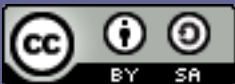


Αριθμητικές- λογικές πράξεις με καταχωρητές

(παραδείγματα στις αρχιτεκτονικές RISC-V και x86-64)

<https://mixstef.github.io/courses/comparch/>

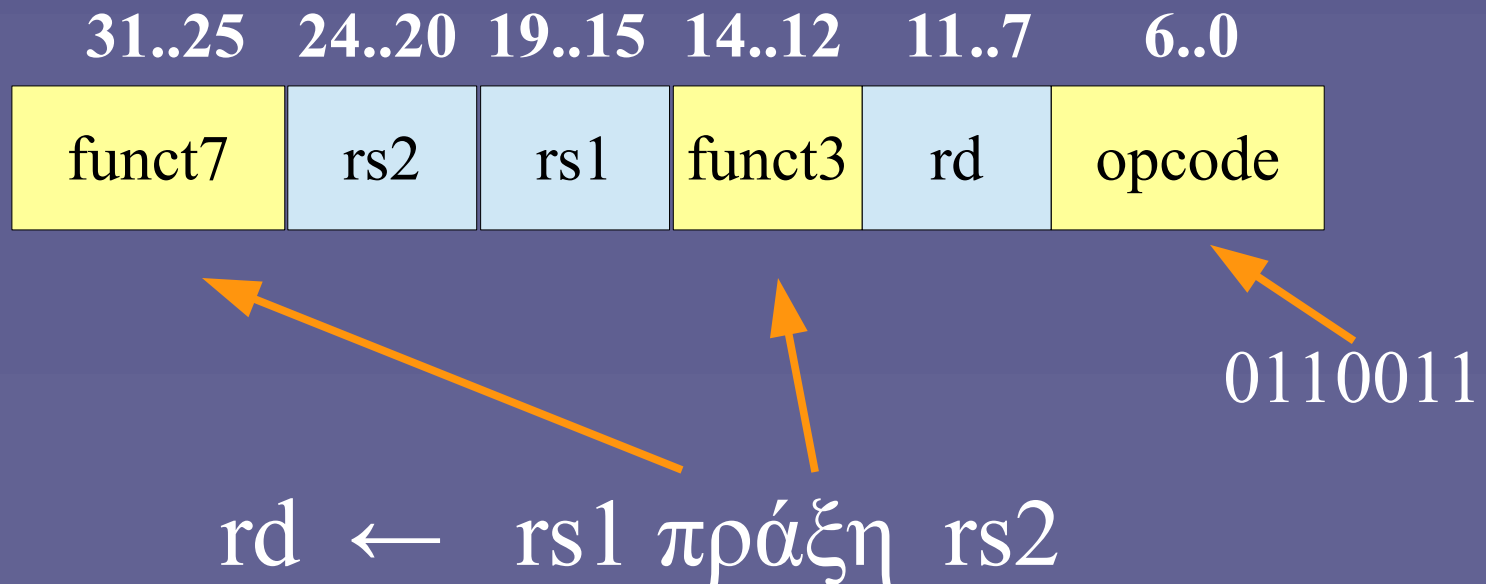
Μ.Στεφανιδάκης



Αριθμητικές-λογικές πράξεις σε RISC-V

- Μόνο μεταξύ καταχωρητών ($rd \leftarrow rs1 \text{ op } rs2$)
 - Κλασική RISC (load – store) αρχιτεκτονική
 - 32 καταχωρητές (**x0** έως **x31**) και επιπλέον ο program counter (**pc**)
 - Με εύρος 32 bits ο καθένας (RV32I ISA)
 - Ή 64 bits στην εκδοχή RV64I
- Μια σταθερά μπορεί να χρησιμοποιηθεί ως πηγή (**immediate addressing**)
 - Σταθερές τιμές που περιλαμβάνονται αυτούσιες μέσα στα bits της εντολής

Πράξεις μεταξύ καταχωρητών (R-type)

