Ύλοποίηση Αλγορίθμου Κρυπτογράφησης Joltik σε VHDL

Σταυρινάκης Παναγιώτης **A.M:** 6217
Αϊβαλιώτης Βασίλειος **A.M:** 5987

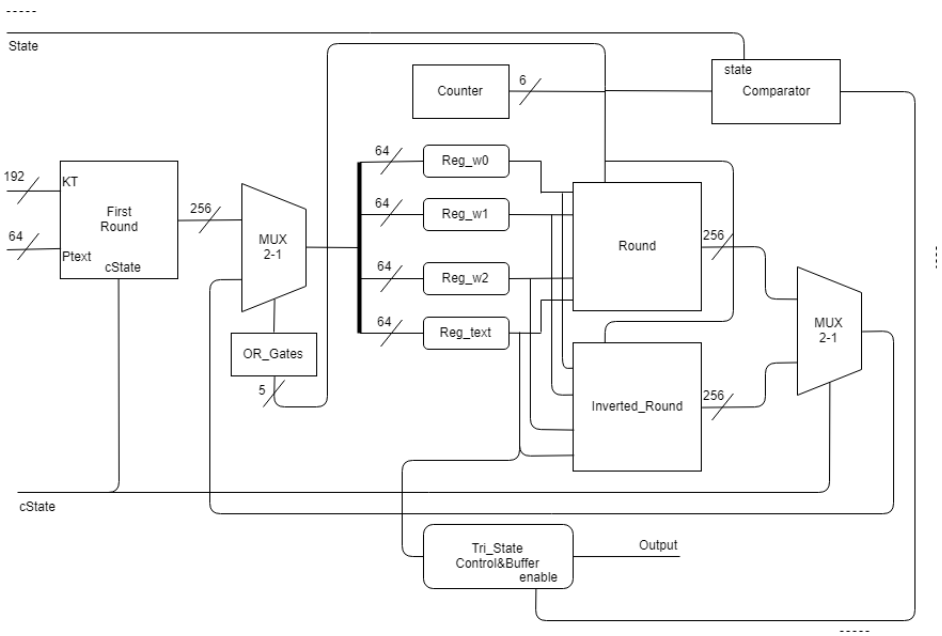
Ιούνιος 2018

Περιγραφή λειτουργίας συστήματος

Το σύστημα μας δέχεται 5 εισόδους και παράγει μία έξοδο. Οι είσοδοι είναι το keytweak, το κείμενο προς κρυπτογράφηση(plaintext), η κατάσταση λειτουργίας του Joltik(128 ή 192), η κατάσταση κρυπτογράφησης ή αποκρυπτογράφησης και το ρολόι. Έξοδος του συστήματος είναι το κρυπτογραφημένο κείμενο(cyphertext).

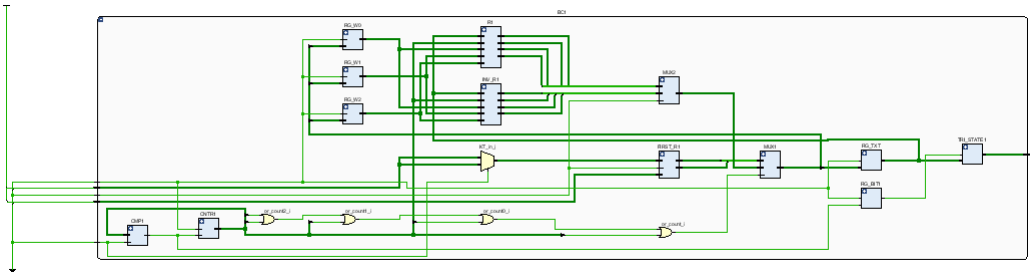
KeyTweak	128-192 bits
plaintext	64 bits
Κατάσταση λειτουργίας(state)	1 bit
Κατάσταση κρυπτογράφησης(cypherstate)	1 bit
Clock	1 bit
cyphertext	64 bits

Ο χρήστης εισάγει το κείμενο που επιθυμεί να μετατρέψει καθώς και το αναγκαίο κλειδί εισόδου. Επίσης επιλέγει εάν επιθυμεί να χρησιμοποιήσει το TWEAKEY framework για τον Joltik-BC-128 ή τον Joltik-BC-192 και αν πρόκειται για λειτουργία κρυπτογράφησης ή αποκρυπτογράφησης. Το κείμενο διαχωρίζεται σε ομάδες των 64 bits πριν εισαχθεί στην εφαρμογή μας(δεν έχει υλοποιηθεί) και μέσω επαναληπτικών διαδικασιών αρχιτεκτονικών που αναλύουμε παρακάτω εξάγεται το τελικό μέρος κειμένου.
Στο Σχήμα 1 φαίνεται μια απλουστευμένη μορφή της αρχιτεκτονικής που σχεδιάζουμε, απαραίτητη για να έχουμε εικόνα του τι επρόκειτο να υλοποιήσουμε.

