

Μάθημα: Εισαγωγή στην Πληροφορική και την Ανοικτότητα

Κεφάλαιο 1.4 Ψηφιακά κυκλώματα

Περιεχόμενα

Εισαγωγή

1.4.1 Άλγεβρα Boole

1.4.2 Λογικές πράξεις

- Πίνακας Αλήθειας

- Η λογική πράξη NOT (άρνηση)

- Η λογική πράξη OR (ή)

- Η λογική πράξη AND (και)

- Βασικά αξιώματα και θεωρήματα της άλγεβρας Boole

- Προτεραιότητα Πράξεων

- Λογικές συναρτήσεις

1.4.3 Ψηφιακές λογικές πύλες

1.4.4 Λογικά κυκλώματα

- Συνδυαστικά κυκλώματα

- Ημιαθροιστής

- Πλήρης αθροιστής

- Ακολουθιακά κυκλώματα

- Flip-flops

 - Το RS flip-flop

 - D flip-flop

 - JK flip-flop

 - T flip-flop

 - Απαριθμητές (counters)

1.4.5 Ολοκληρωμένα κυκλώματα

- Κατηγορίες ολοκληρωμένων κυκλωμάτων

- Οικογένειες ολοκληρωμένων κυκλωμάτων

- Αναγνώριση ολοκληρωμένων κυκλωμάτων



Στόχοι

Μετά την ολοκλήρωση του κεφαλαίου οι σπουδαστές/τριες θα μπορούν να:

- ✓ αναγνωρίζουν και να χρησιμοποιούν τις πράξεις, τα θεωρήματα και τα αξιώματα της άλγεβρας Boole
- ✓ σχεδιάζουν βασικά λογικά κυκλώματα, ν' ανακαλύπτουν τη συνάρτηση που περιγράφει τη λειτουργία τους και να κατασκευάζουν τον πίνακα αληθείας του λογικού κυκλώματος
- ✓ κατανοούν τη διαφορά μεταξύ συνδυαστικού και ακολουθιακού κυκλώματος
- ✓ περιγράφουν τα κυκλώματα αθροιστών
- ✓ διακρίνουν ένα σύγχρονο από ένα ασύγχρονο ακολουθιακό κύκλωμα
- ✓ περιγράφουν τη λειτουργία των διαφόρων τύπων flip-flop
- ✓ περιγράφουν το κύκλωμα του απαριθμητή
- ✓ γνωρίζουν την έννοια του ολοκληρωμένου κυκλώματος

Εισαγωγή

Τα ψηφιακά κυκλώματα αποτελούν το δομικό υλικό μιας πλειάδας ψηφιακών συσκευών που βρίσκονται δίπλα μας για καθημερινή χρήση όπως οι ηλεκτρονικοί υπολογιστές, τα κινητά τηλέφωνα, η ψηφιακή τηλεόραση, η ψηφιακή κάμερα κ.α. Ονομάζονται ψηφιακά γιατί δέχονται σαν μεταβλητές στις εισόδους τους και παράγουν μεταβλητές στις εξόδους τους που παίρνουν μόνο δύο τιμές, την τιμή 0 και την τιμή 1. Οι συναρτήσεις που υλοποιούν τα ψηφιακά κυκλώματα ακολουθούν τους κανόνες (αξιώματα, θεωρήματα, πράξεις) της δυαδικής άλγεβρας ή άλγεβρας Boole.

1.4.1 Άλγεβρα Boole

Το 1854 ο Άγγλος μαθηματικός Τζορτζ Μπουλ (*George Boole*, 1815-1864) παρουσίασε το πρωτοποριακό για την εποχή του βιβλίο «Διερεύνηση των νόμων της σκέψης» εισχωρώντας τότε σ' έναν επιστημονικό τομέα που δεν ανήκε τόσο στα μαθηματικά όσο στη φιλοσοφία.

Ο Μπουλ δημιούργησε μια αλγεβρική δομή που αποτελείται από **λογικές μεταβλητές** που δεν είναι αριθμοί, αλλά οι τιμές αληθείας «αληθές» και «ψευδές», τις **λογικές σταθερές** που οι τιμές τους είναι πάντοτε «αληθές» ή πάντοτε «ψευδές» και τις **τρεις κύριες πράξεις**: τη σύζευξη (και), τη διάζευξη (ή) και την άρνηση (όχι). Ουσιαστικά, παρουσίασε με τα μαθηματικά της εποχής του, την Αριστοτέλεια λογική, του είναι ή δεν είναι. Ή αλλιώς του «1» ή «0».

Η άλγεβρα Boole έτυχε μικρής αναγνώρισης και χρειάστηκαν να περάσουν περίπου 70 χρόνια, το 1938, όπου ο Claude Shannon έδειξε ότι η Άλγεβρα Boole μπορούσε να εφαρμοστεί στο σχεδιασμό των ηλεκτρονικών κυκλωμάτων. Αργότερα, χρησιμοποιήθηκε για τη σχεδίαση του λογισμικού και των κυκλωμάτων των υπολογιστών επειδή αναπαριστά τις λογικές συναρτήσεις με τη βοήθεια των λογικών πράξεων. Σήμερα η Άλγεβρα Boole είναι θεμελιώδους σημασίας για την επιστήμη της Πληροφορικής.

Ας δούμε μερικά παραδείγματα για να καταλάβουμε τους παραπάνω ορισμούς της λογικής μεταβλητής και λογικής σταθεράς:

Η πρόταση «τα Χριστούγεννα είναι πάντοτε στις 25 Δεκεμβρίου» είναι πάντοτε αληθής και θεωρείται μια λογική σταθερά.

Η πρόταση «Η προηγούμενη μέρα από την Παρασκευή ονομάζεται Σάββατο» είναι πάντοτε ψευδής και θεωρείται μια λογική σταθερά.

Η πρόταση « Ο αριθμός x είναι μεγαλύτερος από τον αριθμό 6 » είναι μια λογική μεταβλητή γιατί άλλοτε είναι αληθής, όταν $x=7$ ή 8 ή 9 και άλλοτε ψευδής, όταν ο x παίρνει τις τιμές 0,1,2,3,4,5,6.

1.4.2 Λογικές πράξεις

Στην Άλγεβρα Boole ορίζονται τρεις βασικές πράξεις:

Λογική Πράξη AND με σύμβολο ($\cdot$)

Λογική Πράξη OR με σύμβολο ($+$)

Λογική Πράξη NOT με σύμβολο ($'$)

Αν παραστήσουμε το αποτέλεσμα μιας λογικής πράξης με την μεταβλητή Y και τις μεταβλητές με A, B θα ισχύει για κάθε λογική πράξη και μια λογική συνάρτηση, όπως,

Πράξη AND $Y = A \cdot B$

Πράξη OR $Y = A + B$

Πράξη NOT $Y = A'$

Πίνακας Αλήθειας

Ο πίνακας αληθείας είναι ένας πίνακας που δείχνει όλες τις δυνατές τιμές εισόδου και τις αντίστοιχες τιμές εξόδου μιας λογικής πρότασης ή ενός λογικού κυκλώματος στην Άλγεβρα Boole. Αποτελείται από στήλες που περιέχουν όλες τις μεταβλητές εισόδου και το αποτέλεσμα, ενώ οι γραμμές του περιέχουν όλους τους δυνατούς συνδυασμούς των τιμών των μεταβλητών. Στη στήλη του αποτελέσματος καταγράφονται οι αντίστοιχες τιμές της λογικής πρότασης ή του κυκλώματος.

Ο αριθμός γραμμών του πίνακα εξαρτάται από τον αριθμό των μεταβλητών. Όταν έχουμε 1 μεταβλητή οι δυνατοί συνδυασμοί των τιμών της είναι 2 (2^1), όταν έχουμε 2 μεταβλητές οι δυνατοί συνδυασμοί των τιμών τους είναι 4 (2^2), όταν έχουμε 3 μεταβλητές οι δυνατοί συνδυασμοί των τιμών τους είναι 8 (2^3) κλπ.

Ο πίνακας αληθείας είναι ένα χρήσιμο εργαλείο που μας βοηθά να κατανοήσουμε τη συμπεριφορά μιας λογικής πρότασης και να επαληθεύσουμε την ορθότητα ενός λογικού κυκλώματος.

