



dscal
DIGITAL SYSTEMS & COMPUTER ARCHITECTURE LABORATORY

Εργαστήριο Σχεδίασης Ψηφιακών Συστημάτων

Εισαγωγή

Βασιλόπουλος Διονύσης

Ε.ΔΙ.Π Τμήματος Πληροφορικής & Τηλεπικοινωνιών

Αντικείμενο Μαθήματος

- 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



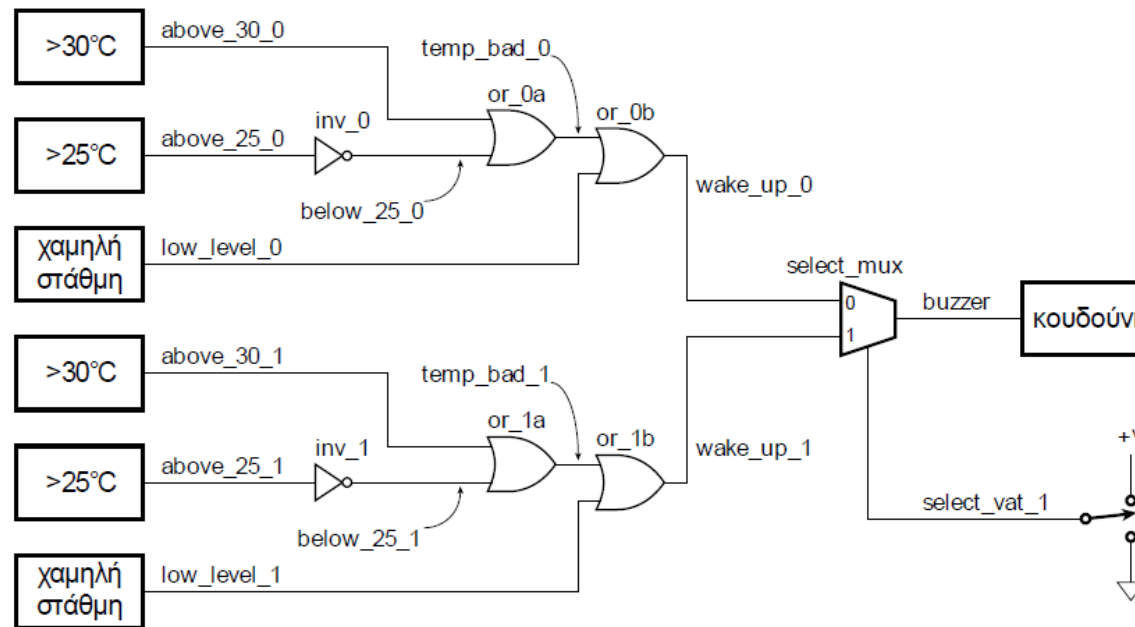
Αντικείμενο Μαθήματος

Εργαστήριο – Περιεχόμενο μαθήματος

- Βασικές Αρχές της **Γλώσσας Προγραμματισμού VHDL**
- **Ανάπτυξη απλών εφαρμογών** για κατανόηση του μοντέλου της VHDL
- **Χρήση του εργαλείου VIVADO** για την ανάπτυξη εφαρμογών
- **Βασικά στοιχεία της θεωρίας Λογικής Σχεδίασης** (δυναμικό σύστημα αρίθμησης, λογικές πύλες, συνδυαστικά κυκλώματα, ακολουθιακά κυκλώματα)
 - ΔΕΝ θα γίνει ανάπτυξη στα θέματα της θεωρίας της Ψηφιακής Σχεδίασης, παρά μόνο συνοπτικά και ΜΟΝΟ όσα χρειάζονται για την ανάπτυξη των εφαρμογών σε VHDL.
- **Προγραμματισμός κάρτας FPGA** (μέσω VHDL+VIVADO) σε πραγματικές συνθήκες

Αντικείμενο Μαθήματος

Λογικά κυκλώματα



Αντικείμενο Μαθήματος

Προγραμματισμός σε VHDL

Χρήση πρότυπης βιβλιοθήκης

```
library ieee; use ieee.std_logic_1164.all;
entity 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_1  : in std_logic;
         buzzer        : out std_logic );
end entity vat_buzzer;
```

Entity name

Port list

Port type

Port mode

Port name

Αντικείμενο Μαθήματος

Εργαστήριο Σχεδίασης Ψηφιακών Συστημάτων – Οργάνωση μαθήματος 1/4

- Διδάσκων Εργαστηρίου: Διονύσης Βασιλόπουλος (denis@di.uoa.gr) – Γραφείο: A33
- 3 διαλέξεις εισαγωγικές και 5 σετ Θεωρίας/εργαστηρίου (θεωρία/παράδοση-χρήση εργαλείου Vivado, επίλυση άσκησης-προγραμματισμός κάρτας) (αρχικός προγραμματισμός μαθήματος).
- Σύνολο 8-9 διαλέξεις στο Αμφιθέατρο A1 και 4-5 μαθήματα στο Εργαστήριο Ψηφιακής Σχεδίασης & ΗΥ Υψηλών Επιδόσεων (Αναγνωστήριο)

Αντικείμενο Μαθήματος

Εργαστήριο Σχεδίασης Ψηφιακών Συστημάτων – Οργάνωση μαθήματος 2/4

Ο τελικός **βαθμός** προκύπτει από:

- Project 0: μέγιστο βαθμού 7. Περιλαμβάνει 3 εργασίες (συνδυαστικά, ακολουθιακά και cpu 4 εντολών). (προπτυχιακοί, μεταπτυχιακοί)
- Project 1: μέγιστο βαθμού 10 (υλοποίηση CPU ενός κύκλου) (προπτυχιακοί).
- Project 1: μέγιστο βαθμού 9 (υλοποίηση CPU ενός κύκλου) (μεταπτυχιακοί).
- Project 2: μέγιστο βαθμού 10 (υλοποίηση CPU πολλών κύκλων) (μεταπτυχιακοί).
- **Τελικός βαθμός:** $\max(\text{project0}, \text{project1})$ (προπτυχιακοί), $\max(\text{project0}, \text{project1}, \text{project2})$ (μεταπτυχιακοί).

Επιλέγετε μόνοι σας πιο project θέλετε να κάνετε. Συνιστάται όμως ακόμα και όσοι επιλέξουν τα project1 και project2 να κάνουν και το project0 γιατί θα τους βοηθήσει πάρα πολύ στην επίλυσή τους, ειδικά αν έχουν ελλείψεις σε θέματα λογικής σχεδίασης και VHDL.

