



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
Εθνικόν και Καποδιστριακόν  
Πανεπιστήμιον Αθηνών

# Σχεδίαση CMOS Ψηφιακών Ολοκληρωμένων Κυκλωμάτων

Ενότητα 6: Σχεδιασμός Κυκλωμάτων σε Επίπεδο  
Τρανζίστορ

Αγγελική Αραπογιάννη

Σχολή Θετικών Επιστημών

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

# CMOS

- Κάθε λογική πύλη αποτελείται από δύο τμήματα
  - p-MOS δικτύωμα, τοποθετείται μεταξύ τροφοδοσίας και εξόδου. Όταν είναι ενεργό φορτίζει την έξοδο στην τάση τροφοδοσίας με αποτέλεσμα η έξοδος να έχει λογική τιμή "1"
  - n-MOS δικτύωμα, τοποθετείται μεταξύ εξόδου και γείωσης. Όταν είναι ενεργό αποφορτίζει την έξοδο στην γείωση με αποτέλεσμα η έξοδος να έχει λογική τιμή "0"



# Προσοχή

- Ποτέ δεν πρέπει να είναι ενεργά και το p-MOS και το n-MOS δίκτυο ταυτόχρονα
  - Σε αυτή την περίπτωση συνδέουμε την γείωση με την πηγή τροφοδοσίας. Το αποτέλεσμα θα είναι μεγάλο ρεύμα μέσω του κυκλώματος που οδηγεί σε υψηλή κατανάλωση και πιθανή καταστροφή του κυκλώματος
  - Κατά την αλλαγή της λογικής τιμής υπάρχει περίπτωση "μερικής" (με αυξημένη αντίσταση) σύνδεσης πηγής τροφοδοσίας και γείωση, σε αυτή την περίπτωση το ρεύμα είναι μειωμένο και υπάρχει μόνο για μικρό χρονικό διάστημα οπότε δεν δημιουργούνται προβλήματα
  - Υπάρχουν λογικές οικογένειες όπου το χρονικό διάστημα δεν είναι περιορισμένο - σε αυτές τις περιπτώσεις πρέπει να εξασφαλίσουμε ότι το διερχόμενο ρεύμα δεν θα δημιουργήσει προβλήματα



# Κυκλώματα Τριών Καταστάσεων

- Μπορούμε να έχουμε και τα δύο τμήματα (n-MOS και p-MOS) ανενεργά ταυτόχρονα
- Σε αυτή την περίπτωση στην έξοδο έχουμε μία τρίτη κατάσταση (υψηλή εμπέδηση - high impedance - hiZ)
- Σε αυτή την κατάσταση η πύλη δεν επιδρά στην έξοδο
  - Μπορούμε να συνδέσουμε εξόδους πολλών πυλών ταυτόχρονα εάν δεν έχουμε περισσότερες από μία σε κατάσταση διαφορετική από hiZ
  - Εάν όλες οι συνδεδεμένες εξοδοί σε κόμβο είναι hiZ η κατάσταση παραμένει σταθερή αλλά είναι ευαίσθητη στον θόρυβο.



# Κανόνες Σχεδιασμού για CMOS

- Μπορούμε να υλοποιήσουμε μόνο αναστρέφουσες συναρτήσεις (inverting functions), NAND, NOR, NOT κ.τ.λ.
  - Όταν όλες οι είσοδοι είναι λογικό "1" το τμήμα n-MOS είναι ενεργό και η έξοδος είναι λογικό "0"
  - Όταν όλες οι είσοδοι είναι λογικό "0" το τμήμα p-MOS είναι ενεργό και η έξοδος είναι λογικό "1"



# NOR πύλη

- n-MOS τμήμα: Εάν έστω και μία είσοδος είναι λογικό "1" η έξοδος πρέπει να είναι λογικό "0",
  - Κάθε είσοδος οδηγεί την πύλη ενός τρανζίστορ που συνδέει την έξοδο με την γείωση
- p-MOS τμήμα: Εάν όλες οι είσοδοι είναι λογικό "0" η έξοδος πρέπει να είναι λογικό "1",
  - Το τμήμα αποτελείται από τρανζίστορ σε σειρά που συνδέουν την τροφοδοσία με την έξοδο, κάθε είσοδος οδηγεί την πύλη ενός τρανζίστορ



# NAND πύλη

- n-MOS τμήμα: Εάν όλες οι είσοδοι είναι λογικό "1" η έξοδος πρέπει να είναι λογικό "0",
  - Το τμήμα αποτελείται από τρανζίστορ σε σειρά που συνδέουν την γείωση με την έξοδο, κάθε είσοδος οδηγεί την πύλη ενός τρανζίστορ
- p-MOS τμήμα: Εάν έστω και μία είσοδος είναι λογικό "0" η έξοδος πρέπει να είναι λογικό "1",
  - Κάθε είσοδος οδηγεί την πύλη ενός τρανζίστορ που συνδέει την έξοδο με την τροφοδοσία



# Γενικοί κανόνες για σειριακά-παράλληλα δικτυώματα (series-parallel networks) (1/3)

## Κανόνες για το n-MOS τμήμα

- Έστω ότι έχω τα n-δικτυώματα των συναρτήσεων  $F'$  και  $G'$
- Το δικτύωμα της  $(F+G)'$  παράγεται εάν τοποθετήσω παράλληλα τα υπάρχοντα δικτυώματα
  - Η  $(F+G)'$  θα είναι λογικό "0" εάν η  $F'$  ή η  $G'$  είναι λογικό "0". Επομένως το n-δικτύωμα της  $(F+G)'$  πρέπει να άγει εάν είτε το n-δικτύωμα της  $F'$  είτε το n-δικτύωμα  $G'$  άγει





# Γενικοί κανόνες για σειριακά-παράλληλα δικτυώματα (series-parallel networks) (2/3)

- Το δικτύωμα της  $(FG)'$  παράγεται εάν τοποθετήσω σε σειρά τα υπάρχοντα δικτυώματα
  - Η  $(FG)'$  θα είναι λογικό "1" εάν και η  $F'$  και η  $G'$  είναι λογικό "1". Επομένως το n-δικτύωμα της η  $(F+G)'$  πρέπει να άγει εάν και το n-δικτύωμα της  $F'$  και το n-δικτύωμα  $G'$  άγει



# Γενικοί κανόνες για σειριακά-παράλληλα δικτυώματα (series-parallel networks)

(3/3)

## Κανόνες για το $p$ -δικτύωμα

- Έστω ότι έχω τα  $p$ -δικτυώματα των συναρτήσεων  $F'$  και  $G'$
- Το δικτύωμα της  $(F+G)'$  παράγεται εάν τοποθετήσω σε σειρά τα υπάρχοντα δικτυώματα
  - Η  $(F+G)'$  θα είναι λογικό "1" εάν και η  $F'$  και η  $G'$  είναι λογικό "1". Επομένως το  $p$ -δικτύωμα της  $(F+G)'$  πρέπει να άγει εάν και το  $p$ -δικτύωμα της  $F'$  και το  $p$ -δικτύωμα της  $G'$  άγει
- Το δικτύωμα της  $(FG)'$  παράγεται εάν τοποθετήσω παράλληλα τα υπάρχοντα δικτυώματα
  - Η  $(FG)'$  θα είναι λογικό "1" εάν η  $F'$  ή η  $G'$  είναι λογικό "1". Επομένως το  $p$ -δικτύωμα της  $(FG)'$  πρέπει να άγει εάν είτε το  $p$ -δικτύωμα της  $F'$  είτε το  $p$ -δικτύωμα  $G'$  άγει



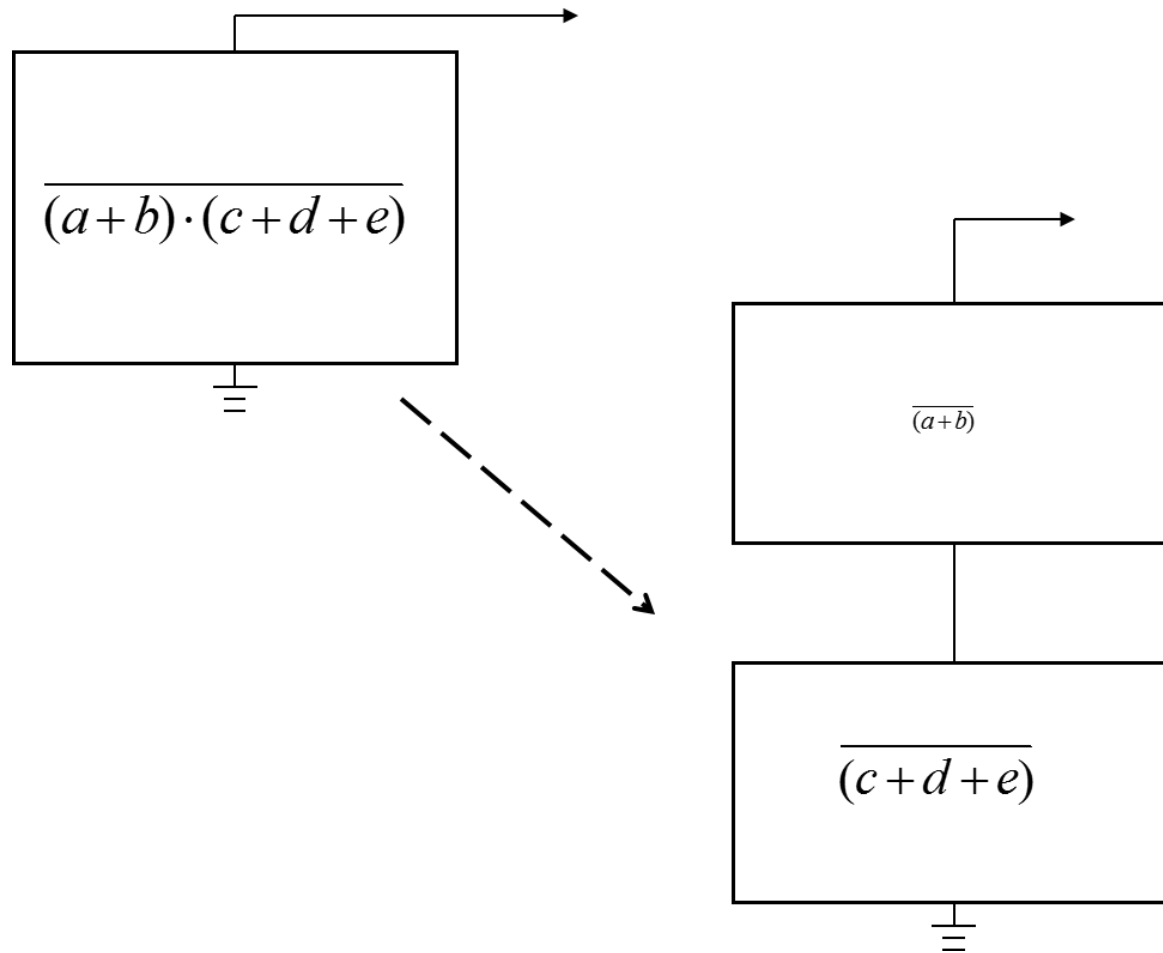
# Παράδειγμα (1 από 3)

Υλοποιήστε το n-δικτύωμα της ακόλουθης συνάρτησης

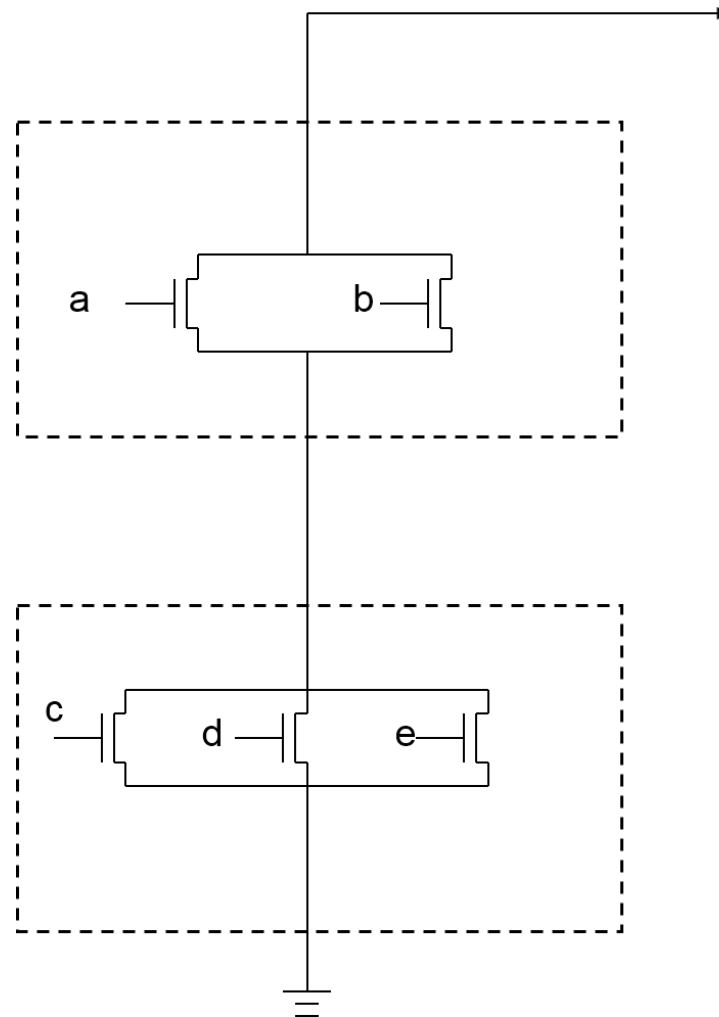
$$\overline{(a + b) \cdot (c + d + e)}$$



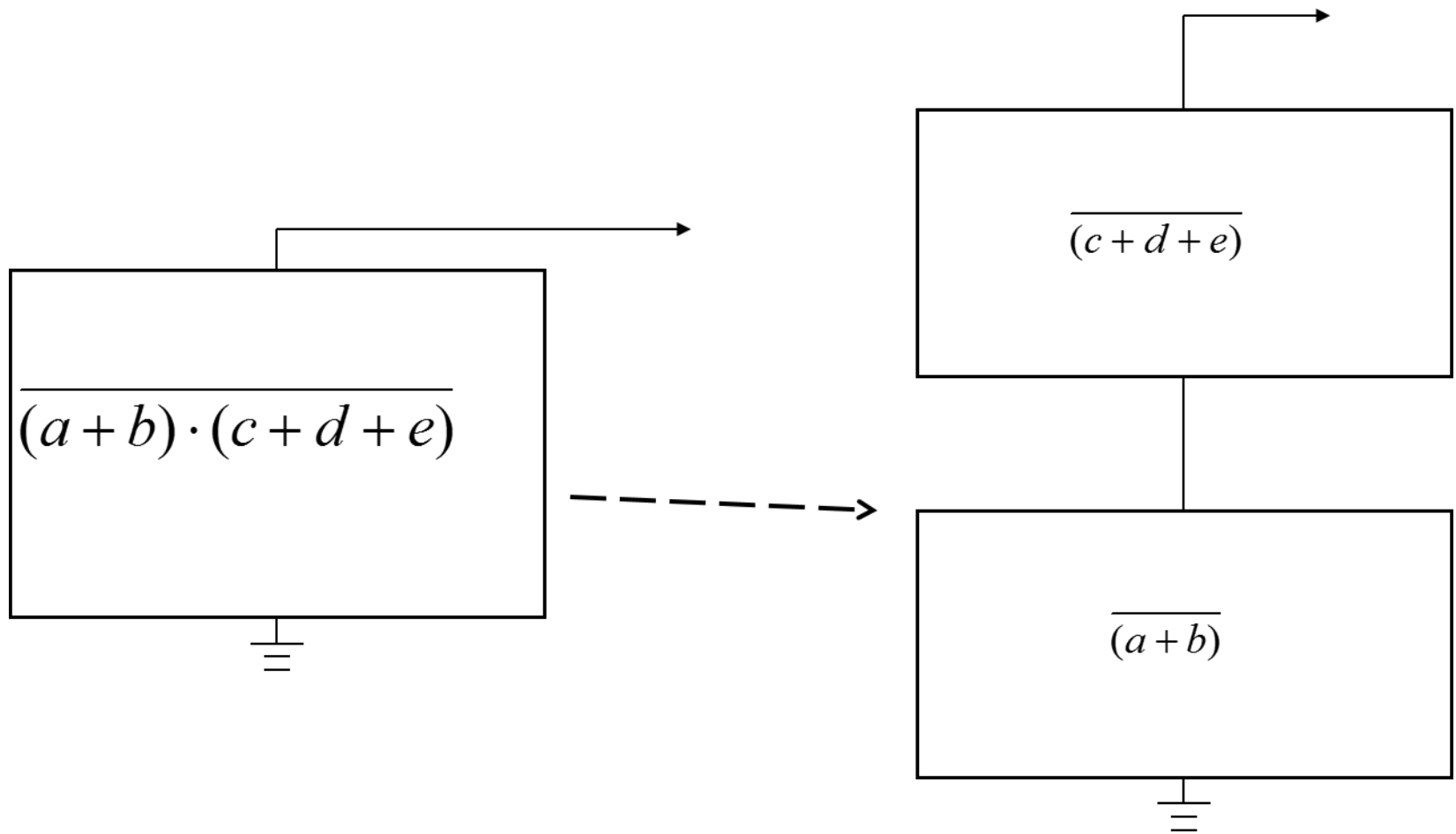
# Παράδειγμα (2 από 3)



