

Κεφάλαιο 7

Μικροαρχιτεκτονική

Αντώνης Πασχάλης, Ιωάννης Σίδερης



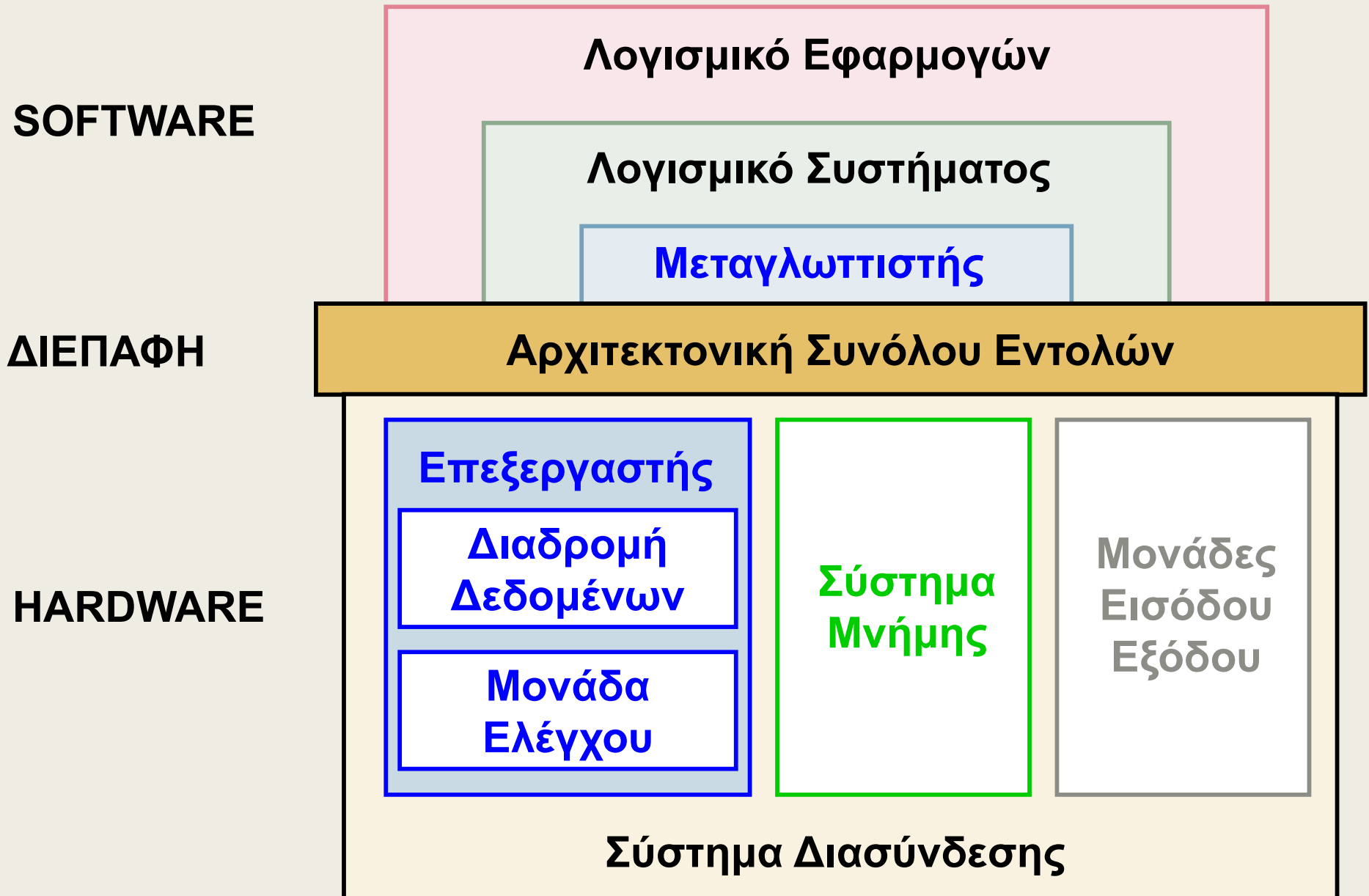
dscal
DIGITAL SYSTEMS & COMPUTER ARCHITECTURE LABORATORY

Περιεχόμενα κεφαλαίου 7

- Εισαγωγή
- Αρχιτεκτονικοί καταχωρητές
- Βασικές μικροαρχιτεκτονικές ARM
- Σχεδίαση μικροαρχιτεκτονικής ενός κύκλου
 - διαδρομή δεδομένων (*datapath*)
 - μονάδα ελέγχου (*control unit*)
 - ανάλυση επιδόσεων
- Σχεδίαση μικροαρχιτεκτονικής πολλών κύκλων
 - διαδρομή δεδομένων (*datapath*)
 - μονάδα ελέγχου (*control unit*)
 - ανάλυση επιδόσεων

Λογισμικό εφαρμογών	
Λειτουργικά συστήματα	
Αρχιτεκτονική	
Μικροαρχιτεκτονική	
Λογική	
Ψηφιακά κυκλώματα	
Αναλογικά κυκλώματα	
Διατάξεις	
Φυσική	

Ο υπολογιστής



Μικροαρχιτεκτονική (microarchitecture)

- Προσδιορίζει **πως υλοποιείται η αρχιτεκτονική στο υλικό**
 - Σχεδίαση της **διαδρομής δεδομένων (datapath)** του επεξεργαστή
 - χωροταξική διάταξη των καταχωρητών, των μνημών, των μονάδων ALU και των άλλων δομικών στοιχείων που απαρτίζουν τη διαδρομή δεδομένων
 - Σχεδίαση της **μονάδας ελέγχου (control unit)** του επεξεργαστή
 - παραγωγή των κατάλληλων σημάτων ελέγχου για τον χρονισμό του επεξεργαστή και των υπολοίπων διατάξεων με τη χρήση:
 - **συνδυαστικής λογικής** (στην περίπτωση επεξεργαστών ενός κύκλου και επεξεργαστών με διοχέτευση) ή
 - **μηχανών πεπερασμένων καταστάσεων** (στην περίπτωση επεξεργαστών πολλών κύκλων)
- Συχνά υπάρχουν πολλές διαφορετικές μικροαρχιτεκτονικές για την ίδια αρχιτεκτονική με διαφορετικούς συμβιβασμούς μεταξύ των τριών βασικών αξόνων σχεδίασης:
 - **επιδόσεις** (χρόνος εκτέλεσης προγράμματος)
 - **κόστος** (απαιτούμενοι πόροι, κατανάλωση ισχύος)
 - **πολυπλοκότητα** στη σχεδίαση

Αρχιτεκτονική κατάσταση

- Η **αρχιτεκτονική υπολογιστών** ορίζεται από το **σύνολο εντολών** της και από την **αρχιτεκτονική κατάσταση** της
- Η **αρχιτεκτονική κατάσταση** του επεξεργαστή ARM συνίσταται σε:
 - **16 καταχωρητές R0-R15 μεγέθους 32 bit** (αρχείο καταχωρητών)
 - ο καταχωρητής **R15** είναι ο **καταχωρητής προγράμματος** (program counter – PC)
 - **1 καταχωρητής κατάστασης τρέχοντος προγράμματος** (CPSR)
- Οι καταχωρητές που συνιστούν την αρχιτεκτονική κατάσταση ονομάζονται **αρχιτεκτονικοί καταχωρητές** (architectural registers)
 - *Κάθε μικροαρχιτεκτονική ARM πρέπει να υλοποιεί **όλους** τους αρχιτεκτονικούς καταχωρητές*
- Κάποιες μικροαρχιτεκτονικές περιλαμβάνουν πρόσθετους καταχωρητές, τους **μη αρχιτεκτονικούς καταχωρητές** (nonarchitectural registers)
 - *για να ελαχιστοποιήσουν τη λογική ή για να βελτιώσουν τις επιδόσεις*

τρέχουσα αρχιτεκτονική
κατάσταση

εκτέλεση εντολής
→
σύνολο δεδομένων

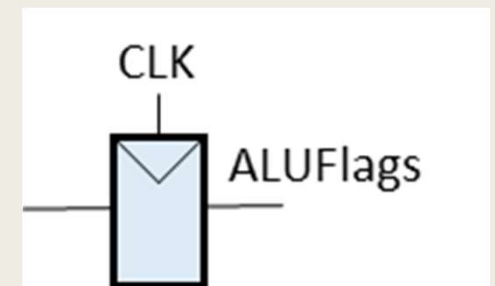
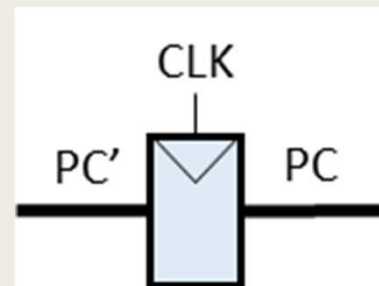
επόμενη αρχιτεκτονική
κατάσταση

Σχεδιαστική διαδικασία μικροαρχιτεκτονικής ARM

- Αρχικά, ορίζουμε μία μικροαρχιτεκτονική που απαρτίζεται από δύο διακριτές μονάδες που αλληλοεπιδρούν μεταξύ τους:
 - τη **διαδρομή δεδομένων** (*datapath*) των 32 bit, που εκτελεί λειτουργίες με λέξεις δεδομένων και περιέχει υπομονάδες όπως:
 - μνήμες εντολών και δεδομένων
 - αρχιτεκτονικούς και μη καταχωρητές
 - μονάδα ALU για την εκτέλεση πράξεων
 - πολυπλέκτες για τον έλεγχο της ροής δεδομένων
 - επιπλέον συνδυαστική λογική για τον υπολογισμό της επόμενης αρχιτεκτονικής κατάστασης και την επέκταση πρόσημου/μηδενός
 - τη **μονάδα ελέγχου** (*control unit*) που δίνει οδηγίες στη διαδρομή δεδομένων σχετικά με το πώς να εκτελέσει κάθε εντολή
 - παράγοντας και συγχρονίζοντας (όπου απαιτείται) τα κατάλληλα σήματα ελέγχου των υπομονάδων της διαδρομής δεδομένων
 - λαμβάνοντας υπόψη τα πεδία δήλωσης μνημονικών συνθήκης (*cond*), κωδικού λειτουργίας (*op*) και επεκτάσεών του (*funct*)
 - λαμβάνοντας υπόψη τις σημαίες συνθήκης (*N, Z, C, V*)

Σχεδιαστική διαδικασία μικροαρχιτεκτονικής ARM

- Στη συνέχεια, ξεκινάμε τη σχεδίαση της διαδρομής δεδομένων των 32 bit στο **επίπεδο μεταφοράς καταχωρητή (RTL)** με τον ορισμό των **αρχιτεκτονικών καταχωρητών** του επεξεργαστή
- **Μετρητής προγράμματος** (program counter - PC) των 32 bit
 - διαχωρίζεται από το αρχείο καταχωρητών, αλλά ανήκει σε αυτό
 - η έξοδός του (PC) δείχνει τη διεύθυνση της τρέχουσας εντολής
 - η είσοδός του (PC') υποδεικνύει τη διεύθυνση της επόμενης εντολής
 - αρχικοποιείται στην τιμή 0 με το σήμα RESET (δεν φαίνεται στο σχήμα)
- **Καταχωρητής κατάστασης** (τρέχοντος προγράμματος) (status register - SR) των 4 bit
 - αποθηκεύει τις τρέχουσες σημαίες συνθήκης (N, Z, C, V)
 - η ενημέρωσή του γίνεται στο τέλος της εκτέλεσης εντολών επεξεργασίας δεδομένων όταν το πεδίο S = 1 (π.χ. CMP, ADDS)
 - χρησιμοποιείται από τη μονάδα ελέγχου

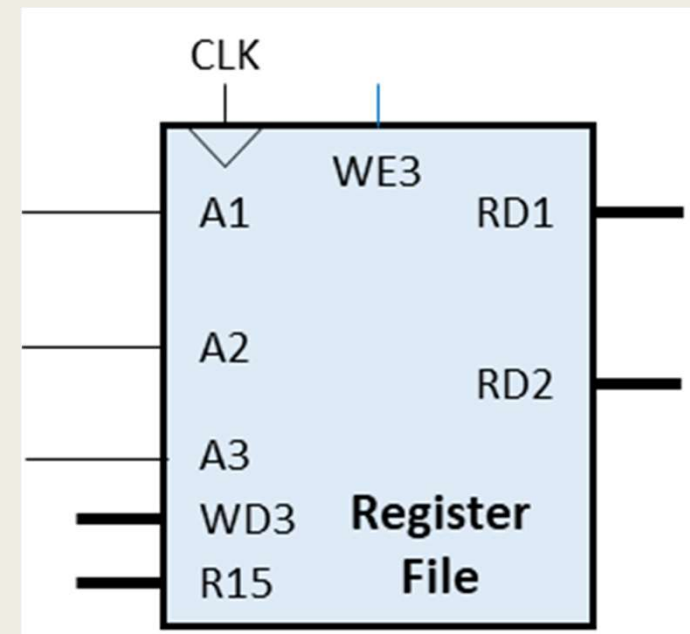


Σχεδιαστική διαδικασία μικροαρχιτεκτονικής ARM

■ Αρχείο καταχωρητών (register file - RF)

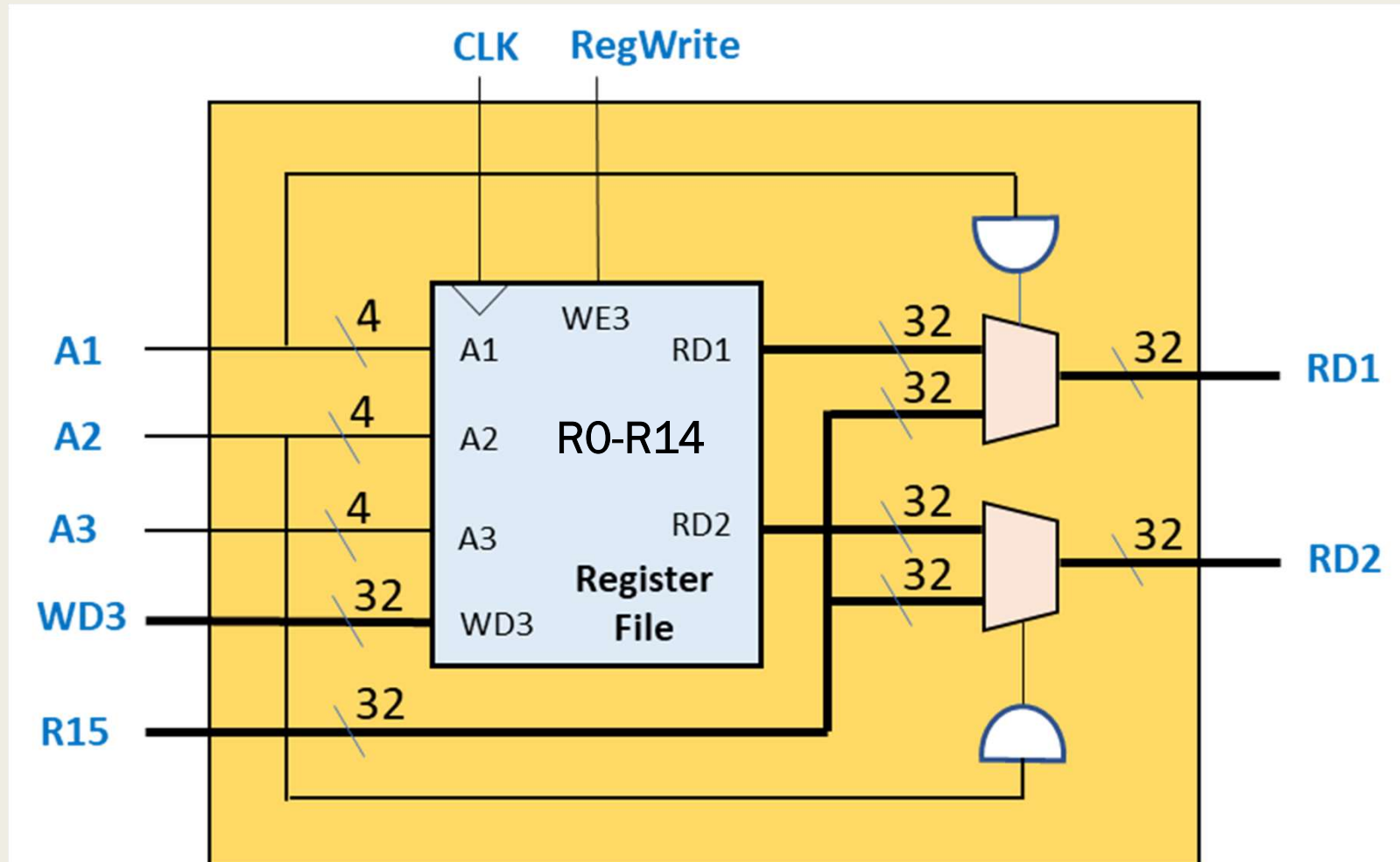
- περιέχει 15 καταχωρητές των 32 bit ($R0-R14$)
- διαθέτει δύο θύρες ανάγνωσης ($A1/RD1$ και $A2/RD2$) και μία θύρα εγγραφής ($A3/WD3$)
- καθεμία από τις διευθύνσεις των 4 bit ($A1$, $A2$ και $A3$) μπορεί να προσπελάσει και τους 15 καταχωρητές
- οι θύρες 1 και 2 διαβάζουν ασύγχρονα τα δεδομένα δύο καταχωρητών $A1$ και $A2$ και τα μεταφέρουν στις εξόδους $RD1$ και $RD2$, αντίστοιχα
- η θύρα 3 εγγράφει σύγχρονα (κατά την ανερχόμενη ακμή του ρολογιού) τα δεδομένα από την είσοδο $WD3$ στον καταχωρητή $A3$, όταν το σήμα ελέγχου $WE3$ έχει την τιμή 1
- η επιπλέον θύρα $R15$ μεταφέρει την τιμή $PC+8$ στην έξοδο $RD1/RD2$, μόνο όταν $A1/A2 = 15$, για τις ανάγκες της ανάγνωσης του μετρητή προγράμματος PC ($R15$)

Να μην ξεχνάμε ότι ο μετρητής προγράμματος PC ανήκει στο αρχείο καταχωρητών ($R15$)



Σχεδιαστική διαδικασία μικροαρχιτεκτονικής ARM

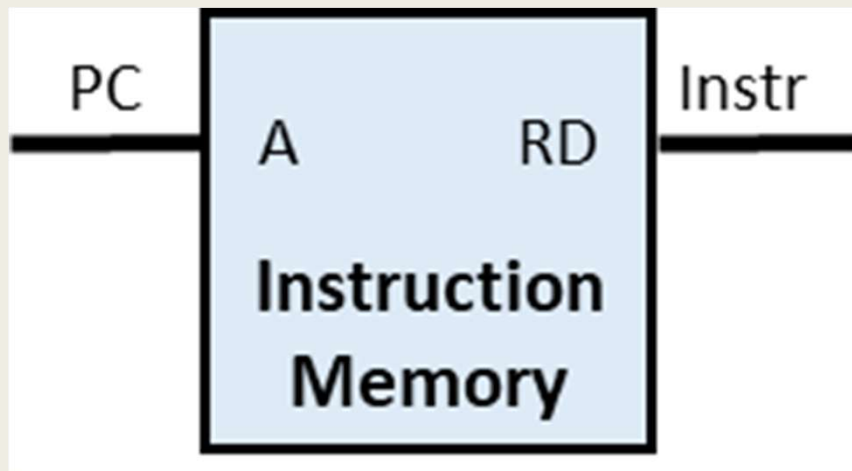
- Αρχείο καταχωρητών (register file - RF)



Να μην ξεχνάμε ότι ο μετρητής προγράμματος PC ανήκει στο αρχείο καταχωρητών (R15)

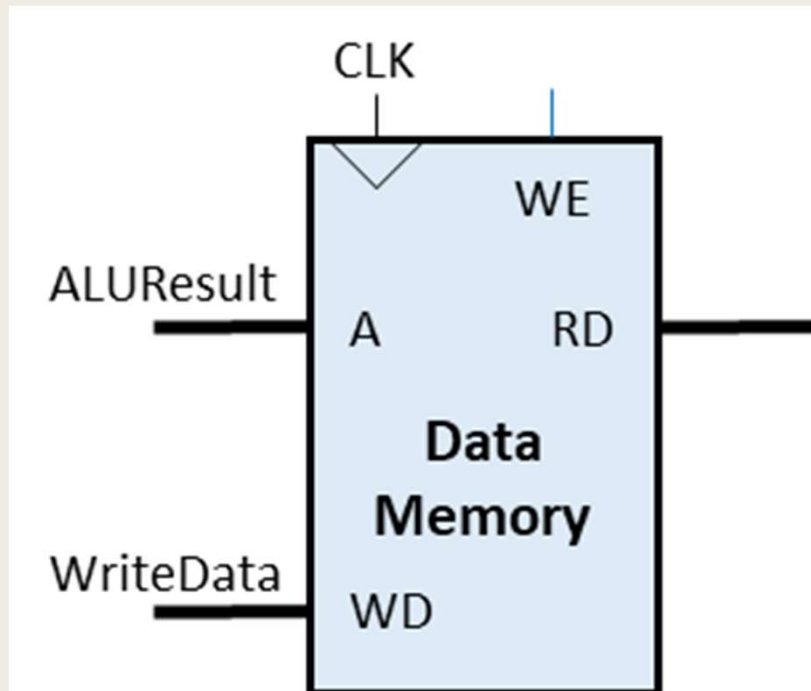
Σχεδιαστική διαδικασία μικροαρχιτεκτονικής ARM

- Στη συνέχεια, προχωράμε τη σχεδίαση της διαδρομής δεδομένων των 32 bit στο **επίπεδο μεταφοράς καταχωρητή (RTL)** με τον ορισμό των **μνημών εντολών και δεδομένων** του επεξεργαστή
- **Μνήμη εντολών** (instruction memory - IM) των 2^N λέξεων των 32 bit
 - δέχεται ως είσοδο (*A*) τη διεύθυνση των *N* bit της τρέχουσας εντολής
 - η διεύθυνση της τρέχουσας εντολής είναι αποθηκευμένη στον μετρητή προγράμματος, ισχύει: $A[N-1:0] = PC[N+1:2]$
 - μεταφέρει στην έξοδο δεδομένων ανάγνωσης (*RD*) την εντολή (*Instr*) σε γλώσσα μηχανής των 32 bit που είναι αποθηκευμένη στη διεύθυνση *A*
 - στην απλουστευμένη σχεδίαση αντιμετωπίζεται ως **μνήμη ROM** (συνδυαστικό κύκλωμα)



Σχεδιαστική διαδικασία μικροαρχιτεκτονικής ARM

- **Μνήμη δεδομένων** (data memory - DM) των 2^M λέξεων των 32 bit
 - δέχεται ως είσοδο (A) τη διεύθυνση των M bit των δεδομένων που διαβάζονται ή γράφονται στη μνήμη δεδομένων
 - η διεύθυνση δεδομένων υπολογίζεται στη μονάδα ALU, ισχύει $A[M-1:0] = \text{ALUResult}[M+1:2]$
 - διαθέτει μία θύρα (A/RD) για ανάγνωση ή (A/WD) για εγγραφή
 - η ανάγνωση γίνεται σύγχρονα (Block RAM) ή ασύγχρονα (Distributed RAM)
 - η εγγραφή γίνεται σύγχρονα, όταν το σήμα ελέγχου WE έχει την τιμή 1



Στην περίπτωση της μικροαρχιτεκτονικής του ενός κύκλου η ανάγνωση γίνεται μόνο ασύγχρονα με Distributed RAM

Σχεδιαστική διαδικασία μικροαρχιτεκτονικής ARM

- Στη συνέχεια, επιλέγουμε ποια από τις βασικές μικροαρχιτεκτονικές θα υλοποιήσουμε:
 - έχουμε τρεις επιλογές διαφορετικής πολυπλοκότητας από την πιο απλή μέχρι την πιο σύνθετη
 - για εκπαιδευτικούς λόγους ξεκινάμε από την πιο απλή
- **Μικροαρχιτεκτονική ενός κύκλου**
 - διαδρομή δεδομένων ενός κύκλου
 - μονάδα ελέγχου ως συνδυαστική λογική
 - χρήση μόνο αρχιτεκτονικών καταχωρητών
 - απαίτηση ξεχωριστών μνημών για εντολές και δεδομένα
 - ολόκληρη η εντολή εκτελείται σε έναν κύκλο ρολογιού
 - η περίοδος του CLK προσδιορίζεται από την εντολή LDR που έχει τη μεγαλύτερη καθυστέρηση διάδοσης

Σχεδιαστική διαδικασία μικροαρχιτεκτονικής ARM

■ Μικροαρχιτεκτονική πολλών κύκλων

- διαδρομή δεδομένων πολλών κύκλων
- μονάδα ελέγχου ως μηχανή πεπερασμένων καταστάσεων (FSM)
- χρήση αρχιτεκτονικών, αλλά και μη αρχιτεκτονικών καταχωρητών
 - προσωρινοί καταχωρητές που τοποθετούνται ανάμεσα σε τμήματα συνδυαστικής λογικής, ώστε να διαιρεθεί η συνδυαστική λογική της διαδρομής δεδομένων σε μικρότερα στάδια τα οποία μπορούν να εκτελούνται με γρηγορότερο CLK
- δυνατότητα κοινής μνήμης για εντολές και δεδομένα
 - θα διατηρήσουμε τις ξεχωριστές μνήμες για λιγότερες αλλαγές στη διαδρομή δεδομένων
- κάθε φορά εκτελείται μόνο μία εντολή, αλλά κάθε εντολή χρειάζεται περισσότερους από έναν κύκλους του ρολογιού για να ολοκληρωθεί
 - οι απλούστερες εντολές εκτελούνται σε λιγότερους κύκλους από ό,τι οι πιο περίπλοκες (π.χ. η εντολή LDR σε 5 κύκλους)
 - η περίοδος του CLK προσδιορίζεται από **το πιο αργό στάδιο**
- χρησιμοποιείται σε απλούς ενσωματωμένους επεξεργαστές

Σχεδιαστική διαδικασία μικροαρχιτεκτονικής ARM

■ Μικροαρχιτεκτονική με διοχέτευση (pipeline)

- διαδρομή δεδομένων με διοχέτευση
- μονάδα ελέγχου ως συνδυαστική λογική, αλλά με χρονισμό των σημάτων ελέγχου μέσω καταχωρητών διοχέτευσης
- χρήση αρχιτεκτονικών, αλλά και μη αρχιτεκτονικών καταχωρητών
 - καταχωρητές διοχέτευσης που τοποθετούνται ανάμεσα σε τμήματα συνδυαστικής λογικής, ώστε να διαιρεθεί η συνδυαστική λογική της διαδρομής δεδομένων σε μικρότερα στάδια τα οποία μπορούν να εκτελούνται με γρηγορότερο CLK
- απαίτηση ξεχωριστών μνημών για εντολές και δεδομένα
- σε κάθε στάδιο της διαδρομής δεδομένων εκτελείται **άλλη εντολή**
 - πολλές εντολές εκτελούνται ταυτόχρονα, άρα βελτιώνεται σημαντικά η διεκπεραιωτική ικανότητα (throughput)
 - όλες οι εντολές εκτελούνται σε τόσους κύκλους όσο είναι και τα διαφορετικά στάδια της διαδρομής δεδομένων (συνήθως σε 5 κύκλους)
 - η περίοδος του CLK προσδιορίζεται από **το πιο αργό στάδιο**
- απαίτηση για προσθήκη λογικής για το **χειρισμό των εξαρτήσεων μεταξύ ταυτόχρονα εκτελούμενων εντολών**
- χρησιμοποιείται στους περισσότερους επεξεργαστές σήμερα

Σχεδιαστική διαδικασία μικροαρχιτεκτονικής ARM

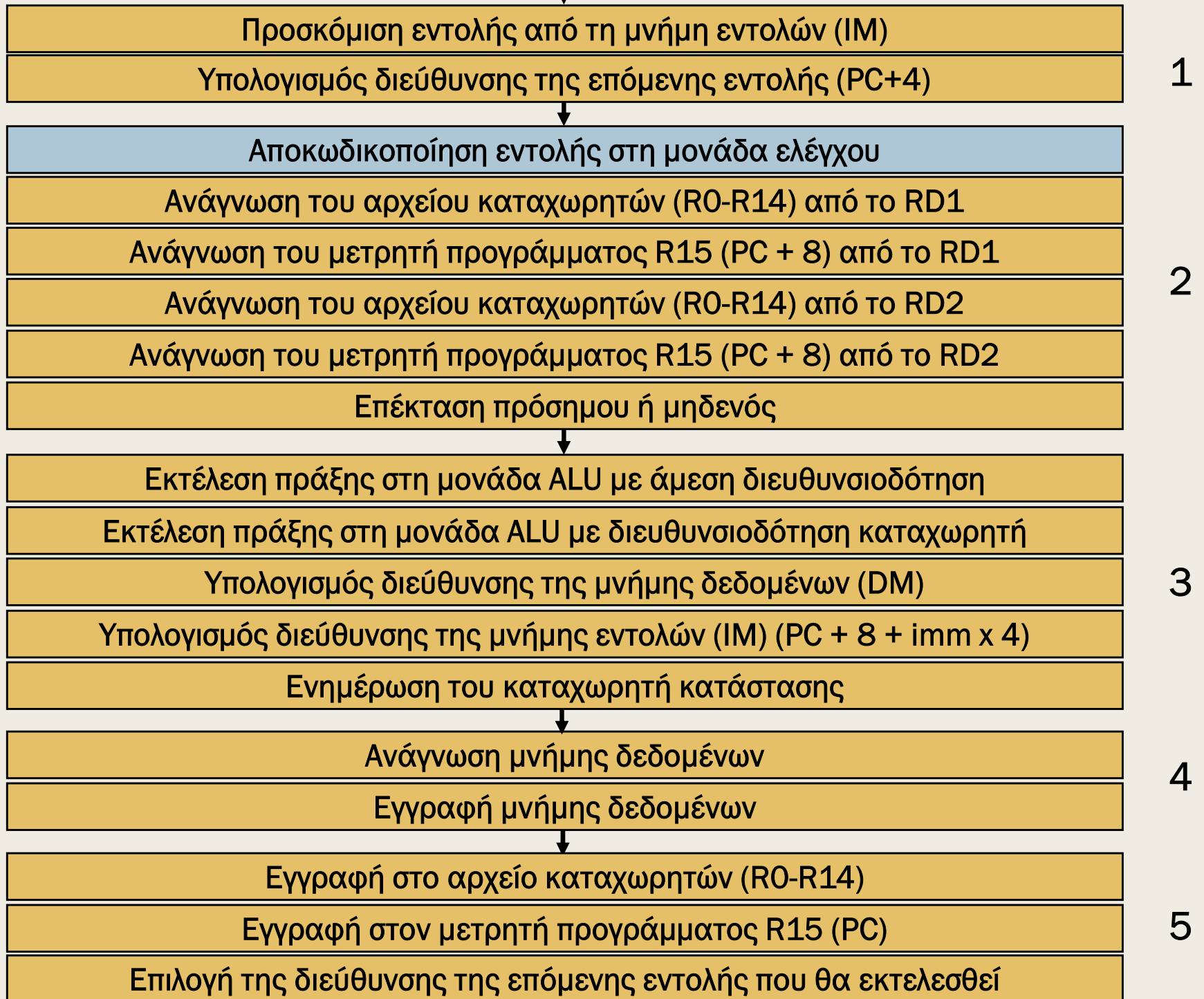
- Στη συνέχεια, επιλέγουμε ποιες εντολές θα υλοποιήσουμε:
 - **Εντολές επεξεργασίας δεδομένων: *ADD(S)*, *SUB(S)*, *AND(S)*, *ORR(S)*, *CMP***
 - εκτελούνται υπό συνθήκη
 - ενημερώνουν τον καταχωρητή καταστάσεων, εάν το πεδίο $S = 1$
 - υποστηρίζουν διευθυνσιοδότηση καταχωρητή
 - υποστηρίζουν άμεση διευθυνσιοδότηση με μη προσημασμένο άμεσο τελεστέο των 8 bit
 - δεν υποστηρίζουν ολίσθηση του περιεχομένου του καταχωρητή και περιστροφή του άμεσου τελεστέου
 - **Εντολές μνήμης: *LDR*, *STR***
 - εκτελούνται υπό συνθήκη
 - υποστηρίζουν διευθυνσιοδότηση βάσης με σχετική απόσταση μη προσημασμένου άμεσου τελεστέου των 12 bit
 - **Εντολές διακλάδωσης: *B***
 - εκτελούνται υπό συνθήκη
 - υποστηρίζουν PC-σχετική διευθυνσιοδότηση με προσημασμένο ακέραιο σε αναπαράσταση συμπληρώματος ως προς δύο των 24 bit
 - η διεύθυνση - στόχος της διακλάδωσης (branch target address, BTA) υπολογίζεται ως εξής: $BTA = PC + 8 + imm24 \times 4$

Επεξεργαστής ενός κύκλου

- Στη συνέχεια, προχωρούμε στη σχεδίαση της **μικροαρχιτεκτονικής ενός κύκλου** που υλοποιεί τις εντολές που αναφέραμε
 - με βάση τους **αρχιτεκτονικούς καταχωρητές** και τις **μνήμες εντολών και δεδομένων**, προσθέτουμε την απαραίτητη **συνδυαστική λογική** μελετώντας τις **λειτουργίες** των εντολών που θα υλοποιήσουμε
 - Οι λειτουργίες ταξινομούνται σε πέντε βήματα εκτέλεσης της εντολής:
 1. Προσκόμιση εντολής και υπολογισμός επόμενης διεύθυνσης (PC+4)
 2. Αποκωδικοποίηση εντολής και ανάγνωση αρχείου καταχωρητών
 3. Εκτέλεση πράξεων στη μονάδα ALU
 4. Ανάγνωση ή εγγραφή στη μνήμη δεδομένων
 5. Ετεροχρονισμένη εγγραφή στο αρχείο καταχωρητών και επιλογή της διεύθυνσης της επόμενης εντολής
- Κατά τη σχεδίαση προβαίνουμε σε:
 - **μελέτη διαδρομής δεδομένων**
 - **μελέτη μονάδας ελέγχου**
 - **ανάλυση επιδόσεων**

Στα σχήματα που ακολουθούν, η σχετική με τη λειτουργία ροή δεδομένων φαίνεται με κόκκινο χρώμα

Λειτουργίες εντολών (18)

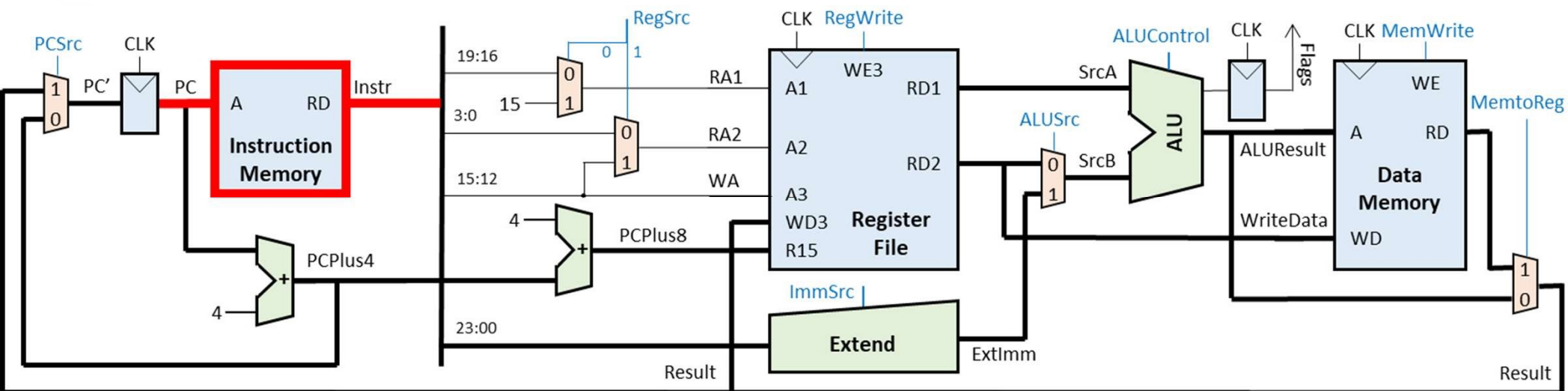


Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων

Προσκόμιση εντολής από τη μνήμη εντολών (IM)

- Ο μετρητής προγράμματος PC περιέχει τη διεύθυνση της τρέχουσας εντολής και διευθυνσιοδοτεί την είσοδο διευθύνσεων A της μνήμης εντολών (IM)
- Η εντολή διαβάζεται ασύγχρονα στη θύρα ανάγνωσης RD της IM των $2^N \times 32$ bit
 - Ο μετρητής προγράμματος αποθηκεύει ευθυγραμμισμένες διευθύνσεις εντολών για μνήμες οργανωμένες σε byte, ενώ η μνήμη εντολών είναι οργανωμένη σε λέξεις
 - Ισχύει $A[N-1:0] = PC[N+1:2]$

Λειτουργία κοινή για όλες τις εντολές

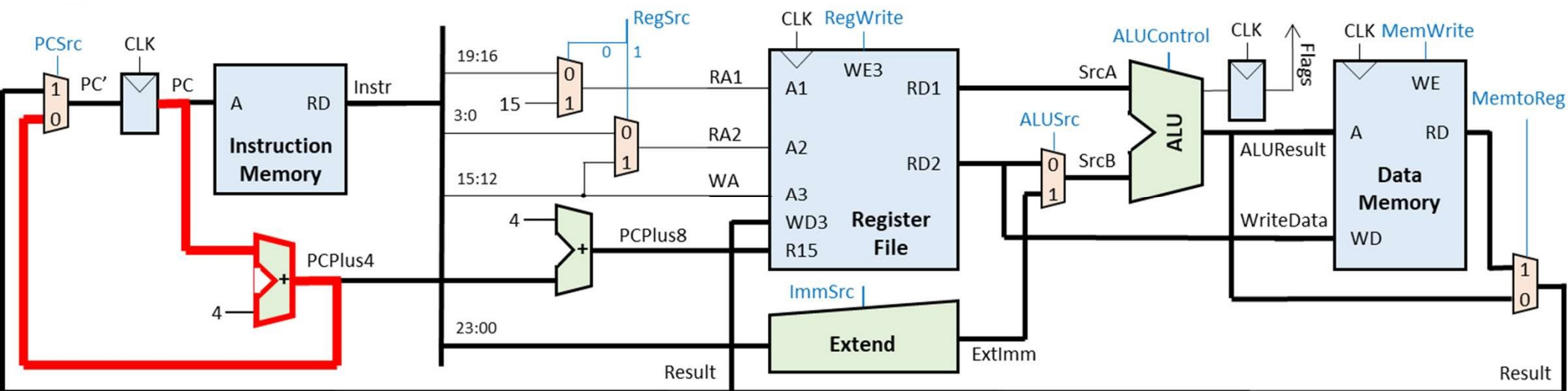


Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων

Υπολογισμός διεύθυνσης της επόμενης εντολής (PC+4)

- Υπολογισμός της διεύθυνσης της αμέσως επόμενης εντολής **PC+4** (παράλληλα με την εκτέλεση της τρέχουσας εντολής)
 - σύνδεση της εξόδου *PC* του μετρητή προγράμματος με την είσοδο του **αθροιστή κατά 4** (*incrementer*)
 - εμφάνιση του αποτελέσματος της αύξησης κατά 4 στην έξοδο *PCPlus4* του αθροιστή κατά 4
 - σύνδεση της εξόδου *PCPlus4* με τον πολυπλέκτη επιλογής διεύθυνσης επόμενης εντολής

Λειτουργία κοινή για όλες τις εντολές



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή LDR

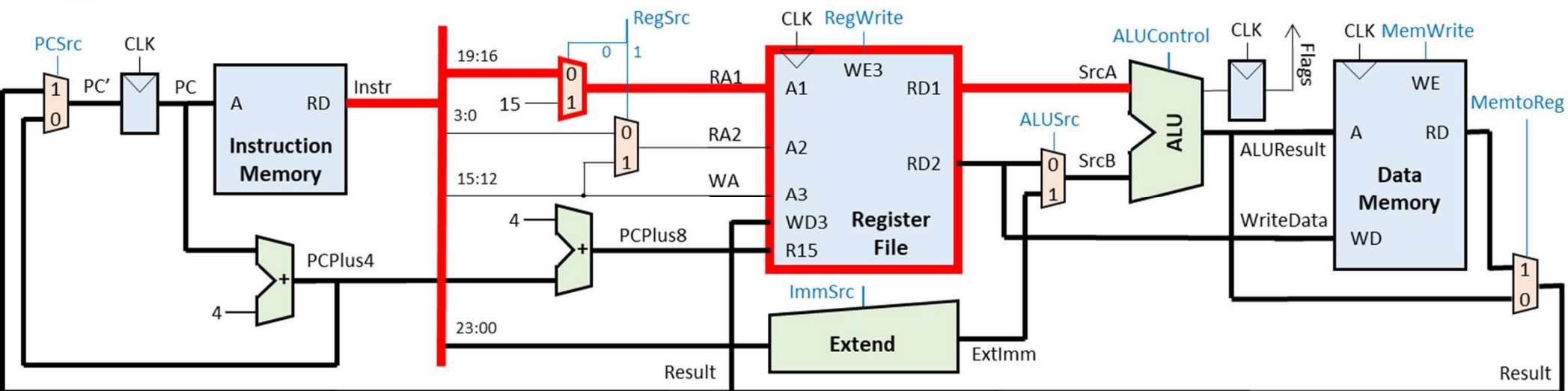
- Αρχικά, θα μελετήσουμε τη διαδρομή δεδομένων που υλοποιεί την εντολή **LDR** ($L = 1$)
- Κώδικας συμβολικής γλώσσας:
 - $LDR\ Rd,\ [Rn,\ \#imm12];\ Rd = DM[Rn + \#imm12]$ ($U = 1$)
 - $LDR\ Rd,\ [Rn,\ \#-imm12];\ Rd = DM[Rn - \#imm12]$ ($U = 0$)
- Στη μορφή της εντολής ορίζονται:
 - οι διευθύνσεις του **καταχωρητή προέλευσης Rn** και του **καταχωρητή προορισμού Rd**
 - ένας **μη προσημασμένος άμεσος τελεστέος των 12 bit**
- Απαιτείται η εκτέλεση των πράξεων της **πρόσθεσης** και της **αφαίρεσης** στη μονάδα ALU



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή LDR

Ανάγνωση του αρχείου καταχωρητών (R0-R14) από το RD1

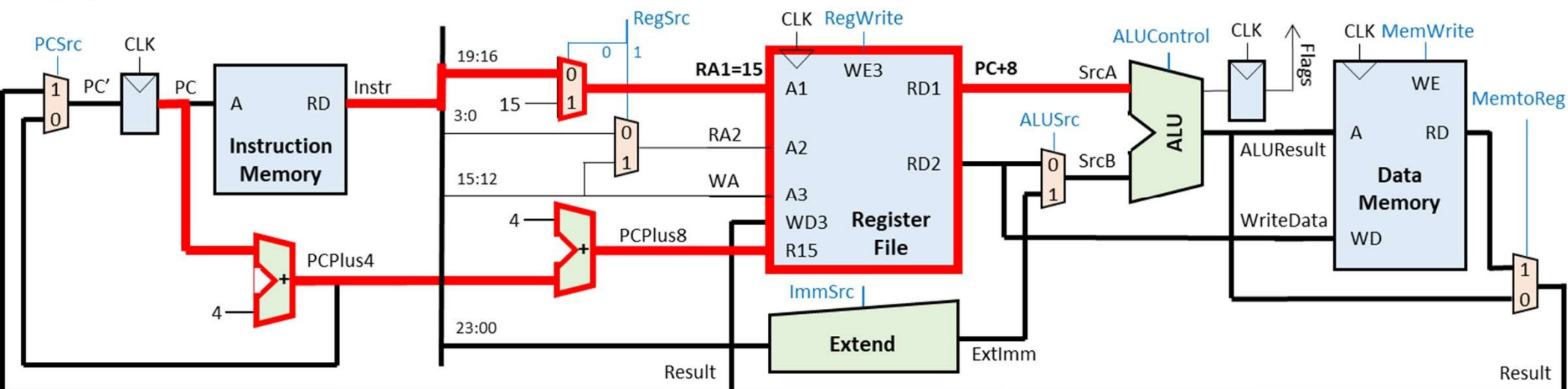
- Η διεύθυνση βάσης που είναι αποθηκευμένη στον **καταχωρητή προέλευσης Rn** του αρχείου καταχωρητών (R0-R14) διαβάζεται στη θύρα ανάγνωσης RD1
 - σύνδεση του πεδίου Rn της εντολής ($Instr_{19:16}$) στην είσοδο διευθύνσεων ανάγνωσης A1
 - ανάγνωση περιεχομένου καταχωρητή από τη θύρα ανάγνωσης RD1
 - σήματα ελέγχου: **RegScr[1:0] = X0** και **RegWrite = 0** (ανάγνωση αρχείου καταχ.)