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 και Linux (όχι για Mac)



Αντικείμενο Μαθήματος

Εργαστήριο Σχεδίασης Ψηφιακών Συστημάτων – Οργάνωση μαθήματος 4/4

Προτεινόμενη Βιβλιογραφία

1. **Ψηφιακή Σχεδίαση. Ενσωματωμένα Συστήματα με VHDL**, Peter J. Ashenden, Εκδόσεις Νέων Τεχνολογιών, 2010
2. **Ψηφιακή Σχεδίαση και Αρχιτεκτονική Υπολογιστών**, S.L.Harris, D.M.Harris, Εκδόσεις ARM®, 2019

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.

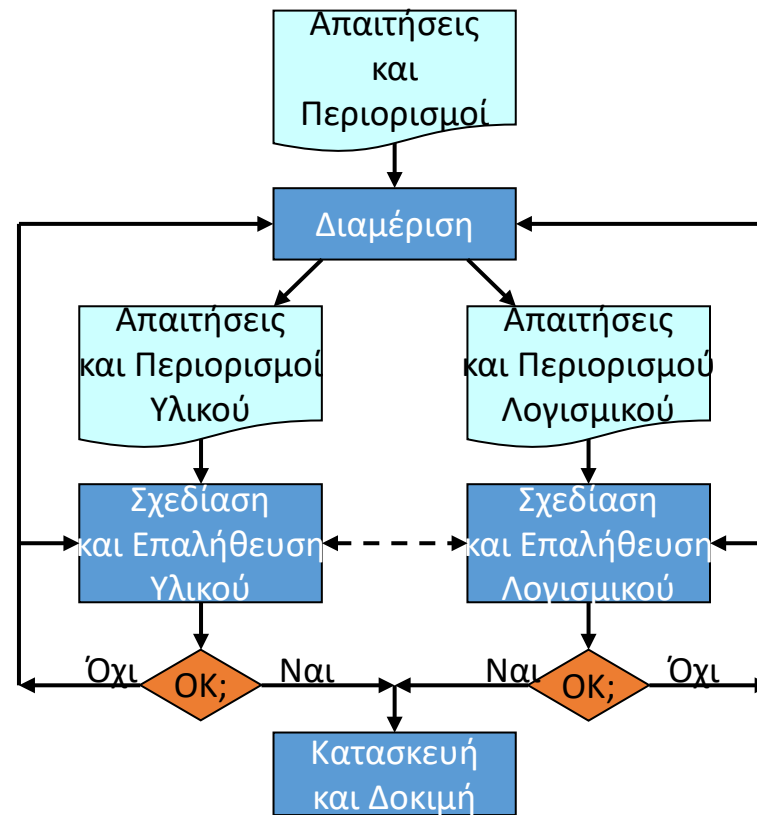
Ψηφιακά Συστήματα

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

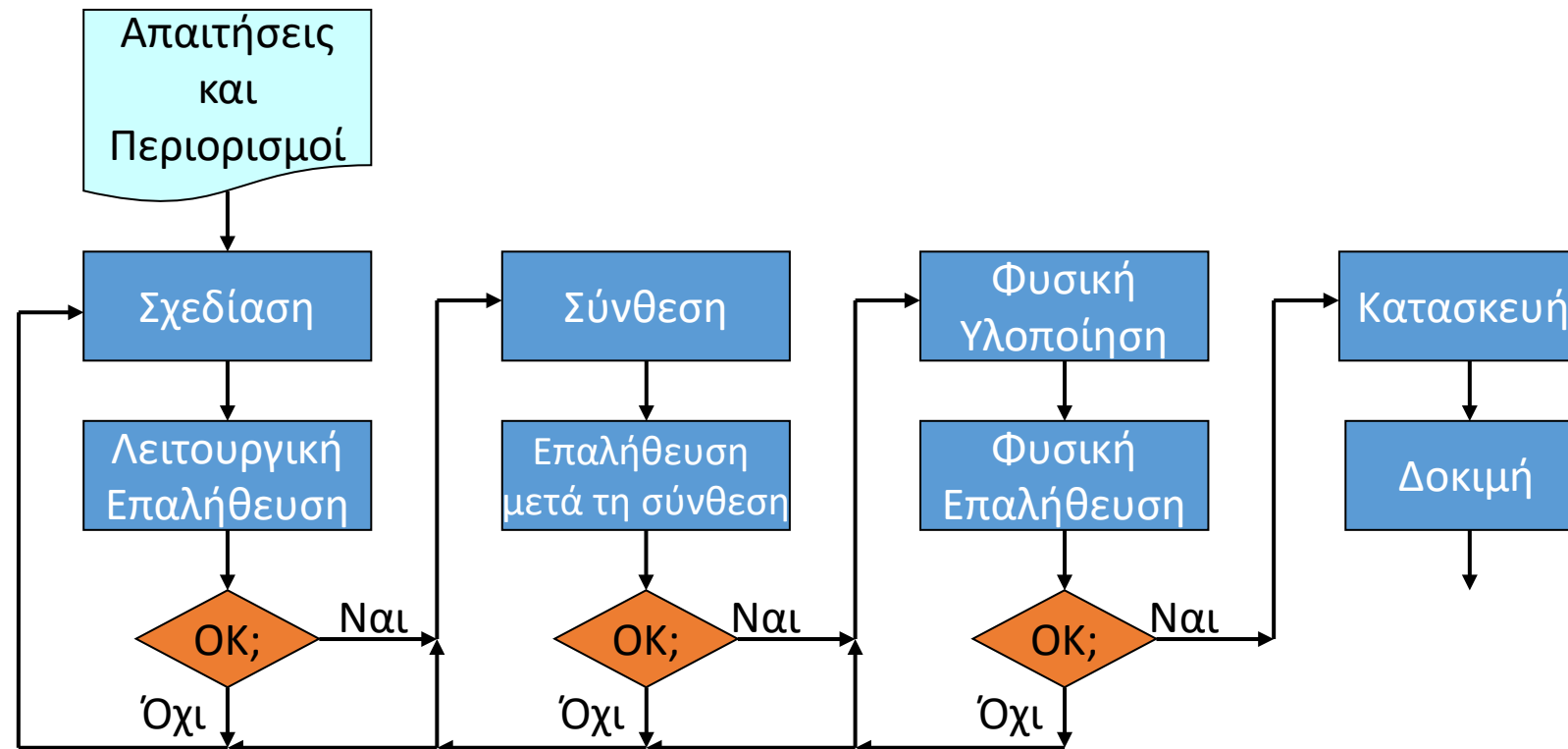
Ψηφιακή Σχεδίαση – Μεθοδολογία Συσχεδίασης



