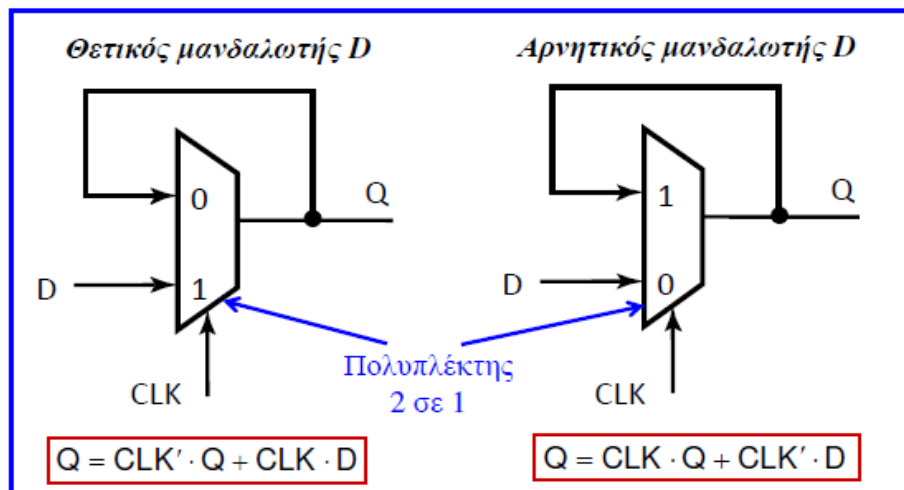
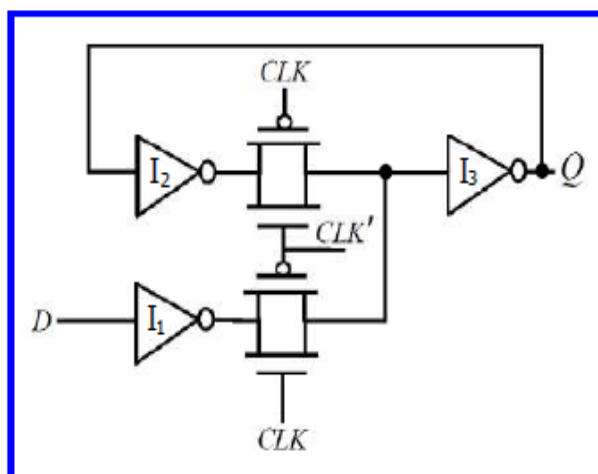


Στατικοί μανδαλωτές με πολυπλέκτες 2 σε 1

- Η λογική έκφραση που διέπει τη λειτουργία ενός μανδαλωτή D, είναι όμοια με τη λογική έκφραση που διέπει τη λειτουργία ενός πολυπλέκτη 2 σε 1.
- Με βάση την παρατήρηση αυτή, προκύπτει μια αποδοτική **υλοποίηση του στατικού μανδαλωτή D με χρήση πολυπλεκτών 2 σε 1**, οι οποίοι υλοποιούνται εύκολα με **πύλες διέλευσης**. Η υλοποίηση αυτή χρησιμοποιείται πολύ συχνά.

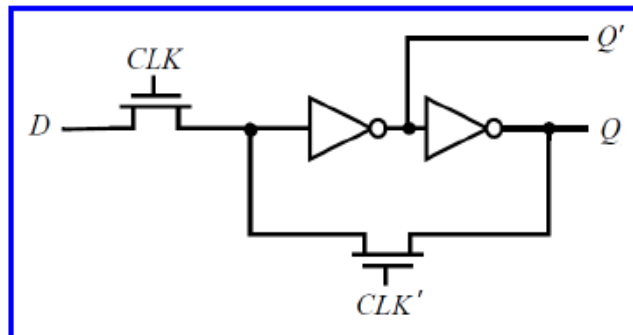


Η ανατροφοδότηση εξασφαλίζει αμετάβλητη έξοδο για όση διάρκεια το σήμα ρολογιού είναι στη χαμηλή στάθμη (θετικός μανδαλωτής) ή στην υψηλή στάθμη (αρνητικός μανδαλωτής)



Οι αντιστροφείς I_2 και I_3 σε σταυροειδή σύνδεση, διατηρούν αμετάβλητη την κατάσταση του μανδαλωτή, όταν το **σήμα ρολογιού βρίσκεται σε χαμηλή στάθμη**, ενώ ο αντιστροφέας I_1 εξασφαλίζει την ορθή λειτουργία του κυκλώματος.

