

Υλοποίηση Μονάδας Γενικού Ασύγχρονου Δέκτη Αποστολέα (UART)

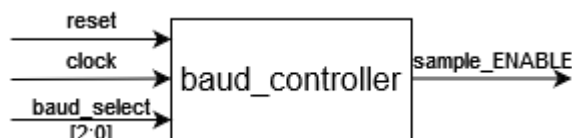
Εργαστήριο Ψηφιακών Συστημάτων (2023-24)

Ιωάννης Αθανασιάδης 03491

15/11/2023

Μέρος Α – Ελεγκτής Baud Rate

Το *module* αυτό το μοιράζονται ο Δέκτης και Αποστολέας και ο στόχος του είναι να παρέχει στο κύκλωμα τον κατάλληλο **παλμό δειγματοληψίας** ανάλογα με το επιλεγμένο **Baud Rate**.



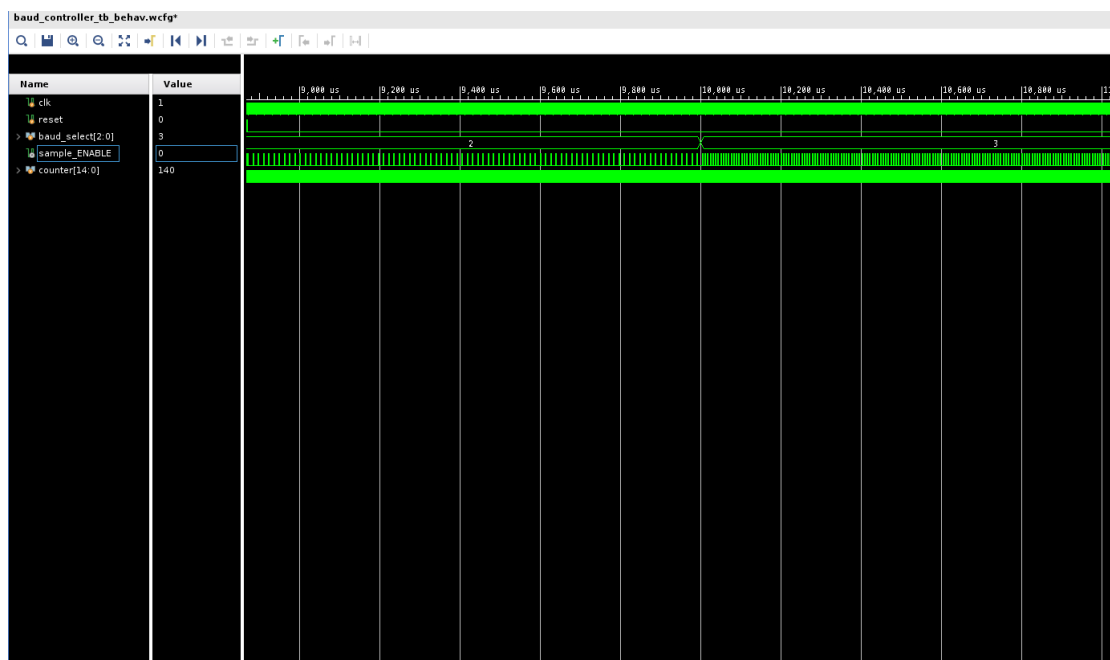
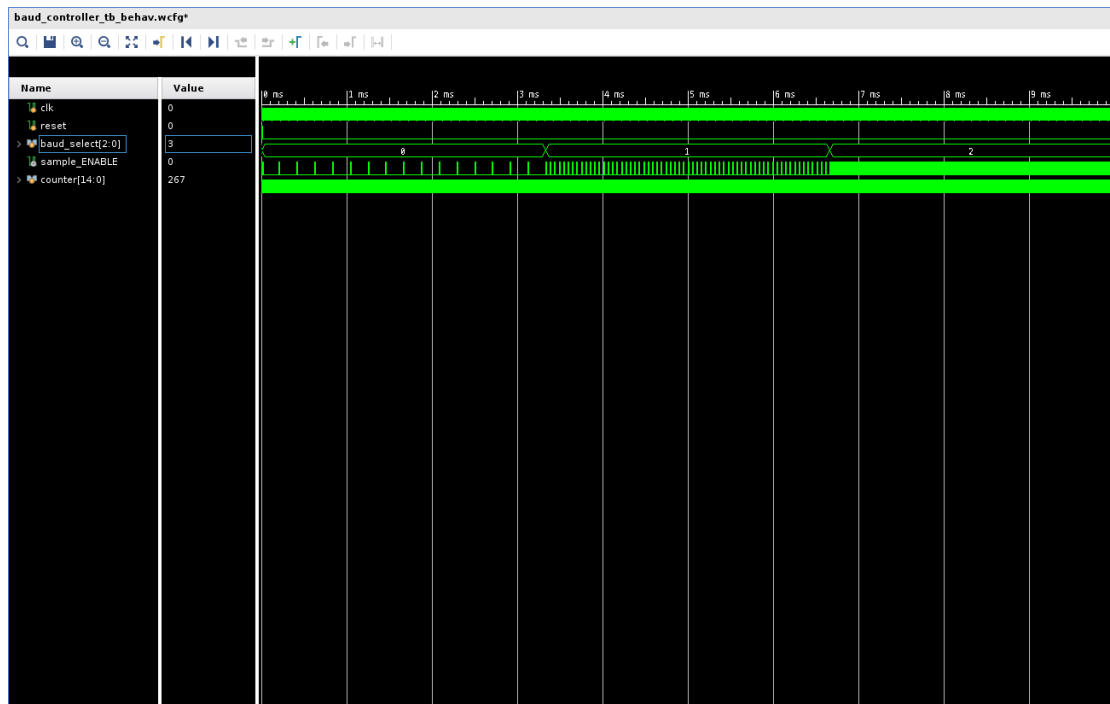
Δεδομένου ότι έχουμε συχνότητα ρολογιού στα *100 MHz* πρέπει να βρούμε τις μέγιστες τιμές του μετρητή μας για να πετύχουμε το σωστό **Baud Rate**.