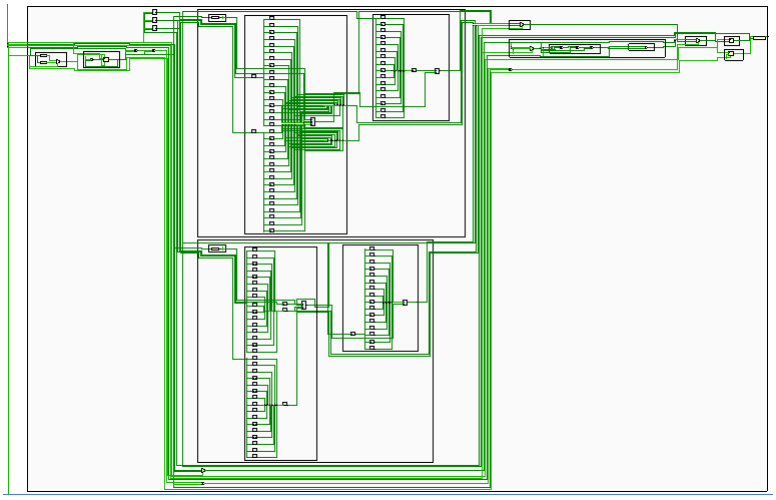


Σχήμα 1: Αρχική προτεινόμενη σχεδίαση

Στο Σχήμα 2 και Σχήμα 3 βλέπουμε τα αποτελέσματα σύνθεσης της τελικής αρχιτεκτονικής μας με χρήση του προγράμματος Vivado. Επεξηγώντας τις αρχιτεκτονικές μας, παρουσιάζουμε και τις αντίστοιχες συνθέσεις τους.



