

ΔΥΑΔΙΚΟ ΑΡΙΘΜΗΤΙΚΟ ΣΥΣΤΗΜΑ

Μετατροπή βάσης

Για να μετατρέψουμε οποιουδήποτε αριθμό που είναι εκφρασμένος σε βάση x σε δεκαδικό σύστημα, δηλαδή σε βάση 10, χρησιμοποιούμε τον νόμο του πολυωνύμου.

$$\text{Γενικά, θα είναι: } b_i \cdot x^i + b_{i-1} \cdot x^{i-1} + \dots + b_1 \cdot x^1 + b_0 \cdot x^0 + b_{-1} \cdot x^{-1} + b_{-2} \cdot x^{-2} + \dots + b_{-i} \cdot x^{-i}$$

όπου x η βάση του αριθμού, b_i, b_{i-1}, \dots , κλπ. είναι η τιμή του κάθε ψηφίου στη συγκεκριμένη θέση και i είναι η θέση του ψηφίου.

Π.χ.: $(1010.01)_2 \rightsquigarrow (\dots)_{10}$

$$(1010.01)_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = (10.25)_{10}$$

Για να μετατρέψουμε έναν αριθμό απ' το δεκαδικό σε ένα οποιουδήποτε άλλο σύστημα, κάνουμε τα εξής:

Αρχικά, χωρίζουμε τον αριθμό σε ακεραίο και κλασματικό μέρος. Έπειτα, παίρνουμε το ακεραίο μέρος και το διαιρούμε με τη βάση του νέου αριθμού. Συνεχίζουμε τις διαιρέσεις έως ότου φτάσουμε στο 0. Σχηματίζοντας τα υπόλοιπα απ' την κάθε διαίρεση και αντιστρέφοντάς τα έχουμε τον καινούριο αριθμό (remainder method).

Το κλασματικό μέρος το πολλαπλασιάζουμε με τη βάση του αριθμού έως ότου φτάσουμε σε ακεραίο αριθμό κι έπειτα συλλέγουμε τα ακεραία μέρη των νέων αριθμών που βγήκαν απ' τους πολλαπλασιασμούς. (multiplication method).

Π.χ.: $(41,6875)_{10} \rightsquigarrow (\dots)_2$

Παίρνουμε το 41:

$41 \div 2$	$20 \div 2$	$10 \div 2$	$5 \div 2$	$2 \div 2$	$1 \div 2$	Άρα: $(41)_{10} = (101001)_2$
$\frac{41}{40} \mid 20$	$\frac{20}{20} \mid 10$	$\frac{10}{10} \mid 5$	$\frac{5}{4} \mid 2$	$\frac{2}{2} \mid 1$	$\frac{1}{1} \mid 0$	
υπόλοιπο 1						

Παίρνουμε το 0,6875:

$0.6875 \cdot 2 = 1.375$	$\left. \begin{array}{l} 0.6875 \cdot 2 = 1.375 \\ 0.375 \cdot 2 = 0.75 \\ 0.75 \cdot 2 = 1.5 \\ 0.5 \cdot 2 = 1.0 \end{array} \right\} (0.6875)_{10} = (0.1011)_2$
$0.375 \cdot 2 = 0.75$	
$0.75 \cdot 2 = 1.5$	
$0.5 \cdot 2 = 1.0$	

Οπότε: $(41.6875)_{10} = (101001.1011)_2$

Π.χ.: $(153,513)_{10} \rightsquigarrow (\dots)_8$

Παίρνουμε το 153:

$153 \div 8$	$19 \div 8$	$2 \div 8$	Άρα: $(153)_{10} = (231)_8$
$\frac{153}{152} \mid 19$	$\frac{19}{16} \mid 2$	$\frac{2}{2} \mid 0$	
υπόλοιπο 1			

Παίρνουμε το 0,513:

$0.513 \cdot 8 = 4.104$	$0.832 \cdot 8 = 6.656$	$0.248 \cdot 8 = 1.984$	$\left. \begin{array}{l} 0.513 \cdot 8 = 4.104 \\ 0.104 \cdot 8 = 0.832 \\ 0.832 \cdot 8 = 6.656 \\ 0.248 \cdot 8 = 1.984 \end{array} \right\} \begin{array}{l} \text{οι πολλαπλασιασμοί δεν} \\ \text{αφαιρούν σε ακεραίο} \end{array}$
$0.104 \cdot 8 = 0.832$	$0.656 \cdot 8 = 5.248$	$0.984 \cdot 8 = 7.872$	

Χρησιμοποιούμε ακριβώς 7 δεκαδικά ψηφία: $(0.513)_{10} = (0.406517 \dots)_8$

Άρα: $(153.513)_{10} = (231.406517 \dots)_8$

Για να μετατρέψουμε ένα αριθμό από βάση 2 σε μια άλλη, η οποία εκφράζεται ως βάση του 2, χρησιμοποιούμε κι εμείς για να αναπαραστήσουμε αριθμούς σε βάση 2⁴. Δηλαδή η βάση 2³=8 χρησιμοποιεί τριάδες και η 16 χρησιμοποιεί τετράδες.

Παραδείγματα μεταστροφής αριθμών σε βάση 2 στις βάσεις 8 και 16:

$$(10110)_2 = (\underline{010})_2 (110)_2 = (2)_8 (6)_8 = (26)_8$$

$$(10110110)_2 = (1011)_2 (0110)_2 = (B)_{16} (6)_{16} = (B6)_{16}$$

ΑΠΕΙΚΟΝΙΣΗ ΠΡΟΤΗΜΑΣΜΕΝΩΝ ΑΡΙΘΜΩΝ

1) Αντικείμενο προσμασθένος μέτρου: το αριστερότερο bit χρησιμοποιείται για την απεικόνιση του προσήτου. Όταν είναι 0 ο αριθμός είναι θετικός και όταν είναι 1 ο αριθμός είναι αρνητικός.

$$(+12)_{10} = (00001100)_2$$

$$(-12)_{10} = (10001100)_2$$

2) απεικόνιση συμπληρώματος ως προς ένα: μετατρέπουμε όλους τους άξονες σε μηδενικά και τα μηδενικά σε άξονες.

$$(+12)_{10} = (00001100)_2$$

Θετικοί αριθμοί έχουν το αποστερότερο bit 100 με 0

$$(-12)_{10} = (11110011)_2$$

αριθμοί " " " " " "

3) Πειρίσθιον υπερηρημίας ως προς δύο: μετατρέπουμε όλους τους ιστούς σε μηδενικά και τα μηδενικά σε άσους. Δεν μετατρέπουμε τον δεύτερο ιστό (αν υπάρχει) και τα μηδενικά που τυχόν του ακολουθούν.

$$(+12)_{10} = (00001100)_2$$

Οεζυκοί αριθμοί είχαν το υψηλότερο bit 100 120

$$(-12)_{10} = (11110100)_2$$

apuzuki " " " " " " " ↓

4) σπεικόνιση σε ~~excess~~ excess x ή biased: παίρνουμε το συγκεκριμένο αριθμό και τον προσέτουμε στον x. Το νέο αριθμό τον σπεικονίζουμε στο διαδικτύ ούστημα.

excess 128: $128 + 12 = 140$. Ans: $(+12)_{10} = (10001100)_2 (= (140)_{10})$

$$128 - 12 = 116 \quad \underline{\text{Ans:}} \quad (-12)_{10} = (01110100)_2 = (116)_{10}$$

ΑΠΕΙΚΟΝΙΣΗ ΑΡΙΘΜΩΝ ΚΙΝΗΤΗΣ ΥΠΟΛΙΑΣΤΟΛΗΣ

Οι αριθμοί κινητών υποδιαστάσεις και αυτοί που έχουν τη φαινή αμφιβολία είναι ευκολότερο να αντιμετω-
νίζονται με δυνατότες καταλληλότερων θέσεων.

Π.χ.: $+6,023 \cdot 10^{23}$, το πρόσημο (sign), ο αριθμός σταθερής υποδιαστολής (significant ni mantissa) και ο εκθέτης (exponent) απεικονίζονται κατά ειδικό τρόπο ώστε να καταλαμβάνουν μικρό αριθμό bits. (πρόσημο - εκθέτης - mantissa)

Κάθε αριθμός κινητής υποδιαστολής στρογγυλεύεται στον υποδοχστή με τη ληφθενη κανονικοποιημένη
 του μορφή (normalized form). Ο εκθέτης της βάσης ρυθμίζεται ώστε η υποδιαστολή του αριθμού
να είναι αριστερότερα κι από το πιο αριστερό μη μηδενικό ψηφίο του αριθμού (π.χ. $0.35841 \cdot 10^4$)
 Στο δυαδικό σύστημα το

Στο δυαδικό σύστημα, το ψηφίο που είναι δεξιά απ' αυτό υποδιαιρείται ενός αριθμού που είναι σε μα-
θηματικώς. Γ' αυτό ονομάζεται κρυφό (hidden) bit.
π.χ., ο αριθμός 0.11010

π.χ., ο αριθμός 0.11010 αποθηκεύεται ως 1010.

