Ελαχιστοποίηση σε επίπεδο πυλών

Βαρτζιώτης Φώτιος

Ορισμός

- ***Ελαχιστοποίηση σε επίπεδο πυλών (Gate-Level Minimization) αποκαλείται η διαδικασία εύρεσης της βέλτιστης υλοποίησης, με χρήση λογικών πυλών, των συναρτήσεων Boole που περιγράφουν ψηφιακά κυκλώματα.***

Η μέθοδος του Χάρτη

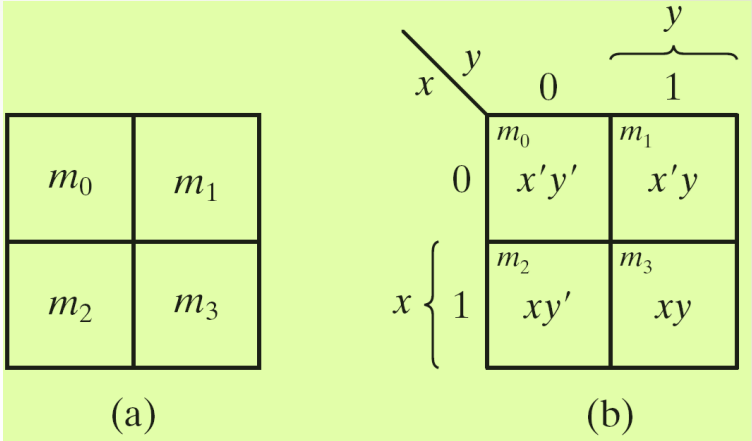
- Η πολυπλοκότητα της σχεδίασης ενός κυκλώματος με ψηφιακές λογικές πύλες εξαρτάται
  - Από την πολυπλοκότητα της αλγεβρικής έκφρασης που περιγράφει το κύκλωμα
- Ελαχιστοποίηση
  - Αλγεβρική προσέγγιση: Χωρίς καθορισμένους κανόνες
  - Ο χάρτης Karnaugh
    - Απλή και αξιόπιστη διαδικασία
    - Αποτελεί γραφική μορφή του πίνακα αληθείας
    - Μπορεί να εφαρμοστεί όταν ο αριθμός των μεταβλητών είναι < 7
- Είναι ένα διάγραμμα που αποτελείται από τετράγωνα
  - Κάθε τετράγωνο αναπαριστά έναν ελαχιστόρο (minterm)

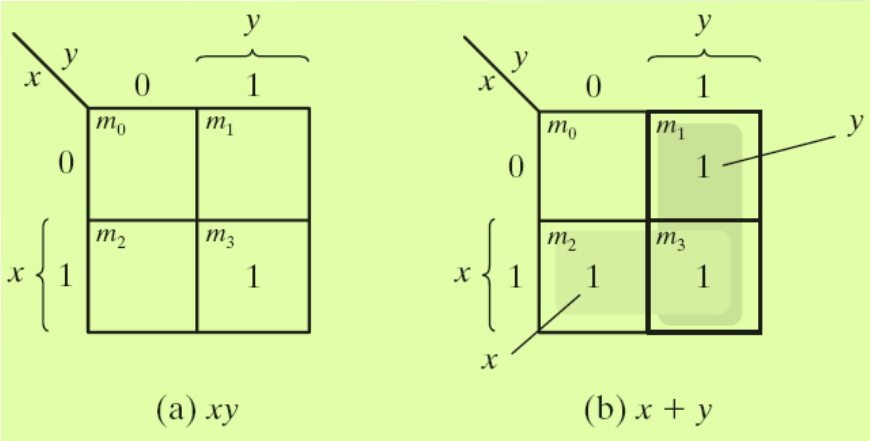
Boolean Συναρτήσεις

- Συνάρτηση Boole
  - Άθροισμα ελαχιστόρων (Sum of minterms)
  - Άθροισμα γινομένων (ή γινόμενο αθροισμάτων) στην πιο απλή μορφή
  - Ελάχιστος αριθμός όρων
  - Ελάχιστος αριθμός παραγόντων
  - Η απλοποιημένη έκφραση μπορεί να μην είναι μοναδική.

Χάρτης Δύο Μεταβλητών

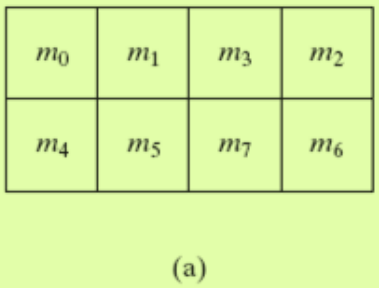
- Τέσσερις ελαχιστόροι
- x' = γραμμή 0; x = γραμμή 1
- y' = στήλη 0; y = στήλη 1
- Ένας πίνακας αληθείας σε μορφή διαγράμματος τετραγώνων
- (a): $xy = m_3$
- (b): $x+y = x'y+xy' +xy = m_1+m_2+m_3$



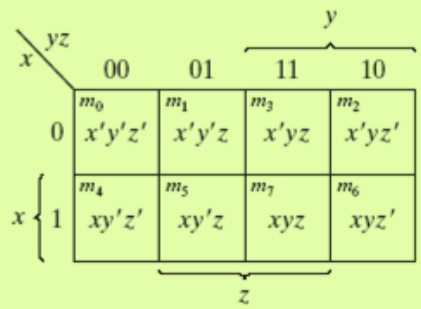


Χάρτης Τριών Μεταβλητών

- Οκτώ ελαχιστόροι
- Ακολουθία κώδικα Gray
- Κάθε δύο γειτονικά τετράγωνα του χάρτη διαφέρουν μόνο κατά μια μεταβλητή:
  - Με τόνο στο ένα τετράγωνο και χωρίς τόνο στο άλλο
  - π.χ., οι m_5 και m_7 μπορούν να απλοποιηθούν σε έναν όρο
  - $m_5 + m_7 = xy'z + xyz = xz\,(y'+y) = xz$



(a)



(b)

Χάρτης Τριών Μεταβλητών

- Οι m_0 και m_2 (m_4 και m_6) θεωρούνται γειτονικοί
- $m_0 + m_2 = x'y'z' + x'yz' = x'z'(y'+y) = x'z'$
- $m_4 + m_6 = xy'z' + xyz' = xz'(y'+y) = xz'$

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

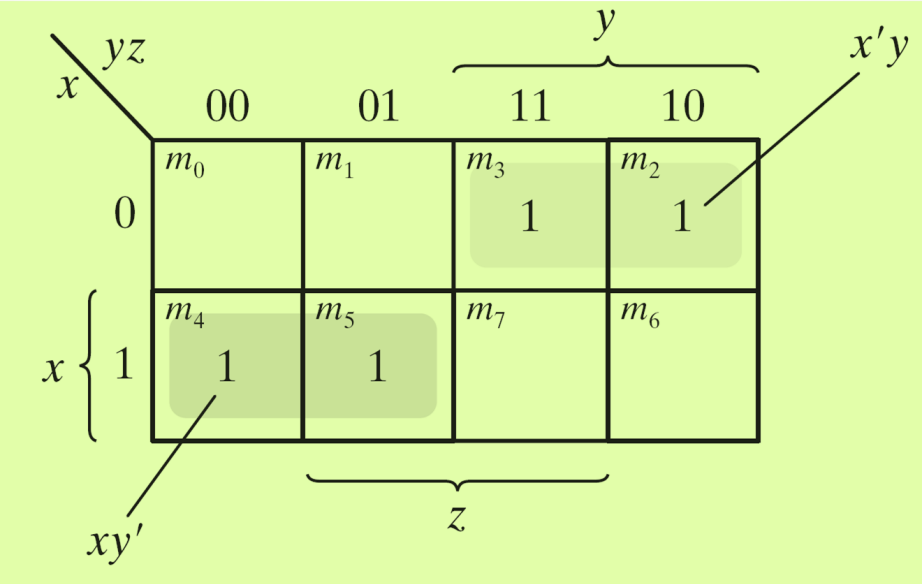
(a)

$x \backslash yz$	00	01	11	10
0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
x 1	$xy'z'$	$xy'z$	xyz	xyz'

(b)

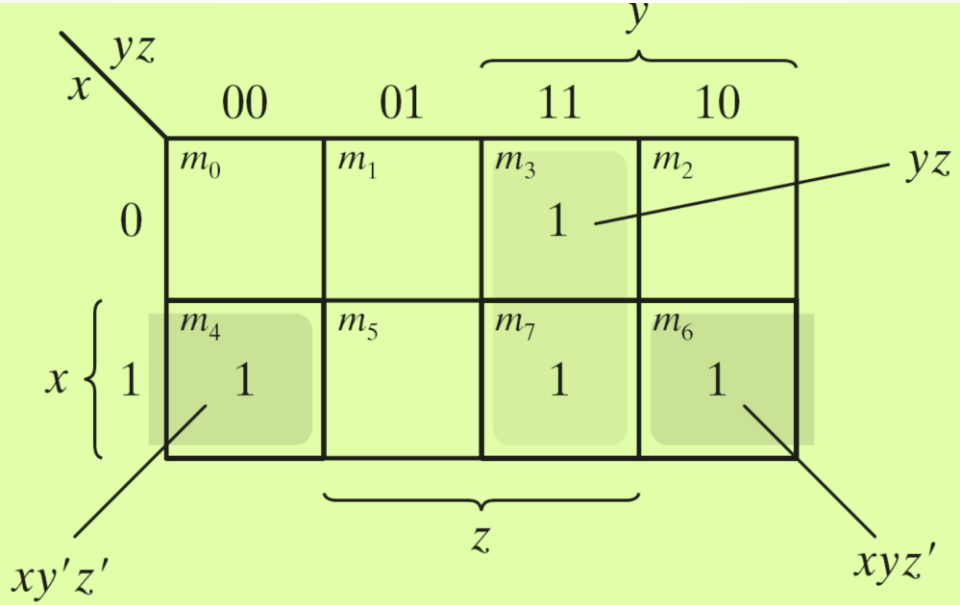
Χάρτης Τριών Μεταβλητών

- Παράδειγμα (3.1): Απλοποιήστε την Boolean συνάρτηση F(x, y, z) = Σ(2, 3, 4, 5)
  - *F*(x, *y*, z) = Σ(2, 3, 4, 5) = x'*y* + x*y*'



Χάρτης Τριών Μεταβλητών

- Παράδειγμα (3.2): Απλοποιήστε την $F(x, y, z) = \Sigma(3, 4, 6, 7)$
  - $F(x, y, z) = \Sigma(3, 4, 6, 7) = yz + xz'$



Δείτε: $xy'z' + xyz' = xz'$

Χάρτης Τριών Μεταβλητών

- Θεωρείστε τέσσερα γειτονικά τετράγωνα
  - 2, 4, και 8 τετράγωνα
  - $m_0+m_2+m_4+m_6$ = x'y'z'+x'yz'+xy'z'+xyz' = x'z'(y'+y) +xz'(y'+y) = x'z' + xz' = z'
  - $m_1+m_3+m_5+m_7$ = x'y'z+x'yz+xy'z+xyz =x'z(y'+y) + xz(y'+y) =x'z + xz = z

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

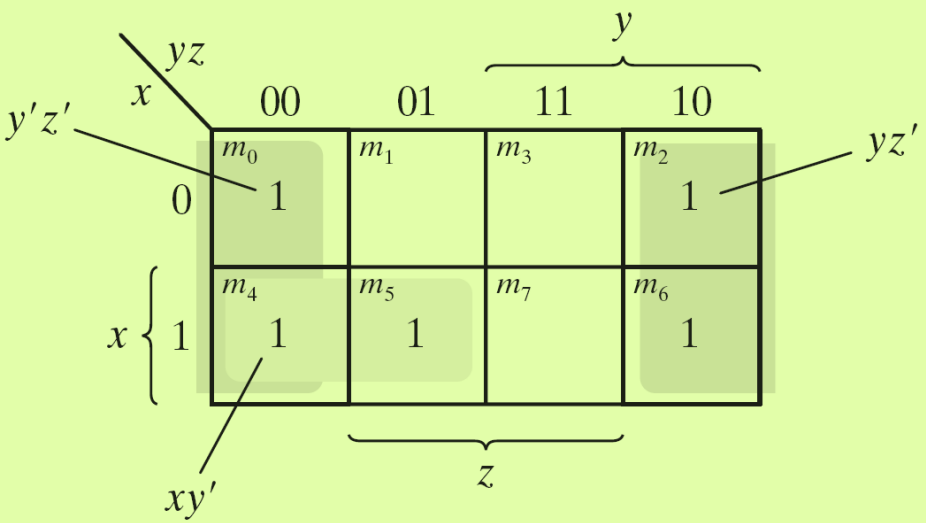
(a)

x \ xz	00	01	11	10
0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
1	$xy'z'$	$xy'z$	xyz	xyz'

(b)

Χάρτης Τριών Μεταβλητών

- Παράδειγμα (3.3): Απλοποιήστε την F(x, y, z) = Σ(0, 2, 4, 5, 6)
- *F*(x, *y*, z) = Σ(0, 2, 4, 5, 6) = z'+ xy'



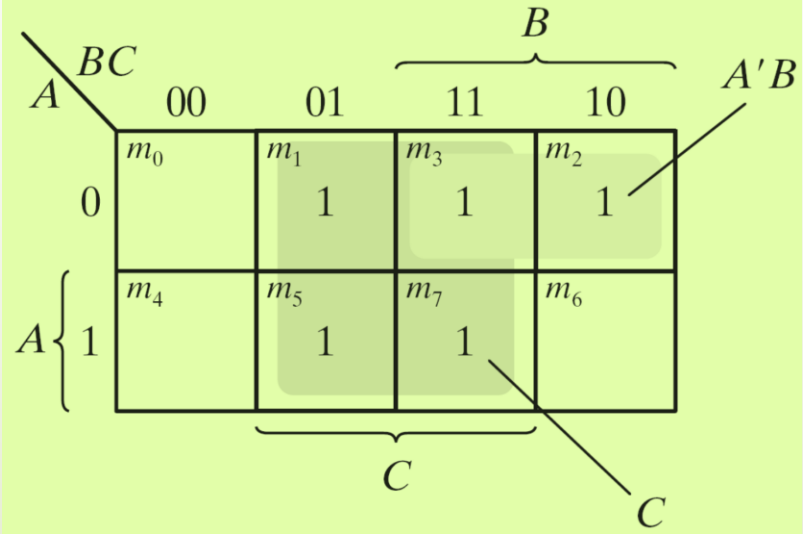
Δείτε: $y'z' + yz' = z'$

Χάρτης Τριών Μεταβλητών

- Παράδειγμα (3.4): Έστω $F = A'C + A'B + AB'C + BC$
  - a) Εκφράστε τη συνάρτηση ως άθροισμα ελαχιστόρων.
  - b) Βρείτε το ελάχιστο άθροισμα γινομένων της F.

