



**dscal**  
DIGITAL SYSTEMS & COMPUTER ARCHITECTURE LABORATORY

# Εργαστήριο Λογικής Σχεδίασης

## 2ο Εργαστηριακό Μάθημα

**Βασιλόπουλος Διονύσης**

**Ε.Δι.Π. Τμήματος Πληροφορικής & Τηλεπικοινωνιών - ΕΚΠΑ**

# 2<sup>ο</sup> Εργαστηριακό Μάθημα

## Περιβάλλον Linux

USER=**guest**

PASSWORD=**linux!**

Για να εκτελεστεί το Vivado πρέπει να γράψετε τις ακόλουθες εντολές:  
(υπάρχει και στο eclass, κάνετε copy/paste κάθε γραμμή χωριστά στο **terminal**)

```
source /opt/Xilinx/Vivado/2022.2/settings64.sh
```

```
cd $home
```

```
cd VIVADO-users/
```

```
mkdir sdi2400XXX
```

```
cd sdi2400XXX
```

```
vivado
```

← Όπου sdi2400XXX είναι ο AM σας

# 2<sup>ο</sup> Εργαστηριακό Μάθημα

## ALU

Να σχεδιάσετε και να προσομοιώσετε στο Vivado μία αριθμητική και λογική μονάδα (ALU) η οποία θα δέχεται στην είσοδο δύο σήματα μη προσημασμένων αριθμών  $a$  και  $b$ , των 3 bit το καθένα. Η ALU κάνει μια απλή πρόσθεση ( $a+b$ ). Στην έξοδο υπάρχει το σήμα Result των 3 bit με το αποτέλεσμα της πράξης και ένα σήμα Carry που έχει τιμή '1' σε περίπτωση που υπάρχει κρατούμενο/υπερχείλιση.

Το όνομα του project θα είναι Lab2, το όνομα του αρχείου (design source) αλλά και η οντότητα σας θα λέγεται alu, ενώ η αρχιτεκτονική Dataflow. Τα αντίστοιχα ονόματα για την προσομοίωση θα είναι alu\_tb, και Dataflow\_tb

Δημιουργήστε την οντότητα, γράψτε την αρχιτεκτονική, εμφανίστε τα RTL και Synthesis διαγράμματα, και τέλος προγραμματίστε την κάρτα.

# 2ο Εργαστηριακό Μάθημα

## Συσχέτιση PIN κάρτας FPGA με port Οντότητας

Είσοδοι	DIP Switch
b[2]	SW5
b[1]	SW4
b[0]	SW3
a[2]	SW2
a[1]	SW1
a[0]	SW0

Έξοδοι	LED
Carry	LD7
Result[2]	LD2
Result[1]	LD1
Result[0]	LD0

# 2ο Εργαστηριακό Μάθημα

## Βήματα επίλυσης

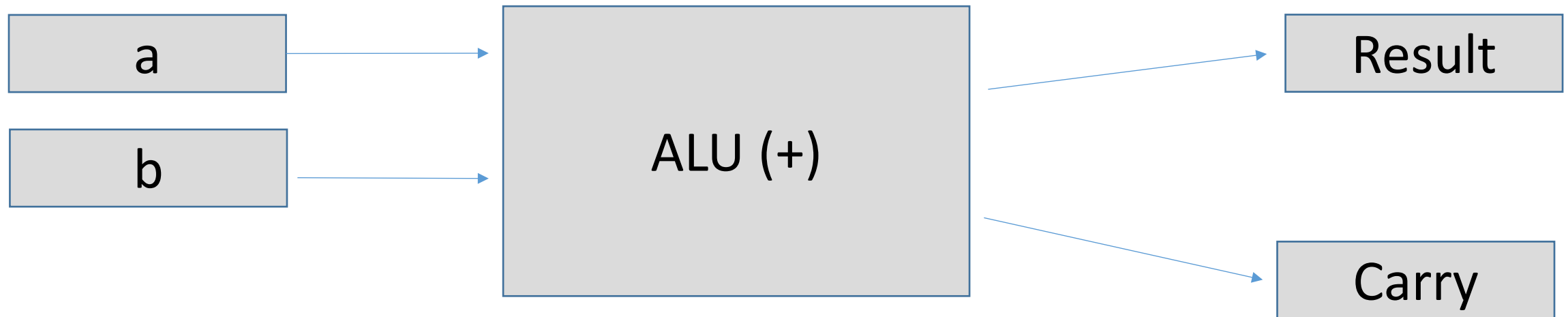
1. Δημιουργία νέου project
2. Δημιουργία Entity – Εντοπισμός Input/Output του συστήματος
3. Εύρεση πίνακα αληθείας για κάθε έξοδο του συστήματος (δεν χρειάζεται στην περίπτωσή μας)
4. Δημιουργία Architecture – Θα έχετε τουλάχιστον τόσες εντολές όσες είναι και οι έξοδοι του συστήματος. Κάθε μία εντολή αντιστοιχεί σε μία έξοδο.
5. Δημιουργία RTL αναπαράστασης
6. Σύνθεση
7. Υλοποίηση

Προγραμματισμός κάρτας (Έγινε μόνο στο Εργαστήριο)

