

Κεφάλαιο 7

Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

Sarah L. Harris και David Money Harris



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®


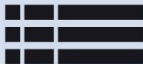
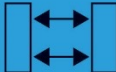
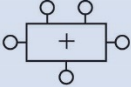
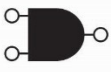
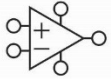

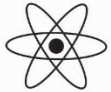
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <1>

Κεφάλαιο 7 :: Θέματα

- Εισαγωγή
- Ανάλυση επιδόσεων
- Επεξεργαστής ενός κύκλου
- Επεξεργαστής πολλών κύκλων
- Επεξεργαστής με υποστήριξη διοχέτευσης
- Προηγμένη μικροαρχιτεκτονική

Application Software	<code>>"hello world!"</code>
Operating Systems	
Architecture	
Micro-architecture	
Logic	
Digital Circuits	
Analog Circuits	
Devices	
Physics	



Εισαγωγή

- **Μικροαρχιτεκτονική:** πώς υλοποιείται μια αρχιτεκτονική σε υλικό
- Επεξεργαστής:
 - **Διαδρομή δεδομένων (datapath):** τμήματα που αφορούν τη λειτουργία
 - **Μονάδα ελέγχου:** σήματα ελέγχου

Λογισμικό εφαρμογών	
Λειτουργικά συστήματα	
Αρχιτεκτονική	
Μικρο-αρχιτεκτονική	
Λογική	
Ψηφιακά κυκλώματα	
Αναλογικά κυκλώματα	
Διατάξεις	
Φυσική	



Μικροαρχιτεκτονική

- Πολλές υλοποιήσεις για μία αρχιτεκτονική:
 - **Ενός κύκλου:** Κάθε εντολή εκτελείται σε έναν κύκλο
 - **Πολλών κύκλων:** Κάθε εντολή χωρίζεται σε μια σειρά από συντομότερα βήματα
 - **Με διοχέτευση:** Κάθε εντολή χωρίζεται σε μια σειρά από βήματα και εκτελούνται ταυτόχρονα περισσότερες από μία εντολές



Επιδόσεις επεξεργαστή

- Χρόνος εκτέλεσης προγραμμάτων

χρόνος εκτέλεσης = (πλήθος εντολών)(κύκλοι/εντολή)(δευτερόλεπτα/κύκλο)

- Ορισμοί:

- CPI: cycles/instruction (κύκλοι/εντολή)
- Περίοδος ρολογιού: δευτερόλεπτα/κύκλο
- IPC: instructions/cycle (εντολές ανά κύκλο)

- Η πρόκληση για τον σχεδιαστή είναι να ικανοποιούνται ταυτόχρονα περιορισμοί που αφορούν

- Το κόστος
- Την κατανάλωση ισχύος
- Τις επιδόσεις



Επεξεργαστής ARM

- Έχουμε μελετήσει ένα **υποσύνολο** των εντολών της αρχιτεκτονικής ARM:
 - Εντολές επεξεργασίας δεδομένων:
 - **ADD, SUB, AND, ORR**
 - Με καταχωρητές και άμεσους τελεστέους ως το πεδίο *Src2*, αλλά **όχι ολισθήσεις**
 - Εντολές μνήμης:
 - **LDR, STR**
 - Με **θετικό άμεσο τελεστέο** για σχετική απόσταση
 - Εντολές διακλάδωσης:
 - **B**



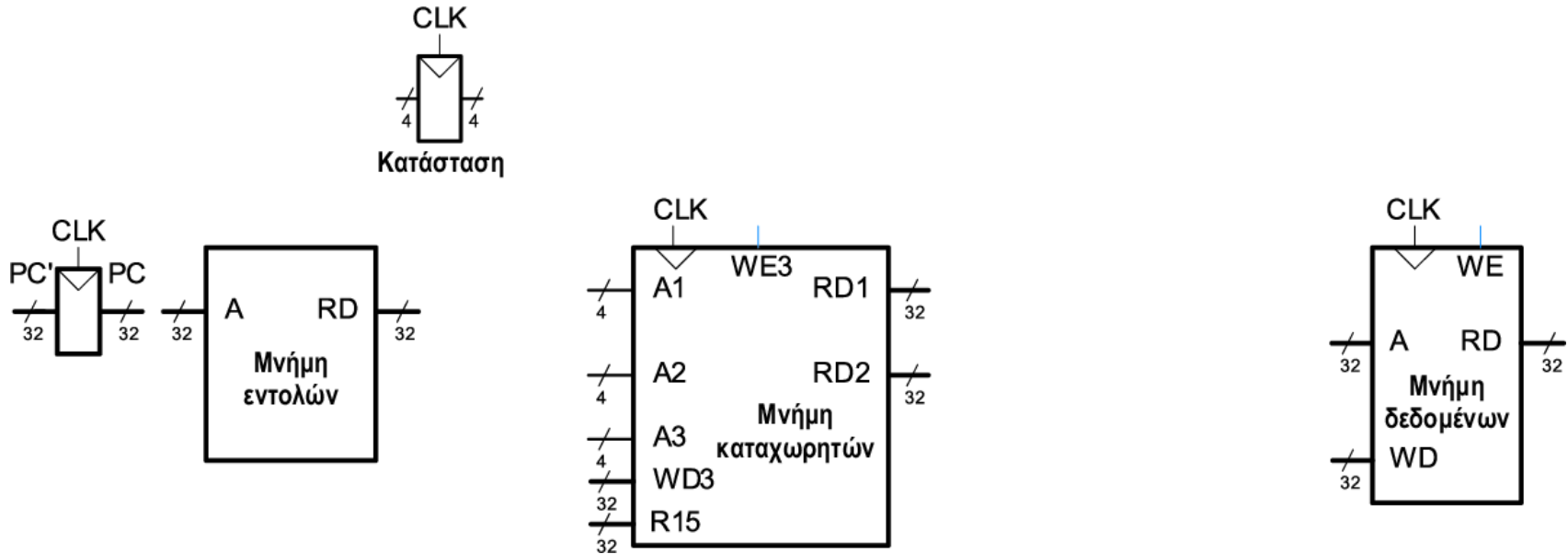
Στοιχεία αρχιτεκτονικής κατάστασης

Τα πάντα για έναν επεξεργαστή καθορίζονται από:

- Την αρχιτεκτονική κατάσταση:
 - 16 καταχωρητές (συμπεριλαμβανομένου του μετρητή PC)
 - Καταχωρητής κατάστασης
- Τη μνήμη

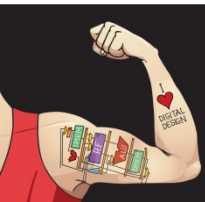


Στοιχεία αρχιτεκτονικής κατάστασης: ARM



Επεξεργαστής ARM ενός κύκλου

- Διαδρομή δεδομένων
- Μονάδα ελέγχου



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <9>

Επεξεργαστής ARM ενός κύκλου

- Διαδρομή δεδομένων
- Μονάδα ελέγχου



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

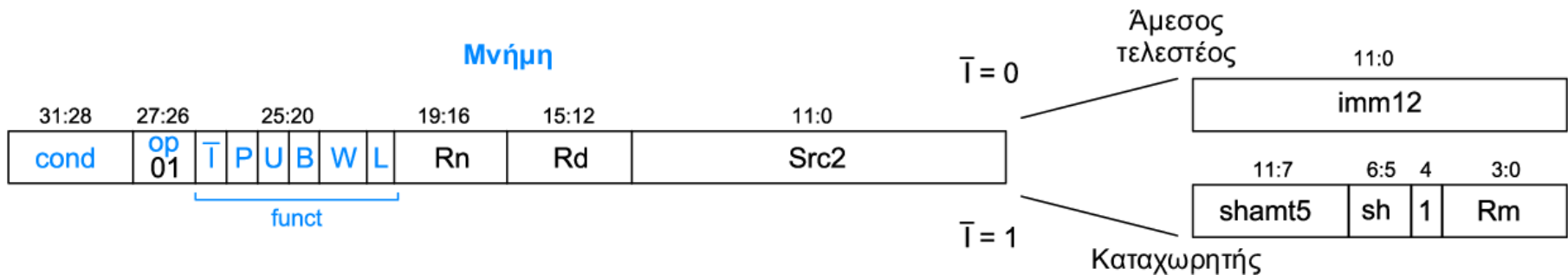
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτότυπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <10>

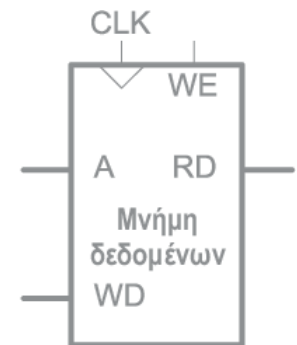
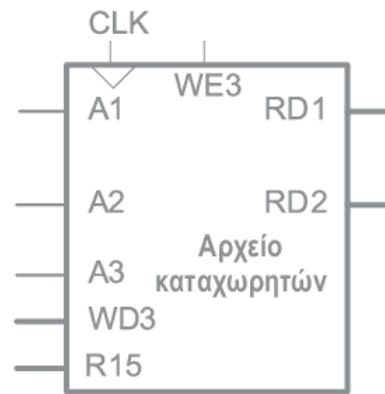
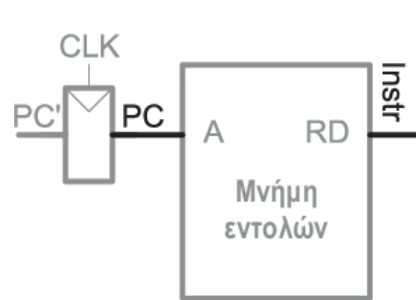
Επεξεργαστής ARM ενός κύκλου

- Διαδρομή δεδομένων: Θα ξεκινήσουμε με την εντολή LDR
- Παράδειγμα: `LDR R1, [R2, #5]`
`LDR Rd, [Rn, imm12]`



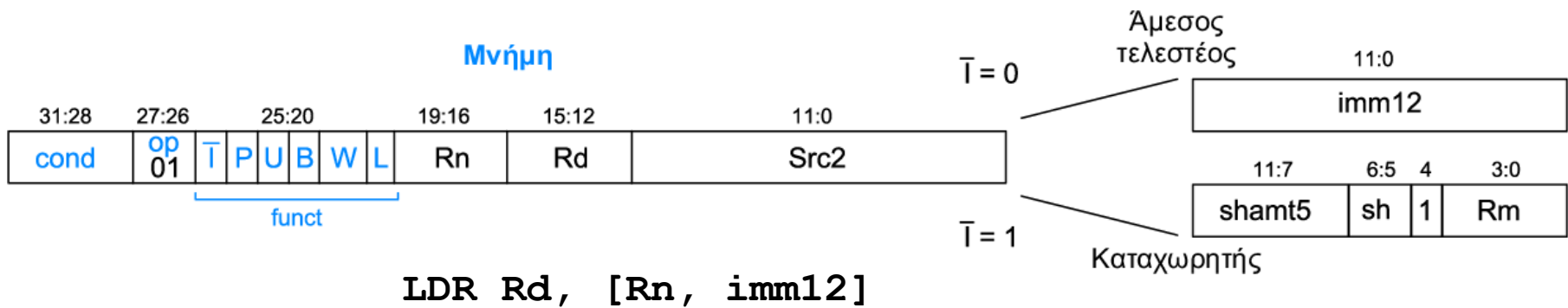
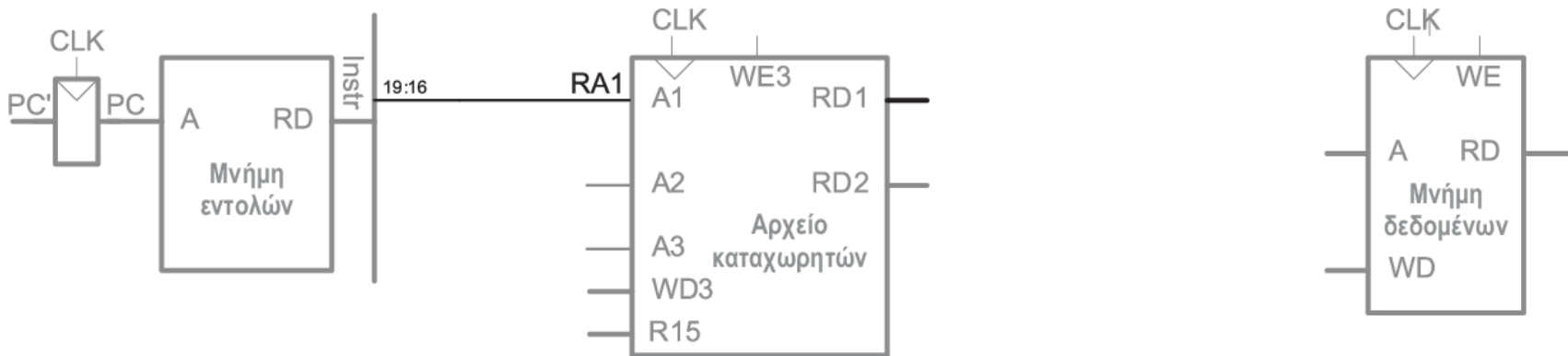
Διαδρομή δεδομένων ενός κύκλου: Εντολή LDR – Προσκόμιση

ΒΗΜΑ 1: Προσκόμιση εντολής



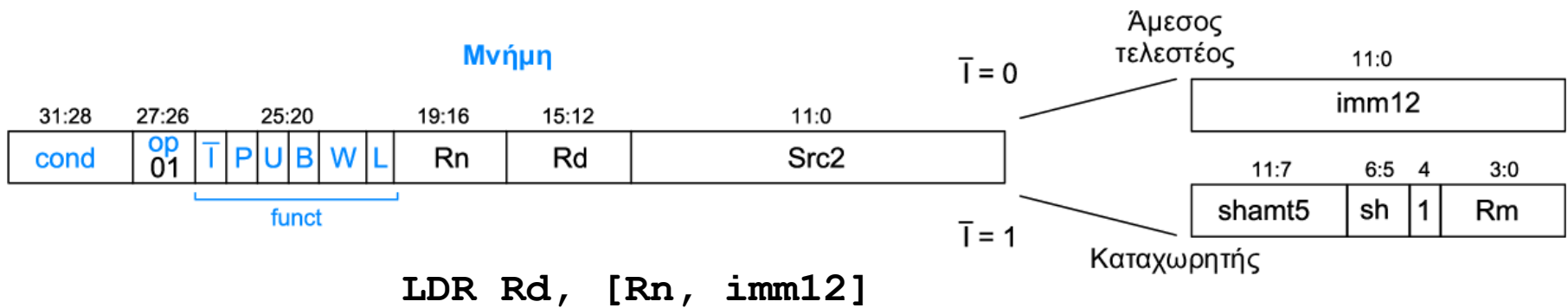
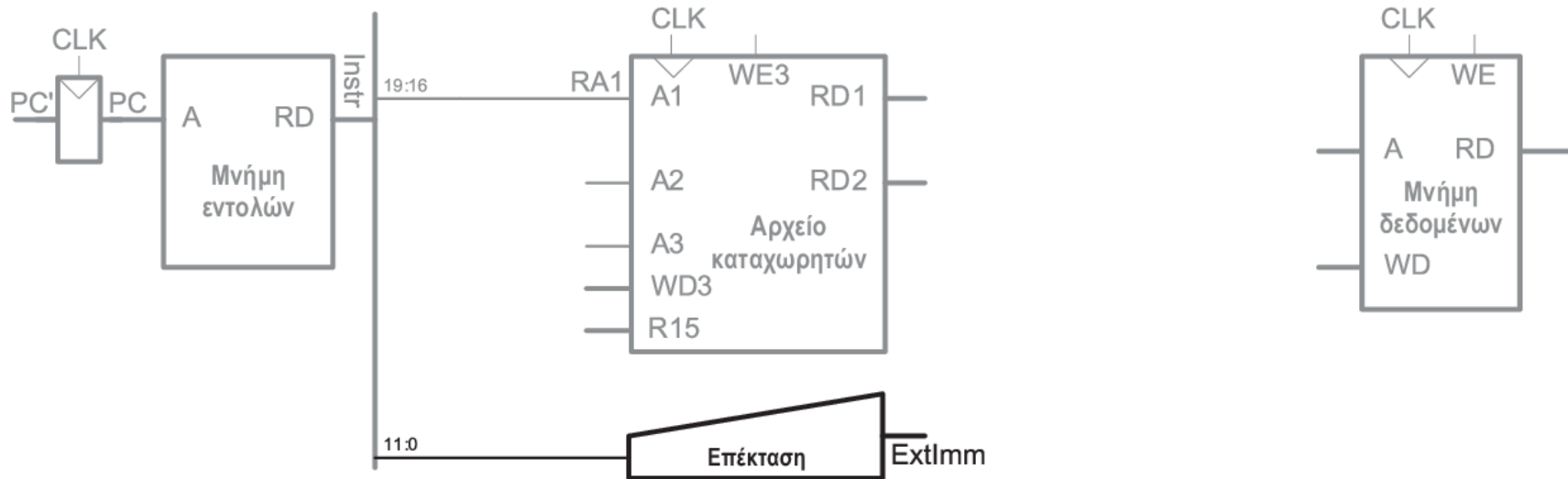
Διαδρομή δεδομένων ενός κύκλου: Εντολή LDR – Ανάγνωση από αρχείο καταχωρητών

ΒΗΜΑ 2: Ανάγνωση τελεστών προέλευσης από το αρχείο καταχωρητών (RF)



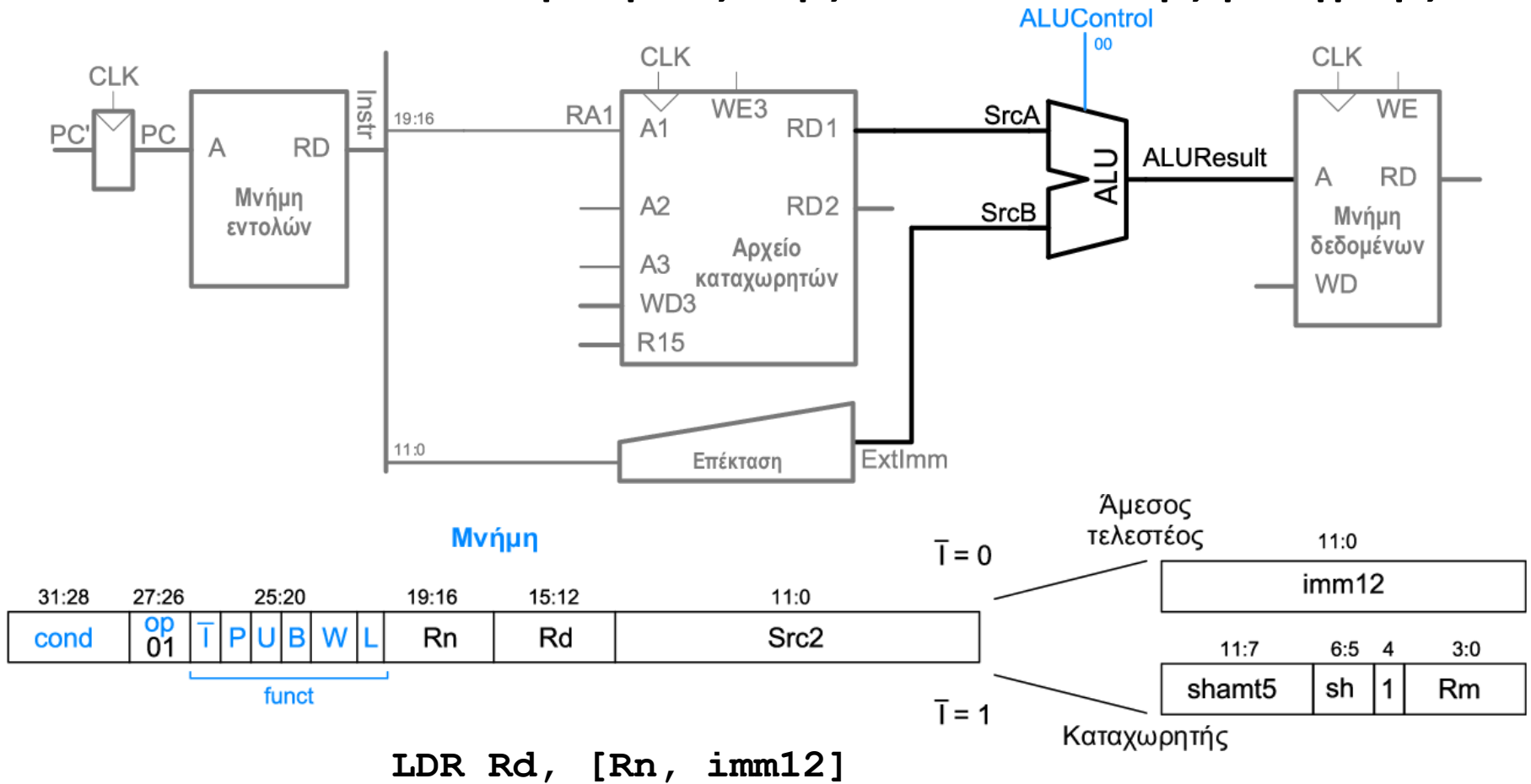
Διαδρομή δεδομένων ενός κύκλου: Εντολή LDR – Άμεσος τελεστής

ΒΗΜΑ 3: Επέκταση του άμεσου τελεστή



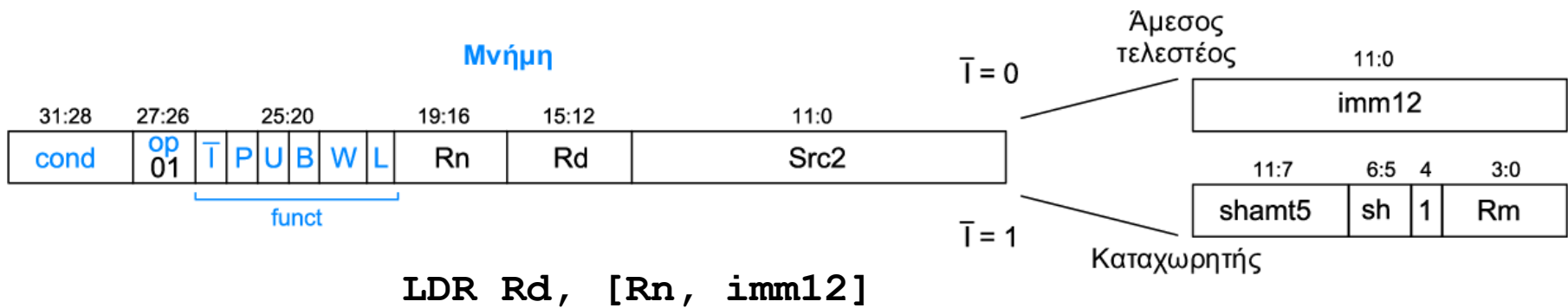
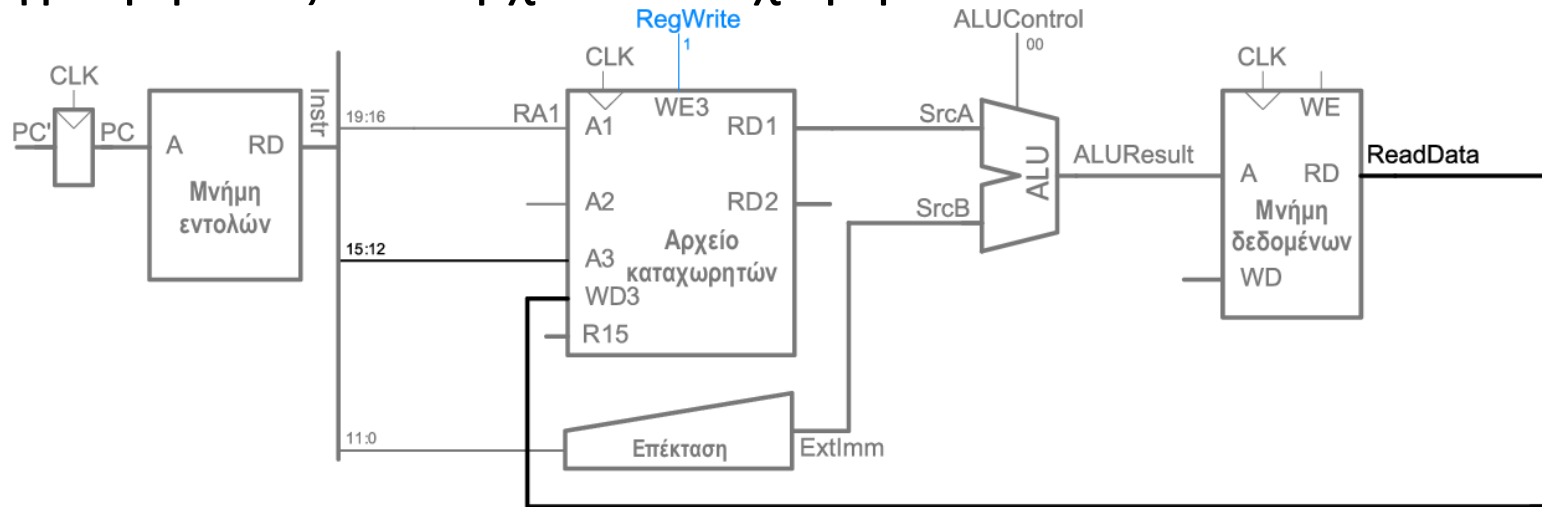
Διαδρομή δεδομένων ενός κύκλου: Εντολή LDR – Διεύθυνση

ΒΗΜΑ 4: Υπολογισμός της διεύθυνσης μνήμης



Διαδρομή δεδομένων ενός κύκλου: Εντολή LDR – Ανάγνωση από τη μνήμη

ΒΗΜΑ 5: Ανάγνωση δεδομένων από τη μνήμη και ετεροχρονισμένη εγγραφή τους στο αρχείο καταχωρητών



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

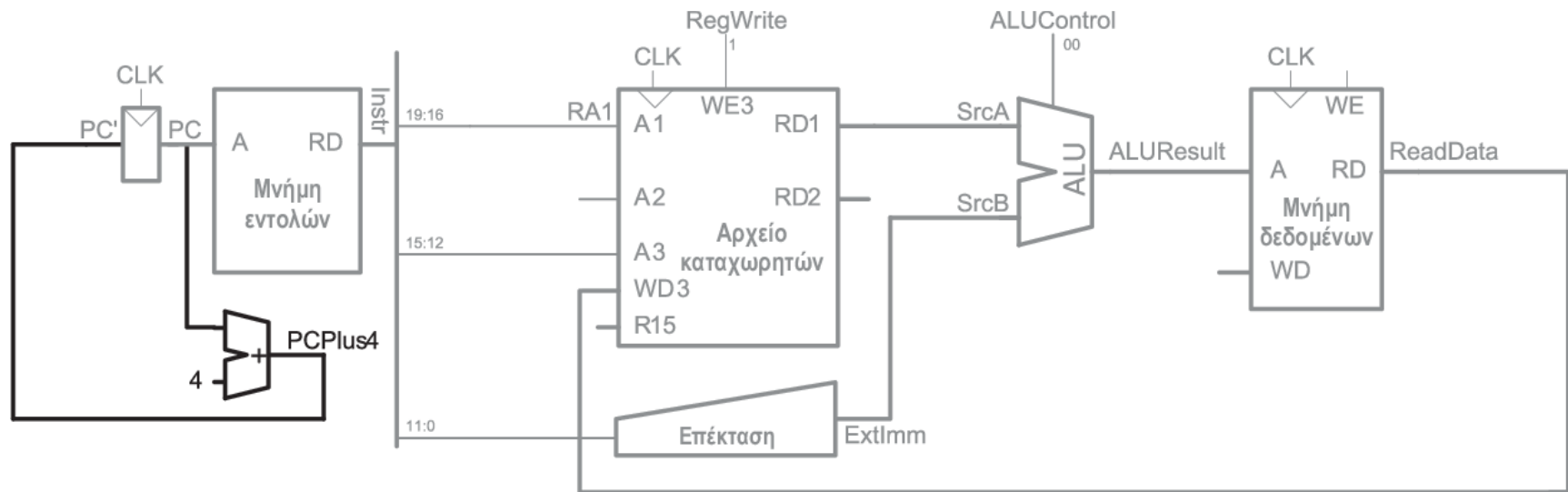
Κεφάλαιο 7 <16>



Διαδρομή δεδομένων ενός κύκλου:

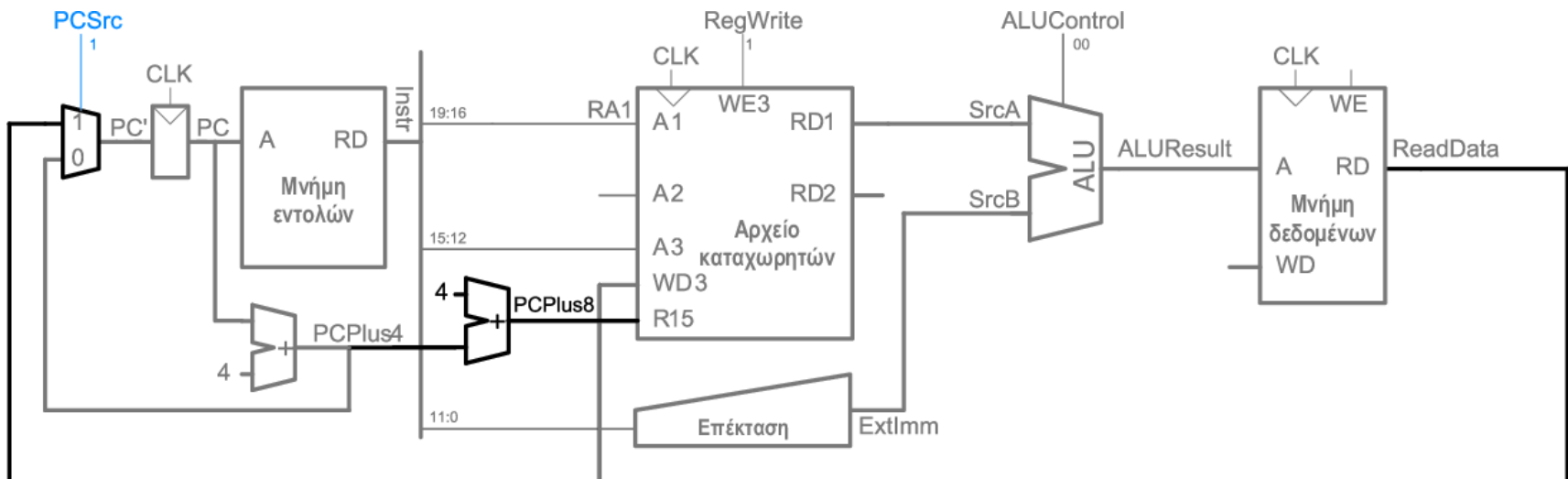
Προσαύξηση του μετρητή PC

ΒΗΜΑ 6: Υπολογισμός της διεύθυνσης της επόμενης εντολής



Διαδρομή δεδομένων ενός κύκλου: Προσπέλαση του μετρητή PC

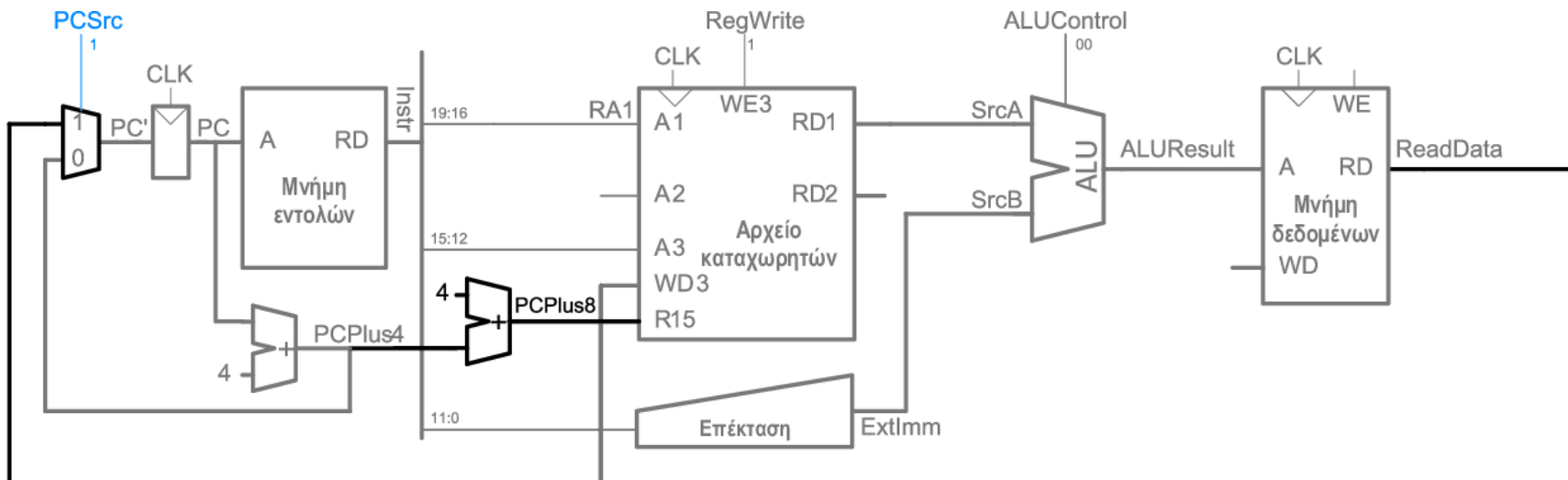
Ο μετρητής PC δείχνει την προέλευση/προορισμό της εντολής



Διαδρομή δεδομένων ενός κύκλου: Προσπέλαση του μετρητή PC

Ο μετρητής PC δείχνει την προέλευση/προορισμό της εντολής

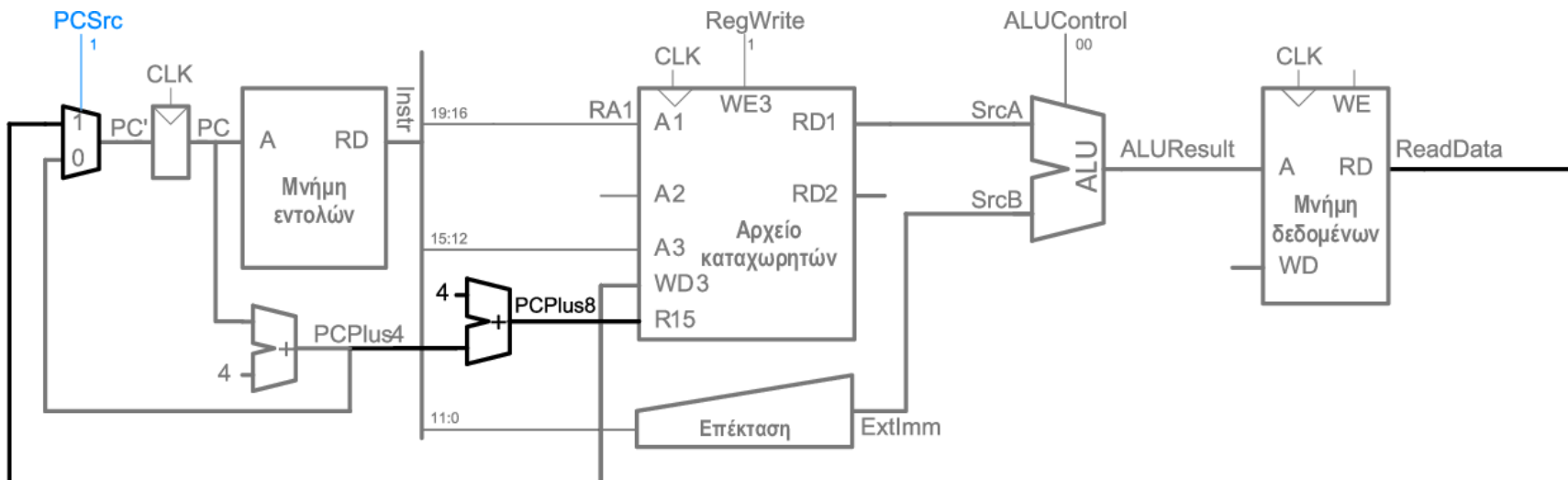
- **Προέλευση:** Ο R15 πρέπει να είναι διαθέσιμος στο αρχείο καταχωρητών
 - Ο μετρητής **PC** έχει τιμή ίση με το τρέχον **PC συν 8**



Διαδρομή δεδομένων ενός κύκλου: Προσπέλαση του μετρητή PC

Ο μετρητής PC δείχνει την προέλευση/προορισμό της εντολής

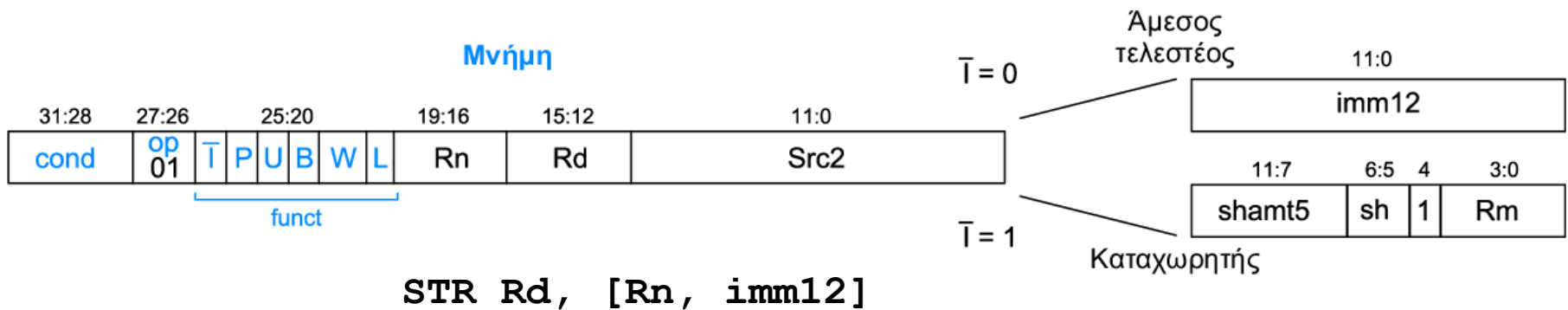
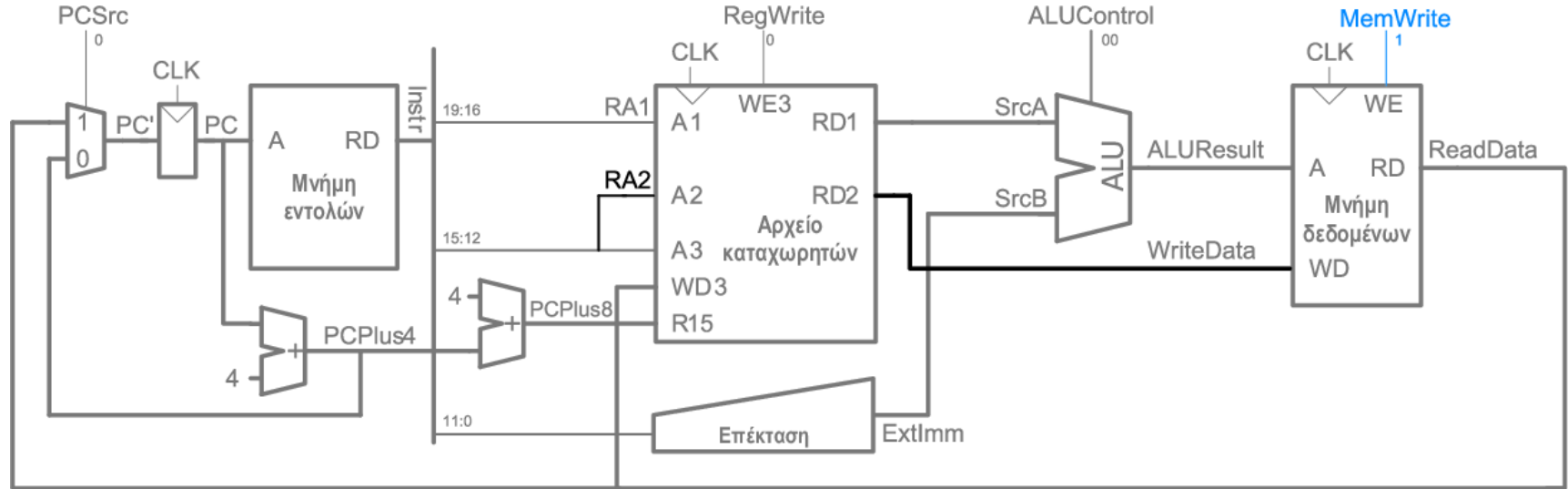
- **Προέλευση:** Ο R15 πρέπει να είναι διαθέσιμος στο αρχείο καταχωρητών
 - Ο μετρητής PC έχει τιμή ίση με το τρέχον PC συν 8
- **Προορισμός:** Πρέπει να είναι εφικτή η εγγραφή του αποτελέσματος στον μετρητή PC



Διαδρομή δεδομένων ενός κύκλου: Εντολή STR

Θα **επεκτείνουμε τη διαδρομή δεδομένων** για να χειρίζεται και την εντολή STR:

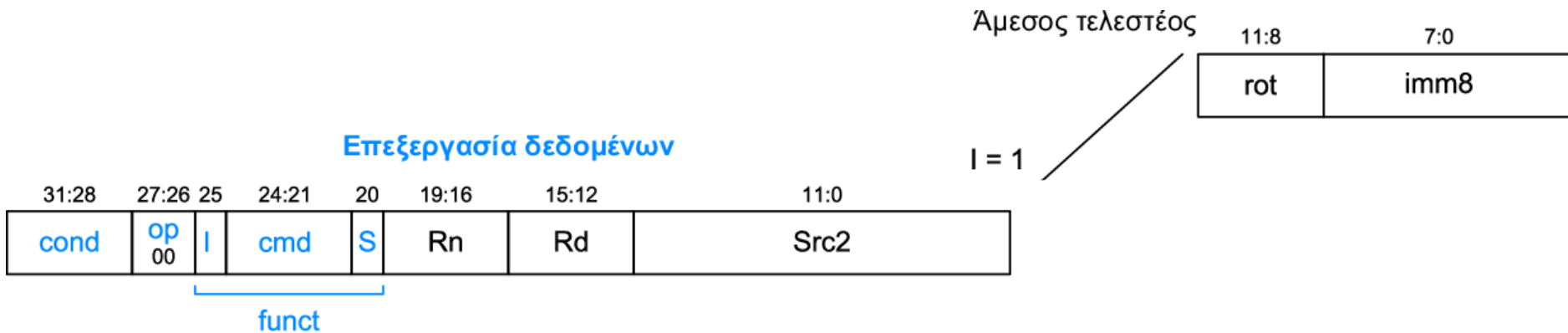
- Εγγραφή δεδομένων στη μνήμη από τον καταχωρητή που καθορίζεται στο πεδίο Rd



Διαδρομή δεδομένων ενός κύκλου: Επεξεργασία δεδομένων

Με άμεσο τελεστέο ως το πεδίο *Src2*:

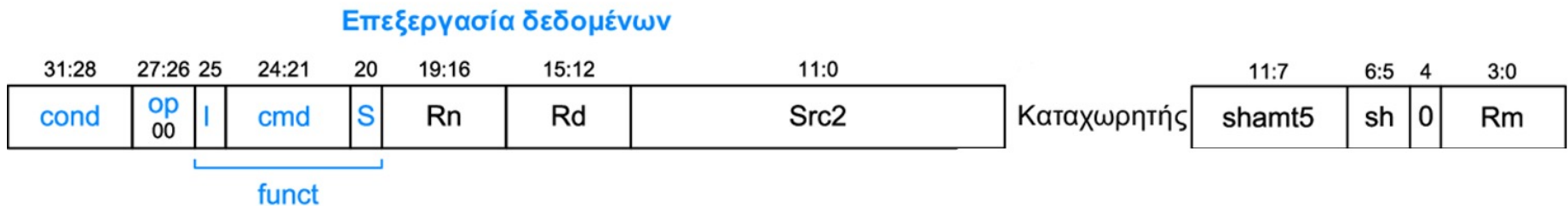
- Ανάγνωση από τα πεδία *Rn* και *Imm8* (το σήμα ελέγχου *ImmSrc* επιλέγει το επεκτεταμένο με μηδενικά πεδίο *Imm8* αντί του *Imm12*)
- Εγγραφή του *ALUResult* στο αρχείο καταχωρητών
- Εγγραφή στον καταχωρητή που καθορίζεται στο πεδίο *Rd*



Διαδρομή δεδομένων ενός κύκλου: Επεξεργασία δεδομένων

Με καταχωρητή ως το πεδίο *Src2*:

- Ανάγνωση από τα πεδία *Rn* και *Rm* (αντί του *Imm8*)
- Εγγραφή του *ALUResult* στο αρχείο καταχωρητών
- Εγγραφή στον καταχωρητή που καθορίζεται στο πεδίο *Rd*



ADD *Rd*, *Rn*, *Rm*

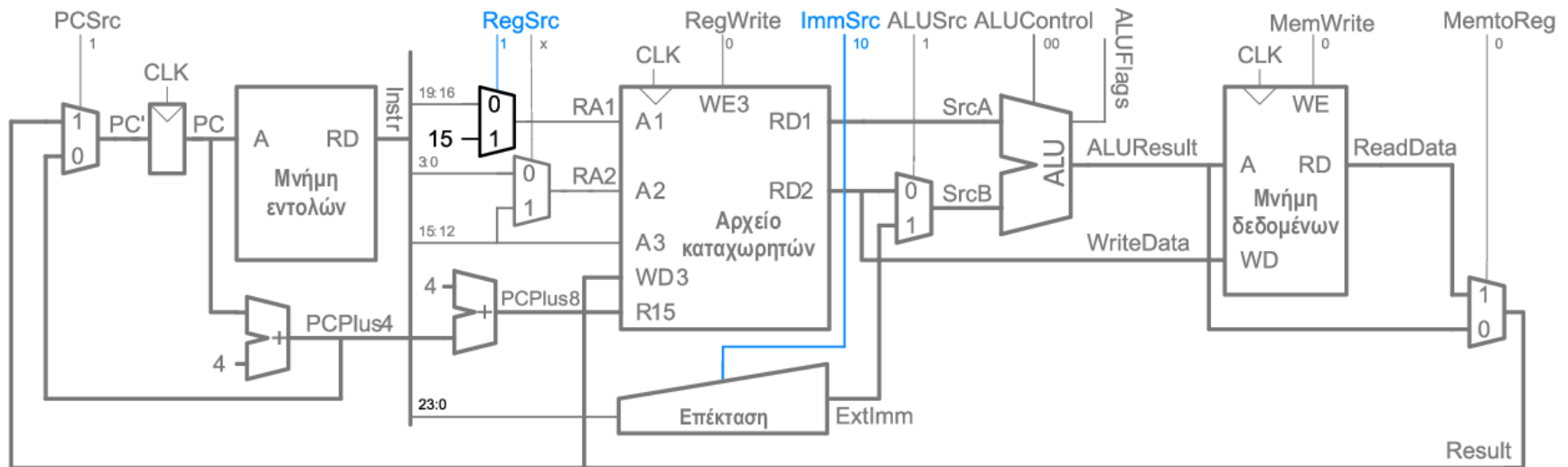


Διαδρομή δεδομένων ενός κύκλου: Εντολή B

Υπολογισμός της διεύθυνσης-στόχου της διακλάδωσης:

$$BTA = (ExtImm) + (PC + 8)$$

$ExtImm = Imm24 \ll 2$ και επέκταση προσήμου



Διακλάδωση



funct

Ετικέτα B

Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

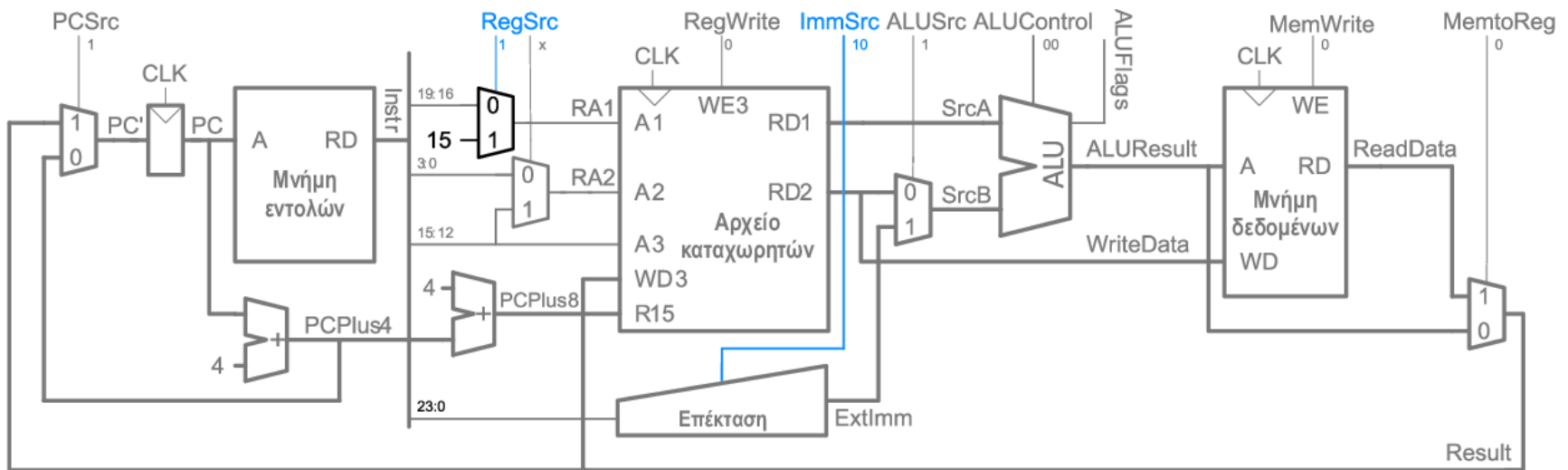
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <26>



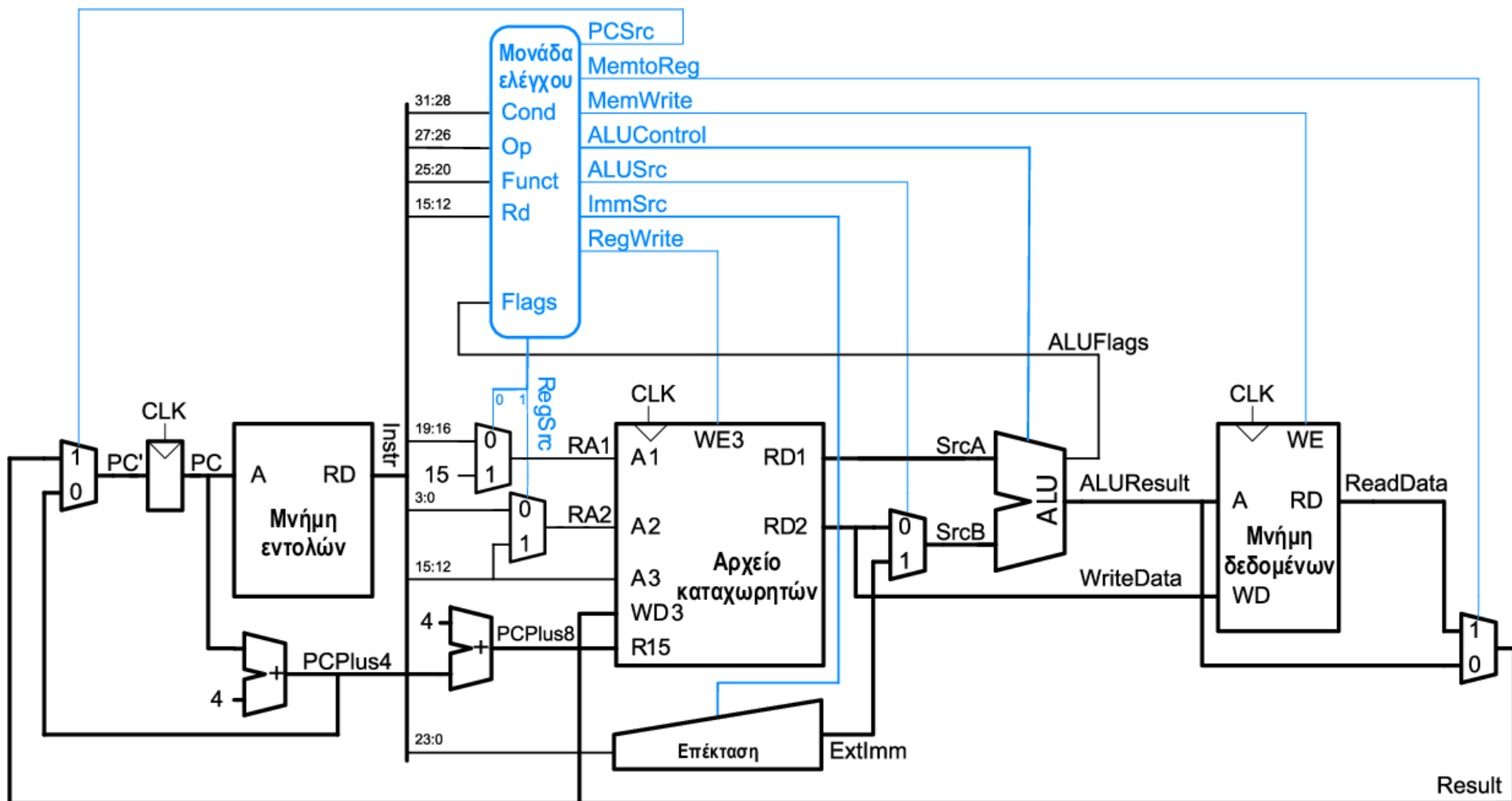
Διαδρομή δεδομένων ενός κύκλου: ExtImm



ImmSrc _{1:0}	ExtImm	Περιγραφή
00	{24'b0, Instr _{7:0} }	Το <i>imm8</i> επεκτεταμένο με μηδενικά
01	{20'b0, Instr _{11:0} }	Το <i>imm12</i> επεκτεταμένο με μηδενικά
10	{6{Instr ₂₃ }, Instr _{23:0} }	Το <i>imm24</i> με επέκταση προσήμου



