Κεφάλαιο 3.

Εισαγωγή στην Ψηφιακή Σχεδίαση

Περιεχόμενα

Εισαγωγή

Στο Κεφάλαιο αυτό θα πραγματοποιήσουμε μια εισαγωγή στο αντικείμενο της ψηφιακής σχεδίασης, στον τρόπο δηλαδή με τον οποίο χρησιμοποιώντας άλγεβρα Boole και τις βασικές λογικές πύλες μπορούμε να συνθέσουμε πιο σύνθετες μονάδες, όπως πολυπλέκτες, στοιχεία μνήμης κλπ. Η γενική κατεύθυνση είναι να δείξουμε πώς ξεκινώντας από λογικά στοιχεία μπορούμε να «χτίσουμε» μεγαλύτερα ψηφιακά συστήματα. Οι γνώσεις που θα αποκτήσουμε στο Κεφάλαιο αυτό θα χρησιμοποιηθούν στη συνέχεια όπου θα περιγράψουμε τη σχεδίαση ενός απλού επεξεργαστή χρησιμοποιώντας στοιχεία λογικής.

3.1 Άλγεβρα Boole

Η δίτιμη άλγεβρα Boole είναι ένα αλγεβρικό σύστημα το οποίο περιλαμβάνει δύο στοιχεία (τα οποία συμβολίζουμε συνήθως με '0' και '1') και τρεις τελεστές, τους οποίους συμβολίζουμε με + (OR, Η), . (AND, ΚΑΙ) ~ (NOT, ΟΧΙ). Συχνά, αντί για το σύμβολο ~ χρησιμοποιούμε το σύμβολο ' (~x = x'). Οι κανόνες για τους τελεστές αυτούς ορίζονται σύμφωνα με τους ακόλουθους πίνακες.

Πίνακας: *Τελεστές της δίτιμης άλγεβρας Boole*

x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1

x	y	x.y
0	0	0
0	1	0
1	0	0
1	1	1

x	~x
0	1
1	0

Οι μεταβλητές της άλγεβρας Boole μπορούν να πάρουν δύο τιμές ‘0’ ή ‘1’. Η άλγεβρα Boole χαρακτηρίζεται από μια σειρά αξιώματα, θεωρήματα και ιδιότητες (αντιμεταθετική, προσεταιριστική κ.λπ.).

Μια συνάρτηση Boole είναι μια έκφραση που σχηματίζεται από δυαδικές μεταβλητές, τους τελεστές Η, ΚΑΙ, ΟΧΙ και παρενθέσεις. Για μια δεδομένη τιμή των μεταβλητών η τιμή της συνάρτησης μπορεί να είναι είτε ‘0’ είτε ‘1’. Μια συνάρτηση Boole μπορεί να οριστεί είτε με τον τύπο της, είτε με τον πίνακα αληθείας της. Για παράδειγμα, η συνάρτηση

$$F = xyz + xyz'$$

Παίρνει την τιμή ‘1’ είτε όταν x=y=z=1 είτε όταν x=y=1, και z=0 (με z‘ συμβολίζουμε την αντίστροφη της μεταβλητής z, η οποία είναι 0 όταν η z είναι 1 και αντίστροφα).

Ο πίνακας αληθείας της συνάρτησης F δίνεται στη συνέχεια.

Πίνακας: *Πίνακας αληθείας της συνάρτησης F = xyz + xyz’*

x	y	z	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

3.2 Λογικές Πύλες

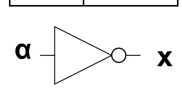
Μια λογική πύλη υλοποιεί μια συνάρτηση Boole και έχει μία, δύο ή περισσότερες εισόδους και μία έξοδο.

Κάθε λογική πύλη έχει ένα σύμβολο και έναν πίνακα αληθείας. Ο πίνακας αυτός δείχνει την τιμή της εξόδου για τις διαφορετικές τιμές των εισόδων. Οι λογικές πύλες χρησιμοποιούνται ακόμη στην υλοποίηση των άλλων μονάδων (αποκωδικοποιητών πολυπλεκτών, αθροιστών, κ.λπ.) που θα περιγραφούν στη συνέχεια.

Οι πιο γνωστές λογικές πράξεις είναι η σύζευξη (AND), η διάζευξη (OR) και η άρνηση (NOT). Οι πράξεις αυτές υλοποιούνται με τις λογικές πύλες AND, OR και NOT αντίστοιχα. Ακόμη, πολύ συχνά χρησιμοποιούνται οι πύλες NAND και NOR, οι οποίες εκτελούν τις αντίστροφες λογικές πράξεις από τις AND και OR αντίστοιχα, καθώς και οι πύλες της αποκλειστικής διάζευξης (XOR) και της άρνησής της (XNOR).

Πύλη NOT: Η λογική πράξη της άρνησης έχει μια μεταβλητή εισόδου και ένα αποτέλεσμα το οποίο είναι ‘1’ αν η είσοδος είναι ‘0’, διαφορετικά το αποτέλεσμα είναι ‘0’. Στο επόμενο Σχήμα παρουσιάζουμε το λογικό σύμβολο της πύλης NOT, καθώς και τον πίνακα αληθείας της.

α	x
0	1
1	0

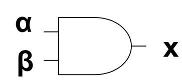


Σχήμα: *Λογικό σύμβολο της πύλης NOT και πίνακας αληθείας*

Συχνά, στην υλοποίηση λογικών συναρτήσεων, αντί για το σύμβολο της πύλης NOT χρησιμοποιούμε για απλότητα ένα κυκλάκι.

Πύλη AND: Η λογική πράξη της σύζευξης έχει δύο (ή περισσότερους) τελεστές και ένα αποτέλεσμα το οποίο είναι '1' αν όλα τα δεδομένα είναι '1'. Σε κάθε άλλη περίπτωση, το αποτέλεσμα είναι '0'. Η πράξη αυτή υλοποιείται με την πύλη AND. Στο επόμενο Σχήμα παρουσιάζουμε το λογικό σύμβολο της πύλης AND δύο εισόδων, καθώς και τον πίνακα αληθείας της.

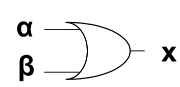
α	β	x
0	0	0
0	1	0
1	0	0
1	1	1



***Σχήμα**: Λογικό σύμβολο της πύλης AND και πίνακας αληθείας*

Πύλη OR: Η λογική πράξη της διάζευξης έχει δύο (ή περισσότερους) τελεστές και ένα αποτέλεσμα το οποίο είναι '0' αν όλα τα δεδομένα είναι '0'. Σε κάθε άλλη περίπτωση, το αποτέλεσμα είναι '1'. Η πράξη αυτή υλοποιείται με την πύλη OR. Στο επόμενο Σχήμα παρουσιάζουμε το λογικό σύμβολο της πύλης OR δύο εισόδων, καθώς και τον πίνακα αληθείας της.

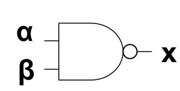
α	β	x
0	0	0
0	1	1
1	0	1
1	1	1



***Σχήμα**: Λογικό σύμβολο της πύλης OR και πίνακας αληθείας*

Πύλη NAND: Η λογική πράξη της άρνησης της σύζευξης έχει δύο (ή περισσότερους) τελεστές και ένα αποτέλεσμα το οποίο είναι '0' αν όλα τα δεδομένα είναι '1'. Σε κάθε άλλη περίπτωση, το αποτέλεσμα είναι '1'. Η πράξη αυτή υλοποιείται με την πύλη NAND. Στο επόμενο Σχήμα παρουσιάζουμε το λογικό σύμβολο της πύλης NAND δύο εισόδων, καθώς και τον πίνακα αληθείας της.

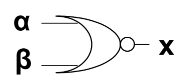
α	β	x
0	0	1
0	1	1
1	0	1
1	1	0



***Σχήμα**: Λογικό σύμβολο της πύλης NAND και πίνακας αληθείας*

Πύλη NOR: Η λογική πράξη της άρνησης της διάζευξης έχει δύο (ή περισσότερους) τελεστές και ένα αποτέλεσμα το οποίο είναι '1' αν όλα τα δεδομένα είναι '0'. Σε κάθε άλλη περίπτωση, το αποτέλεσμα είναι '1'. Στο επόμενο Σχήμα παρουσιάζουμε το λογικό σύμβολο της πύλης NOR δύο εισόδων, καθώς και τον πίνακα αληθείας της.

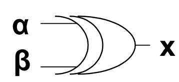
α	β	x
0	0	1
0	1	0
1	0	0
1	1	0



***Σχήμα**: Λογικό σύμβολο της πύλης NOR και πίνακας αληθείας*

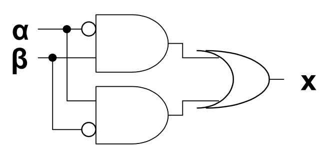
Πύλη XOR: Η λογική πράξη της αποκλειστικής διάζευξης με δύο τελεστές έχει αποτέλεσμα ‘1’ αν οι δύο τελεστές έχουν διαφορετική τιμή και ‘0’ διαφορετικά. Η πράξη αυτή υλοποιείται με την πύλη XOR. Στο επόμενο Σχήμα παρουσιάζουμε το λογικό σύμβολο της πύλης XOR δύο εισόδων, καθώς και τον πίνακα αληθείας της.

α	β	x
0	0	0
0	1	1
1	0	1
1	1	0



***Σχήμα**: Λογικό σύμβολο της πύλης XOR και πίνακας αληθείας*

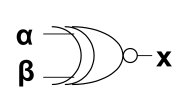
Συνήθως, η λογική πράξη της αποκλειστικής διάζευξης συμβολίζεται με το $\oplus$, δηλαδή $x=a\oplus b$. Μπορεί κανείς να παρατηρήσει ότι η λογική συνάρτηση την οποία υλοποιεί η πύλη XOR είναι η $x=a'b + ab'$. Έτσι, η πύλη XOR είναι ισοδύναμη με την ακόλουθη υλοποίηση.



***Σχήμα**: Πύλη XOR υλοποιημένη με πύλες AND, OR, NOT*

Πύλη XNOR: Η λογική πράξη της άρνησης της αποκλειστικής διάζευξης με δύο τελεστές έχει αποτέλεσμα ‘0’ αν οι δύο τελεστές έχουν διαφορετική τιμή και ‘1’ διαφορετικά. Η πράξη αυτή υλοποιείται με την πύλη XNOR. Στο επόμενο Σχήμα παρουσιάζουμε το λογικό σύμβολο της πύλης XNOR δύο εισόδων, καθώς και τον πίνακα αληθείας της.

α	β	x
0	0	1
0	1	0
1	0	0
1	1	1



***Σχήμα**: Λογικό σύμβολο της πύλης XNOR και πίνακας αληθείας*

3.3 Απομονωτές και transceivers

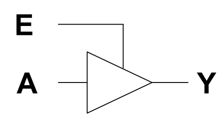
Όπως είδαμε στο προηγούμενο Κεφάλαιο, σε ένα υπολογιστικό σύστημα, υπάρχει ένας διάδρομος, πάνω στον οποίο συνδέονται τα τμήματα του υπολογιστικού συστήματος (ΚΜΕ, μνήμη, συσκευές εισόδου-εξόδου κ.λπ.). Είδαμε επίσης ότι κάθε στιγμή μόνο δύο συσκευές μπορούν να ανταλλάσσουν δεδομένα μέσω του διαδρόμου (η μία να στέλνει δεδομένα και η άλλη να λαμβάνει).

Για να μπορέσει να πραγματοποιηθεί αυτή η λειτουργία, πρέπει να υπάρχει ένας τρόπος ώστε οι έξοδοι (ή είσοδοι) μιας συσκευής ή μονάδας που είναι συνδεδεμένη στο διάδρομο να απομονωθούν, να λειτουργήσουν δηλαδή με τέτοιο τρόπο ώστε η συσκευή ή μονάδα να μην επηρεάζει καθόλου (και να μην επηρεάζεται από) το κύκλωμα στο οποίο συνδέεται. Για να πραγματοποιηθεί αυτό, στους ακροδέκτες κάθε μιας από τις συσκευές που είναι συνδεδεμένες στο διάδρομο συνδέεται ένας απομονωτής (buffer) ή transceiver.

