



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΣΥΓΚΡΙΤΙΚΗ ΑΝΑΛΥΣΗ ΛΟΓΙΚΩΝ
ΠΡΟΣΟΜΟΙΩΤΩΝ

ΔΕΛΕΝΙΚΑΣ ΧΡΗΣΤΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΔΑΔΑΛΙΑΡΗΣ ΑΝΤΩΝΙΟΣ

ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ

Λαμία Μάρτιος έτος 2022



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΣΥΓΚΡΙΤΙΚΗ ΑΝΑΛΥΣΗ ΛΟΓΙΚΩΝ
ΠΡΟΣΟΜΟΙΩΤΩΝ

ΔΕΛΕΝΙΚΑΣ ΧΡΗΣΤΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΔΑΔΑΛΙΑΡΗΣ ΑΝΤΩΝΙΟΣ

ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ

Λαμία Μάρτιος έτος 2022



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

COMPARATIVE ANALYSIS OF LOGICAL
SIMULATORS

DELENIKAS CHRISTOS KONSTANTINOS

FINAL THESIS

ADVISOR

DADALIARIS ANTONIOS

ASSISTANT PROFESSOR

Lamia March year 2022

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.

2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσιάσή τους ως δική μου εργασία.

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία:/...../20.....

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία με τίτλο «Συγκριτική ανάλυση λογικών προσομοιωτών» πραγματοποιεί την εκτενή ανάλυση ενός συνόλου λειτουργιών από επιλεγμένους δωρεάν λογικούς προσομοιωτές ψηφιακής σχεδίασης με σκοπό την ανάλυση τους για την επιλογή και αναφορά των ικανότερων από αυτούς. Αρχικά, διεξάγεται αναφορά στο γνωσιακό υπόβαθρο της ψηφιακής σχεδίασης που θα χρειαστεί για τη χρήση και κατανόηση των λειτουργιών και δυνατοτήτων των λογικών προσομοιωτών. Στη συνέχεια, υλοποιείται αναφορά σε κάθε ένα λογικό προσομοιωτή με χρήση εικόνων από το περιβάλλον χρήσης τους σε συνδυασμό με ανάλυση των επιλεγμένων λειτουργιών τους. Επιπλέον, παρουσιάζονται και αναλύονται συγκεντρωτικά γραφήματα όπου απεικονίζονται οι λειτουργίες των προσομοιωτών με στόχο τη καλύτερη κατανόηση. Τέλος, πραγματοποιείται αναφορά των ικανότερων προσομοιωτών βάση των λειτουργιών και δυνατοτήτων τους.

ABSTRACT

The present thesis entitled "Comparative analysis of logical simulators" carries out the extensive analysis of a set of functions from selected free logical digital design simulators in order to analyze them for the selection and reporting of the most capable of them. Reference is made to each logical simulator using images from their user interface in combination with analysis of their selected functions. In addition, aggregated graphs are presented and analyzed where the functions of the simulators are depicted in order to better understand. Finally, a report of the most capable simulators based on their functions and capabilities is made.

ΕΥΧΑΡΙΣΤΙΕΣ

Ολοκληρώνοντας τις σπουδές μου, με την συγγραφή της παρούσας πτυχιακής εργασίας θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Αντώνιο Δαδαλιάρη για το χρόνο που αφιέρωσε δίνοντας μου καθοριστικές συμβουλές και οδηγίες προκυμμένου να ολοκληρώσω τη πτυχιακή μου εργασία. Στο ίδιο πλαίσιο ευγνωμοσύνης, θα ήθελα να ευχαριστήσω όλους τους καθηγητές του Τμήματος Πληροφορικής και τηλεπικοινωνιών της σχολής θετικών επιστημών του Πανεπιστημίου Θεσσαλίας για τη συμβολή τους στην επιστημονική και τεχνολογική μου συγκρότηση στα χρόνια της φοίτησής μου στο Τμήμα.