# Παράδειγμα (3 από 3)



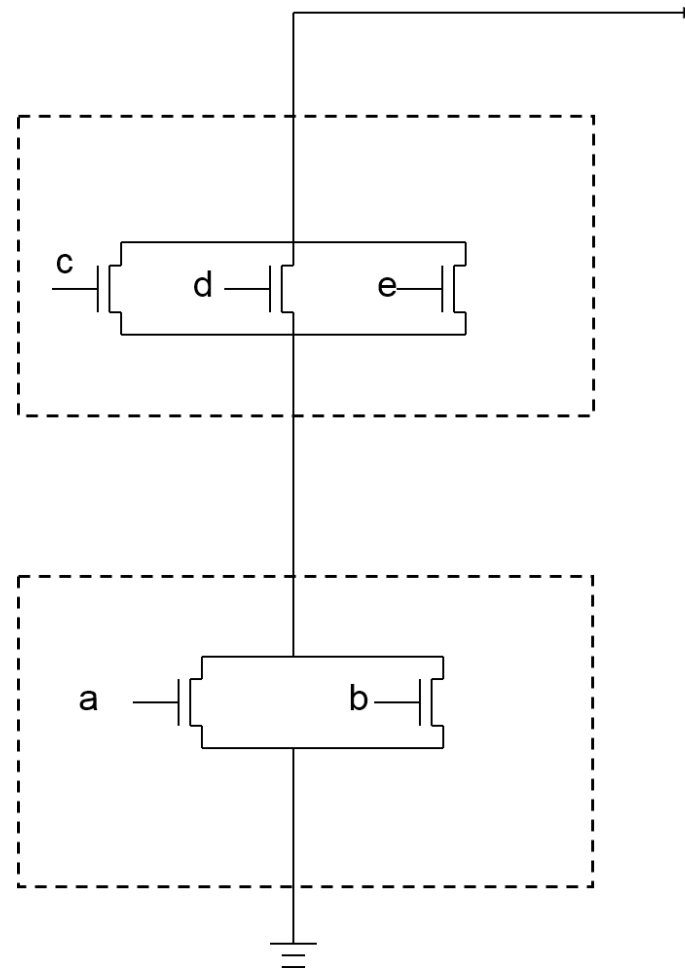
# Εναλλακτική υλοποίηση (1 από 2)



# Εναλλακτική υλοποίηση (2 από 2)

Διαφέρει

- Περισσότερη χωρητικότητα στην έξοδο
- Επιθυμητό η χωρητικότητα να είναι στη γείωση ή την τροφοδοσία, όχι στη έξοδο



## Παράδειγμα 2 (1 από 3)

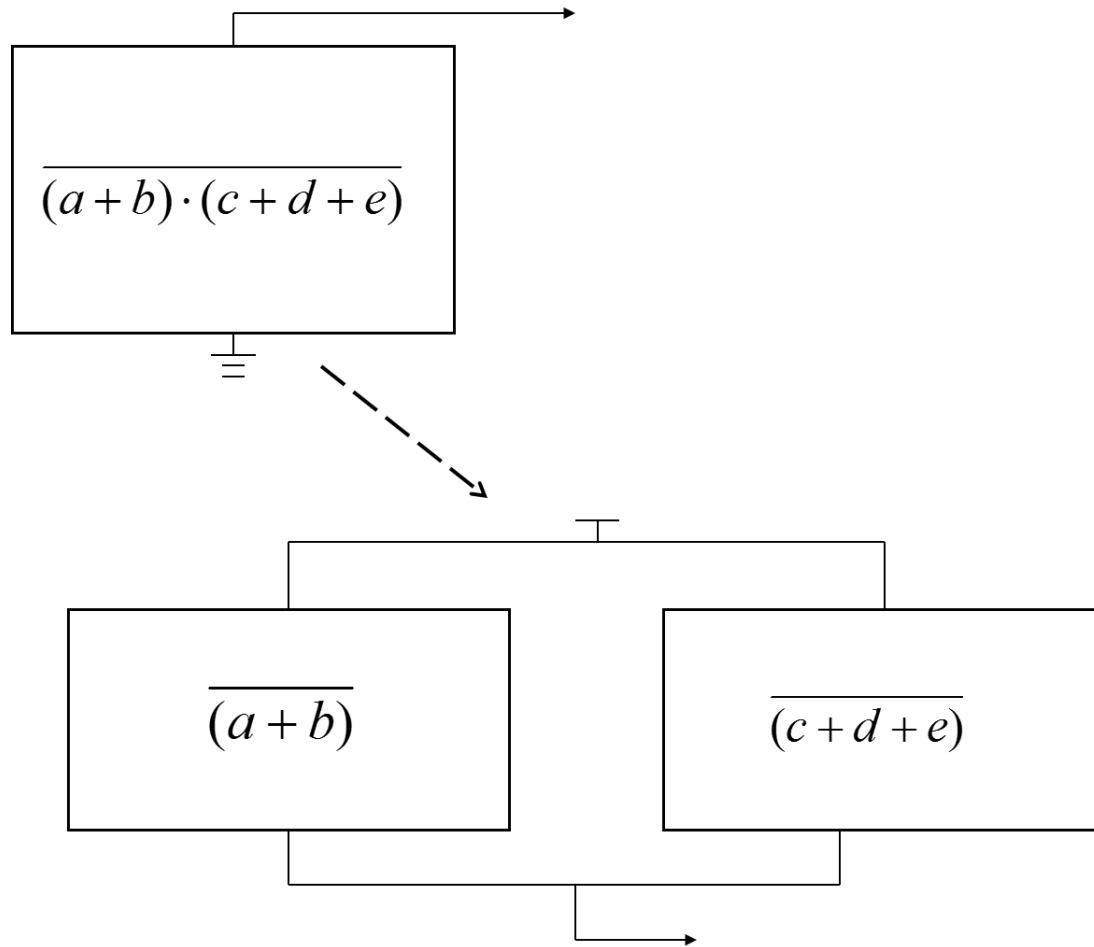
- Υλοποιήστε το p-δικτύωμα της ακόλουθης συνάρτησης

$$\overline{(a + b) \cdot (c + d + e)}$$

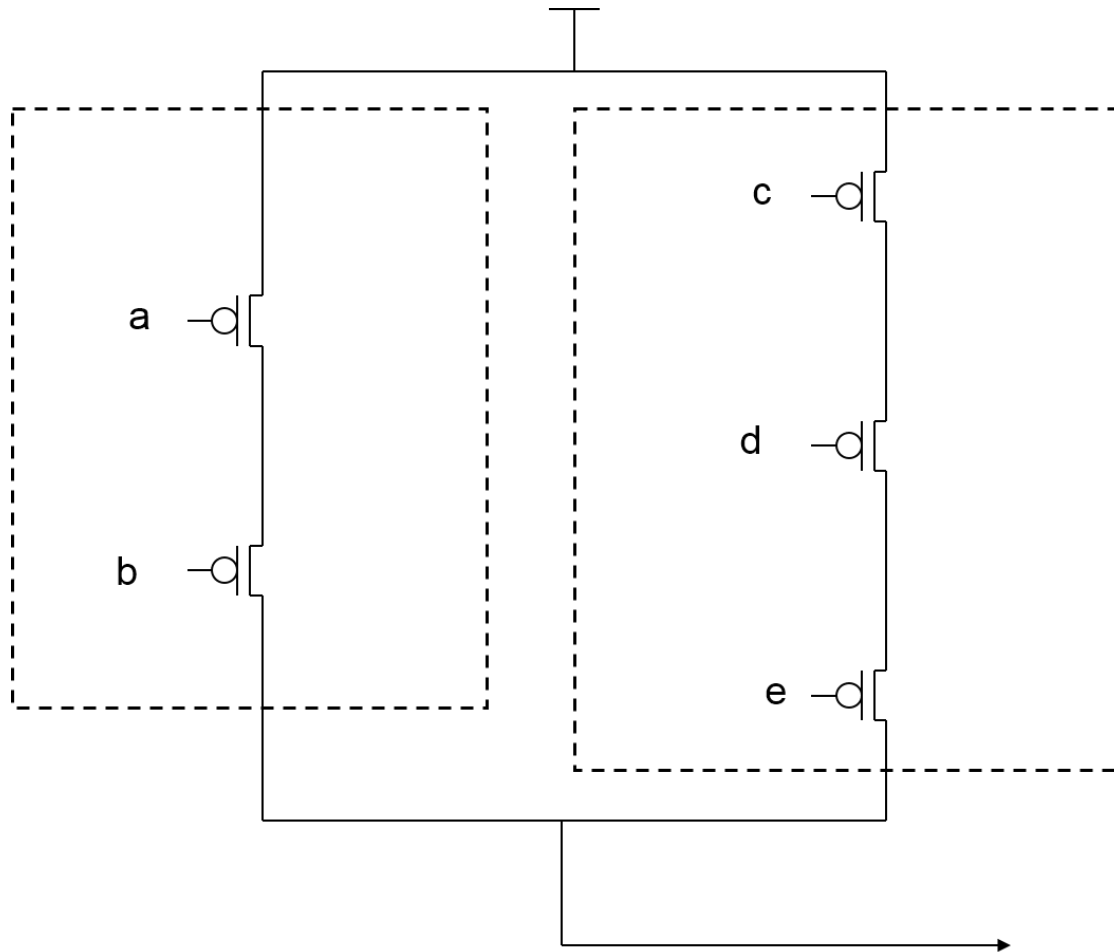




# Παράδειγμα 2 (2 από 3)



# Παράδειγμα 2 (3 από 3)



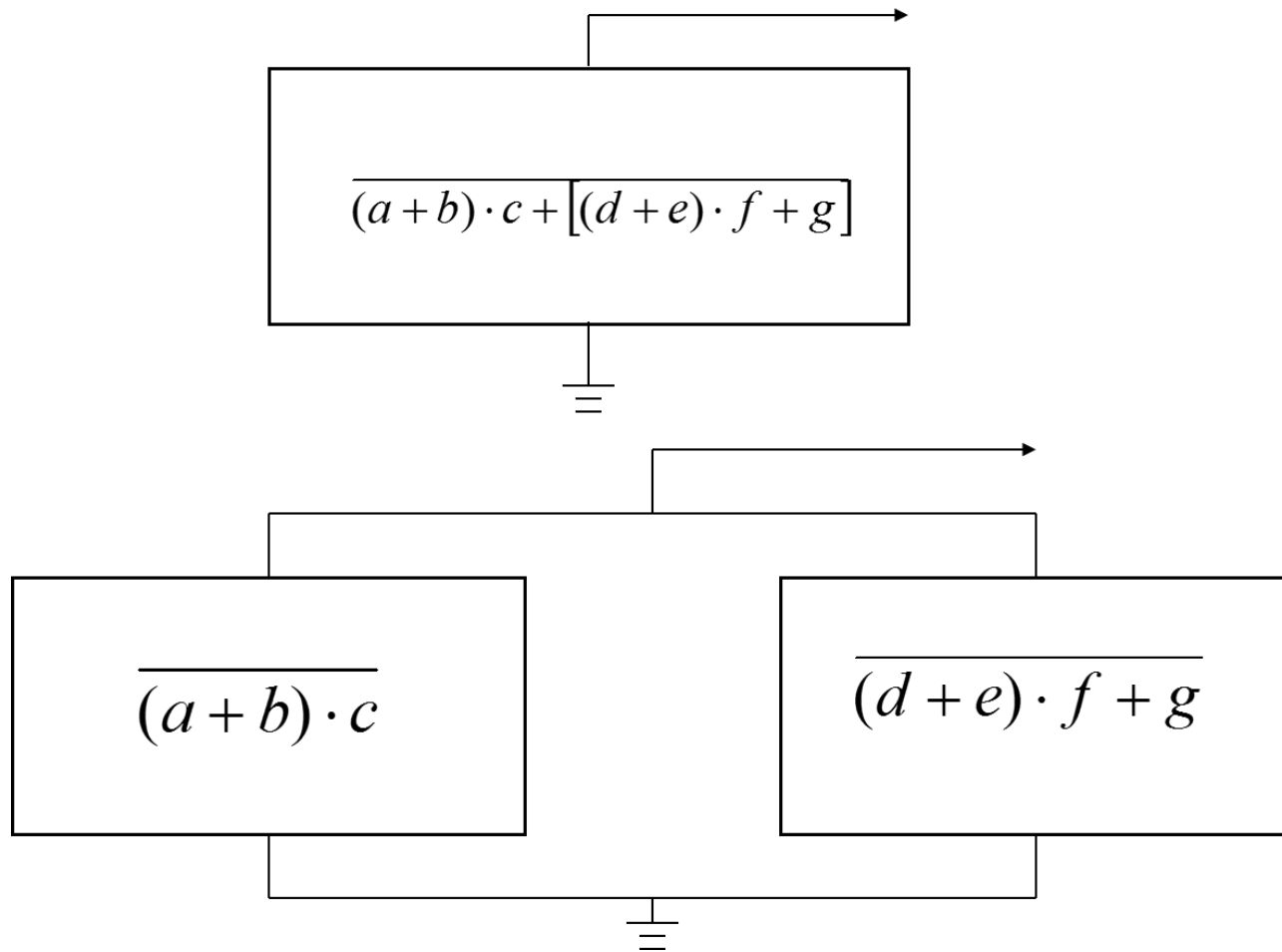
## Παράδειγμα 3 (1 από 4)

Υλοποιήστε το n-δικτύωμα της ακόλουθης συνάρτησης

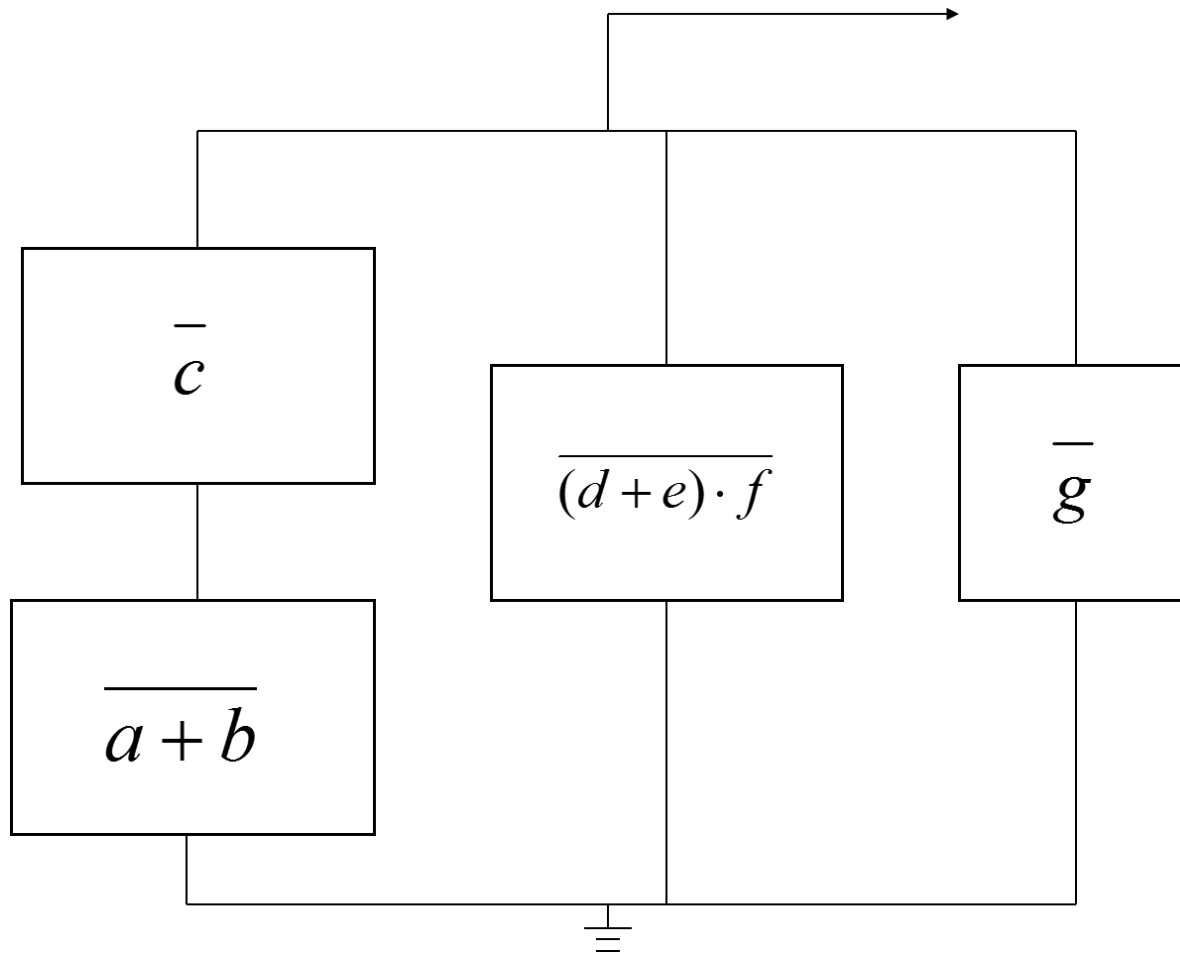
$$\overline{(a + b) \cdot c + [(d + e) \cdot f + g]}$$



