HY220: Εργαστήριο Ψηφιακών Κυκλωμάτων

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

Χειμερινό Εξάμηνο 2016

Εργαστήριο 2:
Υλοποίηση Ελεγκτή Ουράς Δεδομένων

14 Νοεμβρίου έως 27 Νοεμβρίου 2016 (2 εβδομάδες)

1.1 Σκοπός της Εργαστηριακής Άσκησης

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

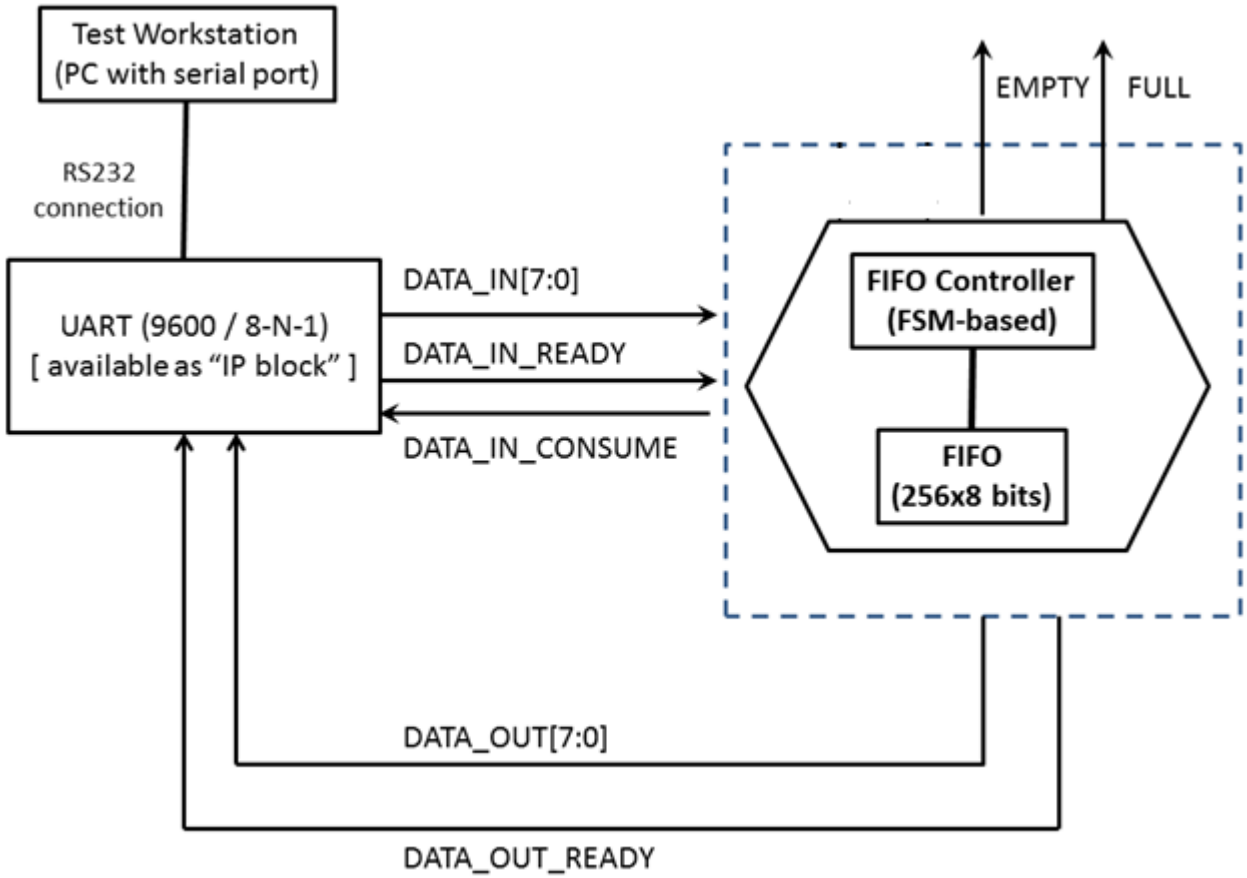

Figure Error! No sequence specified.**: Σύστημα ελέγχου ουράς δεδομένων (256 θέσεις, 8-bit δεδομένα), συνδεδεμένο με σειριακή διεπαφή.**