Ένας απομονωτής επιτρέπει μεταφορά δεδομένων μόνο προς τη μία κατεύθυνση, ενώ ένας transceiver επιτρέπει μεταφορά σημάτων και προς τις δύο κατευθύνσεις, με την ενεργοποίηση κατάλληλου σήματος επιλογής.

Όταν η έξοδος ενός απομονωτή ή transceiver δεν είναι ενεργοποιημένη, δεν επηρεάζει καθόλου το κύκλωμα στο οποίο συνδέεται (λέμε ότι έχει άπειρη αντίσταση). Οι απομονωτές (buffers) χρησιμοποιούνται ακόμη ως ενισχυτές του διαδρόμου διευθύνσεων.

Απομονωτής: Ο απομονωτής έχει μία είσοδο δεδομένων (A), μια είσοδο ελέγχου (E) και μια έξοδο (Y). Όταν η είσοδος ελέγχου βρίσκεται στη λογική κατάσταση ‘1’ (ενεργοποίηση), η είσοδος A μεταφέρεται στην έξοδο. Αν η είσοδος ελέγχου είναι στη λογική κατάσταση ‘0’ (απαγόρευση), η έξοδος είναι σε κατάσταση υψηλής αντίστασης (Three-state, TS).

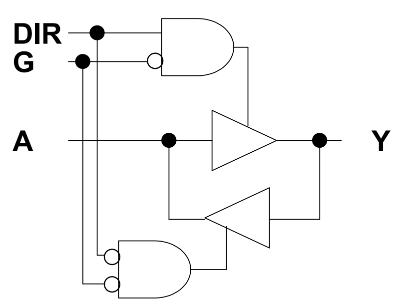


E	Y
1	A
0	TS

***Σχήμα**: Απομονωτής και πίνακας λειτουργίας*

Για την ενίσχυση γραμμών στις οποίες είναι συνδεδεμένες μονάδες διπλής κατεύθυνσης (πχ. μονάδες εισόδου-εξόδου) χρησιμοποιούνται ειδικοί απομονωτές που ονομάζονται transceivers (transmitters/receivers).

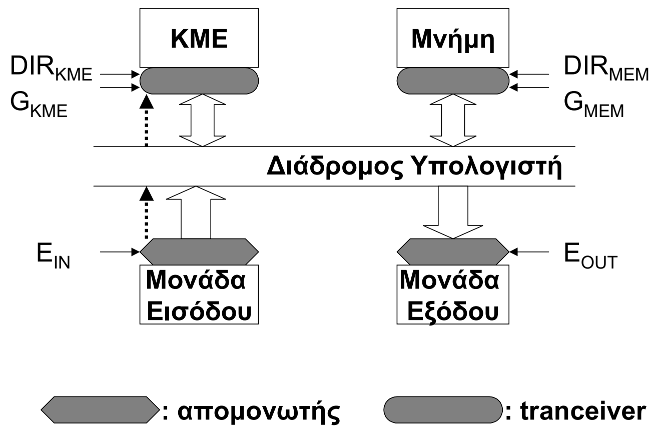
Ο transceiver διαθέτει δύο εισόδους ελέγχου G και DIR. Όταν η είσοδος G (ενεργοποίησης) είναι σε λογική κατάσταση ‘0’, τότε ο transceiver λειτουργεί σε κατάσταση απομόνωσης. Όταν G=1, τότε, η μία είσοδος μεταφέρεται στην άλλη ανάλογα με την τιμή του σήματος DIR.



G	**DIR**	**Λειτουργία**
0	x	TS
1	0	A => Y
1	1	Y => A

***Σχήμα**: Transceiver και πίνακας λειτουργίας*

Για να φανεί η χρησιμότητα των απομονωτών και των transceivers στα υπολογιστικά συστήματα, ας θεωρήσουμε το ακόλουθο απλοποιημένο διάγραμμα ενός υπολογιστικού συστήματος, στο οποίο οι μονάδες συνδέονται μέσω ενός διαδρόμου.



***Σχήμα**: Υπολογιστικό σύστημα με απομονωτές και transceivers*

Σε αυτό το υπολογιστικό σύστημα προκειμένου να αποσταλούν δεδομένα από τη μονάδα εισόδου στην ΚΜΕ, πρέπει τα σήματα να πάρουν τις τιμές που φαίνονται στον επόμενο πίνακα.

***Πίνακας**: Λειτουργία των transceivers και απομονωτών για μεταφορά δεδομένων από τη μονάδα εισόδου στην ΚΜΕ*

Σήμα	Τιμή	Επεξήγηση
DIR_{KME}	0	Είσοδος δεδομένων στην ΚΜΕ
G_{KME}	1	Ενεργοποίηση transeiver
DIR_{MEM}	X	
G_{MEM}	0	Απενεργοποίηση μνήμης
E_{IN}	1	Ενεργοποίηση μονάδας εισόδου
E_{OUT}	0	Απενεργοποίηση μονάδας εξόδου

Με τις τιμές των σημάτων που φαίνονται στον προηγούμενο Πίνακα, μόνο η μονάδα εισόδου μπορεί να στέλνει σήματα, και μόνο η ΚΜΕ μπορεί να λαμβάνει τα σήματα αυτά.

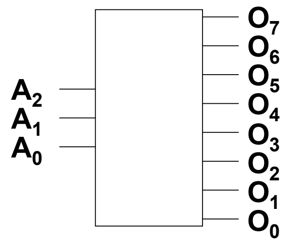
3.4 Κωδικοποιητές και αποκωδικοποιητές

Όταν περισσότερες από δύο συσκευές είναι συνδεδεμένες σε ένα διάδρομο (όπως συμβαίνει στα υπολογιστικά συστήματα), πρέπει να αποφασιστεί ποια από όλες θα επιλεγεί ώστε να πραγματοποιήσει μεταφορά των δεδομένων. Αυτό επιτυγχάνεται με την ενεργοποίηση κατάλληλου σήματος. Αν ο αριθμός των συσκευών που είναι συνδεδεμένες στο διάδρομο είναι μεγάλος, τότε μπορούμε να αποδώσουμε ένα δυαδικό αριθμό σε κάθε συσκευή που θα αποτελεί τη διεύθυνσή της. Έτσι με λίγες γραμμές (π.χ. 3) μπορούμε να ελέγξουμε μεγάλο αριθμό συσκευών (π.χ. 8). Προκύπτει όμως η ανάγκη κυκλωμάτων τα οποία θα μετατρέπουν την πληροφορία από τις λίγες γραμμές στις πολλές. Τα κυκλώματα αυτά ονομάζονται κυκλώματα αποκωδικοποίησης ή αποκωδικοποιητές.

3.4.1 Αποκωδικοποιητές

Ένας αποκωδικοποιητής είναι ένα κύκλωμα (μονάδα) που έχει n γραμμές εισόδου και 2^n γραμμές εξόδου. Κάθε στιγμή μόνο μια από τις 2^n γραμμές εξόδου είναι ενεργή (έχει την τιμή '1'), ανάλογα με το συνδυασμό των τιμών εισόδου.

Για παράδειγμα, στο επόμενο Σχήμα φαίνεται το λογικό διάγραμμα και ο πίνακας αληθείας του αποκωδικοποιητή 3-σε-8.

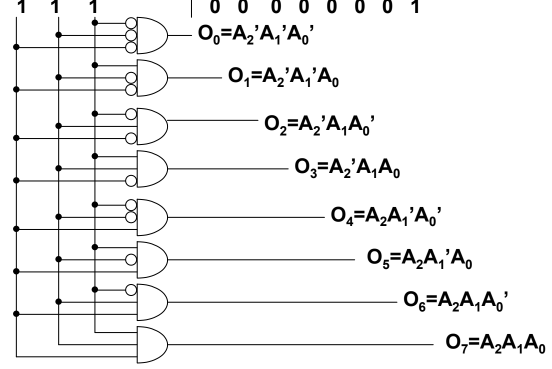


$A_2A_1A_0$	$O_7O_6O_5O_4O_3O_2O_1O_0$
0 0 0	0 0 0 0 0 0 0 1
0 0 1	0 0 0 0 0 0 1 0
0 1 0	0 0 0 0 0 1 0 0
0 1 1	0 0 0 0 1 0 0 0
1 0 0	0 0 0 1 0 0 0 0
1 0 1	0 0 1 0 0 0 0 0
1 1 0	0 1 0 0 0 0 0 0
1 1 1	1 0 0 0 0 0 0 0

***Σχήμα**: Αποκωδικοποιητής 3-σε-8 και πίνακας αληθείας*

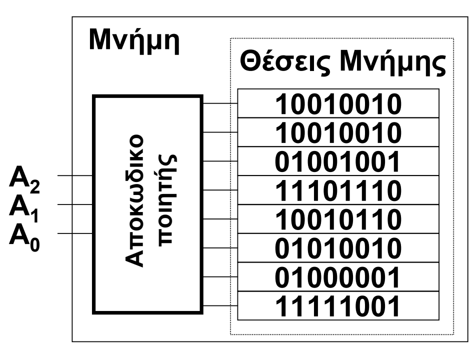
Είναι εύκολο να εξάγει κανείς τη συνάρτηση Boole για κάθε μια από τις εξόδους O_0 έως O_7 του αποκωδικοποιητή. Στο επόμενο Σχήμα παρουσιάζουμε πως μπορούμε να υλοποιήσουμε έναν αποκωδικοποιητή χρησιμοποιώντας πύλες AND και αντιστροφείς. Στο σχήμα αυτό, με κυκλάκι συμβολίζουμε την αντιστροφή (πύλη NOT). Ακόμη, με A’ συμβολίζουμε την αντίστροφη τιμή της μεταβλητής A.

A_2	A_1	A_0	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



***Σχήμα**: Υλοποίηση αποκωδικοποιητή 3-σε-8 με πύλες AND και NOT*

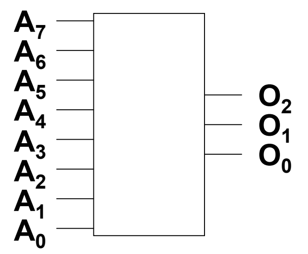
Αποκωδικοποιητές χρησιμοποιούνται στην επιλογή μιας θέσης μνήμης. Οπως γνωρίζουμε, η μνήμη αποτελείται από θέσεις, κάθε μια από τις οποίες έχει μια διεύθυνση. Γνωρίζουμε ακόμη, οτι η επιλογή της θέσης μνήμης γίνεται μέσω του διαδρόμου διευθύνσεων. Αν ο διάδρομος διευθύνσεων αποτελείται από n καλώδια, μπορεί να διευθυνσιοδοτήσει μέχρι 2^n θέσεις μνήμης. Η επιλογή της επιθυμητής θέσης μνήμης γίνεται μέσα στη μονάδα μνήμης, χρησιμοποιώντας έναν αποκωδικοποιητή.



***Σχήμα**: Μνήμη 8 θέσεων που διεθυνσιοδοτείται από 3 γραμμές διευθύνσεων χρησιμοποιώντας αποκωδικοποιητή*

3.4.2 Κωδικοποιητές

Ένας κωδικοποιητής είναι ένα λογικό κύκλωμα με 2^n εισόδους και n εξόδους. Για να λειτουργήσει ο κωδικοποιητής, κάθε στιγμή μόνο μια από τις εισόδους μπορεί να είναι ενεργοποιημένη. Η λογική τιμή της εξόδου εξαρτάται από το συνδυασμό των εισόδων. Στο επόμενο Σχήμα φαίνεται το λογικό διάγραμμα και ο πίνακας αληθείας ενός κωδικοποιητή 8-σε-3.

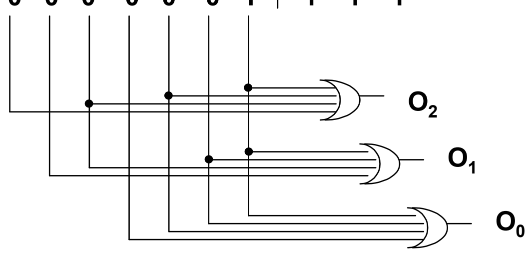


$A_7A_6A_5A_4A_3A_2A_1A_0$	$O_2O_1O_0$
0 0 0 0 0 0 0 1	0 0 0
0 0 0 0 0 0 1 0	0 0 1
0 0 0 0 0 1 0 0	0 1 0
0 0 0 0 1 0 0 0	0 1 1
0 0 0 1 0 0 0 0	1 0 0
0 0 1 0 0 0 0 0	1 0 1
0 1 0 0 0 0 0 0	1 1 0
1 0 0 0 0 0 0 0	1 1 1

***Σχήμα**: Κωδικοποιητής 8-σε-3 και πίνακας αληθείας*

Στο επόμενο Σχήμα παρουσιάζουμε την υλοποίηση ενός κωδικοποιητή 8-σε-3 χρησιμοποιώντας πύλες OR.