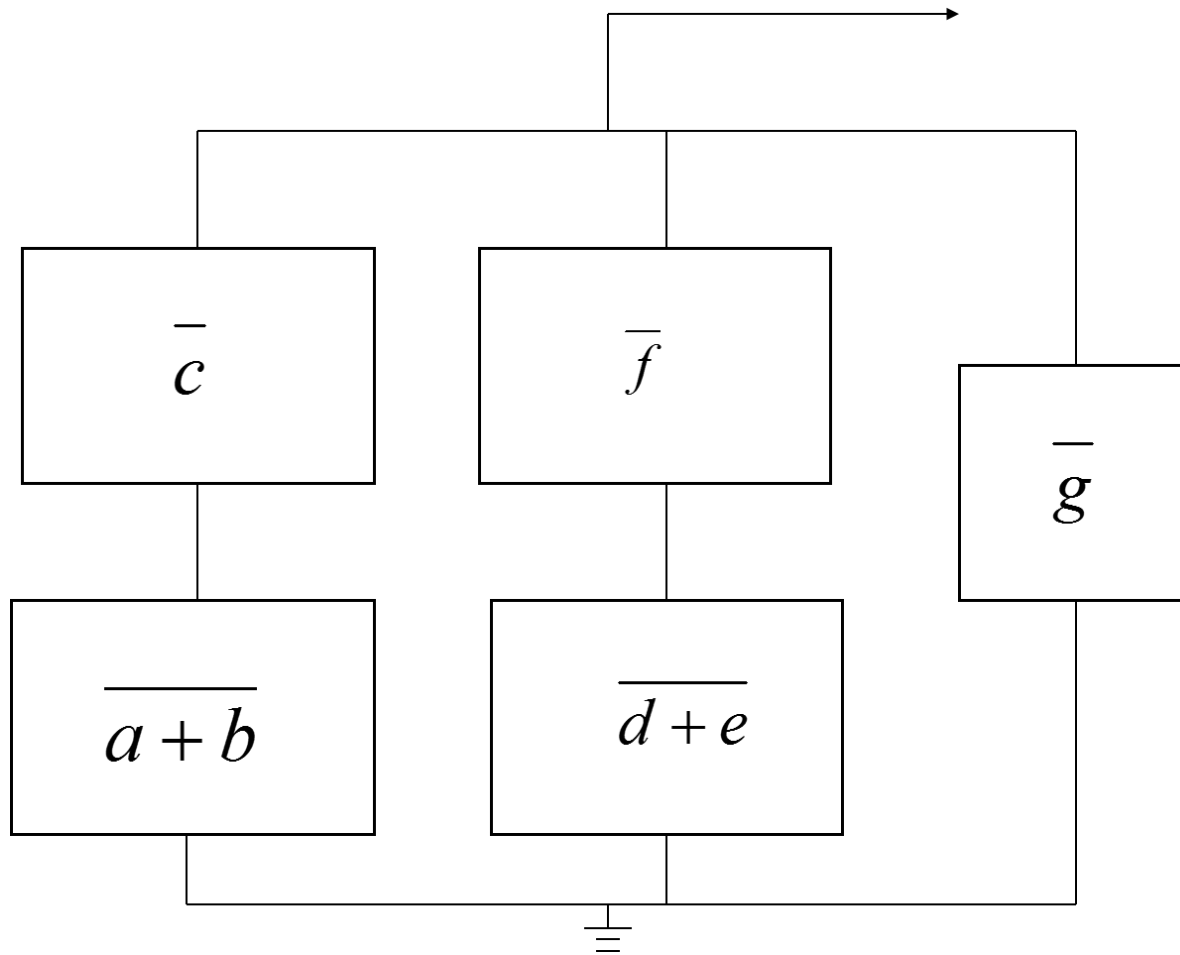
# Παράδειγμα 3 (2 από 4)



# Παράδειγμα 3 (3 από 4)

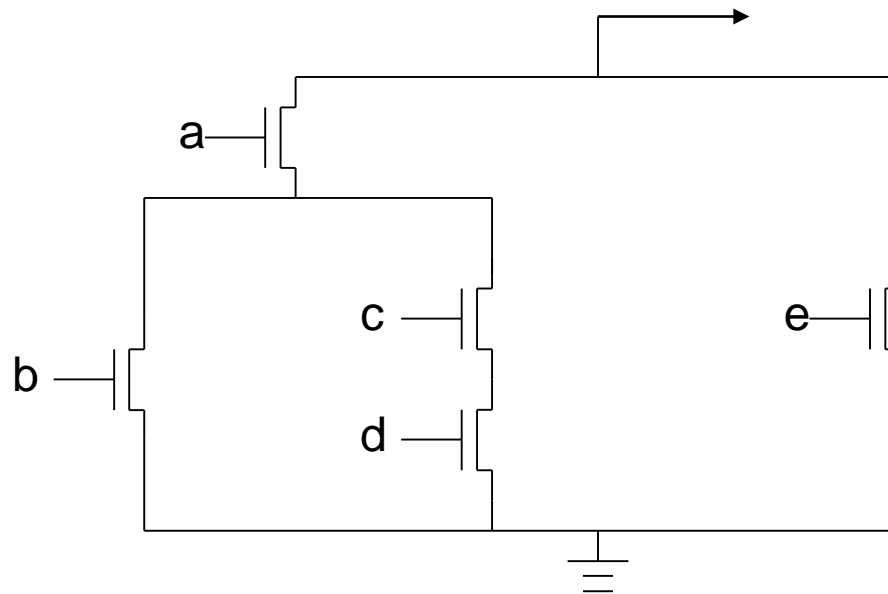


# Παράδειγμα 3 (4 από 4)

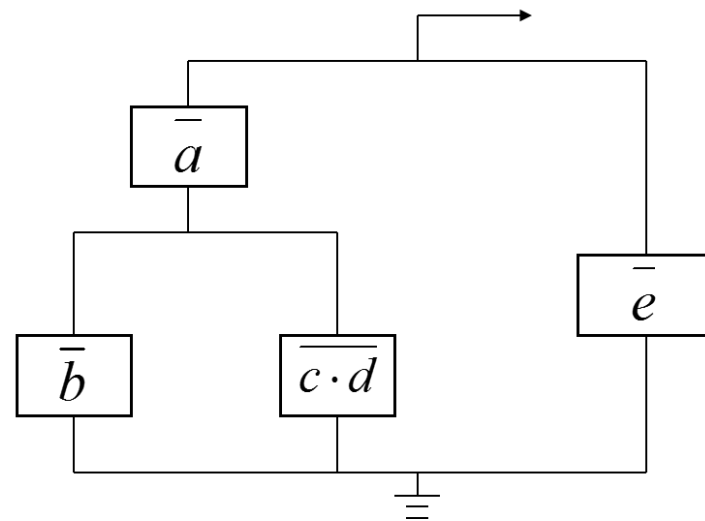
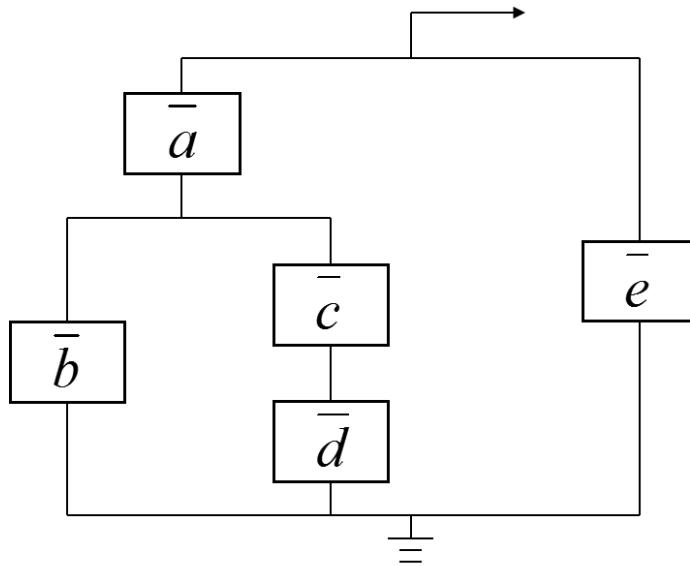


# Εύρεση συνάρτησης από σχέδιο σε επίπεδο τρανζίστορ (1 από 5)

- Σε ποιά συνάρτηση αντιστοιχεί το ακόλουθο n-MOS δικτύωμα;

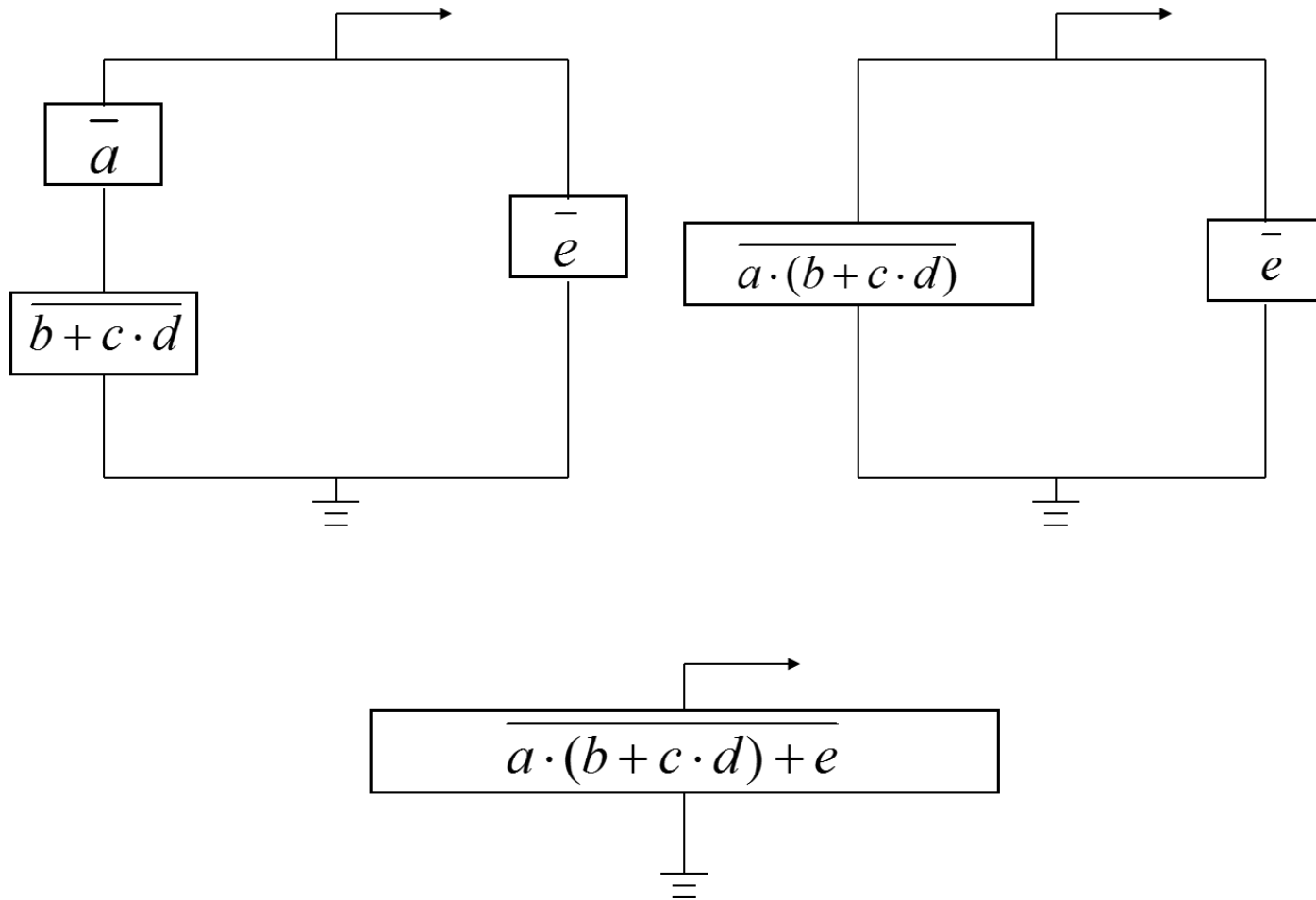


# Εύρεση συνάρτησης από σχέδιο σε επίπεδο τρανζίστορ (2 από 5)





# Εύρεση συνάρτησης από σχέδιο σε επίπεδο τρανζίστορ (3 από 5)

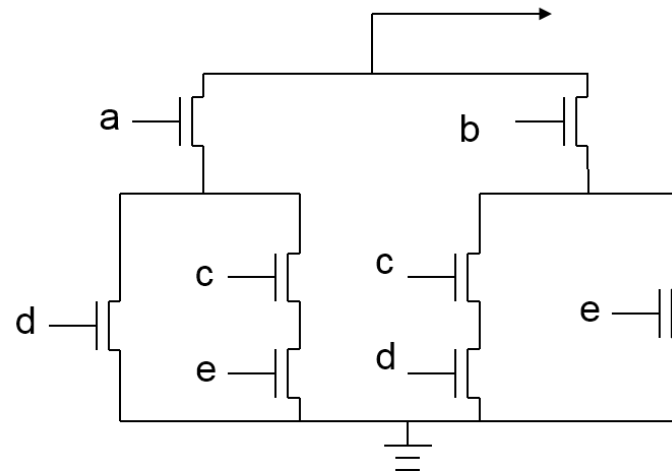
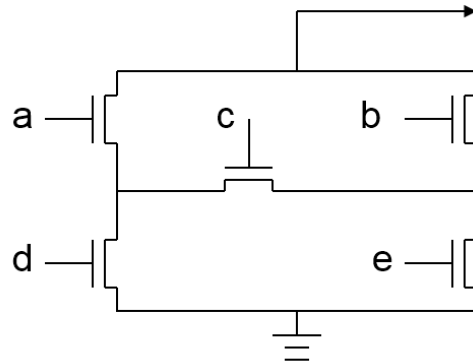


# Εύρεση συνάρτησης από σχέδιο σε επίπεδο τρανζίστορ (4 από 5)

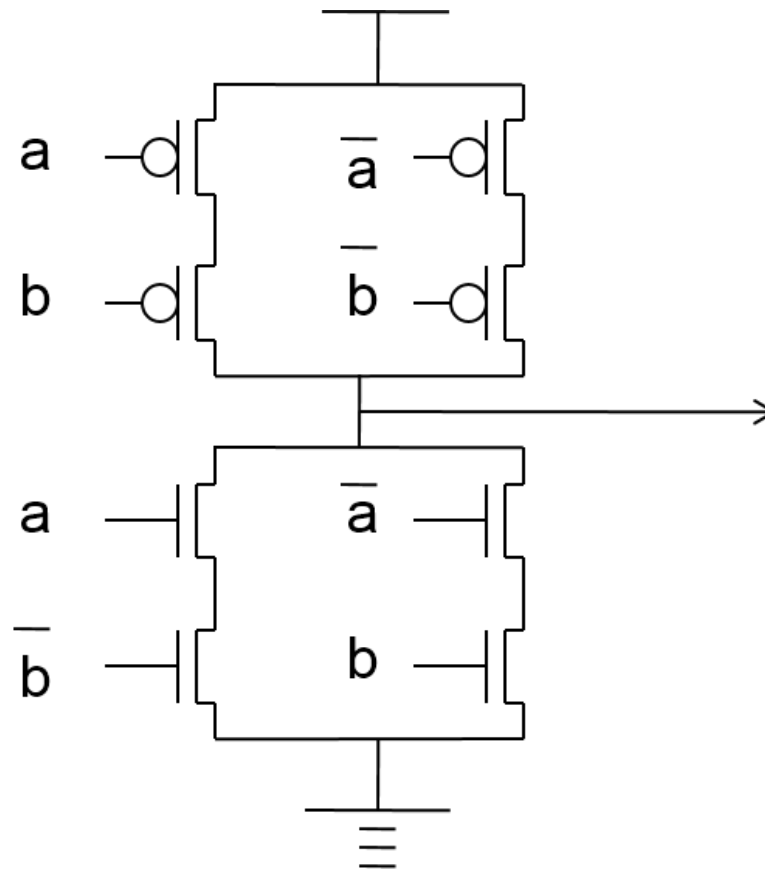
- Οι προηγούμενες μεθοδολογίες ισχύουν για σειριακά-παράλληλα δικτυώματα
- Το ακόλουθο δικτύωμα δεν ανήκει σε αυτή την κατηγορία αλλά ακολουθώντας τα μονοπάτια του (paths) μπορούμε να βρούμε ένα ισοδύναμο κύκλωμα και την συνάρτηση



