



ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ



ΕΥΡΩΠΑΪΚΗ ΕΝΔΕΧ  
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ  
Επιχειρησιακό Πρόγραμμα  
Εκπαίδευσης και Αρχικής  
Επαγγελματικής Κατάρτισης

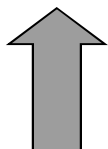
# Σύνθεση Ψηφιακών Συστημάτων

Χ. Καβουσιανός



---

# Μικροηλεκτρονική

---

Αυξανόμενο επίπεδο ολοκλήρωσης ηλεκτρονικών συσκευών  Κατασκευή περίπλοκων συστημάτων (VLSI)

- ✓ Αυξανόμενη πολυπλοκότητα  $\Rightarrow$  καλύτερες διαδικασίες σχεδιασμού.
- ✓ Δυσκολότερη η κατασκευή μη-εσφαλμένων κυκλωμάτων.
- ✓ Τα VLSI κυκλώματα δεν επισκευάζονται  $\Rightarrow$  περιορισμός λαθών

Κατασκευή μεγάλου αριθμού κομματιών  Μείωση Κόστους 

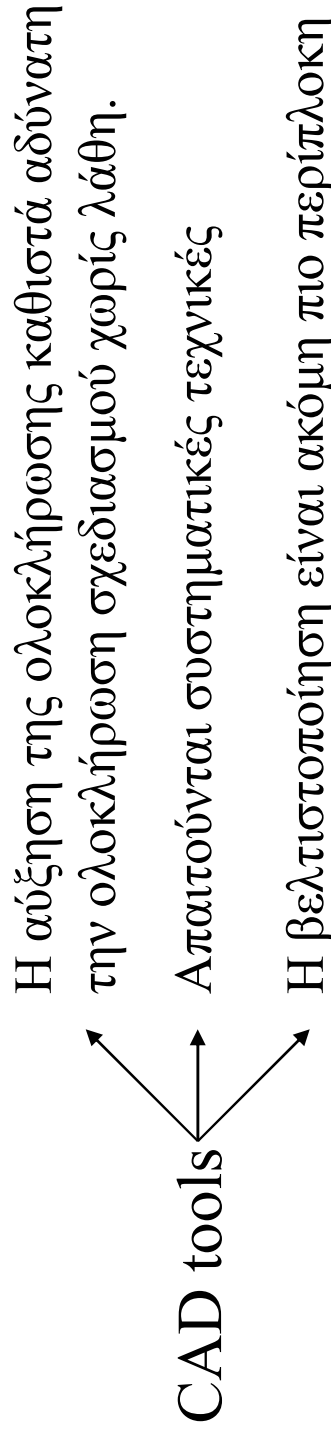
Αντικειμενικός Στόχος Βιομηχανίας

# Μικροηλεκτρονική

---

Τρόποι Μείωσης Κόστους:

- ✓ Η υψηλή ολοκλήρωση οδηγεί σε μείωση των components/system(PCB)  
⇒ Μείωση του κόστους packaging-routing.
- ✓ Μικρός χρόνος και άρα κόστος σχεδιασμού (*μισθοί / time-to-market*).
- ✓ Ποιότητα κατασκευής (*yield*).



---

# Τεχνολογίες Ημιαγωγών

---

Σκοπός Σχεδίασης  $\Rightarrow$  Κατασκευή συστήματος σε ημιαγωγό

Fabrication process  $\swarrow$   
CMOS, BiCMOS...  $\searrow$

Διαμόρφωση υποστρώματος  
Τοπική αλλαγή ιδιοτήτων  
Κατασκευή Επιπέδων διασύνδεσης

Συστήματα  $\left\{ \begin{array}{l} \text{Αναλογικά} \\ \text{Ψηφιακά} \end{array} \right.$

Σύγχρονα με clk (Αρκετά ανεπτυγμένα)  
Ασύγχρονα (Περίπλοκα για VLSI)

## ***Παράγοντες ανταγωνιστικότητας συστημάτων:***

- Απόδοση (ταχύτητα – ρυθμός επεξεργασίας).
- Κατασκευαστικός όγκος (αριθμός πωλήσεων).
- Κόστος πώλησης (καταναλωτικές εφαρμογές).
- Αξιοπιστία (διαστημικές εφαρμογές).

# Σχεδιαστικά Στυλ

---

Full Custom: Η λειτουργική και φυσική περιγραφή γίνονται από τον σχεδιαστή.

Πολύ καλό αποτέλεσμα

Μεγάλη σχεδιαστική προσπάθεια

Μεγάλο κόστος

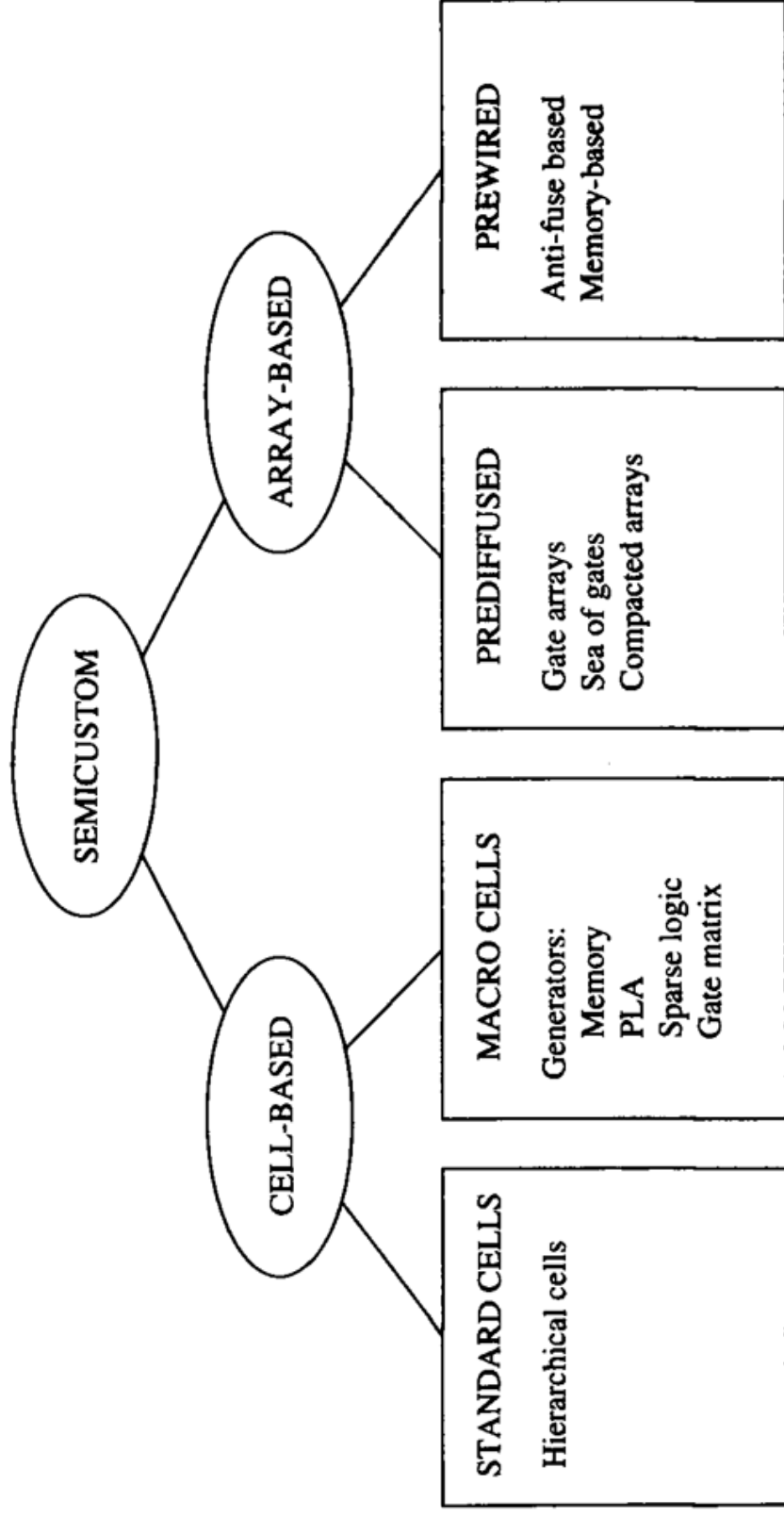
Semicustom: Χρήση και διασύνδεση primitives.



Εφαρμογή των CAD tools

# Σχεδιαστικά Στυλ

---



# Σχεδιαστικά Στυλ

---

Standard Cells: αποθηκεύονται σε βιβλιοθήκη - ανανεώνονται με την εξέλιξη της τεχνολογίας.

Library Binding – Technology Mapping: Έχει αυτοματοποιηθεί.

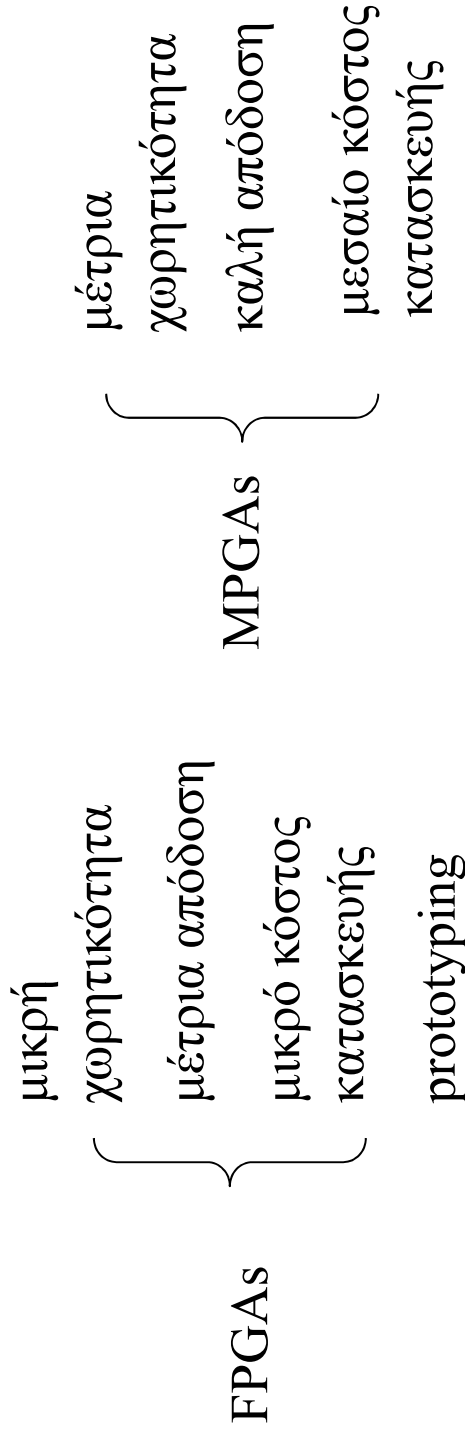
Macro-cell design: Συνεργαζόμενα δομικά blocks που μπορούν να συντεθούν από cell ή module generators. Πχ. μνήμες και PLAs.

