



Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών

Ψηφιακή Σχεδίαση

Ενότητα 10: Καταχωρητές & Μετρητές

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής
Υπολογιστών

<http://arch.icte.uowm.gr/mdasyg>



Άδειες Χρήσης

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



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Σκοπός της ενότητας

- Να γίνει εισαγωγή και ανάλυση στους καταχωρητές και στους μετρητές.



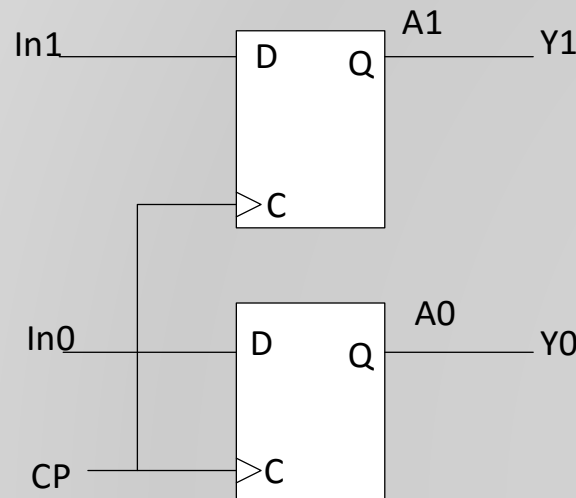
Καταχωρητές & Μετρητές

- Ένας n -bit καταχωρητής είναι ένα σύνολο από n flip-flops, ικανό να αποθηκεύσει n bits δυαδικής πληροφορίας.
- Με επιπρόσθετες συνδυαστικές πύλες, ο καταχωρητής μπορεί να εκτελέσει λειτουργίες επεξεργασίας δεδομένων (data-processing).
- Ένας μετρητής είναι ένας καταχωρητής που έχει μια προκαθορισμένη σειρά καταστάσεων, βάση της εφαρμογής των παλμών του ρολογιού.



Παράδειγμα: καταχωρητής 2bit (1)

1. Πόσες καταστάσεις υπάρχουν;
 2. Πόσοι συνδυασμοί εισόδων/εξόδων;
 3. Ποιά είναι η συνάρτηση εξόδου;
 4. Ποια είναι η συνάρτηση της επόμενης κατάστασης;
 5. Είναι Moore ή Mealy;
- Ποιά είναι η απάντηση για τα 1 και 2 (πιο πάνω) για έναν καταχωρητή με n-bits;



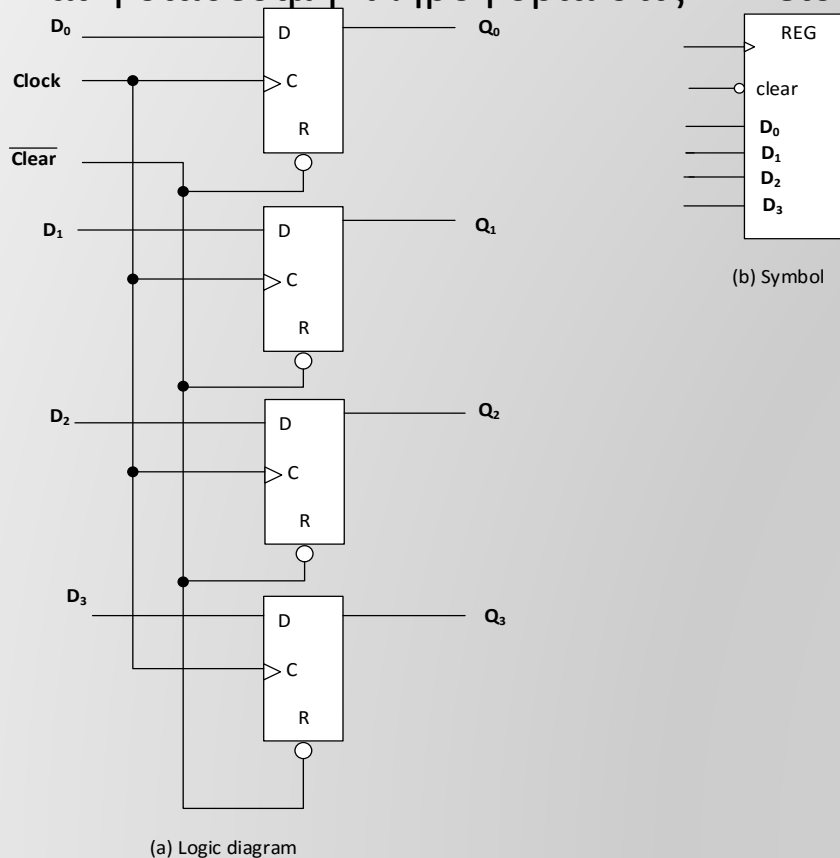
Παράδειγμα: καταχωρητής 2bit (2)

Παρούσα Κατάσταση		Επόμενη Κατάσταση A1 (t + 1) A0(t + 1) Για In1 In0 =				Έξοδος (= A1 A0)	
A1	A0	00	01	10	11	Y1	Y0
0	0	00	01	10	11	0	1
0	1	00	01	10	11	0	1
1	0	00	01	10	11	1	0
1	1	00	01	10	11	1	1



Παράδειγμα: καταχωρητής 4bit

- Η κοινή είσοδος Clock πυροδοτεί όλα τα flip-flops στην θετική ακμή κάθε παλμού, και η διαθέσιμη πληροφορία στις 4 D-εισόδους μεταφέρεται στον καταχωρητή.



Logic diagram: Λογικό διάγραμμα
Symbol: Σύμβολο



Μνήμη Καταχωρητών (1)

- Προσδοκίες:
 - Ένας καταχωρητής πρέπει να μπορεί να αποθηκεύει πληροφορίες για πολλαπλές χρονικές περιόδους.
 - Η «αποθήκευση» ή «φόρτωση» πληροφοριών πρέπει να ελέγχεται από κάποιο σήμα.
- Πραγματικότητα:
 - Ο προηγούμενος καταχωρητής φορτώνει πληροφορίες **σε κάθε χρονική περίοδο του ρολογιού** (clock cycle).



Μνήμη Καταχωρητών (2)

- Λύση:
 - Χρήση του σήματος για μπλοκάρισμα του καταχωρητή από το ρολόι ή
 - χρήση του σήματος για έλεγχο ανάδρασης (feedback control) από την έξοδο του καταχωρητή πίσω στις εισόδους του ή
 - χρήση SR ή JK flip-flops τα οποία κρατούν την παρούσα κατάσταση τους για είσοδο (0, 0).
- Load: συχνή ονομασία για το σήμα που ελέγχει την αποθήκευση και φόρτωση καταχωρητών.
 - Load = 1: Φορτώνει τις τιμές των εισόδων.
 - Load = 0: αποθηκεύει τις τιμές του καταχωρητή.

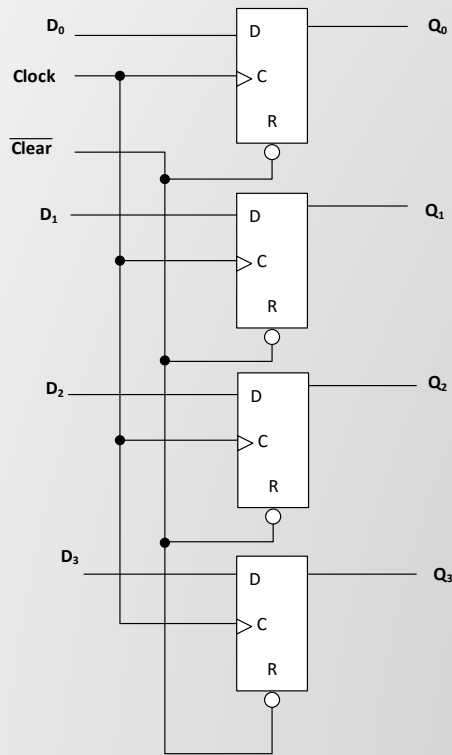


Αποφυγή ελέγχου του ρολογιού

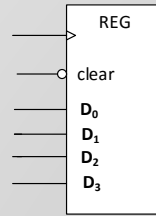
- Μια λύση για να παραμείνουν αναλοίωτα τα περιεχόμενα του καταχωρητή είναι μια πύλη επίτρεψης στο ρολόι.
- Η εισαγωγή λογικών πυλών στη γραμμή του ρολογιού, προκαλεί άνισες καθυστερήσεις διάδοσης.
- Για να είναι συγχρονισμένο το κύκλωμα πρέπει να εξασφαλίσουμε ότι όλοι οι παλμοί φτάνουν την ίδια στιγμή.



Καταχωρητής 4bit

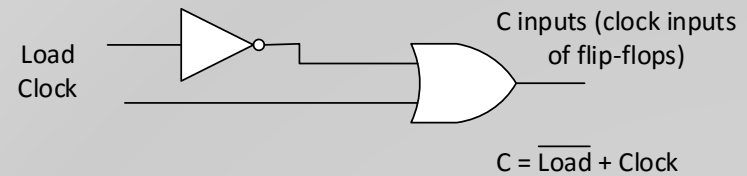


(a) Logic diagram



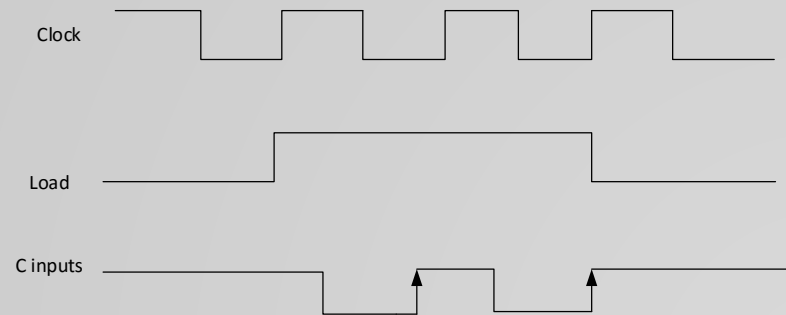
(b) Symbol

Το Clock-gating δεν είναι καλή λύση



(c) Load control input

Τα FFs είναι ευαίσθητα σε αλλαγές μόνο όταν Load=1



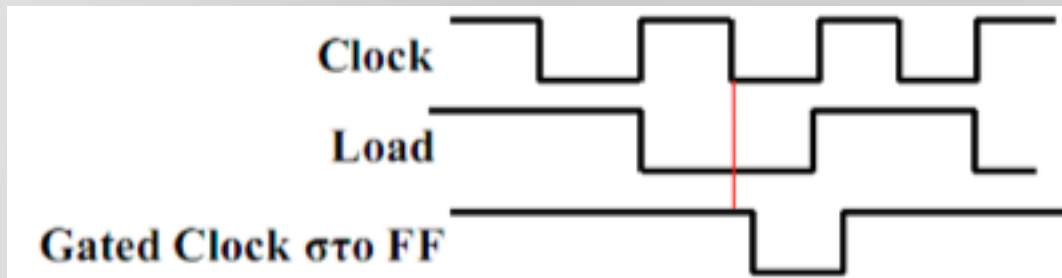
(d) Timing diagram



Καταχωρητές με clock gating

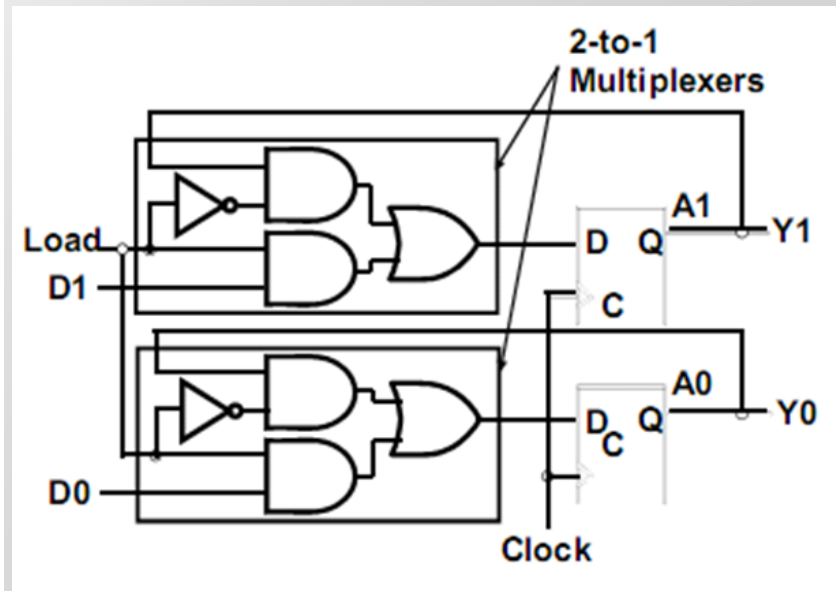
- Το σήμα **Load** χρησιμοποιείται για ενεργοποίηση του σήματος του ρολογιού όταν είναι 1, και απενεργοποιεί το ρολόι όταν είναι 0.
- Παράδειγμα προβλήματος: Για Flip-flops θετικής ακμοπυροδότησης ή αρνητικού-επίπεδου πυροδότησης:
- Ποιό είναι το πρόβλημα;

Απόκλιση του gated clock από το πραγματικό ρολόι λόγω της καθυστέρησης από τις επιπρόσθετες πύλες (clock skew).

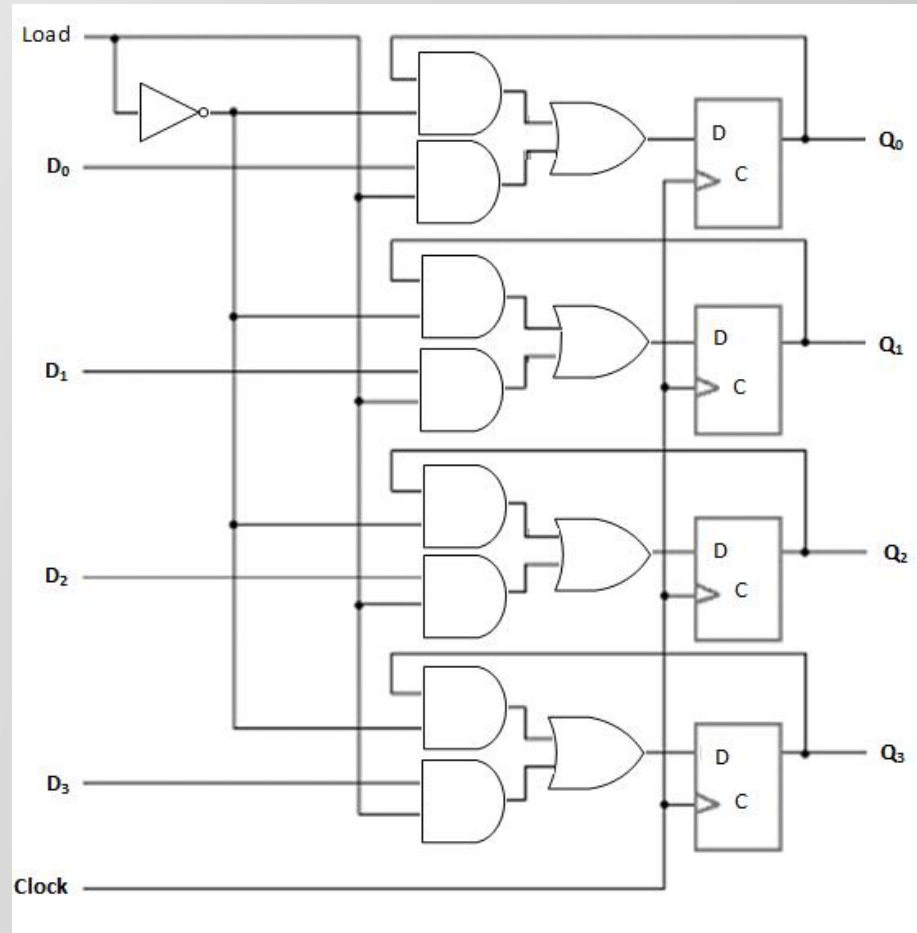


Καταχωρητές παράλληλης φόρτωσης με έλεγχο ανάδρασης (1)

- Πιο αξιόπιστος σχεδιασμός:
 - Το ρολόι δουλεύει ασταμάτητα (στα FFs), και
 - Επιλεκτική χρήση του Load για αλλαγή των περιεχομένων του καταχωρητή.
- Παράδειγμα: καταχωρητής 2-bit:
- Για Load = 0 κράτηση παρούσας κατάστασης.
- Για Load = 1 φότωση τιμών εισόδου, βάση του ρολογιού.
- Πιο σύνθετος σχεδιασμός από clock gating, αλλά ελεύθερος από προβλήματα χρονισμού.

