Ψηφιακή Σχεδίαση

Ενότητα 7: Μανδαλωτές SR, S΄R,΄D Flip-Flops ,Αφέντη Σκλάβου, Σχεδιασμός Ακολουθιακών κυκλωμάτων, Πίνακας Καταστάσεων, Διάγραμμα Καταστάσεων

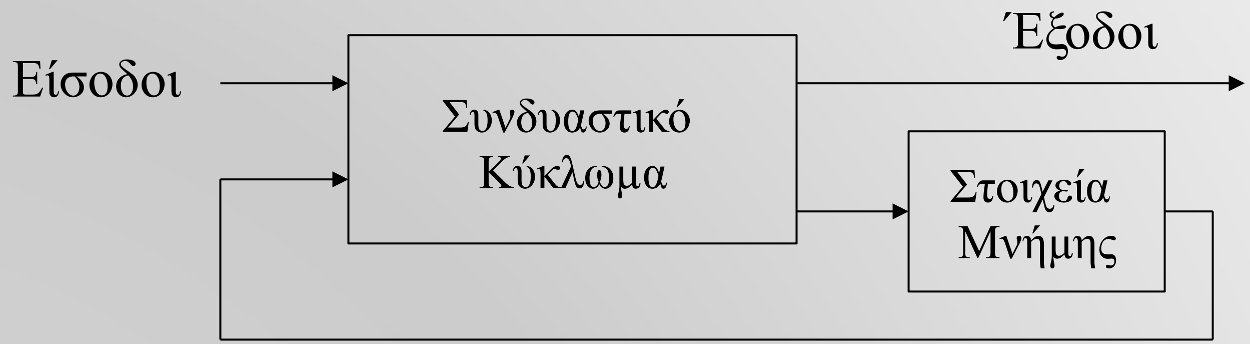
Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Σκοπός της ενότητας

- Να γίνει ανάλυση των Μανδαλωτών SR, Flip Flops (S’ R’ D).
- Να γίνει ο σχεδιασμός ακολουθιακών κυκλωμάτων.
- Να γίνει ανάλυση των πινάκων καταστάσεων και διαγραμμάτων καταστάσεων.

Ακολουθιακά Κυκλώματα

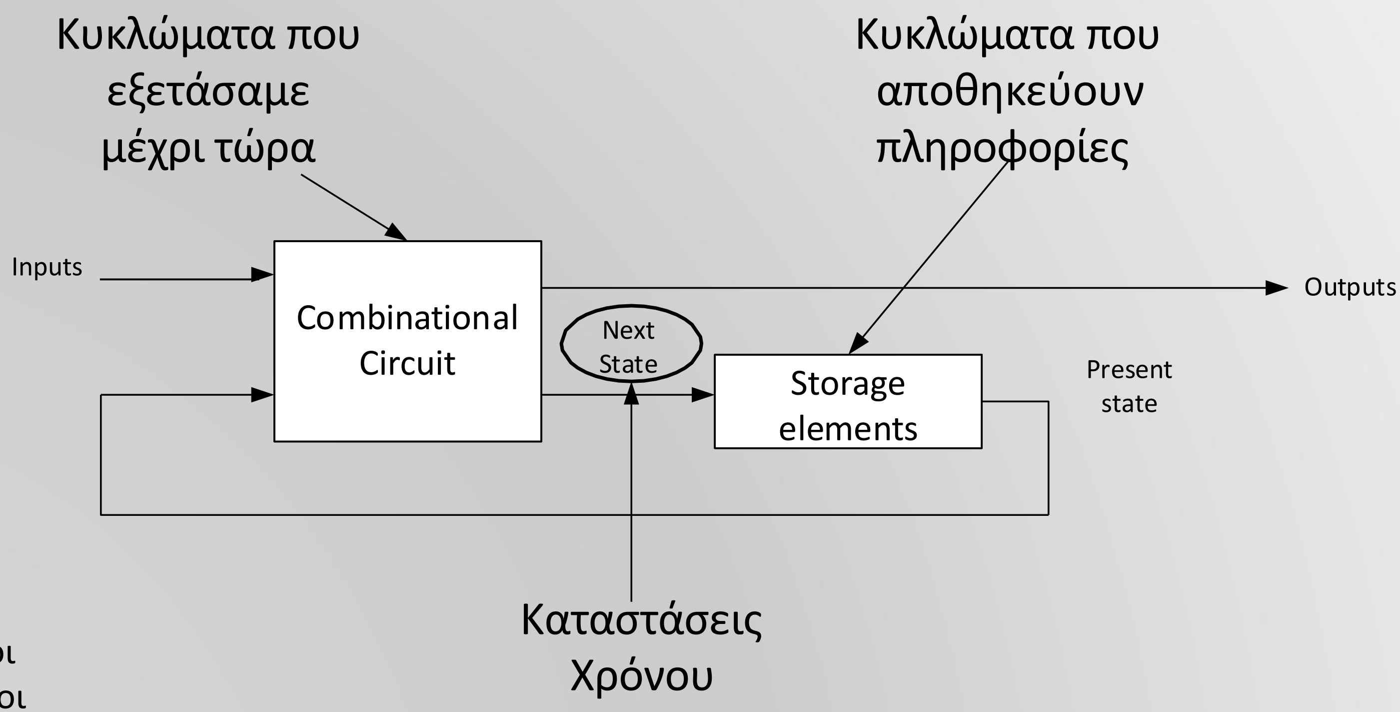


- Κατάσταση Ακολουθιακού Κυκλώματος : περιεχόμενα στοιχείων μνήμης.
- Η έξοδος εξαρτάται από τις εισόδους και την κατάσταση του κυκλώματος.
- Η κατάσταση εξαρτάται από τις εισόδους και την προηγούμενη κατάσταση.

Ακολουθιακά Κυκλώματα

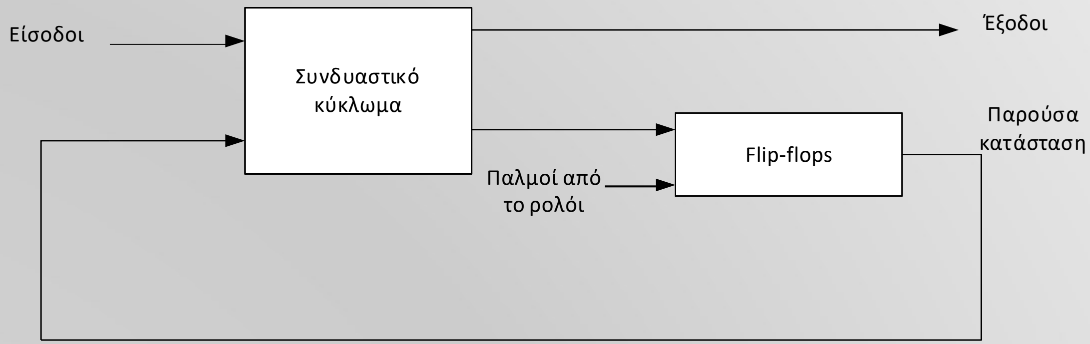
- Σύγχρονα: οι τιμές των σημάτων του αλλάζουν σε διακριτές χρονικές στιγμές (ρολόι).
- Ασύγχρονα: οι τιμές των σημάτων του αλλάζουν σε οποιαδήποτε χρονική στιγμή (συνδυαστικά κυκλώματα με ανάδραση).

Ακολουθιακά Κυκλώματα

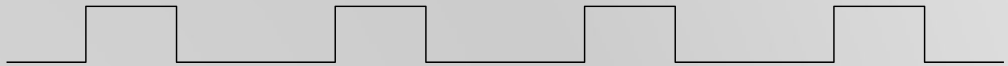


Inputs: Είσοδοι
Outputs: Έξοδοι
Combinational Circuit: Συνδυαστικό κύκλωμα
Next state: Επόμενη κατάσταση
Storage elements: Στοιχεία αποθήκευσης
Present state: Παρούσα κατάσταση

Σύγχρονα ακολουθιακά κυκλώματα



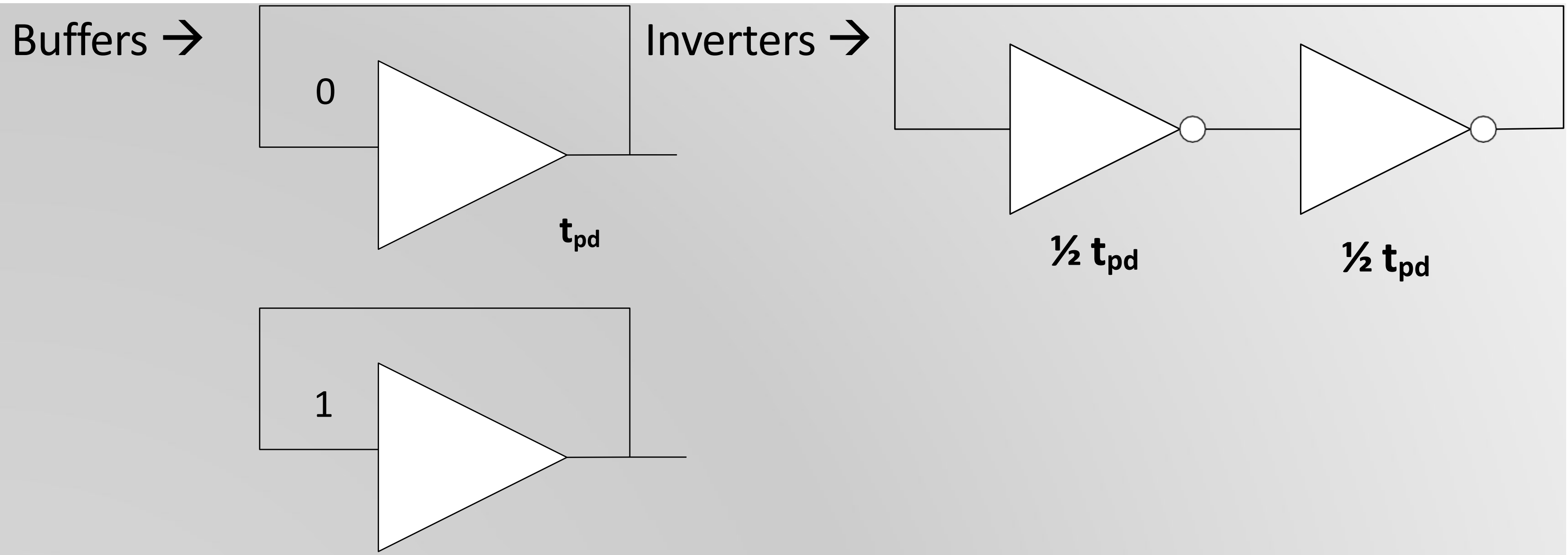
(α) Σχηματικό διάγραμμα



(β) Χρονικό διάγραμμα από τους παλμούς του ρολογιού

- Τα flip-flops έχουν ως εισόδους σήματα από το συνδυαστικό κομμάτι του κυκλώματος καθώς και σήμα από ένα ρολόι με περιοδικούς παλμούς μεταξύ αμετάβλητων περιοδικών διαστημάτων.

Στοιχεία Μνήμης



Η αποθηκευμένη τιμή δεν μπορεί να αλλάξει.

Προσομοίωση Διακριτών Γεγονότων

- Χρησιμοποιείται για την καλύτερη κατανόηση της χρονικής συμπεριφοράς ενός κυκλώματος.
- Κανόνες:
  1. Οι πύλες μοντελοποιούνται με 2 τρόπους:

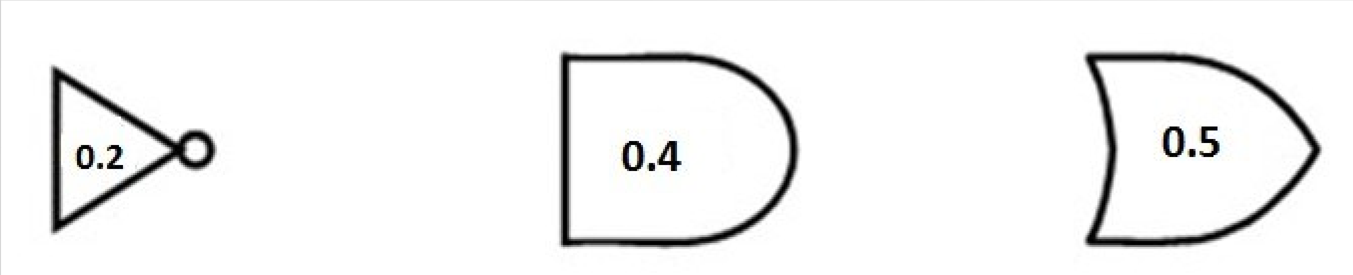
     α) Βάσει λειτουργίας τους, με μηδενική καθυστέρηση (ideal, instantaneous)

     β) Με σταθερή καθυστέρηση ανα πύλη (fixed gate delay).
  2. Κάθε αλλαγή στις τιμές εισόδων αξιολογείται, βάσει του μοντέλου μηδενικής καθυστέρησης, για να υπολογιστούν τυχόν αλλαγές στις τιμές εξόδων (= γεγονος) (evaluation event).
  3. Αλλαγές στις τιμές εξόδων προγραμματίζονται βάσει του μοντέλου σταθερής καθυστέρησης (scheduling event).
  4. Οι τιμές εξόδων (και πιθανόν άλλα επηρεαζόμενα σήματα) αλλάζουν μόνο στον χρόνο του προγραμματιζόμενου γεγονότος.

Μοντέλο Καθυστέρησης Πυλών

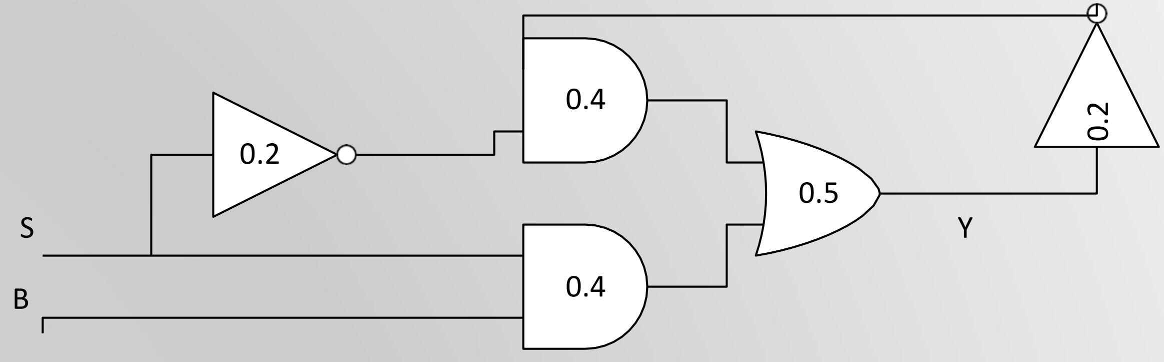
•Θεωρείστε τις πιο κάτω πύλες NOT, AND και OR με καθυστέρηση *n* ns,όπου

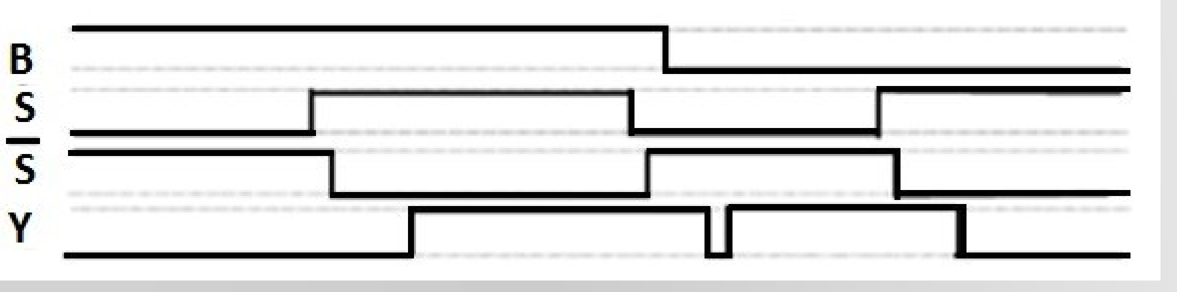
n = 0.2ns, n = 0.4ns και n = 0.5 ns αντίστοιχα:



Μοντέλο Καθυστέρησης Κυκλώματος

- Θεωρείστε έναν απλό 2-σε-1 MUX:
- Με συνάρτηση:
  - Y = A για S = 0
  - Y = B για S = 1

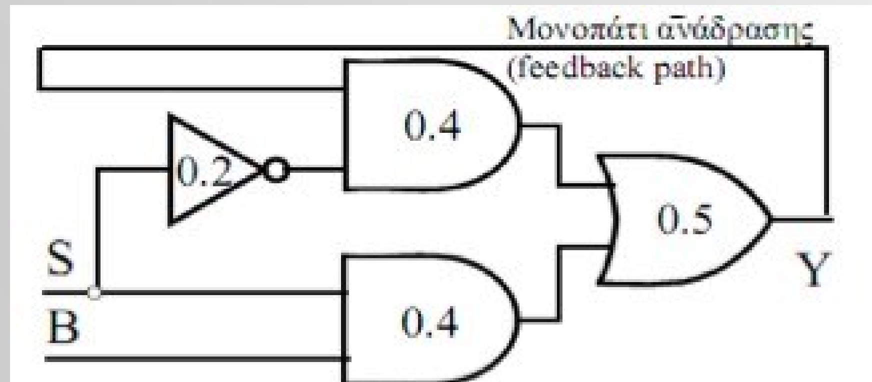


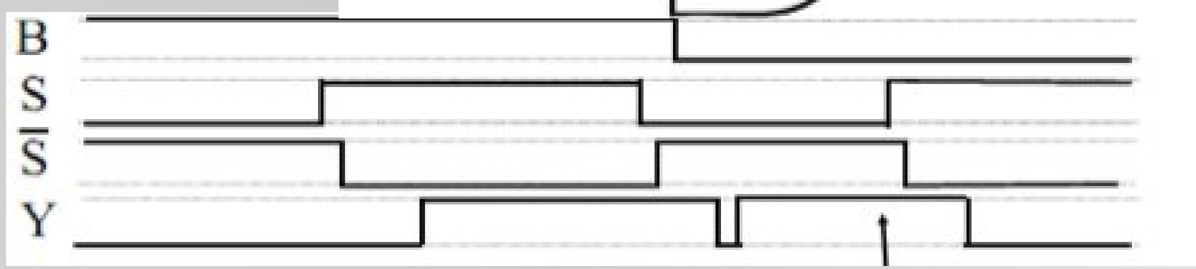


Αποθήκευση Κατάστασης (1)

- Τι γίνεται αν η Α ενωθεί με την Υ;
- Συναρτήσεις:
  - Y = B για S = 1
  - Y(t) εξαρτάται

    Από Y(t - 0. 9) για S = 0





- Το συνδυαστικό κύκλωμα μετατράπηκε σε ακολουθιακό, αφού η συνάρτηση εξόδου εξαρτάται και από τις προηγούμενες τιμές εισόδων.

