# Χρήση **Watchdogs** για την αύξηση της αξιοπιστίας.

Γιαννόπουλος Νικόλαος 9629 , Μπουλιόπουλος Σταύρος Βασίλειος 9671
ngiannop@ece.auth.gr smpoulio@ece.auth.gr
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

***Abstract*—Στην παρούσα εργασία θα παρουσιάσουμε την σημαντικότητα των **watchdogs** καθώς θα δείξουμε την ανάγκη ύπαρξής τους. Επιπλέον, πως χρησιμοποιούνται σε εφαρμογές υψηλού επιπέδου όπως την επικοινωνία μέσω κινητής τηλεφωνίας στην συνέχεια πως ένας αλγόριθμος μηχανικής μάθησης μπορεί να δημιουργήσει για εμάς το λογικό κύκλωμα που θα παρακολουθεί έναν επεξεργαστή γενικής χρήσης **RISC,** ώστε να το αποτρέψει από τις «κακές» κατάστασης του και τέλος την χρήση ενός πολλαπλού **watchdog** κάνοντας την χρήση μιας κάρτας **FPGA** συνδεμένο με παράλληλη διεπαφή και το ενσωματωμένο σύστημα.**

**Keywords: CMOS, Watchdog, Multiple Watchdog, FPGA, RISC, TCP/IP, DTN**

## I. Τι είναι το WATCHDOG/Γιατί το χρειαζόμαστε/Οι εφαρμογές του

Το Watchdog είναι ένα είδους χρονομετρητή λογισμικής ή ηλεκτρονικής φύσης που χρησιμοποιείται για να ανιχνεύσει και να διορθώσει σφάλματα. Κατά την διάρκεια της κανονικής λειτουργίας του, το συνδεδεμένο σύστημα παρακολούθησης επανεκκινεί τακτικά τον χρονομετρητή. Χρειαζόμαστε τα watchdogs σε υπολογιστικές μονάδες για την αποτροπή διακοπής της λειτουργίας του συστήματος από σφάλμα υλικού ή λειτουργικού συστήματος. Ευρεία εφαρμογή των watchdogs στα ενσωματωμένα και λειτουργικά συστήματα πραγματικού χρόνου, όπου η ανθρώπινη παρέμβαση είναι ανέφικτη και τα συστήματα πρέπει να είναι αυτοδύναμα δηλαδή όπως σε διαστημικές αποστολές, όπου είναι απόμακρα πλέον τα συστήματα από τον άνθρωπο. Γι'αυτό τα watchdogs καλούνται να διαχειριστούν τις κρίσιμες χρονικές εργασίες και τις πιθανές αποτυχίες αυτών πριν διαδοθούν στο υπόλοιπο σύστημα εργασιών. Εάν, λόγω σφάλματος υλικού ή σφάλματος προγράμματος, ο υπολογιστής αποτύχει να επανεκκινήσει το Watchdog, τότε ο χρονοδιακόπτης θα παρέλθει και θα δημιουργήσει ένα σήμα χρονικού ορίου. Το σήμα χρονικού ορίου χρησιμοποιείται για την έναρξη διορθωτικών ενεργειών. Οι διορθωτικές ενέργειες περιλαμβάνουν συνήθως την τοποθέτηση του υπολογιστή και του σχετικού υλικού σε ασφαλή κατάσταση και την κλήση επανεκκίνησης του υπολογιστή. Η βασική αρχιτεκτονική ενός watchdog φαίνεται στην Εικόνα 1 .

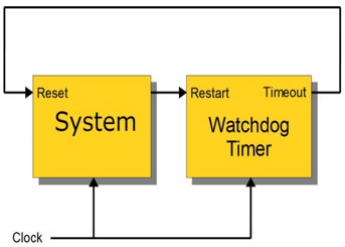


Figure 1: Αρχιτεκτονική ενός watchdog

## II. Ιστορική Αναδρομή και Εξέλιξη

Η τεχνολογία όσο εξελίσσονταν όλο και πιο πολύ τόσο αυξάνεται η ανάγκη ύπαρξης αξιοπιστίας στα κυκλώματα μας. Αυτό γίνεται με την χρήση των watchdogs, καθώς παρακολουθούν το κύριο σύστημα αν συμβεί κάποιο σφάλμα και πάει σε μια άγνωστη κατάσταση, όπου η λειτουργεία του κυκλώματος να επιφέρει άσχημα αποτελέσματα, όπως η καταστροφή του ιδίου του συστήματος, η παραγωγή λάθος αποτελεσμάτων. Γενικότερα, το κύκλωμα αυτό συγκριτικά με έναν απλό RISC CPU καταλαμβάνει πολύ μικρή επιφάνεια οπού μπορεί να «χωρέσει» σε οποιαδήποτε εφαρμογή με τεχνικές όπως 3D Integrated Circuit, οπού κάνουμε stacking των επιμέρους κυκλωμάτων και μπορεί να βρίσκεται σε παλιότερες τεχνολογίας και όχι στις πιο πρόσφατες (πχ 180 nm) και να λειτουργεί πολύ αξιοπιστία και με μικρή κατανάλωση ενέργειας.

## III. Μελέτη σκοπιμότητας των WATCHDOGS σε ευκαιριακά Κινητά Δίκτυα

Τα πρωτόκολλα TCP/IP προϋποθέτουν την ύπαρξη μιας από άκρο σε άκρο διαδρομής μεταξύ μιας πηγής και ενός προορισμού κόμβου σε μια επικοινωνία, και να εγγυάται την παράδοση των πακέτων. Επιπλέον, ορισμένοι μηχανισμοί, όπως οι χρονομετρητές αναμετάδοσης, χρησιμοποιούνται για τη διατήρηση της αξιοπιστίας στην επικοινωνία TCP/IP. Ωστόσο, ορισμένα περιβάλλοντα ασύρματων δικτύων μπορεί να μην υποθέτουν αυτές τις προϋποθέσεις λόγω των προτύπων κινητικότητας, των συχνών αποσυνδέσεων και των περιορισμένων πόρων των κόμβων. Σε αυτό το πλαίσιο, ο ευκαιριακός Mobile Networks (OMNs) στα οποία οι κόμβοι λειτουργούν λαμβάνοντας υπόψη τις μεγάλες καθυστερήσεις και τις διακοπές, όπως τα Δίκτυα με ανοχή στην καθυστέρηση (Delay Tolerant Networks, DTN). Οι αρθρογράφοι [1] πρότειναν την προσθήκη ενός στρώματος δέσμης για να καταστεί δυνατή η επικοινωνία σε λειτουργία DTN, κάθε κόμβος εφαρμόζει το παράδειγμα αποθήκευσης, μεταφοράς και προώθησης. Έτσι, ένα κόμβος μπορεί να μεταφέρει ένα μήνυμα μέχρι να προκύψει μια νέα επαφή, όταν προωθεί το μήνυμα, και ούτω καθεξής μέχρι το μήνυμα να φτάσει στον κόμβο προορισμού. Η επικοινωνία γίνεται χωρίς ιεραρχία, δηλαδή χωρίς υποδομή, εφόσον