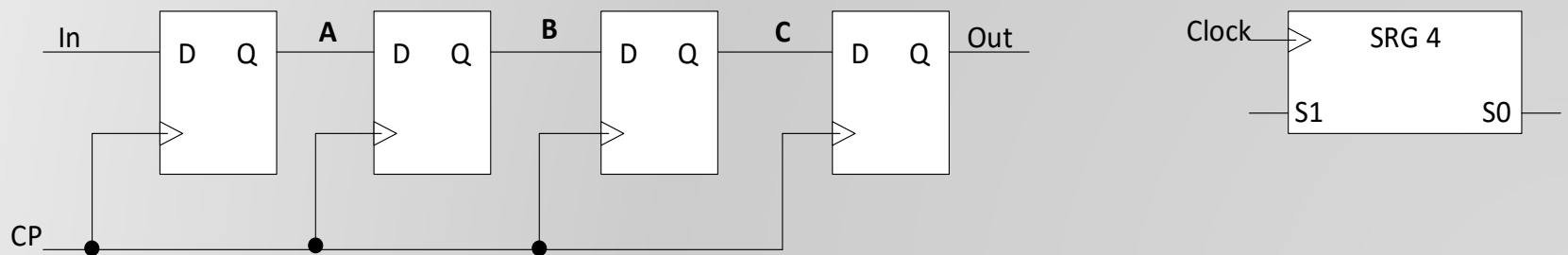


Καταχωρητής Παράλληλης φόρτωσης 4 bit



Καταχωρητές Ολίσθησης (1)

- Οι καταχωρητές ολίσθησης μεταφέρουν πληροφορίες «πλάγια» μέσα στον καταχωρητή, προς την περισσότερο σημαντική (MSB) ή λιγότερο σημαντική (LSB) θέση
- Στην πιο απλή περίπτωση, ο καταχωρητής ολίσθησης είναι απλά ένα σύνολο από D flip-flops ενωμένα διαδοχικά ως ακολούθως:

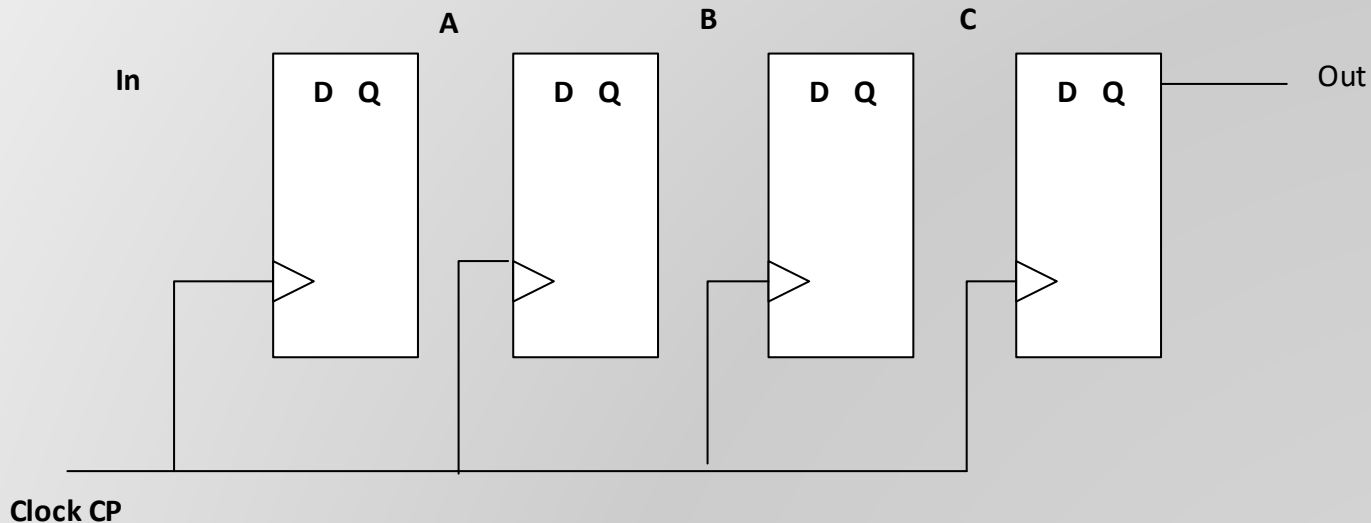


Καταχωρητές Ολίσθησης (2)

- Είσοδος δεδομένων, **In**, ονομάζεται **σειριακή είσοδος** ή **είσοδος δεξιάς ολίσθησης**.
- Έξοδος δεδομένων, **Out**, συχνά ονομάζεται **σειριακή έξοδος**.
- Το διάνυσμα (**A, B, C, Out**) ονομάζεται η **παράλληλη έξοδος**.



Καταχωρητές Ολίσθησης (3)



- Η συμπεριφορά του σειριακού καταχωρητή ολίσθησης δίνεται στον πίνακα απέναντι.
- T0 είναι η κατάσταση ακριβώς πριν την εμφάνιση του 1^{ου} παλμού του ρολογιού.
- T1 έρχεται μετά τον πρώτο παλμό και πριν τον δεύτερο.
- Αρχικές άγνωστες καταστάσεις δηλώνονται με “?”.



Καταχωρητές Ολίσθησης (4)

- Συμπληρώστε τις 3 τελευταίες γραμμές του πίνακα.

CP	In	A	B	C	Out
T0	0	?	?	?	?
T1	1	0	?	?	?
T2	1	1	0	?	?
T3	0	1	1	0	?
T4	1				
T5	1				
T6	1				



Σειριακή Μεταφορά (1)

- Ένα ψηφιακό σύστημα λειτουργεί σειριακά (in serial mode) όταν σε κάθε παλμό του ρολογιού, **ένα bit** μόνο των δυαδικών πληροφοριών μεταφέρεται και υπόκειται σε επεξεργασία.



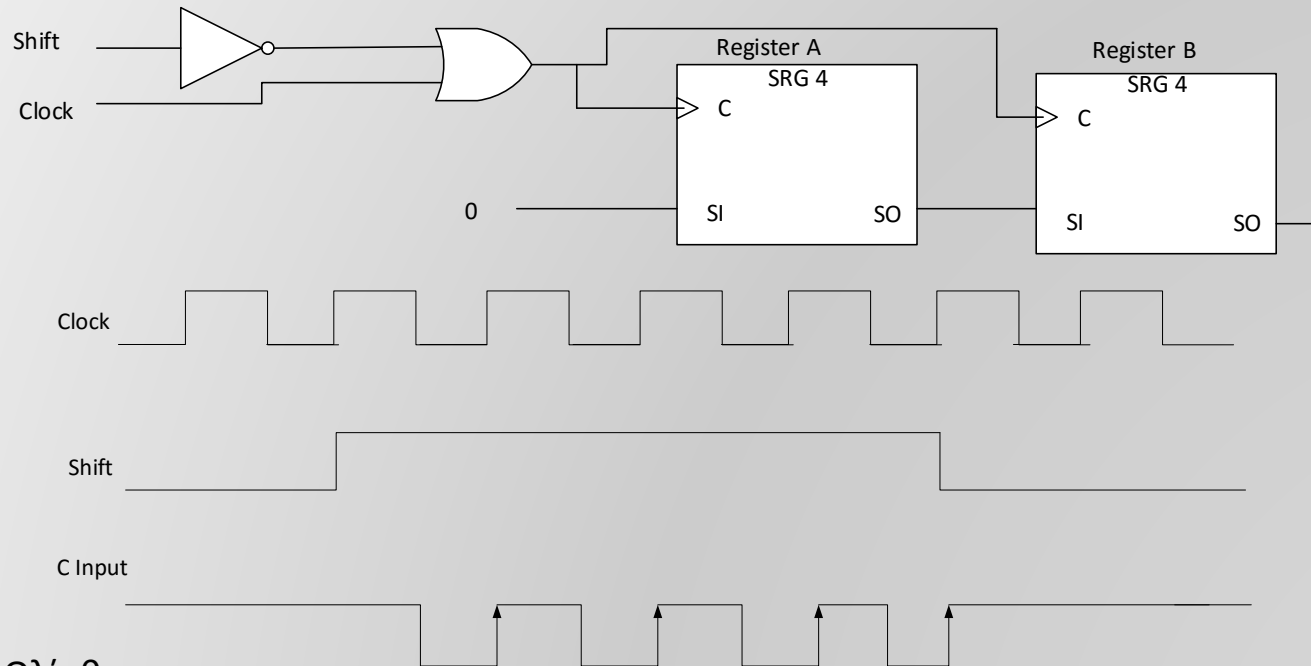
Σειριακή Μεταφορά (2)

- Σε παράλληλα μονοπάτια υπάρχει πρόβλημα συγχρονισμού επειδή υπάρχουν διαφορετικές καθυστερήσεις σε κάθε κάθε αγωγό.
- Το πρόβλημα οξύνεται με την αύξηση της ταχύτητας μετάδοσης bit.
- Ο συγχρονισμός πολύ πιο εύκολος με ένα μόνο καλώδιο με σταθερή καθυστέρηση για όλα τα bit.
- Με αύξηση κόστους μπορούμε να έχουμε ομοιόμορφη καθυστέρηση σε παράλληλες οδεύσεις.
- Παράλληλες οδεύσεις απαιτούν πολλαπλάσιο χώρο από σειριακές οδεύσεις bit (ευελιξία).



Σειριακή Μεταφορά Δεδομένων

- Σειριακή μεταφορά δεδομένων από έναν καταχωρητή A σε έναν καταχωρητή B. Απαιτείται μία περίοδος ρολογιού για κάθε bit.

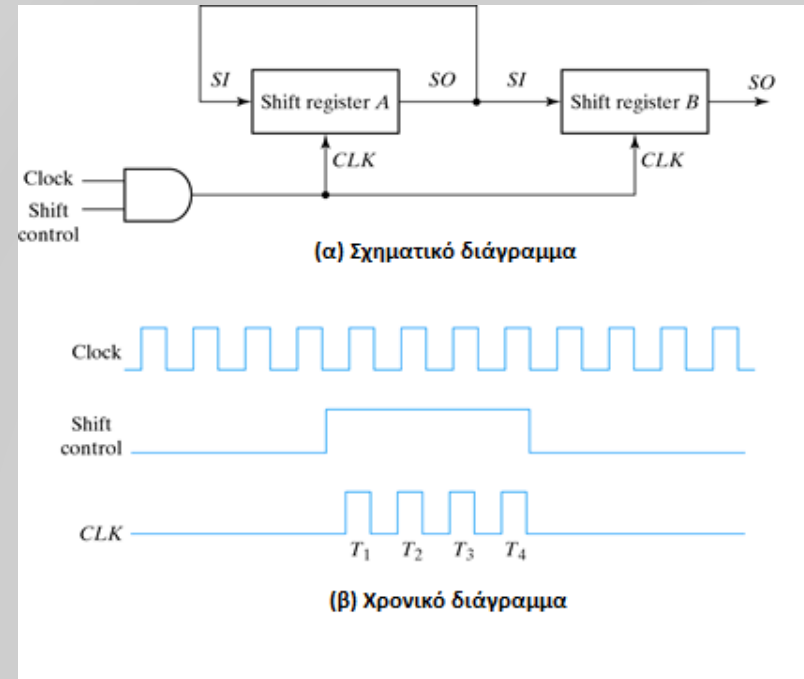


Shift: Ολίσθηση
Register: Καταχωρητής
Clock: Ρολόι
C input: Είσοδος C



Σειριακή μεταφορά από τον A στον B

- Υποθέτουμε 4bit καταχωρητές.
- Οι πληροφορίες επανατροφοδοτούνται στον ίδιο καταχωρητή.
- Κάθε θετική ακμή του παλμού προκαλεί ολίσθηση στους δύο καταχωρητές.



Shift control: Ελεγχόμενη ολίσθηση

Shift register: Καταχωρητής ολίσθησης



Σειριακή Πρόσθεση (1)

- Οι δύο δυαδικοί προσθετέοι αποθηκεύονται σειριακά σε δύο καταχωρητές A και B. Πόσος χρόνος χρειάζεται για να φορτωθούν οι αριθμοί;
- Τα bits προσθέτονται ανά ζεύγος κάθε χρονική στιγμή, μέσω ενός πλήρη αθροιστή (full-adder circuit). Πόσος χρόνος χρειάζεται για την πρόσθεση;
- Το carry out του πλήρη αθροιστή μεταφέρεται σε ένα D flip-flop, του οποίου η έξοδος χρησιμοποιείται ως το carry in για το επόμενο ζεύγος των bits.
- Το sum bit στην έξοδο S του πλήρη αθροιστή μεταφέρεται πίσω στον καταχωρητή A (συσσωρευτής).



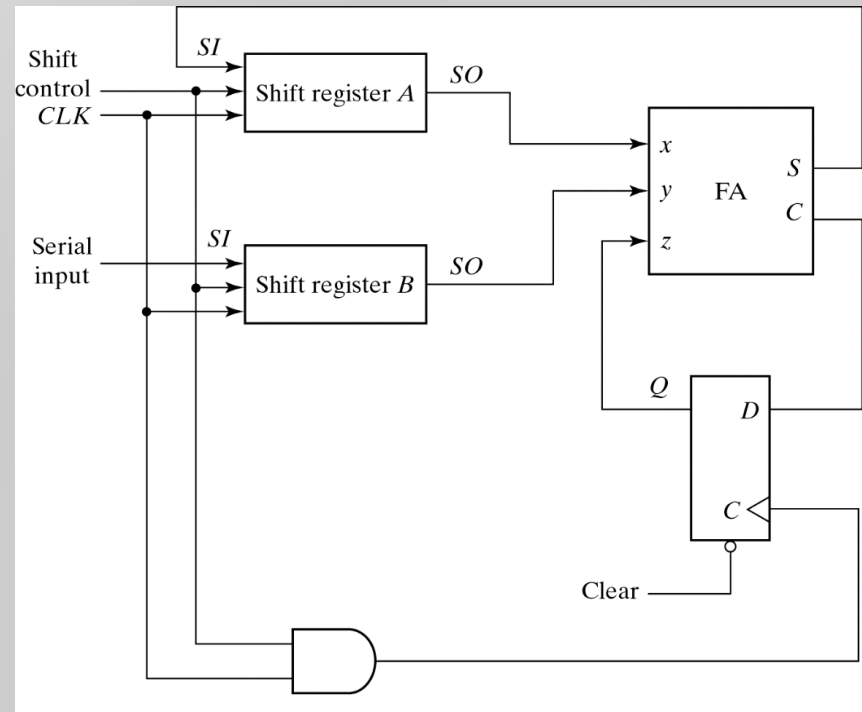
Σειριακή Πρόσθεση (2)

- Πόσος χρόνος χρειάζεται για ολόκληρη τη διαδικασία;
- Σχεδιάστε το διάγραμμα, σε επίπεδο καταχωρητών (ή RTI-level = Register-Transfer Level)....



Διάγραμμα Σειριακού Αθροιστή

- Το αποτέλεσμα αποθηκεύεται στον A.
- Μπορούν να προστεθούν περισσότεροι αριθμοί.



Serial input: Σειριακή είσοδος

Shift control: Ελεγχόμενη ολίσθηση

Shift register: Καταχωρητής ολίσθησης



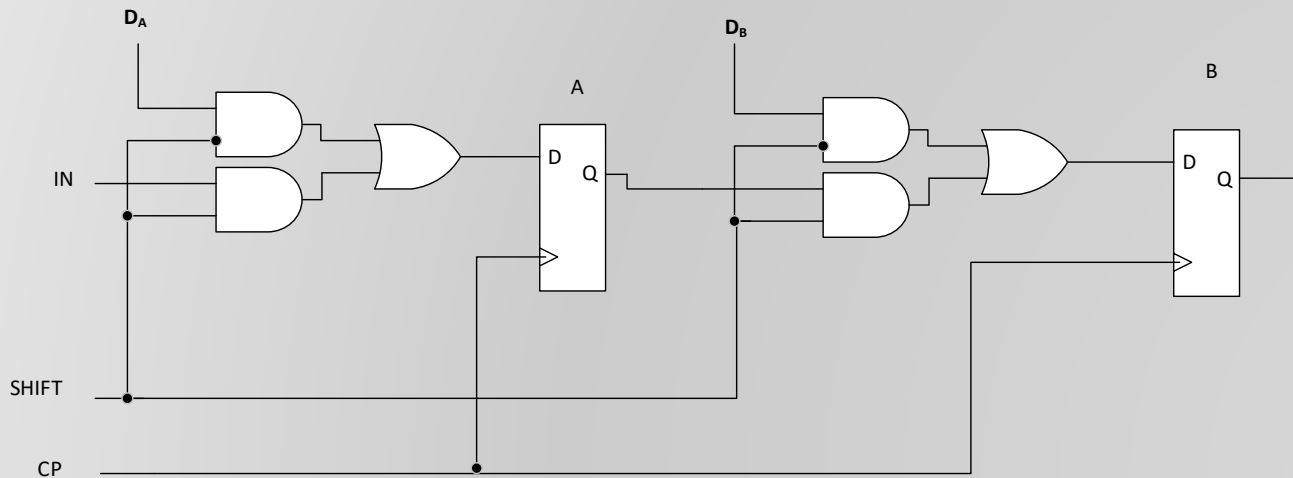
Σειριακή ή παράλληλη πρόσθεση

- Ο παράλληλος αθροιστής είναι ένα συνδυαστικό κύκλωμα, ενώ ο σειριακός ένα ακολουθιακό.
- Ο παράλληλος αθροιστής αποτελείται από n πλήρες 1-bit αθροιστές για προσθετέους των n -bit, ενώ ο σειριακός απαιτεί μόνο 1 πλήρη 1-bit αθροιστή.
- Το σειριακό κύκλωμα παίρνει n περιόδους του ρολογιού για να ολοκληρώσει, ενώ το παράλληλο 1.
- Συνοπτικά, ο παράλληλος αθροιστής είναι n φορές μεγαλύτερος του σειριακού σε χώρο, αλλά είναι και n φορές πιο γρήγορος.

