Κ03 - Λογική Σχεδίαση
Δεύτερο Εξάμηνο Φοίτησης

Σύγχρονα ακολουθιακά κυκλώματα

Γεώργιος Χ. Αλεξανδρόπουλος
Λέκτορας Π.Δ. 407/80
e-mail: alexandg@uop.gr
URL: http://users.iit.demokritos.gr/~alexandg

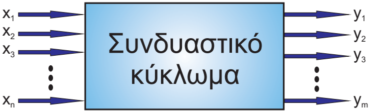
Πανεπιστήμιο Πελοποννήσου
Τμήμα Επιστήμης και Τεχνολογίας Τηλεπικοινωνιών

Περιεχόμενα

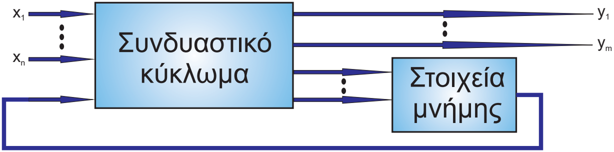
Εισαγωγή

Τα ψηφιακά λογικά κυκλώματα ταξινομούνται σε:

- Συνδυαστικά λογικά κυκλώματα



- Ακολουθιακά λογικά κυκλώματα



Ακολουθιακά Λογικά Κυκλώματα

- Τα στοιχεία μνήμης είναι κυκλώματα που δύνανται να αποθηκεύσουν 2-δική πληροφορία
- Η 2-δική πληροφορία που αποθηκεύεται στα στοιχεία μνήμης ορίζει την κατάσταση του ακολουθιακού κυκλώματος
- Οι τιμές των εξόδων ενός ακολουθιακού κυκλώματος καθορίζονται από τις τιμές των εισόδων του και την τρέχουσα κατάσταση των στοιχείων μνήμης του
- Η χρονική ακολουθία εισόδων, εξόδων κι εσωτερικών καταστάσεων ενός ακολουθιακού κυκλώματος καθορίζει τη λειτουργία του

Ταξινόμηση Ακολουθιακών Κυκλωμάτων

- *Σύγχρονα Ακολουθιακά Κυκλώματα:*
  Οι έξοδοί τους καθορίζονται από τις τιμές των εσωτερικών τους σημάτων σε διακριτές χρονικές στιγμές

- *Ασύγχρονα Ακολουθιακά Κυκλώματα:*
  Οι τιμές των εξόδων τους εξαρτώνται ανά πάσα στιγμή τόσο από τις τιμές των εσωτερικών τους σημάτων όσο κι από τη σειρά με την οποία οι τιμές αυτές άλλαξαν

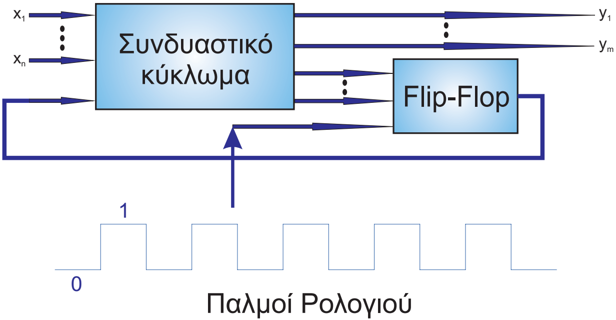
Ασύγχρονα Ακολουθιακά Κυκλώματα

- Τα στοιχεία μνήμης είναι διατάξεις που προκαλούν χρονικές καθυστερήσεις στα εμπλεκόμενα σήματα
- Η δυνατότητα αποθήκευσης μιας διάταξης με χρονική καθυστέρηση εξαρτάται από το χρόνο που χρειάζεται το εμπλεκόμενο σήμα να διαδοθεί μέσα από τη διάταξη αυτή
- Στα ασύγχρονα ακολουθιακά κυκλώματα οι διατάξεις αυτές αποτελούνται *μόνο* από λογικές πύλες κι η χρονική καθυστέρηση που αυτές παρέχουν προκύπτει από την καθυστέρηση διάδοσης του σήματος στις πύλες που τις αποτελούν
- Η καθυστέρηση διάδοσης στις διατάξεις αυτές επιτρέπει την *προσωρινή* αποθήκευση πληροφορίας
- Όμως, η ύπαρξη βρόγχων ανάδρασης αυξάνει την πιθανότητα ασταθούς λειτουργίας, καθιστώντας έτσι δύσκολο το σχεδιασμό ασύγχρονων ακολουθιακών κυκλωμάτων

Σύγχρονα Ακολουθιακά Κυκλώματα

- Οι τιμές των στοιχείων μνήμης επηρεάζονται μόνο σε διακριτές χρονικές στιγμές
- Χρησιμοποιώντας μια ηλεκτρονική διάταξη παραγωγής σειράς τετραγωνικών παλμών, γνωστοί ως *παλμοί ρολογιού*, επιτυγχάνεται ο διακριτός χρονισμός των στοιχείων μνήμης
- Οι παλμοί ρολογιού καθορίζουν πότε θα γίνουν οι αλλαγές στην κατάσταση του συστήματος ενώ τα υπόλοιπα σήματα του κυκλώματος καθορίζουν ποιες αλλαγές θα γίνουν και πώς θα επηρεαστούν τα στοιχεία μνήμης κι οι μεταβλητές εξόδου από αυτές
- Τα σύγχρονα ακολουθιακά κυκλώματα στα οποία χρησιμοποιούνται παλμοί ρολογιού για τον έλεγχο του χρονισμού λειτουργίας των στοιχείων μνήμης ονομάζονται *ακολουθιακά κυκλώματα με ρολόι*

Ακολουθιακό Κυκλώμα με Ρολόι



- Εμφανίζουν σπάνια προβλήματα αστάθειας
- Ο χρονισμός τους διαχωρίζεται εύκολα σε διακριτές, ανεξάρτητες χρονικές περιόδους, καθεμιά από τις οποίες δύναται να εξεταστεί χωριστά

Ακολουθιακό Κυκλώμα με Ρολόι

- Στα ακολουθιακά κυκλώματα με ρολόι χρησιμοποιούνται στοιχεία μνήμης γνωστά ως φλιπ-φλοπ (flip-flop (FF))
- Το FF δύναται να αποθηκεύσει ένα bit πληροφορίας
- Ανάλογα με τις ανάγκες αποθήκευσης σε bit ενός κυκλώματος επιλέγεται το αναγκαίο πλήθος από FFs
- Σε σταθερή κατάσταση η έξοδος ενός FF είναι 0 ή 1
- Οι 2-δικές τιμές αποθήκευσης σε κάθε παλμό ρολογιού εξαρτώνται από τις εισόδους του κυκλώματος κι από τις αποθηκευμένες τιμές στα FF από τον προηγούμενο παλμό ρολογιού

