Τμήμα Πληροφορικής
Πανεπιστήμιο Πειραιώς

Λογική Σχεδίαση Ψηφιακών Συστημάτων

Δυαδικά Συστήματα

Μιχάλης Ψαράκης

Συστήματα αρίθμησης γενικά

- Σε ένα σύστημα αρίθμησης με βάση r οι αριθμοί παριστάνονται με τα ψηφία 0, ..., r-1
- Παραδείγματα συστημάτων αρίθμησης:
  - Δεκαδικό
    - Βάση 10, Ψηφία 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
  - Πενταδικό
    - Βάση 5, Ψηφία 0, 1, 2, 3, 4
  - Οκταδικό
    - Βάση 8, Ψηφία 0, 1, 2, 3, 4, 5, 6, 7
- Ένας αριθμός $\alpha_n \alpha_{n-1} \ldots \alpha_1 \alpha_0 . \alpha_{-1} \ldots \alpha_{-m}$ με n+1 ψηφία αριστερά της υποδιαστολής και m ψηφία δεξιά ισούται με:

  $\alpha_n \times r^n + \alpha_{n-1} \times r^{n-1} + \ldots + \alpha_1 \times r^1 + \alpha_0 \times r^0 + \alpha_{-1} \times r^{-1} + \ldots + \alpha_{-m} \times r^{-m}$

Δεκαδικό – Δυαδικό

- Δεκαδικό σύστημα αρίθμησης – Βάση 10
  - Ψηφία 0, 1, 2, ..., 9
  - $\alpha_5\,\alpha_4\,\alpha_3\,\alpha_2\,\alpha_1\,\alpha_0\,.\,\alpha_{-1}\,\alpha_{-2}\,\alpha_{-3} = \alpha_5 \times 10^5 + \alpha_4 \times 10^4 + \alpha_3 \times 10^3 + \alpha_2 \times 10^2 + \alpha_1 \times 10^1 + \alpha_0 \times 10^0 + \alpha_{-1} \times 10^{-1} + \alpha_{-2} \times 10^{-2} + \alpha_{-3} \times 10^{-3}$
  - $7392 = 7 \times 10^3 + 3 \times 10^2 + 9 \times 10^1 + 2 \times 10^0$
- Δυαδικό σύστημα αρίθμησης – Βάση 2
  - Ψηφία 0, 1
  - $11010.11 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 26.75$
- Το δυαδικό σύστημα αρίθμησης χρησιμοποιείται κατά κανόνα για την αναπαράσταση των αριθμών σε ένα ψηφιακό σύστημα

Δυαδικοί αριθμοί (1)

- $0\,1\,0\,1\,1\,1_2$
  Ποιος είναι ;
- Δεξιότερο ψηφίο
  - «Βάρος» $2^0 = 1$
  - Προς τ' αριστερά διπλασιάζονται τα βάρη
- $0\,1\,0\,1\,1\,1_2$

2^5 2^4 2^3 2^2 2^1 2^0

$16 + 4 + 2 + 1 = 23_{10}$

Δυνάμεις του 2

- $2^0 = 1$
- $2^1 = 2$
- $2^2 = 4$
- $2^3 = 8$
- $2^4 = 16$
- $2^5 = 32$
- $2^6 = 64$
- $2^7 = 128$
- $2^8 = 256$
- $2^9 = 512$
- $2^{10} = 1\,024$
- $2^{11} = 2\,048$

Δυνάμεις του 2

- $2^{12} = 4\,096$
- $2^{13} = 8\,192$
- $2^{14} = 16\,384$
- $2^{15} = 32\,768$
- $2^{16} = 65\,536$
- $2^{17} = 131\,072$
- $2^{18} = 262\,144$
- $2^{19} = 524\,288$
- $2^{20} = 1\,048\,576$
- $2^{21} = 2\,097\,152$
- $2^{22} = 4\,194\,304$
- $2^{23} = 8\,388\,608$

