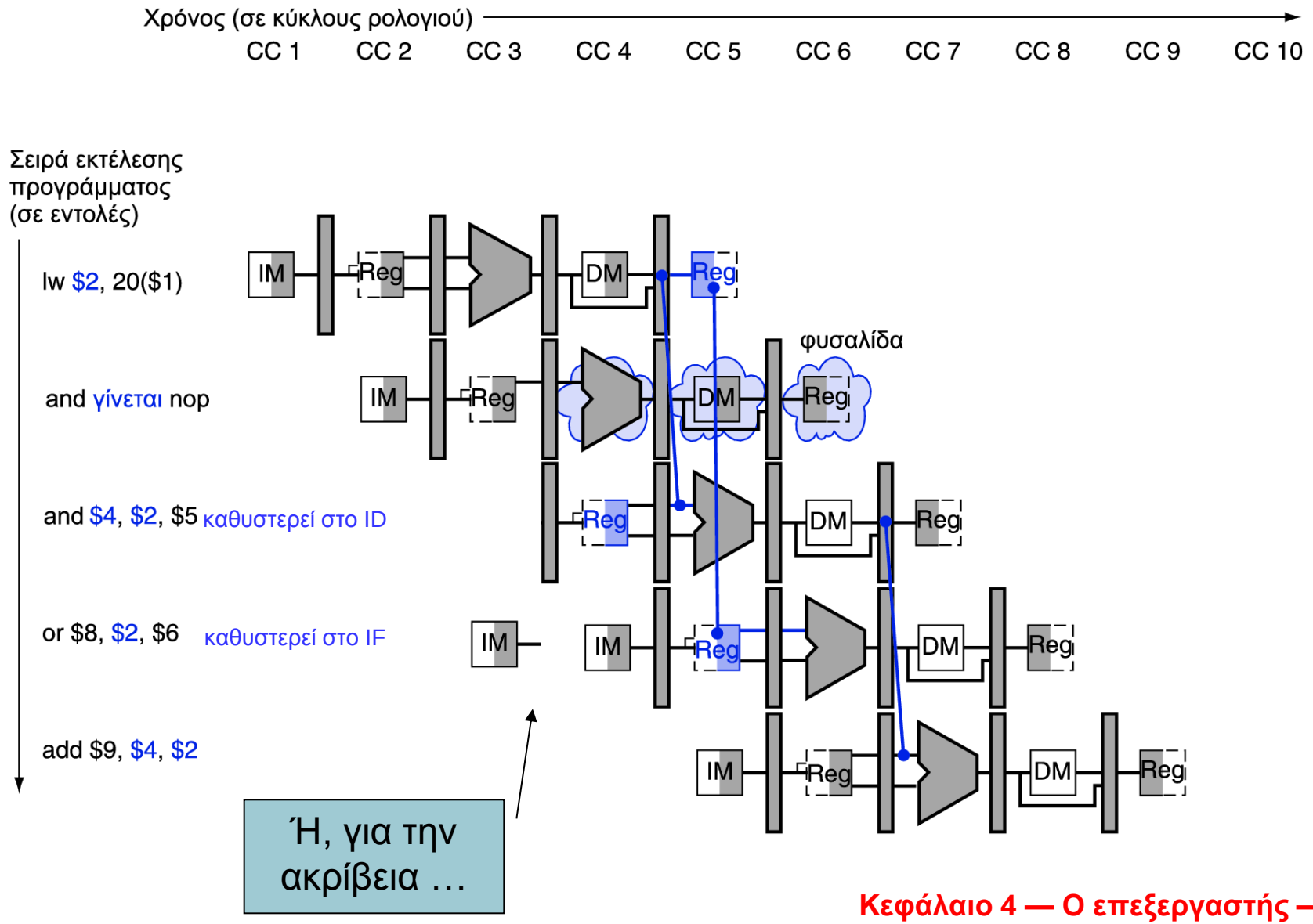
Πώς καθυστερεί η διοχέτευση

- Οι τιμές ελέγχου στον καταχωρητή ID/EX γίνονται 0
  - Τα EX, MEM και WB εκτελούν nop (no-operation, απραξία)
- Αποτρέπεται η ενημέρωση του PC και του καταχωρητή IF/ID
  - Η εντολή που κάνει τη χρήση αποκωδικοποιείται και πάλι
  - Η επόμενη εντολή προσκομίζεται και πάλι
  - Η καθυστέρηση 1 κύκλου επιτρέπει στο MEM να διαβάσει τα δεδομένα για την εντολή lw
    - Μπορεί στη συνέχεια να κάνει προώθηση στο στάδιο EX