  Απάντηση:

  $F(A, B, C) = \Sigma(1, 2, 3, 5, 7) = C + A'B$



Χάρτης Τεσσάρων Μεταβλητών

- Χάρτης με
  - 16 ελαχιστόρους
  - Συνδυασμοί των 2, 4, 8, και 16 γειτονικών τετραγώνων

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

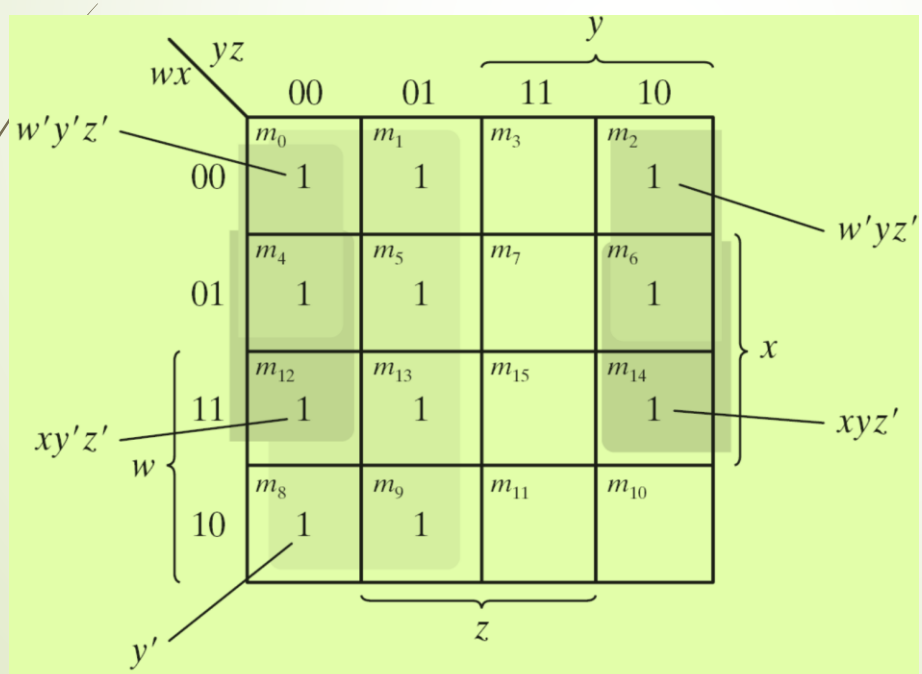
(a)

$wx \backslash yz$	00	01	11	10
00	$w'x'y'z'$	$w'x'y'z$	$w'x'yz$	$w'x'yz'$
01	$w'xy'z'$	$w'xy'z$	$w'xyz$	$w'xyz'$
11	$wxy'z'$	$wxy'z$	$wxyz$	$wxyz'$
10	$wx'y'z'$	$wx'y'z$	$wx'yz$	$wx'yz'$

(b)

Χάρτης Τεσσάρων Μεταβλητών

- Παράδειγμα (3.5): Απλοποιήστε την $F(w, x, y, z) = \Sigma(0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)$



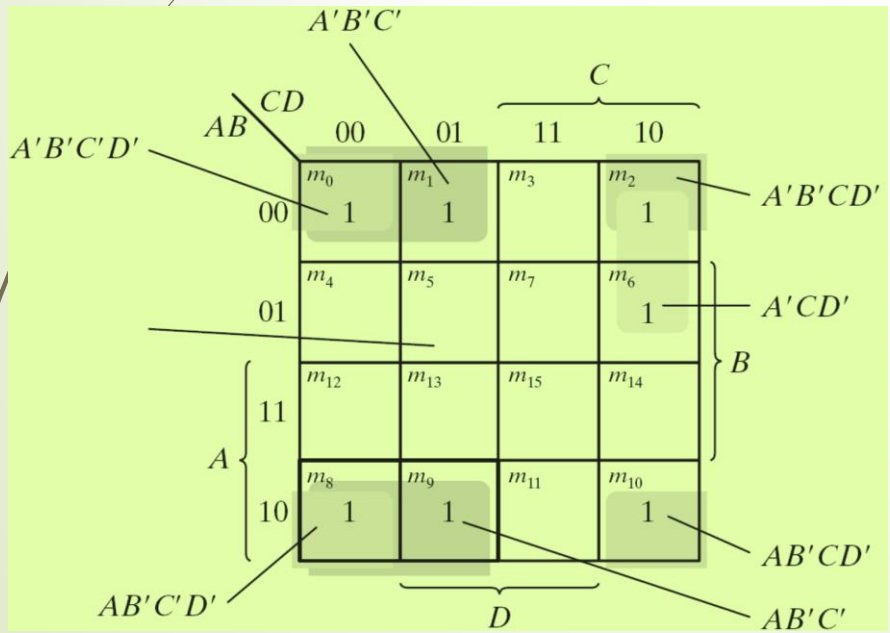
Δείτε: $w'y'z' + w'yz' = w'z'$
$xy'z' + xyz' = xz'$

$F = y' + w'z' + xz'$

Χάρτης Τεσσάρων Μεταβλητών

- Παράδειγμα (3.6): Απλοποιήστε την

$$F = A'B'C' + B'CD' + A'BC'D' + AB'C'$$



$$A'B'C'D' + A'B'CD' = A'B'D'$$
$$AB'C'D' + AB'CD' = AB'D'$$
$$A'B'D' + AB'D' = B'D'$$
$$A'B'C' + AB'C' = B'C'$$

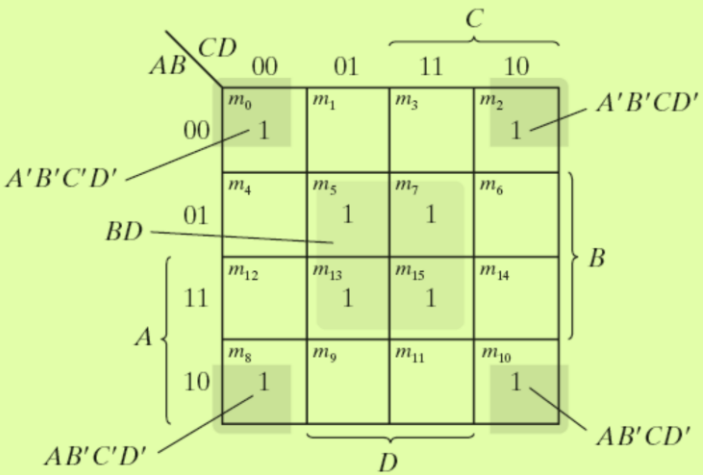
Πρωτεύοντες Όροι

- Πρωτεύοντες Όροι
  - Όλοι οι ελαχιστόροι καλύπτονται
  - Ελάχιστος αριθμός όρων
  - Πρωτεύον όρος (prime implicant): ο απλοποιημένος όρος γινομένου που προκύπτει εάν συνδυάσουμε το μέγιστο πιθανό αριθμό γειτονικών τετραγώνων του χάρτη
  - Θεμελιώδης Π.Ο.: Υπάρχει ένας ελαχιστόρος που καλύπτεται μόνο από αυτόν τον Π.Ο.
  - Ένας θεμελιώδης Π.Ο. περιλαμβάνεται πάντα στην περιγραφή της συνάρτησης

Πρωτεύοντες Όροι

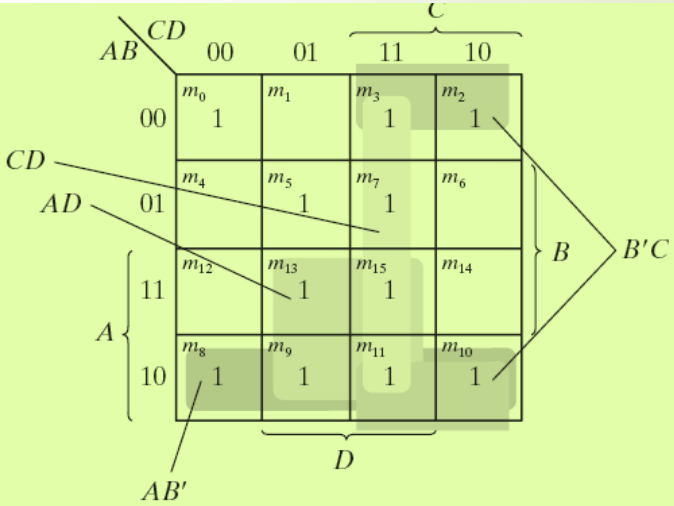
- Έστω η $F(A, B, C, D) = \Sigma(0, 2, 3, 5, 7, 8, 9, 10, 11, 13, 15)$
  - Η απλοποιημένη έκφρασή της μπορεί να μην είναι μοναδική
  - $F = BD+B'D'+CD+AD = BD+B'D'+CD+AB'$

    $= BD+B'D'+B'C+AD = BD+B'D'+B'C+AB'$



Δείτε: $A'B'C'D' + A'B'CD' = A'B'D'$
$AB'C'D' + AB'CD' = AB'D'$
$A'B'D' + AB'D' = B'D'$

Μερικός χάρτης με Θεμελιώδεις Π.Ο.



Συνολικός χάρτης με τους Π.Ο. *CD, B'C, AD και AB'*

Χάρτης Πέντε Μεταβλητών

- Οι χάρτες για περισσότερες από τέσσερις μεταβλητές είναι περίπλοκοι:
  - Χάρτης πέντε μεταβλητών: Δύο χάρτες των τεσσάρων μεταβλητών, σε δύο επίπεδα

$A = 0$

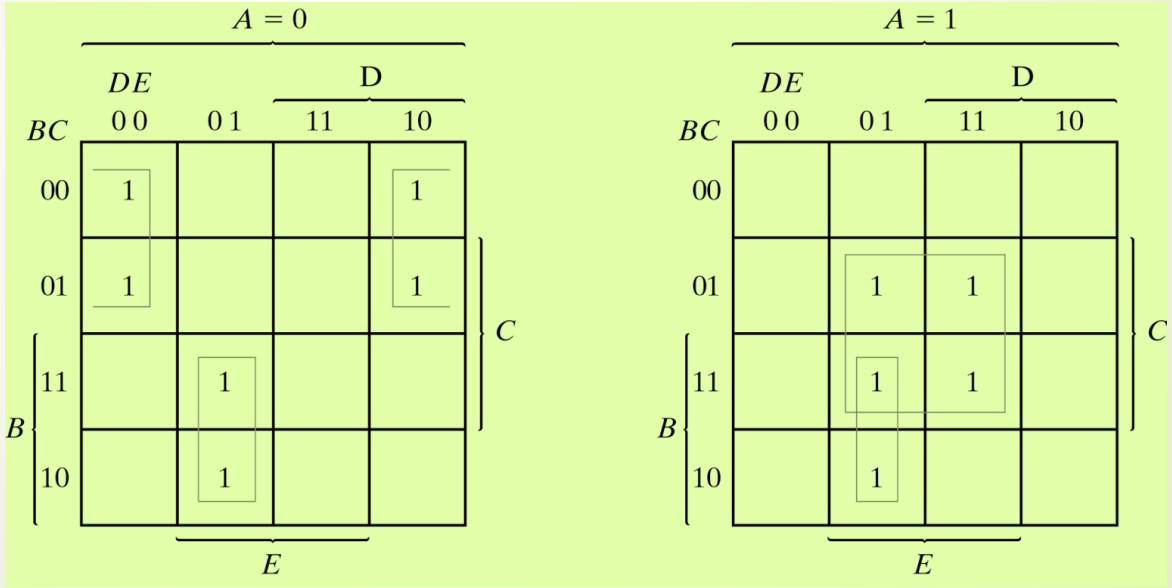
BC \ DE	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$A = 1$