Κάθε ομάδα υλοποιεί ένα ξεχωριστό Module του συστήματος

Ψηφιακά Συστήματα

Ψηφιακή Σχεδίαση – Μεθοδολογία ανάπτυξης



Ψηφιακά Συστήματα

Δυαδική Αναπαράσταση

- Δεκαδικό σύστημα αναπαράστασης αριθμών (0-9)

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

6598₁₀

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

- Δυαδικό σύστημα αναπαράστασης αριθμών (0-1)

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

1101₂

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

Ψηφιακά Συστήματα

Δυαδική Αναπαράσταση

- Δεκαδικό σύστημα αναπαράστασης αριθμών (0-9)

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

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

Έξι Πέντε Εννέα Οκτώ
Χιλιάδες Εκατοντάδες Δεκάδες Μονάδες

- Δυαδικό σύστημα αναπαράστασης αριθμών (0-1)

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

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

Μια Μια Μηδέν Μια
Οκτάδα Τετράδα Δυάδες 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

$$10 = 2(\text{δεκαδικό}) \quad (A)$$

Εφαρμόζουμε αριστερή ολίσθηση και γεμίζουμε δεξιά με το 0

$$10\mathbf{0} = 4(\text{δεκαδικό}) \quad (B) = 2 * (A)$$

$$10\mathbf{00} = 8(\text{δεκαδικό}) \quad (C) = 2 * (B) = 4 * (A)$$

$$1000 = 8(\text{δεκαδικό}) \quad (A)$$

Εφαρμόζουμε δεξιά ολίσθηση διαγράφοντας το πιο δεξί ψηφίο

$$100 = 4(\text{δεκαδικό}) \quad (B) = (A)/2$$

$$10 = 2(\text{δεκαδικό}) \quad (C) = (B)/2 = (A)/4$$

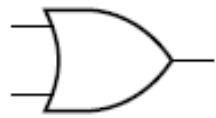
Ψηφιακά Συστήματα

Διαδική Αναπαράσταση

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



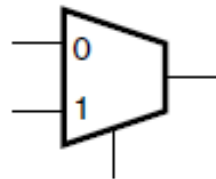
AND gate



OR gate



inverter



multiplexer

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

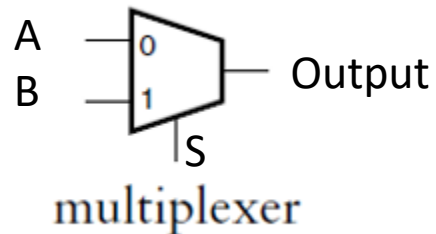
x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

x	\bar{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


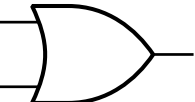
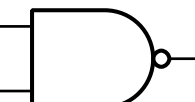



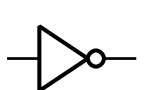


S	Output
0	A
1	B

Αν $A=0$, $B=1$, $S=1$, Output= ;

Αν $A=1$, $B=0$, $S=1$, Output= ;

Λογικοί Τελεστές

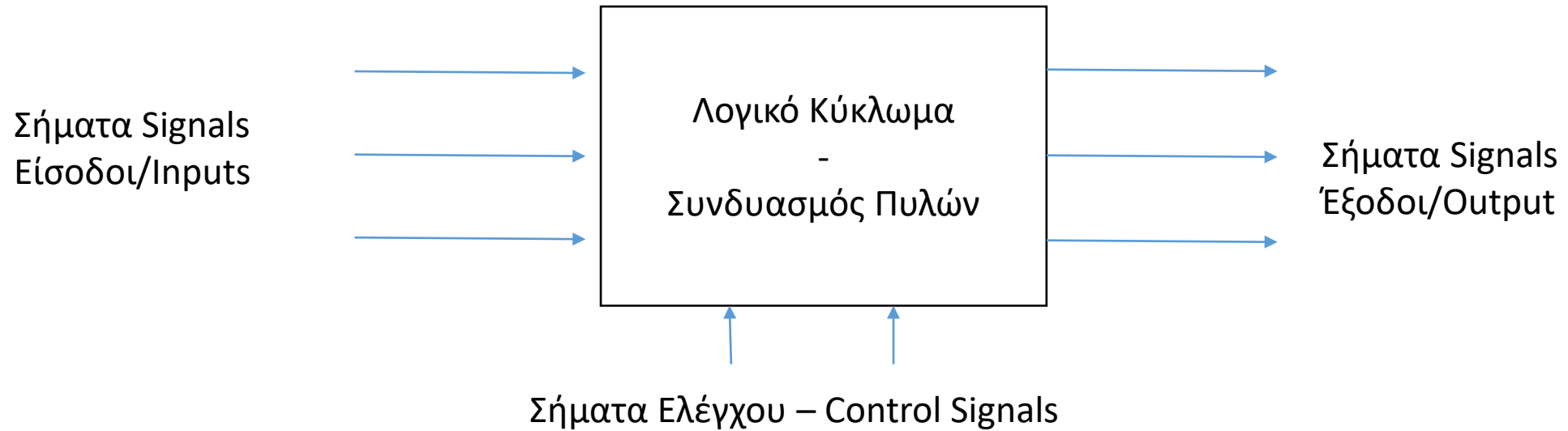
<code>a and b</code>	$a \cdot b$	
<code>a or b</code>	$a + b$	
<code>a nand b</code>	$\overline{a \cdot b}$	
<code>a nor b</code>	$\overline{a + b}$	
<code>a xor b</code>	$a \oplus b$	
<code>a xnor b</code>	$\overline{a \oplus b}$	
<code>not a</code>	\overline{a}	

- 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'

Λογικοί Τελεστές

	Τελεστής				Σημασία
Υψηλότερη	not				NOT
	*	/	mod	rem	MUL, DIV, MOD, REM
	+ -				PLUS, MINUS
	rol	ror	srl sll		Περιστροφή, λογική ολίσθηση
	<	<=	>	>=	Σχετική σύγκριση
	= /=				Σύγκριση ισότητας
Χαμηλότερη	and	or	nand		Λογικές πράξεις (εκτελούνται από αριστερά προς τα δεξιά)
	nor	xor	xnor		

