ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ Η/Υ

ΜΑΘΗΜΑ: ΕΡΓΑΣΤΗΡΙΟ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΝΑΦΟΡΑ

Τίτλος: Παπαναστασίου Στέφανος 1608,21 Νοεμβρίου 2015

Περίληψη: Υλοποίηση Γενικού Ασύγχρονου Δέκτη, Αποστολέα (UART)

Εισαγωγή: Οι στόχοι της εργασίας αρχικά ήταν 4:

1. Δημιουργία Ελεγκτή Baud Rate
2. Δημιουργία UART Αποστολέα
3. Δημιουργία UART Δέκτη
4. Δημιουργία Συστήματος Αποστολέα-Δέκτη για Σειριακή Μεταφορά Δεδομένων

ΜΕΡΟΣ 1 << Ελεγκτής Baud Rate>

Υλοποίηση : Σε αυτό το μέρος χρησιμοποιήθηκε αρχικά ένα always block που ρυθμίζει την μέγιστη τιμή ενός μετρητή ανάλογα με το baud rate. Στη συνέχει χρησιμοποιείται ένας μετρητής που μετράει μέχρι την μέγιστη τιμή και μηδενίζεται. Για την εύρεση της μέγιστης τιμής χρησιμοποιήθηκε ο παρακάτω πίνακας που προέκυψε μέτα από τις κατάλληλες πράξεις.

Timi_metriti = (clock * 10^6)/baud_rate * 16

Baud Rate	Μετρητής	*Bit του μετρητή
300	10416.6666	13.34
1200	2604	11.34
4800	651.04166	9.34
9600	325.520833	8.32
19200	162.7604	7.33
38400	81.3802	6.33
57600	54.253472	5.75
115200	27.103	4.75

*Στην υλοποιήση χρησιμοποιήθηκαν 14 bits για να καλυφθεί ο μεγαλύτερος μετρητής

Επαλήθευση: Για την υλοποίηση σε επίπεδο συμπεριφοράς χρησιμοποιήθηκε ένα testbench που αρχικοποιούσε την τιμή του baud rate και έτρεχε το κύκλωμα με βάση το ρολόι

ΜΕΡΟΣ 2 << Υλοποίηση UART *Αποστολέα>>*

Υλοποίηση : Για τον αποστολέα χρησιμοποιήθηκε μία μηχανή τεσσάρων καταστάσεων:

00: Στέλνει το start bit και αρχικοποιεί κατάλληλα τις μεταβλητές

01: Στέλνει 8bits ,το μήνυμα που θέλει

10: Υπολογίζει το το parity bit και το στέλνει

11:Στέλνει το stop bit

Επαλήθευση: Για την υλοποίηση σε επίπεδο συμπεριφοράς χρησιμοποιήθηκε ένα testbench που αρχικοποιούσε το baud rate, ενεργοποιούσε τον αποστολέα και το μήνυμα που θα στείλει. Τελικά ο αποστολέας έστελνε start_bit,8-bit message,parity_bit,stop_bit ανάλογα με το baud rate

ΜΕΡΟΣ 3 << Υλοποίηση UART Δέκτη>>

Υλοποίηση : Για τον δέκτη χρησιμοποιήθηκε μία μηχανή τεσσάρων καταστάσεων:

00: Λαμβάνει το start bit και αρχικοποιεί κατάλληλα τις μεταβλητές

01: Λαμβάνει 8bits ,το μήνυμα που θέλει

10: Υπολογίζει το το parity bit και το συγκρίνει με αυτό που έλαβε

11:Λαμβάνε το stop bit

Ο δέκτης έχει παρόμοια δομή με τον αποστολέα , με την διαφορά ότι αυτός δέχεται δεδομένα και συγκρίνει για να ενημερώσει για τα κατάλληλα σφάλματα σε περίπτωση που υπάρχει κάποιο λάθος

Επαλήθευση: Για την υλοποίηση σε επίπεδο συμπεριφοράς χρησιμοποιήθηκε ένα testbench που αρχικοποιούσε το baud rate ενεργοποιούσε τον δέκτη και έστελνε bits σε συγκεκριμένα χρονικά διαστήματα. Ελέχθηκαν όλες οι περιπτώσεις(valid,frame_error,parity_error)

ΜΕΡΟΣ 4 << Σύστημα UART Αποστολέα-Δέκτη για Σειριακή Μεταφορά Δεδομένων>>

Υλοποίηση : Για την υλοιποίηση του συστήματος χρησιμοποιήθηκε κοινό baud rate και ένας μετρητής ο οποιός αναλάμβανε να αλλάξει το μήνυμα κάθε φορά που ο αποστολέας ήταν έτοιμος να λάβει νέα δεδομένα

Κώδικας Verilog:

```
reg [1:0]counter=2'b00;
always@(Tx_BUSY)
        if(!Tx_BUSY)
                case(counter)
                        2'b00: Tx_DATA<=8'b10101010;
                        2'b01: Tx_DATA<=8'b11110000;
                        2'b10: Tx_DATA<=8'b00001111;
                        2'b11: Tx_DATA<=8'b01010101;
                endcase
always@(Tx_BUSY)
        if(!Tx_BUSY)
                counter<=counter+1;
```

Επαλήθευση: Για την επαλήθευση σε επίπεδο συμπεριφοράς χρησιμοποιήθηκε ένα testbench που αρχικοποιούσε το baud rate και απλώς έτρεχε το κύκλωμα. Οι τιμές που έστελνε ο αποστολέας και δεχόταν ήταν ίδιες άρα το σύστημα δούλευε σωστά και συγχρονισμένα.

Συμπεράσματα : Σε γενικές γραμμές η πορεία που ακολουθήθηκε αποτελούνταν από τα παραπάνω μέρη. Σχεδίαση και επαλήθευση στον προσομοιωτή (διόρθωση οποιουδήποτε προβλήματος) .Κανένα μέρος δεν περιελάμβανε χρήση της πλακέτας κι έτσι το στάδιο του τελικού πειράματος απουσιάζει .

ΙΣΤΟΡΙΚΟ

Ημερομηνία	**ΚΩΔΙΚΑΣ**	**ΣΧΟΛΙΑ**
6/11/15	baud_controller.v	Ολοκλήρωση πρώτου μέρους και επαλήθευση
7/11/15	uart_transmitter.v	Ολοκλήρωση δεύτερου μέρους και επαλήθευση. Υλοποίηση αποστολέα
14/11/15	uart_receiver.v	Ολοκλήρωση τρίτου μέρους και επαλήθευση. Υλοποίηση δέκτη
18/11/15	system.v	Ολοκλήρωση συστήματος