Βάσεις Δεδομένων
Τομέας Ηλεκτρονικής και Υπολογιστών
Τμήμα ΗΜΜΥ
Α.Π.Θ.

8ο Εξάμηνο
Χειμώνας 2015

Ενσωματωμένα Συστήματα Πραγματικού Χρόνου

Εργασία 2015

Μέκρας Αθανάσιος
7074

Περιεχόμενα

1. Εισαγωγή

Η άσκηση έχει ως σκοπό να αναζητηθεί η οριακή συμπεριφορά του υπολογιστή σε πραγματικό χρόνο. Για τις ανάγκες της άσκησης αναπτύχθηκε πρόγραμμα το οποίο αναγνωρίζει τον χρόνο που ένα σήμα αλλάζει από 0 σε 1.

Πιο συγκεκριμένα, το πρόγραμμα έχει τη δυνατότητα να αναγνωρίζει πολύ γρήγορα, σε χρόνο 1μsec και λιγότερο, τις αλλαγές Ν αριθμού σημάτων. Όπου Ν είναι ο αριθμός σημάτων που δέχεται το πρόγραμμα σαν είσοδο.

2. Περιβάλλον Ανάπτυξης/Εκτέλεσης

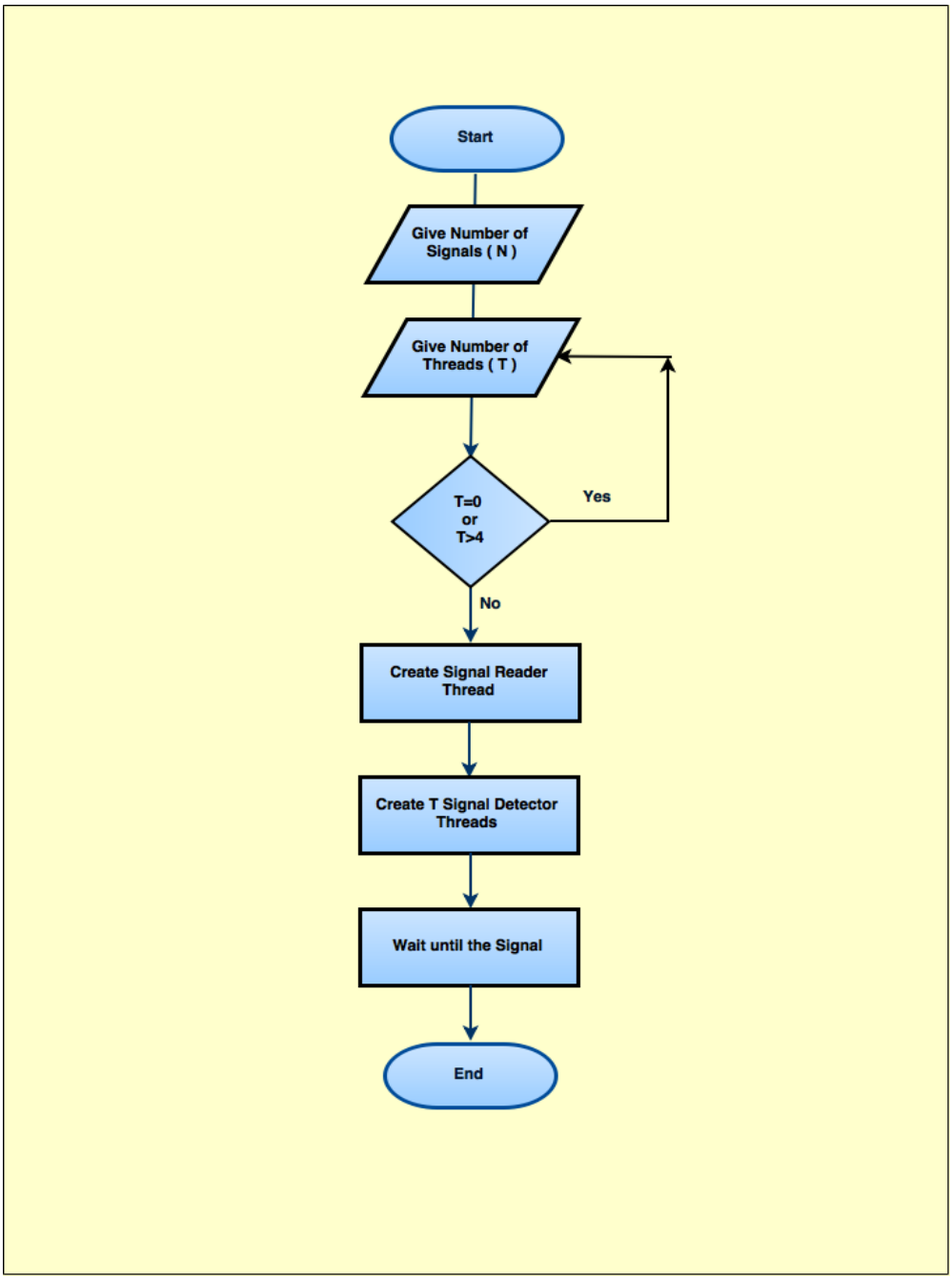
Το πρόγραμμα αναπτύχθηκε, δοκιμάστηκε και εκτελέστηκε στο εξής περιβάλλον :

Λειτουργικό Σύστημα	Windows 10 Home Edition - Cyqwin
System Type	64-bit
CPU	Intel(R) Cote(TM) i7-2720QM @2.20GHz
GPU	Nvidia GeForce GTX 460M
RAM	8 GB

3. Περιγραφή του Προγράμματος

3.1 Διάγραμμα ροής

Στο παρακάτω διάγραμμα ροής παρουσιάζεται μία απλή περιγραφή:



Όπως φαίνεται στην παραπάνω εικόνα, το πρόγραμμα καθώς ξεκινάει παίρνει ως είσοδο τον αριθμό των σημάτων και τον αριθμό των threads αναγνώρισης.

Γίνεται κάποιος έλεγχος ώστε να μην επιτρέπεται η επιλογή πάνω από 4 threads αναγνώρισης, διότι δεν υπάρχει καλή λειτουργία στον υπολογιστή που υλοποιήθηκε η εργασία.

