ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΜΑΘΗΜΑ:
Ψηφιακά Συστήματα HW σε Χαμηλά Επίπεδα Λογικής Ι

Τίτλος Εργασίας:
Υλοποίηση Συστήματος Αποστολέα-Δέκτη σε Verilog με χρήση Πρωτοκόλλου UART και προβολή σε Οθόνη Τεσσάρων LED 7-Τμημάτων

Μπουλέτσης Αλέξης
Τσαμουρίδης Αναστάσιος Αθανάσιος

ΠΕΡΙΕΧΟΜΕΝΑ

Εισαγωγή **2**

Μέρος Α – 7-Segment Display Driver **3**

Μέρος Β – Υλοποίηση Γενικού Ασύγχρονου Δέκτη και Αποστολέα (UART) **7**

Ελεγκτής Baud Rate 7

UART Αποστολέας 10

UART Δέκτης 12

UART Αποστολέας-Δέκτης 15

Μέρος Γ – Σύστημα Δέκτη με προβολή του μηνύματος στην οθόνη τεσσάρων LED 7-τμημάτων **18**

Υλοποίηση Register με ειδικές λειτουργίες 18

Συνένωση Δέκτη-Οθόνης 19

Συνένωση όλων των επιμέρους τμημάτων για την αποστολή - παραλαβή και προβολή μηνύματος 22

Μέρος Δ – Υλοποίηση Κωδικοποιητή-Αποκωδικοποιητή μεμονωμενη λειτουργία κωδικοποιητή - αποκωδικοποιητή **26**

Συνένωση με παρουσία Κωδικοποιητή και αποκωδικοποιητή 28

Εισαγωγή

Στην παρούσα αναφορά γίνεται παρουσίαση της υλοποίησης ενός συστήματος αποστολέα - δέκτη με βάση το πρωτόκολλο UART για την παραλαβή δεδομένων προερχόμενα από κάποιους αισθητήρες. Επιπρόσθετα, έγινε υλοποίηση ενός συστήματος τεσσάρων οθονών φωτοδιόδων 7 τμημάτων (7-segment-LED-displays), για την παρουσίαση των απεσταλμένων μηνυμάτων ή μηνύματος λάθους, στην περίπτωση παραλαβής μη προσδοκώμενου σήματος, μετά την παραλαβή των δεδομένων από τον δέκτη. Τέλος υλοποιήθηκε μια απλή κωδικοποίηση για την προστασία των απεσταλμένων μηνυμάτων.

Η παρουσίαση αποτελείται από τέσσερα μέρη:

- Μέρος Α: Γίνεται η παρουσίαση της υλοποίησης του LED-κωδικοποιητή και του οδηγητή των ανόδων των οθονών.
- Μέρος Β: Παρουσιάζεται η υλοποίηση του ασύγχρονου συστήματος αποστολέα και δέκτη, οι οποίοι ακολουθούν το πρωτόκολλο UART.
- Μέρος Γ: Γίνεται η συνένωση των μερών Α και Β και καταδεικνύεται η ορθή λειτουργία του συστήματος.
- Μέρος Δ: Στην υλοποίηση του μέρους Γ, προστίθεται η κωδικοποίηση των απεσταλμένων σημάτων και η αποκωδικοποίηση τους μετά την παραλαβή τους από τον δέκτη.

Όλες οι υλοποιήσεις έγιναν για λειτουργία ρολογιού με περίοδο 20 nsec. Επίσης σε κάθε module υπάρχει σήμα reset.

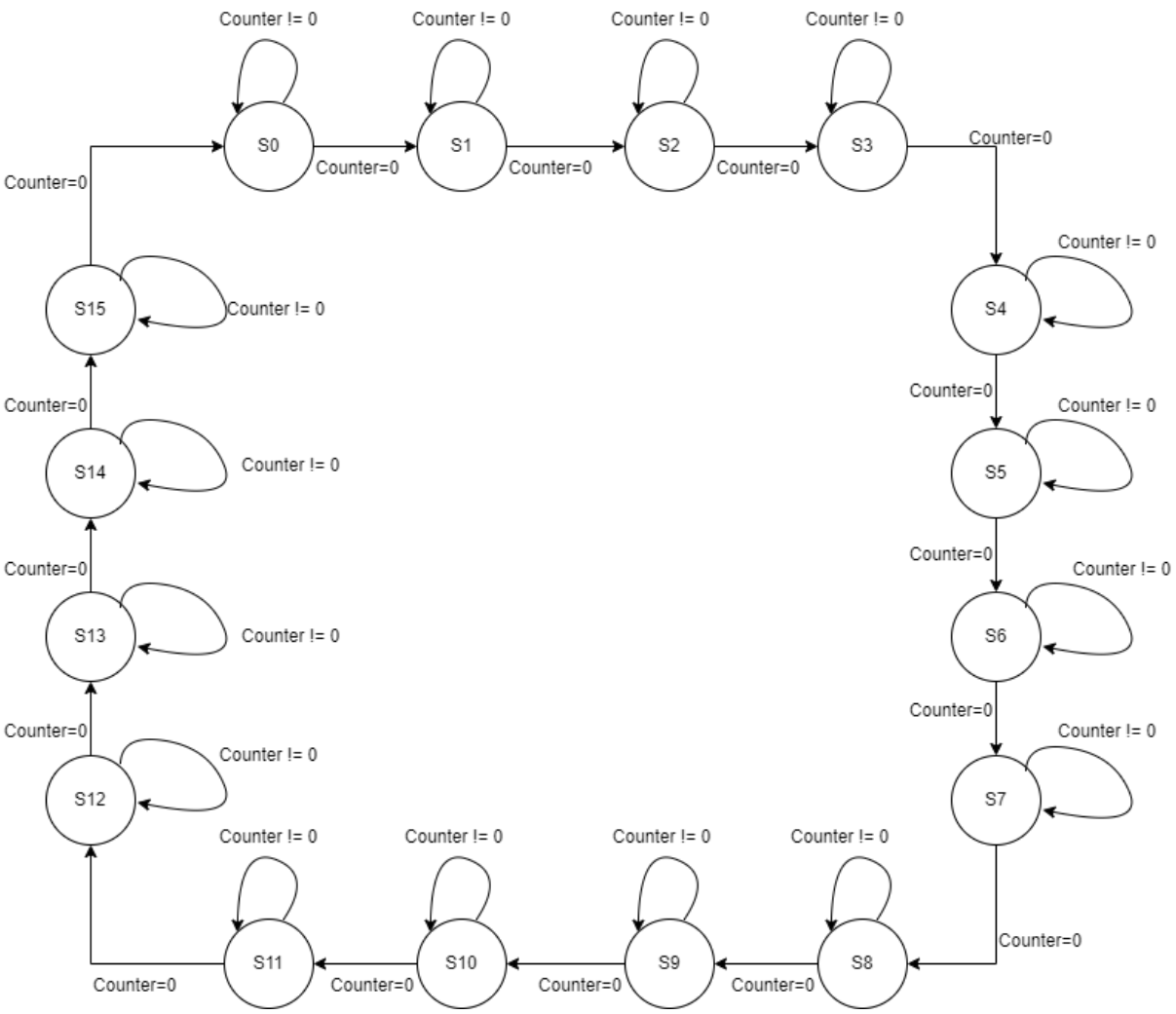
Μέρος Α – 7-Segment Display Driver

Για την υλοποίηση του κωδικοποιητή έγινε η παραδοχή ότι τα απεσταλμένα μηνύματα κωδικοποιούνται με βάση των παρακάτω πίνακα:

Χαρακτήρας	Κωδικοποίηση	Έξοδος του LED decoder
0	00 00	000 000 1
1	00 01	100 111 1
2	00 10	001 001 0
3	00 11	000 011 0
4	01 00	100 110 0
5	01 01	010 010 0
6	01 10	010 000 0
7	01 11	000 111 1
8	10 00	000 000 0
9	10 01	000 010 0
-	10 10	111 111 0
F	10 11	011 100 0
[κενό]	11 00	111 111 1

Πίνακας: οι έξοδοι του LED Decoder και η αντίστοιχη κωδικοποίηση των χαρακτήρων

Με βάση τα παραπάνω δημιουργήθηκε το module του LED-decoder. Για την οδήγηση των ανόδων έγινε χρήση της παρακάτω μηχανής πεπερασμένων καταστάσεων (FSM):

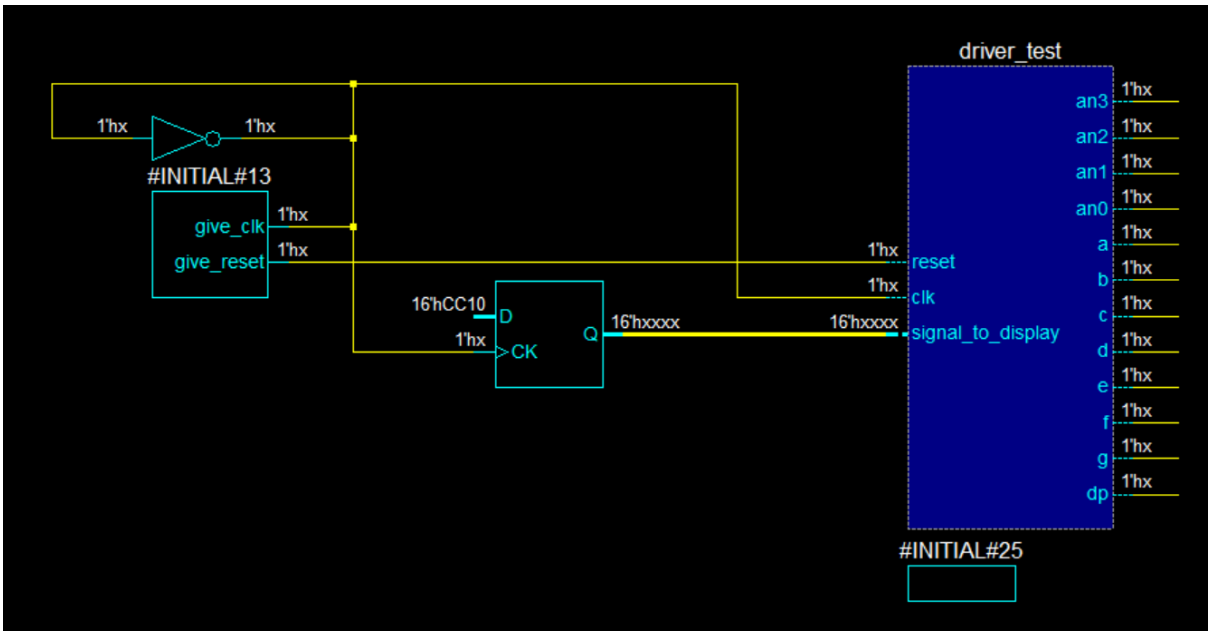


Εικόνα: Το FSM του LED Driver

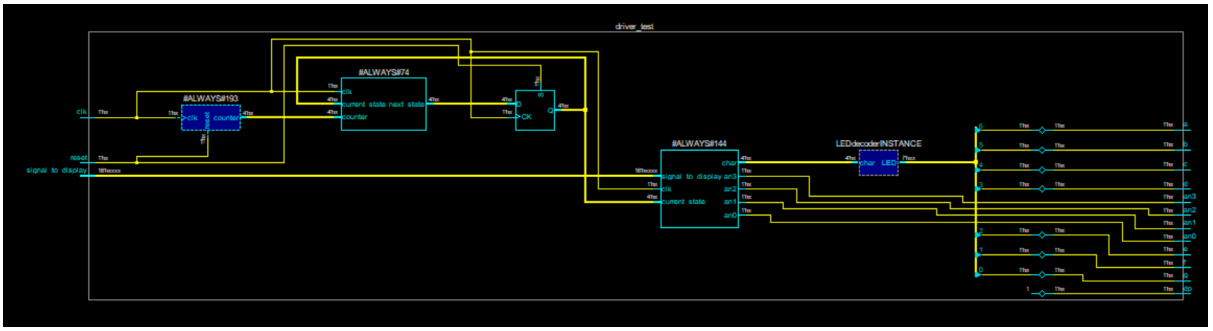
Όπως φαίνεται η μηχανή λειτουργεί κυκλικά για 16 καταστάσεις. Η μετάβαση από την μία κατάσταση στην επόμενη γίνεται με τον μηδενισμό του περιστροφικού μετρητή,ο οποίος αυξάνεται σε κάθε θετική ακμή του ρολογιού (με μέγιστη τιμή μέτρησης το 15 και εκκίνηση από το 0) έτσι, ώστε :

- Οι οθόνες να δέχονται είσοδο από έναν LED κωδικοποιητή και να ανάβουν διαδοχικά ο ένας μετά τον άλλον προβάλλοντας το σωστό μήνυμα ο καθένας και χωρίς να υπάρχει επικάλυψη των προβαλλόμενων χαρακτήρων
- Η κάθε μια από τις 4 ανόδους των οθονών (AN0-3) να παραμένει ενεργή για t = 320 nsec.
- Να μεσολαβεί ένα χρονικό διάστημα (960 nsec = 3t), ως περιθώριο ασφαλείας, στο οποίο καμία οθόνη δεν θα είναι ενεργοποιημένη ώστε να μην υπάρξει επικάλυψη των σημάτων και αλλοιωθούν οι ενδείξεις.

Το σχηματικό που παράγεται από το εργαλείο προσομοίωσης παρατίθεται στην παρακάτω εικόνα:



Εικόνα: Σχηματικό του testbench του LED driver



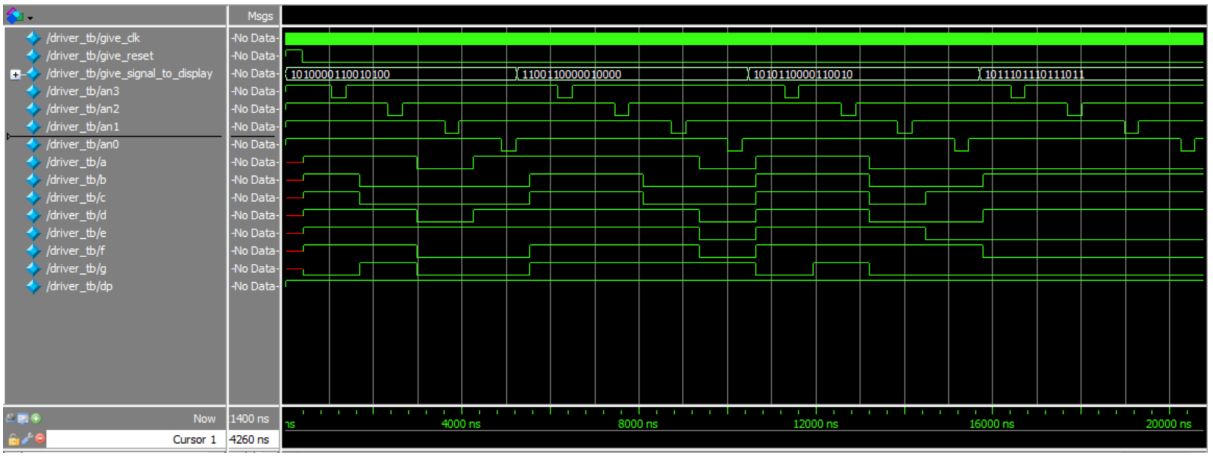
Εικόνα: Σχηματικό του LED driver

Όπως είναι εμφανές ο driver δέχεται ως είσοδο το 16-bit σήμα signal_to_display, το οποίο είναι το σύνολο των τεσσάρων 4-bit χαρακτήρων που θα πρέπει να εμφανιστούν στην έξοδο. Ως έξοδοι, δίνονται οι τιμές των ανόδων και τα σήματα των τμημάτων a-g/dp.