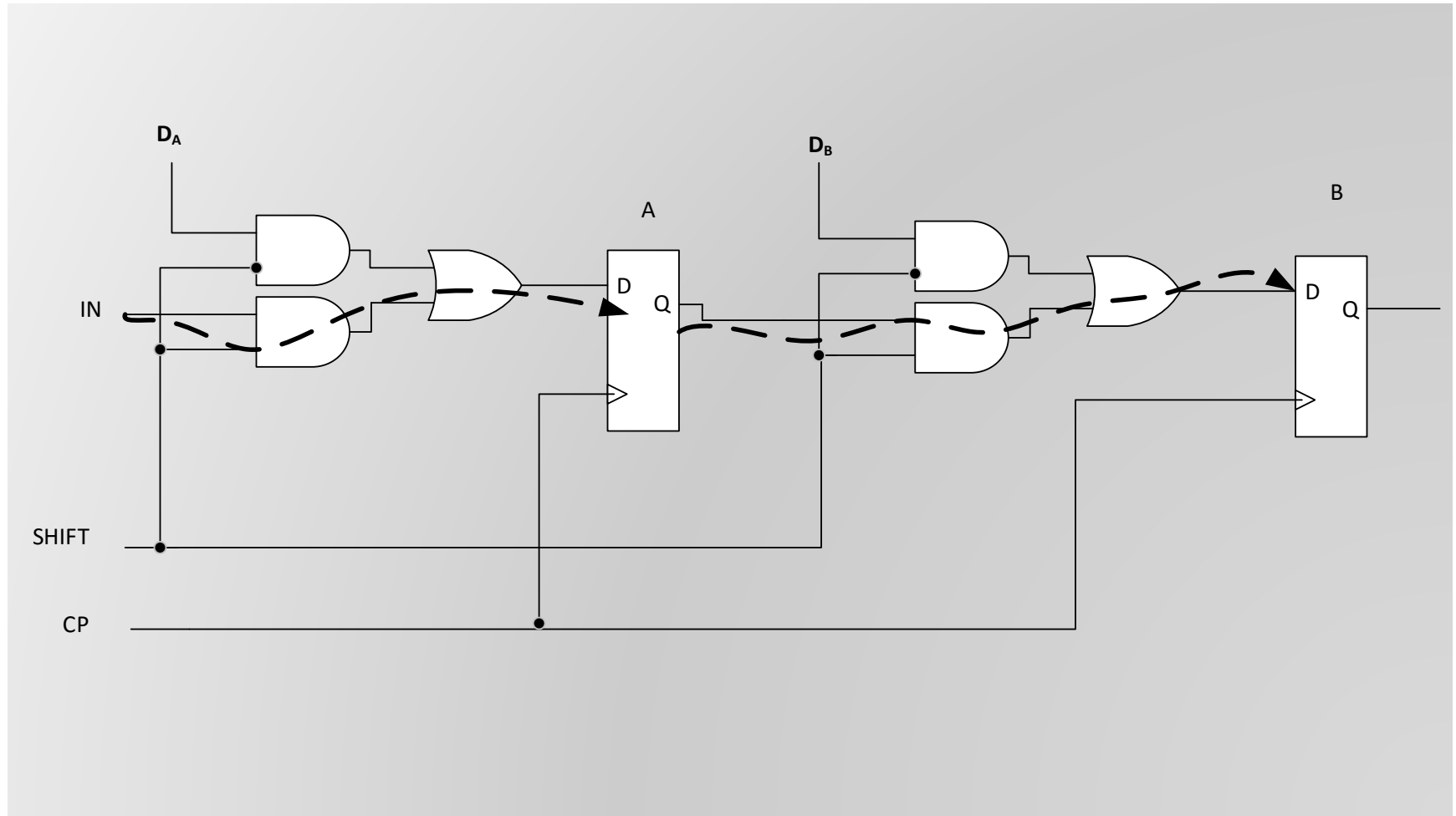


Καταχωρητές Ολίσθησης Παράλληλης Φόρτωσης (1)

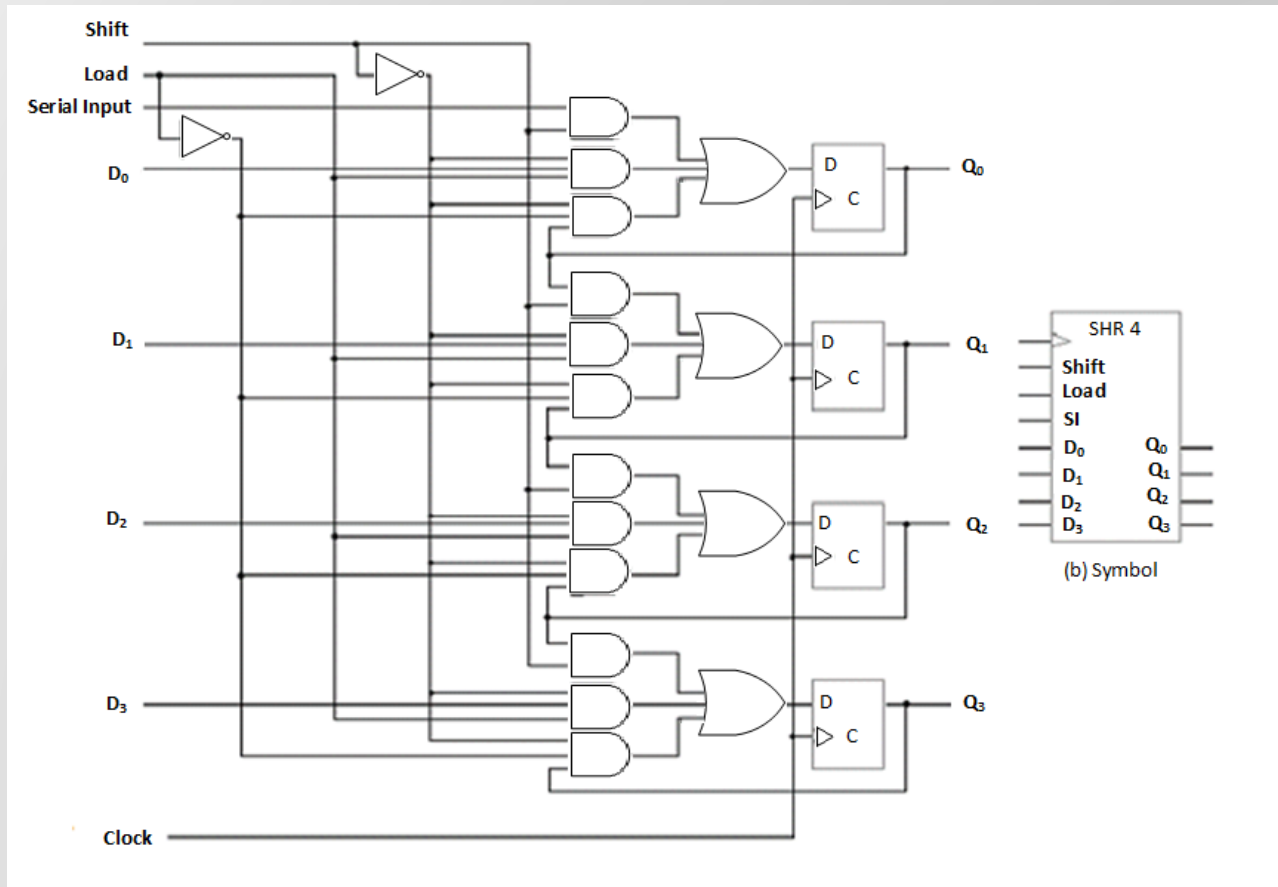
- Προσθέτοντας ένα MUX μεταξύ κάθε «στάδιο» του καταχωρητή ολίσθησης, δεδομένα μπορούν να αποθηκεύονται ή φορτώνονται με αξιόπιστο τρόπο.
- $SHIFT = 0$, τα A και B αντικαθιστούνται με τα δεδομένα στις γραμμές D_A και D_B (παράλληλη φόρτωση), αλλιώς τα δεδομένα ολισθαίνουν προς τα δεξιά στην κάθε περίοδο ρολογιού και φορτώνεται 1-bit (σειριακή φόρτωση).
- Προσθέτοντας και άλλα bits, κατασκευάζουμε καταχωρητή ολίσθησης παράλληλης φόρτωσης n-bit.



Καταχωρητές Ολίσθησης Παράλληλης Φόρτωσης (2)

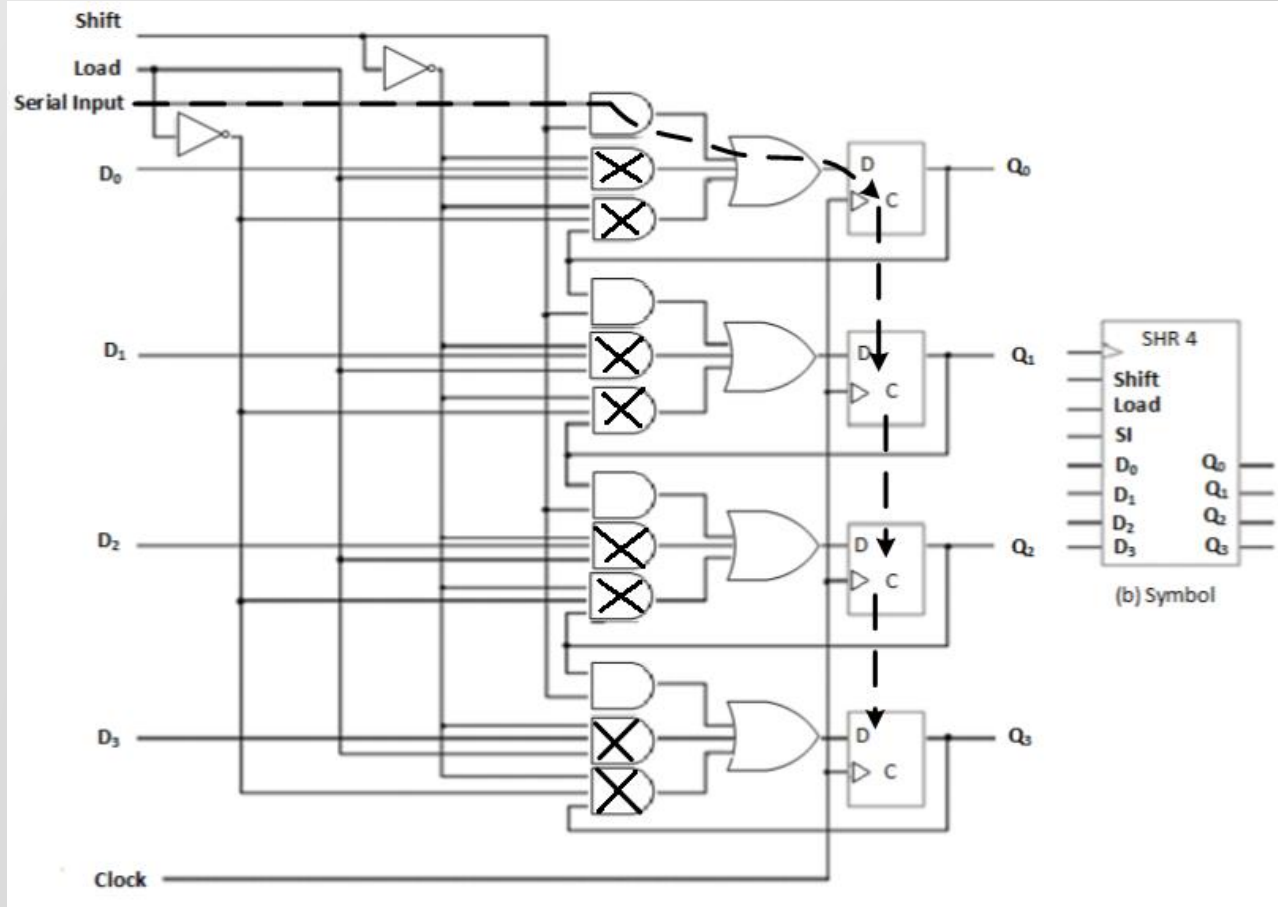


Καταχωρητές Ολίσθησης Παράλληλης Φόρτωσης (3)



Καταχωρητές Ολίσθησης Παράλληλης Φόρτωσης (4)

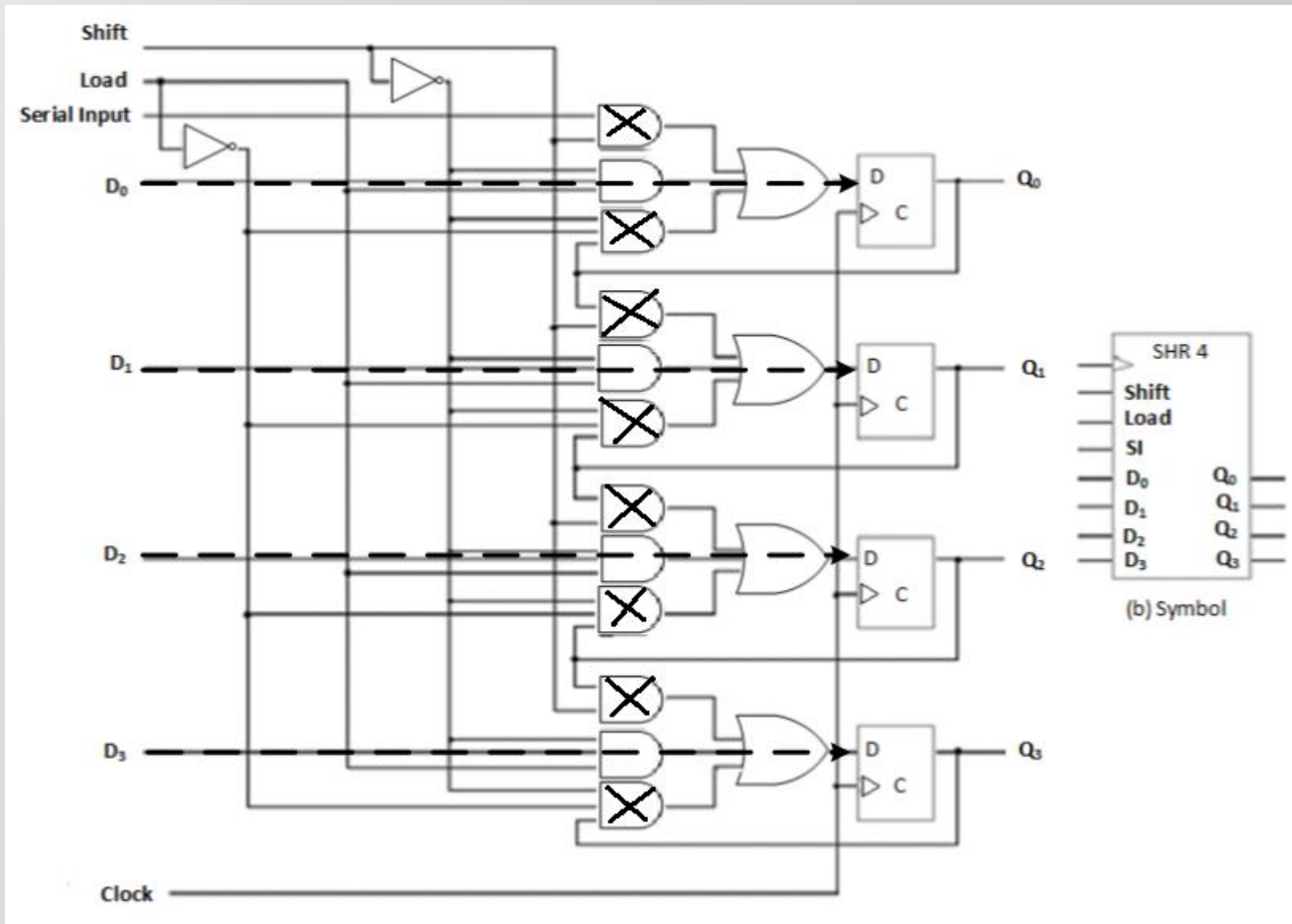
SHIFT = 1



Καταχωρητές Ολίσθησης Παράλληλης Φόρτωσης (5)

SHIFT = 0

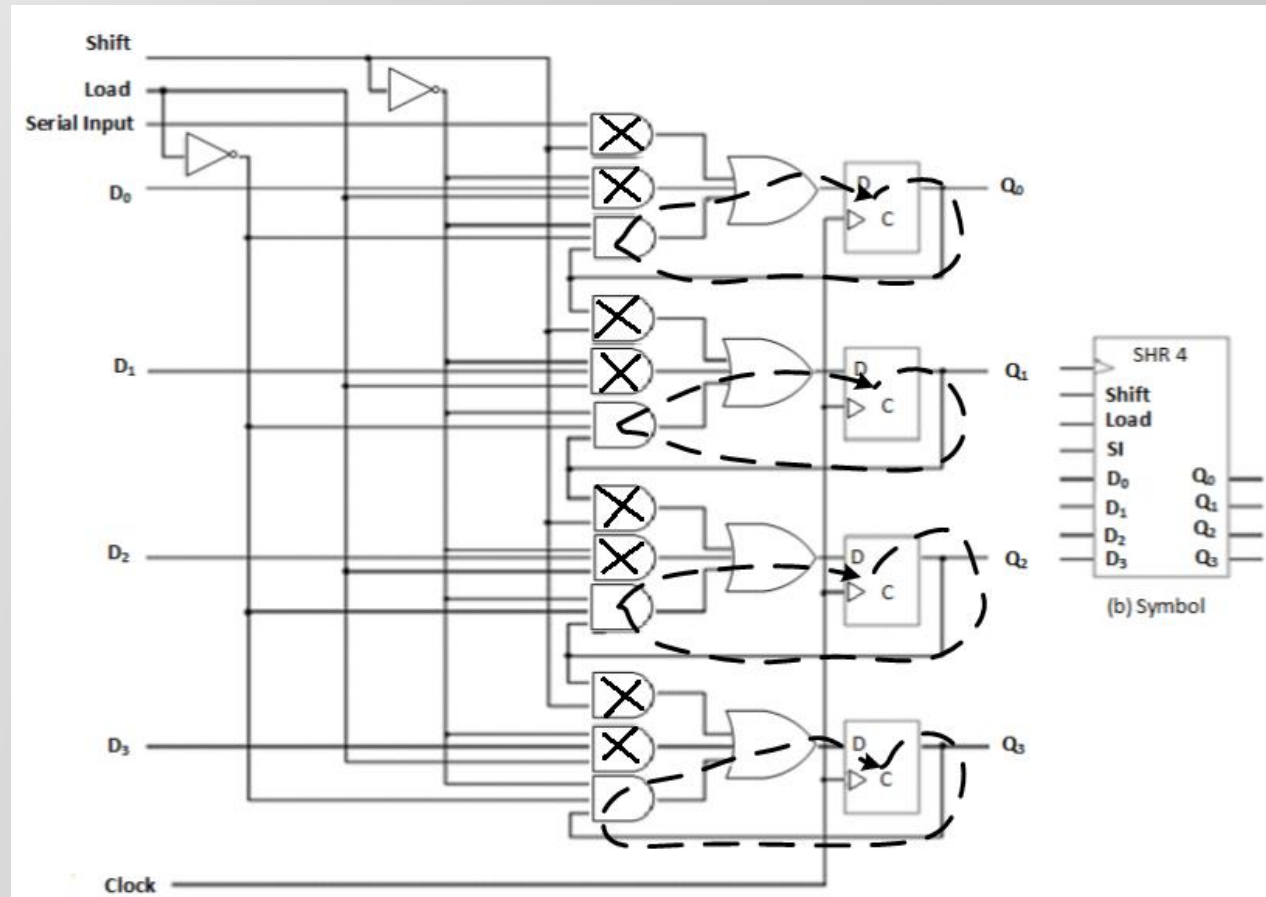
LOAD = 1



Καταχωρητές Ολίσθησης Παράλληλης Φόρτωσης (6)

SHIFT = 0

LOAD = 1



Καταχωρητές Ολίσθησης Παράλληλης Φόρτωσης (7)

- Μπορούμε να προσθέσουμε την λειτουργία “hold” για αποθήκευση/κράτηση της παρούσας πληροφορίας.