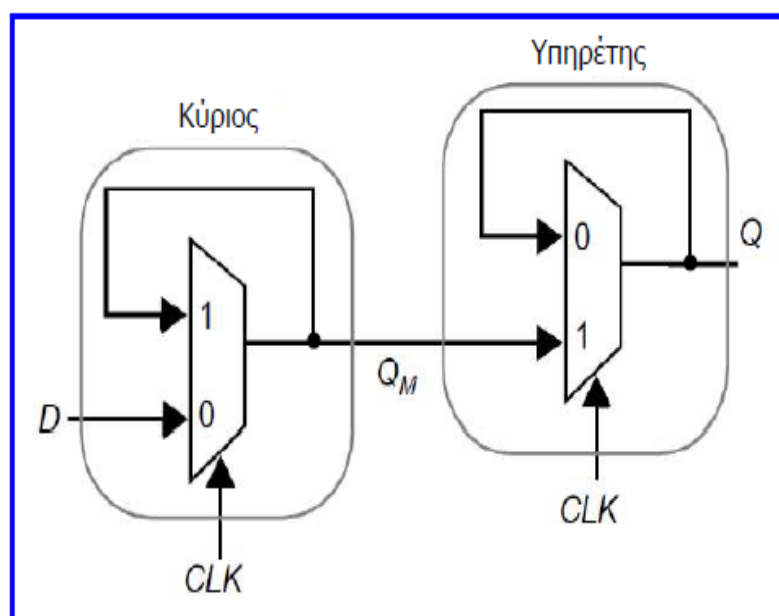
- Όταν το **σήμα ρολογιού είναι σε υψηλή στάθμη**, η κάτω πύλη διέλευσης άγει και η είσοδος D μεταφέρεται στην έξοδο Q.
- Κατά τη διάρκεια της φάσης αυτής, ο βρόχος ανατροφοδότησης είναι ανοιχτός, επειδή η άνω πύλη διέλευσης είναι σε αποκοπή.
- Επομένως, το μέγεθος των τρανζίστορ δεν είναι κρίσιμο για την ορθή υλοποίηση της επιθυμητής συνάρτησης του μανδαλωτή.
- Το πλήθος των τρανζίστορ που οδηγεί το σήμα ρολογιού είναι σχετικά υψηλό (4) και οδηγεί σε υψηλή κατανάλωση ενέργειας, αφού το σήμα ρολογιού έχει πιθανότητα μετάβασης ή δραστηριότητα μετάβασης (switching activity) ίση με 1.



- Είναι δυνατό να μειώσουμε το φορτίο του σήματος ρολογιού από 4 σε 2 τρανζίστορ, υλοποιώντας τον πολυπλέκτη μόνο με τρανζίστορ διέλευσης NMOS.
- Όταν το **σήμα ρολογιού είναι σε υψηλή στάθμη**, η είσοδος D μεταφέρεται στην έξοδο Q.
- Όταν το **σήμα ρολογιού είναι σε χαμηλή στάθμη**, ενεργοποιείται ο βρόχος ανατροφοδότησης, ώστε να διατηρηθεί αμετάβλητη η κατάσταση του μανδαλωτή.
- Ωστόσο, η χρήση μόνο τρανζίστορ διέλευσης NMOS έχει ως αποτέλεσμα τη διέλευση μίας υποβαθμισμένης υψηλής στάθμης τάσης $V_{DD} - V_{Tn}$ στην είσοδο του 1ου αντιστροφέα.
- Αυτό επιδρά αρνητικά στα περιθώρια θορύβου και στην επίδοση και προκαλεί στατική κατανάλωση ενέργειας στον 1ο αντιστροφέα, αφού το τρανζίστορ PMOS του αντιστροφέα δεν αποκόπτεται πλήρως.

