ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
24.11.2014

ΕΡΓΑΣΤΗΡΙΟ ΨΗΦΙΑΚΩΝ ΚΥΚΛΩΜΑΤΩΝ

LAB2

UART IMPLEMENTATION SPARTAN 3

ΜΑΡΕΛΑΣ ΓΙΩΡΓΟΣ
ΑΕΜ 378

Πρόλογος

Ο στόχος της εργαστηριακής άσκησης είναι η υλοποίηση ενός συστήματος σειριακής επικοινωνίας το οποίο χρησιμοποιεί το πρωτόκολλο UART και εξυπηρετεί την επικοινωνία μεταξύ 2 συσκευών. Η υλοποιήση χωρίζεται σε 4 μέρη.Περιλαμβάνει την σειριακή μεταφορά συμβόλων των 8 bit από τον αποστολέα στον δέκτη.

Στο 1ο μέρος υλοποιούμε το κατάλληλο σήμα δειγματοληψίας ανάλογα με το επιλεγμένο baud rate. Στο δεύτερο μέρος έχουμε την υλοποιήση του αποστολέα ο οποίος παραλαβάνει το σύμβολο προς μεταφορά ενώ στο τρίτο μέρος έχουμε τον δέκτη ο οποίος κατά αναλογία με τον αποστολέα ενεργοποιείται και λαμβάνει το σύμβολο μεταφέροντας το στο σύστημα.Τέλος στο τέταρτο μέρος συννενόνονται τα δύο modules σε ένα πλήρες κανάλι UART και επιβεβαιώνεται η ορθή λειτουργία του.

Εισαγωγή

Στόχος της άσκησης είναι να κατανοήσουμε την λειτουργία του πρωτοκόλλου το οποίο για την επικοινωνία του στηρίζετε στην δειγματοληψία των bits με κάποια προσυμφωνημένη σταθερή φάση και όχι με τη χρήση κάποιας στοίβας. Με την δημιουργία του baud rate πετυχαίνουμε τον συγχρονισμό του αποστολέα με τον δέκτη ενώ τα ρολόγια τους είναι ασύγχρονα. Η βάση της λειτουργίας στηρίζεται στο ότι ο αποστολέας σύμφωνα με το baud rate που έχει σχεδιαστεί αναλογικά με το ρολόι του, δειγματοληπτεί 16 φορές πιο αργά και στέλνει ένα σήμα στον δέκτη ο οποίος με την σειρά του έχοντας προσαρμοσμένο το baud rate στο δικό του ρολόι δειγματοληπτεί 16 φορες πιο γρήγορα. Το αποτέλεσμα είναι να παίρνουμε πάντα στο σωστό σήμα αποφεύγοντας λανθασμένη μετάδοση σήματος λόγω μη συγχρονισμένων ρολογιών .Τα modules της υλοποιήσης έχουν ως εξής:

baud_controller_tr & baud_controller_Rx με test bench : baud_controller_tb

Uart_Tx με test bench : UartT_tb

Uart_Receiver με test bech : Uart_Rx_tb

Uart core με test bech : Uart_tb

ΜΕΡΟΣ Α :

BAUD RATE CONTROLLER

Η λειτουργία και δειγματοληψία στο UART γίνεται με πολλαπλάσιο ρυθμό του baud rate και συγκεκριμένα είναι υπολογισμένο ως : 16xBaudRate. Έχουμε 8 τιμές για την προσυμφωνημένη ταχύτητα επικοινωνίας, από 300bits/sec έως και τα 115200bits/sec. Η περίοδος του κάθε ψηφίου που αποστέλεται είναι 1/(16xBaudRate).

