



*Ανάπτυξη παιδιού. Ανάπτυξη για όλους.*

ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ



ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ  
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ  
Επιχειρησιακό Πρόγραμμα  
Εκπαίδευσης και Αρχικής  
Επαγγελματικής Κατάρτισης

---

# Αλγόριθμοι Εκτίμησης Καθυστέρησης και Βελτιστοποίησης

---

---

# Εισαγωγή

*Το κύριο πρόβλημα στην σχεδίαση κυκλωμάτων είναι η επίτευξη της μέγιστης απόδοσης για την δεδομένη τεχνολογία.*

## **Μεγιστοποίηση απόδοσης:**

- (α) Μείωση της καθυστέρησης διάδοσης από τις εισόδους στις εξόδους.
- (β) Για ακολουθιακά κυκλώματα μείωση του κύκλου.

Η εκτίμηση της απόδοσης μπορεί να γίνει με εξομοίωση αλλά είναι εξαιρετικά δύσκολη στα μεγάλα κυκλώματα λόγω μεγάλου αριθμού I/O.

## **Χρονική Ανάλυση.**

Υπολογισμός της καθυστέρησης στατικά από ένα λογικό δίκτυο.

---

# Εισαγωγή

- ✓ Στα κυκλώματα πολλαπλών επιπέδων υπάρχουν πολλά trade-off points ανάμεσα σε επιφάνεια και καθυστέρηση.
- ✓ Το γρηγορότερο κύκλωμα δεν είναι και το μικρότερο.
- ✓ Συνήθως ο σχεδιασμός γίνεται με περιορισμό της μίας παραμέτρου και βελτιστοποίηση της άλλης.

## **Bounded δίκτυο:**

- κάθε κορυφή να αντιστοιχεί σε ένα cell με γνωστή καθυστέρηση.
- Μπορεί να ληφθεί υπόψη το Fanout αλλά και να διαφοροποιηθεί η μετάβαση ανόδου από την μετάβαση καθόδου.

## **Unbounded δίκτυο:**

- Η εκτίμηση είναι λιγότερο ακριβής.
- Κάθε τοπική συνάρτηση υλοποιείται από μία νοητή πύλη.

# Μοντελοποίηση καθυστέρησης

---

- Ένα απλό μοντέλο είναι η χρήση των unit delays για κάθε stage.
  - Υπάρχουν και πιο ακριβή μοντέλα όπως η εκτίμηση μιας ελάχιστης παραγοντοποιημένης συνάρτησης.
- 
- ✓ Θεωρούμε την καθυστέρηση μίας κορυφής ως έναν θετικό αριθμό.
  - ✓ Προτιμάμε την εκτίμηση χειρότερης περίπτωσης για να έχουμε την βεβαιότητα λειτουργίας του κυκλώματος.
  - ✓ **Data Ready/Arrival time:** είναι ο χρόνος στον οποίο σταθεροποιείται ένα σήμα εξόδου μίας κορυφής.
  - ✓ Θεωρούμε τους χρόνους data ready των κυρίων εισόδων ίσους με μηδέν.
  - ✓ χρόνος data ready κορυφής = χρόνος data ready της πιο καθυστερημένης εισόδου συν την καθυστέρηση διάδοσης της κορυφής.  $t_i = d_i + \max(t_j)$

# Μοντελοποίηση καθυστέρησης

---

- ✓ Και οι έξοδοι αντιστοιχούν σε κορυφές οπότε μπορούν να έχουν καθυστέρηση μοντελοποιώντας τις θύρες εξόδου.
- ✓ Όλοι οι χρόνοι data ready μπορούν να υπολογιστούν με προς τα εμπρός διαπέραση του λογικού δικτύου.

## **Τοπολογική κρίσιμη καθυστέρηση:**

- Ο μέγιστος data-ready χρόνος που εμφανίζεται σε κάποια έξοδο του κυκλώματος.
- Αντιστοιχεί στο μονοπάτι με το μεγαλύτερο βάρος όπου βάρος είναι η καθυστέρηση κάθε κορυφής.
- Το μονοπάτι αυτό ονομάζεται **τοπολογικά κρίσιμο**.

## **Χρονική Βελτιστοποίηση:**

- Αναδόμηση του λογικού δικτύου για ικανοποίηση ορίων χρόνων data ready.
- Ορίζουμε τον data ready χρόνο σε μία έξοδο και τον συμβολίζουμε με  $t$ .

# Μοντελοποίηση καθυστέρησης

---

## Slack:

- Είναι η διαφορά του απαιτούμενου data ready χρόνου και του πραγματικού data ready χρόνου.
  - Slack=0 ο απαιτούμενος data ready χρόνος ταυτίζεται με τον πραγματικό
  - Slack>0 ο απαιτούμενος data ready χρόνος είναι μετά τον πραγματικό οπότε υπάρχει περιθώριο αναμονής
  - Slack<0 ο απαιτούμενος data ready χρόνος είναι πριν τον πραγματικό οπότε υπάρχει παραβίαση περιορισμού
- Οι απαιτούμενοι data-ready χρόνοι και τα slacks υπολογίζονται με προς τα πίσω διατέραςη του δικτύου.

**Κρίσιμα μονοπάτια:** Μονοπάτια που καταλήγουν σε κορυφές με 0-slack στις εξόδους του κυκλώματος.



Οι απαιτούμενοι data ready χρόνοι είναι ίσοι με τον μέγιστο data ready χρόνο.

