



Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  
Πανεπιστήμιο Θεσσαλίας

## ECE333 - Εργαστήριο Ψηφιακών Συστημάτων

Χειμερινό Εξάμηνο — Ακαδημαϊκό Έτος 2023-2024

### Εργαστηριακή Εργασία 2<sup>η</sup>

### Υλοποίηση Μονάδας Γενικού Ασύγχρονου Δέκτη Αποστολέα

Universal Asynchronous Receiver Transmitter — UART

20/11/2023 έως 11/12/2023

Χ. Σωτηρίου

## Περιεχόμενα

1	Στόχος της 2ης Εργασίας	2
2	Το UART	2
3	Μέρος Α - Ελεγκτής Baud Rate	5
4	Μέρος Β - Υλοποίηση UART Αποστολέα (Transmitter)	5
4.1	Δομή της Μονάδας	6
4.2	Λειτουργία του Αποστολέα	6
5	Μέρος Γ - Υλοποίηση UART Δέκτη (Receiver)	7
5.1	Δομή της Μονάδας	7
5.2	Λειτουργία του Δέκτη	8
6	Μέρος Δ - Υλοποίηση UART Αποστολέα-Δέκτη για Σειριακή Μεταφορά Δεδομένων	9
6.1	Προχωρημένη Χρήση της UART Μονάδας (Προαιρετικό)	9
7	Προθεσμία Παράδοσης, Υποβολή της Εργασίας και Αναφορά	9
8	Ερωτήσεις και Απορίες	9

# 1 Στόχος της 2ης Εργασίας

Ο στόχος της 2<sup>ης</sup> εργαστηριακής εργασίας είναι η υλοποίηση ενός συστήματος σειριακής επικοινωνίας, το οποίο θα χρησιμοποιεί το πρωτόκολλο UART (Universal Asynchronous Receiver Transmitter — Γενικού Ασύγχρονου Δέκτη Αποστολέα). Το σύστημα θα αποτελείται από έναν UART **Αποστολέα** (Μέρος Β) και έναν UART **Δέκτη** (Μέρος Γ), οι οποίοι μεταφέρουν δεδομένα μόνο προς τη μια κατεύθυνση, από τον Αποστολέα στον Δέκτη (Transmitter→Receiver), μέσω μιας σειριακής σύνδεσης ενός σήματος.

Το UART που θα υλοποιηθεί, θα πρέπει να ελεγχθεί τουλάχιστον για τη σειριακή μεταφορά μιας αλληλουχίας **τεσσάρων διαφορετικών συμβόλων των 8-bit**, από τον Αποστολέα στον Δέκτη. Η προτεινόμενη αλληλουχία ελέγχου φαίνεται στο Σχήμα 1 σε δυαδικό και δεκαεξαδικό σύστημα κωδικοποίησης.

10101010(AA), 01010101(55), 11001100(CC) και 10001001(89).
------------------------------------------------------------

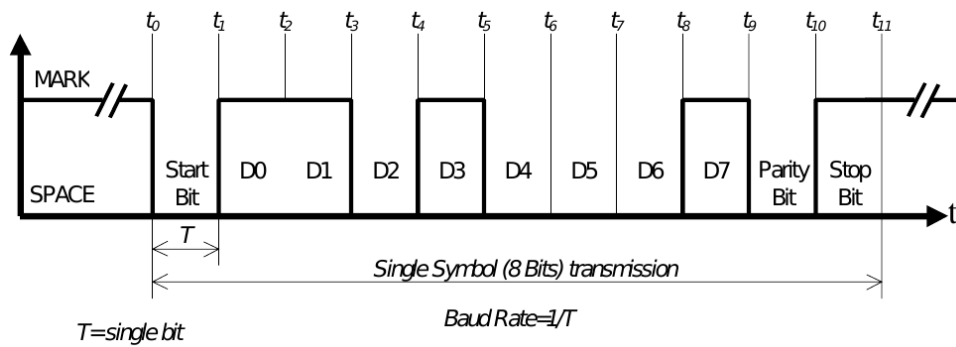
Σχήμα 1: Μια Αλληλουχία Ελέγχου για το UART Πρωτόκολλο

Εναλλακτικά, μπορείτε τουλάχιστον να μεταφέρετε μια άλλη αλληλουχία τεσσάρων συμβόλων, αρκεί να εμπεριέχει αρκετές μεταβάσεις μεταξύ 0 και 1, έτσι ώστε να επαληθεύουν τον ορθό χρονισμό του κάθε ψηφίου που μεταδίδεται σειριακά κατά την αποστολή και λήψη. Προαιρετικά, το τελευταίο σύμβολο που παραλήφθηκε θα παρουσιάζεται στα δυο από τα τέσσερα ψηφία ενδείξεων 7-τμημάτων.

