dscal
DIGITAL SYSTEMS & COMPUTER ARCHITECTURE LABORATORY

1ο Εργαστήριο Σχεδίασης Ψηφιακών Συστημάτων

1η Διάλεξη

Εισαγωγή

Βασιλόπουλος Διονύσης

Ε.ΔΙ.Π. Τμήματος Πληροφορικής & Τηλεπικοινωνιών

ΤΜΗΜΑ
ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Αντικείμενο Εργαστηρίου

- Πρακτικό μέρος του μαθήματος Σχεδίαση Ψηφιακών Συστημάτων
- Συνοπτική επανάληψη θεωρίας (όπου χρειάζεται)
- Προγραμματισμός στη γλώσσα VHDL
- Περιγράφουμε (σε VHDL) ένα Σύστημα Λογικού Κυκλώματος/Ψηφιακό Κύκλωμα
- Με τι προγραμματίζουμε; Με το εργαλείο Vivado
- Τι προγραμματίζουμε; Hardware (κάρτες FPGA)
- Τι κάνει το Λογικό Κύκλωμα; Δίνει λύση σε μία πραγματική ανάγκη

Αντικείμενο Μαθήματος

Πραγματικές Ανάγκες

Ψηφιακά Συστήματα

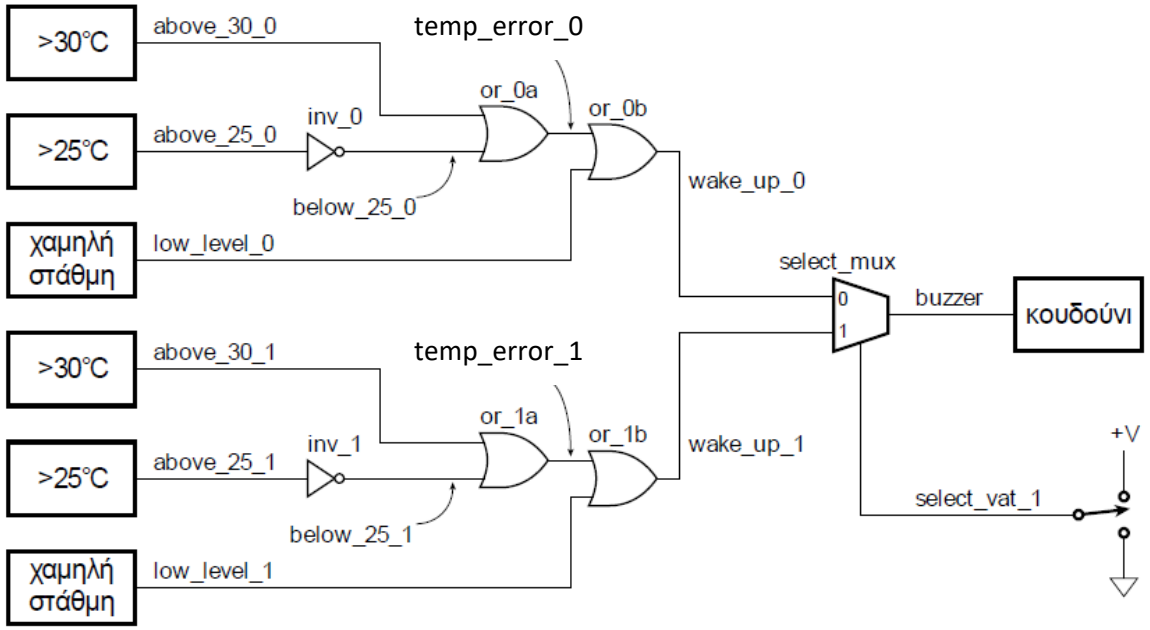
Αντικείμενο Μαθήματος

Εργαστήριο – Περιεχόμενο μαθήματος

- **Εκμάθηση Γλώσσας Προγραμματισμού VHDL**
- **Ανάπτυξη εφαρμογών** για κατανόηση του μοντέλου της VHDL
- **Χρήση του εργαλείου VIVADO** για την ανάπτυξη εφαρμογών
- **Συνοπτική παρουσίαση σε βασικά στοιχεία της θεωρίας Λογικής Σχεδίασης** (δυαδικό σύστημα αρίθμησης, λογικές πύλες, συνδυαστικά κυκλώματα, ακολουθιακά κυκλώματα, μνήμες)
- **Προγραμματισμός κάρτας FPGA** (μέσω VHDL+VIVADO) σε πραγματικές συνθήκες (εργαστήριο)