Οι εντολές εγγραφής ή ανάγνωσης και τα δεδομένα που περιέχουν αυτές αν είναι εντολές εγγραφής, δίδονται από το σειριακό τερματικό ως ακολουθίες χαρακτήρων, τους παραλαμβάνει το UART, τους αποθηκεύει στην ουρά του Receiver του έναν προς έναν και τους διαθέτει στον ελεγκτή που θα υλοποιήσετε σε αυτό τον εργαστήριο.

Στις περιπτώσεις που ο ελεγκτής θα δέχεται εντολή ανάγνωσης τα αποτελέσματα θα στέλνονται στο σειριακό τερματικό.
Με χρήση του σειριακού τερματικού θα πρέπει να επιδείξετε την λειτουργία του συστήματος αυτού.

1.2 Σχεδιασμός Συστήματος και Αναφορά

Πρώτο βήμα για την υλοποίηση της εργασίας είναι ή σύνταξη αναφοράς παρόμοιας με την πρότυπη αναφορά που έχει αναρτηθεί στον site του μαθήματος και πρέπει να περιγράφει κάθε βήμα της ανάπτυξης του συστήματός σας.

Για την λογική ελέγχου της ουράς δεδομένων (FIFO block) η οποία παράγεται από το εργαλείο σχεδίασης Xilinx ISE και ΔΕΝ χρειάζεται να την υλοποιήσετε εσείς, θα χρειαστεί να σχεδιάσετε κατάλληλη μηχανή πεπερασμένων καταστάσεων (Finite State Machines – FSMs) με την βοήθεια της οποίας ο ελεγκτής θα αναγνωρίζει τι είδους εντολή αντιπροσωπεύουν οι χαρακτήρες που έλαβε από το UART.
Οι εντολές δίδονται από το σειριακό τερματικό ως ακολουθίες χαρακτήρων ASCII, με το εξής συντακτικό:

- ENQ: <**W**><**SPACE**><**DATA**><**CR**>
- DEC: <**R**><**CR**>

Τα DATA είναι μία ποσότητα 8 bit που αναπαρίσταται από 2 δεκαεξαδικά ψηφία.
Επειδή το τερματικό 'στέλνει' και 'καταλαβαίνει' ASCII χαρακτήρες, ενώ εμείς αποθηκεύουμε 8bit ποσότητες στη FIFO, είναι προφανές ότι πρέπει να φτιάξετε έναν Decoder ο οποίος δέχεται ASCII χαρακτήρες και να μετατρέπει τον ASCII χαρακτήρα στον αριθμό που θέλουμε σύμφωνα με τη δεκαεξαδική αναπαράσταση. Δηλαδή θα παίρνει από την μία είσοδο τους χαρακτήρες ASCII 0-9 και A-F (κεφαλαία), και θα βγάζει αριθμό 0-15 στην έξοδο.
Επίσης πρέπει να φτιάξετε έναν Encoder που να κάνει την αντίστροφη δουλειά. Να παίρνει από τη μία είσοδο αριθμούς 0-15. Και από την έξοδο να βγάζει χαρακτήρες ASCII 0-9 και A-F (κεφαλαία).

Στην περίπτωση όπου ο ελεγκτής διαπιστώσει ότι διαχειρίζεται μία εντολή εγγραφής, πρέπει να αποθηκεύει το πεδίο <DATA> στην FIFO. Πχ, δίδεται η εντολή **W 40 <CR>**. Το W σηματοδοτεί εντολή εγγραφής (**ENQ**ueue). Η τιμή που πρέπει να αποθηκευτεί στη FIFO είναι το 0x40 =8'b0100000. ΟΧΙ τους χαρακτήρες ASCII '4'=8'b00110100 και '0'=8'b00110000.