Επεξεργαστής ARM ενός κύκλου



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

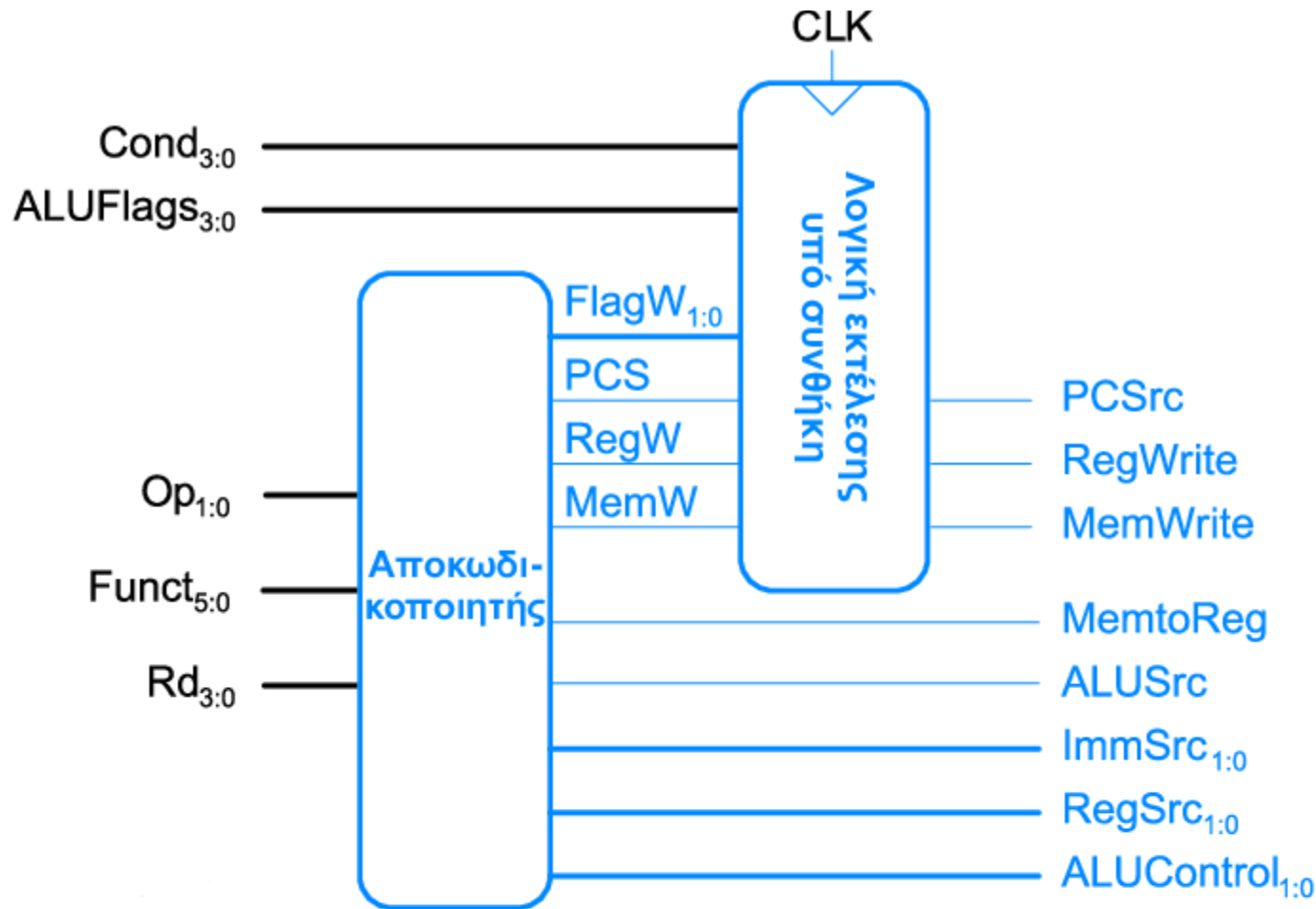
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτότυπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

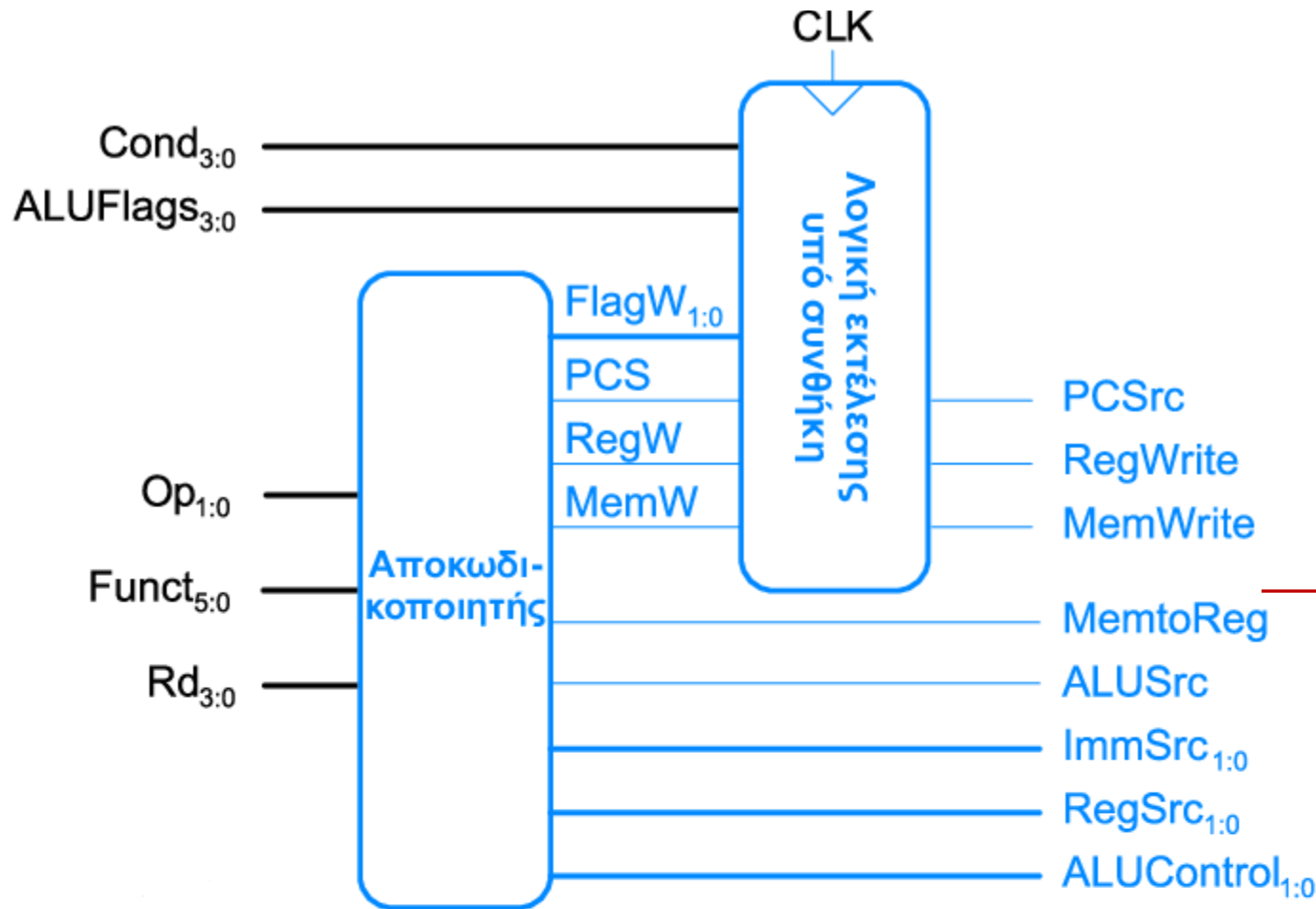
Κεφάλαιο 7 <28>



Μονάδα ελέγχου ενός κύκλου



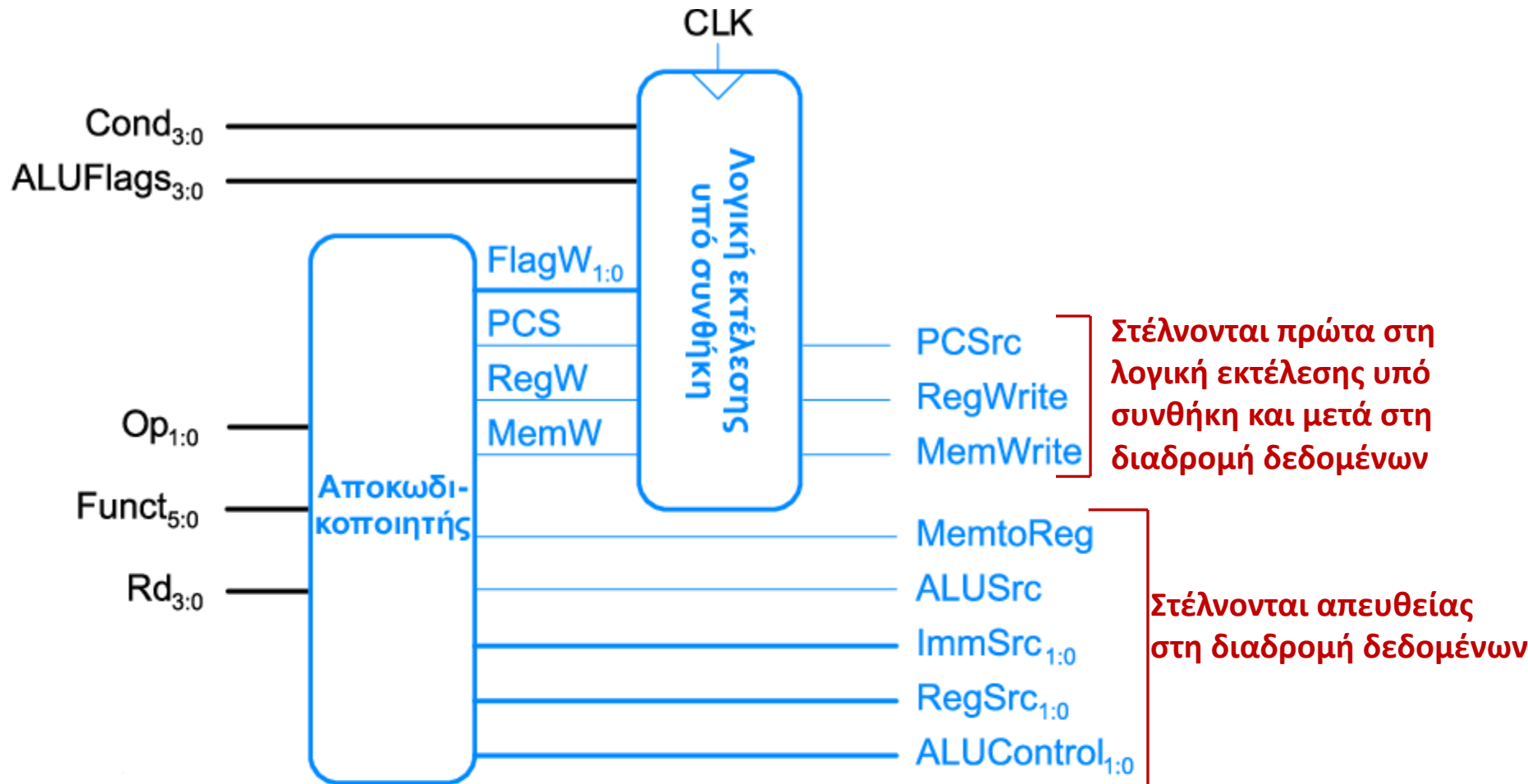
Μονάδα ελέγχου ενός κύκλου



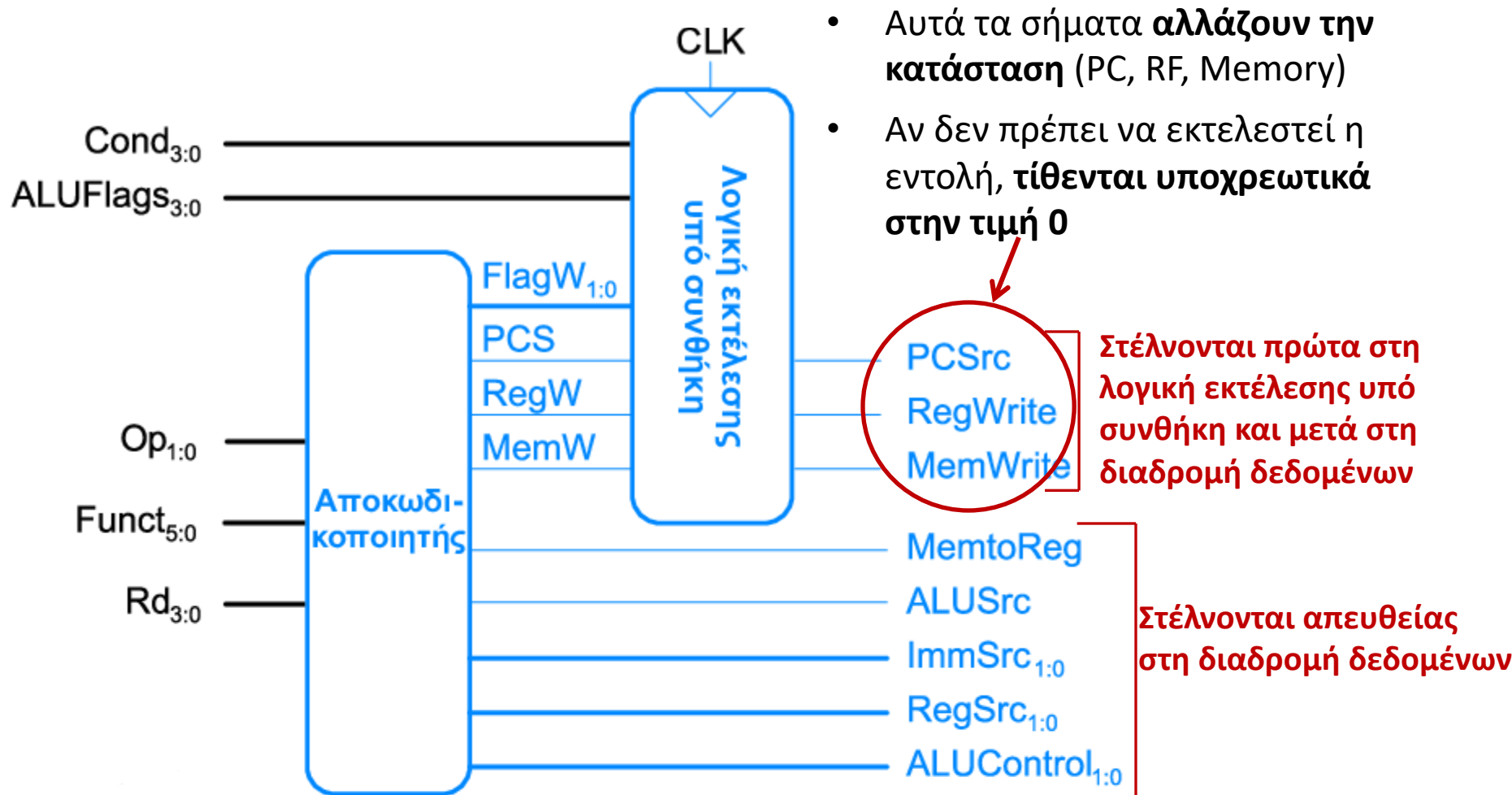
Στέλνονται απευθείας
στη διαδρομή δεδομένων



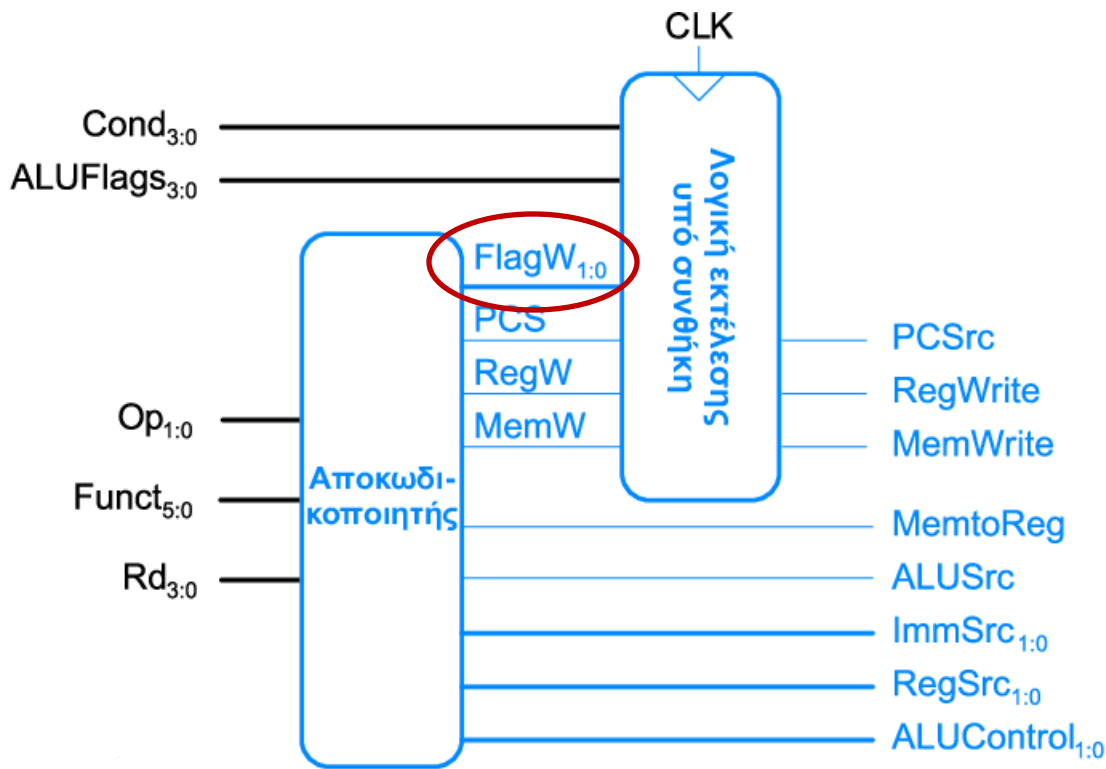
Μονάδα ελέγχου ενός κύκλου



Μονάδα ελέγχου ενός κύκλου



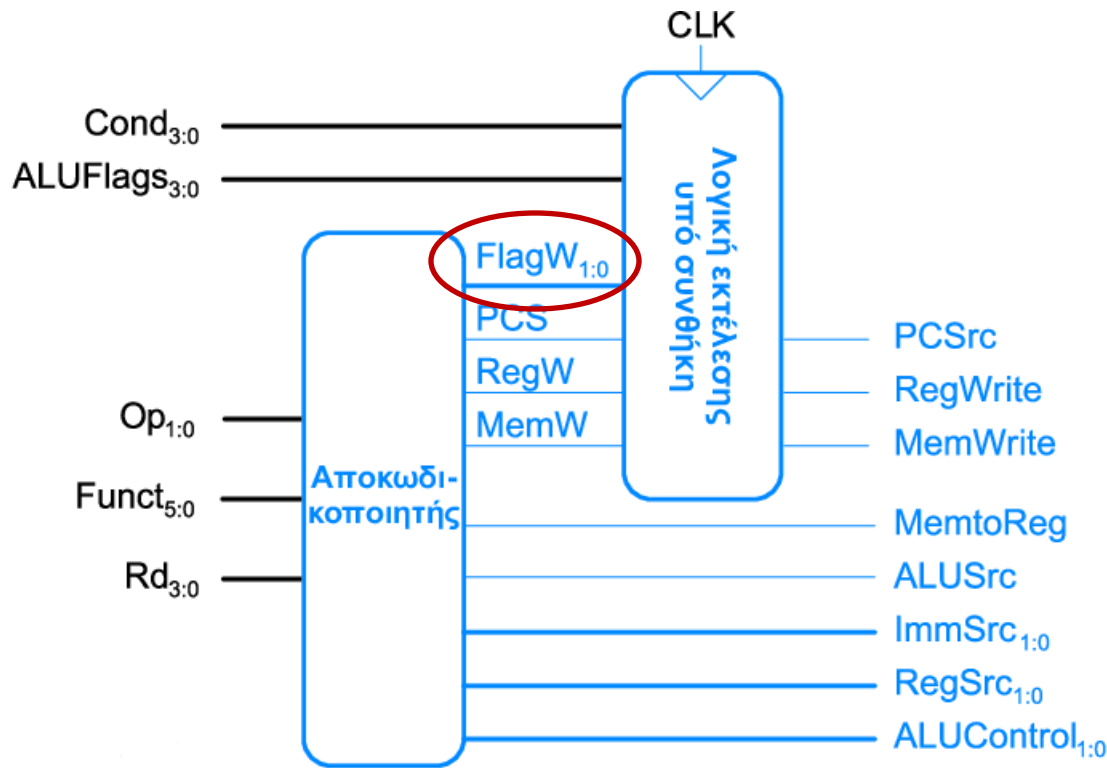
Μονάδα ελέγχου ενός κύκλου



- **FlagW_{1:0}**: Σήμα εγγραφής σημαιών, το οποίο ενεργοποιείται όταν πρέπει να αποθηκευτεί το *ALUFlags* (δηλαδή σε εντολή με *S = 1*)



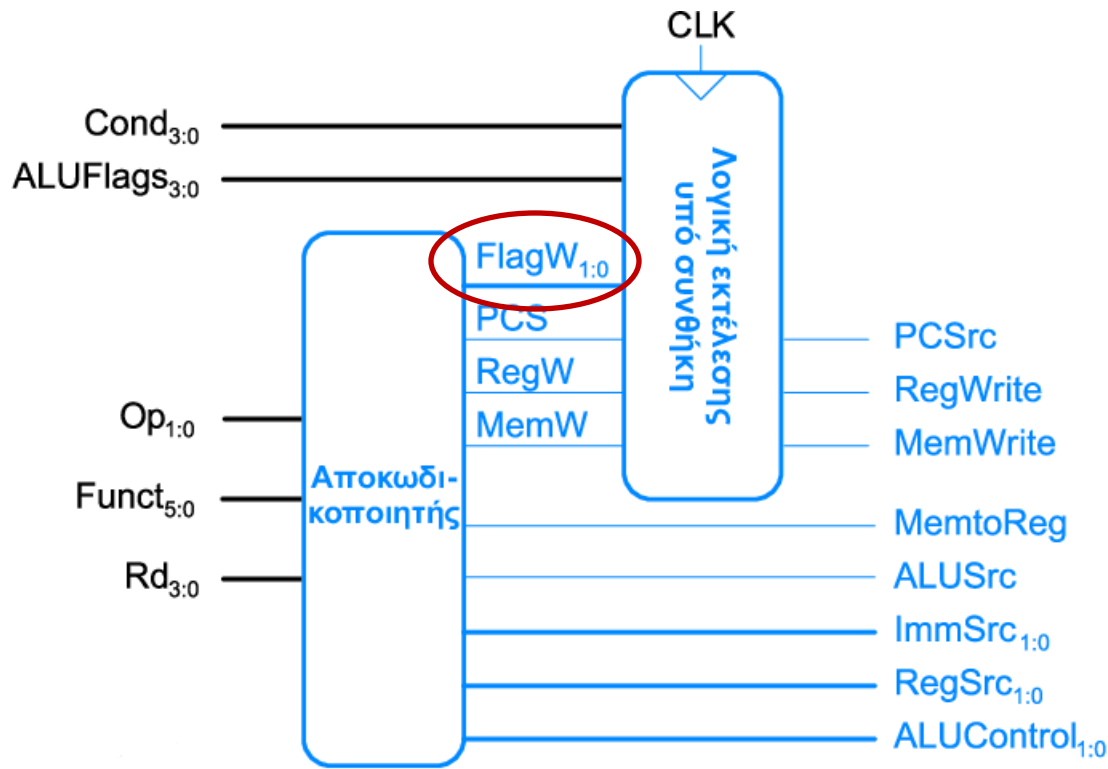
Μονάδα ελέγχου ενός κύκλου



- **FlagW_{1:0}**: Σήμα εγγραφής σημαιών, το οποίο ενεργοποιείται όταν πρέπει να αποθηκευτεί το *ALUFlags* (δηλαδή σε εντολή με *S = 1*)
- Οι ADD, SUB ενημερώνουν όλες τις σημαίες (**NZCV**)
- Οι AND, ORR ενημερώνουν μόνο τις σημαίες **NZ**



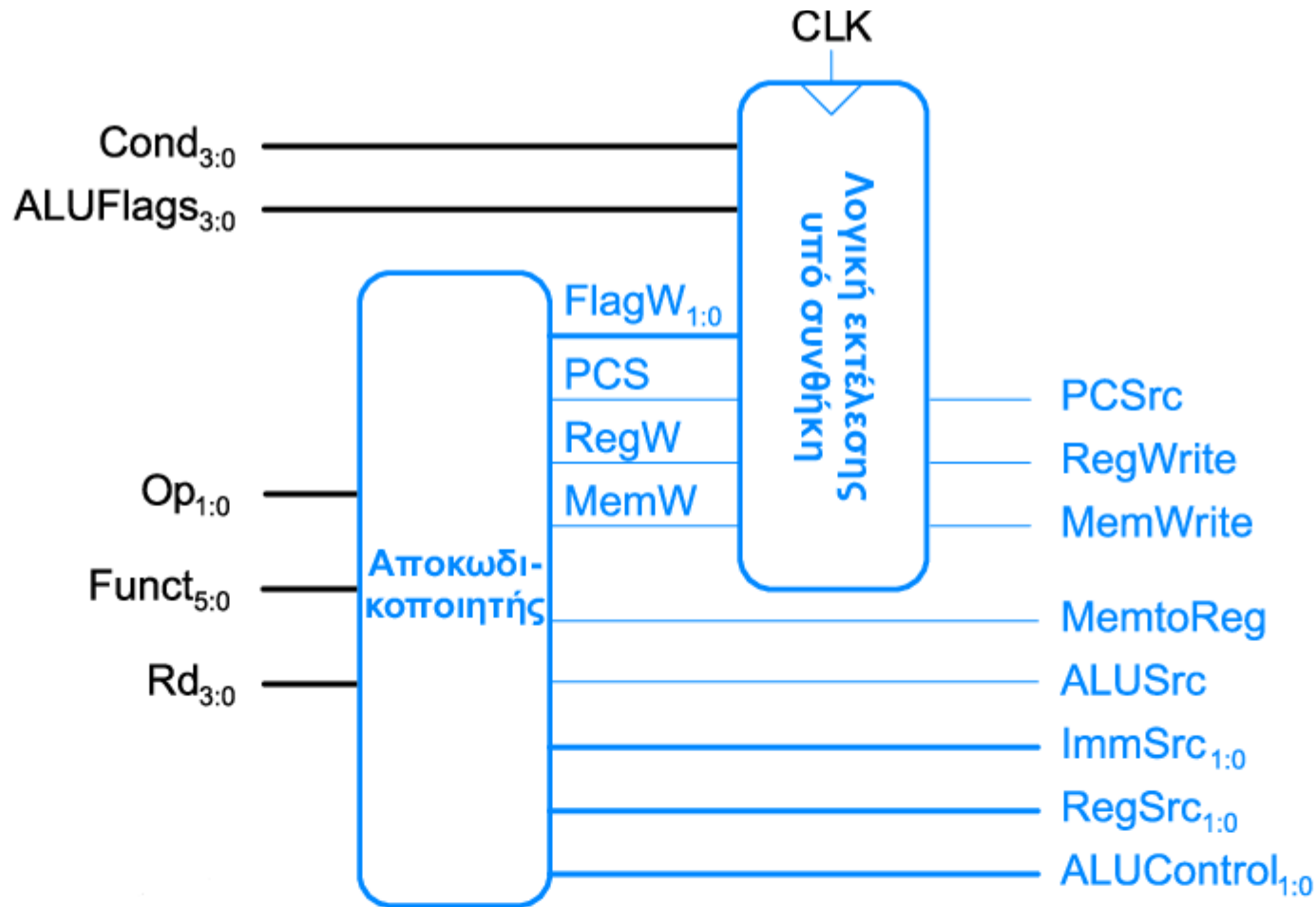
Μονάδα ελέγχου ενός κύκλου



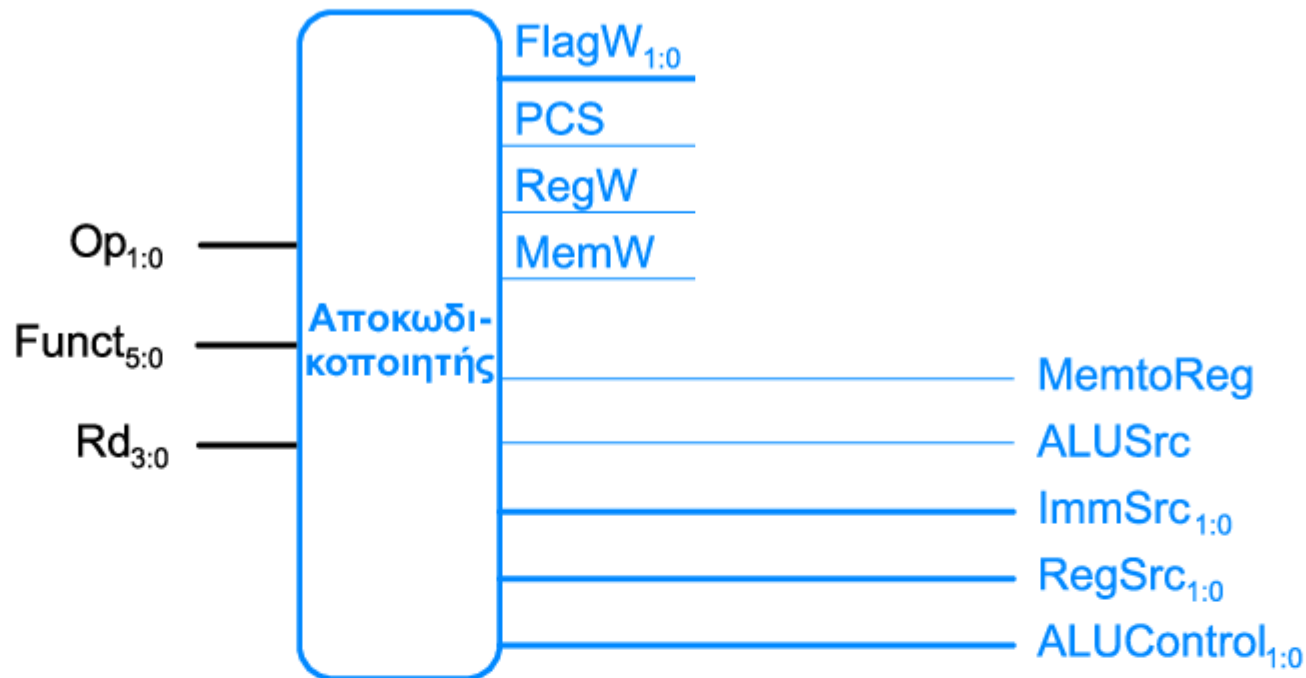
- **FlagW_{1:0}**: Σήμα εγγραφής σημαίων, το οποίο ενεργοποιείται όταν πρέπει να αποθηκευτεί το *ALUFlags* (δηλαδή σε εντολή με *S = 1*)
- Οι ADD, SUB ενημερώνουν όλες τις σημαίες (**NZCV**)
- Οι AND, ORR ενημερώνουν μόνο τις σημαίες **NZ**
- Άρα, απαιτούνται δύο bit:
 - **FlagW₁ = 1**: Αποθηκεύονται οι *NZ* (αποθηκεύεται το *ALUFlags_{3:2}*)
 - **FlagW₀ = 1**: Αποθηκεύονται οι *CV* (αποθηκεύεται το *ALUFlags_{1:0}*)



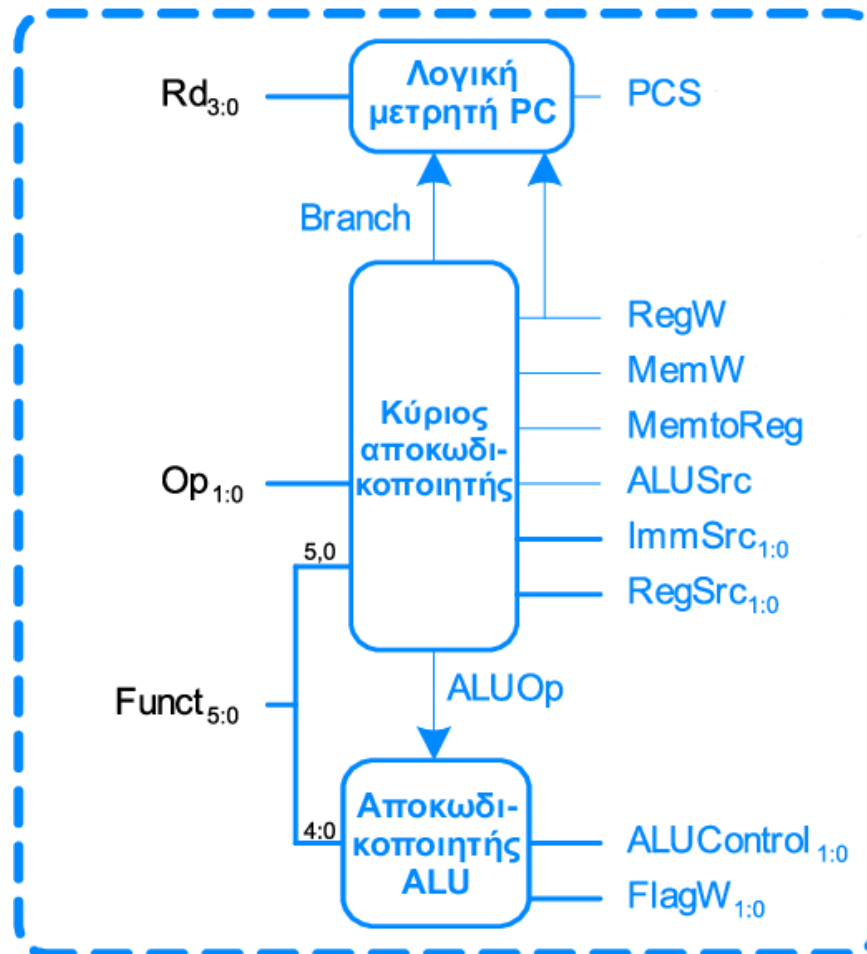
Μονάδα ελέγχου ενός κύκλου



Μονάδα ελέγχου ενός κύκλου: Αποκωδικοποιητής



Μονάδα ελέγχου ενός κύκλου: Αποκωδικοποιητής

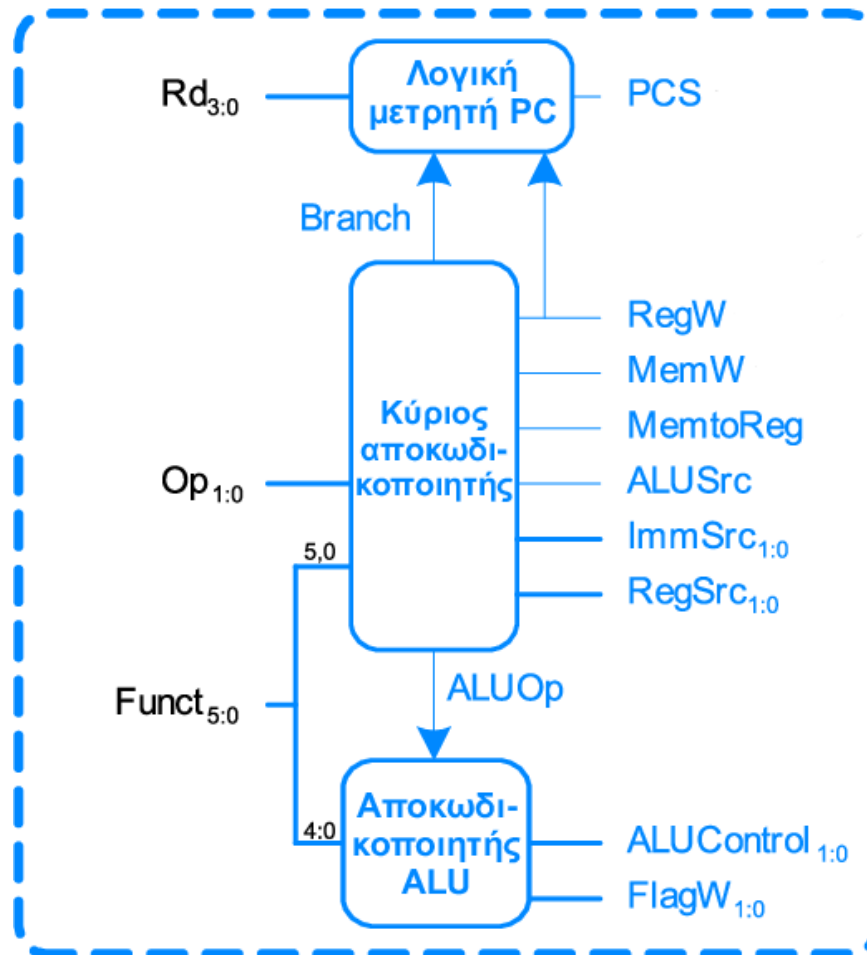


Υπομονάδες:

- Κύριος αποκωδικοποιητής
- Αποκωδικοποιητής ALU
- Λογική μετρητή PC



Μονάδα ελέγχου ενός κύκλου: Αποκωδικοποιητής



Υπομονάδες:

- Κύριος αποκωδικοποιητής
- Αποκωδικοποιητής ALU
- Λογική μετρητή PC

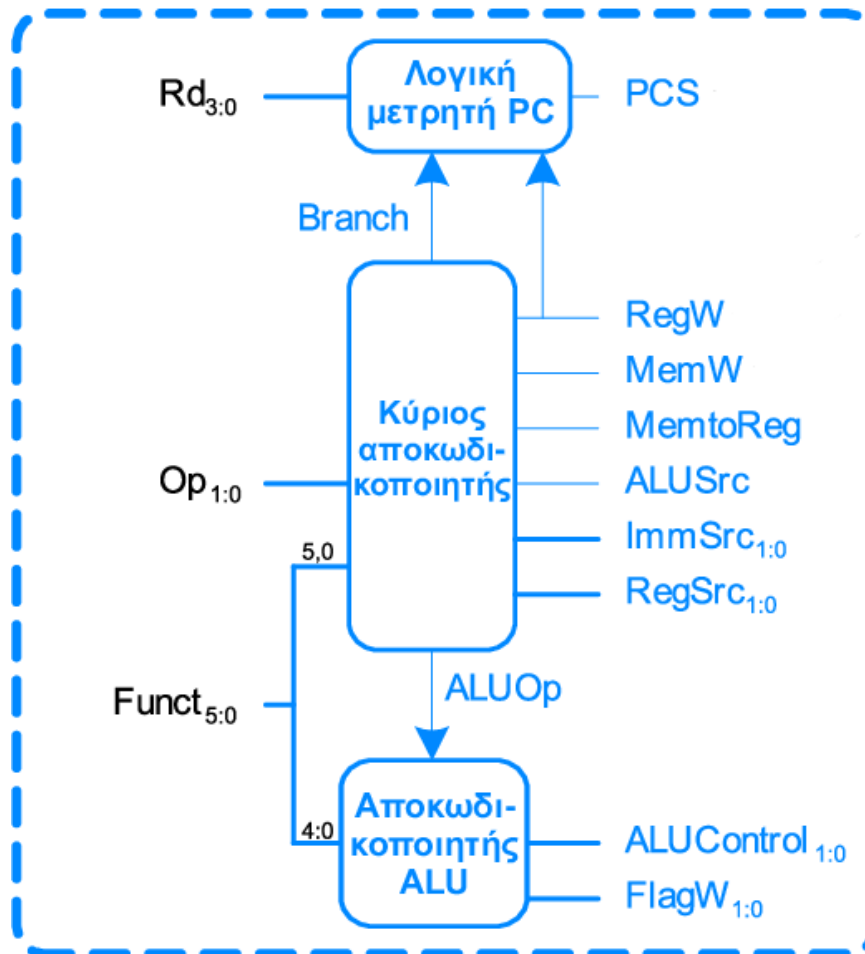


Μονάδα ελέγχου: Κύριος αποκωδικοποιητής

Op	Func ₅	Func ₀	Τύπος	Branch	MemtoReg	MemW	ALUSrc	ImmSrc	RegW	RegSrc	ALUOp
00	0	X	DP Reg	0	0	0	0	XX	1	00	1
00	1	X	DP Imm	0	0	0	1	00	1	X0	1
01	X	0	STR	0	X	1	1	01	0	10	0
01	X	1	LDR	0	1	0	1	01	1	X0	0
11	X	X	B	1	0	0	1	10	0	X1	0



Μονάδα ελέγχου ενός κύκλου: Αποκωδικοποιητής



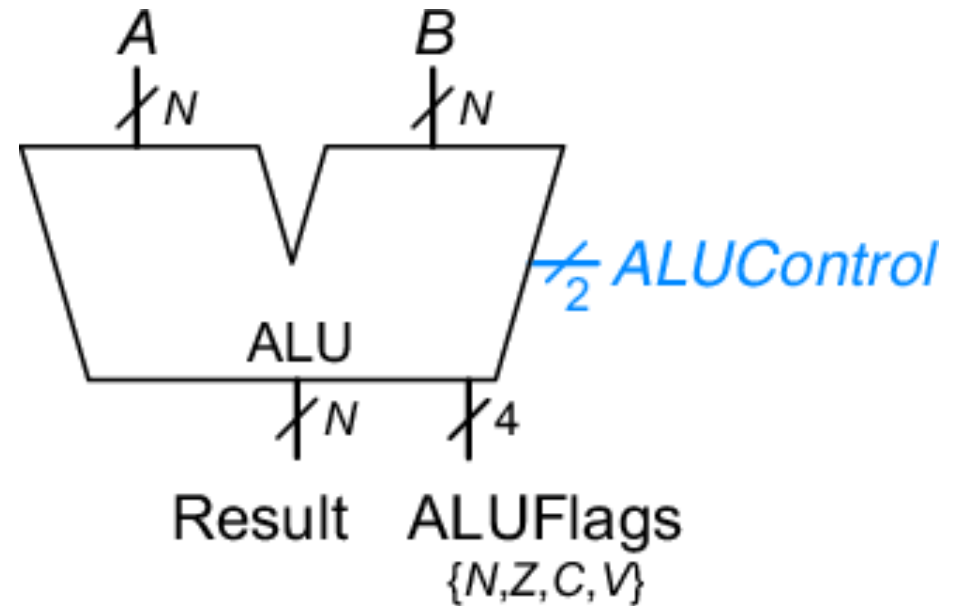
Υπομονάδες:

- Κύριος αποκωδικοποιητής
- **Αποκωδικοποιητής ALU**
- Λογική μετρητή PC

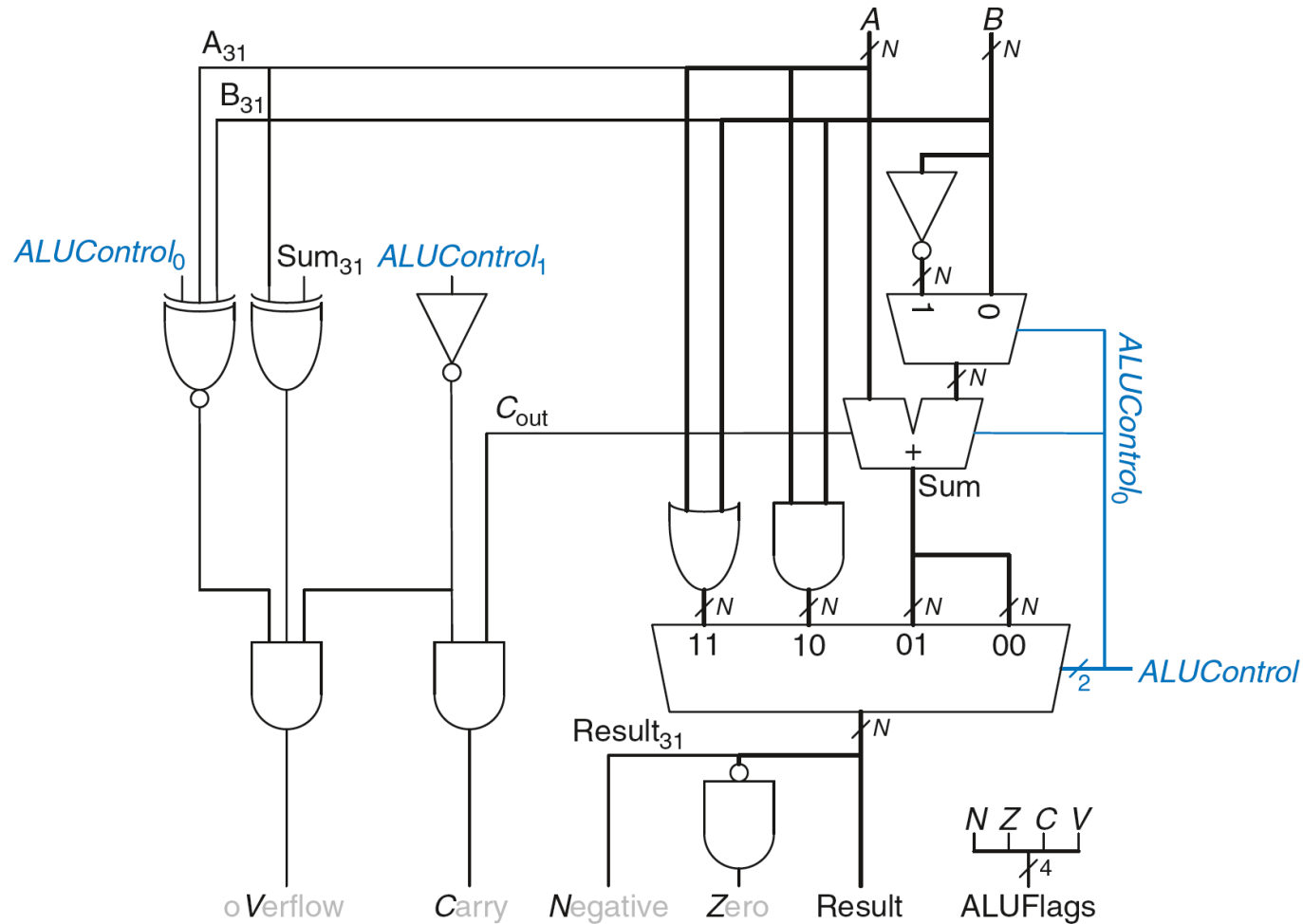


Επανάληψη: Μονάδα ALU

$ALUControl_{1:0}$	Πράξη
00	Add
01	Subtract
10	AND
11	OR



Επανάληψη: Μονάδα ALU



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

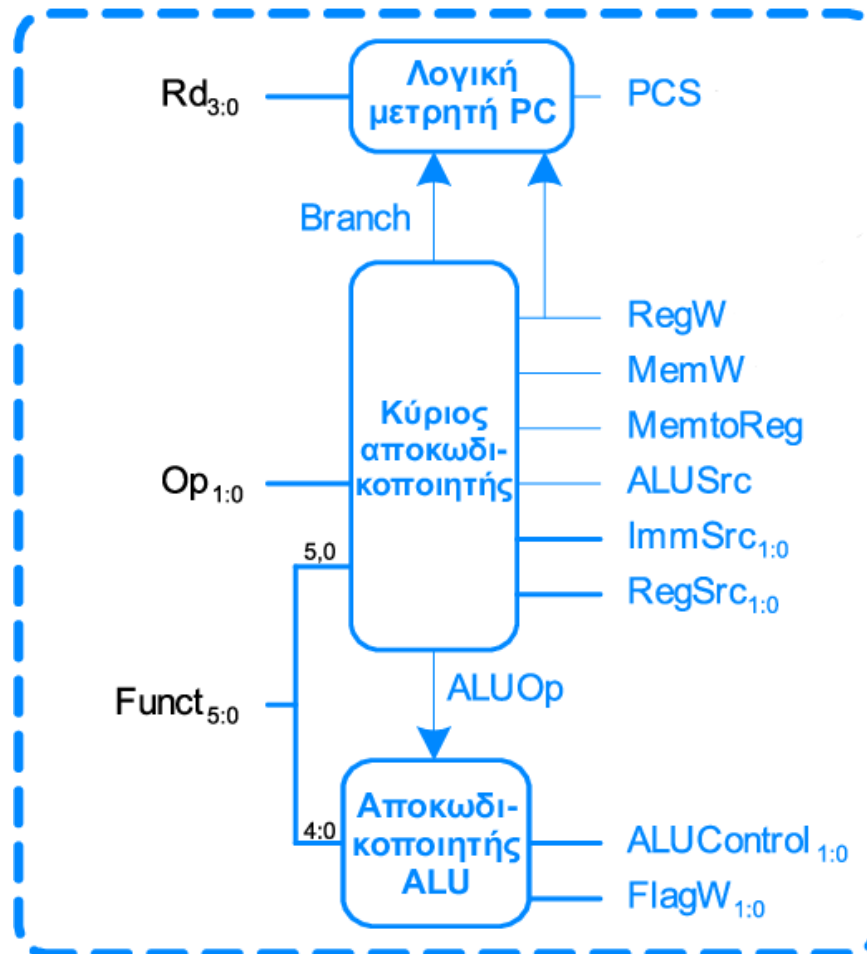
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <43>



Μονάδα ελέγχου ενός κύκλου: Αποκωδικοποιητής



Υπομονάδες:

- Κύριος αποκωδικοποιητής
- **Αποκωδικοποιητής ALU**
- Λογική μετρητή PC



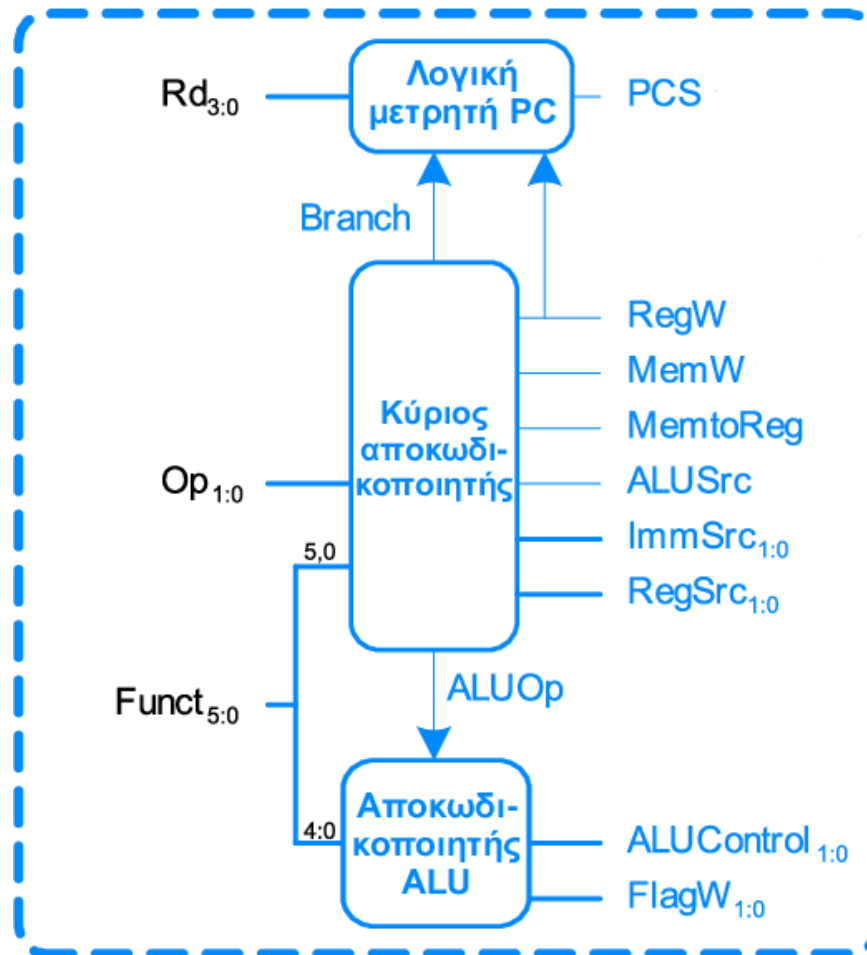
Μονάδα ελέγχου: Αποκωδικοποιητής ALU

<i>ALUOp</i>	<i>Funct</i> _{4:1} (<i>cmd</i>)	<i>Funct</i> ₀ (<i>S</i>)	Τύπος	<i>ALUControl</i> _{1:0}	<i>FlagW</i> _{1:0}
0	X	X	Όχι DP	00	00
1	0100	0	ADD	00	00
		1			11
	0010	0	SUB	01	00
		1			11
	0000	0	AND	10	00
		1			10
	1100	0	ORR	11	00
		1			10