οι κόμβοι συνεργάζονται σε προώθηση των μηνυμάτων. Ωστόσο, οι περιορισμένοι πόροι των κινητών κόμβων, όπως ο αποθηκευτικός χώρος ή η ενέργεια της συσκευής, μπορεί να λήξουν γρηγορότερα. Όταν συμβαίνει αυτό, οι κόμβοι μπορεί να ενεργούν ως εγωιστές, οπότε η επικοινωνία των κόμβων μπορεί να επηρεαστεί. Ορισμένα μοντέλα για την ανίχνευση και τον μετριασμό της εγωιστικής συμπεριφοράς ( selfish node). Αυτές οι μέθοδοι, γνωστές ως watchdogs, προϋποθέτουν την ύπαρξη ενός μηχανισμού για την ανίχνευση του πότε ένας κόμβος ενεργεί εγωιστικά. Τα watchdogs προωθούν πληροφορίες σε έναν άλλο μηχανισμό για να αποφασίσει για την προώθηση του μηνύματος ή όχι. Ωστόσο, υπάρχει έλλειψη μελετών σχετικά με τη σκοπιμότητα των watchdogs στα δίκτυα. Επιπλέον, μια άλλη κατηγορία μελετών χρησιμοποιεί πιθανοτικά watchdogs, δηλαδή όταν τα watchdogs έχουν μια αριθμητική τιμή για τον καθορισμό των αποφάσεών τους. Από αυτές τις ελλείψεις μελετών συνεπάγεται ότι υπάρχει έλλειψη παρατηρήσεων σχετικά με τον αντίκτυπο ή των συνεπειών της στα AMN. Κύριος στόχος της μελέτης είναι η εφαρμογή ενός watchdog χρησιμοποιώντας έναν προσομοιωτή ONE , όπου επιλέγεται ένα ποσοστό κόμβων να εμφανίζουν εγωιστική συμπεριφορά και αξιολογούνται όλοι οι τύποι απόρριψης εκτός των εγωιστικών κόμβων. Ο κύριος στόχος ενός watchdog είναι η παρακολούθηση μηνυμάτων αναμετάδοσης στο δίκτυο. Έτσι, το ορίζουν ως απομονωτή με μήνυμα που αποστέλλεται. Το watchdog συγκρίνει κάθε πακέτο που προωθείται στο δίκτυο με τα περιεχόμενα του απομονωτή του. Εάν η σύγκριση ταιριάζει, το περιεχόμενο του watchdog αφαιρείται. Διαφορετικά, μετά τη λήξη ενός χρονοδιακόπτη, ο κόμβος χαρακτηρίζεται ως εγωιστής επειδή δεν προώθησε το μήνυμα. Έτσι, κατά τη διάρκεια αυτού του έργου τέθηκαν ορισμένα ερωτήματα. Μεταξύ αυτών, επισημαίνουμε:

- Πώς να δημιουργηθεί ένας μηχανισμός που δεν θα περιέχει παρεμβολές, δηλαδή, από πιθανή χειραγώγηση των κόμβων. Εάν ένας κόμβος δεν
προωθήσει ένα μήνυμα (για ιδιοτελή λόγο, για παράδειγμα),τότε αυτός ο κόμβος δεν μπορεί να επιβάλει μια εγγραφή προώθησης μηνύματος σε αυτόν τον απομονωτή,

- Πώς να εξετάσουμε την απόρριψη προώθησης για εγωιστικούς λόγους στο σχεδιασμό του watchdog. Για παράδειγμα, ας υποθέσουμε ότι αν ο κόμβος A αναμεταδίδει ένα μήνυμα για τον κόμβο B, ωστόσο ο B απορρίπτει το μήνυμα λόγω έλλειψης επαρκούς χώρου στο ρυθμιστικό διάδρομο. Ως θέμα πρακτικότητας, τα OMN δεν υλοποιούν την επιβεβαίωση αναμετάδοσης. Έτσι, ο κόμβος A εμπιστεύεται ότι ο B δέχεται να μεταφέρει το μήνυμα,

- Πώς να κάνουμε τη διαδικασία των παρατηρητών να αντικατοπτρίζει την πραγματικότητα: των λειτουργιών του δικτύου σε κάθε κόμβο. Έτσι, συνειδητοποιούμε ότι η διαδικασία παρακολούθησης πρέπει να λειτουργεί στο επίπεδο μεταφοράς και μπορούμε να την ενσωματώσουμε με τη δρομολόγηση ή τη διαχείριση ρυθμιστικού διαύλου.

Στο μηχανισμό μας, κάθε κόμβος διαθέτει έναν απομονωτή προωθούμενων μηνυμάτων, που ονομάζεται watchdog. Έτσι, το δικό μας watchdog είναι ένας απομονωτής που περιέχει την ακόλουθη πλειάδα πληροφοριών (timestamp, messageId, from, to). Με αποτέλεσμα, κάθε γραμμή στο watchdog λειτουργεί επίσης σαν κατακερματισμός για να είναι εύκολη η αναζήτηση και η σύγκριση τιμών.

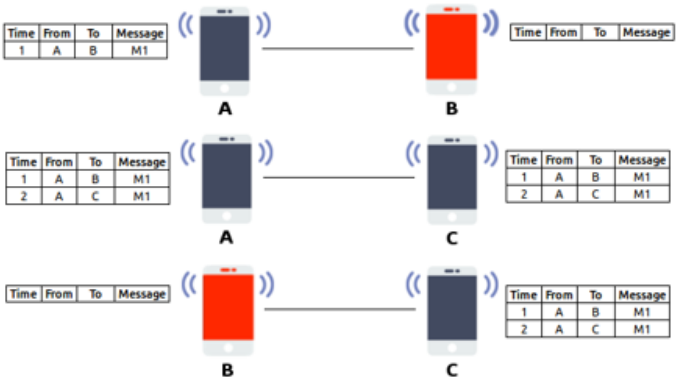


Figure 2: Εφαμοργή watchdog σε mobile network

Στο Εικόνα 2 , παρουσιάζουμε ένα παράδειγμα για το πώς το watchdog μας λειτουργεί. Κατά τη χρονική στιγμή 1, ο κόμβος A συναντά τον κόμβο B και προσπαθεί να στείλει το μήνυμα M1. Ωστόσο, ο B δεν προσθέτει το μήνυμα στον απομονωτή δεδομένων του (σε αυτό το σημείο, ο λόγος δεν είναι σημαντικός λόγω του γεγονότος ότι ο A δεν θα γνωρίζει τον λόγο). Κατά τη χρονική στιγμή 2, Ο A συναντά τον κόμβο Γ, οπότε, αναμεταδίδει επίσης το μήνυμα M1 και και οι δύο κόμβοι μοιράζονται τα watchdogs τους. Τώρα ο Γ γνωρίζει επίσης για τον το μήνυμα M1 που αναμεταδόθηκε από τον A στον B. Τέλος, ο B συναντά τον κόμβο Γ, οπότε, το watchdog του κόμβου Γ μπορεί να ανιχνεύσει τον B με ασυνήθιστο συμπεριφορά. Τέλος, υποθέτουμε επίσης ότι κάθε κόμβος του δικτύου έχει αρκετό χώρο απομόνωσης για τη χρήση του watchdog. Αργότερα, μελετήσαμε επίσης τη συμπεριφορά του watchdog όταν εφαρμόζεται έλεγχος συμφόρησης.
Στην πραγματικότητα, ο έλεγχος συμφόρησης μπορεί να εφαρμοστεί για την αποφυγή υψηλού αριθμού μηνυμάτων watchdog που αναμεταδίδονται. Ο έλεγχος συμφόρησης δεν μελετάται σε βάθος εδώ, αλλά μπορούμε να αναφέρουμε ότι ορισμένες ευρετικές μέθοδοι μπορούν να εφαρμοστούν, όπως η FIFO ή άλλα κοινωνικά χαρακτηριστικά για να αφαίρεση των πληροφοριών των κόμβων που η πιθανότητα συνάντησης είναι χαμηλή. Έτσι, αξιολογούμε το watchdog χωρίς συμφόρηση σύστημα και με κατώφλι 6 ωρών, δηλαδή

μετά από 6 ώρες, η καταχώρηση στο watchdog εξαλείφεται. Ο λόγος γι' αυτό είναι ο χαμηλός χρόνος προσομοίωσης των ιχνών και η έλλειψη του συστήματος συμφόρησης watchdog ελέγχου για την επιλογή μιας τιμής για το σημείο αναφοράς. Το ποσοστό απόρριψης μηνυμάτων λόγω εγωισμού ορίζεται ως εξής ως το ποσοστό μεταξύ της ποσότητας των απορριπτόμενων μηνυμάτων από εγωιστές κόμβους και του συνόλου των απορριφθέντων μηνυμάτων. Αυτή η ανάλυση αποσκοπεί στην κατανόηση του τρόπου με τον οποίο η ποσότητα των απορριπτόμενων μηνυμάτων μπορεί να επηρεάσει την απόδοση του watchdog. Στην Εικόνα 3 παρουσιάζεται η πιθανότητα απόρριψης μεταφοράς μηνυμάτων λόγω εγωιστικών κινήτρων. Μπορούμε να δούμε ότι όταν μειώσουμε το ποσοστό των εγωιστικών κόμβων στο δίκτυο, η απόρριψη ποσοστού των μηνυμάτων λόγω εγωισμού ποικίλλει και στα δύο ίχνη που χρησιμοποιήθηκαν.