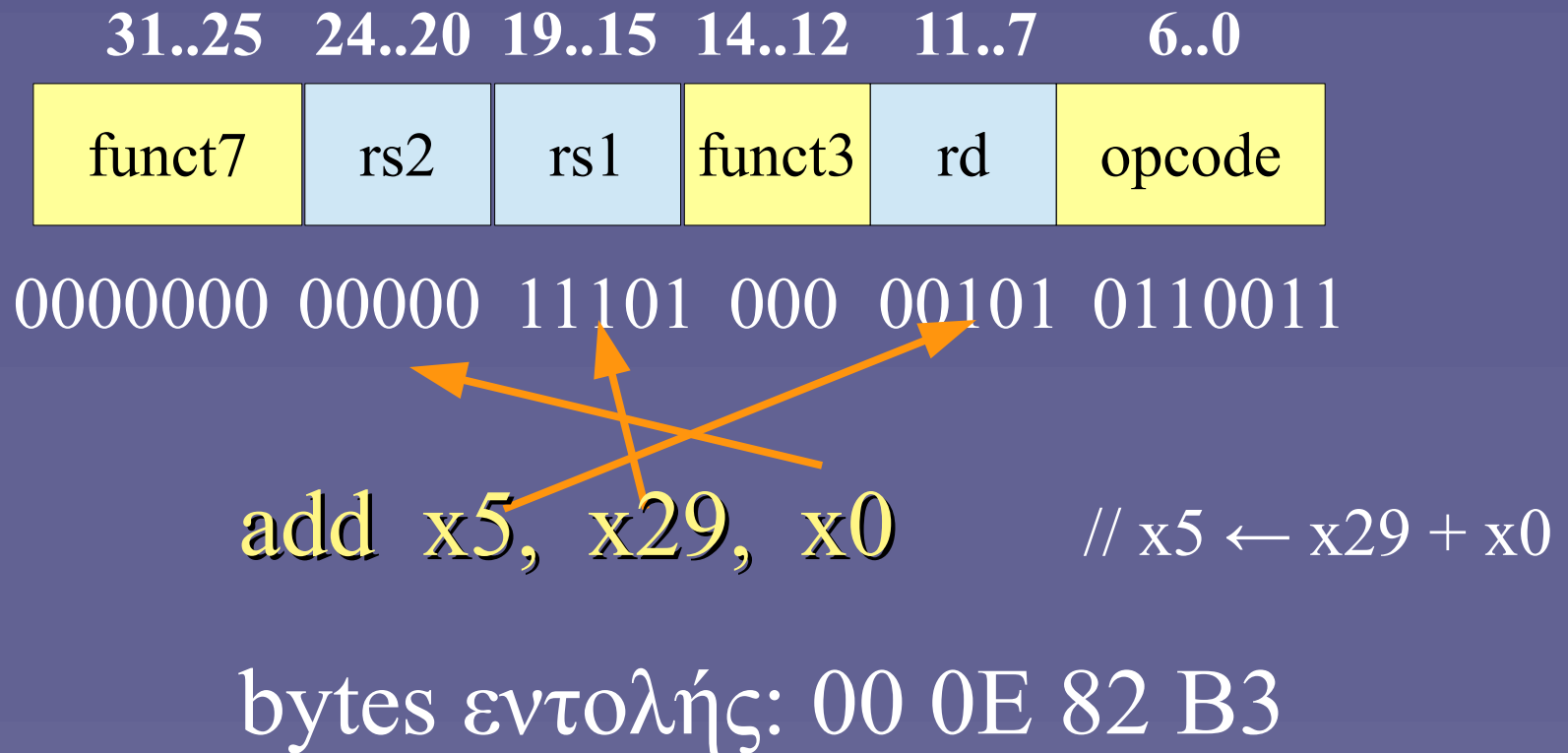


Επιλογή πράξης

funct7	funct3	πράξη
00000000	000	ADD
01000000	000	SUB
00000000	001	SLL
00000000	010	SLT
00000000	011	SLTU
00000000	100	XOR
00000000	101	SRL
01000000	101	SRA
00000000	110	OR
00000000	111	AND

