

ΗΥ 232

Οργάνωση και Σχεδίαση Υπολογιστών

Διάλεξη 10

Διοχέτευση (Pipeline)

Νίκος Μπέλλας

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ

Θέματα Απόδοσης

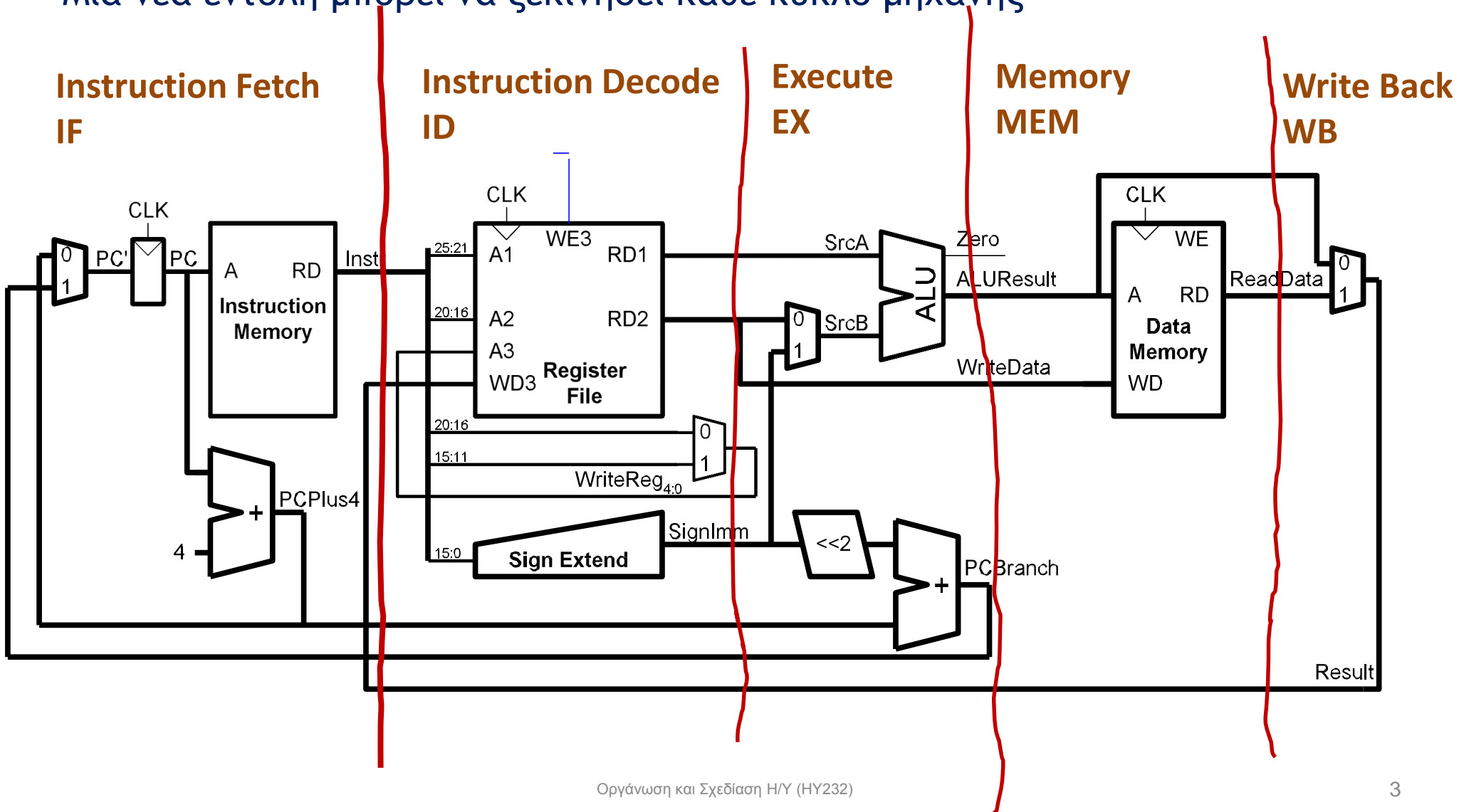
- Αν και απλή, η υλοποίηση ενός κύκλου ρολογιού είναι πολύ αργή
 - Μεγάλη περίοδος ρολογιού

$$T_c = t_{pcq_PC} + t_{mem} + \max(t_{RFread}, t_{sext} + t_{mux}) + t_{ALU} + t_{mem} + t_{mux} + t_{RFsetup}$$

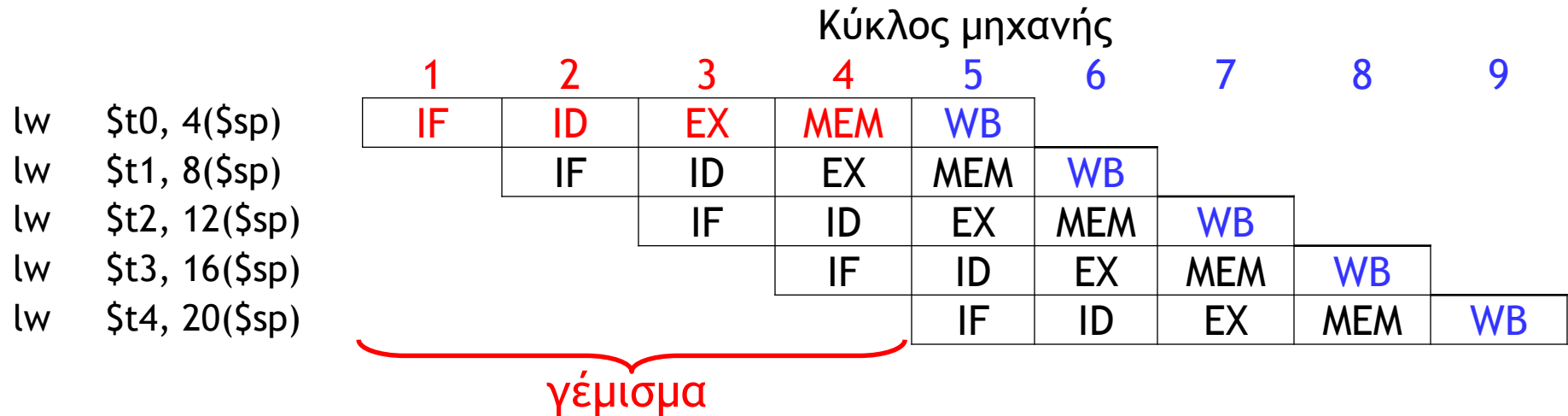
- Ο χρόνος εκτέλεσης είναι ο ίδιος για όλες τις εντολές και ίσος με τον χρόνο εκτέλεσης της πιο αργής εντολής
- Αυτό παραβιάζει μια βασική σχεδιαστική αρχή
 - Βελτιστοποίησε την εκτέλεση της πιο συχνής εντολής
- Η χρήση της διοχέτευσης (pipeline) μειώνει δραστικά την περίοδο ρολογιού

Pipeline 5 κύκλων

Έστω ότι στον επεξεργαστή ενός κύκλου η περίοδος ρολογιού είναι $T_{c1} = 800\text{ps}$
Στον νέο επεξεργαστή με pipeline μπορεί να είναι στην καλύτερη περίπτωση $800/5=160\text{ps}$.
Ας θεωρήσουμε μια πιο ρεαλιστική προσέγγιση $T_{c2} = 200\text{ps}$
Μια νέα εντολή μπορεί να ξεκινήσει κάθε κύκλο μηχανής



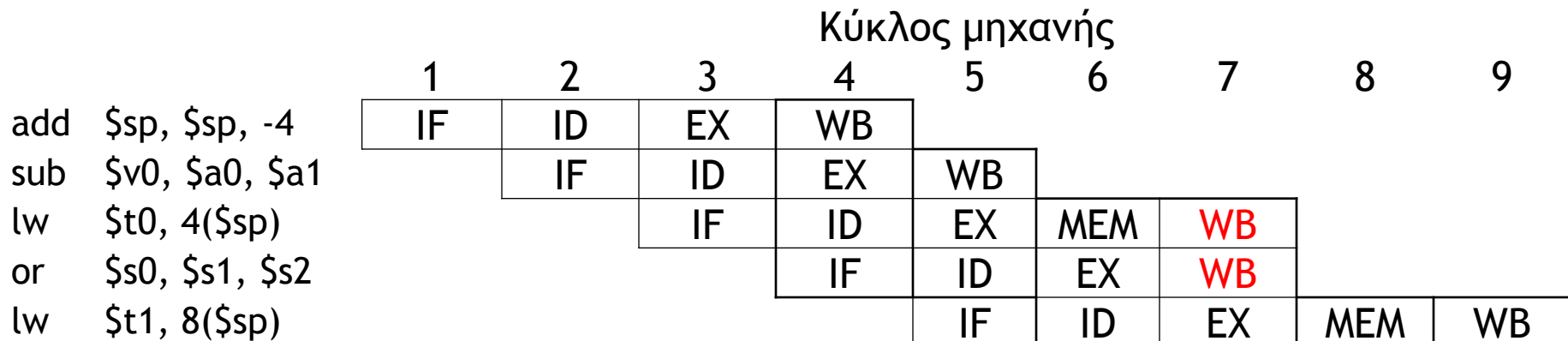
Γιατί Διοχέτευση;



