

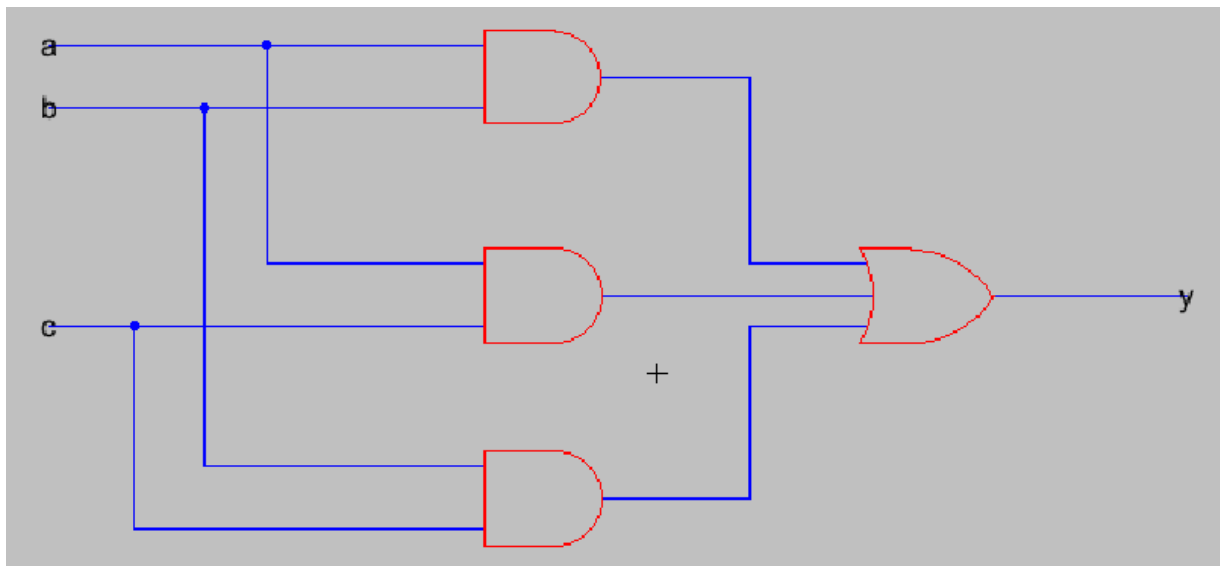
# Παραδείγματα σχεδίασης λογικών κυκλωμάτων

Τα παρακάτω παραδείγματα προσεγγίζουν τη σχεδίαση **πρακτικά**, δηλ. δεν χρησιμοποιείται κάποια συγκεκριμένη μέθοδος ψηφιακής σχεδίασης. Τα κυκλώματα δεν είναι βελτιστοποιημένα ως προς τον αριθμό ή τον τύπο των πυλών που χρησιμοποιούνται.

## Οι τρεις κριτές

Σε πολλά αγωνίσματα όπως η άρση βαρών έχουμε 3 κριτές που αποφασίζουν αν κάθε προσπάθεια είναι έγκυρη ή όχι. Σχεδιάστε ένα λογικό κύκλωμα που θα δέχεται ως είσοδο τις αποφάσεις των 3 κριτών (a, b, c, με τιμές: 1 για έγκυρη προσπάθεια, 0 για άκυρη) και θα δίνει ως έξοδο την τελική απόφαση κατά πλειοψηφία.

**Προσέγγιση:** Αν οποιοδήποτε ζευγάρι κριτών συμφωνεί ότι η προσπάθεια είναι έγκυρη, τότε η τελική έξοδος πρέπει να είναι έγκυρη. Παράγουμε με πύλες AND όλα τα πιθανά ζευγάρια (a-b, a-c και b-c) και μαζεύουμε τις μερικές αποφάσεις με μια ύλη OR. Αν έστω μια μερική απόφαση είναι 1, τότε η τελική έξοδος θα είναι 1.



## Η ένδειξη ζώνης

Στο δεύτερο παράδειγμα έχουμε μια απλοποιημένη εκδοχή του μηχανισμού προειδοποίησης για τις ζώνες του αυτοκινήτου.

Το ζητούμενο κύκλωμα δέχεται είσοδο από τους εξής αισθητήρες:

- engine: μηχανή αναμμένη (1) ή όχι (0)
- driver\_belt: ζώνη οδηγού περασμένη (1) ή όχι (0)
- codriver\_belt: ζώνη συνοδηγού περασμένη (1) ή όχι (0).
- codriver\_seated: υπάρχει συνοδηγός (1) ή όχι (0).