Shift	Load	Λειτουργία
0	0	Κράτηση παρούσας κατάστασης: $Q_0 \rightarrow Q_0, Q_1 \rightarrow Q_1, Q_2 \rightarrow Q_2, \dots$
0	1	Παράλληλη φόρτωση: $D_0 \rightarrow Q_0, D_1 \rightarrow Q_1, D_2 \rightarrow Q_2, \dots$
1	X	Ολίσθηση: $\text{Serial Input} \rightarrow Q_0 \rightarrow Q_1 \rightarrow Q_2 \dots$



Καταχωρητές ολίσθησης με Επιπρόσθετες Λειτουργίες

- Προσθέτοντας ένα MUX 4-εισόδων μπροστά από κάθε flip-flop σε ένα καταχωρητή ολίσθησης, μπορούμε να υλοποιήσουμε ένα κύκλωμα με λειτουργίες:

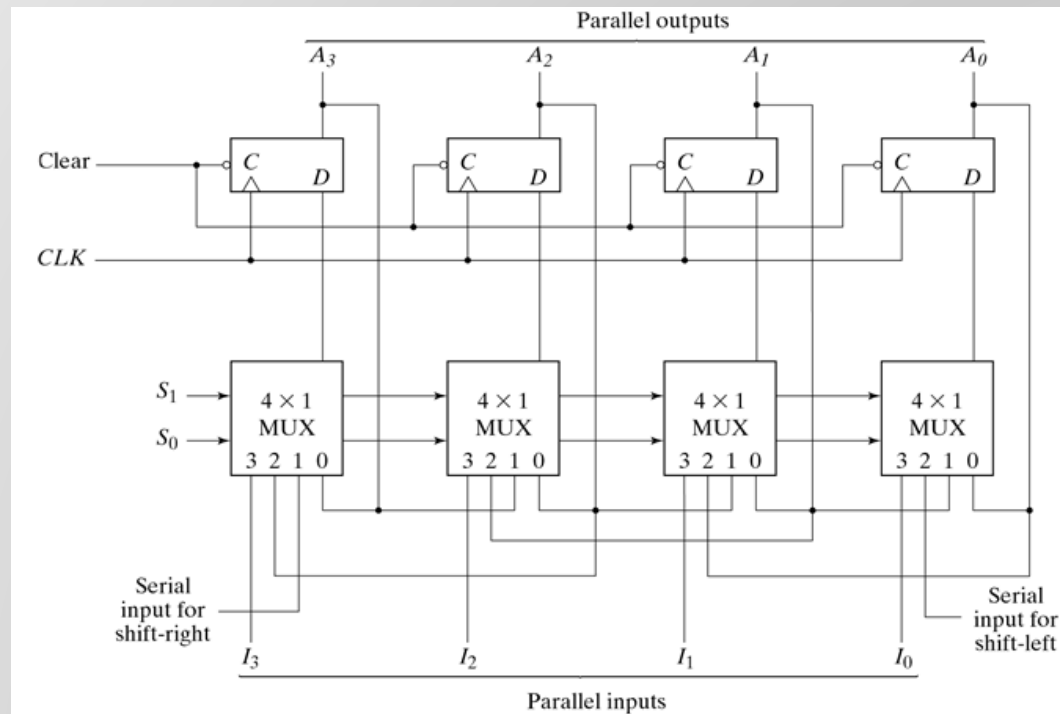
**ολίσθηση δεξιά, ολίσθηση αριστερά,
παράλληλη φόρτωση, κράτηση παρούσας
κατάστασης.**

- Καταχωρητές ολίσθησης μπορούν επίσης να σχεδιαστούν για ολίσθηση περισσότερων του ενός bit \rightarrow ολίσθηση $k (< n)$ bits.



Αμφίδρομος καταχωρητής ολίσθησης με παράλληλη φόρτωση 4bit

- Γενικός καταχωρητής ολίσθησης.



Parallel outputs: Παράλληλες εξόδους

Serial input of shift-right: Σειριακοί είσοδοι για δεξιά ολίσθηση

Serial input of shift-left: Σειριακοί είσοδοι για αριστερή ολίσθηση

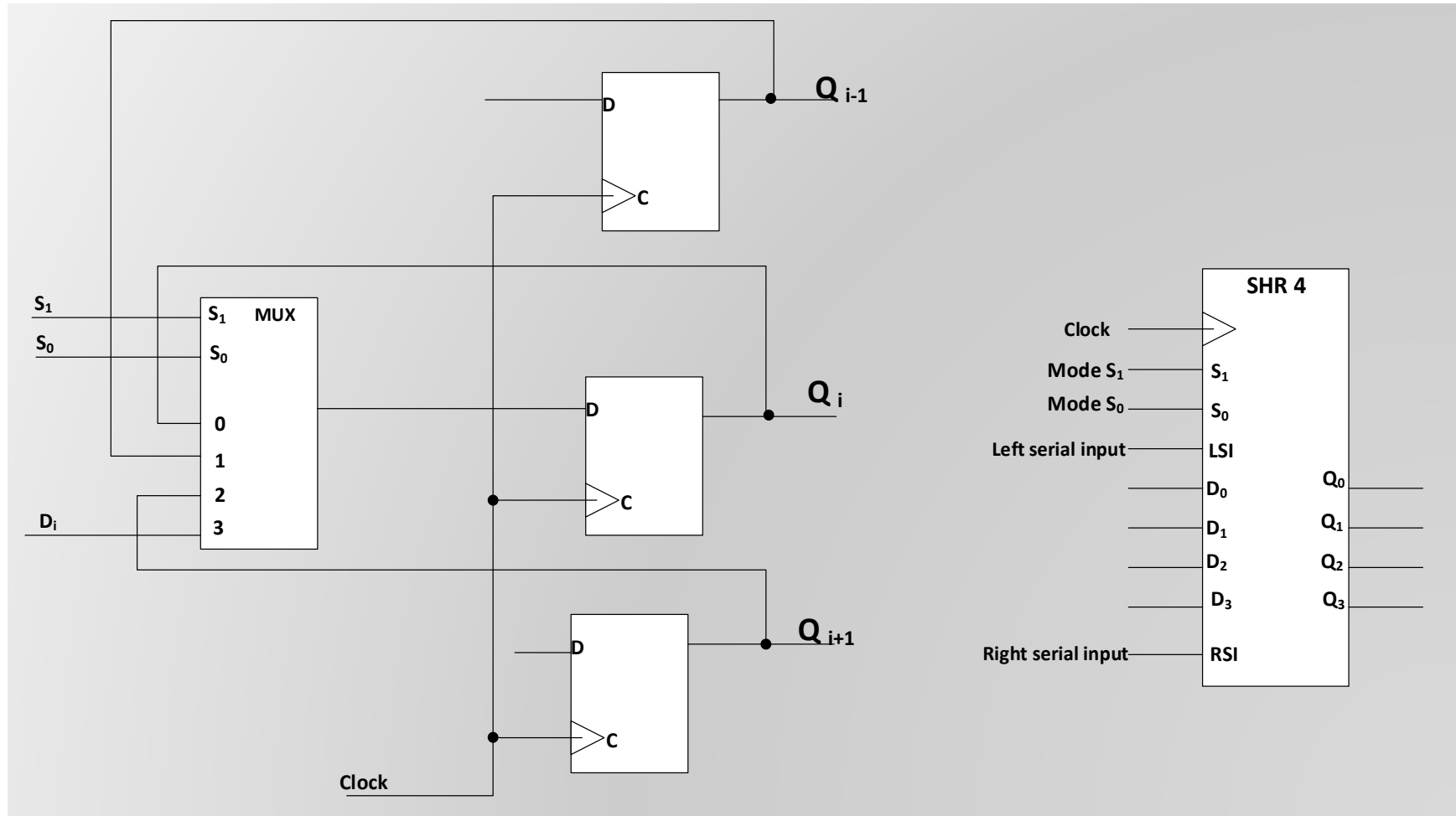


Καταχωρητής Ολίσθησης Διπλής Κατεύθυνσης (1)

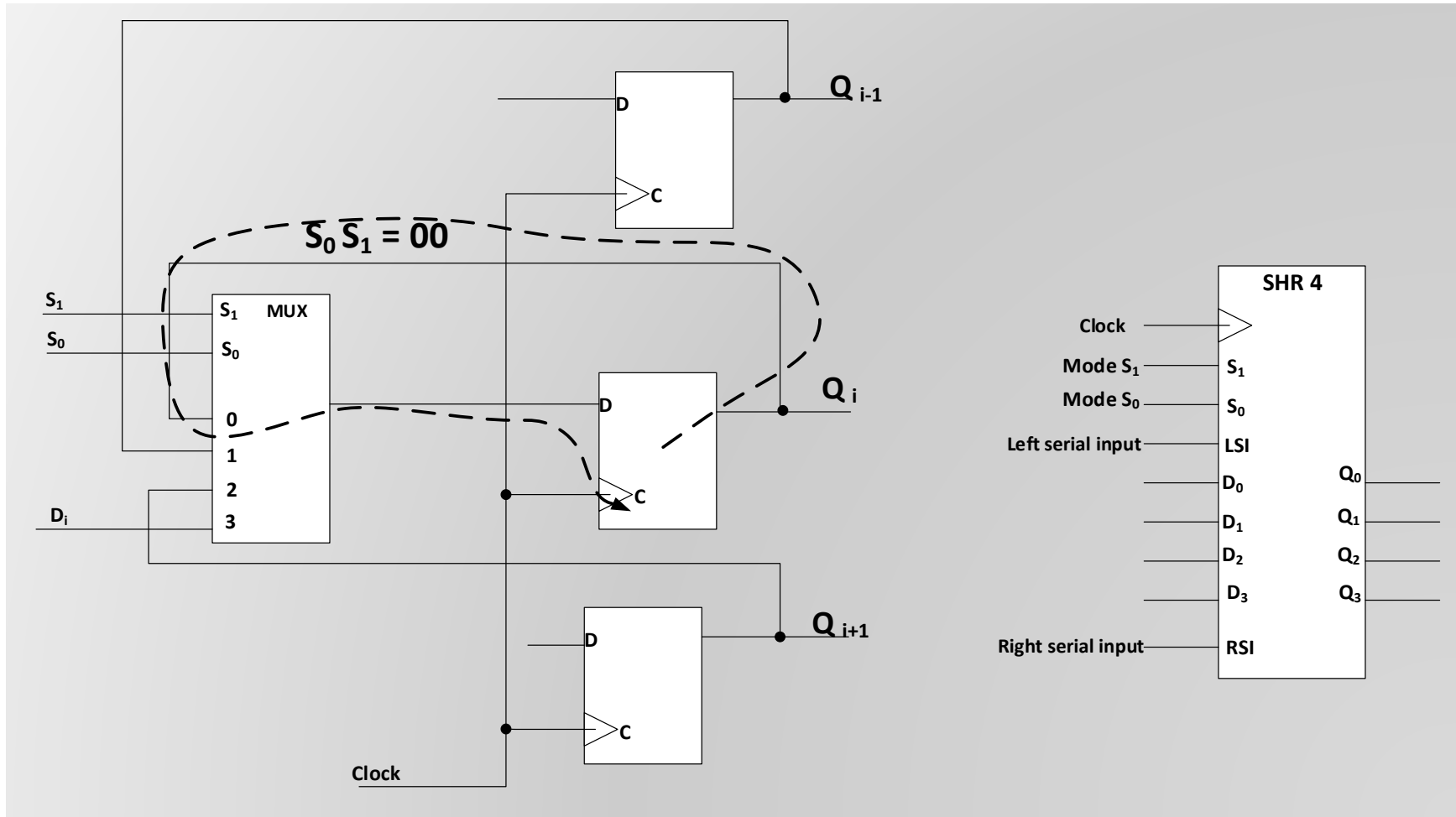
S_1S_0	Λειτουργία
00	Κρατηση παρούσας κατάστασης
01	Ολίσθηση προς τα κάτω
10	Ολίσθηση προς τα πάνω
11	Παράλληλη φόρτωση



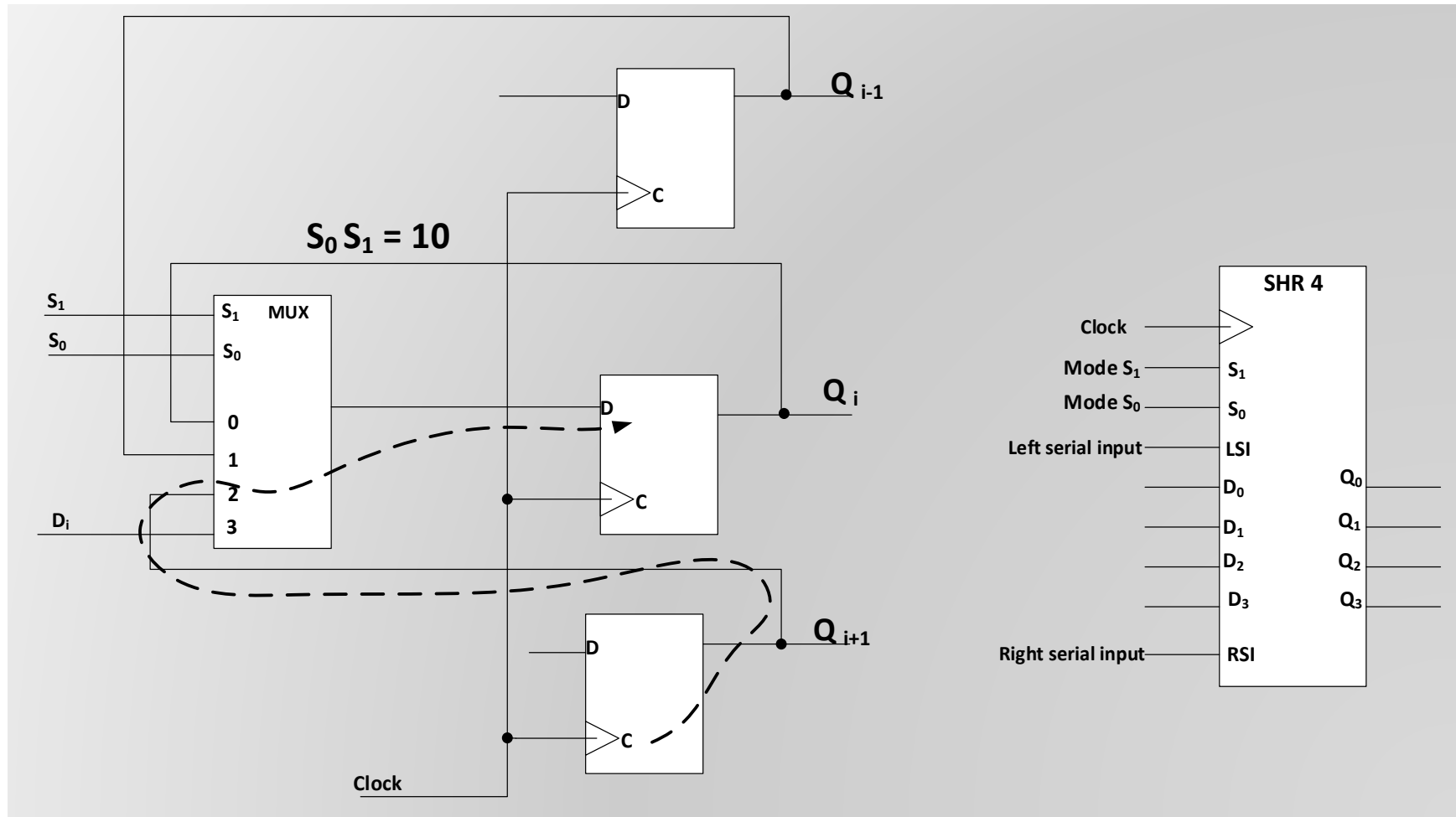
Καταχωρητής Ολίσθησης Διπλής Κατεύθυνσης (2)