BC \ DE	00	01	11	10
00	16	17	19	18
01	20	21	23	22
11	28	29	31	30
10	24	25	27	26

Χάρτης Πέντε Μεταβλητών

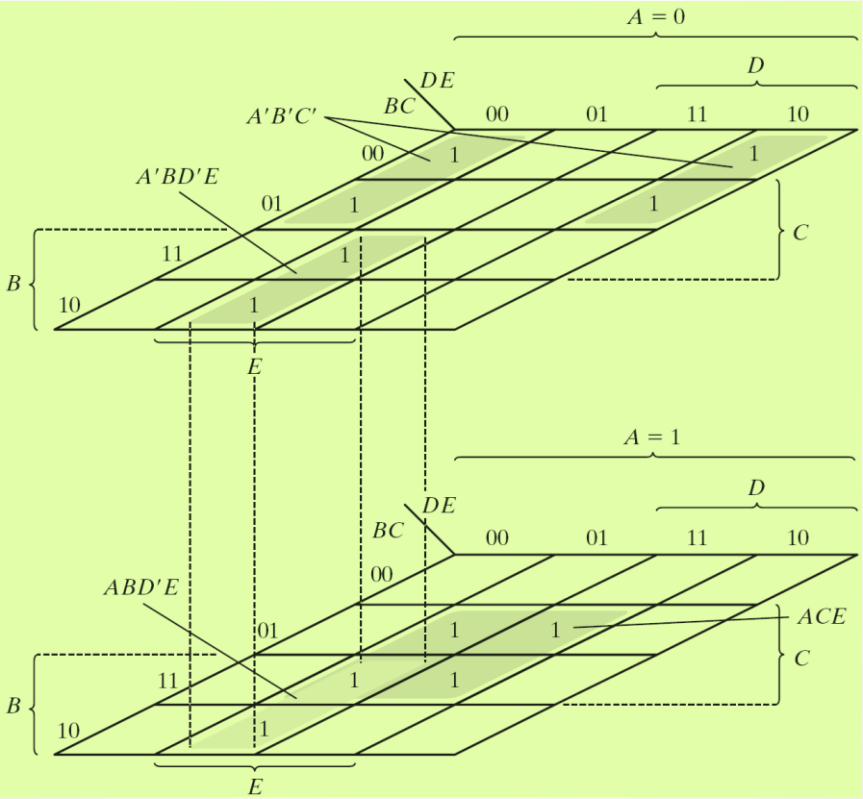
- Παράδειγμα: Απλοποιήστε την $F = \Sigma(0, 2, 4, 6, 9, 13, 21, 23, 25, 29, 31)$



$$F = A'B'E' + BD'E + ACE$$

Χάρτης Πέντε Μεταβλητών

- Διαφορετική όψη του χάρτη πέντε μεταβλητών



$$F = A'B'E' + BD'E + ACE$$

Απλοποίηση σε Μορφή Γινόμενου Αθροισμάτων

- Προσέγγιση #1
  - Βρείτε την απλοποιημένη F' σε μορφή αθροίσματος γινομένων
  - Εφαρμόστε το θεώρημα DeMorgan $F = (F')'$
  - F': άθροισμα γινομένων → F: γινόμενο αθροισμάτων
- Προσέγγιση #2: Δυισμός
  - Μετατροπή ελαχιστόρων σε μεγιστόρους
  - $M_0M_1 = (A+B+C+D)(A+B+C+D') = (A+B+C)+(DD') = A+B+C$

AB \ CD	00	01	11	10
00	M_0	M_1	M_3	M_2
01	M_4	M_5	M_7	M_6
11	M_{12}	M_{13}	M_{15}	M_{14}
10	M_8	M_9	M_{11}	M_{10}

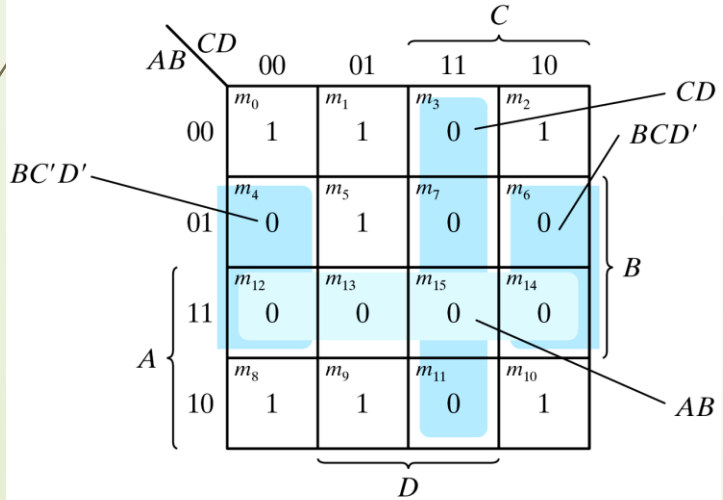
Θυμηθείτε: $m_j' = M_j$

Απλοποίηση σε Μορφή Γινόμενου Αθροισμάτων

Παράδειγμα (3.7): Απλοποιήστε την

$F(A, B, C, D) = \Sigma(0, 1, 2, 5, 8, 9, 10)$

σε (α) άθροισμα γινομένων, και (β) σε γινόμενο αθροισμάτων:



Note: $BC'D' + BCD' = BD'$

a) $F(A, B, C, D) = \Sigma(0, 1, 2, 5, 8, 9, 10) = B'D' + B'C' + A'C'D$

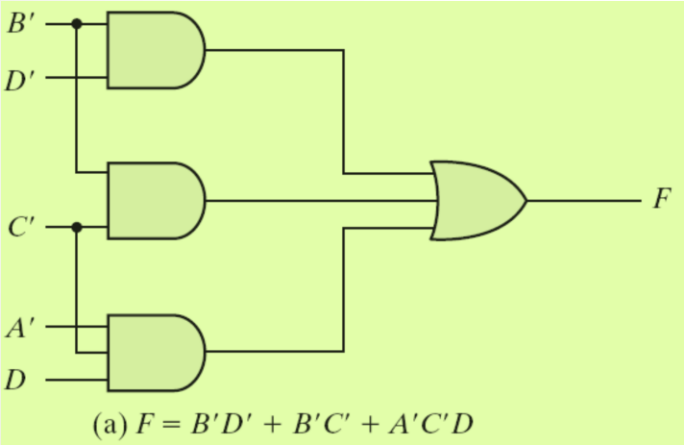
b) $F' = AB + CD + BD'$

- Θεώρημα DeMorgan $F = (A'+B')(C'+D')(B'+D)$
- ή με χρήση απευθείας μεγιστόρων

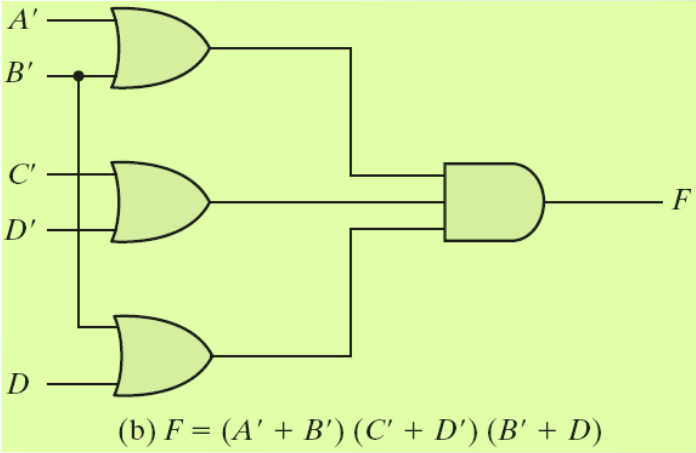
$F(A, B, C, D) = \Sigma(0, 1, 2, 5, 8, 9, 10) = B'D' + B'C' + A'C'D$

Απλοποίηση σε μορφή γινόμενου αθροισμάτων

- Συνέχεια παραδείγματος: Υλοποίηση της συνάρτησης με λογικές πύλες



Άθροισμα γινομένων



Γινόμενο αθροισμάτων

Απλοποίηση σε Μορφή Γινόμενου Αθροισμάτων

- Έστω η συνάρτηση του πίνακα:
  - Σε άθροισμα ελαχιστόρων:

$$F(x,y,z) = \sum(1,3,4,6)$$

  - Σε γινόμενο μεγιστόρων:

$$F(x,y,z) = \prod(0,2,5,7)$$

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

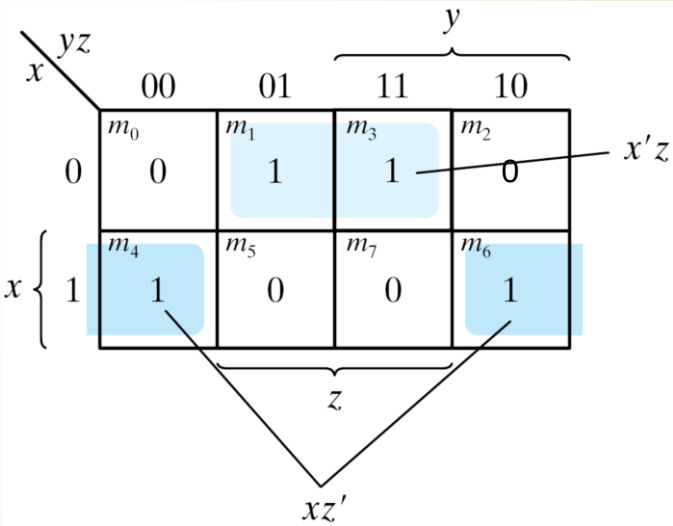
Απλοποίηση σε Μορφή Γινόμενου Αθροισμάτων

- Ο χάρτης της συνάρτησης
  - Χρησιμοποιήστε τα 1:

$$F(x,y,z) = x'z + xz'$$

  - Χρησιμοποιήστε τα 0:

$$F'(x,y,z) = xz + x'z'$$



- Το συμπλήρωμα της F′

$$F(x,y,z) = (x' + z')(x + z)$$

Συνθήκες Αδιάφορης Τιμής(don't care)

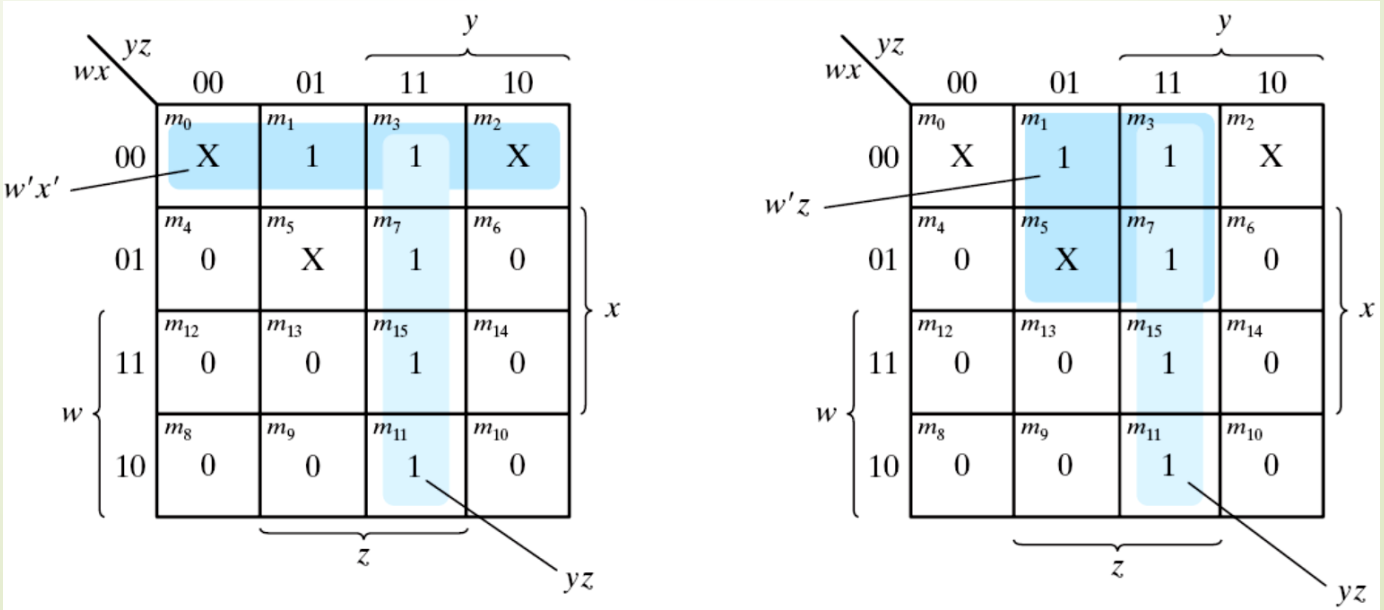
- Υπάρχουν περιπτώσεις όπου η τιμή μιας συνάρτησης δεν δίνεται για κάποιους συνδυασμούς των μεταβλητών, πχ
  - Κώδικας BCD; Οι συνδυασμοί 1010 έως 1111: Αδιάφοροι (don't care)
- Οι αδιάφοροι όροι μπορούν να αξιοποιηθούν στην ελαχιστοποίηση μιας συνάρτησης
  - Μπορεί να θεωρηθούν 0 ή 1, ανάλογα με το όφελος

Συνθήκες Αδιάφορης Τιμής(don't care)

- Παράδειγμα (3.8): Απλοποιήστε την $F(w, x, y, z) = \Sigma(1, 3, 7, 11, 15)$, η οποία έχει ως αδιάφορους όρους τους $d(w, x, y, z) = \Sigma(0, 2, 5)$.

- $F = yz + w'x'$ ή $F = yz + w'z$ *από τις:*
- $F = \Sigma(0, 1, 2, 3, 7, 11, 15)$ και $F = \Sigma(1, 3, 5, 7, 11, 15)$ αντίστοιχα που απεικονίζονται και στον πίνακα

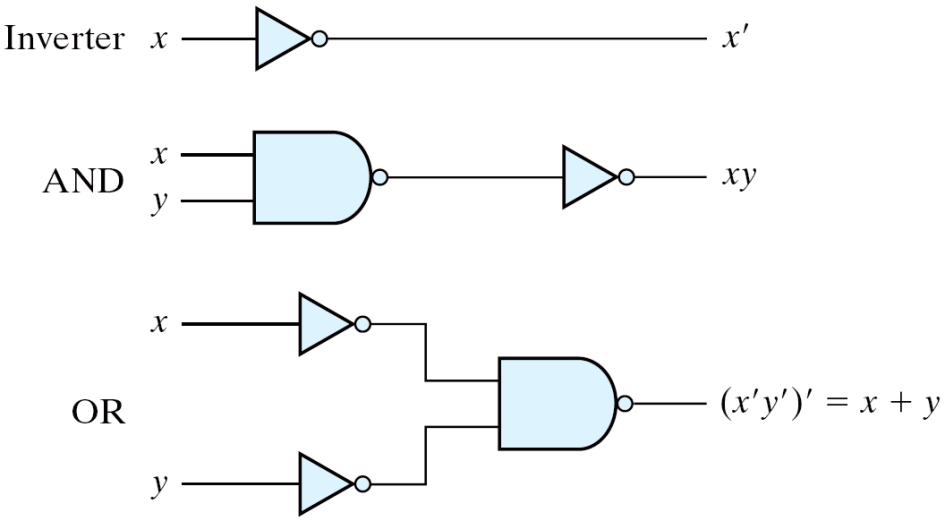
- Και οι δύο εκφράσεις είναι αποδεκτές



(a) $F = yz + w'x'$ (b) $F = yz + w'z$

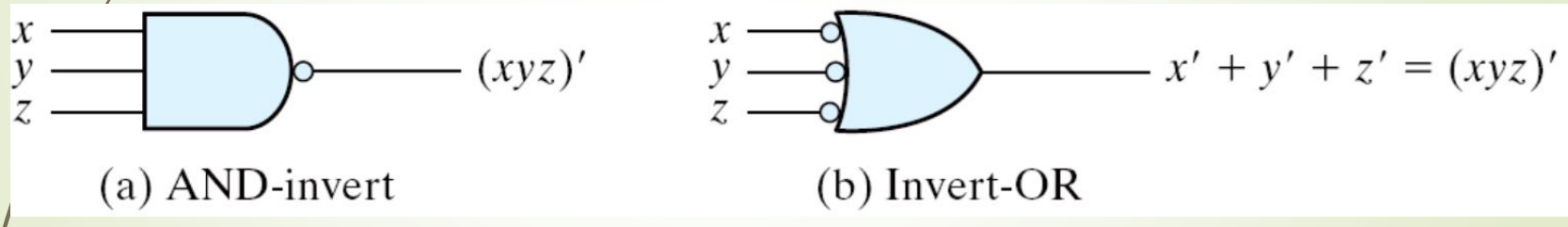
Υλοποίηση με πύλες NAND και NOR

- Οι πύλες NAND χαρακτηρίζονται και ως Οικουμενικές
  - Μπορούν να υλοποιήσουν οποιοδήποτε ψηφιακό σύστημα!