## 2 Το UART

Το UART (Universal Asynchronous Receiver Transmitter — Γενικός Ασύγχρονος Δέκτης και Αποστολέας) είναι ένα σειριακό, ασύγχρονο πρωτόκολλο επικοινωνίας, και σχετικό κύκλωμα που το υλοποιεί, το οποίο επιτρέπει τη μεταφορά δεδομένων μεταξύ δύο (ή περισσότερων γενικά) συσκευών, οι οποίες μπορεί να έχουν ανεξάρτητα και ασυσχέτιστα ρολόγια. Το UART είναι ευρύτατα διαδεδομένο (RS232), λόγω της απλότητας στην υλοποίηση, της πρακτικής ευκολίας στη χρήση και της γενικής του εφαρμογής.

Η ασύγχρονη επικοινωνία του UART συντελείται μέσω **μιας ενσύρματης σύνδεσης ενός bit**, μεταξύ του Αποστολέα (TxD), ο οποίος την οδηγεί, και του Δέκτη (RxD), ο οποίος τη δειγματοληπτεί και την εξετάζει. Το δεδομένο προς επικοινωνία συνήθως είναι 8-bit και ονομάζεται **σύμβολο**. Για να σταλεί σειριακά ένα σύμβολο πρέπει να μετατραπεί στα συναποτελούντα του δυαδικά ψηφία, τα οποία θα σταλούν ένα προς ένα, από το μικρότερο (*bit 0*) στο μεγαλύτερο (*bit 7*). Στο Σχήμα 2, φαίνεται αναλυτικά το χρονοδιάγραμμα επικοινωνίας ενός συμβόλου. Μιας και δεν υπάρχει κάποια εγγύηση ως προς τη σχετική συχνότητα και τη φάση μεταξύ των ρολογιών του Αποστολέα και Δέκτη και πρακτικά η επικοινωνία είναι ασύγχρονη, τα δεδομένα του συμβόλου προπορεύονται από το **bit Εκκίνησης** (Start bit). Το **bit Εκκίνησης** σηματοδοτεί την έναρξη της επικοινωνίας για τον Δέκτη. Αντίστοιχα, το πέρας του συμβόλου και η λήξη της επικοινωνίας σηματοδοτείται από το **bit Παύσης** (Stop bit). Έτσι, όσο το κανάλι επικοινωνίας 1-bit του UART μένει αδρανές, και δε χρησιμοποιείται για μεταφορά δεδομένων, θα πρέπει να παραμένει στο λογικό 1, την τιμή δηλαδή του Stop bit. Για την επαλήθευση της ορθής επικοινωνίας των δεδομένων, αυτά συνοδεύονται από ένα **bit Ισοτιμίας** (Parity), το οποίο καταδεικνύει αν το σύμβολο που μεταβιβάστηκε εμπεριέχει άρτιο ή περιττό αριθμό λογικών άσσων.



Σχήμα 2: Χρονικό Διάγραμμα του Σειριακού Πρωτοκόλλου UART, για επικοινωνία ενός Συμβόλου των 8-bit