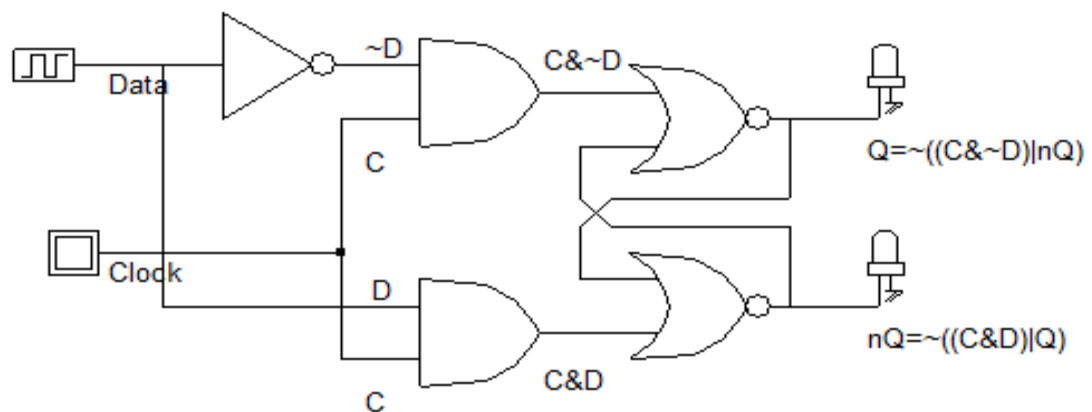
Ακμοπυροδοτούμενοι Καταχωρητές κύριου - υπηρέτη

- Η πιο συνηθισμένη προσέγγιση για την υλοποίηση ενός **στατικού ακμοπυροδοτούμενου καταχωρητή (edge-triggered register)** είναι η **διάταξη κύριου-υπηρέτη (master-slave)**.
- Η υλοποίηση βασίζεται στην τοποθέτηση **σε σειρά ενός αρνητικού μανδαλωτή (βαθμίδα κυρίου) και ενός θετικού μανδαλωτή (βαθμίδα υπηρέτη)**.