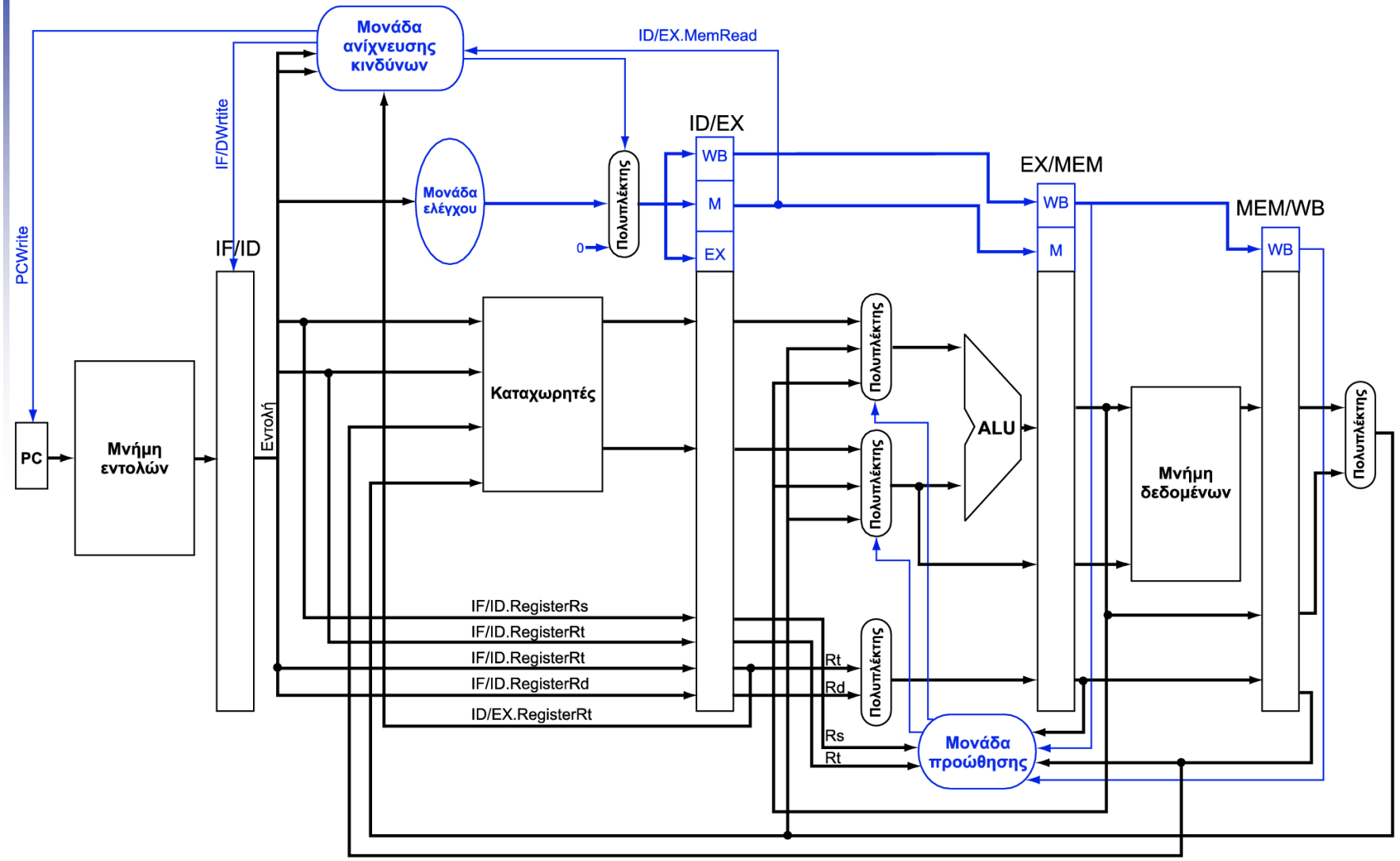
Καθυστέρηση/φυσαλίδα στη διοχέτευση



Καθυστέρηση/φυσαλίδα στη διοχέτευση



Διαδρομή δεδομένων με ανίχνευση κινδύνων



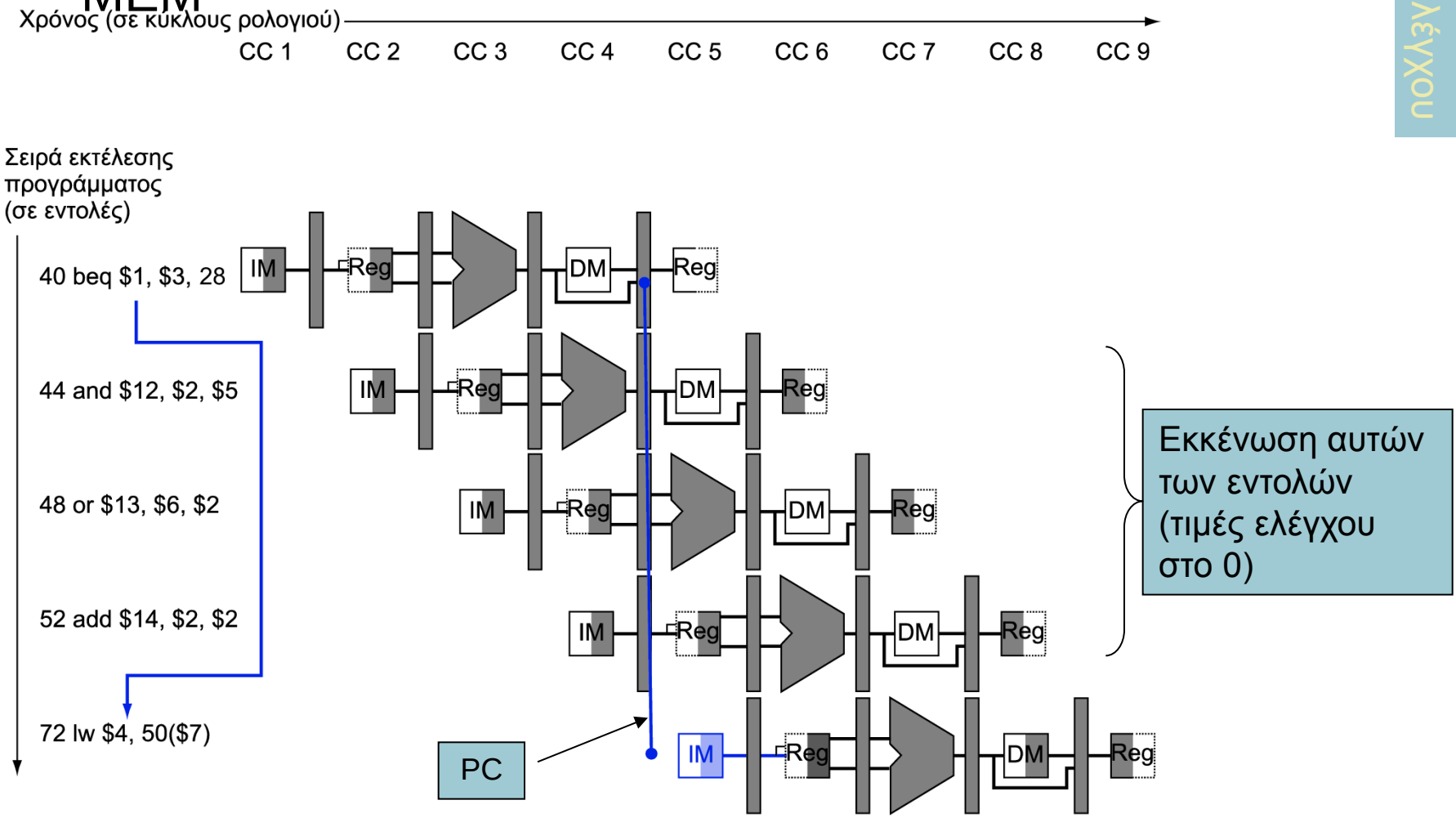
Καθυστερήσεις και απόδοση

ΓΕΝΙΚΗ εικόνα

- Οι καθυστερήσεις μειώνουν την απόδοση
  - Αλλά είναι απαραίτητες για να πάρουμε σωστά αποτελέσματα
- Ο μεταγλωττιστής μπορεί να αναδιατάξει τον κώδικα για να αποφευχθούν οι κίνδυνοι και οι καθυστερήσεις
  - Απαιτεί γνώση της δομής της διοχέτευσης