A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	O_2	O_1	O_0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1



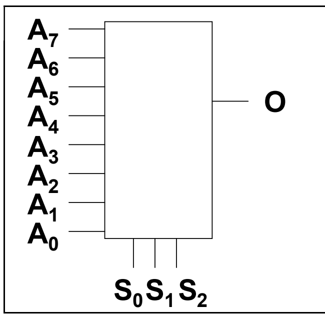
***Σχήμα**: Υλοποίηση κωδικοποιητή 8-σε-3 με πύλες OR*

3.5 Πολυπλέκτες και αποπολυπλέκτες

Ο όρος ‘πολυπλεξία’ σημαίνει τη μεταβίβαση ενός μεγάλου αριθμού πληροφοριών μέσα από ένα μικρότερο αριθμό καναλιών ή γραμμών.

3.5.1 Πολυπλέκτες

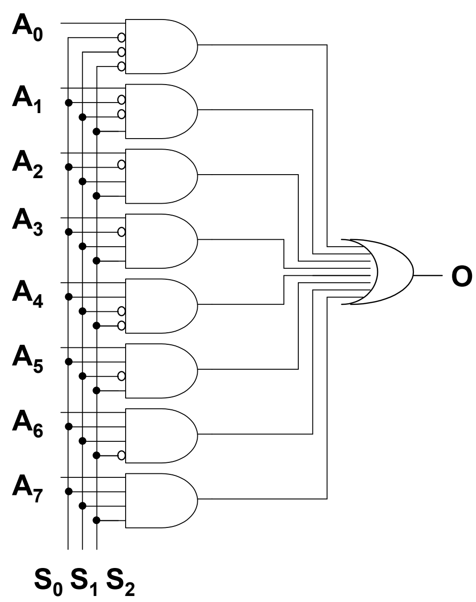
Ένας ψηφιακός πολυπλέκτης είναι ένα συνδυαστικό κύκλωμα που επιλέγει δυαδικές πληροφορίες ανάμεσα σε πολλές γραμμές εισόδου και τις κατευθύνει σε μια μοναδική γραμμή εξόδου. Η επιλογή της γραμμής εισόδου γίνεται μέσω γραμμών επιλογής. Υπάρχουν 2^n γραμμές εισόδου και n γραμμές επιλογής. Η τιμή των γραμμών επιλογής καθορίζει ποια είσοδος επιλέγεται.



$S_2S_1S_0$	O
0 0 0	A_0
0 0 1	A_1
0 1 0	A_2
0 1 1	A_3
1 0 0	A_4
1 0 1	A_5
1 1 0	A_6
1 1 1	A_7

Σχήμα: *Πολυπλέκτης 8 εισόδων και πίνακας αληθείας*

Στο επόμενο Σχήμα παρουσιάζουμε την υλοποίηση ενός πολυπλέκτη 8-σε-1 χρησιμοποιώντας πύλες AND, NOT και μια πύλη OR. Με κυκλάκι συμβολίζουμε την αντιστροφή (πύλη NOT).

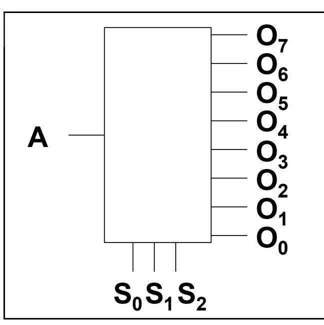


Σχήμα*: Υλοποίηση πολυπλέκτη 8-σε-1 με πύλες AND, NOT, OR*

3.5.2 Αποπολυπλέκτες

Η αποπολυπλεξία είναι η αντίστροφη διαδικασία από την πολυπλεξία. Στην αποπολυπλεξία υπάρχει μια είσοδος και 2^n έξοδοι. Η τιμή της εισόδου μεταφέρεται σε κάποια από τις εξόδους ανάλογα με την τιμή των n σημάτων επιλογής.

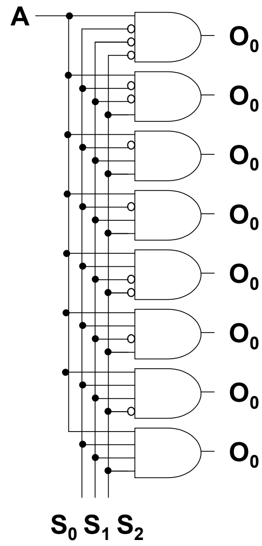
Για παράδειγμα, στο επόμενο Σχήμα παρουσιάζουμε το λογικό διάγραμμα και τον Πίνακα αληθείας για τον αποπολυπλέκτη 2^3=8 εξόδων.



$S_2S_1S_0$	$O_7O_6O_5O_4O_3O_2O_1O_0$
000	X X X X X X X A
001	X X X X X X A X
010	X X X X X A X X
011	X X X X A X X X
100	X X X A X X X X
101	X X A X X X X X
110	X A X X X X X X
111	A X X X X X X X

Σχήμα: *Αποπολυπλέκτης 8 εξόδων και πίνακας αληθείας*

Στο ακόλουθο Σχήμα φαίνεται η υλοποίηση ενός αποπολυπλέκτη 8 εξόδων χρησιμοποιώντας πύλες AND και NOT.



***Σχήμα**: Υλοποίηση αποπολυπλέκτη 8 εξόδων με πύλες AND και NOT*

3.6 Αριθμητικές μονάδες

Όπως αναφέραμε, μια από τις πιο σημαντικές λειτουργίες που εκτελούν τα υπολογιστικά συστήματα είναι οι αριθμητικές πράξεις. Οι πιο σημαντικές πράξεις, πάνω στις οποίες στηρίζονται οι υπόλοιπες, είναι η πρόσθεση και η αφαίρεση. Για την εκτέλεση των πράξεων αυτών χρησιμοποιούνται οι αθροιστές και οι αφαιρέτες. Μια ακόμη ενδιαφέρουσα μονάδα είναι ο πολλαπλασιαστής ο οποίος εκτελεί πολλαπλασιασμό δυαδικών αριθμών.

3.6.1 Αθροιστές

Η πιο βασική αριθμητική πράξη είναι η πρόσθεση δύο δυαδικών ψηφίων. Όπως αναφέραμε, για την πρόσθεση δύο δυαδικών ψηφίων υπάρχουν τέσσερις δυνατές περιπτώσεις: 0+0=0, 0+1=1, 1+0=1, 1+1=10. Οι τρεις πρώτες πράξεις δημιουργούν ένα άθροισμα που το μήκος του είναι ένα ψηφίο. Όταν και οι δύο προσθετέοι είναι 1, το δυαδικό άθροισμα αποτελείται από δύο ψηφία. Το πιο σημαντικό από αυτά τα δύο ψηφία ονομάζεται 'κρατούμενο'. Όταν οι προσθετέοι περιέχουν και άλλα σημαντικά ψηφία, το κρατούμενο που βγαίνει από την πρόσθεση προστίθεται στο επόμενο μεγαλύτερης σημαντικότητας ζευγάρι δυαδικών ψηφίων. Ένα συνδυαστικό κύκλωμα που εκτελεί την πρόσθεση δυο δυαδικών ψηφίων λέγεται 'ημιαθροιστής'. Ένα συνδυαστικό κύκλωμα που εκτελεί την πρόσθεση τριών δυαδικών ψηφίων (δύο σημαντικών ψηφίων και ενός προηγούμενου κρατούμενου) λέγεται 'πλήρης αθροιστής'. Το όνομα του ημιαθροιστή προέρχεται από το γεγονός ότι δύο ημιαθροιστές μπορούν να χρησιμοποιηθούν για να υλοποιήσουν έναν πλήρη αθροιστή, όπως θα δούμε στη συνέχεια.

Ημιαθροιστής

Σύμφωνα με τα προηγούμενα, ο ημιαθροιστής είναι ένα συνδυαστικό κύκλωμα με δύο εξόδους, x και y, και δύο εξόδους s (sum, άθροισμα) και C (Carry, κρατούμενο), του οποίου ο πίνακας αληθείας δίνεται στη συνέχεια.

***Πίνακας**: Πίνακας αληθείας ημιαθροιστή*

x	y	s	C
0	0	0	0
0	1	1	0
1	0	1	0

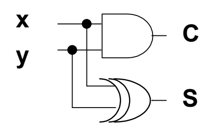
1	1	0	1

Είναι δυνατό να εξάγουμε τις συναρτήσεις Boole από τον πίνακα αληθείας οι οποίες είναι οι ακόλουθες.

$$S = x'y + x\,y' = x \oplus y$$

$$C = xy$$

Όπως έχουμε δει, η συνάρτηση x'y + xy' είναι η πύλη XOR. Το λογικό διάγραμμα της υλοποίησης αυτής χρησιμοποιώντας πύλες NAND και XOR φαίνεται στο ακόλουθο Σχήμα:



***Σχήμα**: Ημιαθροιστής*

Πλήρης Αθροιστής

Ο πλήρης αθροιστής είναι ένα συνδυαστικό κύκλωμα που σχηματίζει το άθροισμα τριών δυαδικών ψηφίων εισόδου. Έχει τρεις εισόδους και δύο εξόδους. Οι δύο από τις μεταβλητές εισόδου παριστάνουν τα δύο σημαντικά ψηφία που προστίθενται. Η τρίτη είσοδος παριστάνει το κρατούμενο από τη λιγότερο σημαντική βαθμίδα. Ο πίνακας αληθείας του πλήρους αθροιστή δίνεται στη συνέχεια:

***Πίνακας**: Πίνακας αληθείας πλήρους αθροιστή*

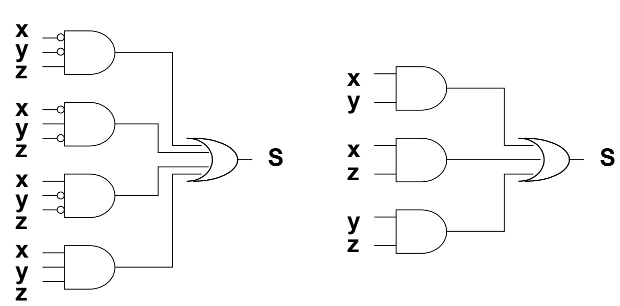
z	x	y	s	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Μπορούμε, χρησιμοποιώντας τον πίνακα αληθείας, να εξάγουμε τις ακόλουθες απλοποιημένες συναρτήσεις για τις συναρτήσεις S και C του πλήρους αθροιστή.

$$S = x'y'z + x'yz' + xy'z' + xyz$$

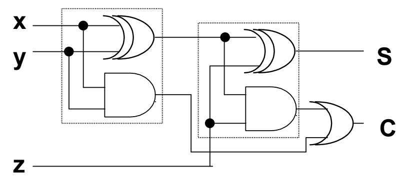
$$C = xy + xz + yz$$

Έτσι, μπορούμε να δώσουμε την ακόλουθη υλοποίηση για το κύκλωμα του πλήρους αθροιστή.



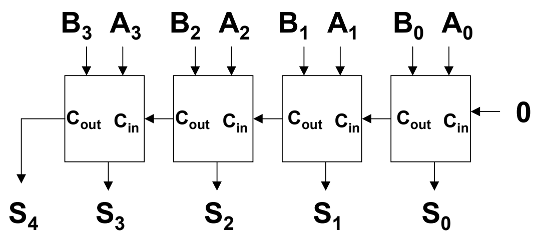
***Σχήμα**: Πλήρης αθροιστής*

Είναι αρκετά ενδιαφέρον να παρατηρήσει κανείς ότι ένα ισοδύναμο σχήμα για τον πλήρη αθροιστή μπορεί να δοθεί με τη βοήθεια ημιαθροιστών, όπως φαίνεται στο επόμενο Σχήμα.