**8. Προσομοίωση (Παρουσιάζεται μόνο στις διαφάνειες)**

# 2ο Εργαστηριακό Μάθημα

## Απλοποιημένη μορφή κυκλώματος – Είσοδοι/Εξοδοι



# 2ο Εργαστηριακό Μάθημα

## Βήμα 2: Περιγραφή Οντότητας

```
entity ALU is
  Port (
    a      : in STD_LOGIC_VECTOR (2 downto 0);
    b      : in STD_LOGIC_VECTOR (2 downto 0);
    Result : out STD_LOGIC_VECTOR (2 downto 0);
    Carry  : out STD_LOGIC
  );
end entity ALU;
```

# 2ο Εργαστηριακό Μάθημα

## Βήμα 3: Πίνακας αληθείας κυκλώματος

ΔΕΝ ΧΡΕΙΑΖΕΤΑΙ ΓΙΑ ΤΗΝ ΠΕΡΙΠΤΩΣΗ ΜΑΣ



# 2ο Εργαστηριακό Μάθημα

## Βήμα 4: Περιγραφή Αρχιτεκτονικής—Μία λύση χωρίς process (a)

```
Result_temp<=unsigned('0'&a)+unsigned('0'&b);
```

```
Result<=std_logic_vector(Result_temp(2 downto 0));
```

```
Carry<=Result_temp(3);
```

# 2ο Εργαστηριακό Μάθημα

## Βήμα 4: Περιγραφή Αρχιτεκτονικής—Μία λύση χωρίς process (b)

```
Result_temp<=resize(unsigned(a),Result_temp'length)+resize(unsigned(b),Result_temp'length);
```

```
Result<=std_logic_vector(Result_temp(2 downto 0));
```

```
Carry<=Result_temp(3);
```

# 2ο Εργαστηριακό Μάθημα

## Βήμα 4: Περιγραφή Αρχιτεκτονικής – Μία λύση με process

```
solution: process (a,b) is
variable Result_var    : unsigned (3 downto 0);
begin

Result_var:=unsigned('0'&a)+unsigned('0'&b);
Result<=std_logic_vector(Result_var(2 downto 0));
Carry<=Result_var(3);

end process solution;
```

# 2ο Εργαστηριακό Μάθημα

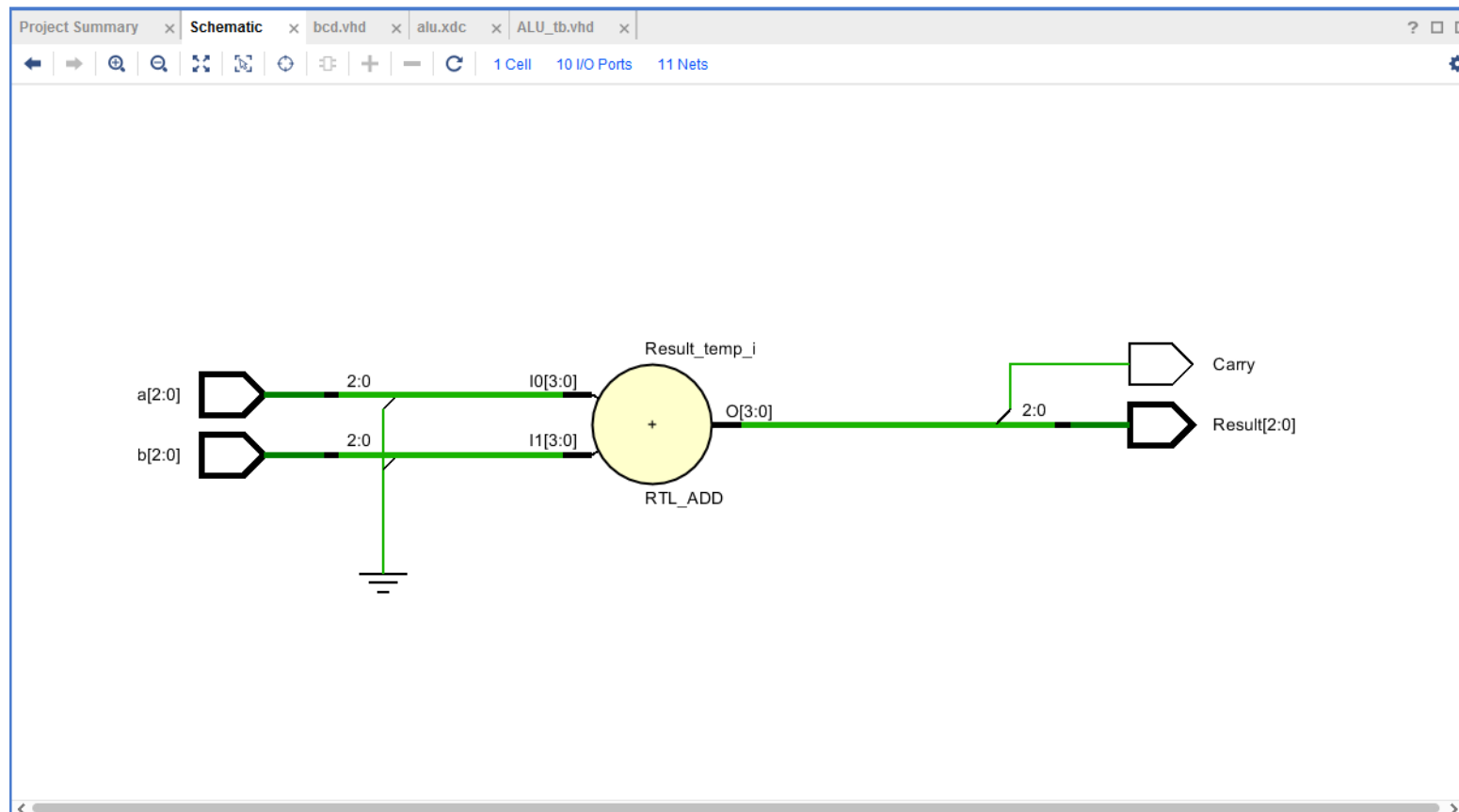
## Βήμα 4: Περιγραφή Αρχιτεκτονικής – Αρχείο Constraints

```
set_property -dict { PACKAGE_PIN F22  IOSTANDARD LVCMOS33 } [get_ports { a[0] }];  
set_property -dict { PACKAGE_PIN G22  IOSTANDARD LVCMOS33 } [get_ports { a[1] }];  
set_property -dict { PACKAGE_PIN H22  IOSTANDARD LVCMOS33 } [get_ports { a[2] }];  
set_property -dict { PACKAGE_PIN F21  IOSTANDARD LVCMOS33 } [get_ports { b[0] }];  
set_property -dict { PACKAGE_PIN H19  IOSTANDARD LVCMOS33 } [get_ports { b[1] }];  
set_property -dict { PACKAGE_PIN H18  IOSTANDARD LVCMOS33 } [get_ports { b[2] }];
```

```
set_property -dict { PACKAGE_PIN T22  IOSTANDARD LVCMOS33 } [get_ports { Result[0] }];  
set_property -dict { PACKAGE_PIN T21  IOSTANDARD LVCMOS33 } [get_ports { Result[1] }];  
set_property -dict { PACKAGE_PIN U22  IOSTANDARD LVCMOS33 } [get_ports { Result[2] }];  
set_property -dict { PACKAGE_PIN U14  IOSTANDARD LVCMOS33 } [get_ports { Carry }];
```