# Μοντελοποίηση καθυστέρησης

## Παράδειγμα

Χρόνοι Data ready κυρίων εισόδων:  $t_a=0$ ,  $t_b=10$ .

Καθυστερήσεις εσωτερικών κορυφών:  $d_g=d_l=d_y=3$ ,  $d_h=8$ ,  $d_m=1$ ,  $d_k=10$ ,  $d_n=5$ ,

$$d_p=d_q=d_x=2$$

$$t_g=3+0=3$$

$$t_h=8+3=11$$

$$t_k=10+3=13$$

$$t_n=5+10=15$$

$$t_p=2+\max\{15,3\}=17$$

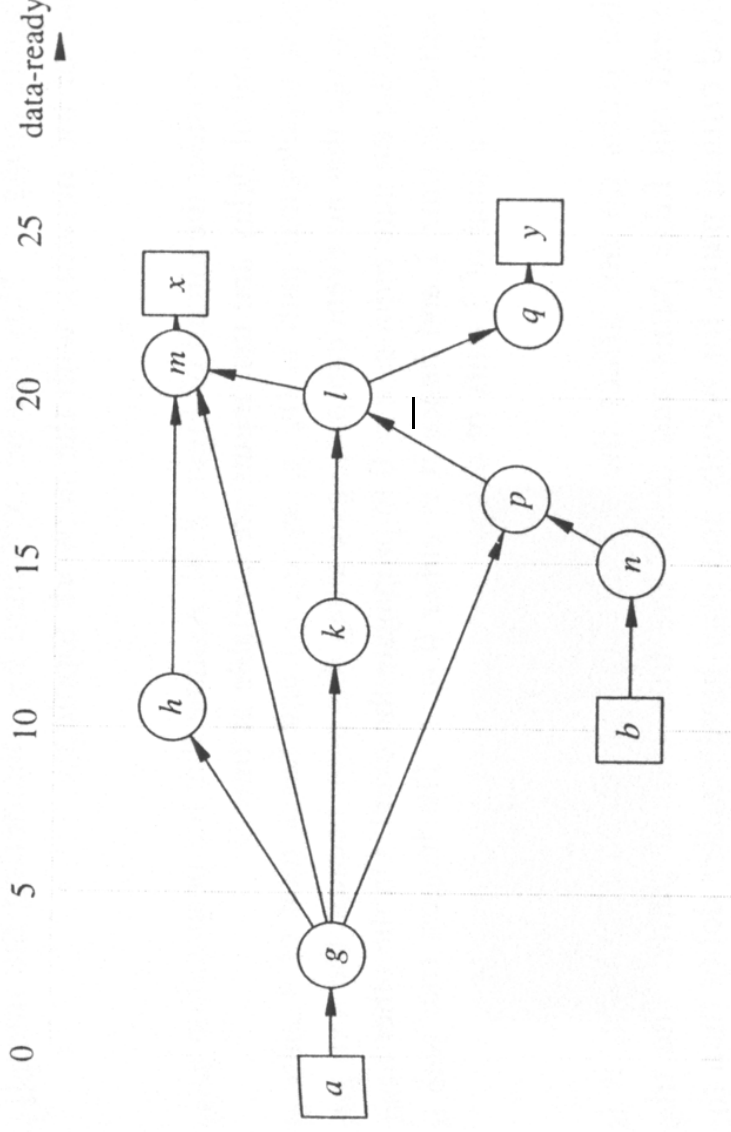
$$t_l=3+\max\{13,17\}=20$$

$$t_m=1+\max\{3,11,20\}=21$$

$$t_x=2+21=23$$

$$t_q=2+20=22$$

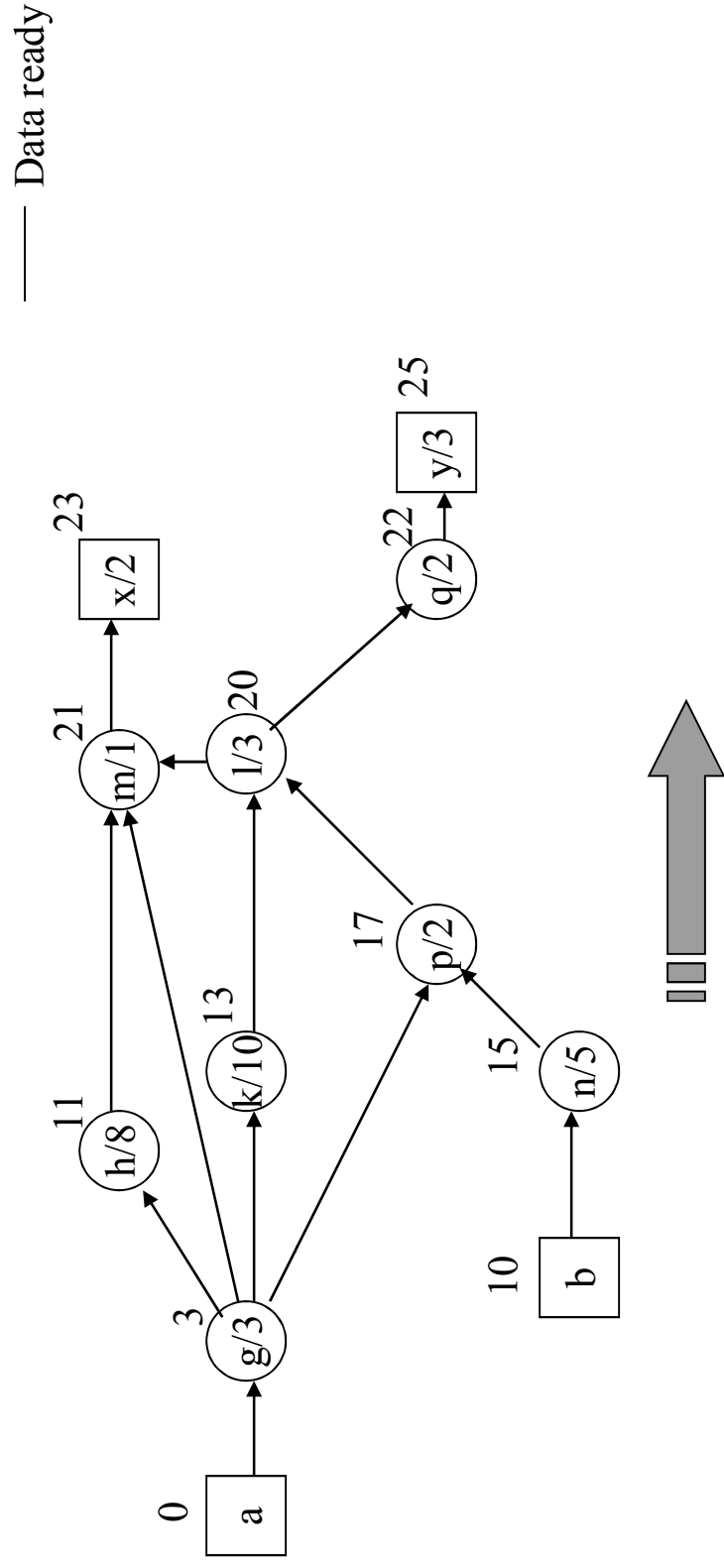
$$t_y=3+22=25$$



# Μοντελοποίηση καθυστέρησης

---

$t_i = \max(t_j) + d_j$   Η πιο αργή από τις εισερχόμενες ακμές + καθυστέρηση κόμβου