Ψηφιακά Συστήματα

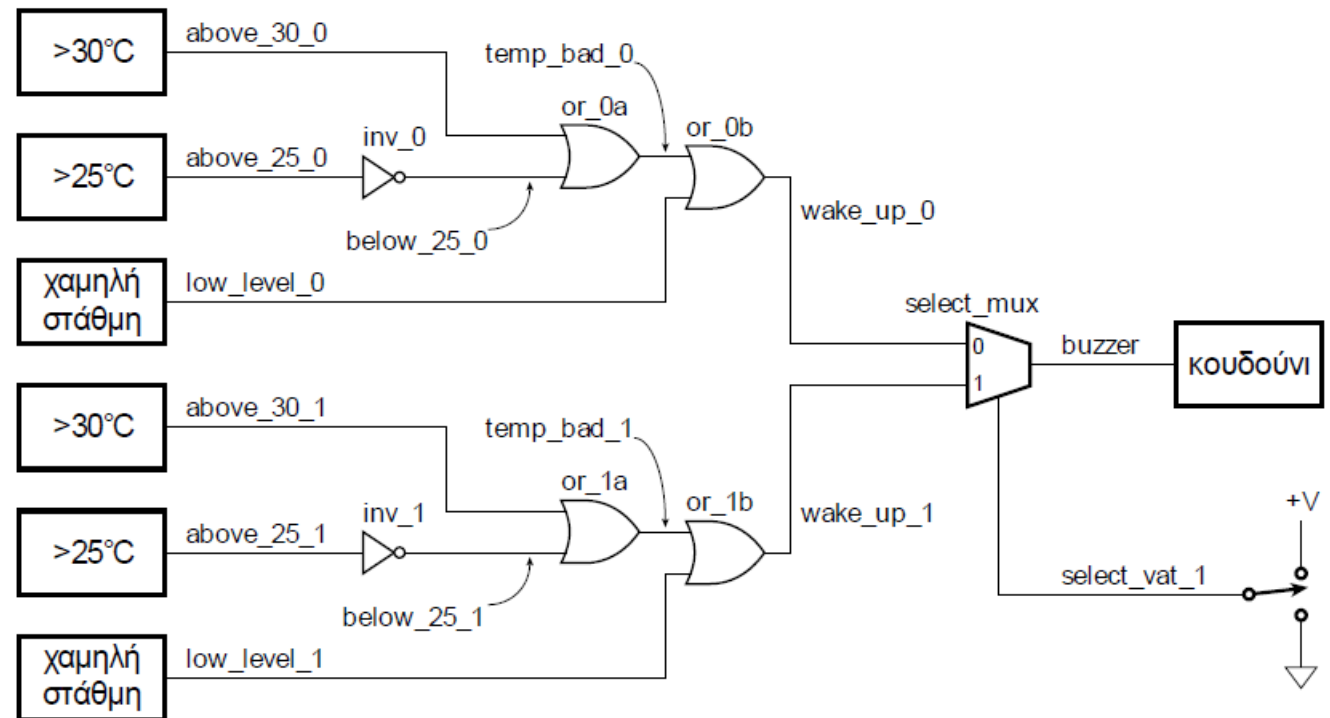


Ψηφιακά Συστήματα

Ψηφιακό κύκλωμα

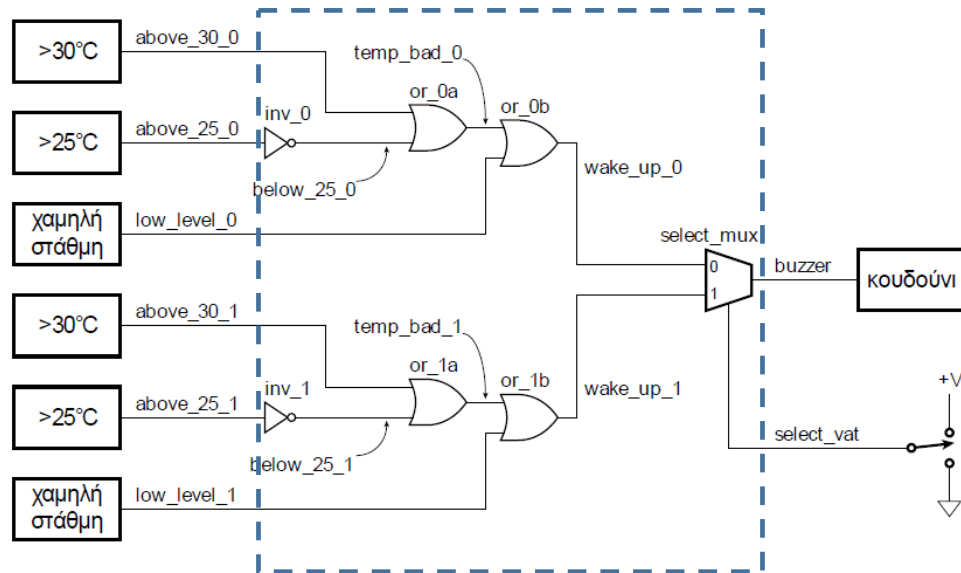
Παράδειγμα:

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 – Structural Architecture



architecture struct of vat_buzzer is

Περιγραφή δομικών στοιχείων

```
signal below_25_0, temp_bad_0, wake_up_0 : std_logic;  
signal below_25_1, temp_bad_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_bad_0);
```

```
or_0b : or2 port map (temp_bad_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_bad_1);
```