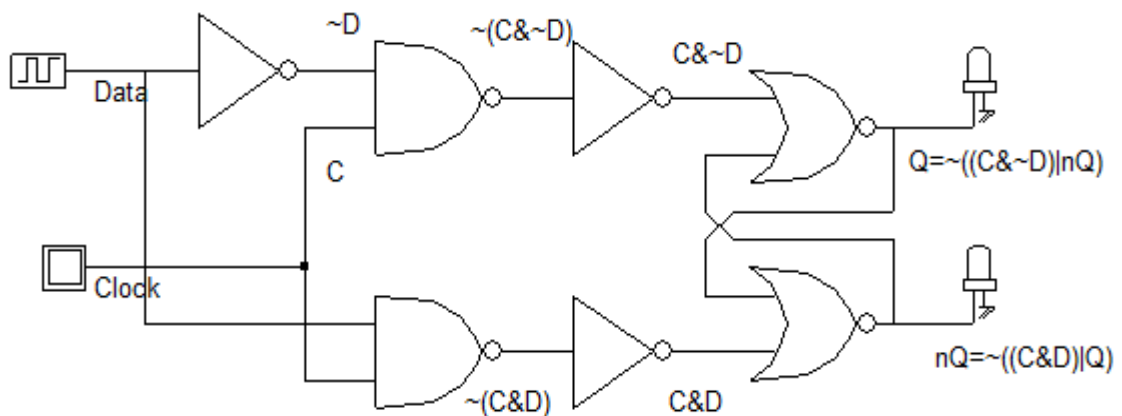


Θετικός στατικός ακμοπυροδοτούμενος καταχωρητής

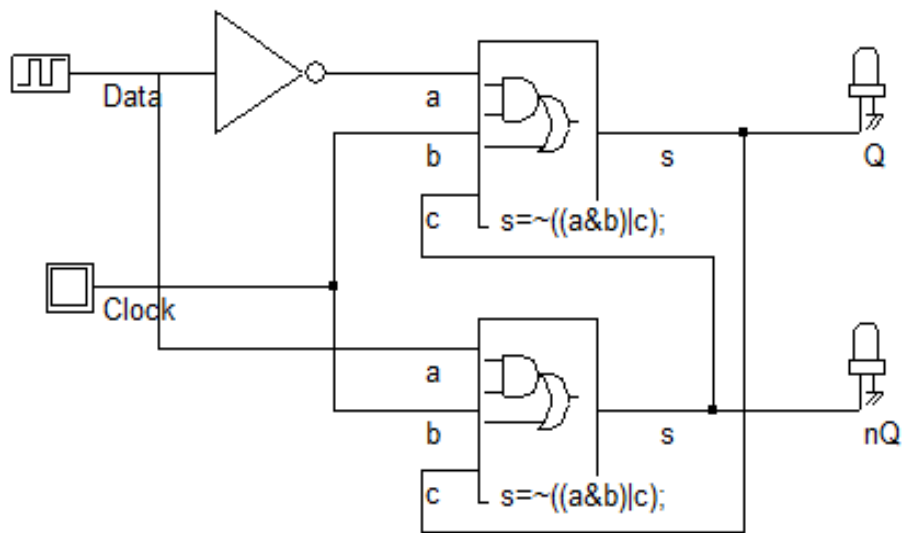
- Όταν το **σήμα ρολογιού είναι σε χαμηλή στάθμη**, η T_1 είναι σε αγωγή, η T_2 είναι σε αποκοπή και η είσοδος D μεταφέρεται στην έξοδο του κύριου μανδαλωτή Q_M .
- Η T_3 είναι σε αποκοπή, η T_4 είναι σε αγωγή και οι αντιστροφείς I_5 και I_6 σε σταυροειδή σύνδεση, διατηρούν αμετάβλητη την κατάσταση του υπηρέτη μανδαλωτή.
- Όταν το **σήμα ρολογιού μεταβαίνει στην υψηλή στάθμη**, η κύρια βαθμίδα σταματά να δειγματοληπτεί την είσοδο και περνά σε διατήρηση της κατάστασής της, η T_1 είναι σε αποκοπή, η T_2 σε αγωγή και οι αντιστροφείς I_2 και I_3 διατηρούν αμετάβλητη την Q_M .
- Η T_3 είναι σε αγωγή, η T_4 είναι σε αποκοπή και η Q_M μεταφέρεται στην έξοδο Q.



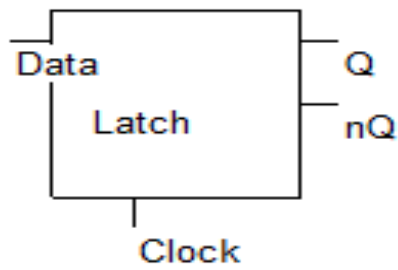
D-latch schematic diagram



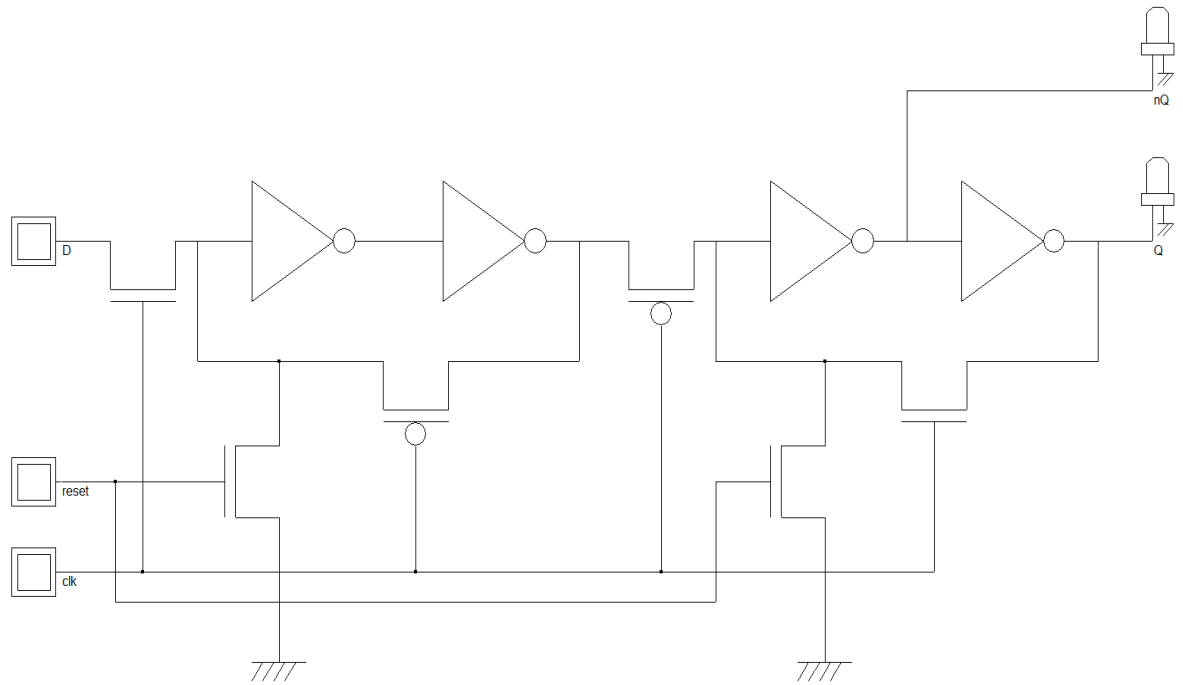
Direct CMOS implementation (poor solution, 22 transistors)