- Χρόνος εκτέλεσης σε ένα ιδανικό pipeline (CPI=1):
 - Χρόνος να γεμίσει το pipeline με εντολές + ένας κύκλος ανά εντολή
 - Για N εντολές χρειαζόμαστε $4+N$ κύκλους μηχανής.
 - Χρόνος pipeline CPU = $(4+N)*200ps = 200.8ns$ για N=1000
- Χρόνος εκτέλεσης σε έναν επεξεργαστή ενός κύκλου
 - Χρόνος single-cycle CPU = $N*800ps = 800ns$
- Επιτάχυνση = $800/200.8 = 4$

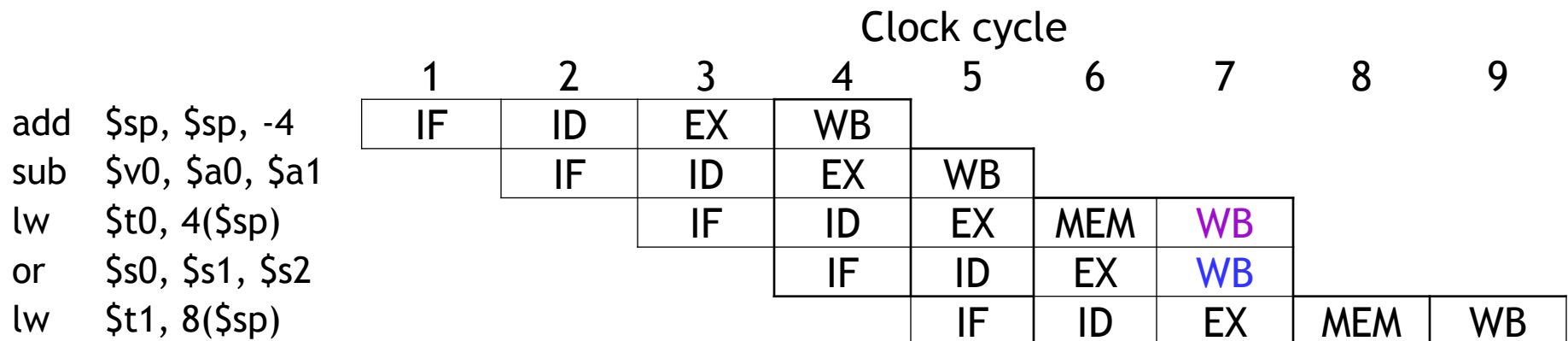
Pipelining για τις υπόλοιπες εντολές

- Η εντολή `lw` χρησιμοποιεί και τα 5 στάδια του pipeline
- Εντολές με R-format δεν χρειάζονται το στάδιο μνήμης MEM, μόνο τα 4 στάδια IF, ID, EX, και WB
- Τι θα συμβεί εάν βάλουμε στο pipeline την `lw` με εντολές R-format;



Δομικοί κίνδυνοι (structural hazards)

- Κάθε δομικό στοιχείο του επεξεργαστή μπορεί να χρησιμοποιηθεί μόνο από μία εντολή κάθε κύκλο μηχανής
- Πρόβλημα:
 - Η εντολή `lw` χρησιμοποιεί το Write Port του Register File στο 5^ο στάδιο της
 - Η εντολή `or` χρησιμοποιεί το Write Port του Register File στο 4^ο στάδιο της



Λύση : χρήση εντολών NOP

- Κανόνες:
 - Όλες οι εντολές παίρνουν 5 κύκλους να εκτελεσθούν
 - Όλες οι εντολές περνάνε από τα ίδια 5 στάδια με την ίδια σειρά.
 - Ακόμη και εάν κάποια στάδια δεν κάνουν καμία επεξεργασία στην εντολή (**NOP**)

R-type

IF	ID	EX	NOP	WB
----	----	----	------------	----

Clock cycle

	1	2	3	4	5	6	7	8	9
add \$sp, \$sp, -4	IF	ID	EX	NOP	WB				
sub \$v0, \$a0, \$a1		IF	ID	EX	NOP	WB			
lw \$t0, 4(\$sp)			IF	ID	EX	MEM	WB		
or \$s0, \$s1, \$s2				IF	ID	EX	NOP	WB	
lw \$t1, 8(\$sp)					IF	ID	EX	MEM	WB

store

IF	ID	EX	MEM	NOP
----	----	----	-----	------------

branch

IF	ID	EX	NOP	NOP
----	----	----	------------	------------

Καταχωρητές Διοχέτευσης

- Για να υλοποιήσουμε ένα σύστημα διοχέτευσης, θα χρησιμοποιήσουμε ενδιάμεσους καταχωρητές στο datapath.
- Όλα τα ενδιάμεσα αποτελέσματα και σήματα ελέγχου πρέπει να σώζονται στο τέλος κάθε κύκλου στον αντίστοιχο καταχωρητή διοχέτευσης.
- Ονόματα καταχωρητών διοχέτευσης.

IF/ID

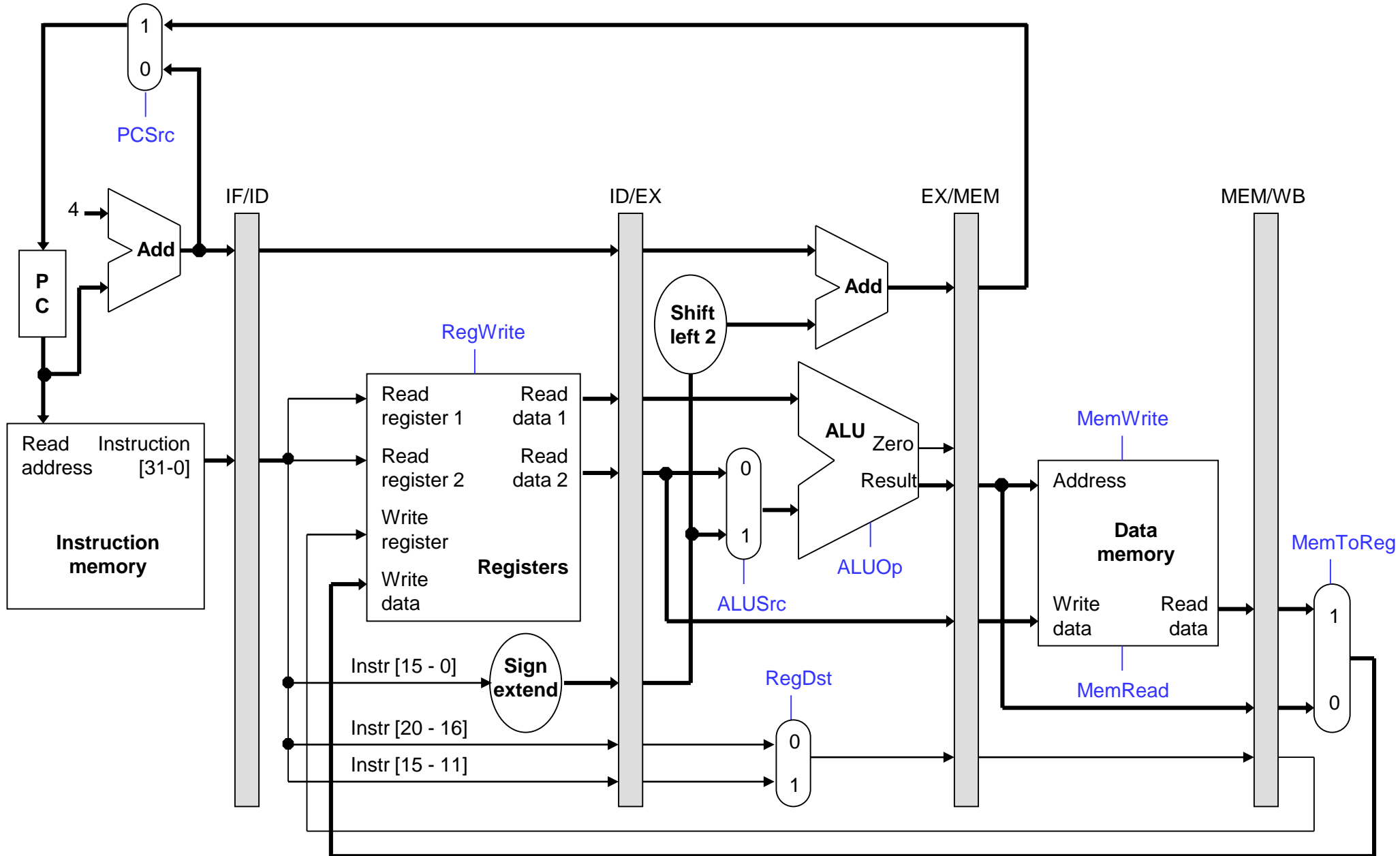
ID/EX

EX/MEM

MEM/WB

- Το τελευταίο στάδιο (Write Back, WB) δεν χρειάζεται καταχωρητές διοχέτευσης γιατί η εντολή τερματίζεται.
- Είναι οι καταχωρητές διοχέτευσης ορατοί στον προγραμματιστή;