Εν συνεχεία, δημιουργούνται τα αντίστοιχα threads :

- 1 για να διαβάζει το σήμα και να το αλλάζει (με τυχαίο αλγόριθμο)
- 1 έως 4 για να αναγνωρίσει την αλλαγή των σημάτων

Κατά την εκτέλεση σημειώνεται ο αριθμός των σημάτων που έχουν αλλάξει συνολικά (είτε από 0 σε 1, είτε από 1 σε 0), ο αριθμός των σημάτων που έχουν αλλάξει από 0 σε 1 και ο αριθμός των σημάτων που έχουν αλλάξει από 0 σε 1 και έχουν αναγνωριστεί από το σύστημα. Με αυτό τον τρόπο επιβεβαιώνεται η σωστή λειτουργία του προγράμματος.

3.2 Signal Reader Thread

Με έναν τυχαίο αλγόριθμο επιλέγονται κάποια σήματα και συγκρίνονται με XOR με το 1. Όπως είναι φυσικό επειδή ο πίνακας των σημάτων έχει αρχικοποιηθεί και έχουν λάβει όλα τα στοιχεία την τιμή 0, κάθε φορά που ένα σήμα που δεν έχει ξανά συγκριθεί με XOR του 1, αλλάζει.

3.3 Signal Detector Thread/Threads

Ανάλογα με τον αριθμό των threads που έχουν επιλεχθεί, ο αριθμός των σημάτων χωρίζονται σε ισόποσα τμήματα και αναγνωρίζονται ξεχωριστά από το κάθε thread. Με αυτό τον τρόπο επιτυγχάνεται ο επιθυμητός χρόνος σε περιπτώσεις μεγάλου αριθμού σημάτων.

4. Μετρήσεις - Παρουσίαση Αποτελεσμάτων

Για να μπορέσουν να δοθούν οι απαραίτητες εγγυήσεις πραγματικού χρόνου, λόγω της τυχαιότητας που διακατέχει το πρόβλημα, πάρθηκαν πολλές μετρήσεις οι οποίες βασίζονται σε διάφορα σενάρια.

Πιο συγκεκριμένα, διενεργήθηκαν μετρήσεις για είσοδο αριθμού σημάτων :

- 1
- 10
- 120
- 300
- 600
- 900
- 1200
- 1500
- 2000
- 5000

Και για κυμαινόμενο αριθμό thread αναγνώρισης :

- 1
- 2
- 4

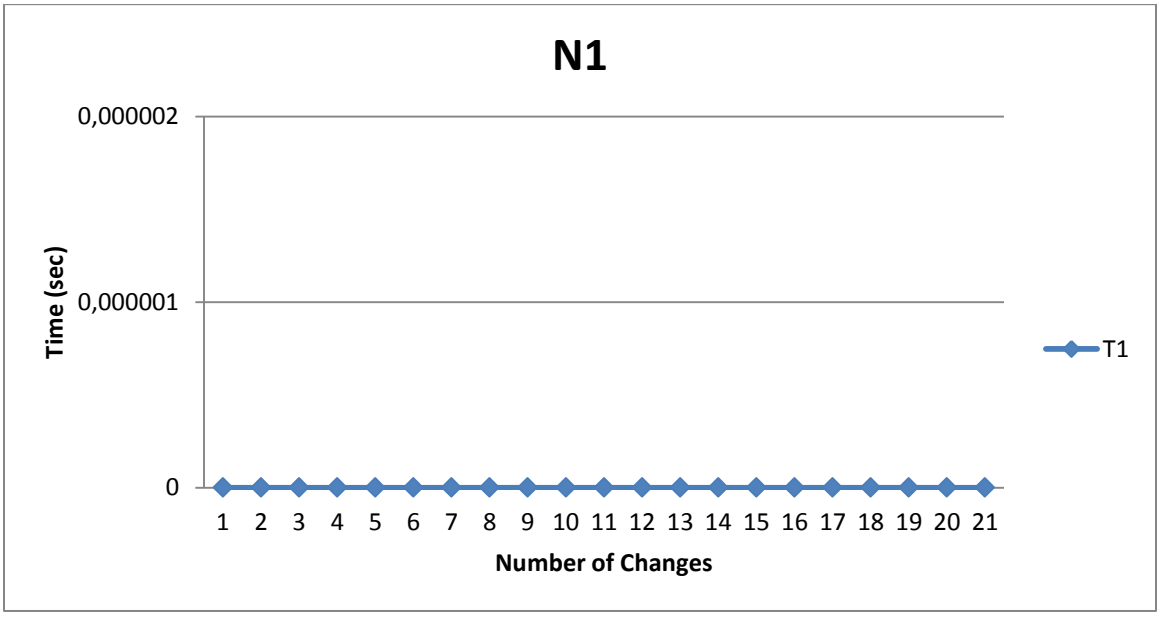
Παρακάτω παρουσιάζεται ένας συγκεντρωτικός πίνακας με τον αριθμό σημάτων και τον αριθμό των thread της κάθε μέτρησης.

Πίνακας Μετρήσεων		
Μέτρηση	**Σήματα**	**Threads**
1	1	1
2	10	1,2
3	120	1,2
4	300	1,4
5	600	1,2,4
6	900	1,2,4
7	1200	1,2,4
8	1500	1,2,4
9	2000	1,2,4
10	5000	1,2,4

