

Σχεδιασμός Κυκλωμάτων σε Επίπεδο Τρανζίστορ

Βασικές CMOS Λογικές οικογένειες (CMOS και Domino)

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 έξοδοι 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 εισοδοι 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)

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

- 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 $(F+G)'$ πρέπει να άγει εάν και το n-δικτύωμα της F' και το n-δικτύωμα G' άγει

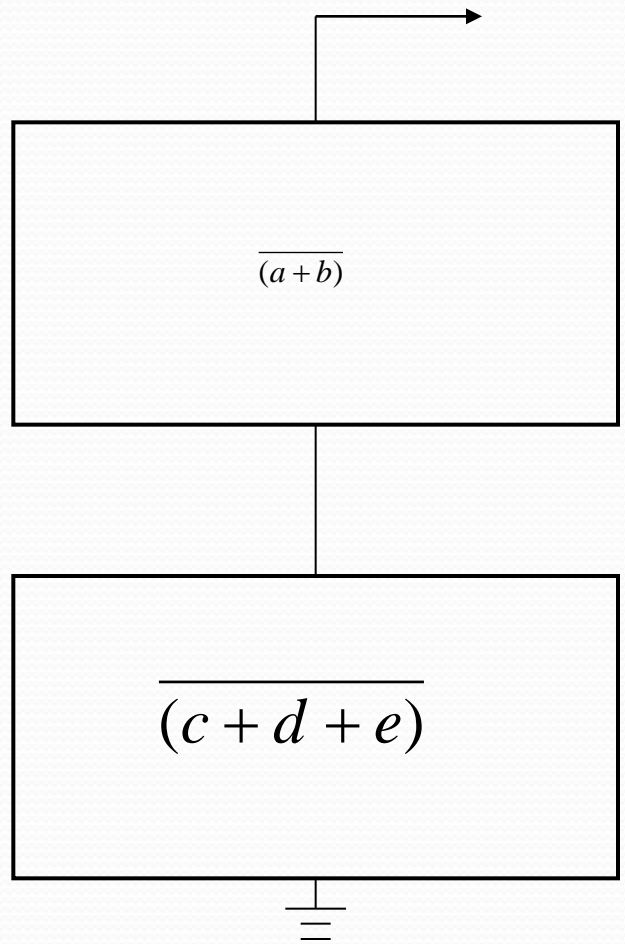
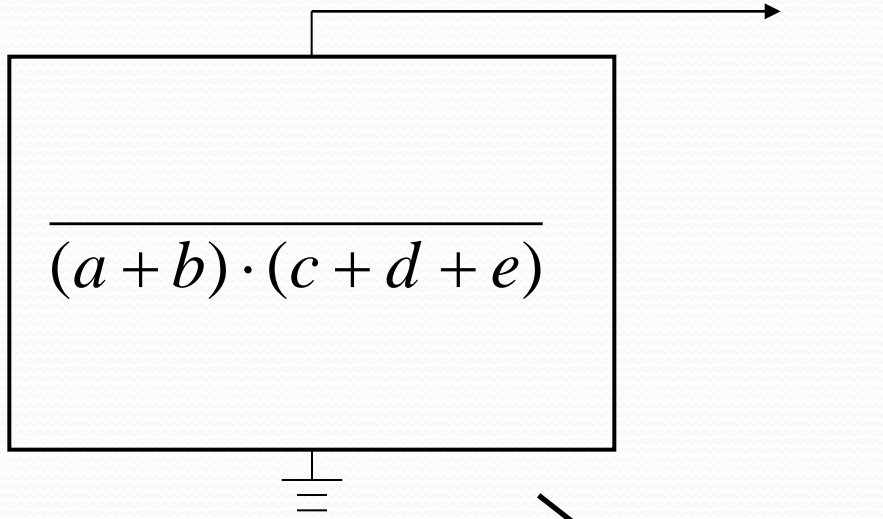
Κανόνες για το ρ -δικτύωμα

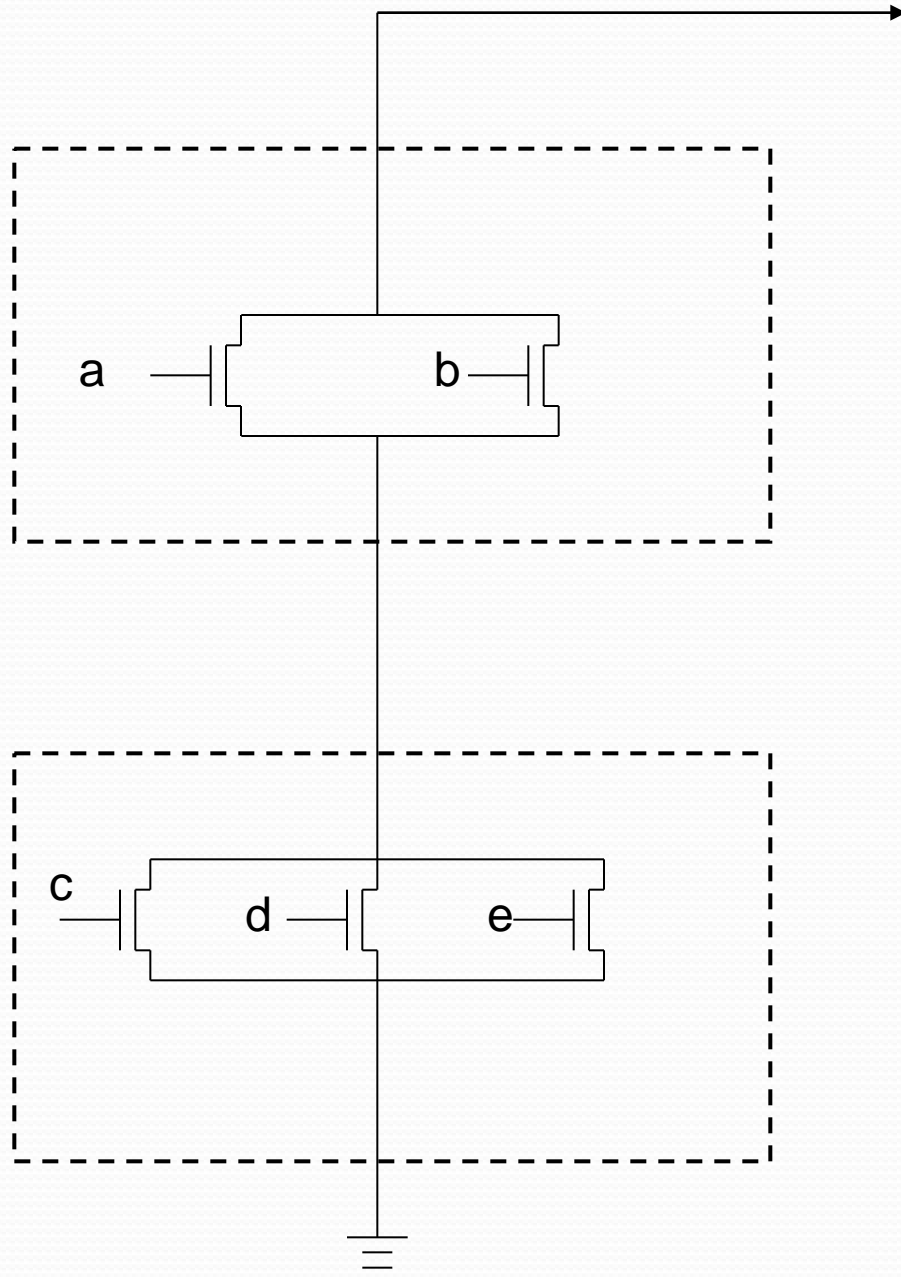
- Έστω ότι έχω τα ρ -δικτυώματα των συναρτήσεων F' και G'
- Το δικτύωμα της $(F+G)'$ παράγεται εάν τοποθετήσω σε σειρά τα υπάρχοντα δικτυώματα
 - Η $(F+G)'$ θα είναι λογικό "1" εάν και η F' και η G' είναι λογικό "1". Επομένως το ρ -δικτύωμα της $(F+G)'$ πρέπει να άγει εάν και το ρ -δικτύωμα της F' και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

Παράδειγμα

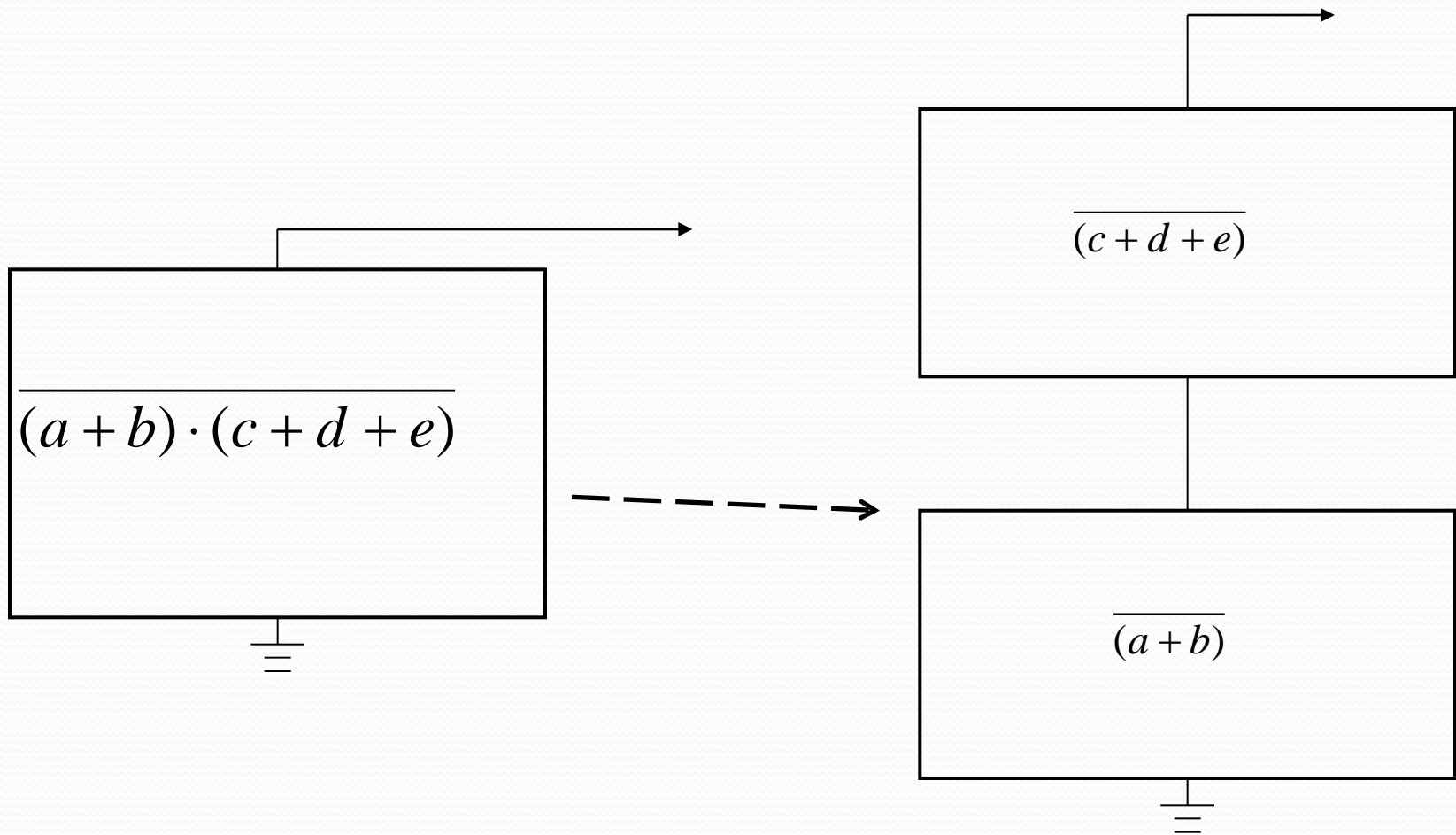
- Υλοποιήστε το n-δικτύωμα της ακόλουθης συνάρτησης