Κίνδυνοι διακλάδωσης

- Αν το αποτέλεσμα της διακλάδωσης καθορίζεται στο MEM

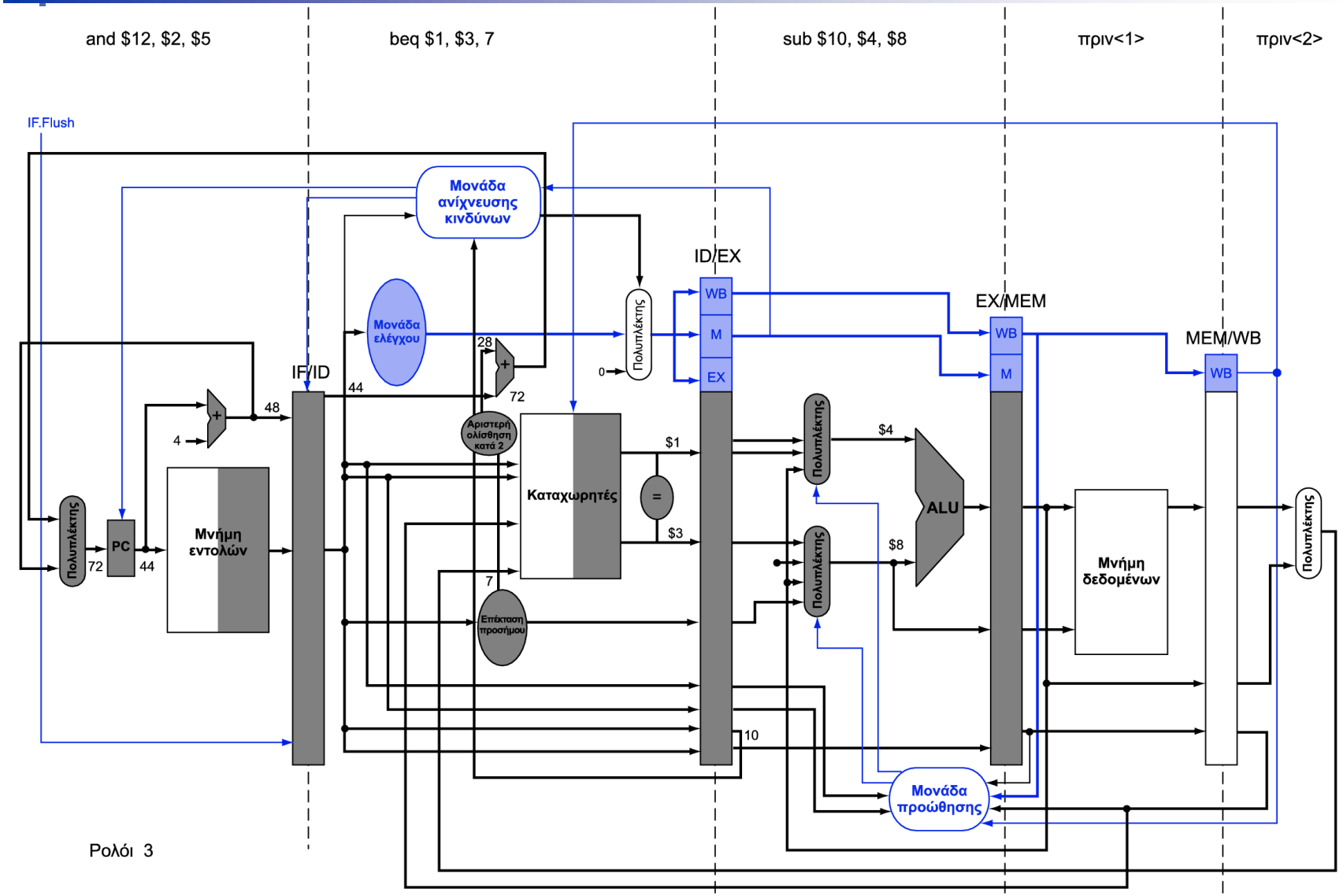


Μείωση καθυστέρησης διακλάδωσης

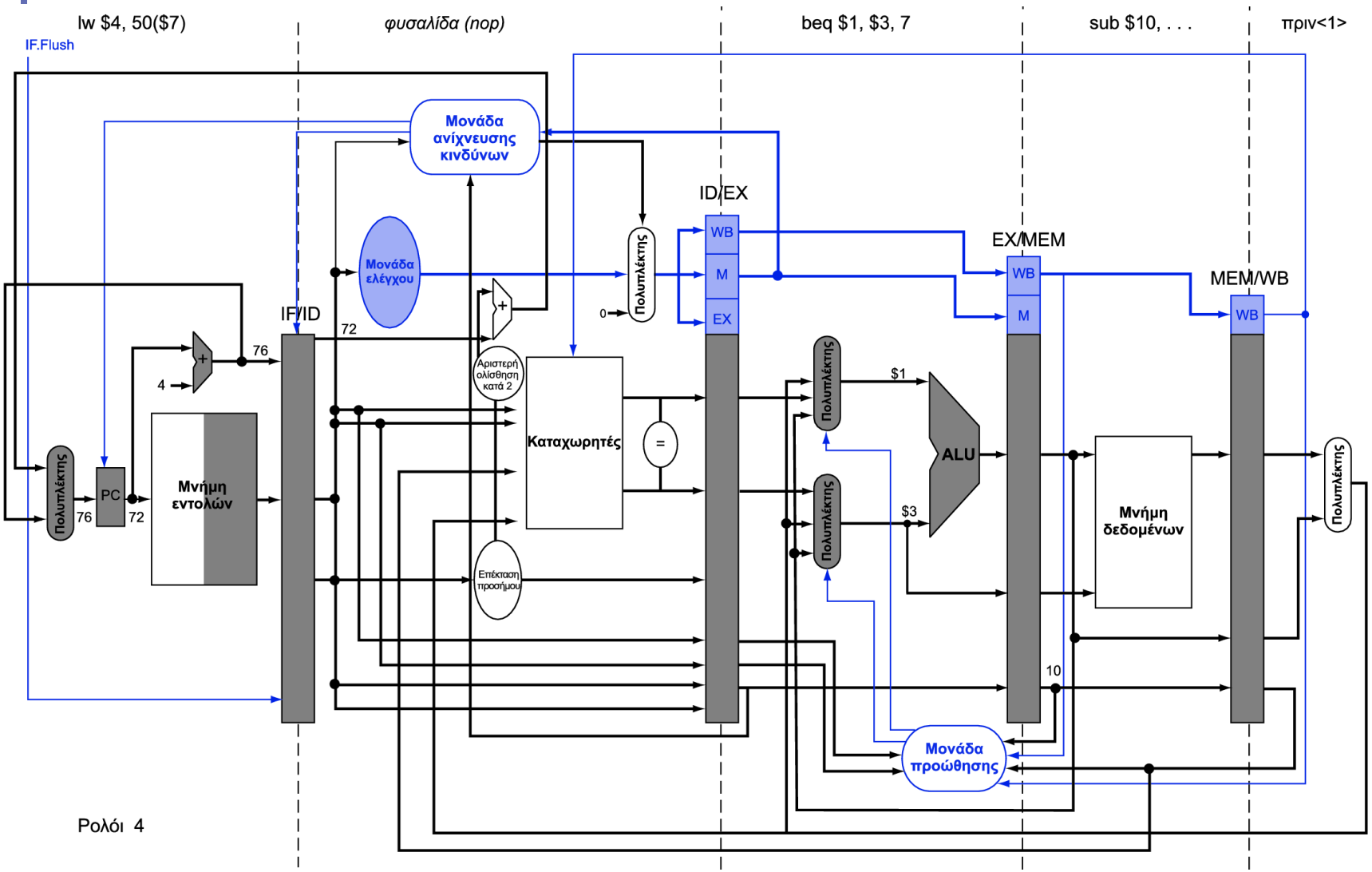
- Μεταφορά υλικού για προσδιορισμό αποτελέσματος στο στάδιο ID
  - Αθροιστής διεύθυνσης προορισμού
  - Συγκριτής καταχωρητών
- Παράδειγμα: λαμβανόμενη διακλάδωση

```
36:  sub  $10, $4, $8