- ***FlagW*₁** = 1: Πρέπει να αποθηκευτούν οι σημαίες *NZ* (*Flags*_{3:2})
- ***FlagW*₀** = 1: Πρέπει να αποθηκευτούν οι σημαίες *CV* (*Flags*_{1:0})



Μονάδα ελέγχου ενός κύκλου: Αποκωδικοποιητής

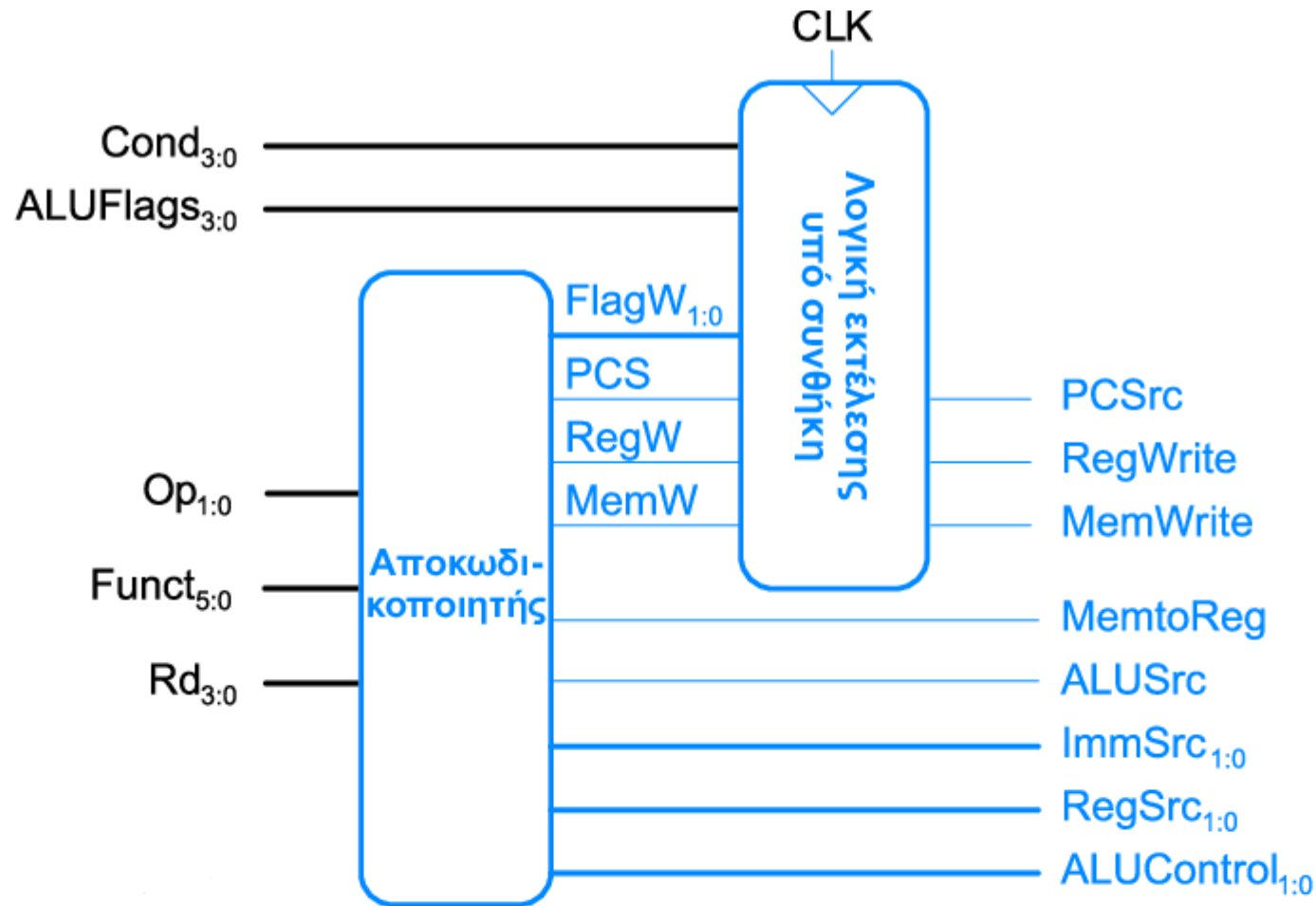


Υπομονάδες:

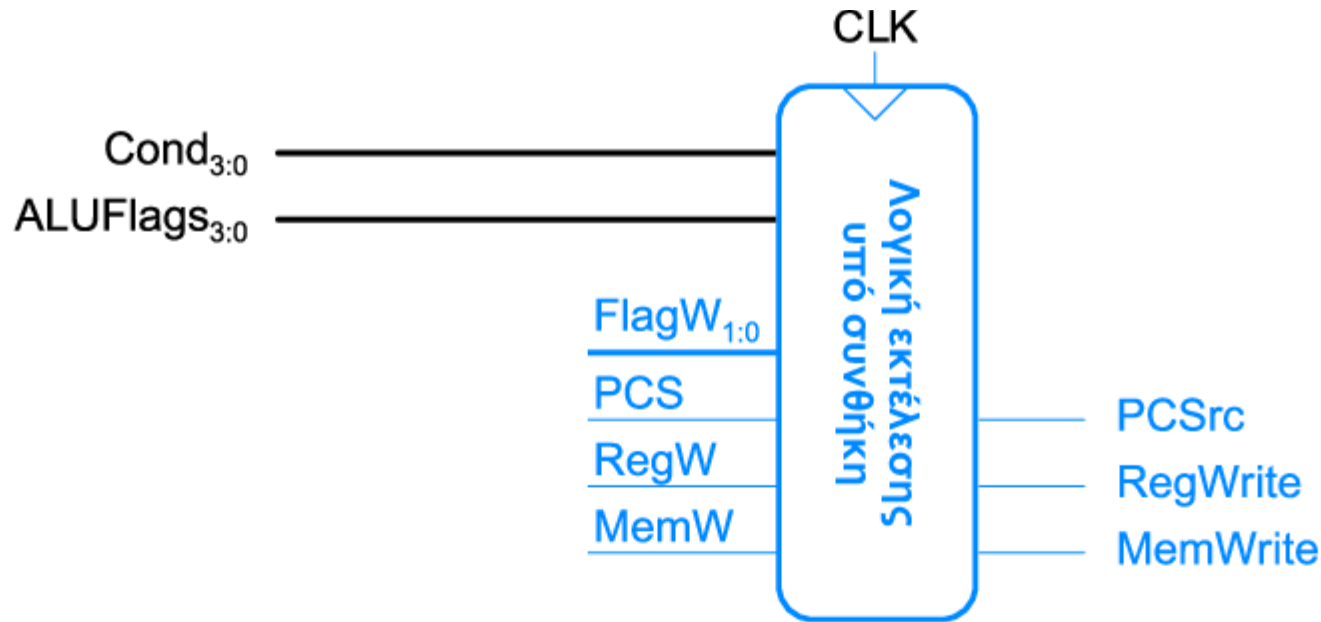
- Κύριος αποκωδικοποιητής
- Αποκωδικοποιητής ALU
- **Λογική μετρητή PC**



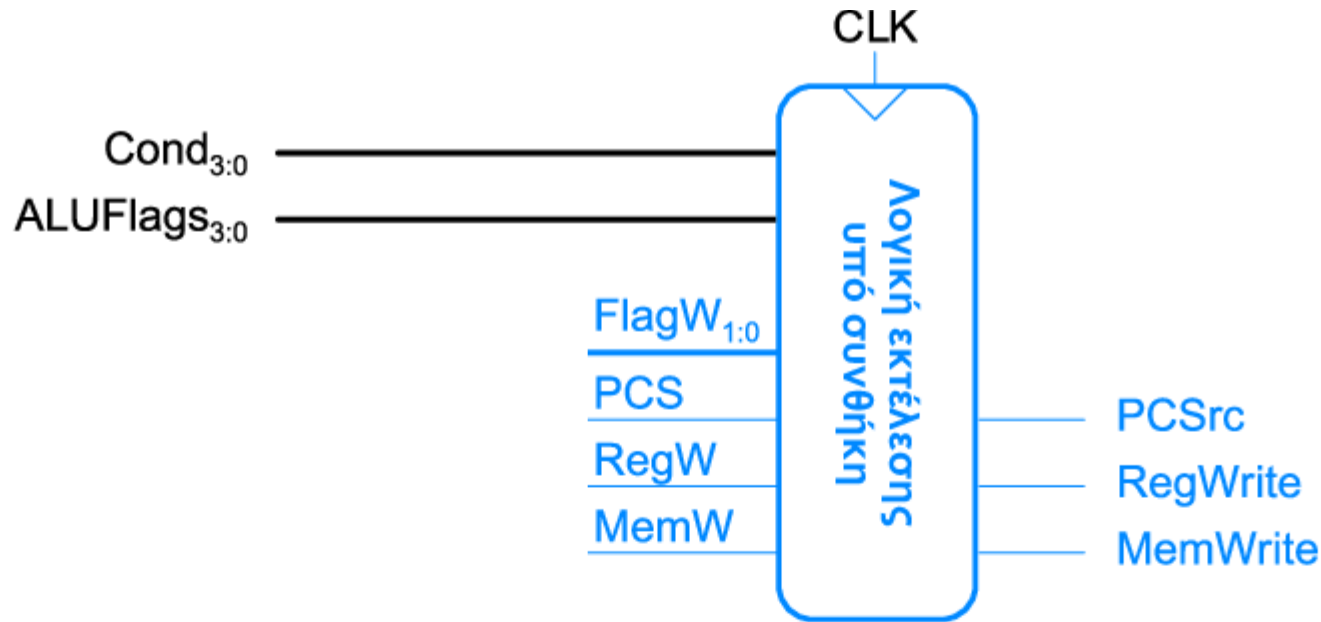
Μονάδα ελέγχου ενός κύκλου



Μονάδα ελέγχου ενός κύκλου: Λογική εκτέλεσης υπό συνθήκη



Λογική εκτέλεσης υπό συνθήκη

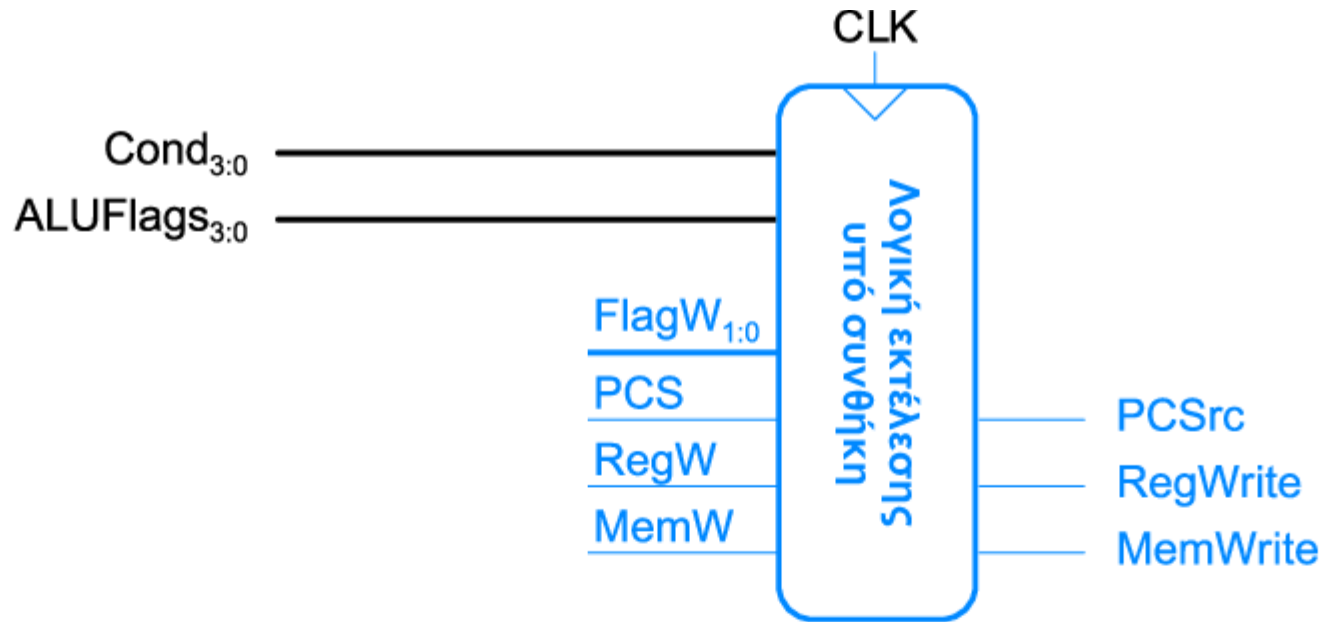


Λειτουργία:

1. Ελέγχει αν η εντολή πρέπει να εκτελεστεί (αν όχι, θέτει υποχρεωτικά τα PCSrc, RegWrite και MemWrite στην τιμή 0)
2. Ενδεχομένως ενημερώνει τον καταχωρητή κατάστασης (Flags_{3:0})



Λογική εκτέλεσης υπό συνθήκη

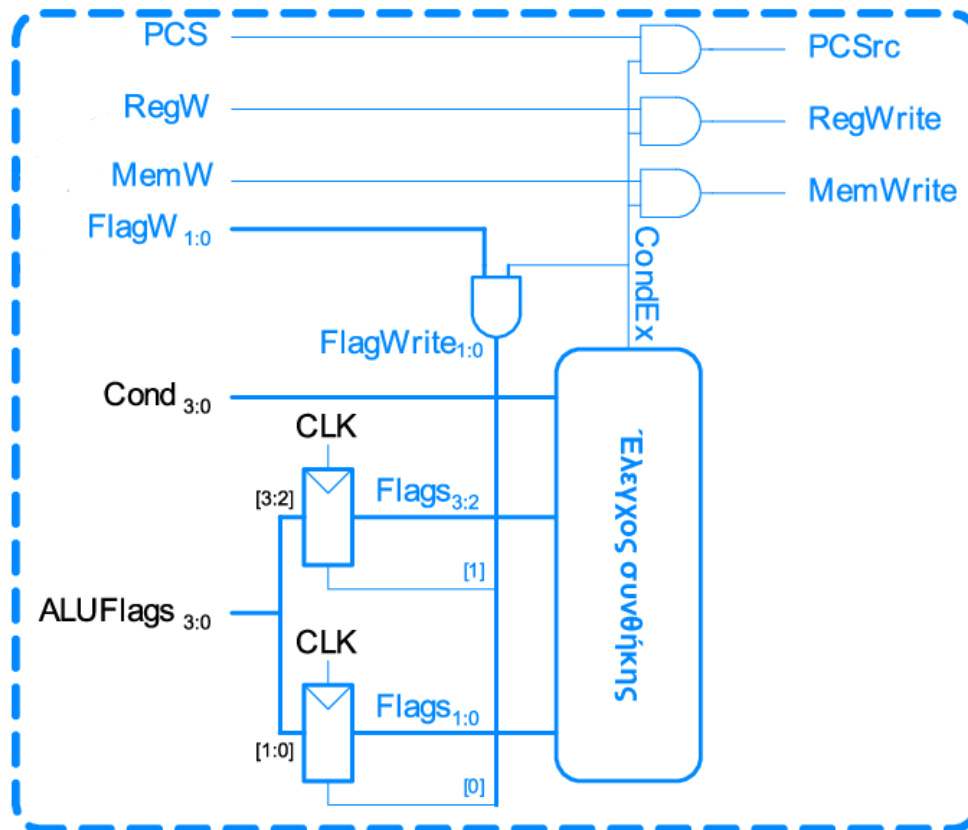


Λειτουργία:

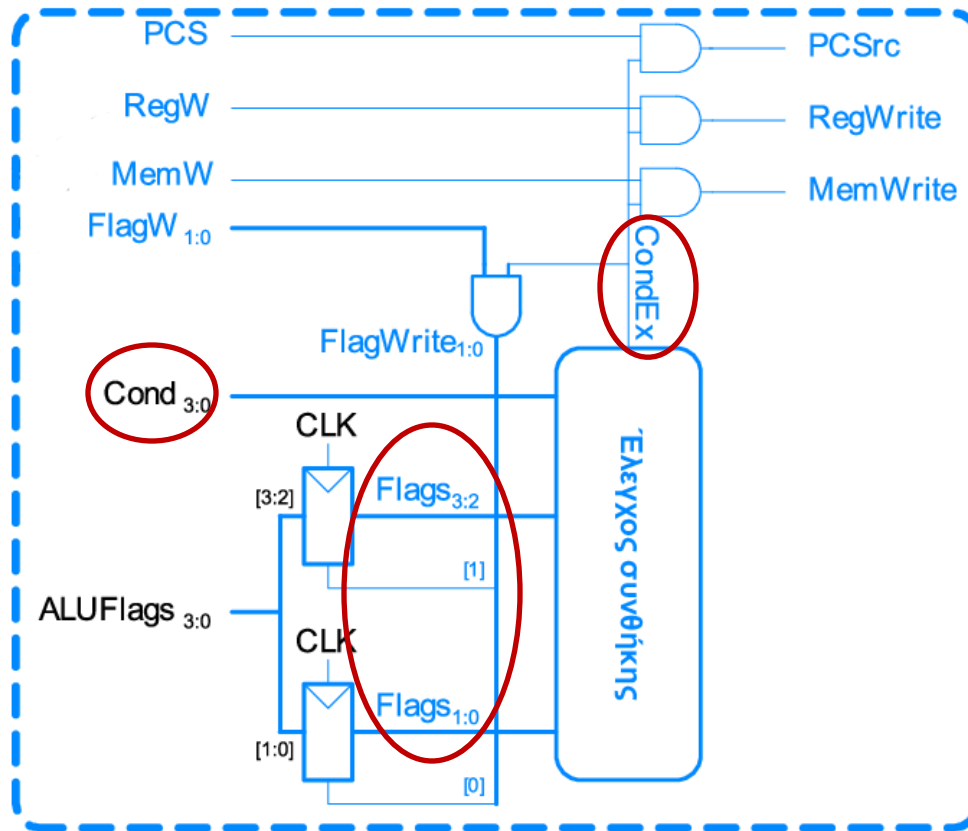
1. Ελέγχει αν η εντολή πρέπει να εκτελεστεί (αν όχι, θέτει υποχρεωτικά τα PCSrc, RegWrite και MemWrite στην τιμή 0)
2. Ενδεχομένως ενημερώνει τον καταχωρητή κατάστασης (Flags_{3:0})



Μονάδα ελέγχου ενός κύκλου: Λογική εκτέλεσης υπό συνθήκη



Λογική εκτέλεσης υπό συνθήκη: Εκτέλεση υπό συνθήκη

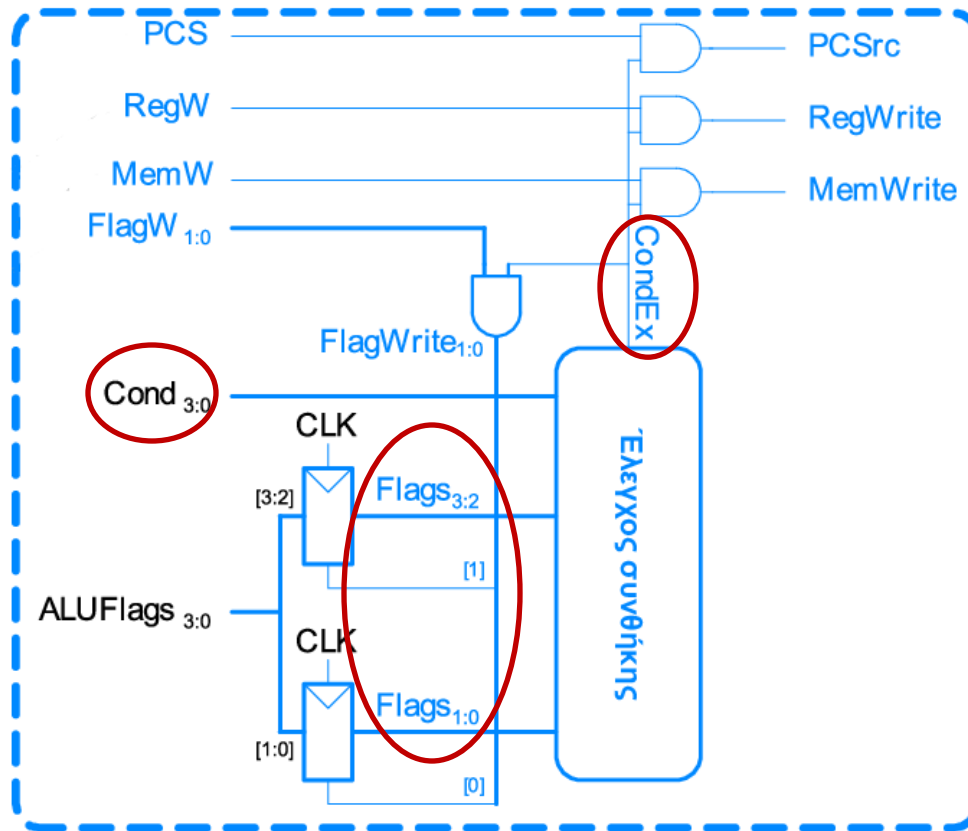


Ανάλογα με το μνημονικό συνθήκης ($Cond_{3:0}$) και τις σημαίες συνθήκης ($Flags_{3:0}$) η εντολή εκτελείται ($CondEx = 1$)



Λογική εκτέλεσης υπό συνθήκη: Εκτέλεση υπό συνθήκη

Το $Flags_{3:0}$ είναι
ο καταχωρητής
κατάστασης



Ανάλογα με το μνημονικό συνθήκης ($Cond_{3:0}$) και τις σημαίες συνθήκης ($Flags_{3:0}$) η εντολή εκτελείται ($CondEx = 1$)



Επανάληψη: Μνημονικά συνθήκης

<i>cond</i>	Μνημονικό	Όνομα	CondEx
0000	EQ	Equal (ίσοι)	Z
0001	NE	Not equal (μη ίσοι)	\bar{Z}
0010	CS / HS	Carry set / Unsigned higher or same (καθορισμός κρατουμένου / μη προσημασμένος υψηλότερος ή ίδιος)	C
0011	CC / LO	Carry clear / Unsigned lower (επαναφορά κρατουμένου / μη προσημασμένος χαμηλότερος)	\bar{C}
0100	MI	Minus / Negative (μείον / αρνητικό)	N
0101	PL	Plus / Positive of zero	\bar{N}
0110	VS	Overflow / Overflow set (υπερχείλιση / ενεργοποίηση υπερχείλισης)	V
0111	VC	No overflow / Overflow clear (μη υπερχείλιση / επαναφορά υπερχείλισης)	\bar{V}
1000	HI	Unsigned higher (μη προσημασμένος υψηλότερος)	$\bar{Z}C$
1001	LS	Unsigned lower or same (μη προσημασμένος χαμηλότερος ή ίδιος)	$Z OR \bar{C}$
1010	GE	Signed greater than or equal (προσημασμένος μεγαλύτερος ή ίσος)	$\overline{N \oplus V}$
1011	LT	Signed less than (προσημασμένος μικρότερος)	$N \oplus V$
1100	GT	Signed greater than (προσημασμένος μεγαλύτερος)	$\bar{Z}(\overline{N \oplus V})$
1101	LE	Signed less than or equal (προσημασμένος μικρότερος ή ίσος)	$Z OR (N \oplus V)$
1110	AL (ή none)	Always / unconditional (πάντα / χωρίς συνθήκη)	Αγνοείται

Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

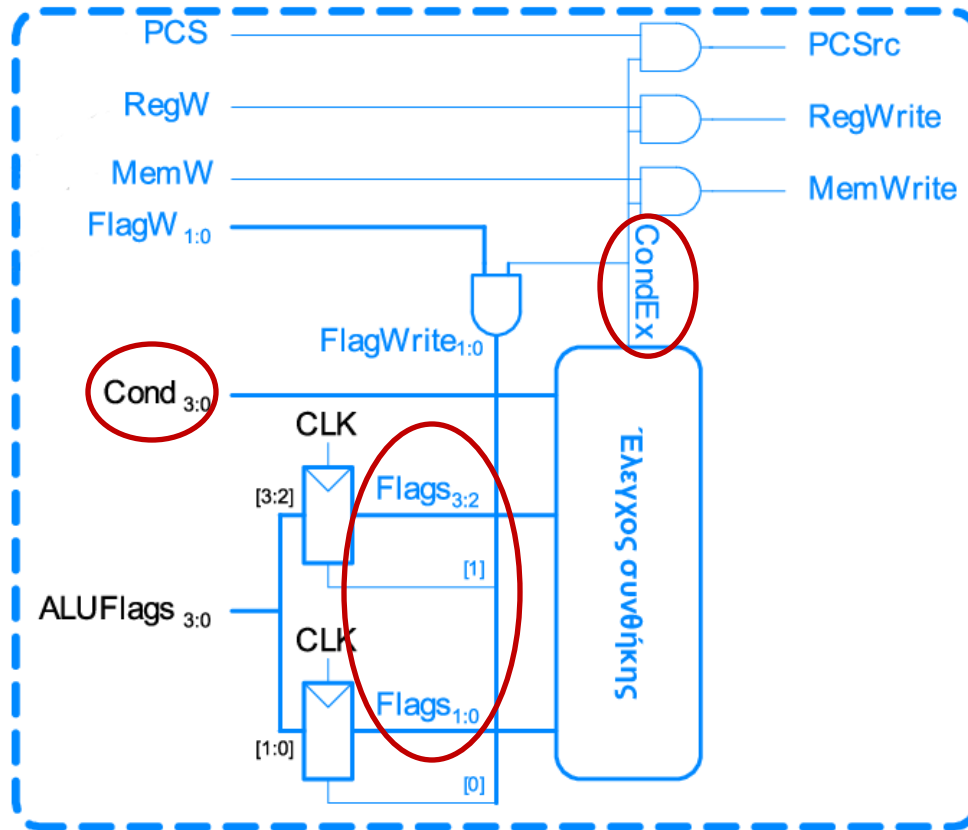
© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <55>



Λογική εκτέλεσης υπό συνθήκη: Εκτέλεση υπό συνθήκη

Flags_{3:0} = NZCV



Παράδειγμα:

AND R1, R2, R3

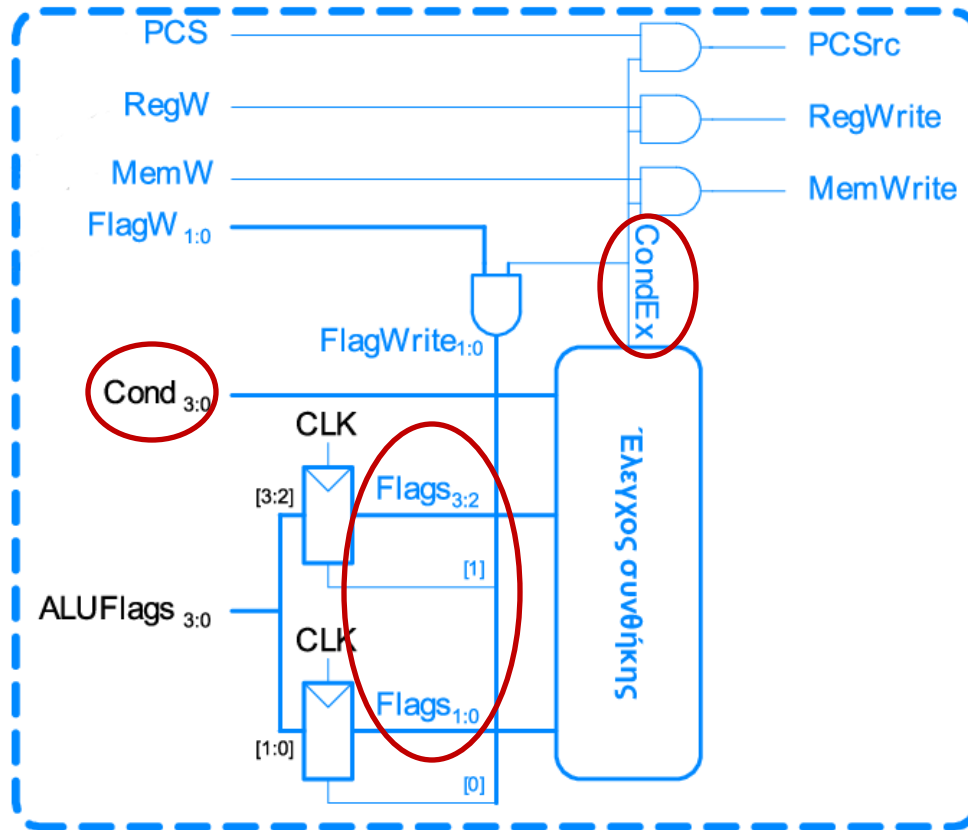
Cond_{3:0}=1110 (χωρίς συνθήκη)

=> CondEx = 1



Λογική εκτέλεσης υπό συνθήκη: Εκτέλεση υπό συνθήκη

Flags_{3:0} = NZCV



Παράδειγμα:

EOREQ R5, R6, R7

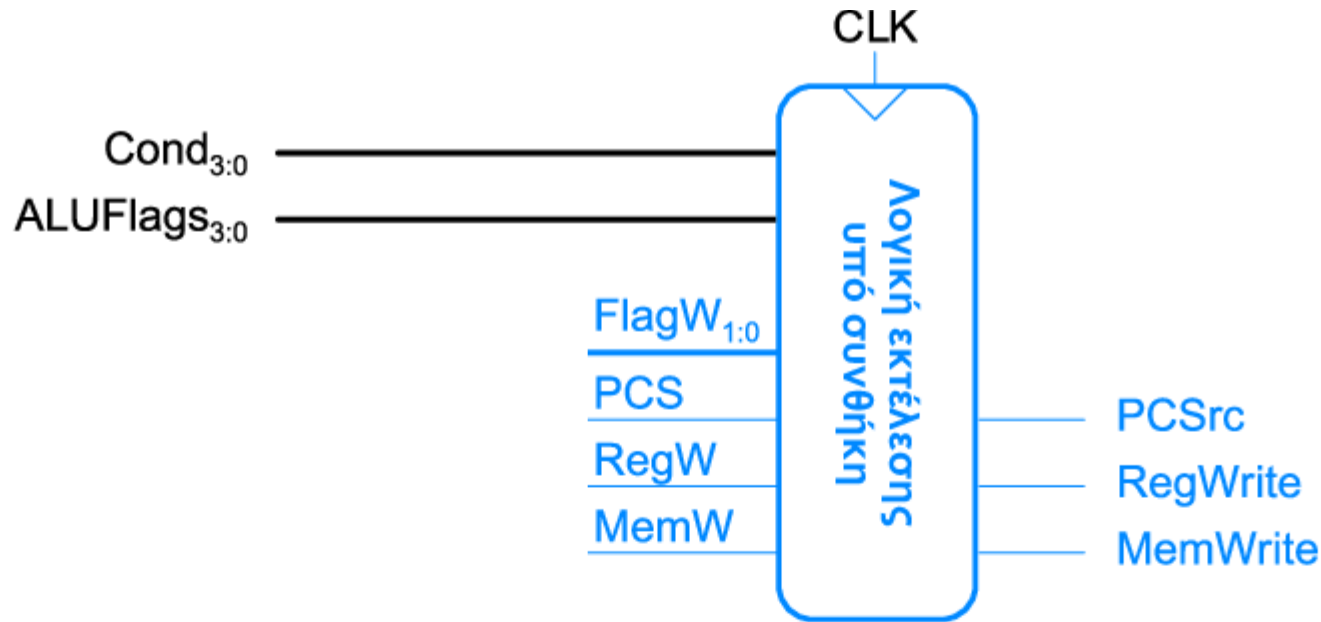
Cond_{3:0}=0000 (EQ):

αν Flags_{3:2}=0100

=> CondEx = 1



Λογική εκτέλεσης υπό συνθήκη



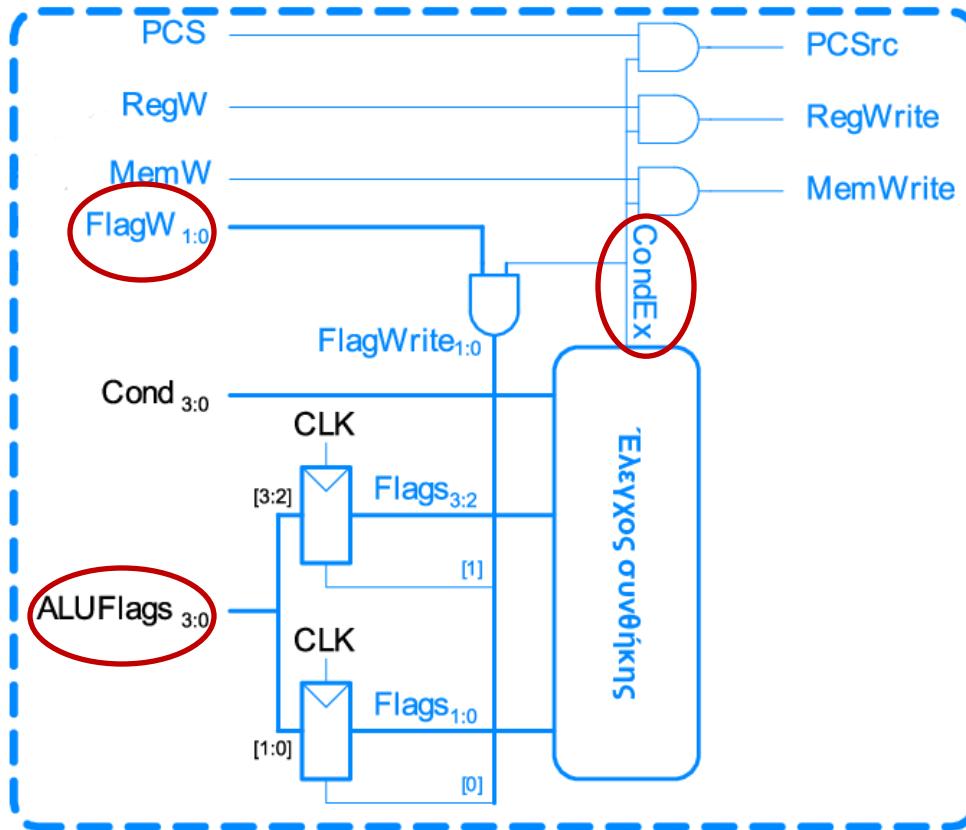
Λειτουργία:

1. Ελέγχει αν η εντολή πρέπει να εκτελεστεί (αν όχι, θέτει υποχρεωτικά τα PCSrc, RegWrite και MemWrite στην τιμή 0)
2. Ενδεχομένως ενημερώνει τον καταχωρητή κατάστασης (Flags_{3:0})



Λογική εκτέλεσης υπό συνθήκη: Ενημέρωση σημαιών (τιμή 1)

$\text{Flags}_{3:0} = \text{NZCV}$

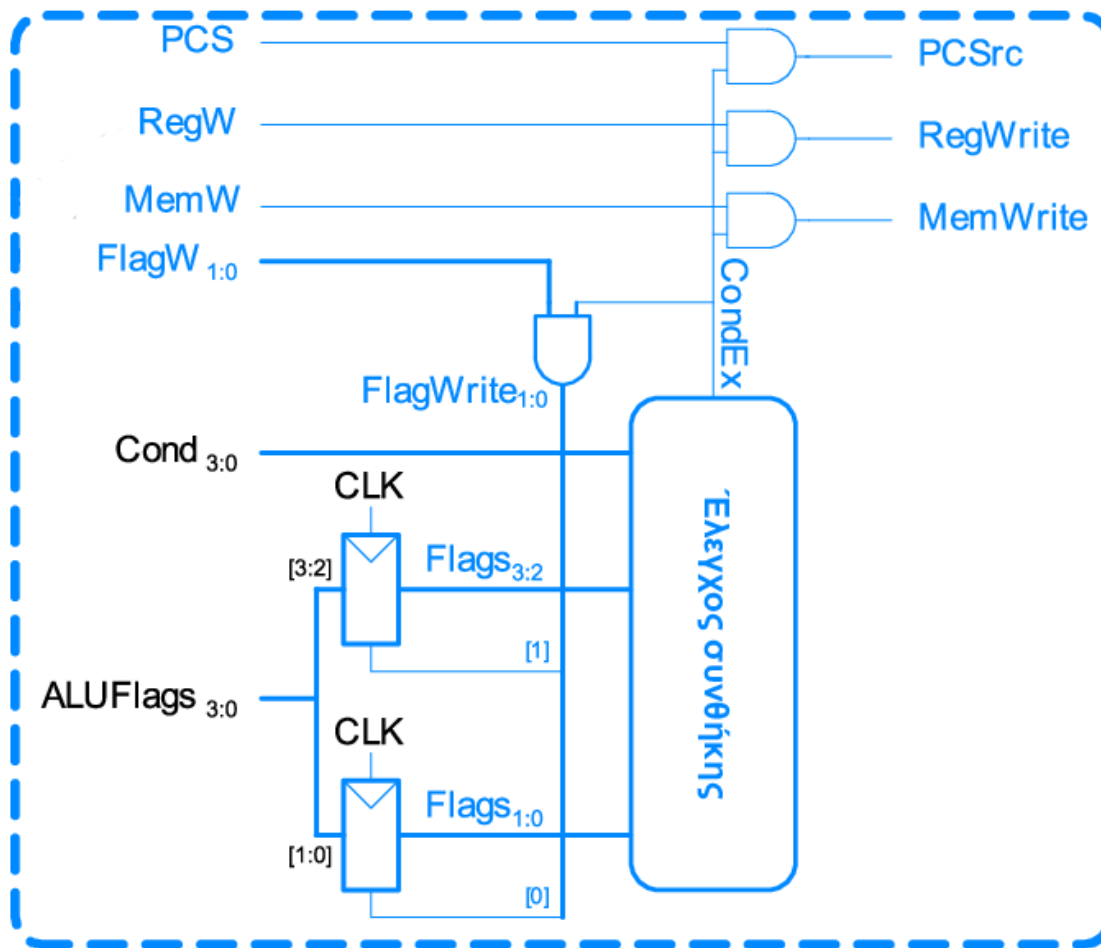


Το $\text{Flags}_{3:0}$ ενημερώνεται (με το $\text{ALUFlags}_{3:0}$) αν:

- Το **FlagW** έχει την τιμή 1 (δηλαδή το bit *S* της εντολής έχει την τιμή 1) AND
- Το **CondEx** έχει την τιμή 1 (η εντολή πρέπει να εκτελεστεί)



Λογική εκτέλεσης υπό συνθήκη: Ενημέρωση σημαιών (τιμή 1)



Υπενθύμιση:

- Οι ADD, SUB ενημερώνουν **όλες** τις σημαίες
- Οι AND, OR ενημερώνουν **μόνο** τις σημαίες **NZ**
- Άρα, ο καταχωρητής κατάστασης σημαιών έχει δύο εγκρίσεις εγγραφής: **FlagW_{1:0}**



Επανάληψη: Αποκωδικοποιητής ALU

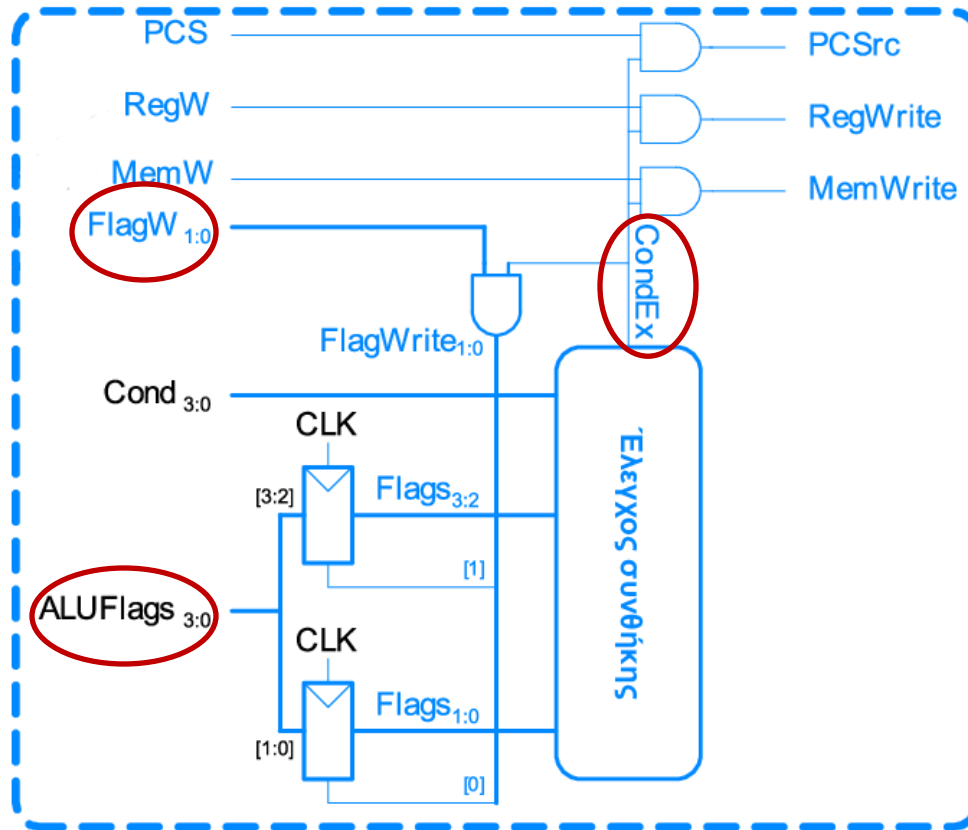
<i>ALUOp</i>	<i>Funct</i> _{4:1} (<i>cmd</i>)	<i>Funct</i> ₀ (<i>S</i>)	Τύπος	<i>ALUControl</i> _{1:0}	<i>FlagW</i> _{1:0}
0	X	X	Όχι DP	00	00
1	0100	0	ADD	00	00
		1			11
	0010	0	SUB	01	00
		1			11
	0000	0	AND	10	00
		1			10
	1100	0	ORR	11	00
		1			10

- $FlagW_1 = 1$: Πρέπει να αποθηκευτούν οι σημαίες *NZ* ($Flags_{3:2}$)
- $FlagW_0 = 1$: Πρέπει να αποθηκευτούν οι σημαίες *CV* ($Flags_{1:0}$)



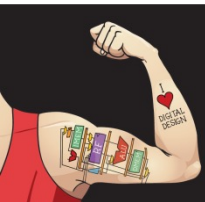
Λογική εκτέλεσης υπό συνθήκη: Ενημέρωση σημαιών (τιμή 1)

Ενημερώνονται
όλες οι σημαίες



Παράδειγμα: SUBS R5, R6, R7

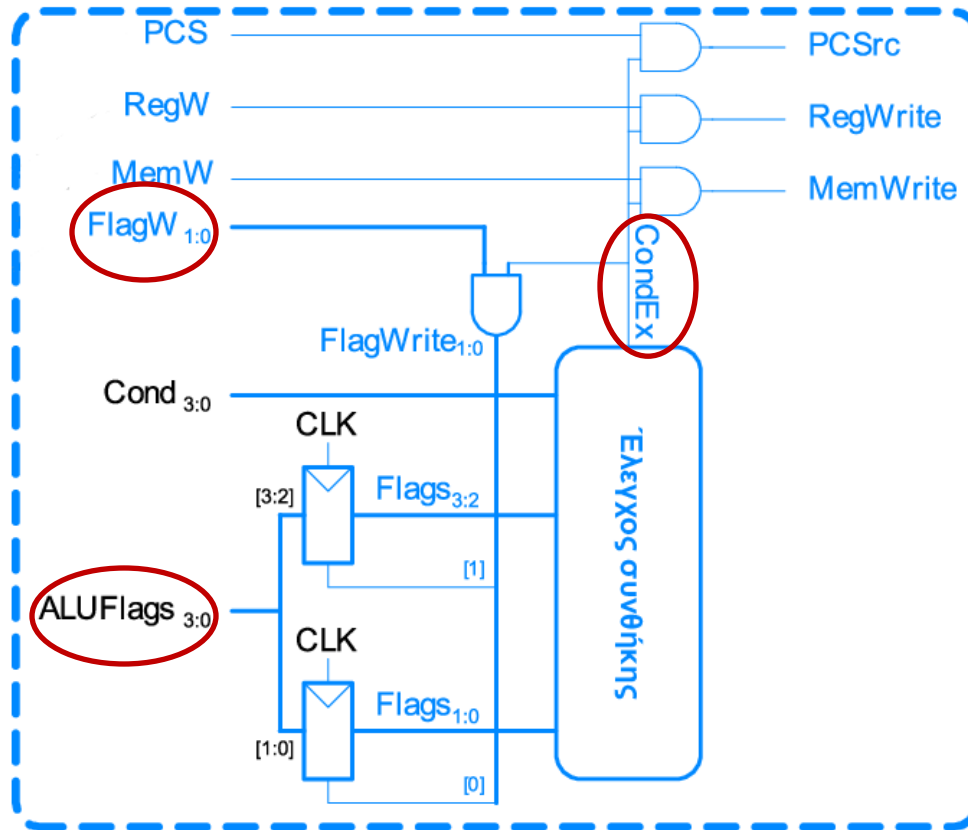
$\text{FlagW}_{1:0} = 11$ AND $\text{CondEx} = 1$ (χωρίς συνθήκη) $\Rightarrow \text{FlagWrite}_{1:0} = 11$



Λογική εκτέλεσης υπό συνθήκη: Ενημέρωση σημαιών (τιμή 1)

$\text{Flags}_{3:0} = \text{NZCV}$

- Ενημερώνεται μόνο το $\text{Flags}_{3:2}$
- Δηλαδή, ενημερώνονται μόνο οι σημαίες **NZ**



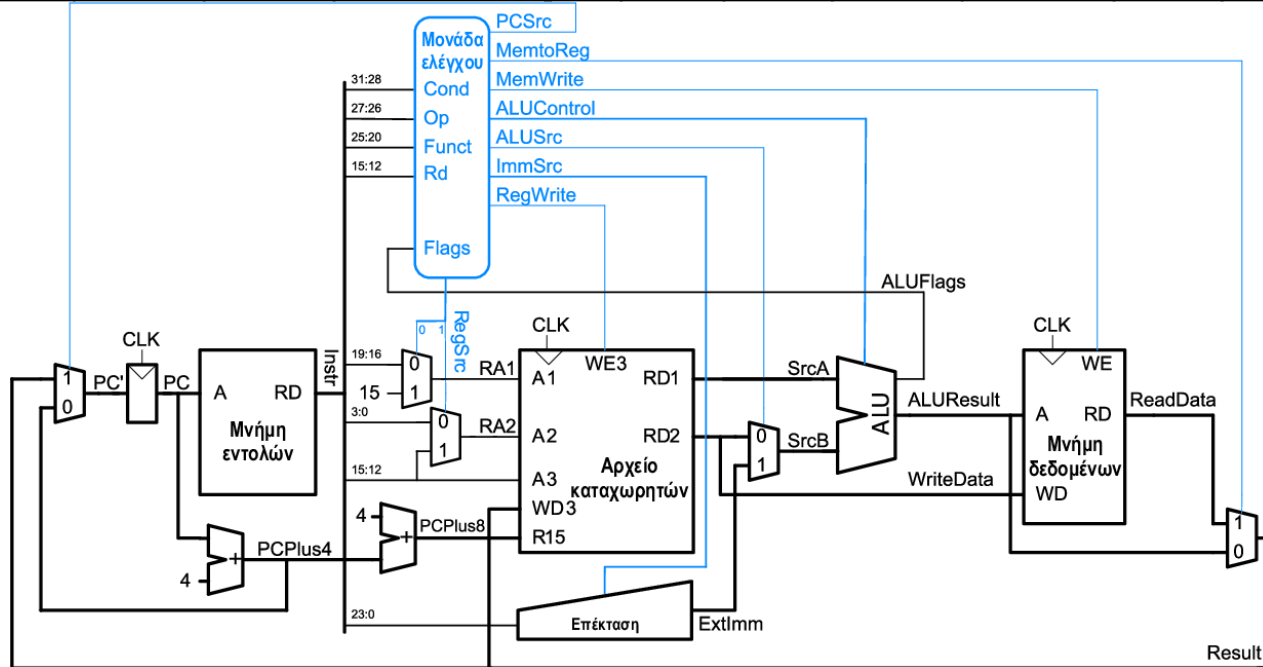
Παράδειγμα: `ANDS R7, R1, R3`

$\text{FlagW}_{1:0} = 10$ AND $\text{CondEx} = 1$ (χωρίς συνθήκη) $\Rightarrow \text{FlagWrite}_{1:0} = 10$



Παράδειγμα: ORR

Op	Funct ₅	Funct ₀	Τύπος	Branch	MemtoReg	MemW	ALUSrc	ImmSrc	RegW	RegSrc	ALUOp
00	0	X	DP Reg	0	0	0	0	XX	1	00	1



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

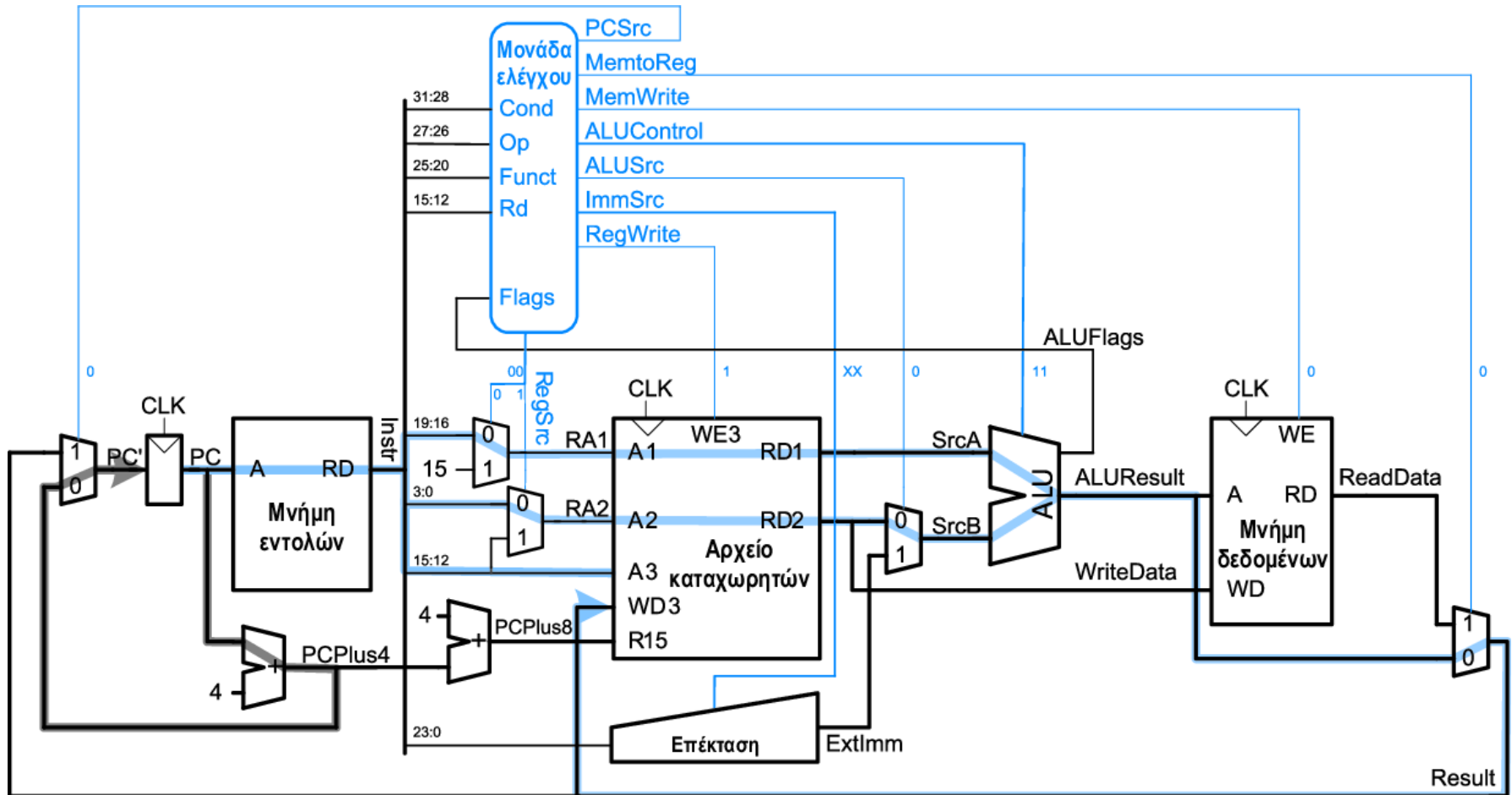
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

