



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
Εθνικόν και Καποδιστριακόν  
Πανεπιστήμιον Αθηνών

# Σχεδίαση CMOS Ψηφιακών Ολοκληρωμένων Κυκλωμάτων

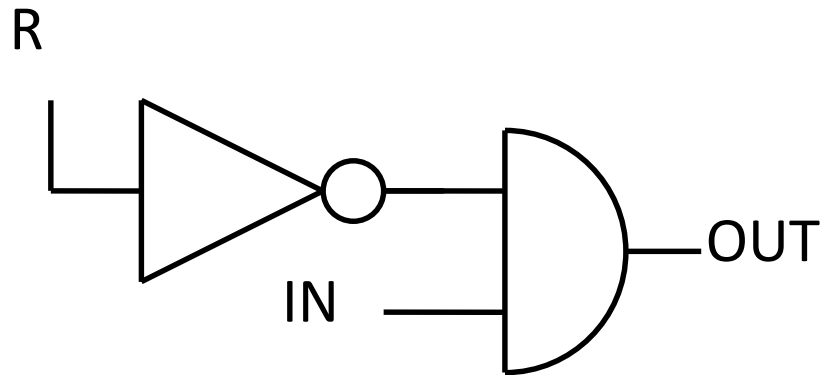
Ενότητα 7: Στοιχεία μνήμης

Αγγελική Αραπογιάννη

Σχολή Θετικών Επιστημών

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

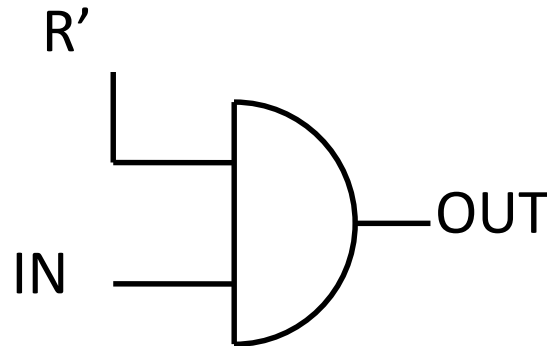
# Η λειτουργία RESET



- Εάν το σήμα R είναι λογικό «1» στην έξοδο έχουμε λογικό «0».
- Εάν το σήμα R είναι λογικό «0» στην έξοδο έχουμε ότι και στην είσοδο



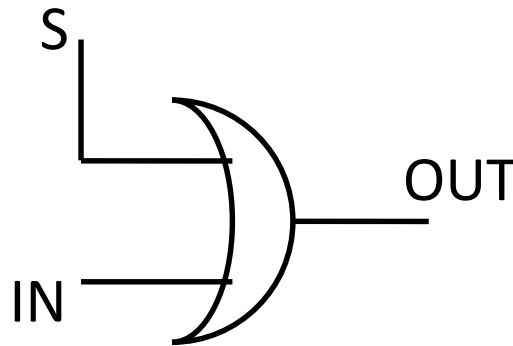
# Εναλλακτική λύση



- Εάν το σήμα  $R'$  είναι λογικό «0» στην έξοδο έχουμε λογικό «0».
- Εάν το σήμα  $R'$  είναι λογικό «1» στην έξοδο έχουμε ότι και στην είσοδο



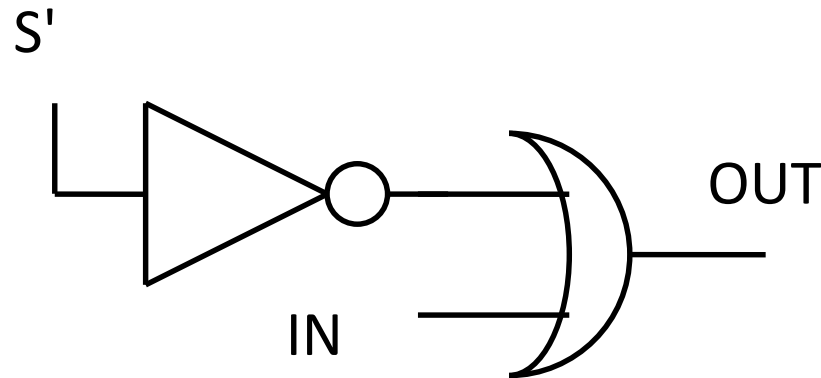
# Η λειτουργία SET



- Εάν το σήμα  $S$  είναι λογικό «1» στην έξοδο έχουμε λογικό «1».
- Εάν το σήμα  $S$  είναι λογικό «0» στην έξοδο έχουμε ότι και στην είσοδο



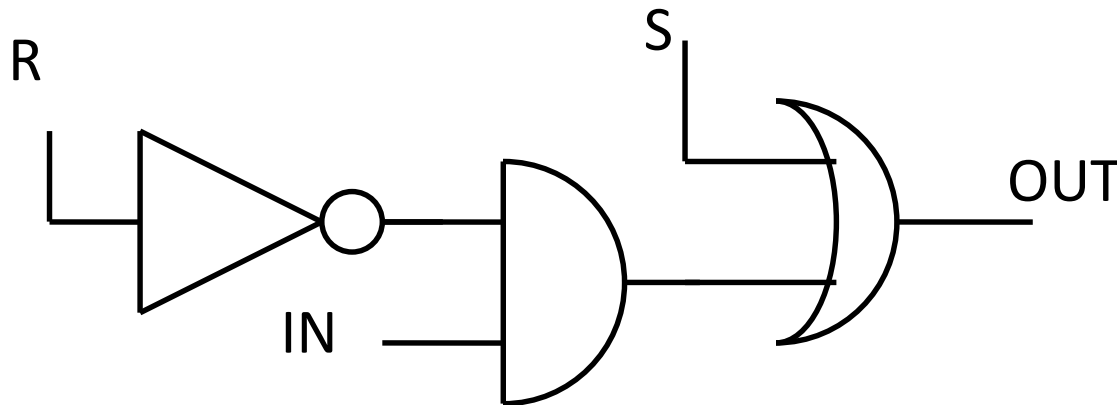
# Εναλλακτική λύση λειτουργίας SET



- Εάν το σήμα  $S'$  είναι λογικό «0» στην έξοδο έχουμε λογικό «1».
- Εάν το σήμα  $S'$  είναι λογικό «1» στην έξοδο έχουμε ότι και στην είσοδο



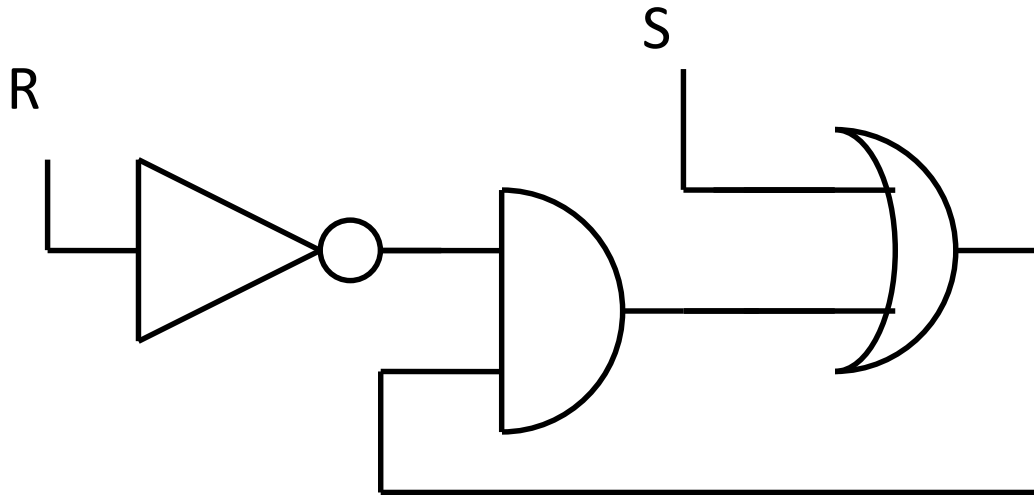
# Ταυτόχρονο SET - RESET



- Με αυτό το κύκλωμα μπορούμε να έχουμε και τις δύο λειτουργίες
- Με  $S=“1”$  η έξοδος είναι λογικό «1»
- Με  $R=“1”$  η έξοδος είναι λογικό «0» (Εάν το  $S$  δεν είναι ενεργοποιημένο)
- Με  $S=R=“0”$  η έξοδος είναι η είσοδος



# Ατέρμων Κύκλος- Μνήμη



- Με αυτό το κύκλωμα έχουμε μνήμη
- Μπορούμε να δίνουμε την τιμή που θέλουμε
- Με  $S=R=“0”$  παραμένει η προϋπάρχουσα τιμή