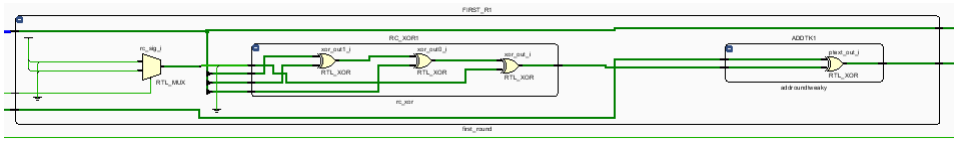
Σχήμα 2: Γενική τελική σύνθεση



Σχήμα 3: Αναλυτική τελική σύνθεση

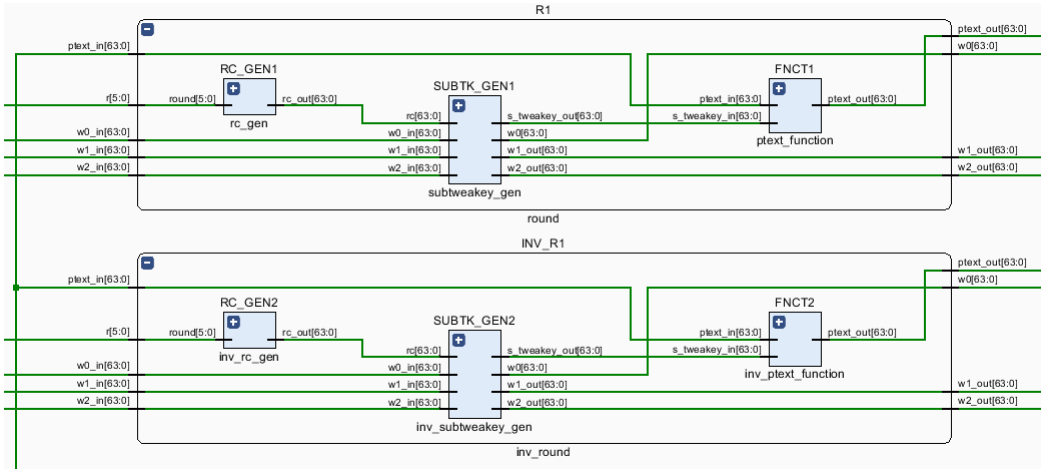
Αρχιτεκτονική συστήματος

Περιγράφουμε αρχικά την αρχιτεκτονική του First Round που αποτελεί το πρώτο κομμάτι της αρχιτεκτονικής μας. Εισάγεται το αρχικό κείμενο προς επεξεργασία, το κλειδί το οποίο διαχωρίζεται σε τρία τμήματα και η κατάσταση λειτουργίας. Γίνεται έλεγχος του RC που παράγεται μέσω της κατάστασης λειτουργίας, εκτελείται bitwise XOR των τμημάτων του κλειδιού και του RC και τέλος bitwise XOR με το κείμενο εισόδου. Στην έξοδο της αρχιτεκτονικής στέλνουμε το κλειδί και το κωδικοποιήμένο κείμενο.



Σχήμα 4: Σύνθεση Αρχιτεκτονικής FirstRound

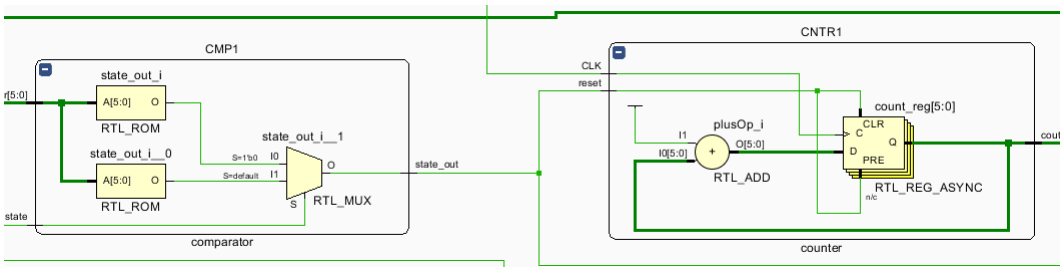
Διαθέτουμε έναν πολυπλέκτη 2 σε 1, ο οποίος επιλέγει τα δεδομένα που θα αποθηκευτούν στους καταχωρητές. Αν είμαστε στον πρώτο γύρο αποθηκεύονται τα δεδομένα του First Round αλλιώς αποθηκεύονται τα δεδομένα κάποιας από τις αρχιτεκτονικές κωδικοποίησης/αποκωδικοποίησης των επόμενων γύρων.
Διαθέτουμε 4 καταχωρητές των 64 bits για την αποθήκευση των δεδομένων.
Περιγράφουμε την αρχιτεκτονική των Round και Inverted_Round που αποτελούν κύριο κομμάτι της αρχιτεκτονικής μας. Δέχονται ως εισόδους τα δεδομένα απο τους καταχωρητές και τον ςουντερ και παράγουν νέα κωδικοποιημένα δεδομένα προς αποθήκευση.



Σχήμα 5: Σύνθεση Αρχιτεκτονικής Round & InvertedRound

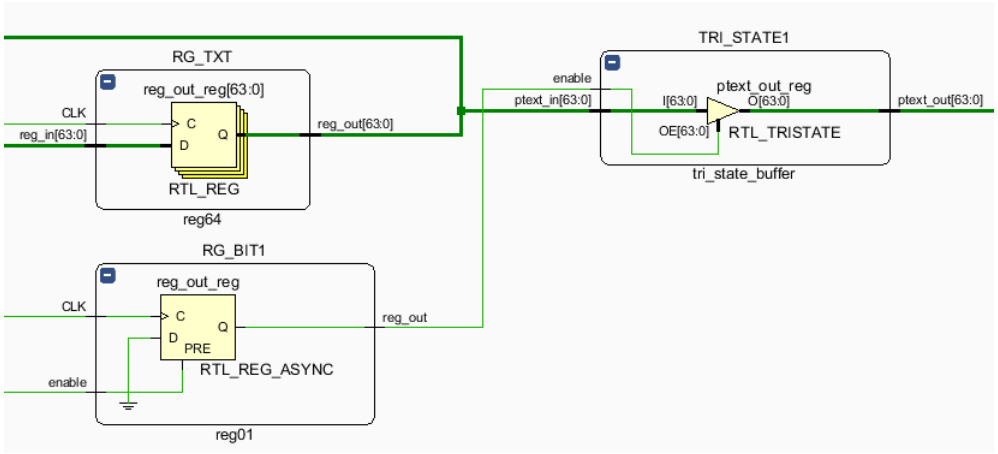
Μέσω του RC_GEN παράγεται το RC, το κλειδί τροποποιείται μέσω αντιμεταθέσεων και ολισθήσεων και εκτελείται bitwise XOR όπως παρουσιάσαμε και στο FirstRound. Το κείμενο δέχεται επίσης αντικαταστάσεις τιμών και ολισθήσεις ώστε με τη σειρά του να εκτελέσει bitwise XOR με το προηγούμενο σήμα και να παράγει το νέο κείμενο προς αποθήκευση. Τέλος, ένας πολυπλέκτης 2 σε 1 αποφασίζει αν στους καταχωρητές αποθηκεύεται το αποτέλεσμα της κρυπτογράφησης ή της αποκρυπτογράφησης(Round ή InvertedRound).