*Με εμπρός διαπέραση του γράφου υπολογίζονται οι data ready χρόνοι.*

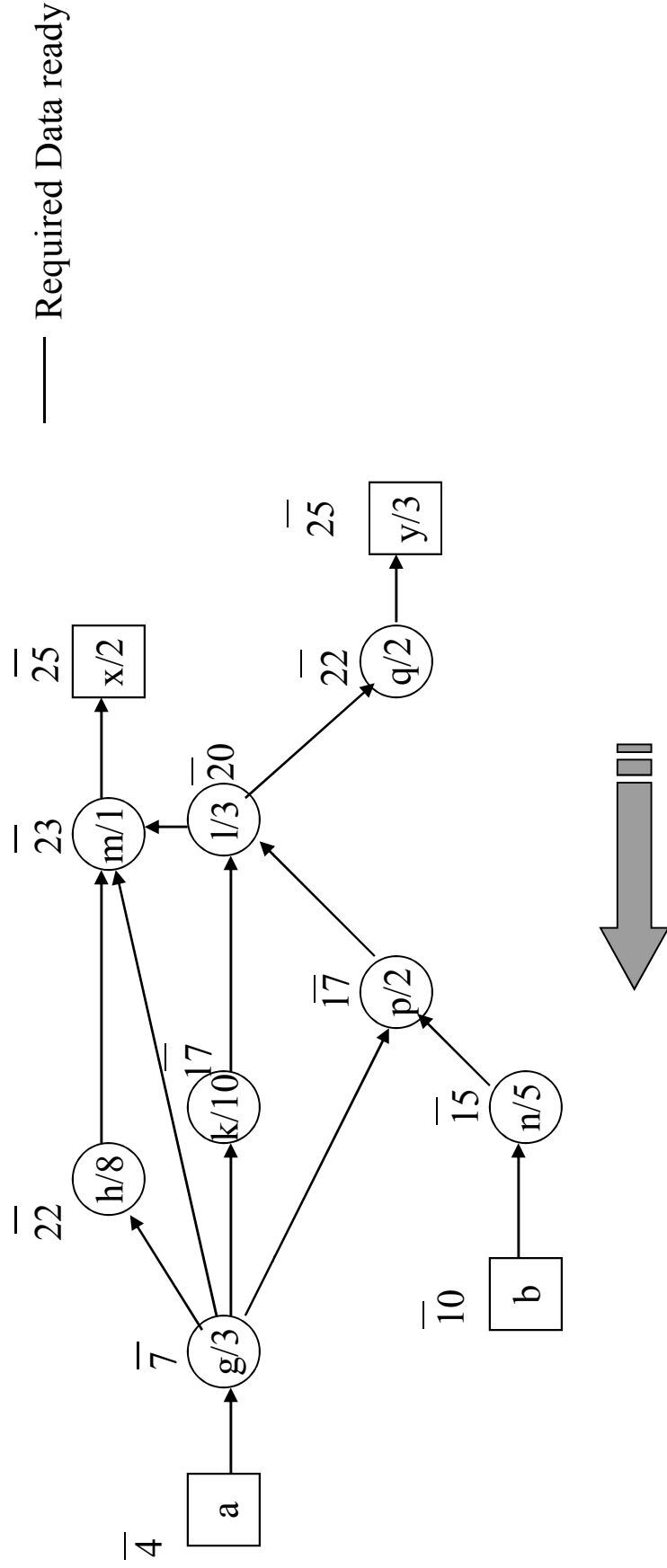


# Μοντελοποίηση καθυστέρησης

---

$$\bar{t}_i = \min(\bar{t}_j) - d_j \quad \Rightarrow$$

Η νωρίτερα απαιτούμενη από τις εξερχόμενες ακμές - καθυστέρηση κόμβου

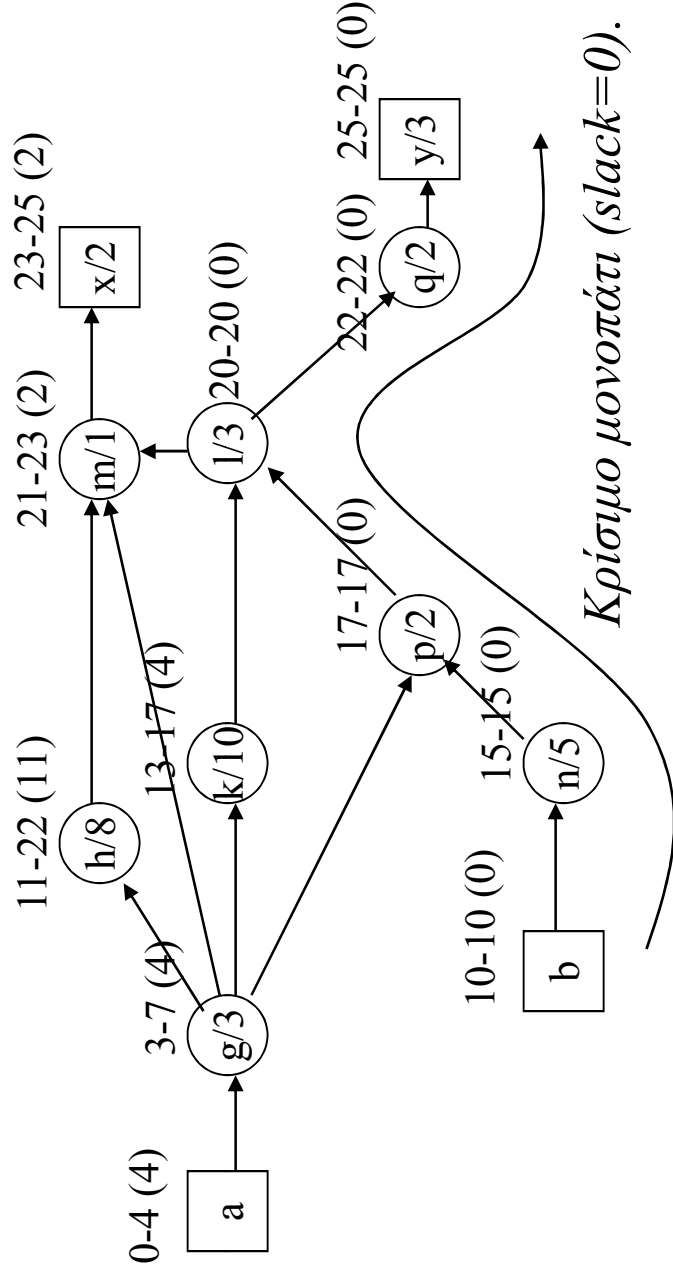


*Με πίσω διαπέραση του γράφου υπολογίζονται οι required data ready χρόνοι.*