# Μεταστάθεια – Metastability

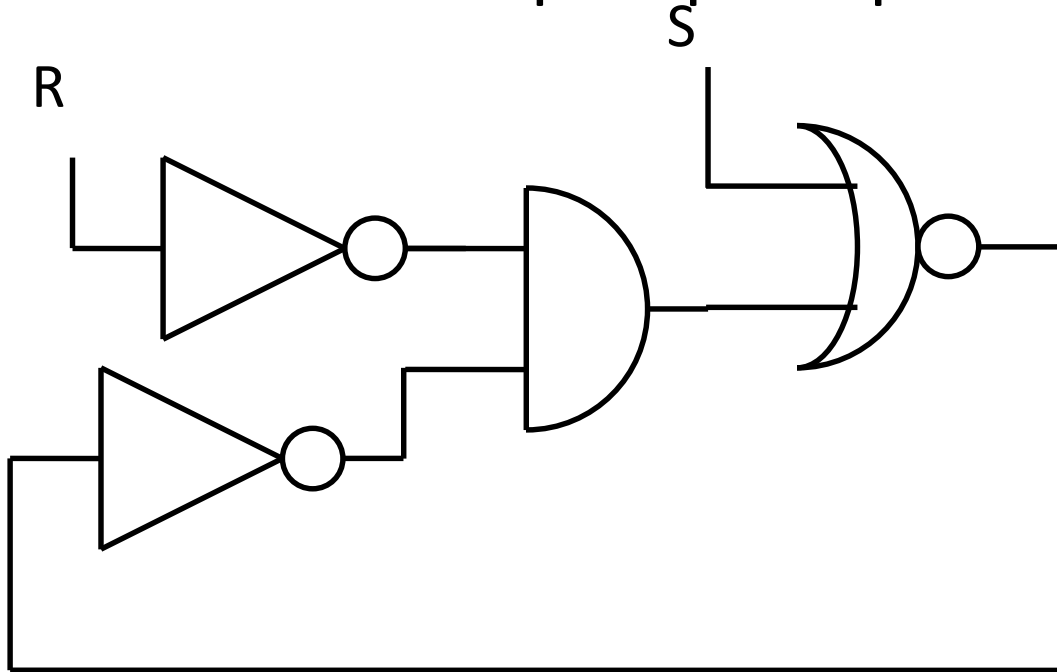
- Υπάρχει πρόβλημα εάν θέσω  $S=R=“1”$ . Τότε δίνω δύο διαφορετικές τιμές στο κύκλωμα μου
- Εάν προσπαθήσω να “αποθηκεύσω” διαφορετικές τιμές θα έχω ταλάντωση
- Το κύκλωμα μετά από αριθμό ταλαντώσεων θα ισορροπήσει σε κάποια τιμή





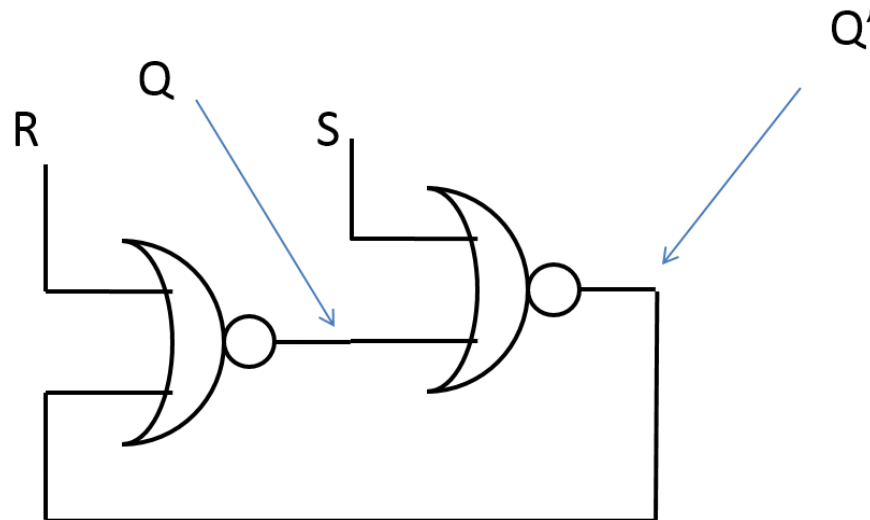
# CMOS Υλοποίηση (1 από 2)

- Θα αντικαταστήσουμε τις χρησιμοποιούμενες πύλες με πύλες NOR
- Αρχικά αντικαθιστούμε την OR με NOR και NOT



# CMOS Υλοποίηση (2 από 2)

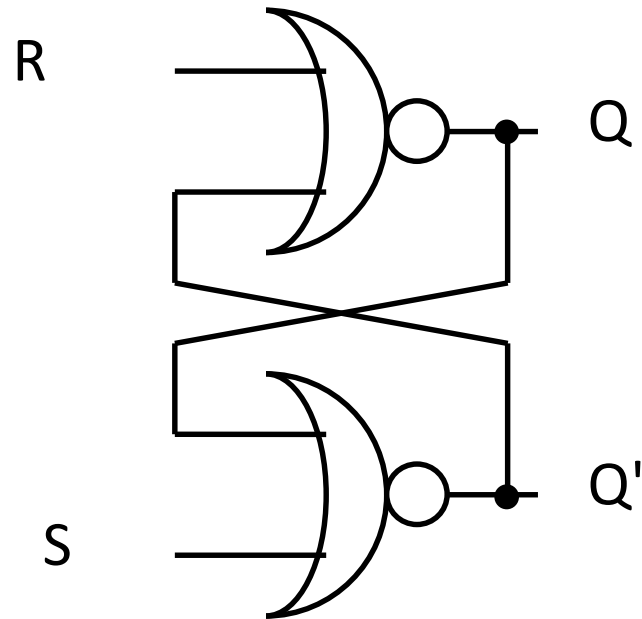
Αντικαθιστώ και τις δύο NOT και την AND πύλη με μία NOR και έχω



Προσοχή! Με αντιστρέψιμες πύλες στον ατέρμων κύκλο έχω Q και Q'



# Εναλλακτική Απεικόνιση

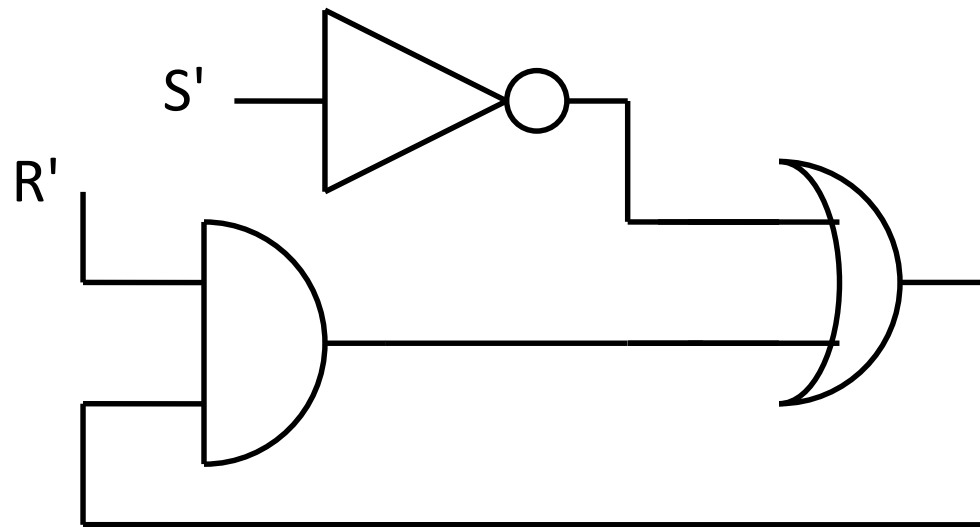


# Λειτουργία

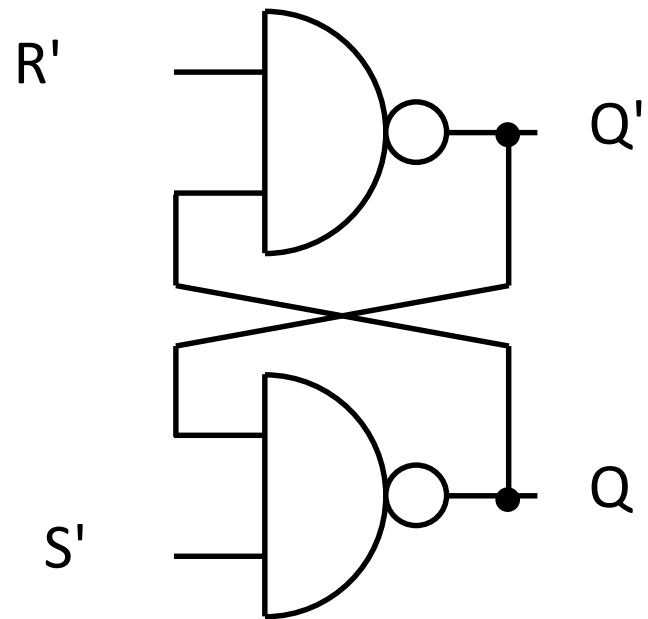
R	S	Λειτουργία
0	0	Μνήμη
0	1	Αποθήκευση 1
1	0	Αποθήκευση 0
1	1	Μεταστάθεια?