40:  beq  $1,  $3, 7
44:  and  $12, $2, $5
48:  or   $13, $2, $6
52:  add  $14, $4, $2
56:  slt  $15, $6, $7
     ...
72:  lw   $4, 50($7)
```

Παράδειγμα: λαμβανόμενη διακλάδωση



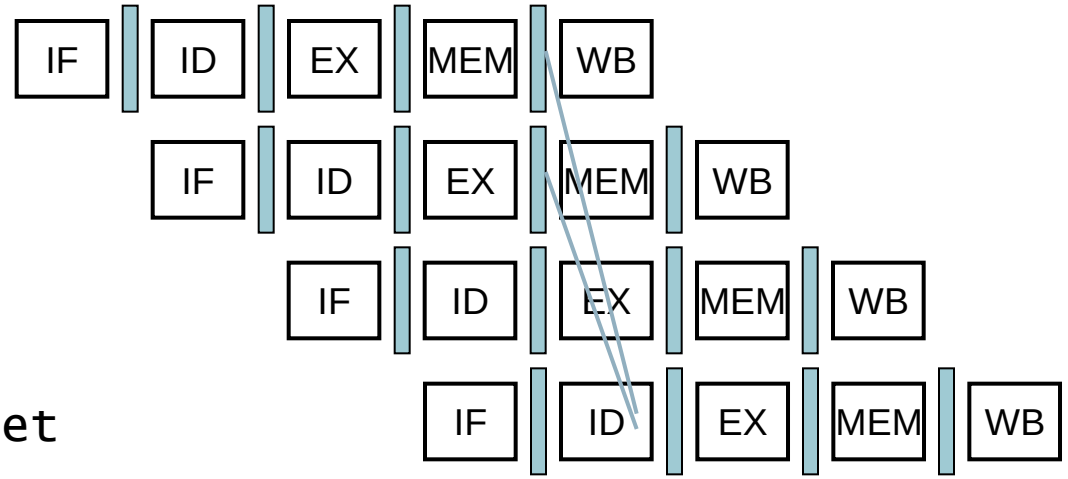
Παράδειγμα: λαμβανόμενη διακλάδωση



Κίνδυνοι δεδομένων για διακλαδώσεις

- Αν ένας καταχωρητής σύγκρισης είναι προορισμός εντολής ALU που προηγείται κατά 2 ή 3 θέσεις