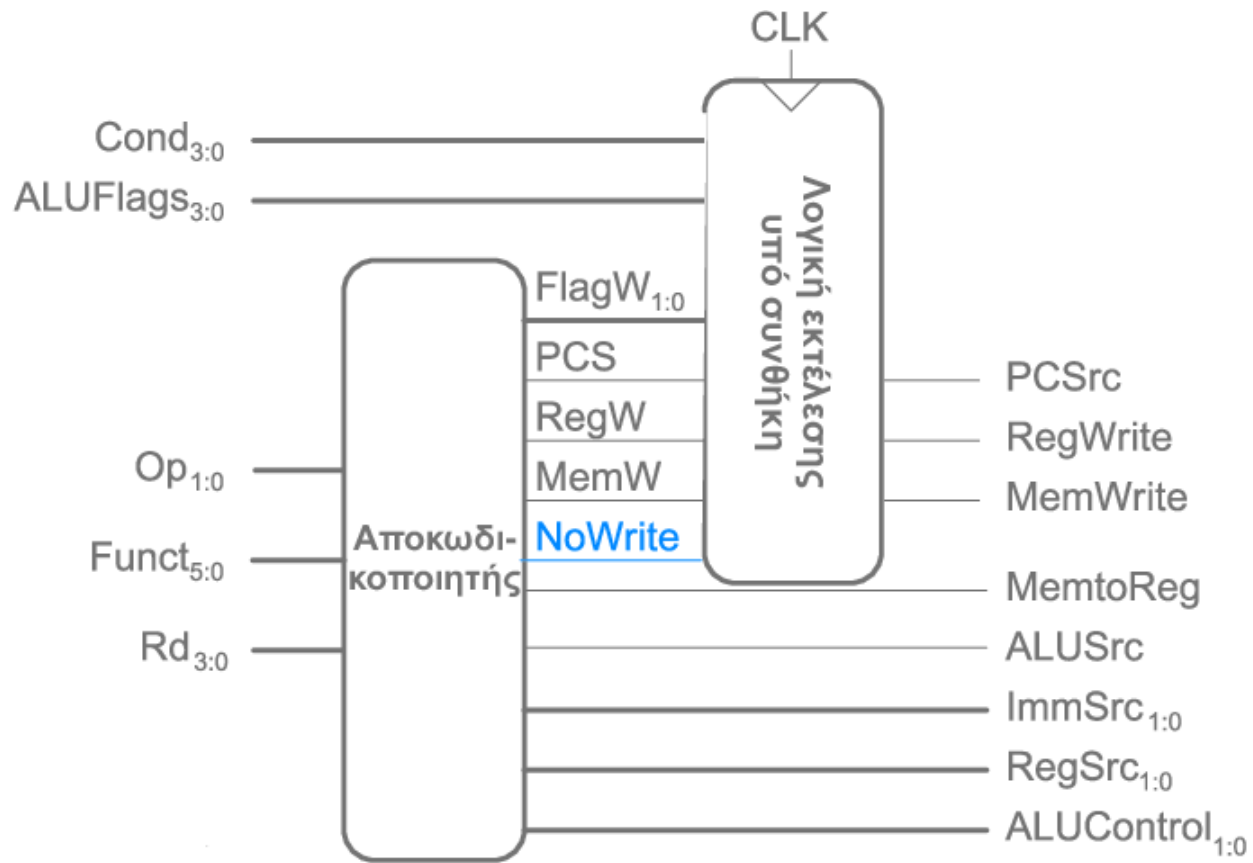
Κεφάλαιο 7 <64>



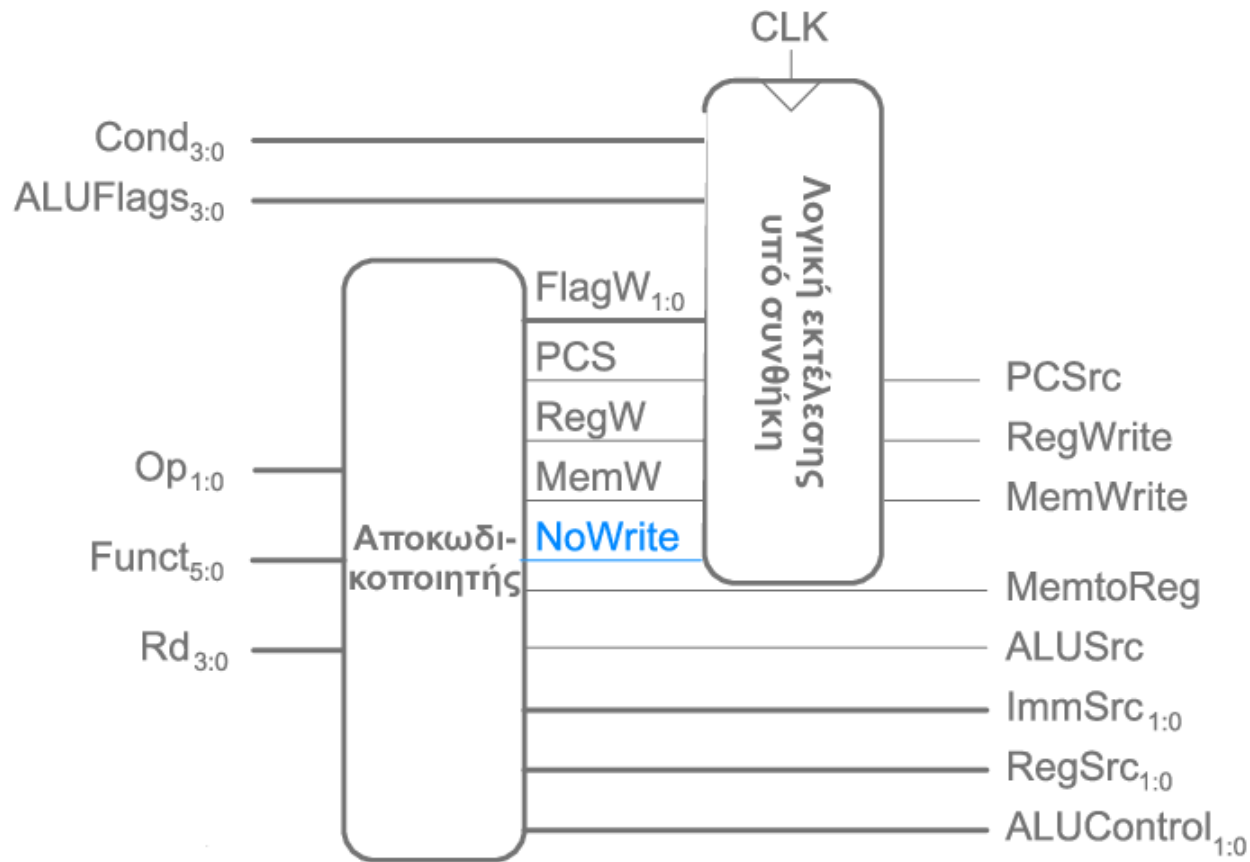
Παράδειγμα: ORR



Πρόσθετη λειτουργικότητα: Εντολή CMP



Πρόσθετη λειτουργικότητα: Εντολή CMP



Καμία αλλαγή στη διαδρομή δεδομένων

Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

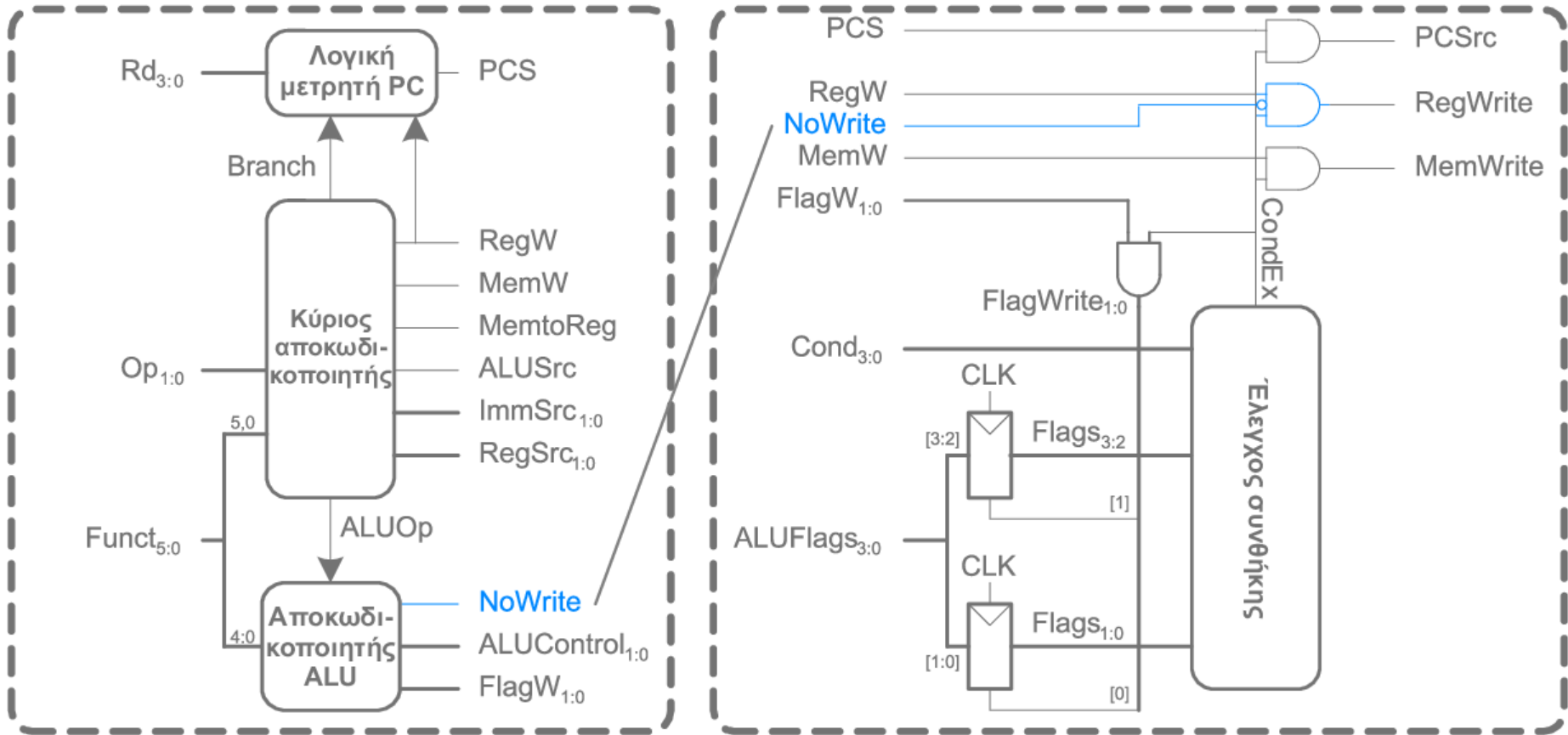
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτότυπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <67>



Πρόσθετη λειτουργικότητα: Εντολή CMP



Πρόσθετη λειτουργικότητα: Εντολή CMP

<i>ALUOp</i>	<i>Funct</i> _{4:1} (<i>cmd</i>)	<i>Funct</i> ₀ (<i>S</i>)	Τύπος	<i>ALUControl</i> _{1:0}	<i>FlagW</i> _{1:0}	<i>NoWrite</i>
0	X	X	Όχι DP	00	00	0
1	0100	0	ADD	00	00	0
		1			11	0
	0010	0	SUB	01	00	0
		1			11	0
	0000	0	AND	10	00	0
		1			10	0
	1100	0	ORR	11	00	0
		1			10	0
1010		1	CMP	01	11	1

Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

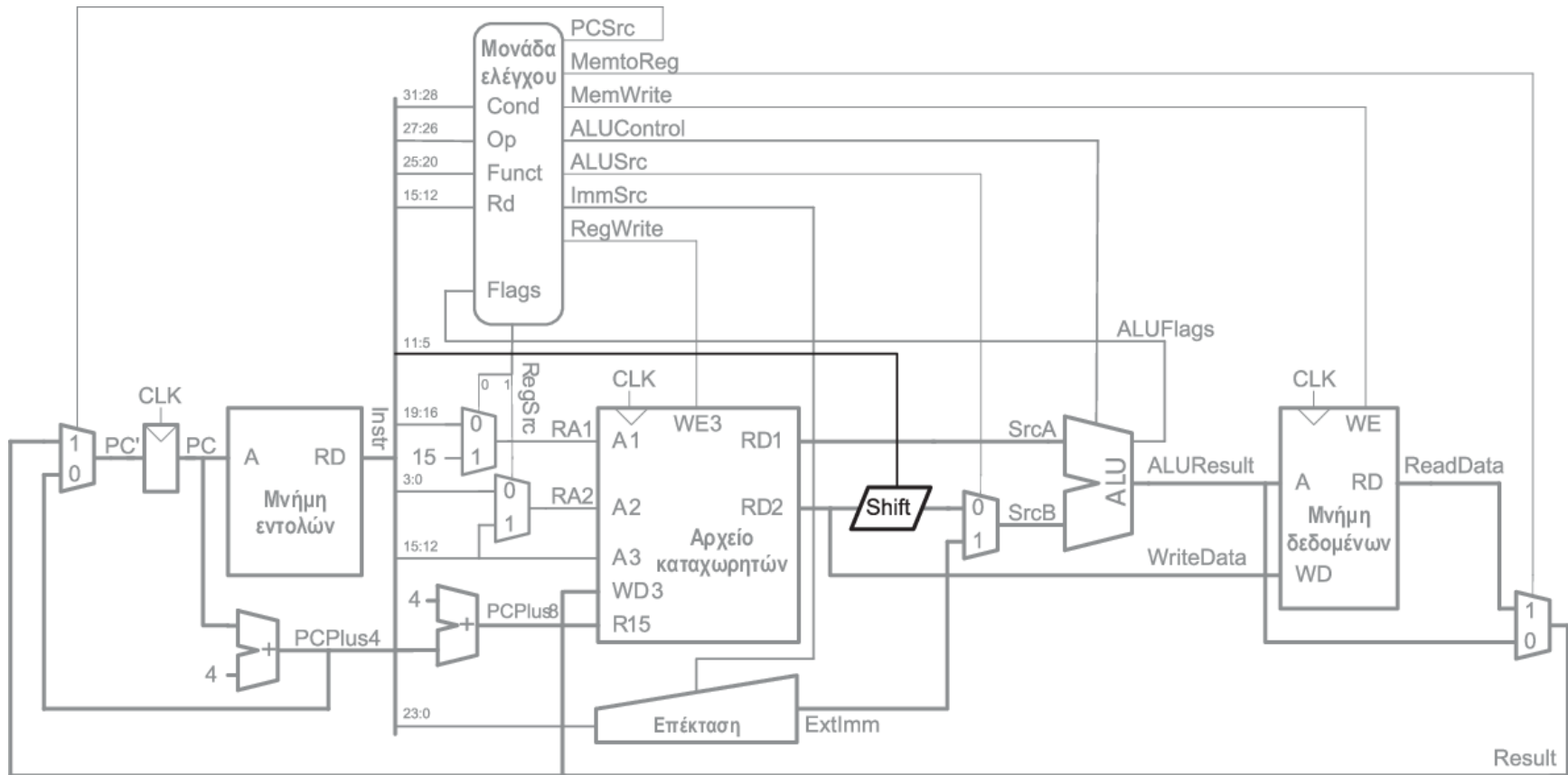
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτότυπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <69>



Πρόσθετη λειτουργικότητα: Ολισθαίνων καταχωρητής



	31:28	27:26	25	24:21	20	19:16	15:12	11:7	6:5	4	3:0
ADD R7, R2, R12, LSR #5	14	0	0	4	0	2	7	5	0	1	12
	cond	op	I	cmd	S	rn	rd	shamt5	sh		rm

Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

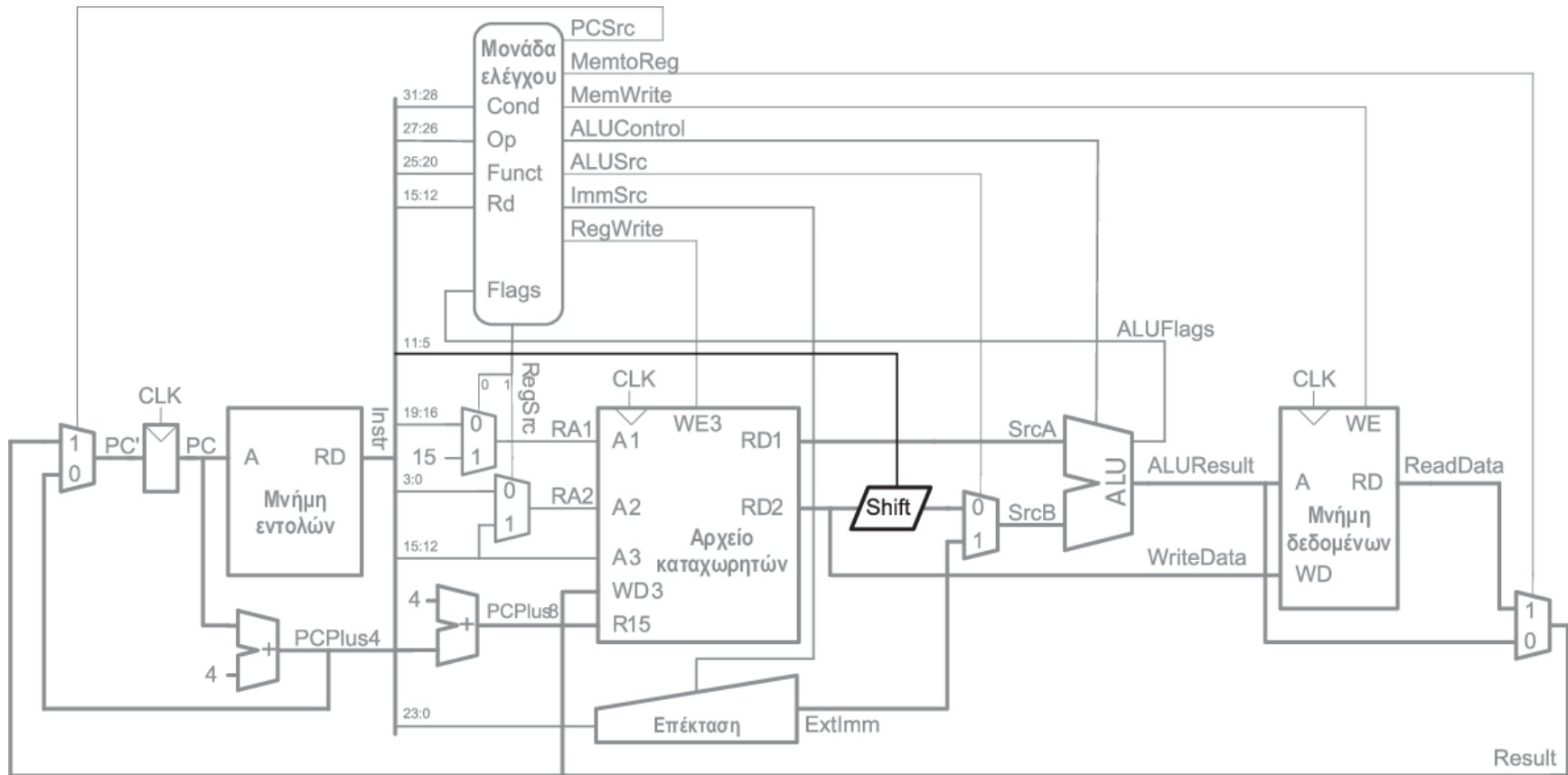
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <70>



Πρόσθετη λειτουργικότητα: Ολισθαίνων καταχωρητής



Καμία αλλαγή στον ελεγκτή

Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <71>



Επανάληψη: Επιδόσεις επεξεργαστή

Χρόνος εκτέλεσης προγραμμάτων

= (πλήθος εντολών)(κύκλοι/εντολή)(δευτερόλεπτα/κύκλο)

= πλήθος εντολών \times CPI \times T_C



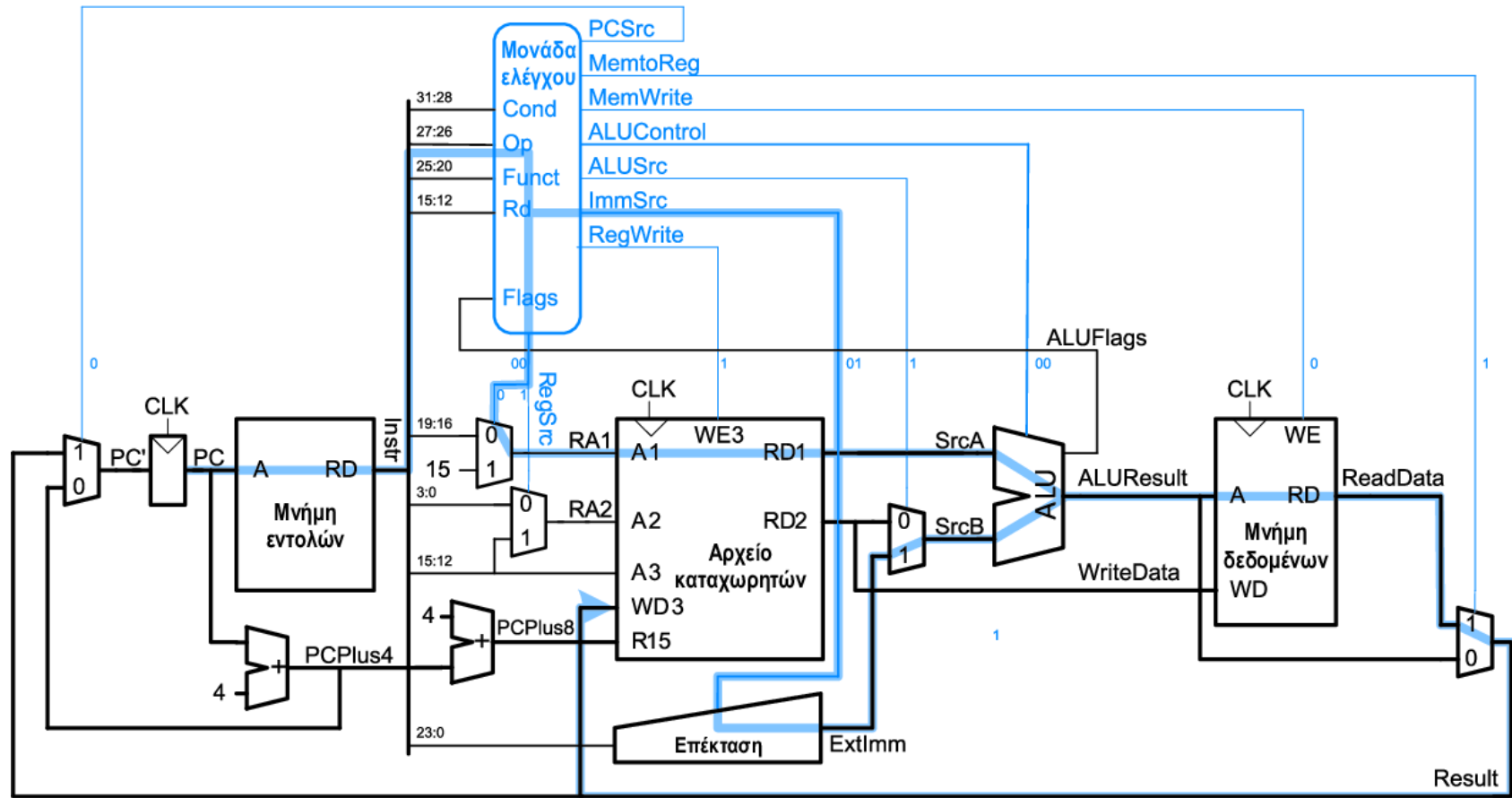
Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <72>

Επιδόσεις επεξεργαστή ενός κύκλου



Ο χρόνος T_C περιορίζεται από την κρίσιμη διαδρομή (LDR)

Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <73>



Επιδόσεις επεξεργαστή ενός κύκλου

- Κρίσιμη διαδρομή ενός κύκλου:

$$T_{cl} = t_{pcq_PC} + t_{mem} + t_{dec} + \max[t_{mux} + t_{RFread}, t_{sext} + t_{mux}] + t_{ALU} + t_{mem} + t_{mux} + t_{RFsetup}$$

- Συνήθως οι περιοριστικές διαδρομές είναι:

– Μνήμη, μονάδα ALU, αρχείο καταχωρητών

– $T_{cl} = t_{pcq_PC} + 2t_{mem} + t_{dec} + t_{RFread} + t_{ALU} + 2t_{mux} + t_{RFsetup}$



Επιδόσεις επεξεργαστή ενός κύκλου: Παράδειγμα

Στοιχείο	Παράμετρος	Καθυστέρηση (ps)
Καθυστέρηση από το ρολόι έως την έξοδο Q (καταχωρητής)	t_{pcq_PC}	40
Σταθεροποίηση καταχωρητή	t_{setup}	50
Πολυπλέκτης	t_{mux}	25
Μονάδα ALU	t_{ALU}	120
Αποκωδικοποιητής	t_{dec}	70
Ανάγνωση από τη μνήμη	t_{mem}	200
Ανάγνωση από το αρχείο καταχωρητών	t_{RFread}	100
Σταθεροποίηση αρχείου καταχωρητών	$t_{RFsetup}$	60

$$T_{cl} = ;$$



Επιδόσεις επεξεργαστή ενός κύκλου: Παράδειγμα

Στοιχείο	Παράμετρος	Καθυστέρηση (ps)
Καθυστέρηση από το ρολόι έως την έξοδο Q (καταχωρητής)	t_{pcq_PC}	40
Σταθεροποίηση καταχωρητή	t_{setup}	50
Πολυπλέκτης	t_{mux}	25
Μονάδα ALU	t_{ALU}	120
Αποκωδικοποιητής	t_{dec}	70
Ανάγνωση από τη μνήμη	t_{mem}	200
Ανάγνωση από το αρχείο καταχωρητών	t_{RFread}	100
Σταθεροποίηση αρχείου καταχωρητών	$t_{RFsetup}$	60

$$\begin{aligned}T_{cl} &= t_{pcq_PC} + 2t_{mem} + t_{dec} + t_{RFread} + t_{ALU} + 2t_{mux} + t_{RFsetup} \\ &= [50 + 2(200) + 70 + 100 + 120 + 2(25) + 60] \text{ ps} \\ &= \mathbf{840 \text{ ps}}\end{aligned}$$



Επιδόσεις επεξεργαστή ενός κύκλου: Παράδειγμα

Πρόγραμμα με 100 δισεκατομμύρια εντολές:

$$\begin{aligned}\text{Χρόνος εκτέλεσης} &= \text{πλήθος εντολών} \times \text{CPI} \times T_C \\ &= (100 \times 10^9)(1)(840 \times 10^{-12} \text{ s}) \\ &= \mathbf{84 \text{ δευτερόλεπτα}}\end{aligned}$$



Επεξεργαστής ARM πολλών κύκλων

- **Ενός κύκλου**

- + Απλός

- Ο χρόνος κύκλου περιορίζεται από τη μεγαλύτερη εντολή (LDR)
 - Διαχωρίζει τις μνήμες για εντολές και δεδομένα
 - 3 αθροιστές/μονάδες ALU

- **Ο επεξεργαστής πολλών κύκλων επιλύει αυτά τα ζητήματα χωρίζοντας την εντολή σε μικρότερα βήματα**

- Οι μικρότερες εντολές απαιτούν λιγότερα βήματα
 - Μπορεί να επαναχρησιμοποιείται υλικό
 - Ο χρόνος κύκλου είναι συντομότερος



Επεξεργαστής ARM πολλών κύκλων

- **Ενός κύκλου**

- + Απλός

- Ο χρόνος κύκλου περιορίζεται από τη μεγαλύτερη εντολή (LDR)

- Διαχωρίζει τις μνήμες για εντολές και δεδομένα

- 3 αθροιστές/μονάδες ALU

- **Πολλών κύκλων:**

- + Υψηλότερη ταχύτητα ρολογιού

- + Οι απλούστερες εντολές εκτελούνται πιο γρήγορα

- + Επαναχρησιμοποιείται ακριβό υλικό σε πολλούς κύκλους

- Η επιβάρυνση δημιουργίας ακολουθίας (sequencing overhead) «χρεώνεται» πολλές φορές



Επεξεργαστής ARM πολλών κύκλων

- **Ενός κύκλου**

- + Απλός

- Ο χρόνος κύκλου περιορίζεται από τη μεγαλύτερη εντολή (LDR)

- Διαχωρίζει τις μνήμες για εντολές και δεδομένα

- 3 αθροιστές/μονάδες ALU

- **Πολλών κύκλων:**

- + Υψηλότερη ταχύτητα ρολογιού

- + Οι απλούστερες εντολές εκτελούνται πιο γρήγορα

- + Επαναχρησιμοποιείται ακριβό υλικό σε πολλούς κύκλους

- Η επιβάρυνση δημιουργίας ακολουθίας (sequencing overhead) «χρεώνεται» πολλές φορές

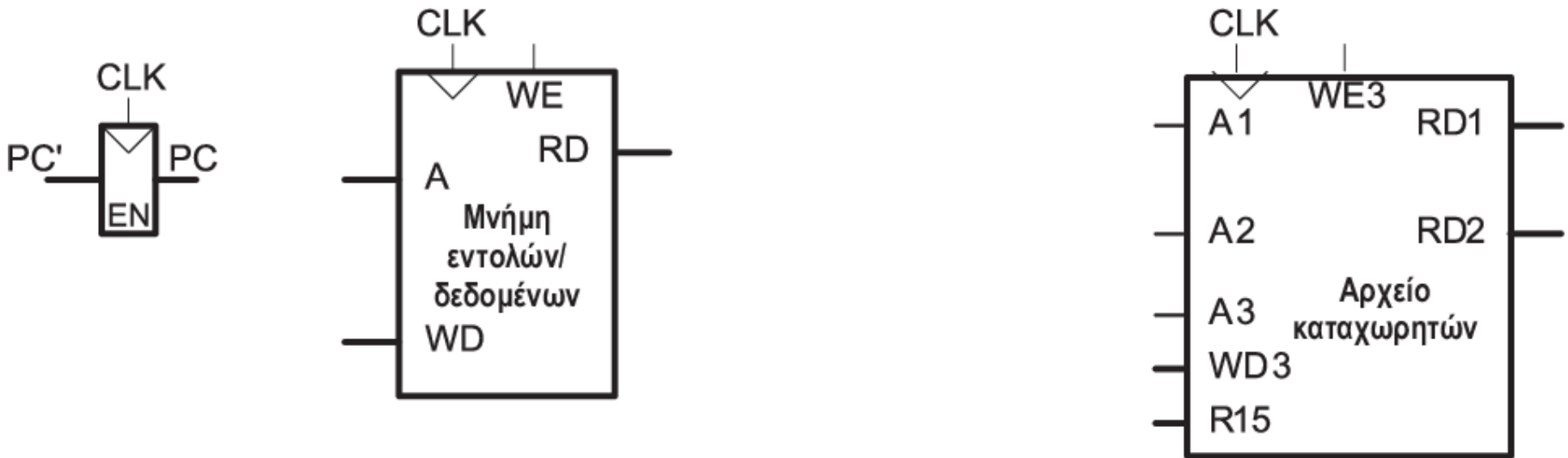
Ίδια σχεδιαστικά βήματα με τον επεξεργαστή ενός κύκλου:

- Πρώτα η διαδρομή δεδομένων
- Μετά η μονάδα ελέγχου



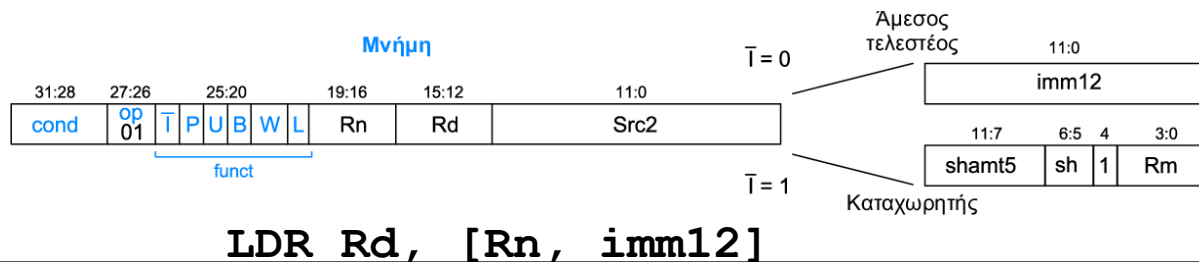
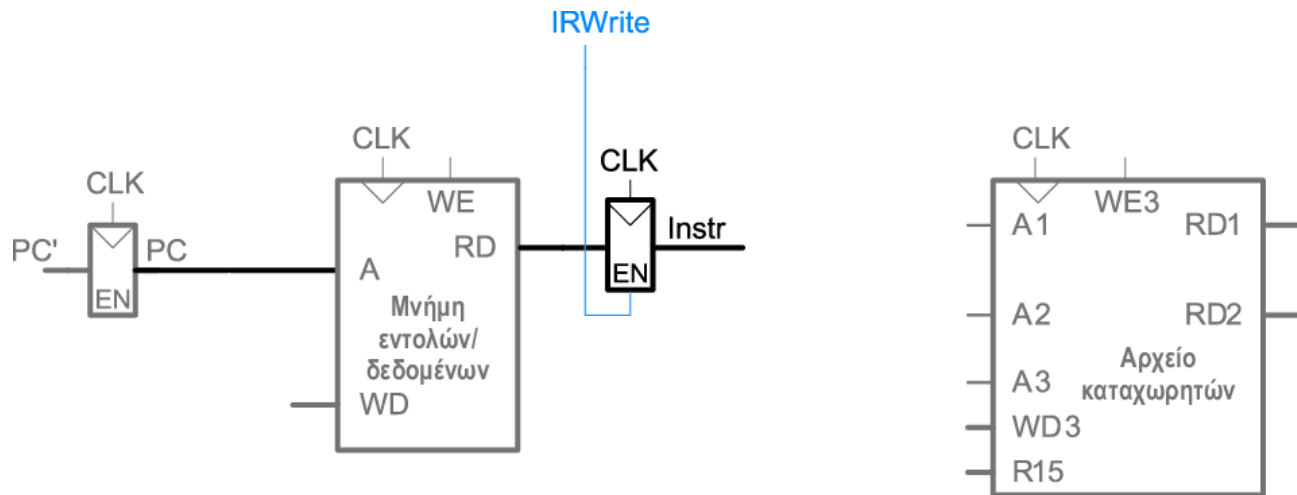
Επεξεργαστής πολλών κύκλων: Στοιχεία κατάστασης

Οι μνήμες εντολών και δεδομένων αντικαθίστανται με μία ενιαία μνήμη –πιο ρεαλιστικό



Διαδρομή δεδομένων πολλών κύκλων: Προσκόμιση εντολής

ΒΗΜΑ 1: Προσκόμιση εντολής



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

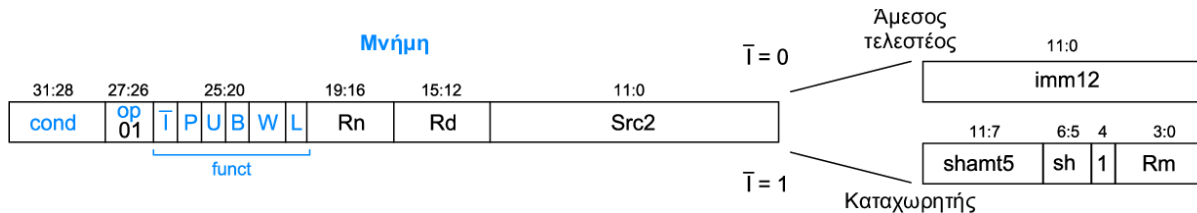
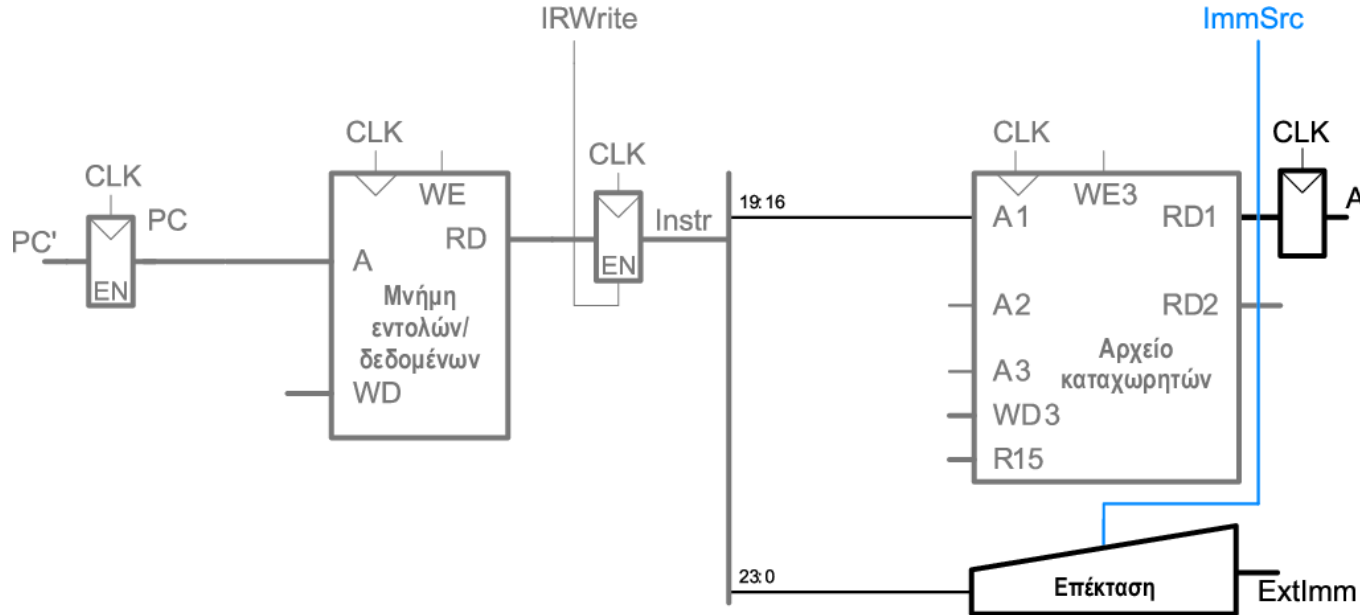
© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <82>



Διαδρομή δεδομένων πολλών κύκλων: Εντολή LDR – Ανάγνωση από αρχείο καταχωρητών

ΒΗΜΑ 2: Ανάγνωση τελεστών προέλευσης από το αρχείο καταχωρητών (RF)



LDR Rd, [Rn, imm12]

Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

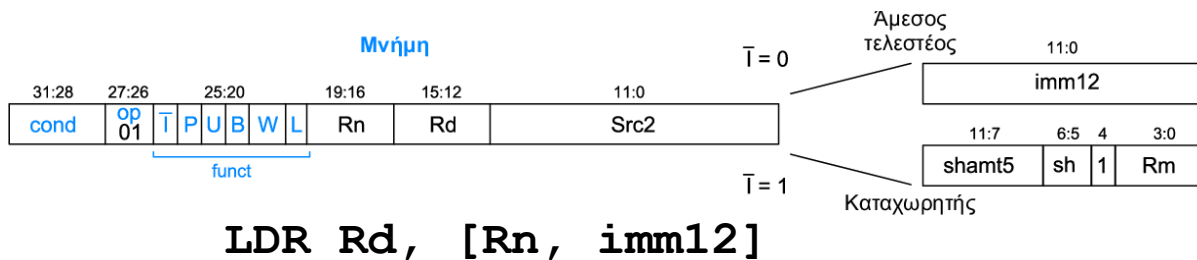
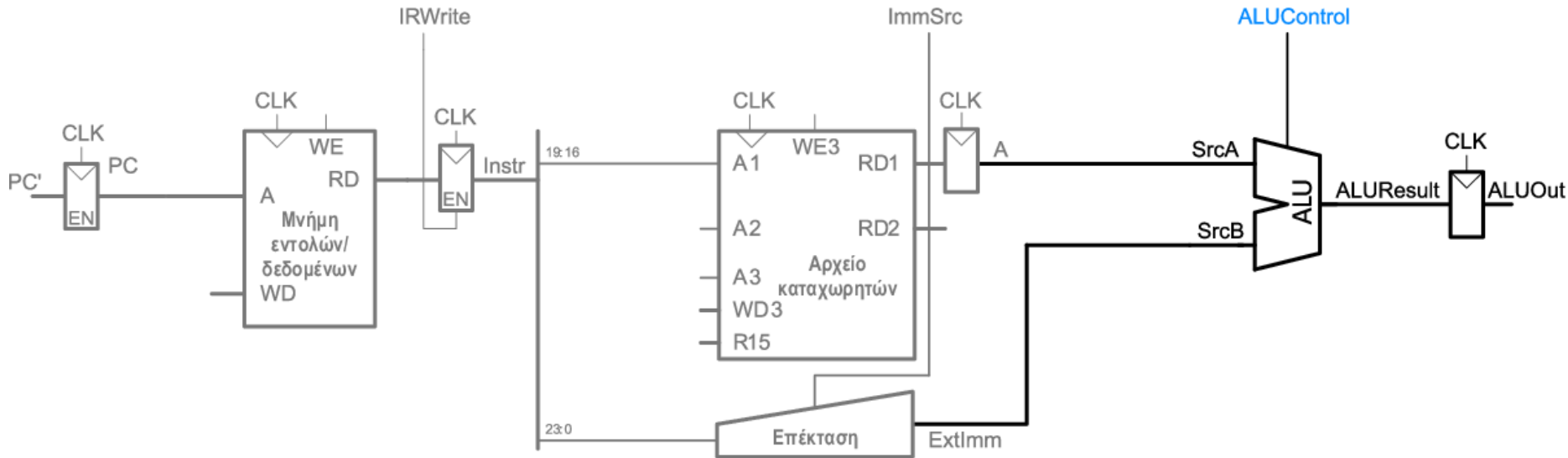
© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <83>



Διαδρομή δεδομένων πολλών κύκλων: Εντολή LDR – Διεύθυνση

ΒΗΜΑ 3: Υπολογισμός της διεύθυνσης μνήμης



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

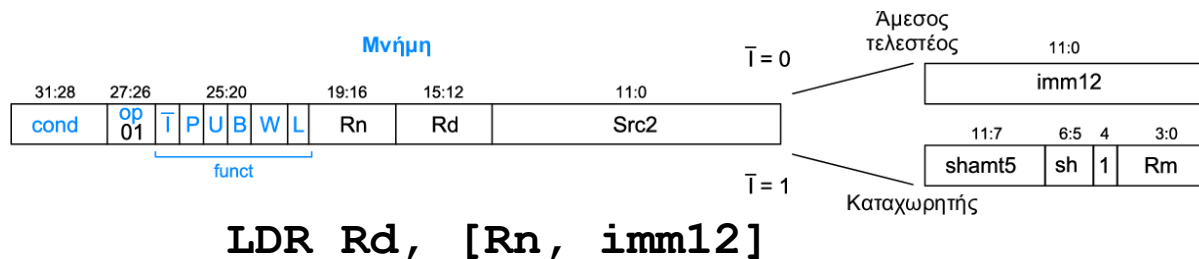
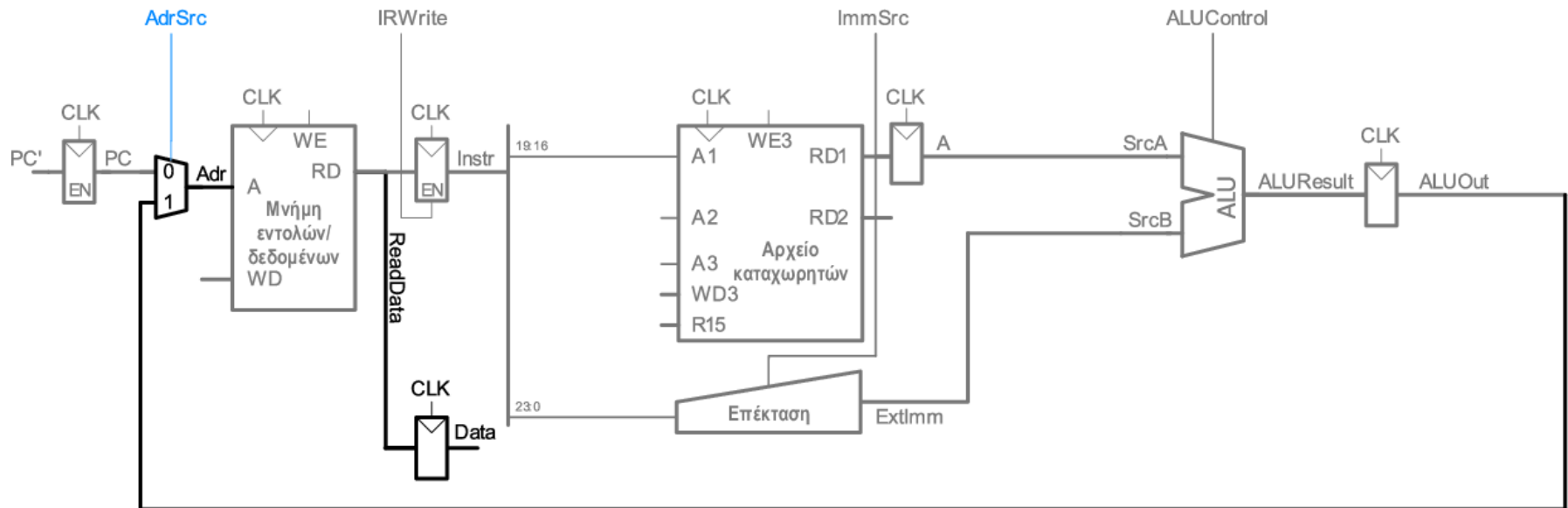
© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <84>



Διαδρομή δεδομένων πολλών κύκλων: Εντολή LDR – Ανάγνωση από τη μνήμη

ΒΗΜΑ 4: Ανάγνωση δεδομένων από τη μνήμη



LDR Rd, [Rn, imm12]

Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

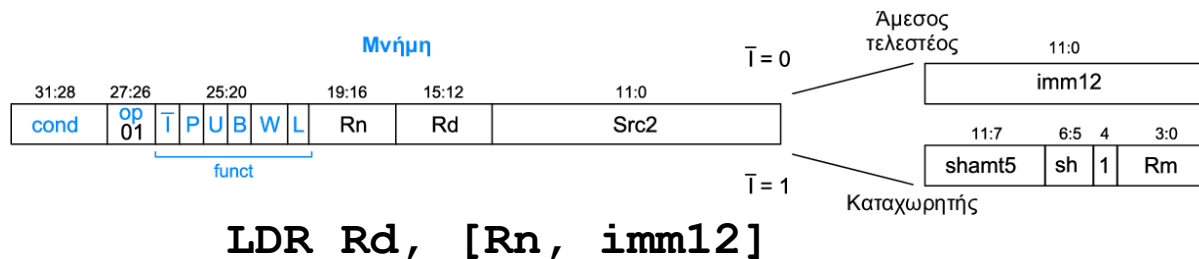
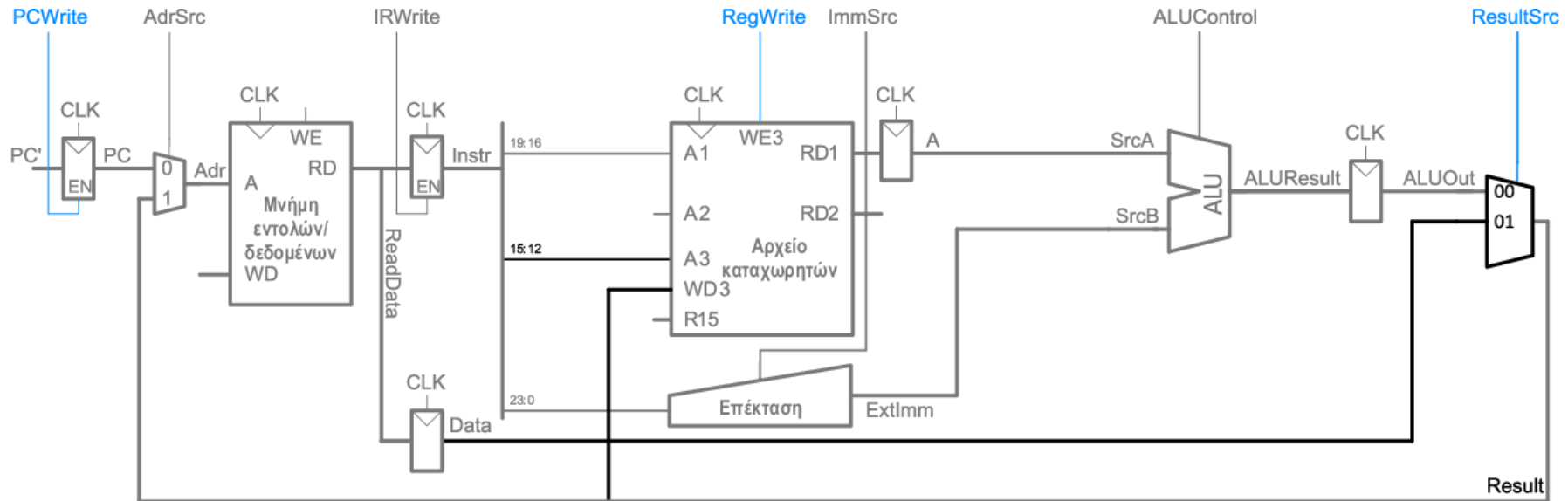
© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <85>



Διαδρομή δεδομένων πολλών κύκλων: Εντολή LDR – Εγγραφή στο αρχείο καταχωρητών

ΒΗΜΑ 5: Ετεροχρονισμένη εγγραφή δεδομένων στο αρχείο καταχωρητών



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

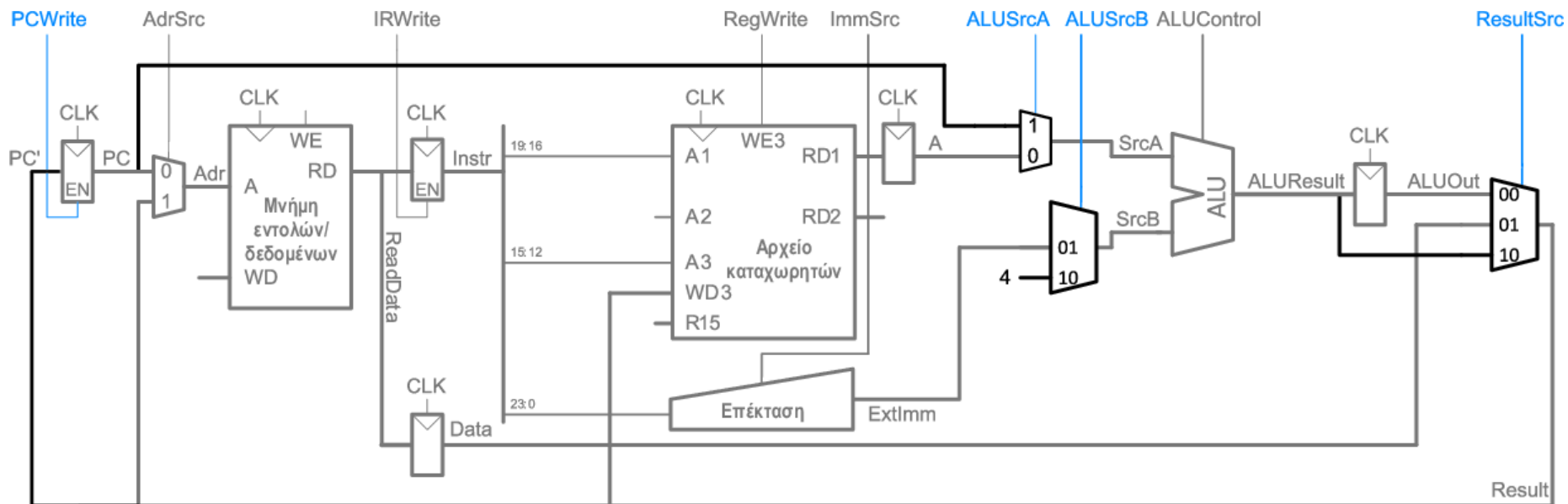
© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <86>