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή LDR

Ανάγνωση του μετρητή προγράμματος R15 (PC + 8) από το RD1

- Η διεύθυνση βάσης, που είναι η διεύθυνση της τρέχουσας εντολής και είναι αποθηκευμένη στον **μετρητή προγράμματος PC** ($Rn = 15$), διαβάζεται στη θύρα ανάγνωσης RD1
 - ορίζεται ο μετρητής προγράμματος PC ως **καταχωρητής προέλευσης Rn** της εντολής
 - η ανάγνωση του R15 του αρχείου καταχωρητών επιστρέφει πάντα την τιμή **PC + 8**
 - σύνδεση του πεδίου Rn της εντολής ($Instr_{19:16}$) στην είσοδο διευθύνσεων ανάγνωσης A1
 - σύνδεση της εξόδου PC του μετρητή προγράμματος με την είσοδο R15 του αρχείου καταχωρητών μέσω δύο αθροιστών κατά 4, ώστε να υπολογιστεί το PC+8
 - απαίτηση για έναν επιπλέον αθροιστή κατά 4 με είσοδο PCPlus4 και έξοδο PCPlus8
 - μεταφορά της εισόδου R15 στη θύρα ανάγνωσης RD1 (μέσω εσωτερικού πολυπλέκτη)
 - σήματα ελέγχου: **RegScr[1:0] = X0** και **RegWrite = 0** (ανάγνωση αρχείου καταχωρητών)

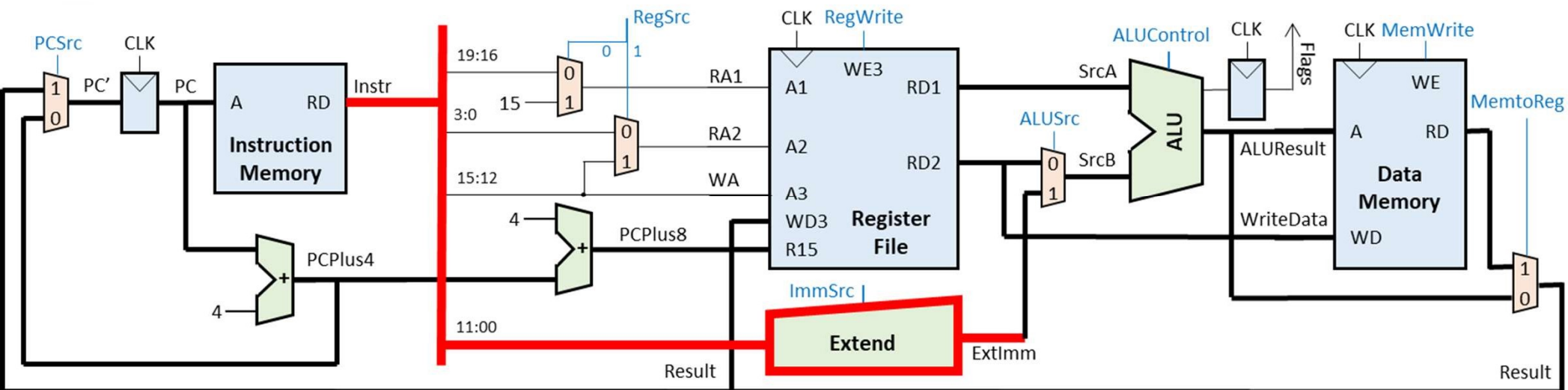


Χρησιμοποιείται όταν είναι κοινή η μνήμη εντολών και δεδομένων

Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή LDR

Επέκταση πρόσημου ή μηδενός

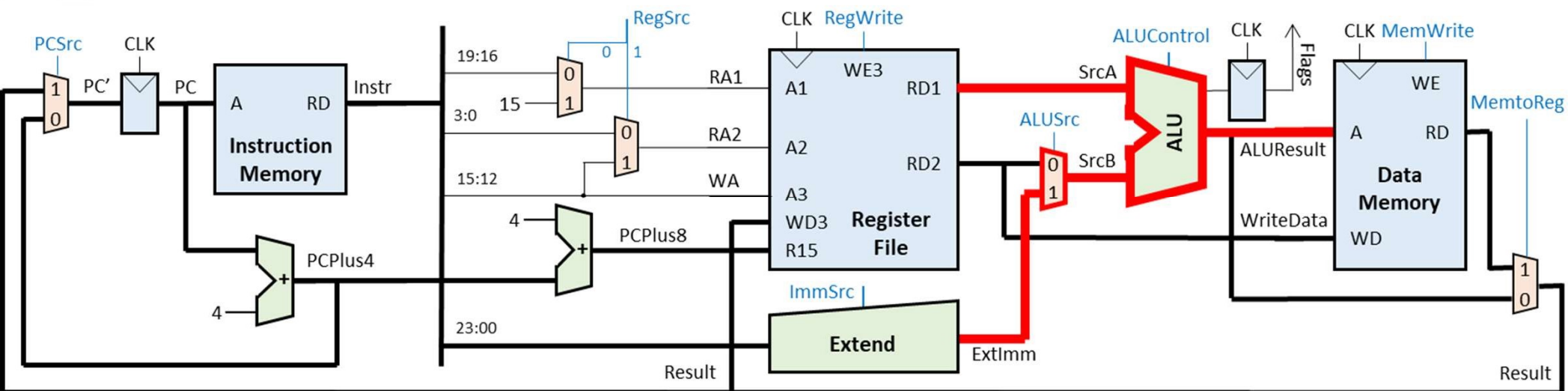
- Η εντολή LDR απαιτεί και μια **σχετική απόσταση** για τον υπολογισμό της διεύθυνσης δεδομένων:
 - σύνδεση του πεδίου άμεσου τελεστέου ($imm12$) της εντολής ($Instr_{11:0}$) στην είσοδο της μονάδας *Extend*
 - στη μονάδα *Extend* εκτελείται **επέκταση μηδενός** στα 32 bit του **μη προσημασμένου άμεσου τελεστέου των 12 bit** και το αποτέλεσμα εμφανίζεται στην έξοδο *ExtImm*
 - σήματα ελέγχου: **ImmScr = 0**



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή LDR

Υπολογισμός διεύθυνσης της μνήμης δεδομένων (DM)

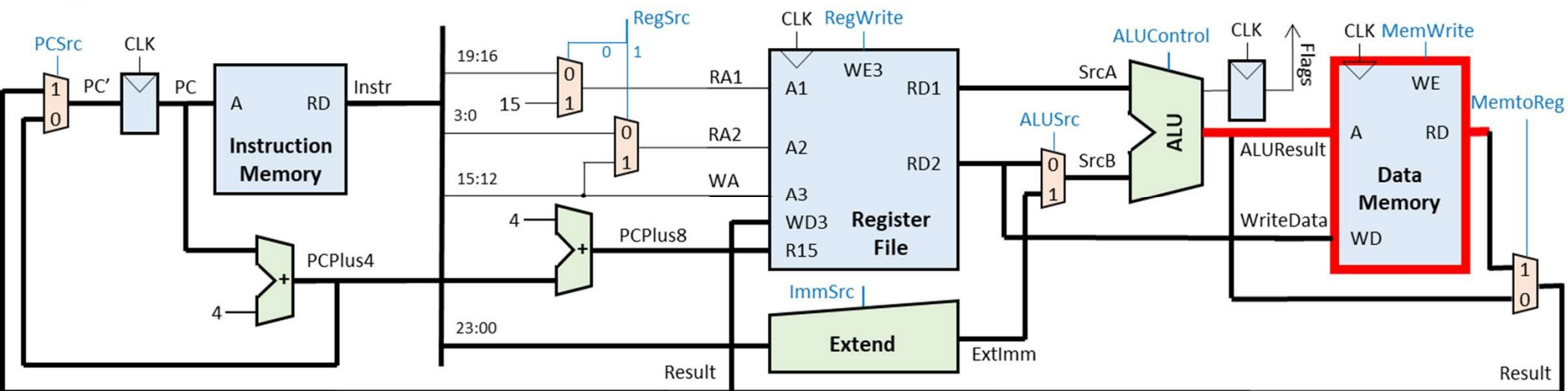
- Η **σχετική απόσταση προστίθεται** στην ή **αφαιρείται** από τη διεύθυνση βάσης στη μονάδα ALU, ώστε να προκύψει η διεύθυνση της μνήμης δεδομένων
 - σύνδεση της θύρας *RD1* του αρχείου καταχωρητών με την είσοδο *SrcA* των 32 bit της μονάδας ALU
 - σύνδεση της εξόδου *ExtImm* της μονάδας *Extend* με την είσοδο *SrcB* των 32 bit της μονάδας ALU
 - εκτελείται πρόσθεση ή αφαίρεση και το αποτέλεσμα της πράξης **$Rn \pm \#imm12$** εμφανίζεται στην έξοδο *ALUResult* των 32 bit της μονάδας ALU
 - σήματα ελέγχου: **$ALUSrc = 1$** και **$ALUControl[1:0] = 00 (+)$** ή **$01 (-)$**



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή LDR

Ανάγνωση μνήμης δεδομένων

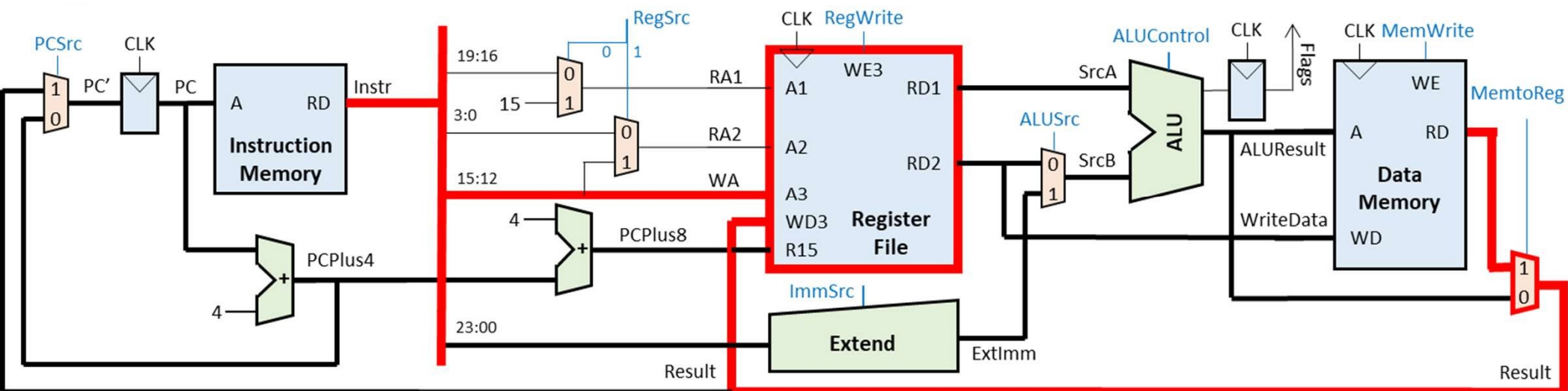
- Η διεύθυνση που εμφανίζεται στην έξοδο ALUResult της μονάδας ALU διευθύνει την είσοδο διευθύνσεων A της μνήμης δεδομένων (DM)
- Η λέξη δεδομένων διαβάζεται σύγχρονα (Block RAM) ή ασύγχρονα (Distributed RAM) στη θύρα ανάγνωσης RD της DM των $2^M \times 32$ bit
 - η διεύθυνση της λέξης δεδομένων είναι ευθυγραμμισμένη και αφορά μνήμες οργανωμένες σε *byte*, ενώ η μνήμη δεδομένων DM είναι οργανωμένη σε λέξεις
 - ισχύει $A[M-1:0] = ALUResult[M+1:2]$
 - σήματα ελέγχου: **MemWrite = 0** (διάβασμα μνήμης δεδομένων)



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή LDR

Εγγραφή στο αρχείο καταχωρητών (R0-R14)

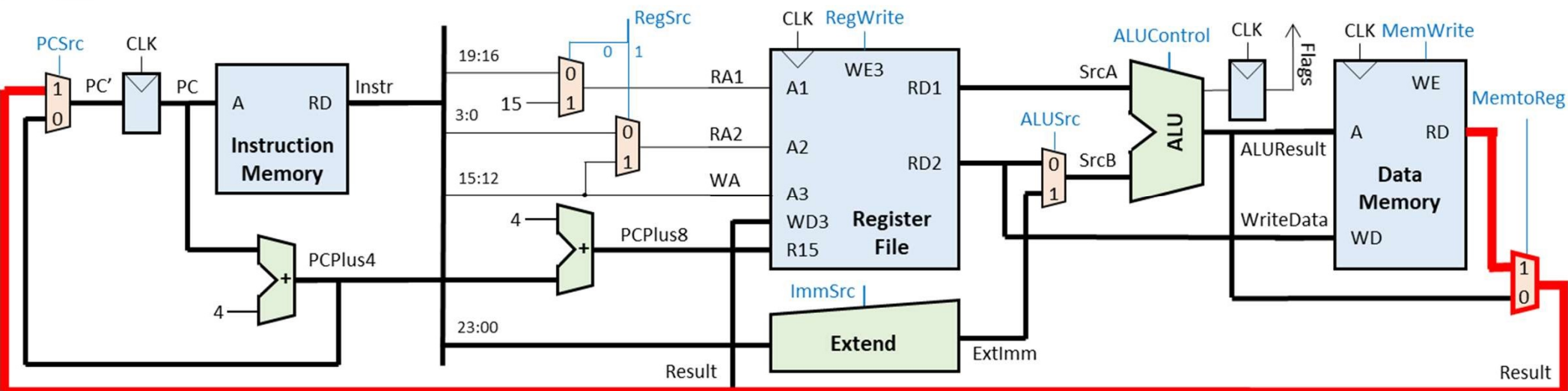
- Η λέξη δεδομένων που διαβάζεται από τη μνήμη δεδομένων αποθηκεύεται στον καταχωρητή προορισμού **Rd** του αρχείου καταχωρητών
 - σύνδεση του πεδίου *Rd* της εντολής ($Instr_{15:12}$) στην είσοδο διευθύνσεων εγγραφής *A3*
 - σύνδεση της θύρας ανάγνωσης *RD* της μνήμης δεδομένων με τη θύρα εγγραφής *WD3* του αρχείου καταχωρητών
 - εγγραφή περιεχομένου καταχωρητή από τη θύρα εγγραφής *WD3*
 - σήματα ελέγχου: **MemtoReg = 1** και **RegWrite = 1** (εγγραφή αρχείου καταχωρητών)



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή LDR

Εγγραφή στον μετρητή προγράμματος R15 (PC)

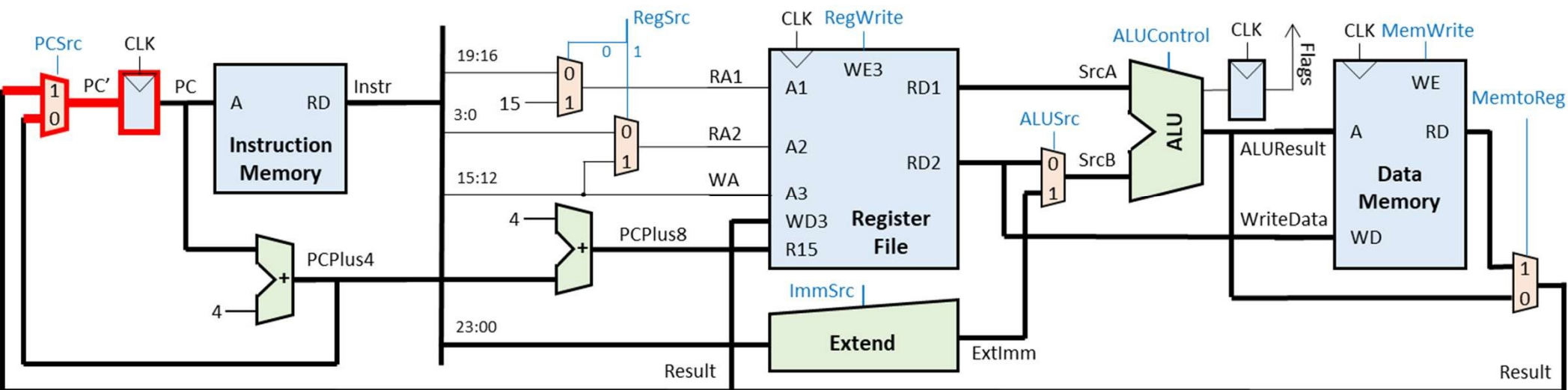
- Η λέξη δεδομένων που διαβάζεται από τη μνήμη δεδομένων αποθηκεύεται στον μετρητή προγράμματος PC (Rd=15) του αρχείου καταχωρητών
 - ορίζεται ο μετρητής προγράμματος PC ως καταχωρητής προορισμού Rd της εντολής LDR
 - η λέξη δεδομένων είναι διεύθυνση της μνήμης εντολών
 - σύνδεση της θύρας ανάγνωσης RD της μνήμης δεδομένων με τον πολυπλέκτη επιλογής διεύθυνσης επόμενης εντολής
 - σήματα ελέγχου: MemtoReg = 1, RegWrite = 1



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή LDR

Επιλογή της διεύθυνσης της επόμενης εντολής που θα εκτελεσθεί

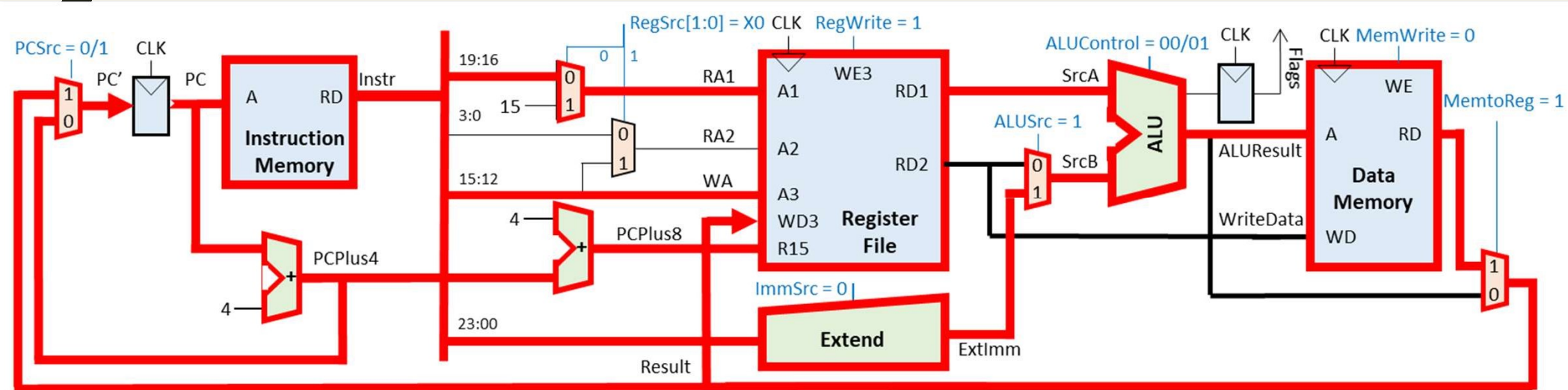
- Ο πολυπλέκτης επιλογής διεύθυνσης επόμενης εντολής επιλέγει ανάμεσα σε δύο πιθανές διευθύνσεις: $PC' = PC + 4$ ή $PC' = DM[Rn \pm \#imm12]$
- Η νέα διεύθυνση αποθηκεύεται στον μετρητή προγράμματος στην επόμενη ανερχόμενη ακμή του CLK
 - σύνδεση της εισόδου 0 του πολυπλέκτη με την έξοδο PCPlus4 του αθροιστή κατά 4
 - σύνδεση της εισόδου 1 του πολυπλέκτη με τη θύρα ανάγνωσης RD της μνήμης DM
 - σύνδεση της εξόδου του πολυπλέκτη με την είσοδο PC' του μετρητή προγράμματος
 - σήματα ελέγχου: $PCSrc = 1$ (RD) ή 0 (PCPlus4), $MemtoReg = 1$



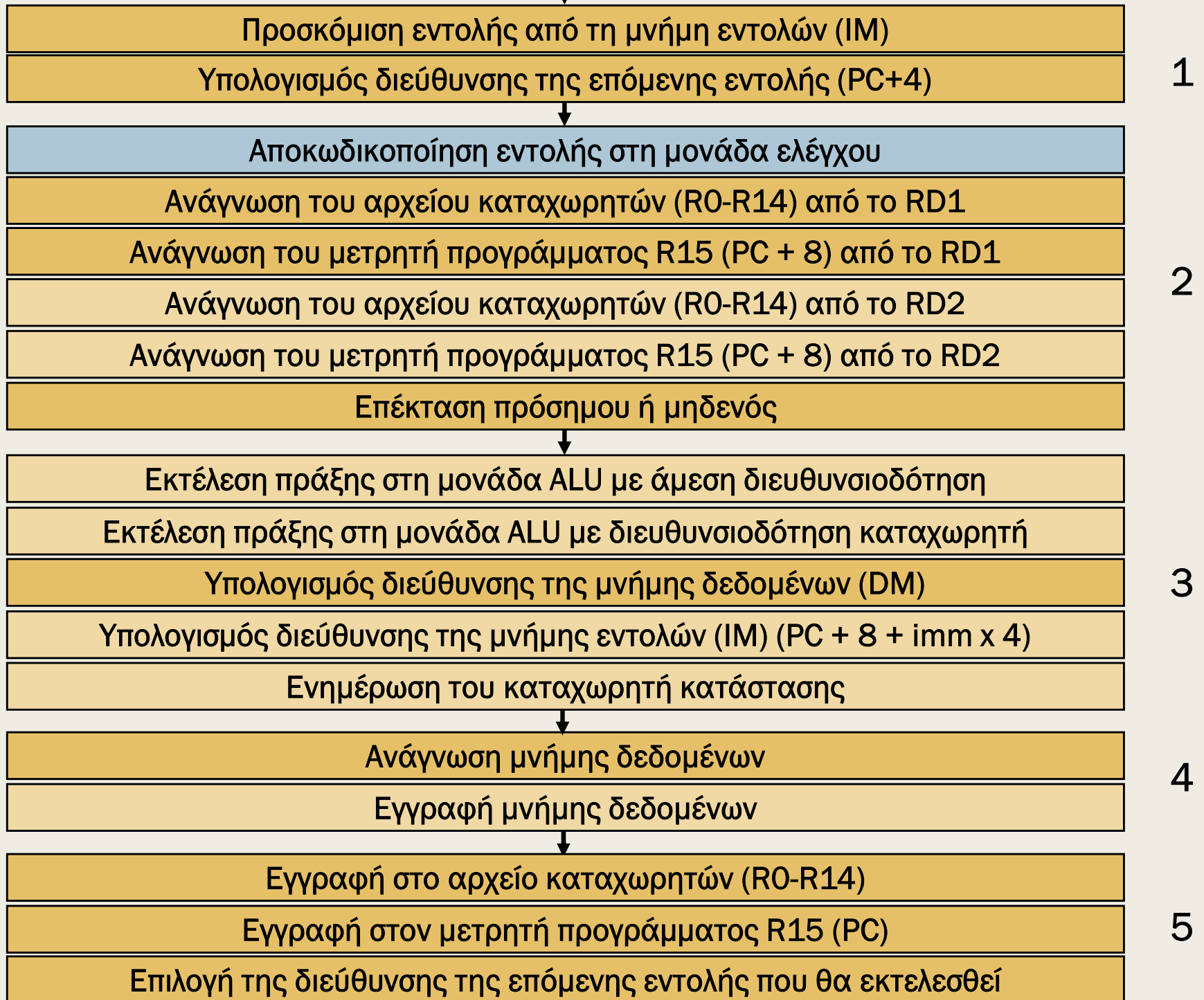
Προσοχή! Υπάρχει και μία τρίτη πιθανή διεύθυνση από την έξοδο της ALU για $MemtoReg = 0$

Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή LDR

- Συνολική ροή δεδομένων και τιμές στα σήματα ελέγχου:



Λειτουργίες εντολής LDR



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή STR

- Στη συνέχεια, θα μελετήσουμε τη διαδρομή δεδομένων που υλοποιεί την εντολή **STR** ($L = 0$)
- Κώδικας συμβολικής γλώσσας:
 - $STR\ Rd, [Rn, \#imm12];\ DM[Rn + \#imm12] = Rd\ (U = 1)$
 - $STR\ Rd, [Rn, \#-imm12];\ DM[Rn - \#imm12] = Rd\ (U = 0)$
- Στη μορφή της εντολής ορίζονται:
 - οι διευθύνσεις του **καταχωρητή προέλευσης Rn** και του **δεύτερου καταχωρητή προέλευσης Rd** και όχι προορισμού
 - ένας **μη προσημασμένος άμεσος τελεστέος των 12 bit**
- Απαιτείται η εκτέλεση των πράξεων της **πρόσθεσης** και της **αφαίρεσης** στη μονάδα ALU



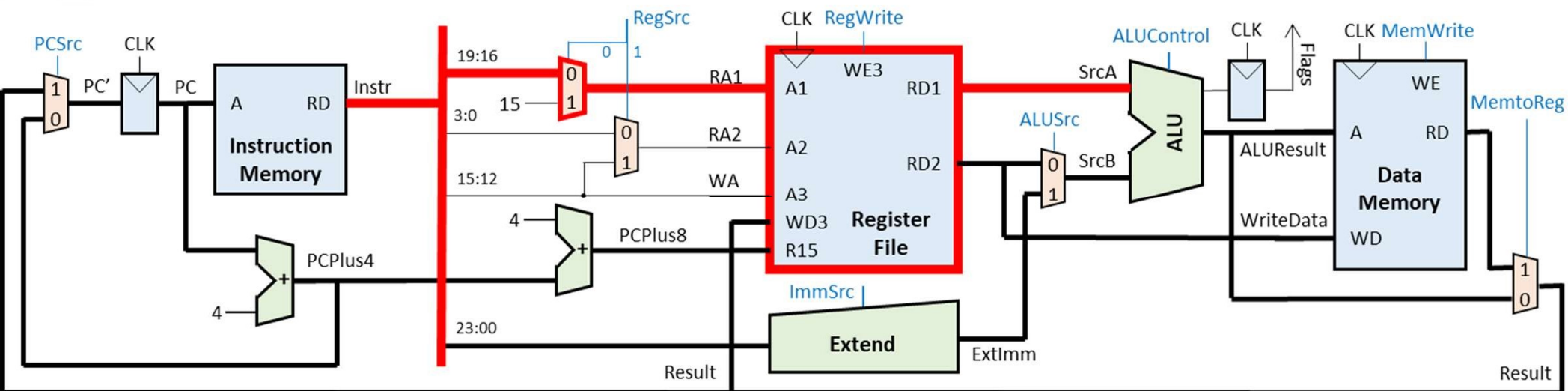
Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή STR

- Απαιτούμενες λειτουργίες της εντολής STR που ήδη υποστηρίζονται:
 - η διεύθυνση βάσης που είναι αποθηκευμένη στον καταχωρητή προέλευσης Rn του αρχείου καταχωρητών διαβάζεται στη θύρα ανάγνωσης $RD1$ (συμπεριλαμβάνεται και ο μετρητής προγράμματος PC)
 - επέκταση μηδενός στα 32 bit του άμεσου τελεστέου των 12 bit
 - η σχετική απόσταση προστίθεται στην ή αφαιρείται από τη διεύθυνση βάσης, ώστε να προκύψει η διεύθυνση της μνήμης δεδομένων
- Νέες λειτουργίες για την εντολή STR:
 - η λέξη δεδομένων που είναι αποθηκευμένη στον δεύτερο καταχωρητή προέλευσης Rd του αρχείου καταχωρητών διαβάζεται στη θύρα ανάγνωσης $RD2$ (συμπεριλαμβάνεται και ο μετρητής προγράμματος PC)
 - σύγχρονη εγγραφή της λέξης δεδομένων στη μνήμη δεδομένων
 - δεν υπάρχει εγγραφή στο αρχείο καταχωρητών ($RegWrite = 0$)
 - ο πολυπλέκτης επιλογής διεύθυνσης επόμενης εντολής επιλέγει αποκλειστικά τη διεύθυνση επόμενης εντολής: $PC' = PC + 4$ ($PCSrc = 0$)

Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή STR

Ανάγνωση του αρχείου καταχωρητών (R0-R14) από το RD1

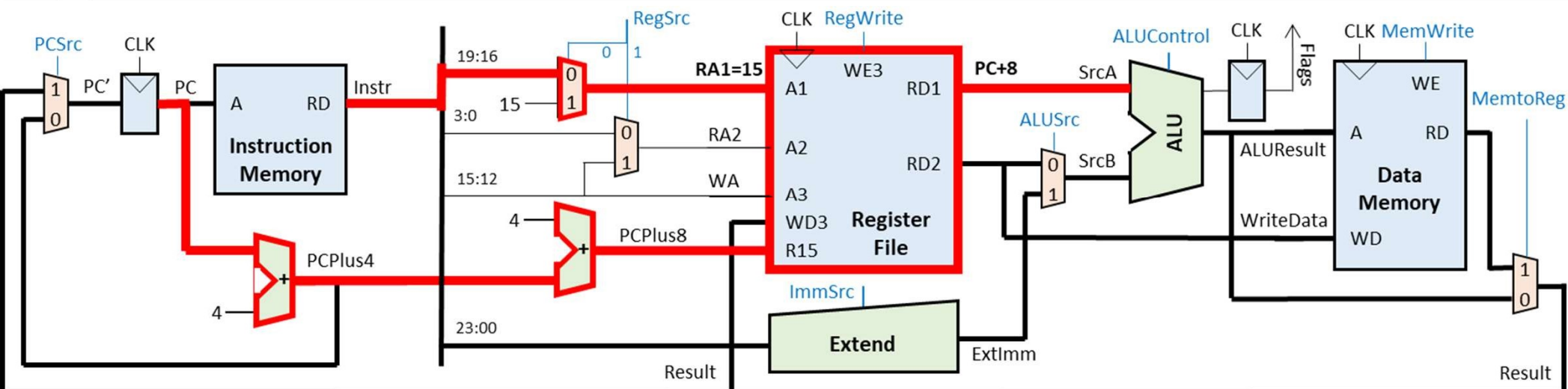
- Η διεύθυνση βάσης που είναι αποθηκευμένη στον **καταχωρητή προέλευσης Rn** του αρχείου καταχωρητών (R0-R14) διαβάζεται στη θύρα ανάγνωσης RD1
 - σύνδεση του πεδίου Rn της εντολής ($Instr_{19:16}$) στην είσοδο διευθύνσεων ανάγνωσης A1
 - ανάγνωση περιεχομένου καταχωρητή από τη θύρα ανάγνωσης RD1
 - σήματα ελέγχου: **RegScr[1:0] = X0** και **RegWrite = 0** (ανάγνωση αρχείου καταχ.)



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή STR

Ανάγνωση του μετρητή προγράμματος R15 (PC + 8) από το RD1

- Η διεύθυνση βάσης, που είναι η διεύθυνση της τρέχουσας εντολής και είναι αποθηκευμένη στον **μετρητή προγράμματος PC** ($R_n = 15$), διαβάζεται στη θύρα ανάγνωσης RD1
 - ορίζεται ο μετρητής προγράμματος PC ως **καταχωρητής προέλευσης R_n** της εντολής
 - η ανάγνωση του R15 του αρχείου καταχωρητών επιστρέφει πάντα την τιμή **PC + 8**
 - σύνδεση του πεδίου R_n της εντολής ($Instr_{19:16}$) στην είσοδο διευθύνσεων ανάγνωσης A1
 - σύνδεση της εξόδου PC του μετρητή προγράμματος με την είσοδο R15 του αρχείου καταχωρητών μέσω δύο αθροιστών κατά 4, ώστε να υπολογιστεί το PC+8
 - απαίτηση για έναν επιπλέον αθροιστή κατά 4 με είσοδο PCPlus4 και έξοδο PCPlus8
 - μεταφορά της εισόδου R15 στη θύρα ανάγνωσης RD1 (μέσω εσωτερικού πολυπλέκτη)
 - σήματα ελέγχου: **RegScr[1:0] = X0** και **RegWrite = 0** (ανάγνωση αρχείου καταχωρητών)

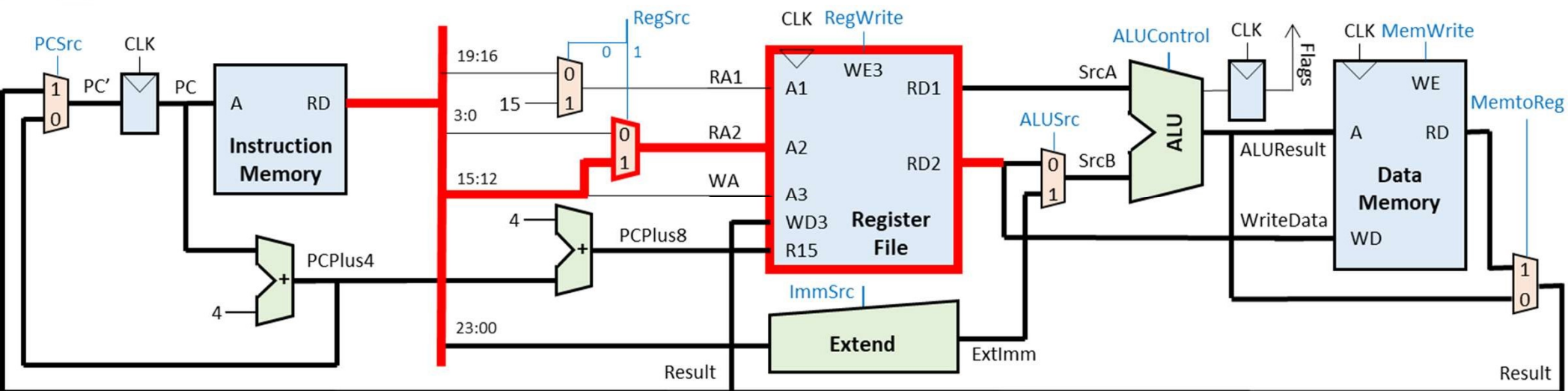


Χρησιμοποιείται όταν είναι κοινή η μνήμη εντολών και δεδομένων

Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή STR

Ανάγνωση του αρχείου καταχωρητών (R0-R14) από το RD2

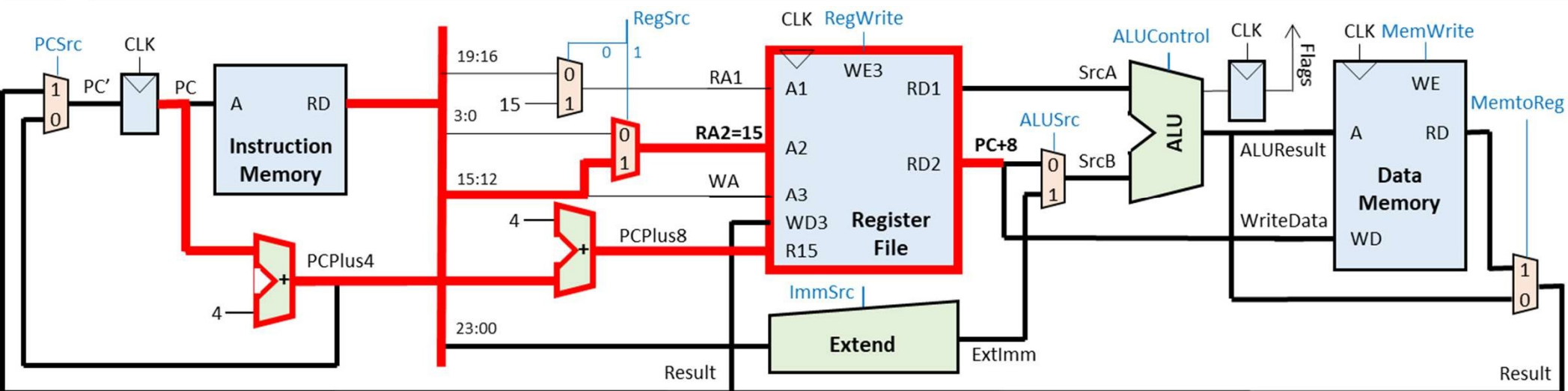
- Η διεύθυνση βάσης που είναι αποθηκευμένη στον **καταχωρητή προέλευσης Rn** του αρχείου καταχωρητών (R0-R14) διαβάζεται στη θύρα ανάγνωσης RD2
 - σύνδεση του πεδίου *Rd* της εντολής ($Instr_{15:12}$) στην είσοδο διευθύνσεων ανάγνωσης A2
 - ανάγνωση περιεχομένου καταχωρητή από τη θύρα ανάγνωσης RD2
 - σήματα ελέγχου: **RegScr[1:0] = 1X** και **RegWrite = 0** (ανάγνωση αρχείου καταχ.)



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή STR

Ανάγνωση του μετρητή προγράμματος R15 (PC + 8) από το RD2

- Η διεύθυνση βάσης, που είναι η διεύθυνση της τρέχουσας εντολής και είναι αποθηκευμένη στον **μετρητή προγράμματος PC** ($R_n = 15$), διαβάζεται στη θύρα ανάγνωσης RD2
 - ορίζεται ο μετρητής προγράμματος PC ως **καταχωρητής προέλευσης Rd** της εντολής
 - η ανάγνωση του R15 του αρχείου καταχωρητών επιστρέφει πάντα την τιμή **PC + 8**
 - σύνδεση του πεδίου Rd της εντολής ($Instr_{15:12}$) στην είσοδο διευθύνσεων ανάγνωσης A2
 - σύνδεση της εξόδου PC του μετρητή προγράμματος με την είσοδο R15 του αρχείου καταχωρητών μέσω δύο αθροιστών κατά 4, ώστε να υπολογιστεί το PC+8
 - απαίτηση για έναν επιπλέον αθροιστή κατά 4 με είσοδο PCPlus4 και έξοδο PCPlus8
 - μεταφορά της εισόδου R15 στη θύρα ανάγνωσης RD2 (μέσω εσωτερικού πολυπλέκτη)
 - σήματα ελέγχου: **RegScr[1:0] = 1X** και **RegWrite = 0** (ανάγνωση αρχείου καταχωρητών)

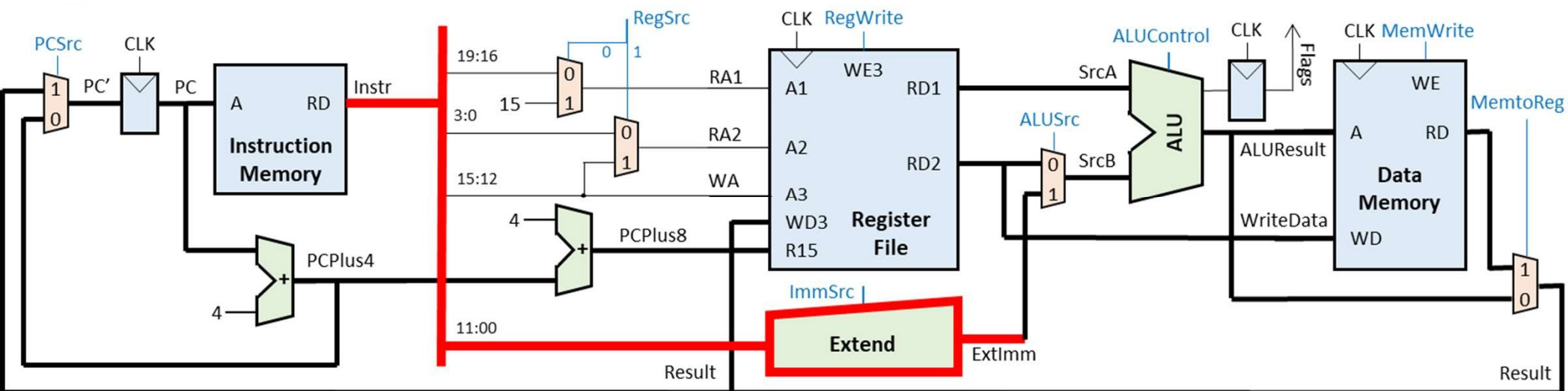


Χρησιμοποιείται όταν είναι κοινή η μνήμη εντολών και δεδομένων

Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή STR

Επέκταση πρόσημου ή μηδενός

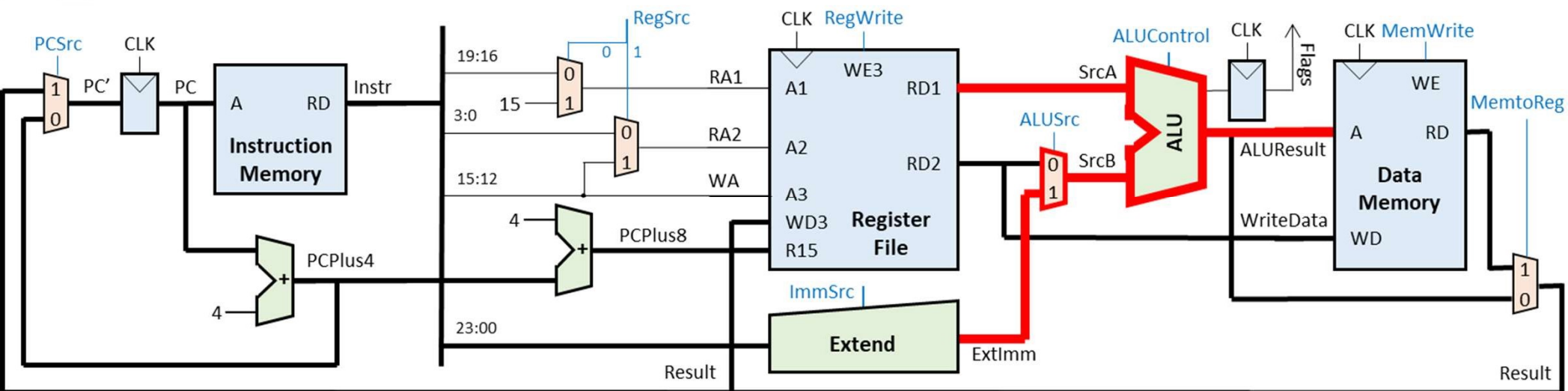
- Η εντολή LDR απαιτεί και μια **σχετική απόσταση** για τον υπολογισμό της διεύθυνσης δεδομένων:
 - σύνδεση του πεδίου άμεσου τελεστέου ($imm12$) της εντολής ($Instr_{11:0}$) στην είσοδο της μονάδας *Extend*
 - στη μονάδα *Extend* εκτελείται **επέκταση μηδενός** στα 32 bit του **μη προσημασμένου άμεσου τελεστέου των 12 bit** και το αποτέλεσμα εμφανίζεται στην έξοδο *ExtImm*
 - σήματα ελέγχου: **ImmScr = 0**



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή STR

Υπολογισμός διεύθυνσης της μνήμης δεδομένων (DM)

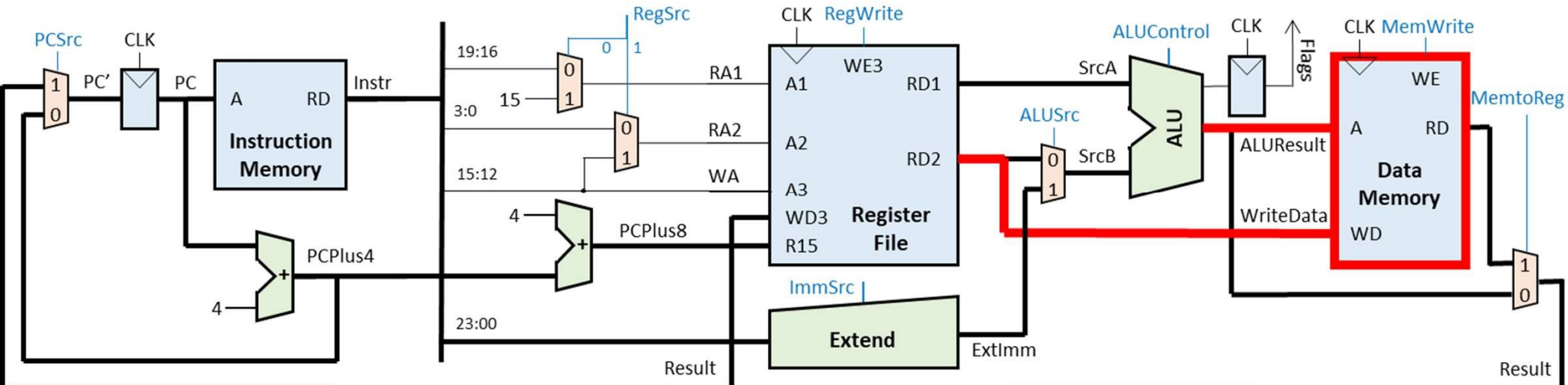
- Η **σχετική απόσταση προστίθεται** στην ή **αφαιρείται** από τη διεύθυνση βάσης στη μονάδα ALU, ώστε να προκύψει η διεύθυνση της μνήμης δεδομένων
 - σύνδεση της θύρας *RD1* του αρχείου καταχωρητών με την είσοδο *SrcA* των 32 bit της μονάδας ALU
 - σύνδεση της εξόδου *ExtImm* της μονάδας *Extend* με την είσοδο *SrcB* των 32 bit της μονάδας ALU
 - εκτελείται πρόσθεση ή αφαίρεση και το αποτέλεσμα της πράξης **$Rn \pm \#imm12$** εμφανίζεται στην έξοδο *ALUResult* των 32 bit της μονάδας ALU
 - σήματα ελέγχου: **$ALUSrc = 1$** και **$ALUControl[1:0] = 00 (+)$ ή $01 (-)$**