Δυαδικοί αριθμοί (2)

- Ποιοι αριθμοί μπορούν να παρασταθούν με n δυαδικά ψηφία ;
- Όταν οι αριθμοί είναι μόνο θετικοί και 0 τότε μπορούν να παρασταθούν οι αριθμοί από 0 (0....000) μέχρι 2^n-1 (1....111).
  - Για n=2 οι αριθμοί είναι 0, 1, 2, 3
    - 00, 01, 10, 11
  - Για n=3 οι αριθμοί είναι 0, 1, 2, 3, 4, 5, 6, 7
    - 000, 001, 010, 011, 100, 101, 110, 111
  - Για n=4 οι αριθμοί είναι 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,15
    - 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111
  - ......

Δεκαεξαδικό - Οκταδικό

- Δεκαεξαδικό σύστημα – Βάση 16
  - Ψηφία 0, 1, 2, ..., 9, A, B, C, D, E, F
  - Τα A, B, C, D, E, F αντιστοιχούν στο 10, 11, 12, 13, 14, 15
  - 1AF = $1 \times 16^2 + 10 \times 16^1 + 15 \times 16^0$ = 431
- Οκταδικό σύστημα – Βάση 8
  - Ψηφία 0, 1, 2, ..., 7
  - 173 = $1 \times 8^2 + 7 \times 8^1 + 3 \times 8^0$ = 123
- Το δεκαεξαδικό σύστημα είναι πιο δημοφιλές στον κόσμο των υπολογιστών. Γιατί;
- Πολλές φορές χρησιμοποιείται το πρόθεμα “0x” για να υποδηλώσει ένα δεκαεξαδικό αριθμό
  - Π.χ. 0x1AF

Μετατροπή δεκαδικού σε δυαδικό

- Πώς μετατρέπουμε το δεκαδικό αριθμό 41_{10} στο δυαδικό σύστημα
- Συνεχείς διαιρέσεις με το 2
  - 41 ÷ 2 = 20 υπόλοιπο 1 (LSB)
  - 20 ÷ 2 = 10 υπόλοιπο 0
  - 10 ÷ 2 = 5 υπόλοιπο 0
  - 5 ÷ 2 = 2 υπόλοιπο 1
  - 2 ÷ 2 = 1 υπόλοιπο 0
  - 1 ÷ 2 = 0 υπόλοιπο 1 (MSB)

32 8 1

απάντηση 101001 = 41_{10}

- LSB: Least Significant Bit
- MSB: Most Significant Bit

Δυαδικό σε δεκαεξαδικό και αντίστροφα

- Από δεξιά προς τα αριστερά ομαδοποίηση σε 4άδες bit

1	1 0 1 1	1 1 1 1	1 0 1 0 $_2$
1	B	F	A

- Αντίστροφα: κάθε δεκαεξαδικό ψηφίο αντιστοιχεί σε μία 4άδα bit

F 4 E 2_{16}

F	4	E	2
1 1 1 1	0 1 0 0	1 1 1 0	0 0 1 0 $_2$

Δυαδικό και οκταδικό

- Ισχύει το ίδιο όπως και με το δεξαεξαδικό, αλλά όχι με 4άδες δυαδικών ψηφίων αλλά με 3άδες
- Από δεξιά προς τα αριστερά ομαδοποίηση σε 3άδες bit
  - $10\ 011\ 101_2 = 235_8$
- Αντίστροφα: κάθε οκταδικό ψηφίο αντιστοιχεί σε μία 3άδα bit
  - $173_8 = 001\ 111\ 011_2$

Απρόσημοι αριθμοί

- Μέχρι τώρα είδαμε ένα δυαδικό αριθμό των n bit να έχει μόνο θετικές τιμές ή μηδέν
- Μια δυαδική σειρά από n bit που ερμηνεύεται σαν απρόσημος αριθμός (unsigned number) παριστάνει τους αριθμούς από $0_{10} = 0...00_2$ μέχρι $(2^n – 1)_{10} = 1…11_2$
- Παράδειγμα για n=3
  - 000 = 0
  - 001 = 1
  - 010 = 2
  - 011 = 3
  - 100 = 4
  - 101 = 5
  - 110 = 6
  - 111 = 7