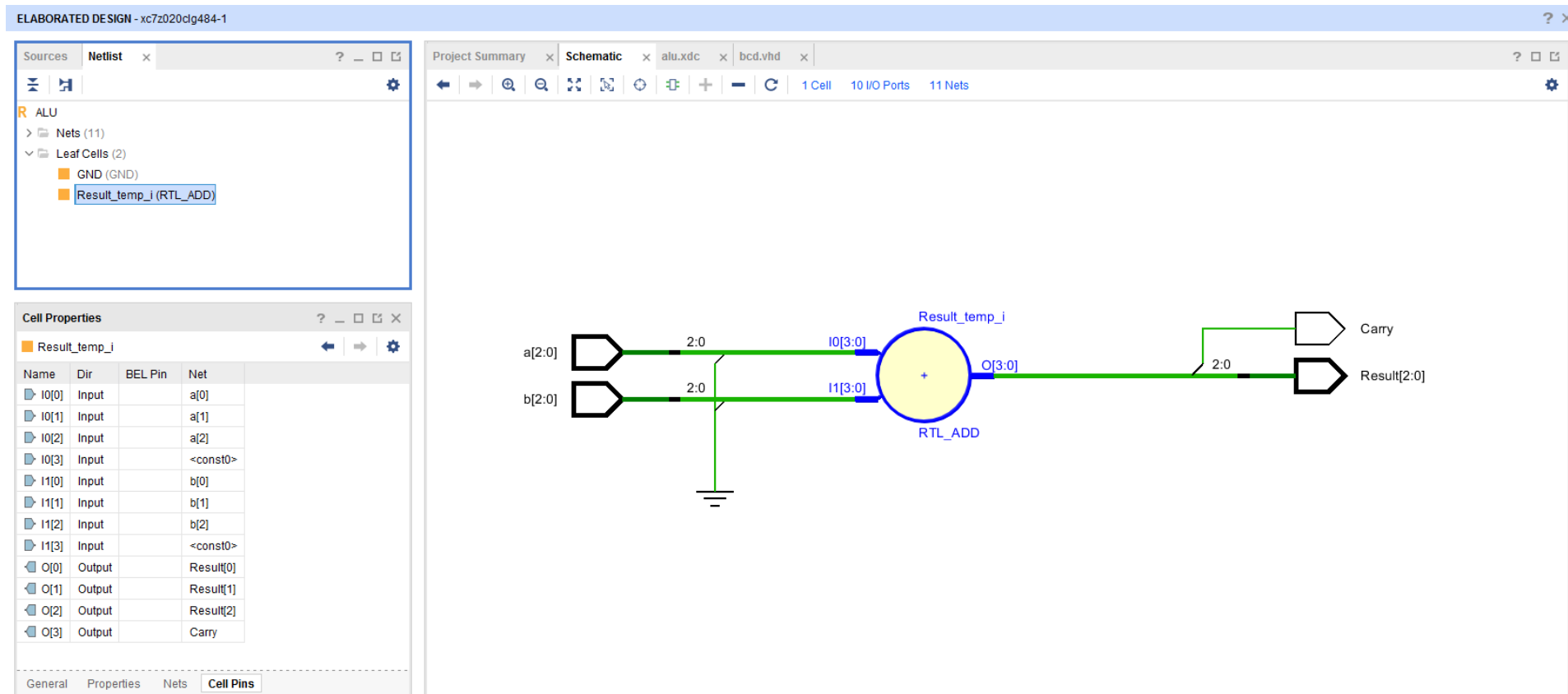
# 2ο Εργαστηριακό Μάθημα

## Βήμα 5a: RTL Analysis



# 2ο Εργαστηριακό Μάθημα

## Βήμα 5b: RTL Analysis



# 2ο Εργαστηριακό Μάθημα

## Βήμα 6-7: Synthesis/Implementation Scematic

Project Summary | Device | bcd.vhd | alu.xdc | ALU\_tb.vhd | Schematic | 14 Cells | 10 I/O Ports | 20 Nets