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή STR

Εγγραφή μνήμης δεδομένων

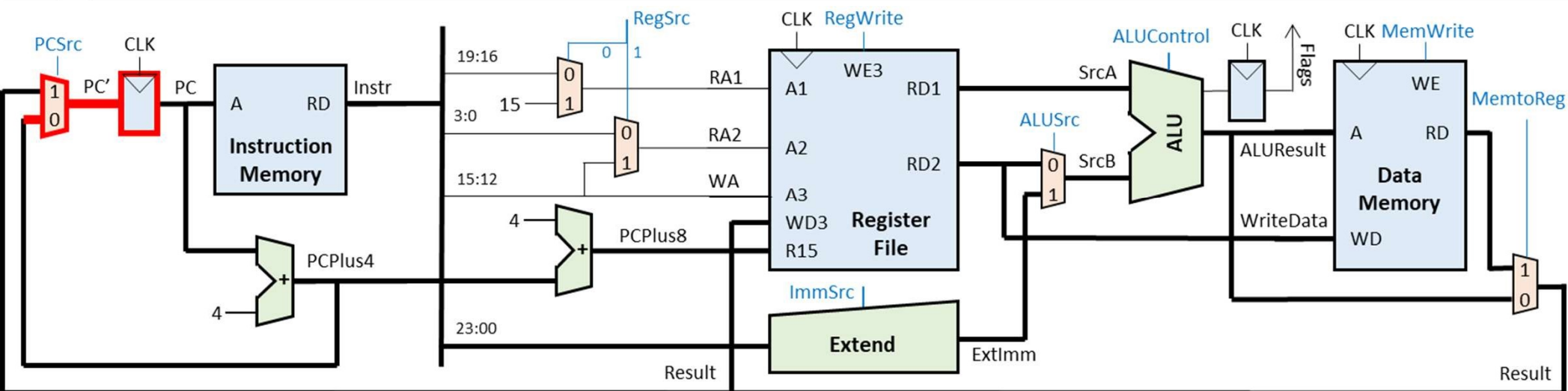
- Η διεύθυνση που εμφανίζεται στην έξοδο ALUResult της μονάδας ALU διευθυνσιοδοτεί την είσοδο διευθύνσεων A της μνήμης δεδομένων (DM)
- Η λέξη δεδομένων (WriteData), που εμφανίζεται στη θύρα ανάγνωσης RD2, γράφεται σύγχρονα στη μνήμη δεδομένων DM των $2^M \times 32$ bit μέσω της θύρας εγγραφής δεδομένων WD της DM
 - η διεύθυνση της λέξης δεδομένων είναι ευθυγραμμισμένη και αφορά μνήμες οργανωμένες σε *byte*, ενώ η μνήμη δεδομένων DM είναι οργανωμένη σε λέξεις
 - ισχύει $A[M-1:0] = ALUResult[M+1:2]$
 - σήματα ελέγχου: **MemWrite = 1** (εγγραφή μνήμης δεδομένων)



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή STR

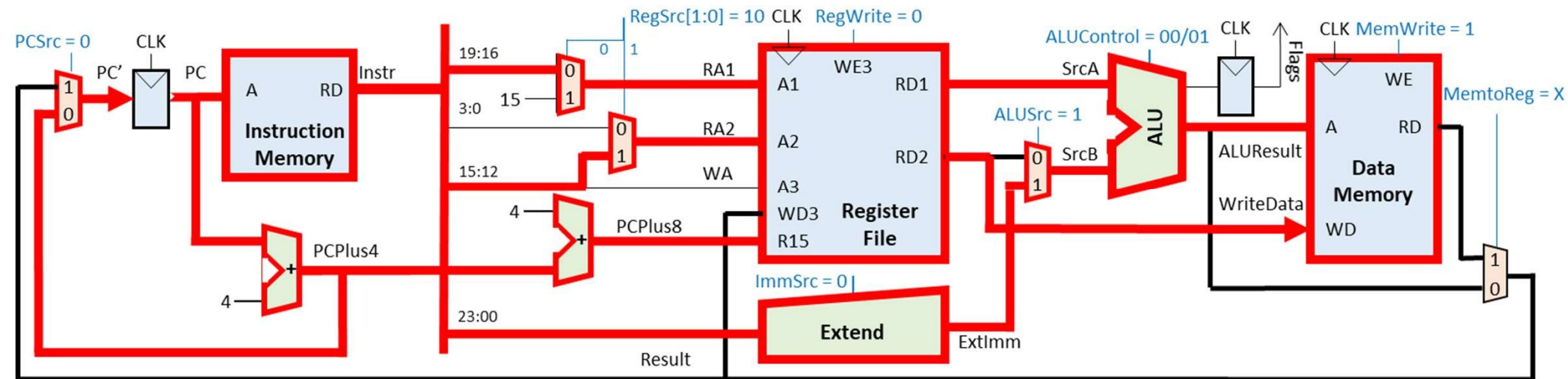
Επιλογή της διεύθυνσης της επόμενης εντολής που θα εκτελεσθεί

- Ο πολυπλέκτης επιλογής διεύθυνσης επόμενης εντολής επιλέγει αποκλειστικά τη διεύθυνση: $PC' = PC + 4$
- Η νέα διεύθυνση αποθηκεύεται στον μετρητή προγράμματος στην επόμενη ανερχόμενη ακμή του CLK
 - σύνδεση της εισόδου 0 του πολυπλέκτη με την έξοδο PCPlus4 του αθροιστή κατά 4
 - σύνδεση της εξόδου του πολυπλέκτη με την είσοδο PC' του μετρητή προγράμματος
 - σήματα ελέγχου: $PCSrc = 0$ (PCPlus4), $MemtoReg = X$

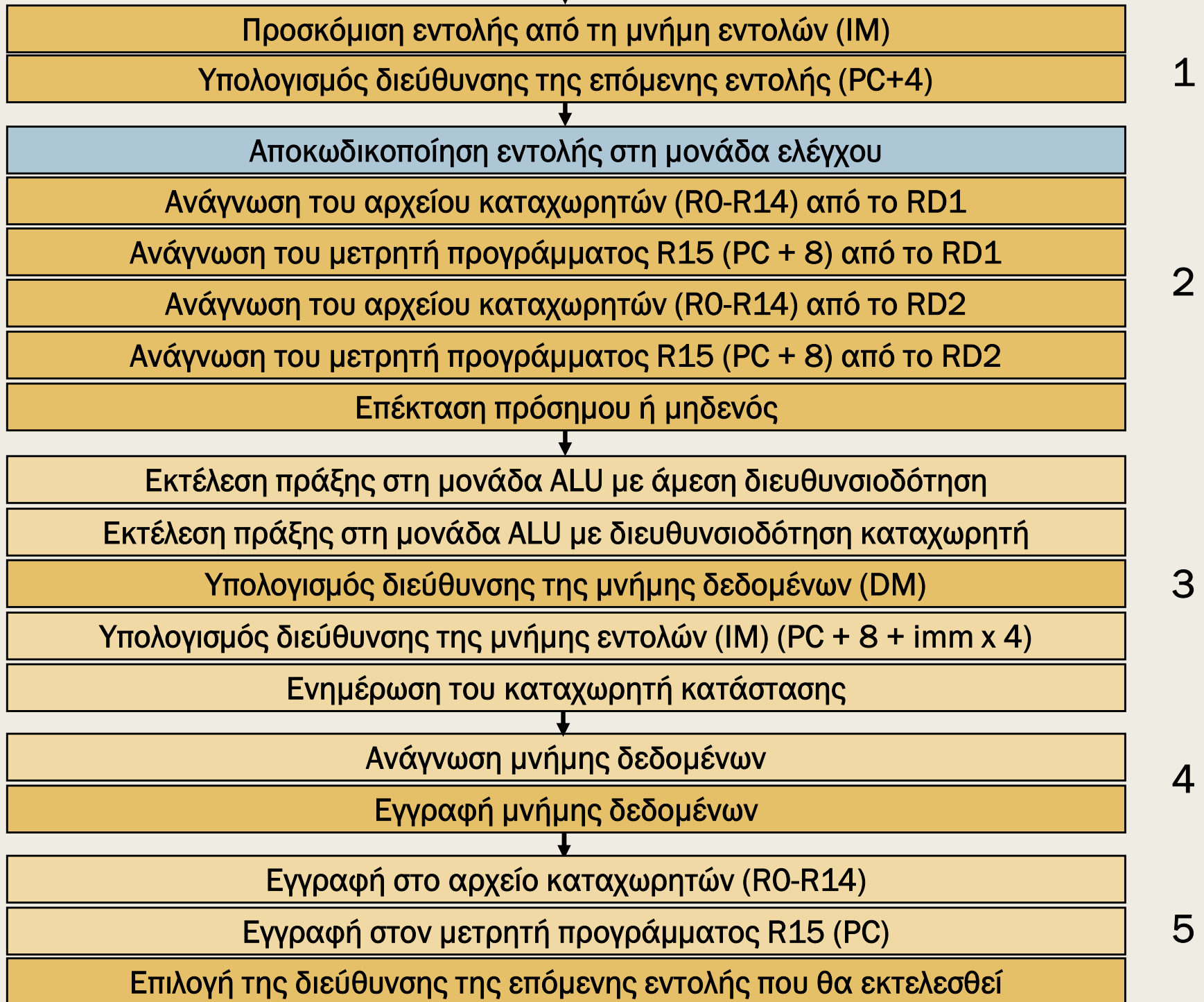


Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή STR

- Συνολική ροή δεδομένων και τιμές στα σήματα ελέγχου:

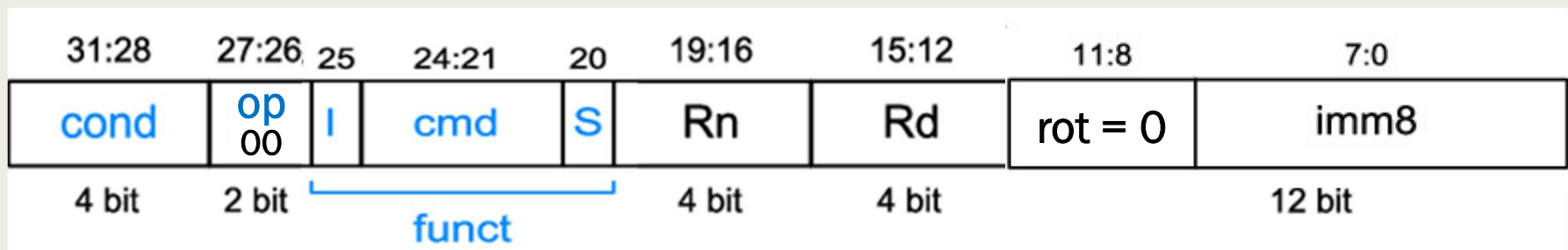


Λειτουργίες εντολής STR



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-I

- Στη συνέχεια, θα μελετήσουμε τη διαδρομή δεδομένων που υλοποιεί τις εντολές επεξεργασίας δεδομένων (ADD(S), SUB(S), AND(S) και ORR(S)) με άμεση διευθυνσιοδότηση
- Κώδικας συμβολικής γλώσσας των εντολών ALU(S)-I:
 - $ALU(S) Rd, Rn, \#imm8 ; Rd = Rn +/-/and/or imm8$
- Στη μορφή των εντολών ορίζονται:
 - οι διευθύνσεις του καταχωρητή προέλευσης Rn και του καταχωρητή προορισμού Rd
 - ένας μη προσημασμένος άμεσος τελεστής των 8 bit
 - δεν υποστηρίζεται δεξιά περιστροφή ($rot = 0$)
- Απαιτείται η εκτέλεση των πράξεων της πρόσθεσης, της αφαίρεσης, του λογικού AND και του λογικού OR στη μονάδα ALU



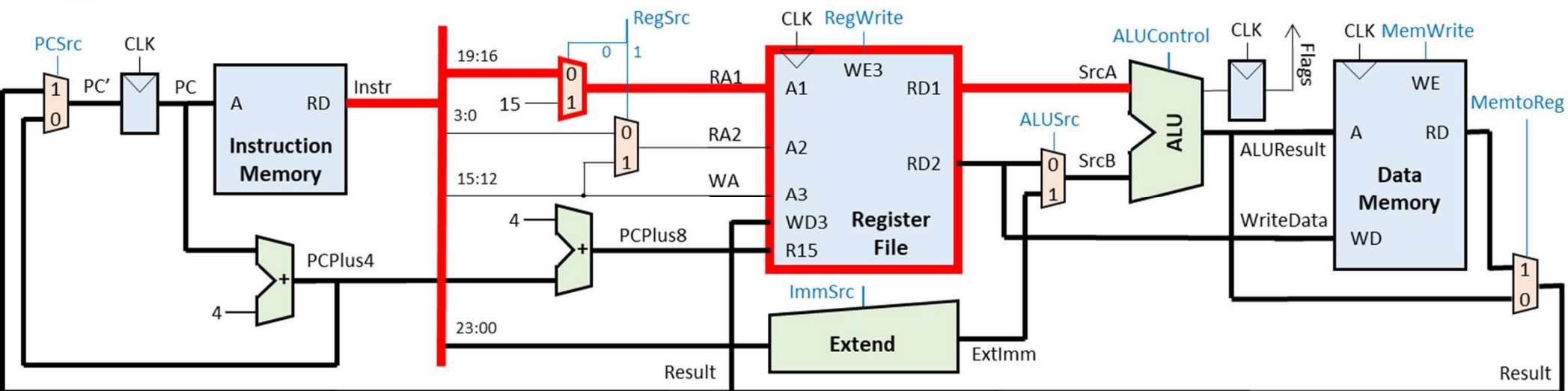
Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-I

- Απαιτούμενες λειτουργίες των εντολών ALU-I που ήδη υποστηρίζονται:
 - ο τελεστέος που είναι αποθηκευμένος στον καταχωρητή προέλευσης Rn του αρχείου καταχωρητών διαβάζεται στη θύρα ανάγνωσης $RD1$ (συμπεριλαμβάνεται και ο μετρητής προγράμματος PC)
 - επέκταση μηδενός στα 32 bit του άμεσου τελεστέου των 12 bit
 - Αφού $rot = 0$, θεωρούμε έναν άμεσο τελεστέο των 12 bit, αντί των 8 bit
- Νέες λειτουργίες για τις εντολές ALU-I:
 - η μονάδα ALU πρέπει να υποστηρίζει πέραν των αριθμητικών πράξεων (της πρόσθεσης και της αφαίρεσης) και τις λογικές πράξεις AND και OR
 - Το αποτέλεσμα των πράξεων στη μονάδα ALU ($ALUResult$) γράφεται απευθείας στον καταχωρητή προορισμού Rd του αρχείου καταχωρητών
 - Εάν το πεδίο $S = 1$, τότε ενημερώνεται ο καταχωρητής κατάστασης με τις σημαίες συνθήκης N, Z, C, V ($C = 0$ και $V = 0$ στις λογικές πράξεις) στην επόμενη ακμή του CLK
 - ο πολυπλέκτης επιλογής διεύθυνσης επόμενης εντολής επιλέγει ανάμεσα σε δύο πιθανές διευθύνσεις: $PC' = PC + 4$ ή $PC' = Rn +/- /and/or imm8$

Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-I

Ανάγνωση του αρχείου καταχωρητών (R0-R14) από το RD1

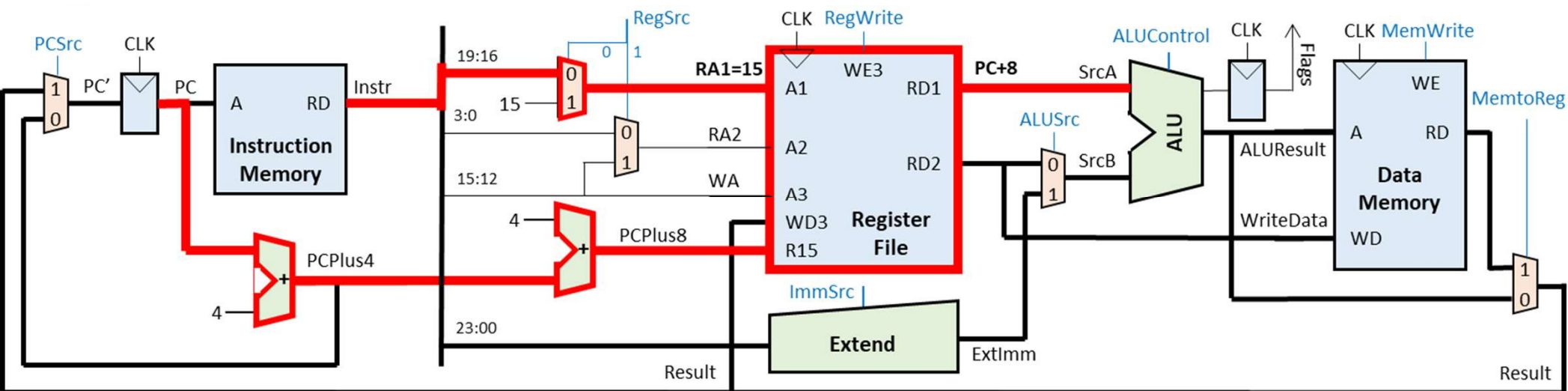
- ο τελεστής που είναι αποθηκευμένος στον **καταχωρητή προέλευσης Rn** του αρχείου καταχωρητών (R0-R14) διαβάζεται στη θύρα ανάγνωσης RD1
 - σύνδεση του πεδίου Rn της εντολής ($Instr_{19:16}$) στην είσοδο διευθύνσεων ανάγνωσης A1
 - ανάγνωση περιεχομένου καταχωρητή από τη θύρα ανάγνωσης RD1
 - σήματα ελέγχου: **RegScr[1:0] = X0** και **RegWrite = 0** (ανάγνωση αρχείου καταχ.)



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-I

Ανάγνωση του μετρητή προγράμματος R15 (PC + 8) από το RD1

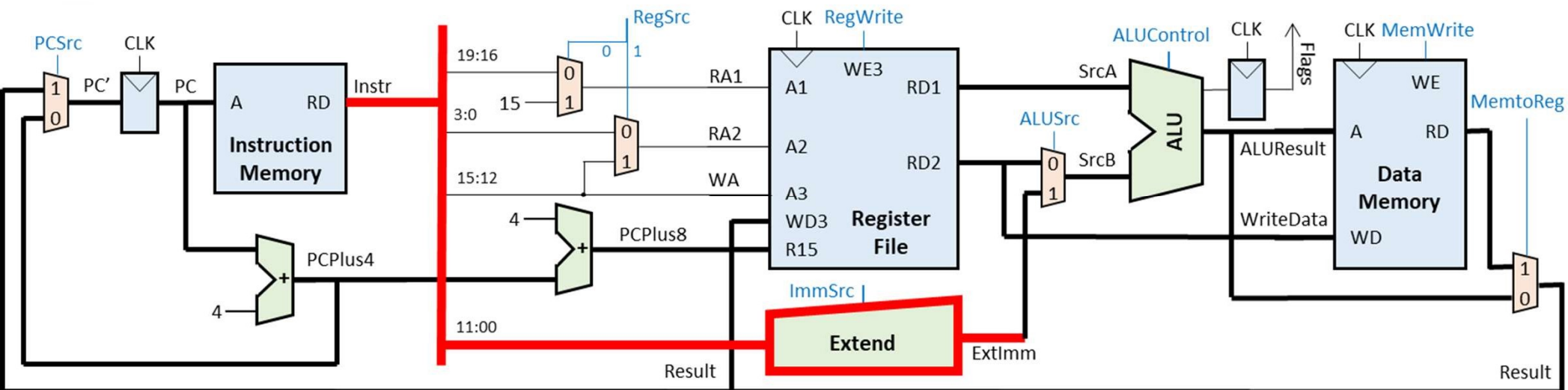
- ο τελεστής, που είναι η διεύθυνση της τρέχουσας εντολής και είναι αποθηκευμένος στον **μετρητή προγράμματος PC** ($R_n = 15$), διαβάζεται στη θύρα ανάγνωσης RD1
 - ορίζεται ο μετρητής προγράμματος PC ως **καταχωρητής προέλευσης R_n** της εντολής
 - η ανάγνωση του R15 του αρχείου καταχωρητών επιστρέφει πάντα την τιμή **PC + 8**
 - σύνδεση του πεδίου R_n της εντολής ($Instr_{19:16}$) στην είσοδο διευθύνσεων ανάγνωσης A1
 - σύνδεση της εξόδου PC του μετρητή προγράμματος με την είσοδο R15 του αρχείου καταχωρητών μέσω δύο αθροιστών κατά 4, ώστε να υπολογιστεί το PC+8
 - απαίτηση για έναν επιπλέον αθροιστή κατά 4 με είσοδο PCPlus4 και έξοδο PCPlus8
 - μεταφορά της εισόδου R15 στη θύρα ανάγνωσης RD1 (μέσω εσωτερικού πολυπλέκτη)
 - σήματα ελέγχου: **RegScr[1:0] = X0** και **RegWrite = 0** (ανάγνωση αρχείου καταχ.)



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-I

Επέκταση πρόσημου ή μηδενός

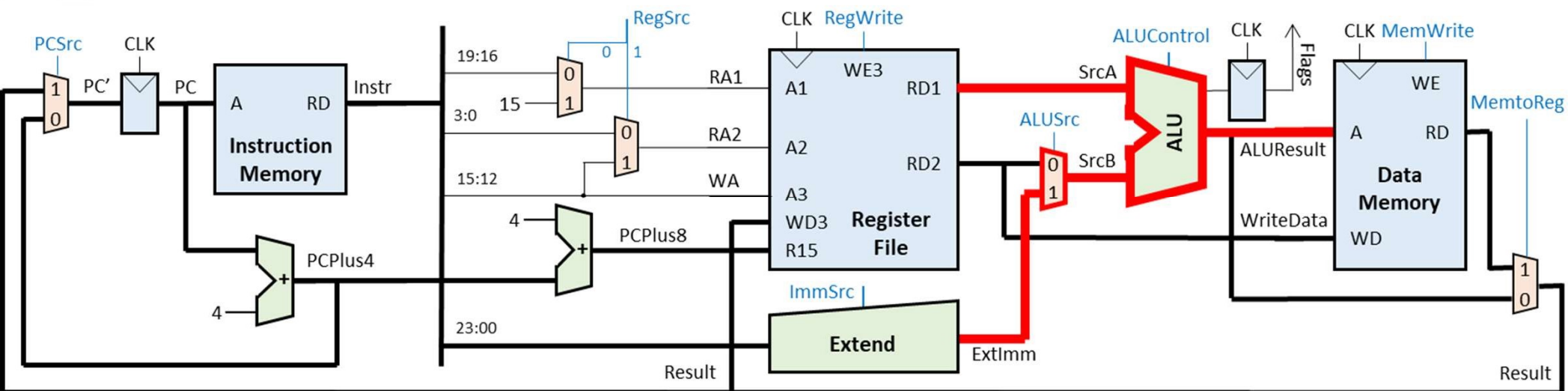
- Η εντολές ALU με άμεση διευθυνσιοδότηση απαιτούν ο άμεσος τελεστής `imm8` να επεκταθεί στα 32 bit:
 - σύνδεση του πεδίου άμεσου τελεστή (`imm12`) της εντολής (`Instr11:0`) στην είσοδο της μονάδας `Extend`
 - Αφού `rot = 0`, θεωρούμε έναν άμεσο τελεστή των 12 bit, αντί των 8 bit
 - στη μονάδα `Extend` εκτελείται **επέκταση μηδενός** στα 32 bit του **μη προσημασμένου άμεσου τελεστή των 12 bit** και το αποτέλεσμα εμφανίζεται στην έξοδο `ExtImm`
 - σήματα ελέγχου: **ImmSrc = 0**



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-I

Εκτέλεση πράξης στη μονάδα ALU με άμεση διευθυνσιοδότηση

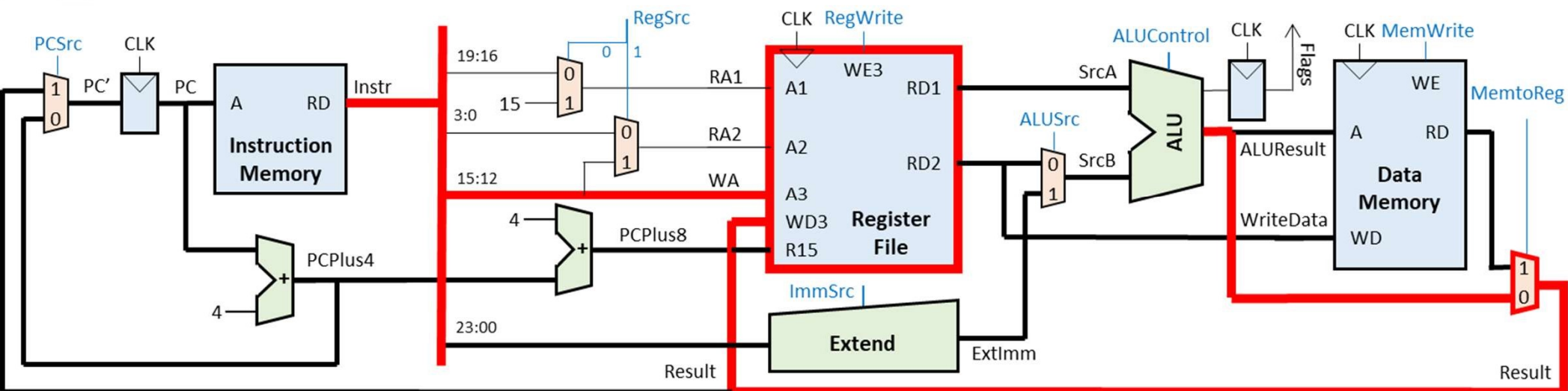
- η μονάδα ALU εκτελεί πέραν των αριθμητικών πράξεων (της πρόσθεσης και της αφαίρεσης) και τις λογικές πράξεις AND και OR
 - σύνδεση της θύρας *RD1* του αρχείου καταχωρητών με την είσοδο *SrcA* των 32 bit της μονάδας ALU
 - σύνδεση της εξόδου *ExtImm* της μονάδας *Extend* με την είσοδο *SrcB* των 32 bit της μονάδας ALU
 - εκτελείται αριθμητική ή λογική πράξη και το αποτέλεσμα της πράξης *Rn +/-/and/or imm8* εμφανίζεται στην έξοδο *ALUResult* των 32 bit της μονάδας ALU
 - σήματα ελέγχου: **ALUSrc = 1** και **ALUControl[1:0] = 00 (+), 01 (-), 10 (AND), 11 (OR)**



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-I

Εγγραφή στο αρχείο καταχωρητών (R0-R14)

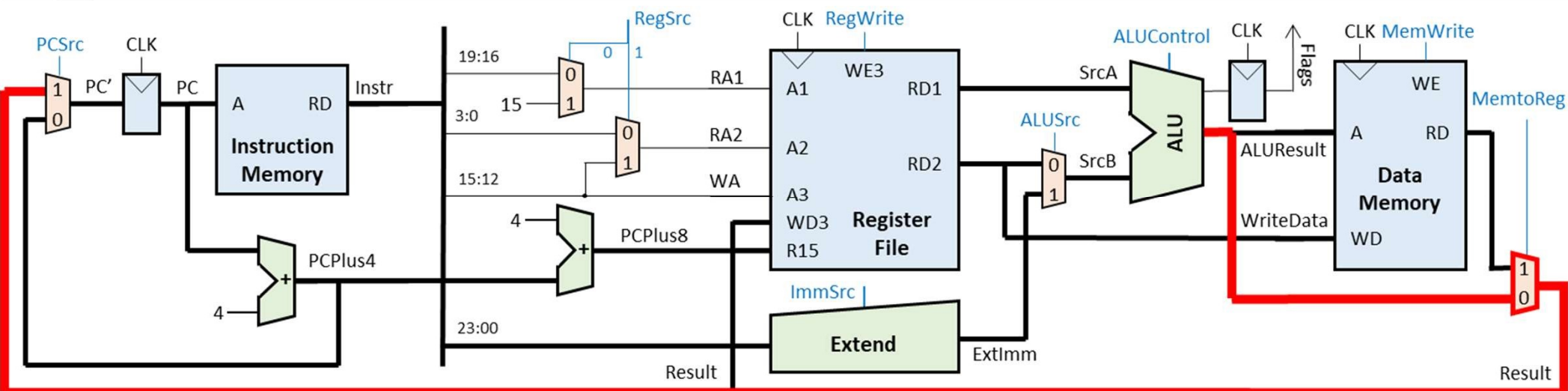
- Το αποτέλεσμα της πράξης στη μονάδα ALU (ALUResult) αποθηκεύεται στον **καταχωρητή προορισμού Rd** του αρχείου καταχωρητών
 - σύνδεση του πεδίου *Rd* της εντολής ($Instr_{15:12}$) στην είσοδο διευθύνσεων εγγραφής *A3*
 - σύνδεση της εξόδου *ALUResult* της μονάδας *ALU* με τη θύρα εγγραφής *WD3* του αρχείου καταχωρητών
 - εγγραφή περιεχομένου καταχωρητή από τη θύρα εγγραφής *WD3*
 - σήματα ελέγχου: **MemtoReg = 0** και **RegWrite = 1** (εγγραφή αρχείου καταχωρητών)



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-I

Εγγραφή στον μετρητή προγράμματος PC (R15)

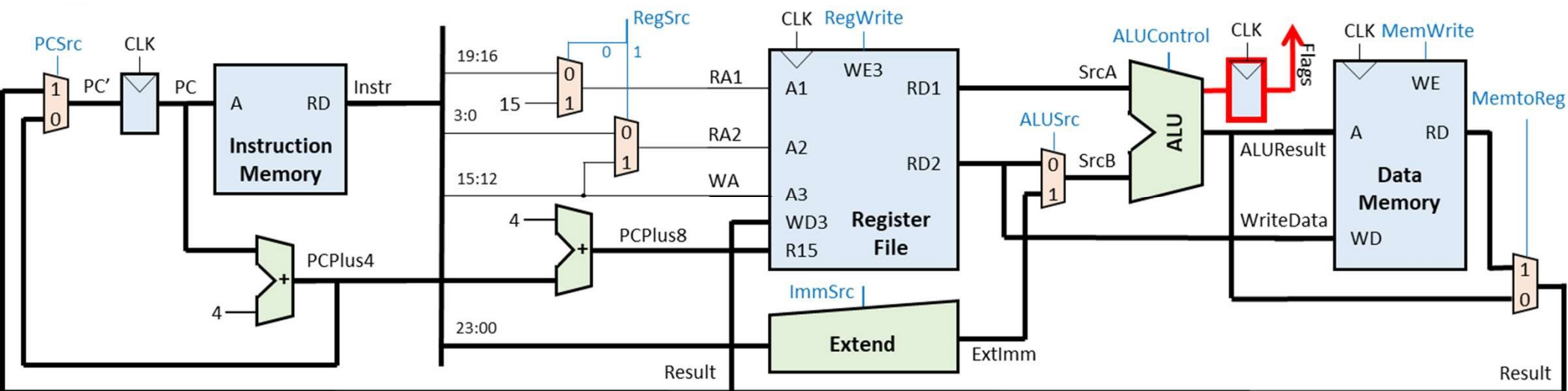
- Το αποτέλεσμα της πράξης στη μονάδα ALU (ALUResult) αποθηκεύεται στον **μετρητή προγράμματος PC (Rd=15)** του αρχείου καταχωρητών
 - ορίζεται ο μετρητής προγράμματος PC ως **καταχωρητής προορισμού Rd** της εντολής ALU(S)-I
 - το αποτέλεσμα της πράξης στην ALU είναι **διεύθυνση της μνήμης εντολών**
 - σύνδεση της εξόδου ALUResult της μονάδας ALU με τον πολυπλέκτη επιλογής διεύθυνσης επόμενης εντολής
 - σήματα ελέγχου: **MemtoReg = 0, RegWrite = 0**



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-I

Ενημέρωση του καταχωρητή κατάστασης

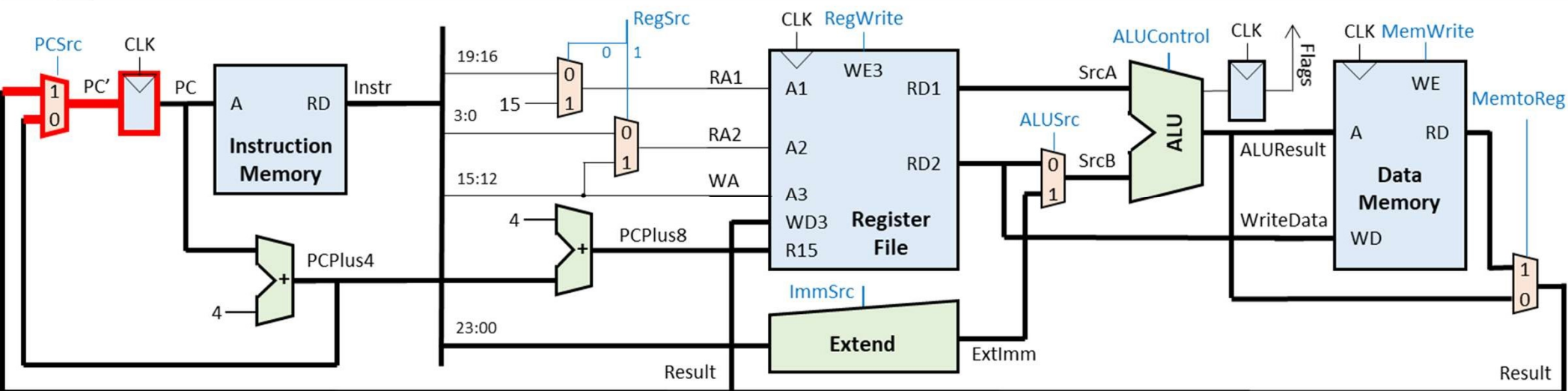
- Εάν το πεδίο S της εντολής είναι 1, τότε ενημερώνεται ο καταχωρητής κατάστασης με τις νέες σημαίες συνθήκης N, Z, C, V (C = 0 και V = 0 στις λογικές πράξεις) στην επόμενη ακμή του CLK
 - το πεδίο S της εντολής ($Instr_{20}$) συμμετέχει στη δημιουργία του σήματος έγκρισης εγγραφής (*FlagsWrite*) του καταχωρητή κατάστασης για εντολές επεξεργασίας δεδομένων ($op = 00$) (το *FlagsWrite* δεν φαίνεται στο σχήμα)
- υλοποιείται στη μονάδα ελέγχου



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-I

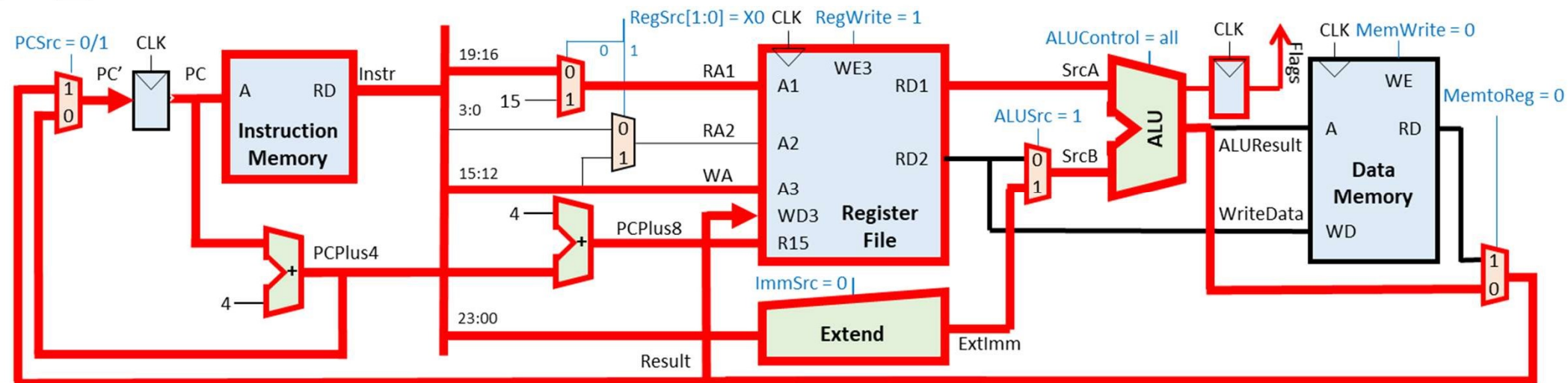
Επιλογή της διεύθυνσης της επόμενης εντολής που θα εκτελεσθεί

- Ο πολυπλέκτης επιλογής διεύθυνσης επόμενης εντολής επιλέγει ανάμεσα σε δύο πιθανές διευθύνσεις: $PC' = PC + 4$ ή $PC' = Rn +/-/and/or imm8$
- Η νέα διεύθυνση αποθηκεύεται στον μετρητή προγράμματος στην επόμενη ανερχόμενη ακμή του CLK
 - σύνδεση της εισόδου 0 του πολυπλέκτη με την έξοδο PCPlus4 του αθροιστή κατά 4
 - σύνδεση της εισόδου 1 του πολυπλέκτη με την έξοδο ALUResult της μονάδας ALU
 - σύνδεση της εξόδου του πολυπλέκτη με την είσοδο PC' του μετρητή προγράμματος
 - σήματα ελέγχου: $PCSrc = 1$ (ALUResult) ή 0 (PCPlus4), $MemtoReg = 0$

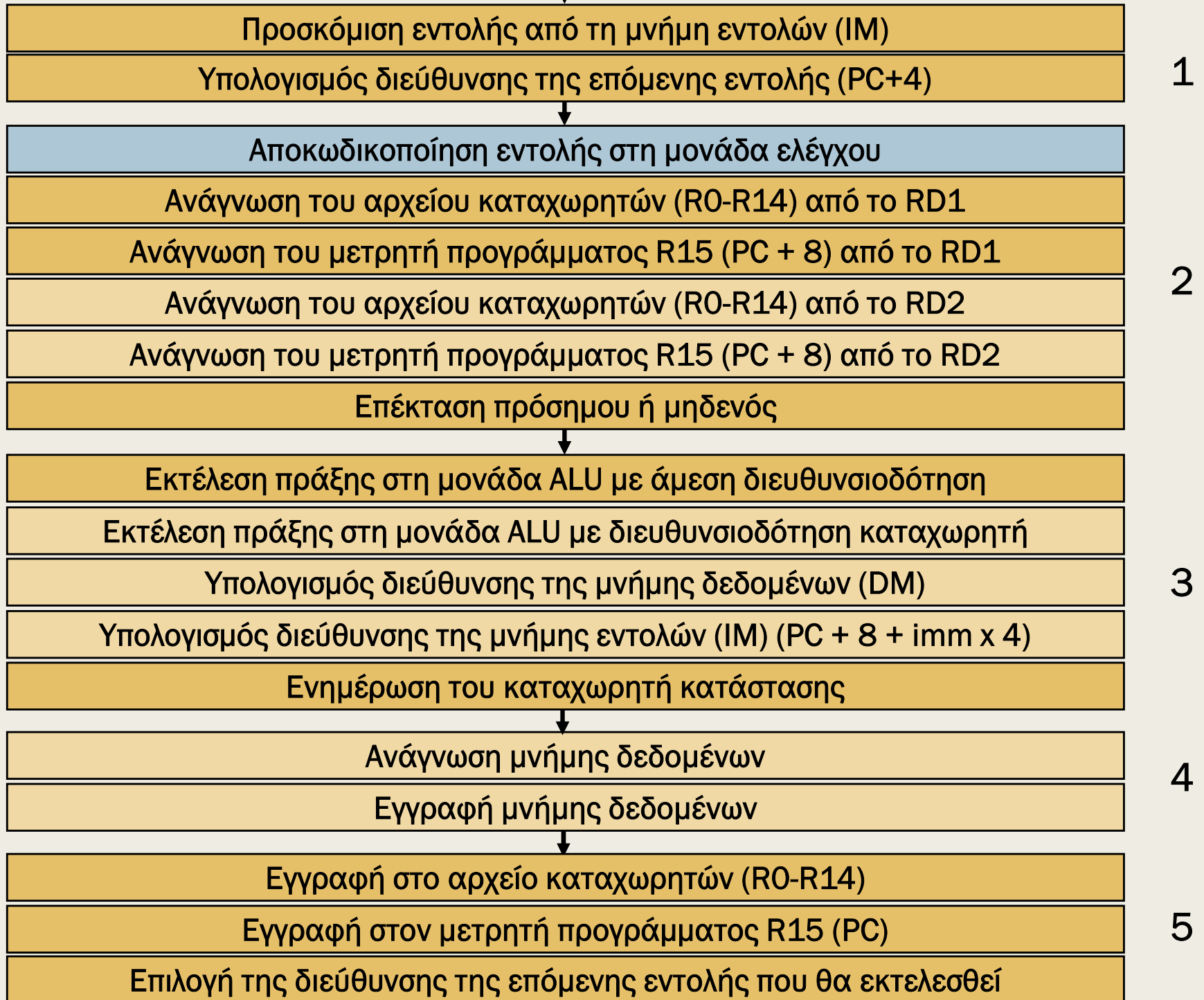


Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-I

- Συνολική ροή δεδομένων και τιμές στα σήματα ελέγχου:

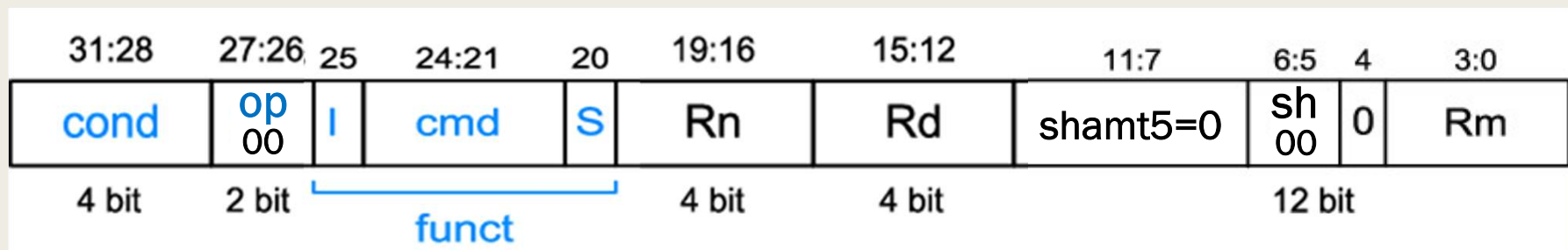


Λειτουργίες εντολών ALU(S)-I



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-R

- Στη συνέχεια, θα μελετήσουμε τη διαδρομή δεδομένων που υλοποιεί τις εντολές επεξεργασίας δεδομένων (ADD(S), SUB(S), AND(S) και ORR(S)) με διευθυνσιοδότηση καταχωρητή
- Κώδικας συμβολικής γλώσσας των εντολών ALU(S)-R:
 - $ALU(S) Rd, Rn, Rm; Rd = Rn +/-/and/or Rm$
- Στη μορφή των εντολών ορίζονται:
 - οι διευθύνσεις των καταχωρητών προέλευσης Rn και Rm , καθώς και του καταχωρητή προορισμού Rd
- Απαιτείται η εκτέλεση των πράξεων της πρόσθεσης, της αφαίρεσης, του λογικού AND και του λογικού OR στη μονάδα ALU



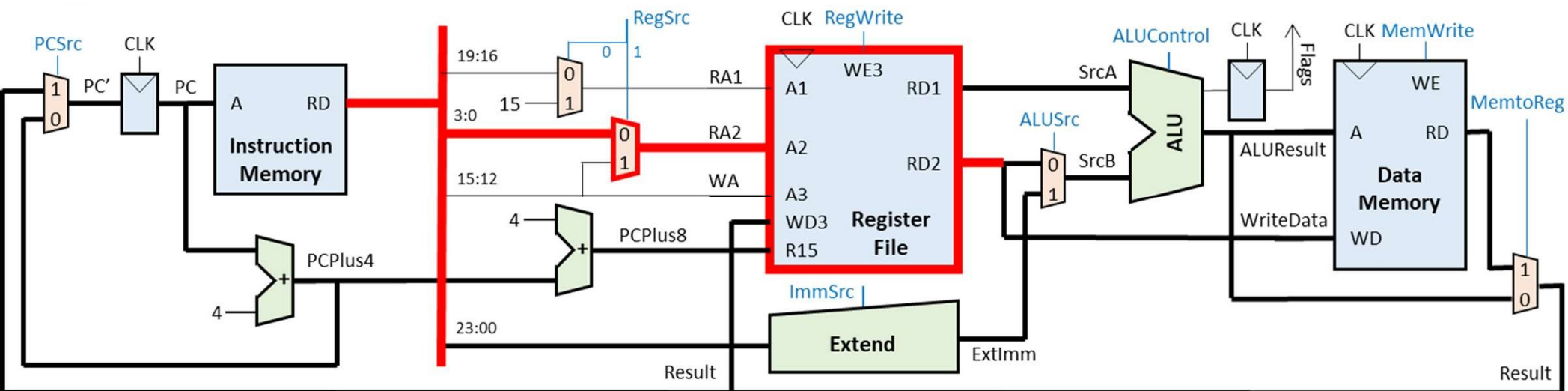
Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-R

- Απαιτούμενες λειτουργίες των εντολών ALU-R που ήδη υποστηρίζονται:
 - ο τελεστέος που είναι αποθηκευμένος στον καταχωρητή προέλευσης R_n του αρχείου καταχωρητών διαβάζεται στη θύρα ανάγνωσης $RD1$ (συμπεριλαμβάνεται και ο μετρητής προγράμματος PC)
 - η μονάδα ALU πρέπει να υποστηρίζει πέραν των αριθμητικών πράξεων (της πρόσθεσης και της αφαίρεσης) και τις λογικές πράξεις AND και OR
 - Το αποτέλεσμα των πράξεων στη μονάδα ALU (ALUResult) γράφεται απευθείας στον καταχωρητή προορισμού R_d του αρχείου καταχωρητών
 - Εάν το πεδίο $S = 1$, τότε ενημερώνεται ο καταχωρητής κατάστασης με τις σημαίες συνθήκης N, Z, C, V ($C = 0$ και $V = 0$ στις λογικές πράξεις) στην επόμενη ακμή του CLK
- Νέες λειτουργίες για τις εντολές ALU-R:
 - ο τελεστέος που είναι αποθηκευμένος στον καταχωρητή προέλευσης R_m του αρχείου καταχωρητών διαβάζεται στη θύρα ανάγνωσης $RD2$ (συμπεριλαμβάνεται και ο μετρητής προγράμματος PC)
 - ο πολυπλέκτης επιλογής διεύθυνσης επόμενης εντολής επιλέγει ανάμεσα σε δύο πιθανές διευθύνσεις: $PC' = PC + 4$ ή $PC' = R_n +/- /and/or R_m$

Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-R

Ανάγνωση του αρχείου καταχωρητών (R0-R14) από το RD2

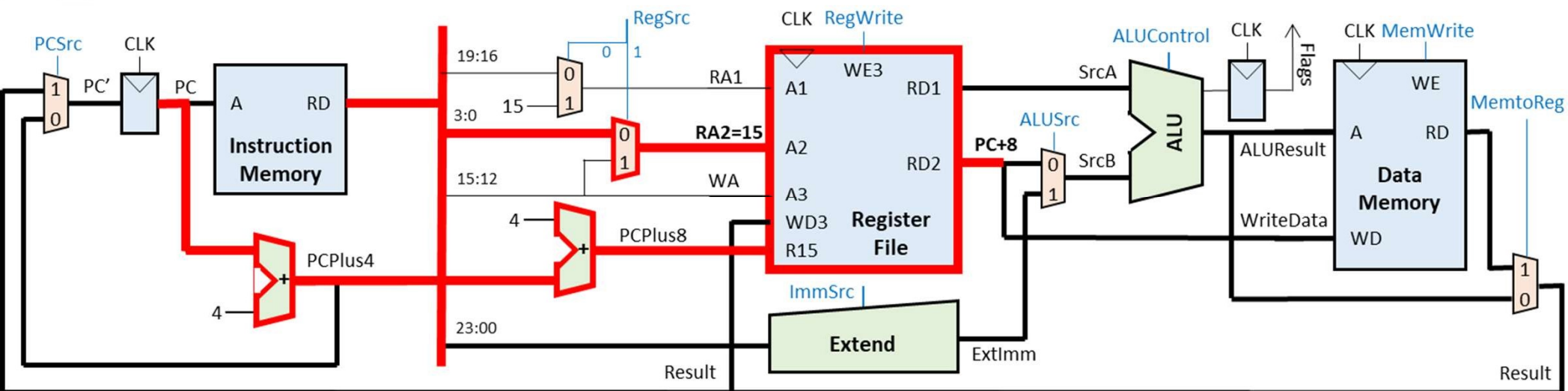
- ο τελεστής που είναι αποθηκευμένος στον **καταχωρητή προέλευσης Rm** του αρχείου καταχωρητών (R0-R14) διαβάζεται στη θύρα ανάγνωσης RD2
 - σύνδεση του πεδίου *Rm* της εντολής ($Instr_{3:0}$) στην είσοδο διευθύνσεων ανάγνωσης A2
 - ανάγνωση περιεχομένου καταχωρητή από τη θύρα ανάγνωσης RD2
 - σήματα ελέγχου: **RegScr[1:0] = 0X** και **RegWrite = 0** (ανάγνωση αρχείου καταχ.)