Ακολουθιακό Κύκλωμα με Ρολόι

- Η αποθήκευση των νέων τιμών στα FFs λαμβάνει χώρα ακριβώς τη στιγμή έλευσης ενός νέου παλμού

Συγχρονισμός Λειτουργιών Κυκλώματος

Το συνδυαστικό τμήμα του κυκλώματος θα πρέπει να έχει φτάσει σε σταθερές, τελικές τιμές εξόδων πριν την άφιξη ενός νέου παλμού ρολογιού$\longrightarrow$
Οι καθυστερήσεις διάδοσης στο κύκλωμα παίζουν σημαντικό ρόλο στον προσδιορισμό της ελάχιστης περιόδου των παλμών ρολογιού ώστε το κύκλωμα να λειτουργεί σωστά

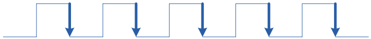
- Η κατάσταση των FFs αλλάζει *μόνο* κατά τις μεταβάσεις του σήματος παλμών ρολογιού $0 \rightarrow 1$ και $1 \rightarrow 0$. Όταν δεν υπάρχει μετάβαση, οι βρόγχοι ανάδρασης ουσιαστικά δε λειτουργούν

Ακολουθιακό Κυκλώμα με Ρολόι

- Τα στοιχεία μνήμης διαφοροποιούνται στο πλήθος των εισόδων τους και στον τρόπο με τον οποίο οι είσοδοί τους επηρεάζουν την επόμενή τους κατάσταση
- Μανδαλωτής (Latch)

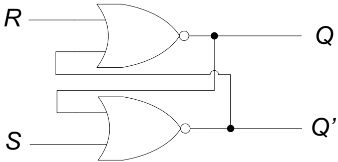
- Θετικά Ακμοπυροδοτούμενο (edge triggered) FF

- Αρνητικά Ακμοπυροδοτούμενο FF



Μανδαλωτής τύπου SR με πύλες NOR

- Δύο είσοδοι: S (Set) και R (Reset)
- Δύο έξοδοι: Q και το συμπλήρωμά της Q'

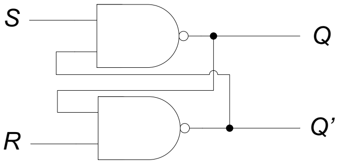


S	R	Q(n-1)	Q(n)	
0	0	0	0	Q(n-1)
0	0	1	1	
0	1	0	0	0 (Reset)
0	1	1	0	
1	0	0	1	1 (Set)
1	0	1	1	
1	1	0	X	X
1	1	1	X	

- Δύο χρήσιμες καταστάσεις:
  $Q = 1\ (Q' = 0) \rightarrow$ *κατάσταση θέσης (Set)*
  $Q = 0\ (Q' = 1) \rightarrow$ *κατάσταση μηδενισμού (Reset)*
- Για $S = R = 1$ προκύπτει απροσδιόριστη κατάσταση, η οποία και πρέπει να αποφεύγεται

Μανδαλωτής τύπου SR με πύλες NAND

- Πλέον, σε κανονική λειτουργία (*αποθήκευσης*) θα πρέπει $S = R = 1$
- Ο μανδαλωτής αυτός είναι γνωστός κι ως μανδαλωτής $S'R'$

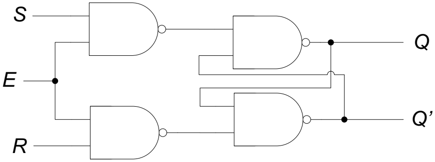


S	R	Q(n-1)	Q(n)	
0	0	0	X	X
0	0	1	X	
0	1	0	1	1 (Set)
0	1	1	1	
1	0	0	0	0 (Reset)
1	0	1	0	
1	1	0	0	Q(n-1)
1	1	1	1	

- Δύο χρήσιμες καταστάσεις:
  $Q = 1$ $(Q' = 0) \rightarrow$ *κατάσταση θέσης (Set)*
  $Q = 0$ $(Q' = 1) \rightarrow$ *κατάσταση μηδενισμού (Reset)*
- Για $S = R = 0$ προκύπτει απροσδιόριστη κατάσταση, η οποία και πρέπει να αποφεύγεται

Μανδαλωτής τύπου SR με είσοδο ελέγχου

- Στο μανδαλωτή τύπου SR προσθέτω δύο πύλες NAND
- Η είσοδος ελέγχου E καθορίζει πότε θα αλλάξει η κατάσταση του μανδαλωτή

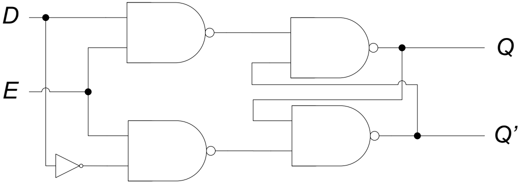


S	R	Q(n-1)	Q(n)	
0	0	0	0	Q(n-1)
0	0	1	1	
0	1	0	0	0 (Reset)
0	1	1	0	
1	0	0	1	1 (Set)
1	0	1	1	
1	1	0	X	X
1	1	1	X	

- Για $E = 0$ ο μανδαλωτής βρίσκεται σε αδράνεια
- Για $E = 1$ ο μανδαλωτής λειτουργεί όπως ο τύπου SR με πύλες NOR

Μανδαλωτής τύπου D

- Για να αποφευχθεί η απροσδιόριστη κατάσταση εξασφαλίζουμε ότι τα S και R δε θα γίνουν ταυτόχρονα 1

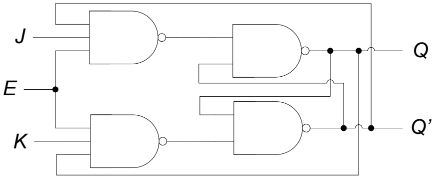


E	D	Q(n)
0	X	Q(n-1)
1	0	0
1	1	1

- Για $E = 0$ ο μανδαλωτής βρίσκεται σε αδράνεια
- Για $E = 1$ η τιμή του D μεταφέρεται στο Q

Μανδαλωτής τύπου JK

- Τρεις είσοδοι: J, K και E
- Δύο έξοδοι: Q και το συμπλήρωμά της Q'