# Χρήση $S'$ , $R'$



# Ισοδύναμη CMOS Υλοποίηση



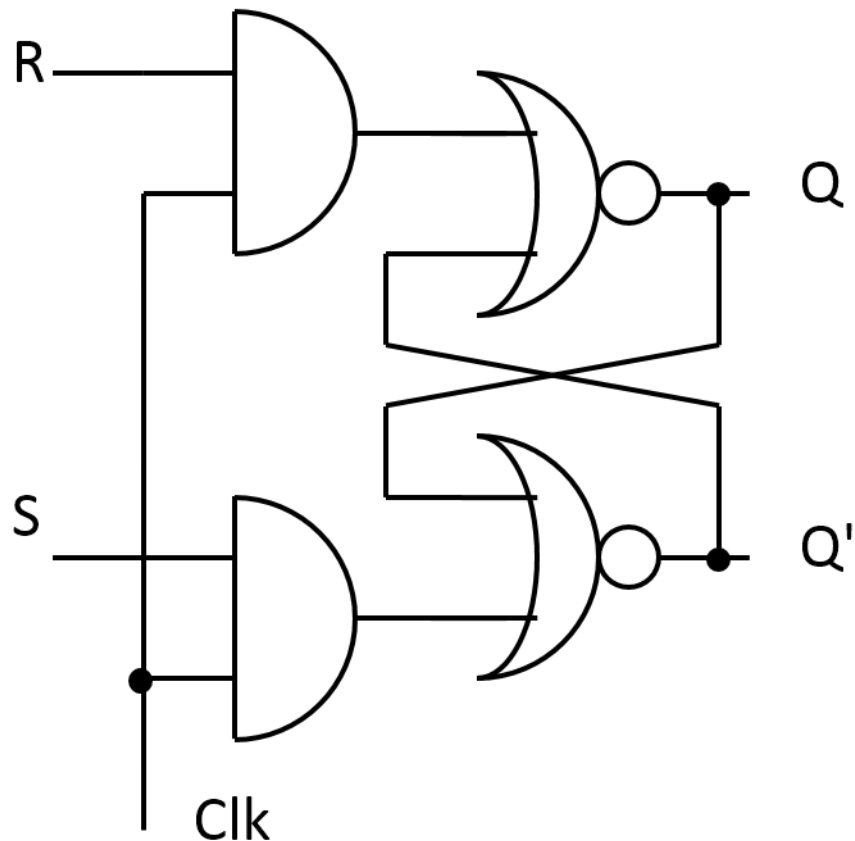
# Χρήση Ρολογιού – Clk (1 από 2)

- Προσθέτουμε σήμα Clk που επιβάλλει κατάσταση μνήμης
- Για παράδειγμα στην υλοποίηση του στοιχείου μνήμης με NOR πύλες
  - Αντί R, έχουμε το λογικό AND του R με το Clk
  - Αντί S, έχουμε το λογικό AND του S με το Clk
- Με Clk λογικό “0” έχουμε μνήμη
- Με Clk λογικό “1” έχουμε την ίδια λειτουργία με το αρχικό κύκλωμα



# Χρήση Ρολογιού – Clk (2 από 2)

CMOS Υλοποίηση με AOI-21 πύλες



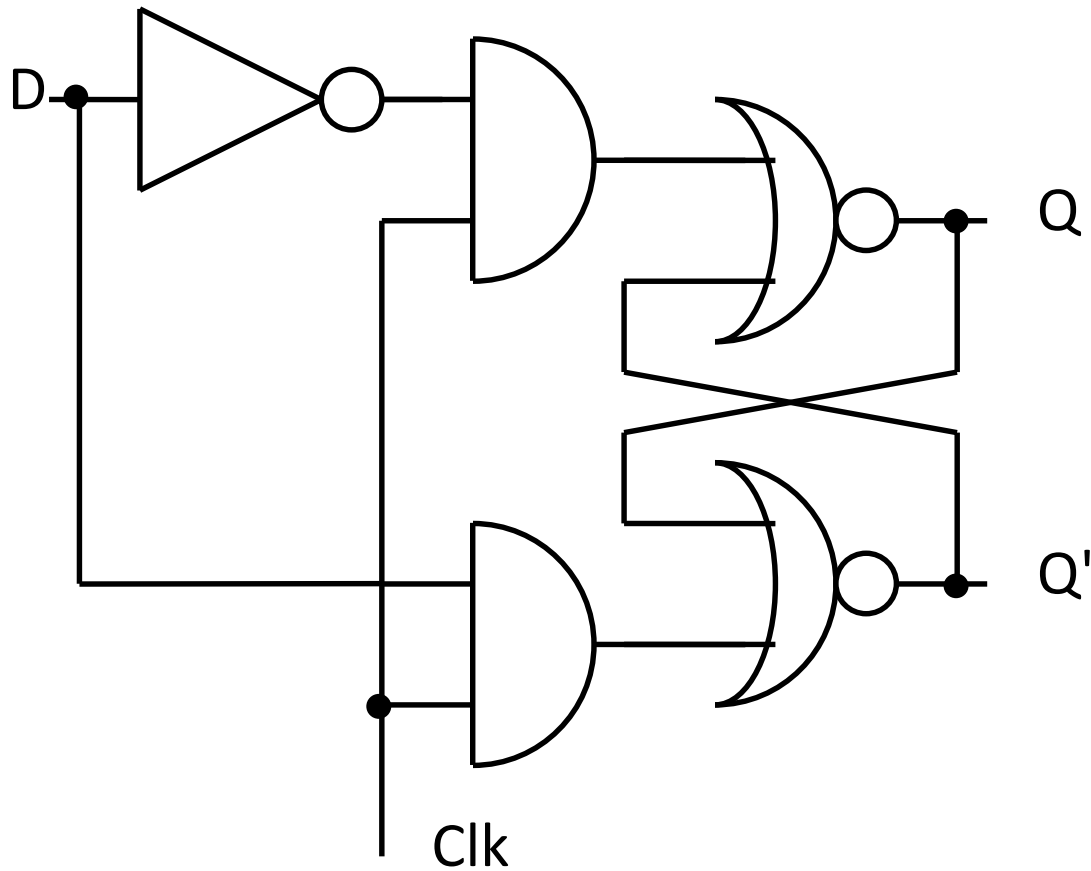


# Αποφυγή Μεταστάθειας

- Για αποφυγή μεταστάθειας αρκεί να εξασφαλίσω ότι ποτέ δεν έχω  $S=R="1"$
- Το εξασφαλίζω εάν  $S$  και  $R$  έχουν πάντοτε αντίθετες τιμές
- Με την αρχική υλοποίηση αυτό δε επιτρέπει να έχω λειτουργία μνήμης ( $S=R="0"$ )
- Εάν υπάρχει  $Clk$  την λειτουργία μνήμης την εξασφαλίζει το  $Clk$ .



# D-Latch ( Μανδαλωτής )



# Εναλλακτική λύση

- Αντικαθιστώντας τις AOI-21 πύλες με OAI-21 πύλες έχουμε μία εναλλακτική υλοποίηση
- Στην περίπτωση αυτή το σήμα Clk' εάν είναι λογικό «1» έχουμε λειτουργία μνήμης