4.1 Μέτρηση 1η

Σήματα	Threads	Σήματα που άλλαξαν
1	1	21

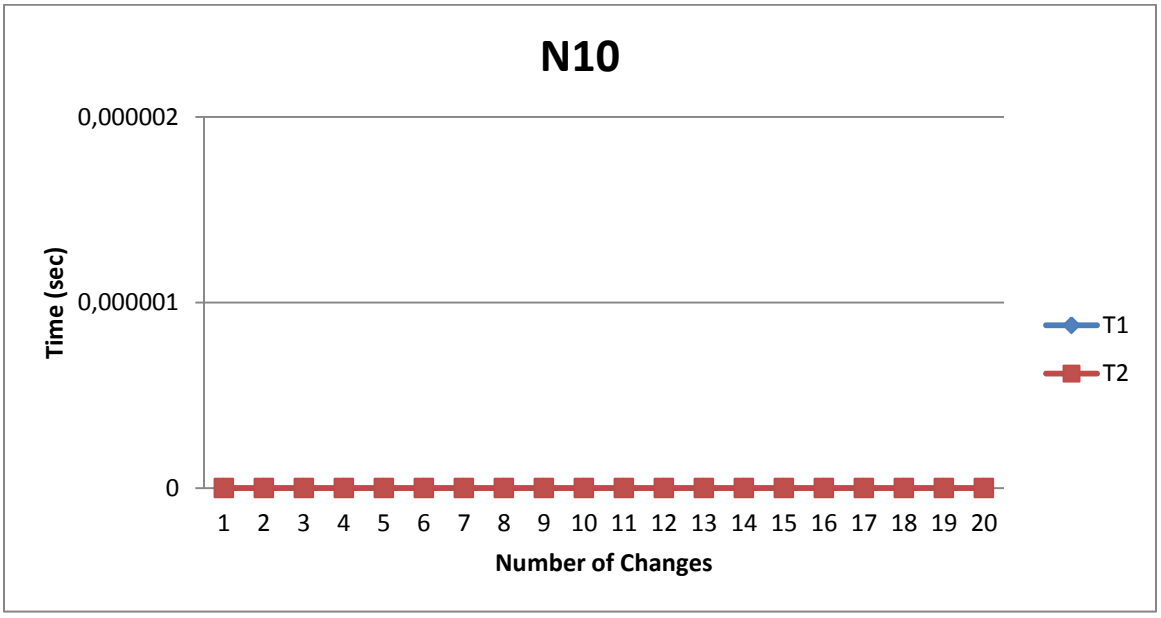
Όπως φαίνεται στο παρακάτω διάγραμμα, για αριθμό 1 σήματος και 1 thread, παρατηρήθηκε εναλλαγή σήματος 21 φορές. Η αλλαγή αναγνωρίστηκε σε χρόνο πολύ μικρότερο του 1μsec ο οποίος θεωρείται αμελητέος και γι' αυτό σημειώνεται ως 0 στο διάγραμμα.



4.2 Μέτρηση 2η

Σήματα	Threads	Σήματα που άλλαξαν
10	1,2	20

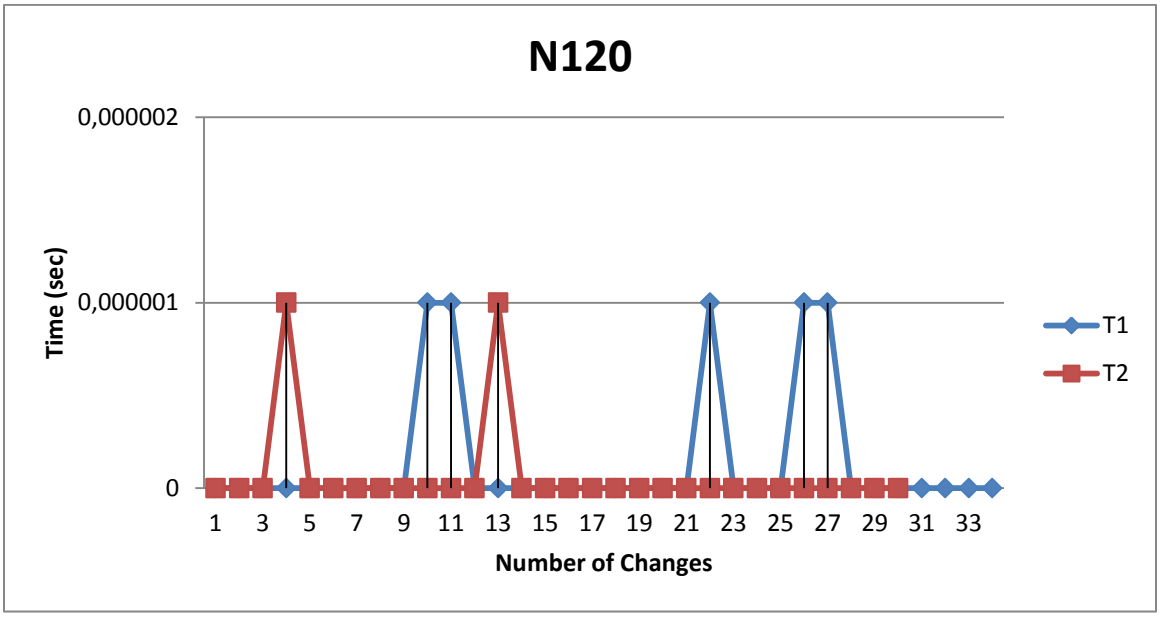
Όπως φαίνεται στο παρακάτω διάγραμμα, για αριθμό 10 σημάτων και για 1 και 2 threads, παρατηρήθηκε εναλλαγή σήματος 20 φορές. Η αλλαγή αναγνωρίστηκε σε χρόνο πολύ μικρότερο του 1μsec ο οποίος θεωρείται αμελητέος και γι' αυτό σημειώνεται ως 0 στο διάγραμμα. (Οι μπλε κουκίδες δεν φαίνονται διότι επικαλύπτονται από τις κόκκινες)



4.3 Μέτρηση 3η

Σήματα	Threads	Σήματα που άλλαξαν
120	1,2	34,30

Όπως φαίνεται στο παρακάτω διάγραμμα, για αριθμό 120 σημάτων και για 1 και 2 threads, παρατηρήθηκε εναλλαγή σήματος 34 και 30 φορές αντίστοιχα.



Η αλλαγή αναγνωρίστηκε :

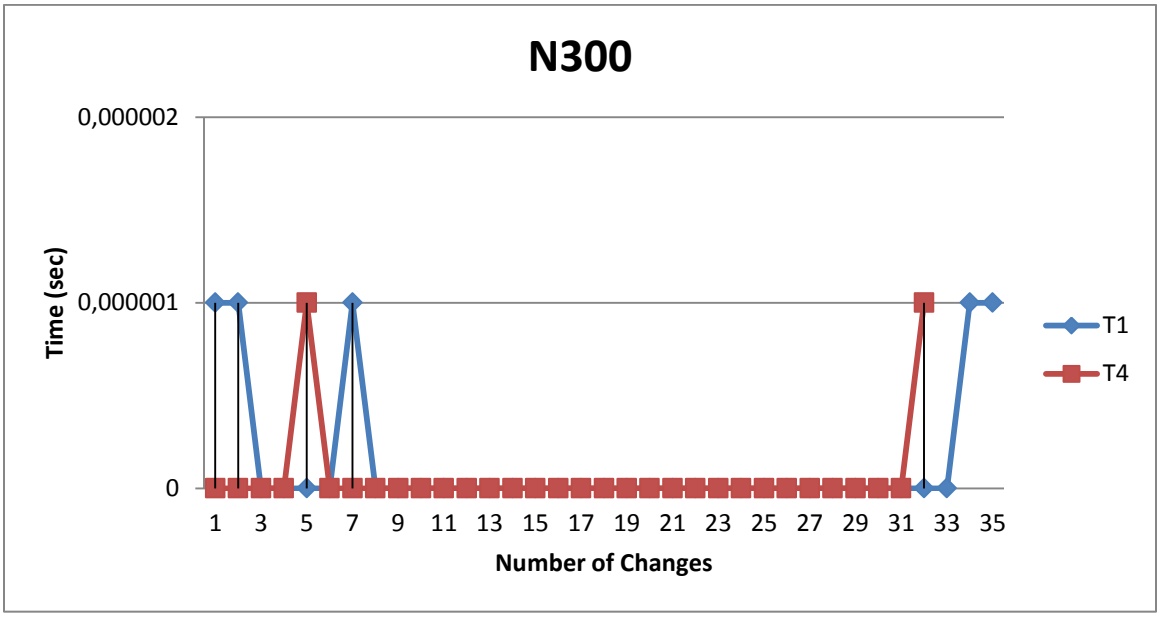
- 5 φορές σε χρόνο 1μsec για την υλοποίηση με 1 thread
- 2 φορές σε χρόνο 1μsec για την υλοποίηση με 1 threads

