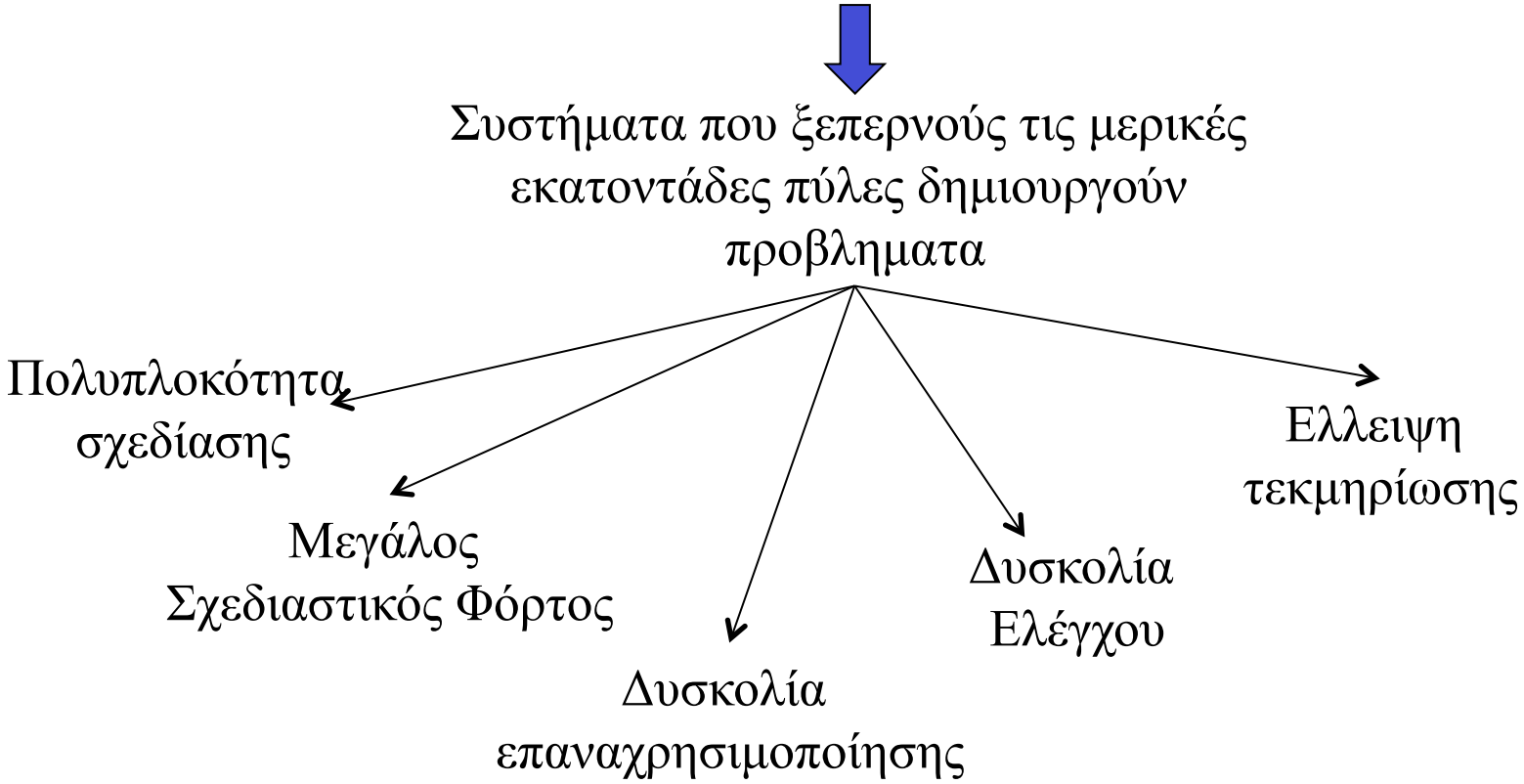
ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ
ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
ΕΛΛΑΔΑ ΚΟΙΝΩΝΙΚΟ ΠΛΑΙΣΙΟ ΣΤΗΡΙΞΗΣ 2008
Ανάπτυξη παντού. Ανάπτυξη για όλους.
Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ
Επιχειρησιακό Πρόγραμμα
Εκπαίδευσης και Αρχικής
Επαγγελματικής Κατάρτισης

7η Θεματική Ενότητα : Εισαγωγή στις Γλώσσες Περιγραφής Υλικού

Εισαγωγή

Η χειρονακτική σχεδίαση ενός ψηφιακού συστήματος είναι εξαιρετικά δύσκολη και επιρρεπής σε λάθη



Γενικά

Τα σύγχρονα ψηφιακά συστήματα είναι αρκετά περίπλοκα (πολλά εκατομμύρια πύλες). Απαιτείται αντιμετώπιση της πολυπλοκότητας για σχεδίαση με σιγουριά τήρησης των προδιαγραφών.

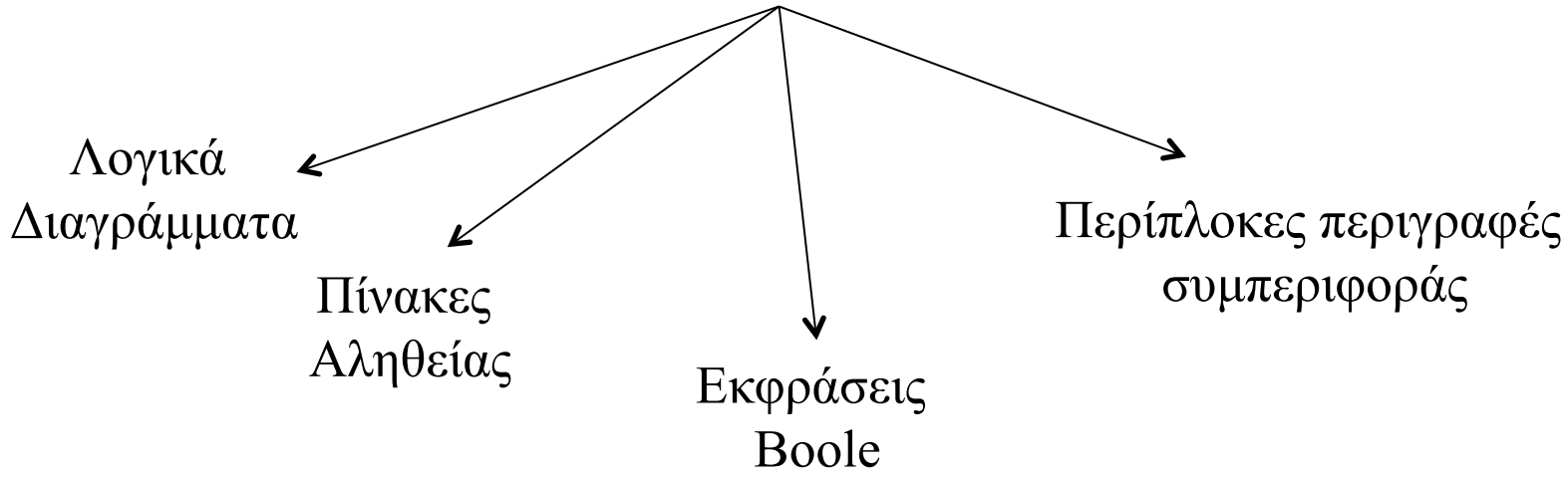
Χρήση αυτοματοποιημένων μεθόδων και τεχνικών τμηματοποίησης της σχεδίασης

Γλώσσες Περιγραφής Υλικού (HDL)

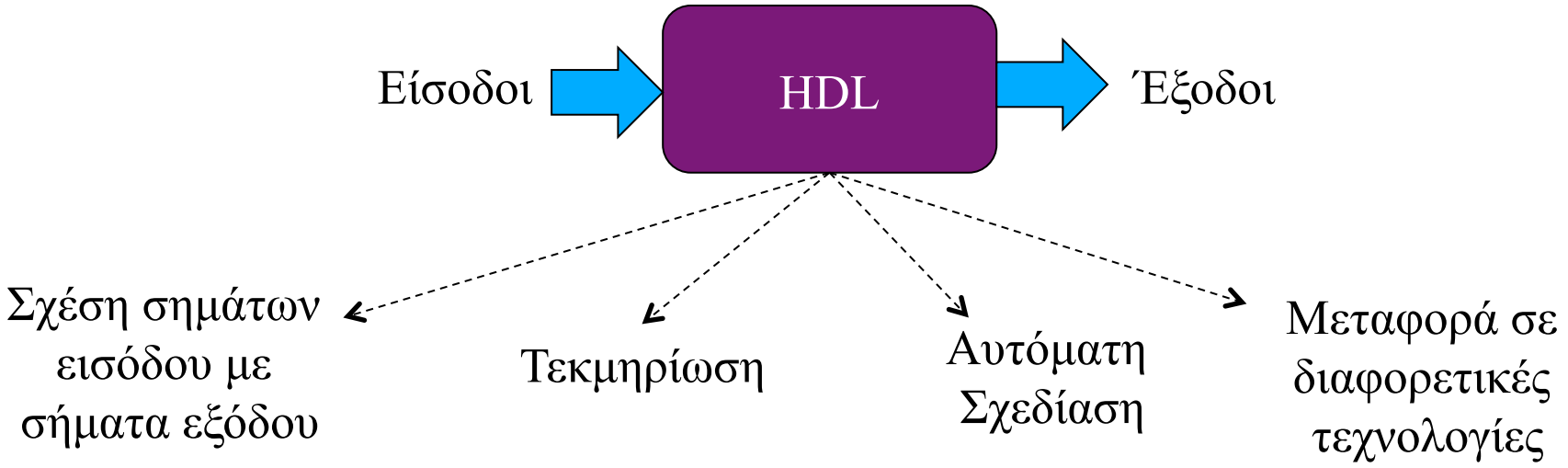
HDLs

Μία γλώσσα περιγραφής υλικού είναι μία γλώσσα για Η/Υ που περιγράφει ενα σύστημα σε μορφή κειμένου.