Προσημασμένοι αριθμοί

- Πώς παριστάνουμε τους αρνητικούς αριθμούς;
  - Δηλαδή πως μπορεί μια σειρά από δυαδικά ψηφία να ερμηνευτεί σαν προσημασμένος αριθμός (signed number);
- Υπάρχουν 2 αναπαραστάσεις:
  - Πρόσημο και μέγεθος (sign and magnitude)
    - ή προσημασμένο μέγεθος (signed-magnitude)
  - Αναπαραστάσεις συμπληρώματος (complement)
    - Προσημασμένου συμπληρώματος ως προς 1 (1's complement)
    - Προσημασμένου συμπληρώματος ως προς 2 (2's complement)
    - Στην **πράξη** χρησιμοποιείται η αναπαράσταση σε 2's complement
- Και στις τρεις το αριστερότερο bit είναι το πρόσημο
  - 0 σημαίνει + (θετικός)
  - 1 σημαίνει – (αρνητικός)

Προσημασμένο μέγεθος ή συμπλήρωμα ;

- Πώς υλοποιείται η πράξη της πρόσθεσης μεταξύ αριθμών προσημασμένου μεγέθους;
- Για να υπολογίσεις τον αντίθετο ενός αριθμού:
  - Πρόσημο και μέγεθος: αλλάζεις το bit προσήμου
  - Αναπαραστάσεις συμπληρώματος: παίρνεις το συμπλήρωμά του
- Στις αναπαραστάσεις συμπληρώματος:
  - Πιο δύσκολη η διαδικασία υπολογισμού του αντίθετου
  - Πιο εύκολη (όπως θα δούμε στη συνέχεια) η υλοποίηση των πράξεων πρόσθεσης & αφαίρεσης

Δυαδική πρόσθεση

- Θα τη δούμε αναλυτικά σε λίγο αλλά για να κατανοήσουμε τα συμπληρώματα ας κάνουμε ένα παράδειγμα:
  - Να προστεθούν οι αριθμοί:
    $0\,1\,1\,1_2 = 7_{10}$
    $0\,1\,1\,0_2 = 6_{10}$

$$\begin{array}{r} 0\,1\,1\,1 = 7_{10} \\ +\,0\,1\,1\,0 = 6_{10} \\ \hline 1\,1\,0\,1 = 13_{10} \end{array}$$

Συμπληρώματα

- Σε κάθε σύστημα αρίθμησης με βάση r υπάρχουν δύο συμπληρώματα
  - Το συμπλήρωμα ως προς βάση r και
  - Το συμπλήρωμα ως προς ελαττωμένη βάση, ή r-1
- Σε σύστημα με βάση r και n ψηφία το συμπλήρωμα ενός αριθμού N:
- Συμπλήρωμα ως προς βάση : $r^n – N$ (για N≠0)
  - Για N=0, το συμπλήρωμα είναι το ίδιο το 0
  - Άρα, ο αριθμός N και το συμπλήρωμά του έχουν άθροισμα ίσο με r^n
- Συμπλήρωμα ως προς ελαττωμένη βάση : $(r^n – 1) – N$
  - Άρα, ο αριθμός N και το συμπλήρωμά του έχουν άθροισμα ίσο με $r^n – 1$
- Συνεπώς, το συμπλήρωμα ως προς βάση ενός αριθμού N είναι ίσο με το συμπλήρωμα ως προς ελαττωμένη βάση + 1