# Εύρεση συνάρτησης από σχέδιο σε επίπεδο τρανζίστορ (5 από 5)



# Ταυτοποίηση n-MOS και p-MOS δικτυώματος



# Για το κύκλωμα έχω

Από το p-MOS δικτύωμα παίρνω

$$\overline{(a + b) \cdot (\bar{a} + \bar{b})}$$

Από το n-MOS δικτύωμα παίρνω

$$\overline{a \cdot \bar{b} + \bar{a} \cdot b}$$

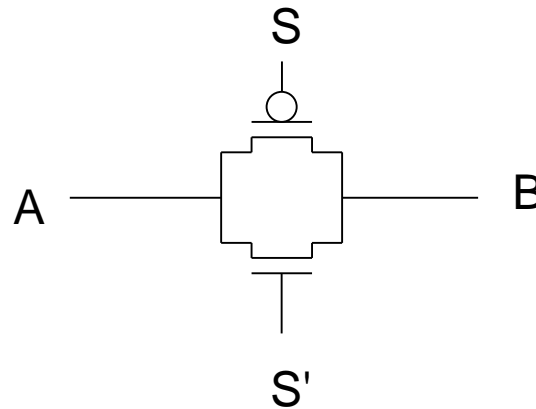
Προσοχή είναι η ίδια συνάρτηση

$$\begin{aligned} \overline{(a + b) \cdot (\bar{a} + \bar{b})} &= \overline{a \cdot \bar{a} + a \cdot \bar{b} + b \cdot \bar{a} + b \cdot \bar{b}} = \\ &= \overline{0 + a \cdot \bar{b} + \bar{a} \cdot b + 0} = \overline{a \cdot \bar{b} + \bar{a} \cdot b} \end{aligned}$$

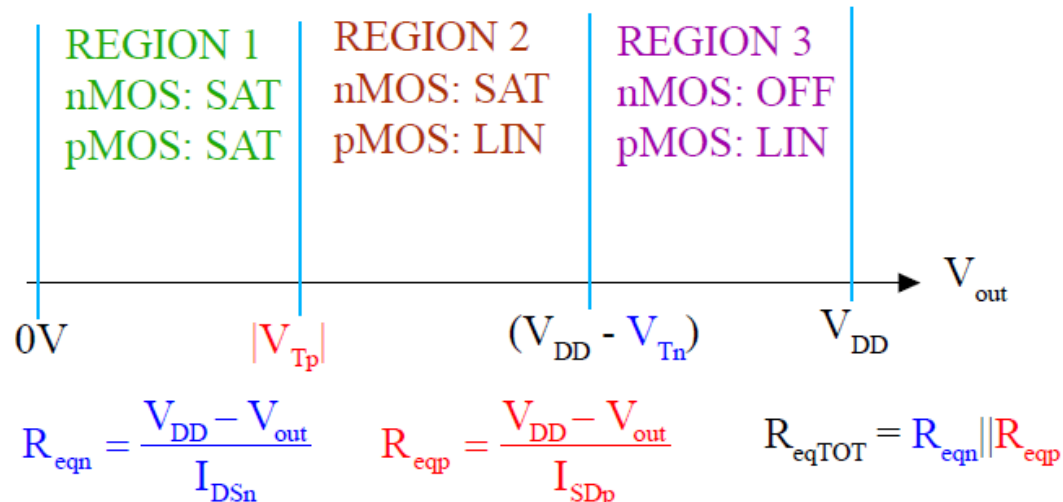
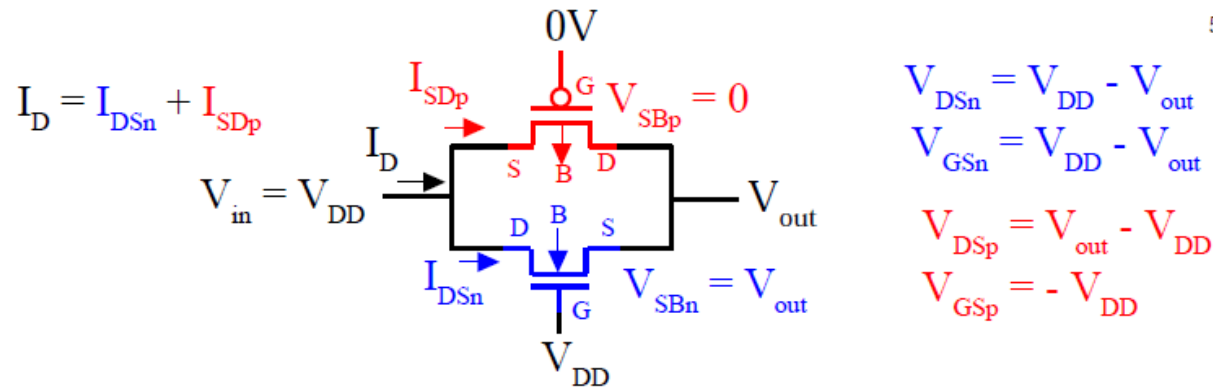


# Λογική τρανζίστορ διέλευσης (pass-logic) (1 από 6)

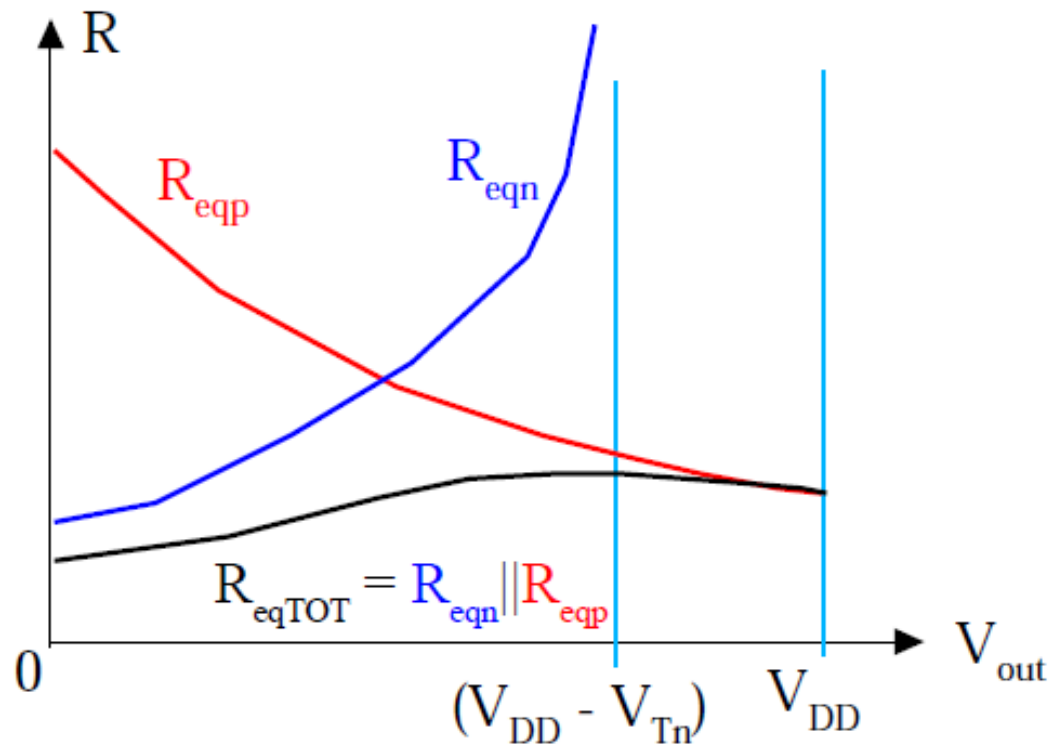
- Το A συνδέεται στο B όταν  $S=0$  (Θα μπορούσα να έχω μόνο το p-MOS ή μόνο το n-MOS αλλά τότε θα υπήρχε πτώση τάσης λόγω τάσης κατωφλίου)



# Λογική τρανζίστορ διέλευσης (pass-logic) (2 από 6)

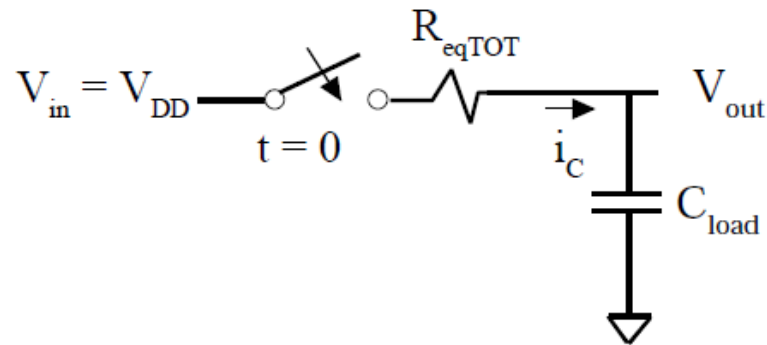
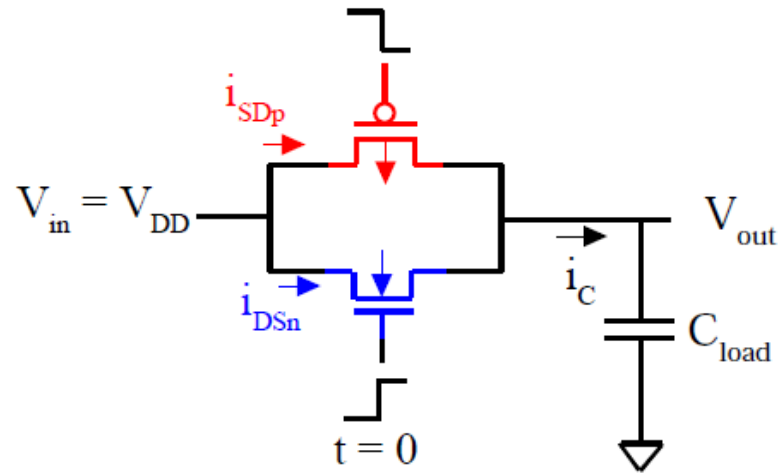


# Λογική τρανζίστορ διέλευσης (pass-logic) (3 από 6)





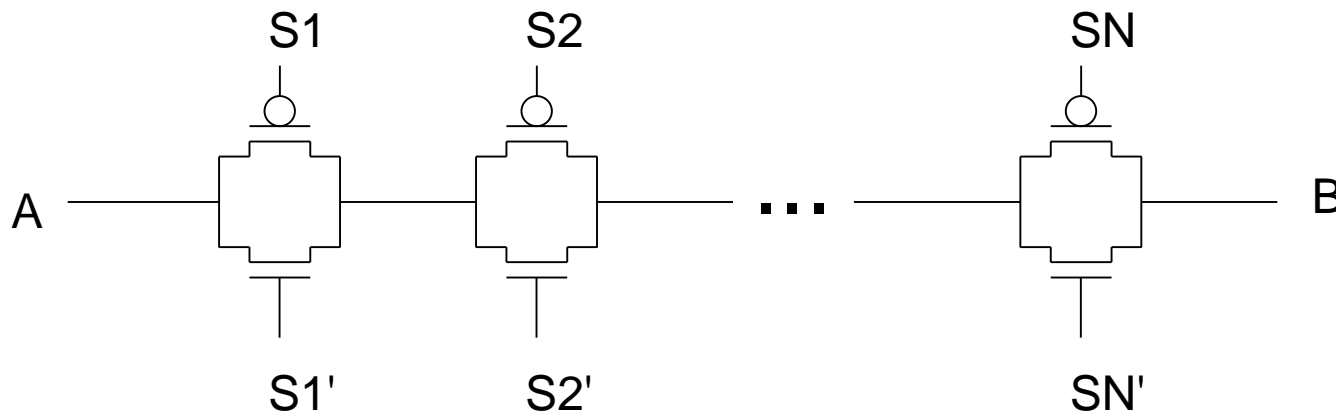
# Λογική τρανζίστορ διέλευσης (pass-logic) (4 από 6)



# Λογική τρανζίστορ διέλευσης (pass-logic) (5 από 6)

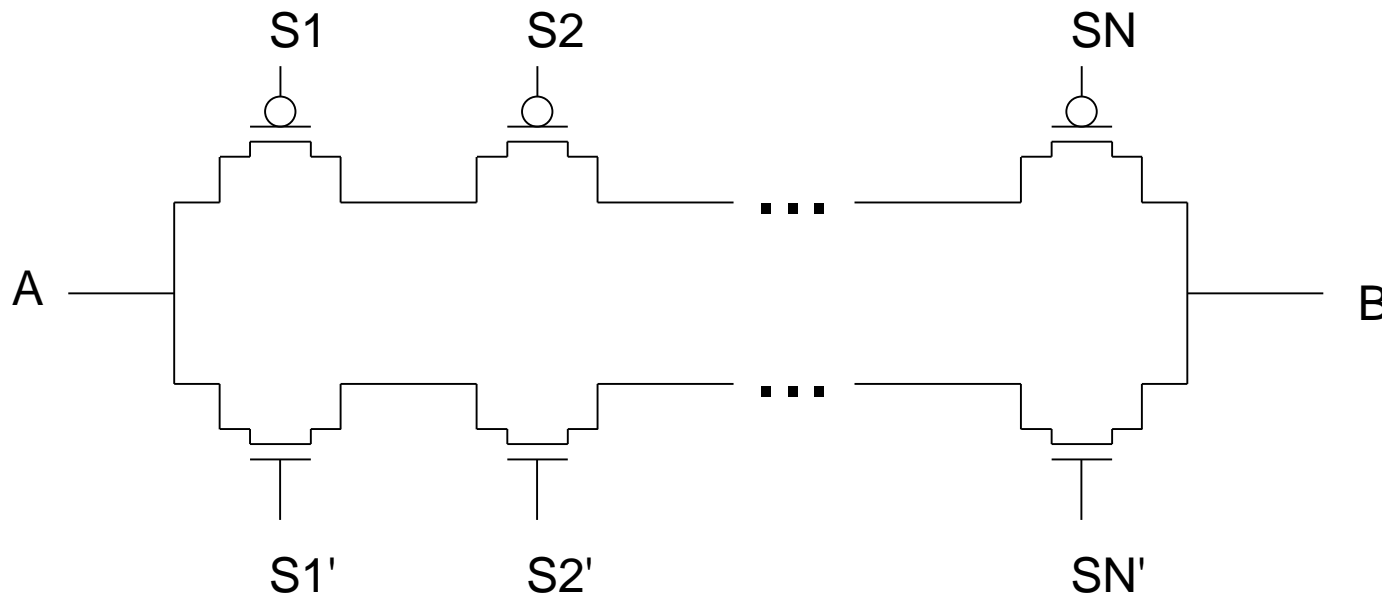
- Στην γενική περίπτωση θα πρέπει

$$S1=S2=\dots=SN=0$$



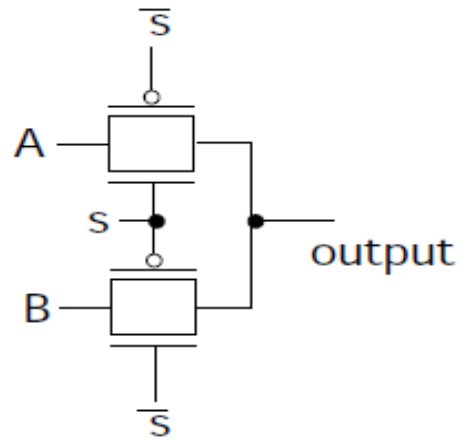
# Λογική τρανζίστορ διέλευσης (pass-logic) (6 από 6)

