

Διπλωματική Εργασία

Σχεδίαση γραφικών διεπαφών χρήστη εργαλείων CAD



**Καραγκεντσιδου Μαρία
Α.Ε.Μ.: 1700049**

**Τμήμα Μηχανικών Η/Υ Τηλεπικοινωνιών και
Δικτύων
Πανεπιστήμιο Θεσσαλίας**

Ευχαριστώ τον επιβλέποντα καθηγητή μου κ. Γεώργιο Σταμούλη για τη βοήθεια και καθοδήγηση καθ'όλη τη διάρκεια εκπόνησης της παρούσας διπλωματικής. Επίσης ευχαριστώ το Δημήτρη Μπουντά για το χρόνο που διέθεσε, την καθοδήγηση, τις ιδέες, τις συμβουλές και παντός είδους βοήθεια που μου παρείχε. Τέλος θα ήθελα να ευχαριστήσω τους Δημήτρη Συρίβελι, Γιώργο Μουρατίδη και Θεόφιλο Νικολάου για τη βοήθεια που μου προσέφεραν.



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΥΠΗΡΕΣΙΑ ΒΙΒΛΙΟΘΗΚΗΣ & ΠΛΗΡΟΦΟΡΗΣΗΣ
ΕΙΔΙΚΗ ΣΥΛΛΟΓΗ «ΓΚΡΙΖΑ ΒΙΒΛΙΟΓΡΑΦΙΑ»**

Αριθ. Εισ.: 3465/1
Ημερ. Εισ.: 11-05-2006
Δωρεά: Συγγραφέα
Ταξιθετικός Κωδικός: ΠΤ- ΜΗΥΤΔ
2005
ΚΑΡ

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. Εισαγωγή.....	5
1.1 Αντικείμενο εργασίας.....	5
1.2 Διάρθρωση εργασίας.....	5
2. Περιγραφή του προγράμματος.....	7
2.1 Κατανάλωση ισχύος σε ολοκληρωμένα κυκλώματα.....	7
2.2 Λειτουργία του προγράμματος.....	9
2.3 Στατική ισχύς.....	9
2.3.1 Ορισμός στατικής ισχύος.....	9
2.3.2 Υπολογισμός στατικής ισχύος μέσω προσομοίωσης.....	10
2.4 Δυναμική ισχύς.....	10
2.4.1 Ορισμός δυναμικής ισχύος.....	10
2.5 Single Event Upset.....	12
2.5.1 Ορισμός Single event upset.....	12
2.5.2 Μελέτη συμπεριφοράς του κυκλώματος σε single event upset μέσω προσομοίωσης.....	13
2.6 Multi Level Function.....	13
3. Γραφικά περιβάλλοντα χρήστη.....	15
3.1 Ορισμός.....	15
3.2 Οργάνωση των γραφικών διεπαφών.....	15
3.3 Πλεονεκτήματα.....	16
3.4 Κανόνες σχεδίασης γραφικών διεπαφών.....	17
4. Παρουσίαση Γραφικού Περιβάλλοντος Διασύνδεσης.....	22
4.1 Περιγραφή κυρίου παράθυρου.....	22
4.2 Τρόποι επιλογής λειτουργίας & αναλυτική περιγραφή των λειτουργιών.....	24
4.2.1 Μενού.....	24
4.2.2 Συνδυασμοί πλήκτρων.....	34
4.2.3 Γραμμή εργαλείων (Tool Bar).....	35
4.2.4 Επιπλέον πληροφορίες για κάθε πύλη.....	36
5. Αναπαράσταση ψηφιακών κυκλωμάτων.....	38
5.1 Συνοπτική περιγραφή.....	38
5.2 Διαδικασία απεικόνισης ψηφιακών κυκλωμάτων.....	39
5.2.1 Δομές οι οποίες χρησιμοποιήθηκαν κατά τη σχεδίαση.....	39
5.2.2 Αναλυτική περιγραφή της διαδικασίας που ακολουθείται κατά τη σχεδίαση του κυκλώματος.....	40
5.2.3 Ακολουθιακά κυκλώματα.....	46
6. Σύγκριση μεταξύ γλωσσών προγραμματισμού και επιλογή της καταλληλότερης.....	49
6.1 Java.....	49
6.1.1 Η βιβλιοθήκη AWT.....	50
6.1.2 Η Βιβλιοθήκη Swing.....	51
6.1.3 AWT vs Swing.....	52
6.1.4 Συμπεράσματα.....	53
6.2 Visual Basic.....	53
6.3 Delphi.....	54
6.4 C++.....	55

6.4.1 Visual C++	56
6.4.2 Qt designer	56
6.4.3 Qt designer vs Visual C++	58
6.5 Συνολική σύγκριση των προαναφερόμενων γλωσσών και εργαλείων	59
<i>Java (Χρήση Swing με την υποστήριξη του AWT)</i>	59
<i>Visual Basic</i>	60
<i>Delphi</i>	60
<i>Visual C++</i>	60
<i>Qt designer</i>	61
6.6 Τελική επιλογή γλώσσας προγραμματισμού	61
<i>Παράρτημα Α: 4 Προειδοποιητικά μηνύματα</i>	63
<i>Παράρτημα Β: Μηνύματα που εμφανίζονται στο παράθυρο "Result"</i>	65
<i>Παράρτημα Γ: Θέματα Βοήθειας</i>	68
<i>Αναφορές</i>	70

1.Εισαγωγή

Αρχικά ο χρήστης επικοινωνούσε με τον υπολογιστή μέσω του λειτουργικού συστήματος. Αργότερα έκαναν την εμφάνισή τους τα γραφικά περιβάλλοντα για κάθε πρόγραμμα έτσι ώστε αυτή η επικοινωνία να διευκολυνθεί αρκετά. Αυτού του είδους οι εφαρμογές εδραιώθηκαν με την εμφάνιση και την εκτεταμένη χρήση των Windows και στις μέρες μας πλέον σχεδόν δεν υπάρχει κανένα πρόγραμμα χωρίς γραφικό περιβάλλον.

1.1 Αντικείμενο εργασίας

Σκοπός της εν λόγω πτυχιακής εργασίας είναι η ανάπτυξη ενός γραφικού περιβάλλοντος διασύνδεσης για το προαναφερόμενο πρόγραμμα. Το περιβάλλον αυτό σχεδιάστηκε με στόχο την ευκολότερη χρήση του προγράμματος. Πιο συγκεκριμένα σχεδιάστηκε για να προσφέρει στους χρήστες έναν πιο εποπτικό και φιλικό προς αυτούς τρόπο για την επικοινωνία τους με τον υπολογιστή.

Αναλυτικότερα η εργασία αυτή αφορούσε αρχικά στην επιλογή της γλώσσας η οποία επρόκειτο να χρησιμοποιηθεί για την υλοποίηση του σκοπού μας και στη συνέχεια στη σχεδίαση και ανάπτυξη του εν λόγω γραφικού περιβάλλοντος. Το τελευταίο έπρεπε να υλοποιηθεί κατά τέτοιο τρόπο, ώστε να εμφανίζονται στην οθόνη το περιγραφόμενο από το αρχείο spice κύκλωμα, καθώς και τα αποτελέσματα της επιλεγόμενης κάθε φορά λειτουργίας. Επιπλέον κάνοντας κλικ με το ποντίκι πάνω σε κάποια πύλη του κυκλώματος έπρεπε να εμφανίζονται πληροφορίες σχετικά με τη συγκεκριμένη πύλη.

1.2 Διάρθρωση εργασίας

Η διάρθρωση της εργασίας έχει ως εξής.

Το δεύτερο κεφάλαιο ξεκινά με μια σύντομη αναφορά για την κατανάλωση ισχύος σε κυκλώματα VLSI και συνεχίζεται με μια αναλυτική περιγραφή του ήδη υπάρχοντος προγράμματος. Πιο συγκεκριμένα εξηγούνται επαρκώς οι λειτουργίες τις

οποιές το πρόγραμμα επιτελεί και αναφέρονται κάποια στοιχεία για τον τρόπο με τον οποίο αυτές είναι υλοποιημένες.

Το τρίτο κεφάλαιο είναι αφιερωμένο στις γραφικές διεπαφές. Πιο αναλυτικά παρουσιάζεται η δομή και η οργάνωση ενός τυπικού γραφικού περιβάλλοντος και κατόπιν σχολιάζουμε τους κανόνες που πρέπει να διέπουν αυτού του είδους τις εφαρμογές.

Το επόμενο κεφάλαιο έχει ως στόχο να περιγράψει τη γραφική διεπαφή που σχεδιάσαμε, το πώς εκτελούνται οι διάφορες λειτουργίες και το πώς επιτυγχάνεται η αλληλεπίδραση μεταξύ χρήστη και υπολογιστή.

Στο πέμπτο κεφάλαιο περιγράφεται ο τρόπος με τον οποίο εμφανίζονται τα ψηφιακά κυκλώματα στην οθόνη, ενώ παρουσιάζεται αναλυτικά και ο τρόπος υλοποίησης της διαδικασίας απεικόνισης των κυκλωμάτων.

Στο έκτο και τελευταίο κεφάλαιο της παρούσας εργασίας γίνεται μια παρουσίαση των γλωσσών προγραμματισμού που εξετάσαμε πριν καταλήξουμε στη συγκεκριμένη, καθώς και μια σύγκρισή μεταξύ των, η οποία μας βοήθησε να κατασταλάξουμε στην τελική μας επιλογή.

2.Περιγραφή του προγράμματος

Για να γίνουν πλήρως κατανοητές οι λειτουργίες του προγράμματός μας, αρχικά παραθέτουμε κάποια θεωρητικά στοιχεία για την κατανάλωση ισχύος σε ολοκληρωμένα κυκλώματα, και στη συνέχεια συνεχίζουμε με περιγραφή των λειτουργιών και συνοπτική περιγραφή του τρόπου υλοποίησης.

2.1 Κατανάλωση ισχύος σε ολοκληρωμένα κυκλώματα

Ήδη γνωρίζουμε ότι η σχεδίαση ενός ψηφιακού κυκλώματος πραγματοποιείται σε τέσσερα κύρια επίπεδα: το επίπεδο πύλης, το επίπεδο φυσικού σχεδίου, το επίπεδο κυκλώματος ή τρανζίστορ, και το φυσικό επίπεδο. Στόχος της σχεδίασης του κυκλώματος στα τέσσερα αυτά στάδια είναι η ελαχιστοποίηση της καθυστέρησης, της επιφάνειας που καταλαμβάνει το κύκλωμα και της κατανάλωσης ισχύος. Εμείς θα ασχοληθούμε με την τρίτη παράμετρο.

Ξεκινάμε με την υπόθεση ότι η πηγή ρεύματος είναι ιδανική, πράγμα το οποίο σημαίνει ότι μπορεί να δώσει οποιαδήποτε στιγμή το απαιτούμενο για τη λειτουργία του κυκλώματος ρεύμα, χωρίς να μεταβληθεί η τάση τροφοδοσίας. Έτσι η ισχύς του κυκλώματος για κάθε χρονική στιγμή θα δίνεται από τον τύπο:

$$P(t) = V_{DD} * I(t).$$

Το μεγαλύτερο μέρος της διαδικασίας διεξάγεται στα δύο ενδιάμεσα επίπεδα σχεδίασης του κυκλώματος, Μπορούμε λοιπόν να υποθέσουμε ότι το σύνολο των πυλών του κυκλώματος τροφοδοτούνται από την ίδια σταθερή τάση. Το γεγονός αυτό σε συνδυασμό με το ότι οι πύλες συνδέονται παράλληλα με τους αγωγούς τροφοδοσίας και γείωσης, και κατά συνέπεια, το συνολικό ρεύμα θα είναι ίσο με το άθροισμα του ρεύματος κάθε πύλης, μας οδηγεί στο συμπέρασμα ότι η στιγμιαία ισχύς ενός κυκλώματος που αποτελείται από n πύλες θα δίνεται από τον παρακάτω τύπο:

$$P(t) = V_{DD} * \sum_{i=1}^n I_i(t)$$

Από τη στιγμιαία ισχύ μπορούμε να εξάγουμε τον τύπο της μέσης ισχύος δηλαδή της ισχύος που καταναλώνει το κύκλωμα σε ένα χρονικό διάστημα T . Η μέση ισχύς είναι ίση με τη μεσή τιμή των τιμών της στιγμιαίας ισχύος για κάθε χρονική μονάδα του διαστήματος T . Ο τύπος ο οποίος την περιγράφει είναι ο ακόλουθος:

$$P_T = (1/T) * \int_0^T P(t)dt = (V_{DD}/T) \int_0^T I(t)dt$$

Η κατανάλωση ισχύος σε ένα ολοκληρωμένο κύκλωμα αποτελείται από τρεις παραμέτρους: την ισχύ μεταγωγής, την ισχύ βραχυκυκλώματος και την ισχύ διαρροής. Οι δύο πρώτες εμφανίζονται μονάχα κατά το διάστημα της αλλαγής κατάστασης του κυκλώματος, ενώ η τρίτη εμφανίζεται πάντα.

Λόγω των δυο πρώτων συνιστωσών για τον προσδιορισμό της ισχύος από τις συγκεκριμένες σχέσεις είναι απαραίτητο να γνωρίζουμε και τα διανύσματα εισόδου του κυκλώματος τη συγκεκριμένη χρονική στιγμή ή το συγκεκριμένο χρονικό διάστημα. Από τις προηγούμενες σχέσεις είναι δυνατός ο ορισμός για κάθε κύκλωμα δύο τιμών ισχύος ανεξάρτητων του χρόνου, οι οποίες καλύπτουν όλο το σύνολο των εισόδων: η μέγιστη ισχύς P_{mx} και η μέση ισχύς P_{av} .

Ως μέση ισχύς ενός κυκλώματος θεωρούμε τη μέση ισχύ που προκύπτει από τον προηγούμενο τύπο για ένα διάστημα το οποίο καλύπτει όλες τις δυνατές μεταβολές μεταξύ των διανυσμάτων εισόδου. Μπορούμε να θεωρήσουμε ότι δίνεται από την παρακάτω σχέση:

$$P_{av} = \lim_{T \rightarrow \infty} P_T = \lim_{T \rightarrow \infty} (1/T) * \int_0^T P(t)dt$$

Όπως εύκολα μπορεί να διαπιστώσει κανείς η μέγιστη ισχύς ενός κυκλώματος είναι ίση με τη μέγιστη τιμή της στιγμιαίας ισχύος. Δηλαδή:

$$P_{mx} = \max_{t \in R} P(t)$$

2.2 Λειτουργία του προγράμματος

Το όλο πρόγραμμα αποτελεί έναν προσομοιωτή ψηφιακών κυκλωμάτων. Η είσοδος του είναι spice αρχεία. Αφού δεχτεί ένα τέτοιο αρχείο ως είσοδο το πρόγραμμα κατασκευάζει το γράφο του κυκλώματος και αφού εκτελέσει κάποιους υπολογισμούς, επιστρέφει αποτελέσματα σχετικά με την κατανάλωση στατικής και δυναμικής ισχύος, καθώς και με τη συμπεριφορά του κυκλώματος σε περιπτώσεις που συμβεί κάποιο single event upset. Επίσης υπολογίζει τη συνάρτηση κόμβων για ένα ή περισσότερα επίπεδα.

2.3 Στατική ισχύς

2.3.1 Ορισμός στατικής ισχύος

Όταν μιλάμε για στατική ισχύ αναφερόμαστε στην ισχύ που καταναλώνει μια πύλη όταν δεν αλλάζει η λογική τιμή της εξόδου της. Αιτία κατανάλωσης στατικής ισχύος στα κυκλώματα CMOS VLSI, αποτελεί η αγωγή υπό κατωφλίου μεταξύ πηγής και υποδοχής. Πιο συγκεκριμένα κατανάλωση στατικής ισχύος εμφανίζεται λόγω αγωγής της ανάστροφα πολωμένης διόδου, μεταξύ των στρωμάτων διαχύσεως και υποστρώματος.

Η συνολική στατική ισχύς που καταναλώνεται σε ένα κύκλωμα είναι ίση με το άθροισμα της κατανάλωσης στατικής ισχύος όλων των κελιών που έχουν χρησιμοποιηθεί στο συγκεκριμένο κύκλωμα. Ο τύπος που χρησιμοποιείται για τον υπολογισμό είναι ο παρακάτω:

$$P_{\text{Leakage Total}} = \sum_{\forall \text{cells}(i)} P_{\text{Cell Leakage } i}$$

όπου:

$P_{\text{Leakage Total}}$ = συνολική κατανάλωση στατικής ισχύος για το κύκλωμα

$P_{\text{Cell Leakage } i}$ = κατανάλωση ισχύος διαρροής για το κάθε κελί

Έχει υπολογιστεί ότι για κυκλώματα τα όποια τον περισσότερο χρόνο είναι ενεργά, η ισχύς διαρροής που τους αντιστοιχεί είναι μικρότερη από το 1% της συνολικής κατανάλωσης ισχύος. Αντίθετα για κυκλώματα που τον περισσότερο χρόνο παραμένουν ανενεργά, το ποσό της ισχύος διαρροής που καταναλώνεται είναι σημαντικό.

2.3.2 Υπολογισμός στατικής ισχύος μέσω προσομοίωσης

Ξεκινάμε με τυχαία αρχικοποίηση των εισόδων. Κατόπιν υπολογίζουμε την κατάσταση των κόμβων που ανήκουν στο πρώτο level και συνεχίζουμε με τον υπολογισμό της κατάστασης των κόμβων όλων των επιπέδων μέχρι να φτάσουμε στο τέλος. Από τον πίνακα κατανάλωσης στατικής ενέργειας της συγκεκριμένης πύλης, εξάγουμε το ποσό κατανάλωσης στατικής ισχύος για κάθε κόμβο. Προσθέτοντας λοιπόν τα ποσά στατικής ισχύος που καταναλώνονται σε κάθε κόμβο όλων των επιπέδων εξάγουμε τη συνολική κατανάλωση στατικής ισχύος του κυκλώματος. Για να πετύχουμε μια καλή προσέγγιση της ποσότητας αυτής επαναλαμβάνουμε την παραπάνω διαδικασία δίνοντας διαφορετικές τιμές στα διανύσματα εισόδου. Με τον τρόπο αυτό προσπαθούμε να καλύψουμε όσο το δυνατό περισσότερες καταστάσεις από αυτές στις οποίες μπορεί να επέλθει το κύκλωμα.

2.4 Δυναμική ισχύς

2.4.1 Ορισμός δυναμικής ισχύος

Ως δυναμική ισχύς θεωρείται η ισχύς που καταναλώνεται όταν μια πύλη είναι ενεργή. Μια πύλη θεωρείται ενεργή κάθε φορά που οι τάσεις των δικτύων της εναλλάσσονται σύμφωνα με τις τιμές των εισόδων του κυκλώματος. Η δυναμική ισχύς έχει δύο συνιστώσες την ισχύ λόγω μεταγωγής λογικής τιμής και την εσωτερική ισχύ.

Η κατανάλωση ισχύος λόγω μεταγωγής λογικής τιμής μιας πύλης, είναι αποτέλεσμα της φόρτισης και εκφόρτισης της χωρητικότητας φορτίου στην έξοδο της. Οι εν λόγω φορτίσεις και εκφορτίσεις λαμβάνουν χώρα εξαιτίας των λογικών μεταγωγών των τιμών της εξόδου. Άρα λοιπόν καταλήγουμε στο συμπέρασμα ότι η

ισχύς λόγω μεταγωγής λογικής τιμής εξαρτάται από τη συνολική χωρητικότητα φορτίου στην έξοδο της πύλης καθώς και από τη συχνότητα των εναλλαγών λογικών τιμών. Πολλά λογισμικά χρησιμοποιούν για τον υπολογισμό της ισχύος μεταγωγής τον παρακάτω τύπο:

$$P_C = (V_{DD}^2/2) * \sum_{\forall nets(1)} (C_{Loadi} \times TR_i)$$

όπου:

P_c = Ισχύς μεταγωγής του κυκλώματος

C_{Loadi} = Χωρητικότητα φορτίου του δικτύου i .

TR_i = Μέσος αριθμός μεταγωγών του δικτύου i μεταγωγές /sec

V_{DD} = Τάση τροφοδοσίας

Η κατανάλωση εσωτερικής ισχύος λαμβάνει χώρα μέσα στα όρια της πύλης. Το κύκλωμα καταναλώνει εσωτερική ισχύ, κατά τη διάρκεια μιας εναλλαγής λογικής τιμής, λόγω της φόρτισης και εκφόρτισης των υπάρχοντων εσωτερικών χωρητικοτήτων. Αυτού του είδους η κατανάλωση ισχύος είναι αποτέλεσμα ενός στιγμιαίου βραχυκυκλώματος μεταξύ του P και N τρανζίστορ της πύλης.

Η εσωτερική ισχύς που καταναλώνεται στα πλαίσια ενός κελιού είναι ίση με το άθροισμα των ισχύων που καταναλώνονται σε κάθε είσοδο και έξοδο του κελιού. Σε πολλά λογισμικά ο υπολογισμός της εσωτερικής ισχύος γίνεται από τους παρακάτω τύπους:

$$P_{int} = E_Z / TR_Z$$

$$E_Z = f [C_{LOAD}, WeightAvg_{(Trans)}]$$

$$WeightAvg_{(Trans)} = \frac{\sum_{i=A,B} (TR_i * Trans_i)}{\sum_{i=A,B} TR_i}$$

όπου:

A, B = οι είσοδοι του κελιού

Z = η έξοδος του κελιού

P_{int} = συνολική εσωτερική ισχύς του κελιού

E_Z = εσωτερική ενέργεια της εξόδου Z συναρτήσει των μεταγωγών της λογικής τιμής της εισόδου και του φορτίου εξόδου.

TR_Z = μέσος αριθμός μεταγωγών ακροδέκτη εξόδου Z .

TR_i = μέσος αριθμός μεταγωγών ακροδέκτη εισόδου i εναλλαγές / sec.

$Trans_i$ = χρόνος μεταγωγής εισόδου i

$WeightAvg_{(Trans)}$ = χρόνος μεταγωγής με βάρη της εξόδου Z .

Τα κελιά συνήθως καταναλώνουν διαφορετική εσωτερική ισχύ σε διαφορετικές χρονικές στιγμές. Η ισχύς που κάθε στιγμή καταναλώνεται εξαρτάται από την κατάσταση του κελιού, καθώς και από τη διαδρομή. Δηλαδή πιο συγκεκριμένα διαφορετικό ποσό ισχύος καταναλώνεται σε ένα κελί, όταν αυτό βρίσκεται σε κατάσταση ανάγνωσης και διαφορετικό όταν βρίσκεται σε κατάσταση εγγραφής. Ακόμα όταν αλλάζει η τιμή ενός σήματος, άλλο ποσό ισχύος καταναλώνεται όταν έχουμε αλλαγή από 0 σε 1 και άλλο όταν έχουμε αλλαγή από 1 σε 0. Όσον αφορά την εξάρτηση από τη διαδρομή τόσο περισσότερη ισχύς καταναλώνεται σε ένα κελί όσα περισσότερα επίπεδα λογικής επηρεάζει η αλλαγή σήματος στον ακροδέκτη.

2.5 Single Event Upset

2.5.1 Ορισμός Single event upset

Ως single event upset αναφέρεται το φαινόμενο της ξαφνικής ανεπιθύμητης αλλαγής της κατάστασης ενός κόμβου του κυκλώματος. Η αλλαγή αυτή μπορεί να είναι αποτέλεσμα διάφορων παραγόντων, όπως για παράδειγμα η αλληλεπίδραση με άλλους κόμβους, ή η επίδραση κάποιας ακτινοβολίας πάνω στο κύκλωμα. Ο τελευταίος παράγοντας είναι αυτός που αποτελεί τη συχνότερη αιτία εμφάνισης του εν λόγω φαινομένου.

Οι κυριότερες μορφές ακτινοβολίας που μπορούν να επηρεάσουν τη λειτουργία ενός κυκλώματος είναι η ακτινοβολία άλφα και η κοσμική ακτινοβολία. Κατά τη διάρκεια της επίδρασης της ακτινοβολίας στο κύκλωμα, ηλεκτρόνια θα συγκεντρωθούν στο N^+ drain, με αποτέλεσμα ο κόμβος να μεταβαίνει σε λογικό 0, αν

βρίσκεται σε λογικό 1. ταυτόχρονα οπές θα συγκεντρωθούν στο P+ drain, με αποτέλεσμα ο κόμβος να μεταβαίνει σε λογικό 1, αν βρίσκεται σε λογικό 0.

2.5.2 Μελέτη συμπεριφοράς του κυκλώματος σε single event upset μέσω προσομοίωσης

Τελευταία λειτουργία του προγράμματός μας είναι η μελέτη κυκλωμάτων σε περιπτώσεις που ένας κόμβος βρίσκεται σε κάποια κατάσταση, ενώ στην πραγματικότητα έπρεπε να βρίσκεται σε κάποια άλλη. Η μελέτη γίνεται πάνω σε ακολουθιακά κυκλώματα, και για να δούμε αν κάποιο single event upset επηρεάζει το κύκλωμα, ελέγχουμε τις εισόδους των DFFs.