Συμπληρώματα στο δυαδικό

- Δυαδικός αριθμός των 4 bit
  - $1001_2 = 9_{10}$
- Συμπλήρωμα ως προς 1 (ελαττωμένη βάση), 1's complement
  - $0110_2 = 6_{10}$
  - έχουν άθροισμα $1111_2 = 15_{10}$ δηλαδή $2^4 – 1$
- Συμπλήρωμα ως προς 2 (βάση), 2's complement
  - $0111_2 = 7_{10}$
  - έχουν άθροισμα $10000_2 = 16_{10}$ δηλαδή 2^4

Κανόνες εύρεσης συμπληρωμάτων

- Το συμπλήρωμα ως προς 1 ενός δυαδικού αριθμού το βρίσκουμε:
  - Αντιστρέφοντας κάθε 0 σε 1 και κάθε 1 σε 0
  - 0101 → 1010
  - 1110 → 0001
  - ...
  - Έχουν άθροισμα 1...11 (n bit) δηλαδή $2^n – 1$.
- Το συμπλήρωμα ως προς 2 ενός δυαδικού αριθμού το βρίσκουμε:
  - Ξεκινώντας από το δεξί άκρο του αριθμού και κινούμενοι προς τα αριστερά αφήνουμε όλα τα 0 και το πρώτο 1 αμετάβλητα
  - Στη συνέχεια αντιστρέφουμε όλα τα bit μέχρι το τέλος
  - 010000 → 110000
  - 110100 → 001100
  - 010111 → 101001
  - Έχουν άθροισμα ίσο με 10...00 (n+1 bit) δηλαδή 2^n
- Συνεπώς, το συμπλήρωμα ως προς 2 μπορεί να παραχθεί από το συμπλήρωμα ως προς 1 προσθέτοντας 1

Προσημασμένοι αριθμοί των 4 bit

Δεκαδικός	Συμπλ. προς 2	Συμπλ. προς 1	Πρόσημο/Μέγεθος
+ 7	0111	0111	0111
+ 6	0110	0110	0110
+ 5	0101	0101	0101
+ 4	0100	0100	0100
+ 3	0011	0011	0011
+ 2	0010	0010	0010
+ 1	0001	0001	0001
+ 0	0000	0000	0000
– 0	–	1111	1000
– 1	1111	1110	1001
– 2	1110	1101	1010
– 3	1101	1100	1011
– 4	1100	1011	1100
– 5	1011	1010	1101
– 6	1010	1001	1110
– 7	1001	1000	1111
– 8	1000	–	–

- Παρατηρήσεις:
  - Οι θετικοί αριθμοί είναι ίδιοι και στις τρεις αναπαραστάσεις
  - Υπάρχει +0 και -0 στο συμπλήρωμα ως προς 1 και στο πρόσημο/μέγεθος
  - Στο συμπλήρωμα ως προς 2 υπάρχει ένας παραπάνω αρνητικός από ότι θετικοί. Δεν έχει αντίστοιχο θετικό

Παράδειγμα

- Ποιος είναι ο αριθμός 10101_2 ;
- Είναι προσημασμένος ή όχι;
- Αν είναι προσημασμένος, σε ποια αναπαράσταση από τις τρεις;
- Ας υποθέσουμε ότι είναι προσημασμένος σε συμπλήρωμα ως προς 1
  - Τότε είναι αρνητικός αφού έχει αριστερότερο ψηφίο 1
  - Το συμπλήρωμά του ως προς 1 (δηλαδή ο αντίθετός του) είναι ο 01010_2 δηλαδή $+10_{10}$
  - Άρα ο αρχικός αριθμός είναι το $–10_{10}$
- Ας υποθέσουμε τώρα ότι είναι σε συμπλήρωμα ως προς 2
  - Τότε και πάλι είναι αρνητικός φυσικά
  - Το συμπλήρωμά του ως προς 2 (αντίθετος) είναι ο 01011_2 δηλαδή ο $+11_{10}$
  - Άρα ο αρχικός αριθμός είναι το -11_{10}