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-R

Ανάγνωση του μετρητή προγράμματος R15 (PC + 8) από το RD2

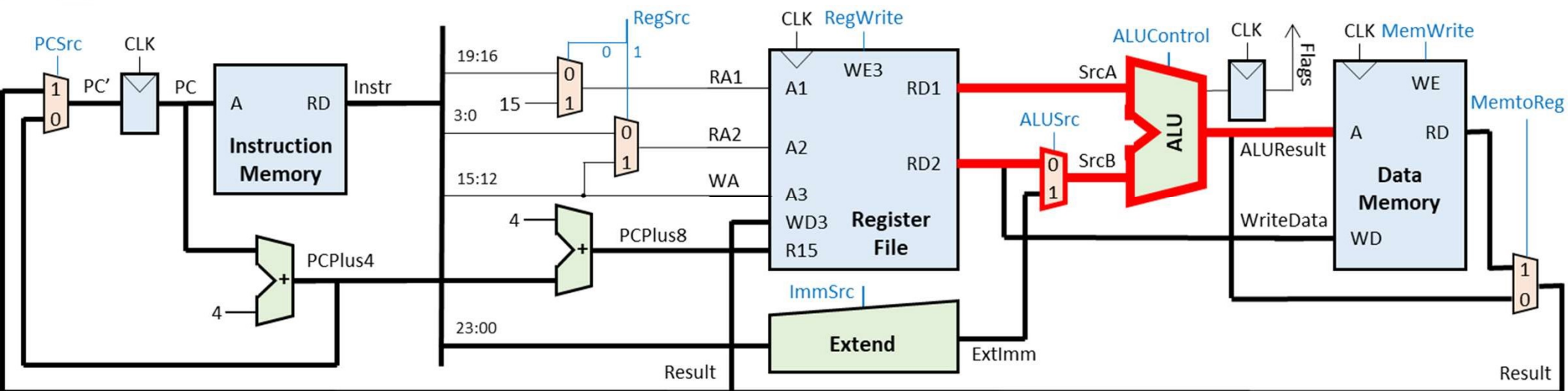
- Ο τελεστέος που είναι αποθηκευμένος στον **μετρητή προγράμματος PC** ($Rm = 15$), διαβάζεται στη θύρα ανάγνωσης RD2
 - ορίζεται ο μετρητής προγράμματος PC ως **καταχωρητής προέλευσης Rm** της εντολής
 - η ανάγνωση του R15 του αρχείου καταχωρητών επιστρέφει πάντα την τιμή **PC + 8**
 - σύνδεση του πεδίου Rm της εντολής ($Instr_{3:0}$) στην είσοδο διευθύνσεων ανάγνωσης A2
 - σύνδεση της εξόδου PC του μετρητή προγράμματος με την είσοδο R15 του αρχείου καταχωρητών μέσω δύο αθροιστών κατά 4, ώστε να υπολογιστεί το PC+8
 - απαίτηση για έναν επιπλέον αθροιστή κατά 4 με είσοδο PCPlus4 και έξοδο PCPlus8
 - μεταφορά της εισόδου R15 στη θύρα ανάγνωσης RD2 (μέσω εσωτερικού πολυπλέκτη)
 - σήματα ελέγχου: **RegScr[1:0] = 0X** και **RegWrite = 0** (ανάγνωση αρχείου καταχωρητών)



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-R

Εκτέλεση πράξης στη μονάδα ALU με διευθυνσιοδότηση καταχωρητή

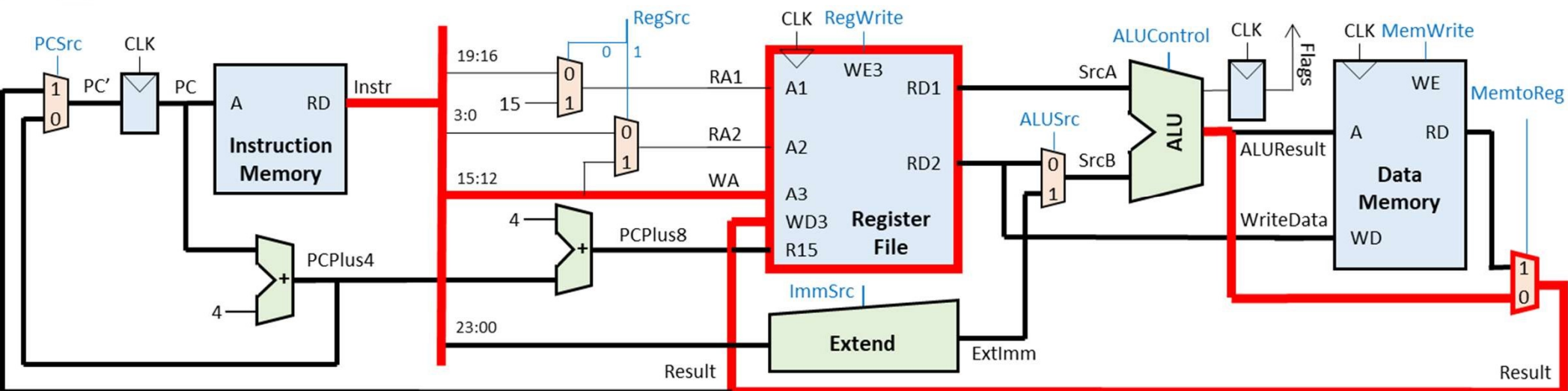
- η μονάδα ALU εκτελεί πέραν των αριθμητικών πράξεων (της πρόσθεσης και της αφαίρεσης) και τις λογικές πράξεις AND και OR
 - σύνδεση της θύρας $RD1$ του αρχείου καταχωρητών με την είσοδο $SrcA$ των 32 bit της μονάδας ALU
 - σύνδεση της θύρας $RD2$ του αρχείου καταχωρητών με την είσοδο $SrcB$ των 32 bit της μονάδας ALU
 - εκτελείται αριθμητική ή λογική πράξη και το αποτέλεσμα της πράξης $Rn +/-/and/or Rm$ εμφανίζεται στην έξοδο $ALUResult$ των 32 bit της μονάδας ALU
 - σήματα ελέγχου: $ALUSrc = 0$ και $ALUControl[1:0] = 00 (+), 01 (-), 10 (AND), 11 (OR)$



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-R

Εγγραφή στο αρχείο καταχωρητών (R0-R14)

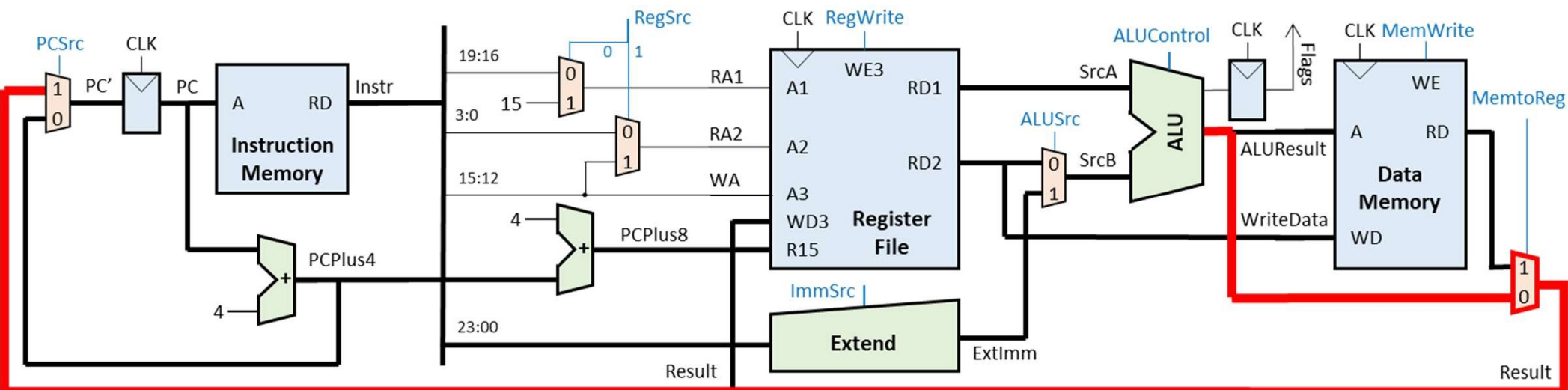
- Το αποτέλεσμα της πράξης στη μονάδα ALU (ALUResult) αποθηκεύεται στον **καταχωρητή προορισμού Rd** του αρχείου καταχωρητών
 - σύνδεση του πεδίου *Rd* της εντολής ($Instr_{15:12}$) στην είσοδο διευθύνσεων εγγραφής *A3*
 - σύνδεση της εξόδου *ALUResult* της μονάδας *ALU* με τη θύρα εγγραφής *WD3* του αρχείου καταχωρητών
 - εγγραφή περιεχομένου καταχωρητή από τη θύρα εγγραφής *WD3*
 - σήματα ελέγχου: **MemtoReg = 0** και **RegWrite = 1** (εγγραφή αρχείου καταχωρητών)



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-R

Εγγραφή στον μετρητή προγράμματος PC (R15)

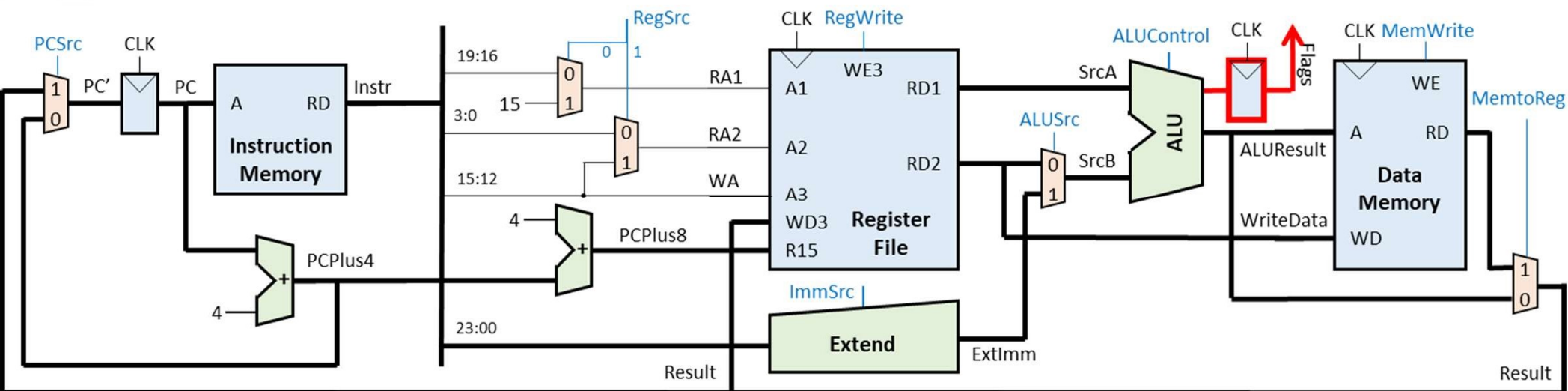
- Το αποτέλεσμα της πράξης στη μονάδα ALU (ALUResult) αποθηκεύεται στον **μετρητή προγράμματος PC (Rd=15)** του αρχείου καταχωρητών
 - ορίζεται ο μετρητής προγράμματος PC ως **καταχωρητής προορισμού Rd** της εντολής ALU(S)-I
 - το αποτέλεσμα της πράξης στην ALU είναι **διεύθυνση της μνήμης εντολών**
 - σύνδεση της εξόδου ALUResult της μονάδας ALU με τον πολυπλέκτη επιλογής διεύθυνσης επόμενης εντολής
 - σήματα ελέγχου: **MemtoReg = 0, RegWrite = 0**



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-R

Ενημέρωση του καταχωρητή κατάστασης

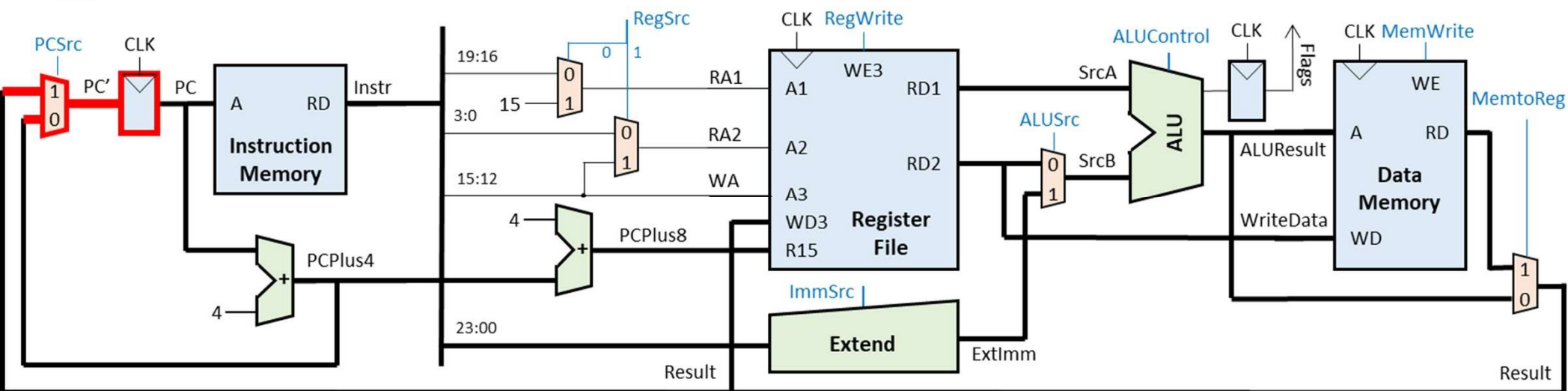
- Εάν το πεδίο S της εντολής είναι 1, τότε ενημερώνεται ο καταχωρητής κατάστασης με τις νέες σημαίες συνθήκης N, Z, C, V (C = 0 και V = 0 στις λογικές πράξεις) στην επόμενη ακμή του CLK
 - το πεδίο S της εντολής ($Instr_{20}$) συμμετέχει στη δημιουργία του σήματος έγκρισης εγγραφής (*FlagsWrite*) του καταχωρητή κατάστασης για εντολές επεξεργασίας δεδομένων ($op = 00$) (το *FlagsWrite* δεν φαίνεται στο σχήμα)
- υλοποιείται στη μονάδα ελέγχου



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-R

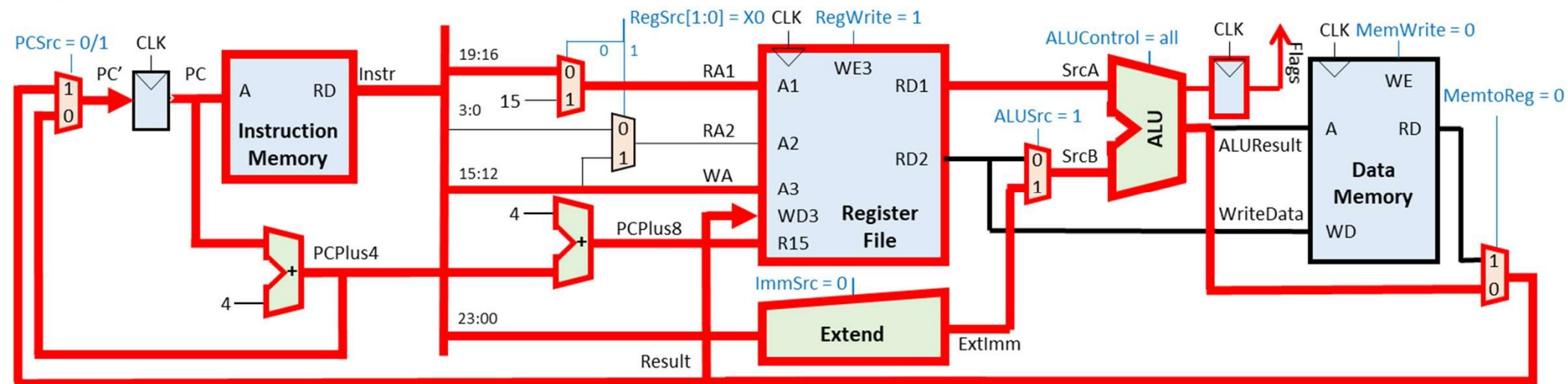
Επιλογή της διεύθυνσης της επόμενης εντολής που θα εκτελεσθεί

- Ο πολυπλέκτης επιλογής διεύθυνσης επόμενης εντολής επιλέγει ανάμεσα σε δύο πιθανές διευθύνσεις: $PC' = PC + 4$ ή $PC' = Rn +/-/and/or Rm$
- Η νέα διεύθυνση αποθηκεύεται στον μετρητή προγράμματος στην επόμενη ανερχόμενη ακμή του CLK
 - σύνδεση της εισόδου 0 του πολυπλέκτη με την έξοδο PCPlus4 του αθροιστή κατά 4
 - σύνδεση της εισόδου 1 του πολυπλέκτη με την έξοδο ALUResult της μονάδας ALU
 - σύνδεση της εξόδου του πολυπλέκτη με την είσοδο PC' του μετρητή προγράμματος
 - σήματα ελέγχου: $PCSrc = 1$ (ALUResult) ή 0 (PCPlus4), $MemtoReg = 0$

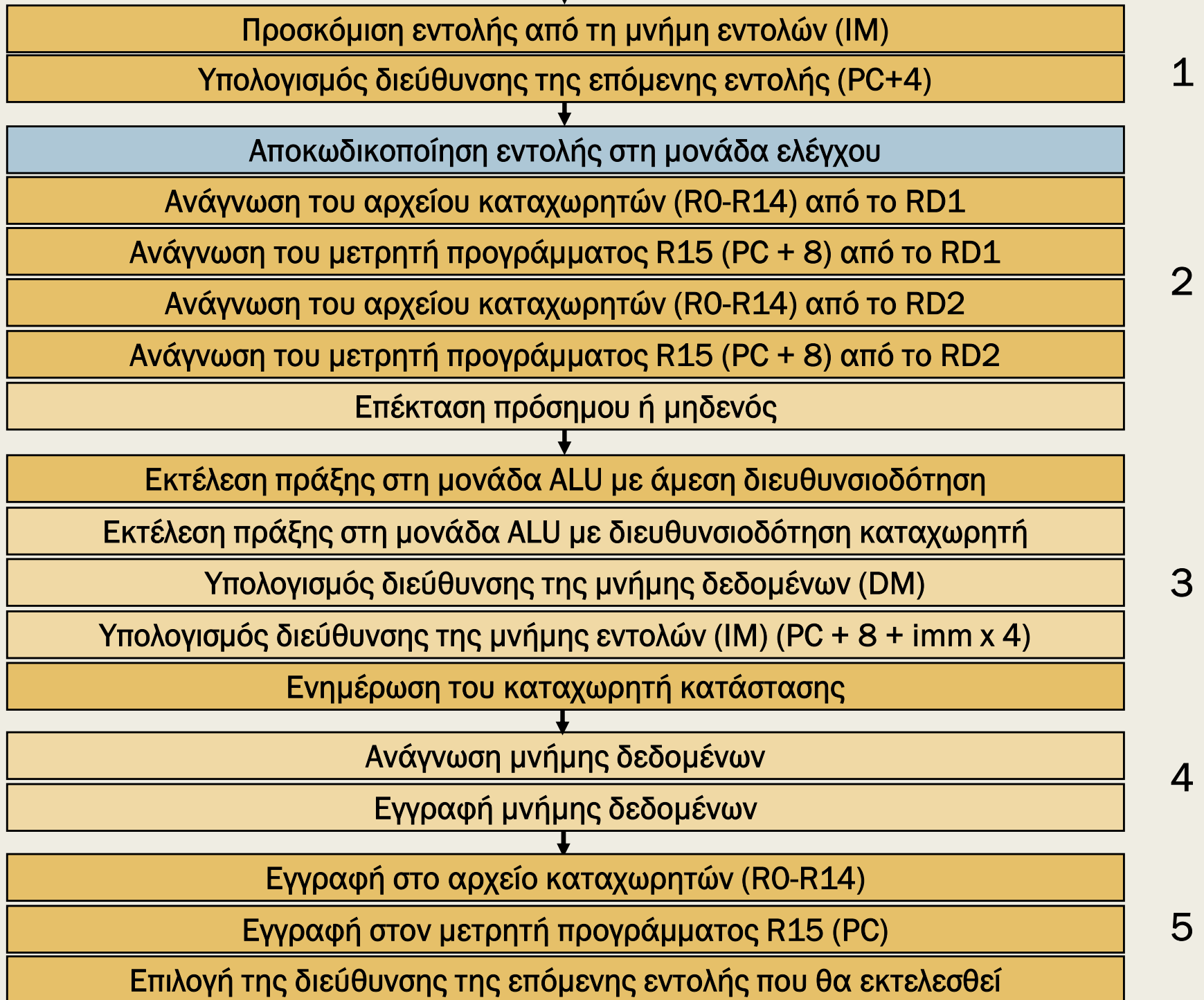


Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολές ALU(S)-R

- Συνολική ροή δεδομένων και τιμές στα σήματα ελέγχου:

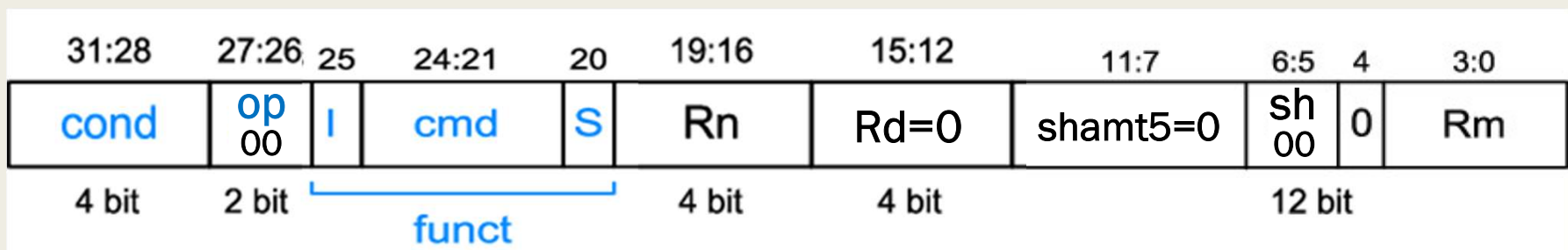
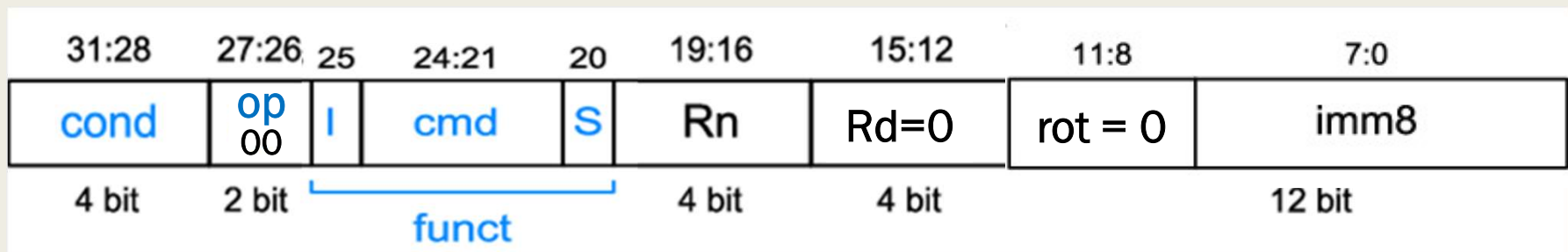


Λειτουργίες εντολών ALU(S)-R



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή CMP

- Στη συνέχεια, θα μελετήσουμε τη διαδρομή δεδομένων που υλοποιεί η **εντολή επεξεργασίας δεδομένων CMP** με **άμεση διευθυνσιοδότηση** και με **διευθυνσιοδότηση καταχωρητή**
- Κώδικας συμβολικής γλώσσας των εντολών CMP:
 - *CMP Rn, #imm8;* ενημέρωση σημαίων για Rn – imm8
 - *CMP Rn, Rm;* ενημέρωση σημαίων για Rn – Rm
- Στη μορφή των εντολών ορίζονται:
 - οι διευθύνσεις των **καταχωρητών προέλευσης Rn** και **Rm**, καθώς και ο **άμεσος τελεστέος imm8** των 8 bit

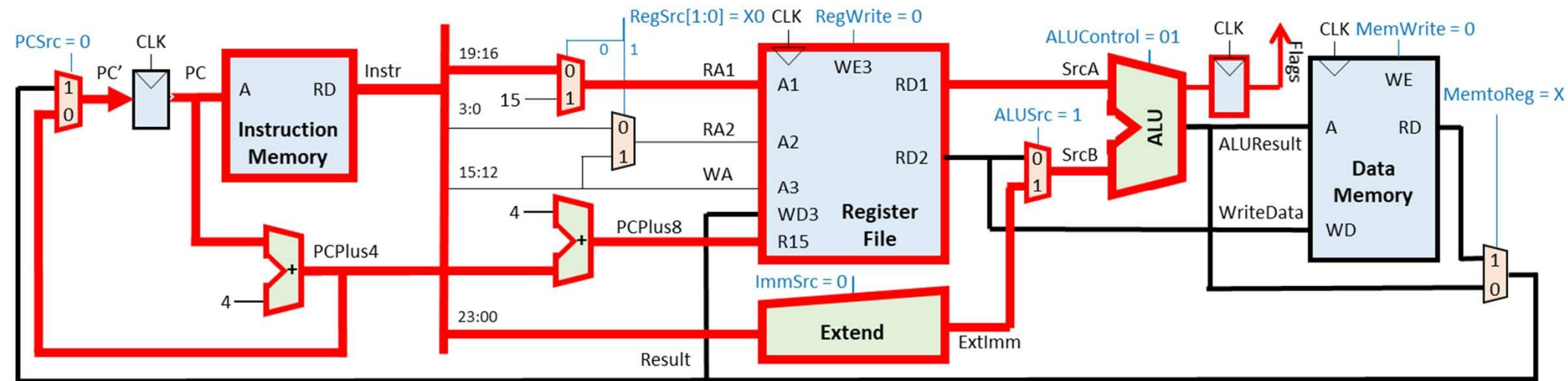


Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή CMP

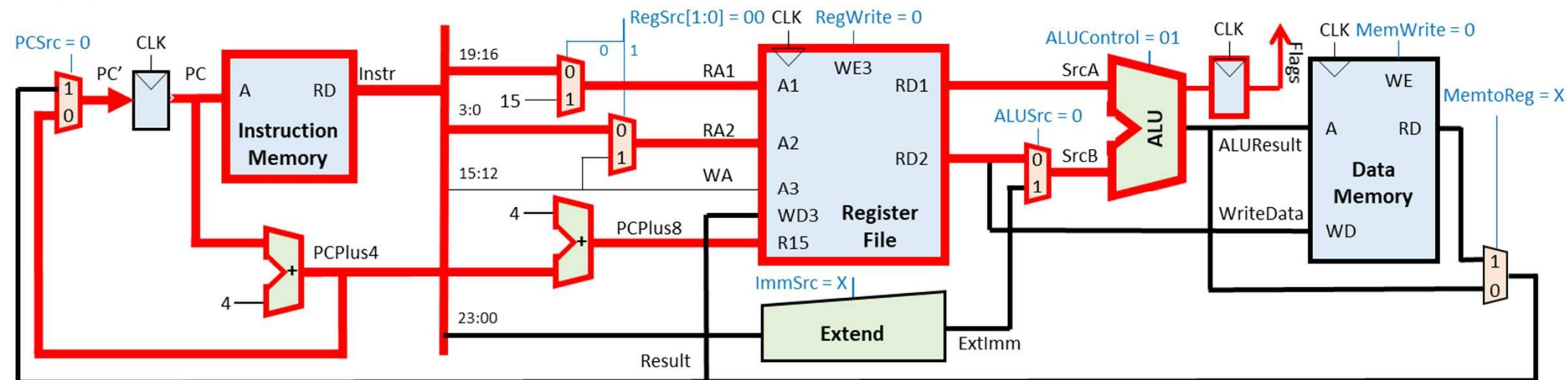
- Η εντολή CMP είναι μία **ειδική περίπτωση της εντολής SUBS**
 - Εκτελεί την πράξη της αφαίρεσης μεταξύ του καταχωρητή προέλευσης Rn και είτε του άμεσου τελεστέου $imm8$, είτε του καταχωρητή προέλευσης Rm
 - το αποτέλεσμα της πράξης δεν αποθηκεύεται στο αρχείο καταχωρητών ($Rd = 0$ και $RegWrite = 0$)
 - ενημερώνεται ο καταχωρητής κατάστασης με τις σημαίες συνθήκης N, Z, C, V στην επόμενη ακμή του CLK
 - ο πολυπλέκτης επιλογής διεύθυνσης επόμενης εντολής επιλέγει αποκλείστηκα τη διεύθυνση επόμενης εντολής: $PC' = PC + 4$ ($PCSrc = 0$)
 - δεν απαιτείται καμία προσθήκη στη διαδρομή δεδομένων, αλλά συγκεκριμένες τιμές στα σήματα ελέγχου:
 - $ALUControl = 01$
 - $RegWrite = 0$
 - $PCSrc = 0$

Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή CMP

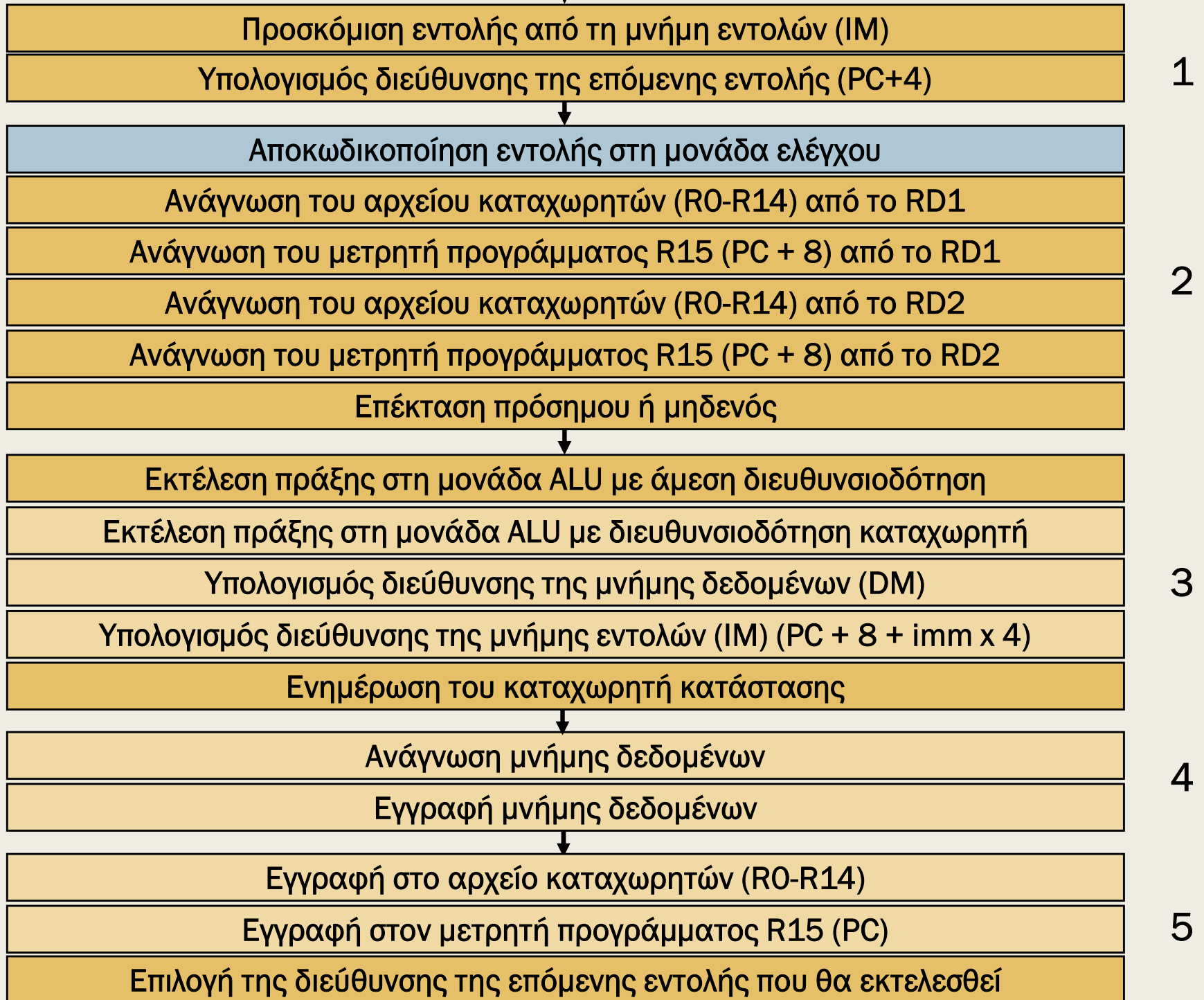
- Συνολική ροή δεδομένων και τιμές στα σήματα ελέγχου:
 - άμεση διευθυνσιοδότηση



- διευθυνσιοδότηση καταχωρητή

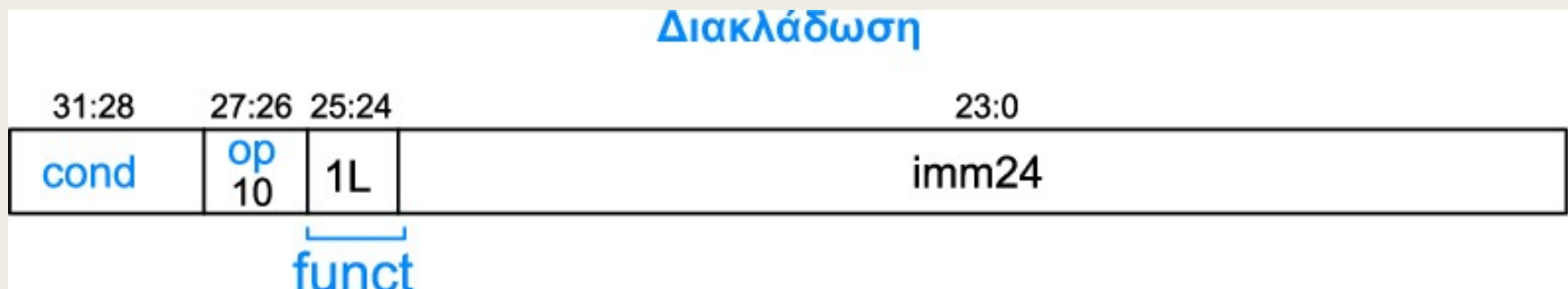


Λειτουργίες εντολής CMP



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή B

- Στη συνέχεια, θα μελετήσουμε τη διαδρομή δεδομένων που υλοποιεί την **εντολή διακλάδωσης B**
- Κώδικας συμβολικής γλώσσας της εντολής B:
 - $B \text{ label}; PC = BTA \rightarrow imm24$
- Στη μορφή των εντολών ορίζονται:
 - προσδιορίζεται ένας **προσημασμένος ακέραιος imm24** σε αναπαράσταση συμπληρώματος ως προς δύο των 24 bit που καθορίζει μία διεύθυνση-στόχο σε σχέση με το **PC+8**
 - η **διεύθυνση - στόχος της διακλάδωσης (BTA)** υπολογίζεται ως εξής:
 $BTA = PC + 8 + imm24 \times 4$ (PC-σχετική διευθυνσιοδότηση)
- Απαιτείται **επέκταση πρόσημου** στη μονάδα Extend και η εκτέλεση της πράξης της **πρόσθεσης** στη μονάδα ALU



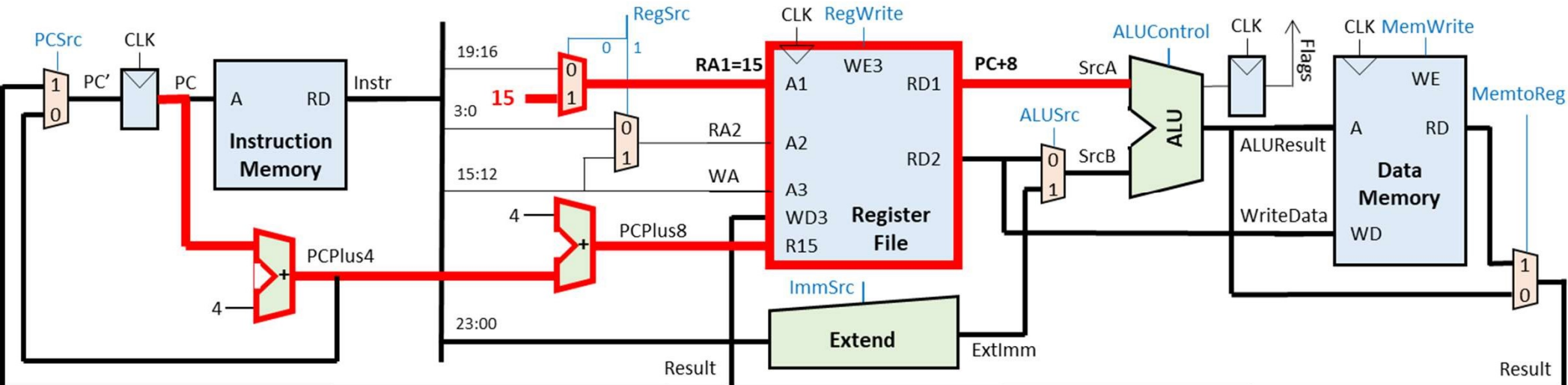
Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή B

- Απαιτούμενες λειτουργίες της εντολής B που ήδη υποστηρίζονται:
 - η διεύθυνση της τρέχουσας εντολής που είναι αποθηκευμένη στον μετρητή προγράμματος PC αυξάνεται κατά 8 ($PC+8$) και διαβάζεται στη θύρα ανάγνωσης RD1 του αρχείου καταχωρητών
 - στη μονάδα ALU εκτελείται η πράξη της πρόσθεσης που υπολογίζει το BTA (Branch Target Address)
- Νέες λειτουργίες για την εντολή B:
 - ο προσημασμένος ακέραιος $imm24$ σε αναπαράσταση συμπληρώματος ως προς δύο των 24 bit, αρχικά πολλαπλασιάζεται επί 4, ώστε να γίνει 26 bit ($imm24 \& "00"$) και στη συνέχεια επεκτείνεται στα 32 bit ($imm32$) με **επέκταση πρόσημου** στη μονάδα Extend
 - ο πολυπλέκτης επιλογής διεύθυνσης επόμενης εντολής επιλέγει τη διεύθυνση: $PC' = PC + 8 + imm32$

Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή B

Ανάγνωση του μετρητή προγράμματος R15 (PC + 8) από το RD1

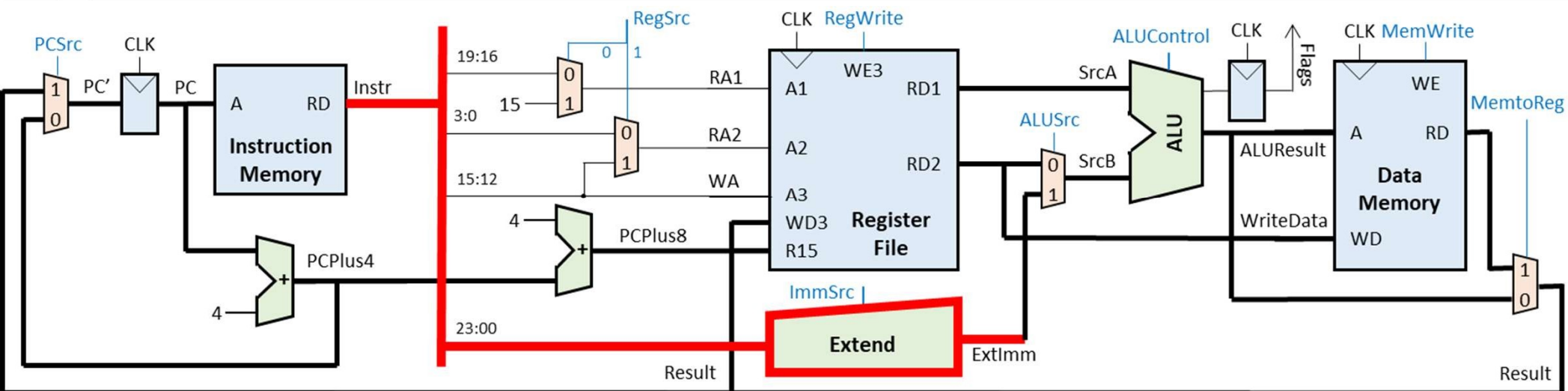
- ο τελεστής, που είναι η διεύθυνση της τρέχουσας εντολής και είναι αποθηκευμένος στον **μετρητή προγράμματος PC** (R15), διαβάζεται στη θύρα ανάγνωσης RD1
 - ορίζεται ο μετρητής προγράμματος PC ως **καταχωρητής προέλευσης R15** της εντολής
 - η ανάγνωση του R15 του αρχείου καταχωρητών επιστρέφει πάντα την τιμή **PC + 8**
 - σύνδεση της **τιμής 15** στην είσοδο διευθύνσεων ανάγνωσης A1
 - σύνδεση της εξόδου PC του μετρητή προγράμματος με την είσοδο R15 του αρχείου καταχωρητών μέσω δύο αθροιστών κατά 4, ώστε να υπολογιστεί το PC+8
 - απαίτηση για έναν επιπλέον αθροιστή κατά 4 με είσοδο PCPlus4 και έξοδο PCPlus8
 - μεταφορά της εισόδου R15 στη θύρα ανάγνωσης RD1 (μέσω εσωτερικού πολυπλέκτη)
 - σήματα ελέγχου: **RegScr[1:0] = X1** και **RegWrite = 0** (ανάγνωση αρχείου καταχωρητών)



