ΗΥ-120: Ψηφιακή Σχεδίαση
Φθινόπωρο 2005

Τμ. Επ. Υπολογιστών

Εργαστήριο 1:
Λογική με Διακόπτες

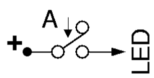
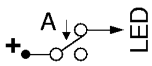
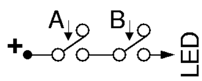
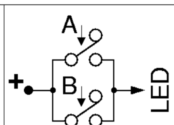
3 έως 7 Οκτωβρίου 2005 (βδομάδα 2)

[Βιβλίο: *προαιρετικά* μπορείτε να διαβάσετε τις παραγράφους **1.1, 1.2** (σελ. 1-7) του βιβλίου (Wakerly)].

1.1 Λογικές Πράξεις και Πίνακας Αληθείας

Το εργαστήριο αυτό αποτελεί συνέχεια και εμβάθυνση του εργ. 0, και ιδιαίτερα του πώς κυκλώματα διακοπτών υλοποιούν τις λογικές πράξεις **ΟΧΙ** (§0.6), **ΚΑΙ** (§0.7), **Ή** (§0.8). Αν δεν προλάβατε να κάνετε τα αντίστοιχα πειράματα την προηγούμενη βδομάδα, ή να τα καταλάβετε σε βάθος, ξεκινήστε σήμερα με εκείνα, ή μελετήστε πρώτα εκείνες τις σημειώσεις γιά να τις καταλάβετε σε βάθος --αποτελούν τη βάση γιά όλο το μάθημα.

Ο πίνακας που ακολουθεί αποτελεί μιά περίληψη των παραπάνω παραγράφων 0.6 - 0.8. Γιά πληρότητα, προσθέσαμε και το κύκλωμα της §0.5 με το όνομα "ταυτότητα" (identity), αν και συνήθως δεν θεωρούμε ενδιαφέρουσα αυτήν την τόσο απλή λογική πράξη. Στον πίνακα αυτόν χρησιμοποιούμε τον όρο "OFF" όταν μεν πρόκειται γιά διακόπτες γιά να σημαίνει "ελεύθερος" (όχι πατημένος), όταν δε πρόκειται γιά LED γιά να σημαίνει "σβηστή"· αντίστροφα, ο όρος "ON" γιά μεν τους διακόπτες σημαίνει "πατημένος" γιά δε τις LED σημαίνει "σβηστή".

Διακοπτης **A**	Διακοπτης **B**	 **IDENTITY** (ταυτοτητα)	 **NOT** (οχι)	 **AND** (και)	 **OR** (ή)
OFF (ελευθερος)	OFF	OFF (σβηστη)	ON (αναμενη)	OFF	OFF
	ON			OFF	ON
ON (πατημενος)	OFF	ON (αναμενη)	OFF (σβηστη)	OFF	ON
	ON			ON	ON

Ο πίνακας αυτός δίνει αναλυτικά τη συμπεριφορά των κυκλωμάτων και των αντίστοιχων "Λογικών Συναρτήσεων" (πράξεων) γιά τις διάφορες περιπτώσεις εισόδων τους, δηλαδή κατάστασης των διακοπτών· τέτοιους πίνακες τους λέμε *Πίνακες Αληθείας (truth tables)*. Γιά το κύκλωμα της ταυτότητας, ο πίνακας αληθείας μας λέει ότι η έξοδός του (φωτοβολία της LED) βρίσκεται πάντα σε ευθεία αντιστοιχία με την είσοδό του (κατάσταση του διακόπτη Α): OFF γιά OFF και ON γιά ON. Γιά τη λογική πράξη ΟΧΙ (NOT), ο πίνακας μας δείχνει ότι η έξοδός της είναι πάντα το αντίστροφο (ανάποδο) της εισόδου της Α, εξ' ου και το όνομα του κυκλώματος αυτού που συχνά λέγεται "αντιστροφέας" (inverter). Γιά την λογική πράξη ΚΑΙ (AND), η έξοδός της είναι συνάρτηση των δύο εισόδων της, Α και Β: είναι ON μόνον στον έναν από τους 4 συνδυασμούς τιμών των Α και Β, όταν Α και Β είναι ON· αυτό προκύπτει όταν οι διακόπτες είναι συνδεδεμένοι "εν σειρά". Τέλος, η λογική συνάρτηση Ή (OR) είναι ON όταν Α είναι ON ή Β είναι ON (ή και οι δύο είναι ON)· αυτό προκύπτει όταν οι διακόπτες είναι συνδεδεμένοι "εν παραλλήλω".

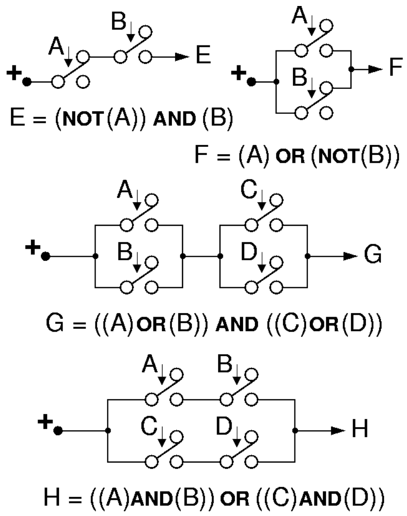
Οι λογικές πράξεις **AND** και **OR** είναι ανάλογες με πλήθος παρόμοιων εννοιών της καθημερινής μας ζωής, π.χ.:

- Γιά να τρέξει νερό από τη βρύση πρέπει να ανοίξουμε τη βρύση **και** να είναι ανοικτός και ο γενικός διακόπτης νερού του διαμερίσματος.
- Θα υπάρχει υπερκατανάλωση νερού αν έχουμε διαρροή στο καζανάκι **ή** αν στάζει η βρύση του μπάνιου **ή** της κουζίνας (ή και περισσότερα από ένα από αυτά ταυτόχρονα).
- Γιά να μπώ στο αυτοκίνητό μου που βρίσκεται στο γκαράζ πρέπει να ανοίξω την πόρτα του

γκαράζ **και** την πόρτα του αυτοκινήτου.

- Μπορώ να μπώ στο εξοχικό μου σπίτι από την κύρια πόρτα **ή** από την πίσω πόρτα του κήπου.
- Από τους περισσότερους κοινούς λογαριασμούς τράπεζας μπορεί να κάνει ανάληψη ο δικαιούχος Α **ή** ο δικαιούχος Β. Υπάρχουν όμως και λογαριασμοί, π.χ. εταιρειών, όπου γιά να γίνει ανάληψη πρέπει να υπογράψουν π.χ. **και** ο διευθυντής **και** ο ταμίας.

1.2 Κυκλώματα γιά Σύνθετες Λογικές Πράξεις



Οι παραπάνω βασικές συνδεσμολογίες διακοπτών μπορούν να συνδυαστούν μεταξύ τους σε οσοδήποτε σύνθετους συνδυασμούς· στο σχήμα δεξιά φαίνονται μερικά παραδείγματα (στα σημεία E, F, G, H, υποτίθεται ότι συνδέονται LED's μεσω αντιστάσεων). Στο πρώτο κύκλωμα, γιά να περάσει ρεύμα προς το E, πρέπει "όχι A πατημένος", επειδή η σύνδεση έγινε στον πάνω δεξιά ακροδέκτη (T0) του A, **ΚΑΙ** (σύνδεση εν σειρά) "B πατημένος", επειδή το E συνδέεται στον κάτω δεξιά ακροδέκτη (T1) του B. Στο δεύτερο κύκλωμα, θα περνάει ρεύμα προς το F όποτε "A πατημένος" (σύνδεση κάτω δεξιά (T1) του A), **Ή** (σύνδεση εν παραλλήλω) "όχι B πατημένος" (σύνδεση πάνω δεξιά (T0) του B). Στο τρίτο κύκλωμα, γιά να περάσει ρεύμα προς το G πρέπει να υπάρχει δρόμος μέσω του αριστερού ζεύγους διακοπτών **ΚΑΙ** μέσω του δεξιού ζεύγους· από το αριστερό ζεύγος μπορεί να περάσει ρεύμα όποτε "A πατημένος" **Ή** "B πατημένος"· ομοίως από το δεξί όποτε "C πατημένος" **Ή** "D πατημένος". Τέλος, στο κύκλωμα που τροφοδοτεί το H, ρεύμα μπορεί να περάσει από το επάνω ζευγάρι διακοπτών **Ή** από το κάτω· γιά να περάσει από επάνω πρέπει "A πατημένος" **ΚΑΙ** "B πατημένος", αντίστοιχα δε από κάτω.