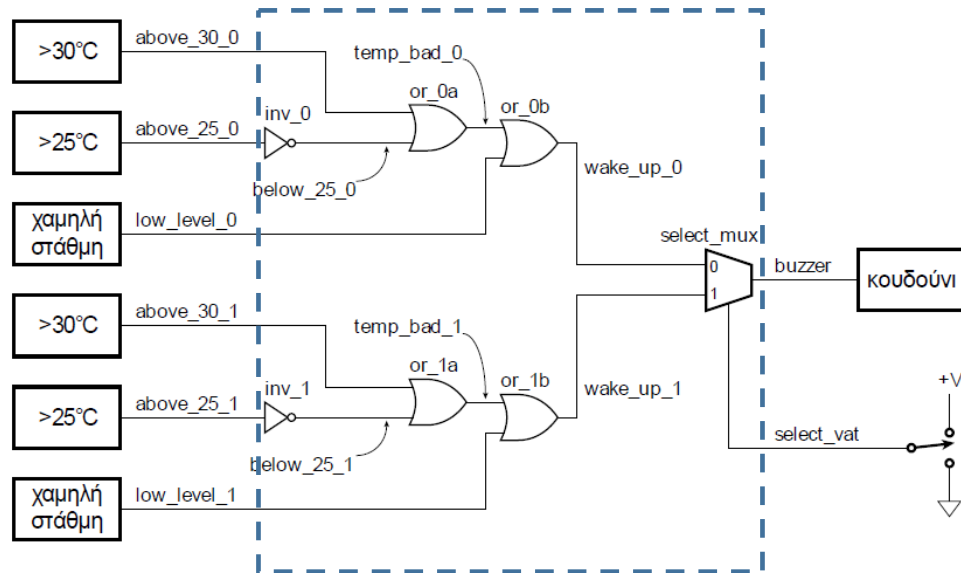
```
or_1b : or2 port map (temp_bad_1, low_level_1, wake_up_1);
```

```
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 behavior of vat_buzzer is

begin

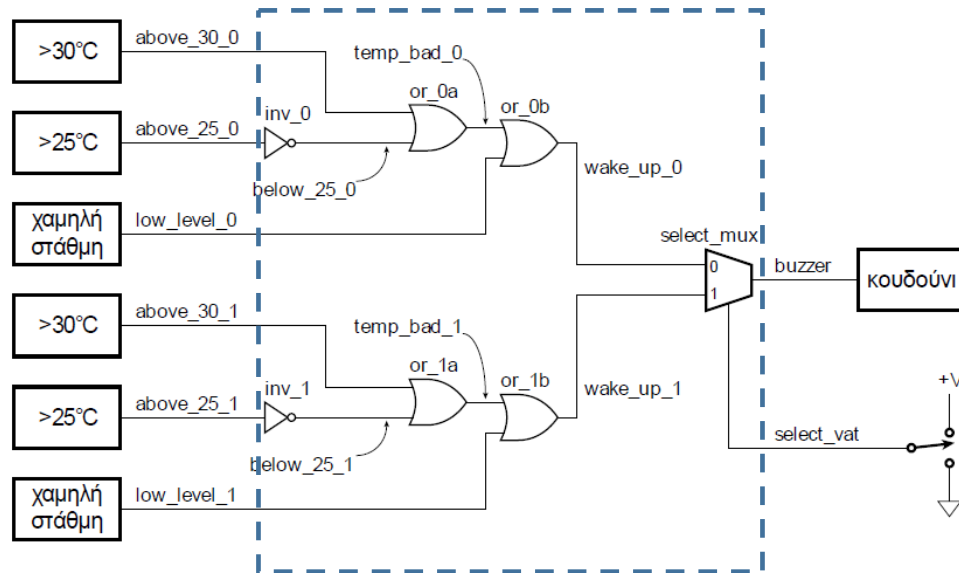
```
buzzer <= low_level_1 or (above_30_1 or not above_25_1)
when select_vat='1' else
low_level_0 or (above_30_0 or not above_25_0);
```

end architecture behavior;

Αρχιτεκτονική
Περιγραφή της λειτουργικότητας
που προσφέρει το λογικό
κύκλωμα

Ψηφιακά Συστήματα

Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – Behavioral Architecture



architecture behavior of vat_buzzer is

begin

process (low_level_1, above_30_1, above_25_1,
low_level_1, above_30_1, above_25_1
select_vat) **is**

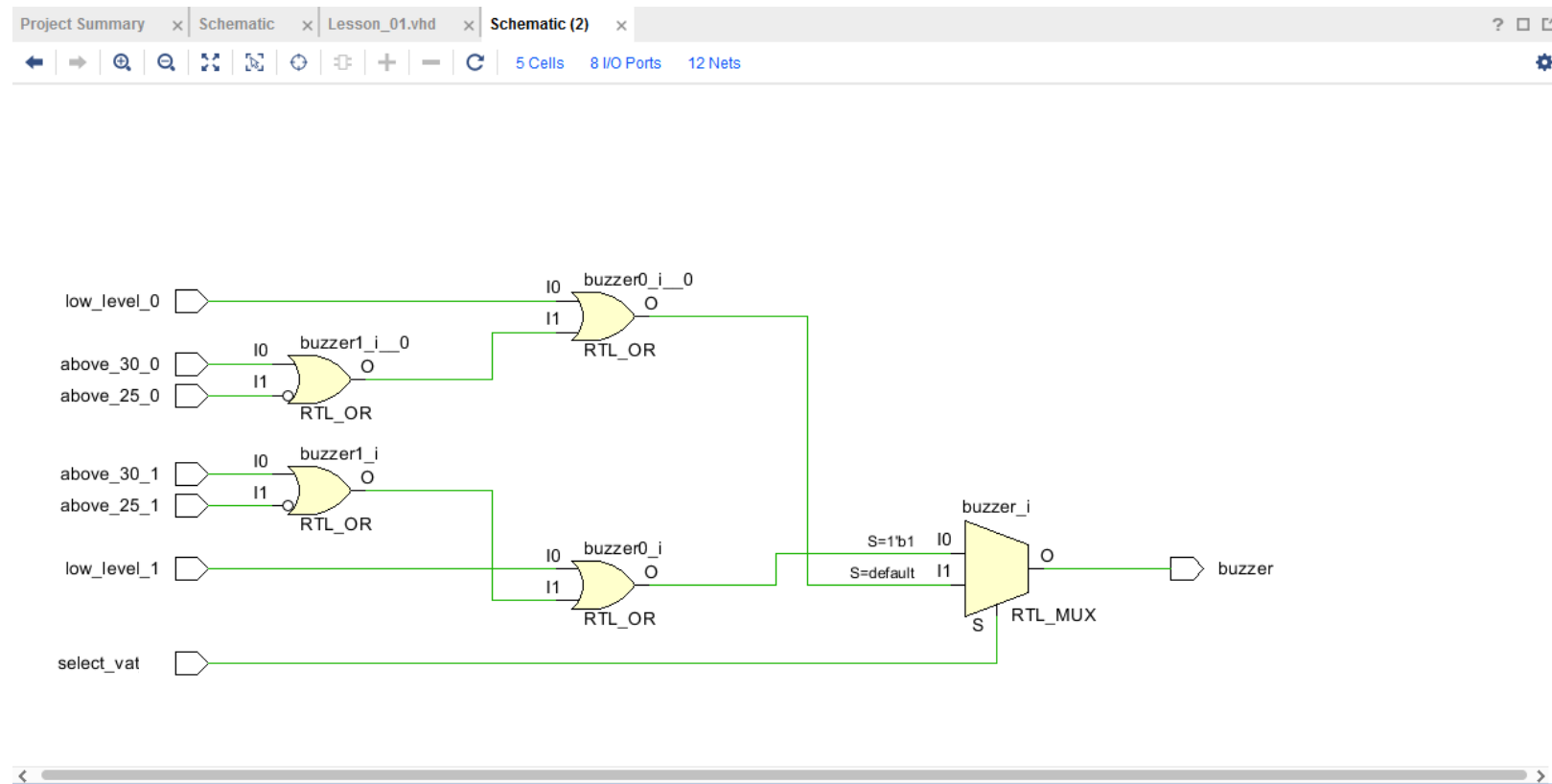