---

# Gate Arrays

## *Είδη Gate Arrays:*

- Field Programmable Gate Arrays (FPGAs)
- Mask Programmable Gate Arrays (MPGAs)





---

## Σύγκριση Σχεδιαστικών Στυλ

---

MPGAs και FPGAs μειώνουν τα βήματα κατασκευής  $\Rightarrow$  μειώνουν το κόστος κατασκευής

Βασίζονται σε μερικούς ή ολικώς προκατασκευασμένα components

	<i>Custom</i>	<i>Cell-based</i>	<i>Prediffused</i>	<i>Prewired</i>
Density	Very High	High	High	Medium-Low
Performance	Very High	High	High	Medium-Low
Flexibility	Very High	High	Medium	Low
Design time	Very Long	Short	Short	Very Short
Manufacturing time	Medium	Medium	Short	Very Short
Cost - low volume	Very High	High	High	Low
Cost - high volume	Low	Low	Low	High

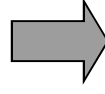
# Αυτοματοποίηση Σχεδίασης

---

## *Πλεονεκτήματα Αυτοματοποίησης Σχεδίασης:*

- Εγγύαται πολύ μικρό κύκλο σχεδίασης
  - Διερεύνηση διαφορετικών σχεδιαστικών στυλ (οι σχεδιασμοί παράγονται και αξιολογούνται γρήγορα).
  - Το σχεδιαστικό αποτέλεσμα είναι καλύτερο από τον μέσο ανθρώπινο σχεδιασμό.
- 

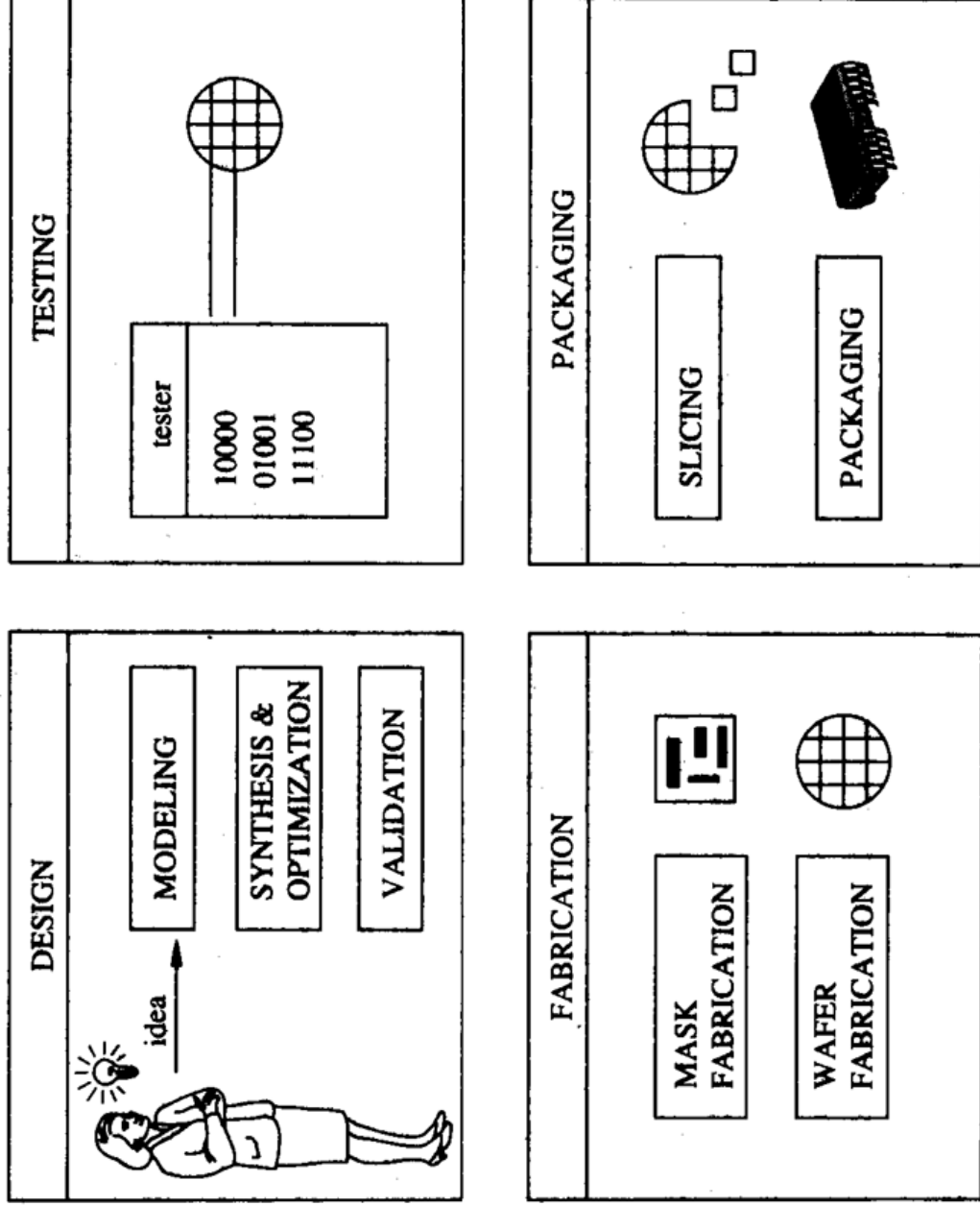
Όσο μεγαλύτερο είναι το προς επίλυση πρόβλημα, τόσο μεγαλύτερος είναι ο χώρος αναζήτησης λύσης και αυξάνει η πολυπλοκότητα.



Δίνουν καλά αποτελέσματα σε συγκεκριμένα προβλήματα αλλά δεν μπορούν να ανταγωνιστούν τον άνθρωπο στον πλήρη κύκλο σχεδιασμού

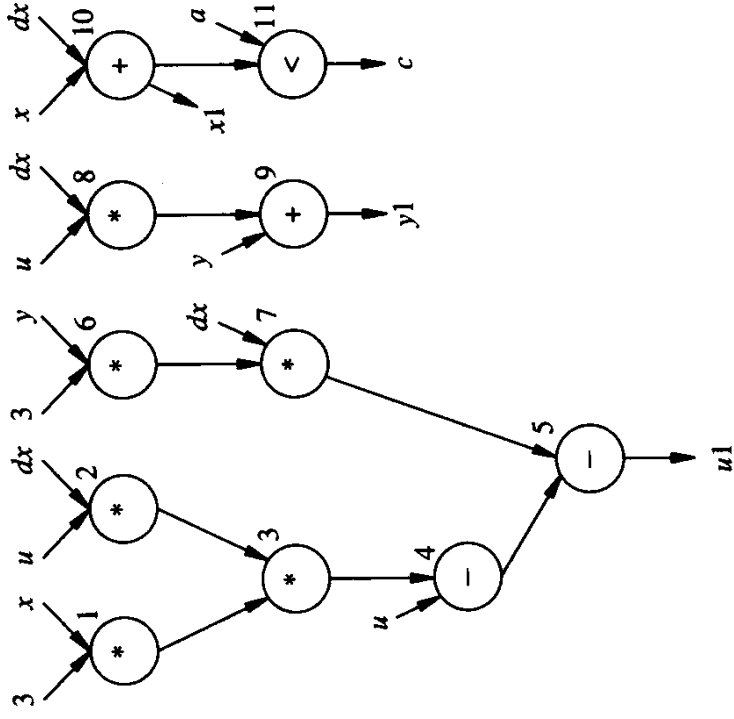
---

# Σχεδιασμός Κυκλωμάτων

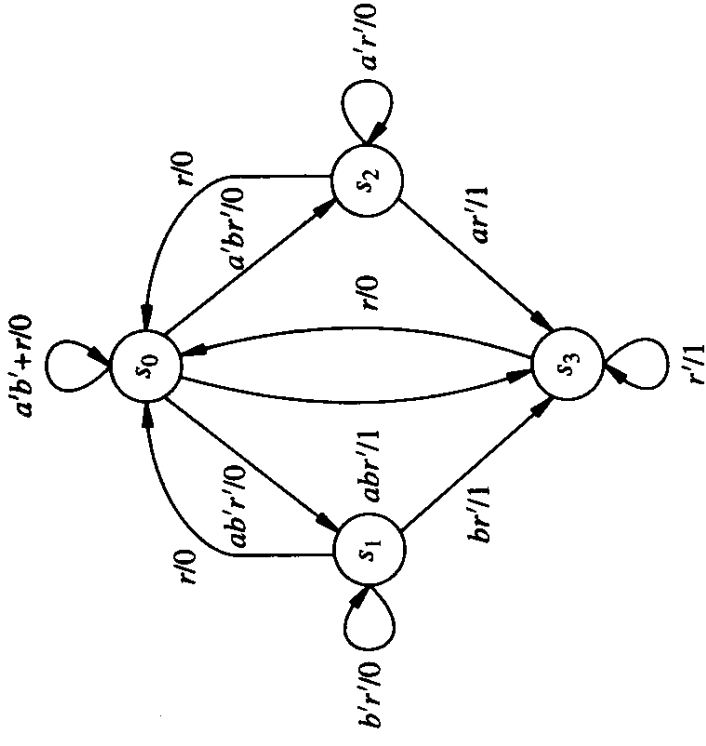


# Modeling

Modeling: Διάφοροι τρόποι περιγραφής. Πχ. μοντελοποίηση με HDLs, flow diagrams, schematic diagram, state diagrams.

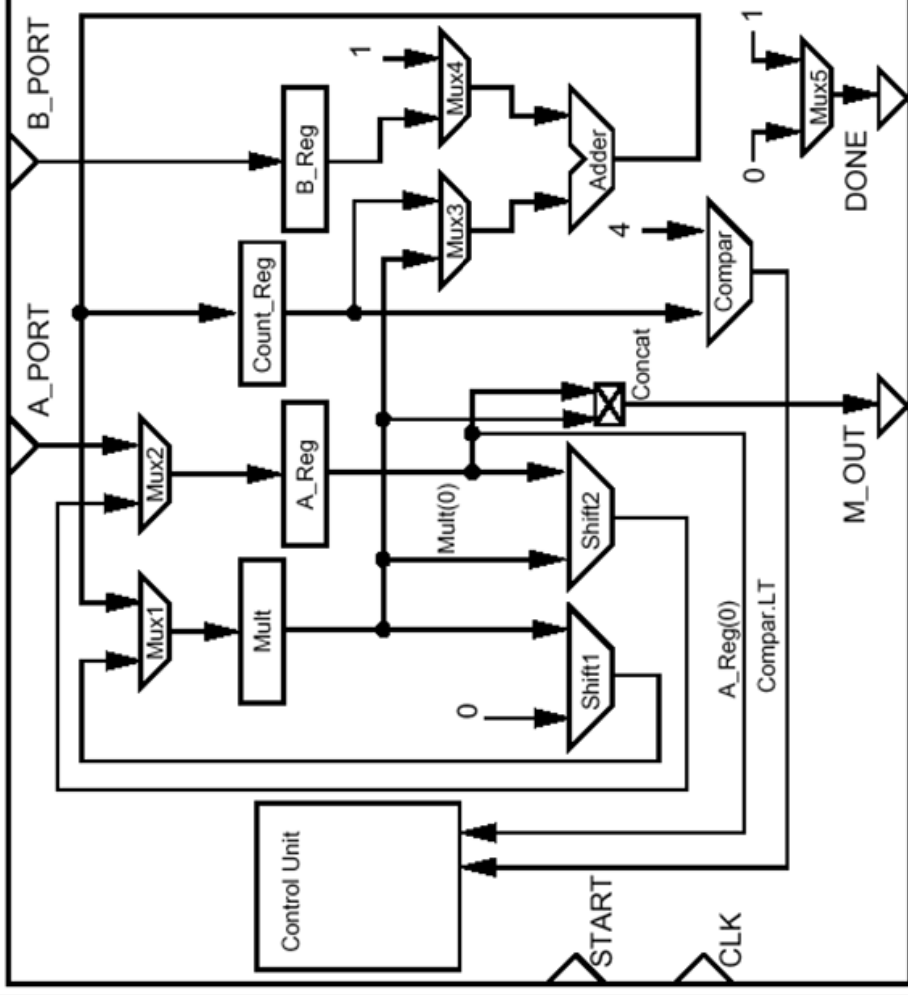


Flow Diagram



State Diagram

# Modeling



Schematic Diagram

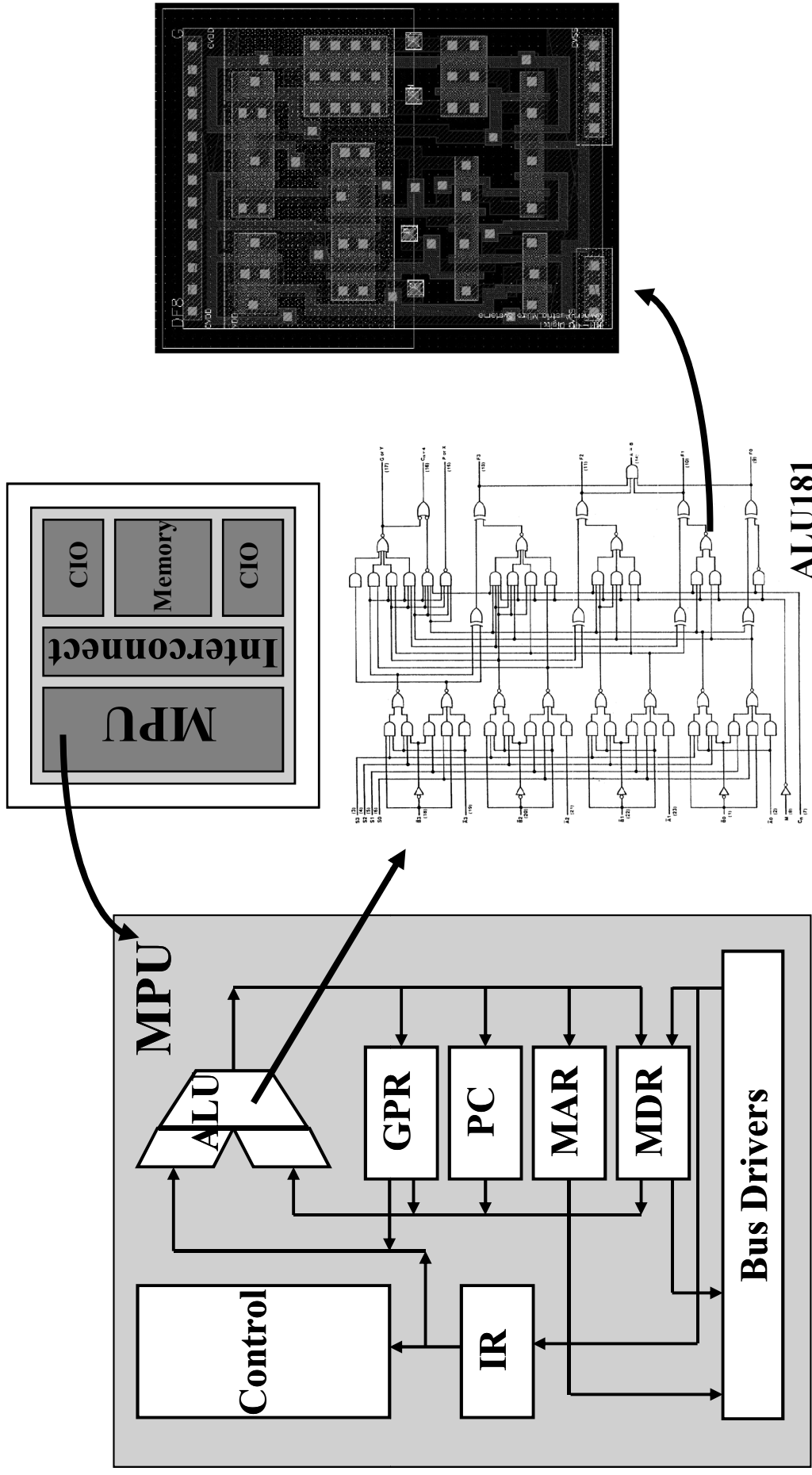
# Σύνθεση

---

Σύνθεση: Σταδιακή εκλέπτυνση του αφηρημένου μοντέλου σε συγκεκριμένη υλοποίηση χρησιμοποιώντας επιπλέον πληροφορίες για την τεχνολογία και το σχεδιαστικό στυλ.

