

Έκδοση RISC_V 2025

Κεφάλαιο 3

Σχεδίαση ακολουθιακής λογικής

Καθηγητής Αντώνης Πασχάλης

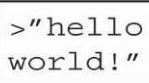
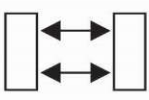
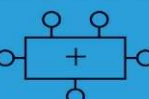
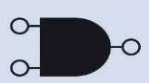
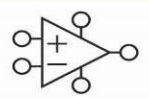
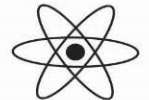


dscal

DIGITAL SYSTEMS & COMPUTER ARCHITECTURE LABORATORY

Περιεχόμενα κεφαλαίου 3

- Εισαγωγή
- Διασυζευγμένοι αντιστροφείς
- Latches και Flip-Flops
- Σύγχρονη ακολουθιακή λογική
- Μηχανές πεπερασμένων καταστάσεων (FSM)
 - Μηχανές Moore-Mealy
- Χρονισμός ακολουθιακής λογικής
 - Συγχρονιστές
- Παραλληλισμός

Application Software	
Operating Systems	
Architecture	
Micro-architecture	
Logic	
Digital Circuits	
Analog Circuits	
Devices	
Physics	

Ακολουθιακή λογική

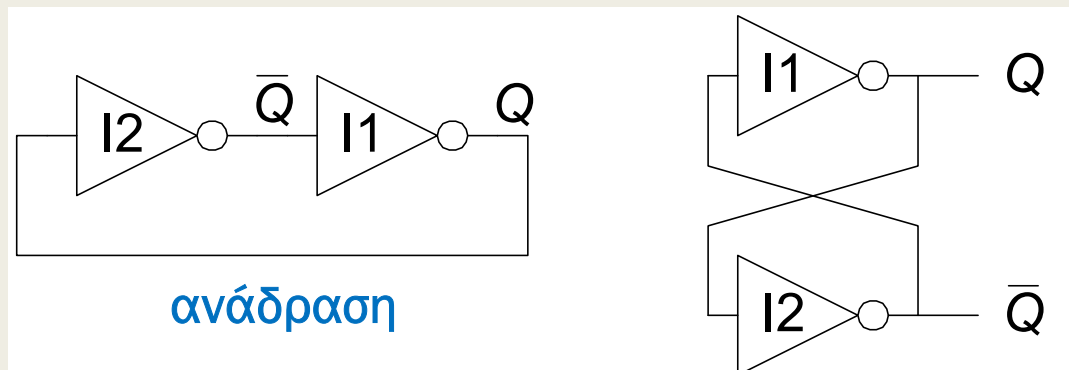
- Οι έξοδοι των **ακολουθιακών κυκλωμάτων** εξαρτώνται όχι μόνο από τις τρέχουσες τιμές των εισόδων, αλλά και από τις προηγούμενες τιμές αυτών
- Τα ακολουθιακά κυκλώματα **διαθέτουν μνήμη**, δηλαδή
 - «διυλίζουν» τις προηγούμενες τιμές εισόδων ώστε να διατηρούν αποθηκευμένη μία μικρότερη ποσότητα πληροφοριών, η οποία ονομάζεται **κατάσταση** (state)
 - για να επιτευχθεί η αποθήκευση μίας κατάστασης απαιτείται **ανάδραση από την έξοδο στην είσοδο**
- Ένα ακολουθιακό κύκλωμα με **N πιθανές καταστάσεις** διατηρεί αποθηκευμένη μία κατάσταση μεγέθους **από $\log_2 N$ μέχρι N bit**
 - Τα *latches* και τα *flip-flops* διατηρούν αποθηκευμένη μία κατάσταση μεγέθους ενός *bit*, δηλαδή έχουν δύο πιθανές καταστάσεις (0 και 1)
- Οι έξοδοι ενός ακολουθιακού κυκλώματος εξαρτώνται από την κατάστασή του και ενδεχομένως από τις τρέχουσες τιμές των εισόδων

Ιδιότητες των κυκλωμάτων με μνήμη

- Ένα κύκλωμα που διαθέτει μνήμη πρέπει να έχει τις εξής 3 ιδιότητες:
 - Πρέπει να μπορεί να **διατηρεί αποθηκευμένη μία κατάσταση**
 - Η αποθηκευμένη κατάσταση ονομάζεται τρέχουσα κατάσταση ή απλώς κατάσταση
 - Πρέπει να μπορούμε να **διαβάζουμε την αποθηκευμένη κατάσταση**
 - Πρέπει να μπορούμε να **αλλάζουμε την αποθηκευμένη κατάσταση σε μία επόμενη (νέα) κατάσταση**
 - Η επόμενη κατάσταση είναι συνάρτηση της τρέχουσας κατάστασης και ενδεχομένως και των τρεχουσών τιμών των εισόδων του ακολουθιακού κυκλώματος

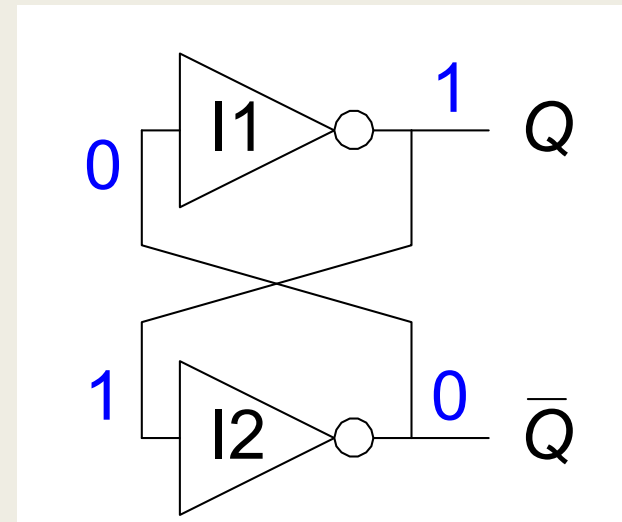
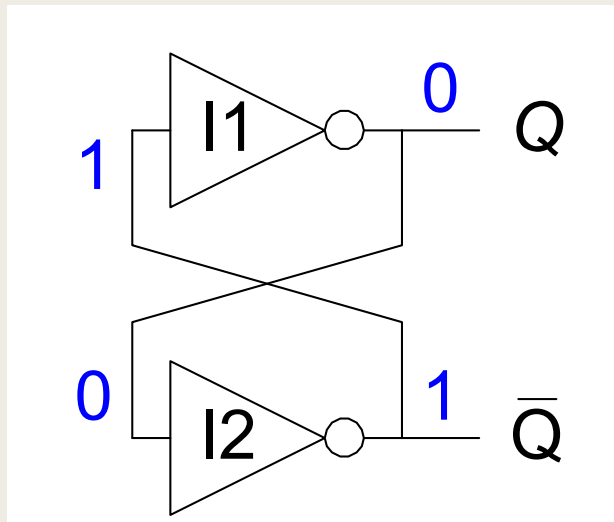
Διασυζευγμένοι αντιστροφείς

- Το θεμελιώδες δομικό στοιχείο μνήμης είναι ένα **δισταθές** (bistable) στοιχείο, δηλαδή ένα στοιχείο με δύο σταθερές καταστάσεις, το 0 και το 1
- Οι αντιστροφείς είναι **διασυζευγμένοι** (cross-coupled), που σημαίνει ότι η είσοδος του αντιστροφέα I1 αποτελεί έξοδο του αντιστροφέα I2 και το αντίστροφο, ώστε να δημιουργείται ανάδραση
- Το κύκλωμα **δεν διαθέτει εισόδους**, αλλά έχει **δύο εξόδους, τις Q και $\bar{Q}$** (διαβάζεται Q bar) που είναι **συμπληρωματικές**
- Το κύκλωμα διατηρεί αποθηκευμένη μία **κατάσταση Q** μεγέθους ενός bit, που ταυτίζεται με την έξοδο **Q**
 - Μπορούμε να αναφερόμαστε και στην μεταβλητή κατάστασης **Q** που έχει ως τιμή την τιμή της εξόδου **Q**



Λειτουργία διασυζευγμένων αντιστροφέων

- Όταν βρίσκεται στην σταθερή κατάσταση $Q = 0$
τότε οι έξοδοι είναι: $Q = 0$ και $\bar{Q} = 1$
- Όταν βρίσκεται στην σταθερή κατάσταση $Q = 1$
τότε οι έξοδοι είναι: $Q = 1$ και $\bar{Q} = 0$

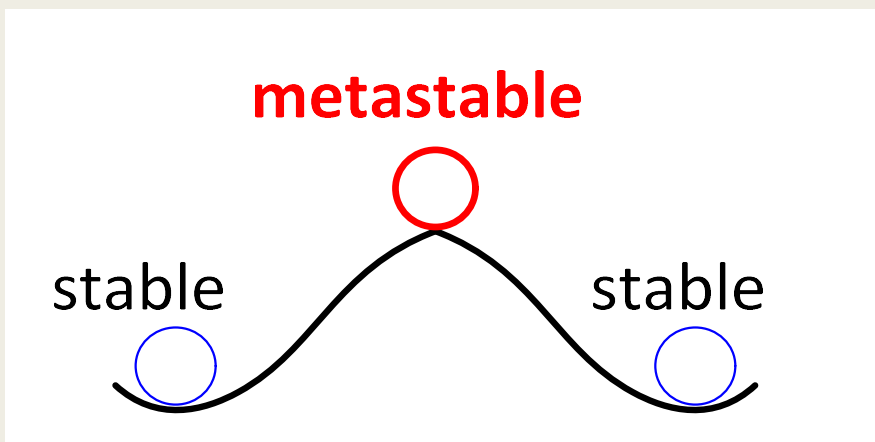
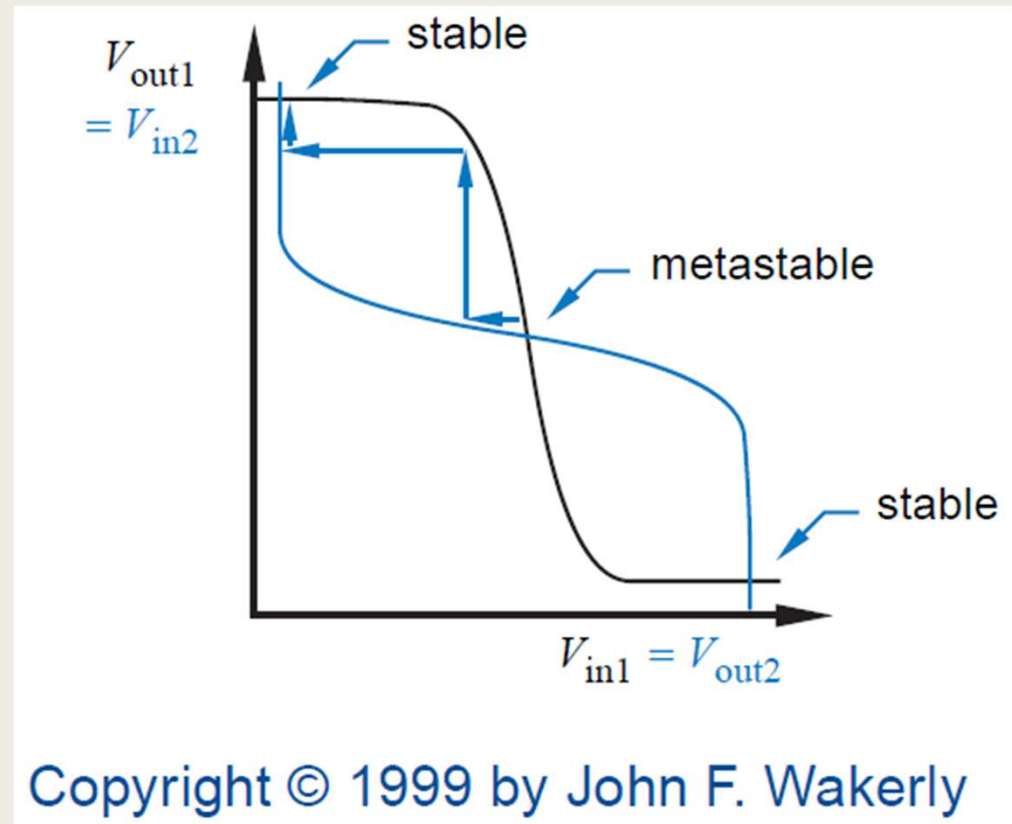
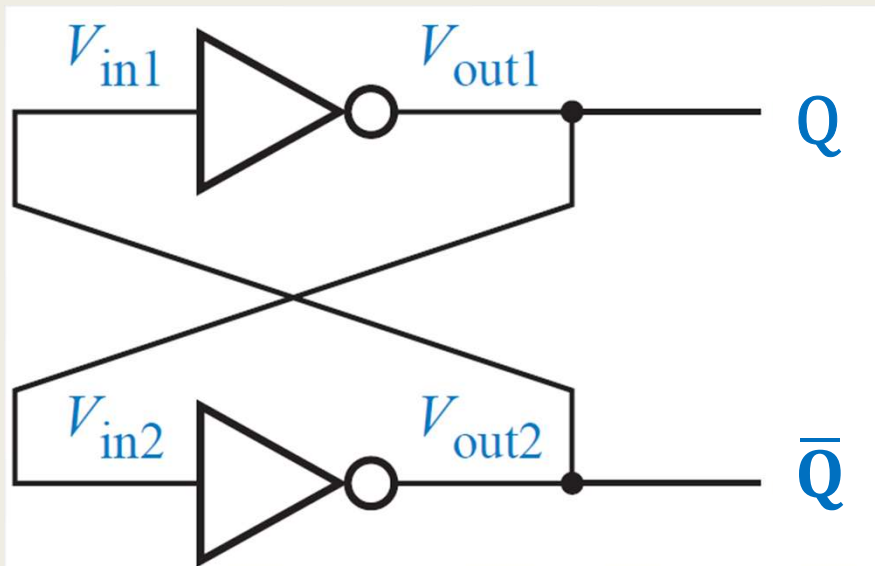


Λειτουργία διασυζευγμένων αντιστροφών

- Το κύκλωμα των διασυζευγμένων αντιστροφών έχει τις εξής ιδιότητες:
 - Μπορεί να διατηρεί αποθηκευμένη μία κατάσταση μεγέθους ενός *bit*
 - Την σταθερή κατάσταση 0 ή την σταθερή κατάσταση 1
 - Μπορούμε να διαβάσουμε την αποθηκευμένη κατάσταση
 - Εξετάζοντας την τιμή της εξόδου Q
 - Δεν μπορούμε να αλλάξουμε την αποθηκευμένη κατάσταση σε μία νέα κατάσταση
 - Το κύκλωμα **δεν έχει εισόδους**, αλλά όταν τίθεται σε λειτουργία πηγαίνει **ανεξέλεγκτα** στην σταθερή κατάσταση 0 ή στην σταθερή κατάσταση 1
 - Υπάρχει ενδεχόμενο για ένα χρονικό διάστημα να παραμείνει σε μία μια τρίτη πιθανή κατάσταση, την **μετασταθερή (metastable) κατάσταση**

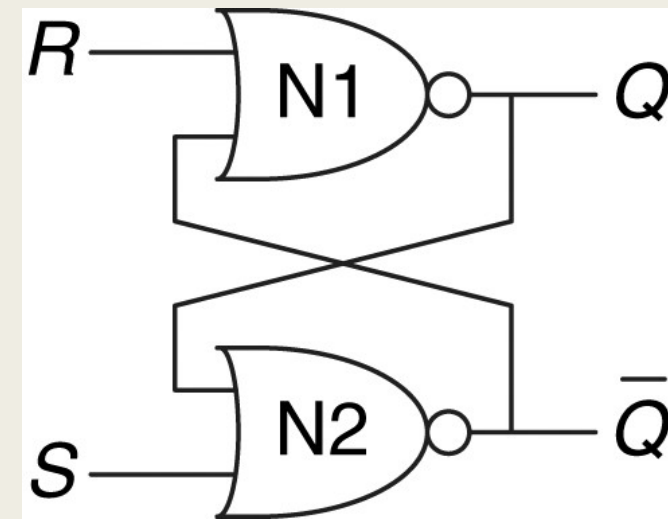
Μετασταθερή Κατάσταση

- Κατά την έναρξη της λειτουργίας του κυκλώματος των διασυζευγμένων αντιστροφέων, το κύκλωμα μπορεί να παραμείνει προσωρινά στην **μετασταθερή (metastable) κατάσταση**
 - όπου και οι δύο εξοδοι Q και $\bar{Q}$ βρίσκονται κατά προσέγγιση στο μέσο μεταξύ του 0 και του 1



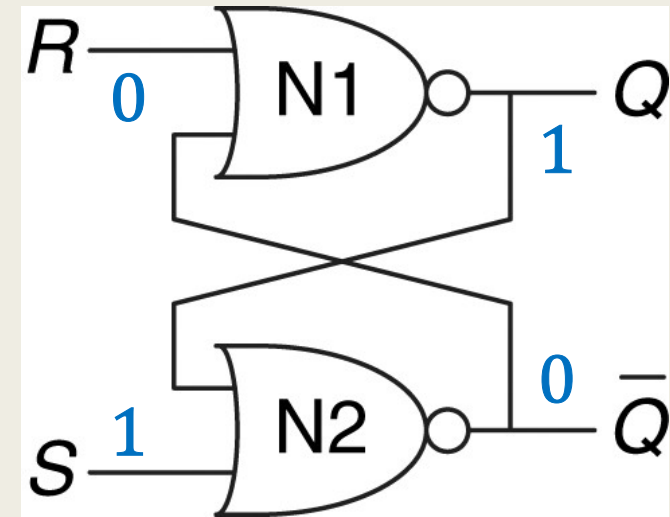
Set-Reset Latch (Μανδαλωτής SR)

- Το **Set-Reset latch** είναι το πιο απλό δισταθές ακολουθιακό κύκλωμα
- Αποτελείται από **δύο διασυνζευγμένες πύλες NOR**
- Η διαφορά του με τους διασυνζευγμένους αντιστροφείς είναι ότι η κατάσταση του μπορεί πλέον να ελεγχθεί μέσω των εισόδων **S (set)** και **R (reset)**
- Έχει **δύο εξόδους Q και $\bar{Q}$** που σε κανονική λειτουργία είναι **συμπληρωματικές**
- Το κύκλωμα διατηρεί αποθηκευμένη μία **κατάσταση Q** μεγέθους ενός bit, που ταυτίζεται με την έξοδο **Q**
 - την σταθερή κατάσταση 0 ($Q = 0, \bar{Q} = 1$) ή
 - την σταθερή κατάσταση 1 ($Q = 1, \bar{Q} = 0$)



Ενεργοποίηση της εισόδου Set

- Η **είσοδος S (set)**, όταν ενεργοποιηθεί (**$S = 1$**) θέτει την έξοδο της πύλης NOR N2 στην τιμή 0 (**$\bar{Q} = 0$**)
- Σε κανονική λειτουργία η έξοδος **$\bar{Q}$** πρέπει να είναι συμπληρωματική της εξόδου **Q** ,
 - άρα η έξοδος της πύλης NOR N1 πρέπει να τεθεί στην τιμή 1 (**$Q = 1$**)
 - άρα η είσοδος **R (reset)** πρέπει να τεθεί στην τιμή 0 (**$R = 0$**)
- Σε κανονική λειτουργία το κύκλωμα τίθεται **στην σταθερή κατάσταση 1** εάν
 - **$S = 1$ και $R = 0$**
- Γενικά, η **είσοδος S (set)**, όταν ενεργοποιηθεί, δίνει στο κύκλωμα την εντολή:
 - **Πήγαινε στην κατάσταση 1**



Πηγαίνει στην κατάσταση 1

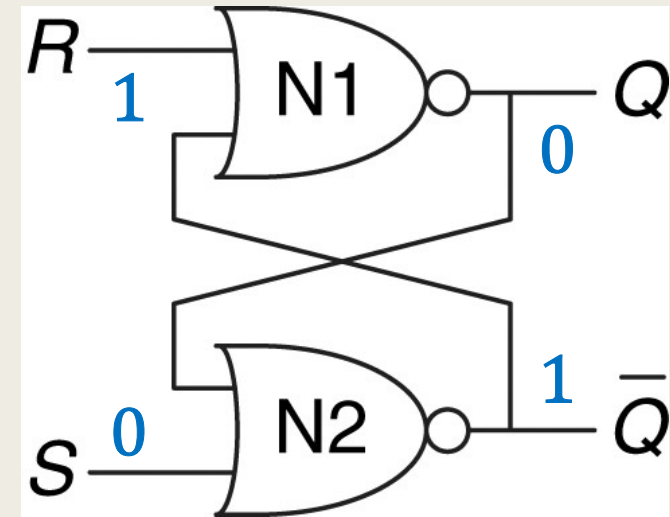
NOR			
A	B	Y	$\bar{Y}$
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	1

Ενεργοποίηση της εισόδου Reset

- Η **είσοδος R (reset)**, όταν ενεργοποιηθεί (**$R = 1$**) θέτει την έξοδο της πύλης NOR N1 στην τιμή 0 (**$Q = 0$**)
- Σε κανονική λειτουργία η έξοδος **$\bar{Q}$** πρέπει να είναι συμπληρωματική της εξόδου **Q** ,
 - άρα η έξοδος της πύλης NOR N2 πρέπει να τεθεί στην τιμή 1 (**$\bar{Q} = 1$**)
 - άρα η είσοδος **S (set)** πρέπει να τεθεί στην τιμή 0 (**$S = 0$**)

- Σε κανονική λειτουργία το κύκλωμα τίθεται **στην σταθερή κατάσταση 0** εάν
 - **$S = 0$ και $R = 1$**

- Γενικά, η **είσοδος R (reset)**, όταν ενεργοποιηθεί, δίνει στο κύκλωμα την εντολή:
 - **Επανάφερε την κατάσταση 0**

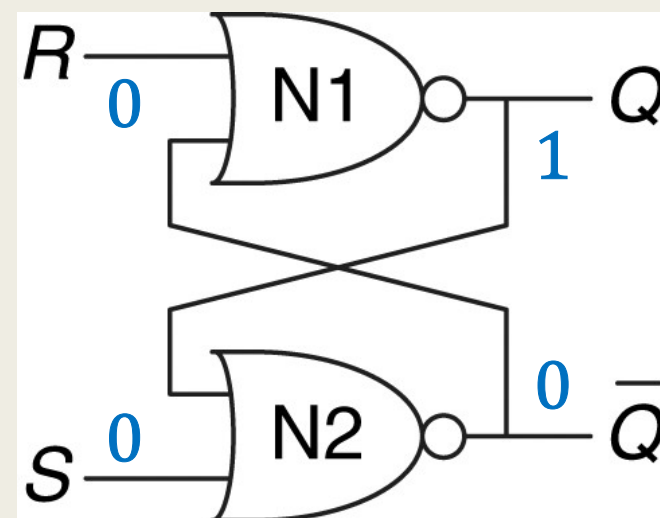


Πηγαίνει στην κατάσταση 0

- Η **είσοδος R (reset)**, όταν ενεργοποιηθεί (**$R = 1$**) θέτει την έξοδο της πύλης NOR N1 στην τιμή 0 (**$Q = 0$**)
- Σε κανονική λειτουργία η έξοδος **$\bar{Q}$** πρέπει να είναι συμπληρωματική της εξόδου **Q** ,
 - άρα η έξοδος της πύλης NOR N2 πρέπει να τεθεί στην τιμή 1 (**$\bar{Q} = 1$**)
 - άρα η είσοδος **S (set)** πρέπει να τεθεί στην τιμή 0 (**$S = 0$**)
- Σε κανονική λειτουργία το κύκλωμα τίθεται **στην σταθερή κατάσταση 0** εάν
 - **$S = 0$ και $R = 1$**
- Γενικά, η **είσοδος R (reset)**, όταν ενεργοποιηθεί, δίνει στο κύκλωμα την εντολή:
 - **Επανάφερε την κατάσταση 0**

Απενεργοποίηση και των δύο εισόδων Set και Reset

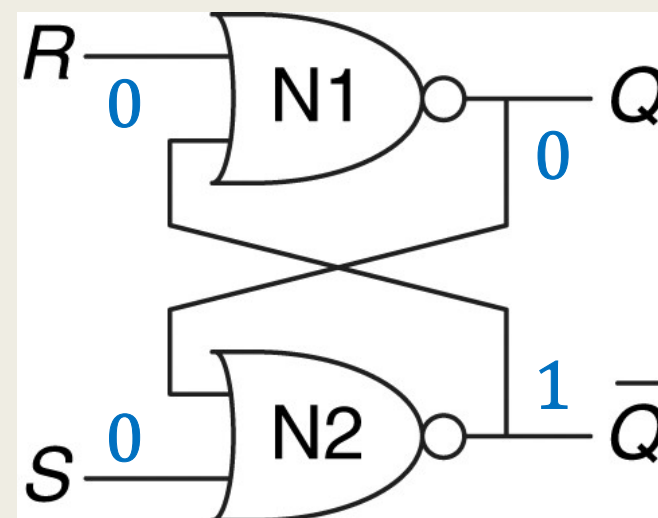
- Η είσοδος **S (set)**, όταν απενεργοποιηθεί (**S = 0**) εξαρτά την έξοδο της πύλης NOR N2 από την προηγούμενη τιμή της εξόδου **Q** (**$\bar{Q} = \text{NOT } Q_{\text{prev}}$**)
- Η είσοδος **R (reset)**, όταν απενεργοποιηθεί (**R = 0**) εξαρτά την έξοδο της πύλης NOR N1 από την προηγούμενη τιμή της εξόδου **$\bar{Q}$** (**$Q = \text{NOT } \bar{Q}_{\text{prev}}$**)
- Εάν το κύκλωμα βρισκόταν στην σταθερή κατάσταση **1** (ήταν **S = 1** και **R = 0**) ΤΟΤΕ **$Q_{\text{prev}} = 1$** και **$\bar{Q}_{\text{prev}} = 0$**
 - άρα **$\bar{Q} = \text{NOT } 1 = 0$** και **$Q = \text{NOT } 0 = 1$**
 - δηλαδή διατηρεί αποθηκευμένη την προηγούμενη κατάσταση **1**



Διατηρεί αποθηκευμένη
την προηγούμενη
κατάσταση 1

Απενεργοποίηση και των δύο εισόδων Set και Reset

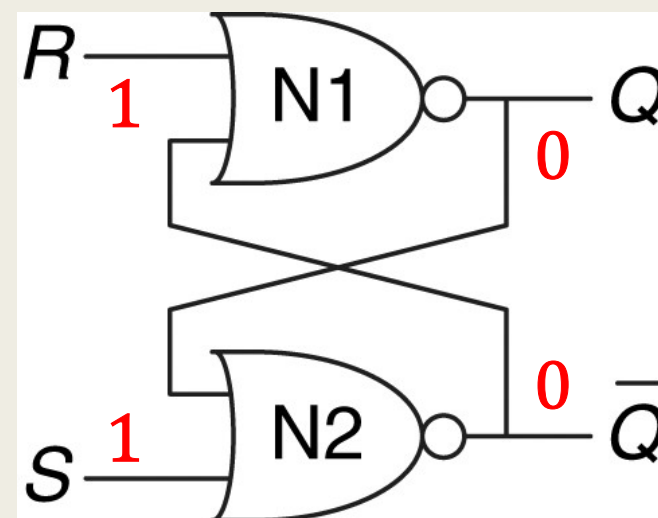
- Η είσοδος **S (set)**, όταν απενεργοποιηθεί (**S = 0**) εξαρτά την έξοδο της πύλης NOR N2 από την προηγούμενη τιμή της εξόδου **Q** (**$\bar{Q} = \text{NOT } Q_{\text{prev}}$**)
- Η είσοδος **R (reset)**, όταν απενεργοποιηθεί (**R = 0**) εξαρτά την έξοδο της πύλης NOR N1 από την προηγούμενη τιμή της εξόδου **$\bar{Q}$** (**$Q = \text{NOT } \bar{Q}_{\text{prev}}$**)
- Εάν το κύκλωμα βρισκόταν στην σταθερή κατάσταση 0 (ήταν **S = 0** και **R = 1**) ΤΟΤΕ **$Q_{\text{prev}} = 0$** και **$\bar{Q}_{\text{prev}} = 1$**
 - άρα **$\bar{Q} = \text{NOT } 0 = 1$** και **$Q = \text{NOT } 1 = 0$**
 - δηλαδή διατηρεί αποθηκευμένη την προηγούμενη κατάσταση 0



Διατηρεί αποθηκευμένη
την προηγούμενη
κατάσταση 0

Ενεργοποίηση και των δύο εισόδων Set και Reset

- Η **είσοδος S (set)**, όταν ενεργοποιηθεί (**$S = 1$**) θέτει την έξοδο της πύλης NOR N2 στην τιμή 0 (**$\bar{Q} = 0$**)
- Η **είσοδος R (reset)**, όταν ενεργοποιηθεί (**$R = 1$**) θέτει την έξοδο της πύλης NOR N1 στην τιμή 0 (**$Q = 0$**)
- Η έξοδος **$\bar{Q}$** παύει να είναι συμπληρωματική της εξόδου **Q** και το κύκλωμα πηγαίνει σε μία **μη έγκυρη κατάσταση ($Q = \bar{Q}$)**

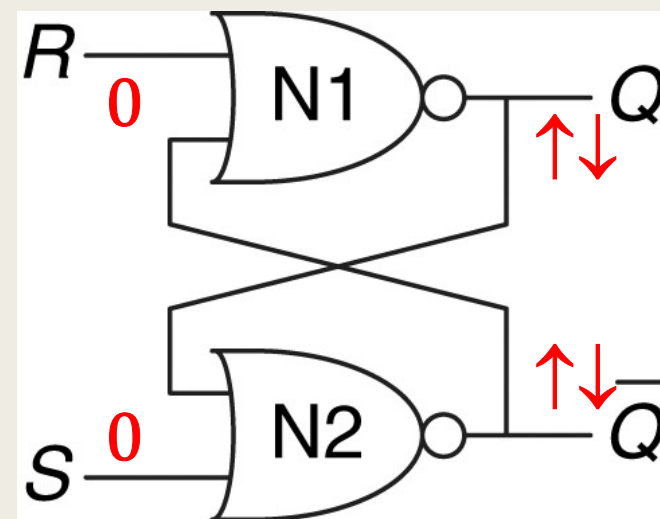


Πηγαίνει σε μη έγκυρη κατάσταση

NOR			
A	B	Y	$\bar{Y}$
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	1

Απενεργοποίηση και των δύο εισόδων Set και Reset

- Η είσοδος **S (set)**, όταν απενεργοποιηθεί (**S = 0**) εξαρτά την έξοδο της πύλης NOR N2 από την προηγούμενη τιμή της εξόδου **Q** (**$\bar{Q} = \text{NOT } Q_{\text{prev}}$**)
- Η είσοδος **R (reset)**, όταν απενεργοποιηθεί (**R = 0**) εξαρτά την έξοδο της πύλης NOR N1 από την προηγούμενη τιμή της εξόδου **$\bar{Q}$** (**$Q = \text{NOT } \bar{Q}_{\text{prev}}$**)
- Εάν το κύκλωμα βρισκόταν στην μη έγκυρη κατάσταση (ήταν **S = 1** και **R = 1**) ΤΟΤΕ **$Q_{\text{prev}} = 0$** και **$\bar{Q}_{\text{prev}} = 0$**
 - άρα **$\bar{Q} = \text{NOT } 0 = 1$** και **$Q = \text{NOT } 0 = 1$**
 - το κύκλωμα θα εμφανίσει ταλαντώσεις ή θα βρεθεί στη μετασταθερή κατάσταση μέχρι ανεξέλεγκτα να ισορροπήσει σε μία σταθερή κατάσταση (0 ή 1)



Πηγαίνει σε ταλαντώσεις ή στη μετασταθερή κατάσταση

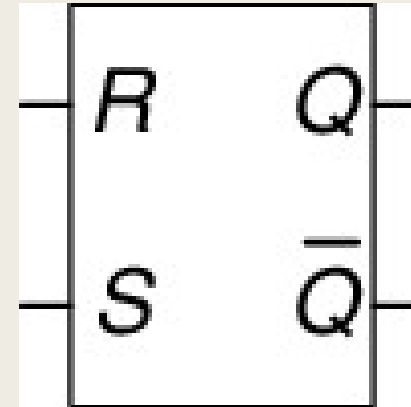
NOR			
A	B	Y	$\bar{Y}$
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	1

Set-Reset (S-R) Latch

■ Πίνακας Αλήθειας

S	R	Q	$\bar{Q}$
0	0	Q_{prev}	$\overline{Q_{\text{prev}}}$
0	1	0	1
1	0	1	0
1	1	0	0

■ Σύμβολο



■ Σύνοψη

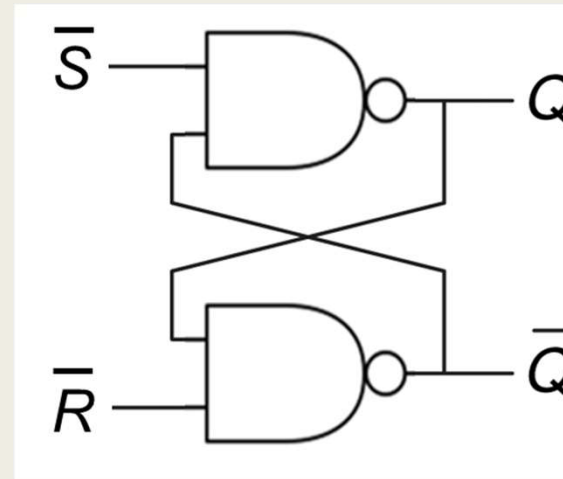
- Το S-R Latch ελέγχει την κατάστασή του μέσω των εισόδων **S (set)** και **R (reset)**
- **S = 1** και **R = 0** $\Rightarrow$ Πηγαίνει στην κατάσταση **Q = 1**
- **S = 0** και **R = 1** $\Rightarrow$ Πηγαίνει στην κατάσταση **Q = 0**
- Το κύκλωμα διατηρεί αποθηκευμένη μία **κατάσταση Q** μεγέθους ενός bit, που ταυτίζεται με την έξοδο **Q**
- **S = 0** και **R = 0** $\Rightarrow$ Αποθηκεύει την προηγούμενη κατάσταση **Q = Q_{prev}**
- **S = 1** και **R = 1** $\Rightarrow$ **Μη έγκυρη κατάσταση ($Q = \bar{Q}$)**
Κάτι πρέπει να κάνουμε ώστε να την αποφύγουμε!

Set-Reset (S-R) Latch (active LOW)

■ Πίνακας Αλήθειας

$\bar{S}$	$\bar{R}$	Q	$\bar{Q}$
1	1	Q_{prev}	$\bar{Q}_{\text{prev}}$
1	0	0	1
0	1	1	0
0	0	1	1

■ Σχηματικό



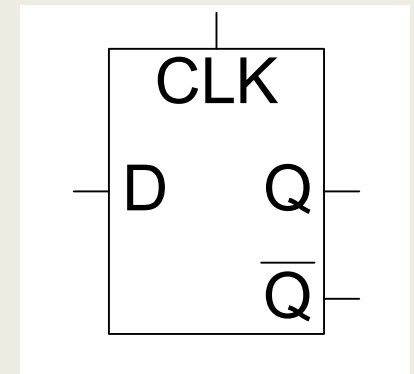
■ Σύνοψη

- Το S-R Latch (active LOW) ελέγχει την κατάστασή του μέσω των εισόδων $\bar{S}$ (set_bar) και $\bar{R}$ (reset_bar)
- $\bar{S} = 0$ και $\bar{R} = 1 \Rightarrow$ Πηγαίνει στην κατάσταση $Q = 1$
- $\bar{S} = 1$ και $\bar{R} = 0 \Rightarrow$ Πηγαίνει στην κατάσταση $Q = 0$
- Το κύκλωμα διατηρεί αποθηκευμένη μία κατάσταση Q μεγέθους ενός bit, που ταυτίζεται με την έξοδο Q
- $\bar{S} = 1$ και $\bar{R} = 1 \Rightarrow$ Αποθηκεύει την προηγούμενη κατάσταση $Q = Q_{\text{prev}}$
- $\bar{S} = 0$ και $\bar{R} = 0 \Rightarrow$ Μη έγκυρη κατάσταση ($Q = \bar{Q}$)
Κάτι πρέπει να κάνουμε ώστε να την αποφύγουμε!

D (Data) Latch (Μανδαλωτής D)

- Το **D (Data) Latch** επιλύει το πρόβλημα της μη έγκυρης κατάστασης ($Q = \bar{Q}$)
- Το κύκλωμα έχει δύο εισόδους:
 - την είσοδο του ρολογιού **CLK** που ελέγχει **πότε** πρέπει να αλλάξει η κατάσταση
 - την είσοδο δεδομένων **D** που ελέγχει **τί τιμή** πρέπει να έχει η κατάσταση **Q**
- Έχει **δύο εξόδους Q και $\bar{Q}$** που είναι πάντα **συμπληρωματικές**
- Το κύκλωμα διατηρεί αποθηκευμένη μία **κατάσταση Q** μεγέθους ενός bit, που ταυτίζεται με την έξοδο **Q**
 - την σταθερή κατάσταση 0 ή
 - την σταθερή κατάσταση 1

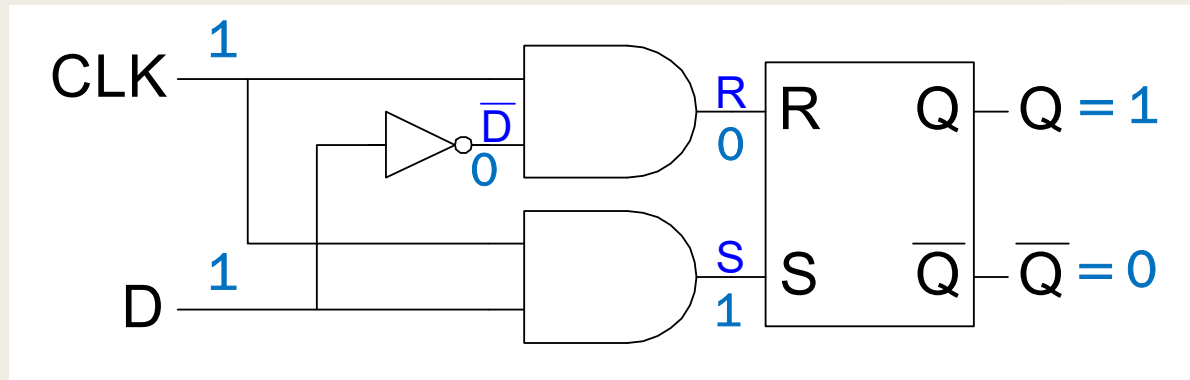
- Σύμβολο



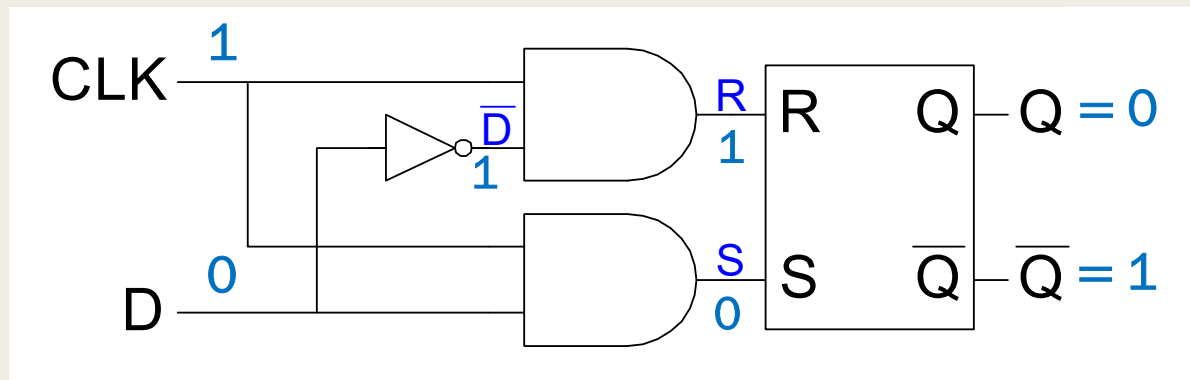
Λειτουργία D Latch (CLK = 1)

■ Όταν CLK = 1

- $D = 1 \Rightarrow S = 1$ και $R = 0 \Rightarrow$ Πηγαίνει στην κατάσταση $Q = 1$



- $D = 0 \Rightarrow S = 0$ και $R = 1 \Rightarrow$ Πηγαίνει στην κατάσταση $Q = 0$

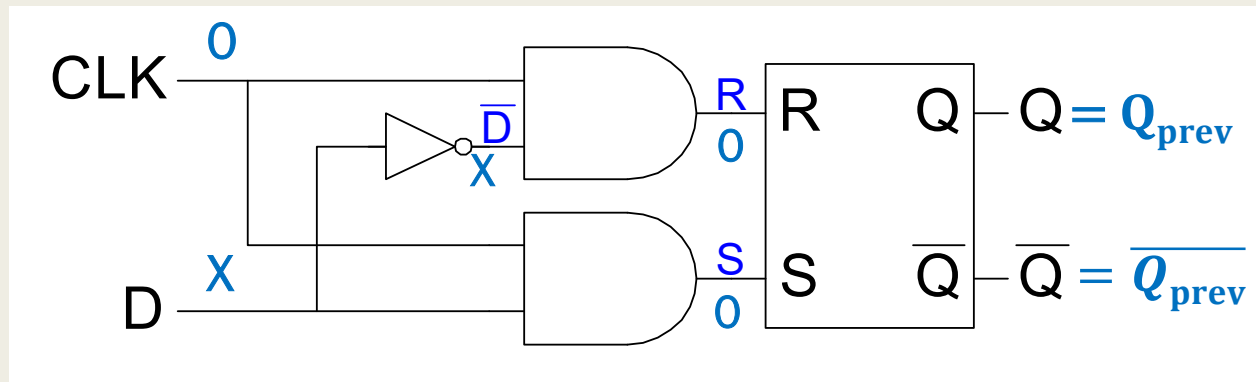


Λειτουργία D Latch (CLK = 0)

- Όταν $CLK = 0$

- $D = X \Rightarrow S = 0$ και $R = 0 \Rightarrow$

Αποθηκεύει την προηγούμενη κατάσταση $Q = Q_{prev}$

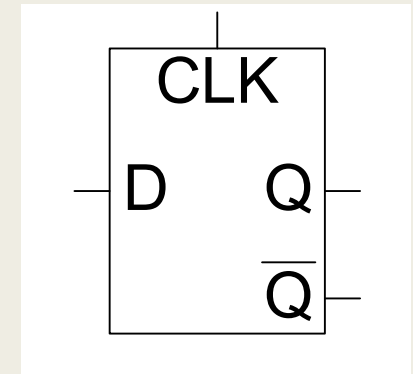


D Latch

■ Πίνακας Αλήθειας

CLK	D	D'	S	R	Q	$\bar{Q}$
0	X	X'	0	0	Q_{prev}	$\overline{Q_{\text{prev}}}$
1	0	1	0	1	0	1
1	1	0	1	0	1	0

■ Σύμβολο



■ Σύνοψη

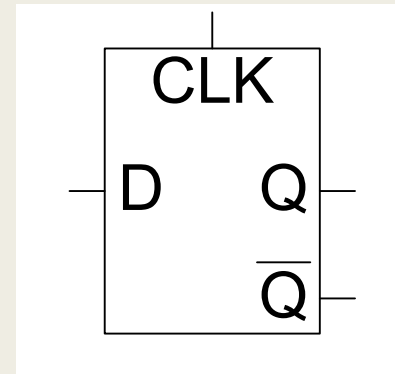
- Το D Latch εξαρτάται από το επίπεδο (την τιμή 0 ή 1) του σήματος CLK
- Το D Latch ενημερώνει την κατάστασή του συνεχώς για όσο διάστημα ισχύει $CLK = 1$ (είναι διάφανο και συμπεριφέρεται ως απομονωτής)
- $CLK = 1$ και $D = 1 \Rightarrow$ Πηγαίνει στην κατάσταση $Q = 1$
- $CLK = 1$ και $D = 0 \Rightarrow$ Πηγαίνει στην κατάσταση $Q = 0$
- Το D Latch διατηρεί αποθηκευμένη μία κατάσταση Q μεγέθους ενός bit, που ταυτίζεται με την έξοδο Q για όσο διάστημα ισχύει $CLK = 0$
- $CLK = 0$ και $D = X \Rightarrow$ Αποθηκεύει την προηγούμενη κατάσταση $Q = Q_{\text{prev}}$
- Δεν πηγαίνει σε μη έγκυρη κατάσταση ($Q = \bar{Q}$)

D Latch

■ Πίνακας Αλήθειας

CLK	D	D'	S	R	Q	$\bar{Q}$
0	X	X'	0	0	Q_{prev}	$\overline{Q_{prev}}$
1	0	1	0	1	0	1
1	1	0	1	0	1	0

■ Σύμβολο

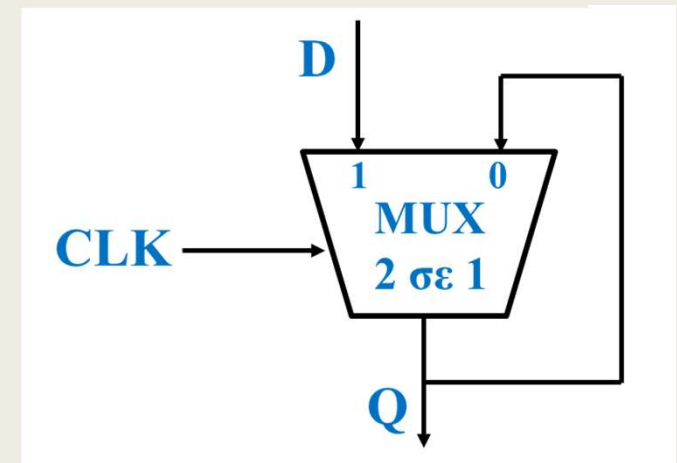


■ Υλοποίηση με πολυπλέκτη 2 σε 1

CLK	D	Q_{prev}	Q
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

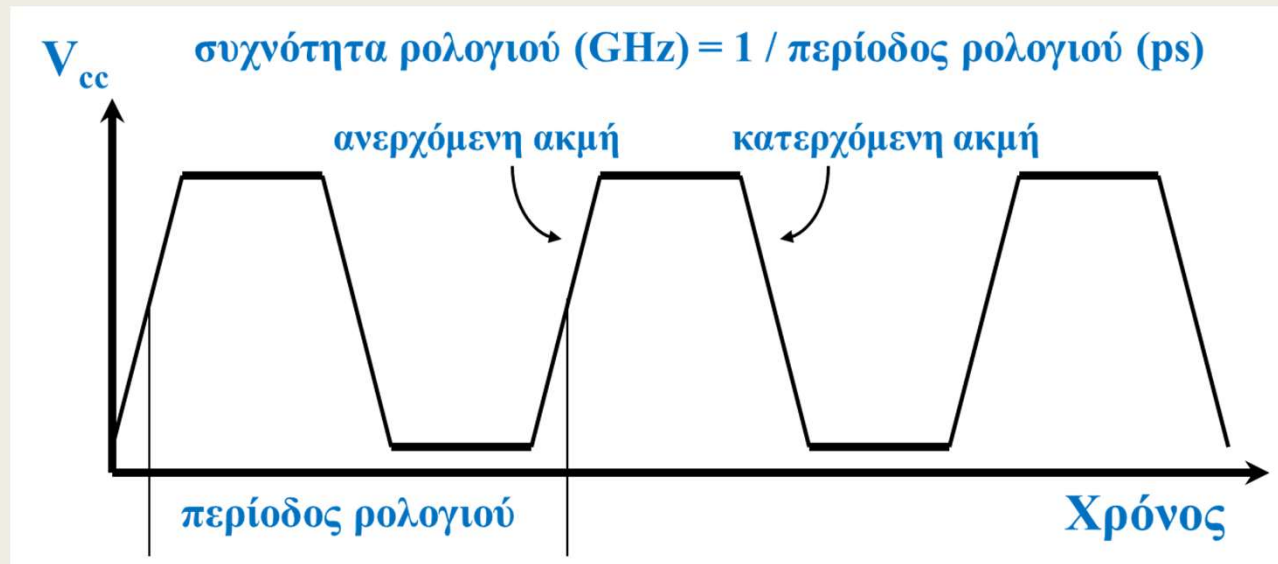
$CLK \ D$		00	01	11	10
Q_{prev}	0	0	0	1	0
	1	1	1	1	0

$$Q = CLK D + \overline{CLK} Q_{prev}$$



Το σήμα CLK: Latches vs Flip-Flops

- Το **σήμα του ρολογιού CLK** είναι ένας τετραγωνικός παλμός με εναλλασσόμενα επίπεδα 0 και 1 και με ανερχόμενες (από το 0 στο 1) και κατερχόμενες (από το 1 στο 0) ακμές
 - Τα δισταθή στοιχεία που μπορούν να ενημερώνουν συνεχώς την κατάστασή τους σε ένα από τα επίπεδα (0 ή 1) του ρολογιού αποκαλούνται **Latches**
 - Τα δισταθή στοιχεία που ενημερώνουν την κατάστασή τους μόνο σε μια ακμή του ρολογιού, όπως για παράδειγμα στην ανερχόμενη ακμή του ρολογιού, αποκαλούνται **Flip-Flops**

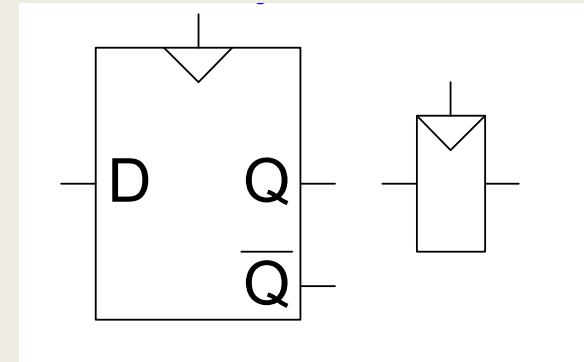


D (Data) Flip-Flop

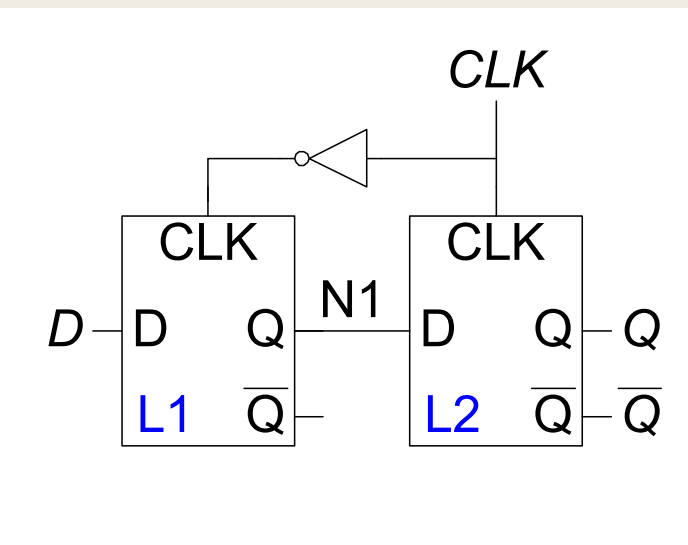
- Ένα **D Flip-Flop** αποτελείται από **δύο D Latches** συνδεδεμένα στην σειρά που ελέγχονται από **συμπληρωματικά CLK**
 - Το πρώτο D Latch (L1) ονομάζεται **αφέντης** (master)
 - Το δεύτερο D Latch (L2) ονομάζεται **σκλάβος** (slave)
 - Ο N1 είναι ενδιάμεσος κόμβος
- Τα D Flip-Flops είναι επίσης γνωστά ως:
 - Flip-Flop **αφέντη-σκλάβου**
 - Flip-Flop **ενεργοποιούμενα στις ακμές** (του CLK).