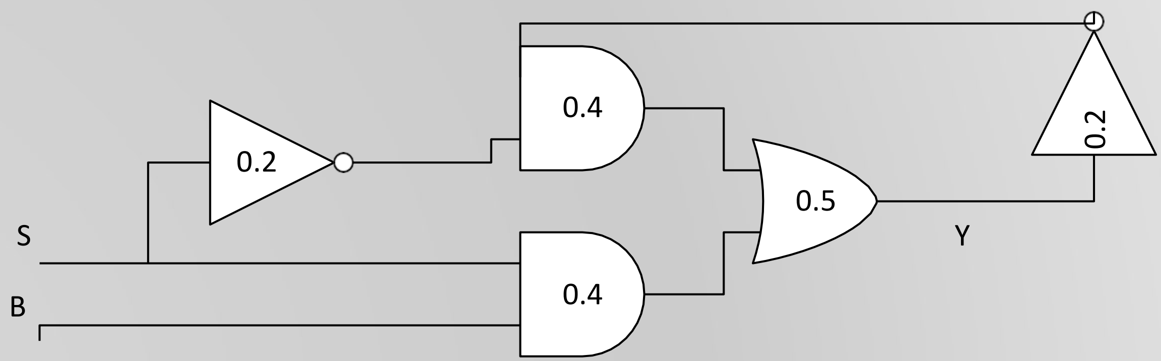
Αποθήκευση Κατάστασης (2)

- Παράδειγμα προσομοίωσης: Οι τιμές αλλάζουν με την πάροδο του χρόνου. Οι αλλαγές σημειώνονται κάθε 100 ns, έτσι ώστε καθυστερήσεις σε δέκατα ns αγνοούνται.

B	S	Y	Σχόλια
1	0	0	Y «Θυμάται» 0
1	1	1	Y = B όταν S = 1
1 0	0 0	1 1	Τώρα Y «θυμαται» B = 1 για S = 0 Καμία αλλαγή για Y όταν αλλάζει το B
0	1	0	Y = B όταν S = 1
0	0	0	Y «Θυμάται» B = 10 για S = 1
1	0	0	Καμία αλλαγή για Y όταν αλλάζει το B

Αποθήκευση Κατάστασης (3)

- Θεωρήστε ότι τοποθετούμε ενα αντιστροφέα στο μονοπάτι ανάδρασης.
- Συμβαίνουν τα ακόλουθα:
  1. Το κύκλωμα γίνεται **ασταθές (unstable).**
  2. Για S = 0 το κύκλωμα γίνεται **ταλαντωτής (oscillator)**. Μπορεί να χρησιμοποιηθεί ως ένα αδρό ρολόι.



Αποθήκευση Κατάστασης (4)

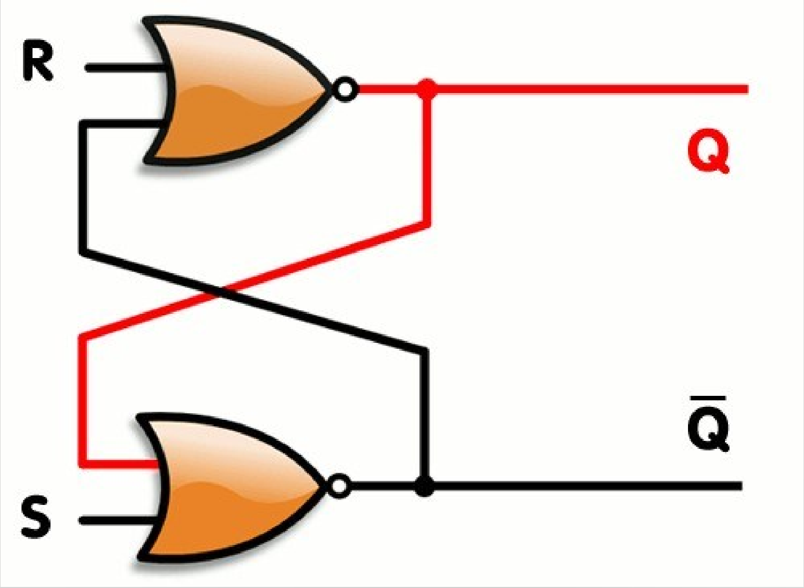
B	S	Y	Σχόλια
0	1	0	Y = B όταν S = 1
1 1	1 0	1 1	Τώρα Y «θυμάται» B
1	0	0	Y, 1.1 ns αργότερα
1	0	1	Y, 1.1 ns αργότερα
1	0	0	Y, 1.1 ns αργότερα

Μανδαλωτές

- Διατηρεί τη δυαδική κατάσταση του επ' αόριστον.
- Οι βασικές διαφορές στα flip-flops εντοπίζονται:
    1. Αριθμό εισόδων.
    2. Τρόπο με τον οποίο οι είσοδοι επηρεάζουν τη δυαδική κατάσταση.
- Οι πλέον στοιχειώδεις τύποι flip-flop δουλεύουν με επίπεδα σημάτων και ονομάζονται μανδαλωτές.

Μανδαλωτής τύπου SR (1)

- SR: “set - reset”, δισταθές στοιχείο με 2 εισόδους. Προσέξτε την «ακαθόριστη» τιμή για S = R = 1.
- Διαβάζοντας τη λογική:
  - Q = (R + Q′) και Q′ = (S + Q)′



Μανδαλωτής τύπου SR (2)

S	R	Q	Q	
1	0	1	0	Set State
0	0	1	0	Set State
0	1	0	1	Reset State
0	0	0	1	Reset State
1	1	x	x	Undefined

Πρόβλημα όταν S = R = 1

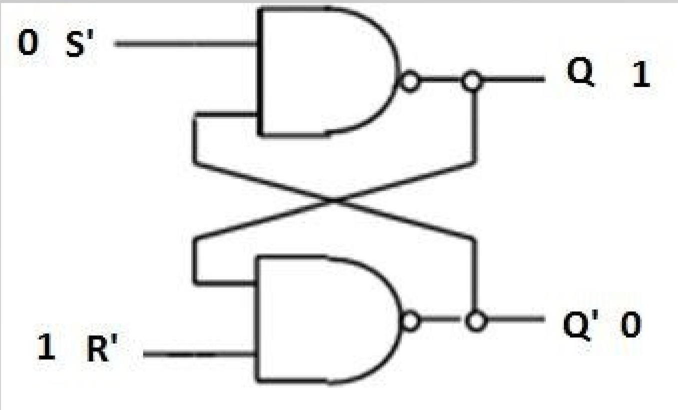
- Ακαθόριστη έξοδος γιατί:
  - Όταν S = R = 1, τότε και οι 2 έξοδοι γίνονται 0.
  - Εάν και οι 2 έξοδοι είναι 0,η κατάσταση του SR latch εξαρτάται από την είσοδο που παραμένει στην τιμή 1 για περισσότερο χρόνο πριν γίνει 0.
  - Άρα είναι όντως, «ακαθόριστη» κατάσταση

    → ΠΡΕΠΕΙ να αποφευχθεί.

Λειτουργία SR latch

- Σε κανονική λειτουργία κρατάμε και τις δύο εισόδους του μανδαλωτή στο 0.
- Αν και οι 2 είσοδοι είναι στο 0, τότε τότε διατηρείται η προηγούμενη κατάσταση.
- Αν επιθυμούμε να αλλάξουμε την κατάσταση τότε δίνουμε είτε S = 1,R = 0 είτε S = 0, R = 1.
- Αν S = R = 0 τότε ο μανδαλωτής μπορεί να είναι είτε σε 'θέση' είτε σε 'επαναφορά‘.

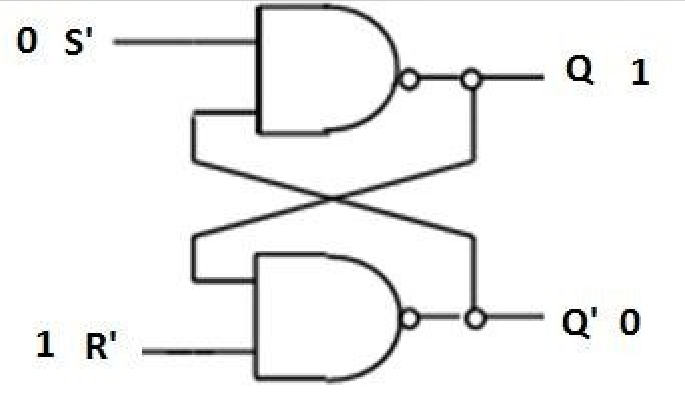
Κατασκευή S΄R΄ latch (NAND)(1)



S’	R’	Q	Q’	
0	0			
0	1	1	0	**Set**
1	0			
1	1			

X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

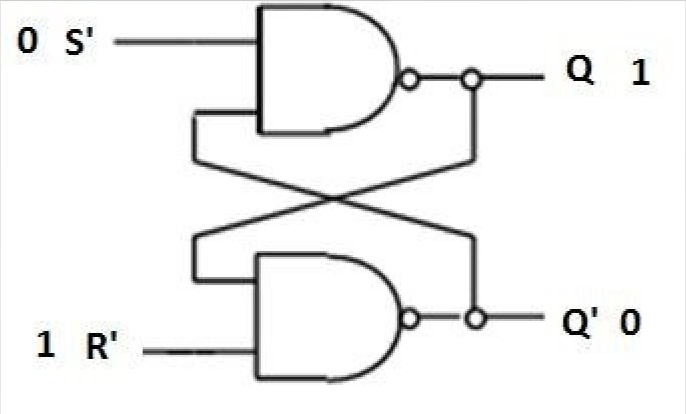
Κατασκευή S΄R΄ latch (NAND)(2)



S’	R’	Q	Q’	
0	0			
0	1	1	0	**Set**
1	0			
1	1	1	0	**Hold**

X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

Κατασκευή S΄R΄ latch (NAND)(3)



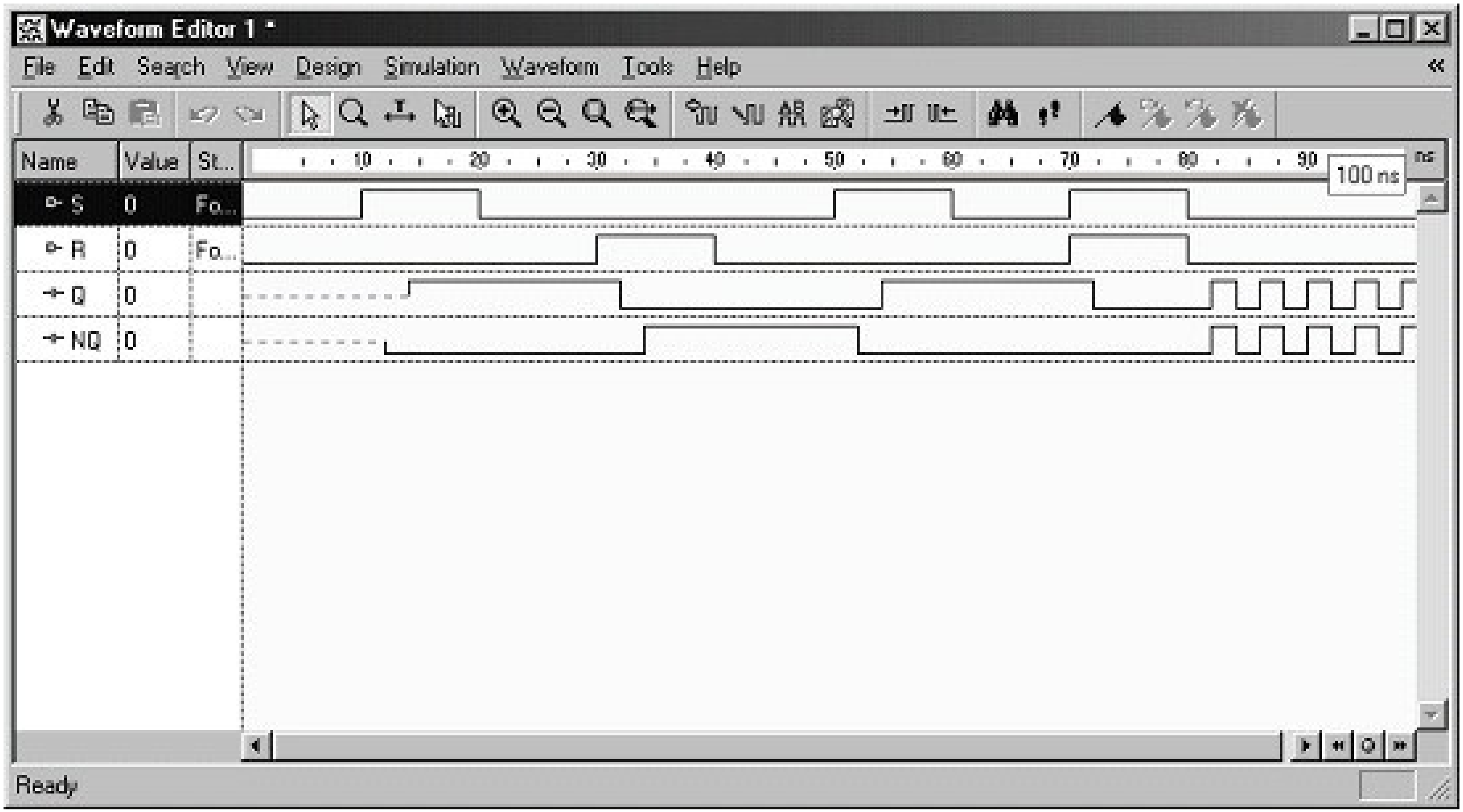
S’	R’	Q	Q’	
0	0			
0	1	1	0	**Set**
1	0	0	1	**Reset**
1	1	1	0	**Hold**
		0	1	**Hold**

X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

Τυπική λειτουργία S΄R΄

- Η λειτουργία του S'R' είναι αντίστροφη του SR.
- Σε κανονική λειτουργία διατηρούμε τις εισόδους S,R στο 1 για να διατηρηθεί η κατάσταση.
- Αν θέσουμε S = 0,R = 1 τότε έχουμε 'θέση'.
- Αν θέσουμε S = 1,R = 0 τότε έχουμε 'επαναφορά'.
- Απροσδιόριστη κατάσταση αν S = R = 0.

Προσομοίωση SR latch

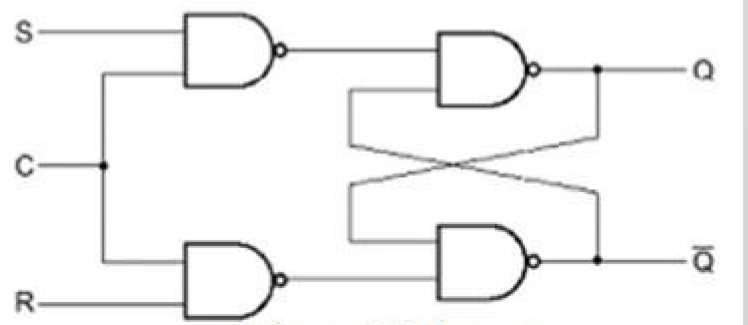


Βελτίωση λειτουργίας SR

- Η λειτουργία του SR μπορεί να βελτιωθεί αν προσθέσουμε μια επιπλέον είσοδο ελέγχου.
- Καθορίζει ΠΟΤΕ μπορεί να αλλάξει η κατάσταση του μανδαλωτή.
- Η **είσοδος επίτρεψης C** λειτουργεί ως σήμα επίτρεψης.

SR latch με σήμα ελέγχου (1)

- Το latch είναι ευαίσθητο σε αλλαγές στις εισόδους ΜΟΝΟ όταν το C = 1.
- Σημαντικό στοιχείο, χρησιμοποιείται για σχεδιασμό άλλων latches και flip-flops.



