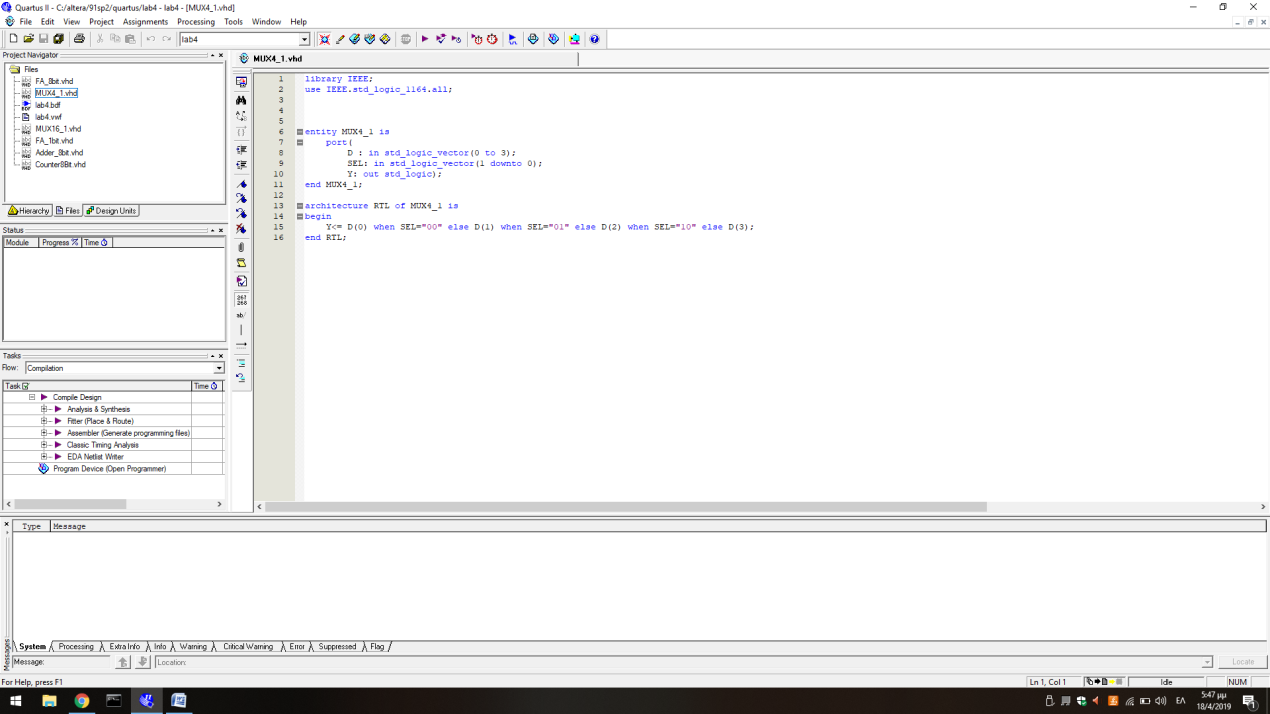
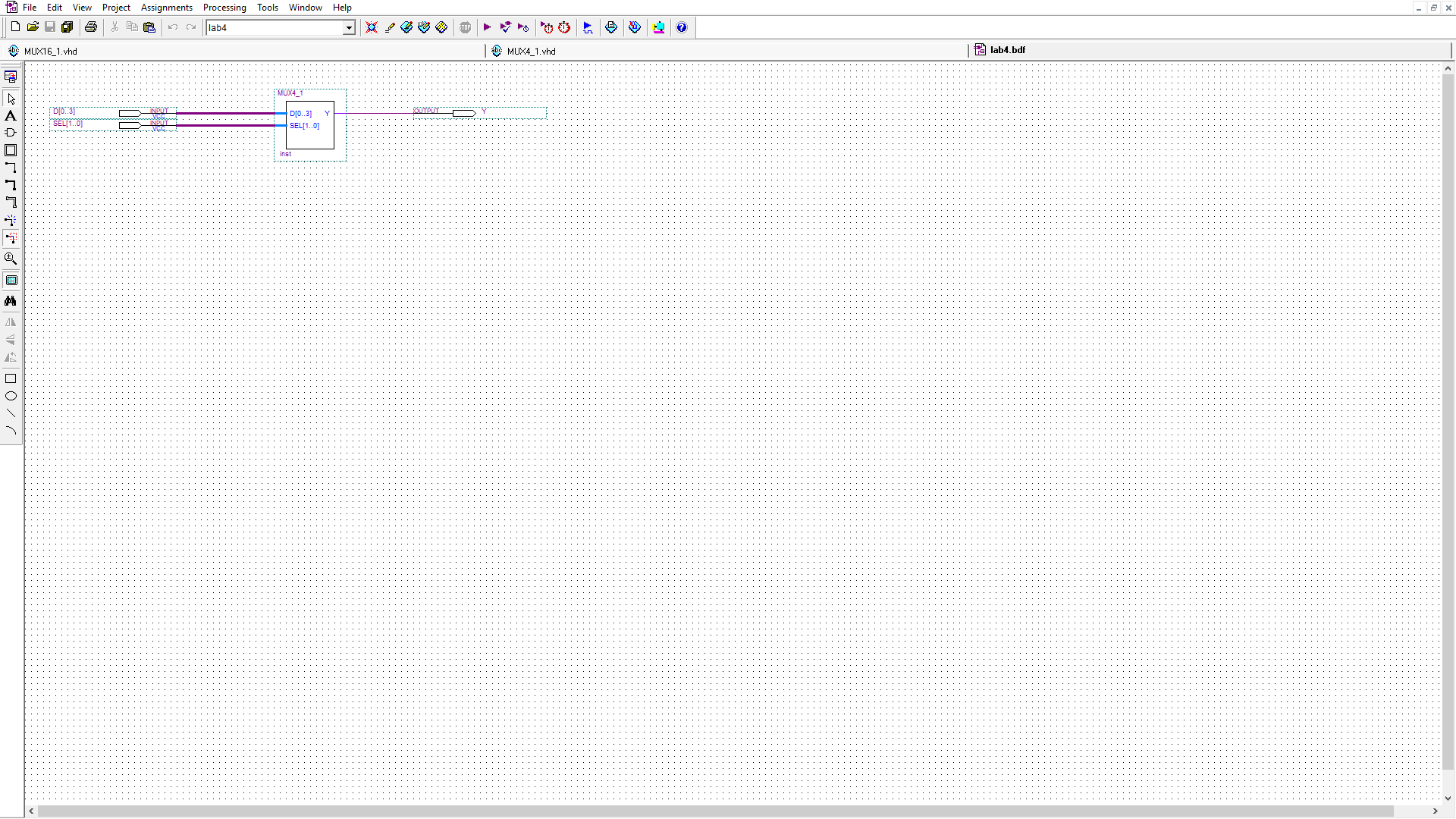
Lab4

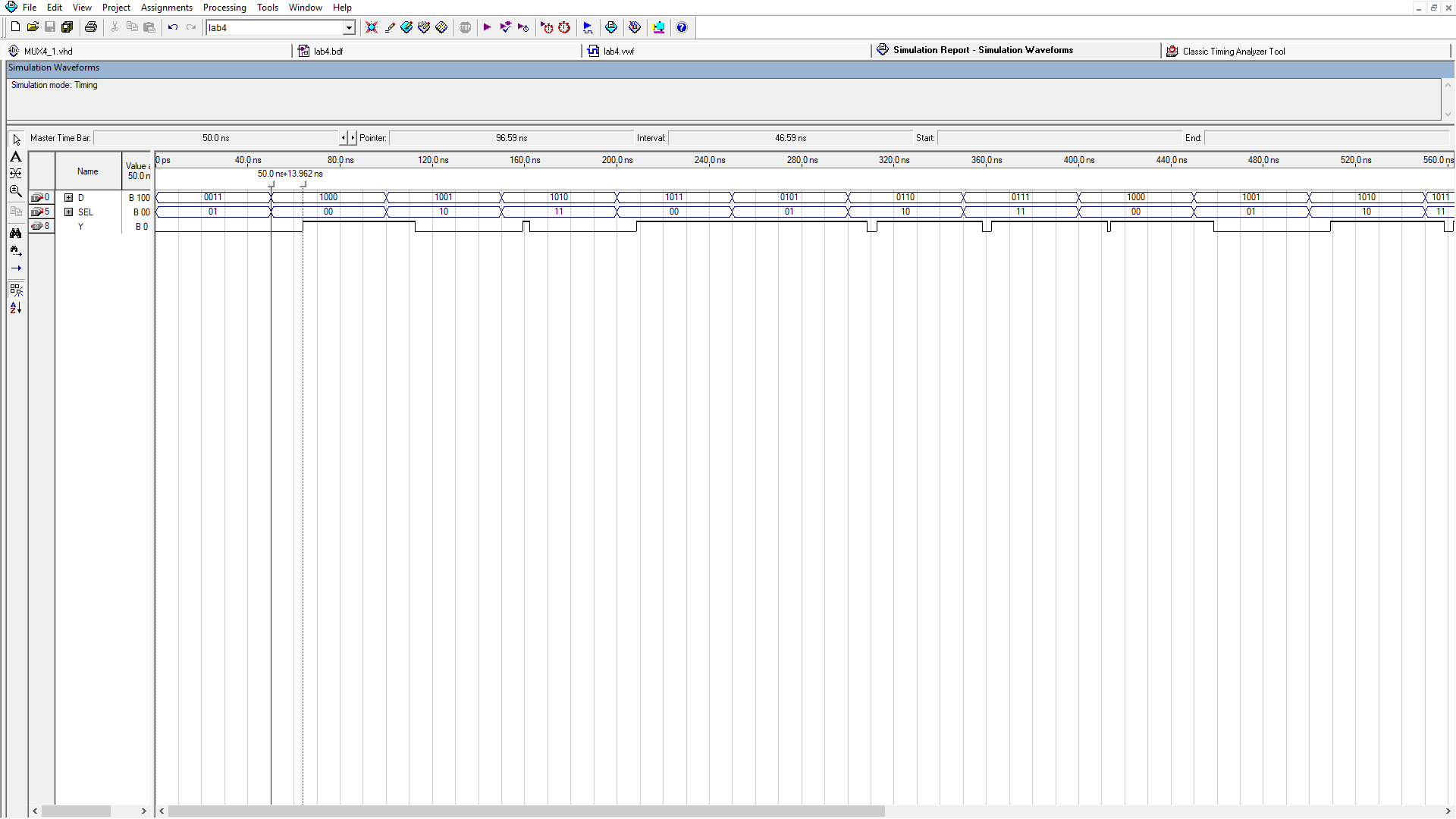
Αρχικά υλοποιούμε έναν πολυπλέκτη 4 σε 1, ο κώδικας του οποίου φαίνεται παρακάτω:



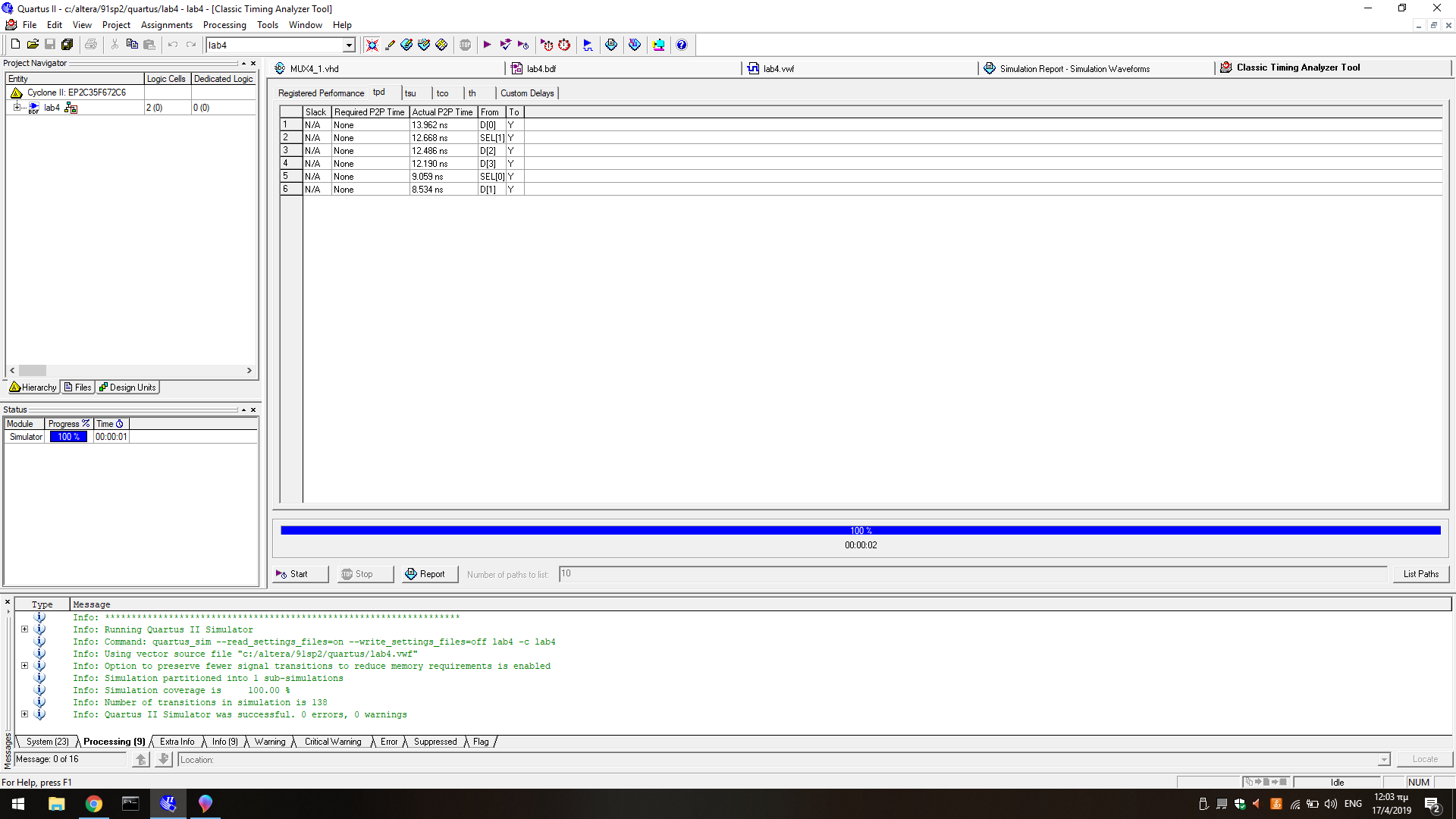
Δημιουργούμε το παρακάτω σχηματικό αφού έχουμε κάνει τον πολυπλέκτη σύμβολο για να ελέγξουμε την λειτουργία του:



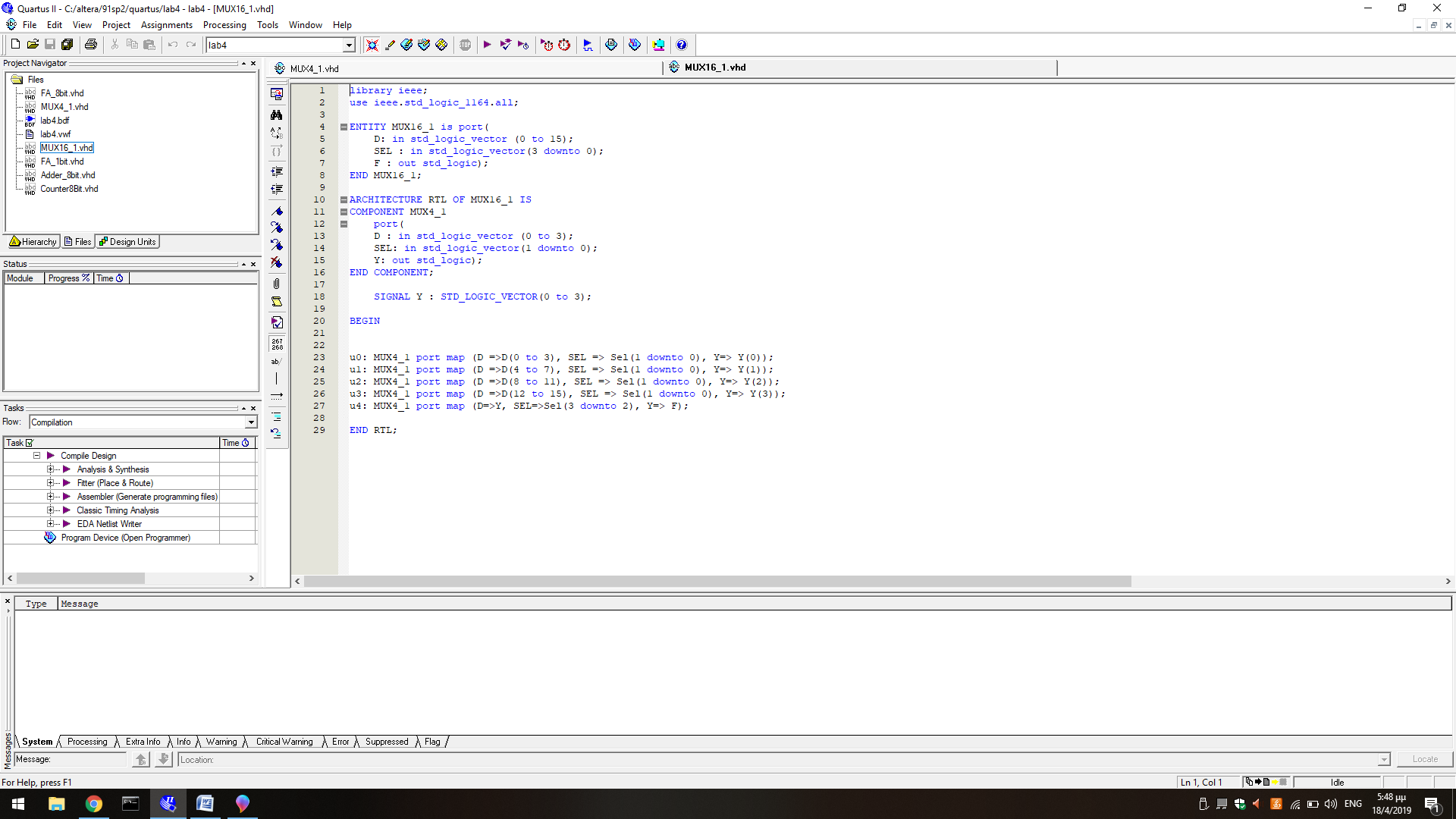
Η κυματομορφή είναι η ακόλουθη, με μέγιστη καθυστέρηση αυτή από το D0 μέχρι το Y που είναι 13.962ns όπως θα δείξουμε παρακάτω και με την χρονική εξομοίωση:



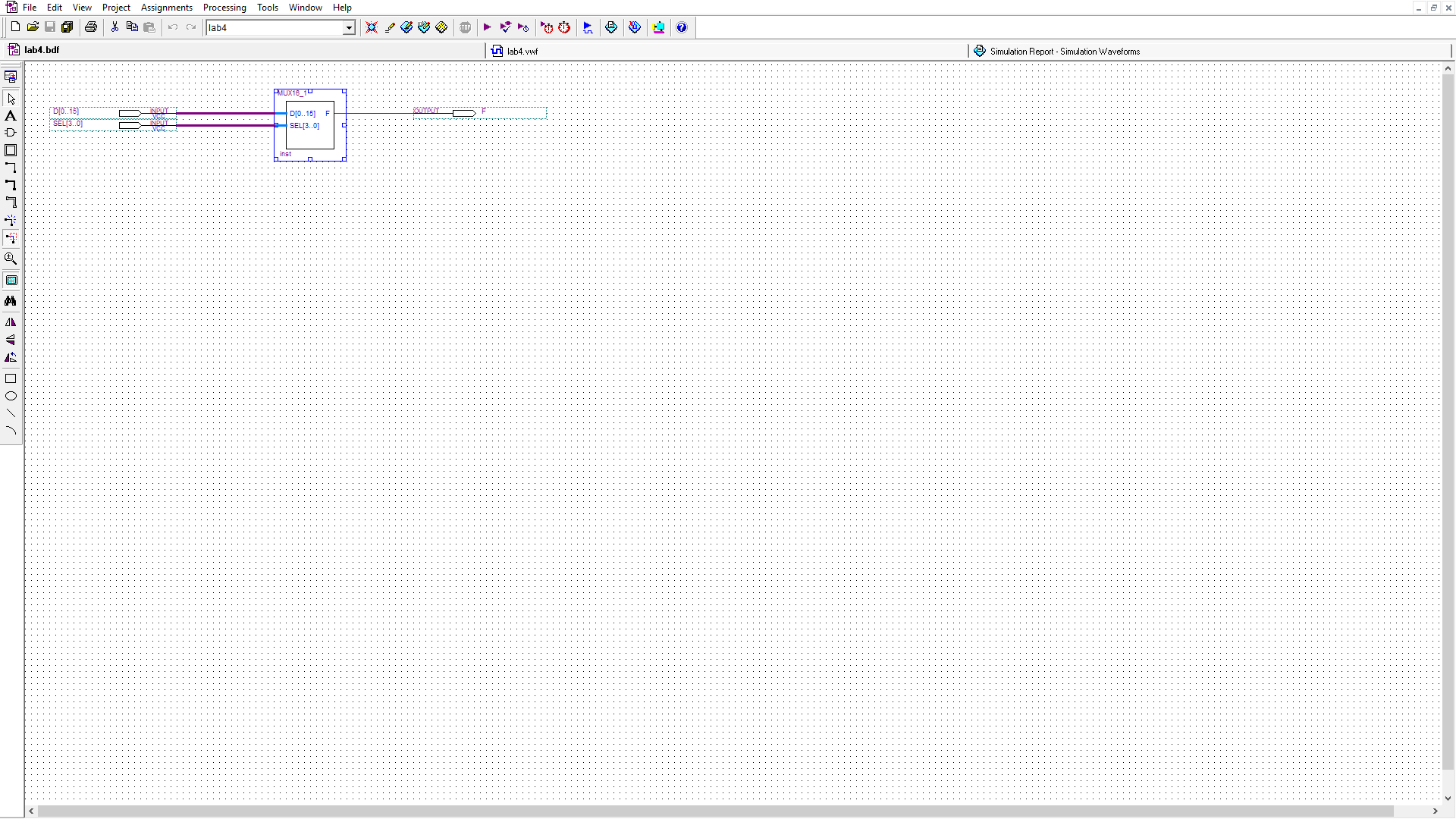
Η χρονική εξομοίωση είναι:



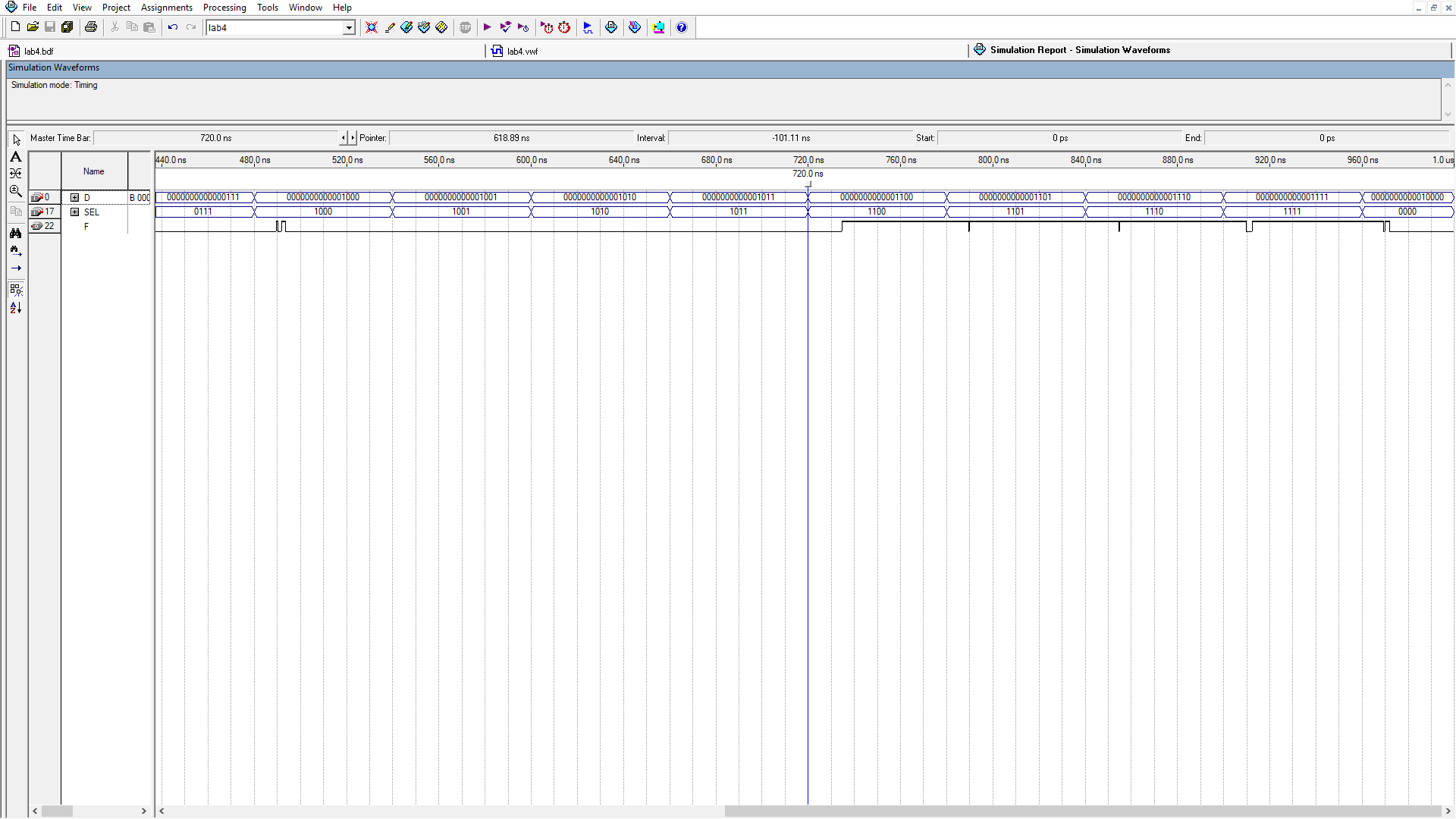
Έπειτα δημιουργούμε ένας πολυπλέκτη 16 σε 1 χρησιμοποιώντας ως component τον πολυπλέκτη που φτιάξαμε προηγουμένως, ο κώδικας φαίνεται παρακάτω:



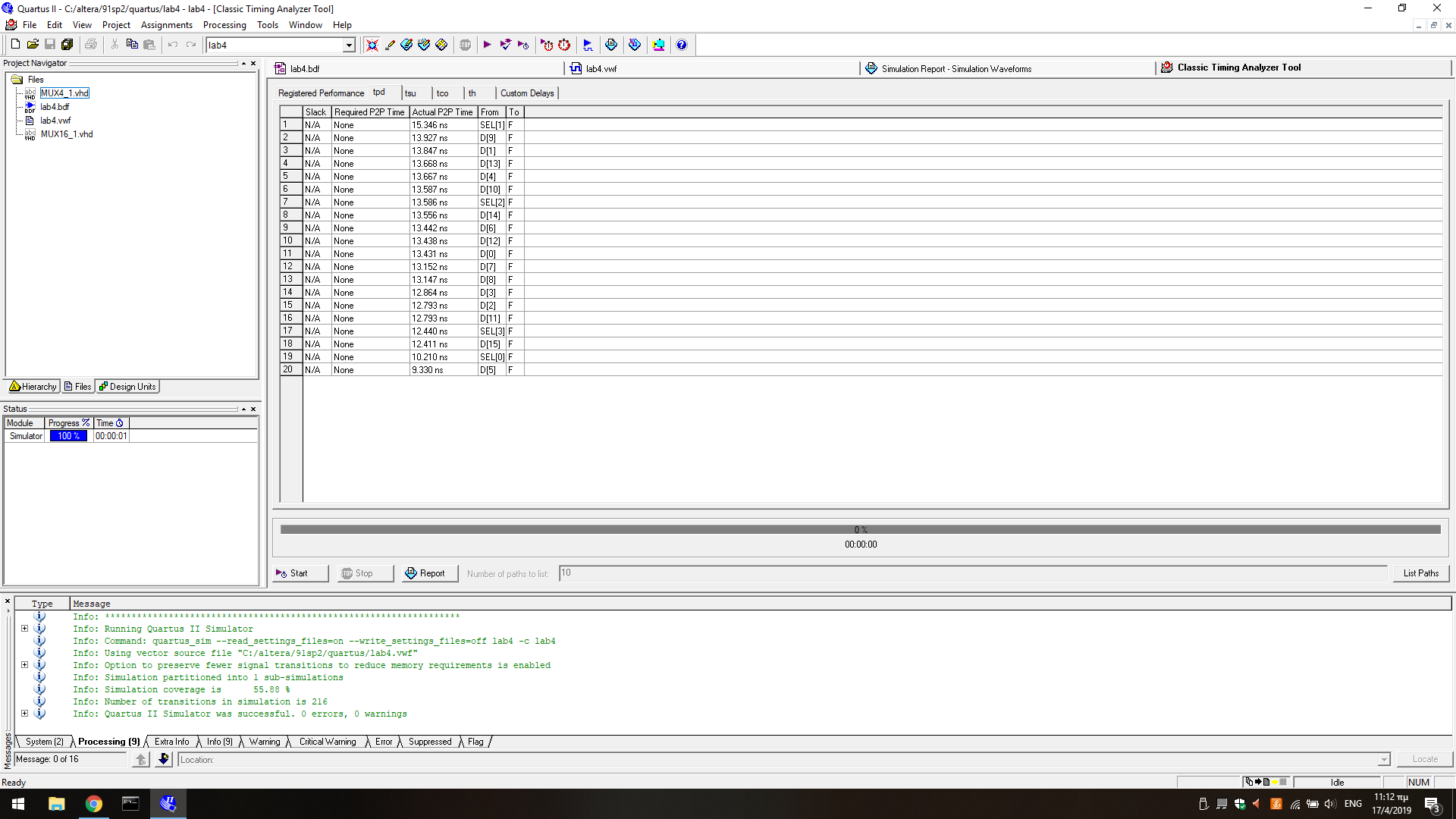
Το σχηματικό για να ελέγξουμε την σωστή λειτουργία του είναι:



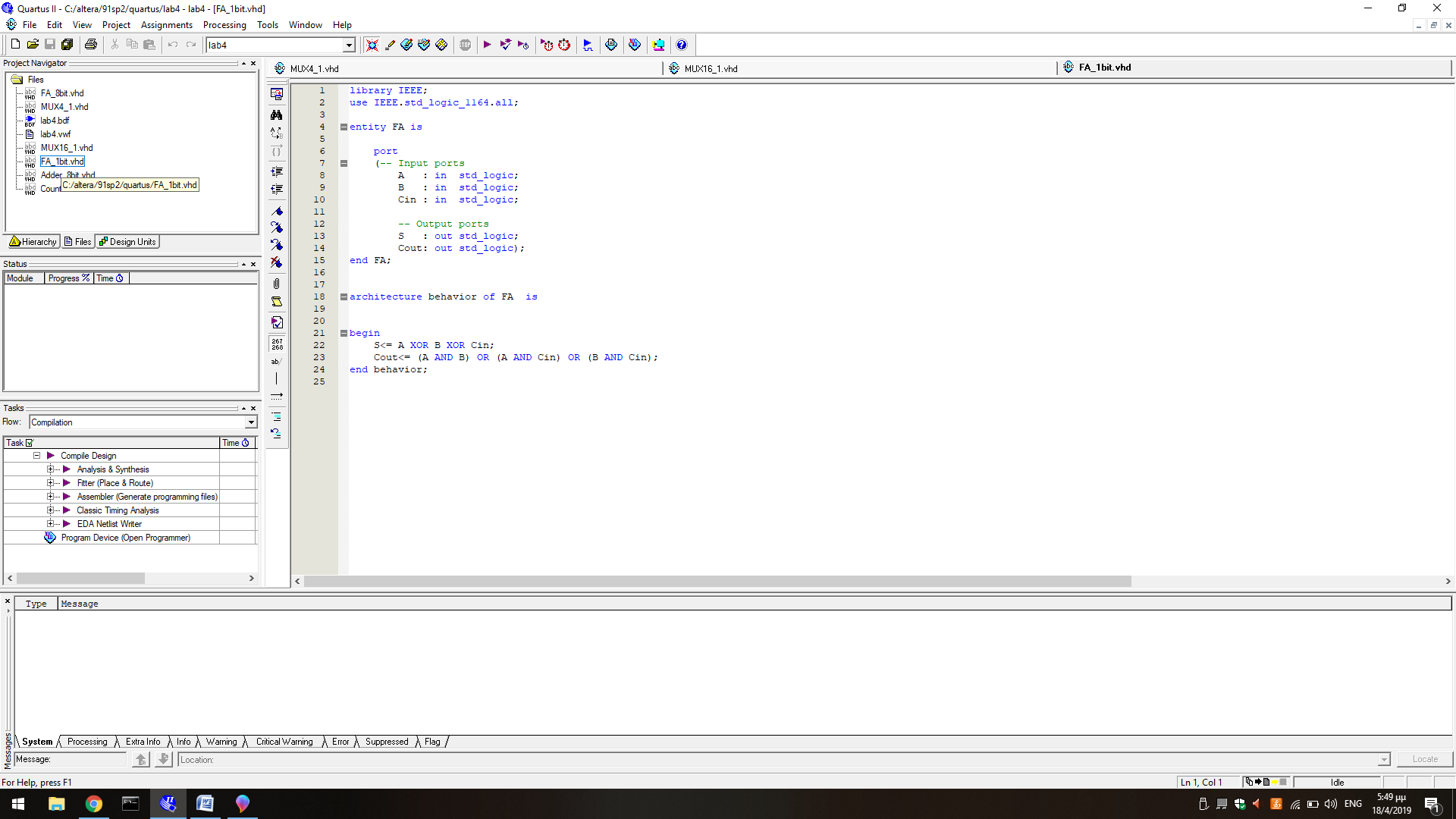
Η κυματομορφή του πολυπλέκτη 16 σε 1 φαίνεται παρακάτω:



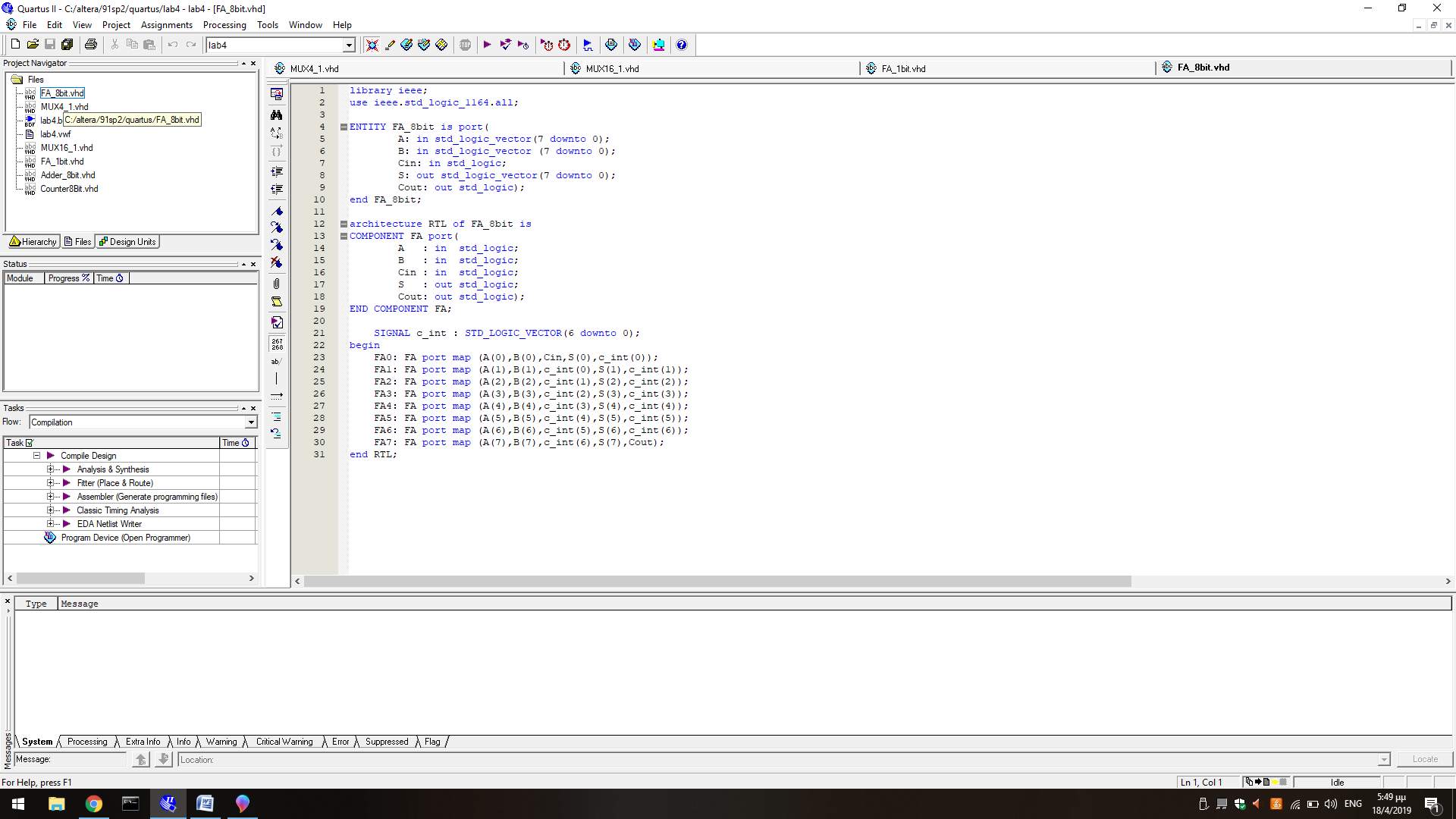
Η μέγιστη καθυστέρηση του πολυπλέκτη 16 σε 1 φαίνεται από την χρονική εξομοίωση και είναι αυτή από την είσοδο SEL[1] έως το F 15.346ns:



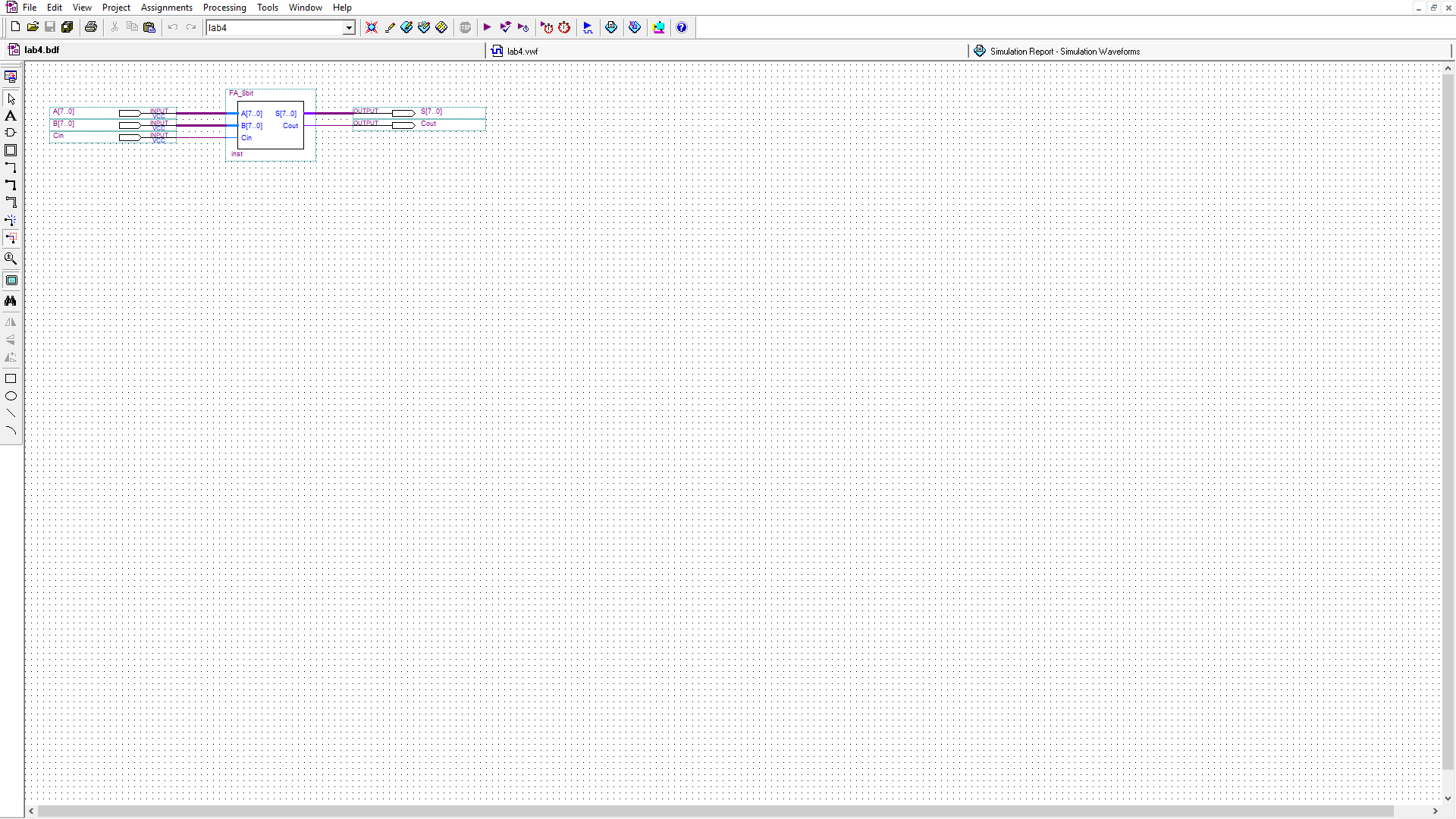
Στην συνέχεια δημιουργούμε έναν πλήρη αθροιστή ο κώδικας είναι ο ακόλουθος:



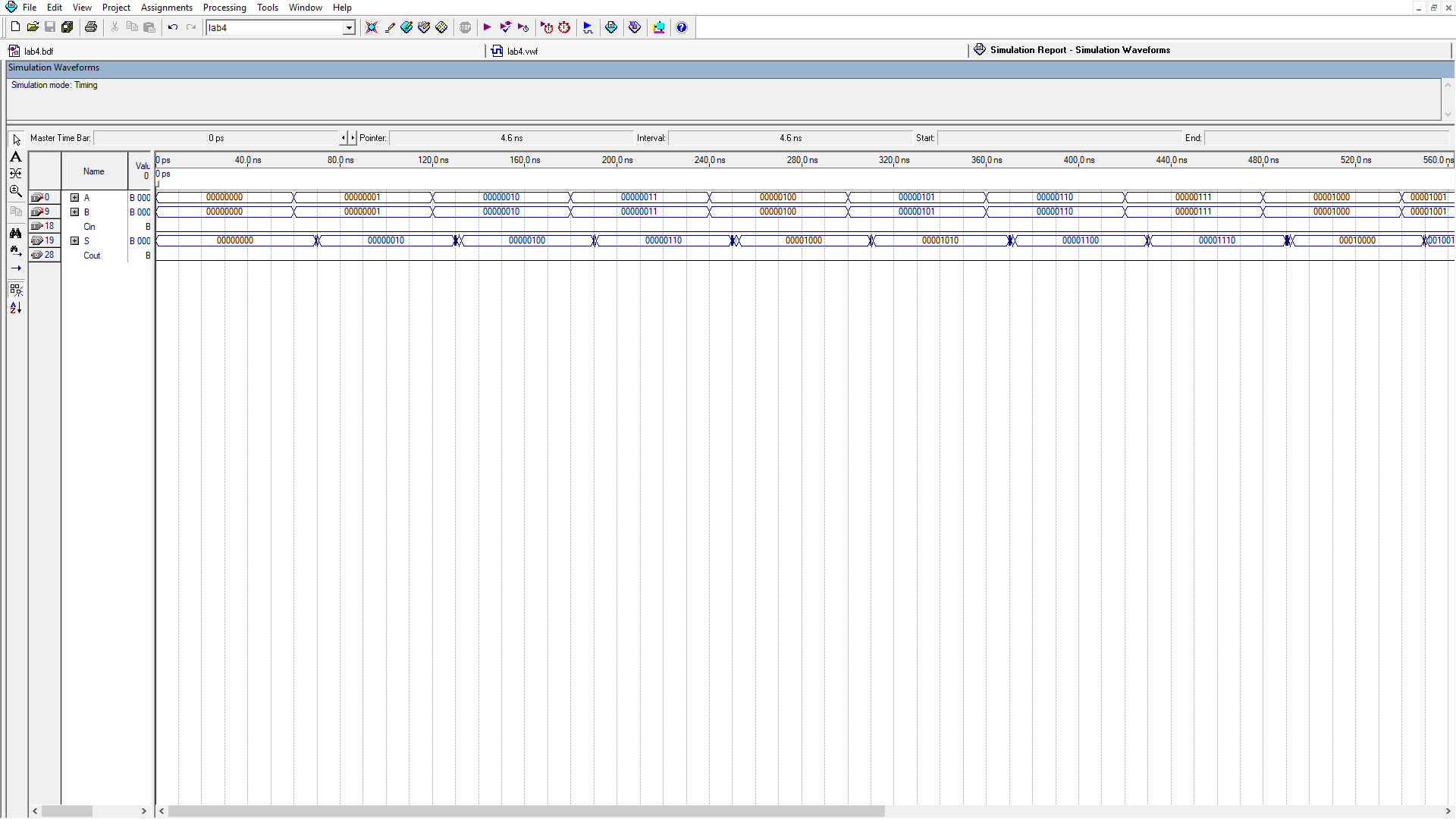
Χρησιμοποιώντας τον πλήρη αθροιστή δημιουργούμε έναν αθροιστή των 8 bits, ο κώδικας φαίνεται παρακάτω:



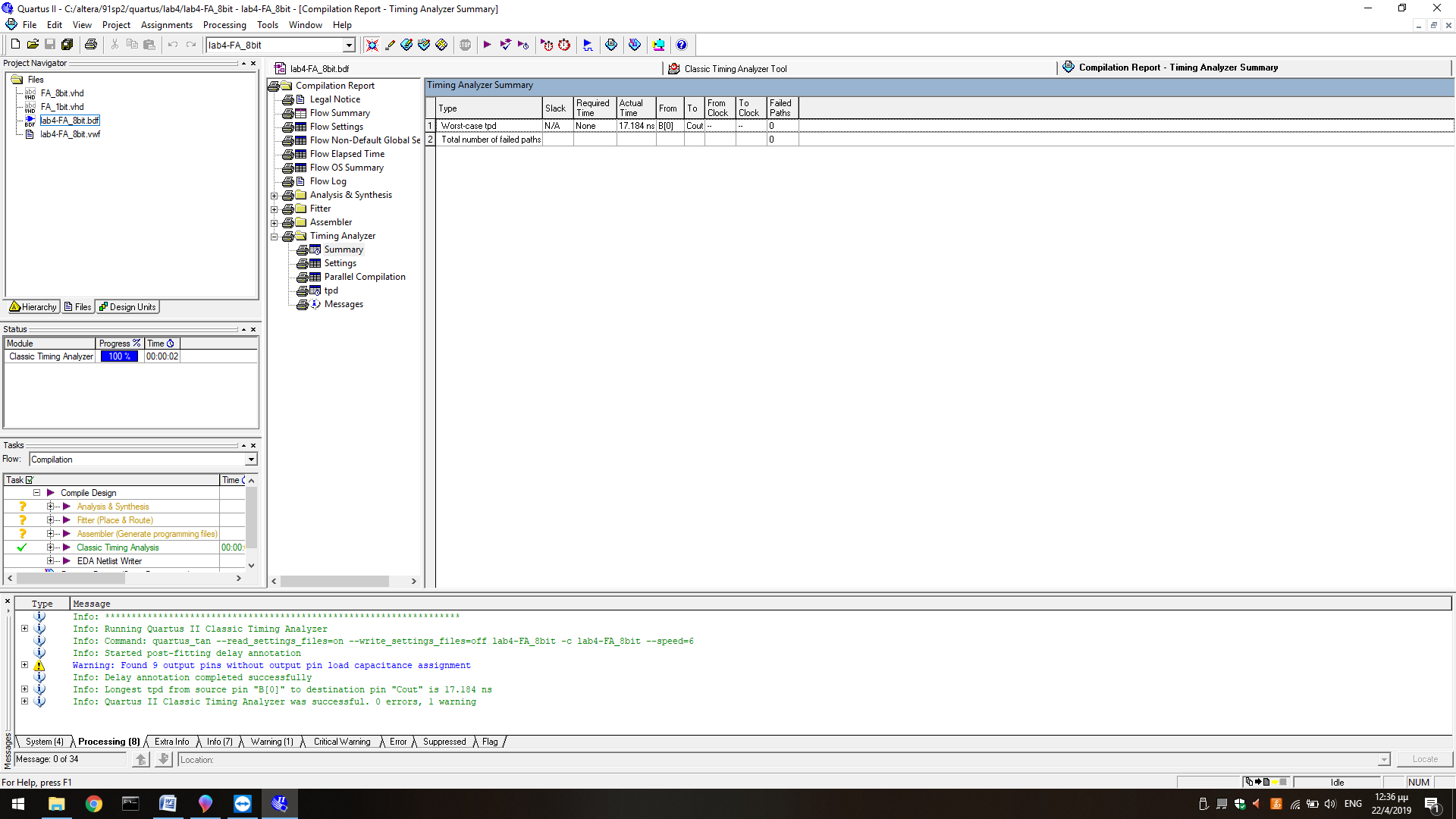
Το σχηματικό με τον αθροιστή των 8 bits είναι το παρακάτω:



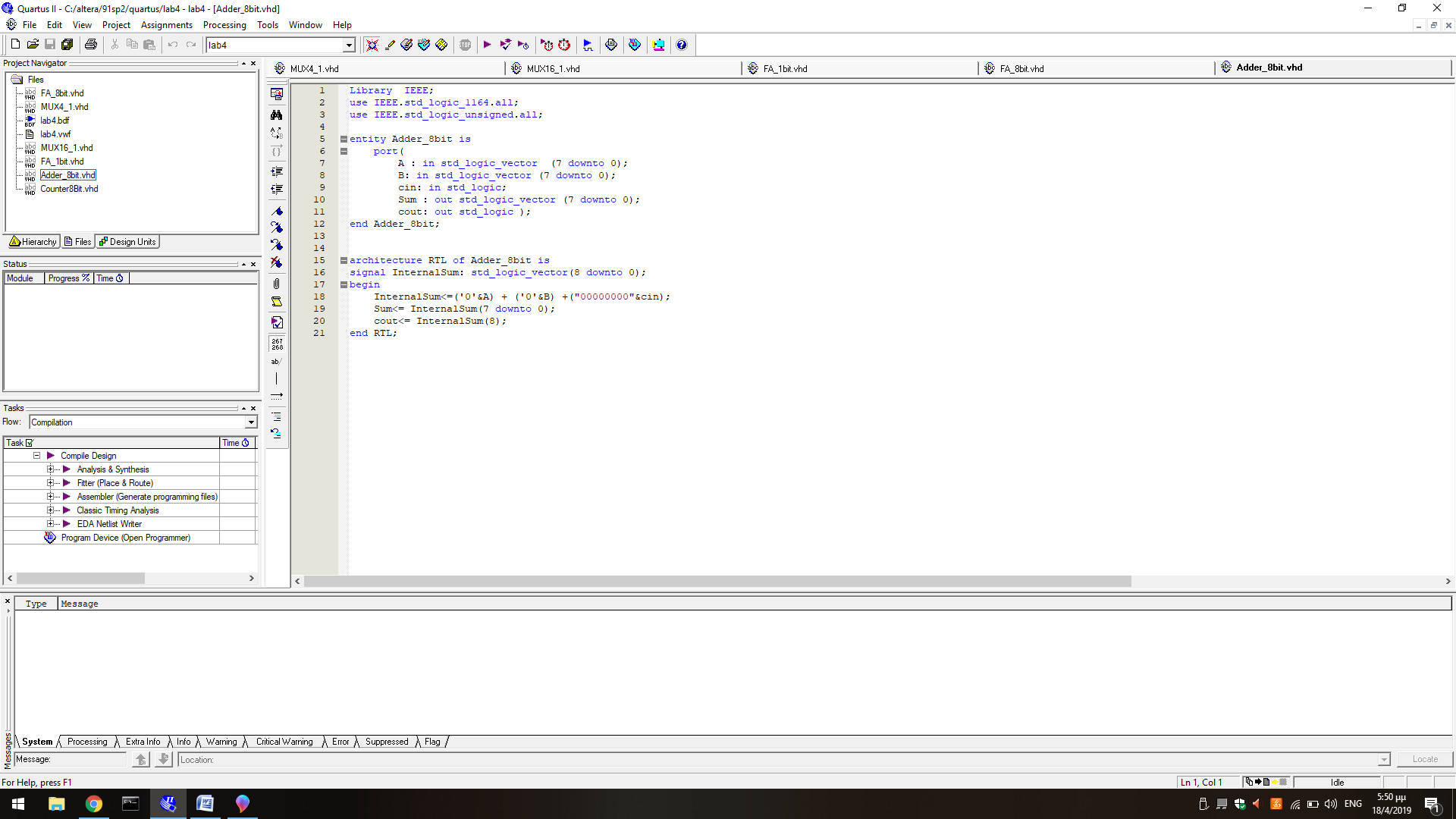
Η κυματομορφή του παραπάνω αθροιστή απεικονίζεται στην επόμενη φωτογραφία και επιβεβαιώνει την ορθή λειτουργία του:



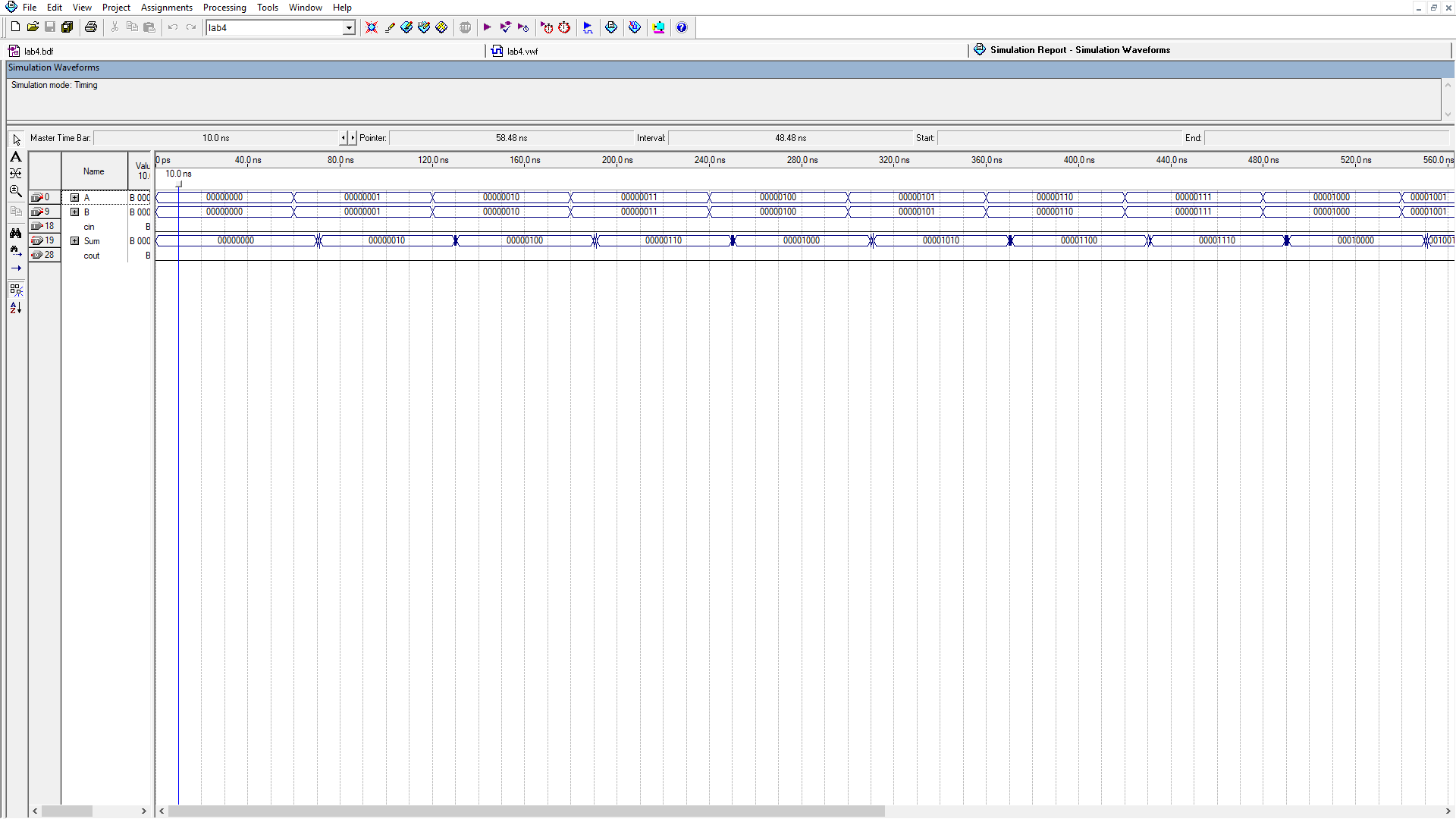
Με την μέγιστη καθυστέρηση να είναι αυτή από το B[0] έως το Cout 17.180 ns :



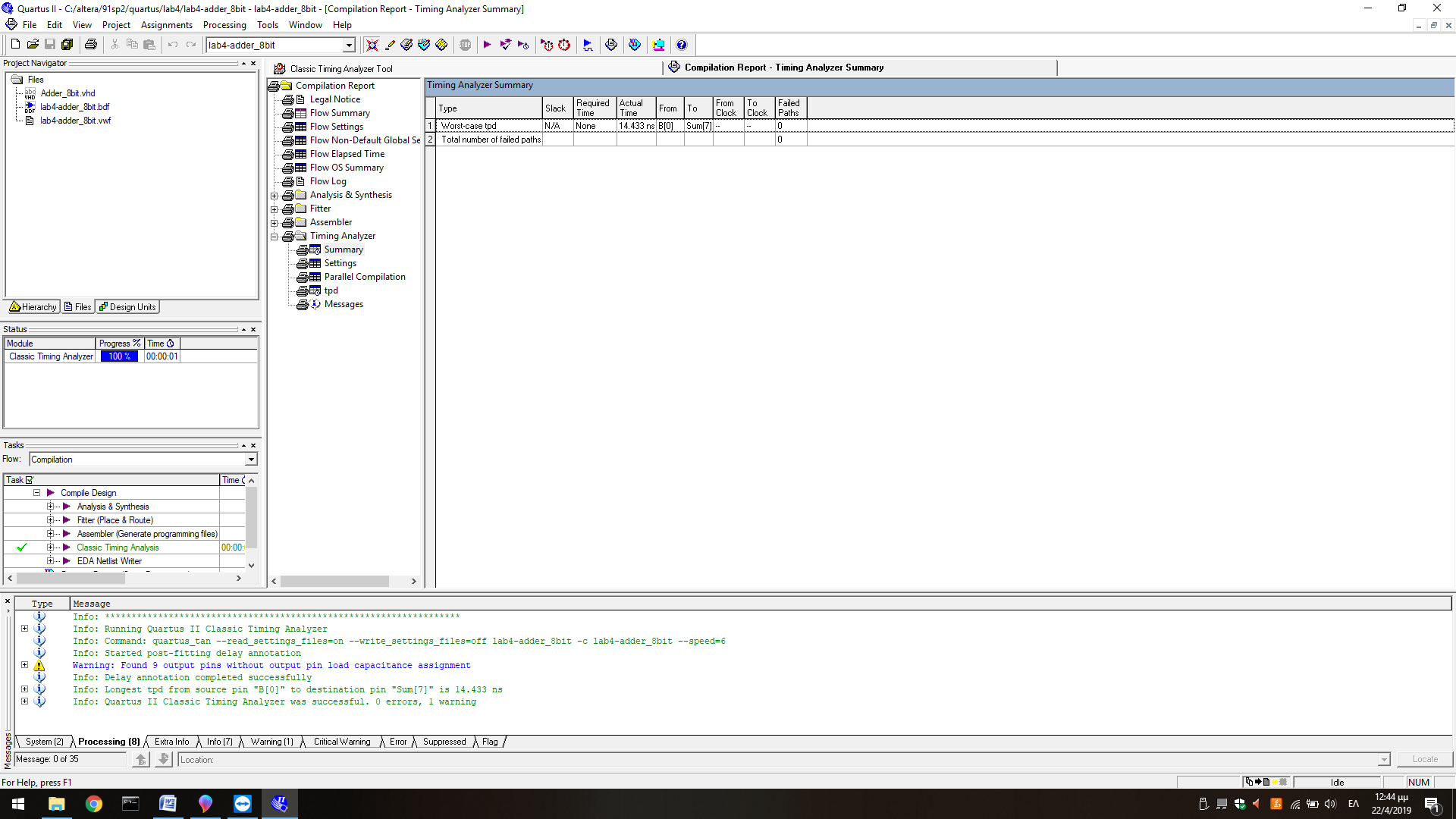
Έπειτα συνεχίζουμε γράφοντας σε VHDL ένας αθροιστή με το σύμβολο `+`.