- Το προηγούμενο κύκλωμα είναι ισοδύναμο με το



# Υλοποίηση Πολυπλέκτη και XOR (1 από 5)

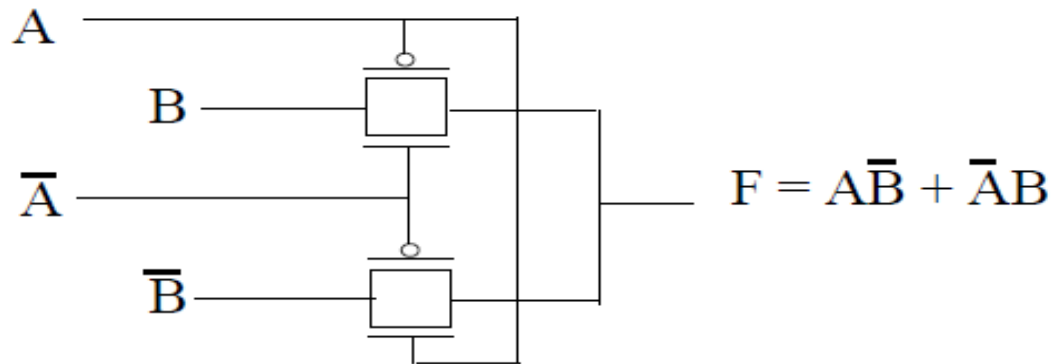
## 2-INPUT MULTIPLEXER



$$\text{output} = A.s + B.\bar{s}$$

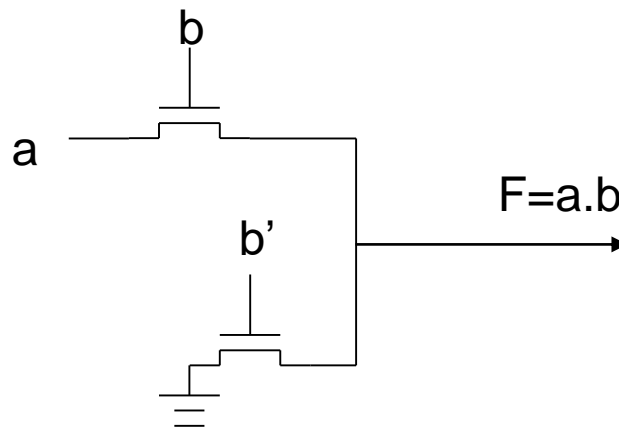
A	B	s	$\bar{s}$	output
x	0	0	1	0 (B)
x	1	0	1	1 (B)
0	x	1	0	0 (A)
1	x	1	0	1 (A)

## XOR



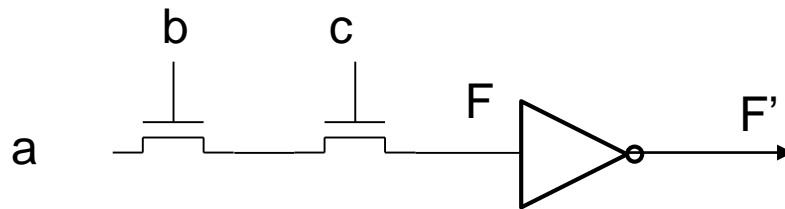
# Υλοποίηση Πολυπλέκτη και XOR (2 από 5)

- Στην γενική περίπτωση για να υλοποιήσω πύλη θα πρέπει για κάθε συνδυασμό εισόδων να συνδέω την έξοδο σε μία τιμή  
(εάν είναι ασύνδετη η τιμή θα είναι hiZ, υψηλή εμπέδηση)



# Υλοποίηση Πολυπλέκτη και XOR (3 από 5)

- Προσοχή η έξοδος μπορεί να επηρεάζει την είσοδο
  - Με χρήση NOT στην έξοδο μπορώ να απομονώσω

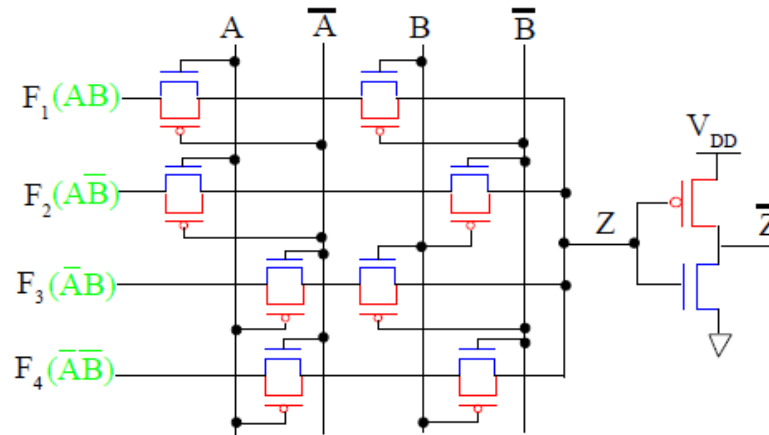


Εάν ένα μονοπάτι μόνο φορτίζει δεν χρειάζονται n-MOS τρανζίστορ

Αντίστοιχα εάν μόνο αποφορτίζει δεν χρειάζονται p-MOS τρανζίστορ



# Υλοποίηση Πολυπλέκτη και XOR (4 από 5)



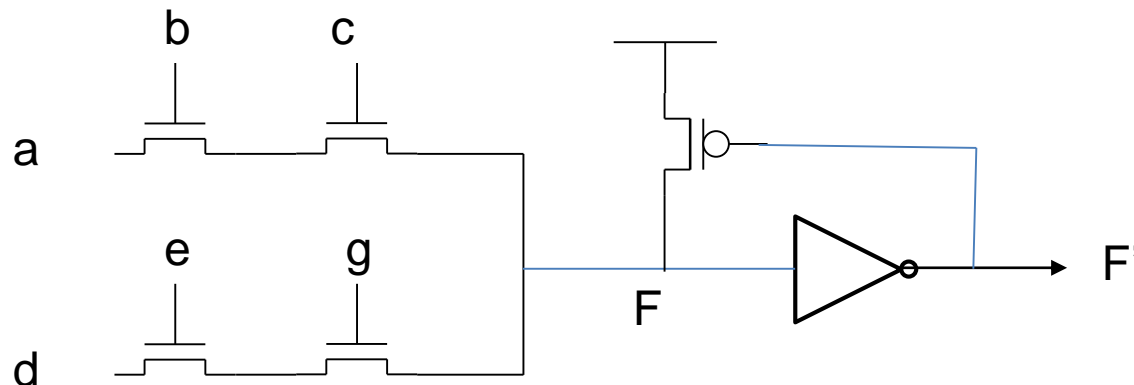
SOME OF THE FUNCTIONS REALIZED BY THE  
BOOLEAN FUNCTION UNIT

OPERATION (Z)	$F_1$	$F_2$	$F_3$	$F_4$
NOR(A,B)	0	0	0	1
XOR(A,B)	0	1	1	0
NAND(A,B)	0	1	1	1
AND(A,B)	1	0	0	0
OR(A,B)	1	1	1	0



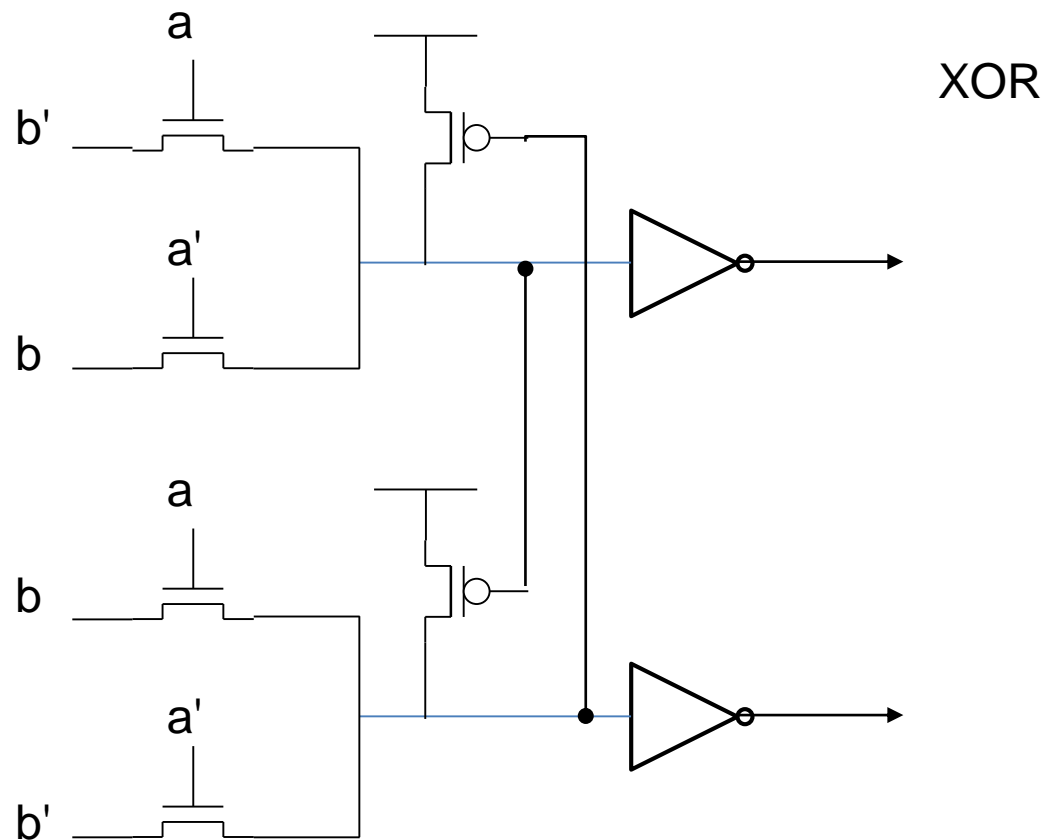
# Υλοποίηση Πολυπλέκτη και XOR (5 από 5)

- Εάν σε μονοπάτι που φορτίζει δεν υπάρχουν τα απαιτούμενα p-MOS τρανζίστορ θα υπάρχει πτώση τάσης στην έξοδο
  - Με p-MOS (ασθενές) που οδηγείται από συμπλήρωμα της εξόδου έχω αποκατάσταση τάσης
- Αντίστοιχα για μονοπάτι που αποφορτίζει χωρίς τα απαιτούμενα n-MOS





# Συμπληρωματική λογική με τρανζίστορ διέλευσης



# Δυναμικά Λογικά Κυκλώματα

**STATIC LOGIC GATES:** valid logic levels are steady-state op points. Outputs are generated in response to input voltage levels after a certain time delay. Output levels are preserved as long as there is power, i.e. no refresh is needed

**DYNAMIC LOGIC GATES:** depends on temporary storage of charge in parasitic node capacitances. Requires periodic updating of internal node voltage levels.

## ADVANTAGES:

1. Allows implementation of simple sequential circuits with memory functions.
2. Use of common clock signals throughout the system enables the synchronization of various circuit blocks
3. Implementation of complex functions generally use less die area than static circuits
4. Often dissipates less dynamic power than static designs, due to smaller parasitic capacitances.



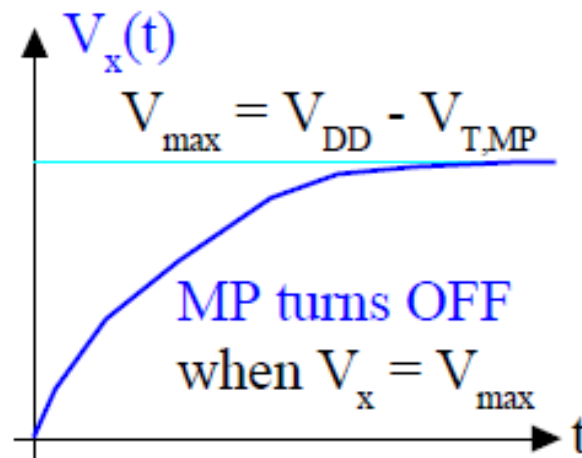
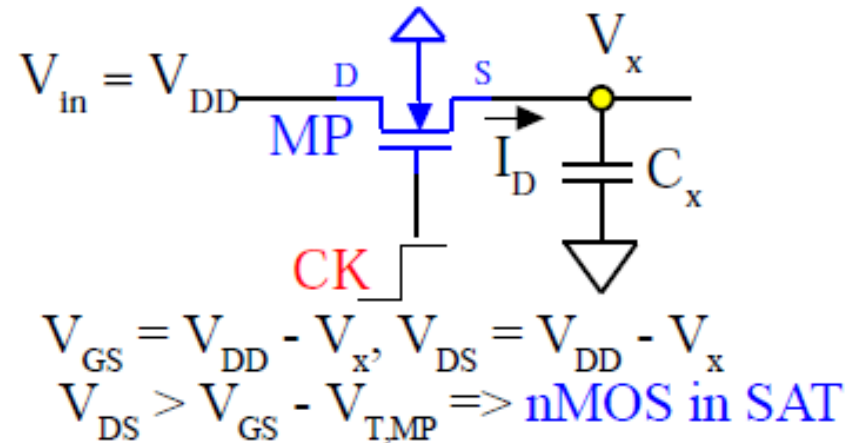
# Logic "1" Transfer

Assume at  $t=0$

$$V_x(t = 0) = 0 V$$

$$V_{in} = V_{OH}$$

$$CK - 0 \rightarrow V_{DD}$$

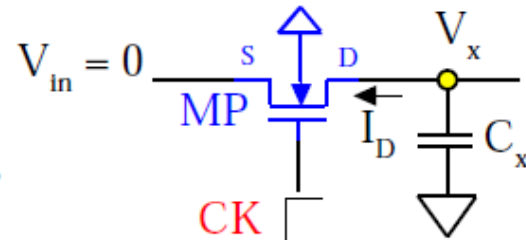


# Logic “0” Transfer

Assume at  $t=0$

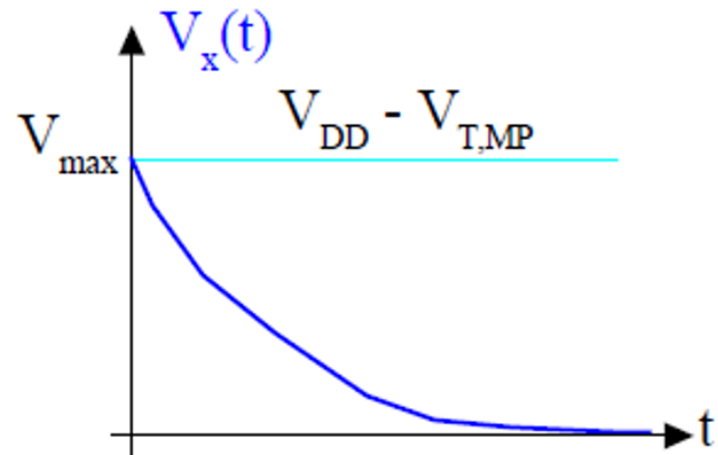
$$V_x(t = 0) = V_{max} = V_{DD} - V_{T,MP}$$

$$CK = 0 \rightarrow V_{DD}$$

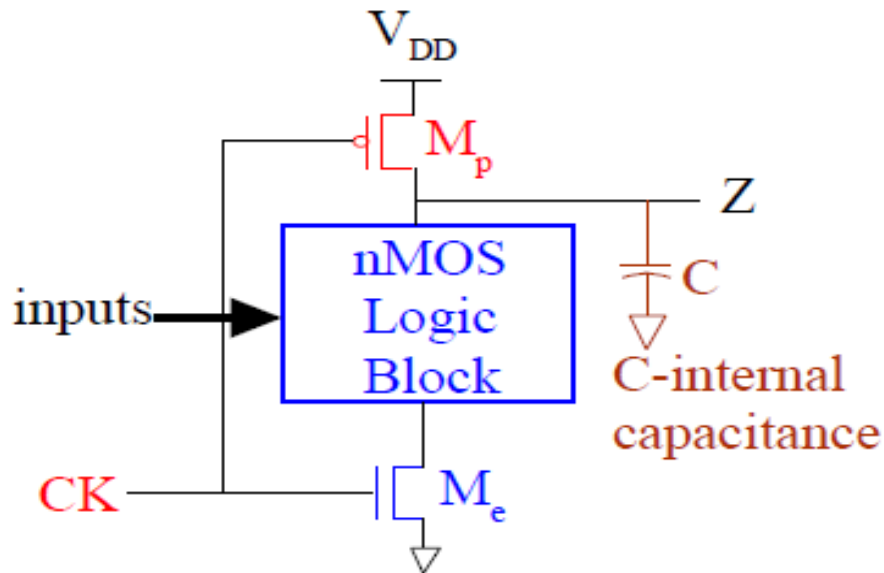


$$V_{GS} = V_{DD}, V_{DS} = V_{max} = V_{DD} - V_{T,MP}$$

$$V_{DS} \leq V_{GS} - V_{T,MP} \Rightarrow \text{nMOS in LIN}$$



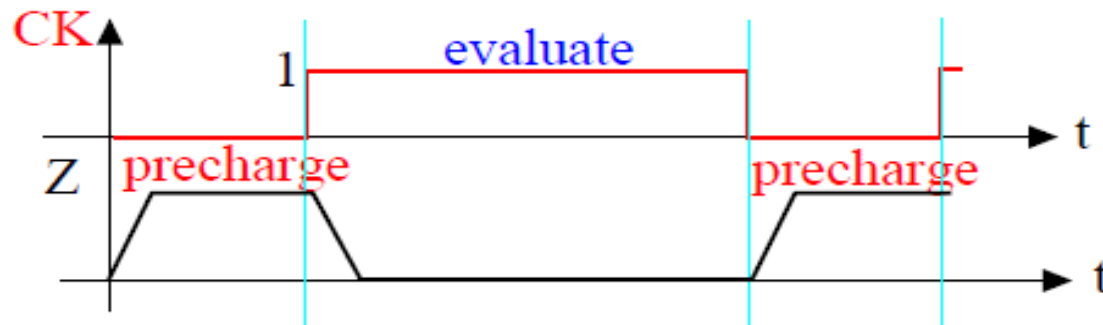
# Δυναμική λογική CMOS δύο φάσεων (1 από 2)