# Μοντελοποίηση καθυστέρησης



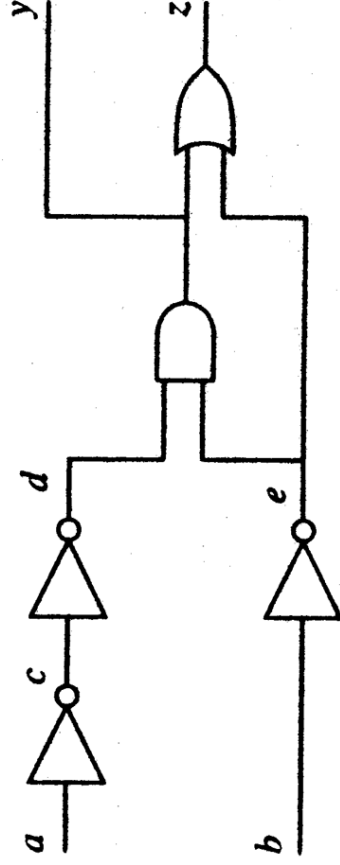
Με αφαίρεση χρόνων *required data ready* – *data ready* υπολογίζεται το *slack*.



# Λανθάνοντα Μονοπάτια

---

- ✓ Ένα τοπολογικά κρίσιμο μονοπάτι μπορεί να οδηγήσει σε κακή εκτίμηση της καθυστέρησης εάν είναι λανθάνων (δεν μεταδίδει γεγονότα).



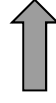
Καθυστέρηση/πύλη=1 unit

Είσοδοι διαθέσιμοι στο 0.