Ο ρυθμός μετάδοσης καθορίζεται πριν την έναρξη της επικοινωνίας και για οι σωστές τιμές του συγχονισμού υπολογίζουμε το εξής: F = (1/ FCLOK(16xBaudRate)). Έτσι προκύπτει κάθε πότε ένα σήμα enable θα ενεργοποιείται,για ένα κύκλο ρολογιου, ώστε ο Αποστολέας και ο Δέκτης να στείλει ή να λάβει αντίστοιχα. Το Fclock προκύπτει σύμφωνα με το ρολόι λειτουργίας της κάθε συσκευής και είναι διαφορετικό για τον Αποστολέα και τον Δέκτη. Οι υπολογισμοί των τιμών έγιναν για τον Αποστολέα με το ρολόι της Spartan3 (50 Mhz), ενώ για τον Δέκτη βάση ενός ρολογιού 150 Mhz. Το δεύτερο ρολόι επιλέχθηκε έτσι ώστε να μην συμπίπτουν οι ακμές του με το ρολόι Αποστολέα και να αποφεύγεται ο συγχρονισμός. Παίρνοντας την επιθυμητή συχνότητα δειγματοληψίας χρησιμοποιήσαμε ένα μετρητή ο οποίος κάθε φορά που ανακυκλώνεται ενεργοποιεί ένα σήμα ενός κύκλου.

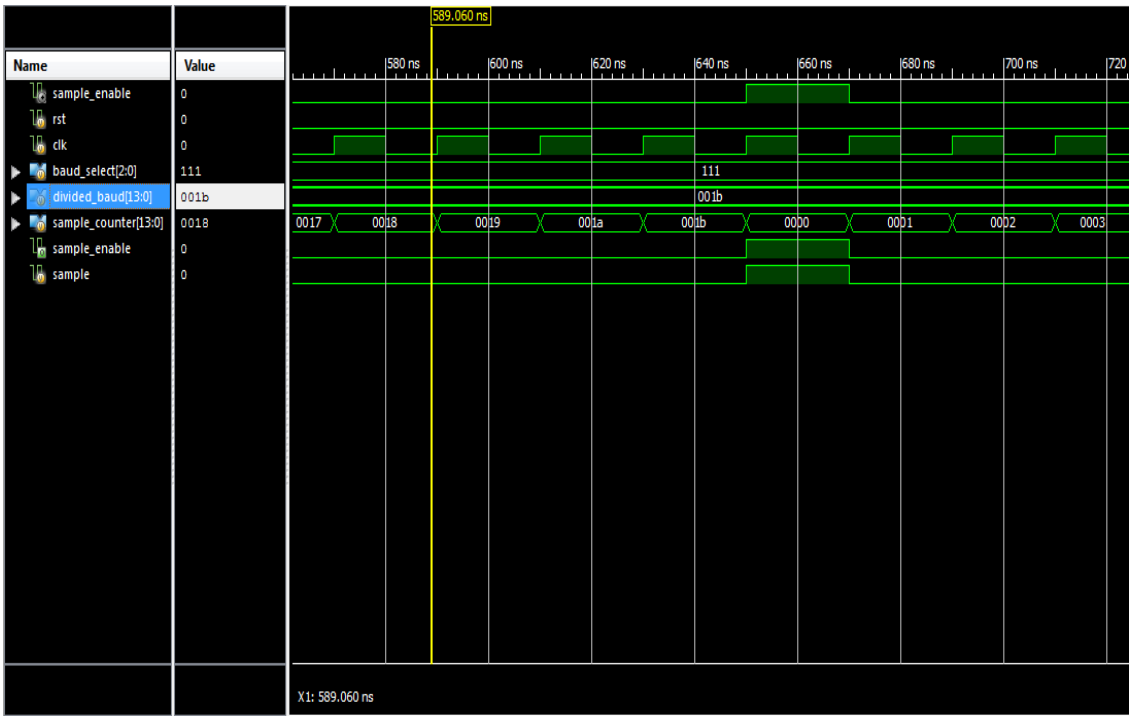
Συγκεκριμένα επιλέγοντας το baud rate στα 115200bits/sec έχουμε :

1)Για την συχνότητα του ρολογιού των 50Mhz κάνοντας τον παραπάνω υπολογισμό παίρνουμε 27,1267 συχνότητα ψηφίου. Το κλασματικό μέρος του αριθμού δεν μπορεί υπολογιστεί στο υλικό και παίρνουμε μόνο το ακέραιο μέρος ως μέγεθος του μετρητή. Το ότι λαμβάνουμε υπόψη μας μόνο τον δεκαδικό αριθμό αυτό δημιουργεί ένα μικρό σφάλμα 0,12ns.