Ο εν λόγω έλεγχος πραγματοποιείται ως εξής. Δίνουμε τυχαίες τιμές στο vector εισόδου και στη συνέχεια αρχικοποιούμε όλους τους κόμβους στις τιμές που πρέπει να έχουν για το συγκεκριμένο διάνυσμα εισόδου. Στη συνέχεια αλλάζουμε για κάθε κόμβο και εξετάζουμε πώς επιδρά η αλλαγή αυτή στο κύκλωμα. Πιο συγκεκριμένα αρχίζουμε να διαδίδουμε την αλλαγή αυτή προς τα εμπρός, έως ότου φτάσουμε στο τέλος του κυκλώματος. Στη συνέχεια εξετάζουμε τις εισόδους των DFFs και αν υπάρχει έστω και μια διαφορετική είσοδος σε σύγκριση με τις εισόδους πριν την αλλαγή της κατάστασης του κόμβου, θεωρούμε ότι η αλλαγή αυτή επηρεάζει το κύκλωμα. Στη συνέχεια αρχικοποιούμε το κύκλωμα και επαναλαμβάνουμε τη διαδικασία για όλους τους κόμβους.

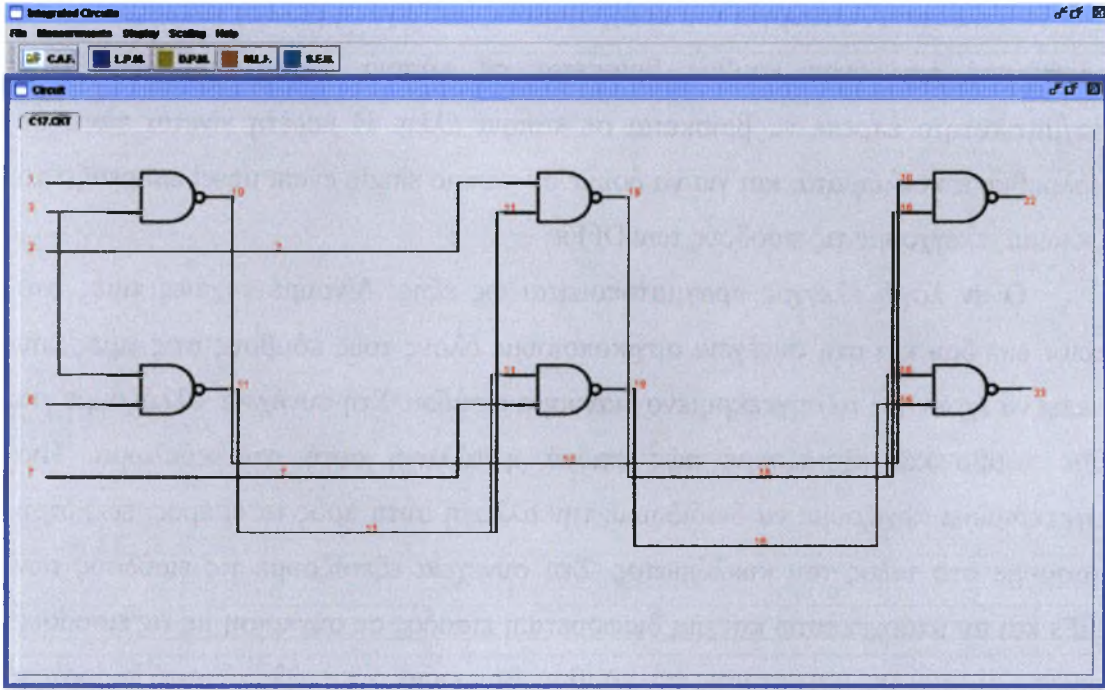
2.6 Multi Level Function

Στόχος της λειτουργίας αυτής είναι ο υπολογισμός της συνάρτησης ενός κόμβου για βάθος ενός ή πολλών επιπέδων. Δηλαδή αν επιλέξουμε έναν κόμβο ενός επιπέδου και ζητήσουμε υπολογισμό της συνάρτησης του κόμβου αυτού για βάθος δύο επιπέδων, το πρόγραμμα θα μας επιστρέψει μια συνάρτηση η οποία προκύπτει από τις πύλες δύο επιπέδων αν θεωρήσουμε τον κόμβο που έχουμε επιλέξει ως έξοδο της πύλης του τελευταίου επιπέδου.

Η διαδικασία που ακολουθούμε είναι η εξής: Ξεκινάμε από τον κόμβο που έχουμε επιλέξει και διατρέχουμε το κύκλωμα προς τα πίσω, υπολογίζοντας τη συνάρτηση κάθε κόμβου που συναντούμε μέχρι να φτάσουμε στο απαιτούμενο βάθος.

Στη συνέχεια αντικαθιστούμε κάθε κόμβο με τη συνάρτησή του, στη συνάρτηση του επιλεγόμενου κόμβου και βρίσκουμε το ζητούμενο αποτέλεσμα.

Θα δείξουμε τη διαδικασία την οποία ακολουθούμε μέσω ενός παραδείγματος. Έστω ότι έχουμε το κύκλωμα της παρακάτω εικόνας και θέλουμε να βρούμε τη συνάρτηση του κόμβου 16 για βάθος ίσο με δύο επίπεδα.



Η συνάρτηση του κόμβου 16 σε βάθος ενός επιπέδου είναι ίση με τη συνάρτηση της πύλης η οποία είναι NAND οπότε $\text{NOT}((2)(11)) = \text{NOT}(2) + \text{NOT}(11)$. Στη συνέχεια βρίσκουμε τις συναρτήσεις για τις εισόδους της πύλης αυτής. Η είσοδος 2 είναι είσοδος του κυκλώματος. Η 11 είναι έξοδος μιας πύλης NAND, η οποία έχει εισόδους την 3 και την 6. Άρα η συνάρτηση του κόμβου 11 είναι: $\text{NOT}((3)(6)) = \text{NOT}(3) + \text{NOT}(6)$. Επομένως η συνάρτηση του κόμβου 16 γίνεται: $\text{NOT}(2) + \text{NOT}(\text{NOT}(3) + \text{NOT}(6)) = \text{NOT}(2) + (3)(6)$.

3. Γραφικά περιβάλλοντα χρήστη

3.1 Ορισμός

Ως γραφική διεπαφή χρήστη ορίζεται εκείνο το τμήμα ενός προγράμματος το οποίο είναι απαραίτητο έτσι ώστε να παρέχονται οι διάφορες λειτουργίες του προγράμματος στο χρήστη με τον απλούστερο και ευκολότερο τρόπο. Μια τέτοια διεπαφή μπορεί να είναι εξαιρετικά απλή, έτσι ώστε απλά να εξυπηρετεί το στόχο για τον οποίο έχει σχεδιαστεί, αλλά μπορεί να είναι και αρκετά πολύπλοκη με πολλά επιπλέον γραφικά, εικόνες, ήχους κλπ, έτσι ώστε η σχεδίασή της να απαιτεί μεγάλο κόπο και χρόνο, καθώς και αρκετές γραμμές κώδικα.

Τα κυριότερα χαρακτηριστικά μιας γραφικής επαφής και αυτά που εμφανίζονται πάντα σε αυτές είναι:

- 1) η γραμμή τίτλου**
- 2) η γραμμή των μενού**
- 3) η γραμμή των εργαλείων**
- 4) η γραμμή κατάστασης**
- 5) οι μπάρες κύλισης**
- 6) τα γραφικά αντικείμενα**
- 7) οι κέρσορες (του ποντικιού και του κειμένου)**

3.2 Οργάνωση των γραφικών διεπαφών

Μια γραφική διεπαφή συνήθως αποτελείται από ένα σύνολο συστατικών τοποθετημένων σε έναν υποδοχέα. Η ονομασία και ο αριθμός των συστατικών που μπορούν να χρησιμοποιηθούν ως υποδοχείς ποικίλλει ανάλογα με το εργαλείο το οποίο χρησιμοποιείται. π.χ. στη Delphi, ο υποδοχέας αναφέρεται ως φόρμα, ενώ στη java ο υποδοχέας μπορεί να είναι είτε ένα panel, είτε ένα frame.

Ως συστατικά ενός γραφικού περιβάλλοντος αναφέρονται τα αντικείμενα εκείνα μέσω των οποίων επιτυγχάνεται η αλληλεπίδραση του χρήστη με τον υπολογιστή με τη βοήθεια του ποντικιού ή του πληκτρολογίου. Δηλαδή συστατικά αποτελούν τα κουμπιά, τα διάφορα μενού κλπ. Καθένα από αυτά μπορεί να έχει διάφορα

χαρακτηριστικά, όπως το μέγεθος, τη θέση πάνω στον υποδοχέα, ή τον τρόπο με τον οποίο ανταποκρίνονται στα διάφορα γεγονότα.

Ως γεγονότα μπορούν να θεωρηθούν το πάτημα ενός κουμπιού, η επιλογή ενός αντικειμένου μενού από ένα μενού κ.α. Κάθε γλώσσα έχει διαφορετικούς τρόπους χειρισμού των γεγονότων αυτών. Το σίγουρο είναι πάντως ότι μόλις συμβεί κάποιο από αυτά το πρόγραμμα ανταποκρίνεται εκτελώντας την καθορισμένη για αυτό το γεγονός λειτουργία. π.χ. με το πάτημα ενός συγκεκριμένου κουμπιού, να εμφανίζεται στην οθόνη ένα παράθυρο. Οι ενέργειες που θα εκτελεστούν με την εμφάνιση κάποιου γεγονότος καθορίζονται συνήθως από το χρήστη.

Με λίγα λόγια δηλαδή θα μπορούσαμε να πούμε ότι ένα γραφικό περιβάλλον αποτελείται από τα συστατικά του, δηλαδή όλα εκείνα που διευκολύνουν την αλληλεπίδραση του προγράμματος με το χρήστη, όπως π.χ. τα κουμπιά ή τα μενού. Αλλαγή στο γραφικό περιβάλλον ή εκτέλεση κάποιας άλλης λειτουργίας λαμβάνει χώρα μόλις συμβεί κάποια ενέργεια. Παραδείγματα ενεργειών αποτελούν το πάτημα ενός κουμπιού ή το κλικ ενός ποντικιού.

3.3 Πλεονεκτήματα

Όπως είδαμε τα γραφικά περιβάλλοντα διασύνδεσης προσφέρουν έναν πιο εύκολο τρόπο αλληλεπίδρασης μεταξύ χρήστη και υπολογιστικού συστήματος. Παρά το γεγονός ότι η λειτουργία των προγραμμάτων μπορεί να εκτελεστεί και χωρίς την ύπαρξή τους, είναι πολύ χρήσιμα καθώς κάνουν το περιβάλλον των προγραμμάτων φιλικότερο προς αυτούς που το χρησιμοποιούν.

Η εκτεταμένη χρήση αντικειμένων ελέγχου π.χ. κουμπιών βοηθά το χρήστη να κάνει τις επιλογές του και να εισάγει δεδομένα στον υπολογιστή ευκολότερα ενώ ταυτόχρονα μειώνεται η πιθανότητα για λανθασμένων επιλογών. Για να αναφέρουμε ένα παράδειγμα αρκεί κανείς να σκεφτεί κάποιο χρήστη ο οποίος θέλει να επιλέξει μια λειτουργία από τις πολλές που προσφέρει ένα πρόγραμμα. Είναι πιο προσιτό για αυτόν να τη διαλέξει επιλέγοντας ένα κουμπί από ένα μενού ή πιέζοντας έναν συνδυασμό πλήκτρων, παρά να την επιλέξει μέσα από το λειτουργικό σύστημα δίνοντας ως είσοδο κάποιο γράμμα ή κάποια λέξη που να την αντιπροσωπεύει. Επιπλέον είναι πιο εύκολο να κάνει λάθος τη στιγμή που δηλώνει στο λειτουργικό ποια λειτουργία θέλει να εκτελεστεί, παρά τη στιγμή που πατάει ένα κουμπί.

Οι περισσότερες σύνθετες εφαρμογές αποτελούνται από περισσότερα από ένα παράθυρα, και πιθανότατα κάθε στιγμή στην οθόνη να είναι απαραίτητη η εμφάνιση περισσότερων από ένα. Ο χρήστης έχει τη δυνατότητα να τα ανοίξει ή να τα κλείσει οποτεδήποτε. Αυτό είναι αρκετά θετικό καθώς κάθε στιγμή εμφανίζονται στην οθόνη μόνο τα δεδομένα τα οποία πρέπει να επεξεργαστεί. Έτσι ο χρήστης δε μπερδεύεται και η δουλειά του γίνεται ευκολότερη καθώς δεν υπάρχουν άχρηστα για την εργασία του δεδομένα. Αυτό δε μπορεί να συμβεί στην περίπτωση του λειτουργικού συστήματος. Είτε θα καθαριστεί οτιδήποτε έχει γραφτεί είτε θα παραμείνουν όλα πάνω στην οθόνη. Δηλαδή δεν υπάρχει η δυνατότητα να παραμείνουν μόνο κάποια από αυτά. Ένα άλλο πλεονέκτημα τους είναι το ότι μπορούν να πετύχουν καλύτερη διαχείριση οθόνης σε τέτοιου είδους εφαρμογές (με πολλά παράθυρα ή σε εφαρμογές που τρέχουν ταυτόχρονα σε πολλά μηχανήματα).

Από την άλλη πλευρά βέβαια υπάρχουν και κάποια μειονεκτήματα. Για παράδειγμα για τη σχεδίαση μιας ολοκληρωμένης γραφικής διεπαφής κάποιου προγράμματος, απαιτείται ένα διόλου ευκαταφρόνητο προγραμματιστικό έργο, η ανάπτυξη του οποίου απαιτεί αρκετό χρόνο και κόπο καθώς πρέπει να υλοποιηθούν ή να κληθούν πολλές ρουτίνες για τον έλεγχο του όλου συστήματος.

Λαμβάνοντας υπόψη όλα τα παραπάνω καταλήγουμε στο γεγονός ότι τα γραφικά περιβάλλοντα διασύνδεσης παρά τα όποια μειονεκτήματα, θεωρούνται πολύ σημαντικά. Κι αυτό γιατί απλά «κάνουν τη ζωή μας πιο εύκολη».

3.4 Κανόνες σχεδίασης γραφικών διεπαφών

Προτού προβεί κανείς στη σχεδίαση μιας γραφικής διεπαφής είναι αρχικά απαραίτητος ο καθορισμός των λειτουργιών οι οποίες θα πρέπει να εκτελούνται από την εφαρμογή. Στη συνέχεια για κάθε λειτουργία θα πρέπει να προσδιοριστούν όλοι οι δυνατοί τρόποι επικοινωνίας του προγράμματος με το χρήστη και στη συνέχεια για κάθε τρόπο οι απαιτούμενες ενέργειες από το χρήστη. Καθεμιά από τις δυνατές περιπτώσεις περιγράφεται από ένα σενάριο. Για κάθε σενάριο γίνεται μια εκτίμηση των απαιτούμενων ενεργειών από το χρήστη και κατά πόσο αυτές θα είναι εύκολα πραγματοποιήσιμες από το χρήστη και θα καθιστούν ευκολότερη την επικοινωνία του με τον υπολογιστή. Αφού γίνει η παραπάνω ανάλυση για κάθε σενάριο επιλέγεται πιο θα είναι αυτό που θα υλοποιηθεί.

Η σχεδίαση γραφικών διεπαφών, όπως η ανάπτυξη των περισσότερων εφαρμογών, πρέπει να υπακούει και αυτή σε κάποιους κανόνες. Σύμφωνα λοιπόν με αυτούς τους κανόνες μια γραφική διεπαφή πρέπει να χαρακτηρίζεται από:

1) ομοιομορφία

2) συνέπεια

3) απλότητα

4) αμεσότητα

5) χρήση μεταφορών

6) ευκαμψία

7) ανθεκτικότητα

8) ελαχιστοποίηση των απαιτούμενων ενεργειών από την πλευρά του χρήστη

9) ελαχιστοποίηση απομνημόνευσης από το χρήστη απαιτούμενων πληροφοριών για τη λειτουργία του προγράμματος.

Οι κανόνες αυτοί περιγράφονται αναλυτικά παρακάτω.

Ο πρώτος από αυτούς είναι η αρχή της ομοιομορφίας. Σύμφωνα με αυτή όλες οι φόρμες οι οποίες χρησιμοποιούνται σε ένα πρόγραμμα πρέπει να έχουν την ίδια εμφάνιση. Δηλαδή τα κουμπιά, τα πεδία κειμένου, οι εικόνες κλπ, πρέπει να εμφανίζονται στα ίδια σημεία και να εκτελούν την ίδια ακριβώς λειτουργία. Αν για παράδειγμα σε μια φόρμα κάτω δεξιά υπάρχουν κουμπιά για ελαχιστοποίηση ή για μεγιστοποίηση πρέπει και σε όλες τις άλλες φόρμες του προγράμματος, τα εν λόγω κουμπιά να είναι τοποθετημένα στην ίδια ακριβώς θέση.

Με βάση πάντα την αρχή της ομοιομορφίας, η γραφική διεπαφή που σχεδιάζουμε πρέπει να είναι σύμφωνη με τη μορφή και τη λειτουργικότητα της διεπαφής του λειτουργικού, για το οποίο προορίζεται η εφαρμογή μας. Δηλαδή αν στο λειτουργικό μας μια λειτουργία ενεργοποιείται με έναν συγκεκριμένο τρόπο, π.χ. με ένα συνδυασμό πλήκτρων, τότε αν και στο δικό μας γραφικό περιβάλλον υπάρχει μια τέτοια λειτουργία πρέπει να ενεργοποιείται με τον ίδιο τρόπο. Κλασικό παράδειγμα αποτελεί η λειτουργία save που ενεργοποιείται πατώντας ένα κουμπί, που φέρει την εικόνα μιας δισκέτας, ή με το συνδυασμό πλήκτρων Ctrl+S. Άρα λοιπόν και στο πρόγραμμά μας όταν θέλουμε να κάνουμε save, πρέπει να χρησιμοποιούμε τον ίδιο συνδυασμό πλήκτρων ή να έχουμε δημιουργήσει για το σκοπό αυτό ένα κουμπί με την ίδια εικόνα.

Συνεχίζουμε με τον κανόνα της συνέπειας. Σύμφωνα με τον κανόνα αυτό αν μια λειτουργία πρέπει να γίνει από διαφορετικά σημεία της εφαρμογής πρέπει κάθε

φορά να γίνεται με τον ίδιο τρόπο. Π.χ. αν μια λειτουργία επιλέγεται πατώντας ένα συγκεκριμένο συνδυασμό πλήκτρων, τότε πρέπει να επιλέγεται με αυτόν τον τρόπο σε οποιοδήποτε σημείο της εφαρμογής.

Ακολουθεί ο κανόνας της απλότητας. Ο κανόνας αυτός μας λέει ότι αν υπάρχουν δύο ή περισσότεροι τρόποι για να εκτελεστεί μια εφαρμογή ή μια λειτουργία πρέπει πάντα να επιλέγεται ο πιο απλός, έτσι ώστε το περιβάλλον να είναι όσο το δυνατό κατανοητότερο από το χρήστη.

Επόμενος κανόνας είναι ο κανόνας της αμεσότητας. Δηλαδή ο τρόπος εκτέλεσης ενεργειών που εκτελούνται συχνά από το χρήστη πρέπει να είναι απλός. Πιο αναλυτικά για να λάβει χώρα μια λειτουργία κρίνεται σκόπιμο αυτό να γίνει με έναν πολύ απλό τρόπο όπως το πάτημα ενός κουμπιού, παρά με περίπλοκες διαδικασίες του τύπου, όπως μια σειρά από μενού και υπομενού.

Επίσης συνετό είναι να γίνεται χρήση μεταφορών. Δηλαδή με άλλα λόγια να χρησιμοποιούνται ενέργειες και τρόποι αναπαράστασης για τις διάφορες λειτουργίες με τις οποίες ο χρήστης να είναι εξοικειωμένος. Αν για παράδειγμα μια εφαρμογή χρησιμοποιεί εκτυπωτή, τότε για τη λειτουργία της εκτύπωσης καλό θα ήταν να χρησιμοποιείται κάποιο κουμπί με την αναπαράσταση ενός εκτυπωτή.

Συνεχίζουμε με τον κανόνα της ελαχιστοποίησης ενεργειών από την πλευρά του χρήστη. Πιο συγκεκριμένα για την εκτέλεση μιας λειτουργίας, ο χρήστης πρέπει να κάνει το μικρότερο δυνατό αριθμό ενεργειών. Ακόμα σε περίπτωση λάθους του χρήστη κρίνεται σκόπιμο το λάθος να μπορεί να διορθωθεί επί τόπου και να μην είναι αναγκασμένος ο χρήστης να εκτελέσει τη διαδικασία και πάλι από την αρχή. Πιο αναλυτικά αν απαιτείται από το χρήστη να εισάγει κάποιες αριθμητικές τιμές και κάνει κάποιο λάθος π.χ. θέσει σε κάποιο πεδίο κάποια τιμή η οποία δεν είναι αριθμητική, τότε είναι αναγκαίο ο χρήστης να ειδοποιείται προκειμένου να διορθώσει το λάθος και να συνεχίσει από εκείνο το σημείο την ανάθεση τιμών και όχι να χαθούν οι τιμές που έχει δώσει πρωτύτερα και να πρέπει να ξεκινήσει την ανάθεση από την αρχή.

Κατόπιν αναλύουμε τον κανόνα ελαχιστοποίησης απομνημόνευσης. Σύμφωνα με τον κανόνα αυτό πρέπει οι διάφορες διαδικασίες να είναι αρκετά απλές έτσι ώστε να μην απαιτείται από το χρήστη να θυμάται πολλές λεπτομέρειες και τη σειρά πολλών βημάτων προκειμένου να τις εκτελέσει.

Άλλος σημαντικός κανόνας είναι η ευκαμψία του προγράμματος στις πληκτρολογήσεις και τις διάφορες ενέργειες του χρήστη. Δηλαδή αν σε κάποια

σημεία χρειάζεται ο χρήστης να πληκτρολογήσει κάποιο κείμενο, το κείμενο θα πρέπει να θεωρηθεί σωστό είτε πληκτρολογηθεί με κεφαλαία γράμματα, είτε με μικρά.

Τελευταίος κανόνας είναι ο κανόνας της ανθεκτικότητας. Ένα καλό πρόγραμμα θα πρέπει να είναι ανθεκτικό στα λάθη του χρήστη. Σε περίπτωση λάθους του χρήστη θα πρέπει το λάθος να διορθώνεται επί τόπου και ο χρήστης να συνεχίζει τη διαδικασία από το σημείο εκείνο στο οποίο εμφανίστηκε το πρόβλημα. Έτσι ο χρήστης κερδίζει χρόνο καθώς δε χρειάζεται να επαναλάβει τη διαδικασία από την αρχή.

Επιπλέον αν υπάρχει κάποιος τρόπος έτσι ώστε κάποια από τα λάθη του χρήστη να μπορούν να προβλεφθούν, καλό είναι να χρησιμοποιείται. Για παράδειγμα αν κατά την εισαγωγή δεδομένων από το χρήστη κάποιες εμφανιζόμενες επιλογές δεν αναφέρονται στη συγκεκριμένη λειτουργία τότε καλό είναι να εμφανίζονται απενεργοποιημένες.

Αν βέβαια η σχεδίαση πρέπει να γίνει με βάση ρητούς κανόνες τους οποίους θέτει ο πελάτης, τότε φυσικά η ανάπτυξη του γραφικού περιβάλλοντος πρέπει να γίνει σύμφωνα με αυτούς.

Επίσης η σχεδίαση πρέπει να γίνεται με τέτοιο τρόπο έτσι ώστε η τελική διεπαφή που θα προκύψει να είναι όσο το δυνατόν περισσότερο ποιοτική. Από την πλευρά του χρήστη ένα γραφικό περιβάλλον θεωρείται ποιοτικό αν είναι φιλικό προς αυτόν, δηλαδή αν τον διευκολύνει σε επαρκή βαθμό κατά την αλληλεπίδρασή του με το συγκεκριμένο πρόγραμμα (π.χ. αν η δομή του είναι τέτοια ώστε να μπορεί να εκτελεί τις διάφορες λειτουργίες γρήγορα και με μικρό αριθμό ενεργειών από το μέρος του ή αν το πρόγραμμα μπορεί να διορθώσει τυχόν λάθη του και να του επιτρέψει να συνεχίσει τη διαδικασία από το σημείο στο οποίο έγινε το λάθος). Επίσης σημαντικό είναι η απόδοση του προγράμματος, δηλαδή η ταχύτητα εκτέλεσης των διαφόρων διαδικασιών και η εξοικονόμηση της μνήμης που απαιτείται κατά τη διάρκεια λειτουργίας της εφαρμογής, να είναι ικανοποιητική.