Τοπολογικά κρίσιμο:  $V_a, V_c, V_d, V_y, V_z$

Για  $b=0 \Rightarrow z=1$  ανεξάρτητα του  $a$

Για  $b=1 \Rightarrow z=0$  ανεξάρτητα του  $a$



Αν και τοπολογικά κρίσιμο το

$V_a, V_c, V_d, V_y, V_z$  δεν μεταδίδει γεγονότα

# Λανθάνοντα Μονοπάτια

---

- ✓ Τα λανθάνοντα μονοπάτια δεν επηρεάζουν την απόδοση του κυκλώματος και πρέπει να μαρκάρονται.
- ✓ **Sensitizable path**: ένα μονοπάτι που μπορεί να διαδώσει ένα γεγονός.
- ✓ **Critical path**: ένα sensitizable path με μέγιστο βάρος.
- ✓ **Παράπλευροι είσοδοι**: οι είσοδοι των κορυφών ενός μονοπατιού οι οποίες όμως δεν ανήκουν στο μονοπάτι.

## Dynamic Sensitization.

Ένα γεγονός μεταδίδεται από ένα μονοπάτι  $P=(v_{x0}, v_{x1}, \dots, v_{xm})$  όταν για κάθε ζεύγος διαδοχικών κορυφών  $x_i, x_{i-1}$  ισχύει  $\partial f_{x_i} / \partial x_{i-1} = 1$ .

## Static Sensitization.

Ένα  $P=(v_{x0}, v_{x1}, \dots, v_{xm})$  είναι statically sensitizable όταν υπάρχει μία ανάθεση των κυρίων εισόδων  $t$  για την οποία ισχύει  $\partial f_{x_i}(t) / \partial x_{i-1} = 1$ .

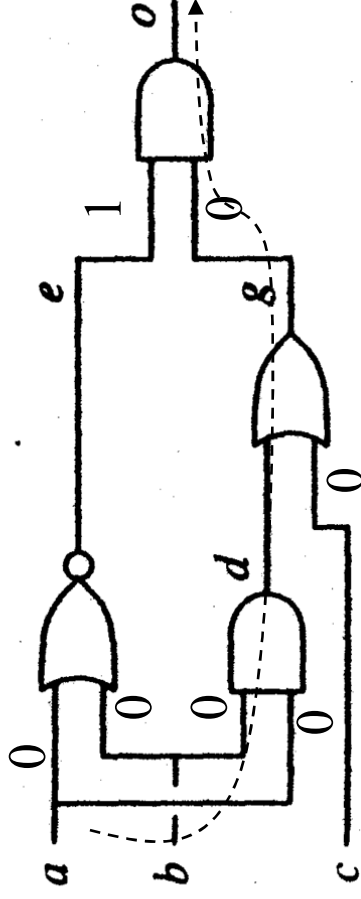
Η συνθήκη αυτή δεν αρκεί για να χαρακτηρίσει ένα λανθάνων μονοπάτι.

---

# Sensitization

---

- ✓ Η **controlling** τιμή μίας πύλης είναι η τιμή που αν εφαρμοστεί σε οποιαδήποτε είσοδο της καθορίζει την έξοδο της πύλης ανεξάρτητα από τις άλλες εισόδους.
- ✓ Μία πύλη έχει **controlled** τιμή όταν κάποια από τις εισόδους της έχει controlling τιμή.
- ✓ Ένα διάνυσμα προκαλεί static co-sensitization σε ένα μονοπάτι ( $v_{x0}, v_{x1}, \dots, v_{xm}$ ) όταν κάθε controlled τιμή στην έξοδο μίας πύλης του μονοπατιού οφείλεται σε controlling τιμή της εισόδου εκείνης που ανήκει στο μονοπάτι.



( $V_a, V_d, V_g, V_o$ )

Stat. Co-sens. στο 0 με το διάνυσμα  
 $a=0, b=0, c=0$

# Sensitization

---

- ✓ Το static co-sensitization είναι αναγκαία συνθήκη για να μην είναι λανθάνων ένα μονοπάτι.
- ✓ Για να μπορεί το statically co-sensitized μονοπάτι να καθορίζει την τιμή εξόδου θα πρέπει
  - (α) σε όλες τις controlled πύλες που οδηγεί να παρέχει την πρώτη σε χρόνο controlling είσοδο (που καθορίζει την έξοδο), και
  - (β) σε όλες τις υπόλοιπες την τελευταία μη-controlling είσοδο (που καθορίζει την έξοδο).

# Sensitization

---

✓ Ένα μονοπάτι είναι λανθάνων εάν για όλα τα διανύσματα εισόδου μία από τις επόμενες συνθήκες είναι αληθής:

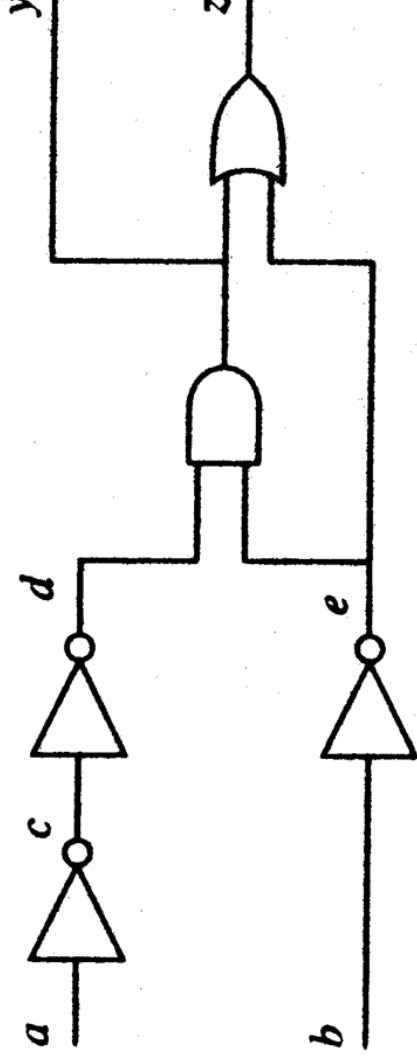
- (1) Μία πύλη στο μονοπάτι έχει controlled τιμή η οποία οφείλεται σε παράπλευρη είσοδο (η είσοδος του μονοπατιού δίνει non-controlling τιμή).
- (2) Μία πύλη στο μονοπάτι έχει controlled τιμή η οποία οφείλεται στο μονοπάτι και σε παράπλευρη είσοδο (η είσοδος του μονοπατιού δίνει controlling τιμή αλλά μετά την παράπλευρη είσοδο).
- (3) Μία πύλη στο μονοπάτι δεν έχει controlled τιμή (η παράπλευρη είσοδος δίνει non-controlling τιμή τελευταία).

