HY220: Εργαστήριο Ψηφιακών Κυκλωμάτων

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης
Χειμερινό Εξάμηνο 2014

Εργαστήριο 3:
Υλοποίηση *Ελεγκτή Ουράς Δεδομένων*

27 Οκτωβρίου έως 14 Νοεμβρίου 2014

1.1 Σκοπός της Εργαστηριακής Άσκησης

Σε αυτήν την Άσκηση θα επιχειρήσουμε να δημιουργήσουμε κύκλωμα που να υλοποιεί ένα ελεγκτή ουράς δεδομένων (FIFO - 256 θέσεις, με πλάτος 8-bit). Ο ελεγκτής 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.

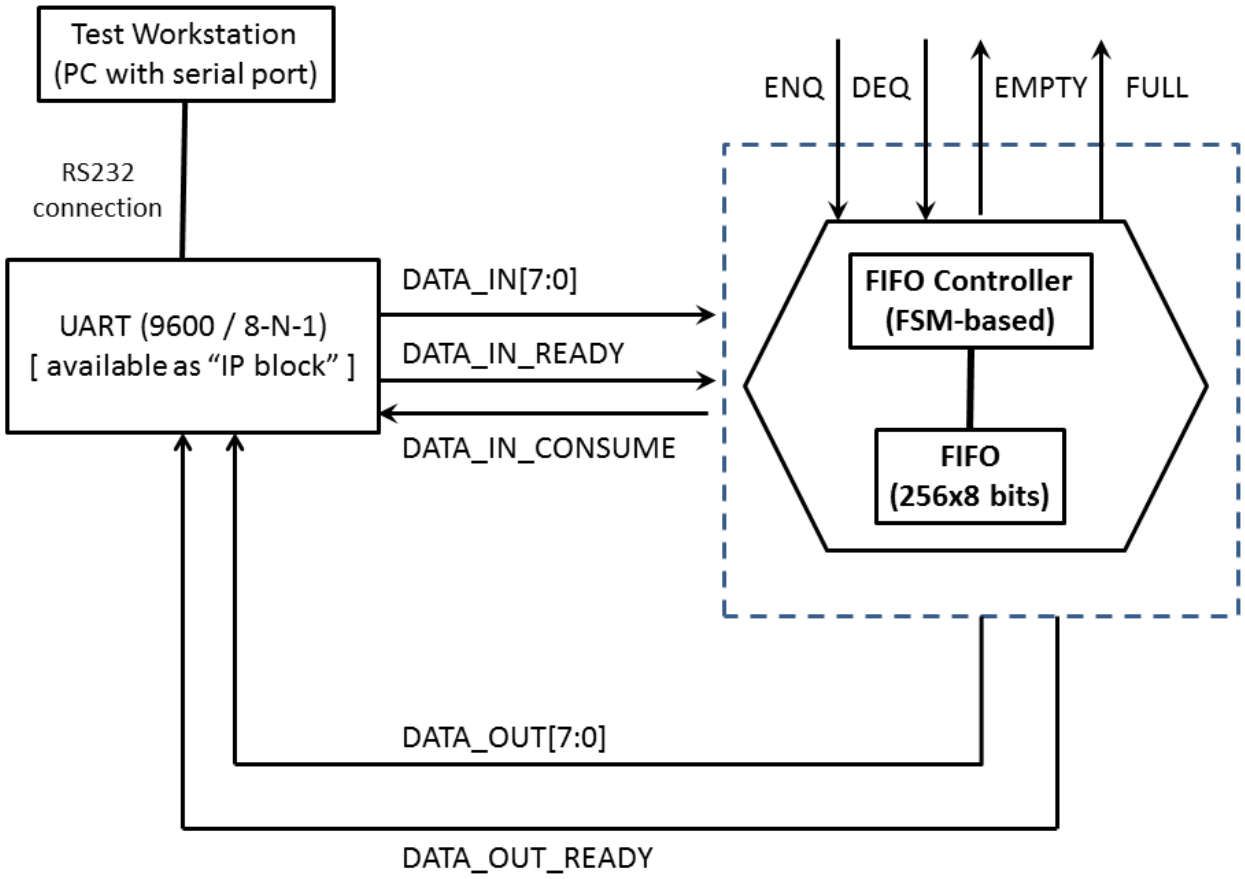

Figure 1: Σύστημα ελέγχου ουράς δεδομένων (256 θέσεις, 8-bit δεδομένα), συνδεδεμένο με σειριακή διεπαφή.

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 ουράς δεδομένων, κάνοντας χρήση του FIFO block που παράγεται από το εργαλείο σχεδίασης Xilinx ISE. Για την λογική ελέγχου της ουράς δεδομένων θα χρειαστεί να σχεδιάσετε κατάλληλες μηχανές πεπερασμένων καταστάσεων (Finite State Machines - FSMs) για την εισαγωγή και την εξαγωγή δεδομένων στην ουρά.

Στο επόμενο (και τελικό) βήμα, το κύκλωμα ελέγχου ουράς δεδομένων θα πρέπει να συνδυαστεί με πραγματικό κύκλωμα UART (το οποίο θα δοθεί έτοιμο για χρήση). Το UART προωθεί δεδομένα από ένα σειριακό τερματικό προς την ουρά δεδομένων - λειτουργία ENQueue, και αντίστοιχα προωθεί δεδομένα από την ουρά δεδομένων προς το σειριακό τερματικό όταν ο χρήστης ενεργοποιήσει την λειτουργία DEQueue. Η λειτουργία DEQueue ενεργοποιείται με το πάτημα πλήκτρου.

Η ουρά δεδομένων παρέχει ένδειξη για το εαν είναι άδεια (EMPTY) ή πλήρης (FULL). Υπο αυτές τις συνθήκες, οι λειτουργίες DEQueue και ENQueue δεν μπορούν να ολοκληρωθούν. Οι ενδείξεις EMPTY, FULL πρέπει να γίνονται ορατές με την χρήση LEDs.

Τα δεδομένα εκφράζονται ως 2 δεκαεξαδικά ψηφία (συνολικά 8 bits πληροφορίας), και τερματίζονται 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.

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.

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).