<u>Πειράματα 1.3:</u> Λογική με Διακόπτες

Προετοιμασία <u>πριν</u> φτάσετε στο Εργαστήριο:
Σε αυτό και σε όλα τα υπόλοιπα εργαστήρια από 'δώ και μπρος, **πριν** φτάσετε στο εργαστηριακό σας τμήμα, θα έχετε διαβάσει λεπτομερώς και προσεκτικά ολόκληρη την εκφώνηση της άσκησης, και θα έχετε ετοιμάσει **γραπτά**, καθαρά, και λεπτομερώς τα πλήρη σχεδιαγράμματα όλων των κυκλωμάτων που σας ζητούνται στην εκφώνηση, καθώς και όλες τις απαντήσεις στα ερωτήματα που σας θέτει η εκφώνηση. Την γραπτή αυτή προεργασία θα την εξετάζει ο βοηθός σας στην αρχή του εργαστηρίου, και θα του την παραδίδετε στο τέλος, από αυτήν δε θα προκύπτει ένα σεβαστό ποσοστό του βαθμού του εργαστηρίου σας. Ένα άλλο σεβαστό ποσοστό θα αντανακλά την από μέρους σας κατανόηση της θεωρίας του μαθήματος που σχετίζεται με το εργαστήριο, όπως αυτή φαίνεται από τις απαντήσεις σας στις σχετικές ερωτήσεις του βοηθού.

Σε αυτό εδώ το πείραμα, **πριν** φτάσετε στο εργαστήριο:

i. Σχεδιάστε τα παρακάτω κυκλώματα που σας ζητώνται. Οι διακόπτες A, B, C είναι όπως αυτοί που είδαμε στην § <u>0.5</u>, και το κάθε κύκλωμα οδηγεί μιά LED μέσω μιάς αντίστασης όπως στην § <u>0.4</u>.
ii. Φτιάξτε τον πίνακα αληθείας του κάθε κυκλώματος, όπως είδαμε παραπάνω στην § <u>1.1</u>. Αφού τα κυκλώματα αυτά έχουν 3 εισόδους (A, B, C), οι πίνακες αληθείας τους θα έχουν 8 γραμμές: μία τετράδα γιά A=OFF και μία τετράδα γιά A=ON· σε κάθε τετράδα, θα υπάρχει ένα ζευγάρι γιά B=OFF και ένα γιά B=ON· και σε κάθε ζευγάρι, θα υπάρχει μία γραμμή γιά C=OFF και μία γιά C=ON.
iii. Διατυπώστε γραπτώς τη συνθήκη γιά την άρνηση της κάθε εξόδου, δηλαδή γιά το πότε η φωτοδίοδος θα είναι σβηστή ή το πότε (1) η μπανιέρα δεν γεμίζει, ή (2) δεν μπορώ να μπώ στο σπίτι, ή (3) δεν μπορώ να φύγω με το αυτοκίνητο, ή (4) δεν θα πάω το αυτοκίνητο στο συνεργείο. Διατυπώστε τη συνθήκη με λόγια, χρησιμοποιώντας την καθημερινή μας λογική, και δείτε ότι εφαρμόζεται πάλι η αρχή του δυϊσμού των παραγράφων <u>0.7</u> και <u>0.8</u>· διασταυρώστε την ορθότητα της συνθήκης με τον πίνακα αληθείας.

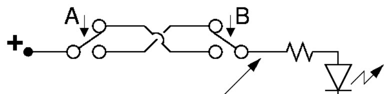
Στο εργαστήριο, κατασκευάστε τα κυκλώματα, ελέγξτε τα, και δείξτε τα στο βοηθό. Γιά να ελεγχθεί πλήρως το κάθε κύκλωμα πρέπει να τού εφαρμόσετε καθέναν από τους 8 συνδυασμούς εισόδων που έχει ο πίνακας αληθείας, και να διαπιστώσετε ότι η LED κάνει το σωστό γιά καθέναν· καθώς τα ελέγχετε, διασταυρώστε την ορθότητα της συνθήκης γιά την άρνηση της κάθε εξόδου.

1. Η LED να ανάβει όταν **θα γεμίσει η μπανιέρα**, δηλαδή όταν:
    - Έχω κλείσει την τάπα (διακόπτης A πατημένος), **ΚΑΙ**
    - ανοίγω τη βρύση του κρύου (διακόπτης B πατημένος) **Ή** ανοίγω αυτήν του ζεστού (διακόπτης C πατημένος).
2. Η LED να ανάβει όταν **μπορώ να μπώ στο σπίτι**, δηλαδή όταν:
    - έχω το κλειδί της κεντρικής πόρτας (A πατημένος), **Ή**
    - έχω το κλειδί της μπαλκονόπορτας (B πατημένος) **ΚΑΙ** το πατζούρι της μπαλκονόπορτας είναι ανοικτό (C πατημένος).
3. Η LED να ανάβει όταν **μπορώ να φύγω με το αυτοκίνητο**, δηλαδή όταν:
    - έχω το κλειδί του γκαράζ (A πατημένος), **ΚΑΙ**
    - έχω το κλειδί του αυτοκινήτου (B πατημένος), **ΚΑΙ**
    - **ΔΕΝ** υπάρχει βλάβη στη μηχανή.

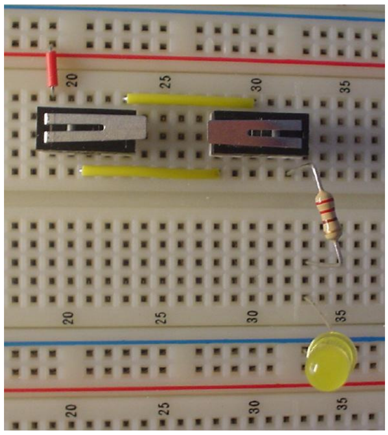
    Όποτε υπάρχει βλάβη στη μηχανή, θα πατιέται ο διακόπτης C.
4. Η LED να ανάβει όταν **πρέπει να πάω το αυτοκίνητο στο συνεργείο**, δηλαδή όταν:
    - υπάρχει βλάβη στη μηχανή (A πατημένος), **Ή**
    - πέρασαν 12 μήνες από το προηγούμενο service (B πατημένος) **ΚΑΙ** **ΔΕΝ** έχω μείνει "πανί-με-πανί".

    Όποτε έχω μείνει "πανί-με-πανί", θα πατιέται ο διακόπτης C.

Πείραμα 1.4: Διακόπτες "Aller-Retour" - Αποκλειστικό Ή, Έλεγχος Ισότητας

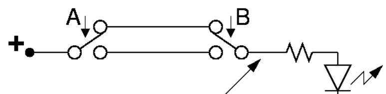


Exclusive OR (odd parity)

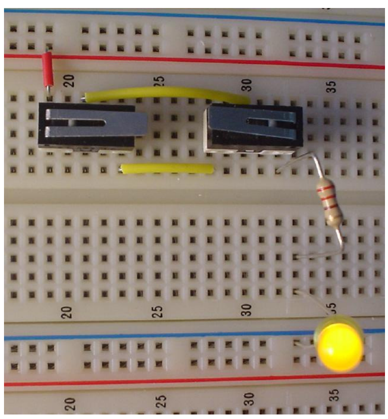


Παρατηρήστε ότι σε όλα τα παραπάνω κυκλώματα υπάρχουν ορισμένες καταστάσεις μερικών από τις εισόδους οι οποίες κάνουν την έξοδο "αναίσθητη" στις (ανεξάρτητη από τις) άλλες εισόδους. Γιά παράδειγμα, στο κύκλωμα ΚΑΙ, όταν ο ένας διακόπτης είναι ελεύθερος, η LED παραμένει σβηστή ό,τι και να κάνει ο άλλος διακόπτης· στο κύκλωμα Ή, όταν ένας διακόπτης είναι πατημένος, η LED ανάβει ό,τι και να κάνει ο άλλος διακόπτης. Σκεφτείτε τώρα τις κρεββατοκάμαρες των σπιτιών, όπου οι διακόπτες γιά τα φώτα είναι συνήθως τύπου "aller-retour", δηλαδή σε όποια κατάσταση και να έχει μείνει ο ένας διακόπτης (π.χ. της πόρτας), ο άλλος διακόπτης (π.χ. του κρεββατιού) μπορεί πάντα να αλλάξει την κατάσταση του φωτός (να το ανάψει ή να το σβήσει). Αυτό γίνεται με ένα από τα δύο κυκλώματα που φαίνονται δεξιά.