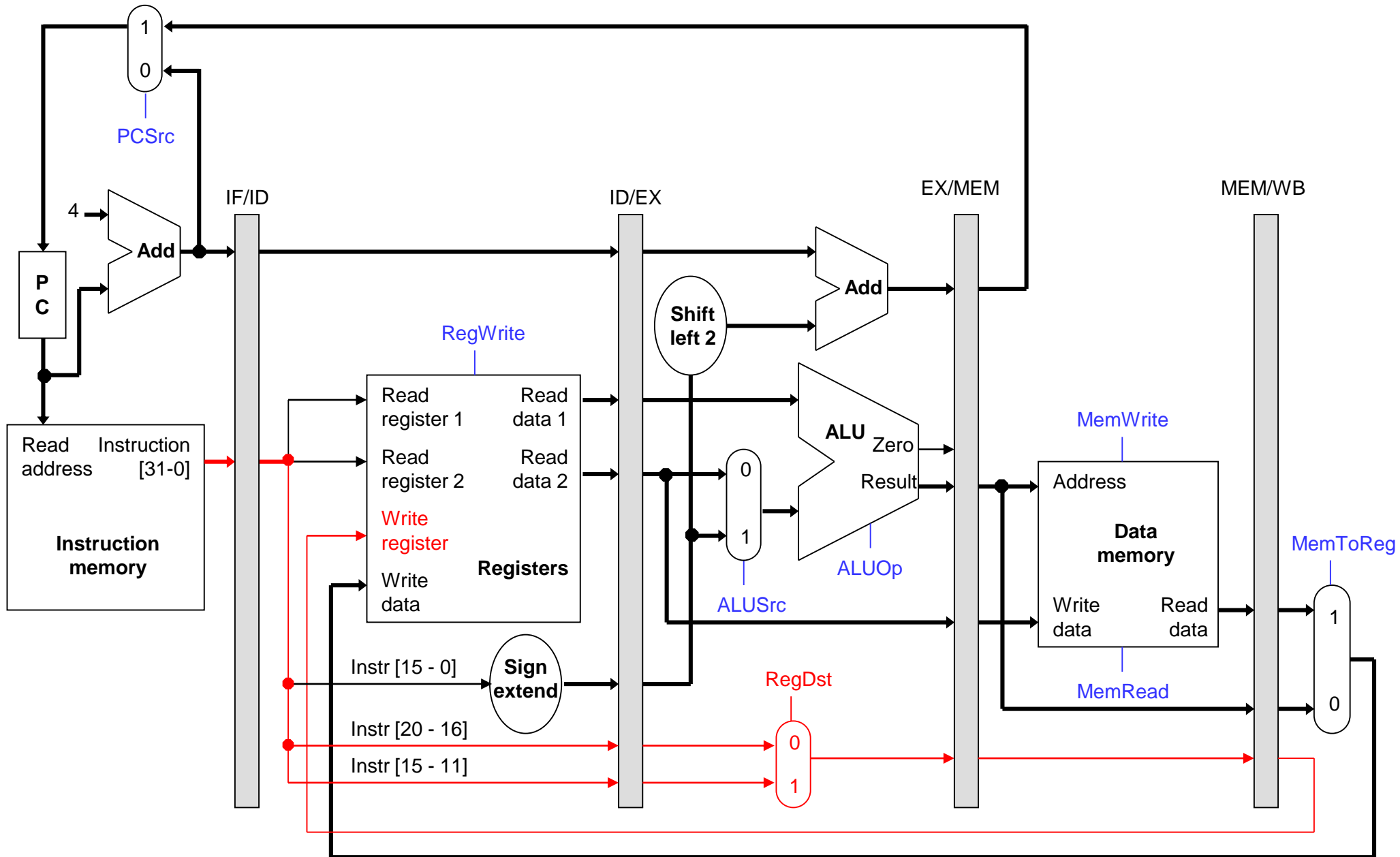
Διοχέτευση Δεδομένων



Πρώθηση δεδομένων

- Οι τιμές των δεδομένων που χρειάζονται σε επόμενα στάδια της διοχέτευσης πρέπει να μεταφερθούν μέσω των καταχωρητών διοχέτευσης
- Ο καταχωρητής προορισμού `rd` (πχ bits [15..11] στο R-format) είναι ένα χαρακτηριστικό παράδειγμα.
 - Το πεδίο `rd` της εντολής ανακτάται στο πρώτο στάδιο (Instruction Fetch, IF). Αλλά χρησιμοποιείται μόνο μετά από 5 κύκλους στο στάδιο Write Back, WB.
 - Συνεπώς, το πεδίο `rd` πρέπει να μεταφερθεί από όλους τους καταχωρητές διοχέτευσης μέχρι το 5^ο στάδιο.
- Δεν μπορούμε να το κρατήσουμε σε ένα απλό καταχωρητή εντολής (instruction register) γιατί σε κάθε κύκλο μηχανής μια νέα εντολή εισέρχεται στον επεξεργαστή από την μνήμη εντολών.

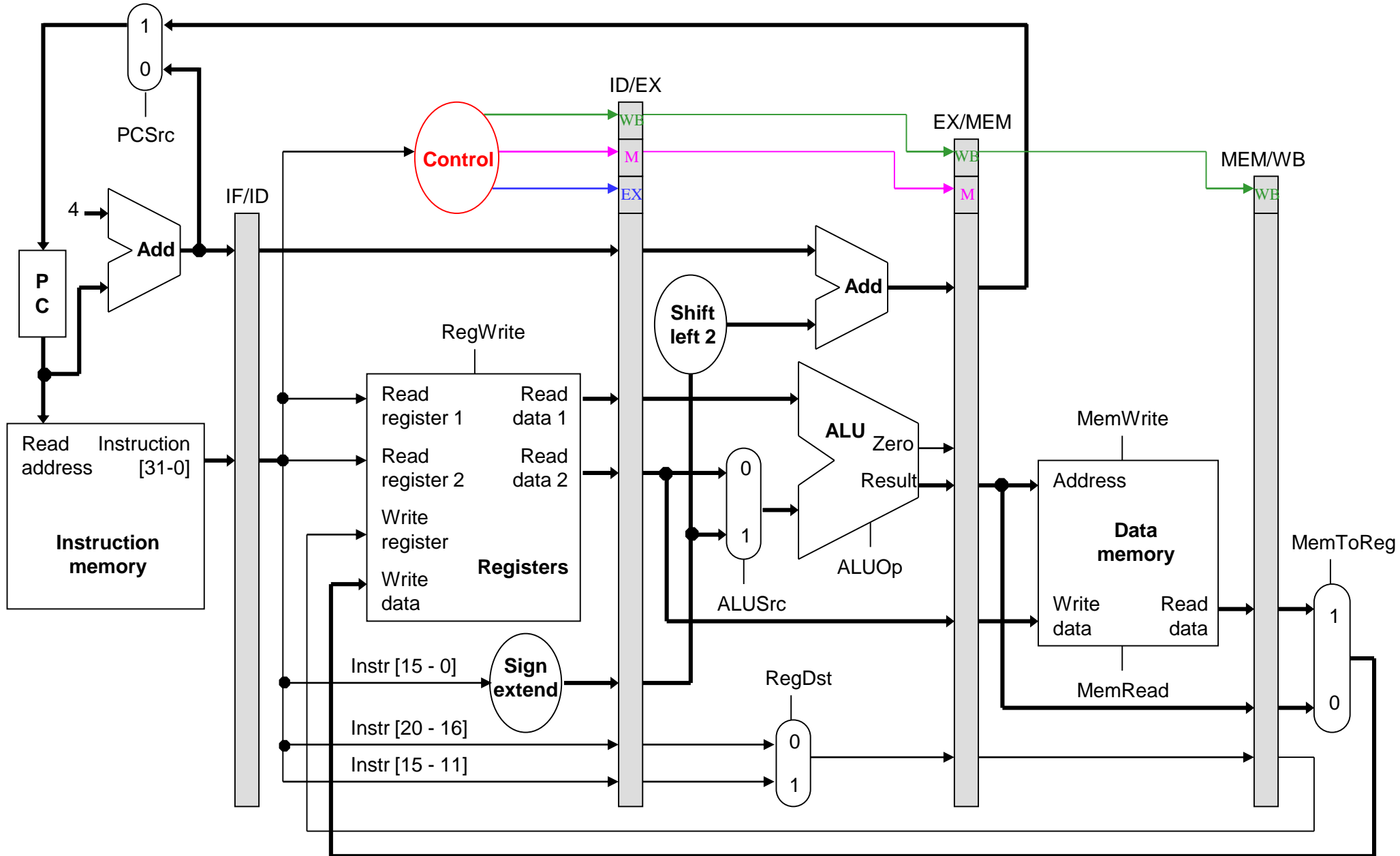
Καταχωρητής Προορισμού



Σήματα Ελέγχου

- Η δημιουργία των σημάτων ελέγχου γίνεται ακριβώς όπως στην περίπτωση του επεξεργαστή ενός κύκλου.
 - Μετά το Instruction Fetch η εντολή αποκωδικοποιείται από την μονάδα ελέγχου και δημιουργούνται τα σήματα ελέγχου.
- Όπως και πριν, κάποια σήματα ελέγχου θα χρειαστούν στα τελευταία στάδια του pipeline (πχ 4^ο ή 5^ο)
- Όπως ακριβώς και τα δεδομένα, έτσι και αυτά τα σήματα ελέγχου θα πρέπει να προωθηθούν μέσω των καταχωρητών διοχέτευσης στα επόμενα στάδια μέχρι να χρησιμοποιηθούν.