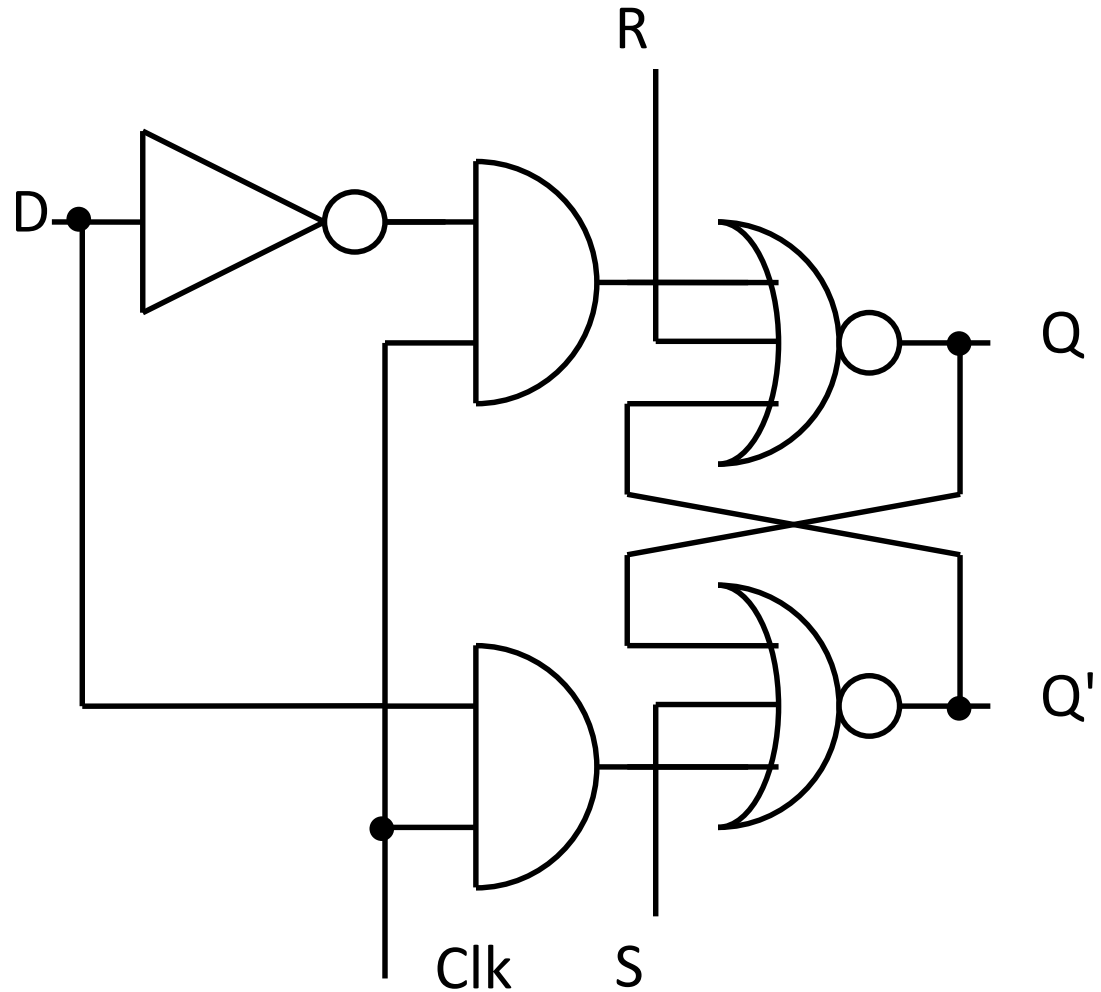
# Ασύγχρονα SET/RESET

- Η σχεδίαση μας μπορεί να απαιτεί την χρήση ασύγχρονων SET/RESET
- Τα σήματα αυτά επιβάλλουν την λειτουργία τους ανεξάρτητα από την τιμή του Clk
- Η τελική σχεδίαση έχει δύο SET
  - Το ένα λειτουργεί ανεξάρτητα από το Clk
  - Το δεύτερο έχει εξαρτώμενη λειτουργία
- Αντίστοιχα για το RESET



# Υλοποίηση (Ασύγχρονα SET/RESET)

- Αντικαθιστώ τις AOI-21 πύλες με AOI-211
- Εναλλακτικά τις OAI-21 με OAI-211

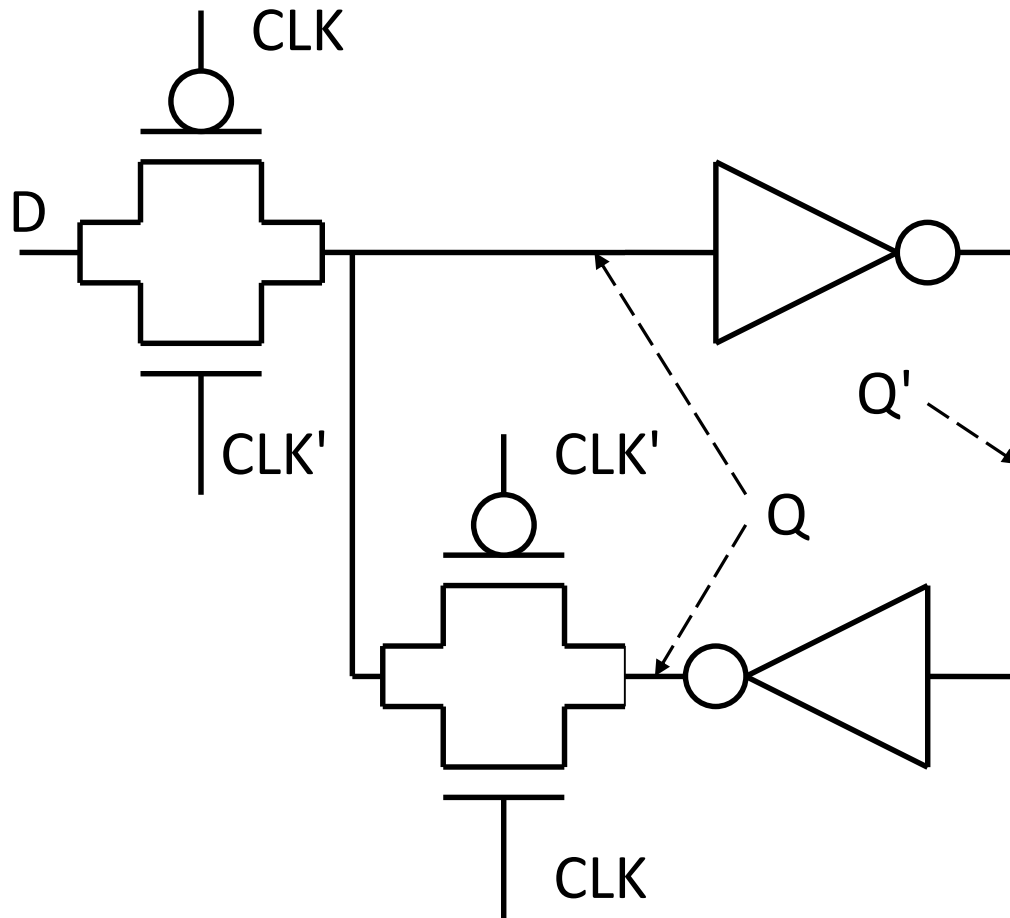


# Υλοποίηση με Πολυπλέκτη (1 από 2)

- Μπορώ να δημιουργήσω έναν ατέρμων κύκλο συνδέοντας την έξοδο ενός πολυπλέκτη σε μία από τις εισόδους του (είσοδος ανάδρασης)
- Με επιλογή της εισόδου ανάδρασης έχω μνήμη
- Με επιλογή άλλης εισόδου εισάγω νέα δεδομένα



# Υλοποίηση με Πολυπλέκτη (2 από 2)



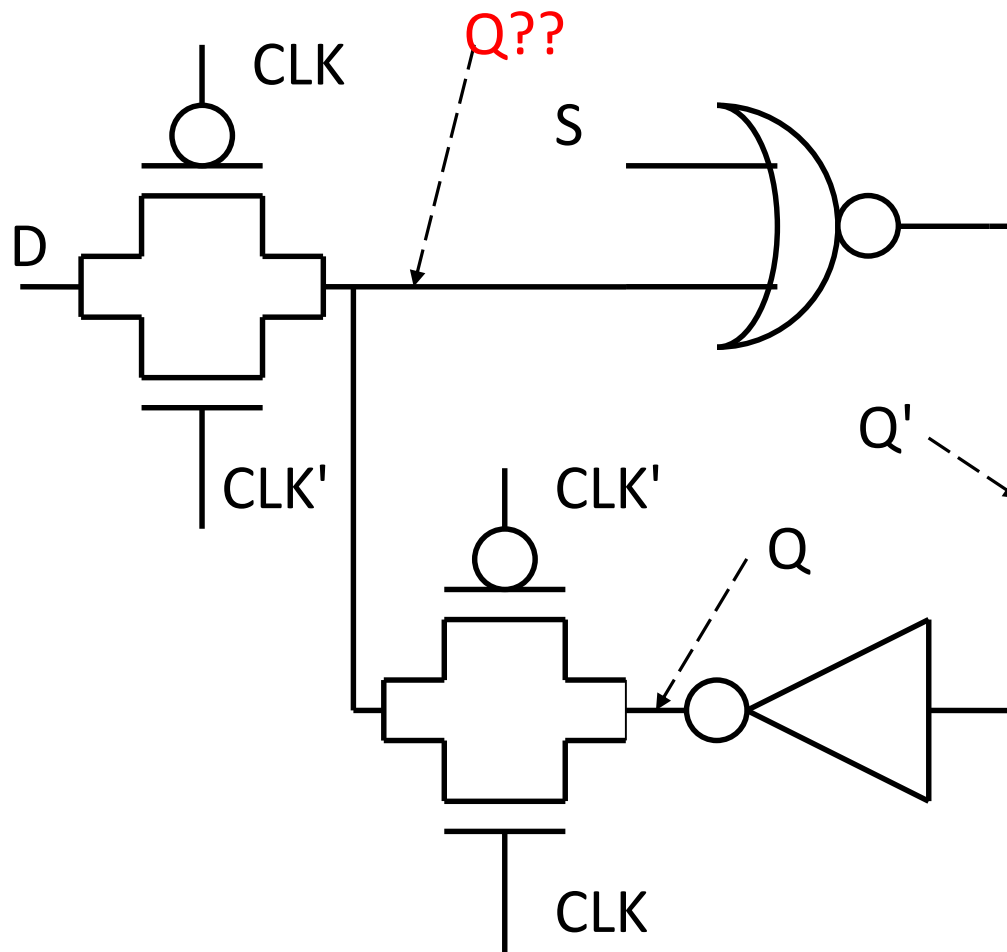
# Ασύγχρονος Έλεγχος

- Αντικαθιστώντας της NOT πύλες με NAND ή NOR μπορώ να έχω ασύγχρονο SET/RESET
- Εάν θέλω ταυτόχρονα SET/RESET μπορώ να αντικαταστήσω την μία για το SET και την άλλη για RESET
- Εναλλακτικά με OAI ή AOI μπορώ να έχω ταυτόχρονα SET/RESET αντικαθιστώντας μία NOT πύλη
- Προσοχή ανάλογα από ποιόν κόμβο παίρνω δεδομένα μπορεί να χρειαστεί SET/RESET στην είσοδο