Σχήμα: *Πλήρης αθροιστής υλοποιημένος με ημιαθροιστές*

Προκειμένου να προσθέσουμε δυαδικούς αριθμούς που αποτελούνται από περισσότερα του ενός ψηφία, χρησιμοποιούμε περισσότερους από έναν πλήρεις αθροιστές. Κάθε πλήρης αθροιστής αντιστοιχεί σε ένα ψηφίο αθροίσματος, και το κρατούμενο εξόδου κάθε αθροιστή συνδέεται στο κρατούμενο εισόδου του αθροιστή που αντιστοιχεί στην επόμενη βαθμίδα. Για παράδειγμα, στο επόμενο Σχήμα παρουσιάζεται ένα τέτοιος αθροιστής τεσσάρων βαθμίδων. Ο αθροιστής αυτός προσθέτει τους τετραψήφιους δυαδικούς αριθμούς $A_3A_2A_1A_0$ και $B_3B_2B_1B_0$ και δίνει σαν αποτέλεσμα τον πενταψήφιο δυαδικό αριθμό $S_4S_3S_2S_1S_0$.



Σχήμα: *Αθροιστής 4 δυαδικών ψηφίων*

3.6.2 Αφαιρέτες

Η αφαίρεση δύο δυαδικών αριθμών πραγματοποιείται παίρνοντας το συμπλήρωμα του αφαιρέτη και προσθέτοντάς το στον αφαιρετέο. Με αυτή την παρατήρηση, η πράξη της αφαίρεσης μετατρέπεται σε πρόσθεση και για την υλοποίησή της μπορούν να χρησιμοποιηθούν πλήρεις αθροιστές. Εναλλακτικά, μπορούμε να υλοποιήσουμε την αφαίρεση με άμεσο τρόπο (όπως την κάνουμε με χαρτί και μολύβι). Με αυτή τη μέθοδο, κάθε δυαδικό ψηφίο του αφαιρέτη αφαιρείται από το αντίστοιχης σημαντικότητας δυαδικό ψηφίο του αφαιρετέου και δίνει ένα δυαδικό ψηφίο διαφοράς. Αν το δυαδικό ψηφίο του αφαιρέτη είναι μεγαλύτερο από το ψηφίο του αφαιρετέου, δανειζόμαστε 1 από την επόμενη σημαντική θέση. Για να μεταβιβαστεί το γεγονός αυτό στην επόμενη βαθμίδα χρησιμοποιείται ένα σήμα δανεισμού (borrow). Όπως ακριβώς υπάρχουν ημιαθροιστές και πλήρεις αθροιστές, υπάρχουν ημιαφαιρέτες και πλήρεις αφαιρέτες.

Ημιαφαιρέτης

Ο ημιαφαιρέτης είναι ένα συνδυαστικό κύκλωμα το οποίο αφαιρεί δυο δυαδικά ψηφία και δίνει τη διαφορά τους. Έχει επίσης μία έξοδο που καθορίζει αν χρειάζεται να δανειστούμε (borrow) μια μονάδα. Αν συμβολίσουμε το ψηφίο του αφαιρετέου με x και του αφαιρέτη με y, για να κάνουμε την αφαίρεση x-y πρέπει να ελέγξουμε τα σχετικά μεγέθη των x και y. Αν x>=y τότε υπάρχουν τρεις περιπτώσεις: 0-0=0, 1-0=1 1-1=0. Το αποτέλεσμα λέγεται δυαδικό ψηφίο διαφοράς. Αν x<y, τότε έχουμε 0-1, και έτσι χρειάζεται να δανειστούμε ένα 1 από την επόμενη βαθμίδα. Το 1 που δανειζόμαστε από την επόμενη θέση προσθέτει 2 στο δυαδικό ψηφίο του αφαιρέτη, όπως στο δεκαδικό σύστημα το κρατούμενο της αφαίρεσης προσθέτει 10 στον αφαιρετέο. Έτσι, με τον αφαιρετέο ίσο με 2, η διαφορά γίνεται 2-1=1. Ο ημιαφαιρέτης χρειάζεται δύο εξόδους. Η μια έξοδος παράγει τη διαφορά και θα συμβολίζεται με D (Difference, διαφορά). Η δεύτερη έξοδος που θα συμβολίζεται με B (Borrow, δανεικό) παράγει το δυαδικό σήμα που πληροφορεί την επόμενη βαθμίδα ότι δανειστήκαμε μια μονάδα. Ο πίνακας αληθείας του ημιαφαιρέτη είναι ο ακόλουθος:

***Πίνακας**: Πίνακας αληθείας ημιαφαιρέτη*

x	**y**	**B**	**D**
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

Οι συναρτήσεις Boole για τις εξόδους του ημιαφαιρέτη βρίσκονται από τον πίνακα αληθείας και είναι:

$$D=x'y + xy'$$

$$B=xy'$$

Πλήρης αφαιρέτης

Ο πλήρης αφαιρέτης είναι ένα συνδυαστικό κύκλωμα που εκτελεί την αφαίρεση μεταξύ δύο δυαδικών ψηφίων παίρνοντας υπόψη ότι μπορεί η λιγότερο σημαντική βαθμίδα να έχει δανειστεί μια μονάδα. Αυτό το κύκλωμα έχει τρεις εισόδους και δύο εξόδους. Οι τρεις είσοδοι x,y,z συμβολίζουν το ψηφίο του αφαιρετέου, του αφαιρέτη και του κρατουμένου, αντίστοιχα. Οι έξοδοι B, D συμβολίζουν τη διαφορά και το κρατούμενο εξόδου, αντίστοιχα. Ο πίνακας αληθείας του κυκλώματος είναι ο εξής:

Πίνακας: Πίνακας αληθείας πλήρους αφαιρέτη

x	y	z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Οι απλοποιημένες συναρτήσεις Boole των δύο εξόδων του πλήρους αφαιρέτη είναι οι ακόλουθες:

$$D=x'y'z + x'y'z' + xy'z' + xyz$$

$$B=x'y+x'z+yz$$

3.6.3 Πολλαπλασιαστής

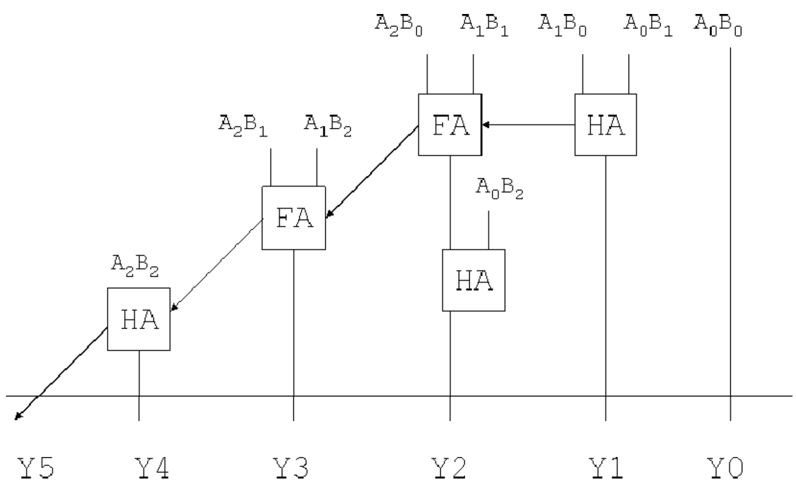
Η πράξη του πολλαπλασιασμού χρησιμοποιείται κατά κόρον στους σύγχρονους επεξεργαστές. Μπορούμε να υλοποιήσουμε τον πολλαπλασιασμό χρησιμοποιώντας διαδοχικές προσθέσεις και ολισθήσεις, στην πράξη όμως χρησιμοποιούμε ειδικές μονάδες που ονομάζονται πολλαπλασιαστές. Για να ξεκινήσουμε την εξήγηση του σχεδίου του πολλαπλασιαστή, θα ξεκινήσουμε από την υλοποίηση του πολλαπλασιασμού δύο 3-bit αριθμών, τους οποίους θα συμβολίσουμε A2 A1 A0 και B2 B1 B0. Στο ακόλουθο σχήμα παρουσιάζεται σχηματικά ο πολλαπλασιασμός των αριθμών.

			A2	A1	A0
		X	B2	B1	B0
			A2.B0	A1.B0	A0.B0
		A2.B1	A1.B1	A0.B1	
	A2.B2	A1.B2	A0.B2		
Y5	Y4	Y3	Y2	Y1	Y0

Το παραπάνω Σχήμα δείχνει ότι για να υπολογίσουμε το γινόμενο δύο αριθμών, αρκεί να υπολογίσουμε το άθροισμα κάποιων μερικών γινομένων, τα οποία υπολογίζονται με τη βοήθεια πυλών AND. Στη συνέχεια θα συμβολίζουμε με AiBj το σήμα που προκύπτει ως το λογικό AND των σημάτων Ai, Bj. Με βάση το συμβολισμό αυτό, το προηγούμενο Σχήμα γίνεται:

			A2	A1	A0
		X	B2	B1	B0
			A_2B_0	A_1B_0	A_0B_0
		A_2B_1	A_1B_1	A_0B_1	
	A_2B_2	A_1B_2	A_0B_2		
Y5	Y4	Y3	Y2	Y1	Y0

Έτσι οδηγούμαστε στην ακόλουθη δομή η οποία υλοποιεί τη ζητούμενη πράξη.



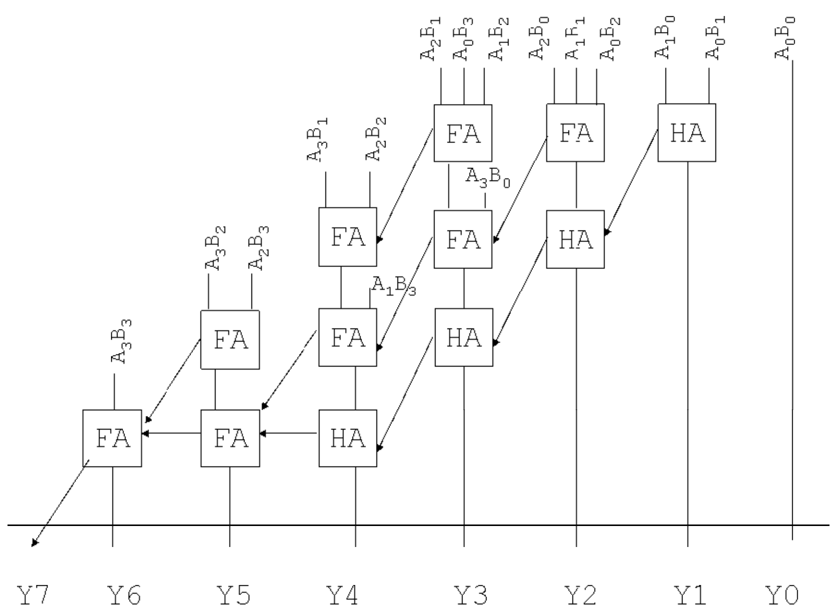
Οι μονάδες με την ένδειξη ‘FA’ είναι πλήρεις αθροιστές, ενώ οι μονάδες με την ένδειξη ‘HA’ είναι ημιαθροιστές. Μπορεί κανείς να παρατηρήσει ότι θα μπορούσαμε να έχουμε διαφορετική διάταξη αθροιστών-ημιαθροιστών. Στην πραγματικότητα, επειδή η πράξη του πολλαπλασιασμού είναι μία ιδιαίτερα χρονοβόρα πράξη σε σχέση π.χ. με την πρόσθεση, έχουν προταθεί διάφορες τεχνικές για την αύξηση της ταχύτητάς της.

