



**ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

---

## **Ενσωματωμένα Συστήματα**

**Ενότητα:** ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ Νο 13

Δρ. Μηνάς Δασυγένης

[mdasyg@ieee.org](mailto:mdasyg@ieee.org)

**Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών**

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

<http://arch.ict.e.uowm.gr/mdasyg>

---

## 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



πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

## Περιεχόμενα

1. Σκοπός της άσκησης .....	4
2. Παραδοτέα .....	4
3. Δοκιμή του ελεγκτή UART.....	5
4. Σύνδεση του ελεγκτή UART στο ricoblaze .....	6

## 1. Σκοπός της άσκησης

- Σχεδίαση Mixed Language Design (**Verilog+VHDL**).
- Προσθήκη ενός ελεγκτή UART I/O.
- Κατασκευή του υλικού (**System on Chip**) και του λογισμικού (*ρουτίνα εξυπηρέτησης διακοπής*).

### (A) 4 ερωτήσεις

### (C) 6 ασκήσεις/προγράμματα

### 1 bonus

## 2. Παραδοτέα

- **Παραδοτέο C1:** Αρχεία VHDL.
- **Παραδοτέο C2:** Αρχεία bitstream.
- **Παραδοτέο C3:** Αρχεία UDF.
- **Παραδοτέο A1:** Screenshot με το schematic στο οποίο φαίνεται το component RS232\_test.
- **Παραδοτέο A2:** Screenshot με το παράθυρο του τερματικού που φαίνεται ο αντικατροπτισμός.
- **Παραδοτέο C4:** Αρχεία VHDL.
- **Παραδοτέο C5:** Αρχεία bitstream.
- **Παραδοτέο C6:** Αρχεία UDF.
- **Παραδοτέο A3:** Screenshot με το schematic στο οποίο φαίνεται η σύνδεση του επεξεργαστή ricoblaze με τον ελεγκτή uart.
- **Παραδοτέο A4:** Screenshot με το παράθυρο του τερματικού.

### **Σε αυτό το εργαστήριο:**

θα προσθέσετε τον ελεγκτή **UART** (*Universal asynchronous receiver/transmitter*) σειριακής θύρας στο ricoblaze, ώστε να μπορεί να επικοινωνήσει ο επεξεργαστής με έναν υπολογιστή μέσω της σειριακής θύρας. Η σειριακή θύρα θα ρυθμιστεί στην ταχύτητα 9600, 8 bit μετάδοσης, χωρίς ισοτιμία (No), 1 bit τερματισμού. Συνοπτικά: 9600,8,N,1 (ή σε ρύθμιση που θα ορίσετε εσείς).

Θα χρησιμοποιήσετε και τον κώδικα εμφάνισης στο LCD που έχετε αναπτύξει σε προηγούμενο εργαστήριο.

Θα εμφανίζεται μια προτροπή αυθεντικοποίησης από το ricoblaze στη σειριακή θύρα (**login:**)

Μέσω της σειριακής θύρας ο χρήστης θα πληκτρολογεί ένα όνομα χρήστη.

Θα εμφανίζεται η προτροπή για τον κωδικό (**password:**)

Μέσω της σειριακής θύρας ο χρήστης θα πληκτρολογεί τον κωδικό του.

Αν το ζευγάρι όνομα χρήστη/κωδικός είναι σωστό θα εμφανίζεται και στη σειριακή θύρα και στο LCD, το μήνυμα "**Succeeded!**", ενώ σε διαφορετική περίπτωση θα εμφανίζεται το μήνυμα "**Failed!**".