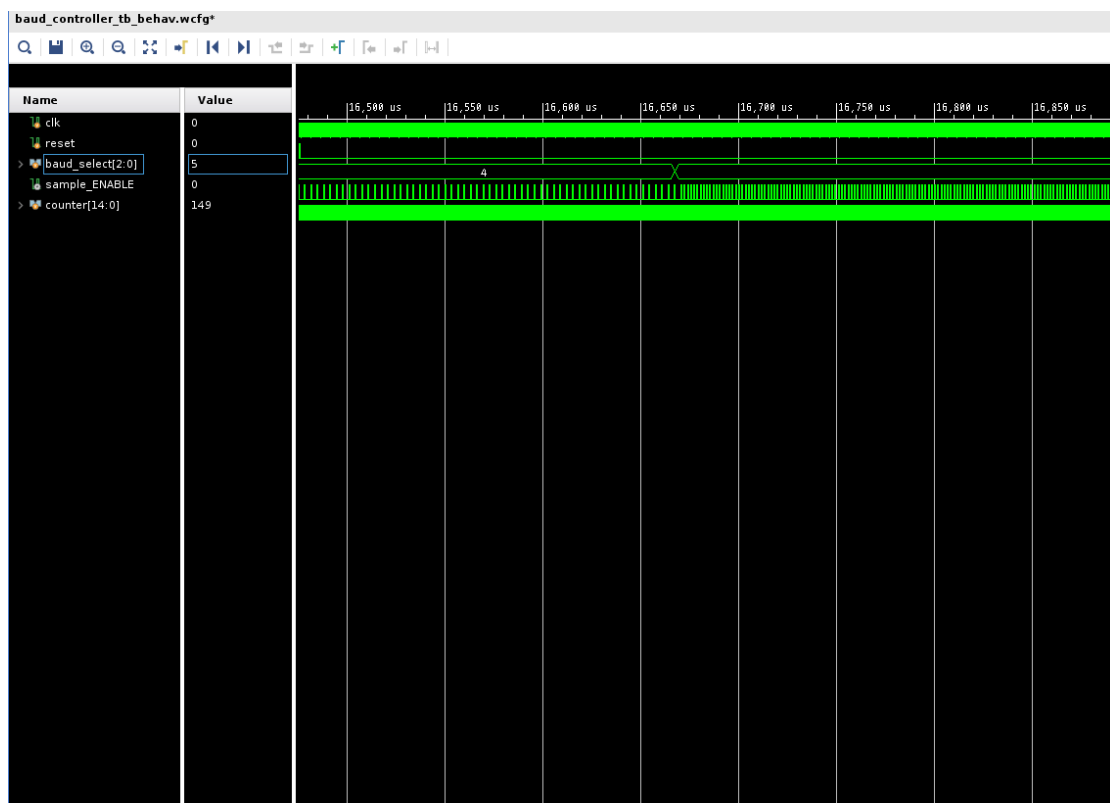
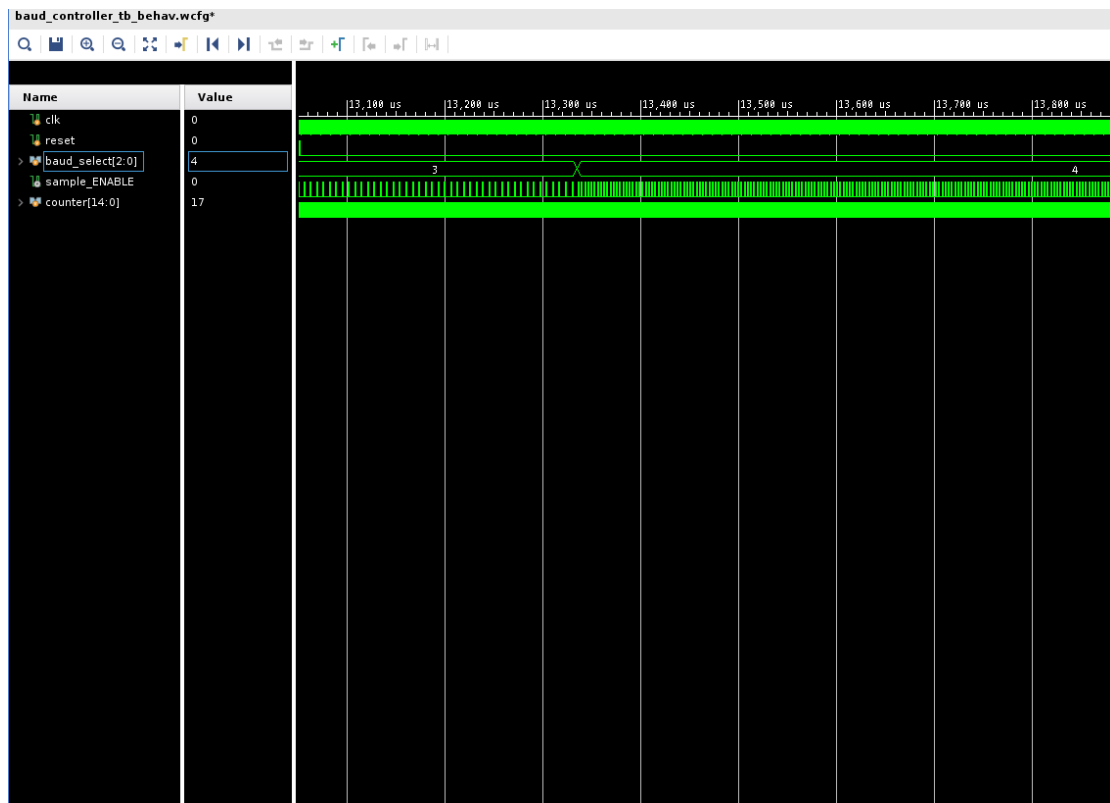
BAUD_SEL	Baud Rate	$T_{\text{Baud Rate}} / 16 \text{ (ns)}$	Τιμή Counter	Σχετικό Σφάλμα
000	300	208333.3	20833	20 ns
001	1200	52083.3	5208	20 ns
010	4800	13020.8	1302	13 ns
011	9600	6510.4	651	6 ns
100	19200	3255.2	325	83 ns
101	38400	1627.6	163	39 ns
110	57600	1085.1	108	81 ns
111	115200	542.5	54	40 ns