Πρόσθεση σε συμπλήρωμα ως προς 2

- Παρατηρείστε ότι:
  - Κάθε αριθμός σε συμπλήρωμα ως προς 2 παράγεται με την πρόσθεση του 1 στον προηγούμενο, αγνοώντας τυχόν κρατούμενα πέραν της n-στής θέσης bit (για σύστημα με n δυαδικά ψηφία)
- Επειδή η πρόσθεση είναι απλά μια επέκταση της απαρίθμησης, για να προσθέσουμε δύο αριθμούς σε συμπλήρωμα ως προς 2:
  - Εκτελούμε δυαδική πρόσθεση
  - Αγνοούμε κάθε κρατούμενο πέραν του MSB
  - Το αποτέλεσμα θα είναι πάντα το σωστό άθροισμα
- Παραδείγματα:

+3	0011	+6	0110	- 2	1110	+4	0100
+ +4	+ 0100	+ - 3	+ 1101	+ - 6	+ 1010	+ - 7	+ 1001
+7	0111	+3	1 0011	- 8	1 1000	-3	1101

Υπερχείλιση (overflow)

- Αν μια πράξη πρόσθεσης δώσει αποτέλεσμα που υπερβαίνει το πεδίο τιμών, τότε έχουμε υπερχείλιση (overflow)
- Πότε μπορεί να συμβεί;
  - Όταν προσθέτουμε ομόσημους αριθμούς ή αφαιρούμε ετερόσημους
- Πότε *δεν* μπορεί να συμβεί;
  - Όταν προσθέτουμε ετερόσημους ή αφαιρούμε ομόσημους *δεν* μπορεί να εμφανιστεί υπερχείλιση

Αναγνώριση υπερχείλισης

- Στους απρόσημους αριθμούς η υπερχείλιση αναγνωρίζεται με την εμφάνιση κρατουμένου
- Παράδειγμα: 10 + 7 σε απρόσημους των 4 bit

```
   1010
 + 0111
 ------
 1 0001
```

- Το κρατούμενο που δημιουργήθηκε δείχνει ότι υπάρχει υπερχείλιση
- Το σωστό αποτέλεσμα της πράξης είναι το 17, που όμως δεν χωράει σε 4 bit
- Με 4 bit μπορούμε να παραστήσουμε απρόσημους αριθμούς από το 0 μέχρι και το 15

Αναγνώριση υπερχείλισης (συν.)

- Στους προσημασμένους αριθμούς σε συμπλήρωμα η εμφάνιση κρατουμένου *δεν* σχετίζεται με την υπερχείλιση
- Κανόνας: Μια πρόσθεση εμφανίζει υπερχείλιση όταν τα πρόσημα των προσθετέων είναι ίδια και το πρόσημο του αθροίσματος διαφορετικό
- Παραδείγματα (σε συμπλήρωμα ως προς 2):

```
   - 3     1101              +5     0101
 + - 6   + 1010            + +6   + 0110
 -----------------         ---------------
   - 9   1 0111 = +7        +11     1011 = -5
```

Αφαίρεση σε συμπλήρωμα ως προς 2

- Αντί να κάνουμε αφαίρεση κάνουμε πρόσθεση με το συμπλήρωμα
  - Ιδέα: οι αρνητικοί αριθμοί παριστάνονται με συμπληρώματα
- Να γίνει η αφαίρεση X – Y
  - Υπολογίζουμε το συμπλήρωμα ως προς 2 του Y
  - Προσθέτουμε X + συμπλήρωμα ως προς 2 του Y
- Παράδειγμα: αφαίρεση X=7 (0111) μείον Y=3 (0011)
  - Συμπλήρωμα ως προς 2 του Y : 1101
  - X συν συμπλήρωμα του Y : 0111 + 1101 = 1 0100 (4)
    - Το 5^{ο} bit στις πράξεις συμπληρώματος ως προς 2 αγνοείται

Αφαίρεση σε συμπλήρωμα ως προς 2 (συν.)