2) Για την συχνότητα του ρολογιού των 150Mhz κάνοντας τον υπολογισμό παίρνουμε 90.4 συχνότητα ψηφίου. Το κλασματικό μέρος του αριθμού δεν μπορεί υπολογιστεί στο υλικό και παίρνουμε μόνο το ακέραιο μέρος ως μέγεθος του μετρητή. Το ότι λαμβάνουμε υπόψη μας μόνο τον δεκαδικό αριθμό αυτό δημιουργεί ένα μικρό σφάλμα 0,4ns.

ΕΠΑΛΗΘΕΥΣΗ :

Χρήση test bench αρχείου το οποίο είχε ως τιμές δοκιμής τις 115200 bits/sec και 38400 bits/sec. Όπως παρατηρούμε ανά σταθερή χρονική περίοδο ενεργοποιείται ένα σήμα ενός κύκλου ρολογιού.



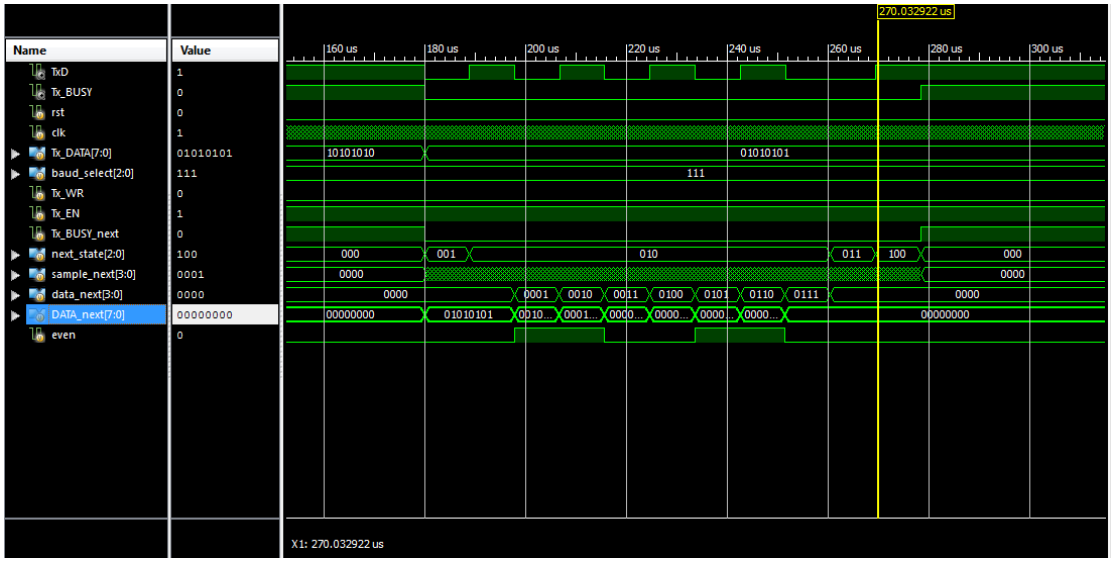
ΜΕΡΟΣ Β :
UART TRANSMITTER

Το δεύτερο μέρος περιλαμβάνει την υλοποίηση του Αποστολέα ως ενεργοποιήσιμο module. Λαμβάνει από το σύστημα ένα σύμβολο 8 ψηφίων και τα στέλνει ένα προς ένα στον Δέκτη. Χρησιμοποιείται μία μηχανή πεπερασμένων καταστάσεων με data path. Έχουμε 5 καταστάσεις που καθορίζουν την αποστολή δεδομένων στον Δέκτη καθώς και την ενημέρωση του συστήματος του Αποστολέα όταν πραγματοποιεί μετάδοση.