Στόχος: η κατασκευή γεωμετρικού μοντέλου.

# Παράδειγμα Σύνθεσης



# Βελτιστοποίηση

---

Βελτιστοποίηση (ποιότητας):

- **Απόδοση κυκλώματος:** α) χρόνος επεξεργασίας πληροφοριών και β) ποσό πληροφοριών σε δεδομένο χρόνο.
- **Επιφάνεια:** μείωση επιφάνειας (περισσότερα ολοκλ./wafer, μικρότερο κόστος ολοκληρωμένου, μικρότερο κόστος package)
- **Ελεγχιμότητα:** δυνατότητα ελέγχου μετά την κατασκευή (fault coverage).



# Επιβεβαίωση

---

## Επιβεβαίωση (Validation)

Επιβεβαίωση ότι το κύκλωμα θα λειτουργήσει όπως σχεδιάστηκε (χωρίς σχεδιαστικά λάθη).

- ✓ Εάν το λάθος ανακαλυφθεί μετά την κατασκευή το κόστος είναι μεγάλο.
- ✓ Έλεγχος λειτουργίας του κυκλώματος με βάση τον σχεδιασμό.
- ✓ Γίνεται με εξομείωση (σε διάφορα επίπεδα) και verification.

Formal Verification: σύγκριση δύο μοντέλων περιγραφής και επιβεβαίωση της ομοιότητάς τους

# Σύνθεση & Βελτιστοποίηση

---

*Τα CAD εργαλεία δίνουν δυνατότητα σχεδιασμού βιώσιμων οικονομικά προϊόντων.*

Σύνθεση: μείωση ανθρώπινης προσπάθειας

Βελτιστοποίηση: βελτίωση ποιότητας

Μοντέλα: Διαφορετικά επίπεδα αφάιρεσης περιγραφής ενός συστήματος (αρχιτεκτονικό, λογικό, γεωμετρικό).

Σύνθεση είναι η παραγωγή ενός μοντέλου από ένα άλλο, λιγότερο λεπτομερές

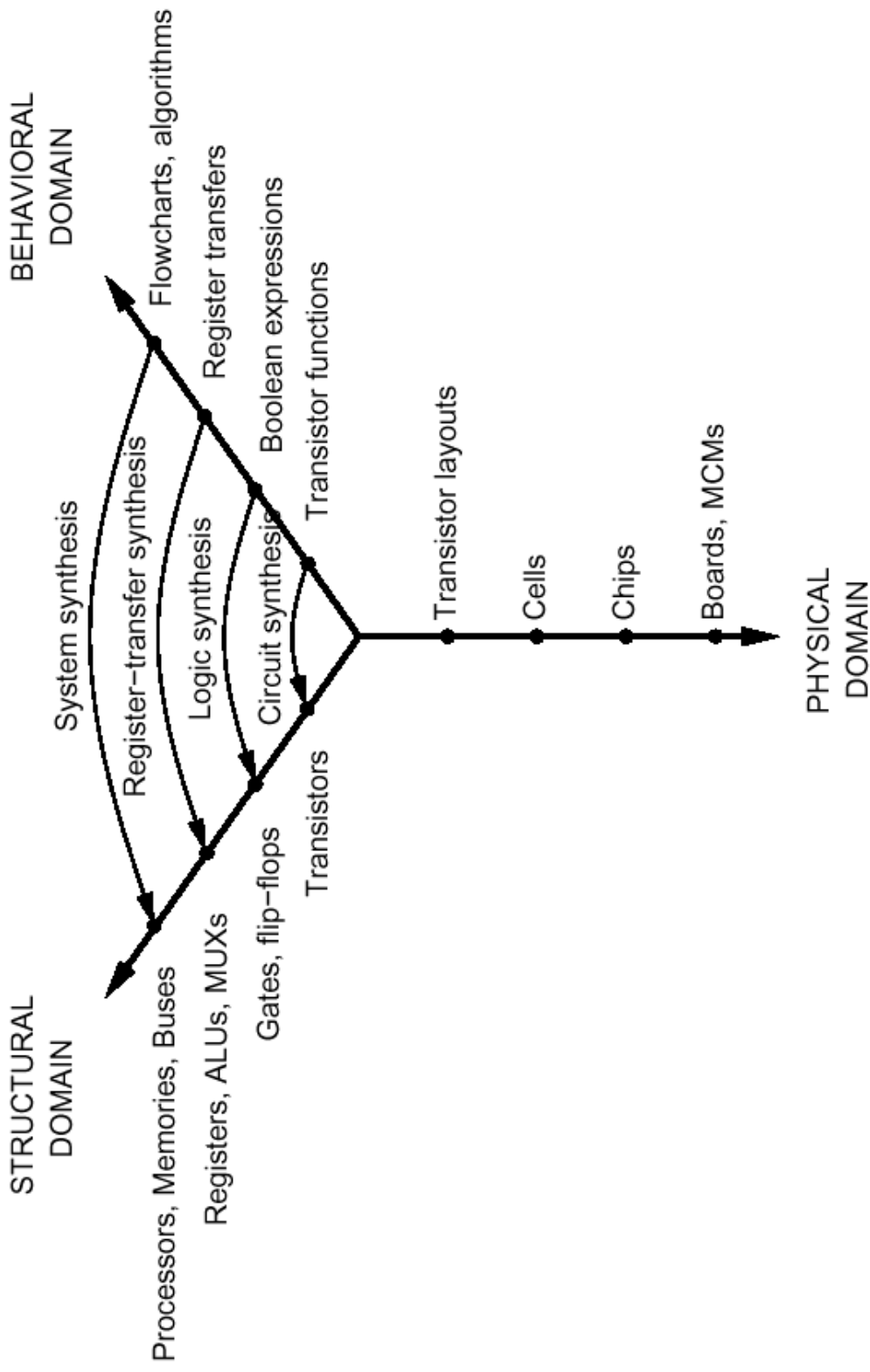
Συμπεριφοράς (behavioral)

Επίπεδα περιγραφής { Κατασκευής (structural)

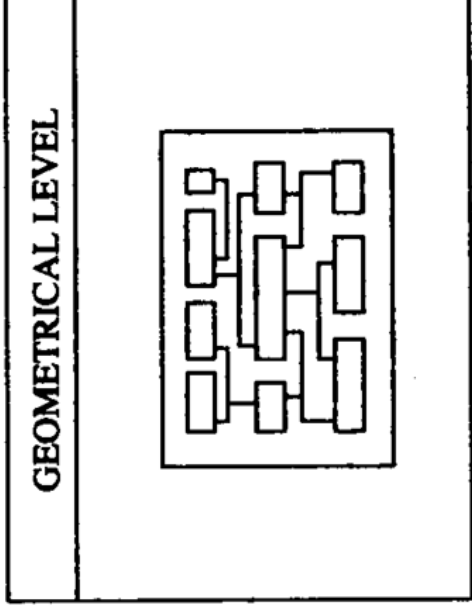
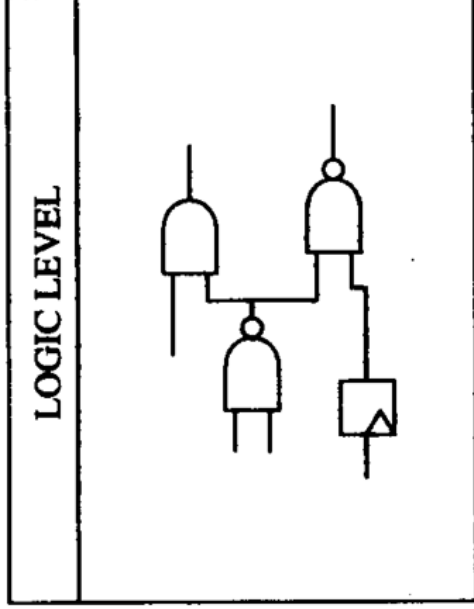
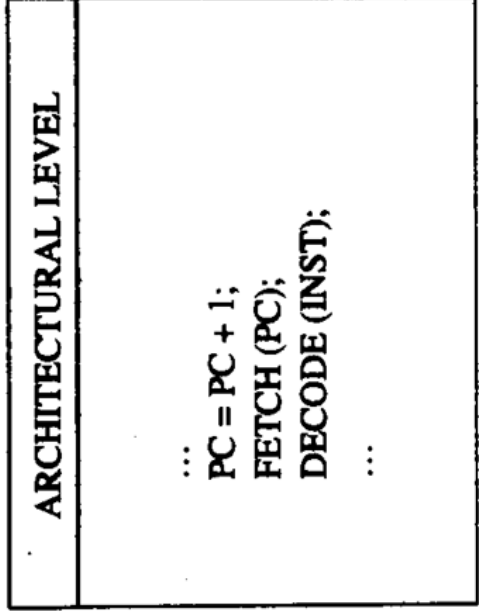
Φυσικό (Physical)

*Μοντέλα σε διαφορετικά επίπεδα αφάιρεσης μπορούν να περιγραφούν με διαφορετικά επίπεδα περιγραφής.*

# Επίπεδα αφάιρεσης



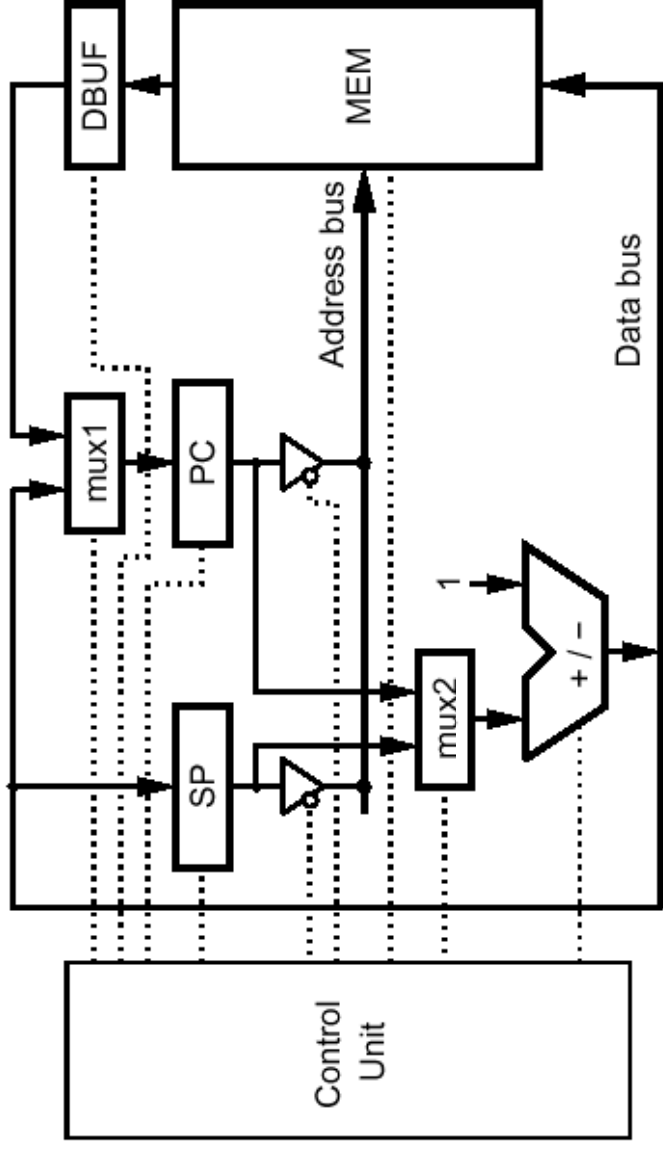
# Επίπεδα αφάιρεσης



# Επίπεδα αφάιρεσης

```
if IR(3) = '0' then
  PC := PC + 1;
else
  DBUF := MEM(PC);
  MEM(SP) := PC + 1;
  SP := SP - 1;
  PC := DBUF;
end if;
```