Πολλαπλασιαστής με την τεχνική carry save

				A3	A2	A1	A0
			X	B3	B2	B1	B0
				A_3B_0	A_2B_0	A_1B_0	A_0B_0
			A_3B_1	A_2B_1	A_1B_1	A_0B_1	
		A_3B_2	A_2B_2	A_1B_2	A_0B_2		
	A_3B_3	A_2B_3	A_1B_3	A_0B_3			
Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

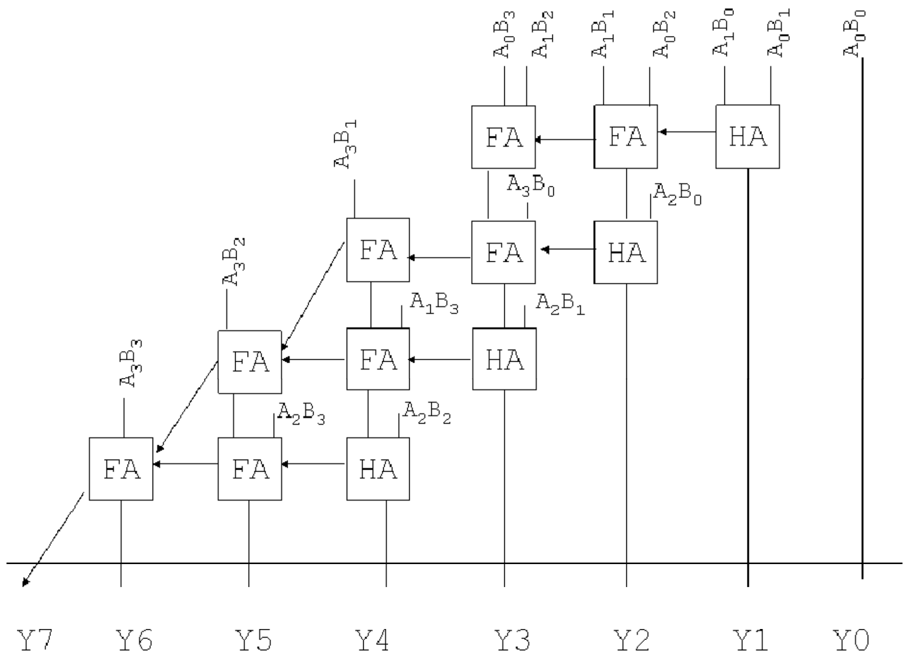
Στο ακόλουθο σχήμα δείχνουμε σχηματικά την πράξη του πολλαπλασιασμού για την περίπτωση των 4X4 bit αριθμών.

Στην ακόλουθη εικόνα παρουσιάζουμε τον πολλαπλασιαστή με μία τεχνική που ονομάζεται carry-save multiplication.



Πολλαπλασιαστής με την τεχνική carry propagate

Στην ακόλουθη εικόνα παρουσιάζουμε τον πολλαπλασιαστή με μία τεχνική που ονομάζεται carry-propagate (στην τεχνική αυτή τα carry από τους FA μεταφέρονται οριζόντια, ενώ στην προηγούμενη τεχνική μεταφέρονταν πλάγια και αριστερά).

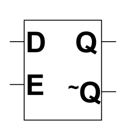


3.7 Ακολουθιακές μονάδες-Στοιχεία Μνήμης

Οι μονάδες στις οποίες αναφερθήκαμε μέχρι το σημείο αυτό ήταν συνδυαστικές, δηλαδή η τιμή της εξόδου κάποια χρονική στιγμή εξαρτάται από την τιμή της εισόδου εκείνη τη χρονική στιγμή. Στη συνέχεια θα αναφερθούμε σε ακολουθιακές μονάδες. Στις ακολουθιακές μονάδες η τιμή της εξόδου εξαρτάται από την τιμή των εισόδων όχι μόνο εκείνη τη χρονική στιγμή, αλλά και από τις τιμές τους προηγούμενες χρονικές στιγμές. Κύριο χαρακτηριστικό των ακολουθιακών μονάδων είναι το ότι χρησιμοποιούμε στοιχεία μνήμης. Υπάρχουν δύο είδη στοιχείων μνήμης: ο μανδαλωτής (latch) και το flip-flop.

Μανδαλωτής

Ένας μανδαλωτής είναι μια ψηφιακή συσκευή στην έξοδο της οποίας μπορούμε να αποθηκεύσουμε ένα ‘1’ ή ένα ‘0’. Ο μανδαλωτής έχει μια είσοδο δεδομένων (D, data) και μια είσοδο ενεργοποίησης (E, Enable). Έχει δύο εξόδους, τις οποίες συμβολίζουμε με Q και ~Q. Οι τιμές των εξόδων Q και ~Q είναι συμπληρωματικές, με άλλα λόγια όταν η έξοδος Q έχει την τιμή ‘1’, η έξοδος ~Q έχει την τιμή ‘0’ και αντίστροφα. Οι λόγοι που στο μανδαλωτή περιλαμβάνεται και η αντιστροφή της εξόδου είναι ότι η ύπαρξη της αντεστραμμένης εξόδου είναι πολύ χρήσιμη στην πράξη, καθώς και το ότι η υλοποίησή της είναι πολύ απλή. Το σχηματικό διάγραμμα και ο πίνακας αληθείας του μανδαλωτή φαίνονται στο επόμενο Σχήμα.



D E	Q
X 0	Q_n
0 1	0
1 1	1

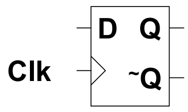
***Σχήμα**: Λογικό διάγραμμα και πίνακας αληθείας μανδαλωτή*

Ο μανδαλωτής λειτουργεί ως εξής: Αν η είσοδος ενεργοποίησης είναι ενεργοποιημένη (‘1’), τότε η τιμή της εισόδου δεδομένων (D) μεταφέρεται στην έξοδο (Q). Διαφορετικά, η τιμή της εξόδου παραμένει ίδια με την προηγούμενη τιμή της (την οποία συμβολίζουμε με Q_n). Με άλλα λόγια ο μανδαλωτής φυλάσσει (μανδαλώνει) μια τιμή στην έξοδό του όσο E=0.

Flip-flop

Το flip flop είναι μια συσκευή η οποία λειτουργεί παρόμοια με το μανδαλωτή μεταφέροντας την τιμή της εισόδου στην έξοδο. Η διαφορά του flip flop είναι πως η μεταφορά γίνεται τη στιγμή που το σήμα επίτρεψης (που στο flip-flop λέγεται ρολόι και συμβολίζεται με clk) αλλάζει τιμή από 0 σε 1. Με άλλα λόγια το flip flop λειτουργεί στην ακμή του ρολογιού.

Στο επόμενο Σχήμα φαίνεται το σχηματικό διάγραμμα του flip flop (που μοιάζει αρκετά με εκείνο του μανδαλωτή) και ο πίνακας λειτουργίας του.

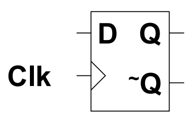


D	clk	Q
X	0	Q_n
X	1	Q_n
0	↑	0
1		1

***Σχήμα**: Flip flop τύπου D θετικής ακμής και πίνακας λειτουργίας*

Στον πίνακα λειτουργίας του flip flop φαίνεται ότι όταν η τιμή του σήματος clk έχει οποιαδήποτε τιμή (0 ή 1), η έξοδος του flip flop δεν επηρεάζεται (παραμένει στην προηγούμενη κατάσταση Q_n). Αντίθετα, όταν το σήμα clk μεταβαίνει από την κατάσταση ‘0’ στην ‘1’ (συμβολίζουμε αυτή τη μετάβαση με το βελάκι που φαίνεται στον πίνακα), η τιμή της εισόδου D μεταφέρεται στην έξοδο Q.

Ένας εναλλακτικός (και πιο απλός) τρόπος παρουσίασης της λειτουργίας του flip flop, είναι ο χαρακτηριστικός πίνακας. Ο πίνακας αυτός δίνει την τιμή της εξόδου ως συνάρτηση της μεταβλητής εισόδου, αφού συμφωνήσουμε ότι οι αλλαγές στην έξοδο γίνονται μόνο στη μετάβαση του παλμού χρονισμού από το 0 στο 1. Έτσι, ο χαρακτηριστικός πίνακας του flip flop τύπου D είναι ο ακόλουθος.



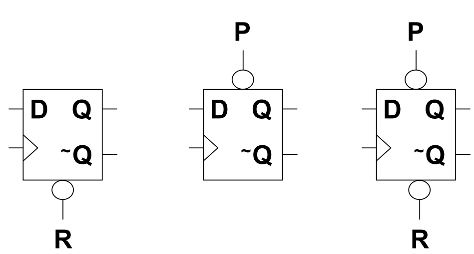
D	Q
0	0
1	1

***Σχήμα**: Flip flop τύπου D θετικής ακμής και χαρακτηριστικός πίνακας*

Ασύγχρονες είσοδοι

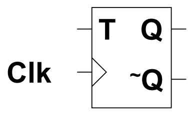
Ένα flip flop μπορεί να περιλαμβάνει ασύγχρονες εισόδους θέσης (P) και μηδένισης (R). Πιο συγκεκριμένα, μπορεί να περιλαμβάνει καμία από τις εισόδους αυτές (όπως τα flip flops που είδαμε μέχρι τώρα), οποιαδήποτε από τις δύο ή και τις δύο.

Η λειτουργία ασύγχρονων εισόδων είναι η εξής. Όταν η τιμή τους είναι 1, δεν επηρεάζεται η λειτουργία του flip flop. Όταν η είσοδος P είναι 0, τότε η τιμή της εξόδου Q γίνεται 1, ανεξάρτητα από την τιμή των εισόδων D και clk. Όταν η είσοδος R είναι 0, τότε η τιμή της εξόδου γίνεται 1, ανεξάρτητα από την τιμή της εισόδου χρονισμού (clk). Οι ασύγχρονες είσοδοι P και R δε μπορούν να είναι ταυτόχρονα 0, διότι, στην περίπτωση αυτή, η έξοδος του flip flop θα είναι απροσδιόριστη. Το επόμενο Σχήμα δείχνει τα σύμβολα του flip flop τύπου D με ασύγχρονη είσοδο P, σύγχρονη είσοδο R και με τις δύο ασύγχρονες εισόδους.



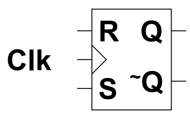
***Σχήμα**: flip flop τύπου D με (α) ασύγχρονη είσοδο μηδένισης (β) ασύγχρονη είσοδο θέσης και (γ) και τις δύο ασύγχρονες εισόδους*

Το flip flop στο οποίο αναφερθήκαμε ονομάζεται flip flop τύπου D. Το όνομα αυτό προέρχεται από την αγγλική λέξη data (δεδομένα). Η λειτουργία του είναι να μεταφέρει τα δεδομένα της εισόδου στην έξοδο. Εκτός από τον τύπο αυτό, υπάρχουν και άλλοι τύποι flip flop, όπως το flip flop reset-set (RS flip flop), Toggle (T flip flop), J-K flip flop. Η λειτουργία αυτών των flip flops φαίνεται στα επόμενα σχήματα.



T	Q
0	Q_n
1	$\sim Q_n$

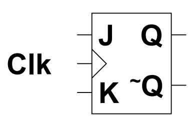
Σχήμα: *Flip flop τύπου T θετικής ακμής και χαρακτηριστικός πίνακας*



R	S	Q
0	0	Q_n
0	1	0
1	0	1
1	1	απρόβλεπτη

Σχήμα*: Flip flop τύπου RS θετικής ακμής και χαρακτηριστικός πίνακας*

Στο flip flop τύπου RS, οι είσοδοι R και S δε μπορούν να είναι ταυτόχρονα στο 1, γιατί στην περίπτωση αυτή η έξοδος του flip flop είναι απρόβλεπτη (αυτό οφείλεται στην κατασκευή του flip flop, με την οποία δε θα ασχοληθούμε εδώ). Λέμε ότι ο συνδυασμός εισόδων R=S=1 δεν είναι επιτρεπτός, και όταν σχεδιάζουμε ψηφιακά συστήματα προσπαθούμε να διασφαλίσουμε ότι ο συνδυασμός αυτός δεν εμφανίζεται στην είσοδο του flip flop.



J	K	Q
0	0	Q_n
0	1	0
1	0	1
1	1	$\sim Q_n$

Σχήμα*: Flip flop τύπου JK θετικής ακμής και χαρακτηριστικός πίνακας*

Τα flip flops των τύπων T, RS, JK μπορούν να επεκταθούν με ασύγχρονες εισόδους P και R, όπως και τα flip flops τύπου D. Η λειτουργία των ασύγχρονων εισόδων είναι ίδια με εκείνη που έχουν στα flip flops τύπου D.

Στη συνέχεια, θα περιγράψουμε τη λειτουργία απλών ακολουθιακών μονάδων χρησιμοποιώντας flip flops τύπου D. Οι ακολουθιακές μονάδες στις οποίες θα αναφερθούμε είναι οι καταχωρητές και οι μετρητές.