Το τρίγωνο στην είσοδο CLK στο σύμβολο του Flip-Flop αναπαριστά ένα κύκλωμα που ενεργοποιείται στην ακμή του CLK

■ Σύμβολο



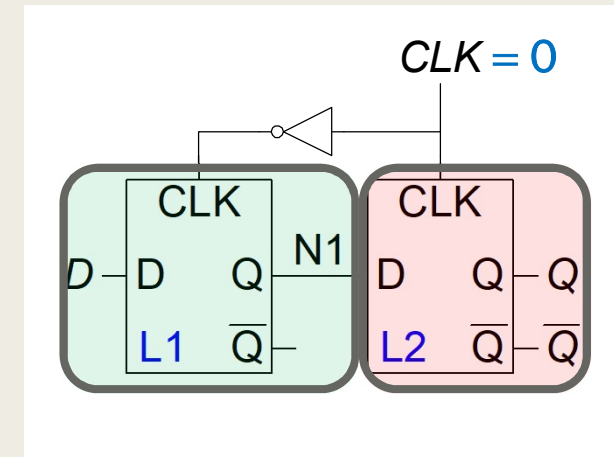
■ Σχηματικό διάγραμμα



Λειτουργία D Flip-Flop

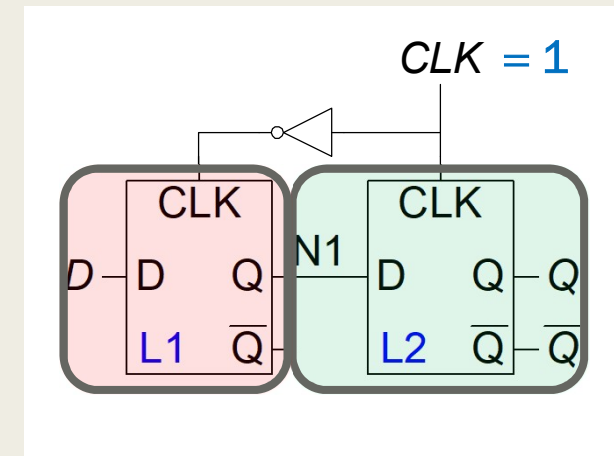
■ Όταν $CLK = 0$

- το D-Latch αφέντης (L1) είναι **διαφανές**
- το D-Latch σκλάβος (L2) είναι **αδιαφανές**
- άρα, η τιμή στην είσοδο δεδομένων D μεταφέρεται μέσω του L1 στον κόμβο N1, χωρίς να επηρεάζεται η έξοδος Q του L2



■ Όταν $CLK = 1$

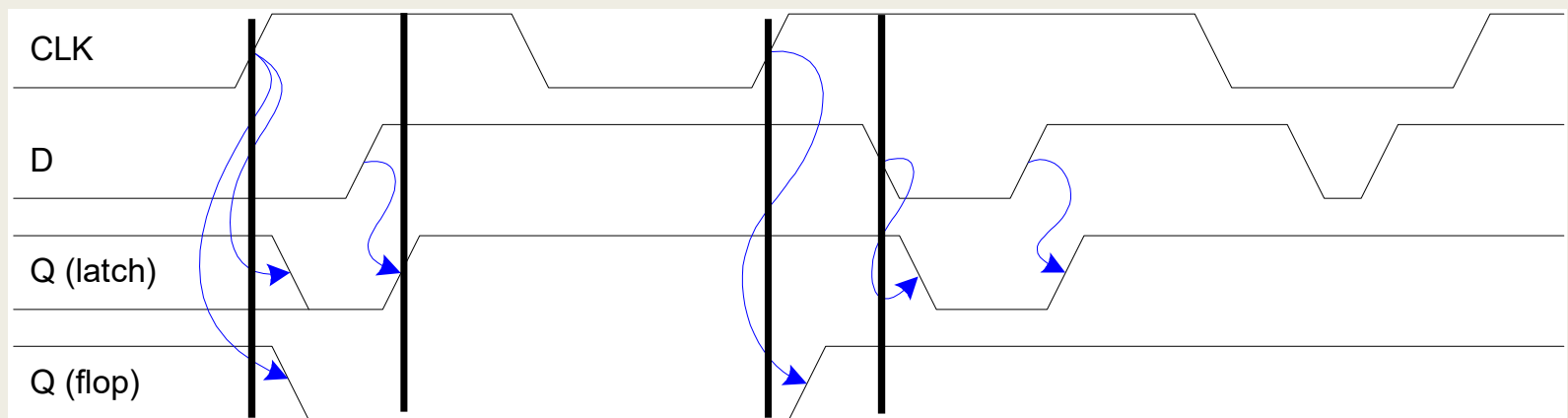
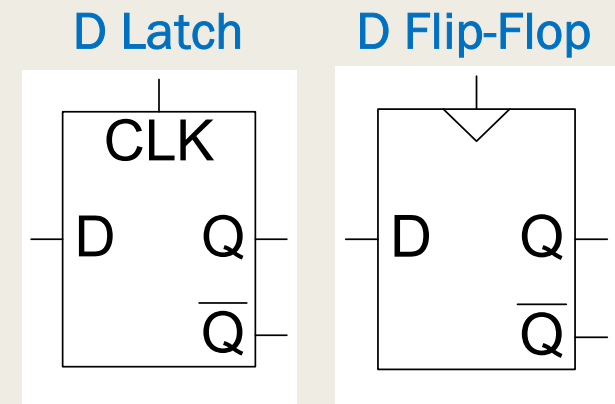
- το D-Latch αφέντης (L1) είναι **αδιαφανές**
- το D-Latch σκλάβος (L2) είναι **διαφανές**
- άρα, η τιμή στον κόμβο N1 μεταφέρεται στην έξοδο Q του L2, ενώ η τιμή στον κόμβο N1 παραμένει σταθερή γιατί παύει να ελέγχεται από την είσοδο δεδομένων D



- Η τιμή που υπήρχε στην είσοδο δεδομένων D αμέσως πριν η τιμή του CLK μεταβεί από το 0 στο 1 αντιγράφεται στην έξοδο Q αμέσως μετά την ανερχόμενη ακμή του CLK (**το D Flip-Flop ενημερώνει την κατάστασή του**)
- Σε όλες τις άλλες περιπτώσεις, **το D Flip-Flop διατηρεί την αποθηκευμένη κατάσταση του**, επειδή υπάρχει πάντα ένας αδιαφανές D Latch που μπλοκάρει τη διαδρομή μεταξύ της εισόδου D και της εξόδου Q

D Latch vs D Flip-Flop*

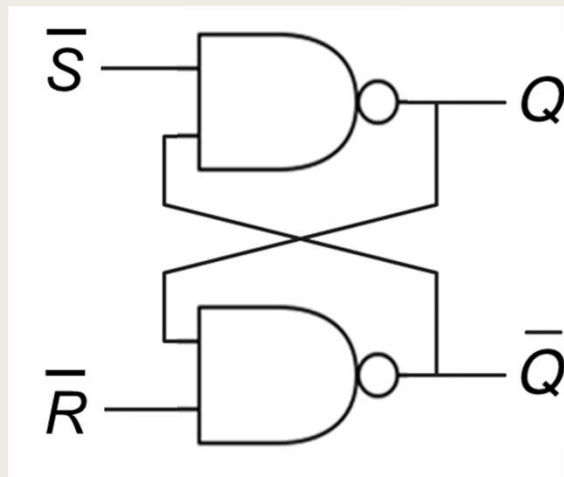
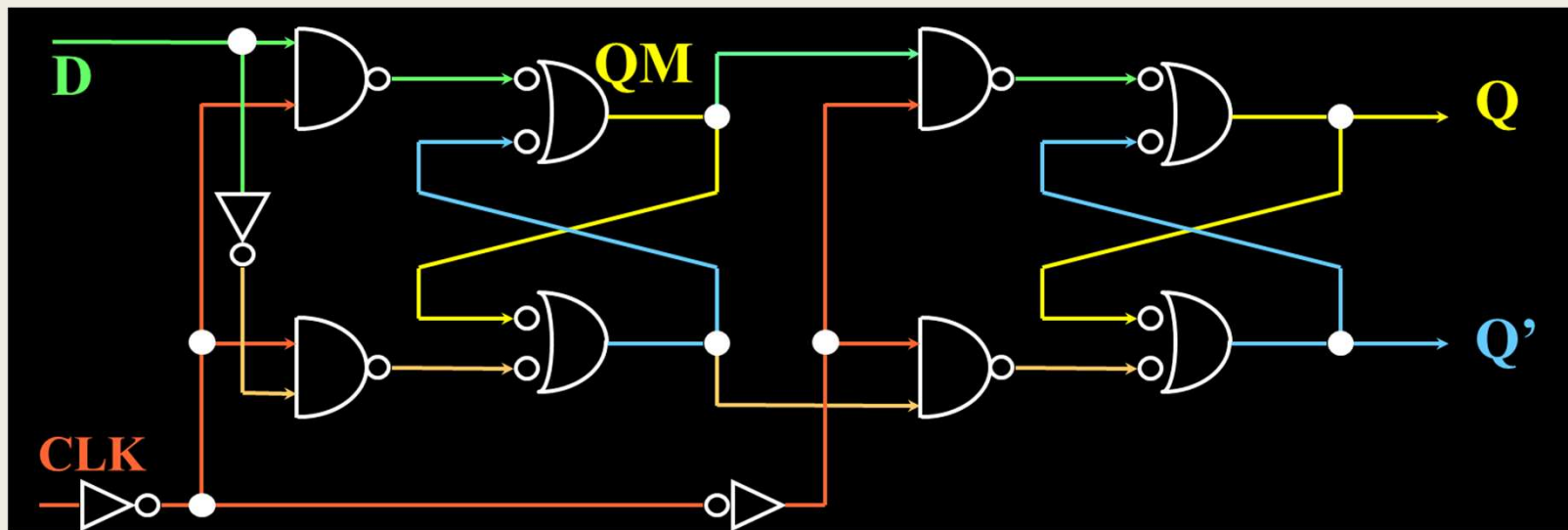
- D Latch: **συνεχής ενημέρωση** της εξόδου Q αντιγράφοντας την τιμή της εισόδου D, όσο **CLK = 1**
- D Flip-Flop: ενημέρωση της εξόδου Q αντιγράφοντας την τιμή της εισόδου D **μόνο στην ανερχόμενη ακμή του CLK**
- Σε όλες τις άλλες περιπτώσεις οι έξοδοι Q διατηρούν τη τιμή τους και τα κυκλώματα την κατάσταση τους
 - Η κατάσταση ταυτίζεται με την τιμή της εξόδου Q
- Η αρχική τιμή της εξόδου Q είναι απροσδιόριστη
 - Στην πράξη τα κυκλώματα πηγαίνουν ανεξέλεγκτα σε μία από τις δύο σταθερές καταστάσεις (0 ή 1)



*Παράδειγμα 3.2

Υλοποίηση D Flip-Flop με πύλες NAND

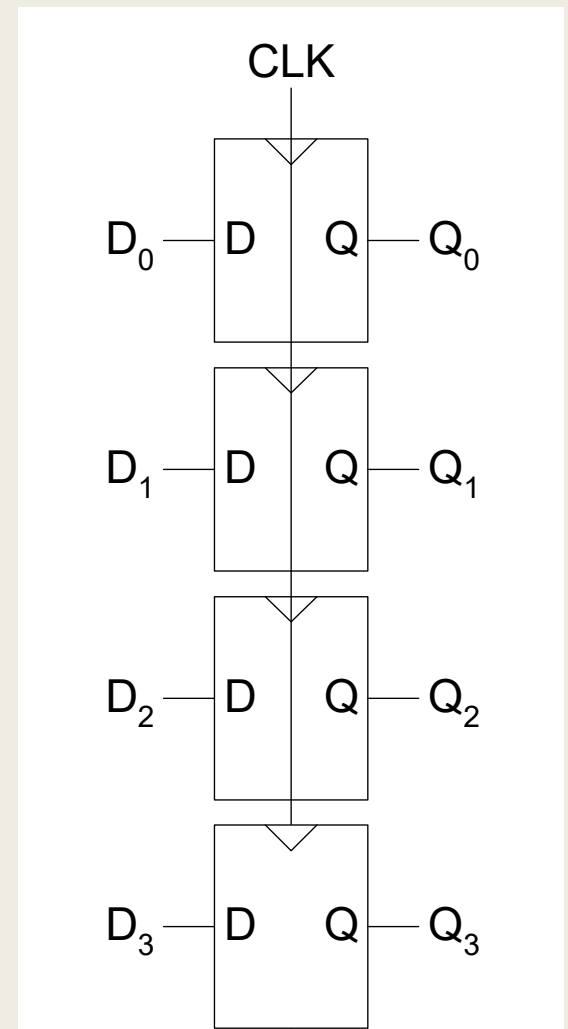
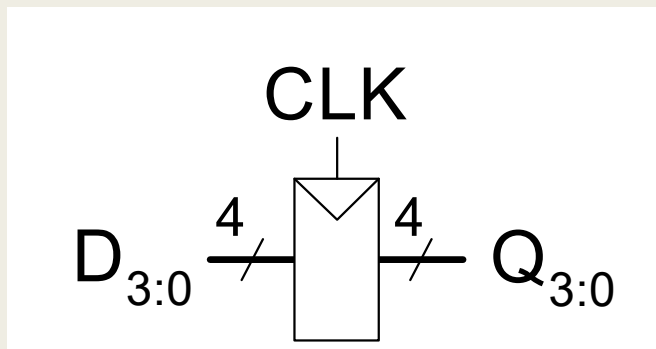
- Με δύο D Latches σε συνδεσμολογία αφέντη-σκλάβου
- Τα D Latches βασίζονται στο S-R Latch (active LOW) με πύλες NAND



S-R Latch (active LOW)
με πύλες NAND – Άσκηση 3.7

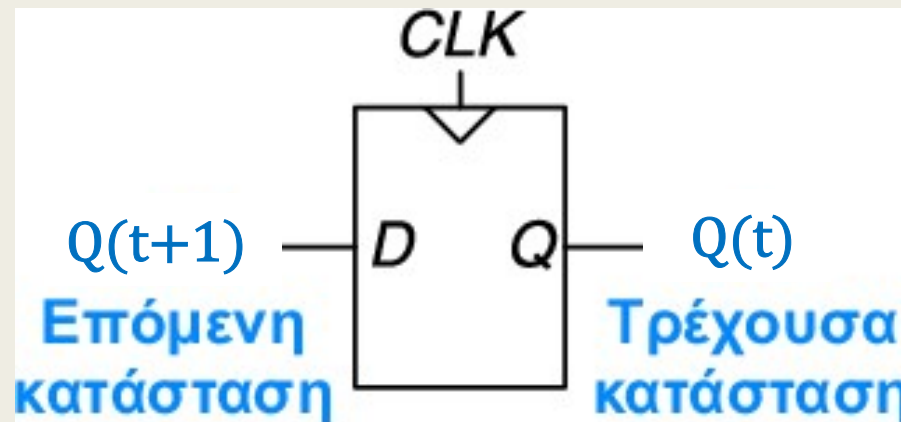
Καταχωρητές

- Ένας **καταχωρητής** (register) μεγέθους N bit είναι μια συστοιχία από N Flip-Flop τα οποία χρησιμοποιούν από κοινού μια είσοδο ρολογιού CLK, έτσι ώστε όλα τα bit του καταχωρητή να ενημερώνονται ταυτόχρονα
- Οι καταχωρητές αποτελούν το θεμελιώδες δομικό στοιχείο στα σύγχρονα ακολουθιακά κυκλώματα (ή μηχανές πεπερασμένων καταστάσεων – FSM)



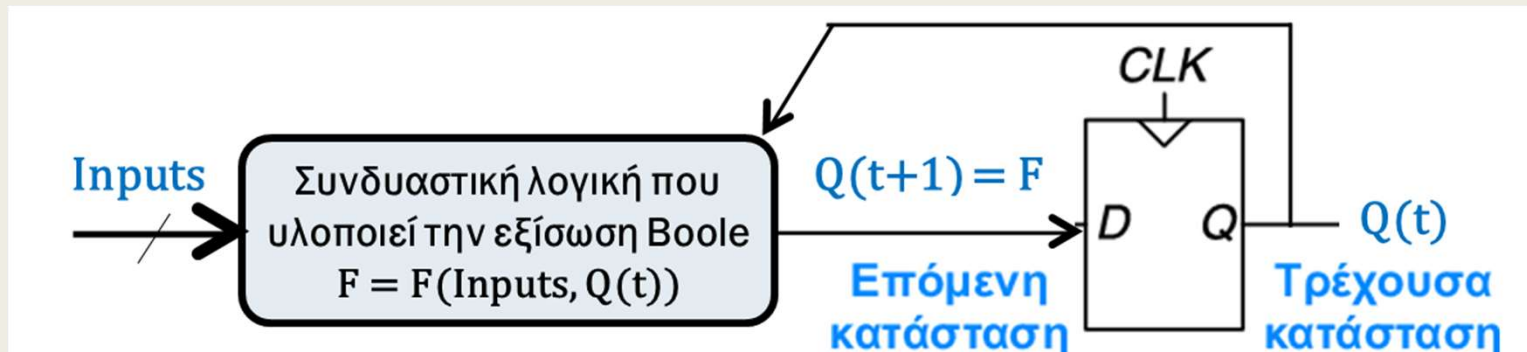
Σχεδίαση άλλων Flip-Flop

- Χρησιμοποιείται σαν βάση το **D Flip-Flop**, το οποίο
 - διατηρεί αποθηκευμένη την **τρέχουσα κατάσταση $Q(t)$** (που ταυτίζεται με την τιμή της εξόδου Q), όσο το CLK δεν βρίσκεται στην ανερχόμενη ακμή του
 - πηγαίνει στην **επόμενη κατάσταση $Q(t+1)$** , που είναι η τιμή που έχει η είσοδος δεδομένων D κατά την επόμενη ανερχόμενη ακμή του CLK
- **$Q(t+1) = D$**



Σχεδίαση άλλων Flip-Flop

- Προσδιορίζουμε τον **πίνακα αλήθειας** του νέου Flip-Flop
 - εξετάζοντας ποιες είναι οι τιμές των εισόδων του κατά την ανερχόμενη ακμή του CLK
 - στις εισόδους του πίνακα συμπεριλαμβάνεται η τρέχουσα κατάσταση $Q(t)$
- Προσδιορίζουμε την **εξίσωση Boole** για την επόμενη κατάσταση $Q(t+1)$
 - ως συνάρτηση των εισόδων και της τρέχουσας κατάστασης $Q(t)$
 - μετά από ελαχιστοποίηση με τον χάρτη Karnaugh)
- Σχεδιάζουμε την **συνδυαστική λογική** με βάση την εξίσωση Boole
 - Η έξοδος F αυτού του συνδυαστικού κυκλώματος συνδέεται στην είσοδο δεδομένων D του D Flip-Flop, άρα $D = F$
- $Q(t+1) = F(\text{Inputs}, Q(t))$



D Flip-Flop με επαναφορά στο 0

- Ένα **D Flip-Flop με δυνατότητα σύγχρονης επαναφοράς στο 0 (resettable Flip-Flop)** διαθέτει άλλη μία είσοδο που ονομάζεται **RESET**
 - Η είσοδος **RESET** είναι **σύγχρονη** (δηλαδή επιδρά στο κύκλωμα στην ανερχόμενη ακμή του CLK)
 - Όταν η είσοδος **RESET** έχει την τιμή **FALSE**, συμπεριφέρεται ως απλό D Flip-Flop
 - Όταν η είσοδος **RESET** έχει την τιμή **TRUE**, **αγνοεί την είσοδο D** και επαναφέρει την κατάσταση και έξοδο Q στην **τιμή 0**
- Στα **D Flip-Flop με δυνατότητα ασύγχρονης επαναφοράς στο 0**, η επαναφορά στο 0 λαμβάνει χώρα αμέσως μόλις η είσοδος **RESET** πάρει την τιμή **TRUE**, ανεξάρτητα από την τιμή του CLK
 - Το ασύγχρονο σήμα επαναφοράς στο 0 ονομάζεται και **CLEAR**
- Αυτού του είδους τα D Flip-Flop είναι χρήσιμα στην περίπτωση που θέλουμε να επιβάλλουμε μια γνωστή κατάσταση στα Flip-Flop ενός συστήματος όταν το ενεργοποιούμε για πρώτη φορά

D Flip-Flop με επαναφορά στο 0

- Σχεδίαση ενός D Flip-Flop με δυνατότητα σύγχρονης επαναφοράς στο 0

– Η είσοδος *RESET* (*R*) είναι ενεργή στο *HIGH* (*active HIGH*)

■ Πίνακας Αλήθειας

RESET	D	Q(t)	Q(t + 1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

LOAD

RESET

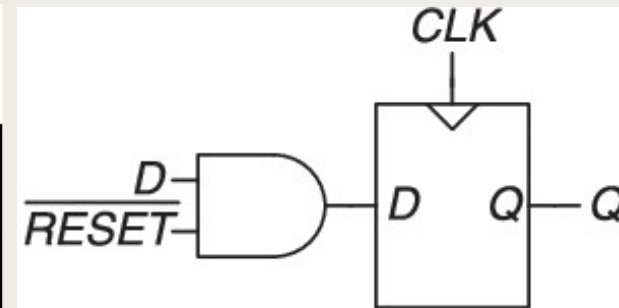
■ K-map και εξίσωση

		R D			
Q(t)		00	01	11	10
	0	0	1	0	0
	1	0	1	0	0

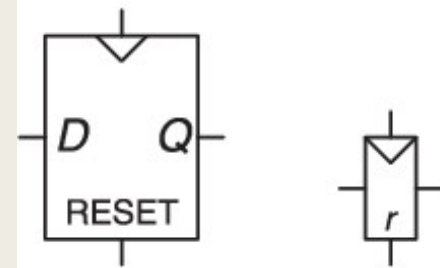
$$Q(t + 1) = \overline{RESET} D$$

Σε αυτή την περίπτωση το $Q(t+1)$ δεν εξαρτάται από το $Q(t)$

■ Σχηματικό διάγραμμα



■ Σύμβολα



D Flip-Flop με τοποθέτηση στο 1

- Ένα **D Flip-Flop με δυνατότητα σύγχρονης τοποθέτησης στο 1 (settable Flip-Flop)** διαθέτει άλλη μία είσοδο που ονομάζεται **SET**
 - Η είσοδος **SET** είναι **σύγχρονη** (δηλαδή επιδρά στο κύκλωμα στην ανερχόμενη ακμή του CLK)
 - Όταν η είσοδος **SET** έχει την τιμή **FALSE**, συμπεριφέρεται ως απλό D Flip-Flop
 - Όταν η είσοδος **SET** έχει την τιμή **TRUE**, **αγνοεί την είσοδο D** και τοποθετεί την κατάσταση και έξοδο Q στην **τιμή 1**
- Στα **D Flip-Flop με δυνατότητα ασύγχρονης τοποθέτησης στο 1**, η τοποθέτηση στο 1 λαμβάνει χώρα αμέσως μόλις η είσοδος **SET** πάρει την τιμή **TRUE**, ανεξάρτητα από την τιμή του CLK
 - Το ασύγχρονο σήμα τοποθέτησης στο 1 ονομάζεται και **PRESET**
- Αυτού του είδους τα D Flip-Flop είναι χρήσιμα στην περίπτωση που θέλουμε να επιβάλλουμε μια γνωστή κατάσταση στα Flip-Flop ενός συστήματος όταν το ενεργοποιούμε για πρώτη φορά

D Flip-Flop με τοποθέτηση στο 1

- Σχεδίαση ενός D Flip-Flop με δυνατότητα σύγχρονης τοποθέτησης στο 1

– Η είσοδος SET (S) είναι ενεργή στο HIGH (active HIGH)

■ Πίνακας Αλήθειας

SET	D	Q(t)	Q(t + 1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

LOAD

SET

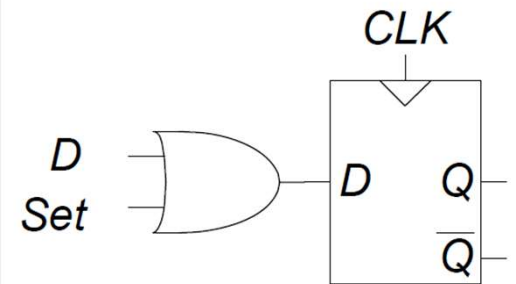
■ K-map και εξίσωση

		SD			
Q(t)		00	01	11	10
	0	0	1	1	1
	1	0	1	1	1

$$Q(t + 1) = S + D$$

Σε αυτή την περίπτωση το Q(t+1) δεν εξαρτάται από το Q(t)

■ Σχηματικό διάγραμμα



■ Σύμβολα

1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

SET

S

Υλοποίηση D Flip-Flop με CLEAR & PRESET (ενεργά στο LOW – active LOW)

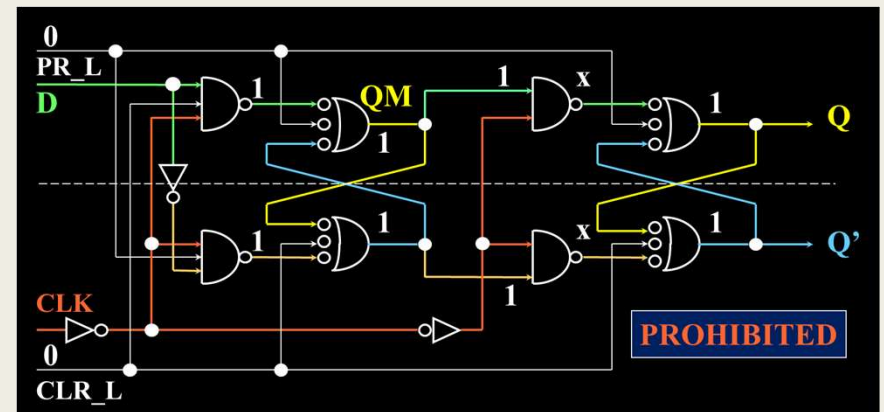
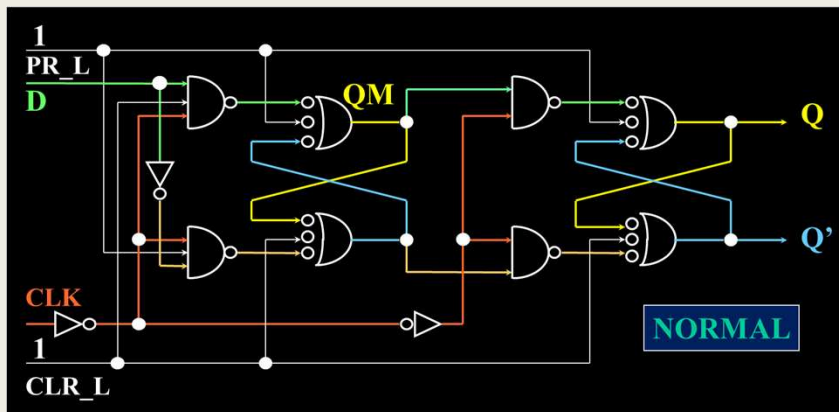
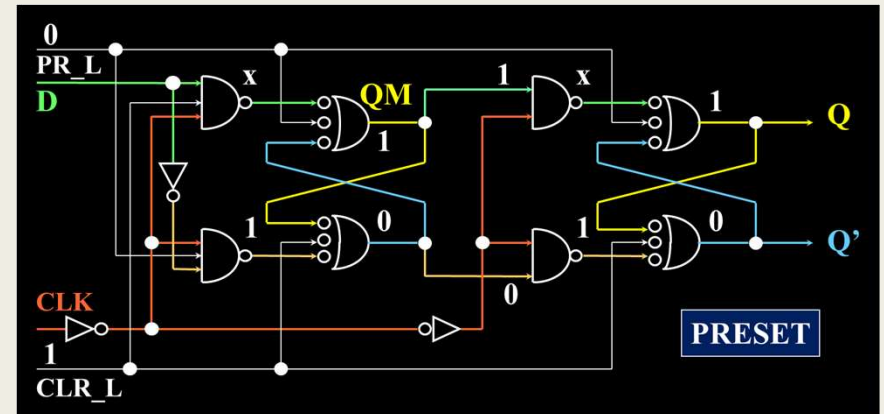
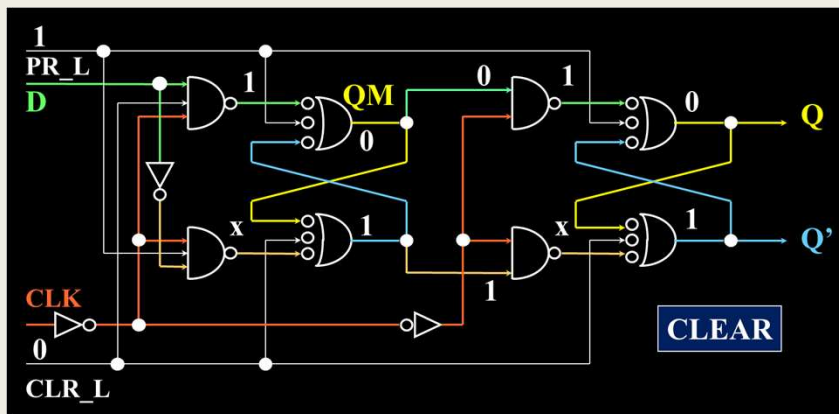
- Ασύγχρονη επαναφορά στο 0
- Ασύγχρονη τοποθέτηση στο 1
- Κανονική λειτουργία
- Μη έγκυρη λειτουργία

(CLR_L=0 και PR_L=1)

(CLR_L=1 και PR_L=0)

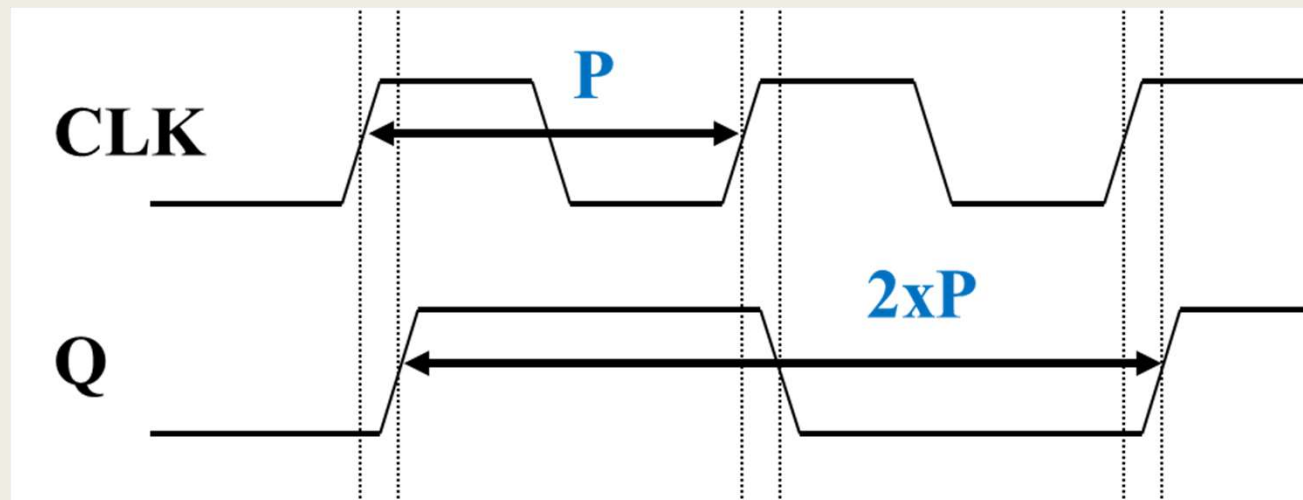
(CLR_L=1 και PR_L=1)

(CLR_L=0 και PR_L=0)



T (Toggle) Flip-Flop

- Το **T (toggle, εναλλαγή) Flip-Flop** διαθέτει μία είσοδο *CLK* και μία έξοδο *Q*
- Σε κάθε ανερχόμενη ακμή του *CLK*, η έξοδος *Q* εναλλάσσει την τιμή της με το συμπλήρωμα της προηγούμενης τιμής της
- Το T Flip-Flop χρησιμοποιείται κυρίως ως **διαιρέτης συχνότητας διά 2**
 - *N* T Flip-Flops στη σειρά απαρτίζουν έναν διαιρέτη συχνότητας δια 2^N



T Flip-Flop

- Σχεδίαση ενός T Flip-Flop

- Πίνακας Αλήθειας

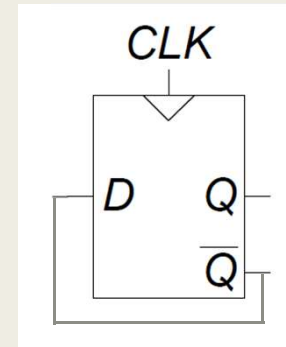
$Q(t)$	$Q(t + 1)$
0	1
1	0

TOGGLE

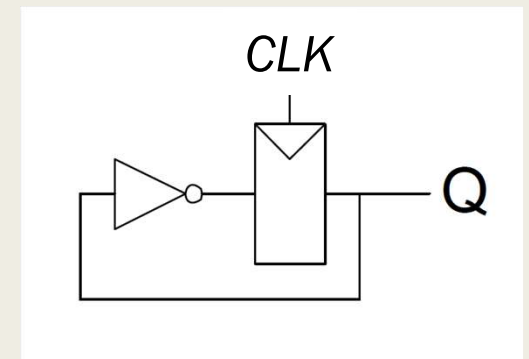
- Εξίσωση Boole

$$Q(t + 1) = \overline{Q(t)}$$

- Σχηματικό διάγραμμα



- Σύμβολο



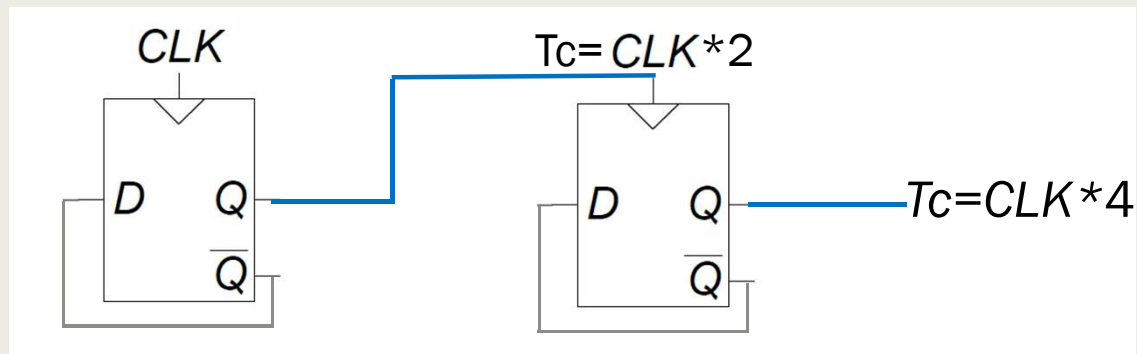
Διαίρετης συχνότητας δια 4

- Σχηματικό διάγραμμα με τη χρήση δύο T Flip-Flops

$$F = 1/CLK$$

$$F_1 = F/2$$

$$F_2 = F/4$$



D Flip-Flop με έγκριση (enabled)

- Ένα **D Flip-Flop με έγκριση (enabled flip-flop)** διαθέτει μία ακόμα είσοδο που ονομάζεται σήμα έγκρισης **EN** ή **ENABLE**
- Η επιπλέον είσοδος **EN**, χρησιμοποιείται για να καθορίζει αν τα δεδομένα θα φορτωθούν ή όχι κατά την επόμενη ακμή του CLK
 - Όταν η **είσοδος EN** έχει την τιμή **TRUE**, το D Flip-Flop με έγκριση συμπεριφέρεται ως απλό D Flip-Flop
 - Όταν η **είσοδος EN** έχει την τιμή **FALSE**, το D Flip-Flop με έγκριση **αγνοεί το CLK** και διατηρεί την κατάστασή του
- Αυτού του είδους τα κυκλώματα είναι χρήσιμα όταν θέλουμε να φορτώνουμε μια νέα τιμή σε ένα Flip-Flop μόνο σε κάποιες περιπτώσεις, και όχι σε κάθε ακμή του ρολογιού
- Το σήμα έγκρισης ονομάζεται και **write enable** ή **clock enable**

D Flip-Flop με έγκριση (enabled)

■ Σχεδίαση ενός D Flip-Flop με έγκριση

- Η είσοδος *ENABLE* (EN) είναι ενεργή στο *HIGH* (active HIGH)

■ Πίνακας Αλήθειας

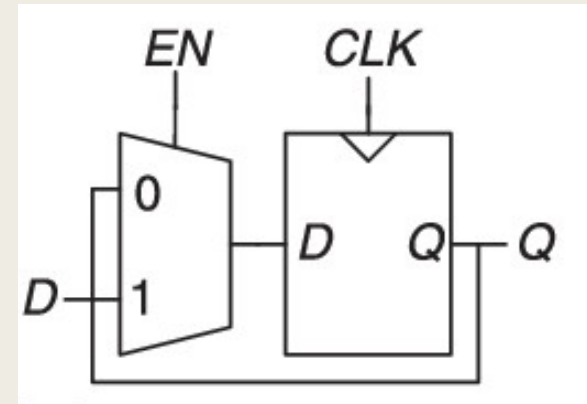
EN	D	Q(t)	Q(t + 1)	
0	0	0	0	HOLD
0	0	1	1	
0	1	0	0	
0	1	1	1	
1	0	0	0	LOAD
1	0	1	0	
1	1	0	1	
1	1	1	1	

■ K-map και εξίσωση

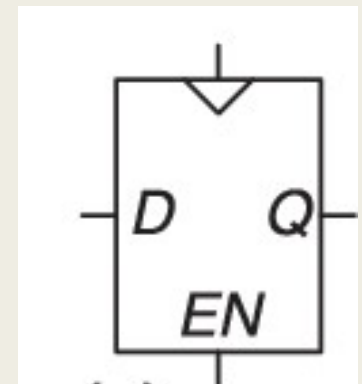
		EN D			
		00	01	11	10
Q(t)	0	0	0	1	0
	1	1	1	1	0

$$Q(t + 1) = \overline{EN} Q(t) + EN D$$

■ Σχηματικό διάγραμμα

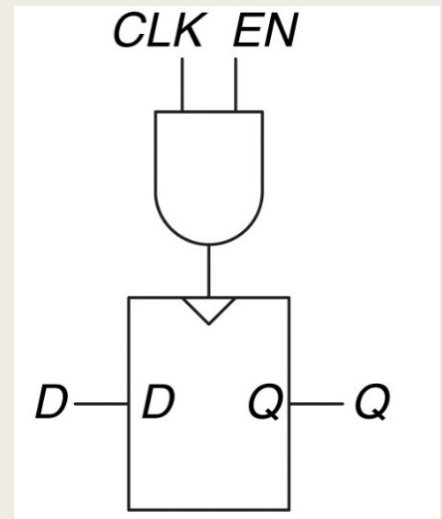


■ Σύμβολο



D Flip-Flop με έγκριση (enabled)

- Μία εναλλακτική σχεδίαση ενός D Flip-Flop με έγκριση
 - Η είσοδος *ENABLE* (*EN*) είναι ενεργή στο *HIGH* (*active HIGH*)
- Το σήμα του ρολογιού *CLK* περνάει μέσα από μία πύλη *AND* πριν συνδεθεί στην είσοδο *CLK* του D Flip-Flop
 - Αν $EN = 1$, τότε η είσοδος *CLK* του D Flip-Flop εναλλάσσεται φυσιολογικά
 - Αν $EN = 0$, τότε η είσοδος *CLK* του D Flip-Flop παγώνει στην τιμή 0, και το D Flip-Flop διατηρεί την κατάσταση του
- Η συγκεκριμένη σχεδίαση εγκυμονεί κινδύνους
 - Η τιμή της εισόδου *EN* δεν πρέπει να μεταβάλλεται ενόσω ισχύει $CLK = 1$, ώστε να μην εμφανισθεί μεταβατικός παλμός (*glitch*) στην είσοδο *CLK* που θα οδηγήσει σε ανεξέλεγκτη αλλαγή κατάστασης
 - Η προσθήκη πύλης στο *CLK* το καθυστερεί και μπορεί να προκαλέσει και σφάλματα χρονισμού



Προσοχή! Δεν συνιστάται η χρήση του CLK με πύλες

T Flip-Flop με έγκριση (enabled)