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



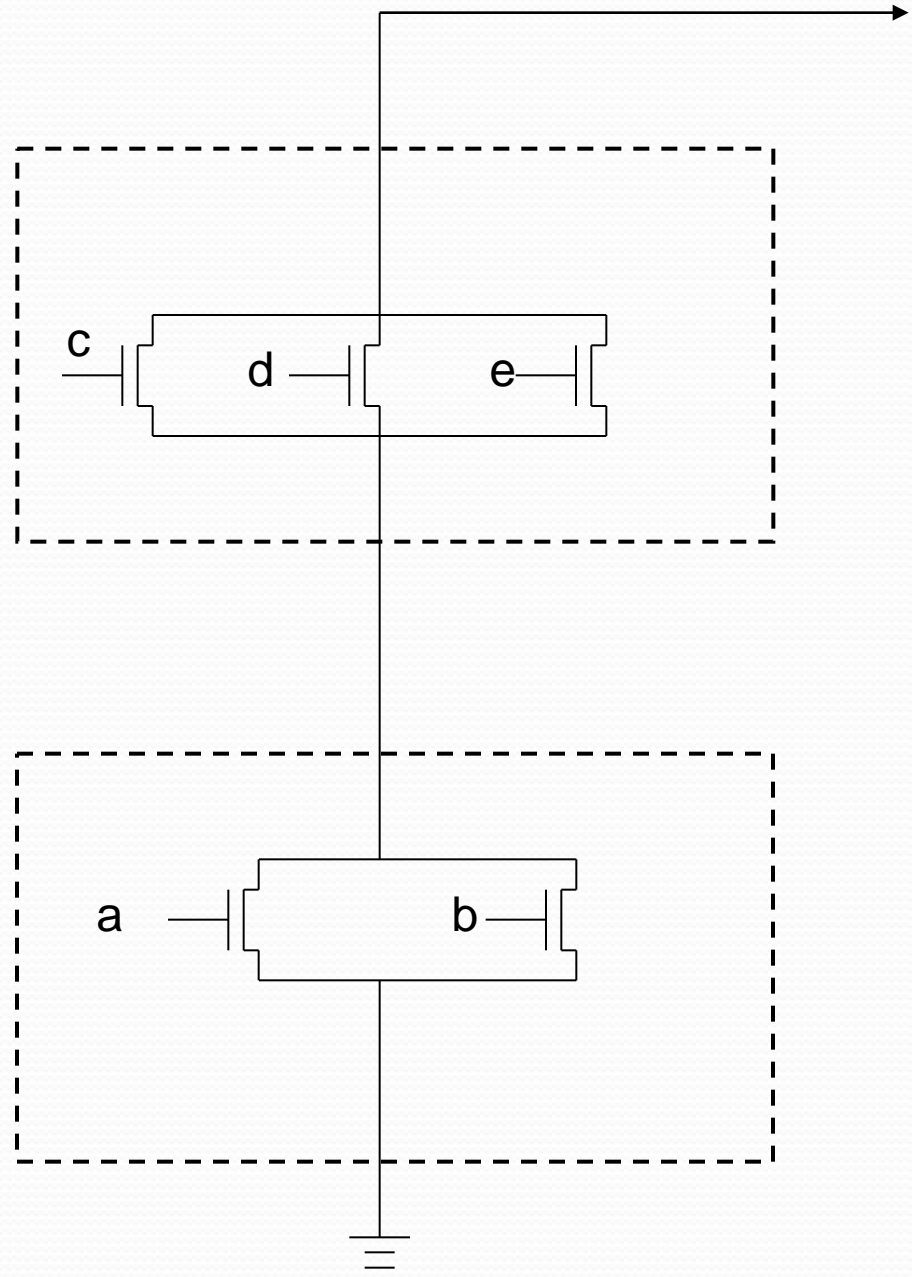


Εναλλακτική υλοποίηση



• Διαφέρει

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

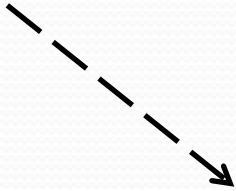


- Υλοποιήστε το ρ -δικτύωμα της ακόλουθης συνάρτησης

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

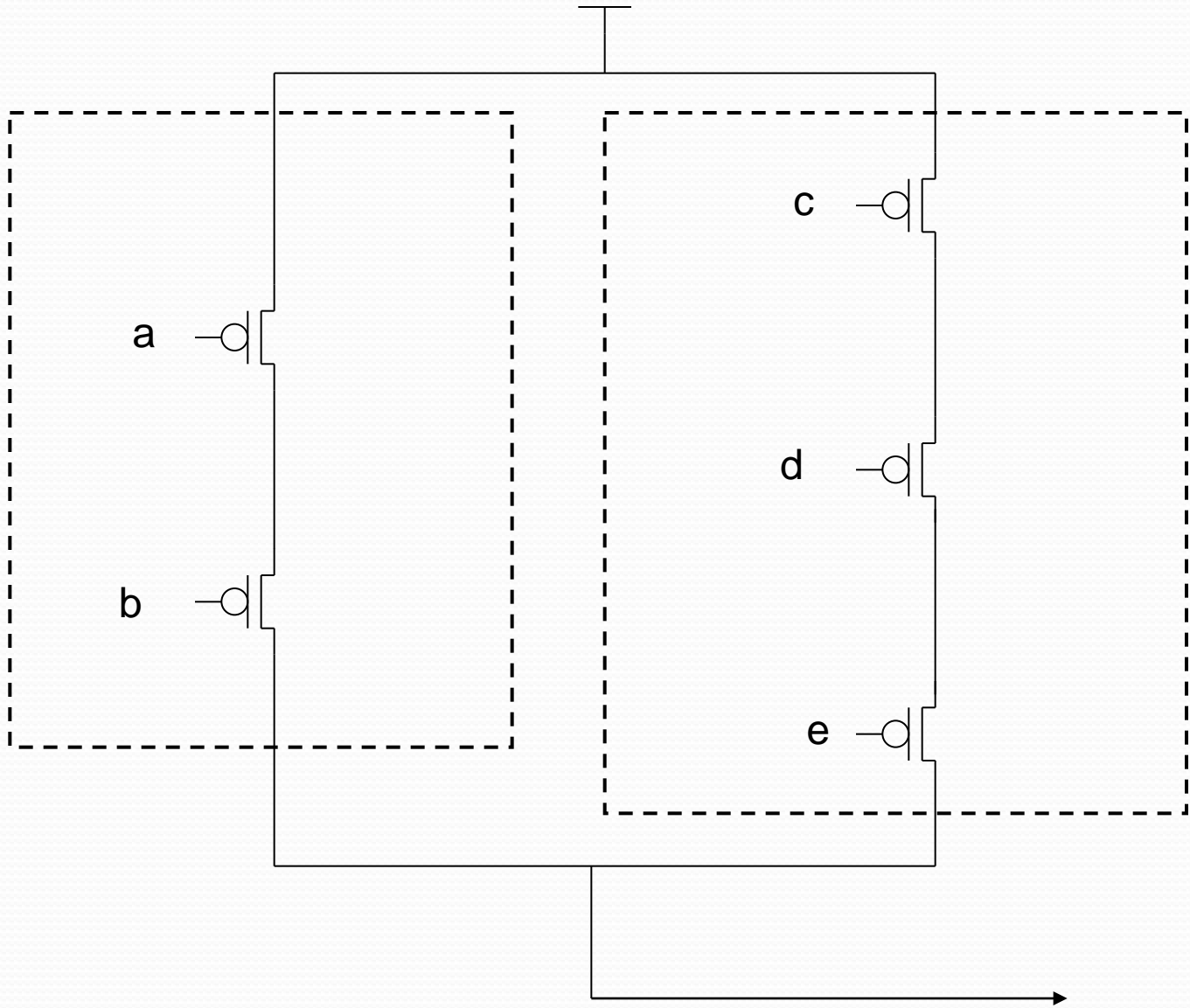
A diagram illustrating the decomposition of a fraction. At the top, a box contains the fraction $\frac{(a+b) \cdot (c+d+e)}{(a+b) \cdot (c+d+e)}$. A solid arrow points from the top of this box to the right. Below the box is a ground symbol. A dashed arrow points from the ground symbol down and to the right towards a tree diagram. The tree diagram has a root node with a vertical line and a horizontal line extending to the right. From the horizontal line, two vertical lines lead down to two separate boxes. The left box contains $\frac{(a+b)}{(a+b)}$ and the right box contains $\frac{(c+d+e)}{(c+d+e)}$. A solid arrow points from the bottom of the tree diagram to the right.