1) Idle : Ορίζουμε το σήμα μετάδοσης ως ανενεργό(στέλνει συνεχώς 1), ενώ χρησιμοποιούμε ένα input enable σήμα (Tx_WR)που ορίζει οτι γίνεται αποθήκευση των δεδομένων προς αποστολή σε έναν καταχωρητή 8bit. Ενεργοποιείται η επόμενη κατάσταση καθώς και ένα σήμα προς το σύστημα του αποστολέα το οποίο δηλώνει πως εκτελείται μετάδοση και ο αποστολέας δεν δέχεται δεδομένα(Tx_BUSY).
2) Start : Ενεργοποίηση σήματος(start bit) και αποστολή στον Δέκτη το οποίο σηματοδοτεί την έναρξη της μετάδοσης.Έχοντας ενεργό το σήμα του baud controller χρησιμοποιούμε ένα μετρητή για να πετύχουμε την καθυστέρηση της μετάδοσης κατα 16 φορές και μόλις ανακυκλώνεται ο μετρητής μεταβαίνουμε στην καταστασηη Data.
3) Data : Κάνοντας δεξιά ολίσθηση στον καταχωρητή με τα δεδομένα αποστέλουμε όλα ψηφία στον Δέκτη.Χρήση μετρητή ώστε η μετάδοση κάθε ψηφίου να γινεταί κάθε 16 ενεργά παλμούς του baud_controller. Επιπλέον υπολογίζουμε τον συνολικό αριθμό των ψηφίων που είναι 1 και ενεργοποιούμε αντίστοιχο ψηφίο ισοτιμίας ανάλογα αν είναι άρτιος ή περιττός ο αριθμός τους.Ενεργοποιήση κατάστασης Parity.
4) Parity : Αποστολή ψηφίου ισοτιμίας στον Δέκτη. Χρησιμοποιούμε ίδιο μετρητή με τις προηγούμενες καταστάσεις για να μεταβούμε στην επόμενη.
5) Stop : Αποστολή ψηφίου 1 στον Δέκτη ώστε ώς stop bit για να δείξουμε το πέρας της αποστολής. Απενεργοποιούμε το Tx_BUSY ώστε το σύστημα του αποστολέα να ενημερωθεί και να μπορεί να λάβει σύμβολα προς αποστολή.Μεταφερόμαστε στην αρχική κατάσταση.

ΕΠΑΛΗΘΕΥΣΗ :

Χρήση test bench αρχείου. Θέτουμε τη χρονική στιγμή 0 την τιμή του ρολογιού στο 0, rst =1,Tx_WR = 0, Tx_EN=0, Tx_DATA=0 και baud_select = 0. Μετά από 1000ns rst = 0,Tx_EN = 1, baud_select = 3'b111, Tx_WR = 1, Tx_DATA = 8'b10101010, μετά από μια περίοδο του ρολογιού 20ns Tx_WR = 0. Η τιμή του Tx_DATA και του Tx_WR αλλάζουν άλλες 3 φορές με 18000ns καθυστέρηση.



ΠΕΙΡΑΜΑΤΑ/ΤΕΛΙΚΗ ΥΛΟΠΟΙΗΣΗ:

Δεν δοκιμάστηκε στην FPGA.