Ο κανονικοποιημένη τομή μπορεί να θεωρησάμε και τη τομή κατά την οποία η υποδια-
σολή είναι δεξιά του πιο αριστερού μη μηδενικού ψηφίου. (π.χ. 1.10101 , $3,5841 \cdot 10^3$)
Σ αυτή την περίπτωση, σε δεξιά είναι πάντα το

της υποδιαστολής, θα δαξεί το 1 (π.χ. $1.10101 \leadsto$ ο αριθμός αυξάνεται ως 1.10101)

→ χρησιμοποιείται κυρίως στο πρότυπο 754 της IEEE

ΤΟ ΠΡΟΤΥΠΟ ΑΠΕΙΚΟΝΙΣΗΣ F34 ΤΗΣ IEEE ΓΙΑ ΤΟΥΣ ΑΡΙΘΜΟΥΣ ΚΙΝΗΤΗΣ ΥΠΟΔΙΑΣΤΟΛΗΣ

Κατά το πρότυπο αυτό υπάρχουν 2 είδη απεικονίσεων των αριθμών κινητής υποδιαστολής: απλής και διπλής ακρίβειας (single και double precision)

Στην απεικόνιση απλής ακρίβειας χρησιμοποιούνται 32 bits ενώ στην διπλής 64 bits. Το bit προσήπου είναι και στις δύο περιπτώσεις στην αριστερότερη θέση και είναι 0 για θετικούς ή 1 για αρνητικούς αριθμούς. Έπειτα ακολουθεί ένα εύρος 8 bits και 11 bits αντίστοιχα για τις δύο απεικονίσεις το οποίο χρησιμοποιείται για τον εκθέτη (exponent). Στην απλή ακρίβεια η απεικόνιση γίνεται με βάση το excess 127 και στη διπλή με βάση το excess 1023. Έπειτα ακολουθούν 23 bits και 52 bits αντίστοιχα για την απεικόνιση του κλάσματος (fraction) δηλαδή του αριθμού σταθερής υποδιαστολής που είναι πολλαπλασιασμένος με δύναμη του 2. Μιλάμε για απεικόνιση κλάσματος γιατί υπάρχει και το κρυφτό (hidden) bit μπροστά από κάθε αριθμό. Φυσικοί όλοι οι αριθμοί είναι εκφρασμένοι σε κανονικοποιημένη μορφή, περιλαμβανομένου όμως και αυτού που δεν είναι και τους οποίους μετατρέπουμε κατάλληλα (denormalized).

Παραδείγματα: $+1.101 \cdot 2^5$: 0 1000100 101000000000000000000000

\downarrow \downarrow \downarrow
 πρόσημο εκθέτης κλάσμα (fraction)
 (θετικός) (5+127=132) (τα ψηφία μετά την υποδιαστολή αμελούνται από μηδενικά).

$-1.01011 \cdot 2^{-126}$: 1 00000001 010110000000000000000000

+ 0 : 0 00000000 000000000000000000000000

+ ∞ : 0 11111111 000000000000000000000000

denormalized $\leftarrow 2^{-128} = 0.01 \cdot 2^{-126}$: 0 00000000 010000000000000000000000

(ο μικρότερος δείκτης που μπορεί να απεικονιστεί είναι ο -126 στον denormalized οι οποίοι απεικονίζονται με εκθέτη 00000000 και για μη μηδενική fraction)

(not a number) $\leftarrow +NaN$: 0 11111111 011011000000000000000000

$\frac{0}{0}, \frac{\infty}{\infty}$

(double precision) $+2^{-128}$: 0 01101111111 000000000000000000000000

\downarrow
 (1023-128=895)

ΠΡΑΞΕΙΣ ΣΤΟΥΣ ΑΡΙΘΜΟΥΣ ΣΤΑΘΕΡΗΣ ΥΠΟΔΙΑΣΤΟΛΗΣ

(FIXED POINT NUMBERS)

• ΠΡΟΣΘΕΣΗ ΚΑΙ ΑΦΑΙΡΕΣΗ

- two's complement: προσθέτουμε τους δύο αριθμούς και το τυχόν κρατούμενο που υπάρχει το απορρίπτουμε (discarded carry). Υπάρχει και η περίπτωση της υπερχείλισης όταν προσθέτουμε αριθμούς μεζέδους και οι οποίοι έχουν το ίδιο πρόσημο. Η υπερχείλιση δεν εμφανίζεται ποτέ όταν προσθέτουμε αριθμούς διαφορετικού προσήμου.

$$\begin{array}{r} 00001010 (+10_{10}) \\ (+) 00010111 (+23_{10}) \\ \hline 00100001 \rightarrow (+33_{10}) \end{array}$$

$$\begin{array}{r} 00000101 (+5_{10}) \\ (+) 11111110 (-2_{10}) \\ \hline 00000011 \rightarrow (+3_{10}) \end{array}$$

discarded carry

$$\begin{array}{r} 11111111 (-1_{10}) \\ (+) 11111100 (-4_{10}) \\ \hline 11111011 (-5_{10}) \end{array}$$

$$\begin{array}{r} 01010000 (+80_{10}) \\ (+) 00110010 (+50_{10}) \\ \hline 10000010 \rightarrow (-126_{10}) \end{array}$$

περίπτωση υπερχείλισης
γιατί $0100(80+50)$ δεν μπορεί να πα-
ρασταθεί με 8bits.

- one's complement: προσθέτουμε τους δύο αριθμούς και το τυχόν κρατούμενο το κρα-
τάμε και το προσθέτουμε στη δεξιά όψη του αποτελέσματος (end-around carry).

$$\begin{array}{r} 10011 (-12_{10}) \\ (+) 01101 (+13_{10}) \\ \hline 00000 \\ (+) 10000 \\ \hline 00001 (+1_{10}) \end{array}$$

$$\begin{array}{r} 0101.1 (+5.5_{10}) \\ (+) 1110.0 (-1.0_{10}) \\ \hline 00011.1 \\ (+) 1.0 \\ \hline 0100.1 (+4.5_{10}) \end{array}$$

- πρόσημο-μέτρο: εξετάζουμε τα πρόσημα των αριθμών. Αν οι αριθμοί είναι θετικοί τότε τους προσθέτουμε. Αν οι αριθμοί είναι αρνητικοί τότε προσθέτουμε τα μέτρα τους και στο αποτέλεσμα θέτουμε 1 στην αριστερότερη θέση. Αν έχουμε ένα θετικό μεγαλύτερο ενός αρνητικού τότε κάνουμε θετικό 1 στην αριστερότερη θέση. Αν έχουμε ένα θετικό και τον αφαιρούμε απ' τον θετικό. Αν έχουμε το αντίστροφο τότε απ' τον αρ-
νητικό αφαιρούμε τον θετικό.

$$\begin{array}{r} 00000101 (+5_{10}) \\ (+) 00000101 (+2_{10}) \\ \hline 00000111 (+7_{10}) \end{array}$$

$$\begin{array}{r} 00000101 (+5_{10}) \\ (-5_{10}) \} \\ (-2_{10}) \} \rightarrow \begin{array}{r} 00000010 (+2_{10}) \\ (-) 00000101 (-7_{10}) \\ \hline 10000111 (-7_{10}) \end{array} \end{array}$$

$$\begin{array}{r} 00000101 (+5_{10}) \\ (+5_{10}) \} \\ (-2_{10}) \} \rightarrow \begin{array}{r} 00000010 (+2_{10}) \\ (-) 00000101 (-7_{10}) \\ \hline 10000111 (-7_{10}) \end{array} \end{array}$$

$$\begin{array}{r} 10000111 (-7_{10}) \\ (-7_{10}) \} \\ (+5_{10}) \} \rightarrow \begin{array}{r} 00000101 (+5_{10}) \\ (-) 10000111 (-7_{10}) \\ \hline 10000010 (-2_{10}) \end{array} \end{array}$$

Γενικώς, σε πρόσημο-μέτρο, αν οι αριθμοί έχουν ίδια πρόσημα βάζουμε το κοινό τους πρό-
σημο στα αποτελέσματα και ελέγχουμε για υπερχείλιση εξετάζοντας το κρατούμενο της
υψηλότερης βαθμίδας. Αν οι αριθμοί έχουν διαφορετικά πρόσημα συγκρίνουμε τα μέ-
τρα τους και το πρόσημο του αποτελέσματος είναι ίσο με το πρόσημο αυτού με το με-
γαλύτερο μέτρο. Έπειτα, αφαιρούμε το μεγαλύτερο απ' το μικρότερο. (μεγαλύτερος-μικρό-
τερος)

• ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΣ ΚΑΙ ΔΙΑΙΡΕΣΗ

Για τον πολλαπλασιασμό, αν οι αριθμοί είναι θετικοί τότε τους πολλαπλασιάζουμε
κανονικά (και στις 3 αναπαράσεις). Αν οι αριθμοί είναι και οι δύο αρνητικοί
τότε το αποτέλεσμα θα είναι ο πολλαπλασιασμός των μέτρων τους (1's-προσ-μέτρο).

Αν έχουμε έναν αρνητικό και ένα θετικό, σε 2's complement, τότε κά-
νουμε επέκταση προσήμου (sign extension) στον αρνητικό κατά τόσα ψηφία
όσα θα έχει και το αποτέλεσμα. Δηλαδή, για δύο αριθμούς 4 bits, όπου το αποτέλε-
σμα θα βγει 8bits (2n), προσθέτουμε στον αρνητικό άλλα 4 ψηφία (άκρων). Αν οι αριθ-
μοί είναι και οι δύο αρνητικοί, σε 2's complement, τότε κάνουμε και στις 2
sign extension. (το αποτέλεσμα θα είναι πάντα 2n).