Πύλη NAND

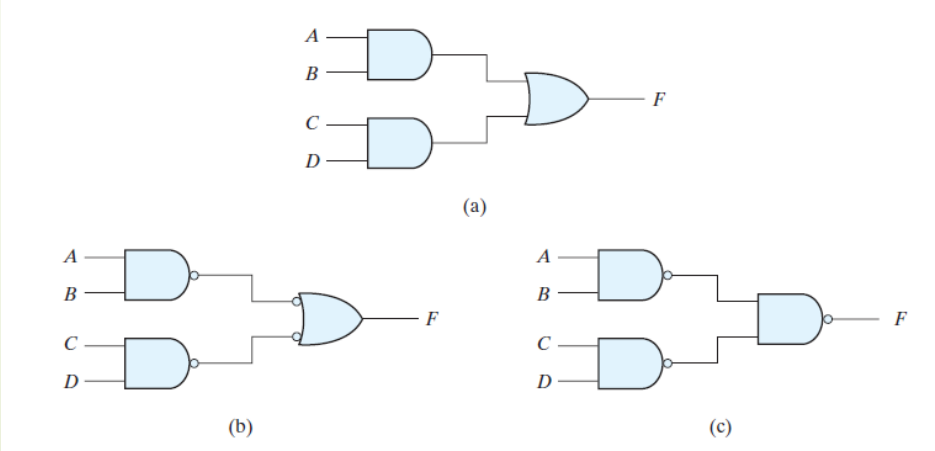
- Χρησιμοποιούνται δύο γραφικά σύμβολα για την πύλη NAND



(a) AND-invert (b) Invert-OR

Υλοποίηση Δύο Επιπέδων NAND

- Λογική δύο επιπέδων
  - NAND-NAND = Άθροισμα γινομένων
  - Example: $F = AB+CD$
  - $F = ((AB)'\ (CD)'\ )' = AB+CD$

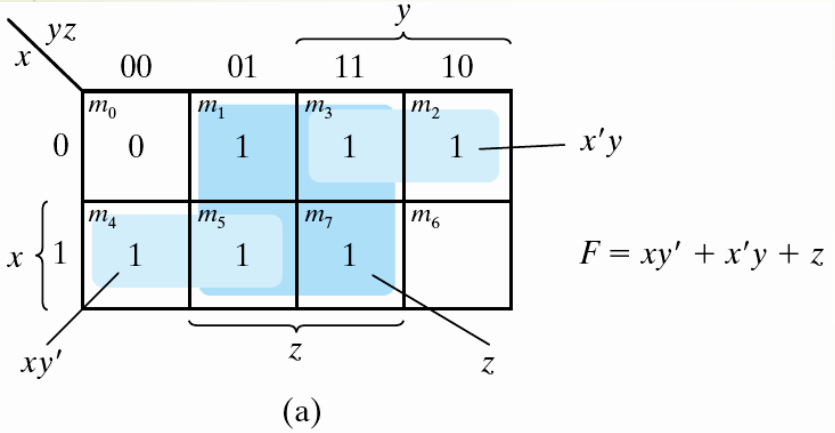


Τρεις υλοποιήσεις της $F = AB + CD$

Υλοποίηση Δύο Επιπέδων NAND

- Παράδειγμα 3.9: Υλοποιήστε την συνάρτηση

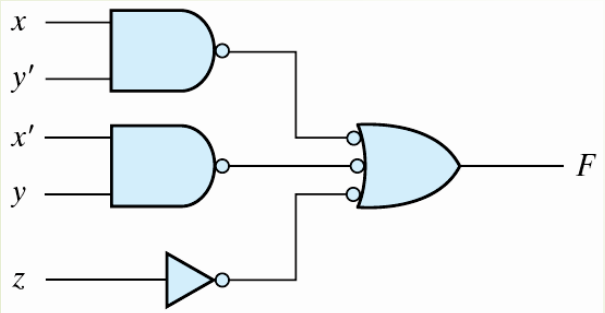
$$F(x,y,z) = \sum(1,2,3,4,5,7)$$



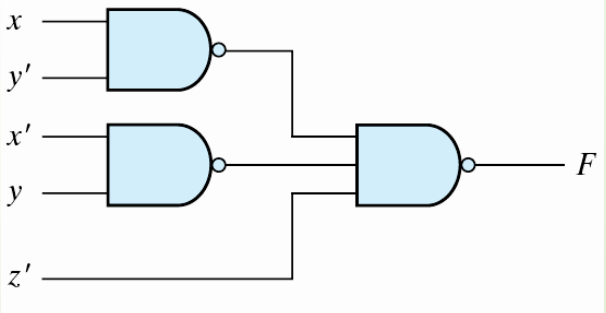
(a)

$$F(x,y,z) = xy' + x'y + z$$

- Λογικό διάγραμμα πυλών



(b)



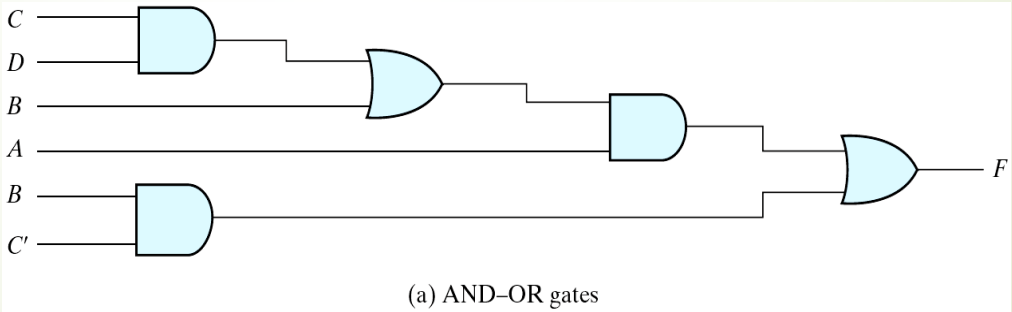
(c)

Διαδικασία Υλοποίησης Δύο Επιπέδων NAND

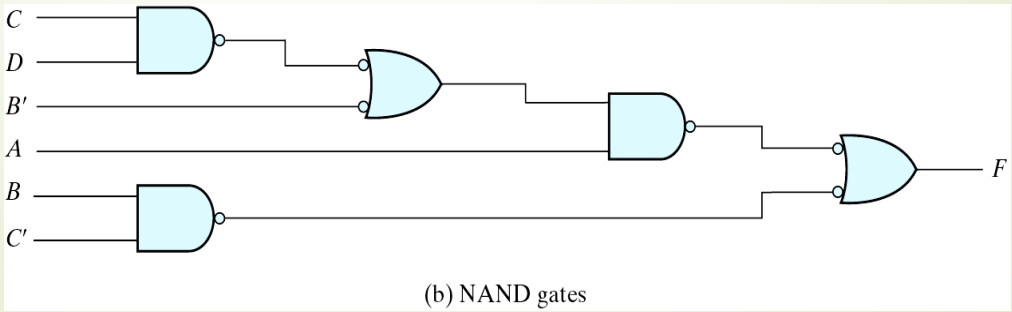
- Τα βήματα
  - Απλοποιήστε τη συνάρτηση σε μορφή αθροίσματος γινομένων
  - Χρησιμοποιήστε στο πρώτο επίπεδο μια πύλη NAND για κάθε γινόμενο. Οι είσοδοι σε κάθε πύλη NAND είναι οι παράγοντες του κάθε ελαχιστόρου
  - Στο δεύτερο επίπεδο χρησιμοποιήστε ***μια*** πύλη NAND
  - Στο πρώτο επίπεδο, όροι με έναν παράγοντα απαιτούν τη χρήση αντιστροφέα

Πολυεπίπεδα κυκλώματα με πύλες NAND

- Υλοποίηση Boolean συνάρτησης
  - AND-OR Λογική → NAND-NAND Λογική
    - AND → AND + Αντιστροφέας = NAND
    - OR: Αντιστροφέας + OR = NAND
    - Για κάθε σύμβολο αντιστροφής (κύκλος) που δεν αντισταθμίζεται από άλλο στην ίδια γραμμή, ενσωματώστε έναν αντιστροφέα