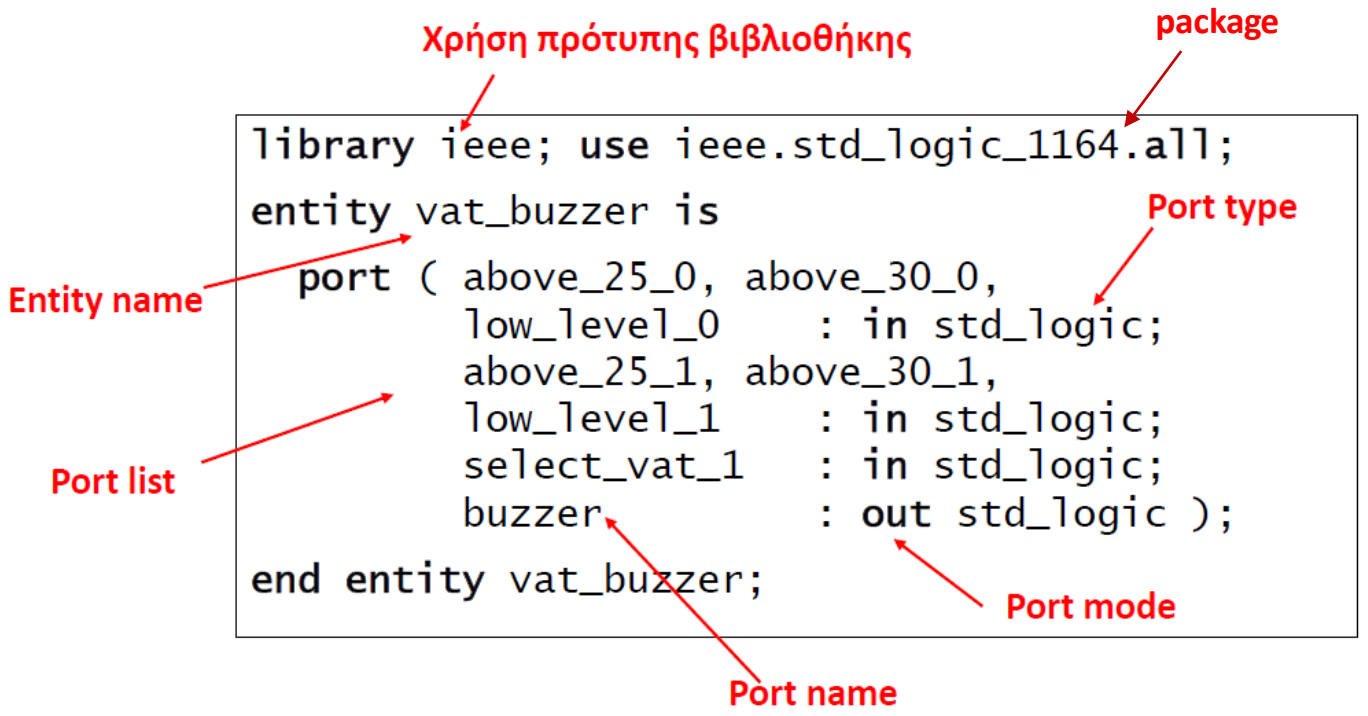
Αντικείμενο Μαθήματος

Λογικά κυκλώματα



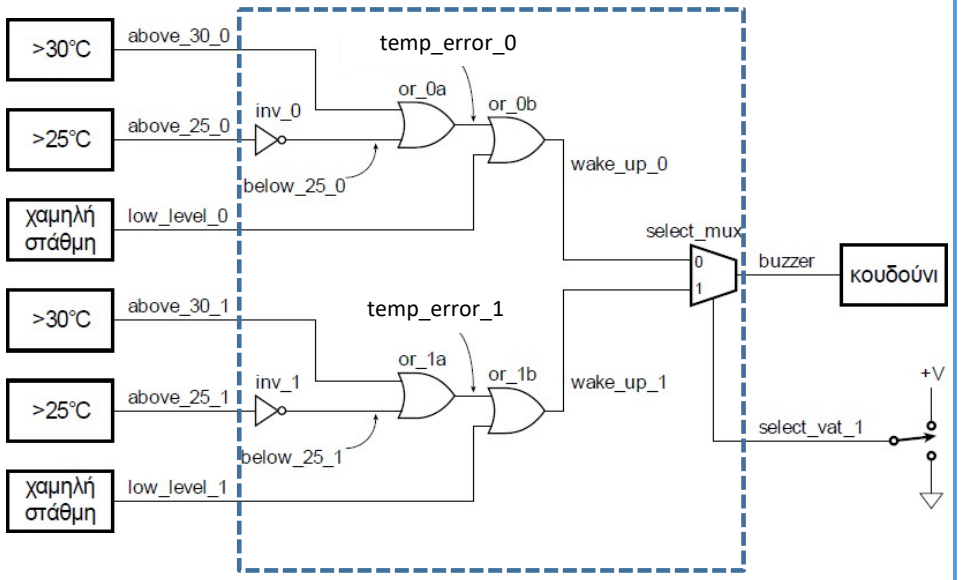
Αντικείμενο Μαθήματος

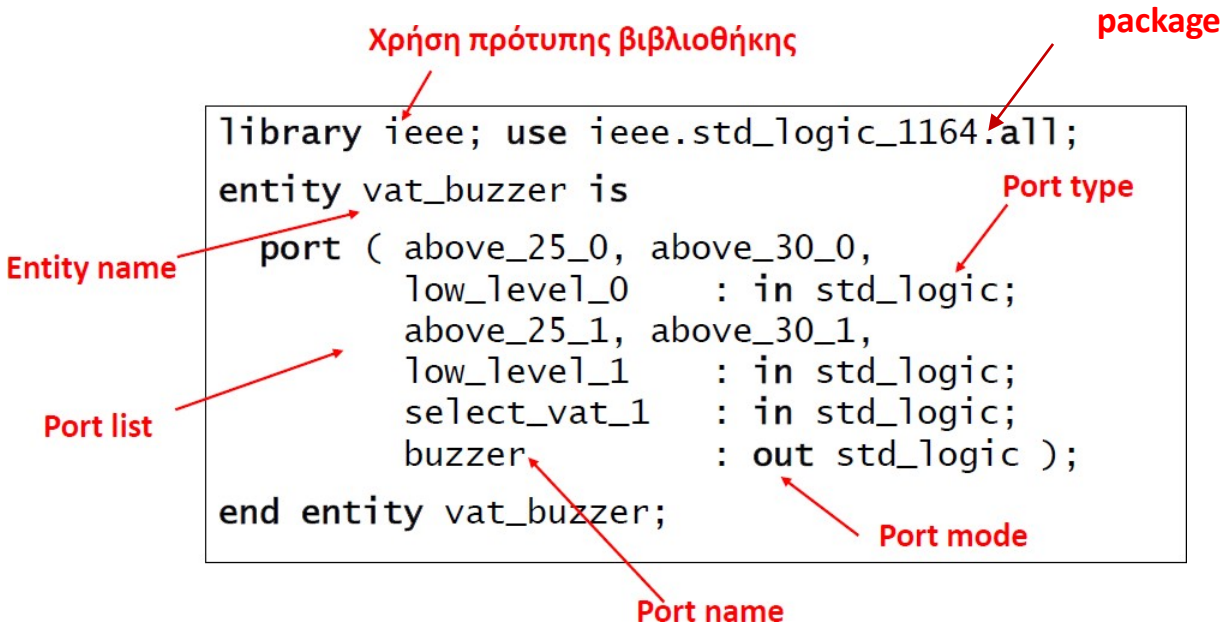
Προγραμματισμός σε VHDL



Ψηφιακά Συστήματα

Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – Entity: Input/Output





Αντικείμενο Μαθήματος

Εργαστήριο Σχεδίασης Ψηφιακών Συστημάτων – Οργάνωση μαθήματος 1/4

- Διδάσκων Εργαστηρίου: Ε.ΔΙ.Π. Διονύσης Βασιλόπουλος (denis@di.uoa.gr) – Γραφείο: Α33
  - Ώρες γραφείου: Τρίτη 15:00-16:00 και Πέμπτη 13:00 – 15:00, κατόπιν επικοινωνίας
- 3 διαλέξεις εισαγωγικές και 5 σετ Θεωρίας/εργαστηρίου (θεωρία/παράδοση-χρήση εργαλείου Vivado, επίλυση άσκησης-προγραμματισμός κάρτας) (αρχικός προγραμματισμός μαθήματος).
- Σύνολο 8 διαλέξεις στην Αίθουσα Β και 5 μαθήματα στο Εργαστήριο Ψηφιακής Σχεδίασης & ΗΥ Υψηλών Επιδόσεων (Αναγνωστήριο)

Αντικείμενο Μαθήματος

Εργαστήριο Σχεδίασης Ψηφιακών Συστημάτων – Οργάνωση μαθήματος 2/4

Ο τελικός **βαθμός** προκύπτει από:

- Εργαστηριακές ασκήσεις: μέγιστο βαθμού 7.
  - Προπτυχιακοί φοιτητές: Περιλαμβάνει 3 εργασίες (συνδυαστικά, ακολουθιακά και υλοποίηση cpu 4 εντολών).
  - Μεταπτυχιακοί φοιτητές: Περιλαμβάνει 3 εργασίες (συνδυαστικά, ακολουθιακά και υλοποίηση πλήρης cpu ενός κύκλου.

  Οι πρώτες δύο ασκήσεις (συνδυαστικά, ακολουθιακά) είναι κοινές.
- Γραπτές εξετάσεις: μέγιστο βαθμού 3.
  - Εξέταση μέσω eclass. Ερωτήσεις πάνω στην άσκηση αλλά και τη θεωρία.
- Εάν ο βαθμός των ασκήσεων έχει διαφορά πάνω από 30% σε σχέση με το βαθμό των γραπτών εξετάσεων τότε ισχύει ΜΟΝΟ ο βαθμός των εξετάσεων.

Αντικείμενο Μαθήματος

Εργαστήριο Σχεδίασης Ψηφιακών Συστημάτων – Οργάνωση μαθήματος 3/4

https://eclass.uoa.gr/ (eclass)

- Ανέβασμα διαφανειών/υλικού και ασκήσεων
- Επικοινωνία μέσω eclass , email
- Παράδοση εργασιών
- Εργαλείο λογισμικού για το μάθημα: VIVADO
  - Στο eclass θα ανέβει αρχείο που περιγράφει τον τρόπο με τον οποίο θα εγκαταστήσουμε το εργαλείο στον υπολογιστή μας (απαραίτητο).
  - Υπάρχει έκδοση για Windows (**2022.2**) και Linux (όχι για Mac)
    - Η έκδοση του Vivado πρέπει **<u>οπωσδήποτε</u>** να είναι η 2022.2

https://delos.uoa.gr/opendelos/

- Καταγραφή μαθημάτων, διαθέσιμο υλικό και στο eclass το Σ/Κ.

Αντικείμενο Μαθήματος

Εργαστήριο Σχεδίασης Ψηφιακών Συστημάτων – Οργάνωση μαθήματος 4/4

Α.Παρουσιάσεις (**ΒΑΣΙΚΟ ΕΓΧΕΙΡΙΔΙΟ**), κώδικας και άλλα κείμενα που ανεβαίνουν στο eclass

Β. Προτεινόμενη Βιβλιογραφία

1. **Ψηφιακή Σχεδίαση και Αρχιτεκτονική Υπολογιστών**, S.L.Harris, D.M.Harris, Εκδόσεις ARM®, 2019
2. **Ψηφιακή Σχεδίαση. Ενσωματωμένα Συστήματα με VHDL**, Peter J. Ashenden, Επιστημονική Επιμέλεια – Μετάφραση: Μ. Ψαράκης, Ν. Κρανίτης, Δ. Γκιζόπουλος, Εκδόσεις Νέων Τεχνολογιών, 2010
3. **Σχεδίαση Ψηφιακών Συστημάτων με τη Γλώσσα VHDL,** S. Brown, Z.Vranesic, Εκδόσεις Τζιόλα, 2024

Ψηφιακά Συστήματα

Ψηφιακή Σχεδίαση

- **Ψηφιακή** (*Digital):* κυκλώματα που χρησιμοποιούν δύο επίπεδα τάσης (π.χ. 0V, +5V) για αναπαράσταση της πληροφορίας
  - –Λογική *(Logic):* χρήση τιμών αληθείας (0/1, true/false) και κανόνων λογικής (άλγεβρα boole) για την ανάλυση των κυκλωμάτων
- **Σχεδίαση** *(Design):* ανταποκρίνονται σε λειτουργικές απαιτήσεις ενώ ταυτόχρονα ικανοποιούν περιορισμούς
  - – Περιορισμοί: απόδοση, μέγεθος (κόστος), ισχύς, κλπ.

Ψηφιακά Συστήματα

Ψηφιακή Σχεδίαση

- **Δυαδικό Σύστημα Αρίθμησης** (0,1)
- **Λογικοί κανόνες** (Άλγεβρα Boole: ΝΑΙ/ΟΧΙ, ΙΣΧΥΕΙ/ΔΕΝ ΙΣΧΥΕΙ)
- **Πύλες** (Στοιχειώδη κυκλώματα, υλοποιούν την άλγεβρα boole σε επίπεδο υλικού, κάνουμε στοιχειώδεις πράξεις στο δυαδικό σύστημα. Στη βάση τους υπάρχει το Περνάει Ρεύμα/Δεν περνάει ρεύμα ή Υπάρχει Τάση/Γείωση)
- **Συνδυαστικά κυκλώματα** (π.χ. ένα κύκλωμα που προσθέτει δύο δυαδικούς αριθμούς). Έχει εισόδους που παράγουν κάποιες τιμές εξόδων.
- **Ακολουθιακά κυκλώματα**. Η τιμή της εξόδου εξαρτάται από τις εισόδους αλλά και από την προηγούμενη τιμή της εξόδου. Συνήθως υπάρχει συγχρονισμός στο πότε αλλάζουν οι τιμές στην είσοδο και στο πότε ελέγχουμε την τιμή στην έξοδο. Συνήθως υπάρχει αποθήκευση δεδομένων σε Flip/Flop. Παράδειγμα είναι το χρονόμετρο.

Ψηφιακά Συστήματα

Ψηφιακή Σχεδίαση – Είδη κυκλωμάτων

- **Συνδυαστικά** κυκλώματα: Οι τιμές των εξόδων εξαρτώνται μόνο από τις τιμές των εισόδων του συστήματος
- **Ακολουθιακά** κυκλώματα: Οι τιμές των εξόδων εξαρτώνται τόσο από τις τιμές των εισόδων του συστήματος όσο και από τις προηγούμενες τιμές των εξόδων (έχουμε ανάδραση).
  - **Σύγχρονα**: Η συμπεριφορά τους ορίζεται από τις τιμές των εξόδων σε διακριτές στιγμές του χρόνου. Υπάρχει σήμα συγχρονισμού (ρολόι/clock-CLK)
  - **Ασύγχρονα**: Οι τιμές των εξόδων αλλάζουν ανά πάσα στιγμή.

Ψηφιακά Συστήματα

Ψηφιακό κύκλωμα – Φυσική υλοποίηση

- Τεχνολογίες υλοποίησης
  - Application-specific ICs (ASICs): Ολοκληρωμένα κυκλώματα εξειδικευμένα για εφαρμογές (δεν προγραμματίζονται)
  - Field-programmable gate arrays (FPGAs): Επιτόπου προγραμματιζόμενοι πίνακες πυλών
- Αντιστοίχιση(mapping): καθορίζει τους πόρους για κάθε υποσύστημα
- Τοποθέτηση (placement): διευθετεί τις πύλες μέσα στα υποσυστήματα
- Δρομολόγηση (routing): ενώνει τις πύλες με αγωγούς
- Φυσική επαλήθευση (physical verification)
  - Το φυσικό κύκλωμα συνεχίζει να ικανοποιεί τους περιορισμούς
  - Χρησιμοποιεί καλύτερες εκτιμήσεις των καθυστερήσεων