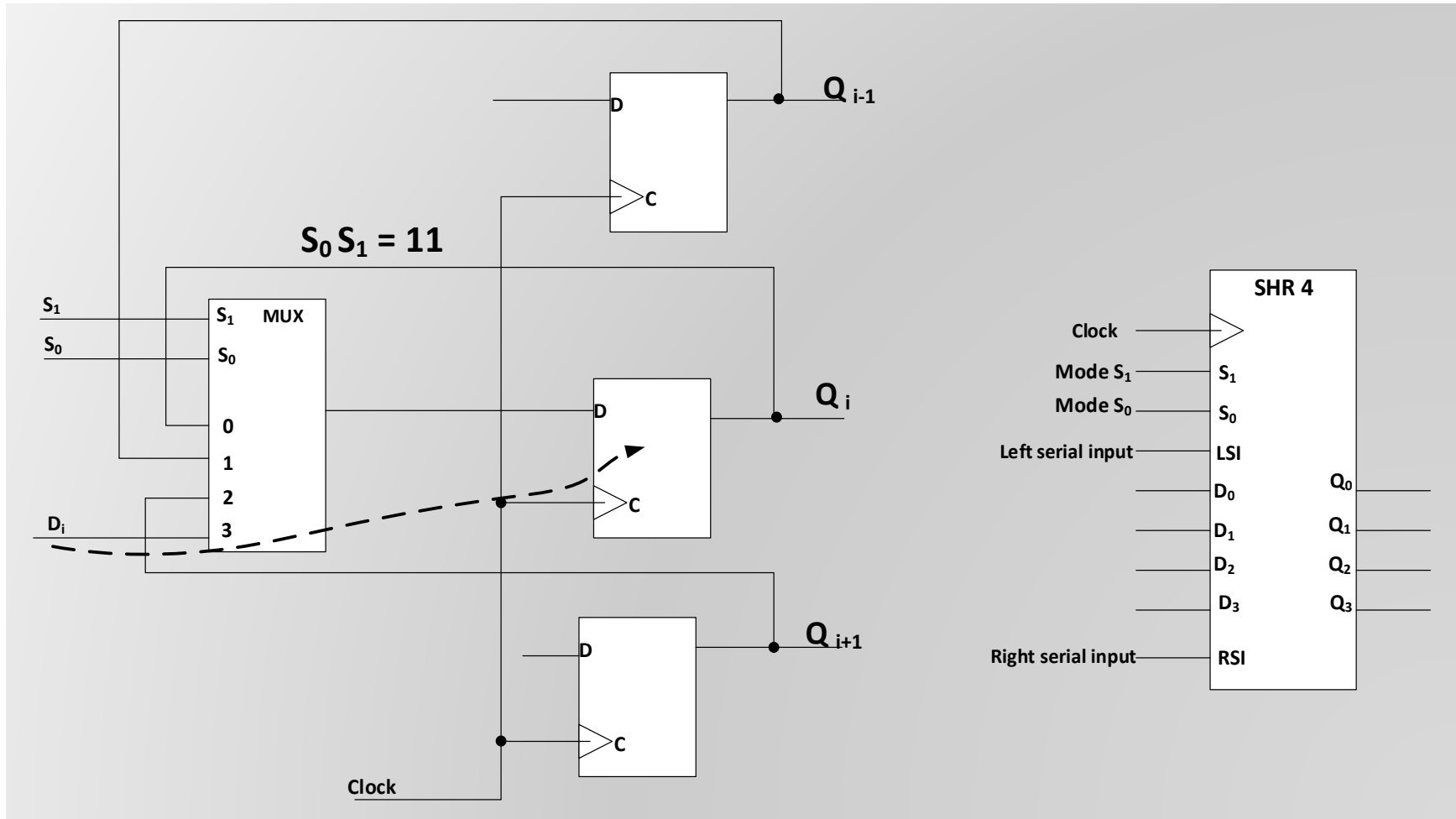
Καταχωρητής Ολίσθησης Διπλής Κατεύθυνσης (3)



Καταχωρητής Ολίσθησης Διπλής Κατεύθυνσης (4)



Καταχωρητής Ολίσθησης Διπλής Κατεύθυνσης (5)



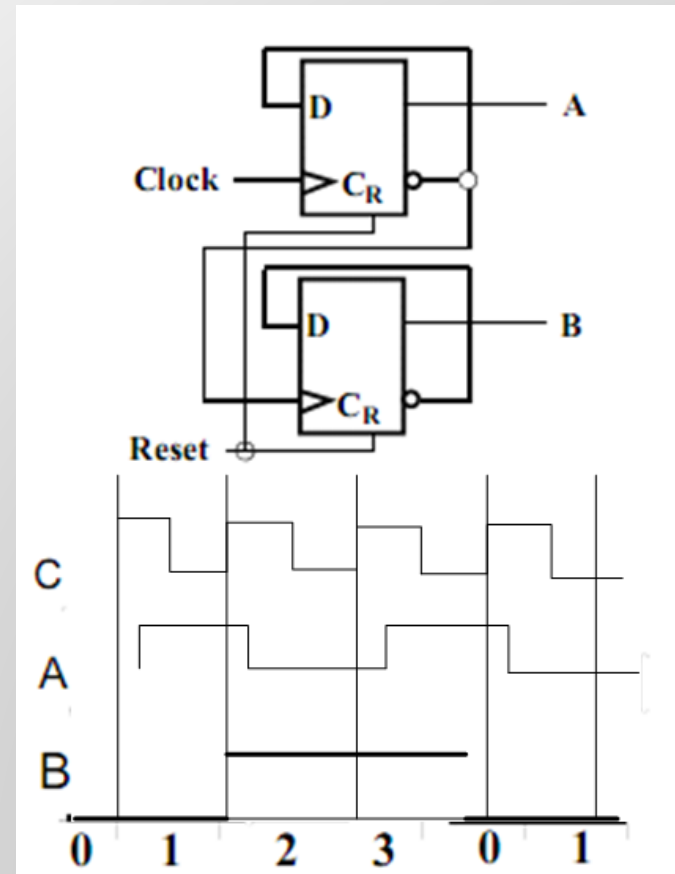
Μετρητές

- Ένας μετρητής είναι ένας καταχωρητής που «μετρά» μια προκαθορισμένοι ακολουθία καταστάσεων, βάση της εφαρμογής παλμών του ρολογιού.
- Οι μετρητές κατηγοριοποιούνται σε:
 - Μετρητές Ριπής:
 - Το ρολόι του συστήματος ενώνεται στην είσοδο ρολογιού του LSB FF.
 - Για τα υπόλοιπα FFs, η έξοδος ενός FF ενώνεται στην είσοδο ρολογιού του επόμενου σημαντικού FF.
 - Δεν υπάρχει κοινό ρολόι.
 - Χαμηλή κατανάλωση ισχύος. Γιατι;
 - Σύγχρονους Μετρητές:
 - Όλα τα FFs έχουν κοινό ρολόι.
 - Χρήση κοινής λογικής για υλοποίηση της επόμενης κατάστασης.



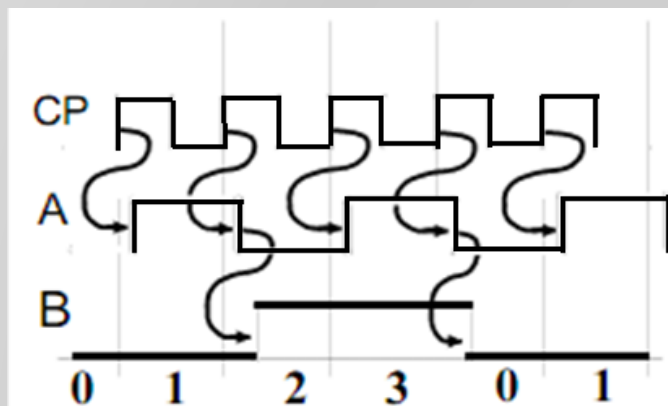
Μετρητής Ριπής (Ripple Counter) (1)

- Πως δουλευει;
 - Στην θετική ακμή στην είσοδο του ρολογιού του A, το A συμπληρώνεται.
 - Η είσοδος του ρολογιού για το B είναι το συμπλήρωμα της εξόδου του A.
 - Όταν το A αλλάξει από 1 σε 0 (αρνητική ακμή), υπάρχει θετική ακμή (0 σε 1) στην είσοφο του ρολογιού του B, προσκαλώντας το B να συμπληρωθεί.
- → Προς-τα-πάνω (upward counting). Γιατι;



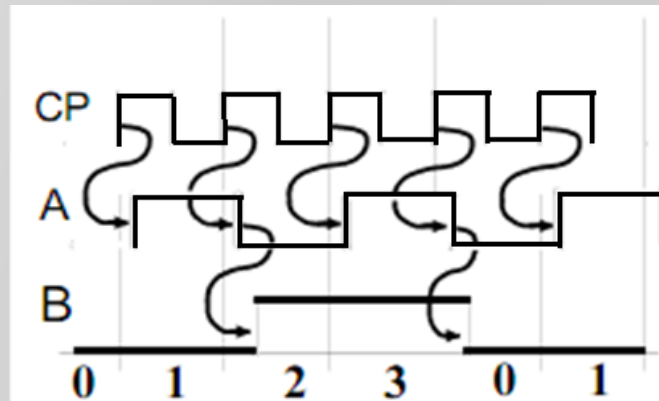
Μετρητής Ριπής (Ripple Counter) (2)

- Τα βέλη δείχνουν την σχέση αιτίας-αποτελέσματος από την προηγούμενη διαφάνεια.
- Η αντίστοιχη ακολουθία καταστάσεων είναι: $(B, A) = (0, 0), (0, 1), (1, 0), (1, 1), (0, 0), (0, 1), \dots$

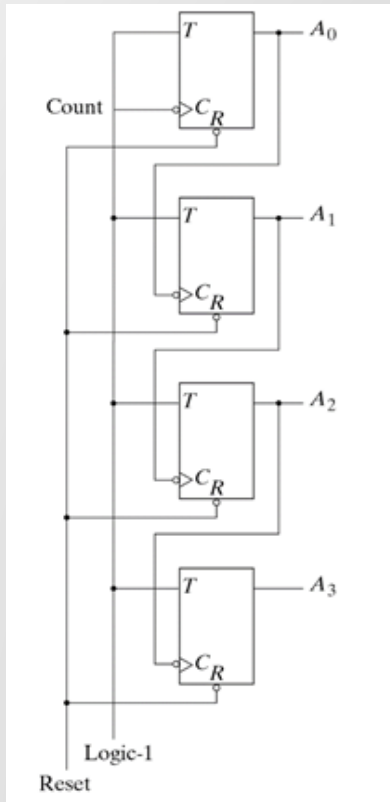


Μετρητής Ριπής (Ripple Counter) (3)

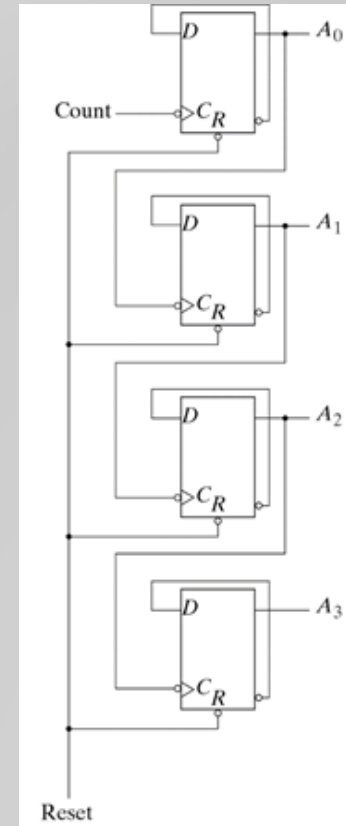
- Κάθε επιπρόσθετο bit, C, D, Συμπεριφέρεται όπως το bit B, αλλάζοντας 50% λιγότερο συχνά από το προηγούμενο bit.
- Για 3 bits: (C, B, A) = (0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1), (0, 0, 0), ...



Δυαδικός Μετρητής Ριπής με T και D flip-flop



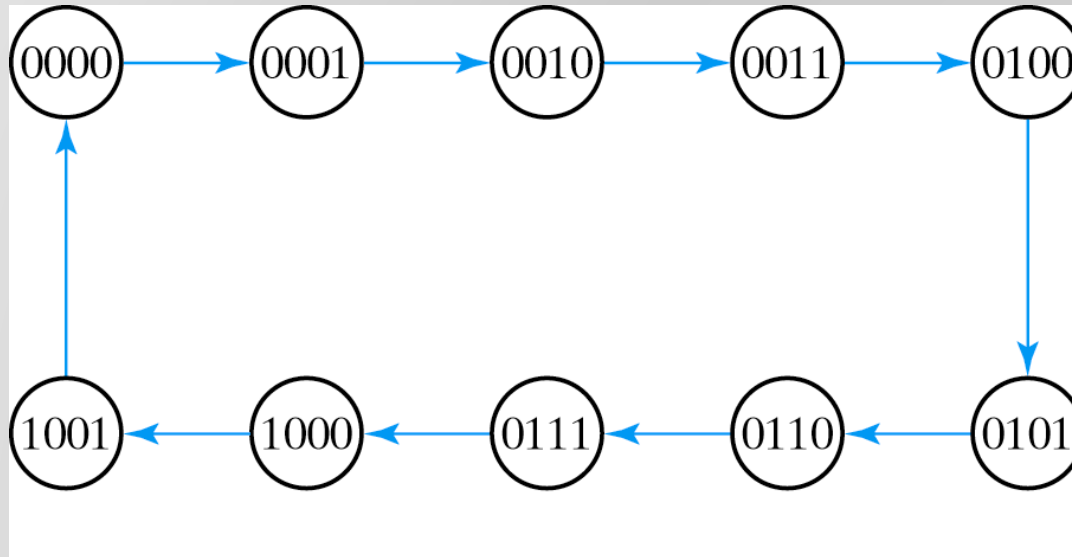
(α) T flip-flops



(β) D flip-flops

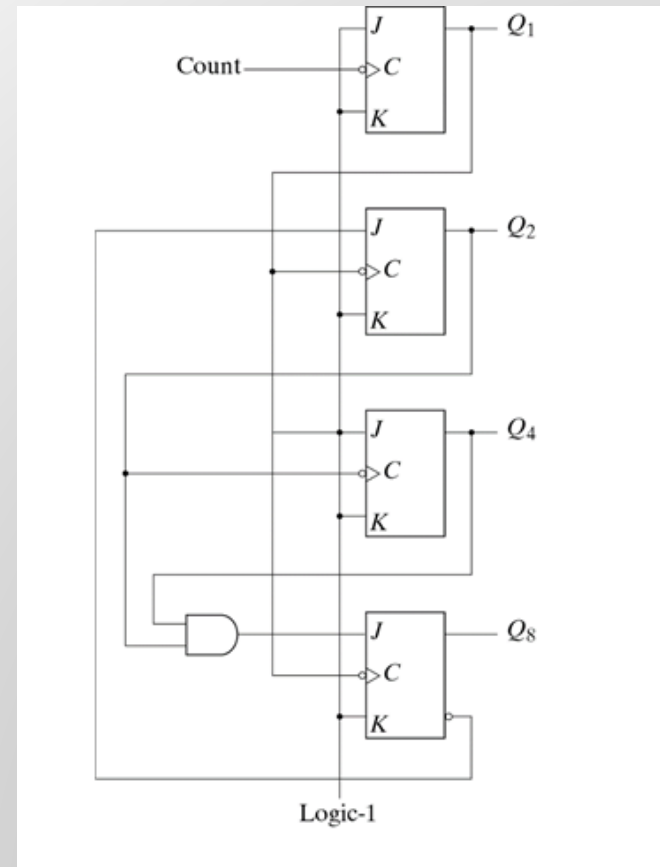
Διάγραμμα καταστάσεων ενός δεκαδικού μετρητή BCD

- Απαιτούνται 4 FF.
- Ίδιο με δυαδικό μετρητή μόνο που μετά το 1001 βρίσκεται το 0000.



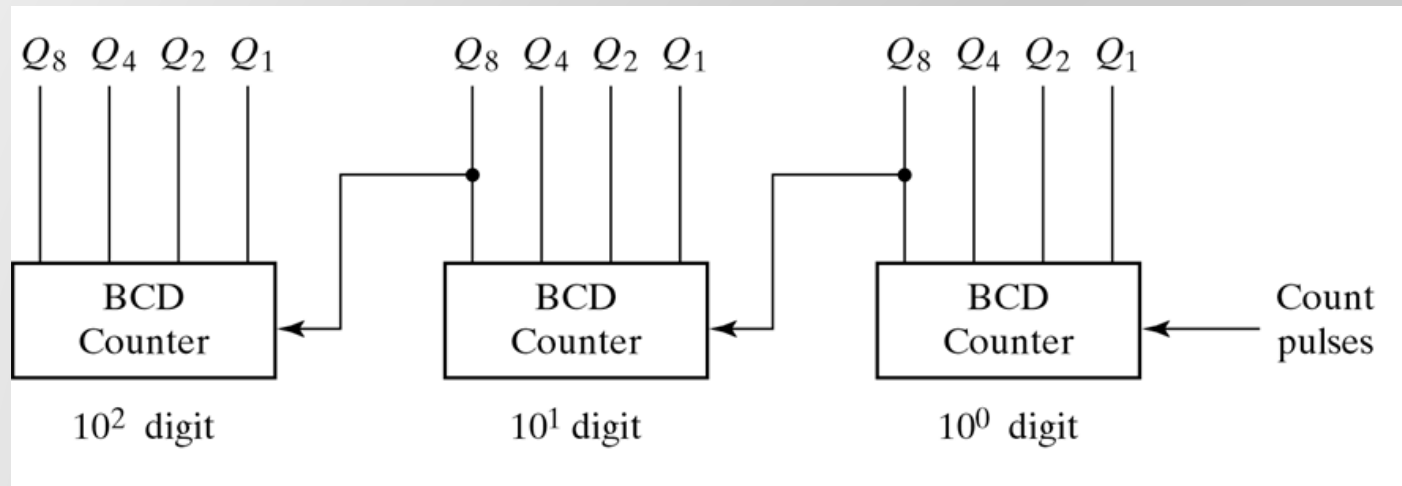
Λογικό διάγραμμα μετρητή ριπής BCD με JK FF

- Ασύγχρονο ακολουθιακό κύκλωμα.
- Θυμηθείτε ότι
 - Αν $J = 1$ $Q = 1$.
 - Αν $K = 1$ $Q = 0$.
 - Αν $J = K = 1$ $Q =$ συμπλήρωση.
 - Αν $J = K = 0$ $Q =$ σταθερό.