Οι παραγόμενες κυματομορφές της προσομοίωσης για τα σήματα εισόδου:

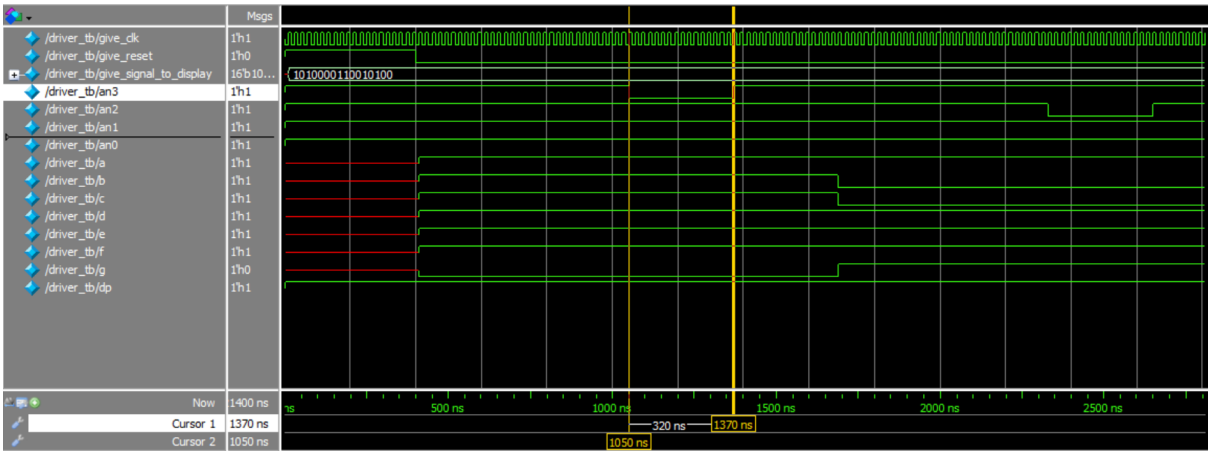
- ‘-194’
- ‘ 10’
- - 32’
- ‘FFFF’

φαίνονται παρακάτω:



Εικόνα: Κυματομορφές της προσομοίωσης για τα σήματα εισόδου ‘-194’, ‘ 10’, - 32’, ‘FFFF’.

Στο παρακάτω, επισημαίνεται για το σήμα ‘-194’, η χρονική διάρκεια της ανόδου AN3 για τον χαρακτήρα ‘-’:



Εικόνα: Έξοδος κυκλώματος για την ενεργοποίηση της AN3.

Είναι εμφανές ότι οι άνοδοι ενεργοποιούνται διαδοχικά, με την παρουσία ενός χρονικού περιθωρίου ενδιάμεσα στις ενεργοποιήσεις. Οι τιμές των εξόδων του κωδικοποιητή a, b, c, d, e, f, g παίρνουν κατάλληλες τιμές για να εμφανίσουν στην έξοδο τον κατάλληλο χαρακτήρα πριν την ενεργοποίηση της εκάστοτε ανόδου. Η έξοδος dp παραμένει διαρκώς στο 1, διότι δεν είναι επιθυμητή η εμφάνιση υποδιαστολής σε οποιαδήποτε ένδειξη.

Μέρος Β – Υλοποίηση Γενικού Ασύγχρονου Δέκτη και Αποστολέα (UART)

Για το μέρος Β, έγινε υλοποίηση ενός ελεγκτή Baud Rate και έπειτα ακολούθησε η υλοποίηση των Αποστολέα, Δέκτη και τέλος η συνένωσή τους.

Ελεγκτής Baud Rate

Ο ελεγκτής Baud Rate χρησιμοποιείται ως υποκύκλωμα στα κυκλώματα του Αποστολέα και του Δέκτη και ορίζει τον ρυθμό μετάδοσης των συμβόλων και δειγματοληψίας στον δέκτη. Δέχεται ως είσοδο ένα 3-bit σήμα baud_select, το οποίο καθορίζει το Baud Rate. Το Baud Rate ορίζει την περίοδο που διαρκεί η αποστολή ενός συμβόλου (T). Το Baud Select θα πρέπει να είναι ίδιο για δέκτη και αποστολέα ώστε να επιτευχθεί η επικοινωνία χωρίς πρόβλημα. Ως έξοδος δίνεται ένα σήμα με περίοδο προσεγγιστικά ίση με $T' = \frac{1}{16 \cdot Baud\ Rate}$. Η προσέγγιση αφορά την αδυναμία ακρίβειας λόγω της χρήσης του ρολογιού των 50 MHz. Πιο αναλυτικά, θα πρέπει η περίοδος $\mathrm{T'(καθώς\ και\ η\ ημιπερίοδος)}$ να είναι πολλαπλάσιο της περιόδου του ρολογιού ώστε να μπορεί να επιτευχθεί με ακρίβεια η έξοδος του ελεγκτή Baud Rate. Ωστόσο, για την υλοποίηση του ελεγκτή Baud Rate χρησιμοποιήθηκαν οι παρακάτω επιλογές, οι οποίες δεν είναι ακέραιο πολλαπλάσιο του ρολογιού, οδηγώντας σε σφάλματα προσέγγισης. Στον πίνακα φαίνεται και η αντιστοίχιση των Baud Rate με τα baud_select.

baud_select	Baud Rate (bits/sec)
000	300
001	1200
010	4800
011	9600
100	19200
101	38400
110	57600
111	115200

Πίνακας: Η κωδικοποίηση των Baud Rate

Για την υλοποίηση του κώδικα *Verilog* έγιναν οι υπολογισμοί που φαίνονται στον παρακάτω πίνακα, οποίοι δίνουν το ελάχιστο δυνατό σφάλμα για την προσέγγιση του

Baud Rate. Οι τιμές υπολογίστηκαν μέσω κώδικα *MATLAB* και μπορούν να βρεθούν στο αρχείο *Baud_rate_calculations.m*.

BAUD_ SEL	**BAUD RATE BR (bits/sec)**	**Επιθυμητή περίοδος T'** $(T'_{desired} = \frac{1}{16 \cdot BR})$	**Τελική τιμή της περιόδου T'** $(T' = \frac{1}{16 \cdot BR})$	**Clocks ως την αναστροφή του sample_ENABLE**	**Clocks ανά περίοδο T'** $(T' = \frac{1}{16 \cdot BR})$
000	300	208333.333	208320	5208	10416
001	1200	52083.333	52080	1302	2604
010	4800	13020.8333	13040	326	652
011	9600	6510.41666	6520	163	326
100	19200	3255.208333	3240	81	162
101	38400	1627.604166	1640	41	82
110	57600	1085.069	1080	27	54
111	115200	542.53	560	14	28

Πίνακας: baud_select και προσεγγίσεις του Baud Rate

Με βάση τα παραπάνω προκύπτουν τα παρακάτω σφάλματα:

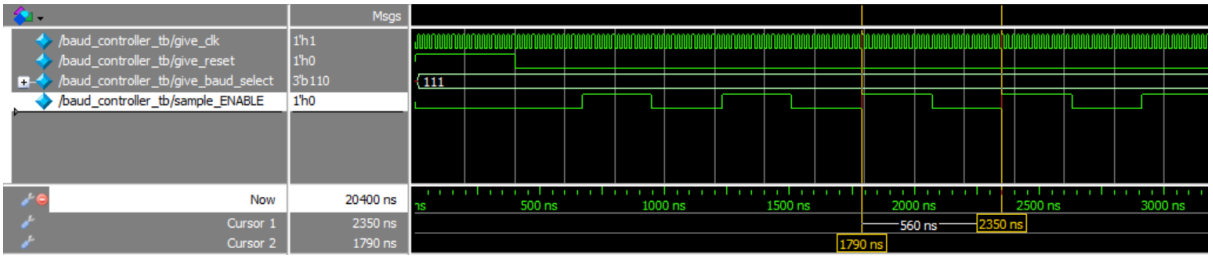
BAUD_SEL	**Επιθυμητή περίοδος T'** $(T'_{desired} = \frac{1}{16 \cdot BR})$	**Τελική τιμή της περιόδου T'** $(T' = \frac{1}{16 \cdot BR})$	**Ποσοστιαίο Σφάλμα (%)**
000	208333.333	208320	0.0064
001	52083.333	52080	0.0064
010	13020.8333	13040	0.1472
011	6510.41666	6520	0.1472
100	3255.208333	3240	0.4672
101	1627.604166	1640	0.7616
110	1085.069	1080	0.4672

111	542.53	560	3.2192

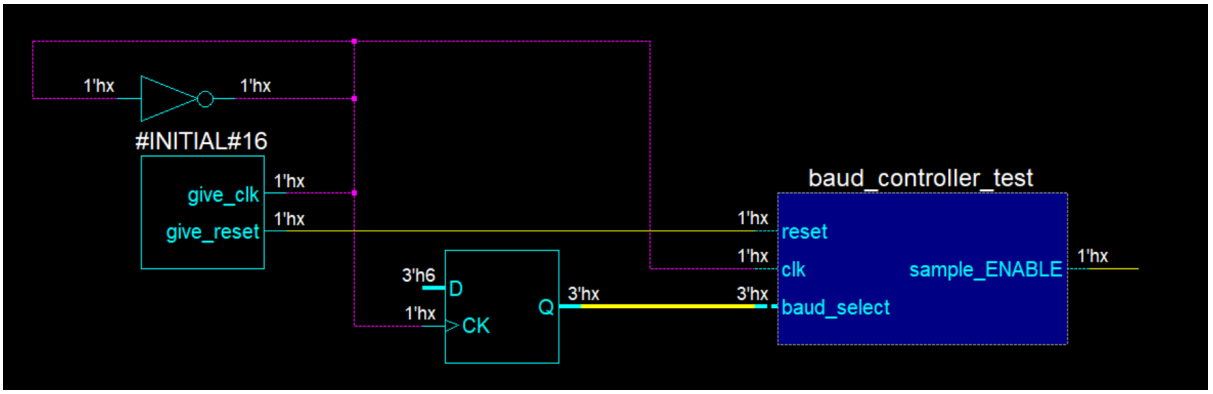
Εικόνα: Σχετικά σφάλματα στους υπολογισμούς του Baud Rate

Φαίνεται ότι τα σχετικά σφάλματα είναι μικρότερα του 5% και η απόκλιση από την επιθυμητή περίοδο δεν μεγάλη. Αυτό, μπορεί να έχει σαν αποτέλεσμα την πιο γρήγορη ή πιο αργή μετάδοση των συμβόλων, ανάλογα με το αν η προσέγγιση έγινε προς μεγαλύτερες ή μικρότερες τιμές του $T'_{desired}$.

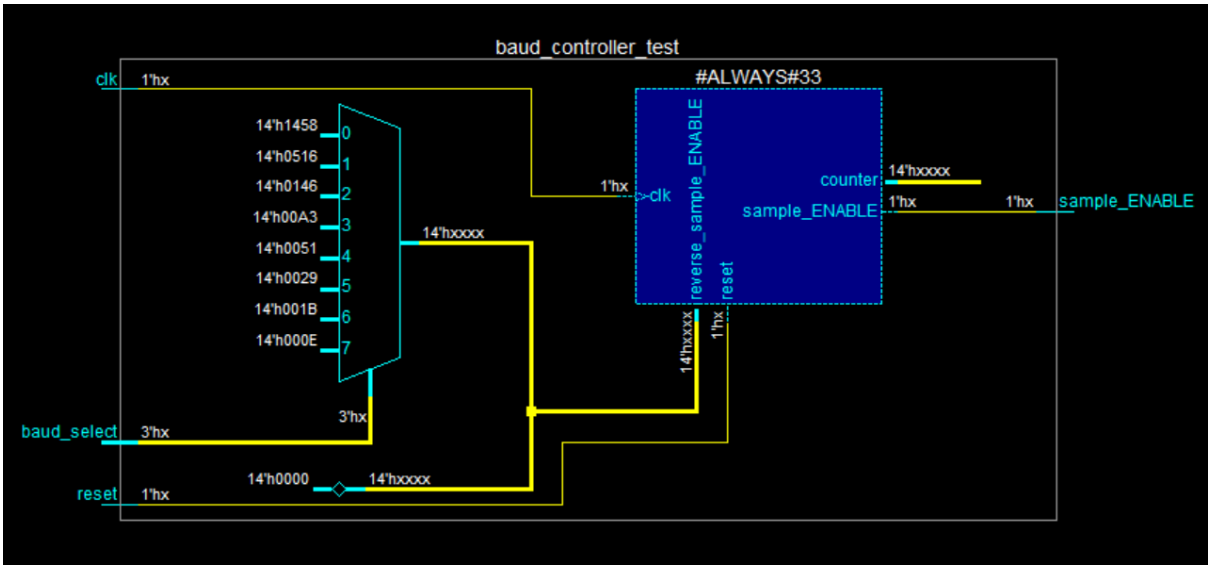
Στην παρακάτω εικόνα φαίνεται η λειτουργία του ελεγκτή για baud_select 111 και πιο κάτω το σχηματικό.



Εικόνα: Κυματομορφές του baud_controller



Εικόνα: Σχηματικό του testbench του baud_controller

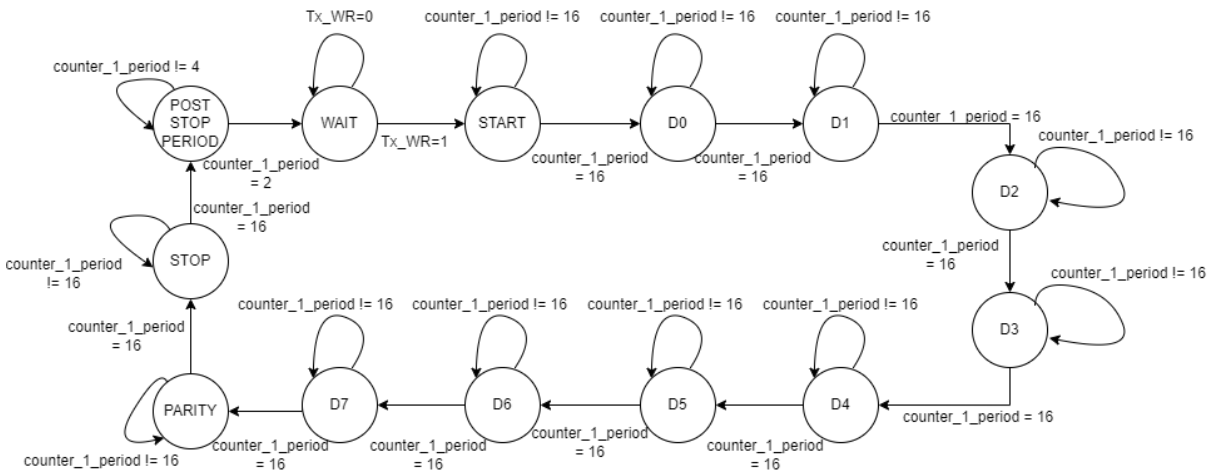


Εικόνα: Σχηματικό του Baud controller

UART Αποστολέας

Ο UART Αποστολέας χρησιμοποιείται για την αποστολή δεδομένων στον δέκτη. Συγκεκριμένα στέλνει 1 start bit, 8-bit δεδομένων, 1 parity bit και 1 stop bit, συνολικά 11 bit. Δέχεται ως είσοδο το 3-bit σήμα baud_select, τα 8-bit δεδομένα Tx_DATA, το σήμα Tx_EN, το οποίο είναι το σήμα ενεργοποίησης του αποστολέα και το οποίο μένει πάντα ενεργό (1) στα testbench. Το σήμα Tx_WR γίνεται 1 όταν ζητείται από τον αποστολέα να διαβάσει τα δεδομένα Tx_DATA. Οι έξοδοι του αποστολέα είναι το σήμα Tx_BUSY, το οποίο όσο μένει 1 δηλώνει ότι ο αποστολέας στέλνει τα δεδομένα στον δέκτη. Το σήμα TxD είναι η σειριακή έξοδος, η οποία οδηγείται στην είσοδο του δέκτη.