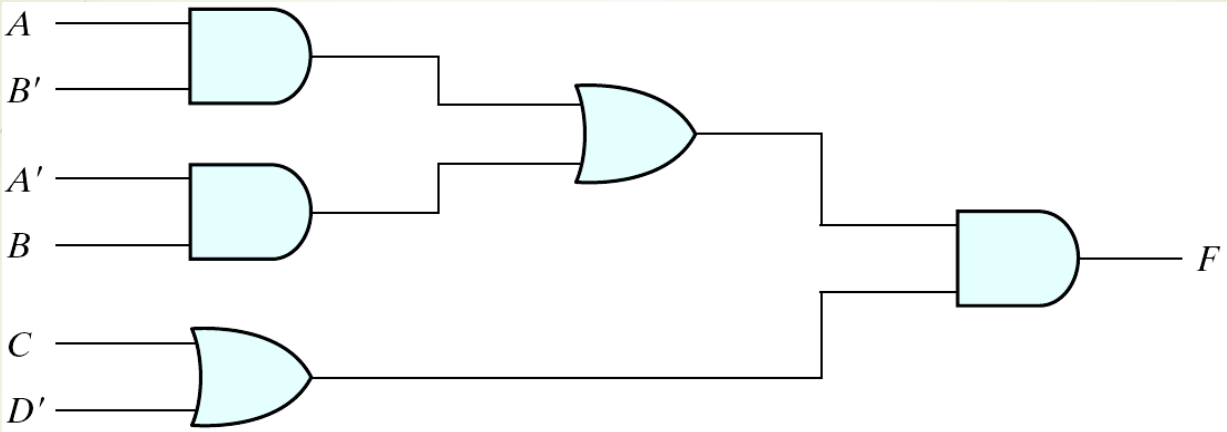
(a) AND–OR gates



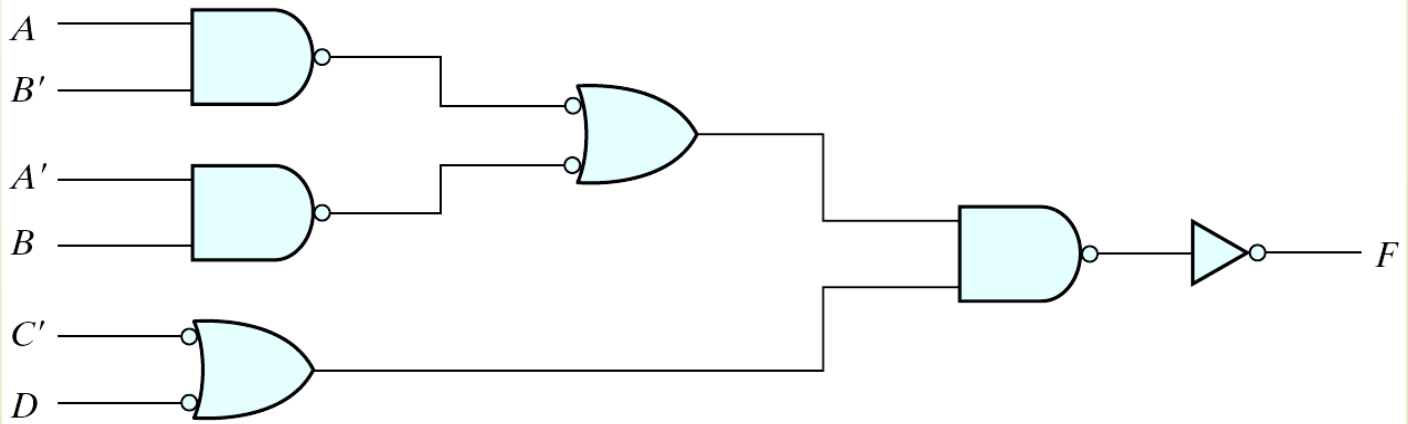
(b) NAND gates

Υλοποίηση της $F = A(CD + B) + BC'$

Υλοποίηση με πύλες NAND



(a) AND–OR gates

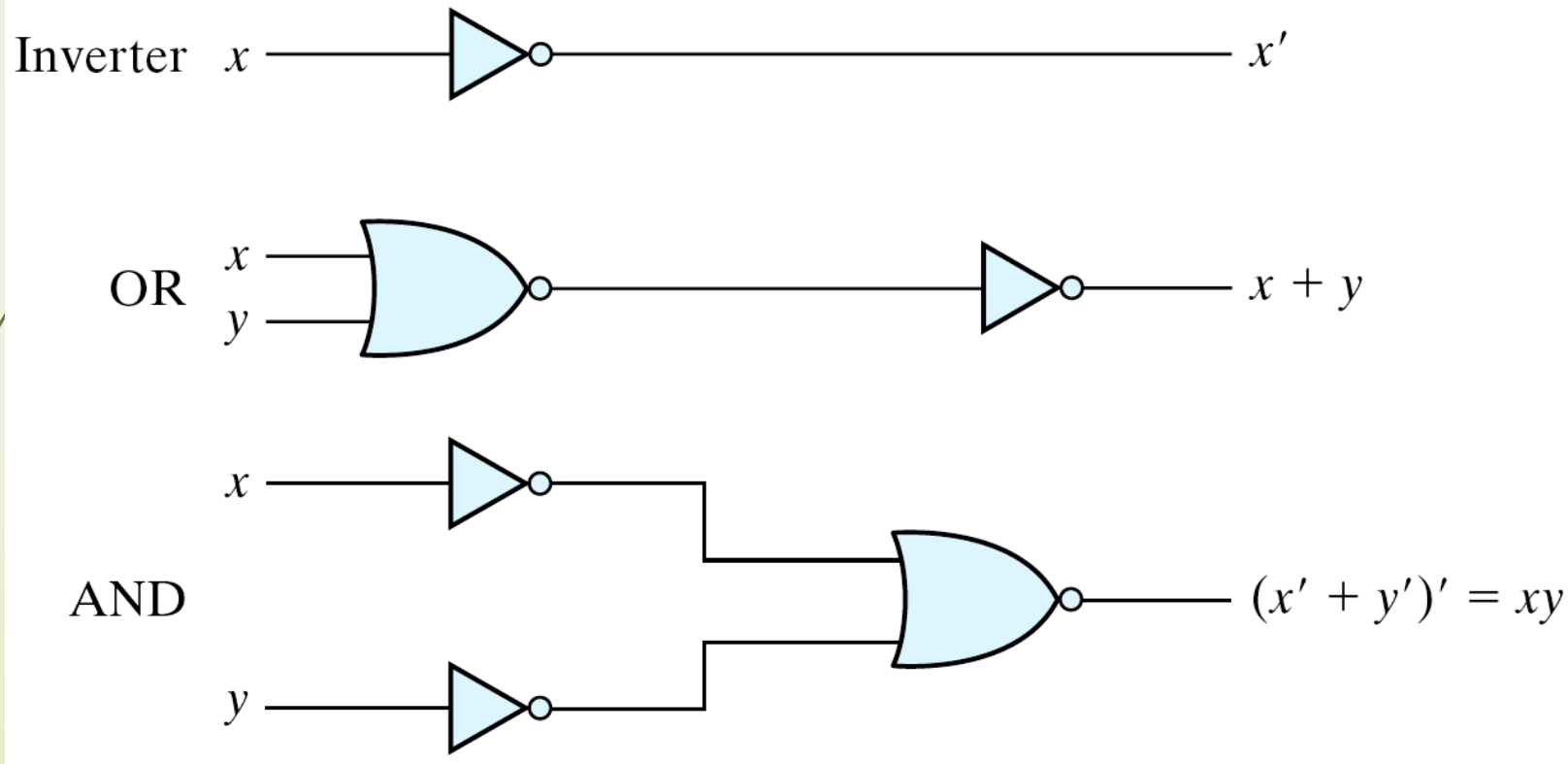


(b) NAND gates

Υλοποίηση της $F = (AB' + A'B)(C + D')$

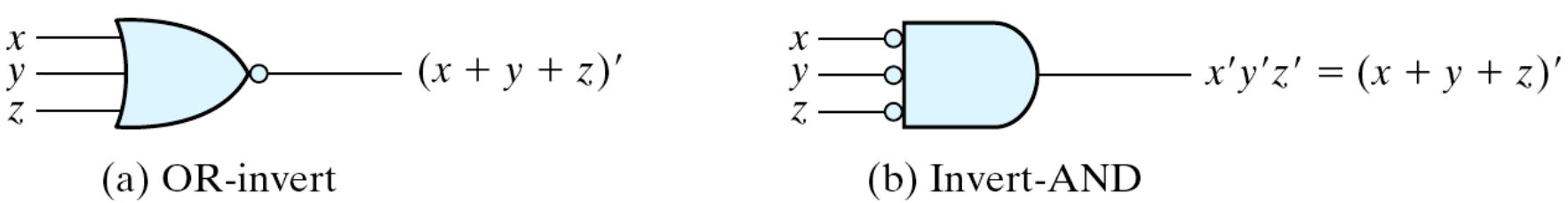
Υλοποίηση με πύλες NOR

- Η συνάρτηση NOR είναι η δυική μορφή της NAND
- Η πύλη NOR είναι επίσης Οικουμενική.

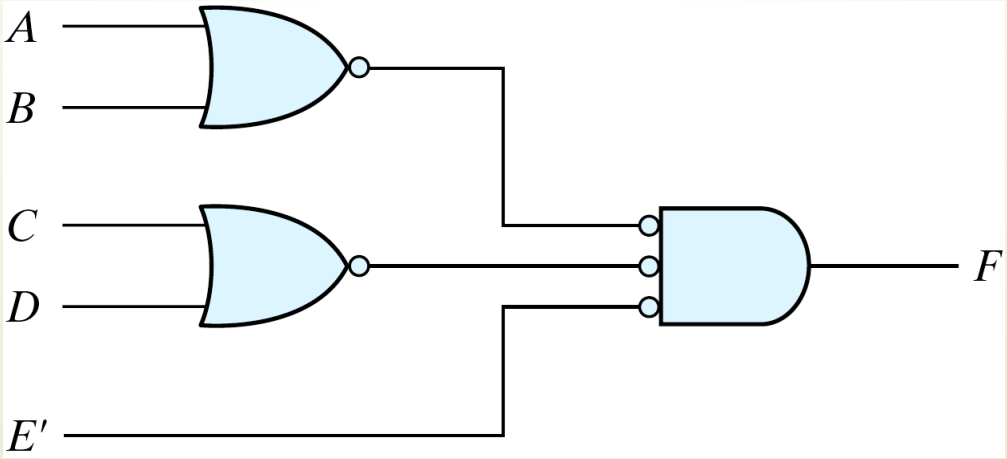


Υλοποίηση με πύλες NOR

Δύο γραφικά σύμβολα για την πύλη NOR



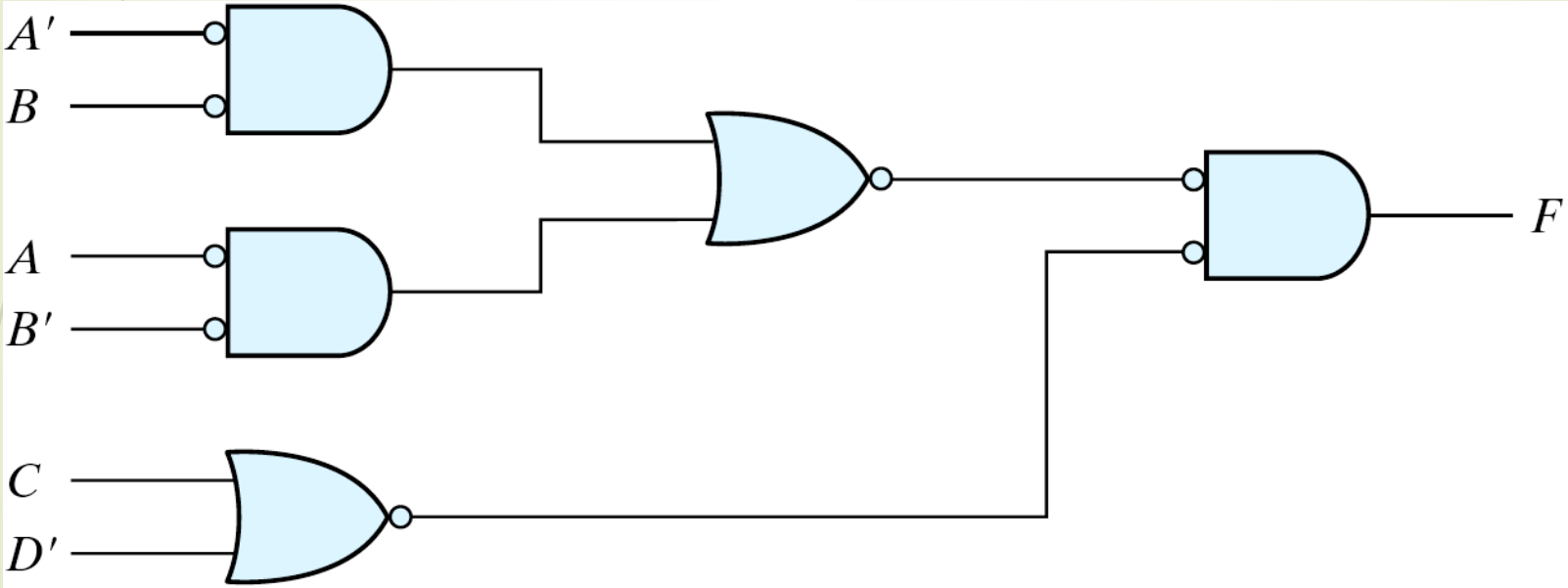
Παράδειγμα: $F = (A + B)(C + D)E$



Υλοποίηση της $F = (A + B)(C + D)E$

Υλοποίηση με πύλες NOR

Παράδειγμα: $F = (AB' + A'B)(C + D')$



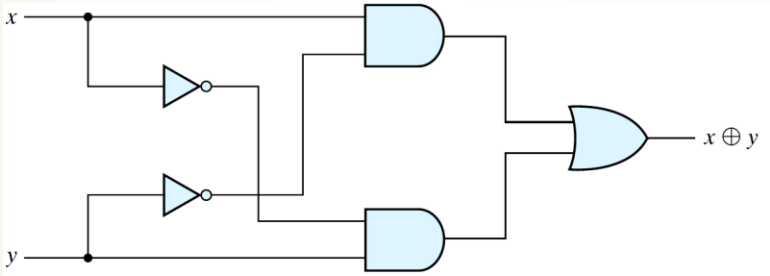
Υλοποίηση της $F = (AB' + A'B)(C + D')$ *με πύλες NOR*

Συνάρτηση Αποκλειστικού-OR (Exclusive-OR)

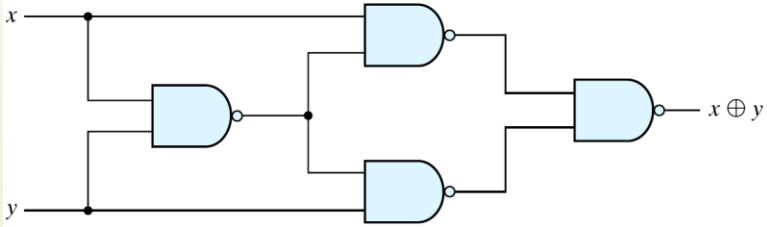
- Exclusive-OR (XOR)
  - $x \oplus y = xy' + x'y$
- Exclusive-NOR (XNOR)
  - $(x \oplus y)' = xy + x'y'$
- Κάποιες ιδιότητες
  - $x \oplus 0 = x$
  - $x \oplus 1 = x'$
  - $x \oplus x = 0$
  - $x \oplus x' = 1$
  - $x \oplus y' = (x \oplus y)'$
  - $x' \oplus y = (x \oplus y)'$
- Προσεταιριστική και Αντιμεταθετική ιδιότητα
  - $A \oplus B = B \oplus A$
  - $(A \oplus B) \oplus C = A \oplus (B \oplus C) = A \oplus B \oplus C$

Υλοποιήσεις με Exclusive-OR

- Παραδείγματα
  - $(x'+y')x + (x'+y')y = xy'+x'y = x \oplus y$



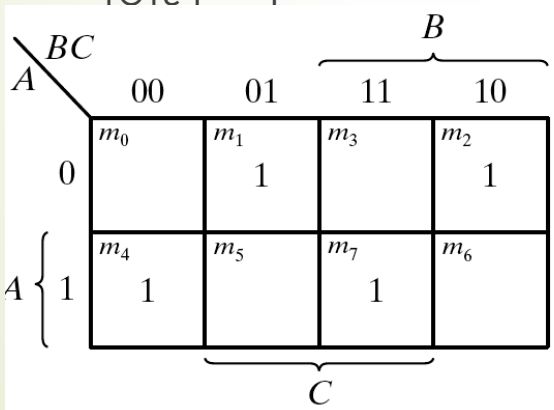
(a) With AND–OR–NOT gates



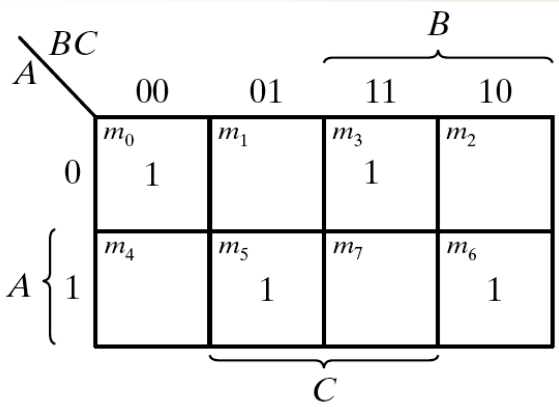
(b) With NAND gates

Περιττή Συνάρτηση

- $A \oplus B \oplus C = (AB'+A'B)C' + (AB+A'B')C = AB'C'+A'BC'+ABC+A'B'C = \Sigma(1, 2, 4, 7)$
- Η XOR είναι μια ***περιττή (odd)*** συνάρτηση → Αν στις εισόδους της εφαρμόζονται περιττός αριθμός λογικών 1, τότε $F = 1$
- Άρα η XNOR είναι μια ***άρτια (even)*** συνάρτηση → Αν στις εισόδους της εφαρμόζονται άρτιος αριθμός λογικών 1, τότε F = 1