BCD Ripple Counter

Σύνδεση μετρητών BCD για πολλαπλά δεκαδικά ψηφία



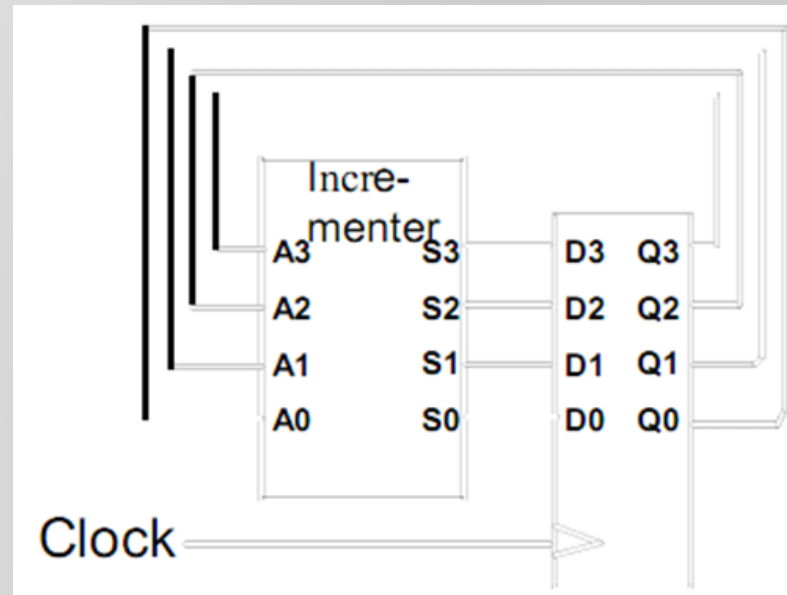
Block Diagram of a Three-Decade Decimal BCD Counter (Σχηματικό διάγραμμα για τρεις δεκαδικούς BCD μετρητές)

Count pulses: Μετρητής παλμών



Σύγχρονοι Δυαδικοί Μετρητές (Synchronous Binary Counters)

- Για εξουδετέρωση του προβλήματος του “ripple effect” χρησιμοποιείται κοινό ρολόι για όλα τα FFs και ένα συνδυαστικό μέρος του κυκλώματος για παραγωγή της επόμενης κατάστασης.
- Για ένα μετρητή προς τα πάνω → χρήση συνδυαστικού Incrementer.

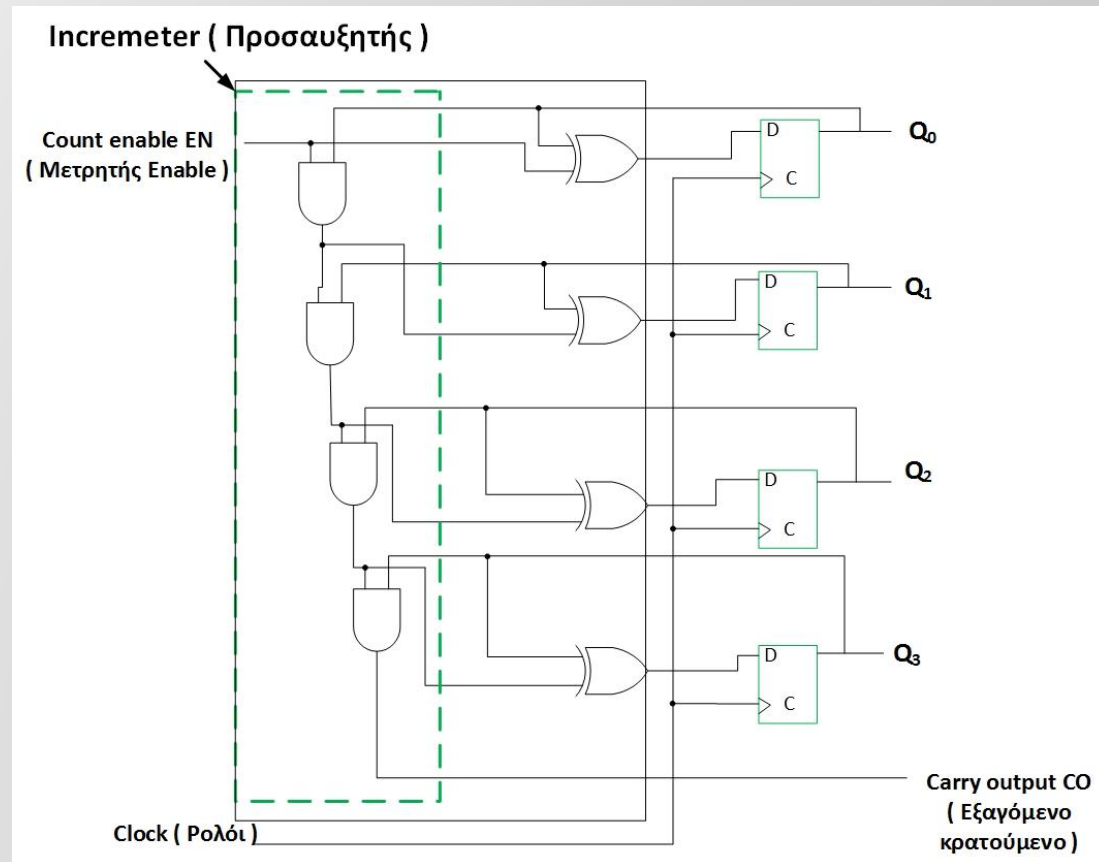


Σειριακή τοποθέτηση Πυλών (Serial Gating) (1)

- Εσωτερική λογική
 - XOR συμπληρώνει ή κρατά το κάθε bit ανάλογα.
 - Αλυσίδα από AND συμπληρώνει ένα bit εάν όλα τα bits από το LSB μέχρι το τρέχον είναι 1.
- Count Enable
 - Θέτει όλες τις εξόδους των AND σε 0 για να «κρατήσει» την παρούσα κατάσταση.
- Carry out
 - Κομμάτι του Incrementer (προσαυξητή).
 - Ενώνεται στο Count Enable του επόμενου μετρητή 4-bit για δημιουργία μεγαλύτερων μετρητών.



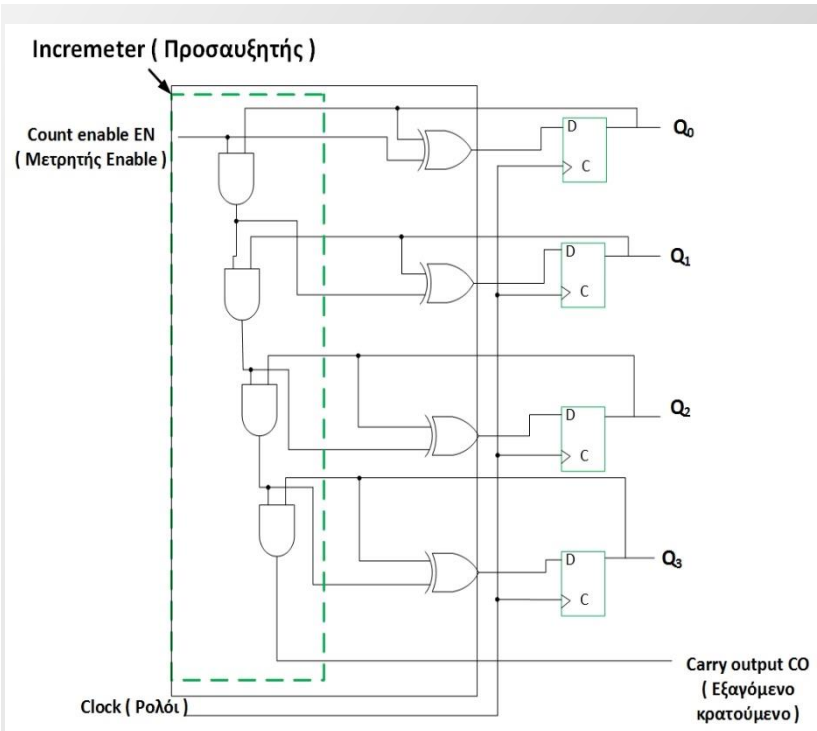
Σειριακή τοποθέτηση Πυλών (Serial Gating) (2)



Logic Diagram-Serial Gating (Λογικό διάγραμμα-Σειριακών πυλών)



Σειριακή τοποθέτηση Πυλών (Serial Gating) (3)



$$Q_0 (t + 1) = Q_0 (t) \oplus EN$$

$$Q_1 (t + 1) = Q_1 (t) \oplus (EN \cdot Q_0 (t))$$

$$Q_2 (t + 1) = Q_2 (t) \oplus (EN \cdot Q_0 (t) \cdot Q_1 (t))$$

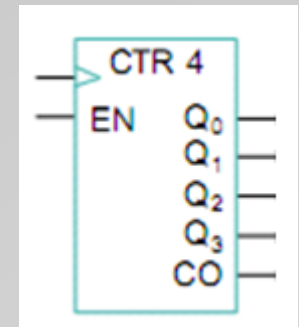
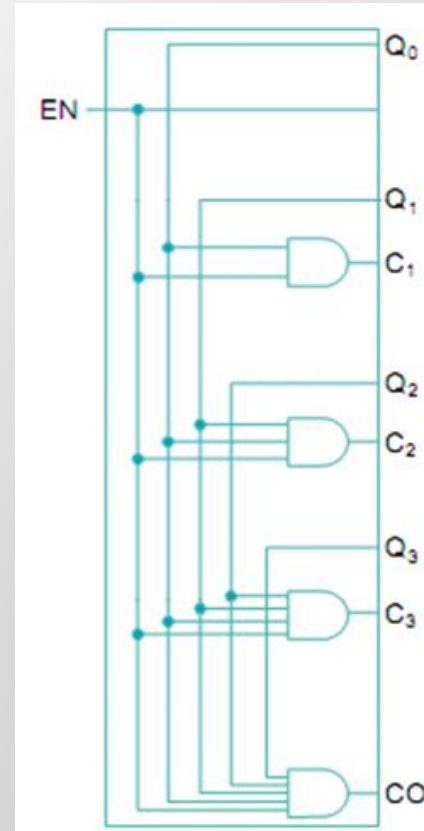
$$Q_3 (t + 1) = Q_3 (t) \oplus (EN \cdot Q_0 (t) \cdot Q_1 (t) \cdot Q_2 (t))$$

Logic Diagram-Serial Gating (Λογικό διάγραμμα-Σειριακών πυλών)



Παράλληλη τοποθέτηση Πύλών (Parallel gating)

- Αλυσίδα του Carry
 - Σειρά από πύλες AND μέσα από την οποία περνά το carry (“ripples”).
 - Δίνει μεγάλες καθυστερήσεις.
 - Ονομάζεται “serial gating”.
- Αντικατάσταση της AND αλυσίδας του carry με πύλες AND παράλληλα
 - Μειώνει τις καθυστερήσεις μονοπατιών.
 - Ονομάζεται “parallel gating”.
 - Παρόμοιο με “carry lookahead”.
 - Το lookahead χρησιμοποιείται στα Cos και Ens για αποτροπή δημιουργίας μεγάλων μονοπατιών σε μεγαλύτερους μετρητές.
 - Πλεονεκτήματα για parallel gating: πιο γρήγορο σε κάποιες περιπτώσεις (1111 → 0000).



Συμβολισμός

Logic Diagram-Parallel Gating (Λογικό διάγραμμα-παράλληλων πυλών)



Σύγχρονοι Δυαδικοί Μετρητές (1)

- Η διαδικασία σχεδιασμού για ένα δυαδικό μετρητή είναι ίδια με αυτή για ένα τυχαίο σύγχρονο ακολουθιακό κύκλωμα.
- Οι είσοδοι του κυκλώματος είναι το ρολόι (CLK) και άλλα απαραίτητα σήματα ελέγχου (EN, Load κτλ.).
- Οι έξοδοι του κυκλώματος είναι οι έξοδοι των FF (παρούσα κατάσταση).
- Συνήθως οι πιο αποτελεσματικές υλοποιήσεις χρησιμοποιούν T-FFs ή JK-FFs.



Μετρητές με JK flip-flop (1)

Σύγχρονοι Δυαδικοί Μετρητές

Σχεδιασμός με J-K FFs για Up Counter $4^{\omega v}$ -bit

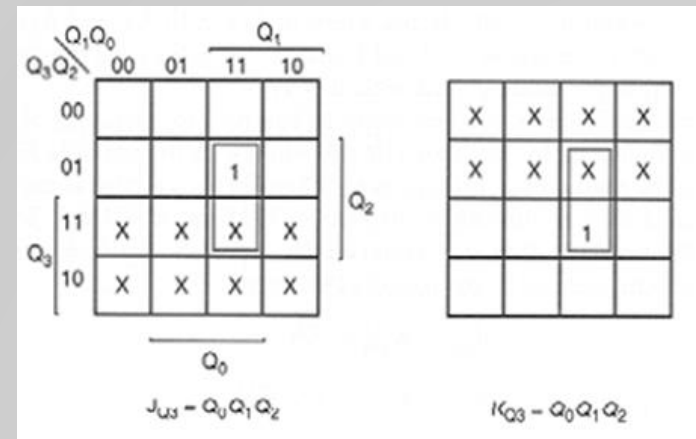
Present state				Next state				Flip-flop inputs							
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	J_{Q3}	K_{Q3}	J_{Q2}	K_{Q2}	J_{Q1}	K_{Q1}	J_{Q0}	K_{Q0}
0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X
0	0	0	1	0	0	1	0	0	X	0	X	1	X	X	1
0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X
0	0	1	1	0	1	0	0	0	X	1	X	X	1	X	1
0	1	0	0	0	1	0	1	0	X	X	0	0	X	1	X
0	1	0	1	0	1	1	0	0	X	X	0	1	X	X	1
0	1	1	0	0	1	1	1	0	X	X	0	X	0	1	X
0	1	1	1	1	0	0	0	1	X	X	1	X	1	X	1
1	0	0	0	1	0	0	1	X	0	0	X	0	X	1	X
1	0	0	1	1	0	1	0	X	0	0	X	1	X	X	1
1	0	1	0	1	0	1	1	X	0	0	X	X	0	1	X
1	0	1	1	1	1	0	0	X	0	1	X	X	1	X	1
1	1	0	0	1	1	0	1	X	0	X	0	0	X	1	X
1	1	0	1	1	1	1	0	X	0	X	0	1	X	X	1
1	1	1	0	1	1	1	1	X	0	X	0	X	0	1	X
1	1	1	1	0	0	0	0	X	1	X	1	X	1	X	1

(a) JK Flip-Flop			
$Q(t)$	$Q(t+1)$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0



Μετρητές με JK flip-flop (2)

Present state				Next state				J_{03}	K_{03}
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0		
0	0	0	0	0	0	0	1	0	X
0	0	0	1	0	0	1	0	0	X
0	0	1	0	0	0	1	1	0	X
0	0	1	1	0	1	0	0	0	X
0	1	0	0	0	1	0	1	0	X
0	1	0	1	0	1	1	0	0	X
0	1	1	0	0	1	1	1	0	X
0	1	1	1	1	0	0	0	1	X
1	0	0	0	1	0	0	1	X	0
1	0	0	1	1	0	1	0	X	0
1	0	1	0	1	0	1	1	X	0
1	0	1	1	1	1	0	0	X	0
1	1	0	0	1	1	0	1	X	0
1	1	0	1	1	1	1	0	X	0
1	1	1	0	1	1	1	1	X	0
1	1	1	1	0	0	0	0	X	1



Μετρητές με JK flip-flop (3)

Present state				Next state				Flip-flop	
Q ₃	Q ₂	Q ₁	Q ₀	Q ₃	Q ₂	Q ₁	Q ₀	J _{Q2}	K _{Q2}
0	0	0	0	0	0	0	1	0	X
0	0	0	1	0	0	1	0	0	X
0	0	1	0	0	0	1	1	0	X
0	0	1	1	0	1	0	0	1	X
0	1	0	0	0	1	0	1	X	0
0	1	0	1	0	1	1	0	X	0
0	1	1	0	0	1	1	1	X	0
0	1	1	1	1	0	0	0	X	1
1	0	0	0	1	0	0	1	0	X
1	0	0	1	1	0	1	0	0	X
1	0	1	0	1	0	1	1	0	X
1	0	1	1	1	1	0	0	1	X
1	1	0	0	1	1	0	1	X	0
1	1	0	1	1	1	1	0	X	0
1	1	1	0	1	1	1	1	X	0
1	1	1	1	0	0	0	0	X	1