Για την υλοποίηση του αποστολέα υλοποιήθηκε το παρακάτω FSM:



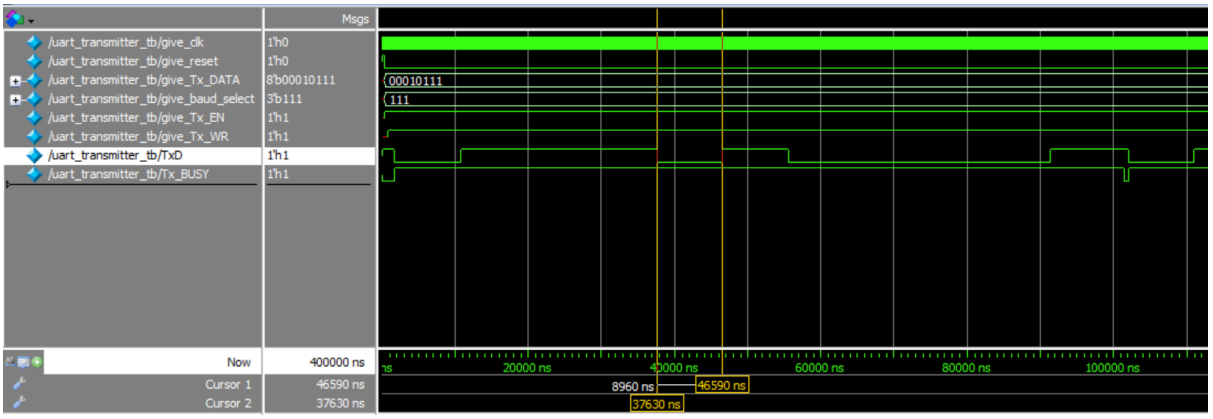
Εικόνα: Το FSM του Αποστολέα

Όπως φαίνεται η μηχανή λειτουργεί κυκλικά για 13 καταστάσεις. Η μετάβαση από την κατάσταση WAIT στην κατάσταση START γίνεται όταν το σήμα Tx_WR πάρει την τιμή 1. Από την κατάσταση START στην επόμενη κατάσταση D0 η μετάβαση γίνεται όταν ο counter_1_period γίνει 16 δηλαδή όταν μετρηθούν 16 θετικές ακμές του ρολογιού. Ομοίως, για την μετάβαση από την κατάσταση D0 στην D1 έως και την POST_STOP_PERIOD. Η μετάβαση από την κατάσταση POST_STOP_PERIOD στην WAIT γίνεται όταν ο counter_1_period γίνει 4, δηλαδή όταν μετρήσει 4 θετικές ακμές του ρολογιού.

- WAIT : Αποθηκεύονται τα 8-bit από το Tx_DATA σε ένα προσωρινό reg temp_data, ορίζεται το σήμα Tx_BUSY 0, αρχικοποιείται ο counter_1_period και το σήμα TxD με την τιμή 1.
- START : Στέλνεται το start bit, το οποίο έχει τιμή 0 για να σηματοδοτήσει την έναρξη της επικοινωνίας. Το σήμα Tx_BUSY παίρνει την τιμή 1.
- D0-D7 : Στέλνονται τα 8-bit δεδομένων από το LSB στο MSB.

- PARITY : Στην κατάσταση parity υπολογίζεται το πλήθος των άσσων που στάλθηκαν από τις καταστάσεις D0 - D7 το οποίο είναι το parity bit. Εδώ σημειώνεται πως το bit ισοτιμίας είναι 1 για άρτιο πλήθος άσσων των δεδομένων και 0 για περιττό. Επομένως υλοποιήθηκε άρτια ισοτιμία.
- STOP: Στέλνεται το stop bit, το οποίο έχει τιμή 1, οπότε και σηματοδοτείται η λήξη της επικοινωνίας.
- POST_STOP_PERIOD : Ο αποστολέας στέλνει το bit 1. Η κατάσταση αυτή αποσκοπεί στην αποφυγή της επικάλυψης του start bit με το Rx_VALID, του δέκτη, γεγονός που θα οδηγούσε σε απώλεια πληροφορίας.

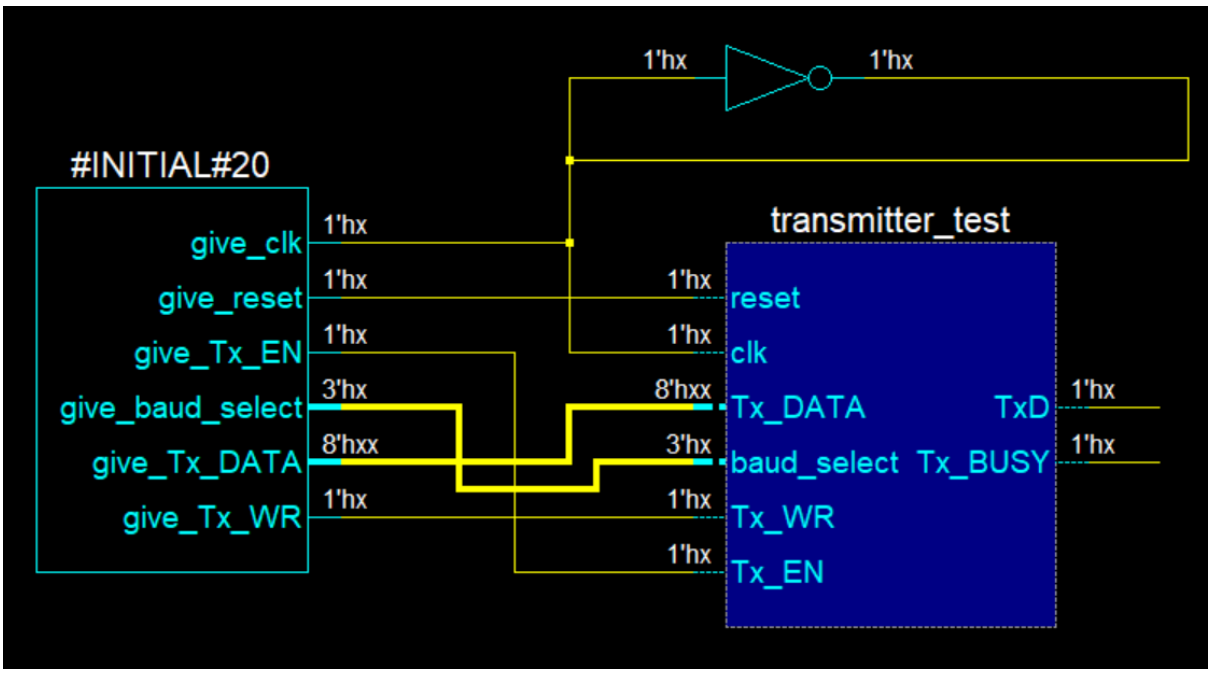
Οι κυματομορφές των σημάτων του αποστολέα από το testbench φαίνονται στο παρακάτω σχήμα:



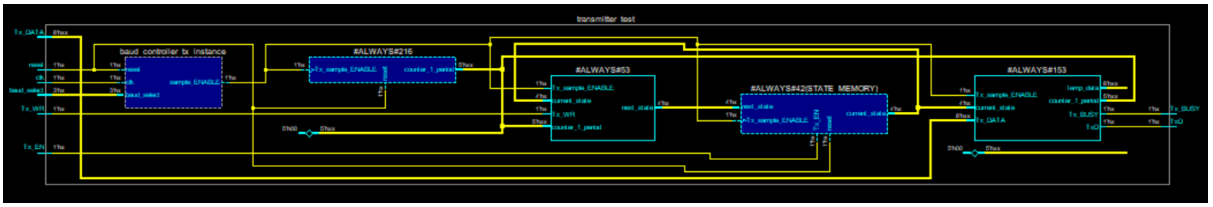
Εικόνα: Κυματομορφές του testbench του αποστολέα

Διακρίνουμε το σήμα της σειριακής εξόδου TxD, ενώ στην κυματομορφή σημειώνεται και η διάρκεια της αποστολής ενός συμβόλου. Διακρίνονται επίσης το start bit στην αρχή της μετάδοσης, καθώς και τα parity και stop bit. Το Tx_BUSY γίνεται 1 όταν ξεκινά αποστολή και 0 όταν γίνεται ξανά διαθέσιμος ο αποστολέας, επιτρέποντας επιτρέποντας νέα μετάδοση δεδομένων.

Παρακάτω, παρατίθεται το σχηματικό του αποστολέα:



Εικόνα: Σχηματικό του testbench του transmitter



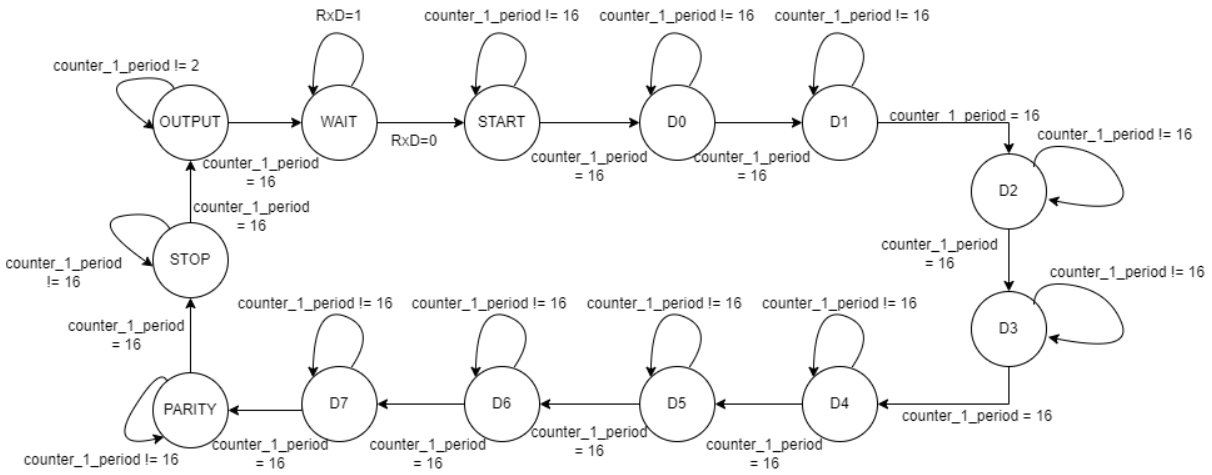
Εικόνα: Σχηματικό του Transmitter

UART Δέκτης

Ο δέκτης δέχεται ως είσοδο το baud_select και το σήμα RxD , το οποίο είναι ίδιο με το μεταδιδόμενο σήμα του αποστολέα (TxD). Επίσης, δέχεται ως είσοδο το σήμα Rx_EN, το οποίο ενεργοποιεί τον receiver. Το Rx_EN θεωρήθηκε πως έχει πάντα τιμή 1 στα αρχεία δοκιμής. Οι έξοδοι του δέκτη είναι οι παρακάτω:

- Rx_DATA: Το ληφθέν σήμα από τον αποστολέα(χωρίς τα start, stop και parity bit) δοσμένα σε παράλληλη έξοδο.
- Rx_FERROR: Σήμα που γίνεται 1 σε περιπτώσεις λανθασμένης σειριακής πλαισίωσης των δεδομένων.
- Rx_PERROR: Σήμα που ισούται με 1 αν υπάρχει σφάλμα στην ισοτιμία.
- Rx_VALID: Γίνεται 1 στην έξοδο αν δεν υπάρχει σφάλμα πλαισίωσης ή σφάλμα ισοτιμίας και δηλώνει ότι το σήμα είναι έγκυρο και μπορεί να προβληθεί στην οθόνη.

Για την υλοποίηση του δέκτη υλοποιήθηκε το παρακάτω FSM:

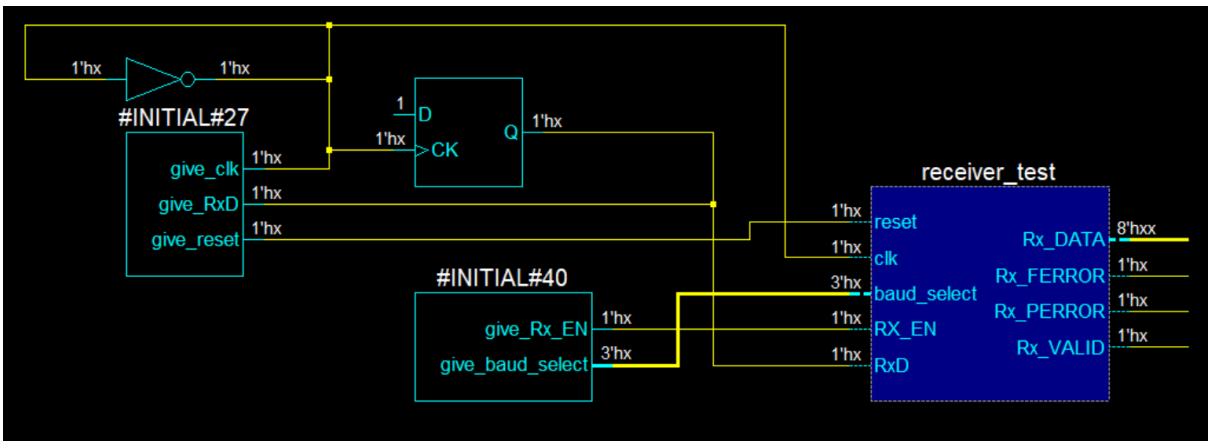


Εικόνα: FSM του δέκτη

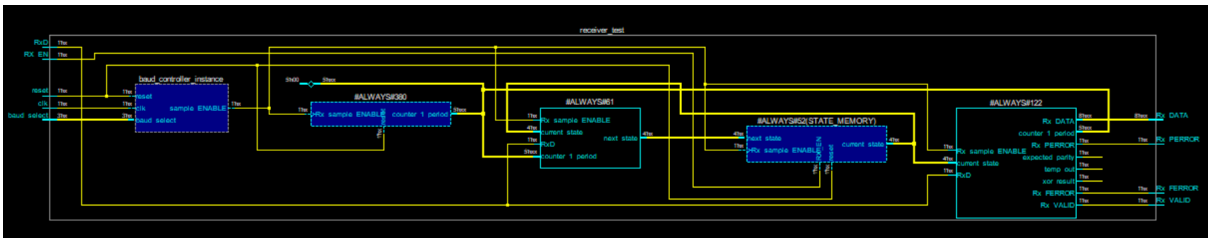
Όπως διακρίνεται από το σχήμα το FSM αποτελείται από 13 καταστάσεις. Αναλυτικότερα:

- WAIT: Ο δέκτης στην κατάσταση WAIT αναμένει το start bit, ενώ τα σήματα σφάλματος γίνονται όλα 0. Επίσης, ο μετρητής μηδενίζεται. Συνεπώς, παραμένει σε αυτή την κατάσταση έως ότου το RxD γίνει 0, οπότε και μεταβαίνει στην κατάσταση start.
- START: Ο δέκτης δειγματοληπτεί το RxD και κάνει έλεγχο για την ύπαρξη σφάλματος πλαισίωσης. Μεταβαίνει στην επόμενη κατάσταση όταν ο μετρητής γίνει ίσος με 16.
- D0-D7: Σε αυτές τις καταστάσεις ο δέκτης παραμένει μέχρι ο μετρητής να γίνει 16. Στην διάρκεια κάθε κατάστασης, ο δέκτης δειγματοληπτεί το σήμα και εξετάζει την ύπαρξη σφάλματος ισοτιμίας. Η έξοδος εμφανίζεται στο Rx_DATA[i], i = 0, ..., 7,(αν δεν έχει βρεθεί σφάλμα πλαισίωσης ως τότε) κάθε φορά που ο μετρητής έχει την τιμή 8, δηλαδή, η έξοδος εμφανίζεται λίγο μετά το μέσο της χρονικής διάρκειας $\Delta t = 16\mathrm{T}'$ από την στιγμή που μηδενίζει ο counter.
- PARITY: Ο δέκτης αναμένει το stop bit και παραμένει σε αυτή την κατάσταση έως ότου ο περιστροφικός μετρητής να γίνει 16. Ελέγχει για την ύπαρξη λάθους ισοτιμίας (Rx_FERROR) ή πλαισίωσης(λόγω ύπαρξης παρεμβολών που οδηγούν σε διακυμάνσεις, από 0 σε 1 και από 1 σε 0, του stop bit) (Rx_FERROR).
- OUTPUT: Ο δέκτης παραμένει σε αυτή την κατάσταση εως ότου ο μετρητής γίνει 4, οπότε και μεταβαίνει στην κατάσταση WAIT. Στην διάρκεια του OUTPUT, δίνεται στην έξοδο σήμα εγκυρότητας Rx_VALID = 1. Διαφορετικά, το σήμα αυτό παραμένει στο 0 και τουλάχιστον ένα από τα σήματα λάθους είναι στο 1.

Το σχηματικό του δέκτη φαίνεται παρακάτω:

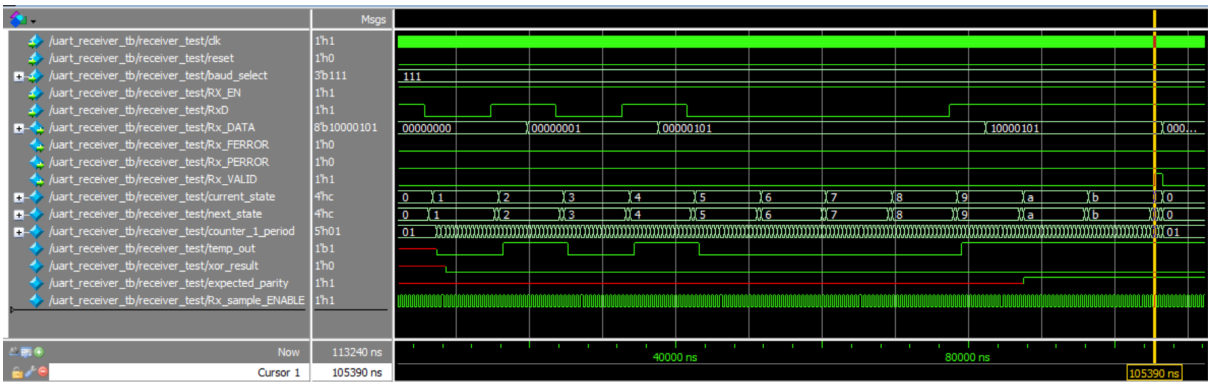


Εικόνα: Σχηματικό του testbench του receiver



Εικόνα: Σχηματικό του receiver

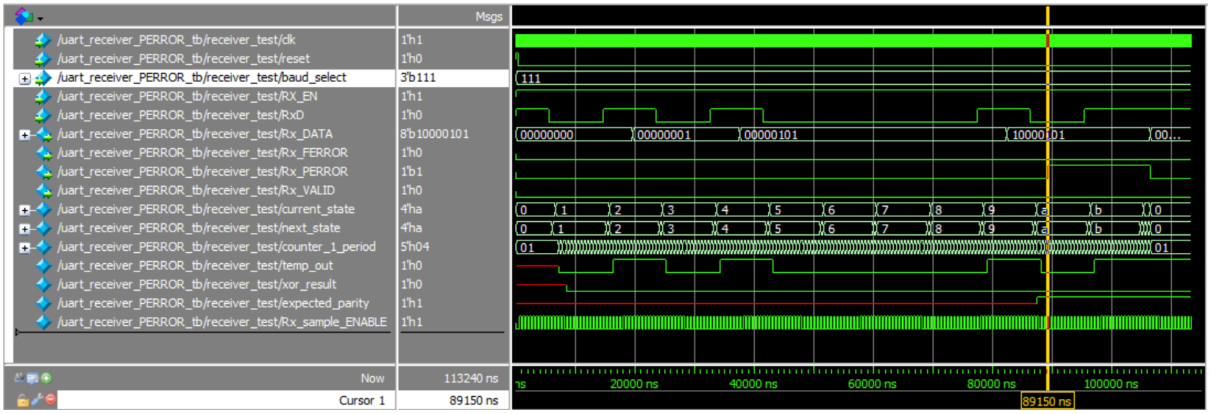
Οι κυματομορφές του testbench φαίνονται παρακάτω:



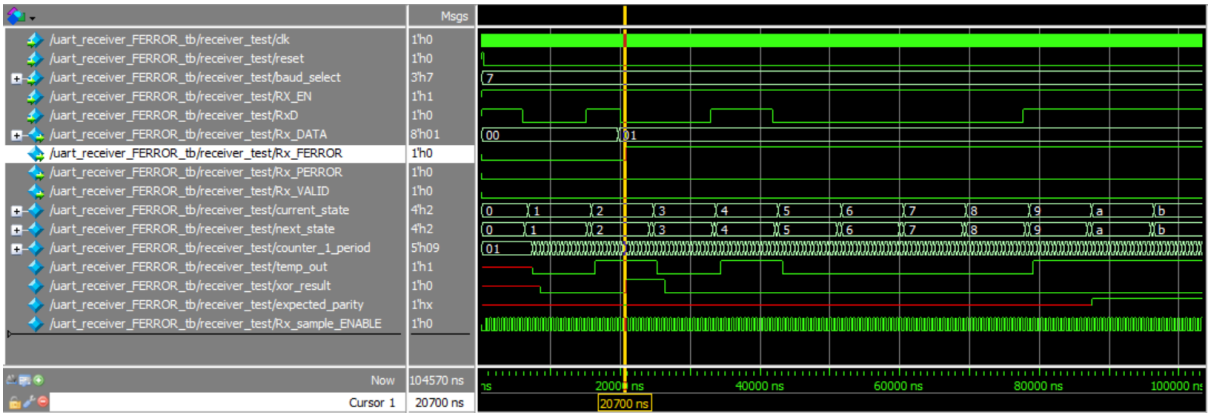
Εικόνα: Κυματομορφές του του testbench του δέκτη

Παρατηρείται ότι η έξοδος του δέκτη είναι η αναμενόμενη την χρονική στιγμή που το εμφανίζεται η θετική ακμή του Rx_VALID.

Παρακάτω εμφανίζονται οι κυματομορφές για λανθασμένο bit ισοτιμίας και για σφάλμα πλαισίωσης.



Εικόνα: Λειτουργία δέκτη σε σφάλμα ισοτιμίας

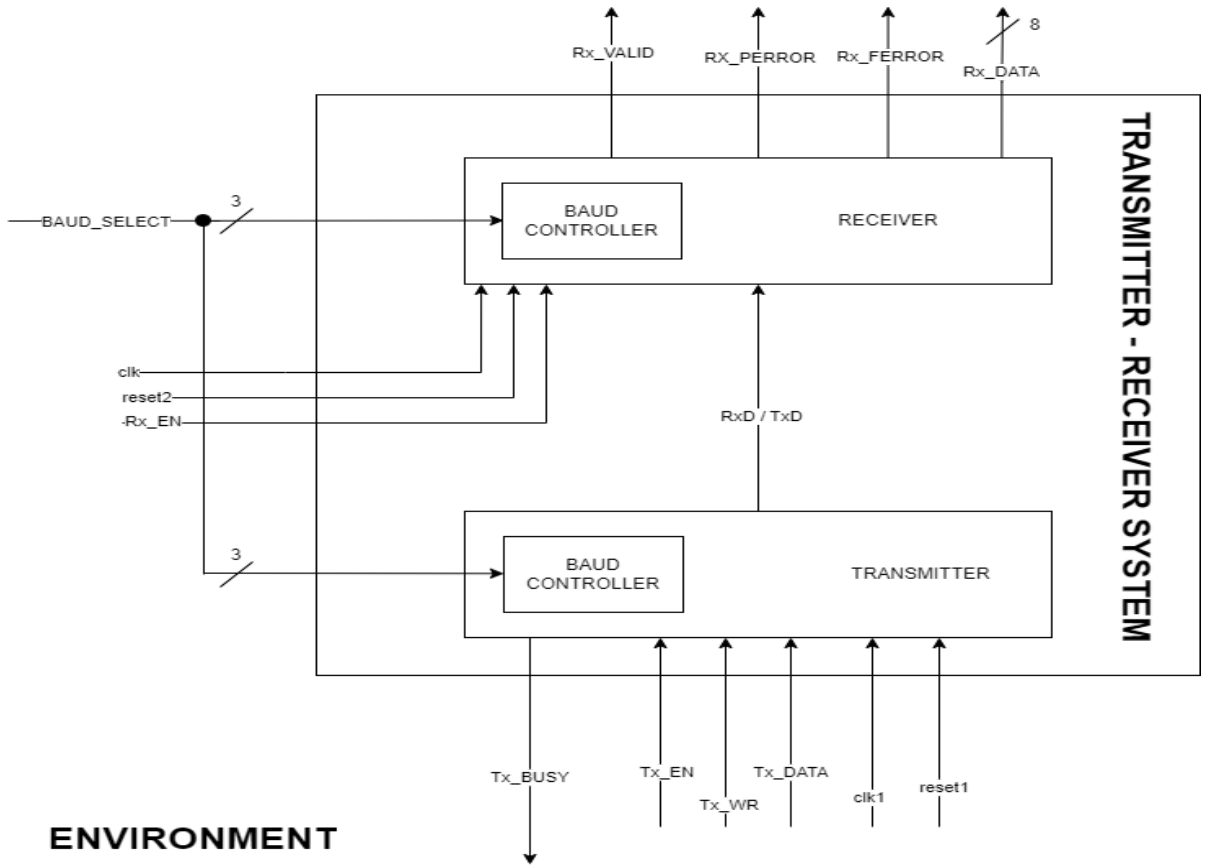


Εικόνα: Λειτουργία δέκτη σε σφάλμα πλαισίωσης

Και στις δύο περιπτώσεις που εμφανίζεται σφάλμα, η τιμή του Rx_VALID παραμένει στο 0.

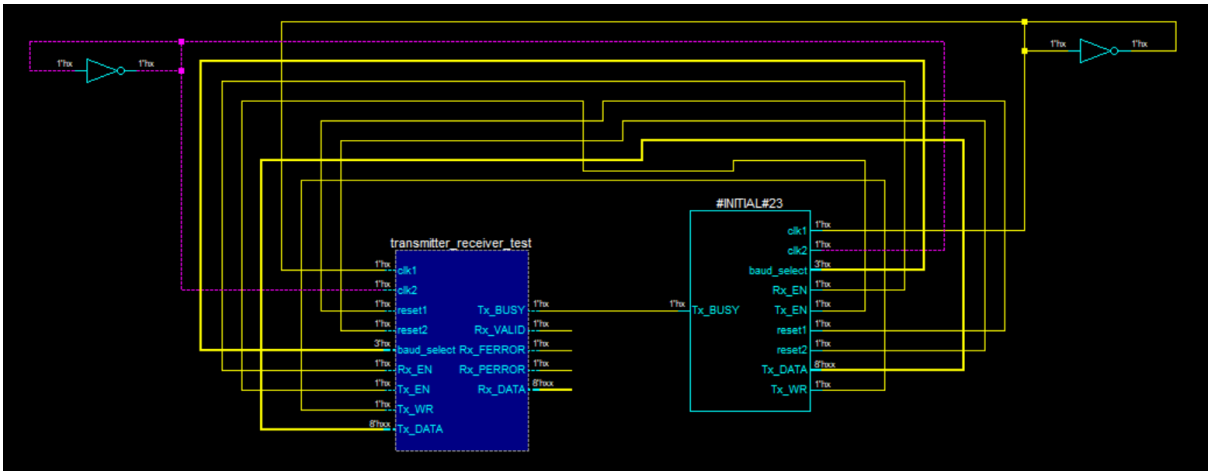
UART Αποστολέας-Δέκτης

Το κύκλωμα UART Αποστολέα-Δέκτη δέχεται ως εισόδους τα σήματα:
Tx_EN, Tx_WR, Tx_DATA, baud_select, Rx_EN
και έχει εξόδους τα σήματα:
Rx_VALID, RX_PERROR, Rx_FERROR ,Rx_DATA και Rx_BUSY.

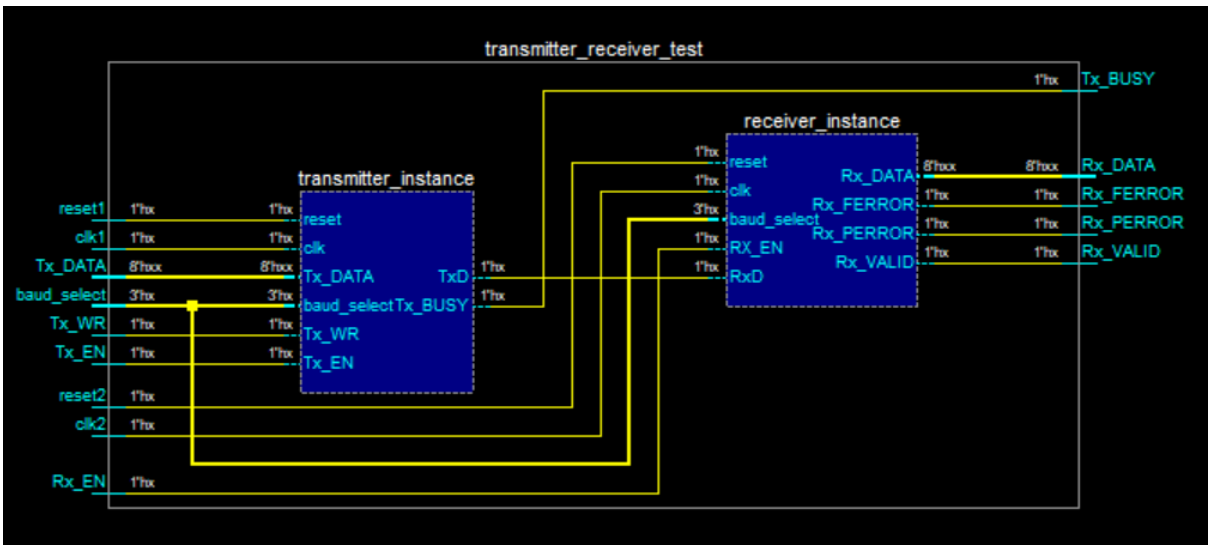


Εικόνα : Πρόχειρο σχηματικό διάγραμμα του κυκλώματος Αποστολέα-Δέκτη.