Περιεχόμενα

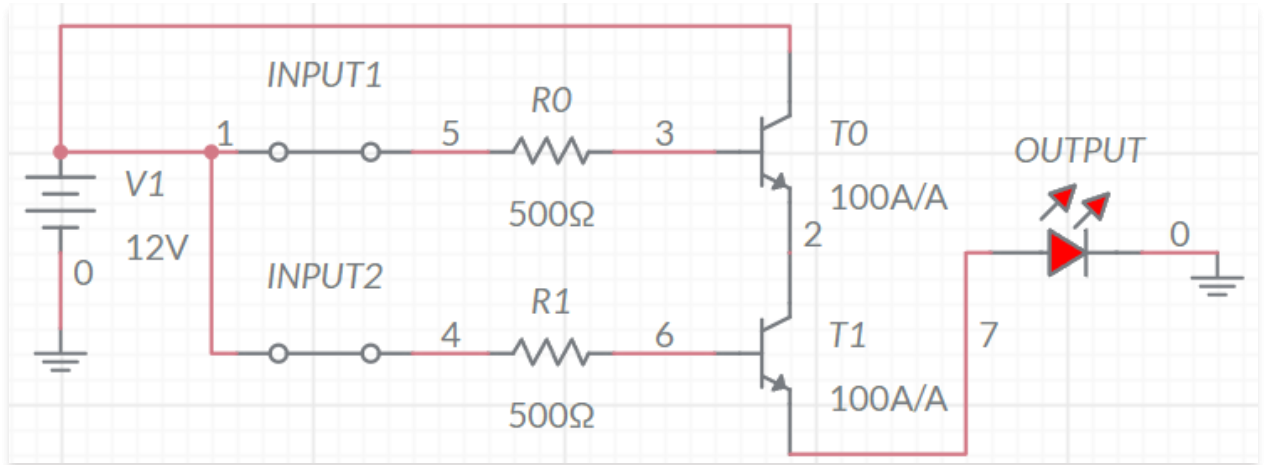
ΠΕΡΙΛΗΨΗ	1
ABSTRACT.....	3
ΕΥΧΑΡΙΣΤΙΕΣ.....	5
ΠΕΡΙΕΧΟΜΕΝΑ.....	7
1 ΕΙΣΑΓΩΓΗ.....	11
1.1 ΒΑΣΙΚΑ ΕΙΔΗ ΛΟΓΙΚΩΝ ΠΥΛΩΝ	11
1.2 ΕΛΑΧΙΣΤΟΠΟΙΗΣΗ ΛΟΓΙΚΩΝ ΣΥΝΑΡΤΗΣΕΩΝ ΚΑΙ ΆΛΓΕΒΡΑ BOOLE.....	13
1.2.1 Αξιώματα της Άλγεβρας Boole	13
1.2.2 Θεωρήματα της Άλγεβρας Boole	14
1.2.3 Ελαχιστοβάθμιοι όροι (Ελαχιστόρος).....	14
1.2.4 Μεγιστοβάθμιοι όροι (Μεγιστόρος).....	15
1.2.5 Προτεραιότητες πράξεων	15
1.2.6 Κανονικοποίηση.....	15
1.2.7 Ελαχιστοποίηση λογικών συναρτήσεων.....	16
1.2.8 Διαγράμματα Venn	16
1.3 ΧΑΡΤΕΣ KARNAUGH.....	16
1.3.1 Χάρτες Karnaugh 3 εισόδων.....	16
1.3.2 Ελαχιστοποίηση K-map.....	17
1.3.3 Χάρτες Karnaugh 4 εισόδων.....	18
1.3.4 Αδιάφορες τιμές σε χάρτες Karnaugh	19
1.4 ΚΩΔΙΚΑΣ BCD (BINARY-CODED-DECIMAL).....	20
1.5 ΑΠΟΚΩΔΙΚΟΠΟΙΗΤΗΣ ΑΠΕΙΚΟΝΙΣΗΣ ΕΠΤΑ ΤΜΗΜΑΤΩΝ (SEVEN SEGMENT DECODER) 20	
1.6 ΠΟΛΥΠΛΕΚΤΗΣ 2 ΣΕ 1	21
1.7 ΠΟΛΥΠΛΕΚΤΗΣ 4 ΣΕ 1	21
1.8 ΑΠΟΠΛΕΚΤΗΣ 1-4	22
1.9 ΧΡΟΝΙΣΜΟΣ ΨΗΦΙΑΚΟΥ ΚΥΚΛΩΜΑΤΟΣ	23
1.10 ΣΥΝΤΟΜΗ ΚΑΙ ΚΡΙΣΙΜΗ ΔΙΑΔΡΟΜΗ	24
1.11 ΑΚΟΛΟΥΘΙΑΚΑ ΚΥΚΛΩΜΑΤΑ.....	24
1.11.1 SR Latch.....	25
1.11.2 D Latch.....	26
1.11.3 Flip - Flops	28
1.11.4 Τρόποι ενεργοποίησης (triggering) Flip – Flop.....	28
1.11.5 D Flip - Flops.....	28
1.11.6 SR Flip – Flops	29
1.11.7 JK Flip – Flops	29
1.11.8 T Flip – Flop.....	30
1.12 ΣΥΓΧΡΟΝΑ ΑΚΟΛΟΥΘΙΑΚΑ ΚΥΚΛΩΜΑΤΑ	31
1.13 FINITE STATE MACHINES FSM	31
1.14 ΔΙΑΓΡΑΜΜΑ ΜΕΤΑΒΟΛΗΣ ΚΑΤΑΣΤΑΣΗΣ.....	32
1.15 ΣΧΕΔΙΑΣΗ FSM.....	32
1.16 ΔΗΜΙΟΥΡΓΙΑ 2-ΨΗΦΙΟΥ ΣΥΓΧΡΟΝΟΥ ΔΥΑΔΙΚΟΥ ΜΕΤΡΗΤΗ.....	36