Τέλος απαραίτητο είναι μια επαγγελματική γραφική διεπαφή να συνοδεύεται από τα απαραίτητα βοηθητικά εγχειρίδια για το χρήστη. Αυτά είναι το εγχειρίδιο χρήσης και το εγχειρίδιο αναφοράς. Το εγχειρίδιο χρήσης περιέχει πληροφορίες για τον τρόπο εκτέλεσης των διαφόρων λειτουργιών και γενικότερα για τον τρόπο λειτουργίας της όλης εφαρμογής. Επίσης πολύ συχνά αναφέρονται τα λάθη και οι δυσλειτουργίες που μπορεί να παρουσιαστούν κατά τη λειτουργία του προγράμματος

καθώς και πιθανούς τρόπους για την αντιμετώπισή τους. Επίσης πολλές φορές συμπεριλαμβάνονται και παραδείγματα χρήσης. Τώρα όσον αφορά το εγχειρίδιο αναφοράς, περιέχει αναλυτική περιγραφή όλων των διαθέσιμων λειτουργιών του προγράμματος συνήθως με αλφαβητική σειρά. Κάθε γραφική διεπαφή διαθέτει ένα ηλεκτρονικό εγχειρίδιο βοήθειας, το οποίο αποτελείται συνήθως από το εγχειρίδιο χρήσης.

4. Παρουσίαση Γραφικού Περιβάλλοντος Διασύνδεσης

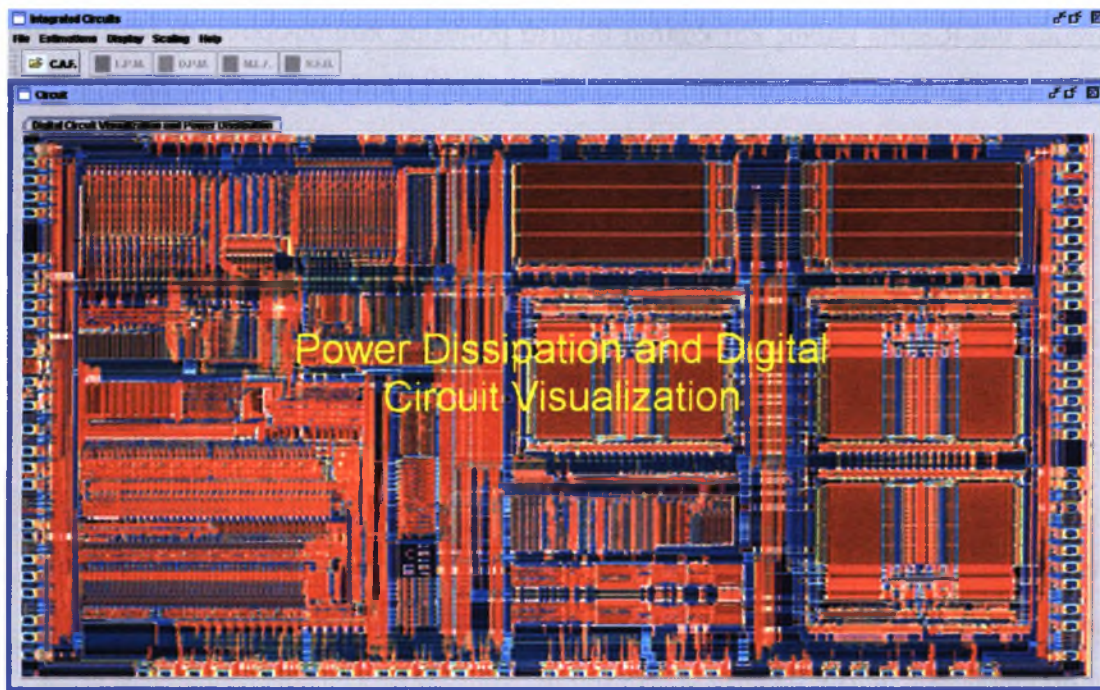
4.1 Περιγραφή κυρίου παράθυρου

Παρακάτω παρουσιάζεται μια σύντομη περιγραφή του γραφικού περιβάλλοντος διασύνδεσης που αναπτύξαμε.

Το κύριο παράθυρο του προγράμματος μας χωρίζεται σε δύο μέρη με τη χρήση μιας μετακινούμενης μπάρας. Σε καθένα από τα δυο μέρη που προκύπτουν υπάρχουν τοποθετημένα δυο εσωτερικά παράθυρα. Το παράθυρο που είναι τοποθετημένο στην πάνω πλευρά του κύριου παραθύρου έχει το όνομα Circuits και χρησιμοποιείται για την γραφική αναπαράσταση των κυκλωμάτων που περιγράφουν τα αρχεία spice τα οποία επεξεργαζόμαστε. Στο κάτω παράθυρο, με το όνομα Results παρουσιάζονται τα αποτελέσματα της εκάστοτε λειτουργίας.

Με την εκκίνηση του προγράμματος η μετακινούμενη μπάρα βρίσκεται στο κάτω μέρος της οθόνης και ολόκληρη η οθόνη καταλαμβάνεται από το παράθυρο Circuits. Το τελευταίο περιλαμβάνει μια καρτέλα με τον τίτλο «Digital Circuit Designing and Power Visualization». Επίσης τα περισσότερα μενού καθώς και τα σχετικά κουμπιά εμφανίζονται απενεργοποιημένα.

Το μονό κουμπί που εμφανίζεται ενεργοποιημένο είναι αυτό της επιλογής αρχείου. Από τα μενού απενεργοποιημένα είναι αυτά που παραπέμπουν σε λειτουργίες οι οποίες πρέπει να εκτελεστούν πάνω σε κάποιο κύκλωμα. Πιο συγκεκριμένα τα μενού που εμφανίζονται απενεργοποιημένα είναι τα μενού edit και scaling, η λειτουργία των οποίων θα αναλυθεί παρακάτω. Το γραφικό περιβάλλον, όπως αυτό εμφανίζεται με την έναρξη του προγράμματος, απεικονίζεται στην παρακάτω εικόνα.



To interface του προγράμματος

Με την επιλογή κάποιου αρχείου spice, ο τίτλος την καρτέλας αυτής μεταβάλλεται και παίρνει το όνομα του κυκλώματος, ενώ στο κύριο σώμα της καρτέλας εμφανίζεται η εικόνα του κυκλώματος. Παράλληλα ενεργοποιούνται τα μενού και τα κουμπιά που μέχρι εκείνη τη στιγμή, εμφανίζονταν απενεργοποιημένα.

Αφού λοιπόν ο χρήστης έχει ανοίξει κάποιο κύκλωμα προς επεξεργασία και επιλέξει κάποια λειτουργία τα διάφορα αποτελέσματα, οι παρατηρήσεις και τα τυχόν λάθη και προειδοποιητικά μηνύματα που μπορεί να προκύψουν κατά την εκτέλεση της λειτουργίας εμφανίζονται στο σχετικό παράθυρο. Παρακάτω παραθέτουμε για κάθε λειτουργία τα πιθανά μηνύματα που μπορεί να εμφανιστούν.

Εφόσον ο χρήστης τελειώσει με την επεξεργασία ενός αρχείου και θελήσει στη συνέχεια να εργαστεί με κάποιο άλλο, αρκεί απλά να το επιλέξει. Στο παράθυρο Circuit θα εμφανιστεί η εικόνα του νέου αρχείου και το πρόγραμμα θα συνεχίσει κανονικά τη λειτουργία του.

Συνοπτικά οι κύριες λειτουργίες που οι εφαρμογή μας πραγματοποιεί είναι ο υπολογισμός κατανάλωσης στατικής και δυναμικής ισχύος του κυκλώματος, καθώς και η μελέτη συμπεριφοράς του κυκλώματος σε Single Event Upsets, και ο υπολογισμός συνάρτησης ενός κόμβου σε ορισμένο βάθος επιπέδου. Επιπλέον το κύκλωμα παρέχει τη δυνατότητα αυξομειώσεως της εικόνας που αναπαριστά τα διάφορα ψηφιακά κυκλώματα, τη δυνατότητα επανεμφάνισης των παραθύρων

Circuits και Results καθώς και κάποια θέματα βοήθειας που αφορούν τη λειτουργία του προγράμματος. Ο τρόπος με τον οποίο εκτελείται καθεμιά από τις λειτουργίες αυτές περιγράφεται αναλυτικά στην παρακάτω παράγραφο.

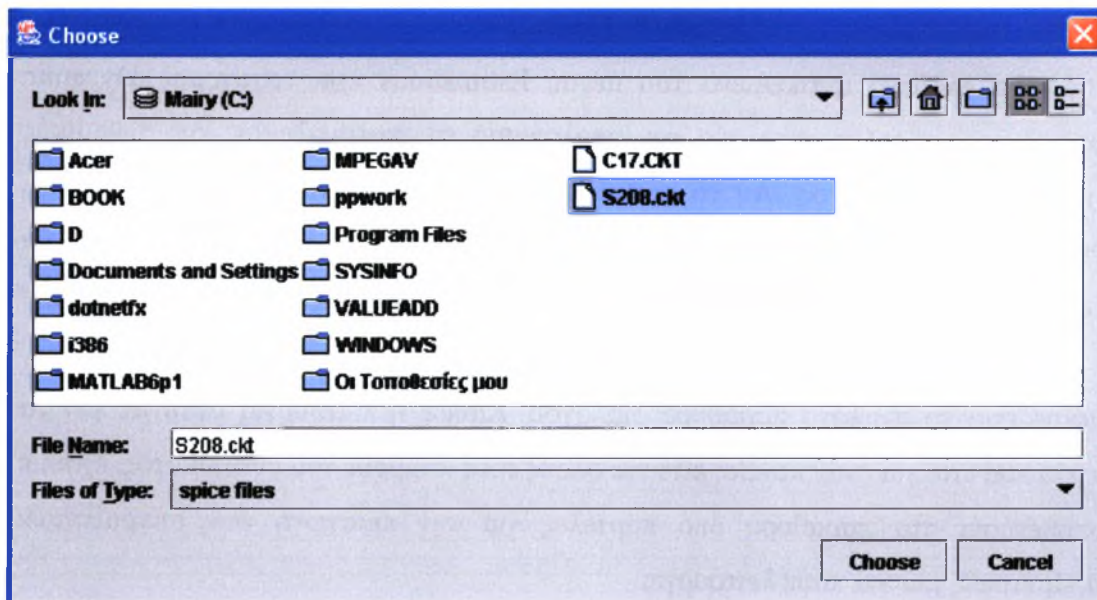
4.2 Τρόποι επιλογής λειτουργίας & αναλυτική περιγραφή των λειτουργιών

Οι χρήστες έχουν τη δυνατότητα να διαλέξουν τη λειτουργία την οποία επιθυμούν να εκτελεστεί με πολλούς τρόπους. Πιο συγκεκριμένα είναι δυνατή η επιλογή μιας λειτουργίας μέσω ενός κουμπιού, μέσω ενός συνδυασμού πλήκτρων, καθώς και μέσω της επιλογής ενός αντικειμένου μενού από μια σειρά μενού. Αρχικά θα ασχοληθούμε με την τελευταία περίπτωση.

4.2.1 Μενού

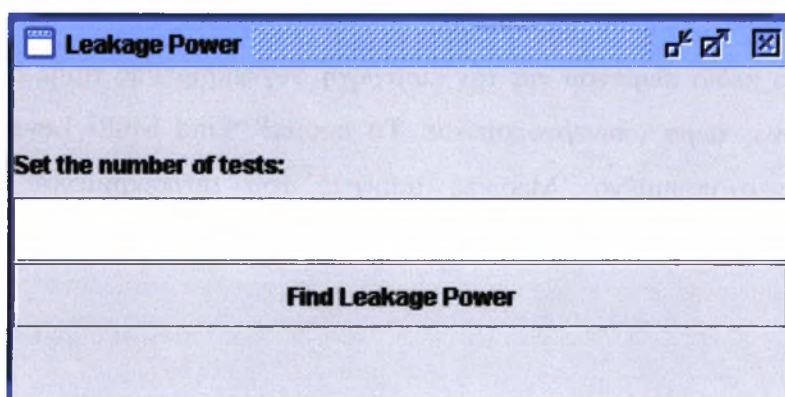
Η γραμμή των μενού του συγκεκριμένου γραφικού περιβάλλοντος περιλαμβάνει τα ακόλουθα μενού: File, Estimations, Display, Scaling και Help. Οι επιλογές που προσφέρονται από το καθένα αναλύονται παρακάτω.

Μενού File: Το μενού File έχει δυο επιμέρους αντικείμενα μενού τα « choose a file » και « exit ». Αν κάποιος επιλέξει το πρώτο αντικείμενο μενού εμφανίζεται ένα παράθυρο με τον τίτλο choose, μέσω του οποίου ο τελευταίος μπορεί να επιλέξει το αρχείο που επιθυμεί. Επιπλέον έχει τη δυνατότητα να επιλέξει τον τύπο των αρχείων που θα εμφανίζονται στο παράθυρο “choose”. Πιο συγκεκριμένα μπορεί να ζητήσει από το πρόγραμμα να εμφανίζονται μόνο αρχεία spice ή να εμφανίζονται όλα τα αρχεία, ανεξαρτήτως επέκτασης. Βέβαια ο χρήστης θα πρέπει να επιλέξει ένα αρχείο spice. Διαφορετικά θα εμφανίζεται μήνυμα το οποίο θα προειδοποιεί τον χρήστη ότι το συγκεκριμένο πρόγραμμα δεν είναι σε θέση να ανοίξει το συγκεκριμένο αρχείο. Το παράθυρο επιλογής αρχείου απεικονίζεται στην ακόλουθη εικόνα. Η επιλογή του κουμπιού exit έχει ως αποτέλεσμα την έξοδο από το πρόγραμμα.



Το παράθυρο choose για την επιλογή των αρχείων

Μενού *Estimations* : Το μενού estimations περιέχει τους υπολογισμούς που υπαγορεύονται από το αρχικό πρόγραμμα. Αν πατήσουμε το πρώτο κουμπί του εν λόγω μενού, εμφανίζεται ένα παράθυρο διαλόγου με τον τίτλο « leakage Power ». Εδώ ο χρήστης καλείται να εισάγει στο πεδίο κειμένου του συγκεκριμένου παραθύρου τον αριθμό επαναλήψεων της διαδικασίας. Μόλις πατήσει το κουμπί που βρίσκεται κάτω από το πεδίο κειμένου, υπολογίζονται τα διάφορα αποτελέσματα και τα σχετικά προειδοποιητικά μηνύματα εμφανίζονται στο παράθυρο Circuit. Το προαναφερόμενο παράθυρο εμφανίζεται στην παρακάτω εικόνα.



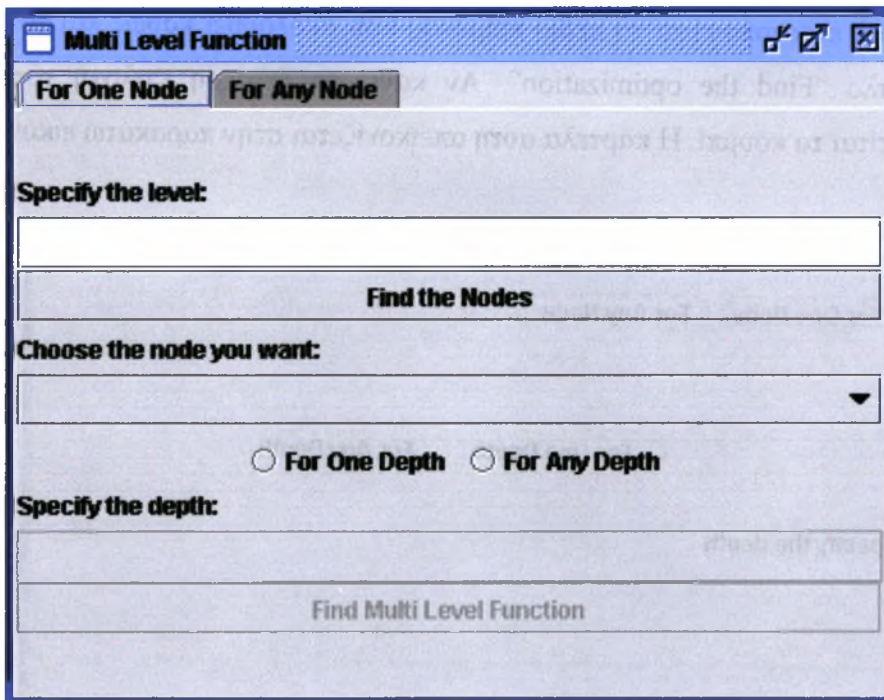
Το παράθυρο Leakage Power για τον υπολογισμό κατανάλωσης της στατικής ισχύος του κυκλώματος.

Το επόμενο αντικείμενο του μενού Estimations έχει το όνομα «Dynamic Power» και χρησιμοποιείται για τον υπολογισμό της κατανάλωσης της δυναμικής ισχύος του κυκλώματος. Αν το επιλέξουμε εμφανίζεται ένα παράθυρο διαλόγου ανάλογο με αυτό της προηγούμενης λειτουργίας. Το μόνο όρισμα που πρέπει να θέσουμε και σε αυτήν την περίπτωση είναι ο αριθμός επαναλήψεων της διαδικασίας.

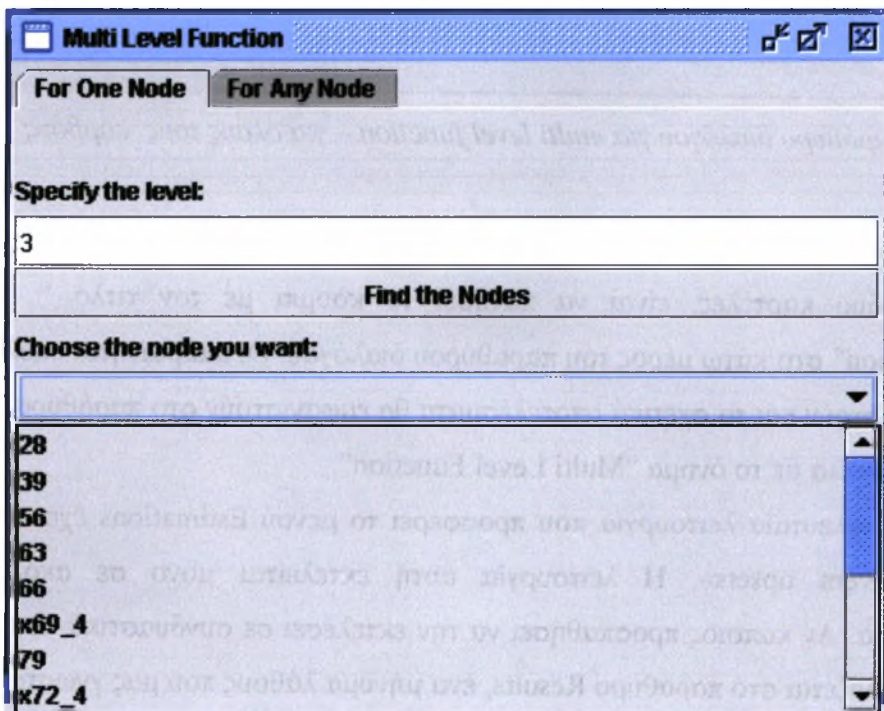
Όμοια αν επιλέξουμε το αντικείμενο μενού «Multi level function», εμφανίζεται το ανάλογο παράθυρο διαλόγου. Καθώς η λειτουργία αυτή μπορεί να εκτελεστεί είτε για έναν κόμβο, είτε για όλους τους κόμβους του κυκλώματος, έχουμε τοποθετήσει στο παράθυρο δυο καρτέλες για την εισαγωγή των απαραίτητων παραμέτρων, μια για κάθε λειτουργία.

Η καρτέλα με τον τίτλο “For one node”, αναφέρεται στην εκτέλεση των υπολογισμών για έναν κόμβο. Πιο συγκεκριμένα υπάρχει ένα πεδίο κειμένου στο οποίο κανείς πρέπει να δηλώσει το επίπεδο στο οποίο θέλει να γίνει η ανάλυση. Αφού εισαχθεί το επίπεδο και πατήσουμε το κουμπί που ακολουθεί, εμφανίζεται μια λίστα με όλους τους κόμβους του εν λόγω επιπέδου.

Αφού ο χρήστης επιλέξει κάποιον από αυτούς, πρέπει να προσδιορίσει για πόσα βάθη πρέπει να γίνει η ανάλυση. Έτσι υπάρχουν δυο κουμπιά επιλογών. Αν κανείς επιλέξει το πρώτο από αυτά τότε η ανάλυση γίνεται σε συγκεκριμένο βάθος. Στην περίπτωση αυτή ενεργοποιείται ένα πεδίο κειμένου στο οποίο πρέπει να εισαχθεί το εν λόγω βάθος και ένα κουμπί που βρίσκεται κάτω από το εν λόγω πεδίο κειμένου και έχει τίτλο “Find Multi Level Function”. Αν τώρα ο χρήστης επιλέξει το δεύτερο κουμπί τότε έχουμε ανάλυση για πολλά βάθη. Στην περίπτωση αυτή δε χρειάζεται κάποια επιπλέον παράμετρος. Αν προηγουμένως είχε επιλέγει το πρώτο κουμπί και το πεδίο κειμένου για την εισαγωγή συγκεκριμένης τιμής βάθους είναι ενεργοποιημένο, τώρα απενεργοποιείται. Το κουμπί “Find Multi Level Function” παραμένει ενεργοποιημένο. Μερικές απόψεις του συγκεκριμένου παραθύρου εμφανίζονται στις παρακάτω εικόνες.



(1)

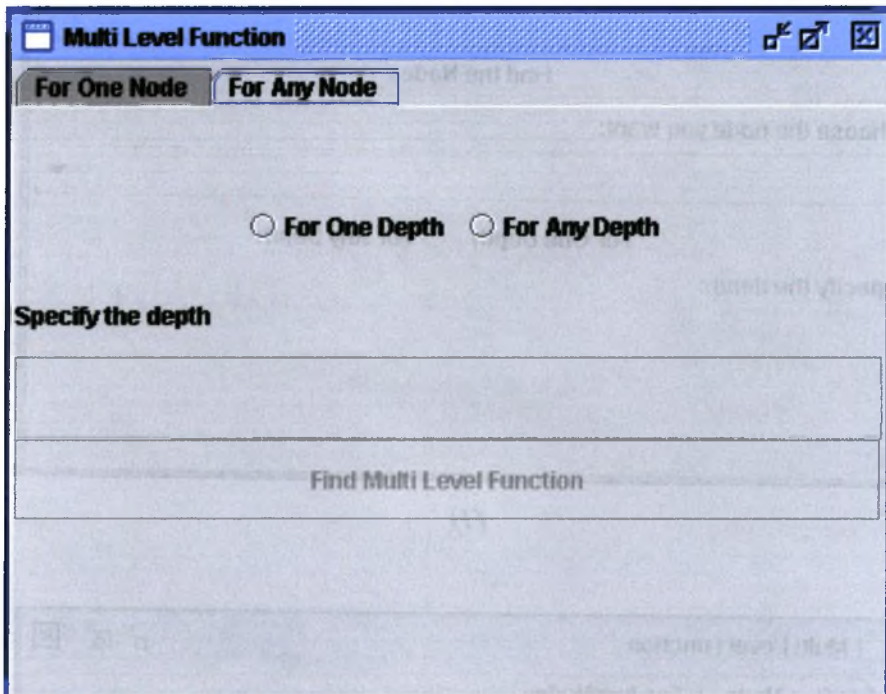


(2)

Παράθυρο διαλόγου για Multi Level Function

Στην καρτέλα « For Any Node », ο χρήστης και πάλι καλείται να επιλέξει αν θέλει να γίνει η ανάλυση για ένα βάθος ή για περισσότερα. Αν κάνει την πρώτη

επιλογή τότε ενεργοποιείται το πεδίο κειμένου που ακολουθεί καθώς και το κουμπί με τον τίτλο “Find the optimization”. Αν κάνει τη δεύτερη επιλογή τότε απλά ενεργοποιείται το κουμπί. Η καρτέλα αυτή απεικονίζεται στην παρακάτω εικόνα:



Παράθυρο διαλόγου για multi level function – για όλους τους κόμβους

Μετά από όλα αυτά το μόνο που πρέπει να κάνει ο χρήστης, σε οποιαδήποτε από τις δυο καρτέλες, είναι να πατήσει το κουμπί με τον τίτλο “ find the optimization” στο κάτω μέρος του παραθύρου διαλόγου. Οι απαραίτητοι υπολογισμοί θα εκτελεστούν και τα σχετικά αποτελέσματα θα εμφανιστούν στο παράθυρο Results, σε μια καρτέλα με το όνομα “Multi Level Function”.

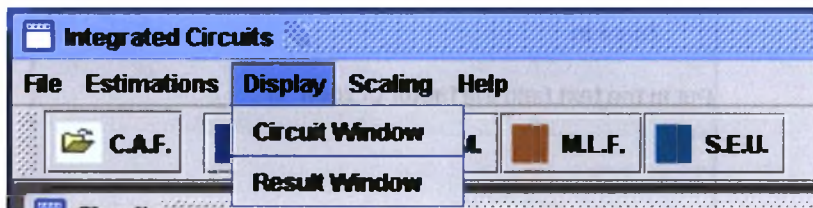
Η τελευταία λειτουργία που προσφέρει το μενού Estimations έχει τον τίτλο «single event upsets». Η λειτουργία αυτή εκτελείται μόνο σε ακολουθιακά κυκλώματα. Αν κάποιος προσπαθήσει να την εκτελέσει σε συνδυαστικά κυκλώματα, τότε εμφανίζεται στο παράθυρο Results, ένα μήνυμα λάθους που μας γνωστοποιεί ότι το κύκλωμά μας δεν έχει flip-flops.