Το πρώτο κύκλωμα υλοποιεί τη λογική πράξη "**αποκλειστικό Ή**" (exclusive OR - XOR), διότι η LED ανάβει όταν είναι πατημένος *αποκλειστικά ο ένας* από τους δύο διακόπτες και *όχι και οι δύο μαζί*. Παρατηρήστε το κύκλωμα (όπου τα δύο χιαστί σύρματα διασταυρώνονται χωρίς να κάνουν επαφή μεταξύ τους): ρεύμα μπορεί να περάσει προς την LED είτε όταν ο A είναι πατημένος και ο B δεν είναι (σύρμα από κάτω αριστερά προς πάνω δεξιά), είτε όταν ο A δεν είναι πατημένος και ο B είναι (σύρμα από πάνω αριστερά προς κάτω δεξιά). Επομένως, η LED ανάβει όταν ((A)**ΚΑΙ**(**ΟΧΙ**(B))) **Ή** ((**ΟΧΙ**(A))**ΚΑΙ**(B)), δηλαδή είναι πατημένος *(μόνο ο A)* ή *(μόνο ο B)*.



Equality check (even parity)



Το δεύτερο κύκλωμα υλοποιεί τη λογική πράξη **ελέγχου ισότητας** (equality check), διότι η LED ανάβει μόνον όταν οι δύο διακόπτες είναι στην ίδια (ίση) κατάσταση --και οι δύο πατημένοι ή και οι δύο ελεύθεροι. Ρεύμα μπορεί να περάσει προς την LED είτε από το επάνω σύρμα (και οι δύο διακόπτες ελευθεροι) είτε από το κάτω σύρμα (και οι δύο διακόπτες πατημένοι), επομένως η LED ανάβει όταν ((**ΟΧΙ**(A))**ΚΑΙ**(**ΟΧΙ**(B))) **Ή** ((A)**ΚΑΙ**(B)), δηλαδή όταν A και B είναι *(και οι δύο OFF)* ή *(και οι δύο ON)*.

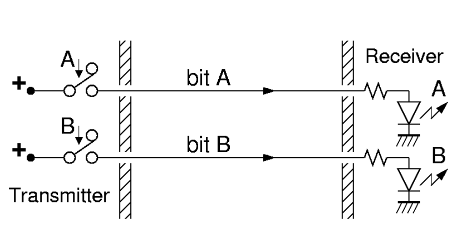
Πριν φτάσετε στο εργαστήριο, γράψτε τον πίνακα αληθείας των εξόδων των δύο κυκλωμάτων. Παρατηρήστε ότι, ανεξαρτήτως της κατάστασης του ενός διακόπτη, ο άλλος μπορεί πάντα, ανοιγοκλείνοντας, να αναβοσβήσει το φώς. **Στο εργαστήριο**, φτιάξτε τα δύο κυκλώματα, δείξτε τα στο βοηθό σας, και επαληθεύστε πειραματικά τις παραπάνω ιδιότητες.

Περιττή και Άρτια Ισοτιμία (Odd and Even Parity) - *γιά όσους ενδιαφέρονται περισσότερο:* η γενίκευση της συνάρτησης αποκλειστικού-Ή σε περισσότερες εισόδους είναι η *περιττή ισοτιμία* (odd parity), η δε γενίκευση της συνάρτησης ελέγχου ισότητας είναι η *άρτια ισοτιμία* (even parity). Όταν έχουμε πολλές εισόδους --πολλούς διακόπτες-- μετράμε το πλήθος τους που είναι πατημένοι (ON) σε δεδομένη στιγμή. Εάν το πλήθος αυτό είναι αριθμός περιττός (μονός - μη ακέραιο πολλαπλάσιο του 2), τότε λέμε ότι έχουμε περιττή ισοτιμία, και η αντίστοιχη συνάρτηση είναι ON (αναμένη), αλλοιώς η συνάρτηση αυτή είναι OFF (σβηστή). Αντίστροφα, όταν το πλήθος των εισόδων που είναι ON είναι αριθμός άρτιος (ζυγός - ακέραιο πολλαπλάσιο του 2), η συνάρτηση άρτιας ισοτιμίας είναι ON, αλλοιώς είναι OFF. Επομένως, οι συναρτήσεις περιττής και άρτιας ισοτιμίας είναι πάντα η μία το αντίστροφο (το λογικό ΌΧΙ) της άλλης (άρα αρκεί να ξέρουμε τη μία τους γιά να βρίσκουμε άμεσα και την άλλη).

Εάν ένας από τους διακόπτες αλλάξει κατάσταση, το πλήθος των εισόδων που είναι ON αλλάζει κατά ένα (+1 ή -1): αν ο διακόπτης που άλλαξε ήταν σβηστός (OFF) και άναψε (ON), το πλήθος αυξήθηκε κατά 1, ενώ αν ήταν αναμένος και έσβησε τότε το πλήθος μειώθηκε κατά 1. Ο,τιδήποτε από τα δύο και να συμβαίνει, η ισοτιμία αλλάζει από άρτια σε περιττή ή από περιττή σε άρτια! Βλέπουμε λοιπόν ότι διατηρείται η βασική ιδιότητα με την οποία ξεκινήσαμε: σε οιαδήποτε κατάσταση και να βρίσκονται οι διακόπτες, αρκεί οιοσδήποτε ένας από αυτούς να αλλάξει κατάσταση γιά να αλλάξει τιμή η έξοδος (από σβηστή να ανάψει ή από αναμένη να σβήσει). Η ιδιότητα αυτή αποτελεί τη βάση γιά την κύρια εφαρμογή των συναρτήσεων ισοτιμίας:

Οι συναρτήσεις ισοτιμίας χρησιμοποιούνται σαν η απλούστερη μορφή **κώδικα ανίχνευσης σφαλμάτων** (error detection codes): αφού μεταδώσουμε μέσα από ένα τηλεπικοινωνιακό δίκτυο κάμποσες πληροφορίες (κάμποσα σύρματα, καθένα "αναμένο" ή "σβηστό"), μεταδίδουμε στο τέλος ακόμα μία επιπλέον πληροφορία που είναι π.χ. η άρτια ισοτιμία όλων των προηγουμένων. Συνήθως, οι πληροφορίες που μεταδίδουμε φτάνουν στην άλλη ακρή όλες σωστές· σε σπάνιες περιπτώσεις, λόγω θορύβου, μία από τις πληροφορίες μπορεί να φτάσει λάθος (ON αντί OFF, ή OFF αντί ON)· σε πολύ σπανιότερες περιπτώσεις μπορεί δύο η περισσότερες πληροφορίες να φτάσουν λάθος (π.χ. αν η πιθανότητα ενός λάθους είναι μία στις χίλιες, και αν τα λάθη είναι ανεξάρτητα μεταξύ τους, τότε η πιθανότητα δύο λαθών είναι περίπου μία στο εκατομμύριο (περίπου το τετράγωνο της πιθανότητας ενός λάθους)). Εάν συμβεί ένα λάθος στη μετάδοση, τότε θα αλλάξει η ισοτιμία των πληροφοριών που στείλαμε, ακριβώς λόγω της παραπάνω ιδιότητας: οιεσδήποτε και αν είναι οι πληροφορίες, η αλλαγή μίας οιασδήποτε από αυτές αλλάζει την ισοτιμία. Αν όμως αλλάξει η ισοτιμία, θα το καταλάβουμε, διότι η επιπλέον πληροφορία ισοτιμίας που στείλαμε δεν θα συμπίπτει πλέον με την ισοτιμία που βλέπει ο παραλήπτης! Έτσι επιτυγχάνεται ο στόχος της ανίχνευσης σφαλμάτων στις περιπτώσεις που συμβαίνει μόνο ένα σφάλμα, που είναι και οι πιό συχνές. Άπαξ και διαπιστωθεί η ύπαρξη σφάλματος, η διόρθωση του μπορεί να γίνει π.χ. με μία αίτηση αναμετάδοσης ("ξαναπές το --δεν άκουσα καλά"). Ο κύριος περιορισμός των συναρτήσεων ισοτιμίας στην ανίχνευση σφαλμάτων είναι ότι αυτές ανιχνεύουν μόνο 1, 3, 5, κλπ. σφάλματα, ενώ τους διαφεύγουν 2, 4, 6, κλπ. σφάλματα. Εάν τα σφάλματα είναι ανεξάρτητα μεταξύ τους, και εάν η πιθανότητα ενός σφάλματος είναι πολύ μικρή, τότε 2 ή περισσότερα "μαζεμένα" σφάλματα είναι πολύ σπάνια. Εάν όμως τα σφάλματα είναι συχνά ή δεν είναι ανεξάρτητα μεταξύ τους, τότε χρειαζόμαστε άλλους, πιό πολύπλοκους κώδικες γιά την ανίχνευσή τους· τέτοια περίπτωση "ομοβροντίας (εκρηκτικών) σφαλμάτων" (burst errors) έχουμε π.χ. όταν ένα κινητό τηλέφωνο, κινούμενο, περνάει γιά λίγο πίσω από μία "σκιά" της κεραίας του σταθμού βάσης· αρκετά όμως είπαμε γιά τώρα --γιά περισσότερες πληροφορίες θα πρέπει να πάρετε κάποιο μάθημα τηλεπικοινωνιών και κωδικοποίησης....