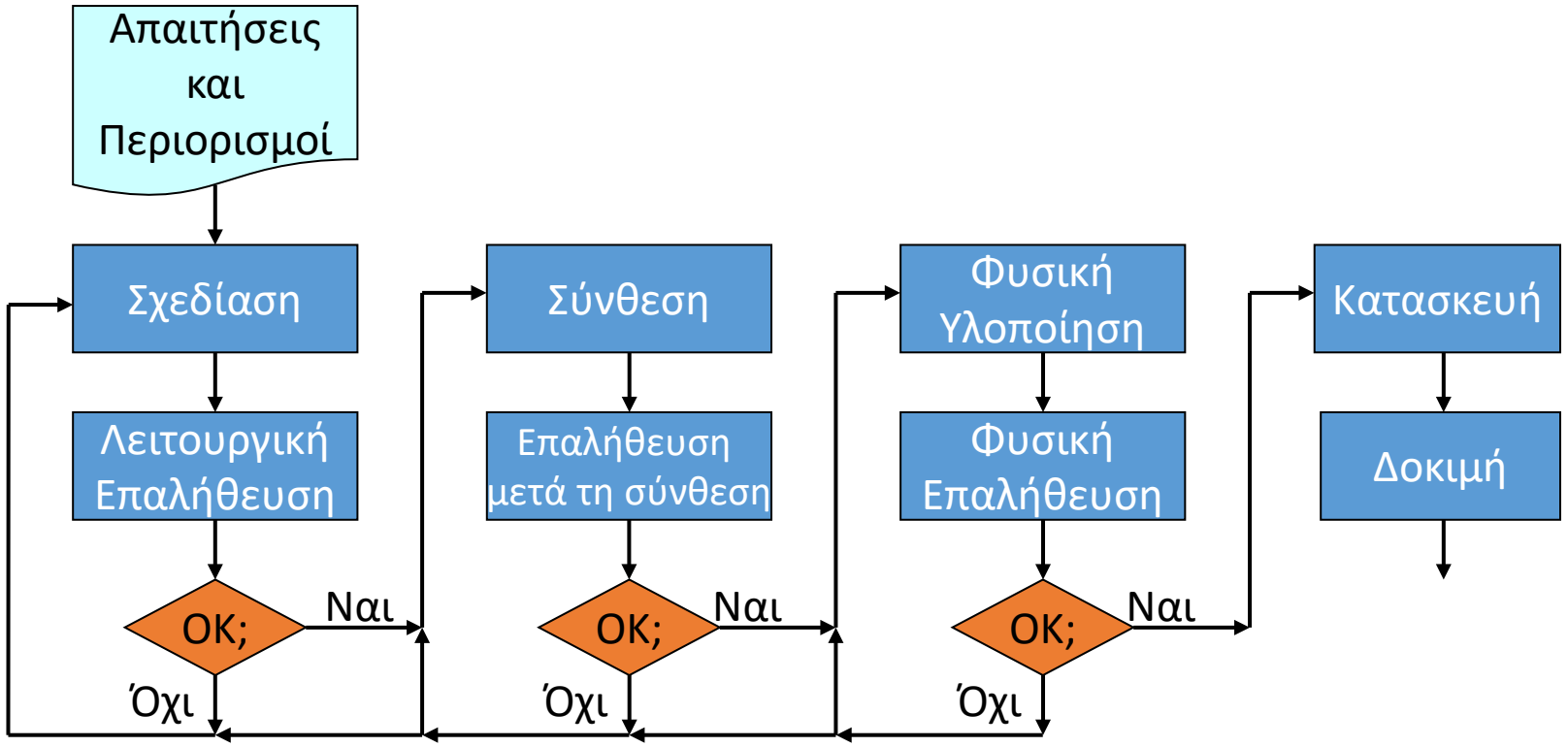
Ψηφιακά Συστήματα

Ψηφιακή Σχεδίαση – Μεθοδολογία Σχεδίασης

- Τα απλά συστήματα μπορούν να σχεδιαστούν από ένα άτομο χρησιμοποιώντας ειδικές μεθόδους
- Τα πραγματικά συστήματα σχεδιάζονται από ομάδες
  - Απαιτούν μια συστηματική μεθοδολογία σχεδίασης
- Καθορίζει
  - Τις εργασίες που αναλαμβάνουμε
  - Την πληροφορία που απαιτείται και παράγεται
  - Τις σχέσεις μεταξύ των εργασιών
    - εξαρτήσεις, καθορισμός των ακολουθιών
  - Τα εργαλεία EDA (Electronic design automation) που χρησιμοποιούνται (αλλιώς και eCAD – electronic Computer Aided Design)

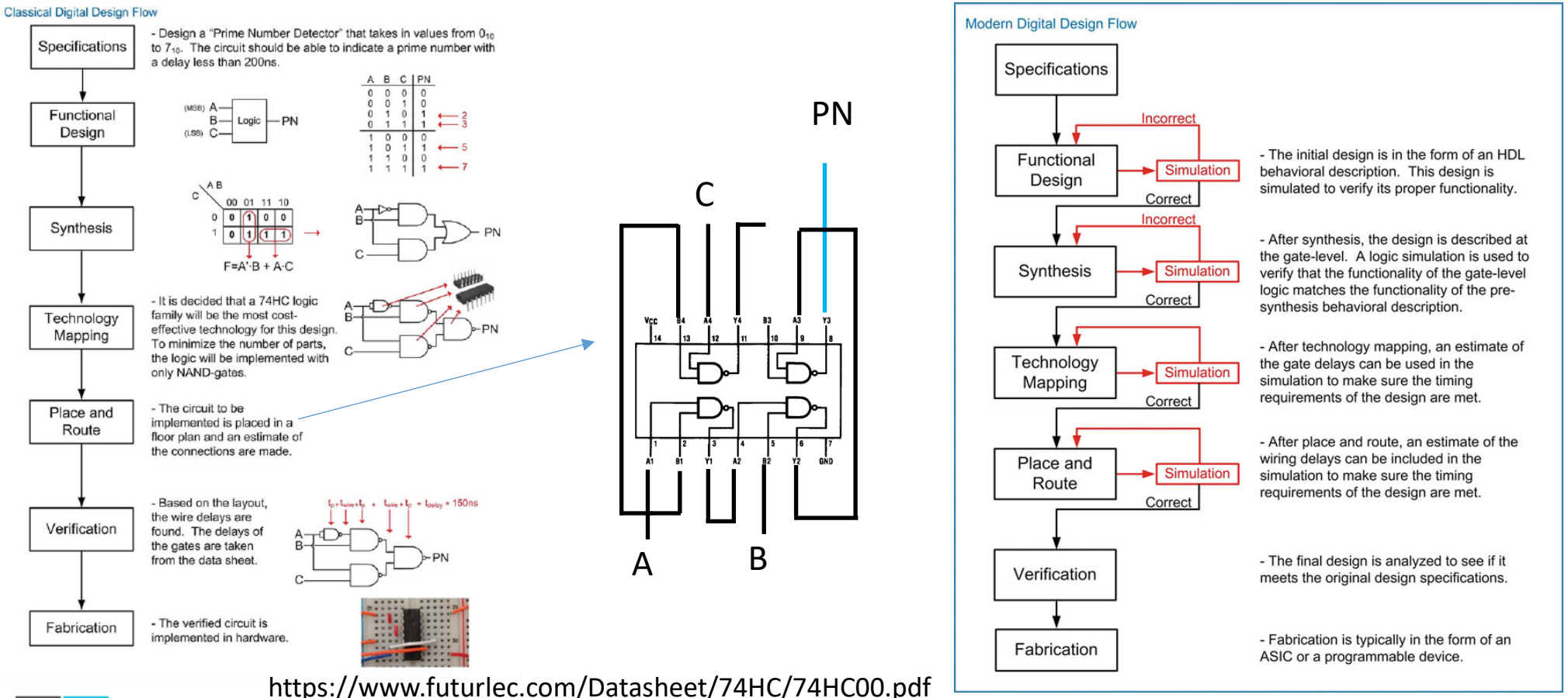
Ψηφιακά Συστήματα

Ψηφιακή Σχεδίαση – Μεθοδολογία ανάπτυξης



Ψηφιακά Συστήματα

Ψηφιακή Σχεδίαση – Μεθοδολογία ανάπτυξης



https://www.futurlec.com/Datasheet/74HC/74HC00.pdf

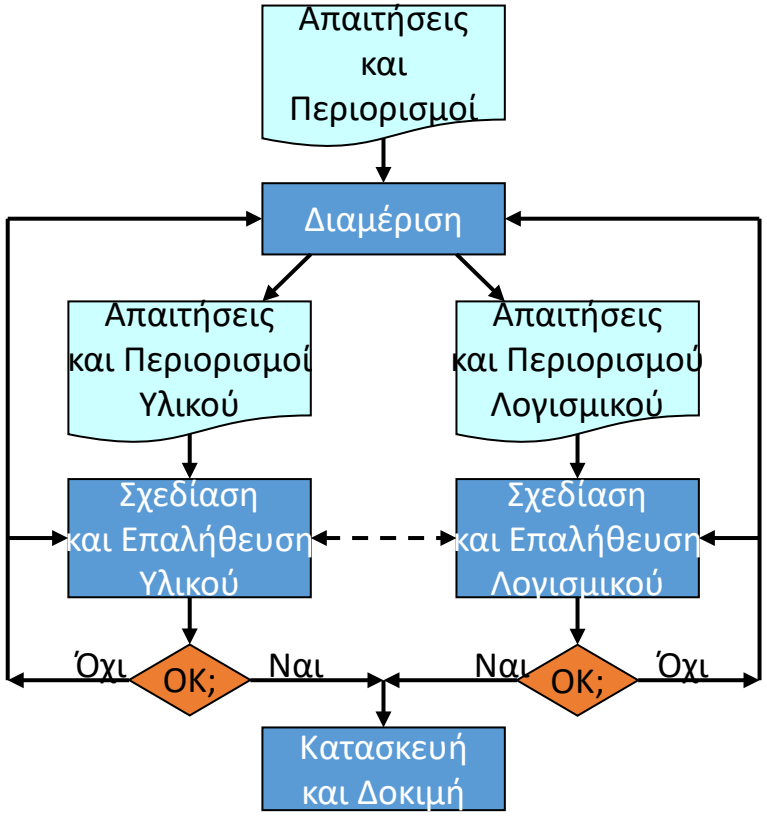
Ψηφιακά Συστήματα

Ψηφιακή Σχεδίαση – Ιεραρχική σχεδίαση

- Τα κυκλώματα είναι αρκετά πολύπλοκα για να σχεδιάσουμε όλες τις λεπτομέρειες με τη μία
- Σχεδιάζουμε υποσυστήματα για απλές λειτουργίες
- Συνθέτουμε το σύστημα από τα υποσυστήματα
  - Αντιμετωπίζουμε τα υποκυκλώματα ως «μαύρα κουτιά»
  - Επαληθεύουμε ανεξάρτητα, και έπειτα επαληθεύουμε τη σύνθεση
- Σχεδίαση top-down (από πάνω προς τα κάτω) ή bottom-up (από κάτω προς τα πάνω)

Ψηφιακά Συστήματα

Ψηφιακή Σχεδίαση – Μεθοδολογία Συσχεδίασης



Κάθε ομάδα υλοποιεί ένα ξεχωριστό Module του συστήματος

Ψηφιακά Συστήματα

Δυαδική Αναπαράσταση

- Δεκαδικό σύστημα αναπαράστασης αριθμών (0-9)

Στήλη των 1000 Στήλη των 100 Στήλη των 10 Στήλη των 1

6598_{10}

Κάθε στήλη ενός δεκαδικού αριθμού έχει δεκαπλάσιο βάρος από την προηγούμενη στήλη. Ξεκινώντας από τα δεξιά προς τα αριστερά τα βάρη είναι: 10^0, 10^1, 10^2, 10^3, ...

- Δυαδικό σύστημα αναπαράστασης αριθμών (0-1)

Στήλη των 8 Στήλη των 4 Στήλη των 2 Στήλη των 1

1101_2

Κάθε στήλη ενός δυαδικού αριθμού έχει διπλάσιο βάρος από την προηγούμενη στήλη. Ξεκινώντας από τα δεξιά προς τα αριστερά τα βάρη είναι: 2^0, 2^1, 2^2, 2^3, ...