2	ΠΡΟΣΟΜΟΙΩΤΕΣ ΨΗΦΙΑΚΩΝ ΚΥΚΛΩΜΑΤΩΝ.....	38
2.1	DIGITAL LOGIC SIM.....	38
2.1.1	<i>Λειτουργικά χαρακτηριστικά</i>	38
2.2	MULTIMEDIA LOGIC.....	39
2.2.1	<i>Λειτουργικά χαρακτηριστικά</i>	39
2.3	CIRCUITLOGIX	39
2.3.1	<i>Λειτουργικά χαρακτηριστικά</i>	40
2.4	LOGIC GATE SIMULATOR	40
2.4.1	<i>Λειτουργικά χαρακτηριστικά</i>	41
2.5	CIRCUITMOD	41
2.5.1	<i>Λειτουργικά χαρακτηριστικά</i>	42
2.6	SIMULATOR.IO.....	42
2.6.1	<i>Λειτουργικά χαρακτηριστικά</i>	43
2.7	LOGIC CIRCUIT.....	43
2.7.1	<i>Λειτουργικά χαρακτηριστικά</i>	44
2.8	CEDAR LOGIC	45
2.8.1	<i>Λειτουργικά χαρακτηριστικά</i>	45
2.9	MULTISIM LIVE.....	46
2.9.1	<i>Λειτουργικά χαρακτηριστικά</i>	46
2.10	CIRCUITVERSE	47
2.10.1	<i>Λειτουργικά χαρακτηριστικά</i>	48
2.11	LOGIC FRIDAY.....	49
2.11.1	<i>Λειτουργικά χαρακτηριστικά</i>	49
2.12	LOGIC.LY	50
2.12.1	<i>Λειτουργικά χαρακτηριστικά</i>	50
2.13	BOOLR	51
2.13.1	<i>Λειτουργικά χαρακτηριστικά</i>	51
2.14	DEEDS	51
2.14.1	<i>Λειτουργικά χαρακτηριστικά</i>	52
2.15	LOGISIM	52
2.15.1	<i>Λειτουργικά χαρακτηριστικά</i>	53
2.16	HADES.....	53
2.16.1	<i>Λειτουργικά χαρακτηριστικά</i>	54
2.17	DIGITAL LOGIC DESIGN	55
2.17.1	<i>Λειτουργικά χαρακτηριστικά</i>	56
2.18	DIGITAL	56
2.18.1	<i>Λειτουργικά χαρακτηριστικά</i>	57
3	ΣΥΝΟΛΙΚΗ ΑΝΑΛΥΣΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΠΡΟΣΟΜΟΙΩΤΩΝ.....	60
3.1	ΜΕΓΕΘΟΣ ΚΑΙ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ	60
3.2	ΑΡΙΘΜΟΣ ΕΙΣΟΔΩΝ, ΔΙΑΔΡΟΜΗ ΚΑΛΩΔΙΩΝ ΚΑΙ ΑΠΕΙΚΟΝΙΣΗ ΤΙΜΗΣ ΣΤΑ ΚΑΛΩΔΙΑ	61
3.3	ΕΞΑΡΤΗΜΑΤΑ FLIP-FLOP, ΕΞΑΡΤΗΜΑ ΡΟΛΟΓΙΟΥ ΚΑΙ ΔΙΑΓΡΑΜΜΑ ΧΡΟΝΙΣΜΟΥ	63
3.4	ΠΙΝΑΚΕΣ ΑΛΗΘΕΙΑΣ, ΛΟΓΙΚΕΣ ΕΚΦΡΑΣΕΙΣ ΚΑΙ ΑΠΛΟΠΟΙΗΣΗ ΚΥΚΛΩΜΑΤΟΣ	64
3.5	ΔΗΜΙΟΥΡΓΙΑ ΚΩΔΙΚΑ ΑΠΟ ΓΛΩΣΣΑΣ ΠΕΡΙΓΡΑΦΗΣ ΥΛΙΚΟΥ ΚΑΙ ΥΠΟΣΤΗΡΙΞΗ SCRIPTING 66	
3.6	ΕΙΔΙΚΑ ΕΞΑΡΤΗΜΑΤΑ, ΕΛΕΓΧΟΣ ΣΦΑΛΜΑΤΩΝ ΚΑΙ ΕΚΤΥΠΩΣΗ ΚΥΚΛΩΜΑΤΟΣ	67

3.7	ΣΥΝΟΛΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΛΕΙΤΟΥΡΓΙΩΝ	69
4	ΣΥΜΠΕΡΑΣΜΑΤΑ	70
4.1	ΣΥΣΤΑΣΗ WINDOWS/MACOS/LINUX BASED ΠΡΟΣΟΜΟΙΩΤΗ	70
4.2	ΣΥΣΤΑΣΗ ONLINE ΠΡΟΣΟΜΟΙΩΤΗ.....	70
4.3	ΑΝΑΦΟΡΑ ΛΙΓΟΤΕΡΟ ΑΠΟΤΕΛΕΣΜΑΤΙΚΩΝ ΠΡΟΣΟΜΟΙΩΤΩΝ.....	70
5	ΑΝΑΦΟΡΕΣ ΚΑΙ ΒΙΒΛΙΟΓΡΑΦΙΑ	71

1 Εισαγωγή

Ένα από τα βασικά στοιχεία της ηλεκτρονικής είναι οι λογικές πύλες. Οι λογικές πύλες είναι ψηφιακές διατάξεις που δέχονται μία ή περισσότερες εισόδους δεδομένων και μετά την υλοποίηση κάποιας λογικής συνάρτησης δίνουν στην έξοδο τους το αποτέλεσμα της πράξης αυτής. Δομικά στοιχεία των λογικών πυλών είναι τρανζίστορ καθώς και οι διόδοι.