- Ένα **T Flip-Flop με έγκριση (enabled flip-flop)** διαθέτει μία ακόμα είσοδο που ονομάζεται **EN** ή **ENABLE**
- Η επιπλέον είσοδος **EN**, χρησιμοποιείται για να καθορίζει εάν θα εναλλαχθεί ή όχι η κατάσταση του T Flip-Flop κατά την επόμενη ακμή του CLK
 - Όταν η **είσοδος EN** έχει την τιμή **TRUE**, το T Flip-Flop με έγκριση εναλλάσσει την κατάσταση του (συμπεριφέρεται σαν T Flip-Flop)
 - Όταν η **είσοδος EN** έχει την τιμή **FALSE**, το T Flip-Flop με έγκριση διατηρεί την κατάσταση του
- Αυτού του είδους τα κυκλώματα είναι ιδιαίτερα χρήσιμα στην υλοποίηση **σύγχρονων μετρητών**

T Flip-Flop με έγκριση (enabled)

■ Σχεδίαση ενός T Flip-Flop με έγκριση

- Η είσοδος *ENABLE* (*EN*) είναι ενεργή στο *HIGH* (active *HIGH*)

■ Πίνακας Αλήθειας

EN	Q(t)	Q(t + 1)
0	0	0
0	1	1
1	0	1
1	1	0

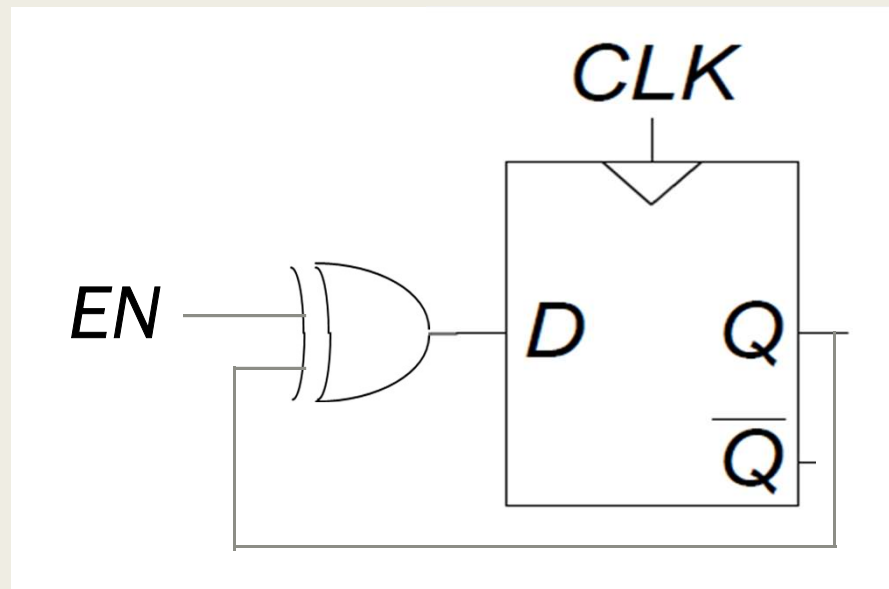
HOLD

TOGGLE

■ Εξίσωση Boole

$$Q(t + 1) = \overline{EN} Q(t) + EN \overline{Q(t)} = EN \oplus Q(t)$$

■ Σχηματικό διάγραμμα



J-K Flip-Flop

- Το **J-K Flip-Flop** δέχεται ένα σήμα CLK και δύο σύγχρονες εισόδους (το J και το K).
- Κατά την ανερχόμενη ακμή του CLK ενημερώνει την έξοδο Q , σύμφωνα με τις τιμές που έχουν οι είσοδοι J και K , ως εξής:
 - Όταν οι είσοδοι J και K έχουν και οι δύο την τιμή 0, η έξοδος Q διατηρεί την προηγούμενη τιμή της (**hold**)
 - Όταν η είσοδος J έχει την τιμή 0 και η είσοδος K έχει την τιμή 1, η έξοδος Q παίρνει την τιμή 0 (**reset**)
 - Όταν η είσοδος J έχει την τιμή 1 και η είσοδος K έχει την τιμή 0, η έξοδος Q παίρνει την τιμή 1 (**set**)
 - Όταν οι είσοδοι J και K έχουν και οι δύο την τιμή 1, η έξοδος Q εναλλάσσει την τιμή της με το συμπλήρωμα της προηγούμενης τιμής της (**toggle**)
- Αυτό το κύκλωμα ήταν ιδιαίτερα δημοφιλές την εποχή που οι υλοποιήσεις βασίζονταν σε πλακέτες με SSI και MSI ολοκληρωμένα κυκλώματα

J-K Flip-Flop

■ Σχεδίαση ενός J-K Flip-Flop

■ Πίνακας Αλήθειας

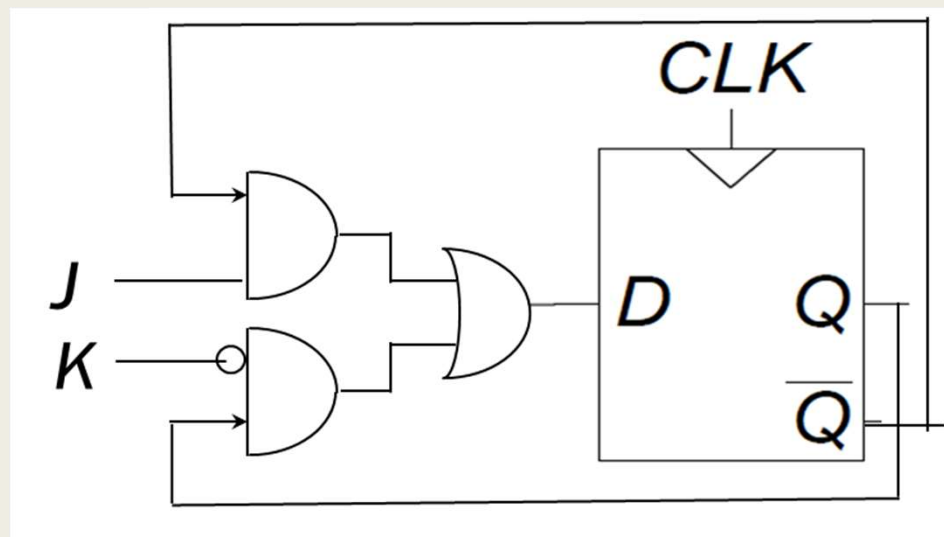
J	K	Q(t)	Q(t + 1)	
0	0	0	0	HOLD
0	0	1	1	
0	1	0	0	RESET
0	1	1	0	
1	0	0	1	SET
1	0	1	1	
1	1	0	1	TOGGLE
1	1	1	0	

■ K-map και εξίσωση

JK		00	01	11	10
Q(t)	0	0	0	1	1
	1	1	0	0	1

$$Q(t + 1) = \overline{Q(t)} J + Q(t) \overline{K}$$

■ Σχηματικό διάγραμμα



Επιλεγμένη άσκηση: το A-B Flip-Flop

- Το **A-B Flip-Flop** δέχεται ένα σήμα CLK και δύο σύγχρονες εισόδους (το A και το B).
- Κατά την ανερχόμενη ακμή του CLK ενημερώνει την έξοδο Q, σύμφωνα με τις τιμές που έχουν οι είσοδοι A και B, ως εξής:
 - Όταν οι είσοδοι A και B έχουν και οι δύο την τιμή 0, η έξοδος Q διατηρεί την προηγούμενη τιμή της (**hold**)
 - Όταν η είσοδος A έχει την τιμή 0 και η είσοδος B έχει την τιμή 1, η έξοδος Q εναλλάσσει την τιμή της με το συμπλήρωμα της προηγούμενης τιμής της (**toggle**)
 - Όταν η είσοδος A έχει την τιμή 1 και η είσοδος B έχει την τιμή 0, η έξοδος Q παίρνει την τιμή 0 (**reset**)
 - Όταν οι είσοδοι A και B έχουν και οι δύο την τιμή 1, η έξοδος Q αντιγράφει την τιμή της εισόδου D (**load**)
- Να σχεδιάσετε το νέο Flip-Flop χρησιμοποιώντας στη συνδυαστική λογική έναν πολυπλέκτη 2 σε 1, μία πύλη XOR και όποια άλλη πύλη απαιτείται

Επιλεγμένη άσκηση: το A-B Flip-Flop

- Από τον Πίνακα Λειτουργιών στον Πίνακα Αλήθειας

AB	Λειτουργία
00	HOLD
01	TOGGLE
10	RESET
11	LOAD

Επιλεγμένη άσκηση: το A-B Flip-Flop

- Από τον Πίνακα Λειτουργιών στον Πίνακα Αλήθειας

AB	Λειτουργία
00	HOLD
01	TOGGLE
10	RESET
11	LOAD

AB	Λειτουργία	$Q(t+1)$
00	HOLD	$Q(t)$
01	TOGGLE	$\overline{Q(t)}$
10	RESET	0
11	LOAD	D

Επιλεγμένη άσκηση: το A-B Flip-Flop

- Από τον Πίνακα Λειτουργιών στον Πίνακα Αλήθειας

AB	Λειτουργία
00	HOLD
01	TOGGLE
10	RESET
11	LOAD

A	B	D	Q(t)	Q(t+1)
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0

A	B	D	Q(t)	Q(t+1)
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Επιλεγμένη άσκηση: το A-B Flip-Flop

- Από τον Πίνακα Αλήθειας στον Χάρτη Karnaugh και την εξίσωση Boole για την επόμενη κατάσταση $Q(t+1)$

A	B	D	Q(t)	Q(t+1)	A	B	D	Q(t)	Q(t+1)
0	0	0	0	0	1	0	0	0	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	1	1	1	0	1	1	0
0	1	0	0	1	1	1	0	0	0
0	1	0	1	0	1	1	0	1	0
0	1	1	0	1	1	1	1	0	1
0	1	1	1	0	1	1	1	1	1

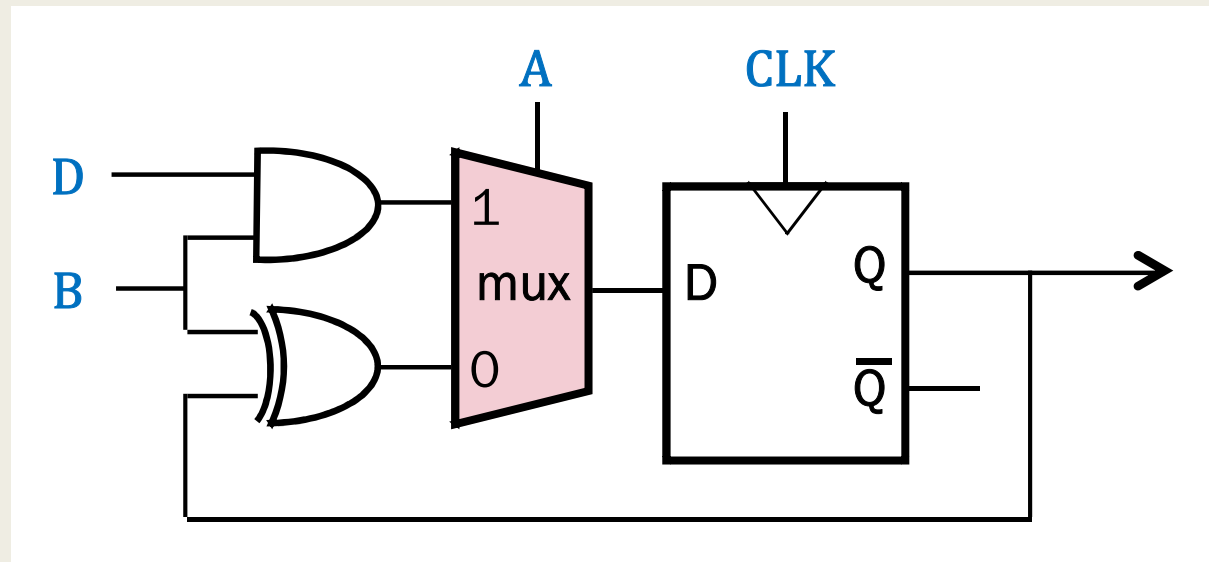
$AB \backslash D Q(t)$	00	01	11	10
00	0	1	0	0
01	1	0	0	0
11	1	0	1	0
10	0	1	1	0

$$\begin{aligned}
 Q(t+1) &= \bar{A}\bar{B}Q(t) + \bar{A}B\overline{Q(t)} + ABD = \bar{A}(\bar{B}Q(t) + B\overline{Q(t)}) + ABD \\
 &= \bar{A}(B \oplus Q(t)) + A(BD)
 \end{aligned}$$

Επιλεγμένη άσκηση: το A-B Flip-Flop

- Από την **εξίσωση Boole** για την επόμενη κατάσταση $Q(t+1)$ στο **σηματικό διάγραμμα** του νέου Flip-Flop
 - Να σχεδιάσετε το νέο Flip-Flop χρησιμοποιώντας στη συνδυαστική λογική έναν πολυπλέκτη 2 σε 1, μία πύλη XOR και όποια άλλη πύλη απαιτείται

$$\begin{aligned} Q(t+1) &= \overline{A}\overline{B}Q(t) + \overline{A}B\overline{Q(t)} + ABD = \overline{A}(\overline{B}Q(t) + B\overline{Q(t)}) + ABD \\ &= \overline{A}(B \oplus Q(t)) + A(BD) \end{aligned}$$



Επιλεγμένη άσκηση 2: το A-B Flip-Flop

- Το **A-B Flip-Flop** δέχεται ένα σήμα CLK και δύο σύγχρονες εισόδους (το A και το B).
- Κατά την ανερχόμενη ακμή του CLK ενημερώνει την έξοδο Q, σύμφωνα με τις τιμές που έχουν οι είσοδοι A και B, ως εξής:
 - Όταν οι είσοδοι A και B έχουν και οι δύο την τιμή 0, η έξοδος Q διατηρεί την προηγούμενη τιμή της (**hold**)
 - Όταν η είσοδος A έχει την τιμή 0 και η είσοδος B έχει την τιμή 1, η έξοδος Q εναλλάσσει την τιμή της με το συμπλήρωμα της προηγούμενης τιμής της (**toggle**)
 - Όταν η είσοδος A έχει την τιμή 1 και η είσοδος B έχει την τιμή 0, η έξοδος Q αντιγράφει την τιμή της εισόδου D (**load**)
 - Όταν οι είσοδοι A και B έχουν και οι δύο την τιμή 1, η έξοδος Q παίρνει την τιμή 1 (**set**)
- Να σχεδιάσετε το νέο Flip-Flop χρησιμοποιώντας στη συνδυαστική λογική έναν πολυπλέκτη 2 σε 1, μία πύλη XOR και όποια άλλη πύλη απαιτείται

Επιλεγμένη άσκηση 2: το A-B Flip-Flop

- Από τον Πίνακα Λειτουργιών στον Πίνακα Αλήθειας

AB	Λειτουργία
00	HOLD
01	TOGGLE
10	LOAD
11	SET

AB	Λειτουργία	$Q(t+1)$
00	HOLD	$Q(t)$
01	TOGGLE	$\overline{Q(t)}$
10	LOAD	D
11	SET	1

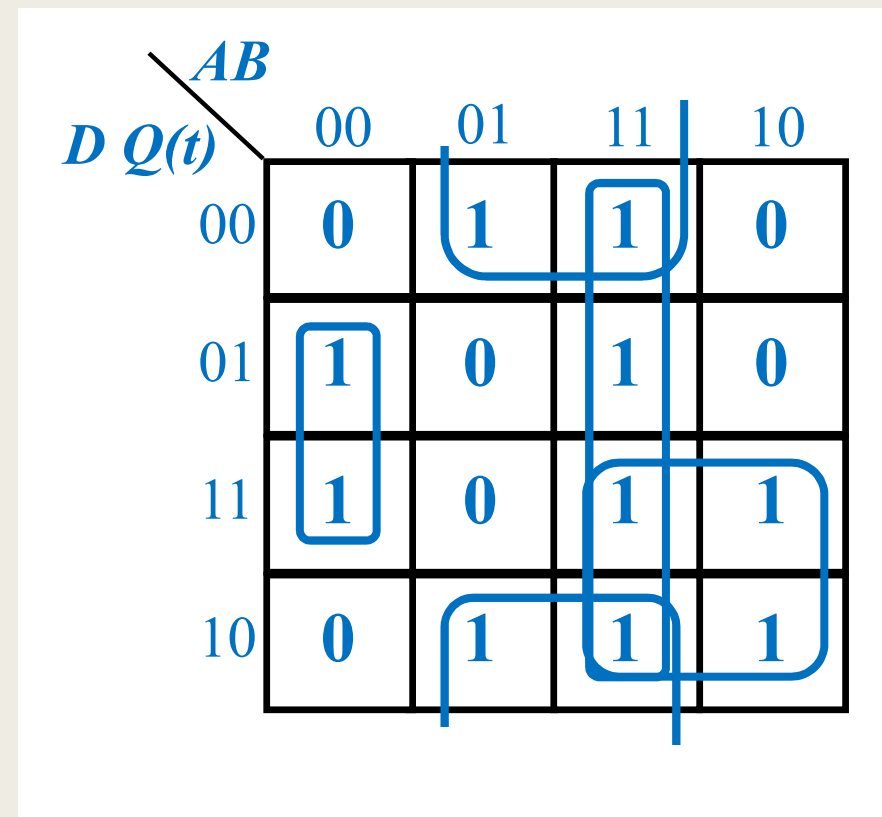
A	B	D	$Q(t)$	$Q(t+1)$
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0

A	B	D	$Q(t)$	$Q(t+1)$
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Επιλεγμένη άσκηση 2: το A-B Flip-Flop

- Από τον Πίνακα Αλήθειας στον Χάρτη Karnaugh και την εξίσωση Boole για την επόμενη κατάσταση $Q(t+1)$

A	B	D	Q(t)	Q(t+1)
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
A	B	D	Q(t)	Q(t+1)
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

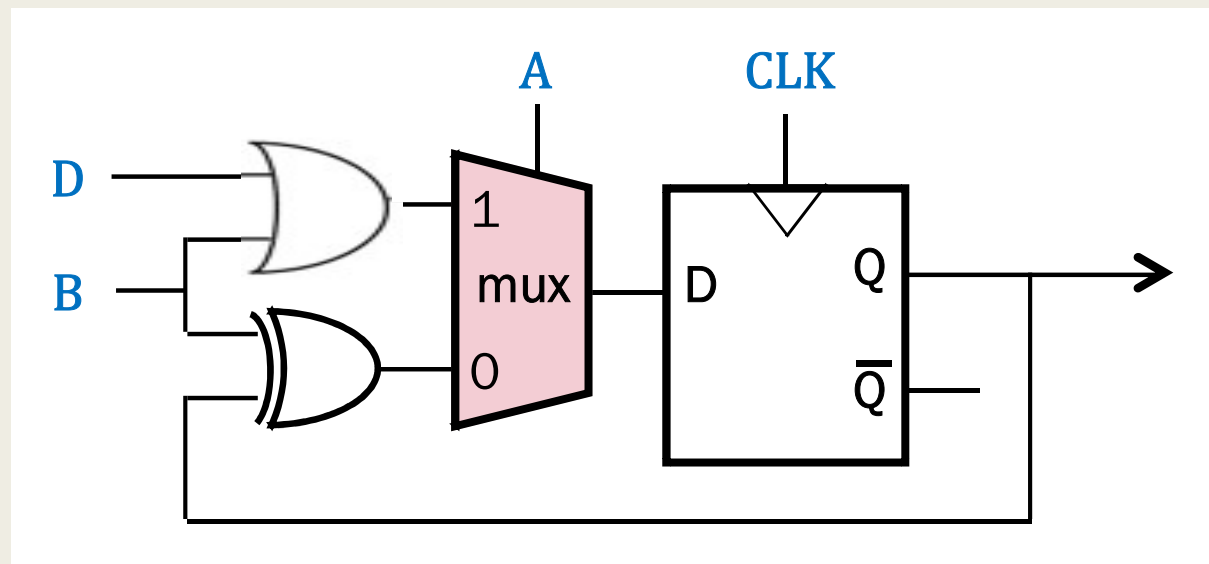


$$\begin{aligned}
 Q(t+1) &= \overline{A}\overline{B}Q(t) + B\overline{Q(t)} + AB + AD = \overline{A}\overline{B}Q(t) + (\overline{A} + A)B\overline{Q(t)} + AB + AD = \\
 &\overline{A}\overline{B}Q(t) + \overline{A}B\overline{Q(t)} + AB\overline{Q(t)} + AB + AD = \overline{A}(\overline{B}Q(t) + B\overline{Q(t)}) + A(B\overline{Q(t)} + B + D) = \\
 &\overline{A}(B \oplus Q(t)) + A(B + D)
 \end{aligned}$$

Επιλεγμένη άσκηση 2: το A-B Flip-Flop

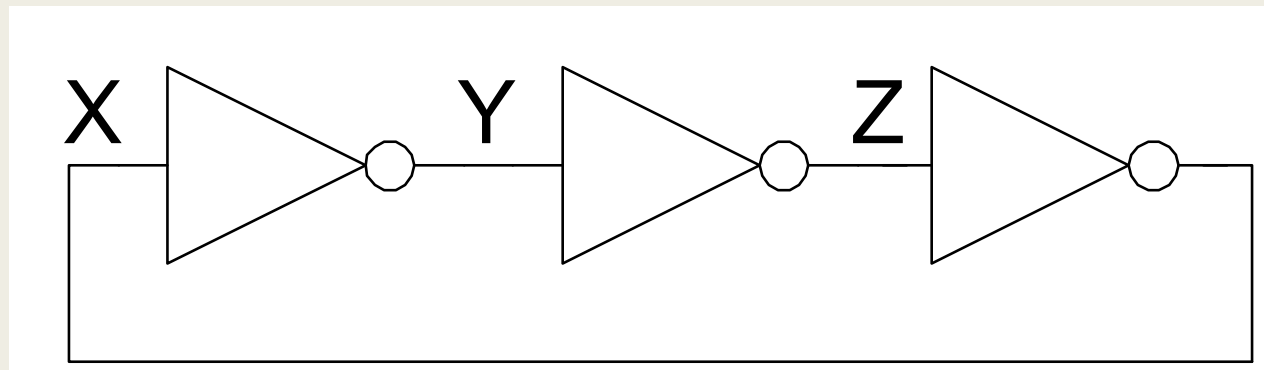
- Από την **εξίσωση Boole** για την επόμενη κατάσταση $Q(t+1)$ στο **σηματικό διάγραμμα** του νέου Flip-Flop
 - Να σχεδιάσετε το νέο Flip-Flop χρησιμοποιώντας στη συνδυαστική λογική έναν πολυπλέκτη 2 σε 1, μία πύλη XOR και όποια άλλη πύλη απαιτείται

$$Q(t+1) = \bar{A}(B \oplus Q(t)) + A(B + D)$$



Ασταθή ακολουθιακά κυκλώματα*

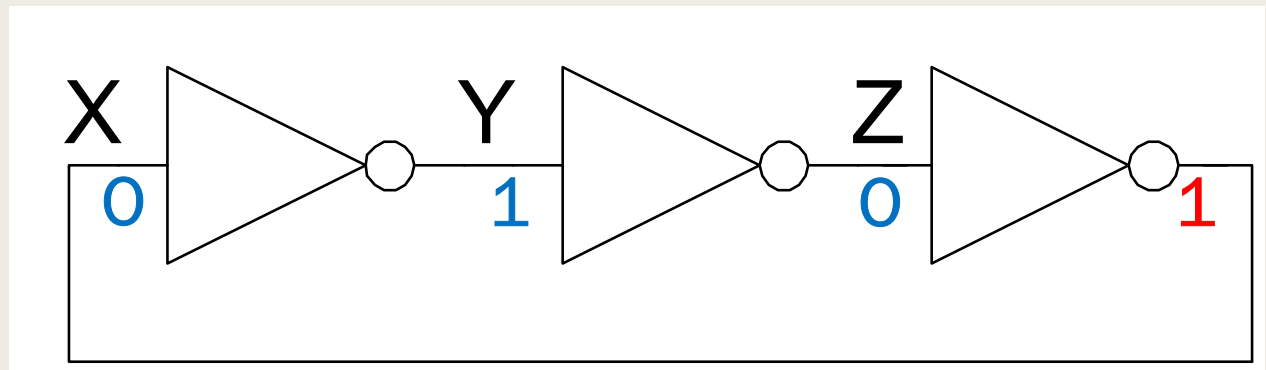
- Τα **ασταθή κυκλώματα** δεν σταθεροποιούνται σε μία κατάσταση, αλλά ταλαντώνονται
- Παράδειγμα: ο **ταλαντωτής δακτυλίου** (ring oscillator)
 - Κατασκευάζεται με **3 διασυνδεδεμένους αντιστροφείς** σε μία **κυκλική διαδρομή**
 - Αρκεί το πλήθος των αντιστροφέων να είναι **περιττό**
 - Δεν υπάρχει είσοδος που να ελέγχει την κατάσταση του κυκλώματος
 - Θεωρούμε έναν κόμβο ως έξοδο, έστω η έξοδος X



* Παράδειγμα 3.3

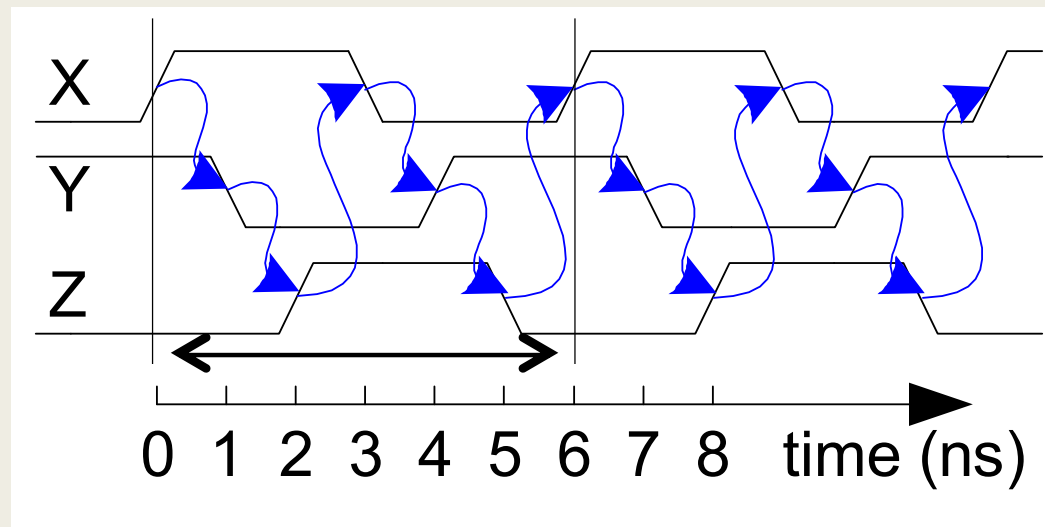
Ασταθή ακολουθιακά κυκλώματα

- **Λειτουργία του ταλαντωτή δακτυλίου** (ring oscillator)
 - Θεωρούμε ότι ο κόμβος X πηγαίνει ανεξέλεγκτα στην τιμή 0 , όταν ενεργοποιούμε το κύκλωμα για πρώτη φορά
 - Ο κόμβος Y πηγαίνει στην τιμή 1 και ο κόμβος Z στην τιμή 0
 - Μετά την ολοκλήρωση της κυκλικής διαδρομής η νέα τιμή του κόμβου X είναι 1
 - που είναι συμπληρωματική της προηγούμενης τιμής ($X=0$)
 - Άρα το κύκλωμα αδυνατεί να σταθεροποιηθεί σε μία κατάσταση, αλλά **ταλαντώνεται**



Ασταθή ακολουθιακά κυκλώματα

- **Περίοδος** του ταλαντωτή δακτυλίου (ring oscillator)
 - Η περίοδος της ταλάντωσης εξαρτάται από την καθυστέρηση διάδοσης του αντιστροφέα
 - η συγκεκριμένη καθυστέρηση εξαρτάται με τη σειρά της από τον τρόπο κατασκευής του αντιστροφέα, την τάση τροφοδοσίας και τη θερμοκρασία
 - είναι δύσκολο να προβλεφθεί με ακρίβεια η περίοδος του ταλαντωτή δακτυλίου (δεν είναι ταλαντωτής ακριβείας)
 - Εάν η καθυστέρηση διάδοσης ενός αντιστροφέα είναι 1 ns , η περίοδος είναι 6 ns

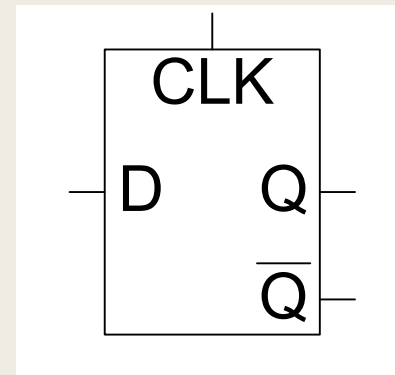


D Latch

■ Πίνακας Αλήθειας

CLK	D	D'	S	R	Q	$\bar{Q}$
0	X	X'	0	0	Q_{prev}	$\overline{Q_{prev}}$
1	0	1	0	1	0	1
1	1	0	1	0	1	0

■ Σύμβολο

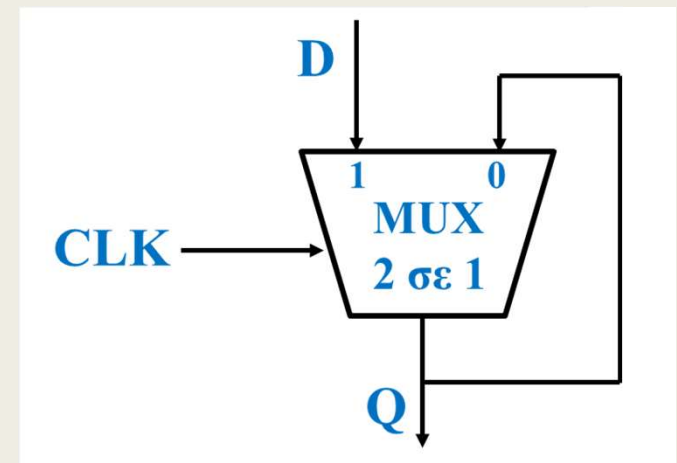


■ Υλοποίηση με πολυπλέκτη 2 σε 1

CLK	D	Q_{prev}	Q
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$CLK \ D$		00	01	11	10
Q_{prev}	0	0	0	1	0
	1	1	1	1	0

$$Q = CLK D + \overline{CLK} Q_{prev}$$

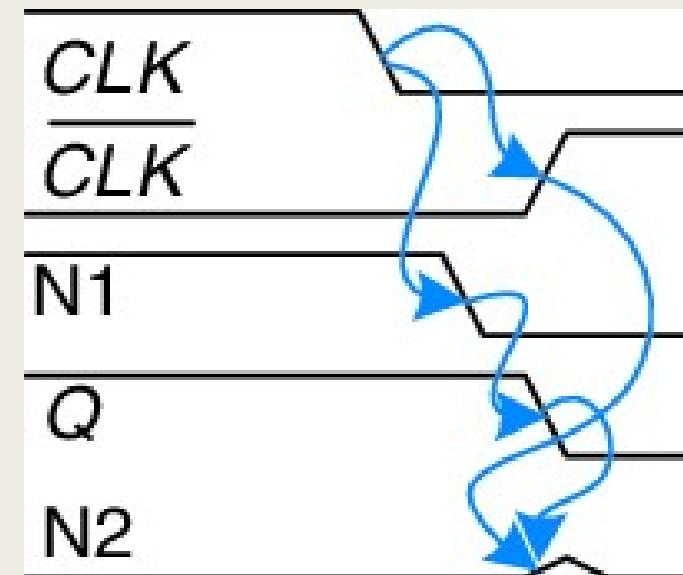
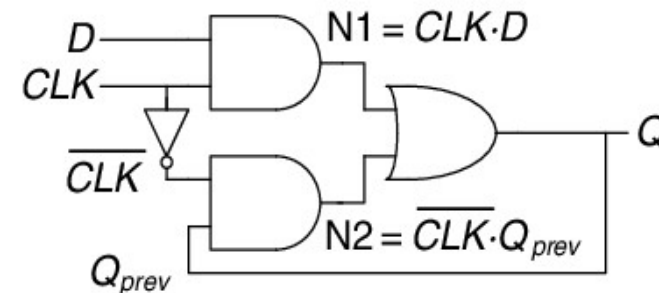


Πιθανόν να εμφανίσει μια συνθήκη συναγωνισμού!

Συνθήκες συναγωνισμού*

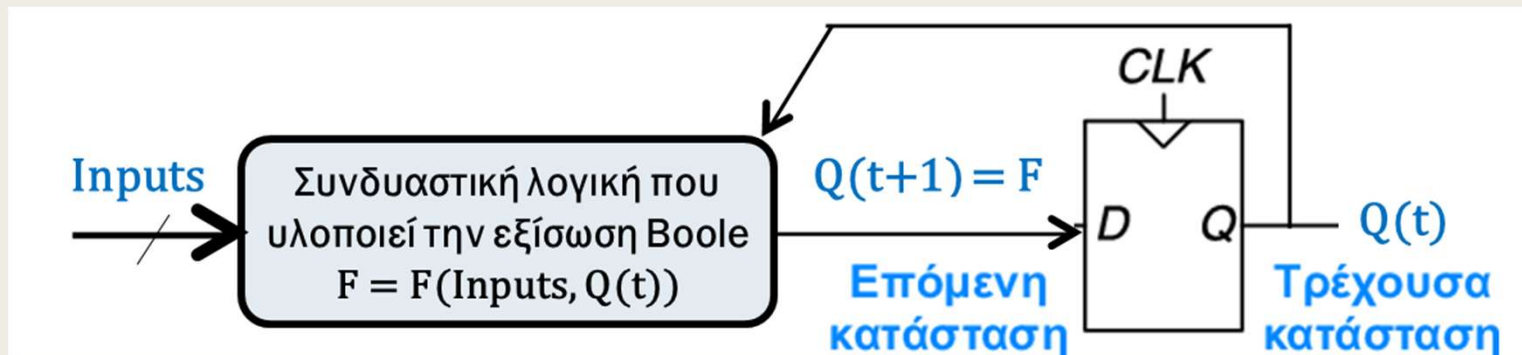
- Το D-Latch στην υλοποίηση με πολυπλέκτη 2 σε 1 πιθανόν να εμφανίσει μια **συνθήκη συναγωνισμού** (race condition)
- Έστω ότι το D-Latch αρχικά είναι σταθεροποιημένο στην κατάσταση **1** ($CLK = D = N1 = Q = 1$)
- Στην κατερχόμενη ακμή του CLK (από το 1 στο 0) το D-Latch πρέπει φυσιολογικά να αποθηκεύσει την προηγούμενη κατάστασή του $Q_{prev} = 1$
- Αν όμως η καθυστέρηση διάδοσης μέσω της πύλης NOT (από CLK σε $\overline{CLK}$) είναι μεγαλύτερη της καθυστέρησης διάδοσης μέσω των πυλών AND και OR (από CLK σε Q), τότε το Q θα γίνει 0 πριν το $\overline{CLK}$ γίνει 1 και θα αποθηκεύσει λανθασμένα ως προηγούμενη κατάσταση το 0, αντί το 1
- Σε μια τέτοια περίπτωση, η άνοδος του κόμβου N2 δεν θα συμβεί ποτέ, και η έξοδος Q θα παραμείνει κολλημένη στο 0

$$Q = CLK \cdot D + \overline{CLK} \cdot Q_{prev}$$



Σύγχρονα ακολουθιακά κυκλώματα

- Η ρίζα του κακού είναι οι **κυκλικές διαδρομές** που δημιουργούν:
 - ανεπιθύμητες **συνθήκες συναγωνισμού** ή **ασταθή συμπεριφορά**
- Οι κυκλικές διαδρομές είναι χαρακτηριστικό των **ασύγχρονων ακολουθιακών κυκλωμάτων**, όπου η έξοδος συνδέεται κατευθείαν στην είσοδο με μία **διαδρομή ανάδρασης**
- Αυτού του είδους τα προβλήματα αντιμετωπίζονται αποτελεσματικά με την **εισαγωγή καταχωρητών** σε ένα ή περισσότερα σημεία της διαδρομής ανάδρασης
 - το κύκλωμα μετασχηματίζεται σε μια συνδεσμολογία συνδυαστικής λογικής και καταχωρητών που ονομάζεται **σύγχρονο ακολουθιακό κύκλωμα**



Προδιαγραφές σύγχρονων ακολουθιακών κυκλωμάτων

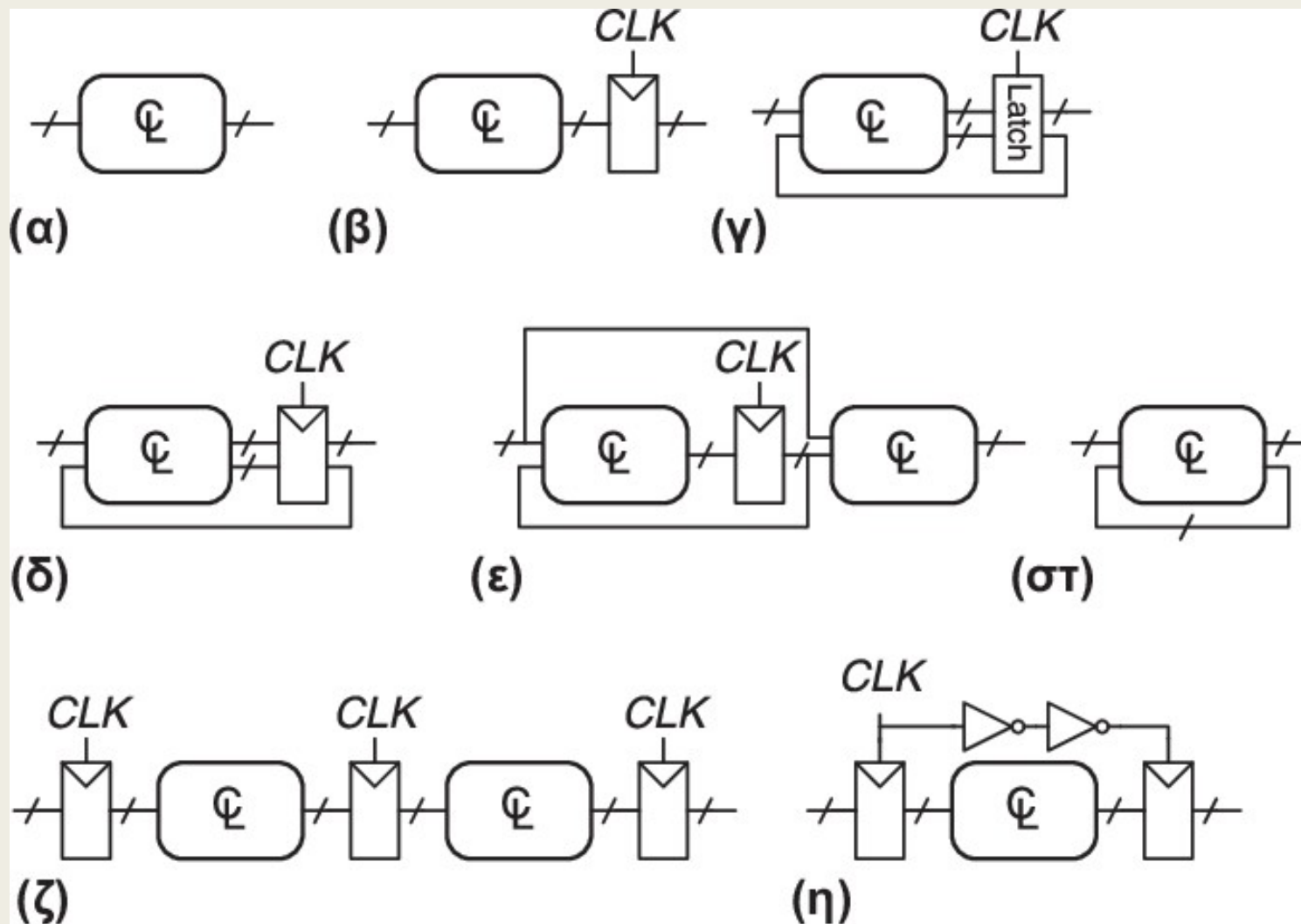
- Η κατάσταση του σύγχρονου ακολουθιακού κυκλώματος μεταβάλλεται στην **ανερχόμενη ακμή του CLK**, αφού πρώτα η συνδυαστική λογική αποκτήσει σταθερές τιμές στις εξόδους της
- Η **λειτουργική προδιαγραφή** περιγράφει λεπτομερώς την **επόμενη κατάσταση $Q(t+1)$** και την **τιμή κάθε εξόδου** για κάθε πιθανό συνδυασμό τιμών της **τρέχουσας κατάστασης $Q(t)$** και των **εισόδων**
- Η **προδιαγραφή χρονισμού για κάθε D Flip-Flop** περιλαμβάνει:
 - τον χρόνο που μεσολαβεί από την ανερχόμενη ακμή του CLK έως τη μεταβολή της εξόδου
 - ένα άνω φράγμα **t_{pcq} (time of propagation from clock to Q)**
 - ένα κάτω φράγμα **t_{ccq} (time of contamination from clock to Q)**
 - τον χρόνο που πρέπει να είναι σταθερή η είσοδος δεδομένων D **πριν** και **μετά** την ανερχόμενη ακμή του CLK
 - τον χρόνο σταθεροποίησης (πριν) **t_{setup} (set-up time)**
 - τον χρόνο διατήρησης (μετά) **t_{hold} (hold time)**

Προδιαγραφές σύγχρονων ακολουθιακών κυκλωμάτων

- Οι κανόνες της **σύνθεσης σύγχρονων ακολουθιακών κυκλωμάτων** ορίζουν ότι ένα κύκλωμα είναι σύγχρονο ακολουθιακό κύκλωμα αν αποτελείται από αλληλοσυνδεόμενα στοιχεία κυκλωμάτων τέτοια ώστε να ισχύουν τα εξής:
 - *κάθε στοιχείο του κυκλώματος είναι είτε καταχωρητής, είτε συνδυαστικό κύκλωμα*
 - *τουλάχιστον ένα στοιχείο του κυκλώματος είναι καταχωρητής*
 - *όλοι οι καταχωρητές δέχονται το ίδιο σήμα ρολογιού*
 - *κάθε κυκλική διαδρομή περιέχει τουλάχιστον έναν καταχωρητή*
- Κάθε ακολουθιακό κύκλωμα που δεν είναι σύγχρονο ονομάζεται **ασύγχρονο ακολουθιακό κύκλωμα**

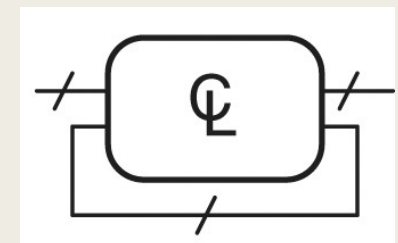
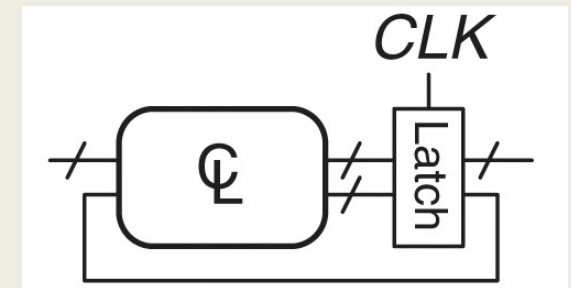
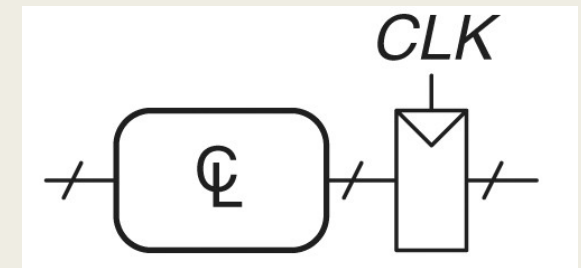
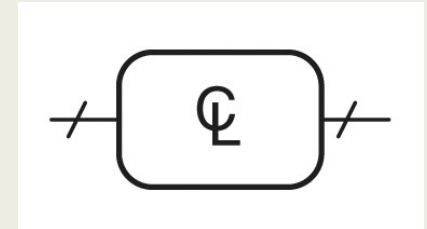
Παράδειγμα 3.5

Ποια από τα παρακάτω κυκλώματα είναι συνδυαστικά κυκλώματα, σύγχρονα ακολουθιακά κυκλώματα ή ασύγχρονα ακολουθιακά κυκλώματα;



Παράδειγμα 3.5

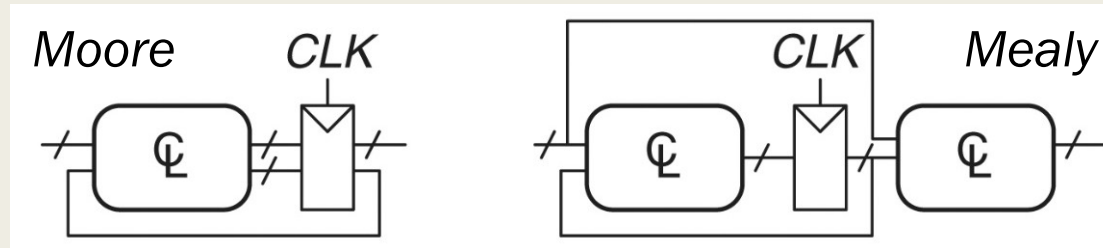
- Το κύκλωμα (α) **είναι συνδυαστικό** δεν περιλαμβάνει ούτε καταχωρητές, ούτε κυκλική διαδρομή
- Το κύκλωμα (β) είναι ένα απλό **σύγχρονο ακολουθιακό κύκλωμα** (συνδυαστική λογική και καταχωρητής)
- Το κύκλωμα (γ) λόγω του Latch είναι **ασύγχρονο ακολουθιακό κύκλωμα**
- Το κύκλωμα (στ) λόγω της κυκλικής διαδρομής ανάδρασης είναι **ασύγχρονο ακολουθιακό κύκλωμα**