Τις υπόλοιπες φορές, αναγνωρίστηκε σε χρόνο πολύ μικρότερο του 1μsec ο οποίος θεωρείται αμελητέος και γι' αυτό σημειώνεται ως 0 στο διάγραμμα.

4.4 Μέτρηση 4η

Σήματα	Threads	Σήματα που άλλαξαν
300	1,4	35,32

Όπως φαίνεται στο παρακάτω διάγραμμα, για αριθμό 300 σημάτων και για 1 και 4 threads, παρατηρήθηκε εναλλαγή σήματος 35 και 32 φορές αντίστοιχα.



Η αλλαγή αναγνωρίστηκε :

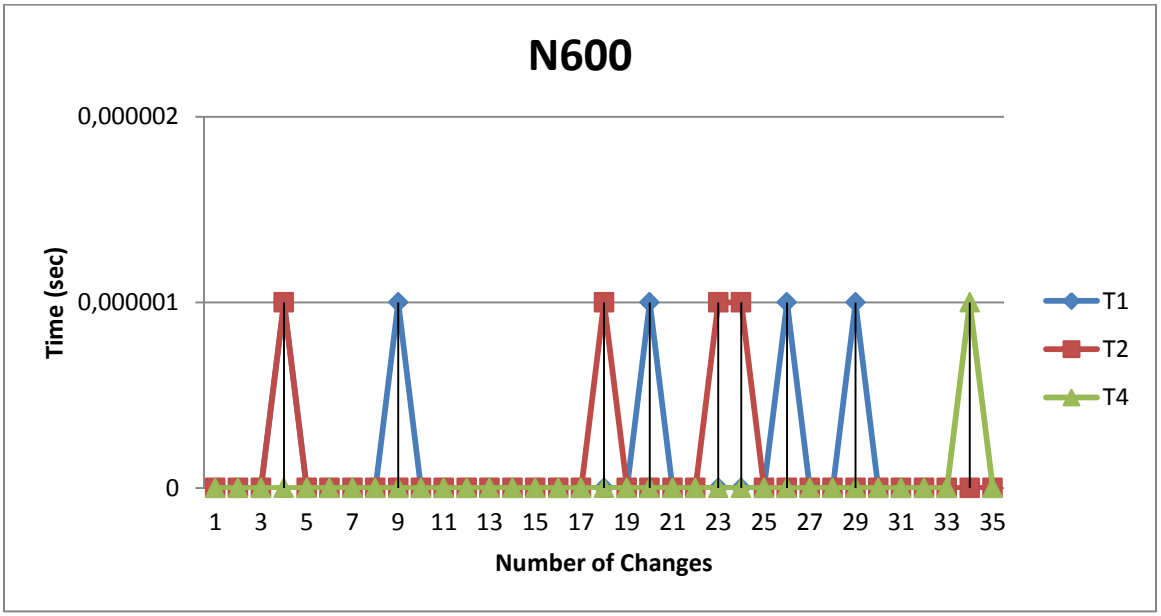
- 5 φορές σε χρόνο 1μsec για την υλοποίηση με 1 thread
- 2 φορές σε χρόνο 1μsec για την υλοποίηση με 4 threads

Τις υπόλοιπες φορές, αναγνωρίστηκε σε χρόνο πολύ μικρότερο του 1μsec ο οποίος θεωρείται αμελητέος και γι’ αυτό σημειώνεται ως 0 στο διάγραμμα.

4.5 Μέτρηση 5η

Σήματα	Threads	Σήματα που άλλαξαν
600	1,2,4	35,35,35

Όπως φαίνεται στο παρακάτω διάγραμμα, για αριθμό 600 σημάτων και για 1, 2 και 4 threads, παρατηρήθηκε εναλλαγή σήματος 35 φορές και στις τρεις περιπτώσεις.



Η αλλαγή αναγνωρίστηκε :

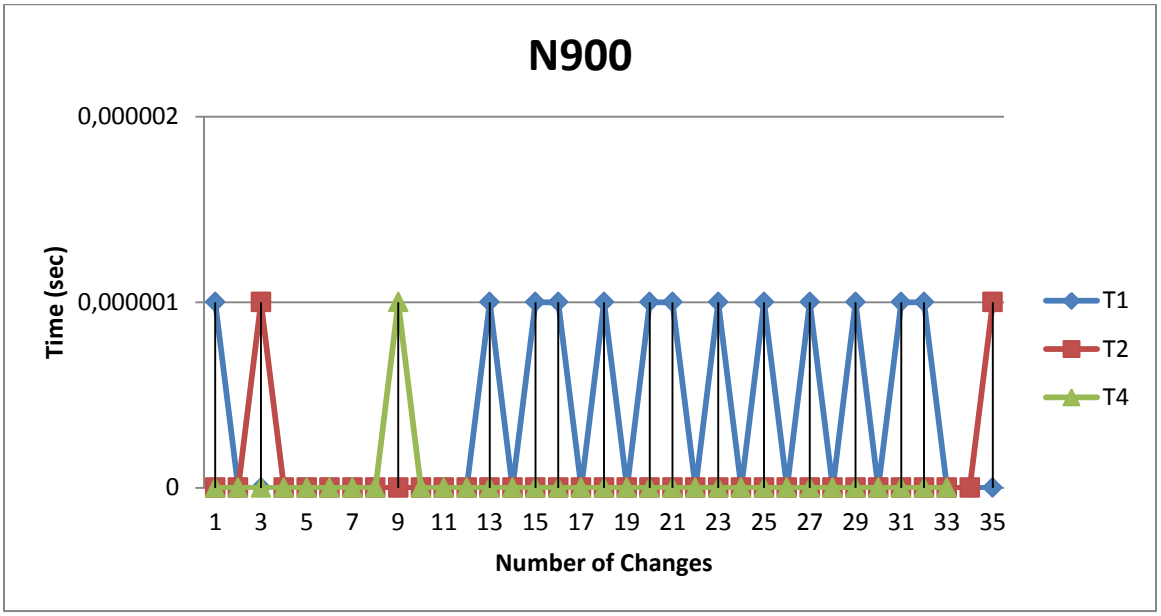
- 5 φορές σε χρόνο 1μsec για την υλοποίηση με 1 thread
- 4 φορές σε χρόνο 1μsec για την υλοποίηση με 2 threads
- 1 φορά σε χρόνο 1μsec για την υλοποίηση με 4 threads