1 Αναπαράσταση λογικής πύλης AND με αντιστάτες(R0,R1) και τρανζίστορ(T0,T1).

1.1 Βασικά είδη λογικών πυλών

Πύλη AND (ΚΑΙ)

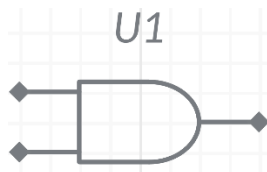


Figure 3 Λογική πύλη And.

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

2 A,B είσοδοι πύλης X έξοδος.

Η λογική πύλη AND περιγράφεται από την συνάρτηση $X = A * B$ και έχει δύο ή περισσότερες εισόδους.

Πύλη OR (Η)

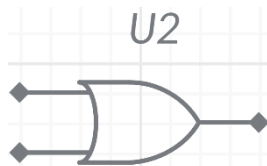
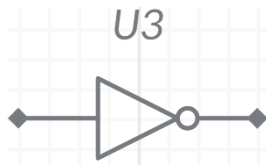


Figure 4 Λογική πύλη OR.

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Η λογική πύλη OR περιγράφεται από την συνάρτηση $X = A + B$ και έχει δύο ή περισσότερες εισόδους.

Πύλη NOT (ΟΧΙ)

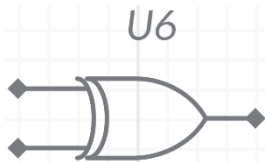


A	X
0	1
1	0

Figure 5 Λογική πύλη NOT.

Η λογική πύλη NOT περιγράφεται από την συνάρτηση $X = A'$ και έχει μόνο μία είσοδο.

Πύλη XOR (Exclusive OR)



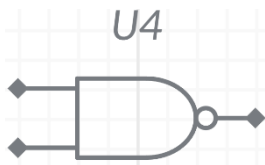
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Figure 6 Λογική πύλη XOR.

Η λογική πύλη XOR περιγράφεται από την συνάρτηση $X = A \oplus B$ και έχει δύο ή περισσότερες εισόδους.

Η πύλη NOT γίνεται να συνδυαστεί με τις πρώτες τρεις πύλες και να δημιουργήσει τρεις νέες λογικές πύλες με τα ονόματα NAND, NOR, XNOR.

Πύλη NAND (OXI-KAI)

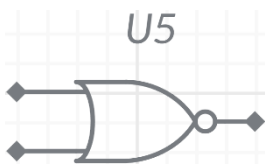


A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

Figure 7 Λογική πύλη NAND.

Η λογική πύλη NAND περιγράφεται από την συνάρτηση $X = \overline{A * B}$ και έχει δύο ή περισσότερες εισόδους.

Πύλη NOR (ΟΥΤΕ)

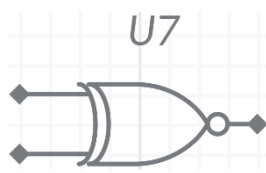


A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

Figure 8 Λογική πύλη NOR.

Η λογική πύλη NOR περιγράφεται από την συνάρτηση $X = \overline{A + B}$ και έχει δύο ή περισσότερες εισόδους.

Πύλη XNOR



A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

Figure 9 Λογική πύλη XNOR.

Η λογική πύλη XNOR περιγράφεται από την συνάρτηση $X = (A * B) + \overline{(A + B)}$ και έχει δύο ή περισσότερες εισόδους.

1.2 Ελαχιστοποίηση λογικών συναρτήσεων και Άλγεβρα Boole

Η άλγεβρα Boole η οποία πείρε το όνομα του Άγγλου μαθηματικού George Boole, που ήταν ο πρώτος που ασχολήθηκε με αυτή, ασχολείται με «δυαδικές» ή «λογικές» μεταβλητές. Η άλγεβρα Boole βρίσκει πολλές εφαρμογές στην σχεδίαση λογικών κυκλωμάτων.

Οι συναρτήσεις στην άλγεβρα Boole παριστάνονται με τους συμβολισμούς V1, V2, ή σαν V, \bar{V} . Οι τιμές των λογικών μεταβλητών μπορούν να λάβουν δύο τιμές «0» ή «1» και οι μεταβλητές που χρησιμοποιούνται γράφονται με κεφαλαία λατινικά γράμματα.