# Υλοποίηση με Ασύγχρονο SET

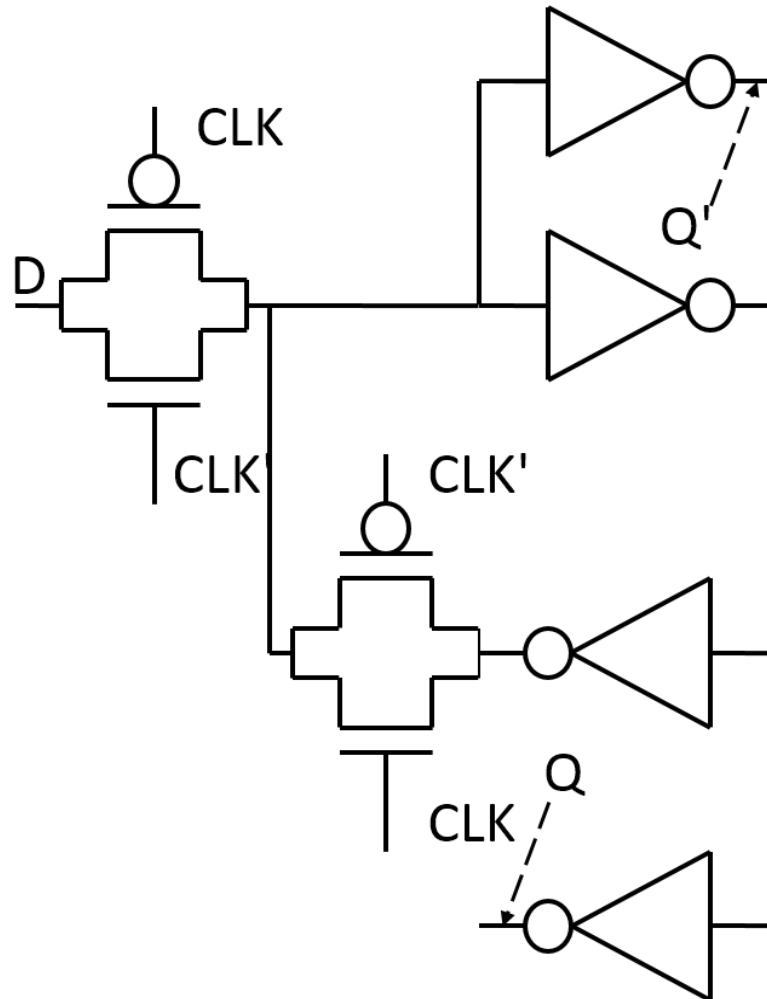


# Επίδραση Εξόδου στα δεδομένα (1 από 2)

- Προσοχή η έξοδος δεν είναι απομονωμένη από τον ατέρμων κύκλο στον οποίο αποθηκεύω δεδομένα
- Άρα η έξοδος μπορεί να συμπεριφερθεί σαν είσοδος
- Για να μην έχω πρόβλημα θα πρέπει να απομονώσω π.χ. με NOT πύλη



# Επίδραση Εξόδου στα δεδομένα (2 από 2)

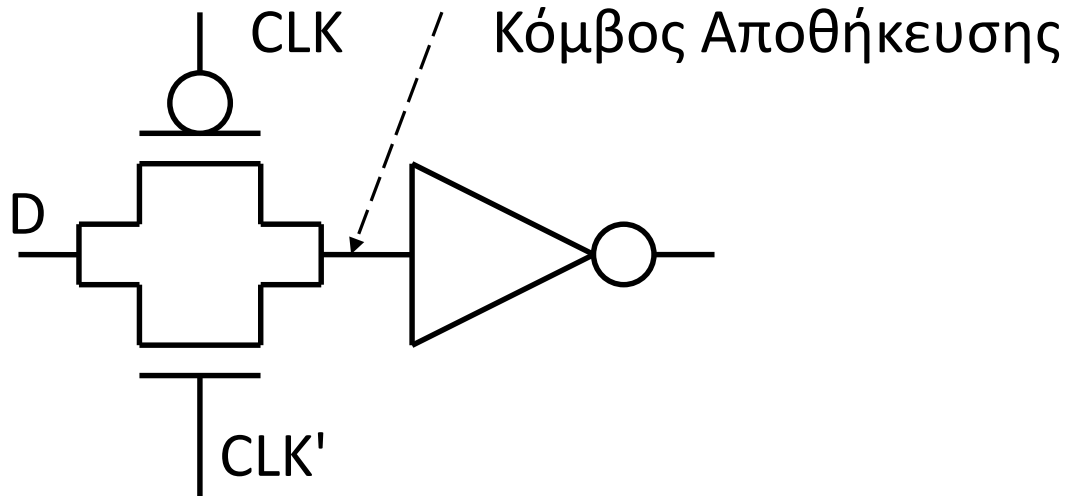


# Εναλλακτικές υλοποιήσεις

- Αντί NOT πύλης σε σειρά με Pass Gate μπορώ να χρησιμοποιήσω
  - Tri-State Buffer
  - Weak NOT



# Δυναμικά Στοιχεία Μνήμης (1 από 2)



- Δεν υπάρχει ατέρμων κύκλος
- Τα δεδομένα αποθηκεύονται σε κόμβο



# Δυναμικά Στοιχεία Μνήμης (2 από 2)

- Απλούστερη υλοποίηση
  - Ταχύτερη λειτουργία συστήματος
  - Χαμηλότερη Κατανάλωση
- Πρόβλημα με το θόρυβο
- Ανικανότητα να διατηρηθούν τα δεδομένα για μεγάλο χρονικά διάστημα (λιγότερο από ms)



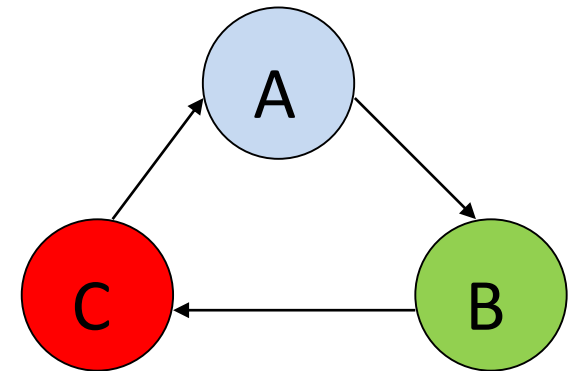
# D-Flip-Flops και Latches

- Το βασικό πρόβλημα με τα Latches είναι ότι δεν μπορούν να εγγυηθούν ότι τα δεδομένα σε μία μηχανή κατάστασης θα αλλάξουν μόνο μία φορά σε κάθε κύκλο ρολογιού
- Εάν υπάρχει ανάδραση μέσα από Latch το πόσες φορές αλλάζουν τα δεδομένα εξαρτάτε από τις καθυστερήσεις στο κύκλωμα και τη διάρκεια του παλμού
  - Δεν υπάρχει εγγύηση ότι η κατάσταση θα αλλάξει μόνο μία φορά

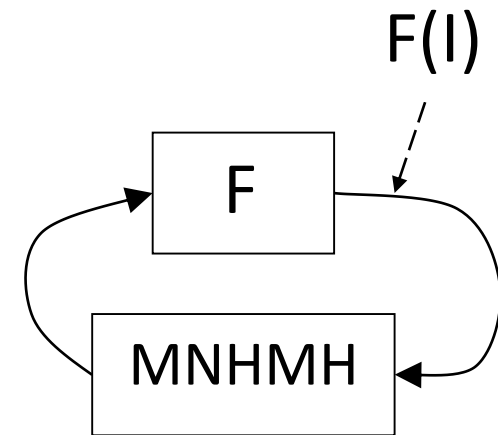


# Παράδειγμα 1 (1 από 2)

- Για να υλοποιήσουμε την μηχανή καταστάσεων του σχήματος χρειαζόμαστε
  - Μνήμη
  - Συνδυαστική λογική για την συνάρτηση  $F$  όπου
    - $A=F(C)$
    - $B=F(A)$
    - $C=F(B)$