Επιλέγοντας αυτή τη λειτουργία εμφανίζεται και πάλι ένα παράθυρο διαλόγου παρόμοιο με αυτό των δύο πρώτων λειτουργιών και ζητά από το χρήστη να εισάγει και πάλι τον αριθμό επαναλήψεων της διαδικασίας. Μόλις ο χρήστης πατήσει το κουμπί εκτελείται η λειτουργία και τα διάφορα μηνύματα εμφανίζονται για ακόμα μια φορά στο παράθυρο Results.

Κάτι που θεωρείται σημαντικό να αναφερθεί είναι το ότι αν ο χρήστης σε οποιοδήποτε από τα προαναφερόμενα διαλόγου δε θέσει τιμές σε όλα τα πεδία κειμένου ή θέσει τιμές λανθασμένες, τότε εμφανίζονται κατάλληλα προειδοποιητικά μηνύματα, τα οποία επισημαίνουν το λάθος. Στο πρώτο παράρτημα της παρούσας εργασίας παρουσιάζονται αναλυτικά όλα τα προειδοποιητικά μηνύματα που μπορούν να παρουσιαστούν κατά τη διάρκεια εκτέλεσης του προγράμματός μας.

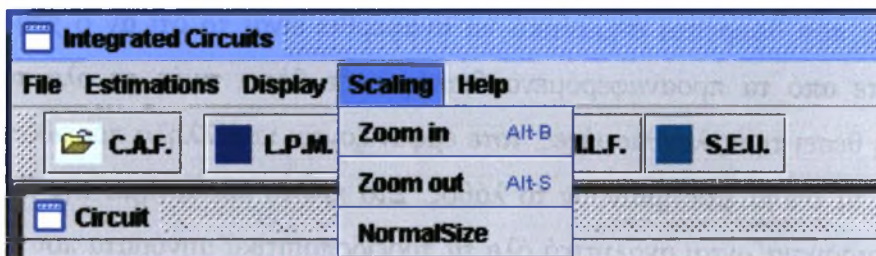
Οι επιλογές του αυτού του μενού στην αρχή του προγράμματός μας, όπως αναφέρθηκε και πρωτύτερα εμφανίζονται απενεργοποιημένες. Ενεργοποιούνται αφότου το πρόγραμμα ανοίξει επιτυχώς ένα αρχείο spice.

Μενού display: Το μενού display, χρησιμοποιείται για την εμφάνιση των εσωτερικών παραθύρων του προγράμματός μας. Έχει δυο αντικείμενα μενού τα “Circuit Window” και “Results Window”. Αν κλείσουμε ένα από τα δυο παράθυρα π.χ. το παράθυρο Circuit, τότε έχουμε τη δυνατότητα να το επανεμφανίσουμε στην οθόνη πατώντας το αντικείμενο μενού που αναφέρεται σε αυτό (Circuit Window). Κατά την εμφάνισή του, το παράθυρο απεικονίζει αυτά ακριβώς που απεικόνιζε τη στιγμή που το κλείσαμε. Την ίδια λειτουργία επιτελεί και το “Results Window” για το παράθυρο των αποτελεσμάτων.



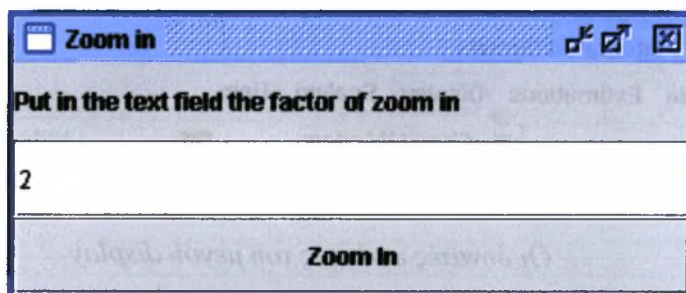
Οι δυνατές επιλογές του μενού display

Μενού scaling: Επόμενο μενού είναι το μενού scaling, με τη χρήση του οποίου μπορούμε να αυξομειώσουμε το μέγεθος του απεικονιζόμενου κυκλώματος. Περιέχει τρία αντικείμενα μενού. Το πρώτο από αυτά έχει το όνομα “zoom in” και χρησιμοποιείται για τη μεγέθυνση της εικόνας, το δεύτερο έχει το όνομα “zoom out” και χρησιμοποιείται για τη σμίκρυνση της εικόνας, ενώ το τρίτο έχει το όνομα “Normal Size” και επαναφέρει την εικόνα στο αρχικό της μέγεθος. Το μενού scaling παρουσιάζεται στην παρακάτω εικόνα.



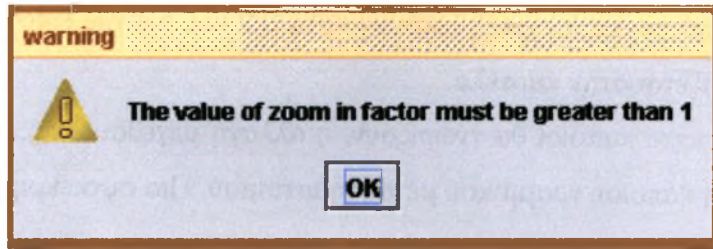
Οι δυνατές επιλογές του μενού scaling

Πατώντας κάποιο από τα εν λόγω αντικείμενα μενού π.χ. το “zoom in”, τότε εμφανίζεται ένα παράθυρο, με ανάλογο τίτλο, στο οποίο πρέπει να θέσουμε τον παράγοντα μεγέθυνσης της εικόνας. Αν στη συνέχεια πατήσουμε το κουμπί, τότε το νέο κύκλωμα εμφανίζεται στην καρτέλα που πριν εμφανιζόταν το κύκλωμα χωρίς κλιμάκωση του μεγέθους του. Αυτό που έχει αλλάξει είναι ο τίτλος της καρτέλας. Ο νέος τίτλος είναι ο εξής: Zoom in (by a factor x), όπου x ο παράγοντας μεγέθυνσης της εικόνας. Το παράθυρο της μεγέθυνσης εμφανίζεται παρακάτω. Στο κύριο σώμα της καρτέλας (όπου απεικονίζεται το κύκλωμα) υπάρχει οριζόντια και κάθετη μπάρα έτσι ώστε ο χρήστης να έχει τη δυνατότητα να δει οποιοδήποτε τμήμα του κυκλώματος. Το παράθυρο που χρησιμοποιείται για τη μεγέθυνση απεικονίζεται στην παρακάτω εικόνα:



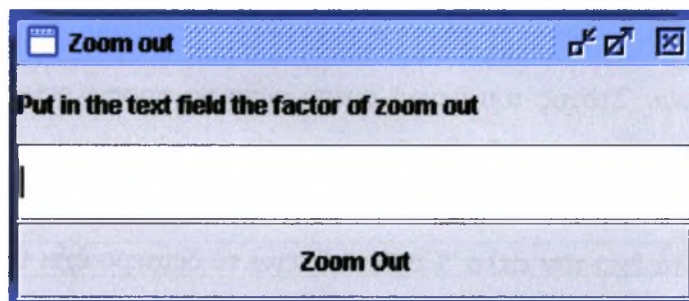
Παράθυρο διαλόγου που χρησιμοποιείται για τη μεγέθυνση της εικόνας

Αν στο πεδίο κειμένου του εν λόγω παραθύρου εισαχθούν τιμές ακατάλληλες για την πραγματοποίηση της συγκεκριμένης λειτουργίας, το πρόγραμμα μας προειδοποιεί ότι κάναμε κάποιο λάθος με κατάλληλα μηνύματα. Αν για παράδειγμα η τιμή που εισαχθεί είναι μη αριθμητική, τότε εμφανίζεται ένα παράθυρο με το μήνυμα: Check the value of zoom in factor. Ένα τέτοιο παράθυρο εμφανίζεται στην ακόλουθη εικόνα.



Προειδοποιητικό παράθυρο διαλόγου που εμφανίζεται σε περίπτωση που ο χρήστης δώσει τιμή μικρότερη από 1.

Πατώντας το αντικείμενο μενού “zoom out”, εκτελούνται λειτουργίες ανάλογες με τις προηγούμενες μόνο που τώρα αφορούν τη σμίκρυνση του κυκλώματος. Πιο συγκεκριμένα εμφανίζεται ένα ανάλογο παραθυράκι στο οποίο πρέπει να θέσουμε τον παράγοντα σμίκρυνσης της εικόνας. Οι τιμές σε αυτή την περίπτωση πρέπει να είναι αριθμητικές, θετικές, και μικρότερες της μονάδας. Η εικόνα του ελαχιστοποιημένου κυκλώματος εμφανίζεται και πάλι στην ίδια καρτέλα, μόνο που αυτή τη φορά ο τίτλος της είναι: “Zoom out (by a factor x)”. Στην ίδια καρτέλα, όπως και πριν εμφανίζονται μπάρες κύλισης, όπου αυτές είναι απαραίτητες. Το παράθυρο “zoom out” εμφανίζεται στην παρακάτω εικόνα.



Παράθυρο διαλόγου που χρησιμοποιείται για τη σμίκρυνση της εικόνας

Τέλος πατώντας το αντικείμενο μενού “Normal Size”, στο κύριο σώμα της καρτέλας εμφανίζεται το κύκλωμα στο αρχικό του μέγεθος, ενώ ο τίτλος της είναι ο τίτλος του κυκλώματος. Δηλαδή πατώντας αυτό το αντικείμενο μενού επαναφέρουμε την καρτέλα στη μορφή που αυτή είχε μόλις επιλέξαμε το κύκλωμα.

Αυτό που πρέπει να τονίσουμε είναι ότι τόσο η σμίκρυνση, όσο και η μεγέθυνση της εικόνας γίνονται με αναφορά το αρχικό μέγεθος της εικόνας. Αν π.χ. στο παράθυρο διαλόγου έχουμε εισάγει μια τιμή 0.4 και στη συνέχεια εισάγουμε 0.2,

θα γίνει σμίκρυνση της αρχικής εικόνας κατά 0.2 και όχι της εικόνας που εκείνη τη στιγμή εμφανίζεται στην καρτέλα.

Όπως ίσως κάποιιοι θα γνωρίζουν, η αλλαγή μεγέθους της εικόνας γίνεται με την εφαρμογή κάποιου γραμμικού μετασχηματισμού. Πιο συγκεκριμένα η κλιμάκωση των διαφόρων σημείων της εικόνας επιτυγχάνεται με πολλαπλασιασμό των συντεταγμένων των σημείων με κάποιο παράγοντα. Δηλαδή αν $s1$ ο παράγοντας για τον άξονα των x , και $s2$ ο παράγοντας για τον άξονα των y , τότε οι νέες συντεταγμένες x' , y' των σημείων της εικόνας προκύπτουν ως εξής:

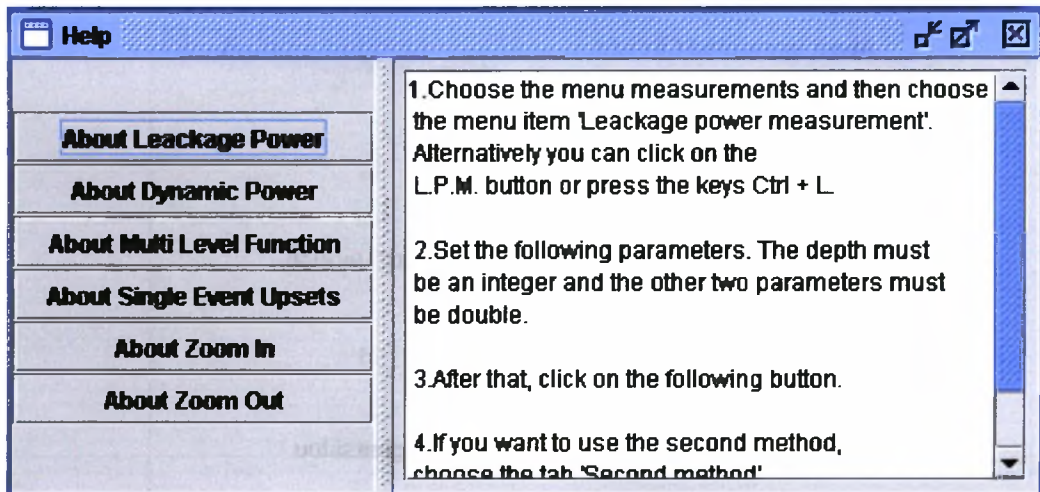
$$(1) x' = s1 * x \qquad (2) y' = s2 * y.$$

Όταν έχουμε σμίκρυνση σε κάποιον άξονα, ο αντίστοιχος παράγοντας πρέπει να είναι θετικός και μικρότερος του 1, ενώ σε αντίθετη περίπτωση πρέπει να είναι μεγαλύτερος.

Αν τα $s1$ και $s2$ δεν είναι ίσα, το σώμα επιμηκώνεται στη μια κατεύθυνση και συρρικνώνεται στην άλλη, διαφορετικά οι αναλογίες του δε μεταβάλλονται. Εμείς χρησιμοποιούμε τον ίδιο παράγοντα x και για τους δυο άξονες, με αποτέλεσμα να μη μεταβάλλονται οι αναλογίες της εικόνας.

Μενού Help: Στόχος του μενού αυτού είναι να προσφέρει στο χρήστη τις απαραίτητες πληροφορίες και οδηγίες έτσι ώστε να μπορέσει να χρησιμοποιήσει την εφαρμογή με όσο το δυνατόν μικρότερο αριθμό λαθών. Έχει δυο αντικείμενα μενού. Το πρώτο από αυτά έχει τον τίτλο “Contents”, ενώ το δεύτερο έχει τον τίτλο “About Integrated Circuits”.

Πατώντας το πρώτο από αυτά εμφανίζεται ένα παραθυράκι το οποίο στην αριστερή του πλευρά έχει κάποια κουμπιά (τα οποία στην ουσία αποτελούν τα περιεχόμενα του ηλεκτρονικού αυτού εγχειριδίου), ενώ στη δεξιά του πλευρά έχει ένα πλαίσιο κειμένου. Αν πατήσουμε κάποιο από τα προαναφερόμενα κουμπιά τότε στο εν λόγω πλαίσιο κειμένου εμφανίζονται τα βήματα που πρέπει να ακολουθήσει ο χρήστης προκειμένου να εκτελέσει με επιτυχία τη λειτουργία στην οποία αναφέρεται το συγκεκριμένο κουμπί. Το παράθυρο αυτό παρουσιάζεται στην παρακάτω εικόνα:

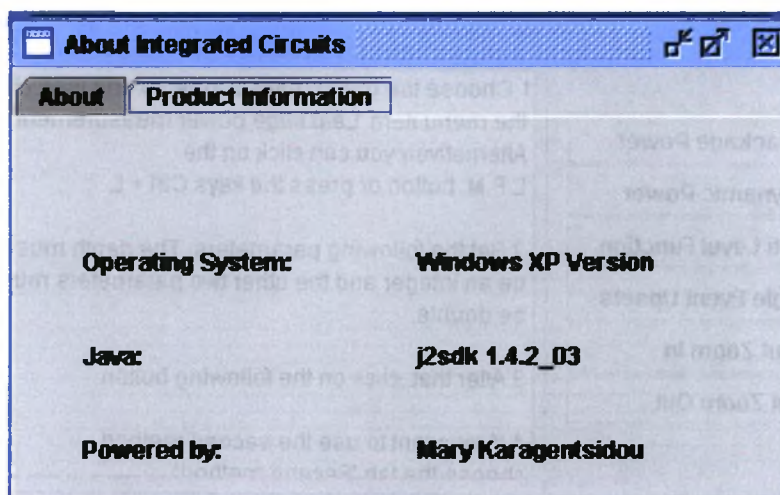


Στο παράρτημα Γ παρουσιάζουμε σε έναν κατάλογο τα κουμπιά του εν λόγω παραθύρου, τις λειτουργίες τις οποίες αντιπροσωπεύουν (οι οποίες φαίνονται από την ετικέτα του κουμπιού), καθώς και το κείμενο που εμφανίζεται στο διπλανό τμήμα του παραθύρου, μόλις πατήσουμε κάποιο από αυτά.

Πατώντας το δεύτερο αντικείμενο μενού εμφανίζεται ένα παραθυράκι με τον τίτλο “About Integrated circuits” το οποίο μας δίνει κάποιες πληροφορίες για το πρόγραμμα, όπως για παράδειγμα την έκδοση της Java, η οποία χρησιμοποιήθηκε για την ανάπτυξη του προγράμματος. Το παράθυρο αυτό έχει δύο καρτέλες με τίτλους “About”, και “Product Information”. Οι δυο αυτές καρτέλες παρουσιάζονται στις ακόλουθες εικόνες.



Η πρώτη καρτέλα του παραθύρου “About Integrated Circuits”



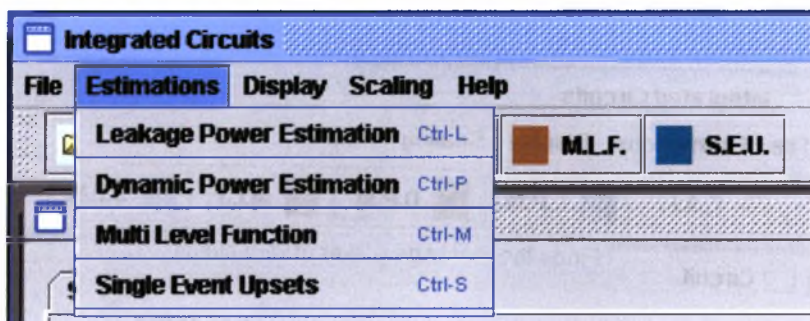
Η δεύτερη καρτέλα του παραθύρου “About Integrated Circuits”

4.2.2 Συνδυασμοί πλήκτρων

Η επιλογή κάποιων από τις προαναφερόμενες λειτουργίες μπορεί να γίνει και με άλλους τρόπους, όπως με το να πατήσουμε ταυτόχρονα περισσότερα από ένα πλήκτρα. Πιέζοντας λοιπόν ένα συνδυασμό πλήκτρων, εμφανίζεται στην οθόνη το σχετικό παράθυρο διαλόγου, ή εκτελείται η εν λόγω λειτουργία. Πιο συγκεκριμένα έχουμε προγραμματίσει τα εξής:

- αν πιάσουμε τα πλήκτρα Ctrl και O εμφανίζεται το παράθυρο επιλογής αρχείου.
- αν πιάσουμε τα πλήκτρα Ctrl και L εμφανίζεται το παράθυρο για leakage power estimation.
- αν πιάσουμε τα πλήκτρα Ctrl και D εμφανίζεται το παράθυρο για dynamic power estimation.
- αν πιάσουμε τα πλήκτρα Ctrl και M εμφανίζεται το παράθυρο για multi level function.
- αν πιάσουμε ταυτόχρονα τα πλήκτρα Ctrl και S εκτελείται η λειτουργία single event upsets και εμφανίζονται τα σχετικά αποτελέσματα.
- αν πιάσουμε ταυτόχρονα τα πλήκτρα Alt και B εμφανίζεται το παράθυρο για τη λειτουργία “Zoom in”.
- αν πιάσουμε ταυτόχρονα τα πλήκτρα Alt και S εμφανίζεται το παράθυρο για τη λειτουργία “Zoom out”.

Κάποιοι από τους προαναφερόμενους συνδυασμούς πλήκτρων και τις ανάλογες λειτουργίες εμφανίζονται στην παρακάτω εικόνα:



Επιλογή λειτουργιών μέσω συνδυασμού πλήκτρων

4.2.3 Γραμμή εργαλείων (Tool Bar)

Όμως παρατηρούμε ότι κάτω από τη γραμμή στην οποία εμφανίζονται τα διάφορα μενού, υπάρχει μια μπάρα κουμπιών. Τα κουμπιά αυτά αποτελούν έναν ακόμα τρόπο επιλογής ορισμένων λειτουργιών του προγράμματος όμως. Οι λειτουργίες που μπορούν να επιλεγούν με αυτό τον τρόπο για εκτέλεση είναι η επιλογή αρχείων και οι τέσσερις που προσφέρονται από το αρχικό πρόγραμμα (δηλαδή με άλλα λόγια οι λειτουργίες που μπορούν να επιλεγούν και με συνδυασμό κάποιων πλήκτρων).

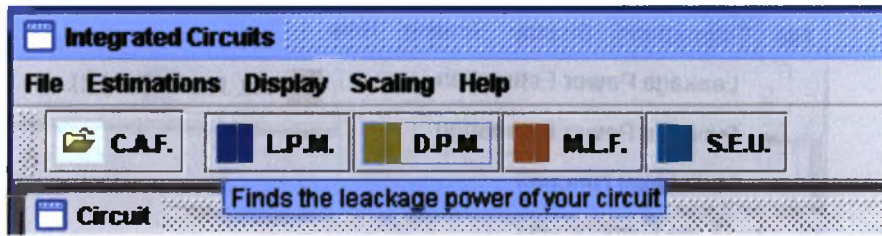
Στην επιφάνεια του κάθε κουμπιού υπάρχει μια εικόνα καθώς και τα αρχικά όμως λειτουργίας την οποία αντιπροσωπεύει. Δηλαδή:

- 1) C.A.F. - Choose a file
- 2) L.P.M. - Leakage Power Estimation
- 3) D.P.M. - Dynamic Power Estimation
- 4) M.L.F. - Multi Level Function
- 5) S.E.U. - Single Event Upsets

Επιπλέον αν τοποθετήσουμε το ποντίκι πάνω σε κάποιο από αυτά χωρίς όμως να κάνουμε κλικ, εμφανίζεται κάτω από τη μπάρα ένα επεξηγηματικό κείμενο για τη λειτουργία που εκτελεί το κάθε κουμπί. Πιο συγκεκριμένα:

- 1) C.A.F. – Choose the spice file which you want
- 2) L.P.M. – Finds the leakage power of your circuit
- 3) D.P.M. – Finds the dynamic power of your circuit

- 4) M.L.F. – Finds the multi level function of your circuit
 5) S.E.U. – Finds the circuit behavior when there are single event upsets
 Τα κουμπιά αυτά απεικονίζονται στην παρακάτω εικόνα:

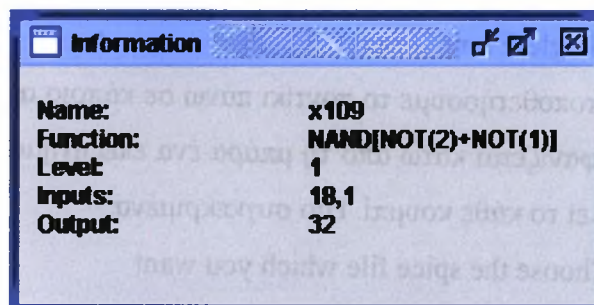


Μπάρα κουμπιών επιλογής λειτουργίας

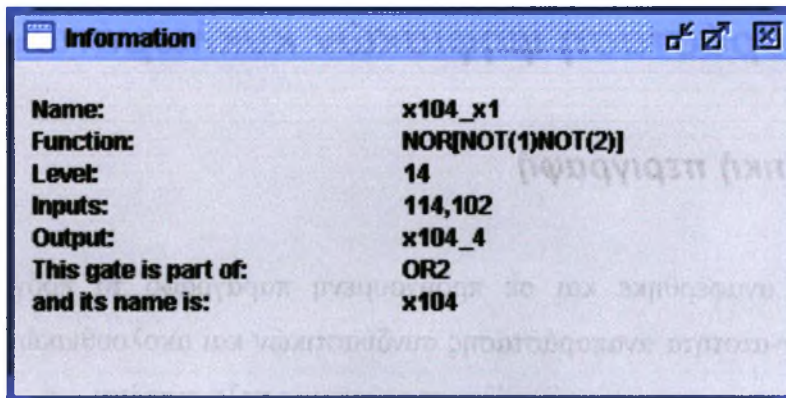
4.2.4 Επιπλέον πληροφορίες για κάθε πύλη

Καθώς κάποιο ψηφιακό κύκλωμα απεικονίζεται στο παράθυρο Circuit, του κυκλώματος, έχουμε τη δυνατότητα να μάθουμε περισσότερες πληροφορίες για τις διάφορες πύλες από τις οποίες αποτελείται.