Η λογική πράξη NOT (άρνηση)

Στην πράξη αυτή συμμετέχει μόνο μια μεταβλητή και το αποτέλεσμα της πράξης είναι το συμπλήρωμα (αντίστροφο) της μεταβλητής αυτής. Αν η μεταβλητή έχει την τιμή 0, τότε το αποτέλεσμα της πράξης NOT είναι 1 και το αντίστροφο. Ο πίνακας αλήθειας της πράξης NOT είναι:

Μεταβλητή A	Αποτέλεσμα $Y = A'$
0	1
1	0

Πίνακας 1.4.2: Πίνακας αλήθειας της πράξης NOT

Η λογική πράξη NOT συχνά συνδυάζεται με άλλες λογικές πράξεις, όπως η σύζευξη (AND) και η διάζευξη (OR), για να δημιουργήσει πιο πολύπλοκες λογικές εκφράσεις.

Η λογική πράξη OR (ή)

Στην πράξη αυτή συμμετέχουν δύο ή περισσότερες μεταβλητές και το αποτέλεσμα της πράξης είναι 1 αν τουλάχιστον μια από τις μεταβλητές είναι 1. Η πράξη αυτή λέγεται και λογική πρόσθεση.

Ο πίνακας αληθείας της πράξης OR με δύο μεταβλητές, είναι:

Μεταβλητές		Αποτέλεσμα $Y = A+B$
A	B	
0	0	0
0	1	1
1	0	1
1	1	1

Πίνακας 1.4.2.α: Πίνακας αλήθειας της πράξης OR



Ένα ενδιαφέρον σημείο σχετικά με την πράξη OR είναι ότι αν η τιμή σε μια από τις μεταβλητές εισόδου είναι 1, δεν χρειάζεται να ελέγξουμε την τιμή των υπολοίπων μεταβλητών: μπορούμε γρήγορα να συμπεράνουμε ότι το αποτέλεσμα είναι 1.

Η λογική πράξη AND (και)

Στην πράξη AND συμμετέχουν δύο ή περισσότερες μεταβλητές και το αποτέλεσμα της πράξης είναι 1, αν και μόνο αν όλες οι μεταβλητές είναι 1. Η πράξη αυτή λέγεται και λογικός πολλαπλασιασμός.

Ο πίνακας αληθείας της πράξης AND, είναι ο ακόλουθος:

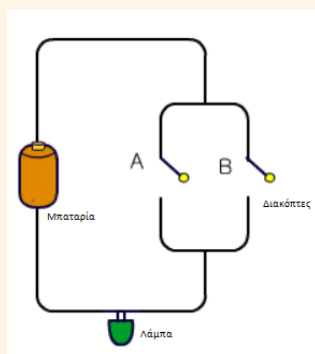
Μεταβλητές		Αποτέλεσμα $Y = A \cdot B$
A	B	
0	0	0
0	1	0
1	0	0
1	1	1

Πίνακας 1.4.2.β: Πίνακας αλήθειας της πράξης AND

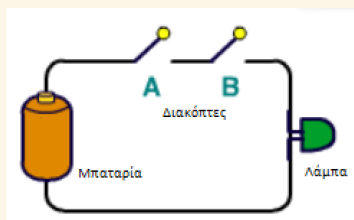


Ένα ενδιαφέρον σημείο σχετικά με την πράξη AND είναι ότι εάν η τιμή σε μια από τις μεταβλητές εισόδου είναι 0, δεν χρειάζεται να ελέγξουμε την τιμή των υπολοίπων μεταβλητών: μπορούμε γρήγορα να συμπεράνουμε ότι το αποτέλεσμα είναι 0.

Τις λογικές πράξεις AND και OR μπορούμε να τις περιγράψουμε με τη χρήση κυκλωμάτων διακοπών.



Για να ανάψει η λάμπα του παραπάνω κυκλώματος θα πρέπει να κλείσει ο A ή ο B διακόπτης.



Για να ανάψει η λάμπα του παραπάνω κυκλώματος θα πρέπει **και** οι δύο διακόπτες να είναι κλειστοί.

Η άλγεβρα Boole στηρίζεται σε μια σειρά αξιωμάτων, που παρουσίασε το 1933 ο μαθηματικός Έντουαρντ Χάντινγκτον (Edward Vermilye Huntington). Τα αξιώματα και τα θεωρήματα της άλγεβρας Boole, χαρακτηρίζονται από μία ιδιότητα που ονομάζεται **δυϊκότητα**. Εξακολουθούν, δηλαδή, να ισχύουν και στην περίπτωση που αντικατασταθούν: η πράξη OR(+) με την πράξη AND(•), το (μηδέν 0) με (ένα 1) και το (ένα 1) με (μηδέν 0).

Βασικά αξιώματα και θεωρήματα της άλγεβρας Boole

1ο Αξίωμα. Εάν $A \neq 1$ θα είναι $A=0$ ή το δυϊκό, εάν $A \neq 0$ θα είναι $A=1$

2ο Αξίωμα. Αντιμετάθεση: $A + B = B + A$, $A \cdot B = B \cdot A$

3ο Αξίωμα. Επιμερισμός: $A \cdot (B+C) = (A \cdot B) + (A \cdot C)$, $A + (B \cdot C) = (A+B) \cdot (A+C)$

4ο Αξίωμα. Προσεταιρισμός: $A \cdot B \cdot C = A \cdot (B \cdot C) = (A \cdot B) \cdot C$, $A+B+C = A + (B+C) = (A+B)+C$

5ο Αξίωμα. Ουδέτερο Στοιχείο: Για την πράξη «+» ή «•» υπάρχει ένα στοιχείο «0» ή «1» τέτοιο ώστε $A+0=A$ και $A \cdot 1=A$

6ο Αξίωμα. Συμπλήρωμα: για κάθε στοιχείο A υπάρχει το συμπλήρωμα του A' ώστε $A+A'=1$ και $A \cdot A'=0$

1ο Θεώρημα: $A + A = A$ και $A \cdot A = A$

2ο Θεώρημα: $A + 1 = 1$ και $A \cdot 0 = 0$

3ο Θεώρημα: $(A')' = A$

4ο Θεώρημα (DeMorgan): $(A + B)' = A' \cdot B'$ και $(A \cdot B)' = A' + B'$

5° Θεώρημα: Απορρόφηση $A + A \cdot B = A$ και $A \cdot (A + B) = A$

Προτεραιότητα Πράξεων

Για την εκτέλεση των πράξεων της Άλγεβρας Boole είναι ανάγκη να καθορισθεί η προτεραιότητα της εκτέλεσής τους. Η προτεραιότητα των πράξεων είναι η εξής:

πρώτα οι πράξεις μέσα σε παρενθέσεις (),

η πράξη NOT δηλ. τα συμπληρώματα,

η πράξη AND και τέλος

η πράξη OR

Λογικές συναρτήσεις

Μία λογική συνάρτηση είναι μία έκφραση που αποτελείται από συνδυασμούς λογικών μεταβλητών (είσοδοι), τους τελεστές OR, AND και NOT, παρενθέσεις και ένα “ίσον”. Για μία δεδομένη τιμή των μεταβλητών, το αποτέλεσμα(έξοδος) της συνάρτησης μπορεί να είναι είτε 0 είτε 1, ανάλογα με τις τιμές των μεταβλητών που συμμετέχουν σ’ αυτήν.

Για παράδειγμα η λογική συνάρτηση $F = A' \cdot B + C$ αποτελείται από 3 λογικές μεταβλητές (A, B και C) και το αποτέλεσμα της F, ισούται με 1 αν ταυτόχρονα $A=0$ και $B=1$ ή αν $C=1$.

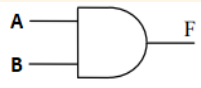
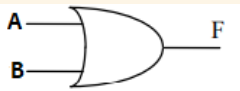
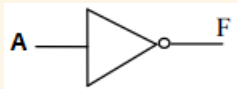
Κάθε λογική συνάρτηση έχει έναν πίνακα αλήθειας που καθορίζει τις τιμές του αποτελέσματος για κάθε συνδυασμό των μεταβλητών. Ο πίνακας αλήθειας αποτελείται από 2^n συνδυασμών 1 και 0, των n λογικών μεταβλητών της συνάρτησης και μία στήλη που να δείχνει το αποτέλεσμα για κάθε συνδυασμό των μεταβλητών.