begin

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;

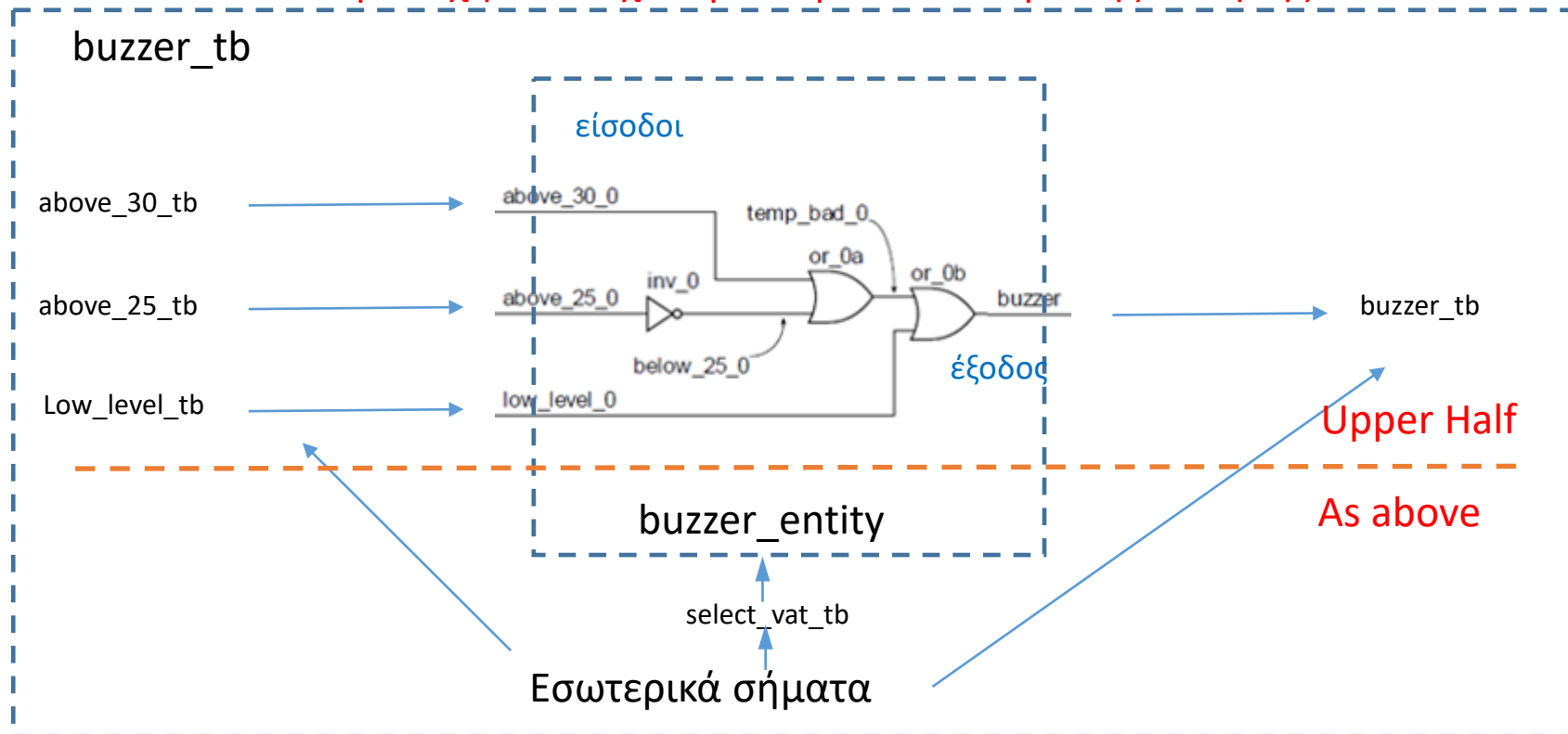
end architecture behavior;

Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – Λογικές πύλες



Ψηφιακό κύκλωμα – VHDL: Προσομοίωση

! Προσοχή ότι δείχνουμε το μισό κύκλωμα της άσκησης



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

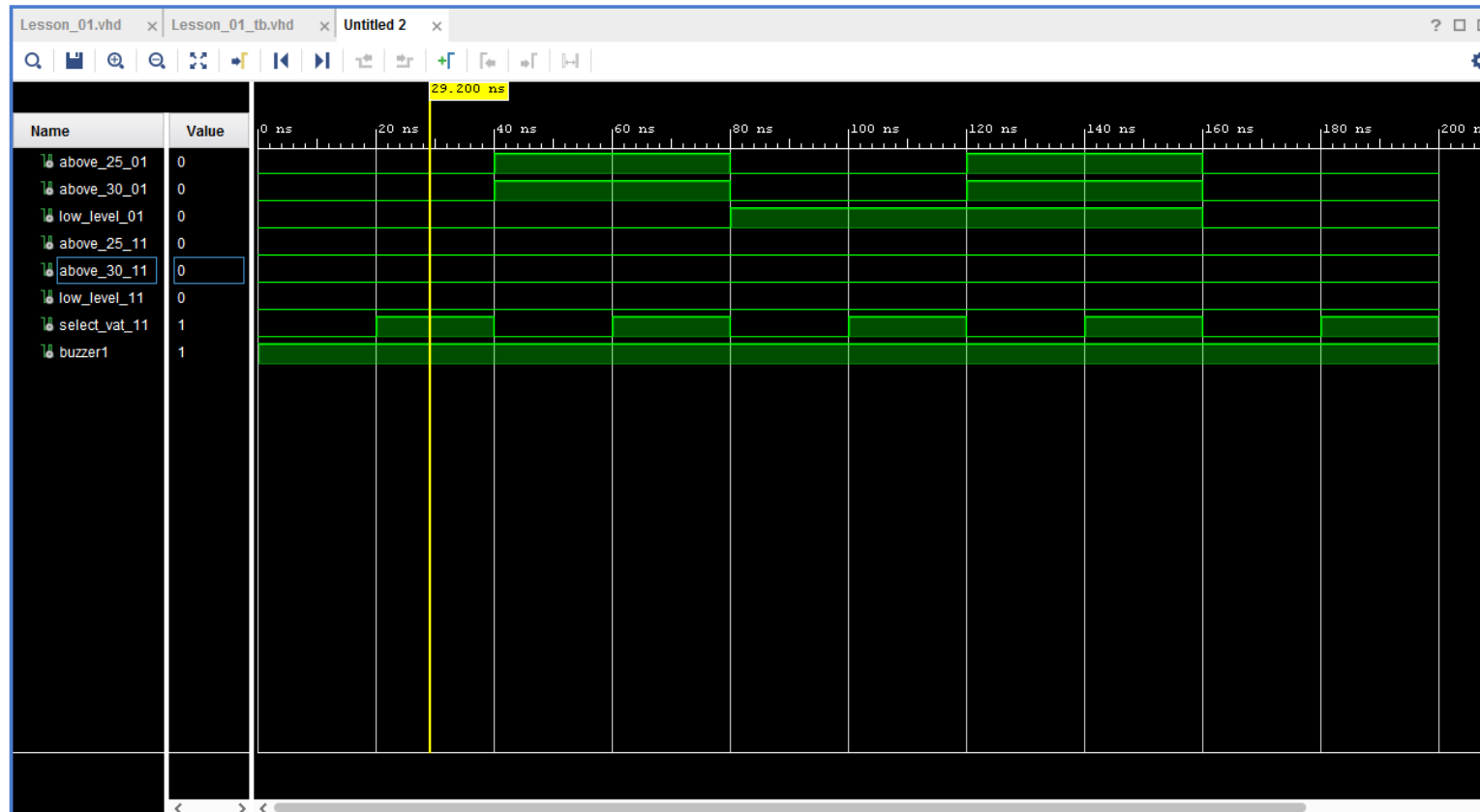
component buzzer is
port (
above_25_0, above_30_0, low_level_0      : in std_logic;
above_25_1, above_30_1, low_level_1      : in std_logic;
select_vat                               : in std_logic;
buzzer                                    : out std_logic;
);
end component buzzer
signal above_25_0_tb, 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 ;
```