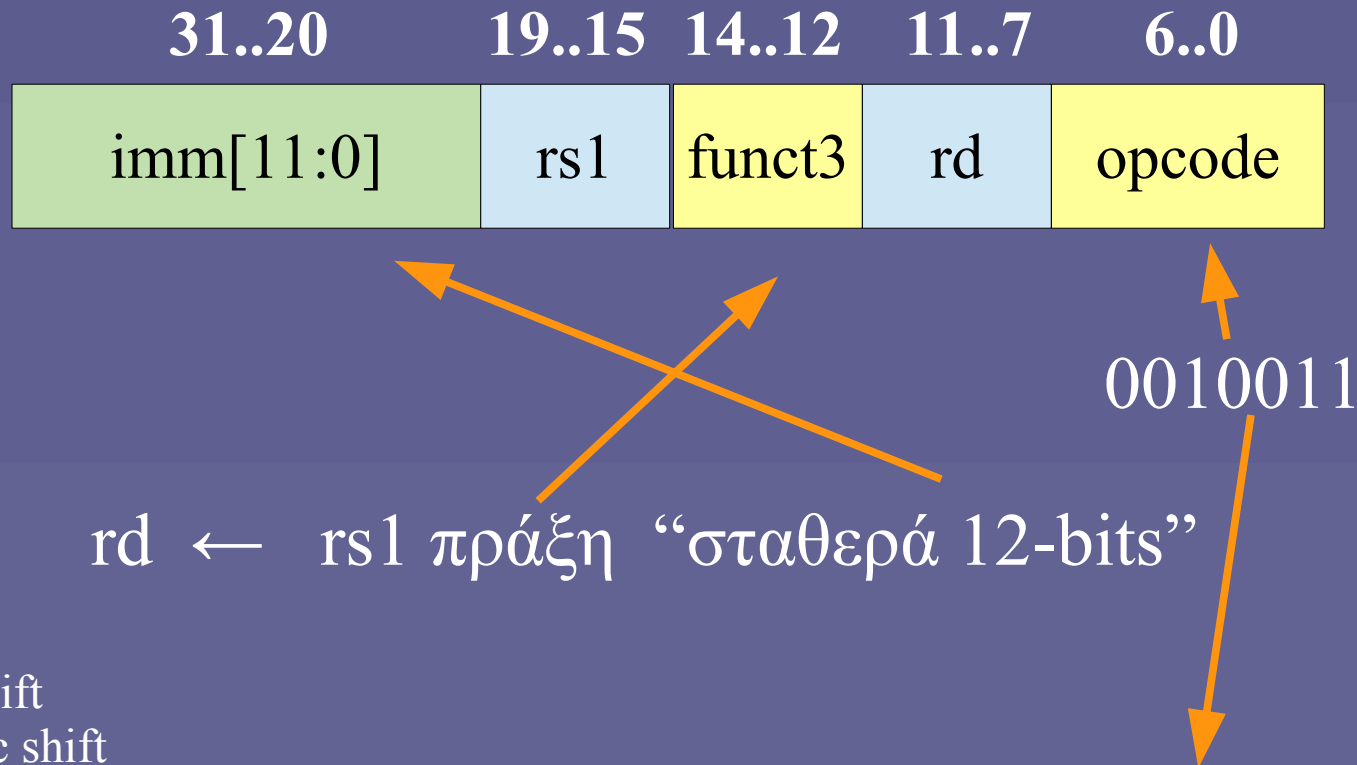
} $rd = (rs1 < rs2)?1:0$

Παράδειγμα

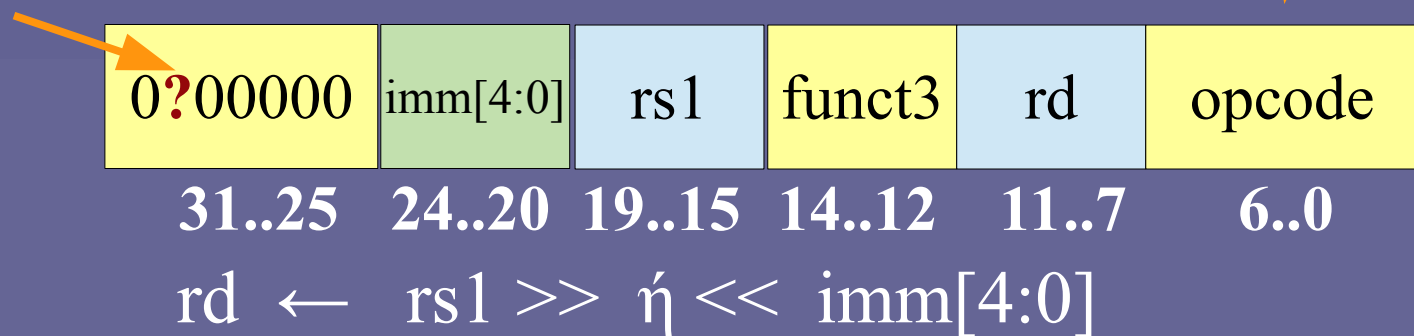


Πράξεις με σταθερή τιμή (I-type)