Μηχανή Καταστάσεων



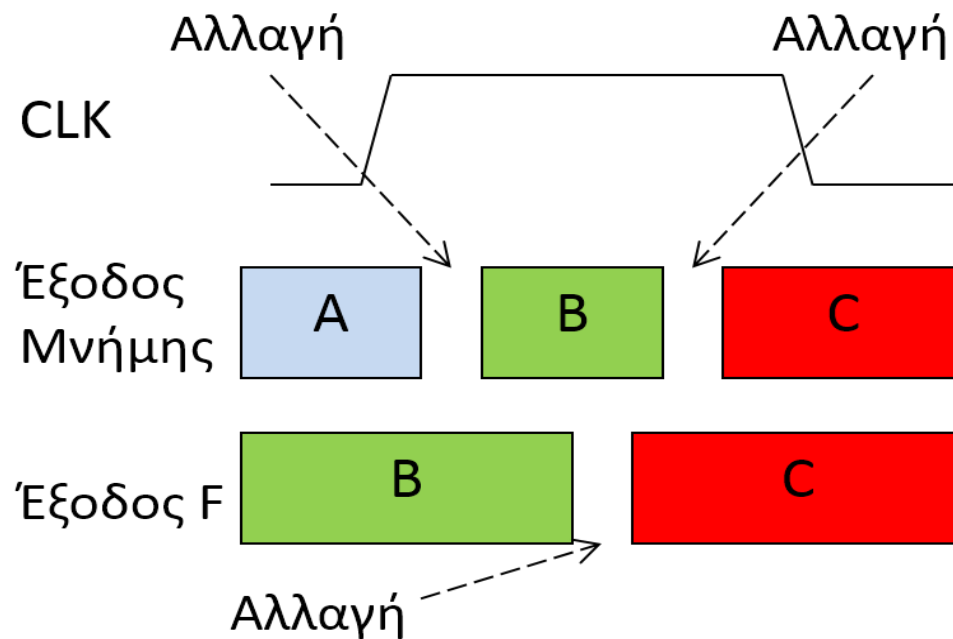
Υλοποίηση





# Παράδειγμα 1 (2 από 2)

Ας υποθέσουμε τώρα ότι η είσοδος περνάει στην έξοδο για  $clk=1$   
Τότε για μία συγκεκριμένη περίοδο ρολογιού μπορώ να έχω το  
ακόλουθο διάγραμμα χρονισμού

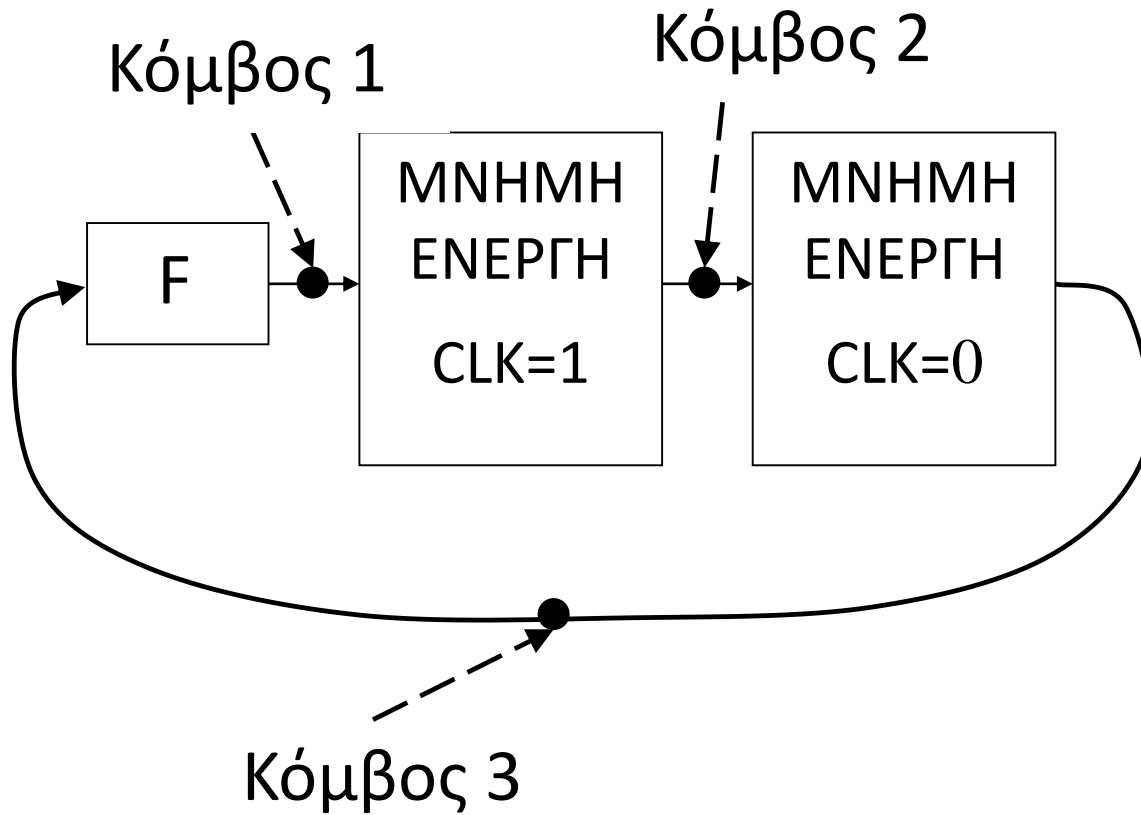


# Λύση D-Flip-Flop (Master-Slave)

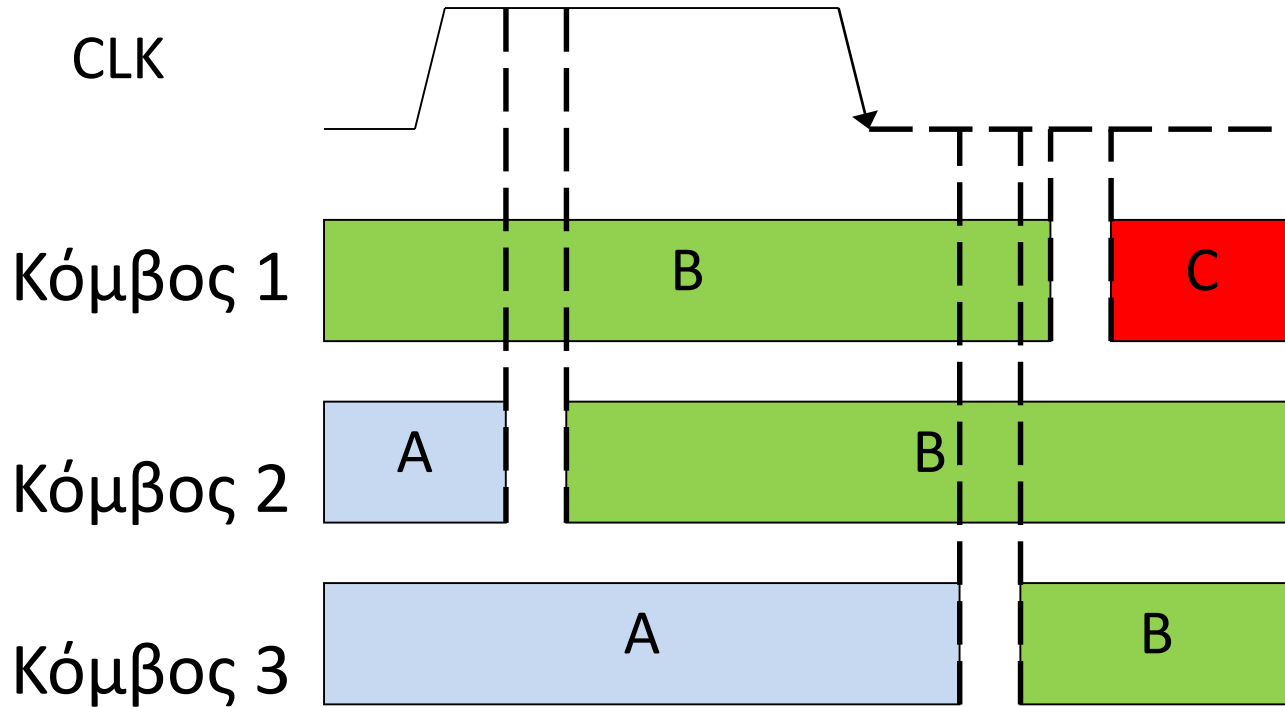
- Δύο επίπεδα μνήμης
- Επιτρέπουν διέλευση για διαφορετικές (αντίθετες) τιμές του ρολογιού Clk
- Διέλευση δεδομένων μία φορά σε κάθε κύκλο του ρολογιού
- Τα νέα δεδομένα εμφανίζονται σε μία καθορισμένη χρονική στιγμή μετά από ακμή του ρολογιού.