---

# Sensitization

## Παράδειγμα

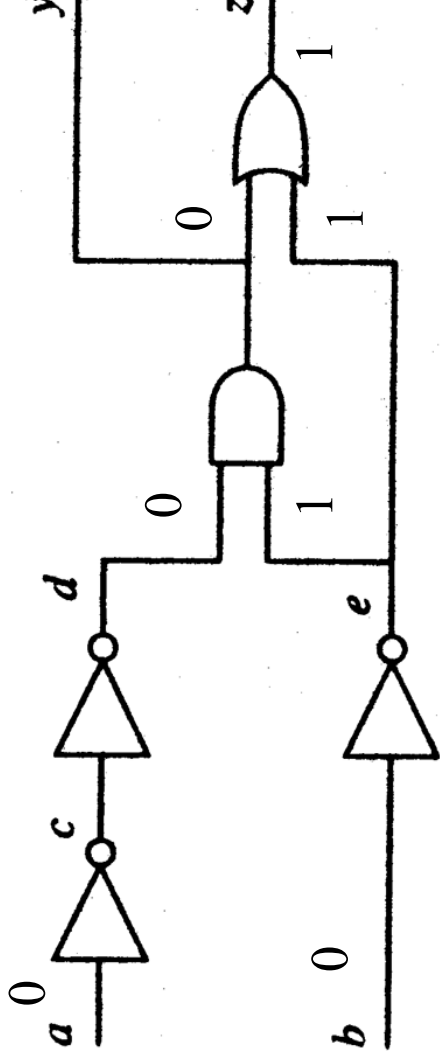
Εξέταση μονοπατιού  $V_a, V_c, V_d, V_y, V_z$





---

# Sensitization



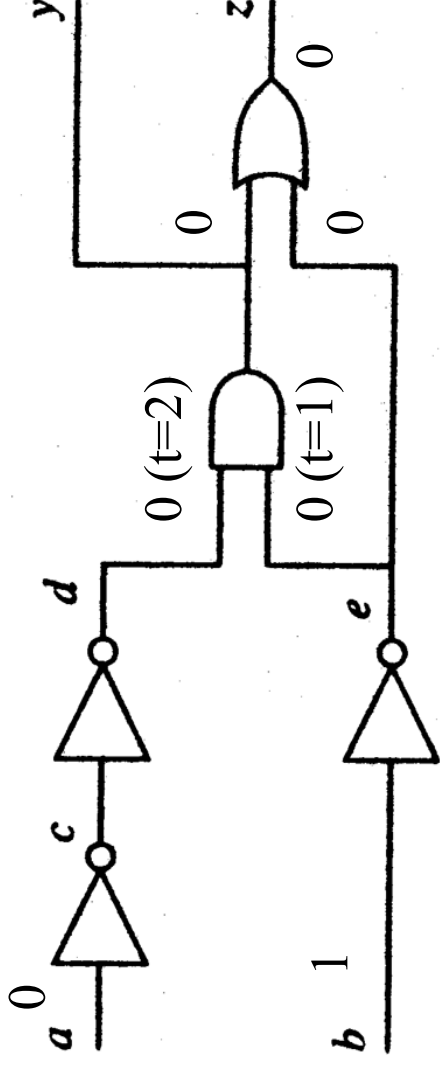
Περίπτωση 1<sup>η</sup> :  $ab=00$ , συνθήκη 1 στην πύλη OR:

*Η πύλη OR στο μονοπάτι έχει controlled τιμή (1) η οποία οφείλεται σε παράπλευρη είσοδο (e) (η είσοδος του μονοπατιού γ δίνει non-controlling τιμή 0).*