Αν κάνουμε κλικ με το ποντίκι πάνω στην πύλη η οποία μας ενδιαφέρει, εμφανίζεται ένα παραθυράκι με τον τίτλο “Information” που μας δίνει τις επιπλέον πληροφορίες για την πύλη αυτήν. Οι διαθέσιμες πληροφορίες για κάθε πύλη, είναι το όνομά της, η συνάρτησή της, οι είσοδοί της, και η έξοδός της. Επιπλέον σε περίπτωση που η πύλη προέκυψε από διάσπαση κάποιας άλλης πύλης (για παράδειγμα μια πύλη NOT μπορεί να προέρχεται από τη διάσπαση μιας πύλης NAND σε μια πύλη AND και μια NOT), τότε εμφανίζεται ο τύπος της αρχικής πύλης καθώς και το όνομά της. Τα εν λόγω παραθυράκια εμφανίζονται στην παρακάτω εικόνα:

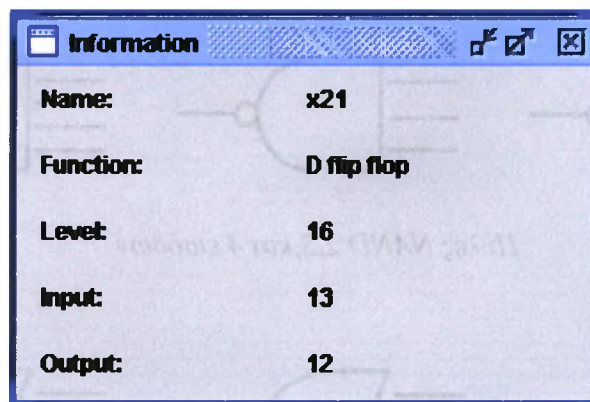


Παράθυρο πληροφοριών σε περίπτωση που η πύλη δεν προκύπτει από τη διάσπαση άλλης πύλης



Παράθυρο πληροφοριών σε περίπτωση που η συγκεκριμένη πύλη προκύπτει από τη διάσπαση άλλης πύλης

Αν έχουμε D flip flop τότε οι διαθέσιμες πληροφορίες είναι το όνομά του, η είσοδός του, η έξοδός του καθώς και το επίπεδο στο οποίο βρίσκεται, το οποίο είναι το τελευταίο επίπεδο του κυκλώματος όπως θα δούμε στη συνέχεια. Το συγκεκριμένο παραθυράκι εμφανίζεται στην παρακάτω εικόνα:



Παράθυρο πληροφοριών για D flip flops

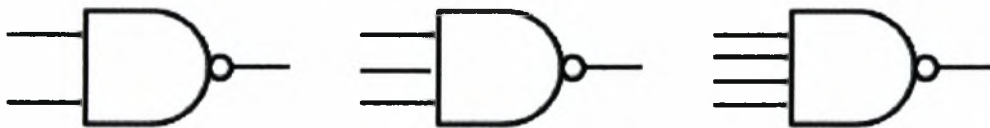
Αυτή η δυνατότητα υπάρχει μόνο στην περίπτωση που το κύκλωμα απεικονίζεται στο φυσικό του μέγεθος. Δηλαδή αν ο χρήστης έχει σμικρύνει ή μεγεθύνει το κύκλωμα, και θέλει να εμφανίσει τις πληροφορίες κάθε πύλης, τότε πρέπει πρώτα να επαναφέρει την εικόνα στην αρχική της κατάσταση επιλέγοντας το αντικείμενο μενού “Normal Size”, από το μενού scaling, και στη συνέχεια να κάνει κλικ με το ποντίκι πάνω στην πύλη που τον ενδιαφέρει.

5. Αναπαράσταση ψηφιακών κυκλωμάτων

5.1 Συνοπτική περιγραφή

Όπως αναφέρθηκε και σε προηγούμενη παράγραφο το πρόγραμμά μας παρέχει τη δυνατότητα αναπαράστασης συνδυαστικών και ακολουθιακών ψηφιακών κυκλωμάτων, που χρησιμοποιούν πύλες τεσσάρων το πολύ εισόδων.

Για τη σχεδίαση του κυκλώματος χρησιμοποιούμε εικόνες της πύλης NOT, των πυλών NAND και NOR, δύο, τριών και τεσσάρων εισόδων, καθώς και D flip flops. Σε περίπτωση που πρέπει να σχεδιάσουμε πύλες διαφορετικού τύπου π.χ. AND ή OR, τις απεικονίζουμε ως συνδυασμό πυλών. Πιο συγκεκριμένα την πύλη AND θα την απεικονίσουμε ως μια πύλη NAND, ακολουθούμενη από μια πύλη NOT, ενώ την πύλη OR ως μια πύλη NOR ακολουθούμενη και πάλι από μια πύλη NOT. Οι εικόνες που χρησιμοποιούνται για τη σχεδίαση του κυκλώματος παρουσιάζονται παρακάτω:



Πύλες NAND 2,3, και 4 εισόδων



Πύλες NOR 2,3 και 4 εισόδων



Πύλη NOT



D flip flop

Οι πύλες του ίδιου επιπέδου σχεδιάζονται η μια κάτω από την άλλη, όπως συνήθως συμβαίνει σε αυτού του είδους τις σχεδιάσεις. Σε περιπτώσεις ακολουθιακών κυκλωμάτων τα D flip flops τοποθετούνται στο τέλος του κυκλώματος. Η απόσταση των πυλών του ίδιου επιπέδου, καθώς και μεταξύ των πυλών διαφορετικών επιπέδων είναι επαρκώς μεγάλη, έτσι ώστε η αναπαράσταση της καλωδίωσης του κυκλώματος, να είναι ευδιάκριτη σε ικανοποιητικό βαθμό.

Προκειμένου να μην προκαλείται σύγχυση κατά την απεικόνιση των καλωδίων, καθώς πολλά καλώδια πιθανόν να ταυτίζονται, αυτά σχεδιάζονται έτσι ώστε να υπάρχει απόσταση μερικών χιλιοστών μεταξύ τους, έτσι ώστε ο χρήστης να μην μπερδεύεται και να μπορεί να κατανοήσει ποιες είναι οι συνδέσεις του κυκλώματος.

Για να είναι η σχεδίαση ακόμη πιο φιλική προς το χρήστη, έχουμε φροντίσει έτσι ώστε δίπλα σε κάθε είσοδο και έξοδο πύλης, να εμφανίζεται το όνομά της με κόκκινη γραμματοσειρά. Επίσης το όνομα κάθε εξόδου (που είναι ίδιο με το όνομα των εισόδων με τις οποίες συνδέεται), εμφανίζεται με κόκκινη γραμματοσειρά και πάνω σε τμήματα της καλωδίωσης που χρησιμοποιείται για τη συγκεκριμένη σύνδεση. Τέλος με κόκκινη γραμματοσειρά εμφανίζονται και τα ονόματα των εισόδων του κυκλώματος.

Παρακάτω θα παρουσιάσουμε αναλυτικά τις δομές που χρησιμοποιήθηκαν κατά τη διάρκεια της σχεδίασης και θα αναλύσουμε ορισμένα τμήματα της διαδικασίας που ακολουθήθηκε κατά τη σχεδίαση των κυκλωμάτων.

5.2 Διαδικασία απεικόνισης ψηφιακών κυκλωμάτων

5.2.1 Δομές οι οποίες χρησιμοποιήθηκαν κατά τη σχεδίαση

Η δομή gate: Είναι η δομή που αναπαριστά τις διάφορες πύλες του κυκλώματος. Όπως αναφέραμε και νωρίτερα οι πύλες που χρησιμοποιούμε είναι τύπου NOT, ή τύπου NAND και NOR, δύο, τριών ή τεσσάρων εισόδων. Τα χαρακτηριστικά της πύλης είναι:

- 1) **το όνομά της**
- 2) **η συνάρτησή της**
- 3) **οι εισοδοί της** (θεωρούμε ότι κάθε πύλη έχει τέσσερις εισόδους. Αν η πύλη έχει δύο ή τρεις εισόδους τότε οι άλλες εισοδοί δηλώνονται ως null)
- 4) **η έξοδος της**
- 5) **το επίπεδο στο οποίο βρίσκεται**

Η δομή signal: Είναι η δομή που χρησιμοποιείται για την περιγραφή των εισόδων και εξόδων των διαφόρων πυλών. Η δομή αυτή αποτελείται από:

- 1) **το όνομα του σήματος**
- 2) **τις συντεταγμένες του x και y**
- 3) **κάποιες ακέραιες παραμέτρους με τα ονόματα *more*, *nextStartX*, και *nextStartY*** (οι οποίες χρησιμοποιούνται για τη σχεδίαση καλωδίων του κυκλώματος).
- 4) **ένα Vector το οποίο σε περίπτωση που ένα signal χρησιμοποιείται ως έξοδος περιέχει τις πύλες των οποίων κάποια είσοδος πρέπει να συνδεθεί με την εν λόγω έξοδο.** Σε περίπτωση που ένα σήμα χρησιμοποιείται ως είσοδος ή χρησιμοποιείται ως έξοδος αλλά δε συνδέεται με καμία είσοδο (δηλαδή με άλλα λόγια αποτελεί έξοδο του κυκλώματος), τότε το εν λόγω Vector θεωρείται κενό.

5.2.2 Αναλυτική περιγραφή της διαδικασίας που ακολουθείται κατά τη σχεδίαση του κυκλώματος

Δεδομένα από το ήδη υπάρχον πρόγραμμα – τοποθέτηση των εικόνων των πυλών: Για την αναπαράσταση του κυκλώματος το πρόγραμμά μας δέχεται ένα αλφαριθμητικό από το ήδη υπάρχον πρόγραμμά, το οποίο περιγράφει το κύκλωμα που πρόκειται να αναπαρασταθεί. Δίνει πληροφορίες για τις πύλες σε περίπτωση που το κύκλωμα είναι συνδυαστικό, ενώ αν έχουμε να κάνουμε με ακολουθιακά κυκλώματα δίνει επιπλέον πληροφορίες και για τα υπάρχοντα D flip flops.

Η μορφή του προαναφερόμενου αλφαριθμητικού για τα συνδυαστικά κυκλώματα είναι η ακόλουθη: “gate1 & NOT(1)+NOT(2) & x1 & x2 & null & null & y1 & 1 & 1 & null & null - -gate2 & NOT(1) & x1 & null & null & null & y2 & 10 & AND2 & x104 -”. Βλέπουμε λοιπόν ότι οι πληροφορίες που παίρνουμε είναι το όνομα της πύλης, η συνάρτησή της, τα ονόματα των εισόδων της (θεωρούμε ότι κάθε πύλη έχει 4 εισόδους. Αν έχει 1, 2, ή 3 εισόδους τότε οι άλλες εισοδοί αναφέρονται ως null), το όνομα της εξόδου της, και το επίπεδο στο οποίο βρίσκεται. Επιπλέον σε περίπτωση που η πύλη προέρχεται από τη διάσπαση κάποιας άλλης πύλης το αλφαριθμητικό αυτό μας πληροφορεί για τη συνάρτηση της συγκεκριμένης πύλης καθώς και για το όνομά της. Διαφορετικά στις αντίστοιχες θέσεις του αλφαριθμητικού αναγράφεται το υποαλφαριθμητικό null. Η τελευταία πληροφορία δε χρησιμοποιείται για τη σχεδίαση του κυκλώματος.

Το αλφαριθμητικό που χρησιμοποιείται για την περιγραφή ακολουθιακών κυκλωμάτων περιγράφει αρχικά τις πύλες του κυκλώματος και στη συνέχεια περιγράφει τα flip – flops. Δηλαδή το αλφαριθμητικό που θα λάβει το πρόγραμμά μας θα είναι της μορφής “gate1 & NOT(1)+NOT(2) & x1& x2 & null & null & y1 & 1 & null & null-..... -gate2& NOT(1)+NOT(2) & x1 & x3 & null & null& y2 & 10 & null & null -DFF1&y1&>null&>null&>null&x3-”. Οι πληροφορίες που παίρνουμε από το πρόγραμμα για το flip – flop είναι το όνομά του, η είσοδος του (θεωρούμε ότι και αυτό έχει 4 εισόδους καθώς χρησιμοποιούμε την ίδια δομή που χρησιμοποιούμε για να αναπαραστήσουμε τις πύλες), και την έξοδό του.

Το πρόγραμμά μας, αφού λάβει ένα αλφαριθμητικό με μορφή παρόμοια με αυτές που περιγράφηκαν παραπάνω, το επεξεργάζεται και δημιουργεί ένα Vector. Στοιχεία αυτού είναι πύλες και τα DFFs (αν αυτά υπάρχουν), τα οποία κατασκευάζονται από το πρόγραμμα με βάση τις πληροφορίες που αντλεί από το εν λόγω αλφαριθμητικό. Αν λάβει αλφαριθμητικό διαφορετικής μορφής τότε το κύκλωμα δε μπορεί να απεικονιστεί και ο χρήστης ενημερώνεται με σχετικό προειδοποιητικό μήνυμα.

Το πρόγραμμα μας ανασύρει τα στοιχεία του vector το ένα μετά το άλλο και αρχικά ελέγχει πόσες εισόδους έχει το κάθε στοιχείο. Όπως είδαμε και νωρίτερα οι δυνατές τιμές του αριθμού εισόδων ενός στοιχείου είναι μία, δύο, τρεις ή τέσσερις. Αν το στοιχείο έχει μια είσοδο τότε, ή θα είναι πύλη NOT ή θα είναι D flip flop. Αν έχει μεγαλύτερο αριθμό πυλών, τότε οι πιθανές τιμές της συνάρτησης είναι NAND

και NOR. Οπότε γίνεται ένας επιπλέον έλεγχος για να εξακριβωθεί ποια εικόνα πρέπει να σχεδιαστεί.

Όπως αναφέρθηκε και νωρίτερα τα διάφορα στοιχεία του κυκλώματος που ανήκουν π.χ. στο πρώτο επίπεδο σχεδιάζονται το ένα κάτω από το άλλο. Τα στοιχεία των άλλων επιπέδων σχεδιάζονται δίπλα σε αυτά του πρώτου επιπέδου. Δηλαδή το πρώτο στοιχείο του δεύτερου και κάθε επιπέδου σχεδιάζεται δίπλα στο πρώτο στοιχείο του πρώτου επιπέδου, και αυτό ισχύει για όλα τα στοιχεία. Στην ουσία δηλαδή οι πύλες τοποθετούνται σε ένα πλέγμα όπου κάθε στήλη αποτελεί ένα επίπεδο του κυκλώματος.

Αφού λοιπόν το πρόγραμμα αποφασίσει ποια εικόνα πρέπει να χρησιμοποιήσει, προχωρά στον υπολογισμό των τιμών των συντεταγμένων των διαφόρων εισόδων και εξόδων. Η διαδικασία αυτή γίνεται με βάση τη θέση των εισόδων και της εξόδου στην εικόνα, καθώς και με βάση τις διαστάσεις της τελευταίας. Έστω για παράδειγμα ότι έχουμε μια εικόνα που αναπαριστά μια πύλη NOT, της οποίας η πάνω αριστερή άκρη έχει συντεταγμένες (50,100). Αν η είσοδος της βρίσκεται 28 pixels κάτω, από το πάνω οριζόντιο τμήμα της εικόνας τότε οι συντεταγμένες της εισόδου είναι (50,128). Η διαδικασία επαναλαμβάνεται για τις εισόδους και εξόδους όλων των στοιχείων του κυκλώματος. Οι υπολογισμοί αυτοί, αν και δεν είναι τόσο σημαντικοί για να αναφερθούν, είναι απαραίτητοι για τη σχεδίαση των καλωδίων.

Για να υπάρχει επαρκής χώρος για το σχεδιασμό των καλωδίων μεταξύ δύο επιπέδων, καθώς όπως είδαμε κάθε κάθετο καλώδιο σχεδιάζεται λίγα χιλιοστά μακρύτερα από τα άλλα για να αποφευχθούν ταυτίσεις, ακολουθούμε την εξής διαδικασία. Για κάθε επίπεδο μετράμε τον αριθμό των πυλών που υπάρχουν σε αυτό και αφήνουμε διάστημα μεταξύ αυτού και του επόμενου επιπέδου, ίσο με τον αριθμό των πυλών που μετρήσαμε επί ένα έναν αριθμό pixels για κάθε πύλη και στο γινόμενο που προκύπτει προσθέτουμε έναν επιπλέον αριθμό από pixels.

Σχεδίαση των εισόδων των πυλών του κυκλώματος με τις εισόδους του:

Επόμενο βήμα αποτελεί η σύνδεση των εισόδων των διαφόρων πυλών με τις εξόδους του κυκλώματος. Αρχικά το κύκλωμα ελέγχει αν η είσοδος κάθε πύλης συνδέεται με έξοδο πύλης προηγούμενου επιπέδου. Αν όχι ελέγχει αν η πύλη βρίσκεται στο πρώτο επίπεδο. Αν η πύλη βρίσκεται στο πρώτο επίπεδο και δεν υπάρχει είσοδος του

κυκλώματος με όνομα ίδιο με το όνομα της εισόδου της πύλης, σχεδιάζεται το όνομα της εισόδου με κόκκινα γράμματα λίγα εκατοστά αριστερότερα της πύλης και από το σημείο αυτό ξεκινά ένα ευθύγραμμο τμήμα το οποίο καταλήγει στην είσοδο της πύλης.

Παρόμοια διαδικασία χρησιμοποιείται και όταν οι πύλες δε βρίσκονται στο ίδιο επίπεδο. Μόνο που σε αυτή την περίπτωση, προκειμένου να αποφύγουμε να σχεδιάσουμε καλώδια πάνω σε εικόνες πυλών προηγούμενων επιπέδων, όσο αυτό είναι δυνατόν, χρησιμοποιούμε δύο ή τρία ευθύγραμμο τμήματα. Πιο συγκεκριμένα το πρώτο (οριζόντιο) ευθύγραμμο τμήμα ξεκινά από το σημείο στο οποίο έχει σχεδιαστεί το όνομα της εισόδου (λίγα εκατοστά αριστερότερα του κυκλώματος και λίγα χιλιοστά κάτω από τις εικόνες των πυλών που ανήκουν στη γραμμή του πλέγματος που ανήκει και η εν λόγω πύλη), και φτάνει μέχρι το σημείο του οποίου η συντεταγμένη x , είναι ίση με τη συντεταγμένη x της εισόδου της εν λόγω πύλης, ή λίγα χιλιοστά αριστερότερα προκειμένου να αποφύγουμε την ταύτιση των κάθετων ευθύγραμμων τμημάτων που χρησιμοποιούνται για τη σύνδεση των εισόδων των διαφόρων πυλών του ίδιου επιπέδου. Το δεύτερο (κάθετο) ευθύγραμμο τμήμα ξεκινά από το τέλος του προαναφερόμενου ευθύγραμμου τμήματος και καταλήγει ή στην είσοδο της πύλης, ή λίγα χιλιοστά αριστερότερα, ανάλογα με τη συντεταγμένη y του τμήματος που περιγράψαμε αμέσως πιο πριν. Στην τελευταία περίπτωση χρησιμοποιείται ένα ακόμη ευθύγραμμο τμήμα, το οποίο καταλήγει στην είσοδο της πύλης.

Αν τώρα έχει ήδη σχεδιαστεί είσοδος του κυκλώματος με όνομα ίδιο με το όνομα της πύλης, η διαδικασία που πρέπει να ακολουθήσουμε είναι η ακόλουθη. Αρχικά ελέγχουμε και πάλι αν η είσοδος που μας ενδιαφέρει δεν συνδέεται με την έξοδο κάποιας πύλης ή κάποιου flip flop. Στη συνέχεια ελέγχουμε αν η πύλη που μας ενδιαφέρει βρίσκεται στο πρώτο επίπεδο. Έστω ότι υπάρχει μια είσοδος πύλης πρώτου επιπέδου που έχει συνδεθεί με μια είσοδο κυκλώματος και θέλουμε να συνδέσουμε άλλες δύο εισόδους πυλών (επίσης του πρώτου επιπέδου) με αυτή την είσοδο κυκλώματος. Από το όνομα της εισόδου του κυκλώματος ξεκινά ένα κάθετο ευθύγραμμο τμήμα μέχρι το σημείο που έχει συντεταγμένη y ίση με τη συντεταγμένη y της εισόδου της πύλης του κυκλώματος. Τώρα όσον αφορά τη σύνδεση της δεύτερης εισόδου, προεκτείνουμε το κάθετο τμήμα που χρησιμοποιήσαμε μέχρι το σημείο του οποίου η συντεταγμένη y , ταυτίζεται με τη συντεταγμένη y της εισόδου

της πύλης, και συνεχίζουμε όπως πριν. Με αυτό τον τρόπο μπορούμε να συνδέσουμε μεγάλο αριθμό εισόδων πυλών σε μια είσοδο κυκλώματος.

Το πρόβλημα που προκύπτει από αυτό τον τρόπο απεικόνισης, είναι το ότι τα διάφορα κάθετα τμήματα που θα ξεκινούν από κάθε είσοδο του κυκλώματος θα ταυτίζονται μεταξύ τους. Για αυτό το λόγο, κάθε κάθετο τμήμα που θα ξεκινά από κάποια είσοδο, θα σχεδιάζεται λίγα χιλιοστά δεξιότερα έτσι ώστε να μην προκαλείται σύγχυση.

Ας θεωρήσουμε την περίπτωση στην οποία η είσοδος κάποιας πύλης του πρώτου επιπέδου έχει συνδεθεί με κάποια είσοδο του κυκλώματος και θέλουμε να συνδέσουμε με αυτή την είσοδο κυκλώματος τις εισόδους δύο άλλων πυλών οι οποίες δε βρίσκονται στο πρώτο επίπεδο. Και σε αυτή την περίπτωση, η σύνδεση γίνεται με σχεδόν παρόμοιο τρόπο. Χρησιμοποιείται ένα κάθετο τμήμα το οποίο ξεκινά από κάποιο σημείο του τμήματος που συνδέει την προαναφερόμενη πύλη του πρώτου επιπέδου με την εν λόγω είσοδο κυκλώματος και εκτείνεται λίγα χιλιοστά πιο κάτω από τις εικόνες της συγκεκριμένης γραμμής του πλέγματος. Από το σημείο στο οποίο σταματά το τμήμα αυτό ξεκινά ένα άλλο οριζόντιο τμήμα, το οποίο εκτείνεται μέχρι το επίπεδο στο οποίο βρίσκεται η πύλη. Τέλος η είσοδος της πύλης συνδέεται με ένα κάθετο τμήμα με αυτό το σημείο. Αν στη συνέχεια θέλουμε να συνδέσουμε και την άλλη πύλη η οποία βρίσκεται σε επόμενο επίπεδο δεν έχουμε παρά να επεκτείνουμε το προαναφερόμενο οριζόντιο τμήμα.

Σύνδεση εξόδων πυλών με τις εισόδους πυλών επόμενου επιπέδου: Εφόσον έχουμε τελειώσει με την απεικόνιση των πυλών και τη σύνδεση των εισόδων των πυλών με τις εισόδους του κυκλώματος, επόμενο βήμα είναι η σχεδίαση των καλωδίων που συνδέουν τις εξόδους των πυλών με τις εισόδους πυλών άλλων επιπέδων, καθώς και των καλωδίων που συνδέουν εισόδους πυλών που βρίσκονται σε οποιοδήποτε άλλο επίπεδο εκτός από το πρώτο με τις εισόδους του κυκλώματος.

Για την εκπλήρωση αυτού του στόχου, το πρόγραμμα για κάθε έξοδο πύλης του Vector που περιγράφει το κύκλωμα, παίρνει το Vector με τους προορισμούς της εν λόγω εξόδου, (δηλαδή με τις πύλες των οποίων κάποια είσοδος πρέπει να συνδεθεί με τη συγκεκριμένη έξοδο). Αφού βρει τις συντεταγμένες της εξόδου και της επιθυμητής εισόδου κάθε πύλης προχωρά στη σχεδίαση των απαραίτητων καλωδίων.

Η σύνδεση μεταξύ εξόδων πυλών με εισόδους πυλών επόμενου επιπέδου, αποτελείται από τα ακόλουθα τμήματα καλωδίου. Έστω π.χ. ότι θέλουμε να

συνδέσουμε τις εξόδους των πυλών, που βρίσκονται στο επίπεδο 1. Ξεκινάμε από την πρώτη πύλη. Το πρώτο τμήμα είναι ένα κάθετο καλώδιο το οποίο ξεκινά από την έξοδο της πύλης και καταλήγει μερικά χιλιοστά κάτω από το ύψος του επιπέδου με τις περισσότερες πύλες. Το επόμενο τμήμα είναι ένα οριζόντιο καλώδιο που έχει ως αρχή το τελευταίο σημείο του προηγούμενου τμήματος και ως τέλος το σημείο που έχει συντεταγμένη x ίση με τη συντεταγμένη x της εισόδου την οποία έχει ως προορισμό. Και τέλος το τελευταίο τμήμα συνδέει το προηγούμενο ευθύγραμμο τμήμα με τη ζητούμενη είσοδο.

Τα πράγματα περιπλέκονται όταν πρέπει να συνδέσουμε την έξοδο της δεύτερης πύλης του συγκεκριμένου επιπέδου. Αν επιχειρήσουμε να ζωγραφίσουμε ένα κάθετο τμήμα από την έξοδο της δεύτερης πύλης αυτό θα ταυτιστεί με το ανάλογο τμήμα που είχε ως αρχή την έξοδο της πρώτης πύλης. Για να αποφύγουμε αυτό το πρόβλημα ξεκινούμε τη σχεδίαση ζωγραφίζοντας ένα οριζόντιο τμήμα από την έξοδο, μήκους λίγων χιλιοστών και στη συνέχεια σχεδιάζουμε το κάθετο τμήμα ξεκινώντας από το ελεύθερο άκρο του. Επειδή ανάλογο πρόβλημα υφίσταται και με τα οριζόντια τμήματα, το εν λόγω κάθετο τμήμα σχεδιάζεται λίγα χιλιοστά πιο κάτω από το ευθύγραμμο τμήμα που σχεδιάστηκε κατά τη σύνδεση της εξόδου της πρώτης πύλης. Αν στη συνέχεια η είσοδος στην οποία πρέπει να καταλήξει το καλώδιο βρίσκεται σε επίπεδο στο οποίο ήδη υπάρχει συνδεδεμένη είσοδος, τότε το τέλος του οριζόντιου τμήματος που προαναφέραμε πρέπει να έχει συντεταγμένη x , μικρότερη κατά λίγα από χιλιοστά από το τέλος του τμήματος το οποίο χρησιμοποιείται για τη σύνδεση της άλλης εισόδου. Επομένως το επόμενο τμήμα ξεκινά από το τέλος του συγκεκριμένου τμήματος και εκτείνεται μέχρι τη συντεταγμένη y της εισόδου. Και τέλος χρησιμοποιούμε ένα ακόμη ευθύγραμμο τμήμα που συνδέει το τμήμα αυτό με την είσοδο.