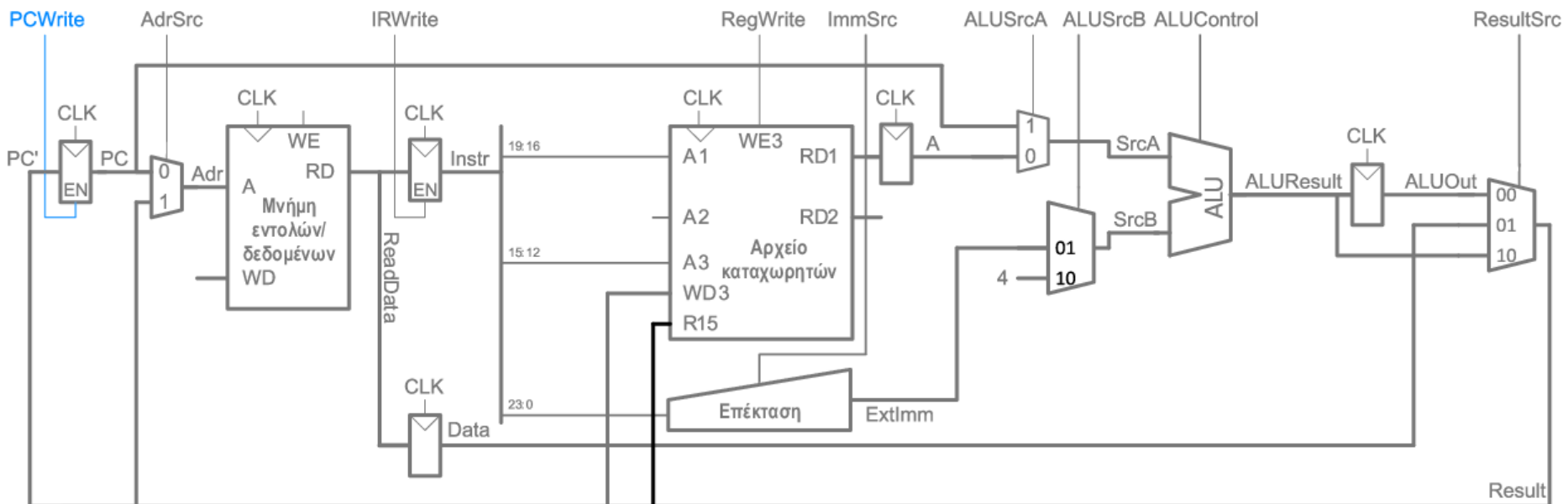
Διαδρομή δεδομένων πολλών κύκλων: Προσαύξηση του μετρητή PC

ΒΗΜΑ 6: Προσαύξηση του μετρητή PC



Διαδρομή δεδομένων πολλών κύκλων: Προσπέλαση του μετρητή PC

Ο μετρητής PC μπορεί να διαβάζεται/εγγράφεται από την εντολή



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτότυπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

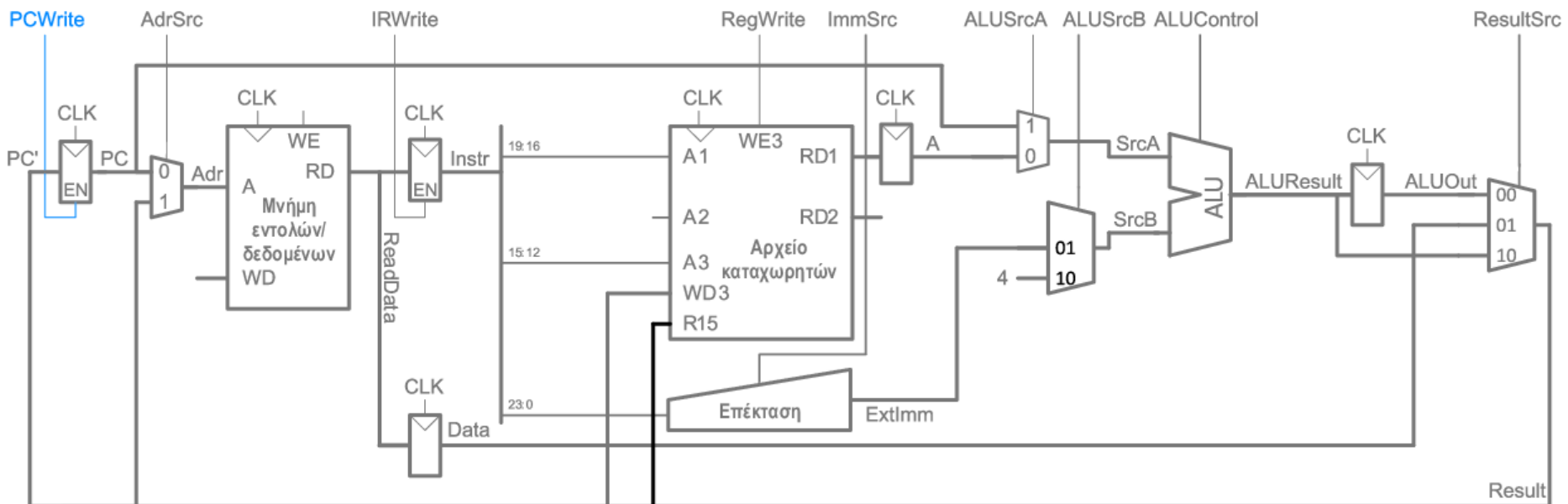
Κεφάλαιο 7 <88>



Διαδρομή δεδομένων πολλών κύκλων: Προσπέλαση του μετρητή PC

Ο μετρητής PC μπορεί να διαβάζεται/εγγράφεται από την εντολή

- **Ανάγνωση:** Ο R15 (PC+8) είναι διαθέσιμος στο αρχείο καταχωρητών



Διαδρομή δεδομένων πολλών κύκλων: Ανάγνωση από τον μετρητή PC (R15)

Παράδειγμα: `ADD R1, R15, R2`



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

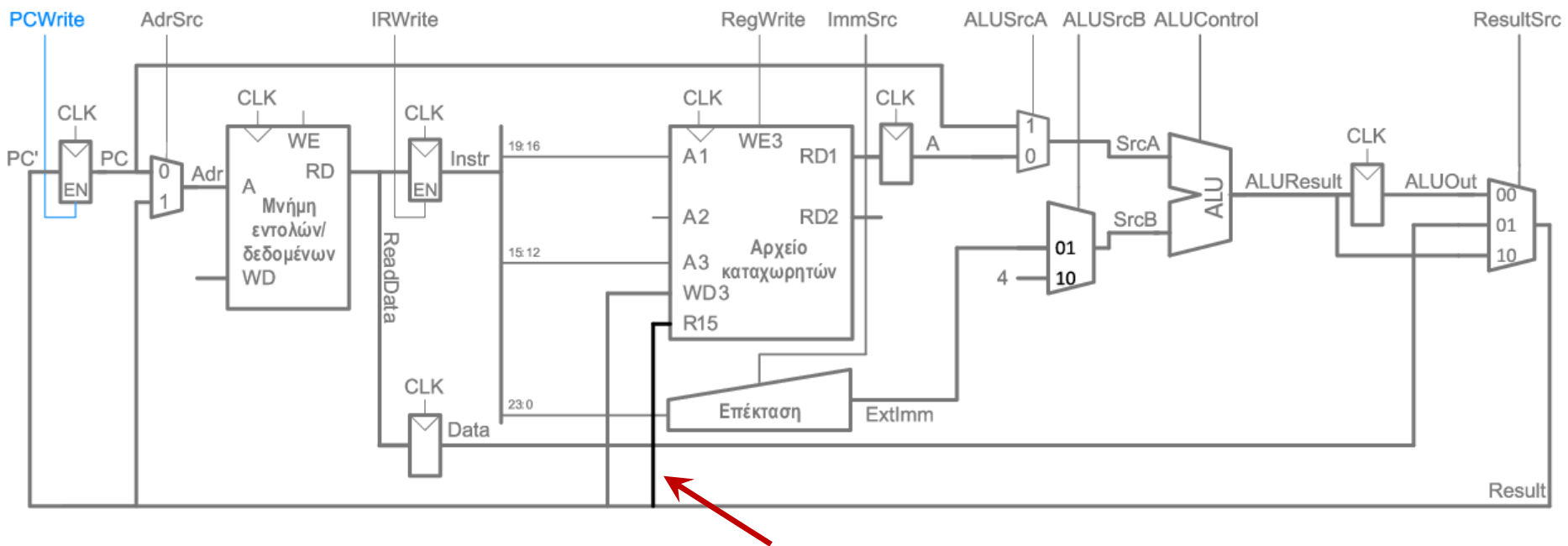
© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <90>

Διαδρομή δεδομένων πολλών κύκλων: Ανάγνωση από τον μετρητή PC (R15)

Παράδειγμα: ADD R1, R15, R2

- Ο R15 πρέπει να διαβαστεί ως PC+8 από το αρχείο καταχωρητών (RF) στο 2ο βήμα
- Άρα (επίσης στο 2ο βήμα), το PC + 8 παράγεται από τη μονάδα ALU και μεταβιβάζεται στην είσοδο R15 του RF



Διαδρομή δεδομένων πολλών κύκλων: Ανάγνωση από τον μετρητή PC (R15)

Παράδειγμα: `ADD R1, R15, R2`

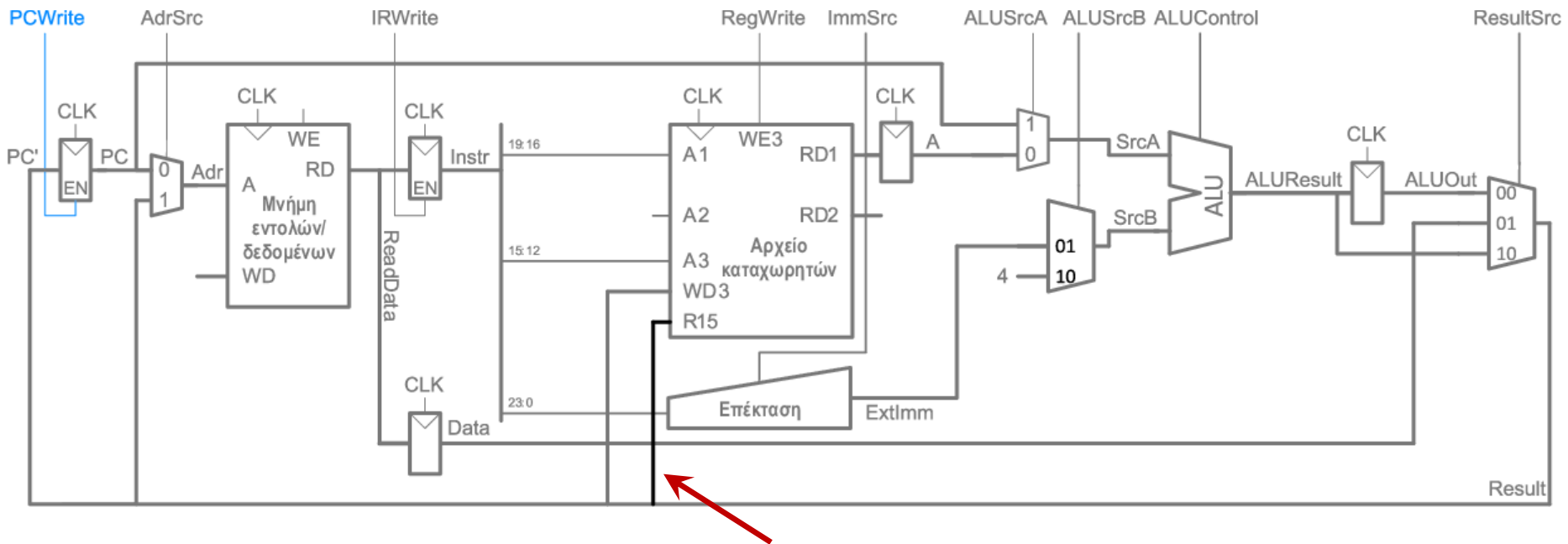
- Ο R15 πρέπει να διαβαστεί ως $PC+8$ από το αρχείο καταχωρητών (RF) στο 2ο βήμα
- Άρα (επίσης στο 2ο βήμα), το $PC + 8$ παράγεται από τη μονάδα ALU και μεταβιβάζεται στην είσοδο R15 του RF
 - $SrcA = PC$ (που, στο βήμα 1, έχει ήδη ενημερωθεί με την τιμή $PC + 4$)
 - $SrcB = 4$
 - $ALUResult = PC + 8$
- Το $ALUResult$ μεταβιβάζεται στη θύρα εισόδου R15 του RF στο 2ο βήμα (η οποία κατόπιν μεταβιβάζεται στην έξοδο RD1 του RF)



Διαδρομή δεδομένων πολλών κύκλων: Ανάγνωση από τον μετρητή PC (R15)

Παράδειγμα: ADD R1, R15, R2

- Ο R15 πρέπει να διαβαστεί ως PC+8 από το αρχείο καταχωρητών (RF) στο 2ο βήμα
- Άρα (επίσης στο 2ο βήμα), το PC + 8 παράγεται από τη μονάδα ALU και μεταβιβάζεται στην είσοδο R15 του RF



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτότυπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

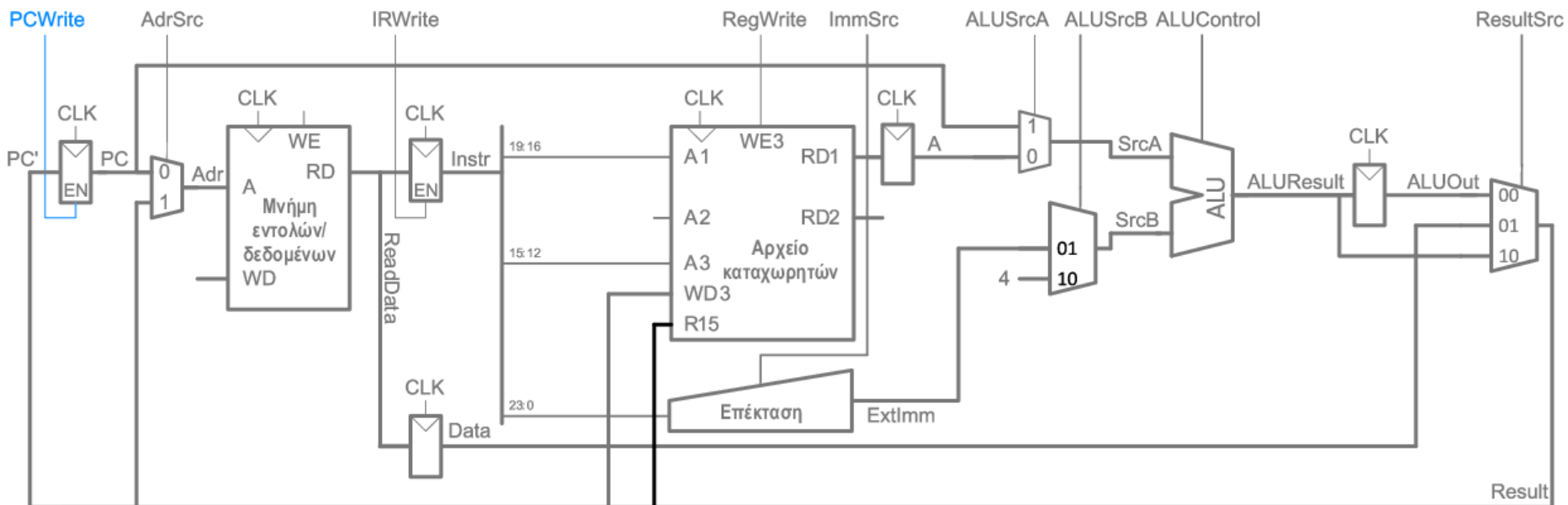
Κεφάλαιο 7 <93>



Διαδρομή δεδομένων πολλών κύκλων: Προσπέλαση του μετρητή PC

Ο μετρητής PC μπορεί να διαβάζεται/εγγράφεται από την εντολή

- **Ανάγνωση:** Ο R15 (PC+8) είναι διαθέσιμος στο αρχείο καταχωρητών
- **Εγγραφή:** Πρέπει να είναι εφικτή η εγγραφή του αποτελέσματος στον μετρητή PC



Διαδρομή δεδομένων πολλών κύκλων: Εγγραφή στον μετρητή PC (R15)

Παράδειγμα: SUB R15, R8, R3



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <95>

Διαδρομή δεδομένων πολλών κύκλων: Εγγραφή στον μετρητή PC (R15)

Παράδειγμα: SUB R15, R8, R3

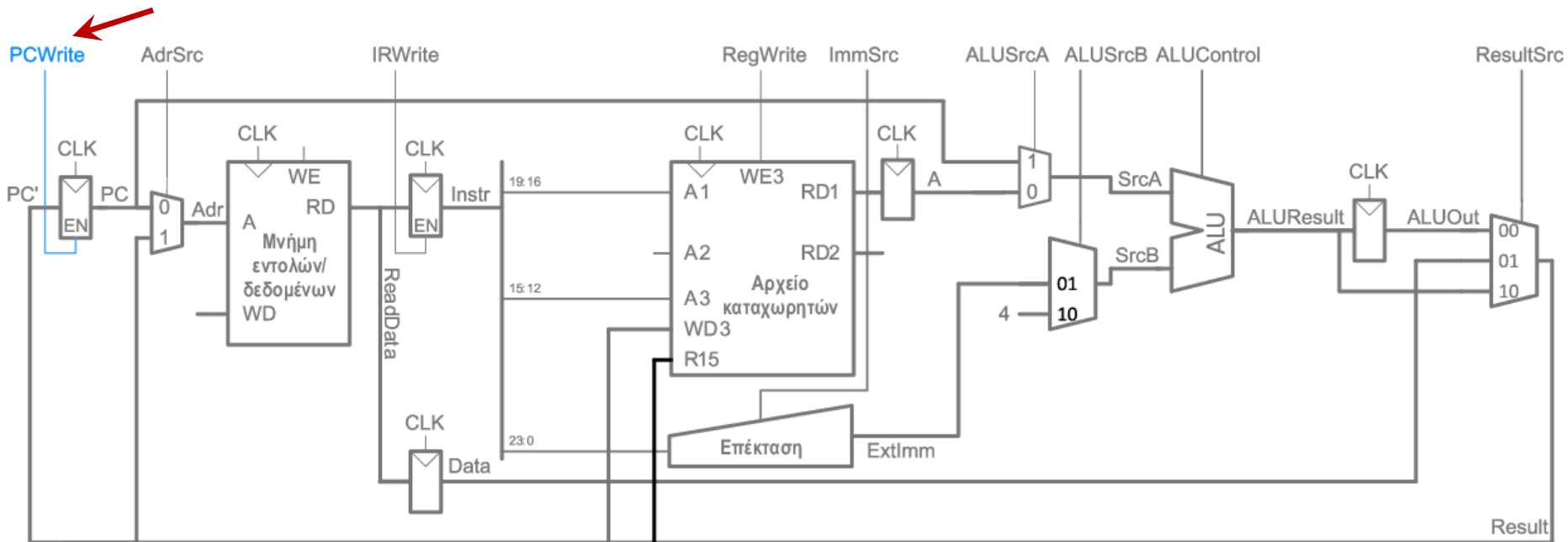
- Το αποτέλεσμα της εντολής πρέπει να εγγραφεί στον καταχωρητή PC
- Το *ALUResult* έχει ήδη μεταβιβαστεί στον καταχωρητή PC, απλώς πρέπει να ενεργοποιηθεί το *PCWrite*



Διαδρομή δεδομένων πολλών κύκλων: Εγγραφή στον μετρητή PC (R15)

Παράδειγμα : SUB R15, R8, R3

- Το αποτέλεσμα της εντολής πρέπει να εγγραφεί στον καταχωρητή PC
- Το *ALUResult* έχει ήδη μεταβιβαστεί στον καταχωρητή PC, απλώς πρέπει να ενεργοποιηθεί το *PCWrite*



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

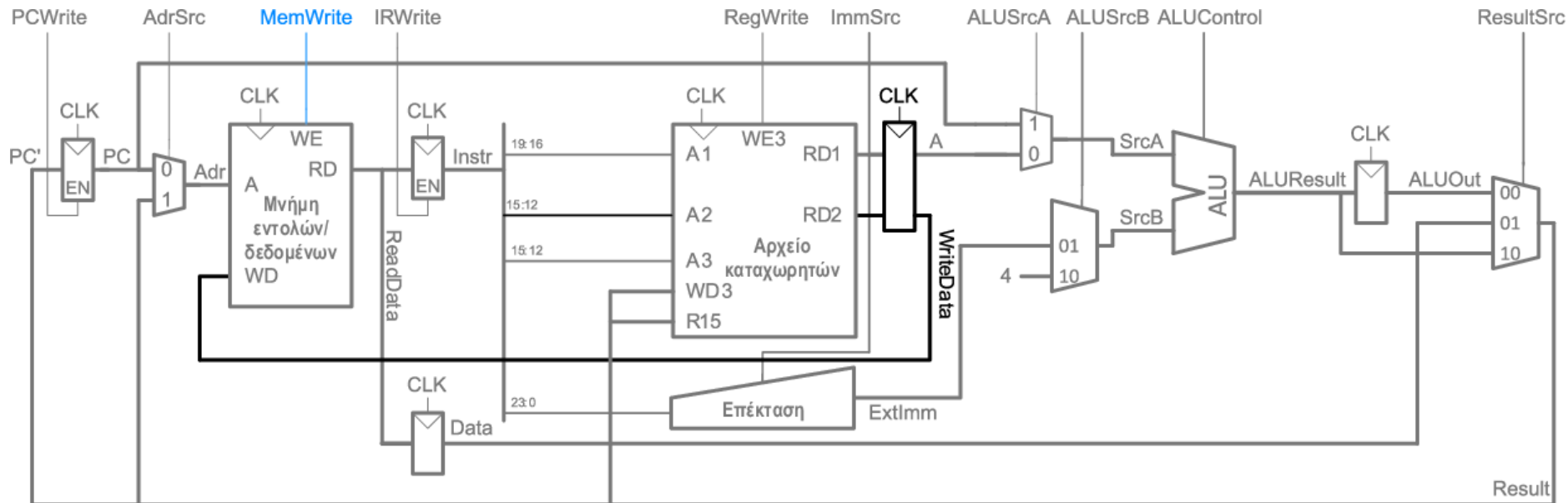
© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <97>



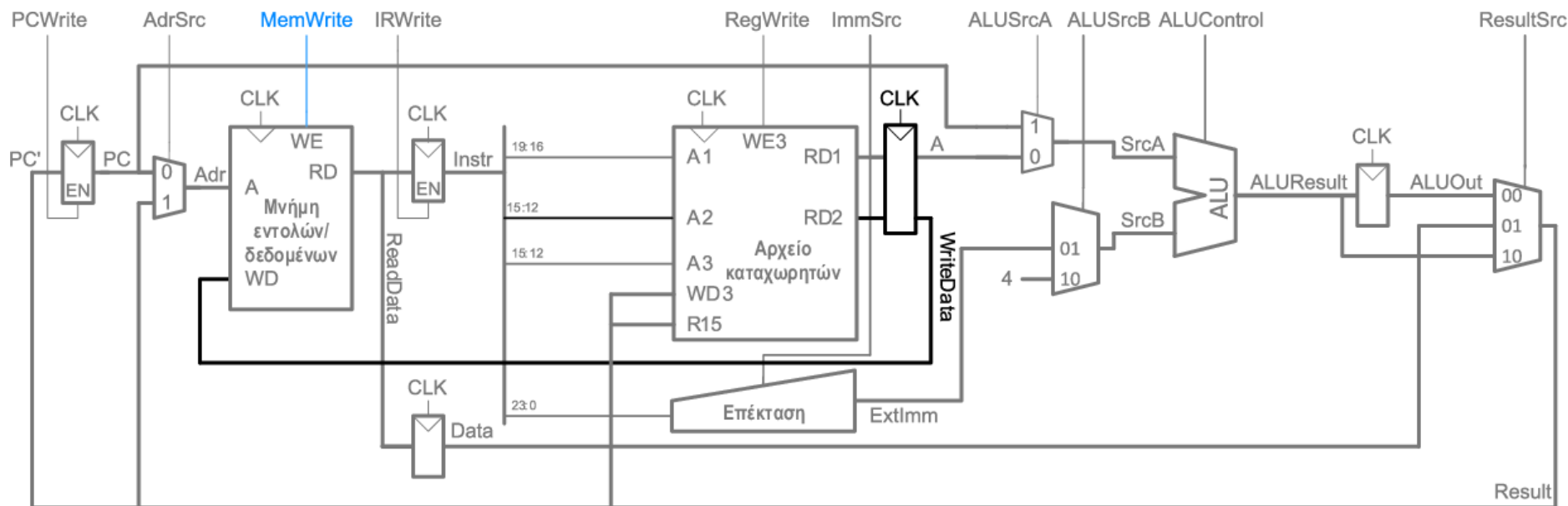
Διαδρομή δεδομένων πολλών κύκλων: Εντολή STR

Εγγραφή των δεδομένων του Rn στη μνήμη



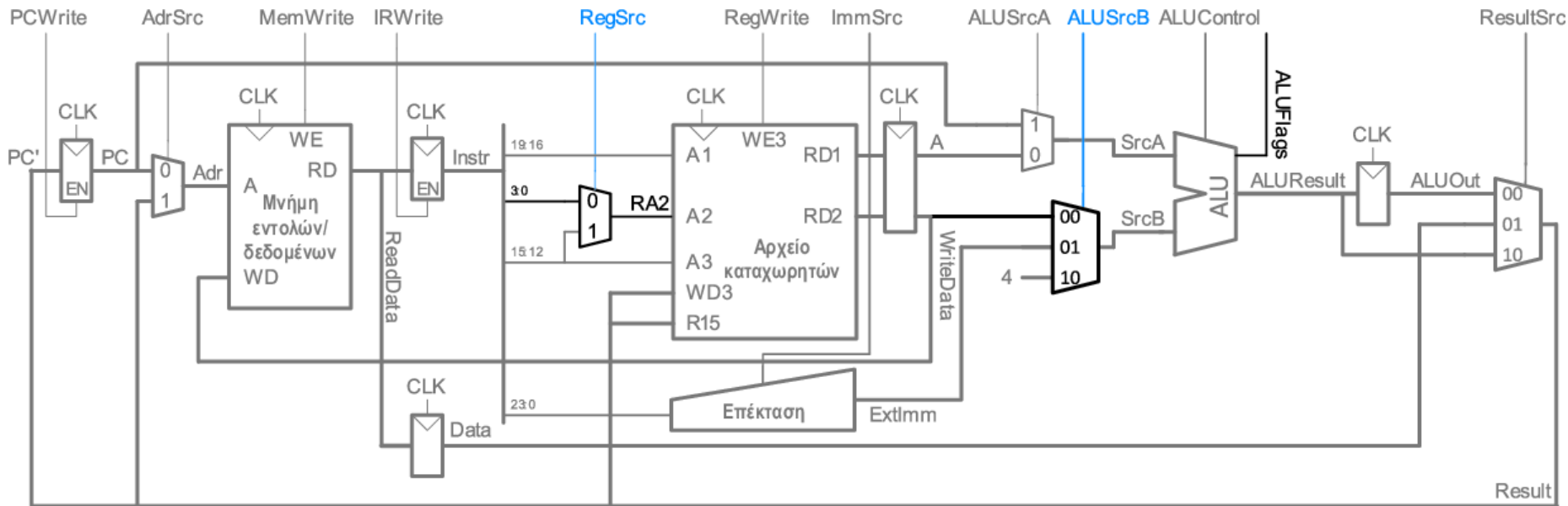
Διαδρομή δεδομένων πολλών κύκλων: Επεξεργασία δεδομένων

Με άμεση διευθυνσιοδότηση (δηλαδή έναν άμεσο τελεστέο ως το *Src2*), δεν απαιτούνται πρόσθετες αλλαγές για τη διαδρομή δεδομένων



Διαδρομή δεδομένων πολλών κύκλων: Επεξεργασία δεδομένων

Με διευθυνσιοδότηση βάσει καταχωρητή (καταχωρητής ως το *Src2*): Ανάγνωση από τα *Rn* και *Rm*



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <100>

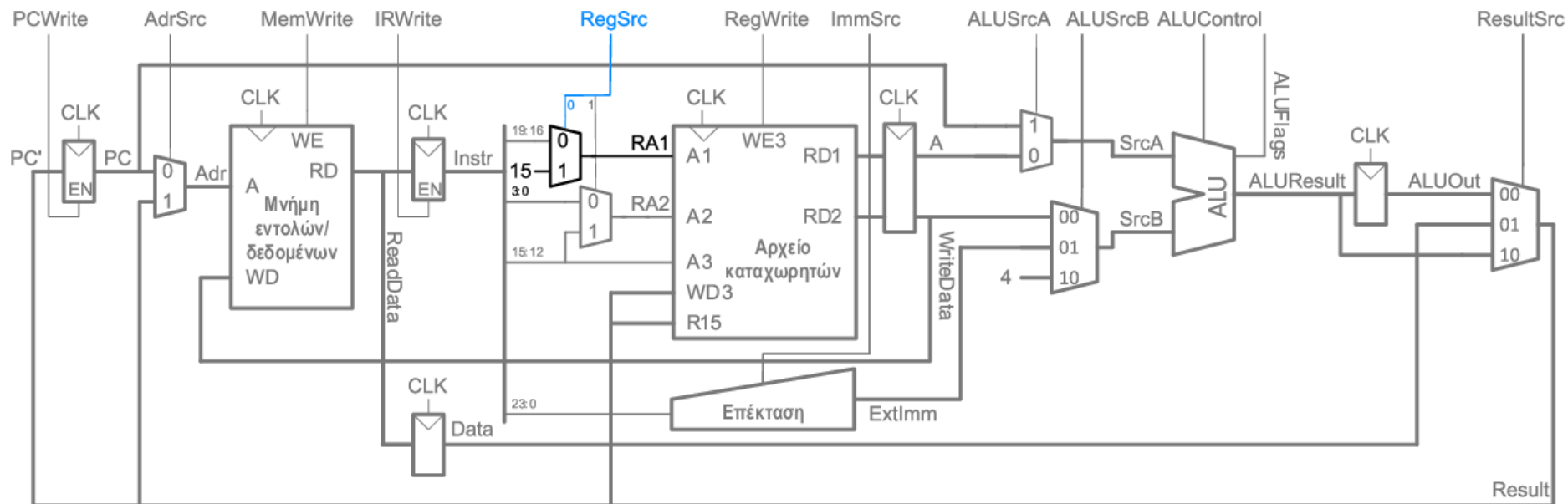


Διαδρομή δεδομένων πολλών κύκλων: Εντολή B

Υπολογισμός της διεύθυνσης-στόχου της διακλάδωσης:

$$BTA = (ExtImm) + (PC + 8)$$

$ExtImm = Imm24 \ll 2$ και επέκταση προσήμου



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

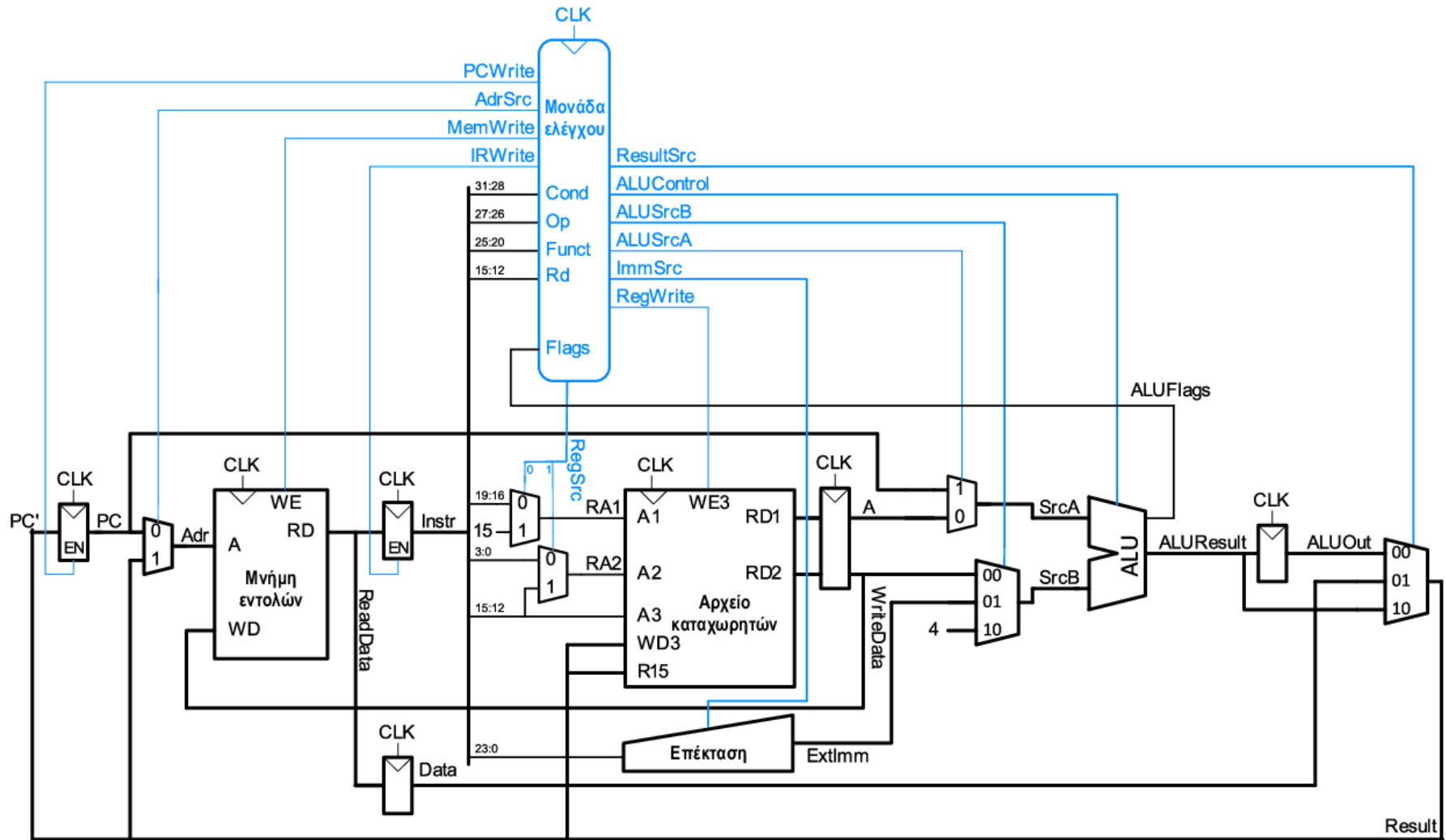
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτότυπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <101>



Επεξεργαστής ARM πολλών κύκλων



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

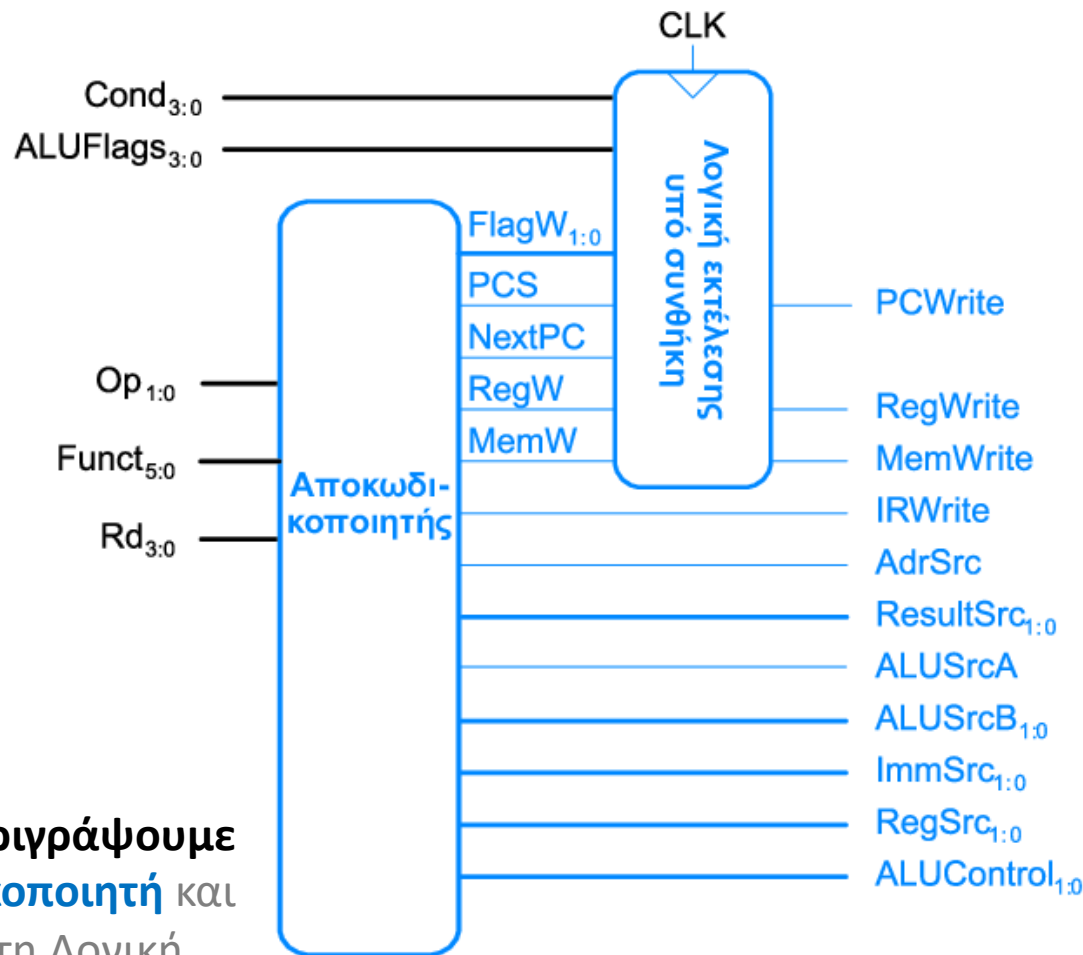
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτότυπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <102>



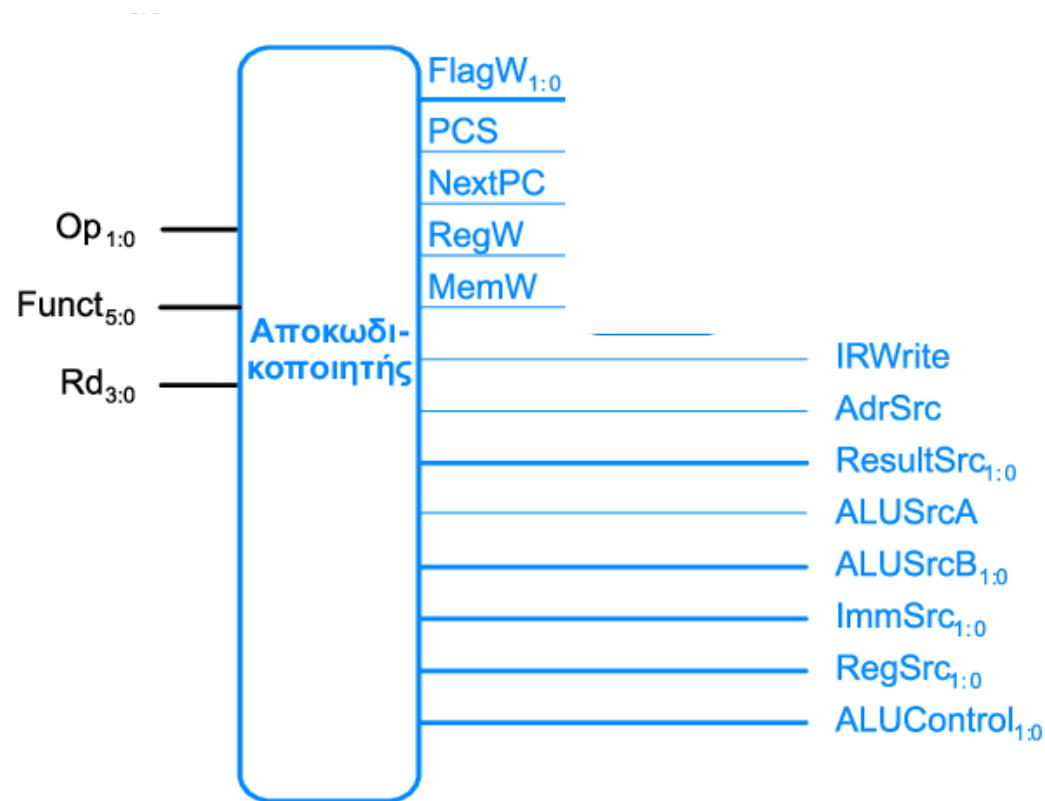
Μονάδα ελέγχου πολλών κύκλων



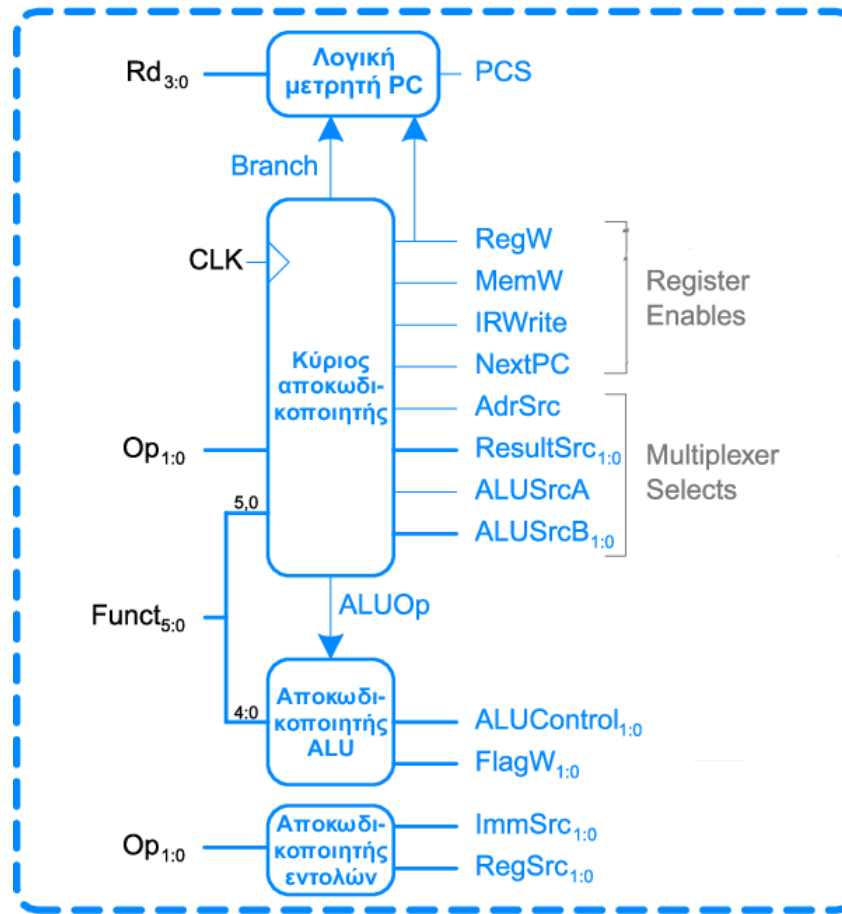
- Πρώτα θα περιγράψουμε τον **Αποκωδικοποιητή** και
- στη συνέχεια τη Λογική εκτέλεσης υπό συνθήκη



Μονάδα ελέγχου πολλών κύκλων: Αποκωδικοποιητής



Μονάδα ελέγχου πολλών κύκλων: Αποκωδικοποιητής



Αποκωδικοποιητής

Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

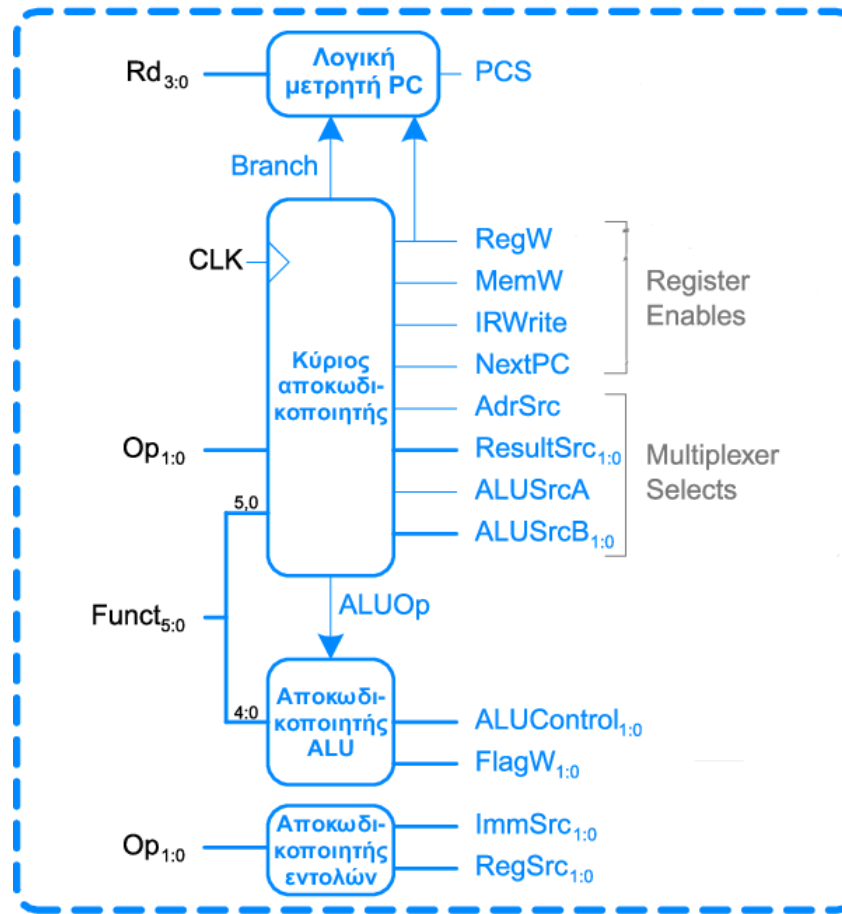
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <105>



Μονάδα ελέγχου πολλών κύκλων: Αποκωδικοποιητής



Ο αποκωδικοποιητής ALU και η λογική του μετρητή PC είναι ίδια όπως στη μονάδα ελέγχου ενός κύκλου

Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <106>



Μονάδα ελέγχου πολλών κύκλων: Αποκωδικοποιητής εντολών



$$RegSrc_0 = (Op == 10_2)$$

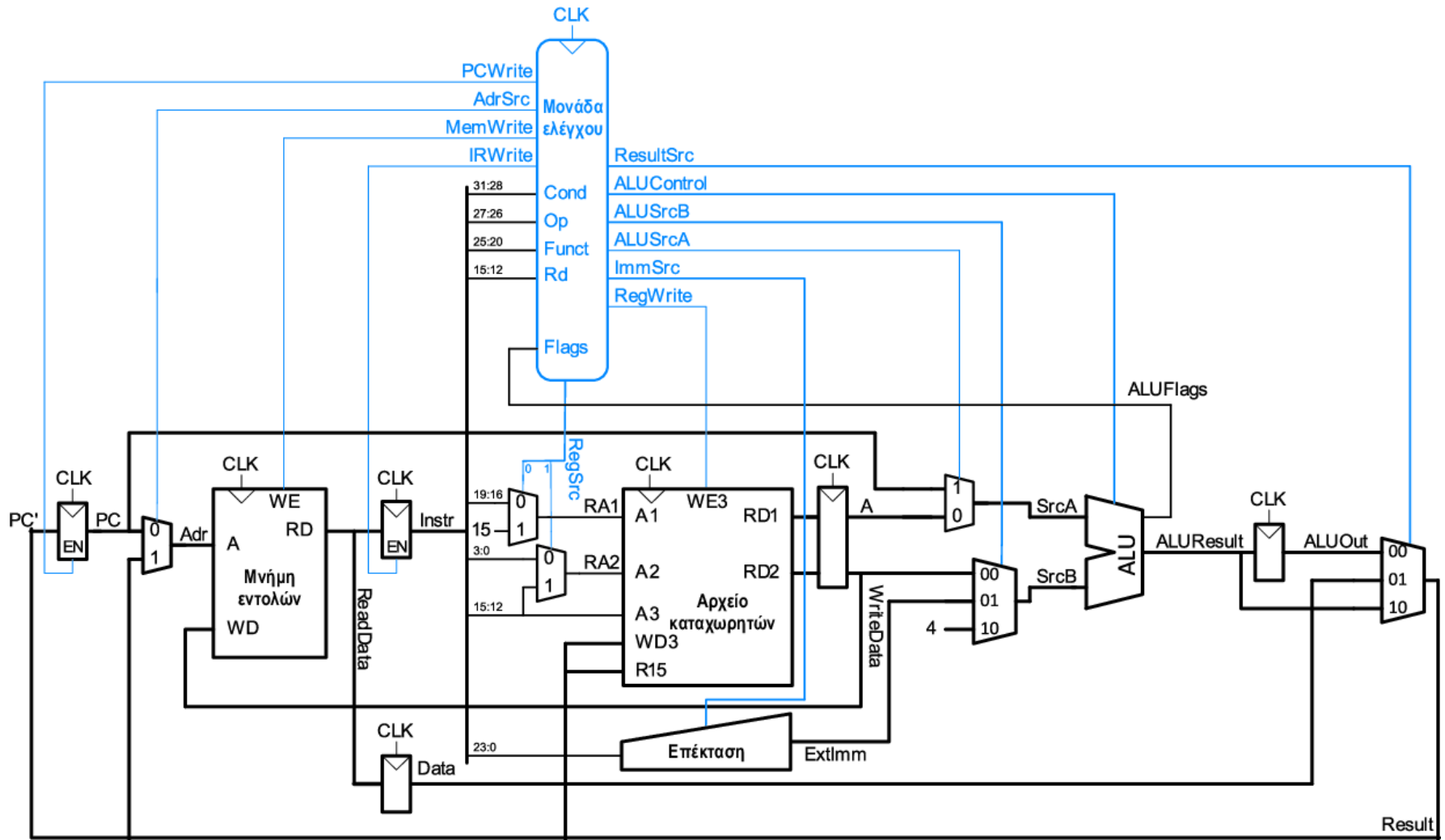
$$RegSrc_1 = (Op == 01_2)$$

$$ImmSrc_{1:0} = Op$$

Εντολή	Op	$Funct_5$	$Funct_0$	$RegSrc_0$	$RegSrc_1$	$ImmSrc_{1:0}$
LDR	01	X	1	0	X	01
STR	01	X	0	0	1	01
DP άμεσου τελεστέου	00	1	X	0	X	00
DP καταχωρητή	00	0	X	0	0	00
B	10	X	X	1	X	10



Επεξεργαστής ARM πολλών κύκλων



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

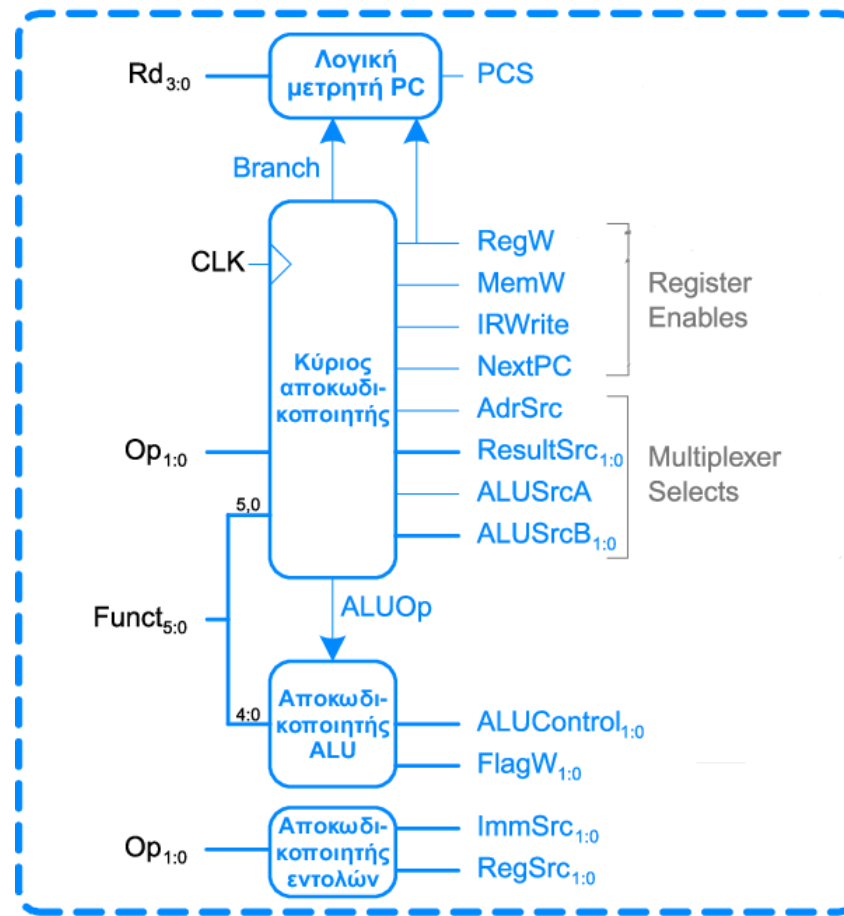
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <108>



Μονάδα ελέγχου πολλών κύκλων: Κύριος αποκωδικοποιητής (μηχανή FSM)