CMOS implementation with complex gates (Better design, 14 transistors)



Latch symbol



Τι είναι οι στατικοί καταχωρητές;

Οι **στατικοί καταχωρητές** υλοποιούνται με **στατικά flip-flops** (π.χ. **master-slave D flip-flops** ή **edge-triggered D FFs**) που διατηρούν την πληροφορία τους **στατικά**, δηλαδή όσο υπάρχει τροφοδοσία, χωρίς να χρειάζεται ανανέωση (σε αντίθεση με τους δυναμικούς καταχωρητές).

◆ Πώς λειτουργεί ο χρονισμός τους;

Ο χρονισμός καθορίζεται από:

1. Edge-triggering

Οι περισσότεροι στατικοί καταχωρητές καταχωρούν την τιμή στην:

- ανερχόμενη ακμή του ρολογιού (posedge), ή
- κατερχόμενη ακμή (negedge)

Δηλαδή η είσοδος **D** δειγματοληπτείται μόνο στη συγκεκριμένη ακμή και η έξοδος **Q** ενημερώνεται άμεσα μετά.

◆ Σημαντικές χρονικές παράμετροι

1. Setup time (t_{setup})

Ο χρόνος που η είσοδος **D** πρέπει να είναι σταθερή ΠΡΙΝ την ακμή του clock.

2. Hold time (t_{hold})

Ο χρόνος που η είσοδος **D** πρέπει να παραμείνει σταθερή ΜΕΤΑ την ακμή του clock.

3. Propagation delay (t_{pd} / $t_{\text{clk} \rightarrow \text{Q}}$)

Ο χρόνος που απαιτείται για να εμφανιστεί η νέα τιμή στην έξοδο **Q** μετά την ενεργή ακμή του clock.

◆ Συνθήκη σωστής λειτουργίας

Για να μην υπάρξουν σφάλματα (π.χ. metastability), πρέπει να ισχύει:

$$t_{data\ arrival} \leq t_{clock} - t_{setup}$$

και

$$t_{hold} \leq t_{data\ remain}$$

Σε αλυσίδες από καταχωρητές (pipelines), προστίθεται και ο delay της λογικής ανάμεσα.

◆ Πλεονεκτήματα στατικών καταχωρητών

- Σταθερή αποθήκευση (δεν χρειάζονται refresh)
 - Γρήγορη λειτουργία
 - Αξιόπιστος χρονισμός
 - Μεγάλη ανοχή σε θόρυβο
-

◆ Πού χρησιμοποιούνται;

- CPU registers
- pipeline registers
- flip-flop-based storage
- state machines (FSMs)
- counters / shift registers

Αναλυτικές εξισώσεις timing για ολόκληρο μονοπάτι

Παρακάτω δίνουμε τις **γεωμετρικές (και πρακτικές)** εξισώσεις που χρησιμοποιούνται για τον έλεγχο **setup** και **hold** σε ένα πλήρες timing path μεταξύ δύο στατικών flip-flops (πηγή → προορισμός). Θα βρούμε επίσης

ορισμούς, παραδείγματα για υπολογισμό του slack και σύντομες οδηγίες για worst-/best-case.