Μοιάζουν πολύ σε γλώσσες προγραμματισμού όπως η C
αλλά περιγράφουν την συμπεριφορά λογικών κυκλωμάτων



HDLs



Μοντέλο:
Τρόπος αναπαράστασης πληροφοριών και επίπεδο αφαίρεσης.

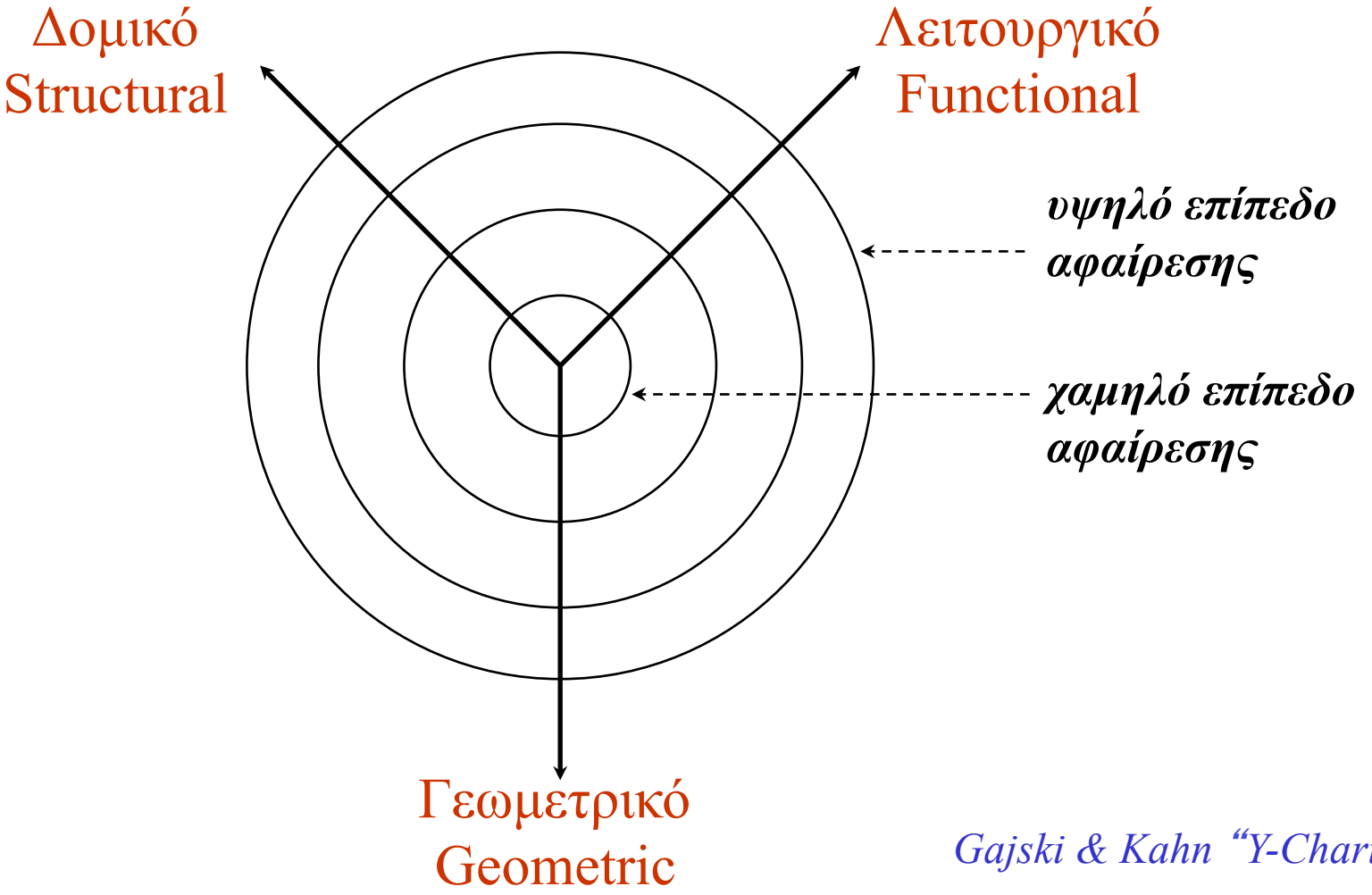
Οι HDLs υποστηρίζουν πολλά επίπεδα αφαίρεσης/ μοντέλα

Βασικές Αρχές

- *Μοντέλα Διασυνδέσεων (Interface Models)*
- *Μοντέλα Συμπεριφοράς (Behaviour Models)*
- *Μοντέλα Δομής (Structure Models)*
- *Μοντέλα Ελέγχου-Επαλήθευσης (Test Bench Models)*
- *Προσομοίωση (Simulation)*
- *Σύνθεση (Synthesis)*

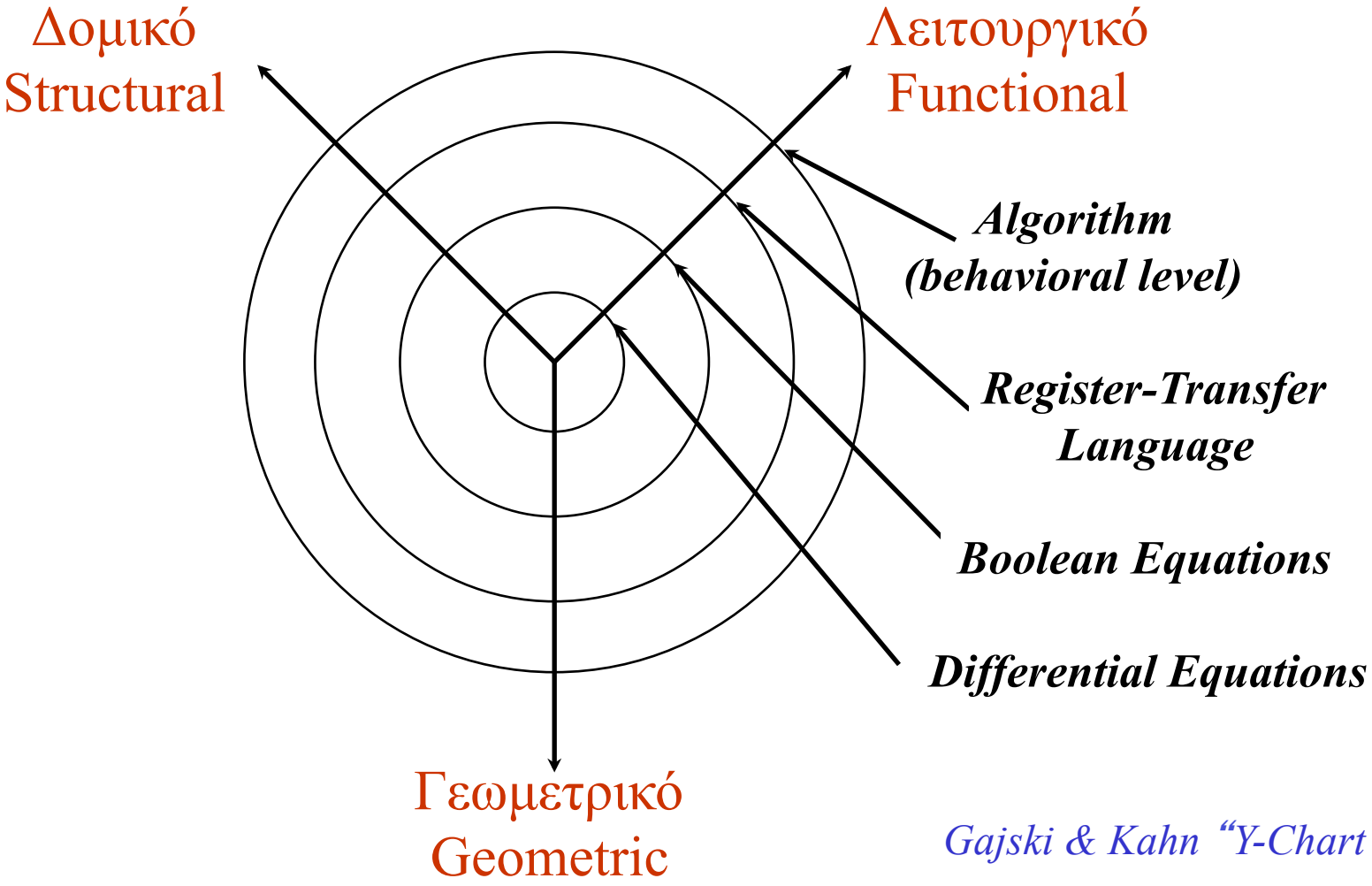
Η δομή μιας HDL σχεδίασης μοιάζει με τη δομή μίας σύγχρονης σχεδίασης λογισμικού: η HDL περιγράφει από κοινού ένα τρόπο διασύνδεσης/επικοινωνίας του σχεδιασμού με το εξωτερικό περιβάλλον καθώς και μία εσωτερική υλοποίηση.