Αποκωδικοποιητής

Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

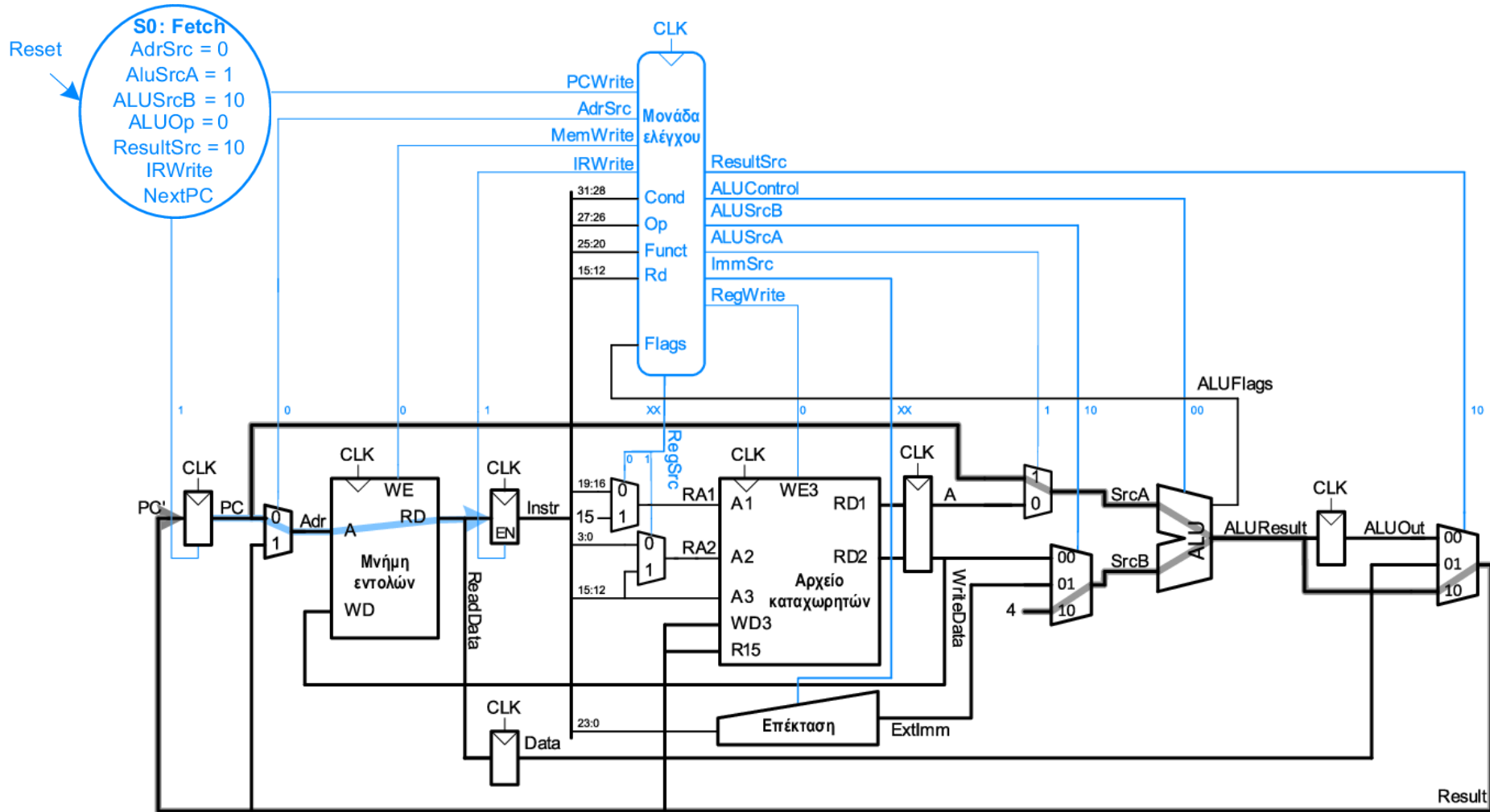
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <109>



Μηχανή FSM κύριου αποκωδικοποιητή: Προσκόμιση



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

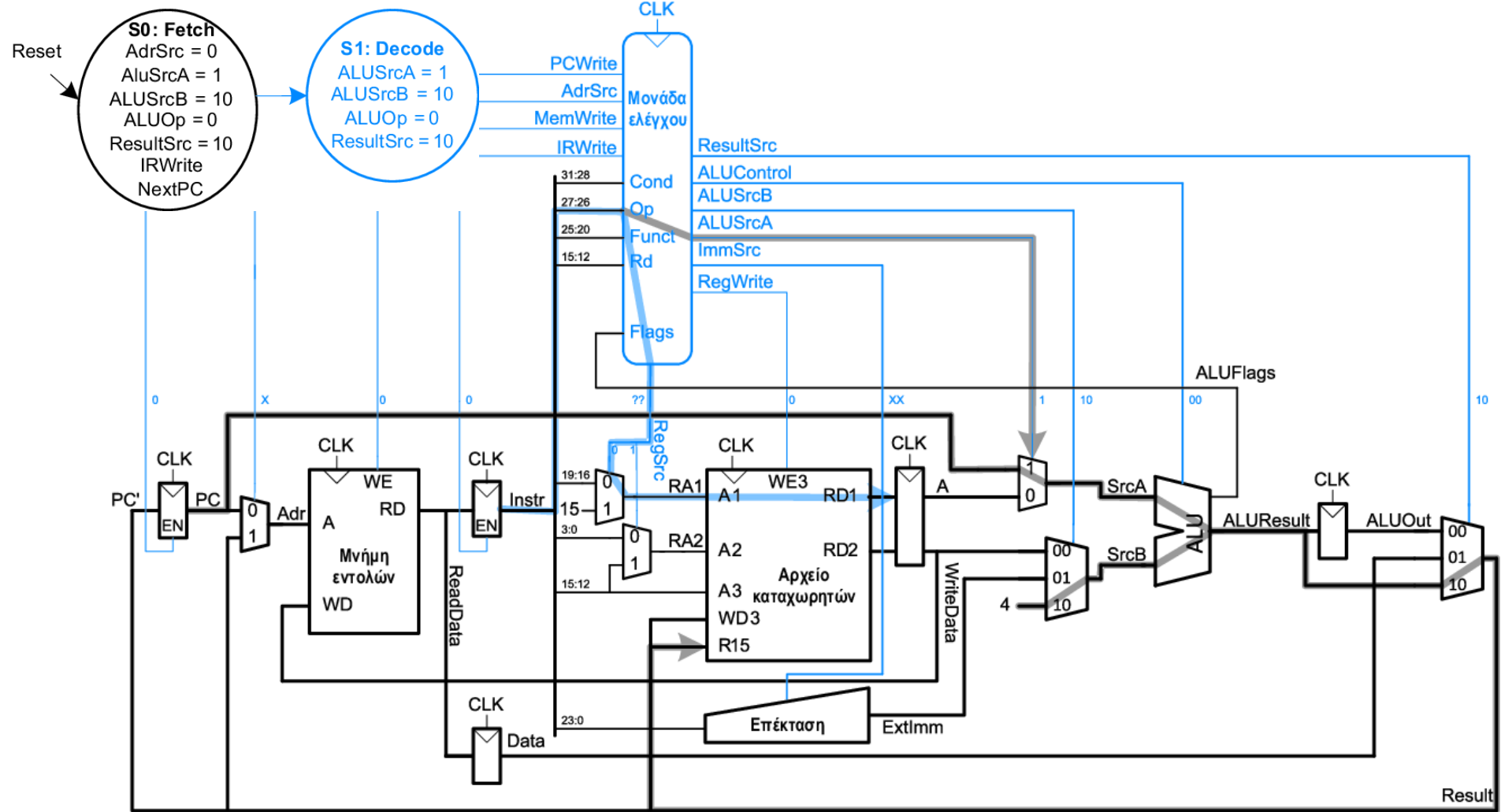
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <110>



Μηχανή FSM κύριου αποκωδικοποιητή: Αποκωδικοποίηση



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

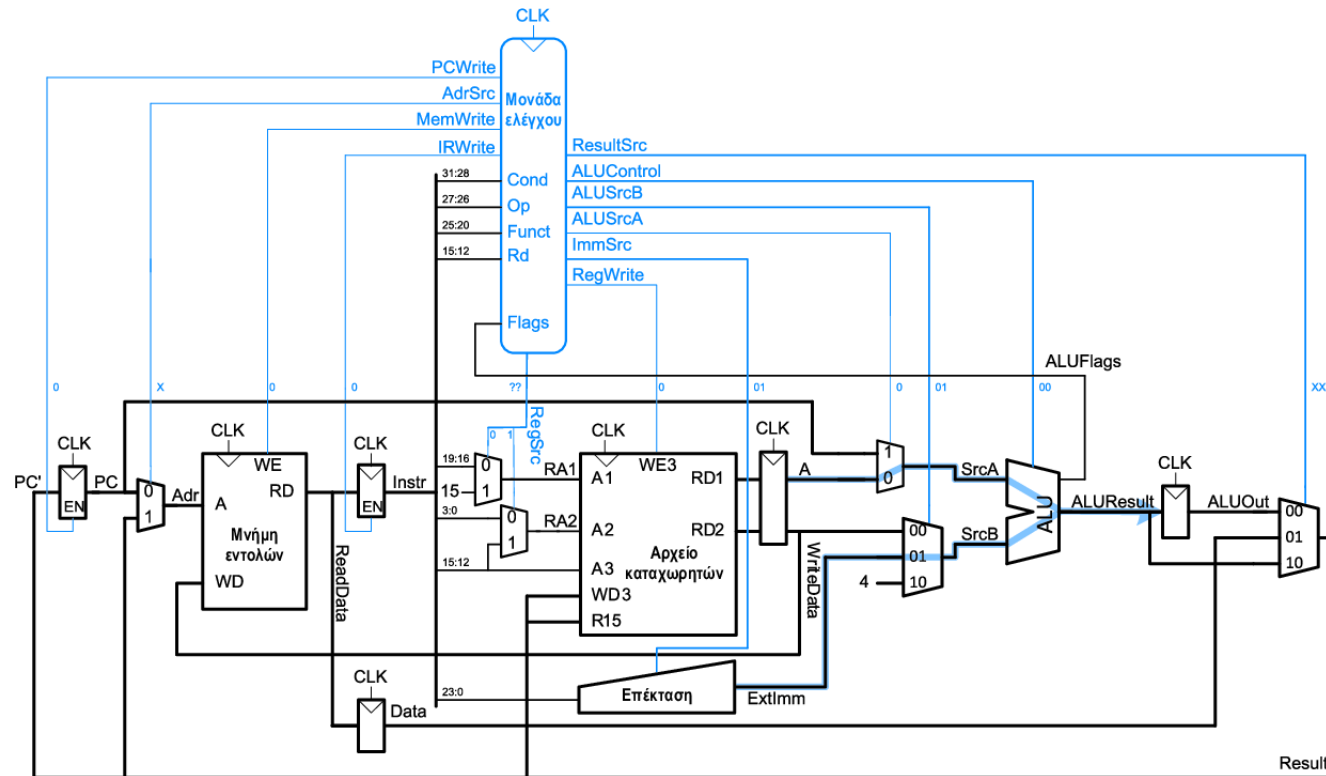
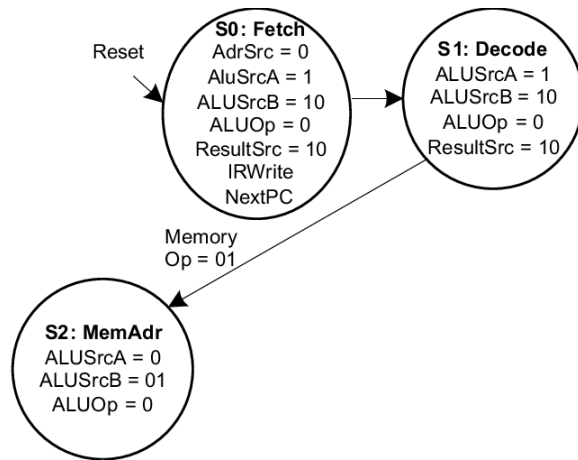
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτότυπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

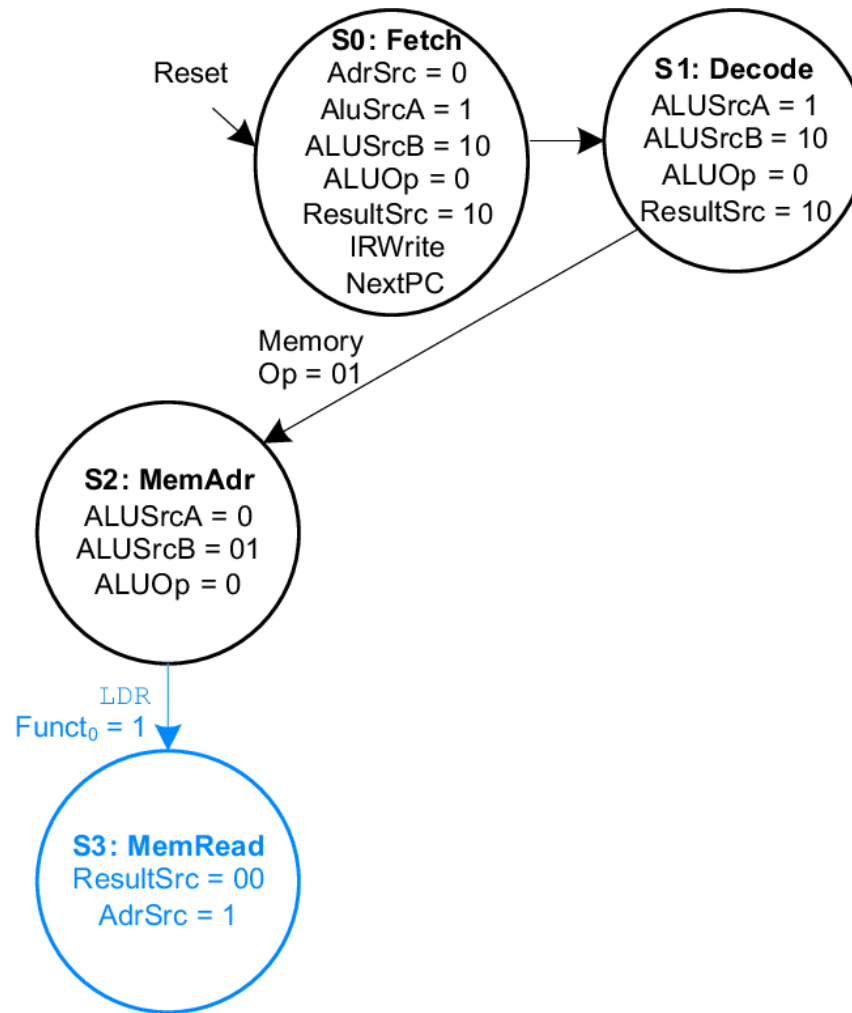
Κεφάλαιο 7 <111>



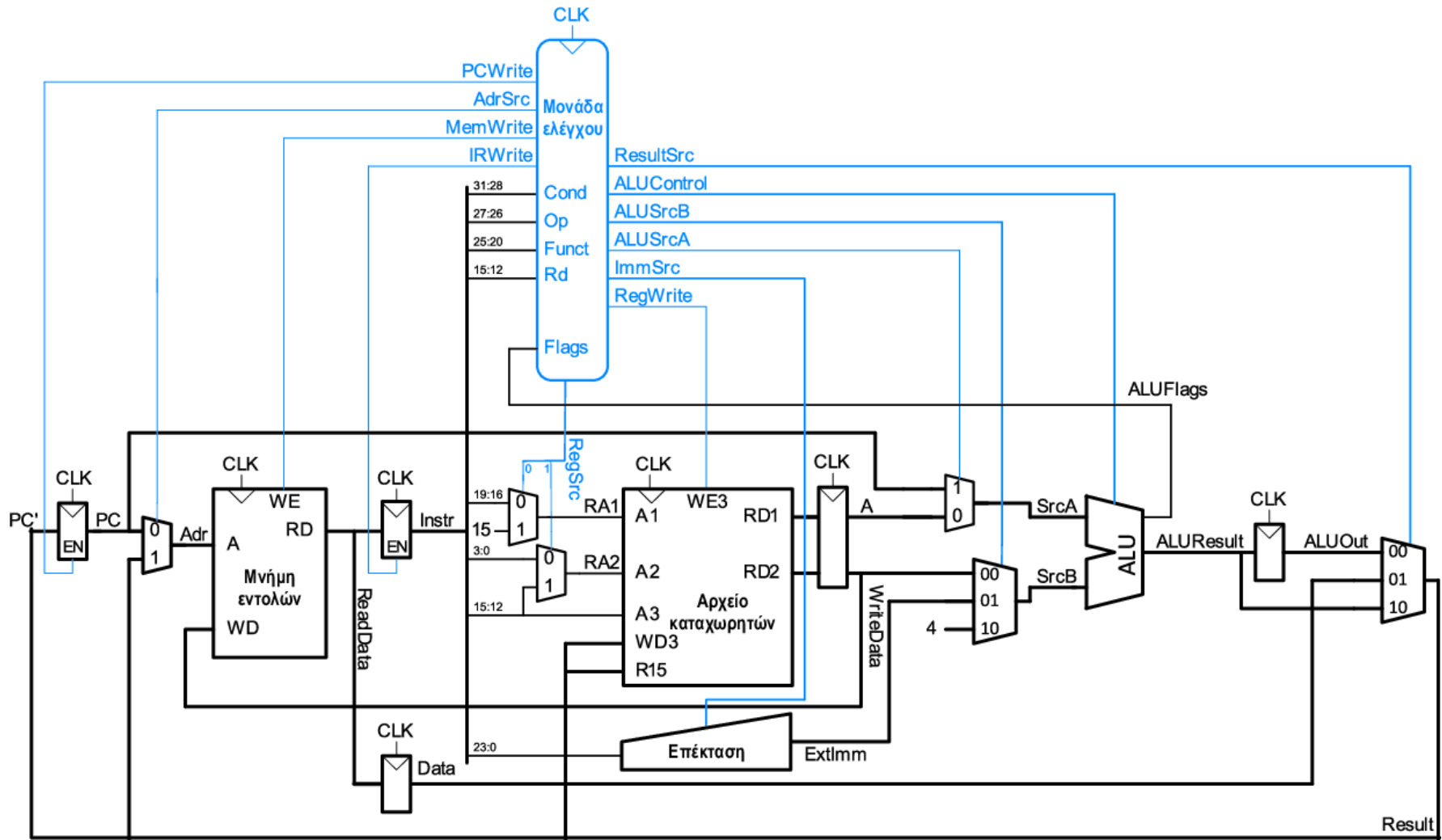
Μηχανή FSM κύριου αποκωδικοποιητή: Διεύθυνση



Μηχανή FSM κύριου αποκωδικοποιητή: Ανάγνωση από τη μνήμη



Επεξεργαστής ARM πολλών κύκλων



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

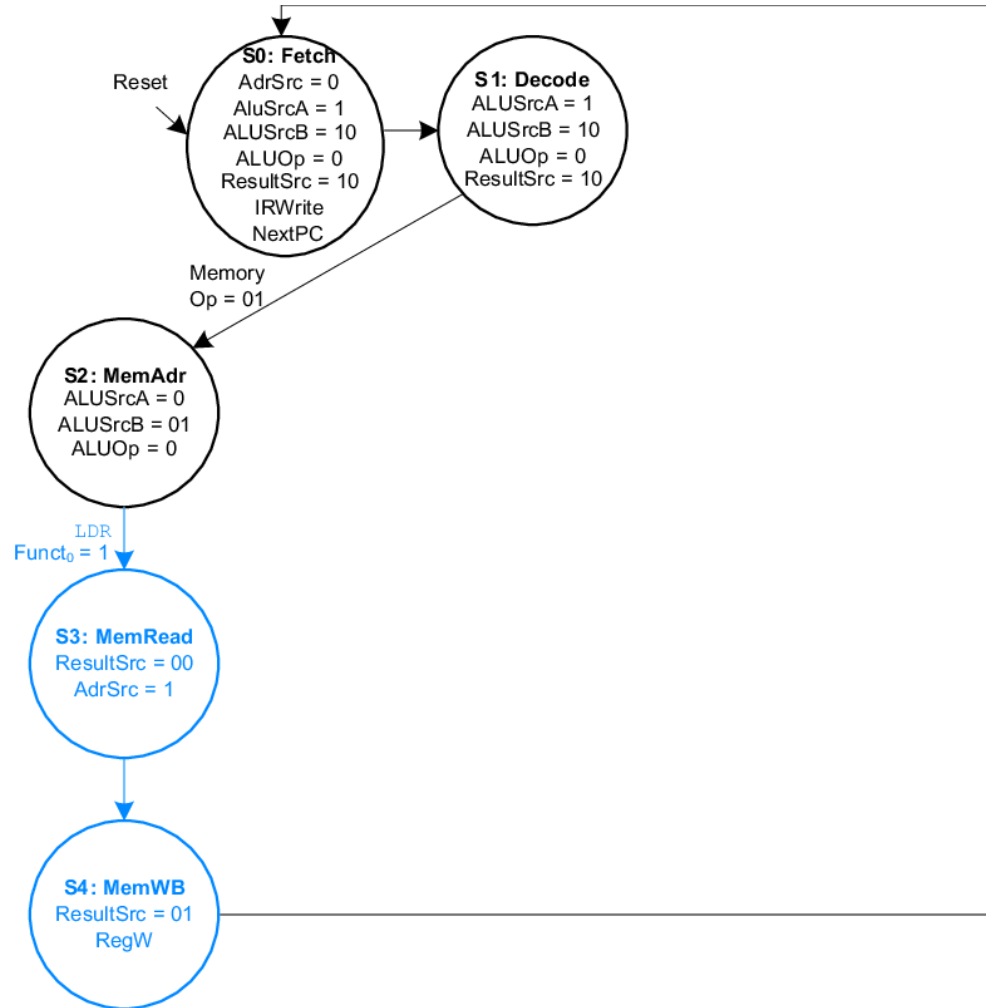
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτότυπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

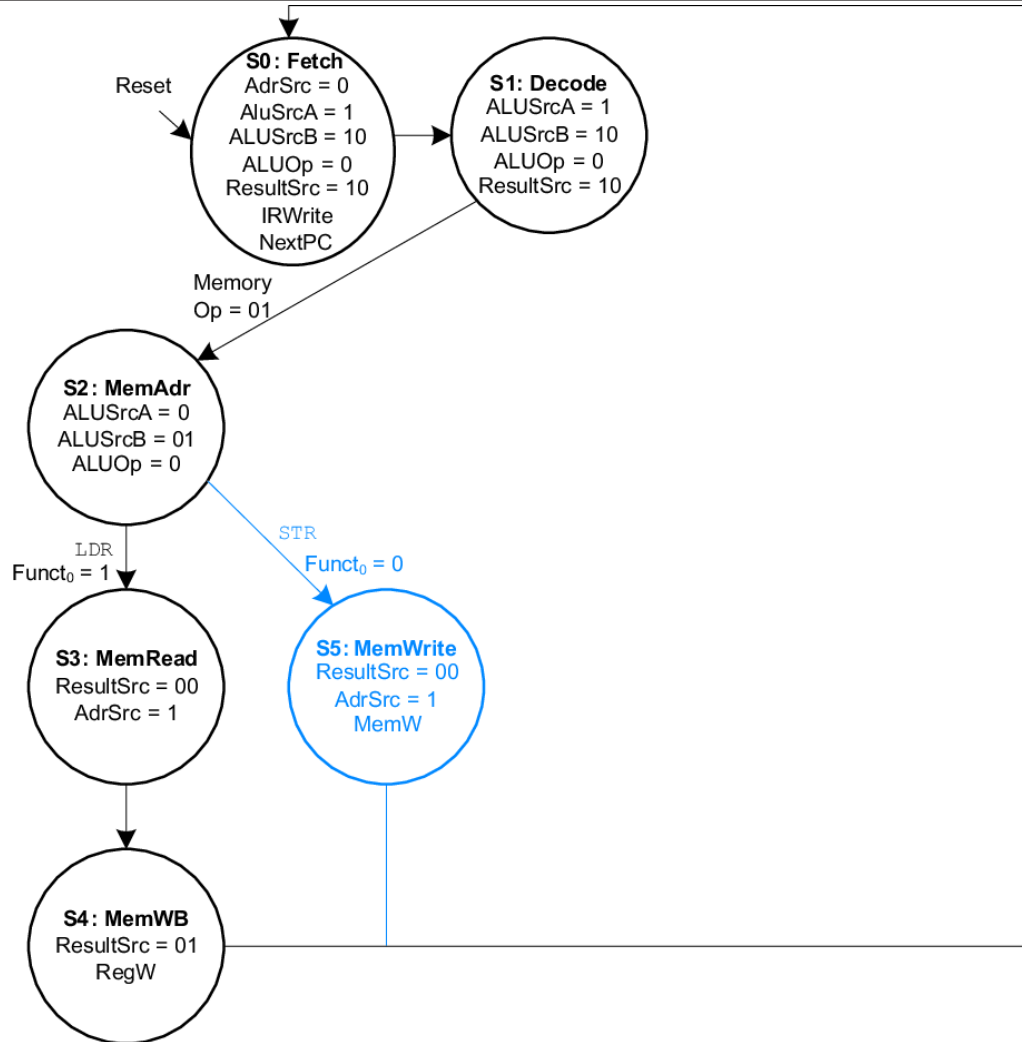
Κεφάλαιο 7 <114>



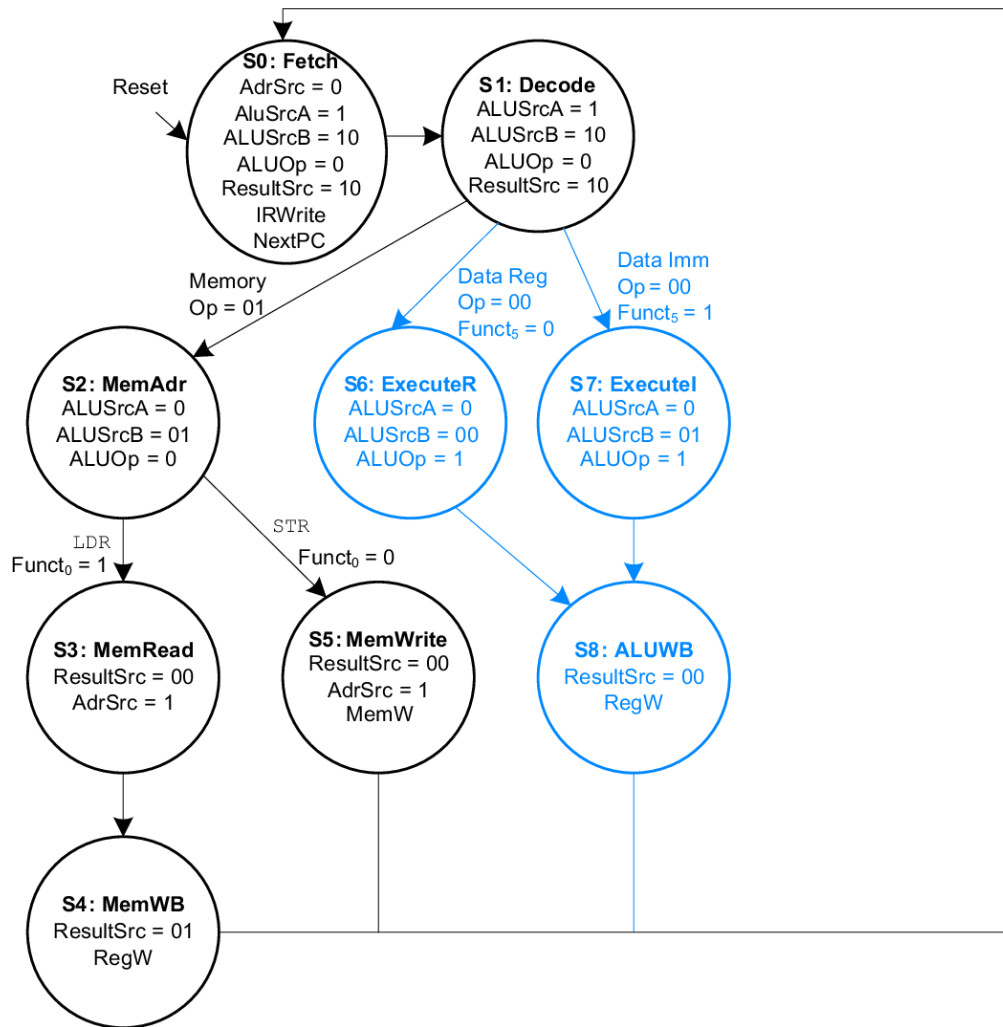
Μηχανή FSM κύριου αποκωδικοποιητή: Εντολή LDR



Μηχανή FSM κύριου αποκωδικοποιητή: Εντολή STR



Μηχανή FSM κύριου αποκωδικοποιητή: Επεξεργασία δεδομένων



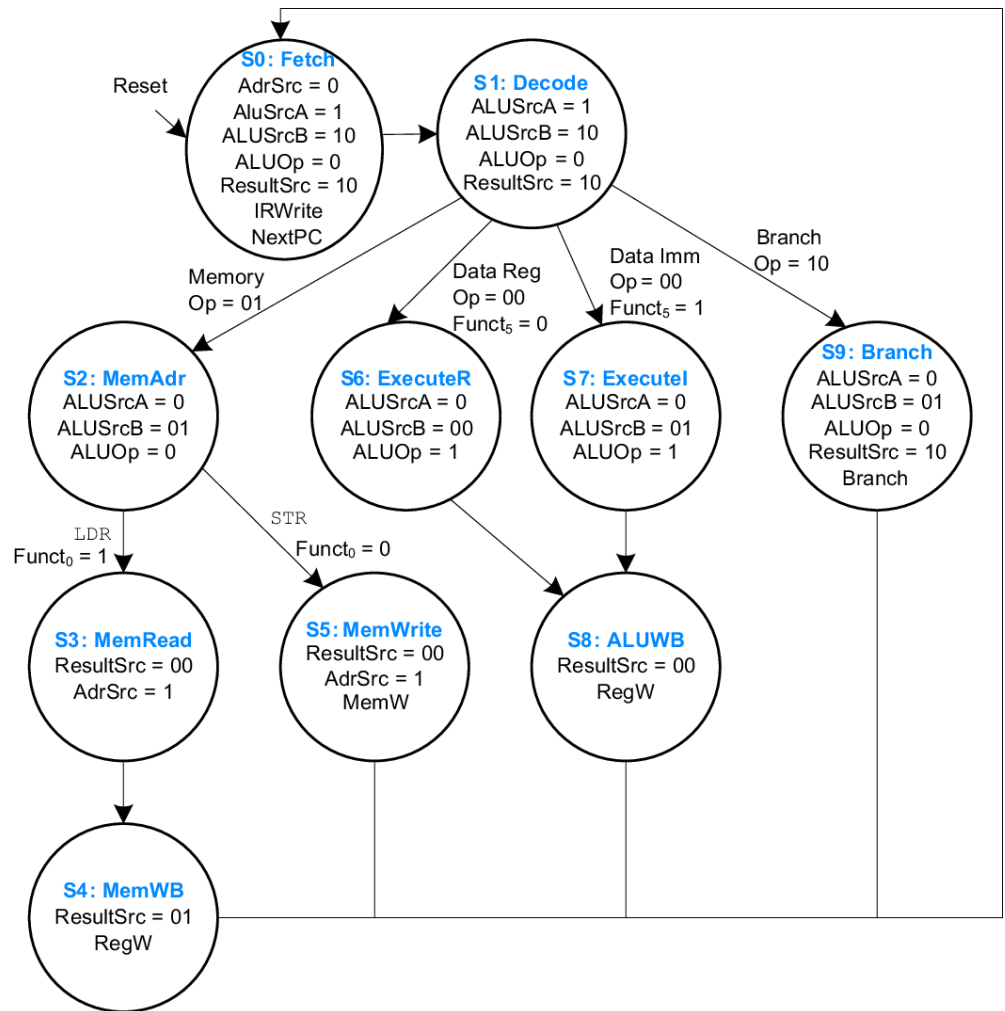
Μηχανή FSM κύριου αποκωδικοποιητή πολλών κύκλων

State

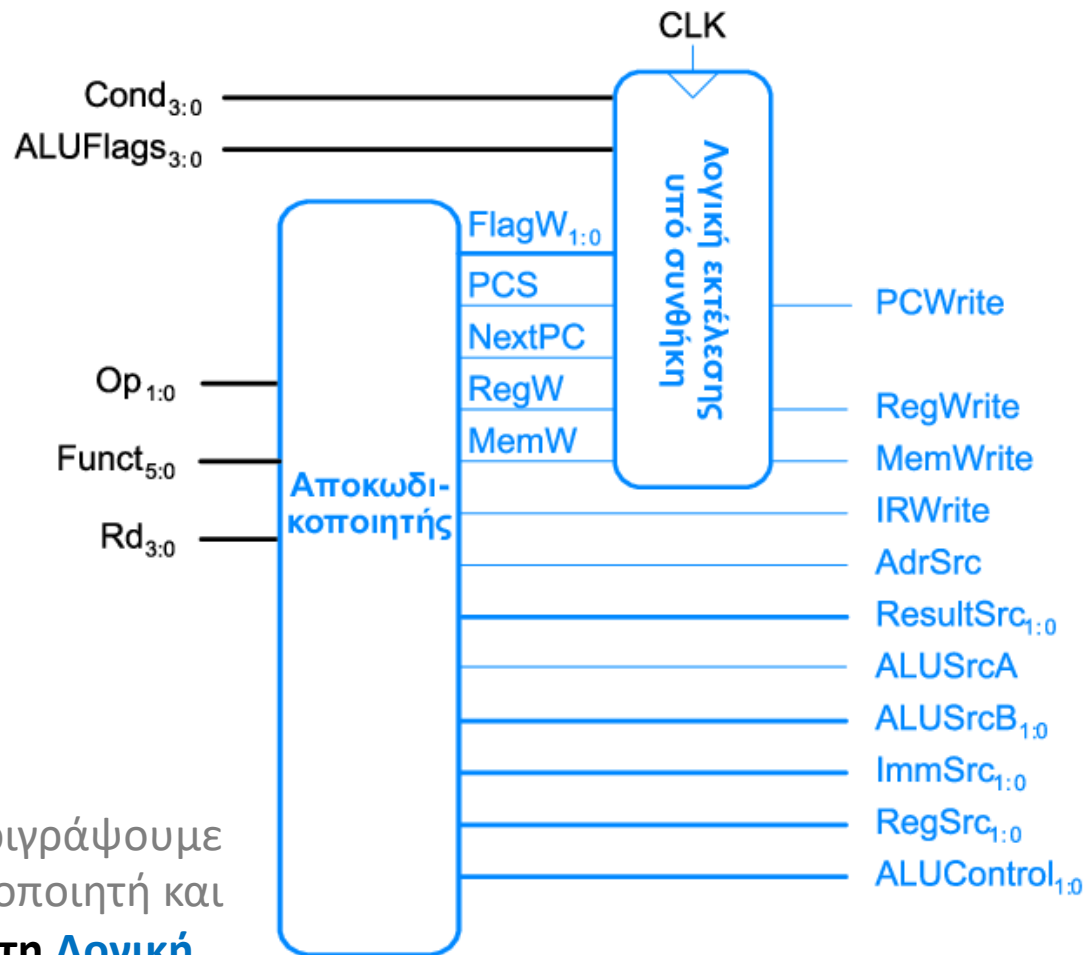
Fetch
 Decode
 MemAdr
 MemRead
 MemWB
 MemWrite
 ExecuteR
 Executel
 ALUWB
 Branch

Datapath μ Op

Instr \leftarrow Mem[PC]; PC \leftarrow PC+4
 ALUOut \leftarrow PC +4
 ALUOut \leftarrow Rn + Imm
 Data \leftarrow Mem[ALUOut]
 Rd \leftarrow Data
 Mem[ALUOut] \leftarrow Rd
 ALUOut \leftarrow Rn op Rm
 ALUOut \leftarrow Rn op Imm
 Rd \leftarrow ALUOut
 PC \leftarrow R15 + offset



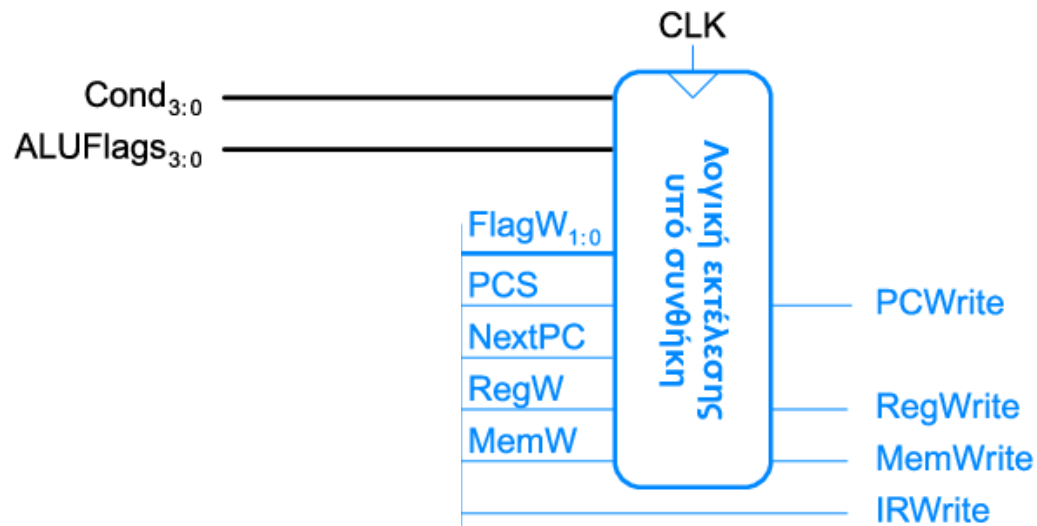
Μονάδα ελέγχου πολλών κύκλων



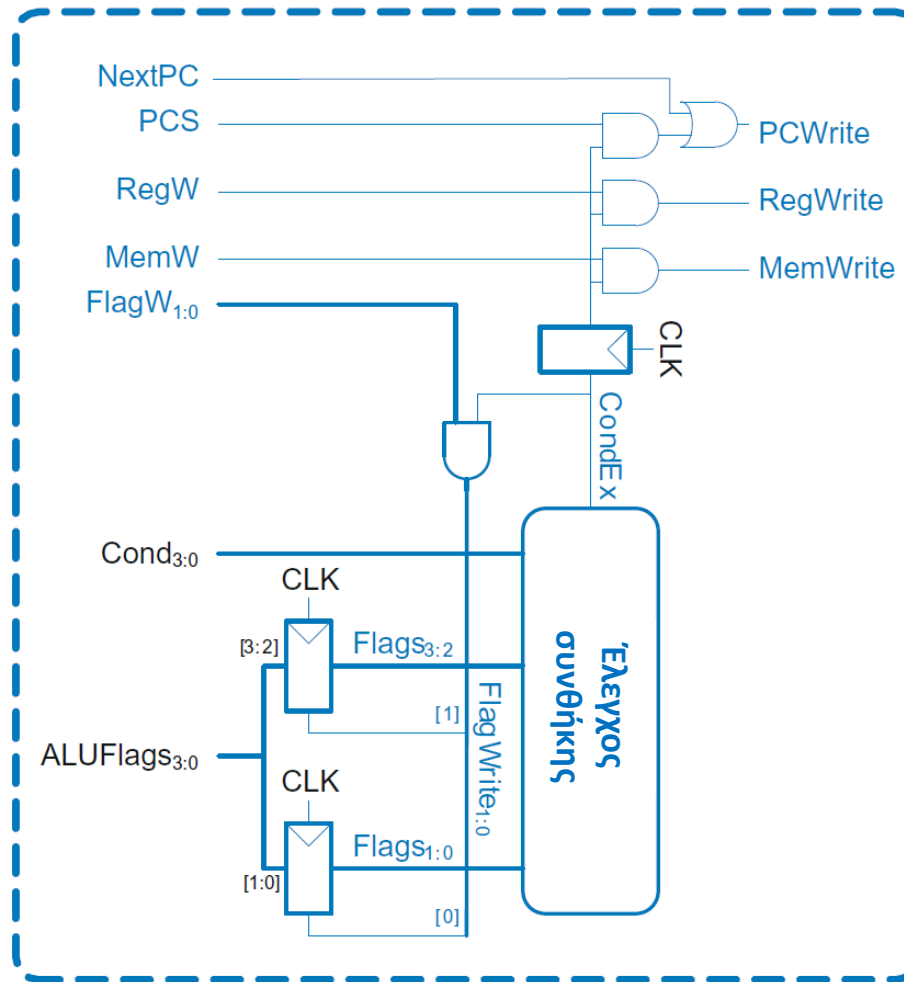
- Πρώτα θα περιγράψουμε τον Αποκωδικοποιητή και
- στη συνέχεια τη **Λογική εκτέλεσης υπό συνθήκη**



Μονάδα ελέγχου πολλών κύκλων: Λογική εκτέλεσης υπό συνθήκη



Μονάδα ελέγχου ενός κύκλου: Λογική εκτέλεσης υπό συνθήκη



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

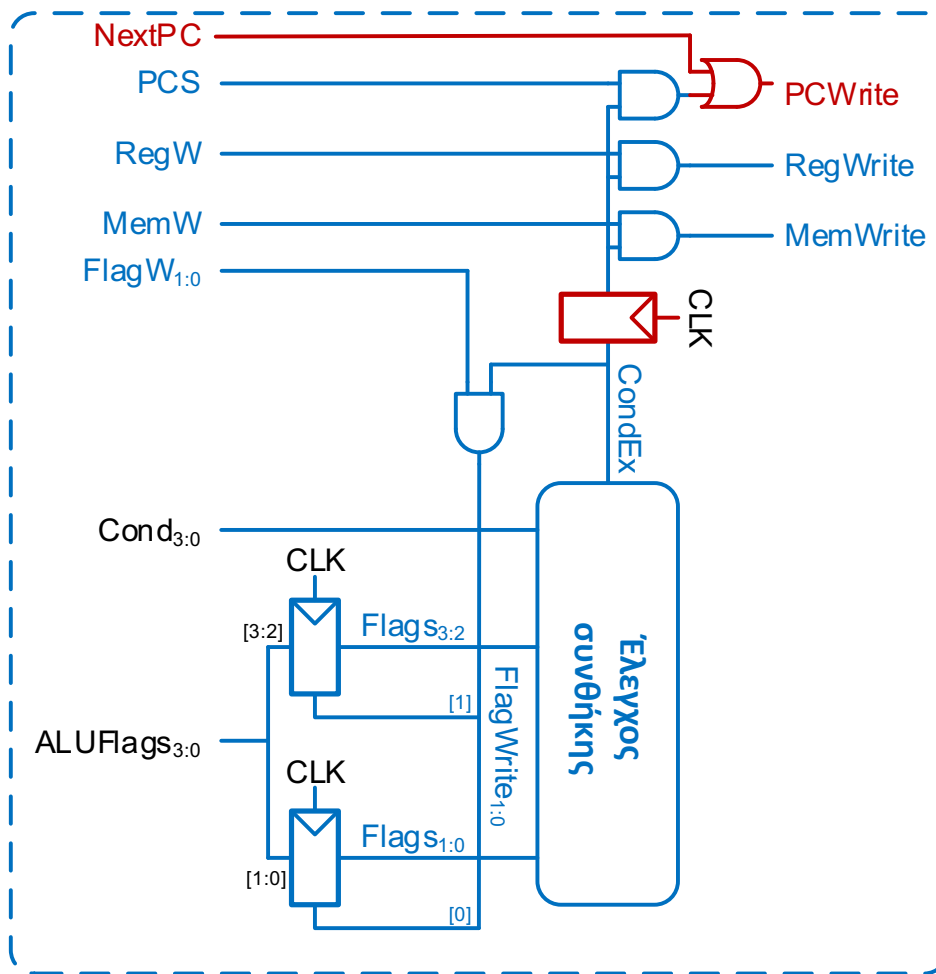
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <121>



Μονάδα ελέγχου πολλών κύκλων: Λογική εκτέλεσης υπό συνθήκη



- Το **PCWrite** ενεργοποιείται στην κατάσταση προσκόμισης
- Κατάσταση **ExecuteI/ExecuteR**: Ενεργοποιείται το **CondEX** Παράγεται το **ALUFlags**
- Κατάσταση **ALUWB**: Ενημερώνεται το **Flags** Μεταβάλλεται το **CondEX** Δεν μεταβάλλονται τα **PCWrite**, **RegWrite** και **MemWrite** μέχρι τη νέα εντολή (κατάσταση Προσκόμισης)



Επιδόσεις επεξεργαστή πολλών κύκλων

- Οι εντολές απαιτούν διαφορετικό πλήθος κύκλων



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <123>

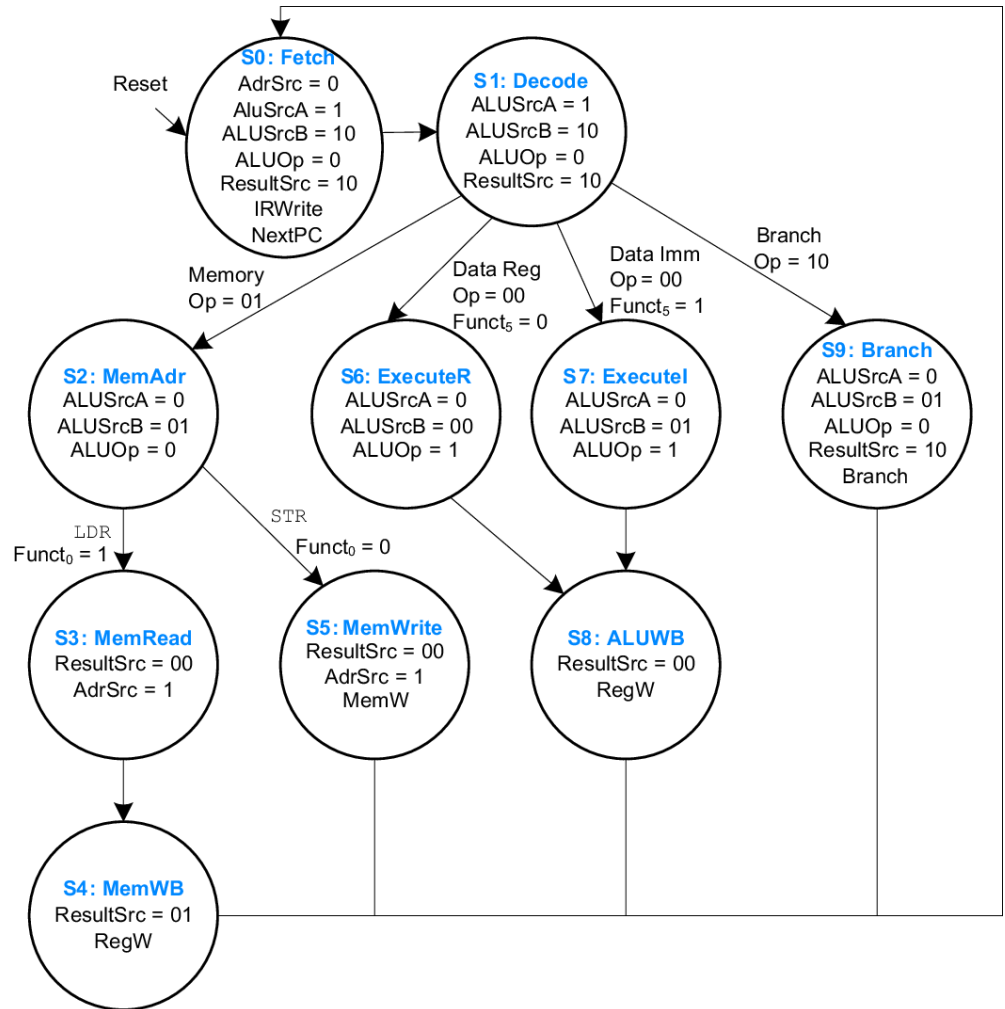
Μηχανή FSM κύριου αποκωδικοποιητή πολλών κύκλων

State

Fetch
 Decode
 MemAdr
 MemRead
 MemWB
 MemWrite
 ExecuteR
 Executel
 ALUWB
 Branch

Datapath μ Op

Instr \leftarrow Mem[PC]; PC \leftarrow PC+4
 ALUOut \leftarrow PC +4
 ALUOut \leftarrow Rn + Imm
 Data \leftarrow Mem[ALUOut]
 Rd \leftarrow Data
 Mem[ALUOut] \leftarrow Rd
 ALUOut \leftarrow Rn op Rm
 ALUOut \leftarrow Rn op Imm
 Rd \leftarrow ALUOut
 PC \leftarrow R15 + offset



Επιδόσεις επεξεργαστή πολλών κύκλων

- Οι εντολές απαιτούν διαφορετικό πλήθος κύκλων:
 - 3 κύκλοι:
 - 4 κύκλοι:
 - 5 κύκλοι:



Επιδόσεις επεξεργαστή πολλών κύκλων

- Οι εντολές απαιτούν διαφορετικό πλήθος κύκλων:
 - 3 κύκλοι: B
 - 4 κύκλοι: DP, STR
 - 5 κύκλοι: LDR



Επιδόσεις επεξεργαστή πολλών κύκλων

- Οι εντολές απαιτούν διαφορετικό πλήθος κύκλων:
 - 3 κύκλοι: B
 - 4 κύκλοι: DP, STR
 - 5 κύκλοι: LDR
- Ο δείκτης CPI είναι σταθμισμένος μέσος όρος
- Μετροπρόγραμμα SPECINT2000:
 - 25% για τις εντολές φόρτωσης
 - 10% για τις εντολές αποθήκευσης
 - 13% για τις εντολές διακλάδωσης
 - 52% για τις εντολές επεξεργασίας δεδομένων



Επιδόσεις επεξεργαστή πολλών κύκλων

- Οι εντολές απαιτούν διαφορετικό πλήθος κύκλων:
 - 3 κύκλοι: B
 - 4 κύκλοι: DP, STR
 - 5 κύκλοι: LDR
- Ο δείκτης CPI είναι σταθμισμένος μέσος όρος
- Μετροπρόγραμμα SPECINT2000:
 - **25%** για τις εντολές φόρτωσης
 - **10%** για τις εντολές αποθήκευσης
 - **13%** για τις εντολές διακλάδωσης
 - **52%** για τις εντολές επεξεργασίας δεδομένων

$$\text{Μέσος δείκτης CPI} = (0.13)(3) + (0.52 + 0.10)(4) + (0.25)(5) = 4.12$$



Επιδόσεις επεξεργαστή πολλών κύκλων

Κρίσιμη διαδρομή πολλών κύκλων:

- Παραδοχές:
 - Το RF είναι γρηγορότερο από τη μνήμη
 - Η εγγραφή στη μνήμη είναι γρηγορότερη από ό,τι η ανάγνωση από τη μνήμη

$$T_{c2} = t_{pcq} + 2t_{mux} + \max(t_{ALU} + t_{mux}, t_{mem}) + t_{setup}$$



Επιδόσεις επεξεργαστή πολλών κύκλων: Παράδειγμα

Στοιχείο	Παράμετρος	Καθυστέρηση (ps)
Καθυστέρηση από το ρολόι έως την έξοδο Q (καταχωρητής)	t_{pcq_PC}	40
Σταθεροποίηση καταχωρητή	t_{setup}	50
Πολυπλέκτης	t_{mux}	25
Μονάδα ALU	t_{ALU}	120
Αποκωδικοποιητής	t_{dec}	70
Ανάγνωση από τη μνήμη	t_{mem}	200
Ανάγνωση από το αρχείο καταχωρητών	t_{RFread}	100
Σταθεροποίηση αρχείου καταχωρητών	$t_{RFsetup}$	60

$$T_{c2} = ;$$



Επιδόσεις επεξεργαστή πολλών κύκλων: Παράδειγμα

Στοιχείο	Παράμετρος	Καθυστέρηση (ps)
Καθυστέρηση από το ρολόι έως την έξοδο Q (καταχωρητής)	t_{pcq_PC}	40
Σταθεροποίηση καταχωρητή	t_{setup}	50
Πολυπλέκτης	t_{mux}	25
Μονάδα ALU	t_{ALU}	120
Αποκωδικοποιητής	t_{dec}	70
Ανάγνωση από τη μνήμη	t_{mem}	200
Ανάγνωση από το αρχείο καταχωρητών	t_{RFread}	100
Σταθεροποίηση αρχείου καταχωρητών	$t_{RFsetup}$	60

$$\begin{aligned} T_{c2} &= t_{pcq} + 2t_{mux} + \max[t_{ALU} + t_{mux}, t_{mem}] + t_{setup} \\ &= [40 + 2(25) + 200 + 50] \text{ ps} = \mathbf{340 \text{ ps}} \end{aligned}$$



Επιδόσεις επεξεργαστή πολλών κύκλων: Παράδειγμα

Για ένα πρόγραμμα με **100 δισεκατομμύρια** εντολές που εκτελούνται σε έναν επεξεργαστή ARM **πολλών κύκλων**

- **CPI = 4.12** κύκλοι/εντολές
- **Χρόνος κύκλου ρολογιού: $T_{c2} = 340$ ps**

Χρόνος εκτέλεσης = ;



Επιδόσεις επεξεργαστή πολλών κύκλων: Παράδειγμα

Για ένα πρόγραμμα με **100 δισεκατομμύρια** εντολές που εκτελούνται σε έναν επεξεργαστή ARM **πολλών κύκλων**

- **CPI = 4.12** κύκλοι/εντολές
- **Χρόνος κύκλου ρολογιού: $T_{c2} = 340$ ps**

$$\begin{aligned}\text{Χρόνος εκτέλεσης} &= (\text{πλήθος εντολών}) \times \text{CPI} \times T_c \\ &= (100 \times 10^9)(4.12)(340 \times 10^{-12}) \\ &= \mathbf{140 \text{ δευτερόλεπτα}}\end{aligned}$$



Επιδόσεις επεξεργαστή πολλών κύκλων: Παράδειγμα

Για ένα πρόγραμμα με **100 δισεκατομμύρια** εντολές που εκτελούνται σε έναν επεξεργαστή ARM **πολλών κύκλων**