## BEHAVIOR

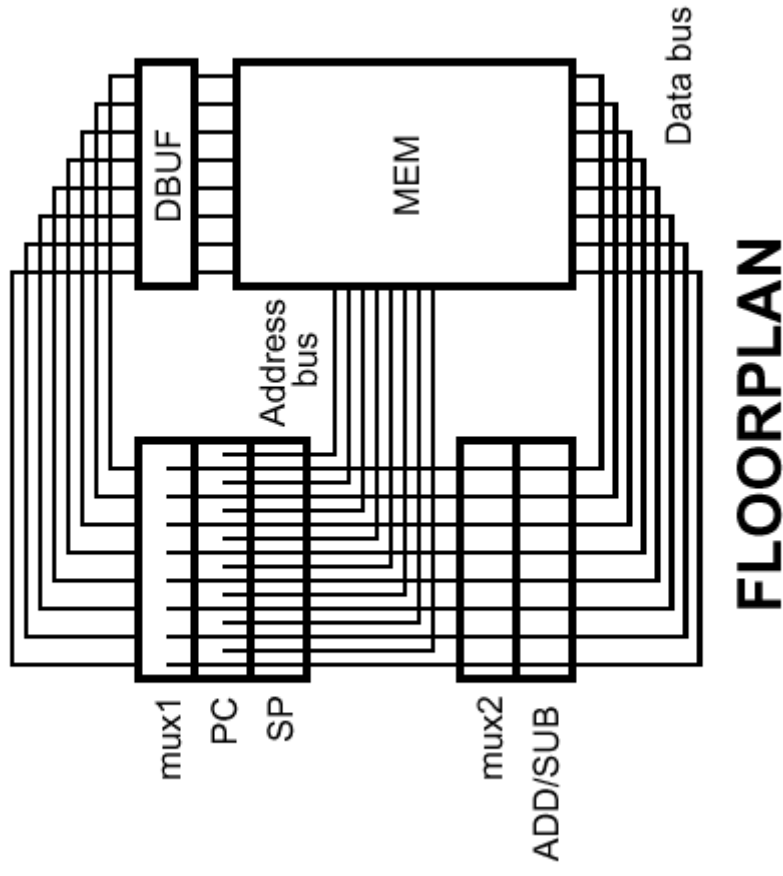


## STRUCTURE

Οι λειτουργίες (πχ +, -) δεν αντιστοιχούν μία προς μία στα λειτουργικά blocks (αθροιστές). Η διαδικασία σύνθεσης καθορίζει διαμοιραζόμενες μονάδες

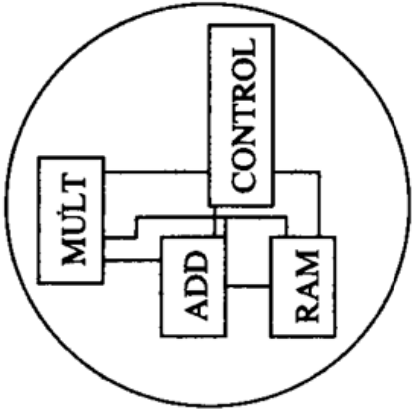
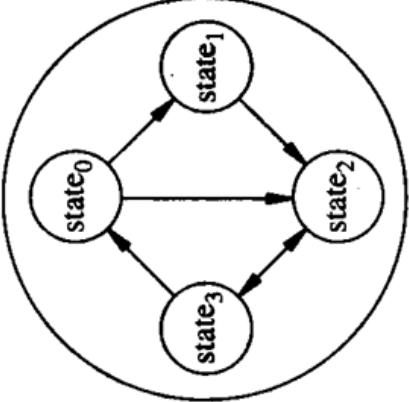
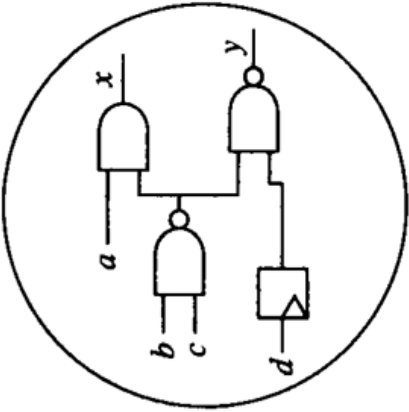
# Επίπεδα αφαίρεσης

---



Σε κάθε επίπεδο προστίθενται πληροφορίες  
σε σχέση με τα προηγούμενα  Αφαίρεση

# Επίπεδα αφαίρεσης / περιγραφής

BEHAVIORAL VIEW	STRUCTURAL VIEW	VIEWS LEVELS
<p>...            PC = PC + 1;            FETCH (PC);            DECODE (INST);            ...</p>		<p>ARCHITECTURAL LEVEL</p>
		<p>LOGIC LEVEL</p>

# Σύνθεση

---

*Σύνθεση είναι η μετάφραση μιας περιγραφής συμπεριφοράς σε περιγραφή κατασκευής.*

Architecture-level synthesis: Δημιουργία περιγραφής κατασκευής (structural) από μοντέλο αρχιτεκτονικής:

ανάθεση λειτουργιών σε τελεστές (resources),  
διασύνδεση  
χρονισμός εκτέλεσης

Logic-level synthesis: Δημιουργία περιγραφής κατασκευής από λογικό μοντέλο:

διαχείριση λογικών εξισώσεων  
δημιουργία δικτύων από logic-primitives  
technology mapping.



---

# Σύνθεση

---

Geometrical-level synthesis: Δημιουργία φυσικής περιγραφής (layout).  
placement & routing  
Δημιουργία μασκών

---

Περιγραφή μεταφοράς καταχωρητή (Register Transfer)

- Σύνολο καταστάσεων
  - Μεταφορά περιεχομένων καταχωρητών σε κάθε κατάσταση
- Σύνθεση
- Datapath
- Control Unit
- 
- The diagram illustrates the synthesis process. It starts with a description of Register Transfer (RT) at the top. Two arrows point downwards from this description to two main components: 'Σύνθεση' (Synthesis) and 'Control Unit'. Below 'Σύνθεση', there is a list of two items: 'Σύνολο καταστάσεων' (Set of states) and 'Μεταφορά περιεχομένων καταχωρητών σε κάθε κατάσταση' (Transfer of register contents in each state). A large grey arrow points from this list up to the 'Σύνθεση' label. To the right of 'Σύνθεση', there is a bracketed group containing 'Datapath' and 'Control Unit'.

**Netlist:** Λίστα αντικειμένων με τις διασυνδέσεις τους

# Σύνθεση σε επίπεδο Αρχιτεκτονικής

---

Στόχοι της διαδικασίας:

- ✓ Επιλογή hardware resources.
  - ✓ Scheduling λειτουργιών (χρόνος).
  - ✓ Δέσμευση λειτουργιών σε resources (binding).
- 
- ✓ Δημιουργία data path
  - ✓ Δημιουργία control unit

*Σε αυτό το στάδιο καθορίζεται σε μεγάλο βαθμό η επιφάνεια του τελικού κωκλώματος όπως και η απόδοσή του.*

*Η σύνθεση αρχιτεκτονικής καθορίζει τον βαθμό παραλληλισμού των λειτουργιών.*

## Σύνθεση σε επίπεδο Αρχιτεκτονικής

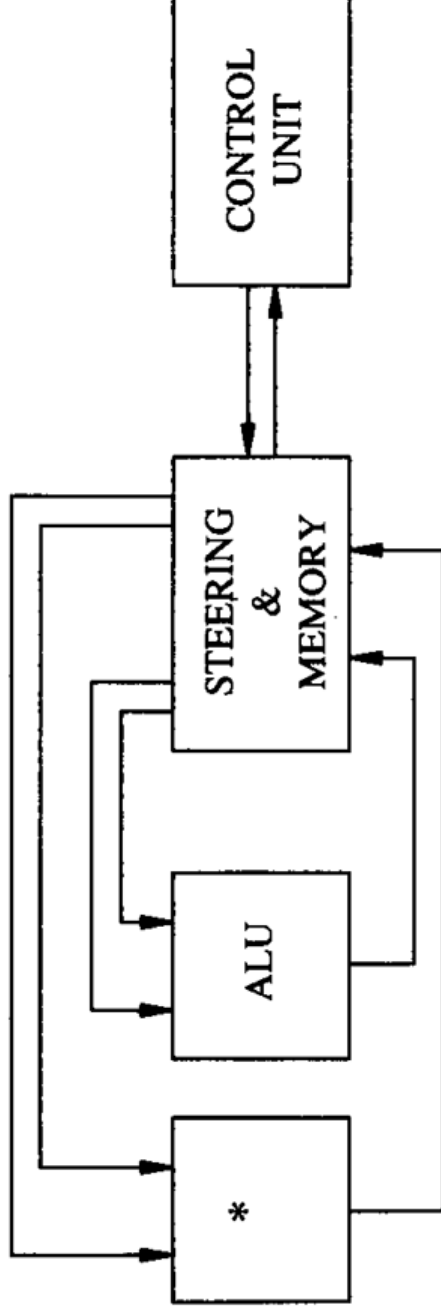
---

```
diffeq {
  read ( x, y, u, dx, a );
  repeat {
    x1 = x + dx;
    u1 = u - ( 3 * x * u * dx ) - ( 3 * y * dx );
    y1 = y + u * dx;
    c  = x1 < a;
    x = x1 ; u = u1 ; y = y1;
  }
  until ( c );
  write ( y );
}
```

Λύση της εξίσωσης  $y'' + 3xy' + 3y = 0$  στο διάστημα  $[0, a]$  με βήμα  $dx$  και αρχικές τιμές  $x(0)=x; y(0)=y, y'(0)=u$ .

# Σύνθεση σε επίπεδο Αρχιτεκτονικής

---



Λύση της εξίσωσης  $y'' + 3xy' + 3y = 0$  στο διάστημα  $[0, a]$  με βήμα  $dx$  και αρχικές τιμές  $x(0) = x$ ;  $y(0) = y$ ,  $y'(0) = u$ .

---

# Λογική Σύνθεση

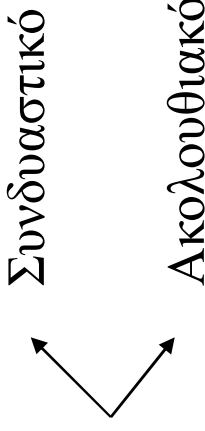
---

Λογικό μοντέλο κυκλώματος:

- ✓ State machine.
- ✓ Σχηματικό διάγραμμα.
- ✓ Μοντέλο HDL.

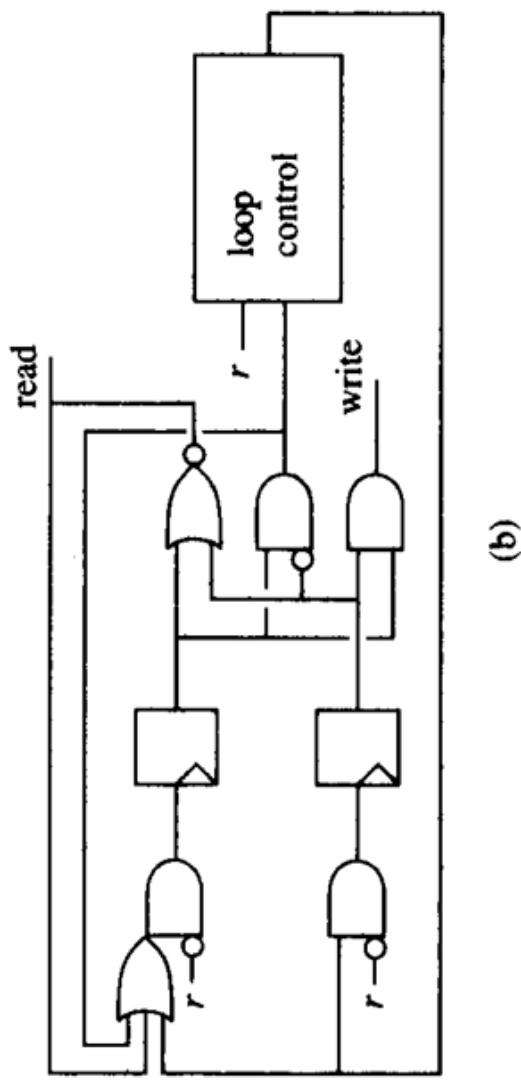
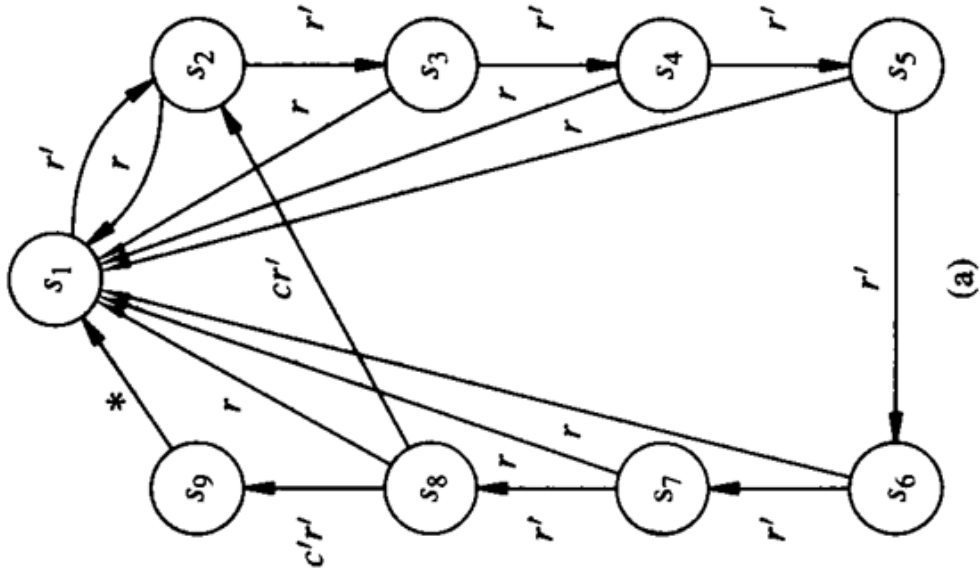
- ✓ Καθορίζεται από τον σχεδιαστή ή προκύπτει από σύνθεση
  - ✓ Το αποτέλεσμα είναι gate-level netlist.
- 