J	K	Q(n-1)	Q(n)	
0	0	0	0	Q(n-1)
0	0	1	1	
0	1	0	0	0
0	1	1	0	
1	0	0	1	1
1	0	1	1	
1	1	0	1	Q'(n-1)
1	1	1	0	

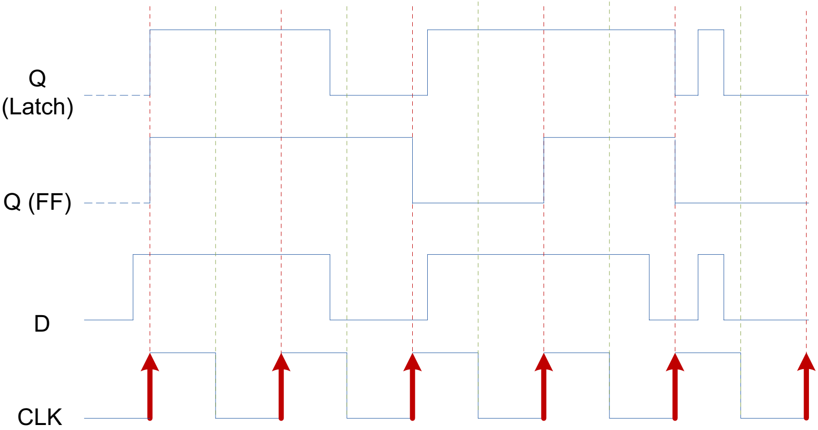
- Ο μανδαλωτής τύπου JK λειτουργεί όπως ο τύπου SR με σήμα ελέγχου, δεχόμενος πλέον την κατάσταση $J = K = 1$
- Για $J = K = 1$ η έξοδος είναι το συμπλήρωμα της προηγούμενης κατάστασης

Φλιπ-Φλοπ

- Αντίθετα με τους μανδαλωτές που ενεργοποιούνται με τη στάθμη δυναμικού του παλμού ρολογιού (level triggered), τα FFs είναι ακμοπυροδοτούμενα
- Τα FFs ουσιαστικά δειγματοληπτούν την είσοδό τους σε κάποια από τις δύο ακμές ρολογιού
- Θετικά Ακμοπυροδοτούμενο FF

- Αρνητικά Ακμοπυροδοτούμενο FF

Μανδαλωτές και Φλιπ-Φλοπ



Μανδαλωτές και Φλιπ-Φλοπ

- Με τους μανδαλωτές, οι καταστάσεις μεταβάλλονται ανάλογα με τις είσοδους τους όσο ο παλμός ρολογιού είναι στο 1
- Ενδέχεται, λοιπόν, κατά τη διάρκεια ενός παλμού στο 1 να ανατραφοδοτηθούν μέσω του βρόγχου ανάδρασης πολλές νέες καταστάσεις στους μανδαλωτές συντελώντας έτσι στην έλλειψη δυνατότητας πρόβλεψής τους
- Συνεπώς, η έξοδος ενός μανδαλωτή δεν πρέπει να συνδέεται είτε απευθείας είτε μέσω συνδυαστικής λογικής με την είσοδο του ίδιου ή άλλου μανδαλωτή, όταν όλοι πυροδοτούνται από το ίδιο ρολόι
- Σε αντίθεση, τα FFs είναι κατάλληλα για χρήση σε ακολουθιακά κυκλώματα που χρησιμοποιούν ένα κοινό ρολόι για όλα τα στοιχεία μνήμης

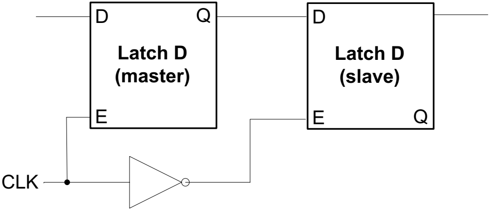
Μετατροπή Μανδαλωτή σε Φλιπ-Φλοπ

1ος Τρόπος: Διασυνδέω με ειδικό τρόπο δύο μανδαλωτές έτσι ώστε η έξοδος του FF να απομονώνεται από τις εισόδους του ώστε αυτή να μην αλλάξει τιμή όσο διαρκούν οι αλλαγές των εισόδων του (*απλούστερος τρόπος*)

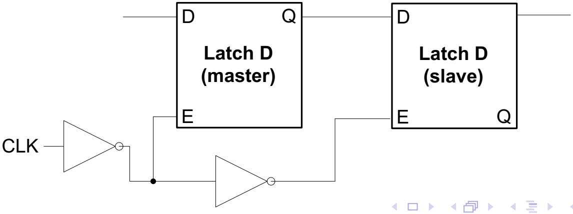
2ος Τρόπος: Σχεδιάζω κατάλληλα το FF έτσι ώστε να πυροδοτείται μόνο κατά τη μετάβαση των παλμών ρολογιού και να απενεργοποιείται καθ' όλη τη διάρκεια του ίδιου του παλμού

Ακμοπυροδοτούμενο Φλιπ-Φλοπ D

- Αρνητικά Ακμοπυροδοτούμενο D FF

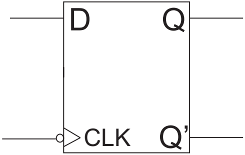


- Θετικά Ακμοπυροδοτούμενο D FF

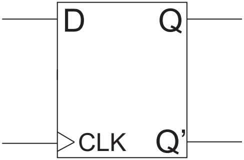


Ακμοπυροδοτούμενο Φλιπ-Φλοπ D

- Αρνητικά Ακμοπυροδοτούμενο D FF

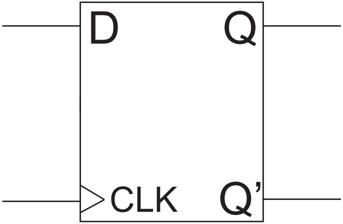


- Θετικά Ακμοπυροδοτούμενο D FF



- Πρόκειται για το πιο οικονομικό κι αποδοτικό FF

Ακμοπυροδοτούμενο Φλιπ-Φλοπ D

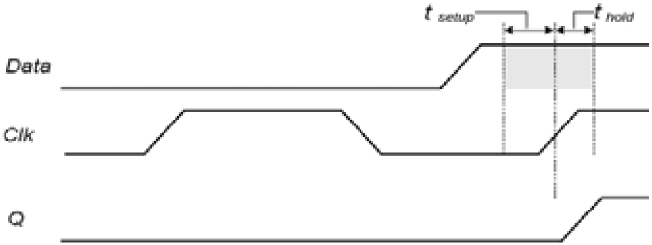


D	Q(n)
0	0
1	1

Χαρακτηριστική εξίσωση: $Q(n) = D$

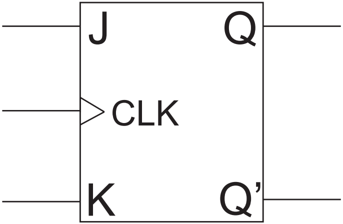
Με την έλευση θετικής ακμής ρολογιού η τιμή της εξόδου θα είναι αυτή που έχει η είσοδος