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή B

Επέκταση πρόσημου ή μηδενός

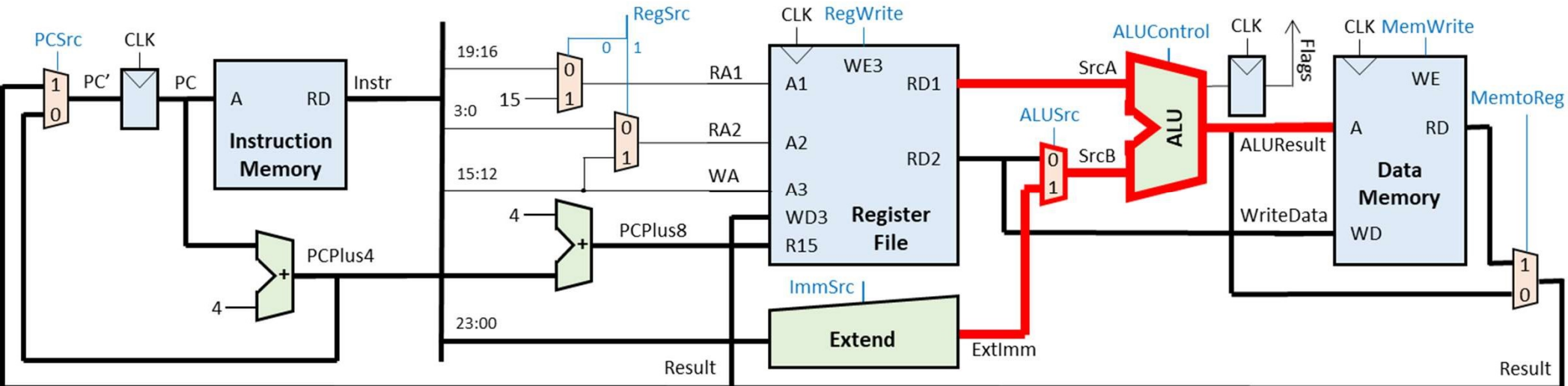
- Η εντολή διακλάδωσης υποστηρίζει PC-σχετική διευθυνσιοδότηση με βάση τον άμεσο τελεστή των 24 bit, που είναι προσημασμένος ακέραιος σε αναπαράσταση συμπληρώματος ως προς δύο
 - σύνδεση του πεδίου του άμεσου τελεστού ($imm24$) της εντολής ($Instr_{23:0}$) στην είσοδο της μονάδας *Extend*
 - πολλαπλασιασμό του άμεσου τελεστού επί 4 ($imm24 \times 4 = imm24 \& "00"$)
 - επέκταση πρόσημου από τα 26 bit στα 32 bit στη μονάδα *Extend*
 - το αποτέλεσμα εμφανίζεται στην έξοδο *ExtImm* της μονάδας *Extend*
 - σήματα ελέγχου: **ImmScr = 1**



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή B

Υπολογισμός διεύθυνσης της μνήμης εντολών (IM) ($PC + 8 + imm \times 4$)

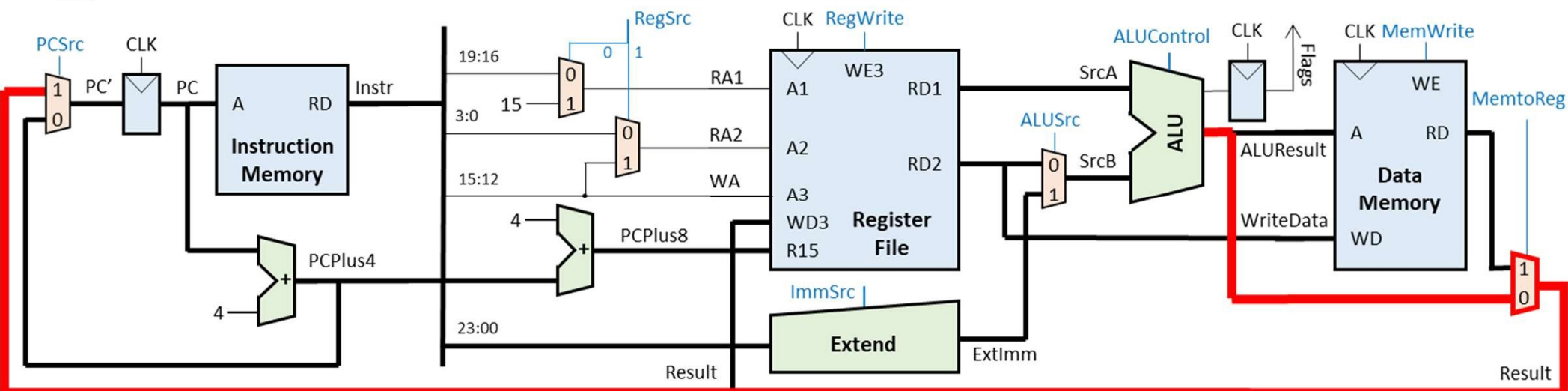
- στη μονάδα ALU υπολογίζεται η διεύθυνση της μνήμης εντολών (IM)
 - σύνδεση της θύρας $RD1$ ($PC+8$) του αρχείου καταχωρητών με την είσοδο $SrcA$ των 32 bit της μονάδας ALU
 - σύνδεση της εξόδου $ExtImm$ ($imm24 \times 4$) της μονάδας $Extend$ με την είσοδο $SrcB$ των 32 bit της μονάδας ALU
 - εκτελείται πρόσθεση στη μονάδα ALU και το αποτέλεσμα της πράξης ($PC + 8$) + ($imm24 \times 4$) εμφανίζεται στην έξοδο $ALUResult$ των 32 bit
 - σήματα ελέγχου: $ALUSrc = 1$ και $ALUControl[1:0] = 00 (+)$



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή B

Εγγραφή στον μετρητή προγράμματος PC (R15)

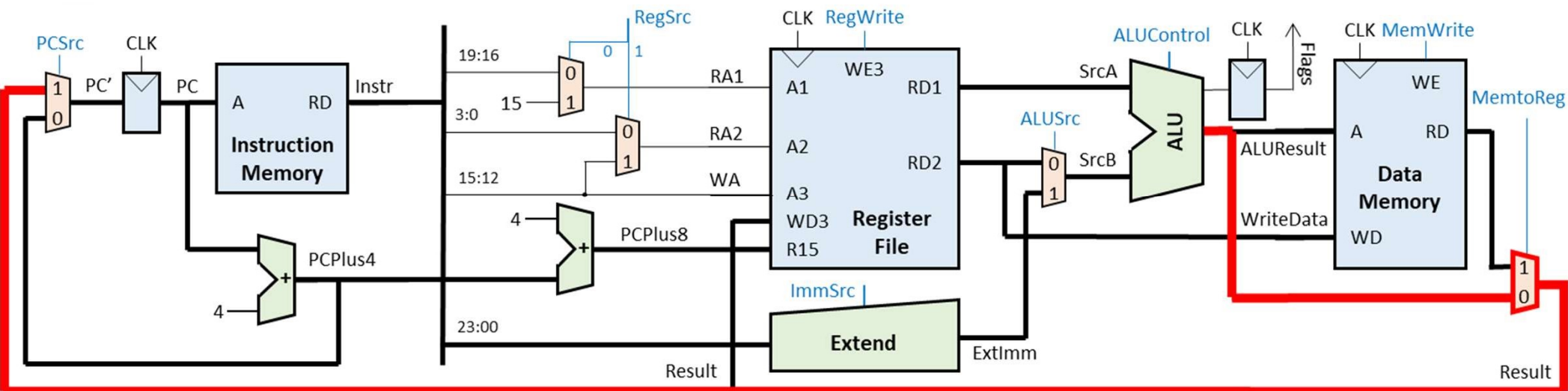
- Το αποτέλεσμα της πράξης στη μονάδα ALU (ALUResult) αποθηκεύεται στον **μετρητή προγράμματος PC** (R15) του αρχείου καταχωρητών
 - ορίζεται ο μετρητής προγράμματος PC ως **καταχωρητής προορισμού R15**
 - το αποτέλεσμα της πράξης στην ALU είναι **διεύθυνση της μνήμης εντολών**
 - σύνδεση της εξόδου ALUResult της μονάδας ALU με τον πολυπλέκτη επιλογής διεύθυνσης επόμενης εντολής
 - σήματα ελέγχου: **MemtoReg = 0, RegWrite = 0**



Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή B

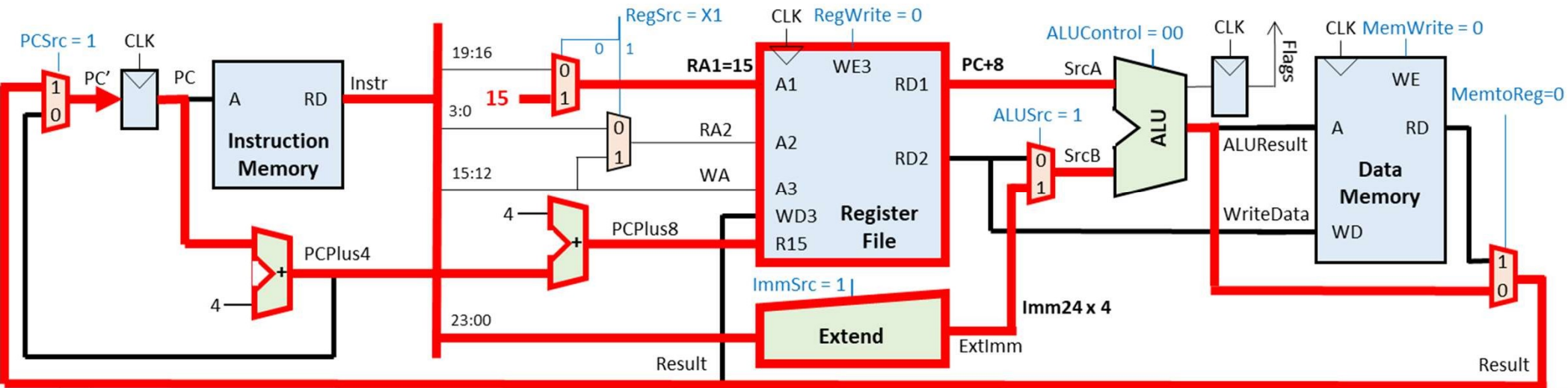
Επιλογή της διεύθυνσης της επόμενης εντολής που θα εκτελεσθεί

- Ο πολυπλέκτης επιλογής διεύθυνσης επόμενης εντολής επιλέγει τη διεύθυνση:
 $PC' = PC + 8 + (imm24 \times 4)$
- Η νέα διεύθυνση αποθηκεύεται στον μετρητή προγράμματος στην επόμενη ανερχόμενη ακμή του CLK
 - σύνδεση της εισόδου 1 του πολυπλέκτη με την έξοδο ALUResult της μονάδας ALU
 - σύνδεση της εξόδου του πολυπλέκτη με την είσοδο PC' του μετρητή προγράμματος
 - σήματα ελέγχου: **PCSrc = 1 (ALUResult), MemtoReg = 0**

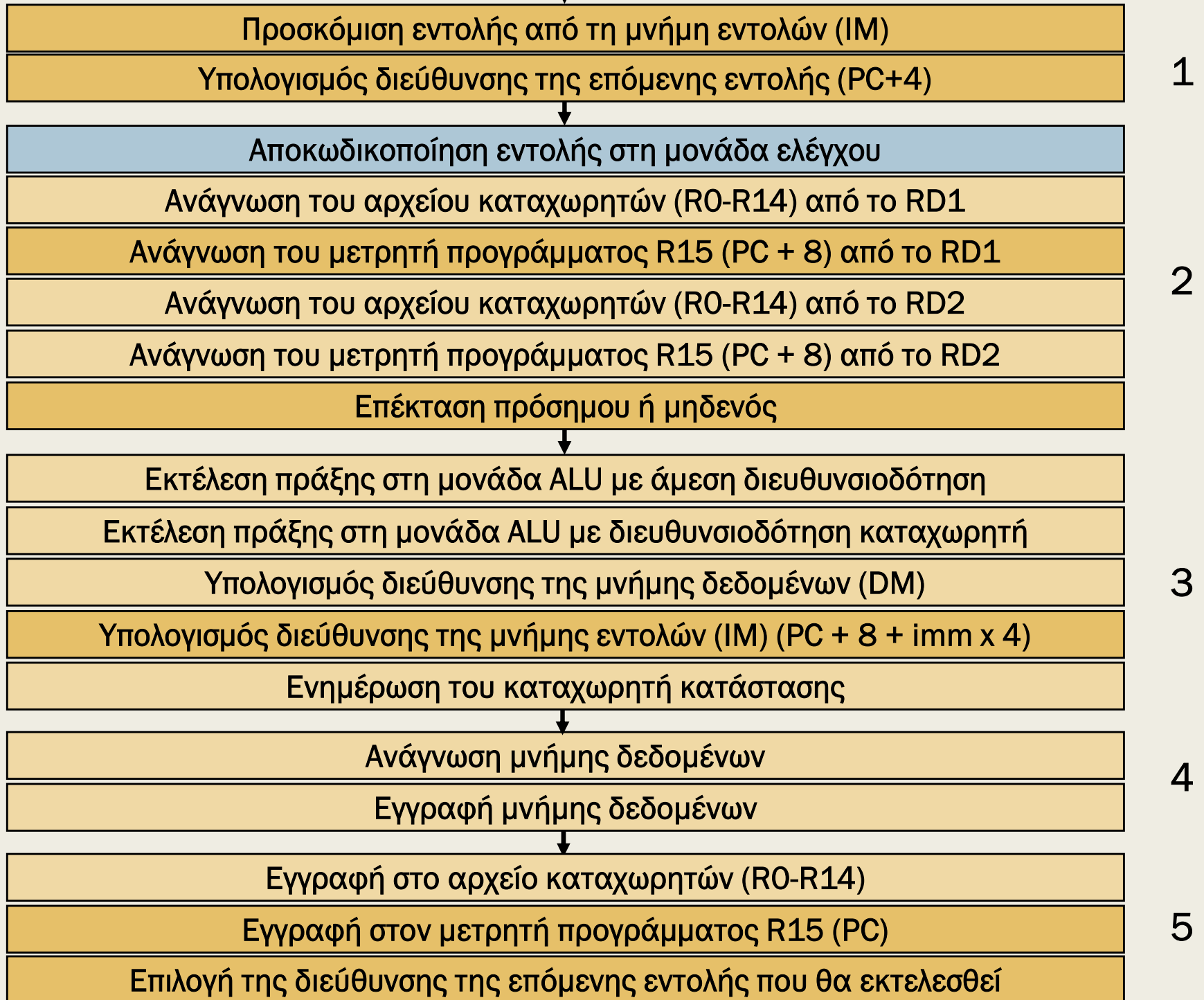


Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Εντολή B

- Συνολική ροή δεδομένων και τιμές στα σήματα ελέγχου:



Λειτουργίες εντολής B

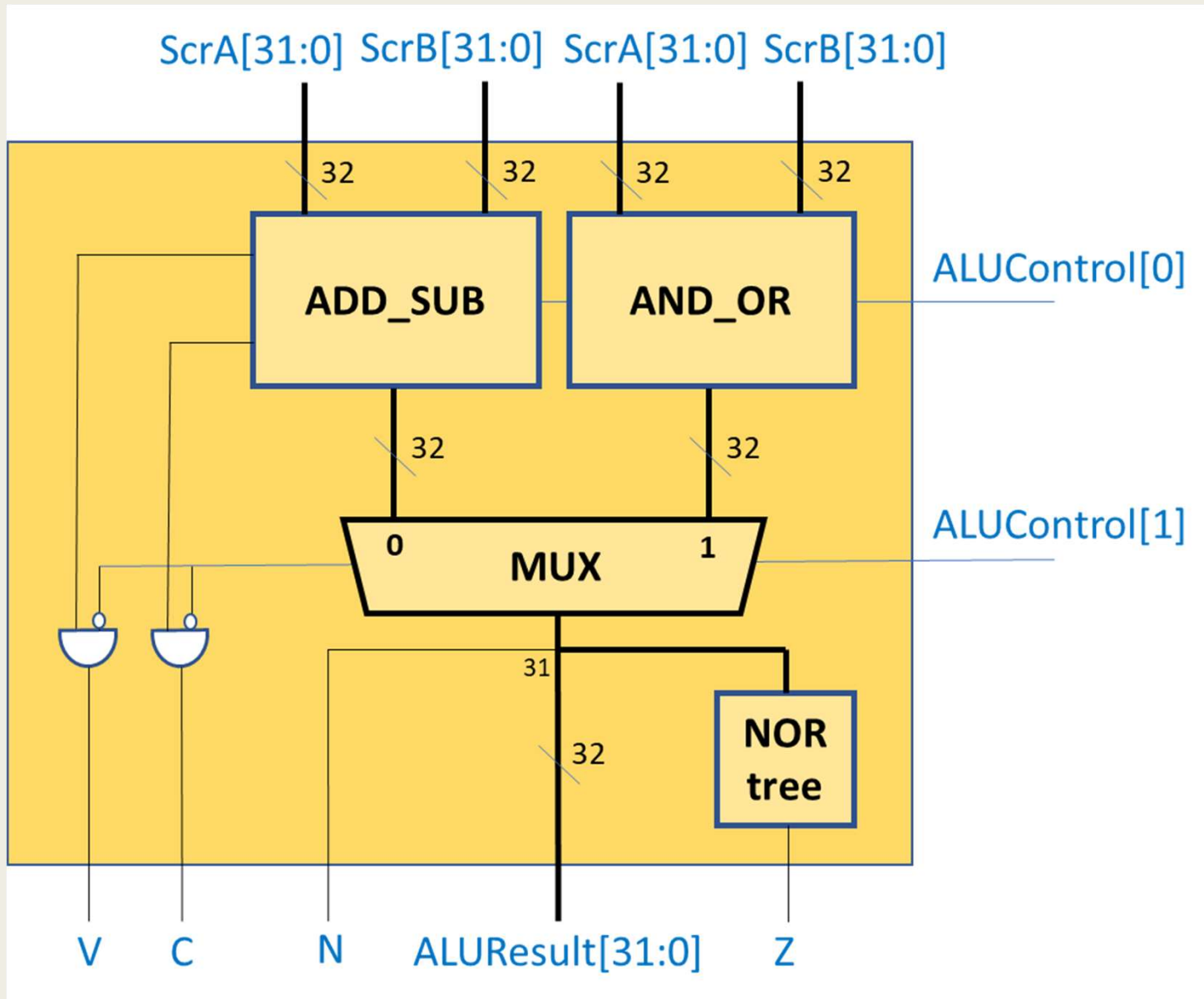


Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Η μονάδα ALU

- Τέλος, θα ολοκληρώσουμε τη διαδρομή δεδομένων με την περιγραφή της δομής της μονάδας ALU
- Με βάση τις εντολές που έχουμε μελετήσει μέχρι τώρα η μονάδα ALU πρέπει να υλοποιεί:
 - τις αριθμητικές πράξεις της πρόσθεσης και της αφαίρεσης, και
 - τις λογικές πράξεις AND και OR
- Η μονάδα ALU λαμβάνει το σήμα ελέγχου **ALUControl[1:0]** που ορίζει τη λειτουργία της και τις εισόδους **SrcA** και **SrcB** των 32 bit
- Η μονάδα ALU παράγει το αποτέλεσμα της πράξης **ALUResult** των 32 bit και τις **σημείες συνθήκης N, Z, C, V** (C = 0 και V = 0 στις λογικές πράξεις)

ALUControl[1:0]	Πράξη
00	Add
01	Subtract
10	AND
11	OR

Επεξεργαστής ενός κύκλου: μελέτη διαδρομής δεδομένων – Η μονάδα ALU



Επεξεργαστής ενός κύκλου: μελέτη μονάδας ελέγχου

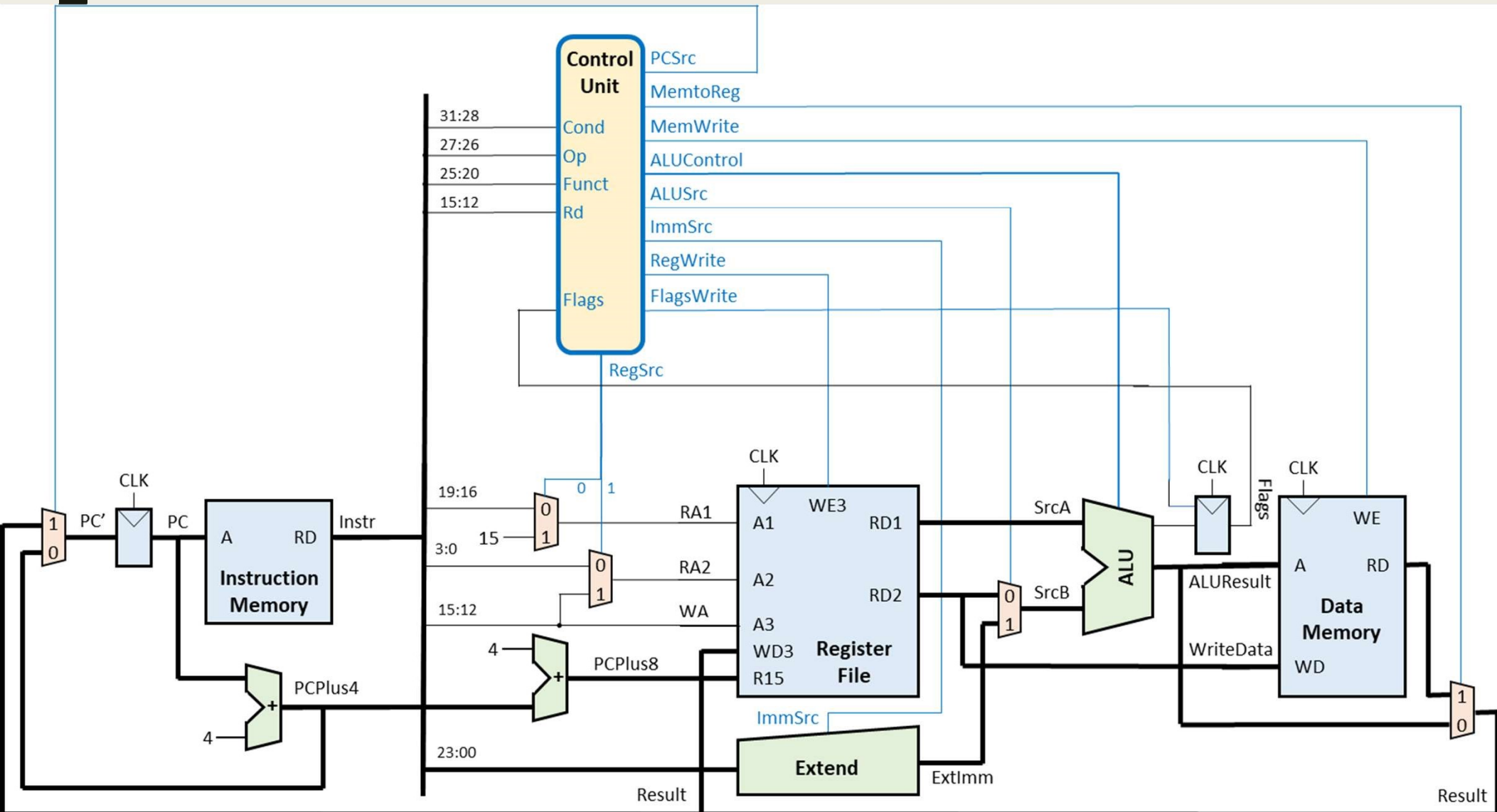
- Η **μονάδα ελέγχου** (control unit) παράγει τα κατάλληλα σήματα ελέγχου για τον χρονισμό του επεξεργαστή με τη χρήση:
 - συνδυαστικής λογικής στην περίπτωση επεξεργαστή ενός κύκλου
- Είσοδοι της μονάδας ελέγχου:
 - πεδίο **cond** της εντολής ($Instr_{31:28}$) για υποστήριξη εντολών υπό συνθήκη
 - πεδίο **op** της εντολής ($Instr_{27:26}$) για τον προσδιορισμό του τύπου της εντολής (επεξεργασία δεδομένων, μνήμης, διακλάδωσης)
 - πεδίο **funct** της εντολής ($Instr_{25:20}$) για τη δήλωση της λειτουργίας/πράξης
 - σημαίες συνθήκης **flags** (**N, Z, C, V**)
 - πεδίο **Rd** της εντολής ($Instr_{15:12}$) για τη δήλωση του μετρητή προγράμματος PC ως καταχωρητή προορισμού ($Rd = "1111"$)
- Η μονάδα ελέγχου ελέγχει εάν ικανοποιείται η συνθήκη που ορίζεται στο πεδίο **cond** της εντολής με βάση τις τρέχουσες σημαίες συνθήκης
 - παράγει το εσωτερικό σήμα **CondEx_in** που εγκρίνει την εκτέλεση της εντολής

Επεξεργαστής ενός κύκλου: μελέτη μονάδας ελέγχου

- Έξοδοι της μονάδας ελέγχου:
 - *σήματα επιλογής πολυπλεκτών*
 - **RegSrc[1:0]** (ανάγνωση από το αρχείο καταχωρητών)
 - **ALUSrc** (μονάδα ALU)
 - **MemtoReg** (ετεροχρονισμένη εγγραφή στο αρχείο καταχωρητών και στον μετρητή προγράμματος)
 - *σήματα ελέγχου λογικής*
 - **ALUControl[1:0]** (μονάδα ALU)
 - **ImmSrc** (μονάδα Extend)
 - *σήματα έγκρισης εγγραφής – ελέγχονται από το σήμα **CondEx_in***
 - **RegWrite** (αρχείο καταχωρητών)
 - **FlagsWrite** (καταχωρητής καταστάσεων)
 - **MemWrite** (μνήμη δεδομένων)
 - *σήμα επιλογής διεύθυνσης επόμενης εντολής*
 - **PCSrc** (μετρητής προγράμματος) – ελέγχεται από το σήμα **CondEx_in**

Επεξεργαστής ενός κύκλου: μελέτη μονάδας ελέγχου

- Ολοκληρωμένος επεξεργαστής ενός κύκλου



Επεξεργαστής ενός κύκλου: μελέτη μονάδας ελέγχου

- Στη συνέχεια θα σχεδιάσουμε τη μονάδα ελέγχου που απαρτίζεται από τις ακόλουθες υπομονάδες:

1. Αποκωδικοποιητής εντολής (InstrDec)

- η υπομονάδα που αποκωδικοποιεί τα πεδία **op** και **funct** της εντολής ($Instr_{27:20}$) και παράγει τα σήματα επιλογής πολυπλεκτών που ρυθμίζουν τη ροή δεδομένων και τα σήματα ελέγχου συνδυαστικής λογικής στη διαδρομή δεδομένων

2. Αποκωδικοποιητής για σήματα έγκρισης εγγραφής (WELogic)

- η υπομονάδα που παράγει τα σήματα έγκρισης εγγραφής

3. Λογική επιλογής διεύθυνσης επόμενης εντολής (PCLogic)

- η υπομονάδα που ενεργοποιεί το σήμα επιλογής διεύθυνσης επόμενης εντολής εάν απαιτείται εγγραφή στον R15 (PC) ή εκτελείται εντολή διακλάδωσης

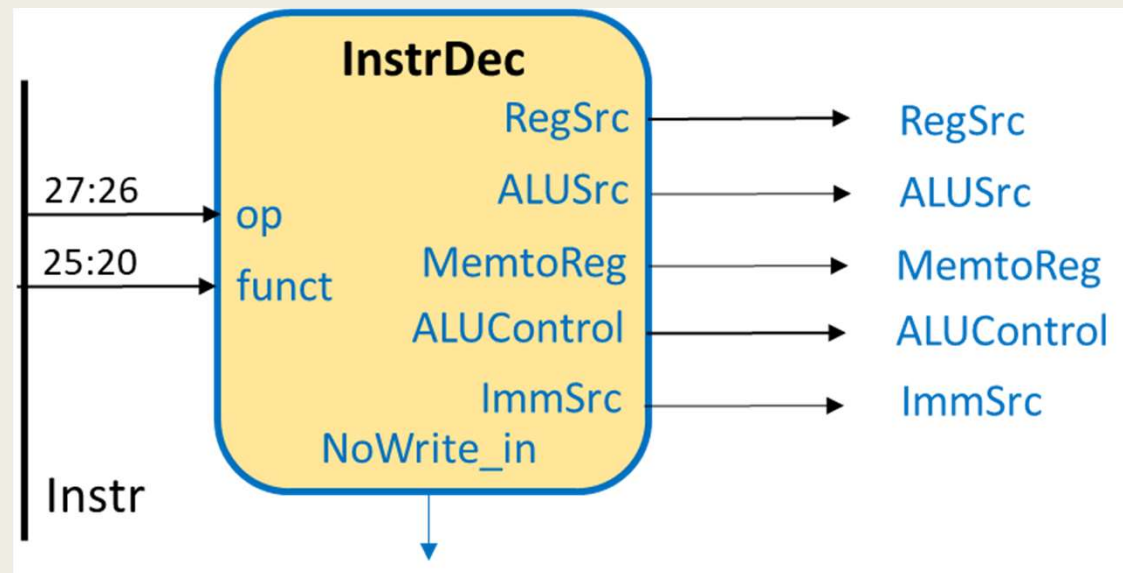
4. Λογική ελέγχου συνθήκης (CONDLogic)

- η υπομονάδα που ελέγχει εάν ικανοποιείται η συνθήκη που ορίζεται στο πεδίο **cond** της εντολής με βάση τις τρέχουσες σημαίες συνθήκης **flags**, ώστε να προχωρήσει η εκτέλεση της εντολής

Επεξεργαστής ενός κύκλου: μελέτη μονάδας ελέγχου

1. Αποκωδικοποιητής εντολής (InstrDec)

- είσοδοι: το πεδίο **op** της εντολής ($Instr_{27:26}$) και το πεδίο **funct** της εντολής ($Instr_{25:20}$)
- έξοδοι: τα σήματα επιλογής πολυπλεκτών **RegSrc**[1:0], **ALUSrc** και **MemtoReg**, τα σήματα ελέγχου λογικής **ALUControl**[1:0] και **ImmSrc**
- εσωτερικό σήμα: το σήμα **NoWrite_in** που εμποδίζει την εγγραφή στο αρχείο καταχωρητών, όταν ενεργοποιείται



Επεξεργαστής ενός κύκλου: μελέτη μονάδας ελέγχου

1. Αποκωδικοποιητής εντολής (InstrDec)

- πίνακας αλήθειας εντολών επεξεργασίας δεδομένων

Εντολή	Instr _{27:26} op	Instr _{25:20} funct	Τύπος	RegSrc	ALUSrc	Imm Src	ALU Control	Memto Reg	NoWrite _in
ADD	00	1 0100 X	DP Imm	X0	1	0	00	0	0
ADD	00	0 0100 X	DP Reg	00	0	X	00	0	0
SUB	00	1 0010 X	DP Imm	X0	1	0	01	0	0
SUB	00	0 0010 X	DP Reg	00	0	X	01	0	0
CMP	00	1 1010 1	DP Imm	X0	1	0	01	X	1
CMP	00	0 1010 1	DP Reg	00	0	X	01	X	1
AND	00	1 0000 X	DP Imm	X0	1	0	10	0	0
AND	00	0 0000 X	DP Reg	00	0	X	10	0	0
ORR	00	1 1100 X	DP Imm	X0	1	0	11	0	0
ORR	00	0 1100 X	DP Reg	00	0	X	11	0	0

Επεξεργαστής ενός κύκλου: μελέτη μονάδας ελέγχου

1. Αποκωδικοποιητής εντολής (InstrDec)

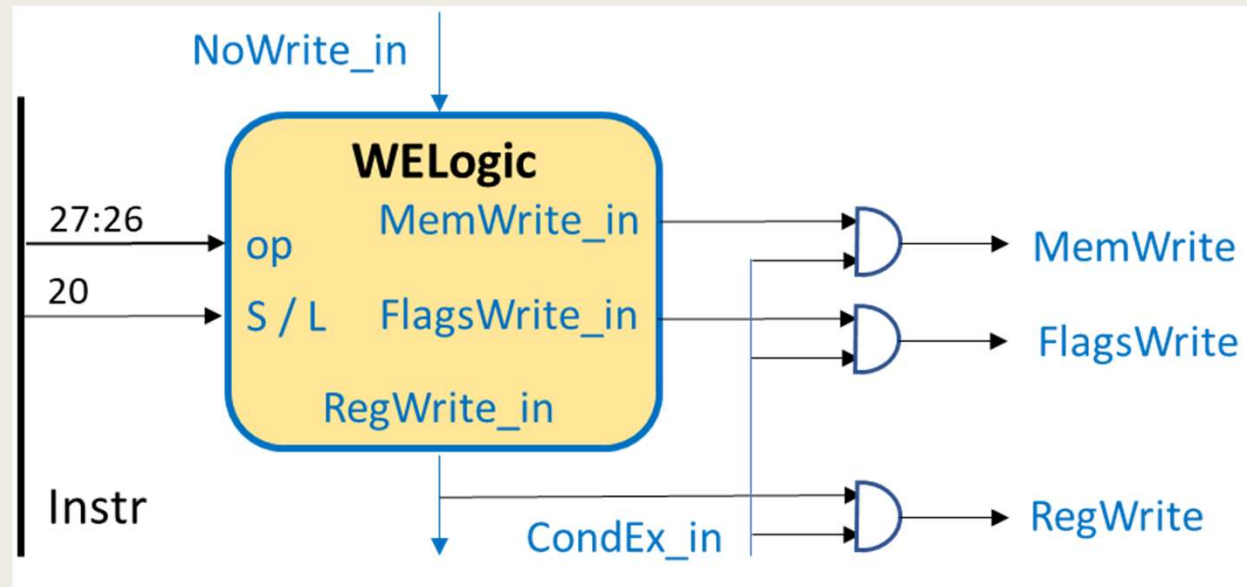
- πίνακας αλήθειας εντολών μνήμης και διακλάδωσης

Εντολή	Instr _{27:26} op	Instr _{25:20} funct	Τύπος	RegSrc	ALUSrc	Imm Src	ALU Control	Memto Reg	NoWrite _in
LDR	01	0 1100 1	M Imm +	X0	1	0	00	1	0
LDR	01	0 1000 1	M Imm -	X0	1	0	01	1	0
STR	01	0 1100 0	M Imm +	10	1	0	00	X	0
STR	01	0 1000 0	M Imm -	10	1	0	01	X	0
B	10	1 0XXX X	B Imm +	X1	1	1	00	0	0

Επεξεργαστής ενός κύκλου: μελέτη μονάδας ελέγχου

2. Αποκωδικοποιητής για σήματα έγκρισης εγγραφής (WELogic)

- είσοδοι: το πεδίο **op** της εντολής ($Instr_{27:26}$), το εσωτερικό σήμα **NoWrite_in** και το πεδίο **S** ή **L** της εντολής ($Instr_{20}$)
- έξοδοι: τα εσωτερικά σήματα **RegWrite_in**, **FlagsWrite_in** και **MemWrite_in**
- εσωτερικό σήμα: το σήμα **CondEx_in** που εμποδίζει την εγγραφή στο αρχείο καταχωρητών, στον καταχωρητή κατάστασης και στη μνήμη δεδομένων, όταν απενεργοποιείται



Επεξεργαστής ενός κύκλου: μελέτη μονάδας ελέγχου

2. Αποκωδικοποιητής για σήματα έγκρισης εγγραφής (WELogic)

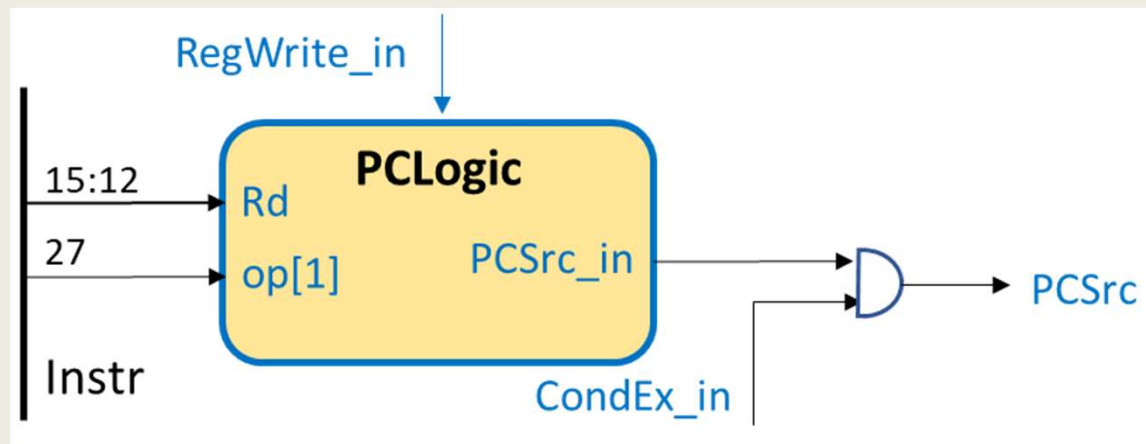
- πίνακας αλήθειας

Τύπος	Instr _{27:26} op	Instr ₂₀ S / L	No Write in	Reg Write in	Mem Write in	Flags Write in
DP	00	0	0	1	0	0
DP	00	1	0	1	0	1
CMP	00	1	1	0	0	1
LDR	01	1	0	1	0	0
STR	01	0	0	0	1	0
B	10	X	0	0	0	0

Επεξεργαστής ενός κύκλου: μελέτη μονάδας ελέγχου

3. Λογική επιλογής διεύθυνσης επόμενης εντολής (PCLogic)

- ενεργοποιεί το σήμα επιλογής διεύθυνσης επόμενης εντολής εάν
 - απαιτείται εγγραφή στον R15 (PC) ($Rd = 15$ και $RegWrite_in = 1$) ή
 - εκτελείται εντολή διακλάδωσης ($op[1] = 1$)
- είσοδοι: το πεδίο Rd της εντολής ($Instr_{15:12}$), το πεδίο $op[1]$ και το εσωτερικό σήμα $RegWrite_in$
- έξοδος: το σήμα $PCSrc_in$ που επιλέγει τη διεύθυνση της επόμενης εντολής που θα εκτελεσθεί
- εσωτερικό σήμα: το σήμα $CondEx_in$ που απενεργοποιεί το σήμα $PCSrc$ ($PC' = PC + 4$)



Επεξεργαστής ενός κύκλου: μελέτη μονάδας ελέγχου

3. Λογική επιλογής διεύθυνσης επόμενης εντολής (PCLogic)

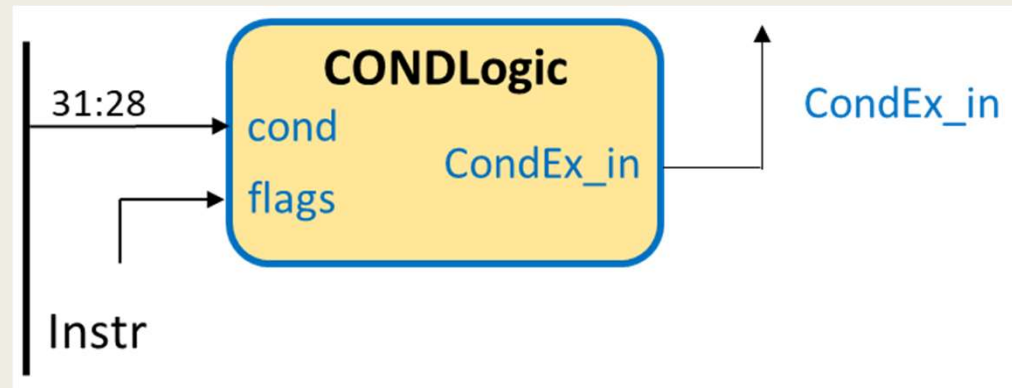
- πίνακας αλήθειας

Τύπος	Instr _{27:26} op	Instr _{15:12} Rd	Reg Write_in	PCSrc_in
DP	00	0000 1110	1	0
DP	00	1111	1	1
CMP	00	XXXX	0	0
LDR	01	0000 1110	1	0
LDR	01	1111	1	1
STR	01	XXXX	0	0
B	10	XXXX	0	1

Επεξεργαστής ενός κύκλου: μελέτη μονάδας ελέγχου

4. Λογική ελέγχου συνθήκης (CONDLogic)

- ελέγχει εάν ικανοποιείται η συνθήκη που ορίζεται στο πεδίο **cond** της εντολής με βάση τις τρέχουσες σημαίες συνθήκης **flags**
- είσοδοι: το πεδίο **cond** της εντολής ($Instr_{31:28}$) και η έξοδος **flags** του καταχωρητή καταστάσεων
- έξοδος: το σήμα **CondEx_in** που εγκρίνει την εκτέλεση της εντολής



Επεξεργαστής ενός κύκλου: μελέτη μονάδας ελέγχου

4. Λογική ελέγχου συνθήκης (CONDLogic)

- πίνακας επιλογής συνθήκης που ελέγχεται εάν αληθεύει σύμφωνα με το πεδίο **cond** (στη στήλη **CondEx** παρατίθεται η εξίσωση Boole)

Instr _{31:28} cond	Μνημονικό	Περιγραφή συνθήκης	CondEx
0000	EQ	Equal	Z
0001	NE	Not equal	\bar{Z}
0010	CS/HS	Carry set / unsigned higher or same	C
0011	CC/LO	Carry clear / unsigned lower	\bar{C}
0100	MI	Minus / negative	N
0101	PL	Plus / positive or zero	\bar{N}
0110	VS	Overflow / overflow set	V
0111	VC	No overflow / overflow clear	\bar{V}

Οι συνθήκες με $\text{Instr}_{28} = 0$ είναι συμπληρωματικές των συνθηκών με $\text{Instr}_{28} = 1$
Δεν ισχύει για $\text{cond} = 1110/1111$

Επεξεργαστής ενός κύκλου: μελέτη μονάδας ελέγχου

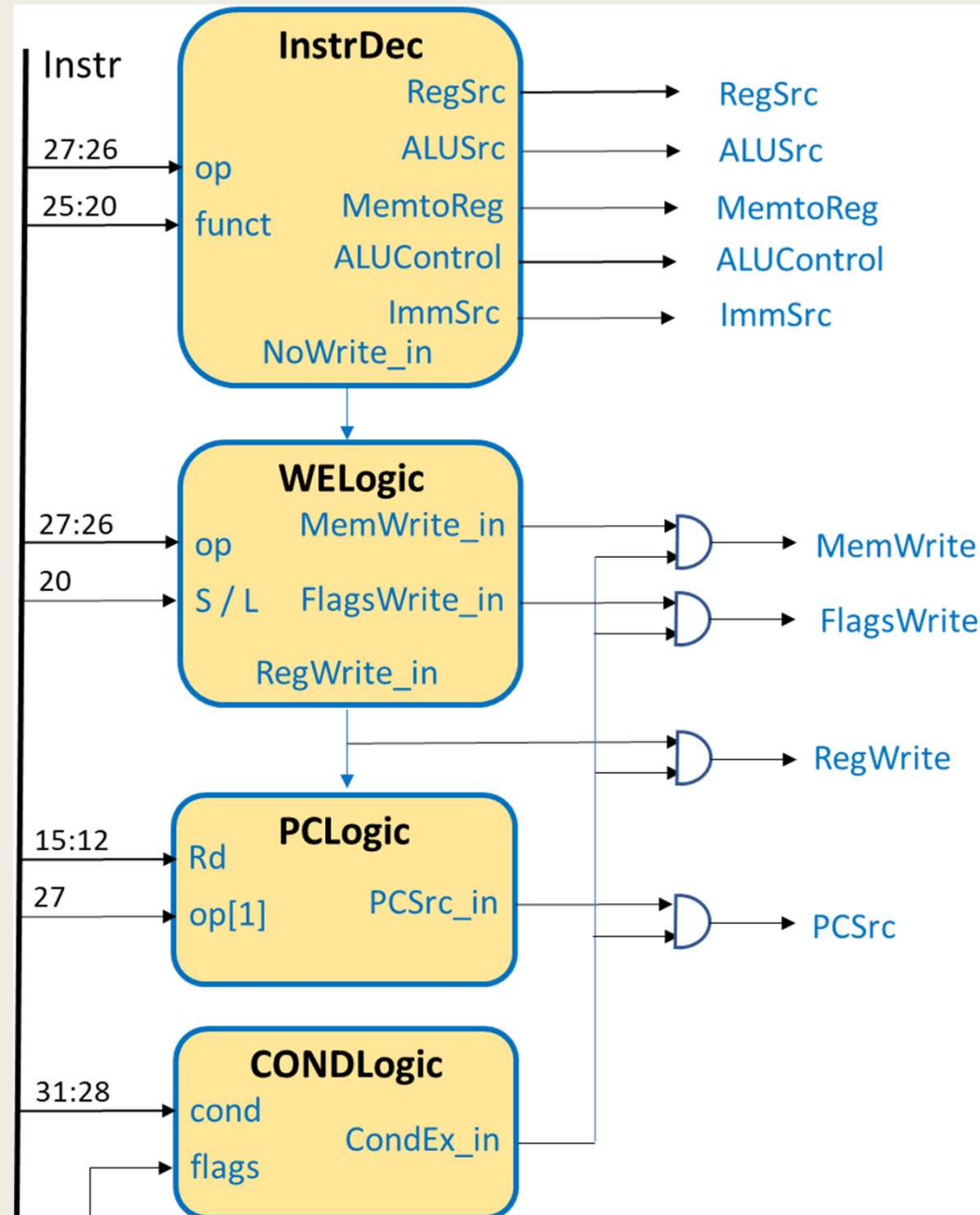
4. Λογική ελέγχου συνθήκης (CONDLogic)