- Υλοποίηση:
  - Υπολογίζουμε το συμπλήρωμα ως προς 1 του Y
  - Προσθέτουμε X + συμπλήρωμα ως προς 1 του Y + 1 (αρχικό κρατούμενο Cin =1)
  - Τι πλεονέκτημα έχει αυτή η υλοποίηση;
- Παραδείγματα:

		1 (Cin)
+4	0100	0100
- +3	- 0011	+ 1100
+1		1 0001

		1 (Cin)
+3	0011	0011
- +4	- 0100	+ 1011
- 1		1111

		1 (Cin)
+7	0111	0111
- - 3	- 1101	+ 0010
+10		0 1010

Υπερχείλιση

Κώδικας BCD – Binary Coded Decimal

- Δυαδικά κωδικοποιημένοι δεκαδικοί αριθμοί
- Κάθε ψηφίο του κώδικα BCD παριστάνει ένα δεκαδικό ψηφίο, δηλαδή από το 0 μέχρι το 9
  - 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001
  - Οι υπόλοιποι 6 συνδυασμοί τεσσάρων δυαδικών ψηφίων είναι άκυροι για τον κώδικα BCD, δηλαδή δεν πρέπει να εμφανίζονται
- Πώς παριστάνονται οι δεκαδικοί αριθμοί πολλών ψηφίων στον BCD;
  - $10_{10} = 0001\ 0000_{BCD} = 1010_2$
  - $168_{10} = 0001\ 0110\ 1000_{BCD}$
  - $794_{10} = 0111\ 1001\ 0100_{BCD}$
- Χρειάζονται περισσότερα bit απ' ότι στο δυαδικό σύστημα
- Π.χ. ο δεκαδικός 168_{10} είναι:
  - $0001\ 0110\ 1000_{BCD}$ (12 bit)
  - 10101000_2 (8 bit)

Πρόσθεση BCD

- Να γίνει η πρόσθεση 170 + 239 σε BCD

$170_{10} = \quad 0001\ 0111\ 0000_{BCD}$

$239_{10} = \quad 0010\ 0011\ 1001_{BCD}$

1 κρατούμενο

$0100\ 0000\ 1001_{BCD}$

- Διαφέρει από τη δυαδική πρόσθεση
- Αργότερα θα δούμε κυκλώματα που εκτελούν
  - Δυαδική πρόσθεση
  - Πρόσθεση BCD

Κώδικας Gray

- Ιδιότητα : κάθε κωδική λέξη (δυαδική σειρά) διαφέρει από την επόμενη και την προηγούμενή της μόνο σε 1 bit
  - Θα τον χρησιμοποιήσουμε στο χάρτη Karnaugh για την ελαχιστοποίηση λογικών συναρτήσεων
- Κώδικας Gray των 2 bit

0 0	= 0
0 1	= 1
1 1	= 2
1 0	= 3

- Κώδικας Gray των 3 bit

0 0 0	= 0
0 0 1	= 1
0 1 1	= 2
0 1 0	= 3
1 1 0	= 4
1 1 1	= 5
1 0 1	= 6
1 0 0	= 7

- Κώδικας Gray των 4 bit

0 0 0 0	= 0
0 0 0 1	= 1
0 0 1 1	= 2
0 0 1 0	= 3
0 1 1 0	= 4
0 1 1 1	= 5
0 1 0 1	= 6
0 1 0 0	= 7
1 1 0 0	= 8
1 1 0 1	= 9
1 1 1 1	= 10
1 1 1 0	= 11
1 0 1 0	= 12
1 0 1 1	= 13
1 0 0 1	= 14
1 0 0 0	= 15

Κώδικας ASCII

- American Standard Code for Information Interchange
- Απεικόνιση χαρακτήρων και συμβόλων εσωτερικά σε έναν υπολογιστή
  - Πρέπει να είναι πρότυπος κώδικας ώστε να υπάρχει συμβατότητα μεταξύ διαφορετικών υπολογιστικών συστημάτων
  - Είναι κώδικας των 7 bit αλλά επειδή οι υπολογιστές χειρίζονται byte των 8 bit το $8^{ο}$ bit χρησιμοποιείται για να δείξει άλλους χαρακτήρες πέρα από τα γράμματα τους αριθμούς και τα σημεία στίξης