Ψηφιακά Συστήματα

Δυαδική Αναπαράσταση

- Δεκαδικό σύστημα αναπαράστασης αριθμών (0-9)

Στήλη των 1000 Στήλη των 100 Στήλη των 10 Στήλη των 1

$$6598_{10} = 6 \times 10^3 + 5 \times 10^2 + 9 \times 10^1 + 8 \times 10^0$$

Έξι Χιλιάδες — Πέντε Εκατοντάδες — Εννέα Δεκάδες — Οκτώ Μονάδες

- Δυαδικό σύστημα αναπαράστασης αριθμών (0-1)

Στήλη των 8 Στήλη των 4 Στήλη των 2 Στήλη των 1

$$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{10}$$

Μια Οκτάδα — Μια Τετράδα — Μηδέν Δυάδες — Μια Μονάδα

Δυαδικό Σύστημα

Αριθμοί χωρίς πρόσημο (Unsigned – Μόνο θετικοί)

Πρόσθεση

```
 0    0    1    1
+0   +1   +0   +1
---  ---  ---  ---
 0    1    1   10
```

Κρατούμενο/(C)arry)

```
 0    0    0    1
```

```
 00   01   11   11
 01  +01  +01  +11
----  ----  -----  -----
 01   10  100  110
```

Κρατούμενο/(C)arry)

```
 0    0    1    1
```

Δυαδικό Σύστημα

Αριθμοί χωρίς πρόσημο (Unsigned – Μόνο θετικοί)

Πολλαπλασιασμός/Διαίρεση με πολλαπλάσια του δύο

10 =2(δεκαδικό) (A)
Εφαρμόζουμε αριστερή ολίσθηση και γεμίζουμε δεξιά με το 0
10**0** =4(δεκαδικό) (B) = 2*(A)
10**00** =8(δεκαδικό) (C) = 2*(B)=4*(A)

1000 =8(δεκαδικό) (A)
Εφαρμόζουμε δεξιά ολίσθηση διαγράφοντας το πιο δεξί ψηφίο
100 =4(δεκαδικό) (B) = (A)/2
10 =2(δεκαδικό) (C)= (B)/2=(A)/4

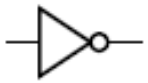
Ψηφιακά Συστήματα

Δυαδική Αναπαράσταση

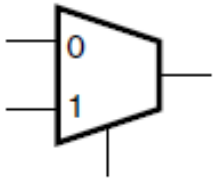
- Συστήματα που αναπαριστούν την πληροφορία με δύο τιμές (0,1 ή True,False)
- Βασικές ψηφιακές λογικές πύλες και πίνακες αληθείας

AND gate

OR gate



inverter



multiplexer

x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

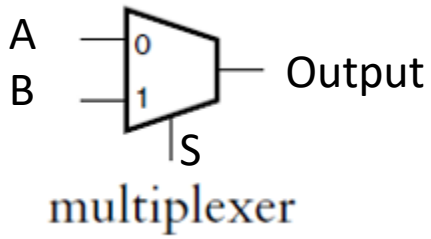
x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

x	$\overline{x}$
0	1
1	0

Ψηφιακά Συστήματα

Δυαδική Αναπαράσταση

- Σε κάθε Πύλη ή Κύκλωμα αντιστοιχεί ένας Πίνακας Αληθείας (Truth Table).
- Ο Πίνακας Αληθείας καθορίζει την τιμή εξόδου για κάθε συνδυασμό εισόδων στην Πύλη ή το Κύκλωμα

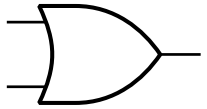
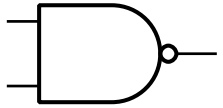
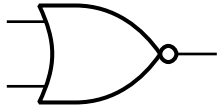
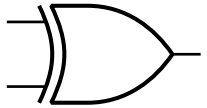
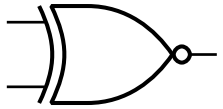
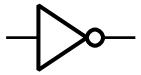


S	Output
0	A
1	B

Αν A=0, B=1, S=1, Output= ;
Αν A=1, B=0, S=1, Output= ;

VHDL

Λογικοί Τελεστές

a **and** b	$a \cdot b$	
a **or** b	$a + b$	
a **nand** b	$\overline{a \cdot b}$	
a **nor** b	$\overline{a + b}$	
a **xor** b	$a \oplus b$	
a **xnor** b	$\overline{a \oplus b}$	
not a	$\overline{a}$	

- Προτεραιότητα
  - το **not** έχει την υψηλότερη
  - οι υπόλοιποι τελεστές έχουν ίση προτεραιότητα
  - από αριστερά προς τα δεξιά
  - χρησιμοποιούμε παρενθέσεις για να διακρίνουμε τη σειρά υπολογισμού
- Τιμές bit στην VHDL
  - '0' και '1'

HDL

- Hardware Description Language (HDL)
  - Μια γλώσσα για την μοντελοποίηση της συμπεριφοράς και της δομής των ψηφιακών συστημάτων
- Electronic Design Automation (EDA) using HDL
  - Σχεδίαση ηλεκτρονικών κυκλωμάτων με χρήση εργαλείων CAD (computer-aided design)
  - Εισαγωγή σχεδίασης (design entry)
    - Χρήση κώδικα αντί για σχηματικά διαγράμματα
  - Επαλήθευση(verification)
    - Προσομοίωση (simulation) του κώδικα
  - Σύνθεση (synthesis)
    - Αυτόματη παραγωγή των κυκλωμάτων
  - Φυσική υλοποίηση (implementation)
    - Υλοποίηση του κυκλώματος στην τεχνολογία επιλογής

HDL

- Hardware Description Language (HDL)
  - Μια γλώσσα για την μοντελοποίηση της συμπεριφοράς και της δομής των ψηφιακών συστημάτων
- Electronic Design Automation (EDA) using HDL
  - Σχεδίαση ηλεκτρονικών κυκλωμάτων με χρήση εργαλείων CAD (computer-aided design)
  - Εισαγωγή σχεδίασης (design entry)
    - Χρήση κώδικα αντί για σχηματικά διαγράμματα
  - Επαλήθευση(verification)
    - Προσομοίωση (simulation) του κώδικα
  - Σύνθεση (synthesis)
    - Αυτόματη παραγωγή των κυκλωμάτων
  - Φυσική υλοποίηση (implementation)
    - Υλοποίηση του κυκλώματος στην τεχνολογία επιλογής

HDL

Πλεονεκτήματα των HDLs

- Υπερτερούν από τα σχηματικά διαγράμματα
  - Η μοντελοποίηση του συστήματος μπορεί να γίνει σε όλα τα επίπεδα (από τα υψηλότερα ως τα χαμηλότερα)
  - Η περιγραφή σε HDL είναι συνήθως πιο κατανοητή από ένα σχηματικό διάγραμμα
  - Η περιγραφή σε HDL είναι ανεξάρτητη από τις βιβλιοθήκες σχεδίασης (design libraries) και τα εργαλεία CAD
- Υπερτερούν από τις γλώσσες προγραμματισμού
  - Παρέχουν δομές που περιγράφουν καλύτερα το υλικό
  - Παράλληλη εκτέλεση εντολών αντί για ακολουθιακή/σειριακή
  - Παρέχουν δυνατότητα για περιγραφή χρονισμών

HDL

Μοντελοποίηση και προσομοίωση

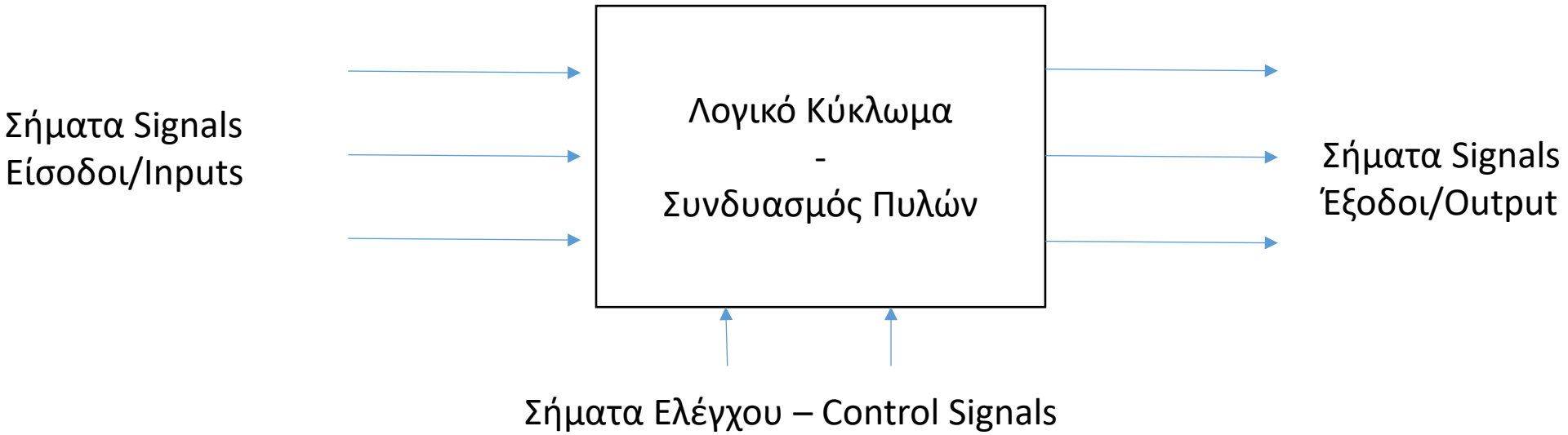
- Αρχικά οι HDLs σχεδιάστηκαν για τη μοντελοποίηση και τη προσομοίωση των συστημάτων υλικού στα υψηλότερα επίπεδα αφαίρεσης
- Χαρακτηριστικά μοντελοποίησης των HDLs:
  - παράλληλη εκτέλεση
  - ιεραρχική σχεδίαση
  - περιγραφή χρονισμών
  - περιγραφή ακολουθίας γεγονότων
  - περιγραφή σύγχρονης/ασύγχρονης συμπεριφοράς

VHDL - VIVADO

Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – Σύνθεση

- Συνήθως σχεδιάζουμε χρησιμοποιώντας HDL επιπέδου μεταφοράς καταχωρητή (Register Transfer Level - RTL)
  - υψηλότερο επίπεδο αφαίρεσης από τις πύλες
- Το εργαλείο σύνθεσης μεταφράζει το μοντέλο σε ένα κύκλωμα από πύλες που εκτελεί την ίδια λειτουργία
- Προσδιορίζουμε στο εργαλείο
  - την τεχνολογία υλοποίησης που στοχεύουμε
  - περιορισμούς στο χρονισμό, στην επιφάνεια, κλπ.
- Επαλήθευση μετά τη σύνθεση
  - το κύκλωμα που προέκυψε από τη σύνθεση ικανοποιεί τους περιορισμούς