Η σύνθεση εξαρτάται από το είδος του κυκλώματος



*Σε αυτό το στάδιο μεγάλη επίδραση στην επιφάνεια του τελικού κυκλώματος όπως και στην απόδοσή του έχει η διαδικασία της βελτιστοποίησης.*

# Λογική Σύνθεση



# Γεωμετρική Σύνθεση

---

*Είναι το τελικό βήμα στην διαδικασία σχεδιασμού*

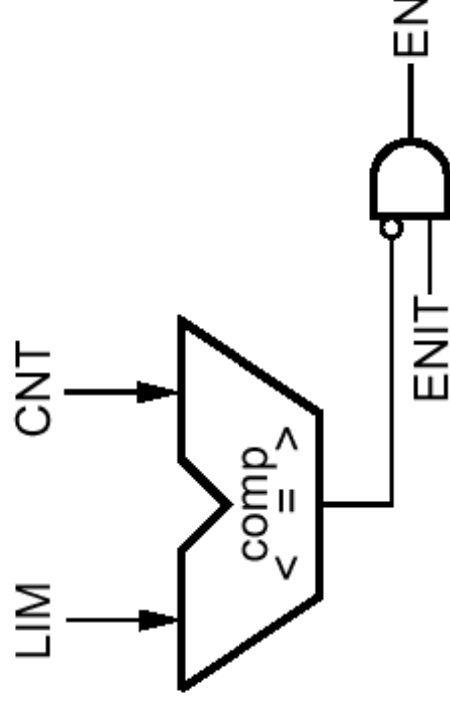
- ✓ Κατασκευή του layout του chip.
- ✓ Placement και routing.
- ✓ Εξαρτάται από το σχεδιαστικό στυλ και την διαδικασία κατασκευής (layers ...).
- ✓ Στον σχεδιασμό custom design η σχεδίαση γίνεται με layout editors.
- ✓ Η χρήση αυτόματων εργαλείων βοηθάει πολύ στην βελτιστοποίηση σε αυτό το επίπεδο.

# Σύνθεση & Περιγραφή

---

*Ο τρόπος περιγραφής επηρεάζει πολύ το αποτέλεσμα της σύνθεσης*

```
if CNT ≠ LIM then
  EN <= ENIT;
else
  EN <= '0';
end if;
```



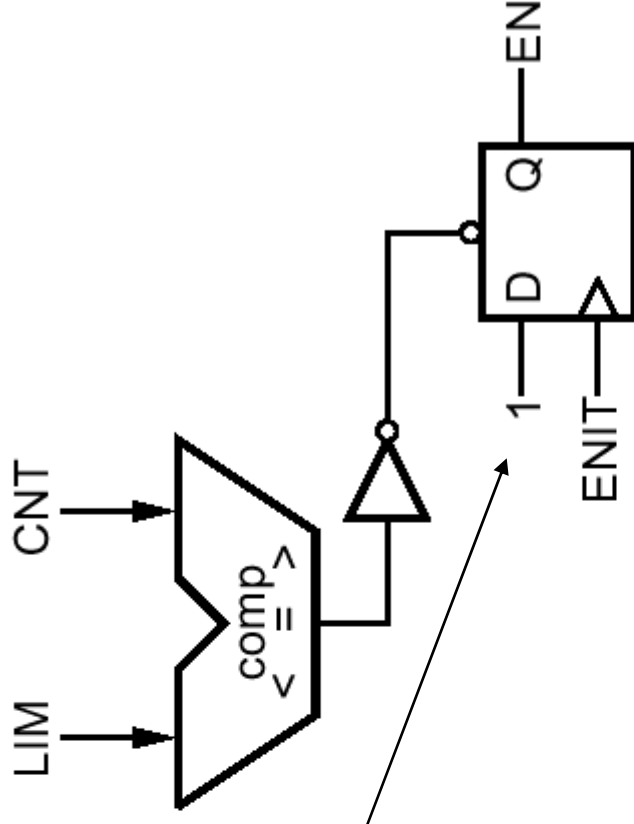
**Level sensitive**



# Σύνθεση & Περιγραφή

---

```
if ENIT = '1' and not ENIT' stable then  
  EN <= '1';  
elseif CNT = LIM then  
  EN <= '0';  
end if;
```

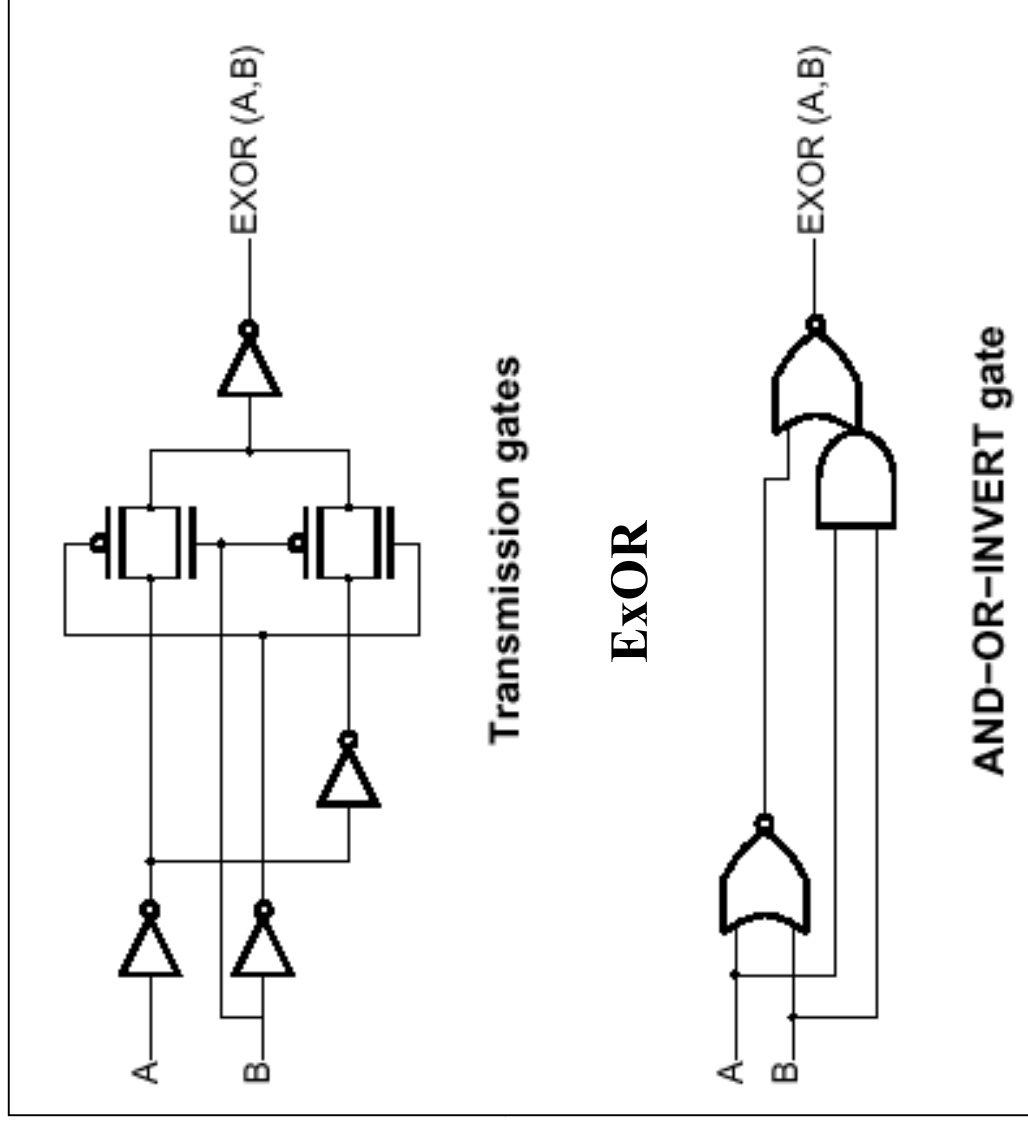


## Edge sensitive

*Μία “κακή” περιγραφή μπορεί να οδηγήσει στην χρήση περιττού κωκλώματος*

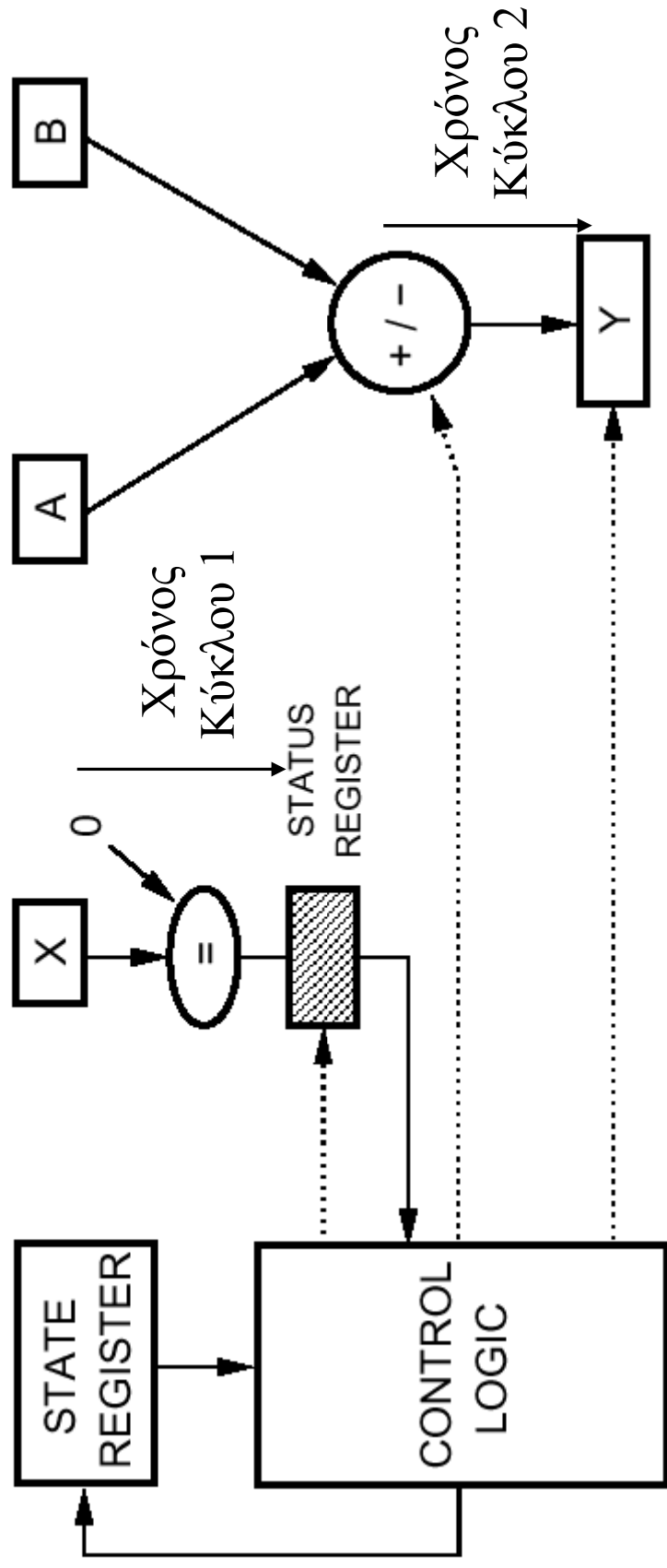
# Σύνθεση & Περιγραφή

*Για μία συνάρτηση  
υπάρχουν πολλές  
υλοποιήσεις για  
διαφορετικές απαιτήσεις  
και περιορισμούς  
(επιφάνειας, ταχύτητας,  
κατανάλωσης, ...)*