Ειδικά στο ίχνος Infocom05, το ποσοστό απόρριψης των μηνυμάτων λόγω εγωισμού είναι σχεδόν 30% όταν έχουμε 10% εγωιστές κόμβους και περίπου 50% στο ίχνος Sassy. Μπορούμε να δούμε στην κατάσταση όπου έχουμε λίγους εγωιστές κόμβους, το ποσοστό απόρριψης των μηνυμάτων λόγω εγωισμού που αυξάνεται για άλλους λόγους όπως η έλλειψη επαρκούς χώρου προσωρινής αποθήκευσης ή το απασχολημένο ασύρματο κανάλι. Αυτό θα μπορούσε να οδηγήσει σε ενδεχόμενα σφάλματα στις ανιχνεύσεις του watchdog.

Επιπλέον, τονίζουμε ότι ο αριθμός των ταυτόχρονων μηνυμάτων που χρησιμοποιήθηκε σε αυτό το πείραμα ήταν 1. Καταλήγουμε ότι οι τιμές που ορίζονται ως το ποσοστό μεταξύ των εγωιστικών κόμβων και του ποσοστού απόρριψης των μηνυμάτων λόγω εγωισμού. Τα αποτελέσματά μας δείχνουν ότι υπάρχει συσχέτιση περίπου 0,62 για το σενάριο Sassy και περίπου 0,87 για το σενάριο Infocom05. Ο κύριος λόγος για τη μεγαλύτερη συσχέτιση για το σενάριο Infocom05 είναι το υψηλό ποσοστό επαφών,δηλαδή όταν αυξάνεται ο αριθμός των εγωιστικών κόμβων, τότε αυτό επηρεάζει περισσότερο από ό,τι όταν ο ρυθμός των επαφών είναι λίγο χαμηλός.

Τέλος, μπορούμε να συμπεράνουμε ότι και στις δύο περιπτώσεις, υπάρχει γραμμική συσχέτιση μεταξύ του αριθμού των εγωιστικών κόμβων και του ποσοστού απόρριψης των μηνυμάτων, ακόμη και όταν το ποσοστό επαφών μεταξύ των κόμβων είναι χαμηλός.

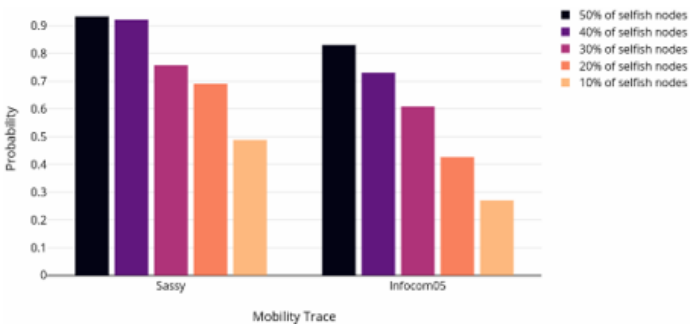


Figure 3: Η Πιθανότητα απόρριψης μεταφοράς μηνυμάτων

## IV. Σχεδιασμός κυκλώματος WATCHDOG με χρήση δέντρων αποφάσεων για την ανίχνευση σφαλμάτων στον επεξεργαστή

Οι εξελίξεις στις εμπορικές διαστημικές εφαρμογές έχουν την υιοθέτηση συσκευών FPGA με μαλακό πυρήνα σε δορυφορικά συστήματα. Οι επεξεργαστές είναι ζωτικής σημασίας σε όλα τα δορυφορικά ηλεκτρονικά συστήματα που χρησιμοποιούνται για τον έλεγχο λειτουργίες και την επεξεργασία δεδομένων επί του σκάφους. Η εχθρότητα του διαστημικού περιβάλλοντος, η κλιμάκωση του μεγέθους των τρανζίστορ, η υψηλή συχνότητα λειτουργίας και τα χαμηλά επίπεδα τάσης αυξάνουν δραματικά την εμφάνιση των παροδικών βλαβών που μπορούν να επηρεάσουν τη λειτουργία των υπολογιστικών συστημάτων στα διαστημόπλοια.

Ένα soft-σφάλμα συμβαίνει όταν ένα συμβάν ακτινοβολίας είναι αρκετά ισχυρό ώστε να αναποδογυρίσει ένα ή περισσότερα bits ενός καταχωρητή χωρίς να προκληθεί μόνιμη βλάβη στον επεξεργαστή. Τα soft errors που προκαλούνται από ακτινοβολία μπορούν να διακριθούν και να κατηγοριοποιηθούν σε γενικές γραμμές στους εξής δύο τύπους

- (i) Σιωπηλή καταστροφή δεδομένων (SDC) και
- (ii) Λειτουργική διακοπή (FI).

Ένα SDC συμβαίνει όταν η εφαρμογή εκτελείται κανονικά αλλά η τελική της έξοδος διαφέρει από την αναμενόμενη έξοδο. Οι λειτουργικές διακοπές είναι απροσδόκητοι τερματισμοί της εφαρμογής. Ευαισθησία στα μαλακά σφάλματα είναι πιο έντονη στους κόμβους νεότερης τεχνολογίας και στις FPGA που βασίζονται σε SRAM. Διάφορα συστήματα ανοχής σφαλμάτων τεχνικές για τον μετριασμό των SEU σε ενσωματωμένους επεξεργαστές.

Τεχνικές όπως την τεχνητή νοημοσύνη (AI) και την μηχανική μάθηση (ML) αρχίζουν να βρίσκουν εκτεταμένη χρήση στην πολύ μεγάλης κλίμακας ολοκλήρωσης (VLSI) για δοκιμές αξιοπιστίας, εκτός από τις χρήσεις στον σχεδιασμό, την ανάλυση και την αυτοματοποίηση.

Ένας γενικός επεξεργαστής RISC έχει τις εντολές Fetch, Decode, Execute, Mem, και Write/Read. Σε αυτό το άρθρο, ένας ο επεξεργαστής SPARC-v8 χρησιμοποιείται ως μελέτη περίπτωσης. Ο πυρήνας του επεξεργαστή περιέχει ένα σύνολο καταχωρητών (καταχωρητές κατάστασης) που καθορίζουν τη ροή εκτέλεσης των εντολών μέσω του επεξεργαστή. Αυτό το σύνολο καταχωρητών κατάστασης του επεξεργαστή περιέχει καταχωρητές μετρητών προγράμματος, καταχωρητές εντολών, καταχωρητές μηχανής πεπερασμένων καταστάσεων (FSM), καταχωρητές pipeline, καταχωρητές αριθμητικών πράξεων, καταχωρητές σημαιών και άλλους καταχωρητές ελέγχου, όπως οι καταχωρητές παγίδευσης και οι καταχωρητές παραθύρου.