(α) Λογικό διάγραμμα

SR latch με σήμα ελέγχου (2)

C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	Q = 0; Reset state
1	1	0	Q = 1; Set state
1	1	1	Undefined X

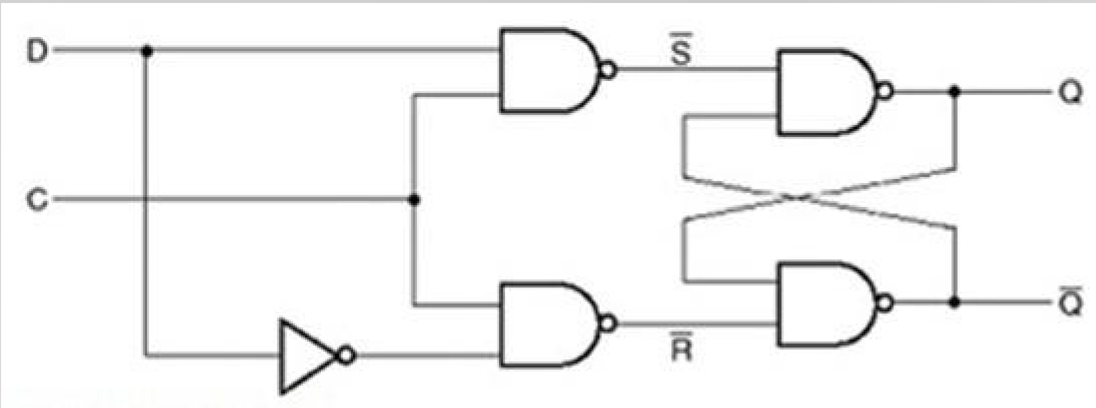
Το SR latch είναι δυσχρηστο

- Αν C = S = R = 1 τότε προκύπτει η μοναδική απροσδιόριστη κατάσταση.
- Δε μπορούμε να προβλέψουμε με σιγουριά την επόμενη κατάσταση.
- Αυτή η αόριστη κατάσταση κάνει το κύκλωμα δύσχρηστο.
- Μπορεί να εξαλειφθεί αν εξασφαλίσουμε ότι οι είσοδοι SR δεν παίρνουν πότε την τιμή 1 → D-latch.

Μανδαλωτής τύπου D

•Ένας τρόπος αποφυγής των ανεπιθύμητων ακαθόριστων καταστάσεων στο RS flip-flop, είναι η εξασφάλιση ότι οι είσοδοι S και R δεν θα πάρουν ποτέ την τιμή 1 ταυτόχρονα.

- Αυτό επιτυγχάνεται με ένα SR-latch όπου
  S = D και R = D′ → D-latch:



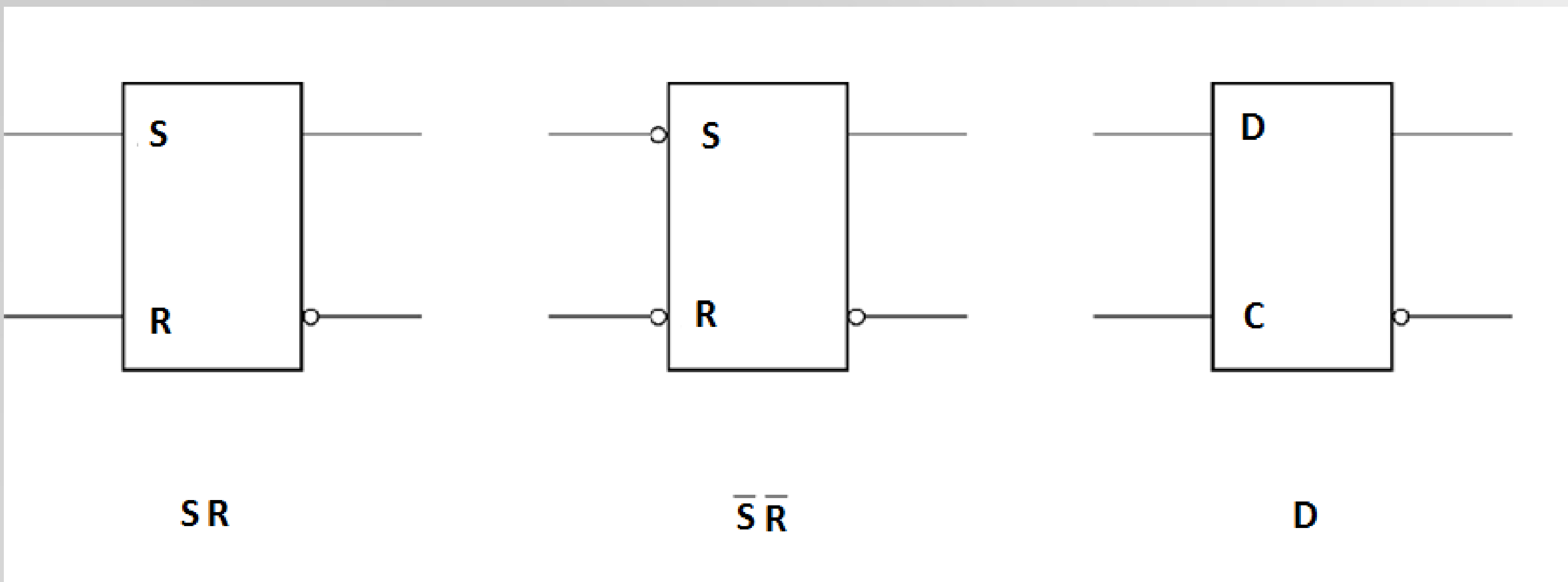
C	D	Next state of Q
0	X	No change
1	0	Q=0; Reset state
1	1	Q=1;Set state

• Δυο είσοδοι μόνο.

Τυπική λειτουργία D latch

- Ονομάζεται **D** από τη λέξη **DATA** αφού μπορεί να συγκρατήσει δεδομένα.
- Ονομάζεται **διαφανής μανδαλωτής** αφού όταν το C = 1 τότε σχηματίζεται μια λογική σύνδεση της εισόδου D με την έξοδο.
- Όταν το C = 0 τότε η δυαδική πληροφορία που ήταν στην είσοδο D διατηρείται στην έξοδο Q.

Σχηματικά Σύμβολα μανδαλωτών



Flip Flop

- Τα Latches είναι “διαυγή” (transparent) δηλ. οποιαδήποτε αλλαγή στην κατάσταση του latch είναι αντιληπτή και στις εξόδους (αν υπάρχει σήμα ελέγχου C, αυτό ισχύει κατά τη διάρκεια που C = 1).
- Αυτό προκαλεί προβλήματα συγχρονισμού, αφού η κατάσταση ενός latch μπορεί να αλλάξει πολλαπλές φορές όταν C = 1!
- Λύση:

  Χρησιμοποιούμε latches για την δημιουργία των flip-flops που μπορούν να ανταποκριθούν (update) ΜΟΝΟ σε ΣΥΓΚΕΚΡΙΜΕΝΕΣ χρονικές στιγμές (όχι ανά πάσα στιγμή ή κατά τη διάρκεια ενός διαστήματος).

Πυροδότηση (triggering)

- Ο μηχανισμός που επιτρέπει σε ένα στοιχείο μνήμης (latch ή FF) να αλλάξει κατάσταση.

- Τρόποι πυροδότησης:
    - Ασύγχρονα, δηλ. εντελώς διαυγή (πχ. SR-latch).
    - Πυροδότηση-επιπέδου (level trigger, C = 1) (πχ. SR-latch ή D-latch με σήμα έλεγχου C).
    - Master-Slave (πχ. SR-FF, D-FF).
    - Πυροδότηση-ακμής: θετική ή αρνητική ακμή του C (rising trigger, C = ↑ ή C = ↓) (πχ. SR-FF, D-FF).

Εναλακτικες Υλοποιήσεις FF

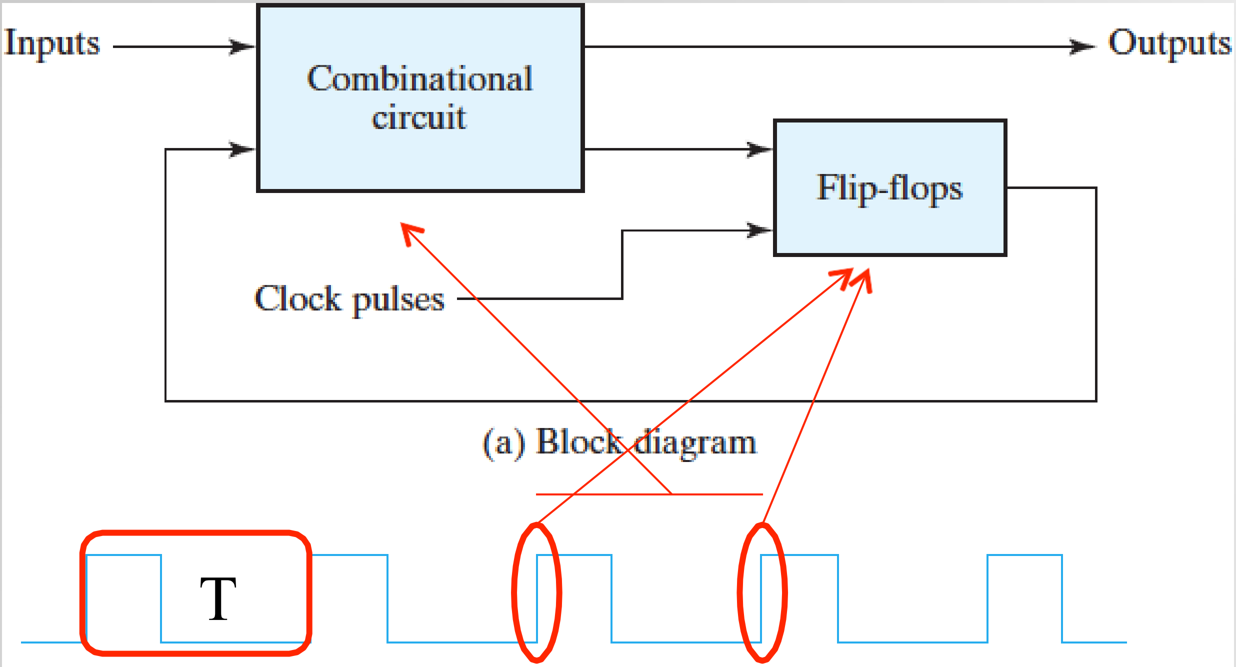
- Τύποι FF:
  - SR
  - D
  - JK

- Τρόποι ενεργοποίησης (triggering):
  - **Master Slave**: χρησιμοποιεί πυροδότηση-επιπέδου αλλά με 2 latches, έτσι ώστε η κατάσταση του FF αλλάζει μόνο μια φορά σε μια περίοδο ρολογιού.
  - **Ενεργοποίηση-ακμής**: θετική ή αρνητική ακμή του C (rising or falling edge trigger, C = ↑ ή C = ↓).

Ρολόι και Συγχρονισμός

F = 1/T

T = Περίοδος Ρολογιού



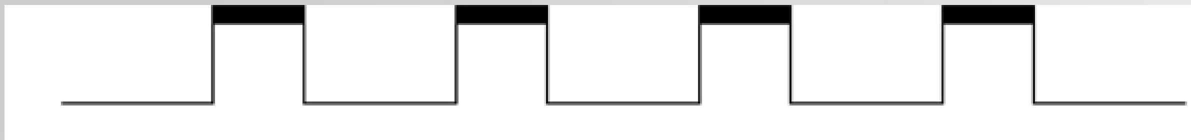
(a) Block diagram

(b) Timing diagram of clock pulses

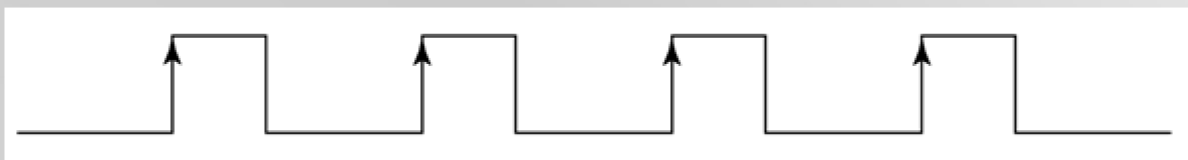
Η αποθήκευση γίνεται σε συγκεκριμένες διακριτές χρονικές στιγμές:

θετικές ή **αρνητικές** ακμές

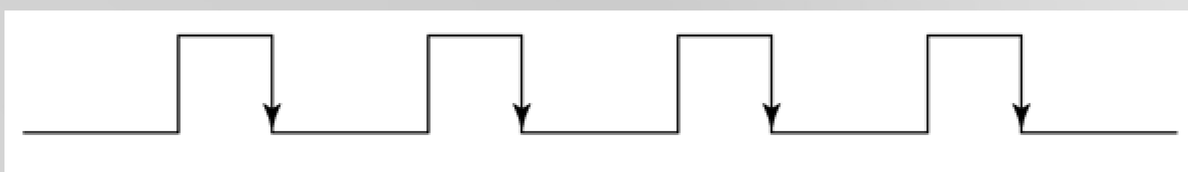
Απόκριση μανδαλωτών και flip-flop σε σήμα ρολογιού



(a) Response to positive level (ανταπόκριση στο θετικό επίπεδο)



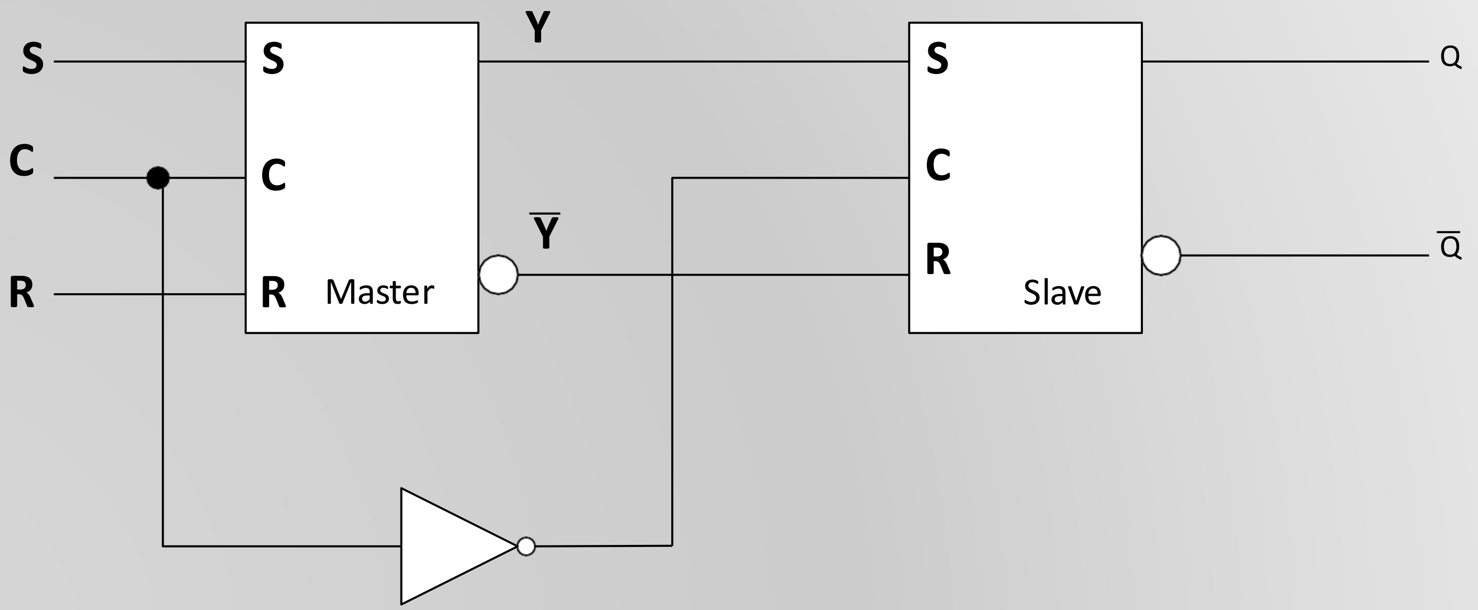
(b) Positive-edge response (ανταπόκριση στις θετικές ακμές)



(c)Negative-edge response (ανταπόκριση στις αρνητικές ακμές)

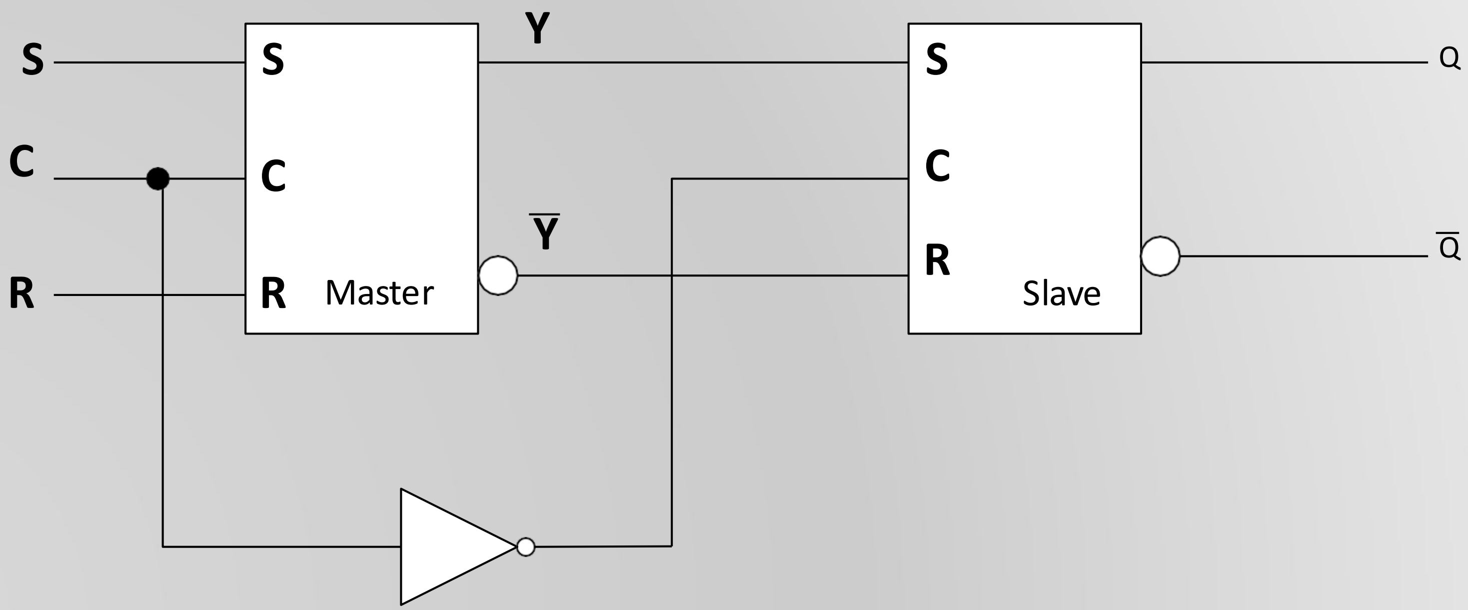
Master Slave SR (1)

- Χρησιμοποιεί πυροδότηση-επιπέδου.
- Κατάσταση Q = Y, όταν C = 0. Επίσης το Y δεν μπορεί να αλλάξει τιμή όταν C = 0.