Cell Properties: Result\_OBUF[0]\_inst\_i\_1

I1	I0	O=I0 & I1 + !I0 & I1
0	0	0
0	1	1
1	0	1
1	1	0

Πίνακας Αληθείας για ψηφίο Result(0)

Κάθε LUT υλοποιεί τον Πίνακα Αληθείας για ένα ψηφίο του αποτελέσματος. Έχουμε 4 ψηφία (bit) στην έξοδο άρα και 4 LUT.

<https://nandland.com/lesson-4-what-is-a-look-up-table-lut/>

# 2ο Εργαστηριακό Μάθημα

## Βήμα 6-7: Synthesis/Implementation Device

Επιλέξτε  
Show Cell Connections

Επιλέξτε  
Κάποια ή όλα  
τα nets

Τα στοιχεία στο Device  
φαίνονται ορθά μετά  
το implementation



# 2ο Εργαστηριακό Μάθημα

## Βήμα 6-7: Synthesis/Implementation Device

The screenshot shows the Xilinx Vivado IDE interface during the implementation phase. The main window displays a schematic diagram of the device, showing various logic blocks and interconnections. The left sidebar shows the Netlist and Cell Properties for the selected component, Result\_OBUF[0]\_inst\_i\_1.

**Netlist:**

- a\_IBUF[1]\_inst (IBUF)
- a\_IBUF[2]\_inst (IBUF)
- b\_IBUF[0]\_inst (IBUF)
- b\_IBUF[1]\_inst (IBUF)
- b\_IBUF[2]\_inst (IBUF)
- Carry\_OBUF\_inst (OBUF)
- Carry\_OBUF\_inst\_i\_1 (LUT6)
- Result\_OBUF[0]\_inst (OBUF)
- Result\_OBUF[0]\_inst\_i\_1 (LUT2)

**Cell Properties for Result\_OBUF[0]\_inst\_i\_1:**

- Name: Result\_OBUF[0]\_inst\_i\_1
- Reference name: LUT2
- Type: LUT
- BEL:  A5LUT  Fixed
- Site: SLICE\_X113Y101
- Title: CLBLM\_R\_X71Y101
- Clock region: X1Y2
- Number of cell pins: 3

**Cell Properties for Result\_OBUF[0]\_inst:**

- BEL: 'CARRY4'
- Type: CARRY4
- Site: SLICE\_X113Y101
- Title: CLBLM\_R\_X71Y101

# 2ο Εργαστηριακό Μάθημα

## Implementation – Timing Reports (1/2)

The screenshot displays the Xilinx Vivado IDE. The top window shows the schematic of a 3-bit adder circuit. The bottom window shows the 'Timing Checks - Setup' report. The report table is as follows:

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Logic %	Net %	Requirement	Source Clock	Destination Clock	Exception	Skew	Clock Uncertainty
Unconstrained Paths (1)																	
(none) (4)																	
Path 5	∞	3	2	3	b[1]	Carry	12.528	5.060	7.469	40.4	59.6	∞	input port clock				0.000
Path 6	∞	3	2	3	b[1]	Result[2]	10.398	5.089	5.309	48.9	51.1	∞	input port clock				0.000
Path 7	∞	3	2	4	a[0]	Result[0]	10.349	5.361	4.988	51.8	48.2	∞	input port clock				0.000
Path 8	∞	3	2	3	b[1]	Result[1]	10.228	5.071	5.157	49.6	50.4	∞	input port clock				0.000

Menu Reports->Timing->Report Timing

Setup Time:

Αναφέρεται στις αργές διαδρομές  
(καθυστέρηση διάδοσης)

# 2ο Εργαστηριακό Μάθημα

## Implementation – Timing Reports (2/2)

Menu Reports->Timing->Report Timing

Hold Time:  
Αναφέρεται στις γρήγορες διαδρομές  
(καθυστέρηση μόλυνσης)