---

# Sensitization

---

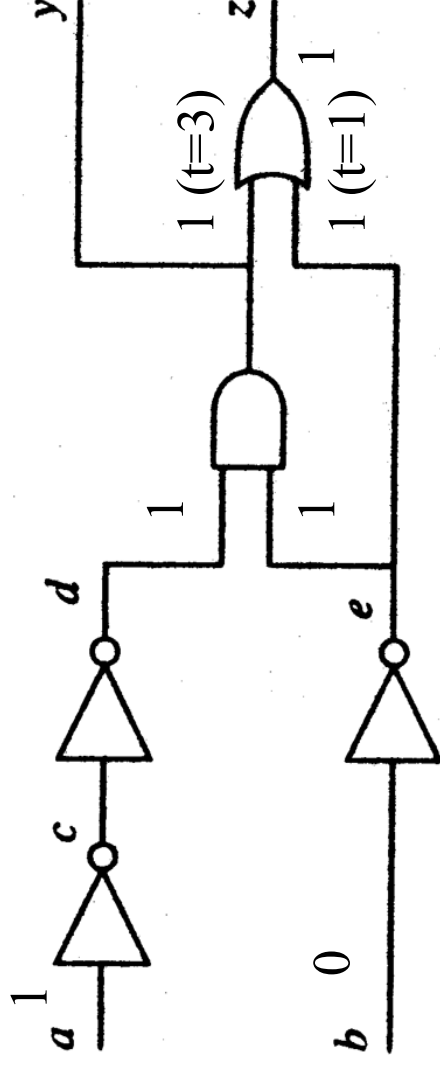


Περίπτωση 2<sup>η</sup> :  $ab=01$ , συνθήκη 2 στην πύλη And :

Η πύλη AND στο μονοπάτι έχει *controlled* τιμή (0) η οποία οφείλεται στο μονοπάτι ( $d$ ) και σε παράπλευρη είσοδο ( $e$ ) (η είσοδος του μονοπατιού ( $d$ ) δίνει *controlling* τιμή 0 αλλά μετά την παράπλευρη είσοδο  $e$ ).

---

# Sensitization

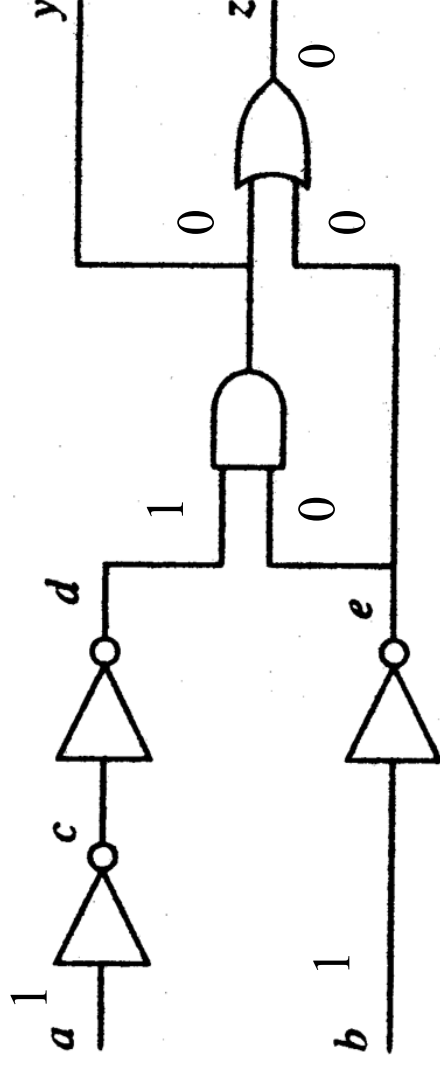


Περίπτωση 3<sup>η</sup> :  $ab=10$ , συνθήκη 2 στην πύλη Or :

*Η πύλη OR στο μονοπάτι έχει controlled τιμή (1) η οποία οφείλεται στο μονοπάτι (y) και σε παράπλευρη είσοδο (e) (η είσοδος του μονοπατιού y δίνει controlling τιμή αλλά μετά την παράπλευρη είσοδο e).*

---

# Sensitization

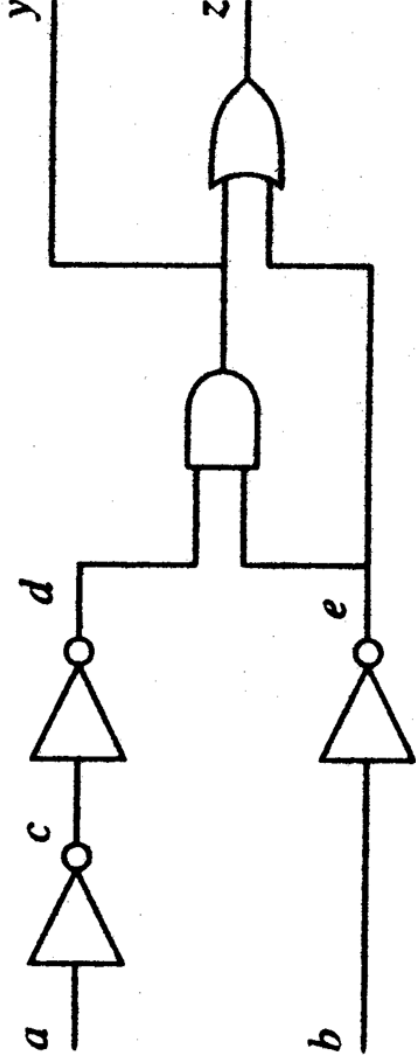


Περίπτωση 4<sup>η</sup> :  $ab=11$ , συνθήκη 1 στην πύλη And :

*Η πύλη AND στο μονοπάτι έχει controlled τιμή (0) η οποία οφείλεται σε παράπλευρη είσοδο (e) (η είσοδος του μονοπατιού d δίνει non-controlling τιμή 1).*

---

# Sensitization



Μονοπάτι:  $V_a, V_c, V_d, V_y, V_z$

- $ab=00$ , συνθήκη 1 στην πύλη Or
- $ab=01$ , συνθήκη 2 στην πύλη And
- $ab=10$ , συνθήκη 2 στην πύλη Or
- $ab=11$ , συνθήκη 1 στην πύλη And

Λαμβάνων  
Μονοπάτι

# Sensitization

---

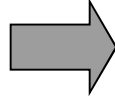
## **Ισοδύναμα προβλήματα :**

- ✓ Εάν ένα κύκλωμα μπορεί να λειτουργήσει σε μία δεδομένη ταχύτητα ή
- ✓ Εάν ένα κύκλωμα έχει κρίσιμο μονοπάτι με μικρότερη καθυστέρηση ή
- ✓ Εάν όλα τα μονοπάτια με μεγαλύτερη καθυστέρηση είναι λανθάνοντα.