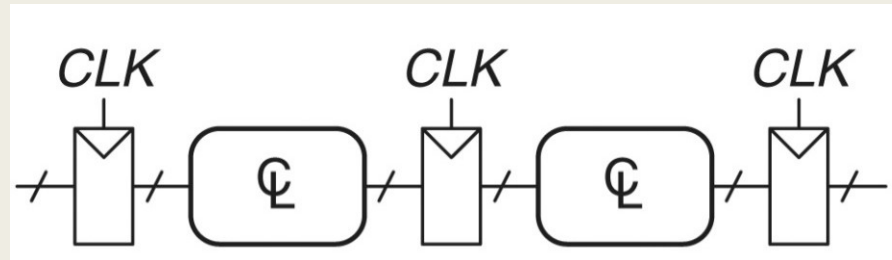
Παράδειγμα 3.5

- Τα κυκλώματα (δ), (ε) και (ζ) παραπέμπουν σε **σύγχρονη ακολουθιακή λογική**

- πρόκειται για τις δύο μορφές **μηχανών πεπερασμένων καταστάσεων**



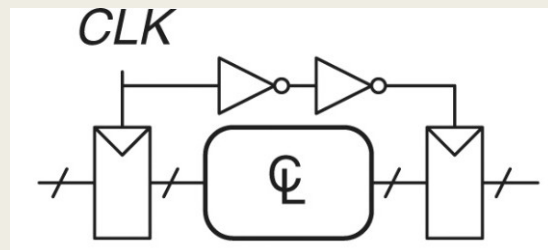
- πρόκειται για συνδεσμολογία **διοχέτευσης (pipelining)**



Σχεδίαση στο επίπεδο της μεταφοράς καταχωρητών
Register Transfer Level – RTL

- Το κύκλωμα (η) **δεν είναι σύγχρονο ακολουθιακό κύκλωμα** με την αυστηρή έννοια του όρου

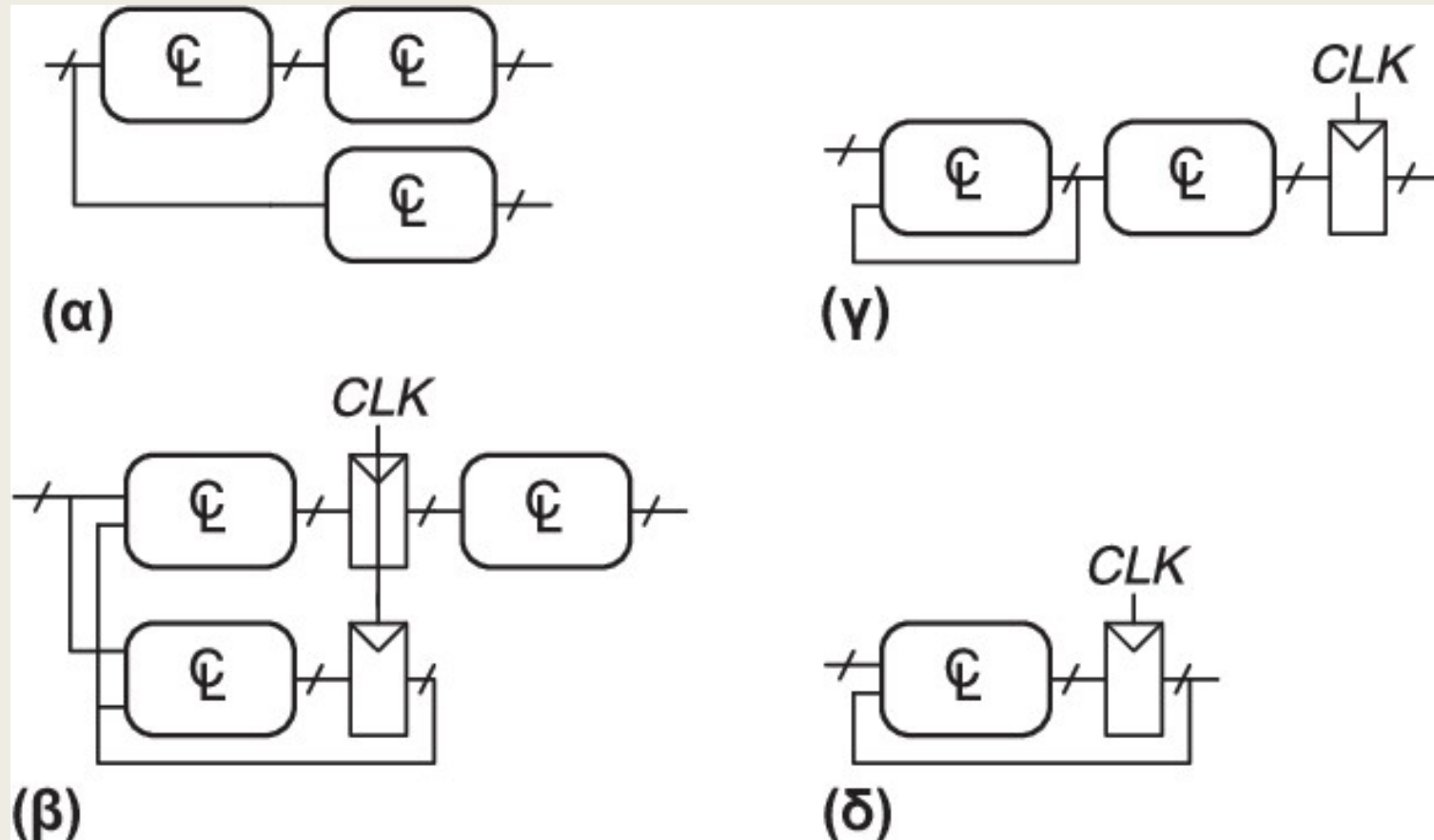
- ο δεύτερος καταχωρητής δέχεται το σήμα με καθυστέρηση δύο πυλών NOT



Επιλεγμένες ασκήσεις

■ Άσκηση 3.18

Ποια από τα κυκλώματα της Εικόνας 3.68 είναι σύγχρονα ακολουθιακά κυκλώματα; Αιτιολογήστε την απάντησή σας

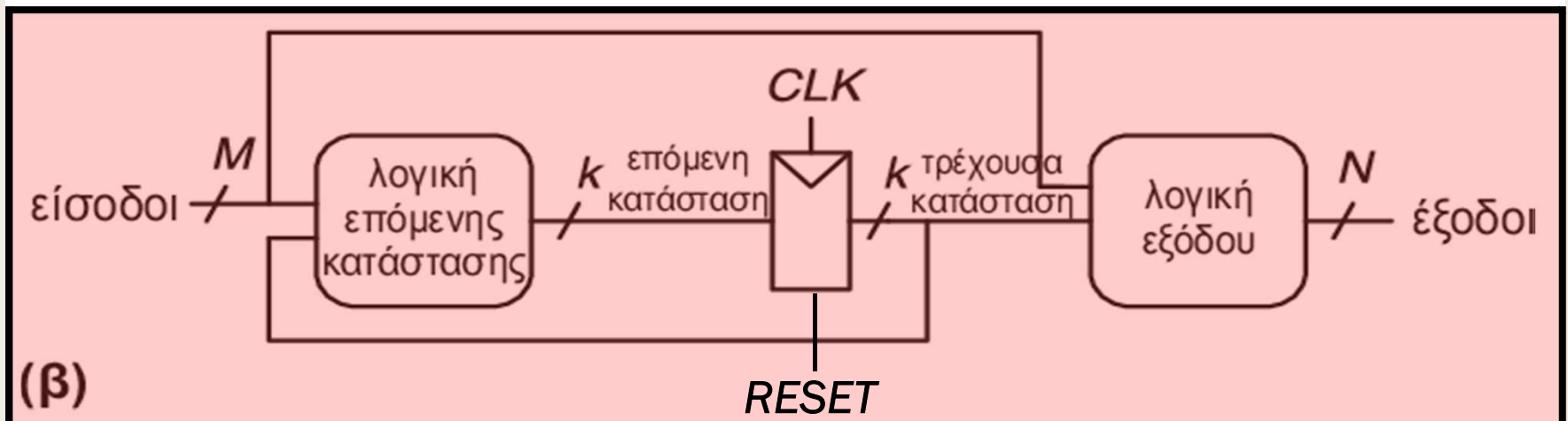
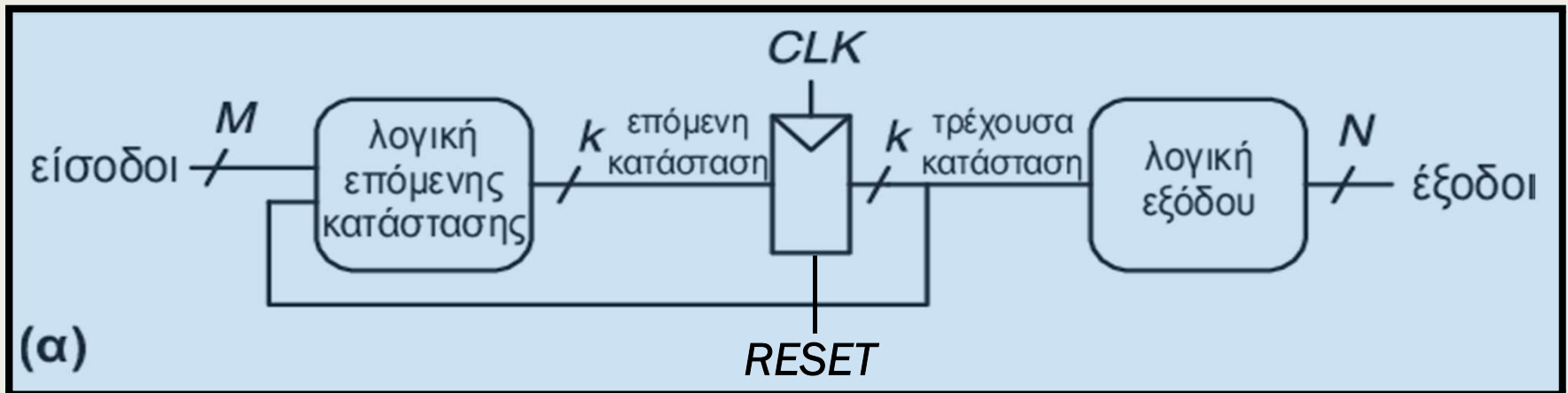


Μηχανές πεπερασμένων καταστάσεων

- Μία **μηχανή πεπερασμένων καταστάσεων** (finite state machines, FSM) είναι ένα σύγχρονο ακολουθιακό κύκλωμα που διαθέτει **M εισόδους**, **N εξόδους**, το σήμα ρολογιού **CLK**, το σήμα επαναφοράς **RESET** και προαιρετικά το σήμα έγκρισης **EN**, και συνίσταται από:
 - έναν **καταχωρητή καταστάσεων (state register) μεγέθους k bit**
 - που αποθηκεύει από **k μέχρι το πολύ 2^k** διακριτές καταστάσεις
 - μία **λογική επόμενης κατάστασης (next state logic)**
 - που υπολογίζει την επόμενη κατάσταση $Q(t+1)$ ως συνάρτηση της τρέχουσας κατάστασης $Q(t)$ και των M εισόδων
 - μία **λογική εξόδου (output logic)**
 - που υπολογίζει τις τιμές των N εξόδων ως συνάρτηση της τρέχουσας κατάστασης $Q(t)$ και προαιρετικά των M εισόδων
- Το σήμα επαναφοράς **RESET** στον καταχωρητή καταστάσεων είναι αναγκαίο ώστε να επιβάλλουμε μια **αρχική κατάσταση** στη μηχανή FSM, όταν την ενεργοποιούμε για πρώτη φορά
- Το προαιρετικό σήμα έγκρισης **EN** χρησιμοποιείται για να καθορίζει αν η μηχανή FSM θα αλλάξει κατάσταση κατά την επόμενη ακμή του CLK

Μηχανές πεπερασμένων καταστάσεων

- Υπάρχουν δύο τύποι μηχανών πεπερασμένης κατάστασης
 - (α) **Μηχανές τύπου Moore** (οι έξοδοι εξαρτώνται μόνο από την παρούσα κατάσταση)
 - (β) **Μηχανές τύπου Mealy** (οι έξοδοι εξαρτώνται από την παρούσα κατάσταση και τις εισόδους)



Edward Moore, 1925-2003 and George Mealy, 1927-2010

- Οι μηχανές Moore και Mealy πήραν τα ονόματά τους από τους ερευνητές των εργαστηρίων Bell Labs που τις σχεδίασαν, οι οποίοι ανέπτυξαν τη θεωρία αυτομάτων (automata theory) που θεμελιώνει μαθηματικά τις μηχανές πεπερασμένων καταστάσεων.
- Ο Edward F. Moore (1925–2003) –προσοχή, δεν πρόκειται για τον ιδρυτή της εταιρείας Intel, Gordon Moore– δημοσίευσε το 1956 το πρωτοποριακό άρθρο του με τίτλο “Gedanken-experiments on Sequential Machines”. Αργότερα έγινε καθηγητής Μαθηματικών και Επιστήμης των Υπολογιστών στο Πανεπιστήμιο του Wisconsin–Madison.
- Ο George H. Mealy (1927–2010) δημοσίευσε το 1955 το δικό του άρθρο με τίτλο “A Method of Synthesizing Sequential Circuits”. Στη συνέχεια έγραψε το πρώτο λειτουργικό σύστημα της εταιρείας Bell Labs για τον υπολογιστή IBM 704. Αργότερα εργάστηκε στην RAND Corporation, στην IBM, όπου εργάστηκε πάνω στην αρχιτεκτονική του λειτουργικού συστήματος 360, και έγινε μέλος του διδακτικού προσωπικού στο Πανεπιστήμιο Harvard.

Μηχανές πεπερασμένων καταστάσεων

- Η μηχανή πεπερασμένων καταστάσεων τύπου Mealy είναι πιο γενική από τη μηχανή τύπου Moore
- Οι μηχανές πεπερασμένων καταστάσεων **τύπου Moore**
 - έχουν **περισσότερες καταστάσεις**
 - συνήθως πλεονεκτούν σε ταχύτητα και μέγεθος της **λογικής εξόδου**
 - Η έξοδος αλλάζει στον **επόμενο κύκλο ρολογιού**
- Οι μηχανές πεπερασμένων καταστάσεων **τύπου Mealy**
 - έχουν **λιγότερες καταστάσεις**
 - συνήθως πλεονεκτούν σε ταχύτητα και μέγεθος της **λογικής επόμενης κατάστασης**
 - Η έξοδος αλλάζει στον **ίδιο κύκλο ρολογιού**
- Μία μηχανή πεπερασμένων καταστάσεων μπορεί να έχει εξόδους και των δύο τύπων

Μηχανές πεπερασμένων καταστάσεων

- Η απόδοση και το κόστος της μηχανής εξαρτάται από:
 - Το **πλήθος των καταστάσεων** (περισσότερες στη μηχανή τύπου Moore)
 - Την **πολυπλοκότητα των διακλαδώσεων** ανά κατάσταση (μεγαλύτερη στη μηχανή τύπου Mealy)
 - Το **μέγεθος (k bit)** του καταχωρητή καταστάσεων
 - Το **πλήθος των ψηφίων που αλλάζουν τιμή** από την τρέχουσα κατάσταση στην επόμενη κατάσταση
 - Εξαρτάται από την κωδικοποίηση των καταστάσεων
 - **Δυαδική** (μέχρι 2^k καταστάσεις, αλλάζουν τιμή περισσότερα από 2 ψηφία κατά μέσο όρο)
 - **Gray** (μέχρι 2^k καταστάσεις, αλλάζει τιμή συνήθως 1 ψηφίο – λιγότερα από 2 ψηφία κατά μέσο όρο)
 - **Μονόθερμη (one-hot)** (μέχρι K καταστάσεις, αλλάζουν τιμή πάντα 2 ψηφία)

Διάγραμμα μεταβολής κατάστασης

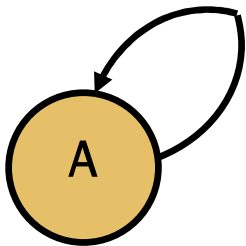
- Το πρώτο στάδιο στη σχεδίαση μίας μηχανής FSM είναι η σχεδίαση του **διαγράμματος μεταβολής κατάστασης** (state transition diagram)
 - Στη σύγχρονη ψηφιακή σχεδίαση το διάγραμμα μεταβολής κατάστασης κωδικοποιείται σε μία γλώσσα περιγραφής υλικού (π.χ. VHDL) και όλα τα υπόλοιπα στάδια της σχεδίασης τα αναλαμβάνει το εργαλείο λογισμικού
- Το διάγραμμα μεταβολής κατάστασης περιγράφει:
 - τις **πιθανές διακριτές καταστάσεις** της μηχανής και πως αυτές μεταβάλλονται σύμφωνα με τις **συνθήκες εισόδου** κατά την ανερχόμενη ακμή του CLK
 - τις αντίστοιχες **εξόδους** τύπου Moore ή Mealy
- Το διάγραμμα μεταβολής κατάστασης απαρτίζεται από:
 - **Κύκλους** που προσδιορίζουν την τρέχουσα κατάσταση $Q(t)$
 - **Βέλη** που προσδιορίζουν τη μετάβαση από την τρέχουσα κατάσταση $Q(t)$ στην επόμενη κατάσταση $Q(t+1)$
 - Όταν υπάρχει μετάβαση με συνθήκη εισόδου, οι τιμές των εισόδων που ικανοποιούν τη συνθήκη γράφονται δίπλα στο βέλος

Διάγραμμα μεταβολής κατάστασης

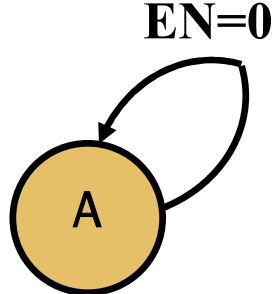
■ Παραδείγματα πιθανών **διαγραμμάτων μεταβολής κατάστασης**

1. Όταν η επόμενη κατάσταση A είναι ίδια με την τρέχουσα κατάσταση A, χωρίς συνθήκη εισόδου
2. Όταν η επόμενη κατάσταση A είναι ίδια με την τρέχουσα κατάσταση A με συνθήκη εισόδου (π.χ. $EN=0$)
3. Όταν η επόμενη κατάσταση B είναι διαφορετική από την τρέχουσα κατάσταση A χωρίς συνθήκη εισόδου
4. Όταν η επόμενη κατάσταση B είναι διαφορετική από την τρέχουσα κατάσταση A με συνθήκη εισόδου (π.χ. $EN=1$)

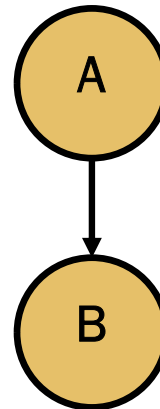
1.



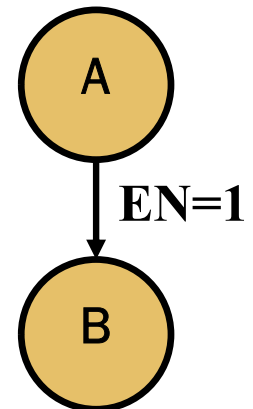
2.



3.

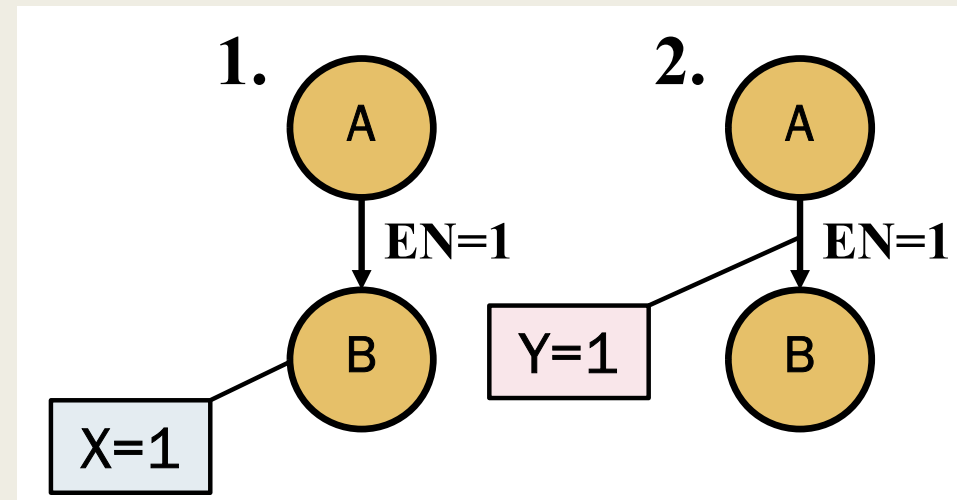


4.



Διάγραμμα μεταβολής κατάστασης

- Επιπλέον, το διάγραμμα μεταβολής κατάστασης απαρτίζεται από:
 - **Πλαίσια** εντός των οποίων γράφονται οι τιμές των εξόδων της μηχανής FSM, που εξαρτώνται αποκλειστικά από την τρέχουσα κατάσταση (**έξοδοι τύπου Moore**)
 - Τα πλαίσια αυτά συνδέονται με τον κύκλο της σχετικής τρέχουσας κατάστασης
 - **Πλαίσια** εντός των οποίων γράφονται οι τιμές των εξόδων της μηχανής FSM, που εξαρτώνται από την τρέχουσα κατάσταση και την αντίστοιχη συνθήκη εισόδου (**έξοδοι τύπου Mealy**).
 - Τα πλαίσια αυτά συνδέονται με το αντίστοιχο βέλος, δίπλα στις τιμές των εισόδων που ικανοποιούν τη συνθήκη
 - Παραδείγματα χρήσης πλαισίων:
 1. Η έξοδος X είναι σταθερά 1 στην κατάσταση B (**τύπου Moore**)
 2. Η έξοδος Y γίνεται 1 στην κατάσταση A, όταν EN είναι 1 ($EN=1$) (**τύπου Mealy**)

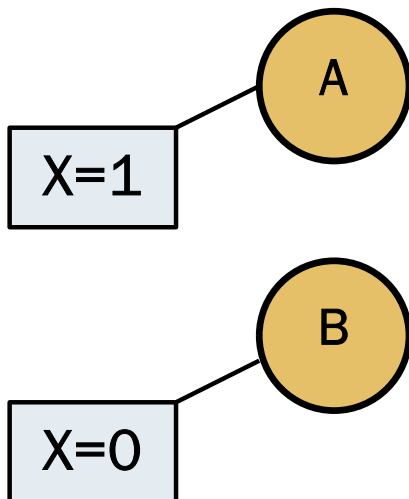


Η χρήση των πλαισίων, αν και δεν υποστηρίζεται στο συγκεκριμένο βιβλίο, διευκολύνει πολύ στην κατανόηση ιδίως όταν το πλήθος των σχετικών εξόδων είναι μεγάλο

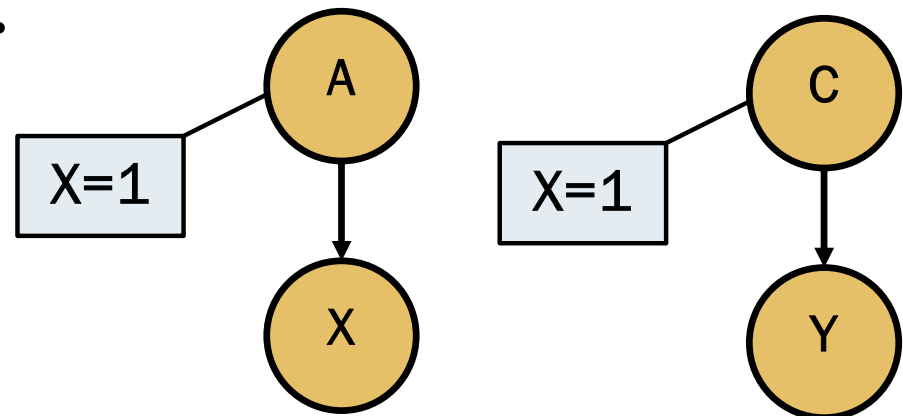
Διαδικασία σχεδίασης μηχανών FSM

- Βήμα 1: Προσδιορίζουμε τις εισόδους, τις εξόδους, και τις διακριτές καταστάσεις
 - Δύο τρέχουσες καταστάσεις χαρακτηρίζονται ως διακριτές μεταξύ τους, εάν :
 - τουλάχιστον μία έξοδος, που εξαρτάται αποκλειστικά από την τρέχουσα κατάσταση, έχει διαφορετική τιμή (παράδειγμα 1), ή/και
 - έχουν τις ίδιες τιμές στις εξόδους, αλλά διαφορετική επόμενη κατάσταση που είναι ανεξάρτητη από τις εισόδους (παράδειγμα 2)
 - Μη διακριτές καταστάσεις ενοποιούνται σε μία κατάσταση, ώστε να προκύψει ελαχιστοποίηση των καταστάσεων

1.



2.

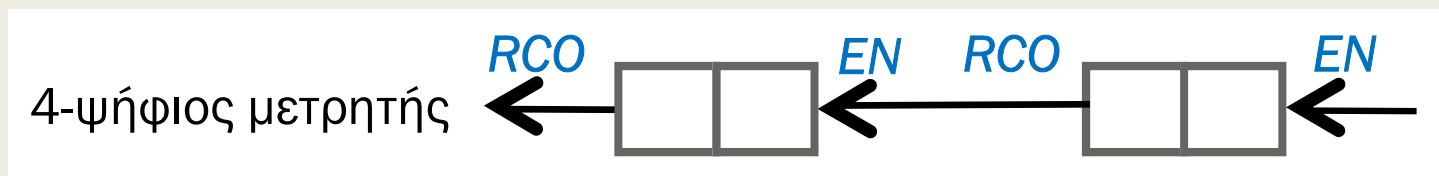


Διαδικασία σχεδίασης μηχανών FSM

- Βήμα 1 (συνέχεια): Προσδιορίζουμε τις **μεταβάσεις** ανάμεσα στις διακριτές καταστάσεις με τις αντίστοιχες **συνθήκες εισόδου**
- Προσδιορίζουμε τις τιμές των **εξόδων** που εξαρτώνται:
 - αποκλειστικά από την **τρέχουσα κατάσταση** (**έξοδοι τύπου Moore**)
 - από την **τρέχουσα κατάσταση** και την αντίστοιχη **συνθήκη εισόδου** (**έξοδοι τύπου Mealy**)
- Βήμα 2: Σχεδιάζουμε το **διάγραμμα μεταβολής κατάστασης**
- Βήμα 3: Σχεδιάζουμε τον **πίνακα μεταβολής κατάστασης**
- Βήμα 4: Επιλέγουμε την **κωδικοποίηση των καταστάσεων** και ενημερώνουμε **τον πίνακα μεταβολής κατάστασης** (μετατρέπεται σε πίνακα αλήθειας)
- Βήμα 5: Απλοποιούμε με K-map και βρίσκουμε τις **εξισώσεις Boole των μεταβλητών των επόμενων καταστάσεων** (συμβολίζουμε με Q^*)
- Βήμα 6: Σχεδιάζουμε **τον πίνακα αληθείας για τις εξόδους**
 - **Τύπου Moore**: παρούσες καταστάσεις, έξοδοι
 - **Τύπου Mealy**: παρούσες καταστάσεις, είσοδοι, έξοδοι
- Βήμα 7: Απλοποιούμε με K-map και βρίσκουμε τις **εξισώσεις Boole των εξόδων**
- Βήμα 8: Σχεδιάζουμε το **σηματικό διάγραμμα**

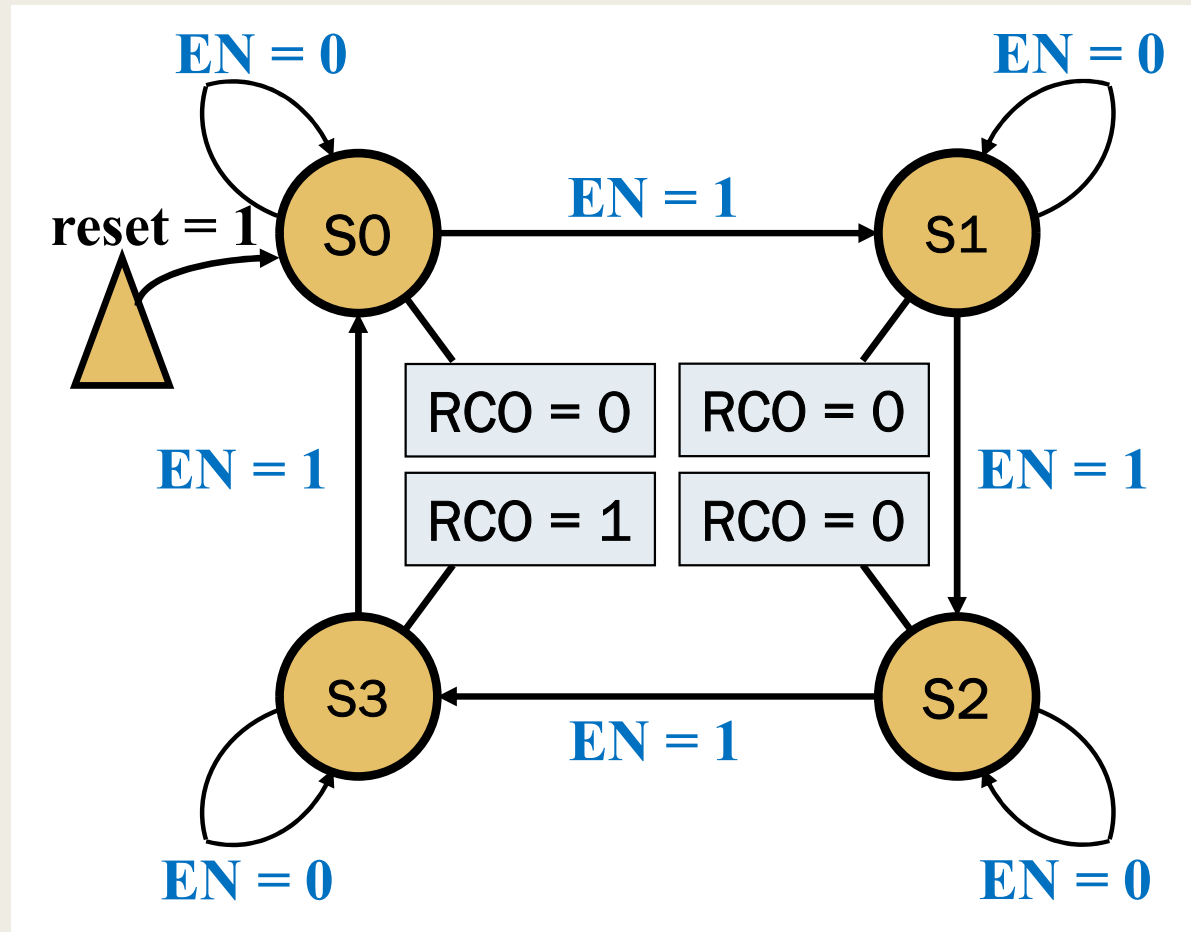
2-ψήφιος σύγχρονος δυαδικός μετρητής

- Βήμα 1: Προσδιορίζουμε τις εισόδους, τις εξόδους, και τις διακριτές καταστάσεις
- Ο μετρητής μετρά στο δυαδικό σύστημα από το 0 μέχρι το 3
 - Έχει 4 διακριτές καταστάσεις *S0, S1, S2 και S3*
- Το σήμα **RESET** αρχικοποιεί τον μετρητή στην κατάσταση S0
 - Η αρχικοποίηση γίνεται σύγχρονα ή ασύγχρονα του CLK με κατάλληλη επιλογή των D Flip-Flop
- Το σήμα έγκρισης **EN** χρησιμοποιείται για να καθορίζει αν ο μετρητής αλλάζει κατάσταση ή όχι κατά την επόμενη ακμή του CLK
 - **EN = 1**: πηγαίνει στην επόμενη κατάσταση
 - **EN = 0**: παραμένει στην τρέχουσα κατάσταση
- Το σήμα **CLK** επιδρά ταυτόχρονα σε όλα τα D Flip-Flop
- Η έξοδος **RCO** (ripple carry output) λαμβάνει την τιμή 1 μόνο όταν ο μετρητής βρίσκεται στην κατάσταση S3
 - Χρησιμοποιείται σε συνδυασμό με το σήμα EN για την κατασκευή ενός μεγάλου μετρητή από πολλούς μικρούς



2-ψήφιος σύγχρονος δυαδικός μετρητής

- Βήμα 2: Σχεδιάζουμε το διάγραμμα μεταβολής κατάστασης

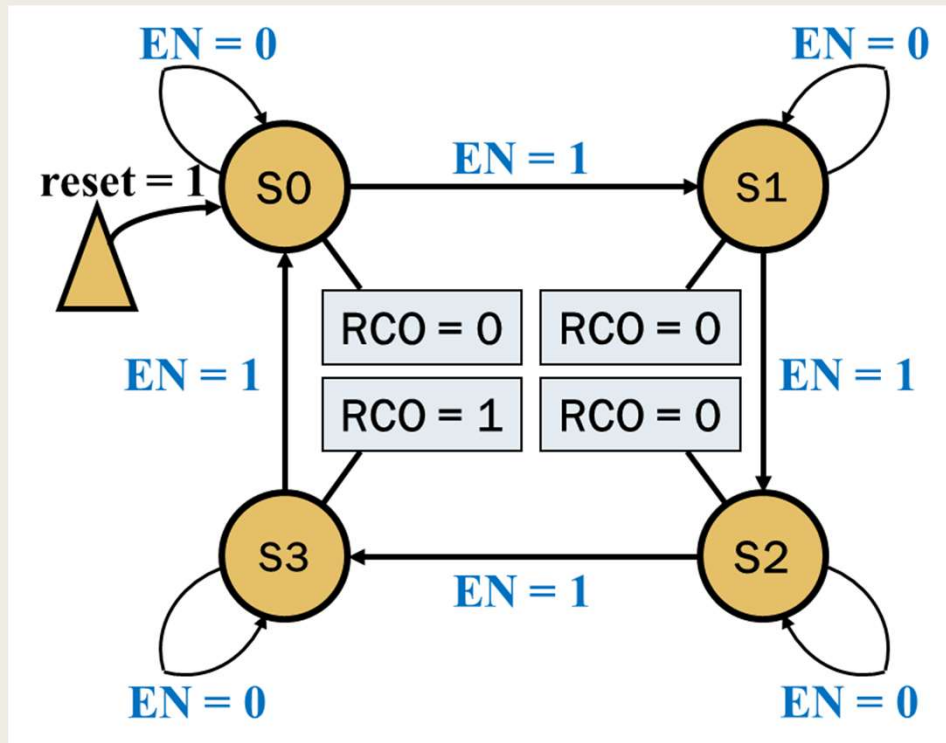


Πίνακας μεταβολής κατάστασης

- Μπορούμε να ορίζουμε το διάγραμμα μεταβολής κατάστασης και με τη μορφή ενός **πίνακα μεταβολής κατάστασης** ο οποίος δείχνει, για κάθε τρέχουσα κατάσταση $Q(t)$ και είσοδο, ποια πρέπει να είναι η επόμενη κατάσταση $Q(t+1)$
 - Στον πίνακα γίνεται χρήση του **συμβόλου** X για τις «αδιάφορες» τιμές οποτεδήποτε η επόμενη κατάσταση δεν εξαρτάται από μια συγκεκριμένη είσοδο.
 - Η είσοδος **RESET** έχει παραλειφθεί από τις εισόδους του πίνακα
 - χρησιμοποιούμε ένα D Flip-Flop με δυνατότητα σύγχρονης ή ασύγχρονης επαναφοράς στην αρχική κατάσταση S_0

2-ψήφιος σύγχρονος δυαδικός μετρητής

- Βήμα 3: Σχεδιάζουμε τον πίνακα μεταβολής κατάστασης



Current State	Inputs	Next State
Q(t)	EN	Q(t+1)
S0	0	S0
S0	1	S1
S1	0	S1
S1	1	S2
S2	0	S2
S2	1	S3
S3	0	S3
S3	1	S0

2-ψήφιος σύγχρονος δυαδικός μετρητής

- Βήμα 4: Κωδικοποιούμε τις καταστάσεις και ενημερώνουμε τον πίνακα μεταβολής κατάστασης (μετατρέπεται σε πίνακα αληθείας για τις μεταβλητές επόμενης κατάστασης)
 - Στους δυαδικούς μετρητές, όπου καταστάσεις και έξοδοι ταυτίζονται χρησιμοποιείται η **δυαδική κωδικοποίηση**

Κατάσταση	Δυαδική Κωδικοποίηση
S0	00
S1	01
S2	10
S3	11

Current State	Inputs	Next State
Q1 Q0	EN	Q1* Q0*
0 0	0	0 0
0 0	1	0 1
0 1	0	0 1
0 1	1	1 0
1 0	0	1 0
1 0	1	1 1
1 1	0	1 1
1 1	1	0 0

2-ψήφιος σύγχρονος δυαδικός μετρητής

- Βήμα 5: Ελαχιστοποιούμε με K-map και βρίσκουμε τις εξισώσεις Boole των μεταβλητών των επόμενων καταστάσεων
 - συμβολίζουμε με $Q1^*$ και $Q0^*$ ($Q^* = Q(t+1)$)

Current State	Inputs	Next State
$Q1 \ Q0$	EN	$Q1^* \ Q0^*$
0 0	0	0 0
0 0	1	0 1
0 1	0	0 1
0 1	1	1 0
1 0	0	1 0
1 0	1	1 1
1 1	0	1 1
1 1	1	0 0

EN	$Q1 \ Q0$			
	00	01	11	10
0	0	0	1	1
1	0	1	0	1

$$Q1^* = \overline{EN}Q1 + Q1\overline{Q0} + EN\overline{Q1}Q0$$

EN	$Q1 \ Q0$			
	00	01	11	10
0	0	1	1	0
1	1	0	0	1

$$Q0^* = \overline{EN}Q0 + EN\overline{Q0}$$

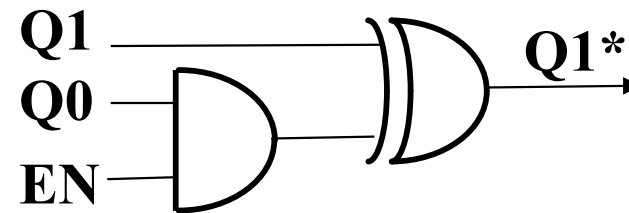
2-ψήφιος σύγχρονος δυαδικός μετρητής

- Βήμα 5 (συνέχεια): Περαιτέρω απλοποίηση υλικού με τη χρήση των πυλών XOR

Current State	Inputs	Next State
Q1 Q0	EN	Q1* Q0*
0 0	0	0 0
0 0	1	0 1
0 1	0	0 1
0 1	1	1 0
1 0	0	1 0
1 0	1	1 1
1 1	0	1 1
1 1	1	0 0

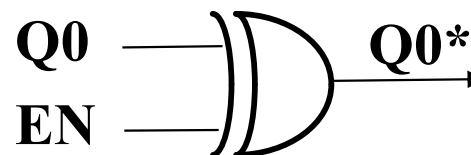
$$Q1^* = \overline{EN}Q1 + Q1\overline{Q0} + EN\overline{Q1}Q0$$

$$\begin{aligned} Q1^* &= \overline{EN} Q1 + Q1 \overline{Q0} + EN \overline{Q1} Q0 \\ Q1^* &= (\overline{EN} + \overline{Q0}) Q1 + EN \overline{Q1} Q0 \\ Q1^* &= (\overline{EN} \overline{Q0}) Q1 + (EN Q0) \overline{Q1} \\ Q1^* &= (EN Q0) \oplus Q1 \end{aligned}$$



$$Q0^* = \overline{EN}Q0 + EN\overline{Q0}$$

$$Q0^* = EN \oplus Q0$$

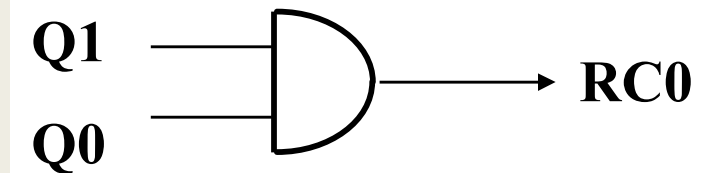


2-ψήφιος σύγχρονος δυαδικός μετρητής

- Βήμα 6: Σχεδιάζουμε τον πίνακα αληθείας για τις εξόδους
 - *Τύπου Moore*: παρούσες καταστάσεις, έξοδοι
- Βήμα 7: Βρίσκουμε τις εξισώσεις Boole των εξόδων

Current State	Outputs
Q1 Q0	RCO
0 0	0
0 1	0
1 0	0
1 1	1

$$RCO = Q1 Q0$$

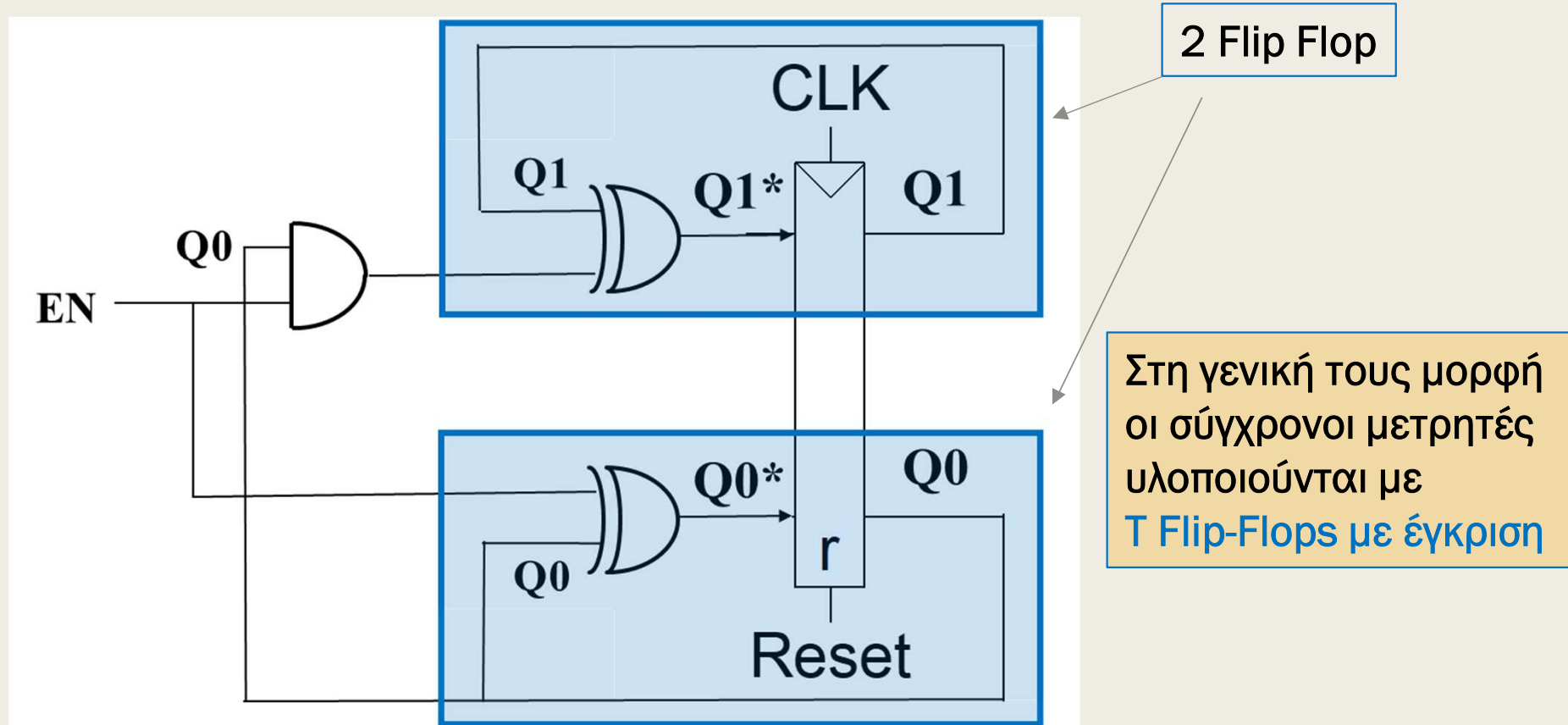


2-ψήφιος σύγχρονος δυαδικός μετρητής

- Βήμα 8: Σχεδιάζουμε το σχηματικό διάγραμμα
 - Καταχωρητής καταστάσεων και λογική επόμενης κατάστασης

$$Q1^* = (EN \ Q0) \oplus Q1$$

$$Q0^* = EN \oplus Q0$$



Εμφανίζονται T Flip-Flop με έγκριση

T Flip-Flop με έγκριση (enabled)

■ Σχεδίαση ενός T Flip-Flop με έγκριση

- Η είσοδος *ENABLE* (*EN*) είναι ενεργή στο *HIGH* (active *HIGH*)