Τις υπόλοιπες φορές, αναγνωρίστηκε σε χρόνο πολύ μικρότερο του 1μsec ο οποίος θεωρείται αμελητέος και γι' αυτό σημειώνεται ως 0 στο διάγραμμα.

4.6 Μέτρηση 6η

Σήματα	Threads	Σήματα που άλλαξαν
900	1,2,4	35,35,33

Όπως φαίνεται στο παρακάτω διάγραμμα, για αριθμό 900 σημάτων και για 1, 2 και 4 threads, παρατηρήθηκε εναλλαγή σήματος 35 φορές στην υλοποίηση με 1 και 2 threads και 33 φορές σε αυτή με τα 4 threads.



Η αλλαγή αναγνωρίστηκε :

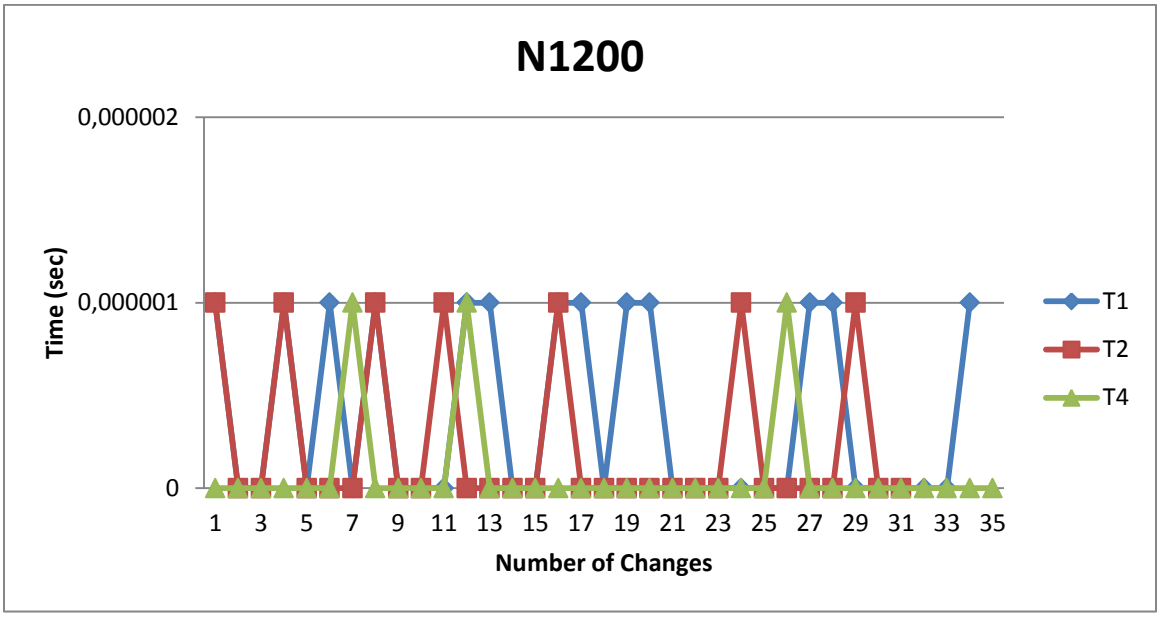
- 13 φορές σε χρόνο 1μsec για την υλοποίηση με 1 thread
- 2 φορές σε χρόνο 1μsec για την υλοποίηση με 2 threads
- 1 φορά σε χρόνο 1μsec για την υλοποίηση με 4 threads

Τις υπόλοιπες φορές, αναγνωρίστηκε σε χρόνο πολύ μικρότερο του 1μsec ο οποίος θεωρείται αμελητέος και γι' αυτό σημειώνεται ως 0 στο διάγραμμα.

4.7 Μέτρηση 7η

Σήματα	Threads	Σήματα που άλλαξαν
1200	1,2,4	34,31,35

Όπως φαίνεται στο παρακάτω διάγραμμα, για αριθμό 1200 σημάτων και για 1, 2 και 4 threads, παρατηρήθηκε εναλλαγή σήματος 34, 31 και 35 φορές αντίστοιχα.



Η αλλαγή αναγνωρίστηκε :