Η αρχιτεκτονική Comparator συγκρίνει τον αριθμό του γύρου που εκτελείται με τον επιθυμητό αριθμό γύρων αναλόγως την κατάσταση λειτουργίας.

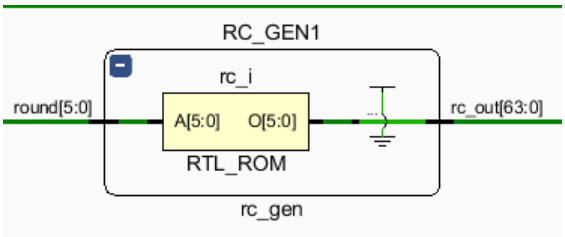


Σχήμα 6: Σύνθεση Αρχιτεκτονικής Comparator & Counter

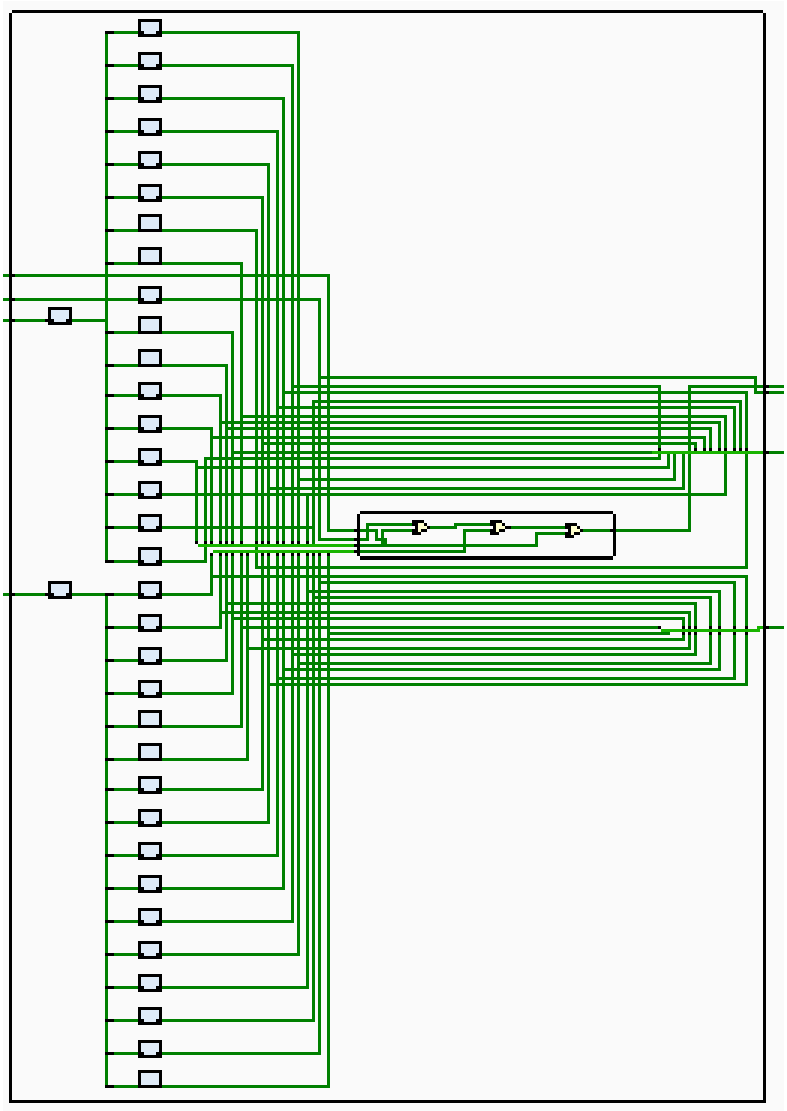
Τέλος, η αρχιτεκτονική TryStateControl&Buffer δέχεται ως είσοδο τα δεδομένα του καταχωρητή Reg_text και την έξοδο του Comparator. Όταν η έξοδος του Comparator γίνει 1 επιτρέπει την έξοδο των δεδομένων εισόδου για ένα κύκλο ρολογιού.