Master Slave SR (2)

- Όταν C = 1, ο master ενεργοποιείται και φυλάει νέα δεδομένα, και ο slave αποθηκεύει παλιά δεδομένα.
- Όταν C = 0, η κατάσταση του master αποθηκεύεται στον slave (Q = Y), ενώ ο master δεν είναι ευαίσθητος σε νέα δεδομένα.



Master Slave SR (3)

S	R	C	Q	Q′	
0	0	1	Q_0	$Q_{0'}$	Store
0	1	1	0	1	Reset
1	0	1	1	0	Set
1	1	1	1	1	Disallowed
X	X	0	Q_0	$Q_{0'}$	Store

2 προβλήματα

1. Η αλλαγή στις εξόδους του FF έχει καθυστέρηση κατά ½ περίοδο του ρολογιού → το κύκλωμα γίνεται πιο αργό.

2. S και / ή R μπορούν να αλλάξουν πολλαπλές φορές όταν C = 1.

   Q = 1, S = 0 → 1 → 0 και R = 0

   Master latch = 1(set)

   Slave = 1(set),όταν C = 0

   Q = 1, S = 0 → 1 → 0 και R = 0 → 1 → 0

   Master latch = 1(set) και μετά = 0 (reset)

   Slave = 0 (reset), όταν C = 0

   Γνωστό ως = «I's catching»

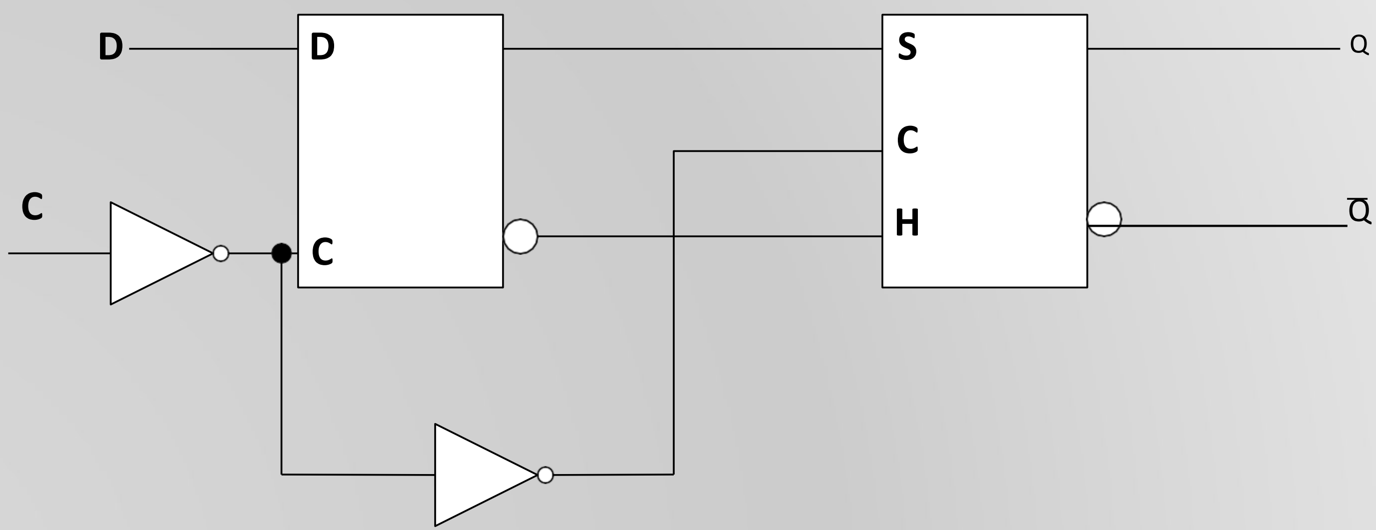
Λύση: πυροδότηση ακμής

- Ένα ακμοπυροτούμενο FF, αγνοεί τις αλλαγές κατά τη διάρκεια ενός παλμού.
- Πυροδοτείται μόνο όταν υπάρχει μετάβαση της τιμής του ρολογιού (clock transition. ↑ / ↓).

- Υλοποίηση ακμοπυροτούμενων FF:
  - Άμεσα, σε επίπεδο ηλεκτρονικού κυκλώματος.
  - Με master-slave D-FF.

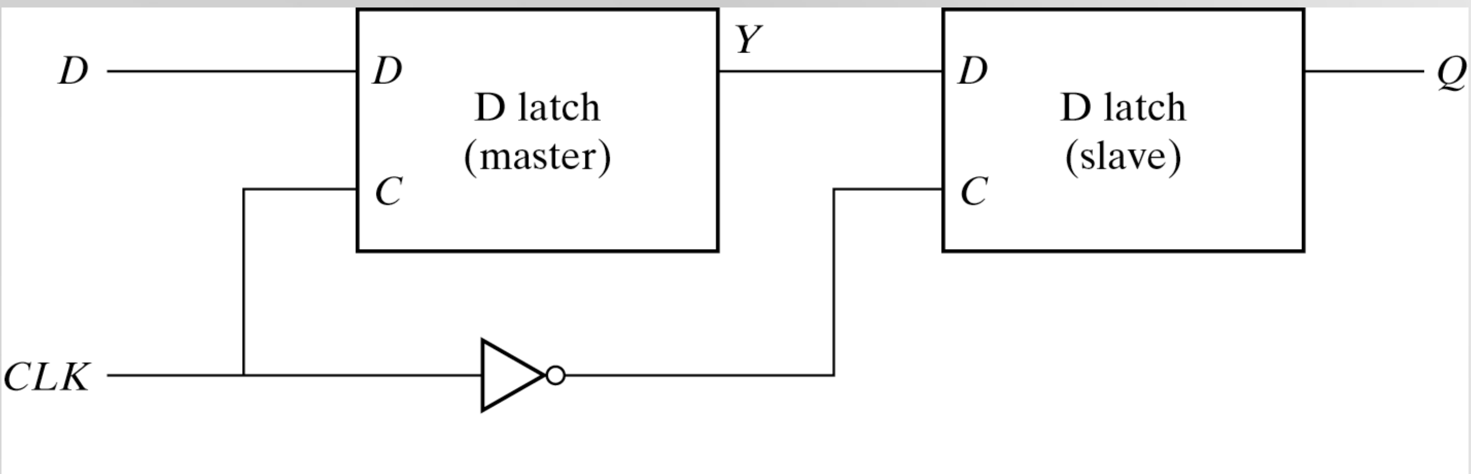
Ακμοπυροδοτούμενα FF

- Συνδέουμε ένα D-latch με πυροδότηση-επιπέδου (master) με ένα SR-latch με πυροδότηση-επιπέδου (slave) και συμπληρωματικά ρολόγια.



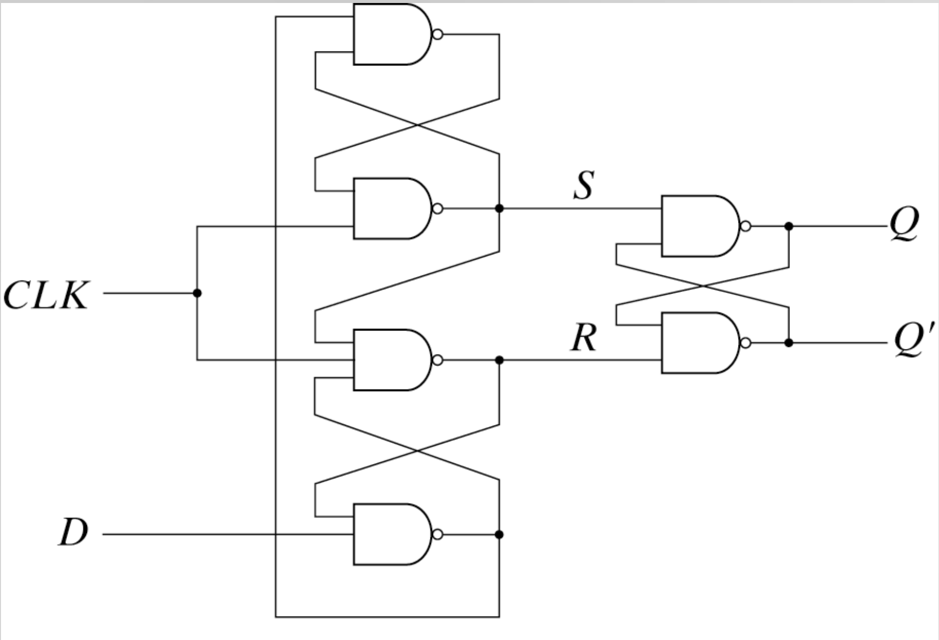
D flip-flop αφέντη-σκλάβου

- Η έξοδος του flip-flop μπορεί να αλλάξει μόνο κατά τη μετάβαση του ρολογιού από το 1 στο 0, δηλαδή μόνο κατά την αρνητική ακμή του ρολογιού.
- Αν τοποθετηθεί ένας επιπλέον αντιστροφέας τότε αλλάζει η έξοδος κατά την θετική ακμή του ρολογιού.



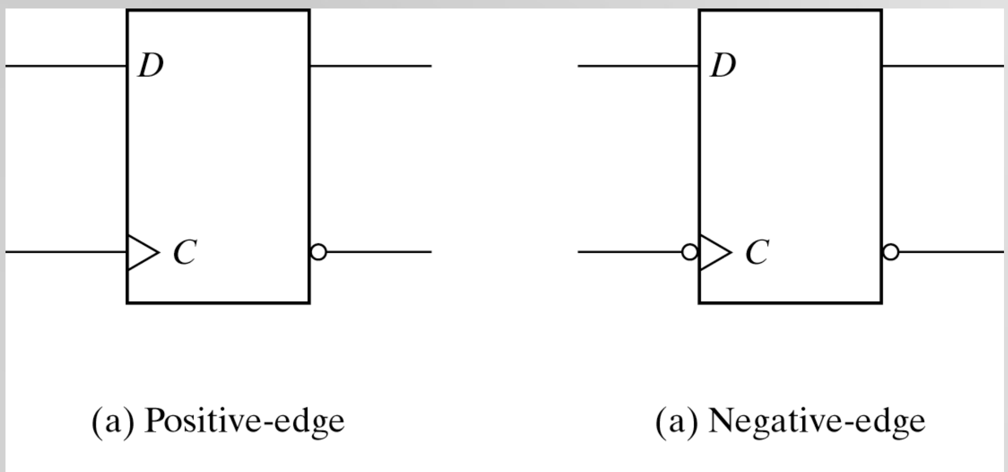
Ακμοπυρόδοτο D flip-flop

- Αν CLK = 0 τότε τα S, R διατηρούνται στο 1.
- Αν D = 0, CLK = 1 τότε Q = 0.
- Ακόμη και αν αλλάξει D τότε το R παραμένει στο 0.



Σχηματικό σύμβολο για το ακμοδοπυροδότητο flip-flop

- Το τριγωνικό σύμβολο μπροστά από το C υποδηλώνει μια δυναμική είσοδο, δηλαδή ότι αντιδρά σε μια ακμή του ρολογιού.



Positive-edge: Θετική ακμή

Negative-edge: Αρνητική ακμή

Άλλα flip-flop

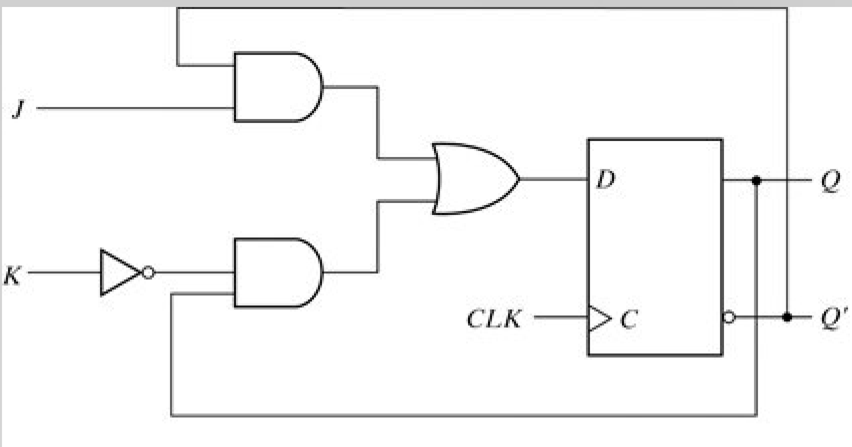
- Το πιο οικονομικό και αποτελεσματικό flip-flop είναι το D, επειδή απαιτεί το μικρότερο αριθμό πυλών.
- Άλλοι τύποι μπορούν να κατασκευαστούν με το D f-f και εξωτερικής λογικής.
- Οι πιο γνωστοί τύποι είναι τα JK και τα T.

JK flip-flop (1)

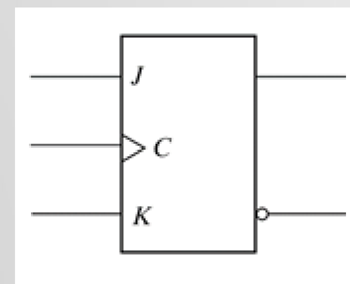
- 3 Λειτουργίες εκτελούνται σε ένα flip-flop.
  - Να το θέσουμε στο 1.
  - Να το επαναφέρουμε στο 0.
  - Να συμπληρώσουμε την έξοδο.
- Το JK εκτελεί και τις 3 λειτουργίες.

JK flip-flop (2)

- Η είσοδος J φέρνει το flip-flop σε κατάσταση θέσης.
- Η είσοδος K φέρνει το flip-flop σε κατάσταση επαναφοράς.
- Ταυτόχρονα J = K = 1 τότε η έξοδος συμπληρώνεται.

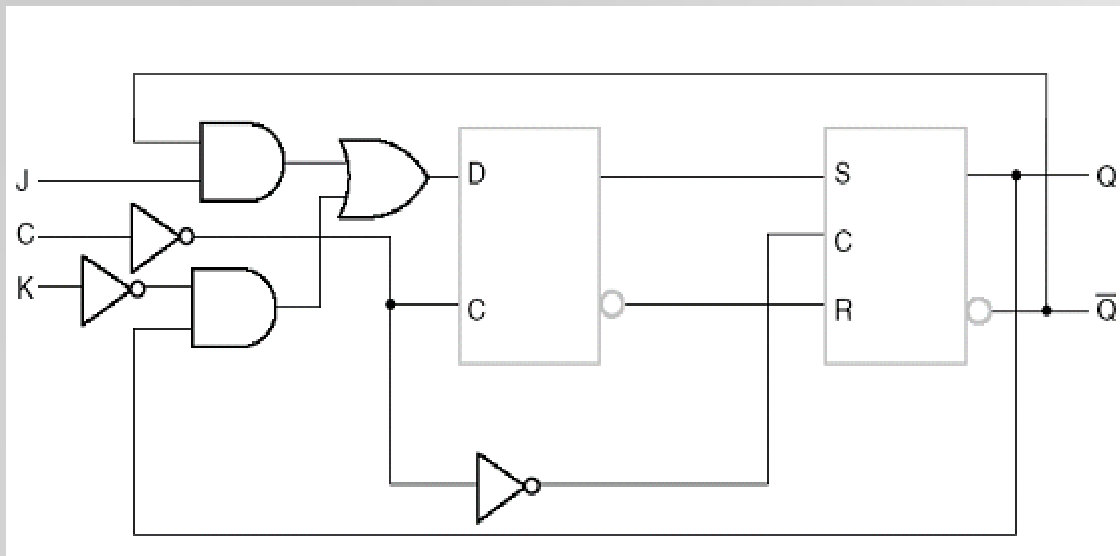


(α) Διάγραμμα κυκλώματος



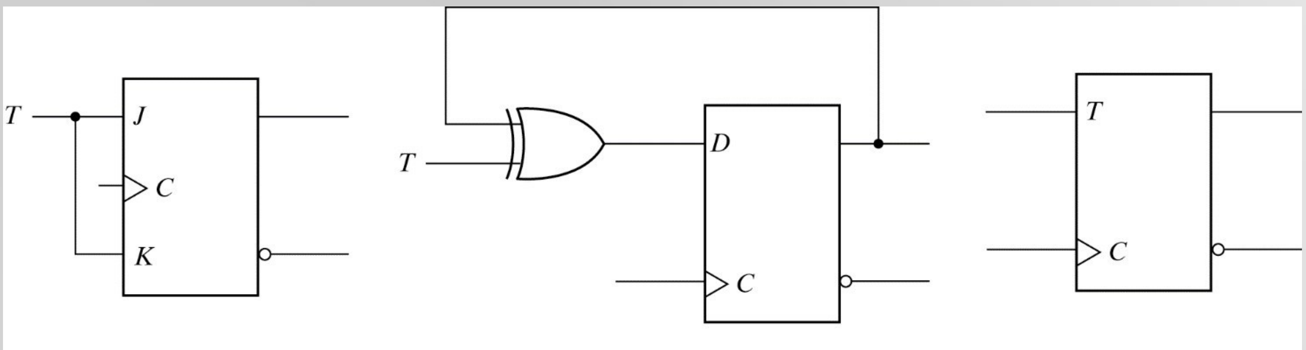
(β) Γραφικό σύμβολο

JK flip-flop (3)