# Σύνθεση & Περιγραφή

if  $x=0$  then  $y=a+b$  else  $y=a-b$

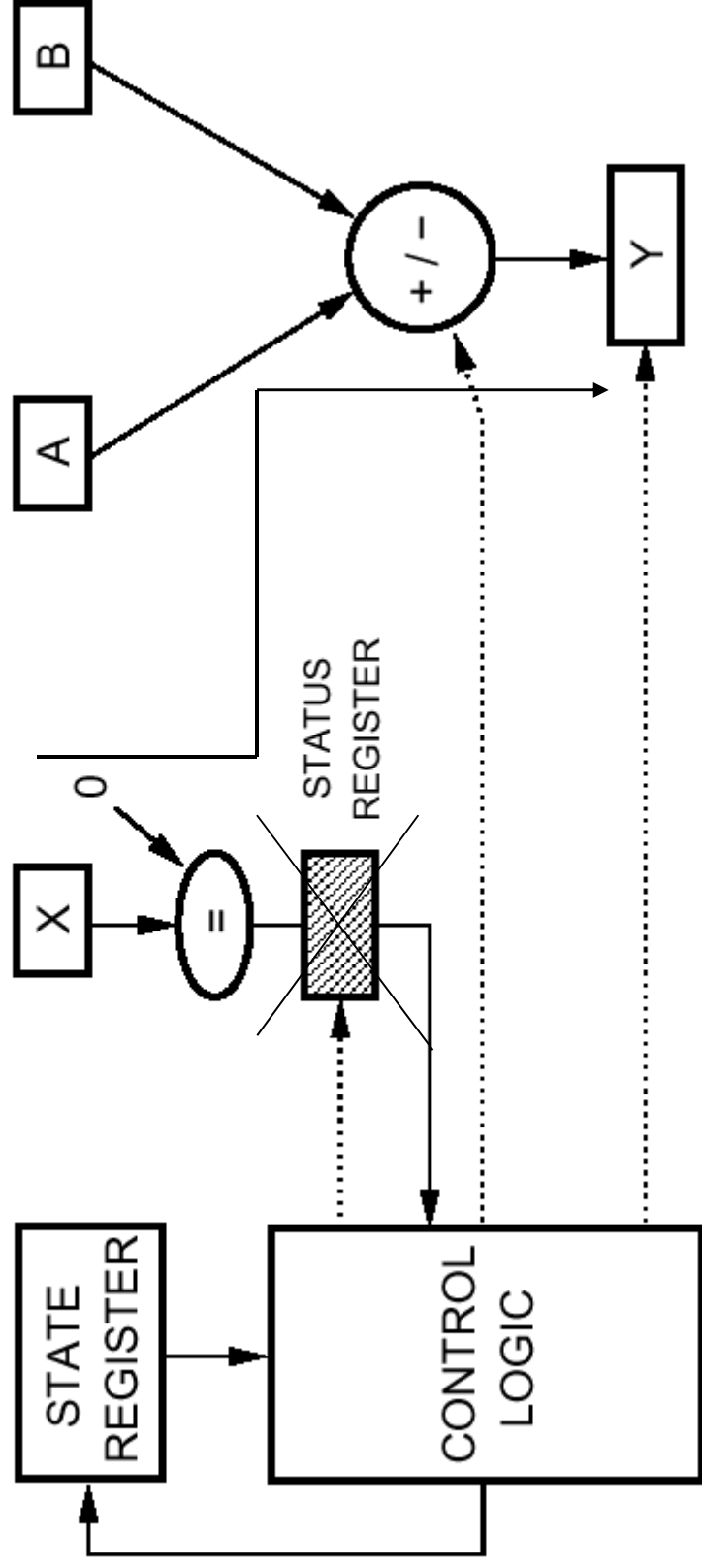


Με καταχωρητή status  $\rightarrow$  2 κύκλοι – κύκλος συντομότερος

# Σύνθεση & Περιγραφή

---

if  $x=0$  then  $y=a+b$  else  $y=a-b$



Χωρίς καταχωρητή status  $\rightarrow$  1 κύκλος μεγαλύτερος

---

# Βελτιστοποίηση

---

**Επιφάνεια:** Αριθμός βασικών δομικών μονάδων με γνωστή επιφάνεια (Πύλες, καταχωρητές) και η διασύνδεση (routing).

## **Απόδοση:**

➤ Συνδυαστικά κυκλώματα: καθυστέρηση διάδοσης εισόδου/εξόδου από το κρίσιμο μονοπάτι.

➤ Ακολουθιακά κυκλώματα:

Μικρότερη δυνατή περίοδος ρολογιού.

Latency (αριθμός κύκλων για ολοκλήρωση).

Throughput (για pipeline).

*Βελτιστοποίηση κυκλώματος είναι η ελαχιστοποίηση της επιφάνειας και η μεγιστοποίηση της απόδοσης.*

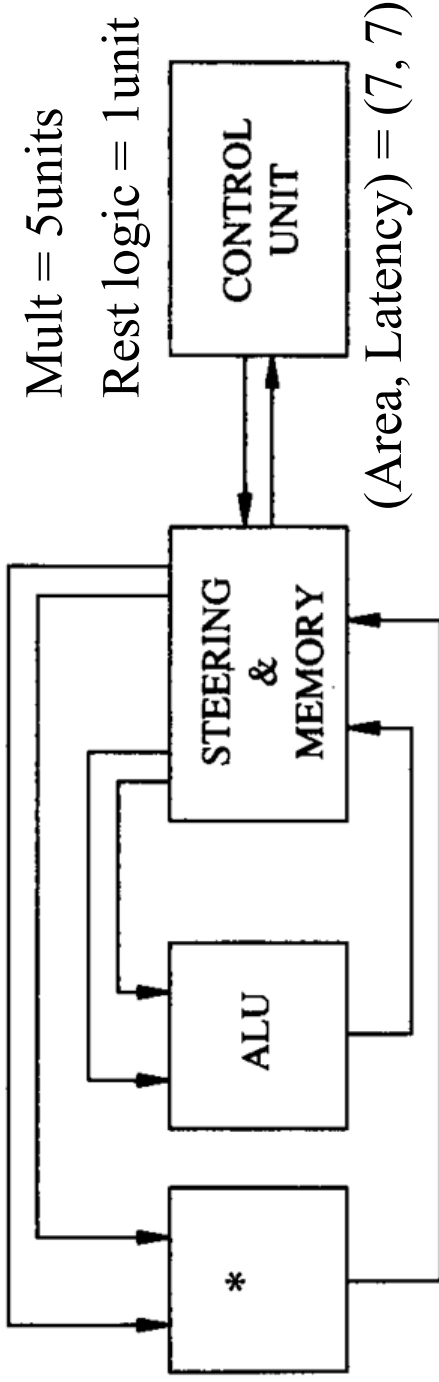
*Με καθορισμό περιορισμών μπορεί να σταθεροποιηθεί ένας παράγοντας και να βελτιωθεί ο άλλος.*

# Βελτιστοποίηση

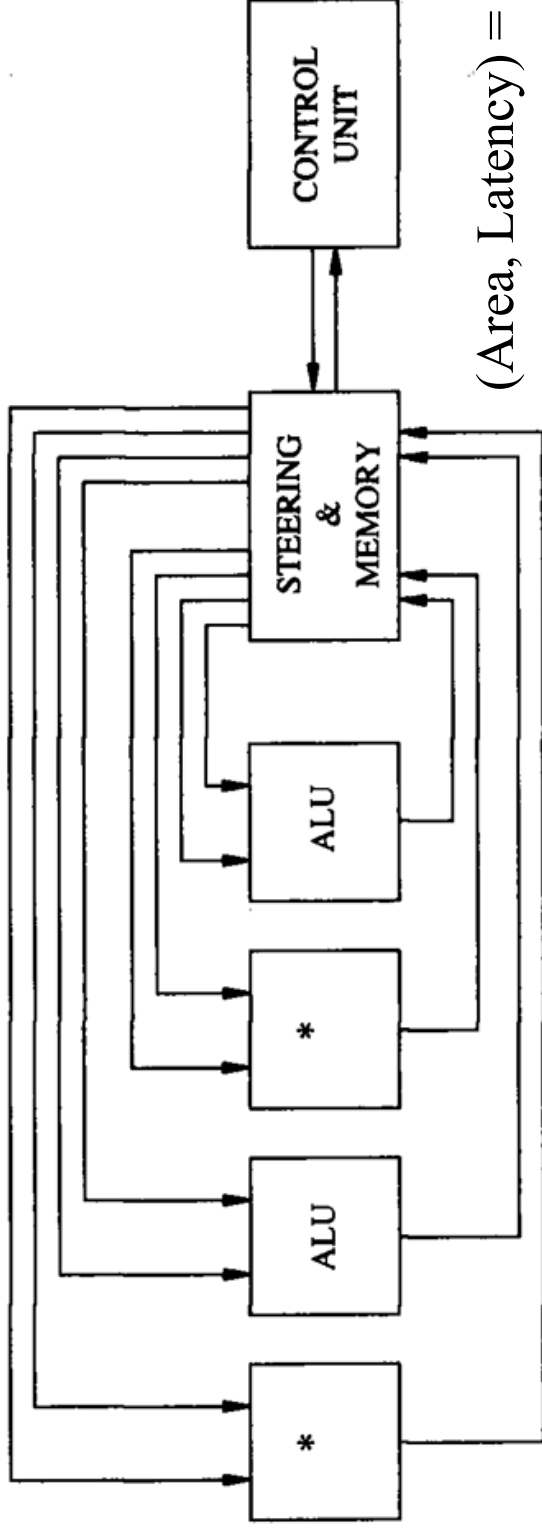
ALU = 1 unit

Mult = 5 units

Rest logic = 1 unit



(Area, Latency) = (7, 7)



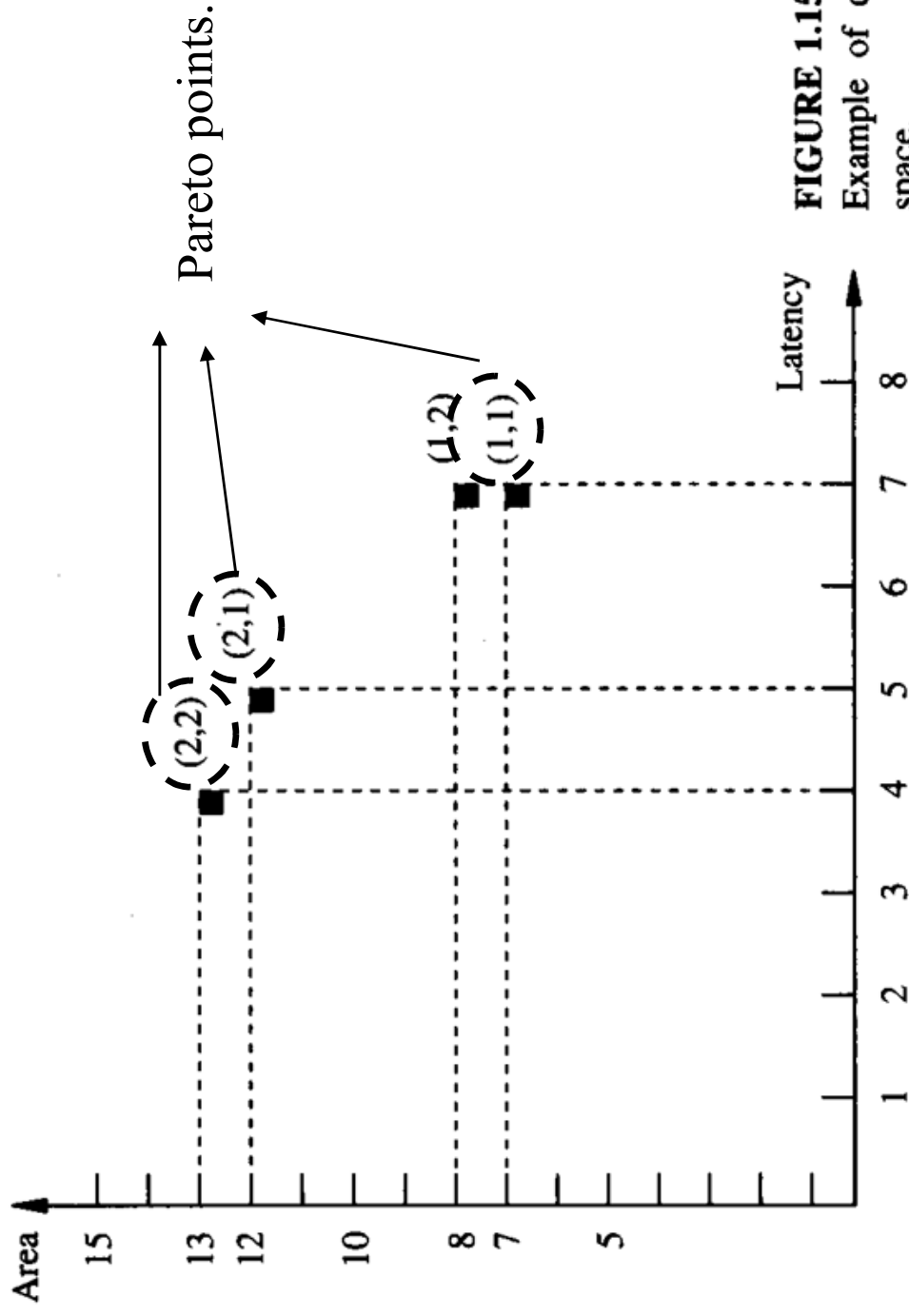
(Area, Latency) = (13, 4)

