



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

Ψηφιακή Σχεδίαση

Ενότητα: ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ Νο: 12

Δρ. Αλέξανδρος Λαζαρίδης

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα του Πανεπιστημίου Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Περιεχόμενα

1. Σκοπός της άσκησης 4
2. Ερωτήσεις/Ασκήσεις..... 4

1. Σκοπός της άσκησης

- Περιγραφή απλών συνδυαστικών κυκλωμάτων

2. Ερωτήσεις/Ασκήσεις

Το παρακάτω πρόγραμμα προσομοιώνει μια πύλη AND δύο εισόδων

```
library IEEE;

    use IEEE.std_logic_1164.all

    USE ieee.std_logic_arith.all

    USE ieee.std_logic_unsigned.all;

    use ieee.std_logic_textio.all; --in order to use hread ()

library work;

use work all;


---


entity AND_gate is

    port(

        x: in std_logic;

        y: in std_logic;

        z: out std_logic

    );

end AND_gate;


---


architecture rtl of AND_gate is

begin

process (x, y)

    begin

        z<=x AND y;

    end process;

end architecture;
```

```
end rtl;
```

Τα προγράμματα της VHDL αποτελούνται από 3 τμήματα:

- Το τμήμα δήλωσης των βιβλιοθηκών (library).
- Το τμήμα δήλωσης διεπαφών (εισόδων και εξόδων) του κυκλώματος (entity)
- Το τμήμα δήλωσης της συμπεριφοράς – αρχιτεκτονικής του κυκλώματος

Επαλήθευση ορθής λειτουργίας του κυκλώματος

Για να επαλυθέψουμε την λειτουργία ενός κυκλώματος χρησιμοποιούμε ένα αρχείο δοκιμών που ονομάζεται test-bench. (Το αρχείο δοκιμών συνήθως ονομάζεται όπως και το αρχείο που δηλώνει το κύκλωμα προσθέτοντας την ένδειξη _TB στο τέλος.

Παράδειγμα: AND_gate_TB.vhd)

Το αρχείο δοκιμών έχει την ακόλουθη μορφή:

```
library IEEE;
```

```
use IEEE.std_logic_1164.all;
```

```
use IEEE.std_logic_unsigned.all;
```

```
use ieee.std_logic_arith.all;
```

```
use IEEE.std_logic_textio.all;
```

```
use ieee.std_logic_arith.all;
```

```
library work;
```

```
use work.all;
```

```
entity AND_gate_tb is
```

```
end AND_gate_tb;
```

```
architecture AND_gate_tb_a of AND_gate_tb is
```

```
    component AND_gate is
```

```
        port (
```

```
            x: in std_logic;
```

```
            y: in std_logic;
```

```

        z: out std_logic
    );
end component;

signal X,y,z: std_logic;

begi
AND_gate_inst: AND_gate
    Port map(
        x,y,z );

```

```

stimulus_proc : process is
begin
x<='0'; y<='0';
wait for 12ns;
x<='1'; y<='0';
wait for 2ns;
x<='1'; y<='1';
wait for 3ns;
x<='0'; y<='1';
wait for 7ns;
x<='0'; y<='0';
wait for 2ns;
    wait;
end process stimulus_proc;
end architecture AND_gate_tb_a;

```

To test bench αποτελείται από 4 μέρη

- Δήλωση βιβλιοθηκών
- Δήλωση οντότητας (παρατηρείστε ότι δεν υπάρχουν είσοδοι και έξοδοι αφού δεν πρόκειται για κύκλωμα με εισόδους και εξόδους αλλά για ένα περιβάλλον δοκιμής)
- Την αρχιτεκτονική η οποία αποτελείτε από 2 τμήματα

- Τμήμα δήλωσης του κυκλώματος που θέλουμε να ελέγξουμε και τον ονομάτων που δίνουμε στα σήματα αυτού
- Τους συνδυασμούς εισόδων που θέλουμε να δοκιμάσουμε.

Όταν θέλουμε να ελέγξουμε ένα συδυαστικό κύκλωμα 2 εισόδων δοκιμάζουμε και τους 4 δυνατούς συνδυασμούς των εισόδων, δηλαδή τον πίνακα αληθείας.

Ο έλεγχος ορθής λειτουργίας γίνεται με την βοήθεια εργαλείων προσομοίωσης. Ένα τέτοιο εργαλείο είναι το Modelsim.

1. Κάνουμε συντακτικό έλεγχο του αρχείου που περιγράφει το κυκλωμα. (επιλέγουμε το αρχείο με δεξί click επιλέγουμε "compile".
2. Στο κάτω τμήμα της οθόνης (παράθυρο transcript) με πράσινα γράμματα φαίνεται το επιτυχές αποτέλεσμα του συντακτικού ελέγχου ενώ με κόκκινα γράμματα αν η προσπάθεια δεν ήταν επιτυχής. Στη δεύτερη περίπτωση πρέπει να κάνουμε διπλό click στα κόκκινα γράμματα για να μας δώσει περισσότερες λεπτομέρειες για τα λάθη μας. Οπότε
 - a. Ανοίγουμε το αρχείο περιγραφής του κώδικα
 - b. Πηγαίνουμε στην γραμμή που μας υποδεικνύει το Modelsim και διορθώνουμε το λάθος
 - c. Αποθηκεύουμε το αρχείο και
 - d. Επαναλαμβάνουμε το συντακτικό έλεγχο
3. Από το menu του Modelsim επιλέγουμε simulate/startsimulation/design και επιλέγουμε work/and_gate_tb
4. Από το παράθυρο objects επιλέγουμε όλα τα σήματα και με δεξί click επιλέγουμε add to wave.
5. Μεγιστοποιούμε το παράθυρο wave και επιλέγουμε run. Τώρα μπορούμε να δούμε στην οθόνη πως αλλάζει η έξοδος z για διαφορετικές τιμές των εισόδων, όπως θα βλέπαμε και στο led στον πάγκο.