Flip-flop τύπου T

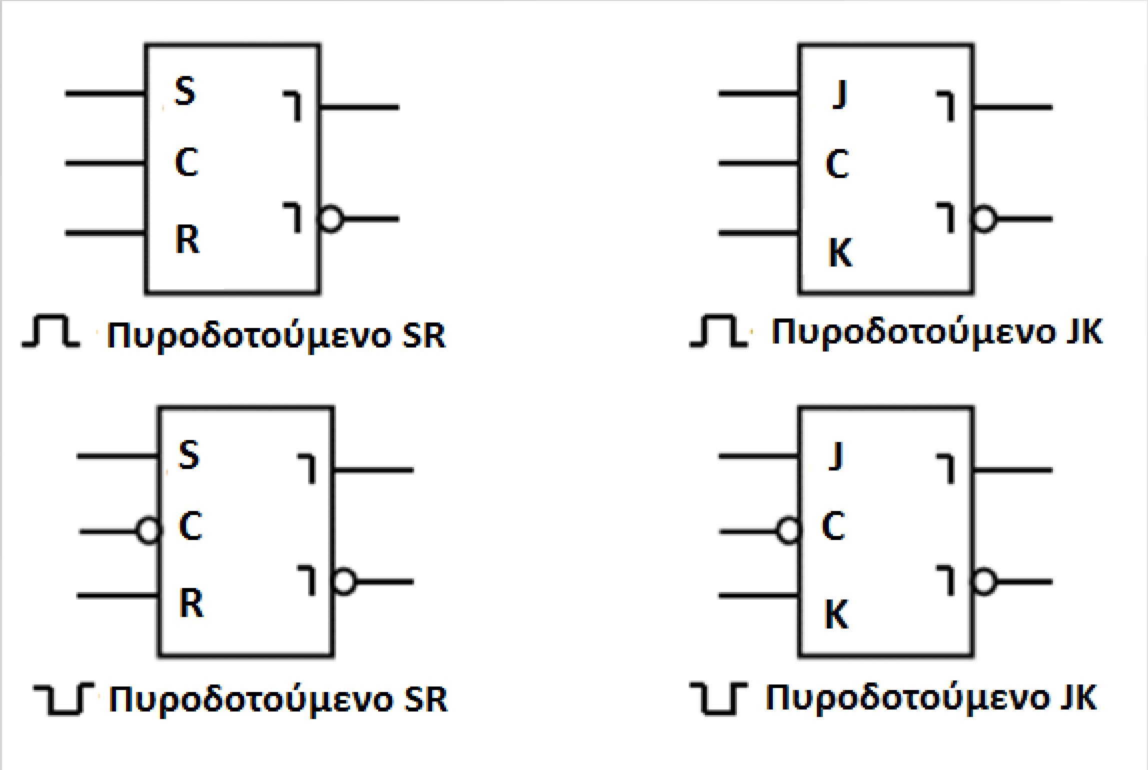
- Αν T = 0 δεν υπάρχει αλλαγή.
- Αν T = 1 η έξοδος συμπληρώνεται.



(a) From *JK* flip-flop (b) From *D* flip-flop (c) Graphic symbol

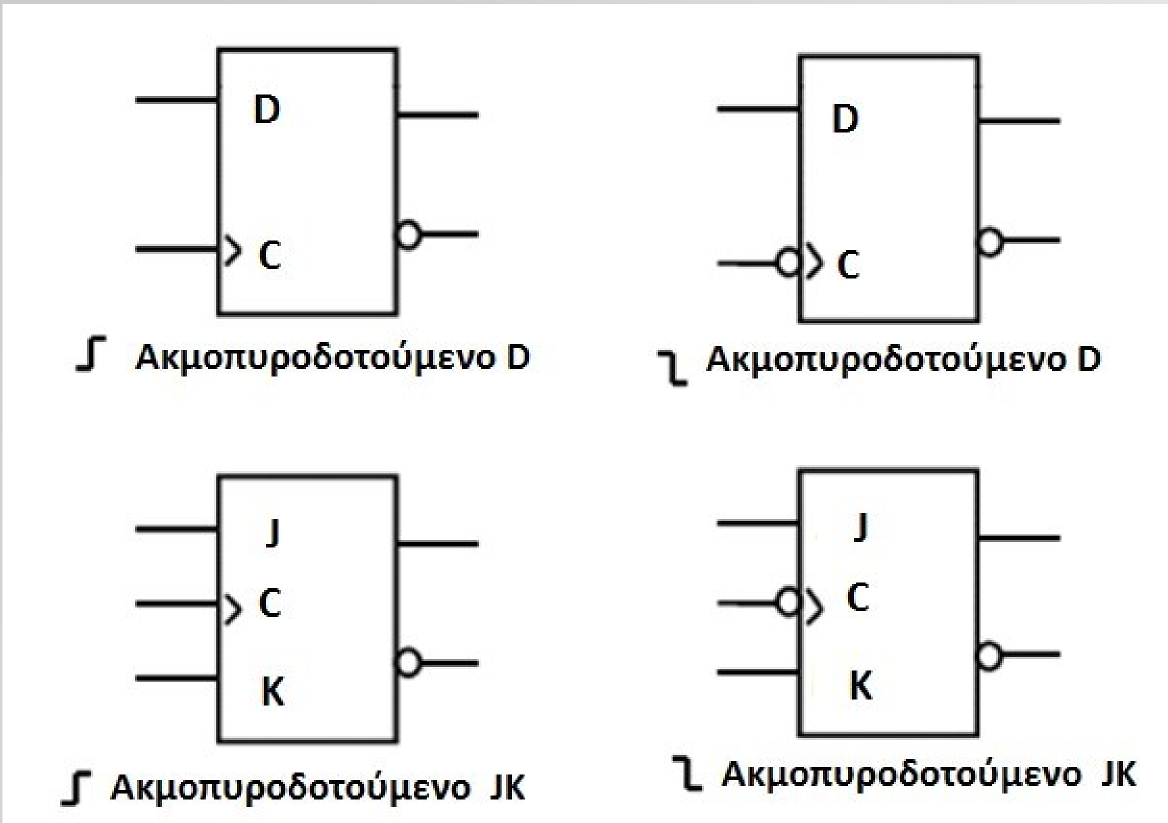
Καθιερωμένα γραφικά σύμβολα (1)

- Master-Slave Flip Flops – Πυροδότηση Επιπέδου (level-triggering).



Καθιερωμένα γραφικά σύμβολα (2)

- Ακμοπυροδοτούμενα (Edge-triggered) Flip Flops.



Χαρακτηριστικός Πινακας

- Καθορίζει τις λογικές ιδιότητες / χαρακτηριστικά ενός flip / flop (όπως ένας πίνακας αληθείας για μια λογική πύλη).

- Q(t) - παρούσα κατάσταση στο χρόνο t.
- Q(t + 1) - επόμενη κατάσταση στο χρόνο t + 1.

Χαρακτηριστικός Πινακας JK

J	K	Q(t+1)	Λειτουργία
0	0	Q(t)	Καμία αλλαγή/Hold
0	1	0	Reset
1	0	1	Set
1	1	Q(t)΄	Συμπλήρωμα

Χαρακτηριστικός Πινακας SR

S	R	Q(t + 1)	Λειτουργία
0	0	Q(t)	Καμία αλλαγή/Hold
0	1	0	Reset
1	0	1	Set
1	1	?	Ακαθόριστο/Άκυ ρο

Χαρακτηριστικός Πινακας D

- Χαρακτηριστικη Εξίσωση: $Q(t+1) = D(t)$
  - Εκφράζει την τιμή των εξόδων στο χρόνο t + 1 σε σχέση με την τιμή των εισόδων στο χρόνο t, για ένα flip-flop.

D	Q(t + 1)	Λειτουργία
0	0	Reset
1	1	Set

Χαρακτηριστικός Πινακας T

- Χαρακτηριστική Εξίσωση: $Q(t+1) = T'Q(t) + TQ(t)'$.
  - T Flip-Flop(από JK Flip-Flop με J = K = T).

T	Q(t + 1)	Λετουργία
0	Q(t)	Καμία Αλλαγή/Hold
1	Q(t)΄	Συμπλήρωμα

Ασύγχρονο Set/Reset

•Πολλές φορές είναι επιθυμητό να μπορούμε να θέσουμε την τιμή ενός FF (set ή reset) ανεξάρτητα με το ρολόι.

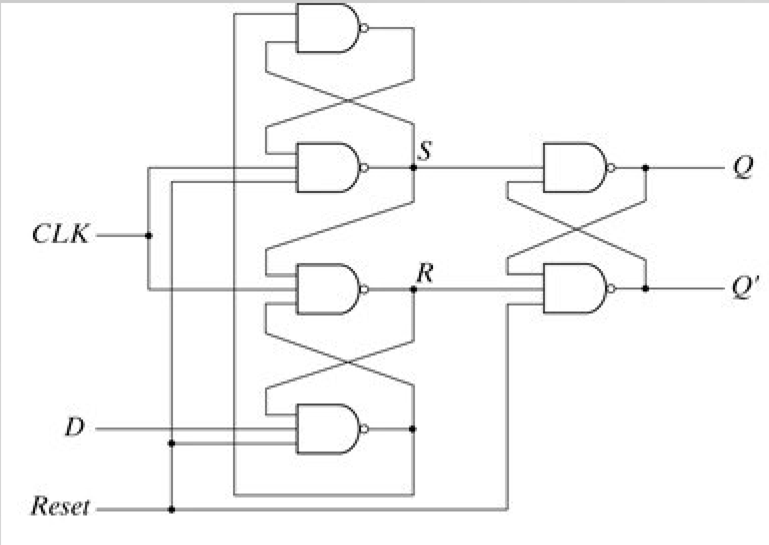
→ ασύγχρονο set / reset

•<u>Παράδειγμα</u>: Στο ξεκίνημα (power-up) χρησιμοποιούμε ασύγχρονο set / reset έτσι ώστε να ξεκινούμε από μια γνωστή κατάσταση (known state).

•Ασύγχρονο set == άμεσο set == <u>Preset</u>

•Ασύγχρονο reset == άμεσο reset == <u>Clear</u>

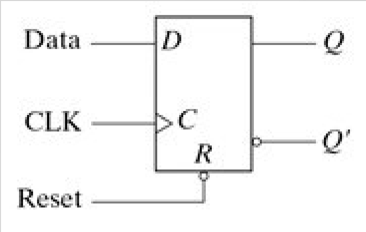
D ff με ασύγχρονη επαναφορά



(α) Διάγραμμα κυκλώματος

R	C	D	Q	Q’
0	X	X	0	1
1	|	0	0	1
1	|	1	1	0

(β) Πίνακας συνάρτησης



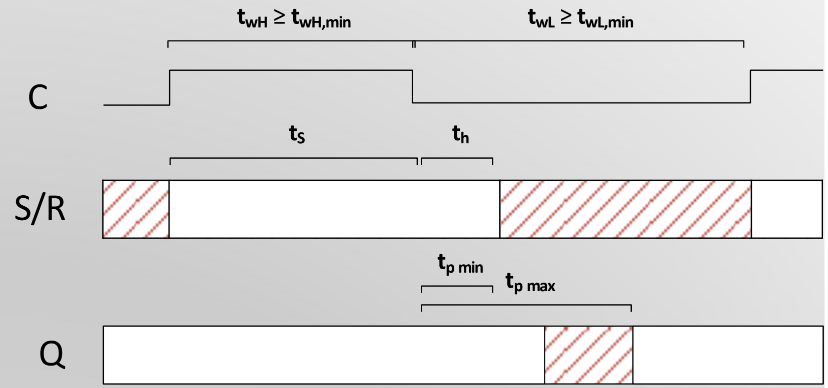
(γ) Γραφικό σύμβολο

Παράμετροι χρονισμού για FF (1)

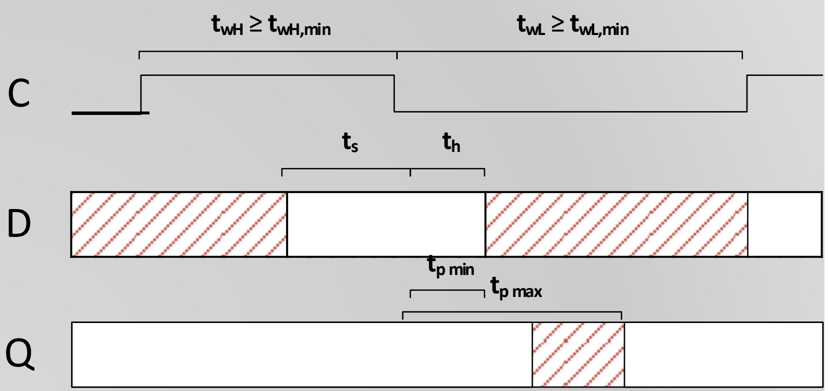
- t_s – setup time : Απαραίτητος χρόνος όπου οι είσοδοι του FF πρέπει να παραμείνουν σε σταθερές τιμές, <u>πριν την πυροδότηση</u>, για να παρατηρηθεί αλλαγή στην έξοδο.
  - Master-slave: ίσο με το πλάτος του παλμού πυροδότησης.
  - Edge-triggered: ίσο με ένα διάστημα, πολύ μικρό από αυτό του πλάτους του παλμού πυροδότησης.
- t_h – hold time: απαραίτητος χρόνος όπου οι είσοδοι του FF πρέπει να κρατήσουν τις τιμές τους, <u>μετά την πυροδότηση.</u>
  - Συχνά μπορεί να αγνοηθεί (κοντά στο 0).
- t_{px} – propagation delay: καθυστέρηση μετάδοσης, δηλ. χρόνος από την πυροδότηση μέχρι την σταθεροποίηση της νέας τιμής στην έξοδο.
  - Μετριέται από την ακμή που πυροδοτεί την αλλαγή στην έξοδο μέχρι την εμφάνιση της αλλαγής στην έξοδο.
- Απαραίτητα $t_{px} > t_h$

Παράμετροι χρονισμού για FF (2)

- Pulse-triggered (positive pulse)



- Edge-triggered (negative edge)



- t_S – setup time (χρόνος εγκατάστασης)
- t_h – hold time (χρόνος αναμονής)
- t_w – clock pulse width

(πλάτος παλμού ρολογιού)

•t_{px} – propagation delay (καθυστέρηση διάδοσης)

- t_{PHL} – High to low (χρόνος από το υψηλό στο χαμηλό)
- t_{PLH} – Low to high (χρόνος από το χαμηλό στο υψηλό)
- t_{PHL} – max (t_{PHL} , t_{PLH}) (μέγιστο ανάμεσα του t_{PHL} και του t_{PLH})

Ανάλυση ακολουθιακών κυκλωμάτων

- Ανάλυση: Ο καθορισμός μιας <u>κατάλληλης</u> περιγραφής, η οποία επιδεικνύει τη <u>χρονική ακολουθία</u> εισόδων, εξόδων και καταστάσεων (states).

- Λογικό Διάγραμμα: Λογικές πύλες, flip-flops και κατάλληλες διασυνδέσεις.
- Το λογικό διάγραμμα μπορεί να καθοριστεί από ένα από τα ακόλουθα:
  - Εξισώσεις (FF-Εισόδων, Εξόδων).
  - Πίνακα Καταστάσεων (State Table ή Transition Table).
  - Διάγραμμα καταστάσεων (State Diagram ή Transition Diagram ή Finite State Machine – FSM).

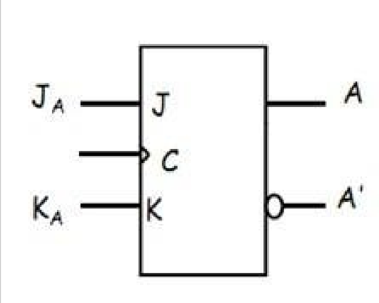
Εξισώσεις εισόδων FF (1)

- Αλγεβρικές αναπαραστάσεις που χρησιμοποιούνται για την περιγραφή της λογικής που οδηγεί τις εισόδους των FFs.

- Υπονοούν τον τύπο των FFs που θα χρησιμοποιηθούν και καθορίζουν πλήρως την συνδυαστική λογική που οδηγεί τις εισόδους των FFs.