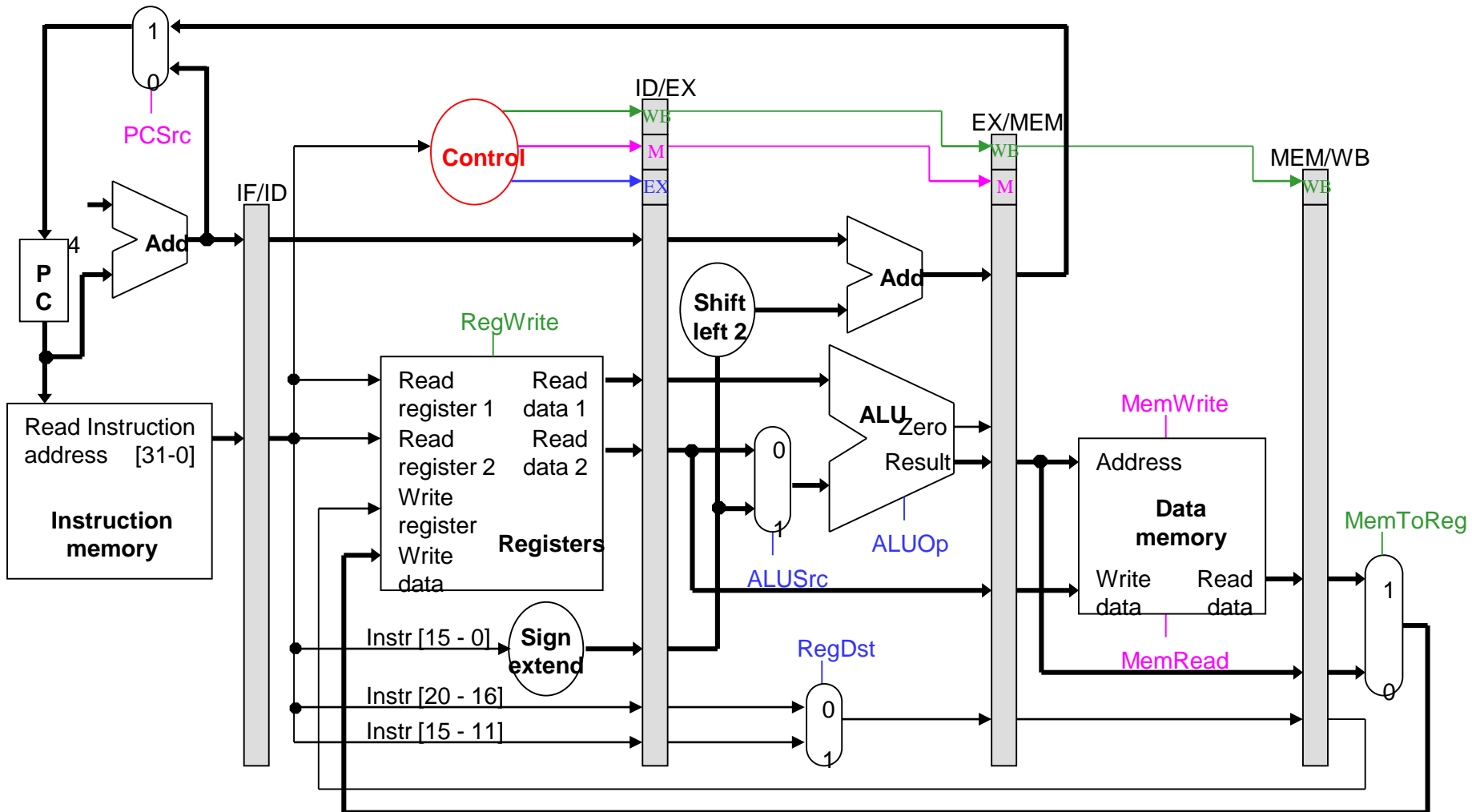
Διοχέτευση Σημάτων Ελέγχου και Δεδομένων



Διοχέτευση Σημάτων Ελέγχου και Δεδομένων

Stage	Control signals needed		
EX	ALUSrc	ALUOp	RegDst
MEM	MemRead	MemWrite	PCSrc
WB	RegWrite	MemToReg	

Σημ: Δεν δείχνουμε όλα τα σήματα



Οργάνωση και Σχεδίαση Η/Υ
(HY232)

Μερικές Παρατηρήσεις

- Συνήθως, όταν θέλουμε να γράψουμε σε έναν καταχωρητή (και μόνον τότε) θέτουμε το αντίστοιχο WriteEnable=1.
 - Ο PC γράφεται σε κάθε κύκλο μηχανής επειδή πάντα πρέπει να δείχνουμε στην επόμενη εντολή και συνεπώς πάντα WriteEnable=1.
 - Το ίδιο και για όλους τους καταχωρητές διοχέτευσης: WriteEnable=1

Παράδειγμα εκτέλεσης εντολών

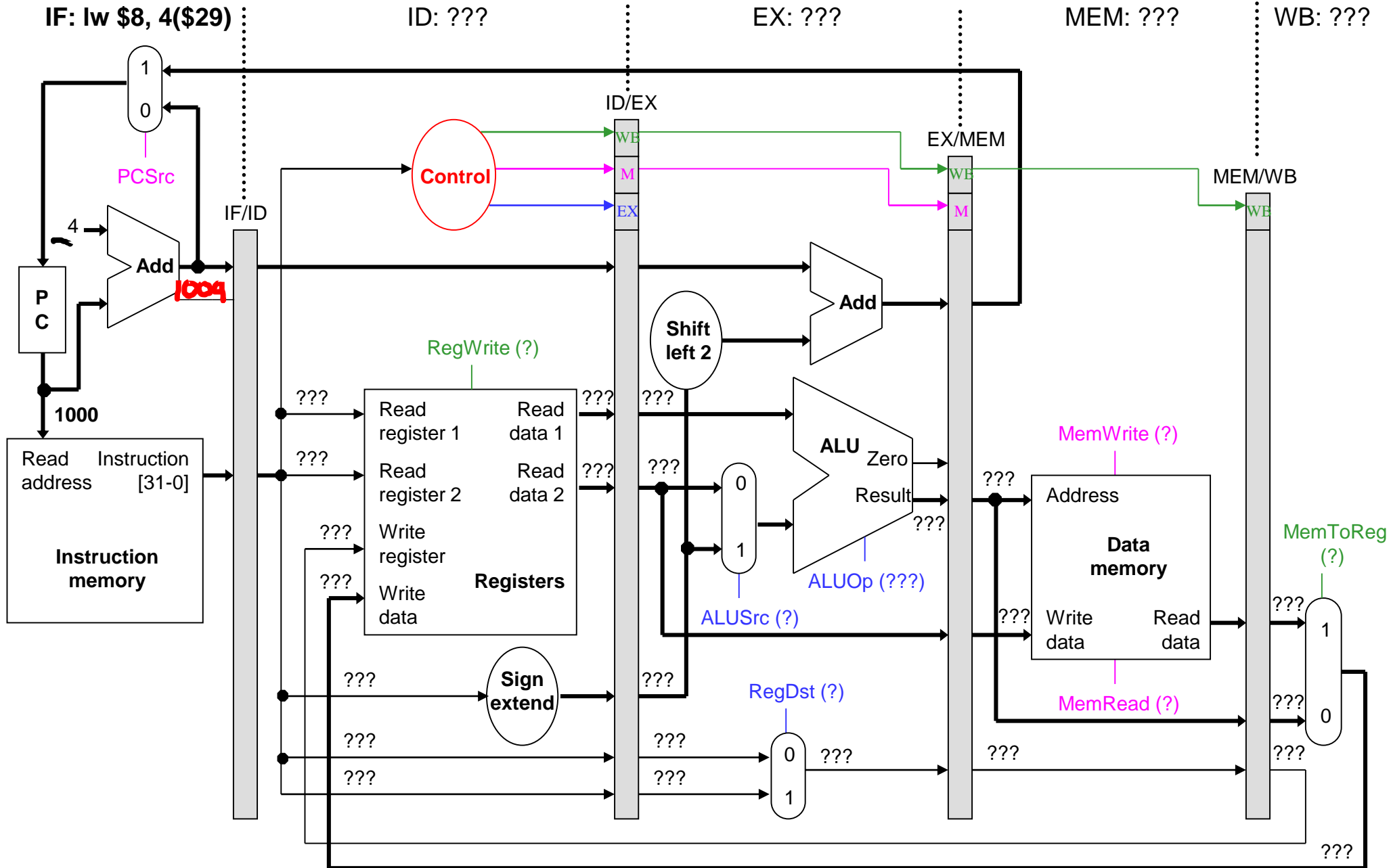
Διευθύνσεις
στο δεκαδικό
σύστημα

1000:	lw	\$8,	4 (\$29)
1004:	sub	\$2,	\$4, \$5
1008:	and	\$9,	\$10, \$11
1012:	or	\$16,	\$17, \$18
1016:	add	\$13,	\$14, \$0