Ο πίνακας αλήθειας της λογικής συνάρτησης $F = A' \cdot B + C$ είναι:

Μεταβλητές			Αποτέλεσμα $F = A' \cdot B + C$
A	B	C	
0	0	0	$1 \cdot 0 = 0$
0	0	1	$1 \cdot 0 = 0$
0	1	0	$1 \cdot 1 = 1$
0	1	1	$1 \cdot 1 = 1$
1	0	0	$0 \cdot 0 = 0$
1	0	1	$0 \cdot 0 = 0$
1	1	0	$0 \cdot 1 = 0$
1	1	1	$0 \cdot 1 = 0$

1.4.3 Ψηφιακές λογικές πύλες

Τα ηλεκτρονικά κυκλώματα που υλοποιούν τις παραπάνω λογικές πράξεις της Άλγεβρας Boole ονομάζονται ψηφιακές λογικές πύλες. Κάθε λογική πύλη αντιστοιχεί σε μια από τις πράξεις της Άλγεβρας Boole και αναπαρίσταται γραφικά με διαφορετικό σχήμα. Στον παρακάτω πίνακα δίνονται τα σχήματα των λογικών πυλών, οι λογικές πράξεις που υλοποιούν και ο πίνακας αλήθειας της κάθε πράξης.

ΟΝΟΜΑ	ΣΥΜΒΟΛΟ	ΠΡΑΞΗ	ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ															
AND		$F = A \cdot B$	<table><tr><th>A</th><th>B</th><th>A AND B</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	A AND B	0	0	0	0	1	0	1	0	0	1	1	1
A	B	A AND B																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = A + B$	<table><tr><th>A</th><th>B</th><th>A OR B</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	A OR B	0	0	0	0	1	1	1	0	1	1	1	1
A	B	A OR B																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NOT		$F = A'$	<table><tr><th>A</th><th>NOT A</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	NOT A	0	1	1	0									
A	NOT A																	
0	1																	
1	0																	

Οι ψηφιακές πύλες μπορούν να συνδυαστούν μεταξύ τους για να δημιουργήσουν πιο πολύπλοκα ψηφιακά κυκλώματα. Για παράδειγμα, μπορούμε να συνδυάσουμε:

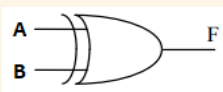
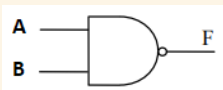
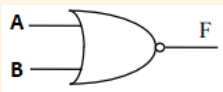
Πύλες AND και OR για να δημιουργήσουμε πύλες XOR (Αποκλειστικό OR). Η πύλη XOR δίνει αποτέλεσμα 1 μόνο εάν μια και μόνο μία από τις μεταβλητές είναι 1.

Πύλες NOT και AND και να δημιουργήσουμε πύλες NAND. Η πύλη NAND δίνει αποτέλεσμα 0 μόνο εάν όλες οι μεταβλητές είναι 1, διαφορετικά δίνει αποτέλεσμα 1.

Πύλες NOT και OR και να δημιουργήσουμε πύλες NOR. Η πύλη NOR δίνει αποτέλεσμα 1 μόνο όταν όλες οι μεταβλητές είναι 0, διαφορετικά δίνει αποτέλεσμα 0.

Πύλες NOT και XOR και να δημιουργήσουμε πύλες XNOR. Η πύλη XNOR δίνει αποτέλεσμα 1 όταν οι μεταβλητές έχουν την ίδια τιμή (0 ή 1), διαφορετικά δίνει αποτέλεσμα 0.

Στον παρακάτω πίνακα δίνονται τα σχήματα των παραπάνω λογικών πυλών, οι λογικές πράξεις που υλοποιούν και ο πίνακας αλήθειας της κάθε πράξης.

ΟΝΟΜΑ	ΣΥΜΒΟΛΟ	ΠΡΑΞΗ	ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ															
XOR		$F = A \cdot B' + A' \cdot B$ $= A \oplus B$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
NAND		$F = \overline{A \cdot B}$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	1	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$F = A' + B'$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	1	0	1	0	1	0	0	1	1	0
A	B	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XNOR		$F = A \cdot B + A' \cdot B'$ $= A' \oplus B'$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	1	0	1	0	1	0	0	1	1	1
A	B	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																



Μια ιδιότητα της πύλης XOR είναι ότι εάν η τιμή μιας μεταβλητής είναι 1, το αποτέλεσμα είναι το συμπλήρωμα της τιμής της άλλης μεταβλητής.

Η υλοποίηση των λογικών πυλών σε επίπεδο υλικού βασίζεται κυρίως στη χρήση των τρανζίστορ(transistor). Το τρανζίστορ το συναντήσαμε στο κεφάλαιο 1.1 όταν μιλήσαμε για τις γενιές των υπολογιστών. Ένα τρανζίστορ (transistor) είναι ένα στοιχείο κυκλώματος που έχει συνήθως τρεις ακροδέκτες. Το βασικό χαρακτηριστικό του τρανζίστορ είναι ότι ενώ στην είσοδο του μπορούμε να δώσουμε ένα μεγάλο αριθμό διαφορετικών καταστάσεων, στην έξοδο του παίρνουμε μόνο δύο διακριτές καταστάσεις: απουσία ρεύματος ή ύπαρξη ρεύματος. Με κατάλληλους συνδυασμούς τρανζίστορ (ή και άλλων ηλεκτρονικών εξαρτημάτων) μπορούμε να δημιουργήσουμε λογικές πύλες.

1.4.4 Λογικά κυκλώματα

Οι λειτουργίες που ενεργοποιούνται στον υπολογιστή, είτε από το χρήστη είτε από τον προγραμματιστή, υλοποιούνται στο εσωτερικό του υπολογιστή από τα κατάλληλα λογικά κυκλώματα που αποτελούνται από συνδυασμούς των λογικών πυλών.

Τα λογικά κυκλώματα μπορούν να σχεδιαστούν και να κατασκευαστούν σε διάφορα επίπεδα πολυπλοκότητας. Από απλά κυκλώματα που αποτελούνται από μερικές λογικές πύλες, μέχρι πολύπλοκα ολοκληρωμένα κυκλώματα που αποτελούνται από εκατομμύρια ή ακόμα και δισεκατομμύρια λογικές πύλες.

Τα λογικά κυκλώματα χωρίζονται σε δύο μεγάλες κατηγορίες, στα συνδυαστικά και στα ακολουθιακά.

Συνδυαστικά κυκλώματα

Ένα συνδυαστικό κύκλωμα αποτελείται από λογικές πύλες των οποίων οι έξοδοι, σε κάθε χρονική στιγμή, εξαρτώνται από κάποιο λογικό συνδυασμό των τιμών στις εισόδους του κυκλώματος. Για η μεταβλητές εισόδου, υπάρχουν 2^n δυνατοί συνδυασμοί τιμών εισόδου. Για κάθε συνδυασμό τιμών εισόδων, υπάρχει ένας και μόνον ένας δυνατός συνδυασμός τιμών εξόδων. Τα συνδυαστικά κυκλώματα δεν διαθέτουν κάποιο στοιχείο μνήμης, επομένως το κύκλωμα δεν μπορεί να αποθηκεύσει προηγούμενες τιμές των μεταβλητών εισόδου.

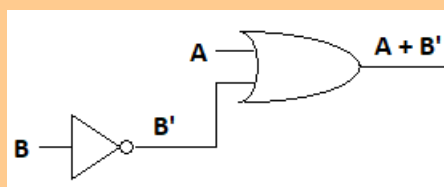


Παραδείγματα υλοποίησης συνδυαστικού κυκλώματος

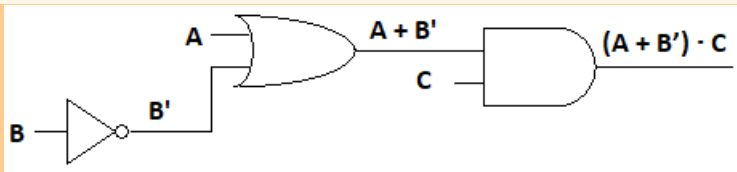
Οποιαδήποτε λογική συνάρτηση μπορεί να μετατραπεί σε κύκλωμα, συνδυάζοντας τις λογικές πύλες με έναν σχετικά εύκολο τρόπο.