Περιοχές & Επίπεδα Μοντελοποίησης I



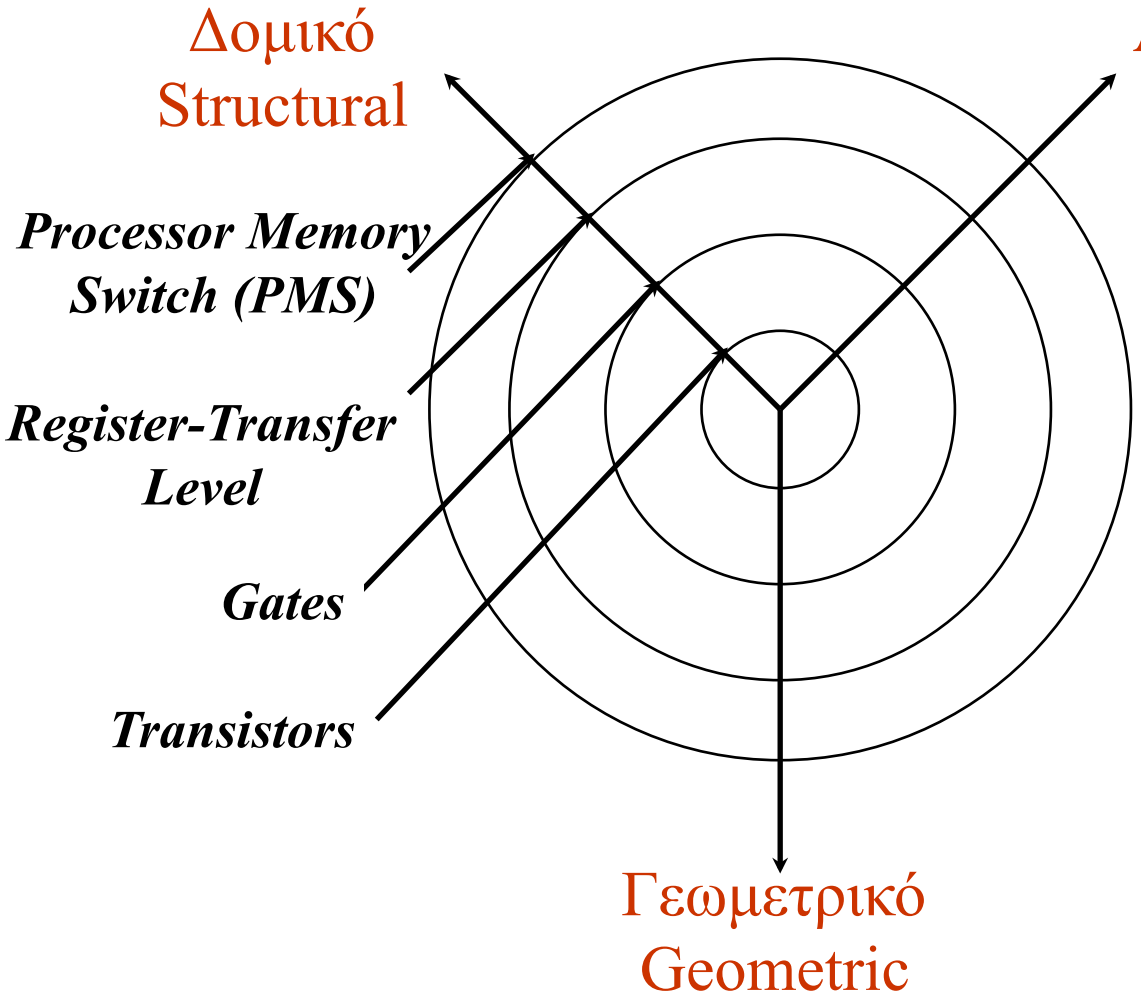
Gajski & Kahn "Y-Chart"

Περιοχές & Επίπεδα Μοντελοποίησης ΙΙ



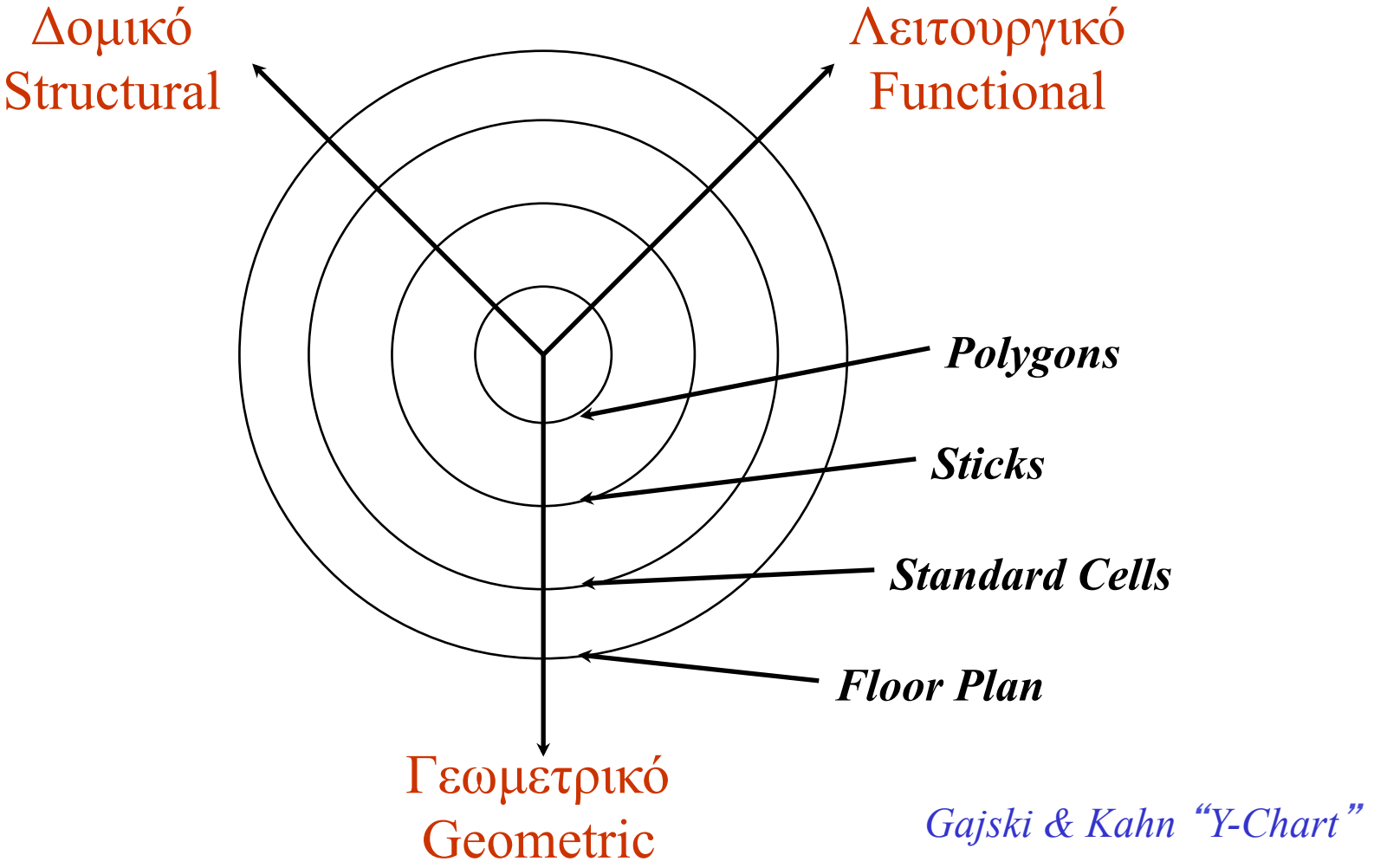
Gajski & Kahn "Y-Chart"