Στο εκτός από αυτό το σύνολο, υπάρχει επίσης ένα αρχείο καταχωρητών γενικής χρήσης και μνήμη κρυφής μνήμης. Ένα μοντέλο σφάλματος επεξεργαστή προέκυψε με τη χρήση SEU injection μόνο στους καταχωρητές κατάστασης

προκειμένου να παρατηρηθεί και να μελετηθεί η επίδραση των SEUs κατά τη διάρκεια προσομοιώσεων σφαλμάτων. Η κατάσταση του επεξεργαστή ορίζεται ως η συλλογή των τιμών στους καταχωρητές κατάστασης τη στιγμή κάθε δεδομένη στιγμή.
Μια καλή κατάσταση είναι ένας συνδυασμός τιμών που υπάρχει όταν ο επεξεργαστής εκτελείται κανονικά χωρίς καμία SEU. Μια ελαττωματική κατάσταση είναι ένας συνδυασμός τιμών που παράγεται από μία ή περισσότερες διαταραχές/αλλαγές bit στους καταχωρητές κατάστασης ως αποτέλεσμα μιας SEU. Οι ακόλουθες πέντε κατηγορίες σφαλμάτων προσδιορίστηκαν για να μελετηθεί ο αντίκτυπος μιας SEU στην κατάσταση του επεξεργαστή:

- Σφάλματα του μετρητή προγράμματος: SEU σε οποιονδήποτε από τους μετρητές προγράμματος (PC).
- Σφάλματα μητρώου εντολών: SEU σε οποιονδήποτε από τους καταχωρητών που αποθηκεύουν την εντολή.
- Σφάλματα μητρώου αριθμητικής λογικής μονάδας (ALU): SEU στους καταχωρητές ALU του σταδίου εκτέλεσης.
- Σφάλματα μητρώου ελέγχου επεξεργαστή: SEU στους καταχωρητές ελέγχου καταχωρητές του επεξεργαστή, όπως ένας δείκτης Window καταχωρητής, ένας καταχωρητής κωδικού κατάστασης, ή ένας καταχωρητής εποπτικού Setting register.
- Σφάλματα του μητρώου ελέγχου αγωγού (Pipeline Control Register Faults): SEU στους καταχωρητές που σχετίζονται με τη λειτουργία ελέγχου του αγωγού, όπως οι καταχωρητής που ελέγχει τη ροή των εντολών στο pipeline για την καθυστέρηση για λειτουργίες πολλαπλών κύκλων, ή για μηδενισμό της εντολής.

Μια SEU σε οποιονδήποτε από αυτούς τους καταχωρητές κατάστασης σε μια κρίσιμη στιγμή του μπορεί να επηρεάσει τη λειτουργία του συστήματος επεξεργαστή. Για το παραδείγματος, μια SEU στον καταχωρητή εντολής κατά τη διάρκεια της αποκωδικοποίησης μπορεί να οδηγήσει στην εκτέλεση μιας λανθασμένης εντολής. Ακόμη και αν και η ροή εντολών μπορεί να συνεχιστεί, αυτό θα έχει ως αποτέλεσμα ένα κίνδυνο αθόρυβης αλλοίωσης δεδομένων (SDC) και, συνεπώς, θα αλλοιωθεί η τελική εξόδου. Ομοίως, εάν η SEU τροποποιήσει τα περιεχόμενα του καταχωρητή εντολών για να δημιουργήσει μια παράνομη εντολή, θα προκαλέσει εξαίρεση και θα οδηγήσει σε λειτουργική διακοπή (FI) του εφαρμογής. Μια SEU στον καταχωρητή Program Counter κατά τη διάρκεια του σταδίου Fetch μπορεί να έχει ως αποτέλεσμα την εμφάνιση λανθασμένης εντολής η οποία θα προκαλέσει ένα SDC. Ομοίως, εάν ο SEU τροποποιεί την τιμή του μετρητή προγράμματος σε μια τιμή εκτός ορίων διεύθυνση, τότε αυτό μπορεί να οδηγήσει σε FI.

Τα δεδομένα εκπαίδευσης για το μοντέλο ML συλλέχθηκαν από Προσομοιώσεις Register Transfer Level (RTL) του επεξεργαστή που εκτελούν τυπικούς φόρτους εργασίας. Στην Εικονα 4 παρουσιάζεται η προσομοίωση σφάλματος που χρησιμοποιήθηκε για τη δημιουργία των δεδομένων εκπαίδευσης.
Η διάταξη αποτελείται από το μοντέλο RTL του επεξεργαστή, ένα μοντέλο εξωτερικής μνήμης για την αποθήκευση το πρόγραμμα εφαρμογής, και μια διάταξη εγχυτήρα σφαλμάτων για την έγχυση σφάλματα στους καταχωρητές κατάστασης του επεξεργαστή. Πριν από την έναρξη της προσομοίωσης, το λογισμικό της εφαρμογής φορτώθηκε στην εξωτερική μνήμη και στη συνέχεια διαβάστηκε στο μοντέλο του επεξεργαστή κατά τη διάρκεια της προσομοίωσης. Ο αριθμός και οι τύποι των σφαλμάτων ορίστηκαν στη διαμόρφωση του εγχυτήρα σφαλμάτων.
Χρήση του αυτή η διαμόρφωση, το σενάριο του εγχυτήρα σφαλμάτων παρήγαγε λογική προσομοωτή για την αναστροφή των αντίστοιχων bits του καταχωρητή σε τυχαίες χρονικές στιγμές κατά τη διάρκεια της λειτουργικής προσομοίωσης για να μιμηθούν σφάλματα SEU. Όταν ένα σωματίδιο ακτινοβολίας προσκρούει σε ένα bit καταχωρητή, ο αντίκτυπος της SEU στον εν λόγω καταχωρητή προσομοιώθηκε με την αναστροφή του αντίστοιχου bit του καταχωρητή. Μερικές φορές, ακόμη και αν μια SEU συμβαίνει, υπάρχει η δυνατότητα ανάκτησης της τιμής ενός καταχωρητή σε μια καλή κατάσταση κατά τη διάρκεια της λειτουργίας του αγωγού.
Για να ληφθεί μέριμνα για αυτό το αποτέλεσμα στις προσομοιώσεις σφαλμάτων μας, η αντιστραμμένη τιμή δεν εξαναγκάστηκε στον καταχωρητή επ' αόριστον, αλλά διαμορφώθηκε ως "κατατεθεί" στην εντολή του λογικού προσομοιωτή, έτσι ώστε να επιτρέπεται η ανάκτηση της κατάστασης του συστήματος, όπου αυτό ήταν δυνατό.

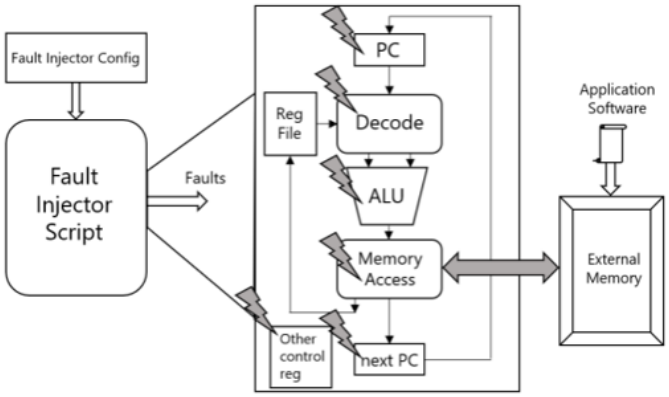


Figure 4: Αρχιτεκτονική Εκμάθησης του μοντέλου ML

Για τη δημιουργία των δεδομένων εκπαίδευσης, ένα λογισμικό εφαρμογής πρόγραμμα που απεικονίζει έναν τυπικό φόρτο εργασίας με τη χρήση όλου του συνόλου εντολών SPARC-v8 δημιουργήθηκε, το οποίο μπορεί να εκτελεστεί για περίοδο 544.000 κύκλων ρολογιού. Το πρόγραμμα εκτελέστηκε με διαφορετικές διαμορφώσεις σφαλμάτων επιπλέον στην καλή εκτέλεση προσομοίωσης. Οι ακόλουθες δύο κατηγορίες προσομοιώσεις σφαλμάτων εκτελέστηκαν πολλές φορές για τη δημιουργία εκπαιδευτικών δεδομένων από διαφορετικά σενάρια:

- Υψηλός ρυθμός έγχυσης σφαλμάτων: 100.000 - 200.000 σφάλματα ανά εκτέλεση προσομοίωσης
- Χαμηλός ρυθμός εισαγωγής σφαλμάτων: 10-100 σφάλματα ανά προσομοίωση.