Χρονικοί Περιορισμοί Φλιπ-Φλοπ



- *Χρόνος προετοιμασίας:* ο ελάχιστος χρόνος που πρέπει να κρατηθούν σταθερά τα δεδομένα πριν την ακμή ρολογιού
- *Χρόνος κρατήματος:* ο ελάχιστος χρόνος που πρέπει να κρατηθούν σταθερά τα δεδομένα μετά την ακμή ρολογιού
- *Χρόνος καθυστέρησης διάδοσης:* ο ελάχιστος χρόνος μετά την ακμή πυροδότησης που απαιτείται για να σταθεροποιηθούν οι έξοδοι του FF

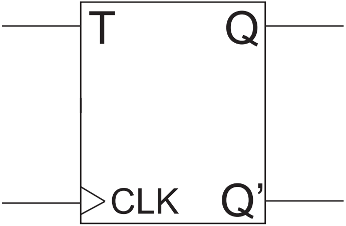
Φλιπ-Φλοπ JK



J	K	Q(n)
0	0	Q(n-1)
0	1	0
1	0	1
1	1	Q'(n-1)

Χαρακτηριστική εξίσωση: $Q(n) = JQ'(n-1) + K'Q(n-1)$

Φλιπ-Φλοπ T



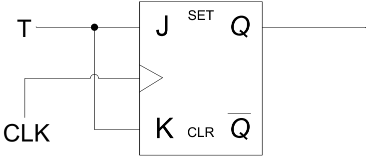
T	Q(n)
0	Q(n-1)
1	Q'(n-1)

Χαρακτηριστική εξίσωση: $Q(n) = TQ'(n-1) + T'Q(n-1)$

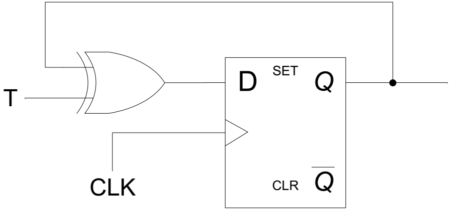
Με την πυροδότησή του η τιμή της εξόδου θα είναι το συμπλήρωμα της τιμής της εισόδου

Φλιπ-Φλοπ T από άλλα

- T FF από JK FF



- T FF από D FF



Άμεσοι Είσοδοι σε Φλιπ-Φλοπ

- Για την αρχικοποίηση ενός FF σε κατάσταση θέσης ή μηδενισμού χρησιμοποιούνται ως άμεσοι είσοδοι:

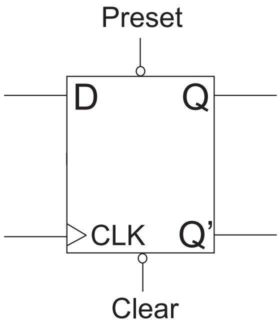
  Είσοδος θέσης (Preset/Direct Set) $\longrightarrow Q = 1$

  Είσοδος μηδενισμού (Clear/ReSet) $\longrightarrow Q = 0$

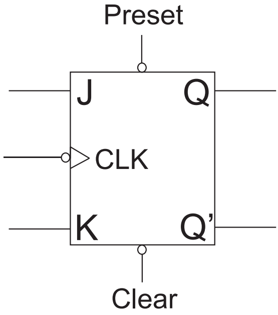
- Οι άμεσες είσοδοι δύνανται να δρούν *σύγχρονα* ή *ασύγχρονα*

Φλιπ-Φλοπ με Preset και Clear

Θετικά ακμοπυροδοτούμενο D FF με Preset και Clear



Αρνητικά ακμοπυροδοτούμενο JK FF με Preset και Clear

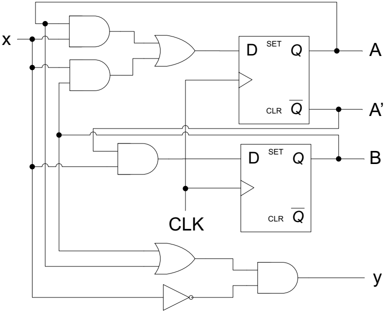


Ανάλυση Ακολουθιακών Κυκλωμάτων με Ρολόι

- Η διαδικασία της ανάλυσης κυκλώματος περιλαμβάνει την εύρεση:
  1) των εξισώσεων καταστάσεων
  2) του πίνακα καταστάσεων
  3) του διαγράμματος κατάστασης
- Οι εξισώσεις κατάστασης περιγράφουν την επόμενη χρονικά κατάσταση του κυκλώματος ως συνάρτηση της παρούσας κατάστασης και των εισόδων του
- Ο πίνακας κατάστασης περιέχει όλους τους δυνατούς συνδυασμούς μεταξύ των μεταβλητών της παρούσας κατάστασης και των μεταβλητών εισόδου σε σχέση με τις μεταβλητές της επόμενης κατάστασης και των μεταβλητών εξόδου
- Το διάγραμμα κατάστασης αναπαριστά σχηματικά τον πίνακα κατάστασης

Ανάλυση Κυκλώματος με Φλιπ-Φλοπ D

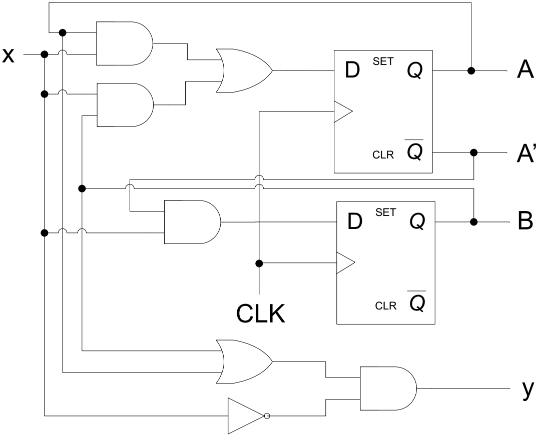
- Μία είσοδος x, μία έξοδος y και δύο FFs D
- FF A:
  $A(n+1) = A(n)x(n) + B(n)x(n)$
- FF B:
  $B(n+1) = A'(n)x(n)$
- Έξοδος κυκλώματος:
  $y(n) = [A(n) + B(n)]\, x(n)$