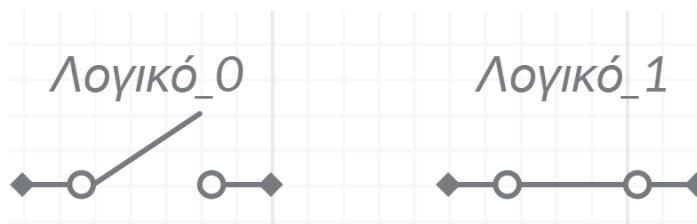


Figure 10 Αναπαράσταση λογικού 0 και 1 με χρήση διακοπών.

Η χρήση παύλας πάνω από μεταβλητή δηλώνει το συμπλήρωμα της μεταβλητής αυτής ή αλλιώς την άρνηση της π.χ. $\Delta = 0$, $\bar{\Delta} = 1$.

Στην άλγεβρα Boole γίνεται χρήση συναρτήσεων που αποτελούνται από λογικές μεταβλητές και ονομάζονται Λογικές συναρτήσεις. Η αναπαράσταση μίας λογικής συνάρτησης γίνεται με τη χρήση του γράμματος «T» ή με τα γράμματα f, g. Εάν πρέπει να χρησιμοποιούμε σύνολα η ένωση «U» εκφράζει πρόσθεση και η τομή «∩» γινόμενο. Ενώ εάν πρέπει να γίνει αναπαράσταση μεταβλητών με διακόπτες το λογικό άθροισμα παριστάνεται με διακόπτες συνδεδεμένους παράλληλα και το λογικό γινόμενο με διακόπτες συνδεδεμένους σε σειρά.

1.2.1 Αξιώματα της Άλγεβρας Boole

Αξίωμα 1: Αντιμεταθετικότητα ως προς +, * : $A + B = B + A$, $A * B = B * A$

Αξίωμα 2: Επιμεριστικοί νόμοι :

$$\text{Πράξη } + \text{ ως προς } * : A + (B * C) = (A + B) * (A + C)$$

$$\text{Πράξη } * \text{ ως προς } + : A * (B + C) = (A * B) + (A * C)$$

Αξίωμα 3: Ουδέτερο στοιχείο

$$\text{Το } 0 \text{ είναι το ουδέτερο στοιχείο της } + : A + 0 = 0 + A = A \text{ β.}$$

Το 1 είναι το ουδέτερο στοιχείο της : $A \cdot 1 = 1 \cdot A = A$

Αξίωμα 4: Συμπληρώματα ως προς την πράξη +: $A + A' = A' + A = 1$, ως προς την πράξη \cdot : $A \cdot A' = A' \cdot A = 0$

1.2.2 Θεωρήματα της Άλγεβρας Boole

Θεώρημα 1: $A + 1 = 1$

Θεώρημα 2: $A + A = A$

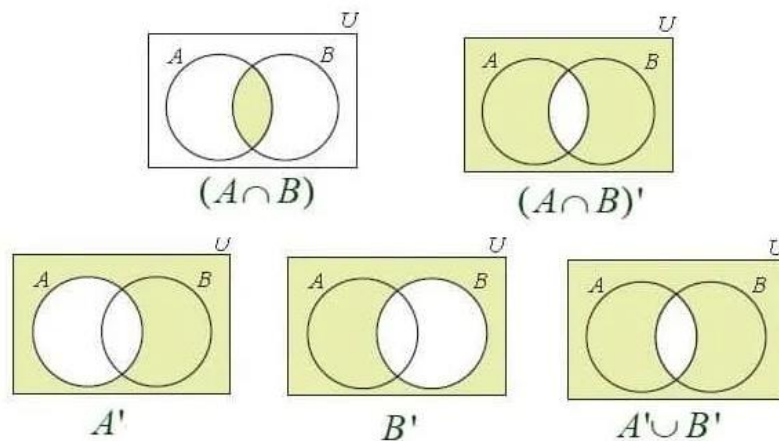
Θεώρημα 3: $A'' = A$

Θεώρημα 4: $A + (A \cdot B) = A$

Θεώρημα 5: $A + (A' \cdot B) = A + B$

Θεώρημα 6: $(A \cdot B) \cdot C = A \cdot (B \cdot C)$

Θεώρημα 7 (De Morgan): $(A + B) + A' \cdot B' = [(A + B) + A'] \cdot [(A + B) + B']$



Αναπαράσταση θεωρήματος De Morgan με διαγράμματα Venn.

Κάθε ελαχιστοποιημένη τελική εξίσωση είναι ισοδύναμη της αρχικής της. Η μείωση του πλήθους των πυλών έχει ως αποτέλεσμα το μικρότερο μέγεθος και κόστος, την χαμηλότερη κατανάλωση ισχύος και την αύξηση της ταχύτητας.

1.2.3 Ελαχιστοβάθμιοι όροι (Ελαχιστόρος)

Ονομάζεται το γινόμενο που περιέχει όλες τις μεταβλητές εισόδου.

π.χ. Εάν έχουμε 2 μεταβλητές $f(A,B)$ θα υπάρχουν $2^2 = 4$ ελαχιστόροι: $\bar{A} \bar{B}$, $\bar{A} B$, $A \bar{B}$ και $A B$.