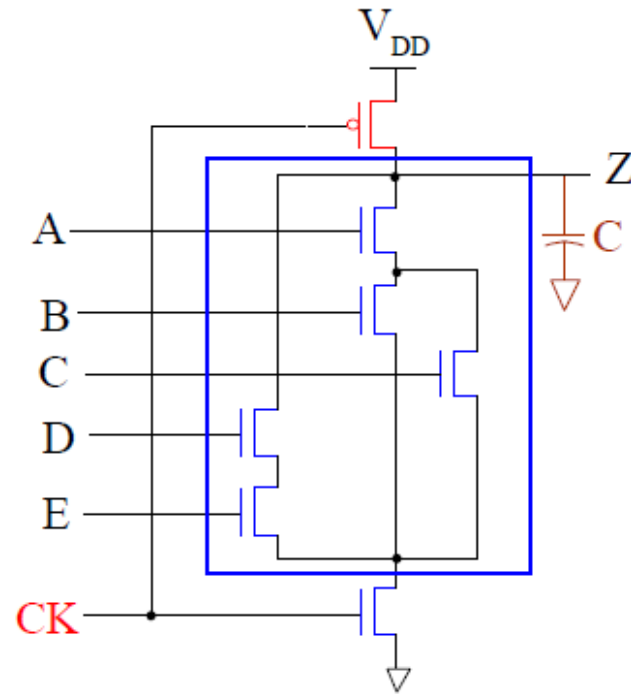
$CK = 0 \rightarrow C$  precharges to  $V_{DD}$   
(output is unavailable during precharge)

$CK = 1 \rightarrow C$  is selectively discharges to 0

(output is only available after discharge is complete)



# Δυναμική λογική CMOS δύο φάσεων (2 από 2)



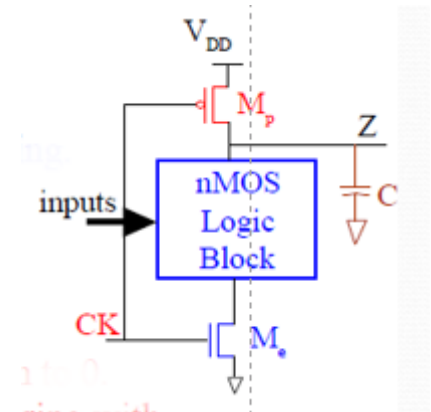
$$Z = A.(B+C)+(D.E) \text{ when } CK = 1$$

$$Z = \text{HIGH} \text{ when } CK = 0$$

# Advantages-Disadvantages

## Advantages

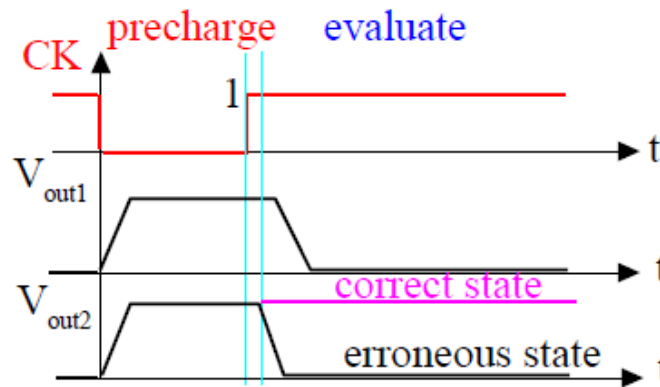
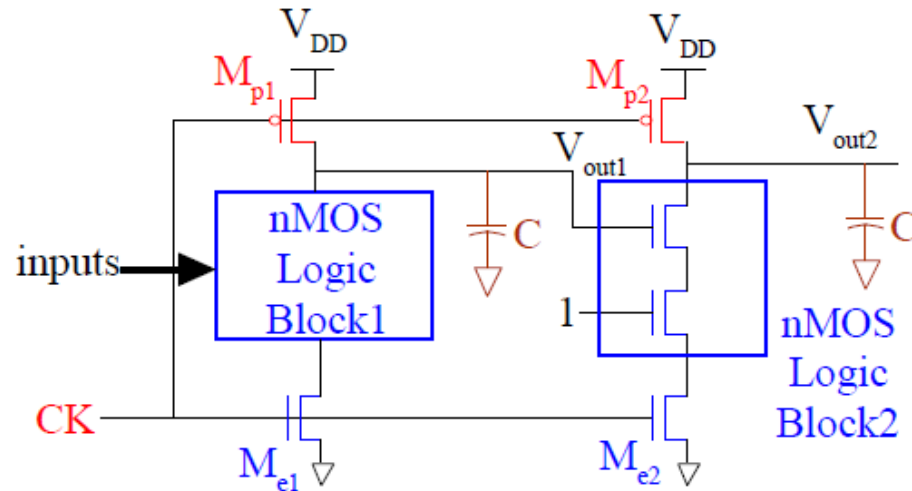
1. Requires  $N+2$  transistors to realize an  $N$ -input gate.
2. Low static power dissipation
3. No dc current paths to place constraints on device sizing
4. Input capacitance same as pseudo nMOS gate.
5. Pull-up time is improved by active switch to  $V_{DD}$ .



## Disadvantages

1. Output is available  $\leq 50\%$  of the time.
2. Pull-down time is degraded due to series active switch to 0
3. Logic output value can be degraded due to charge sharing with other gate capacitances connect to the output.
4. Minimum clock rate determined by leakage on  $C$ .
5. Maximum clock rate determined by circuit delays.
6. Inputs can only change during the precharge phase. Inputs must be stable during evaluation; otherwise an incorrect value of an input could erroneously discharge the output node. (single phase P-E logic gates can not be cascaded)
7. Outputs must be stored during precharge, if they are required during the next evaluate phase.

# Περίπτωση εσφαλμένης απόκρισης



evaluate:  
 $M_{e1}, M_{e2} \rightarrow$  ON  
 $M_{p1}, M_{p2} \rightarrow$  OFF

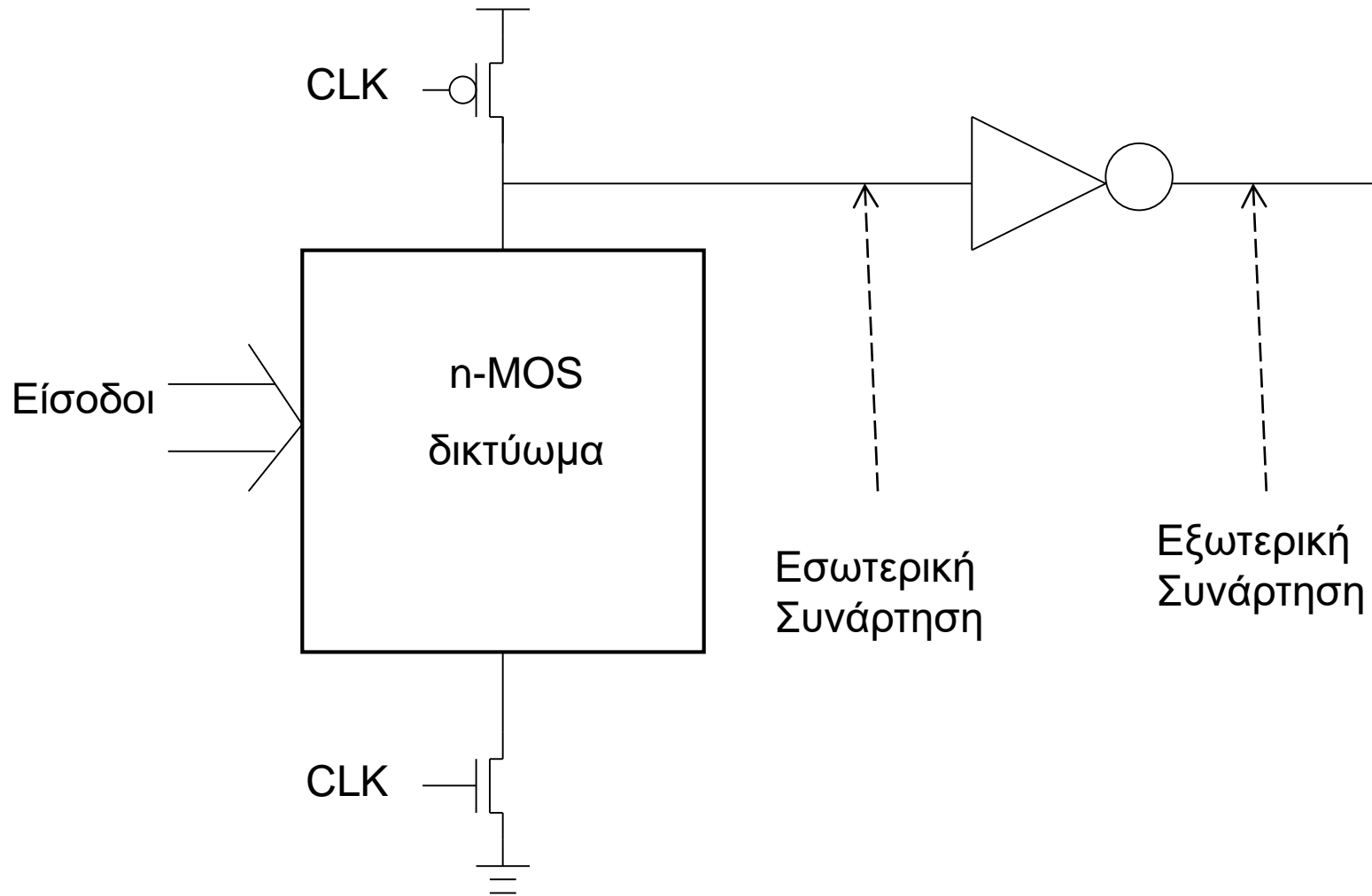


# Domino

- Το κύκλωμα Domino λειτουργεί σε δύο φάσεις
- Την φάση προφόρτισης - precharge
- Την φάση εκτίμησης - evaluation
- Στην φάση προφόρτισης η λειτουργία είναι ανεξάρτητη της λογικής πύλης που υλοποιείται
- Στη φάση εκτίμησης υπολογίζεται τιμή εξόδου που εξαρτάται από την υλοποιούμενη συνάρτηση
  - Προσοχή σε κάθε φάση υπολογισμού μπορούμε να υπολογίσουμε μόνο μία τιμή. Εάν θέλουμε να υπολογίσουμε και νέα τιμή εξόδου για διαφορετικό συνδυασμό θα πρέπει πρώτα να παρεμβάλουμε μία φάση προφόρτισης



# Κύκλωμα Πύλης Domino



# Αρχή Λειτουργίας

- Η εσωτερική συνάρτηση είναι αναστρέφουσα συνάρτηση (inverting function), NAND, NOR, κ.τ.λ.
- Η τελική (εξωτερική συνάρτηση) είναι μη αναστρέφουσα (non-inverting), AND, OR κ.τ.λ.
- Το τμήμα του κυκλώματος που εξαρτάται από τις εισόδους και παράγει την εσωτερική συνάρτηση είναι n-MOS (δεν υπάρχει αντίστοιχο p-MOS)



# Φάση προφόρτισης

- CLK="0"
- Το p-MOS τρανζίστορ που οδηγείται από το CLK φορτίζει τον κόμβο της εσωτερικής συνάρτησης
- Το n-MOS τρανζίστορ που οδηγείται από το CLK εξασφαλίζει ότι δεν υπάρχει μονοπάτι από την τροφοδοσία στη γείωση κατά τη διάρκεια της προφόρτισης ανεξάρτητα από τις τιμές στη είσοδο



# Φάση Εκτίμησης

- CLK="1"
- Το p-MOS τρανζίστορ που οδηγείται από το CLK είναι ανενεργό
- Το n-MOS τρανζίστορ που οδηγείται από το CLK είναι ενεργό
- Εάν υπάρχει ενεργό μονοπάτι στο n-MOS δικτύωμα ο κόμβος της εσωτερικής συνάρτησης θα αποφορτιστεί
  - Προσοχή εάν υπάρξει συνδυασμός εισόδων που αποφορτίζει κανένας μετέπειτα συνδυασμός δεν θα φορτίσει τον κόμβο



# Λογική της Domino Πύλης

- Δεν χρειάζομαι p-MOS δίκτυο, φορτίζω πάντα ανεξάρτητα από την συνάρτηση κατά την προφόρτιση
- Κατά την φάση εκτίμησης εάν η συνάρτηση το απαιτεί αποφορτίζω τον εσωτερικό κόμβο - Εάν όχι έχω κατάσταση hiZ και κρατάω την παλιά τιμή (ό,τι δόθηκε από την προφόρτιση)



# Πλεονεκτήματα/Μειονεκτήματα

- Πλεονεκτήματα
  - ΔΕΝ χρειαζόμαστε p-MOS δικτύωμα
  - Ταχύτερα κυκλώματα
- Μειονεκτήματα
  - Χρειαζόμαστε σήμα χρονισμού
  - Ευαισθησία στο θόρυβο
  - Προβλήματα διαμοίρασης φορτίου (charge sharing)



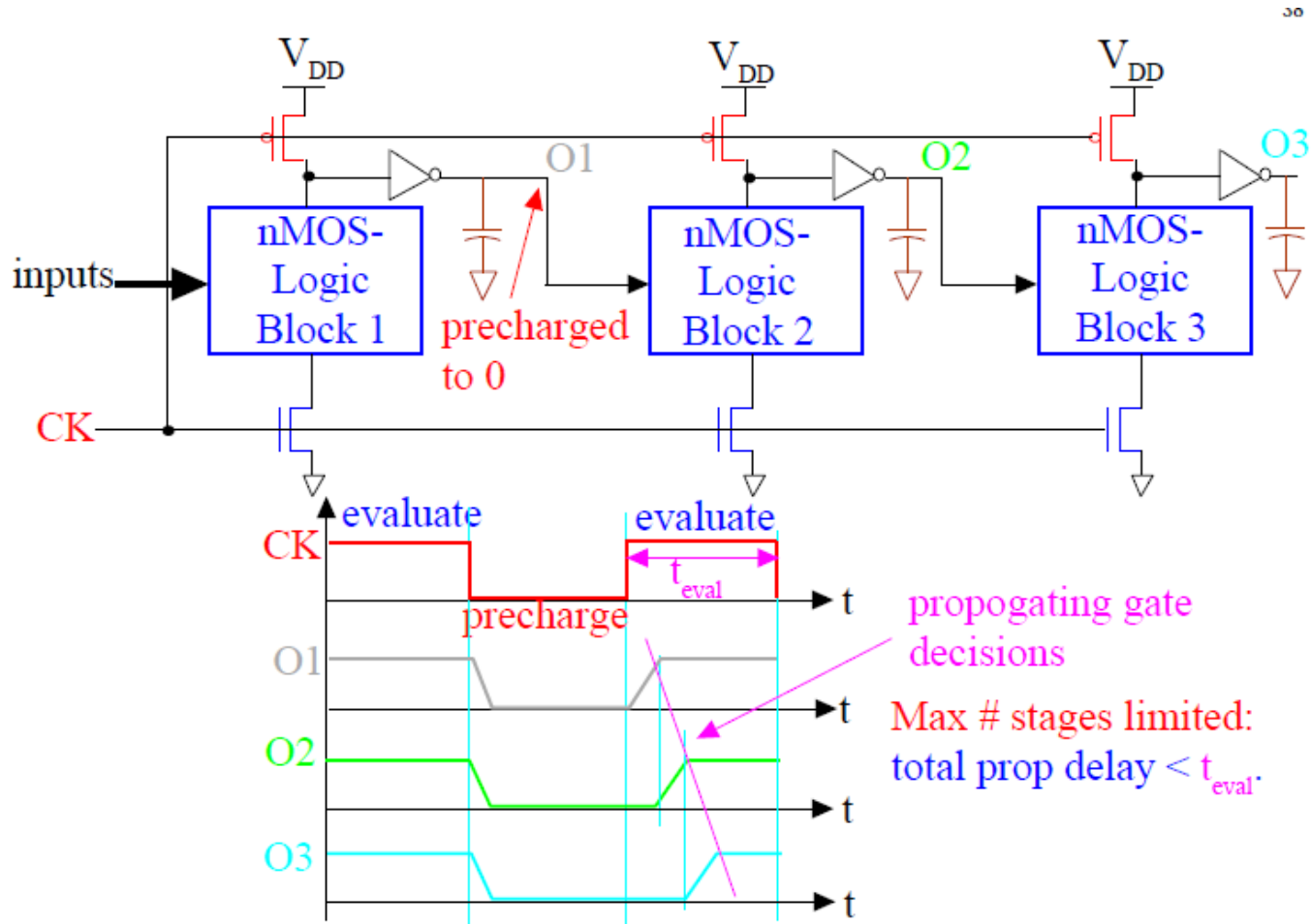
# Γιατί χρειάζεται η NOT(race conditions)