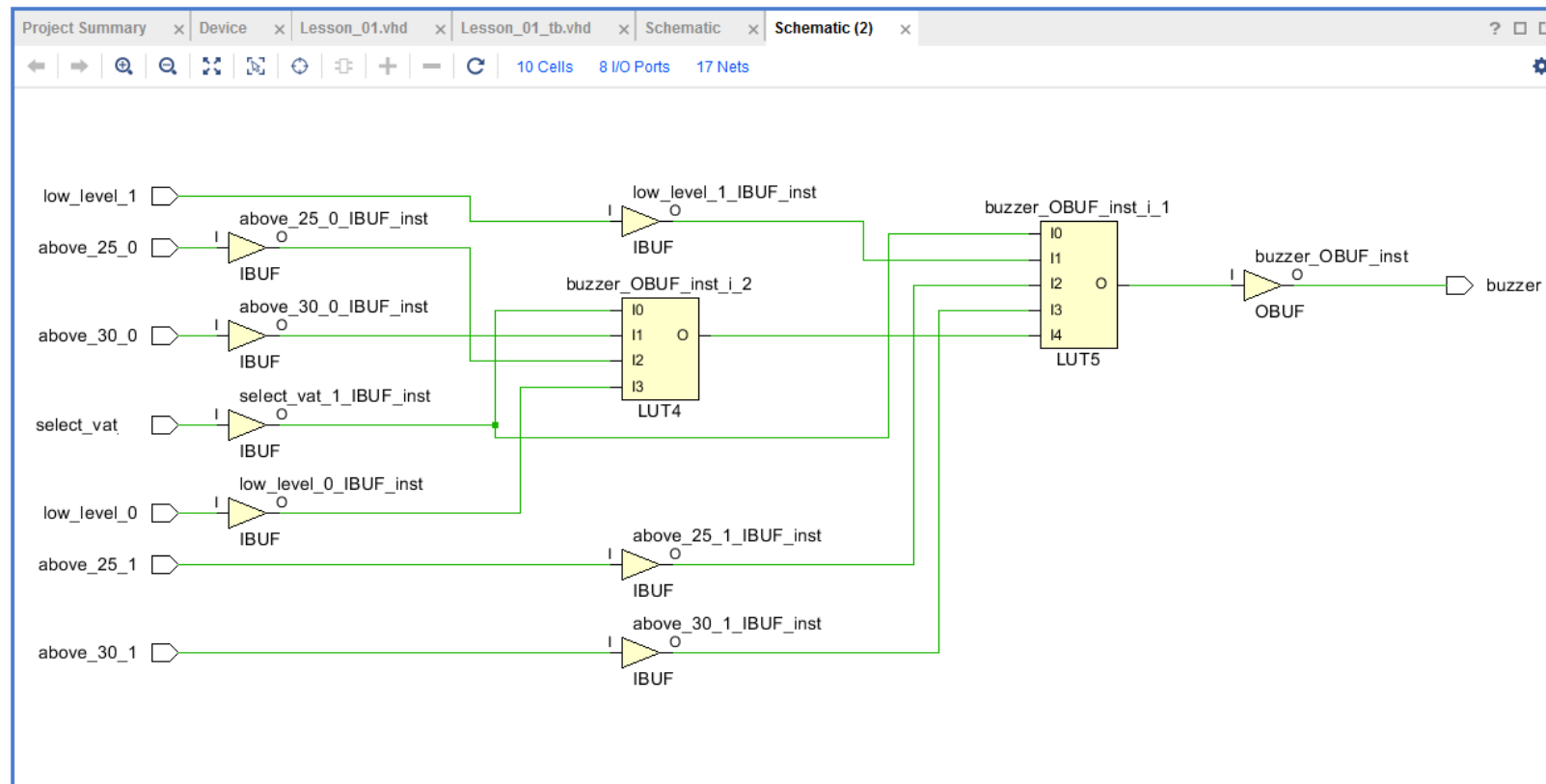
Ενδεικτικές τιμές. Όχι όλος ο πίνακας αληθείας