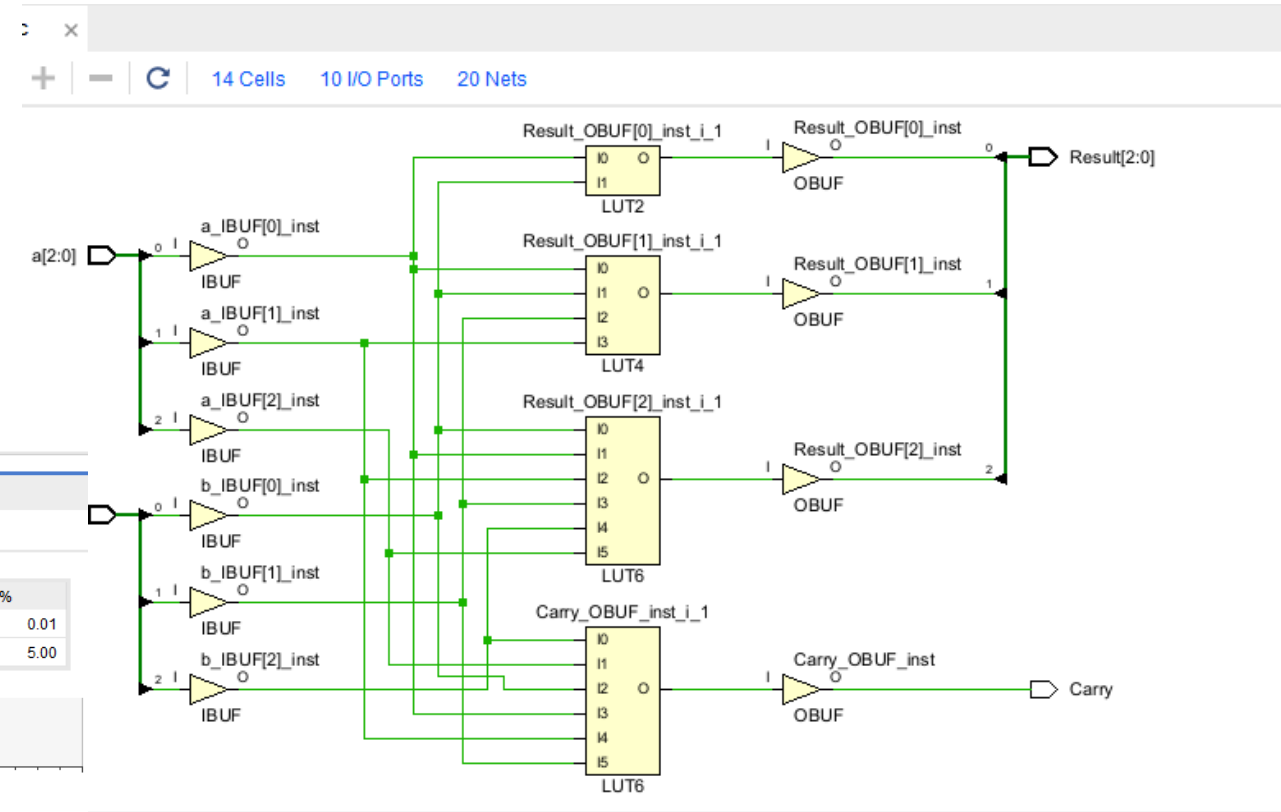
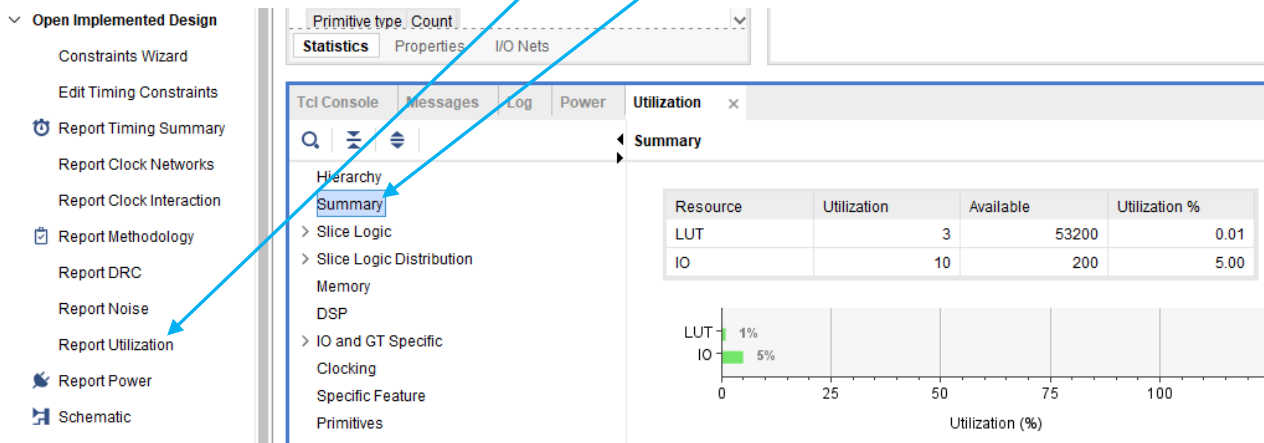
Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Logic %	Net %	Requirement	Source Clock	Destination Clock	Exception	Skew	Clock Uncertainty
Path 1	∞	3	2		a[1]	Result[1]	2.972	1.497	1.475	50.4	49.6	-∞	input port clock				0.000
Path 2	∞	3	2		b[2]	Result[2]	3.045	1.494	1.550	49.1	50.9	-∞	input port clock				0.000
Path 3	∞	3	2		a[0]	Result[0]	3.283	1.586	1.697	48.3	51.7	-∞	input port clock				0.000
Path 4	∞	3	2		a[2]	Carry	3.864	1.472	2.392	38.1	61.9	-∞	input port clock				0.000