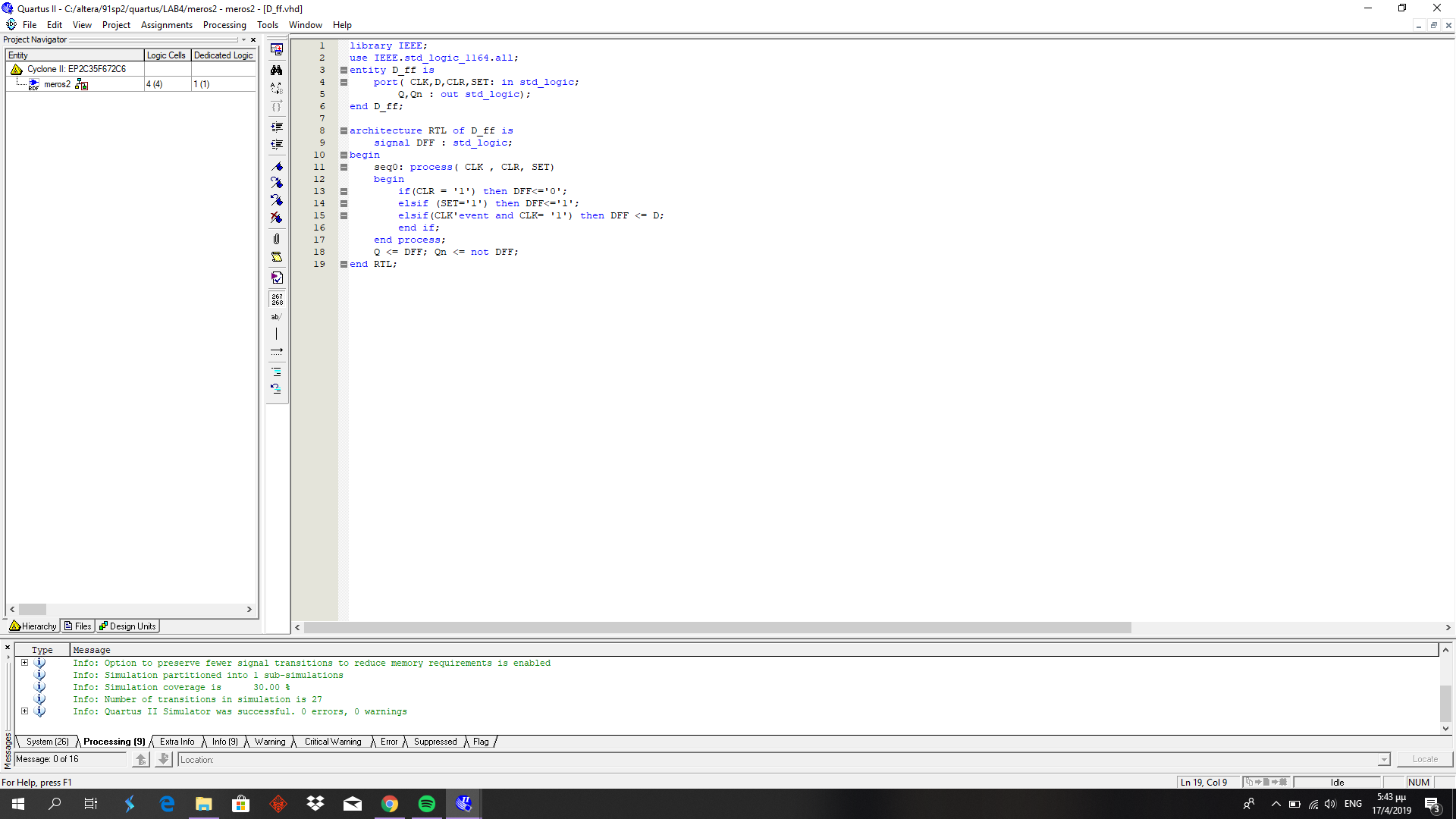
Στην παρακάτω κυματομορφή φαίνεται η σωστή λειτουργία του:



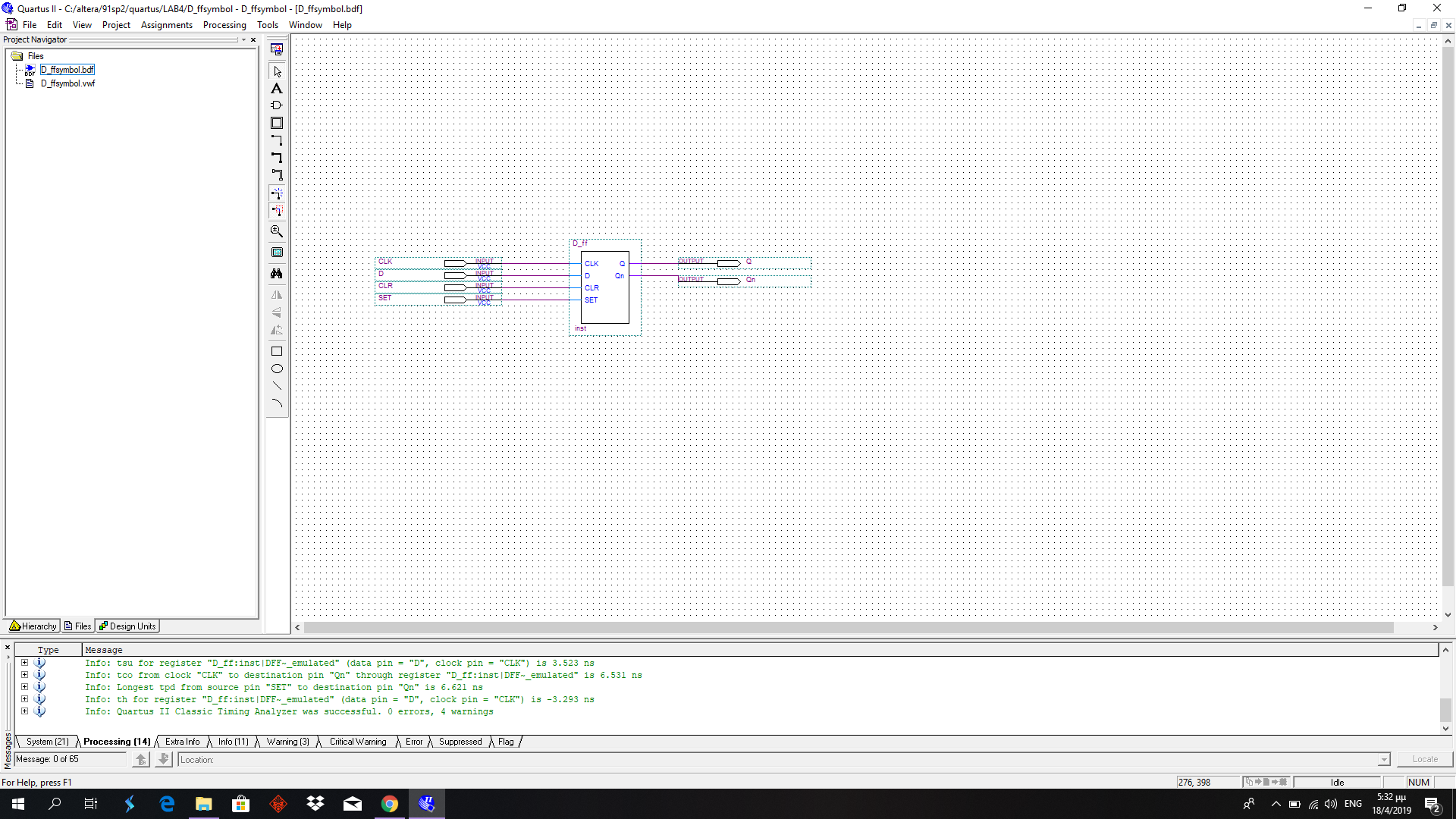
Η μέγιστη καθυστέρηση του κυκλώματος είναι από το B[0] έως το Sum[7], 14.433 ns. Αυτό αποδεικνύει ότι η συγκεκριμένη υλοποίηση του αθροιστή είναι γρηγορότερη και πιο αποδοτική απ’ ότι αυτή του προηγούμενου αθροιστή:

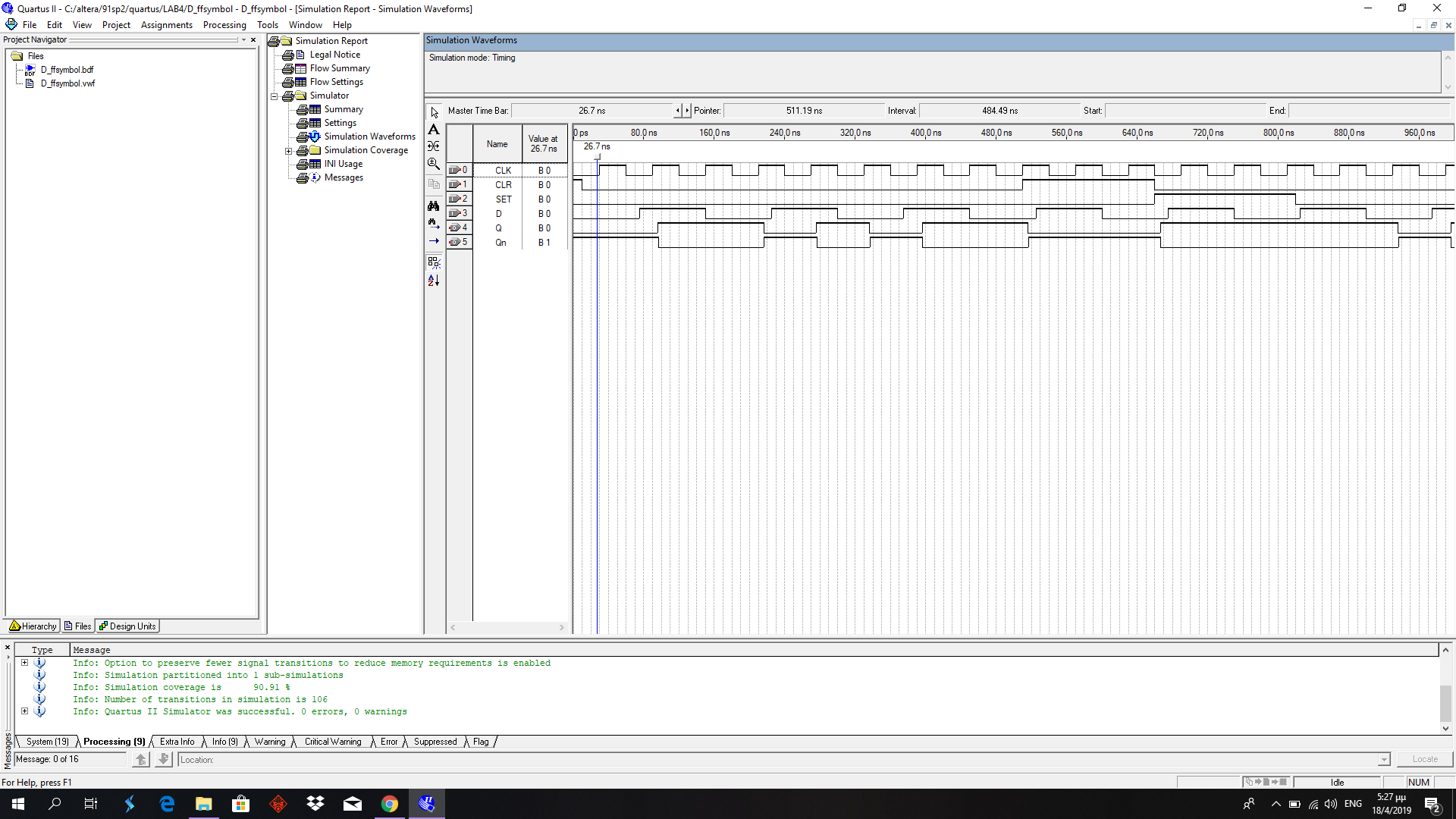


Συνεχίζουμε με την υλοποίηση του D-flip flop σε VHDL και ο κώδικας φαίνεται παρακάτω:

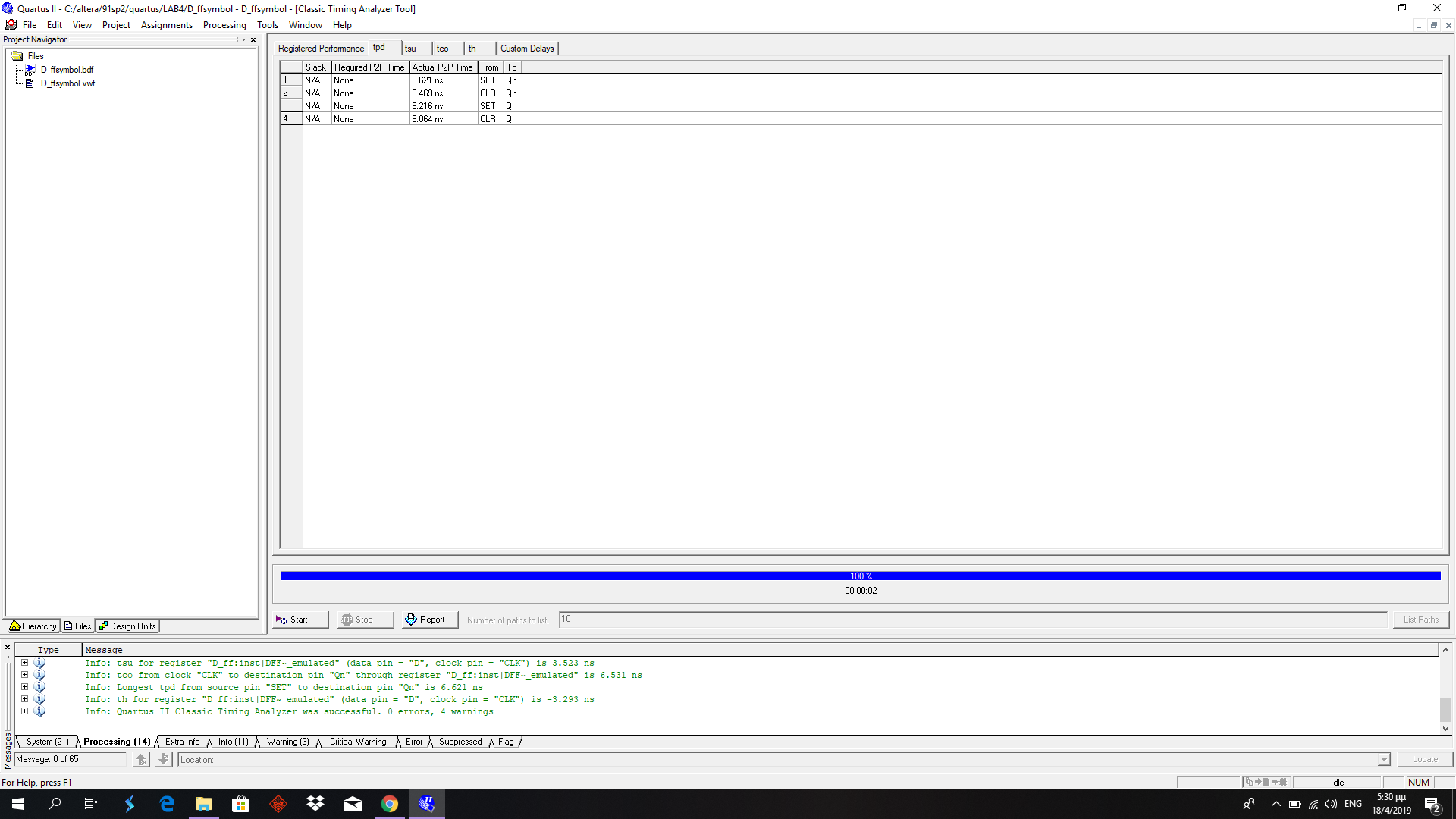


Έπειτα δημιουργούμε το παρακάτω σχηματικό ώστε αργότερα με εξομοίωση να ελέγξουμε την λειτουργία του:

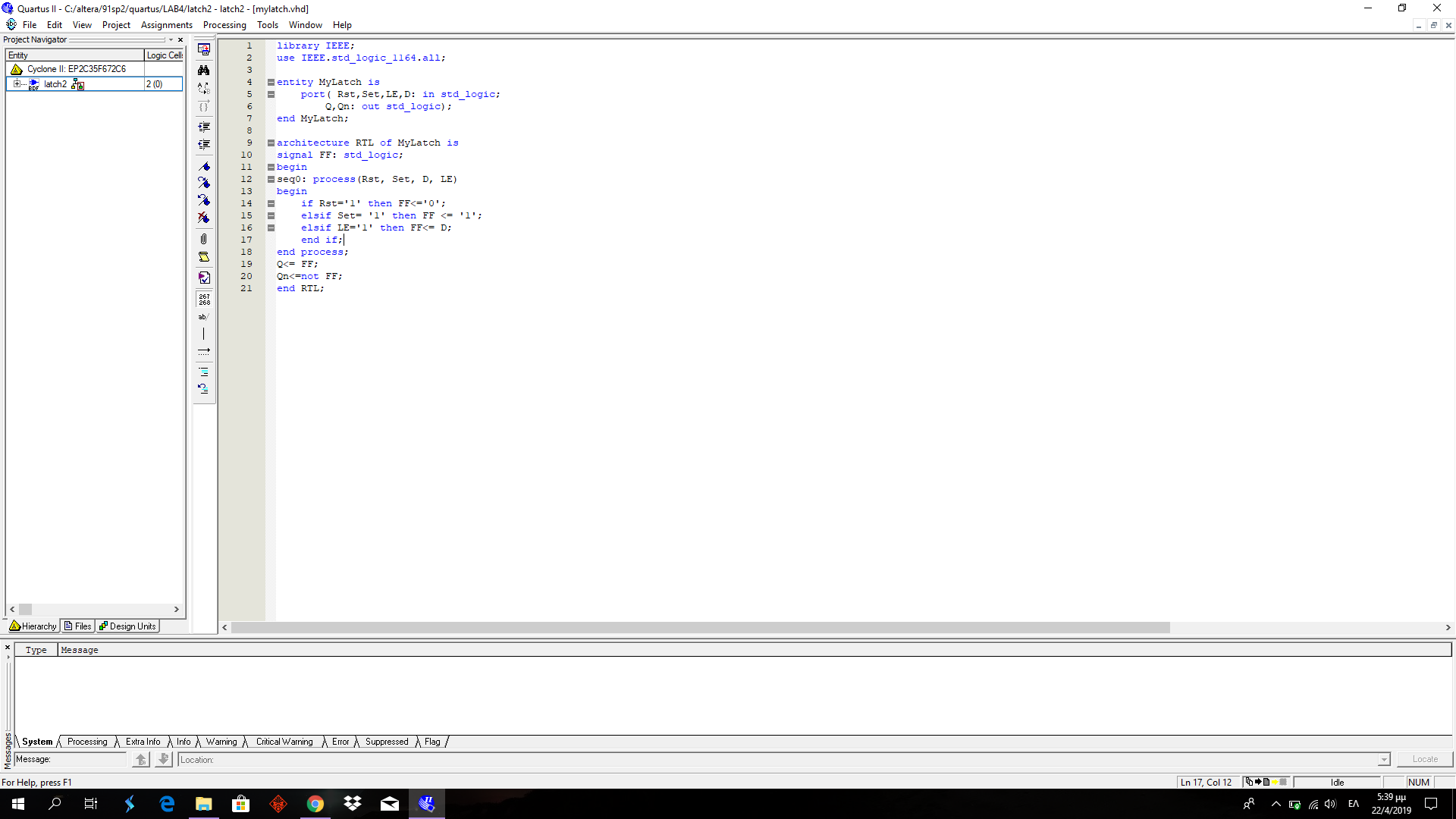


Στην κυματομορφή που ακολουθεί φαίνεται η ορθή του λειτουργία:

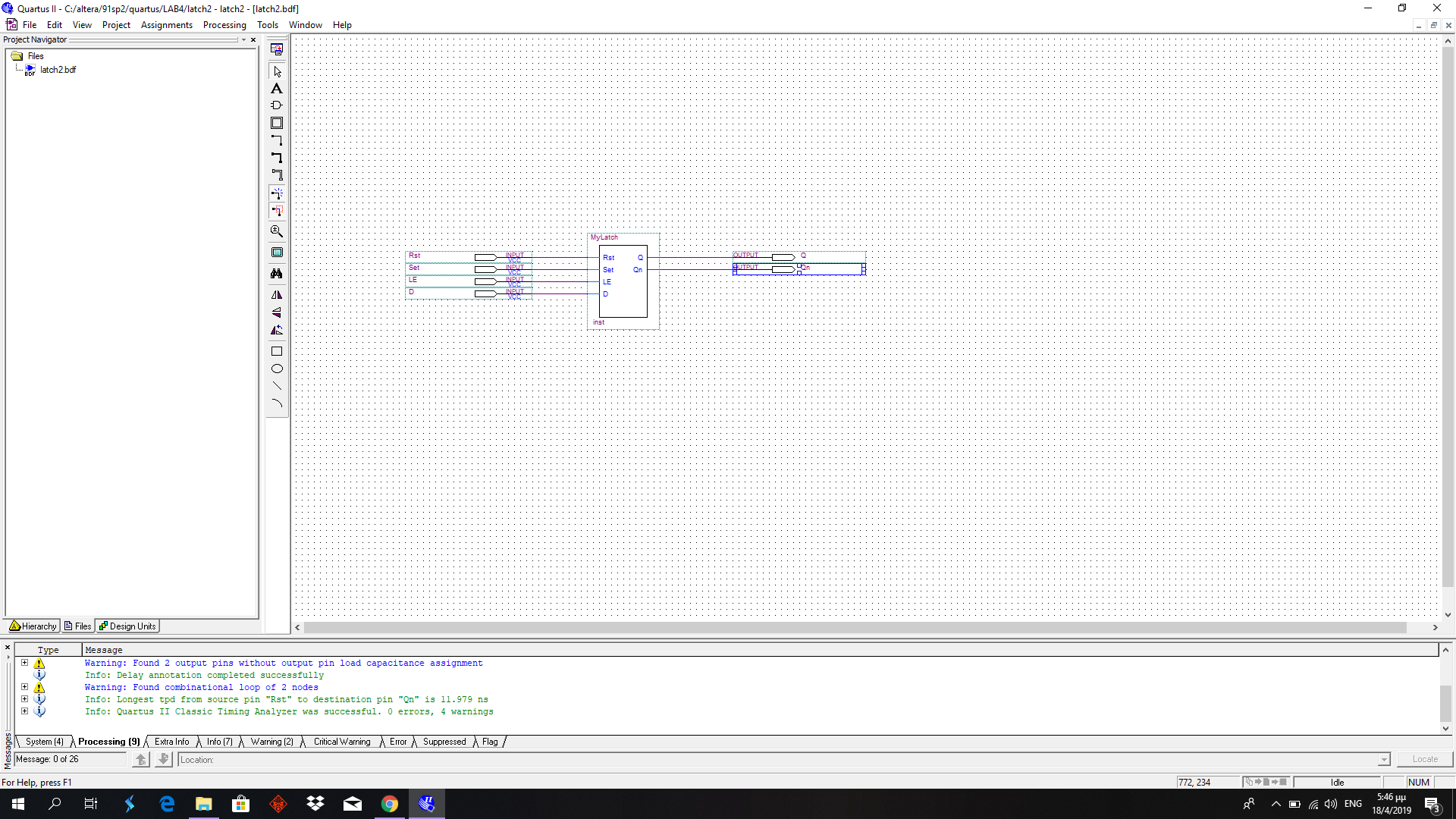
Στην χρονική εξομοίωση που κάναμε φαίνεται ότι η μέγιστη καθυστέρηση του κυκλώματος παρουσιάζεται από την είσοδο SET μέχρι το Qn και είναι 6.621 ns επομένως το ρολόι πρέπει να έχει το λιγότερο 6.621+20% περίοδο, δηλαδή περίπου 8 ns:

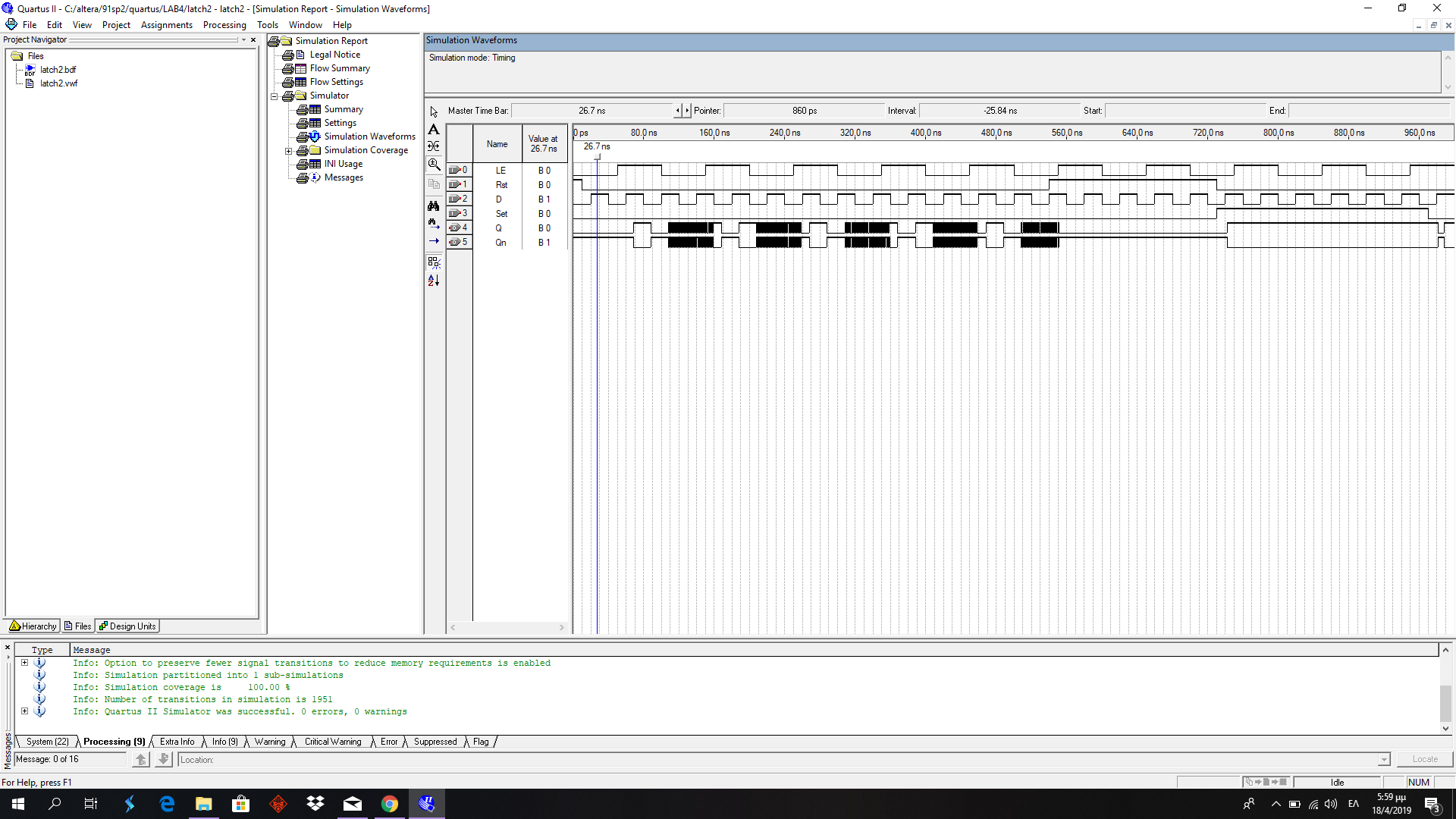


Ακολούθως δημιουργούμε ένα latch σε VHDL:

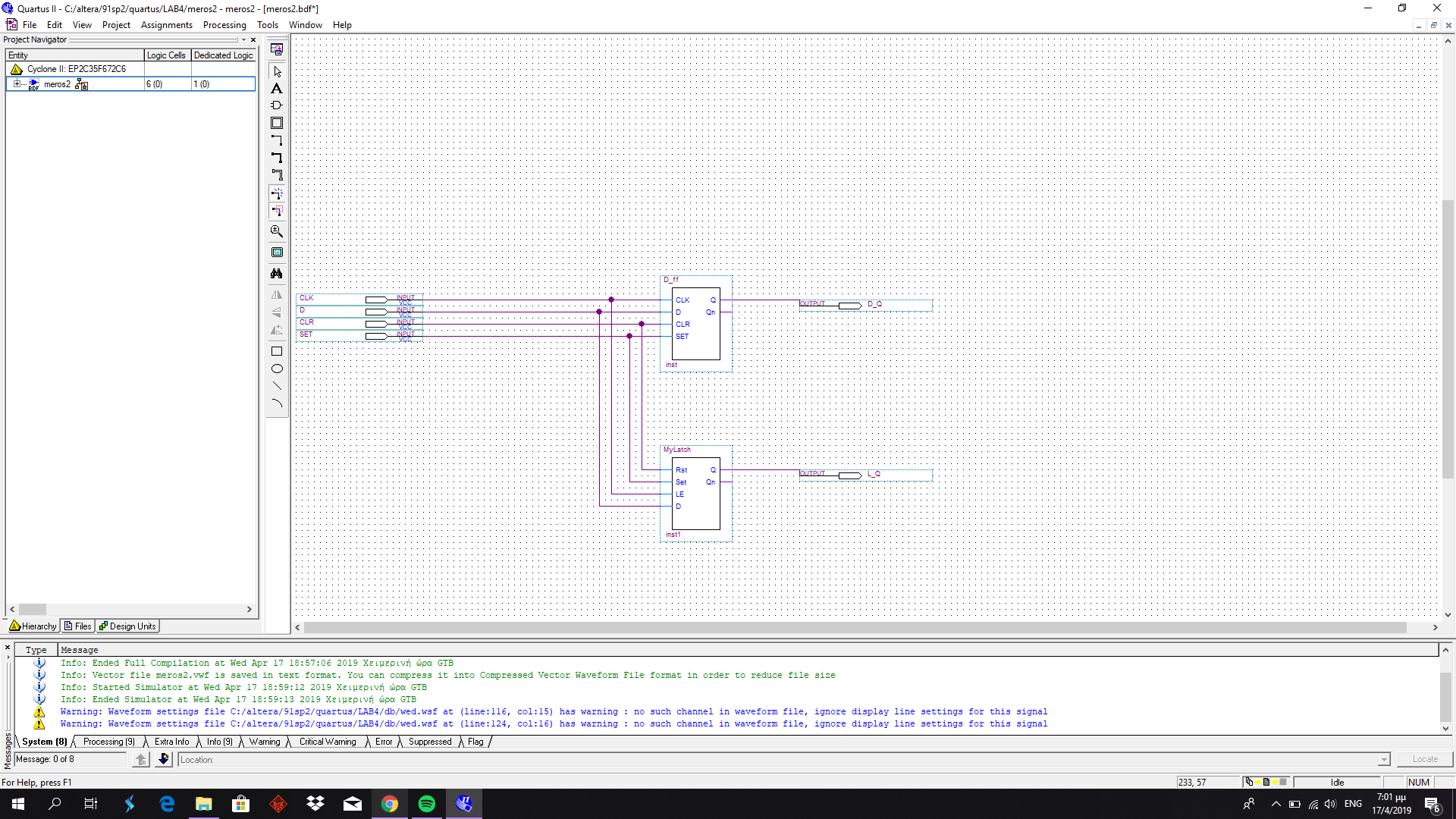


Το σχηματικό για το έλεγχο της σωστής λειτουργίας του Latch φαίνεται παρακάτω:

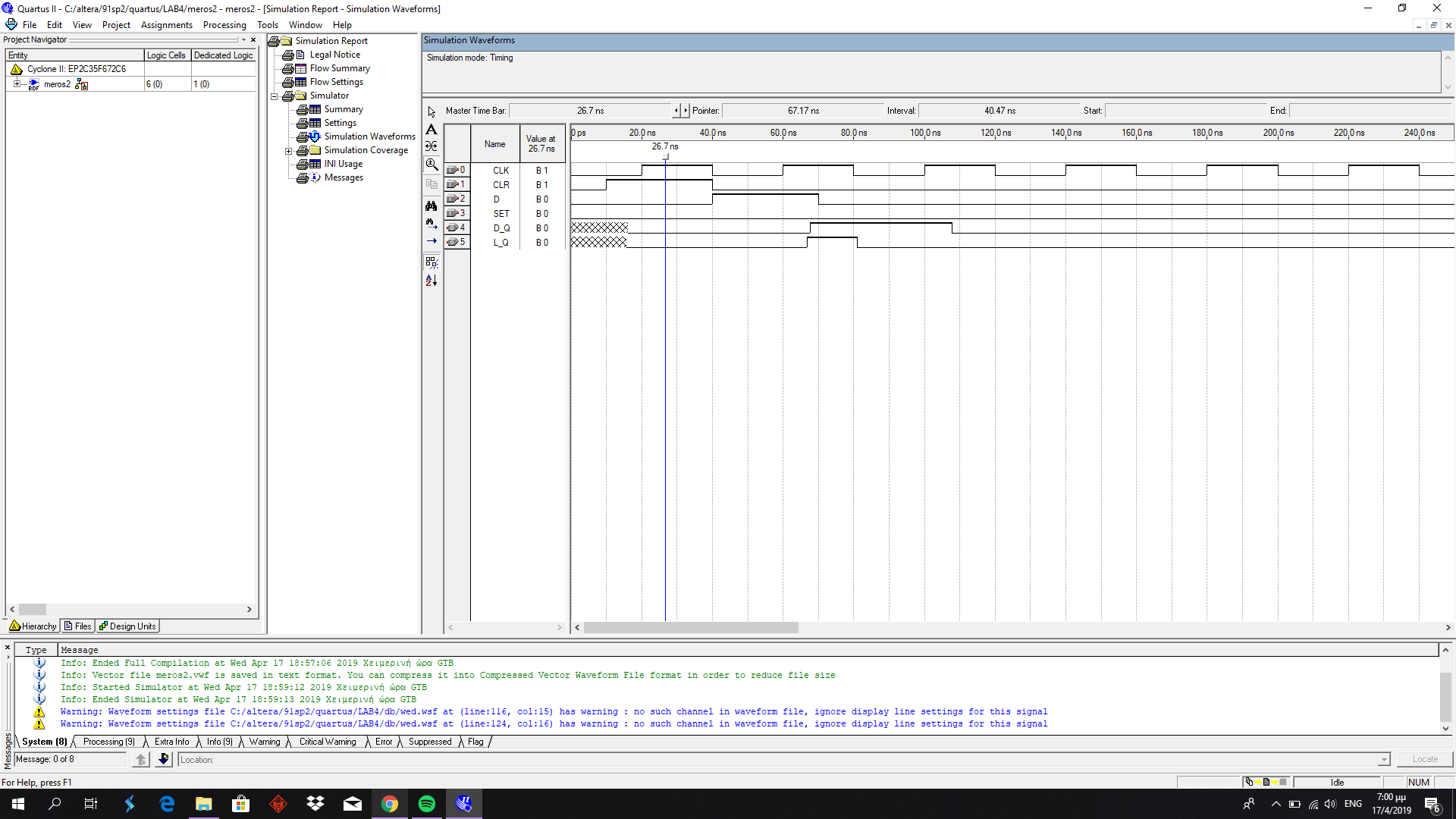


Η κυματομορφή του παραπάνω σχηματικού είναι η ακόλουθη:

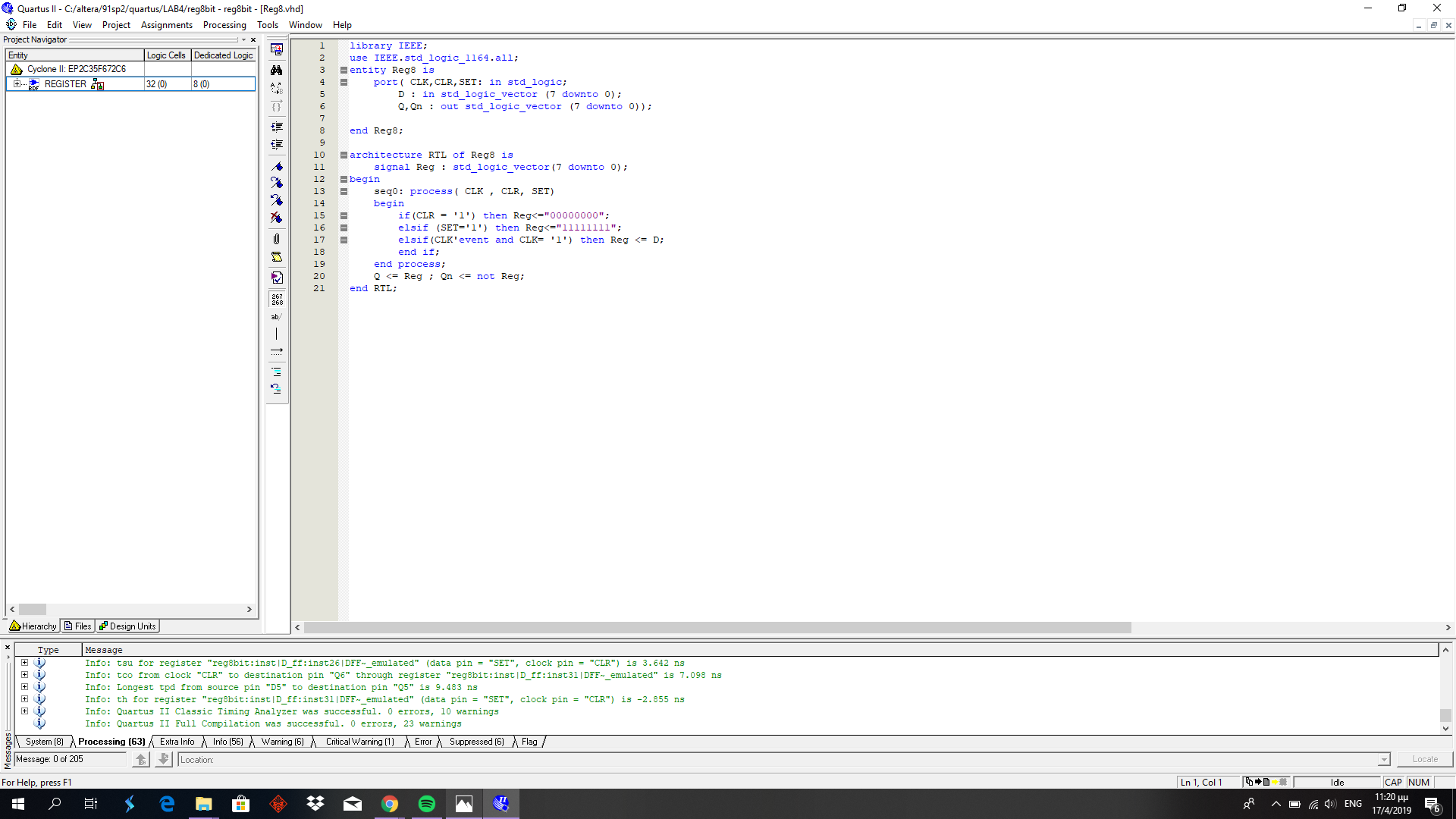
Έπειτα δημιουργούμε το παρακάτω σχηματικό για την σύγκριση της λειτουργίας του D–flip flop και του Latch:



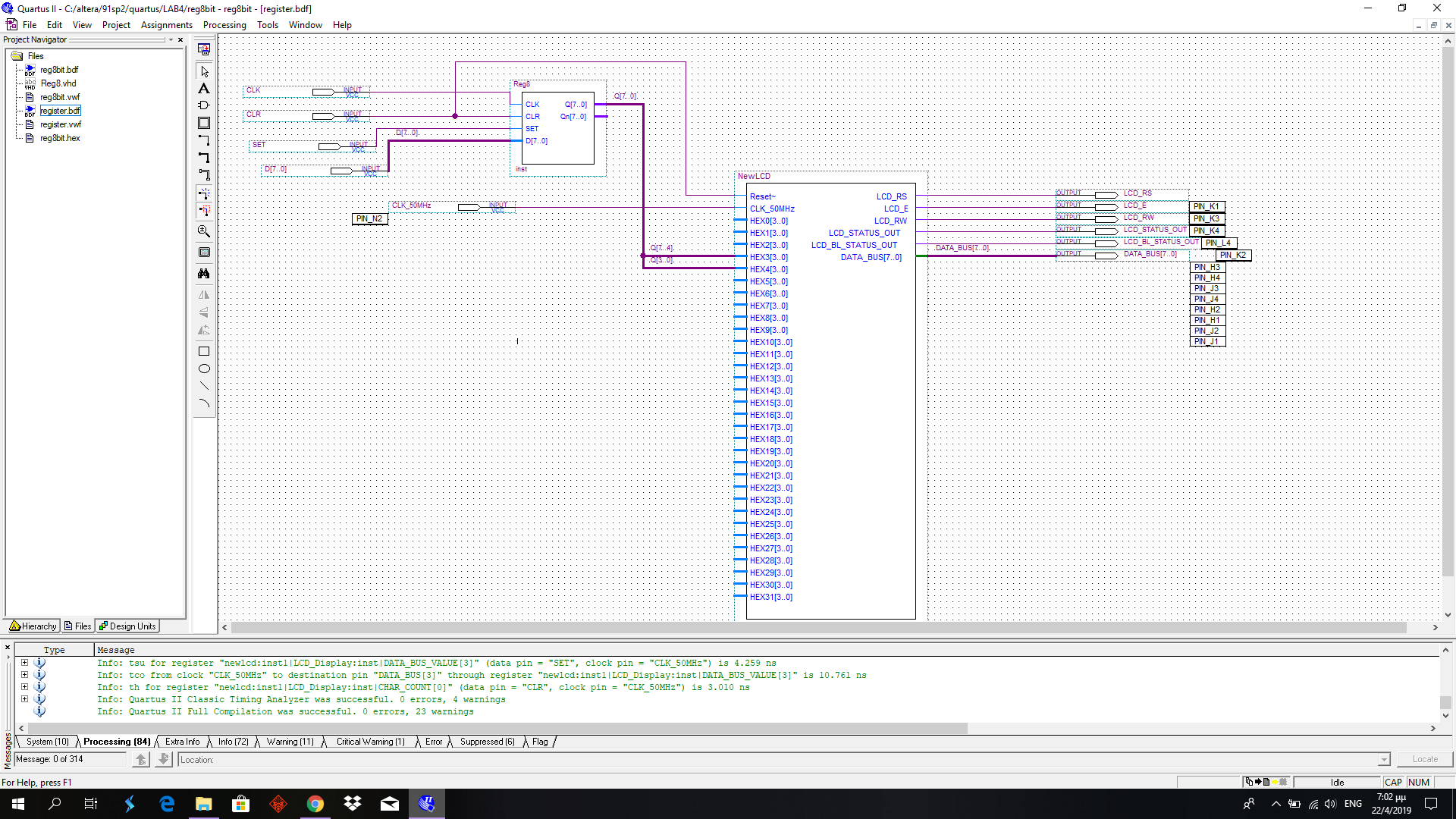
Η κυματομορφή του παραπάνω κυκλώματος είναι η ακόλουθη και επιβεβαιώνει τη διαφορετική λειτουργία της κάθε οντότητας:



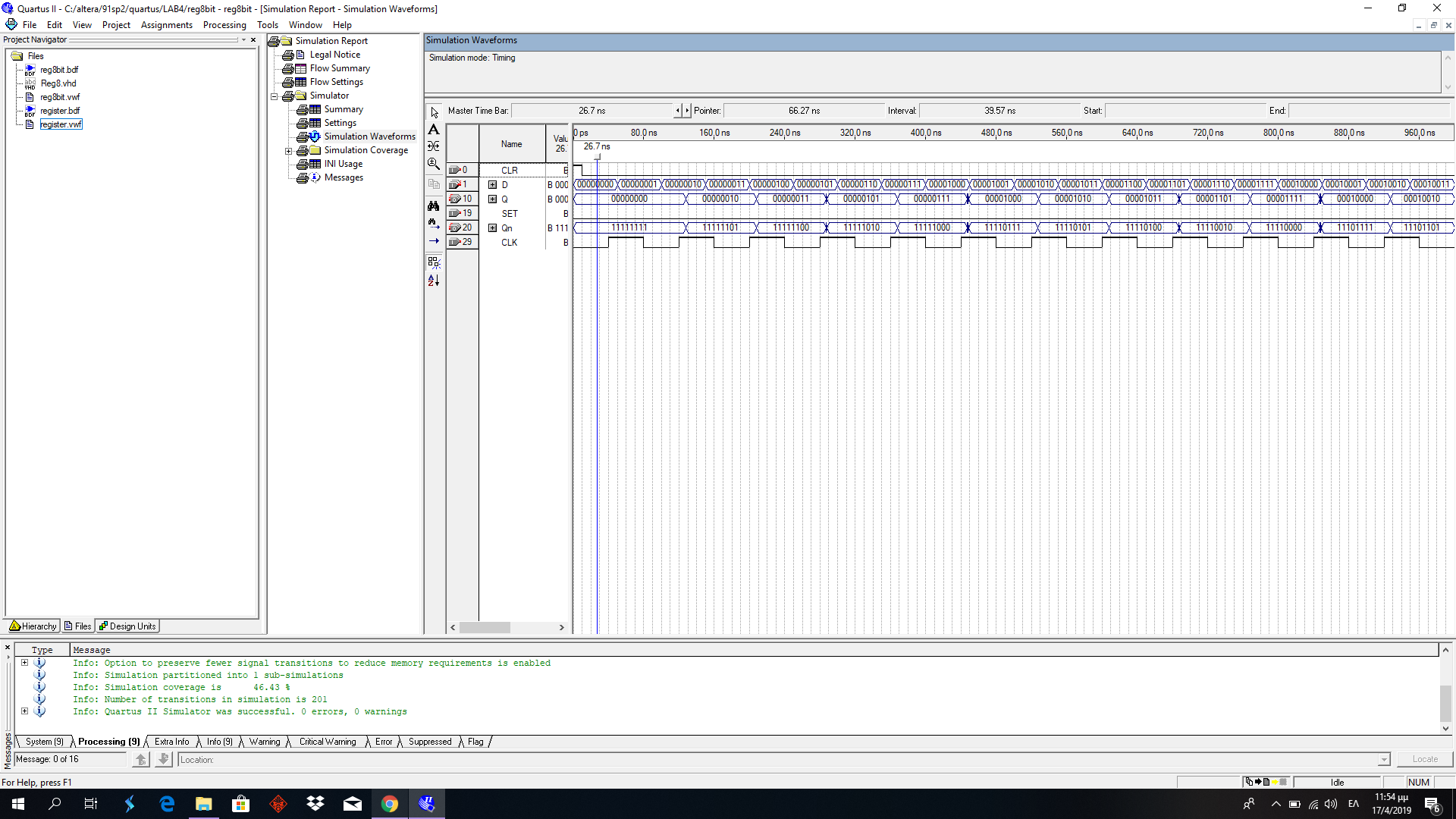
Συνεχίζουμε με τον καταχωρητή των 8 bits και ο κώδικας είναι ο ακόλουθος:



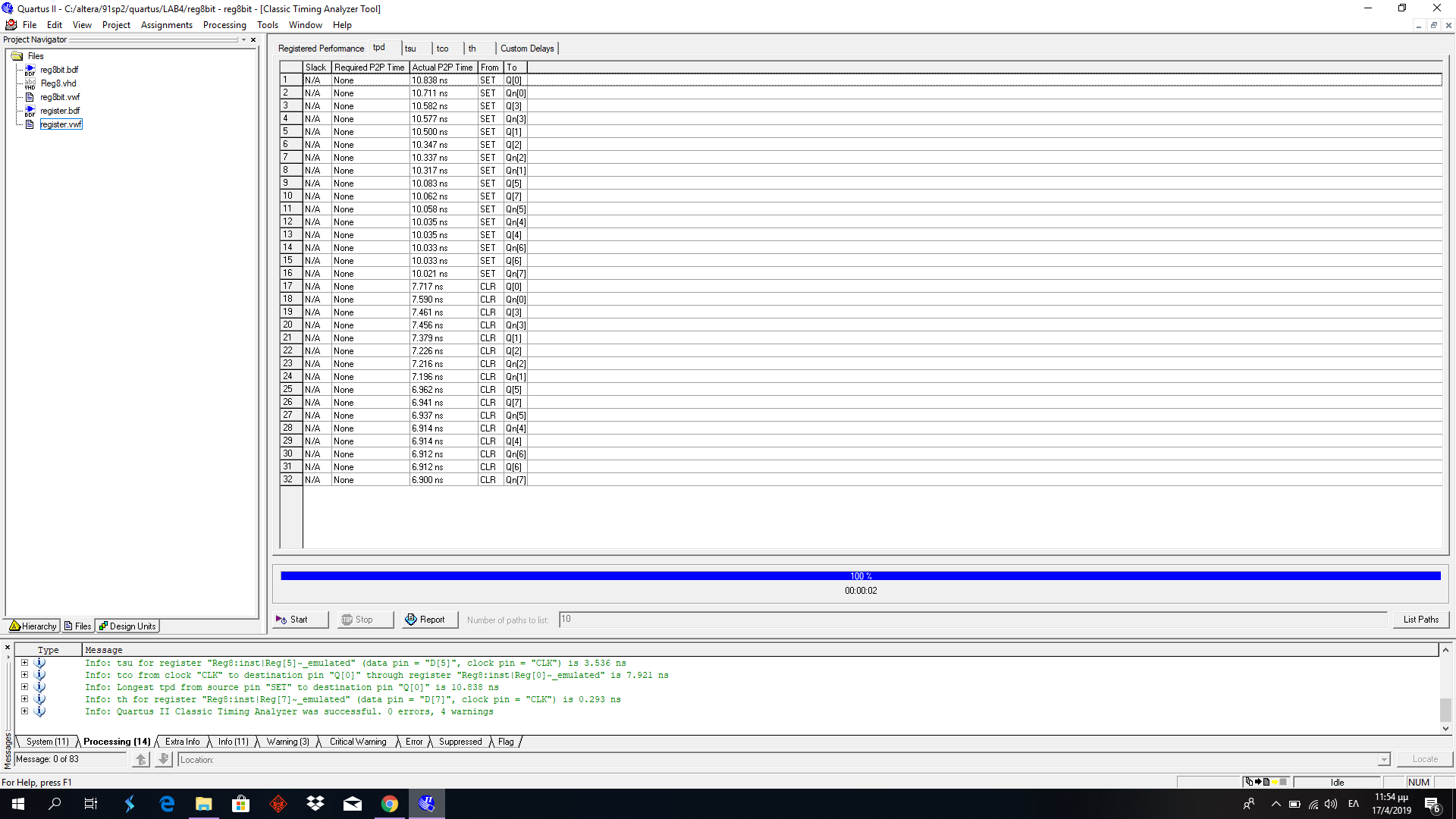
Το σχηματικό του παραπάνω κώδικα φαίνεται παρακάτω αφού έχει συνδεθεί και στο LCD Display:



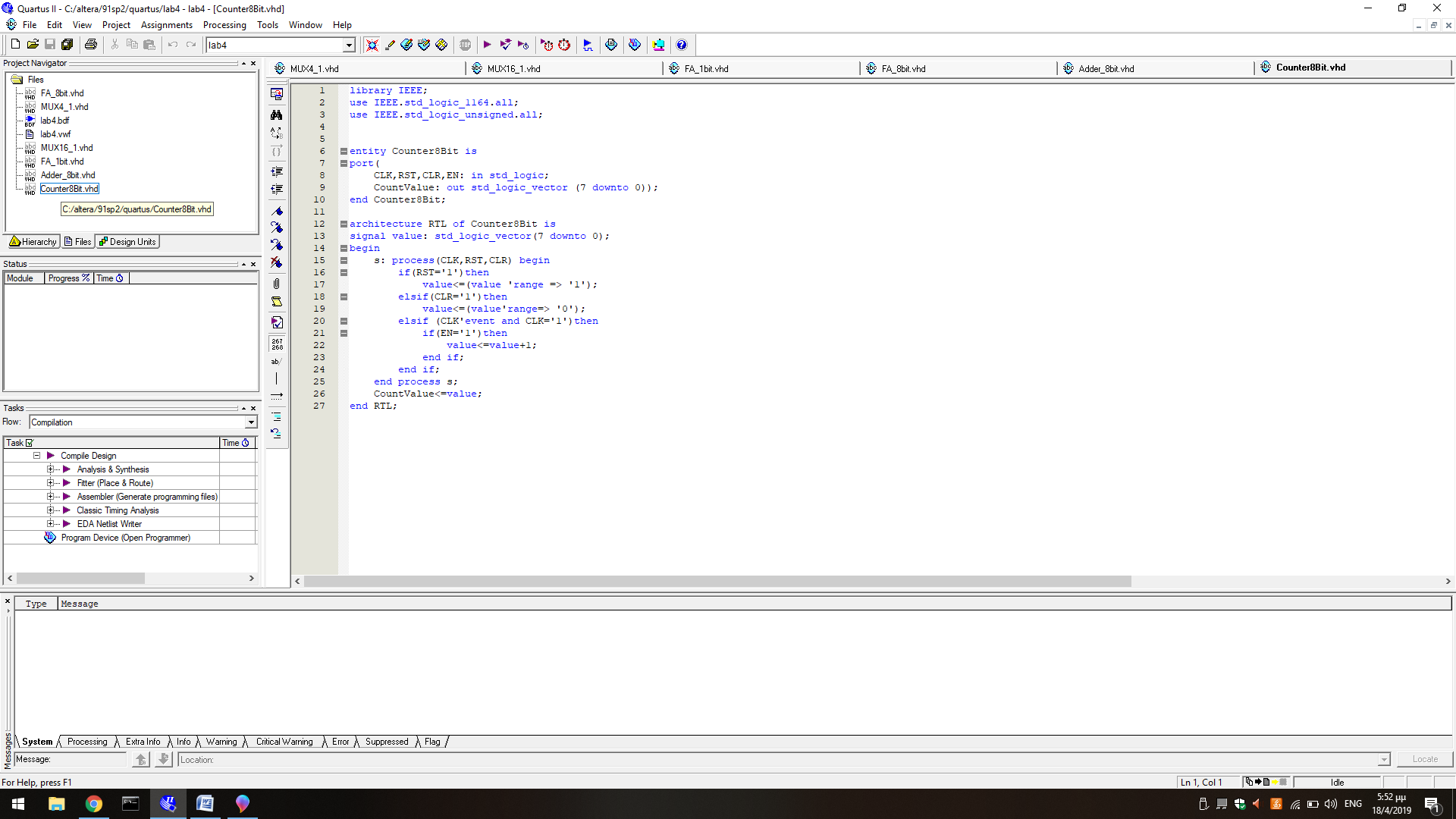
Η κυματομορφή πριν την σύνδεση του LCD Display φαίνεται παρακάτω:



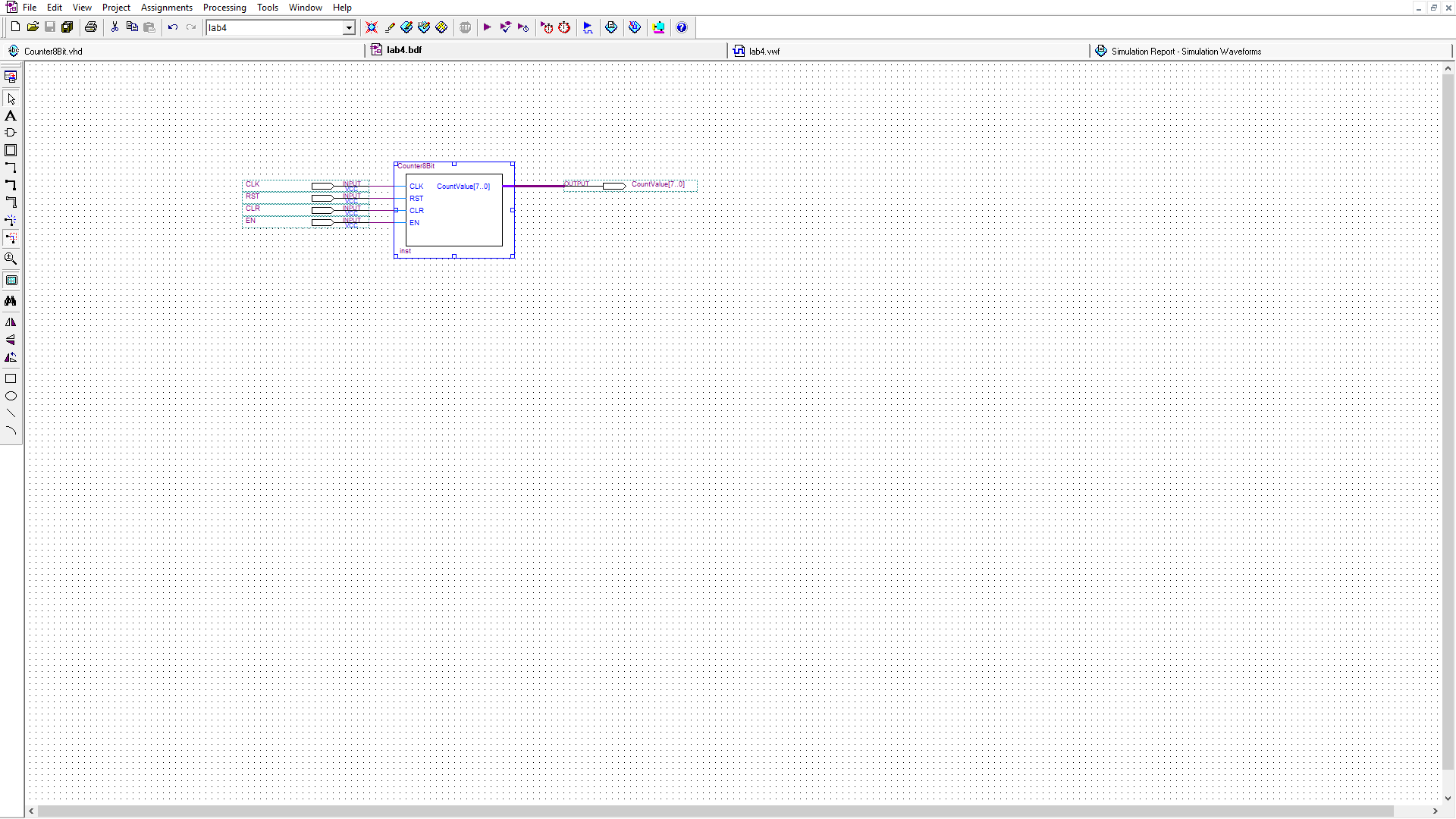
Η ελάχιστη συχνότητα του ρολογιού που μπορεί να χρησιμοποιηθεί στο συγκεκριμένο κύκλωμα είναι περίπου 13 ns :



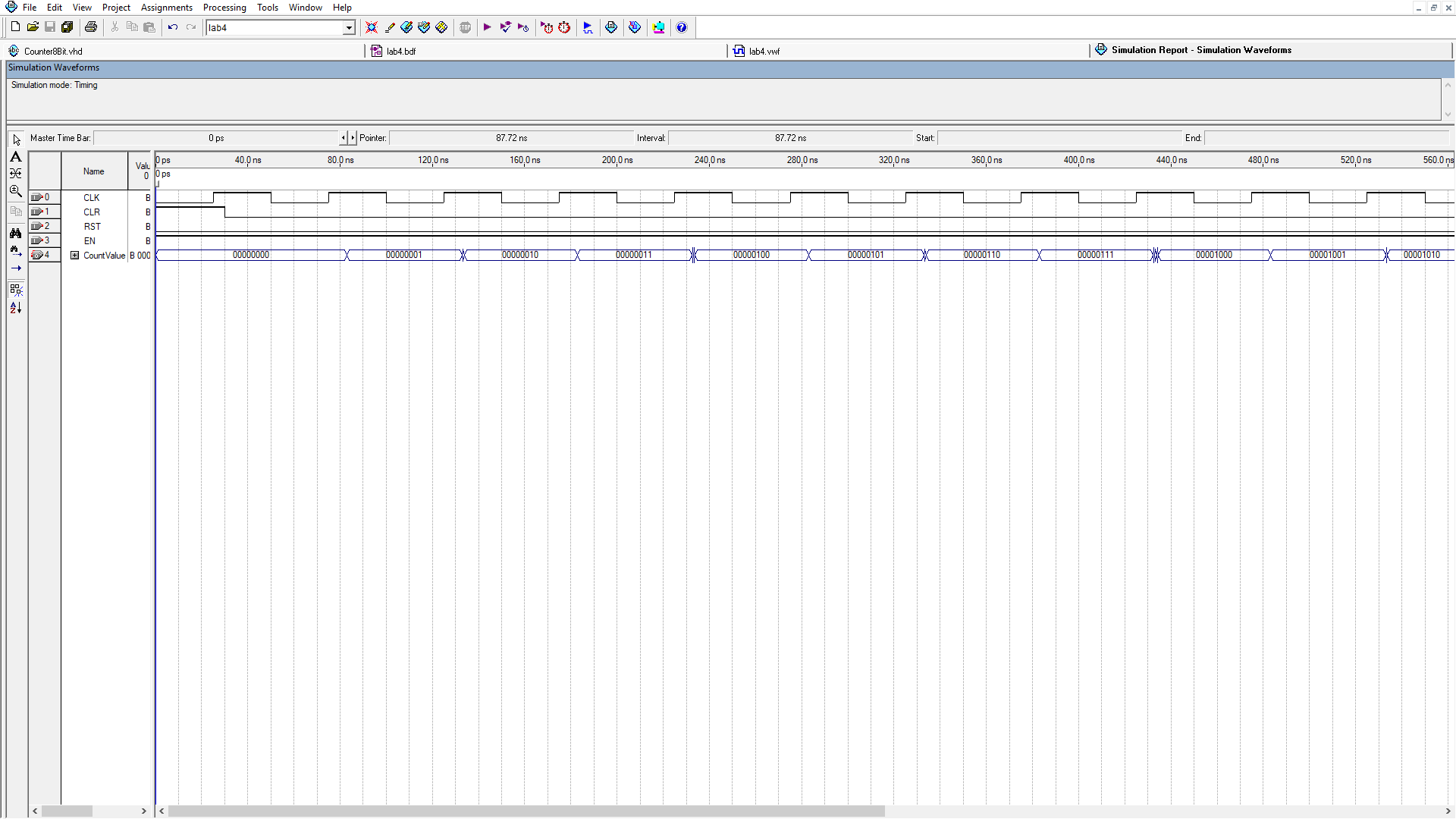
Τέλος δημιουργούμε έναν μετρητή των 8 bits και ο κώδικας σε VHDL φαίνεται παρακάτω:



Παρακάτω φαίνεται το σχηματικό του φτιάξαμε για να επιβεβαιώσουμε την λειτουργία του:



Στην κυματομορφή που ακολουθεί φαίνεται η σωστή λειτουργία του:



Στην στατική χρονική ανάλυση του κυκλώματος φαίνεται η μέγιστη ταχύτητά του η οποία είναι 7.935 ns από το RST μέχρι το CounterValue[6] :