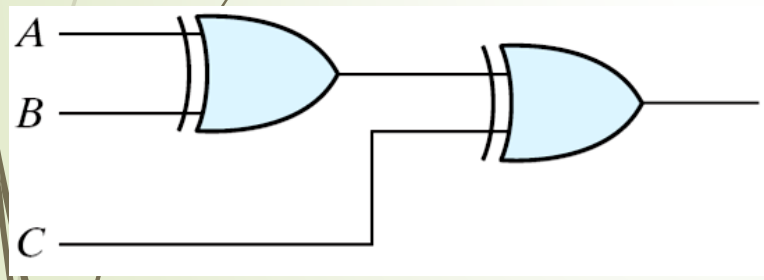
(a) Odd function $F = A \oplus B \oplus C$

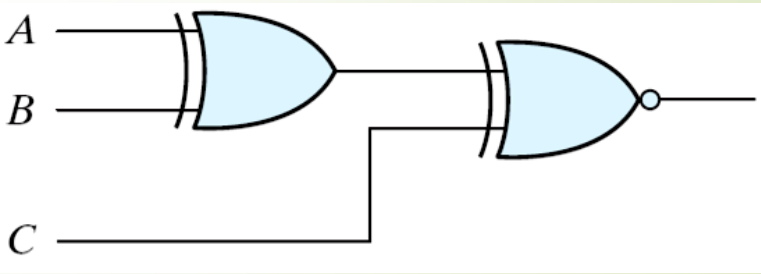


(b) Even function $F = (A \oplus B \oplus C)'$

XOR και XNOR

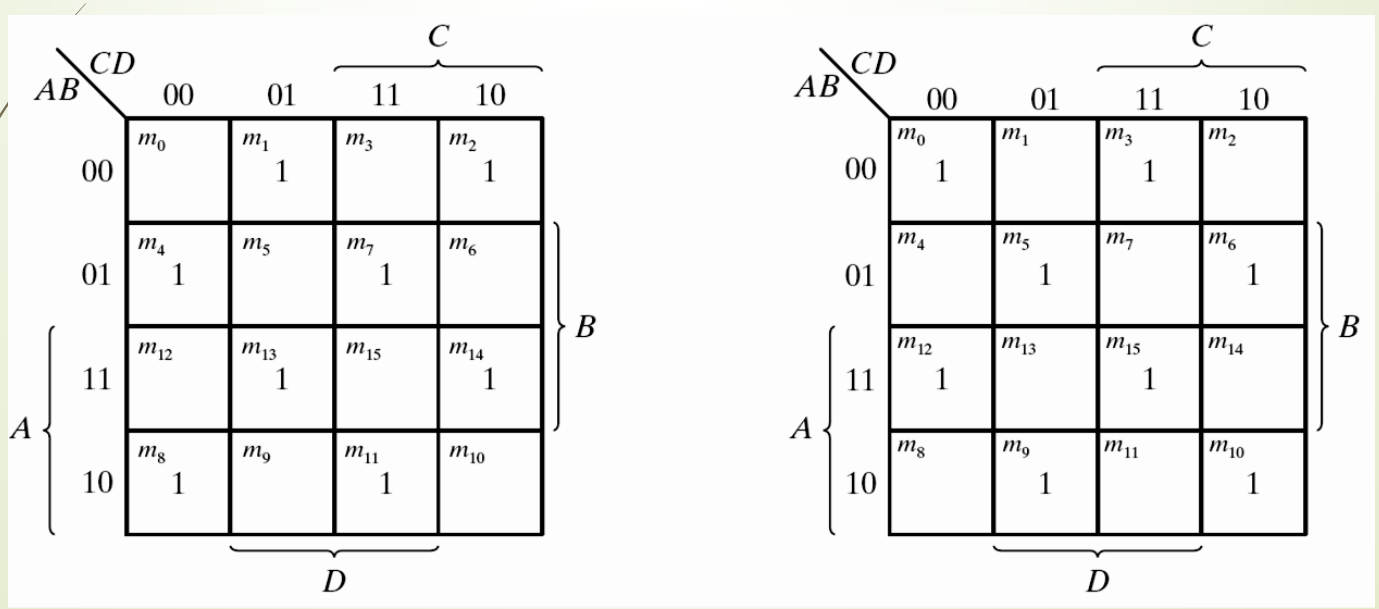
- Λογικό διάγραμμα περιττής και άρτιας συνάρτησης τριών εισόδων





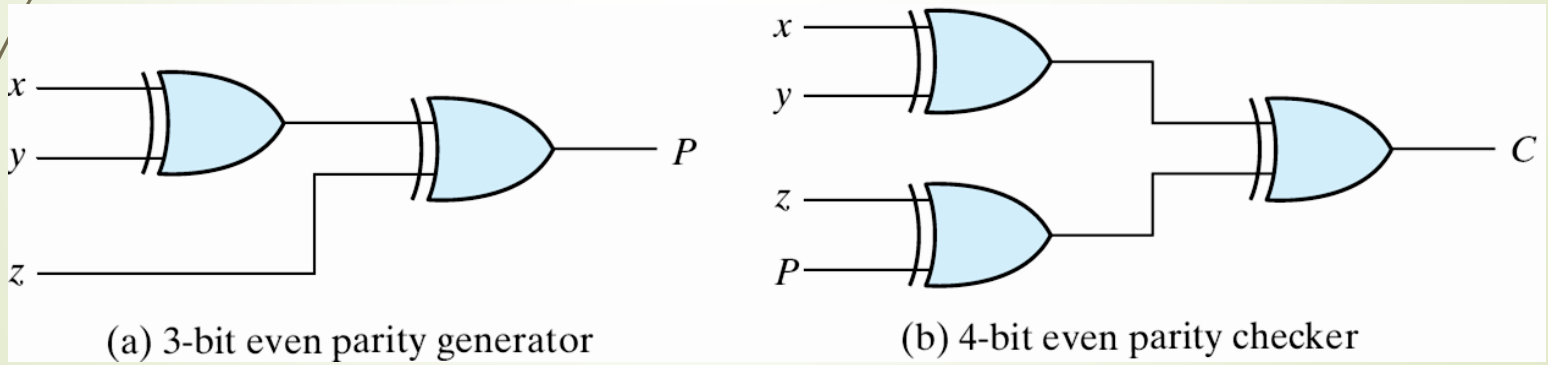
Συνάρτηση Αποκλειστικού-OR Τεσσάρων Μεταβλητών

- $A \oplus B \oplus C \oplus D = (AB'+A'B) \oplus (CD'+C'D) = (AB'+A'B)(CD+C'D')+(AB+A'B')(CD'+C'D)$



Δημιουργία και Έλεγχος Ισοτιμίας

- Bit Ισοτιμίας: $P = x \oplus y \oplus z$
- Έλεγχος Ισοτιμίας: $C = x \oplus y \oplus z \oplus P$
  - C=1: Σφάλμα ενός ή περιττός αριθμός σφαλμάτων σε bit
  - C=0: Χωρίς σφάλμα ή άρτιος αριθμός σφαλμάτων σε bit



(a) 3-bit even parity generator (b) 4-bit even parity checker

Δημιουργία Ισοτιμίας

Μήνυμα 3-bit			Bit ισοτιμίας
x	***y***	***z***	***P***
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

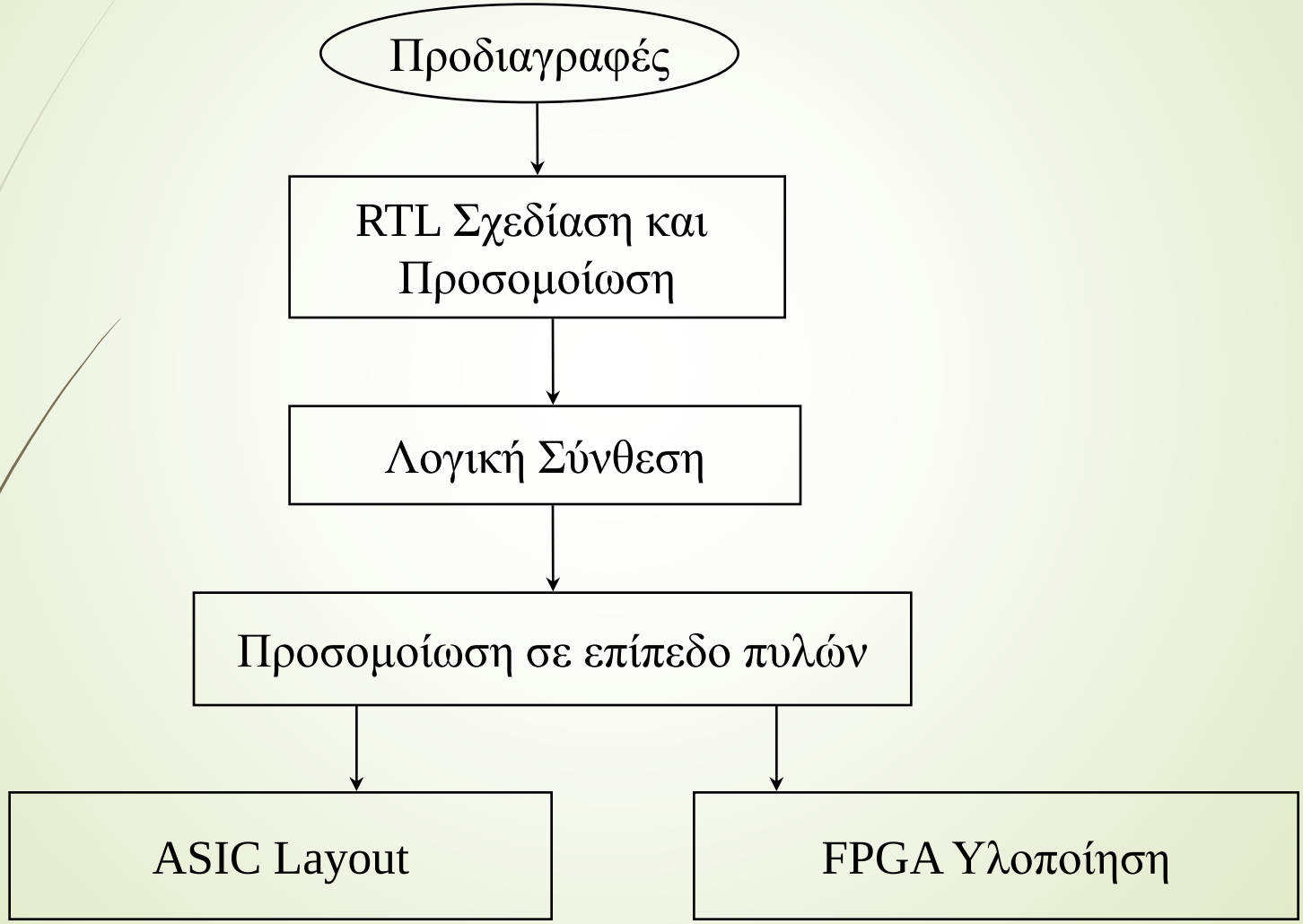
Έλεγχος Ισοτιμίας

Λήψη 4-bit				Έλεγχος
x	***y***	***z***	***P***	***C***
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Γλώσσα Περιγραφής Υλικού Hardware Description Language (HDL)

- Περιγραφή σχεδίασης ψηφιακού συστήματος με τη μορφή κειμένου
  - Δομή
  - Συνάρτηση / Συμπεριφορά
  - Χρονισμός
- VHDL και Verilog HDL

Ροή Σχεδίασης Top-Down



VHDL

- Η αποτύπωση ενός κυκλώματος σε κώδικα vhdl χωρίζεται σε δύο μέρη:
  - την οντότητα (entity): Όνομα και διεπαφή (Είσοδοι / Έξοδοι) κυκλώματος
  - την αρχιτεκτονική (architecture): Περιγραφή λογικής / λειτουργίας

Δήλωση οντότητας

Περιγραφή

- **entity** όνομα_οντότητας **is**
  - **port**(
    - όνομα_σήματος : κατεύθυνση_σήματος τύπος_σήματος;
    - όνομα_σήματος : κατεύθυνση_σήματος τύπος_σήματος;

      ...
    - όνομα_σήματος : κατεύθυνση_σήματος τύπος_σήματος
  -);
- **end** όνομα_οντότητας;

in, out, inout, buffer

Παράδειγμα

- **entity** Example **is**
  - **port**(y_out: out bit; x_in: in bit);
- **end** Example;

- **Σημείωση**: Η γλώσσα vhdl δεν κάνει διάκριση μεταξύ κεφαλαίων και πεζών χαρακτήρων

Δήλωση Αρχιτεκτονικής

Περιγραφή

- **architecture** όνομα_αρχιτεκτονικής **of** όνομα_οντότητας **is**
  - δηλώσεις τύπων δεδομένων
  - δηλώσεις σημάτων
  - δήλωσης σταθερών
  - ορισμοί συναρτήσεων
  - ορισμοί διαδικασιών
  - δηλώσεις στοιχειωδών κυκλωμάτων
- **begin**
  - εντολές ταυτόχρονης εκτέλεσης
- **end** όνομα_αρχιτεκτονικής