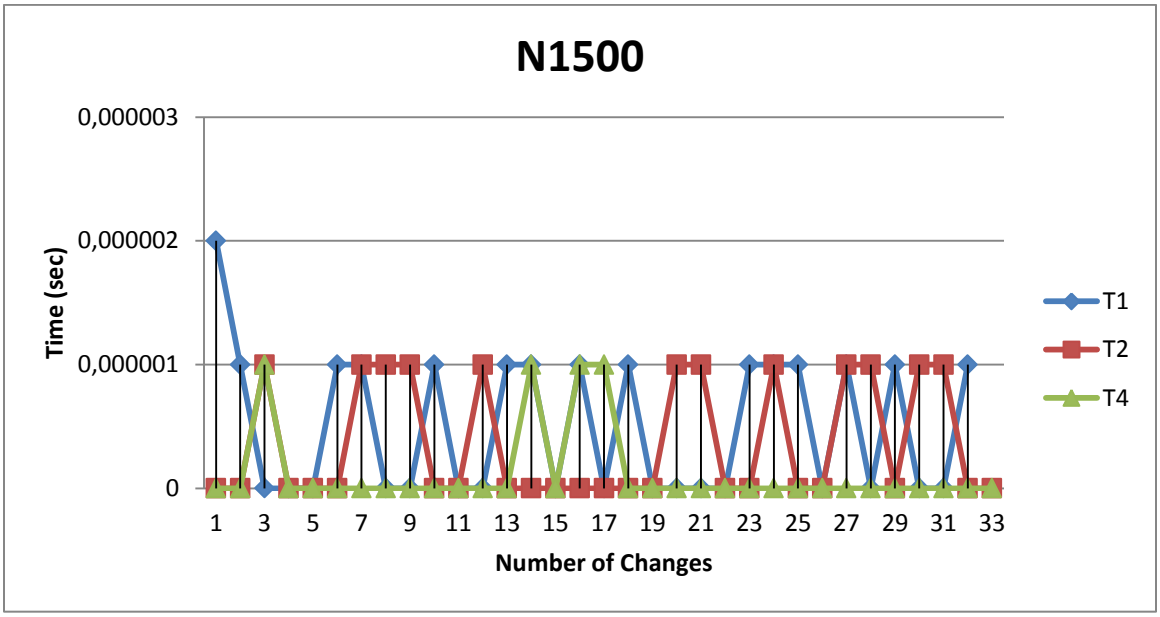
- 13 φορές σε χρόνο 1μsec για την υλοποίηση με 1 thread
- 7 φορές σε χρόνο 1μsec για την υλοποίηση με 2 threads
- 3 φορά σε χρόνο 1μsec για την υλοποίηση με 4 threads

Τις υπόλοιπες φορές, αναγνωρίστηκε σε χρόνο πολύ μικρότερο του 1μsec ο οποίος θεωρείται αμελητέος και γι' αυτό σημειώνεται ως 0 στο διάγραμμα.

4.8 Μέτρηση 8η

Σήματα	Threads	Σήματα που άλλαξαν
1500	1,2,4	32,33,33

Όπως φαίνεται στο παρακάτω διάγραμμα, για αριθμό 1500 σημάτων και για 1, 2 και 4 threads, παρατηρήθηκε εναλλαγή σήματος 32, 33 και 33 φορές αντίστοιχα.



Η αλλαγή αναγνωρίστηκε :

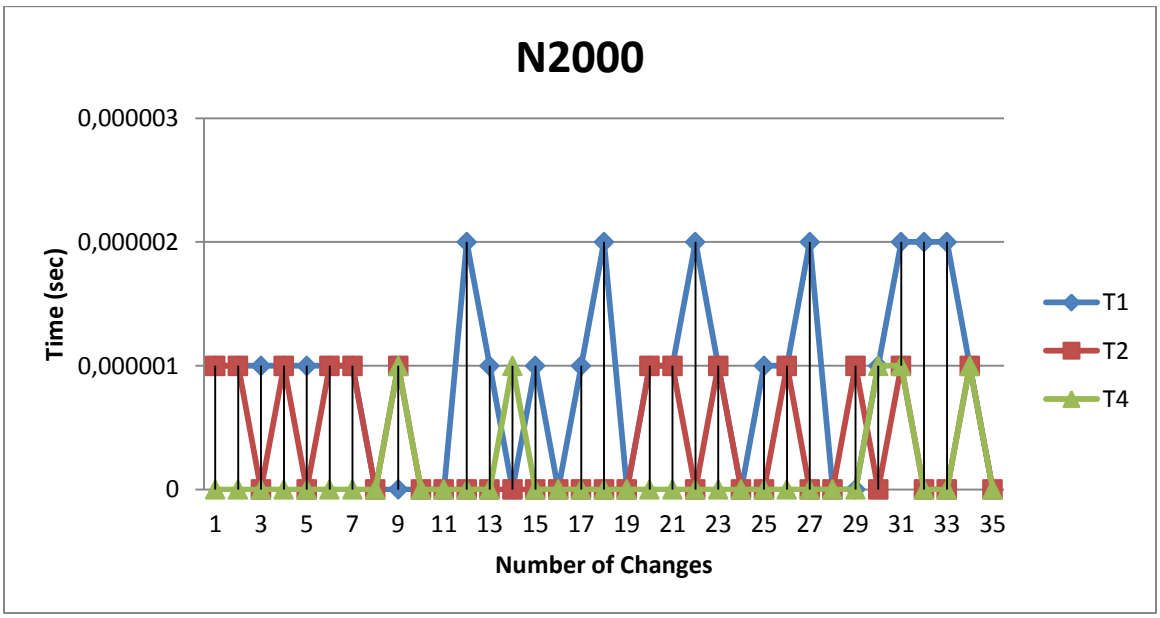
- 14 φορές σε χρόνο 1μsec για την υλοποίηση με 1 thread
- 1 φορά σε χρόνο 2μsec για την υλοποίηση με 1 thread
- 12 φορές σε χρόνο 1μsec για την υλοποίηση με 2 threads
- 4 φορά σε χρόνο 1μsec για την υλοποίηση με 4 threads

Τις υπόλοιπες φορές, αναγνωρίστηκε σε χρόνο πολύ μικρότερο του 1μsec ο οποίος θεωρείται αμελητέος και γι' αυτό σημειώνεται ως 0 στο διάγραμμα.

4.9 Μέτρηση 9η

Σήματα	Threads	Σήματα που άλλαξαν
2000	1,2,4	35,35,35

Όπως φαίνεται στο παρακάτω διάγραμμα, για αριθμό 2000 σημάτων και για 1, 2 και 4 threads, παρατηρήθηκε εναλλαγή σήματος 35 φορές και στις τρεις περιπτώσεις.



Η αλλαγή αναγνωρίστηκε :