Ορισμοί (σύντομος λεξιλόγιος)

- t_{clk} ή T_{clk} : περίοδος ρολογιού.
- t_{CQ} : clock→Q propagation delay του source flip-flop (μερ. $t_{clk→Q}$).
- t_{comb} : καθαρό delay (worst-case) της συνδυαστικής λογικής ανάμεσα στα FFs.
- t_{comb}^{min} : ελάχιστο (best-case) delay της συνδυαστικής λογικής.
- t_{su} : setup time του destination FF.
- t_h : hold time του destination FF.
- t_{skew} : clock skew = $t_{clk,src} - t_{clk,dst}$ (θετικό αν το clock φτάνει νωρίτερα στο source).
- t_j ή t_{unc} : clock uncertainty/jitter/variation (συνήθως προστίθεται στο skew ή στο setup guard).
- margin / safety: επιπλέον περιθώριο σχεδίασης (π.χ. OCV/PVT margins).
- **Slack**: διαθέσιμος χρόνος = $T_{clk} -$ (αναγκαία καθυστέρηση).

Συνθήκη Setup (μέγιστες καθυστερήσεις — worst-case)

Για να περάσει σωστά ένα bit από source FF σε destination FF:

$$T_{clk} \geq t_{CQ}^{max} + t_{comb}^{max} + t_{su} + t_{skew} + t_{unc}$$

ή ισοδύναμα το **setup slack**:

$$\text{SetupSlack} = T_{clk} - (t_{CQ}^{max} + t_{comb}^{max} + t_{su} + t_{skew} + t_{unc})$$

Όπου:

- t_{CQ}^{max} : worst-case clock→Q (με αργές γωνίες λειτουργίας).
- t_{comb}^{max} : worst-case path delay (με αργά gates, worst PVT).
- t_{skew} προστίθεται με θετικό πρόσημο αν το skew μειώνει τον διαθέσιμο χρόνο (clock φτάνει πρώτα στο destination → negative skew θα βοηθούσε, θετικό χειροτερεύει).
- t_{unc} (uncertainty/jitter) συνήθως προστεθεί στο skew ή στο setup ως ξεχωριστός όρος.

Εάν $SetupSlack \geq 0$ τότε το path ικανοποιεί το setup.

Συνθήκη Hold (ελάχιστες καθυστερήσεις — best-case)

Για να μην αλλάξει η είσοδος του destination πριν περάσει ο hold χρόνος:

$$t_{CQ}^{min} + t_{comb}^{min} \geq t_h + t_{skew_hold} - t_{unc_hold}$$

Συνηθισμένη γραφή (με slack):

$$HoldSlack = (t_{CQ}^{min} + t_{comb}^{min}) - (t_h + t_{skew} + t_{unc})$$

Εάν $HoldSlack \geq 0$ τότε το path ικανοποιεί το hold.

Σημειώσεις:

- Για hold, χρησιμοποιούμε min τιμές για t_{CQ} και t_{comb} (fast-corners).
- Το t_{skew} μπορεί να εμφανιστεί με αντίθετο πρόσημο ανάλογα με την σύμβαση — εδώ υποθέτουμε ότι positive skew μειώνει το διαθέσιμο hold guard, οπότε το προσθέτουμε. Να ελέγχεις σύμβαση εργαλείου.

Λεπτομέρειες clock skew / uncertainty

Πιο ρεαλιστική μορφή είναι να διαχωρίσουμε skew και uncertainty:

$$T_{clk} \geq t_{CQ}^{max} + t_{comb}^{max} + t_{su} + t_{skew_pos} + t_{unc}$$

όπου $t_{skew_pos} = \max(0, t_{clk,dst} - t_{clk,src})$.

(Στα EDA flows συνήθως μεταχειρίζονται $t_{skew} + t_{unc}$ σαν έναν όρο $t_{clk_arrival_uncertainty}$.)

Πολυβάθμιο (multi-stage) μονοπάτι / Pipeline με λογική ανάμεσα

Για path που περνά πάνω από πολλές ενδιάμεσες πηγές/προορισμούς, κάθε flip-flop πρέπει να ικανοποιεί τις εξισώσεις χωριστά. Για end-to-end latency interest (π.χ. latency-through N stages) το άθροισμα των delays δίνει:

$$\text{TotalLatency} = \sum_{\text{stages}} (t_{CQ_i} + t_{comb_i})$$

αλλά **timing constraints** ελέγχονται κατά χαμηλού επιπέδου ανά FF.