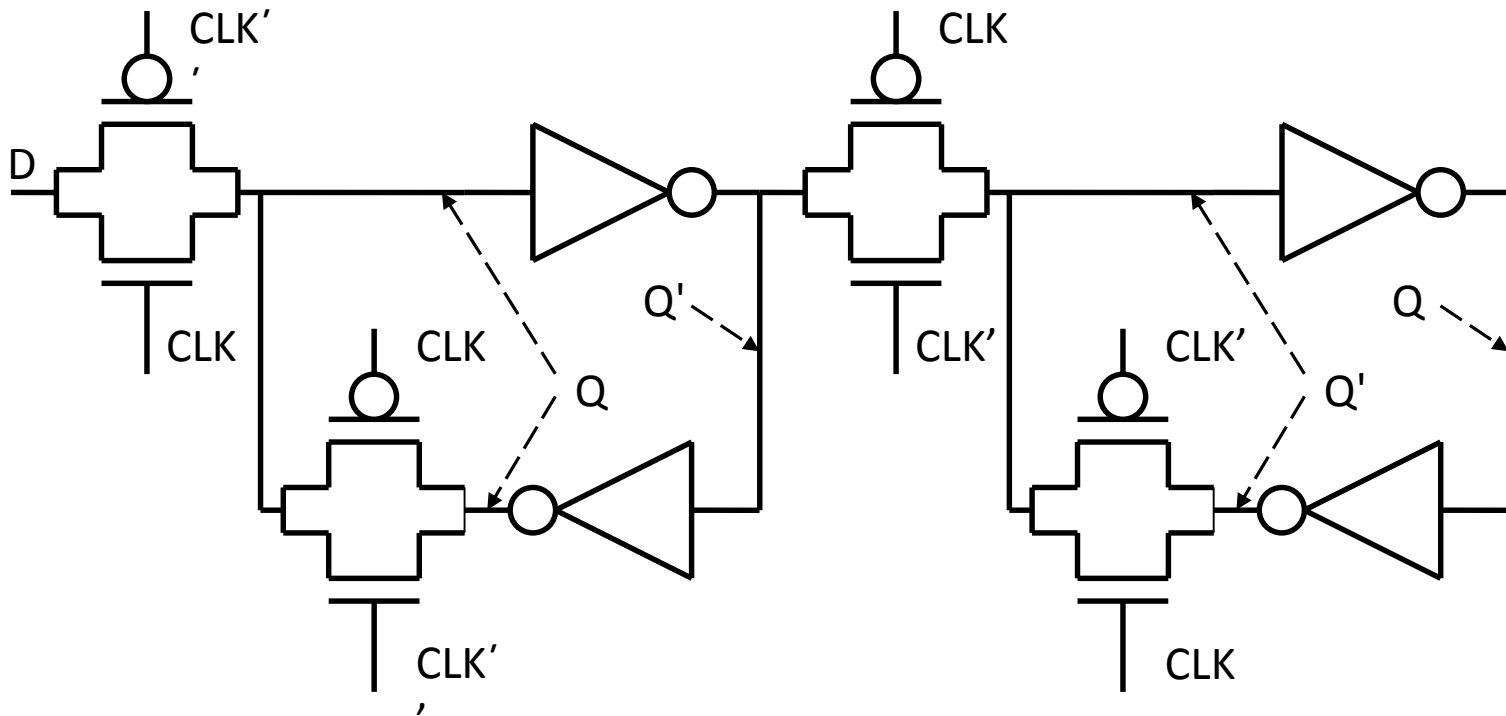
# Υλοποίηση Λύσης



# Χρονισμοί



# Παράδειγμα 2

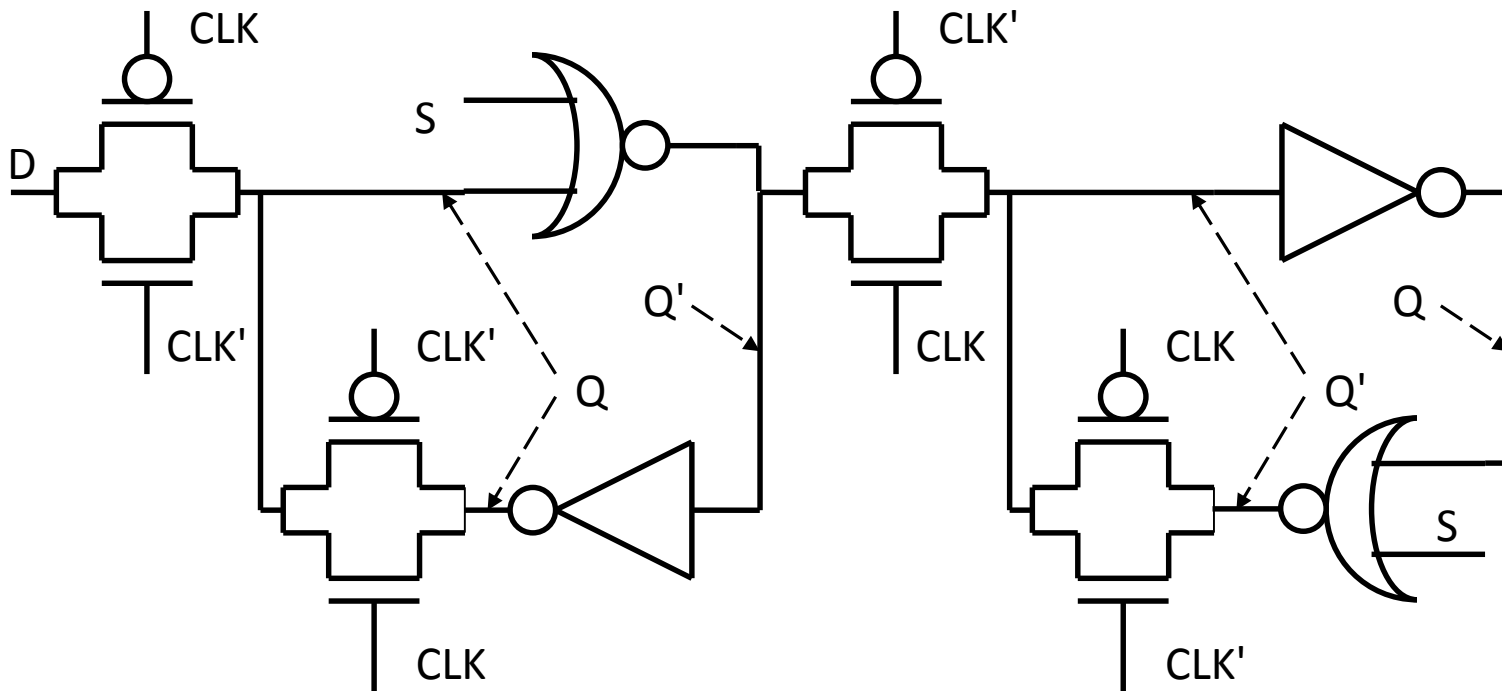


# Ασύγχρονο SET - RESET

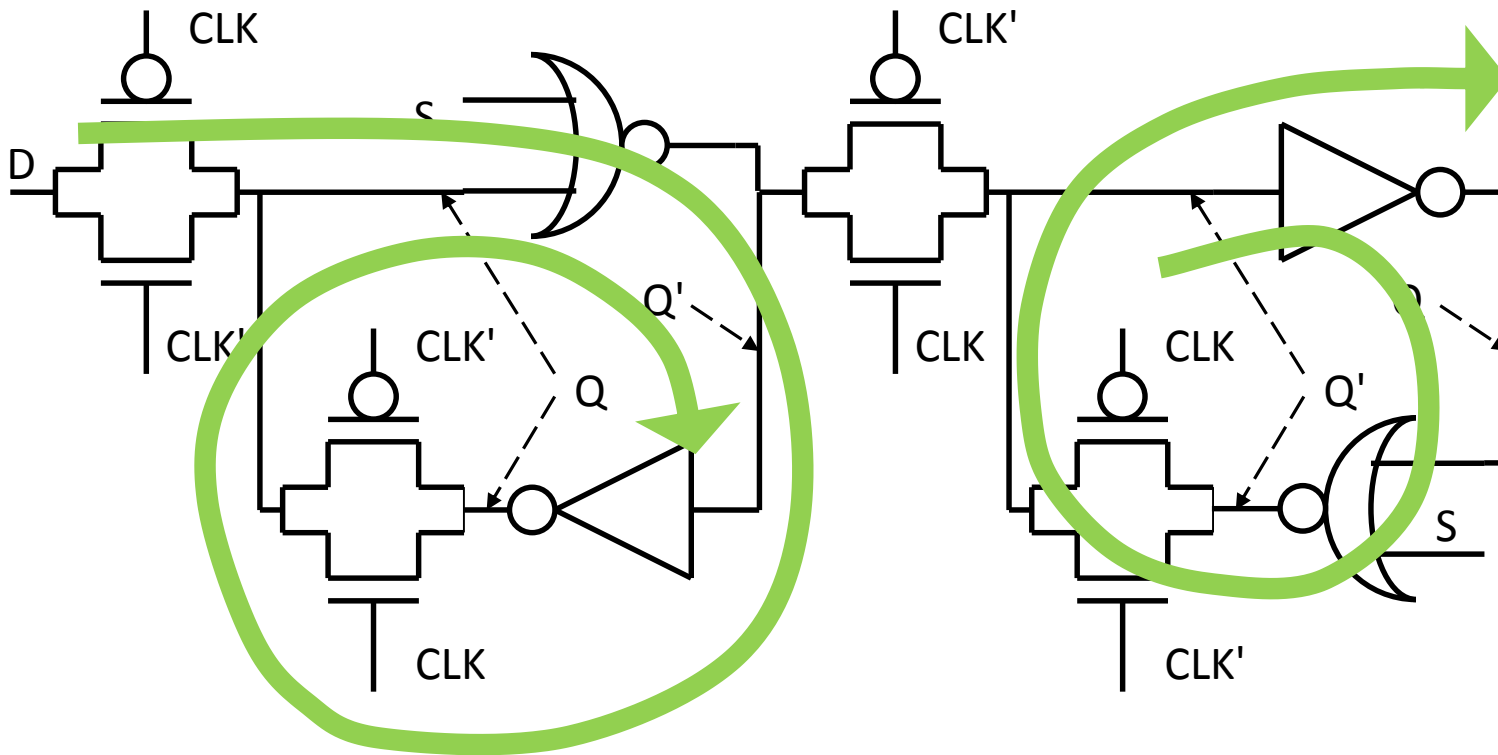
- Σχεδίαση παρόμοια με την περίπτωση του Latch
- Προσοχή θα πρέπει να υπάρχει SET ή/και RESET και στα δύο Latch
  - Εάν μόνο στο πρώτο το ασύγχρονο σήμα μπορεί να μην οδηγεί σε άμεση επίδραση στη έξοδο
  - Μόνο στο δεύτερο, αλλαγή στο CLK μπορεί να οδηγήσει σε « ανεπιθύμητη » αλλαγή στη έξοδο