ΠΑΡΑΔΕΙΓΜΑΤΑ ΠΟΛΥΠΛΑΣΙΑΣΜΟΥ ΑΡΙΘΜΩΝ ΣΤΑΘΕΡΗΣ ΥΠΟΔΙΑΣΤΟ-

ΛΗΣ

• $\left. \begin{matrix} +2: 0010 \\ +5: 0101 \end{matrix} \right\}$

0010 (+2 ₁₀)
(x) 0101 (+5 ₁₀)
0010
0000
0010
(+) 0000
<u>00001010</u> (+7 ₁₀)

1101 (+3 ₁₀)
(x) 1011 (+11 ₁₀)
1101
1101
1101
0000
(+) 1101
<u>10001111</u> (+14 ₁₀)

1's complement
2's complement
πρόσημο-μέτρο

• 2's complement

$\left. \begin{matrix} +2: 0010 \\ +5: 0101 \end{matrix} \right\} \rightarrow \begin{matrix} 1110 (-2_{10}) \\ 1101 (-5_{10}) \end{matrix}$

Καίναμε sign extension κατά 4-bits

1111 1110 (-2₁₀)

(x) 1111 1101 (-5₁₀)

1111 1110
0000 0000
1111 10
1111 0
1110
110
10
(+) 0
<u>00001110</u> (+10 ₁₀)

discarded carry

1111 1110
(+) 1111 1000
① 1111 0110
(+) 1111 0000
① 1110 0110
(+) 1110 0000
① 1100 0110
(+) 1100 0000
① 1000 0110
(+) 1000 0000
① 0000 0110

• 2's complement

$\left. \begin{matrix} +2: 0010 \\ +5: 0101 \end{matrix} \right\} \begin{matrix} 1110 (-2_{10}) \\ 0101 (+5_{10}) \end{matrix}$

Καίναμε sign extension του -2 κατά 4-bits:

1111 1110 (-2₁₀)

(x) 0101 (+5₁₀)

1111 1110
0000 0000
1111 10
(+) 0000 0
① 1111 0110

discarded carry

→ ανίχνευση του -10₁₀ στο 2's complement

Είνα: $+10_{10} = 00001010 \xrightarrow{-10} 11110110 (-10_{10})$

ΠΡΑΞΕΙΣ 2 1042 ΑΡΙΘΜΟΥΣ ΚΙΝΗΤΗΣ ΥΠΟΔΙΑΣΤΟΛΗΣ

(floating point numbers)

• Πρόσθεση και αφαίρεση

Αρχικά, θα πρέπει να μετατρέψουμε τους αριθμούς που θα προσθέσουμε ή θα αφαιρέσουμε ώστε να έχουν τον ίδιο εκθέτη. Έπειτα, τα κλασματικά μέρη (fractions) προστίθενται ή αφαιρούνται. Το τελικό αποτέλεσμα κανονικοποιείται. Μ' αυτό τον τρόπο υπάρχει απώλεια ακρίβειας (loss of precision).

Π.χ. έχουμε τους αριθμούς: $0.101 \cdot 2^3$ και $0.111 \cdot 2^4$. Για να τους προσθέσουμε, θα κινάμε ρυθρίζοντας τον μικρότερο εκθέτη ώστε να γίνει ίσος με το μεγαλύτερο.

Οπότε: $0.101 \cdot 2^3 = .010 \cdot 2^4$ (χάνουμε $.001 \cdot 2^3$)

Άρα: $.010 \cdot 2^4 + .111 \cdot 2^4 = 1.001 \cdot 2^4 = .1001 \cdot 2^5$ (κανονικοποίηση)

Πρέπει όμως να στρογγυλοποιήσουμε το αποτέλεσμα στα 3 σημαντικά ψηφία του αριθμού:

$$.101 \cdot 2^3 + .111 \cdot 2^4 = .100 \cdot 2^5$$

• Πολλαπλασιασμός και διαίρεση

Για να πολλαπλασιάσουμε δύο αριθμούς κινητής υποδιαστολής προσθέτουμε τους εκθέτες τους και έπειτα πολλαπλασιάζουμε τα κλασματικά μέρη (fractions). Τέλος, φέρνουμε το αποτέλεσμα στην κανονική του μορφή και το στρογγυλοποιούμε. Αριθμοί με ίδιο πρόσημο δίνουν θετικό αποτέλεσμα και με αντίθετο πρόσημο αρνητικό αποτέλεσμα.

Π.χ. $+0.101 \cdot 2^2$ και $-0.110 \cdot 2^{-3}$. Είναι: $2 + (-3) = -1$

Επίσης:

$$\begin{array}{r} 0.101 \\ \times 0.110 \\ \hline 0000 \\ 0101 \\ 0101 \\ \hline 0.01110 \end{array}$$

Άρα:

$$\begin{aligned} (+.101 \cdot 2^2) \times (-.110 \cdot 2^{-3}) &= -.01111 \cdot 2^{-1} = \\ &= -.1111 \cdot 2^{-2} = \underline{-.111 \cdot 2^{-2}} \text{ (κανονικοποίηση και} \\ &\text{στρογγυλοποίηση)} \end{aligned}$$

Για να διαιρέσουμε δύο αριθμούς, αφαιρούμε τους εκθέτες και διαιρούμε τους συντελεστές. Το αποτέλεσμα κανονικοποιείται και στρογγυλοποιείται.

ΕΠΙΚΟΙΝΩΝΙΑ ΜΕΣΟ ΥΠΟΛΟΓΙΣΤΩΝ

- Είναι αναγκαία η διαμοίραση πόρων (resource sharing) για την κάλυψη των αναγκών όλων και περισσότερων χρηστών.
- Δημιουργία αυξανόμενων αναγκών (στρατιωτικός τομέας, οικονομικός τομέας, ανθρωπιστικοί παράγοντες)

Καναάλια επικοινωνίας

- σύρματα, καλώδια
- δορυφορικά
- ασύρματα

Ποιότητα επικοινωνίας

Εύρος καναλιού ή ζώνης (bandwidth) ορίζεται ο μέγιστος αριθμός πληροφοριών ανά μονάδα χρόνου που μπορεί να μεταφερθεί.

Είναι αναπόφευκτος ο περιορισμός του διατιθέμενου εύρους καναλιού ή ζώνης (bandwidth) εξαιτίας 2 παραγόντων:

- * Ασφάλεια (απαιτώση της κρυπταγραφίας)
 - * Αξιοπιστία δεδομένων (data integrity)
- } είναι απαραίτητη η μεταφορά πληροφοριών για τον καθορισμό τους (άρα ποτέ δεν πετυχαίνεται μέγιστη μεταφορά δεδομένων).

MODEM S (modulator - demodulator)

Σε γενικές γραμμές, ένα modem μετατρέπει τα ψηφιακά σε αναλογικά σήματα και αντίστροφα. Ειδικότερα, είναι μια συσκευή που δέχεται χαρακτηριστικά απ' τον υπολογιστή στη μορφή διεπίπεδων σημάτων (two-level signals), ένα bit σε καθορισμένη χρονική περίοδο, και μεταδίδει τα bits σε ομάδες του ενός ή των δύο, στη διαμόρφωση αμplitudes (amplitude modulation) ή της συχνότητας (frequency modulation) ή της φάσης (phase modulation), και αντίστροφα.

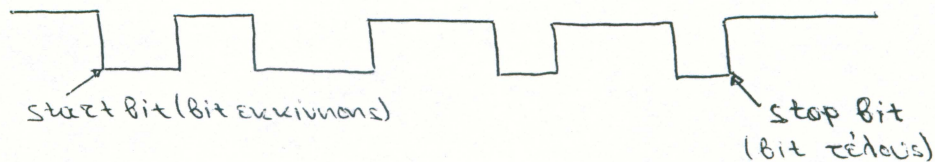
Η μεταφορά δεδομένων μεταξύ 2 ή περισσότερων modems γίνεται με βάση την ταχύτητα μετάδοσης του πιο αργού modem.

Κατηγορίες καναλιών

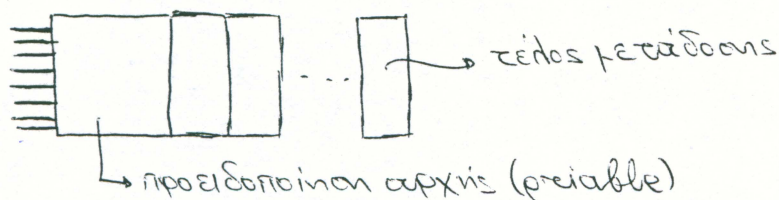
- simplex line: μετάδοση πληροφοριών κατά μία μόνο διεύθυνση. Η μία άκρη του καλωδίου (γραμμής) είναι ο αποστολέας και η άλλη ο παραλήπτης.
- half duplex line: μετάδοση πληροφοριών και προς τις δύο κατευθύνσεις, όχι όμως ταυτόχρονα. Εναλλαγή ρόλων μεταξύ αποστολέα και παραλήπτη.
- full duplex line: μετάδοση πληροφοριών (αποστολή και παραλαβή) και προς τις δύο κατευθύνσεις ταυτόχρονα. Επειδή δύο μεταδόσεις στέλνονται παράλληλα, για κάθε διεύθυνση, μια τέτοια γραμμή μπορεί να μεταδώσει περισσότερες πληροφορίες απ' όσα η half-duplex γραμμή. Επιπλέον δε σπαταλάται χρόνος ~~για~~ για την εναλλαγή των διευθύνσεων.