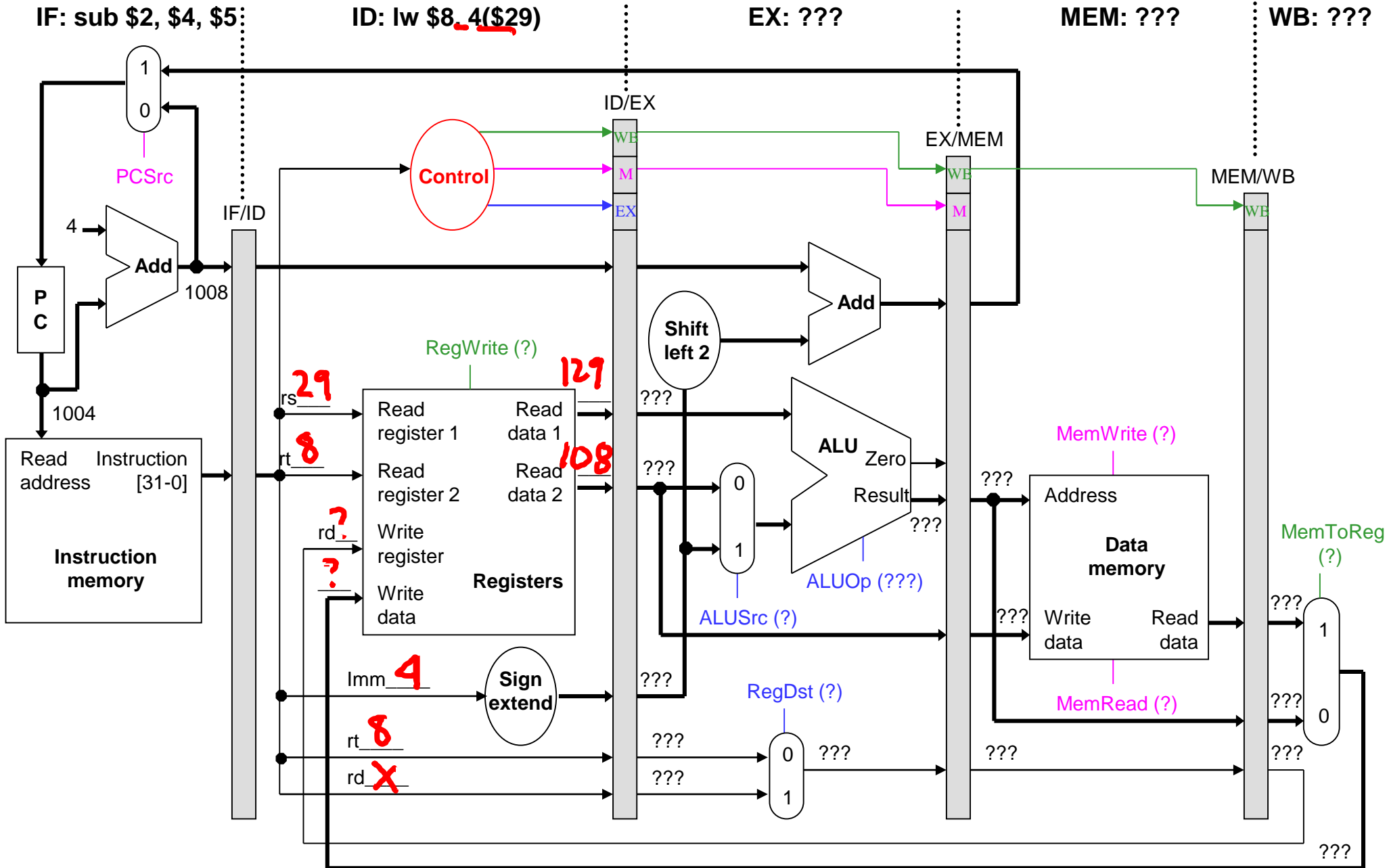
- Υποθέτουμε ότι:

- Η αρχική τιμή κάθε καταχωρητή $\$N$ είναι $N+100$. Για παράδειγμα, ο $\$8$ ($\$t0$) περιέχει την τιμή $d'108$.
- Αρχικά, όλες οι θέσεις μνήμης περιέχουν την τιμή 99.
- Το X στο διάγραμμα είναι Don't Care.
- Τα ερωτηματικά ??? σημαίνει ότι η τιμή του σήματος είναι άγνωστη
- Παρατηρείστε ότι οι εντολές είναι ανεξάρτητες; δεν υπάρχουν εξαρτήσεις μεταξύ καταχωρητών εξόδου μιας εντολής και εισόδου των επόμενων.
 - Περισσότερα για αυτό στα επόμενα μαθήματα

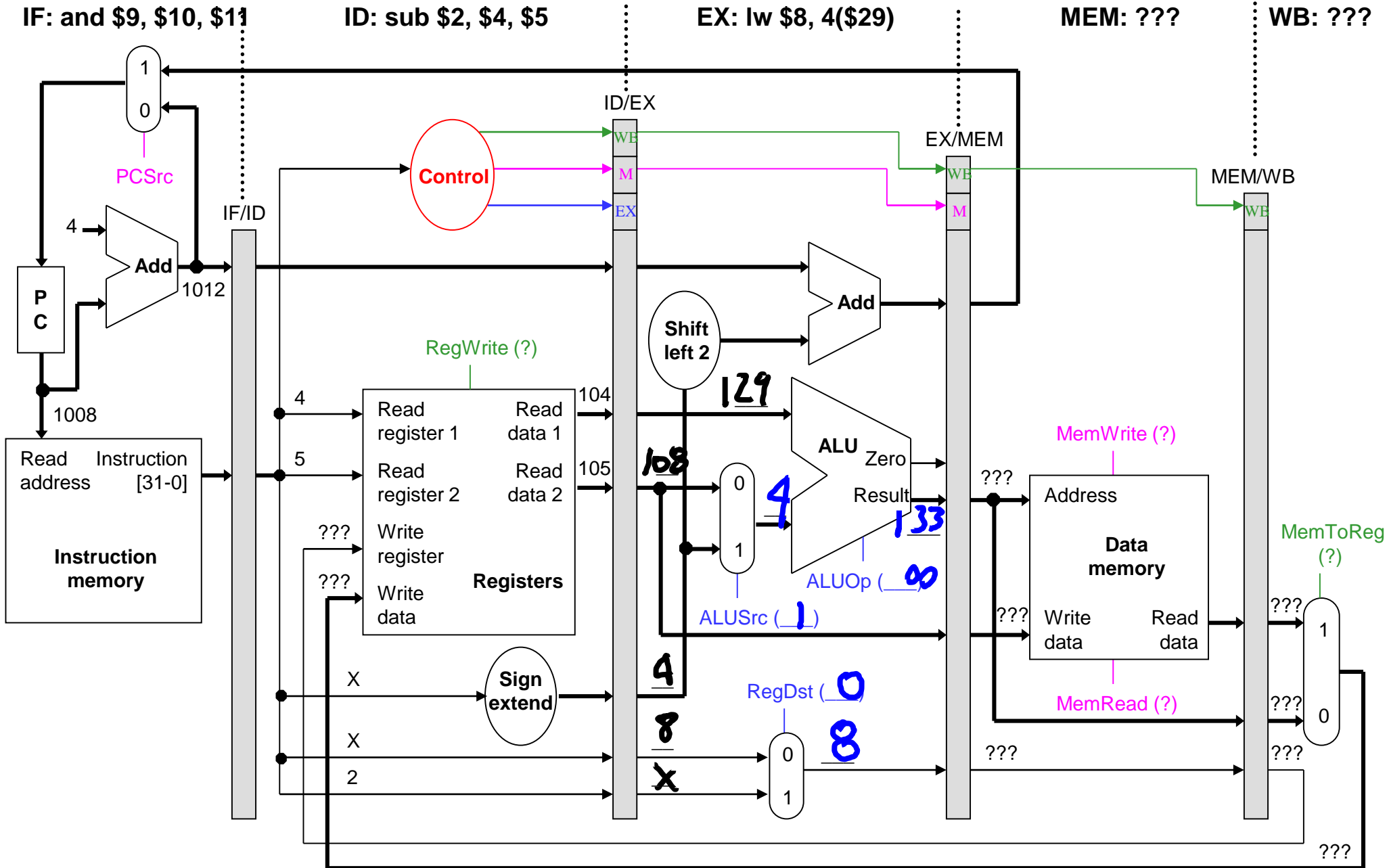
Κύκλος 1 (γέμισμα)