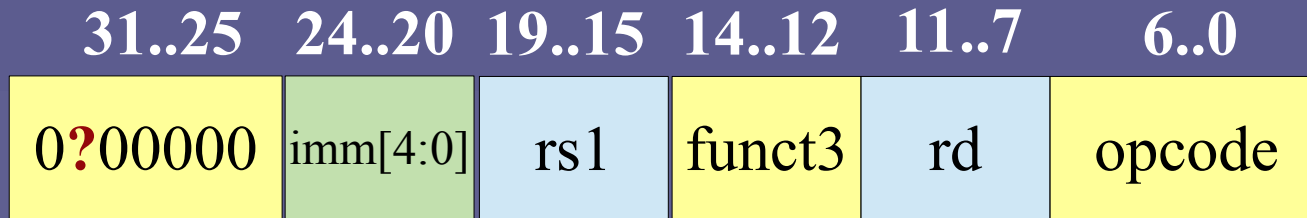
funct3	πράξη
000	ADDI
001	SLLI
010	SLTI
011	SLTIU
100	XORI
101	SRLI/SRAI
110	ORI
111	ANDI



0 = logical shift
1 = arithmetic shift



Παράδειγμα



0100000 00110 00010 101 00100 0010011

srai x4, x2, 6

// $x4 \leftarrow x2 \gg 6$

bytes εντολής: 40 61 52 13

Η αρχιτεκτονική x86-64 (64-bit mode)

- CISC αρχιτεκτονική συνόλου εντολών
 - Διευθύνσεις μνήμης (πηγή ή προορισμός) μπορούν να εμφανιστούν και σε άλλες εντολές εκτός από εντολές τύπου load – store (π.χ. add).
 - Μεταβλητού μήκους εντολές μηχανής (1-15 bytes)
 - Default: 64-bit διευθύνσεις, 32-bit δεδομένα
 - Αλλαγή (π.χ. 64-bit δεδομένα) μέσω **prefix bytes**
 - 16 64-bit καταχωρητές γενικού σκοπού
 - RAX, RBX, RCX, RDX, RSI, RDI, RBP, RSP, R8 – R15
 - Ορισμένες εντολές χρησιμοποιούν συγκεκριμένους καταχωρητές (μη ομοιόμορφο σετ εντολών)
 - Μπορεί να χρησιμοποιηθεί το 32-bit, 16-bit, 8-bit μέρος ενός καταχωρητή, π.χ. **RAX** (64 bits)/**EAX** (32-bits)/**AX** (16-bits)/**AL** (8-bits)