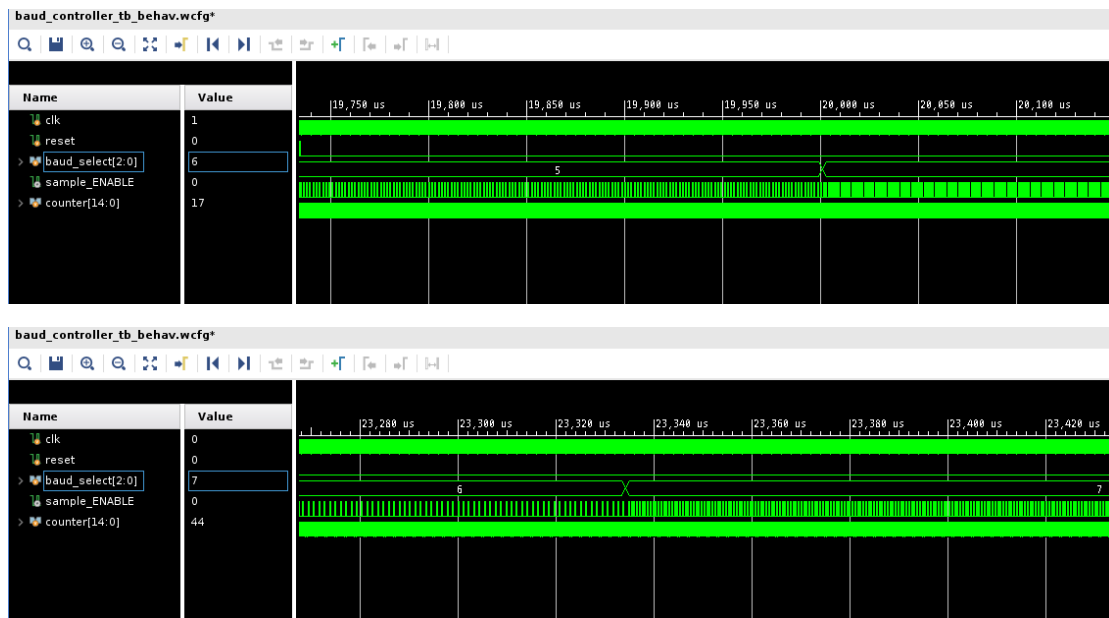
Παρατηρούμε ότι η μέγιστη τιμή του **counter** είναι το 20833 επομένως για να βρούμε το μέγεθος του μετρητή σε bit βρίσκουμε το άνω φράγμα του $\log_2 20833$, δηλαδή **15 bit**.

Σημείωση: Το **σχετικό σφάλμα** αν και σημαντικό, υπό τις ιδανικές καταστάσεις του εργαστηρίου δεν δημιουργεί μεγάλο πρόβλημα γιατί ο Αποστολέας και Δέκτης έχουν το ίδιο σφάλμα στην παραγωγή του **Baud Rate**.

Μονάδα Δοκιμής





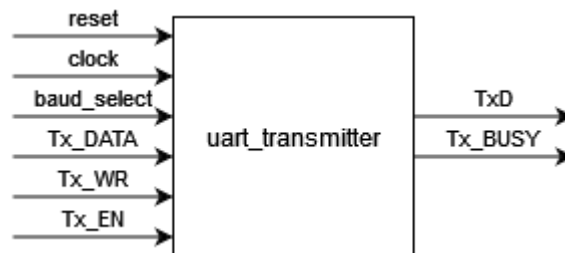


Μέρος Β – Υλοποίηση UART Αποστολέα (Transmitter)

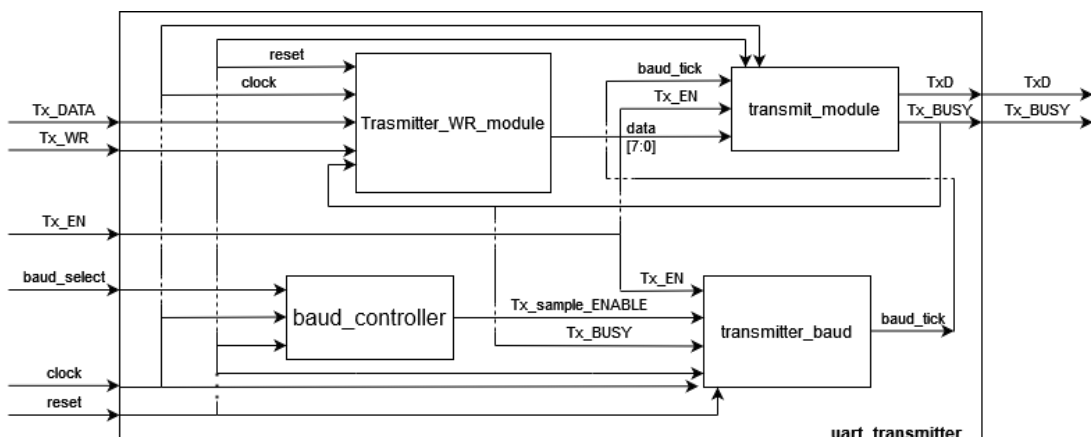
Το *top-level module* είναι το `uart_transmitter`:

Μονάδα `uart_transmitter`

Χρησιμοποιούμε αυτή την μονάδα για *top-level module* και αποτελεί ότι έχει να κάνει με την **μετάδοση** δεδομένων μέσω UART.