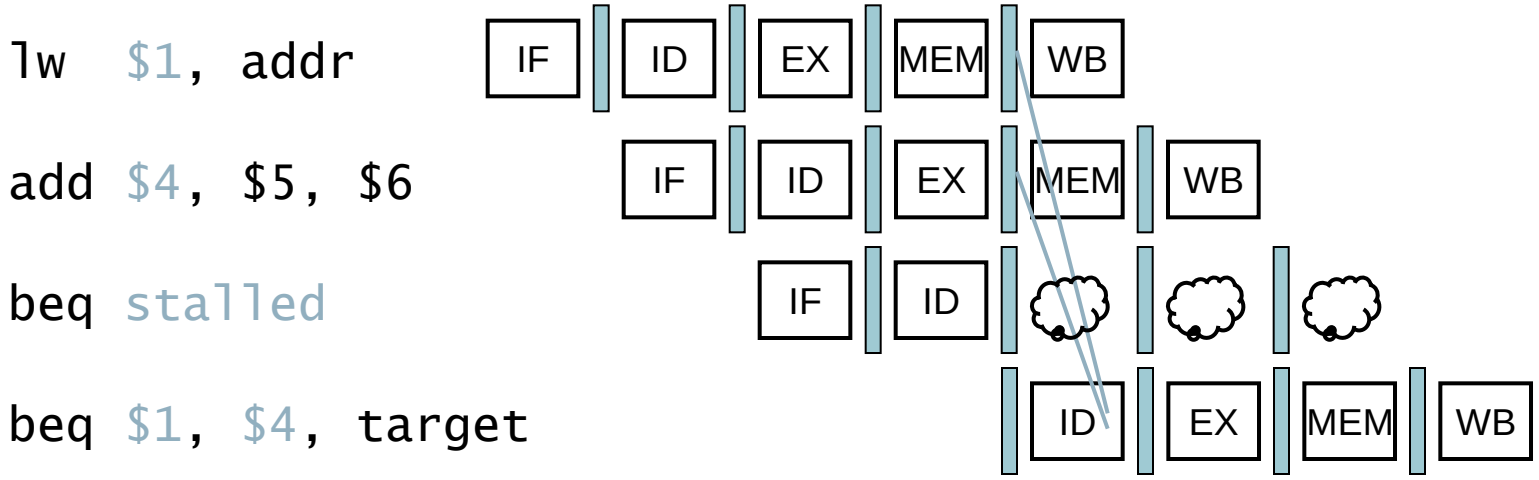
```
add $1, $2, $3
add $4, $5, $6
…
beq $1, $4, target
```



- Μπορεί να λυθεί με προώθηση

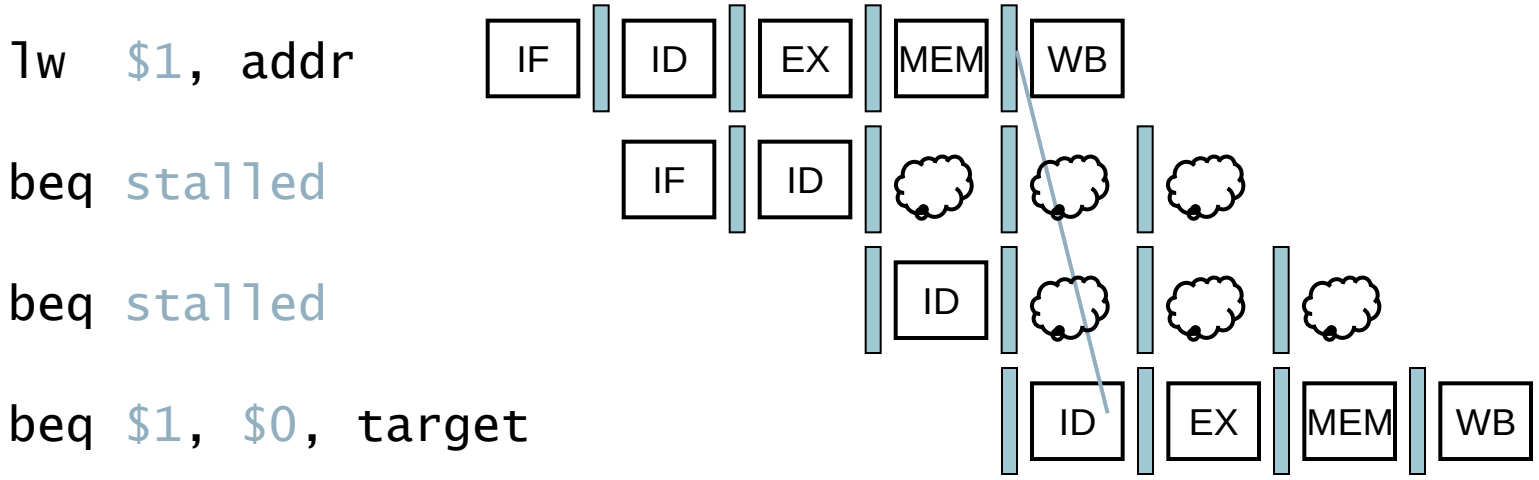
Κίνδυνοι δεδομένων για διακλαδώσεις

- Αν ένας καταχωρητής σύγκρισης είναι προορισμός εντολής ALU που προηγείται αμέσως ή εντολής φόρτωσης που προηγείται κατά 2 θέσεις
  - Χρειάζεται 1 κύκλος καθυστέρησης