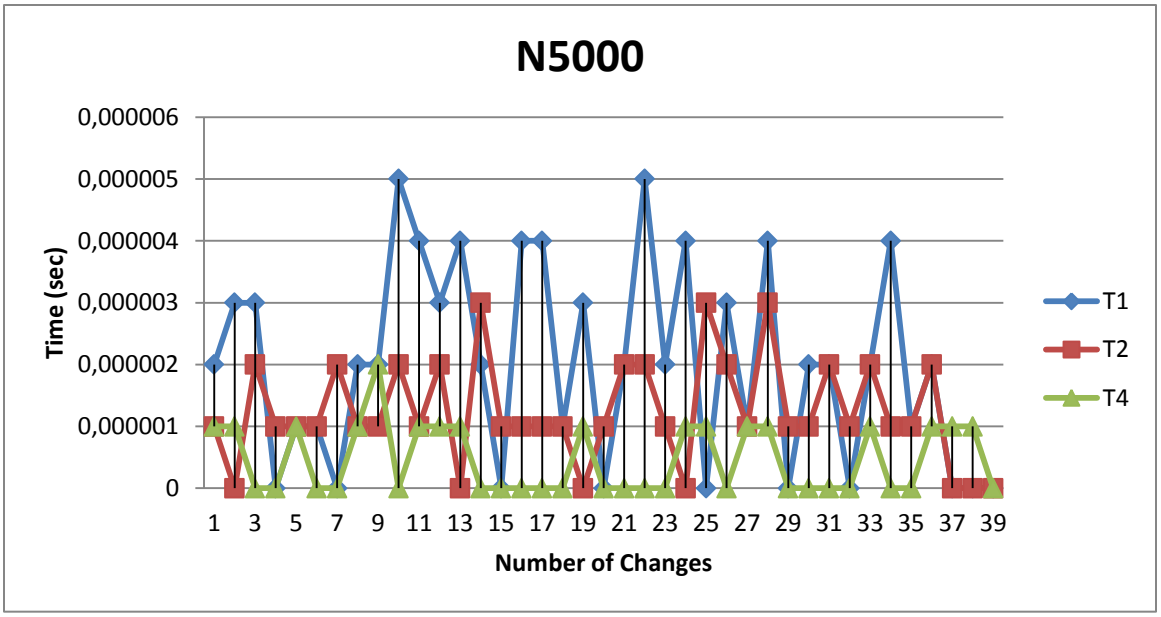
- 17 φορές σε χρόνο 1μsec για την υλοποίηση με 1 thread
- 7 φορές σε χρόνο 2μsec για την υλοποίηση με 1 thread
- 13 φορές σε χρόνο 1μsec για την υλοποίηση με 2 threads
- 5 φορά σε χρόνο 1μsec για την υλοποίηση με 4 threads

Τις υπόλοιπες φορές, αναγνωρίστηκε σε χρόνο πολύ μικρότερο του 1μsec ο οποίος θεωρείται αμελητέος και γι' αυτό σημειώνεται ως 0 στο διάγραμμα.

4.10 Μέτρηση 10η

Σήματα	Threads	Σήματα που άλλαξαν
5000	1,2,4	36,36,39

Όπως φαίνεται στο παρακάτω διάγραμμα, για αριθμό 5000 σημάτων και για 1, 2 και 4 threads, παρατηρήθηκε εναλλαγή σήματος 36, 36 και 39 φορές αντίστοιχα.



Η αλλαγή αναγνωρίστηκε :

- 5 φορές σε χρόνο 1μsec για την υλοποίηση με 1 thread
- 10 φορές σε χρόνο 2μsec για την υλοποίηση με 1 thread
- 5 φορές σε χρόνο 3μsec για την υλοποίηση με 1 thread
- 7 φορές σε χρόνο 4μsec για την υλοποίηση με 1 thread
- 2 φορές σε χρόνο 5μsec για την υλοποίηση με 1 thread
- 19 φορές σε χρόνο 1μsec για την υλοποίηση με 2 threads
- 10 φορές σε χρόνο 2μsec για την υλοποίηση με 2 threads
- 3 φορές σε χρόνο 3μsec για την υλοποίηση με 2 threads
- 16 φορές σε χρόνο 1μsec για την υλοποίηση με 4 threads
- 1 φορά σε χρόνο 2μsec για την υλοποίηση με 4 threads

Τις υπόλοιπες φορές, αναγνωρίστηκε σε χρόνο πολύ μικρότερο του 1μsec ο οποίος θεωρείται αμελητέος και γι' αυτό σημειώνεται ως 0 στο διάγραμμα.

4.11 Συμπεράσματα

Στο πρόβλημα μας θελήσαμε να εξετάσουμε την χειρότερη/οριακή συμπεριφορά του υπολογιστή μας να αναγνωρίζει σήματα μέχρι το όριο του ενός μsec.

Πραγματοποιήθηκαν πολλές μετρήσεις σε πολλά σενάρια με διάφορα threads αναγνώρισης σήματος με σκοπό να μοντελοποιηθεί όσο γίνεται η τυχαιότητα της εμφάνισης σημάτων.

Το συμπέρασμα βάσει των παραπάνω αποτελεσμάτων είναι πως το σύστημα με 1 thread αναγνώρισης κατάφερε να ανταποκριθεί πολύ καλά σε είσοδο με μικρό αριθμό σημάτων. Παρόλα αυτά, και με μέτριο αριθμό σημάτων, περί τα 600 σήματα, τα αποτελέσματα ήταν αρκετά ικανοποιητικά.

Όσο ανέβαινε ο αριθμός της εισόδου των σημάτων (π.χ. στα 900 σήματα) παρατηρήθηκε ότι η συχνότητα εμφάνισης αναγνώρισης σε 1μsec αυξήθηκε κατακόρυφα. Σε εκείνο το σημείο παρατηρούμε και την χρησιμότητα του multithreading, όπου με 1 thread υπήρξε 13 φορές η εμφάνιση της οριακής τιμής ενώ με 2 και 4 threads, η εμφάνιση της μειώθηκε σε 2 και 1 φορές αντίστοιχα.

Οι μετρήσεις με είσοδο 1200 και 1500 σήματα μας αποδεικνύουν ότι στο διάστημα αυτό το σύστημα μας φτάνει και ξεπερνάει την οριακή τιμή του 1μsec με την χρήση 1 thread. Καθώς αυξάνονται τα threads σε 2 και 4, το σύστημα επανέρχεται σε αποδεκτές τιμές.

Τέλος, οι μετρήσεις στην περιοχή των 2000 και 5000 σημάτων, επαληθεύουν την παραπάνω εκτίμηση. Παρατηρείται ότι το σύστημα για να μην ξεπεράσει το 1μsec, μπορεί να διαχειρίζεται ως είσοδο ελάχιστα πάνω από 1200 σήματα σε κάθε thread αναγνώρισης. Οτιδήποτε πάνω από αυτό το όριο, έχει ως αποτέλεσμα χρόνου από 2μsec και πάνω.

Στον κώδικα της υλοποίησης επιλέχθηκε να μπορεί ο χρήστης να επιλέγει ο ίδιος τον αριθμό των σημάτων και εν συνεχεία τον αριθμό των threads αναγνώρισης. Αυτό γιατί αφενός θέλαμε να βρεθεί μια στατιστική σχέση μεταξύ των σημάτων και των threads αναγνώρισης και αφετέρου διότι κάθε υπολογιστής έχει διαφορετικά χαρακτηριστικά.