- πίνακας επιλογής συνθήκης που ελέγχεται εάν αληθεύει σύμφωνα με το πεδίο *cond* (στη στήλη *CondEx_in* παρατίθεται η εξίσωση Boole)

Instr _{31:28} cond	Μνημονικό	Περιγραφή συνθήκης	CondEx_in
1000	HI	Unsigned higher	$\bar{Z}C$
1001	LS	Unsigned lower or same	$Z+\bar{C}$
1010	GE	Signed greater or equal	$\overline{N\oplus V}$
1011	LT	Signed less	$N\oplus V$
1100	GT	Signed greater	$\bar{Z} \overline{N\oplus V}$
1101	LE	Signed less or equal	$Z+(N\oplus V)$
1110	AL (ή none)	Always / unconditional	1
1111	none	For unconditional instructions	1

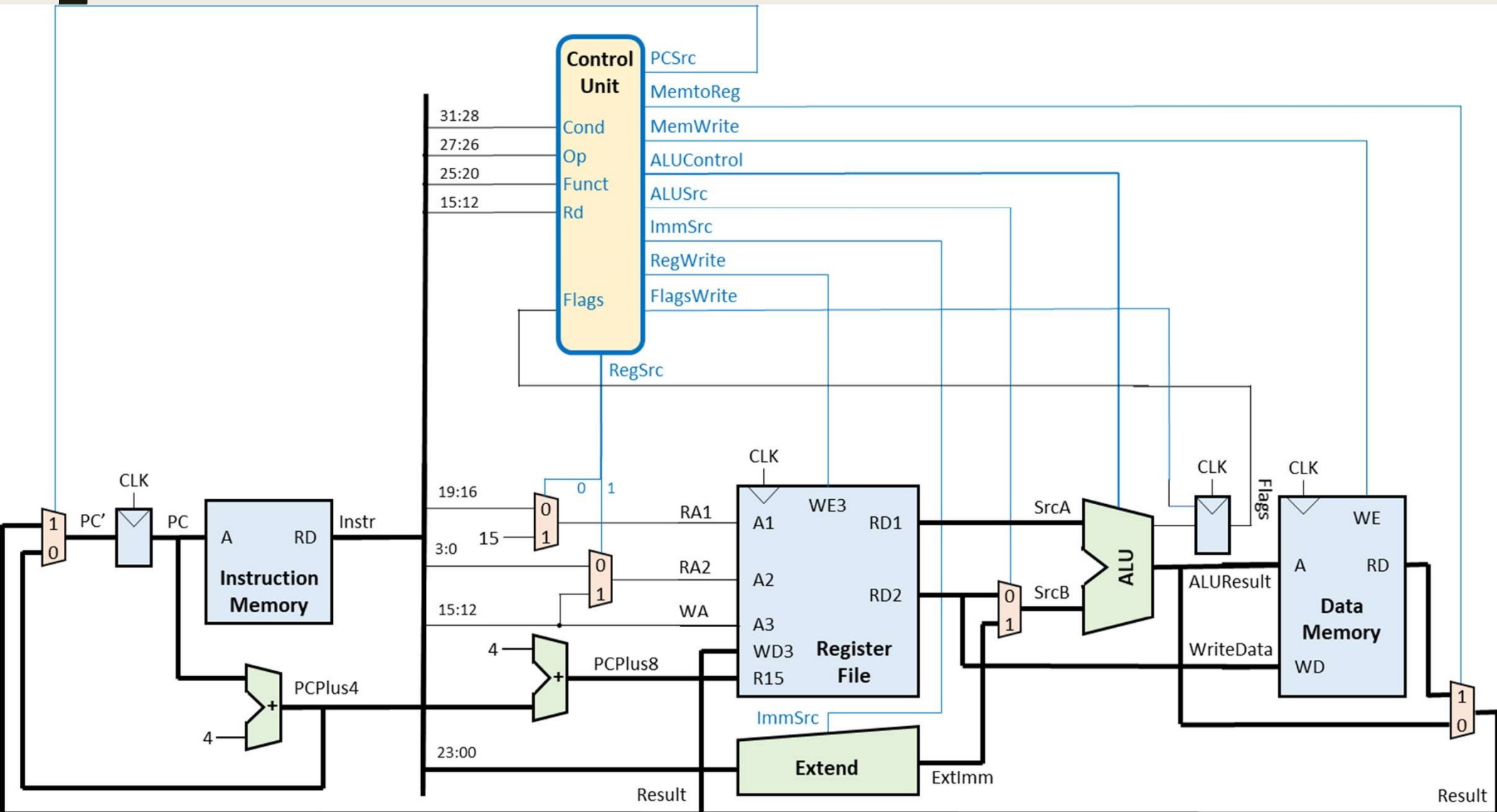
Οι συνθήκες με $\text{Instr}_{28} = 0$ είναι συμπληρωματικές των συνθηκών με $\text{Instr}_{28} = 1$
Δεν ισχύει για $\text{cond} = 1110/1111$

Επεξεργαστής ενός κύκλου: μελέτη μονάδας ελέγχου – Υπομονάδες



Επεξεργαστής ενός κύκλου: μελέτη μονάδας ελέγχου

- Ολοκληρωμένος επεξεργαστής ενός κύκλου



Επεξεργαστής ενός κύκλου: πρόσθετες εντολές

- Τέλος, έχοντας μελετήσει ένα περιορισμένο υποσύνολο του συνόλου εντολών της αρχιτεκτονικής ARM, θα τροποποιήσουμε τη μονάδα ελέγχου και θα προσθέσουμε επιπλέον συνδυαστική λογική στη μονάδα ALU, ώστε ο επεξεργαστής ενός κύκλου να υποστηρίζει την εκτέλεση επιπλέον εντολών επεξεργασίας δεδομένων, όπως:
 - *MOV Rd, #imm8; Rd = imm8*
 - *MOV Rd, Rm; Rd = Rm* και *NOP (MOV R0, R0; R0 = R0)*
 - απαιτείται τροποποίηση της μονάδας ALU, ώστε να εκτελείται και η πράξη **ALUResult = SrcB**
 - *MVN Rd, #imm8; Rd = not imm8*
 - *MVN Rd, Rm; Rd = not Rm*
 - απαιτείται τροποποίηση της μονάδας ALU, ώστε να εκτελείται και η πράξη **ALUResult = not SrcB**
 - *EOR Rd, Rn, #imm8; Rd = Rn xor imm8*
 - *EOR Rd, Rn, Rm; Rd = Rn xor Rm*
 - απαιτείται τροποποίηση της μονάδας ALU, ώστε να εκτελείται και η πράξη **ALUResult = SrcA xor SrcB**

Επεξεργαστής ενός κύκλου: πρόσθετες εντολές

- Τέλος, έχοντας μελετήσει ένα περιορισμένο υποσύνολο του συνόλου εντολών της αρχιτεκτονικής ARM, θα τροποποιήσουμε τη μονάδα ελέγχου και θα προσθέσουμε επιπλέον συνδυαστική λογική στη μονάδα ALU, ώστε ο επεξεργαστής ενός κύκλου να υποστηρίζει την εκτέλεση επιπλέον εντολών επεξεργασίας δεδομένων, όπως:
 - *LSL Rd, Rm, #shamt5; Rd = Rm LSL by #shamt5*
 - *LSR Rd, Rm, #shamt5; Rd = Rm LSR by #shamt5*
 - *ASR Rd, Rm, #shamt5; Rd = Rm ASR by #shamt5*
 - *ROR Rd, Rm, #shamt5; Rd = Rm ROR by #shamt5*
 - απαιτείται προσθήκη ενός **κυκλώματος ολίσθησης** στη μονάδα ALU
 - σύνδεση του πεδίου **shamt** της εντολής ($\text{Instr}_{11:7}$) με την αντίστοιχη είσοδο της μονάδας ALU
 - το σήμα ελέγχου **ALUControl[3:0]** της μονάδας ALU που ορίζει τις λειτουργίες της επεκτείνεται από τα 2 bit στα 4 bit
 - το κύκλωμα ολίσθησης τοποθετείται **παράλληλα** με τη μονάδα ALU, ώστε να μην επιβαρυνθεί σημαντικά η συχνότητα λειτουργίας του επεξεργαστή ενός κύκλου
 - δεν υποστηρίζεται διευθυνσιοδότηση βάσης με ολισθαίνοντα καταχωρητή που απαιτεί το κύκλωμα ολίσθησης **priv** την είσοδο SrcB της μονάδας ALU

Επεξεργαστής ενός κύκλου: πρόσθετες εντολές

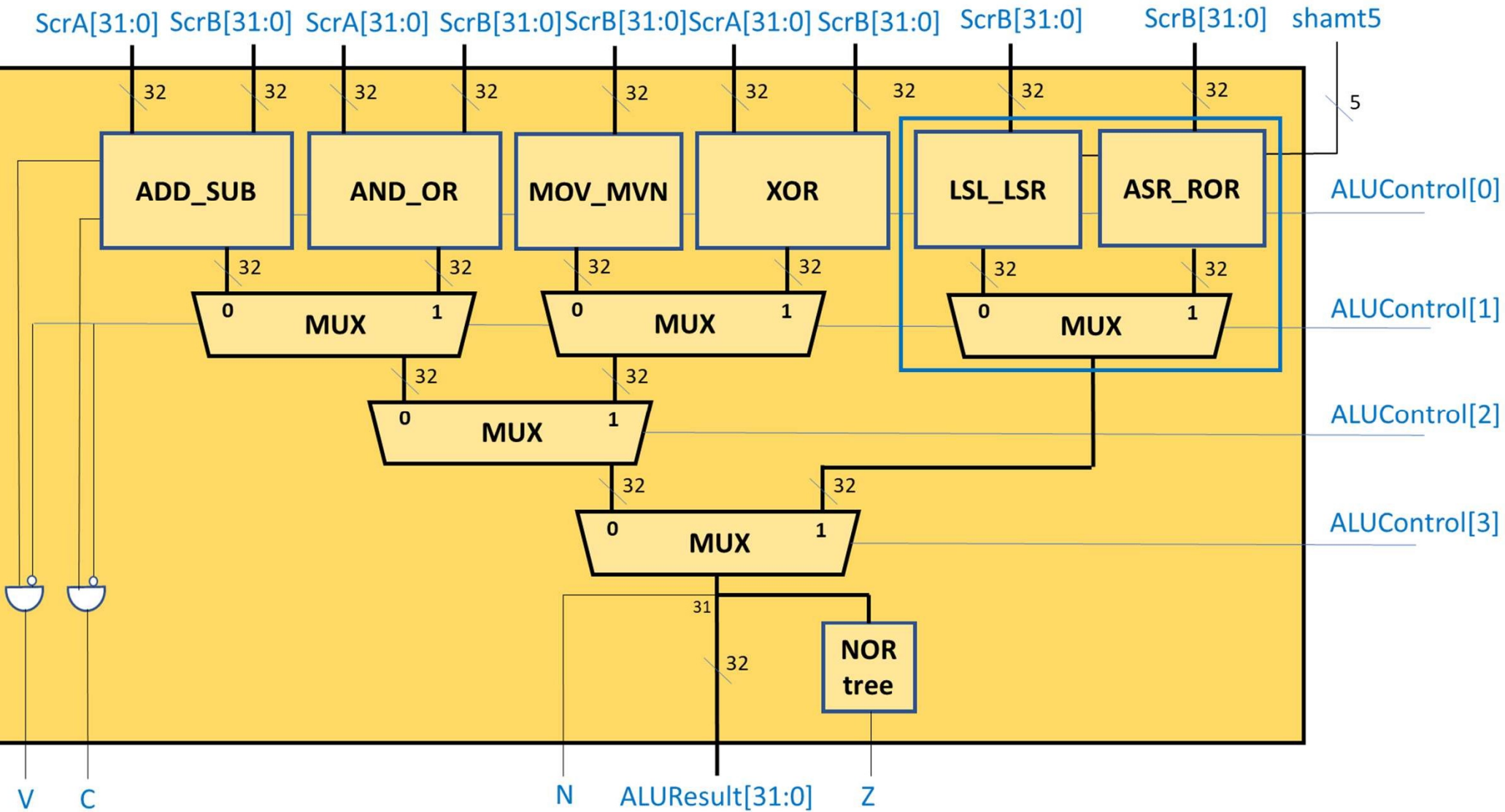
- Πίνακας λειτουργιών της νέας μονάδας ALU

ALUControl	πράξη	ALUControl	πράξη
0000	Add	1000	LSL
0001	Subtract	1001	LSR
0010	AND	1010	ASR
0011	OR	1011	ROR
0100	SrcB	1100	LSL
0101	NOT SrcB	1101	LSR
0110	XOR	1110	ASR
0111	XOR	1111	ROR

Τα δύο λιγότερα σημαντικά ψηφία του ALUControl[1:0] εξακολουθούν να ορίζουν τις ίδιες λειτουργίες που όριζαν στην απλή μονάδα ALU, όταν ALUControl[3:2] = 00

Επεξεργαστής ενός κύκλου: πρόσθετες εντολές

- Περιγραφή δομής της νέας μονάδας ALU



Επεξεργαστής ενός κύκλου: επιλεγμένη άσκηση

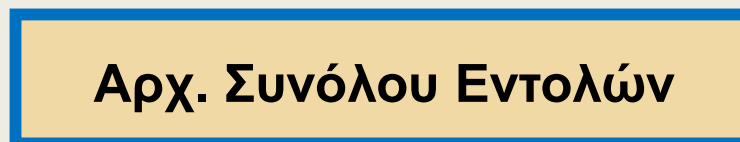
- Να τροποποιήσετε κατάλληλα τον επεξεργαστή ενός κύκλου, ώστε να εκτελεί και την **εντολή BL**
 - το πεδίο *L* (*link*, 1 bit, 24) έχει την τιμή 1 για τη δήλωση της υποστήριξης της διαδικασίας σύνδεσης ($LR = PC + 4$)
- Απαιτούμενες λειτουργίες της εντολής BL που ήδη υποστηρίζονται:
 - το $PC+8$ διαβάζεται στη θύρα ανάγνωσης *RD1* του αρχείου καταχωρητών
 - ο προσημασμένος ακέραιος $imm24 \times 4$ επεκτείνεται στα 32 bit ($imm32$) με επέκταση πρόσημου στη μονάδα *Extend*
 - στη μονάδα *ALU* εκτελείται η πράξη της πρόσθεσης που υπολογίζει το *BTA* (*Branch Target Address*) ($BTA = PC + 8 + imm32$)
 - ο πολυπλέκτης επιλογής διεύθυνσης επόμενης εντολής επιλέγει το *BTA*
- Νέες λειτουργίες για τις εντολές *ALU-R*:
 - το $PC+4$ εγγράφεται στον καταχωρητή *R14* (*link register* – *LR*) του αρχείου καταχωρητών ($A3 = 14$ και $WD3 = PC+4$)
 - σύνδεση της σταθερής τιμής 14 μέσω πολυπλέκτη στην είσοδο διευθύνσεων *A3*
 - σύνδεση της εξόδου $PC+4$ μέσω πολυπλέκτη με τη θύρα *WD3*
 - απαιτούνται δύο νέοι πολυπλέκτες 2 σε 1 και υποστήριξη του κοινού νέου σήματος επιλογής *RegSrc[2]* από τη μονάδα ελέγχου

Ανάλυση επιδόσεων επεξεργαστών

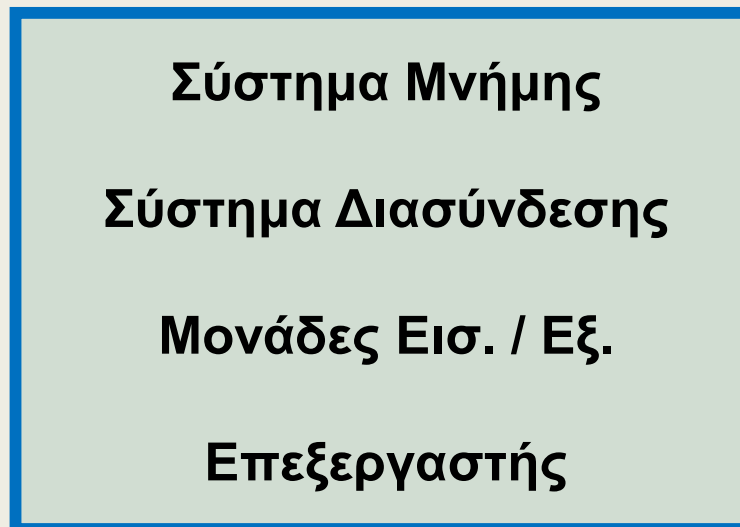
- Τρόποι μέτρησης των επιδόσεων



← Απαντήσεις ανά μήνα
Χρήσιμες λειτουργίες ανά δευτ/πτο



← Millions of Instructions Per Sec. (MIPS)
Millions of Floating Point Operations per Sec. (MFLOP/s)



← Μεταφορά πληροφορίας (MBytes/s)
Χρόνος εκτέλεσης προγράμματος (s)
Συχνότητα λειτουργίας (MHz)

Ανάλυση επιδόσεων επεξεργαστών

- Ο μόνος τρόπος μέτρησης των επιδόσεων χωρίς τεχνάσματα εντυπωσιασμού είναι με βάση τον **χρόνο εκτέλεσης του προγράμματος (CPU_{time})** που μας ενδιαφέρει
 - Η αμέσως επόμενη καλύτερη επιλογή είναι να μετρήσουμε το συνολικό χρόνο εκτέλεσης μιας **συλλογής προγραμμάτων** (μετροπρογραμμάτων - *benchmarks*) που είναι παρόμοια με εκείνα που εκτελούμε
- Ο χρόνος εκτέλεσης ενός προγράμματος (σε δευτερόλεπτα) ορίζεται ως:
 - $\text{CPU}_{\text{time}} = (\# \text{ instructions/program}) \times \text{CPI} \times T_c = \text{seconds/program}$
 - πλήθος εντολών (# instructions/program)
 - μέσος αριθμός κύκλων ανά εντολή (average cycles per instruction – CPI)
 - περίοδος σήματος CLK (seconds/cycle – T_c)
- Στόχοι κατά την σχεδίαση ενός επεξεργαστή:
 - ελαχιστοποίηση χρόνου εκτέλεσης
 - ικανοποίηση περιορισμών που αφορούν: κόστος υλοποίησης και κατανάλωση ισχύος
- Αν και οι συνολικές επιδόσεις ενός υπολογιστή επηρεάζονται από πολλούς παράγοντες, θα εστιάσουμε μόνο στις επιδόσεις του επεξεργαστή

Ανάλυση επιδόσεων επεξεργαστών

- Πώς επηρεάζεται η επίδοση

- $\text{CPU}_{\text{time}} = (\# \text{ instructions/program}) \times \text{CPI} \times T_c = \text{seconds/program}$

	Πλήθος Εντολών	CPI	Περίοδος CLK
πρόγραμμα	X		
μεταγλωττιστής	X	X	
αρχ. συν. εντολών	X	X	X
οργάνωση		X	X
τεχνολογία			X

Η επίδοση του επεξεργαστή διαφοροποιείται από πρόγραμμα σε πρόγραμμα

Ανάλυση επιδόσεων επεξεργαστών

■ Πώς επηρεάζεται η επίδοση

- $\text{CPU}_{\text{time}} = (\# \text{ instructions/program}) \times \text{CPI} \times T_c = \text{seconds/program}$

■ Μέσος αριθμός κύκλων ανά εντολή (cycles per instruction - CPI):

- μέσος αριθμός κύκλων ανά εντολή του ρολογιού που απαιτούνται για την εκτέλεση μιας εντολής ενός προγράμματος
- εξαρτάται από το μίγμα των εντολών σε ένα πρόγραμμα (πόσες εντολές είναι επεξεργασίας δεδομένων, μνήμης ή διακλάδωσης)

■ Εντολές ανά κύκλο (instructions per cycle - IPC):

- διεκπεραιωτική ικανότητα - το αντίστροφο του CPI

■ Περίοδος T_c του ρολογιού (clock period):

- το πλήθος των νανοδευτερολέπτων ανά κύκλο ρολογιού
- καθορίζεται από την **κρίσιμη διαδρομή (critical path)** μέσω της λογικής της εκάστοτε μικροαρχιτεκτονικής του επεξεργαστή
- ο τρόπος σχεδίασης της διαδρομής δεδομένων και των μονάδων που την απαρτίζουν επηρεάζουν την **κρίσιμη διαδρομή (critical path)** και συνεπώς την περίοδο του ρολογιού

Επεξεργαστής ενός κύκλου: ανάλυση επιδόσεων

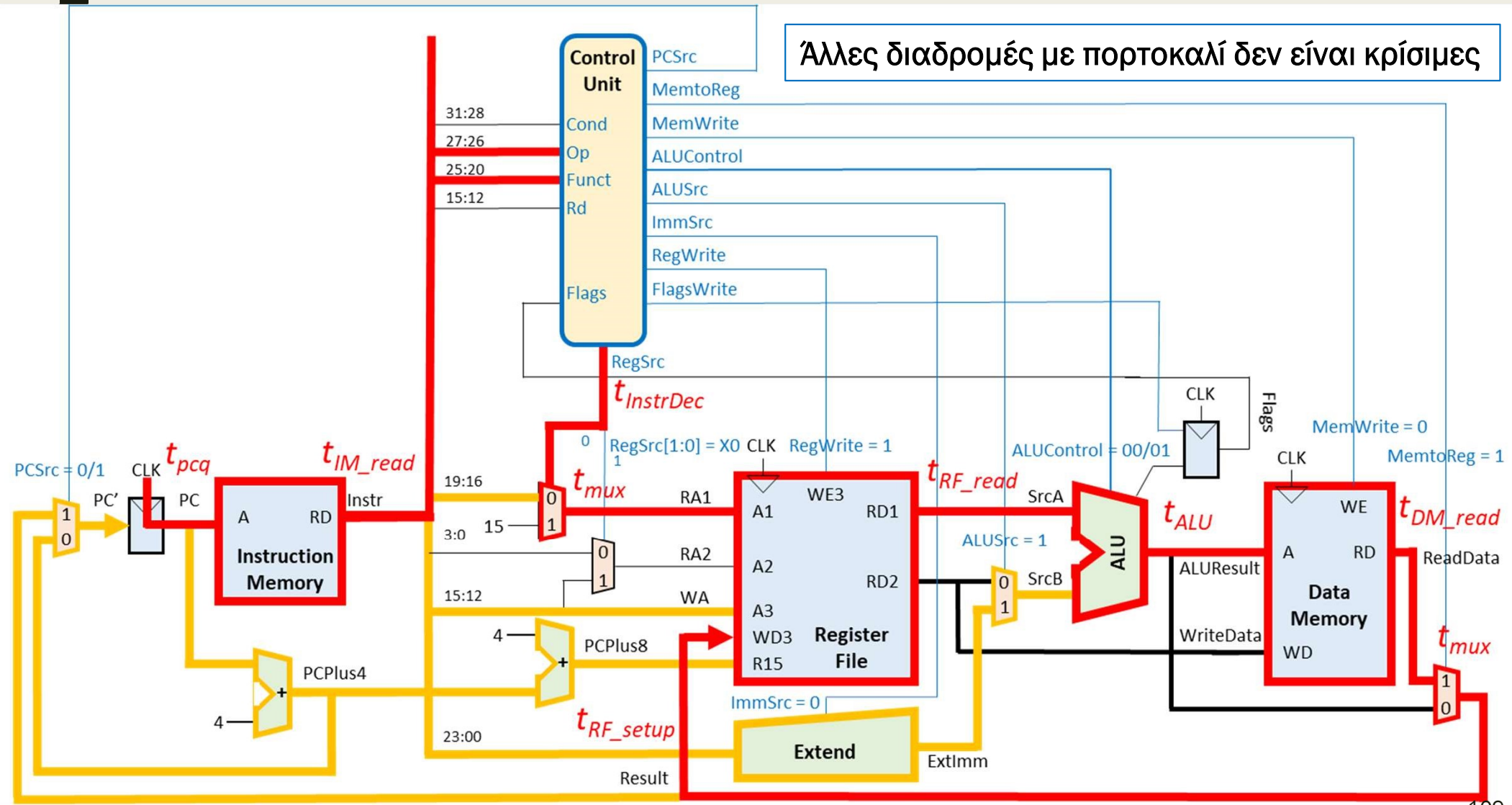
- Ο επεξεργαστής ενός κύκλου έχει πάντα $CPI = 1$
- Η περίοδος του CLK καθορίζεται από την **κρίσιμη διαδρομή (critical path)** που ενεργοποιείται κατά την εκτέλεση της **πιο αργής εντολής** μέσω της λογικής της μικροαρχιτεκτονικής ενός κύκλου
 - η πιο αργή εντολή είναι η **εντολή LDR**
- Λειτουργίες της εντολής LDR με την αντίστοιχη καθυστέρηση διάδοσης που συμβάλλουν στην κρίσιμη διαδρομή:
 - φόρτωση νέας διεύθυνσης εντολής στον μετρητή PC (t_{pcq})
 - ανάγνωση νέας εντολής από τη μνήμη εντολών IM ως Instr (t_{IM_read})
 - υπολογισμός του RegSrc₀ στον αποκωδικοποιητή εντολής InstrDec ($t_{InstrDec}$)
 - επιλογή του πεδίου Rn (Instr_{19:16}) ως RA1 μέσω πολυπλέκτη 2 σε 1 (t_{mux})
 - ανάγνωση του αρχείου καταχωρητών στην έξοδο RD1 ως SrcA (t_{RF_read})
 - υποθέτουμε ότι η δημιουργία του SrcB έχει μικρότερη καθυστέρηση διάδοσης
 - πρόσθεση SrcA με SrcB στη μονάδα ALU - το άθροισμα ως ALUResult (t_{ALU})
 - ανάγνωση τελεστέου από τη μνήμη δεδομένων DM ως ReadData (t_{DM_read})
 - επιλογή του ReadData μέσω πολυπλέκτη 2 σε 1 (t_{mux})
 - σταθεροποίηση ReadData για εγγραφή στο αρχείο καταχωρητών (t_{RF_setup})

Επεξεργαστής ενός κύκλου: ανάλυση επιδόσεων

- Η περίοδος του CLK (T_c) καθορίζεται από την **κρίσιμη διαδρομή (critical path)** που ενεργοποιείται κατά την εκτέλεση της **εντολής LDR**

$$T_c = t_{pcq} + t_{IM_read} + t_{InstrDec} + t_{mux} + t_{RF_read} + t_{ALU} + t_{DM_read} + t_{mux} + t_{RF_setup}$$

Άλλες διαδρομές με πορτοκαλί δεν είναι κρίσιμες



Επεξεργαστής ενός κύκλου: ανάλυση επιδόσεων

- Η περίοδος του CLK (T_c) καθορίζεται από την **κρίσιμη διαδρομή (critical path)** που ενεργοποιείται κατά την εκτέλεση της **εντολής LDR**
 - $T_c = t_{pcq} + t_{IM_read} + t_{InstrDec} + t_{mux} + t_{RF_read} + t_{ALU} + t_{DM_read} + t_{mux} + t_{RF_setup}$
- Παράδειγμα υπολογισμού επιδόσεων επεξεργαστή ενός κύκλου:
 - η περίοδος του ρολογιού (CLK) με ενδεικτικές καθυστερήσεις είναι:
 - $T_c = (40 + 200 + 60 + 40 + 250 + 250 + 300 + 40 + 60) \text{ ps} = 1240 \text{ ps}$
 - ο χρόνος εκτέλεσης ενός προγράμματος των 100 δις εντολών είναι:
 - $\text{CPU}_{\text{time}} = (\# \text{ instr/program}) \times \text{CPI} \times T_c = 100 \times 10^9 \times 1 \times 1240 \times 10^{-12} = 124 \text{ s}$

Ενδεικτικές τιμές

Στοιχεία	Παράμετρος	Καθυστερήση (ps)
Καθυστερήση από το CLK έως την έξοδο Q (καταχωρητής)	T_{pcq}	40
Χρόνος σταθεροποίησης καταχωρητή	T_{setup}	50
Χρόνος σταθεροποίησης αρχείου καταχωρητών	t_{RF_setup}	60
Καθυστερήση διάδοσης πολυπλέκτη	T_{mux}	40
Καθυστερήση διάδοσης στη μονάδα ALU για πρόσθεση	t_{ALU}	250
Καθυστερήση διάδοσης στον αποκωδικοποιητή εντολών	t_{Instr_Dec}	60
Χρόνος ανάγνωσης από τη μνήμη εντολών (ROM)	t_{IM_read}	200
Χρόνος ανάγνωσης από τη μνήμη δεδομένων (Distr.-RAM)	t_{DM_read}	300
Χρόνος ανάγνωσης από το αρχείο καταχωρητών (Distr.-RAM)	t_{RF_read}	250

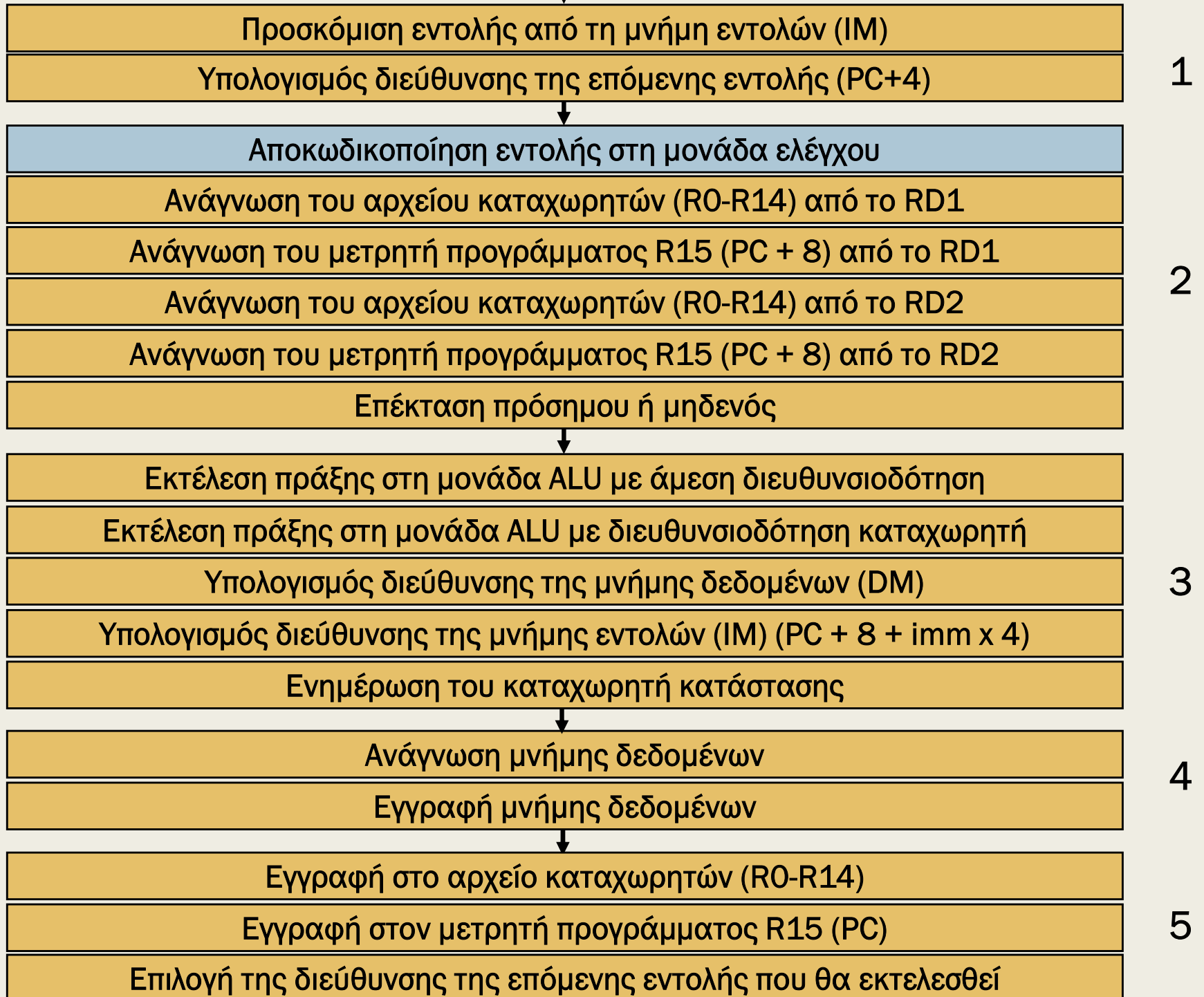
Επεξεργαστής πολλών κύκλων

- Στη συνέχεια, προχωρούμε στη σχεδίαση της **μικροαρχιτεκτονικής πολλών κύκλων** με βάση τη μικροαρχιτεκτονική ενός κύκλου
- Υλοποίηση των βασικών εντολών υπό συνθήκη:
 - εντολές επεξεργασίας δεδομένων: **ADD(S), SUB(S), AND(S), ORR(S), CMP**
 - εντολές μνήμης: **LDR, STR**
 - εντολές διακλάδωσης: **B**
- Υλοποίηση της **διαίρεσης** της εκτέλεσης των εντολών σε **πέντε βήματα** με την προσθήκη **μη-αρχιτεκτονικών καταχωρητών**, στοχεύοντας:
 - **διαφορετικό** πλήθος βημάτων ανά εντολή
 - **μικρότερο** πλήθος βημάτων όταν δεν ικανοποιείται η συνθήκη
 - **μικρό** κύκλο ρολογιού που ορίζεται από την **κρίσιμη διαδρομή (critical path)** που ενεργοποιείται κατά την εκτέλεση του πιο αργού βήματος
- Υλοποίηση της μονάδα ελέγχου κατά τέτοιο τρόπο, ώστε:
 - τα σήματα ελέγχου που **παραμένουν σταθερά** κατά την εκτέλεση της εντολής να παράγονται από **συνδυαστική λογική**
 - τα σήματα **έγκρισης εγγραφής** και το σήμα **PCSrc** να παράγονται από **μηχανή πεπερασμένων καταστάσεων**

Επεξεργαστής πολλών κύκλων

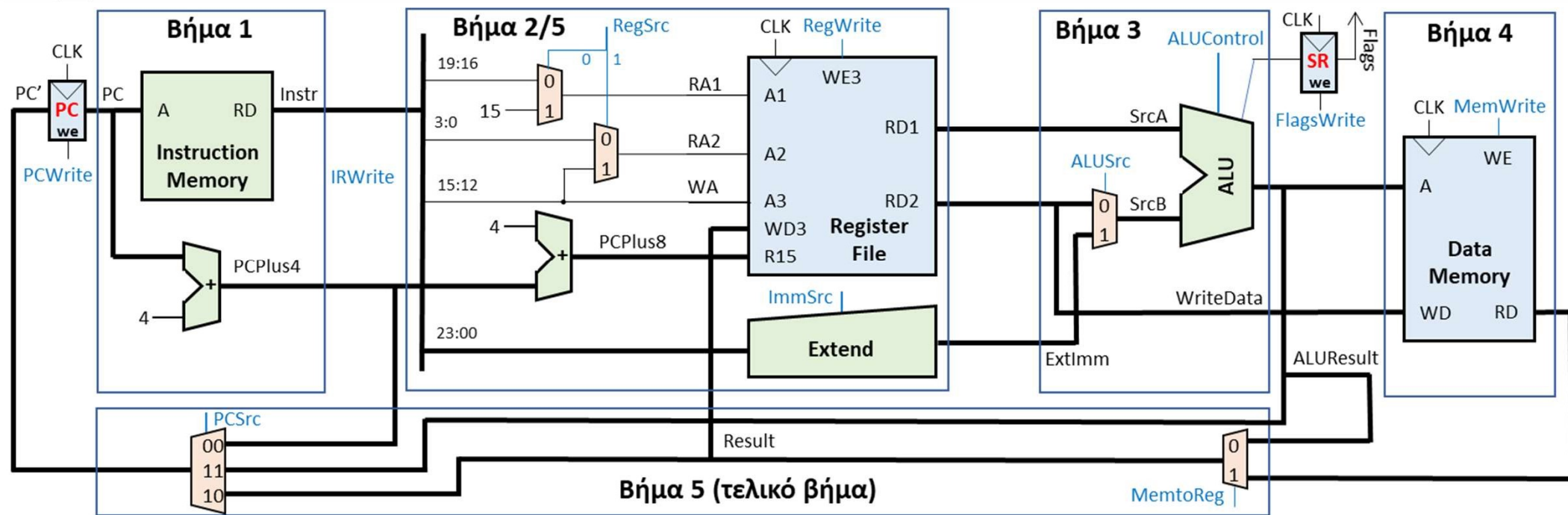
- Αρχικά, οι λειτουργίες ταξινομούνται σε πέντε βήματα εκτέλεσης της εντολής, όπου κάθε ένα βήμα υλοποιείται χωριστά:
 1. Προσκόμιση εντολής και υπολογισμός επόμενης διεύθυνσης (PC+4)
 2. Αποκωδικοποίηση εντολής και ανάγνωση αρχείου καταχωρητών
 3. Εκτέλεση πράξεων στη μονάδα ALU
 4. Ανάγνωση ή εγγραφή στη μνήμη δεδομένων
 5. Ετεροχρονισμένη εγγραφή στο αρχείο καταχωρητών και επιλογή της διεύθυνσης της επόμενης εντολής
- Συνεκτιμώντας ότι η υλοποίηση θα γίνει σε τεχνολογία FPGA προβαίνουμε στις ακόλουθες αρχιτεκτονικές επιλογές:
 - η **μνήμη εντολών** παραμένει **χωριστά** από τη **μνήμη δεδομένων**, ώστε να υλοποιηθεί ως μνήμη ROM
 - διατηρούνται οι **δύο αθροιστές κατά 4** για την αύξηση της τιμής του PC, αξιοποιώντας τους διαθέσιμους πόρους (CARRY4)
 - ανάμεσα στα υλοποιημένα βήματα τοποθετούνται **μη αρχιτεκτονικοί καταχωρητές**, αξιοποιώντας διαθέσιμα D-Flip-flops

Λειτουργίες εντολών (18)



Επεξεργαστής πολλών κύκλων

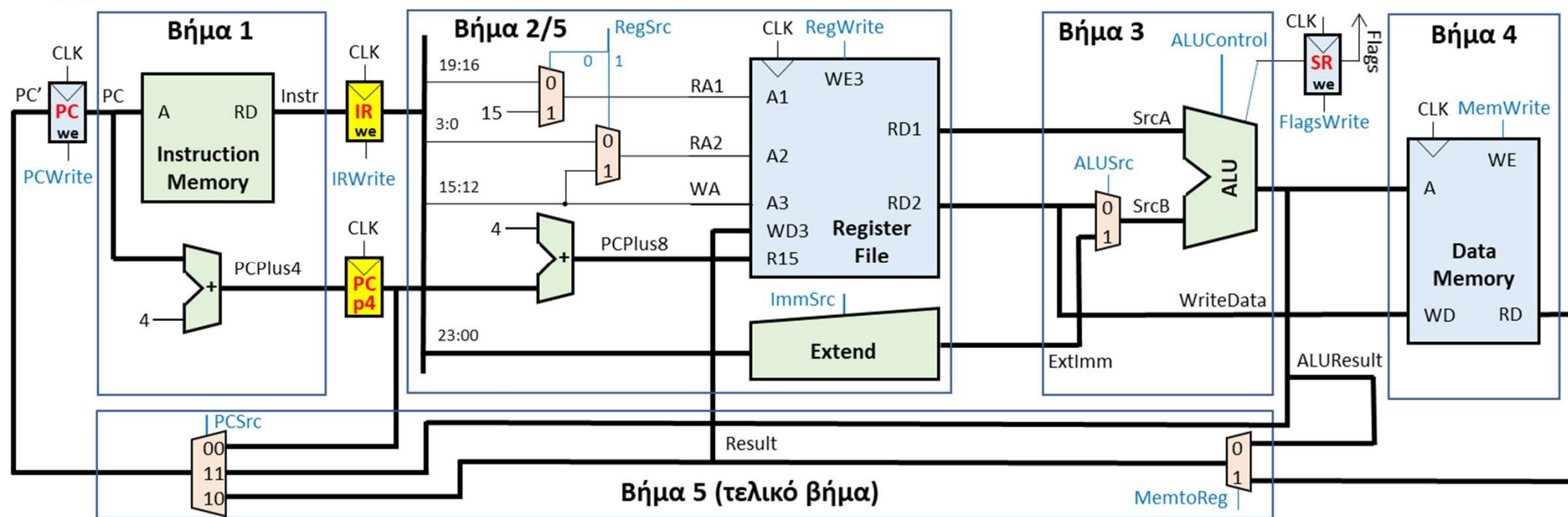
- Κατά τη σχεδίαση προβαίνουμε σε:
 - μελέτη διαδρομής δεδομένων ανά βήμα εκτέλεσης της εντολής
 - μελέτη μονάδας ελέγχου
 - ανάλυση επιδόσεων
- Ξεκινάμε από τη διαδρομή δεδομένων του επεξεργαστή ενός κύκλου, όπου οριοθετούμε τα βήματα από 1 μέχρι 5:



Το Βήμα 2/5 αφορά στην ανάγνωση/εγγραφή του αρχείου καταχωρητών

Επεξεργαστής πολλών κύκλων: μελέτη διαδρομής δεδομένων

- Στη συνέχεια, τοποθετούμε τους μη αρχιτεκτονικούς καταχωρητές ανάμεσα στα υλοποιημένα βήματα 1 και 2:
 - **Instruction Register (IR)** στη θύρα ανάγνωσης (A/RD) της μνήμης εντολών με σήμα έγκρισης εγγραφής **IRWrite**
 - Κατά την εκτέλεση του Βήματος 1 είναι $IRWrite = 1$, αλλιώς $IRWrite = 0$
 - **PCPlus4 Register (PCp4)** στην έξοδο **PCPlus4** του αθροιστή

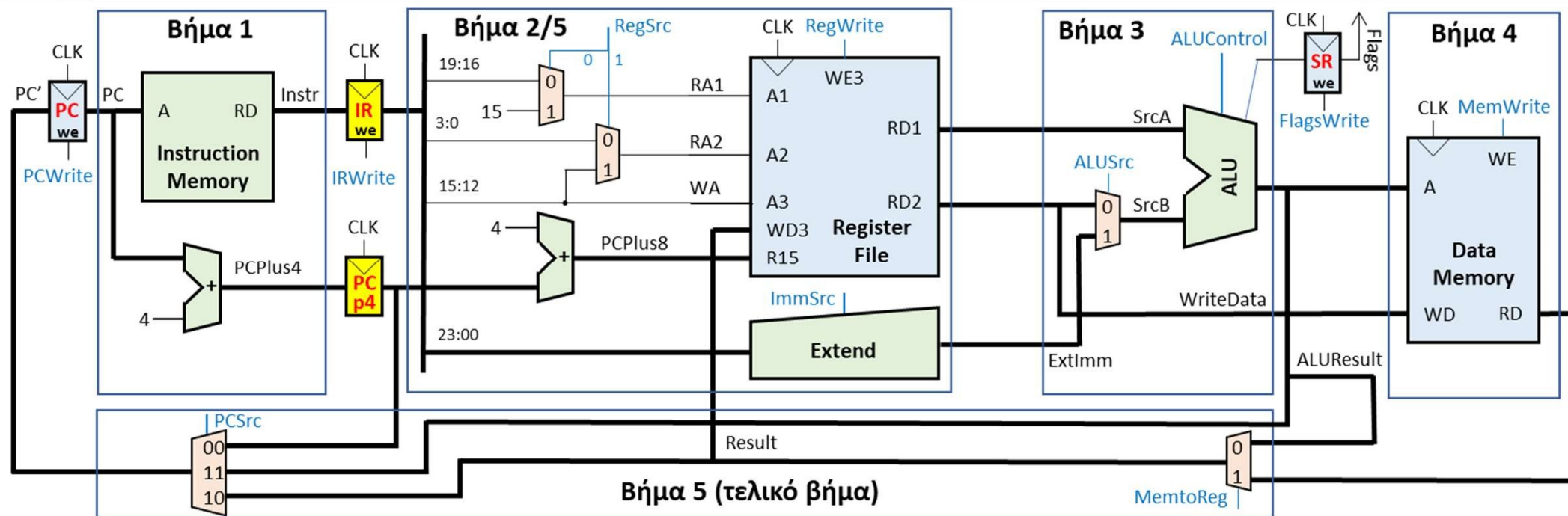


Ο **program counter (PC)** απαιτεί τη χρήση σήματος έγκρισης εγγραφής **PCWrite**, ώστε να διατηρεί το περιεχόμενο του κατά τη βηματική εκτέλεση της εντολής