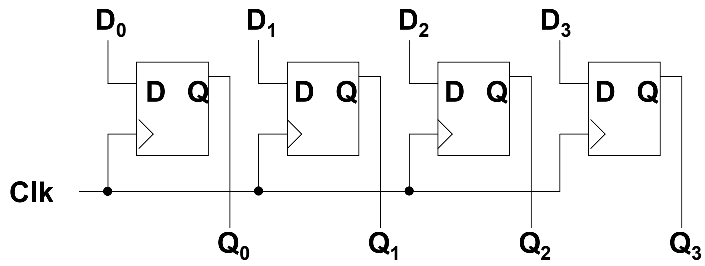
3.8 Καταχωρητές

Με τον όρο καταχώρηση εννοούμε τη λειτουργία με την οποία (δυαδικές) πληροφορίες φυλάσσονται για μετέπειτα επεξεργασία και χρήση. Στη σχεδίαση και υλοποίηση υπολογιστικών συστημάτων χρησιμοποιούνται κυρίως δύο είδη καταχωρητών. Οι παράλληλοι καταχωρητές και οι καταχωρητές ολίσθησης.

3.8.1 Παράλληλοι Καταχωρητές

Ένας παράλληλος καταχωρητής χρησιμοποιείται για τη φύλαξη δεδομένων και αποτελείται από στοιχεία μνήμης (flip flops). Τα στοιχεία μνήμης ενός παράλληλου καταχωρητή οδηγούνται απο μια κοινή είσοδο ρολογιού. Αυτή η είσοδος ρολογιού πυροδοτεί όλα τα flip flops, ώστε οι πληροφορίες που βρίσκονται εκείνη τη στιγμή στις εισόδους του καταχωρητή να μεταφερθούν στις εξόδους του.

Στο επόμενο Σχήμα φαίνεται το σχηματικό διάγραμμα ενός παράλληλου καταχωρητή 4 βαθμίδων (συνήθως αναφερόμαστε σε ένα τέτοιο καταχωρητή με τον όρο 4-μπιτο (4-bit), επειδή μπορεί να αποθηκεύσει πληροφορία 4 δυαδικών ψηφίων.

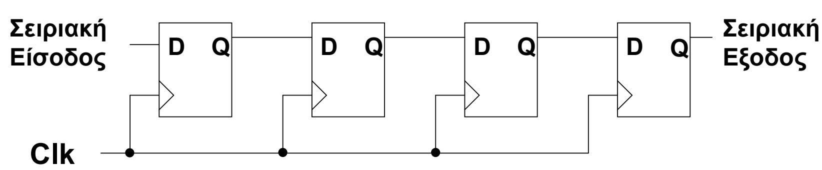


Σχήμα: *Παράλληλος καταχωρητής 4 βαθμίδων*

3.8.2 Καταχωρητές Ολίσθησης

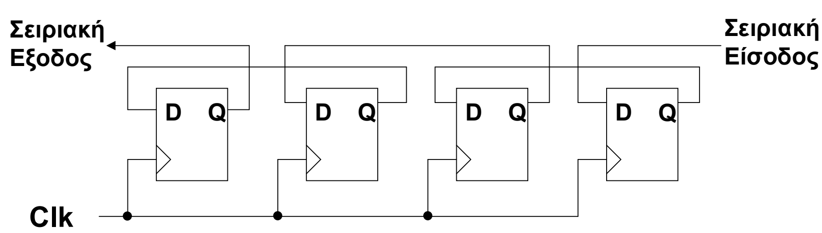
Ένας καταχωρητής που μπορεί να ‘ολισθαίνει’ τις πληροφορίες που περιέχει προς τη μια ή προς την άλλη κατεύθυνση ονομάζεται ‘καταχωρητής ολίσθησης’ (shift register). Ένας τέτοιος καταχωρητής αποτελείται από μια αλυσίδα από flip flops συνδεδεμένα στη σειρά, στα οποία η έξοδος του ενός τροφοδοτεί την είσοδο του γειτονικού του. Όλα τα flip flops παίρνουν ένα κοινό ρολόι, η ενεργοποίηση του οποίου προκαλεί την ολίσθηση από τη μια βαθμίδα στην επόμενη.

Σε ένα καταχωρητή δεξιάς ολίσθησης, σε κάθε παλμό του ρολογιού, το περιεχόμενο του καταχωρητή ολισθαίνει κατά μια θέση προς τα δεξιά. Η σειριακή είσοδος ρυθμίζει τι θα μπει στην είσοδο του πιο αριστερού flip flop σε κάθε ολίσθηση. Τη σειριακή έξοδο την παίρνουμε από την έξοδο του ακραίου δεξιού flip flop με την εφαρμογή του παλμού του ρολογιού.



Σχήμα: *Καταχωρητής δεξιάς ολίσθησης τεσσάρων βαθμίδων*

Αντίστοιχα, σε έναν καταχωρητή αριστερής ολίσθησης, σε κάθε παλμό του ρολογιού, το περιεχόμενο του καταχωρητή ολισθαίνει κατά μια θέση προς τα αριστερά.



Σχήμα: *Καταχωρητής αριστερής ολίσθησης τεσσάρων βαθμίδων*

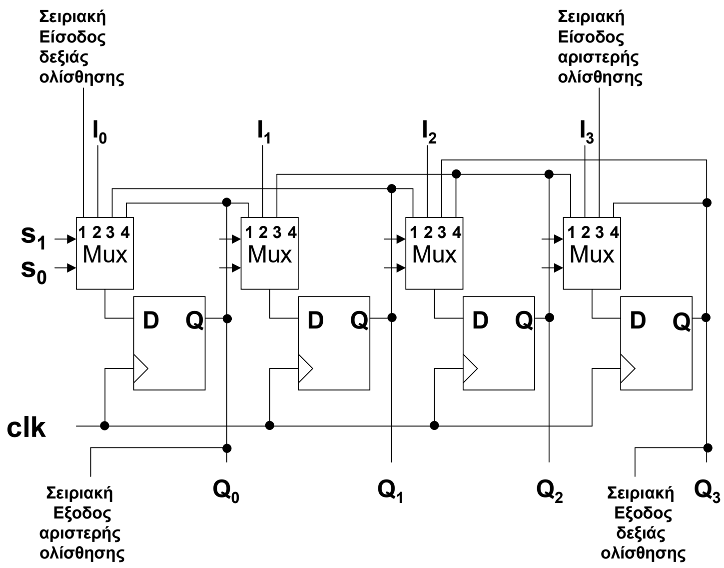
3.8.3 Καταχωρητές ολίσθησης με παράλληλη φόρτιση

Αν έχουμε στη διάθεσή μας τις εξόδους όλων των flip flops ενός καταχωρητή ολίσθησης ώστε να μπορούμε να κάνουμε τις κατάλληλες συνδέσεις, τότε τις πληροφορίες που εισάγουμε σειριακά στον καταχωρητή μπορούμε να τις πάρουμε παράλληλα από τις εξόδους των flip flops. Αν προσθέσουμε στον καταχωρητή ολίσθησης και τη δυνατότητα παράλληλης φόρτισης, τότε μπορούμε επιπλέον να παίρνουμε σειριακά από την έξοδο τα δεδομένα εκείνα τα οποία βάλαμε παράλληλα στην είσοδο, ολισθαίνοντάς τα μέσα από τον καταχωρητή. Έτσι, οι καταχωρητές ολίσθησης μπορούν να χρησιμοποιηθούν για τη μετατροπή σειριακών δεδομένων σε παράλληλα και αντίστροφα ως εξής. Ένας καταχωρητής ολίσθησης με παράλληλη φόρτιση έχει τις ακόλουθες δυνατότητες.

- Παράλληλη φόρτωση δεδομένων στον καταχωρητή
- δεξιά ολίσθηση
- αριστερή ολίσθηση

Ένας καταχωρητής ικανός για ολισθήσεις τόσο προς τα δεξιά όσο και προς τα αριστερά λέγεται αμφίδρομος καταχωρητής ολίσθησης. Αν ακόμη μπορεί να φορτωθεί παράλληλα, ονομάζεται καταχωρητής ολίσθησης με παράλληλη φόρτωση.

Ένας καταχωρητής που έχει τις δυνατότητες αυτές φαίνεται στο επόμενο Σχήμα. Αποτελείται από 4 flip flops, η είσοδος δεδομένων (D) κάθε ενός από τα οποία τροφοδοτείται από την έξοδο ενός πολυπλέκτη (multiplexer, MUX). Οι πολυπλέκτες έχουν δύο κοινές μεταβλητές επιλογής, s_1 και s_0, οι οποίες ρυθμίζουν τη λειτουργία του καταχωρητή.



Σχήμα: *Αμφίδρομος καταχωρητής ολίσθησης με παράλληλη φόρτωση 4 ψηφίων*

Ο επόμενος Πίνακας δείχνει τη λειτουργία του καταχωρητή για τις διαφορετικές τιμές των s1 και s0.

Πίνακας: Λειτουργία καταχωρητή ολίσθησης με παράλληλη φόρτιση

s1 s0	Είσοδος	Λειτουργία
0 0	1	δεξιά ολίσθηση
0 1	2	παράλληλη φόρτιση
1 0	3	αριστερή ολίσθηση
1 1	4	προηγούμενη τιμή

Η στήλη ‘Είσοδος΄ του πίνακα δείχνει ποια από τις τέσσερις εισόδους του πολυπλέκτη μεταφέρεται στην είσοδο D του flip flop. Η λειτουργία ‘προηγούμενη τιμή’ σημαίνει ότι οι έξοδοι του καταχωρητή διατηρούνται αμετάβλητες.

Χρήση καταχωρητών ολίσθησης

Γνωρίζουμε ότι μέσα στον υπολογιστή τα δεδομένα μεταφέρονται παράλληλα, μέσω του διαδρόμου. Κάποιες όμως από τις περιφερειακές μονάδες εισόδου ή εξόδου (πχ. πληκτρολόγιο, ποντίκι) επικοινωνούν με τον υπολογιστή σειριακά, στέλνοντας και λαμβάνοντας ένα-ένα τα δυαδικά ψηφία.

Οι καταχωρητές ολίσθησης μπορούν να χρησιμοποιηθούν για τη μετάδοση και λήψη των δεδομένων από τις μονάδες αυτές.

Για παράδειγμα, ας θεωρήσουμε ότι το υπολογιστικό σύστημα θέλει να αποστείλει δεδομένα (π.χ. 1 byte, 8 δυαδικά ψηφία) σειριακά. Για το σκοπό αυτό μπορεί να χρησιμοποιήσει ένα καταχωρητή ολίσθηση 8 βαθμίδων. Ο καταχωρητής αυτός φορτώνεται με παράλληλη φόρτιση (s_1s_0=01). Στη συνέχεια, λειτουργεί με δεξιά ολίσθηση για 8 κύκλους και μεταφέρει τα δεδομένα, bit-προς-bit, στη μονάδα μέσω της σειριακής εξόδου δεξιάς ολίσθησης.

Αντίστροφα, αν το υπολογιστικό σύστημα θέλει να λάβει δεδομένα, (πχ. 8 δυαδικά ψηφία) από τη μονάδα, ο καταχωρητής λειτουργεί με αριστερή ολίσθηση για 8 κύκλους λαμβάνοντας ένα-ένα τα δυαδικά ψηφία από τη σειριακή είσοδο αριστερής ολίσθησης. Μόλις φορτωθεί ένα ολόκληρο byte στον καταχωρητή, στέλνεται (παράλληλα) στη μονάδα του υπολογιστικού συστήματος για την οποία προορίζεται (ΚΜΕ, μνήμη κ.λπ.) μέσω του διαδρόμου του συστήματος.