Slack — πώς το υπολογίζεις και τι σημαίνει

- **SetupSlack** θετικό: περίσσεια χρόνου — μπορείς να αυξήσεις λογική ή να μειώσεις Tclk.
- **HoldSlack** θετικό: δεν υπάρχει race / early arrival.

Υπολογισμός (setup example):

$$\text{SetupSlack} = T_{clk} - (t_{CQ}^{max} + t_{comb}^{max} + t_{su} + t_{skew} + t_{unc})$$

Αν θελω να βρω το **μέγιστο ρολόι (min Tclk)** που επιτρέπεται:

$$T_{clk}^{min} = t_{CQ}^{max} + t_{comb}^{max} + t_{su} + t_{skew} + t_{unc}$$

Παραδείγματα (γρήγορος αριθμητικός υπολογισμός)

Παράδειγμα (τυπικά νούμερα, απλώς για επίδειξη):

- $t_{CQ}^{max} = 100$ ps
- $t_{comb}^{max} = 400$ ps
- $t_{su} = 100$ ps
- $t_{skew} = 50$ ps
- $t_{unc} = 20$ ps

Τότε

$$T_{clk}^{min} = 100 + 400 + 100 + 50 + 20 = 670 \text{ ps} \Rightarrow f_{max} \approx 1/0.67 \text{ ns} \approx 1.49 \text{ GHz}$$

(Αυτό είναι απλό παράδειγμα — στην πράξη χρησιμοποιούμε worst-case corners.)

Πρακτικές προσθήκες που συχνά λαμβάνονται υπόψη

- OCV/IR drop/PVT margins: προστίθενται στο t_{comb}^{max} ή στο t_{unc} .
- Multi-corner analysis: υπολογίζεις setup στο slow-slow corner και hold στο fast-fast corner.
- Paths με πολλαπλά εξόδους/πολυπλοκότητα: εξετάζεις critical path (max) και shortest path (min).
- Μεταβλητότητα (on-chip variation) μπορεί να εισάγει επιπλέον guard bands.
- Για asynchronous clock domains: χρειάζονται synchronizers — οι παραπάνω εξισώσεις δεν αρκούν.

Σύντομη λίστα βημάτων για έλεγχο ενός path

1. Βρες t_{CQ}^{max} (vendor datasheet ή characterization).
2. Πλήρωσε t_{comb}^{max} (STA tool / sum gate delays worst-case).
3. Επέλεξε t_{skew} , t_{unc} ανάλογα με την τοπολογία ρολογιού.
4. Υπολόγισε T_{clk}^{min} από την setup εξίσωση.
5. Έλεγξε hold με min τιμές t_{CQ}^{min} , t_{comb}^{min} .
6. Υπολόγισε slack και βρες critical και short paths.

1) Χρόνος αποκατάστασης — Setup time (t_{su})

Ορισμός: ο ελάχιστος χρόνος πριν την ενεργή ακμή του ρολογιού που η είσοδος D ενός flip-flop πρέπει να είναι σταθερή (χωρίς αλλαγές) ώστε η τιμή να δειγματοληπτηθεί σωστά.

Σκοπός: αποφυγή metastability / λανθασμένου δειγματοληπτικού αποτελέσματος.

Σχέση στο μονοπάτι (setup constraint):

$$T_{clk} \geq t_{CQ}^{max} + t_{comb}^{max} + t_{su} + t_{skew} + t_{unc}$$

όπου:

- T_{clk} περίοδος ρολογιού,
- t_{CQ}^{max} = max clock→Q delay του source FF,
- t_{comb}^{max} = worst-case καθυστέρηση της συνδυαστικής λογικής,
- t_{skew} θετικό αν το clock φτάνει νωρίτερα στο destination (μειώνει διαθέσιμο χρόνο),
- t_{unc} jitter/uncertainty.

Setup slack:

$$\text{SetupSlack} = T_{clk} - (t_{CQ}^{max} + t_{comb}^{max} + t_{su} + t_{skew} + t_{unc})$$

Αν $\text{SetupSlack} \geq 0 \rightarrow$ ικανοποιείται.