Περιοχές & Επίπεδα Μοντελοποίησης ΙΙΙ



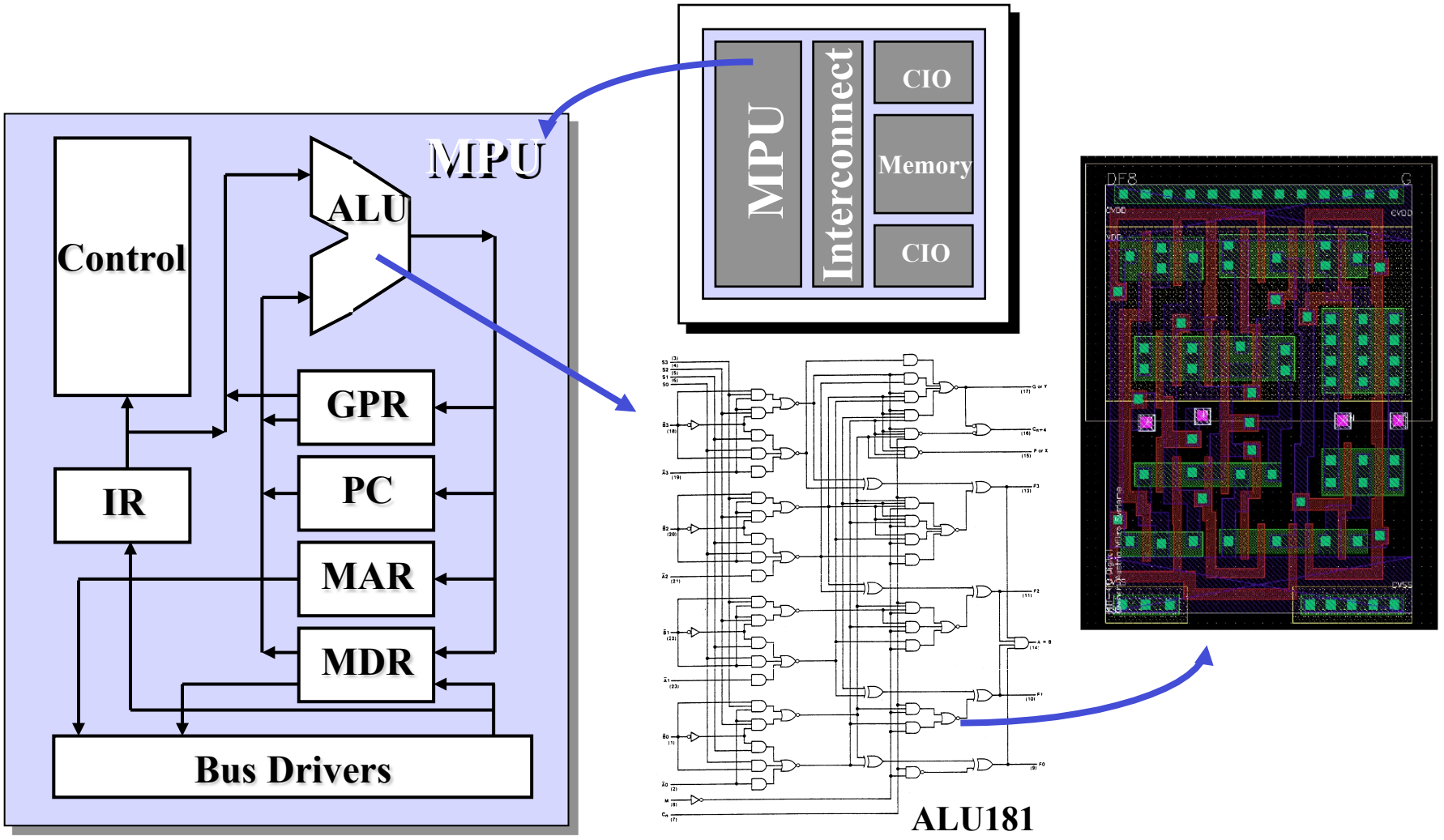
Gajski & Kahn "Y-Chart"

Περιοχές & Επίπεδα Μοντελοποίησης IV



Gajski & Kahn "Y-Chart"

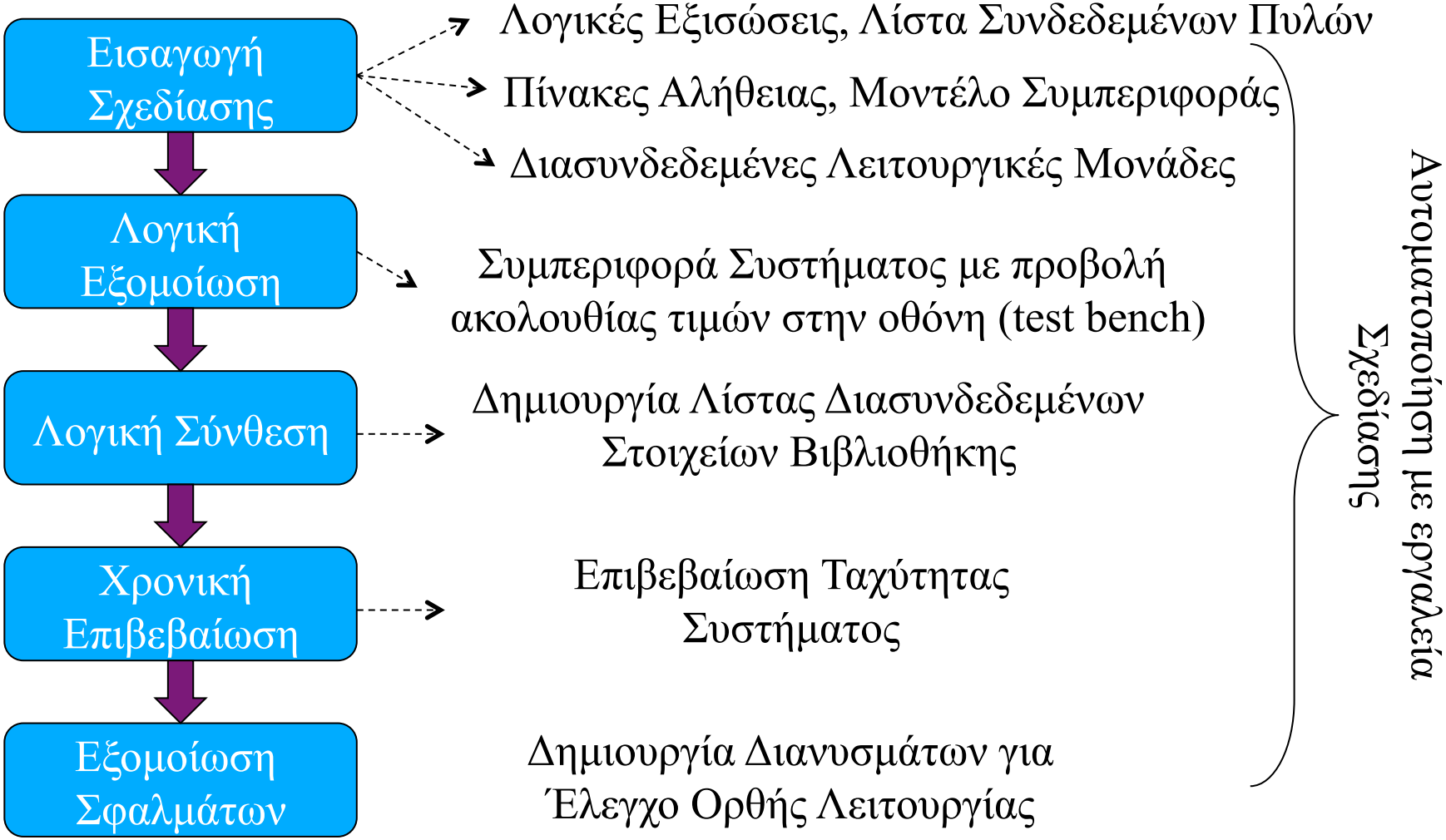
Παράδειγμα


MPU
Control
ALU
GPR
PC
IR
MAR
MDR
Bus Drivers
MPU
Interconnect
CIO
Memory
CIO
ALU181

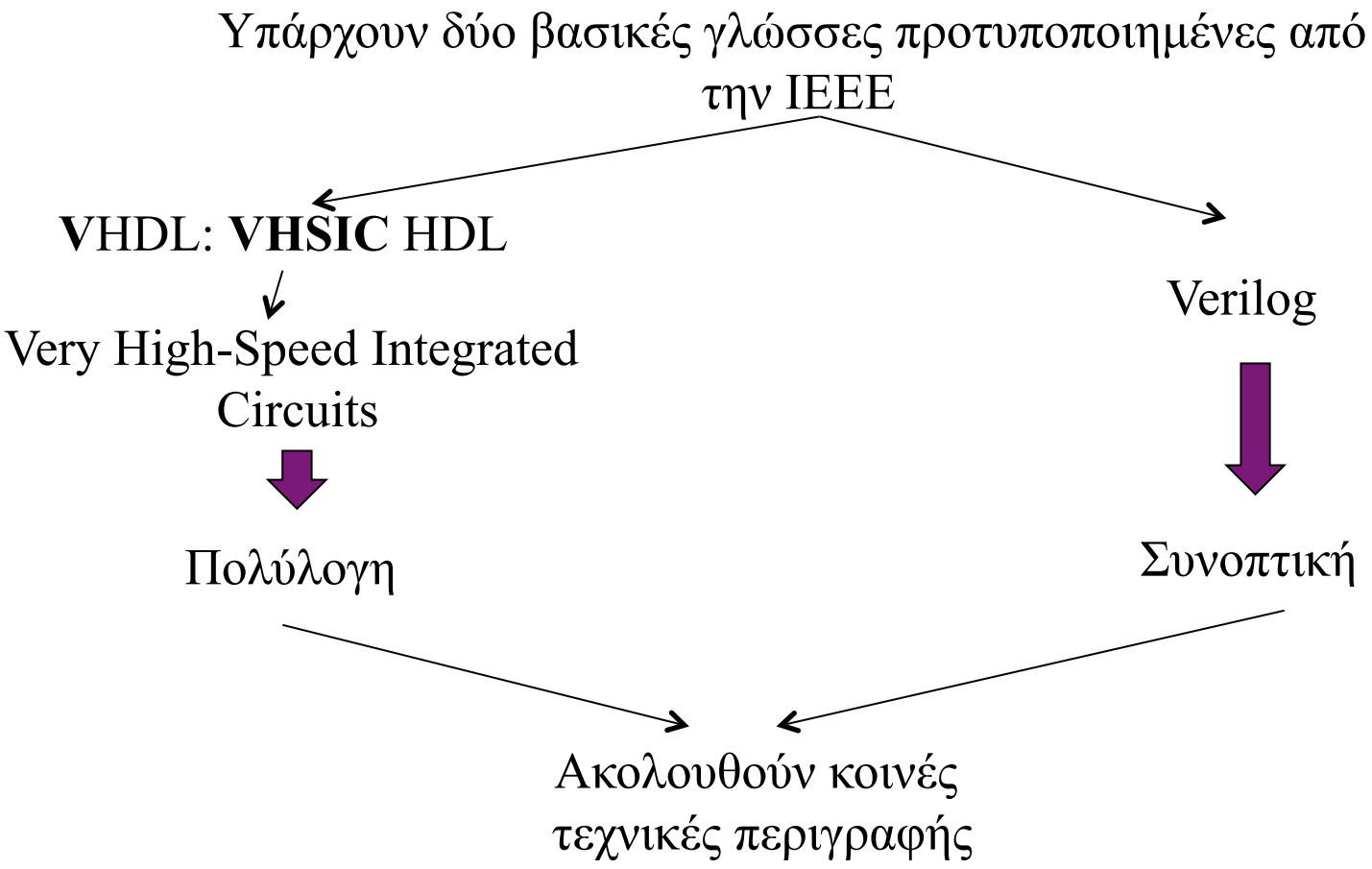
Μοντελοποίηση συστημάτων

- Ανάγκες χρήσης μοντέλων:
  - *καταγραφή προδιαγραφών*
  - *τεκμηρίωση*
  - *επαλήθευση με τη χρήση προσομοίωσης*
  - *τυπική επαλήθευση (formal verification)*
  - *σύνθεση*
- Στόχος
  - *αξιόπιστη διαδικασία σχεδίασης, με ταυτόχρονη ελαχιστοποίηση του κόστους και του απαιτούμενου χρόνου*
  - *αποφυγή σχεδιαστικών λαθών*