1.5 Τα Bits και η Κωδικοποίηση Μηνυμάτων



Κυκλώματα σαν αυτά που συζητάμε --*ψηφιακά,* όπως θα τα πούμε παρακατω-- χρησιμοποιούνται γιά την αποθήκευση, επεξεργασία, και αποστολή κωδικοποιημένων μηνυμάτων και πληροφοριών. Στο σχήμα φαίνεται ένα απλό σύστημα μετάδοσης πληροφοριών από έναν πομπό (transmitter) σε έναν δέκτη (receiver)· έχει δύο σύρματα, καθένα από τα οποία μπορεί να διαρρέεται ή όχι από ρεύμα.

Πληροφορία, εδώ, είναι το αν υπάρχει ή δεν υπάρχει ρεύμα στο κάθε σύρμα, πράγμα που ο πομπός το ρυθμίζει με τη θέση του κάθε διακόπτη, και ο δέκτης το καταλαβαίνει από το αν ανάβει ή όχι η κάθε LED. Κάθε σύρμα μεταδίδει μιά *ποσότητα πληροφορίας* ικανή να επιλέξει *ένα από δύο* πράγματα, καταστάσεις, ή μηνύματα. Γιά το λόγο αυτό, την ποσότητα αυτή πληροφορίας την ονομάζουμε *δυαδικό ψηφίο (binary digit)* ή **δυφίο (bit)** --όνομα που αποτελεί τον πιό γνωστό όρο της Επιστήμης Υπολογιστών! Το σύστημα του σχήματος μεταδίδει δύο bits πληροφορίας από τον πομπό στο δέκτη, το bit A και το bit B.

Κάθε bit μεταδιδόμενης πληροφορίας μπορεί να έχει δύο μόνο διαφορετικές τιμές· από ηλεκτρική άποψη αυτές είναι απουσία ή παρουσία ρεύματος γιά τα μέχρι στιγμής κυκλώματά μας, ή χαμηλή ή ψηλή ηλεκτρική τάση γιά τα συνηθισμένα chips τύπου CMOS των σημερινών μικροηλεκτρονικών συστημάτων. Η **ερμηνεία** όμως που δίδεται στις δύο αυτές διαφορετικές τιμές είναι θέμα *σύμβασης* (συμφωνίας) ανάμεσα στον πομπό και το δέκτη, και είναι εντελώς *αυθαίρετη*, καθώς και *ανεξάρτητη* από τις δύο ηλεκτρικές τιμές του bit· πολές φορές μάλιστα, η ερμηνεία αυτή είναι σκόπιμα κρυφή και μη αυτονόητη ώστε να αποφευχθούν υποκλοπές του μηνύματος, όπως π.χ. όταν χρησιμοποιούνται κώδικες κρυπτογραφίας. Προφανώς, γιά να υπάρξει επιτυχής επικοινωνία, πρέπει ο πομπός και ο δέκτης να έχουν προσυμφωνήσει στην ίδια ερμηνεία των ηλεκτρικών τιμών· πέραν των συμμετεχόντων στην επικοινωνία, όμως, δεν είναι ανάγκη κανείς άλλος να ξέρει ή να έχει συμφωνήσει με αυτή την ερμηνεία. Ο επόμενος πίνακας δίνει, σε δύο ζευγάρια στηλών, μερικά δημοφιλή ζευγάρια ερμηνείας των δύο τιμών ενός bit· τα δύο πρώτα ζευγάρια είναι οι συνηθισμένες ηλεκτρικές αναπαραστασεις που είπαμε παραπάνω, και τα υπόλοιπα είναι μερικές δημοφιλείς ερμηνείες τους.

ρεύμα	όχι ρεύμα	ναι	όχι
ψηλή τάση	χαμηλή τάση	αληθές	ψευδές
πατημένος	ελεύθερος	1	0
ON	OFF	ενεργός	αδρανής
αναμένος	σβηστός	θετικός	αρνητικός
πάνω	κάτω	αρνητικός	θετικός

Απ' όλα αυτά τα ζευγάρια, το πιό σύντομο στη γραφή είναι το "1, 0", γι' αυτό πολύ συχνά υιοθετούμε αυτά τα σύμβολα γιά τις δύο δυνατές τιμές ενος bit πληροφορίας. Ένα άλλο σχόλιο αρμόζει στα δύο τελευταία ζευγάρια: μερικές φορές μπορεί το ON ή το 1 να το ερμηνεύουμε σαν "θετικός", και μερικές σαν "αρνητικός", και αντίστροφα γιά το OFF ή το 0. Όταν βλέπουμε την έννοια "θετικός" σαν συναφή με τις έννοιες "ενεργός" ή "αληθής", έρχεται φυσικό να τις συμβολίσουμε όλες αυτές με την ίδια τιμή του bit (την τιμή "1"). Από την άλλη, όπως θα δούμε στο εργαστήριο 6, όταν παριστάνουμε προσημασμένους ακεραίους αριθμούς με τον "κώδικα συμπληρώματος-ως-προς-2", τότε το "αριστερό" bit της αναπαράστασης είναι 1 γιά τους αρνητικούς αριθμούς και 0 γιά τους μεγαλύτερους ή ίσους του μηδενός. Αυτές οι δύο, αντίθετες μεταξύ τους ερμηνείες της τιμής ενός bit δείχνουν καθαρά το πόσο σχετική και αυθαίρετη είναι η ερμηνεία αυτή, και πόσο αυτή είναι θέμα απλής σύμβασης μεταξύ αυτών που χρησιμοποιούν το bit.

Με ένα μόνο bit, πολύ μικρή ποσότητα πληροφορίας μπορούμε να μεταφέρουμε --μπορούμε να επιλέξουμε ανάμεσα σε ίσα-ίσα δύο μόνο προκαθορισμένα μηνύματα· γιά περισσότερες επιλογές (περισσότερη πληροφορία) χρειαζόμαστε και περισσότερα bits. Στο σύστημα του σχήματος στην αρχή της παραγράφου, ο πομπός μετέδιδε στο δέκτη δύο (2) bits πληροφορίας· ας δούμε τώρα πώς μπορούμε να τα εκμεταλλευτούμε αυτά. Ο πρώτος τρόπος εκμετάλλευσης πολλαπλών bits πληροφορίας είναι το **κάθε bit να έχει τη δική του, χωριστή ερμηνεία·** ας δούμε δύο παραδείγματα. Σαν πρώτο παράδειγμα, ας πούμε ότι ο πομπός του παραπάνω σχήματος βρίσκεται σε ένα δωμάτιο ξενοδοχείου, ο δέκτης βρίσκεται στην κουζίνα του ξενοδοχείου, το bit A σημαίνει "πεινάω - παρακαλώ φέρτε μου το πιάτο της ημέρας", και το bit B σημαίνει "διψάω - παρακαλώ φέρτε μου το ποτό της ημέρας". Τότε, οι 4 δυνατοί συνδυασμοί καταστάσεων των δύο διακοπτών A και B μεταφέρουν ένα από τα εξής τέσσερα δυνατά μηνύματα στην κουζίνα:

- A=0, B=0: είμαι μιά χαρά - δεν θέλω τίποτα·
- A=0, B=1: παρακαλώ πολύ φέρτε μου ένα ποτό της ημέρας (χωρίς φαγητό)·
- A=1, B=0: παρακαλώ πολύ φέρτε μου ένα πιάτο της ημέρας (χωρίς ποτό)·
- A=1, B=1: παρακαλώ πολύ φέρτε μου ένα πιάτο και ένα ποτό της ημέρας.