ΜΕΡΟΣ Γ :
UART RECEIVER

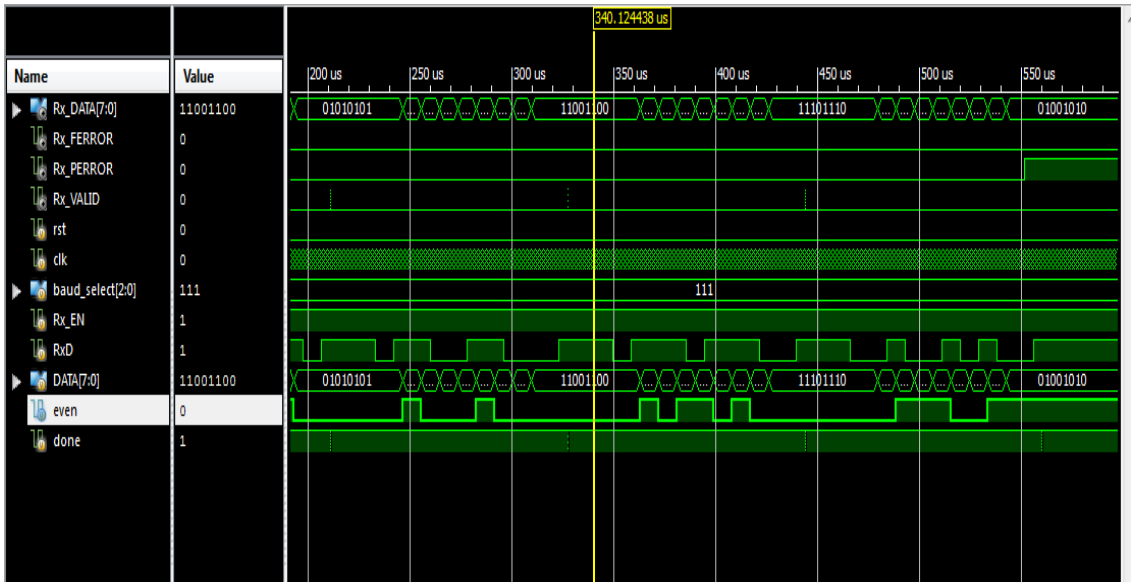
Το τρίτο μέρος περιλαμβάνει την υλοποίηση του Δέκτη ως ενεργοποιήσιμο module. Λαμβάνει ένα σύμβολο 8 ψηφίων και ελέγχει την ορθότητα του ως προς την ισοτιμία και την σειριακή πλαισίωση των δεδομένων ώστε να διαβαστούν από το σύστημα.Χρησιμοποιείται μία μηχανή πεπερασμένων καταστάσεων με data path. Έχουμε 5 καταστάσεις που καθορίζουν την λήψη δεδομένων καθώς και την ενημέρωση του συστήματος του Δέκτη όταν μπορεί να γίνει ανάγνωση.

1) Idle : Όταν ο Δέκτης είναι ενεργοποιημένος και το σήμα μετάδοσης του Αποστολέα γίνει 0 αρχικοποιούνται οι βοηθητικοί μετρητές και μεταβαίνουμε στη κατάσταση Start.
2) Start : Δεδομένου ότι ο Δέκτης δειγματοληπτεί 16 φορές πιο γρήγορα από τον αποστολέα πάντα με την χρήση του baud_controller, χρησιμοποιούμε ένα μετρητή ο οποίος φτάνει έως το 8 για να ελέγχουμε το μέσο του παλμού του αποστολέα και αν είναι 0 (start bit) τότε ενεργοποιείται η επόμενη κατάσταση.
3) Data : Με τη χρήση μετρητή ώστε η μετάδοση κάθε ψηφίου να γινεταί κάθε 16 ενεργούς παλμούς του baud_controller μεταβαίνουμε στο επόμενο μεσαίο παλμό του αποστολέα. Παίρνουμε την τιμή του σήματος και την κάνουμε δεξιά ολίσθηση στον καταχωρητή που αποθηκευουμε τα δεδομένα μας.Επιπλέον υπολογίζουμε τον συνολικό αριθμό των ψηφίων που είναι 1 και ενεργοποιούμε αντίστοιχο ψηφίο ισοτιμίας ανάλογα αν είναι άρτιος ή περιττός ο αριθμός τους. Μετάβαση στην κατάσταση Parity.
4) Parity : Έλεγχος ψηφίου ισοτιμίας με το parity bit του Αποστολέα και ενεργοποιήση σήματος ορθότητας.Χρησιμοποιούμε ίδιο μετρητή με τις προηγούμενες καταστάσεις για να μεταβούμε στην επόμενη.
5) Stop : Χρησιμοποιούμε σήμα enable για να δείξει οτι τελείωσε η λήψη δεδομένων και μεταβαίνουμε στην κατασταση idle.