Αντίστοιχα, στην περίπτωση όπου ο ελεγκτής διαπιστώσει ότι διαχειρίζεται μία εντολή ανάγνωσης, πρέπει να κάνει **DEQ**queue ενός byte από την FIFO και να στείλει στο UART τους κατάλληλους ASCII χαρακτήρες για εκτύπωση στο τερματικό. Το τελευταίο, θα αναλάβει να αποστείλει τους χαρακτήρες στο σειριακό τερματικό. Πχ, δίδεται η εντολή **R<CR>**. Η FIFO έχει μέσα data για κατανάλωση την τιμή 8'b10110001=0x71. Πρέπει να εκτυπωθεί στο τερματικό οι δύο ASCII χαρακτήρες '7' και '1'. ΟΧΙ ο ASCII χαρακτήρας 0x71= 'q'.

Η ουρά δεδομένων παρέχει ένδειξη για το εάν είναι άδεια (EMPTY) ή πλήρης (FULL). Υπό αυτές τις συνθήκες, οι λειτουργίες **DEQ**ueue και **ENQ**ueue αντίστοιχα δεν μπορούν να ολοκληρωθούν. Τα δεδομένα που γράφτηκαν μέχρι να προκύψει η διακοπή παραμένουν στην ουρά.
Οι ενδείξεις **EMPTY**, **FULL** πρέπει να γίνονται ορατές με την χρήση των LEDs οι οποίες βρίσκονται πάνω στο board BASYS2 και θα εντοπίσετε στο .ucf αρχείο για να τις ορίσετε ως έξοδο αυτών των σημάτων.

Στην προσομοίωση δεν χρειάζεται να περιλάβετε το UART, θα προσομοιώσετε τα σήματα εισόδου και εξόδου του UART με “test vectors".
Στο επόμενο βήμα, το κύκλωμα ελέγχου ουράς δεδομένων πρέπει να συνδυαστεί με πραγματικό κύκλωμα UART (το οποίο θα δοθεί έτοιμο για χρήση).

1.3 Συγγραφή του Κώδικα Verilog και Προσομοίωση

Αφού ολοκληρώσετε το implementation σκέλος της αναφοράς, 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.

1.5 Παράδοση

Την 1η εβδομάδα (14/11-20/11) θα παραδώσετε την αναφορά σας, και την 2η εβδομάδα (21/12-27/11) θα παραδώσετε την Verilog. Την 3η εβδομάδα θα εξεταστείτε ελέγχοντας το σύστημά σας στην πλακέτα.

Για την παράδοση χρησιμοποιείτε την εφαρμογή TURNIN από τα μηχανήματα linux του τμήματος. Γράφουμε turnin lab2-report@hy220 για την αναφορά ενώ για τον κώδικα turnin lab2-code.

Για τις ώρες της εξέτασης κλείνετε timeslot μέσω του Rendezvous.

ΥΠΕΝΘΥΜΙΣΗ!! :

ΠΑΡΑΔΙΔΕΤΕ ΜΑΖΙ ΜΕ ΟΛΑ ΤΑ ΑΡΧΕΙΑ ΠΟΥ ΠΕΡΙΕΧΟΥΝ ΤΟΝ VERILOG ΚΩΔΙΚΑ ΣΑΣ, ΕΝΑ ΑΡΧΕΙΟ ΤΟ ΟΠΟΙΟ ΠΕΡΙΕΧΕΙ ΤΟ TOP MODULE. ΤΟ TOP MODULE ΠΡΕΠΕΙ ΝΑ ΠΕΡΙΕΧΕΙ ΤΟΝ ΕΛΕΓΚΤΗ ΣΑΣ ΚΑΙ ΤΟ UART ΚΑΘΩΣ ΚΑΙ ΤΗΝ ΣΥΝΔΕΣΗ ΜΕΤΑΞΥ ΤΟΥΣ, ΔΙΟΤΙ ΑΥΤΟ ΕΙΝΑΙ ΤΟ ΟΛΟΚΛΗΡΩΜΕΝΟ ΣΥΣΤΗΜΑ ΣΤΟ ΟΠΟΙΟ ΘΑ ΕΞΕΤΑΣΤΕΙΤΕ.