Η έξοδος του κυκλώματος εξαρτάται από την παρούσα κατάσταση των FFs και την είσοδο του κυκλώματος

Ανάλυση Κυκλώματος με Φλιπ-Φλοπ D

- Είσοδος FF A:
  $D_A = Ax + Bx$
- Είσοδος FF B:
  $D_B = A'x$
- Έξοδος κυκλώματος:
  $y = (A + B)x$
- Επόμενη κατάσταση FFs:
  $A(n+1) = D_A$
  $B(n+1) = D_B$



Ανάλυση Κυκλώματος με Φλιπ-Φλοπ D

- $A(n+1) = A(n)x(n) + B(n)x(n)$
- $B(n+1) = A'(n)x(n)$
- $y(n) = [A(n) + B(n)]\, x(n)$

Current State		Input	Next State		Output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

Ανάλυση Κυκλώματος με Φλιπ-Φλοπ D

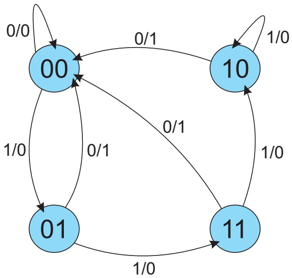
- $A(n+1) = A(n)x(n) + B(n)x(n)$
- $B(n+1) = A'(n)x(n)$
- $y(n) = [A(n) + B(n)]\, x(n)$

Current State	Next State		Output	
	x=0	**x=1**	**x=0**	**x=1**
AB	AB	AB	y	y
00	00	01	0	0
01	00	11	1	0
10	00	10	1	0
11	00	10	1	0

Ανάλυση Κυκλώματος με Φλιπ-Φλοπ D

Current State	Next State		Output	
	x=0	x=1	x=0	x=1
AB	AB	AB	y	y
00	00	01	0	0
01	00	11	1	0
10	00	10	1	0
11	00	10	1	0

Διάγραμμα Καταστάσεων



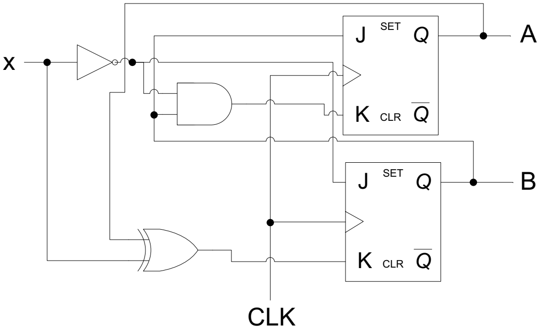
- Κύκλοι → αναπαριστούν τις καταστάσεις
- Βέλη → αναπαριστούν τις μεταβάσεις για δεδομένη είσοδο
- Είσοδοι/Έξοδοι

Ανάλυση Κυκλωμάτων με Φλιπ-Φλοπ JK και T

- Στα FFs D η εξίσωση κατάστασής τους είνα ίδια με την εξίσωση εξόδου τους
- Στα FFs JK και T η επόμενη τους κατάσταση προκύπτει από τη χαρακτηριστική τους εξίσωση
- Έτσι, με βάση την παρούσα κατάσταση, τις εισόδους του κυκλώματος και το χαρακτηριστικό πίνακα των FFs προσδιορίζω την επόμενη τους κατάσταση

Ανάλυση Κυκλώματος με Φλιπ-Φλοπ JK

- Μία είσοδος x
- Είσοδος FF A:
  $J_A = B$
  $K_A = Bx'$
- Είσοδος FF B:
  $J_B = x'$
  $K_B = A \oplus x$
- Δύο έξοδοι, οι έξοδοι των FFs



Ανάλυση Κυκλώματος με Φλιπ-Φλοπ JK

- Μία είσοδος x
- Είσοδος FF A:
  $J_A = B$
  $K_A = Bx'$
- Είσοδος FF B:
  $J_B = x'$
  $K_B = A \oplus x$
- Δύο έξοδοι, οι έξοδοι των FFs

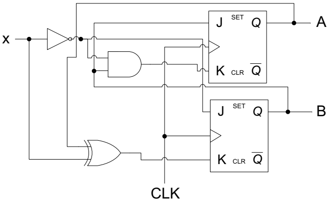
Current State		Input	Next State		Flip-Flop Inputs			
A	B	x	A	B	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

Ανάλυση Κυκλώματος με Φλιπ-Φλοπ JK

- Επόμενη κατάσταση FF A:
  $A(n+1) = J'_A K_A + J_A K'_A \;\Rightarrow$
  $A(n+1) = A \oplus B + Ax$
- Επόμενη κατάσταση FF B:
  $B(n+1) = J'_B K_B + J_B K'_B \;\Rightarrow$
  $B(n+1) = (A'B + B')x' + ABx$

Current State		Input	Next State	
A	B	x	A	B
0	0	0	0	1
0	0	1	0	0
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

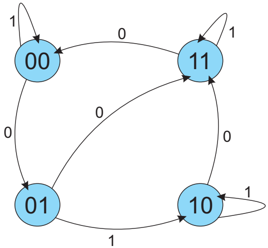
Ανάλυση Κυκλώματος με Φλιπ-Φλοπ JK



Πίνακας Καταστάσεων

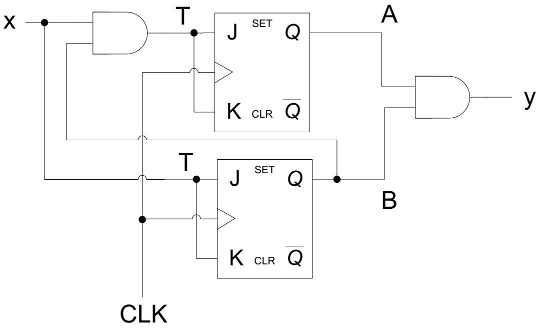
Current State		Input	Next State	
A	B	x	A	B
0	0	0	0	1
0	0	1	0	0
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

Διάγραμμα Καταστάσεων



Ανάλυση Κυκλώματος με Φλιπ-Φλοπ T

- Μία είσοδος x
- Είσοδος FF A: $T_A = Bx$
- Είσοδος FF B: $T_B = x$
- Έξοδος κυκλώματος: $y = AB$