Να σχεδιαστεί συνδυαστικό κύκλωμα που να υλοποιεί τη συνάρτηση: $F = (A + B') \cdot C + A'$. Το κύκλωμα έχει τρεις μεταβλητές εισόδου τις A, B και C. Σύμφωνα με την προτεραιότητα των πράξεων πρώτα εκτελούνται οι πράξεις μέσα σε παρενθέσεις (), μετά η πράξη NOT δηλ. τα συμπληρώματα, μετά η πράξη AND και τέλος η πράξη OR.

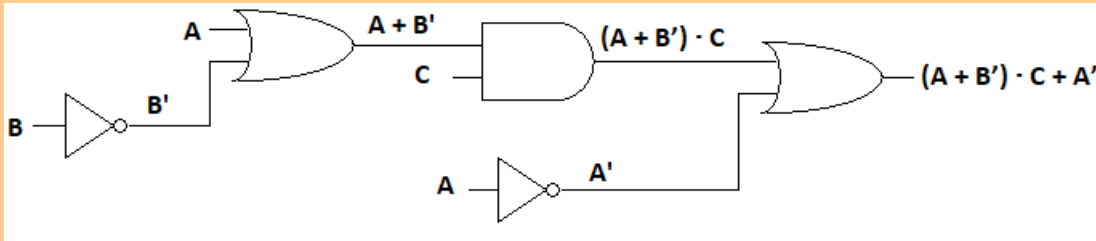
Στην παρένθεση έχουμε την πράξη OR μεταξύ της μεταβλητής εισόδου A και του συμπληρώματος (NOT) της μεταβλητής εισόδου B. Το κύκλωμα που εκτελεί αυτές τις πράξεις είναι:



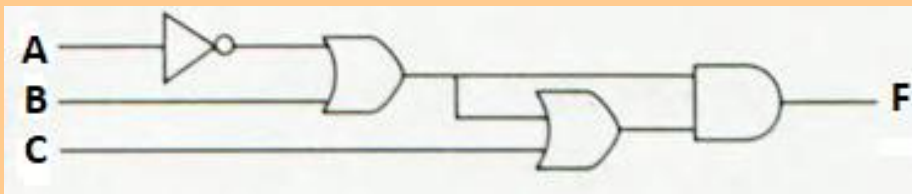
Το αποτέλεσμα (η έξοδος) του παραπάνω κυκλώματος είναι η μια μεταβλητή εισόδου της πράξης AND και η άλλη μεταβλητή εισόδου είναι η C. Το κύκλωμα τώρα γίνεται:



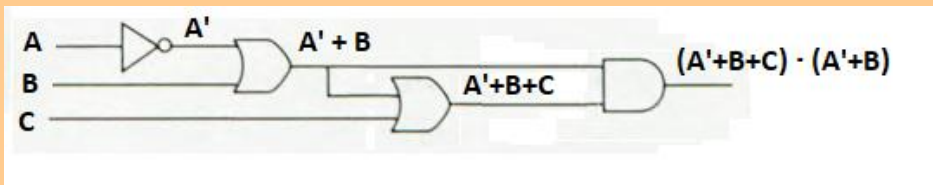
Τέλος, στην πράξη OR που ακολουθεί συμμετέχουν ως μεταβλητές εισόδου το αποτέλεσμα (η έξοδος) του παραπάνω κυκλώματος και το συμπλήρωμα (NOT) της μεταβλητής A. Το κύκλωμα που υλοποιεί τη συνάρτηση είναι:



Να αναλύσετε το παρακάτω λογικό κύκλωμα για να βρείτε τη συνάρτηση εξόδου F.



Το κύκλωμα αναλύεται ως εξής:



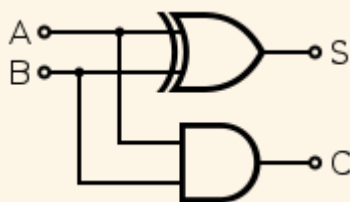
Η συνάρτηση εξόδου είναι: $F = (A' + B + C) \cdot (A' + B)$

Ημιαθροιστής

Τα ψηφιακά κυκλώματα στο εσωτερικό των ηλεκτρονικών υπολογιστών εκτελούν πολλές διεργασίες επεξεργασίας πληροφορίας όπως για παράδειγμα αριθμητικές πράξεις. Η βασικότερη πράξη για έναν ηλεκτρονικό υπολογιστή είναι η πρόσθεση δύο δυαδικών ψηφίων.

Το συνδυαστικό κύκλωμα που εκτελεί την πράξη της πρόσθεσης δύο δυαδικών αριθμών μήκους ενός bit ο καθένας, ονομάζεται ημιαθροιστής (half-adder) ενώ το κύκλωμα που προσθέτει και το κρατούμενο μαζί με τα δυο bit τους προσθετέους ονομάζεται πλήρης αθροιστής (full adder).

Ο ημιαθροιστής είναι ένα κύκλωμα άθροισης με δύο εισόδους, A και B και δύο εξόδους. Οι δύο είσοδοι ορίζουν τα δύο bit που θα προστεθούν. Η πρώτη έξοδος είναι το άθροισμα S (sum) των δύο bit, ενώ η δεύτερη έξοδος είναι το τυχόν κρατούμενο C (carry). Η εικόνα 1.4.4 δείχνει έναν ημιαθροιστή με τον πίνακα αληθείας του και τις λογικές πύλες που χρησιμοποιούνται για τη δημιουργία του κυκλώματος.



A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

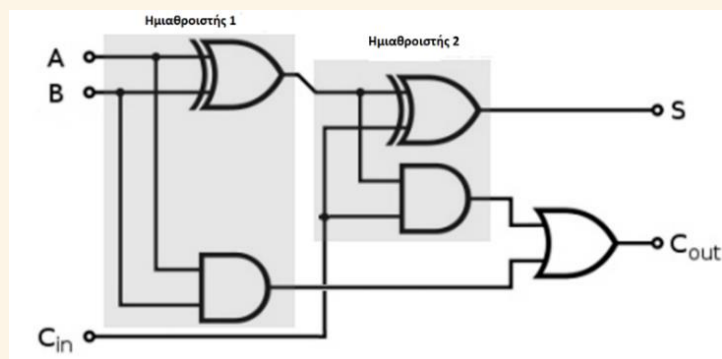
Εικόνα 1.4.4. Κύκλωμα ημιαθροιστή και πίνακας αλήθειας

Οι λογικές συναρτήσεις των εξόδων του κυκλώματος που προκύπτουν από τον πίνακα αλήθειας είναι οι ακόλουθες: $S = A' \cdot B + A \cdot B' = A \oplus B$ και $C = A \cdot B$.

Πλήρης αθροιστής

Ένα συνδυαστικό κύκλωμα που εκτελεί την πρόσθεση τριών δυαδικών αριθμών (δυο δυαδικών αριθμών και ενός προηγούμενου κρατούμενου) λέγεται πλήρης αθροιστής (full adder). Ο πλήρης αθροιστής έχει τρεις εισόδους και δύο εξόδους. Οι δύο από τις μεταβλητές εισόδου είναι τα δύο ψηφία που προστίθενται και η τρίτη είναι το κρατούμενο.

Το όνομα του ημιαθροιστή προέρχεται από το γεγονός ότι δυο κυκλώματα ημιαθροιστών μπορούν να χρησιμοποιηθούν για να υλοποιηθούν έναν πλήρη αθροιστή. Η εικόνα 1.4.4.α δείχνει έναν πλήρη αθροιστή με τον πίνακα αληθείας του και τις λογικές πύλες που χρησιμοποιούνται για τη δημιουργία του κυκλώματος.