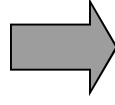
# Αλγόριθμοι και μετασχηματισμοί

---

**Στόχος:** Η ελαχιστοποίηση της καθυστέρησης ενός κυκλώματος



Όσο μικρότερη είναι η διαφορά ανάμεσα στο κρίσιμο μονοπάτι και το αργότερο από τα υπόλοιπα, τόσο μικρότερη είναι η βελτίωση που επιτυγχάνουμε εάν ασχοληθούμε μόνο με το κρίσιμο μονοπάτι.



Θα πρέπει να λάβουμε υπόψη ένα μεγαλύτερο σύνολο μονοπατιών από το κρίσιμο (πολλά μονοπάτια θα είναι κοντά στο κρίσιμο σε καθυστέρηση)

# Αλγόριθμοι και μετασχηματισμοί

---

```
REDUCE_DELAY( $G_n(V, E), \epsilon$ ) {  
  repeat {  
    Compute critical paths and critical delay  $\tau$ ;  
    Set output required data-ready times to  $\tau$ ;  
    Compute slacks;  
     $U$  = vertex subset with slack lower than  $\epsilon$ ;  
     $W$  = select vertices in  $U$ ;  
    Apply transformations to vertices  $W$ ;  
  } until (no transformation can reduce  $\tau$ )  
}
```

Η παράμετρος  $\epsilon$  είναι ένα κατώφλι που μας επιτρέπει να λάβουμε υπόψη ένα μεγαλύτερο σύνολο μονοπατιών (από το κρίσιμο).

Με  $\epsilon=0$  θεωρούμε μόνο τα κρίσιμα μονοπάτια (slack=0).



# Αλγόριθμοι και μετασχηματισμοί

---

- ✓ Για την μείωση του χρόνου data-ready μίας κορυφής υπάρχουν δύο πιθανότητες:
  - α) μείωση της καθυστέρησης μετάδοσης της κορυφής και
  - β) μείωση των εξαρτήσεων της κορυφής από κρίσιμες εισόδους
- ✓ Κάθε μετασχηματισμός δεν πρέπει να έχει αποτέλεσμα την δημιουργία ενός άλλου κρίσιμου μονοπατιού με ίση ή μεγαλύτερη καθυστέρηση.
- ✓ Αυτό γίνεται με παρακολούθηση των διακυμάνσεων στους χρόνους των γύρω κορυφών, οι οποίοι πρέπει να περιορίζονται άνω από τα slacks.
- ✓ Ένα σημαντικό θέμα της υπολογιστικής αποδοτικότητας του αλγορίθμου είναι ο επανυπολογισμός των κρίσιμων μονοπατιών και των slacks.
- ✓ Οι περισσότεροι αλγόριθμοι διαφέρουν στην επιλογή του υπο-συνόλου των κορυφών και στην επιλογή των μετασχηματισμών.
- ✓ Το χρησιμοποιούμενο μοντέλο καθυστέρησης μετάδοσης των τοπικών συναρτήσεων είναι ιδιαίτερα σημαντικό.

---

# Βελτιστοποίηση με Κανόνες

---

Τα συστήματα που βασίζονται σε κανόνες είναι μία κατηγορία έμπειρων συστημάτων που χρησιμοποιούν κανόνες για να καθορίσουν τις ενέργειες τους για την επίτευξη ενός στόχου.

Οι μετασχηματισμοί γίνονται βηματικά με τοπικούς μετασχηματισμούς που διατηρούν την λειτουργικότητα. Κάθε υποδίκτυο αντικαθίσταται με ένα άλλο για την επίτευξη του στόχου.

Ένα σύστημα βασισμένο σε κανόνες αποτελείται από:

1. Την βάση δεδομένων με τους κανόνες. Περιέχει ζεύγη διανυσμάτων (αρχικό, τελικό) όπου το αρχικό αναζητείται στο κύκλωμα και όταν βρεθεί αντικαθίσταται με το τελικό.
2. Ένα σύστημα διατήρησης της βάσης.
3. Ευριστικό αλγόριθμο εξαγωγής συμπερασμάτων.

# Βελτιστοποίηση με κανόνες

---



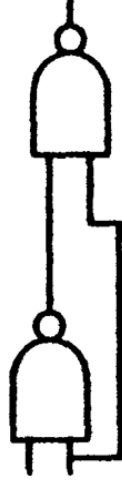
**Πλεονέκτημα:** μπορούν να προστεθούν κανόνες βελτιστοποίησης και μετά την υλοποίηση του αλγορίθμου.

(a)



**Πρόβλημα:** επιλογή κανόνων που θα εφαρμοστούν και κινήσεις με εμπρός βλέψεις και προς τα πίσω (backtracking). Τις καθορίζει ο ευριστικός αλγόριθμος.

(b)



(c)