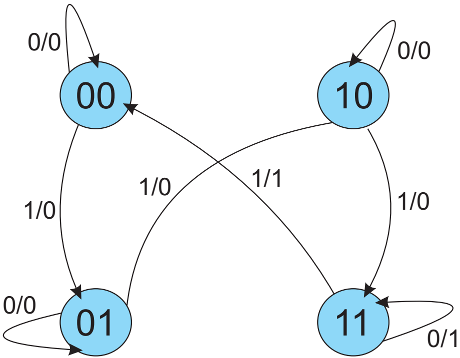
Ανάλυση Κυκλώματος με Φλιπ-Φλοπ T

- Επόμενη κατάσταση FF A:
  $A(n+1) = T'_A A + T_A A' \Rightarrow$
  $A(n+1) = A \oplus (Bx)$
- Επόμενη κατάσταση FF B:
  $B(n+1) = T'_B B + T_B B' \Rightarrow$
  $B(n+1) = B \oplus x$

A	B	x	A(n+1)	B(n+1)	y	T_A	T_B
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1
0	1	0	0	1	0	0	0
0	1	1	1	0	0	1	1
1	0	0	1	0	0	0	0
1	0	1	1	1	0	0	1
1	1	0	1	1	1	0	0
1	1	1	0	0	1	1	1

Για $x = 1$ το κύκλωμα λειτουργεί ως 2-bit 2-δικός μετρητής

Ανάλυση Κυκλώματος με Φλιπ-Φλοπ T



A	B	x	A(n+1)	B(n+1)	y	T_A	T_B
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1
0	1	0	0	1	0	0	0
0	1	1	1	0	0	1	1
1	0	0	1	0	0	0	0
1	0	1	1	1	0	0	1
1	1	0	1	1	1	0	0
1	1	1	0	0	1	1	1

Για $x = 1$ το κύκλωμα λειτουργεί ως 2-bit 2-δικός μετρητής

Ελαχιστοποίηση και Κωδικοποίηση Καταστάσεων

- *Ανάλυση ακολουθιακού κυκλώματος:*
  Από το λογικό του διάγραμμα καταλήγω στον πίνακα καταστάσεων ή/και στο λογικό διάγραμμα καταστάσεών του
- *Σχεδίαση ακολουθιακού κυκλώματος:*
  Από ένα σύνολο προδιαγραφών λειτουργίας καταλήγω στο σχετικό λογικό διάγραμμα
- Συγκεκριμένες ιδιότητες των ακολουθιακών κυκλωμάτων δύναται να χρησιμοποιηθούν για την απλοποίηση ενός προς σχεδίαση κυκλώματος
- Σκοπός είναι η μείωση του πλήθος των πυλών και των FFs του

Ελαχιστοποίηση Καταστάσεων

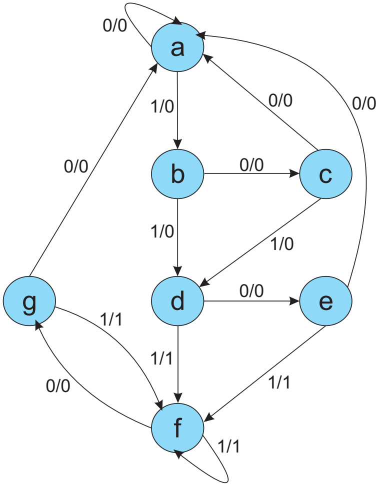
- Η ελαχιστοποίηση καταστάσεων αποσκοπεί στη μείωση του πλήθους των FFs, σε ένα ακολουθιακό κύκλωμα
- Η διαδικασία της ελαχιστοποίησης καταστάσεων ενός ακολουθιακού κυκλώματος δεν αλλάζει τη λειτουργία του

Ελαχιστοποίηση Καταστάσεων

Για την αναπαράσταση 2^n καταστάσεων απαιτούνται n FFs$\longrightarrow$
Η μείωση του αριθμού των καταστάσεων ενδέχεται να συντελέσει στη μείωση του απαιτούμενου πλήθους από FFs

Παράδειγμα Ελαχιστοποίησης Καταστάσεων

- Έστω το διάγραμμα καταστάσεων που συνοψίζει την προδιαγραφή ενός κυκλώματος
- Έστω a η αρχική κατάσταση
- Έστω ότι εισάγω στο κύκλωμα την ακολουθία από bits: 0 1 0 1 0 1 1 0 1 0 0



Παράδειγμα Ελαχιστοποίησης Καταστάσεων

Τότε, η ακολουθία καταστάσεων κι εξόδων είναι:

State	a	a	b	c	d	e	f	f	g	f	g	a
Input	0	1	0	1	0	1	1	0	1	0	0	
Output	0	0	0	0	0	1	1	0	1	0	0	

Ο πίνακας καταστάσεων του κυκλώματος είναι:

Current State	Next State		Output	
	x=0	**x=1**	**x=0**	**x=1**
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

Παράδειγμα Ελαχιστοποίησης Καταστάσεων

- Κυκλώματα τα οποία για τις ίδιες ακολουθίες εισόδων παράγουν ίδιες ακολουθίες εξόδων κι αυτό για κάθε δυνατή ακολουθία εισόδων, θεωρούνται *ισοδύναμα*
- Σε ένα ακολουθιακό κύκλωμα, δύο καταστάσεις θεωρούνται *ισοδύναμες* αν για κάθε μέλος του συνόλου δυνατών εισόδων δίνουν ακριβώς την ίδια έξοδο και στέλνουν το κύκλωμα στην ίδια ή σε ισοδύναμη κατάσταση

Current State	Next State		Output	
	x=0	**x=1**	**x=0**	**x=1**
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

Παράδειγμα Ελαχιστοποίησης Καταστάσεων

Διαγράφοντας τη g κι αντικαθιστώντας την από την e:

Current State	Next State		Output	
	x=0	x=1	x=0	x=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

Τελικά, διαγράφοντας την f κι αντικαθιστώντας την από τη d:

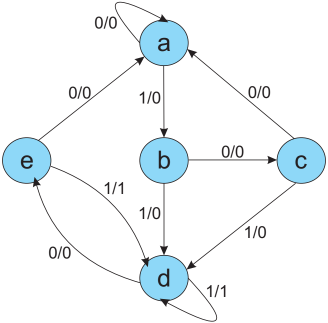
Current State	Next State		Output	
	x=0	x=1	x=0	x=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1

Παράδειγμα Ελαχιστοποίησης Καταστάσεων

Η ακολουθία καταστάσεων κι εξόδων είναι ίδια με την αρχική:

State	a	a	b	c	d	e	d	d	e	d	e	a
Input	0	1	0	1	0	1	1	0	1	0	0	0
Output	0	0	0	0	0	1	1	0	1	0	0	0