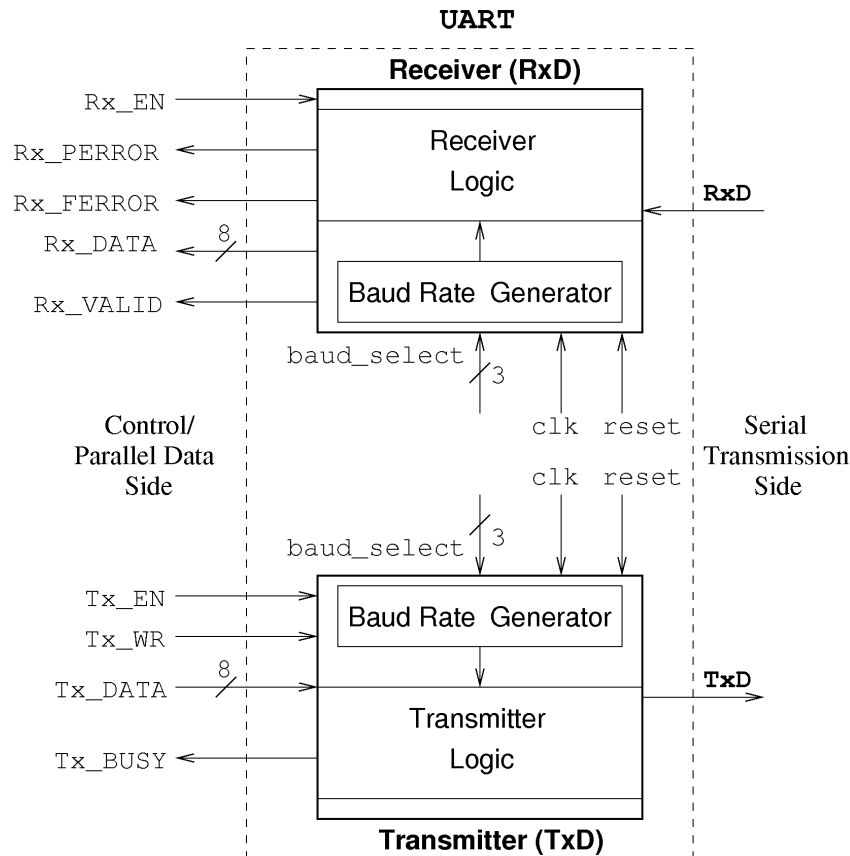
Για να μη χάνονται ή για να μην πολλαπλασιάζονται δεδομένα στο πρωτόκολλο UART, ο Αποστολέας και Δέκτης προσυμφωνούν σε έναν κοινό ρυθμό δειγματοληψίας, δηλαδή προσυμφωνούν την ταχύτητα της μεταξύ τους επικοινωνίας σε μονάδες Baud(bits/sec). Η προσυμφωνία και η επιλογή μίας κοινής ταχύτητας δεν είναι μέρος του πρωτοκόλλου επικοινωνίας αλλά γίνεται σε υψηλότερο επίπεδο. Στον παρακάτω πίνακα παρουσιάζονται οι ταχύτητες επικοινωνίας του UART που πρέπει να υποστηριχθούν, και η σχετική κωδικοποίηση του σήματος επιλογής του ρυθμού επικοινωνίας BAUD\_SEL (3-bit).

BAUD_SEL	Baud Rate (bits/sec)
0 0 0	300
0 0 1	1200
0 1 0	4800
0 1 1	9600
1 0 0	19200
1 0 1	38400
1 1 0	57600
1 1 1	115200

Η περίοδος μετάδοσης του κάθε ψηφίου από τον Αποστολέα στο Σχήμα 2 αναλογεί σε  $T = \frac{1}{\text{BaudRate}}$ . Ωστόσο, η λειτουργία και δειγματοληψία του UART Δέκτη γίνεται γρηγορότερα της προσυμφωνημένης συχνότητας Baud Rate βασιζόμενη ενός ακέραιου πολλαπλασίου αυτής. Η πιο συνήθης συχνότητα δειγματοληψίας είναι  $\times 16$ , δηλαδή το κάθε μεταδιδόμενο ψηφίο του Αποστολέα, εξετάζεται με 16πλάσιο ρυθμό από τον Δέκτη, για την προσυμπεφωνημένη ταχύτητα. Πρακτικά, ο Δέκτης λειτουργεί σε ταχύτητα  $\text{BaudRate} \times 16$ , ενώ ο Αποστολέας σε ταχύτητα  $\text{BaudRate}$ . Επιπλέον, ο Δέκτης, αφού συγχρονιστεί με το Start bit, και με την κατάλληλη καθυστέρηση από τη δειγματοληψία του τελευταίου, θα πρέπει να ευθυγραμμίσει τη δειγματοληψία του στο κέντρο της περιόδου,  $\frac{1}{\text{BaudRate}}$ .