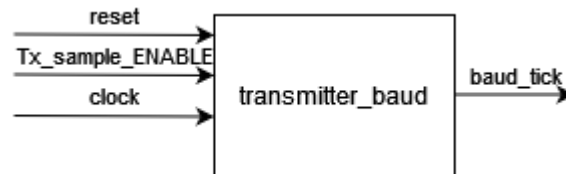


Η μονάδα αποτελείται από το module για την εγγραφή του μηνύματος, **`transmitter_WR_module`**, τις μονάδες για την παραγωγή του αργού και γρήγορου baud rate, **`baud_controller`** και **`transmitter_baud`** αντίστοιχα, και τέλος την **`transmit_module`** που αναλαμβάνει την αποστολή του μηνύματος.

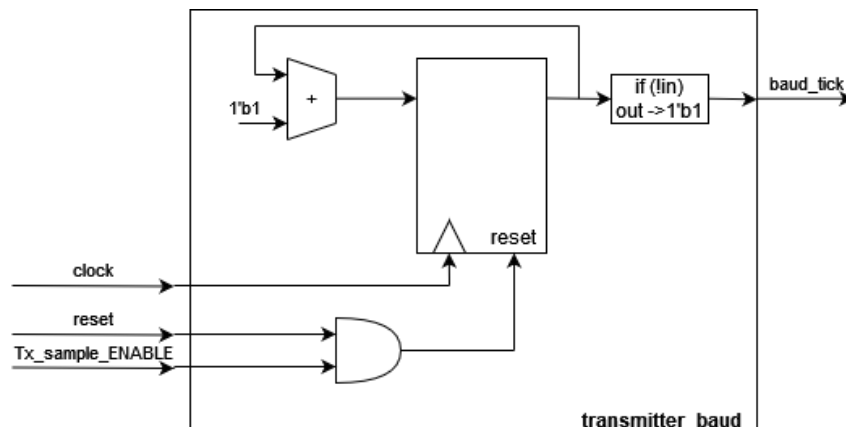


Μονάδα *transmitter_baud*

Το *module* αυτό χρησιμοποιείτε για να βρούμε το **σωστό** *baud rate* για τον Αποστολέα, δηλαδή για να το κάνουμε **16 φορές** αργότερο από την έξοδο του *baud_controller*.

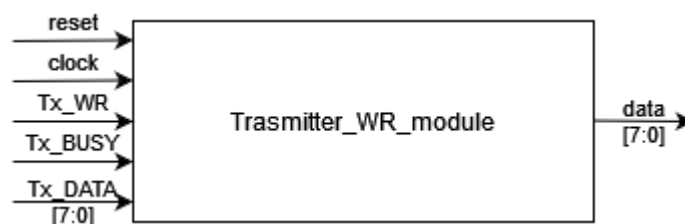


Πρακτικά αποτελείται από έναν μετρητή των **4-bit** που παράγει έναν θετικό παλμό, με διάρκεια ενός κύκλου.

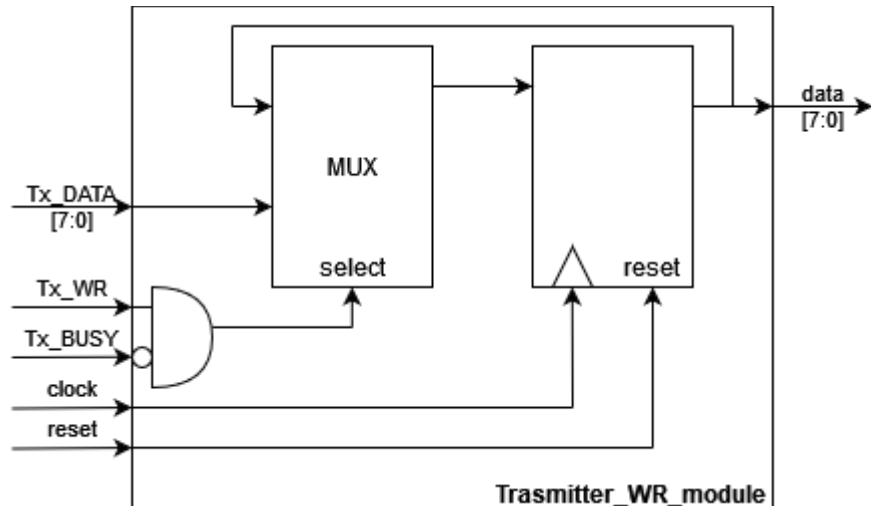


Μονάδα *Trasmitter_WR_module*

Η μονάδα αυτή είναι υπεύθυνη για την λήψη των **δεδομένων** από το **σύστημα**.