		1	
X	X	X	X
X	X	X	X
		1	

$$J_{Q2} = Q_0 Q_1$$

X	X	X	X
		1	
		1	
X	X	X	X

$$K_{Q2} = Q_0 Q_1$$



Μετρητές με JK flip-flop (4)

Present state				Next state				p inputs	
Q ₃	Q ₂	Q ₁	Q ₀	Q ₃	Q ₂	Q ₁	Q ₀	J _{Q1}	K _{Q1}
0	0	0	0	0	0	0	1	0	X
0	0	0	1	0	0	1	0	1	X
0	0	1	0	0	0	1	1	X	0
0	0	1	1	0	1	0	0	X	1
0	1	0	0	0	1	0	1	0	X
0	1	0	1	0	1	1	0	1	X
0	1	1	0	0	1	1	1	X	0
0	1	1	1	1	0	0	0	X	1
1	0	0	0	1	0	0	1	0	X
1	0	0	1	1	0	1	0	1	X
1	0	1	0	1	0	1	1	X	0
1	0	1	1	1	1	0	0	X	1
1	1	0	0	1	1	0	1	0	X
1	1	0	1	1	1	1	0	1	X
1	1	1	0	1	1	1	1	X	0
1	1	1	1	0	0	0	0	X	1

		1	X	X
		1	X	X
		1	X	X
		1	X	X

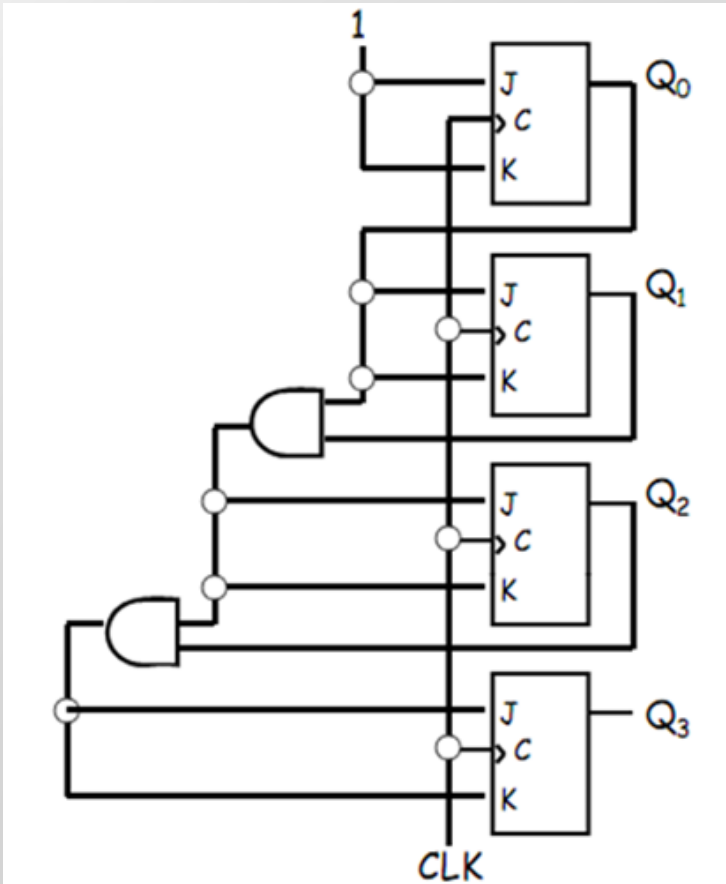
$$J_{Q1} = Q_0$$

X	X	1	
X	X	1	
X	X	1	
X	X	1	

$$K_{Q1} = Q_0$$



Μετρητές με JK flip-flop (5)



$$J_{Q0} = 1$$

$$K_{Q0} = 1$$

$$J_{Q1} = Q_0$$

$$K_{Q1} = Q_0$$

$$J_{Q2} = Q_0 Q_1$$

$$K_{Q2} = Q_0 Q_1$$

$$J_{Q3} = Q_0 Q_1 Q_2$$

$$K_{Q3} = Q_0 Q_1 Q_2$$



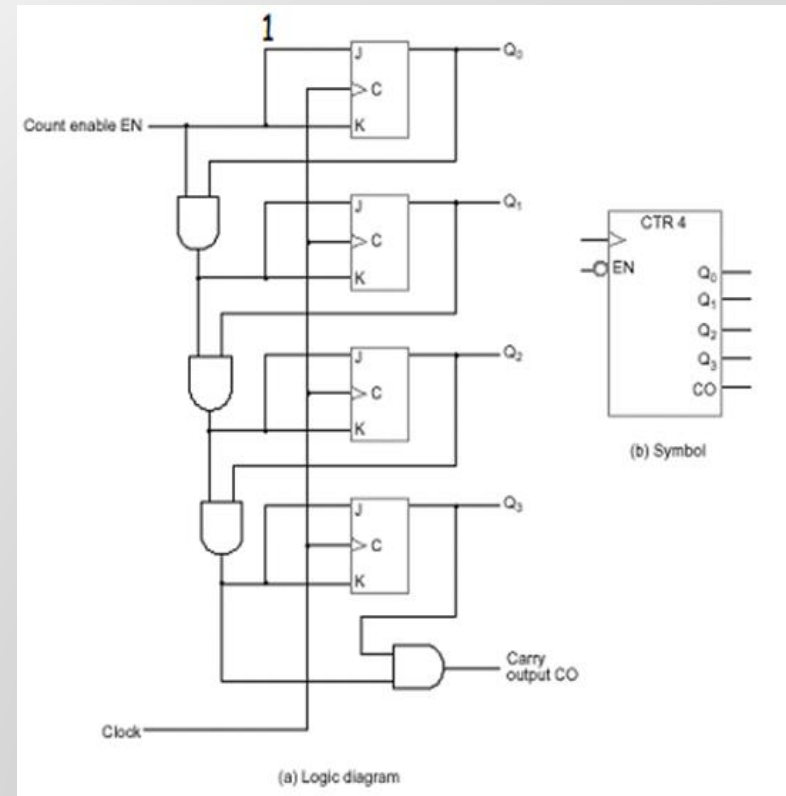
Σχεδιασμός μετρητή με JK Flip-Flop με EN, CO (1)

- EN = σήμα ενεργοποίησης όταν είναι 0 ο μετρητής διατηρεί την παρούσα κατάσταση, όταν είναι 1 μετρά.
- CO = σήμα εξόδου carry (κρατούμενου), χρησιμοποιείται για την κατασκευή μεγαλύτερων μετρητών.



Σχεδιασμός μετρητή με JK Flip-Flop με EN, CO (2)

- $J_{Q_0} = 1 \times EN$
- $K_{Q_0} = 1 \times EN$
- $J_{Q_1} = Q_0 \times EN$
- $K_{Q_1} = Q_0 \times EN$
- $J_{Q_2} = Q_0 Q_1 \times EN$
- $K_{Q_2} = Q_0 Q_1 \times EN$
- $J_{Q_3} = Q_0 Q_1 Q_2 \times EN$
- $K_{Q_3} = Q_0 Q_1 Q_2 \times EN$
- $CO = Q_0 Q_1 Q_2 Q_3 \times EN$

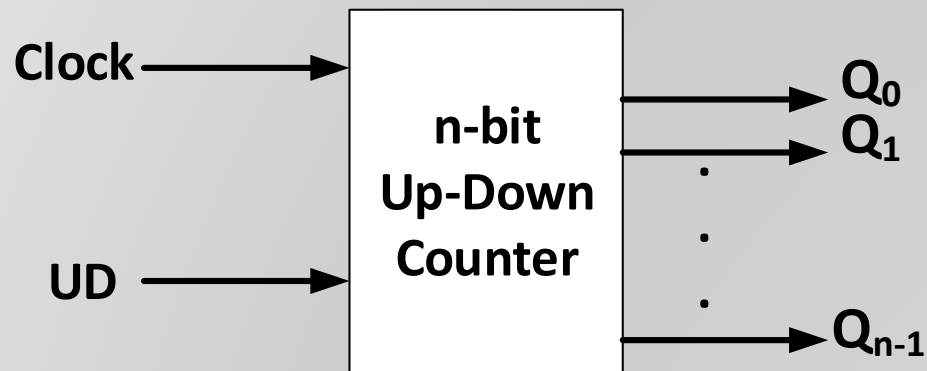


Αν $J=K=0$ τότε συμπλήρωμα



Δυαδικός Μετρητής Πάνω-Κάτω (1)

- $UD = 0$: μετρά προς τα πάνω.
- $UD = 1$: μετρά προς τα κάτω.



Up-Down Counter: Μετρητής Πάνω-Κάτω.



Δυαδικός Μετρητής Πάνω-Κάτω (2)

UD	Q2	Q1	Q0	Q2.D	Q1.D	Q0.D
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	1
0	0	1	1	1	0	0
0	1	0	0	1	0	1
0	1	0	1	1	1	0
0	1	1	0	1	1	1
0	1	1	1	0	0	0



Δυαδικός Μετρητής Πάνω-Κάτω (3)

UD	Q2	Q1	Q0	Q2.D	Q1.D	Q1.D
1	0	0	0	1	1	1
1	0	0	1	0	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

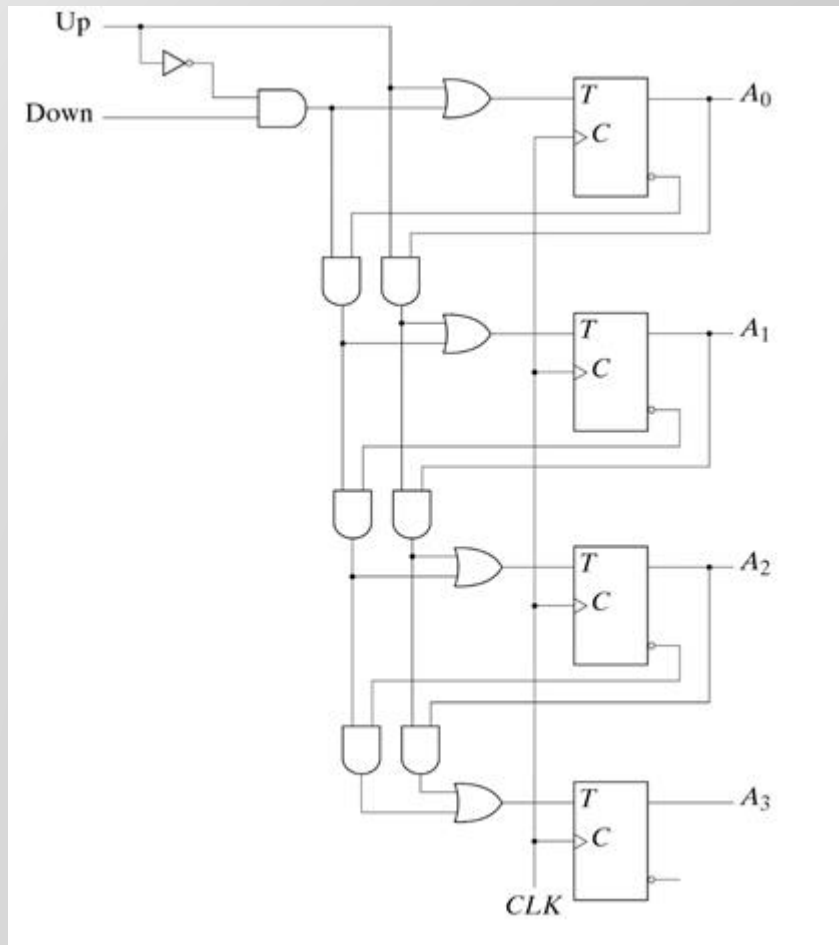


Δυαδικός Μετρητής Πάνω-Κάτω (4)

- Συμπληρώστε τον K-χάρτη για Q2.D, Q1.D και Q0.D, απλοποιείστε, και βρείτε το λογικό διάγραμμα με (a) D-FFs και (b) T-FFs.



Διαδικός Μετρητής Πάνω-Κάτω (5)



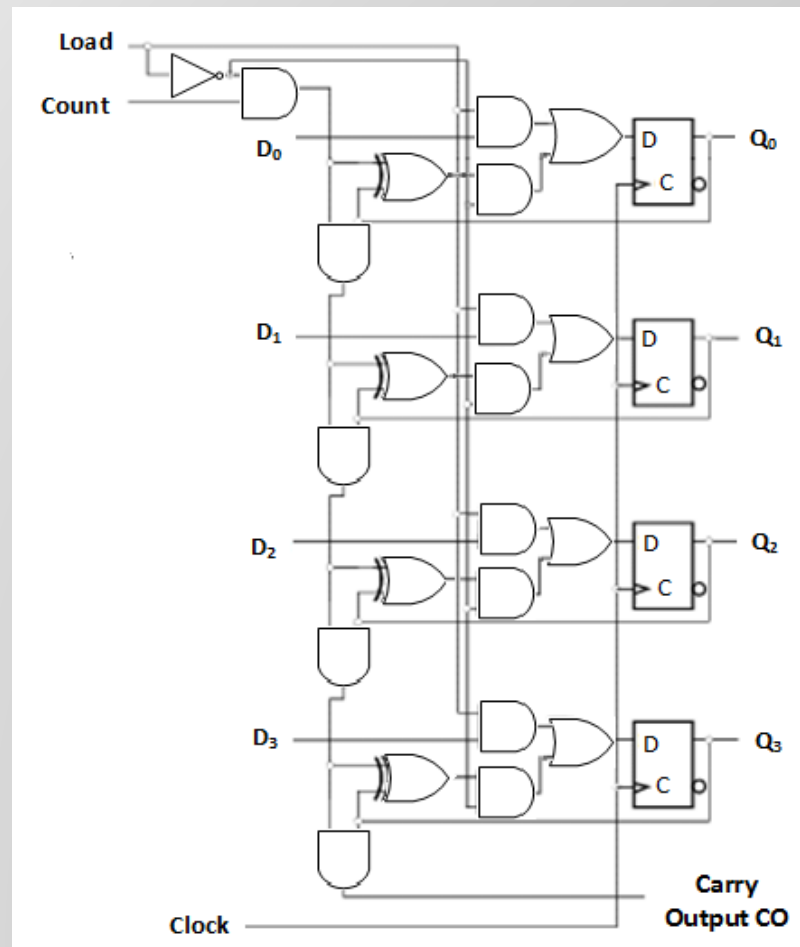
Δυαδικός μετρητής Παράλληλη Φορτωση (1)

- Πρόσθεση μονοπατιού για δεδομένα εισόδων:
 - Ενεργοποιείται για Load = 1.
- Πρόσθεση Λογικής για:
 - Πάυση μέτρησης για Load = 1.
 - Πάυση κράτησης παρούσας κατάστασης για Load = 1.
 - Ενεργοποίηση μέτρησης για Load = 0 και Count = 1.
- Ο πίνακας λειτουργίας:

Load	Count	Λειτουργία
0	0	Κρατηση παρούσας κατάστασης
0	1	Μέτρηση προς τα πάνω
1	X	Παράλληλη Φορτωση

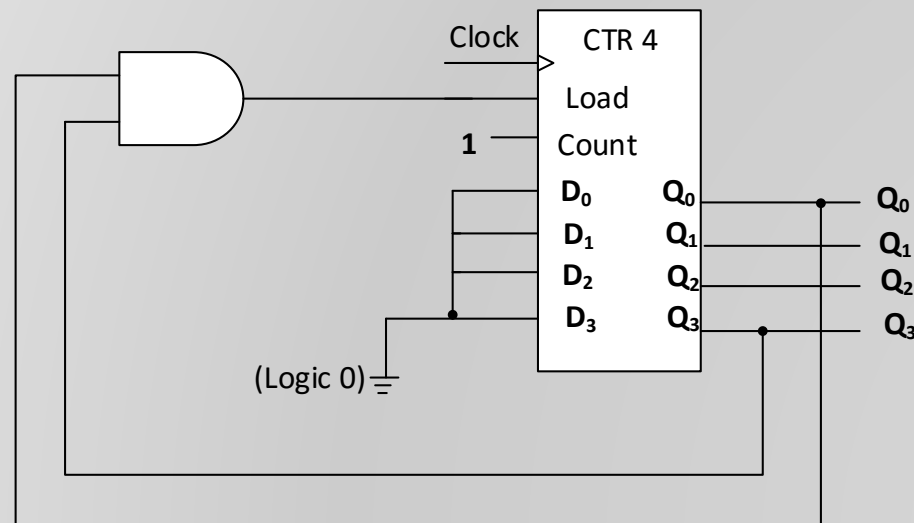


Δυαδικός μετρητής Παράλληλη Φορτωση (2)



Μετρητής BCD (1)

- Ένας δυαδικός μετρητής με παράλληλη φόρτωση μπορεί να μετατραπεί σε ένα σύγχρονο μετρητή BCD με μία μόνο επιπρόσθετη πύλη AND.
- Εξηγήστε πως λειτουργεί αυτό το κύκλωμα...



Μετρητής BCD (2)

- Ο μετρητής ξεκινά με έξοδο = 0000.
- Εάν η έξοδος της AND είναι 0, σε κάθε θετικό παλμό του ρολογιού η τιμή αυξάνεται κατά 1.
- Όταν η έξοδος γίνει 1001, Q_0 και Q_3 γίνονται 1, κάνονται την έξοδο της AND ίση με 1. Αυτή η κατάσταση ενεργοποιεί το Load, έτσι στον επόμενο παλμό μετρητής δεν μετρά, αλλά φορτώνει από τις τέσσερις εισόδους δεδομένων.
- Η τιμή που φορτώνεται είναι 0000.