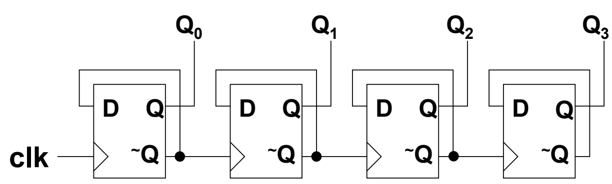
3.9 Μετρητές

Ένας δυαδικός μετρητής ή απαριθμητής (counter) είναι μια μονάδα που αποτελείται από στοιχεία μνήμης (flip flops) και έχει μια είσοδο χρονισμού (ρολόϊ). Σε κάθε μεταβολή της εισόδου χρονισμού από 0 σε 1, οι έξοδοι του μετρητή μεταβάλλονται με τέτοιο τρόπο ώστε η δυαδική τιμή τους να είναι μεγαλύτερη κατά 1 από την δυαδική τιμή που είχαν στην προηγούμενη κατάσταση. Ένας μετρητής χαρακτηρίζεται από το πλήθος των βαθμίδων (στοιχείων μνήμης) από τα οποία αποτελείται. Οι έξοδοι ενός μετρητή που αποτελείται από n βαθμίδες, μεταβάλλονται με τέτοιο τρόπο ώστε η αριθμητική τους τιμή να αυξάνεται από 0 έως 2^n-1. Για παράδειγμα, ένας μετρητής τριών βαθμίδων μπορεί να παίρνει τις τιμές 000, 001, 010, 011, 100, 101, 110, 111, 000, …

Οι μετρητές διακρίνονται, ανάλογα με τη σχεδίαση, σε ασύγχρονους μετρητές (μετρητές ριπής) και σε σύγχρονους μετρητές. Στη συνέχεια θα αναφερθούμε στη σχεδίαση των δύο αυτών ειδών μετρητών.

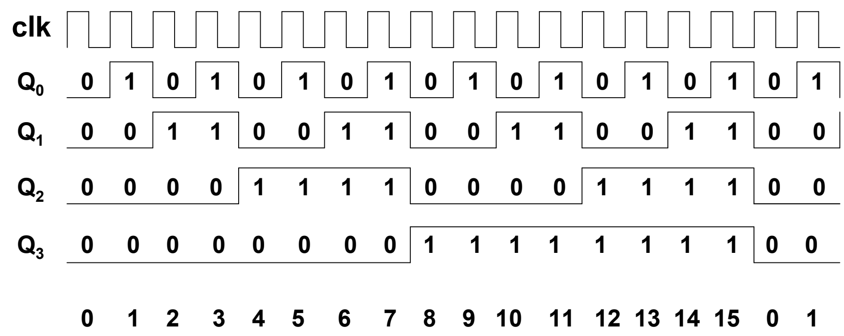
Ασύγχρονοι μετρητές

Ένας ασύγχρονος μετρητής αποτελείται από flip flops συνδεδεμένα έτσι ώστε η έξοδος κάθε flip flop να συνδέεται στην είσοδο χρονισμού του flip flop της επόμενης βαθμίδας. Το flip flop της πρώτης βαθμίδας δέχεται τους εισερχόμενους παλμούς μέτρησης. Ένας ασύγχρονος απαριθμητής 4 βαθμίδων αποτελούμενος από flip flops τύπου D φαίνεται στο επόμενο Σχήμα.



Σχήμα: *Ασύγχρονος απαριθμητής 4 βαθμίδων*

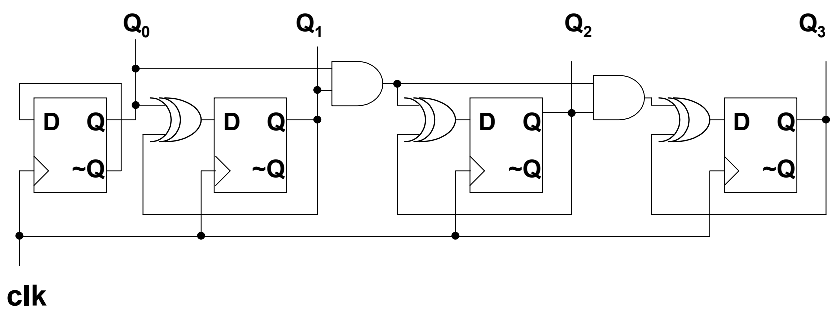
Ο απαριθμητής αυτός λειτουργεί ως εξής. Αρχικά οι έξοδοι όλων των βαθμίδων είναι 0. Όταν το σήμα χρονισμού γίνει 1 από 0, η τιμή του flip flop Q_0 θα αλλάξει από 0 σε 1. Σε κάθε μεταβολή του σήματος χρονισμού από '0' σε '1', η τιμή του flip flop Q_0 θα μεταβάλλεται. Επιπλέον, κάθε φορά που η τιμή του flip flop Q_0 αλλάζει από 0 σε 1, η τιμή του flip flop Q_1 θα μεταβάλλεται. Γενικά, όταν η τιμή μιας βαθμίδας μεταβάλλεται από '0' σε '1', η τιμή της επόμενης βαθμίδας θα αλλάζει. Στο επόμενο Σχήμα φαίνεται η ακολουθία που παράγεται από το δυαδικό απαριθμητή του προηγούμενου σχήματος ξεκινώντας από την τιμή 0000.



Σχήμα*: Ακολουθία που παράγεται από τον απαριθμητή 4-βαθμίδων*

Σύγχρονοι Μετρητές

Οι σύγχρονοι μετρητές διαφέρουν από τους μετρητές ριπής στο ότι οι παλμοί του ρολογιού εφαρμόζονται στις εισόδους χρονισμού όλων των flip flops. Ο κοινός παλμός πυροδοτεί όλα τα flip flops συγχρόνως, και όχι το ένα μετά το άλλο, όπως στους μετρητές ριπής. Ένας σύγχρονος δυαδικός μετρητής σχεδιασμένος με flip flops τύπου D φαίνεται στο επόμενο Σχήμα.

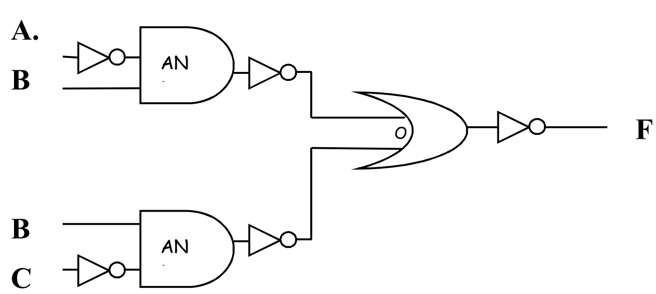


Σχήμα*: Σύγχρονος δυαδικός απαριθμητής τεσσάρων βαθμίδων*

Η είσοδος δεδομένων κάθε flip flop οδηγείται από μια πύλη XOR της οποίας η μια είσοδος είναι η έξοδος του flip flop. Η απόφαση σχετικά με το αν η έξοδος ενός flip flop πρέπει να αντιστραφεί ή όχι στηρίζεται στην τιμή της εξόδου της πύλης AND που τροφοδοτεί την άλλη έξοδο της πύλης XOR. Η έξοδος του πρώτου flip flop αντιστρέφεται σε κάθε κύκλο ρολογιού. Η έξοδος τους δεύτερου flip flop αντιστρέφεται όταν η έξοδος του πρώτου flip flop είναι 1. Η έξοδος των επόμενων βαθμίδων αντιστρέφεται όταν οι έξοδοι όλων των προηγουμένων βαθμίδων είναι ‘1’. Έτσι, το Q_2 αντιστρέφεται όταν Q_0=Q_1=1, ενώ η Q_3 αντιστρέφεται όταν Q_0=Q_1=Q_2=1.

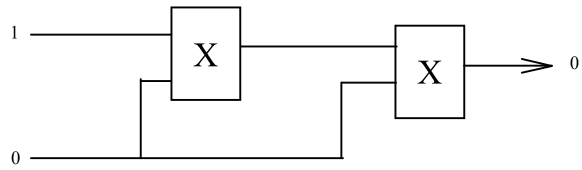
3.10 Ασκήσεις

Άσκηση 1. Δίνεται το παρακάτω κύκλωμα. Να βρεθεί ο πίνακας αλήθειας.

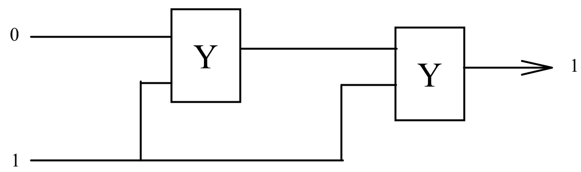


Άσκηση 2. Σε κάθε ένα από τα παρακάτω κυκλώματα τα τετράγωνα αναπαριστούν τον ίδιο τύπο λογικής πύλης. Λαμβάνοντας υπόψη τις τιμές των εισόδων και εξόδων όπως φαίνονται παρακάτω καθορίστε τον εκάστοτε τύπο πύλης για κάθε σχήμα χωριστά. Δικαιολογήστε την απάντησή σας.

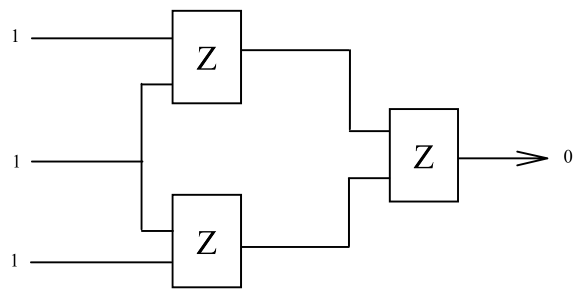
α.



β.



γ.



Άσκηση 3. Έστω η ακόλουθη λογική συνάρτηση f.

f = ((NOT A) AND B AND (NOT C)) OR ((NOT A) AND B AND C) OR (A AND B AND (NOT C))

Μια λογική συνάρτηση μπορεί να απλοποιηθεί, δηλαδή να παραχθεί μια απλούστερη λογική συνάρτηση που θα έχει τον ίδιο πίνακα αλήθειας και η οποία θα υλοποιείται με ένα απλούστερο δίκτυο πυλών. Η λογική συνάρτηση g είναι η απλουστευμένη μορφή της λογικής συνάρτησης f.

g = B AND ((NOT A) OR (NOT C))

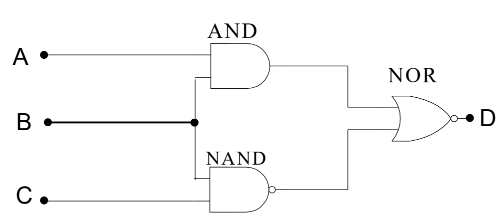
α) Χρησιμοποιώντας πύλες AND, OR και NOT σχεδιάστε το κύκλωμα που θα υλοποιήσει την λογική συνάρτηση f.

β) βρείτε τον πίνακα αλήθειας της λογικής συνάρτησης g.

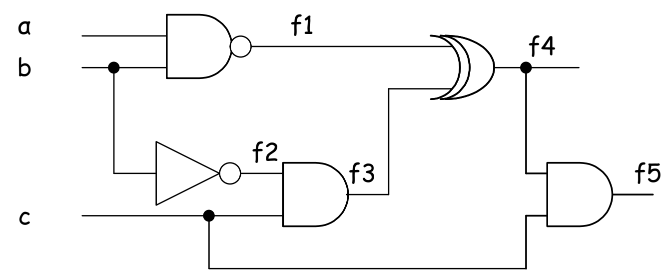
γ) βρείτε τον πίνακα αλήθειας της ακόλουθης λογικής συνάρτησης h. Τι παρατηρείτε;

h = ((*NOT* A) *AND* B) *OR* (B *AND* (*NOT* C))

Άσκηση 4. Δίνεται το ακόλουθο λογικό κύκλωμα. Κατασκευάστε αλγόριθμο που να διαβάζει τις τιμές εισόδου του κυκλώματος A, B και C και εμφανίζει την τιμή εξόδου D (το πρόγραμμα να ελέγχει αν οι τιμές των A, B και C είναι 0 ή 1 και σε περίπτωση που δεν είναι να επαναλαμβάνει την ανάγνωση).

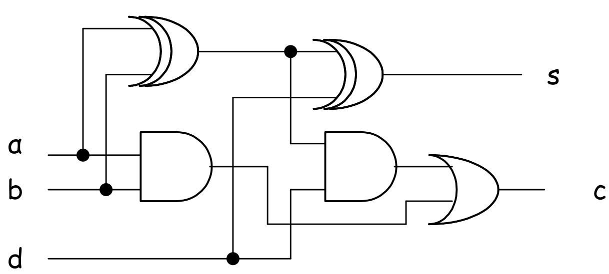


Άσκηση 5. Βρείτε τον πίνακα αληθείας για όλες τις δυνατές τιμές των μεταβλητών εισόδου a, b, c του κυκλώματος στο ακόλουθο σχήμα.