Το σχηματικό από το εργαλείο προσομοίωσης είναι το εξής:

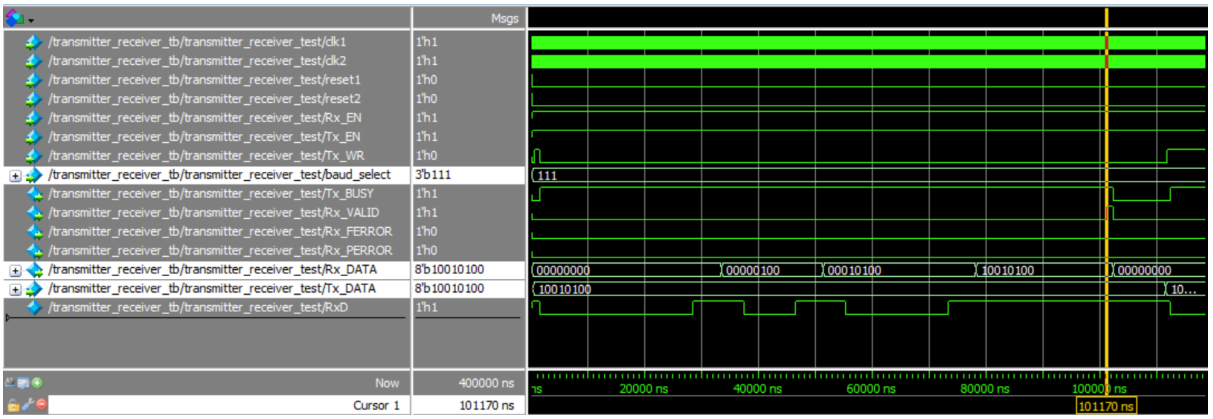


Εικόνα: Σχηματικό του testbench του κυκλώματος transmitter_receiver

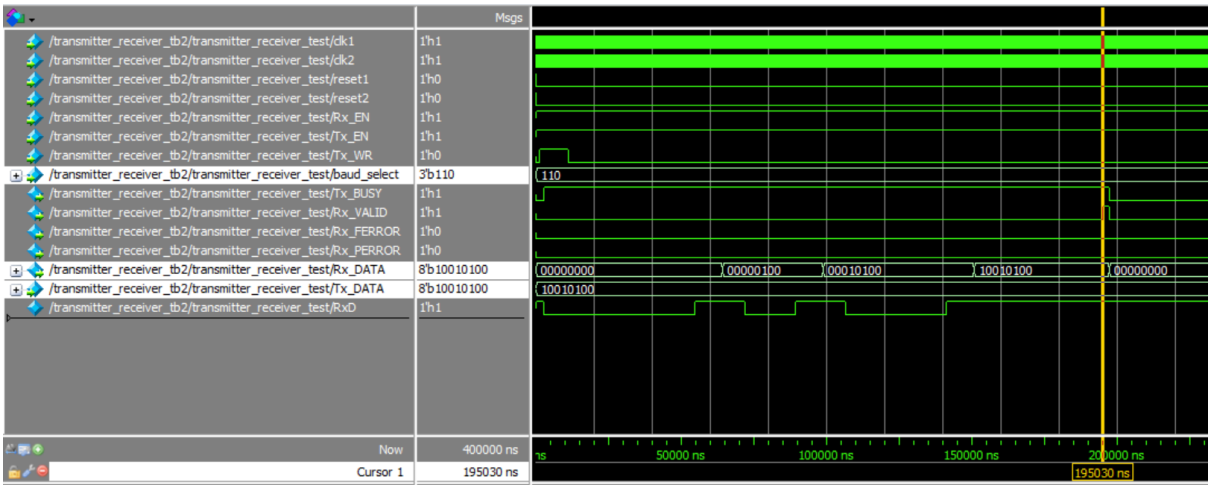


Εικόνα: Σχηματικό του κυκλώματος transmitter_receiver

Παρακάτω παρουσιάζονται οι κυματομορφές για baud_select 111 και baud_select 110.



Εικόνα: Κυματομορφές του συστήματος αποστολέα - δέκτη για baud_select 111



Εικόνα: Κυματομορφές του συστήματος αποστολέα - δέκτη για baud_select 110

Παρατηρείται ότι η λειτουργία είναι η αναμενόμενη και η έξοδος του δέκτη είναι ίδια με την είσοδο του αποστολέα στην θετική ακμή του Rx_VALID.

Μέρος Γ – Σύστημα Δέκτη με προβολή του μηνύματος στην οθόνη τεσσάρων LED 7-τμημάτων

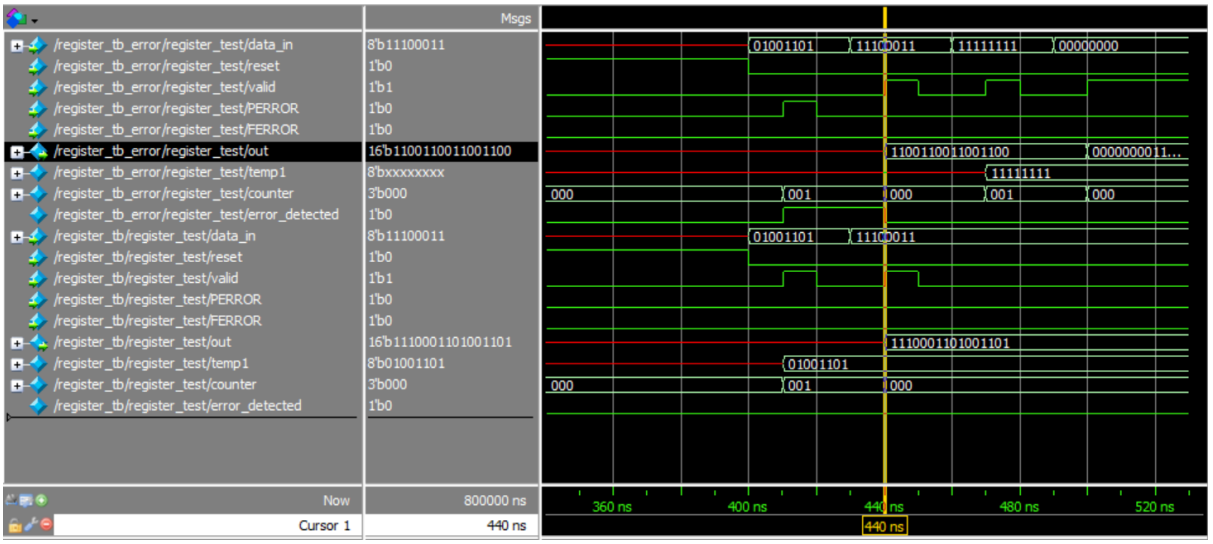
Υλοποίηση Register με ειδικές λειτουργίες

Για την οδήγηση της εξόδου του παραλήπτη στον οδηγητή των οθονών έγινε χρήση ενός ειδικού καταχωρητή (βρίσκεται στο αρχείο με ονομασία *register.v*), ο οποίος:

- Δέχεται ως είσοδο τα σήματα Rx_VALID, Rx_PERROR, Rx_FERROR, Rx_DATA
- Έξοδος είναι το σήμα signal_to_display, το οποίο οδηγείται στην είσοδο του LED decoder.

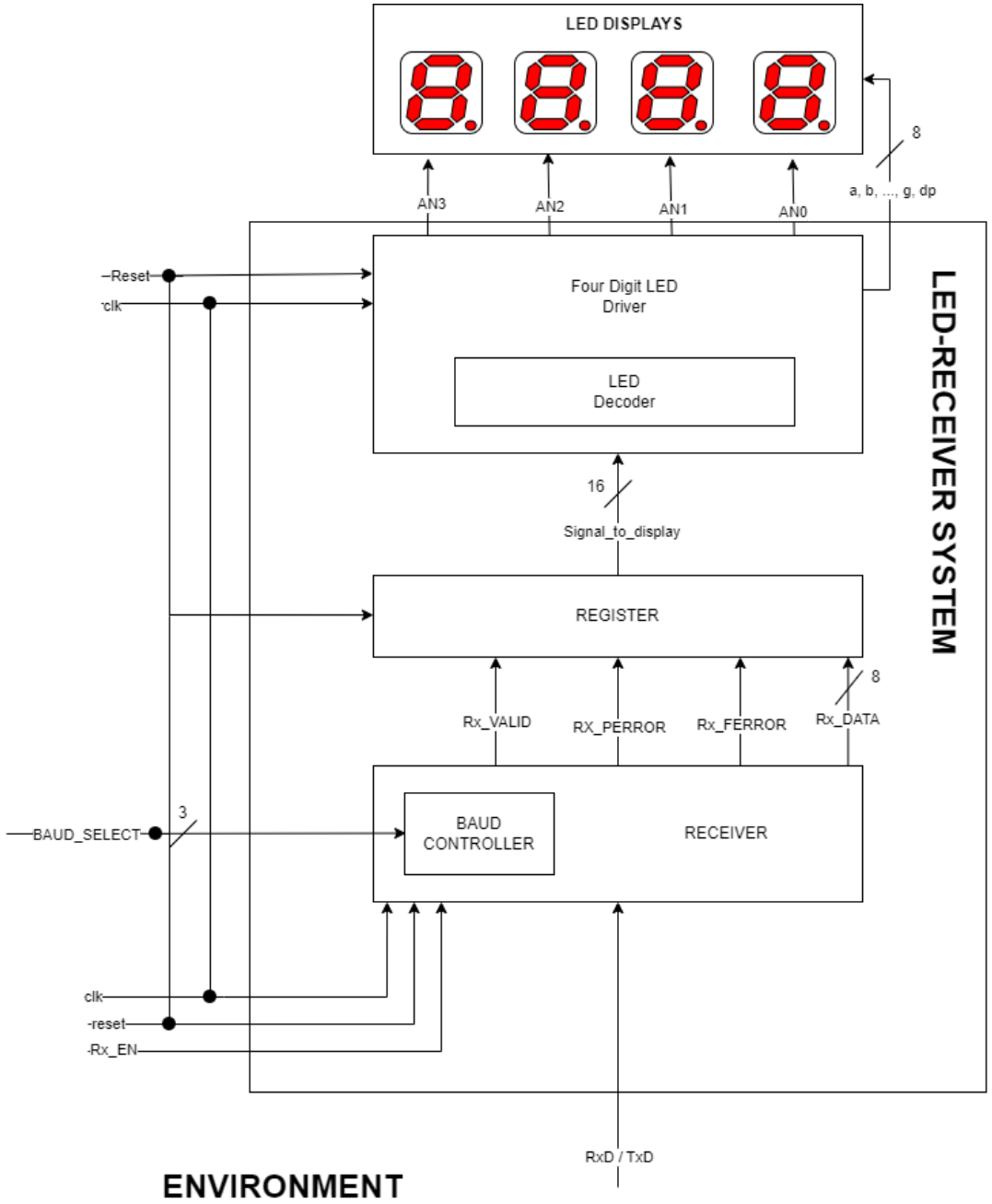
Η λειτουργία του έχει ως εξής: Τα δεδομένα του δέκτη αποθηκεύονται προσωρινά και εμφανίζονται μόνο όταν δεχτεί συνολικά 2 πακέτα δεδομένων των 8 bit. Επίσης, πρέπει να αναφερθεί ότι τα δεδομένα εμφανίζονται στην έξοδο μόνο αν δεν υπάρξει κάποιο σφάλμα στην μετάδοση των δύο πακέτων. Αν υπάρξει έστω και ένα σφάλμα, στις LED οθόνες εμφανίζεται το μήνυμα λάθους 'FFFF'.

Οι κυματομορφές ορθής λειτουργίας και λειτουργίας με σφάλμα είναι οι παρακάτω:



Εικόνα: Κυματομορφές του testbench του ειδικού καταχωρητή

Συνένωση Δέκτη-Οθόνης

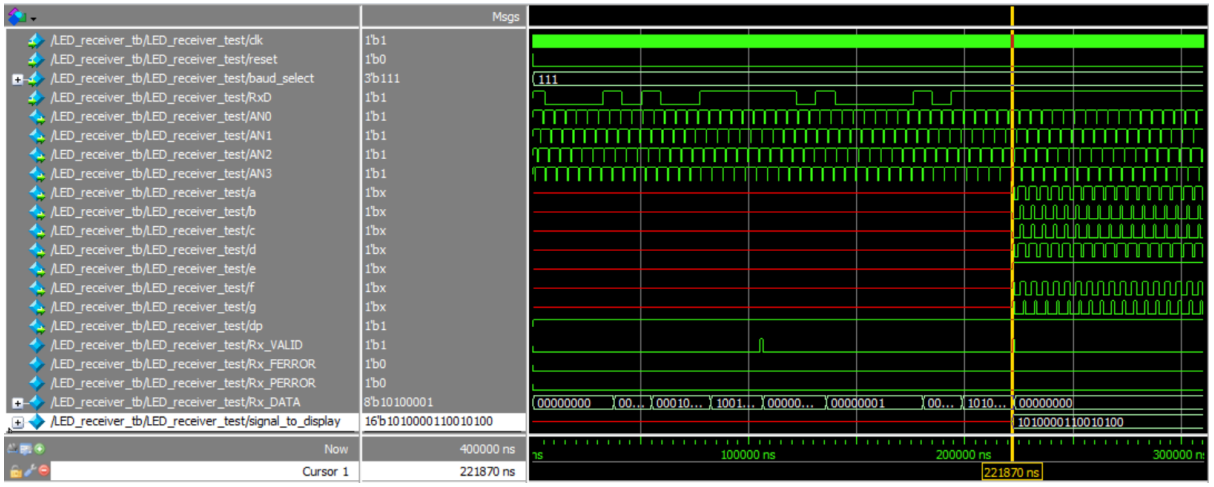


Εικόνα: Πρόχειρο Σχηματικό του συστήματος δέκτη - οδηγητή LED

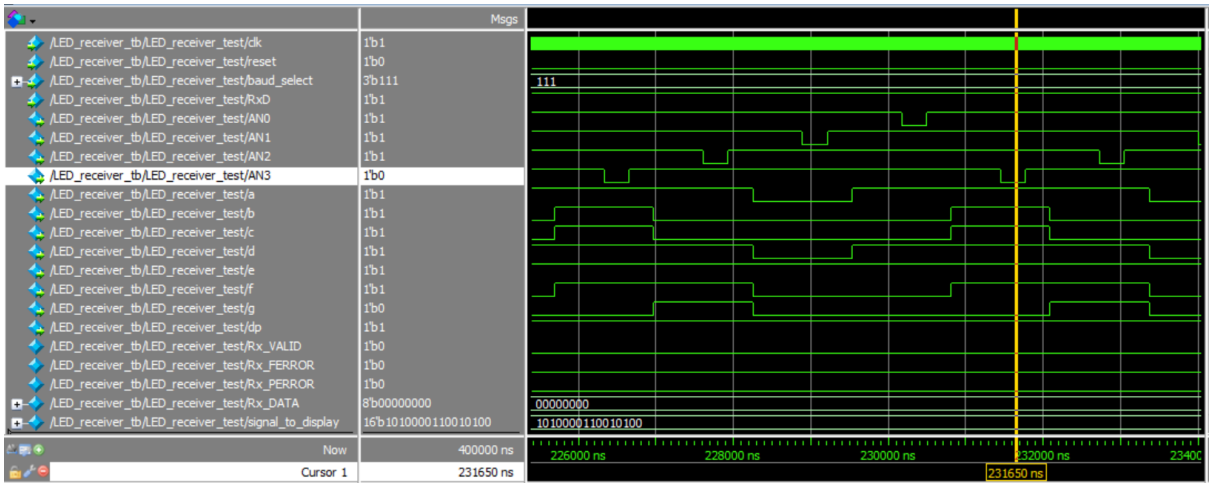
Εικόνα: πρόχειρο σχηματικό διάγραμμα του κυκλώματος Δέκτης-Οθόνης

Το κύκλωμα Δέκτη-Οθόνης δέχεται ως εισόδους τα σήματα:
Rx_EN, RxD/TxD, baud_select, Rx_EN
και έχει εξόδους τα σήματα:
AN3, AN2, AN1, AN0, a, b, c, d, e, f, g, dp.