x86-64 Int Καταχωρητές Γενικού Σκοπού

64

32

16

8

RAX

EAX

AX

AH*

AL

RBX

EBX

BX

BH*

BL

RCX

ECX

CX

CH*

CL

RDX

EDX

DX

DH*

DL

** Μη διαθέσιμοι όταν στην εντολή υπάρχουν και οι «νέοι» καταχωρητές*

RDI

ESI

DI

DIL

RSI

EDI

SI

SIL

RBP

EBP

BP

BPL

RSP

ESP

SP

SPL

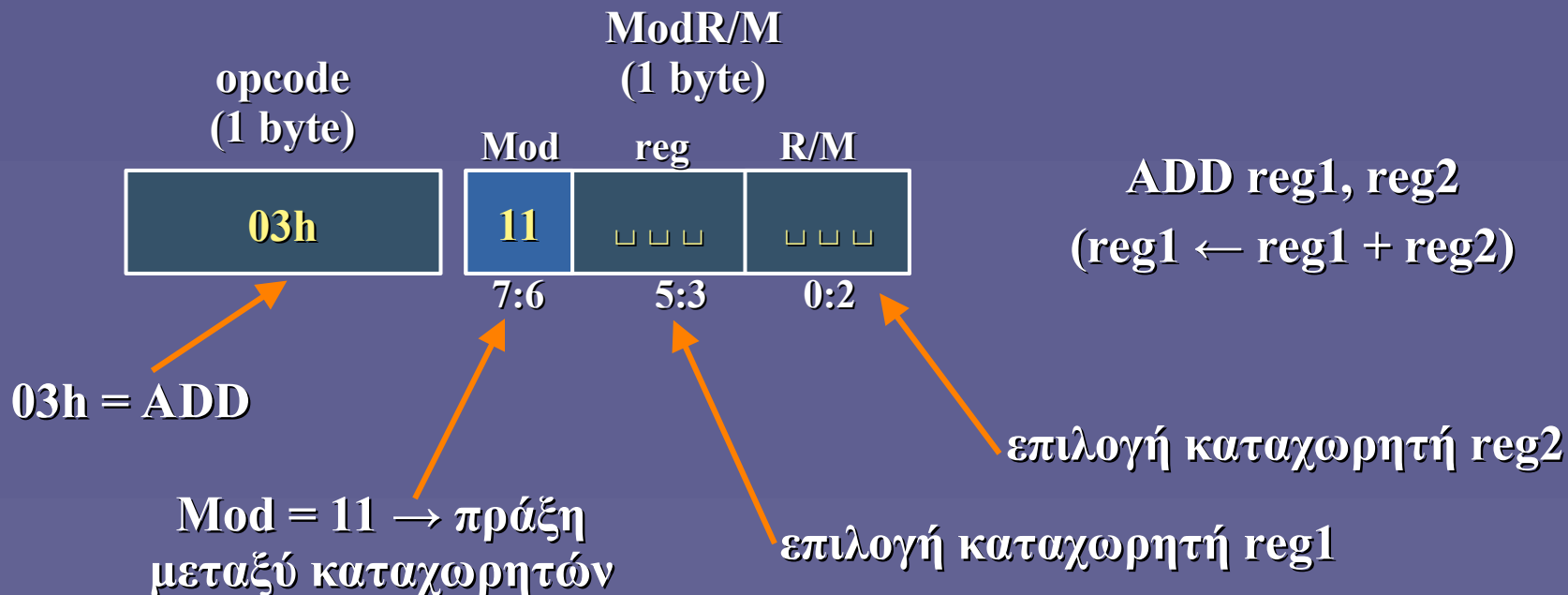
R8-15

R8-15D

R8-15W

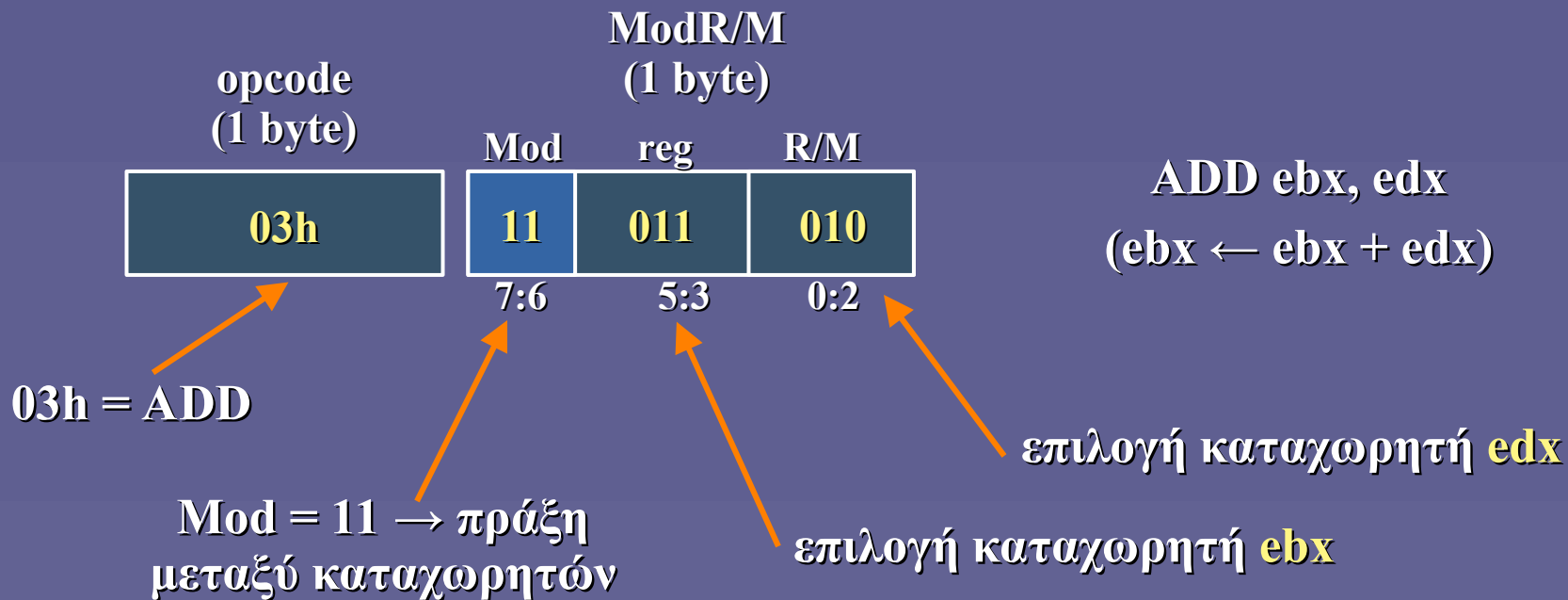
R8-15B