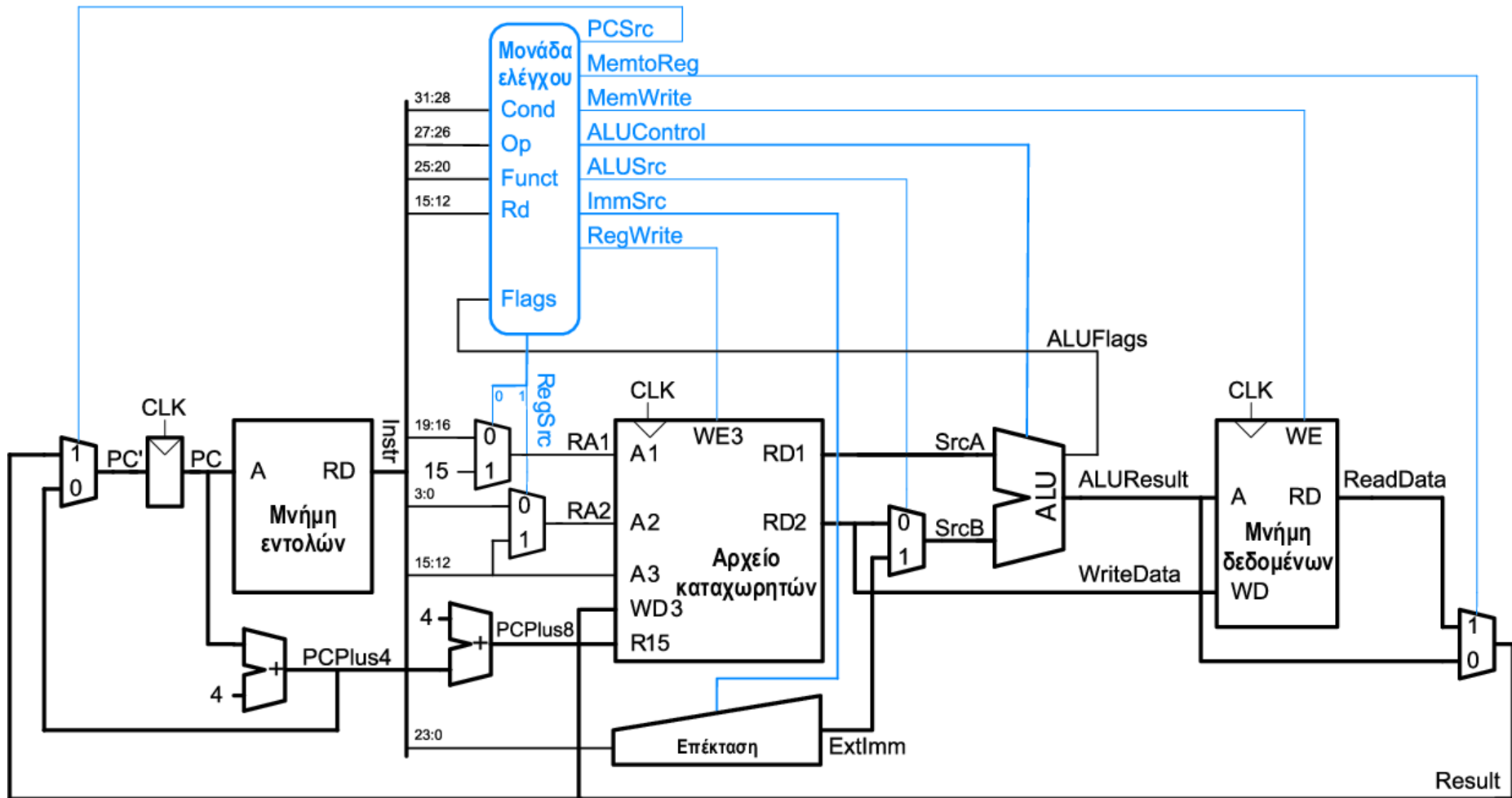
- **CPI = 4.12** κύκλοι/εντολές
- **Χρόνος κύκλου ρολογιού: $T_{c2} = 340$ ps**

$$\begin{aligned}\text{Χρόνος εκτέλεσης} &= (\text{πλήθος εντολών}) \times \text{CPI} \times T_c \\ &= (100 \times 10^9)(4.12)(340 \times 10^{-12}) \\ &= \mathbf{140 \text{ δευτερόλεπτα}}\end{aligned}$$

Είναι **πιο αργός** από τον επεξεργαστή ενός κύκλου (84 δευτερόλεπτα)



Επανάληψη: Επεξεργαστής ARM ενός κύκλου



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

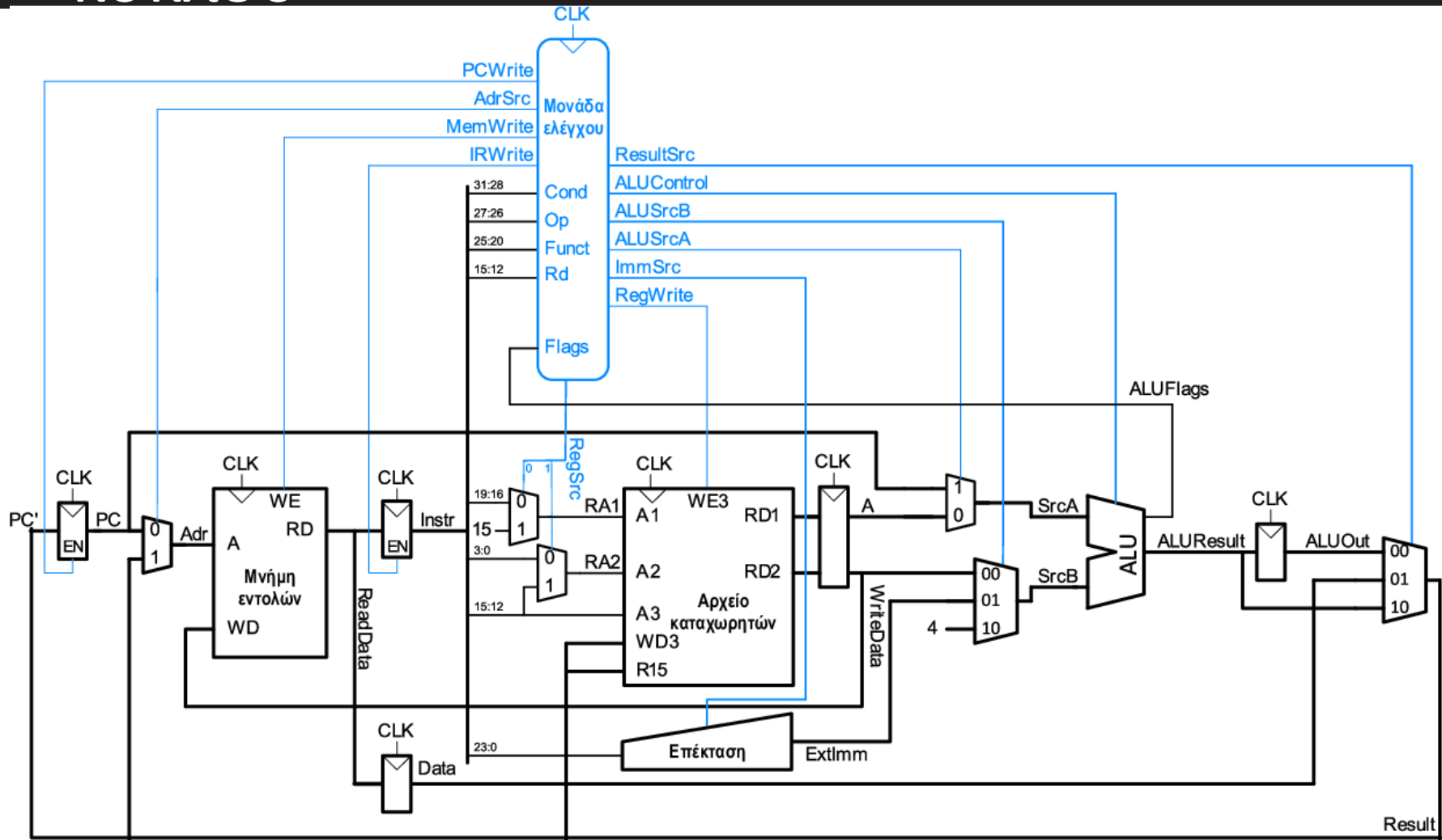
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτότυπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <135>



Επανάληψη: Επεξεργαστής ARM ενός κύκλου



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτότυπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <136>



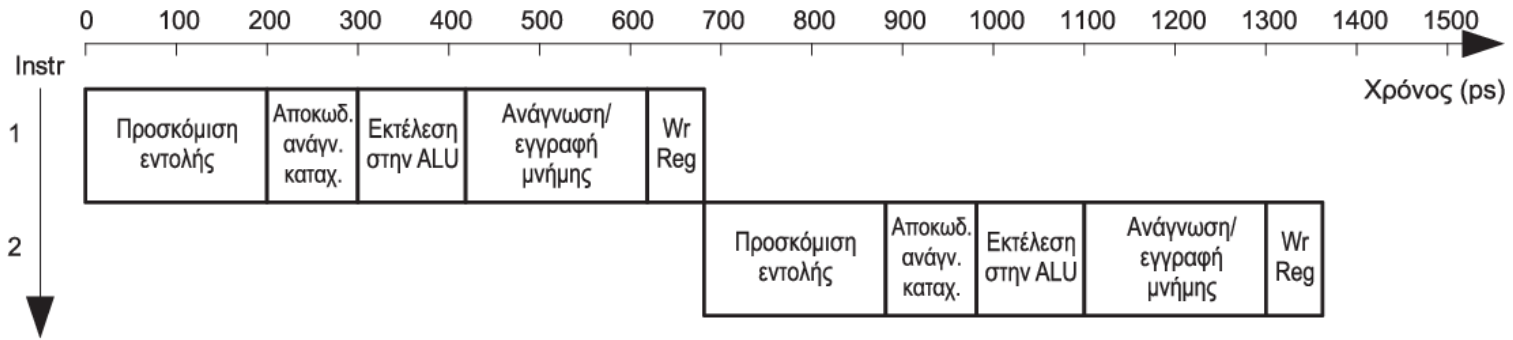
Επεξεργαστής ARM με υποστήριξη διοχέτευσης

- Χρονικός παραλληλισμός
- Ο επεξεργαστής ενός κύκλου χωρίζεται σε 5 στάδια:
 - *Fetch* (προσκόμιση)
 - *Decode* (αποκωδικοποίηση)
 - *Execute* (εκτέλεση)
 - *Memory* (μνήμη)
 - *Writeback* (ετεροχρονισμένη εγγραφή)
- Προστίθενται καταχωρητές διοχέτευσης μεταξύ των σταδίων



Σύγκριση επεξεργαστή ενός κύκλου και επεξεργαστή με υποστήριξη διοχέτευσης

Ενός κύκλου



Με διοχέτευση



* Wr Reg = Ετεροχρονισμένη εγγραφή καταχωρητών

Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

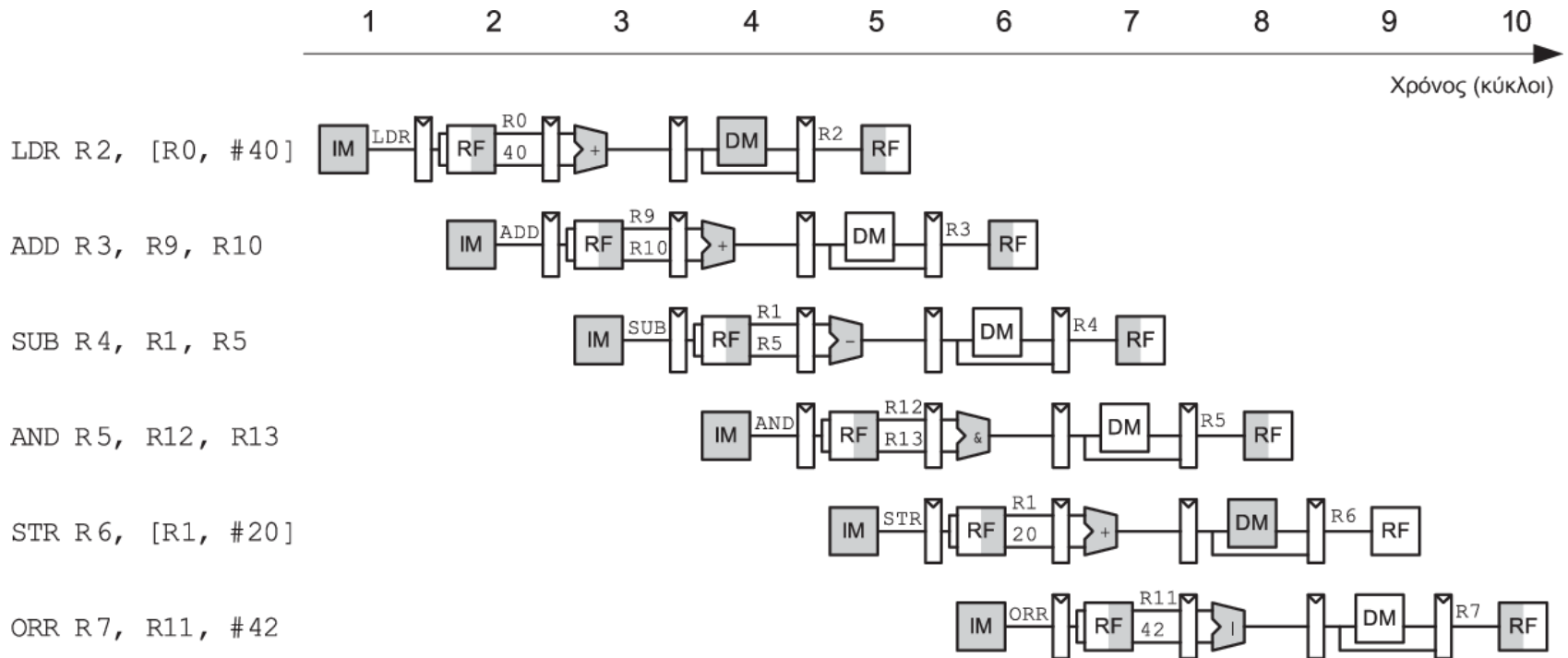
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <138>

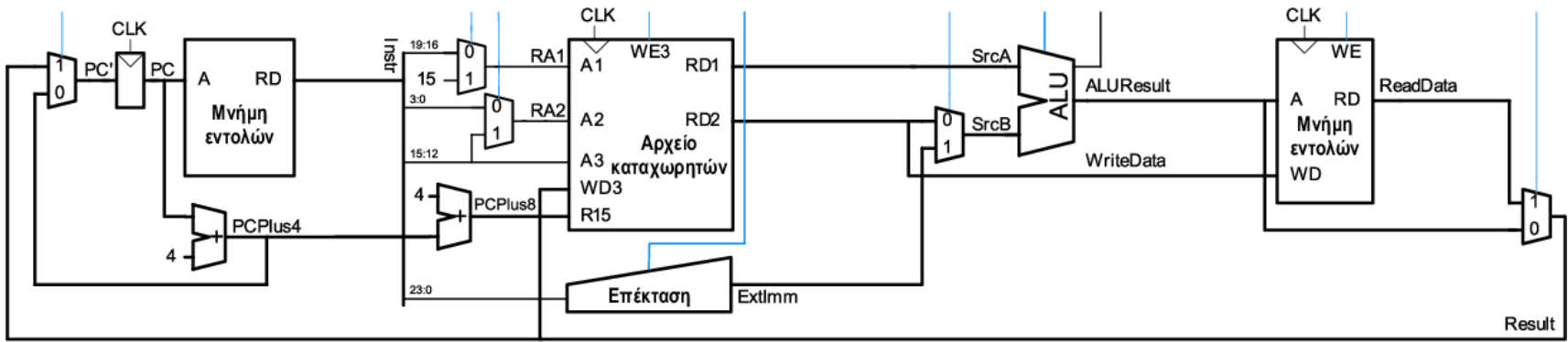


Επεξεργαστής ARM με υποστήριξη διοχέτευσης: Αφηρημένη θεώρηση

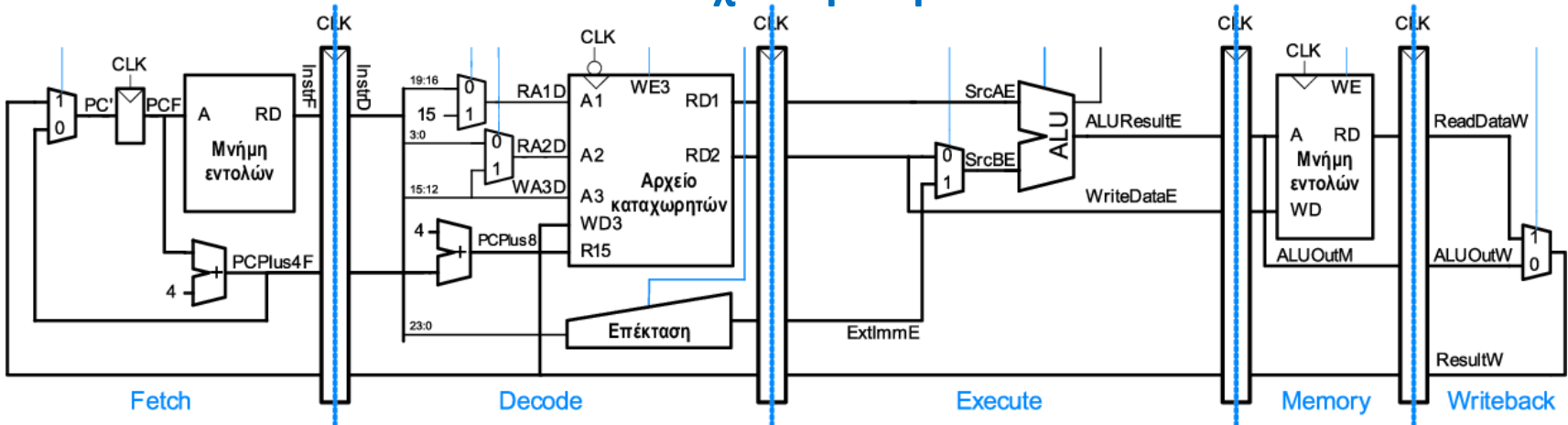


Σύγκριση διαδρομής δεδομένων: Ενός κύκλου και διοχετευμένης

Ενός κύκλου



Διοχετευμένη



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

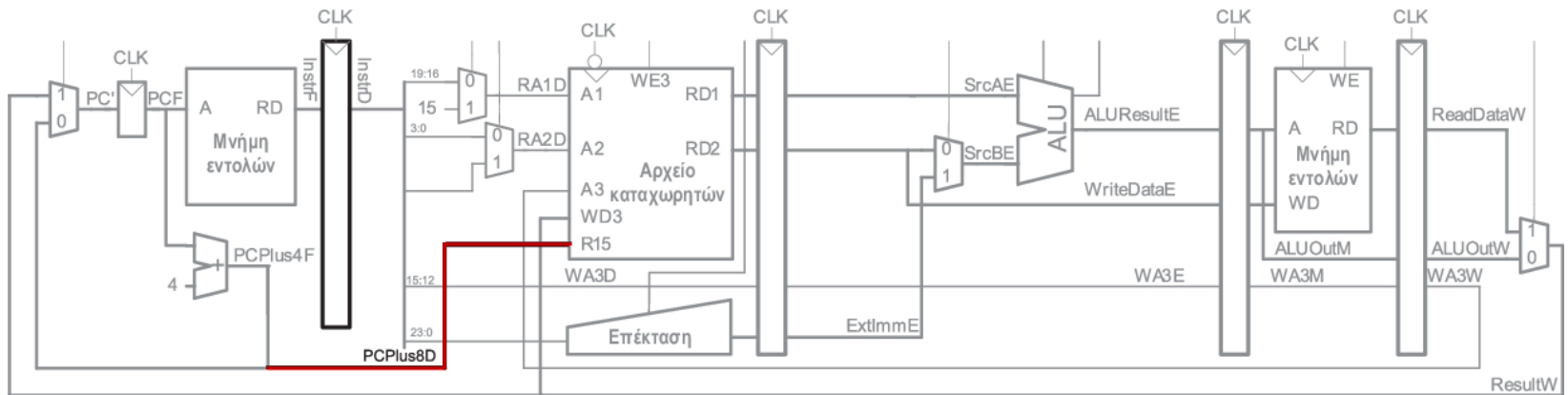
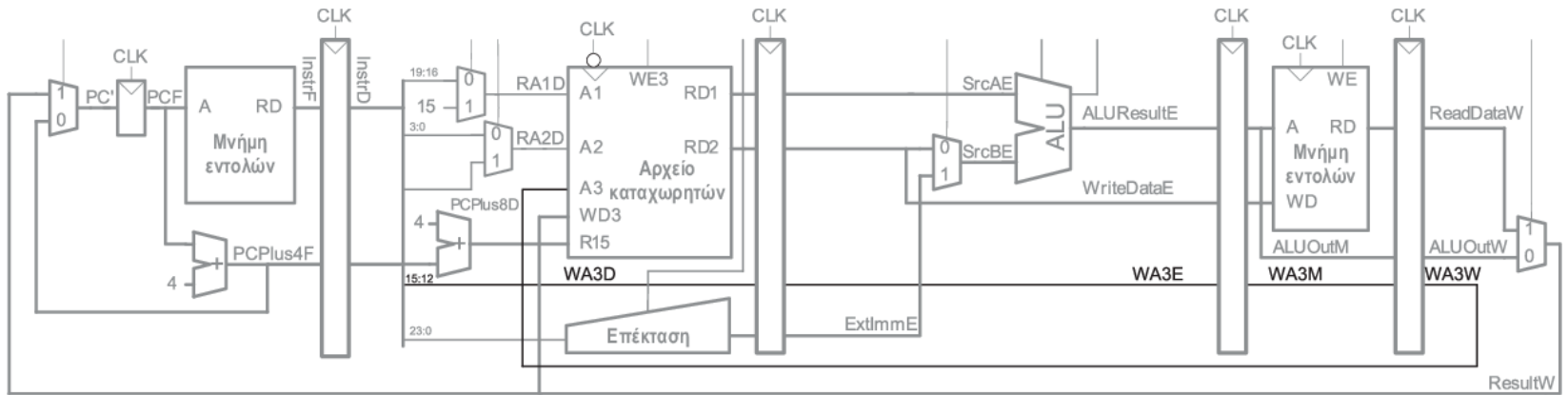
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτότυπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <140>



Βελτιστοποιημένη διοχετευμένη διαδρομή δεδομένων



Εξοικονόμηση ενός αθροιστή: Χρησιμοποιείται το *PCPlus4F* μόλις ο μετρητής *PC* ενημερωθεί με την τιμή *PC+4*

Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

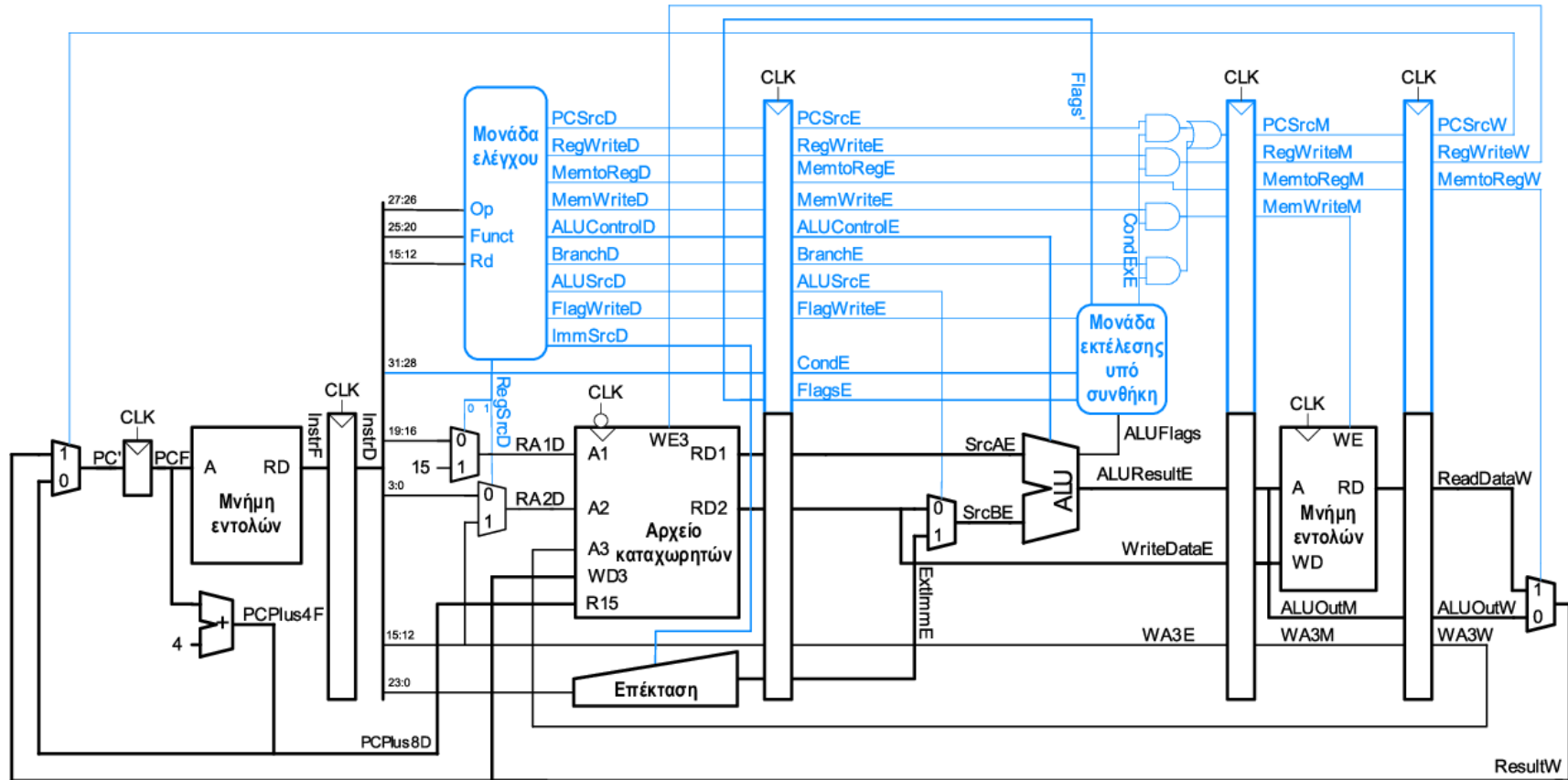
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <142>



Επεξεργαστής με διοχέτευση: Μονάδα ελέγχου



- Ίδια μονάδα ελέγχου όπως στον επεξεργαστή ενός κύκλου
- Ο έλεγχος καθυστερείται μέχρι το κατάλληλο στάδιο διοχέτευσης

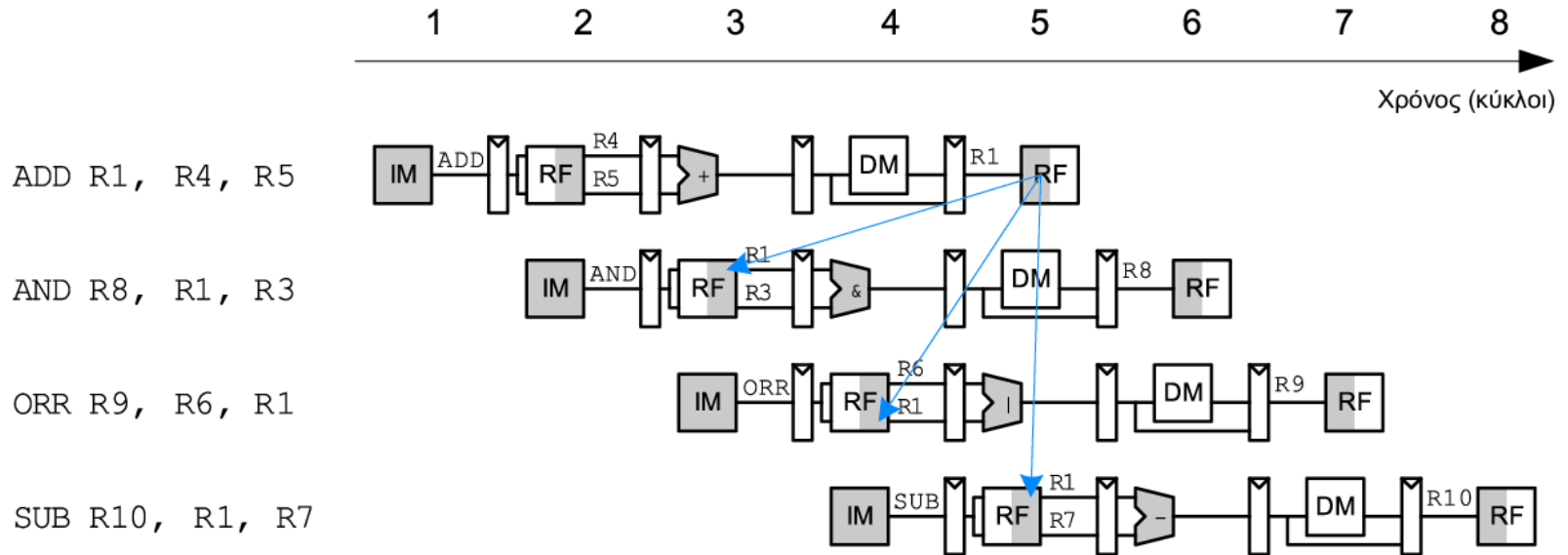


Διοχέτευση: Κίνδυνοι

- Όταν μια εντολή εξαρτάται από το αποτέλεσμα μιας εντολής που δεν έχει ολοκληρωθεί ακόμα
- Τύποι:
 - **Κίνδυνος δεδομένων:** Τα δεδομένα ενός καταχωρητή δεν έχουν ακόμα γραφεί ετεροχρονισμένα στο αρχείο καταχωρητών
 - **Κίνδυνος ελέγχου:** Δεν έχει αποφασιστεί ακόμα ποια θα είναι η επόμενη εντολή (προκαλείται από τη διακλάδωση)



Κίνδυνος δεδομένων



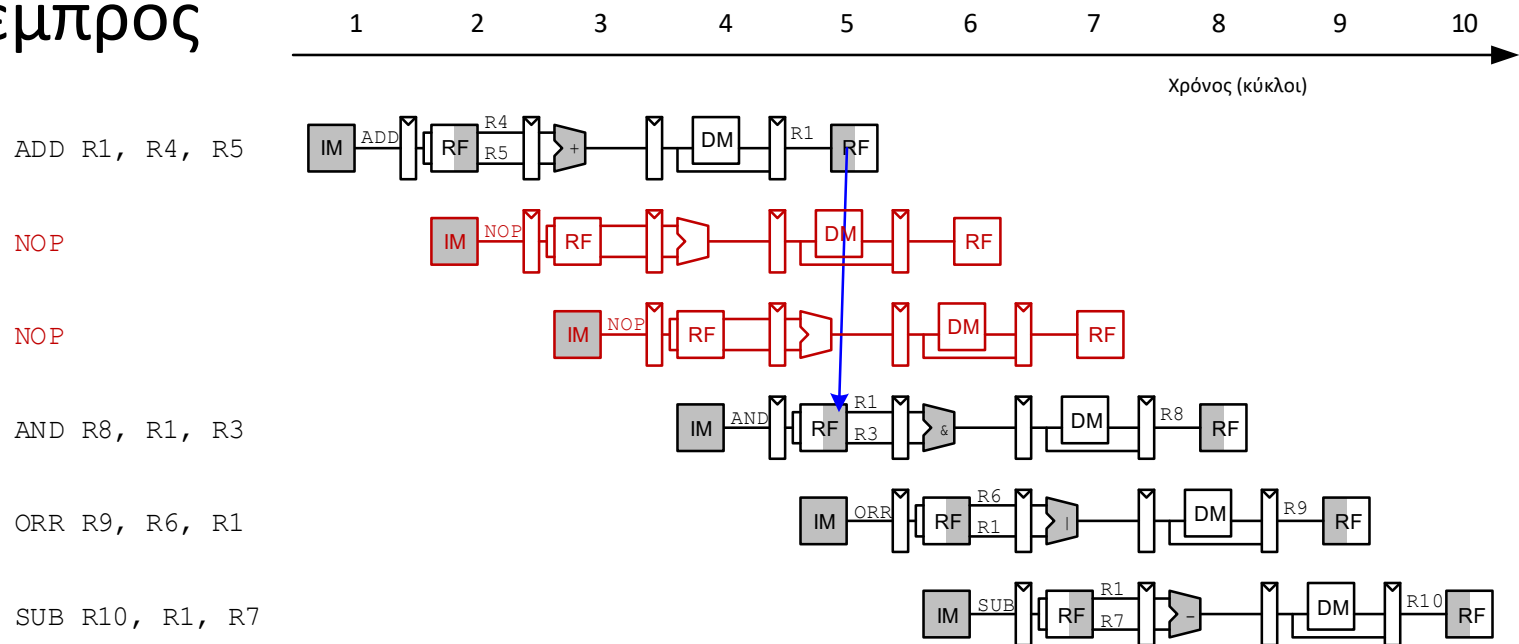
Χειρισμός κινδύνων δεδομένων

- Εισαγωγή εντολών NOP στον κώδικα κατά τον χρόνο μεταγλώττισης
- Αναδιάταξη του κώδικα code κατά τον χρόνο μεταγλώττισης
- Προώθηση δεδομένων κατά τον χρόνο εκτέλεσης
- Παρακώλυση του επεξεργαστή κατά τον χρόνο εκτέλεσης

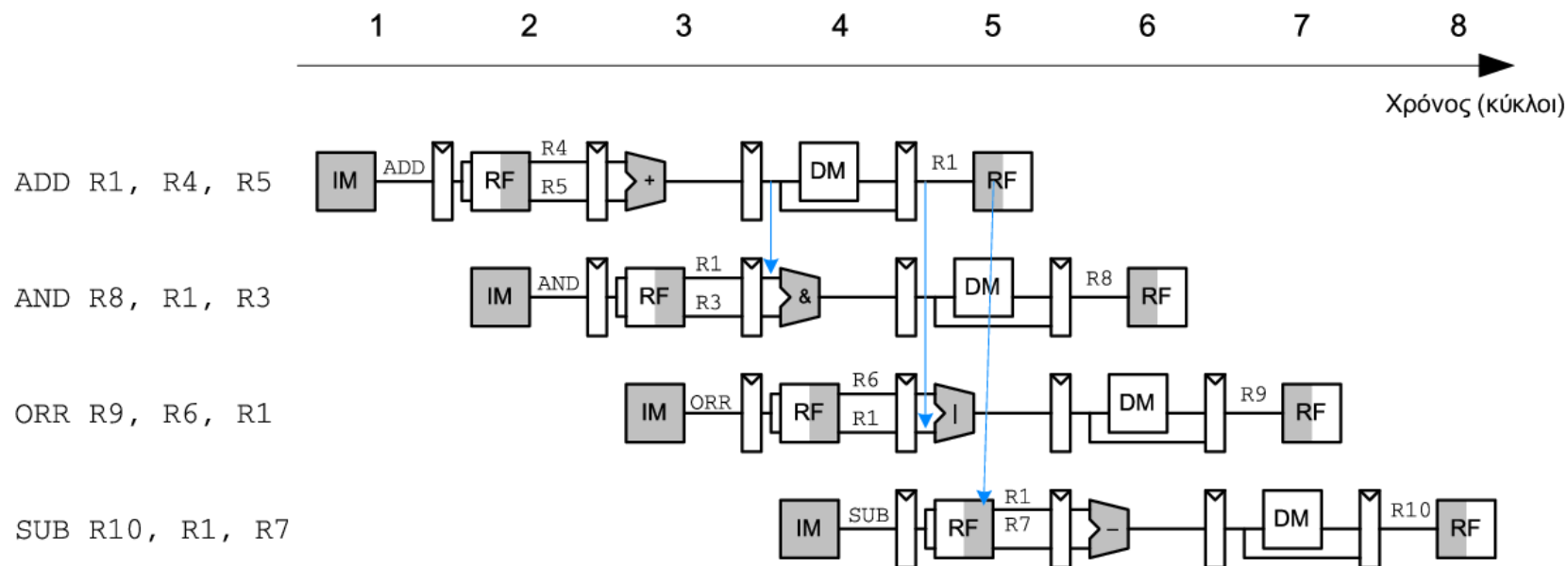


Εξάλειψη κινδύνων κατά τον χρόνο μεταγλώττισης

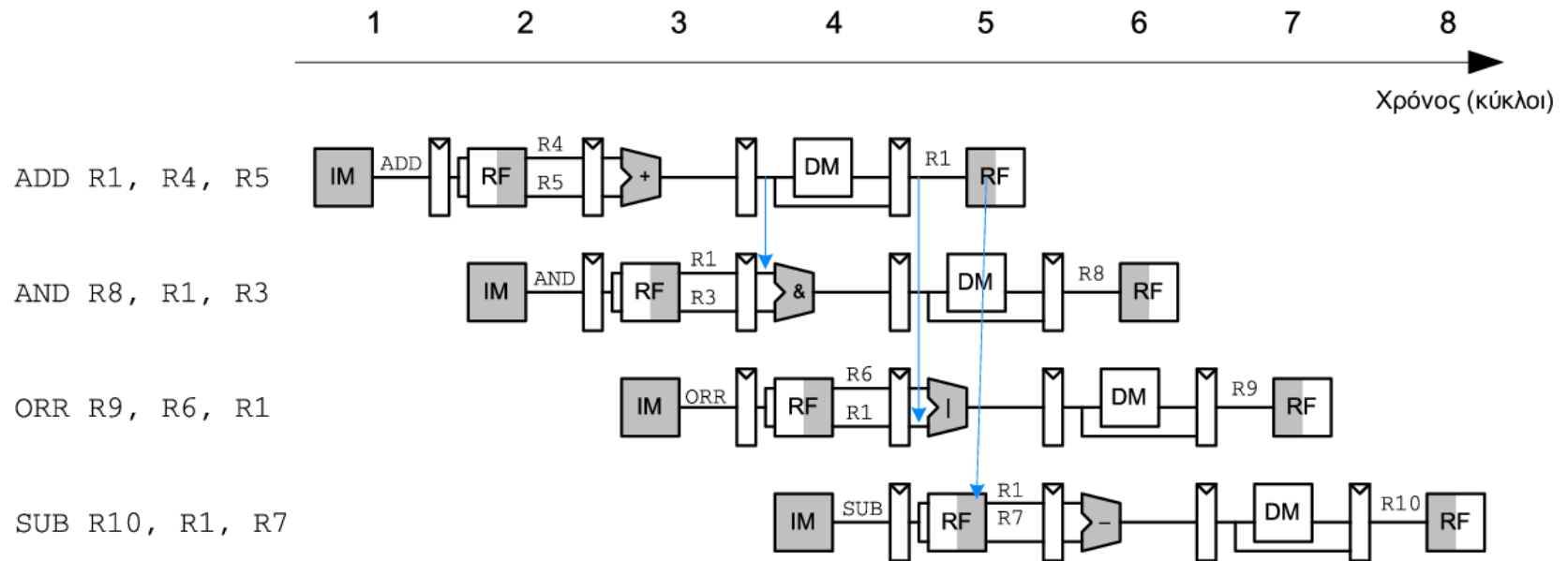
- Εισαγωγή αρκετών εντολών NOP, ώστε το αποτέλεσμα να προλάβει να είναι έτοιμο ή
- Μετακίνηση ανεξάρτητων χρήσιμων εντολών προς τα εμπρός



Πρώθηση δεδομένων



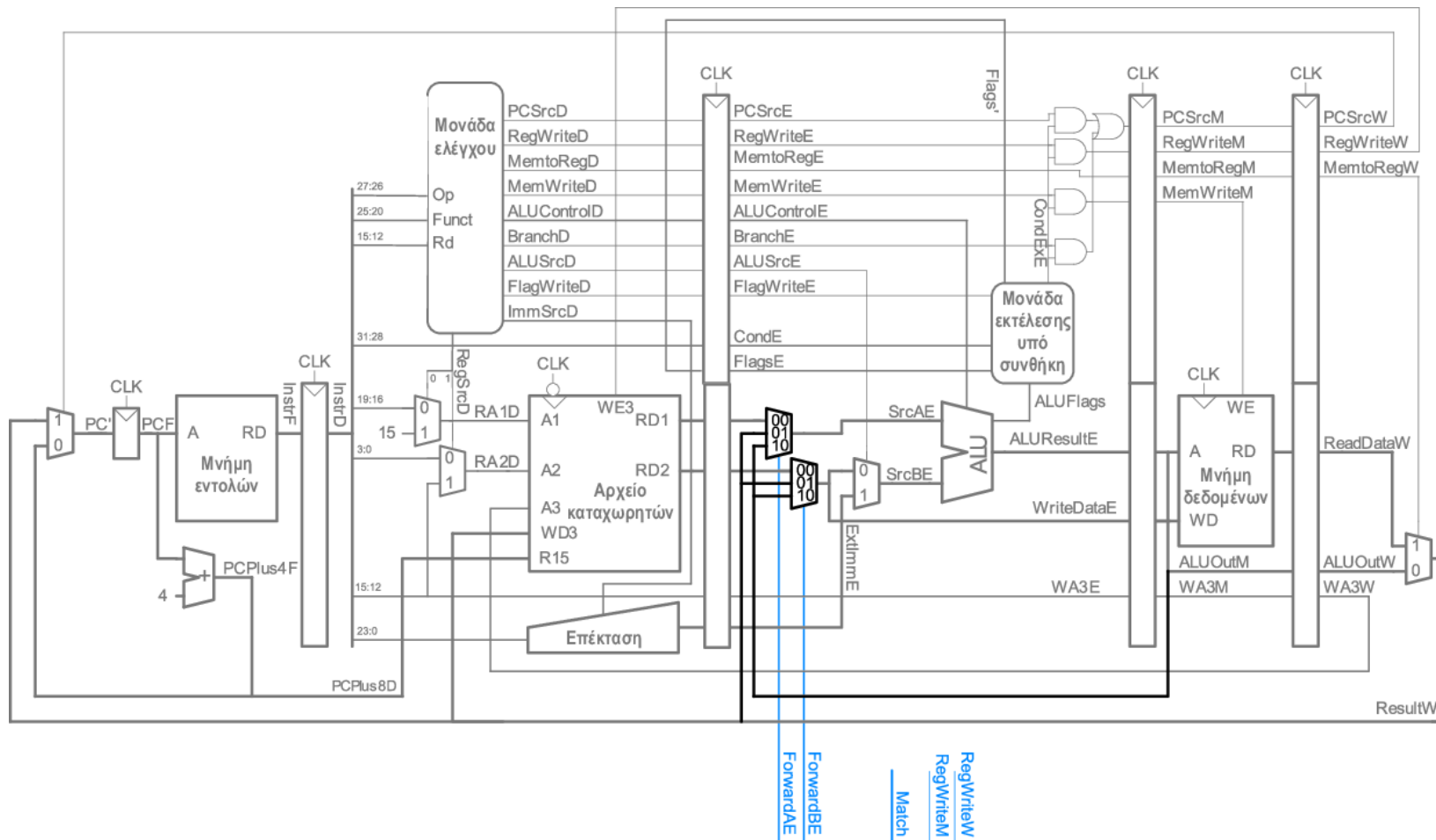
Πρωώθηση δεδομένων



- Ελέγχεται το αν ο καταχωρητής που διαβάζεται στο στάδιο Execute (εκτέλεση) ταιριάζει με τον καταχωρητή που εγγράφεται στο στάδιο Memory (μνήμη) ή Writeback (ετεροχρονισμένη εγγραφή)
- Αν ταιριάζει, το αποτέλεσμα προωθείται



Πρώθηση δεδομένων



Μονάδα κινδύνων



Πρώθηση δεδομένων

- Ταιριάζει ο καταχωρητής κατάστασης στο στάδιο **Execute** με τον καταχωρητή κατάστασης στο στάδιο **Memory**;
 $Match_1E_M = (RA1E == WA3M)$
 $Match_2E_M = (RA2E == WA3M)$
- Ταιριάζει ο καταχωρητής κατάστασης στο στάδιο **Execute** με τον καταχωρητή κατάστασης στο στάδιο **Writeback**;
 $Match_1E_W = (RA1E == WA3W)$
 $Match_2E_W = (RA2E == WA3W)$
- Αν ταιριάζει, πρώθηση του αποτελέσματος:

```
if      (Match_1E_M • RegWriteM)    ForwardAE = 10;  
else if (Match_1E_W • RegWriteW)    ForwardAE = 01;  
else                                  ForwardAE = 00;
```



Πρώθηση δεδομένων

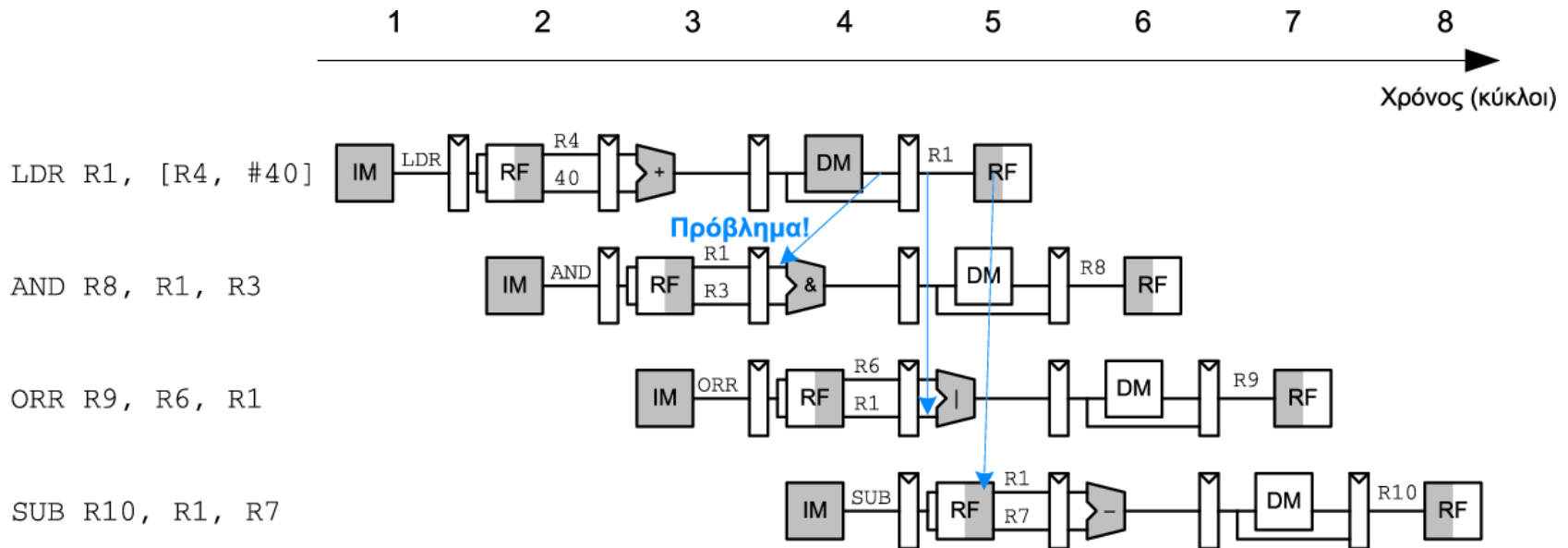
- Ταιριάζει ο καταχωρητής κατάστασης στο στάδιο **Execute** με τον καταχωρητή κατάστασης στο στάδιο **Memory**;
 $Match_1E_M = (RA1E == WA3M)$
 $Match_2E_M = (RA2E == WA3M)$
- Ταιριάζει ο καταχωρητής κατάστασης στο στάδιο **Execute** με τον καταχωρητή κατάστασης στο στάδιο **Writeback**;
 $Match_1E_W = (RA1E == WA3W)$
 $Match_2E_W = (RA2E == WA3W)$
- Αν ταιριάζει, πρώθηση του αποτελέσματος:

```
if      (Match_1E_M • RegWriteM)    ForwardAE = 10;  
else if (Match_1E_W • RegWriteW)    ForwardAE = 01;  
else                                  ForwardAE = 00;
```

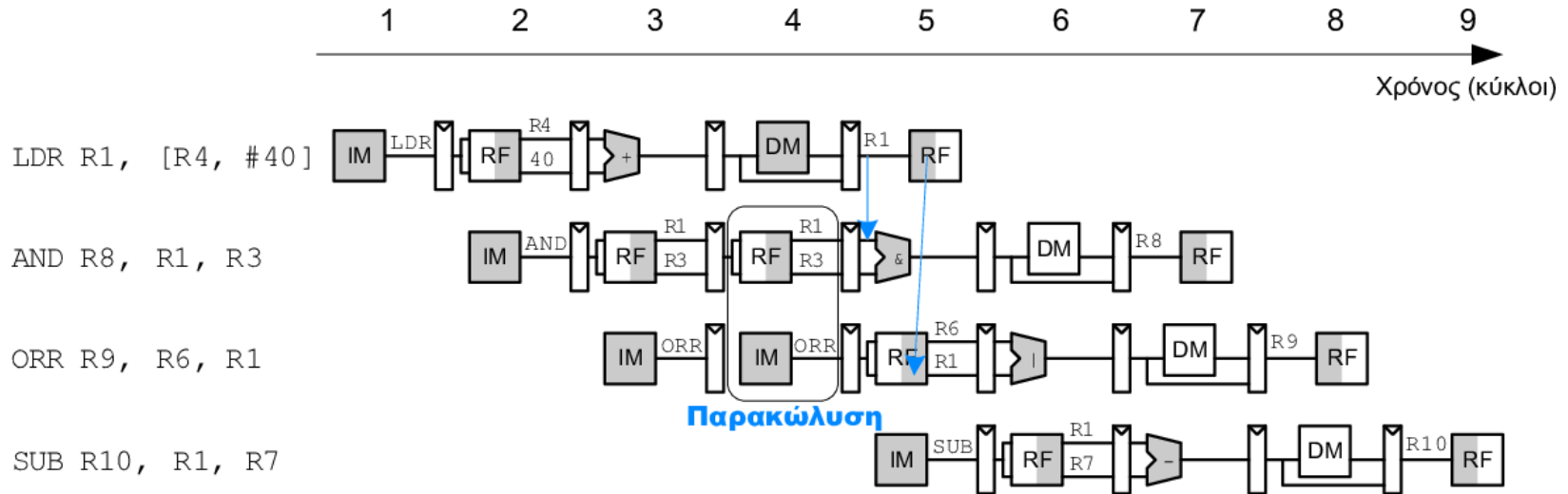
Το ίδιο για το ForwardBE αλλά με το Match2E