Το αποτέλεσμα ήταν να δημιουργηθεί, για τον συγκεκριμένο υπολογιστή που υλοποιήθηκε η εργασία, ένα μοντέλο έξυπνης και αυτοματοποιημένης επιλογής του αριθμού των threads αναγνώρισης ανάλογα με τον αριθμό των σημάτων που

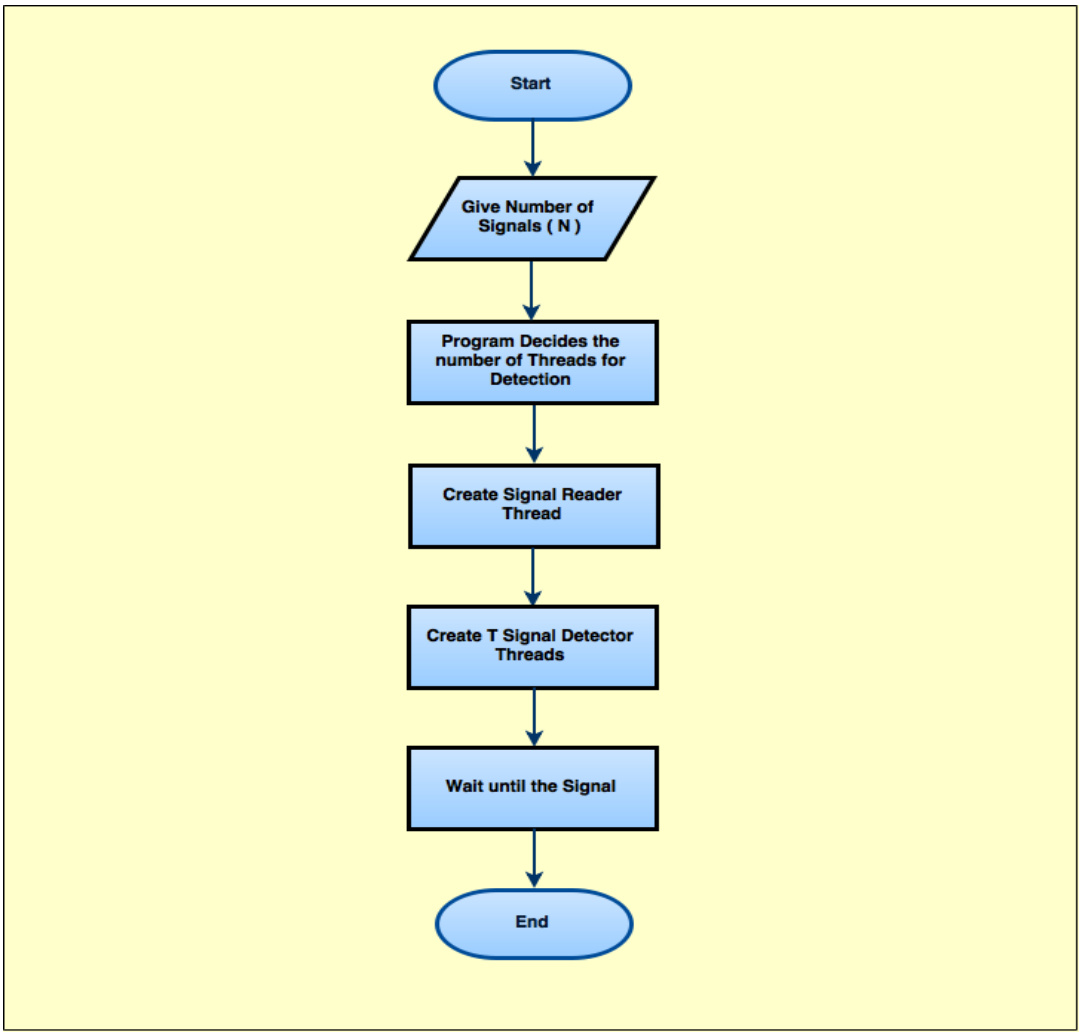
εισέρχονται στο σύστημα. Ο κώδικας αυτός που μπορεί να ενσωματωθεί στο πρόγραμμα μας (αντί για την χειροκίνητη επιλογή) φαίνεται παρακάτω :

```
if (N>=1 && N<=600){
    T=1;
}
if (N>600 && N<=1100){
    T=2;
}
if (N>1100 && N<=4400){
    T=4;
}

if (N>4400){
    printf ("\n\n current system cant handle over 4400 Signals in order
to achieve time lower than 1usec\n\n  ");
    return(1);
}
```

Η επιλογή των 1100 και 4400 σημάτων έγινε διότι θέλαμε να δώσουμε κάποιο περιθώριο από την κρίσιμη τιμή των περίπου 1200 σημάτων.

Το νέο διάγραμμα ροής :



Είναι σημαντικό να αναφερθεί ξανά πως στον χρόνο αναγνώρισης και επιλογής των threads αναγνώρισης σημαντικό ρόλο παίζουν και τα χαρακτηριστικά του υπολογιστή, τα οποία και αναφέρονται στην αρχή του παρόντος εγγράφου.

5.Συχνότερη Αλλαγή Σήματος–Τελικό Συμπέρασμα

Στην περίπτωση που αλλάξουμε το usleep(t*100000), για παράδειγμα το μειώσουμε στο μισό, δηλαδή usleep(t*50000), τότε τα σήματα που αλλάζουν συνολικά αλλά και ειδικά από 0 σε 1 διπλασιάζονται.

Πιο συγκεκριμένα, όσο περισσότερο μειώνουμε το usleep, τόσο περισσότερα σήματα αλλάζουν και ως αποτέλεσμα αυξάνεται ο χρόνος αναγνώρισης.

Μπορεί το σύστημα ως έχει να λειτουργεί σωστά για είσοδο έως 1200 σήματα ανά thread αναγνώρισης, αλλά αυτό δεν σημαίνει ότι και τα 1200 σήματα θα αλλαχθούν και θα αναγνωρισθούν, διότι ο χρόνος που εκτελείται το πρόγραμμα είναι 20 δευτερόλεπτα. Αν αναλογιστούμε ότι ένα σήμα δεν αλλάζει σε λιγότερο από 1/10 του δευτερολέπτου τότε ο αριθμός επεξεργασίας, αλλαγής και αναγνώρισης είναι πολύ μικρότερος του 1200.