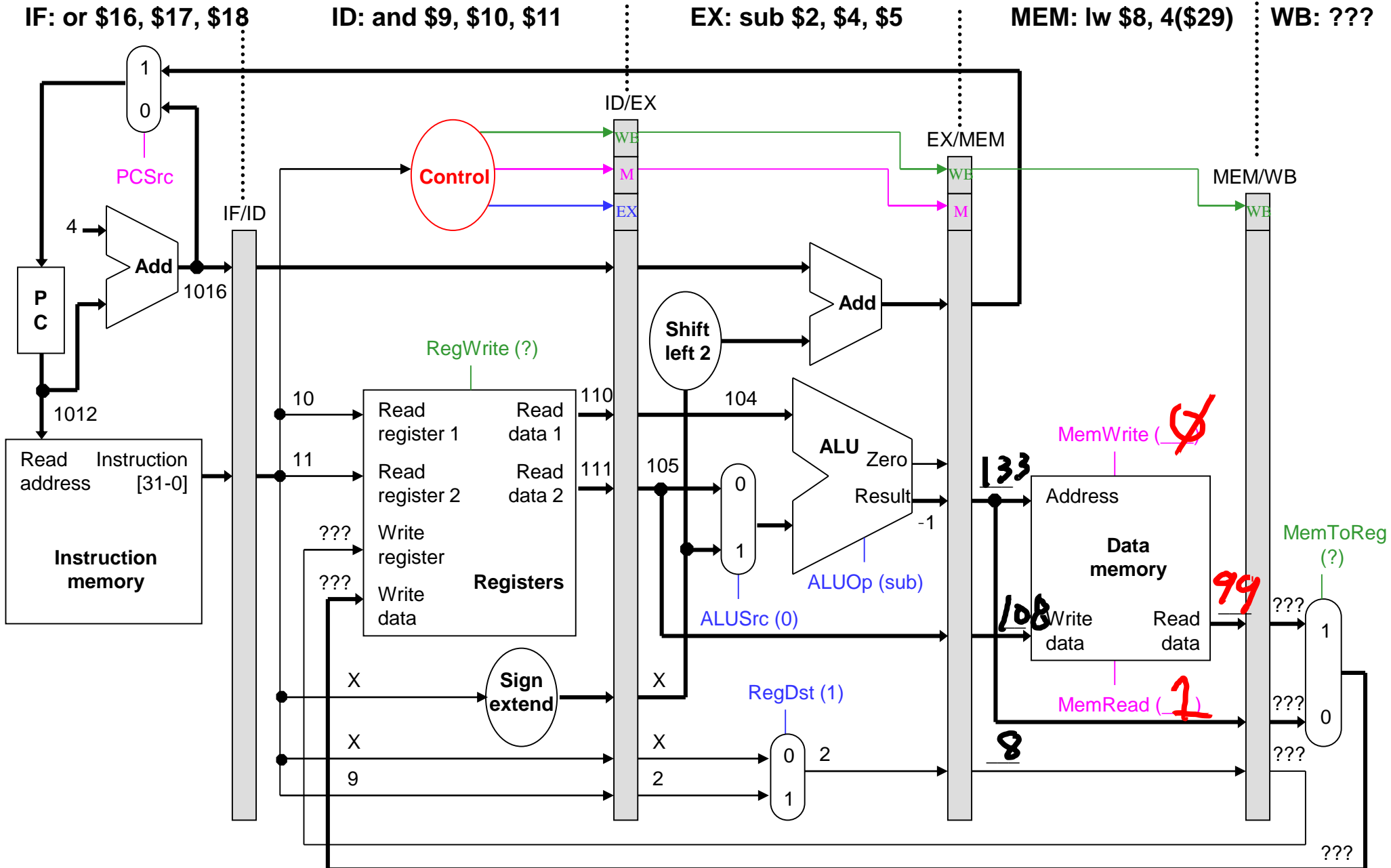
Κύκλος 2 (γέμισμα)



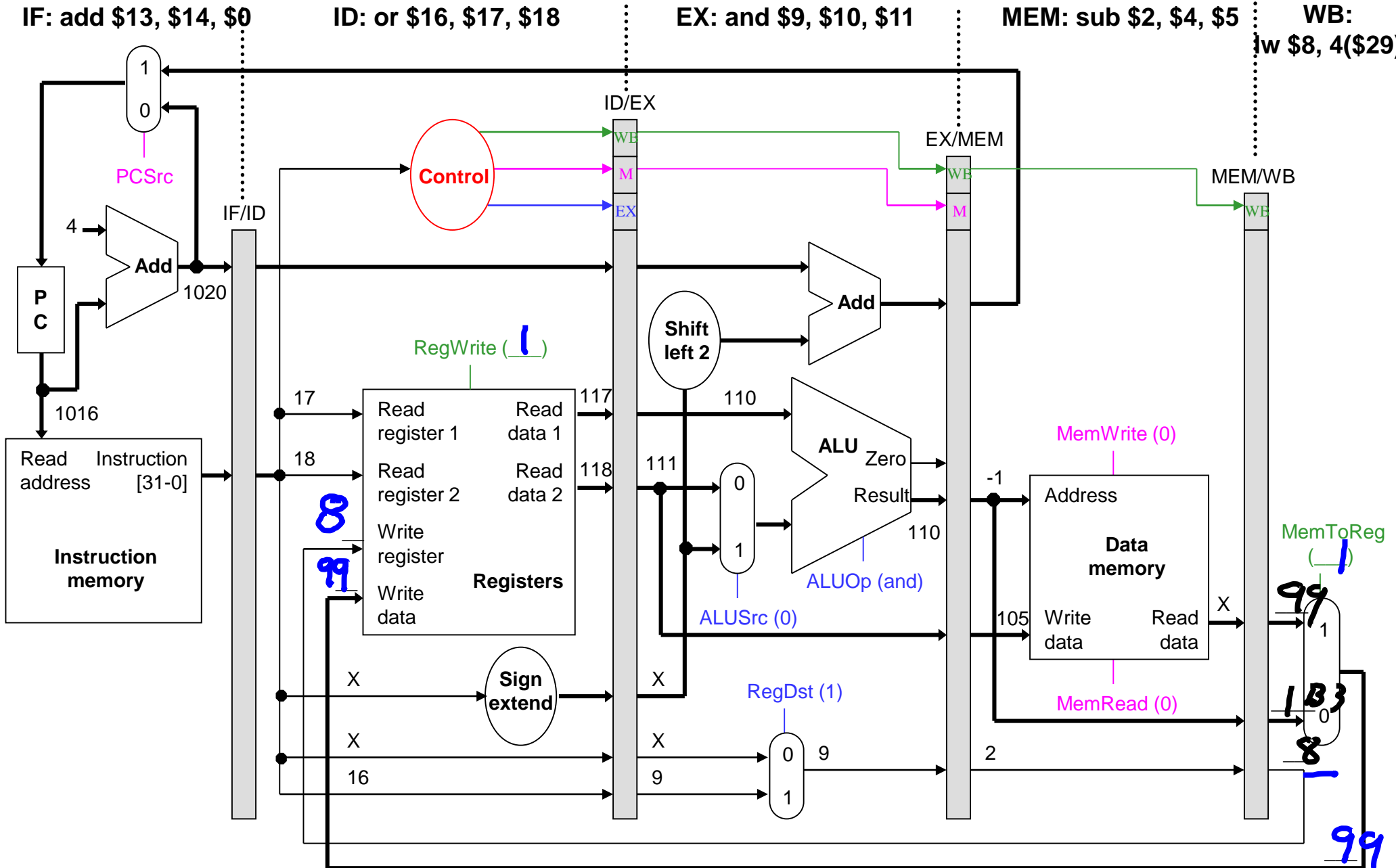
Κύκλος 3 (γέμισμα)



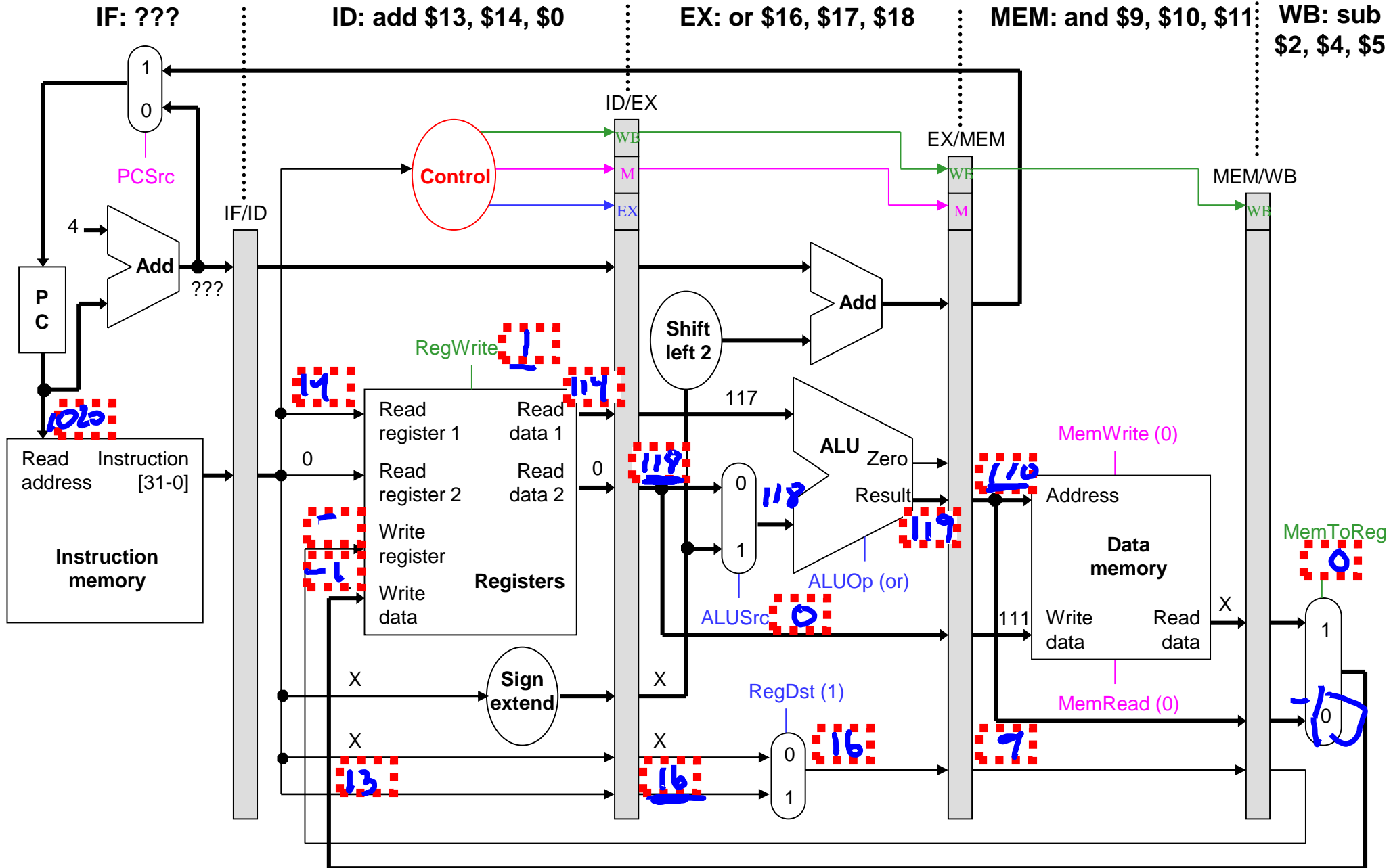
Κύκλος 4 (γέμισμα)