1.2.4 Μεγιστοβάθμιοι όροι (Μεγιστόρος)

Ονομάζεται το άθροισμα των όρων που η κάθε μεταβλητή είσοδος εμφανίζεται μοναδικά σε κανονική ή συμπληρωματική μορφή.

π.χ. Εάν έχουμε 3 μεταβλητές $f(A,B,C)$ θα υπάρχουν $2^3 = 8$ μεγιστόροι: $\bar{A} + \bar{B} + \bar{C}$, $\bar{A} + \bar{B} + C$,

$\bar{A} + B + C$, $A + B + C$, $A + \bar{B} + C$, $A + B + \bar{C}$, $\bar{A} + B + \bar{C}$, $A + \bar{B} + \bar{C}$.

A	B	X	Ελαχιστόρος
0	0	1	$\bar{A} \bar{B}$
0	1	0	$\bar{A} B$
1	0	0	$A \bar{B}$
1	1	0	$A B$

Σχεδιασμός κυκλώματος για $X=1$

$$E = \bar{A} \bar{B}$$

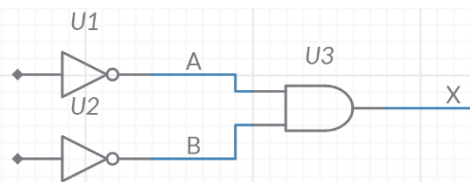


Figure 12 Αναπαράσταση λογικής συνάρτησης

$E = A \bar{B}'$ σε προσομοιωτή.

A	B	X	Μεγιστόρος
0	0	1	$\bar{A} + \bar{B}$
0	1	0	$\bar{A} + B$
1	0	0	$A + \bar{B}$
1	1	0	$A + B$

Σχεδιασμός κυκλώματος για $X=0$

$$E = (\bar{A} + B) * (A + \bar{B}) * (A + B) = \overline{A + B}$$

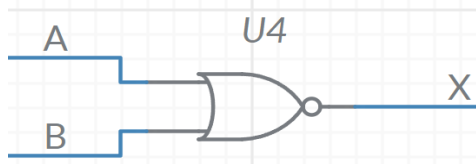


Figure 11 Αναπαράσταση λογικής συνάρτησης

$E = (A + B) * (A + B)' * (A + B) = (A + B)'$ σε προσομοιωτή.

1.2.5 Προτεραιότητες πράξεων

Στην ερμηνεία των εξισώσεων Boole υπάρχει προτεραιότητα πράξεων. Έτσι από μεγαλύτερη προς μικρότερη προτεραιότητα έχουμε τις πράξεις : NOT > AND > OR.

Προτεραιότητα	Πράξη
1	()
2	NOT
3	AND
4	OR

π.χ. $F = A + B + C * B - D$ μεταφράζεται σε: $F = A \text{ OR } B \text{ OR } C \text{ AND } B \text{ NOT } D$ και ύστερα από τη χρήση του παραπάνω κανόνα προτεραιοτήτων με την χρήση παρενθέσεων καταλήγουμε σε: $F = A \text{ OR } B \text{ OR } (C \text{ AND } (B \text{ NOT } D))$.

1.2.6 Κανονικοποίηση

Μια συνάρτηση βρίσκεται στη κανονικοποιημένη μορφή της όταν όλες οι μεταβλητές της ή το συμπλήρωμα τους εμφανίζονται ως άθροισμα ελαχιστόρων ή γινόμενο μεγιστόρων.

1.2.7 Ελαχιστοποίηση λογικών συναρτήσεων

Η ελαχιστοποίηση γίνεται με την χρήση αξιωμάτων και θεωρημάτων της άλγεβρας Boole.

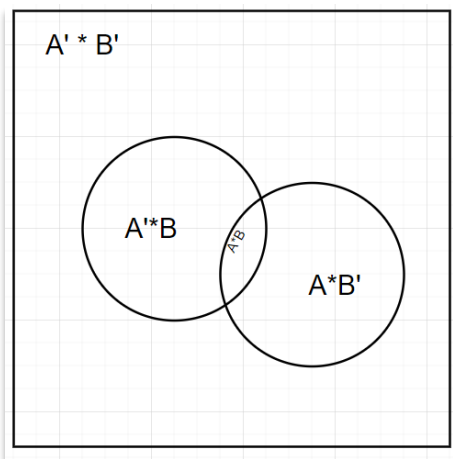
π.χ.

$$F=A*B*C*D+A*B*\bar{C}*D+A*B*C*\bar{D}+A*B*\bar{C}*\bar{D}=A*B*(C*D+\bar{C}*D+C*\bar{D}+\bar{C}*\bar{D})=A*B$$

Η παράσταση μέσα στην παρένθεση ισούται με τη μονάδα, αφού περιέχει το άθροισμα όλων των δυνατών συνδυασμών δύο λογικών μεταβλητών.