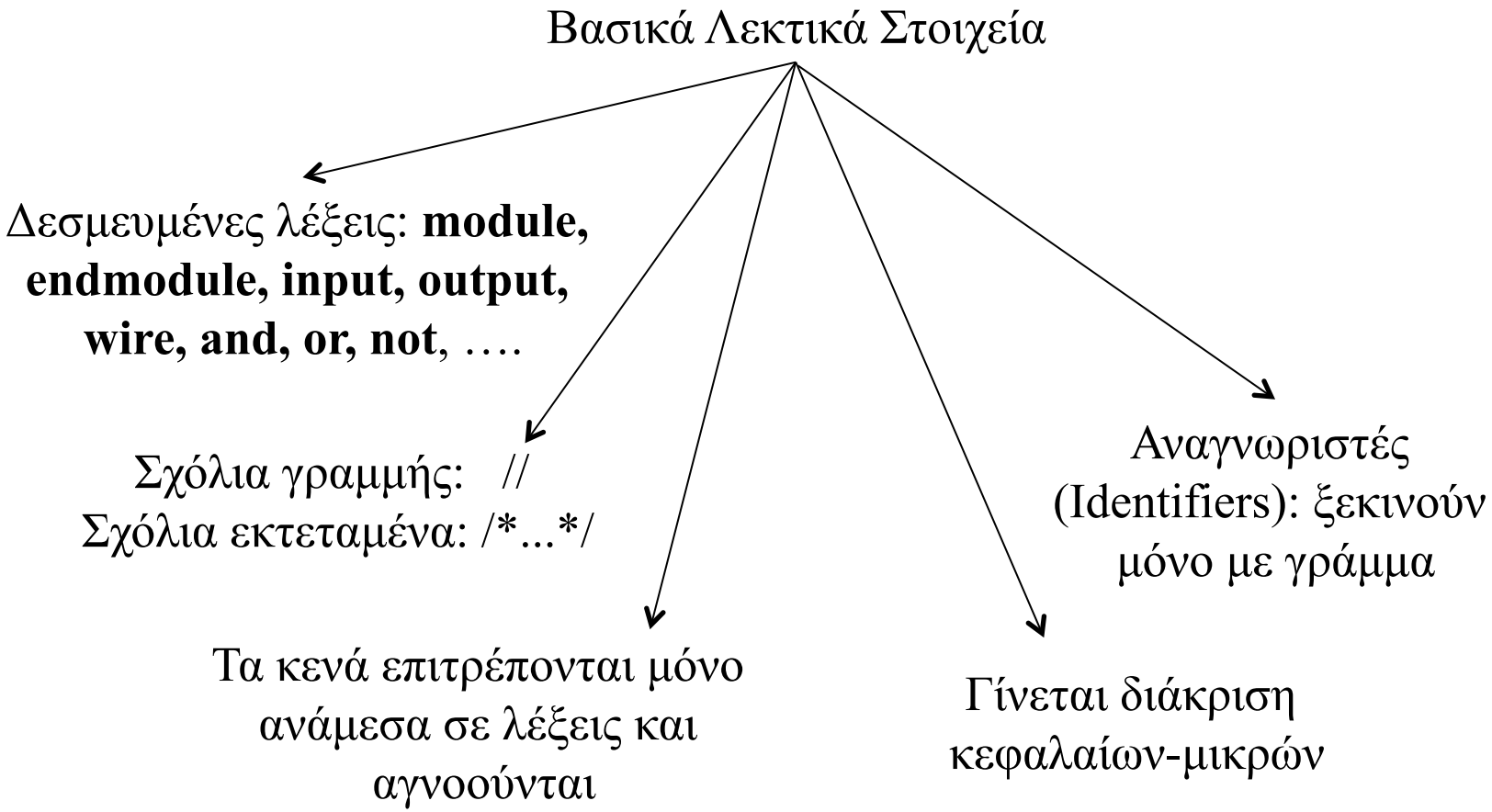
Βήματα Σχεδίασης Συστημάτων



Γλώσσες Περιγραφής



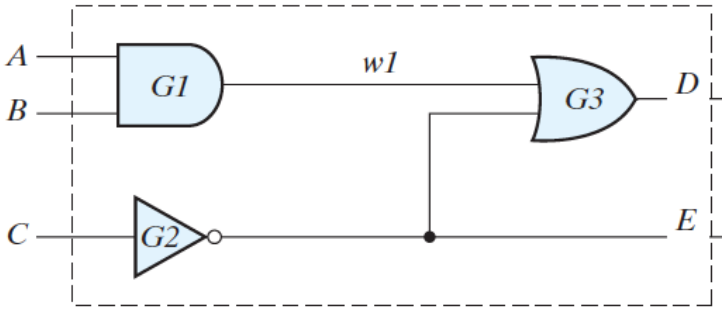
Η Γλώσσα Verilog



Το "Module"

Περιγραφή Συνδυαστικής Λογικής σε HDL

- Σχηματική Διασύνδεση Πυλών
- Σύνολο Λογικών Εξισώσεων
- Πίνακας Αλήθειας



Σχόλιο

```
// Verilog model of circuit of Figure 3.35. IEEE 1364–1995 Syntax
module  Simple_Circuit (A, B, C, D, E);
  output      D, E;
  input       A, B, C;
  wire        w1;

  and         G1 (w1, A, B); // Optional gate instance name
  not         G2 (E, C);
  or          G3 (D, w1, E);
endmodule
```

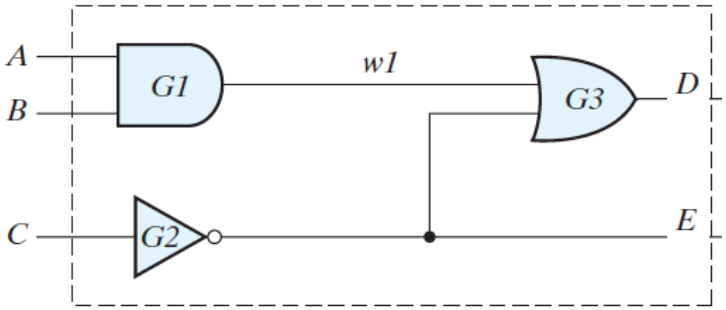
Δεσμευμένες Λέξεις

Αναγνωριστές

Το "Module"

Περιγραφή Συνδυαστικής Λογικής σε HDL

- Σχηματική Διασύνδεση Πυλών
- Σύνολο Λογικών Εξισώσεων
- Πίνακας Αλήθειας



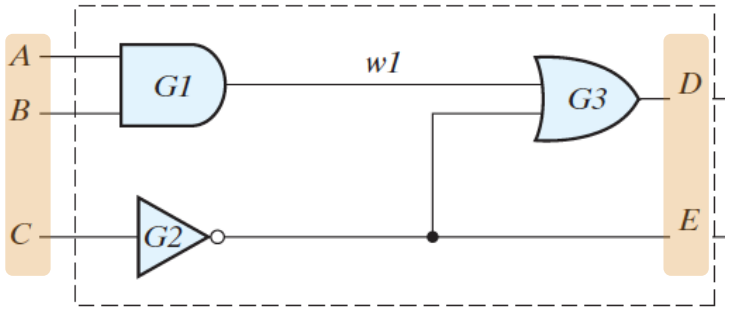

```
// Verilog model of circuit of Figure 3.35. IEEE 1364–1995 Syntax

module  Simple_Circuit (A, B, C, D, E);
  output        D, E;
  input         A, B, C;
  wire          w1;

  and           G1 (w1, A, B); // Optional gate instance name
  not           G2 (E, C);
  or            G3 (D, w1, E);
endmodule
```

Εδώ δεν μπαίνει ";"

Το "Module"

Περιγραφή Συνδυαστικής Λογικής σε HDL

- Σχηματική Διασύνδεση Πυλών
- Σύνολο Λογικών Εξισώσεων
- Πίνακας Αλήθειας



Όνομα Κυκλώματος

```
// Verilog model of circuit of Figure 3.35. IEEE 1364–1995 Syntax

module  Simple_Circuit (A, B, C, D, E);
output  D, E;
input   A, B, C;
wire    w1;

and     G1 (w1, A, B); // Optional gate instance name
not     G2 (E, C);
or      G3 (D, w1, E);
endmodule
```