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



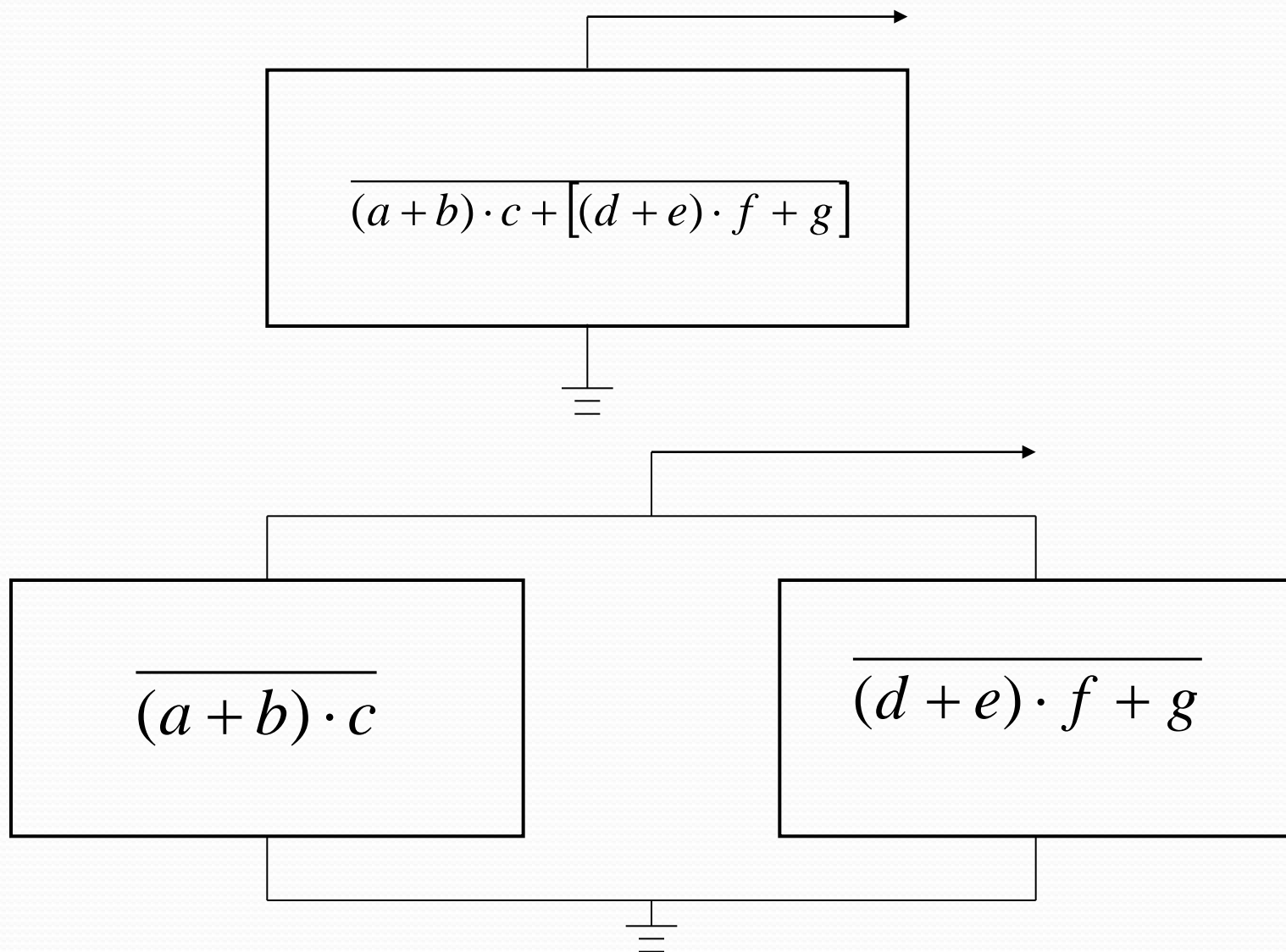
$$\frac{(a+b)}{(a+b)}$$

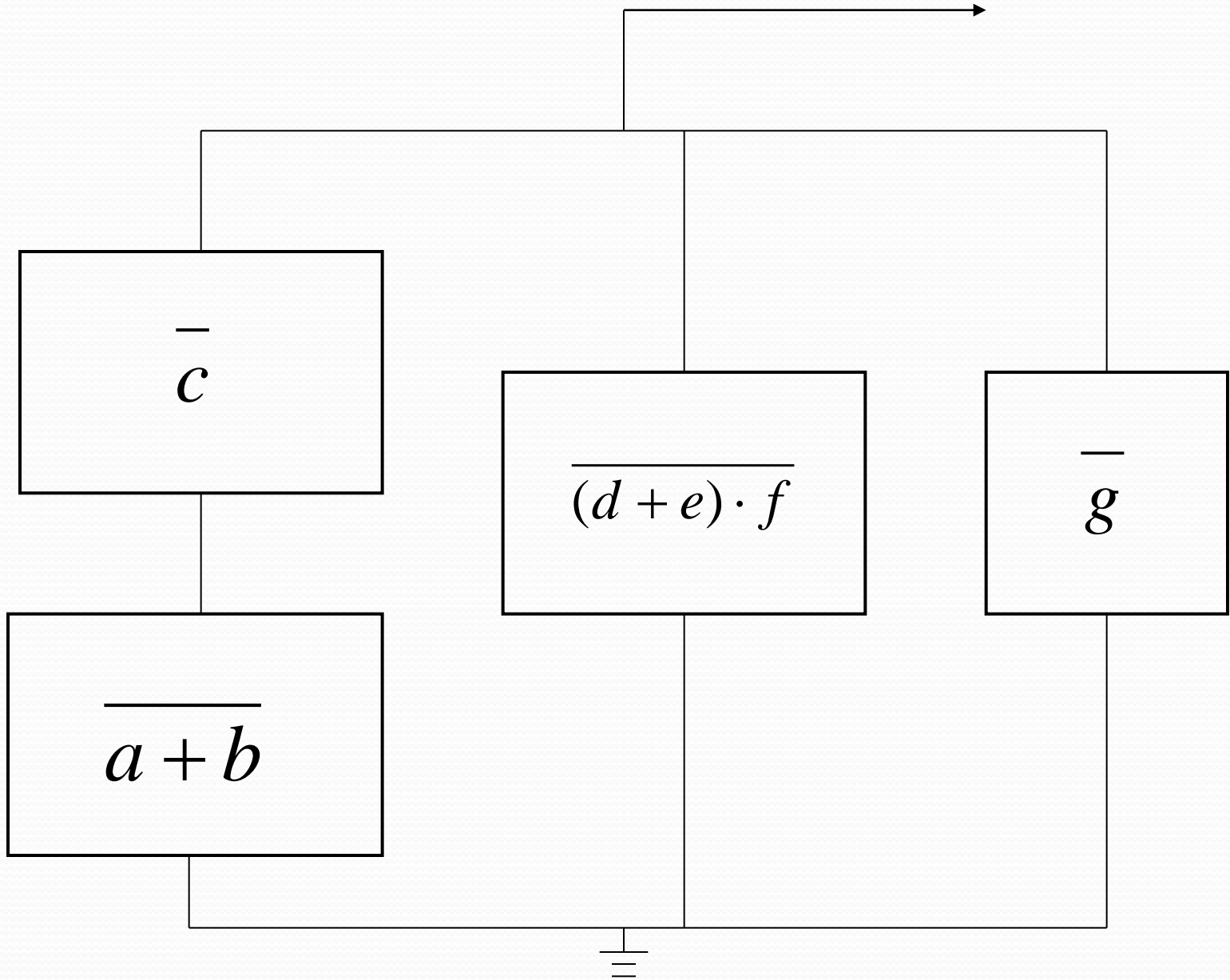
$$\frac{(c+d+e)}{(c+d+e)}$$

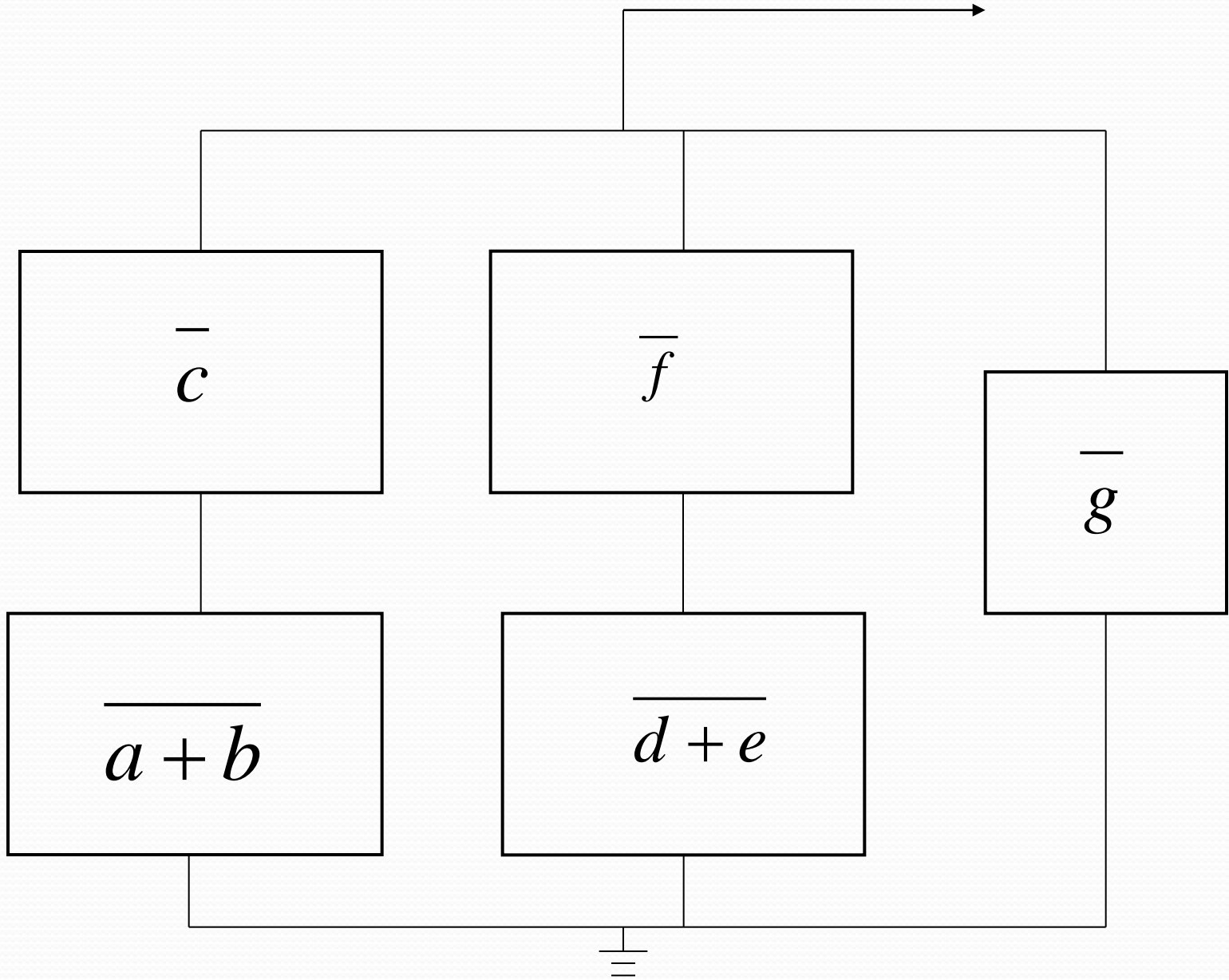


- Υλοποιήστε το n-δικτύωμα της ακόλουθης συνάρτησης

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

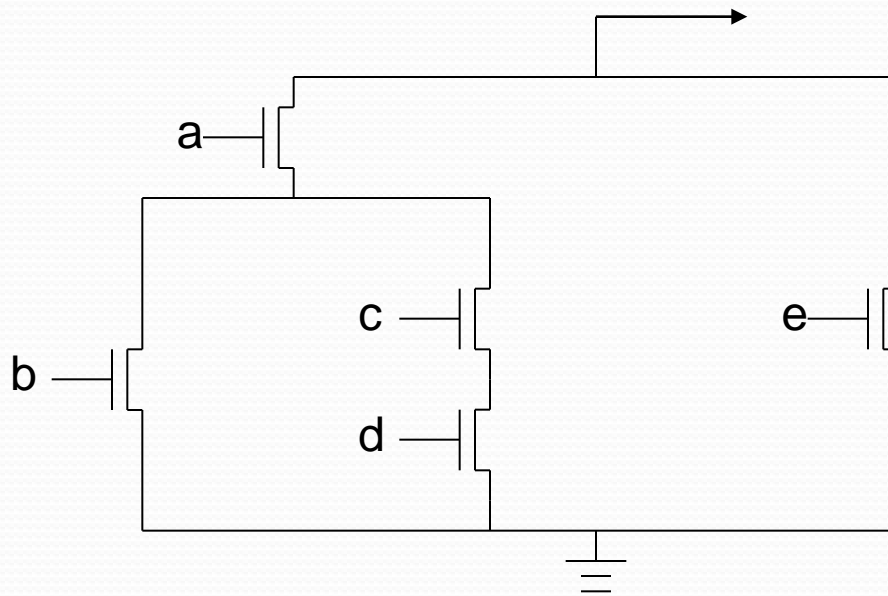


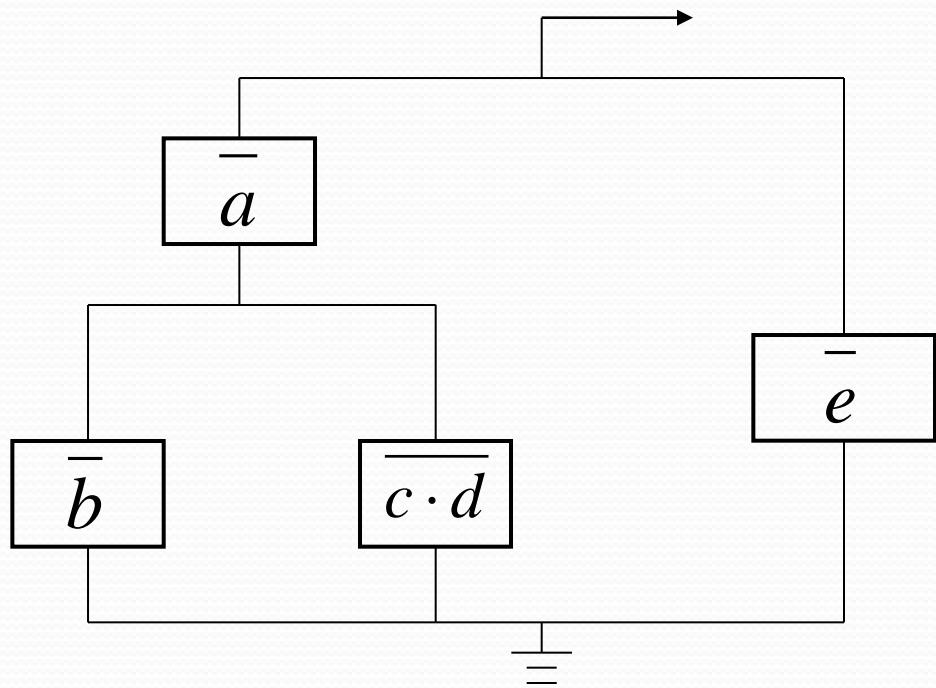
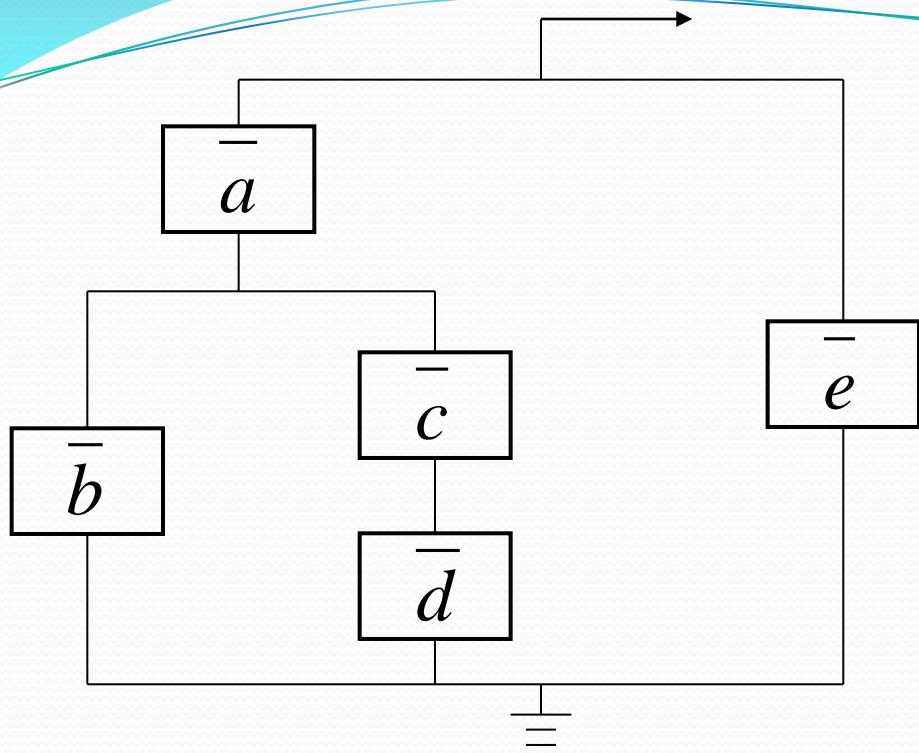


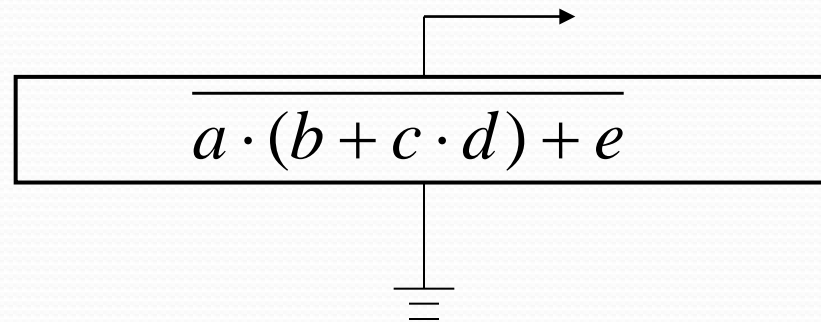
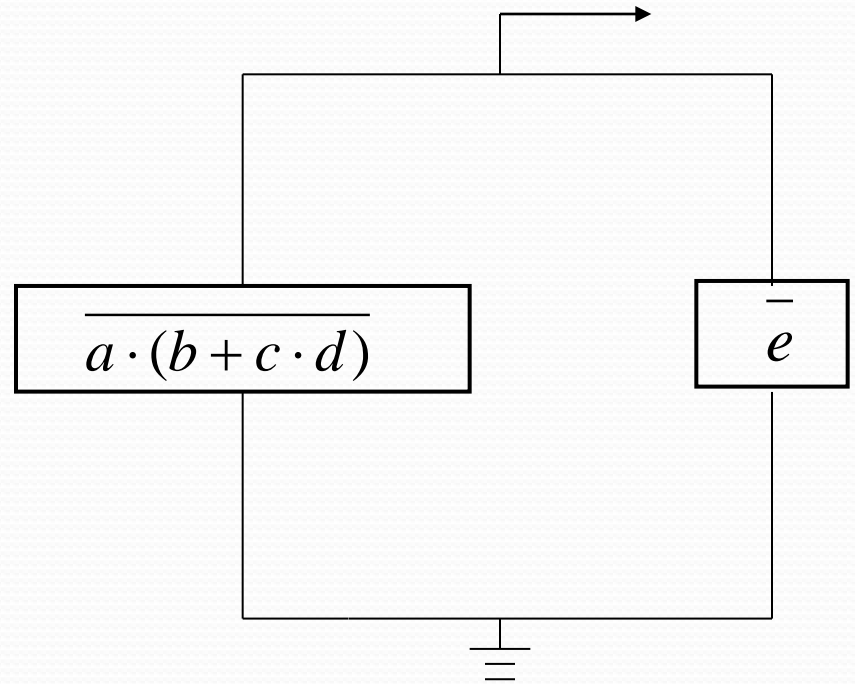
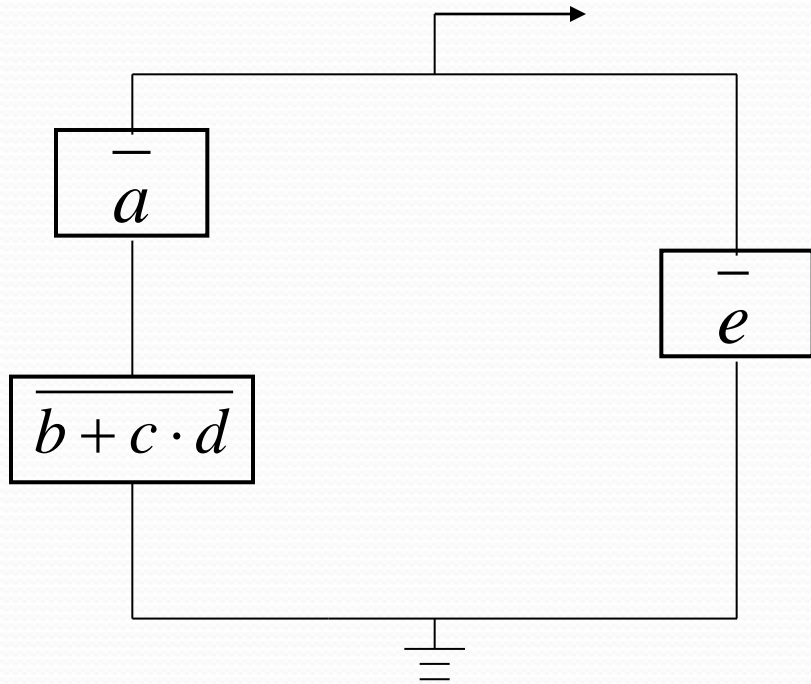


Εύρεση συνάρτησης από σχέδιο σε επίπεδο τρανζίστορ

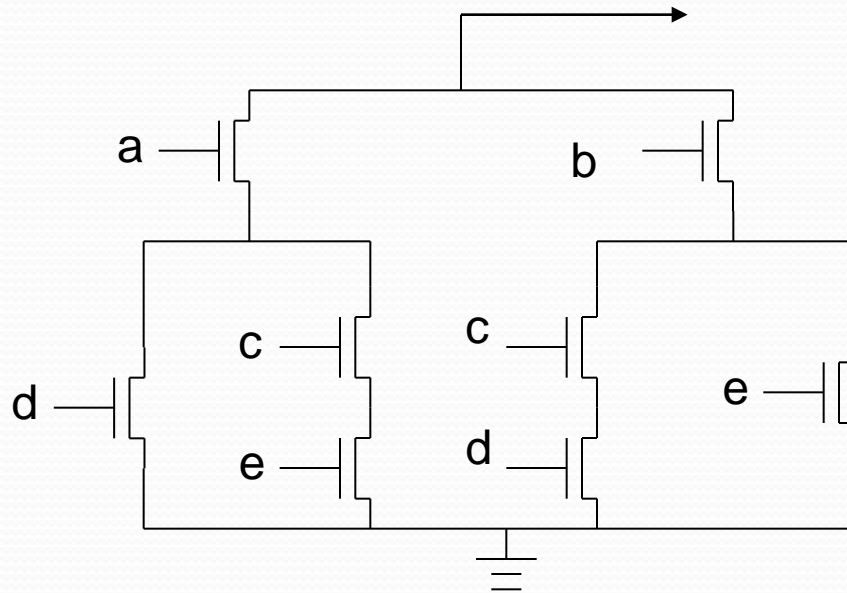
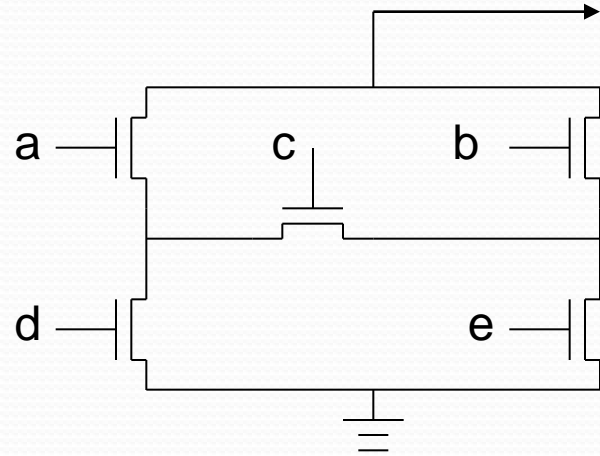
- Σε ποιά συνάρτηση αντιστοιχεί το ακόλουθο n-MOS δικτύωμα;



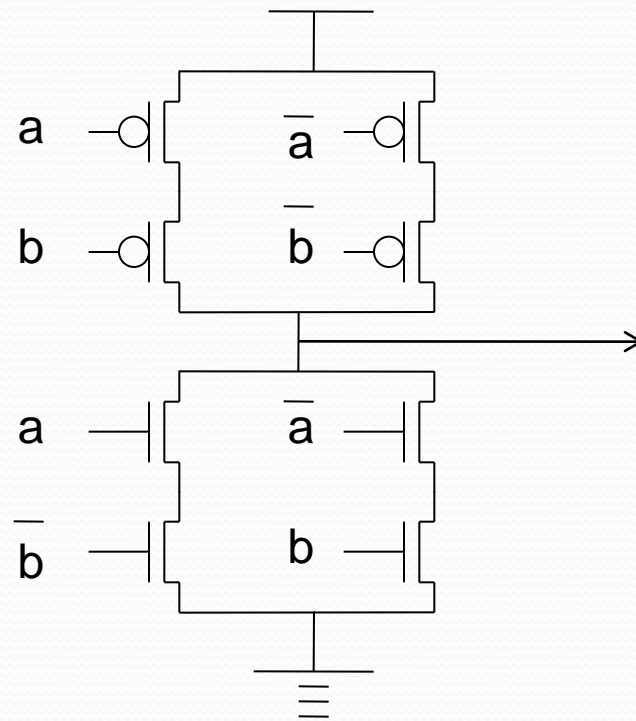




- Οι προηγούμενες μεθοδολογίες ισχύουν για σειριακά-παράλληλα δικτυώματα
- Το ακόλουθο δικτύωμα δεν ανήκει σε αυτή την κατηγορία αλλά ακολουθώντας τα μονοπάτια του (paths) μπορούμε να βρούμε ένα ισοδύναμο κύκλωμα και την συνάρτηση