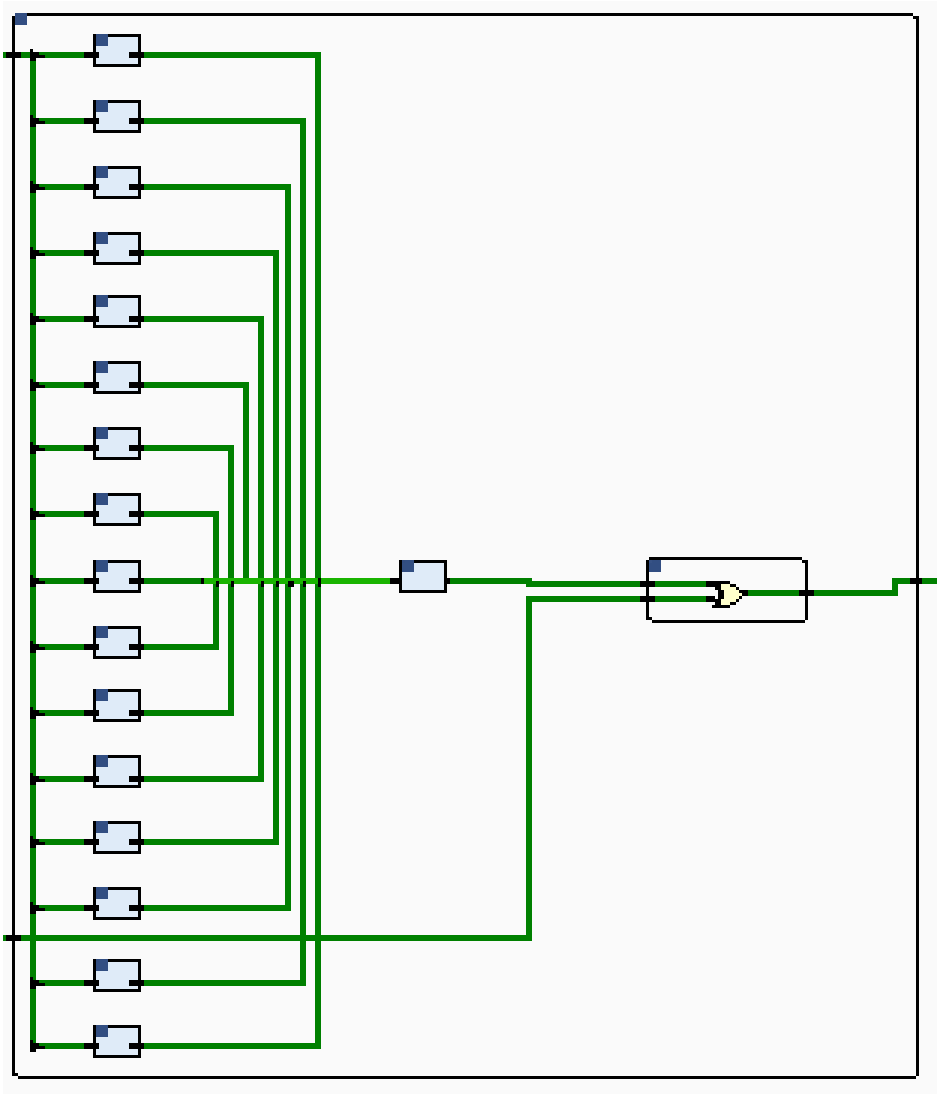
Σχήμα 7: Σύνθεση Αρχιτεκτονικής TryStateControl&Buffer