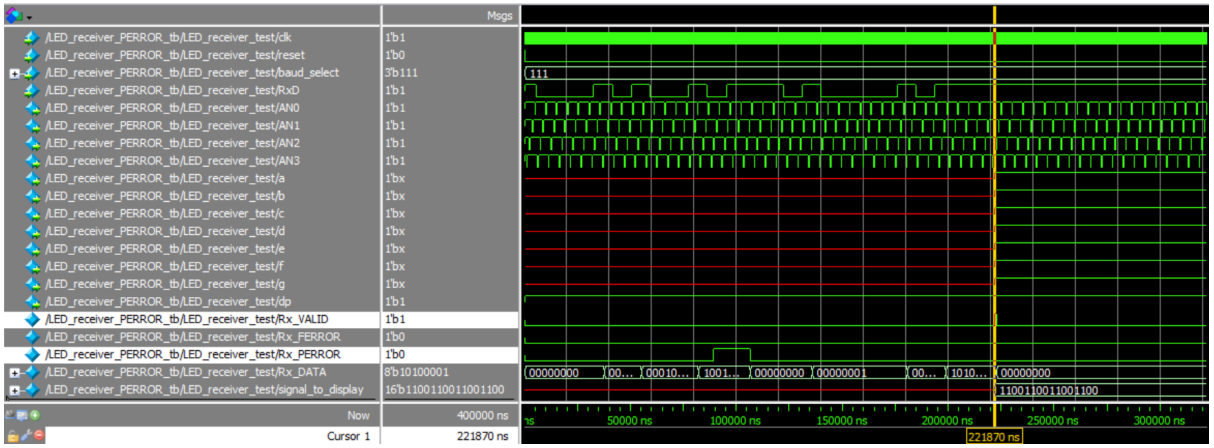
Οι κυματομορφές του testbench είναι οι παρακάτω. ΠΑρατίθενται και οι περιπτώσεις εμφάνισης σφάλματος:



Εικόνα: Κυματομορφές συστήματος δέκτη-LED driver

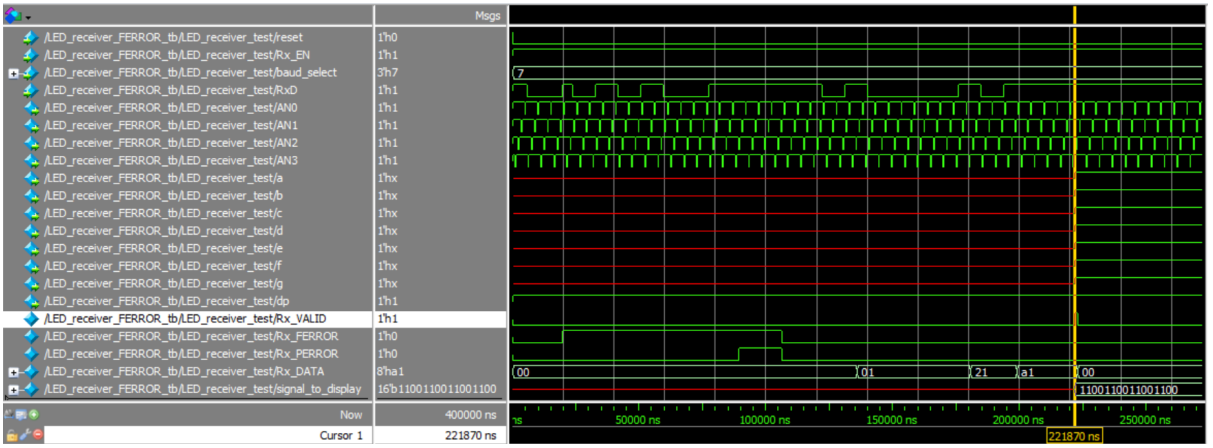


Εικόνα: Μεγέθυνση των προηγούμενων κυματομορφών



Εικόνα:Κυματομορφές συστήματος δέκτη-LED driver με PERROR

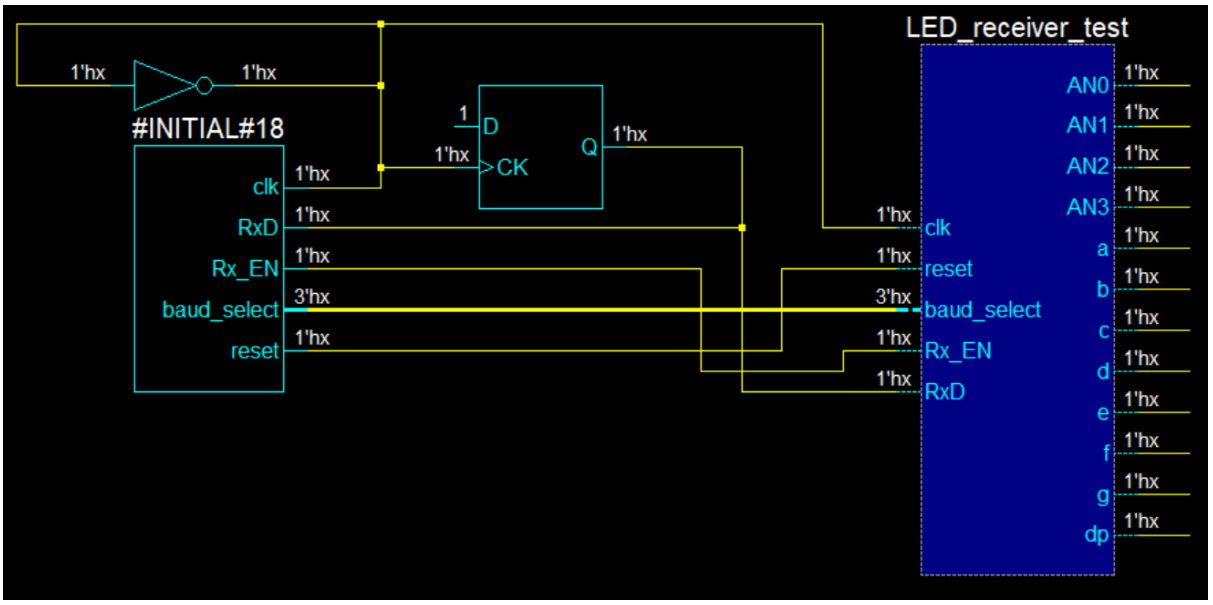
Βλέπουμε ότι έχουμε PERROR και η έξοδος εμφανίζεται ‘FFFF’



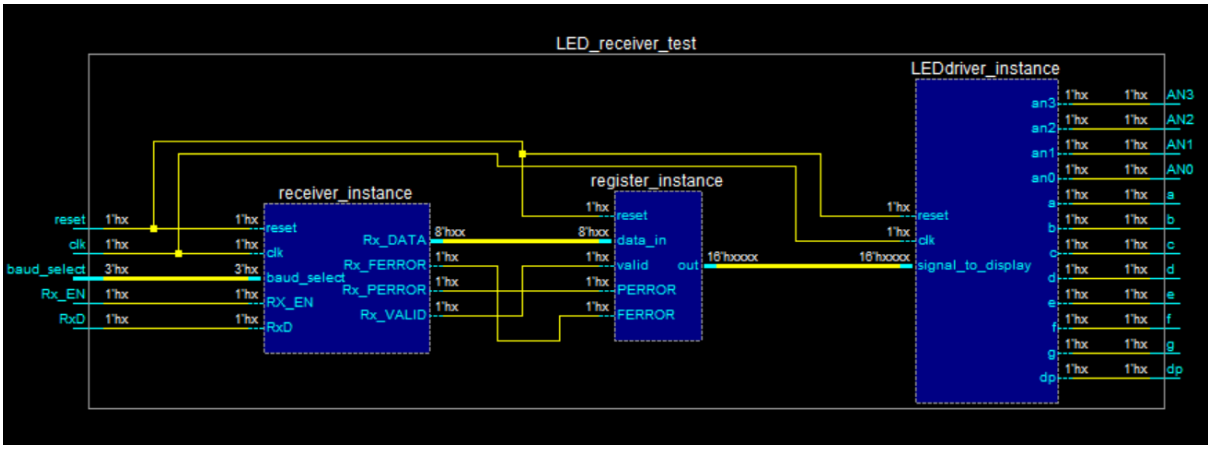
Εικόνα:Κυματομορφές συστήματος δέκτη-LED driver με FERROR

Και για δοκιμή του κυκλώματος με Rx_FERROR βγαίνει ‘FFFF’

Το σχηματικό φαίνεται παρακάτω:



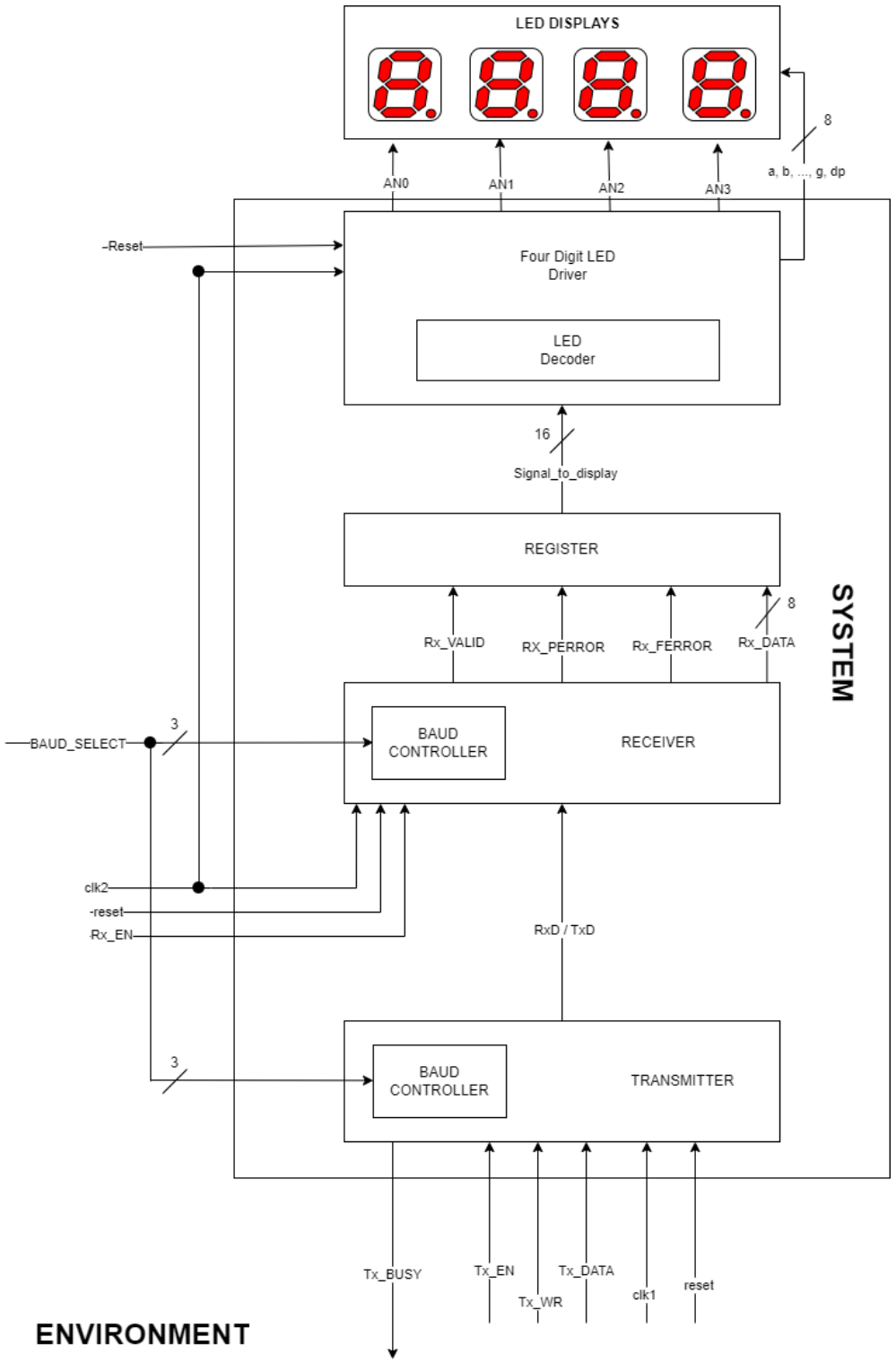
Εικόνα: Σχηματικό του testbench του κυκλώματος LED_receiver



Εικόνα: Σχηματικό του κυκλώματος LED_receiver

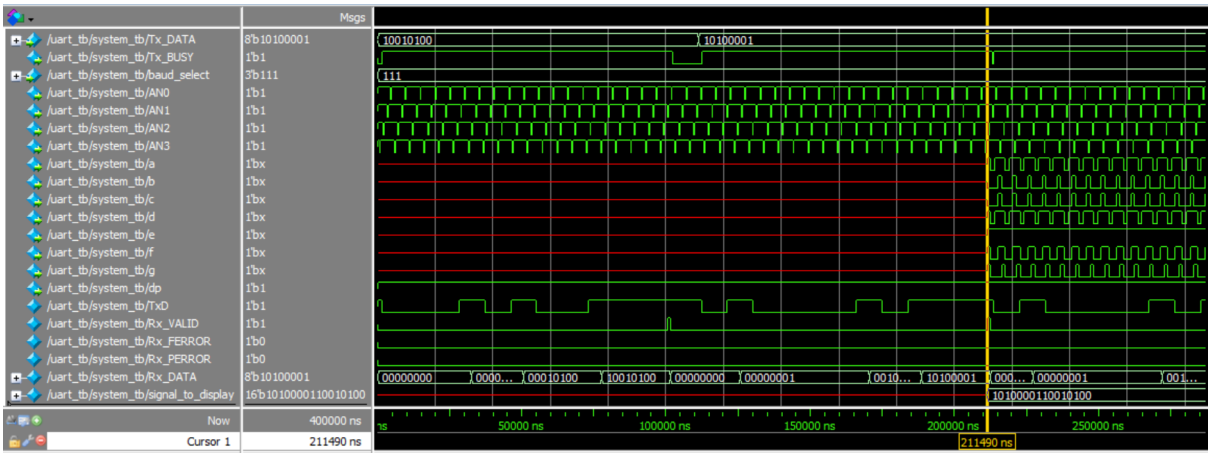
Συνένωση όλων των επιμέρους τμημάτων για την αποστολή - παραλαβή και προβολή μηνύματος