Κίνδυνοι δεδομένων για διακλαδώσεις

- Αν ένας καταχωρητής σύγκρισης είναι προορισμός μιας εντολής φόρτωσης που προηγείται αμέσως
  - Χρειάζονται 2 κύκλοι καθυστέρησης

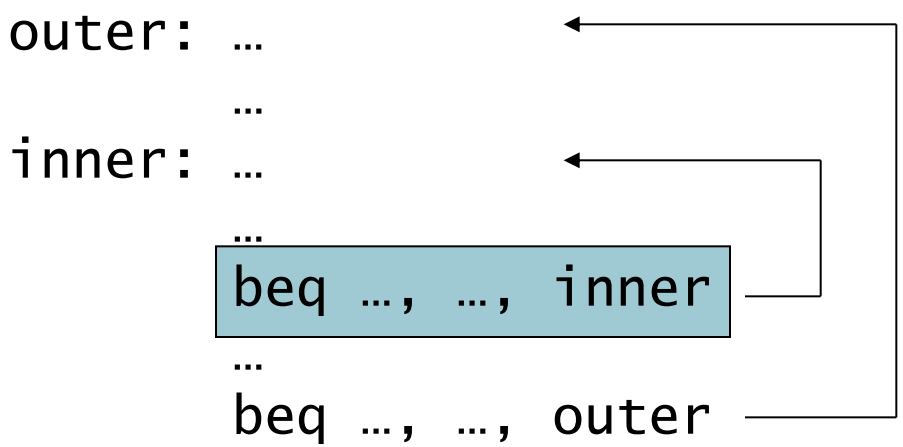


Δυναμική πρόβλεψη διακλάδωσης

- Σε πιο βαθιές και υπερβαθμωτές διοχετεύσεις, η ποινή της διακλάδωσης είναι πιο σημαντική
- Χρήση δυναμικής πρόβλεψης
  - Προσωρινή μνήμη πρόβλεψης διακλάδωσης (branch prediction buffer), που λέγεται και πίνακας ιστορικού διακλάδωσης (branch history table)
  - Δεικτοδοτείται από τις διευθύνσεις της πρόσφατης εντολής διακλάδωσης
  - Αποθηκεύει το αποτέλεσμα (λήψη/μη λήψη)
  - Για εκτέλεση μιας διακλάδωσης
    - Έλεγχος του πίνακα, υπόθεση του ίδιου αποτελέσματος
    - Εκκίνηση προσκόμισης από την επόμενη ή τον προορισμό
    - Αν λάθος, εκκένωση διοχέτευσης και αντιστροφή πρόβλεψης

Διάταξη πρόβλεψης 1 bit: μειονέκτημα

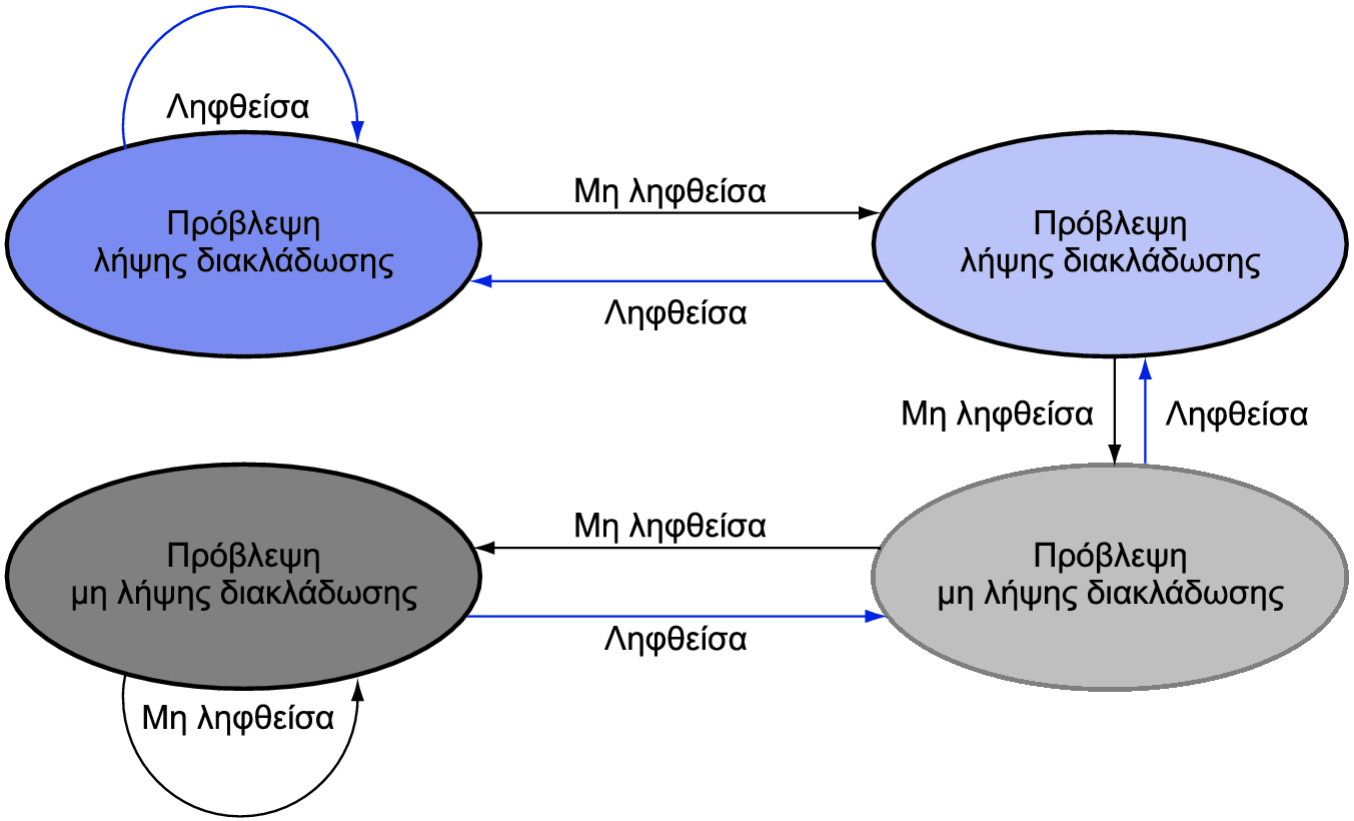
- Οι διακλαδώσεις του εσωτερικού βρόχου προβλέπονται λανθασμένα δύο φορές