ΜΕΤΑΔΟΣΗ ΠΛΗΡΟΦΟΡΙΩΝ

Σειριακά: Μετάδοση πληροφοριών bit προς bit μέσα από ένα μόνο αγωγό



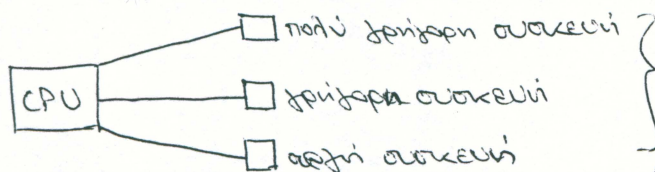
Παράλληλα: Μετάδοση πληροφοριών μέσω πολλών αγωγών ταυτόχρονα.



Μετάδοση πληροφορίας χωρίς συγχρονισμό (asynchronous transmission)

Σ' αυτή την περίπτωση, τα χρονικά κενά μεταξύ δύο χαρακτήρων δεν είναι καθορισμένα. Ωστόσο, τα χρονικά διαστήματα μεταξύ δύο συνεχόμενων bits μέσα στον ίδιο χαρακτήρα είναι τα ίδια. Για να επιτρέψουμε στον παραλήπτη να αναγνωρίσει την αρχή ενός χαρακτήρα, ένα bit εκκίνησης (start bit) μεταδίδεται πριν από κάθε χαρακτήρα. Επίσης, μετά από το τέλος κάθε χαρακτήρα, μεταδίδεται και ένα ή δύο bits τέλους (stop bits), για αλίσθηση της αξιολογίας. Τα start bits έχουν τιμή 0 και τα stop bits τιμή 1.

Μέσα στον υπολογιστή, η μεταφορά πληροφοριών γίνεται ως εξής:

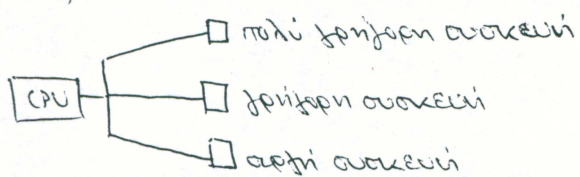


ο επεξεργαστής ευθύνεται για τη λειτουργία της κάθε συσκευής (η χρήση θα δουλεύει μέτρια και η αργή αργά).
Κάθε συσκευή ερμηνεύεται ανεξάρτητα απ' τις άλλες.

Μετάδοση πληροφορίας με συγχρονισμό (synchronous transmission)

Σ' αυτή την περίπτωση, τα χρονικά κενά μεταξύ δύο χαρακτήρων είναι αβυσσώδη τα ίδια. Εδώ η αναγνώριση για μετάδοση start και stop bits εξαλείφεται. Τα moderns συγχρονίζονται με το να συνεχίζουν να στέλνουν χαρακτήρες ακόμη κι όταν καμία πληροφορία δεν μεταδίδεται. Ένας ειδικός χαρακτήρας στέλνεται όταν δε υπάρχουν πλέον πληροφορίες προς μετάδοση. Η σύγχρονη μετάδοση απαιτεί τα ρολόγια (clocks) του παραλήπτη και του αποστολέα να παραμένουν συγχρονισμένα για μεγάλα χρονικά διαστήματα.

Μέσα, στον υπολογιστή, η μετάδοση πληροφοριών γίνεται ως εξής:



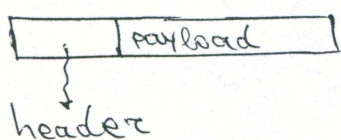
Η μεταφορά πληροφορίας επιτελείται με διευκρίνιση με συγκεκριμένο ρυθμό και επιτυγχάνεται με βοήθεια την πιο αργή συσκευή.
Μ' αυτό τον τρόπο χάνουμε σε ταχύτητα, κερδίζουμε όμως σε αξιοπιστία.

ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ

Τα δίκτυα υπολογιστών χωρίζονται στα τοπικά δίκτυα ή LAN's (Local Area Networks) και στα ευρείας περιοχής δίκτυα ή WAN's (Wide Area Networks). Τα πρώτα υποστηρίζουν υποδοχιστές μέχρι αποστάσεις χιλιομέτρων. Τα δεύτερα καλύπτουν ανάγκες δικτύωσης υπολογιστών σε μεγάλες αποστάσεις ή συνδέουν πολλά τοπικά δίκτυα μαζί.

Ένα δίκτυο αποτελείται απ' το υλικό (καλώδια, κυκλώματα προσαρμογής), το λογισμικό (κομμάτι του λειτουργικού συστήματος) και από κανόνες (πρωτόκολλα) που έχουν να κάνουν με τον τύπο δεδομένων, το χρονισμό, κτλ. χαρακτηριστικών του δικτύου.

Η μεταφορά δεδομένων γίνεται μέσω ειδικών πακέτων μεταφοράς (packets) για τη συβλετότητα των οποίων ευθύνονται τα διάφορα πρωτόκολλα επικοινωνίας. Η δομή ενός πακέτου είναι:



Η κεφαλή (header) περιέχει πληροφορίες για τον αριθμό του πακέτου (timestamp), για τον έλεγχο των δεδομένων (π.χ. CRC) και στοιχεία σχετικά με τον αποστολέα ή τον παραλήπτη. (σε ποιόν θα απευθύνεται το πακέτο)

Στα δίκτυα υπάρχουν 7 επίπεδα (standards ISO)

- Application: επίπεδο εφαρμογών (e-mail, browsing, ...)
- Presentation: επιβάλλει κοινά μοτίβα πάνω στο δίκτυο (π.χ. για μηχανές που επικοινωνούν με one's ή two's complement, big ή little endian, κτλ.)
- Session: μεταφορά αξιοπιστών πληροφοριών
- Transport: φροντίζει για την αξιοπιστία των δεδομένων που πρόκειται να μεταφερθούν
- Network: τοποθετεί σαν header τη διεύθυνση του παραλήπτη, φροντίζει για την αξιοπιστία, την ταυτότητα των δεδομένων και έχει πλήρη επίγνωση της τοπολογίας του δικτύου.
- Data link: μετατροπή των πακέτων σε ωρίμα δεδομένα και αντίστροφα
- Physical: μετάδοση αμείνων δεδομένων

* Κύριο πλεονέκτημα του διαδρόμου (bus) είναι ότι μπορούμε εύκολα να τον επεκτείνουμε προσθέτοντας επιπλέον περιφερειακές μονάδες. Αρκεί η κάθε μονάδα να ακολουθεί κάποιο συγκεκριμένο πρωτόκολλο επικοινωνίας.

Με τον όρο πρωτόκολλο επικοινωνίας διαδρόμου (bus communication protocol) αναφερόμαστε στο σύνολο των κανόνων που καθορίζουν τον τρόπο με τον οποίο γίνεται η ανταλλαγή σημάτων και δεδομένων μεταξύ δύο περιφερειακών μονάδων που είναι διασυνδεδεμένες μέσω του διαδρόμου αυτού.

Προτεραιότητα περιφερειακών συσκευών (priority of I/O devices)

Τι θα συμβεί σε περίπτωση που η ΚΜΕ και ένας ελεγκτής περιφερειακών συσκευών (I/O controller) χρειαστούν την ίδια στιγμή τον διαδρόμο; Η απάντηση είναι ότι ένα ειδικό κύκλωμα που καλείται «διατητής διαδρόμου» (bus arbiter) αποφασίζει για την εν λόγω προτεραιότητα. Γενικά δίνεται προτίμηση στις περιφερειακές συσκευές εις βάρος της ΚΜΕ γιατί οι δίσκοι και άλλες κινούμενες συσκευές αν σταματήσουν μπορεί να χάσουν δεδομένα.

Όταν καμία πληροφορία I/O δεν είναι σε εξέλιξη, η ΚΜΕ μπορεί να έχει όλους τους κύκλους λειτουργίας (συχνότητα ρολογιού) του διαδρόμου για τον εαυτό της ώστε να δίνει αναφορά στη μνήμη. Ωστόσο, όταν κάποια ή κάποιες περιφερειακές συσκευές τρέχουν (είναι σε λειτουργία), αυτές θα ζητήσουν και θα τους παραχωρηθεί ο διαδρόμος όταν τον χρειαστούν. Αυτή η διεργασία καλείται κλέψιμο κύκλου (cycle stealing) και μειώνει την ταχύτητα του υπολογιστή. Οι μεγάλοι υπολογιστές έχουν πολλούς διαδρόμους, γ'αυτό δεν υποφέρουν από τέτοιου είδους προβλήματα.