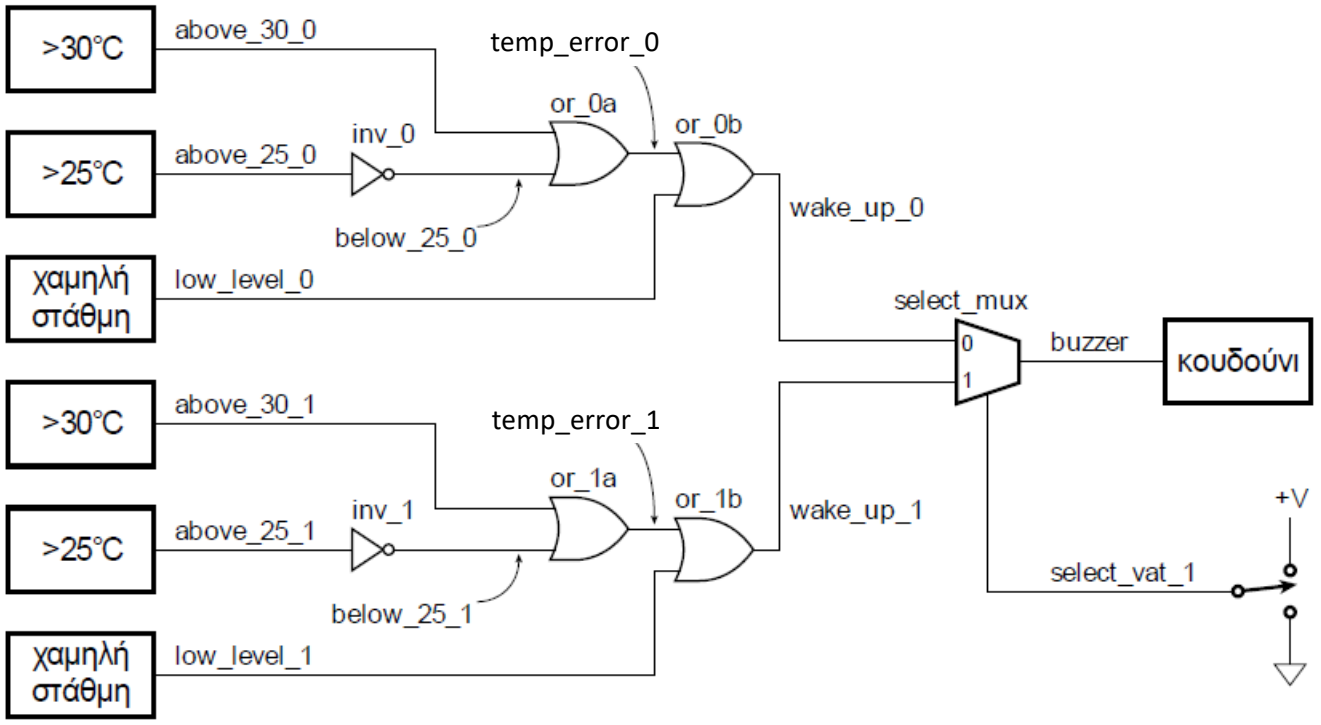
Ψηφιακά Συστήματα



Ψηφιακά Συστήματα

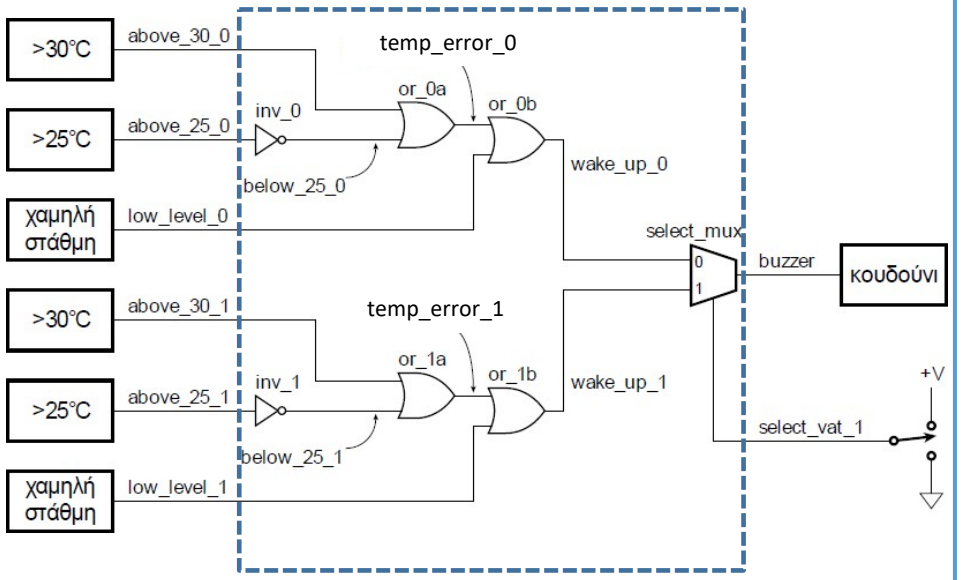
Ψηφιακό κύκλωμα

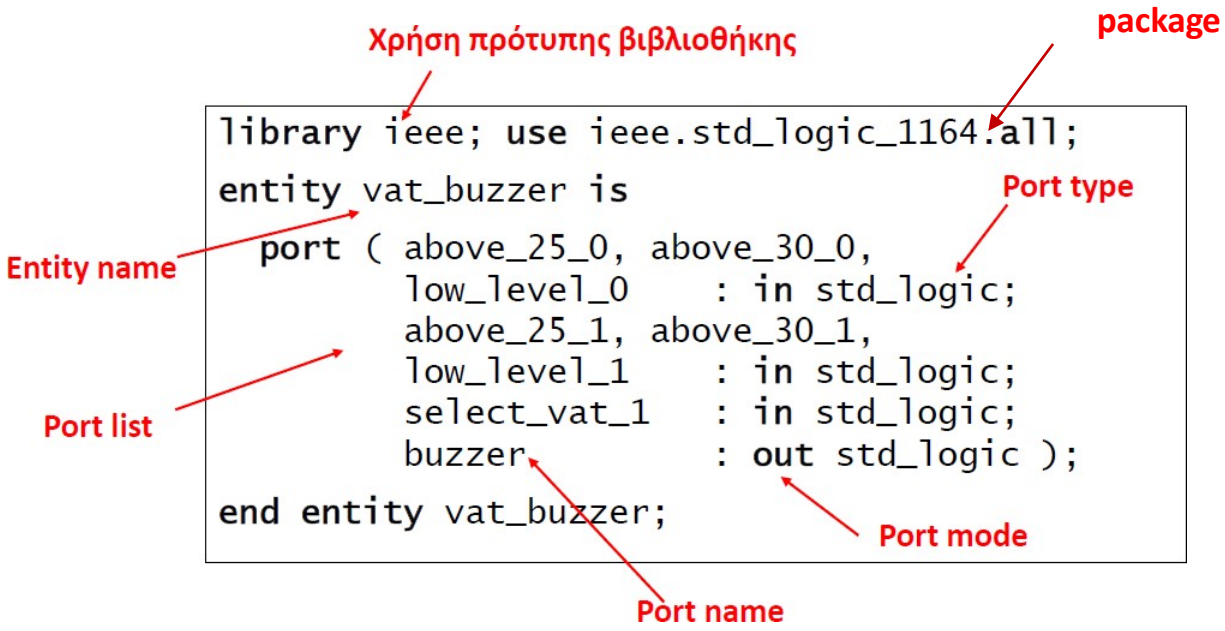
Παράδειγμα:
Εργοστάσιο με 2 δοχεία επεξεργασίας υγρών. Το κάθε δοχείο πρέπει α)να έχει θερμοκρασία μεταξύ 25 και 30 βαθμών και β) η στάθμη του πρέπει να είναι πάνω από ένα επίπεδο. Ο επόπτης επιλέγει ποιο από τα 2 δοχεία χρησιμοποιείται. Σε περίπτωση που το α) ή το β) δεν ικανοποιούνται πρέπει να ενεργοποιηθεί το κουδούνι.



Ψηφιακά Συστήματα

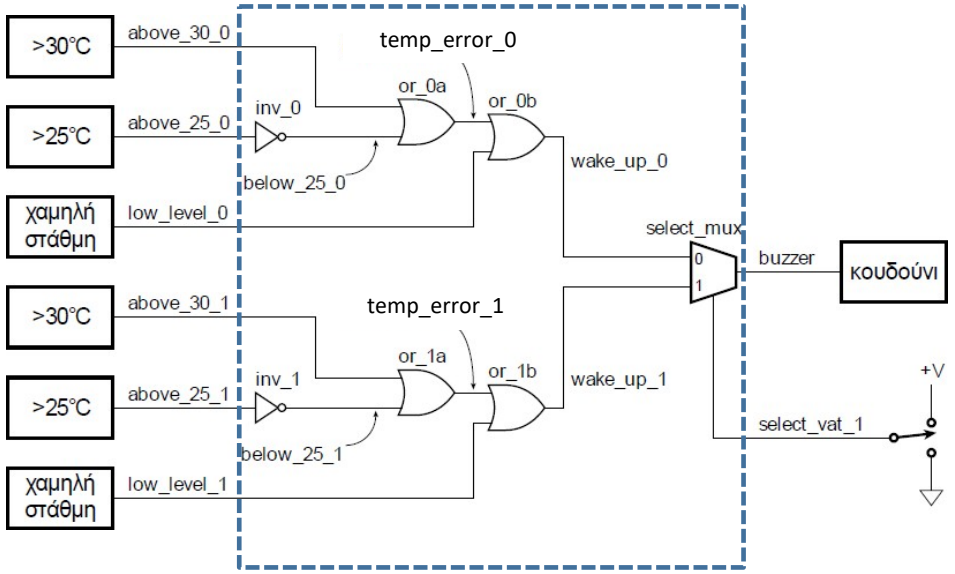
Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – Entity: Input/Output





Ψηφιακά Συστήματα

Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – Structural Architecture

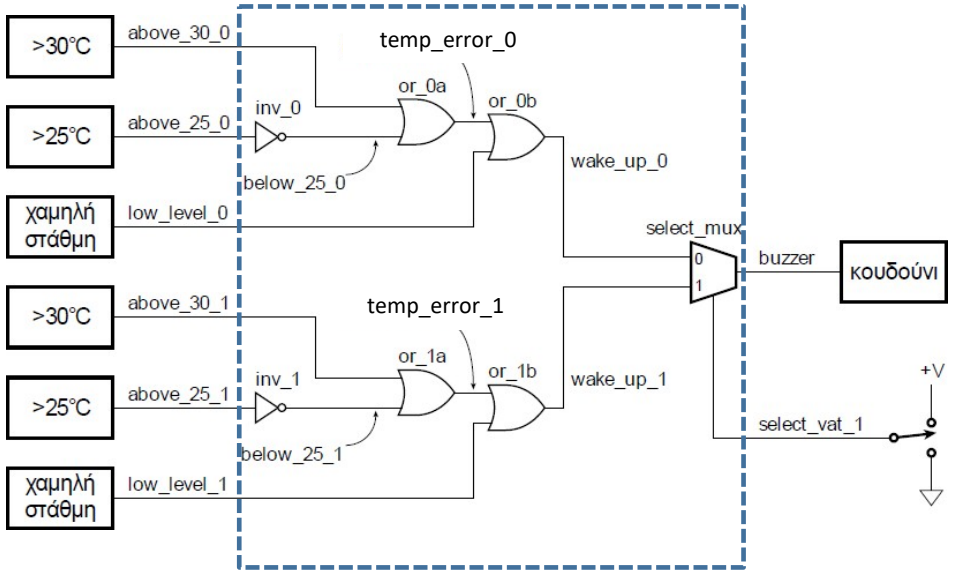


Περιγραφή δομικών στοιχείων

```
architecture struct of vat_buzzer is

        signal below_25_0, temp_error_0, wake_up_0 : std_logic;
        signal below_25_1, temp_error_1, wake_up_1 : std_logic;

begin
        -- components for vat 0
        inv_0 : inv port map (above_25_0, below_25_0);
        or_0a : or2 port map (above_30_0, below_25_0, temp_error_0);
        or_0b : or2 port map (temp_error_0, low_level_0, wake_up_0);

        -- components for vat 1
        inv_1 : inv port map (above_25_1, below_25_1);
        or_1a : or2 port map (above_30_1, below_25_1, temp_error_1);
        or_1b : or2 port map (temp_error_1, low_level_1, wake_up_1);

        select_mux : mux2 port map ( wake_up_0, wake_up_1, select_vat, buzzer);

end architecture struct;
```