A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

A: προσθετέος
B: προσθετέος
Cin: προηγούμενο κρατούμενο
S: άθροισμα
Cout: νέο κρατούμενο

Εικόνα 1.4.4.α. Κύκλωμα και πίνακας αλήθειας πλήρη αθροιστή

Οι λογικές συναρτήσεις των εξόδων του κυκλώματος που προκύπτουν από τον πίνακα αλήθειας είναι οι ακόλουθες:

$$S = A \oplus B \oplus \text{Cin} \text{ και } \text{Cout} = A \cdot B + B \cdot \text{Cin} + A \cdot \text{Cin}$$

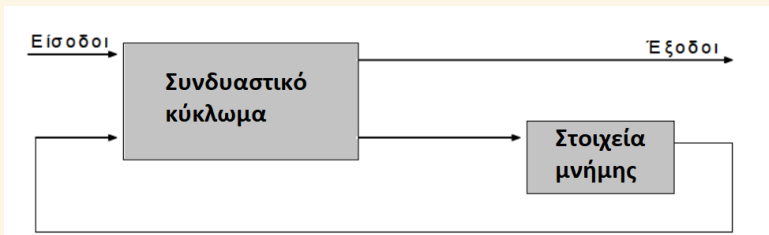
Ακολουθιακά κυκλώματα

Τα συνδυαστικά κυκλώματα που μελετήσαμε είναι χωρίς «μνήμη», δεν θυμούνται την προηγούμενη τιμή (κατάσταση) στην έξοδο τους αλλά οι έξοδοί τους την κάθε στιγμή εξαρτώνται αποκλειστικά και μόνο από τις τιμές των εισόδων τους εκείνη τη χρονική στιγμή.

Τα ακολουθιακά κυκλώματα εκτός από τις λογικές πύλες (όπως κι ένα συνδυαστικό κύκλωμα) περιλαμβάνουν και επιπλέον στοιχεία μνήμης. Η έξοδος του κυκλώματος εξαρτάται κάθε φορά όχι μόνο

από τις τρέχουσες εισόδους του αλλά και από την κατάσταση (τιμή) της προηγούμενης εξόδου που είναι αποθηκευμένη στα στοιχεία μνήμης. Με άλλα λόγια, ένα ακολουθιακό κύκλωμα έχει τη δυνατότητα να «θυμάται» την προηγούμενη κατάσταση του.

Η δομή ενός ακολουθιακού κυκλώματος φαίνεται στο παρακάτω σχήμα:



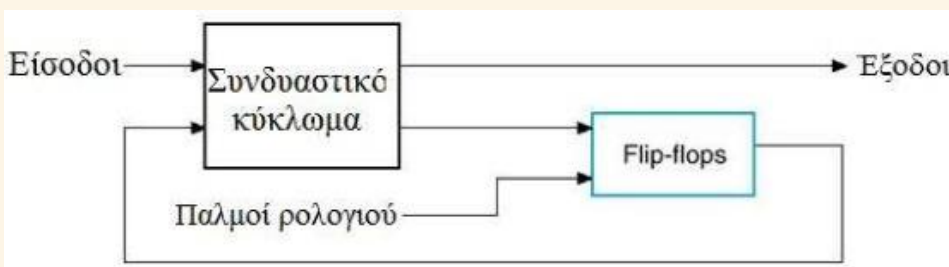
Για να καταλάβουμε τη σημασία της μνήμης στο σχεδιασμό ενός ψηφιακού κυκλώματος ας δούμε το παρακάτω παράδειγμα συσκευής που χρησιμοποιούμε καθημερινά και περιλαμβάνει ένα ακολουθιακό κύκλωμα:

Το χειριστήριο της τηλεόρασης έχει τα κουμπιά «πάνω» και «κάτω» για την επιλογή του επόμενου ή του προηγούμενου καναλιού. Πατώντας το κουμπί "πάνω" δίνεται στην τηλεόραση μια είσοδος που της λέει να μεταβεί στο επόμενο κανάλι από αυτό που δείχνει αυτήν τη στιγμή. Έτσι αν η τηλεόραση δείχνει το κανάλι 5, πατώντας το κουμπί «πάνω» θα δείξει το κανάλι 6. Ενώ αν δείχνει το κανάλι 8, πατώντας το κουμπί «κάτω» θα δείξει το κανάλι 7. Για να λειτουργήσει σωστά η επιλογή καναλιού, η τηλεόρασή σας πρέπει κάθε φορά να αποθηκεύει το τρέχον κανάλι. Το ακολουθιακό κύκλωμα επιλογής καναλιού δέχεται ως εισόδους το πάτημα του κουμπιού «πάνω» ή «κάτω» και το τρέχον κανάλι και υπολογίζει το νέο κανάλι που θα δείξει.

Τα ακολουθιακά κυκλώματα χωρίζονται σε δύο μεγάλες κατηγορίες:

Ασύγχρονα, όπου η κατάσταση της εξόδου του κυκλώματος αλλάζει οποιαδήποτε χρονική στιγμή και με τη προϋπόθεση ότι υπάρχει κάποια μεταβολή στις εισόδους.

Σύγχρονα, όπου η κατάσταση της εξόδου του κυκλώματος αλλάζει σε καθορισμένες χρονικές στιγμές που καθορίζονται από τους παλμούς ενός ρολογιού (clock). Οι παλμοί ρολογιού χρησιμοποιούνται με τέτοιο τρόπο ώστε οι τιμές των εξόδων των στοιχείων μνήμης να μεταβάλλονται μόνο κατά τη διάρκεια εφαρμογής των παλμών ρολογιού. Όταν δεν υπάρχει παλμός ρολογιού, οι τιμές των εξόδων δεν αλλάζουν, ακόμα και αν οι τιμές των εισόδων τους μεταβληθούν. Στο παρακάτω σχήμα φαίνεται η δομή ενός σύγχρονου ακολουθιακού κυκλώματος.

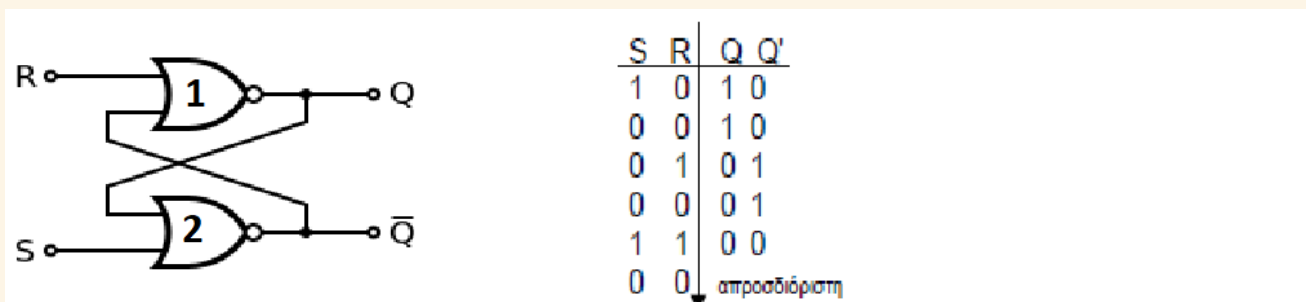


Flip-flops

Τα στοιχεία μνήμης που χρησιμοποιούνται σ' ένα σύγχρονο ακολουθιακό κύκλωμα λέγονται flip-flop. Το flip-flop είναι ένα κύκλωμα που μπορεί να αποθηκεύσει ένα bit πληροφορίας. Έχει δύο εξόδους, μία για την τιμή του bit που είναι αποθηκευμένο μέσα τους και μία για το συμπλήρωμά της. Ένα κύκλωμα flip-flop μπορεί να διατηρηθεί σε μία δυαδική κατάσταση επ' αόριστο, έως ότου κάποιο σήμα προκαλέσει αλλαγή της κατάστασης του. Ανάλογα με τον αριθμό των εισόδων που έχουν και τον τρόπο με τον οποίο αλλάζουν κατάσταση, υπάρχουν τέσσερις βασικοί τύποι flip-flop:

Το RS flip-flop

Είναι ο απλούστερος τύπος flip-flop. Έχει δύο εισόδους S (set) και R (reset) και δύο εξόδους Q και Q', οι οποίες είναι πάντα συμπληρωματικές μεταξύ τους. Το κύκλωμα κατασκευάζεται συνδέοντας δύο πύλες NOR όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 1.4.4.β. Κύκλωμα και πίνακας αλήθειας RS flip-flop

Το κύκλωμα λειτουργεί ως εξής:

Αν $S=1$ και $R=0$ η έξοδος στην πύλη 2 $Q' = 0$ ανεξάρτητα από την τιμή της άλλης εισόδου (πίνακας αλήθειας της πύλης NOR). Συνεπώς, η πύλη 1 έχει και τις δύο εισόδους της $R=0$ και $Q'=0$ και άρα $Q=1$. Δηλαδή, για $S=1$ και $R=0$ έχουμε $Q=1$ και $Q' = 0$ και η κατάσταση του flip-flop γίνεται 1 (SET).