Στο πρώτο σετ εκτελέσεων προσομοίωσης σφαλμάτων (ρυθμός υψηλής έγχυσης σφαλμάτων), παρατηρήθηκε ότι ο επεξεργαστής εισέρχεται σε κατάσταση παγίδευσης. Οι παγίδες και οι εξαιρέσεις χρησιμεύουν ως ενσωματωμένοι δείκτες για την ειδοποίηση του επεξεργαστή ότι συνέβησαν σφάλματα κατά την διάρκεια της εκτέλεσης. Σε αυτό το σενάριο, μερικά σήματα είχαν υψηλή συσχέτιση με την καλή και ελαττωματικές καταστάσεις προσομοίωσης. Ως εκ τούτου, ένα απλό δέντρο αποφάσεων και η αντίστοιχη λογική συνάρτηση που χρησιμοποιούσε μόνο αυτά τα υψηλά συσχετιζόμενα σήματα, χρησίμευσε ως κύκλωμα επιτήρησης.
Στη δεύτερη σειρά εκτελέσεων προσομοίωσης σφάλματος (ρυθμός χαμηλής έγχυσης σφαλμάτων) παρατηρήθηκε ότι ο επεξεργαστής έκανε συχνά δεν εισερχόταν στην κατάσταση παγίδευσης. Είναι επιτακτική ανάγκη για το κύκλωμα watchdog να συλλαμβάνει τέτοια συμβάντα SEU, ακόμη και αν οι ενσωματωμένοι μηχανισμοί (καταστάσεις παγίδευσης/σφαλμάτων) δεν ενεργοποιούνται. Δημιουργήθηκαν δεδομένα εκπαίδευσης για προσομοιώσεις χαμηλού σφάλματος με ενεργοποιημένες παγίδες και εξαιρέσεις, όπως καθώς και απενεργοποιημένες.

Το δενδρικό μοντέλο με τις καλύτερες μετρικές επικύρωσης από την αναζήτηση πλέγματος επιλέχθηκε. Ένας αναδρομικός αλγόριθμος δενδροειδούς περιπάτου όπου φαινεται στην Εικόνα 5 για τη μετατροπή αυτού του δέντρου απόφασης σε ένα ισοδύναμο λογική συνάρτηση, η οποία εξάγει True σε περίπτωση σφάλματος και False σε περίπτωση ενός καλού διανύσματος. Τέλος, αυτή η λογική συνάρτηση αξιολογήθηκε στο σύνολο δοκιμών για να προκύψουν οι μετρικές γενίκευσης.

```
input: tree – a graph describing a decision tree for binary data
output: logic – a string of nested logical operations

function get_logic(tree)
  function subtree(node,depth)
    logic = ""
    if at an interior node
      x ← tree.get_feature_at_node[node]
      logic += "~x & ("
      logic += subtree(tree.left_child[node],depth+1)
      logic += ")"
      logic += " | x & ("
      logic += subtree(tree.right_child[node],depth+1)
      logic += ")"
      return logic
    else # leaf node
      return "True" if more fault examples at leaf else "False"
  return subtree(0,1)
```

Figure 5: Αλγόριθμος απόφασης

Ο πίνακας σύγχυσης για τα δεδομένα δοκιμής στο υψηλό σενάριο ρυθμού έγχυσης σφαλμάτων παρουσιάζεται στην Εικόνα 6.

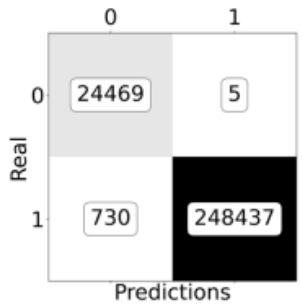


Figure 6: High Fault Injection

Στο σενάριο υψηλού ρυθμού έγχυσης σφαλμάτων το καλύτερο μοντέλο επιλέγεται από την αναζήτηση πλέγματος με τις εξής παραμέτρους:

- βάθος - 10
- κόμβοι φύλλων - 16
- ελάχιστα δείγματα ανά κόμβο φύλλων - 1E-3 του συνολικά

Από τα 511 έγκυρα ψηφία καταχωρητών του επεξεργαστή μόνο 14 καταχωρητές χρησιμοποιήθηκαν από το μοντέλο για εξαγωγή συμπερασμάτων. Αυτό το μοντέλο είχε ως αποτέλεσμα σε μετρικές:

- 0,9971 ανάκληση
- 0,9999 ακρίβεια
- 0,9985 Fscore.

Η προσθήκη της χωρητικότητας σε αυτό το σημείο δεν βελτίωσε την απόδοση και προσαρμόστηκε υπερβολικά στα δεδομένα.

Ο πίνακας σύγχυσης για τα δεδομένα δοκιμής στο χαμηλό σενάριο ρυθμού έγχυσης σφαλμάτων παρουσιάζεται στην Εικόνα 7.

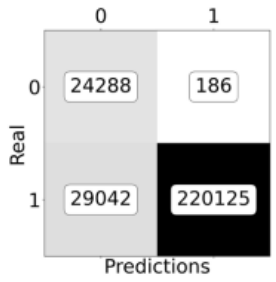


Figure 7: Low Fault Injection

το μοντέλο που ανακαλύφθηκε με τις καλύτερες μετρικές γενίκευσης είχε παραμέτρους:

- βάθος - 32
- κόμβοι φύλλων - 128
- ελάχιστα δείγματα ανά κόμβο φύλλων - 1E-6 του συνολικά

Από τους 506 έγκυρους επεξεργαστές bits καταχωρητών, 104 bits καταχωρητών χρησιμοποιήθηκαν από το μοντέλο για εξαγωγή συμπερασμάτων. Αυτό το μοντέλο οδήγησε σε μετρικές:

- 0,8834 ανάκληση
- 0,9992 ακρίβεια
- 0,9379 Fscore.

Τα μοντέλα με υψηλότερο χωρητικότητα παρατηρήθηκε ότι προσαρμόζονταν υπερβολικά στα δεδομένα εκπαίδευσης. Απλούστερη μοντέλα παρατηρήθηκε υποπροσαρμογή, με τη μεροληψία να εκδηλώνεται ως τάση να ταξινομείται λανθασμένα η σπανιότερη κλάση ως η πιο συχνή κλάση στο σύνολο δεδομένων.

Το κύκλωμα watchdog σχεδιάστηκε με βάση τη λογική λειτουργία που προέκυψε στο σενάριο χαμηλού ρυθμού έγχυσης σφαλμάτων. Αυτό είναι το χειρότερο σενάριο, όπου η διαχωρισιμότητα των καλών και των σφαλμάτων ήταν χαμηλή. Ο σχεδιασμός του κυκλώματος μεταφράστηκε απευθείας από τη λογική συνάρτηση με τη χρήση της γλώσσας περιγραφής υλικού VHDL.
Το εργαλείο σύνθεσης Synopsys Design Compiler συνέθεσε περαιτέρω και βελτιστοποίησε το κύκλωμα watchdog χρησιμοποιώντας το CMOS 180 nm και υλοποίησε τη λογική λειτουργία με:

- 148 λογικές πύλες δύο εισόδων
- 42 λογικούς αντιστροφείς
- βάθος 26 επιπέδων.