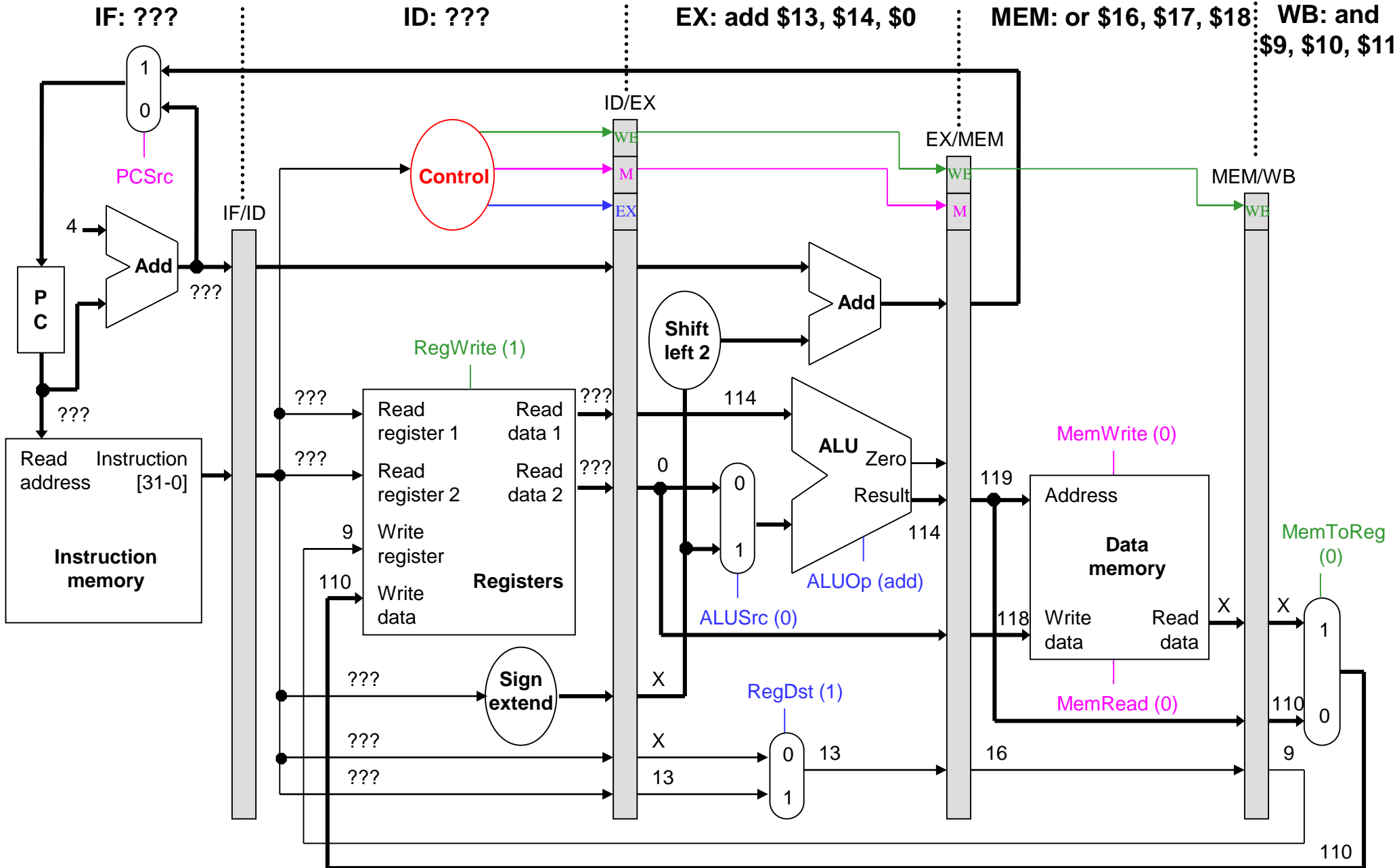
Κύκλος 5



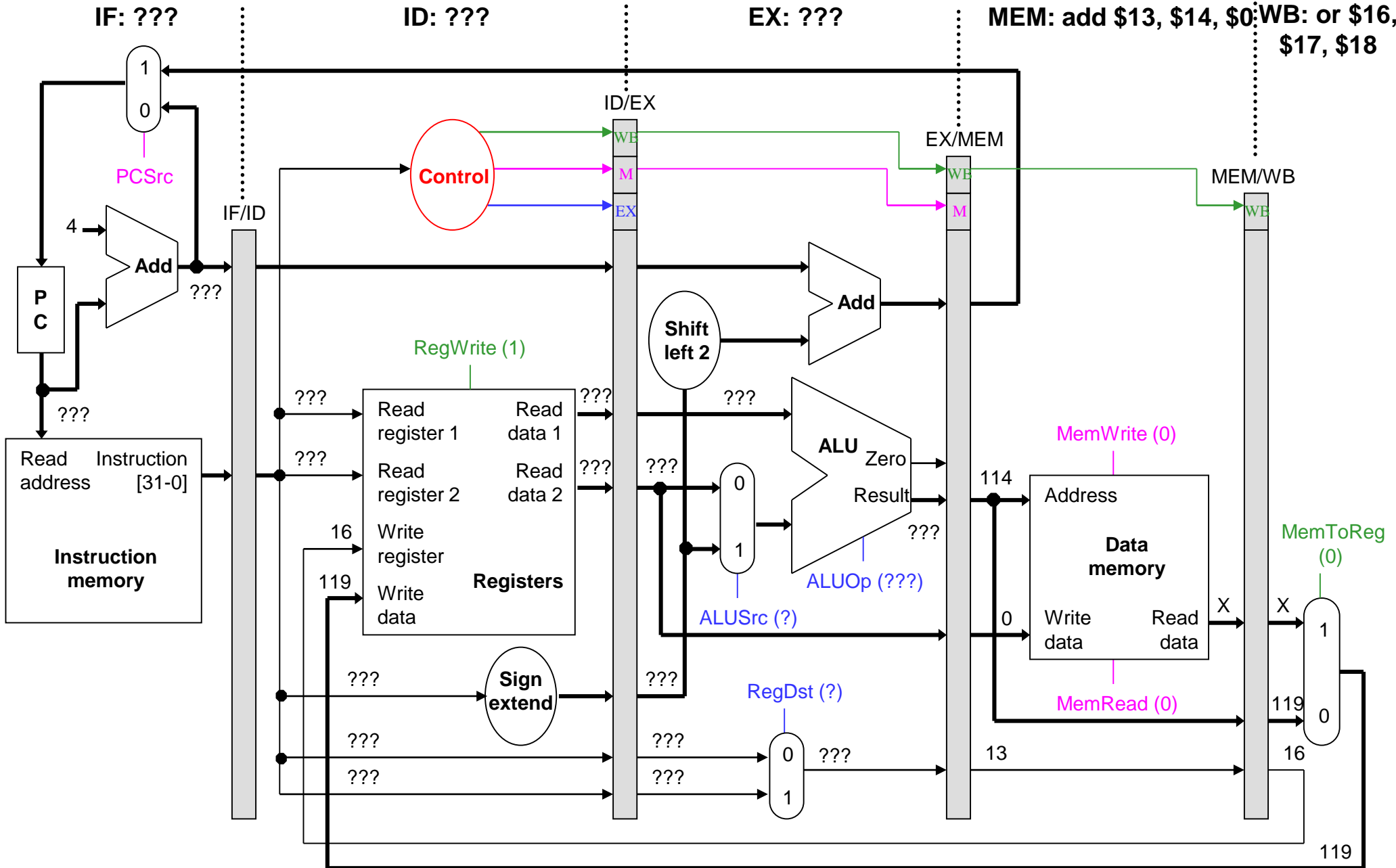
Κύκλος 6 (άδειασμα)



