

# Παραδείγματα κωδικοποίησης εντολών μηχανής

Έστω ότι έχετε αρχιτεκτονική συνόλου εντολών όπου κάθε εντολή είναι 32 bits (4 bytes).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Ο επεξεργαστής διαθέτει 32 καταχωρητές (r31..r0).

## Πληροφοριακά

Η αρχιτεκτονική του παραδείγματος είναι υποθετική. Βασίζεται όμως σε μεγάλο βαθμό στην πραγματική αρχιτεκτονική συνόλου εντολών RISC-V<sup>1</sup>, μετά από απλούστευση των τεχνικών λεπτομερειών. Έτσι, παρόλο που τα παραδείγματα είναι υποθετικά, παραμένουν απολύτως ρεαλιστικά.

## Εντολές που εκτελούν πράξεις μεταξύ καταχωρητών

Έστω ότι οι εντολές που εκτελούν πράξεις μεταξύ καταχωρητών είναι στη μορφή:

$rd \leftarrow rs1 \text{ πράξη } rs2$

(rs1, rs2 καταχωρητές δεδομένων εισόδου, rd καταχωρητής αποθήκευσης δεδομένων)

Έχει αποφασιστεί ότι για τις εντολές αυτές τα bits 0..6 της κωδικοποίησης θα είναι **0110011**.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																									0	1	1	0	0	1	1

Τι άλλο πρέπει να περιέχεται στα υπόλοιπα bits της εντολής;

- η πράξη:** στην ομάδα των εντολών της μορφής αυτής κάθε εντολή εκτελεί διαφορετική πράξη (αριθμητική ή λογική). Χρειαζόμαστε κάποια bits (π) στην κωδικοποίηση της εντολής, τα οποία θα προσδιορίζουν το είδος της πράξης.
- ποιοι είναι οι καταχωρητές:** ποιος καταχωρητής ακριβώς είναι ο rs1, ο rs2 και ο rd. Η αρχιτεκτονική διαθέτει 32 καταχωρητές, συνεπώς τα αντίστοιχα bits (rs1, rs2, rd) στην κωδικοποίηση μιας εντολής θα επιλέγουν έναν από τους καταχωρητές αυτούς. Θεωρήστε τα bits αυτά ως τη «διεύθυνση» (τον αριθμό) του επιλεγόμενου καταχωρητή.

Πόσα bits χρειάζονται για το (β);

Οι καταχωρητές είναι **32** ( $=2^5$ ). Για κάθε ένα από τα rs1, rs2 και rd, απαιτούνται 5 bits. Συνολικά  $3 \times 5 = 15$  bits.

Πόσες διαφορετικές πράξεις μπορούν να κωδικοποιηθούν;

Για την πράξη υπάρχουν **10 bits διαθέσιμα** ( $32 - 7 - 15 = 10$ ), άρα μπορούμε να έχουμε μέχρι και  **$2^{10}$  ( $=1024$ )** διαφορετικές πράξεις. Φυσικά, σε μια ρεαλιστική αρχιτεκτονική οι αριθμητικές/λογικές πράξεις δεν είναι περισσότερες από μερικές δεκάδες.

Δώστε ένα πιθανό σχήμα κωδικοποίησης

Στο σχήμα που ακολουθεί έχει τοποθετηθεί στα bits της εντολής όλη η προηγούμενη πληροφορία:

<sup>1</sup> <https://riscv.org/technical/specifications/>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
π	π	π	π	π	π	π	π	π	π	rd	rd	rd	rd	rd	rs2	rs2	rs2	rs2	rs2	rs1	rs1	rs1	rs1	rs1	0	1	1	0	0	1	1

## Παραδείγματα

Αν η πρόσθεση κωδικοποιείται ως 0000000000 και η αφαίρεση ως 1000000000, ποια τα bits της εντολής:

$r6 \leftarrow r7 + r2$  ;  $rd=r6$  (00110)  $rs1=r7$  (00111)  $rs2=r2$  (00010)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	1	1	1	0	1	1	0	0	1	1

Και ποια τα bits της εντολής:

$r3 \leftarrow r3 - r5$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0	1	1	0	1	1	0	0	1	1

Διαλέξτε κωδικούς για τις λογικές πράξεις AND, OR και XOR.