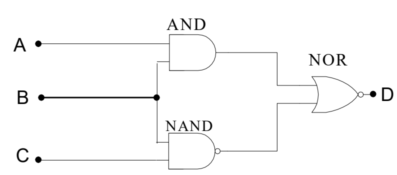


Άσκηση 6. Δίνεται το κύκλωμα του πλήρους αθροιστή. Ζητείται (α) να συμπληρώσετε τον πίνακα για τις τιμές των S και C και (β) να δώσετε τις συναρτήσεις των εξόδων S, C.

a	b	d	s	c
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



Άσκηση 7. Δίνεται το ακόλουθο λογικό κύκλωμα: Να δημιουργηθεί ο πίνακας αλήθειας της εξόδου D για όλους τους δυνατούς συνδυασμούς των εισόδων A, B, C.

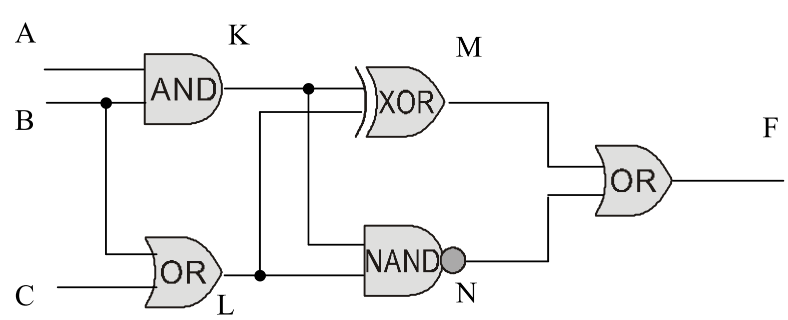


Άσκηση 8. Σχεδιάστε το λογικό κύκλωμα που υλοποιεί τη λογική συνάρτηση:

F = (A XOR B) OR ((NOT(B)) AND (A NAND C))

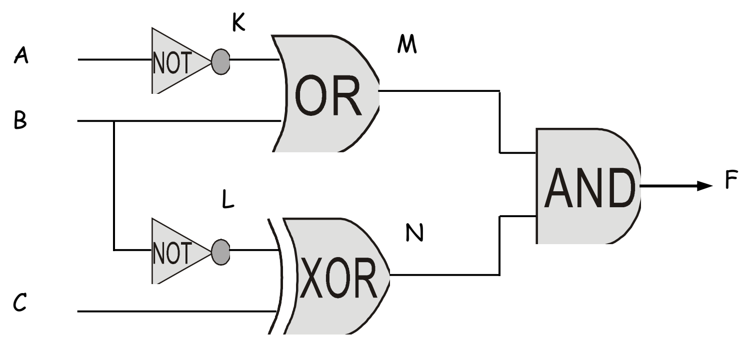
Στη συνέχεια δώστε τον πίνακα αληθείας της παραπάνω λογικής συνάρτησης F.

Άσκηση 9. Η λογική συνάρτηση F υλοποιείται από το παρακάτω λογικό κύκλωμα:



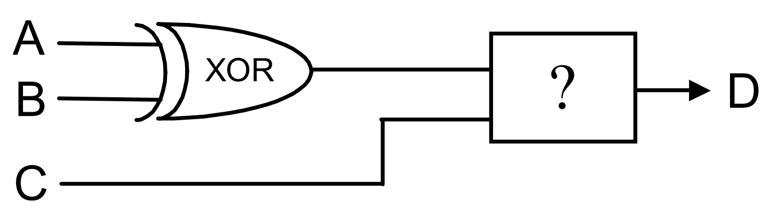
α) Γράψτε τον αναλυτικό τύπο της F, ως συνάρτηση μόνο των τριών εισόδων του κυκλώματος A, B και C.

β) Σχεδιάστε τον πίνακα αληθείας της F, συμπεριλαμβάνοντας τις εισόδους A, B, C, την έξοδο F και τα ενδιάμεσα αποτελέσματα K, L, M και N.



Άσκηση 10. Δίνεται το ακόλουθο κύκλωμα. Σχηματίστε τον πίνακα αληθείας της συνάρτησης F για όλες τις πιθανές τιμές των εισόδων A, B και C, συμπεριλαμβάνοντας και τα ενδιάμεσα αποτελέσματα (K, L, M και N).

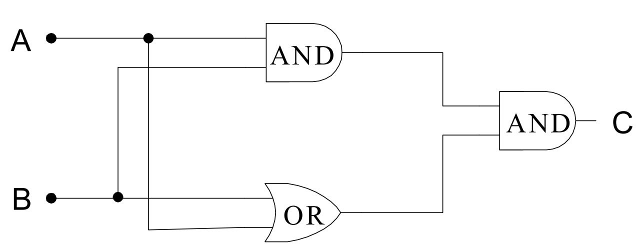
Άσκηση 11. Δίνεται το ακόλουθο λογικό κύκλωμα:



Αν ο αντίστοιχος πίνακας αληθείας είναι αυτός που ακολουθεί, να προσδιοριστεί η πύλη που σημειώνεται με ? και να γραφεί ο τύπος της λογικής της συνάρτησης (D = …).

A	B	C	D
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Άσκηση 12. Δίνεται το ακόλουθο λογικό κύκλωμα



α. Δώστε τις τιμές στον πίνακα αλήθειας για κάθε πιθανή τιμή εισόδου των A, B. Δηλαδή συμπληρώστε τις τιμές του C σε ένα αντίστοιχο πίνακα:

A	B	C

0	0	
0	1	
1	0	
1	1	

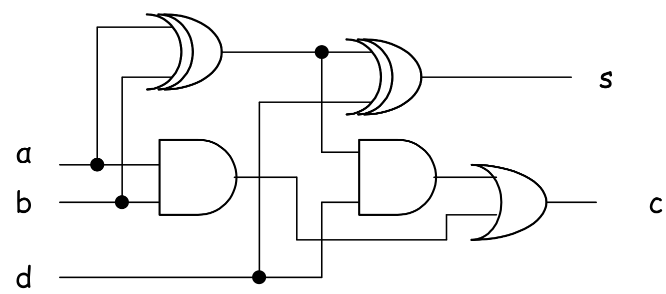
β. Με ποια (μία) λογική πύλη είναι ισοδύναμος ο πίνακας αλήθειας που προκύπτει;

Άσκηση 13. Δίνεται η λογική συνάρτηση f = B AND ((NOT A) OR (NOT C)). Χρησιμοποιώντας τα σχήματα των πυλών AND, OR και NOT σχεδιάστε το κύκλωμα που θα υλοποιήσει την Λογική συνάρτηση f και στη συνέχεια βρείτε τον πίνακα αληθείας της.

Άσκηση 14. Δίνεται η λογική συνάρτηση g = ((NOT A) AND B) OR (B AND (NOT C)). Βρείτε τον πίνακα αλήθειας της g.

Άσκηση 15. Δίνεται το κύκλωμα του πλήρους αθροιστή. Ζητείται (α) να συμπληρώσετε τον πίνακα για τις τιμές των S και C και (β) να δώσετε τις συναρτήσεις των εξόδων S, C.

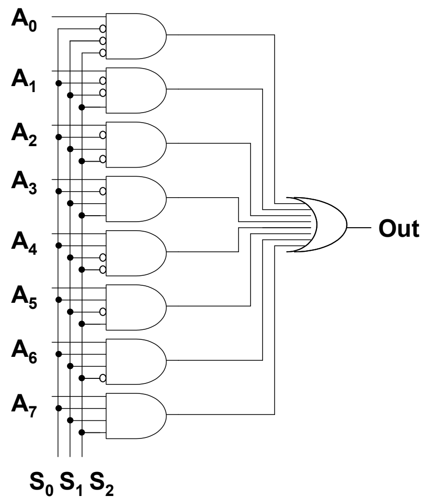
a	b	d	s	c
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



Άσκηση 16. Δίνεται η λογική συνάρτηση f f= B AND ((NOT A) OR (NOT C)). Χρησιμοποιώντας πύλες AND, OR και NOT σχεδιάστε το κύκλωμα που θα υλοποιήσει τη λογική συνάρτηση f και στη συνέχεια βρείτε τον πίνακα αλήθειας της.

Άσκηση 17. Δίνεται η λογική συνάρτηση f = ((A AND B) XOR (B OR C)) NAND (NOT(A)). (α) σχεδιάστε το λογικό κύκλωμα που την υλοποιεί. (β) σχηματίστε τον πίνακα αληθείας της F, συμπεριλαμβάνοντας και όσα ενδιάμεσα αποτελέσματα δημιουργούνται.

Άσκηση 18. Δίνεται η λογική συνάρτηση F = ((A OR B) XOR (B AND C)) AND (NOT(A)). (α) σχεδιάστε το λογικό κύκλωμα που υλοποιεί την F. (β) σχηματίστε τον πίνακα αληθείας της F, συμπεριλαμβάνοντας και όσα ενδιάμεσα αποτελέσματα δημιουργούνται.

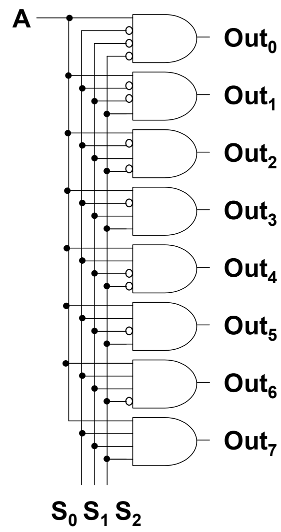


Άσκηση19. Δίνεται το ακόλουθο κύκλωμα με εισόδους A0-A7 και S0-S2, και έξοδο Out. Το κυκλάκι στην είσοδο μιας πύλης, σημαίνει ότι η αντίστοιχη είσοδος αντιστρέφεται πριν την είσοδό της στην πύλη.

Χαρακτηρίστε ως σωστή ή λάθος κάθε μια από τις ακόλουθες προτάσεις:

1. Όταν οι είσοδοι S0-S2 είναι όλες 0, η έξοδος Out είναι 0.
2. Για να έχει η έξοδος Out την τιμή 0, αρκεί η είσοδος A3 να έχει την τιμή 1 και οι είσοδοι S0, S1, S2 να έχουν τις τιμές 0, 1, 0 αντίστοιχα.
3. Αν οι τιμές όλων των εισόδων A0-A7 είναι 1, τότε η έξοδος Out έχει την τιμή 1, ανεξάρτητα από τις τιμές των εισόδων S0, S1, S2.
4. Υπάρχει τουλάχιστο ένας συνδυασμός εισόδων που δίνει την τιμή 1 στην έξοδο Out.
5. Υπάρχουν τουλάχιστο 8 συνδυασμοί εισόδων που δίνουν την τιμή 1 στην έξοδο Out.
6. Το πλήθος των συνδυασμών εισόδων του κυκλώματος είναι 512.
7. Για να έχει τιμή ‘1’ η έξοδος της πύλης με είσοδο την A3, είναι αρκετό η είσοδος A3 να έχει την τιμή ‘1’.
8. Για να έχει η έξοδος Out την τιμή 1, είναι απαραίτητο η τιμή των εξόδων όλων των πυλών AND να είναι ‘1’.
9. Η συνάρτηση εξόδου της πύλης με είσοδο A3, είναι ‘A3 . S0’ . S1 . S2’ όπου το ‘.’ συμβολίζει την λογική πράξη της σύζευξης.
10. Όταν οι είσοδοι S0, S1, S2 έχουν τιμές 0, 1, 0 αντίστοιχα, η τιμή της εξόδου δεν εξαρτάται από την τιμή της εισόδου.

Άσκηση 20. Δίνεται το ακόλουθο κύκλωμα. Να συμπληρώσετε τον πίνακα για τις τιμές των εισόδων.



A	S0	S1	S2	Out0	Out1	Out2	Out3	Out4	Out5	Out6	Out7
0	0	0	0								
0	0	0	1								
0	0	1	0								
0	0	1	1								
0	1	0	0								
0	1	0	1								
0	1	1	0								
0	1	1	1								
1	0	0	0								
1	0	0	1								
1	0	1	0								
1	0	1	1								
1	1	0	0								
1	1	0	1								
1	1	1	0								
1	1	1	1								

Άσκηση 21. Σχεδιάστε πολλαπλασιαστή 6Χ6 με την τεχνική carry save

Άσκηση 22. Σχεδιάστε πολλαπλασιαστή 6Χ6 με την τεχνική carry propagate