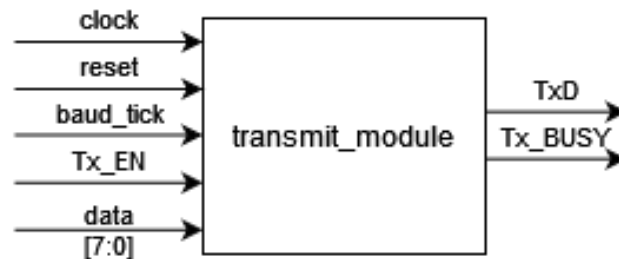


Εσωτερικά του *module* μεταφέρουμε το *Tx_DATA* στο κομμάτι της μνήμης *data* από το οποίο γίνεται η αποστολή του μηνύματος. Για να το κάνουμε αυτό ελέγχουμε αν ο αποστολέας είναι σε κατάσταση αποστολής δεδομένων και αν το σήμα *Tx_WR* είναι σηκωμένο.



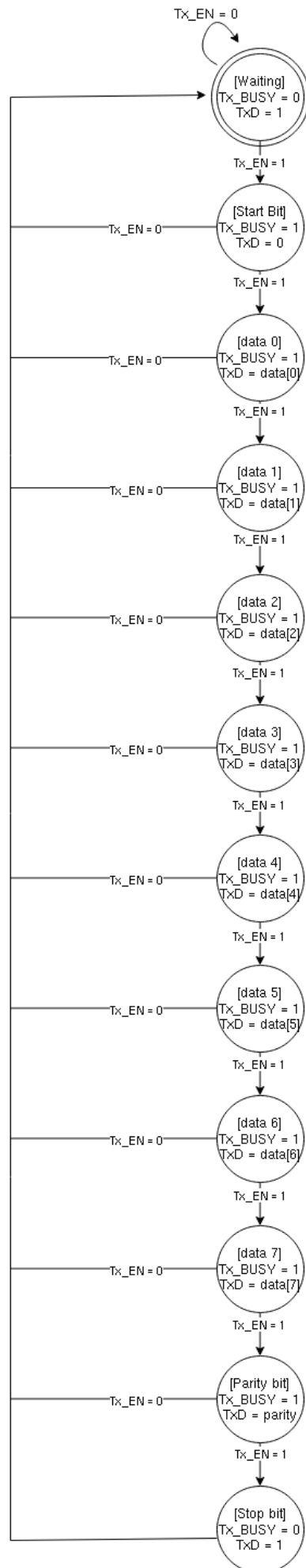
Μονάδα *transmit_module*

Αυτή η μονάδα αναλαμβάνει την **αποστολή δεδομένων** στην ενεργοποίηση του ***Tx_EN***.

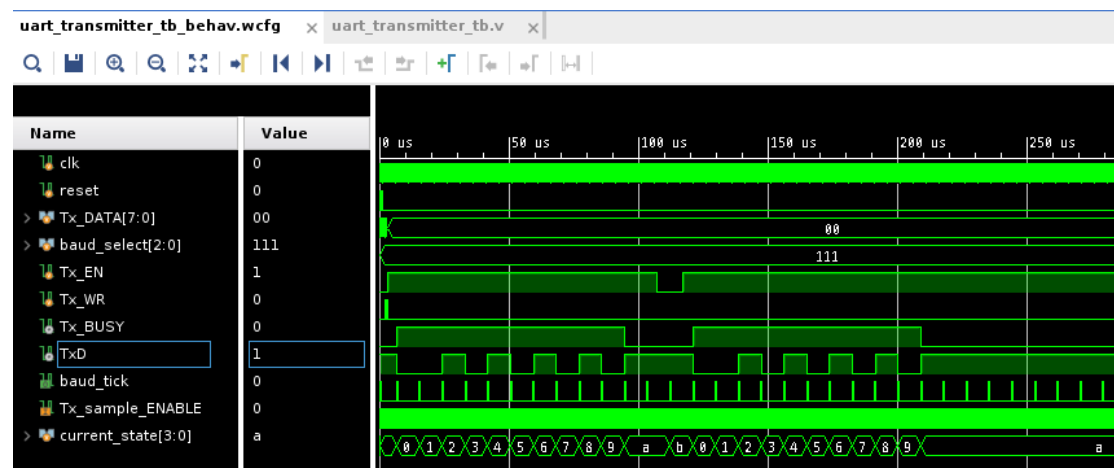


Σημείωση: Το ***Tx_EN*** σημαίνει έναρξη της αποστολής μηνύματος, και **όχι** του αποστολέα γενικότερα. Η **λήψη του μηνύματος** Tx_DATA από το σύστημα γίνεται με ***Tx_EN=0*** και ***Tx_WR=1***.

Εσωτερικά αυτό το *module* περιέχει την **FSM** για την μετάδοση του **data**. Παρακάτω ακολουθεί το **dataflow** της μηχανής καταστάσεων:



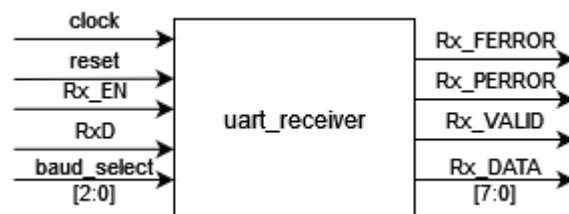
Μονάδα Δοκιμών



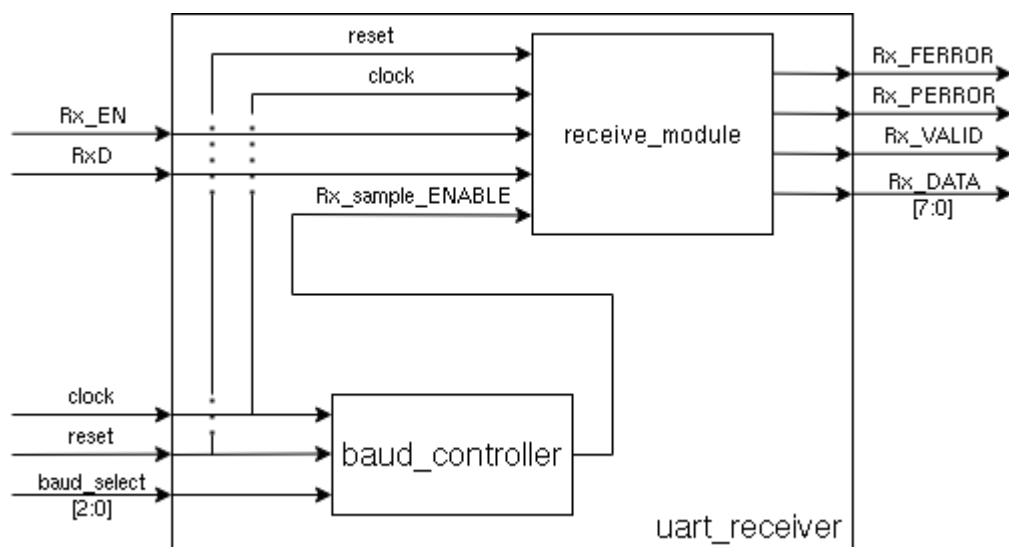
Μέρος Γ – Υλοποίηση UART Δέκτη (Receiver)

Η top-level μονάδα αυτού του μέρους είναι η *uart_receiver*.

Μονάδα *uart_receiver*

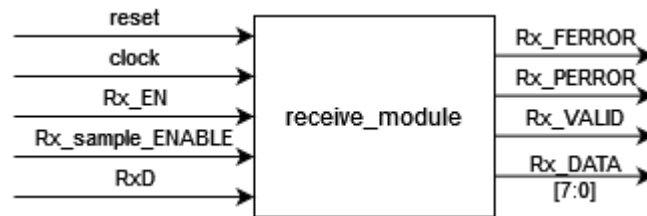


Η μονάδα αυτή εμφανίζει την **μηχανή καταστάσεων** για την λήψη των δεδομένων και το *module* ***baud_controller*** για την δημιουργία του *baud rate*.



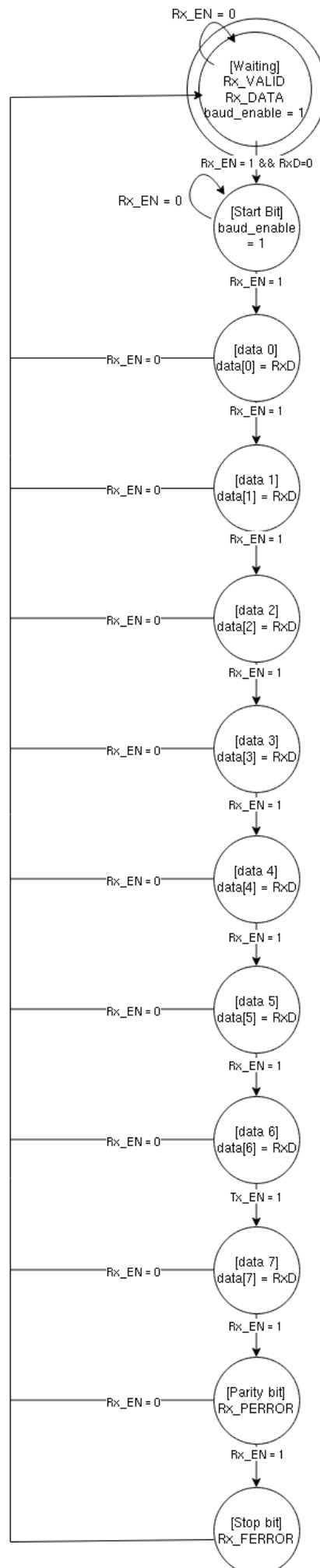
Μονάδα *receive_module*

Αυτή η μονάδα υλοποιεί την **μηχανή καταστάσεων** για την ανάγνωση του μηνύματος.

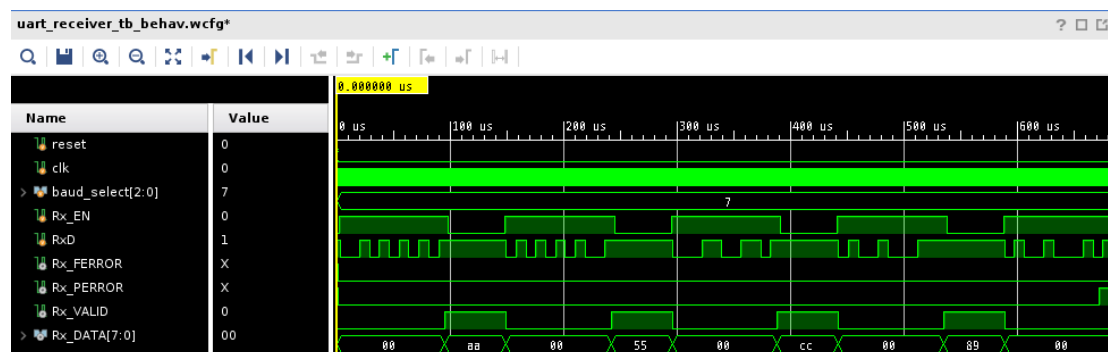


Η μονάδα πέρα από την *FSM* περιέχει την μονάδα ***receiver_baud***, με σήμα ενεργοποίησης *baud_enable*, η οποία παράγει το *baud rate* του *transmitter* με την σωστή μετατόπιση ώστε να γίνεται ανάγνωση στην μέση κάθε *bit*.