Η επιλογή είναι τυχαία. Η μόνη απαίτηση είναι να μην χρησιμοποιήσουμε κωδικούς άλλων πράξεων. Έστω ότι είναι:

AND = 0000000111

OR = 0000000110

XOR = 0000000100

Δώστε τα bits της εντολής:

$r4 \leftarrow r4 \text{ xor } r4$  ;  $r4$  (00100)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1	0	0	1	1

**Τι κάνει ουσιαστικά η προηγούμενη εντολή;**

Μηδενίζει τον r4. Θυμηθείτε ότι  $A \text{ xor } A = 0$  πάντοτε.

Αν είχατε 64 καταχωρητές πώς θα ήταν μια πιθανή κωδικοποίηση;

Θα έπρεπε να διατεθούν **18 bits** ( $3 \cdot 6$ ) για τους καταχωρητές, άρα για την πράξη θα έμεναν **7 bits** διαθέσιμα, π.χ.:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
π	π	π	π	π	π	π	rd	rd	rd	rd	rd	rd	rs2	rs2	rs2	rs2	rs2	rs2	rs1	rs1	rs1	rs1	rs1	rs1	0	1	1	0	0	1	1

## Εντολές με σταθερές (immediate)

Παραλλαγή των προηγούμενων εντολών, μία από τις πηγές εισόδου είναι **αριθμητική σταθερά** (αντί για καταχωρητής). Η μορφή των εντολών είναι η ακόλουθη:

$rd \leftarrow rs1$  πράξη immediate

Η κωδικοποίηση αυτής της δεύτερης μορφής εντολών θα πρέπει να σηματοδοτείται από διαφορετικά bits 0..6, έστω ότι είναι τα 0010011.

Ποια πληροφορία περιλαμβάνεται τώρα στα υπόλοιπα bits της εντολής;

- α) η πράξη (π)
- β) η πληροφορία του rd και του rs1
- γ) η αριθμητική σταθερά (immediate)

Αν υποθέσουμε ότι θέλουμε **8 ( $2^3$ ) διαφορετικές πράξεις**, πόσα τα bits της σταθεράς immediate; Χωρίς τη σταθερά, έχουμε 7 (opcode) + 5 (rs1) + 5 (rd) + 3 (πράξη) = **20 bits**, για τη σταθερά μένουν  $32-20 = 12$  bits.

Δώστε μια πιθανή κωδικοποίηση των εντολών αυτής της μορφής:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
π	π	π	i 1 1	i 1 0	i 9	i 8	i 7	i 6	i 5	r d	r d	r d	r d	r d	i 4	i 3	i 2	i 1	i 0	r s 1	r s 1	r s 1	r s 1	r s 1	0	0	1	0	0	1	1

### Η χρησιμότητα των bits 6..0

Παρατηρήστε ότι μόνο όταν εξετάσει η μονάδα ελέγχου της ΚΜΕ τα bits 6..0 της εντολής, μπορεί να ξέρει τι υπάρχει στα υπόλοιπα bits!

### Γιατί η σταθερά immediate σπάει σε δύο μέρη;

Δεν υπάρχει κανένας περιορισμός στην κωδικοποίηση που να επιβάλλει τα bits ενός πεδίου να βρίσκονται σε συνεχόμενες θέσεις. Σε πραγματικές σχεδιάσεις πάντως, προτιμούμε να κρατάμε κάποια πεδία σε σταθερή θέση για όλες τις μορφές εντολών: αυτό διευκολύνει τη διασύνδεση με τις πύλες που χειρίζονται την πληροφορία αυτή. Στο προηγούμενο παράδειγμα παρατηρήστε ότι προτιμήσαμε να κρατήσουμε σταθερή τη θέση των rs1 και rd, όπως και στην πρώτη μορφή εντολών, και να «μοιράσουμε» τη σταθερά(i11..i0) στα bits που απομένουν.

Με βάση τη νέα κωδικοποίηση, ποια τα bits της εντολής:

$r4 \leftarrow r4 + 33$  (δεκαδικό) ;  $33 = 000000100001$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	1