  - Λανθασμένη πρόβλεψη λήψης στην τελευταία επανάληψη του εσωτερικού βρόχου
  - Μετά λανθασμένη πρόβλεψη μη λήψης στην πρώτη επανάληψη του εσωτερικού βρόχου την επόμενη φορά που θα εκτελεστεί

Διάταξη πρόβλεψης των 2 bit

- Αλλάζει η πρόβλεψη μόνο μετά από δύο διαδοχικές λανθασμένες προβλέψεις



Υπολογισμός προορισμού διακλάδωσης

- Ακόμη και με πρόβλεψη, πρέπει ακόμη να υπολογιστεί η διεύθυνση διακλάδωσης
  - Ποινή 1 κύκλου για λαμβανόμενη διακλάδωση
- Προσωρινή μνήμη προορισμού διακλάδωσης (branch target buffer)
  - Κρυφή μνήμη για διευθύνσεις προορισμού
  - Δεικτοδοτείται από τον PC όταν προσκομίζεται η εντολή
    - Αν υπάρχει ευστοχία (hit) και η εντολή είναι διακλάδωση με πρόβλεψη λήψης, μπορεί να γίνει άμεση προσκόμιση του προορισμού

Εξαιρέσεις και διακοπές

- «Μη αναμενόμενα» συμβάντα που απαιτούν αλλαγή της ροής του ελέγχου
  - Διαφορετικές αρχιτεκτονικές συνόλου εντολών χρησιμοποιούν τους όρους διαφορετικά
- Εξαίρεση (exception)
  - Παρουσιάζεται μέσα στη CPU
    - π.χ., μη ορισμένος κωδικός λειτουργίας (undefined opcode), υπερχείλιση (overflow), κλήση συστήματος (syscall), …
- Διακοπή (interrupt)
  - Από έναν εξωτερικό ελεγκτή εισόδου/εξόδου
- Δύσκολος ο χειρισμός τους χωρίς να θυσιαστεί απόδοση

Χειρισμός εξαιρέσεων

- Στο MIPS, τις εξαιρέσεις διαχειρίζεται ένας Συνεπεξεργαστής Ελέγχου Συστήματος (System Control Coprocessor), ο CP0
- Αποθήκευση του PC της εντολής που διακόπτεται
  - Στο MIPS: Exception Program Counter (EPC)
- Αποθήκευση της ένδειξης του προβλήματος
  - Στον MIPS: καταχωρητής Cause (Αιτίου)
  - Υποθέτουμε 1 bit μόνο
    - 0 για μη ορισμένο κωδικό λειτουργίας (undefined opcode), και 1 για υπερχείλιση
- Άλμα στο χειριστή στη δ/νση 8000 00180

Εναλλακτικός μηχανισμός

- Διανυσματικές διακοπές (vectored interrupts)
  - Η δ/νση του χειριστή καθορίζεται από την αιτία
- Παράδειγμα:
  - Μη ορισμένος opcode: C000 0000
  - Υπερχείλιση: C000 0020
  - …: C000 0040
- Οι εντολές είτε
  - Ασχολούνται με τη διακοπή, είτε
  - Κάνουν άλμα στον πραγματικό χειριστή

Ενέργειες του χειριστή

- Ανάγνωση αιτίου, και μετάβαση στο σχετικό χειριστή
- Καθορισμός απαιτούμενης ενέργειας
- Αν η εντολή είναι επανεκκινήσιμη (restartable)
  - Εκτέλεση διορθωτικής ενέργειας
  - Χρήση του EPC για επιστροφή στο πρόγραμμα
- Αλλιώς
  - Τερματισμός προγράμματος
  - Αναφορά σφάλματος με χρήση του EPC, του αιτίου, …