Αν τώρα αλλάξουμε την τιμή της εισόδου $S = 0$, αυτή η αλλαγή δεν επηρεάζει την κατάσταση του κυκλώματος, αφού η άλλη είσοδος της πύλης 2 εξακολουθεί να είναι $Q=1$. Δηλαδή, το κύκλωμα "θυμάται" την προηγούμενη κατάσταση του.

Αν $R=1$ και $S=0$, η έξοδος της πύλης 1 γίνεται $Q=0$ και κατά συνέπεια η έξοδος της πύλης 2 γίνεται $Q' = 1$, αφού και οι δύο εισόδους της (S και Q) έχουν τιμή 0. Άρα, για $S=0$ και $R=1$ επιτυγχάνεται "μηδενισμός" (RESET) του στοιχείου μνήμης.

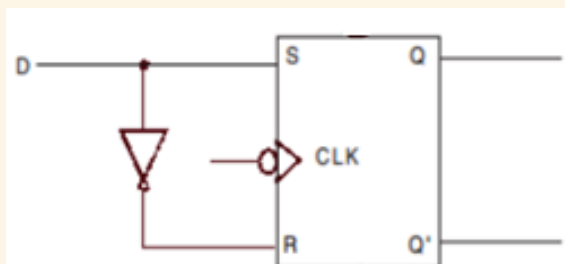
Αν τώρα αλλάξουμε την τιμή της εισόδου $R = 0$, η κατάσταση του κυκλώματος δεν αλλάζει, δηλαδή $Q=0$ και $Q' = 1$. Άρα και πάλι μπορούμε να πούμε ότι το κύκλωμα "θυμάται" την προηγούμενη κατάστασή του.

Αν και οι δύο εισόδους S και R γίνονται 1 ($S=1$, $R=1$), τότε και οι δύο εξόδους των πυλών 1 και 2 γίνονται 0 ($Q=0$, $Q' = 0$). Παρατηρούμε ότι σε αυτή την περίπτωση οι εξόδους δεν είναι η μία συμπληρωματική της άλλης και το flip-flop οδηγείται σε απροσδιόριστη κατάσταση η οποία θα πρέπει να αποφεύγεται.

D flip-flop

Ένας τρόπος για να εξαλειφθεί η απροσδιόριστη κατάσταση στο RS flip-flop είναι να διασφαλιστεί ότι οι εισόδους S και R δεν θα είναι ποτέ ίσες με 1 ταυτόχρονα. Αυτό επιτυγχάνεται στο D flip-flop. Το D flip-flop έχει την είσοδο D (data), μία είσοδο για τον παλμό του ρολογιού (CLK) και δύο εξόδους Q και Q'. Η λειτουργία του είναι ότι μεταφέρει στην έξοδο του την τιμή της εισόδου μόλις εφαρμοστεί ένας παλμός ρολογιού ($CLK=1$). Αν υπάρξει αλλαγή στην τιμή της εισόδου D, αυτή η αλλαγή δεν επηρεάζει την τιμή της εξόδου του κυκλώματος Q όταν δεν υπάρχει παλμός ρολογιού ($CLK=0$).

Το D flip-flop μπορεί να κατασκευαστεί εύκολα από ένα RS flip-flop απλώς ενσωματώνοντας μια πύλη NOT μεταξύ S και R έτσι ώστε η είσοδος της NOT να βρίσκεται στο S άκρο και η έξοδος της στο R άκρο. Μπορούμε έτσι να απαλλαγούμε από την απροσδιόριστη κατάσταση, $S = R = 1$ του RS flip-flop. Η δομή και ο πίνακας αλήθειας του D flip-flop φαίνονται στο παρακάτω σχήμα.

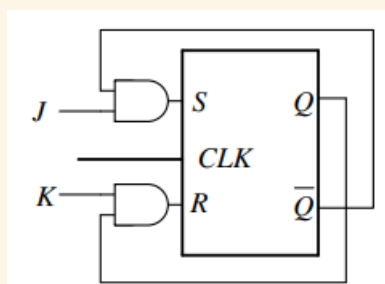


CLK	D	Q	Q'
0	0	Q	$\overline{Q}$
0	1	Q	$\overline{Q}$
1	0	0	1
1	1	1	0

Παρατηρούμε ότι, η έξοδος Q έχει την ίδια τιμή με την είσοδο D και παραμένει ίδια μέχρι να δοθεί μια νέα τιμή στην είσοδο D. Δηλαδή, μπορούμε να πούμε ότι το κύκλωμα "θυμάται" τις τιμές της εισόδου του.

JK flip-flop

Το JK flip-flop έχει δύο σύγχρονες εισόδους που ονομάζονται J και K και μπορεί να κατασκευαστεί από ένα RS flip-flop προσθέτοντας δυο πύλες AND όπως φαίνεται στο παρακάτω σχήμα.



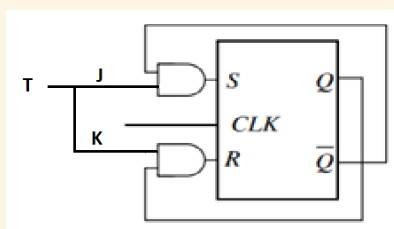
J	K	CLK	Q	Q'	Παρατηρήσεις
0	0	1	Q	Q'	Καμία αλλαγή στην κατάσταση
0	1	1	0	1	reset
1	0	1	1	0	set
1	1	1	Q'	Q	Αντιστροφή κατάστασης

Από τον πίνακα αλήθειας του παραπάνω σχήματος συμπεραίνουμε ότι:

Η έξοδος Q οδηγείται μαζί με την είσοδο K σε μια πύλη AND που τροφοδοτεί την είσοδο R και η έξοδος Q' οδηγείται μαζί με την είσοδο J στην άλλη πύλη AND που τροφοδοτεί την είσοδο S. Η λειτουργία του είναι ακριβώς ίδια με εκείνη του RS flip-flop, με τη διαφορά ότι το κύκλωμα αυτό δεν έχει απροσδιόριστη κατάσταση όταν και οι δύο εισοδοί του είναι 1. Για J=1 και K=0 η έξοδος Q=1. Για J=0 και K=1 η έξοδος Q=0. Όταν J=K=1 και ο παλμός του ρολογιού είναι 1 (CLK=1), τότε η έξοδος Q μεταβαίνει κάθε φορά στη συμπληρωματική τιμή στην οποία ήταν, δηλαδή από 1 γίνεται 0 ή από 0 γίνεται 1.

T flip-flop

Το T flip-flop μπορεί να κατασκευαστεί από το JK flip-flop συνδέοντας τις δύο εισόδους J και K σε μία είσοδο την T, όπως φαίνεται στο παρακάτω σχήμα.



T	CLK	Q	Q'	Παρατηρήσεις
0	1	Q	Q'	Καμία αλλαγή στην κατάσταση
1	1	Q'	Q	Αντιστροφή κατάστασης

Όταν T=0 και CLK=1 οι έξοδοι Q και Q' παραμένουν στην προηγούμενη κατάσταση, ενώ όταν T=1 οι έξοδοι Q και Q' μεταβαίνουν στη συμπληρωματική τους τιμή.

Απαριθμητές (counters)

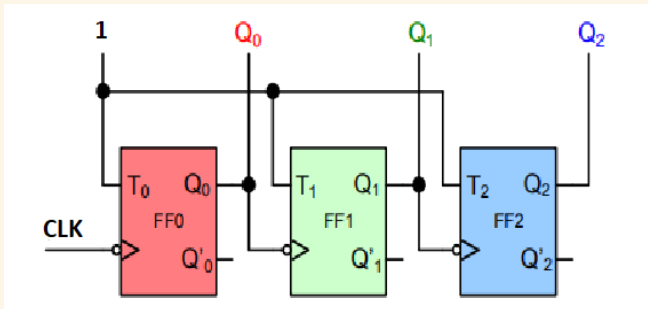
Μια σημαντική εφαρμογή των flip-flop είναι τα ακολουθιακά κυκλώματα απαριθμητών. Οι απαριθμητές χρησιμοποιούνται στα ψηφιακά συστήματα για, να μετρούν τις φορές που εμφανίστηκαν κάποια γεγονότα, για να δημιουργούν χρονικά διαστήματα για τον έλεγχο διαφόρων εργασιών ενός συστήματος κ.ο.κ. Ένας

απαριθμητής αποτελείται από flip-flop συνδεδεμένα σε σειρά. Η λειτουργία του είναι να μετράει στην έξοδο του τους παλμούς ρολογιού (CLK) που δέχθηκε στην είσοδο του.