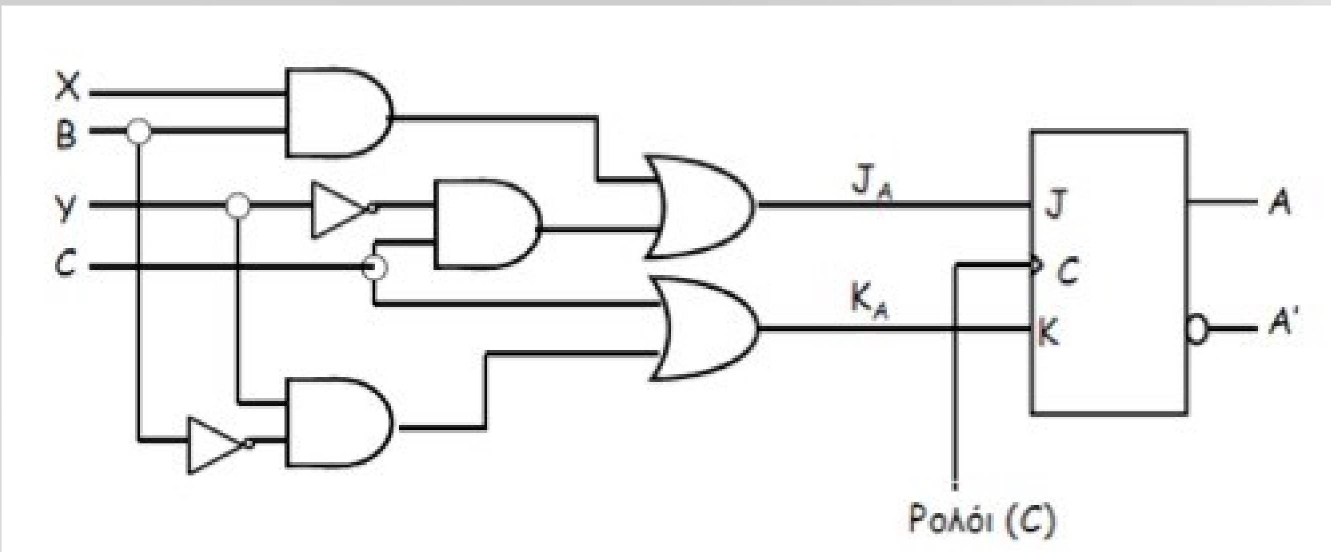
Εξισώσεις εισόδων FF (2)

- Θεωρήστε: J_A = XB + Y'C και K_A= YB' + C.
- Τα J,K υπονοούν τον τύπο του FF (σε αυτή την περίπτωση είναι JK-FF).
- Ο δείκτης ($_A$) ορίζει την έξοδο του FF.
- Παρατηρήστε ότι ο τύπος πυροδότησης δεν καθορίζεται από τις εξισώσεις εισόδων FF. Αυτός είτε δίνεται ή καθορίζεται από τον αναλυτή. Γι αυτό το παράδειγμα, θεωρούμε ότι η πυροδότηση γίνεται στη θετική ακμή.



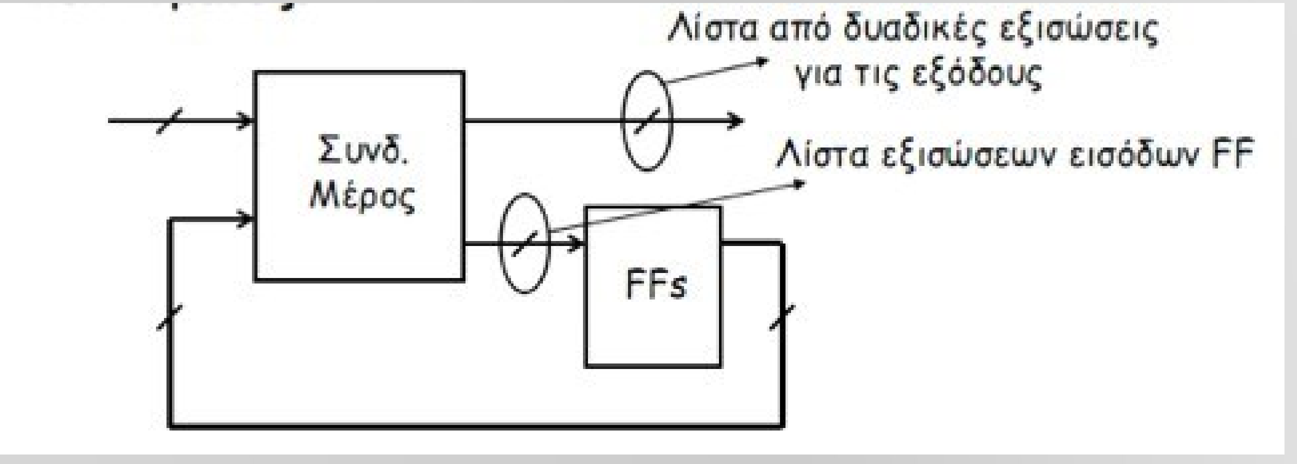
Υλοποίηση Λογικού διαγράμματος

- $J_A = XB + Y'C$
- $K_A = YB' + C$



Πλήρως καθορισμένα λογικά διαγράμματα

- Μπορούν οι εξισώσεις εισόδων FF να καθορίσουν πλήρως το λογικό διάγραμμα ενός ακολουθιακού κυκλώματος;
- Χρειαζόμαστε και τις εξισώσεις για τις εξόδους του κυκλώματος.



Πίνακας Καταστάσεων

- Απαριθμεί τις σχέσεις μεταξύ εισόδων, εξόδων και καταστάσεων (states = τιμές στα FF) ενός ακολουθιακού κυκλώματος.
- Αποτελείται από 4 μέρη:
    - Παρούσα κατάσταση: τις τιμές των FFs για κάθε επιτρεπτή κατάσταση, σε χρόνο t.
    - Είσοδοι: οι επιτρεπτοί συνδυασμοί εισόδων.
    - Επόμενη Κατάσταση: τις τιμές των FFs για κάθε επιτρεπτή κατάσταση σε χρόνο t + 1, βάσει των τιμών στις εισόδους και της παρούσας κατάστασης.
    - Έξοδοι: οι τιμές των εξόδων σε σχέση με την παρούσα κατάσταση και πιθανόν, τις τιμές των εισόδων.
- Δεδομένου ενός κυκλώματος με n εισόδους και m flip-flops, ο αντίστοιχος πίνακας καταστάσεων αποτελείται από 2^{n+m} γραμμές.

Πίνακας Καταστάσεων για JK FF

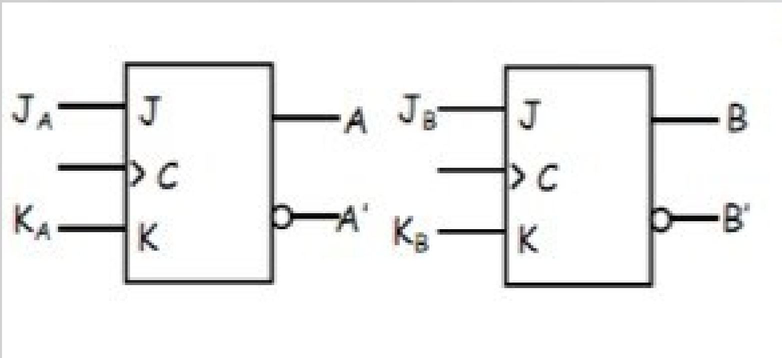
- Διαδικασία σε 2 φάσεις:
  1. Καθορισμός δυαδικών τιμών για κάθε είσοδο FF βάση των εξισώσεων εισόδων FF, σε σχέση με την παρούσα κατάσταση και τις μεταβλητές εισόδου.

  2. Χρήση αντίστοιχων <u>χαρακτηριστικών πινάκων FF</u> για καθορισμό της επόμενης κατάστασης.

Παράδειγμα (1)

- $J_A = B$, $K_A = BX'$
- $J_B = X'$, $K_B = AX' + A'X$

→ χρειαζόμαστε 2 JK-FFs:



Παράδειγμα (2)

J	K	Q(t + 1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Q(t)΄

Παράδειγμα (3)

Παρούσα Κατάσταση	Παρούσα Κατάσταση	Είσοδος	Επόμενη Κατάσταση	Επόμενη Κατάσταση	Είσοδοι FF	Είσοδοι FF	Είσοδοι FF	Είσοδοι FF
A(t)	B(t)	X	A(t+1)	B(t+1)	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

Μηχανές Mealy & Moore

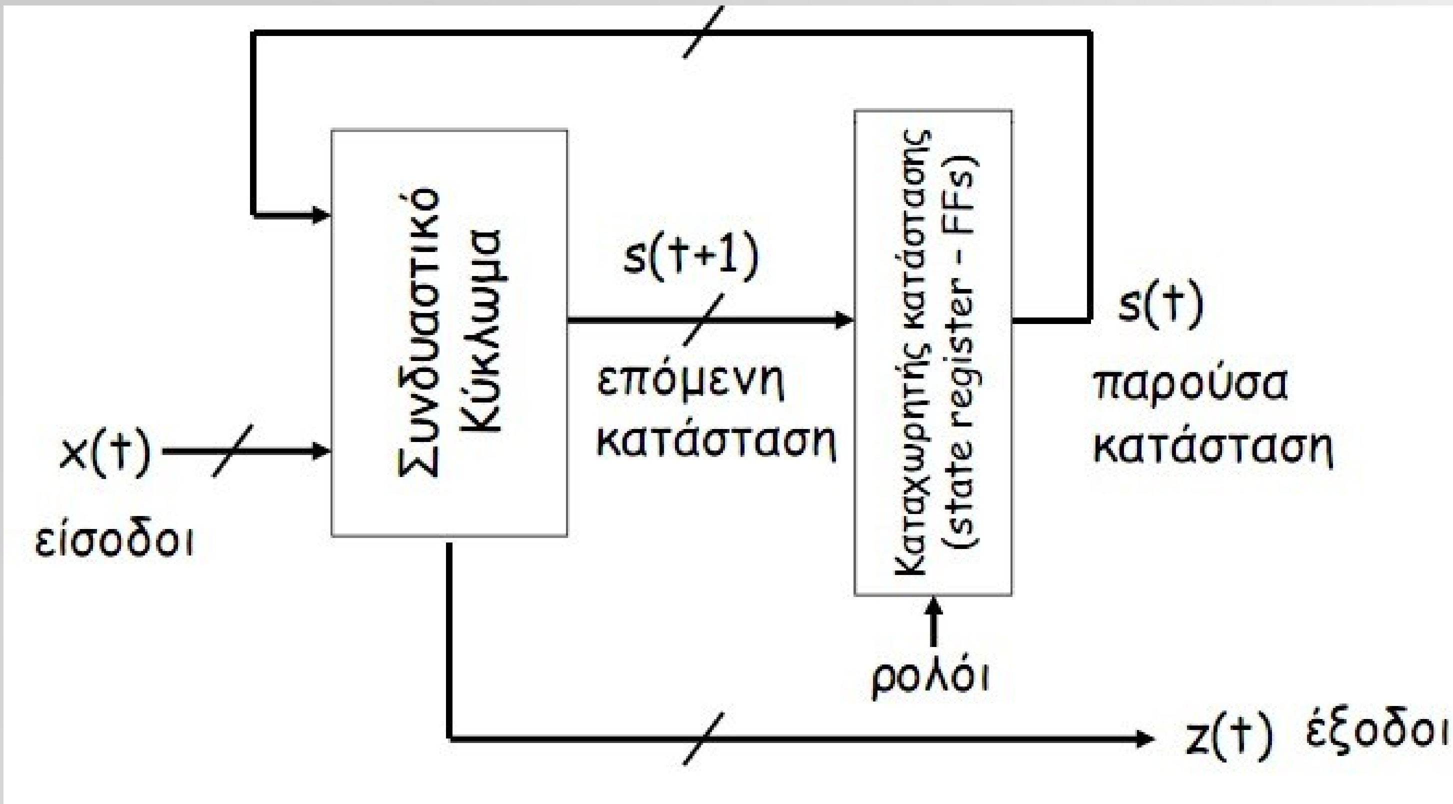
- Μοντέλο Mealy:
  - Έξοδοι **ΚΑΙ** επόμενη κατάσταση εξαρτούνται άμεσα από τις τιμές των εισόδων **ΚΑΙ** της παρούσας κατάστασης.

- Μοντέλο Moore:
  - **ΜΟΝΟ** η επόμενη κατάσταση εξαρτάται άμεσα από τις τιμές των εισόδων **ΚΑΙ** της παρούσας κατάστασης. Οι τιμές στις εξόδους εξαρτούνται μόνο από την παρούσα κατάσταση. (δεν εξαρτούνται άμεσα από τις τιμές των εισόδων).

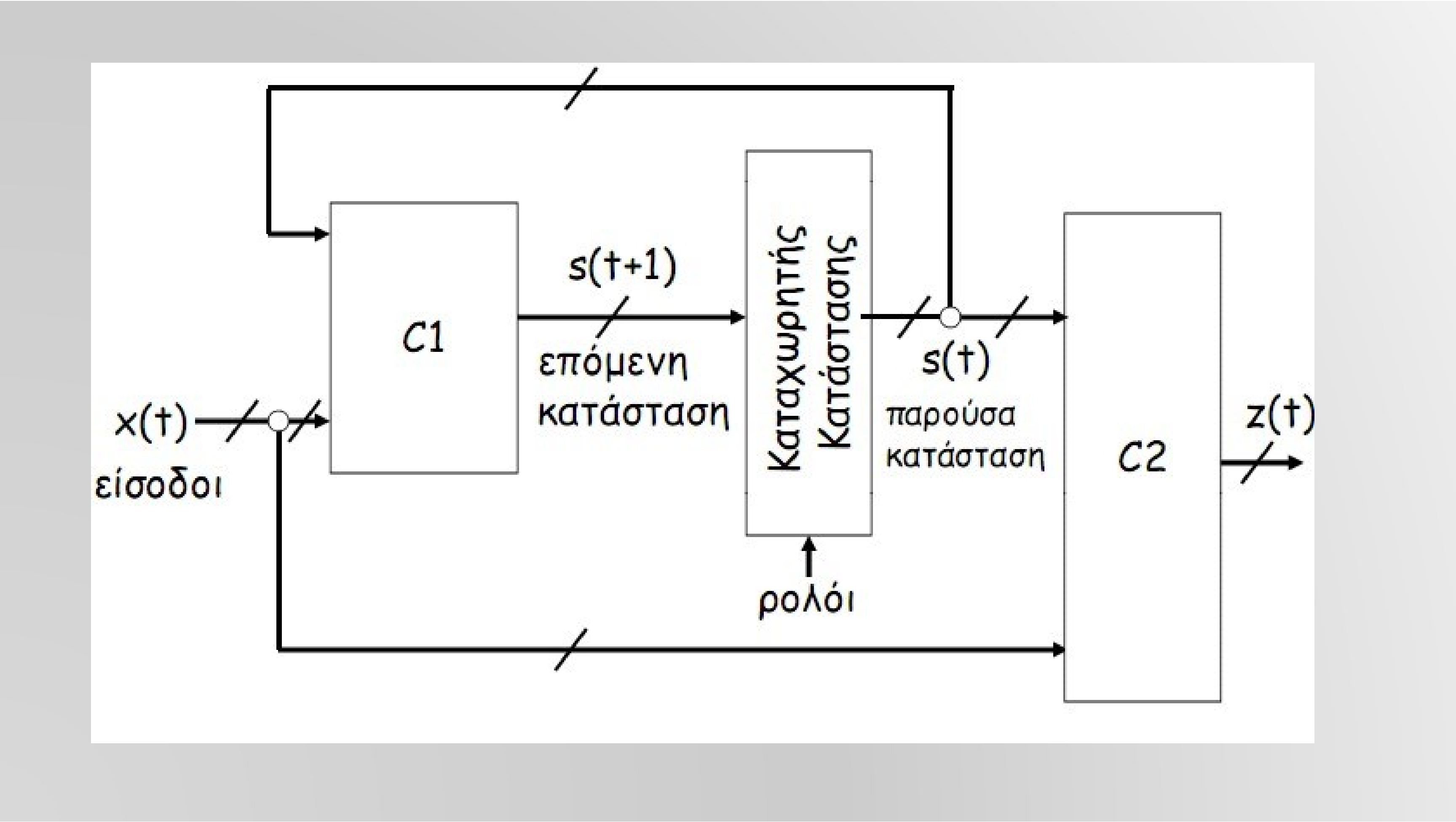
FSM

- Σε οποιαδήποτε κατηγορία ανήκουν τα ακολουθιακά κυκλώματα αναφέρονται ως 'Μηχανές Πεπερασμένων Καταστάσεων' (Finite State Machines – FSM).