- Εάν δεν θέλω να αποφορτίσω τον εσωτερικό κόμβο δεν πρέπει ποτέ να ενεργοποιήσω το n-MOS δικτύωμα κατά τη φάση εκτίμησης
  - Εάν το ενεργοποιήσω και μετά το απενεργοποιήσω η πύλη θα δει μόνο την ενεργοποίηση
- Κανόνας - Η είσοδος γίνεται λογικό "1" μόνο εάν η τελική της τιμή είναι λογικό "1"
- Χωρίς NOT στο τέλος της προφόρτισης όλες οι έξοδοι θα είναι λογικό "1", χωρίς NOT παραβιάζεται ο κανόνας





# Χρονικοί περιορισμοί σε DOMINO πολλών βαθμίδων



# Σχεδιασμός DOMINO (1 από 2)

- Από την συνάρτηση εξόδου  $F$  γνωρίζω την εσωτερική συνάρτηση  $F'$ . Το n-MOS δικτύωμα της Domino πύλης είναι το n-MOS δικτύωμα της CMOS συνάρτησης  $F'$
- Για παράδειγμα για τη συνάρτηση

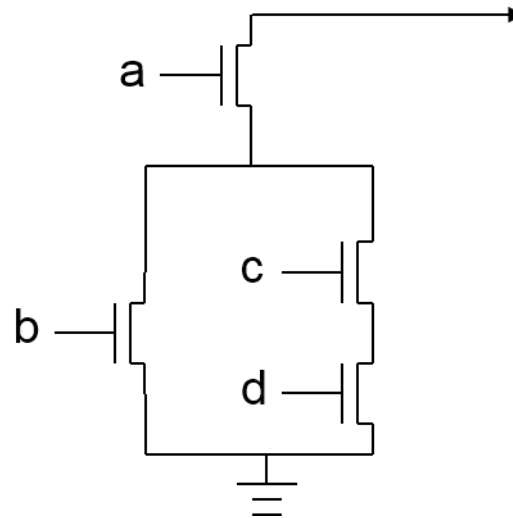
$$F = a \cdot (b + c \cdot d)$$

# Σχεδιασμός DOMINO (2 από 2)

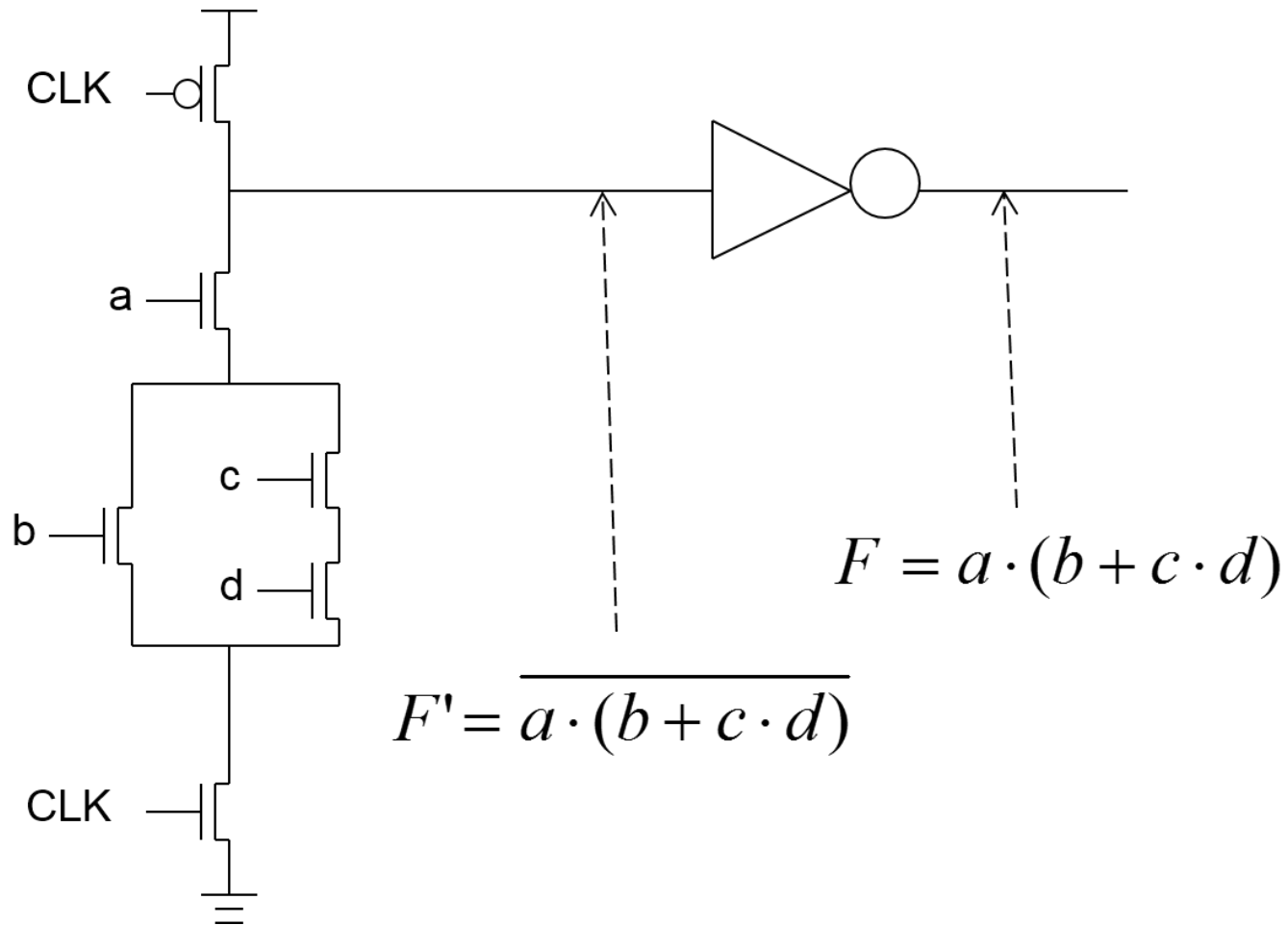
Η  $F'$  είναι

$$F' = \overline{a \cdot (b + c \cdot d)}$$

Με n-δικτύωμα



# Η Domino Πύλη είναι

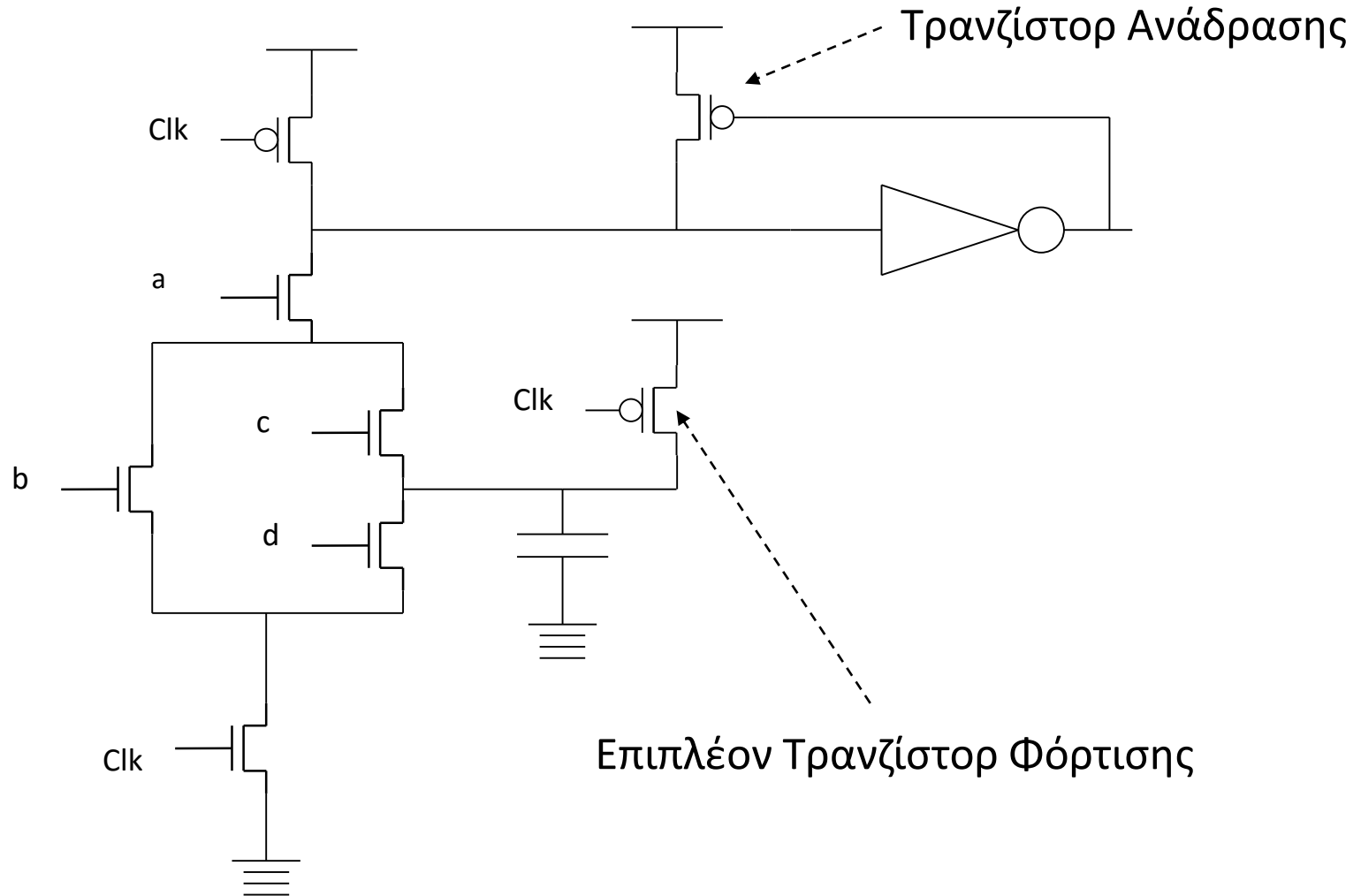


# Διαμοιρασμός φορτίου (1 από 2)

- Ένα πρόβλημα που αντιμετωπίζουν τα DOMINO κυκλώματα είναι ο διαμοιρασμός φορτίου
- Για παράδειγμα στην προηγούμενη πύλη ας θεωρήσουμε πυκνωτή μεταξύ των τρανζίστορ που οδηγούνται από τις πύλες c και d
- Με είσοδο  $abcd=0001$  πυκνωτής αποφορτίζεται και εάν η επόμενη είσοδος είναι  $abcd=1010$  το φορτίο στον κόμβο της εσωτερικής συνάρτησης διαμοιράζεται και άρα έχουμε πτώση τάσης χωρίς να υπάρχει μονοπάτι αποφόρτισης
- Η πτώση τάσης μπορεί να οδηγήσει σε αλλαγή λογικής τιμής
- Μπορεί να αντιμετωπιστεί είτε με ανάδραση από την έξοδο είτε με επιπλέον τρανζίστορ φόρτισης



# Διαμοιρασμός φορτίου (2 από 2)

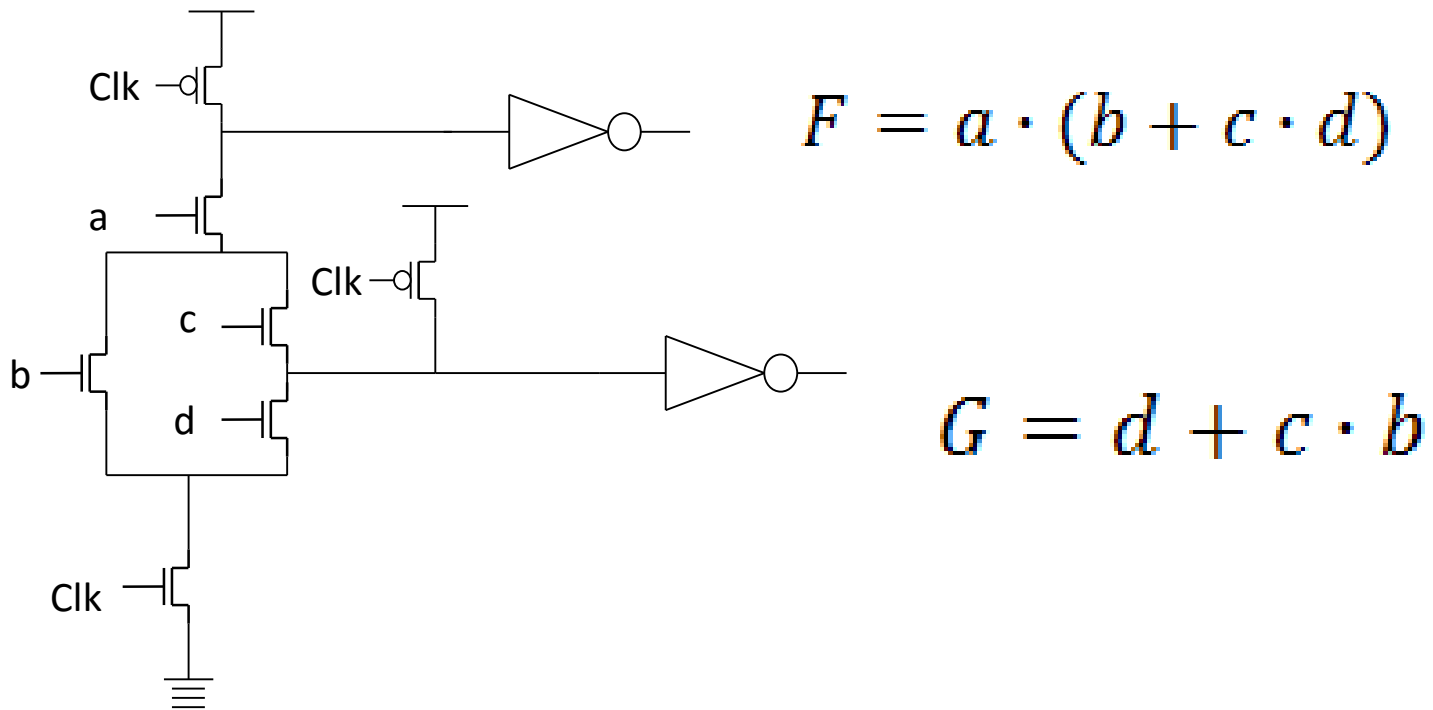


# Domino Πολλαπλών εξόδων (Multiple-Output Domino) (1 από 2)

- Μπορούμε να υλοποιούμε από κοινό δικτύωμα περισσότερες από μία Domino λογικές πύλες.
- Για παράδειγμα εάν χρησιμοποιήσουμε το κόμβο μεταξύ των τρανζίστορ που οδηγούνται από τις πύλες c και d υπολοποιούμε την συνάρτηση  $G=d+cb$