Υπάρχουν δύο βασικές κατηγορίες απαριθμητών: οι ασύγχρονοι και οι σύγχρονοι.

Στους ασύγχρονους απαριθμητές η έξοδος κάθε flip-flop είναι η είσοδος του επόμενου.

Το παρακάτω σχήμα δείχνει ένα ασύγχρονο δυαδικό απαριθμητή των τριών (3) bit αύξουσας αρίθμησης κατασκευασμένο με T flip-flop.



Από το παραπάνω σχήμα παρατηρούμε τα εξής:

1. Το πρώτο flip-flop (FF0) δέχεται παλμό ρολογιού (CLK) από εξωτερικό ρολόι, ενώ στα υπόλοιπα ο παλμός ρολογιού είναι η έξοδος Q του προηγούμενου flip-flop.
2. Η είσοδος T όλων των flip-flop είναι 1.
3. Η έξοδος του πρώτου flip-flop (FF0) αντιστοιχεί στο ελάχιστο σημαντικό bit (LSB) του απαριθμητή και η έξοδος του FF2 αντιστοιχεί στο πλέον σημαντικό bit (MSB).
4. Ένας απαριθμητής που αποτελείται από n flip-flop μπορεί να μετρήσει 2^n διαφορετικές δυαδικές τιμές. Ο παραπάνω απαριθμητής αποτελείται από 3 flip-flop και μπορεί να μετρήσει έως $2^3=8$ διαφορετικές δυαδικές τιμές (από 000 έως 111). Μόλις η αρίθμηση φτάσει την τιμή 111, τότε με τον επόμενο παλμό ρολογιού η μέτρηση θα ξεκινήσει πάλι από την αρχή (000).

Μπορούμε να καταλάβουμε τη λειτουργία του απαριθμητή από τον παρακάτω πίνακα.

CLK	Q ₀ (LSB)	Q ₁	Q ₂ (MSB)
0	0	0	0
1	1	0	0
1	0	1	0
1	1	1	0
1	0	0	1
1	1	0	1
1	0	1	1
1	1	1	1

Πίνακας 1.4.4.γ Δυαδική ακολουθία μέτρησης ενός απαριθμητή 3-bit

Η αρχική κατάσταση των εξόδων είναι 0.

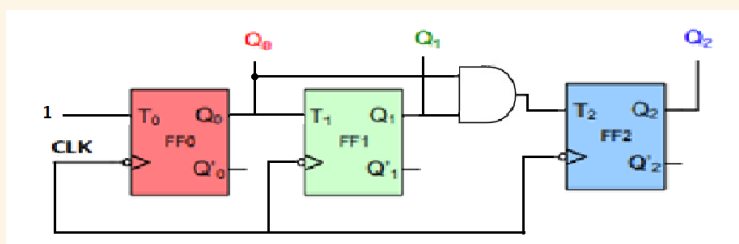
Η έξοδος Q₀ αντιστρέφει την τιμή της σε κάθε παλμό ρολογιού, η έξοδος Q₁ αντιστρέφει την τιμή της κάθε φορά που η έξοδος Q₀ μεταβαίνει από την τιμή 1 στην τιμή 0 και η έξοδος Q₂ αντιστρέφει την τιμή της κάθε φορά που η έξοδος Q₁ μεταβαίνει από την τιμή 1 στην τιμή 0.

Για να κατασκευάσουμε έναν ασύγχρονο απαριθμητή φθίνουσας αρίθμησης εργαζόμαστε με τον ίδιο τρόπο με τη μόνη διαφορά ότι σε όλα τα flip-flop εκτός από το πρώτο (FF0), ο παλμός ρολογιού είναι η έξοδος Q' του προηγούμενου flip-flop.

Εκτός από τους δυαδικούς απαριθμητές μπορούν να κατασκευαστούν απαριθμητές που το περιεχόμενό τους αυξάνεται, ή ελαττώνεται, κατά μια μονάδα, ανάλογα με το αριθμητικό σύστημα που χρησιμοποιούν (δεκαδικό, δεκαεξαδικό κ.τ.λ), με τη βοήθεια λογικών πυλών που επιδρούν κατάλληλα στις εξόδους του απαριθμητή.

Οι σύγχρονοι απαριθμητές διαφέρουν από τους ασύγχρονους στο ότι ένας κοινός παλμός ρολογιού εφαρμόζεται στις εισόδους όλων των flip-flop ταυτόχρονα. Με αυτό τον τρόπο πετυχαίνουμε μεγαλύτερο ρυθμό απαρίθμησης αφού όλα τα flip-flop αλλάζουν κατάσταση ταυτόχρονα, και όχι το ένα μετά το άλλο. Όμως η βελτίωση του χρόνου απόκρισης έχει σαν κόστος την πολυπλοκότητα του κυκλώματος αφού απαιτείται μεγαλύτερος αριθμός λογικών πυλών.

Το παρακάτω σχήμα δείχνει ένα σύγχρονο δυαδικό απαριθμητή των τριών (3) bit αύξουσας αρίθμησης κατασκευασμένο με T flip-flop και μια πύλη AND.



Για να κατασκευάσουμε ένα σύγχρονο απαριθμητή φθίνουσας αρίθμησης εργαζόμαστε με τον ίδιο τρόπο με τις εξής διαφορές:

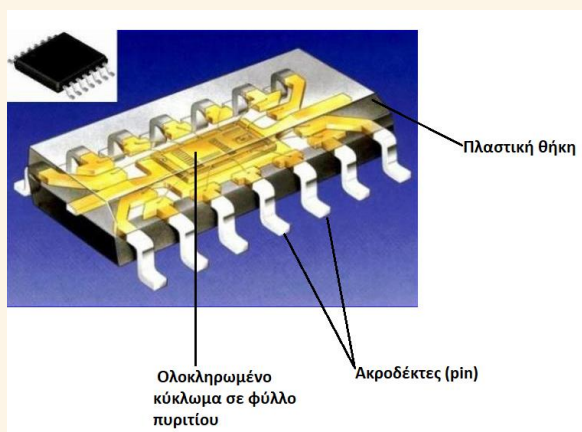
- Οι εισόδους της πύλης AND δεν είναι οι εξόδους Q_0 και Q_1 , αλλά οι συμπληρωματικές τους Q_0' και Q_1' .
- Η έξοδος Q_0' είναι η είσοδος του 2^{ου} flip-flop (FF1).

1.4.5 Ολοκληρωμένα κυκλώματα

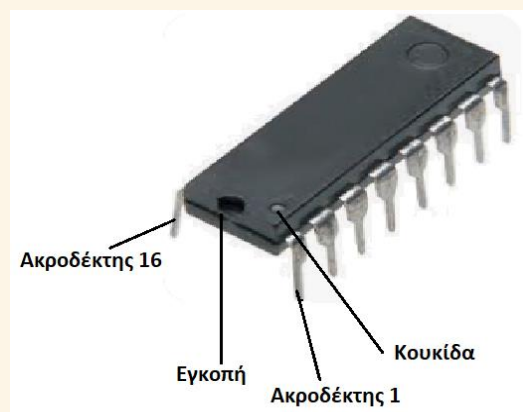
Τα ψηφιακά ολοκληρωμένα κυκλώματα (Digital Integrated Circuits - IC) χρησιμοποιούνται σχεδόν σε όλες τις ηλεκτρονικές συσκευές, όπως οι υπολογιστές, τα αυτοκίνητα, οι τηλεοράσεις, τα κινητά τηλέφωνα, οι ψηφιακές οικιακές συσκευές κτλ. και θεωρούνται επανάσταση στον τομέα της ηλεκτρονικής.