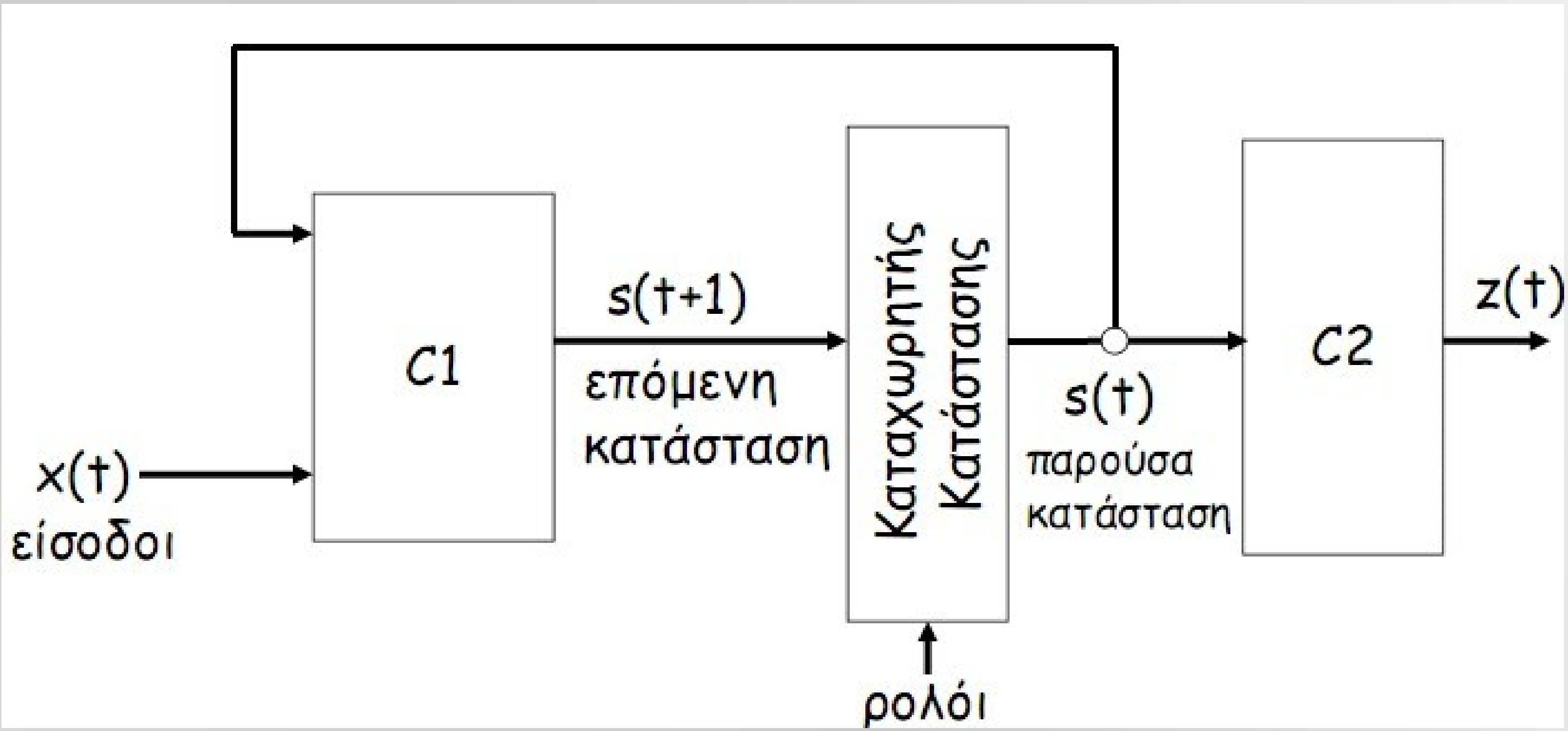
Ένα ακολουθιακό κύκλωμα



Μηχανή Mealy

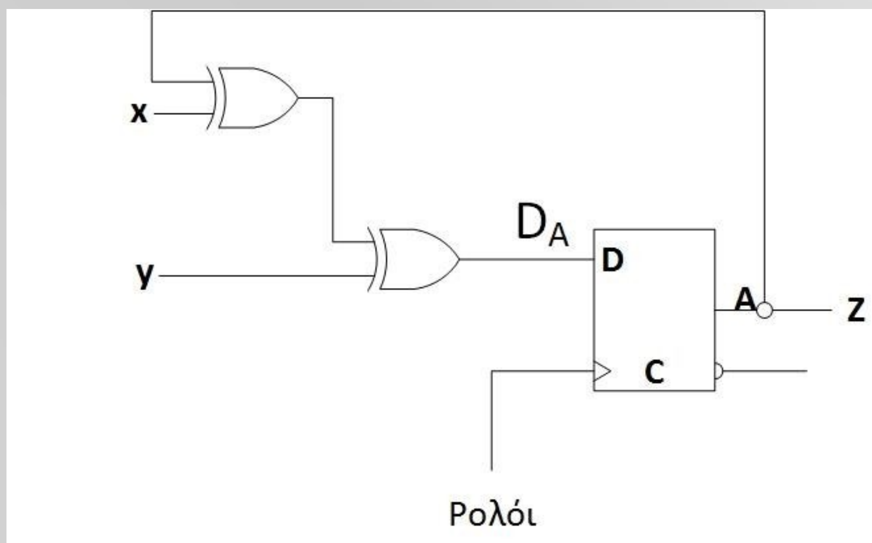


Μηχανή Moore



Παράδειγμα μηχανής Moore

- Βρείτε το λογικό διάγραμμα και τον πίνακα κταστάσεων για:
  - D_A = A © X © Y
  - Z = A



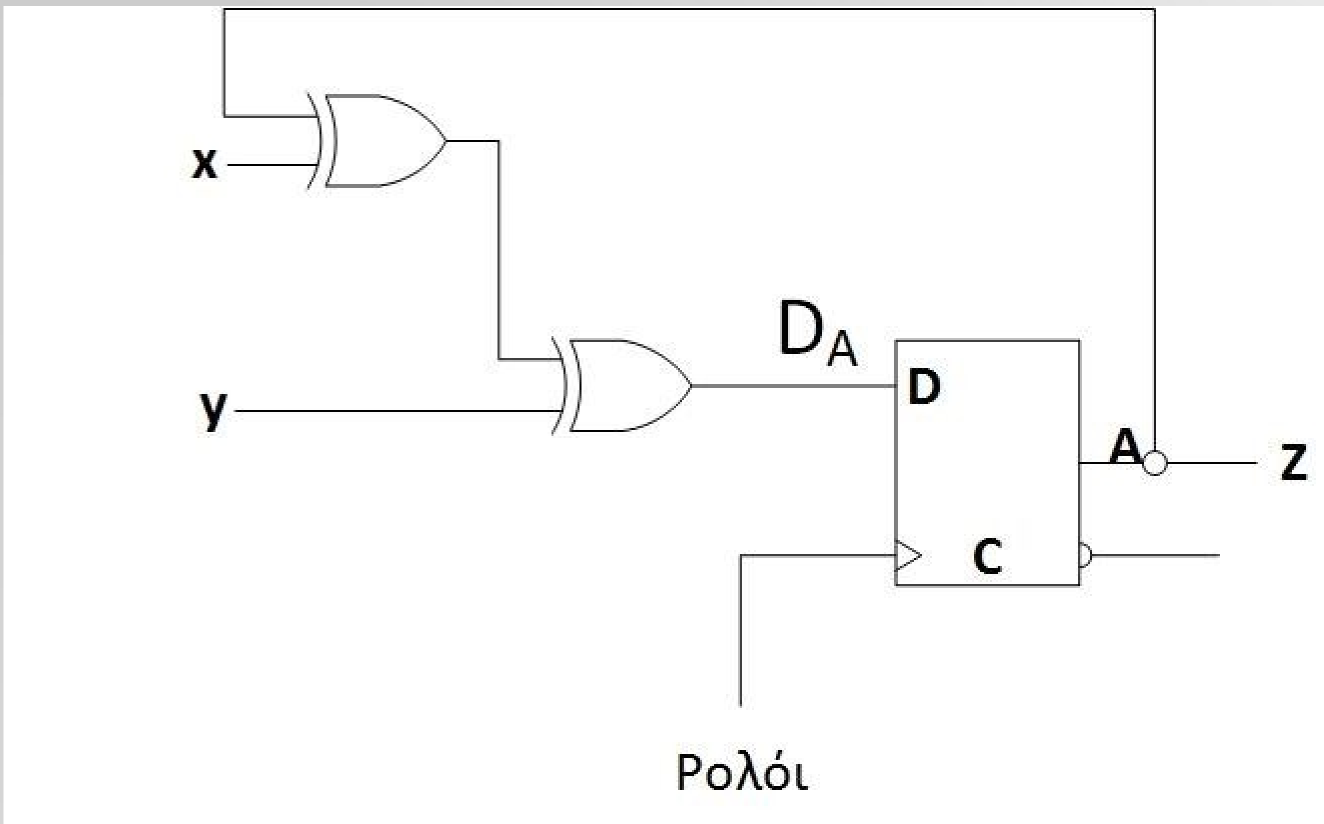
Παράδειγμα μηχανής Moore (1)

Παρούσα Κατάσταση	Είσοδος	Είσοδος	Επόμενη Κατάσταση	Έξοδος
A(t)	X	Y	A(t + 1)	Z
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Παράδειγμα μηχανής Moore (2)

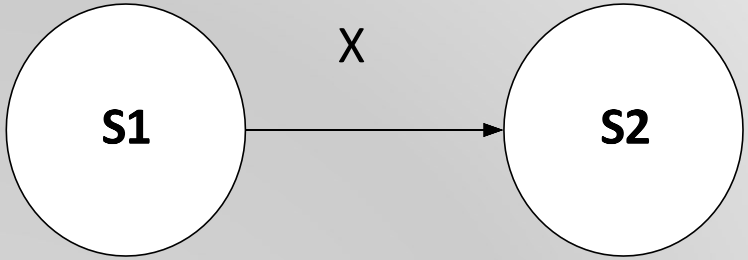
Παρούσα Κατάσταση	Επόμενη Κατάσταση	Επόμενη Κατάσταση	Επόμενη Κατάσταση	Επόμενη Κατάσταση	Έξοδος
A(t)	XY = 00 A(t + 1)	XY = 01 A(t + 1)	XY = 10 A(t + 1)	XY = 11 A(t + 1)	Z
0	0	1	1	0	0
1	1	0	0	1	1

Παράδειγμα μηχανής Moore (3)



Διαγράμματα καταστάσεων

- Γραφική αναπαράσταση του πίνακα καταστάσεων.
- Ένας κόμβος με σήμανση S αντιστοιχεί σε κάθε πιθανή κατάσταση (state) s.
- Μια ακμή με σήμανση X δηλώνει την μετάβαση μεταξύ δύο καταστάσεων (state transition), όταν μια τιμή X εφαρμόζεται στις εισόδους. Δηλ. Αν παρούσα κατάσταση = S1 και input = X, τότε η επόμενη κατάσταση = S2.
- Το διάγραμμα διαφέρει, αναλόγως του τύπου του κυκλώματος (Mealy ή Moore).



Παράδειγμα Mealy (1)

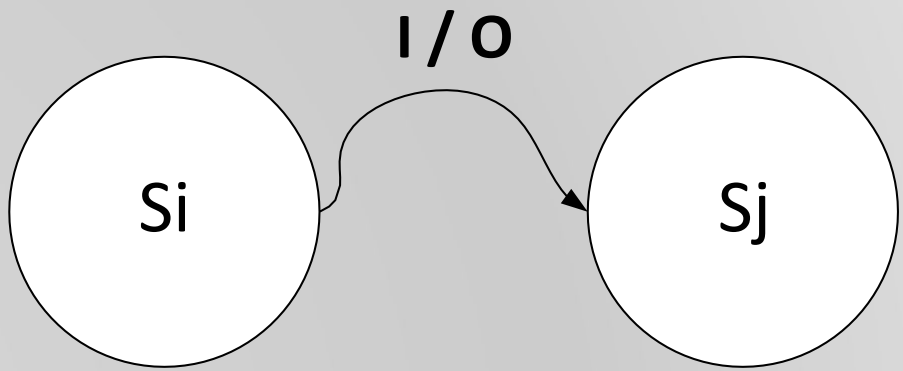
- Πιθανές Καταστάσεις = { 00 , 01 , 10 , 11 } = { s0, s1, s2, s3 } → 4 κόμβοι στο διάγραμμα καταστάσεων.

Παρούσα Κατάσταση	Παρούσα Κατάσταση	Είσοδος	Επόμενη Κατάσταση	Επόμενη Κατάσταση	Έξοδος
A(t)	B(t)	X	A(t + 1)	B(t + 1)	Y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	1
1	1	0	0	0	1
1	1	1	1	0	0

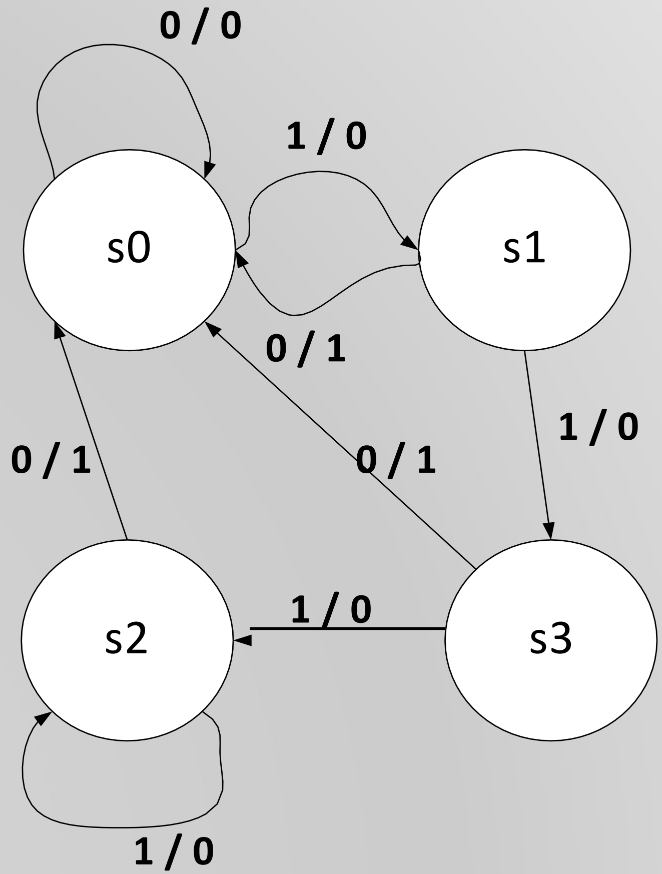
Παράδειγμα Mealy (2)

- Διαβάζεται ως ακολούθως: Όταν η παρούσα κατάσταση είναι S_i και η είσοδος I εφαρμοστεί, έχουμε έξοδο O και η επόμενη κατάσταση είναι η S_j.

  → Τιμές εισόδων/εξόδων πάνω στην κάθε ακμή.



Παράδειγμα Mealy (3)



Παράδειγμα Moore (1)

- Πιθανές Καταστάσεις = { 0, 1 } = { S0, S1 } → 2 κόμβοι στο διάγραμμα καταστάσεων.

Παρουσα Κατάσταση	Είσοδοι	Είσοδοι	Επόμενη Κατάσταση	Έξοδος
A(t)	X	Y	A(t + 1)	Z
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

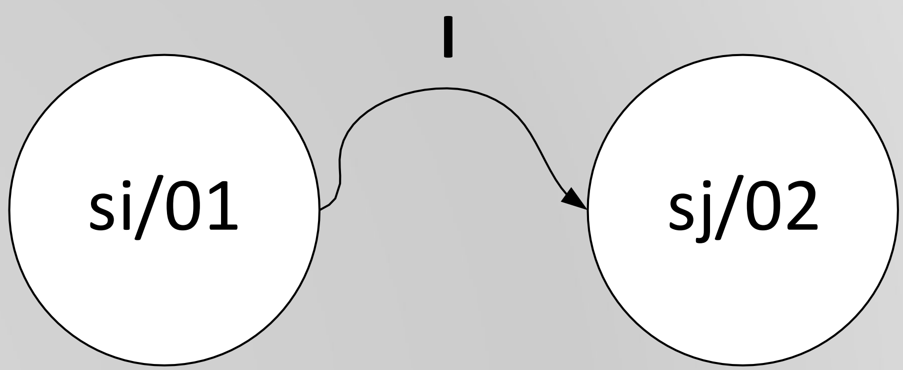
Παράδειγμα Moore (2)

- Πιθανές Καταστάσεις = { 0, 1 } = { S0, S1 } → 2 κόμβοι στο διάγραμμα καταστάσεων.

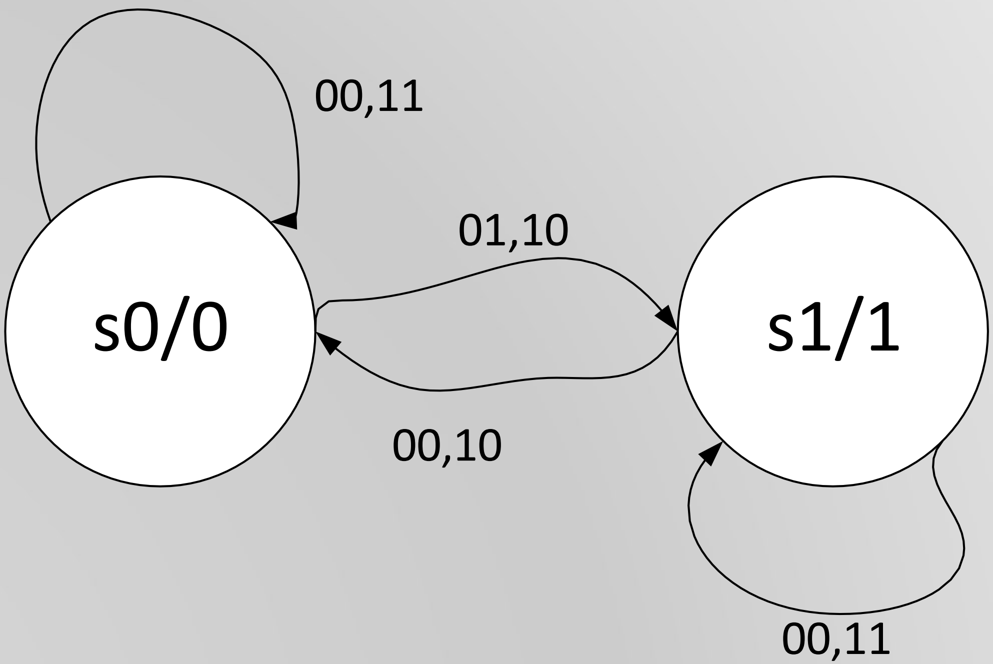
Παρουσα Κατάσταση	Είσοδοι	Είσοδοι	Επόμενη Κατάσταση	Έξοδος
A(t)	X	Y	A(t + 1)	Z
S0	0	0	S0	0
S0	0	1	S1	0
S0	1	0	S1	0
S0	1	1	S0	0
S1	0	0	S1	1
S1	0	1	S0	1
S1	1	0	S0	1
S1	1	1	S1	1

Παράδειγμα Moore (3)

- Διαβάζεται ως ακολούθως: Όταν η παρούσα κατάσταση είναι S_i με έξοδο O1 και η είσοδος I εφαρμοστεί, έχουμε έξοδο O2 και η επόμενη κατάσταση είναι η S_j.
  - → Τιμές εισόδων πάνω στην κάθε ακμή.
  - → Τιμές εξόδων στον κάθε κόμβο.