Την ίδια διαδικασία χρησιμοποιούμε για να συνδέσουμε και τις υπόλοιπες εισόδους του συγκεκριμένου επιπέδου και κάθε επιπέδου. Ανακεφαλαιώνοντας λοιπόν μπορούμε να πούμε ότι καθώς συνδέουμε την έξοδο μια επιπλέον πύλης ενός επιπέδου, η απόσταση του κάθετου καλωδίου από την πύλη αυξάνει κατά λίγα χιλιοστά. Επίσης κατά λίγα χιλιοστά αυξάνει και το οριζόντιο τμήμα που χρησιμοποιείται στη συγκεκριμένη καλωδίωση. Και όσον αφορά τη σύνδεση με την επιθυμητή είσοδο, για κάθε επιπλέον πύλη που συνδέεται, αυξάνει η απόσταση του κάθετου τμήματος του καλωδίου από την πύλη.

Επόμενο πρόβλημα που έπρεπε να λυθεί ήταν η σύνδεση με την έξοδο μιας πύλης με περισσότερες από μια εισόδους πυλών. Ας θεωρήσουμε ότι έχουμε ήδη συνδέσει την έξοδο μιας πύλης με την είσοδο μιας άλλης πύλης, και ότι θέλουμε να συνδέσουμε ακόμα μια είσοδο. Το πρώτο ευθύγραμμο τμήμα που πρέπει να σχεδιαστεί ξεκινά από το οριζόντιο τμήμα που χρησιμοποιήθηκε στην προηγούμενη σχεδίαση και εκτείνεται μέχρι τη συντεταγμένη x της εισόδου ή λίγα χιλιοστά πιο πριν, σύμφωνα με αυτά που αναφέρθηκαν παραπάνω. Έτσι συνδέουμε κάθε έξοδο με όλες τις εισόδους με τις οποίες πρέπει αυτή να συνδεθεί.

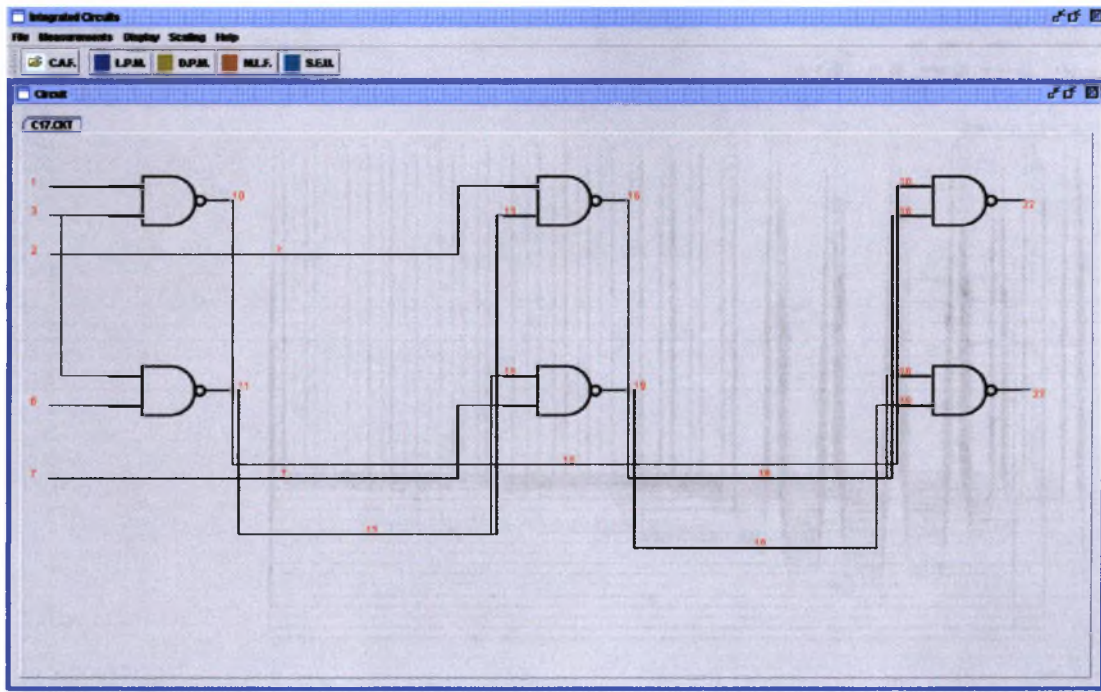
5.2.3 Ακολουθιακά κυκλώματα

Όσον αφορά το σχεδιασμό ακολουθιακών κυκλωμάτων όπως αναφέραμε και πιο πριν τα D flip flops είναι τοποθετημένα στο τελευταίο επίπεδο του κυκλώματος.

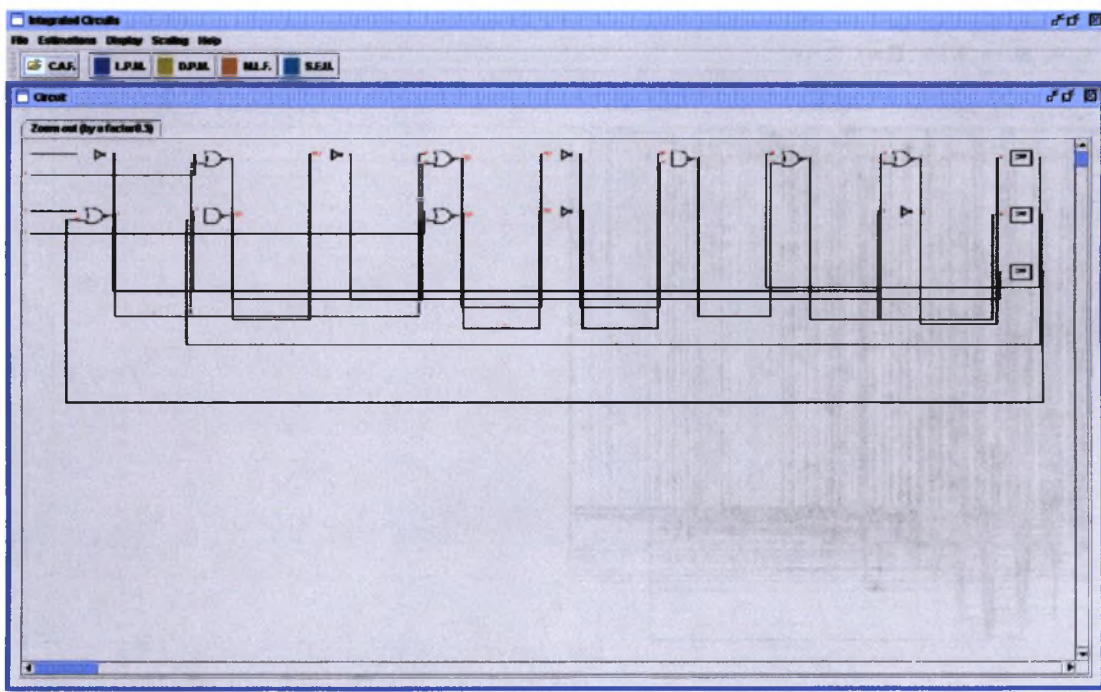
Για τη σύνδεση των εξόδων τους με τις εισόδους του κυκλώματος ακολουθείται παρόμοια διαδικασία με τις παραπάνω. Από την έξοδο κάθε D flip flop ξεκινά ένα ευθύγραμμο τμήμα και καταλήγει κάποια χιλιοστά ή εκατοστά κάτω από το ύψος του επιπέδου με τις περισσότερες πύλες. Η σύνδεση συνεχίζεται με ένα οριζόντιο τμήμα, το οποίο διασχίζει το κύκλωμα και καταλήγει λίγο αριστερότερα από τις πύλες του πρώτου επιπέδου. Βέβαια και σε αυτή την περίπτωση τα οριζόντια καλώδια σχεδιάζονται το ένα λίγα χιλιοστά πιο κάτω από το άλλο ενώ χρησιμοποιούνται και πάλι μικρά οριζόντια τμήματα στις εισόδους του κυκλώματος και στις εξόδους των flip flops προκειμένου να αποφύγουμε ταυτίσεις των κάθετων τμημάτων της σχεδίασης.

Για να αναπαραστήσουμε στην οθόνη όλα τα κάθετα καλώδια που συνδέουν τα οριζόντια τμήματα της σύνδεσης με τις εισόδους του κυκλώματος, ακολουθούμε την εξής διαδικασία. Μετράμε τον αριθμό των D flip flops του τελευταίου επιπέδου και προτού σχεδιάσουμε τις πύλες του πρώτου επιπέδου αφήνουμε διάστημα ίσο με το γινόμενο του αριθμού των DFFs επί έναν αριθμό από pixels για κάθε DFF.

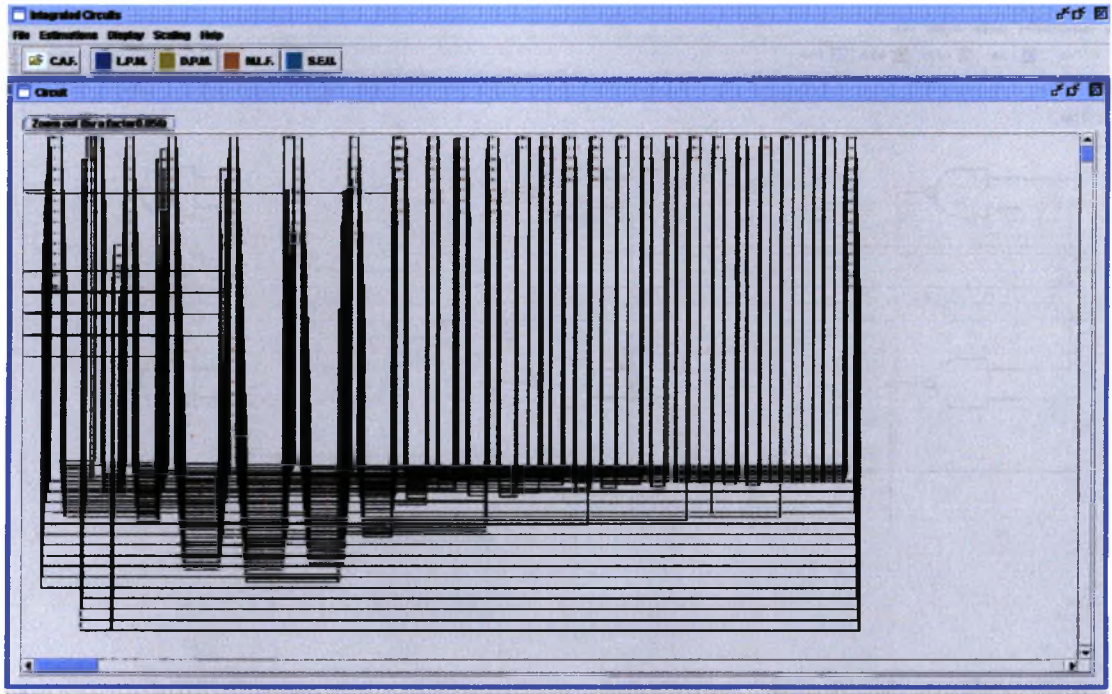
Στις παρακάτω εικόνες παρουσιάζονται ενδεικτικά κάποια κυκλώματα, τόσο συνδυαστικά, όσο και ακολουθιακά.



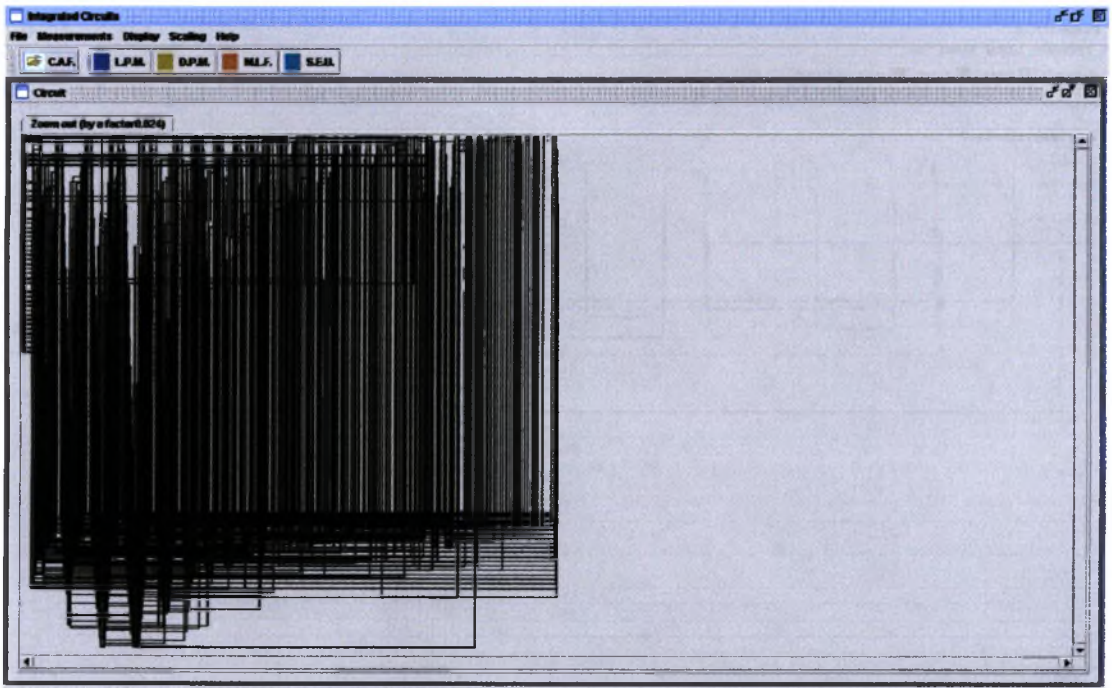
Το κύκλωμα C17.ckt



Το κύκλωμα S27.ckt με μείωση του μεγέθους της εικόνας κατά 0.3



Το κύκλωμα S349.ckt με μείωση του μεγέθους της εικόνας κατά 0.056



Το κύκλωμα S1238.ckt με μείωση του μεγέθους της εικόνας κατά 0.024

6. Σύγκριση μεταξύ γλωσσών προγραμματισμού και επιλογή της καταλληλότερης

Οι πλατφόρμες που κυκλοφορούν σήμερα στην αγορά είναι Windows, Linux και MAC. Τα Windows είναι αυτά που κυριαρχούν, καθώς προσφέρουν ένα μεγάλο σύνολο εφαρμογών και είναι εύκολα στη χρήση. Τα μειονεκτήματά τους είναι το ότι είναι ακριβά και το ότι παρουσιάζουν πολλά κενά στην ασφάλειά τους. Το Linux από την άλλη πλευρά είναι περισσότερο ασφαλές, πολλές από τις εκδόσεις του διανέμονται ελεύθερα, αλλά είναι αρκετά πιο δύσκολο στη χρήση του σε σύγκριση με τα Windows. Η πλατφόρμα MAC είναι εύκολη στη χρήση, καθώς και ασφαλής, αλλά υπάρχει ελάχιστο υλικό και λογισμικό συμβατό με αυτήν. Η έκδοση του δικού μας προγράμματος ζητάμε να είναι συμβατή με τα Windows, αλλά να μπορεί με λίγες αλλαγές να χρησιμοποιηθεί και στο Linux.

Πριν λοιπόν ξεκινήσουμε τη σχεδίαση, τέθηκε το ζήτημα επιλογής της γλώσσας σχεδίασης της εφαρμογής. Η επιλογή έγινε με βάση τη συμβατότητα της συγκεκριμένης γλώσσας με τις δυο προαναφερόμενες πλατφόρμες, καθώς και με βάση τα συστατικά σχεδίασης που αυτή προσφέρει και τη γενική προγραμματιστική συμπεριφορά της. Επίσης καθοριστικό ρόλο στην επιλογή έπαιξε και το πόσο εύκολη είναι η επικοινωνία της συγκεκριμένης γλώσσας με τη γλώσσα C++, στην οποία είναι υλοποιημένο το υπόλοιπο πρόγραμμα. Οι γλώσσες και τα εργαλεία που εξετάστηκαν ήταν η Java (βιβλιοθήκες AWT και Swing), η Visual Basic, η Visual C++, η Delphi και το πρόγραμμα σχεδίασης Qt designer (C++). Στη συνέχεια παρατίθεται μια περιγραφή των παραπάνω καθώς και μια σύγκριση μεταξύ αυτών και των συστατικών τους.

6.1 Java

Η Java αναπτύχθηκε από μέλη της εταιρίας SUN, και εδραιώθηκε στο χώρο του προγραμματισμού το 1995. Βασίζεται πάνω στη φιλοσοφία της C++. Πιο

συγκεκριμένα είναι και αυτή μια αντικειμενοστραφής γλώσσα και έχει παρόμοια δομή, αλλά είναι πολύ πιο απλή στη χρήση της.

Η Java είναι πλήρως ανεξάρτητη από το λειτουργικό σύστημα του υπολογιστή. Η μεταγλώττιση πηγαίου κώδικα σε Java οδηγεί στην παραγωγή ενός ειδικού κώδικα, γνωστού ως bytecode. Ο κώδικας αυτός έχει πολλές ομοιότητες με κώδικα γραμμένο σε γλώσσα μηχανής και μπορεί να εκτελεστεί σε οποιαδήποτε πλατφόρμα, στην οποία υπάρχει εγκατεστημένος κάποιος compiler της γλώσσας αυτής.

Επομένως οδηγούμαστε στο συμπέρασμα ότι αποτελεί πολύτιμο εργαλείο για όσους θέλουν να σχεδιάσουν εφαρμογές, οι οποίες να είναι συμβατές με περισσότερα από ένα λειτουργικά συστήματα, καθώς και με λειτουργικά συστήματα τα οποία πιθανών να σχεδιαστούν στο μέλλον. Όπως μάλιστα χαρακτηριστικά αναφέρεται για τα προγράμματα της Java, τα τελευταία « write once, run everywhere ».

Οι βιβλιοθήκες πακέτων που προσφέρει η Java για τη σχεδίαση γραφικών διεπαφών χρήστη είναι οι AWT και Swing, και SWT. Η τελευταία είναι μια βιβλιοθήκη η οποία αναπτύχθηκε από την IBM και προς το παρόν δε θα ασχοληθούμε με αυτήν. Καθεμιά από αυτές προσφέρει ένα μεγάλο σύνολο συστατικών σχεδίασης, έτσι ώστε να δίνονται στο χρήστη πολλές σχεδιαστικές επιλογές. Οι βιβλιοθήκες αυτές αναλύονται παρακάτω.

6.1.1 Η βιβλιοθήκη AWT

Η παραθυρική βιβλιοθήκη Abstract Window Toolkit περιέχει πάνω από 25 πακέτα. Μέσα σε αυτά ορίζονται πολλές κλάσεις και διασυνδέσεις που αφορούν την ανάπτυξη γραφικών διεπαφών, τη ζωγραφική κ.α. Παρά το γεγονός ότι περιέχει έναν αρκετά μεγάλο αριθμό συστατικών ανάπτυξης GUI, η βιβλιοθήκη αυτή θεωρείται περισσότερο κατάλληλη για τη σχεδίαση απλών Applets, (οι περισσότεροι Web browsers υποστηρίζουν τις κλάσεις του AWT, και κατά συνέπεια δεν είναι απαραίτητη η ύπαρξη plug-ins για την εκτέλεσή τους) και όχι για την ανάπτυξη ενός ολοκληρωμένου γραφικού περιβάλλοντος χρήστη. Ως σημαντικότερες υλοποιήσεις της βιβλιοθήκης AWT μπορούν να αναφερθούν αυτές που σχετίζονται με τους διαχειριστές διάταξης και το χειρισμό συμβάντων.

Οι διαχειριστές διάταξης είναι κλάσεις οι οποίες αναλαμβάνουν τη σχετική διάταξη των συστατικών σε μια εφαρμογή γραφικών. Αυτό διευκολύνει πολύ τους

προγραμματιστές, καθώς δεν είναι αναγκασμένοι να τοποθετούν τα συστατικά στην εφαρμογή δίνοντας τις συντεταγμένες τους. Επιπλέον είναι απαραίτητοι καθώς μπορούν να εγγυηθούν την επιθυμητή εμφάνιση μιας εφαρμογής καθώς αλλάζουμε το μέγεθός της, ή καθώς την εκτελούμε σε διαφορετικές πλατφόρμες.

Τα γεγονότα έχουν να κάνουν με την αλληλεπίδραση μεταξύ χρήστη και υπολογιστή. Στόχος τους είναι η γρήγορη και αποδοτική ανταπόκριση του προγράμματος σε ενέργειες του χρήστη. Τα γεγονότα δημιουργούνται όταν συμβαίνουν τέτοιες ενέργειες, φέρουν πληροφορίες τόσο για τα ίδια όσο και για την πηγή του γεγονότος και καλούν τις αρμόδιες μεθόδους για την αντιμετώπισή τους. Έτσι καθίσταται δυνατή η αλληλεπίδραση του προγράμματος με το χρήστη.

6.1.2 Η Βιβλιοθήκη Swing

Μια συνεργασία της SUN με άλλες εταιρίες και χρησιμοποιώντας ως βάση την τεχνολογία του AWT, είχε ως αποτέλεσμα την ανάπτυξη μιας βιβλιοθήκης πακέτων με το όνομα Swing. Η βιβλιοθήκη αυτή είναι από τα πιο σύνθετα εργαλεία ανάπτυξης γραφικών διεπαφών που αναπτύχθηκαν ποτέ. Περιλαμβάνει ένα πλήρες σύνολο συστατικών ανάπτυξης GUI, από τα πιο απλά και συνηθισμένα όπως κουμπιά και μενού μέχρι trees και παράθυρα λάθους. Με αυτό τον τρόπο παρέχει τη δυνατότητα για έναν πιο επαγγελματικό και πιο πλήρη τρόπο σχεδιασμού γραφικών διεπαφών.

Τα συστατικά του Swing αναπτύχθηκαν έχοντας ως βάση την ήδη υπάρχουσα τεχνολογία του AWT, και είναι υλοποιημένα εξ ολοκλήρου σε γλώσσα Java (για αυτό το λόγο χαρακτηρίζονται και ως lightweight).

Οι αρχικές εκδόσεις του Swing, περιείχαν ένα μεγάλο αριθμό από bugs και λάθη εκτέλεσης, τα οποία έκαναν δύσκολη τη χρήση του και δυσχέραιναν την εκτέλεση των όποιων προγραμμάτων. Επίσης είχε αναφερθεί το ότι οι εφαρμογές Swing ήταν αρκετά αργές ή το ότι χρησιμοποιούσαν ένα αρκετά μεγάλο ποσοστό της μνήμης κατά τη διάρκεια της εκτέλεσής τους.

Τώρα όμως τα περισσότερα bugs έχουν διορθωθεί και το νέο υλικό έχει δώσει λύση στην καθυστέρηση του Swing, καθιστώντας την πλέον πολύ μικρή. Τα γεγονότα αυτά σε συνδυασμό με το ότι πολλές νέες πλατφόρμες ανάπτυξης γραφικών

διεπαφών του εμπορίου βασίζονται στο Swing, καθίστα τη βιβλιοθήκη αυτή ως ένα από τα σημαντικότερα εργαλεία ανάπτυξης γραφικών περιβαλλόντων διασύνδεσης.

6.1.3 AWT vs Swing.

Στη συνέχεια θα αναφερθούμε στα πλεονεκτήματα και στα μειονεκτήματα του καθενός έτσι ώστε να μπορέσουμε να επιλέξουμε το καταλληλότερο για τη σχεδίασή μας.

Όπως είδαμε το Swing περιέχει βελτιωμένες εκδόσεις όλων των συστατικών της βιβλιοθήκης AWT, καθώς και πολλά νέα υψηλότερου επιπέδου συστατικά. Κατά συνέπεια το Swing παρέχει τη δυνατότητα για τη δημιουργία περισσότερο ολοκληρωμένων και περισσότερο επαγγελματικών διεπαφών χρήστη. Επιπλέον πολλά από τα νέα συστατικά του Swing, όπως για παράδειγμα οι λεζάντες που εμφανίζονται αν τοποθετήσουμε το ποντίκι πάνω σε ένα κουμπί και περιγράφουν τη λειτουργία που θα συμβεί αν το πατήσουμε (*tool tip texts*), κάνουν το γραφικό περιβάλλον πιο αποδεκτό και πιο φιλικό από και προς το χρήστη.