Σαν δεύτερο παράδειγμα, ας πούμε ότι η επικοινωνία αυτή τη φορά είναι *οπτική* αντί ηλεκτρική όπως πρίν: πομπός είναι τα πίσω φώτα φρένων και όπισθεν ενός αυτοκινήτου, και δέκτης είναι ο οδηγός του από πίσω αυτοκινήτου που βλέπει τα φώτα. Ας ονομάσουμε bit A την κατάσταση των φρένων (1=πατημένα, 0=ελεύθερα), και bit B την κατάσταση του λεβιέ ταχυτήτων (1=όπισθεν, 0=άλλη θέση). Τότε, οι 4 δυνατοί συνδυασμοί καταστάσεων των δύο bits πληροφορίας που μεταφέρουν τα κόκκινα και άσπρα πίσω φώτα δίνουν στον πίσω οδηγό ένα από τα εξής τέσσερα δυνατά μηνύματα:

- A=0, B=0: προχωρώ μπροστά, κανονικά·
- A=0, B=1: κάνω όπισθεν·
- A=1, B=0: προχωρώ μπροστά αλλά φρενάρω·
- A=1, B=1: φρενάρω και προτίθεμαι να κάνω όπισθεν·

Ο δεύτερος τρόπος εκμετάλλευσης πολλαπλών bits πληροφορίας είναι η κάθε ερμηνεία να αντιστοιχεί σε **όλα τα bits μαζί, σαν ομάδα·** ας δούμε το ανάλογο των δύο προηγουμένων παραδειγμάτων σε αυτό το στυλ. Στην περίπτωση του ξενοδοχείου, μπορεί τα δύο bits να χρησιμοποιούνται γιά την παραγγελία ενός φαγητού από ένα μικρό μενού· ελλείψει περισσοτέρων bits, δεν υπάρχει δυνατότητα μεγαλύτερου μενού ή συνδυασμού πολλαπλών παραγγελιών (φαγητό, σαλάτα, ποτό, κλπ):

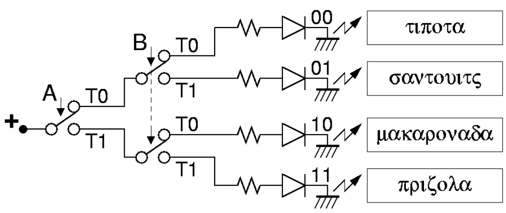
- A=0, B=0: είμαι μιά χαρά - δεν θέλω τίποτα·
- A=0, B=1: παρακαλώ πολύ φέρτε μου ένα σάντουιτς·
- A=1, B=0: παρακαλώ πολύ φέρτε μου μία μακαρονάδα·
- A=1, B=1: παρακαλώ πολύ φέρτε μου μία πριζόλα.

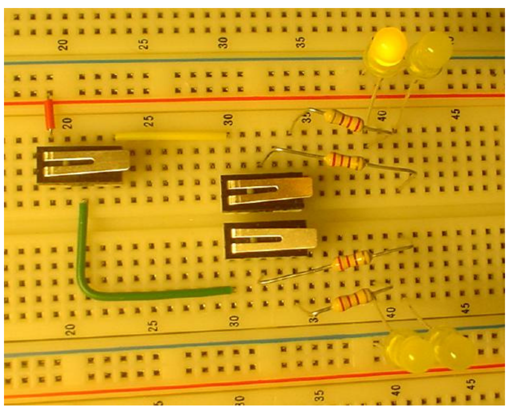
Όπως βλέπουμε εδώ, δεν υπάρχει ερμηνεία γιά το καθένα bit χωριστά από το άλλο: δεν μπορεί να ερμηνευτεί το bit A σαν μακαρονάδα και το bit B σαν σάντουιτς, διότι η πριζόλα δεν αποτελεί συνδυασμό... σάντουιτς με μακαρονάδα. Στο δεύτερο παράδειγμα, ας πούμε ότι τώρα το bit A είναι το πίσω αριστερό πορτοκαλί φώς (φλας) του αυτοκινήτου, και το bit B είναι το πίσω δεξί φλας. Σε αυτή την περίπτωση, η πληροφορία προς τον πίσω οδηγό είναι η παρακάτω (εδώ, "0" σημαίνει σβηστό φώς, και "1" σημαίνει ότι το φώς αναβοσβήνει). Όπως και στο τελευταίο παράδειγμα, το bit A δεν μπορεί να ερμηνευτεί πάντα σαν "στρίβω αριστερά" και το bit B σαν "στρίβω δεξιά", δίοτι ο συνδυασμός A=1, B=1 δεν σημαίνει "στρίβω αριστερά και δεξιά".

- A=0, B=0: προχωρώ ίσια·
- A=0, B=1: θα στρίψω δεξιά·
- A=1, B=0: θα στρίψω αριστερά·
- A=1, B=1: πρόσεχε - προσπαθώ να βρώ πού θα παρκάρω.

Τέλος, ας κάνουμε κι ένα πιό σύνθετο παράδειγμα: το ξενοδοχείο αποφασίζει να αναβαθμίσει τις υπηρεσίες εστιατορίου του και εγκαθιστά σε κάθε δωμάτιο 5 διακόπτες και 5 σύρματα προς την κουζίνα, μεταδίδοντας έτσι 5 bits πληροφορίας, A, B, C, D, και E. Τα 3 πρώτα bits, A, B, C χρησιμοποιούνται γιά την παραγγελία φαγητού, και τα 2 υπόλοιπα γιά την παραγγελία ποτού. Οι 8 συνδυασμοί τιμών των bits ABC αποφασίζεται να σημαίνουν: (000) δεν θέλω φαγητό, (001) σάντουιτς, (010) μακαρονάδα, (011) γεμιστά, (100) μουσακά, (101) ψάρι, (110) μπιφτέκια, (111) πριζόλα. Οι 4 συνδυασμοί τιμών των bits DE αποφασίζεται να σημαίνουν: (00) δεν θέλω ποτό, (01) μπύρα, (10) κρασί, (11) ούζο. Έτσι, όταν η κουζίνα βλέπει ABCDE = 00011 στέλνει ένα σκέτο ουζάκι, ενώ όταν βλέπει 10110 στέλνει ένα ψάρι με κρασί. Συνολικά, υπάρχουν 32 (=8x4) δυνατά μυνύματα: 1 μήνυμα ότι ο πελάτης δεν θέλει τίποτα (00000), 7 μηνύματα φαγητού χωρίς ποτό, 3 μηνύματα ποτού χωρίς φαγητό, και 21 (=7x3) συνδυασμοί κάποιου φαγητού με κάποιο ποτό.

<u>Πείραμα 1.6:</u> Αποκωδικοποιητής 2-σε-4





Ο μάγειρας του προηγουμένου ξενοδοχείου δεν μπορούσε ποτέ να θυμηθεί ποιός από τους κώδικες 01, 10, και 11 αντιστοιχούσε στη μακαρονάδα, ποιός στην πριζόλα, και ποιός στο σάντουιτς (γιά να περιοριστούμε απλώς στα προ της αναβάθμισης). Γιά να τον βοηθήστε, φτιάξτε το κύκλωμα που φαίνεται δεξιά. Το κύκλωμα αυτό έχει δύο εισόδους, τα bits A και B· το bit A ελέγχει το διακόπτη A. Το bit B ελέγχει τους *δύο* δεξιούς διακόπτες που φαίνονται στο σχήμα: αυτοί πρέπει πάντα να αναβοσβήνουν και οι δύο μαζί. Παρ' ότι υπάρχουν και διπλοί διακόπτες (DPDT - double pole double throw), εμείς στο εργαστήριο δεν έχουμε τέτοιους, γι' αυτό θα χρησιμοποιήσετε δύο απλούς SPDT βαλμένους δίπλα-δίπλα όπως φαίνεται στη φωτογραφία ώστε να πατιόνται κι οι δύο μαζί μ' ένα δάκτυλο.