Ταυτοποίηση n-MOS και p-MOS δικτύωματος



Για το κύκλωμα έχω

- Από το p-MOS δικτύωμα παίρνω

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

- Από το n-MOS δικτύωμα παίρνω

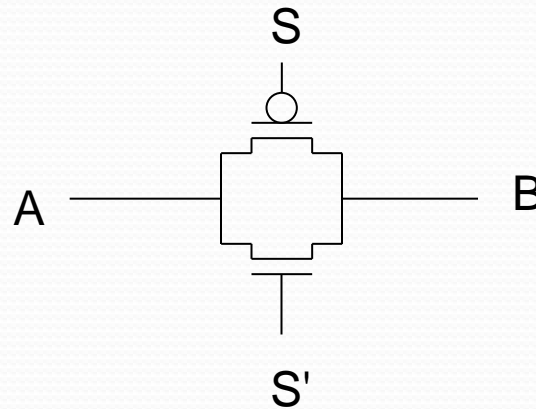
$$\overline{a \cdot \bar{b} + \bar{a} \cdot b}$$

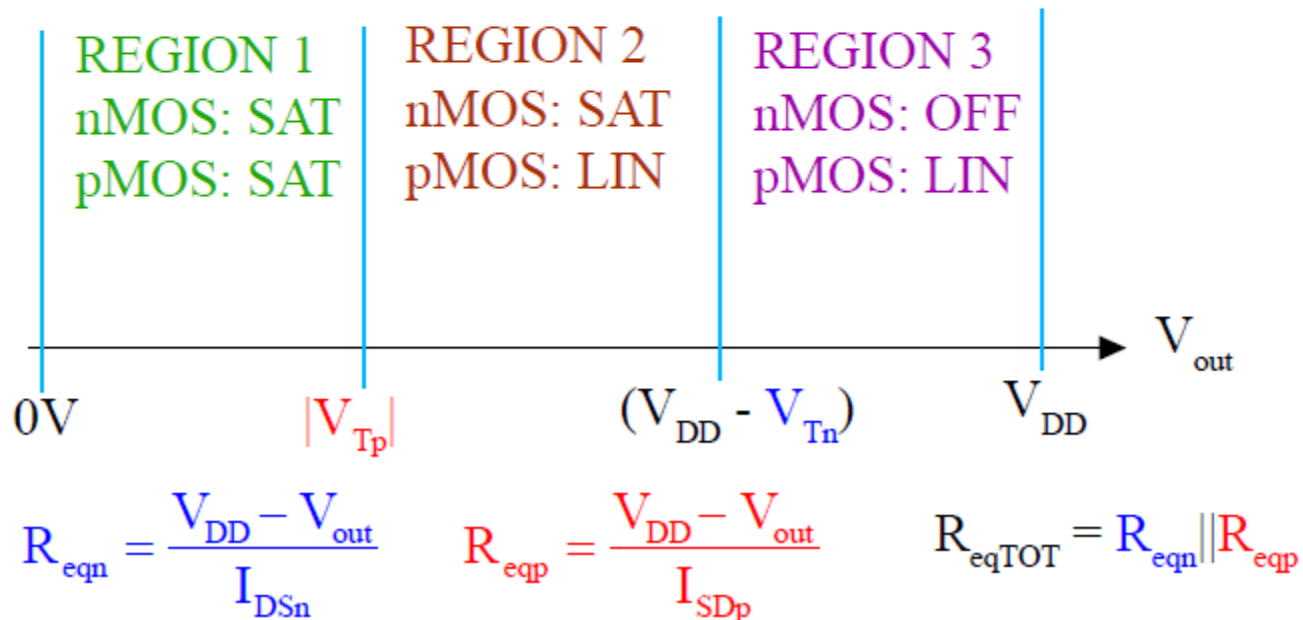
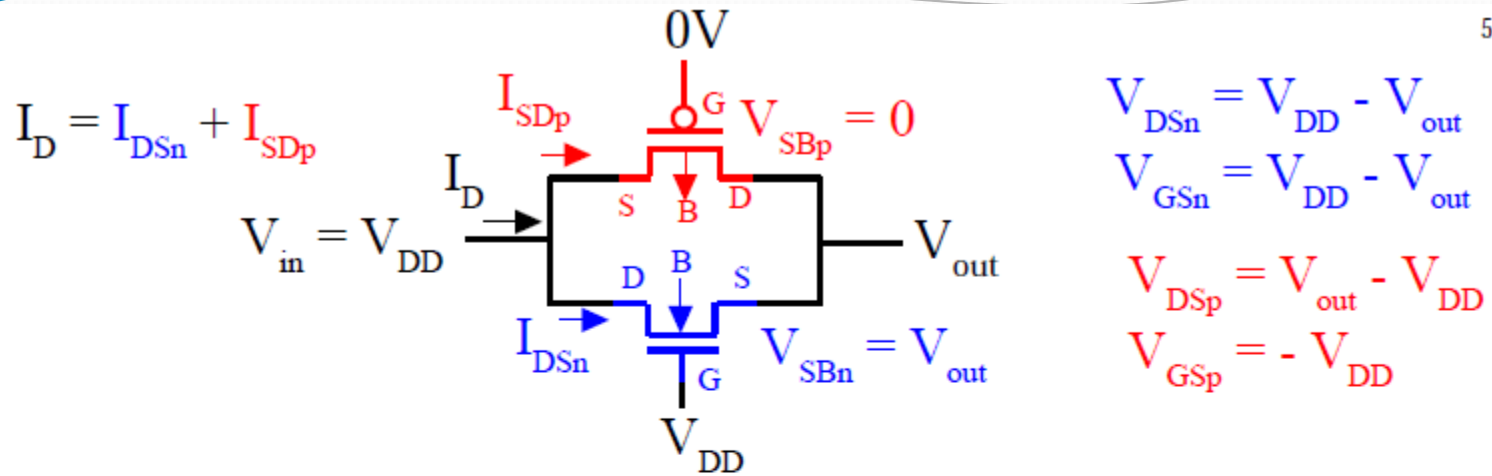
- Προσοχή είναι η ίδια συνάρτηση

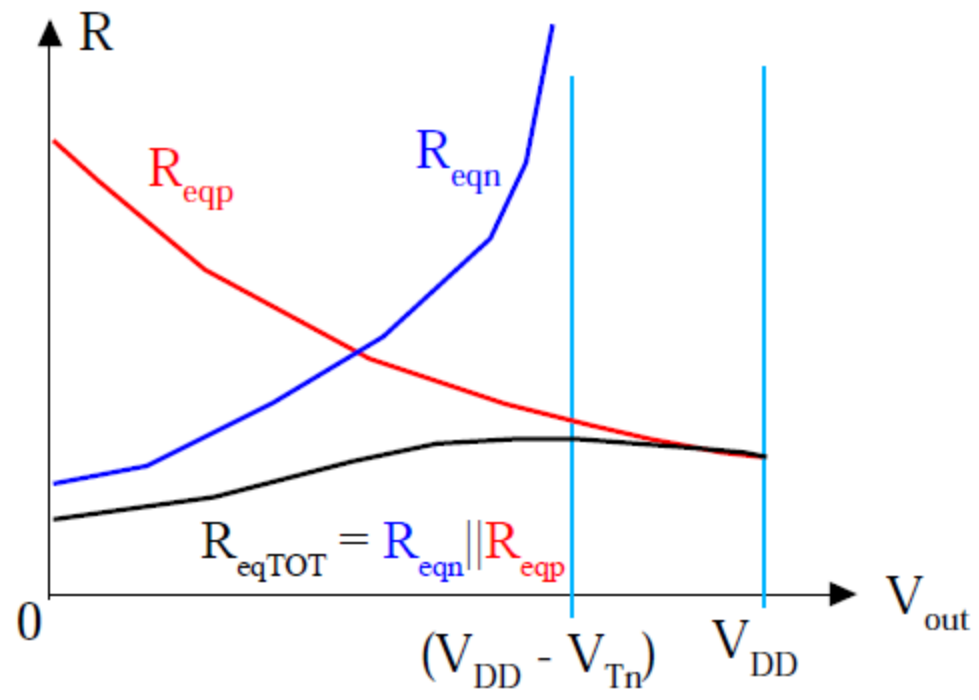
$$\begin{aligned} \overline{(a + b) \cdot (\bar{a} + \bar{b})} &= \overline{a \cdot \bar{a} + a \cdot \bar{b} + b \cdot \bar{a} + b \cdot \bar{b}} = \\ &= \overline{0 + a \cdot \bar{b} + \bar{a} \cdot b + 0} = \overline{a \cdot \bar{b} + \bar{a} \cdot b} \end{aligned}$$

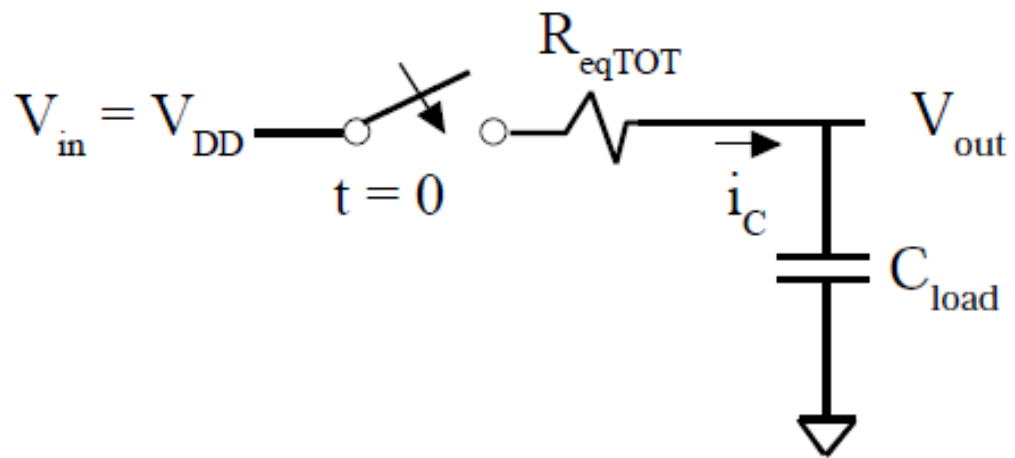
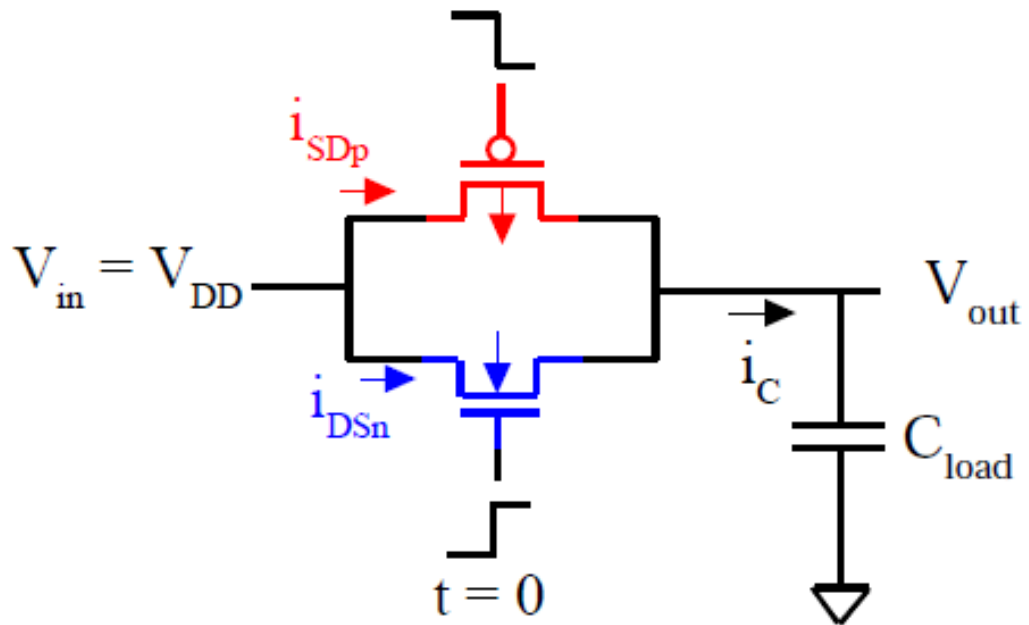
Λογική τρανζίστορ διέλευσης (pass-logic)

- 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)