x86-64: πρόσθεση (ADD) με int καταχωρητές



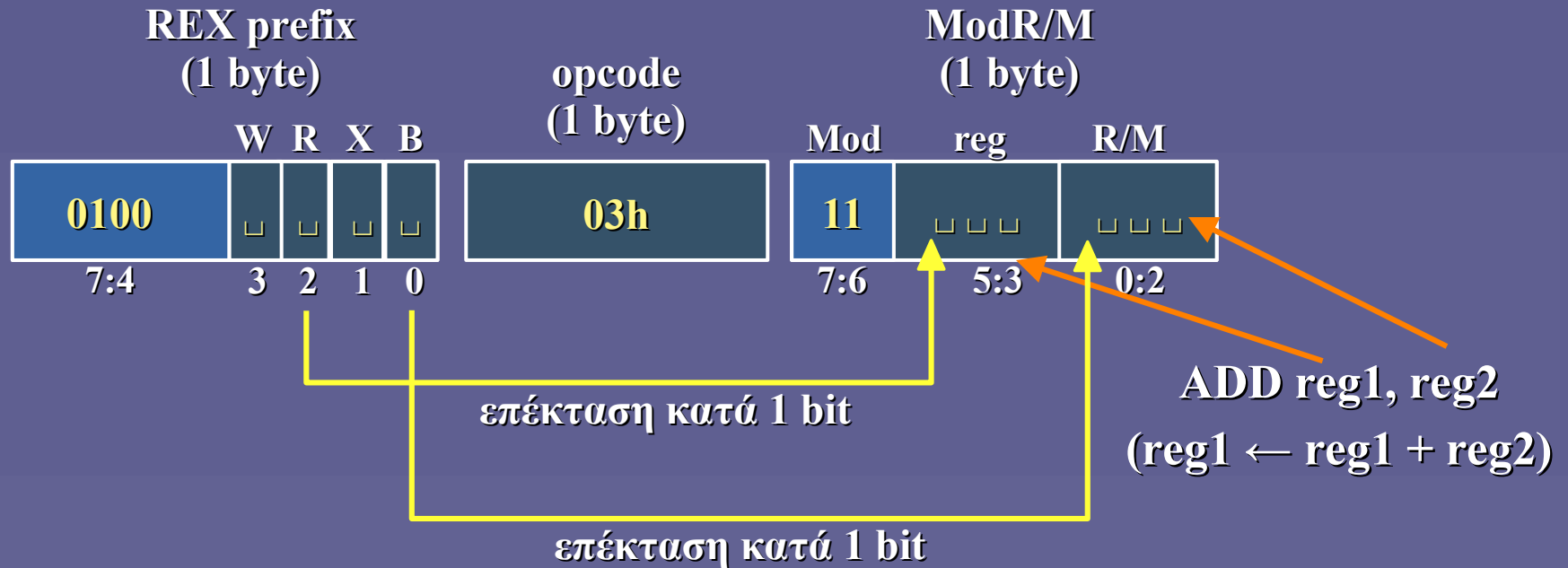
- Πηγές και προορισμός οι «κλασικοί» καταχωρητές
 - Ως CISC αρχιτεκτονική η x86-64 μπορεί να κάνει πράξεις και με μία πηγή/προορισμό από τη/στη **μνήμη** (θα το δούμε αργότερα)
 - Στο παράδειγμα τα δεδομένα είναι **32 bits**
 - Με τα κατάλληλα **opcodes** και **προθεμάτα** στην εντολή μπορούμε να επιλέξουμε 8, 16 ή 64 bit δεδομένα