Ψηφιακά Συστήματα

Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – Dataflow Architecture

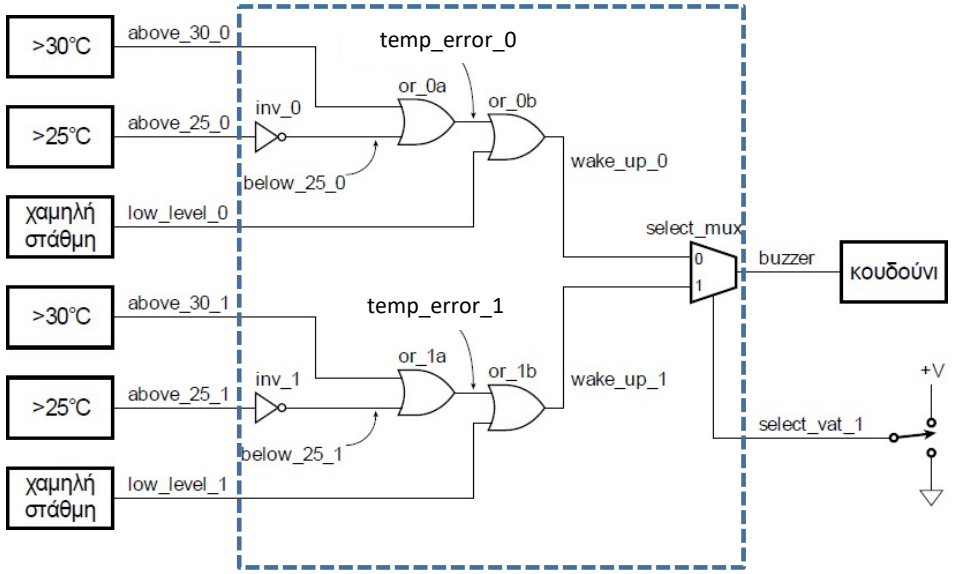



```
architecture Dataflow of vat_buzzer is

begin

        buzzer <=    low_level_1 or (above_30_1 or not above_25_1)
                     when select_vat='1'  else
                     low_level_0 or (above_30_0 or not above_25_0);

end architecture Dataflow;
```

Αρχιτεκτονική
Περιγραφή της λειτουργικότητας που προσφέρει το λογικό κύκλωμα

Ψηφιακά Συστήματα

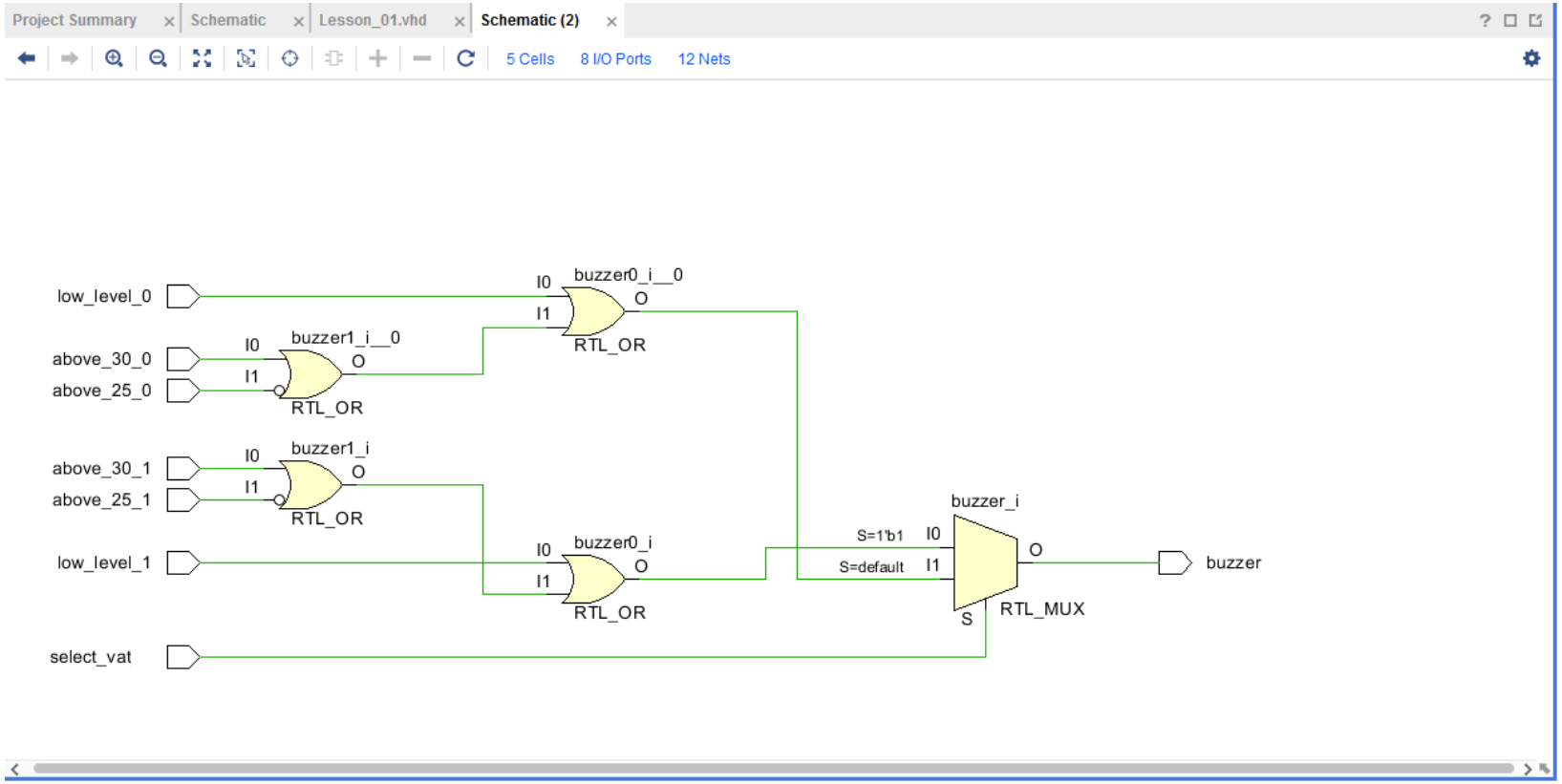
Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – Behavioral Architecture




```
architecture behavioral of vat_buzzer is

begin

process (low_level_0, above_30_0, above_25_0,
         low_level_1, above_30_1, above_25_1
         select_vat ) is
begin
buzzer <=     low_level_1 or (above_30_1 or not above_25_1)
                  when select_vat='1'   else
              low_level_0 or (above_30_0 or not above_25_0);
end process;
end architecture behavioral;
```

VHDL - VIVADO

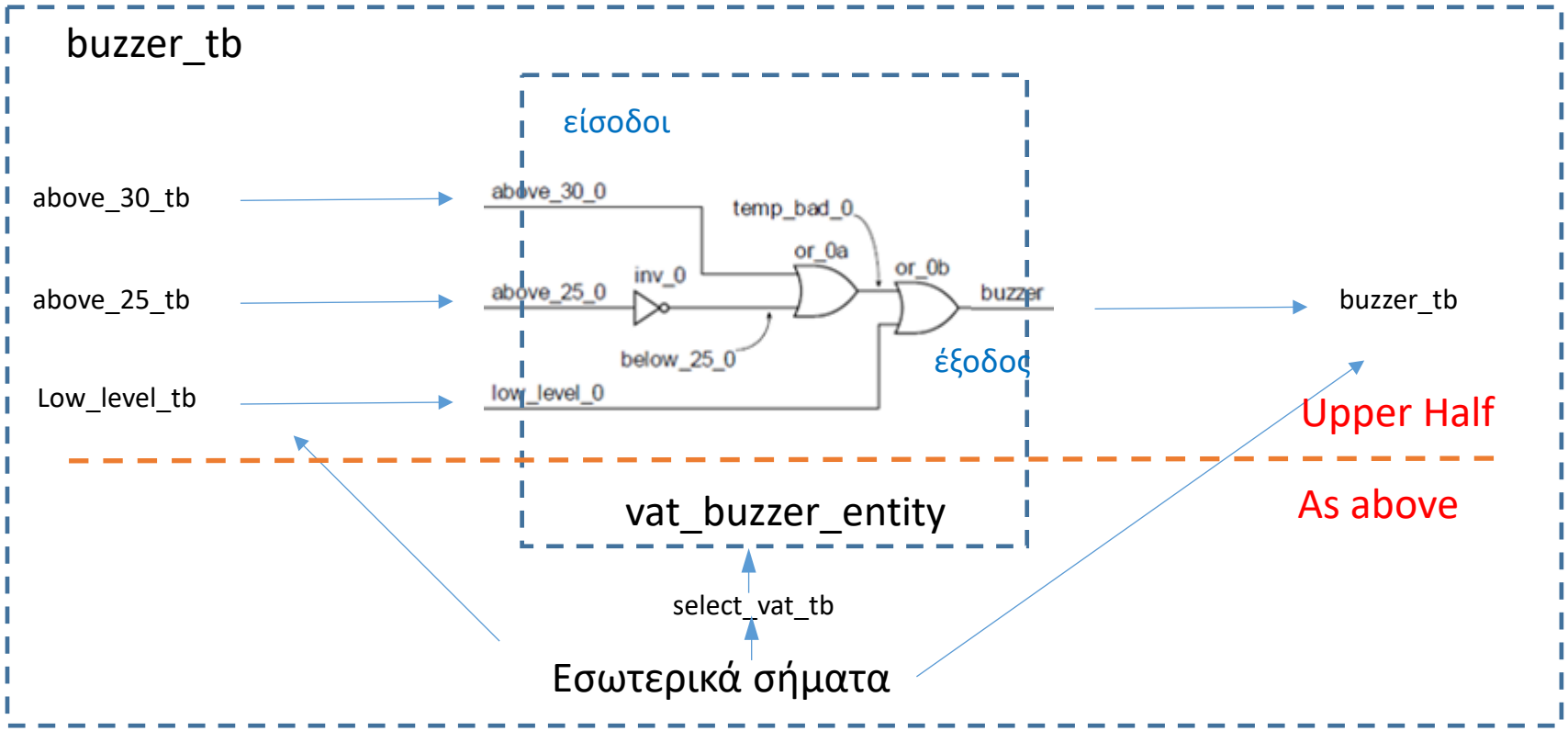
Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – Λογικές πύλες