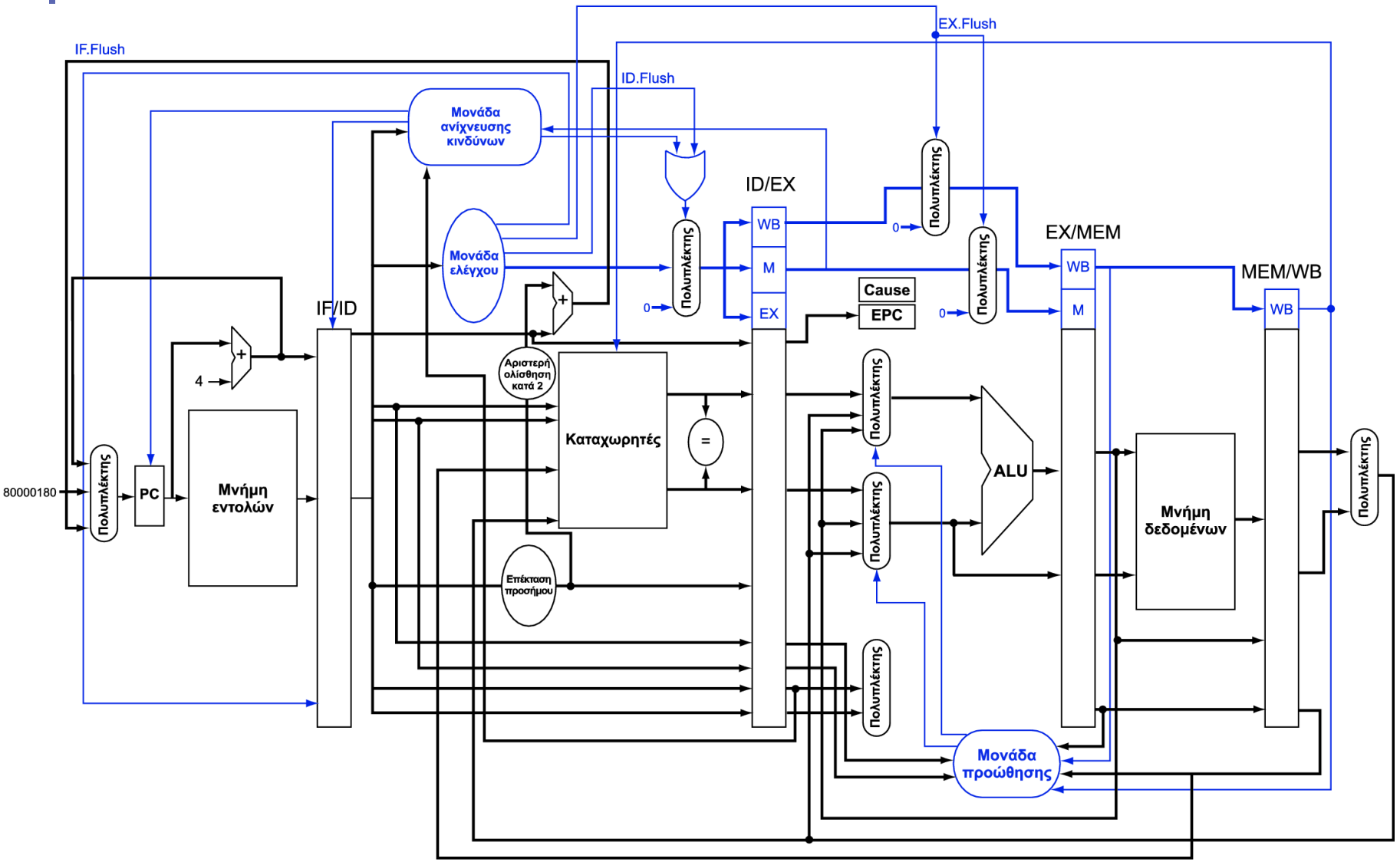
Εξαιρέσεις σε μια διοχέτευση

- Άλλη μορφή κινδύνου ελέγχου
- Θεωρήστε υπερχείλιση στην πρόσθεση στο στάδιο EX

```
add $1, $2, $1
```

  - Αποφυγή ζημιάς (εγγραφής) στον $1
  - Ολοκλήρωση των προηγούμενων εντολών
  - Εκκένωση της add και των επόμενων εντολών
  - Ρύθμιση τιμών των καταχωρητών Cause και EPC
  - Μεταφορά ελέγχου στο χειριστή
- Όμοια με λανθασμένη πρόβλεψη διακλάδωσης
  - Χρήση μεγάλου μέρους του ίδιου υλικού

Διοχέτευση με εξαιρέσεις



Πλάνες

- Η διοχέτευση είναι εύκολη (!)
  - Η βασική ιδέα είναι εύκολη
  - Ο διάβολος κρύβεται στις λεπτομέρειες
    - π.χ., ανίχνευση κινδύνων δεδομένων
- Η διοχέτευση είναι ανεξάρτητη από την τεχνολογία
  - Τότε γιατί δεν κάναμε πάντα διοχέτευση;
  - Τα περισσότερα τρανζίστορ κάνουν εφικτές τις πιο προηγμένες τεχνικές
  - Η σχεδίαση αρχιτεκτονικών συνόλου εντολών που σχετίζεται με τη διοχέτευση πρέπει να λαμβάνει υπόψη της τις τεχνολογικές τάσεις

Παγίδες

- Η φτωχή σχεδίαση της αρχιτεκτονικής συνόλου εντολών μπορεί να κάνει δυσκολότερη τη διοχέτευση
  - π.χ., πολύπλοκα σύνολα εντολών (VAX, IA-32)
    - Σημαντική επιβάρυνση για να δουλέψει η διοχέτευση
    - Προσέγγιση του IA-32 με μικρολειτουργίες (micro-ops)
  - π.χ., πολύπλοκοι τρόποι διευθυνσιοδότησης
    - Παρενέργειες ενημέρωσης καταχωρητών, εμμεσότητα μνήμης
  - π.χ., καθυστερημένες διακλαδώσεις
    - Οι προηγμένες διοχετεύσεις έχουν μεγάλες υποδοχές καθυστέρησης

Συμπερασματικές παρατηρήσεις

- Η αρχιτεκτονική συνόλου εντολών επηρεάζει τη σχεδίαση της διαδρομής δεδομένων και της μονάδας ελέγχου
- Η διαδρομή δεδομένων και η μονάδα ελέγχου επηρεάζουν τη σχεδίαση της αρχιτεκτονικής συνόλου εντολών
- Η διοχέτευση βελτιώνει τη διεκπεραιωτική ικανότητα εντολών με τη χρήση παραλληλίας
  - Περισσότερες εντολές ολοκληρώνονται ανά δευτερόλεπτο
  - Ο λανθάνων χρόνος κάθε εντολής δε μειώνεται
- Κίνδυνοι: δομής, δεδομένων, ελέγχου
- Πολλαπλή εκκίνηση και δυναμικός χρονοπρογραμματισμός (παραλληλία επιπέδου εντολής – ILP)
  - Οι εξαρτήσεις περιορίζουν την επιτεύξιμη παραλληλία
  - Η πολυπλοκότητα οδηγεί στο τείχος της ισχύος (power wall)