# 2ο Εργαστηριακό Μάθημα

## Implementation – Report Utilization

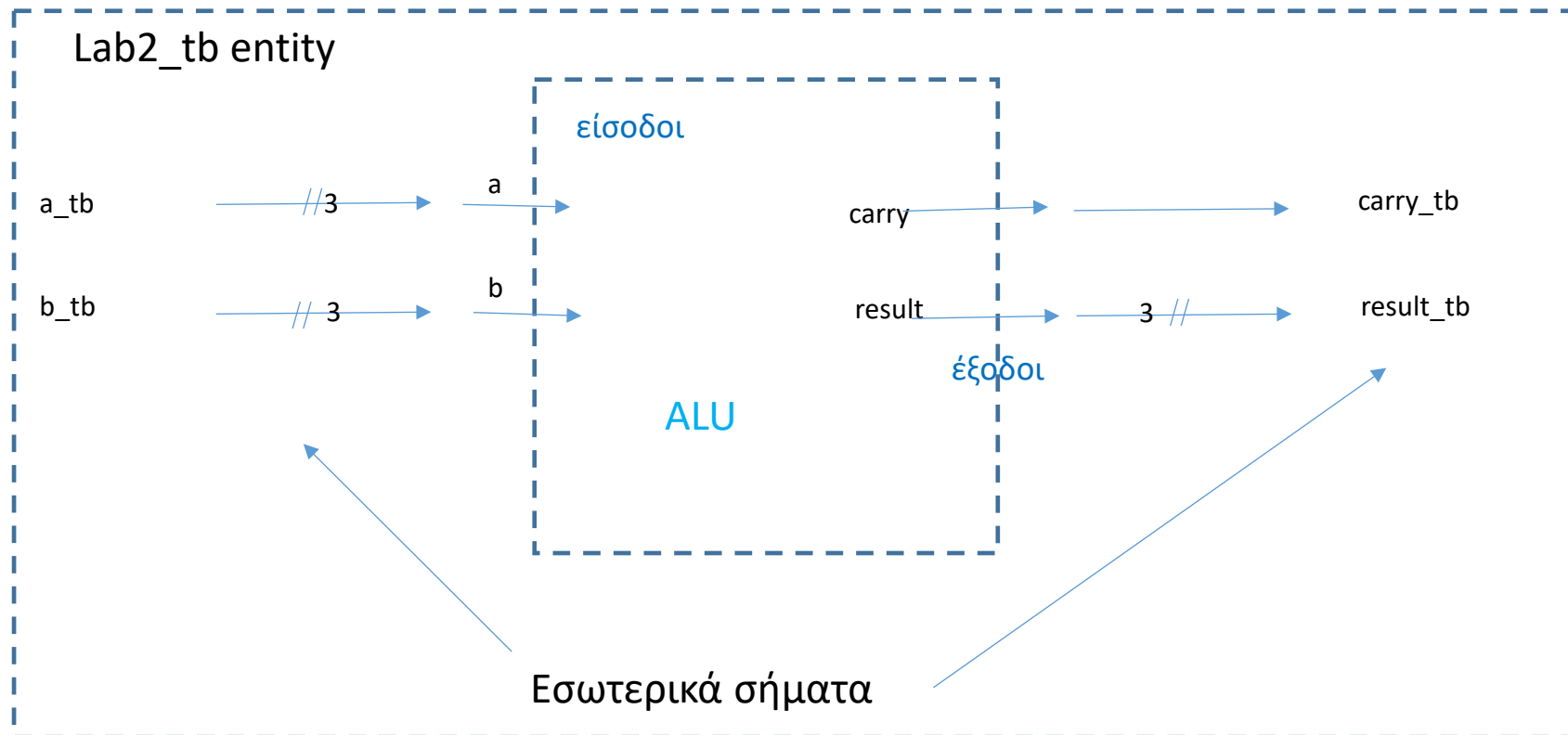
Open Implemented Design->Report Utilization->Summary  
αλλά και από Project Summary

Διαφορετικά Lut στο report από το Design (3 αντί για 4).  
Κάποια συνενώνονται στο τελικό αποτέλεσμα  
σηματίζοντας Ζεύγη (lutpairs).