Το κύκλωμα αυτό λέγεται **αποκωδικοποιητής** (decoder) και έχει τη βασική ιδιότητα ότι πάντα είναι αναμένη *μία και μόνο μία* από τις εξόδους του --εκείνη που αντιστοιχεί στο συνδυασμό τιμών που υπάρχουν στις εισόδους του την παρούσα στιγμή. Ο συγκεκριμένος αποκωδικοποιητής εδώ είναι μεγέθους **2-σε-4**, δηλαδή αποκωδικοποιεί 2 εισόδους στους 4 συνδυασμούς τους, άρα έχει 4 εξόδους· το κύκλωμα εδώ έχει σαν εξόδους 4 LED's. Το κύκλωμα λειτουργεί ως εξής (βάσει της τοπολογίας "δυαδικού δέντρου αποφάσεων" (binary decision tree), όπως θα μάθετε σε άλλα μαθήματα): ο διακόπτης Α "παραλαμβάνει" ρεύμα από την τροφοδοσία μέσω του πόλου του, και το διοχετεύει σε ακριβώς ένα από τα δύο μισά του κυκλώματος --το πάνω ή το κάτω-- ανάλογα με την παρούσα τιμή της εισόδου Α. Στη φωτογραφία, όταν Α=0 (ελεύθερος) το ρεύμα οδηγείται μέσω του Τ0 και του κίτρινου σύρματος στον επάνω διακόπτη Β, ενώ όταν Α=1 (πατημένος) το ρεύμα οδηγείται μέσω του Τ1 και του πράσινου σύρματος στον κάτω διακόπτη Β. Στη συνέχεια, οι διακόπτες Β οδηγούν το ρεύμα σε ένα από τα δύο μισά του υπολοίπου κυκλώματος, ανάλογα με την παρούσα τιμή της εισόδου Β· αφού ρεύμα υπάρχει σε ακριβώς ένα από τα δύο πρώτα μισά Α, ο αντίστοιχος διακόπτης Β το οδηγεί σε ακριβώς ένα από τα δύο μισά αυτού του μισού, δηλαδή σε ακριβώς ένα από τα τέσσερα σκέλη του κυκλώματος που αποτελούν και τις 4 τελικές εξόδους του. Η συνέπεια είναι ότι ανάβει η μία και μόνη LED που αντιστοιχεί στο συνδυασμό τιμών Α και Β, όπως δείχνουν τα σύμβολα 00, 01, 10, 11 στο σχήμα. Εάν πάνω από κάθε LED βάλουμε ένα ημιδιαφανές πλαστικό με γραμμένο επάνω το αντίστοιχο μήνυμα, όπως στο σχήμα δεξιά, θα έχουμε προσφέρει την επιθυμητή βοήθεια στο μάγειρα του ξενοδοχείου.

Πριν φτάσετε στο εργαστήριο, φτιάξτε τον πίνακα αληθείας γιά καθεμιά από τις 4 εξόδους του κυκλώματος. Επίσης, γιά την κάθε έξοδο παρατηρήστε το κύκλωμα που την τροφοδοτεί (δύο διακόπτες εν σειρά) και εκφράστε την με μιάν "εξίσωση" (π.χ. (Α)**ΚΑΙ**(**ΟΧΙ**(Β))), βάσει των όσων είπαμε στην §1.2. **Στο εργαστήριο**, φτιάξτε και ελέγξτε το κύκλωμα, και δείξτε το στον βοηθό σας.

1.7 Πλήθος Συνδυασμών (Μηνυμάτων) των *n* bits

Σε πόσους διαφορετικούς συνδυασμούς τιμών μπορεί να βρεθεί μιά ομάδα από *n* το πλήθος bits; Με άλλα λόγια, πόσα διαφορετικά μηνύματα μπορούμε να κωδικοποιήσουμε αν έχουμε στη διάθεσή μας *n* bits; Ή, έχοντας μιά ομάδα *n* bits, ανάμεσα σε πόσα πολλά διαφορετικά πράγματα μπορούμε να επιλέξουμε (υποδείξουμε) ένα; Ξέρουμε ήδη ότι ένα bit έχει δύο δυνατές τιμές. Επίσης ξέρουμε ότι δύο bits μπορούν να βρίσκονται σε έναν από 4 διαφορετικούς συνδυασμούς τιμών: γιά την κάθε μιά από τις 2 τιμές του πρώτου υπάρχουν 2 διαφορετικοί συνδυασμοί με τις 2 διαφορετικές τιμές του δεύτερου.

Γενικότερα, κάθε φορά που προσθέτουμε **άλλο ένα bit** στην ομάδα, **διπλασιάζεται** το πλήθος των συνδυασμών: γιά την τιμή 0 του νέου bit έχουμε τους συνδυασμούς - κώδικες - μηνύματα που είχαμε και πριν βάσει των υπολοίπων bits, και γιά την τιμή 1 του νέου bit έχουμε άλλους τόσους νέους συνδυασμούς, πάλι βάσει των υπολοίπων bits. Έτσι προκύπτει ότι τα *n* bits μπορούν να βρίσκονται σε 2^n διαφορετικούς συνδυασμούς τιμών, ή με *n* bits μπορούμε να διαλέξουμε ένα ανάμεσα σε 2^n πράγματα, ή να κωδικοποιήσουμε ένα ρεπερτόριο 2^n διαφορετικών επιτρεπτών μηνυμάτων. Γι' αυτό, οι δυνάμεις του 2 παίζουν κεφαλαιώδη ρόλο στους υπολογιστές, και θα τις βρίσκουμε μπροστά μας συνεχώς:

- **1** bit μπορεί να επιλέξει ένα ανάμεσα σε 2^1 = **2** διαφορετικά πράγματα/συνδυασμούς/μηνύματα,
- **2** bits μπορούν να επιλέξουν ένα ανάμεσα σε 2^2 = **4** διαφορετικά πράγματα/συνδυασμούς,
- **3** bits μπορούν να επιλέξουν ένα ανάμεσα σε 2^3 = **8** διαφορετικά πράγματα/συνδυασμούς,
- **4** bits επιλέγουν ανάμεσα σε 2^4 = **16** διαφορετικά πράγματα/συνδυασμούς/μηνύματα,
- **5** bits επιλέγουν ανάμεσα σε 2^5 = **32** διαφορετικά πράγματα/συνδυασμούς/μηνύματα,
- **6** bits επιλέγουν ανάμεσα σε 2^6 = **64** διαφορετικά πράγματα/συνδυασμούς/μηνύματα,
- **7** bits επιλέγουν ανάμεσα σε 2^7 = **128** διαφορετικά πράγματα/συνδυασμούς/μηνύματα,
- **8** bits επιλέγουν ανάμεσα σε 2^8 = **256** διαφορετικά πράγματα/συνδυασμούς/μηνύματα,
- **9** bits επιλέγουν ανάμεσα σε 2^9 = **512** διαφορετικά πράγματα/συνδυασμούς/μηνύματα,
- **10** bits επιλέγουν ανάμεσα σε 2^{10} = **1024 = 1 K** (Kilo) διαφορετικά πράγματα,
- **11** bits επιλέγουν ανάμεσα σε 2^{11} = **2048 = 2 K** διαφορετικά πράγματα,
- **12** bits επιλέγουν ανάμεσα σε 2^{12} = **4096 = 4 K** διαφορετικά πράγματα,
- **13** bits επιλέγουν ανάμεσα σε 2^{13} = **8192 = 8 K** διαφορετικά πράγματα,
- **14** bits επιλέγουν ανάμεσα σε 2^{14} = **16,384 = 16 K** διαφορετικά πράγματα,
- **15** bits επιλέγουν ανάμεσα σε 2^{15} = **32,768 = 32 K** διαφορετικά πράγματα,
- **16** bits επιλέγουν ανάμεσα σε 2^{16} = **65,536 = 64 K** διαφορετικά πράγματα, ...
- **20** bits επιλέγουν ανάμεσα σε 2^{20} = **1,048,576 = 1 M** (Mega) διαφορετικά πράγματα, ...
- **30** bits επιλέγουν ανάμεσα σε 2^{30} = **1,073,741,824 = 1 G** (Giga) διαφορετικά πράγματα, ...
- **40** bits επιλέγουν ανάμεσα σε 2^{40} = **1,099,511,627,776 = 1 T** (Tera) διαφορετικά πράγματα, κ.ο.κ.

