

# HY220: Εργαστήριο Ψηφιακών Κυκλωμάτων

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης  
Χειμερινό Εξάμηνο 2014

## Εργαστήριο 4: Υλοποίηση *Ελεγκτή Μνήμης*

**17 Νοεμβρίου έως 28 Νοεμβρίου 2014**

### 1.1 Σκοπός της Εργαστηριακής Άσκησης

Σε αυτήν την Άσκηση θα επιχειρήσουμε να δημιουργήσουμε κύκλωμα που να υλοποιεί ένα ελεγκτή μνήμης (256 θέσεις, με πλάτος 8-bit). Ο ελεγκτής μνήμης στην συνέχεια πρέπει να συνδεθεί με ένα κύκλωμα σειριακής επικοινωνίας (UART) κύκλωμα (βλ. παρακάτω σχήμα), το οποίο προωθεί εντολές για ανάγνωση και εγγραφή δεδομένων μέσω σειριακής σύνδεσης.

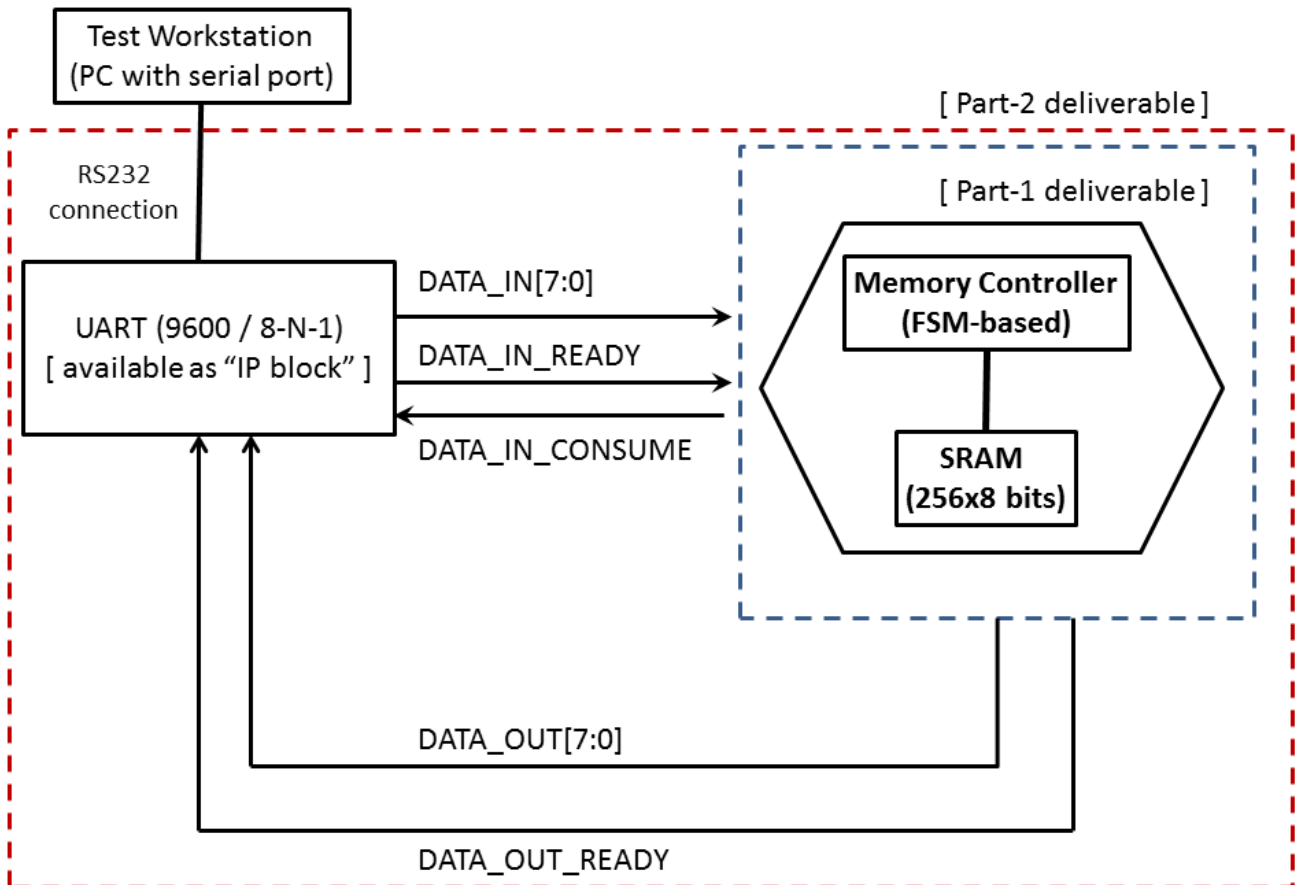


Figure 1: Σύστημα ελέγχου μνήμης (256 θέσεις, 8-bit δεδομένα), συνδεδεμένο με σειριακή διεπαφή (interface).