Στο Σχήμα 3 παρουσιάζεται αναλυτικά η κυκλωματική δομή ενός ζεύγους Αποστολέα-Δέκτη UART. Και οι δύο μονάδες έχουν δύο πλευρές, την πλευρά που επιτελείται η σειριακή επικοινωνία, δηλ. τα σήματα RxD και TxD, τα οποία συνδέονται μεταξύ τους για να δημιουργήσουν το σειριακό κανάλι, και την πλευρά ελέγχου/δεδομένων, στην οποία το οποιοδήποτε σύστημα επικοινωνεί με τις μονάδες του UART, και αποστέλλει ή λαμβάνει τα (παράλληλα) δεδομένα των συμβόλων. Η κάθε μονάδα ενεργοποιείται για επικοινωνία με το σχετικό σήμα ενεργοποίησης Rx\_EN ή Tx\_EN. Επίσης, η κάθε μονάδα περιλαμβάνει

τον δικό της προγραμματιζόμενο ελεγκτή Baud Rate, ο οποίος θέτει τον ρυθμό μετάδοσης πριν την έναρξη, και όχι κατά τη διάρκεια, της επικοινωνίας. Ο ελεγκτής αυτός ορίζει την ταχύτητα των μονάδων όσο αυτές είναι ενεργές, δηλαδή βάση των κύκλων ρολογιού παράγει την κατάλληλη χρονική στιγμή το αντίστοιχο σήμα ενεργοποίησης λειτουργίας της κάθε μονάδας.



Σχήμα 3: Κυκλωματική Δομή και Διασύνδεση των Μονάδων UART

Όπως ειπώθηκε νωρίτερα, ο Αποστολέας υπολογίζει και στέλνει το ψηφίο Parity, ενώ ο Δέκτης θα πρέπει να επαληθεύσει ότι η ισοτιμία του συμβόλου είναι σωστή. Σε περίπτωση που διαπιστωθεί σφάλμα στην ισοτιμία, ο Δέκτης θα πρέπει να πληροφορήσει την πλευρά του συστήματος ότι τα δεδομένα που παρελήφθησαν δεν είναι σωστά, για να αγνοηθούν σε υψηλότερο επίπεδο. Αυτό επιτυγχάνεται με το σήμα Rx\_ERROR. Επιπλέον, στην περίπτωση που ο Δέκτης δε δειγματοληπτήσει το Stop bit στον χρόνο που το περιμένει ή δεν μπορεί να ευθυγραμμιστεί με τη μέση του Start bit, οι οποίες δυο περιπτώσεις σημαίνουν ότι τα δεδομένα δεν έχουν πλασιωθεί σωστά (λ.χ. το Baud Rate μπορεί να είναι διαφορετικό, ή ο Δέκτης να ενεργοποιηθεί καθυστερημένα), θα πληροφορήσει το σύστημα για το σφάλμα με το σήμα Rx\_FERROR.

Επισημαίνεται ότι το πρωτόκολλο UART δεν περιλαμβάνει έλεγχο ροής μεταξύ Δέκτη-Αποστολέα, δηλαδή ο Δέκτης δεν μπορεί να σταματήσει τον Αποστολέα, έτσι **και οι δυο θα πρέπει να είναι έτοιμοι για διαρκή επικοινωνία**. Στην πλευρά του συστήματος, όπου και οι ταχύτητες μπορεί να είναι πολύ μεγαλύτερες, είναι απαραίτητο ένα απλό πρωτόκολλο επικοινωνίας που να καταδεικνύει: **(1)** ότι ο Αποστολέας βρίσκεται σε διαδικασία μετάδοσης και δεν μπορεί να λάβει το επόμενο σύμβολο, και **(2)** ότι ο Δέκτης έλαβε σύμβολο που πρέπει να αναγνωστεί. Αυτές οι δυο απαιτήσεις επιτυγχάνονται με τα σήματα Tx\_BUSY και Rx\_VALID αντίστοιχα.

### 3 Μέρος Α - Ελεγκτής Baud Rate

Ο Ελεγκτής θα χρησιμοποιηθεί εσωτερικά στα κυκλώματα Αποστολέα και Δέκτη. Ο στόχος του είναι να παρέχει το κατάλληλο σήμα δειγματοληψίας, ανάλογα με τον επιλεγμένο Baud Rate. Το σήμα δειγματοληψίας σας προτείνεται να είναι θετικά ενεργό και να μένει ενεργό για ένα κύκλο, ενώ όπως επεξηγήθηκε παραπάνω θα πρέπει να έχει συχνότητα  $16 \times$  Baud Rate. Η προτεινόμενη μορφή του Ελεγκτή Baud Rate φαίνεται στο Σχήμα 4:

```
module baud_controller(reset, clk, baud_select, sample_ENABLE);  
  
    input reset, clk;  
    input [2:0] baud_select;  
    output sample_ENABLE;  
  
    ...  
  
endmodule
```