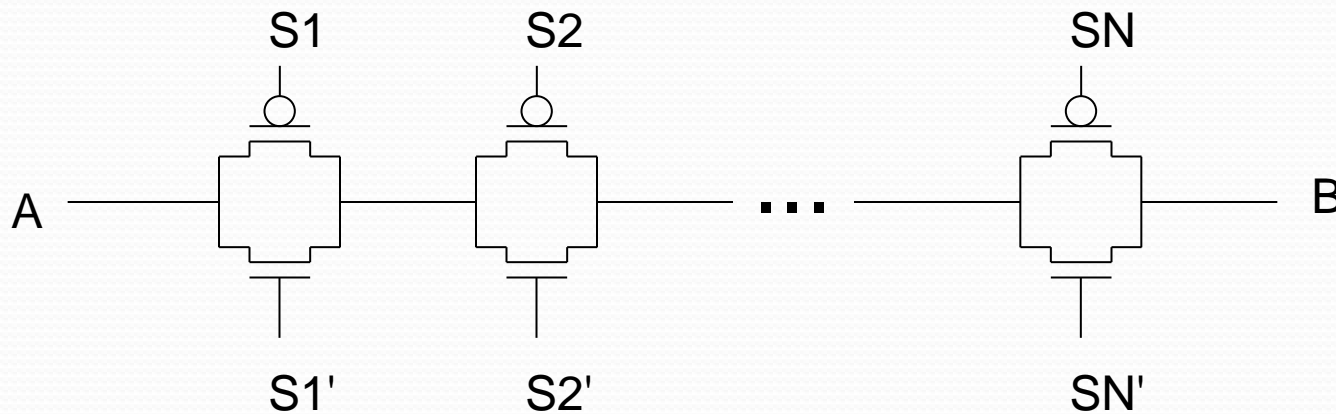




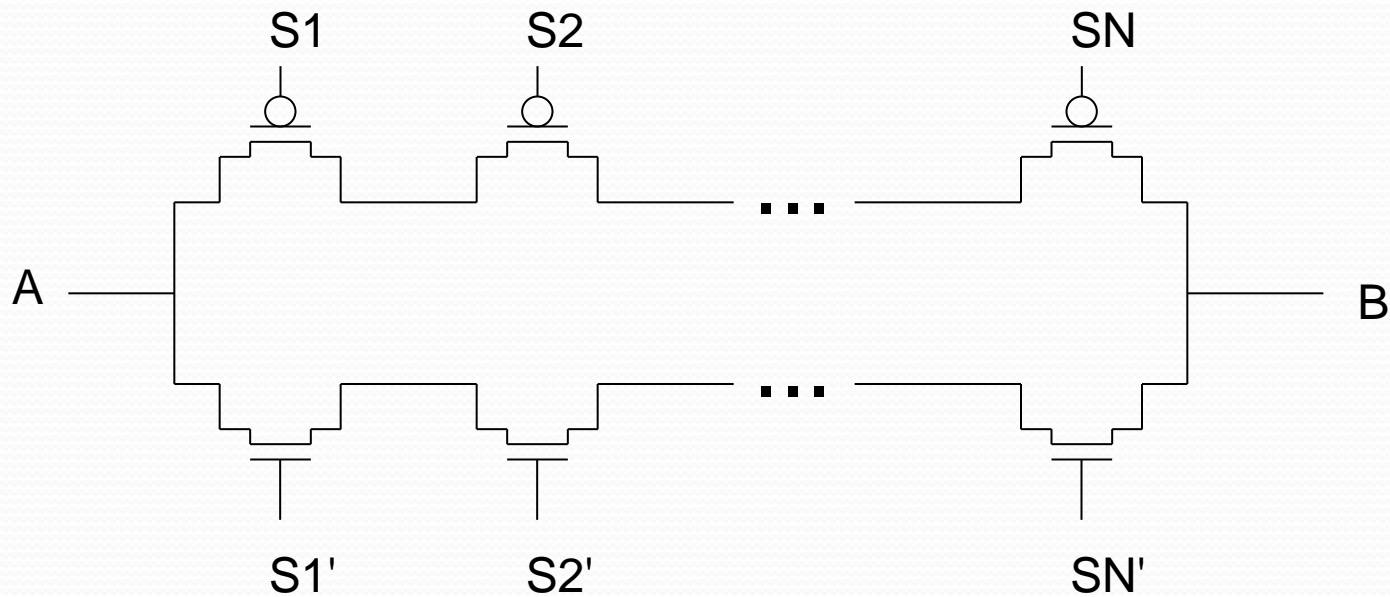


- Στην γενική περίπτωση θα πρέπει

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

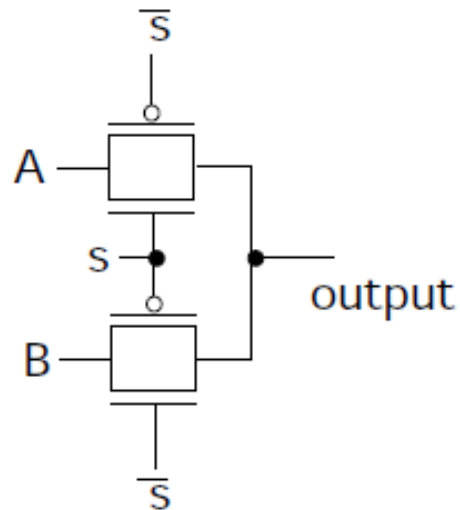


Το προηγούμενο κύκλωμα είναι ισοδύναμο με το



Υλοποίηση Πολυπλέκτη και XOR

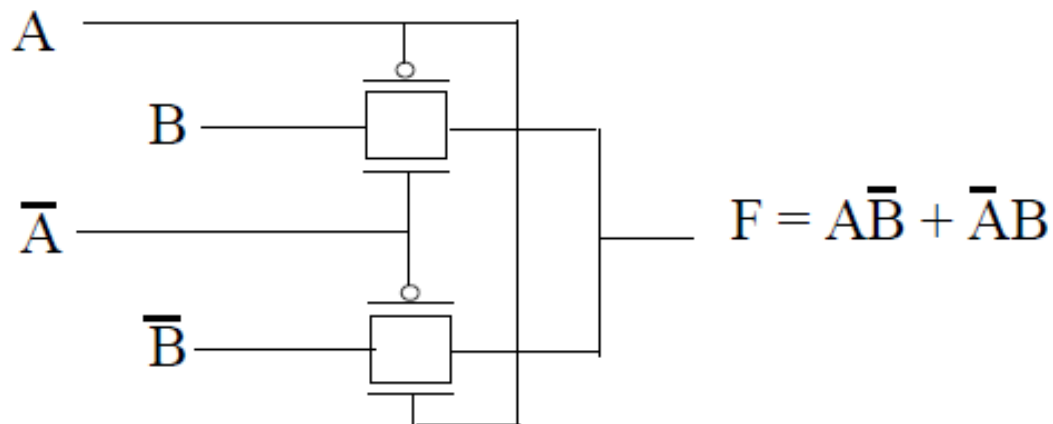
2-INPUT MULTIPLEXER



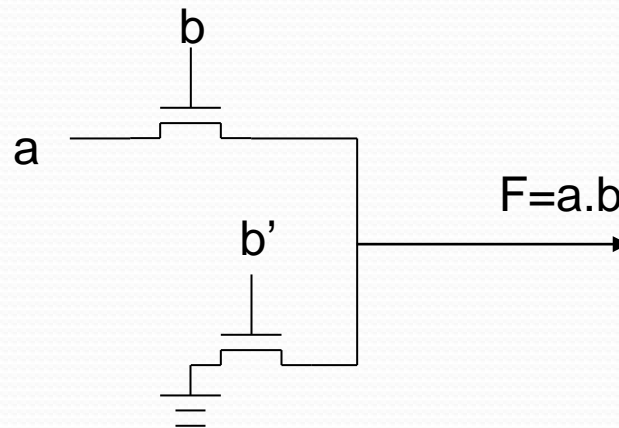
$$\text{output} = A.s + B.\bar{s}$$

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)

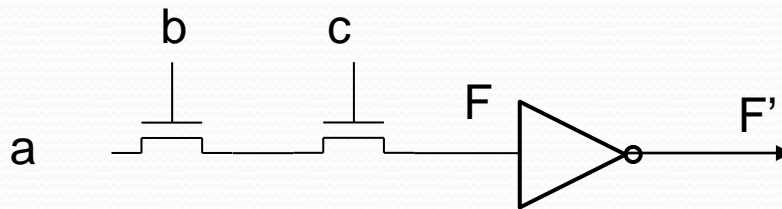
XOR



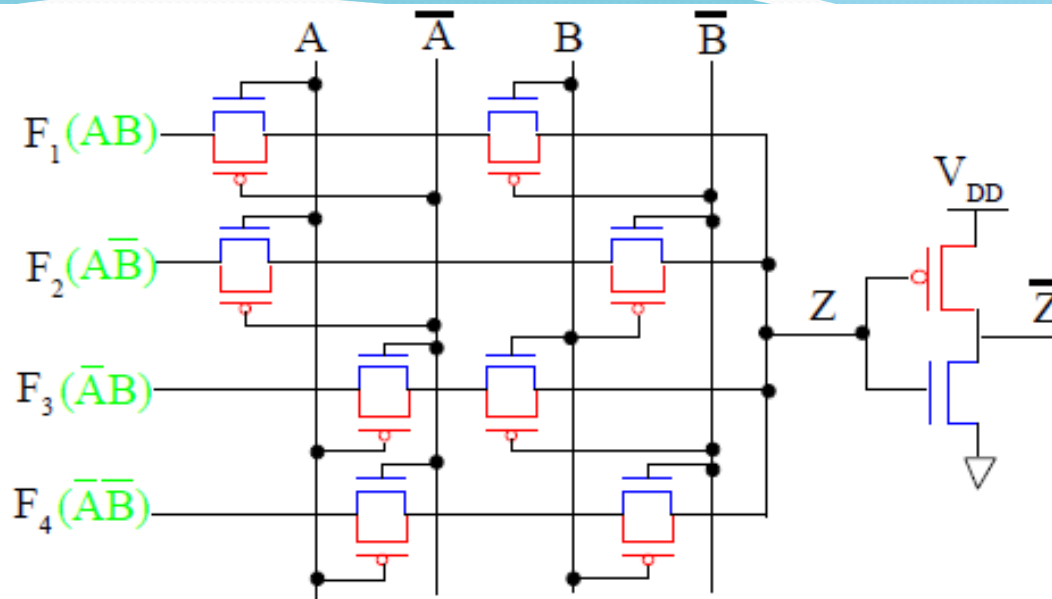
- Στην γενική περίπτωση για να υλοποιήσω πύλη θα πρέπει για κάθε συνδυασμό εισόδων να συνδέω την έξοδο σε μία τιμή
(εάν είναι ασύνδετη η τιμή θα είναι hiZ, υψηλή εμπέδηση)



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



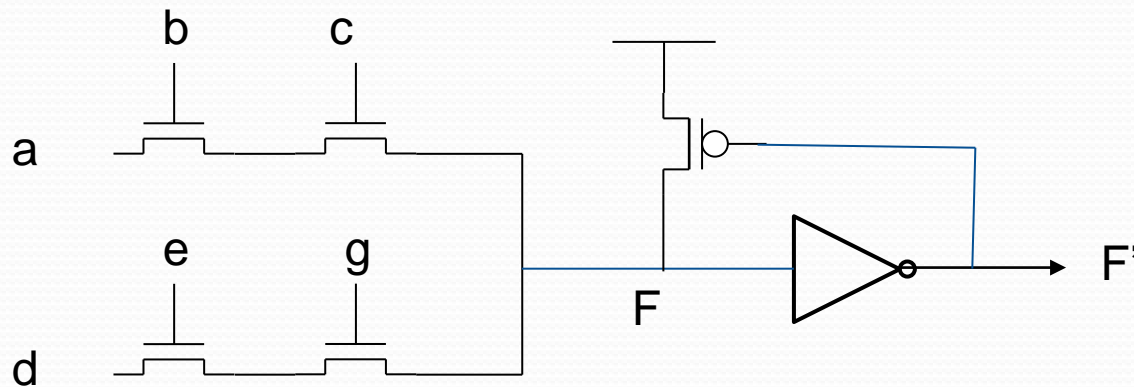
- Εάν ένα μονοπάτι μόνο φορτίζει δεν χρειάζονται n-MOS τρανζίστορ
- Αντίστοιχα εάν μόνο αποφορτίζει δεν χρειάζονται p-MOS τρανζίστορ



SOME OF THE FUNCTIONS REALIZED BY THE BOOLEAN FUNCTION UNIT

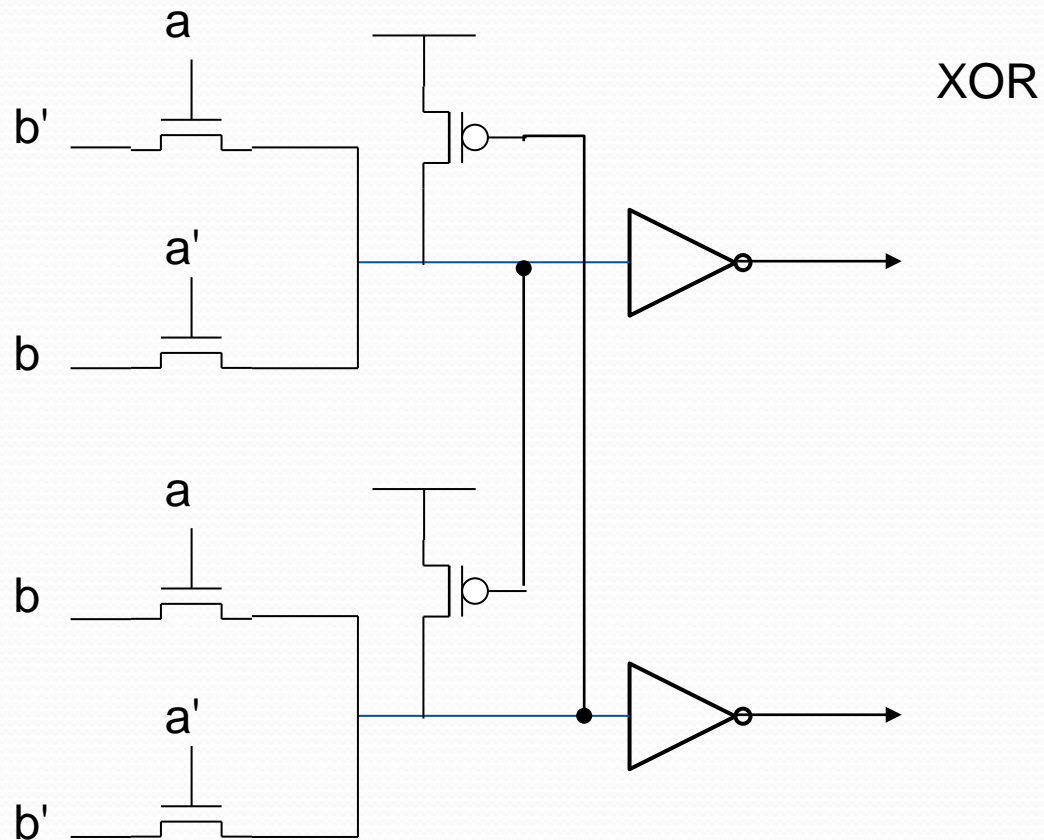
OPERATION (Z)	F_1	F_2	F_3	F_4
NOR(A,B)	0	0	0	1
XOR(A,B)	0	1	1	0
NAND(A,B)	0	1	1	1
AND(A,B)	1	0	0	0
OR(A,B)	1	1	1	0

- 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.

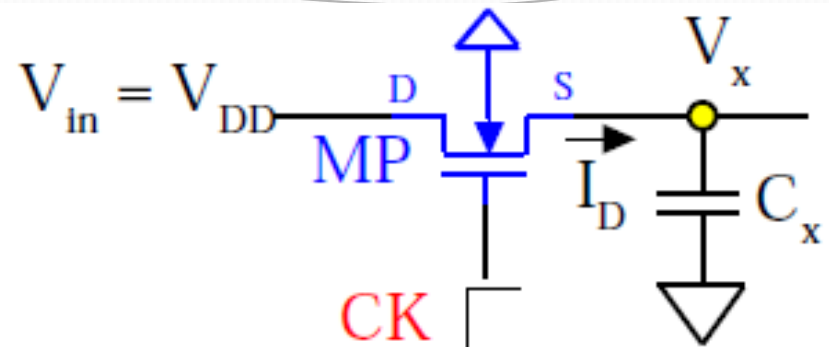
LOGIC "1" TRANSFER:

Assume at $t = 0$:

$$V_x(t = 0) = 0 \text{ V}$$

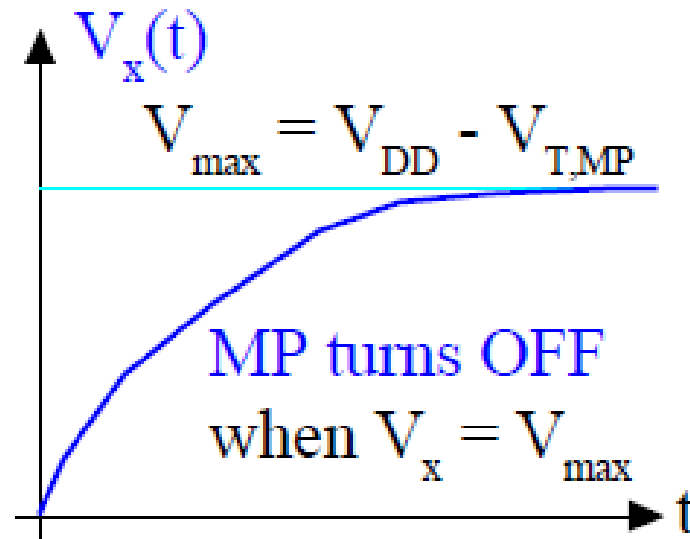
$$V_{in} = V_{OH}$$

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



$$V_{GS} = V_{DD} - V_x, \quad V_{DS} = V_{DD} - V_x$$

$$V_{DS} > V_{GS} - V_{T,MP} \Rightarrow \text{nMOS in SAT}$$

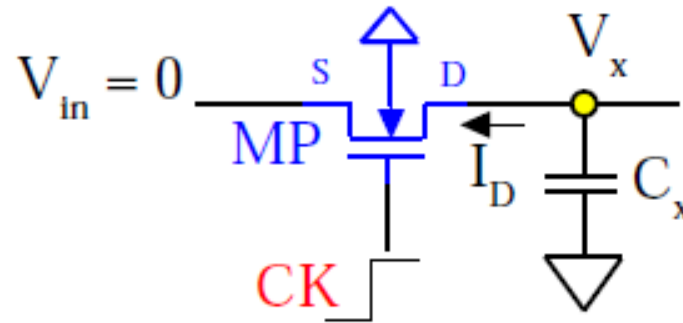


LOGIC "0" TRANSFER:

Assume at $t = 0$:

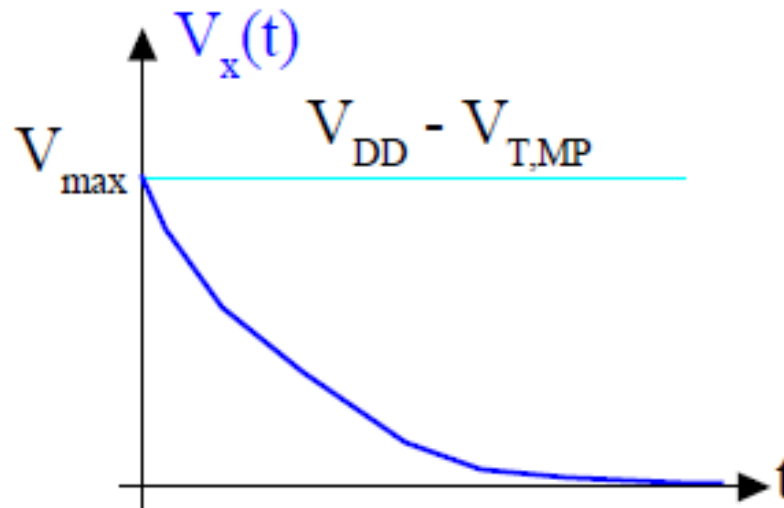
$$V_x(t = 0) = V_{\max} = V_{DD} - V_{T,MP}$$