Το απλοποιημένο διάγραμμα καταστάσεων είναι:

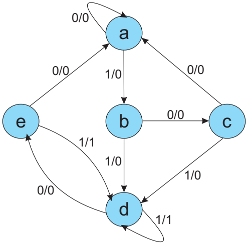


Current State	Next State		Output	
	x=0	**x=1**	**x=0**	**x=1**
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1

Κωδικοποίηση Καταστάσεων

- Η υλοποίηση ενός ακολουθιακού κυκλώματος απαιτεί την απόδοση διακριτών 2-δικών τιμών στις διαφορετικές καταστάσεις του $\longrightarrow$ *Κωδικοποίηση Καταστάσεων*
- Για την κωδικοποίηση m καταστάσεων χρειάζονται n-bit 2-δικές λέξεις, όπου $2^n \geqslant m$
- Οι αχρησιμοποίητες 2-δικές λέξεις θεωρούνται συνθήκες αδιαφορίας και χρησιμοποιούνται συχνά για την εύρεση απλούστερων υλοποιήσεων ακολουθιακών κυκλωμάτων

Παράδειγμα Κωδικοποίησης Καταστάσεων



- Χρειάζονται 3 bits για την κωδικοποίηση των καταστάσεων
- Η ανάθεση κώδικα ενός ενεργού οδηγεί σε απλούστερη αποκωδικοποίηση των εξόδων και της επόμενης κατάστασης

State	Binary Coding	Gray Coding	One-hot Coding
a	000	000	00001
b	001	001	00010
c	010	011	00100
d	011	010	01000
e	100	110	10000

Κωδικοποίηση Καταστάσεων

- Κατά τη σχεδίαση κυκλωμάτων, οι αχρησιμοποίητοι κωδικοί αριθμοί θεωρούνται συνθήκες αδιαφορίας και τους χρησιμοποιώ έτσι ώστε να καταλήξω στην απλούστερη δυνατή υλοποίηση
- Η χρήση του κώδικα Gray διευκολύνει την κατασκευή του χάρτη Karnaugh
- Για την ανάθεση ενός ενεργού χρησιμοποιείται μεν ένα FF ανά κατάσταση, αλλά οδηγούμε δε σε απλούστερο συνδυαστικό κύκλωμα για την παραγωγή των εξόδων κι εισόδων των FF
- Η 2-δική μορφή του πίνακα καταστάσεων χρησιμοποιείται για τη δημιουργία του συνδυαστικού τμήματος του ακολουθιακού κυκλώματος, δηλαδή για την παραγωγή των εξόδων του και την προετοιμασία των επόμενων καταστάσεών του

Διαδικασία Σχεδίασης

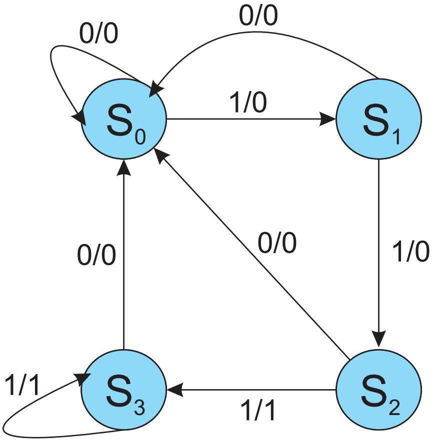
- Βήμα 1: Δημιουργώ το διάγραμμα κατάστασης του κυκλώματος με βάση την επιθυμητή λειτουργία του
- Βήμα 2: Ελαχιστοποιώ το πλήθος των καταστάσεων κι έπειτα τις κωδικοποιώ
- Βήμα 3: Εξάγω το 2-δικά κωδικοποιημένο πίνακα καταστάσεων
- Βήμα 4: Επιλέγω το είδος από FF που θα χρησιμοποιήσω
- Βήμα 5: Υπολογίζω τις απλοποιημένες εξισώσεις εισόδων των FFs κι εξόδων του κυκλώματος
- Βήμα 6: Σχεδιάζω το συνδυαστικό μέρος του κυκλώματος και το λογικό διάγραμμα του συνολικού ακολουθιακού κυκλώματος

Ανιχνευτής Ακολουθίας με Φλιπ-Φλοπ D

Να σχεδιαστεί κύκλωμα εντοπισμού ακολουθίας τριών ή περισσοτέρων διαδοχικών 1 σε μια σειριακή ακολουθία από bits

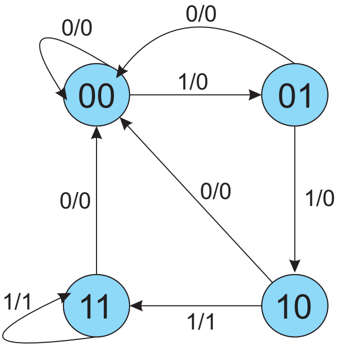
Δημιουργία διαγράμματος καταστάσεων

- Χρειάζομαι 4 καταστάσεις, έστω τις S_0, S_1, S_2 και S_3
- Έστω S_0 η αρχική κατάσταση
- Η μία έξοδος του κυκλώματος είναι 1 όταν έχω μια ακολουθία από τουλάχιστον τρία 1



Ανιχνευτής Ακολουθίας με Φλιπ-Φλοπ D

- Κωδικοποιώ τις καταστάσεις:
  $S_0 \rightarrow 00$
  $S_1 \rightarrow 01,\ S_2 \rightarrow 10,\ S_3 \rightarrow 11$



- Εξάγω τον 2-δικά κωδικοποιημένο πίνακα καταστάσεων

Current State		Input	Next State		Output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	1	1	1