Σχήμα 4: Μια Μορφή Υλοποίησης του Ελεγκτή Baud Rate

Για τον Ελεγκτή Baud Rate, σας προτείνεται να χρησιμοποιήσετε το ρολόι των 100MHz της πλακέτας. Επιπλέον, για την παραγωγή του σήματος δειγματοληψίας, σας προτείνεται να χρησιμοποιήσετε έναν μετρητή κύκλων ρολογιού, όπου για κάθε απαιτούμενο Baud Rate, θα ορίσετε μια μέγιστη τιμή του μετρητή, η οποία **θα μετράει, με το μικρότερο δυνατό ποσοστό λάθους, την ποσότητα**  $T_{sc} = \frac{1}{16 \times \text{BaudRate}}$ , η οποία αντιστοιχεί στην περίοδο δειγματοληψίας, σε αριθμό κύκλων. Έτσι, ανάλογα με το απαιτούμενο Baud Rate, όταν ο μετρητής θα φτάνει τη σχετική μέγιστη τιμή, το σήμα sample\_ENABLE θα γίνεται 1, και ο μετρητής θα επιστρέφει στο 0. Με αυτόν τον τρόπο, κάθε περίοδο  $T_{sc} = \frac{1}{16 \times \text{BaudRate}}$ , το σήμα sample\_ENABLE θα γίνεται 1 για έναν κύκλο. Όταν υπολογίσετε τις απαιτούμενες τιμές του μετρητή, υπολογίστε και το σχετικό σφάλμα στο Baud Rate. Το τελευταίο θα πρέπει να το συμπεριλάβετε και να το σχολιάσετε στην εργαστηριακή αναφορά.

**Όταν η κυκλωματική υλοποίηση είναι ορθή, επιδειξτε τον κώδικα Verilog που γράψατε για το κύκλωμα (RTL Design) και το πλαίσιο δοκιμής (Testbench) και τα αποτελέσματα της προσομοίωσης σε επιτηρητή του εργαστηρίου.**

### 4 Μέρος Β - Υλοποίηση UART Αποστολέα (Transmitter)

Ο UART Αποστολέας θα πρέπει, όπως εξηγήθηκε στην Ενότητα 2, να είναι δυναμικά ενεργοποιήσιμος, να επικοινωνεί με το σύστημα για να παραλαμβάνει το σύμβολο προς μεταφορά, αλλά και να παράγει μια ένδειξη διαθεσιμότητας ή όχι, ανάλογα με το αν εκτελεί μεταφορά ή περιμένει επόμενο δεδομένο.

Για την πλευρά ελέγχου/δεδομένων μεταξύ του Αποστολέα και του συστήματος τα σχετικά σήματα και η λειτουργία τους περιγράφονται ως εξής:

- Το σήμα Tx\_EN αποτελεί το σήμα ενεργοποίησης του Αποστολέα. Το συγκεκριμένο σήμα θα πρέπει να μένει ενεργό, από το σύστημα (ή το πλαίσιο δοκιμής), όσο ο Αποστολέας πρέπει να είναι ενεργός ή όσο δεν έχει ακόμα ολοκληρώσει την τρέχουσα μεταβίβαση. Το Baud Rate θα πρέπει να έχει τεθεί πριν ενεργοποιηθεί ο Αποστολέας.

- Για την παραλαβή δεδομένων από το σύστημα που πρέπει να μεταβιβαστούν, σας προτείνετε να χρησιμοποιήσετε το σήμα Tx\_WR, όπου το τελευταίο θα γίνεται 1 για ένα κύκλο, ενώ τα δεδομένα του συμβόλου προς μεταφορά θα βρίσκονται στο Tx\_DATA[7:0].
- Το σήμα Tx\_BUSY, με κατεύθυνση από τον Αποστολέα προς το σύστημα, σηματοδοτεί ότι ο Αποστολέας βρίσκεται σε κατάσταση μεταβίβασης δεδομένων (εν λειτουργία) και θα πρέπει να παραμένει σταθερό στο λογικό 1 καθ'όλη τη διάρκεια της σειριακής αποστολής.
  - ο Έτσι (1) δεν επιτρέπεται να πέσει το Tx\_EN και (2) δεν επιτρέπεται να του μεταβιβαστούν νέα δεδομένα μέσω της κοινής λειτουργίας του ζεύγους Tx\_WR και Tx\_DATA[7:0].