---

# Βελτιστοποίηση

- ✓ Η βελτιστοποίηση υπόκειται σε περιορισμούς σχεδίασης (constraints).
- ✓ Χώρος εκτίμησης σχεδιασμού (design evaluation space): είναι ένας πολυδιάστατος χώρος με διαστάσεις τις παραμέτρους βελτιστοποίησης.
- ✓ Κάθε πιθανός σχεδιασμός (λύση) είναι ένα σημείο σε αυτόν τον χώρο.
- ✓ **Pareto point**: σημείο για το οποίο δεν υπάρχει άλλο με μία τουλάχιστον παράμετρο καλύτερη και τις υπόλοιπες ίδιες ή καλύτερες.
- ✓ Όλα τα Pareto points καθορίζουν την καμπύλη trade-off του σχεδιασμού.

# Βελτιστοποίηση



**FIGURE 1.15**  
Example of design evaluation space.

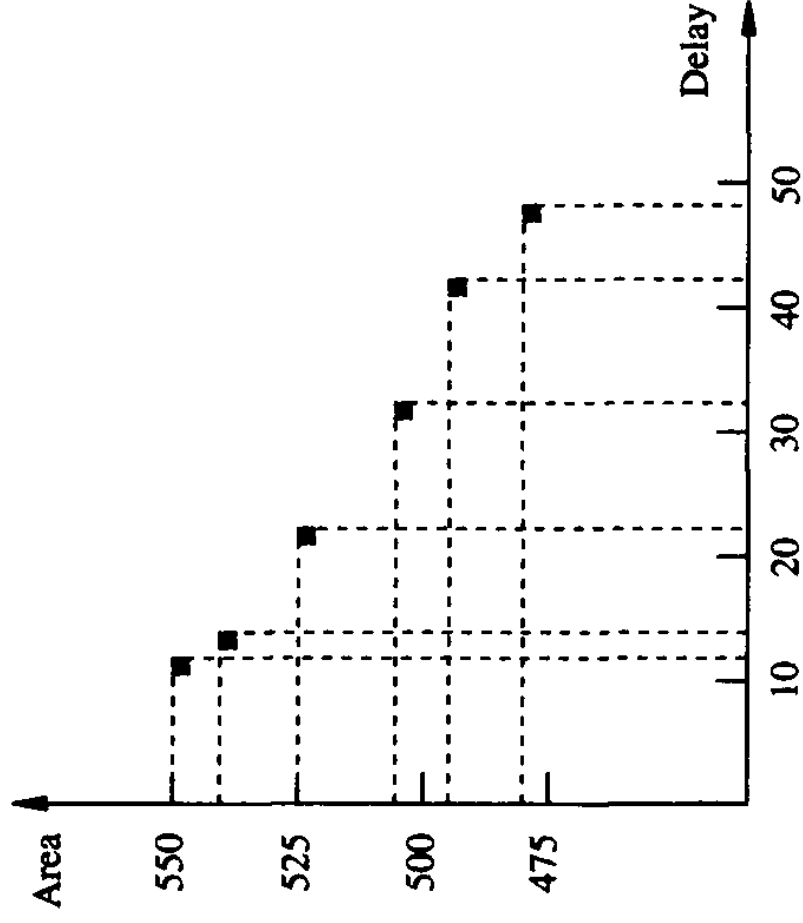
Πολλαπλασιαστής = 5 units,  $A_{lu} = 1$  unit, υπόλοιπο κύκλωμα = 1 unit.



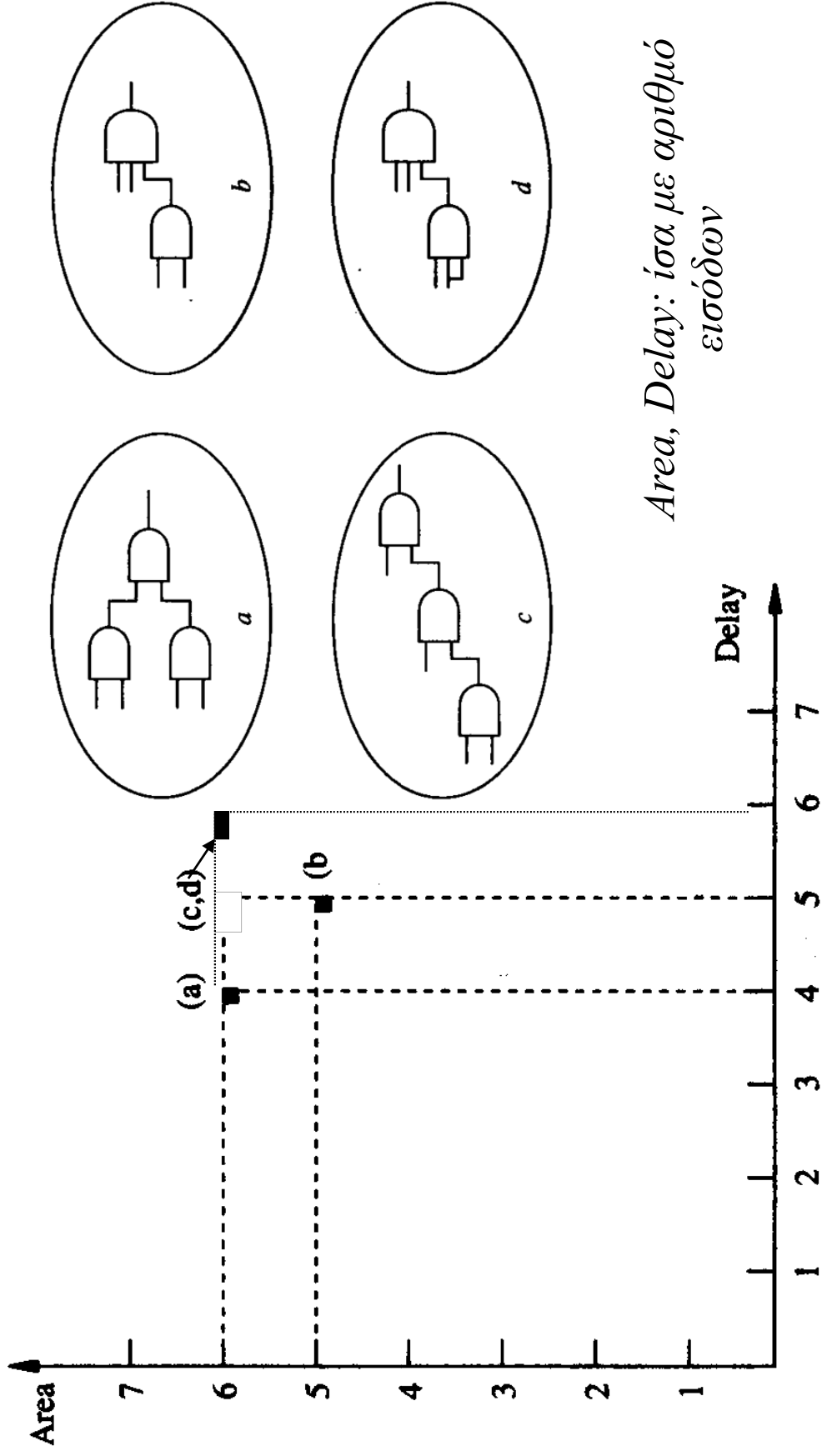
---

# Συνδυαστικά Λογικά Κυκλώματα

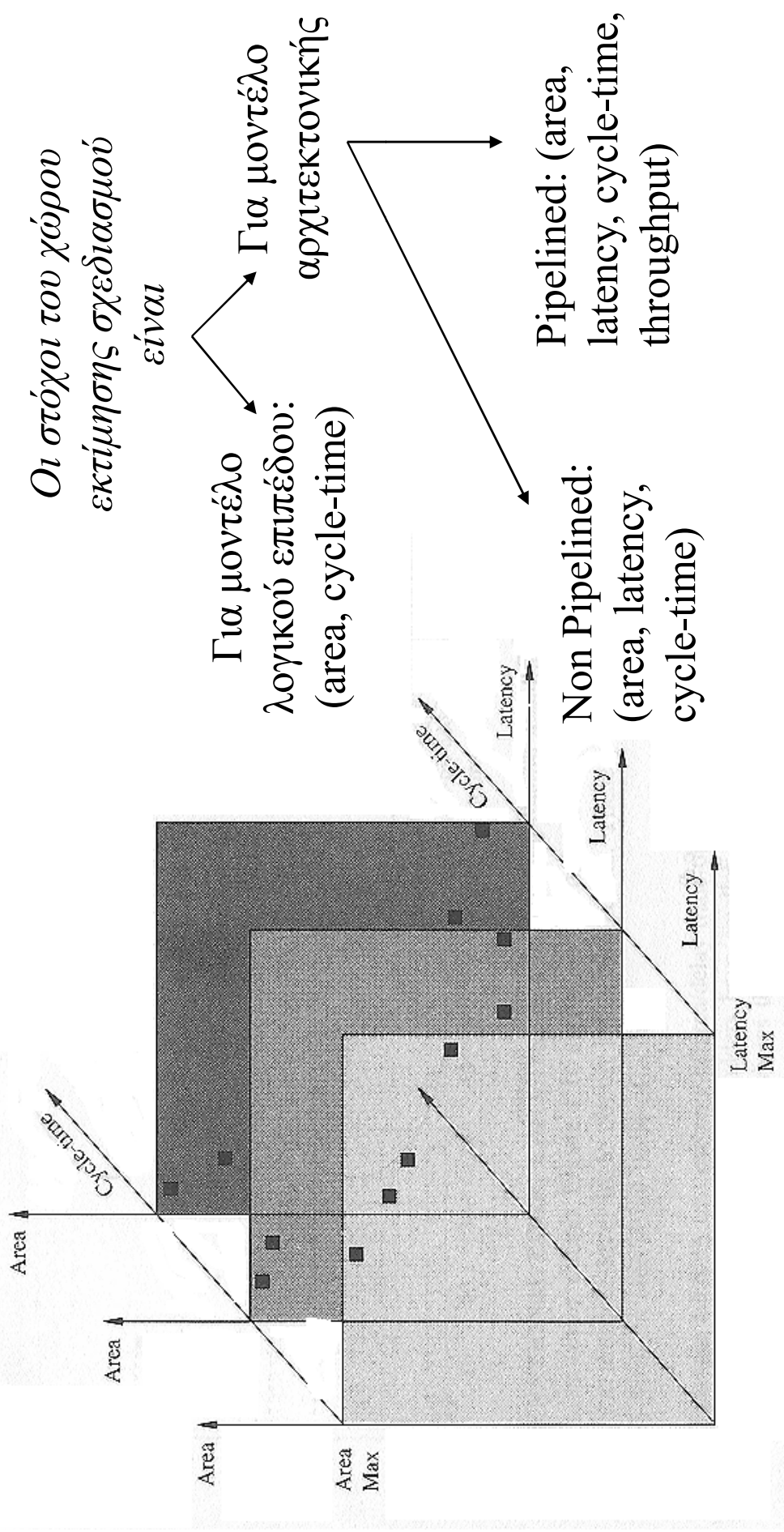
*Οι στόχοι του χώρου εκτίμησης σχεδιασμού είναι η επιφάνεια και η καθυστέρηση.*



# Συνδυαστικά Λογικά Κυκλώματα



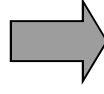
# Ακολουθιακά Λογικά Κυκλώματα



# Προσεγγίσεις Βελτιστοποίησης

---

Η βελτιστοποίηση είναι δύσκολο πρόβλημα (εμπλέκονται πολλές παράμετροι).