# 2ο Εργαστηριακό Μάθημα

## Βήμα 8: Simulation



# 2ο Εργαστηριακό Μάθημα

## Βήμα 8α: Simulation

```
entity ALU_tb is
-- Port ( );
end ALU_tb;

architecture Behavioral of ALU_tb is
component ALU is
    Port ( a      : in STD_LOGIC_VECTOR (2 downto 0);
          b      : in STD_LOGIC_VECTOR (2 downto 0);
          Result  : out STD_LOGIC_VECTOR (2 downto 0);
          Carry   : out STD_LOGIC
        );
end component ALU;

signal a_tb      : STD_LOGIC_VECTOR (2 downto 0);
signal b_tb      : STD_LOGIC_VECTOR (2 downto 0);
signal Result_tb : STD_LOGIC_VECTOR (2 downto 0);
signal Carry_tb  : STD_LOGIC;

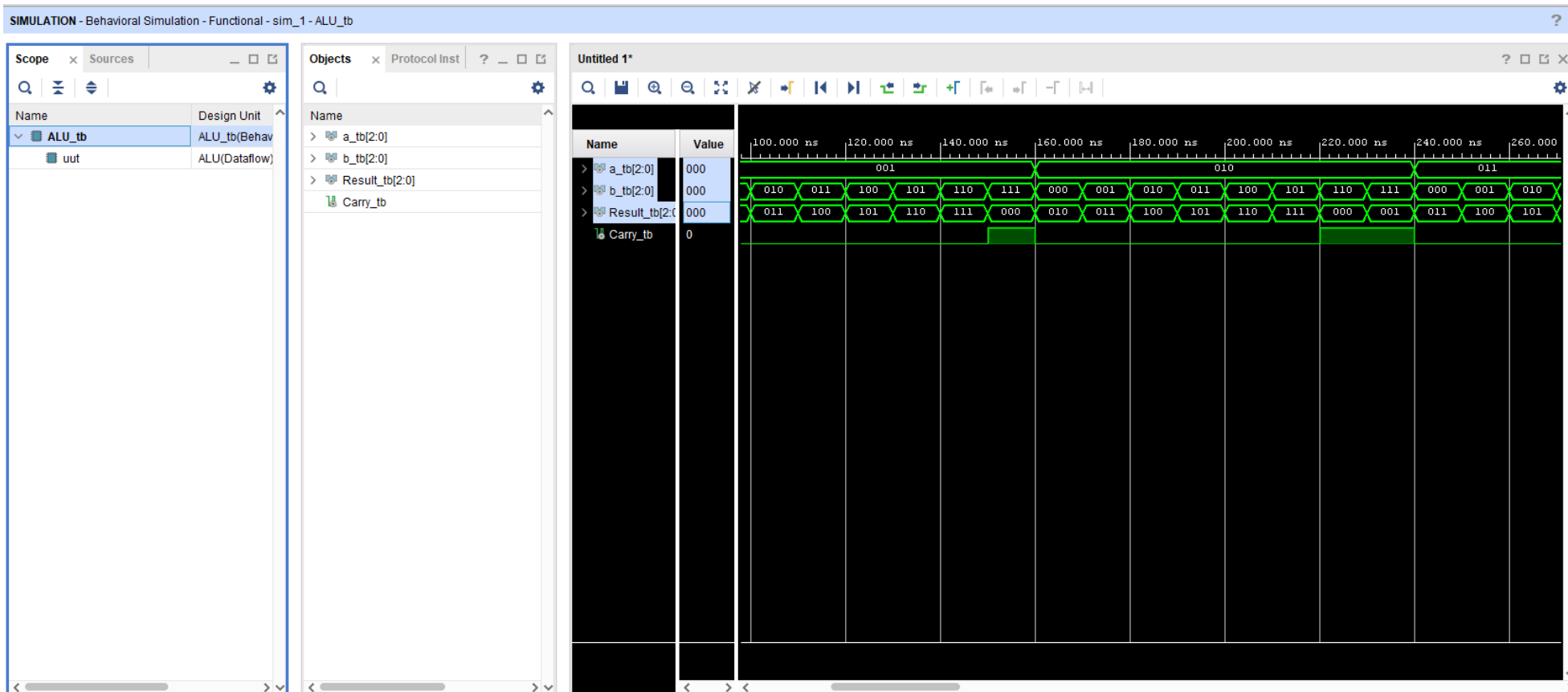
Begin
 uut: ALU port map (a_tb,b_tb,Result_tb, Carry_tb);

test: process is
Begin
for i in 0 to 7 loop
    a_tb<=std_logic_vector(to_signed(i,a_tb'length));
    for j in 0 to 7 loop
        b_tb<=std_logic_vector(to_signed(j,a_tb'length));
        wait for 10ns;
    end loop;
end loop;

end process test;
end architecture Behavioral;
```