1.2.8 Διαγράμματα Venn

Η αναπαράσταση λογικών συναρτήσεων Boole μπορεί να γίνει με τα διαγράμματα Venn υπό την μορφή γραφικών απεικονίσεων. Στο παρακάτω διάγραμμα απεικονίζονται οι περιοχές: $A*B$, $\bar{A}*\bar{B}$, $\bar{A}*B$ και $A*\bar{B}$, με την περιοχή που περιλαμβάνουν και οι δύο κύκλοι να μπορεί να εκφραστεί ως $A+B$ ή $A\cup B$.



Εικόνα 1 Διάγραμμα Venn συνάρτησης δύο μεταβλητών.

1.3 Χάρτες Karnaugh

Οι χάρτες Karnaugh είναι πίνακες που χρησιμοποιούνται για την ελαχιστοποίηση εξισώσεων Boole και τους επινόησε το 1953 ο Maurice Karnaugh.

1.3.1 Χάρτες Karnaugh 3 εισόδων

Στην πάνω γραμμή ενός χάρτη Karnaugh υπάρχουν οι τιμές 00,01,11,10 (κώδικας Grey). Η τιμή 01 ακολουθείται από τη τιμή 11 και όχι από τη 10 για να υπάρχει διαφορά σε δύο γειτονικά τετράγωνα στη τιμή μίας εισόδου για τις εισόδους A και B, καθώς και η αριστερή στήλη περιέχει τις τιμές 0 και 1 που μπορεί να πάρει η είσοδος C. Κάθε τετράγωνο του χάρτη αντιστοιχεί σε μία γραμμή του πίνακα αληθείας και αναπαριστά έναν ελαχιστόρο. Τα τετράγωνα που βρίσκονται τελείως δεξιά είναι γειτονικά με τα τετράγωνα που βρίσκονται τελείως αριστερά.

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

C\AB	00	01	11	10
0	0	1	0	0
1	0	1	0	0

C\AB	00	01	11	10
0	ABC	AB'C	A'B'C	A'BC
1	ABC'	AB'C'	A'B'C'	A'BC'

Η εύρεση των εξισώσεων από τον χάρτη Karnaugh είναι ίδια με εύρεση αθροίσματος γινομένων από πίνακα αληθείας. Για το προηγούμενο πίνακα η εξίσωση θα είναι η $F = A\bar{B}C + A\bar{B}\bar{C}$. Μετά την ελαχιστοποίηση της εξίσωση με τη βοήθεια της άλγεβρας Boole προκύπτει:

$F = A\bar{B}(C + \bar{C}) = A\bar{B} * 1 = A\bar{B}$. Ο χάρτης Karnaugh θα κάνει ακριβώς την ίδια απλοποίηση γραφικά κυκλώνοντας τους άσους δύο γειτονικών τετραγώνων.

C\AB	00	01	11	10
0	ABC	AB'C	A'B'C	A'BC
1	ABC'	AB'C'	A'B'C'	A'BC'

Διατηρώντας τις μεταβλητές που είναι κοινές και στα δύο γειτονικά τετράγωνα προκύπτει $F = A\bar{B}$.

1.3.2 Ελαχιστοποίηση K-map

Η ελαχιστοποίηση ενός χάρτη Karnaugh θα πρέπει να ικανοποιεί τους ακόλουθους κανόνες:

- Χρήση του μικρότερου δυνατού αριθμού κύκλων για την κάλυψη όλων των άσων.

C\AB	00	01	11	10
0	0	1	1	0
1	0	1	1	0

C\AB	00	01	11	10
0	0	1	1	0
1	0	1	1	0

- Όλες οι τιμές εντός των κύκλων θα πρέπει να είναι 1.
- Κάθε κυκλωμένο τμήμα στο χάρτη θα πρέπει να είναι δύναμη του 2 (1, 2, 4 ή 8).

- Κάθε κύκλος πρέπει να έχει το μεγαλύτερο δυνατό μέγεθος έτσι ώστε να αναπαριστά τα λιγότερα λεκτικά.
- Ο χάρτης Karnaugh μπορεί να αναδιπλωθεί.

C\AB	00	01	11	10
0	1	0	0	1
1	1	0	0	1

C\AB	00	01	11	10
0	1	0	0	0
1	0	0	0	1

- Ένας άσος μπορεί να κυκλωθεί παραπάνω από μία φορά.

C\AB	00	01	11	10
0	0	1	1	0
1	0	1	0	0

1.3.3 Χάρτες Karnaugh 4 εισόδων

Στους χάρτες Karnaugh τεσσάρων εισόδων ισχύει ότι ακριβώς ίσχυε στους χάρτες Karnaugh 3 εισόδων με μόνη διαφορά ότι στην αριστερή στήλη του χάρτη υπάρχουν οι τιμές 00, 01, 11, 10 (κώδικας Grey) για τις εισόδους CD.