Σχήμα 8: Σύνθεση Αρχιτεκτονικής RC_Gen

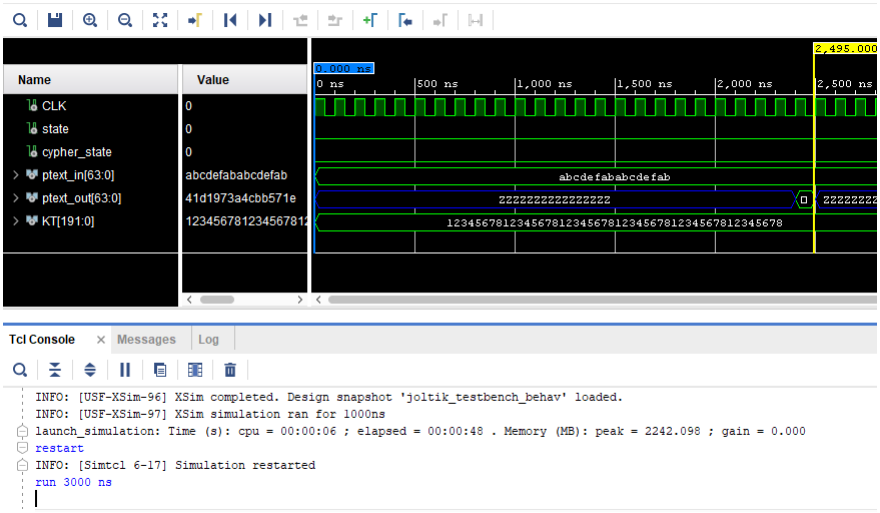


Σχήμα 9: Σύνθεση Αρχιτεκτονικής SUBTK_Gen

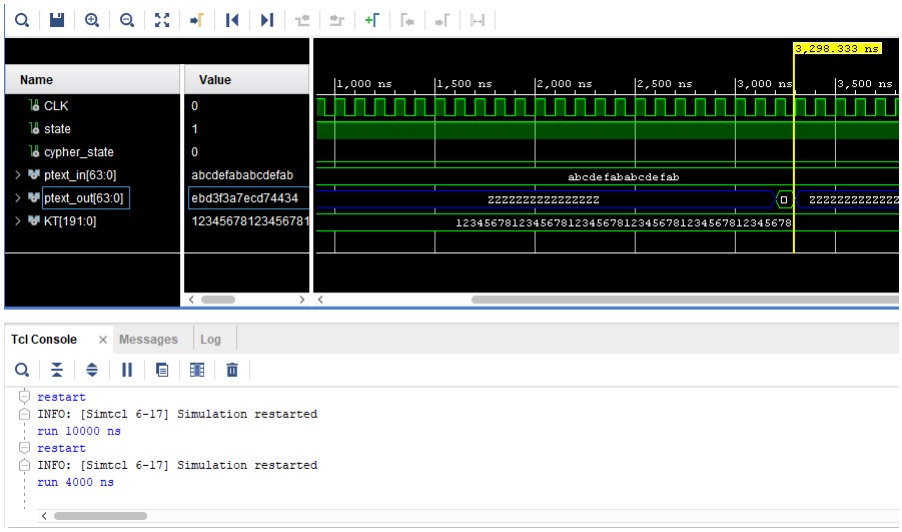


Σχήμα 10: Σύνθεση Αρχιτεκτονικής FNCT

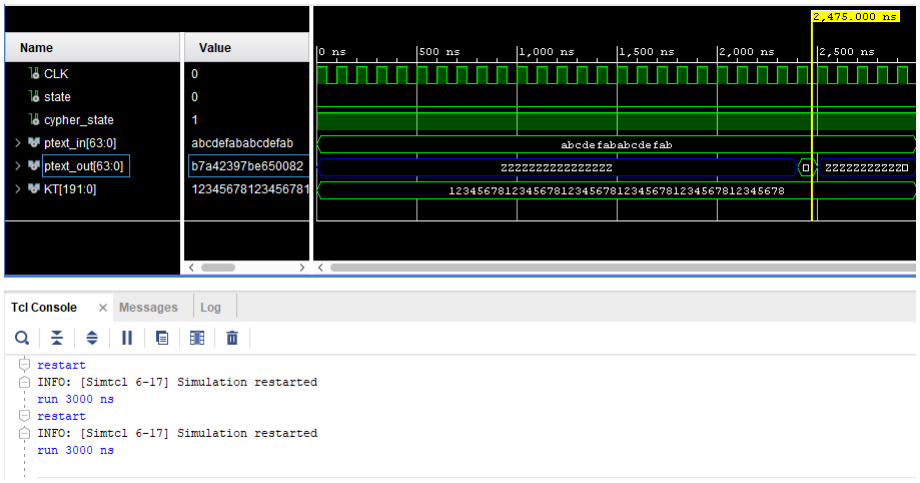
Αποτελέσματα προσομοίωσης



Σχήμα 11: Προσομοίωση Κρυπτογράφησης Joltik-BC-128



Σχήμα 12: Προσομοίωση Κρυπτογράφησης Joltik-BC-192

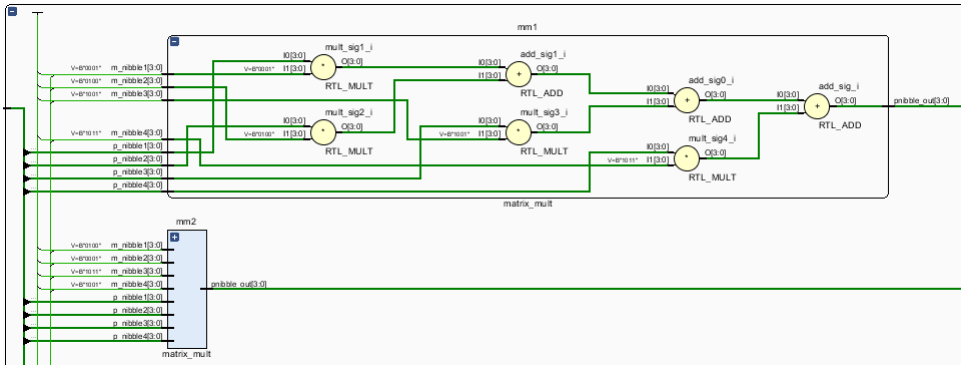


Σχήμα 13: Προσομοίωση Αποκρυπτογράφησης Joltik-BC-128