■ Πίνακας Αλήθειας

EN	Q(t)	Q(t + 1)
0	0	0
0	1	1
1	0	1
1	1	0

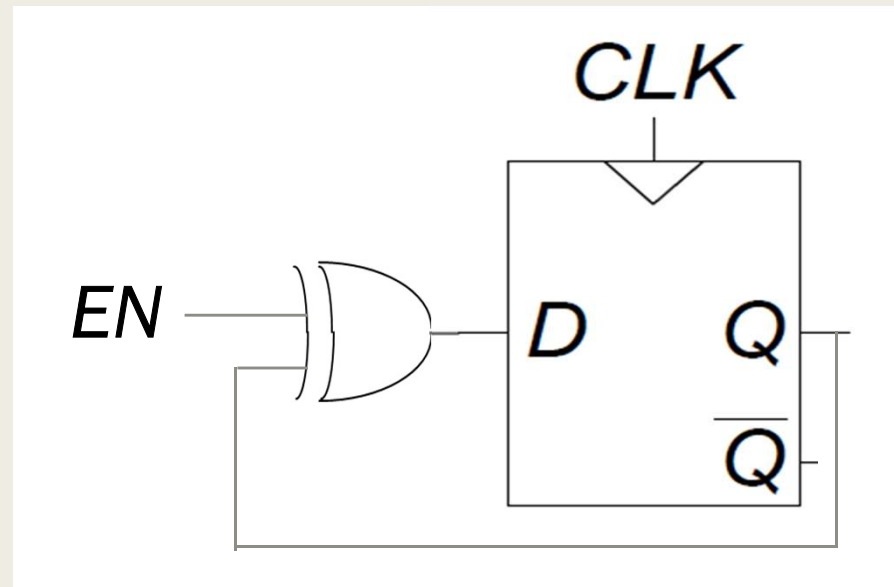
HOLD

TOGGLE

■ Εξίσωση Boole

$$Q(t + 1) = \overline{EN} Q(t) + EN \overline{Q(t)} = EN \oplus Q(t)$$

■ Σχηματικό διάγραμμα



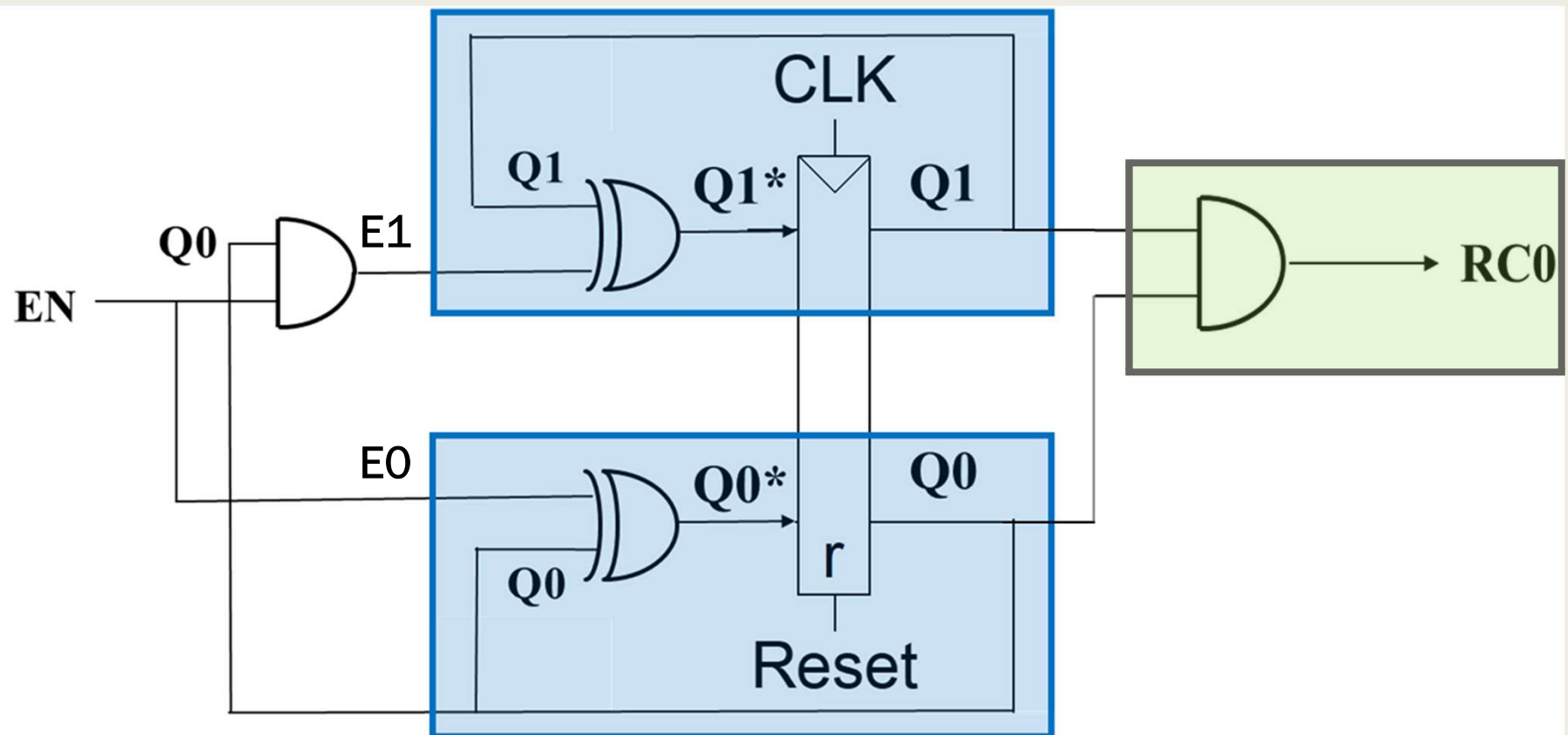
2-ψήφιος σύγχρονος δυαδικός μετρητής

- Βήμα 8 (συνέχεια): Σχεδιάζουμε το σχηματικό διάγραμμα
 - Συμπληρώνουμε και τη λογική εξόδου και τα σήματα έγκρισης $E1$ και $E0$ των T Flip-Flops με έγκριση

$$RC0 = Q1 Q0$$

$$E0 = EN$$

$$E1 = EN Q0$$



Κωδικοποίηση των καταστάσεων

- Στην πράξη συνήθως χρησιμοποιούνται οι ακόλουθες κωδικοποιήσεις:
 - *Binary (δυναδική)*
 - για δυαδικούς μετρητές, όπου καταστάσεις και έξοδοι ταυτίζονται
 - *Μοναδικού σημαντικού (one-hot) – ένα ξεχωριστό bit ανά κατάσταση*
 - για υλοποιήσεις σε FPGA και πλήθος καταστάσεων από 10 μέχρι 30
 - ελαχιστοποιεί τις εξισώσεις Boole, αλλά αυξάνει το μέγεθος του καταχωρητή
 - *Gray ή τροποποιημένη Gray*
 - Η πιο διαδεδομένη γιατί συνδυάζει το μικρότερο δυνατό μέγεθος του καταχωρητή καταστάσεων με αρχική τιμή στο όλα-0 και το ελάχιστο πλήθος των ψηφίων που αλλάζουν τιμή από κατάσταση σε κατάσταση
 - Μόνο ένα ψηφίο στις περισσότερες περιπτώσεις, κάτω από 2 ψηφία κατά μέσο όρο

6 καταστάσεις	One-Hot	Binary	τροπ. Gray
A	000001	000	000
B	000010	001	001
C	000100	010	011
D	001000	011	010
E	010000	100	110
F	100000	101	100 (αντί 111)

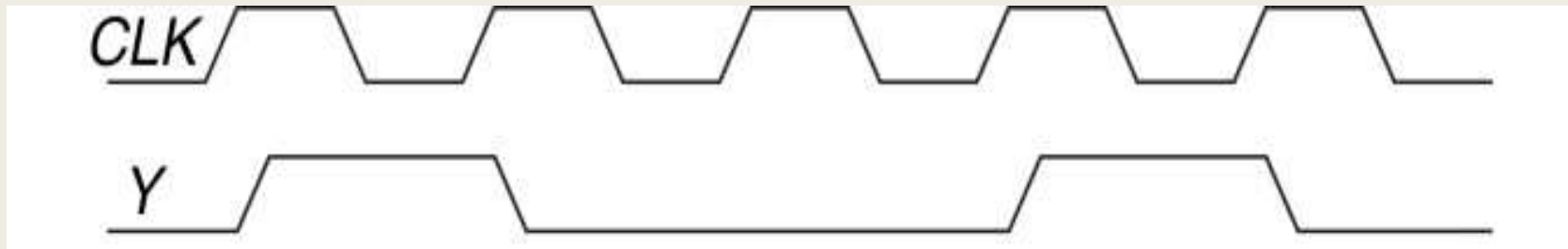
Κωδικοποίηση των καταστάσεων

- **Κωδικοποίηση Gray** (Κατασκευή ακολουθίας για N bit)
 - Για $N=0 \Rightarrow$ κώδικας $\{0\}$
 - Για $N=1 \Rightarrow$ κώδικας $\{0, 1\}$
 - Για $N>1$, τοποθετούμε '0' μπροστά από κάθε κωδική λέξη του κώδικα Gray των $N-1$ bit, και ακολούθως '1' μπροστά από κάθε κωδική λέξη σε αντίστροφη σειρά του κώδικα Gray των $N-1$ bit

N=0	N=1	N=2	N=3
0	0	00	000
	1	01	001
		11	011
		10	010
			110
			111
			101
			100

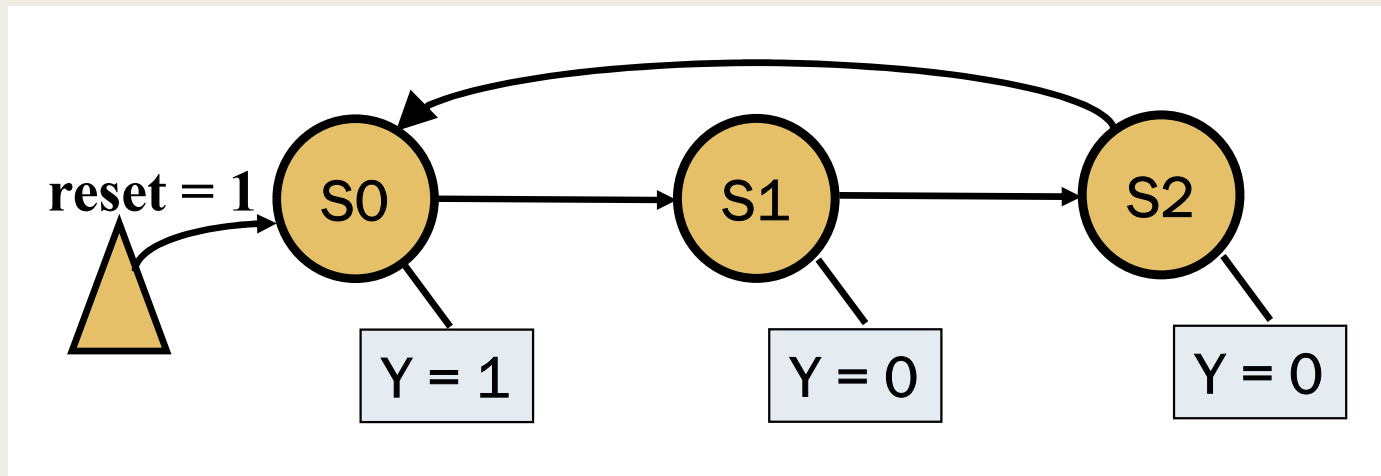
Μετρητής διαίρεσης διά του 3*

- Ένας **μετρητής διαίρεσης διά του N** (divide-by-N counter) διαθέτει μία έξοδο Y και καμία είσοδο
- Η έξοδος Y έχει την τιμή HIGH για έναν από κάθε N κύκλους του CLK
 - Η έξοδος Y διαιρεί τη συχνότητα του ρολογιού δια N
- Δίδεται το διάγραμμα χρονισμού για έναν **μετρητή διαίρεσης διά του 3**
- Μελετήστε την επίδραση που έχει στην πολυπλοκότητα της σχεδίασης η επιλογή της κωδικοποίησης των καταστάσεων
 - (α) δυαδική κωδικοποίηση και
 - (β) κωδικοποίηση μοναδικού σημαντικού (one-hot)



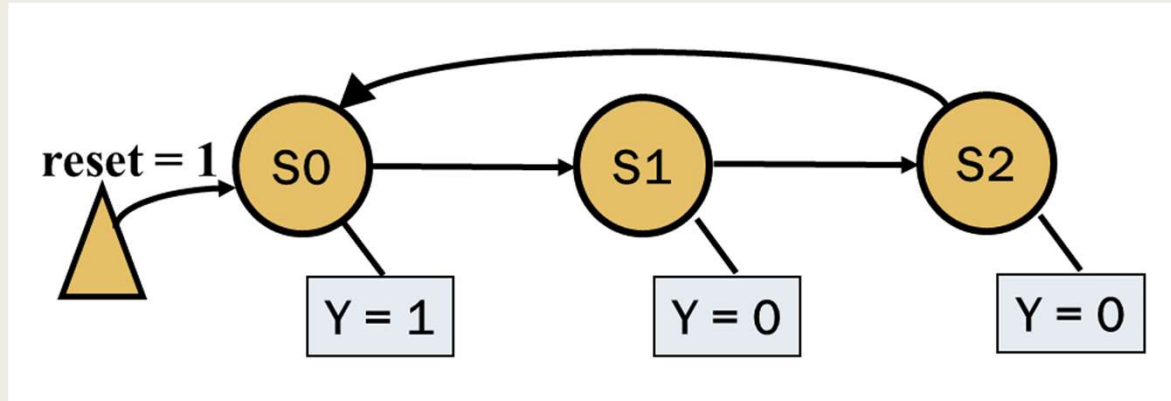
Μετρητής διαίρεσης διά του 3*

- Βήμα 2: Σχεδιάζουμε το διάγραμμα μεταβολής κατάστασης
 - Ο μετρητής έχει 3 διακριτές καταστάσεις $S0$, $S1$ και $S2$
 - Σχηματίζουν έναν κύκλο
 - Το σήμα **RESET** αρχικοποιεί τον μετρητή στην κατάσταση $S0$
 - Η αρχικοποίηση γίνεται σύγχρονα ή ασύγχρονα του CLK με κατάλληλη επιλογή των D Flip-Flop



Μετρητής διαίρεσης διά του 3*

- Βήμα 3: Σχεδιάζουμε τον πίνακα μεταβολής κατάστασης



Current State	Next State
Q(t)	Q(t+1)
S0	S1
S1	S2
S2	S0

- Βήμα 4: Κωδικοποιούμε τις καταστάσεις

Κατάσταση	Δυαδική Κωδικοποίηση	One-hot Κωδικοποίηση
S0	00	001
S1	01	010
S2	10	100

Μετρητής διαίρεσης διά του 3*

- Βήματα 4 & 6: Ενημερώνουμε τον πίνακα μεταβολής κατάστασης και ενσωματώνουμε τον πίνακα αληθείας για την έξοδο Y

Current State	Next State
Q(t)	Q(t+1)
S0	S1
S1	S2
S2	S0

Κατάσταση	Δυαδική Κωδικοποίηση	One-hot Κωδικοποίηση
S0	00	001
S1	01	010
S2	10	100

Δυαδική Κωδικοποίηση		
Current State	Next State	Output
Q1 Q0	Q1* Q0*	Y
0 0	0 1	1
0 1	1 0	0
1 0	0 0	0

One-hot Κωδικοποίηση		
Current State	Next State	Output
Q2 Q1 Q0	Q2* Q1* Q0*	Y
0 0 1	0 1 0	1
0 1 0	1 0 0	0
1 0 0	0 0 1	0

Μετρητής διαίρεσης διά του 3*

- Βήματα 5 & 7: Βρίσκουμε τις εξισώσεις Boole των μεταβλητών των επόμενων καταστάσεων (συμβολίζουμε με Q^*) και της εξόδου

Δυαδική Κωδικοποίηση		
Current State	Next State	Output
Q1 Q0	Q1* Q0*	Y
0 0	0 1	1
0 1	1 0	0
1 0	0 0	0

One-hot Κωδικοποίηση		
Current State	Next State	Output
Q2 Q1 Q0	Q2* Q1* Q0*	Y
0 0 1	0 1 0	1
0 1 0	1 0 0	0
1 0 0	0 0 1	0

$$Q1^* = \overline{Q1} Q0$$

$$Q0^* = \overline{Q1} \overline{Q0}$$

$$Y = \overline{Q1} \overline{Q0}$$

$$Q2^* = Q1$$

$$Q1^* = Q0$$

$$Q0^* = Q2$$

$$Y = Q0$$

Ελαχιστοποιεί τις εξισώσεις Boole, αλλά αυξάνει το μέγεθος του καταχωρητή

Μετρητής διαίρεσης διά του 3*

■ Βήμα 8: Σχεδιάζουμε το σχηματικό διάγραμμα

Δυαδική
κωδικοποίηση
καταστάσεων

$$Q1^* = \overline{Q1} Q0$$

$$Q0^* = \overline{Q1} \overline{Q0}$$

$$Y = \overline{Q1} \overline{Q0}$$

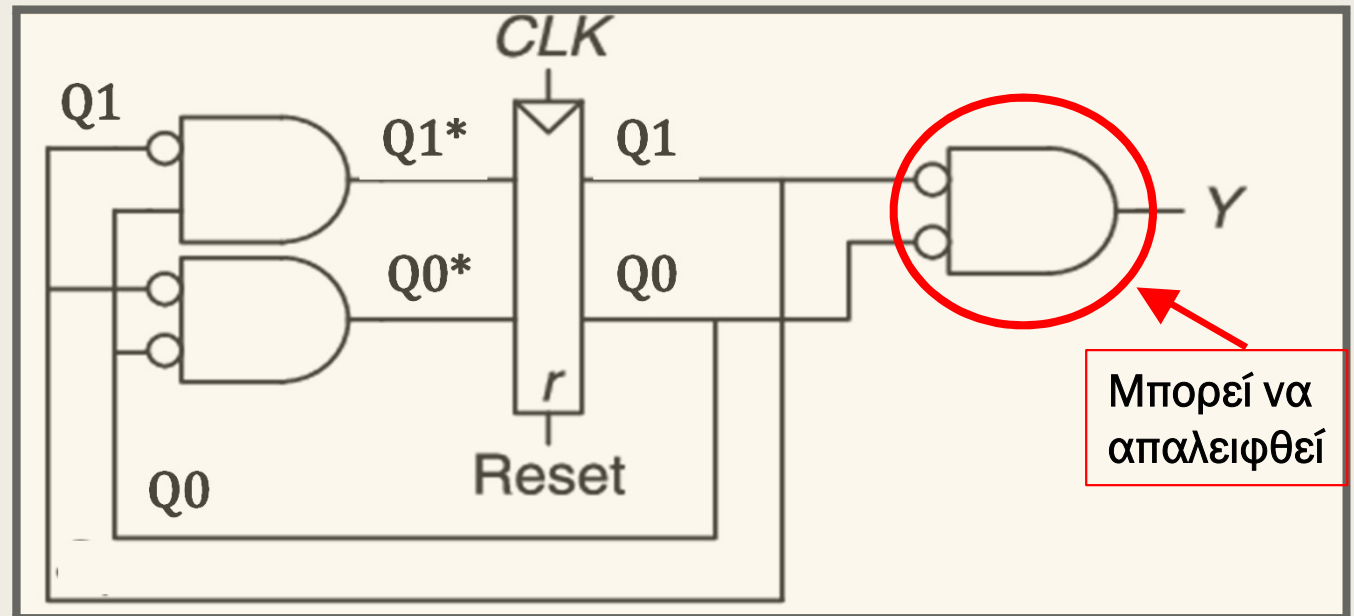
One-hot
κωδικοποίηση
καταστάσεων

$$Q2^* = Q1$$

$$Q1^* = Q0$$

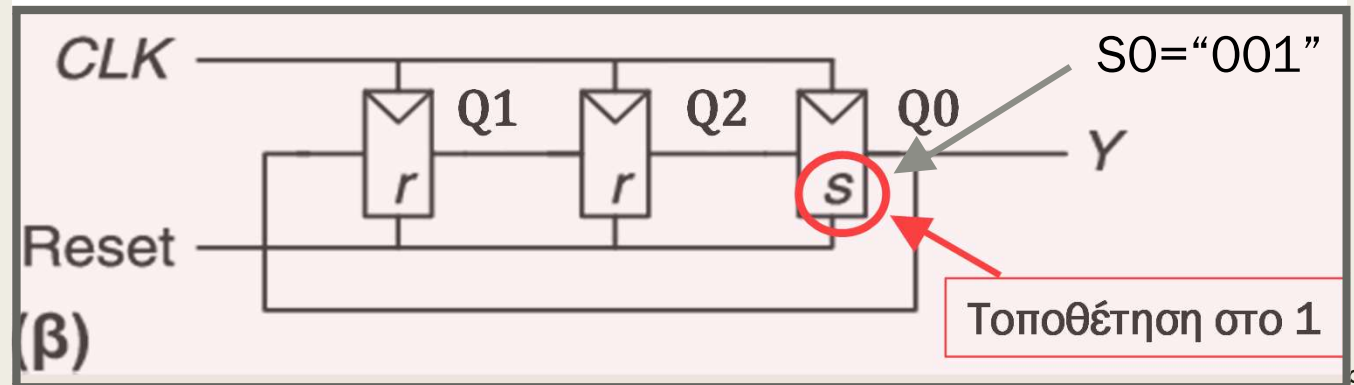
$$Q0^* = Q2$$

$$Y = Q0$$



λογική επόμενης καταστάσης καταχωρητής καταστάσης λογική εξόδου έξοδος

(α)



(β)

Επιλεγμένες ασκήσεις

■ Άσκηση 3.27

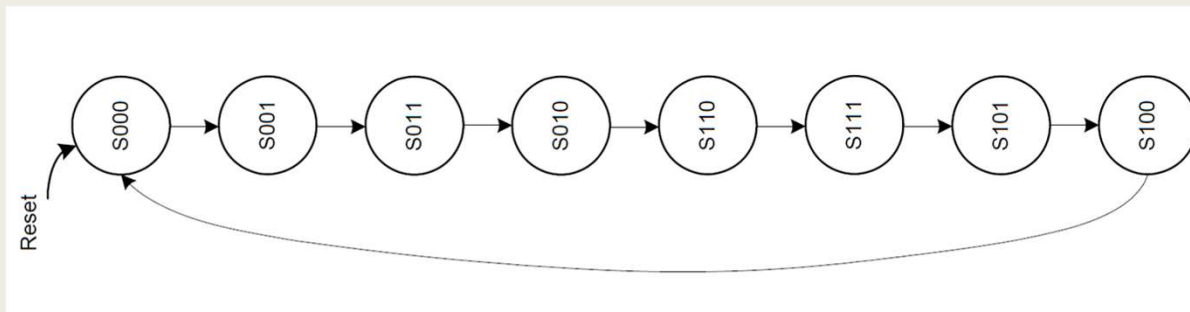
Σχεδιάστε έναν μετρητή FSM υπολοίπου διαίρεσης ως προς το 8 (modulo 8 counter), των 3 bit, ο οποίος χρησιμοποιεί κώδικα Gray

- Ο μετρητής έχει τρεις εξόδους και καμία είσοδο
- Ο μετρητής αρχικοποιείται με το σήμα *RESET* στην κατάσταση 000.
- Σε κάθε ανερχόμενη ακμή του CLK, η έξοδος προχωράει στον επόμενο αριθμό του κώδικα Gray
- Αφού φτάσει στον αριθμό 7 (κατάσταση 100 στον κώδικα Gray), πρέπει να ξεκινάει από την αρχή με τον αριθμό 0 (κατάσταση 000 στον κώδικα Gray).
- Να χρησιμοποιήσετε κωδικοποίηση Gray, ώστε να ταυτίζονται οι εξοδοί με τις καταστάσεις

Αριθμός	Κώδικας Gray		
0	0	0	0
1	0	0	1
2	0	1	1
3	0	1	0
4	1	1	0
5	1	1	1
6	1	0	1
7	1	0	0

Μετρητής Gray των 3 bit

- **Βήμα 1: Προσδιορίζουμε τις εισόδους, τις εξόδους, και τις καταστάσεις**
 - Ο μετρητής δεν διαθέτει άλλη είσοδο πέραν του σήματος *Reset*
 - Οι καταστάσεις ταυτίζονται με τις εξόδους (τύπου *Moore*)
 - Έχει 8 καταστάσεις με κωδικοποίηση Gray
- **Βήμα 2: Σχεδιάζουμε το διάγραμμα μεταβολής κατάστασης**



- **Βήμα 3: Σχεδιάζουμε τον πίνακα μεταβολής κατάστασης**

Current State	Next State
S000	S001
S001	S011
S011	S010
S010	S110
S110	S111
S111	S101
S101	S100
S100	S000

Μετρητής Gray των 3 bit

- Βήμα 4: Κωδικοποιούμε τις καταστάσεις και ενημερώνουμε τον πίνακα μεταβολής κατάστασης

– Διαμορφώνουμε κατάλληλα τις γραμμές του Πίνακα Αλήθειας

Cur. State	Next State
S000	S001
S001	S011
S011	S010
S010	S110
S110	S111
S111	S101
S101	S100
S100	S000

Cur. State Q2 Q1 Q0	Next State Q2* Q1* Q0*
0 0 0	0 0 1
0 0 1	0 1 1
0 1 1	0 1 0
0 1 0	1 1 0
1 1 0	1 1 1
1 1 1	1 0 1
1 0 1	1 0 0
1 0 0	0 0 0

Cur. State Q2 Q1 Q0	Next State Q2* Q1* Q0*
0 0 0	0 0 1
0 0 1	0 1 1
0 1 0	1 1 0
0 1 1	0 1 0
1 0 0	0 0 0
1 0 1	1 0 0
1 1 0	1 1 1
1 1 1	1 0 1

Αναδιάταξη του
προηγούμενου πίνακα
στον δυαδικό κώδικα

Μετρητής Gray των 3 bit

- Βήμα 5: Ελαχιστοποιούμε με K-map και βρίσκουμε τις εξισώσεις Boole των μεταβλητών των επόμενων καταστάσεων (εξόδων)

Cur. State Q2 Q1 Q0	Next State Q2* Q1* Q0*
0 0 0	0 0 1
0 0 1	0 1 1
0 1 0	1 1 0
0 1 1	0 1 0
1 0 0	0 0 0
1 0 1	1 0 0
1 1 0	1 1 1
1 1 1	1 0 1

		Q2 Q1			
Q0		00	01	11	10
	0	0	1	1	0
	1	0	0	1	1

$$Q2^* = Q1 \overline{Q0} + Q2 Q0$$

		Q2 Q1			
Q0		00	01	11	10
	0	0	1	1	0
	1	1	1	0	0

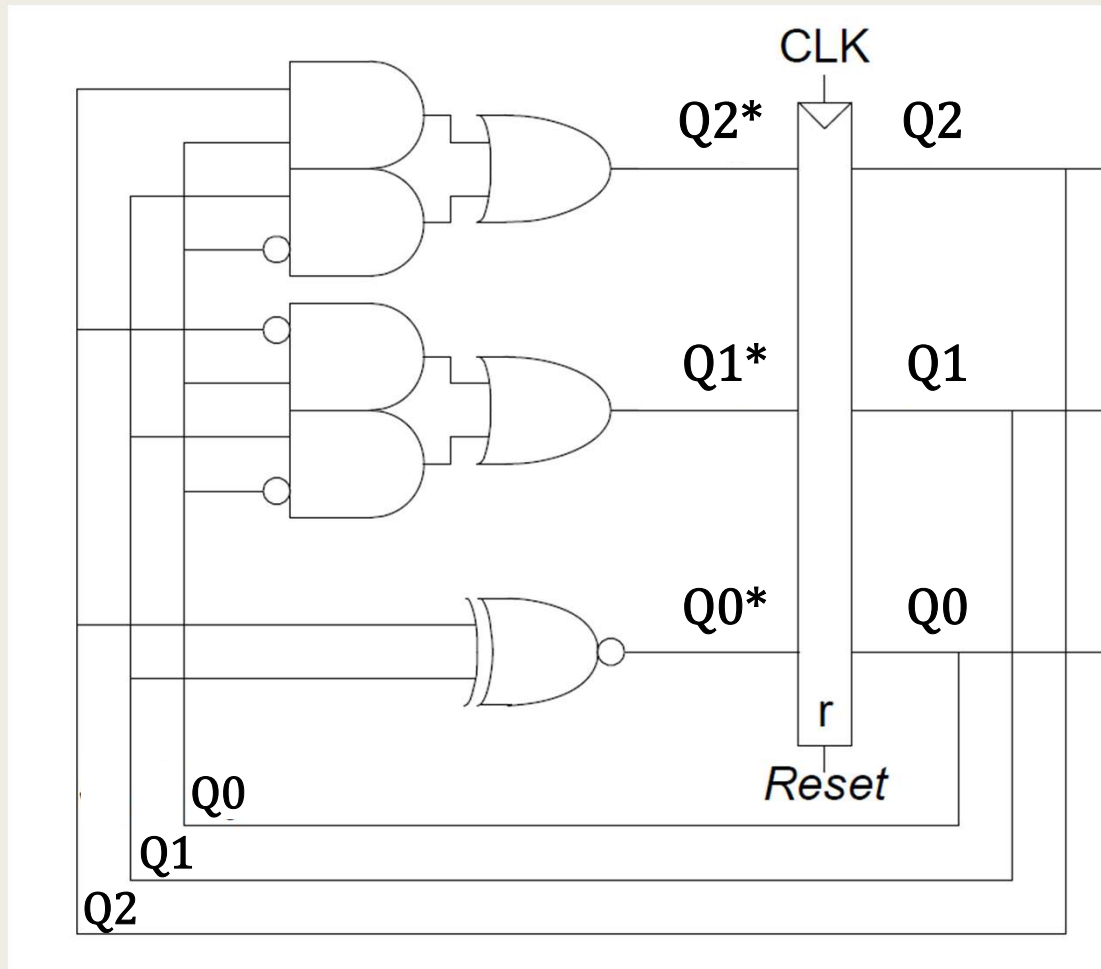
$$Q1^* = Q1 \overline{Q0} + \overline{Q2} Q0$$

		Q2 Q1			
Q0		00	01	11	10
	0	1	0	1	0
	1	1	0	1	0

$$Q0^* = \overline{Q2} \overline{Q1} + Q2 Q1$$

Μετρητής Gray των 3 bit

- Βήμα 8: Σχεδιάζουμε το σχηματικό διάγραμμα



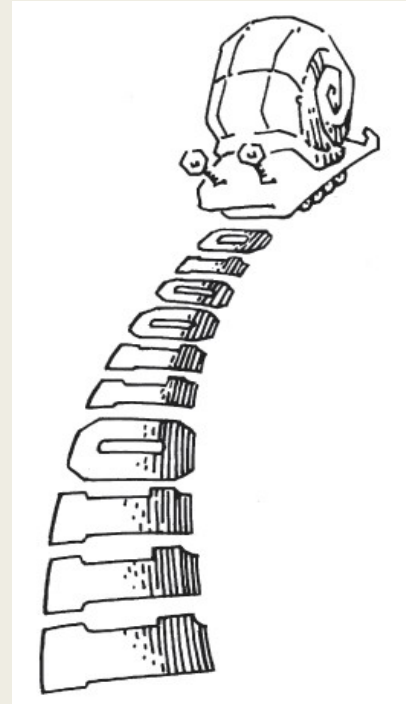
$$Q2^* = Q1 \overline{Q0} + Q2 Q0$$

$$Q1^* = Q1 \overline{Q0} + \overline{Q2} Q0$$

$$Q0^* = \overline{Q2} \overline{Q1} + Q2 Q1$$

Μηχανές FSM: Moore vs Mealy*

- Έστω ότι έχουμε ένα σαλιγκάρι-ρομπότ ως κατοικίδιο, το οποίο διαθέτει μια μηχανή FSM για εγκέφαλο
 - Το σαλιγκάρι έρπει κατά μήκος μιας χάρτινης ταινίας που περιέχει μια ακολουθία από 0 και 1
 - Σε κάθε κύκλο του ρολογιού, το σαλιγκάρι έρπει έως το επόμενο *bit* της ακολουθίας
 - Το σαλιγκάρι χαμογελάει όταν τα δύο τελευταία **διαδοχικά bit** πάνω από τα οποία έχει περάσει είναι **01**
- Σχεδιάστε τη μηχανή FSM έτσι ώστε να υπολογίζει πότε το σαλιγκάρι πρέπει να χαμογελάει
 - Η μηχανή FSM του σαλιγκαριού-ρομπότ είναι ένας **ανιχνευτής ακολουθίας 2 διαδοχικών bit** που μεταδίδονται σειριακά στην είσοδο X
 - Το σαλιγκάρι χαμογελάει όταν η έξοδος Y γίνεται 1
- Συγκρίνετε τις σχεδιάσεις των μηχανών FSM **τύπων Moore και Mealy**
- Σχεδιάστε ένα **διάγραμμα χρονισμού** για κάθε τύπο της μηχανής FSM
 - στο οποίο θα φαίνονται η είσοδος, οι καταστάσεις, και η έξοδος καθώς το σαλιγκάρι έρπει κατά μήκος της ακολουθίας **0100110111**



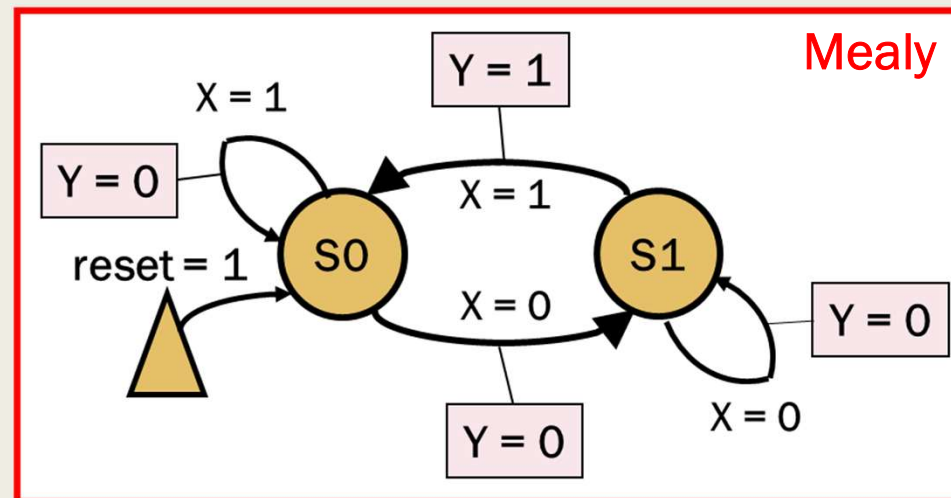
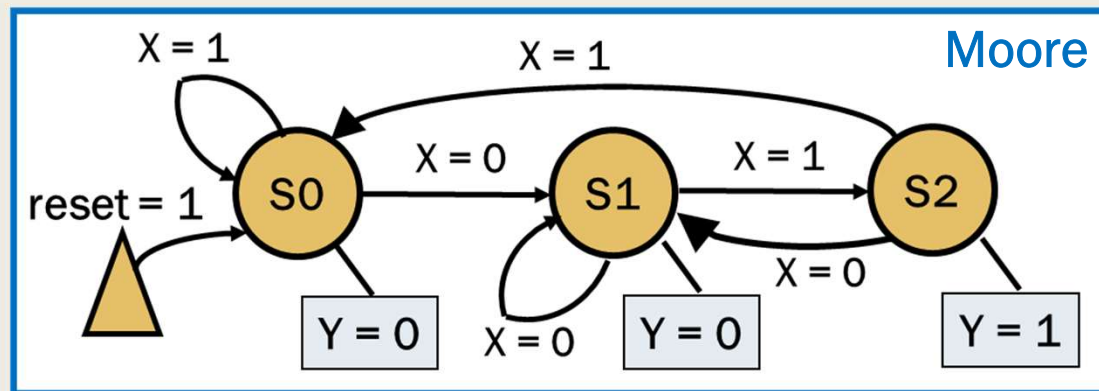
Ανιχνευτής ακολουθίας 2 διαδοχικών bit

■ Βήμα 1: Προσδιορίζουμε τις εισόδους, τις εξόδους, και τις καταστάσεις

- Ο ανιχνευτής έχει μία σειριακή είσοδο X και μία έξοδο Y
 - $Y = 1$, όταν τα δύο τελευταία διαδοχικά bit της εισόδου X είναι 01
- Καταστάσεις μηχανής **τύπου Moore**:
 - $S0$ = αρχική κατάσταση, δεν έχει ανιχνευθεί κανένα ψηφίο, $Y = 0$
 - $S1$ = έχει ανιχνευθεί στην είσοδο X ένα bit 0, $Y = 0$
 - $S2$ = έχουν ανιχνευθεί στην είσοδο X δύο διαδοχικά bit 01, $Y = 1$
- Καταστάσεις μηχανής **τύπου Mealy**:
 - $S0$ = αρχική κατάσταση δεν έχει ανιχνευθεί κανένα ψηφίο
 - εάν $X = 1$, τότε $Y = 0$ και παραμένει στην $S0$
 - εάν $X = 0$, τότε $Y = 0$ και πηγαίνει στην $S1$ (ανίχνευση 0)
 - $S1$ = έχει ανιχνευθεί στην είσοδο X ένα bit 0
 - εάν $X = 1$, τότε $Y = 1$ και πηγαίνει στην $S0$ (ανίχνευση 01)
 - εάν $X = 0$, τότε $Y = 0$ και παραμένει στην $S1$ (ανίχνευση 0)
 - Δεν χρειάζεται η κατάσταση $S2$ για τον προσδιορισμό του $Y = 1$
 - Η έξοδος $Y = 1$ προσδιορίζεται από την είσοδο $X = 1$, όταν βρίσκεται στην κατάσταση $S1$

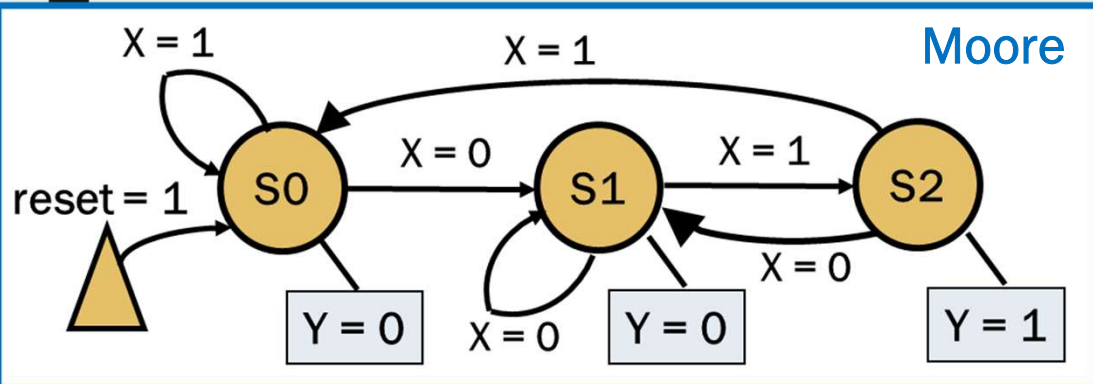
Ανιχνευτής ακολουθίας 2 διαδοχικών bit

- Βήμα 2: Σχεδιάζουμε τα διαγράμματα μεταβολής κατάστασης
 - Ο ανιχνευτής τύπου Moore έχει 3 καταστάσεις $S0$, $S1$ και $S2$
 - Ο ανιχνευτής τύπου Mealy έχει 2 καταστάσεις $S0$ και $S1$
 - Το σήμα **RESET** αρχικοποιεί τον ανιχνευτή στην κατάσταση $S0$
 - Η αρχικοποίηση του ανιχνευτή γίνεται σύγχρονα ή ασύγχρονα του CLK με κατάλληλη επιλογή των D Flip-Flop

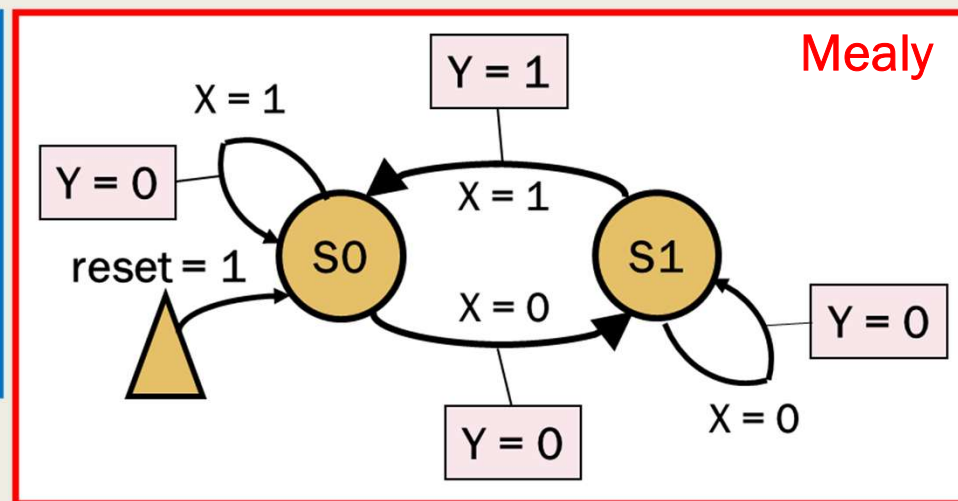


Ανιχνευτής ακολουθίας 2 διαδοχικών bit

- Βήμα 3: Σχεδιάζουμε τους πίνακες μεταβολής κατάστασης



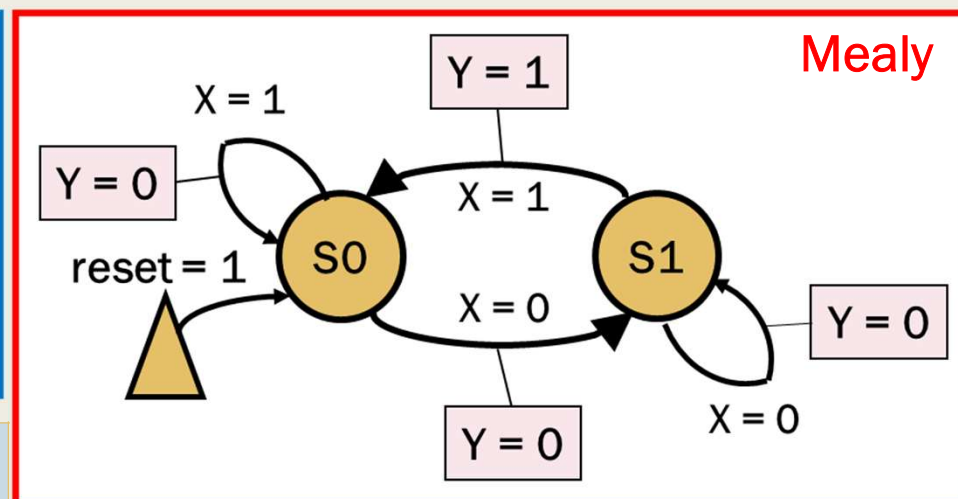
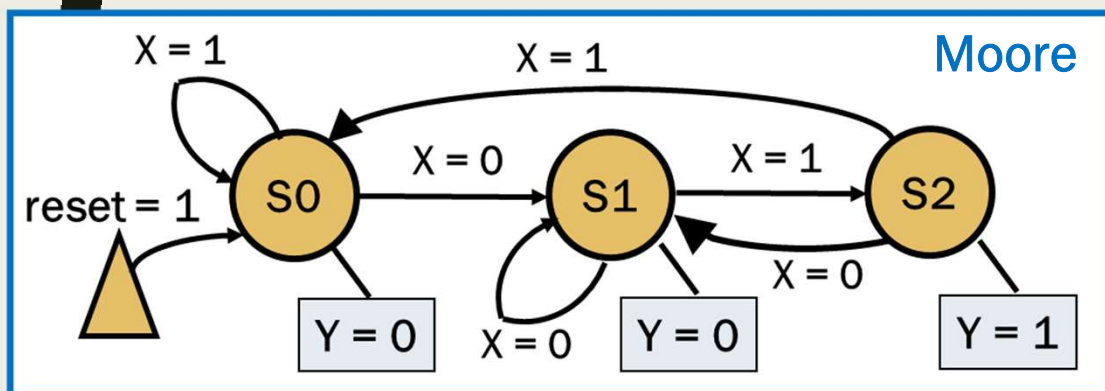
Current State	In put	Next State
Q(t)	X	Q(t+1)
S0	0	S1
S0	1	S0
S1	0	S1
S1	1	S2
S2	0	S1
S2	1	S0



Current State	In put	Next State
Q(t)	X	Q(t+1)
S0	0	S1
S0	1	S0
S1	0	S1
S1	1	S0

Ανιχνευτής ακολουθίας 2 διαδοχικών bit

- Βήμα 4: Κωδικοποιούμε τις καταστάσεις και ενημερώνουμε τους πίνακες μεταβολής κατάστασης



Current State	In put	Next State	Κατά σταση	Κωδικο ποίηση
Q1 Q0	X	Q1* Q0*	S0	00
0 0	0	0 1	S1	01
0 0	1	0 0	S2	10
0 1	0	0 1		
0 1	1	1 0		
1 0	0	0 1		
1 0	1	0 0		

Current State	In put	Next State	Κατ.	Κωδ.
Q	X	Q*	S0	0
0	0	1	S1	1
0	1	0		
1	0	1		
1	1	0		

Ανιχνευτής ακολουθίας 2 διαδοχικών bit

- Βήμα 5: Ελαχιστοποιούμε με K-map και βρίσκουμε τις εξισώσεις Boole των μεταβλητών των επόμενων καταστάσεων

Moore

Current State	In put	Next State
Q1 Q0	X	Q1* Q0*
0 0	0	0 1
0 0	1	0 0
0 1	0	0 1
0 1	1	1 0
1 0	0	0 1
1 0	1	0 0
1 1	0	X X
1 1	1	X X

$$Q1^* = Q0 X$$

$$Q0^* = \bar{X}$$

Mealy

Current State	In put	Next State
Q	X	Q*
0	0	1
0	1	0
1	0	1
1	1	0

$$Q^* = \bar{X}$$

Ανιχνευτής ακολουθίας 2 διαδοχικών bit

- Βήμα 6: Σχεδιάζουμε τους πίνακες αληθείας για τις εξόδους
 - *Τύπου Moore*: παρούσες καταστάσεις, έξοδοι
 - *Τύπου Mealy*: παρούσες καταστάσεις, είσοδοι, έξοδοι
- Βήμα 7: Βρίσκουμε τις εξισώσεις Boole των εξόδων

Moore

Current State	Output
Q1 Q0	Y
0 0	0
0 1	0
1 0	1
1 1	X

$$Q1^* = Q0 X$$

$$Q0^* = \bar{X}$$

$$Y = Q1$$

Mealy

Current State	Input	Output
Q	X	Y
0	0	0
0	1	0
1	0	0
1	1	1

$$Y = Q X$$

$$Q^* = \bar{X}$$

Ανιχνευτής ακολουθίας 2 διαδοχικών bit

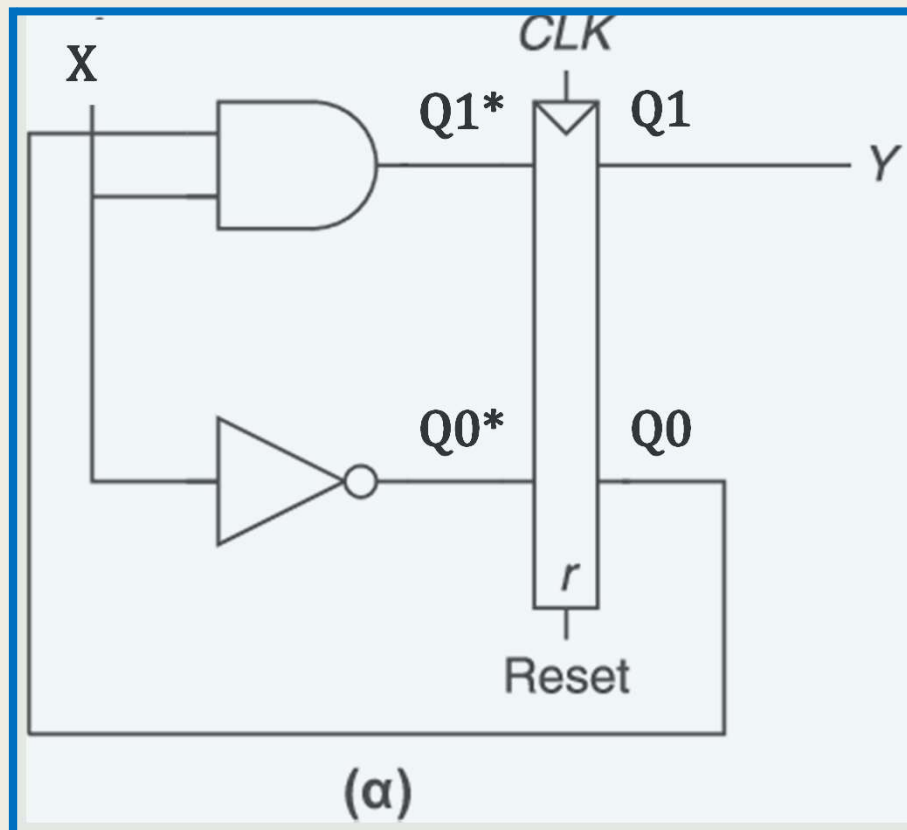
- Βήμα 8: Σχεδιάζουμε το σχηματικό διάγραμμα