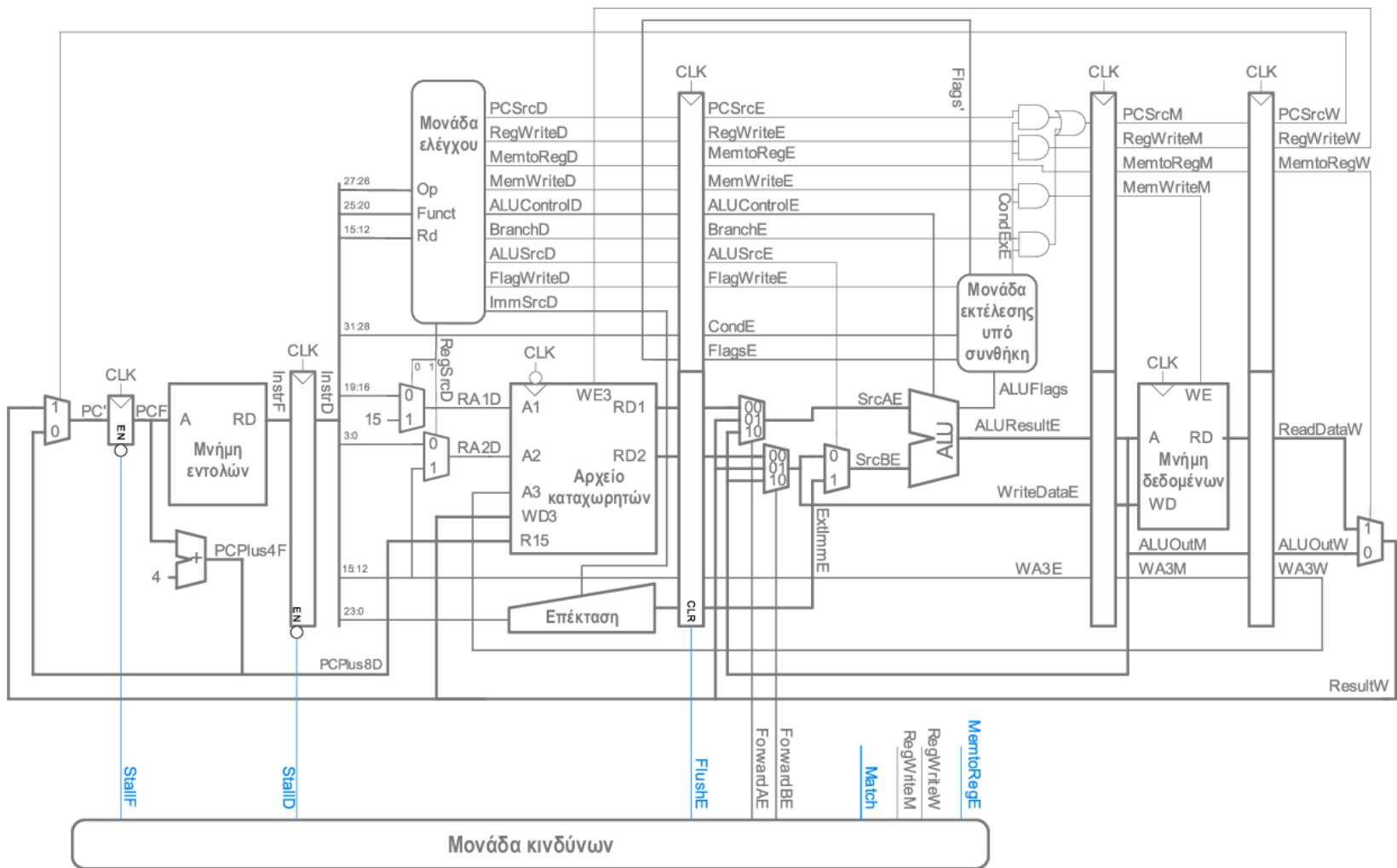
Παρακώλυση



Παρακώλυση



Υλικό παρακώλυσης



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <155>



Λογική παρακώλυσης

- Είναι οποιοσδήποτε από τους καταχωρητές προέλευσης στο στάδιο Decode (αποκωδικοποίηση) ίδιος με εκείνον που εγγράφεται στο στάδιο Execute (εκτέλεση);

$$Match_12D_E = (RA1D == WA3E) + (RA2D == WA3E)$$

- Είναι μια εντολή LDR στο στάδιο Execute ισοδύναμη με $AND\ Match_12D_E$;

$$Ildrstall = Match_12D_E \cdot MemtoRegE$$

$$StallF = StallD = FlushE = Ildrstall$$

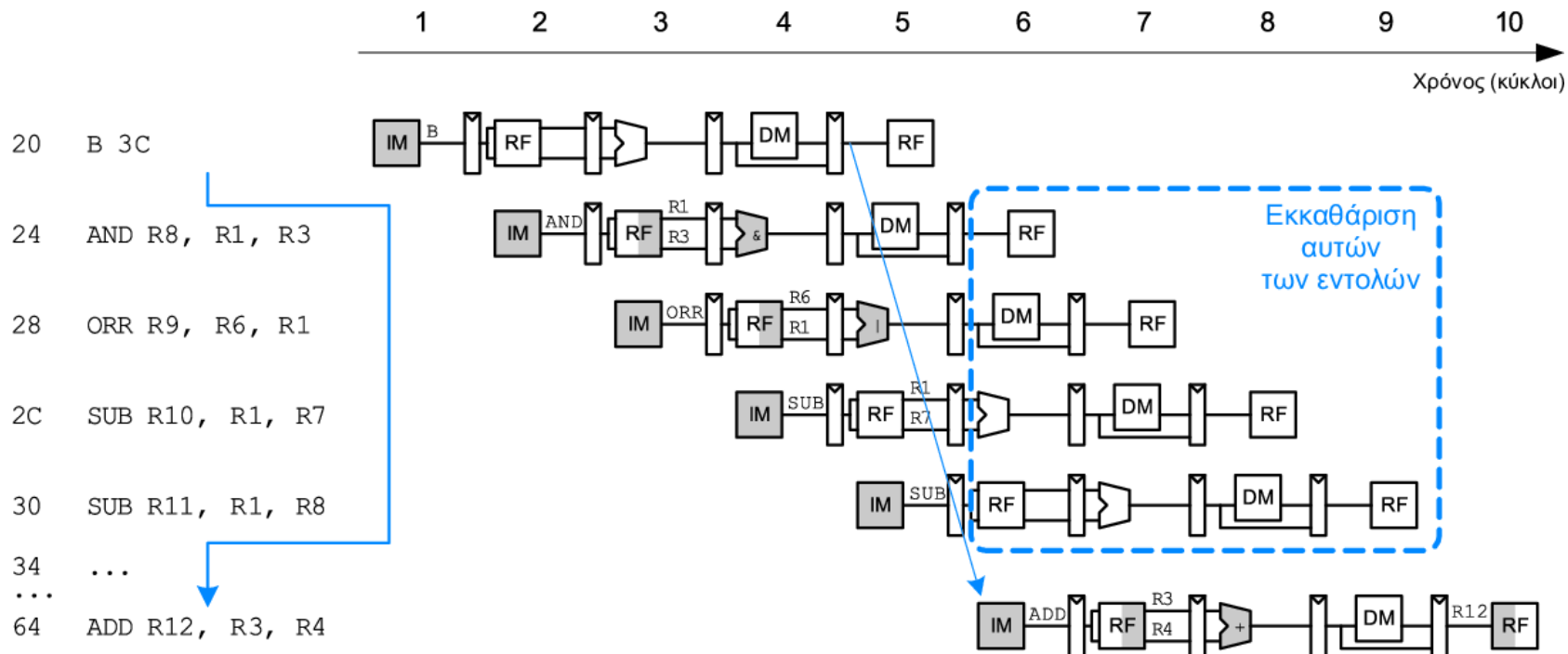


Κίνδυνοι ελέγχου

- **B:**
 - Η διακλάδωση δεν καθορίζεται μέχρι το στάδιο Writeback (ετεροχρονισμένη εγγραφή) της διοχέτευσης
 - Οι εντολές μετά τη διακλάδωση προσκομίζονται πριν λάβει χώρα η διακλάδωση
 - Αυτές οι 4 εντολές πρέπει να απορριφθούν αν λάβει χώρα η διακλάδωση
- **Ομοίως οι εγγραφές στον μετρητή PC (R15)**



Κίνδυνοι ελέγχου



Ποινή εσφαλμένης πρόβλεψης διακλάδωσης

- Πλήθος των εντολών που απορρίπτονται όταν ακολουθείται η διακλάδωση (4)
- Μπορεί να μειωθεί αν προσδιοριστεί νωρίτερα η διεύθυνση ΒΤΑ

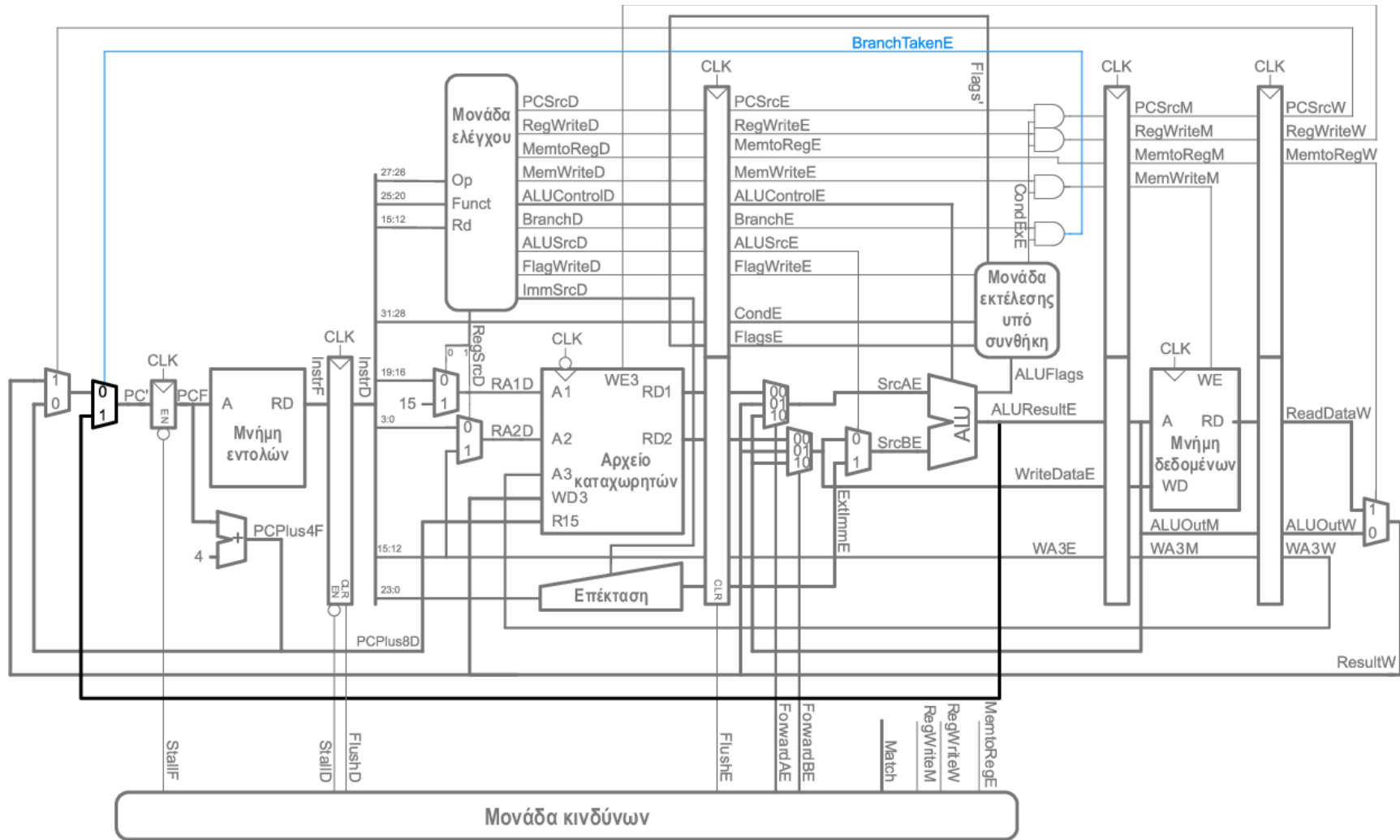


Έγκαιρη λήψη απόφασης για διακλάδωση

- Προσδιορισμός της διεύθυνσης BTA στο στάδιο **Execute**
 - Ποινή εσφαλμένης πρόβλεψης διακλάδωσης = 2 κύκλοι
- **Μεταβολές στο υλικό**
 - Πριν από τον καταχωρητή *PC* προστίθεται ένας πολυπλέκτης διακλάδωσης ο οποίος επιλέγει τξ διεύθυνση BTA από το *ALUResultE*
 - Προστίθεται το σήμα επιλογής *BranchTakenE* γι' αυτόν τον πολυπλέκτη (το οποίο ενεργοποιείται μόνο αν ικανοποιείται η συνθήκη διακλάδωσης)
 - Το *PCSrcW* ενεργοποιείται πλέον μόνο για εγγραφές στον καταχωρητή *PC*



Επεξεργαστής με διοχέτευση και έγκαιρη λήψη απόφασης για διακλάδωση



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

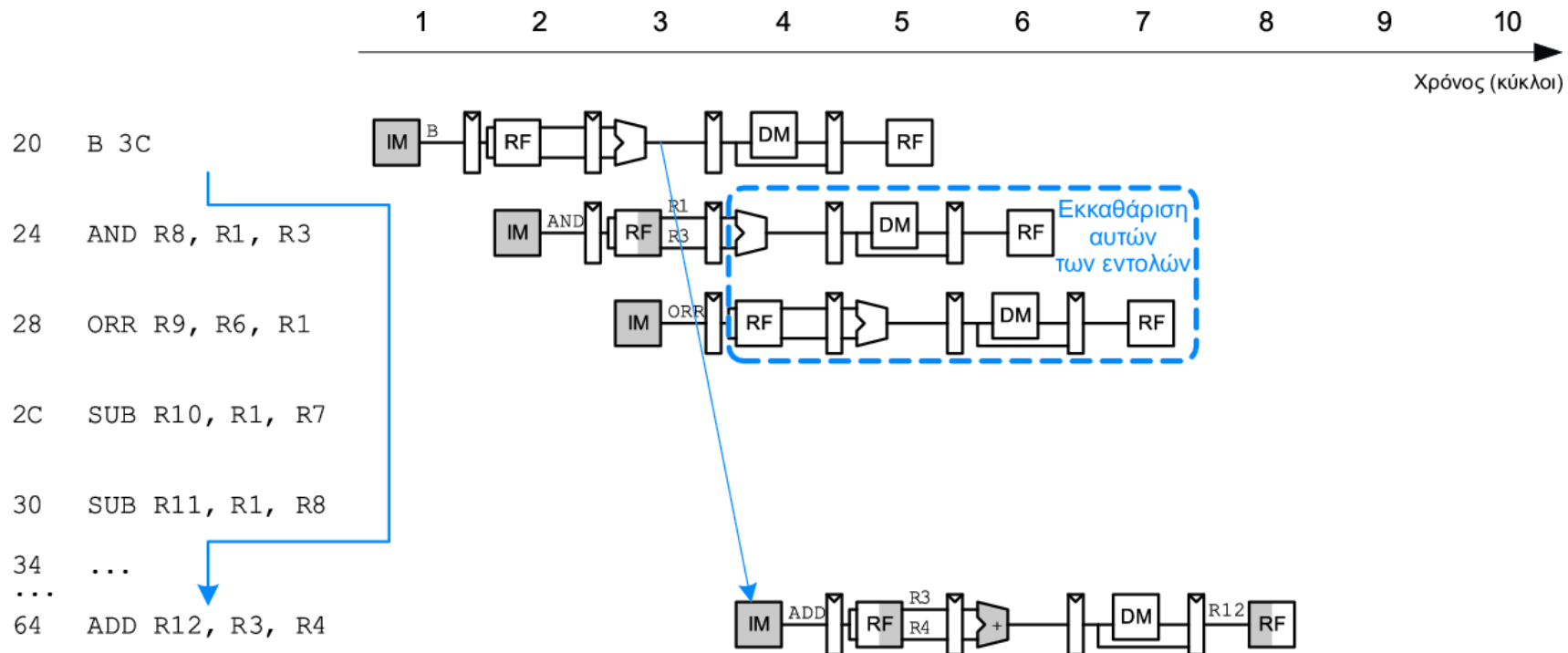
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτότυπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <160>



Κίνδυνοι ελέγχου με έγκαιρη λήψη απόφασης για διακλάδωση



Λογική παρακώλυσης ελέγχου

- **$PCWrPendingF = 1$** αν γίνεται εγγραφή στον καταχωρητή PC στο στάδιο Decode, Execute ή Memory

$$PCWrPendingF = PCSrcD + PCSrcE + PCSrcM$$

- **Παρακώλυση της προσκόμισης (Fetch)** αν το $PCWrPendingF$ έχει την τιμή 1

$$StallF = IdrStallD + PCWrPendingF$$

- **Εκκαθάριση της αποκωδικοποίησης (Decode)** αν το $PCWrPendingF$ έχει την τιμή 1 OR ο μετρητής PC εγγράφεται στο στάδιο Writeback OR ακολουθείται η διακλάδωση

$$FlushD = PCWrPendingF + PCSrcW + BranchTakenE$$

- **Εκκαθάριση της εκτέλεσης (Execute)** αν ακολουθείται η διακλάδωση

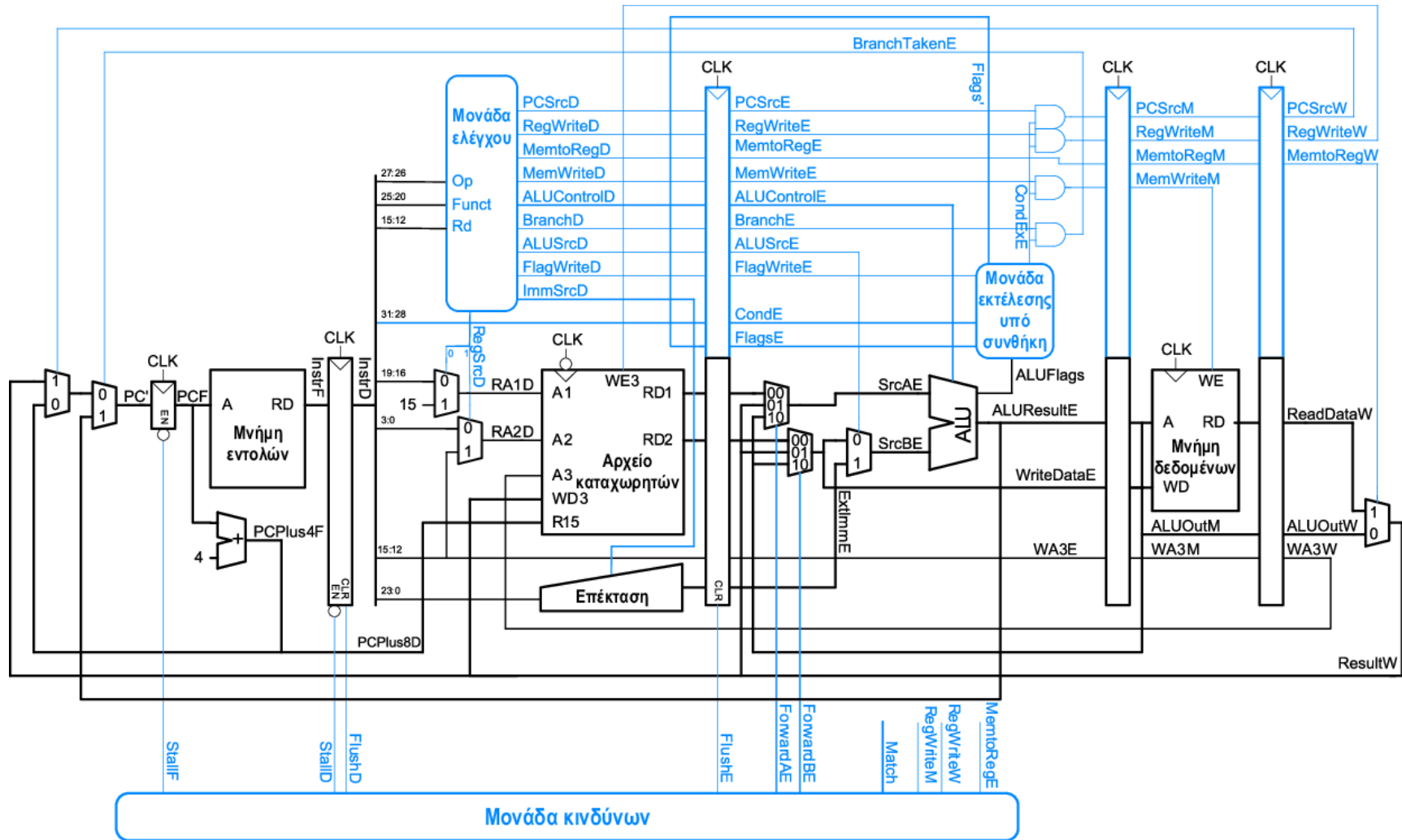
$$FlushE = IdrStallD + BranchTakenE$$

- **Παρακώλυση της αποκωδικοποίησης (Decode)** αν το $IdrStallD$ έχει την τιμή 1 (όπως προηγουμένως)

$$StallD = IdrStallD$$



Επεξεργαστής ARM με διοχέτευση και μονάδα κινδύνων



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτότυπο: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <163>



Ανάλυση επιδόσεων για επεξεργαστή με διοχέτευση: Παράδειγμα

- **Μετροπρόγραμμα SPECINT2000:**
 - 25% για τις εντολές φόρτωσης
 - 10% για τις εντολές αποθήκευσης
 - 13% για τις εντολές διακλάδωσης
 - 52% για τις εντολές επεξεργασίας δεδομένων
- **Έστω ότι:**
 - Το 40% των εντολών φόρτωσης χρησιμοποιούνται από την επόμενη εντολή
 - Το 50% των διακλαδώσεων προβλέπεται εσφαλμένα
- **Ποια είναι η τιμή του μέσου δείκτη CPI;**



Ανάλυση επιδόσεων για επεξεργαστή με διοχέτευση: Παράδειγμα

- **Μετροπρόγραμμα SPECINT2000:**

- 25% για τις εντολές φόρτωσης
- 10% για τις εντολές αποθήκευσης
- 13% για τις εντολές διακλάδωσης
- 52% για τις εντολές επεξεργασίας δεδομένων

- **Έστω ότι:**

- Το 40% των εντολών φόρτωσης χρησιμοποιούνται από την επόμενη εντολή
- Το 50% των διακλαδώσεων προβλέπεται εσφαλμένα

- **Ποια είναι η τιμή του μέσου δείκτη CPI;**

- Φόρτωση: CPI = 1 όταν δεν γίνεται παρακώλυση, 2 όταν γίνεται
Άρα, $CPI_{lw} = 1(0.6) + 2(0.4) = 1.4$
- Διακλάδωση: CPI = 1 όταν δεν γίνεται παρακώλυση, 3 όταν γίνεται
Άρα, $CPI_{beq} = 1(0.5) + 3(0.5) = 2$

$$\text{Μέσος δείκτης CPI} = (0.25)(1.4) + (0.1)(1) + (0.13)(2) + (0.52)(1) = \mathbf{1.23}$$



Ανάλυση επιδόσεων για επεξεργαστή με διοχέτευση

- Κρίσιμη διαδρομή του επεξεργαστή με διοχέτευση:

$$T_{c3} = \max [$$

$$t_{pcq} + t_{mem} + t_{setup}$$

$$2(t_{RFread} + t_{setup})$$

$$t_{pcq} + 2t_{mux} + t_{ALU} + t_{setup}$$

$$t_{pcq} + t_{mem} + t_{setup}$$

$$2(t_{pcq} + t_{mux} + t_{RFwrite})]$$

Fetch

Decode

Execute

Memory

Writeback



Ανάλυση επιδόσεων για επεξεργαστή με διοχέτευση: Παράδειγμα

Στοιχείο	Παράμετρος	Καθυστέρηση (ps)
Καθυστέρηση από το ρολόι έως την έξοδο Q (καταχωρητής)	t_{pcq_PC}	40
Σταθεροποίηση καταχωρητή	t_{setup}	50
Πολυπλέκτης	t_{mux}	25
Μονάδα ALU	t_{ALU}	120
Αποκωδικοποιητής	t_{dec}	70
Ανάγνωση από τη μνήμη	t_{mem}	200
Ανάγνωση από το αρχείο καταχωρητών	t_{RFread}	100
Σταθεροποίηση αρχείου καταχωρητών	$t_{RFsetup}$	60
Εγγραφή στο αρχείο καταχωρητών	$t_{RFwrite}$	70

Χρόνος κύκλου: $T_{c3} = ;$



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <167>

Ανάλυση επιδόσεων για επεξεργαστή με διοχέτευση: Παράδειγμα

Στοιχείο	Παράμετρος	Καθυστέρηση (ps)
Καθυστέρηση από το ρολόι έως την έξοδο Q (καταχωρητής)	t_{pcq_PC}	40
Σταθεροποίηση καταχωρητή	t_{setup}	50
Πολυπλέκτης	t_{mux}	25
Μονάδα ALU	t_{ALU}	120
Αποκωδικοποιητής	t_{dec}	70
Ανάγνωση από τη μνήμη	t_{mem}	200
Ανάγνωση από το αρχείο καταχωρητών	t_{RFread}	100
Σταθεροποίηση αρχείου καταχωρητών	$t_{RFsetup}$	60
Εγγραφή στο αρχείο καταχωρητών	$t_{RFwrite}$	70

Χρόνος κύκλου: $T_{c3} = 2(t_{RFread} + t_{setup}) = 2[100 + 50] \text{ ps} = 300 \text{ ps}$



Ανάλυση επιδόσεων για επεξεργαστή με διοχέτευση: Παράδειγμα

Πρόγραμμα με 100 δισεκατομμύρια εντολές

$$\begin{aligned}\text{Χρόνος εκτέλεσης} &= (\text{πλήθος εντολών}) \times \text{CPI} \times T_c \\ &= (100 \times 10^9)(1.23)(300 \times 10^{-12}) \\ &= \mathbf{36.9 \text{ seconds}}\end{aligned}$$



Επιδόσεις επεξεργαστών: Σύγκριση

Επεξεργαστής	Χρόνος εκτέλεσης (δευτ/πτα)	Αύξηση ταχύτητας (βάση αναφοράς: επεξεργαστής ενός κύκλου)
Ενός κύκλου	84	1
Πολλών κύκλων	140	0.6
Με διοχέτευση	36.9	2.28



Προηγμένη μικροαρχιτεκτονική

- Διοχετεύσεις μεγάλου βάθους
- Μικρολειτουργίες
- Πρόβλεψη διακλαδώσεων
- Υπερβαθμωτοί επεξεργαστές
- Μετονομασία καταχωρητών
- SIMD
- Πολυνημάτωση
- Πολυεπεξεργαστές



Διοχετεύσεις μεγάλου βάθους

- Συνήθως 10-20 στάδια
- Το μέγιστο πλήθος των σταδίων περιορίζεται από:
 - Τους κινδύνους της διοχέτευσης
 - Την επιβάρυνση δημιουργίας ακολουθίας
 - Την ισχύ
 - Το κόστος



Μικρολειτουργίες

- Ανάλυση πιο σύνθετων εντολών σε μια σειρά από απλές εντολές που ονομάζονται *μικρολειτουργίες* (micro-operations, ή *micro-ops* ή *μ-ops*)
- Κατά τον χρόνο εκτέλεσης οι σύνθετες εντολές αποκωδικοποιούνται σε μία ή περισσότερες μικρολειτουργίες
- Χρησιμοποιούνται πολύ στις αρχιτεκτονικές CISC (complex instruction set computer, υπολογιστές με σύνθετο σύνολο εντολών), π.χ. η αρχιτεκτονική x86
- Χρησιμοποιούνται για κάποιες εντολές της αρχιτεκτονικής ARM. Για παράδειγμα:

Σύνθετη πράξη

```
LDR R1, [R2], #4
```

Ακολουθία μικρολειτουργιών

```
LDR R1, [R2]
```

```
ADD R2, R2, #4
```

Χωρίς μ-ops θα έπρεπε να υπάρχει 2η θύρα εγγραφής στο αρχείο καταχωρητών



Μικρολειτουργίες

- Επιτρέπουν πυκνό κώδικα (λιγότερες προσπελάσεις μνήμης)...
- ...αλλά διατηρούν την απλότητα του υλικού RISC
- Η αρχιτεκτονική ARM επιτυγχάνει μια ισορροπία επιλέγοντας εντολές που:
 - Δίνουν καλύτερη πυκνότητα κώδικα από ό,τι τα σύνολα εντολών των αμιγών αρχιτεκτονικών RISC (όπως η MIPS)
 - Επιτρέπει πιο αποδοτική αποκωδικοποίηση από ό,τι τα σύνολα εντολών των αρχιτεκτονικών CISC (όπως η x86)



Πρόβλεψη διακλαδώσεων

- Οι επεξεργαστές μαντεύουν αν η διακλάδωση πρέπει να ακολουθηθεί
 - Συνήθως ακολουθούν τις διακλαδώσεις προς τα πίσω (βρόχοι)
 - Λαμβάνουν υπόψη το ιστορικό για να βελτιώσουν την πρόβλεψη
- Η καλή πρόβλεψη μειώνει το ποσοστό των διακλαδώσεων που απαιτούν εκκαθάριση



Πρόβλεψη διακλαδώσεων

- Ιδανικός επεξεργαστής με διοχέτευση: $CPI = 1$
- Η εσφαλμένη πρόβλεψη διακλαδώσεων αυξάνει τον δείκτη CPI
- **Στατική πρόβλεψη διακλαδώσεων:**
 - Ελέγχει την κατεύθυνση της διακλάδωσης (προς τα εμπρός ή προς τα πίσω)
 - Αν είναι προς τα πίσω, ακολουθείται η πρόβλεψη
 - Διαφορετικά, δεν ακολουθείται η πρόβλεψη
- **Δυναμική πρόβλεψη διακλαδώσεων:**
 - Διατηρούν το ιστορικό των τελευταίων πολλών εκατοντάδων (ή χιλιάδων) διακλαδώσεων στην περιοχή προσωρινής αποθήκευσης στόχου διακλάδωσης (branch target buffer), όπου καταγράφονται:
 - Ο προορισμός της διακλάδωσης
 - Το αν η διακλάδωση έχει ακολουθηθεί



Πρόβλεψη διακλαδώσεων: Παράδειγμα

```
MOV R1, #0           ; R1 = sum
MOV R0, #0           ; R0 = i

FOR                   ; for (i=0; i<10; i=i+1)
  CMP R0, #10
  BGE DONE
  ADD R1, R1, R0      ; sum = sum + i
  ADD R0, R0, #1
  B FOR

DONE
```

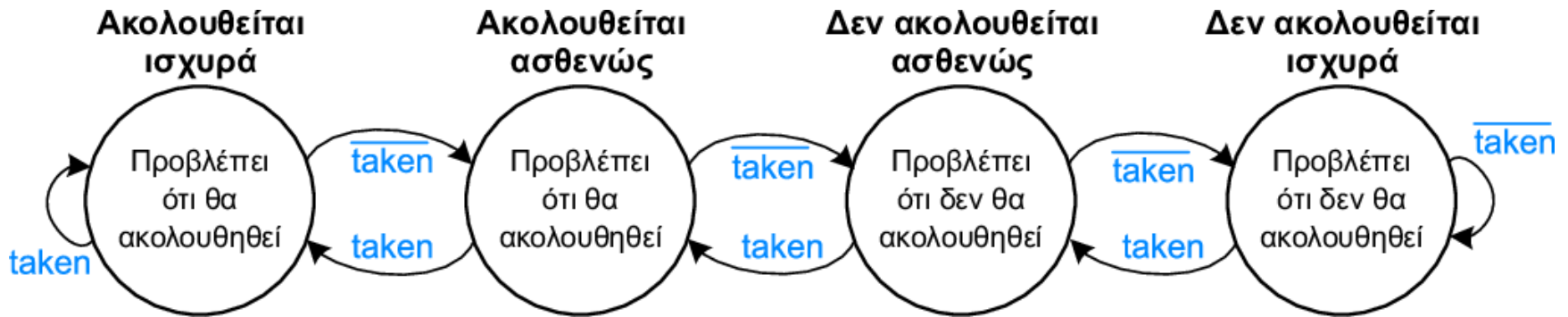


Προγνώστης διακλαδώσεων μεγέθους 1 bit

- Θυμάται αν η διακλάδωση ακολουθήθηκε την τελευταία φορά και προβλέπει ότι θα συμβεί το ίδιο πράγμα και την επόμενη φορά
- Προβλέπει εσφαλμένα την πρώτη και την τελευταία διακλάδωση ενός βρόχου



Προγνώστης διακλαδώσεων μεγέθους 2 bit

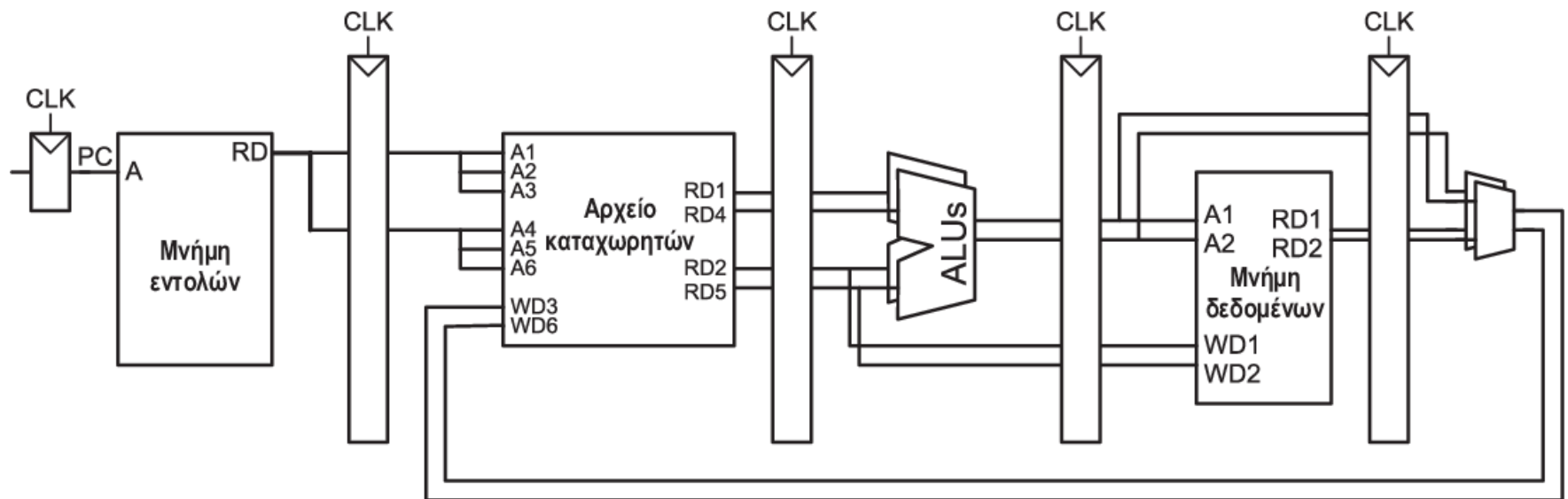


Προβλέπει εσφαλμένα μόνο την τελευταία διακλάδωση ενός βρόχου



Υπερβαθμωτός επεξεργαστής

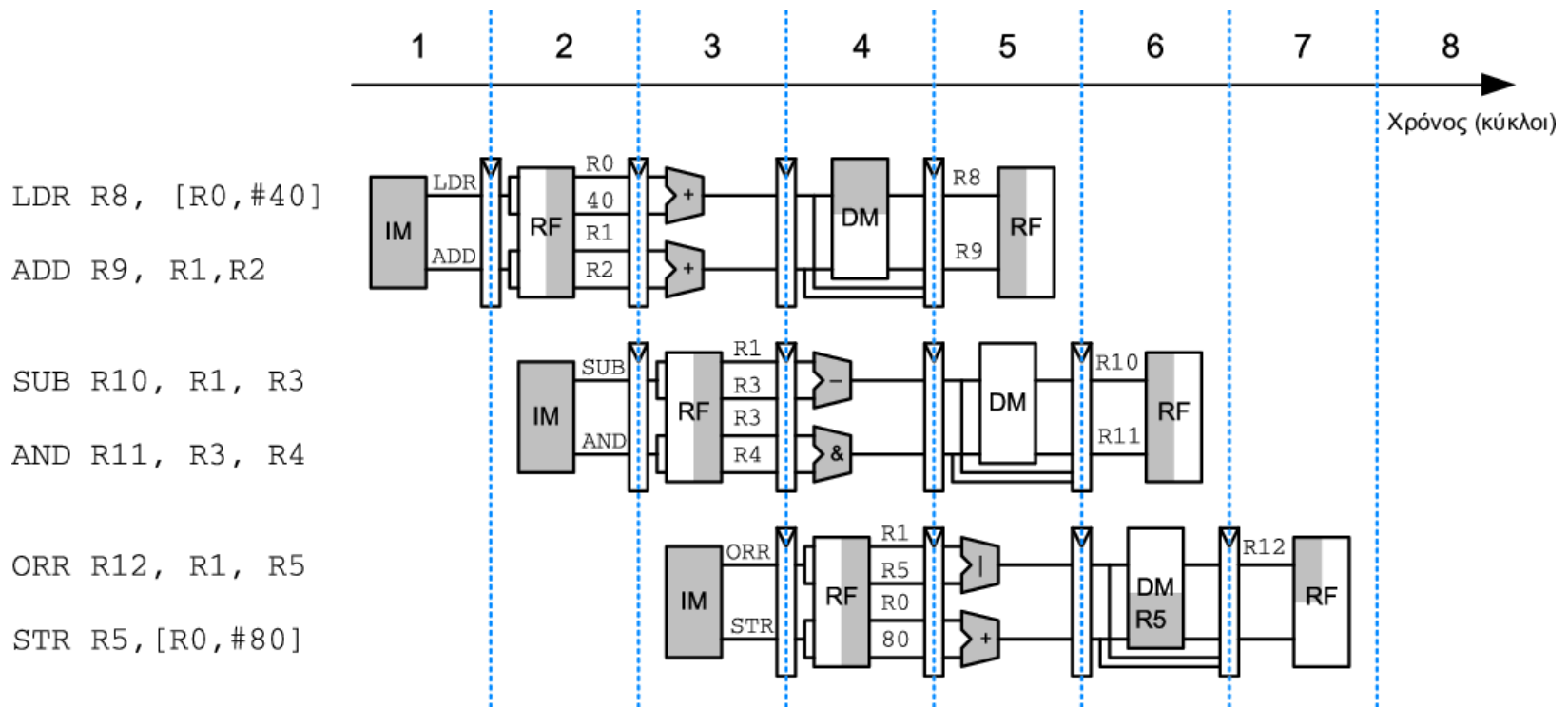
- Πολλά αντίγραφα της διαδρομής δεδομένων εκτελούν πολλές εντολές ταυτόχρονα
- Οι εξαρτήσεις δυσκολεύουν την έκδοση πολλών εντολών ταυτόχρονα



Υπερβαθμωτός επεξεργαστής: Παράδειγμα

Ιδανικός δείκτης IPC: 2

Πραγματικός δείκτης IPC: 2



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

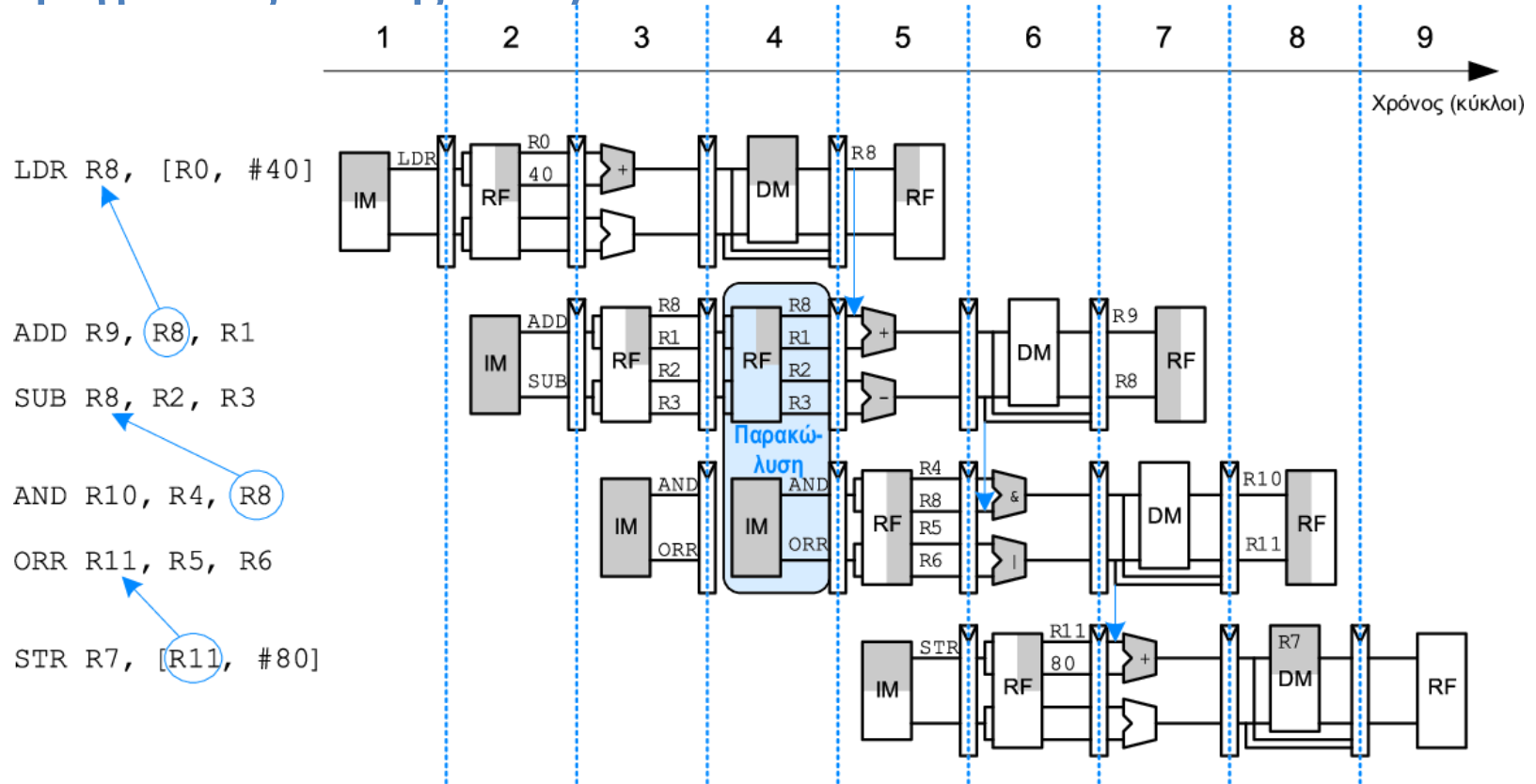
Κεφάλαιο 7 <181>



Υπερβαθμωτός επεξεργαστής με εξαρτήσεις

Ιδανικός δείκτης IPC: 2

Πραγματικός δείκτης IPC: $6/5 = 1.2$



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <182>



Επεξεργαστής εκτός σειράς

- «Κοιτάζει μπροστά» εξετάζοντας πολλές εντολές
- Εκδίδει όσο τον δυνατόν περισσότερες εντολές ταυτόχρονα
- Εκδίδει εντολές εκτός σειράς (με την προϋπόθεση ότι δεν παραβιάζονται εξαρτήσεις)
- **Εξαρτήσεις:**
 - **RAW** (read after write, ανάγνωση μετά από εγγραφή): Η μία εντολή εγγράφει δεδομένα σε έναν καταχωρητή, από τον οποίο διαβάζει δεδομένα μια επόμενη εντολή
 - **WAR** (write after read, εγγραφή μετά από ανάγνωση): Η μία εντολή διαβάζει δεδομένα από έναν καταχωρητή, στον οποίο εγγράφει δεδομένα μια επόμενη εντολή
 - **WAW** (write after write, εγγραφή μετά από εγγραφή): Η μία εντολή εγγράφει δεδομένα σε έναν καταχωρητή, στον οποίο επίσης εγγράφει δεδομένα μια επόμενη εντολή



Επεξεργαστής εκτός σειράς

- **Παραλληλισμός στο επίπεδο εντολών (ILP):**
Το πλήθος των εντολών που μπορούν να εκδοθούν ταυτόχρονα (μέσος όρος < 3)
- **Πίνακας αποτελεσμάτων (scoreboard):**
Πίνακας στον οποίο καταγράφονται:
 - Οι εντολές που αναμένουν έκδοση
 - Οι διαθέσιμες λειτουργικές μονάδες
 - Οι εξαρτήσεις

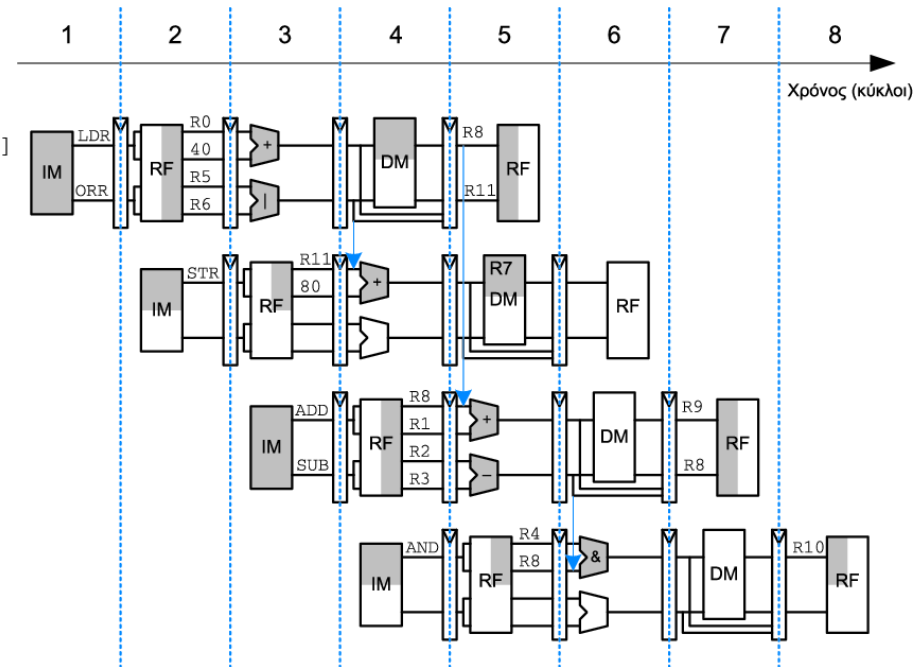
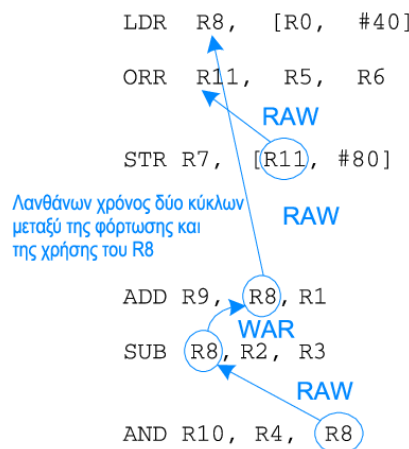


Επεξεργαστής εκτός σειράς: Παράδειγμα

```
LDR R8, [R0, #40]
ADD R9, R8, R1
SUB R8, R2, R3
AND R10, R4, R8
ORR R11, R5, R6
STR R7, [R11, #80]
```

Ιδανικός δείκτης IPC: 2

Πραγματικός δείκτης IPC: $6/4 = 1.5$



Μετονομασία καταχωρητών

```
LDR R8, [R0, #40]
ADD R9, R8, R1
SUB R8, R2, R3
AND R10, R4, R8
ORR R11, R5, R6
STR R7, [R11, #80]
```

Ιδανικός δείκτης IPC: 2

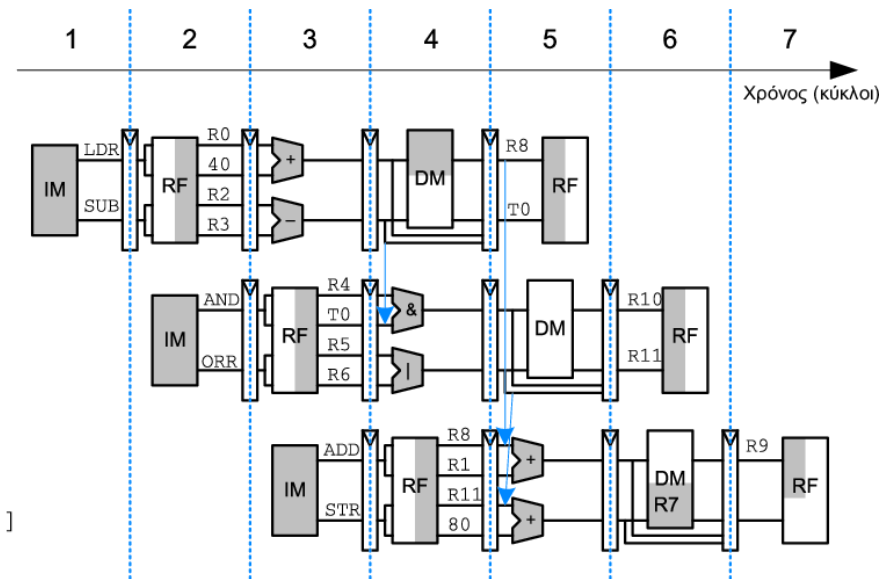
Πραγματικός δείκτης IPC: $6/3 = 2$

LDR R8, [R0, #40]
SUB T0, R2, R3
AND R10, R4, T0
ORR R11, R5, R6
ADD R9, R8, R1
STR R7, [R11, #80]

RAW δύο κύκλων

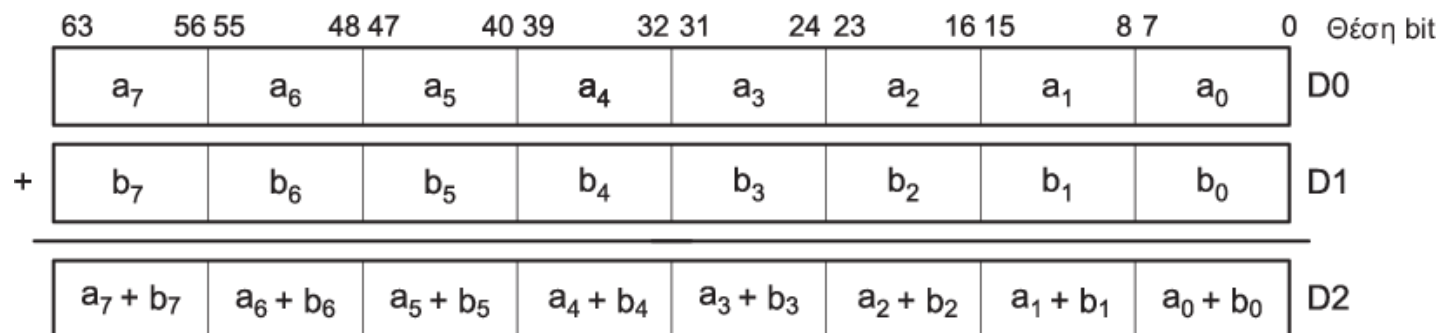
RAW

RAW



SIMD

- Επεξεργασία μίας εντολής πολλών δεδομένων (single Instruction Multiple Data, SIMD)
 - Μία εντολή χειρίζεται πολλά δεδομένα παράλληλα
 - Συνήθης εφαρμογή: επεξεργασία γραφικών
 - Εκτέλεση σύντομων αριθμητικών πράξεων (επίσης γνωστή ως *συνεπτυγμένη* (packed) αριθμητική)
- Για παράδειγμα, πρόσθεση οκτώ στοιχείων των 8 bit



Τεχνικές προηγμένης αρχιτεκτονικής

- **Πολυνημάτωση**

- Πρόγραμμα επεξεργασίας κειμένου: νήματα για πληκτρολόγηση, ορθογραφικό έλεγχο, εκτύπωση

- **Πολυπεξεργαστές**

- Πολλοί επεξεργαστές (πυρήνες) στο ίδιο τσιπ



Νημάτωση: Ορισμοί

- **Διεργασία:** πρόγραμμα που εκτελείται σε έναν υπολογιστή
 - Πολλές διεργασίες μπορούν να εκτελούνται ταυτόχρονα: π.χ. περιήγηση στον Ιστό, αναπαραγωγή μουσικής, συγγραφή ενός άρθρου
- **Νήμα:** μέρος ενός προγράμματος
 - Κάθε διεργασία έχει πολλά νήματα: π.χ. ένα πρόγραμμα επεξεργασίας κειμένου μπορεί να έχει νήματα για πληκτρολόγηση, ορθογραφικό έλεγχο, εκτύπωση



Νήματα σε συμβατικούς επεξεργαστές

- Ένα νήμα εκτελείται ανά πάσα στιγμή
- Όταν ένα νήμα παρακωλυθεί (για παράδειγμα, περιμένοντας δεδομένα από τη μνήμη):
 - Αποθηκεύεται η αρχιτεκτονική κατάστασή του
 - Φορτώνεται στον επεξεργαστή η αρχιτεκτονική κατάσταση του νήματος που περιμένει και το νήμα εκτελείται
 - Η διαδικασία αυτή ονομάζεται **θεματική εναλλαγή** (context switching)
- Ο χρήστης έχει την ψευδαίσθηση ότι όλα τα νήματα εκτελούνται ταυτόχρονα



Πολυνημάτωση

- Περισσότερα από ένα αντίγραφα της αρχιτεκτονικής κατάστασης
- Περισσότερα από ένα νήματα είναι **ενεργά** ταυτόχρονα:
 - Όταν ένα νήμα παρακωλυθεί, εκτελείται αμέσως ένα άλλο νήμα
 - Αν ένα νήμα δεν μπορεί να κρατήσει όλες τις μονάδες εκτέλεσης απασχολημένες, τότε κάποιο άλλο νήμα μπορεί να τις χρησιμοποιήσει
- Δεν αυξάνει τον παραλληλισμό στο επίπεδο εντολών (ILP) ενός μεμονωμένου νήματος, αλλά αυξάνει τη διεκπεραιωτική ικανότητά του

Η Intel χρησιμοποιεί την ονομασία «υπερνημάτωση»



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 7 <191>

Πολυεπεξεργαστές

- Αποτελούνται από πολλούς επεξεργαστές (πυρήνες) και μια μέθοδο επικοινωνίας μεταξύ αυτών
- Τύποι:
 - **Συμμετρικοί (ή ομογενείς):** πολλοί πυρήνες που χρησιμοποιούν από κοινού μία κύρια μνήμη
 - **Ετερογενείς:** ξεχωριστοί πυρήνες για διαφορετικές εργασίες (για παράδειγμα, οι επεξεργαστές DSP και CPU στα κινητά τηλέφωνα)
 - **Συστοιχίες (clusters):** Κάθε πυρήνας διαθέτει το δικό του σύστημα μνήμης



Πρόσθετοι πόροι

- Patterson & Hennessy's: *Computer Architecture: A Quantitative Approach*
- Συνέδρια:
 - www.cs.wisc.edu/~arch/www/
 - ISCA (International Symposium on Computer Architecture)
 - HPCA (International Symposium on High Performance Computer Architecture)