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

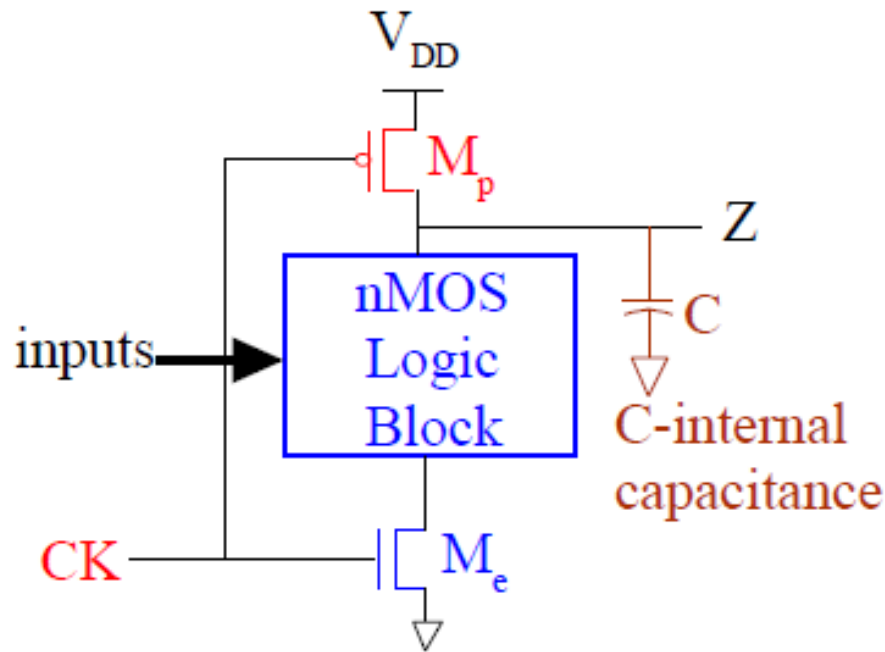


$$V_{GS} = V_{DD}, V_{DS} = V_{\max} = V_{DD} - V_{T,MP}$$

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



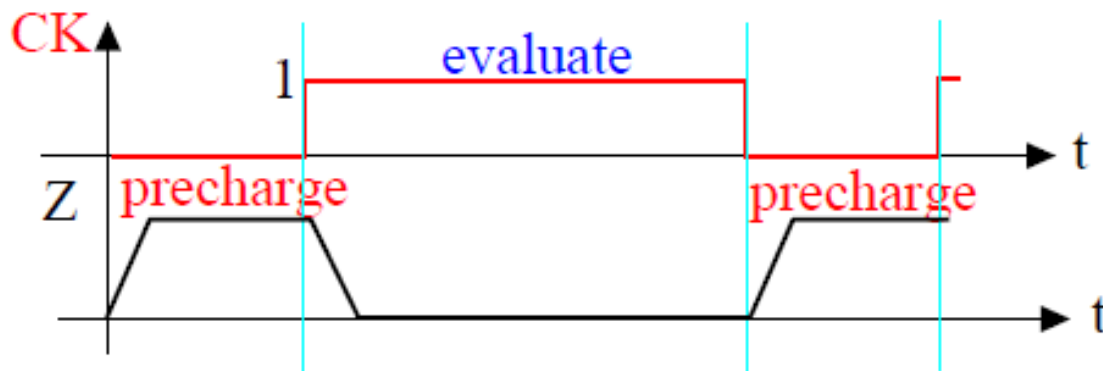
Δυναμική λογική CMOS δύο φάσεων

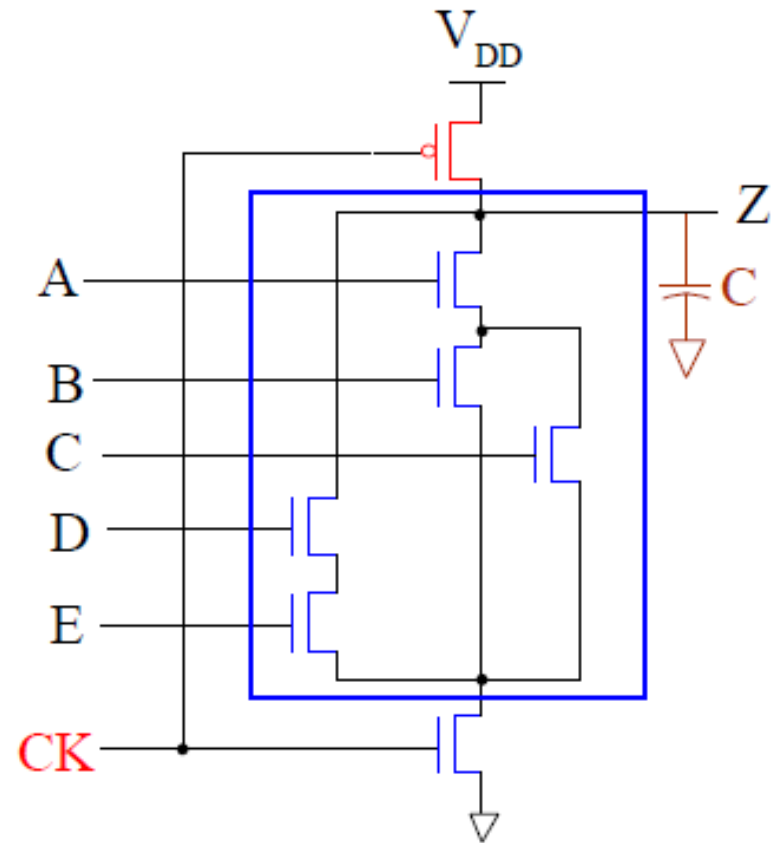


CK = 0 \rightarrow C precharges to V_{DD}
(output is unavailable during precharge)

CK = 1 \rightarrow C is selectively discharges to 0

(output is only available after discharge is complete)



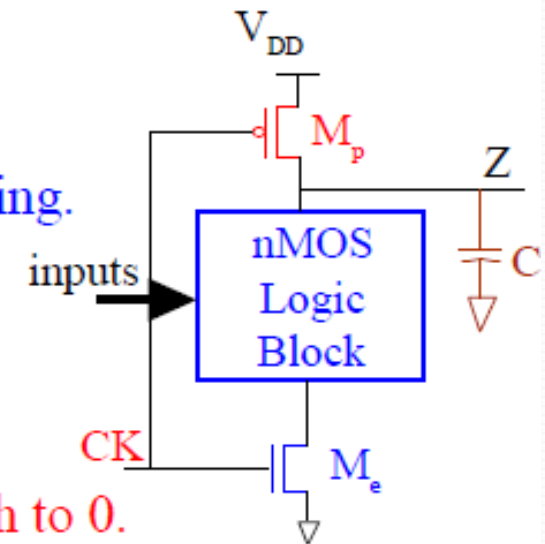


$$Z = A.(B + C) + (D.E) \text{ when } CK = 1$$

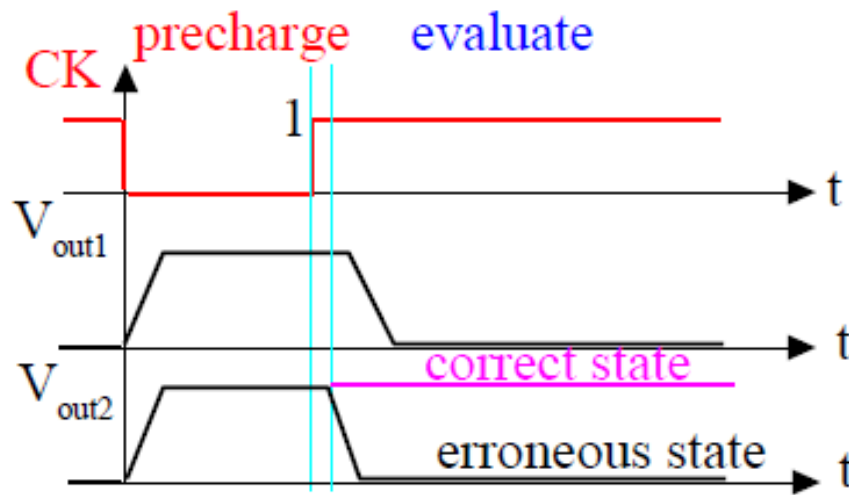
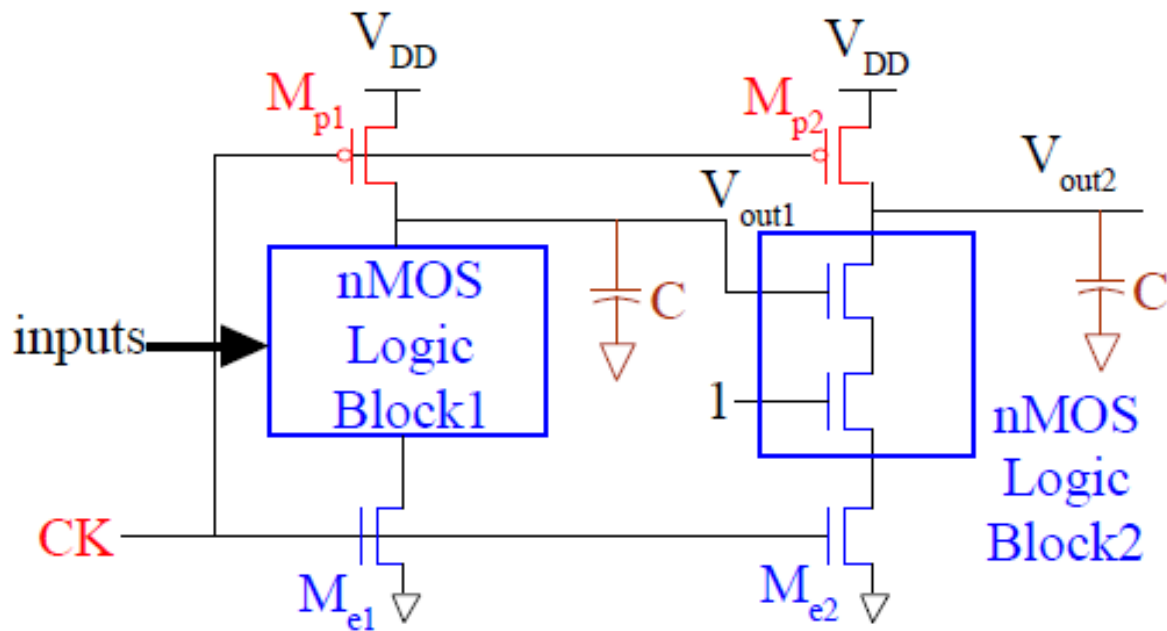
$$Z = \text{HIGH} \text{ when } CK = 0$$

ADVANTAGES/DISADVANTAGES

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} .



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 connecte 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 on 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.



evaluate:

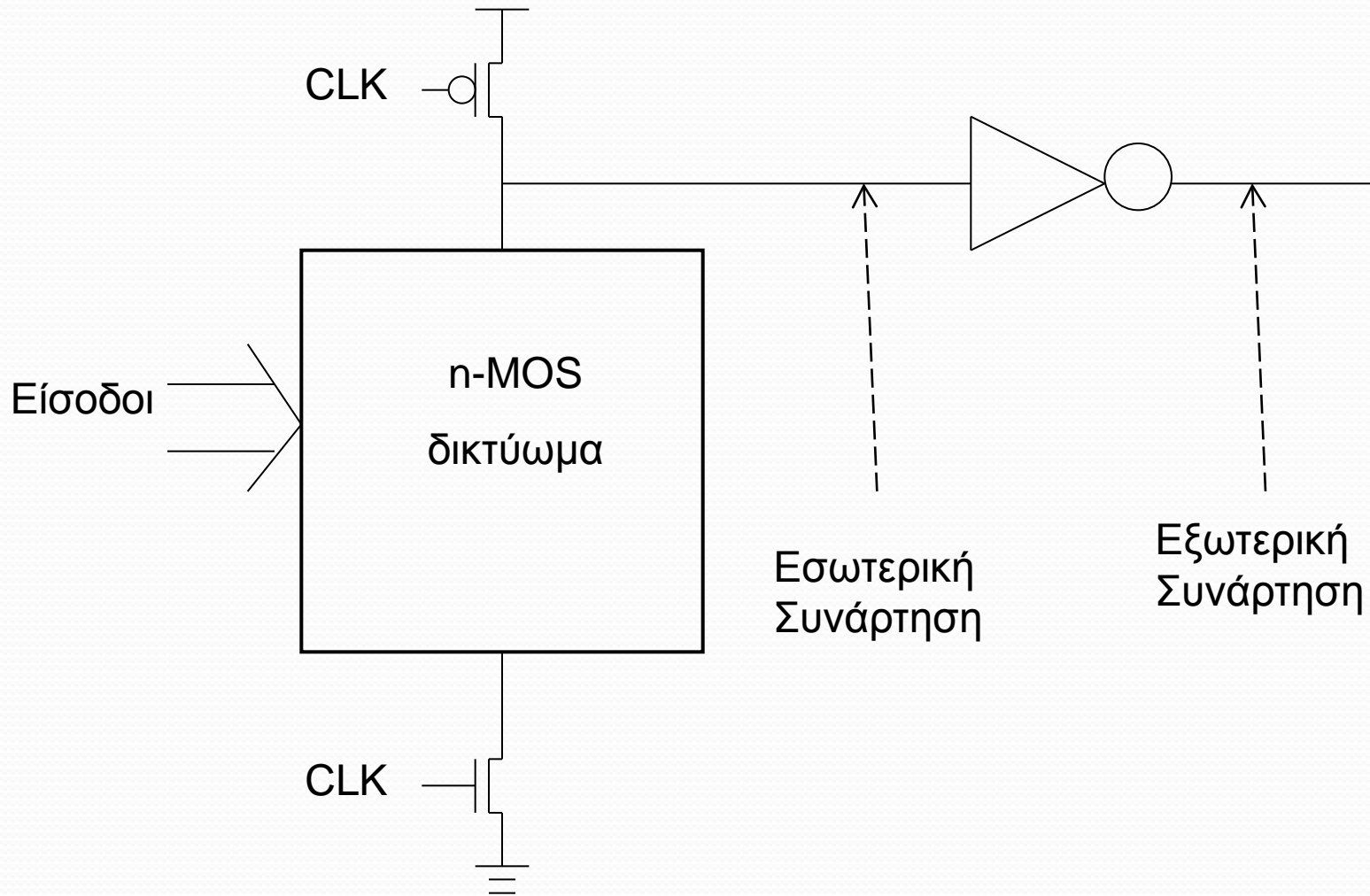
$M_{e1}, M_{e2} \rightarrow$ ON

$M_{p1}, M_{p2} \rightarrow$ 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 έξοδου 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 οδηγείτε από το CLK φορτίζει τον κόμβο της εσωτερικής συνάρτησης
- Το n-MOS τρανζίστορ που οδηγείτε 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 χρειαζόμαστε 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