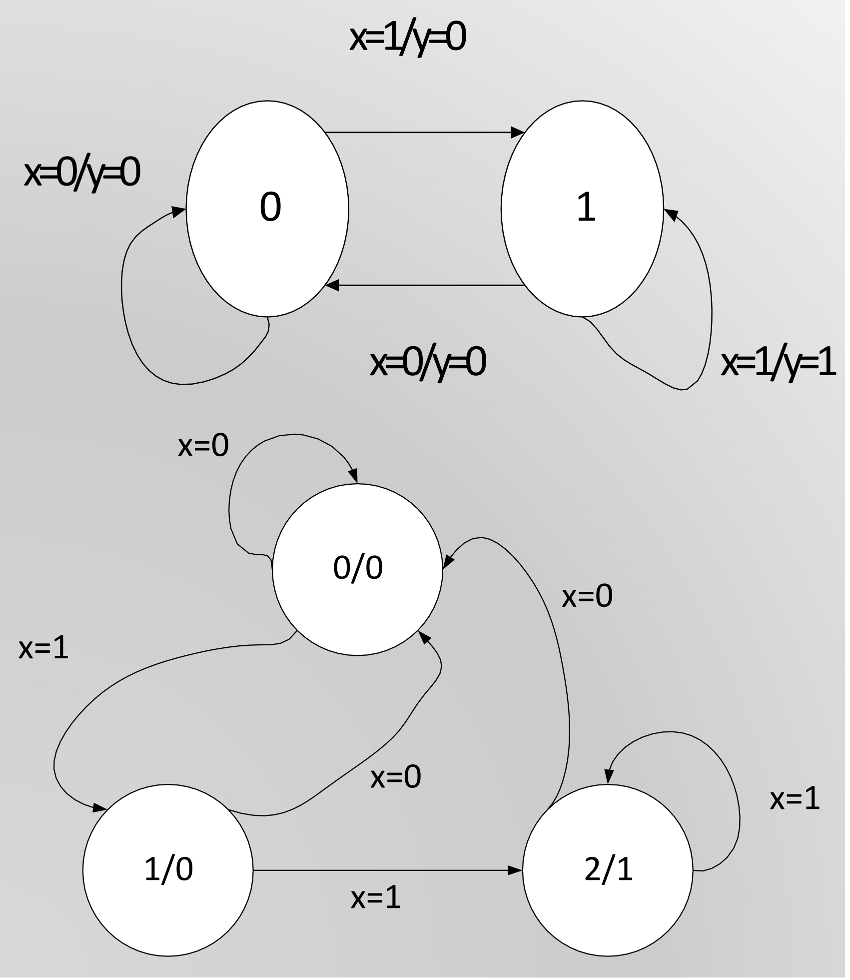


Παράδειγμα Moore (4)



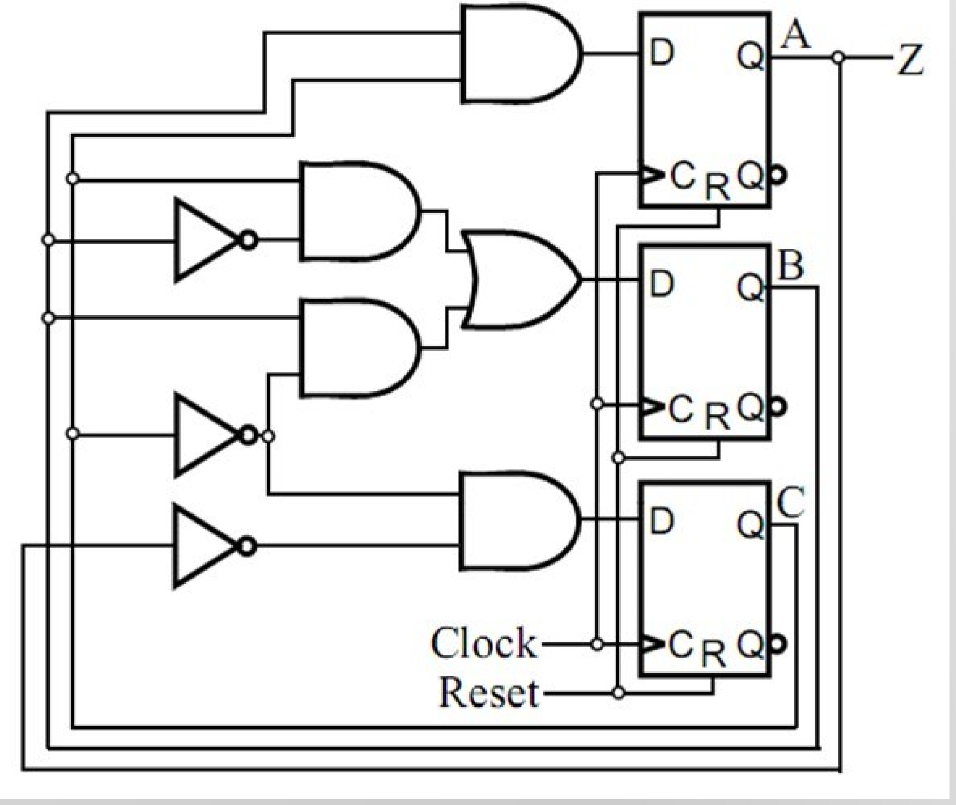
Άλλα παραδείγματα

- Μοντέλο Mealy: Αντιστοιχεί **τιμές εισόδων και καταστάσεων** σε εξόδους.
- Μοντέλο Moore: Αντιστοιχεί **καταστάσεις** σε εξόδους.
- Συμβαίνει το ίδιο με τα διαγράμματα.



Παράδειγμα ανάλυσης (1)

• Λογικό Διάγραμμα



Παράδειγμα ανάλυσης (2)

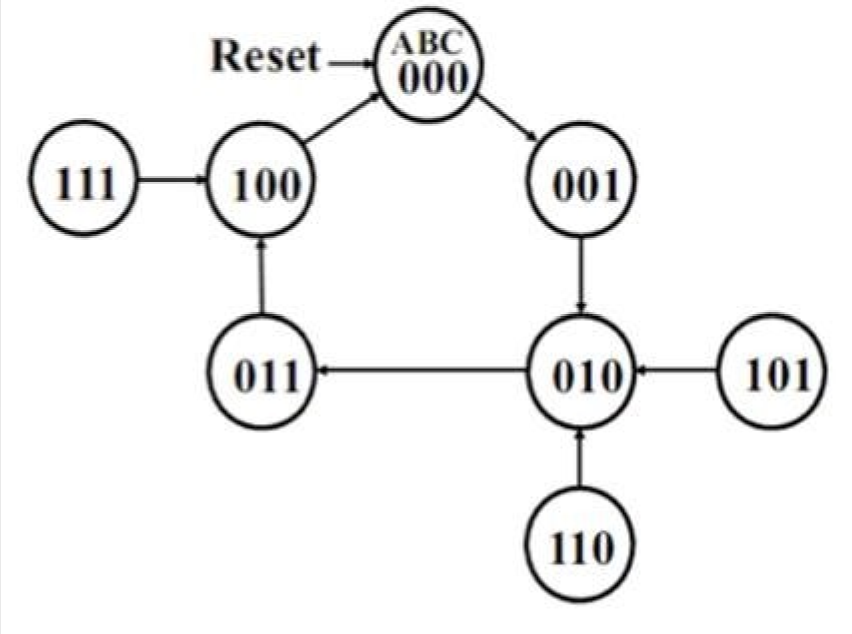
- Μεταβλητές:
  - Είσοδοι: Καμία
  - Έξοδοι: Z
  - Μεταβλητές καταστάσεων: A, B, C
- Αρχικοποίηση : Reset = 1 (A, B, C) = (0, 0, 0)
- Εξισώσεις:
  - $A(t+1) =$
  - $B(t+1) =$
  - $C(t+1) =$
  - $Z =$

Παράδειγμα ανάλυσης (3)

Παρούσα Κατάσταση	Επόμενη Κατάσταση	Έξοδος
A(t) B(t) C(t)	A(t+1) B(t+1) C(t+1)	Z
0 0 0		
0 0 1		
0 1 0		
0 1 1		
1 0 0		
1 0 1		
1 1 0		
1 1 1		

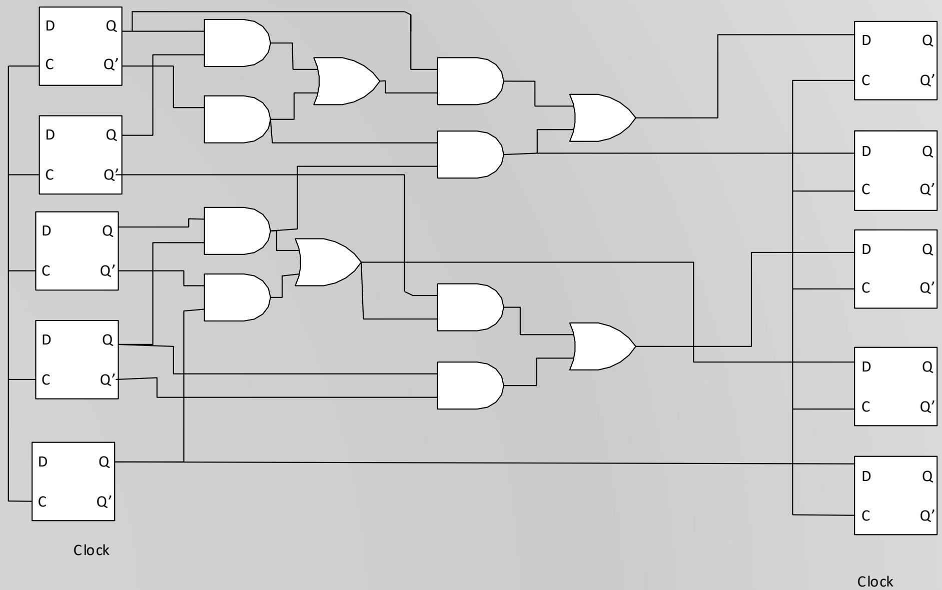
Παράδειγμα ανάλυσης (4)

- Ποιές καταστάσεις χρησιμοποιούνται;
- Ποιά η κύρια λειτουργία του κυκλώματος;



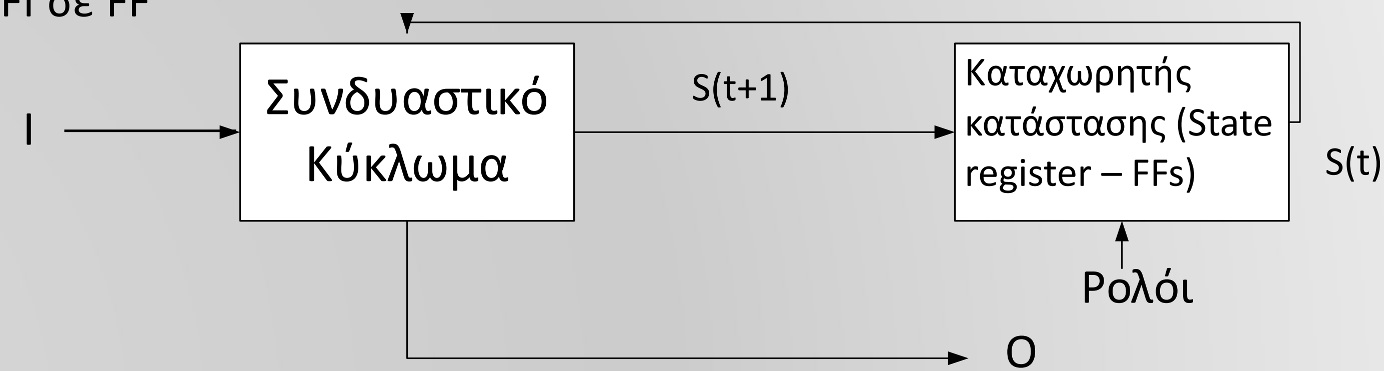
Ανάλυση Χρονισμού (1)

- Θεωρείστε ένα ακολουθιακό κύκλωμα το οποίο αποτελείται από ομάδες FFs, συνδεδεμένες μέσω συνδυαστικής λογικής.
- Αν η περίοδος του ρολογιού είναι πολύ μικρή, πιθανόν κάποιες αλλαγές στις τιμές των δεδομένων να ΜΗΝ προλάβουν να διαδοθούν μέσω της λογικής στις εισόδους των FFs ΠΡΙΝ ξεκινίσει το setup των FFs.



Ανάλυση Χρονισμού (2)

- Πρέπει να καθοριστεί η μέγιστη καθυστέρηση max_{pd} έτσ ώστεη περίοδος του ρολογιού να οριστεί ως $t_p >= max_{pd}$
- Για την μέγιστη καθυτέρηση, πρέπει να εξετάσουμε τα διάφορα μονοπάτια του κυκλώματος.
- Υπάρχουν 4^{ων} ειδών μονοπάτια
  - I/O – είσοδο σε έξοδο
  - I/FF – είσοδο σε FF
  - FF/O – FF σε έξοδο
  - FF/FF – Ff σε FF

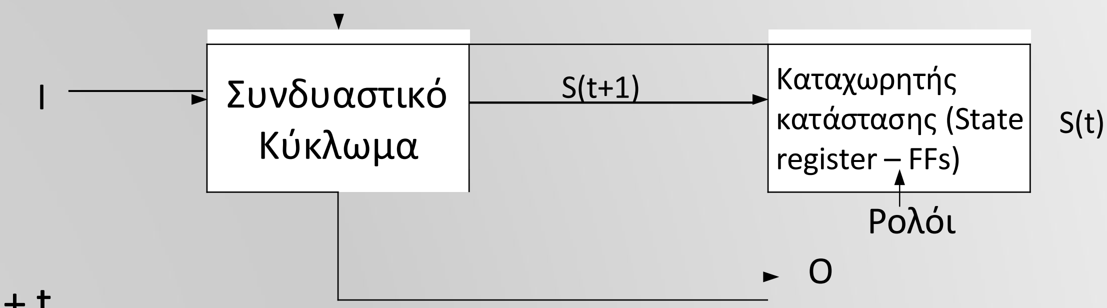


Ανάλυση Χρονισμού (3)

- Καθυστερήσεις:
  - $t_{pd,FF}$ = καθυστέρηση μετάδοσης FF
  - $t_{pd,COMP}$ = καθυστέρηση μετάδοσης συνδυαστικού μέρους
  - t_s = FF setup time
  - t_{slack} = πιθανόν επιπρόσθετος χρόνος που παρέχεται πέραν της καθυστέρησης ενός μονοπατιού

  →

  - I/O = $t_{pd,COMP}$
  - I/FF = $t_{pd,COMP} + t_s$
  - FF/O = $t_{pd,FF} + t_{pd,COMP}$
  - FF/FF = $t_{pd,FF} + t_{pd,COMP} + t_s$

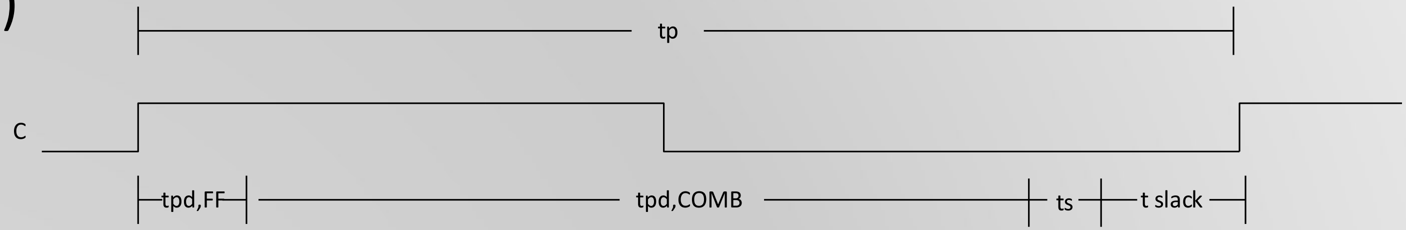


Ανάλυση Χρονισμού (4)

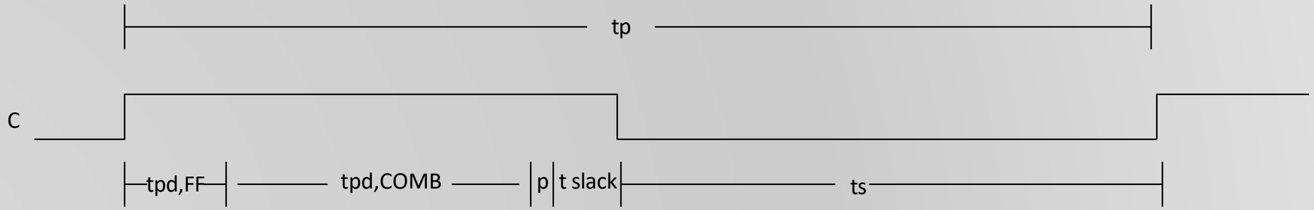
- Σκοπός μας είναι να ελαχιστοποιήσουμε την περίοδο του ρολογιού t_p (για να μεγιστοποιήσουμε την ταχύτητα).

→ $T_p >= max_{pd}$

→ $Max_{pd} = \max \{ t_{pd,FF} + t_{pd,COMB} + t_s \} = t_{pmin}$ (για όλα τα μονοπάτια FF / FF)



(α) Positive Edge triggered (Θετικές πυροδοτούμενες ακμές)



(β) Negative Pulse/Level triggered (Αρνητικός παλμός/πυροδοτούμενο επίπεδο)

Τέλος Ενότητας