Η κρίσιμη διαδρομή χρονισμού (μέγιστη καθυστέρηση), και ισοδύναμα, ο μέγιστος χρόνος συμπερασμού για αυτό το κύκλωμα είναι:

- 2,11 ns

Αυτό εξασφαλίζει ότι το κύκλωμα watchdog είναι είναι σε θέση να ειδοποιήσει το σύστημα για ένα σφάλμα γρήγορα. Μετά την ενσωμάτωση του κυκλώματος watchdog στον επεξεργαστή, επανεκτελέστηκαν προσομοιώσεις καλής λειτουργίας και σφαλμάτων. Αυτές οι προσομοιώσεις λειτούργησαν ως ζωντανός έλεγχος της αποτελεσματικότητας του watchdog στον εντοπισμό μιας πιθανής αλλοίωσης της κατάστασης του επεξεργαστή λόγω SEU. Τα αποτελέσματα από τις διάφορες εκτελέσεις προσομοίωσης με 544.000 κύκλους ρολογιού έχουν ως εξής:

- Καμία έγχυση σφάλματος: Το κύκλωμα watchdog λειτουργεί σωστά και ταξινόμησε την κατάσταση ως καλή στο 99,72% των περιπτώσεων
- Υψηλό ποσοστό ένεσης σφαλμάτων: Η έξοδος του watchdog υπέδειξε σφάλμα για το 99,97% των διανυσμάτων σφάλματος.
- Χαμηλό ποσοστό ένεσης σφαλμάτων: Το watchdog σωστά προέβλεψε σφάλμα στο 89,29% των περιπτώσεων, εξαιρουμένου ενός

Εάν το σφάλμα ήταν σοβαρό και ο επεξεργαστής εξαναγκάστηκε σε κατάσταση σφάλματος που δεν μπορεί να αποκατασταθεί, οι επιδόσεις έμοιαζαν με το σενάριο της υψηλής έγχυσης σφάλματος. Οι παραπάνω παρατηρήσεις δείχνουν ότι το κύκλωμα watchdog ήταν σε θέση να προβλέψει ένα σφάλμα στην κατάσταση του επεξεργαστή σε ζωντανή ανάπτυξη με μετρικές επιδόσεων σύμφωνες με τις παρατηρήσεις από την ανάλυση δέντρου αποφάσεων.

## V. Λειτουργικό πρωτότυπο πολλαπλού WATCHDOG σε FPGA

Τα ανεκτικά σε σφάλματα ενσωματωμένα συστήματα χαρακτηρίζονται από αυστηρές απαιτήσεις για την ασφάλεια της λειτουργίας τους. Οι απαιτήσεις αυτές περιλαμβάνουν

- εγγυημένο χρόνο απόκρισης
- συνεχή λειτουργία
- αυτοδιάγνωση

Ένα τυπικό ενσωματωμένο σύστημα ανεκτικό σε σφάλματα ή σε πραγματικό χρόνο που εκτελεί κρίσιμες διαδικασίες (π.χ. στην αυτοκινητοβιομηχανία ή στη βιομηχανία) είναι συνήθως εξοπλισμένο με ένα μόνο υλικό watchdog χρονοδιακόπτη που εξασφαλίζει ολόκληρο το σύστημα με όλες τις ταυτόχρονες διεργασίες που εκτελούνται. Η χρήση ενός τέτοιου watchdog για την επίβλεψη πολλών παράλληλα εκτελούμενων διεργασιών είναι αναποτελεσματική, καθώς στην περίπτωση αυτή υπάρχει μόνο ένα λογισμικό οδηγός που εξομοιώνει πολλαπλά watchdogs και στην πραγματικότητα λειτουργεί ο χρονοδιακόπτης watchdog υλικού. Στην περίπτωση αποτυχίας σε μία από τις διεργασίες που εκτελούνται, είναι δύσκολο να προσδιοριστεί ποια διεργασία μόλις απέτυχε. Και αν το πρόγραμμα οδήγησης λογισμικού δεν εκτελείται σωστά,τότε είναι σχεδόν αδύνατο. Εάν το watchdog υλικού αποτύχει ή εάν κάποια λανθασμένη εφαρμογή επαναφέρει το watchdog χωρίς να το γνωρίζει,τότε όλες οι διεργασίες που εκτελούνται δεν θα παραμείνουν ασφαλείς πλέον.

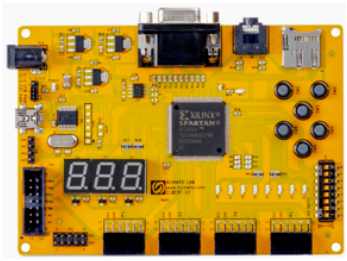

Figure 8: FPGA Spartan-3A

Ένα πολλαπλό σύστημα watchdog υλικού με την ικανότητα να διασφαλίζει όλες τις διεργασίες και ολόκληρο το ενσωματωμένο σύστημα μπορεί ανεξάρτητα να αυξήσει σημαντικά την αξιοπιστία του συστήματος πραγματικού χρόνου. Ένα από τα βασικά χαρακτηριστικά ενός watchdog είναι να "εκδίδει ένα υλικό επαναφορά σε περίπτωση χρονικού ορίου". Ο συνήθης τρόπος για την εξασφάλιση πολλαπλών διεργασιών ενσωματωμένων συστημάτων είναι η χρήση λογισμικών watchdogs. Το watchdog λογισμικού παρέχει διάφορους εικονικούς χρονομετρητές watchdog

ή άλλα λογισμικά μηχανισμών για την εξασφάλιση ταυτόχρονων διεργασιών, ενώ χρησιμοποιείται μόνο ένα watchdog υλικού. Η πλατφόρμα FPGA που χρησιμοποιήθηκε φαίνεται στην Εικόνα 8

Υπάρχουν διάφορες εμπορικές επεκτάσεις watchdog, οι οποίες μπορούν να συνδεθούν σε προσωπικούς ή βιομηχανικούς υπολογιστές. Ωστόσο, τα συστήματα αυτά λειτουργούν μόνο με έναν ή λίγους χρονιστές watchdog υλικού. Οι πιο συνήθη χρησιμοποιούμενες διεπαφές επικοινωνίας για αυτά είναι η σειριακή θύρα, το περιφερειακό στοιχείο (PCI) και η θύρα πληκτρολογίου. Βασίζουμε την εργασία μας στο πειραματικό πολλαπλό σύστημα watchdog υλικού που υλοποιήθηκε σε FPGA. Αυτή η αρχιτεκτονική του συστήματος είναι αρθρωτή και εύκολα επεκτάσιμη. Χρησιμοποιεί την τυπική σειριακή θύρα (UART) για την επικοινωνία με το ασφαλές σύστημα.
Το βασικό Counter Unit χωρίζεται σε δύο στοιχεία. Το πρώτο στοιχείο είναι ένας μικρός χρονομετρητής 24-bit, ενώ το δεύτερο στοιχείο είναι μια λογική που ενεργοποιεί τον χρονοδιακόπτη. Η κύρια μονάδα του συστήματος είναι η μονάδα ελέγχου η οποία παρέχει διασύνδεση μεταξύ των κυκλωμάτων watchdog και UART. Κάθε watchdog έχει ξεχωριστή διευθυνσιοδότηση από τη μονάδα ελέγχου και η διευθυνσιοδότηση ή διασύνδεση είναι κλιμακωτή για να καταστεί δυνατή η μελλοντική επέκταση. Το σχεδιασμένο σύστημα απαιτεί λογισμικό στην πλευρά του ασφαλούς συστήματος.