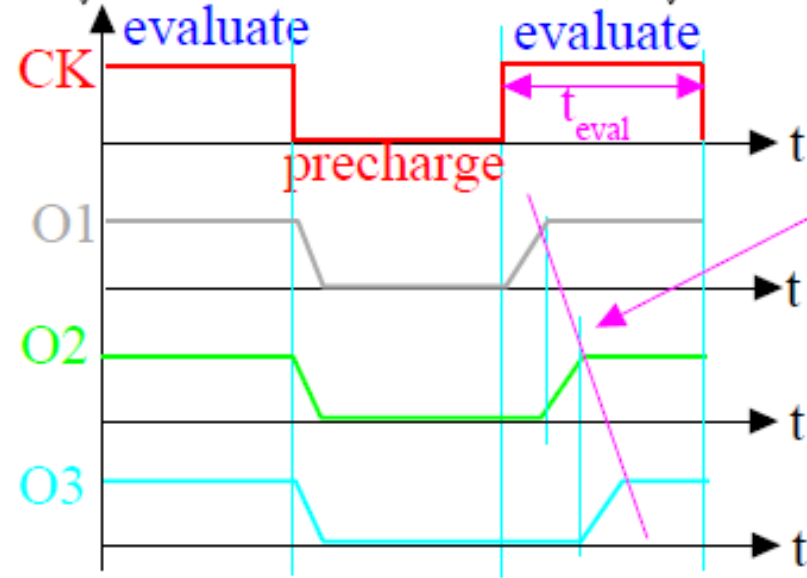
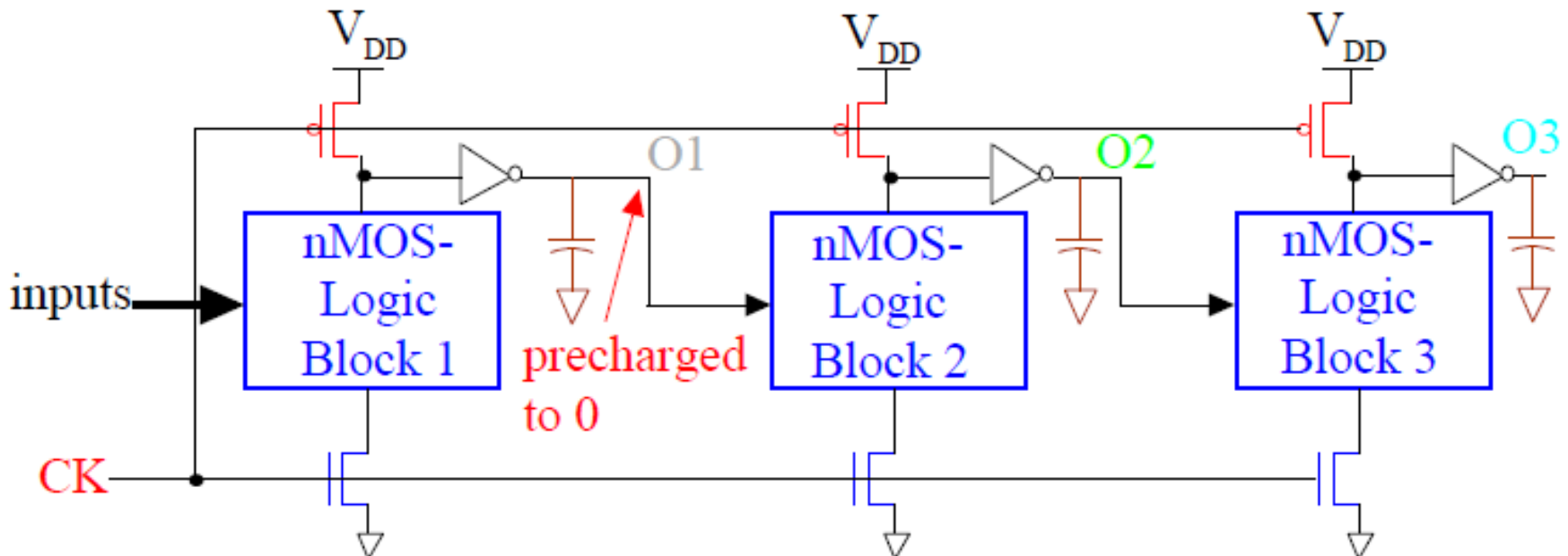
- ΔΕΝ χρειαζόμαστε p-MOS δικτύωμα
- Ταχύτερα κυκλώματα

- 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



propogating gate decisions

Max # stages limited:
total prop delay < t_{eval} .

Σχεδιασμός DOMINO

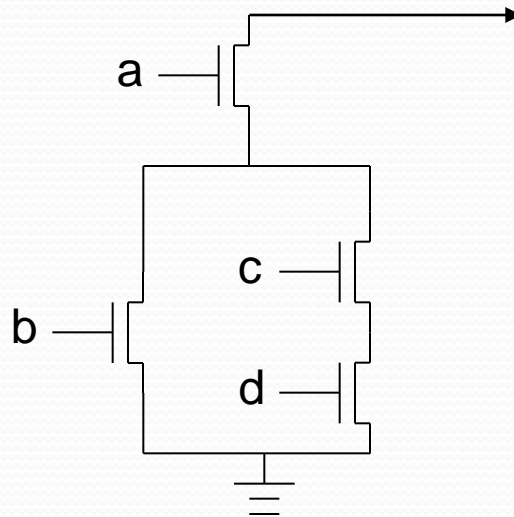
- Από την συνάρτηση εξόδου F γνωρίζω την εσωτερική συνάρτηση F' . Το n-MOS δικτύωμα της Domino πύλης είναι το n-MOS δικτύωμα της CMOS συνάρτησης F'
- Για παράδειγμα για τη συνάρτηση