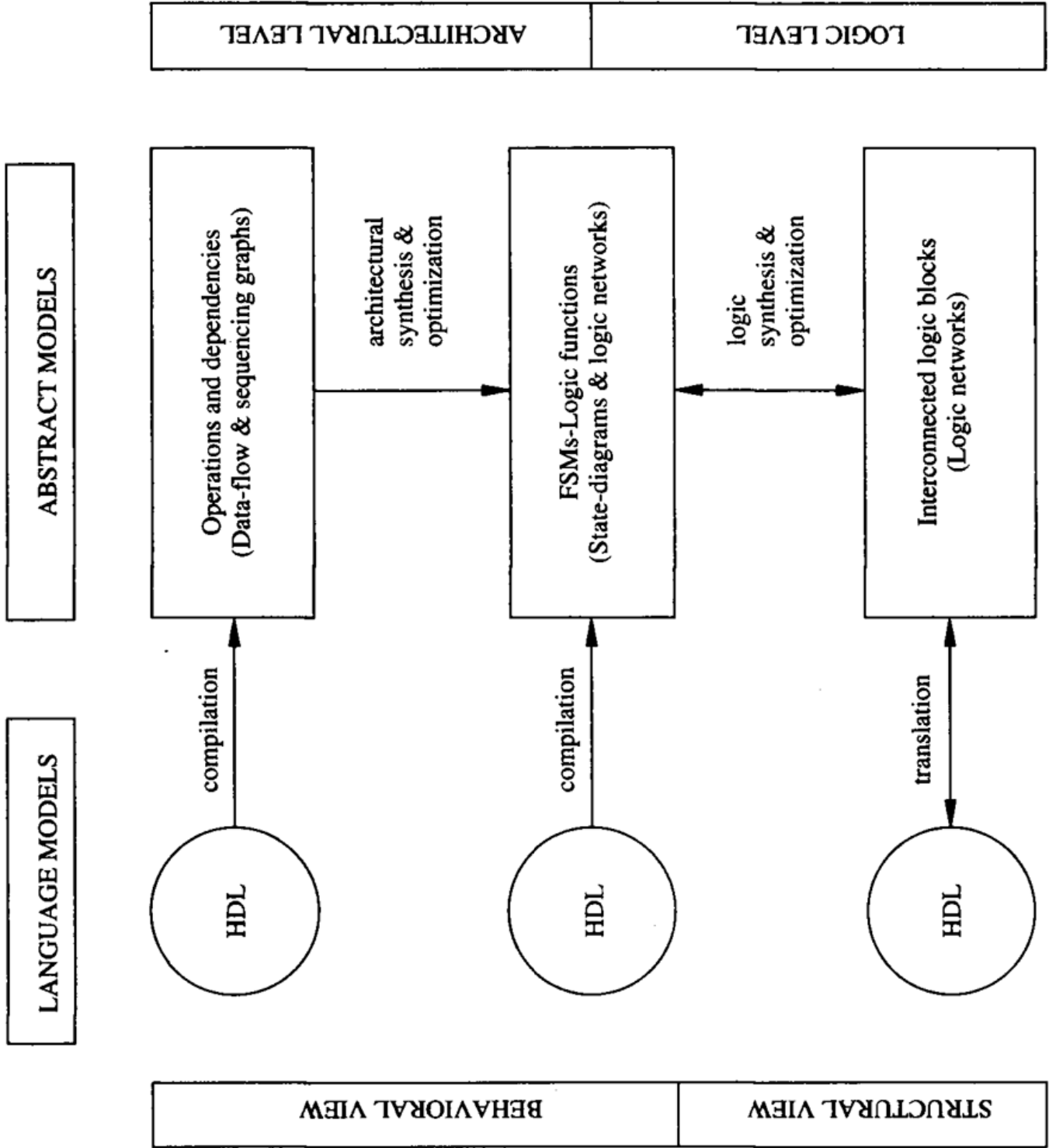
Τα Pareto points είναι λύσεις σε προβλήματα βελτιστοποίησης υπό περιορισμούς (constrained).

## Βελτιστοποίηση σε επίπεδο λογικής

- ✓ Ελαχιστοποίηση επιφάνειας με περιορισμό στην καθυστέρηση.
- ✓ Ελαχιστοποίηση καθυστέρησης με περιορισμό στην επιφάνεια.

## Βελτιστοποίηση σε επίπεδο αρχιτεκτονικής

- ✓ Ελαχιστοποίηση επιφάνειας με περιορισμό στο latency.
- ✓ Ελαχιστοποίηση του latency με περιορισμό στην επιφάνεια.



# Βασικά στοιχεία σύνθεσης

---

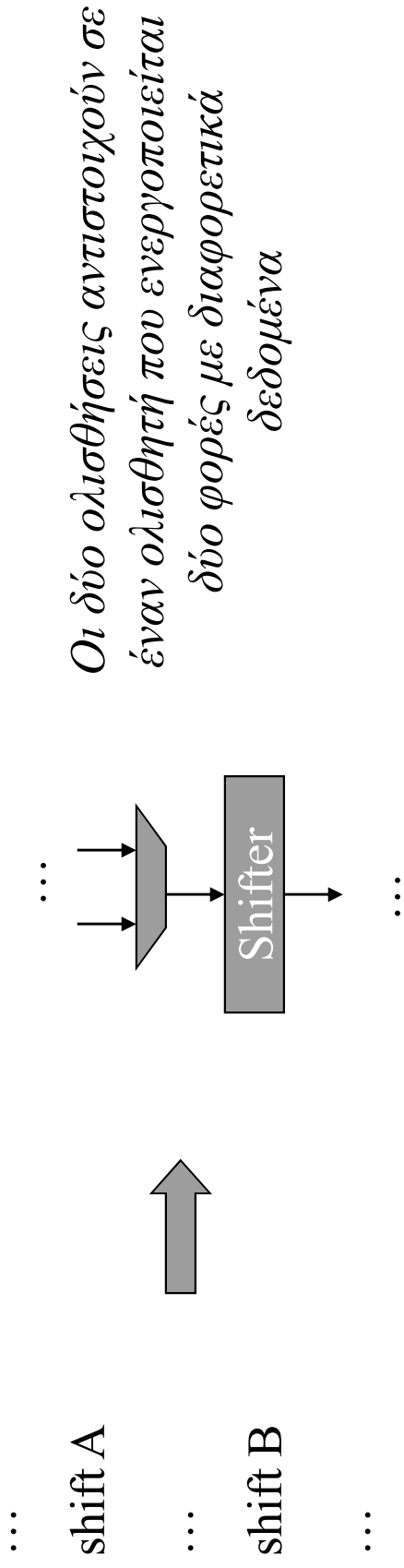
- ✓ High-level formulation
- ✓ Languages and representations
- ✓ Design modeling (διαφορετικοί τρόποι μοντελοποίησης).
- ✓ Design quality measures (προσεγγιστικές για αποφάσεις σε πρώιμο στάδιο).
- ✓ High-level synthesis algorithms
- ✓ Technology mapping
- ✓ Databases
- ✓ Interactive environments

# Design modeling

---

- ✓ Υπάρχουν πολλοί τρόποι μοντελοποίησης.
- ✓ Η περιγραφή συμπεριφοράς διαφέρει πολύ από το τελικό κατασκευαστικό μοντέλο.

## Παράδειγμα



# Design Quality Measures

---

- ✓ Απαιτείται εκτίμηση απόδοσης σχεδίασης.
- ✓ Η μέτρηση μόνο του αριθμού των λειτουργικών μονάδων δεν αρκεί.
- ✓ Η ακριβής μέτρηση γίνεται με την ολοκλήρωση της σχεδίασης (χρονοβόρο).
- ✓ Η μέτρηση πρέπει να γίνεται νωρίς για να παίρνονται αποφάσεις σχεδιαστικές.

Ακρίβεια: λάθος εκτιμώμενου και πραγματικού

Κριτήρια  
μέτρησης

Αξιοπιστία: Απόκλιση μέσου λάθους

Απλότητα: Μικρός χρόνος υπολογισμού

Ακόμη και **μικρή ακρίβεια** με **μεγάλη αξιοπιστία** είναι χρήσιμη στην σύγκριση  
σχεδιάσεων

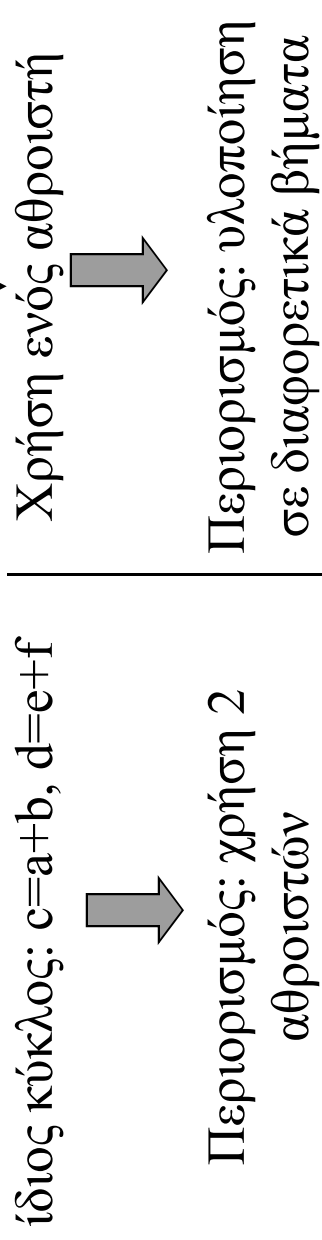
(η επιλογή της καλύτερης θα είναι ορθή ασχέτως με το πόσο καλύτερη θα είναι αυτή)



# Αλγόριθμοι Σύνθεσης

---

- ✓ Partitioning: διαίρεση μίας περιγραφής σε τμήματα.
- ✓ Scheduling: ανάθεση λειτουργιών σε χρονικά διαστήματα.  
Το ένα θέτει περιορισμούς στο άλλο
- ✓ Allocation: ανάθεση λειτουργιών σε λειτουργικές μονάδες και μονάδες αποθήκευσης.
- ✓ Optimization: βελτιστοποίηση σχεδίασης.



# Technology Mapping

---


- ✓ Παρέχονται μονάδες ποικίλων παραμέτρων (μέγεθος, ταχύτητα, ...)
- ✓ Η αντιστοίχιση γίνεται αυτόματα από τα εργαλεία.
- ✓ Κάθε εξέλιξη της τεχνολογίας οδηγεί στην δημιουργία νέων βιβλιοθηκών.
- ✓ Κάθε παλιός σχεδιασμός μπορεί να αντιστοιχηθεί αυτόματα σε νέα τεχνολογία.
- ✓ Υπάρχουν τεχνικές αντιστοίχισης που λαμβάνουν υπόψη περιορισμούς

# Εξομοίωση (simulation)

---

*Αναλύει τις αποκρίσεις του κυκλώματος σε μία ακολουθία εισόδων.*

## **Είδη εξομοίωσης:**

- Circuit Level simulation: Καθορισμός δυναμικού σε κάθε σημείο του κυκλώματος (διαφορικές εξισώσεις).  Μεγάλη υπολογιστική ισχύ - χρόνο.
- Logic Level simulation: Καθορισμός λογικών τιμών στους κόμβους του κυκλώματος. Δεν είναι ακριβής
- Functional Level simulation: Εξομοίωση σε υψηλό επίπεδο των HDL μοντέλων. Προλαμβάνει τα περισσότερα σχεδιαστικά λάθη

# Verification

---

- Η επιβεβαίωση είναι η σύγκριση δύο μοντέλων για έλεγχο συνέπειας
- Implementation verification (επιβεβαίωση της σύνθεσης) : Συγκρίνονται δύο μοντέλα στο ίδιο επίπεδο που έχουν όμως προκύψει με διαφορετικό τρόπο.
  - Design verification: Σύγκριση του μοντέλου στο επίπεδο της αρχιτεκτονικής με κάποιο αφαιρετικό μοντέλο (επιβεβαίωση μοντελοποίησης)
- 

*Οι αυτόματες μέθοδοι σύνθεσης (CAD) δεν είναι απαραίτητα αλάνθαστες στην σύνθεση κυκλωμάτων. Το verification και σε αυτήν την περίπτωση είναι απαραίτητο*

# Testing

---

Ο έλεγχος μετά την κατασκευή είναι απαραίτητος για την ανίχνευση κατασκευαστικών λαθών.

- Πρόσθετες συναρτήσεις τοποθετούνται για να κάνουν έναν σχεδιασμό ελέγξιμο (TPGs, LSSDs, scan chains, ...).
- Το κύκλωμα που δεν είναι πλήρως ελέγξιμο έχει μικρότερη αξία.
- Υπάρχουν τεχνικές βελτίωσης της ελεγχιμότητας σε λογικό και αρχιτεκτονικό επίπεδο.
- Απαιτείται απομάκρυνση του πλεονασμού.
- Εξαρτάται από το μοντέλο σφαλμάτων που χρησιμοποιείται.