Moore

$$Q1^* = Q0 X$$

$$Q0^* = \bar{X}$$

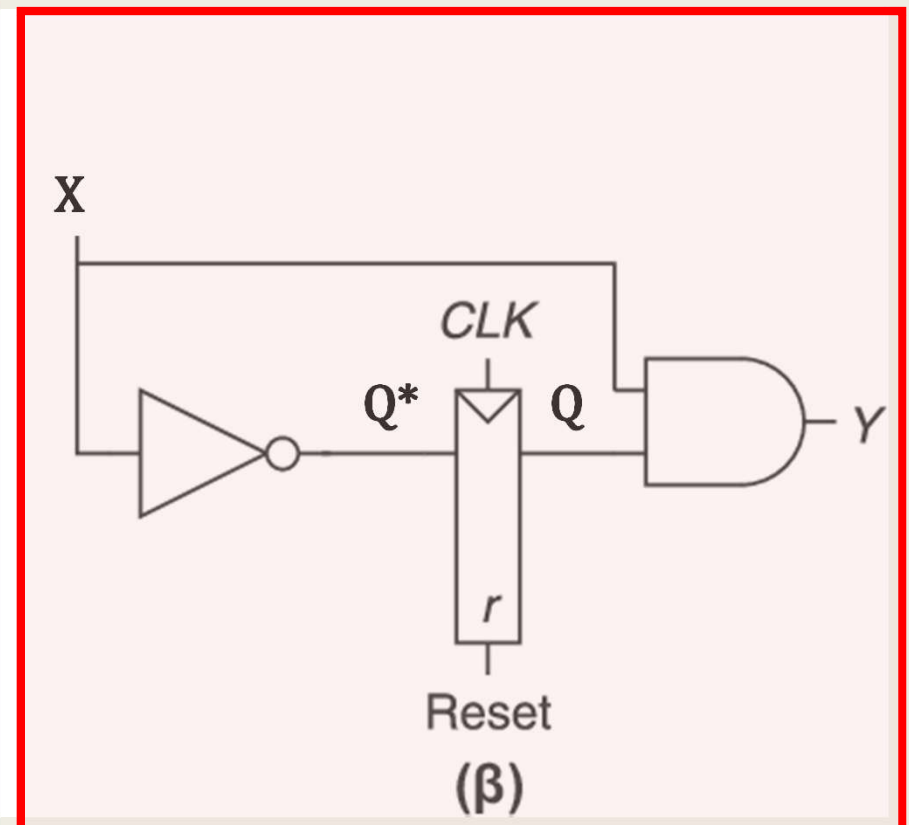
$$Y = Q1$$



Mealy

$$Y = Q X$$

$$Q^* = \bar{X}$$



Ανιχνευτής ακολουθίας 2 διαδοχικών bit

■ Διάγραμμα χρονισμού

Moore

$$Q1^* = Q0 X$$

$$Q0^* = \bar{X}$$

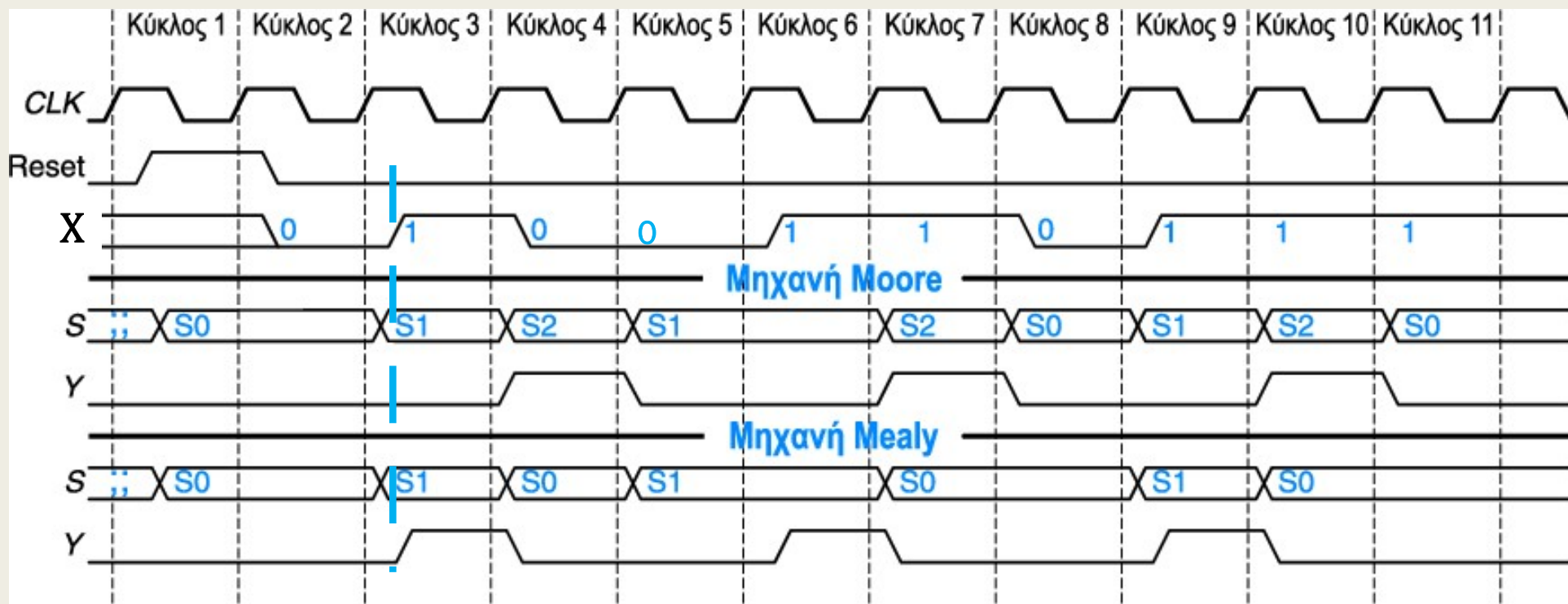
$$Y = Q1$$

Mealy

$$Y = Q X$$

$$Q^* = \bar{X}$$

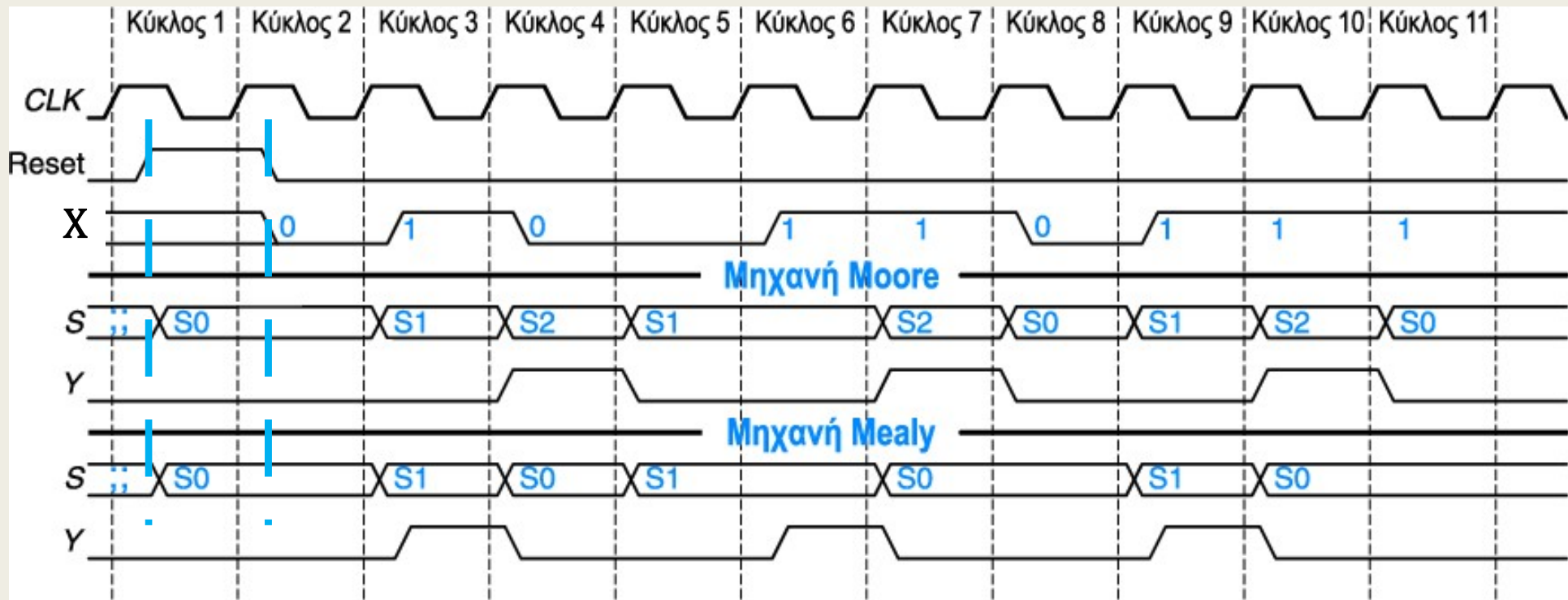
Αν η έξοδος Y της μηχανής τύπου Mealy καθυστερούσε μέσω ενός D Flip-Flop, τότε θα ταίριαζε με την έξοδο Y της μηχανής τύπου Moore



Ανιχνευτής ακολουθίας 2 διαδοχικών bit

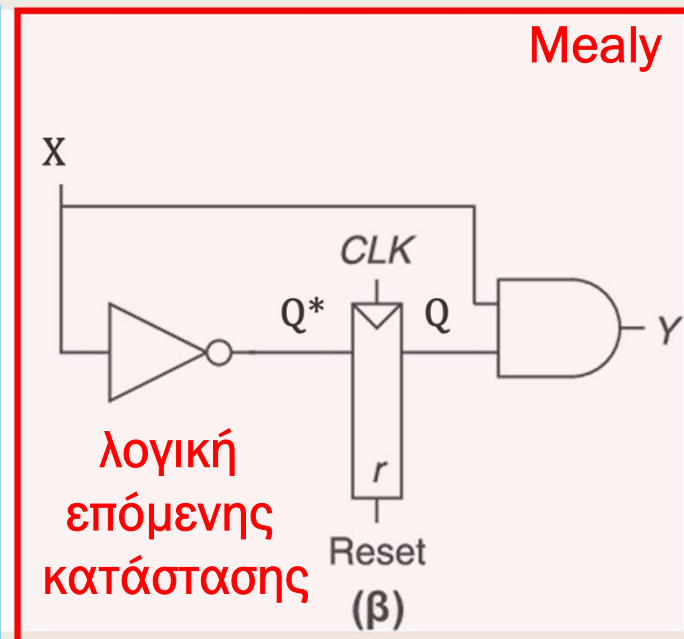
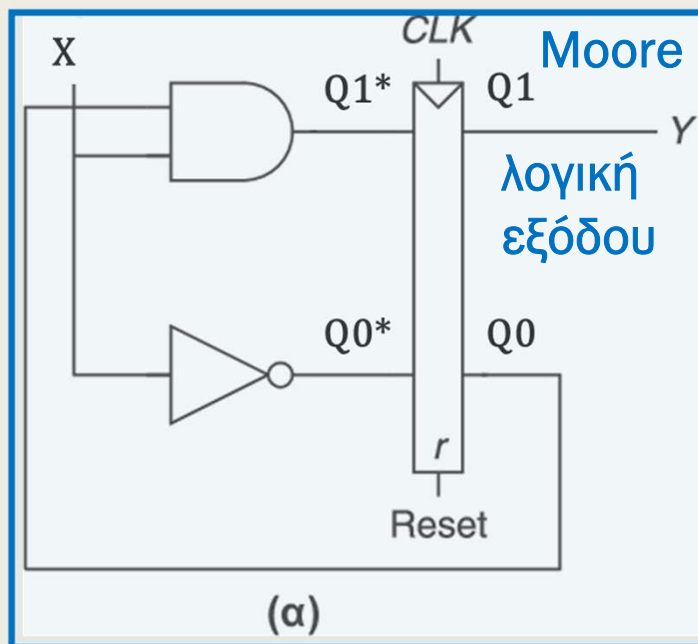
■ Διάγραμμα χρονισμού

- Όταν ενεργοποιηθεί για πρώτη φορά η μηχανή FSM, η κατάστασή της είναι **απροσδιόριστη (::)**
- Με την ενεργοποίηση του **ασύγχρονου σήματος Reset** η μηχανή τίθεται στην αρχική κατάσταση S0 (πριν την ανερχόμενη ακμή του CLK)
- Με την απενεργοποίηση του ασύγχρονου σήματος Reset η είσοδος X παίρνει την πρώτη τιμή της (X=0 στον κύκλο 2)
- Οι δύο μηχανές ακολουθούν **διαφορετική ακολουθία καταστάσεων**
- Η άνοδος της εξόδου Y της μηχανής Mealy συμβαίνει **έναν κύκλο νωρίτερα**, επειδή η έξοδος αποκρίνεται στην είσοδο αντί να περιμένει για τη μεταβολή της κατάστασης



Μηχανές FSM: Moore vs Mealy*

- Η μηχανή πεπερασμένων καταστάσεων τύπου Mealy είναι πιο γενική από τη μηχανή τύπου Moore
 - Η **έξοδος εμφανίζεται πιο γρήγορα** (σε λιγότερους κύκλους)
- Οι μηχανές πεπερασμένων καταστάσεων **τύπου Moore**
 - έχουν **περισσότερες καταστάσεις**
 - συνήθως πλεονεκτούν σε ταχύτητα και μέγεθος της **λογικής εξόδου**
- Οι μηχανές πεπερασμένων καταστάσεων **τύπου Mealy**
 - έχουν **λιγότερες καταστάσεις**
 - συνήθως πλεονεκτούν σε ταχύτητα και μέγεθος της **λογικής επόμενης κατάστασης**



Ανιχνευτής ακολουθίας 4 διαδοχικών bit

- Σχεδιάστε το διάγραμμα μεταβολής κατάστασης για τον ανιχνευτή της ακολουθίας **0101**
 - Στηρίζεται σε μηχανή τύπου *Moore*

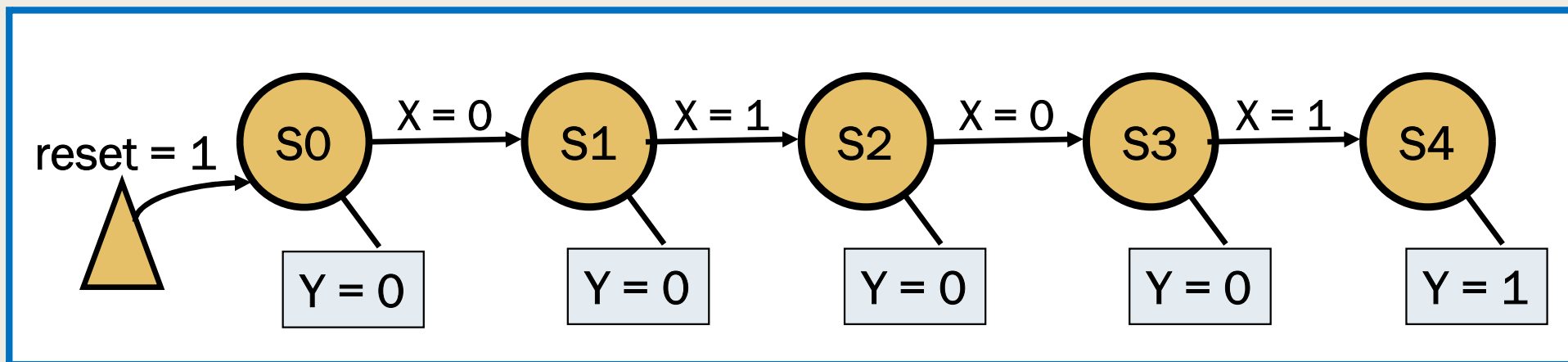
Ανιχνευτής ακολουθίας 4 διαδοχικών bit

- **Βήμα 1: Προσδιορίζουμε τις εισόδους, τις εξόδους, και τις καταστάσεις**
 - Ο ανιχνευτής έχει μία σειριακή είσοδο X και μία έξοδο Y
 - $Y = 1$, όταν τα 4 τελευταία διαδοχικά bit της εισόδου X είναι 0101
 - Καταστάσεις μηχανής **τύπου Moore**:
 - $S0$ = αρχική κατάσταση, δεν έχει ανιχνευθεί κανένα ψηφίο, $Y = 0$
 - $S1$ = έχει ανιχνευθεί στην είσοδο X ένα bit 0, $Y = 0$
 - $S2$ = έχουν ανιχνευθεί στην είσοδο X δύο διαδοχικά bit 01, $Y = 0$
 - $S3$ = έχουν ανιχνευθεί στην είσοδο X τρία διαδοχικά bit 010, $Y = 0$
 - $S4$ = έχουν ανιχνευθεί στην είσοδο X τέσσερα διαδοχικά bit 0101, $Y = 1$

Ανιχνευτής ακολουθίας 4 διαδοχικών bit

■ Βήμα 2: Σχεδιάζουμε το διάγραμμα μεταβολής κατάστασης

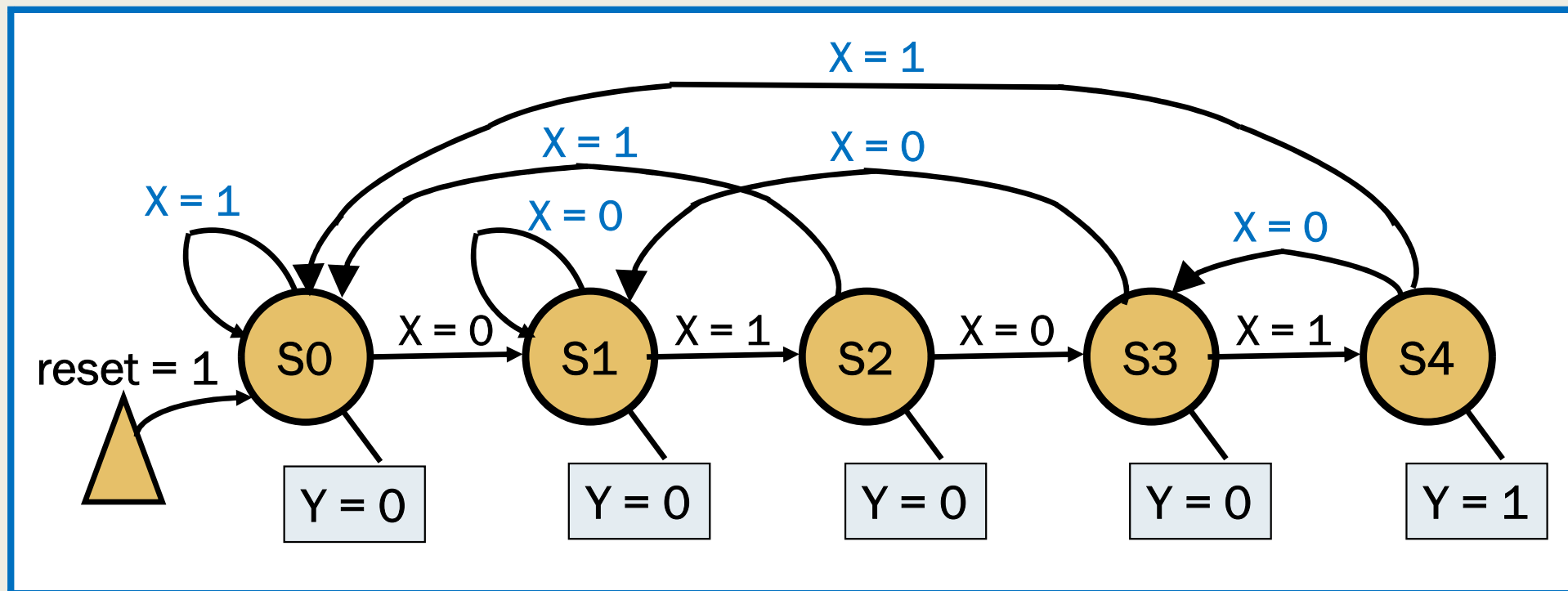
- Ο ανιχνευτής τύπου Moore έχει 5 καταστάσεις S_0, S_1, S_2, S_3 και S_4
- Το σήμα **RESET** αρχικοποιεί τον ανιχνευτή στην κατάσταση S_0
 - Η αρχικοποίηση του ανιχνευτή γίνεται σύγχρονα ή ασύγχρονα του CLK με κατάλληλη επιλογή των D Flip-Flop
- Η σχεδίαση του διαγράμματος γίνεται σε δύο φάσεις:
 - Στην πρώτη φάση σχεδιάζουμε τις τρέχουσες καταστάσεις (κύκλους) και τα βέλη με τη συνθήκη εισόδου ($X = 0$ ή $X = 1$) που προσδιορίζει τη μετάβαση από την τρέχουσα κατάσταση στην επόμενη κατάσταση, σύμφωνα με την ακολουθία των 4 bit (0101)



Ανιχνευτής ακολουθίας 4 διαδοχικών bit

■ Βήμα 2: Σχεδιάζουμε το διάγραμμα μεταβολής κατάστασης

- Ο ανιχνευτής τύπου Moore έχει 5 καταστάσεις $S0$, $S1$, $S2$, $S3$ και $S4$
- Η σχεδίαση του διαγράμματος γίνεται σε δύο φάσεις:
 - Στη δεύτερη φάση σχεδιάζουμε τα βέλη με την **συμπληρωματική συνθήκη εισόδου** ($X = 1$ ή $X = 0$, αντίστοιχα)



Επιλεγμένες ασκήσεις

- Άσκηση 3.25

Το σαλιγκάρι-ρομπότ έχει μία κόρη που διαθέτει για εγκέφαλο μια μηχανή FSM που στηρίζεται σε μηχανή τύπου Moore

- Το σαλιγκάρι-κόρη χαμογελάει όταν έρπει πάνω από την ακολουθία **1101** ή την ακολουθία **1110**.

- Σχεδιάστε το διάγραμμα μεταβολής κατάστασης γι' αυτό το χαρούμενο σαλιγκάρι

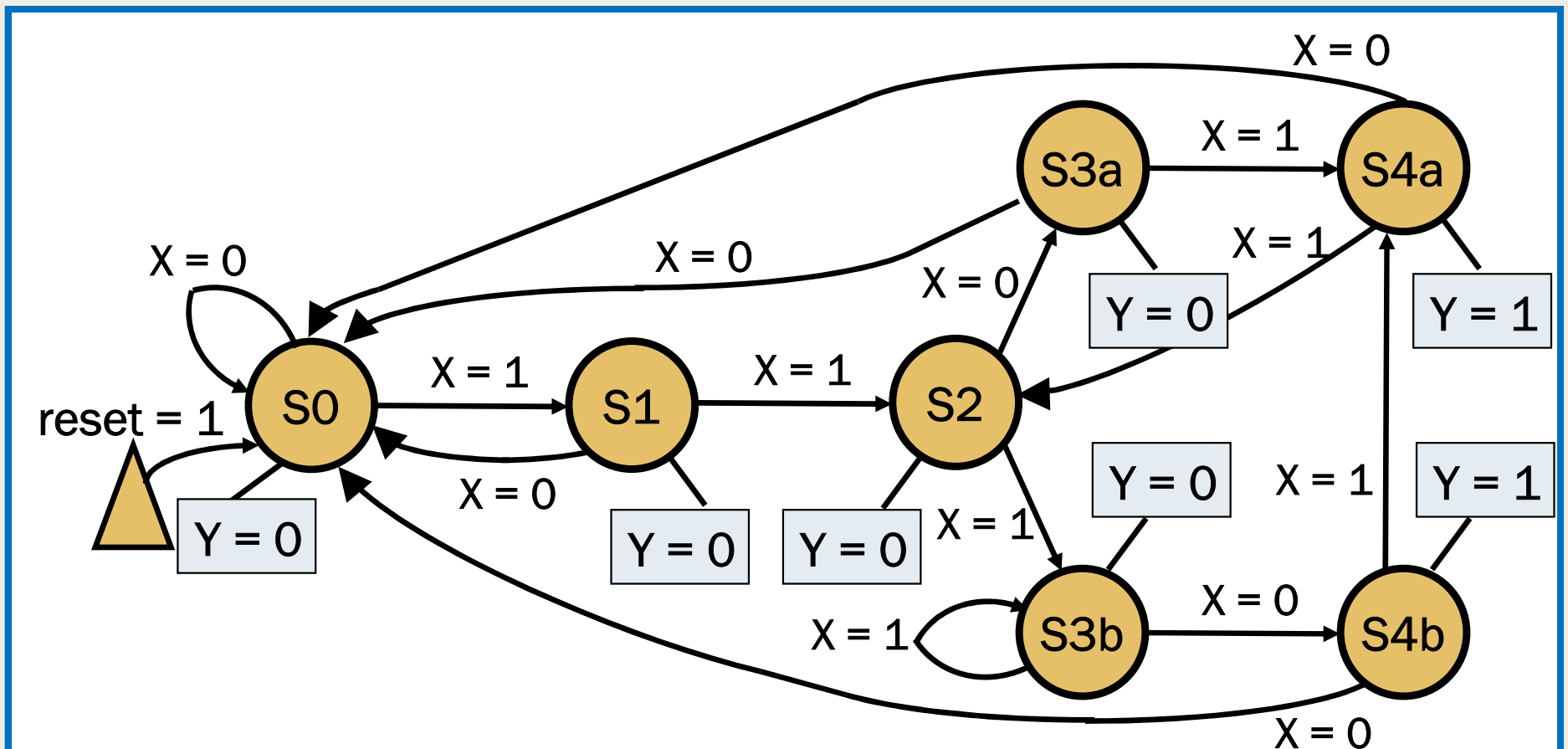
Άσκηση 3.25: Λύση (FSM τύπου Moore)

- **Βήμα 1: Προσδιορίζουμε τις εισόδους, τις εξόδους, και τις καταστάσεις**
 - Ο ανιχνευτής έχει μία σειριακή είσοδο X και μία έξοδο Y
 - $Y = 1$, όταν τα 4 τελευταία διαδοχικά bit της εισόδου X είναι **1101** ή **1110**
 - Καταστάσεις μηχανής **τύπου Moore**:
 - $S0$ = αρχική κατάσταση, δεν έχει ανιχνευθεί κανένα ψηφίο, $Y = 0$
 - $S1$ = έχει ανιχνευθεί στην είσοδο X ένα bit 1, $Y = 0$
 - $S2$ = έχουν ανιχνευθεί στην είσοδο X δύο διαδοχικά bit 11, $Y = 0$
 - $S3a$ = έχουν ανιχνευθεί στην είσοδο X τρία διαδοχικά bit 110, $Y = 0$
 - $S3b$ = έχουν ανιχνευθεί στην είσοδο X τρία διαδοχικά bit 111, $Y = 0$
 - $S4a$ = έχουν ανιχνευθεί στην είσοδο X τέσσερα διαδοχικά bit 1101, $Y = 1$
 - $S4b$ = έχουν ανιχνευθεί στην είσοδο X τέσσερα διαδοχικά bit 1110, $Y = 1$

Άσκηση 3.25: Λύση (FSM τύπου Moore)

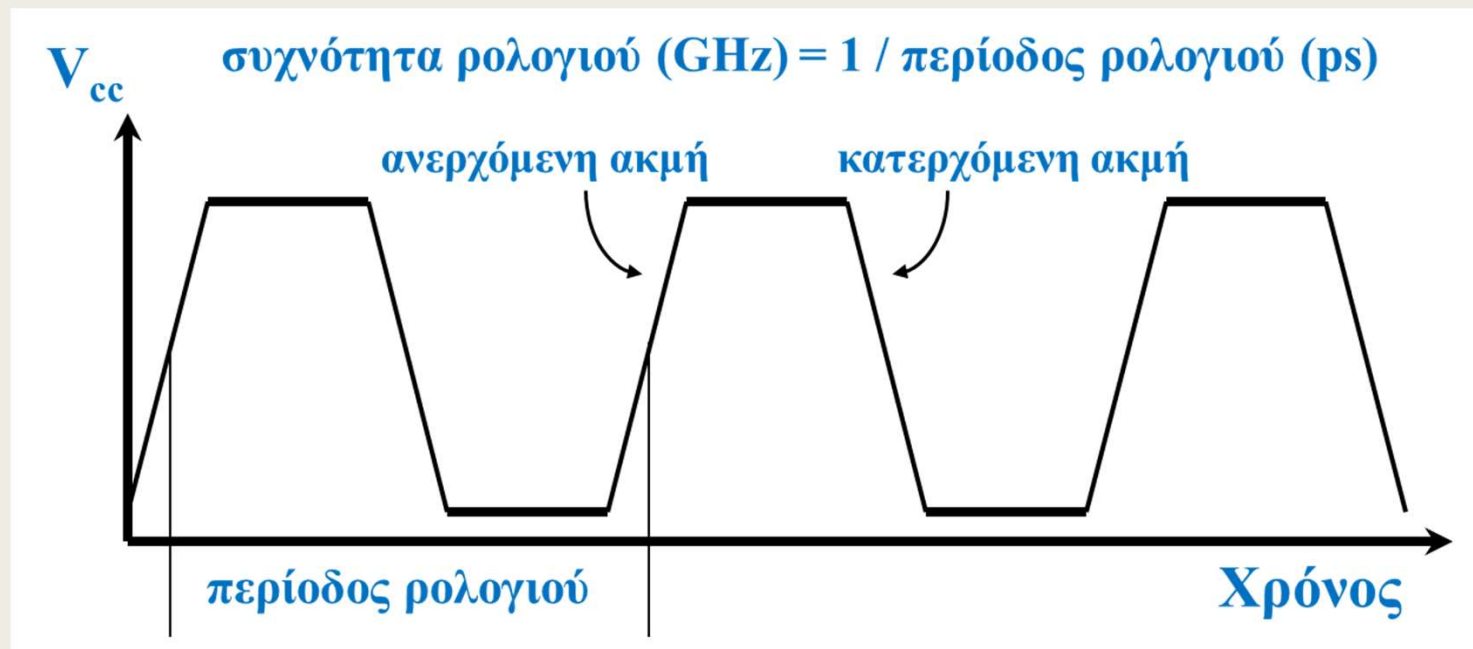
■ Βήμα 2: Σχεδιάζουμε το διάγραμμα μεταβολής κατάστασης

- Ο ανιχνευτής τύπου Moore έχει **7 καταστάσεις** **S0, S1, S2, S3a, S3b, S4a και S4b**
- Το σήμα **RESET** αρχικοποιεί τον ανιχνευτή στην κατάσταση **S0**
 - Η αρχικοποίηση του ανιχνευτή γίνεται σύγχρονα ή ασύγχρονα του CLK με κατάλληλη επιλογή των D Flip-Flop



Χρονισμός ακολουθιακής λογικής

- Το **σήμα του ρολογιού CLK** είναι ένας **τετραγωνικός παλμός**
- Ένα D Flip-Flop αντιγράφει την είσοδο D στην έξοδο Q **κατά την ανερχόμενη ακμή του CLK**
- Η συγκεκριμένη διαδικασία ονομάζεται **δειγματοληψία** (sampling) του D με βάση την ανερχόμενη ακμή του CLK
- Για να επιτευχθεί η δειγματοληψία **η είσοδος D** πρέπει να είναι **σταθερή πριν και μετά** την ανερχόμενη ακμή του CLK (δυναμική πειθαρχία)
 - Αλλιώς μπορεί να εμφανισθεί **μετασταθερότητα**

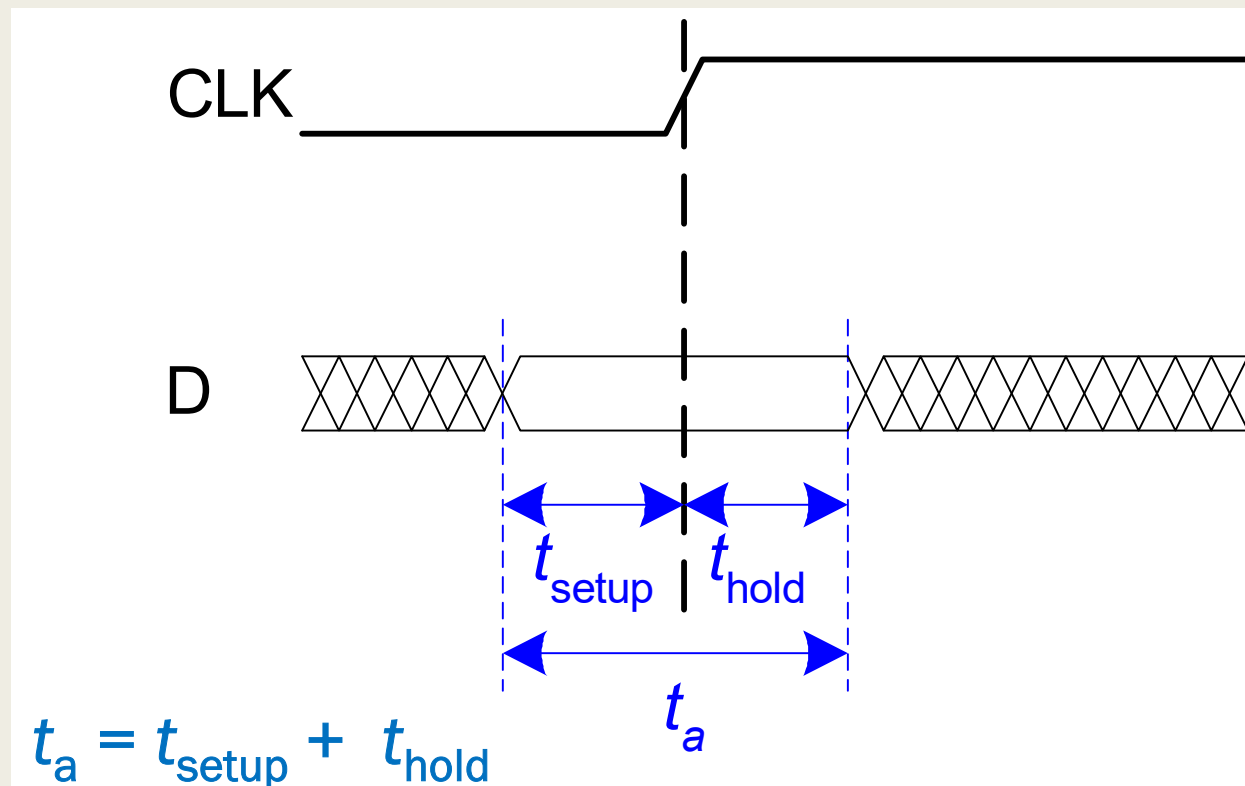


Χρονικοί περιορισμοί στη δειγματοληψία

■ Χρονικοί περιορισμοί στην είσοδο D του D Flip-Flop

που πρέπει να ικανοποιούνται κατά τη δειγματοληψία:

- **Χρόνος σταθεροποίησης (set-up time, t_{setup})** = ο χρόνος που η είσοδος D πρέπει να παραμένει σταθερή **πριν** από την ανερχόμενη ακμή του CLK
- **Χρόνος διατήρησης (hold time, t_{hold})** = ο χρόνος που η είσοδος D πρέπει να παραμένει σταθερή **μετά** από την ανερχόμενη ακμή του CLK
- **Χρόνος ανοίγματος (aperture time, t_a)** = ο χρόνος που η είσοδος D πρέπει να παραμένει σταθερή **πριν και μετά** από την ανερχόμενη ακμή του CLK

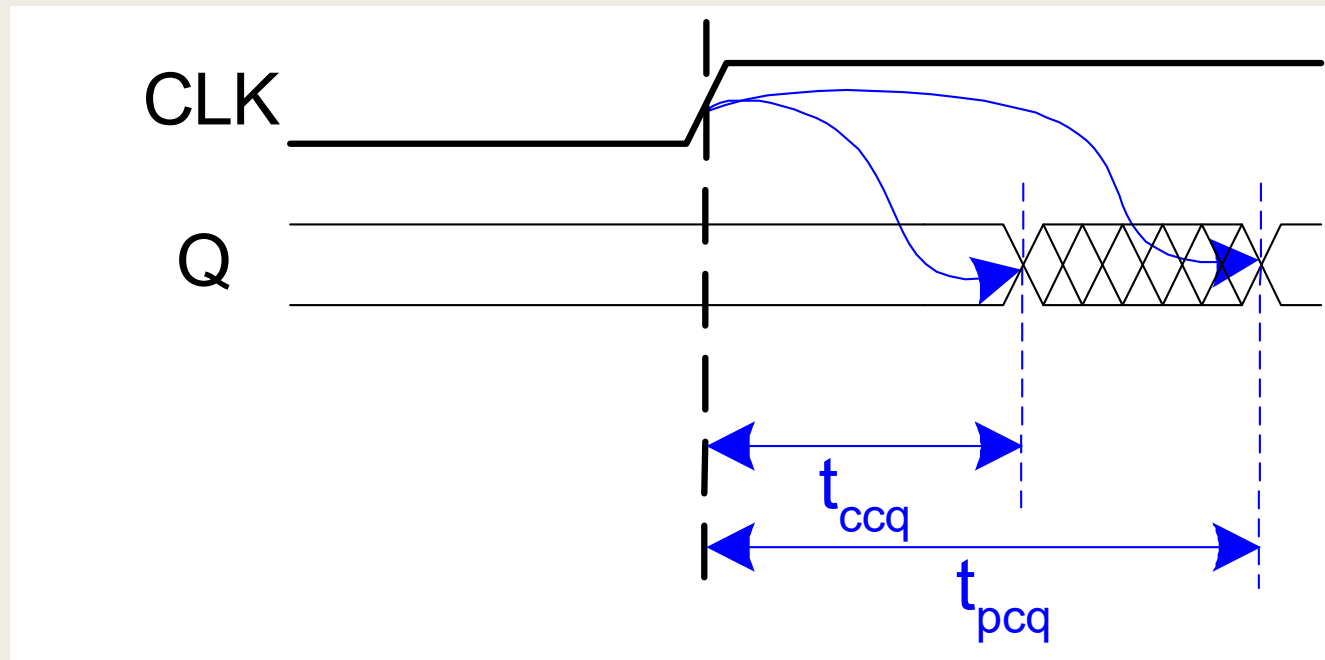


Χρονικοί περιορισμοί στη δειγματοληψία

■ Χρονικοί περιορισμοί στην έξοδο Q του D Flip-Flop

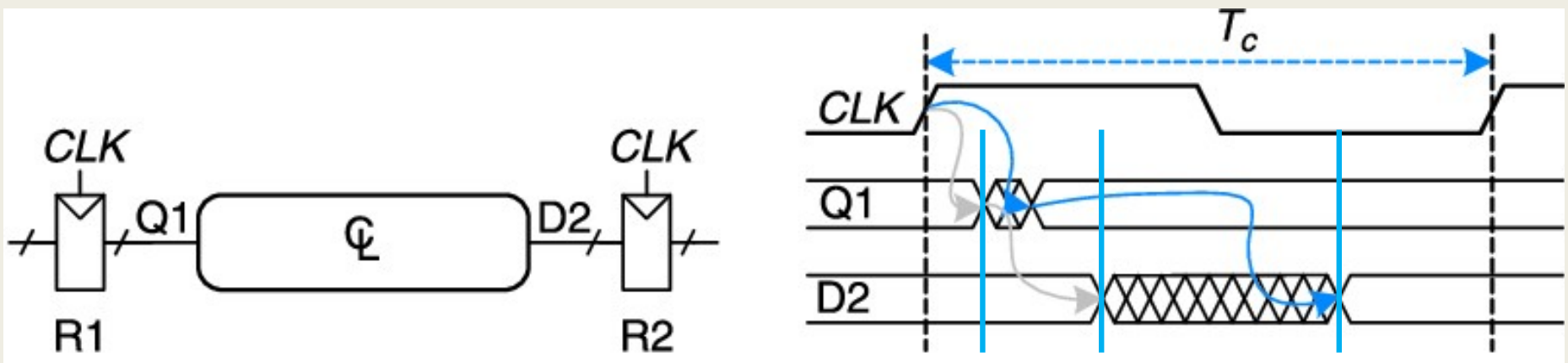
που πρέπει να ικανοποιούνται κατά τη δειγματοληψία:

- *Καθυστέρηση διάδοσης (propagation clock to Q delay, t_{pcq})* = ο μέγιστος χρόνος εντός του οποίου η έξοδος Q πρέπει να έχει οπωσδήποτε σταθεροποιηθεί μετά από την ανερχόμενη ακμή του CLK
- *Καθυστέρηση μόλυνσης (contamination clock to Q delay, t_{ccq})* = ο ελάχιστος χρόνος που πρέπει να παρέλθει ώστε η έξοδος Q να αρχίσει να μεταβάλλεται μετά από την ανερχόμενη ακμή του CLK



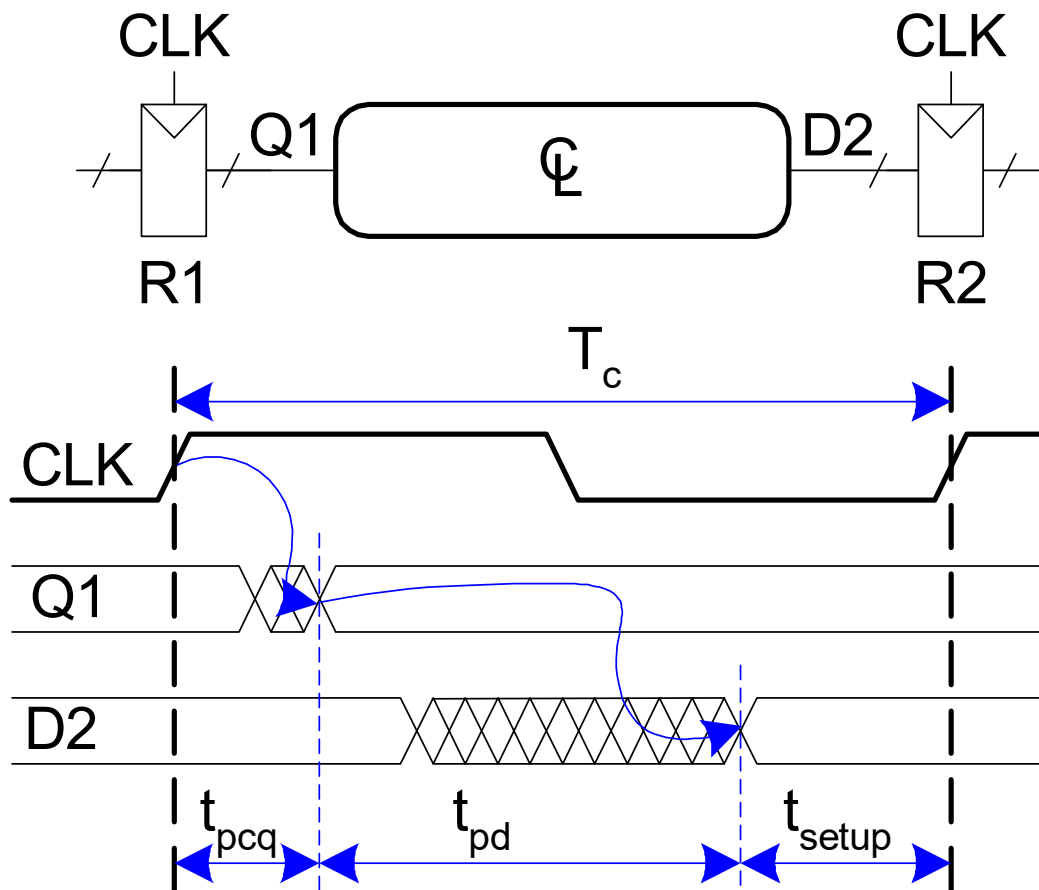
Σχεδίαση στο επίπεδο της μεταφοράς καταχωρητών

- Οι **μέγιστες** και οι **ελάχιστες** καθυστερήσεις ανάμεσα στους καταχωρητές εξαρτώνται από τις καθυστερήσεις στους καταχωρητές και την συνδυαστική λογική (καθυστέρηση διάδοσης t_{pd} και μόλυνσης t_{cd})
- Ένα σήμα κατά τη διάδοσή του μέσα από τη συνδυαστική λογική μπορεί να εμφανίσει μεταβατικό παλμό και να ταλαντώνεται για κάποιο φραγμένο χρονικό διάστημα
- Σύμφωνα με τη αρχή της δυναμικής πειθαρχίας το σήμα θα πρέπει να έχει σταθεροποιηθεί πριν την επόμενη ανερχόμενη ακμή του CLK
 - Υπολογίζεται η **ελάχιστη περίοδος του CLK (T_c)** που εξασφαλίζει την κανονική λειτουργία του συστήματος
 - Αντίστοιχα, ορίζεται η **μέγιστη συχνότητα του CLK ($F_c = 1/T_c$)**