Επεξεργαστής πολλών κύκλων: μελέτη διαδρομής δεδομένων

■ Βήμα 1 εκτέλεσης της εντολής:

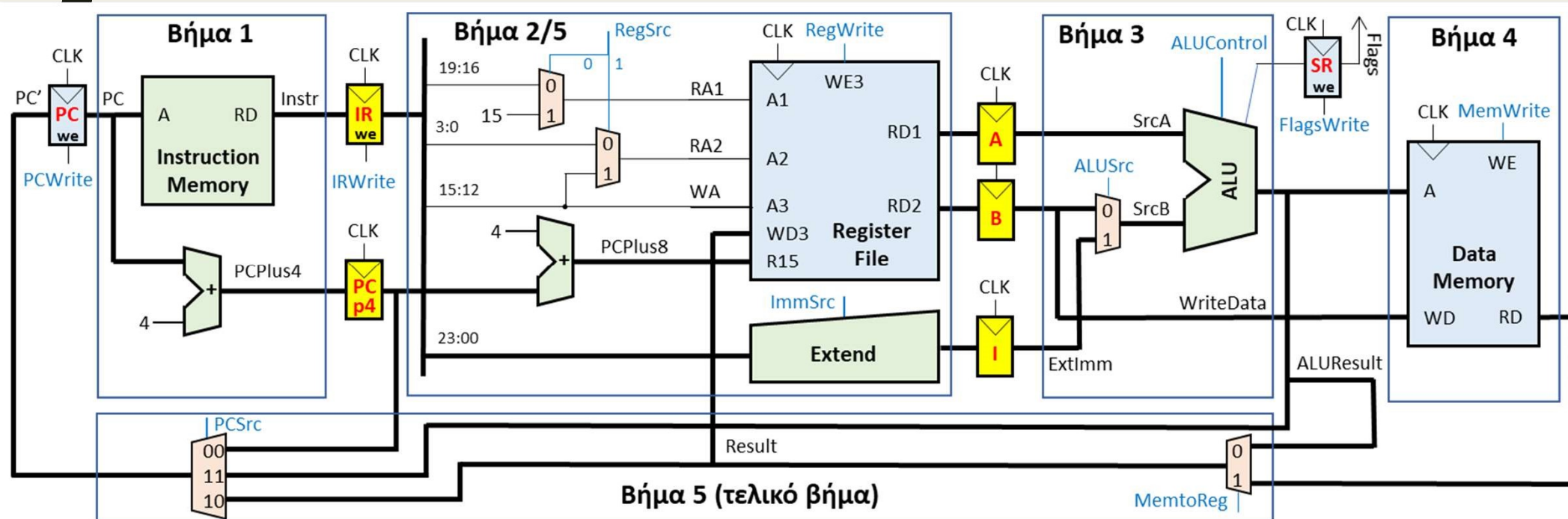
- Προσκόμιση εντολής από τη μνήμη εντολών (IM)
 - $IR = IM(PC)$ (απαιτείται $IRWrite = 1$ και $PCWrite = 0$)
- Υπολογισμός διεύθυνσης της επόμενης εντολής ($PC+4$)
 - $PCp4 = PC + 4$ (απαιτείται $PCWrite = 0$)



Ο **program counter (PC)** απαιτεί τη χρήση σήματος έγκρισης εγγραφής **PCWrite**, ώστε να διατηρεί το περιεχόμενο του κατά την εκτέλεση όλων των βημάτων της εντολής

Επεξεργαστής πολλών κύκλων: μελέτη διαδρομής δεδομένων

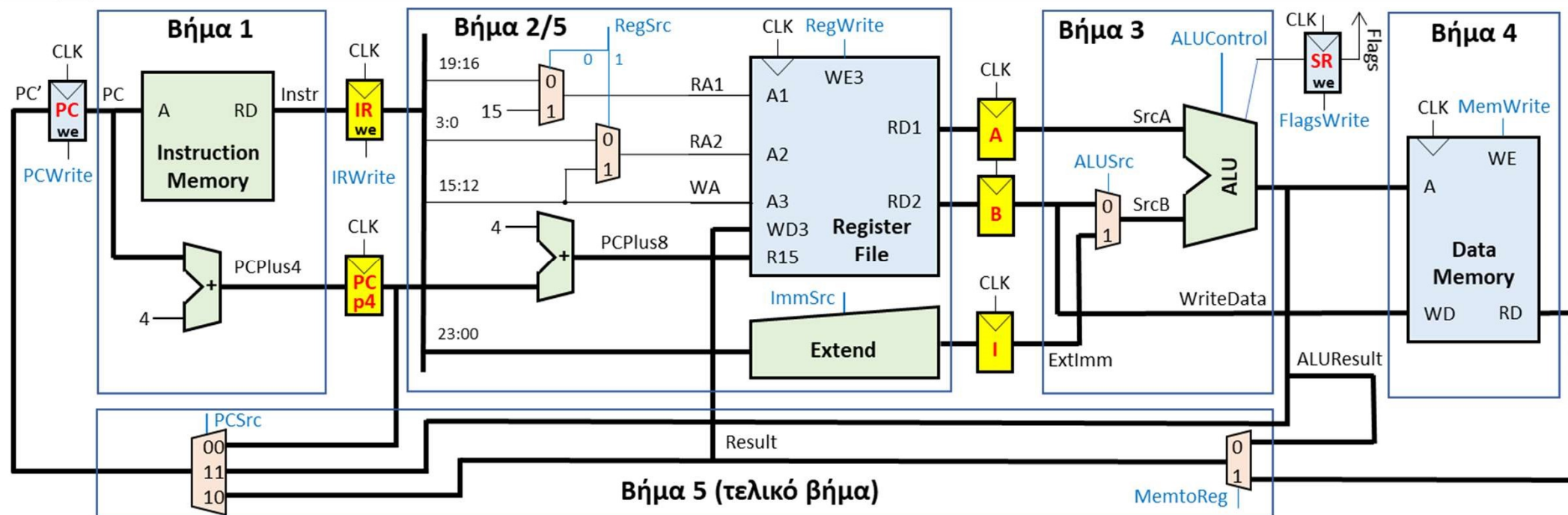
- Στη συνέχεια, τοποθετούμε τους μη αρχιτεκτονικούς καταχωρητές ανάμεσα στα υλοποιημένα βήματα 2 και 3:
 - **Register A (A)** στη θύρα ανάγνωσης (A1/RD1) του αρχείου καταχωρητών
 - **Register B (B)** στη θύρα ανάγνωσης (A2/RD2) του αρχείου καταχωρητών
 - **Register I (I)** στην έξοδο ExtImm της μονάδας Extend



Επεξεργαστής πολλών κύκλων: μελέτη διαδρομής δεδομένων

■ Βήμα 2 εκτέλεσης της εντολής:

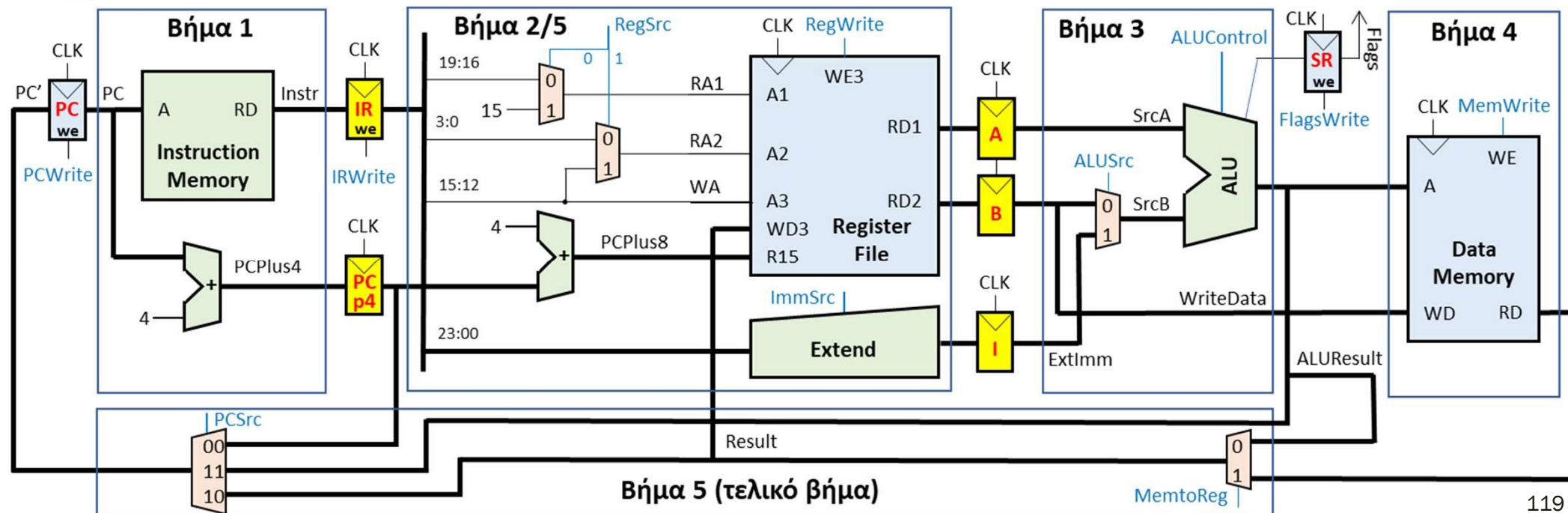
- Ανάγνωση του αρχείου καταχωρητών (R0-R14) από το RD1/RD2
 - $A = RF(IR[19:16]) = R_n$ (RegSrc[0] = 0, RegWrite = 0)
 - $B = RF(IR[3:0]) = R_m$ (RegSrc[1] = 0, RegWrite = 0, για εντολές DP-R)
 - $B = RF(IR[15:12]) = R_d$ (RegSrc[1] = 1, RegWrite = 0, για την εντολή STR)



Επεξεργαστής πολλών κύκλων: μελέτη διαδρομής δεδομένων

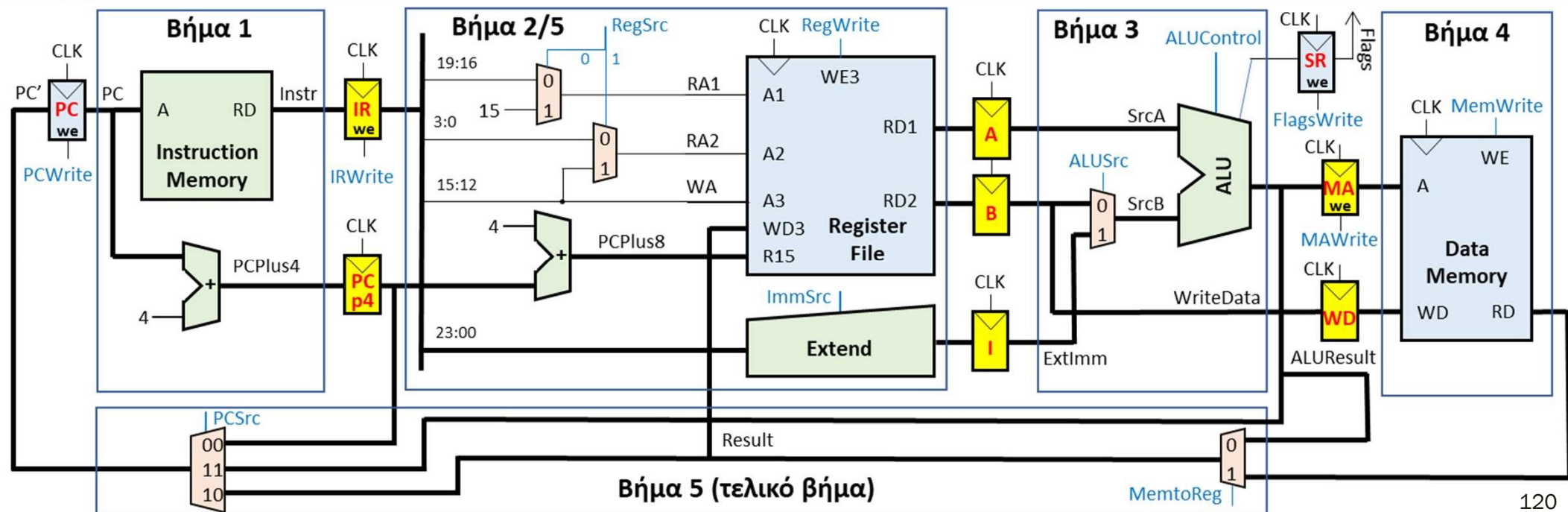
■ Βήμα 2 εκτέλεσης της εντολής:

- Ανάγνωση του μετρητή προγράμματος $R15$ ($PC + 8$) από το $RD1/RD2$
 - $A = RF(15) = PCp4 + 4 = PC + 8$ ($RA1 = Rn = 15$, $RegSrc[0] = 0$, $RegWrite = 0$)
 - $A = RF(15) = PCp4 + 4 = PC + 8$ ($RA1 = 15$, $RegSrc[0] = 1$, $RegWrite = 0$)
 - $B = RF(15) = PCp4 + 4 = PC + 8$ ($RA2 = Rm = 15$, $RegSrc[1] = 0$, $RegWrite = 0$)
 - $B = RF(15) = PCp4 + 4 = PC + 8$ ($RA2 = Rd = 15$, $RegSrc[1] = 1$, $RegWrite = 0$)
- Επέκταση πρόσημου ή μηδενός
 - $I = \text{zero_extend}(IR[11:0])$ ($IR[11:0] = \text{Imm8}/\text{Imm12}$ ($\text{rot} = 0$), $\text{ImmSrc} = 0$))
 - $I = \text{sign_extend}(IR[23:0] \times 4)$ ($IR[23:0] = \text{Imm24}$, $\text{ImmSrc} = 1$)



Επεξεργαστής πολλών κύκλων: μελέτη διαδρομής δεδομένων

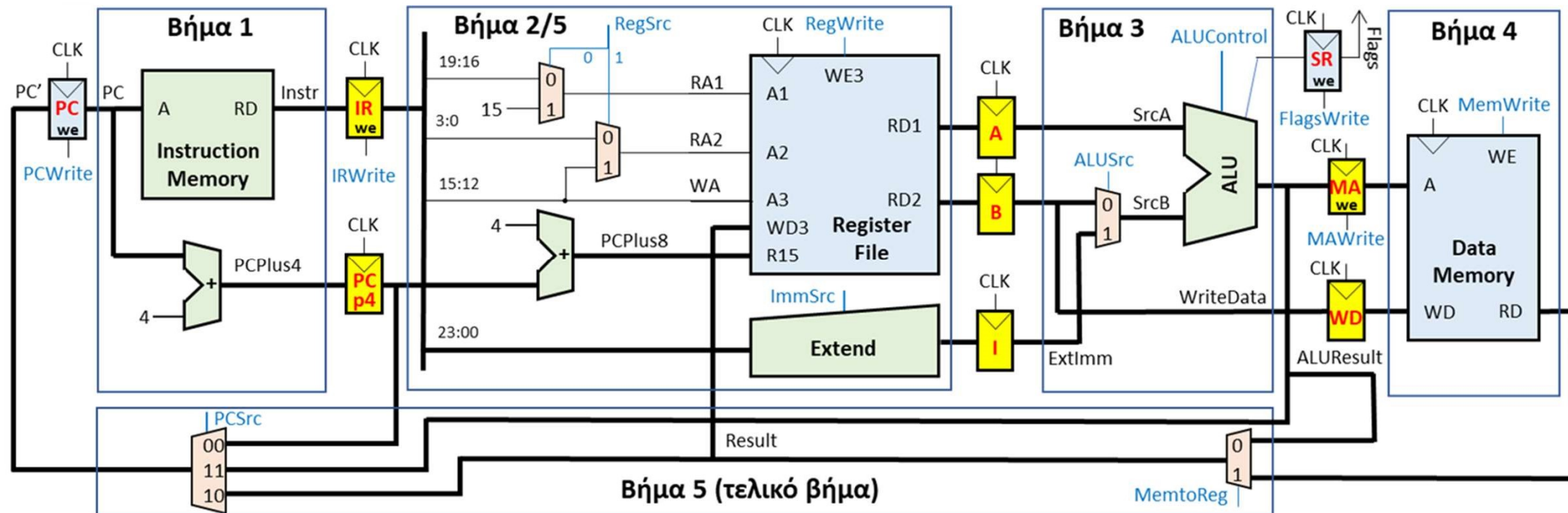
- Στη συνέχεια, τοποθετούμε τους μη αρχιτεκτονικούς καταχωρητές ανάμεσα στα υλοποιημένα βήματα 3 και 4 για προσπέλαση της μνήμης δεδομένων:
 - **Memory Address Register (MA)** στην έξοδο ALUResult της μονάδας ALU με σήμα έγκρισης εγγραφής **MAWrite**
 - Κατά την εκτέλεση του Βήματος 3 των εντολών μνήμης είναι $MAWrite = 1$, αλλιώς $MAWrite = 0$
 - **Memory Write Data Register (WD)** στην έξοδο του Register B (B)
 - Μπορεί και να **απαλειφθεί**.



Επεξεργαστής πολλών κύκλων: μελέτη διαδρομής δεδομένων

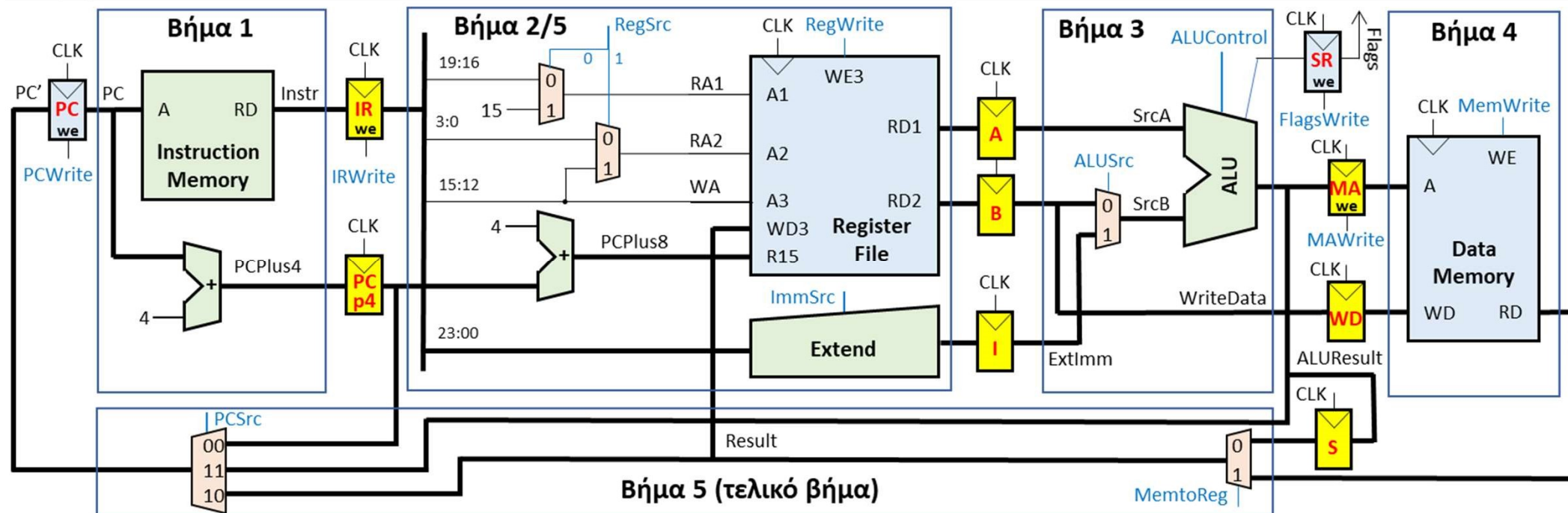
■ Βήμα 3 εκτέλεσης της εντολής (για εντολές LDR, STR):

- Υπολογισμός διεύθυνσης της μνήμης δεδομένων (DM)
 - $MA = A + I$ ($ALUSrc = 1$, $ALUControl[1:0] = 00$, $MAWrite = 1$)
- Μεταφορά δεδομένων που γράφονται στη μνήμη δεδομένων
 - $WD = B$



Επεξεργαστής πολλών κύκλων: μελέτη διαδρομής δεδομένων

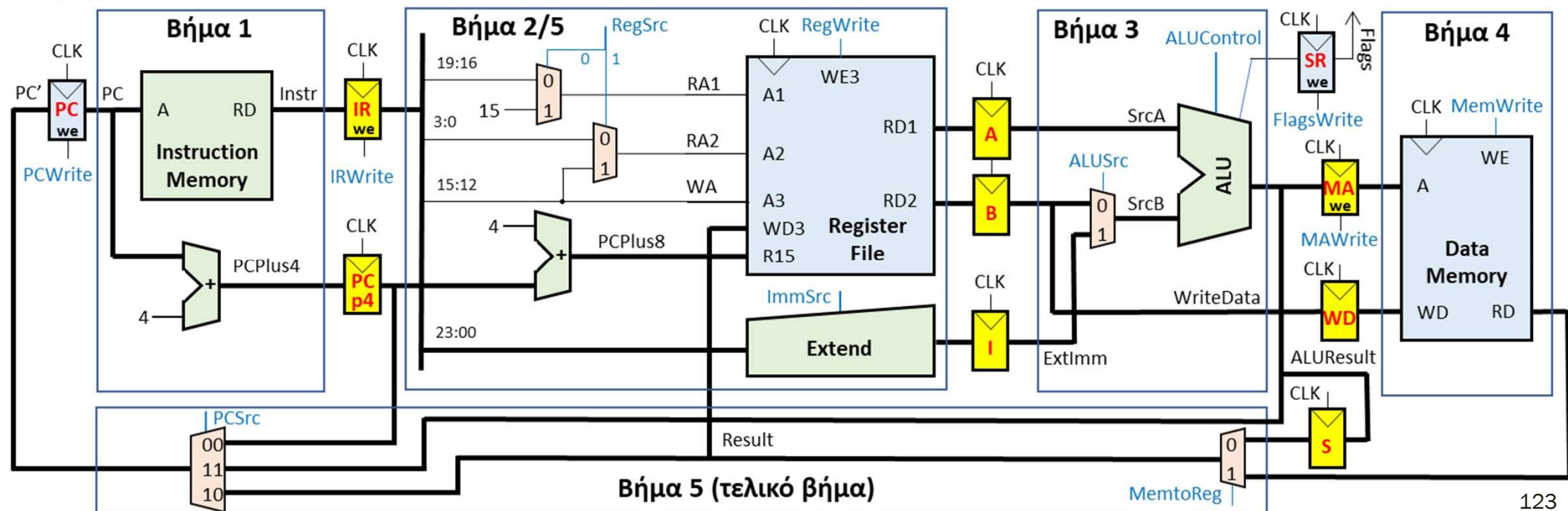
- Στη συνέχεια, τοποθετούμε τους μη αρχιτεκτονικούς καταχωρητές ανάμεσα στα υλοποιημένα βήματα 3 και 5:
 - **Register S (S)** στην έξοδο *ALUResult* της μονάδας *ALU*
 - Ο καταχωρητής **S παρακάμπτεται** κατά την εκτέλεση της εντολής **B**



Επεξεργαστής πολλών κύκλων: μελέτη διαδρομής δεδομένων

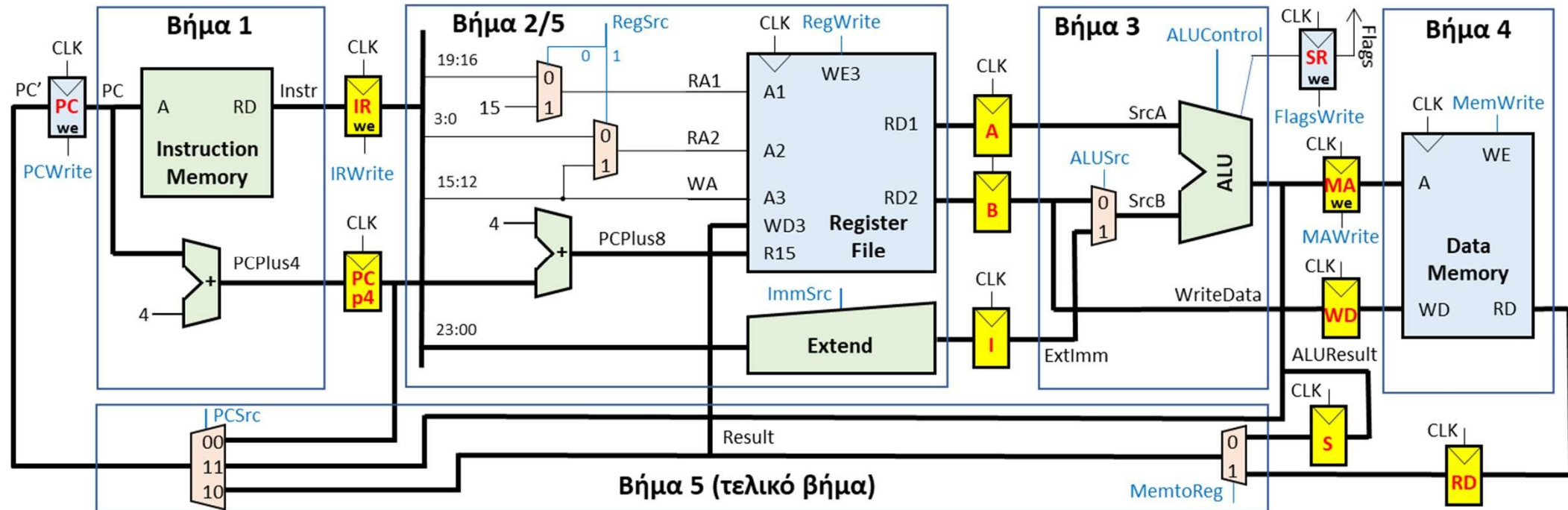
■ Βήμα 3 εκτέλεσης της εντολής (για εντολές DP και B):

- Εκτέλεση πράξης στη μονάδα ALU με άμεση διευθυνσιοδότηση ή διευθυνσιοδότηση καταχωρητή
 - $S = A +/- \text{and/or } I$ (ALUSrc = 1, ALUControl[1:0] = 00/01/10/11, MAWrite = 0)
 - $S = A +/- \text{and/or } B$ (ALUSrc = 0, ALUControl[1:0] = 00/01/10/11, MAWrite = 0)
- Υπολογισμός διεύθυνσης της μνήμης εντολών (IM) ($PC + 8 + imm24 \times 4$)
 - $S = A + I$ (ALUSrc = 1, ALUControl[1:0] = 00, MAWrite = 0)
- Ενημέρωση του καταχωρητή κατάστασης
 - $SR = (V, C, N, Z)$ (FlagsWrite = 1)



Επεξεργαστής πολλών κύκλων: μελέτη διαδρομής δεδομένων

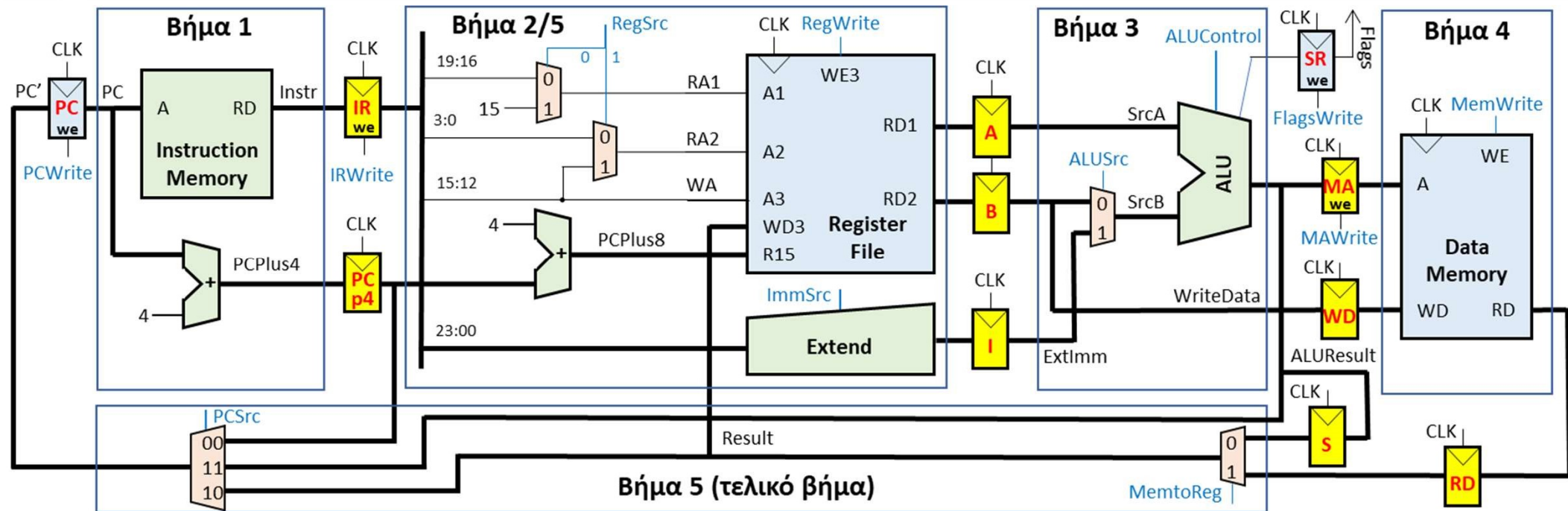
- Στη συνέχεια, τοποθετούμε τους μη αρχιτεκτονικούς καταχωρητές ανάμεσα στα υλοποιημένα βήματα 4 και 5:
 - *Memory Read Data Register (RD)* στη θύρα ανάγνωσης (A/RD) της μνήμης δεδομένων



Επεξεργαστής πολλών κύκλων: μελέτη διαδρομής δεδομένων

■ Βήμα 4 εκτέλεσης της εντολής (για εντολές LDR, STR):

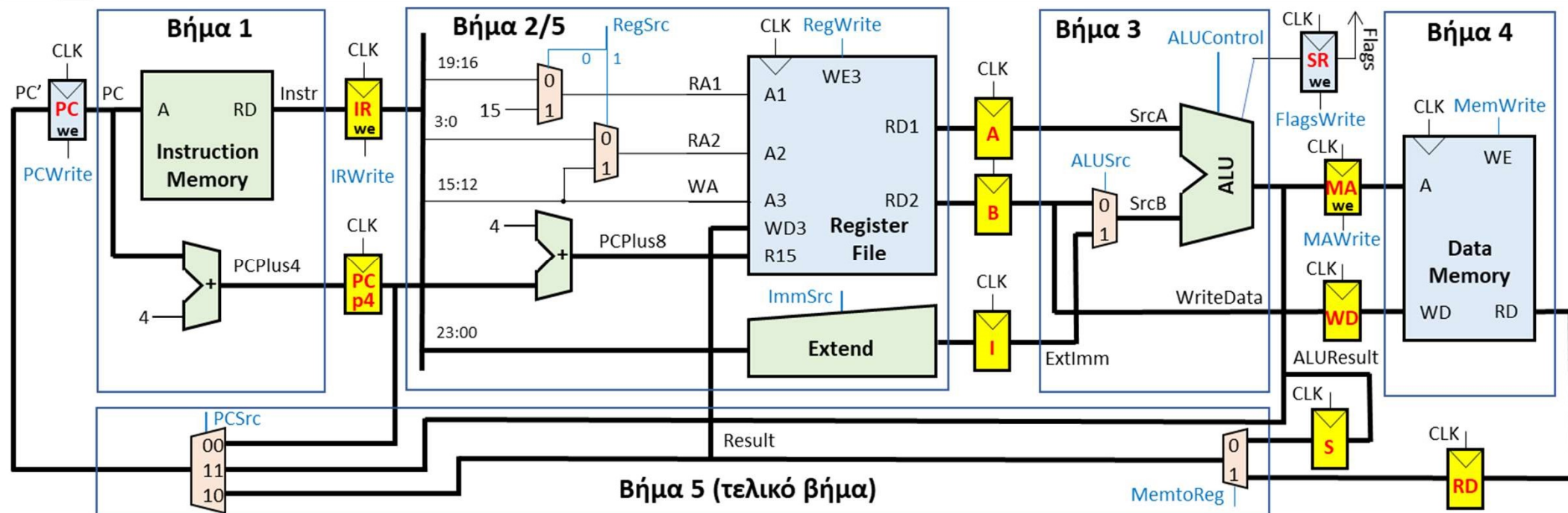
- Ανάγνωση μνήμης δεδομένων (DM)
 - $RD = DM(MA)$ (MemWrite = 0)
- Εγγραφή μνήμης δεδομένων (DM)
 - $DM(MA) = WD$ (MemWrite = 1)



Ανάλογα με την υλοποίηση της μνήμης δεδομένων (Block RAM) ο καταχωρητής RD μπορεί να είναι ήδη ενσωματωμένος σε αυτή και να μην απαιτείται προσθήκη νέου

Επεξεργαστής πολλών κύκλων: μελέτη διαδρομής δεδομένων

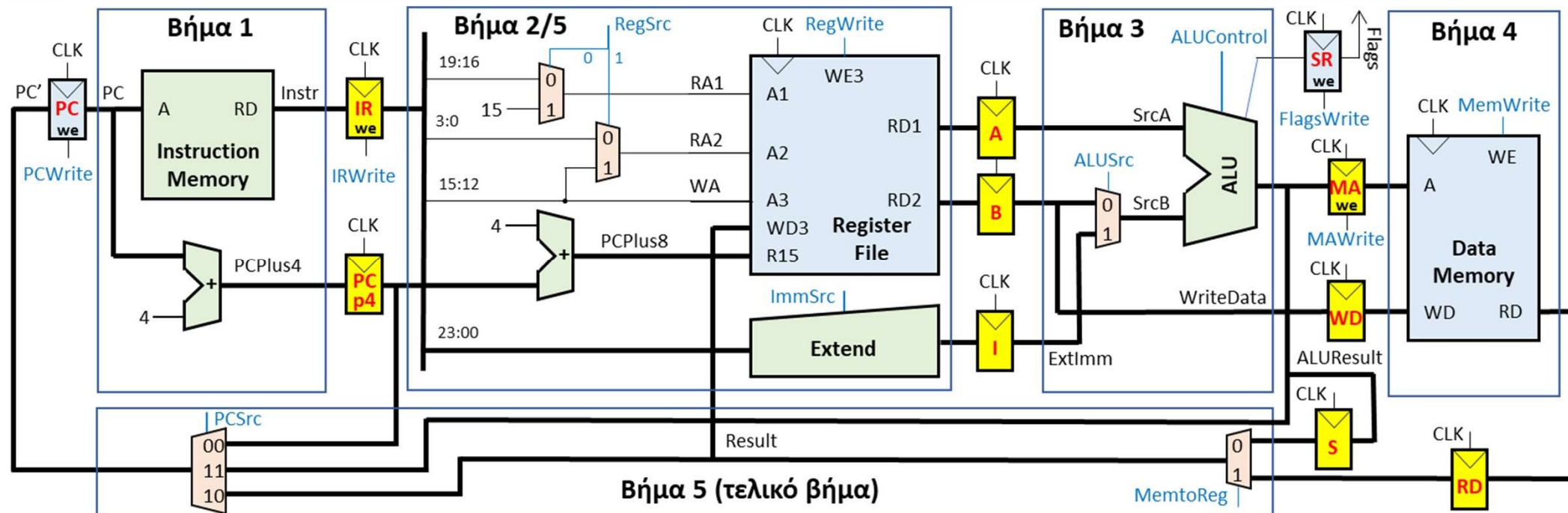
- Τέλος, τροποποιούμε τον αρχιτεκτονικό καταχωρητή program counter, ώστε να υποστηρίζει το σήμα έγκρισης εγγραφής **PCWrite**
 - κατά την εκτέλεση του Βήματος 5 είναι $PCWrite = 1$, αλλιώς $PCWrite = 0$
- Επίσης, τροποποιούμε τον πολυπλέκτη επιλογής διεύθυνσης επόμενης εντολής από 2 σε 1 σε 3 σε 1
 - με αντίστοιχη αύξηση του σήματος επιλογής $PCSrc$ σε 2 bit (**$PCSrc[1:0]$**)



Επεξεργαστής πολλών κύκλων: μελέτη διαδρομής δεδομένων

■ Βήμα 5 εκτέλεσης της εντολής:

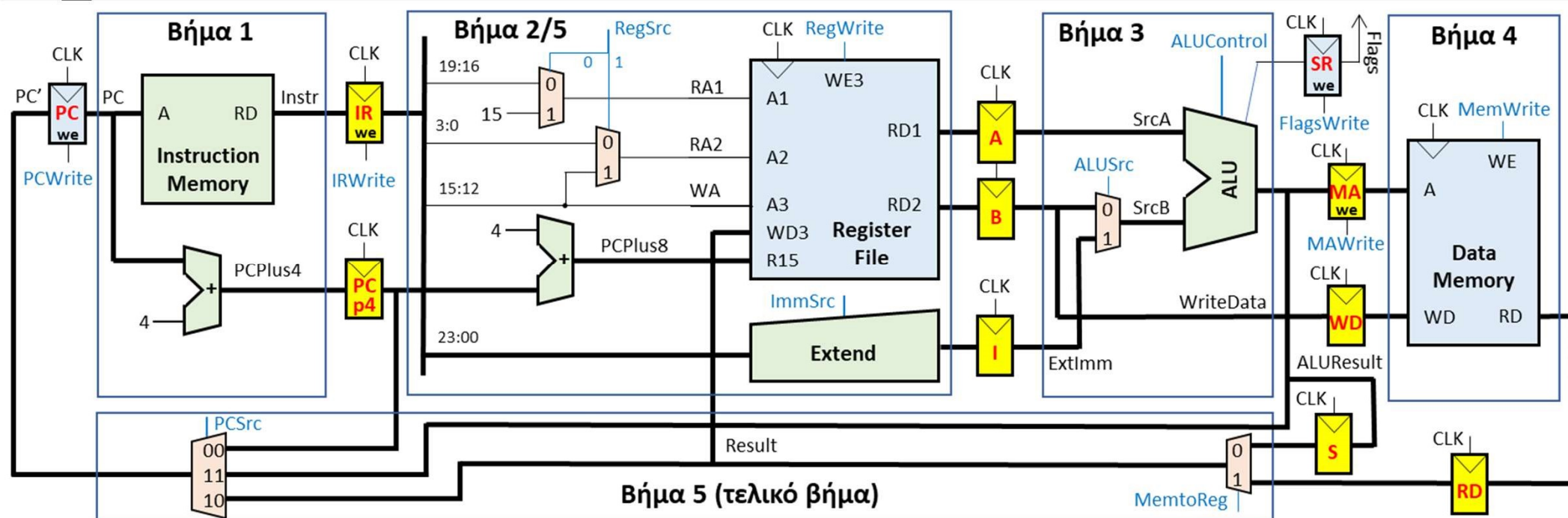
- Ετεροχρονισμένη εγγραφή στο αρχείο καταχωρητών (R0-R14)
 - $Rd = RF(IR[15:12]) = S$ (για εντολές DP, $RegWrite = 1$, $MemtoReg = 0$)
 - $Rd = RF(IR[15:12]) = RD$ (για την εντολή LDR, $RegWrite = 1$, $MemtoReg = 1$)
- Ετεροχρονισμένη εγγραφή στον μετρητή προγράμματος PC (R15)
 - $PC = S$ (DP, $PCWrite = 1$, $Rd = R15$, $MemtoReg = 0$, $PCSrc[1:0] = 10$)
 - $PC = RD$ (LDR, $PCWrite = 1$, $Rd = R15$, $MemtoReg = 1$, $PCSrc[1:0] = 10$)



Επεξεργαστής πολλών κύκλων: μελέτη διαδρομής δεδομένων

■ Βήμα 5 εκτέλεσης της εντολής:

- Επιλογή της διεύθυνσης της επόμενης εντολής που θα εκτελεσθεί, όταν δεν υπάρχει εγγραφή στον μετρητή προγράμματος PC (R15)
 - $PC = PCp4 = PC + 4$
(για εντολές DP και μνήμης, $PCWrite = 1$, $PCSrc[1:0] = 00$)
 - $PC = ALUResult = PC + 8 + imm24 \times 4$
(για την εντολή B, $PCWrite = 1$, $PCSrc[1:0] = 11$)



Επεξεργαστής πολλών κύκλων: μελέτη μονάδας ελέγχου

- Η **μονάδα ελέγχου** (control unit) παράγει τα κατάλληλα σήματα ελέγχου για τον χρονισμό του επεξεργαστή με τη χρήση:
 - συνδυαστικής λογικής για τα σήματα ελέγχου που παραμένουν σταθερά κατά την εκτέλεση της εντολής
 - μηχανής πεπερασμένων καταστάσεων για τα σήματα **έγκρισης εγγραφής** και το σήμα επιλογής **PCSrc**
- Είσοδοι της μονάδας ελέγχου (όπως και στον επεξεργαστή ενός κύκλου):
 - πεδίο **cond** του IR ($Instr_{31:28}$) για υποστήριξη εντολών υπό συνθήκη
 - πεδίο **op** του IR εντολής ($Instr_{27:26}$) για τον προσδιορισμό του τύπου της εντολής (επεξεργασίας δεδομένων, μνήμης, διακλάδωσης)
 - πεδίο **funct** του IR ($Instr_{25:20}$) για τη δήλωση της λειτουργίας/πράξης
 - σημαίες συνθήκης **flags** (**N, Z, C, V**) του SR
 - πεδίο **Rd** του IR ($Instr_{15:12}$) για τη δήλωση του μετρητή προγράμματος PC ως καταχωρητή προορισμού ($Rd = "1111"$)
- Η μονάδα ελέγχου ελέγχει εάν ικανοποιείται η συνθήκη που ορίζεται στο πεδίο **cond** του IR με βάση τις τρέχουσες σημαίες συνθήκης
 - παράγει το εσωτερικό σήμα **CondEx_in** που εγκρίνει την εκτέλεση της εντολής

Επεξεργαστής πολλών κύκλων: μελέτη μονάδας ελέγχου

- Έξοδοι της μονάδας ελέγχου που παραμένουν σταθερές κατά την εκτέλεση της εντολής:
 - τα σήματα επιλογής πολυπλεκτών
 - **RegSrc[1:0]** (ανάγνωση από το αρχείο καταχωρητών)
 - **ALUSrc** (μονάδα ALU)
 - **MemtoReg** (ετεροχρονισμένη εγγραφή στο αρχείο καταχωρητών και στον μετρητή προγράμματος)
 - τα σήματα ελέγχου λογικής
 - **ALUControl[1:0]** (μονάδα ALU)
 - **ImmSrc** (μονάδα Extend)
 - το εσωτερικό σήμα **CondEx_in** που ενεργοποιείται όταν ικανοποιείται η συνθήκη εκτέλεσης της εντολής
- Παράγονται με τα ίδια συνδυαστικά κυκλώματα που μελετήσαμε στον επεξεργαστή ενός κύκλου