### 3. Δοκιμή του ελεγκτή UART

- Μεταφορτώστε από τα έγγραφα του εργαστηρίου το αρχείο **mmuart\_latest.tar.gz**, το οποίο υλοποιεί έναν απλό ελεγκτή σειριακής θύρας (χωρίς σήματα ελέγχου) και έχει αναπτυχθεί στο project <http://opencores.org/project,mmuart>.
- Ο ελεγκτής είναι γραμμένος σε γλώσσα περιγραφής υλικού verilog. Θα χρησιμοποιήσετε το αρχείο "**uart\_transceiver.v**" το οποίο υλοποιεί τον πυρήνα του ελεγκτή.
- Μεταφορτώστε το αρχείο "**uart\_top.vhd**" το οποίο περιγράφει σε VHDL το κύκλωμα που θα χρησιμοποιήσει το **uart\_transceiver**.
- Δημιουργήστε ένα νέο project στο ISE, το οποίο θα είναι τύπου mixed language design (αν σας γίνει αυτή η ερώτηση). Προκειμένου να χρησιμοποιήσουμε τη σχεδίαση "mixed language design", δηλαδή να χρησιμοποιήσουμε ένα module που έχει γραφεί σε Verilog σε μια άλλη γλώσσα περιγραφής υλικού όπως τη VHDL, θα πρέπει να ισχύουν τα παρακάτω (από τον οδηγό χρήσης του Xilinx ISE):

#### To Instantiate a Verilog Module in a VHDL Design Unit

- Declare a VHDL component with the same name as the Verilog module (*respecting case sensitivity*) that you want to instantiate. For example,

```
COMPONENT FD PORT (  
  Q : out  STD_ULOGIC;  
  D : in   STD_ULOGIC;  
  C : in   STD_ULOGIC );  
END COMPONENT;
```

- Use named association to instantiate the Verilog module. For example,

```
UUT : FD PORT MAP (  
  Q => O,  D => I,  C => CLK );
```

Since Verilog is **case sensitive**, named associations and the local port names that you use in the component declaration must match the case of the corresponding Verilog port names.

- Παρατηρήστε ότι το αρχείο VHDL που σας δίνεται, χρησιμοποιεί το αρχείο verilog σύμφωνα με τις παραπάνω οδηγίες. Το αρχείο `uart_top.vhdl` εκτελεί τη λειτουργία αντικατοπτρισμού, δηλαδή διαβάζει ένα χαρακτήρα από τη σειριακή και στέλνει πάλι τον ίδιο πίσω.
- Να δημιουργήσετε το κατάλληλο constrain file, ώστε να συνδεθούν τα ports του RS232\_test (δηλαδή τα `clk_50MHz`, `rst`, `RS232_rx`, `RS232_tx`, `LED`) στο ρολόι, σε ένα κουμπί που θα λειτουργεί ως reset, στο pin rx της σειριακής, στο pin tx της σειριακής, και στα 8 LED της πλακέτας μας.
- Συνθέστε το project, συνδέστε ένα σειριακό καλώδιο και επιβεβαιώστε την ορθή λειτουργία. Θα πρέπει μόλις πατάτε ένα χαρακτήρα στη σειριακή θύρα να εμφανίζεται ξανά ο ίδιος.

**Παραδοτέα :** Αρχεία vhdl (C1), bitstream (C2), ucf (C3)

**Παραδοτέα :** Screenshots

→ το schematic στο οποίο φαίνεται το component RS232\_test (A1).

→ το παράθυρο του τερματικού που φαίνεται ο αντικατοπτρισμός (A2).

## 4. Σύνδεση του ελεγκτή UART στο picoblaze

- [Hardware] Αφού επιβεβαιώστε την ορθή λειτουργία του UART, συνδέστε το UART σε μια θύρα εξόδου και σε μια θύρα εισόδου του pBlaze.
- Συνδέστε τα υπόλοιπα περιφερειακά της άσκησης που αναφέρονται στην πρώτη σελίδα.
- Να έχετε υπόψιν ότι:
- η ανάγνωση από τον επεξεργαστή πρέπει να γίνεται όταν επιλεχθεί το σωστό port και όταν το σήμα read\_strobe πάρει την τιμή 1. Παράδειγμα:

```
signal_a12<= '1' when port_id = x"BB" and read_strobe='1'
else '0';
```