$$A(n+1) = \sum(3,5,7)$$
$$B(n+1) = \sum(1,5,7)$$
$$y = \sum(5,7)$$

Ανιχνευτής Ακολουθίας με Φλιπ-Φλοπ D

A \ Bx	00	01	11	10
0	0	0	1	0
1	0	1	1	0

B: columns 11, 10; A: row 1; x: columns 01, 11

$$A(n+1) = Ax + Bx$$

A \ Bx	00	01	11	10
0	0	1	0	0
1	0	1	1	0

B: columns 11, 10; A: row 1; x: columns 01, 11

$$B(n+1) = Ax + B'x$$

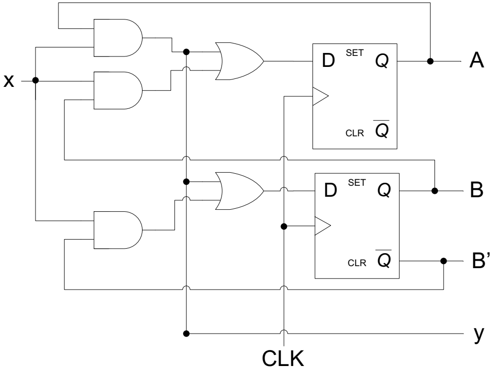
A \ Bx	00	01	11	10
0	0	0	0	0
1	0	1	1	0

B: columns 11, 10; A: row 1; x: columns 01, 11

$$y = Ax$$

Ανιχνευτής Ακολουθίας με Φλιπ-Φλοπ D

$$D_A = Ax + Bx$$
$$D_B = Ax + B'x$$
$$y = Ax$$



Σχεδίαση Κυκλωμάτων με Άλλα Φλιπ-Φλοπ

- Όταν χρησιμοποιώ D FFs, οι εξισώσεις εισόδων τους προκύπτουν άμεσα από τις τιμές των επόμενων καταστάσεων τους
- Στην περίπτωση των T και JK FFs, οι εξισώσεις εισόδων τους προκύπτουν έμμεσα από τις τιμές της παρούσας και της επόμενης κατάστασης
- Ο *πίνακας διέγερσης* ενός FF περιλαμβάνει τις απαιτούμενες εισόδους του για καθεμιά από όλες τις δυνατές αλλαγές κατάστασής του

Q(n-1)	Q(n)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Q(n-1)	Q(n)	T
0	0	0
0	1	1
1	0	1
1	1	0

Σύνθεση με Φλιπ-Φλοπ JK

Να σχεδιαστεί κύκλωμα με JK FFs του οποίου η λειτουργία περιγράφεται από τον ακόλουθο πίνακα καταστάσεων:

Current State		Input	Next State		Flip-Flop Inputs			
A	B	x	A	B	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	0	0	1	X	X	1
0	1	1	1	0	0	X	X	0
1	0	0	0	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	0	0	X	0	X	0
1	1	1	1	1	X	1	X	1

Το απαιτούμενο κύκλωμα θα περιέχει δύο JK FFs, με τις εισόδους του καθενός να προκύπτουν έμμεσα, χρησιμοποιώντας τον πίνακα διέγερσης του JK FF, από τις τιμές της παρούσας και της επόμενης του κατάστασης

Σύνθεση με Φλιπ-Φλοπ JK

A \ Bx	00	01	11	10
0	0	0	0	1
1	X	X	X	X

$$J_A = Bx'$$

A \ Bx	00	01	11	10
0	0	1	X	X
1	0	1	X	X

$$J_B = x$$

A \ Bx	00	01	11	10
0	X	X	X	X
1	0	0	1	0

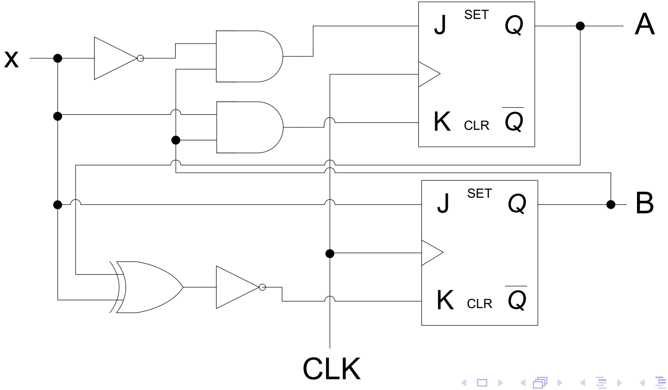
$$K_A = Bx$$

A \ Bx	00	01	11	10
0	X	X	0	1
1	X	X	1	0

$$K_B = (A \oplus x)'$$

Σύνθεση με Φλιπ-Φλοπ JK

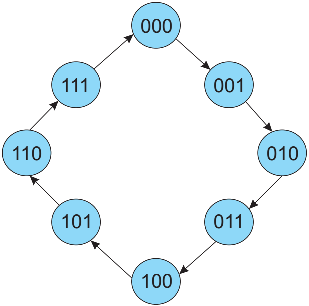
- Από τις αλγεβρικές εκφράσεις Boole των εισόδων των FFs κατασκευάζω το συνδυαστικό τμήμα του ακολουθιακού κυκλώματος
- Έξοδοι του ζητούμενου κυκλώματος είναι οι έξοδοι των JK FFs



3-bit 2-δικός Μετρητής με Φλιπ-Φλοπ T

- Ο n-bit 2-δικός μετρητής αποτελείται από n FFs και μετρά από το 0 έως το $2^n - 1$
- Είσοδος του κυκλώματος είναι το ρολόι κι έξοδοί του οι έξοδοι των FFs

Current State			Next State			Flip-Flop Inputs		
A_2	A_1	A_0	A_2	A_1	A_0	T_{A2}	T_{A1}	T_{A0}
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1



3-bit 2-δικός Μετρητής με Φλιπ-Φλοπ T

$A_2 \backslash A_1A_0$	00	01	11	10
0	0	0	1	0
1	0	0	1	0

$$T_{A_2} = A_0 A_1$$

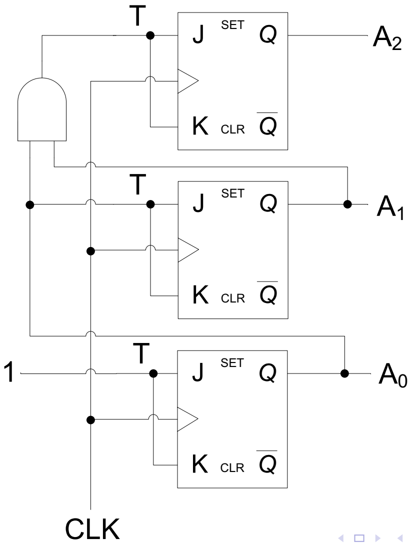
$A_2 \backslash A_1A_0$	00	01	11	10
0	0	1	1	0
1	0	1	1	0

$$T_{A_1} = A_0$$

$A_2 \backslash A_1A_0$	00	01	11	10
0	1	1	1	1
1	1	1	1	1

$$T_{A_0} = 1$$

3-bit 2-δικός Μετρητής με Φλιπ-Φλοπ T



Τέλος Θεματικής Ενότητας

Ευχαριστώ για την προσοχή σας

Γεώργιος Χ. Αλεξανδρόπουλος

e-mail: alexandg@uop.gr
URL: http://users.iit.demokritos.gr/~alexandg

eclass: http://eclass.uop.gr/courses/TST289/