Σημείωση: Σε κάθε κατάσταση αναφέρονται τα **σήματα** που **αλλάζουν** ενδεχομένως τιμή. Για παράδειγμα το *Rx_VALID* και *Rx_DATA* στην κατάσταση ***waiting*** θα αλλάξουν τιμή μόνο αν έχουμε ***έγκυρη*** ανάγνωση μηνύματος.



Μονάδα Δοκιμών



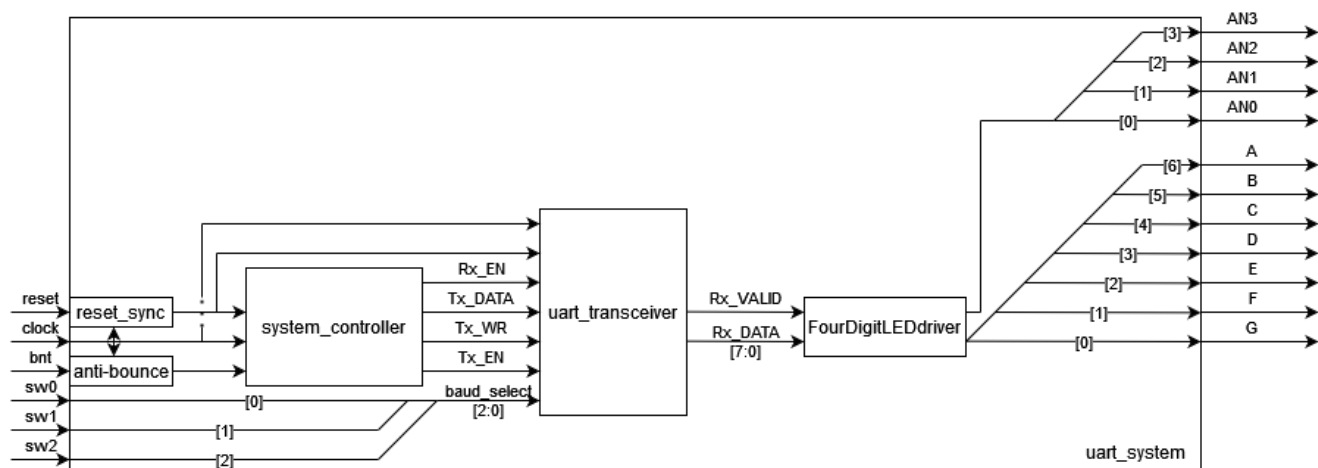
Μέρος Δ – Υλοποίηση UART Αποστολέα-Δέκτη (Transceiver)

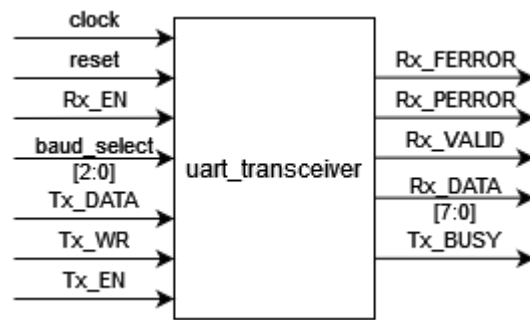
Για το *τελευταίο* μέρος της εργασίας ολοκληρώθηκε επιπλέον και το **προαιρετικό** κομμάτι της στο οποίο δοκιμάζουμε την χρήση του *transceiver* με εμφάνιση του μηνύματος που λαμβάνει ο δέκτης στην **7-Segment οθόνη** της προηγούμενης εργασίας.

Η επιλογή του **baud_rate** γίνεται με του διακόπτες **sw0, sw1, sw2**.

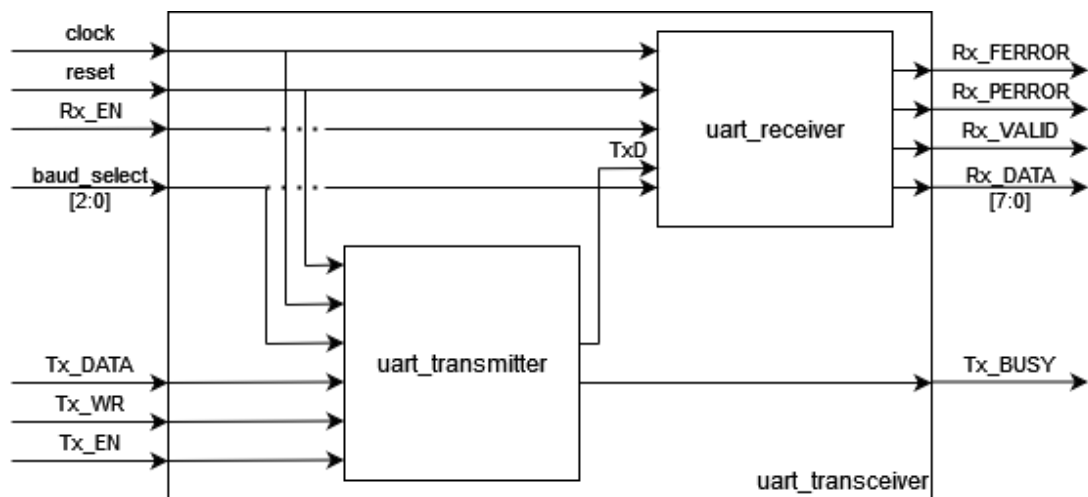
Μονάδα *uart_system*

Αυτή η μονάδα αποτελεί το *top-level* του κυκλώματος που θα τρέξει στην **FPGA**. Περιέχει κύκλωμα *συγχρονισμού* και *anti-bounce* για το κουμπί αποστολής νέου μηνύματος, κύκλωμα *συγχρονισμού* για το *reset button*, το κύκλωμα του *uart_transceiver* και το κύκλωμα διαχείρισης του συστήματος, *system_controller*.