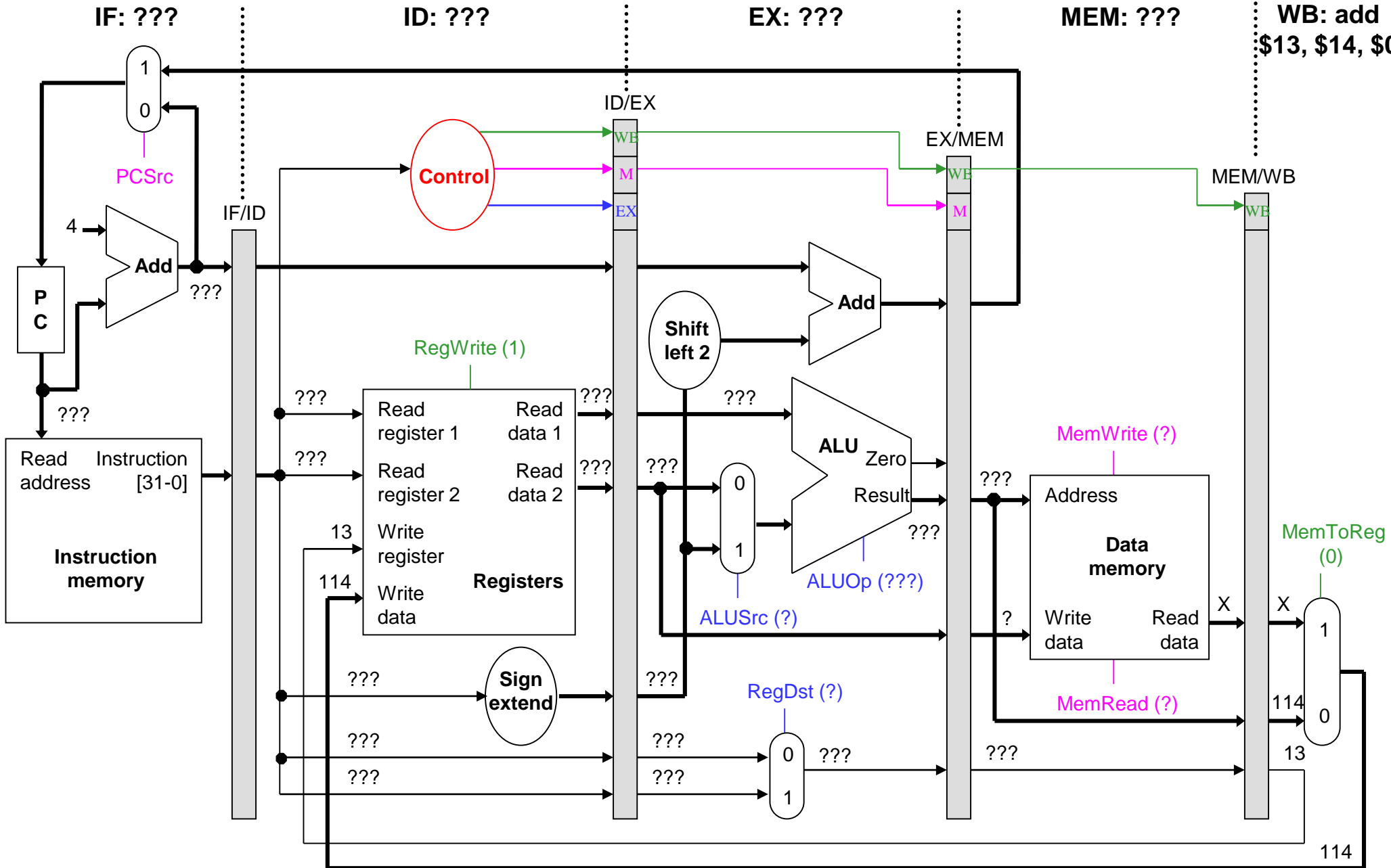
Κύκλος 7 (άδειασμα)



Κύκλος 8 (άδειασμα)



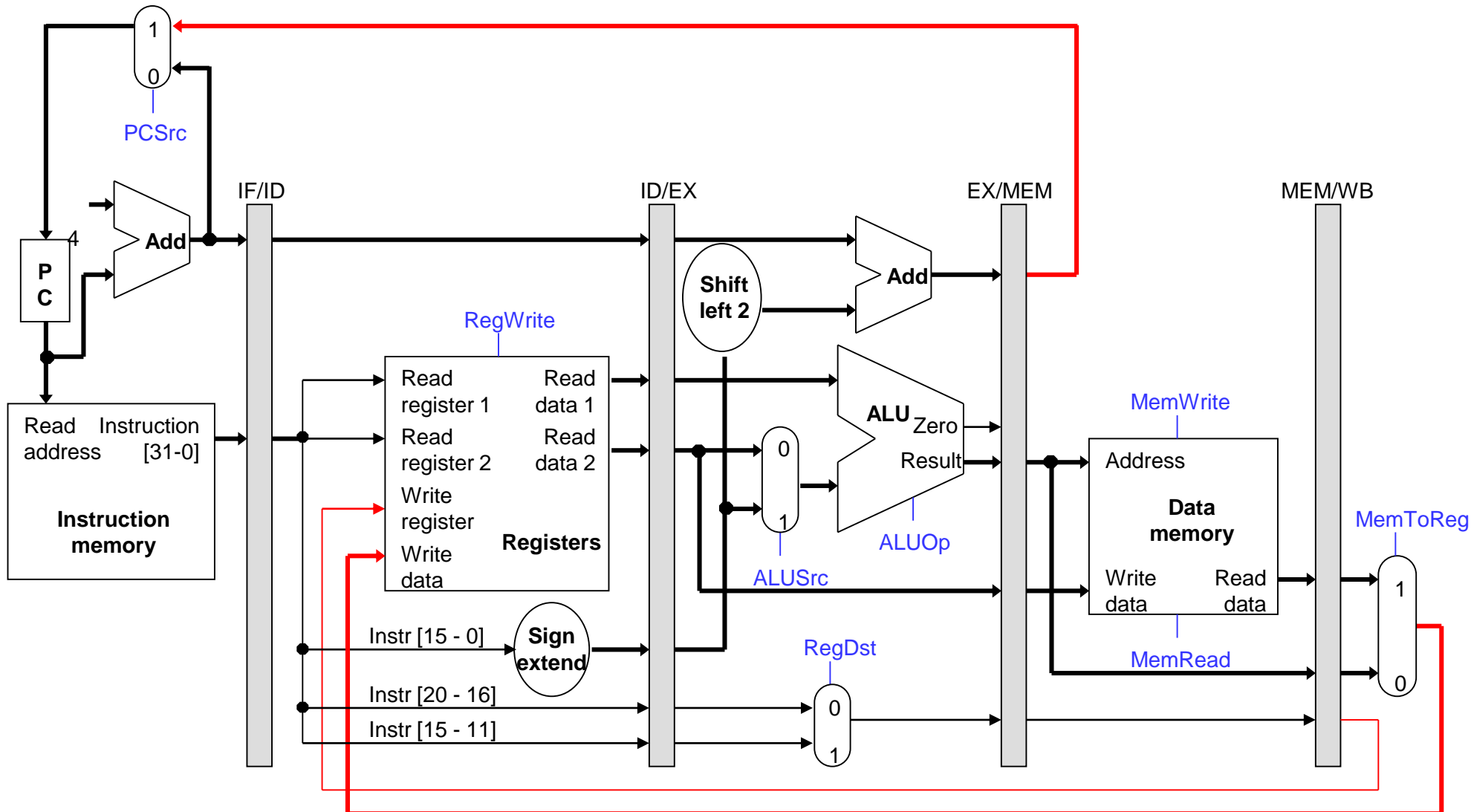
Κύκλος 9 (άδειασμα)



Προσοχή

Τα πάντα κινούνται από αριστερά στα δεξιά εκτός:

1. Της απόφασης διακλάδωσης
2. Του Write Back



Πιο συμβολικά

		Κύκλος ρολογιού								
		1	2	3	4	5	6	7	8	9
lw	\$t0, 4(\$sp)	IF	ID	EX	MEM	WB				
sub	\$v0, \$a0, \$a1		IF	ID	EX	MEM	WB			
and	\$t1, \$t2, \$t3			IF	ID	EX	MEM	WB		
or	\$s0, \$s1, \$s2				IF	ID	EX	MEM	WB	
add	\$t5, \$t6, \$0					IF	ID	EX	MEM	WB

Στον κύκλο 5 ο επεξεργαστής έχει γεμίσει: υπάρχουν τόσες εντολές εν πτήση όσα και τα στάδια της διοχέτευσης

Από εδώ και πέρα θα εξετάσουμε τα προβλήματα που μπορεί να δημιουργήσει η μέθοδος της διοχέτευσης και πως αυτά επιλύονται.