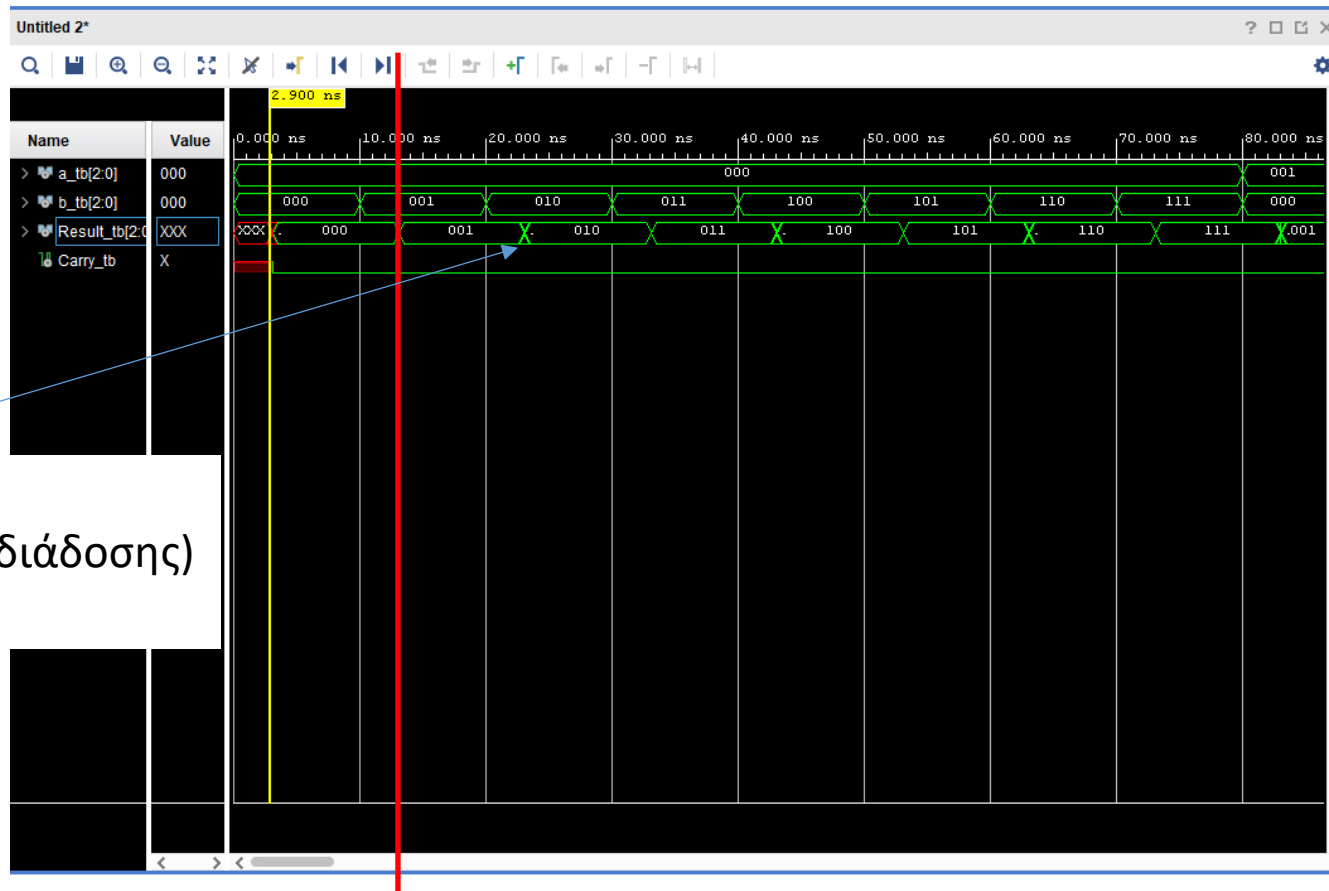
# 2ο Εργαστηριακό Μάθημα

## Βήμα 8b: Simulation: Behavioral



# 2ο Εργαστηριακό Μάθημα

## Βήμα 8c: Simulation: Post Synthesis Timing Simulation (1)

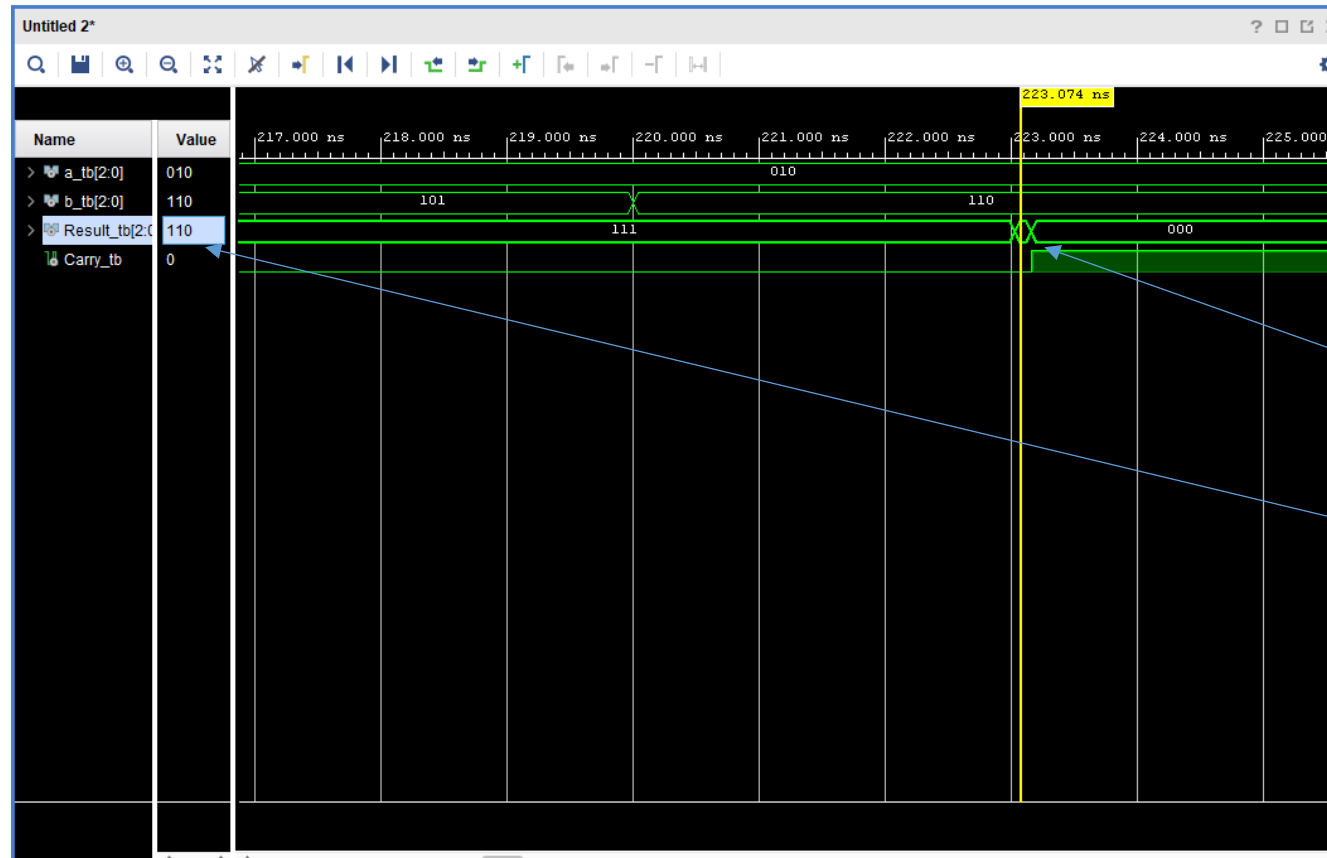


Delays (Καθυστερήσεις διάδοσης)



# 2ο Εργαστηριακό Μάθημα

## Βήμα 8c: Simulation: Post Synthesis Timing Simulation (2)



Λάθος τιμή λόγω  
Delays (Καθυστερήσεις μόλυνσης)