Το πλήρες πρόχειρο διάγραμμα όλου του κυκλώματος φαίνεται παρακάτω:

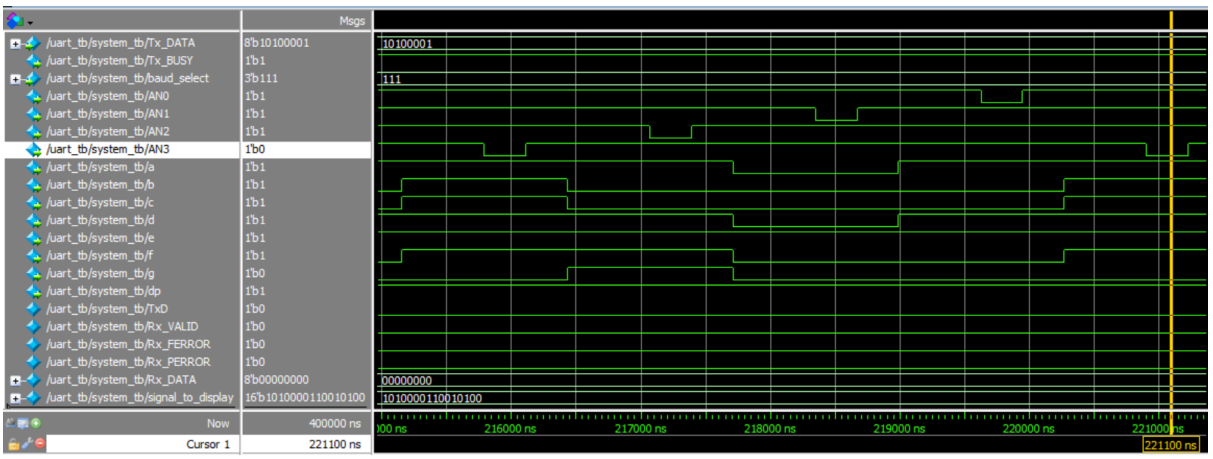


Εικόνα: Πρόχειρο σχηματικό όλου του συστήματος μετάδοσης-παραλαβής και προβολής μηνύματος

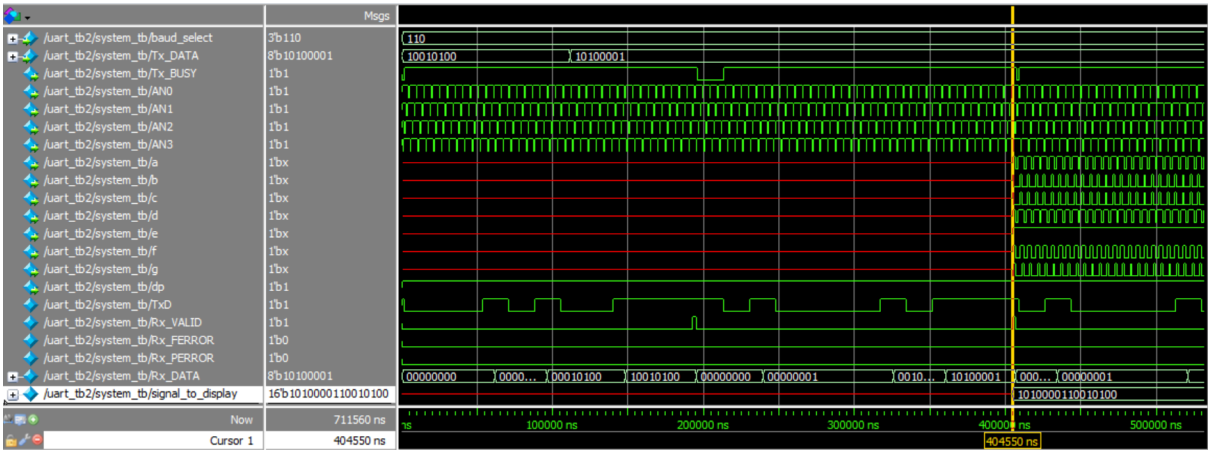
Οι κυματομορφές, όπως αυτές εξάγονται από το εργαλείο προσομοίωσης για διαφορετικά Baud Rate είναι:



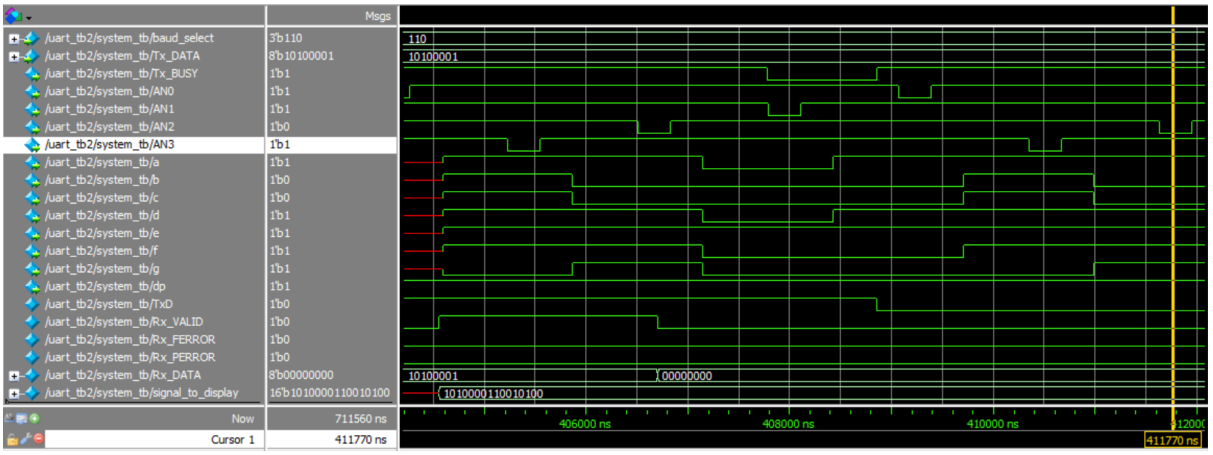
Εικόνα: Το αποτέλεσμα του testbench για όλο το σύστημα (BR = 111)



Εικόνα: Μεγέθυνση της προηγούμενης εικόνας

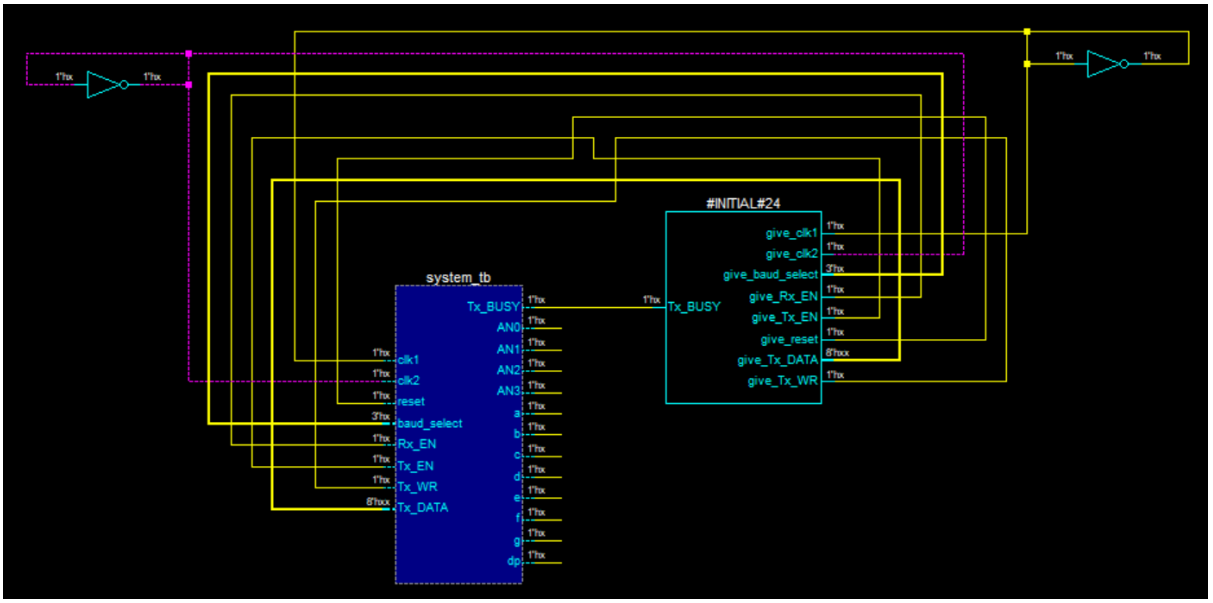


Εικόνα: Το αποτέλεσμα του testbench για όλο το σύστημα (BR = 110)

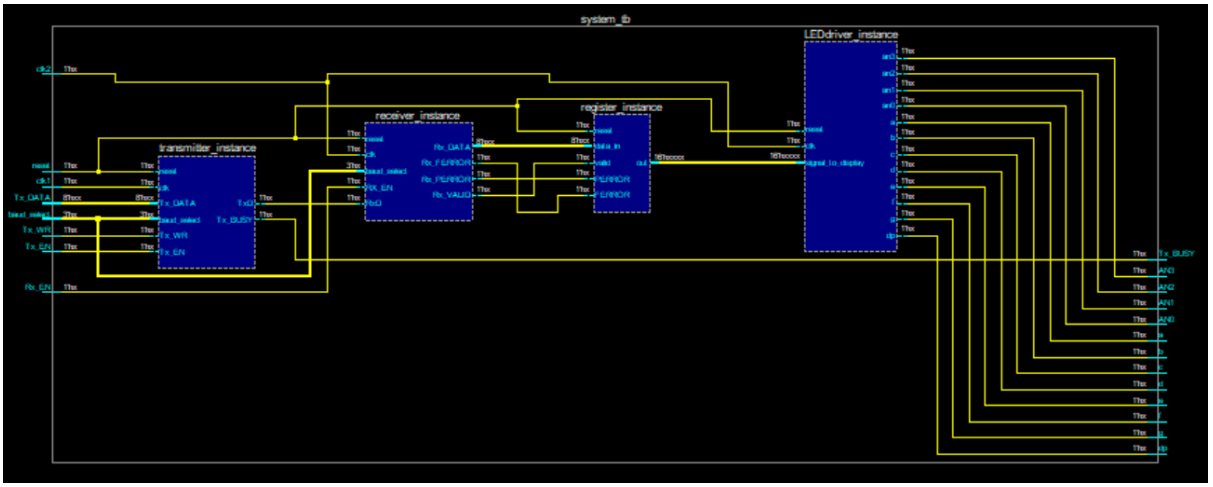


Εικόνα: Μεγέθυνση της προηγούμενης εικόνας

Σχηματικό



Εικόνα: Σχηματικό του testbench του κυκλώματος όλων των επιμέρους τμημάτων της υλοποίησης χωρίς κωδικοποίηση



Εικόνα: Σχηματικό του κυκλώματος όλων των επιμέρους τμημάτων της υλοποίησης χωρίς κωδικοποίηση

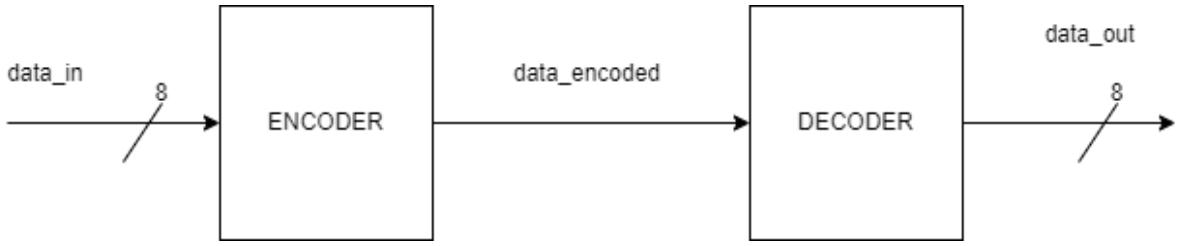
Μέρος Δ – Υλοποίηση Κωδικοποιητή-Αποκωδικοποιητή μεμονωμενη λειτουργία κωδικοποιητή - αποκωδικοποιητή

Για το μέρος αυτό υλοποιήθηκε ένας απλός κωδικοποιητής και ένας απλός αποκωδικοποιητής 8 σε 8 ώστε οι κωδικοποιημένοι χαρακτήρες να εμφανίζονται αυξημένοι κατά 1.
Πιο αναλυτικά, στον παρακάτω πίνακα, εμφανίζονται οι κωδικοποιημένοι χαρακτήρες:

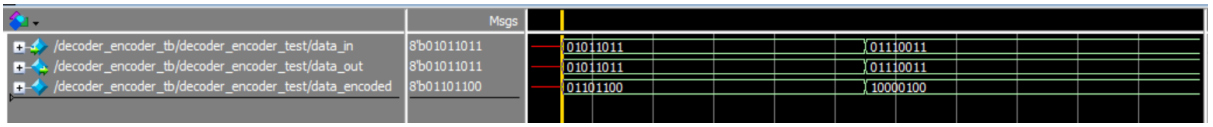
Χαρακτήρας	**Κωδικοποίηση**	**Έξοδος του encoder**
0	00 00	00 01
1	00 01	00 10
2	00 10	00 11
3	00 11	01 00
4	01 00	01 01
5	01 01	01 10
6	01 10	01 11
7	01 11	10 00
8	10 00	10 01
9	10 01	10 10
-	10 10	10 11
F	10 11	11 00
[κενό]	11 00	11 01

Πίνακας: οι έξοδοι του LED Decoder και η αντίστοιχη κωδικοποίηση των χαρακτήρων

Για τον έλεγχο λειτουργίας της κωδικοποίησης υλοποιήθηκε το παρακάτω κύκλωμα, του οποίου οι έξοδοι από το testbench εμφανίζονται παρακάτω:

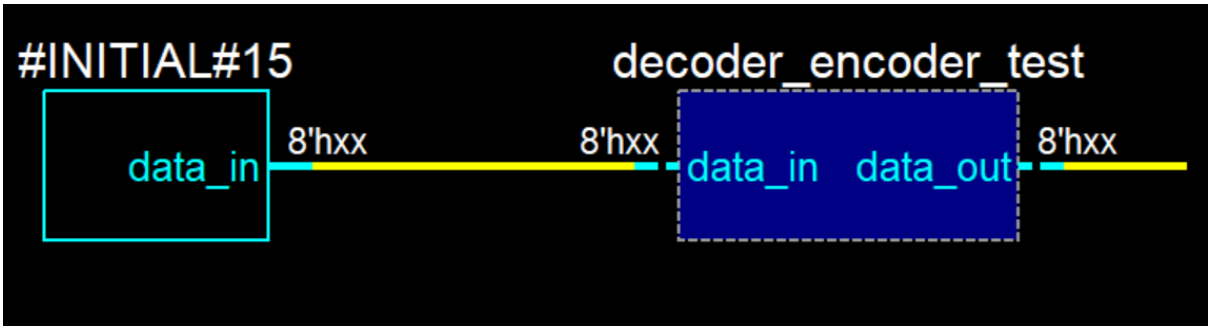


Εικόνα: Πρόχειρο σχηματικό του συστήματος encoder-decoder

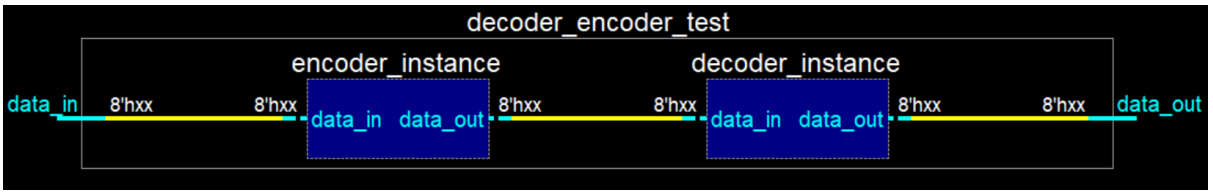


Εικόνα: Κυματομορφές του συστήματος encoder-decoder

Φαίνεται πως οι έξοδοι είναι ίδιες με τις εισόδους μετά την αποκωδικοποίηση.