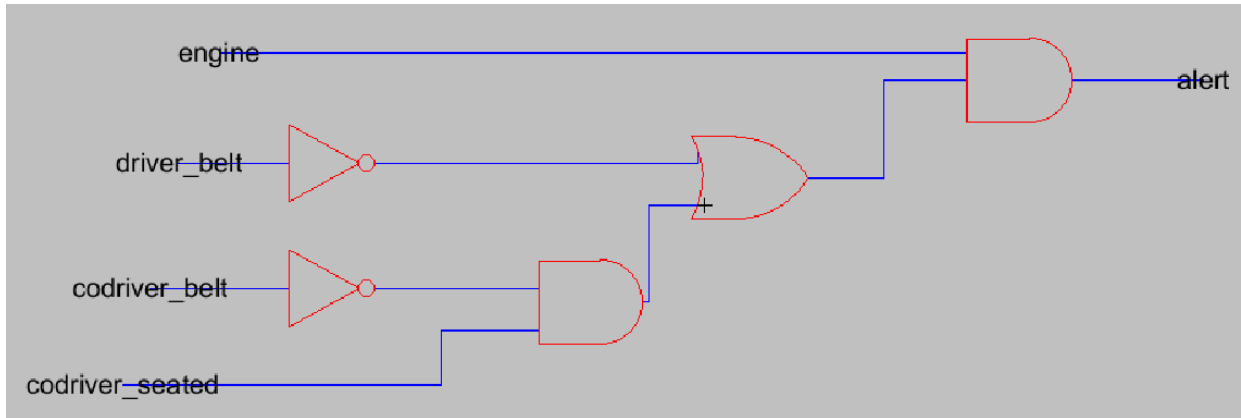
Η έξοδος θα πρέπει να είναι ενεργή (1) μόνο όταν η μηχανή είναι αναμμένη και υπάρχει πρόβλημα με κάποια ή και τις δύο ζώνες.

**Προσέγγιση:** Προσπαθούμε να υλοποιήσουμε με πύλες τη λεκτική περιγραφή:

- «μόνο όταν η μηχανή είναι αναμμένη», αυτό σημαίνει ότι οποιαδήποτε ένδειξη θα πρέπει να φράσσεται από την είσοδο engine (πύλη AND).
- «υπάρχει πρόβλημα με κάποια ή και τις δύο ζώνες», δηλαδή μια πύλη OR θα πρέπει να συγκεντρώνει τις ενδείξεις από τις 2 ζώνες.

- Οι ενδείξεις ζώνης πρέπει να αντιστραφούν (πύλες NOT) για να δίνουν 1 όταν υπάρχει πρόβλημα (μη φορεμένες ζώνες).
- Για τη ζώνη του συνοδηγού, το πρόβλημα υπάρχει μόνο όταν υπάρχει συνοδηγός (φραγή με πύλη AND).

Το τελικό κύκλωμα φαίνεται στο επόμενο σχήμα. Θυμηθείτε ότι αυτός δεν είναι ο μοναδικός τρόπος να υλοποιηθεί, ούτε ο βέλτιστος από άποψη αριθμού και τύπου πυλών.



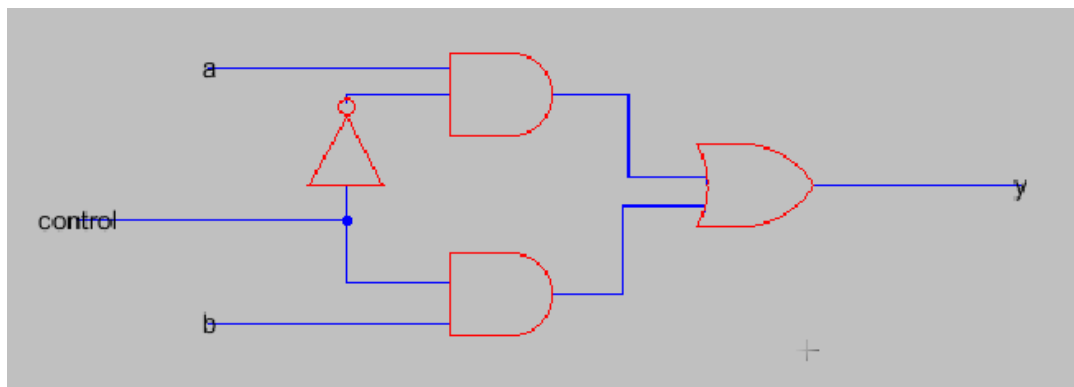
## Ο πολυπλέκτης 2-σε-1

Θέλουμε ένα λογικό κύκλωμα το οποίο θα επιλέγει μία από δύο εισόδους (a και b) να εμφανιστεί στην έξοδό του (y), ανάλογα με την τιμή μιας τρίτης εισόδου ελέγχου (control). Η λειτουργία φαίνεται στον επόμενο πίνακα αλήθειας:

control	y
0	a
1	b

Ο πίνακας αλήθειας είναι λίγο διαφορετικός απ' ό,τι συνήθως: η έξοδος y είναι είτε η είσοδος a είτε η είσοδος b, ανεξάρτητα από τις τιμές των a και b.

Στη σχεδίαση που ακολουθεί κάθε ένα από τα a και b φράσσεται εναλλάξ με το σήμα control (χρησιμοποιείται κανονικό και αντεστραμμένο) και πύλες AND. Μια τελική πύλη OR μαζεύει το τελικό αποτέλεσμα. Η OR λειτουργεί κατά τον επιθυμητό τρόπο επειδή κάθε στιγμή υπάρχει εγγύηση ότι μία από τις εισόδους της είναι 0.



**Σημείωση:** στα λογικά κυκλώματα των επεξεργαστών (και όχι μόνον) παίζουν πολύ σημαντικό ρόλο τα κυκλώματα επιλογής ενός από  $2^N$  σήματα εισόδου με βάση  $N$  σήματα ελέγχου, τα οποία ονομάζονται και **πολυπλέκτες** (multiplexers ή multiplexors).