Παράδειγμα



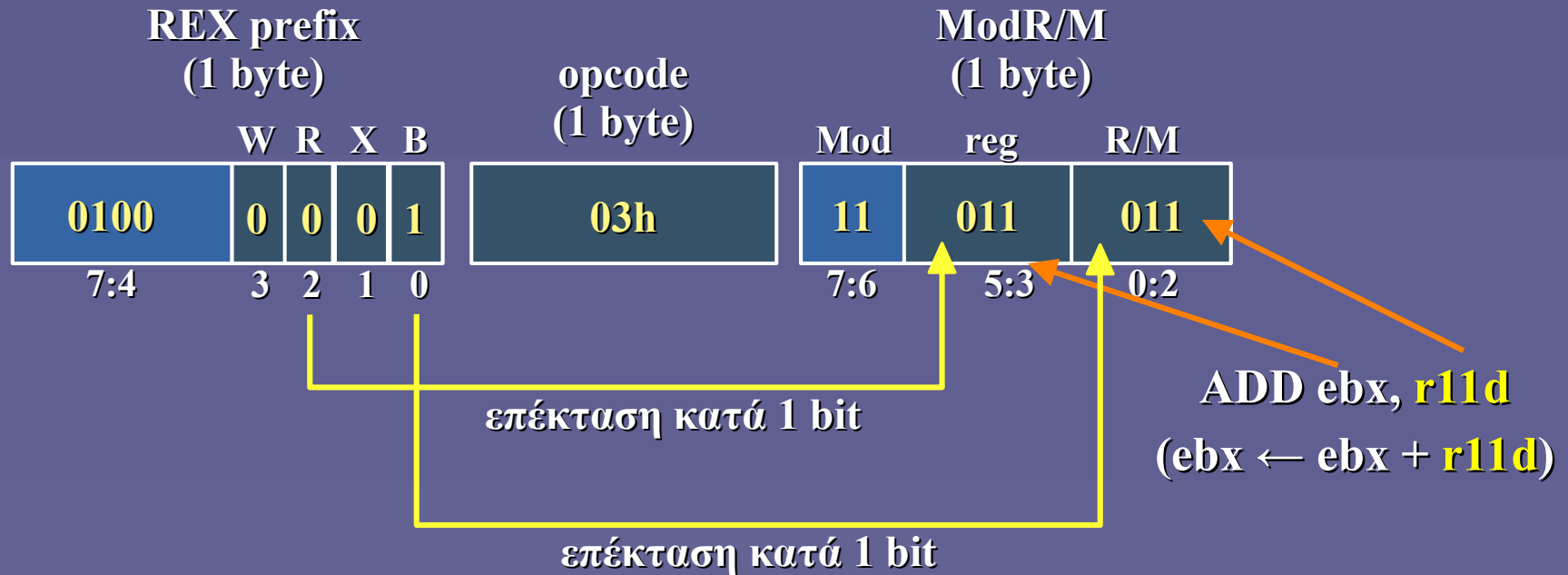
- Κωδικοποίηση των 8 κλασικών int καταχωρητών 32-bits
 - Από το 0 έως το 7: EAX, ECX, EDX, EBX, ESP, EBP, ESI, EDI
 - Συνεπώς, τα τελικά bytes της εντολής θα είναι: **03 DA**
 - Πόσοι καταχωρητές μπορεί να επιλέξει το πεδίο **reg** ή το **R/M**;

x86-64: ADD με τους «νέους» καταχωρητές R8-R15



- Χρήση προθέματος (REX prefix) για την αύξηση των bits επιλογής καταχωρητών κατά 1
 - Η εντολή τώρα θα αποτελείται από **3 bytes** αντί για 2
 - REX.R**: προσθέτει ένα bit στο πεδίο reg
 - REX.B**: προσθέτει (κατά περίπτωση) ένα bit στο πεδίο R/M

Παράδειγμα



- Χρήση προθέματος (REX prefix) για την αύξηση των bits επιλογής καταχωρητών κατά 1
 - Για να χωρέσει ο αριθμός επιλογής του R11D (1011)
 - Τα τελικά bytes της εντολής τώρα θα είναι: **41 03 DB**