Αυτή η ενότητα παρουσιάζει τη λειτουργία του σχεδιασμένου συστήματος ως μαύρο κουτί. Σε αυτό το έγγραφο χρησιμοποιούμε δύο όρους για να διακρίνουμε δύο διαφορετικά συστήματα.

- Το ασφαλές σύστημα είναι το ενσωματωμένο σύστημα που εκτελεί πολλαπλές διεργασίες.
- Το σύστημα ασφαλείας είναι το σχεδιασμένο μας σύστημα με πολλαπλούς watchdogs

Το σύστημά μας παρέχει μια σειρά από υλικό για την ασφάλεια κάθε διεργασίας που εκτελείται σε ένα ενσωματωμένο σύσταμα. Η σύνδεση μεταξύ των ασφαλών και του συστήματος ασφαλείας δημιουργείται με το σύστημα USB διεπαφή USB. Το ασφαλές σύστημα απαιτείται να διαθέτει οδηγό λογισμικού που είναι σε θέση να επικοινωνεί με το σύστημα ασφαλείας και να εκχωρεί ενάν watchdog σε κάθε διεργασία που εκτελείται. Το σύστημα ασφαλείας είναι σε θέση να φορτώσει μια νέα τιμή στο συγκεκριμένο διευθυνσιοδοτούμενο watchdog, να το μηδενίσει ή να το τερματίσει σωστά. Κάθε διεργασία μπορεί να έχει διαφορετικές απαιτήσεις για την τιμή που φορτώνεται στο watchdog. Η τιμή αυτή πρέπει να έχει υπολογιστεί και να είναι γνωστή στο ασφαλές σύστημα πριν από την έναρξη της διαδικασίας. Όλα τα παρεχόμενα watchdogs είναι 24-bit και μετρούν το χρόνο σε χιλιοστά του δευτερολέπτου από 0 ms έως 4,66 ώρες. Αυτό το χρονικό διάστημα είναι κατάλληλο για διαφορετικές απαιτήσεις των διεργασιών που εκτελούνται.

Όταν ένας χρονομετρητής watchdog υπερχειλίσει, το διασφαλισμένο σύστημα αποστέλλει ένα μήνυμα υπερχείλισης που περιέχει τη συγκεκριμένη διεύθυνση του χρονοδιακόπτη υπερχείλισης στο ασφαλές σύστημα. Μετά τη λήψη αυτού του μηνύματος, το ασφαλές σύστημα πρέπει να αντιδράσει κατάλληλα, ενώ γενικά έχει τις εξής δυνατότητες:

- Επανεκκίνηση/επαναφορά της αποτυχημένης διεργασίας και επανεκκίνηση του watchdog του με την ίδια τιμή.
- Επανεκκίνηση/επαναφορά της αποτυχημένης διεργασίας και φόρτωση μιας νέας (π.χ. υψηλότερης) τιμής στο watchdog χωρίς να χρειάζεται επανεκκίνηση.
- Τερματισμός της αποτυχημένης διεργασίας σωστά και απελευθέρωση του διατιθέμενου watchdog. Όταν υπερχειλίζουν περισσότερα από ένα watchdog, αποστέλλονται διαδοχικά μηνύματα.

Στην περίπτωση που όλες οι εκτελούμενες και ασφαλείς διεργασίες αποτυγχάνουν και ο οδηγός λογισμικού watchdog (ο οποίος επίσης θα έπρεπε να έχει εκχωρήσει ένα χρονοδιακόπτη watchdog) αποτυγχάνει επίσης, το δικό μας σύστημα έχει τη δυνατότητα να ανιχνεύει αυτόματα την αποτυχία και να επαναφέρει ολόκληρο το σύστημα μετά από μια δεδομένη περίοδο χρονικού διαστήμα. Αυτός ο χρόνος καθυστέρησης είναι διαμορφώσιμος από ένα από τα μηνύματα του πρωτοκόλλου επικοινωνίας.
Η καθυστέρηση είναι σημαντικό επειδή το ασφαλές σύστημα μπορεί να έχει χρησιμοποιήσει άλλους μηχανισμούς ανίχνευσης αποτυχίας και μπορεί να είναι σε θέση να επανεκκινήσει με μεγαλύτερη ασφάλεια χωρίς τη βοήθεια του ασφαλούς συστήματος. Η έξοδος επαναφοράς υλικού του πολλαπλού πρέπει να συνδεθεί απευθείας με το σύστημα επαναφοράς του ασφαλούς συστήματος και πρέπει να έχει ανατροφοδότηση υλικού για να γνωρίζει αν το ασφαλές σύστημα έχει επανεκκινηθεί. Στην Εικόνα 9 παρουσιάζεται η αρχιτεκτονική σε επίπεδο συστήματος.

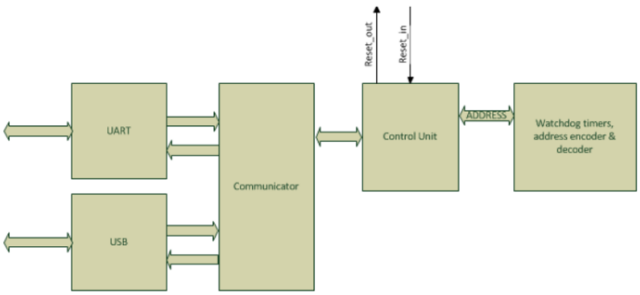


Figure 9: Αρχιτεκτονική Συστήματος

- Communicator

Χρησιμοποιούμε ένα μπλοκ επικοινωνίας με παράλληλη διεπαφή. Αυτό το μπλοκ συνδέεται με τις πραγματικές διεπαφές επικοινωνίας. Είναι υπεύθυνο για την αποκωδικοποίηση των ληφθέντων μηνυμάτων από το ασφαλές σύστημα, την αποστολή της επιβεβαίωσης και άλλων μηνυμάτων του πρωτοκόλλου επικοινωνίας. Η λειτουργία του είναι η διαμεσολάβηση μεταξύ της μονάδας ελέγχου και των

διεπαφών επικοινωνίας. Το πρωτόκολλο επικοινωνίας έχει σχεδιαστεί για να είναι μινιμαλιστικό, επειδή ο χρόνος είναι η κρίσιμη παράμετρος σε αυτό το σύστημα και η επικοινωνία πρέπει να είναι όσο το δυνατόν απλούστερη όσο το δυνατόν πιο απλή. Από την άλλη πλευρά, πρέπει να διασφαλίσουμε ότι ο τυχαίος κώδικας μιας πιθανής αποτυχημένης εφαρμογής δεν θα μπορούσε να επαναπρογραμματίσει τη συσκευή. Στην Εικόνα 10 φαίνονται όλα τα μηνύματα που χρησιμοποιήθηκαν για την υλοποίηση της επικοινωνίας μεταξύ ενσωματωμένου συστήματος και FPGA.

| Message | Operation Code | First 8 bits | Next 24 bits |
|---|---|---|---|
| INIT | 0x2D | Priority watchdog quantity (5 b), Delay in sec. (3 b) | - |
| LOAD | 0x3B | Address | Value (lower bytes first) |
| RESET | 0x1A | Address | - |
| TERMINATE | 0xF9 | Address | - |
| WARNING | 0x49 | Delay in sec. (lower 3 b) | - |
| ZERO | 0x52 | Address | - |
| ACK | 0x6E | - | - |