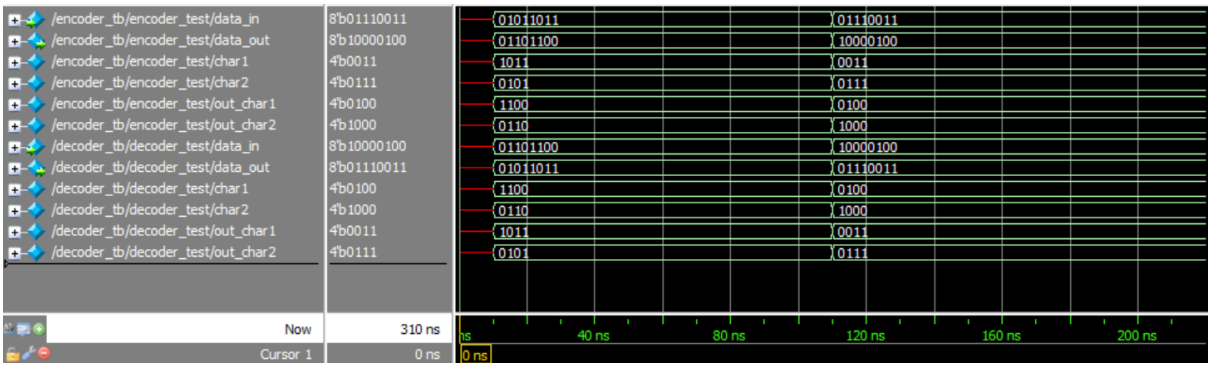


Εικόνα: Σχηματικό του testbench του ειδικού καταχωρητή



Εικόνα: Σχηματικό του ειδικού καταχωρητή

Για πληρότητα της ανάλυσης, παρατίθενται και τα αποτελέσματα των testbench για τα μεμονωμένα module των encoder και decoder:

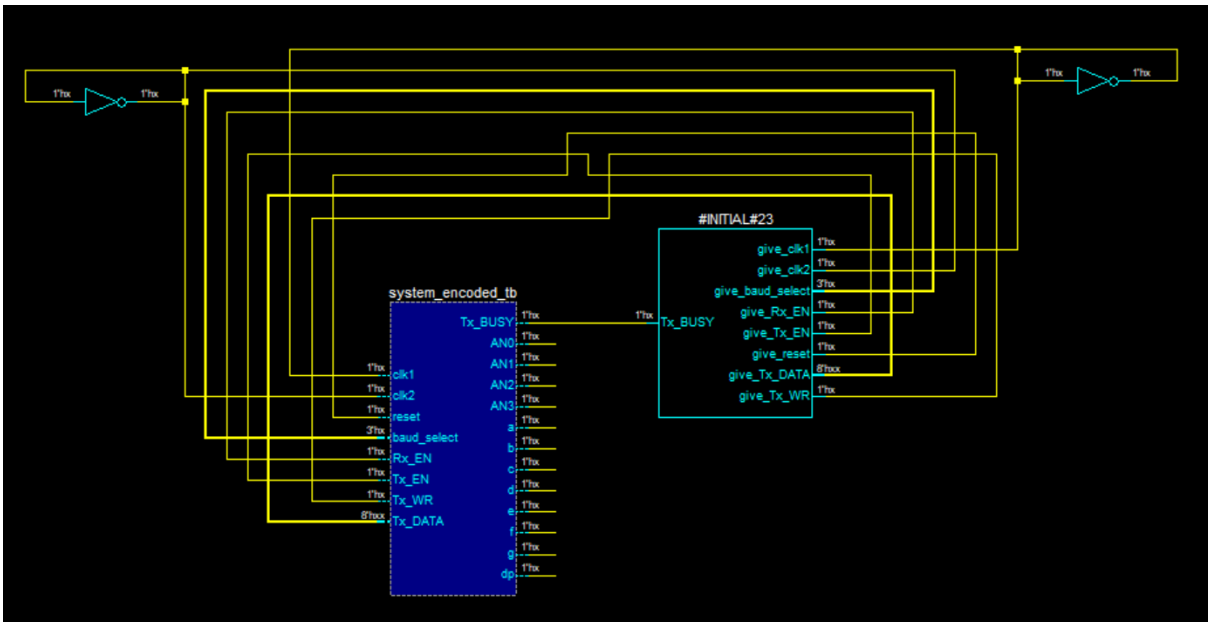


Εικόνα: αποτελέσματα των testbench για τα μεμονωμένα module των encoder και decoder

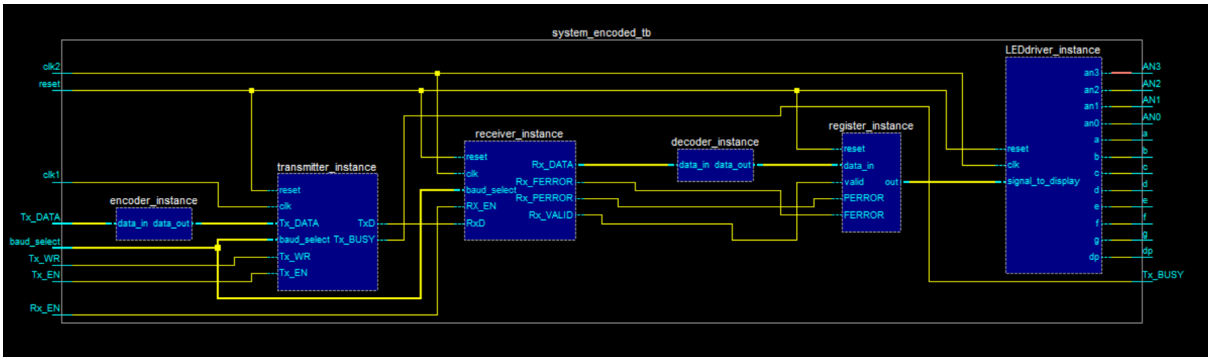
Συνένωση με παρουσία Κωδικοποιητή και αποκωδικοποιητή

Το πρόχειρο διάγραμμα με την εισαγωγή του κωδικοποιητή και αποκωδικοποιητή εμφανίζεται ως εξής:

Το σχηματικό του εργαλείου προσομοίωσης είναι το εξής:

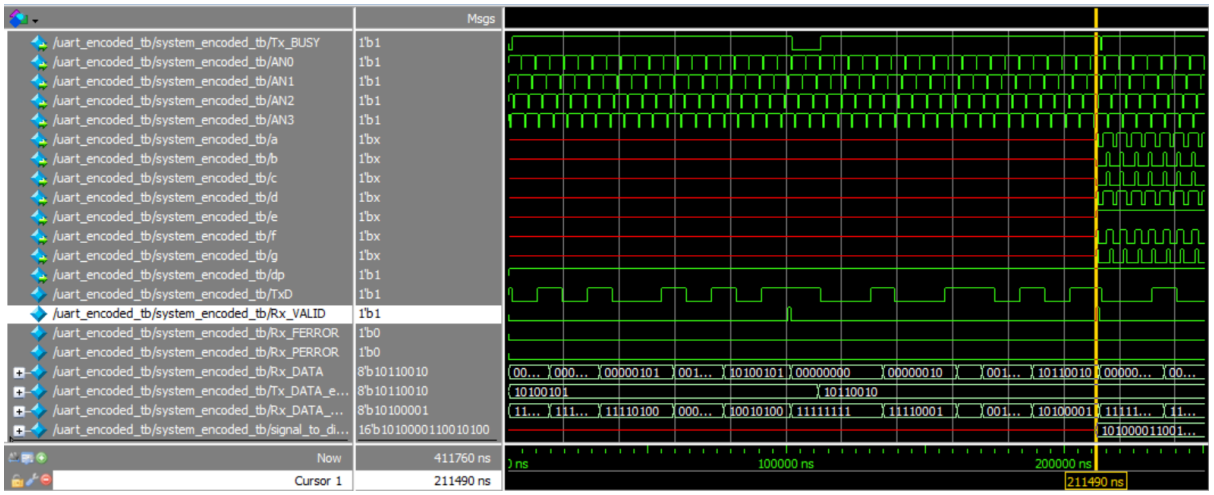


Εικόνα: Σχηματικό του testbench του κυκλώματος όλων των επιμέρους τμημάτων της υλοποίησης με κωδικοποίηση

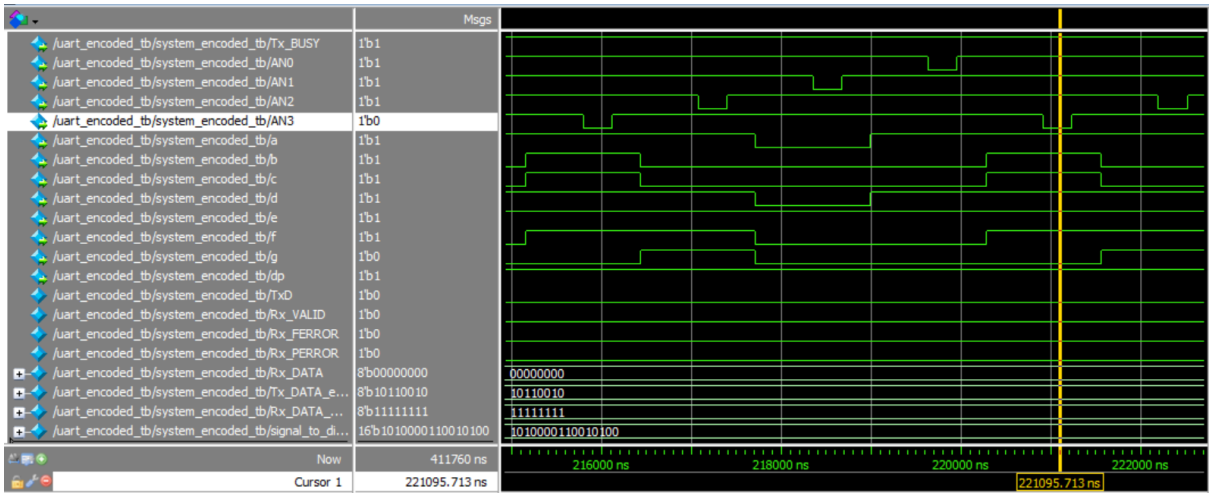


Εικόνα: Σχηματικό του κυκλώματος όλων των επιμέρους τμημάτων της υλοποίησης με κωδικοποίηση

Οι κυματομορφές είναι:

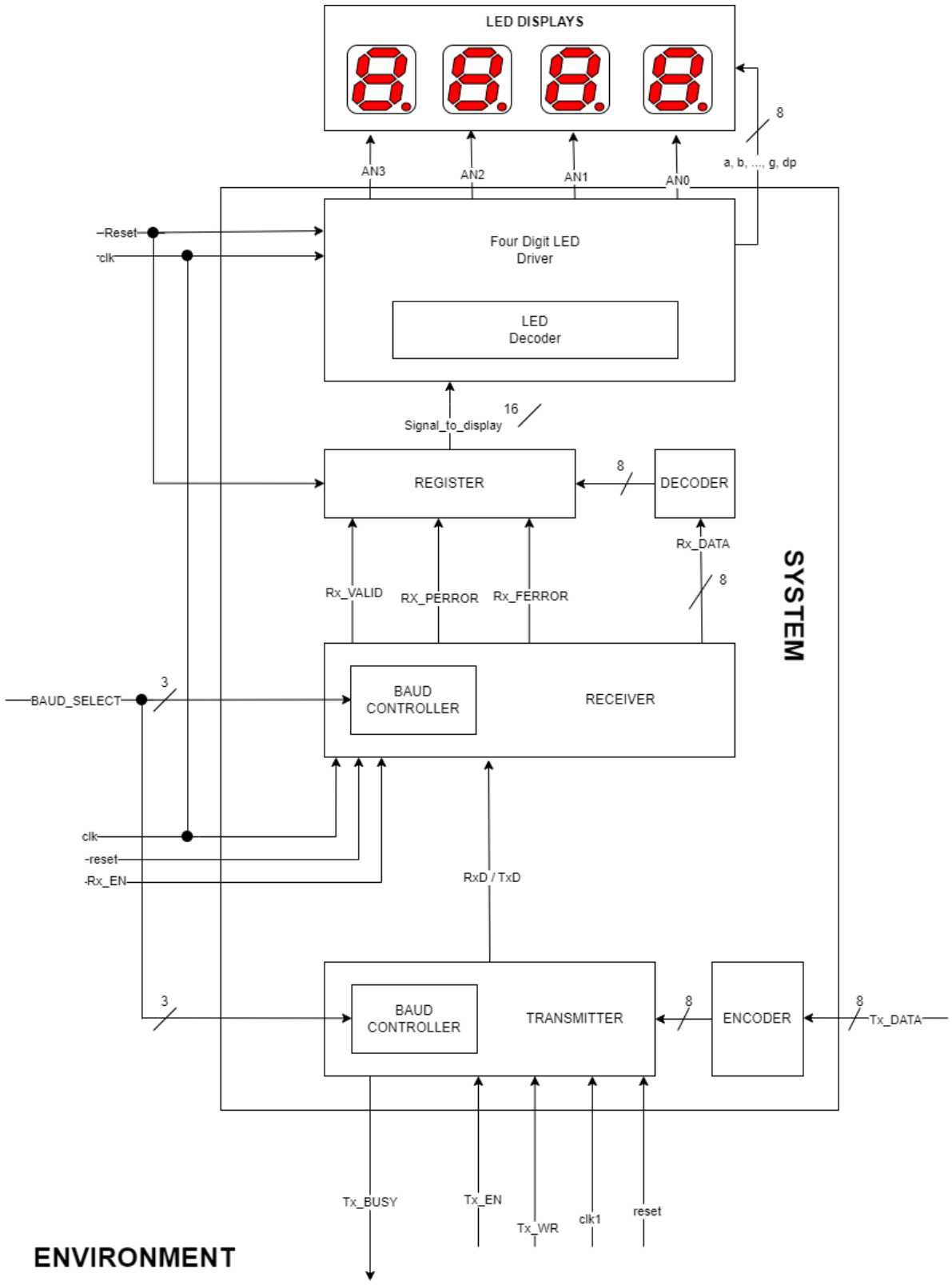


Εικόνα: Κυματομορφές όλου του συστήματος μετάδοσης-παραλαβής και προβολής μηνύματος



Εικόνα: Μεγέθυνση της προηγούμενης εικόνας

Τέλος, παρατίθεται το πρόχειρο σχηματικό που υλοποιήθηκε, για πληρότητα:



Εικόνα: Πρόχειρο σχηματικό όλου του συστήματος μετάδοσης-παραλαβής και προβολής μηνύματος με κωδικοποίηση