Ελάχιστη περίοδος του CLK T_c

- Υπολογίζεται από τη **μέγιστη καθυστέρηση διάδοσης** μεταξύ των καταχωρητών R1 και του R2 διαμέσου της συνδυαστικής λογικής
 - Η είσοδος του R2 πρέπει να έχει σταθεροποιηθεί τουλάχιστον κατά τον **χρόνο σταθεροποίησης t_{setup}** πριν την ανερχόμενη ακμή του CLK
 - Εξαρτάται από την **καθυστέρηση διάδοσης t_{pd}** της συνδυαστικής λογικής
 - πλήθος διαδοχικών πυλών που μπορούν να υπάρχουν στην κρίσιμη διαδρομή



$$T_c \geq t_{pcq} + t_{pd} + t_{setup}$$
$$t_{pd} \leq T_c - (t_{pcq} + t_{setup})$$

t_{pcq} : καθυστέρηση διάδοσης του καταχωρητή

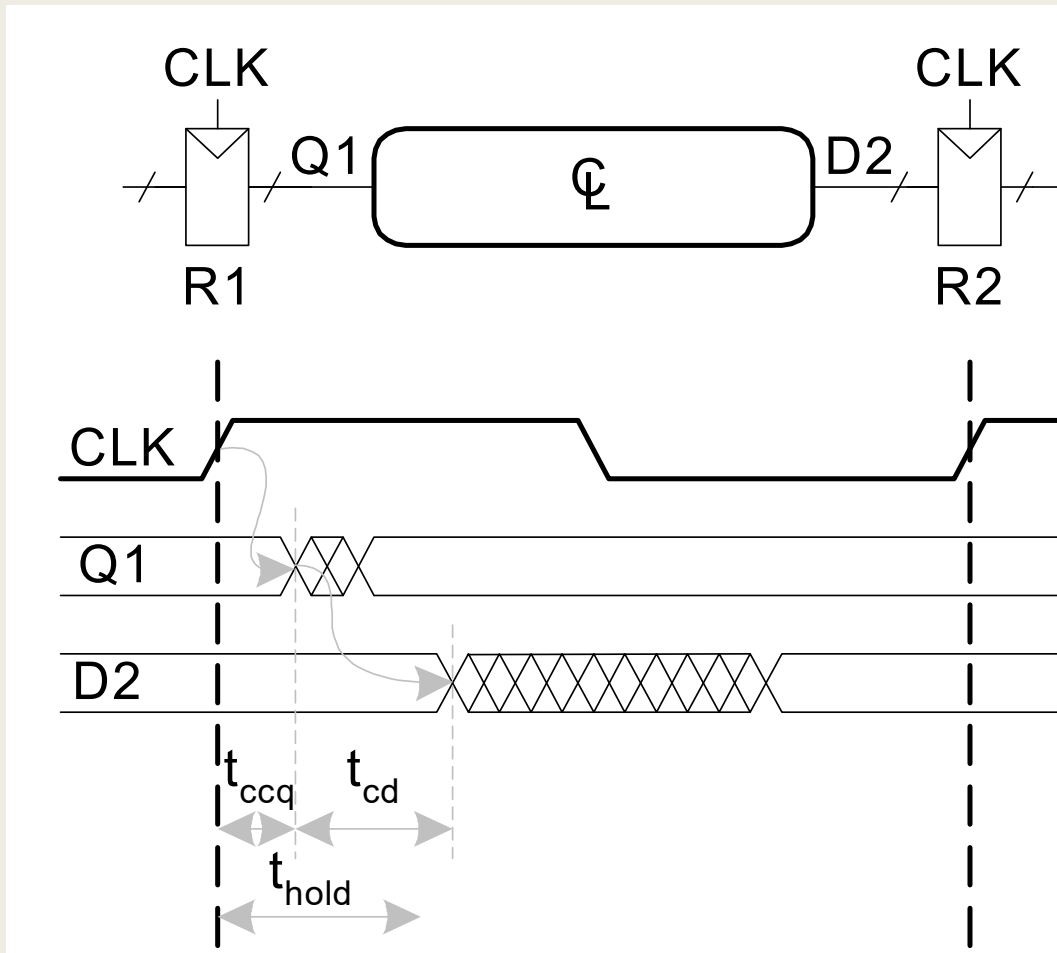
t_{pd} : καθυστέρηση διάδοσης της συνδυαστικής λογικής

t_{setup} : χρόνος σταθεροποίησης

Επιβάρυνση δημιουργίας ακολουθίας (sequencing overhead): $t_{pcq} + t_{setup}$

Ελάχιστη καθυστέρηση μόλυνσης t_{cd}

- Εξαρτάται από την **ελάχιστη καθυστέρηση διάδοσης** μεταξύ των καταχωρητών R1 και του R2 διαμέσου της συνδυαστικής λογικής
 - Η είσοδος του R2 πρέπει να παραμένει σταθερή τουλάχιστον κατά τον **χρόνο διατήρησης t_{hold}** μετά την ανερχόμενη ακμή του CLK
 - Εξαρτάται από την **καθυστέρηση μόλυνσης t_{cd}** της συνδυαστικής λογικής
 - πλήθος διαδοχικών πυλών που μπορούν να υπάρχουν στη σύντομη διαδρομή



$$t_{hold} < t_{ccq} + t_{cd}$$

$$t_{cd} > t_{hold} - t_{ccq}$$

t_{ccq} : καθυστέρηση μόλυνσης
του καταχωρητή

t_{cd} : καθυστέρηση μόλυνσης
της συνδυαστικής λογικής

t_{hold} : χρόνος διατήρησης

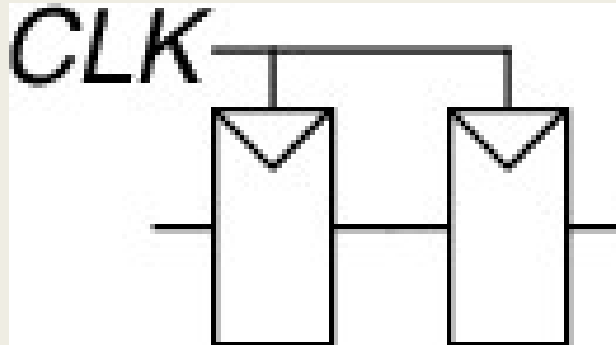
Εξετάζουμε τα κυκλώματα
για πιθανή παραβίαση του
χρόνου διατήρησης

Τοποθέτηση D Flip-Flop στη σειρά

- Για να τοποθετηθούν δύο D Flip-Flops στη σειρά θα πρέπει να ισχύουν οι συνθήκες:

$$T_c \geq t_{pcq} + t_{\text{setup}}$$

$$t_{\text{hold}} < t_{ccq}$$

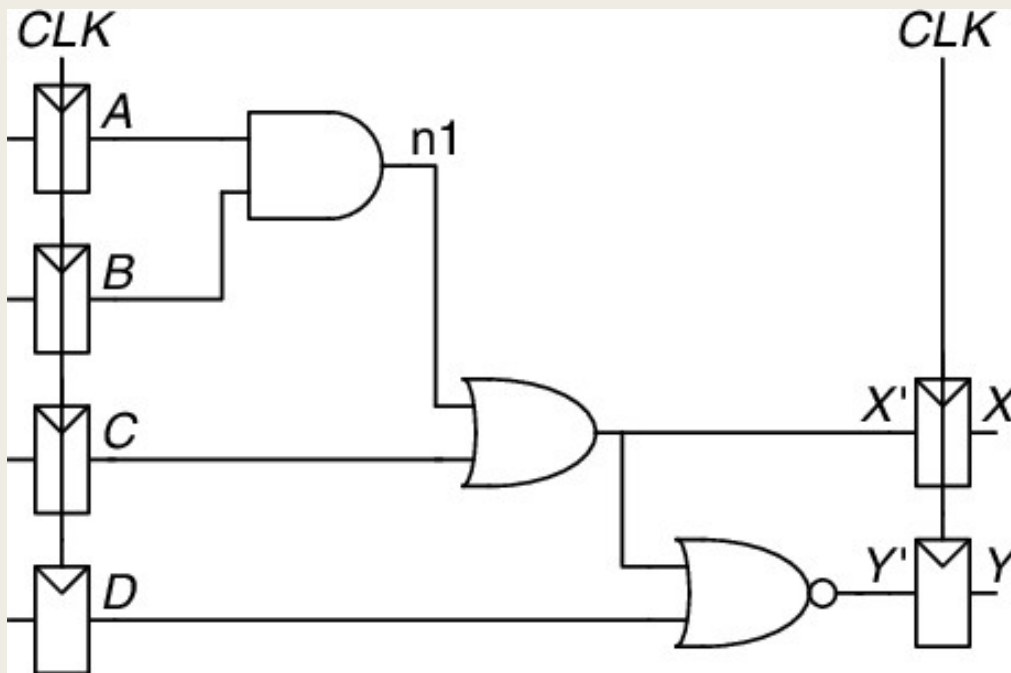


$$T_c \geq t_{pcq} + \cancel{t_{pd}} + t_{\text{setup}}$$
$$t_{\text{hold}} < \cancel{t_{ccq}} + \cancel{t_{cd}}$$

- Συνήθως, στην πράξη $t_{\text{hold}} = 0$
- Οι **περιορισμοί λόγω του χρόνου διατήρησης** είναι σημαντικοί
 - Αν παραβιάζονται, η μόνη λύση είναι να αυξηθεί η καθυστέρηση μόλυνσης για τη διέλευση μέσα από τα κυκλώματα λογικής, κάτι που απαιτεί την **εκ νέου σχεδίαση του κυκλώματος**
 - Σε αντίθεση με τους περιορισμούς λόγω του χρόνου σταθεροποίησης, δεν μπορούμε να διορθώσουμε τους περιορισμούς λόγω του χρόνου διατήρησης ρυθμίζοντας απλώς την περίοδο του ρολογιού.

Ανάλυση Χρονισμού*

- Δεδομένου του σχηματικού διαγράμματος ενός κυκλώματος και των χαρακτηριστικών χρονισμού συγκεκριμένης τεχνολογίας υπολογίζουμε:
 - τη **μέγιστη συχνότητα λειτουργίας** ($F_c = 1/T_c$)



Συνδυαστική λογική:

$$t_{pd} = 3 \times 40 \text{ ps} = 120 \text{ ps}$$

$$t_{cd} = 1 \times 25 \text{ ps} = 25 \text{ ps}$$

Χαρακτηριστικά χρονισμού

Καταχωρητή:

$$t_{ccq} = 30 \text{ ps}$$

$$t_{pcq} = 80 \text{ ps}$$

$$t_{\text{setup}} = 50 \text{ ps}$$

$$t_{\text{hold}} = 60 \text{ ps}$$

Πύλης

$$t_{pd} = 40 \text{ ps}$$

$$t_{cd} = 25 \text{ ps}$$

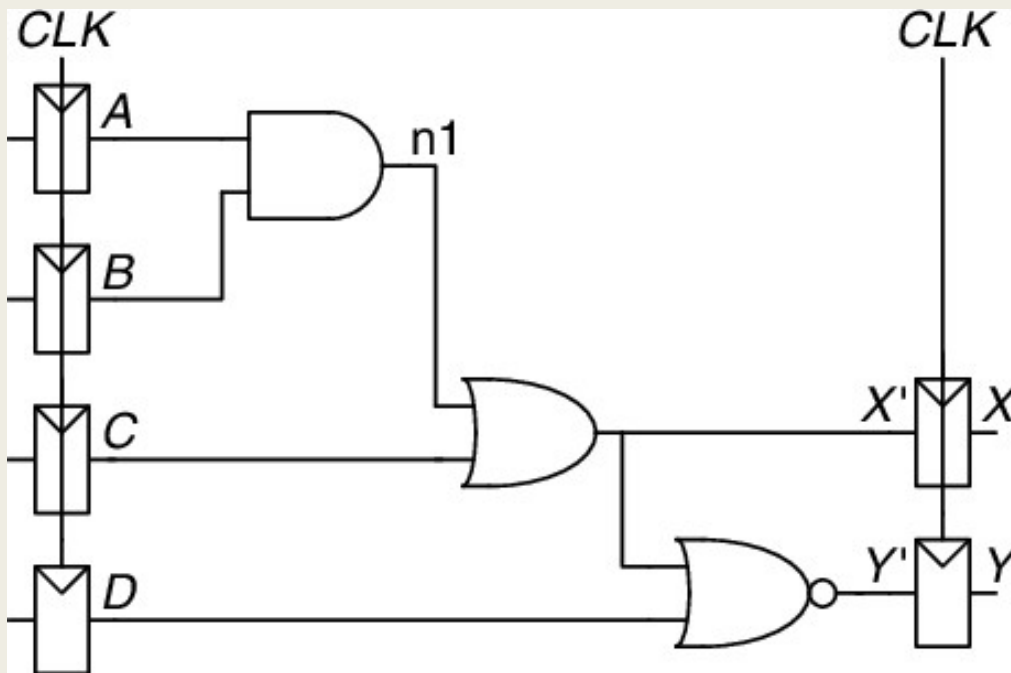
$$T_c \geq t_{pcq} + t_{pd} + t_{\text{setup}} = (80 + 120 + 50) \text{ ps} = 250 \text{ ps}$$

$$F_c = 4 \text{ GHz}$$

*Παράδειγμα 3.10

Ανάλυση Χρονισμού*

- Δεδομένου του σχηματικού διαγράμματος ενός κυκλώματος και των χαρακτηριστικών χρονισμού συγκεκριμένης τεχνολογίας υπολογίζουμε:
 - την *πιθανή παραβίαση του χρόνου διατήρησης*



Συνδυαστική λογική:

$$t_{pd} = 3 \times 40 \text{ ps} = 120 \text{ ps}$$

$$t_{cd} = 1 \times 25 \text{ ps} = 25 \text{ ps}$$

Χαρακτηριστικά χρονισμού

Καταχωρητή:

$$t_{ccq} = 30 \text{ ps}$$

$$t_{pcq} = 80 \text{ ps}$$

$$t_{\text{setup}} = 50 \text{ ps}$$

$$t_{\text{hold}} = 60 \text{ ps}$$

Πύλης

$$t_{pd} = 40 \text{ ps}$$

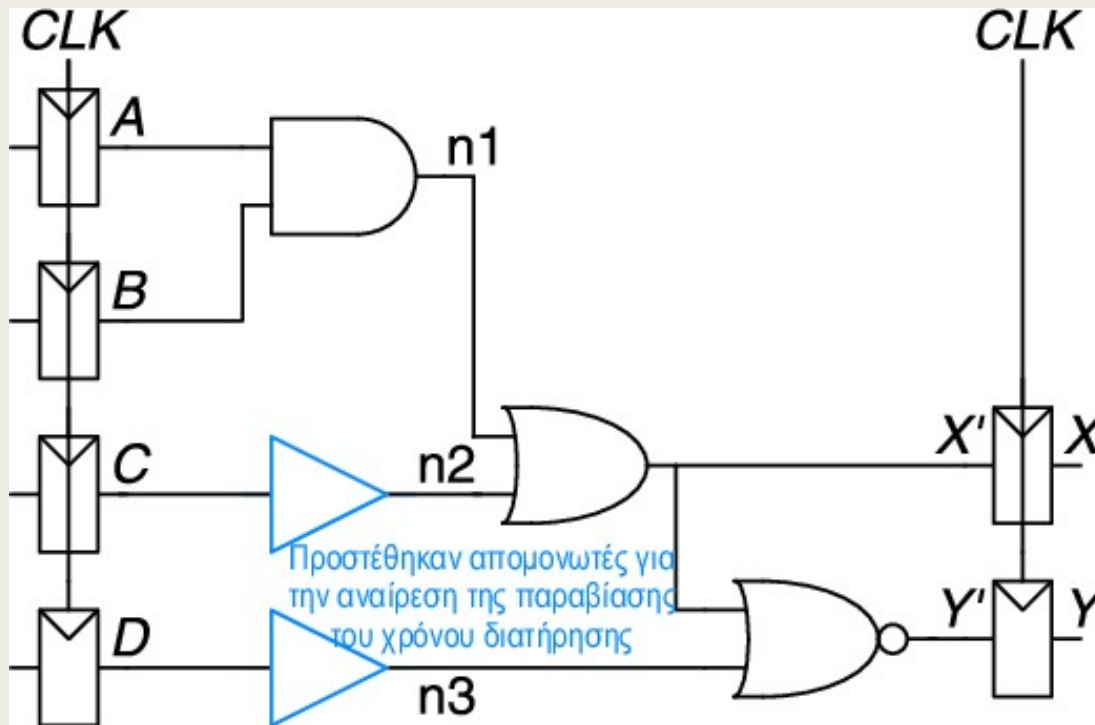
$$t_{cd} = 25 \text{ ps}$$

$$t_{\text{hold}} < t_{ccq} + t_{cd} \Rightarrow 60 \text{ ps} < (30 + 25) \text{ ps} \text{ (Δεν ισχύει)}$$

*Παράδειγμα 3.10

Διόρθωση παραβιάσεων του χρόνου διατήρησης*

- Θα πρέπει να επανασχεδιάσουμε το κύκλωμα με την προσθήκη απομονωτών στις σύντομες διαδρομές, ώστε να μην παραβιάζεται ο χρόνος διατήρησης



Χαρακτηριστικά χρονισμού

Καταχωρητή:

$$t_{ccq} = 30 \text{ ps}$$

$$t_{pcq} = 80 \text{ ps}$$

$$t_{\text{setup}} = 50 \text{ ps}$$

$$t_{\text{hold}} = 60 \text{ ps}$$

Πύλης

$$t_{pd} = 40 \text{ ps}$$

$$t_{cd} = 25 \text{ ps}$$

Συνδυαστική λογική:

$$t_{pd} = 3 \times 40 \text{ ps} = 120 \text{ ps}$$

$$t_{cd} = 2 \times 25 \text{ ps} = 50 \text{ ps}$$

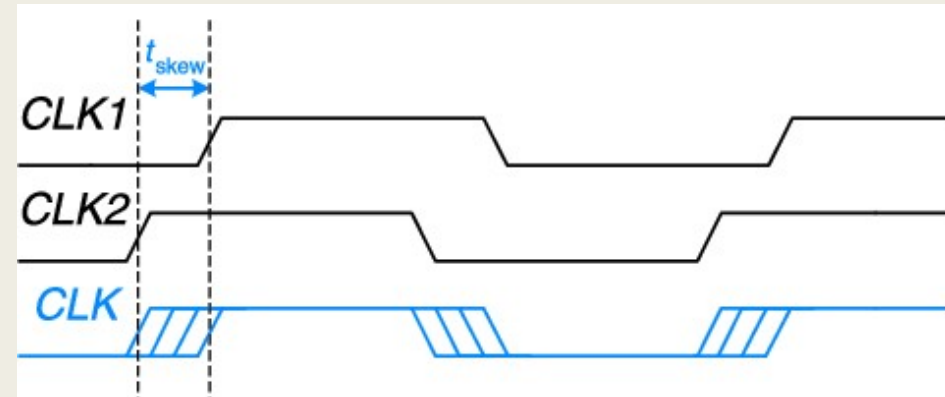
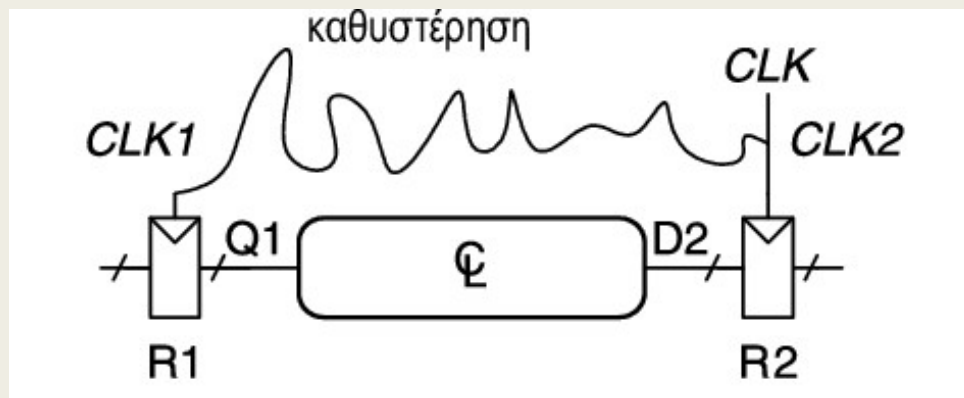
$$t_{\text{hold}} < t_{ccq} + t_{cd} \Rightarrow 60 \text{ ps} < (30 + 50) \text{ ps} \text{ (OK!)}$$

*Παράδειγμα 3.11

Προσοχή! Κάποιοι μικροεπεξεργαστές υψηλών επιδόσεων, χρησιμοποιούν pulsed latches με μικρή καθυστέρηση από το ρολόι έως την έξοδο Q , αλλά μεγάλο χρόνο διατήρησης

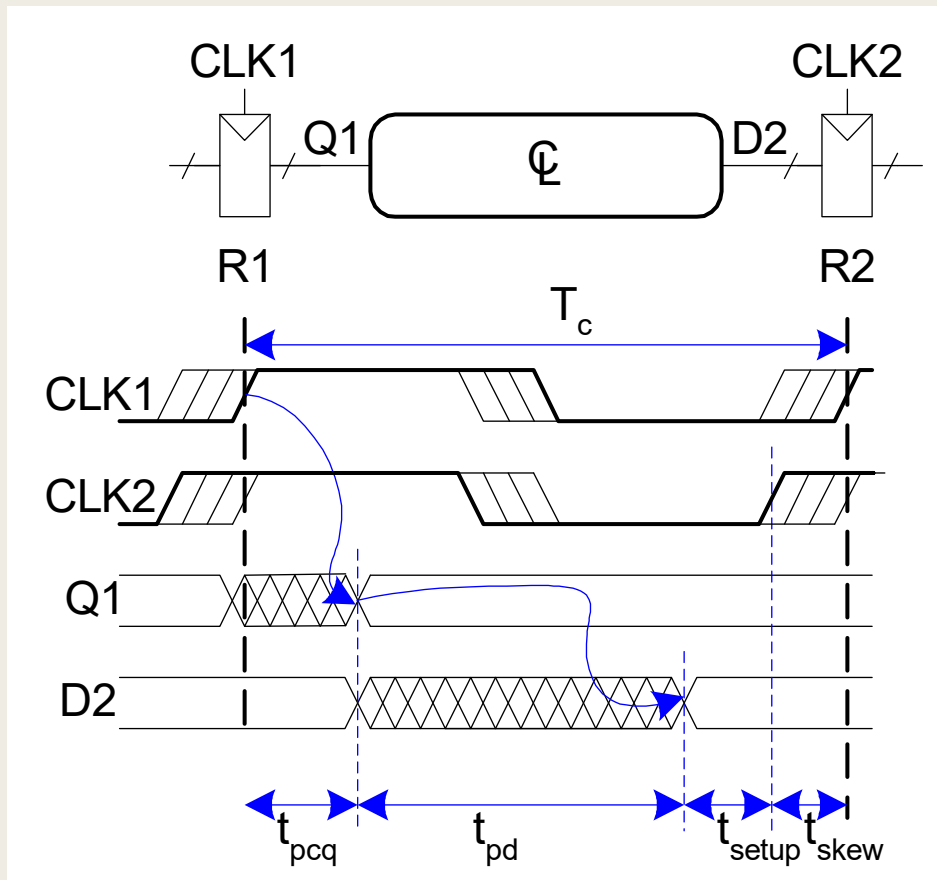
Απόκλιση ρολογιού t_{skew}

- Το σήμα του ρολογιού CLK δεν φθάνει συνήθως στους καταχωρητές ακριβώς την ίδια χρονική στιγμή εξ αιτίας διαφορετικών καθυστερήσεων διάδοσης που οφείλονται:
 - Στην ύπαρξη διαφορετικού μήκους διαδρομών από την πηγή του σήματος CLK μέχρι την είσοδο του ρολογιού στον εκάστοτε καταχωρητή
 - Στην επίδραση του θορύβου
 - Στην προσθήκη πύλης σε κάποια από τις διαδρομές του ρολογιού
- Απόκλιση ρολογιού ονομάζεται η αυξομείωση στην εμφάνιση των ανερχόμενων ακμών του σήματος CLK
 - Στην πράξη λαμβάνουμε υπόψη τη χειρότερη περίπτωση του χρόνου απόκλισης ρολογιού (clock skew) t_{skew}



Νέα ελάχιστη περίοδος του CLK T_c

- Στην χειρότερη περίπτωση το CLK2 έρχεται **νωρίτερα** από το CLK1 κατά t_{skew}
- Ο χρόνος σταθεροποίησης t_{setup} επιβαρύνεται με τον χρόνο απόκλισης t_{skew}



$$T_c \geq t_{pcq} + t_{pd} + (t_{setup} + t_{skew})$$
$$t_{pd} \leq T_c - (t_{pcq} + t_{setup} + t_{skew})$$

t_{pcq} : καθυστέρηση διάδοσης του καταχωρητή

t_{pd} : καθυστέρηση διάδοσης της συνδυαστικής λογικής

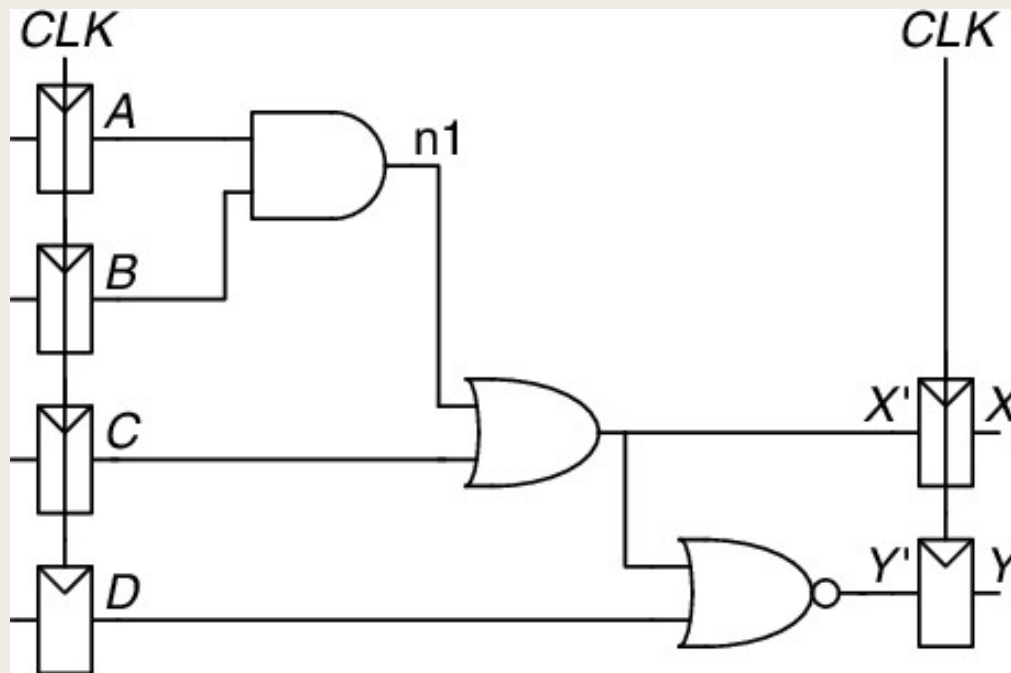
t_{setup} : χρόνος σταθεροποίησης

t_{skew} : χρόνος απόκλισης

Επιβάρυνση δημιουργίας ακολουθίας (sequencing overhead): $t_{pcq} + t_{setup} + t_{skew}$

Ανάλυση Χρονισμού με t_{skew}^*

- Δεδομένου του σχηματικού διαγράμματος ενός κυκλώματος και των χαρακτηριστικών χρονισμού συγκεκριμένης τεχνολογίας υπολογίζουμε:
 - τη **μέγιστη συχνότητα λειτουργίας** ($F_c = 1/T_c$) λαμβάνοντας υπόψη και τη χειρότερη περίπτωση του **χρόνου απόκλισης**



Χαρακτηριστικά χρονισμού

Καταχωρητή:

$$t_{ccq} = 30 \text{ ps}$$

$$t_{pcq} = 80 \text{ ps}$$

$$t_{setup} = 50 \text{ ps}$$

$$t_{hold} = 60 \text{ ps}$$

$$t_{skew} = 50 \text{ ps}$$

Πύλης

$$t_{pd} = 40 \text{ ps}$$

$$t_{cd} = 25 \text{ ps}$$

Συνδυαστική λογική:

$$t_{pd} = 3 \times 40 \text{ ps} = 120 \text{ ps}$$

$$t_{cd} = 1 \times 25 \text{ ps} = 25 \text{ ps}$$

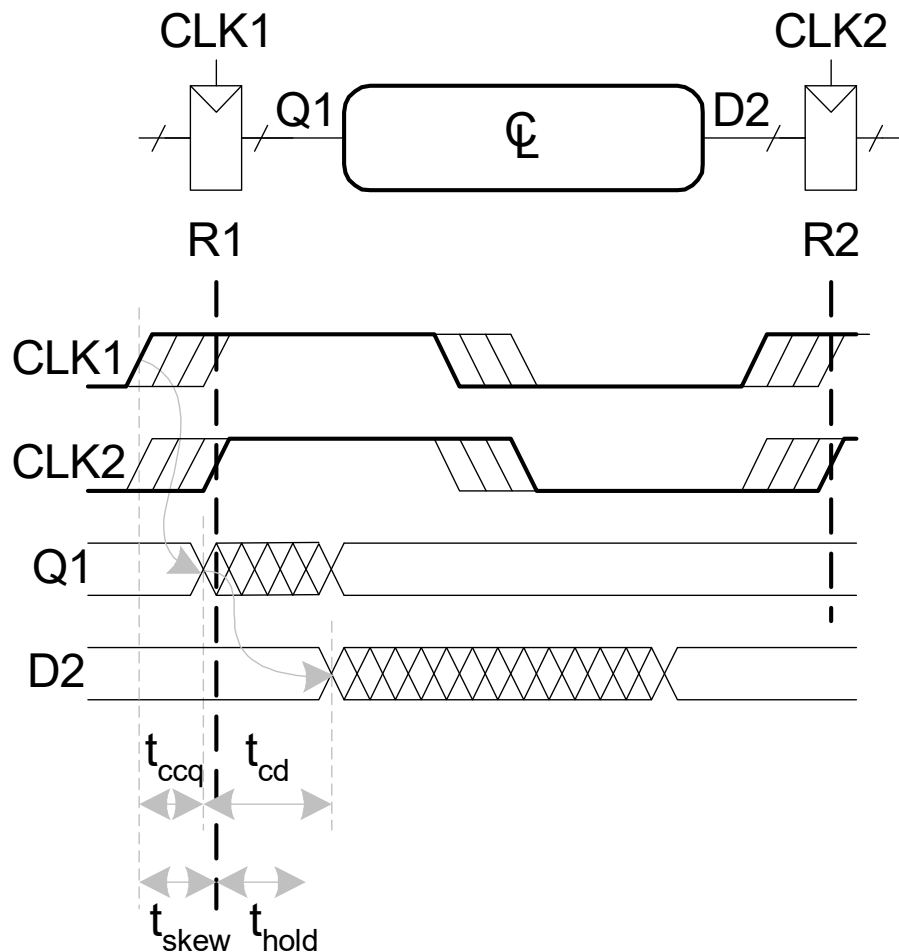
$$T_c \geq t_{pcq} + t_{pd} + t_{setup} + t_{skew} = (80 + 120 + 50 + 50) \text{ ps} = 300 \text{ ps}$$

$$F_c = 3,33 \text{ GHz αντί } 4 \text{ GHz}$$

*Παράδειγμα 3.12

Νέα ελάχιστη καθυστέρηση μόλυνσης t_{cd}

- Στην χειρότερη περίπτωση το CLK1 έρχεται **νωρίτερα** από το CLK2 κατά t_{skew}
- Ο χρόνος διατήρησης t_{hold} επιβαρύνεται με τον **χρόνο απόκλισης** t_{skew}



$$t_{hold} + t_{skew} < t_{ccq} + t_{cd}$$
$$t_{cd} > (t_{hold} + t_{skew}) - t_{ccq}$$

t_{ccq} : καθυστέρηση μόλυνσης του καταχωρητή

t_{cd} : καθυστέρηση μόλυνσης της συνδυαστικής λογικής

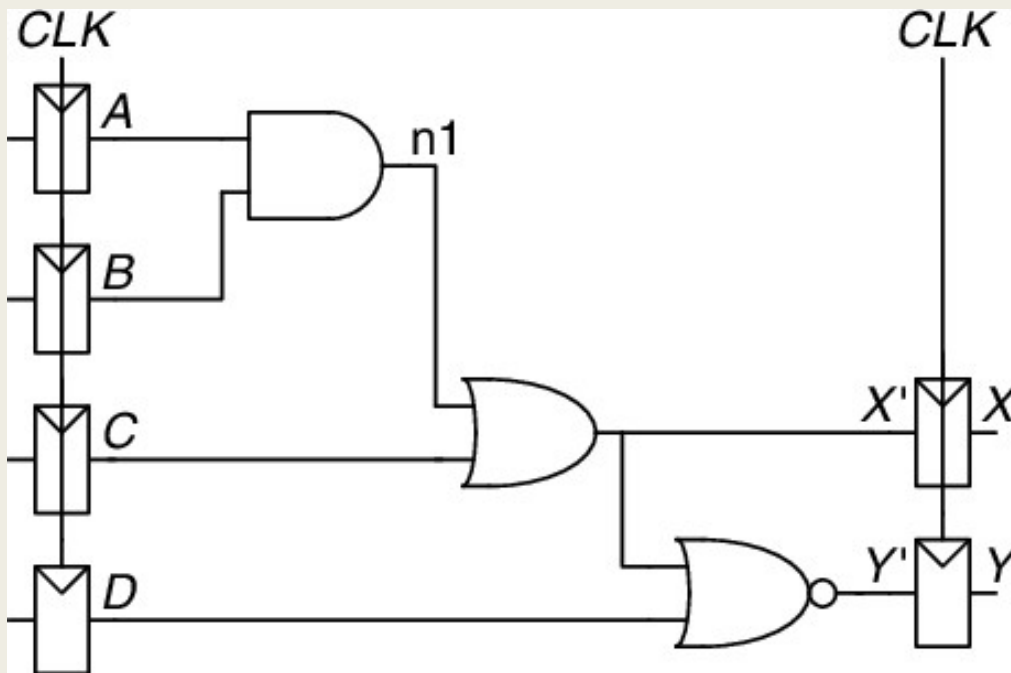
t_{hold} : χρόνος διατήρησης

t_{skew} : χρόνος απόκλισης

Εξετάζουμε τα κυκλώματα για πιθανή παραβίαση του **χρόνου διατήρησης** με την επιβάρυνση του **χρόνου απόκλισης**

Ανάλυση Χρονισμού με t_{skew} *

- Δεδομένου του σχηματικού διαγράμματος ενός κυκλώματος και των χαρακτηριστικών χρονισμού συγκεκριμένης τεχνολογίας υπολογίζουμε:
 - την **πιθανή παραβίαση του χρόνου διατήρησης** λαμβάνοντας υπόψη και τη χειρότερη περίπτωση του **χρόνου απόκλισης**



Συνδυαστική λογική:

$$t_{pd} = 3 \times 40 \text{ ps} = 120 \text{ ps}$$

$$t_{cd} = 1 \times 25 \text{ ps} = 25 \text{ ps}$$

Χαρακτηριστικά χρονισμού

Καταχωρητή:

$$t_{ccq} = 30 \text{ ps}$$

$$t_{pcq} = 80 \text{ ps}$$

$$t_{setup} = 50 \text{ ps}$$

$$t_{hold} = 60 \text{ ps}$$

Πύλης

$$t_{pd} = 40 \text{ ps}$$

$$t_{cd} = 25 \text{ ps}$$

$$t_{skew} = 50 \text{ ps}$$

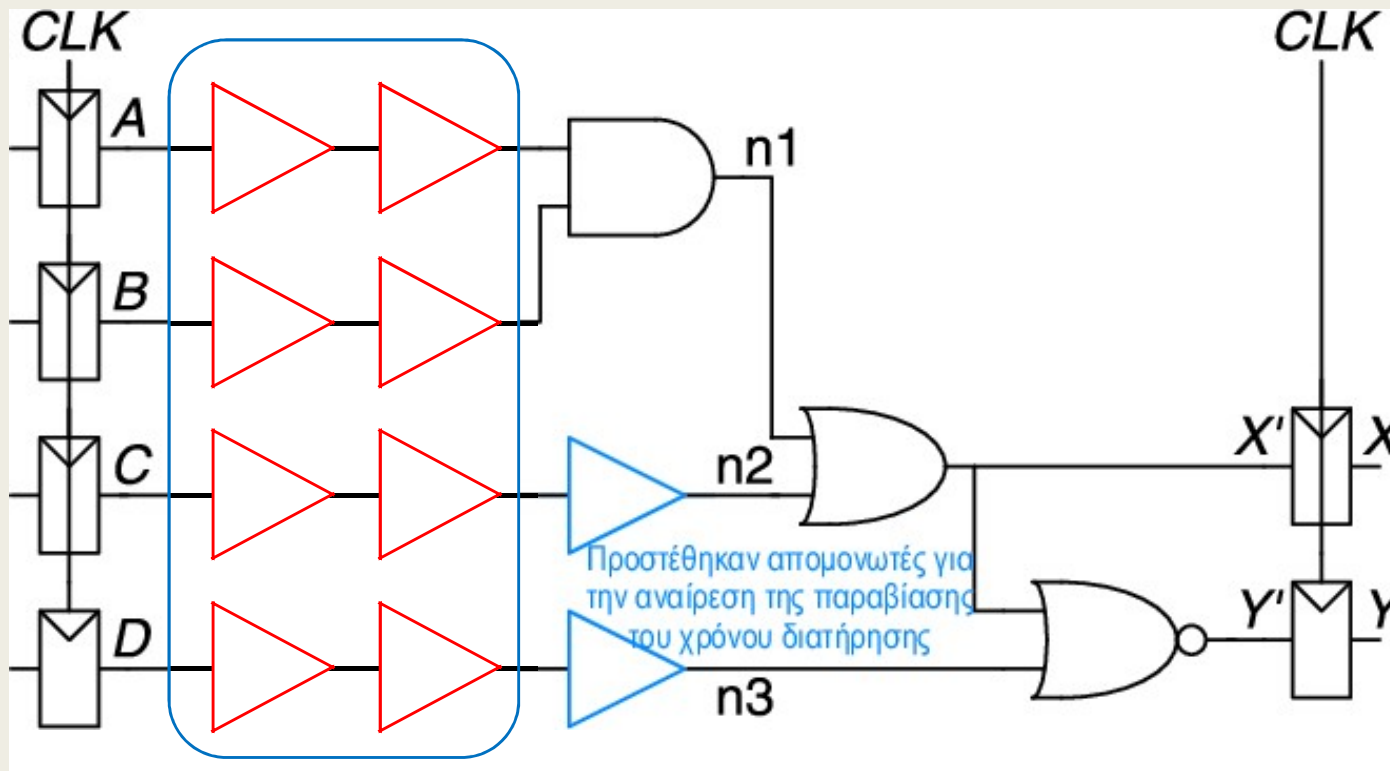
$$t_{hold} + t_{skew} < t_{ccq} + t_{cd} \Rightarrow (60+50) \text{ ps} < (30+25) \text{ ps}$$

(Δεν ισχύει και γίνεται χειρότερο, 110 ps αντί 60 ps)

*Παράδειγμα 3.12

Διόρθωση παραβιάσεων του χρόνου διατήρησης*

- Θα πρέπει να επανασχεδιάσουμε το κύκλωμα με την προσθήκη απομονωτών στις σύντομες διαδρομές, ώστε να μην παραβιάζεται ο χρόνος διατήρησης λαμβάνοντας υπόψη και τη χειρότερη περίπτωση του **χρόνου απόκλισης**
 - Απαιτούνται περισσότεροι απομονωτές (επιπλέον 2 σε κάθε διαδρομή), αλλά αυξάνει σημαντικά η ελάχιστη περίοδος του CLK T_c (από 300 ps σε 380 ps)



Χαρ/κά χρονισμού

Καταχωρητή:

$$t_{ccq} = 30 \text{ ps}$$

$$t_{pcq} = 80 \text{ ps}$$

$$t_{\text{setup}} = 50 \text{ ps}$$

$$t_{\text{hold}} = 60 \text{ ps}$$

$$t_{\text{skew}} = 50 \text{ ps}$$

Πύλης

$$t_{pd} = 40 \text{ ps}$$

$$t_{cd} = 25 \text{ ps}$$

Συνδυαστική λογική:

$$t_{pd} = 5 \times 40 \text{ ps} = 200 \text{ ps}$$

$$t_{cd} = 4 \times 25 \text{ ps} = 100 \text{ ps}$$

$$t_{\text{hold}} + t_{\text{skew}} < t_{ccq} + t_{cd} \Rightarrow (60 + 50) \text{ ps} < (30 + 100) \text{ ps}$$

(OK!)

*Παράδειγμα 3.13

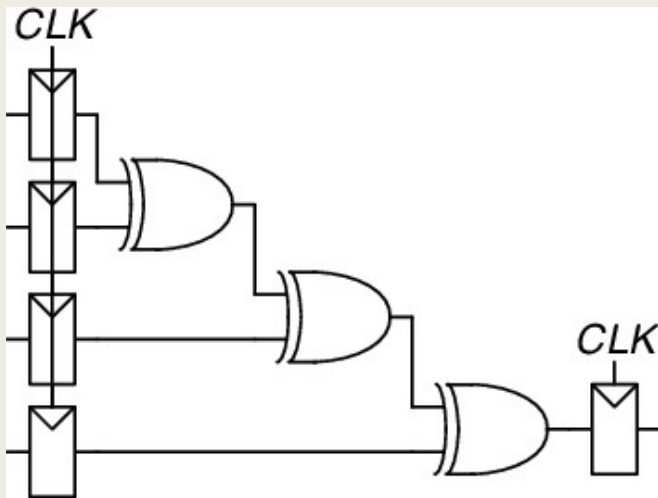
Επιλεγμένες ασκήσεις

■ Άσκηση 3.33

Ο Γιάννης Μπιτάκης έχει σχεδιάσει το κύκλωμα της Εικόνας 3.74 ώστε να υπολογίζει την τιμή μιας καταχωρισμένης (registered) συνάρτησης XOR με 4 εισόδους.

- (a) Ποια είναι η μέγιστη συχνότητα λειτουργίας του κυκλώματος με ρολόι χωρίς απόκλιση;
- (b) Πόση απόκλιση του ρολογιού μπορεί να ανέχεται το κύκλωμα αν πρέπει να λειτουργεί στα 2 GHz;
- (c) Πόση απόκλιση του ρολογιού μπορεί να ανέχεται το κύκλωμα προτού αντιμετωπίσει μία πιθανή παραβίαση του χρόνου διατήρησης;

Στον πίνακα δίδονται τα χαρακτηριστικά χρονισμού των στοιχείων του κυκλώματος



Συνδυαστική λογική:

$$t_{pd} = 3 \times 100 \text{ ps} = 300 \text{ ps}$$

$$t_{cd} = 1 \times 55 \text{ ps} = 55 \text{ ps}$$

Χαρ/κά χρονισμού

Καταχωρητή:

$$t_{ccq} = 50 \text{ ps}$$

$$t_{pcq} = 70 \text{ ps}$$

$$t_{setup} = 60 \text{ ps}$$

$$t_{hold} = 20 \text{ ps}$$

Πύλης XOR

$$t_{pd} = 100 \text{ ps}$$

$$t_{cd} = 55 \text{ ps}$$

$$(a) \quad T_c \geq t_{pcq} + t_{pd} + t_{setup} = (70+300+60) \text{ ps} = 430 \text{ ps}$$
$$f = 1 / 430 \text{ ps} = 2,33 \text{ GHz}$$

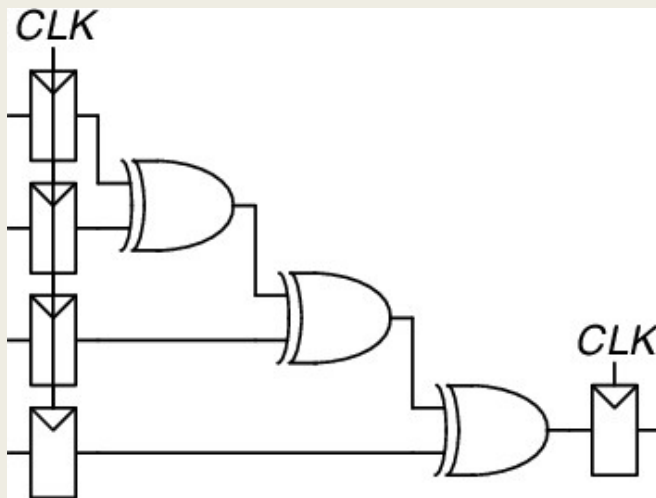
Επιλεγμένες ασκήσεις

■ Άσκηση 3.33

Ο Γιάννης Μπιτάκης έχει σχεδιάσει το κύκλωμα της Εικόνας 3.74 ώστε να υπολογίζει την τιμή μιας καταχωρισμένης (registered) συνάρτησης XOR με 4 εισόδους.