Το κλέψιμο κύκλου (cycle stealing) επιτελείται ως εξής:

- 1) Πραγματοποιείται αίτηση απ' το περιφερειακό για να λειτουργήσει διαμέσου της διακοπής της ΚΜΕ.
- 2) Η ΚΜΕ αποθηκεύει το τωρινό περιβάλλον που ήδη έτρεχε και μεταφέρει τις απαραίτητες πληροφορίες στον DMA controller για το ποιές θέσεις μνήμης θα χρησιμοποιήσει.
- 3) Όταν ο DMA controller βρει κύκλο στον οποίο η ΚΜΕ δεν χρησιμοποιεί τη μνήμη αναλαμβάνει μία-μία τη μεταφορά δεδομένων.
- 4) Όταν ο DMA controller τελειώσει τη μεταφορά, στέλνεται ένα σήμα διακοπής (interrupt) μέσω του interrupt controller, όπου ειδοποιεί την ΚΜΕ ότι τελείωσε η μεταφορά.

Ελαχιστοποίηση προβλημάτων που σχετίζονται με τις περιφερειακές συσκευές:

• Τα προβλήματα που έχουμε να αντιμετωπίσουμε είναι:

Έλεγχος της κατάστασης της κάθε συσκευής

Μεταφορά δεδομένων από και προς τη συσκευή

Θέση μεταξύ απόδοσης και κόστους

Η ΚΜΕ είναι υπεύθυνη κάθε φορά για τη δημιουργία καταχωρητών στη μνήμη μέσα στους οποίους θα αποθηκεύει δεδομένα που σχετίζονται με την κατάσταση της κάθε συσκευής.

Η χρήση και διαχείριση αυτών περιφερειακών απαιτεί την ύπαρξη μιας μόνο ΚΜΕ. Γενικώς, όμως, η χρήση της κάθε συσκευής διατέσσεται του συστήματος είναι μια αβή διαδικασία. Αυτό οφείλεται στο χρόνο που δαπανείται για τον έλεγχο της κατάστασης της κάθε συσκευής (χρόνος Polling), στη μεταφορά δεδομένων μέσω του διαδρόμου και στην εναλλαγή των προγραμμάτων ή διεργασιών (process swapping).

Επικεντρώνοντας την προσοχή μας στην ελαχιστοποίηση του χρόνου Polling, αυτή απαιτεί τη δημιουργία μιας μονάδας ελέγχου διακοπών (interrupt controller) και τη χρήση μονάδων απευθείας προσελεύσεως μνήμης (DMA controllers) καθώς και τη χρήση κυκλωμάτων bus arbiters.

Μ'αυτού τον τρόπο είναι δυνατός ο προγραμματισμός της προτεραιότητας μεταξύ των συσκευών, και η επίτευξη του cycle stealing. Η μονάδα ελέγχου διακοπών μέσω των διανυσμάτων διακοπών (interrupt vectors) καθορίζει το πότε η ΚΜΕ θα ασχοληθεί ή όχι με την κάθε συσκευή.

ΛΑΘΗ - ΚΩΔΙΚΕΣ ΑΝΙΧΝΕΥΣΗΣ, ΔΙΟΡΘΩΣΗΣ ΛΑΘΩΝ

Απόσταση (distance) μεταξύ 2 λέξεων: ο αριθμός των δυαδικών ψηφίων που πρέπει να αλλάξουν τιμή ώστε να μετατραπεί μια λέξη του κώδικα (codeword) σε μία άλλη του ίδιου κώδικα.

Ο αριθμός των θέσεων bits κατά τον οποίο διαφέρουν δύο λέξεις ενός κώδικα (codewords) αναφέρεται ως απόσταση Hamming (Hamming distance). Αυτή είναι ισού με την ελάχιστη απόσταση (distance) μεταξύ 2 λέξεων.

Οι ιδιότητες ανίχνευσης και διόρθωσης λάθους από έναν κώδικα εξαρώνται απ' την απόσταση Hamming που έχει αυτός. Γενικά, ένας κώδικας για να είναι σε θέση να ανιχνεύει d λάθη, θα πρέπει η απόσταση Hamming των λέξεων του (codewords) να είναι $d+1$. Επίσης, για τη διόρθωση k λαθών είναι απαραίτητος κώδικας με απόσταση Hamming ισού με $2k+1$.

Π.χ., ο κώδικας με τις παρακάτω κωδικολέξεις, έχει απόσταση Hamming ισού με 5, αφού κάθε λέξη διαφέρει απ' τις άλλες κατά ελάχιστη απόσταση 5 ψηφίων.

0000000000, 0000011111, 1111100000, 1111111111

Αυτός ο κώδικας είναι σε θέση να ανιχνεύσει $d+1=5 \Rightarrow d=4$ λάθη και να διορθώσει $2k+1=5 \Rightarrow k=2$ λάθη.

Γενικά, οι κώδικες ανίχνευσης και διόρθωσης λαθών δημιουργούνται με την προϋπόθεση κατάλληλων bits ισότητας (parity bits) έτσι ώστε σε κάθε αναπαράσταση της κωδικολέξης - πληροφορίας να υπάρχει άρτος ή περιττός αριθμός άρτων.

Έστω ότι έχουμε n -bits πληροφορίας και προσθέτουμε k -bits ελέγχου (check bits). Οπότε οι κωδικολέξεις θα έχουν μήκος $n+k$ -bits. Οι νόμιμες (legal) κωδικολέξεις είναι 2^n σε πλήθος. Επίσης, ξέρουμε ότι ένας κώδικας για να ~~ανιχνεύει~~ ^{ανιχνεύει} $n+k$ bits πρέπει η απόσταση Hamming των λέξεών του να είναι $(n+k)+1$. Οι δυνατοί συνδυασμοί όλων των κωδικολέξεων των $n+k$ bits είναι 2^{n+k} . Οπότε, για να βρούμε πόσα bits πρέπει να προσθέσουμε στην πληροφορία, δηλαδή για να βρούμε το k , θα ισχύει:

$$2^n \cdot (n+k+1) \leq 2^{n+k} \Rightarrow \boxed{n+k+1 \leq 2^k}$$

Για δεδομένο n βρίσκουμε τον ελάχιστο αριθμό για το k ώστε να διορθώσουμε αυτά λάθη.

Π.χ., για $n=10$ bit πληροφορίας θα πρέπει να προσθέσουμε k -bits ελέγχου:

$$11+k \leq 2^k \Rightarrow \boxed{k \geq 4}$$

10 bit πληροφορίας:

0	1	1	1	0	0	1	0	1	0
10	9	8	7	6	5	4	3	2	1

Το συνολικό μήνυμα θα έχει $n+k=14$ ελάχιστων bits

Το επόμενο πρόβλημα που έχουμε είναι το πού θα τοποθετήσουμε τα 4 parity bits για την ανίχνευση απλού λάθους. Η απάντηση είναι ότι όλοι οι αριθμοί απ' το 1 έως το $n+k$ ελέγχονται για το ποιοι απ' αυτούς είναι δυνάμεις του 2. Όσοι απ' αυτούς βρούμε ότι είναι δυνάμεις του 2 τότε στην αντίστοιχη θέση τους τοποθετούμε τα parity bits. Όλες οι άλλες θέσεις χρησιμοποιούνται για την αρχική πληροφορία. Δηλαδή, στο προηγούμενο παράδειγμα, οι αριθμοί $1(2^0)$, $2(2^1)$, $4(2^2)$, $8(2^3)$ είναι δυνάμεις του 2.

Άρα:

0	1	1	1	0	0	0	1	0	1	0	0	1	1
14	13	12	11	10	9	8	7	6	5	4	3	2	1

Στη διαδικασία αυτή σπρίτται και ο κώδικας Hamming (Hamming's algorithm)

ΚΟΔΙΚΑΣ HAMMING

Αρχικά, κάνουμε σύμβαση ως προς την ισοτιμία και θα μετράμε άρτια ισοτιμία, δηλαδή άρτιο αριθμό άσσων.

Κάθε bit ελέγχου που τοποθετείται μέσα σε m -bits πληροφορίας ελέγχει τον εαυτό του και ανά μία n -άδα απ' τον εαυτό του, όπου n είναι ο αριθμός της θέσης που έχει τοποθετηθεί το συγκεκριμένο bit ελέγχου. Δηλαδή στο μήνυμα που φαίνεται παρακάτω, όπου τα c_8, c_4, c_2, c_1 είναι check bits θα ισχύει:

0	1	1	1	0	0	c_8	1	0	1	c_4	0	c_2	c_1
14	13	12	11	10	9	8	7	6	5	4	3	2	1

Το c_1 ελέγχει τα bits που βρίσκονται στις θέσεις: 1, 3, 5, 7, 9, 11, 13

Ανάλογα, το c_2 θα ελέγχει τα bits: 2, 3, 6, 7, 10, 11, 14

" " c_4 " " " : 4, 5, 6, 7, 12, 13, 14

" " c_8 " " " : 8, 9, 10, 11, 12, 13, 14

Τα check bits τίθενται ανάλογα στο 0 ή στο 1 αν τα bits που ελέγχουν έχουν συνολικά άρτιο αριθμό άσσων ή περιττό αριθμό άσσων αντίστοιχα.

Π.χ., για το c_1 θα είναι: οι θέσεις 3, 5, 7, 9, 11, 13 έχουν άρτιο αριθμό άσσων, και μαζί στα 4 στον αριθμό. Άρα το c_1 τίθεται στο 0 ώστε να διατηρηθεί ο άρτιος αριθμός άσσων.