- η εγγραφή από τον επεξεργαστή γίνεται με παρόμοιο τρόπο χρησιμοποιώντας το σήμα write\_strobe. Παράδειγμα:

```
leds_reg <= out_port when rising_edge(clk) and (port_id =  
x"CC") and (write_strobe='1');
```

- Τροποποιήστε κατάλληλα το **uart\_top.vhdl**, ώστε:
- να μη γράφει κάτι πίσω στη σειριακή αν δε πάρει εντολή από τον επεξεργαστή.
- να υπάρχει μια ακόμη output port στο declaration ώστε να στέλνει ένα interrupt στον επεξεργαστή, όταν λαμβάνει ένα χαρακτήρα.
- Να υπάρχει μια ακόμη input port, ώστε όταν λαμβάνει ένα write strobe σήμα από τον επεξεργαστή να γράφει στη σειριακή θύρα το χαρακτήρα.
- Διαγράψτε τον κώδικα που δε χρησιμοποιείται (π.χ. η εγγραφή σε LED).
- **[Software]** Δημιουργήστε μια συνάρτηση χειρισμού interrupt, η οποία θα καλείται όταν ενεργοποιείται το interrupt από το σειριακό ελεγκτή, ο οποίος θα διαβάζει τον χαρακτήρα από τη σειριακή θύρα και θα τον τοποθετεί σε μια θέση μνήμης. Η συνάρτηση αυτή θα αποθηκεύει τους 8 χαρακτήρες σε 8 θέσεις μνήμης. Όταν διαβαστούν 8 χαρακτήρες, θα τροποποιεί έναν καταχωρητή που θα επιλέξετε εσείς (π.χ. θα γράφει 11001100) για να σηματοδοτήσει στο κυρίως πρόγραμμα ότι έχουν διαβαστεί 8 χαρακτήρες. Το κυρίως πρόγραμμα μπορεί να είναι σε ένα βρόχο που θα ελέγχει συνεχώς αυτόν τον καταχωρητή με τη τιμή 11001100 και αν δεν έχει πάρει αυτή την τιμή θα πηγαίνει συνεχώς λίγες γραμμές πιο πάνω (βρόχος επανάληψης).
- Ο επεξεργαστής θα εκτυπώνει στη σειριακή θύρα την προτροπή και θα περιμένει. Όταν σηματοδοτηθεί ότι έχουν διαβαστεί 8 χαρακτήρες, θα διαβάζει τους 8 χαρακτήρες από τη συγκεκριμένη θέση στη μνήμη και θα τους συγκρίνει με ένα συγκεκριμένο string που θα έχετε δώσει εσείς στη assembly. Τέλος, θα εκτυπώνεται το κατάλληλο μήνυμα επιτυχίας ή όχι.

**Παραδοτέα :** Αρχεία vhd (C4), bitstream (C5), ucf (C6)

**Παραδοτέα :** Screenshots

→ το schematic στο οποίο φαίνεται η σύνδεση του επεξεργαστή ricoblaze με τον ελεγκτή uart (A3)

→ το παράθυρο του τερματικού (A4).

### **BONUS1 +0.5 μονάδες:**

Αντικατάσταση του απλού ελεγκτή UART της προηγούμενης άσκησης με τον πιο σύνθετο ελεγκτή UART που βρίσκεται στο αρχείο **extended\_uart.zip**. Ο ελεγκτής χρησιμοποιεί ενδιάμεσες μνήμες buffer των 16 θέσεων, προκειμένου να ανταποκρίνεται καλύτερα σε απαιτητικές υλοποιήσεις. Θα πρέπει να μελετήσετε το αρχείο **UART\_Manual.pdf** που περιλαμβάνεται στο zip.

Μπορείτε να χρησιμοποιήσετε το αρχείο **clockdivider.vhd** για να δημιουργήσετε τα σήματα χρονισμού.

#### **Παραδοτέα bonus:**

→ Αρχεία vhd, bitstream, ucf

→ Screenshot με το schematic στο οποίο φαίνεται η σύνδεση του επεξεργαστή riscoblaze με τον προχωρημένο ελεγκτή uart.