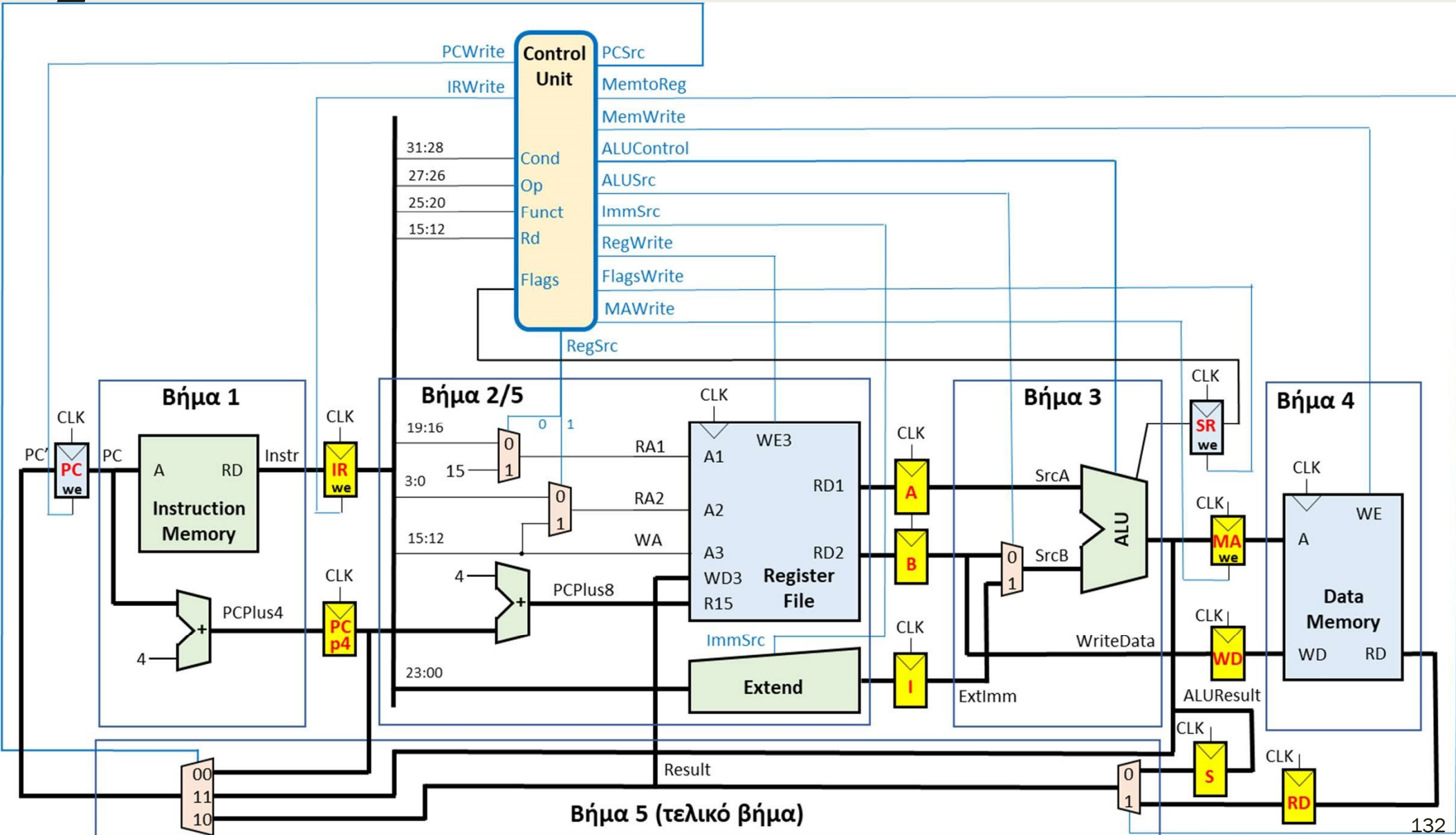
Επεξεργαστής πολλών κύκλων: μελέτη μονάδας ελέγχου

- Έξοδοι της μονάδας ελέγχου που ενεργοποιούνται (παίρνουν την τιμή 1) σε συγκεκριμένο Βήμα (κύκλο) κατά την εκτέλεση της εντολής:
 - τα σήματα έγκρισης εγγραφής και το βήμα στο οποίο ενεργοποιούνται
 - **PCWrite** (μετρητής προγράμματος - Βήμα 5)
 - **IRWrite** (καταχωρητής εντολών - Βήμα 1)
 - **RegWrite** (αρχείο καταχωρητών - Βήμα 5)
 - **FlagsWrite** (καταχωρητής καταστάσεων - Βήμα 5)
 - **MAWrite** (καταχωρητής διευθύνσεων μνήμης δεδομένων - Βήμα 3)
 - **MemWrite** (μνήμη δεδομένων - Βήμα 4)
 - το σήμα επιλογής διεύθυνσης επόμενης εντολής και το βήμα στο οποίο ενεργοποιείται
 - **PCSrc[1:0]** (μετρητής προγράμματος - Βήμα 5)
- Παράγονται από μηχανή πεπερασμένων καταστάσεων (FSM)

Με κόκκινο τα σήματα που απαιτούνται στον επεξεργαστή πολλών κύκλων

Επεξεργαστής πολλών κύκλων: μελέτη μονάδας ελέγχου

- Ολοκληρωμένος επεξεργαστής πολλών κύκλων



Επεξεργαστής πολλών κύκλων: μελέτη μονάδας ελέγχου

- Στη συνέχεια θα σχεδιάσουμε τη μονάδα ελέγχου που απαρτίζεται από τις ακόλουθες υπομονάδες:

1. Αποκωδικοποιητής εντολής (InstrDec)

- η υπομονάδα που αποκωδικοποιεί τα πεδία **op** και **funct** του IR (*Instr27:20*) και παράγει τα σήματα επιλογής πολυπλεκτών που ρυθμίζουν τη ροή δεδομένων και τα σήματα ελέγχου συνδυαστικής λογικής στη διαδρομή δεδομένων

2. Λογική ελέγχου συνθήκης (CONDLogic)

- η υπομονάδα που ελέγχει εάν ικανοποιείται η συνθήκη που ορίζεται στο πεδίο **cond** του IR με βάση τις τρέχουσες σημαίες συνθήκης **flags** του SR, ώστε να εγκριθεί η εκτέλεση της εντολής

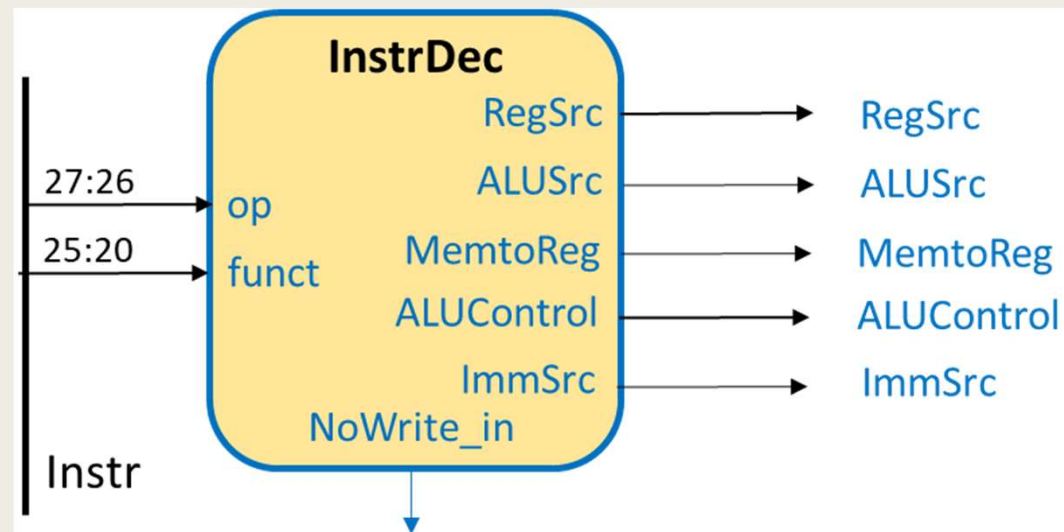
3. Μηχανή πεπερασμένων καταστάσεων (FSM)

- η υπομονάδα που παράγει τα σήματα έγκρισης εγγραφής
- η υπομονάδα που ενεργοποιεί το σήμα επιλογής διεύθυνσης επόμενης εντολής **PCSrc[1:0]** εάν απαιτείται εγγραφή στον R15 (PC) ή εκτελείται εντολή διακλάδωσης

Επεξεργαστής πολλών κύκλων: μελέτη μονάδας ελέγχου

1. Αποκωδικοποιητής εντολής (InstrDec)

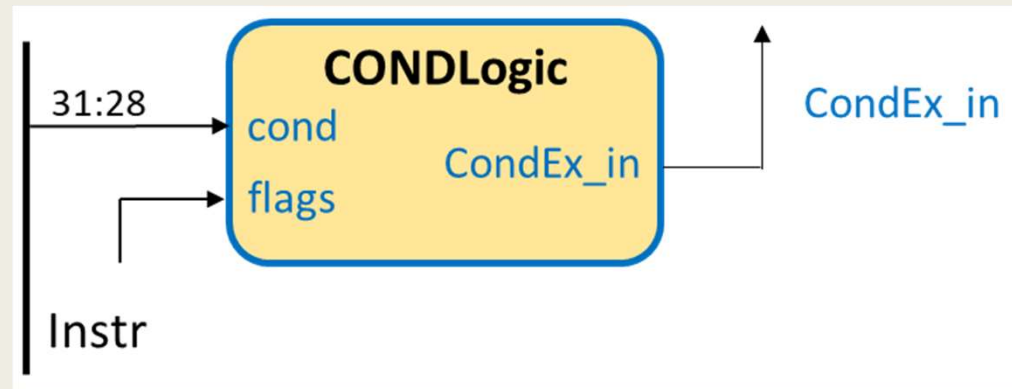
- είσοδοι: το πεδίο **op** του IR ($Instr_{27:26}$) και το πεδίο **funct** του IR ($Instr_{25:20}$)
- έξοδοι: τα σήματα επιλογής πολυπλεκτών **RegSrc[1:0]**, **ALUSrc** και **MemtoReg**, τα σήματα ελέγχου λογικής **ALUControl[1:0]** και **ImmSrc**
- εσωτερικό σήμα εξόδου: το σήμα **NoWrite_in** που εμποδίζει την εγγραφή στο αρχείο καταχωρητών, όταν ενεργοποιείται
 - ενεργοποιείται κατά την εκτέλεση της εντολής CMP
- Όλες οι έξοδοι παράγονται κατά την εκτέλεση του βήματος 2
 - το σήμα επιλογής **RegSrc[1:0]** είναι στην **κρίσιμη διαδρομή (critical path)** που ενεργοποιείται κατά την εκτέλεση του βήματος 2



Επεξεργαστής πολλών κύκλων: μελέτη μονάδας ελέγχου

2. Λογική ελέγχου συνθήκης (CONDLogic)

- ελέγχει εάν ικανοποιείται η συνθήκη που ορίζεται στο πεδίο **cond** του IR με βάση τις τρέχουσες σημαίες συνθήκης **flags**
- είσοδοι: το πεδίο **cond** του IR ($Instr_{31:28}$) και η έξοδος **flags** του SR
- εσωτερικό σήμα εξόδου: το σήμα **CondEx_in** που εγκρίνει την εκτέλεση της εντολής, όταν ενεργοποιείται



Προσοχή! Για να λειτουργεί σωστά η λογική ελέγχου συνθήκης θα πρέπει οι σημαίες να ενημερώνονται στο τέλος της εντολής (οι σημαίες παράγονται στο Βήμα 3, αλλά ο καταχωρητής καταστάσεων ενημερώνεται στο Βήμα 5)

Επεξεργαστής πολλών κύκλων: μελέτη μονάδας ελέγχου

3. Μηχανή πεπερασμένων καταστάσεων (FSM)

- είσοδοι:
 - το πεδίο **op** του IR ($\text{Instr}_{27:26}$)
 - το πεδίο **S** (για εντολές DP) ή **L** (για εντολές μνήμης) του IR (Instr_{20})
 - το πεδίο **Rd** του IR ($\text{Instr}_{15:12}$)
 - τα εσωτερικά σήματα **NoWrite_in** και **CondEx_in** που παράγονται κατά την αποκωδικοποίηση της εντολής (βήμα 2)
- έξοδοι:
 - τα σήματα έγκρισης εγγραφής (**PCWrite**, **IRWrite**, **RegWrite**, **FlagsWrite**, **MAWrite** και **MemWrite**)
 - το σήμα επιλογής διεύθυνσης επόμενης εντολής **PCSrc[1:0]**
- εάν **CondEx_in = 0**, τότε ενεργοποιείται πρόωρα το **PCWrite**, ενώ όλα τα υπόλοιπα σήματα έχουν την τιμή 0, και τερματίζεται πρόωρα η εκτέλεση της εντολής
 - στον επόμενο κύκλο εκτελείται το Βήμα 1 της αμέσως επόμενης εντολής στη διεύθυνση PC+4

Επεξεργαστής πολλών κύκλων: μελέτη μονάδας ελέγχου

- Βήματα (κύκλοι) εκτέλεσης των εντολών
 - η διαδρομή δεδομένων υλοποιεί τα ακόλουθα πέντε βήματα:
 1. Προσκόμιση εντολής και υπολογισμός επόμενης διεύθυνσης (PC+4)
 2. Αποκωδικοποίηση εντολής και ανάγνωση αρχείου καταχωρητών
 3. Εκτέλεση πράξεων στη μονάδα ALU
 4. Ανάγνωση ή εγγραφή στη μνήμη δεδομένων
 5. Ετεροχρονισμένη εγγραφή στο αρχείο καταχωρητών και επιλογή της διεύθυνσης της επόμενης εντολής
 - κατά την εκτέλεση μίας εντολής πιθανώς να μην απαιτούνται όλα τα βήματα ή να εκτελούνται παράλληλα στον ίδιο κύκλο, όσα δεν δημιουργούν εξαρτήσεις

Κύκλοι	LDR	STR	DP	CMP	B	No executed
1	1	1	1	1	1	1
2	2	2	2	2	2	2
3	3	3	3	3, 5	3, 5	5
4	4	4, 5	5			
5	5					

Επεξεργαστής πολλών κύκλων: μελέτη μονάδας ελέγχου – η εντολή LDR

3. Μηχανή πεπερασμένων καταστάσεων (FSM) τύπου Moore

- προσδιορισμός των απαιτούμενων καταστάσεων με βάση τις τιμές των σημάτων ελέγχου, ανά βήμα εκτέλεσης της εντολής

Βήμα	Curr. State	Next State	op	S / L	Rd	NoWrite_in	Cond Ex_in	IR Write	Reg Write	MA Write	Mem Write	Flags Write	PC Src	PC Write
1	S0	S1	XX	X	X	X	X	1	0	0	0	0	00	0
2	S1	S2a	01	X	X	X	1	0	0	0	0	0	00	0
2	S1	S4c	XX	X	X	X	0	0	0	0	0	0	00	0
3	S2a	S3	XX	1	X	X	X	0	0	1	0	0	00	0
4	S3	S4a	XX	X	Not 15	X	X	0	0	0	0	0	00	0
4	S3	S4b	XX	X	15	X	X	0	0	0	0	0	00	0
5	S4a	S0	XX	X	XX	X	X	0	1	0	0	0	00	1
5	S4b	S0	XX	X	XX	X	X	0	0	0	0	0	10	1
5	S4c	S0	XX	X	XX	X	X	0	0	0	0	0	00	1

Επεξεργαστής πολλών κύκλων: μελέτη μονάδας ελέγχου – η εντολή STR

3. Μηχανή πεπερασμένων καταστάσεων (FSM) τύπου Moore

- προσδιορισμός των απαιτούμενων καταστάσεων με βάση τις τιμές των σημάτων ελέγχου, ανά βήμα εκτέλεσης της εντολής

Βήμα	Curr. State	Next State	op	S / L	Rd	NoWrite_in	Cond Ex_in	IR Write	Reg Write	MA Write	Mem Write	Flags Write	PC Src	PC Write
1	S0	S1	XX	X	X	X	X	1	0	0	0	0	00	0
2	S1	S2a	01	X	X	X	1	0	0	0	0	0	00	0
2	S1	S4c	XX	X	X	X	0	0	0	0	0	0	00	0
3	S2a	S4d	XX	0	X	X	X	0	0	1	0	0	00	0
5	S4c	S0	XX	X	XX	X	X	0	0	0	0	0	00	1
4,5	S4d	S0	XX	X	XX	X	X	0	0	0	1	0	00	1

Επεξεργαστής πολλών κύκλων: μελέτη μονάδας ελέγχου – οι εντολές DP

3. Μηχανή πεπερασμένων καταστάσεων (FSM) τύπου Moore

- προσδιορισμός των απαιτούμενων καταστάσεων με βάση τις τιμές των σημάτων ελέγχου, ανά βήμα εκτέλεσης της εντολής

Βήμα	Curr. State	Next State	op	S / L	Rd	NoWrite_in	Cond Ex_in	IR Write	Reg Write	MA Write	Mem Write	Flags Write	PC Src	PC Write
1	S0	S1	XX	X	X	X	X	1	0	0	0	0	00	0
2	S1	S2b	00	X	X	0	1	0	0	0	0	0	00	0
2	S1	S4c	XX	X	X	X	0	0	0	0	0	0	00	0
3	S2b	S4a	XX	0	Not 15	X	X	0	0	0	0	0	00	0
3	S2b	S4b	XX	0	15	X	X	0	0	0	0	0	00	0
3	S2b	S4e	XX	1	Not 15	X	X	0	0	0	0	0	00	0
3	S2b	S4f	XX	1	15	X	X	0	0	0	0	0	00	0
5	S4a	S0	XX	X	XX	X	X	0	1	0	0	0	00	1
5	S4b	S0	XX	X	XX	X	X	0	0	0	0	0	10	1
5	S4c	S0	XX	X	XX	X	X	0	0	0	0	0	00	1
5	S4e	S0	XX	X	XX	X	X	0	1	0	0	1	00	1
5	S4f	S0	XX	X	XX	X	X	0	0	0	0	1	10	1 ⁴⁰

Επεξεργαστής πολλών κύκλων: μελέτη μονάδας ελέγχου – η εντολή CMP

3. Μηχανή πεπερασμένων καταστάσεων (FSM) τύπου Moore

- προσδιορισμός των απαιτούμενων καταστάσεων με βάση τις τιμές των σημάτων ελέγχου, ανά βήμα εκτέλεσης της εντολής
- Κατά την εκτέλεση της εντολής CMP ο αποκωδικοποιητής εντολών ενεργοποιεί το εσωτερικό σήμα `NoWrite_in`

Βήμα	Curr. State	Next State	op	S / L	Rd	NoWrite_in	Cond Ex_in	IR Write	Reg Write	MA Write	Mem Write	Flags Write	PC Src	PC Write
1	S0	S1	XX	X	X	X	X	1	0	0	0	0	00	0
2	S1	S4g	00	X	X	1	1	0	0	0	0	0	00	0
2	S1	S4c	XX	X	X	X	0	0	0	0	0	0	00	0
5	S4c	S0	XX	X	XX	X	X	0	0	0	0	0	00	1
3,5	S4g	S0	XX	X	XX	X	X	0	0	0	0	1	00	1

Επεξεργαστής πολλών κύκλων: μελέτη μονάδας ελέγχου – η εντολή B

3. Μηχανή πεπερασμένων καταστάσεων (FSM) τύπου Moore

- προσδιορισμός των απαιτούμενων καταστάσεων με βάση τις τιμές των σημάτων ελέγχου, ανά βήμα εκτέλεσης της εντολής

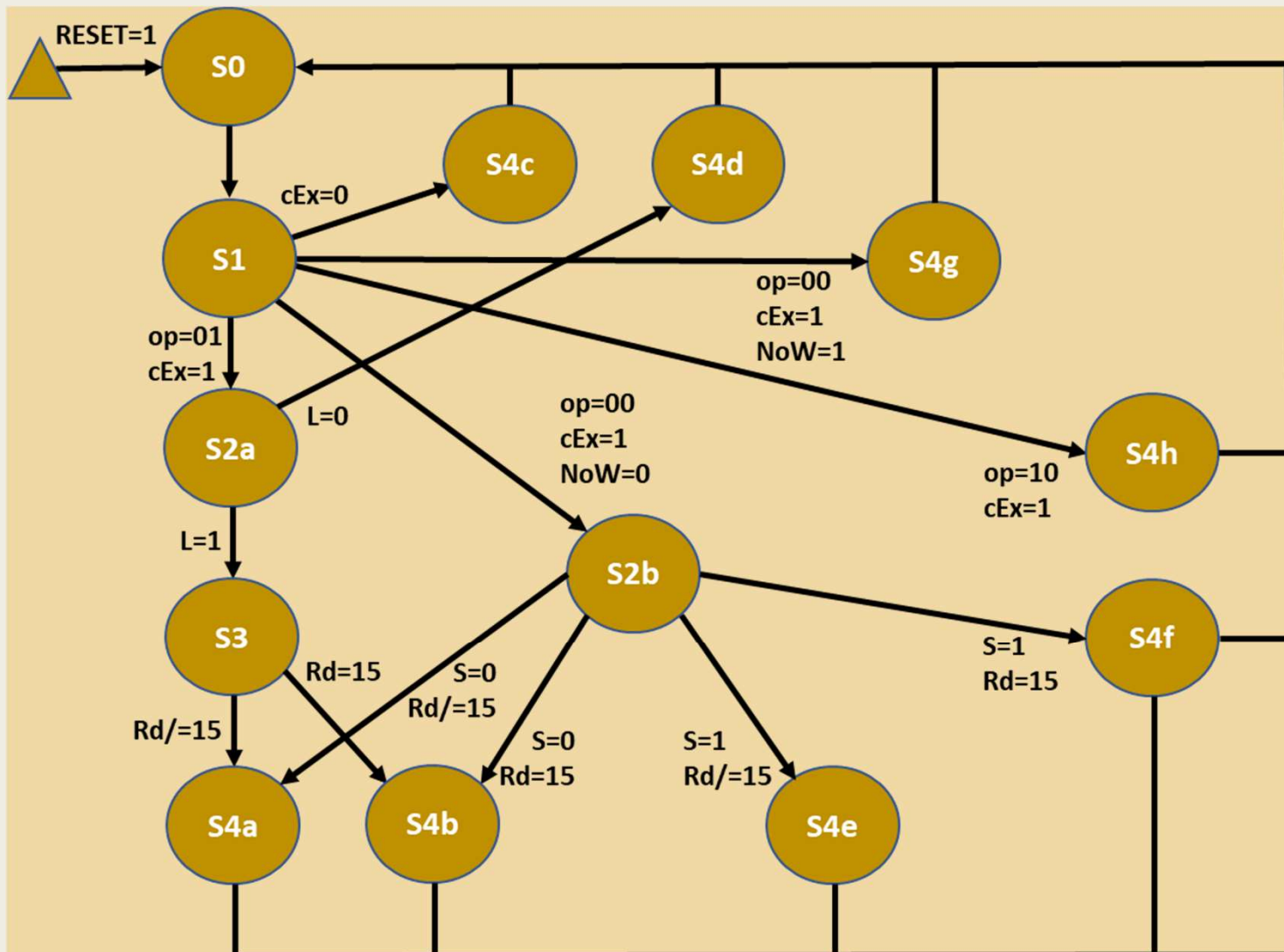
Βήμα	Curr. State	Next State	op	S / L	Rd	NoWrite_in	Cond Ex_in	IR Write	Reg Write	MA Write	Mem Write	Flags Write	PC Src	PC Write
1	S0	S1	XX	X	X	X	X	1	0	0	0	0	00	0
2	S1	S4h	10	X	X	X	1	0	0	0	0	0	00	0
2	S1	S4c	XX	X	X	X	0	0	0	0	0	0	00	0
5	S4c	S0	XX	X	XX	X	X	0	0	0	0	0	00	1
3,5	S4h	S0	XX	X	XX	X	X	0	0	0	0	0	11	1

Επεξεργαστής πολλών κύκλων: μελέτη μονάδας ελέγχου

3. Μηχανή πεπερασμένων καταστάσεων (FSM) τύπου Moore

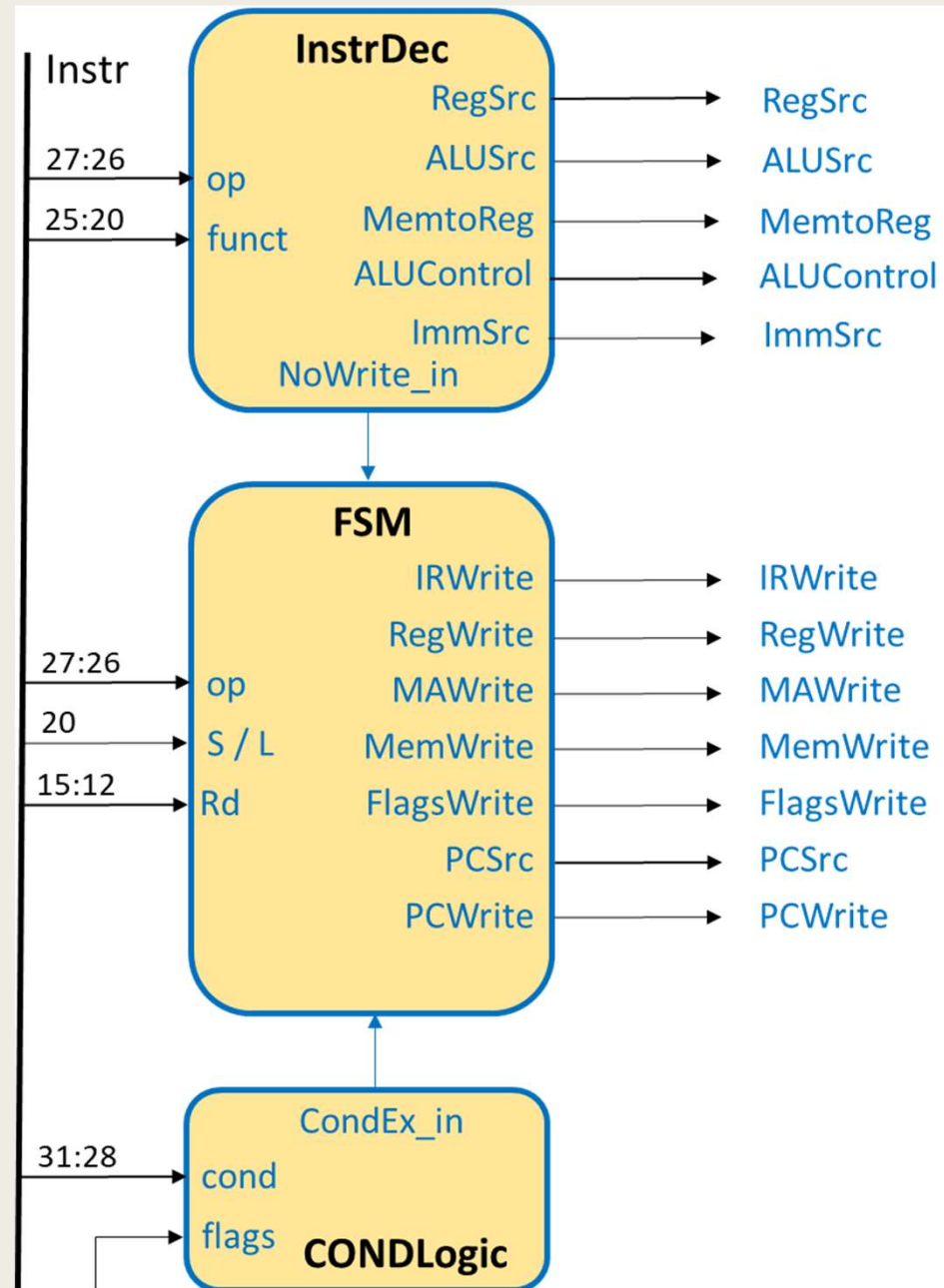
- διάγραμμα μεταβολής κατάστασης

- αναφέρονται μόνο οι συνθήκες εισόδου, που ορίζουν την επόμενη κατάσταση



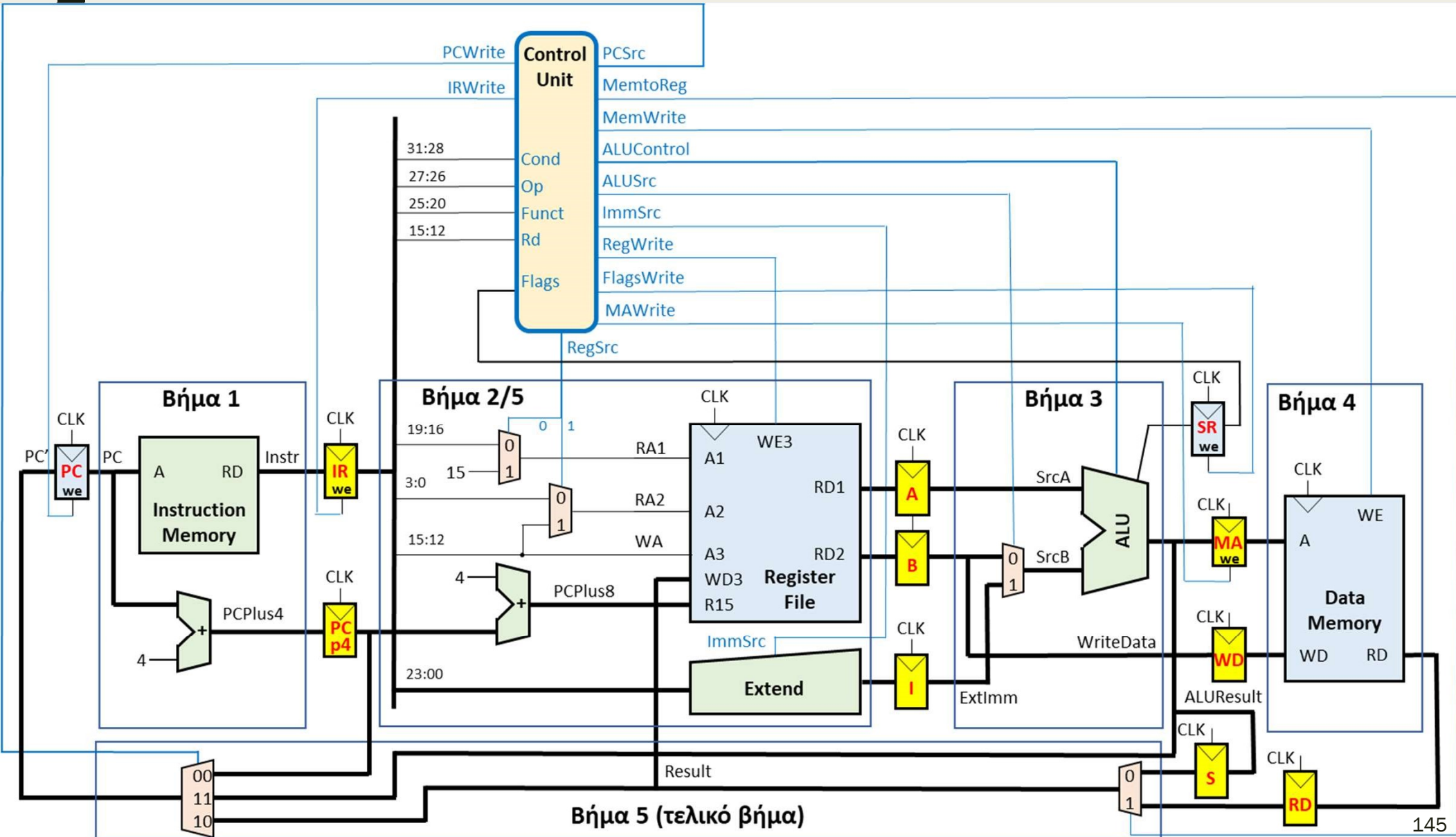
CondEx_in = cEx
NoWrite_in = NoW

Επεξεργαστής πολλών κύκλων: μελέτη μονάδας ελέγχου – Υπομονάδες



Επεξεργαστής πολλών κύκλων: μελέτη μονάδας ελέγχου

- Ολοκληρωμένος επεξεργαστής πολλών κύκλων



Ανάλυση επιδόσεων επεξεργαστών

■ Πώς επηρεάζεται η επίδοση

- $\text{CPU}_{\text{time}} = (\# \text{ instructions/program}) \times \text{CPI} \times T_c = \text{seconds/program}$

■ Μέσος αριθμός κύκλων ανά εντολή (cycles per instruction - CPI):

- μέσος αριθμός κύκλων ανά εντολή του ρολογιού που απαιτούνται για την εκτέλεση μιας εντολής ενός προγράμματος
- εξαρτάται από το μίγμα των εντολών σε ένα πρόγραμμα (πόσες εντολές είναι επεξεργασίας δεδομένων, μνήμης ή διακλάδωσης)

■ Εντολές ανά κύκλο (instructions per cycle - IPC):

- διεκπεραιωτική ικανότητα - το αντίστροφο του CPI

■ Περίοδος T_c του ρολογιού (clock period):

- το πλήθος των νανοδευτερολέπτων ανά κύκλο ρολογιού
- καθορίζεται από την **κρίσιμη διαδρομή (critical path)** μέσω της λογικής της εκάστοτε μικροαρχιτεκτονικής του επεξεργαστή
- ο τρόπος σχεδίασης της διαδρομής δεδομένων και των μονάδων που την απαρτίζουν επηρεάζουν την **κρίσιμη διαδρομή (critical path)** και συνεπώς την περίοδο του ρολογιού

Επεξεργαστής πολλών κύκλων: ανάλυση επιδόσεων

- Η τιμή του **CPI** στον επεξεργαστή πολλών κύκλων εξαρτάται από:
 - την ποσόστωση (%) των πιθανών κατηγοριών εντολών που εκτελούνται σε ένα συγκεκριμένο πρόγραμμα
 - από τον αριθμό των κύκλων που απαιτούνται κατά την εκτέλεση των εντολών ανά κατηγορία εντολών
 - Παράδειγμα υπολογισμού CPI:

Κατηγορία εντολών	Αριθμός κύκλων	Ενδεικτική ποσόστωση	Υπολογισμός CPI
Επεξεργασίας δεδομένων DP	4	40%	1.60
Μνήμης LDR	5	16%	0.80
Μνήμης STR	4	8%	0.32
Σύγκρισης CMP	3	6%	0.18
Διακλάδωσης B	3	10%	0.30
Μη εκτελούμενες εντολές	3	20%	0.60
		100%	3.80

Επεξεργαστής πολλών κύκλων: ανάλυση επιδόσεων

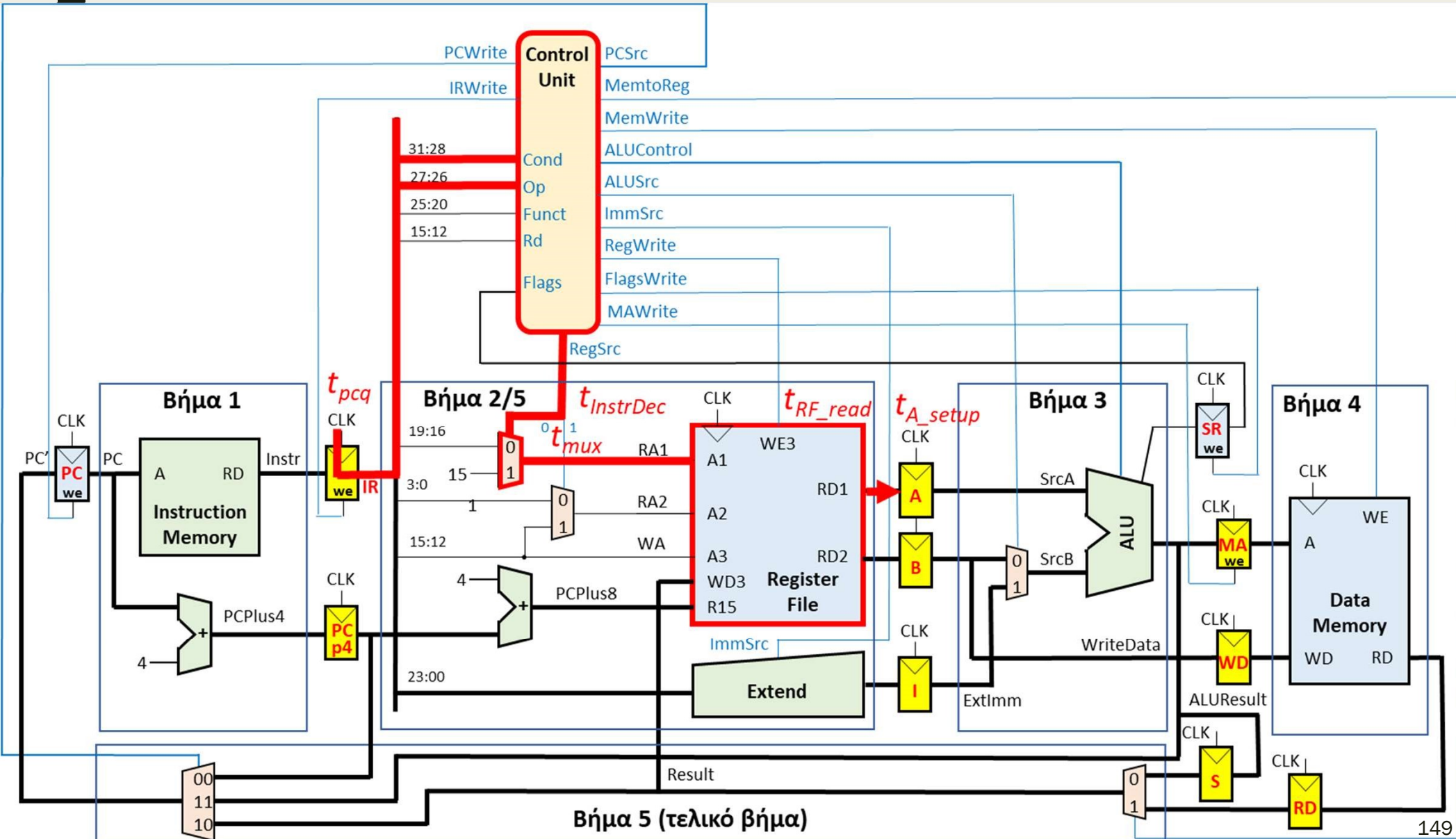
- Η περίοδος του CLK καθορίζεται από την **κρίσιμη διαδρομή (critical path)** που ενεργοποιείται κατά την εκτέλεση του **πιο αργού βήματος της εντολής** μέσω της λογικής της μικροαρχιτεκτονικής πολλών κύκλων
 - Ας υποθέσουμε ότι το πιο αργό βήμα είναι το **Βήμα 2 της αποκωδικοποίησης της εντολής και του διαβάσματος του αρχείου των καταχωρητών**
 - Εξαρτάται από την καθυστέρηση διάδοσης στη μονάδα ALU
- Λειτουργίες που εκτελούνται στο **Βήμα 2** (κοινές για όλες τις εντολές) με την αντίστοιχη καθυστέρηση διάδοσης που συμβάλλουν στην κρίσιμη διαδρομή:
 - φόρτωση νέας εντολής στον καταχωρητή εντολών IR (t_{pcq})
 - υπολογισμός του $RegSrc_0$ στον αποκωδικοποιητή εντολής InstrDec ($t_{InstrDec}$)
 - επιλογή του πεδίου Rn ($Instr_{19:16}$) ως RA1 μέσω πολυπλέκτη 2 σε 1 (t_{mux})
 - ανάγνωση του αρχείου καταχωρητών στην έξοδο RD1 ως SrcA (t_{RF_read})
 - υποθέτουμε ότι η δημιουργία του SrcB έχει μικρότερη καθυστέρηση διάδοσης
 - σταθεροποίηση SrcA για εγγραφή στον καταχωρητή A (t_{A_setup})

Προσοχή! Η καθυστέρηση διάδοσης στον αποκωδικοποιητή εντολής λαμβάνεται υπόψη μόνο για την εύρεση της πιο κρίσιμης διαδρομής που ενεργοποιείται κατά την εκτέλεση του Βήματος 2 της εντολής, γιατί στον ίδιο κύκλο γίνεται πρώτα η αποκωδικοποίηση της εντολής και μετά το διάβασμα του αρχείου καταχωρητών

Επεξεργαστής πολλών κύκλων: ανάλυση επιδόσεων

- Η περίοδος του CLK (T_c) καθορίζεται από την **κρίσιμη διαδρομή (critical path)** που ενεργοποιείται κατά την εκτέλεση του **Βήματος 2** της εντολής

$$T_c = t_{pcq} + t_{InstrDec} + t_{mux} + t_{RF_read} + t_{A_setup}$$



Επεξεργαστής πολλών κύκλων: ανάλυση επιδόσεων

- Η περίοδος του CLK (T_c) καθορίζεται από την **κρίσιμη διαδρομή (critical path)** που ενεργοποιείται κατά την εκτέλεση του **Βήματος 2** της εντολής
 - $T_c = t_{pcq} + t_{InstrDec} + t_{mux} + t_{RF_read} + t_{A_setup}$
- Παράδειγμα υπολογισμού επιδόσεων επεξεργαστή ενός κύκλου:
 - η περίοδος του ρολογιού (CLK) με ενδεικτικές καθυστερήσεις είναι:
 - $T_c = (40 + 60 + 40 + 250 + 50) \text{ ps} = 440 \text{ ps}$
 - ο χρόνος εκτέλεσης ενός προγράμματος των 100 δις εντολών είναι:
 - $\text{CPU}_{\text{time}} = (\# \text{ instr/program}) \times \text{CPI} \times T_c = 100 \times 10^9 \times 3.8 \times 440 \times 10^{-12} = 167.2 \text{ s}$

Ενδεικτικές τιμές

Στοιχεία	Παράμετρος	Καθυστέρηση (ps)
Καθυστέρηση από το CLK έως την έξοδο Q (καταχωρητής)	T_{pcq}	40
Χρόνος σταθεροποίησης καταχωρητή	T_{setup}	50
Χρόνος σταθεροποίησης αρχείου καταχωρητών	t_{RF_setup}	60
Καθυστέρηση διάδοσης πολυπλέκτη	T_{mux}	40
Καθυστέρηση διάδοσης στη μονάδα ALU για πρόσθεση	t_{ALU}	250
Καθυστέρηση διάδοσης στον αποκωδικοποιητή εντολών	t_{Instr_Dec}	60
Χρόνος ανάγνωσης από τη μνήμη εντολών (ROM)	t_{IM_read}	200
Χρόνος ανάγνωσης από τη μνήμη δεδομένων (Distr.-RAM)	t_{DM_read}	300
Χρόνος ανάγνωσης από το αρχείο καταχωρητών (Distr.-RAM)	t_{RF_read}	250

Επεξεργαστής πολλών κύκλων: ανάλυση επιδόσεων

■ Επεξεργαστής ενός κύκλου:

- Η περίοδος του CLK (T_c) καθορίζεται από την **κρίσιμη διαδρομή (critical path)** που ενεργοποιείται κατά την εκτέλεση της **εντολής LDR**
- $T_c = t_{pcq} + t_{IM_read} + t_{InstrDec} + t_{mux} + t_{RF_read} + t_{ALU} + t_{DM_read} + t_{mux} + t_{RF_setup}$
- η περίοδος του ρολογιού (CLK) με ενδεικτικές καθυστερήσεις είναι:
 - $T_c = (40 + 200 + 60 + 40 + 250 + 250 + 300 + 40 + 60) \text{ ps} = 1240 \text{ ps}$
- ο χρόνος εκτέλεσης ενός προγράμματος των 100 δις εντολών είναι:
 - $\text{CPU}_{\text{time}} = (\# \text{ instr/program}) \times \text{CPI} \times T_c = 100 \times 10^9 \times 1 \times 1240 \times 10^{-12} = 124 \text{ s}$

■ Επεξεργαστής πολλών κύκλων

- Η περίοδος του CLK (T_c) καθορίζεται από την **κρίσιμη διαδρομή (critical path)** που ενεργοποιείται κατά την εκτέλεση του **Βήματος 2** της εντολής
- $T_c = t_{pcq} + t_{InstrDec} + t_{mux} + t_{RF_read} + t_{A_setup}$
- η περίοδος του ρολογιού (CLK) με ενδεικτικές καθυστερήσεις είναι:
 - $T_c = (40 + 60 + 40 + 250 + 50) \text{ ps} = 440 \text{ ps}$
- ο χρόνος εκτέλεσης ενός προγράμματος των 100 δις εντολών είναι:
 - $\text{CPU}_{\text{time}} = (\# \text{ instr/program}) \times \text{CPI} \times T_c = 100 \times 10^9 \times 3.8 \times 440 \times 10^{-12} = 167.2 \text{ s}$

Ο επεξεργαστής πολλών κύκλων είναι τελικά πιο αργός από τον επεξεργαστή ενός κύκλου

Επεξεργαστής πολλών κύκλων: ανάλυση επιδόσεων

- Ο επεξεργαστής πολλών κύκλων είναι τελικά **πιο αργός** από τον επεξεργαστή ενός κύκλου
 - Αν και η πιο αργή εντολή (LDR) χωρίστηκε σε πέντε βήματα, ο χρόνος κύκλου του επεξεργαστή πολλών κύκλων δεν βελτιώθηκε ούτε κατά διάνοια στο πενταπλάσιο (είναι κοντά στο τριπλάσιο)
 - Δεν έχουν όλα τα βήματα ακριβώς το ίδιο μήκος
 - Τα 90 ps της επιβάρυνσης **δημιουργίας ακολουθίας** (sequencing overhead) προκαλούνται **σε κάθε ξεχωριστό βήμα** εκτέλεσης της εντολής, και **όχι μόνο μία φορά** για την εκτέλεση ολόκληρης της εντολής
 - Ο επεξεργαστής ενός κύκλου έχει $CPI = 1$, ενώ ο επεξεργαστής πολλών κύκλων έχει περίπου τετραπλάσιο CPI ($CPI = 3.8$)
- Ένας επεξεργαστής πολλών κύκλων γίνεται **γρήγορος** μόνο όταν υποστηρίζεται η **διοχέτευση (pipeline)**
 - Σε κάθε βήμα εκτελείται διαφορετική εντολή

Επιβάρυνση δημιουργίας ακολουθίας = καθυστέρηση διάδοσης από το CLK έως την έξοδο Q του καταχωρητή + χρόνος σταθεροποίησης καταχωρητή