Με παρόμοια διαδικασία, το c_2 τίθεται στο 0 (2-άσσοι), το c_4 τίθεται στο 0 (4-άσσοι) και το c_8 τίθεται στο 1 (3-άσσοι). Επομένως, το τελικό μήνυμα θα έχει τη μορφή: 01110011010000

Για να δούμε, τώρα, πώς δουλεύει η διόρθωση αυτού λάθους, θα υποθέσουμε ότι κατά τη μετάδοση άλλαξε το 4^ο bit αρχικής πληροφορίας, δηλαδή το 7^ο bit του μηνύματος, από 1 σε 0. Δηλαδή, το μήνυμα που θα λάβει ο αποστολέας θα είναι: 01110010010000. Ο αποδέκτης πληροφορίας θα ελέγξει το λάθος, υπολογίζοντας κι αυτός άρτια ισοτιμία. Ειδικότερα, θα ελέγξει τα bit-ισοτιμίας (parity bits)

P_8, P_4, P_2, P_1 . Οπότε: $P_1 = 1, P_2 = 1, P_4 = 1, P_8 = 0$. Αυτές οι τιμές βρέθηκαν εφαρμόζοντας την ίδια διαδικασία με τα check bits τα οποία ελέγχουν τα bits των αντίστοιχων θέσεων. Το P_1 , π.χ., τέθηκε στο 1 επειδή στις θέσεις 1, 3, 5, 7, 9, 11, 13 του λαμβανόμενου μηνύματος υπάρχει περιττός αριθμός άσσων (3-άσσοι). Το ίδιο συμβαίνει και με τα άλλα. Για να βρούμε, τώρα, σε ποιά θέση έχει γίνει το λάθος (στην 7^η θέση) προσθέτουμε στις δείκτες όσων των parity bits που είναι λάθος (incorrect), που έχουν τελεί δηλαδή στο 1. Στην προκειμένη περίπτωση έχουμε: $1 + 2 + 4 = 7$, άρα το 7^ο bit του μηνύματος είναι λάθος και πρέπει να αλλάχθει από 0 σε 1. (ή μεση βουίδα της σειράς $P_8, P_4, P_2, P_1 \rightarrow 0111 = 7$)

Γενικώς, θα ισχύει:

αν τα P_8, P_4, P_2, P_1 είναι:

- $0, 0, 0, 0 \rightarrow$ δεν υπάρχει λάθος (σωστό μήνυμα)
- $0, 0, 0, 1$
- $0, 0, 1, 0$
- $1, 1, 1, 0$
- $1, 1, 1, 1 \rightarrow$ πιθανώς υπάρχουν περισσότερα λάθη που δεν μπορούν να διορθωθούν

Τα διπλάσια ισχύουν και για περισσότερα parity bits.

ΚΩΔΙΚΑΣ ΙΣΟΤΙΜΙΑΣ ΣΤΗΛΗΣ - ΓΡΑΜΜΗΣ

Ο κώδικας αυτός δημιουργείται προσθέτοντας bits-ισοτιμίας (parity bits) ώστε να έχουμε άρτιο ή περιττό αριθμό αΐσεων, σε κάθε τετράδα bits μιας πληροφορίας. Θεωρούμε, π.χ., ότι έχουμε την πληροφορία:

00110001110. Τότε, χωρίζουμε αυτή σε τετράδες: 0011, 0001, 1110

Οπότε:

0011	0
0001	1
1110	1
1100	0

Θεωρούμε ότι έχουμε άρτια ισοτιμία. Οπότε στο παραπάνω μήνυμα πρέπει να προσεθούν θ ακόμη bits, όσα βρίσκονται μέσα σε κύκλο.

Επομένως, το κανονικό μήνυμα θα αποτελείται από $12 + 8 = 20$ ψηφία, ή θα έχει όπως λέτε overhead 8/12.

στην προκείμενη περίπτωση
έχουμε 4 γραμμές και 5 στήλες ή άρτιο αριθμό αΐσεων

Επίσης, έχουμε την πληροφορία 1011101111011. Τότε:

προσθέτουμε
καταλλήλως
αριθμό
ή ξενικών

1011	1
1101	1
1101	1
0010	1
1001	0

Εδώ, το κανονικό μήνυμα θα έχει $14 + 2 + 4 + 5 = 25$ bits πληροφορίας (overhead 11/14)

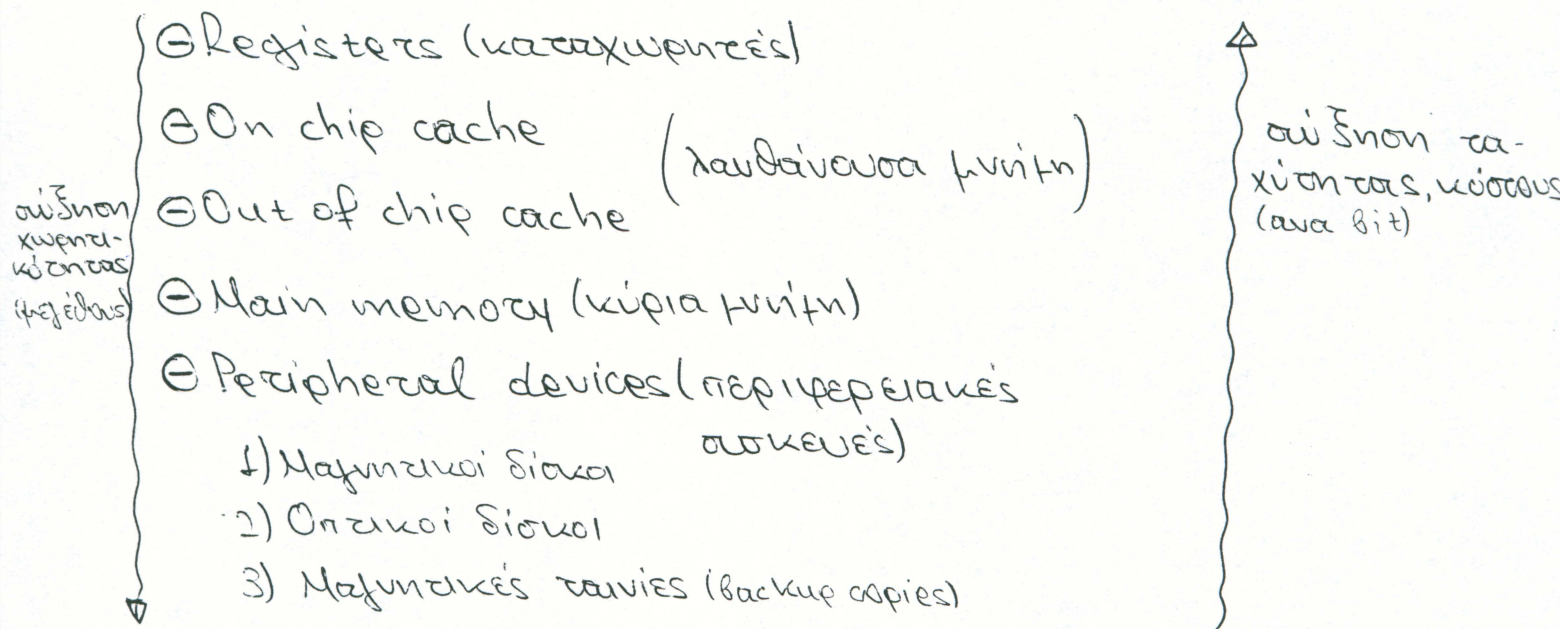
Οι δύο κώδικες Hamming και στήλης-γραμμής είναι οι πιο εύχρηστοι. Κάθε φορά θα επιλέξουμε όποιον μας συμφέρει με βάση πάντα το μικρότερο αριθμό των bits του τελικού μηνύματος που προκύπτει αν εφαρμόσουμε τον καθένα απ' αυτούς στην αρχική πληροφορία.

Μνήμη - βασικά αξιώματα:

- 1) Η SRAM είναι από κατασκευαστικής πλευράς, πιο γρήγορη απ' τη DRAM ($SRAM \sim 2ns$, $DRAM \sim 20ns$)
- 2) Όσο πιο γρήγορη είναι για μνήμη τόσο πιο ακριβή είναι
- 3) Όσο μεγαλύτερη είναι για μνήμη τόσο πιο αργή είναι

Για x θέσεις μνήμης έχω $\log_2 x$ bits διεύθυνσης και από 0 έως $x-1$ φυσικές διευθύνσεις.

Ιεραρχία μνήμης: η μνήμη αποτελείται απ' τα εξής κομμάτια:



Στην «αρχή» της ιεραρχίας, όπου το μέγεθος της μνήμης είναι πολύ μικρό, βρίσκονται οι πιο γρήγορες και ακριβές μνήμες. Στη «βύση» της ιεραρχίας το μέγεθος της μνήμης είναι μεγάλο και τοποθετούνται αργές και φθηνές μνήμες.

Με τον τρόπο αυτό ελαχιστοποιείται ο μέσος χρόνος προσπέλασης στη μνήμη και το κόστος αποθήκευσης ανά μονάδα πληροφορίας, ενώ μεγιστοποιείται ο συνολικός χώρος αποθήκευσης του συστήματος.