# Υλοποίηση με ασύγχρονο SET/RESET (1 από 3)

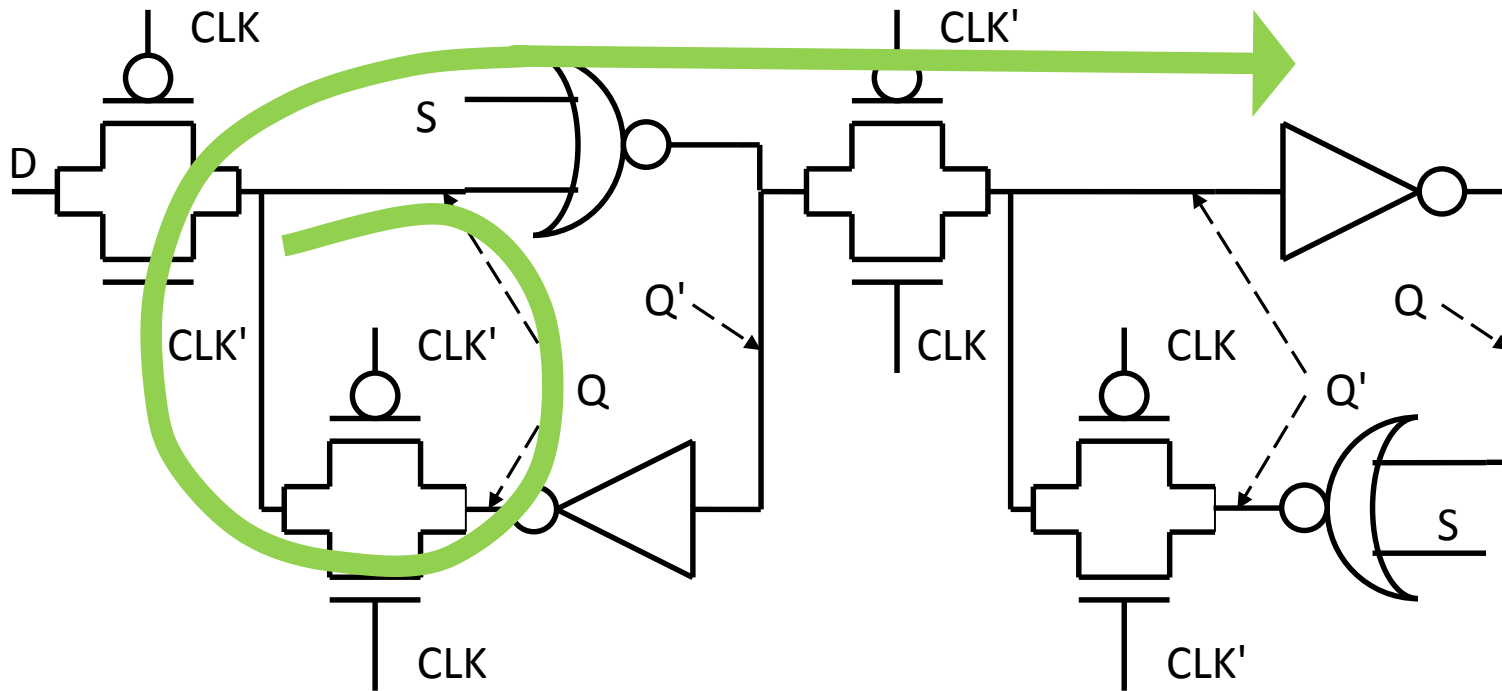


# Υλοποίηση με ασύγχρονο SET/RESET (2 από 3)





# Υλοποίηση με ασύγχρονο SET/RESET (3 από 3)



# Περιορισμοί (1 από 2)

- Το ρολόι και τα δεδομένα είόδου δεν πρέπει να αλλάζουν ταυτόχρονα
  - Hold Time  $T_{hold}$ : Ο χρόνος που απαιτείτε να περάσει από την αλλαγή του ρολογιού μέχρι την αλλαγή των δεδομένων στην είσοδο
  - Setup Time  $T_{setup}$ : Ο χρόνος που πρέπει να περάσει από την αλλαγή των δεδομένων στην είσοδο μέχρι τη αλλαγή του ρολογιού



# Περιορισμοί (2 από 2)

- Επίσης υπάρχει και η καθυστέρηση του στοιχείου  $T_{\text{memory\_delay}}$ 
  - Σε Latch από αλλαγή του σήματος εισόδου ή του ρολογιού μέχρι την αλλαγή στην έξοδο
  - Σε D-flip-flop από την ακμή του ρολογιού μέχρι την αλλαγή στην έξοδο
- Μέγιστη συχνότητα λειτουργίας



# Clock Skew (1 από 3)

- Έστω σύστημα με ρολόι με περίοδο  $P$ , δύο Flip-Flop και συνδυαστική λογική ανάμεσα τους.
- Θεωρούμε ότι η μέγιστη καθυστέρηση στη συνδυαστική λογική είναι  $T_{max}$  και η ελάχιστη  $T_{min}$
- Υποθέτουμε κοινό ρολόι που φτάνει με καθυστέρηση  $T_1$  στο πρώτο FF και  $T_2$  στο δεύτερο. (Εάν  $T_1$  η  $T_2$  αρνητικό τότε φτάνει νωρίτερα)



# Clock Skew (2 από 3)

- Για το πρώτο FF έχουμε  $T_{hold1}$ ,  $T_{setup1}$  και  $T_{memory\_delay1}$
- Αντίστοιχα για το δεύτερο έχουμε  $T_{hold2}$ ,  $T_{setup2}$  και  $T_{memory\_delay2}$
- Εάν  $T_{memory\_delay1} + T_1 + T_{max} + T_{setup2} - T_2 > Period$ ,  
έχουμε Zero Clocking,
  - Τα δεδομένα δεν μπορούνε να φράσουν εγκαίρως στην είσοδο του δεύτερου Flip-Flop
  - Η λύση είναι η αύξηση της περιόδου, άρα η μείωση της συχνότητας λειτουργίας



# Clock Skew (3 από 3)

- Εάν  $T_{\text{memory\_delay1}} + T1 + T_{\text{min}} < T_{\text{Hold2}} + T2$ , τότε έχουμε double clocking
  - Τα δεδομένα μπορούν να περάσουν σε μία περίοδο του ρολογιού από δύο διαδοχικά Flip-Flop
  - Το πρόβλημα λύνεται με αύξηση του  $T1$  ως προς το  $T2$ , (αυτό μπορεί να οδηγήσει σε Zero Clocking)
- Σε pipelines με κατάλληλη καθυστέρηση του ρολογιού στα στοιχεία μνήμης μπορούμε να «δανειζόμαστε» χρόνο από ένα επίπεδο (που δεν τον χρειάζεται) και να τον χρησιμοποιούμε σε άλλο βελτιστοποιώντας (μειώνοντας) την περίοδο.



# 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. Στοιχεία μνήμης.». 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.