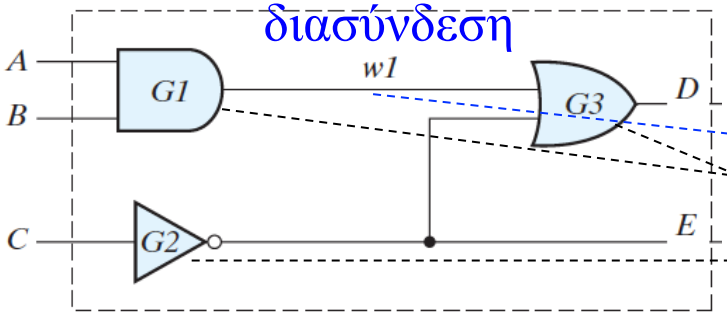
Θύρες E/E

Τύπος Θυρών

Το "Module"

Περιγραφή Συνδυαστικής Λογικής σε HDL

- Σχηματική Διασύνδεση Πυλών
- Σύνολο Λογικών Εξισώσεων
- Πίνακας Αλήθειας




```
// Verilog model of circuit of Figure 3.35. IEEE 1364–1995 Syntax

module  Simple_Circuit (A, B, C, D, E);
  output        D, E;
  input         A, B, C;
  wire          w1;

  and           G1 (w1, A, B); // Optional gate instance name
  not           G2 (E, C);
  or            G3 (D, w1, E);
endmodule
```

Έξοδοι (προηγούνται)

Είσοδοι (σε οποιαδήποτε σειρά)

Primitive Gates (πρωταρχικές πύλες)

Στιγμιότυπα Πυλών

Συνδέσεις θυρών

Δήλωση ή Στιγμιότυπο;

```
// Verilog model of circuit of Figure 3.35. IEEE 1364–1995 Syntax

module  Simple_Circuit (A, B, C, D, E);
 output         D, E;
 input          A, B, C;
 wire           w1;

 and            G1 (w1, A, B); // Optional gate instance name
 not            G2 (E, C);
 or             G3 (D, w1, E);
endmodule
```

Δήλωση του Module *Simple_Circuit*

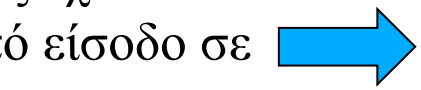
Μπορούμε να δημιουργήσουμε πολλά στιγμιότυπα του *Simple_Circuit*

Στιγμιότυπα Πρωταρχικών Πυλών (δεν χρειάζονται δήλωση)

Η σειρά των στιγμιότυπων δεν παίζει κανένα ρόλο.

Καθυστερήσεις Πυλών

Όλες οι πύλες έχουν καθυστέρηση από είσοδο σε έξοδο



Οι καθυστερήσεις είναι σημαντικές στην εξομοίωση

Μονάδα μέτρησης (default : 1ns)

Οδηγία προς μεταγλωτιστή

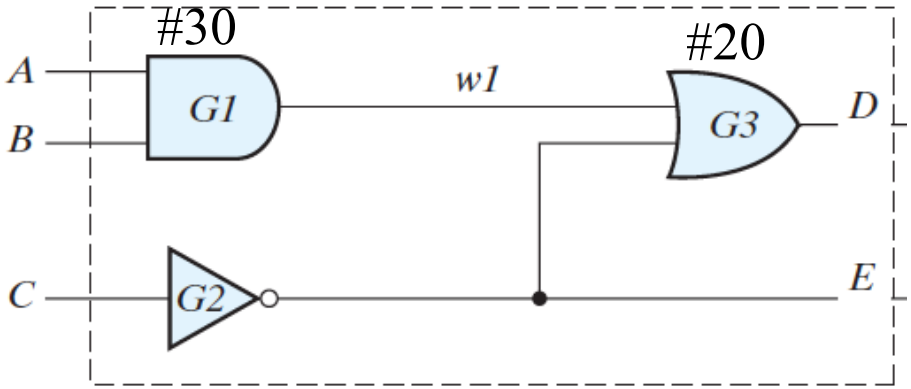
timescale **1ns/100ps**

Μονάδα στρογγυλοποίησης

```
// Verilog model of simple circuit with propagation delay

module Simple_Circuit_prop_delay (A, B, C, D, E);
 output D, E;
 input  A, B, C;
 wire   w1;

 and    #(30) G1 (w1, A, B);
 not    #(10) G2 (E, C);
 or     #(20) G3 (D, w1, E);
endmodule
```

Καθυστερήσεις Πυλών (#)

Καθυστερήσεις Πυλών



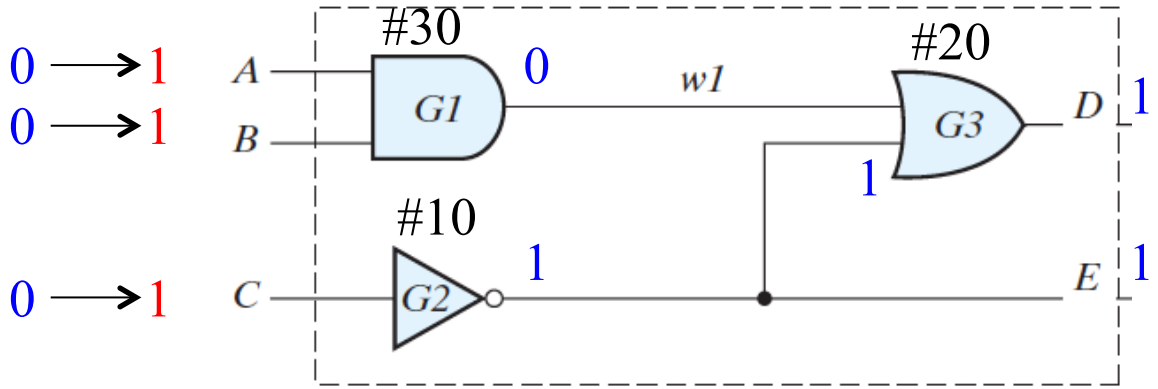
Αρχικό διάνυσμα εισόδου
(A, B, C) = (0, 0, 0)

Μετάβαση στο διάνυσμα
(A, B, C) = (1, 1, 1)

Τελική έξοδος (D, E) = (1, 0)

Οι εσωτερικές γραμμές και οι έξοδοι μεταβάλονται σε διακριτές χρονικές στιγμές που καθορίζονται από τις καθυστερήσεις των πυλών

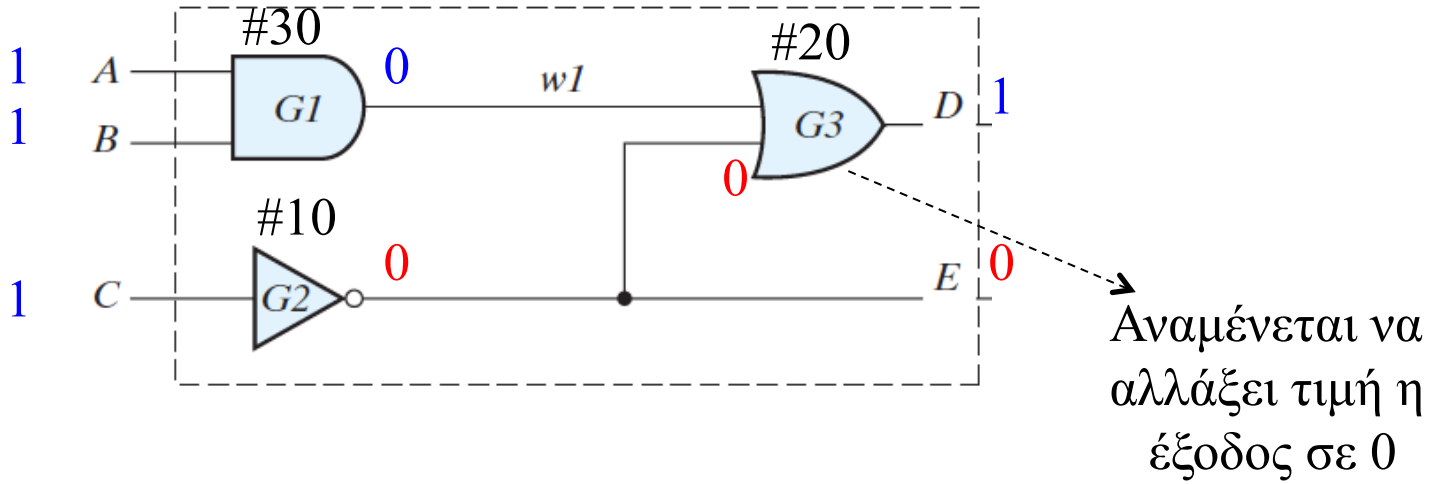
Καθυστερήσεις Πυλών



Χρονική στιγμή **0 ns**:

Οί είσοδοι μεταβαίνουν στην τιμή 1
όλες μαζί ταυτόχρονα

Καθυστερήσεις Πυλών

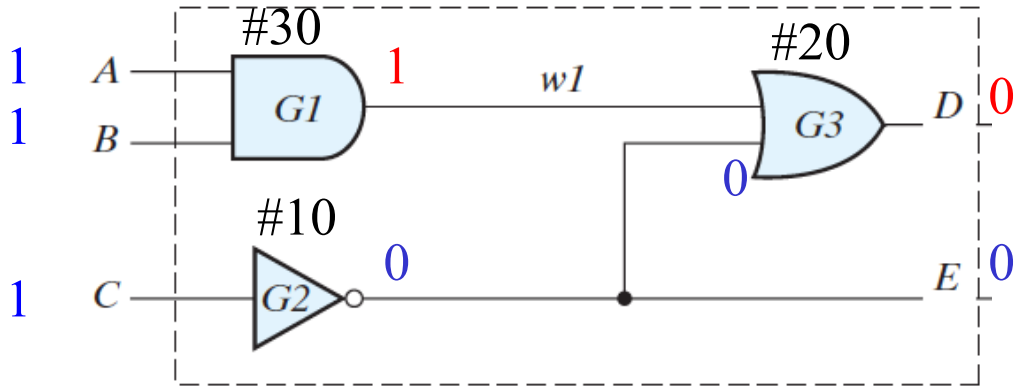


Χρονική στιγμή **10 ns**:

Η έξοδος της G2 μεταβαίνει στην τιμή 0

Μαζί αλλάζουν τιμή και η είσοδος της πύλης G3 και η έξοδος E αφού συνδέονται απευθείας στην έξοδο της G2

Καθυστερήσεις Πυλών

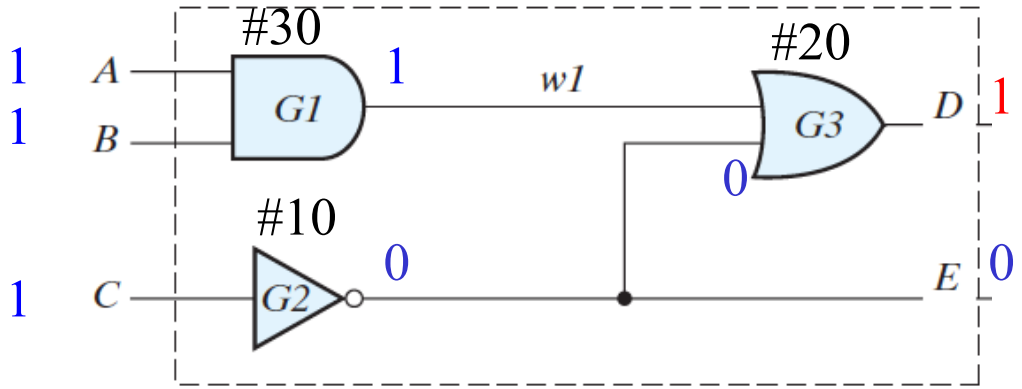


Χρονική στιγμή **30 ns**:

Η έξοδος της G1 μεταβαίνει στην τιμή 1

Προσοχή: έχουν περάσει 20 ns από την στιγμή που άλλαξε η κάτω είσοδος της G3 στο 0 οπότε και η έξοδος D μεταβαίνει στο 0

Καθυστερήσεις Πυλών



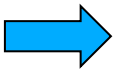
Χρονική στιγμή **50 ns**:

Η έξοδος της G3 μεταβαίνει στην τιμή 1 και σταθεροποιείται

	Time Units (ns)	Input ABC	Output E w1 D
Initial	—	0 0 0	1 0 1
Change	—	1 1 1	1 0 1
	10	1 1 1	0 0 1
	20	1 1 1	0 0 1
	30	1 1 1	0 1 0
	40	1 1 1	0 1 0
	50	1 1 1	0 1 1

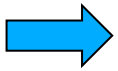
Test Bench

Η διαδικασία αυτή είναι πολύ δύσκολη αν γίνεται χειρονακτικά



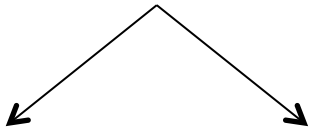
Χρησιμοποιούνται εξομοιωτές

Οι είσοδοι εφαρμόζωνται στο κύκλωμα με την χρήση ειδικών περιγραφών, τα test benches



Οι εσωτερικές τιμές και οι έξοδοι αποτυπώνονται στην οθόνη με χρήση ειδικού λογισμικού

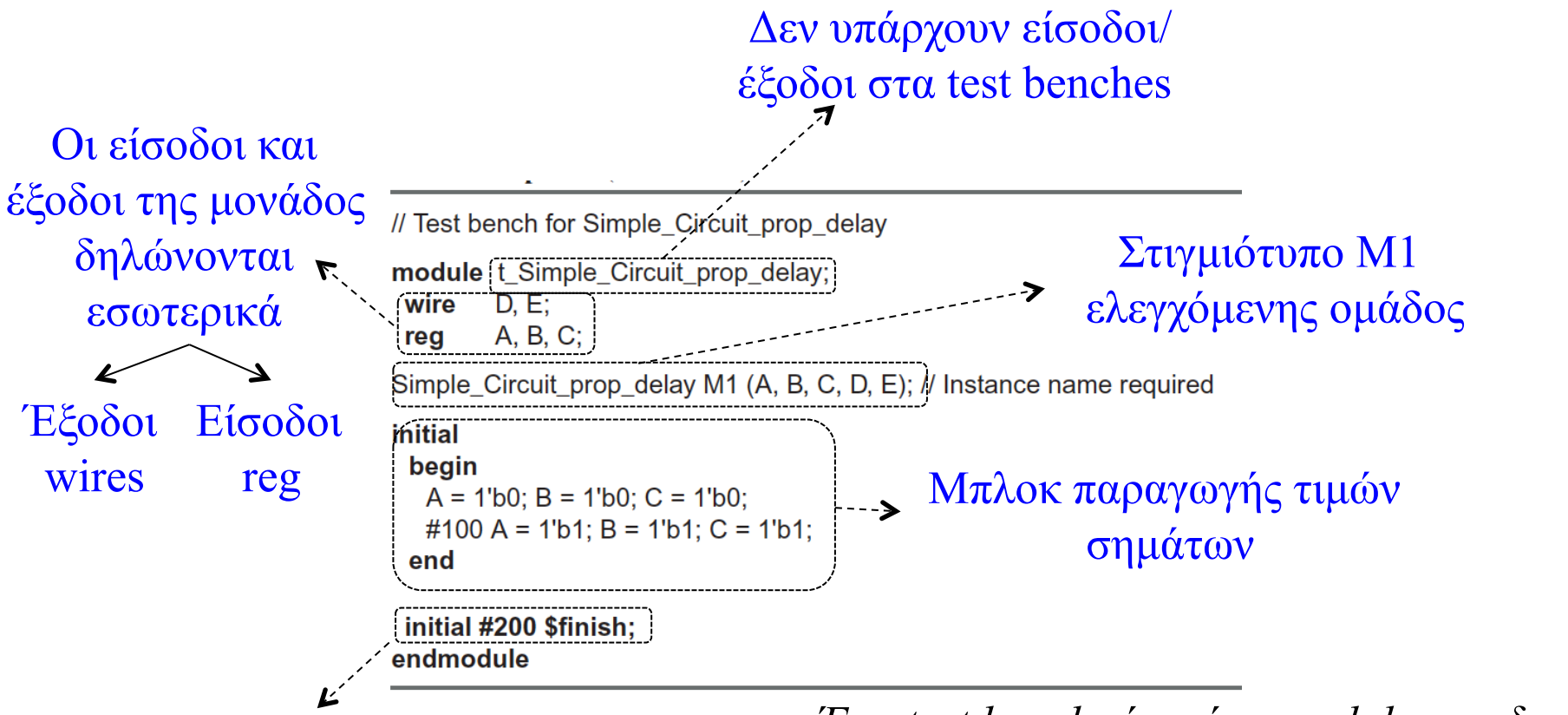
Κυμματομορφές Εξόδου & επιλεγμένων εσωτερικών κόμβων