# Domino Πολλαπλών Εξόδων (Multiple-Output Domino) (2 από 2)

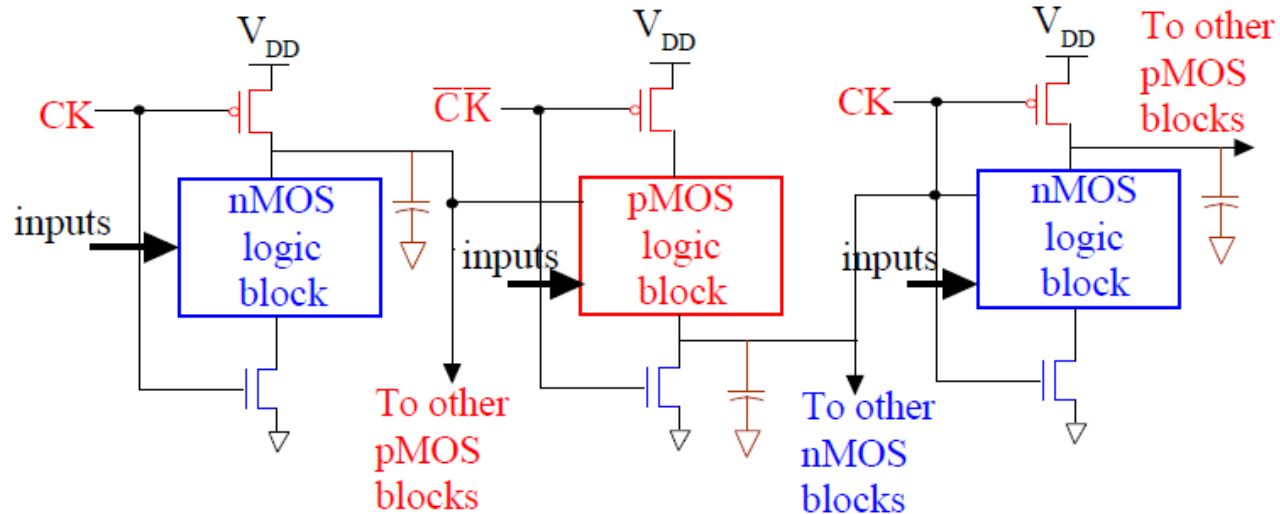




# NP DOMINO LOGIC (1 από 2)

45

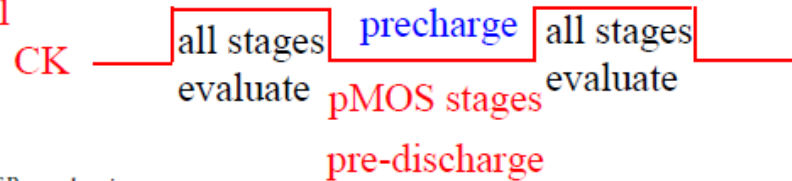
## NP DOMINO LOGIC (NORA or ZIPPER CMOS)



NOTE: INVERTERS ARE NOT REQUIRED AT OUTPUTS OF STAGES

ALL inputs stable  
when CK = 1

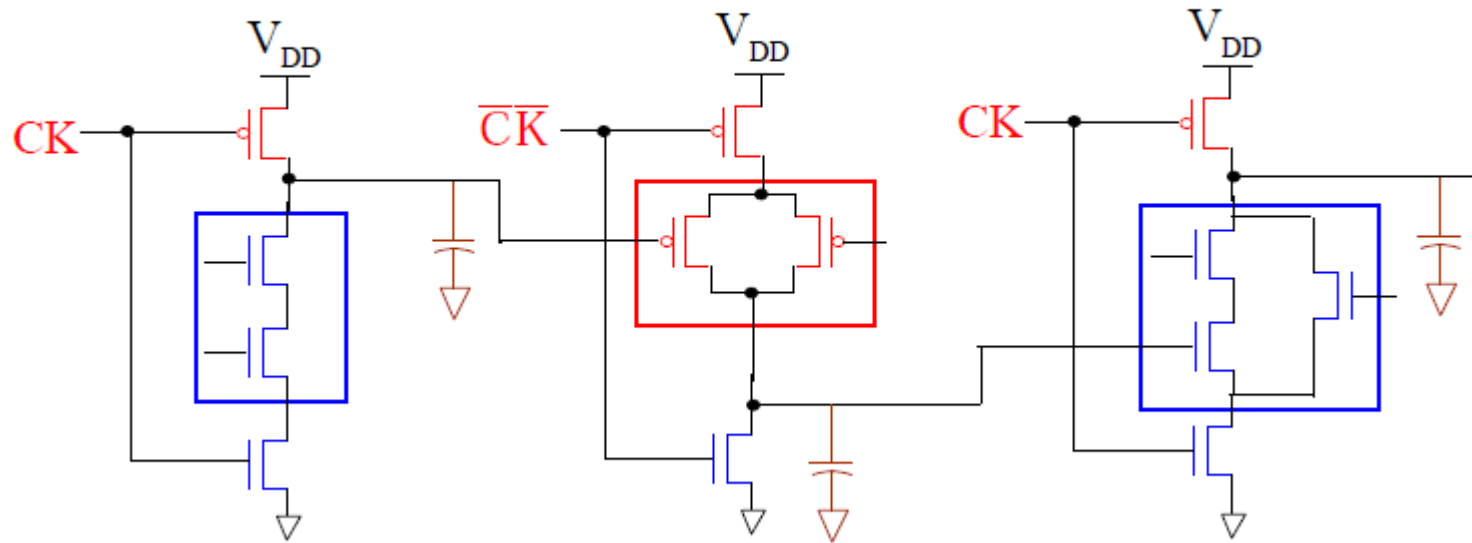
nMOS stages



Kenneth R. Laker, University of Pennsylvania

# NP DOMINO LOGIC (2 από 2)

## NP DOMINO LOGIC (NORA or ZIPPER CMOS) EXAMPLE



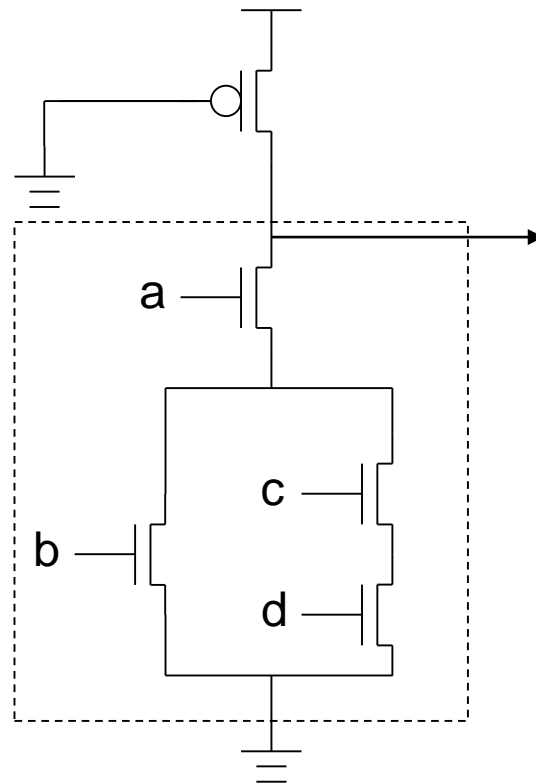
# Ψευδο-nMOS πύλες ( pseudo-nMOS)

- Κρατάω μόνο τον n-Δικτύωμα
- Το p-MOS δικτύωμα αντικαθίσταται από ένα p-MOS τρανζίστορ που λειτουργεί σαν αντίσταση
  - Η πύλη στη γείωση
  - Συνήθως μεγαλύτερο μήκος από πλάτος για περιορισμό του ρεύματος



# Παράδειγμα

$$F = \overline{a \cdot (b + c \cdot d)}$$



## DCVS (1 από 3)

Εάν υπάρχει η συμπληρωματική συνάρτηση το p-MOS δικτύωμα μπορεί να αντικατασταθεί από ένα p-MOS τρανζίστορ που οδηγείται από την συμπληρωματική

Για παράδειγμα η συνάρτηση

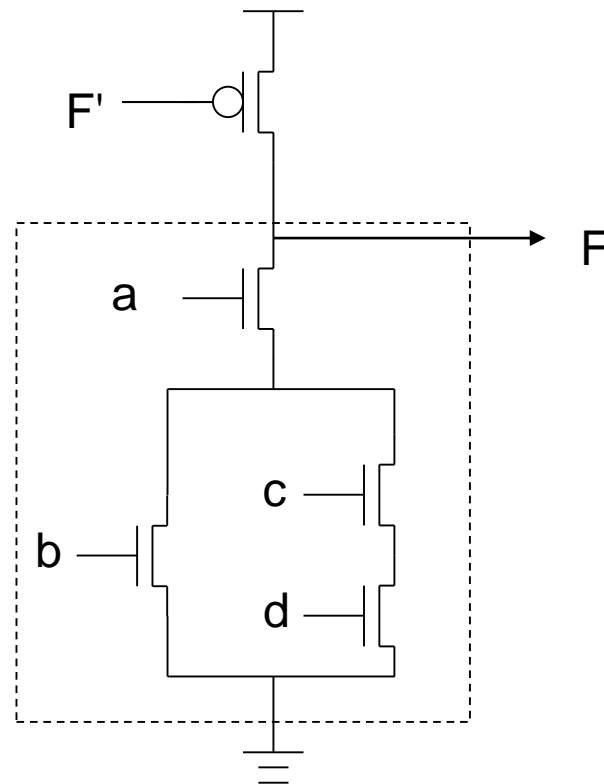
$$F = \overline{a \cdot (b + c \cdot d)}$$

εάν υπάρχει η

$$F' = a \cdot (b + c \cdot d)$$

# DCVS (2 από 3)

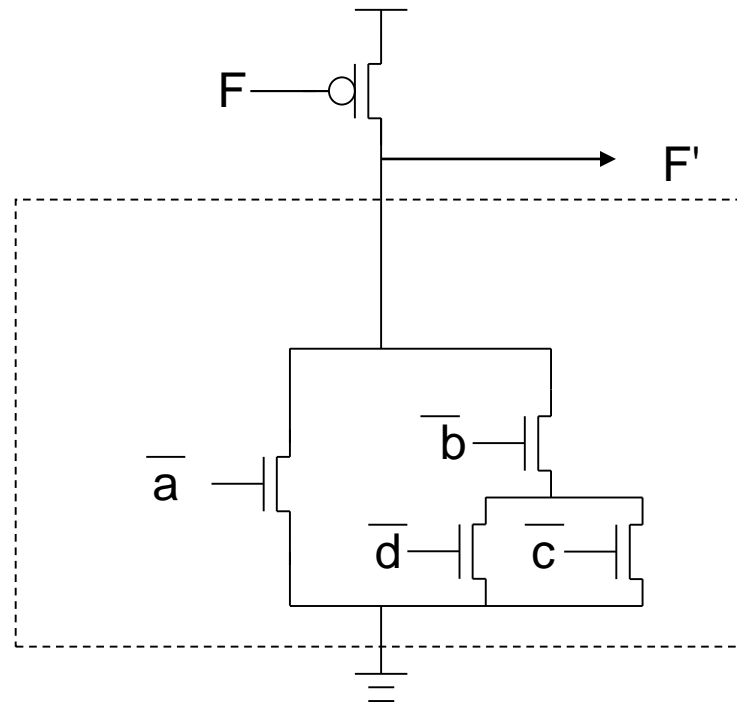
Μπορεί να υλοποιηθεί ως



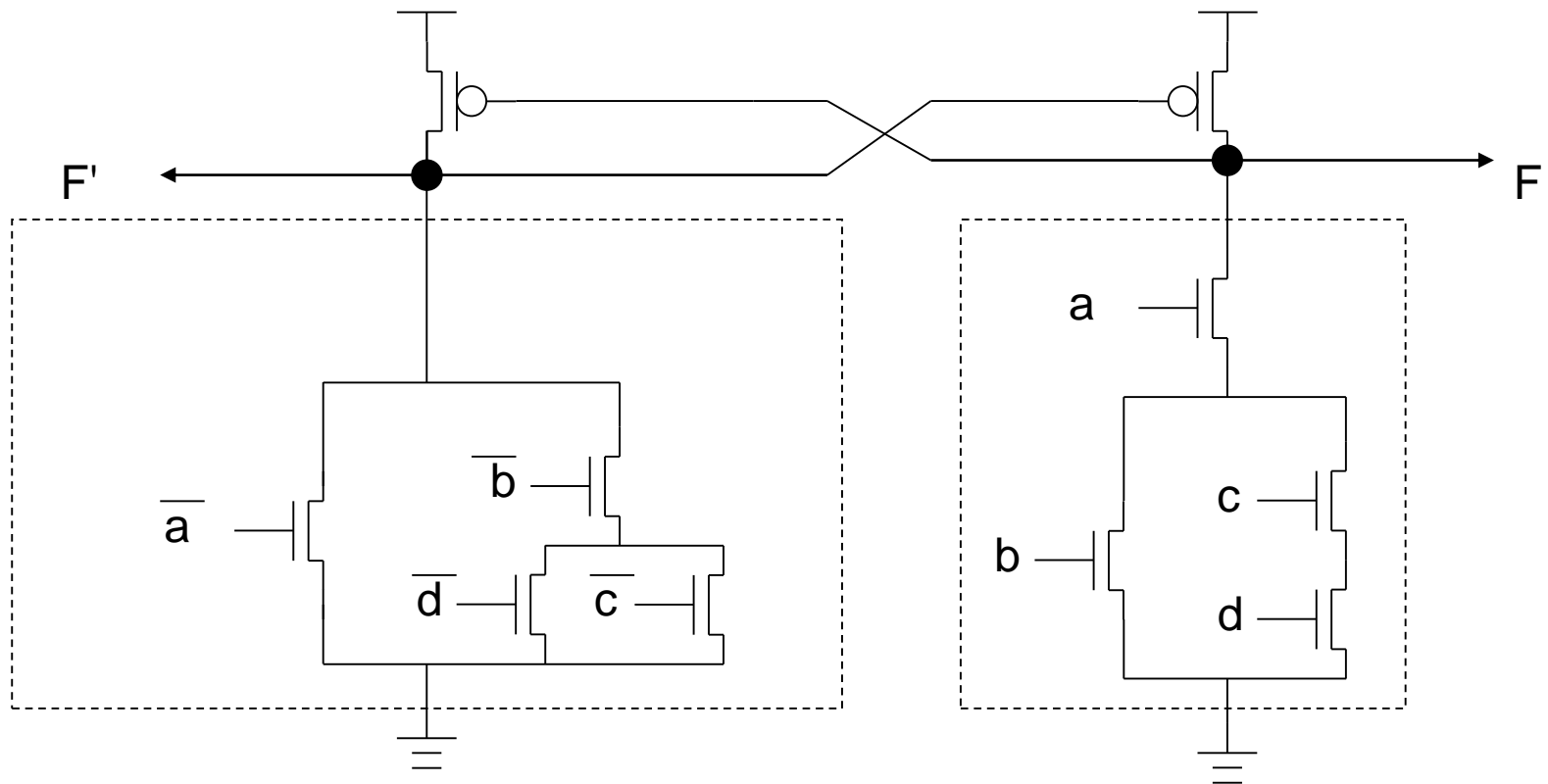
# DCVS (3 από 3)

Κατά αντίστοιχο τρόπο μπορεί να υλοποιηθεί η  $F'$  εάν υπάρχει η  $F$

$$F' = a \cdot (b + c \cdot d) = \overline{\overline{a} + \overline{b} \cdot (\overline{c} + \overline{d})}$$



# Ταυτόχρονη Υλοποίηση (1 από 2)





# Ταυτόχρονη Υλοποίηση (2 από 2)

- Το n-δικτύωμα της  $F'$  είναι το ίδιο με το p-δικτύωμα της  $F$ .

Τα p-MOS τρανζίστορς έχουν αντικατασταθεί από n-MOS και οι είσοδοι από τα συμπληρώματα τους.



# Τέλος Ενότητας



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Σημειώματα

# Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0.



# Σημείωμα Αναφοράς

Copyright Εθνικών και Καποδιστριακών Πανεπιστημίων Αθηνών, Αραπογιάννη Αγγελική 2015. «Σχεδίαση CMOS Ψηφιακών Ολοκληρωμένων Κυκλωμάτων. Σχεδιασμός Κυκλωμάτων σε Επίπεδο Τρανζίστορ.». Έκδοση: 1.0. Αθήνα 2015. Διαθέσιμο από τη δικτυακή διεύθυνση:  
<http://opencourses.uoa.gr/courses/DI102/>.



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.



# Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.





# Σημείωμα Χρήσης Έργων Τρίτων

Το Έργο αυτό κάνει χρήση των ακόλουθων έργων:

Οι εικόνες και τα διαγράμματα που χρησιμοποιούνται είναι από το βιβλίο:

Sung-Mo Kang, Yusuf Leblebici. 1996. *CMOS Digital Integrated Circuits* (1 ed.). McGraw-Hill, Inc., New York, NY, USA © 1996.