## 4.1 Δομή της Μονάδας

Στην υλοποίηση του UART Αποστολέα, θα εμπεριέχεται ο Ελεγκτής Baud Rate που υλοποιήσατε στο πρώτο μέρος. Σας προτείνεται η εξής δομή Αποστολέα του Σχήματος 5:

```
module uart_transmitter(reset, clk, Tx_DATA, baud_select, Tx_WR, Tx_EN, Tx_D, Tx_BUSY);

    input reset, clk;
    input [7:0] Tx_DATA;
    input [2:0] baud_select;
    input Tx_EN;
    input Tx_WR;

    output Tx_D;
    output Tx_BUSY;

    ...

    baud_controller baud_controller_tx_inst(reset, clk, baud_select, Tx_sample_ENABLE);

    ...

endmodule
```

Σχήμα 5: Μια Μορφή Υλοποίησης του Αποστολέα UART

## 4.2 Λειτουργία του Αποστολέα

Ο Αποστολέας μπορεί να βρίσκεται σε μια από δύο καταστάσεις, είτε θα είναι ανενεργός, είτε θα βρίσκεται σε διαδικασία μετάδοσης δεδομένων. Μετά την παραλαβή συμβόλου από το σύστημα, όπως επεξηγήθηκε παραπάνω, και εφόσον είναι ενεργός από το σήμα Tx\_EN, ο Αποστολέας θα πρέπει να μπει στην κατάσταση μετάδοσης, και να ξεκινήσει την αποστολή στη σειριακή σύνδεση Tx\_D, στέλνοντας το Start bit. Οι κύκλοι που το σήμα Tx\_sample\_ENABLE είναι 1, σηματοδοτούν τους ενεργούς κύκλους για τον Αποστολέα. Έτσι, η μετάδοση δεδομένων, συμπεριλαμβανομένου και του Start bit, θα γίνεται κάθε 16 κύκλους ενεργοποίησης του Tx\_sample\_ENABLE, το οποίο και εξαρτάται από το επιλεγμένο

**Baud Rate.** Αλλάζοντας το Baud Rate, θα αλλάξουν και οι ενεργοί κύκλοι μετάδοσης του Αποστολέα, οπότε η σειριακή αλληλουχία των δεδομένων είτε θα πλαταίνει, είτε θα συρρικνώνεται χρονικά.

Για την υλοποίηση του Αποστολέα, σας προτείνεται να χρησιμοποιήσετε έναν μετρητή δεδομένων, ο οποίος κάθε 16 παλμούς του Tx\_sample\_ENABLE, θα οδηγεί κατάλληλα το TxD. Συνολικά, θα πρέπει μεταδώσετε σειριακά 11bit (8-bit δεδομένων, 1-bit αρχής, 1-bit τέλους και 1-bit ισοτιμίας). Ολοκληρώνοντας τη διαδικασία, ο μετρητής δεδομένων μπορεί να επιστρέψει στο 0, και ο Αποστολέας στην ανενεργή κατάσταση.

Όταν η κυκλωματική υλοποίηση είναι ορθή, επιδείξτε τον κώδικα Verilog που γράψατε για το κύκλωμα (RTL Design) και το πλαίσιο δοκιμής (Testbench) και τα αποτελέσματα της προσομοίωσης σε επιτηρητή του εργαστηρίου.

## 5 Μέρος Γ - Υλοποίηση UART Δέκτη (Receiver)

Ο UART Δέκτης, κατά αναλογία με τον Αποστολέα (Ενότητα 4), θα πρέπει να είναι ενεργοποιήσιμος δυναμικά με τον ίδιο τρόπο, και επιπλέον να επιδεικνύει στο σύστημα ότι υπάρχει διαθέσιμο σύμβολο προς ανάγνωση. Αντίστοιχα με τον Αποστολέα λοιπόν, για την πλευρά ελέγχου/δεδομένων μεταξύ του Δέκτη και του συστήματος τα σχετικά σήματα και οι λειτουργία τους περιγράφονται ως εξής:

- Το σήμα Rx\_EN αποτελεί το σήμα ενεργοποίησης του Δέκτη, με ανάλογη συμπεριφορά λειτουργίας και περιορισμών με του Αποστολέα. Ομοίως, λειτουργεί η επιλογή Baud Rate, όπου θα πρέπει να έχει τεθεί πριν ενεργοποιηθεί ο Δέκτης.
- Κατά την ολοκλήρωση της λήψης από τον Δέκτη, αν διαπιστωθεί σφάλμα είτε στην ισοτιμία είτε στη σειριακή πλαισίωση των δεδομένων (όπως εξηγήθηκε στην Ενότητα 2), θα πρέπει να σηματοδοτηθεί το αντίστοιχο σφάλμα με λογικό 1 από τα σχετικά σήματα λάθους Rx\_PERROR και Rx\_FERROR.
- Εναλλακτικά, στην περίπτωση ορθής λειτουργίας όπου δεν παρατηρείται κάποιο σφάλμα, το σύμβολο που παρελήφθη θα πρέπει να εμφανιστεί στο Rx\_DATA[7:0], και για τουλάχιστον έναν κύκλο να σηκωθεί (λογικό 1) το σήμα Rx\_VALID, το οποίο σηματοδοτεί ότι τα δεδομένα είναι έγκυρα και μπορούν να διαβαστούν από το σύστημα.

Επίσης, για ευκολία σας προτείνεται, στην επικοινωνία μεταξύ του συστήματος (πλαίσιο δοκιμής) και του Δέκτη, το ζεύγος σημάτων ορθής λειτουργίας Rx\_DATA[7:0] και Rx\_VALID να μένει έγκυρο, έως ότου ξεκινήσει νέα επικοινωνία από τον Αποστολέα. Δηλαδή να τα δύο σήματα να παραμένουν σταθερά εάν δεν έχει παραλήφθει νέο Start bit. Ομοίως, σας προτείνεται να υποστηριχθεί η ίδια λειτουργικότητα και για τα δύο σήματα σφάλματος Rx\_PERROR και Rx\_FERROR.

### 5.1 Δομή της Μονάδας

Όπως στον Αποστολέα έτσι και στον Δέκτη, θα εμπεριέχεται ο Ελεγκτής Baud Rate που υλοποιήσατε στο πρώτο μέρος. Σας προτείνεται η εξής δομή Δέκτη του Σχήματος 6:

```

module uart_receiver(reset, clk, Rx_DATA, baud_select, Rx_EN, Rx_D, \
Rx_FERROR, Rx_PERROR, Rx_VALID);

    input reset, clk;
    input [2:0] baud_select;
    input Rx_EN;
    input Rx_D;

    output [7:0] Rx_DATA;
    output Rx_FERROR; // Framing Error //
    output Rx_PERROR; // Parity Error //
    output Rx_VALID; // Rx_DATA is Valid //

    ...

    baud_controller baud_controller_rx_inst(reset, clk, baud_select, Rx_sample_ENABLE);

    ...

endmodule

```

Σχήμα 6: Μια Μορφή Υλοποίησης του Δέκτη UART

## 5.2 Λειτουργία του Δέκτη

Ο Δέκτης επίσης έχει δυο καταστάσεις, την ανενεργή, όπου περιμένει δεδομένα, και την κατάσταση λήψης, όπου έχει ξεκινήσει και είναι ενεργή η σειριακή παραλαβή των bit. Ο Δέκτης μπαίνει στην κατάσταση λήψης με την ανίχνευση του **Start bit**. Μια και ο Δέκτης είναι ασύγχρονος ως προς τον Αποστολέα, θα πρέπει πριν την οποιαδήποτε δειγματοληψία, να συγχρονιστεί η ασύγχρονη σειριακή είσοδος.

Είναι σημαντικό ο Δέκτης να δειγματοληπτεί σωστά τα σειριακά δεδομένα. Έτσι, για μέγιστη αξιοπιστία, η δειγματοληψία του Δέκτη θα πρέπει να πραγματοποιείται στην προβλεπόμενη μέση των 16 ενεργοποιήσεων για το επόμενο bit. Η συγκεκριμένη πρόβλεψη, και η κατάλληλη ευθυγράμμιση της δειγματοληψίας, γίνεται σε σχέση με το **Start bit**, μετρώντας τον κατάλληλο αριθμό ενεργών κύκλων, σύμφωνα με το **Baud Rate**. Στο Δέκτη, οι ενεργοί κύκλοι σηματοδοτούνται από το σχετικό σήμα **Rx\_sample\_ENABLE**.