ΔΙΕΥΘΥΝΣΕΙΣ ΜΝΗΜΗΣ

Οι μνήμες αποτελούνται από ένα σχετικά μεγάλο αριθμό κελιών (cells ή locations) καθένα απ' τα οποία μπορεί να αποθηκεύσει ένα κομμάτι πληροφορίας. Κάθε κελί έχει έναν αριθμό που καλείται διεύθυνση. Αυτών ακριβώς τον αριθμό χρησιμοποιούν τα προγράμματα για να αναφέρονται σ' αυτό.

Αν μια μνήμη έχει n κελιά τότε οι διευθύνσεις τους θα κυμαίνονται από 0 έως $n-1$. Όλα τα κελιά μιας μνήμης περιέχουν τον ίδιο αριθμό bits. Αν ένα κελί αποτελείται από k bits τότε σ' αυτό το κελί μπορεί να αποθηκευθεί κάθε ένας απ' τους 2^k συνδυασμούς των k bits. Γειτονικά κελιά έχουν πάντα διαφορετικές διευθύνσεις.

Οι διευθύνσεις μνήμης απεικονίζονται στο δυαδικό σύστημα. Έτσι, αν μια διεύθυνση μνήμης περιέχει m bits τότε οι διευθυνοδοτούμενες θέσεις μνήμης (κελιά) που είναι δυνατές είναι 2^m . Κατά συνέπεια, όταν μια διεύθυνση μνήμης έχει m bits εύρος τότε το ίδιο εύρος θα έχει και η αρτηρία διευθύνσεων του διαδρόμου (bus) του συστήματος.

Π.χ., όταν τα κελιά μιας μνήμης είναι 12 και αριθμούνται απ' το 0 μέχρι το 11 τότε οι διευθύνσεις τους θα είναι 4-bit εύρους. Βέβαια με 4 bits έχουμε $2^4 = 16$ συνδυασμούς. Σ' αυτή την περίπτωση μόνο οι 12 πρώτοι συνδυασμοί μας ενδιαφέρουν.

Όταν μια μνήμη έχει 16384 διαφορετικές διευθυνοδοτούμενες θέσεις μνήμης (κελιά) τότε οι διευθύνσεις για τα κελιά θα είναι:

$2^m = 16384 \Rightarrow \underline{m = 14}$. Οι διευθύνσεις κυμαίνονται απ' το 0 μέχρι το 16383 και η κάθε μια απ' αυτές αναπαρίστανται με 14 bits, ενώ ο αριθμός τους (16384) είναι ίσος με το συνδυασμό των 14 bits μεταξύ τους (2^{14}). Σ' αυτή την περίπτωση η αρτηρία διευθύνσεων θα έχει εύρος 14 bits.

ελαχιστοποίηση χρόνου αναζήτησης → χρέιον πολλών κεφαλών
" " αναμονής → τοποθέτηση κεφαλών
σε αλληλεπιδραστικά ονεία

Βοηθητική φύση:

- ασφαδής αποθήκευση δεδομένων
- μεγάλη αξιοπιστία

Ιδεατή μνήμη: υλοποιείται με υλικό και λογισμικό
(ευδιάμεσος μηχανισμός κύριος-βοηθητικής
μνήμης)

Διαφορές ιδεατής - κρυφής μνήμης (cache)

- υλοποιείται με υλικό και λογισμικό

- διευκόλυνση του προγραμματιστή

- αν αφαιρεθεί κάποια προγράμματα θα τρέξουν και κάποια άλλα όχι (αυτά που δε χωρούν στη μνήμη)

- γίνεται αντιληπτή απ' το λογισμικό του συστήματος αλλά όχι απ' το λογισμικό των εφαρμογών

- υλοποιείται με υλικό

- αύξηση απόδοσης

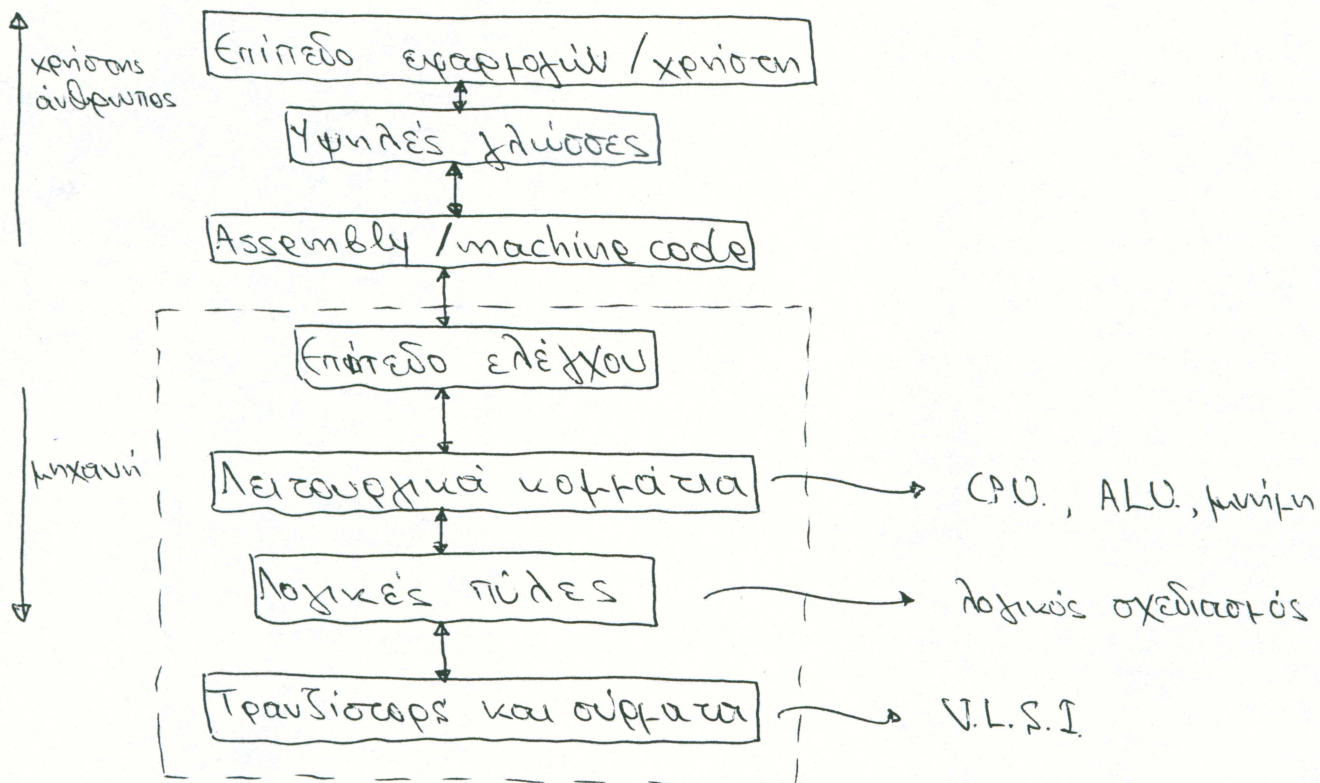
- δεν είναι απαραίτητη για τα προγράμματα

- δε γίνεται αντιληπτή ούτε απ' το λογισμικό του συστήματος ούτε απ' το λογισμικό των εφαρμογών (εκτός της file cache που γίνεται αντιληπτή απ' το λειτουργικό σύστημα).

- translators (μεταφραστές): ειδικά προγράμματα που μετατρέπουν τα προγράμματα εγερτομένων από μια μορφή κατανοητή στον άνθρωπο στη μορφή που μπορεί να εκτελεστεί απ' τον υπολογιστή.

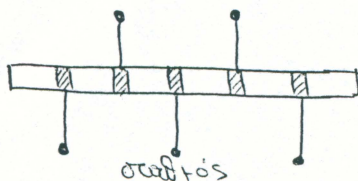
Διακρίνονται σε:

- assemblers (συμβολομεταφραστές): ειδικά προγράμματα για τη μετατροπή ελλών από συμβολική γλώσσα σε γλώσσα μηχανής.
(γραπμένοι σε γλώσσα μηχανής)
- compilers (μεταγλωττιστές): ειδικά προγράμματα που διαβάζουν ένα πρόγραμμα σε μια γλώσσα προγραμματισμού και το μετατρέπουν στην αντίστοιχη ακολουθία από εντολές μηχανής.
- interpreters (διερμηνευτές): ειδικά προγράμματα που συγχρόνως μεταφράζουν και εκτελούν ένα πρόγραμμα εντολή προς εντολή.



ΤΟΠΟΛΟΓΙΕΣ ΔΙΚΤΥΩΝ (LAN):

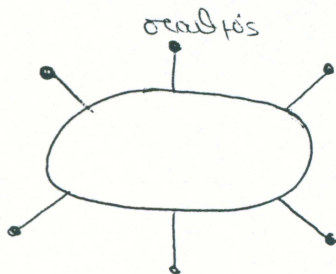
1) Τοπολογία αρτηρίας (bus): Όλοι οι σταθμοί διαφορίζονται την ίδια αρτηρία μέσω ειδικού interface.