2) Χρόνος συγκράτησης — Hold time (t_h)

Ορισμός: ο ελάχιστος χρόνος μετά την ενεργή ακμή που η είσοδος D πρέπει να παραμείνει σταθερή ώστε το destination FF να μη «πιάσει» μεταβατική (πρώιμη) τιμή.

Σκοπός: αποφυγή race conditions όπου η νέα τιμή φτάνει «πολύ γρήγορα» και αλλάζει την είσοδο πριν ολοκληρωθεί ο hold χρόνος.

Σχέση στο μονοπάτι (hold constraint):

$$t_{CQ}^{min} + t_{comb}^{min} \geq t_h + t_{skew} + t_{unc}$$

ή ως slack:

$$\text{HoldSlack} = (t_{CQ}^{min} + t_{comb}^{min}) - (t_h + t_{skew} + t_{unc})$$

Αν $\text{HoldSlack} \geq 0 \rightarrow$ ικανοποιείται.

Σημείωση: για hold χρησιμοποιούμε min τιμές (fast corners) για t_{CQ} και t_{comb} .

3) Καθυστέρηση διάδοσης — Propagation delay (t_{pd})

Ορισμός (γενικά): χρόνος από την αλλαγή εισόδου μέχρι την αλλαγή εξόδου σε ένα στοιχείο/πύλη/FF.

Υπάρχουν δύο σημαντικές ειδικές εκδοχές που χρησιμοποιούνται στην ανάλυση timing:

- **Clock→Q propagation delay** (t_{CQ} ή $t_{clk \rightarrow Q}$): χρόνος από την ενεργή ακμή ρολογιού στο source flip-flop μέχρι να εμφανιστεί η νέα τιμή στην έξοδο Q . Χρήσιμο στην setup εξίσωση (χρειάζεται η μέγιστη τιμή t_{CQ}^{max} για worst-case).
- **Combinational propagation delay** (t_{comb} ή $t_{pd,comb}$): συνολική καθυστέρηση της λογικής που συνδέει τα FFs — συνάθροισμα καθυστερήσεων πυλών, buffers, wires, κ.λπ. Χρησιμοποιείται ως t_{comb}^{max} (worst) στη setup εξίσωση και t_{comb}^{min} (best/fast) στην hold.

Μονάδες: συνήθως ps ή ns.

Αριθμητικό παράδειγμα — Setup (βήμα-βήμα)

Δεδομένα:

- $t_{CQ}^{max} = 100$ ps
- $t_{comb}^{max} = 600$ ps
- $t_{su} = 80$ ps
- $t_{skew} = 20$ ps
- $t_{unc} = 10$ ps
- $T_{clk} = 810$ ps (περίοδος)

Υπολογισμοί (αθροίζουμε τις απαιτήσεις):

1. $t_{CQ}^{max} + t_{comb}^{max} = 100 + 600 = 700$ ps.
2. Προσθέτουμε t_{su} : $700 + 80 = 780$ ps.
3. Προσθέτουμε skew: $780 + 20 = 800$ ps.
4. Προσθέτουμε uncertainty: $800 + 10 = 810$ ps.

Άρα η απαιτούμενη περίοδος $T_{clk}^{min} = 810$ ps.

$SetupSlack = T_{clk} - 810 = 810 - 810 = 0$ ps → οριακά ικανοποιημένο (χωρίς περιθώριο).

Συχνά υπολογίζουμε $f_{max} = 1/T_{clk}^{min} = 1/0.810$ ns ≈ 1.2345679 GHz.

Αριθμητικό παράδειγμα — Hold (βήμα-βήμα)

Δεδομένα (fast corner):

- $t_{CQ}^{min} = 20$ ps
- $t_{comb}^{min} = 50$ ps
- $t_h = 50$ ps
- $t_{skew} = 20$ ps
- $t_{unc} = 10$ ps

Υπολογισμοί:

1. Αθροισμα source arrival (min): $t_{CQ}^{min} + t_{comb}^{min} = 20 + 50 = 70$ ps.
2. Απαιτούμενο hold: $t_h + t_{skew} + t_{unc} = 50 + 20 + 10 = 80$ ps.
3. $HoldSlack = 70 - 80 = -10$ ps → **violation** 10 ps (το σήμα φτάνει πολύ νωρίς, απαιτείται επιβράδυνση).