Μονάδα *uart_transceiver*

Αυτή η μονάδα απλά εμφανίζει τις δύο μονάδες *uart_transmitter* και *uart_receiver* με σύνδεση *loop-back*, δηλαδή το *RxD* του ενός είναι το *TxD* του άλλου.

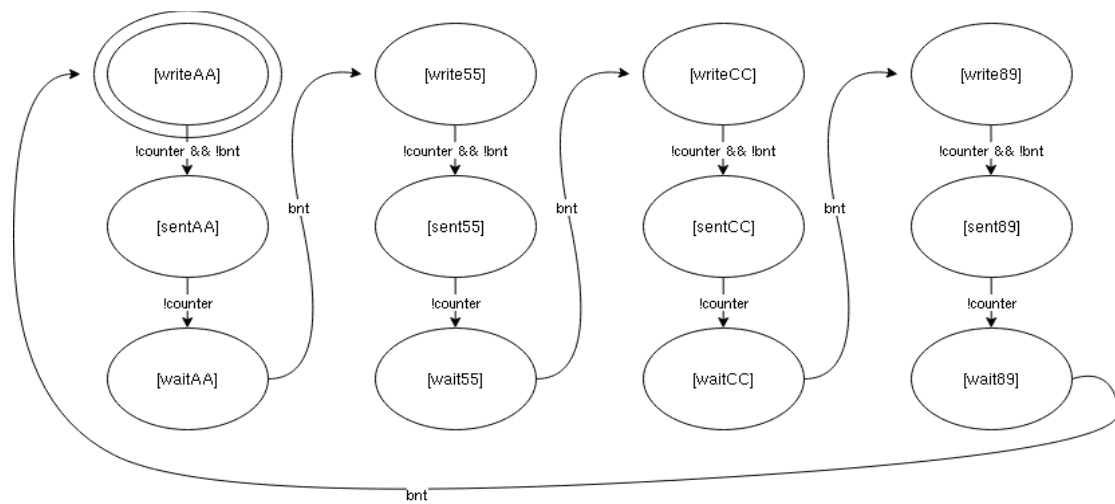
Μονάδα *system_controller*

Η μονάδα *system_controller* είναι μία **μηχανή καταστάσεων** ως αντικαταστάτης της *initial* που συμβατικά έχει ένα *testbench*, η οποία όμως δεν είναι *συνθέσιμη*. Χρησιμοποιούμε 25-bit counter για να μεταβούμε στην *επόμενη κατάσταση* με *καθυστέρηση* σε ορισμένες περιπτώσεις.

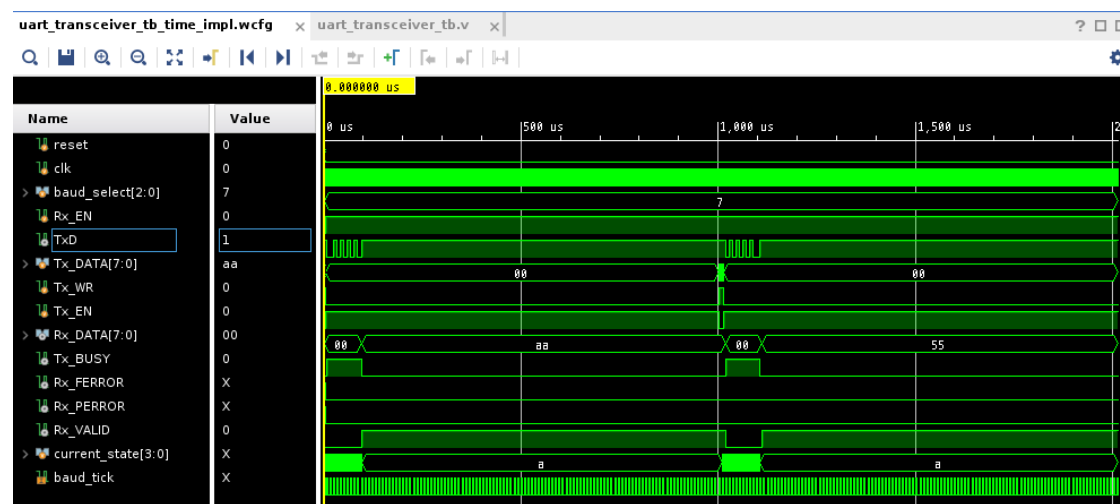
Για κάθε μήνυμα υπάρχουν **3 βασικές καταστάσεις**:

1. Την **write**, όπου γράφουμε μέσω του *Tx_WR* το νέο μήνυμα που θέλουμε να στείλουμε,
2. Την **sent**, όπου στέλνουμε το μήνυμα με την ακμή των σημάτων *Rx_EN* και *Tx_EN*,
3. Και την **wait**, όπου περιμένει τον χρήστη να πατήσει το κουμπί επόμενου μηνύματος στο ενδιάμεσο του οποίου η οθόνη μας προβάλλει την δεκαεξαδική αναπαράσταση του μηνύματος που στείλαμε.

Πιο αναλυτικά στέλνουμε τα μηνύματα **0xAA**, **0x55**, **0xCC** και **0x89**:



Μονάδα Δοκιμών



Η μονάδα δοκιμών που επιδεικνύεται είναι μόνο της μονάδας του *uart_transceiver* όπως παρουσιάστηκε στην εξέταση του εργαστηρίου.

Σημείωση: Έχω επισυνάψει ένα βίντεο που δείχνει εν-λειτουργία το κύκλωμα πάνω στην **FPGA**.