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

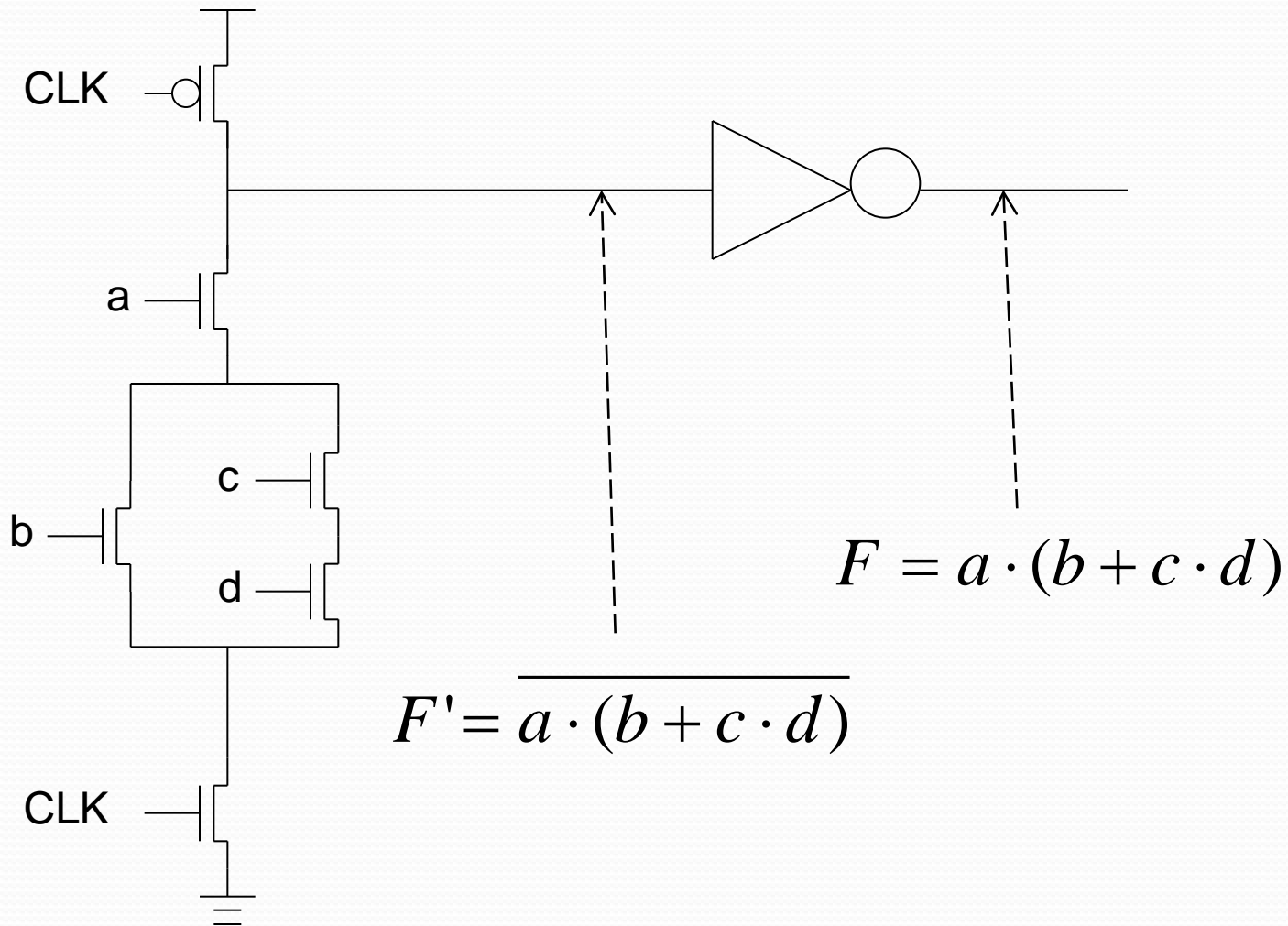
- Η F' είναι

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

- Με η-δικτύωμα



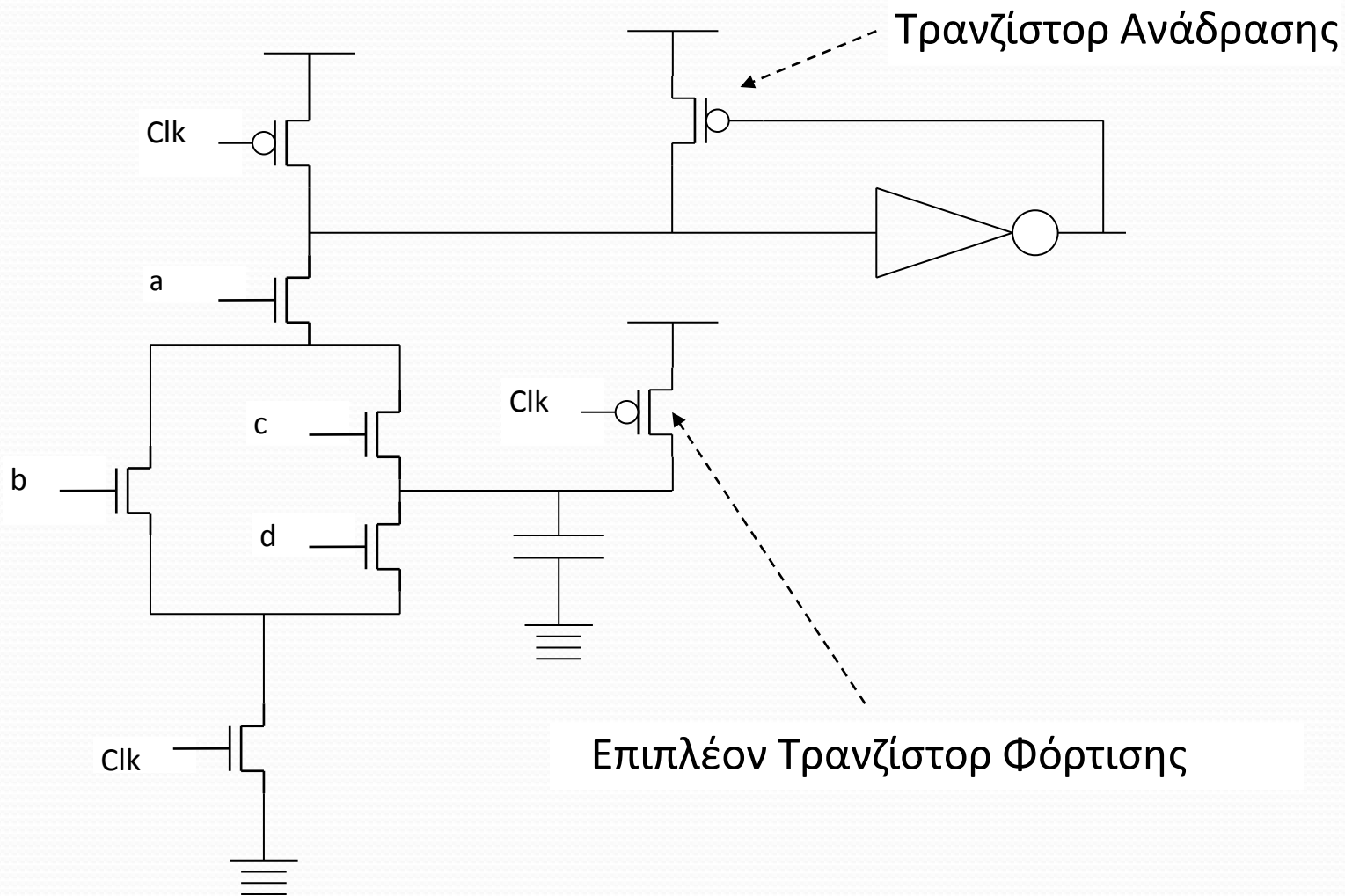
Η Domino Πύλη είναι



Διαμοιρασμός φορτίου

- 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

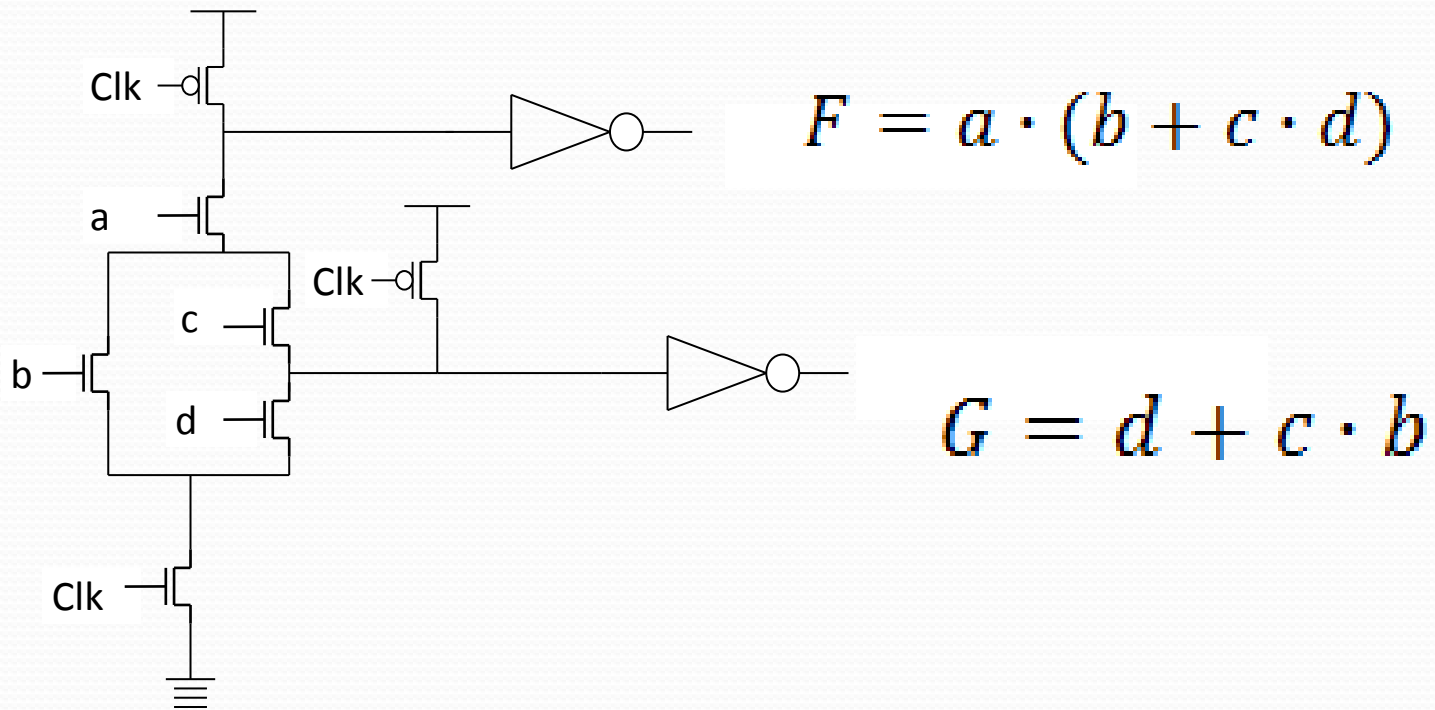
Διαμοιρασμός φορτίου



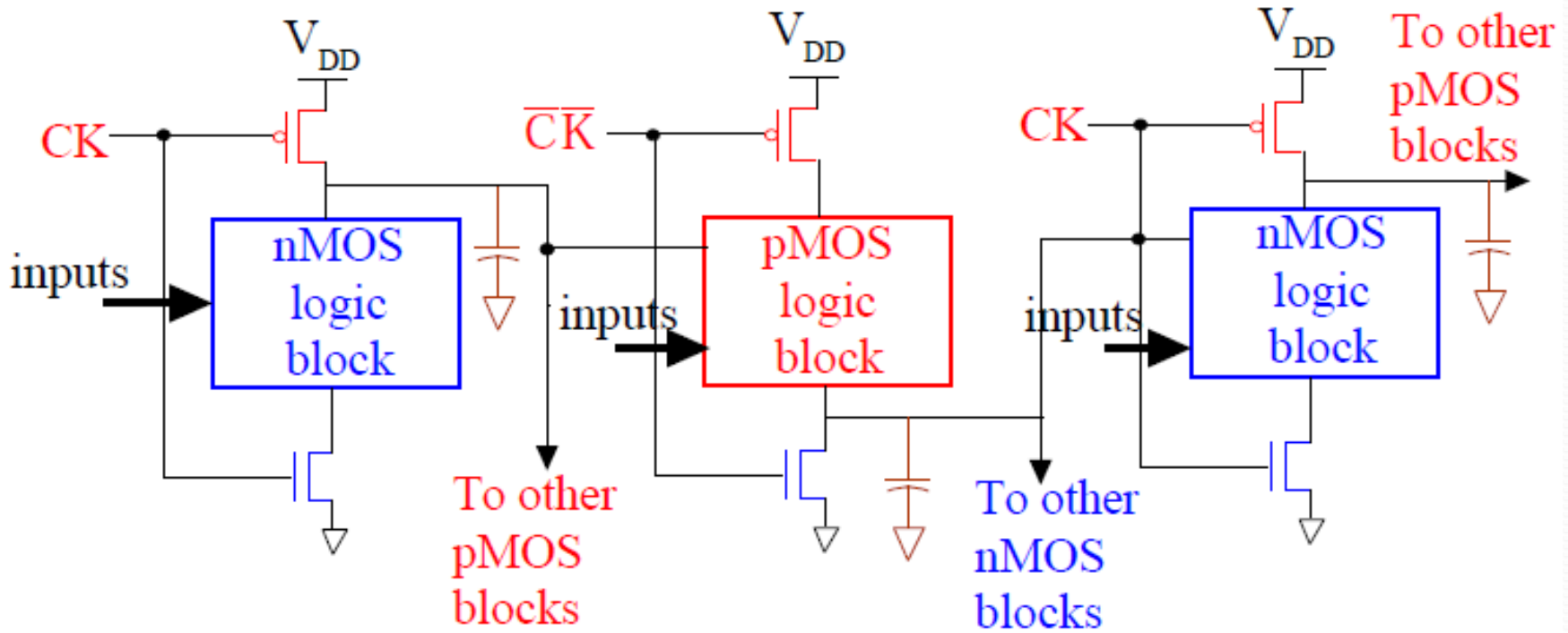
Domino Πολλαπλών εξόδων (Multiple-Output Domino)

- 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)

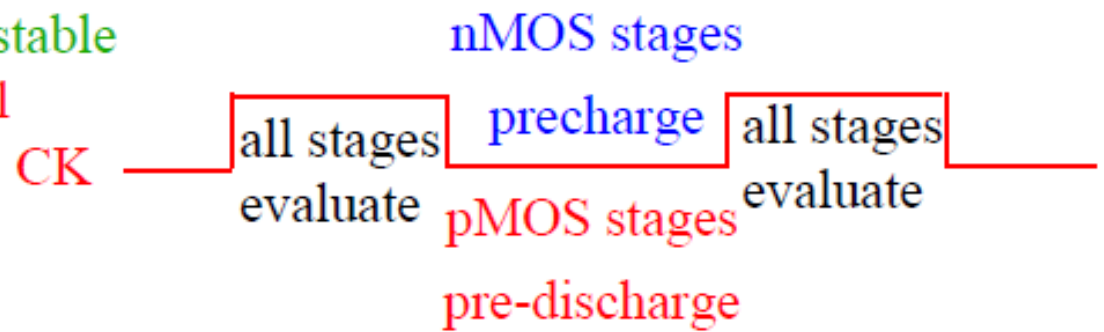


NP DOMINO LOGIC (NORA or ZIPPER CMOS)

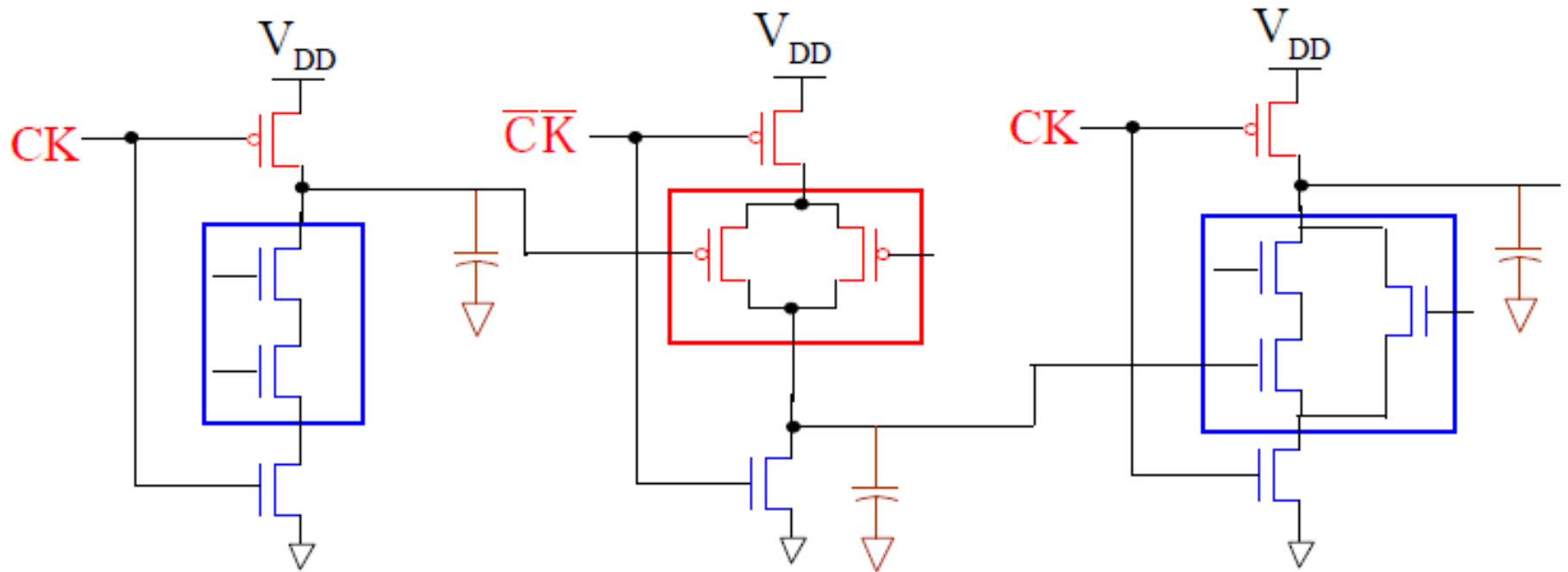


NOTE: INVERTERS ARE NOT REQUIRED AT OUTPUTS OF STAGES

ALL inputs stable
when $CK = 1$



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

DCVS

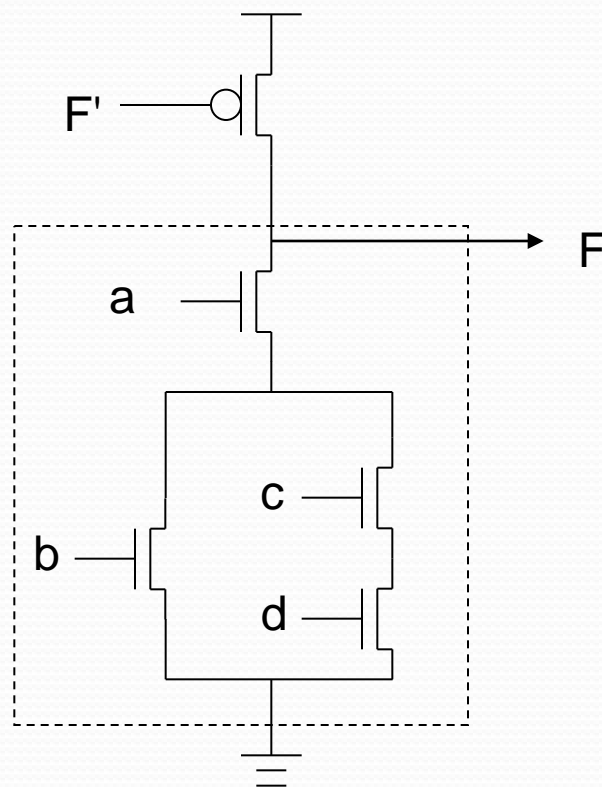
- Εάν υπάρχει η συμπληρωματική συνάρτηση το p-MOS δικτύωμα μπορεί να αντικατασταθεί από ένα p-MOS τρανζίστορ που οδηγείται από την συμπληρωματική
- Για παράδειγμα η συνάρτηση

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

εάν υπάρχει η

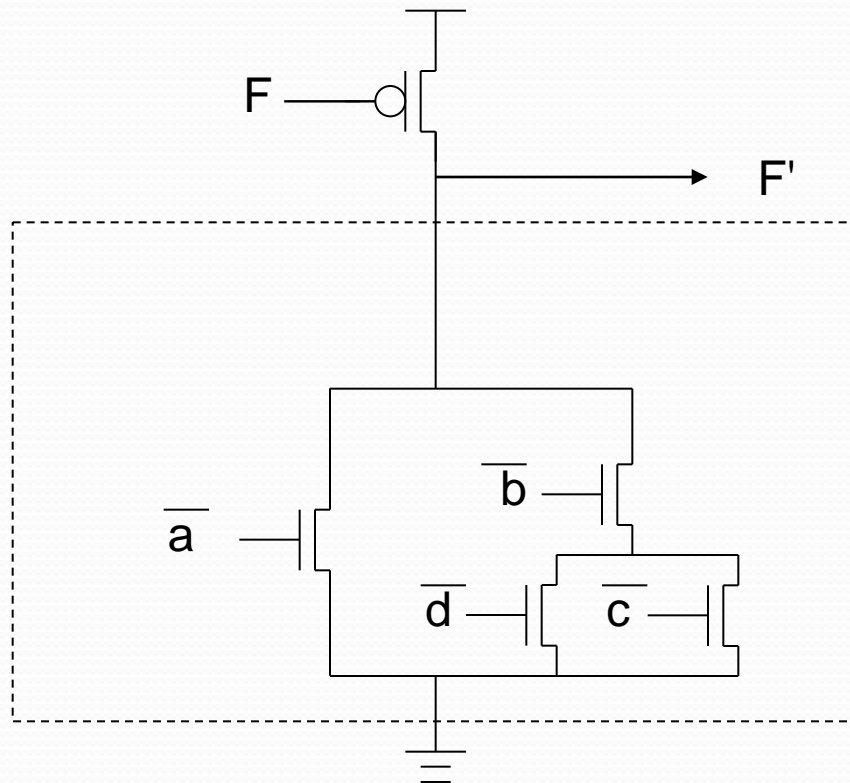
$$F' = a \cdot (b + c \cdot d)$$

- Μπορεί να υλοποιηθεί ως

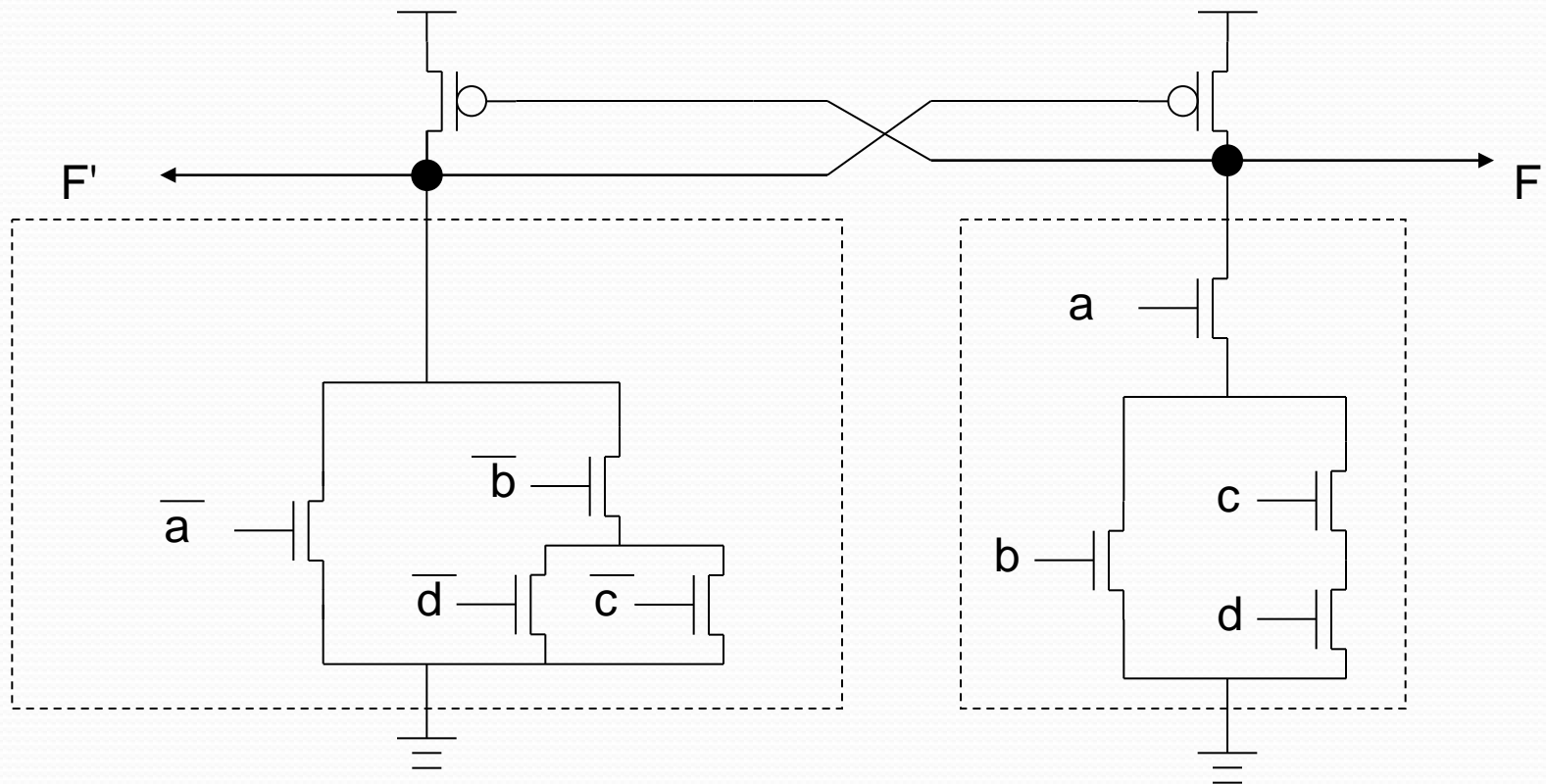


- Κατά αντίστοιχο τρόπο μπορεί να υλοποιηθεί η F' εάν υπάρχει η F

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



Ταυτόχρονη Υλοποίηση



- 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.