Ακόμη το γεγονός ότι η συντριπτική πλειοψηφία των συστατικών του Swing, είναι *lightweight*, δίνει το πλεονέκτημα της ανεξαρτησίας τις διάφορες πλατφόρμες. Ως επέκταση αυτής της ανεξαρτησίας, η εμφάνιση των Swing εφαρμογών μπορεί να είναι η ίδια σε οποιαδήποτε πλατφόρμα. Δηλαδή μια εφαρμογή η οποία έχει το στυλ των Windows, είναι δυνατό, αν φυσικά το θελήσουμε, να έχει το ίδιο στυλ ακόμη και όταν την εκτελούμε στο Linux.

Σε αντίθεση τα συστατικά του AWT, είναι εξαρτημένα από το εκάστοτε λειτουργικό σύστημα, καθώς η υλοποίηση της λειτουργικότητάς τους βασίζεται σε *native peers*. Έτσι η εμφάνιση των AWT εφαρμογών δε μπορεί να αλλάζει με δυναμικό τρόπο όπως συμβαίνει στις εφαρμογές, που έχουν αναπτυχθεί με τη βοήθεια του Swing. Ένα άλλο αρνητικό που προκύπτει από τον τρόπο υλοποίησης των συστατικών του AWT, είναι το γεγονός ότι υπάρχουν συστατικά των οποίων η λειτουργικότητα δε μπορεί να υποστηριχτεί πλήρως σε ορισμένες πλατφόρμες.

Βέβαια τα *heavyweight* συστατικά του AWT έχουν κι αυτά τα πλεονεκτήματά τους. Ένα από αυτά είναι το γεγονός ότι η εξάρτηση της εφαρμογής από τα *native peers*, της παρέχει ένα *look and feel*, απόλυτα συμβατό με το *look and feel* της πλατφόρμας εκτέλεσης, πράγμα το οποίο δε μπορεί να πετύχει το Swing. Κι αυτό

γιατί ακόμη και στην περίπτωση που θέσουμε στην εφαρμογή Swing το look and feel της συγκεκριμένης πλατφόρμας, η εμφάνισή της θα είναι σε μικρότερο βαθμό εναρμονισμένη με το υπόλοιπο σύστημα από την ανάλογη του AWT, καθώς η τελευταία βασίζεται σε peers του ίδιου του συστήματος.

6.1.4 Συμπεράσματα

Λαμβάνοντας υπόψη όλα τα παραπάνω καταλήγουμε στο ακόλουθο συμπέρασμα. Η βιβλιοθήκη πακέτων AWT ενδείκνυται για εφαρμογές οι οποίες προορίζονται για εκτέλεση σε μια μόνο συγκεκριμένη πλατφόρμα. Σε όλες τις άλλες περιπτώσεις καλύτερη λύση θεωρείται η βιβλιοθήκη Swing, με ταυτόχρονη την υποστήριξη της βιβλιοθήκης AWT σε θέματα χειρισμού γεγονότων και διαχειριστών διάταξης (Η βιβλιοθήκη Swing περιέχει το πακέτο event το οποίο προορίζεται για χειρισμό γεγονότων. Υπάρχει όμως υποστήριξη μόνο για τα συστατικά του Swing, για τα οποία δεν υπάρχουν αντίστοιχες εκδόσεις στη βιβλιοθήκη AWT. Ακόμα περιέχει μόνο ένα διαχειριστή διάταξης. Για όλα τα υπόλοιπα συστατικά και τους υπόλοιπους διαχειριστές είναι απαραίτητη η χρήση πακέτων του AWT.).

Βέβαια όταν σχεδιάζουμε μια εφαρμογή με τη χρήση της βιβλιοθήκης Swing δε συνίσταται η χρήση συστατικών της βιβλιοθήκης AWT, πέρα από αυτά που αναφέρθηκαν παραπάνω. Κι αυτό γιατί τα heavyweight συστατικά του AWT, θα τοποθετηθούν πάνω από τα lightweight, με αποτέλεσμα τα τελευταία να καταστούν αόρατα.

6.2 Visual Basic

Η Visual Basic αναπτύχθηκε από τη Microsoft και χρησιμοποιείται μεταξύ άλλων και για την ανάπτυξη γραφικών περιβαλλόντων διασύνδεσης. Είναι μια γλώσσα απλή και εύκολη στη μάθηση σε σύγκριση με άλλες γλώσσες, καθώς και γρήγορη στην ανάπτυξη εφαρμογών. Εφαρμογές των οποίων ο προγραμματισμός σε άλλες γλώσσες απαιτεί κόπο, χρόνο και αρκετές γραμμές κώδικα, στη Visual Basic αναπτύσσονται με λιγότερη προσπάθεια και σε μικρότερο χρονικό διάστημα.

Εξαιτίας της ευκολίας στην εκμάθηση η γλώσσα αυτή έχει γίνει ιδιαίτερα αγαπητή, με αποτέλεσμα να υπάρχουν πολλές πηγές και πολλά εργαλεία τα οποία

μπορεί κάποιος να χρησιμοποιήσει προκειμένου να μάθει τη γλώσσα (αν δεν τη γνωρίζει ήδη) και να τη χρησιμοποιήσει όσο γίνεται πιο αποδοτικά για την ανάπτυξη των εφαρμογών του. Ακόμα υποστηρίζει αρκετά συστατικά σχεδίασης έτσι ώστε να δίνει αρκετές σχεδιαστικές επιλογές στους προγραμματιστές.

Από την άλλη πλευρά όμως η Visual Basic έχει και αρκετά μειονεκτήματα. Για παράδειγμα είναι αρκετά πιο αργή στην εκτέλεση των προγραμμάτων της και παρέχει μειωμένη λειτουργικότητα συγκριτικά με άλλες γλώσσες, όπως π.χ. η C++. Επιπλέον τα συστατικά που χρησιμοποιεί είναι υλοποιημένα συνήθως σε C/C++, καθιστώντας έτσι πολύπλοκη την ανάπτυξη συστατικών από τον ίδιο το χρήστη. Τα τελευταία χρόνια δε χρησιμοποιείται αρκετά συχνά καθώς έχουν αναπτυχθεί πιο ολοκληρωμένες ως προς τη λειτουργικότητά τους γλώσσες.

Λαμβάνοντας όλα αυτά υπόψη οδηγούμαστε στα παρακάτω συμπεράσματα. Η Visual Basic ενδείκνυται κυρίως για νέους προγραμματιστές, έτσι ώστε να αποκτήσουν τις πρώτες γνώσεις και να εξοικειωθούν με αυτού του είδους τη σχεδίαση, και όχι για την ανάπτυξη ολοκληρωμένων γραφικών διεπαφών.

Επιπλέον η Visual Basic αποτελεί καλή επιλογή για εφαρμογές οι οποίες θα εκτελούνται αποκλειστικά σε windows, σε σύγκριση με άλλες γλώσσες. Κι αυτό γιατί λόγω του γεγονότος ότι αποτελεί και αυτή προϊόν της Microsoft, η λειτουργικότητα και το look and feel των εφαρμογών της θα είναι απόλυτα σύμφωνο με αυτό των windows.

6.3 Delphi

Η Delphi είναι ένα εργαλείο προγραμματισμού που αναπτύχθηκε αρχικά από τη Borland και που χρησιμοποιείται κυρίως για τη σχεδίαση παραθυρικών εφαρμογών. Η Delphi είναι αρκετά δημοφιλής καθώς συνδυάζει την ευχρηστία της Visual Basic, με την αυξημένη λειτουργικότητα και τη δύναμη της C++. Η γλώσσα αυτή αποτελεί στην πραγματικότητα μια βελτιωμένη έκδοση της γλώσσας Pascal, η οποία υποστηρίζει πολλά νέα χαρακτηριστικά.

Η Delphi περιέχει και αυτή ένα επαρκές σύνολο συστατικών σχεδίασης, τα οποία τοποθετούνται στην εφαρμογή μέσω drag and drop. Για υψηλότερου επιπέδου εφαρμογές, όπου τα standard συστατικά της γλώσσας δεν επαρκούν, υπάρχει στο

internet μια πληθώρα υψηλότερου επιπέδου χαρακτηριστικών , πολλά από τα οποία διανέμονται δωρεάν.

Ακόμα η Delphi παρέχει την ικανότητα διαμόρφωσης και τροποποίησης της εμφάνισης των συστατικών σχεδίασης, όπως το γεγονός ότι το μέγεθος ενός κουμπιού δύναται να καθοριστεί από το χρήστη. Αυτού του είδους η δυνατότητα δεν παρέχεται από πολλές γλώσσες. Για παράδειγμα η Visual Basic επιτρέπει απλά την τοποθέτηση των συστατικών και όχι τη μορφοποίησή τους.

Το παραπάνω γεγονός σε συνδυασμό με το ότι δεν είναι απαραίτητη η συγγραφή μεγάλου τμήματος κώδικα, καθώς τα διάφορα συστατικά σχεδίασης τοποθετούνται στην εφαρμογή μέσω drag and drop, έχει ως αποτέλεσμα να μην απαιτείται σημαντικός χρόνος για τα βασικά της σχεδίασης, έτσι ώστε ο προγραμματιστής να μπορεί να αφιερώσει αρκετό χρόνο στις λεπτομέρειες. Επομένως καταλήγουμε στο συμπέρασμα ότι η ανάπτυξη των εφαρμογών στη γλώσσα αυτή είναι εύκολη και γρήγορη.

Τα συστατικά της γλώσσας Delphi έχουν υλοποιηθεί μέσα στο ίδιο το περιβάλλον ανάπτυξης εφαρμογών, σε αντίθεση με ορισμένες άλλες γλώσσες όπου τα συστατικά έχουν δημιουργηθεί με τη βοήθεια εξωτερικών μεταγλωττιστών. Κάποια από τα συστατικά αυτά μπορούν να λειτουργήσουν ως πρότυπα για την ανάπτυξη standard εφαρμογών όπως για παράδειγμα ένα παράθυρο διαλόγου. Η λειτουργικότητα των συστατικών αυτών μπορεί να επεκταθεί έτσι ώστε ο χρήστης να μπορεί να δημιουργήσει τα δικά του συστατικά δίνοντάς τους τη συμπεριφορά που επιθυμεί, και στη συνέχεια να τα προσθέσει στο υπάρχον σύνολο προτύπων.

Η Delphi έχει αναπτυχθεί για να χρησιμοποιείται σε περιβάλλον windows. Επομένως οι εφαρμογές που έχουν αναπτύσσονται σε αυτή τη γλώσσα μπορούν να εκτελεστούν μόνο σε αυτό το λειτουργικό σύστημα. Βέβαια υπάρχει και μια έκδοση της Delphi για το Linux, η οποία αναφέρεται ως Kylix. Όμως και σε αυτή την περίπτωση οι εφαρμογές που αναπτύσσονται στο Linux, δεν είναι συμβατές με τα windows.

6.4 C++

Η γλώσσα C++, είναι μια από τις παλιότερες γλώσσες και έχει μεγάλο κύρος στο χώρο του προγραμματισμού. Παρέχει πολλές δυνατότητες καθώς και αυξημένη

λειτουργικότητα στις εφαρμογές της και είναι από τις πιο γρήγορες γλώσσες όσον αφορά την εκτέλεση των προγραμμάτων της. Ακόμα μπορεί να υποστηρίξει αυξημένη λειτουργικότητα καθώς υπάρχουν διαθέσιμες πολλές βιβλιοθήκες για C++, με επιπλέον χαρακτηριστικά.

Παρόλα αυτά βέβαια έχει και κάποια μειονεκτήματα. Λόγω κάποιων χαρακτηριστικών της, όπως για παράδειγμα οι δείκτες, την καθιστούν αρκετά δύσκολη στη χρήση και αρκετά χρονοβόρα κατά την ανάπτυξη των εφαρμογών της. Επιπλέον τέτοια χαρακτηριστικά είναι δυνατό να αποτελέσουν πηγή λαθών (που σε άλλες περιπτώσεις θα είχαν αποφευχθεί) η διόρθωση των οποίων μπορεί να είναι δύσκολη, ή ακόμη και αδύνατη.

6.4.1 Visual C++

Η Visual C++ αποτελεί και αυτή προϊόν της Microsoft. Βασίζεται στη γλώσσα προγραμματισμού C++, και χρησιμοποιείται ευρύτατα για την ανάπτυξη γραφικών περιβαλλόντων διασύνδεσης.

Χρησιμοποιεί βιβλιοθήκες, όπως η MFC (Microsoft Foundation Classes) οι οποίες περιέχουν ένα ευρύ σύνολο συστατικών σχεδίασης και σχετικών ελέγχων, προσφέροντας έτσι στους προγραμματιστές αυξημένη ευελιξία και λεπτομέρεια κατά την ανάπτυξη, καθώς και πρόσθετη λειτουργικότητα στα προγράμματά τους.

Βασίζεται στη γλώσσα προγραμματισμού C++, η οποία είναι μια γλώσσα με πολλά επιθυμητά χαρακτηριστικά. Όπως αναφέρεται πιο πάνω είναι από τις πιο γρήγορες γλώσσες κατά την εκτέλεση των προγραμμάτων της και είναι σε θέση να υποστηρίξει πολλές λειτουργίες. Έτσι και η Visual C++, κληρονομώντας τα χαρακτηριστικά της παρέχει αυξημένη λειτουργικότητα στις διάφορες εφαρμογές και συνίσταται για πιο σύνθετες σχεδιάσεις.

Όπως αναφέρθηκε και πιο πάνω για τη Visual Basic, αποτελεί και αυτή καλή επιλογή για εφαρμογές των Windows, καθώς έχει σχεδιαστεί από τη Microsoft, ώστε να είναι απόλυτα σύμφωνη με τα χαρακτηριστικά αυτών.

6.4.2 Qt designer

Το Qt designer είναι ένα εργαλείο σχεδίασης γραφικών, το οποίο χρησιμοποιεί τη γλώσσα C++, και αναπτύχθηκε από την TrollTech. Αποτελεί μέρος

του Qt, μιας ολοκληρωμένης εφαρμογής C++, η οποία αποτελείται από μια βιβλιοθήκη τάξεων καθώς και από εργαλεία διεθνοποίησης και ανάπτυξης cross – platform εφαρμογών.

Το Qt designer είναι ένα από τα πιο εξελιγμένα εργαλεία όσον αφορά τη σχεδίαση διεπαφών χρήστη. Χρησιμοποιείται για την ανάπτυξη cross – platform γραφικών περιβαλλόντων χρήστη παρέχοντας παράλληλα τη δυνατότητα για αυτόματη μετακίνηση και αναβάθμιση των συστατικών της σχεδίασης καθώς και των απαραίτητων ελέγχων. Οι παραγόμενες διεπαφές είναι ταυτόχρονα λειτουργικές και απόλυτα συμβατές με το look and feel της πλατφόρμας του μηχανήματος (οι εφαρμογές μοιάζουν με native εφαρμογές και επομένως έχουν και τα χαρακτηριστικά τους).

Εφαρμογές του Qt designer μπορούν να αναπτυχθούν σε όλα τα λειτουργικά συστήματα. Εκδόσεις του Qt κυκλοφορούν στο εμπόριο για τα ακόλουθα λειτουργικά συστήματα: windows, Linux, macintosh, και embedded (embedded Linux). Μια όχι και τόσο ολοκληρωμένη έκδοση του προγράμματος είναι ενσωματωμένη σε αρκετές εκδόσεις του Linux. Επίσης παρέχεται και μια δωρεάν έκδοση αυτού για windows.

Το Qt designer είναι εύκολο στη χρήση, καθώς πέρα από μια μεγάλη γκάμα συστατικών σχεδίασης, υπάρχει ένα μόνο API, το οποίο μπορούν να συμβουλευούνται και να χρησιμοποιούν οι προγραμματιστές κατά την ανάπτυξη των εφαρμογών τους, ανεξάρτητα από την πλατφόρμα στην οποία προγραμματίζουν. Άρα οι χρήστες είναι ελεύθεροι να αναπτύσσουν τις εφαρμογές τους σε οποιαδήποτε πλατφόρμα είναι πιο φιλική σε αυτούς.

Οι εφαρμογές του εργαλείου αυτού δεν απαιτούν κάποια virtual machine, ή κάτι ανάλογο για την εκτέλεσή τους. Κι αυτό γιατί το Qt είναι υλοποιημένο με χαμηλού επιπέδου συναρτήσεις γραφικών, όπως και οι εφαρμογές native. Έτσι οι εφαρμογές του Qt, δε διαφέρουν και πολύ από τις native εφαρμογές και εκτελούνται με την ίδια ταχύτητα που εκτελούνται και οι τελευταίες δηλαδή αρκετά γρήγορα.

Επιπλέον για υψηλού επιπέδου εφαρμογές οι οποίες απαιτούν χαρακτηριστικά πέραν αυτών που προσφέρει το ίδιο το εργαλείο, υπάρχουν πολλά επιπλέον widgets και συστατικά για το Qt, τα οποία είναι σε θέση να παρέχουν την απαιτούμενη λειτουργικότητα.

Όπως είδαμε πιο πάνω το Qt designer χρησιμοποιεί τη γλώσσα C++. Κατά συνέπεια οι εφαρμογές που αναπτύσσονται μέσω του εργαλείου αυτού να απολαμβάνουν όλα τα πλεονεκτήματα της γλώσσας αυτής. Πιο συγκεκριμένα είναι

αρκετά γρήγορες, έχουν υψηλή λειτουργικότητα και μπορούν να χρησιμοποιήσουν και επιπρόσθετα χαρακτηριστικά της C++, π.χ. `socket classes`, σε περίπτωση επέκτασης τους σε δικτυακό επίπεδο.

Παρά το γεγονός ότι στη συνέχεια της εργασίας παρουσιάζεται συνολική σύγκριση μεταξύ των προαναφερόμενων γλωσσών και εργαλείων, είναι αρκετά ενδιαφέρον να κάνουμε μια επιμέρους σύγκριση μεταξύ των δυο προηγούμενων εργαλείων, του Qt designer και της Visual C++.

6.4.3 Qt designer vs Visual C++

Η σημαντικότερη διαφορά αφορά στη συμβατότητα του κάθε εργαλείου με τα διάφορα λειτουργικά συστήματα. Η Visual C++ ως προϊόν της Microsoft είναι συμβατή μόνο με τα windows. Έτσι καθίσταται ακατάλληλη για εφαρμογές προορισμένες να εκτελούνται σε περισσότερες από μια πλατφόρμες όπως για παράδειγμα το δικό μας τελικό πρόγραμμα. Από την άλλη πλευρά υπάρχουν εκδόσεις του Qt designer για τα λειτουργικά συστήματα που μας ενδιαφέρουν, και οι εφαρμογές που αναπτύσσονται με τη βοήθεια αυτού μπορούν να εκτελεστούν χωρίς προβλήματα σε οποιαδήποτε πλατφόρμα

Μια άλλη διαφορά αφορά το χειρισμό των γεγονότων. Στην Visual C++, η επικοινωνία με τα συστατικά του προγράμματος γίνεται μέσω μηνυμάτων που στέλνονται από το λειτουργικό σύστημα στα διάφορα παράθυρα. Ο αριθμός των μηνυμάτων αυτών είναι υπερβολικά μεγάλος, έτσι ώστε ο προγραμματιστής να δυσκολεύεται να καταλάβει πιο είναι το καταλληλότερο για να λύσει το πρόβλημά του, καθώς είναι δύσκολο να ξέρει την ακριβή λειτουργικότητα του καθενός. Ακόμα είναι αρκετά δύσκολο να γνωρίζει κανείς αν έχουν παραδοθεί όλα τα μηνύματα, αν έχουν παραδοθεί στα σωστά παράθυρα με αποτέλεσμα να μη μπορεί κανείς να προβλέψει με βεβαιότητα την ακριβή λειτουργικότητα των εφαρμογών του.

Από την άλλη πλευρά το Qt designer για αυτό το σκοπό περιέχει έναν μηχανισμό ο οποίος βασίζεται στην εκπομπή και τη λήψη των μηνυμάτων σε χρονικές θυρίδες. Η επικοινωνία μεταξύ των αντικειμένων γίνεται μέσω του συστήματος. Ο προγραμματιστής στέλνει τα ανάλογα σήματα κατά τη διάρκεια της συγκεκριμένης χρονικής θυρίδας. Έτσι ξέρει ποιος είναι ο αποδέκτης τους. Ακόμη ο αριθμός των μηνυμάτων που εκπέμπονται από κάποια κλάση είναι μικρός (συνήθως 4

ή 5). Έτσι λοιπόν γίνεται εμφανές ότι μέσω του Qt designer ο προγραμματιστής μπορεί να πετύχει καλύτερο έλεγχο των προγραμμάτων του. Ο μηχανισμός αυτός που χρησιμοποιεί το Qt designer μοιάζει αρκετά με τις υλοποιήσεις των listeners στη Java.

Μια άλλη σημαντική διαφορά οφείλεται στην ύπαρξη ή όχι διαχειριστών διάταξης. Η Visual C++, δε χρησιμοποιεί διαχειριστές διάταξης. Αυτό δημιουργεί πρόβλημα σε περίπτωση που θελήσουμε να αλλάξουμε το μέγεθος κάποιου παραθύρου (η εμφάνισή του θα αλλοιωθεί). Επίσης πρόβλημα δημιουργείται και στην περίπτωση που μια εφαρμογή πρέπει να μεταφραστεί σε κάποια άλλη γλώσσα, καθώς κάθε γλώσσα μπορεί να χρησιμοποιεί διαφορετικού μεγέθους φράσεις για να εκφράσει ορισμένες καταστάσεις. Η μόνη λύση για να βελτιώσουμε την εμφάνιση του παραθύρου σε αυτές τις περιπτώσεις, είναι να αναδιατάξουμε τα συστατικά με το χέρι κάτι που είναι πολύ κουραστικό και μη πρακτικό. Από την άλλη πλευρά το Qt designer χρησιμοποιεί διαχειριστές διάταξης με αποτέλεσμα να εξαλείφονται όλα τα παραπάνω προβλήματα.

6.5 Συνολική σύγκριση των προαναφερόμενων γλωσσών και εργαλείων.

Παρακάτω παρουσιάζονται συγκεντρωμένα τα πλεονεκτήματα και τα μειονεκτήματα των παραπάνω γλωσσών και εργαλείων έτσι ώστε να έχουμε μια καλύτερη άποψη για το ποια είναι τελικά η καταλληλότερη γλώσσα για την επίλυση του προβλήματός μας.

Java (Χρήση Swing με την υποστήριξη του AWT)

Πλεονεκτήματα:

- 1) Συμβατή με όλα τα υπάρχοντα λειτουργικά συστήματα – write once, run everywhere.
- 2) Εύκολη στη χρήση και στην ανάπτυξη εφαρμογών.
- 3) Σχετικά εύκολη στη μάθηση, αν υπάρχει προηγούμενη εμπειρία στον προγραμματισμό.
- 4) Μεγάλο σύνολο προηγμένων συστατικών σχεδίασης και ελέγχων → αυξημένης λειτουργικότητας γραφικά περιβάλλοντα διασύνδεσης
- 5) Χρήση διαχειριστών διάταξης

- 6) Εύκολος τρόπος χειρισμού των γεγονότων

Μειονεκτήματα:

- 1) η ταχύτητα → σχετικά αργή γλώσσα
- 2) Απαραίτητη η χρήση μηχανισμού για την επικοινωνία της Java με τη C++, στην οποία είναι γραμμένο το υπόλοιπο πρόγραμμά μας .

Visual Basic

Πλεονεκτήματα:

- 1) εύκολη στη μάθηση
- 2) γρήγορη στην ανάπτυξη μιας εφαρμογής
- 3) επικοινωνία με Visual

Μειονεκτήματα:

- 1) πιο αργή από αρκετές γλώσσες
- 2) δεν παρέχει υψηλή λειτουργικότητα
- 3) Μη χρήση διαχειριστών διάταξης
- 4) προϊόν της Microsoft, μη συμβατές οι εφαρμογές της με άλλα λειτουργικά συστήματα πέραν των Windows

Delphi

Πλεονεκτήματα:

- 1) Εύκολη στη μάθηση και στον προγραμματισμό εφαρμογών.
- 2) Γρήγορα περιβάλλοντα ανάπτυξης εφαρμογών
- 3) Εκδόσεις του εργαλείου και για το λειτουργικό σύστημα windows καθώς και για το Linux (kylinx).

Μειονεκτήματα:

- 1) Όχι τόσο προσαρμόσιμη στις διάφορες εφαρμογές
- 2) Μη χρήση διαχειριστών διάταξης
- 3) Ανάπτυξη των εφαρμογών σε windows – συμβατότητα μόνο με windows.
- 4) Ανάπτυξη των εφαρμογών σε Linux – συμβατότητα μόνο με Linux.

Visual C++

Πλεονεκτήματα:

- 1) Χρήση βιβλιοθηκών, με πολλά συστατικά σχεδίασης και ελέγχους
- 2) Χρήση της γλώσσας C++ → γρήγορες, ευέλικτες με υψηλή λειτουργικότητα εφαρμογές, δυνατότητα χρήσης βιβλιοθηκών της C++.

- 3) Δεν τίθεται πρόβλημα επικοινωνίας μεταξύ γλωσσών, καθώς το υπόλοιπο πρόγραμμα είναι υλοποιημένο σε C++.