- (a) Ποια είναι η μέγιστη συχνότητα λειτουργίας του κυκλώματος με ρολόι χωρίς απόκλιση;
- (b) Πόση απόκλιση του ρολογιού μπορεί να ανέχεται το κύκλωμα αν πρέπει να λειτουργεί στα 2 GHz;
- (c) Πόση απόκλιση του ρολογιού μπορεί να ανέχεται το κύκλωμα προτού αντιμετωπίσει μία πιθανή παραβίαση του χρόνου διατήρησης;

Στον πίνακα δίδονται τα χαρακτηριστικά χρονισμού των στοιχείων του κυκλώματος



Συνδυαστική λογική:

$$t_{pd} = 3 \times 100 \text{ ps} = 300 \text{ ps}$$

$$t_{cd} = 1 \times 55 \text{ ps} = 55 \text{ ps}$$

Χαρ/κά χρονισμού

Καταχωρητή:

$$t_{ccq} = 50 \text{ ps}$$

$$t_{pcq} = 70 \text{ ps}$$

$$t_{setup} = 60 \text{ ps}$$

$$t_{hold} = 20 \text{ ps}$$

Πύλης XOR

$$t_{pd} = 100 \text{ ps}$$

$$t_{cd} = 55 \text{ ps}$$

(b) $T_c = 1 / f = 1 / 2 \text{ GHz} = 500 \text{ ps}$

$$T_c \geq t_{pcq} + t_{pd} + t_{setup} + t_{skew} \Rightarrow t_{skew} \leq T_c - (t_{pcq} + t_{pd} + t_{setup})$$

$$t_{skew} \leq [500 - (70 + 300 + 60)] \text{ ps} = (500 - 430) \text{ ps} = 70 \text{ ps}$$

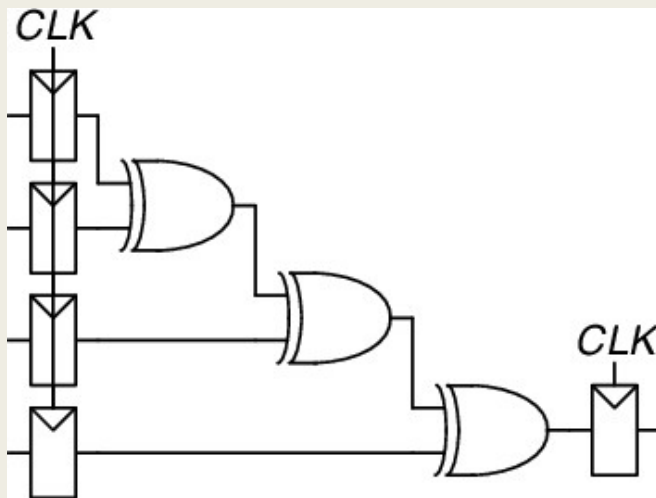
Επιλεγμένες ασκήσεις

■ Άσκηση 3.33

Ο Γιάννης Μπιτάκης έχει σχεδιάσει το κύκλωμα της Εικόνας 3.74 ώστε να υπολογίζει την τιμή μιας καταχωρισμένης (registered) συνάρτησης XOR με 4 εισόδους.

- (a) Ποια είναι η μέγιστη συχνότητα λειτουργίας του κυκλώματος με ρολόι χωρίς απόκλιση;
- (b) Πόση απόκλιση του ρολογιού μπορεί να ανέχεται το κύκλωμα αν πρέπει να λειτουργεί στα 2 GHz;
- (c) Πόση απόκλιση του ρολογιού μπορεί να ανέχεται το κύκλωμα προτού αντιμετωπίσει μία πιθανή παραβίαση του χρόνου διατήρησης;

Στον πίνακα δίδονται τα χαρακτηριστικά χρονισμού των στοιχείων του κυκλώματος



Συνδυαστική λογική:

$$t_{pd} = 3 \times 100 \text{ ps} = 300 \text{ ps}$$

$$t_{cd} = 1 \times 55 \text{ ps} = 55 \text{ ps}$$

Χαρ/κά χρονισμού

Καταχωρητή:

$$t_{ccq} = 50 \text{ ps}$$

$$t_{pcq} = 70 \text{ ps}$$

$$t_{\text{setup}} = 60 \text{ ps}$$

$$t_{\text{hold}} = 20 \text{ ps}$$

Πύλης XOR

$$t_{pd} = 100 \text{ ps}$$

$$t_{cd} = 55 \text{ ps}$$

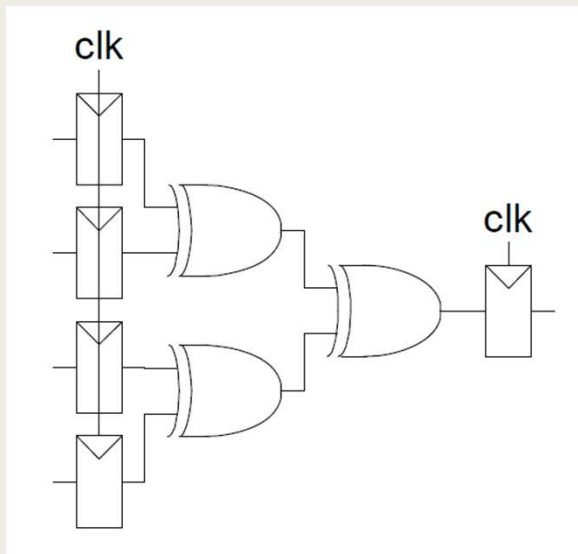
$$(c) \quad t_{\text{hold}} + t_{\text{skew}} < t_{ccq} + t_{cd} \Rightarrow t_{\text{skew}} < t_{ccq} + t_{cd} - t_{\text{hold}} \Rightarrow \\ t_{\text{skew}} < (50 + 55 - 20) \text{ ps} = 85 \text{ ps}$$

Επιλεγμένες ασκήσεις

■ Άσκηση 3.33

Η Μαρία Χακερίδου ισχυρίζεται ότι μπορεί να σχεδιάσει εκ νέου το τμήμα συνδυαστικής λογικής μεταξύ των καταχωρητών ώστε να είναι πιο γρήγορο.

- Το βελτιωμένο κύκλωμά της χρησιμοποιεί επίσης πύλες XOR με δύο εισόδους, οι οποίες όμως έχουν διαφορετική διάταξη
- (a) Ποια είναι η μέγιστη συχνότητά λειτουργίας του κυκλώματος;
- (b) Πόση απόκλιση του ρολογιού μπορεί να ανέχεται το κύκλωμα προτού αντιμετωπίσει μία πιθανή παραβίαση του χρόνου διατήρησης;



Συνδυαστική λογική:

$$t_{pd} = 2 \times 100 \text{ ps} = 200 \text{ ps}$$

$$t_{cd} = 2 \times 55 \text{ ps} = 110 \text{ ps}$$

Χαρ/κά χρονισμού

Καταχωρητή:

$$t_{ccq} = 50 \text{ ps}$$

$$t_{pcq} = 70 \text{ ps}$$

$$t_{setup} = 60 \text{ ps}$$

$$t_{hold} = 20 \text{ ps}$$

Πύλης XOR

$$t_{pd} = 100 \text{ ps}$$

$$t_{cd} = 55 \text{ ps}$$

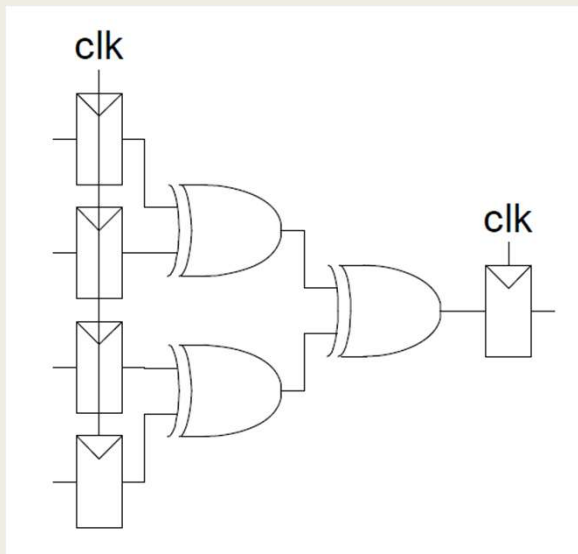
$$(a) \quad T_c \geq t_{pcq} + t_{pd} + t_{setup} = (70+200+60) \text{ ps} = 330 \text{ ps}$$
$$f = 1 / 330 \text{ ps} = 3,03 \text{ GHz}$$

Επιλεγμένες ασκήσεις

■ Άσκηση 3.33

Η Μαρία Χακερίδου ισχυρίζεται ότι μπορεί να σχεδιάσει εκ νέου το τμήμα συνδυαστικής λογικής μεταξύ των καταχωρητών ώστε να είναι πιο γρήγορο.

- Το βελτιωμένο κύκλωμά της χρησιμοποιεί επίσης πύλες XOR με δύο εισόδους, οι οποίες όμως έχουν διαφορετική διάταξη
- Ποιο είναι το σχηματικό διάγραμμα του κυκλώματός της;
- (a) Ποια είναι η μέγιστη συχνότητά λειτουργίας του κυκλώματος;
- (b) Πόση απόκλιση του ρολογιού μπορεί να ανέχεται το κύκλωμα προτού αντιμετωπίσει μία πιθανή παραβίαση του χρόνου διατήρησης;



Συνδυαστική λογική:

$$t_{pd} = 2 \times 100 \text{ ps} = 200 \text{ ps}$$

$$t_{cd} = 2 \times 55 \text{ ps} = 110 \text{ ps}$$

Χαρ/κά χρονισμού

Καταχωρητή:

$$t_{ccq} = 50 \text{ ps}$$

$$t_{pcq} = 70 \text{ ps}$$

$$t_{setup} = 60 \text{ ps}$$

$$t_{hold} = 20 \text{ ps}$$

Πύλης XOR

$$t_{pd} = 100 \text{ ps}$$

$$t_{cd} = 55 \text{ ps}$$

$$(b) \quad t_{hold} + t_{skew} < t_{ccq} + t_{cd} \Rightarrow t_{skew} < t_{ccq} + t_{cd} - t_{hold} \Rightarrow \\ t_{skew} < (50 + 110 - 20) \text{ ps} = 140 \text{ ps}$$

Επιλεγμένες ασκήσεις

■ Άσκηση 3.35

Μια προγραμματιζόμενη από τον χρήστη διάταξη πυλών FPGA, υλοποιεί κυκλώματα συνδυαστικής λογικής χρησιμοποιώντας διαμορφώσιμες λογικές δομικές μονάδες (configurable logic blocks, CLB) στη θέση των λογικών πυλών.

- Η διάταξη Spartan 3 της εταιρείας Xilinx έχει καθυστέρηση διάδοσης και μόλυνσης ίση με 0,61 και 0,30 ns, αντίστοιχα, για κάθε CLB.
- Περιέχει επίσης D Flip-Flop με καθυστερήσεις διάδοσης και μόλυνσης ίσες με 0,72 και 0,50 ns, και χρόνους σταθεροποίησης και διατήρησης ίσους με 0,53 και 0 ns, αντίστοιχα.

Αν κατασκευάζετε ένα σύστημα που πρέπει να λειτουργεί στα 40 MHz, πόσες διαδοχικές μονάδες CLB μπορείτε να χρησιμοποιήσετε μεταξύ δύο D Flip-Flop;

- Υποθέστε ότι το ρολόι δεν παρουσιάζει απόκλιση και δεν παρατηρείται καθυστέρηση κατά τη διέλευση μέσα από τα σύρματα μεταξύ των CLB

Επιλεγμένες ασκήσεις

■ Άσκηση 3.35

Μια προγραμματιζόμενη από τον χρήστη διάταξη πυλών FPGA, υλοποιεί κυκλώματα συνδυαστικής λογικής χρησιμοποιώντας διαμορφώσιμες λογικές δομικές μονάδες (configurable logic blocks, CLB) στη θέση των λογικών πυλών.

- Η διάταξη Spartan 3 της εταιρείας Xilinx έχει καθυστέρηση διάδοσης και μόλυνσης ίση με 0,61 και 0,30 ns, αντίστοιχα, για κάθε CLB.
- Περιέχει επίσης D Flip-Flop με καθυστερήσεις διάδοσης και μόλυνσης ίσες με 0,72 και 0,50 ns, και χρόνους σταθεροποίησης και διατήρησης ίσους με 0,53 και 0 ns, αντίστοιχα.

Αν κατασκευάζετε ένα σύστημα που πρέπει να λειτουργεί στα 40 MHz, πόσες διαδοχικές μονάδες CLB (N) μπορείτε να χρησιμοποιήσετε μεταξύ δύο D Flip-Flop;

- Υποθέστε ότι το ρολόι δεν παρουσιάζει απόκλιση και δεν παρατηρείται καθυστέρηση κατά τη διέλευση μέσα από τα σύρματα μεταξύ των CLB

$$T_c = 1 / f = 1 / 40 \text{ MHz} = 25 \text{ ns}$$

$$T_c \geq t_{pcq} + t_{pd} + t_{\text{setup}} + t_{\text{skew}} \Rightarrow T_c \geq t_{pcq} + N t_{pdCLB} + t_{\text{setup}} + t_{\text{skew}} \Rightarrow$$

$$N \leq [T_c - (t_{pcq} + t_{\text{setup}} + t_{\text{skew}})] / t_{pdCLB} \Rightarrow N \leq [25 - (0,72 + 0,53 + 0)] \text{ ns} / 0,61 \text{ ns} \Rightarrow$$

$$N \leq 38,9 \Rightarrow N = 38$$

Επιλεγμένες ασκήσεις

■ Άσκηση 3.35

Μια προγραμματιζόμενη από τον χρήστη διάταξη πυλών FPGA, υλοποιεί κυκλώματα συνδυαστικής λογικής χρησιμοποιώντας διαμορφώσιμες λογικές δομικές μονάδες (configurable logic blocks, CLB) στη θέση των λογικών πυλών.

- Η διάταξη Spartan 3 της εταιρείας Xilinx έχει καθυστέρηση διάδοσης και μόλυνσης ίση με 0,61 και 0,30 ns, αντίστοιχα, για κάθε CLB.
- Περιέχει επίσης D Flip-Flop με καθυστερήσεις διάδοσης και μόλυνσης ίσες με 0,72 και 0,50 ns, και χρόνους σταθεροποίησης και διατήρησης ίσους με 0,53 και 0 ns, αντίστοιχα.

Πόση απόκλιση του ρολογιού μπορεί να ανεχτεί η διάταξη FPGA χωρίς να παραβιάζει τον χρόνο διατήρησης;

- Υποθέστε ότι όλες οι διαδρομές μεταξύ D Flip-Flop περνούν από τουλάχιστον μία μονάδα CLB.

Επιλεγμένες ασκήσεις

■ Άσκηση 3.35

Μια προγραμματιζόμενη από τον χρήστη διάταξη πυλών FPGA, υλοποιεί κυκλώματα συνδυαστικής λογικής χρησιμοποιώντας διαμορφώσιμες λογικές δομικές μονάδες (configurable logic blocks, CLB) στη θέση των λογικών πυλών.

- Η διάταξη Spartan 3 της εταιρείας Xilinx έχει καθυστέρηση διάδοσης και μόλυνσης ίση με 0,61 και 0,30 ns, αντίστοιχα, για κάθε CLB.
- Περιέχει επίσης D Flip-Flop με καθυστερήσεις διάδοσης και μόλυνσης ίσες με 0,72 και 0,50 ns, και χρόνους σταθεροποίησης και διατήρησης ίσους με 0,53 και 0 ns, αντίστοιχα.

Πόση απόκλιση του ρολογιού μπορεί να ανεχτεί η διάταξη FPGA χωρίς να παραβιάζει τον χρόνο διατήρησης;

- Υποθέστε ότι όλες οι διαδρομές μεταξύ D Flip-Flop περνούν από τουλάχιστον μία μονάδα CLB.

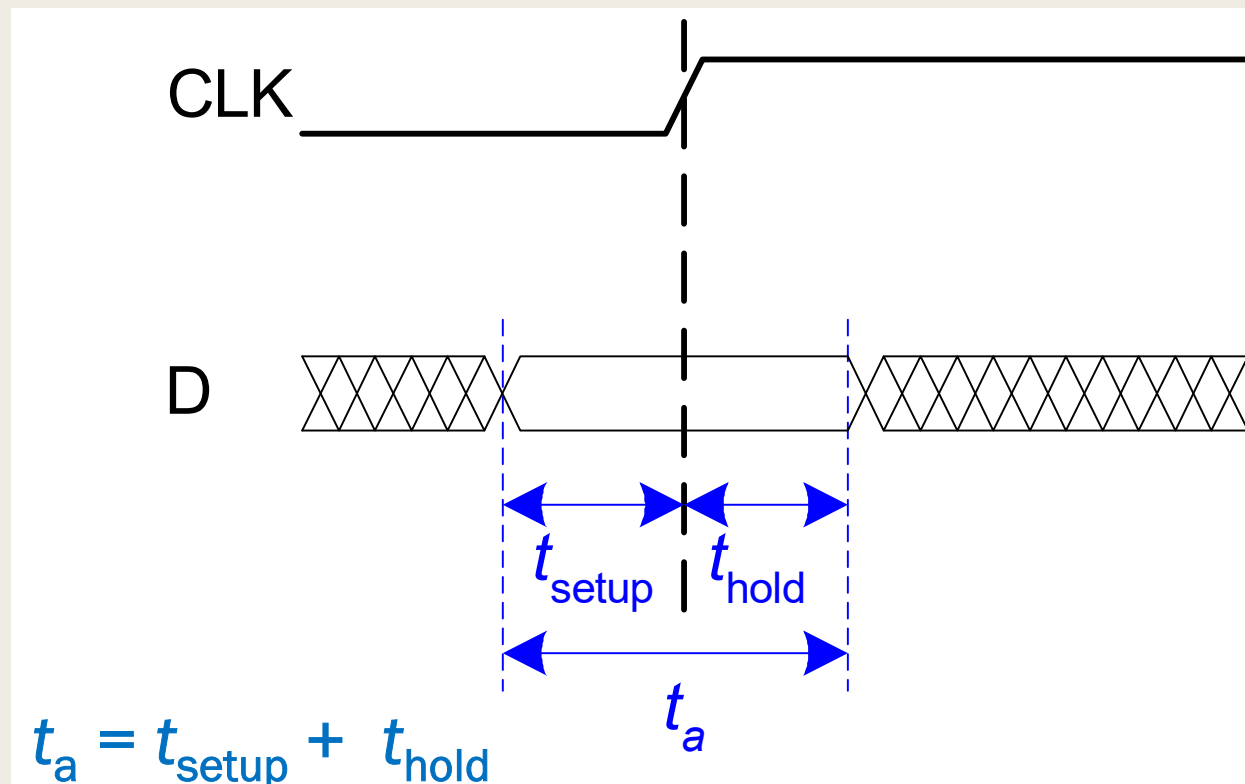
$$t_{\text{hold}} + t_{\text{skew}} < t_{\text{ccq}} + t_{\text{cd}} \Rightarrow t_{\text{skew}} < t_{\text{ccq}} + t_{\text{cd}} - t_{\text{hold}} \Rightarrow$$
$$t_{\text{skew}} < (0,50 + 0,30 - 0) \text{ ns} = 0,8 \text{ ns} = 800 \text{ ps}$$

Παραβίαση της δυναμικής πειθαρχίας

■ Χρονικοί περιορισμοί στην είσοδο D του D Flip-Flop

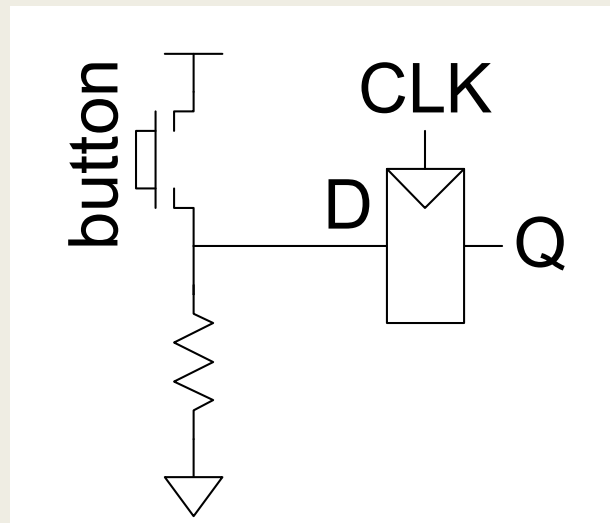
που πρέπει να ικανοποιούνται κατά τη δειγματοληψία:

- **Χρόνος σταθεροποίησης (set-up time, t_{setup})** = ο χρόνος που η είσοδος D πρέπει να παραμένει σταθερή **πριν** από την ανερχόμενη ακμή του CLK
- **Χρόνος διατήρησης (hold time, t_{hold})** = ο χρόνος που η είσοδος D πρέπει να παραμένει σταθερή **μετά** από την ανερχόμενη ακμή του CLK
- **Χρόνος ανοίγματος (aperture time, t_a)** = ο χρόνος που η είσοδος D πρέπει να παραμένει σταθερή **πριν και μετά** από την ανερχόμενη ακμή του CLK



Παραβίαση της δυναμικής πειθαρχίας

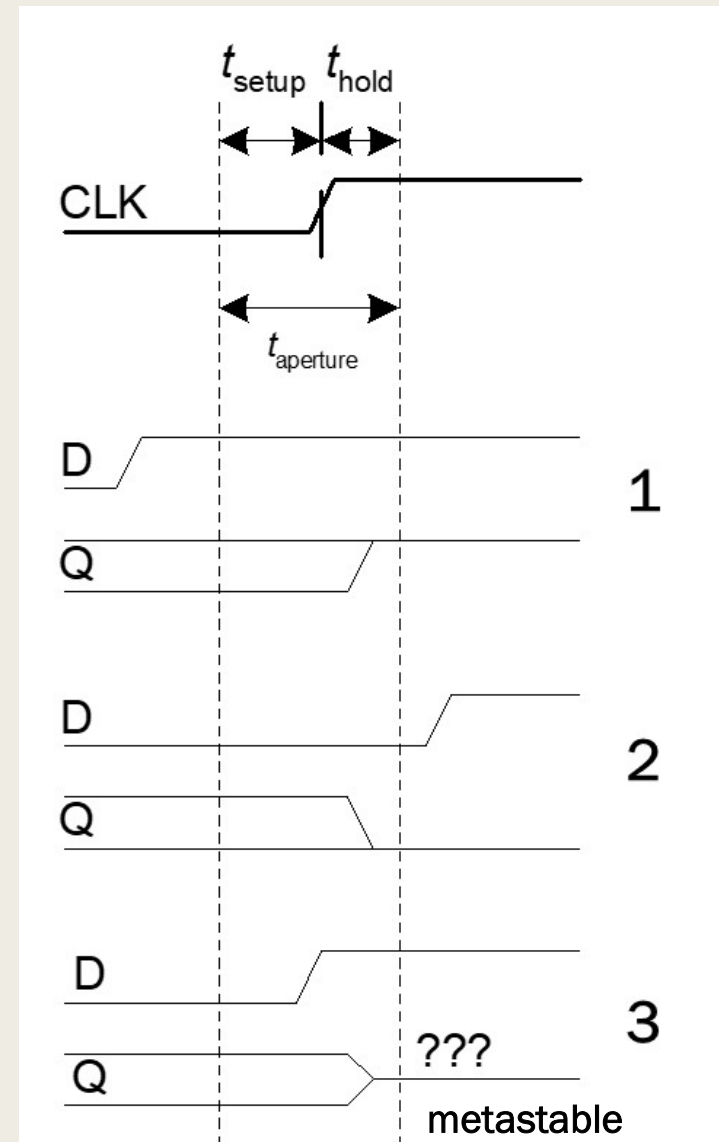
- Μία **ασύγχρονη είσοδος** μπορεί να παραβιάσει την δυναμική πειθαρχία
- Για παράδειγμα μία είσοδος D ενός D Flip-Flop που προέρχεται από ένα κουμπί (button)
 - $D = 0$, όταν το κουμπί δεν είναι πατημένο
 - $D = 1$, όταν το κουμπί είναι πατημένο
 - Το κουμπί πατιέται σε κάποια τυχαία χρονική στιγμή (ασύγχρονα) σε σχέση με την ανερχόμενη ακμή του ρολογιού



Παραβίαση της δυναμικής πειθαρχίας

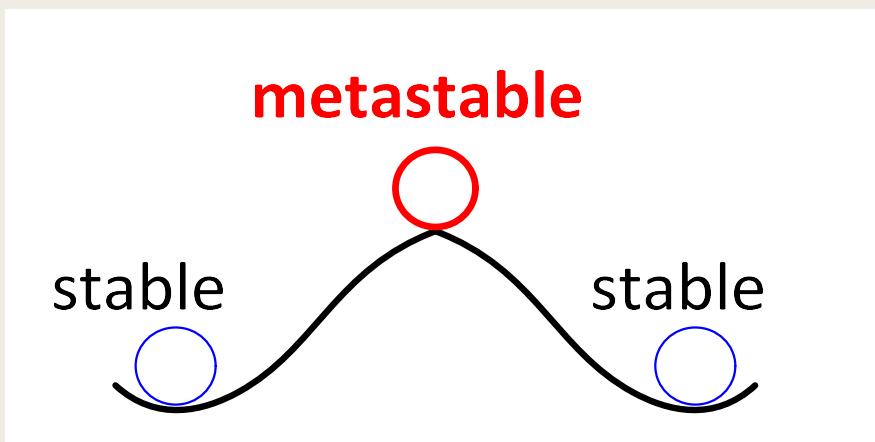
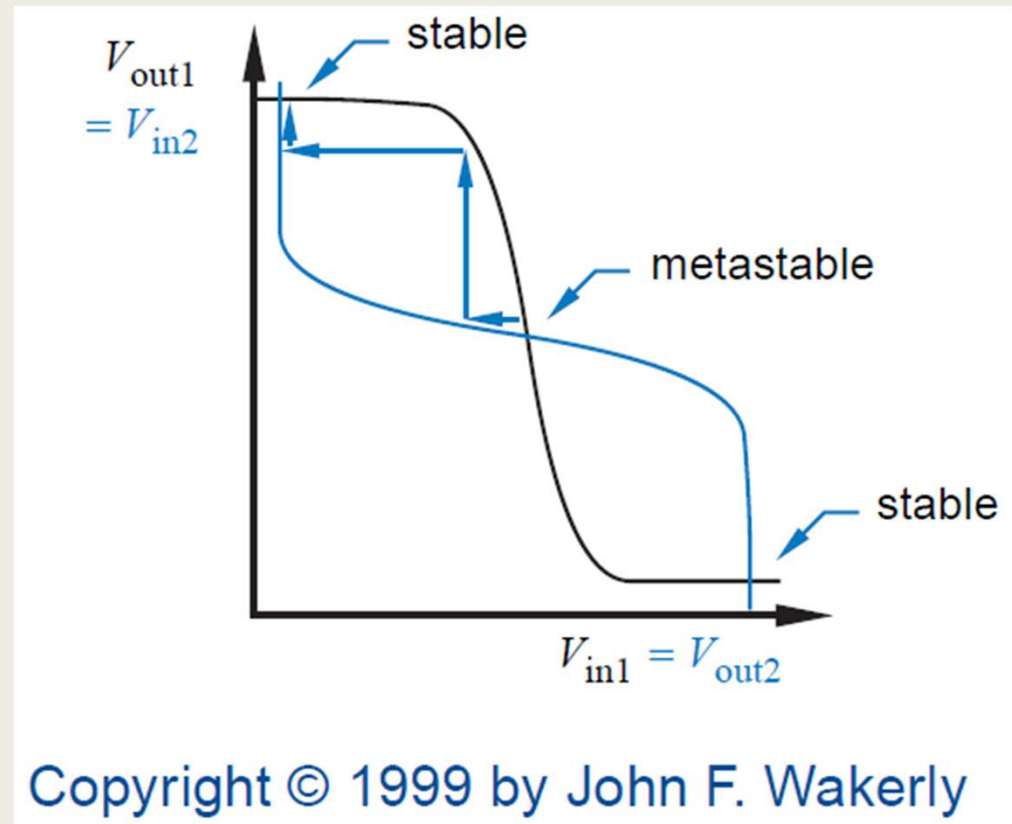
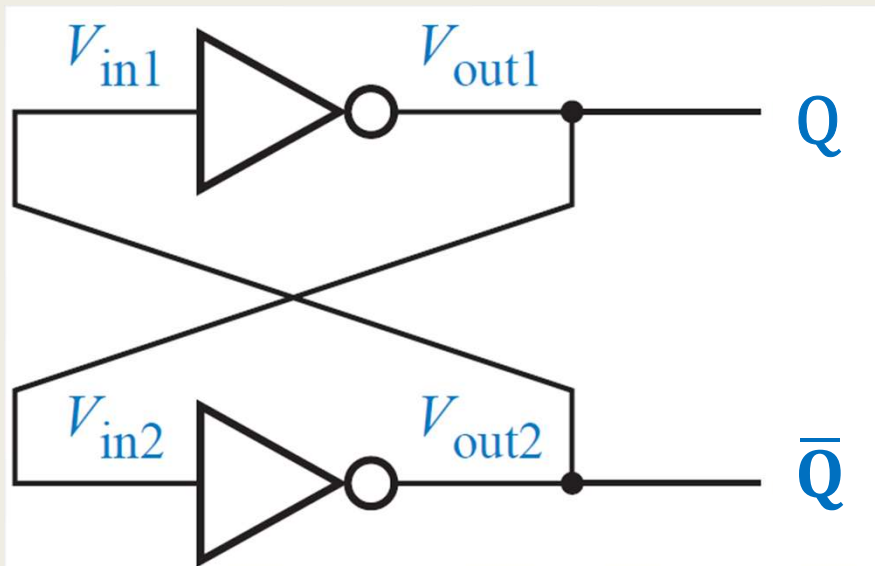
- Το κουμπί πατιέται σε κάποια τυχαία χρονική στιγμή (ασύγχρονα) σε σχέση με την ανερχόμενη ακμή του ρολογιού CLK

1. Όταν το κουμπί πατηθεί πολύ πριν την ανερχόμενη ακμή, $Q = 1$
2. Όταν το κουμπί δεν πατηθεί αρκετά μετά την ανερχόμενη ακμή, $Q = 0$
3. Όταν το κουμπί πατηθεί εντός του χρόνου ανοίγματος, $Q = 0-1$
 - Ενδέχεται στιγμιαία το Q να πάει σε μία **μετασταθερή κατάσταση (metastable)** για κάποιο χρονικό διάστημα, πριν καταλήξει σε μια σταθερή τιμή 0 ή 1
 - Αν περιμένουμε για αρκετό χρόνο, πολύ μεγαλύτερο της **καθυστερήσης διάδοσης (t_{pcq})**, μπορούμε να αναμένουμε με πολύ μεγάλη πιθανότητα ότι το Q θα καταλήξει σε μια σταθερή τιμή 0 ή 1, εντός του **χρόνου κατάληξης t_{tres}**



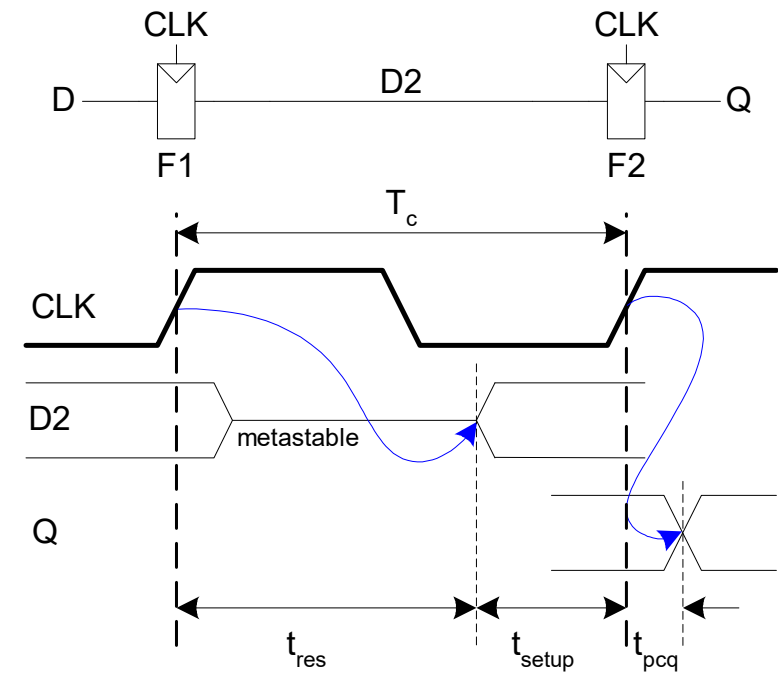
Μετασταθερή Κατάσταση

- Κατά την έναρξη της λειτουργίας του κυκλώματος των διασυζευγμένων αντιστροφέων, το κύκλωμα μπορεί να παραμείνει προσωρινά στην **μετασταθερή (metastable) κατάσταση**
 - όπου και οι δύο εξοδοι Q και $\bar{Q}$ βρίσκονται κατά προσέγγιση στο μέσο μεταξύ του 0 και του 1



Συγχρονιστές

- Σε ένα ψηφιακό σύστημα οι ασύγχρονες είσοδοι είναι αναπόφευκτες
 - Στις διεπαφές χρήστη,
 - Σε συστήματα με διαφορετικά ρολόγια που αλληλεπιδρούν
- Ο σχεδιαστής ενός ψηφιακού συστήματος πρέπει να εξασφαλίζει ότι, στην περίπτωση των ασύγχρονων εισόδων, η πιθανότητα να προκύψει μια μετασταθερή κατάσταση που θα προκαλέσει δυσλειτουργία στο σύστημα είναι επαρκώς μικρή.
- Αυτό επιτυγχάνεται με την χρήση ενός **συγχρονιστή** (2 D-FF στη σειρά)
 - Δέχεται μια **ασύγχρονη είσοδο D**
 - Παράγει με πολύ μεγάλη πιθανότητα μια **σύγχρονη έξοδο Q**
 - Όταν το εσωτερικό σήμα D2 καταλήξει σε μια σταθερή τιμή 0 ή 1 εντός του χρονικού διαστήματος ($t_{res} \leq T_c - t_{setup}$)



Παραλληλισμός: βασικοί ορισμοί

- **Οντότητα δεδομένων**: μια ομάδα δεδομένων στην είσοδο ενός συστήματος, που υπόκεινται σε επεξεργασία εντός του συστήματος, ώστε να παραχθεί μια άλλη ομάδα δεδομένων στην έξοδο του συστήματος
- **Λανθάνων χρόνος (latency)**: ο χρόνος που χρειάζεται η οντότητα δεδομένων για να διέλθει από το σύστημα, από την αρχή έως το τέλος (από τον πρώτο κύκλο ρολογιού μέχρι τον τελευταίο)
- **Διεκπεραιωτική ικανότητα (throughput)**: το πλήθος των οντοτήτων δεδομένων που μπορούν να παράγονται ανά μονάδα χρόνου (π.χ. ανά sec)
- **Παραλληλισμός (parallelism)**: η δυνατότητα του συστήματος να επεξεργάζεται πολλές οντότητες δεδομένων ταυτόχρονα
 - **Χωρικός παραλληλισμός**: πλεονασμός στο υλικό, ώστε να είναι εφικτή η ταυτόχρονη επεξεργασία πολλών οντοτήτων δεδομένων
 - **Χρονικός παραλληλισμός ή διοχέτευση (pipelining)**: η επεξεργασία εντός του συστήματος χωρίζεται σε στάδια, όπως σε μία γραμμή παραγωγής
 - Σε κάθε δεδομένη στιγμή μια διαφορετική οντότητα δεδομένων θα βρίσκεται σε κάθε στάδιο, οπότε πολλές οντότητες δεδομένων, όσες και τα στάδια, υπόκεινται σε ταυτόχρονη επεξεργασία

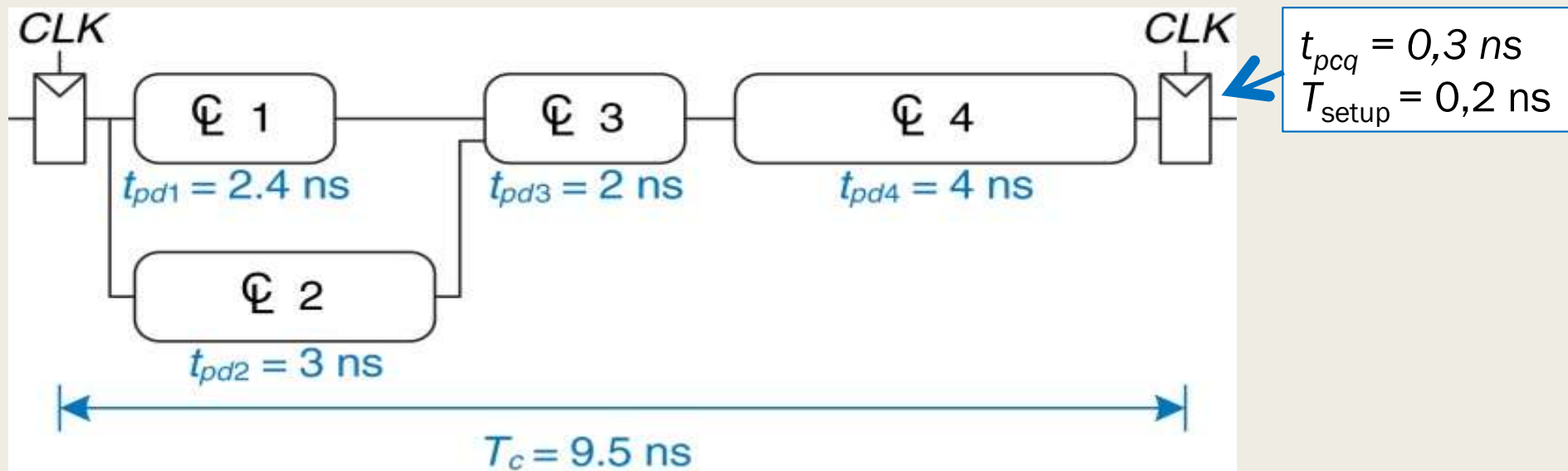
Διοχέτευση

- Η διοχέτευση είναι ιδιαίτερα ελκυστική επειδή αυξάνει την ταχύτητα ενός κυκλώματος χωρίς να απαιτεί πλεονασμό του υλικού.
- Αντ' αυτού, τοποθετούνται καταχωρητές ανάμεσα σε τμήματα συνδυαστικής λογικής, ώστε να διαιρεθεί η λογική σε μικρότερα στάδια τα οποία μπορούν να εκτελούνται με γρηγορότερο ρολόι
 - *Η περίοδος του ρολογιού εξαρτάται από το πιο αργό στάδιο*
- Οι καταχωρητές δεν επιτρέπουν σε μια οντότητα δεδομένων που βρίσκεται σε ένα στάδιο διοχέτευσης να «προφθάσει» και να αλλοιώσει την οντότητα δεδομένων που βρίσκεται στο επόμενο στάδιο

Διοχέτευση: παράδειγμα

■ Κύκλωμα χωρίς διοχέτευση

- Περιέχει τέσσερα τμήματα λογικής μεταξύ των καταχωρητών
 - Η κρίσιμη διαδρομή περνάει από τα τμήματα 2, 3 και 4



$$T_c \geq (t_{pcq} + t_{\text{setup}}) + t_{pd} = (0,5 + 2 + 3 + 4) \text{ ns} = 9,5 \text{ ns}$$

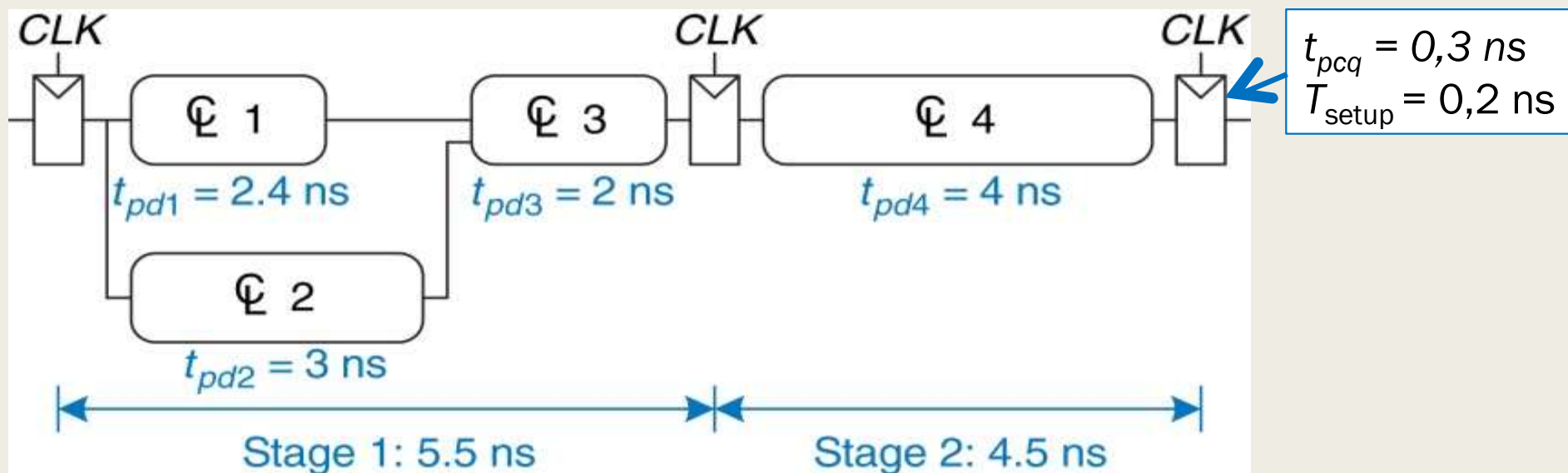
Επιβάρυνση δημιουργίας ακολουθίας
(sequencing overhead): $t_{pcq} + t_{\text{setup}} = 0,5 \text{ ns}$

Latency = 9,5 ns
Throughput = $1/9,5 \text{ ns} = 105 \text{ MHz}$

Διοχέτευση: παράδειγμα

■ Κύκλωμα με διοχέτευση δύο σταδίων

- Προσθήκη επιπλέον καταχωρητή ανάμεσα στα τμήματα 3 και 4
 - Η κρίσιμη διαδρομή περνάει από τα τμήματα 2 και 3 (στάδιο 1)



$$T_c \geq (t_{pcq} + t_{\text{setup}}) + t_{pd} = (0.5 + 2 + 3) \text{ ns} = 5.5 \text{ ns}$$

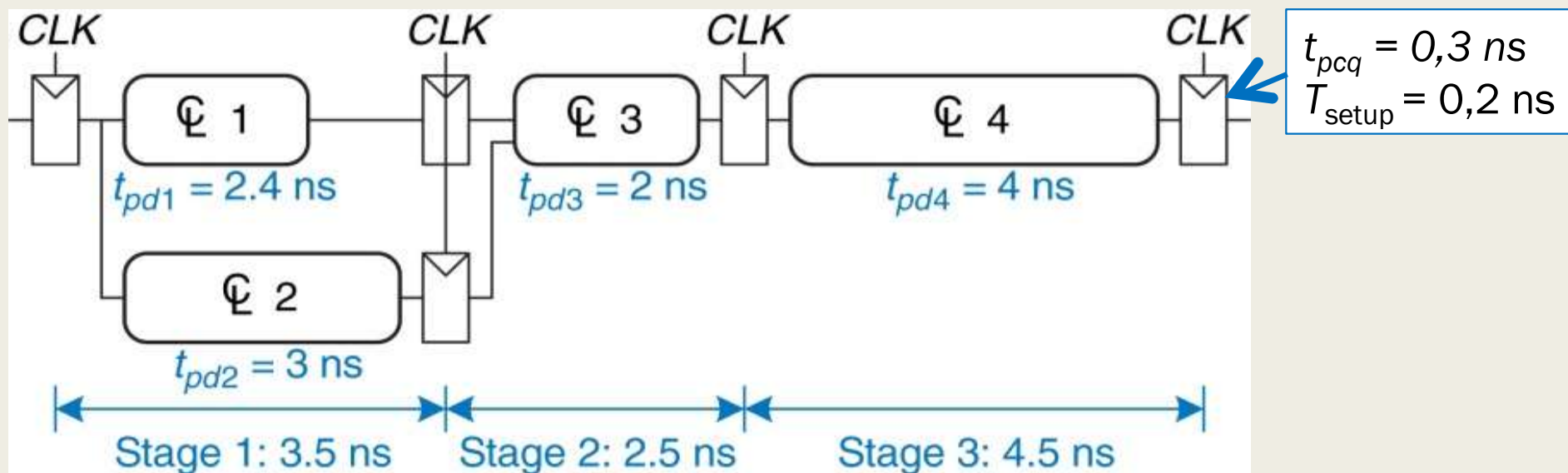
Επιβάρυνση δημιουργίας ακολουθίας
(sequencing overhead): $t_{pcq} + t_{\text{setup}} = 0.5 \text{ ns}$

Latency = $2 \times 5.5 \text{ ns} = 11 \text{ ns}$
Throughput = $1/5.5 \text{ ns} = 182 \text{ MHz}$

Διοχέτευση: παράδειγμα

■ Κύκλωμα με διοχέτευση τριών σταδίων

- Προσθήκη επιπλέον καταχωρητή ανάμεσα στα τμήματα 3 και 4
 - Η κρίσιμη διαδρομή περνάει από τα τμήματα 2 και 3 (στάδιο 1)



$$T_c \geq (t_{pcq} + t_{\text{setup}}) + t_{pd} = (0,5 + 4) \text{ ns} = 4,5 \text{ ns}$$

Επιβάρυνση δημιουργίας ακολουθίας
(sequencing overhead): $t_{pcq} + t_{\text{setup}} = 0,5 \text{ ns}$

Latency = $3 \times 4,5 \text{ ns} = 13,5 \text{ ns}$
Throughput = $1/4,5 \text{ ns} = 222 \text{ MHz}$

Διοχέτευση: συμπεράσματα

- Κύκλωμα χωρίς διοχέτευση

Latency = 9,5 ns

Throughput = $1/9,5 \text{ ns} = 105 \text{ MHz}$

- Κύκλωμα με διοχέτευση δύο σταδίων

Latency = $2 \times 5,5 \text{ ns} = 11 \text{ ns}$

Throughput = $1/5,5 \text{ ns} = 182 \text{ MHz}$

- Κύκλωμα με διοχέτευση τριών σταδίων

Latency = $3 \times 4,5 \text{ ns} = 13,5 \text{ ns}$

Throughput = $1/4,5 \text{ ns} = 222 \text{ MHz}$

- Σε ένα πραγματικό κύκλωμα, η διοχέτευση με N στάδια πολλαπλασιάζει σχεδόν κατά N φορές τη διεκπεραιωτική ικανότητα (throughput) και αυξάνει ανεκτά τον λανθάνοντα χρόνο (latency).

- Η ιδανική διοχέτευση θα πολλαπλασίαζε ακριβώς κατά N φορές τη διεκπεραιωτική ικανότητα χωρίς να αύξανε τον λανθάνοντα χρόνο
- Η διαφορά που παρατηρείται οφείλεται στο γεγονός ότι:
 - το κύκλωμα δεν μπορεί να χωριστεί σε N στάδια, όπου όλα τα στάδια να έχουν ακριβώς την ίδια καθυστέρηση διάδοσης,
 - που να ισούται με το $1/N$ της συνολικής καθυστέρησης διάδοσης που είχε το κύκλωμα πριν την εφαρμογή της διοχέτευσης
 - οι καταχωρητές της διοχέτευσης εισάγουν πρόσθετη επιβάρυνση δημιουργίας ακολουθίας $N(t_{pcq} + t_{setup})$ (όταν $t_{skew} = 0$)