Τα δεδομένα από το σειριακό τερματικό δίδονται ως ακολουθίες χαρακτήρων, και αντίστοιχα στέλνονται τα αποτελέσματα στο σειριακό τερματικό. Με χρήση του σειριακού τερματικού θα πρέπει να επιδείξετε την λειτουργία του συστήματος αυτού.

## 1.2 Σχεδιασμός Συστήματος

Πρώτο βήμα για την υλοποίηση της εργασίας είναι ο σχεδιασμός, στο χαρτί ή και σε ηλεκτρονική μορφή, το κύκλωμα που υλοποιεί την λογική ελέγχου μνήμης, κάνοντας χρήση του BRAM block που παράγεται από το εργαλείο σχεδίασης Xilinx ISE. Για την λογική ελέγχου της μνήμης θα χρειαστεί να σχεδιάσετε κατάλληλη μηχανή πεπερασμένων καταστάσεων (Finite State Machine - FSM). Το κύκλωμα ελέγχου μνήμης θα πρέπει να ενσωματωθεί σε περιβάλλον προσομοίωσης, στο οποίο με χρήση “test vectors” θα προσομοιώσετε τα σήματα εισόδου και εξόδου του UART.

Στο επόμενο (και τελικό) βήμα, το κύκλωμα ελέγχου μνήμης θα πρέπει να συνδυαστεί με πραγματικό κύκλωμα UART (το οποίο θα δοθεί έτοιμο για χρήση την 2<sup>η</sup> εβδομάδα αυτής της εργαστηριακής άσκησης). Το UART προωθεί εντολές εγγραφής και ανάγνωσης από ένα σειριακό τερματικό. Οι εντολές δίδονται από το σειριακό τερματικό ως ακολουθίες χαρακτήρων ASCII, με το εξής συντακτικό:

- Εγγραφή: **W**<space><Address><space><Data><CR>
- Ανάγνωση: **R**<space><Address><CR>

Οι παράμετροι <Address> και <Data> δίδονται ως 2 δεκαεξαδικά ψηφία. Η κάθε εντολή τερματίζεται με τον ειδικό χαρακτήρα carriage return (<CR>).

## 1.3 Συγγραφή του Κώδικα Verilog και Προσομοίωση

Αφού ετοιμάσατε το σχέδιο του κυκλώματος, θα πρέπει να το μετατρέψετε σε κώδικα Verilog. Αμέσως μετά πρέπει να δημιουργήσετε ένα testbench module με το οποίο θα δοκιμάσετε να προσομοιώσετε το σύστημα. Για την παρούσα άσκηση, η παράδοση θα γίνει σε δύο βήματα:

1. Κύκλωμα ελέγχου μνήμης, με εξομοίωση του συστήματος προώθησης εντολών και αποτελεσμάτων από/προς το σειριακό τερματικό.
2. Διασύνδεση του κυκλώματος ελέγχου μνήμης με το κύκλωμα UART που θα σας δοθεί για την επικοινωνία με τον σταθμό ελέγχου.

## 1.4 Δημιουργία UCF και Τοποθέτηση

Αφού ολοκληρώσετε τα προηγούμενα βήματα, πρέπει να δημιουργήσετε το Αρχείο Περιορισμών Χρήστη (User Constraints File – UCF) δηλώνοντας τις εισόδους και τις εξόδους που θα χρησιμοποιήσετε στην πλακέτα και την τάση που θα τους ασκήσετε, όπως δείχνει το ucf αρχείο που λάβατε στο εργαστήριο 0 το οποίο είναι κομμάτι του master ucf που έχει δοθεί από τους κατασκευαστές και υπάρχει στο Documentation της πλακέτας στη σελίδα: [http://www.csd.uoc.gr/~hy220/2014f/Basys2\\_100\\_250General.ucf](http://www.csd.uoc.gr/~hy220/2014f/Basys2_100_250General.ucf).

## 1.5 Παράδοση

Θα δείξετε τις post-RTL και post-PnR κυματομορφές στους βοηθούς του εργαστηρίου, την λειτουργία του κυκλώματος στην FPGA, και θα παραδώσετε (εκτυπωμένη ή χειρόγραφη) την αναφορά που θα περιέχει (i) την περιγραφή του κυκλώματος σε κείμενο και σε block διάγραμμα (ii) την FSM (iii) και τον κώδικα της προσομοίωσης (testbench).