Μειονεκτήματα:

- 1) Προϊόν της Microsoft – μη συμβατές οι εφαρμογές της με άλλα λειτουργικά συστήματα πέραν των Windows.
- 2) Χρήση της C++ → δύσκολη και χρονοβόρα η ανάπτυξη των εφαρμογών.
- 3) Μη χρήση διαχειριστών διάταξης

Qt designer

Πλεονεκτήματα:

- 1) Γραφική διεπαφή που στηρίζεται σε κάποια standard πρότυπα.
- 2) Γρήγορες εφαρμογές GUI ειδικά σε σύγκριση με αυτές της Java.
- 3) Χρήση της γλώσσας C++, δυνατή η χρήση πολλών επιπλέον συστατικών, όπως π.χ. Socket classes σε περιπτώσεις επέκτασης του προγράμματος.
- 4) Εκδόσεις του εργαλείου αυτού τόσο για το λειτουργικό σύστημα windows, όσο και για το Linux.
- 5) Δεν τίθεται πρόβλημα επικοινωνίας μεταξύ γλωσσών, καθώς το υπόλοιπο πρόγραμμα είναι υλοποιημένο σε C++.
- 6) Συμβατότητα των εφαρμογών με οποιαδήποτε πλατφόρμα

Μειονεκτήματα:

- 1) Χρήση της C++ → δύσκολη και χρονοβόρα η ανάπτυξη των εφαρμογών.
- 2) Απαραίτητη η απόκτηση ενός license για κάθε χρήστη για απόκτηση επαγγελματικών εκδόσεων του εργαλείου αυτού.
- 3) Απαραίτητη η χρήση επιπλέον βιβλιοθηκών για συμβατότητα με άλλα λειτουργικά συστήματα.

6.6 Τελική επιλογή γλώσσας προγραμματισμού

Μετά την περιγραφή όλων των παραπάνω είμαστε πλέον σε θέση να επιλέξουμε ποιο είναι το καταλληλότερο εργαλείο για την ανάπτυξη της εφαρμογής μας. Όπως αναφέρθηκε στην αρχή της παρούσας πτυχιακής εργασίας στόχος μας είναι να δημιουργήσουμε ένα γραφικό περιβάλλον διασύνδεσης το οποίο θα πρέπει

να επικοινωνεί με ένα πρόγραμμα γραμμένο σε C++, και να είναι συμβατό τόσο με το λειτουργικό σύστημα windows, όσο και με το Linux.

Μια γλώσσα την οποία εξαρχής θεωρείται σκόπιμο να απορρίψουμε είναι η Visual Basic. Κι αυτό γιατί από όλα τα παραπάνω γίνεται φανερό ότι η τελευταία είναι μια γλώσσα για τη σχεδίαση σχετικά απλών γραφικών εφαρμογών και η οποία απευθύνεται σε νέους προγραμματιστές έτσι ώστε να αποκτήσουν κάποιες αρχικές γνώσεις στη σχεδίαση περιβαλλόντων αλληλεπίδρασης. Επιπλέον η Visual Basic αποτελεί προϊόν της Microsoft, κατά συνέπεια μπορεί να χρησιμοποιηθεί μόνο για εφαρμογές των Windows (υπάρχουν και εκδόσεις συμβατές με υπολογιστές Macintosh, αλλά όχι με Linux).

Επίσης αναγκαζόμαστε να απορρίψουμε και τη Visual C++, παρά το γεγονός ότι μας παρέχει ένα αξιολογο σύνολο συστατικών σχεδίασης, κατάλληλο για υψηλού επιπέδου γραφικά περιβάλλοντα. Όπως και η Visual Basic, έτσι και η Visual C++ αποτελεί προϊόν της Microsoft και άρα είναι ασύμβατη με το Linux.

Επόμενη γλώσσα την οποία απορρίψαμε είναι η Delphi. Όπως είδαμε και πιο πριν αν μια εφαρμογή δημιουργηθεί σε Windows χρησιμοποιείται αποκλειστικά εκεί, ένα αν έχει δημιουργηθεί στο Linux (Kylux), εκτελείται μόνο στο Linux. Άρα θεωρείται για την περίπτωση μας ακατάλληλη, καθώς στόχος μας είναι ένα αρχείο εκτελέσιμο που να μπορεί να τρέχει και στα δύο προαναφερόμενα λειτουργικά.

Για αυτόν τον λόγο απορρίπτουμε και το Qt designer γιατί παρά το γεγονός ότι κώδικας που έχει γραφτεί σε αυτό το εργαλείο μπορεί να εκτελεστεί σε οποιοδήποτε λειτουργικό μεταγλωττιστεί, δεν είναι δυνατή η ύπαρξη ενός αρχείου που να μπορεί απευθείας να τρέχει και στα δυο λειτουργικά.

Μετά από όλα αυτά καταλήγουμε στη Java καθώς είναι η μόνη γλώσσα από αυτές που εξετάσαμε στην οποία να μπορεί να δημιουργηθεί κάποιο εκτελέσιμο που να μπορεί να χρησιμοποιηθεί και σε Linux και σε Windows εφόσον φυσικά υπάρχει εγκατεστημένη κάποια έκδοση της Java. Το μόνο πρόβλημα που προκύπτει είναι η επικοινωνία με τη C++, αλλά και αυτό το πρόβλημα μπορεί να λυθεί μέσω της χρήσης JNI και μιας βιβλιοθήκης dll. Βέβαια η τελευταία βιβλιοθήκη είναι διαφορετική για κάθε λειτουργικό αλλά η δημιουργία της είναι αρκετά εύκολη.

Παράρτημα Α: 4 Προειδοποιητικά μηνύματα

Στο εν λόγω παράρτημα παρατίθεται ένας κατάλογος με όλα τα προειδοποιητικά μηνύματα που πιθανά να εμφανιστούν κατά τη λειτουργία του προγράμματος:

1) **“You haven’t set all parameters.”** Εμφανίζεται σε περιπτώσεις στις οποίες ο χρήστης δεν έχει δώσει τιμές σε όλα τα πεδία κειμένου του παραθύρου διαλόγου που αντιπροσωπεύει για τη λειτουργία την οποία έχει επιλέξει.

2) **“Check the values of parameters please.”** Εμφανίζεται σε περιπτώσεις στις οποίες ο χρήστης έχει πληκτρολογήσει κάποια μη αποδεκτή τιμή, όπως για παράδειγμα μια τιμή δεκαδική σε ένα πεδίο που ζητά αριθμό επαναλήψεων της διαδικασίας. Χρησιμοποιείται σε όλα τα παράθυρα διαλόγου εκτός από αυτά που αναφέρονται στην κλιμάκωση της εικόνας του ψηφιακού κυκλώματος, και σε αυτά στα οποία ο χρήστης καλείται να δώσει μόνο μια τιμή. Το ποιες τιμές θεωρούνται αποδεκτές ποικίλλει από παράμετρο σε παράμετρο. Στο πρόγραμμά μας οι περισσότερες παράμετροι είναι θετικοί αριθμοί, συνήθως ακέραιοι, και σε μερικές περιπτώσεις δεκαδικοί. Όλες οι άλλες περιπτώσεις αριθμών, θεωρούνται λανθασμένες.

3) **“You haven’t set the zoom in factor.”** Εμφανίζεται όταν στο παράθυρο διαλόγου που αναφέρεται στη μεγέθυνση της εικόνας επιχειρήσουμε να πατήσουμε το σχετικό κουμπί χωρίς να έχουμε θέσει τον παράγοντα σμίκρυνσης.

3) **“The value of zoom in factor must be greater than 1”.** Εμφανίζεται όταν στο παράθυρο διαλόγου που αναφέρεται στη μεγέθυνση της εικόνας θέσουμε τιμή μικρότερη του 1.

4) **“Check the value of zoom in factor”.** Εμφανίζεται όταν στο παράθυρο που αναφέρεται στη μεγέθυνση της εικόνας ο χρήστης δώσει στον παράγοντα μεγέθυνσης μη αριθμητική τιμή.

5) **“You haven’t set the zoom out factor.”** Εμφανίζεται όταν στο παράθυρο διαλόγου που αναφέρεται στη σμίκρυνση της εικόνας ο χρήστης πατήσει το σχετικό κουμπί χωρίς να έχει θέσει τον παράγοντα σμίκρυνσης.

- 6) ***“The value of zoom out factor can’t be greater than 1”***. Εμφανίζεται όταν στο παράθυρο διαλόγου που αναφέρεται στη σμίκρυνση της εικόνας του κυκλώματος θέσουμε τιμή μεγαλύτερη από 1.
- 7) ***“The value of zoom out factor can’t be negative”***. Εμφανίζεται όταν στο παράθυρο διαλόγου που αναφέρεται στη σμίκρυνση της εικόνας του κυκλώματος θέσουμε αρνητική τιμή. Απεικονίζεται παρακάτω.
- 8) ***“The value of zoom out factor can’t be 0”***. Εμφανίζεται όταν στο παράθυρο που αναφέρεται στη σμίκρυνση της εικόνας θέσουμε μηδενική τιμή.
- 9) ***“Check the value of zoom out factor”***. Εμφανίζεται όταν στο παράθυρο διαλόγου που αναφέρεται στη σμίκρυνση της εικόνας ο χρήστης θέσει μη αριθμητική τιμή.
- 10) ***“You haven’t set the number of tests”***. Εμφανίζεται όταν στο παράθυρο διαλόγου που χρησιμοποιείται για τον υπολογισμό της δυναμικής ισχύος ή στη δεύτερη καρτέλα του παραθύρου που χρησιμοποιείται για τον υπολογισμό της στατικής ισχύος ο χρήστης δεν έχει θέσει τον αριθμό των επαναλήψεων.
- 11) ***“Check the value of number of tests please”***. Εμφανίζεται όταν στο παράθυρο διαλόγου που χρησιμοποιείται για τον υπολογισμό της δυναμικής ισχύος ή στη δεύτερη καρτέλα του παραθύρου που χρησιμοποιείται για τον υπολογισμό της στατικής ισχύος ο χρήστης έχει θέσει μη αποδεκτό αριθμό επαναλήψεων. Στη συγκεκριμένη περίπτωση η πληκτρολογούμενη τιμή θα χαρακτηριστεί μη αποδεκτή, αν είναι δεκαδική, αρνητική, ή μη αριθμητική.
- 12) ***“You haven’t set the depth”***. Εμφανίζεται στη δεύτερη καρτέλα του παραθύρου που αντιπροσωπεύει τη λειτουργία “Multi Level Function”, σε περίπτωση που ο χρήστης δε δώσει τιμή για το βάθος του κυκλώματος στο οποίο πρέπει να γίνει η λειτουργία.
- 13) ***“Please check the value of depth”***. Εμφανίζεται στη δεύτερη καρτέλα του παραθύρου που αντιπροσωπεύει τη λειτουργία “Multi Level Function”, σε περίπτωση που ο χρήστης δώσει μη αποδεκτή τιμή για το βάθος του κυκλώματος στο οποίο πρέπει να γίνει η ανάλυση. Σε αυτή την περίπτωση η πληκτρολογούμενη τιμή θα θεωρηθεί μη αποδεκτή αν είναι αρνητική, δεκαδική, μη αριθμητική ή είναι μεγαλύτερη από το μέγιστο βάθος του κυκλώματος.

Παράρτημα Β: Μηνύματα που εμφανίζονται στο παράθυρο “Result”

Παρακάτω παραθέτουμε σε έναν κατάλογο τα κυριότερα μηνύματα που εμφανίζονται στο παράθυρο “Results”, μετά την εκτέλεση κάποιας λειτουργίας. Βέβαια είναι δυνατή η εμφάνιση και άλλων μηνυμάτων τα οποία αναφέρονται κυρίως σε λάθη που πιθανά να προκύψουν κατά την ανάγνωση του αρχείου spice, ή κατά την ανάγνωση δεδομένων που προέρχονται από το τμήμα του προγράμματος που κάνει την προσομοίωση, αλλά αναφέρουμε τα πιο σημαντικά.

1) Complete state 1(reading file C:/D:/diplwmatikh:/circuits:/s-circuits:/s27.ckt)

Complete state 2(finding levels): Το μήνυμα αυτό εμφανίζεται μετά την επιλογή ενός αρχείου και εφόσον έχει ολοκληρωθεί επιτυχώς το ανάλογο τμήμα της προσομοίωσης, ανεξάρτητα από το αν το κύκλωμα έχει σχηματιστεί. Πιο συγκεκριμένα μας ειδοποιεί ότι το κύκλωμα διαβάστηκε επιτυχώς και επίσης έχει γίνει και η επιπεδοποίησή του.

2) Can't open file C:/D:/diplwmatikh:/circuits:/s-circuits:/s27.ckt. Όπως είναι εύκολο να καταλάβει κανείς εμφανίζεται σε περίπτωση που το πρόγραμμα δε μπορεί να ανοίξει το αρχείο.

3) ERROR: didn't succeed to read line #34 from the file.

I read VG0 1 0 pulse (0 5 15n) Εμφανίζεται σε περίπτωση που το πρόγραμμα δε μπορεί να διαβάσει το αρχείο και κατά συνέπεια να συνεχίσει την προσομοίωση. Μάλιστα αναφέρεται σε ποια γραμμή του αρχείου υπάρχει το πρόβλημα, και εμφανίζει τη γραμμή.

4) MEMORY ERROR: Not enough memory. Εμφανίζεται σε περιπτώσεις που δεν επαρκεί η διαθέσιμη μνήμη για την εκτέλεση της συγκεκριμένης λειτουργίας.

5) MEMORY ERROR: Can't create new decode. Εμφανίζεται κατά το σχηματισμό των απαραίτητων decodes, σε περιπτώσεις που λόγω μη επαρκούς διαθέσιμης μνήμης είναι αδύνατη η κατασκευή κάποιου από αυτά.

6) MEMORY ERROR: At main. Can't create new signal: Εμφανίζεται κατά τη διαδικασία δημιουργίας των απαραίτητων signals, σε περιπτώσεις όπου λόγω μη αρκετής διαθέσιμης μνήμης είναι αδύνατη η δημιουργία κάποιου από αυτά.

7) **ERROR: Signal 345 already exists.** Εμφανίζεται κατά τη διαδικασία δημιουργίας των απαραίτητων signals, σε περιπτώσεις επαναδημιουργίας ενός σήματος.

8) **ERROR: Miss some inputs!** Εμφανίζεται στην περίπτωση που η προσομοίωση δε μπορεί να συνεχιστεί καθώς δεν έχουν δημιουργηθεί όλα τα απαραίτητα σήματα με αποτέλεσμα κάποιες πύλες να έχουν μικρότερο αριθμό εισόδων από τον απαραίτητο.

9) **Too many inputs at sub – circuit gate2 (line 534)** Εμφανίζεται στην

9) **ERROR: Din't find the drain for the subckt:** Εμφανίζεται κατά τη διαδικασία σχηματισμού των transistors, σε περιπτώσεις που δε μπορεί να βρει την υποδοχή του transistor προκειμένου να καταφέρει να το σχηματίσει.

10) **Failed to make tranzistors,**

11) **Didn't manage to make transistor:**

Εμφανίζονται κατά τη διαδικασία σχηματισμού των transistors σε περιπτώσεις που η κατασκευή κάποιου από αυτά είναι αδύνατη.

12) **ERROR: Didn't find end at the end of the file.** Εμφανίζεται σε περιπτώσεις όπου το format του αρχείου δεν είναι σωστό με αποτέλεσμα το πρόγραμμα να μη μπορεί να σταματήσει την ανάγνωση.

13) **Complete state 3 (no delay output) for times 3 times. Complete printing results for the simulation at file C:/ D:/ diphwmatikh:/ circuits:/ s-circuits:/ s27_simulation.txt.** Εμφανίζεται μετά την επιτυχή εκτέλεση της μέτρησης της στατικής ισχύος και μας ειδοποιεί ότι τα αποτελέσματα της μέτρησης έχουν αποθηκευτεί στο συγκεκριμένο αρχείο στο συγκεκριμένο φάκελο στο δίσκο.

14) **Can't open file to write results for Static Energy test.** Εμφανίζεται όταν δημιουργηθεί κάποιο πρόβλημα με το προαναφερόμενο αρχείο και δεν είναι δυνατή η αποθήκευση των αποτελεσμάτων.

15) **Internal power estimation complete in 156ms:** Εμφανίζεται μετά την επιτυχή εκτέλεση της λειτουργίας για τη μέτρηση της δυναμικής ενέργειας και μας λέει σε πόσο χρόνο εκτελέστηκε η συγκεκριμένη λειτουργία.

16) **Complete state 3 (ERROR CHECK) for 4 times. Complete printing results for Single Event Upsets Test at file C:/ D:/ diphwmatikh:/ circuits:/ s-circuits:/ s27_error_check.txt.** Εμφανίζεται για να μας ειδοποιήσει ότι το σχετικό τμήμα της προσομοίωσης πραγματοποιήθηκε με επιτυχία και ότι τα εν λόγω αποτελέσματα είναι αποθηκευμένα στο εν λόγω αρχείο.

17) Can't open file to write results for Single Event Upsets test. Εμφανίζεται και αυτό όταν δεν είναι δυνατό το άνοιγμα του αρχείου για την αποθήκευση των αποτελεσμάτων της προσομοίωσης.

18) Start Multi Level Function calculation. Εμφανίζεται με την εκκίνηση της λειτουργίας Single Event Upsets.

19) ERROR: At find_multi_level_function. Εμφανίζεται στην περίπτωση που συμβεί κάποιο λάθος κατά τη διάρκεια της προσομοίωσης και πλέον δεν είναι δυνατή η εξέλιξη της λειτουργίας.

20) Can't open file to write results for Multi Level Function: Εμφανίζεται σε περίπτωση που δεν είναι δυνατή η δημιουργία του αρχείου για την αποθήκευση των αποτελεσμάτων της διαδικασίας Multi Level Function για πολλούς κόμβους.

21) Complete printing results for Multi Level Function at file C:/S208_multi_level_func: Εμφανίζεται σε περίπτωση που η διαδικασία Multi Level Function για πολλούς κόμβους εκτελεστεί επιτυχώς και μας λέει ότι τα σχετικά αποτελέσματα βρίσκονται αποθηκευμένα στο συγκεκριμένο αρχείο στο δίσκο.

22) Start Multi Level Function for node 66: Εμφανίζεται σε περίπτωση που ο χρήστης θέσει σωστά όλες τις παραμέτρους στο σχετικό παράθυρο διαλόγου και μας ειδοποιεί ότι έχει ξεκινήσει η διαδικασία Multi Level Function για τον κόμβο 66.

23) There is no level 34: Εμφανίζεται σε περίπτωση που ο χρήστης θέσει αριθμό επιπέδου το οποίο δεν υπάρχει στο κύκλωμα.

Εμφανίζεται σε περιπτώσεις που δεν υπάρχει αποθηκευμένη κάποια συνάρτηση για τον εν λόγω κόμβο

24) Node 66 has no function: Εμφανίζεται σε περιπτώσεις που το πρόγραμμα δε μπορεί να βρει κάποια συνάρτηση για τον εν λόγω κόμβο.

25) Node 66 has function NOT(26)(1) for 3 levels: Εμφανίζει το αποτέλεσμα της λειτουργίας Multi Level Function για έναν κόμβο.

Παράρτημα Γ: Θέματα Βοήθειας

Το εν λόγω παράρτημα παραθέτει τα θέματα βοήθειας για τη χρήση του προγράμματος αυτού. Πιο αναλυτικά περιγράφει τα κουμπιά του παραθύρου Help που παρουσιάστηκε στο κεφάλαιο 4 καθώς και το κείμενο που εμφανίζεται μόλις ο χρήστης πατήσει κάποιο από αυτά.

1. “About Leakage Power”: Με το πάτημα του κουμπιού αυτού εμφανίζεται ένα κείμενο στο οποίο περιγράφονται τα βήματα για τη μέτρηση της κατανάλωσης της στατικής ισχύος του κυκλώματος. Το κείμενο αυτό είναι το παρακάτω:

- 1. Choose the menu Estimations and then choose the menu item 'Leackage Power Estimation. Alternatively you can click on the L.P.M. button or press the keys Ctrl + L.*
- 2. Set the following parameters. The depth must be an integer and the other two parameters must be double.*
- 3. After that, click on the following button.*
- 4. If you want to use the second method, choose the tab 'Second method. Enter in the text field the number of iterations will the program run*

2. “About Dynamic Power”: Με το πάτημα του κουμπιού αυτού εμφανίζεται ένα κείμενο στο οποίο περιγράφονται τα βήματα για τη μέτρηση της κατανάλωσης της δυναμικής ισχύος του κυκλώματος. Το κείμενο αυτό είναι το παρακάτω:

- 1. Choose the menu Estimations and then choose the menu item 'Dynamic Power Estimation. Alternatively you can click on the D.P.M. button or press Ctrl + P.*
- 2. Enter in the text field the number of iterations will the program run.*

3. “About Multi Level Function”: Με το πάτημα του κουμπιού αυτού εμφανίζεται ένα κείμενο στο οποίο περιγράφει τα βήματα για την υλοποίηση της λειτουργίας Multi Level Function. Το κείμενο αυτό είναι το παρακάτω:

- 1. Choose the menu Estimations and then choose the menu item Multi Level Function. Alternatively you can click on the M.L.F. or press Ctrl + M.*

2. *If you want to do the analysis of one node you must choose the tab with the title 'For one node', else you have to choose the tab with the title 'For any node'.*
3. a. *If you choose the first tab, you have to enter an integer at the level text field and then to choose one node from the following list.*
 - b. *Next you have to decide if you want to do this operation for one depth (choose the first radio button), or for any depth (choose the second radio button).*
 - c. *At the first case enter an integer in the following text field and then click the Multi Level Function Button. Else, click this button.*
4. a. *If you choose the second tab you have to decide if you want to do the calculation for one depth or for many depths.*
 - b. *At the first case enter an integer in the following text field and click the button. Else click the button.*

4. **“About Single Event Upsets”**: Με το πάτημα του κουμπιού αυτού εμφανίζεται ένα κείμενο στο οποίο περιγράφονται τα βήματα για την υλοποίηση της λειτουργίας Single Event Upsets. Το κείμενο αυτό είναι το παρακάτω:

1. *Choose the menu Estimations and then choose the menu item Single Event Upsets. Alternatively you can do a click on the S.E.U. or you can press Ctrl + S”.*

5. **“About Zoom In”**: Με το πάτημα του κουμπιού αυτού εμφανίζεται ένα κείμενο στο οποίο περιγράφονται τα βήματα που πρέπει να ακολουθήσει ο χρήστης για την μεγέθυνση μιας εικόνας. Το κείμενο αυτό είναι το παρακάτω:

1. *Choose the menu scaling and then choose the menu item Zoom In.*
2. *In the window which appears set the factor of zoom in. The factor must be greater than 1.*

6. **“About Zoom Out”**: Με το πάτημα του κουμπιού αυτού εμφανίζεται ένα κείμενο στο οποίο περιγράφονται τα βήματα που πρέπει να ακολουθήσει ο χρήστης για τη σμίκρυνση μιας εικόνας. Το κείμενο αυτό είναι το παρακάτω:

1. *Choose the menu scaling and then choose the menu item Zoom out.*
2. *In the window which appears set the factor of zoom out. The factor must be positive and can't be greater than one.*

Αναφορές

- 1) Υλοποίηση ενός προσομοιωτή ψηφιακών κυκλωμάτων για την μελέτη της κατανάλωσης ισχύος (Δημήτρης Μπουντάς – διπλωματική εργασία)
- 2) <http://java.sun.com/docs/books/tutorial/uiswing/>
- 3) <http://www.festra.com/>
- 4) www.blibbleblobble.co.uk/Programming/visualc/index
- 5) <http://www.techforums.net/showthread.php?s=&threadid=22943>
- 6) <http://bdn.borland.com/article/0,1410,26970,00.html>
- 7) <http://channel9.msdn.com/ShowPost.aspx?PostID=31267>
- 8) <http://www.trolltech.com/>
- 9) <http://cuinl.tripod.com/tutorials.htm>
- 10) <http://www.geocities.com/SiliconValley/Bay/5707/vbasic.html>
- 11) <http://mindprod.com/jgloss/swing.html>
- 12) <http://www.artima.com/forums/flat.jsp?forum=17&thread=21306>
- 13) http://leepoint.net/notes-java/10background/10applications_and_applets/05intro/05swing-vs-awt.html
- 14) http://www.sunncity.com/Tutorial_Java/partTwo/gui.html
- 15) <http://lists.trolltech.com/qt-interest/2004-07/thread00977-0.html>
- 16) <http://www.onjava.com/pub/a/onjava/2004/02/18/desktop.html>
- 17) http://en.wikipedia.org/wiki/Microsoft_Foundation_Classes
- 18) <http://phil.freehackers.org/kde/qt-vs-mfc.html>
- 19) http://phil.freehackers.org/kde/qt-vs-mfc_comments.html
- 20) <http://www.insomnia.gr/vb3/showthread.php?t=74692>
- 21) <http://www.cs.utexas.edu/users/almstrum/cs370/elvisino/rules.html>
- 22) <http://www.sena.gr/~dds/c3/vb/>



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ



004000074809