Παρατηρούμε πως ο χρόνος προσομοίωσης για τον Joltik-BC-128 είναι 24 κύκλοι + 1 κύκλος για την έξοδο, ενώ για τον Joltik-BC-192 είναι 32 κύκλοι + 1 κύκλος για την έξοδο

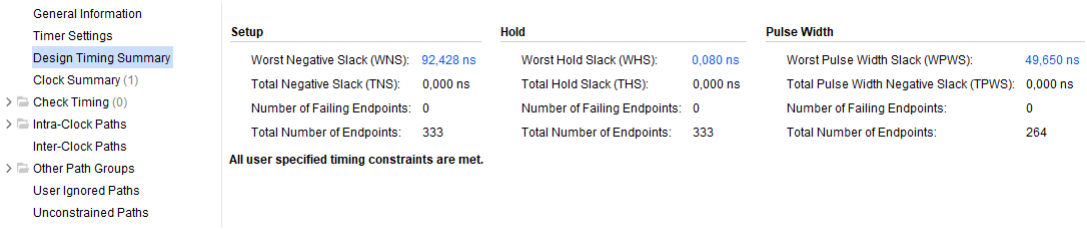
UPDATE

Επιχειρήθηκε υλοποίηση πολλαπλασιασμού μητρώων(MixNibbles) και προστέθηκε στο υλικό(εξακολουθεί να μην λειτουργεί σωστά η αποκρυπτογράφηση).

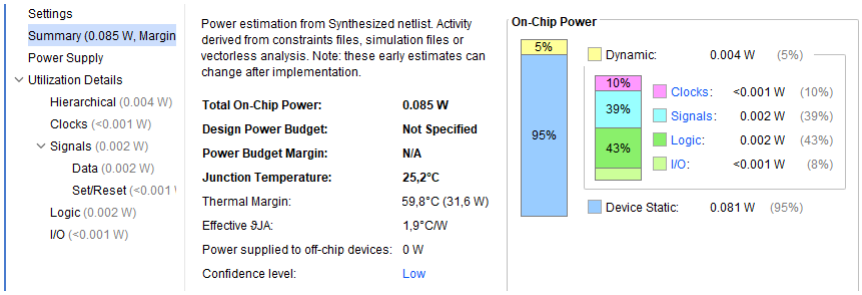


Σχήμα 14: Σύνθεση Αρχιτεκτονικής matrix_mult

Παρακάτω φαίνονται οι μετρήσεις της υλοποίησης για ρολόι περιόδου 100ns:



Σχήμα 15: Χρονισμός Κυκλώματος



Σχήμα 16: Μετρήσεις Ισχύος Κυκλώματος