Ολοκληρωμένο ονομάζεται ένα κύκλωμα συνδεδεμένων λογικών πυλών, δημιουργημένο πάνω σε ένα φύλλο ημιαγωγικού υλικού, κατά κύριο λόγο πυρίτιο. Το φύλλο (ημιαγωγού) ονομάζεται στα αγγλικά τσιπ (chip), από το οποίο προκύπτει μια εναλλακτική ονομασία του ολοκληρωμένου κυκλώματος. Όταν αυτό το φύλλο είναι της κλίμακας των μικρομέτρων ονομάζεται και μικροτσιπ. Το chip περιλαμβάνει ηλεκτρικά στοιχεία όπως τρανζίστορς, διόδους αντιστάσεις και πυκνωτές. Τα στοιχεία συνδέονται μεταξύ τους μέσα στο chip ώστε να σχηματίζουν ένα ηλεκτρονικό κύκλωμα και δεν είναι δυνατός ο διαχωρισμός τους ή η αντικατάστασή τους από άλλα στοιχεία.

Μετά τη δημιουργία του, το τσιπ τοποθετείται σε προστατευτική πλαστική ή κεραμική συσκευασία από την οποία εξέρχουν μόνο οι ακροδέκτες (pins) για να μπορεί το ολοκληρωμένο κύκλωμα να συνδεθεί με άλλες συσκευές. Στην Εικόνα 1.4.5.α φαίνεται η εσωτερική όψη ενός ολοκληρωμένου κυκλώματος σε συσκευασία ακροδεκτών διπλής σειράς (Dual In-line Package - DIP).



Εικόνα 1.4.5



Εικόνα 1.4.5.α

Οι ακροδέκτες (pins) αριθμούνται ξεκινώντας από τον ακροδέκτη που βρίσκεται δίπλα στην κουκίδα ή από τον ακροδέκτη που βρίσκεται αριστερά της εγκοπής (Εικόνα 1.4.5.β). Ο αριθμός των pins σε μια συσκευασία μπορεί να είναι από 14 έως και 68 ακροδέκτες. Για μεγάλα τσιπ χρησιμοποιούνται τετραγωνικές συσκευασίες με ακροδέκτες και στις τέσσερις πλευρές.

Τα πλεονεκτήματα των ολοκληρωμένων κυκλωμάτων είναι το χαμηλό κόστος κατασκευής τους, η χαμηλή κατανάλωση ενέργειας, το πολύ μικρό τους μέγεθος και η μεγάλη αξιοπιστία στη λειτουργία τους. Το μικρό τους μέγεθος βοηθά σημαντικά στην κατασκευή μικρού μεγέθους και βάρους ηλεκτρονικές συσκευές, όπως κινητά τηλέφωνα, τηλεοράσεις, ηλεκτρονικοί υπολογιστές κ.ά.

Σαν μειονεκτήματα των ολοκληρωμένων κυκλωμάτων μπορούν να θεωρηθούν η αδυναμία επισκευής του κυκλώματος, όταν χαλάσει έστω και ένα από τα εξαρτήματα που αποτελούν το IC, πρέπει να αντικατασταθεί ολόκληρο το κύκλωμα. Ένα άλλο μειονέκτημα είναι ότι δεν μπορούν να χρησιμοποιηθούν σε κυκλώματα μεγάλης ισχύος. Οι μεγάλες τάσεις μπορεί να διασπάσουν το μονωτικό υλικό μεταξύ των στοιχείων και να δημιουργηθεί βραχυκύκλωμα.

Τα πλεονεκτήματα τους υπερτερούν των μειονεκτημάτων τους, γι' αυτό και χρησιμοποιούνται στις σύγχρονες ηλεκτρονικές κατασκευές.

Κατηγορίες ολοκληρωμένων κυκλωμάτων

Τα ολοκληρωμένα κυκλώματα χωρίζονται σε κατηγορίες ανάλογα με τον αριθμό των λογικών πυλών από τα οποία αποτελούνται. Οι κατηγορίες αυτές είναι:

Μικρής κλίμακας ολοκλήρωσης (Small Scale Integration SSI) που περιλαμβάνουν από 1 έως 10 λογικές πύλες ανά κύκλωμα.

Μεσαίας κλίμακας ολοκλήρωσης (Medium – Scale Integration MSI) που περιλαμβάνουν από 10 έως 100 λογικές πύλες ανά κύκλωμα.

Μεγάλης κλίμακας ολοκλήρωσης (Large – Scale Integration LSI) που περιλαμβάνουν από 100 έως 10.000 λογικές πύλες ανά κύκλωμα.

Πολύ μεγάλης κλίμακας ολοκλήρωσης (Very Large – Scale Integration VLSI) που περιλαμβάνουν από 10.000 έως 100.000 λογικές πύλες ανά κύκλωμα.

Πάρα πολύ μεγάλης κλίμακας ολοκλήρωσης (Ultra Large Scale Integration) που περιλαμβάνει περισσότερες από 100.000 λογικές πύλες ανά κύκλωμα.

Οικογένειες ολοκληρωμένων κυκλωμάτων

Εκτός από την κατηγοριοποίηση με βάση την κλίμακα ολοκλήρωσης, τα ολοκληρωμένα κυκλώματα κατατάσσονται σε λογικές οικογένειες που έχουν κοινά χαρακτηριστικά, υλοποιούν όμως διαφορετικές λογικές συναρτήσεις.

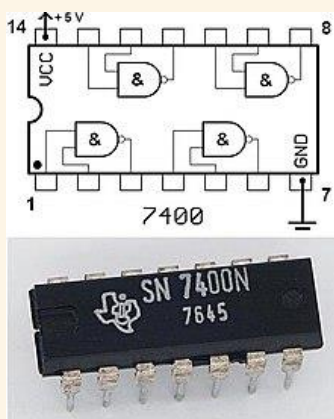
Οι πρώτες οικογένειες που εμφανίστηκαν στη δεκαετία του 1960 ήταν η RTL (Resistor-Transistor Logic) και η DTL (Diode- Transistor Logic), οι οποίες σήμερα δεν χρησιμοποιούνται. Άλλες οικογένειες είναι: ECL (Emitter Coupled Logic), η HTL (High Threshold Logic), η TTL (Transistor-Transistor Logic), η MOS (Metal Oxide Semiconductor) , η CMOS (Complementary MOS) και η BICMOS (Bipolar CMOS).

Στα σύγχρονα ηλεκτρονικά χρησιμοποιούνται κυρίως δύο οικογένειες ολοκληρωμένων κυκλωμάτων, η οικογένεια TTL και η οικογένεια CMOS.

Αναγνώριση ολοκληρωμένων κυκλωμάτων

Κάθε ολοκληρωμένο κύκλωμα, ανάλογα με την εταιρεία που το έχει κατασκευάσει, φέρει ένα κωδικό στο πάνω μέρος της συσκευασίας του που δηλώνει τα λειτουργικά χαρακτηριστικά του και την κατασκευάστρια εταιρεία. Επίσης, κάθε ολοκληρωμένο κύκλωμα συνοδεύεται από τα φύλλα οδηγιών (data sheets) του κατασκευαστή του όπου εκεί μπορούμε να αναζητήσουμε λεπτομερείς πληροφορίες για το κύκλωμα για να μπορέσουμε να το αξιοποιήσουμε.

Στην Εικόνα 1.4.5.δ βλέπουμε το ολοκληρωμένο κύκλωμα με κωδικό SN7400N. Το πρόθεμα SN υποδηλώνει ότι κατασκευάστηκε από την εταιρεία Texas Instruments. Ο αριθμός 74 υποδηλώνει ότι το κύκλωμα είναι της σειράς 7400 και επίσης δηλώνει την περιοχή θερμοκρασίας μέσα στην οποία το κύκλωμα λειτουργεί σύμφωνα με τις προδιαγραφές του κατασκευαστή (0°C - 70°C). Ο αριθμός 00 δηλώνει ότι το κύκλωμα περιέχει τέσσερις πύλες NAND δύο εισόδων. Το επίθημα N είναι ένας κωδικός του συγκεκριμένου προμηθευτή που υποδεικνύει πλαστική συσκευασία με δύο παράλληλες σειρές ακροδεκτών. Η δεύτερη σειρά αριθμών (7645) είναι ένας κωδικός ημερομηνίας. αυτό το τσιπ κατασκευάστηκε την 45η εβδομάδα του 1976.



Εικόνα 1.4.5.β