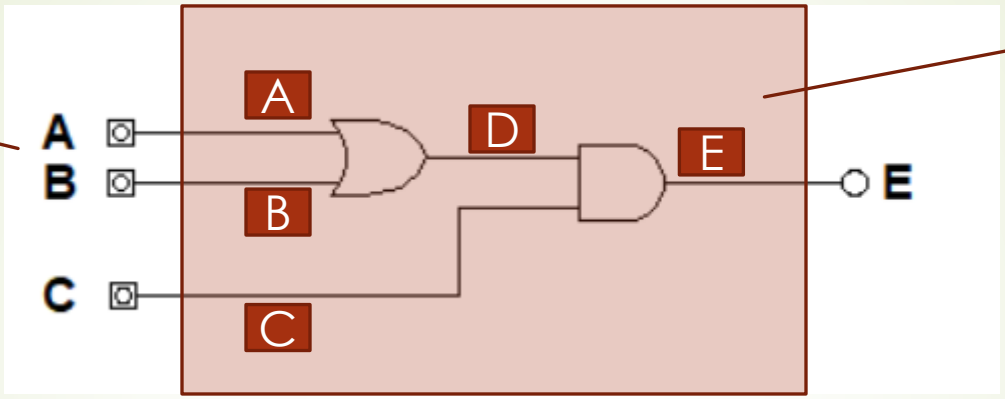
Σήματα

- Σε επίπεδο υλικού τα σήματα υλοποιούνται ως ηλεκτρικές συνδέσεις ενός κυκλώματος και αναπαριστούν τα λογικά δεδομένα που επεξεργάζεται αυτό το κύκλωμα.
- Δήλωση σήματος: **signal** λίστα_ονομάτων_σημάτων : τύπος_σήματος;
  - Πχ signal A, B, C: bit;

Παράδειγμα VHDL

Κύκλωμα ***or_and_vhdl*** για επίδειξη της VHDL

Οντότητα

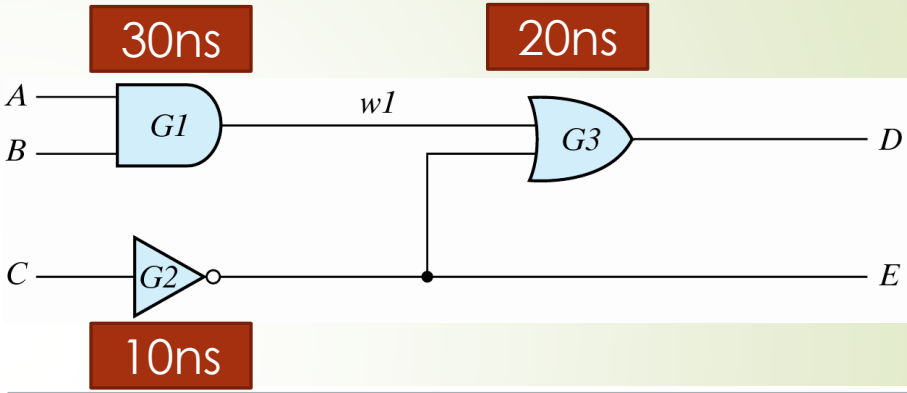


Αρχιτεκτονική

```
entity or_and_vhdl is
      port(E: out bit; A, B, C: in bit);
end or_and_vhdl;

architecture Boolean_Equations of or_and_vhdl is
      signal D: bit;
begin
      D <= A or B;
      E <= C and D;
end Boolean_Equations;
```

Καθυστερήσεις Διάδοσης Πυλών

- Καθυστέρηση Διάδοσης
  - Υπάρχει χρονική καθυστέρηση μεταξύ της αλλαγής τιμής μιας εισόδου του κυκλώματος και της αντίστοιχα προκαλούμενης αλλαγής τιμής κάποιας εξόδου του.



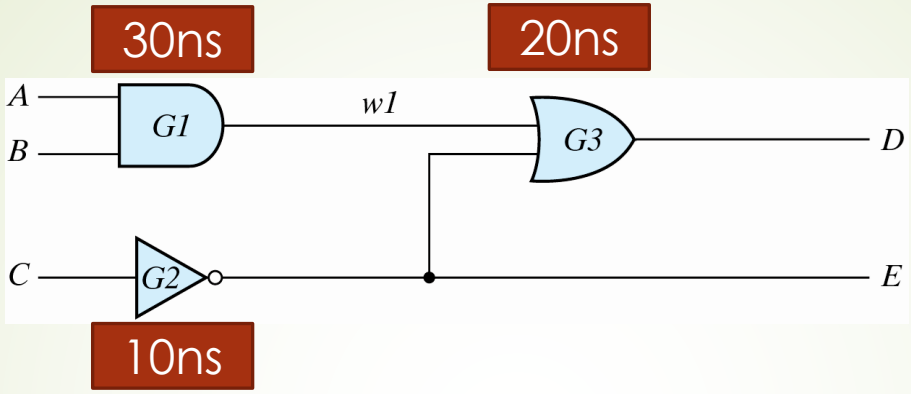
	Μονάδα Χρόνου (ns)	Είσοδος ABC	Έξοδος E w1 D
Αρχική Αλλαγή	—	0 0 0	1 0 1
	—	1 1 1	1 0 1
	10	1 1 1	0 0 1
	20	1 1 1	0 0 1
	30	1 1 1	0 1 0
	40	1 1 1	0 1 0
	50	1 1 1	0 1 1

Μοντελοποίηση Δομής

- Τα δομικά μοντέλα περιγράφουν πως συντίθεται ένα κύκλωμα από διάφορα διασυνδεδεμένα μεταξύ τους στοιχεία όπως λογικές πύλες ή και πιο σύνθετα στοιχειώδη κυκλώματα
- Περιγραφή στοιχειώδους κυκλώματος στη VHDL:
  - **Component** όνομα_στοιχειώδους_κυκλώματος
    - **Port** (
      - όνομα_σήματος : κατεύθυνση_σήματος τύπος_σήματος;
      - όνομα_σήματος : κατεύθυνση_σήματος τύπος_σήματος;

        …
      - όνομα_σήματος : κατεύθυνση_σήματος τύπος_σήματος
    -);
  - **end component;**

Παράδειγμα (3.2)



Δίνεται αρχικά η περιγραφή των στοιχειωδών κυκλωμάτων:

```
entity and2_gate is
      port(w1: out bit; A, B: in bit);
end and2_gate;
architecture Boolean_Operator of and2_gate is
Begin
--Τελεστής με καθυστέρηση διάδοσης
      w1 <= A and B after 30 ns;
end Boolean_Operator;
```

```
entity or2_gate is
      port(D: out bit; w1, E: in bit);
end or2_gate;
architecture Boolean_Operator of or2_gate is
Begin
--Τελεστής με καθυστέρηση διάδοσης
      D <= w1 or E after 20 ns;
end Boolean_Operator;
```

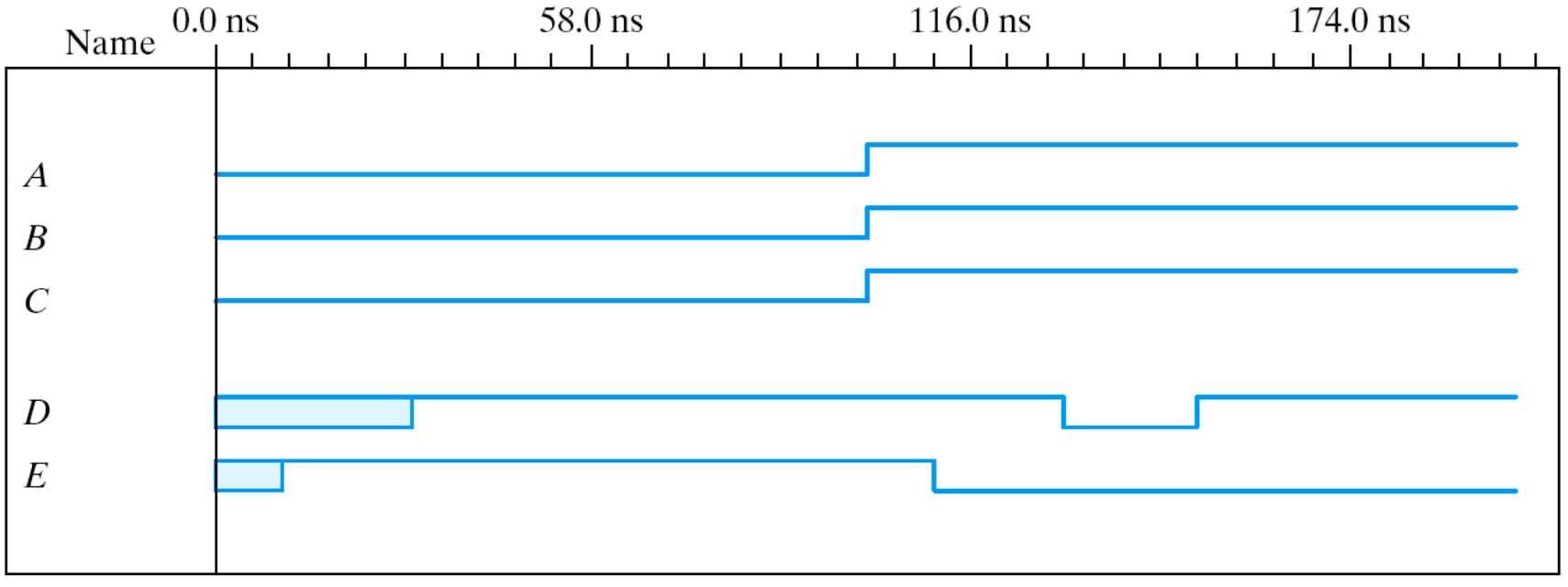
```
entity inv_gate is
      port(B: out bit; A: in bit);
end inv_gate;
architecture Boolean_Operator of inv_gate is
Begin
--Τελεστής με καθυστέρηση διάδοσης
      B <= not A after 10 ns;
end Boolean_Operator;
```

Παράδειγμα (3.2)

Στη συνέχεια ορίζουμε το τελικό κύκλωμα ή αλλιώς την κορυφαία σχεδίαση:

```
entity and_or_prop_delay is
      port(D: out bit; A, B, C: in bit; E: buffer bit);
end and_or_prop_delay;

architecture Structure of and_or_prop_delay is
      component and2_gate
            port(w1: out bit; A, B: in bit);
      end component;

      component or2_gate
            port(D: out bit; w1, E: in bit);
      end component;

      component inv_gate
            port(B: out bit; A: in bit);
      end component;

      signal w1: bit;

begin
--Δημιουργία στιγμιότυπων των στοιχειωδών κυκλωμάτων
      G1:and2_gate port map (w1=>w1, A=>A, B=>B);
      G2:inv_gate port map (B=>E, A=>C);
      G3:or2_gate port map (D=>D, w1=>w1, E=>E);
 end Structure;
```

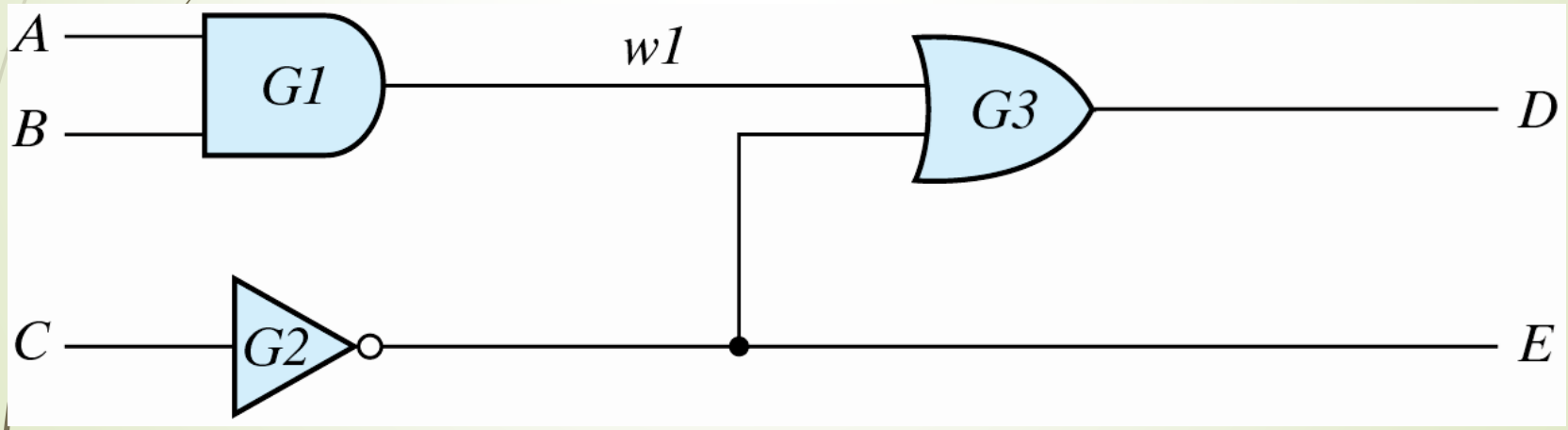
Έξοδος Προσομοίωσης HDL



Verilog – Δήλωση Οντότητας

- Λέξεις κλειδιά:

  module, **end-module**, **input**, **output**, **wire**, **and**, **or**, και **not**



Κύκλωμα για Παράδειγμα

Παράδειγμα Verilog (3.3)

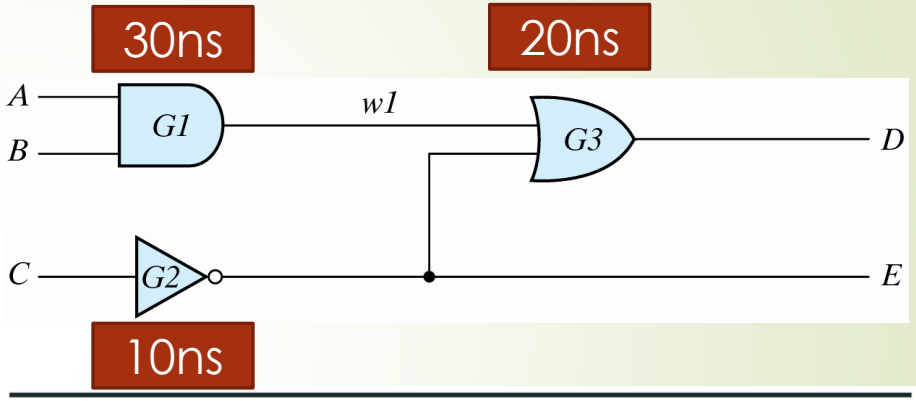
```
// Verilog model of circuit of Figure 3.37. IEEE 1364–1995 Syntax

module  Simple_Circuit (A, B, C, D, E);
 output          D, E;
 input           A, B, C;
 wire            w1;

 and             G1 (w1, A, B); // Optional gate instance name
 not             G2 (E, C);
 or              G3 (D, w1, E);
endmodule
```