VHDL - Vivado

Ψηφιακό κύκλωμα – VHDL: Προσομοίωση

! Προσοχή ότι δείχνουμε το μισό κύκλωμα της άσκησης



VHDL - VIVADO

Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – Προσομοίωση

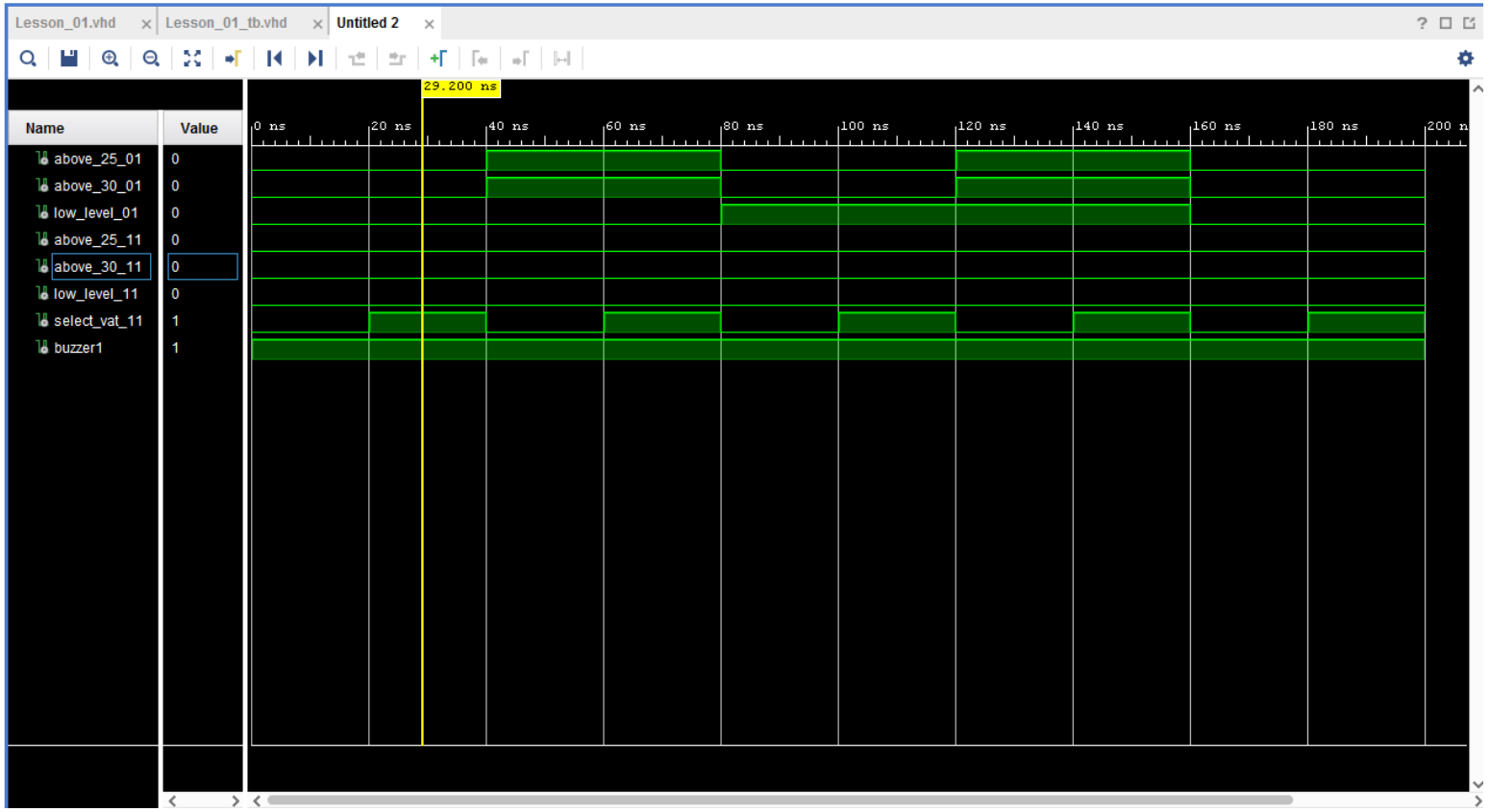
```
library IEEE;use IEEE.STD_LOGIC_1164.ALL;

entity buzzer_tb is
end entity buzzer_tb;

architecture Behavioral of buzzer_tb is

component vat_buzzer is
port (
above_25_0, above_30_0, low_level_0          : in std_logic;
above_25_1, above_30_1, low_level_1          : in std_logic;
select_vat                                   : in std_logic;
buzzer                                       : out std_logic;
);
end component vat_buzzer
signal  above_25_0_tb, above_30_0_tb, low_level_0_tb          :std_logic;
signal  above_25_1_tb, above_30_1_tb, low_level_1_tb          :std_logic;
signal  select_vat_tb                                         :std_logic;
signal  buzzer_tb                                             : std_logic;
begin
duv: vat_buzzer port map (above_25_0 => above_25_0_tb, above_25_1 =>
above_25_1_tb, above_30_0 => above_30_0_tb, above_30_1 => above_30_1_tb,
low_level_0 => low_level_0_tb, low_level_1 => low_level_1_tb, select_vat =>
select_vat_tb, buzzer => buzzer_tb);
```

```
apply_test_cases: process is
begin
   above_25_0_tb <='0'; above_25_1_tb <='0'; above_30_0_tb <='0'; above_30_1_tb
<='0'; low_level_0_tb <='0'; low_level_1_tb <='0';select_vat_tb<='0'; wait for 20 ns;
 above_25_0_tb <='1'; above_25_1_tb <='0'; above_30_0_tb <='1'; above_30_1_tb
<='0'; low_level_0_tb <='0'; low_level_1_tb <='0';select_vat_tb<='0'; wait for 20 ns;
above_25_0_tb <='0'; above_25_1_tb <='0'; above_30_0_tb <='0'; above_30_1_tb
<='0'; low_level_0_tb <='1'; low_level_1_tb <='1';select_vat_tb<='0'; wait for 20 ns;
above_25_0_tb <='0'; above_25_1_tb <='0'; above_30_0_tb <='0'; above_30_1_tb
<='0'; low_level_0_tb <='0'; low_level_1_tb <='1';select_vat_tb<='1'; wait for 20 ns;
above_25_0_tb <='0'; above_25_1_tb <='0'; above_30_0_tb <='1'; above_30_1_tb
<='0'; low_level_0_tb <='0'; low_level_1_tb <='0';select_vat_tb<='1'; wait for 20 ns;
above_25_0_tb <='0'; above_25_1_tb <='1'; above_30_0_tb <='0'; above_30_1_tb
<='0'; low_level_0_tb <='1'; low_level_1_tb <='0';select_vat_tb<='0'; wait for 20 ns;
above_25_0_tb <='0'; above_25_1_tb <='1'; above_30_0_tb <='1'; above_30_1_tb
<='1'; low_level_0_tb <='1'; low_level_1_tb <='0';select_vat_tb<='0'; wait for 20 ns;
above_25_0_tb <='0'; above_25_1_tb <='0'; above_30_0_tb <='0'; above_30_1_tb
<='1'; low_level_0_tb <='1'; low_level_1_tb <='1';select_vat_tb<='0'; wait for 20 ns;
end process apply_test_cases;
end architecture Behavioral ;
```

Συνήθως ενδεικτικές τιμές. Όχι όλος ο πίνακας αληθείας. Υπάρχει και πιο δόκιμος τρόπος

VHDL - VIVADO

Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – Προσομοίωση Χρονική



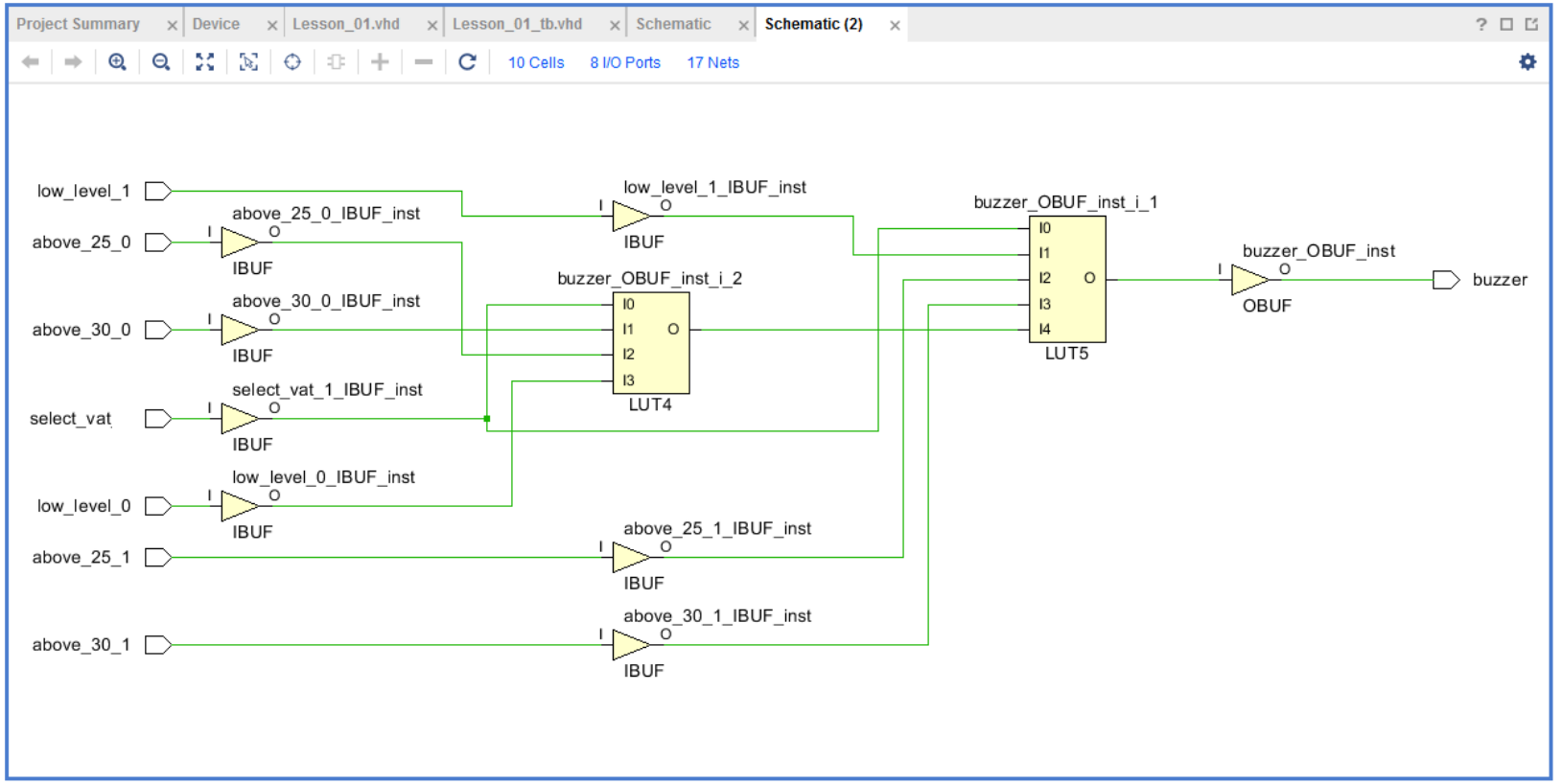
VHDL - VIVADO

Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – Σύνθεση

- Το εργαλείο σύνθεσης μεταφράζει το μοντέλο σε ένα κύκλωμα από ψηφιακά στοιχεία (πύλες, πολυπλέκτες, καταχωρητές, LUT) που εκτελεί την ίδια λειτουργία
- Προσδιορίζουμε στο εργαλείο
  - την τεχνολογία υλοποίησης που στοχεύουμε
  - περιορισμούς στο χρονισμό, στην επιφάνεια, κλπ.
- Επαλήθευση μετά τη σύνθεση
  - το κύκλωμα που προέκυψε από τη σύνθεση ικανοποιεί τους περιορισμούς