- Η προσθήκη ή η αφαίρεση ενός σταθμού είναι σχετικά εύκολη υπόθεση.
- Περιοριστός σαφής του καλωδίου
- Ο έλεγχος είναι αποκεντρωμένος
- Είναι εύκολη η υποκλοπή (broadcasting)
- Η ταυτόχρονη μετάδοση πληροφοριών μπορεί να δημιουργήσει παρεμπόδιση (collision)

⊗ Ενδείνεται όταν έχουμε λίγους σταθμούς και εφαρμογές που η χρήση τους το απαιτεί.

2) Τοπολογία δακτυλίου (ring): η σύνδεση όλων των σταθμών αποτελεί έναν ανοιχτό ή κλειστό δακτύλιο.

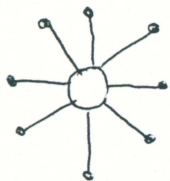


- Τα δεδομένα μεταδίδονται πάντα προς την ίδια κατεύθυνση.
- Όταν γίνεται η μετάδοση ενός συγκεκριμένου πακέτου κανένας σταθμός δεν μπορεί να μεταδώσει ταυτόχρονα πληροφορίες.
- Οι πληροφορίες που μεταδίδονται (πακέτο) περνούν απ' όλους τους σταθμούς.

⊗ Ο μέγιστος χρόνος που ένας σταθμός μπορεί να μείνει ανενεργός είναι:

αριθμός σταθμών $\cdot t$, όπου t είναι ο χρόνος μετάδοσης ενός πακέτου πληροφορίας στο δακτύλιο.

3) Τοπολογία αστέρα (star): όλες οι πληροφορίες περνούν απ' τον εξυπηρετητή (server) του δικτύου ο οποίος ρυθμίζει την επικοινωνία των υπολογιστών μεταξύ τους.



- Υψηλή ταχύτητα μετάδοσης δεδομένων
- Απαραίτητη η ύπαρξη εξυπηρετητή

Κάθε σταθμός για να επικοινωνήσει με έναν άλλο στέλνει αρχικά μια «αίτηση» προς τον server ο οποίος τον συνδέει με τον άλλο σταθμό.

Περιφερειακές συσκευές:

- Στους μεγάλους υπολογιστές (mainframes) υπάρχουν ειδικοί επεξεργαστές I/O (I/O processors) οι οποίοι καλούνται κανάλια πληροφοριών (data channels). Όλες οι περιφερειακές συσκευές συνδέονται σ' αυτά τα κανάλια.

Όταν η κΜΕ χρειαστεί μια περιφερειακή μονάδα τότε φορτώνει ένα ειδικό πρόγραμμα σε ένα απ' τα κανάλια, και λέει σ' αυτό να το εκτελέσει. Το κανάλι χειρίζεται όλες τις πληροφορίες I/O από και προς την κύρια μνήμη, αφήνοντας έτσι την κΜΕ να ασχοληθεί με άλλα πράγματα. Όταν το κανάλι τελειώσει, στέλνει στην κΜΕ ένα ειδικό σήμα, που ονομάζεται διακοπή (interrupt), ώστε να κάνει την κΜΕ να σκεφαστεί ότι κάνει και να προσέξει το κανάλι. Το πλεονέκτημα αυτής της διαδικασίας είναι ότι επιτρέπει στην κΜΕ να μεταφέρει όλη τη δουλειά που σχετίζεται με I/O πληροφορίες στο κανάλι. Έτσι, I/O πληροφορίες και υπολογισμοί διαχειρίζονται ταυτόχρονα.

Στους μεγάλους υπολογιστές υπάρχουν τουλάχιστον 3 διαδρομές (buses). Ο διαδρομός μνήμης (memory bus) επιτρέπει στα κανάλια να διαβάζουν και να γράφουν λέξεις απ' τη μνήμη. Ο διαδρομός εισόδου-εξόδου (I/O bus) επιτρέπει στην κΜΕ να εκδίδει εντολές στα κανάλια και να επιτρέπει στα κανάλια να διακοπτούν (interrupt) την κΜΕ. Ο τρίτος διαδρομός επιτρέπει στην κΜΕ να προσπελάει τη μνήμη χωρίς να χρησιμοποιεί τους άλλους διαδρόμους.

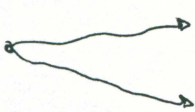
• Στους προσωπικούς υπολογιστές (PCs) υπάρχει συνήθως ένας απλός διαδρομός που συνδέει την κΜΕ, τη μνήμη και τις περιφερειακές συσκευές. Κάθε περιφερειακή συσκευή (I/O device) αποτελείται απ' τον ελεγκτή, που περιέχει τα περισσότερα ηλεκτρονικά μέρη, και το ίδιο το περιφερειακό. Η δουλειά του ελεγκτή είναι να ελέγχει το περιφερειακό και να χειρίζεται την προσπέλαση στο διαδρόμο (bus) για χάρη του. Ειδικότερα ο ελεγκτής είναι προγραμματισμένος να σπάζει το ρεύμα των bits σε λέξεις και να γράφει τις λέξεις στη μνήμη. Ένας ελεγκτής που διαβάζει ή γράφει ένα σύνολο πληροφοριών (block of data) από και προς τη μνήμη χωρίς τη διαμεσολάβηση της κΜΕ, καλείται μονάδα απευθείας προσπέλασης μνήμης (DMA unit ~ direct access memory).

⊗ Ο απλός διαδρομός του συστήματος (system bus) αποτελείται, επιμέρους, απ' το διαδρόμο πληροφοριών (data bus), το διαδρόμο διευθύνσεων (address bus), που δείχνει το μέρος που έχει στείλει η πληροφορία, το διαδρόμο ελέγχου (control bus), που περιγράφει τον τρόπο και το πώς θα στείλουν οι πληροφορίες, και το διαδρόμο ενέργειας (power bus).

Πλεονεκτήματα και μειονεκτήματα της γλώσσας assembly :

⊕ Πλήρης έλεγχος στο υλικό

⊕ Μεγαλύτερη αξιοπιστία

Ευκολία 

- εξοργική προφραγία των
- εύκολο debugging (ανίχνευση-διόρθωση λαθών) σε σχέση με τον κώδικα μηχανής (machine code)

⊕ Βελτιστοποιημένος κώδικας σε όγκο και ταχύτητα

— * —

⊖ Δεν είναι μεταφέρσιμος ο κώδικας (διαφορά μηχανημάτων)

⊖ Περισσότερη προσπάθεια-πολύς κώδικας-λιγότερα άτομα-δυσκολότερο debugging

Η απόδοση ενός υπολογιστή δεν εξαρτάται μόνο απ' την ταχύτητα των ολοκληρωμένων κυκλωμάτων που τον απαρτίζουν, εξαρτάται, επίσης, απ' τις δραστηριότητες εισόδου/εξόδου, τις περιφερειακές συσκευές που είναι συνδεδεμένες μ' αυτόν, την ταχύτητα του λειτουργικού συστήματος, την ταχύτητα προπελάσης δίσκων και κύριας μνήμης.

Η έννοια της απόδοσης δεν είναι ταυτοτική για όλα τα πιθανά είδη χρήσών. Ο χρήσις ενός υπολογιστή μετράει την απόδοσή του σε σχέση με το πόσο γρήγορα εκτελεί ένα πρόγραμμα. Ο χρήσις ενός υπολογιστικού κέντρου, που εστιάζεται σε πολύπλοκο περιβάλλον, καταλαβαίνει ότι ο υπολογιστής του είναι πιο αποδοτικός όταν ολοκληρώνεται η εκτέλεση περισσότερων προγραμμάτων ανά ώρα (throughput).

Γενικά, σε ένα σύστημα με πολλούς χρήστες, ο χρόνος απασχόλησης του επεξεργαστή ή άλλου διαθέσιμου πόρου από ένα πρόγραμμα είναι ο ίδιος. Όμως, σε ένα σύστημα με πολλούς χρήστες, ο χρόνος εκτέλεσης ενός προγράμματος μπορεί να διαφέρει. Αυτό εξαρτάται απ' τον αριθμό των προγραμμάτων που περιμένουν να εκτελεστούν και απ' τον αριθμό των διαθέσιμων πόρων που τυχόν απαιτεί το καθένα για την εκτέλεσή του.

⊗ Δύο υπολογιστικά συστήματα που διαθέτουν ακριβώς το ίδιο σύνολο εντολών σε γλώσσα Assembly δεν μπορούν να εκτελέσουν τον ίδιο κώδικα μηχανής επειδή μπορεί να έχουν διαφορετικούς συμβολομεταφραστές (assemblers).

⊗ Δύο υπολογιστικά συστήματα που διαθέτουν ακριβώς το ίδιο σύνολο εντολών σε γλώσσα Assembly και μας δίνουν τον ίδιο αριθμό για το δείκτη MIPS δεν έχουν την ίδια απόδοση επειδή μπορεί να διαχειρίζονται διαφορετικές πολυπλοκότητες εντολής και να συνδέονται με διαφορετικές περιφερειακές συσκευές.

⊗ Ένα υπολογιστικό σύστημα το οποίο διαθέτει εντολές για λέξεις (words) των 16 δυαδικών ψηφίων σε χρειάζεται να διαθέτει πάντα και αντίστοιχες εντολές για bytes. Αυτό είναι αναγκαίο όταν θέλουμε να εδo-σφαλίσουμε για συμβατότητα και να μειώσουμε το χρόνο προπελάσης στη μνήμη όταν ο διάδρομος δεδομένων (data bus) είναι 8bit εύρους.