Εφόσον τα τρία σήματα ελέγχου δεν είναι είναι ενεργά (Rx_PERROR , Rx_FERROR, done) ενεργοποιείται το σήμα Rx_VALID που ενημερώνει για την παραλαβή των δεδομένων.

ΕΠΑΛΗΘΕΥΣΗ :

Για την επαλήθευση του Δέκτη δημιούργησα ένα test bench οποίο την χρονική στιγμή 0ns θέτει clk = 0, rst = 1, baud_select = 0, Rx_EN = 0 ύστερα από 200ns Rx_EN = 1, reset = 0, RxD = 1, baud_select = 3'b111 και την εξής συμβολοσειρά 10101010 η οποία αλλάζει για 3 ακόμη φορές κάθε 26880ns σε 01010101, 11001100 και 11101110. Ως τελευταία συμβολοσειρά χρησιμοποίσα την 01001010 ώστε να φαίνεται η λειτουργία του ψηφίου ελέγχου ισοτιμίας και της μη ανάγνωσης άρτιου αριθμού 1.



ΠΕΙΡΑΜΑΤΑ/ΤΕΛΙΚΗ ΥΛΟΠΟΙΗΣΗ:

Δεν δοκιμάστηκε στην FPGA.

ΜΕΡΟΣ Δ :
UART SYSTEM

Στο τελευταίο μέρος συνδέουμε τον Αποστολέα με τον Δέκτη χρησιμοποιώντας διαφορετικό ρολόι για το κάθε υποσύστημα όπως ανέφερα και παραπάνω. Υλοποιώντας τις εκατέρωθεν συνδέσεις των 2 modules με τα συστήματα τους αντίστοιχα συνδέουμε και το TxD με το RxD ώστε να ολοκληρωθεί η επικοινωνία τους σε ένα κανάλι. Η χρήση 2 διαφορετικών ρολογιών γινεται για να μη συμπίπτουν οι ακμές των παλμών Αποστολέα και Δέκτη ώστε να έχουμε ασύγχρονη επικοινωνία (Ρολόι 50Mhz για τον Αποστολέα και 150Mhz για τον Δέκτη). Όπως φαίνεται αναλυτικά και στα 2 προηγούμενα στάδια υλοποίησης ο Αποστολέας λαμβάνοντας τα δεδομένα στην ενεργοποιήση σήματος ενός κυκλου (Tx_WR) ξεξινάει την μετάδοση μέσω του TxD προς τον Δέκτη ενώ ταυτόχρονα έχει ενεργοποιήσει σήμα Tx_BUSY(1), το οποίο δείχνει πως δεν δέχεται δεδομένα για αποστολή. Αντίστοιχα ο Δεκτης χρησιμοποιεί σήμα Rx_VALID για να ενημέρωση ότι ολοκληρώθηκε σωστα η ανάγνωση(0) ή όχι (1).

ΕΠΑΛΗΘΕΥΣΗ:

Για την επαλήθευση δημιούργησα ένα test bench στο οποίο την στιγμη 0ns δίνουμε τις τιμές : clkT = 0, clkR = 0 , rst = 1, baud_select = 0; Tx_DATA = 0, Tx_WR = 0, Tx_EN = 0. Μετά απο 100 ns rst = 0, Tx_EN = 1, baud_select = 3'b111, Tx_WR = 1, Tx_DATA = 8'b10101010. Έπειτα μετά από 20 ns Tx_WR = 0 και μετά από 180000 Tx_WR = 1, Tx_DATA = 8'b01010101.Επαναλαμβάνω τους ίδιους χρόνους, για τα σύμβολα 11001100 , 10001001.