Μετρητής BCD (3)

- Χρήση ακολουθιακού μοντέλου για το σχεδιασμό σύγχρονου μετρητή BCD με D flip-flops.
- Πίνακας καταστάσεων (επόμενη διαφάνεια).
- Συνδυασμοί εισόδων 1010 μέχρι 1111 είναι συνδυασμοί αδιαφορίας (don't cares).



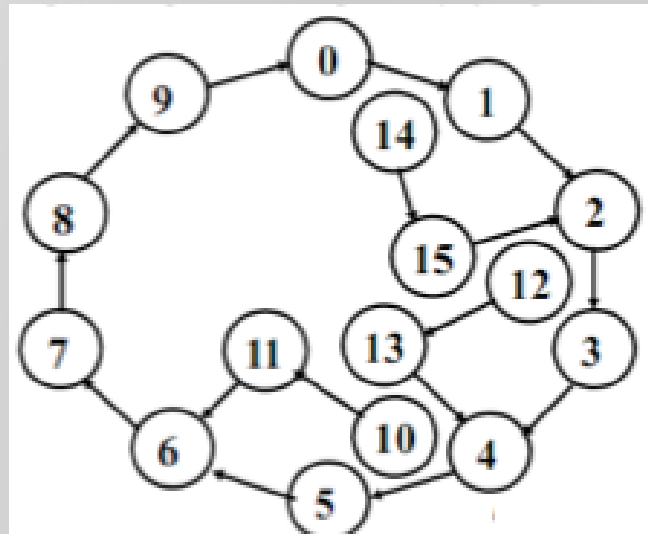
Μετρητής BCD (4)

Παρούσα Κατάσταση				Επόμενη Κατάσταση			
Q8	Q4	Q2	Q1	Q8	Q4	Q2	Q1
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0



Μετρητής BCD (5)

- Βρείτε τις τιμές των 6 επόμενων καταστάσεων για τους συνδυασμούς αδιαφορίας από τις εξισώσεις της προηγούμενης διαφάνειας.
- Βρείτε το ολοκληρωμένο διάγραμμα του κυκλώματος για τις συνθήκες εργασίας.



Μετρητής BCD (6)

Παρούσα Κατάσταση				Επόμενη Κατάσταση			
Q8	Q4	Q2	Q1	Q8	Q4	Q2	Q1
1	0	1	0	1	0	1	1
1	0	1	1	0	1	1	0
1	1	0	0	1	1	0	1
1	1	0	1	0	1	0	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	1	0



Μετρητής BCD (7)

- Για τον προηγούμενο σχεδιασμό, σε περίπτωση που το κύκλωμα έρθει σε μια άκυρη κατάσταση (invalid state) θα επανέλθει σε μια έγκυρη κατάσταση μέσα σε 2 περιόδους του ρολογιού (clock cycles).
- Είναι αυτό ικανοποιητικό; Αν όχι:
 - Χρειάζεται κάποιο σήμα που να υποδεικνύει ότι το κύκλωμα εισήλθε σε άκυρη κατάσταση; Ποια η εξίσωση τέτοιου σήματος;
 - Χρειάζεται τροποποίηση του κυκλώματος για να μπορεί να επιστρέψει σε έγκυρη κατάσταση μέσα σε 1 περίοδο του ρολογιού;
 - Χρειάζεται τροποποίηση του κυκλώματος για να μπορεί να επιστρέψει σε συγκεκριμένη έγκυρη κατάσταση (όπως 0);
- Τυχόν ενέργειες εξαρτώνται από:
 - Την εφαρμογή που θα χρησιμοποιήσει το κύκλωμα.
 - Πολιτική της ομάδας σχεδιασμού.



Μετρητές Τυχαίων Ακολουθιών

- Δεδομένης μιας τυχαίας ακολουθίας, σχεδιάστε ένα μετρητή που να αναπαράγει την ακολουθία.
- Διαδικασία:
 - Παραγωγή πίνακα / διάγραμμα καταστάσεων βάση δεδομένης ακολουθίας.
 - Ελαχιστοποίηση (με K-χάρτες, κτλ.).
 - Σχεδιασμός λογικού διαγράμματος.
- Παράδειγμα: Σχεδιάστε το λογικό διάγραμμα για τον μετρητή με την εξής ακολουθία: $0 \rightarrow 7 \rightarrow 6 \rightarrow 1 \rightarrow 0$ ($000 \rightarrow 111 \rightarrow 110 \rightarrow 001 \rightarrow 000$). Χρησιμοποιήστε D FFs.



Μετρώντας modulo n (1)

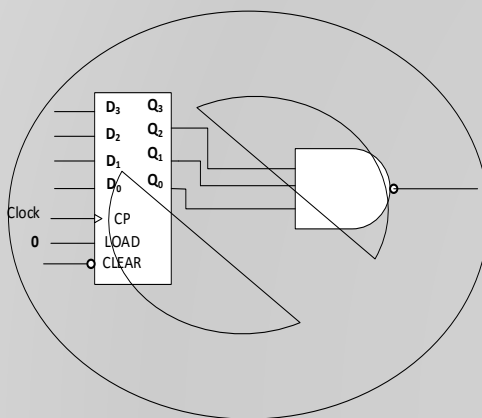
- Μετρητής Διαίρεσης-δια- n (Modulo n)
 - Μετρά το υπόλοιπο της διαίρεσης δια n , όπου το n μπορεί να μην είναι δύναμη του 2 ή
 - μετρά τυχαία ακολουθία από n καταστάσεις, ειδικά σχεδιασμένος βάσης κάθε κατάστασης.
 - Περιλαμβάνει modulo 10 που είναι ο μετρητής BCD.



Μετρώντας modulo n με ασύγχρονη επαναφορά

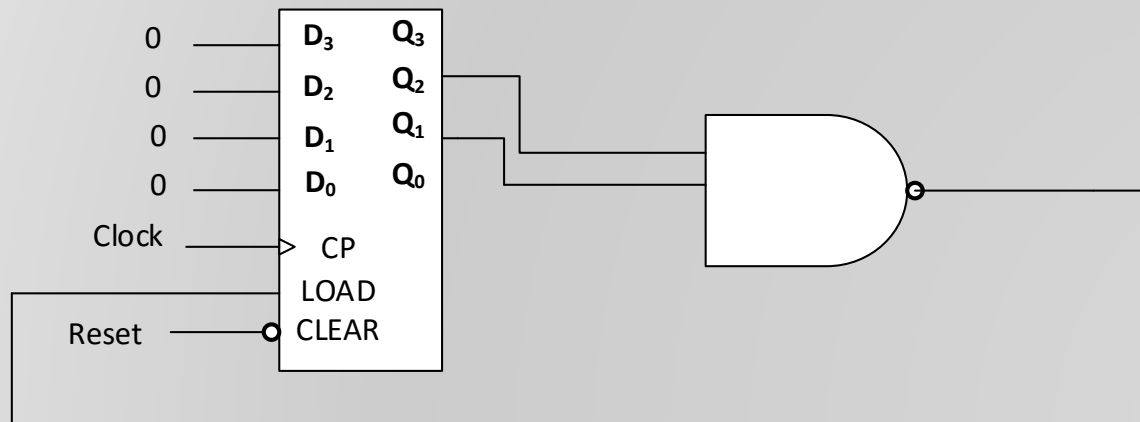
Χρήση ενός σύγχρονου δυαδικού μετρητή 4^{ων}-bit με σύγχρονη φόρτωση (load) και ασύγχρονο clear για την κατασκευή μετρητή Modulo 7.

- Όταν ανιχνεύεται το 7, το κύκλωμα επανέρχεται στο 0 σύγχρονα αφού φορτώνει το 0 από τις εισόδους. Άρα δίνει: $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$ (για πολύ λίγο) $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$ (για πολύ λίγο) κτλ.
- Αυτό ΔΕΝ ΠΡΕΠΕΙ ΝΑ ΓΙΝΕΤΑΙ! Αναφέρεται ως μετρητής «αυτοκτονίας»! (το 7 «σκοτώνεται», αλλά μαζί του και η δουλειά του σχεδιαστή).



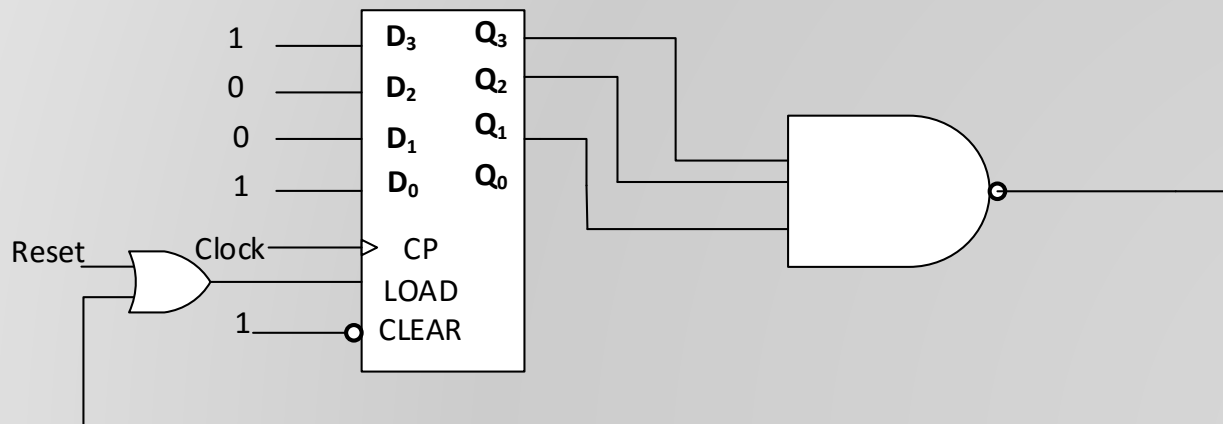
Μετρώντας modulo n με παράλληλη φόρτωση 0

- Χρήση ενός σύγχρονου δυαδικού μετρητή 4^{ων}-bit με σύγχρονη φόρτωση (load) και ασύγχρονο clear για την κατασκευή μετρητή Modulo 7.
- Όταν ανιχνεύεται το 6, το κύκλωμα επανέρχεται στο 0 σύγχρονα αφού φορτώνει το 0 από τις εισόδους. Άρα δίνει: 0 → 1 → 2 → 3 → 4 → 5 → 6 → 0 → 1 → 2 → 3 → 4 → 5 → 6 ...
- Για καταστάσεις μεγαλύτερες (σε δυαδική τιμή) του 0110, χρησιμοποιούνται συνθήκες αδιαφορίας, και έτσι Load = Q2 Q1 (Q0 = X, Q3 = X).



Άλλο παράδειγμα μετρητής 9-14 με σύγχρονη φόρτωση του 9

- Χρήση ενός σύγχρονου δυαδικού μετρητή 4^{ων}-bit με σύγχρονη φόρτωση (load) και ασύγχρονο clear για την κατασκευή μετρητή Modulo 6.
- Το Load χρησιμοποιείται για να θέσει την κατάσταση στο δυαδικό 9 είτε στο Reset είτε όταν ανιχνευτεί 14.
- Αυτό δίνει
- 9 → 10 → 11 → 12 → 13 → 14 9 → 10 → 11 → 12 → 13 → 14 → 9 ...



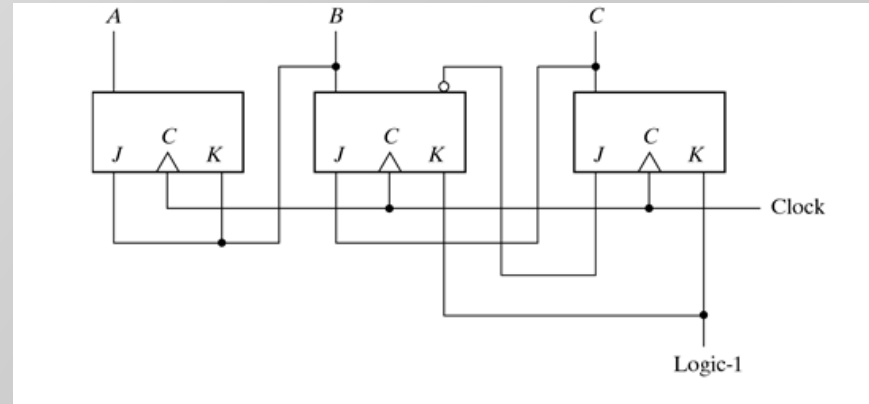
Μετρητές με αχρησιμοποίητες καταστάσεις (1)

- η flip-flop έχουν 2^n δυαδικές καταστάσεις.
- Αν δε χρησιμοποιούνται όλες τότε υπάρχουν αχρησιμοποίητες καταστάσεις.
- Πρέπει να υπάρχει τρόπος ώστε σε περίπτωση που βρεθεί το κύκλωμα σε αχρησιμοποίητη κατάσταση να μπορεί να επανέλθει.
- Μπορούμε να προσδιορίσουμε την επόμενη κατάσταση από την ανάλυση του κυκλώματος.

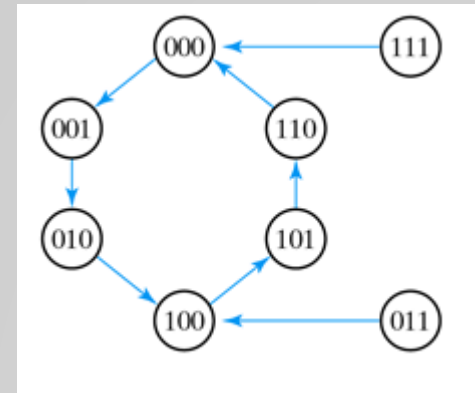


Μετρητές με αχρησιμοποίητες καταστάσεις (2)

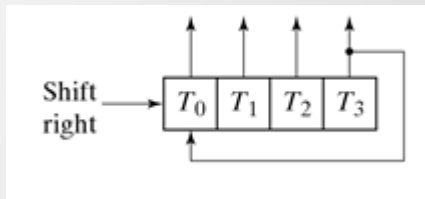
(α) Λογικό διάγραμμα



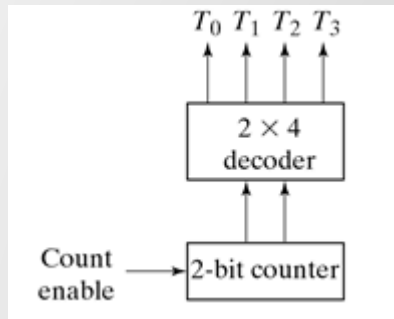
(β) Διάγραμμα καταστάσεων



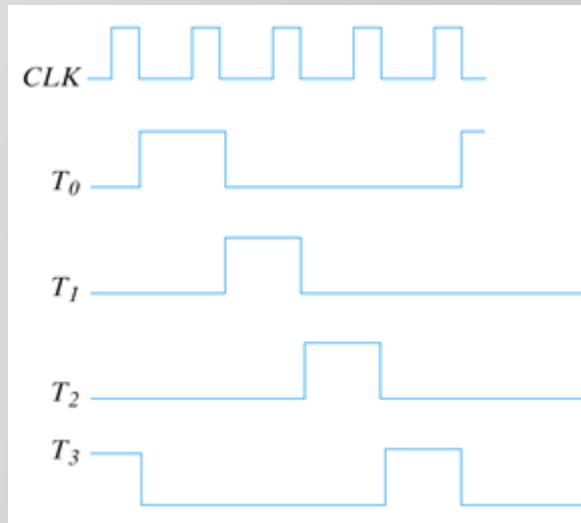
Στο Μετρητή δακτυλίου μόνο ένα FF έχει τιμή 1 ανά κατάσταση



Ring-counter (initial value = 1000) (Μετρητής δακτυλίου (Αρχική τιμή = 1000))



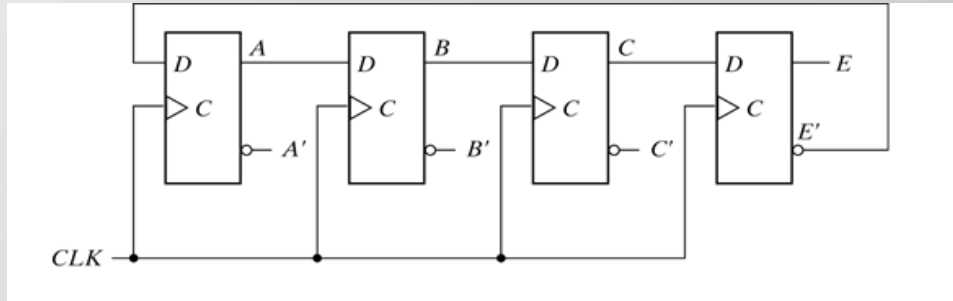
Counter and decoder (Μετρητής και κωδικοποιητής)



Sequence of four signals (Ακολουθία τεσσάρων σημάτων)



Μετρητής Δακτυλίου με αντιστροφή ούρας (switch-tail ring counter)



(a) Four-stage switch-tail ring counter (Τεσσάρων καταστάσεων διακόπτης-μετρητής δακτυλίου ούρας)

Sequence number	A	B	C	E	AND gate required for output
1	0	0	0	0	$A'E'$
2	1	0	0	0	AB'
3	1	1	0	0	BC'
4	1	1	1	0	CE'
5	1	1	1	1	AE
6	0	1	1	1	$A'B$
7	0	0	1	1	$B'C$
8	0	0	0	1	$C'E$

(b) Count sequence and required decoding (Μετρητής ακολουθίας και απαιτείται αποκωδικοποίηση)



Τέλος Ενότητας