Για την υλοποίηση του Δέκτη, σας προτείνεται να χρησιμοποιήσετε αναλόγως έναν μετρητή δεδομένων, όπου ανάλογα με το bit που αναμένεται, θα γίνονται οι κατάλληλες ενέργειες. Τα bit δεδομένων θα πρέπει να ολισθαίνουν ένα προς ένα σε έναν καταχωρητή, έτσι ώστε να παραδοθούν μετά παράλληλα στο σύστημα. Το πέρας της επικοινωνίας σηματοδοτείται από τη δειγματοληψία του **Stop bit** στον κατάλληλο κύκλο λήψης. Τότε, θα πρέπει να ενεργοποιείται η λειτουργία των **Rx\_VALID**, **Rx\_DATA[7:0]**, όπως εξηγήθηκε παραπάνω.

**Όταν η κυκλωματική υλοποίηση είναι ορθή, επιδείξτε τον κώδικα Verilog που γράψατε για το κύκλωμα (RTL Design) και το πλαίσιο δοκιμής (Testbench) και τα αποτελέσματα της προσομοίωσης σε επιτηρητή του εργαστηρίου.**



## 6 Μέρος Δ - Υλοποίηση UART Αποστολέα-Δέκτη για Σειριακή Μεταφορά Δεδομένων

Έχοντας υλοποιήσει τον Αποστολέα και Δέκτη, ο στόχος τώρα είναι να συνενωθούν σε ένα πλήρες κανάλι UART. Υλοποιήστε το ζεύγος Αποστολέα-Δέκτη, και το κατάλληλο πλαίσιο δοκιμής για να ελέγξετε ότι η μεταφορά των τεσσάρων διαδοχικών λέξεων γίνεται σωστά, και ότι τα απεσταλμένα δεδομένα παραλαμβάνονται σωστά στην πλευρά του Δέκτη. Η επικοινωνία του πλαισίου δοκιμής με τους Αποστολέα και Δέκτη, θα πρέπει να βασίζεται αποκλειστικά στο πρωτόκολλο των σχετικών σημάτων, λ.χ. Rx\_VALID, Tx\_BUSY, και όχι σε μέτρηση κύκλων ή απόλυτες καθυστερήσεις.

### 6.1 Προχωρημένη Χρήση της UART Μονάδας (Προαιρετικό)

Προαιρετικά, μετατρέψτε το πλαίσιο δοκιμής σε δυο κυκλώματα τα οποία θα είναι συνθέσιμα στην FPGA. Το πρώτο, οδηγεί τον Αποστολέα με τον ίδιο τρόπο, δηλαδή του παρέχει την αλληλουχία των τεσσάρων δεδομένων. Το δεύτερο, λαμβάνει το δεδομένο που παρέλαβε ο Δέκτης, και χρησιμοποιεί μια κατάλληλη τροποποίηση του κυκλώματος του Οδηγού 7-τμημάτων που φτιάξατε στη 1<sup>η</sup> Εργαστηριακή Εργασία, για να απεικονίσει τα δυο δεκαεξάδικα ψηφία του. Επιπλέον, σας προτείνεται να χρησιμοποιήσετε τους σταθερούς διακόπτες της FPGA για επιλογή του Baud Rate.

(Θα δοθεί μόνους στην βαθμολογία για την υλοποίηση του προαιρετικού μέρους)

Όταν η κυκλωματική υλοποίηση είναι ορθή, επιδείξτε τον κώδικα Verilog που γράψατε για το κύκλωμα (RTL Design) και το πλαίσιο δοκιμής (Testbench), τα αποτελέσματα της προσομοίωσης, και προαιρετικά το κύκλωμα εν λειτουργία σε επιτηρητή του εργαστηρίου.

## 7 Προθεσμία Παράδοσης, Υποβολή της Εργασίας και Αναφορά

Η προθεσμία παράδοσης της 2<sup>ης</sup> εργασίας είναι η 11/12/2023.

Μέχρι την προθεσμία της εργασίας θα πρέπει:

- να έχετε επιδείξει όλα τα επιμέρους μέρη της εργασίας στους επιτηρητές,
- να έχετε υποβάλλει τον κώδικα σας (RTL Design και Testbench) για κάθε σκέλος υλοποίησης ξεχωριστά,
- να έχετε υποβάλλει μια εργαστηριακή αναφορά, βάση των σημειώσεων από το βιβλίο του εργαστηρίου, με **όλη την απαραίτητη πληροφορία** (Dataflow, FSM State Graph(s), Simulation Screenshot(s), etc.).

## 8 Ερωτήσεις και Απορίες

Για οποιεσδήποτε ερωτήσεις και απορίες εκμεταλλευτείτε τον ιστιότοπο του e-Class και το χρόνο του εργαστηρίου ρωτώντας τους επιτηρητές.