Figure 10: Πίνακας μηνυμάτων

Μήνυμα init περιέχει παραμέτρους για τη διαμόρφωση της μονάδας ελέγχου. Οι παράμετροι διαμόρφωσης είναι η κλίμακα προτεραιότητας και ο χρόνος καθυστέρησης σε δευτερόλεπτα. Τα watchdogs προτεραιότητας είναι watchdogs με υψηλότερη προτεραιότητα. Εάν όλα αυτά τα watchdogs προτεραιότητας αποτύχουν, το σύστημα ασφαλείας δεν περιμένει για άλλα watchdogs υπερχείλισης και στέλνει την προειδοποίηση και την επαναφορά υλικού. Προτεραιότητες δεν έχουν ακόμη υλοποιηθεί σε αυτό το πρωτότυπο. Ο χρόνος καθυστέρησης είναι ο χρόνος κατά τον οποίο το σύστημα ασφαλείας περιμένει την αυτοεπαναφορά του συστήματος ασφαλείας. Το φορτίο χρησιμοποιείται για τη φόρτωση μιας νέας τιμής 24 bit στο συγκεκριμένο χρονοδιακόπτη. Η επαναφορά του μηνύματος επανεκκινεί το συγκεκριμένο watchdog. Μηνύματα που αποστέλλονται στο ασφαλές σύστημα προειδοποιούν πριν από την επαναφορά του υλικού με καθυστέρηση και μηδέν που υποδεικνύει ότι ένα συγκεκριμένο watchdog έχει υπερχειλίσει.
Κάθε λαμβανόμενο byte ενός μηνύματος από ασφαλές σύστημα επιβεβαιώνεται με το μήνυμα ACK, λόγω των μικρών απομονωτών στη μονάδα ελέγχου ή μνήμης του επικοινωνητή. Το μήνυμα αυτό υποδεικνύει την απάντηση του συστήματος ασφαλείας. Εάν το σύστημα ασφαλείας δεν απαντά σε κανένα μήνυμα ACK στα μηνύματα αποστολής από το ασφαλές σύστημα, αυτό υποδεικνύει ότι το πολλαπλό σύστημα παρακολούθησης έχει αποτύχει.

- Control Unit

Η μονάδα ελέγχου λαμβάνει και παράγει τα μηνύματα στον επικοινωνητή, ελέγχει όλα τα watchdogs, ανιχνεύει αν το ασφαλιζόμενο σύστημα έχει αποτύχει και ενεργοποιεί την επαναφορά υλικού του ασφαλισμένου συστήματος όταν χρειάζεται. Η αποτυχία του συστήματος που απαιτεί επαναφορά υλικού ανιχνεύεται από δύο καταχωρητές που συγκρίνουν το ποσό των ενεργοποιημένων watchdogs και των υπερχειλίσεων και των μη ακόμη επανεκκινήσιμων watchdogs.
Εάν αυτοί οι δύο μετρητές είναι ίσοι και δεν είναι μηδέν, θεωρείται ότι το ασφαλισμένο σύστημα έχει αποτύχει. Ο δίαυλος διευθύνσεων που χρησιμοποιείται για τη διευθυνσιοδότηση των watchdogs έχει χωριστεί σε δύο ξεχωριστούς διαύλους. Ο δίαυλος εξόδου διευθύνσεων από τη μονάδα ελέγχου χρησιμοποιείται κατά τη διευθυνσιοδότηση ενός watchdog και π.χ. τη φόρτωση της νέας τιμής σε αυτό. Η δίαυλος εισόδου διευθύνσεων χρησιμοποιείται για την ανάγνωση της διεύθυνσης του watchdog που έχει υπερχειλίσει.

- Watchdog

Κάθε χρονοδιακόπτης watchdog έχει μόνο μία είσοδο ρολογιού για να επικοινωνεί με τη μονάδα ελέγχου. Η τιμή φόρτωσης αποστέλλεται από τη μονάδα ελέγχου σειριακά. Ένα εσωτερικό ρολόι διαιρέτης που περιέχεται σε κάθε χρονοδιακόπτη watchdog χρησιμοποιείται για την καταμέτρηση του χρόνου από την τιμή φόρτωσης. Η μέτρηση συχνότητας είναι 1 kHz. Είναι απλούστερο να υπάρχει μόνο ένα ρολόι που δρομολογείται στην FPGA σε πολλά μπλοκ. Ένα watchdog είναι διασύνδεση με τη μονάδα ελέγχου από το block enabler που ενεργοποιεί το συγκεκριμένο watchdog σύμφωνα με την δίαυλο διευθύνσεων. Αυτό το μπλοκ διαθέτει έναν καταχωρητή για την αποθήκευση της τιμής παύσης. Ο χρονοδιακόπτης watchdog έχει πέντε βασικές καταστάσεις:

- Κατάσταση λειτουργίας - ο χρονοδιακόπτης μετράει από τη φορτωμένη τιμή
- Κατάσταση επαναφοράς - ο χρονοδιακόπτης επαναφέρεται από την τρέχουσα τιμή στην αρχική τιμή.
- Κατάσταση παύσης - υποδεικνύει ότι ο χρονομετρητής δεν χρησιμοποιείται.
- Κατάσταση φόρτωσης - ο χρονοδιακόπτης φορτώνει την τιμή από τη μονάδα ελέγχου.
- Κατάσταση υπερχείλισης - υποδεικνύει ότι ο χρονοδιακόπτης έχει υπερχειλίσει.

Το watchdog διαθέτει ένα ακόμη σήμα (που ονομάζεται μηδενικό) που συνδέεται με τον ενεργοποιητή. Από κάθε ενεργοποιητή είναι το σήμα μηδέν συνδεδεμένο σε έναν πολυπλέκτη. Ο έλεγχος μπορεί να επιλέξει και να διαβάσει ένα σήμα σύμφωνα με την διεύθυνση του ρολογιού. Αυτό το σήμα υποδεικνύει ότι το συγκεκριμένο watchdog έχει υπερχειλίσει. Όταν περισσότερα από ένα watchdog υπερχειλίζουν, το σήμα υπερχείλισης πρέπει να επιστρέψει σε ανενεργό επίπεδο για να επιτρέψει σε άλλα watchdogs να υποδείξουν την κατάσταση υπερχείλισης. Στην Εικόνα 11 δείχνει την σύνδεση του σήματος μηδέν.

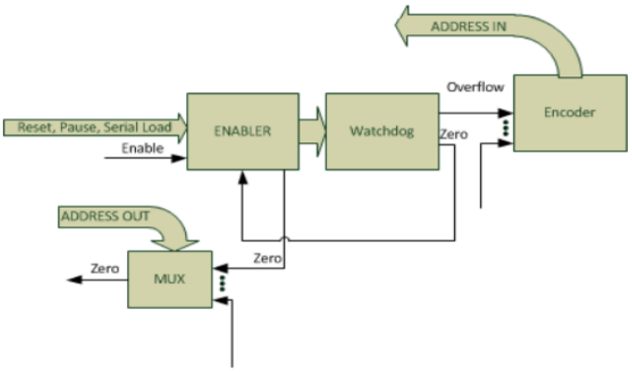


Figure 11: Watchdog Circuit

## REFERENCES

[1] D. Soares, B. Matthaus, E. S. Mota and C. B. Carvalho, ”A Feasibility Study of Watchdogs on Opportunistic Mobile Networks,” 2018 IEEE Symposium on Computers and Communications (ISCC), 2018, pp. 00123-00128, doi: 10.1109/ISCC.2018.8538648.

[2] B. K. S. V. L. Varaprasad, R. Anilkumar, B. A. Prasad, S. P. Bondapalli and K. Padmapriya, ”Design of Watchdog Circuit using Decision Trees for Detection of Single Event Upsets in Processor,” 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), 2021, pp. 1306-1311, doi: 10.1109/ICIRCA51532.2021.9544797.

[3] M. Duríček, M. Pohronská and T. Krajčovič, ”Functional prototype of multiple watchdog system implemented in FPGA,” 2012 International Conference on Applied Electronics, 2012, pp. 75-78.

[4] M. Pohronská and T. Krajčovič, ”Embedded systems with increased reliability using the multiple watchdog timers approach,” 2010 International Conference on Applied Electronics, 2010, pp. 1-4.