Κώδικας ASCII (συν.)

				b_7 b_6 b_5 →	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
Bits b_4 ↓	b_3 ↓	b_2 ↓	b_1 ↓	Column → / Row ↓	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FC	,	<	L	\	l	|
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Κώδικας ισοτιμίας (parity code)

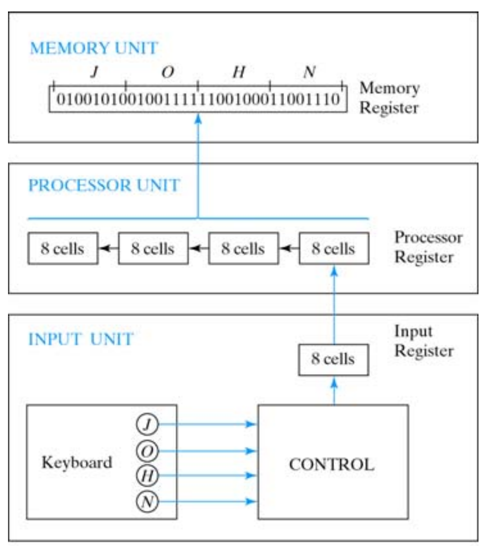
- Σε μια λέξη των k bit προστίθεται άλλο ένα bit που ονομάζεται bit ισοτιμίας (parity bit)
  - Άρτια ισοτιμία (even parity) σημαίνει ότι τα k+1 bit έχουν συνολικά άρτιο πλήθος άσσων (μονάδων)
  - Περιττή ισοτιμία (odd parity) σημαίνει ότι τα k+1 bit έχουν συνολικά περιττό πλήθος άσσων (μονάδων)
- Παράδειγμα (λέξη των 7 bit)
  - 1000001 με bit άρτιας ισοτιμίας γίνεται 01000001
  - 1000100 με bit περιττής ισοτιμίας γίνεται 11000100

Δυαδική αποθήκευση - Καταχωρητές

- Δυαδικό κύτταρο ή κελί (binary cell)
  - Όποια συσκευή έχει 2 καταστάσεις και μπορεί να αποθηκεύσει ένα δυαδικό ψηφίο, δηλαδή 0 ή 1
- Καταχωρητής (register)
  - Μια ομάδα k δυαδικών κυττάρων που αποθηκεύει k bit
- Θα δούμε σε επόμενα κεφάλαια πως σχεδιάζονται τα κυκλώματα που αποθηκεύουν δυαδικούς αριθμούς
  - Δηλαδή οι καταχωρητές

Μεταφορά σε επίπεδο καταχωρητών

- Λειτουργίες σε ένα ψηφιακό σύστημα
  - Επεξεργασία δυαδικής πληροφορίας
  - Αποθήκευση δυαδικής πληροφορίας
- Βασικά μέρη ενός ψηφιακού υπολογιστή (σχήμα):
  - Είσοδος (και έξοδος)
  - Επεξεργαστής
  - Μονάδα μνήμης
- Παράδειγμα:
  - ανάγνωσης χαρακτήρα ASCII από πληκτρολόγιο και μεταφορά του στη μνήμη



Επεξεργασία δυαδικής πληροφορίας

- Πρόσθεση δύο αριθμών από τη μνήμη και αποθήκευση του αποτελέσματος πάλι στη μνήμη
- Μέσα στον επεξεργαστή βρίσκονται:
  - οι καταχωρητές
  - ο αθροιστής, δηλαδή το λογικό κύκλωμα που προσθέτει αριθμούς
- Τα κελιά που αποθηκεύουν bit στη μνήμη είναι διαφορετικά από αυτά των καταχωρητών του επεξεργαστή