VHDL - VIVADO

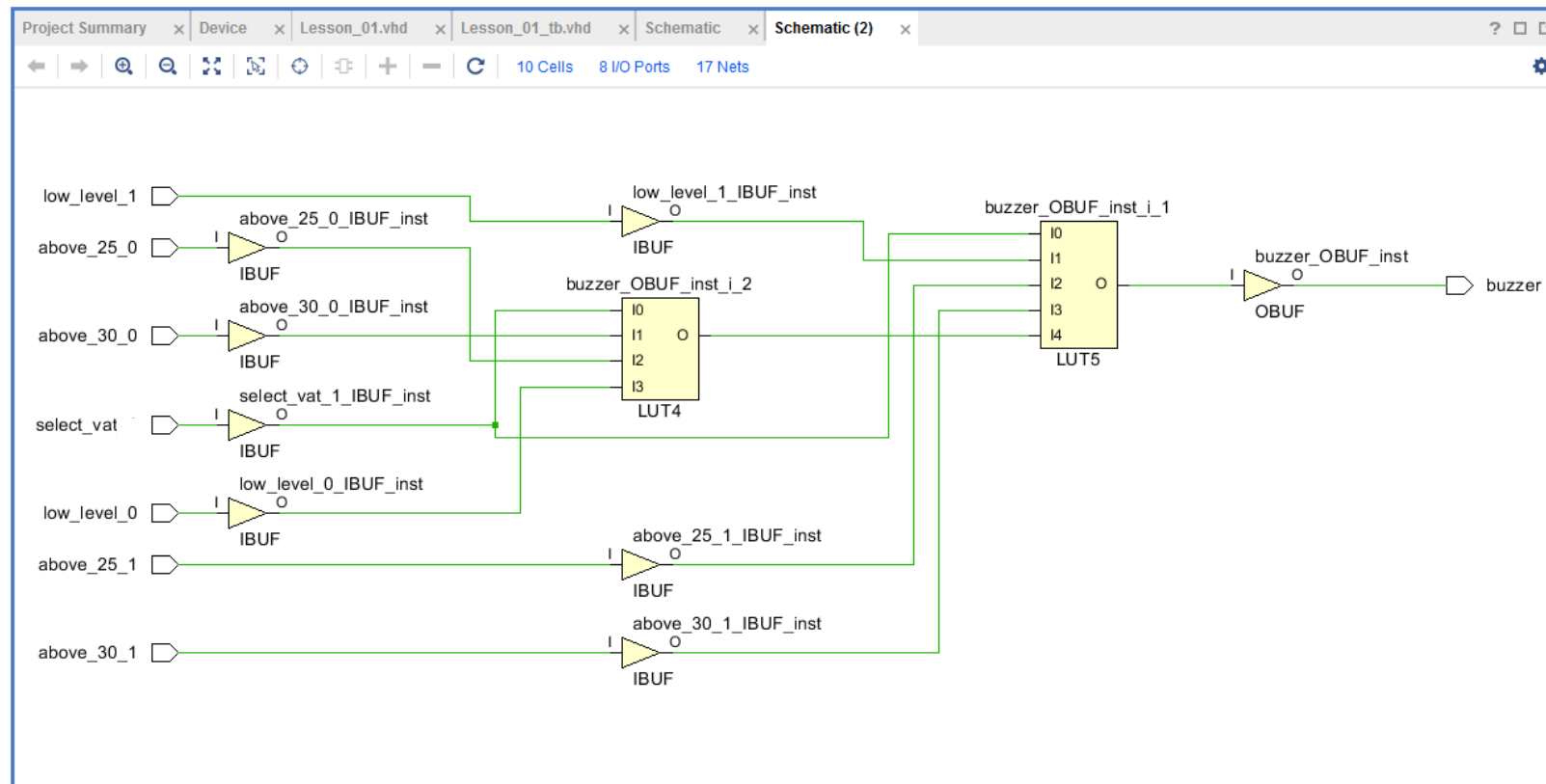
Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – Προσομοίωση Χρονική



Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – Σύνθεση

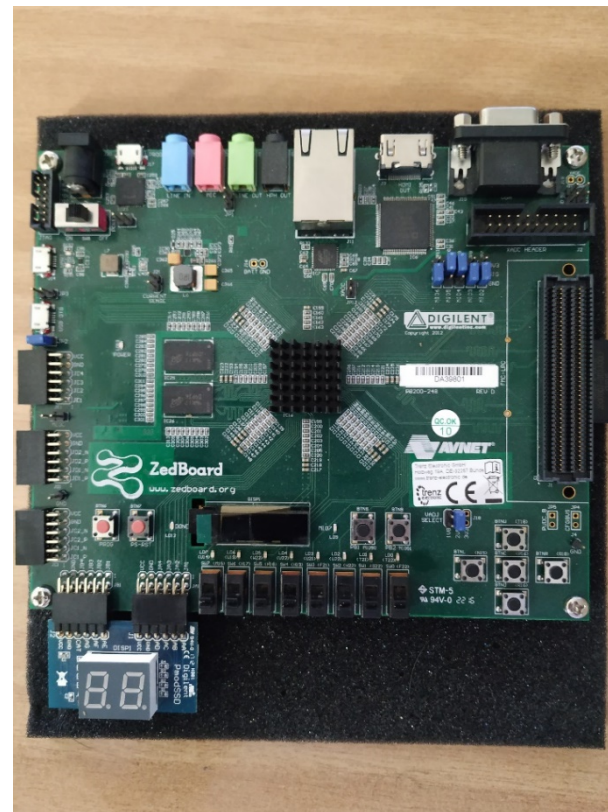


Ψηφιακό κύκλωμα – Αναπαράσταση VHDL – Υλοποίηση



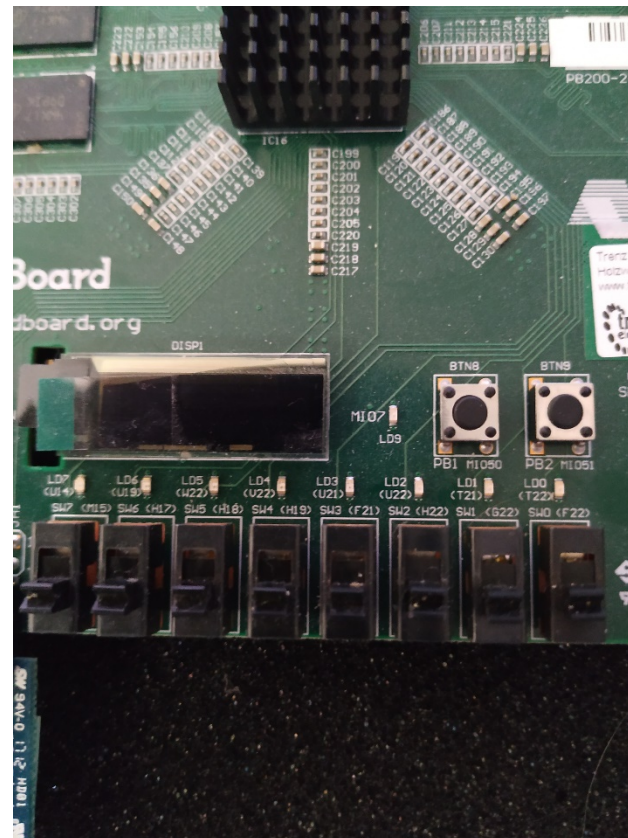
VHDL - VIVADO

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 srxxxxxxx@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 srxxxxxxx@o365.uoa.gr και μετά θα σας ζητηθούν τα ακαδημαϊκά σας credentials)

Προσοχή! Δεν είναι βέβαιο ότι θα ισχύουν για όλα τα account

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 από το βιβλίο των Harris.
- VHDL Reference Guide: https://peterfab.com/ref/vhdl/vhdl_renerta/source/vhd00000.htm