Καθυστερήσεις Διάδοσης Πυλών

- Η οδηγία timescale της Verilog
  - ‘timescale μονάδες / ακρίβεια κλίμακας
  - Π.χ.‘timescale 1 ns/100ps

HDL Example 3.2 (Gate-level model with propagation delays)

```
// Verilog model of simple circuit with propagation delay
‘timescale 1 ns/100ps
module Simple_Circuit_prop_delay (A, B, C, D, E);
 output  D, E;
 input   A, B, C;
 wire    w1;

 and          #(30) G1 (w1, A, B);
 not          #(10) G2 (E, C);
 or           #(20) G3 (D, w1, E);
endmodule
```



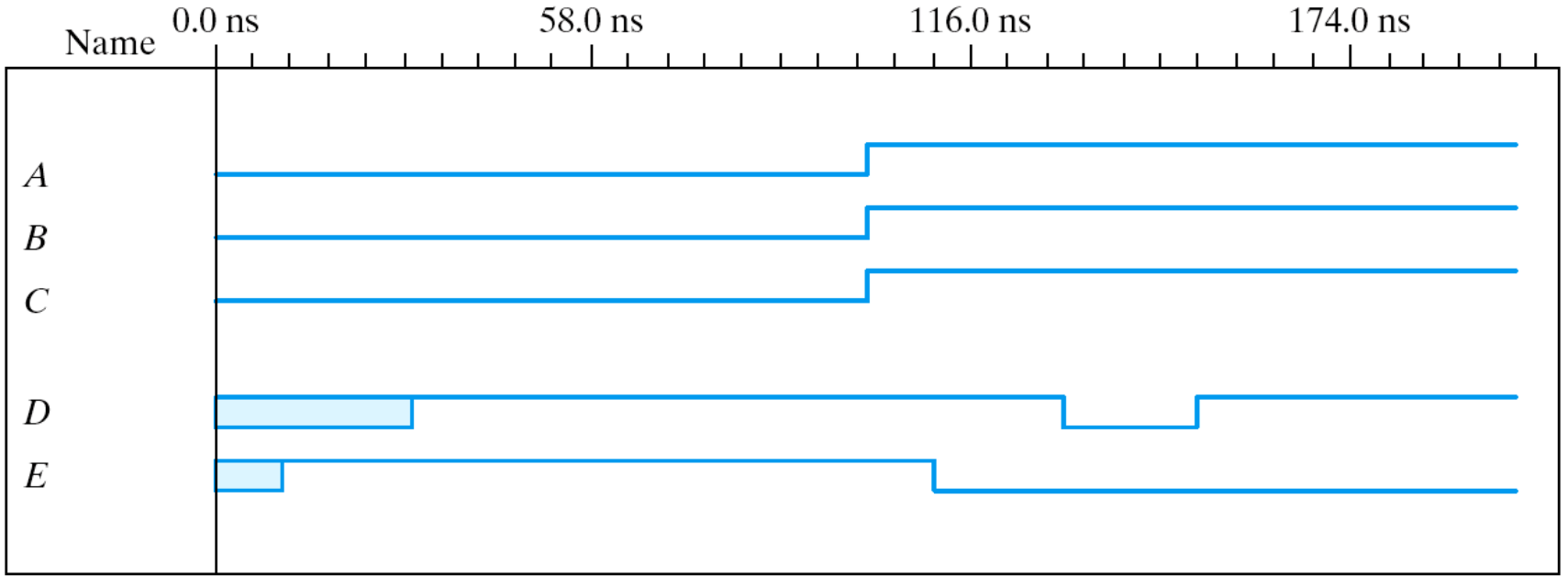
	Μονάδα Χρόνου (ns)	Είσοδος ABC	Έξοδος E w1 D
Αρχική Αλλαγή	—	0 0 0	1 0 1
	—	1 1 1	1 0 1
	10	1 1 1	0 0 1
	20	1 1 1	0 0 1
	30	1 1 1	0 1 0
	40	1 1 1	0 1 0
	50	1 1 1	0 1 1

Παράδειγμα Verilog (3.3)

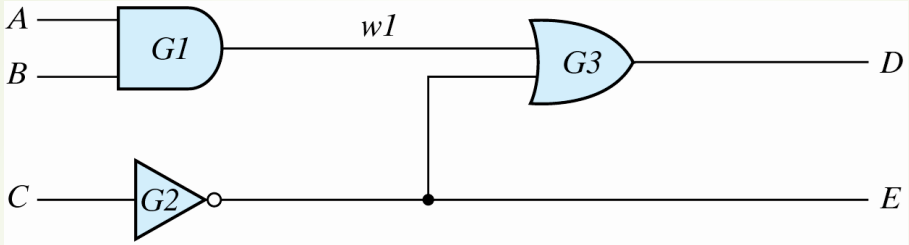
- Μοντέλο δοκιμής (Test bench) για την προσομοίωση του κυκλώματος με καθυστέρηση

```
// Test bench for Simple_Circuit_prop_delay

module t_Simple_Circuit_prop_delay;
 wire      D, E;
 reg       A, B, C;

 Simple_Circuit_prop_delay M1 (A, B, C, D, E); // Instance name required

 initial
  begin
   A = 1'b0; B = 1'b0; C = 1'b0;
   #100 A = 1'b1; B = 1'b1; C = 1'b1;
  end

 initial #200 $finish;
endmodule
```

Έξοδος Προσομοίωσης



Παράδειγμα Verilog Boolean Συναρτήσεις



- Boolean έκφραση για το παραπάνω κύκλωμα

```
assign D = (A & B)|~C;
```

Παράδειγμα Verilog Boolean Συναρτήσεις

$$E = A + BC + B'D$$

$$F = B'C + BC'D'$$

```
// Verilog model: Circuit with Boolean expressions

module Circuit_Boolean_CA (E, F, A, B, C, D);
 output      E, F;
 input       A, B, C, D;

 assign E = A | (B & C) | (~B & D);
 assign F = (~B & C) | (B & ~C & ~D);
endmodule
```

Στοιχειώδη Κυκλώματα Ορισμένα από τον Χρήστη

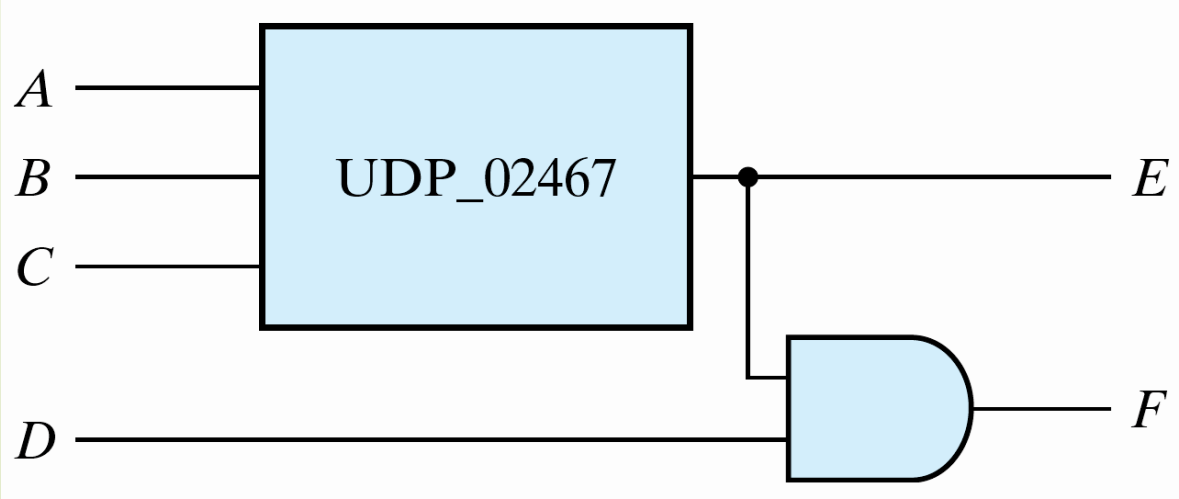
- Περιγραφή στοιχειώδους κυκλώματος στη Verilog (Δεν υπάρχει η δυνατότητα αυτή στη VHDL):
  - Το στοιχειώδες κύκλωμα δηλώνεται με τη δεσμευμένη λέξη **primitive**, την οποία ακολουθούν το όνομα του στοιχειώδους κυκλώματος και ένας κατάλογος θυρών
  - Επιτρέπεται να υπάρχει μόνο μία έξοδος κυκλώματος, η οποία πρέπει να αναφέρεται πρώτη στον κατάλογο θυρών και να δηλώνετε με τη δεσμευμένη λέξη **output**
  - Δεν υπάρχει περιορισμός στον αριθμό των εισόδων. Η σειρά με την οποία αναφέρονται στη δήλωση εισόδων δηλαδή την εντολή με τη δεσμευμένη λέξη **input** πρέπει να ακολουθεί τη σειρά με την οποία αυτές λαμβάνουν τιμές στον πίνακα που ακολουθεί
  - Ο πίνακας αληθείας περικλείεται από τις δεσμευμένες λέξεις **table** και **endtable**
  - Οι τιμές των εισόδων σε κάθε γραμμή του πίνακα τίθενται με τη σειρά με την οποία δηλώθηκαν στην εντολή input. Μετά την τελευταία τιμή γράφεται ':' και ακολουθεί η αντίστοιχη έξοδος ως τελευταία καταχώρηση κάθε γραμμής του πίνακα. Ακολουθείται από ένα ελληνικό ερωτηματικό
  - Η δήλωση ενός στοιχειώδους κυκλώματος τελειώνει με τη δεσμευμένη λέξη **endprimitive**

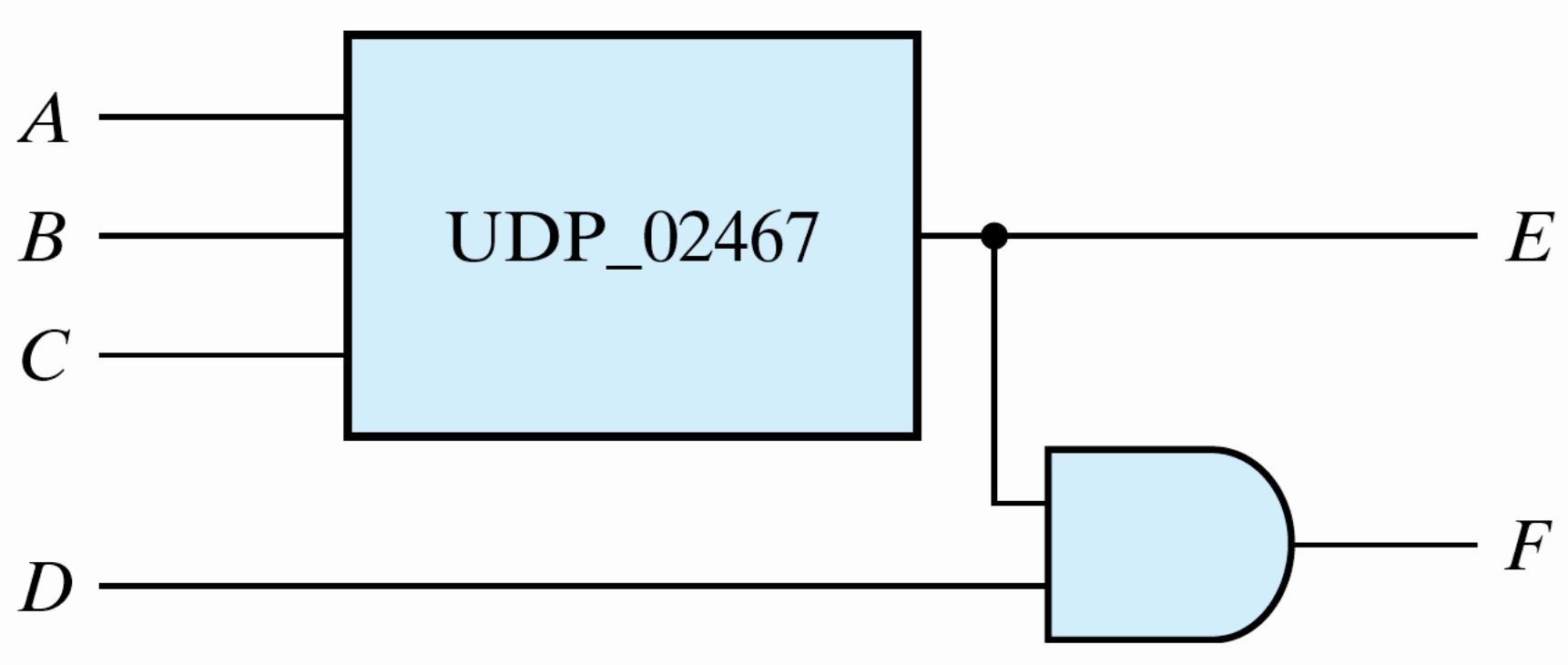
Παράδειγμα (3.4)

```
// Verilog model: User-defined Primitive

primitive UDP_02467 (D, A, B, C);
 output D;
 input   A, B, C;

// Truth table for D = f (A, B, C) = Σ (0, 2, 4, 6, 7);
 table
//    A     B     C     :     D     // Column header comment
      0     0     0     :     1;
      0     0     1     :     0;
      0     1     0     :     1;
      0     1     1     :     0;
      1     0     0     :     1;
      1     0     1     :     0;
      1     1     0     :     1;
      1     1     1     :     1;
 endtable
endprimitive
```

Παράδειγμα (3.4)

```
// Instantiate primitive

// Verilog model: Circuit instantiation of Circuit_UDP_02467

module Circuit_with_UDP_02467 (e, f, a, b, c, d);
 output      e, f;
 input       a, b, c, d;

 UDP_02467          (e, a, b, c);
 and                (f, e, d);          // Option gate instance name omitted
endmodule
```




A
B
C
D
UDP_02467
E
F