Τα bits πληροφορίας, εκτός από το να τα μεταδίδουμε ή επεξεργαζόμαστε, τα αποθηκεύουμε επίσης, σε **μνήμες** (memories), γιά τις οποίες θα μιλήσουμε αργότερα. Όταν μιλάμε γιά τις μνήμες και τη χωρητικότητά τους, δεν πρέπει να συγχέουμε το πόσα bits χωράνε με το πόσους συνδυασμούς μπορεί να παραστήσει ένα πλήθος από bits. Γιά παράδειγμα, έστω ότι το πληροφοριακό σύστημα ενός Πανεπιστημίου κωδικοποιεί τον αριθμό μητρώου του κάθε φοιτητή με 24 bits. Αυτό σημαίνει ότι στο πληροφοριακό αυτό σύστημα δεν μπορούν να χωρέσουν πάνω από περίπου 16 εκατομμύρια (16 M) φοιτητές (παρελθόντων και παρόντων ετών) (16,777,216 φοιτητές, γιά την ακρίβεια). Αν τώρα ένας υπολογιστής στη γραμματεία αυτού του Πανεπιστημίου έχει μιά (μάλλον μικρή) μνήμη 16 M**bits**, αυτό σημαίνει ότι στη μνήμη αυτή χωράνε να αποθηκευτούν μέχρι περίπου 16 εκατομμύρια διαφορετικά bits (16,777,216 bits γιά την ακρίβεια). Τα bits αυτά, αν τα βλέπαμε σαν μία μόνο ομάδα, μπορούν να βρεθούν σε ένα πλήθος συνδυασμών τόσο τεράστιο που ούτε καν να το φανταστούμε μπορούμε (κάπου γύρω στο 1.6-εκατομμυριάκις εκατομμύριο...). Όμως, τα bits αυτά δεν τα κοιτάμε ποτέ σαν μία μόνο ομάδα, αλλά σαν πολλές· γιά παράδειγμα, αν σε αυτά αποθηκεύσουμε αριθμούς μητρώου φοιτητών, "κολλητά" τον έναν με τον άλλον, τότε θα χωρέσουν γύρω στις 700 χιλιάδες τέτοιοι αριθμοί μητρώου (γιά την ακρίβεια, 16777216 / 24 = 699050.67 αριθμοί μητρώου).

Ένα σκέτο bit μπορεί να μεταφέρει πολύ μικρή ποσότητα πληροφορίας, γι' αυτό συχνά τα bits τα χρησιμοποιούμε κατά ομάδες, όπως είπαμε παραπάνω. Μεταξύ των διαφόρων δυνατών μεγεθών ομάδων, η πιό συνηθισμένη, σε όλους ανεξαίρετα τους σημερινούς υπολογιστές, είναι τα **οκτώ (8) bits** που ονομάζονται ένα **Byte**. Έτσι, η κωδικοποίηση του κάθε αριθμού μητρώου στο προηγούμενο παράδειγμα ήταν σε 3 Bytes (= 3x8 = 24 bits). Σαν σύντμηση, το "b" μικρό συμβολίζει το bit, και το "B" κεφαλαίο συμβολίζει το Byte.

Ας θεωρήσουμε τώρα έναν υπολογιστή που έχει μνήμη 256 MBytes· αυτό σημαίνει 256x8 = 2048 Mbits = 2 Gbits. Από τις μνήμες σαν αυτήν, θέλουμε να ζητάμε να διαβάσουμε ορισμένα κομάτια τους που μας ενδιαφέρουν κατ' επιλογή· συνήθως, το κομάτι που μας ενδιαφέρουν είναι ένα Byte, και όχι μεμονωμένα bits διότι αυτά θεωρούνται πολύ μικρά. Κάθε φορά που θέλουμε να διαβάσουμε λοιπόν από τη μνήμη μας των 2 Gb = 256 MB, πρέπει να της διευκρινίσουμε ποιό από τα 256 εκατομμύρια Bytes που αυτή περιέχει εμείς θέλουμε να διαβάσουμε. Πόσα bits χρειαζόμαστε γιά να επιλέξουμε ένα ανάμεσα σε 256 M πράγματα; Σύμφωνα με τα παραπάνω, χρειαζόμαστε 28 bits. Τα 28 αυτά bits που πρέπει να δώσουμε στη μνήμη τα λέμε **διεύθυνση** του Byte της το οποίο επιθυμούμε να επιλέξουμε: κάθε Byte της μνήμης έχει τη δική του διεύθυνση, σαν να είναι ένα σπιτάκι σε έναν πολύ μακρύ δρόμο.

Στις μονάδες της Φυσικής, ο πολλαπλασιαστής "k" (μικρό) σημαίνει 1000· π.χ. 1 kg = 1000 g, 1 km = 1000 m, 1 kHz = 1000 Hz. Όταν μιλάμε γιά bits ή γιά Bytes, ο πολλαπλασιαστής "K" (κεφαλαίο) σημαίνει 1024, όπως τον ορίσαμε παραπάνω· π.χ. 1 Kb = 1024 bits, 1 KB = 1024 Bytes. Εκεί που τα πράγματα είναι διφορούμενα, είναι με τους πολλαπλασιαστές "M" και "G" (κεφαλαία): αυτοί, άλλοτε σημαίνουν 1,000,000 και 1,000,000,000 αντίστοιχα, μπροστά από τις παραδοσιακές φυσικές μονάδες, και άλλοτε σημαίνουν 1,048,576 και 1,073,741,824 αντίστοιχα, μπροστά από τα bits και τα Bytes που αφορούν χωρητικότητα μνημών. Έτσι, 1 MHz = 10^6 Hz ενώ 1 Mb = 2^{20} bits, και 1 GHz = 10^9 Hz ενώ 1 GB = 2^{30} Bytes. (Τα πράγματα χειροτερεύουν όταν μιλάμε γιά ταχύτητες δικτύων υπολογιστών: συνήθως, 1 Mb/s = 10^6 bits/second και 1 Gb/s = 10^9 b/s, επειδή οι ταχύτητες αυτές πηγάζουν από ρολόγια του 1 MHz ή 1 GHz!...).

1.8 Αναλογικά και Ψηφιακά Ηλεκτρονικά Συστήματα

Μία εταιρεία εμπορίας κατεψυγμένων ειδών θέλει να παρακολουθεί εξ' αποστάσεως τη θερμοκρασία του ψυγείου της, προκειμένου να εντοπίζει γρήγορα τυχόν βλάβες. Το ηλεκτρονικό θερμόμετρο που υπάρχει μέσα στο ψυγείο έχει περιοχή λειτουργίας από -25°C έως +25°C, και ακρίβεια ±0.1°C· επομένως, οι μετρήσεις του που έχει νόημα να μεταφέρονται είναι -25.0°C, -24.8°C, -24.6°C, ..., +24.8°C, +25.0°C. Η μέτρηση θα μεταφέρεται από το ψυγείο ως το γραφείο του φύλακα μέσω ηλεκτρικών καλωδίων. Υπάρχουν δύο (τουλάχιστο) τρόποι μετάδοσης αυτής της μέτρησης:

Η **αναλογική** (analog) μετάδοση λειτουργεί περίπου ως εξής: χρησιμοποιούμε ένα ηλεκτρικό σύρμα (συν την αναγκαία γείωση γιά να κλείνει κύκλωμα), και πάνω σε αυτό βάζουμε μιάν ηλεκτρική τάση ίση με τη θερμοκρασία επί έναν συντελεστή αναλογίας π.χ. 0.1 V/°C· επειδή η τάση που μεταδίδουμε είναι ανάλογη προς την θερμοκρασία, η μετάδοση λέγεται "αναλογική". Π.χ. αν η μέτρηση του θερμομέτρου είναι -18.4°C τότε στο σύρμα θα βάλουμε μιά τάση -1.84 Volt. Αφού το θερμόμετρο έχει περιοχή λειτουργίας από -25°C έως +25°C, τα ηλεκτρονικά κυκλώματα που οδηγούν το σύρμα θα πρέπει να μπορούν να ρυθμίζουν την τάση του από -2.5 V έως + 2.5 V. Προκειμένου να μην υπάρχει απώλεια ακρίβειας στη μετάδοση της μέτρησης, θα πρέπει το **συνολικό σφάλμα** των ηλεκτρονικών μετάδοσης να μην υπερβαίνει τα ±0.01 Volt (δηλ. ±10 mV). Παραδείγματος χάριν, έστω ότι η μέτρηση είναι -18.4°C, και θα έπρεπε να μεταδώσουμε -1.840 V, αλλά ο πομπός έχει σφάλμα +3 mV κι έτσι στην πραγματικότητα μεταδίδει -1.837 V· σε αυτό προστίθεται ηλεκτρικός θόρυβος +8 mV από παρεμβολές κατά μήκος του σύρματος μετάδοσης, κι έτσι στον δέκτη φτάνει τάση -1.829 V· ο δέκτης έχει σφάλμα +2 mV, κι έτσι νομίζει ότι βλέπει -1.827 V· ξέροντας ότι οι μετρήσεις είναι ακέραια πολλαπλάσια του 0.2°C, ο δέκτης ερμηνεύει την τάση που (νομίζει ότι) βλέπει σαν -18.2°C αντί του σωστού -18.4°C. Το λάθος συνέβη επειδή το συνολικό σφάλμα του συστήματος μετάδοσης είναι 3+8+2 = 13 mV που ξεπερνά την επιτρεπτή ανοχή του ±0.01 V που αντιστοιχεί στην ανοχή της μέτρησης θερμοκρασίας των ±0.1°C.

Η **ψηφιακή** (digital) μετάδοση λειτουργεί περίπου ως εξής: χρησιμοποιούμε οκτώ ηλεκτρικά σύρματα (συν την αναγκαία γείωση γιά να κλείνει κύκλωμα), και πάνω σε αυτά τα σύρματα βάζουμε 8 bits πληροφορίας που αποτελούν την κωδικοποίηση της θερμοκρασίας σύμφωνα με έναν κώδικα που εμείς αποφασίσαμε· επειδή ο κώδικας αποτελείται από "ψηφία" (digits), η μετάδοση λέγεται "ψηφιακή". Υπάρχουν 251 διαφορετικές δυνατές μετρήσεις θερμοκρασίας από το -25°C έως το +25°C ανά 0.2°C (50°C / 0.2°C = 250), επομένως ξέρουμε ότι 8 bits αρκούν γιά την κωδικοποίηση ενός τέτοιου ρεπερτορίου μηνυμάτων, αφού 8 bits έχουν 256 διαφορετικούς συνδυασμούς. Γιά τη μετάδοση του καθενός bit πληροφορίας, ας χρησιμοποιήσουμε μιάν ηλεκτρική τάση 0.0 V γιά την τιμή OFF και +5.0 V γιά την τιμή ON· η περιοχή αυτή τάσεων είναι η συνηθισμένη στα κυκλώματα του εργαστηρίου μας, και το εύρος της είναι 5 Volt, όσο δηλαδή και το εύρος λειτουργίας των αναλογικών ηλεκτρονικών στο προηγούμενο παράδειγμα (από -2.5 V έως + 2.5 V). Προκειμένου να μην υπάρξει λάθος στη μετάδοση, τώρα, πρέπει το **συνολικό σφάλμα** των ηλεκτρονικών μετάδοσης να μην υπερβαίνει τα ±2.50 Volt (δηλ. ±2500 mV). Παραδείγματος χάριν, έστω ότι το bit που θέλουμε να μεταδώσουμε είναι OFF, άρα πρέπει να μεταδώσουμε 0.0 V, αλλά ο πομπός έχει σφάλμα +400 mV κι έτσι στην πραγματικότητα μεταδίδει +0.4 V· σε αυτό προστίθεται ηλεκτρικός θόρυβος +1500 mV από παρεμβολές κατά μήκος του σύρματος μετάδοσης, κι έτσι στον δέκτη φτάνει τάση +1.9 V· ο δέκτης έχει σφάλμα +200 mV, κι έτσι νομίζει ότι βλέπει +2.1 V. Ξέροντας όμως ότι οι αναμενόμενες τιμές του bit είναι είτε 0.0 V είτε 5.0 V, και δεδομένου ότι τα +2.1 V που (νομίζει ότι) βλέπει είναι πιό κοντά στο 0 απ' ό,τι στο 5, ερμηνεύει σωστά το bit που λαμβάνει σαν OFF και όχι σαν ON.

Βλέπουμε ότι σε αυτό το παράδειγμα το ψηφιακό σύστημα στέλνει οκτώ (8) ηλεκτρικές τάσεις σαν πληροφορία, αντί της μίας (1) μόνο ηλεκτρικής τάσης που στέλνει το αναλογικό, αλλά το ψηφιακό σύστημα ανέχεται διακόσιες πενήντα (250) φορές περισσότερο θόρυβο και έλλειψη ακρίβειας στις τάσεις των ηλεκτρονικών κυκλωμάτων λειτουργίας του απ' όσο το αναλογικό (±2500 mV αντί ±10 mV). Η τεράστια εξάπλωση των ψηφιακών ηλεκτρονικών συστημάτων, σήμερα, οφείλεται στα εξής πλεονεκτήματά τους:

- **Ανοχή στο θόρυβο:** όπως εξηγήσαμε με το παραπάνω παράδειγμα, τα ψηφιακά συστήματα μπορούν να ανεχθούν τόσο πολύ θόρυβο και ανακρίβειες στη λειτουργία τους ώστε μπορούν να καταστούν σχεδόν "αλάνθαστα", σε αντίθεση με τα αναλογικά όπου όλο και κάποιος θόρυβος τελικά παρεισφρύει.
- **Χαμηλό κόστος** των ψηφιακών ηλεκτρονικών κυκλωμάτων σε σύγκριση με αναλογικά που θα έκαναν αντίστοιχη επεξεργασία σήματος: αναλογικά ηλεκτρονικά κυκλώματα με χαμηλό θόρυβο και υψηλή γραμμικότητα (χαμηλή παραμόρφωση) είναι πολύ δυσκολότερο να κατασκευαστούν απ' ό,τι ψηφιακά κυκλώματα που κάνουν την αντίστοιχη επεξεργασία μέσω αριθμητικών πράξεων πάνω σε κατάλληλα κωδικοποιημένες πληροφορίες.
- **Αναπαράσταση εγγενώς διακριτών πληροφοριών:** πέραν των εγγενώς αναλογικών πληροφοριών (π.χ. ήχος και εικόνα), υπάρχουν και εγγενώς διακριτές πληροφορίες όπως π.χ. το κείμενο. Θα ήταν εξαιρετικά δύσκολο και αφύσικο να παριστάνουμε κάθε γράμμα της αλφαβήτου μέσω μιάς διαφορετικής ηλεκτρικής τάσης, η δε λογική επεξεργασία του κειμένου ή των πληροφοριών που αυτό παριστά θα ήταν αδύνατη με αναλογικό τρόπο.

Transistors και Διακόπτες: Τα ψηφιακά συστήματα κατασκευάζονται σήμερα σε μορφή μικροηλεκτρονικών chips (IC - integrated circuit - ολοκληρωμένο κύκλωμα) που περιέχουν το καθένα χιλιάδες ή εκατομμύρια transistors. Τα transistors αυτά, όταν λειτουργούν ψηφιακά, συμπεριφέρονται σαν διακόπτες, που άλλοτε κάνουν επαφή (ανάβουν) και άλλοτε την διακόπτουν (σβήνουν). Φυσικά, δεν υπάρχει κανένα μαγικό χέρι που να αναβοσβήνει αυτούς τους διακόπτες --αυτοί ανοιγοκλείνουν υπο την επίδραση (ψηφιακών) ηλεκτρικών τάσεων. Γιά το λόγο αυτό, ξεκινήσαμε τη μελέτη των ψηφιακών συστημάτων μελετώντας απλούς, καθημερινούς διακόπτες.

Up to the Home Page of CS-120