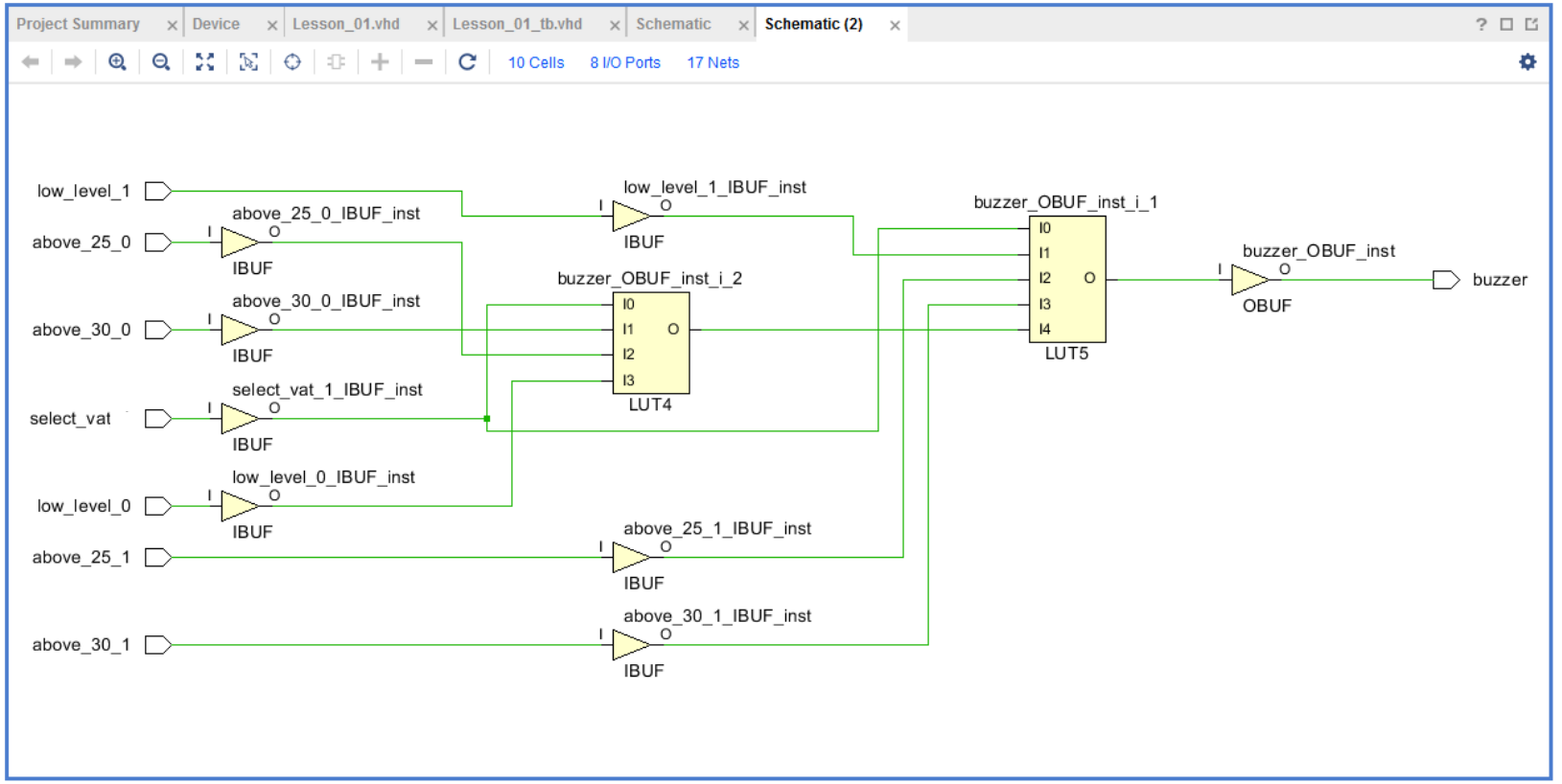
VHDL - VIVADO

Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – Σύνθεση



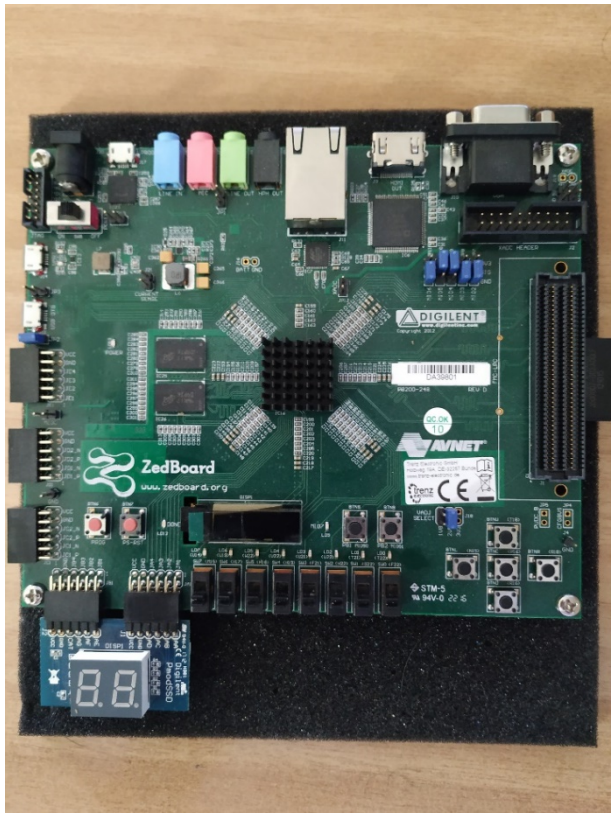
VHDL - VIVADO

Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – Υλοποίηση



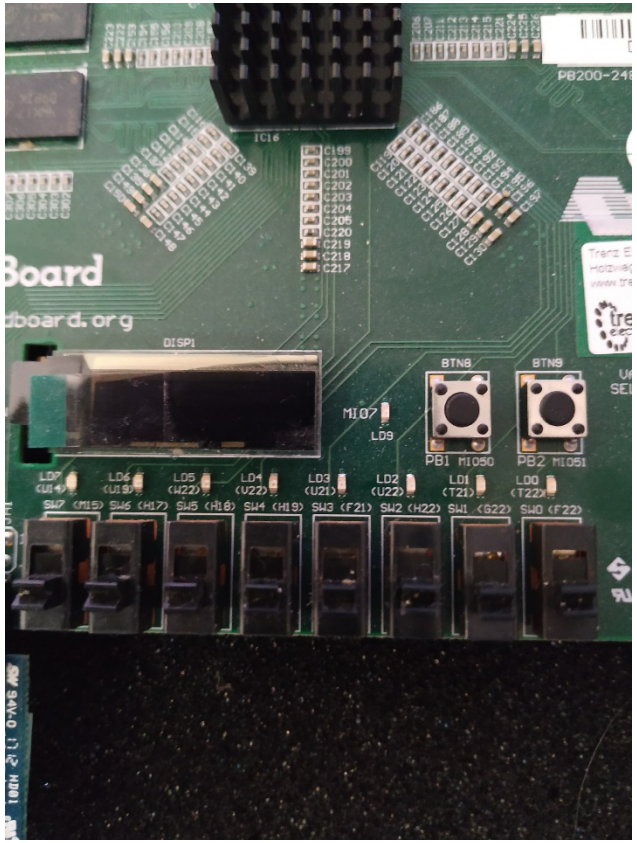
VHDL - VIVADO

Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – FPGA



VHDL - VIVADO

Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – FPGA



Παροχές ΕΚΠΑ

- https://delos365.grnet.gr/
  - Microsoft office (Word, Excel, Powerpoint,...) σε online και σε desktop έκδοση. Έχετε χώρο και στο onedrive (1TB). Σύνδεση απλά με τα ακαδημαϊκά σας credentials.
- google.com (ΠΡΟΣΟΧΗ: σύνδεση ως sdixxxxxxx@uoa.gr και μετά θα σας ζητηθούν τα ακαδημαϊκά σας credentials)
  - Εφαρμογές google + 50GB (google drive)
- https://azureforeducation.microsoft.com/devtools
  - Windows 10, 11 και εργαλεία ανάπτυξης λογισμικού της Microsoft (ΠΡΟΣΟΧΗ: σύνδεση ως sdixxxxxxx@o365.uoa.gr και μετά θα σας ζητηθούν τα ακαδημαϊκά σας credentials)

Προσοχή! Δεν είναι βέβαιο ότι θα ισχύουν για όλα τα account και ότι είναι ενεργή η συνδρομή του ΕΚΠΑ για αυτές τις υπηρεσίες

Περίληψη

- Εισαγωγή στο «Εργαστήριο Σχεδίαση Ψηφιακών Συστημάτων - VHDL»
- Μεθοδολογία σχεδίασης
- Εισαγωγή στο δυαδικό σύστημα αρίθμησης και στις λογικές πύλες
- Εισαγωγική παρουσίαση VHDL και Vivado
- Διαβάζετε το κεφάλαιο 1 (εκτός του 1.3) και τις παραγράφους 2.1 και 3.1 από το βιβλίο του Ashenden.
- Διαβάζετε τις παραγράφους 1.1, 1.2, 1.3, 1.4.1 – 1.4.5, 1.5, 4.1 από το βιβλίο των Harris.
- Διαβάζετε τις παραγράφους 1.1, 1.2, 1.3, 1.5, 1.5.1, 1.5.2, 2.1, 2.2, 2.3, 2.4, 2.5, 2.5.2, 2.5.3, 2.9. 2.10, 2.10.1 από το βιβλίο των Brown - Vranesic
- VHDL Reference Guide: https://peterfab.com/ref/vhdl/vhdl_renerta/source/vhd00000.htm