Ενσωματώνουν τις περιγραφές των κυκλωμάτων

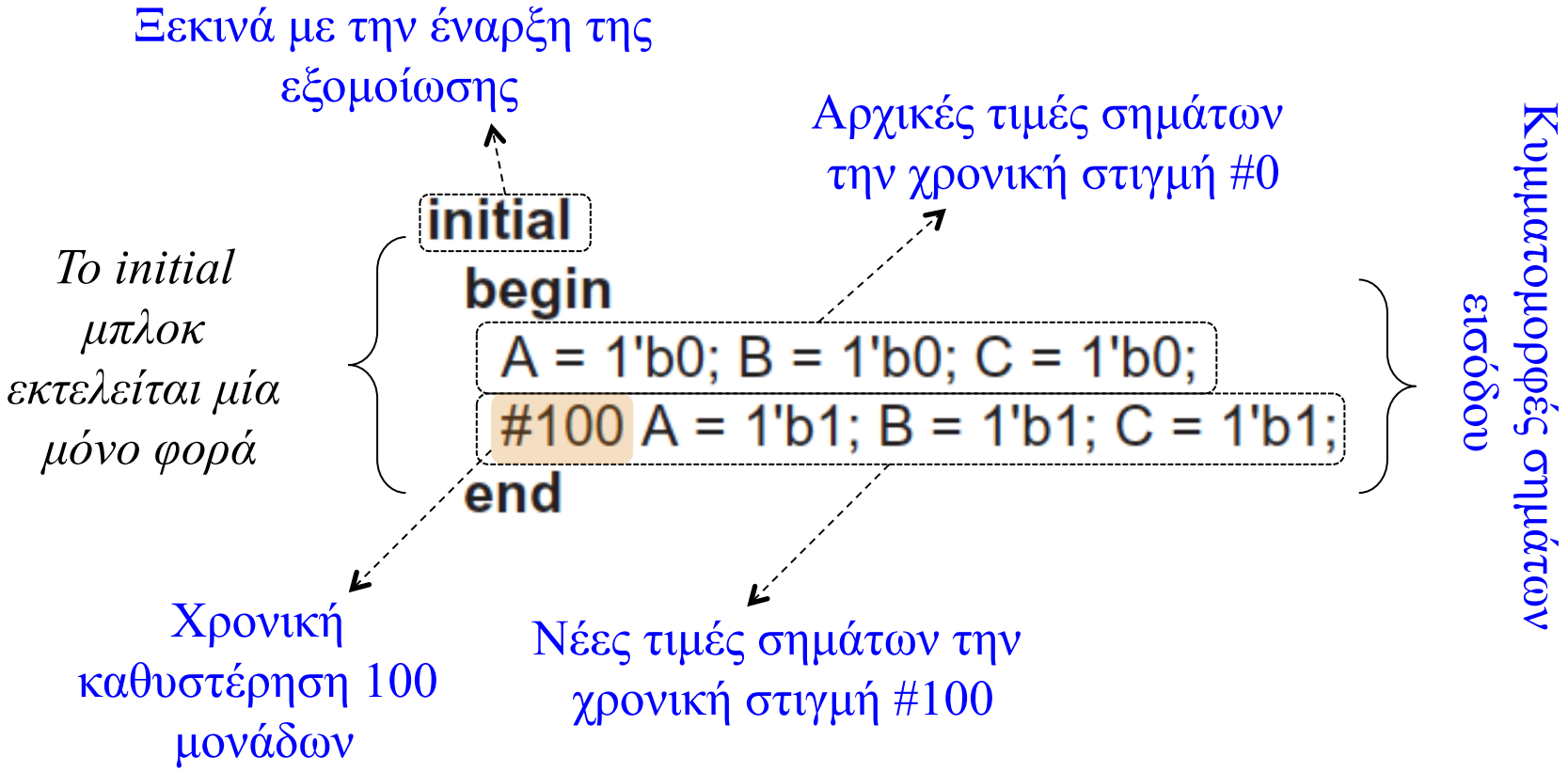
Ενσωματώνουν τις κυμματομορφές εισόδου

Test Bench

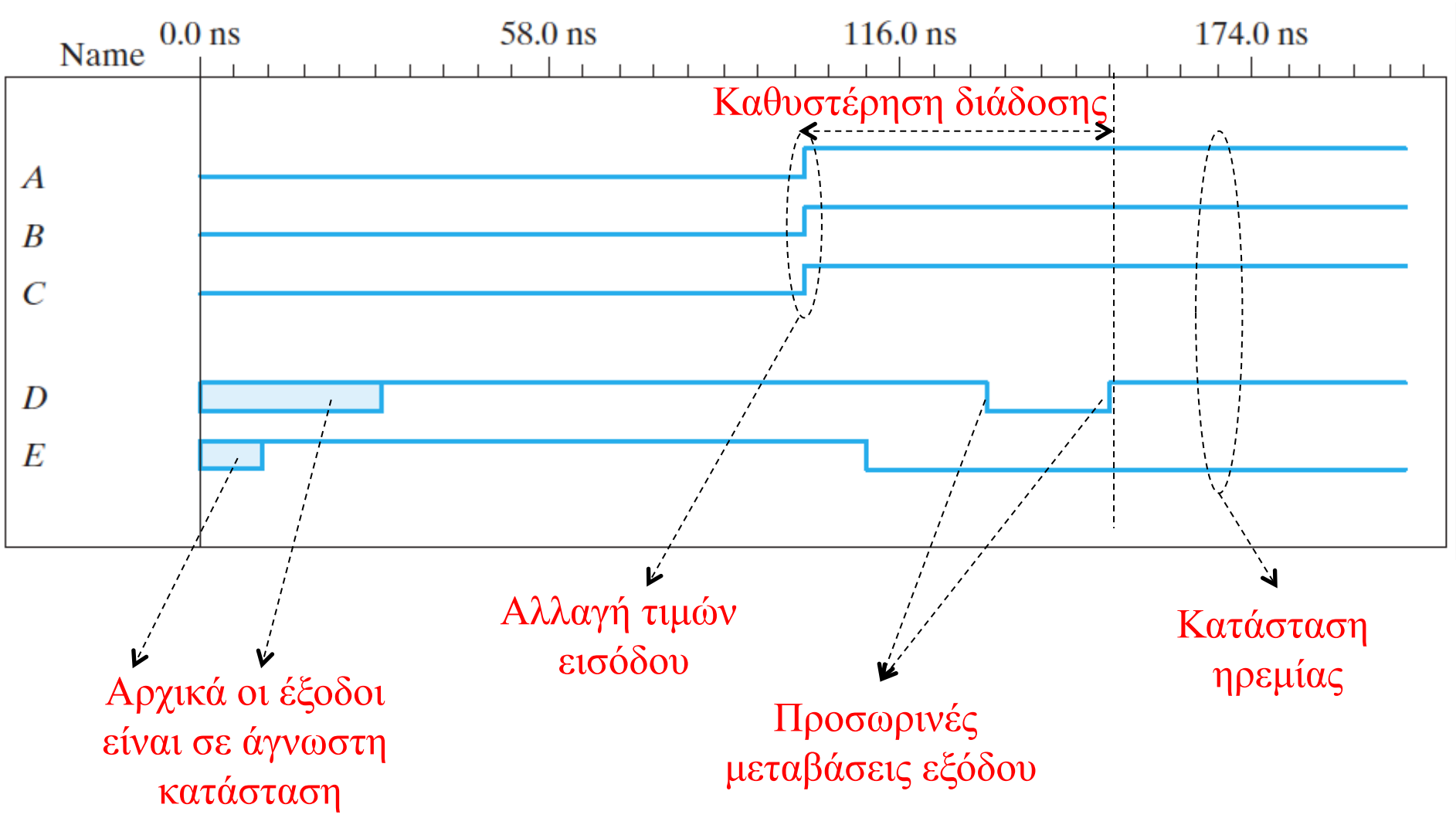


Ένα test bench είναι ένα module που δεν αντιστοιχεί σε πραγματικό κύκλωμα (δεν συντίθεται)

Signal Generator



Κυμματομορφές Εξόδου



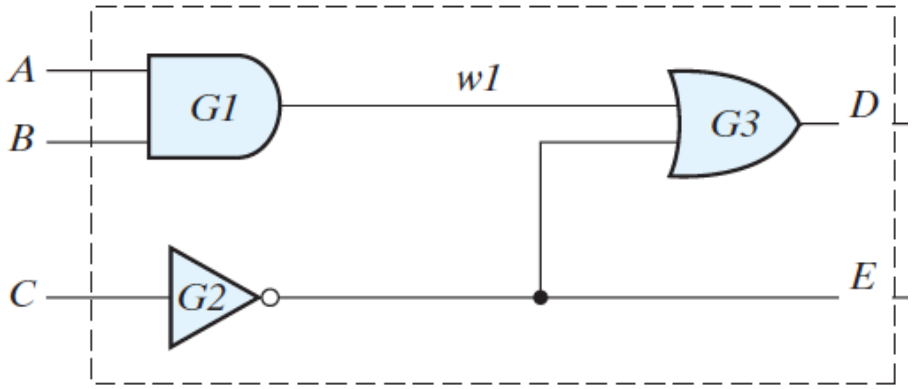
Λογικές Εκφράσεις

Στην Verilog μία έκφραση μπορεί να αποδοθεί απευθείας με τελεστές

assign

AND : &&
OR : ||
NOT : !

} Λογικοί Τελεστές



assign D = (A && B) || (!C);

assign E = !C;

Λογικές Εκφράσεις

Παράδειγμα

$E = A + BC + B'D$

$F = B'C + BC'D'$

```
// Verilog model: Circuit with Boolean expressions

module Circuit_Boolean_CA (E, F, A, B, C, D);
  output      E, F;
  input       A, B, C, D;

  assign E = A || (B && C) || ((!B) && D);
  assign F = ((!B) && C) || (B && (!C) && (!D));
endmodule
```

Δεν υπάρχουν ενδιάμεσα σήματα

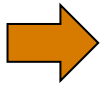
Η σειρά δεν έχει σημασία

Ταυτόχρονα (concurrent)

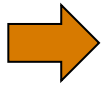
Οποιαδήποτε αλλαγή στα σήματα A, B, C, D προκαλεί επαναϋπολογισμό των E, F

Primitives οριζόμενα από τον χρήστη

```
and     G1 (w1, A, B);
not     G2 (E, C);
or      G3 (D, w1, E);
```



Primitives Συστήματος



Ο χρήστης μπορεί να ορίσει δικά του primitives

```
// Verilog model: User-defined Primitive
primitive UDP_02467 (D, A, B, C);
  output  D;
  input   A, B, C;
//Truth table for D 5 f (A, B, C) 5 Σ(0, 2, 4, 6, 7);
  table
//    A    B    C    :    D     // Column header comment
      0    0    0    :    1;
      0    0    1    :    0;
      0    1    0    :    1;
      0    1    1    :    0;
      1    0    0    :    1;
      1    0    1    :    0;
      1    1    0    :    1;
      1    1    1    :    1;
  endtable
endprimitive
```

Δεσμευμένη λέξη

Δήλωση Πίνακα

Πίνακας Αλήθειας του Primitive

Primitives οριζόμενα από τον χρήστη

```
module Circuit_with_UDP_02467 (e, f, a, b, c, d);
 output        e, f;
 input         a, b, c, d

 UDP_02467     (e, a, b, c);
 and           (f, e, d);          // Option gate instance name omitted
endmodule
```