A	B	C	D	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

CD\AB	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	1	0	0	0
10	1	0	0	1

$$F = \bar{A}BC + A\bar{B}\bar{D} + A\bar{C}\bar{D}$$

Για κάθε χάρτη Karnaugh 4 εισόδων ισχύει ότι :

- Κάθε τετράγωνο με τιμή 1 αντιστοιχεί σε έναν ελαχιστόρο με 4 λεκτικά. Είναι πρώτος όρος, εάν όλα τα γειτονικά του τετράγωνα έχουν τιμή 0.
- Κάθε 2 γειτονικά τετράγωνα με τιμή 1 αντιστοιχούν σε έναν όρο με 3 λεκτικά.
- Κάθε 4 γειτονικά τετράγωνα με τιμή 1 αντιστοιχούν σε έναν όρο με 2 λεκτικά.
- Κάθε 8 γειτονικά τετράγωνα με τιμή 1 αντιστοιχούν σε έναν όρο με 1 λεκτικό.

- Κάθε 16 γειτονικά τετράγωνα με τιμή 1 αντιστοιχούν στη συνάρτηση $F = 1$.

1.3.4 Αδιάφορες τιμές σε χάρτες Karnaugh

Για να πετύχει η μείωση των γραμμών σε ένα πίνακα αληθείας χρησιμοποιείται το σύμβολο X για την αναπαράσταση των αδιάφορων τιμών του πίνακα. Αδιάφορες τιμές μπορούν να εντοπιστούν και σε εξόδους πινάκων αληθείας. Έτσι με αυτό το τρόπο επιτυγχάνετε η ακόμα μεγαλύτερη ελαχιστοποίηση της τελικής συνάρτησης του πίνακα Karnaugh. Αν κάποιο X είναι γειτονικό τετράγωνο κάποιου 1, το συγκεκριμένο X ισούται με τη τιμή 1 και στη συνέχεια πραγματοποιούνται οι απλοποιήσεις που έχουν προαναφερθεί. Σε αντίθεση με όσα X δεν γειτονεύουν με 1 τα οποία αγνοούνται.

Δεκαδική απεικόνιση	Δυαδική απεικόνιση (ABCD)	Έξοδος
0	0000	1
1	0001	0
2	0010	1
3	0011	0
4	0100	1
5	0101	0
6	0110	1
7	0111	0
8	1000	1
9	1001	0
10	1010	X
11	1011	X
12	1100	X
13	1101	X
14	1110	X
15	1111	X

Ο παραπάνω πίνακας περιγράφει παράδειγμα στο οποίο γίνεται χρήση των αδιάφορων όρων. Έστω ότι υπάρχουν τα δέκα ψηφία του δεκαδικού συστήματος αρίθμησης (1-9) και πρέπει να σχεδιαστεί ένα λογικό κύκλωμα που στην έξοδό του θα παράγει λογικό 1 μόνο, όταν στην είσοδο του κυκλώματος υπάρχει ένας δυαδικός αριθμός που αντιπροσωπεύει έναν άρτιο δεκαδικό. Δεδομένου ότι τα δέκα δεκαδικά ψηφία περιγράφονται με δέκα συνδυασμούς των τεσσάρων δυαδικών ψηφίων, ο πίνακας αλήθειας, που θα περιγράφει τη συνάρτηση θα περιλαμβάνει $2^4 = 16$ γραμμές, εκ των οποίων μόνο οι 10 θα είναι δυνατόν να επιβληθούν στην είσοδο του κυκλώματος. Οι υπόλοιπες 6 εισοδοί αναφέρονται σε συνδυασμούς οι οποίοι δεν περιγράφουν αριθμούς του δεκαδικού συστήματος και συνεπώς, είναι αδιάφοροι όροι. Ο χάρτης Karnaugh που απεικονίζει το παραπάνω πίνακα:

CD\AB	00	01	11	10
00	1	1	0	1
01	0	0	X	0
11	0	0	X	X
10	1	1	0	X

$$F = \bar{D} \bar{A} + \bar{D} \bar{B}$$

1.4 Κώδικας BCD (Binary-Coded-Decimal)

Η απεικόνιση των μονοψήφιων δεκαδικών αριθμών γίνεται με την χρήση του κώδικα BCD. Παρατηρούνται τα ίδια βάρη με των δυαδικών αριθμών, δηλαδή τα 8-4-2-1 και χρησιμοποιείται κυρίως για την αναπαράσταση αποκωδικοποιητών απεικόνισης επτά τμημάτων. Έτσι, για την μετατροπή του νούμερου 40 από τη δεκαδική βάση σε βάση BCD προκύπτει:

Decimal	BCD	
40	8421	8421
	0010	0000
	4	0

1.5 Αποκωδικοποιητής απεικόνισης επτά τμημάτων (Seven segment decoder)

Ο αποκωδικοποιητής απεικόνισης επτά τμημάτων μπορεί να δεχθεί τέσσερα bit εισόδου B0-B3 και παράγει επτά bit εξόδου A-G. Το κάθε bit από τα επτά αυτά αντιστοιχεί σε ένα led του δεξιού σχήματος που ανάβει όταν το αντίστοιχο σήμα ελέγχου είναι ενεργοποιημένο δηλαδή έχει τη τιμή ένα.

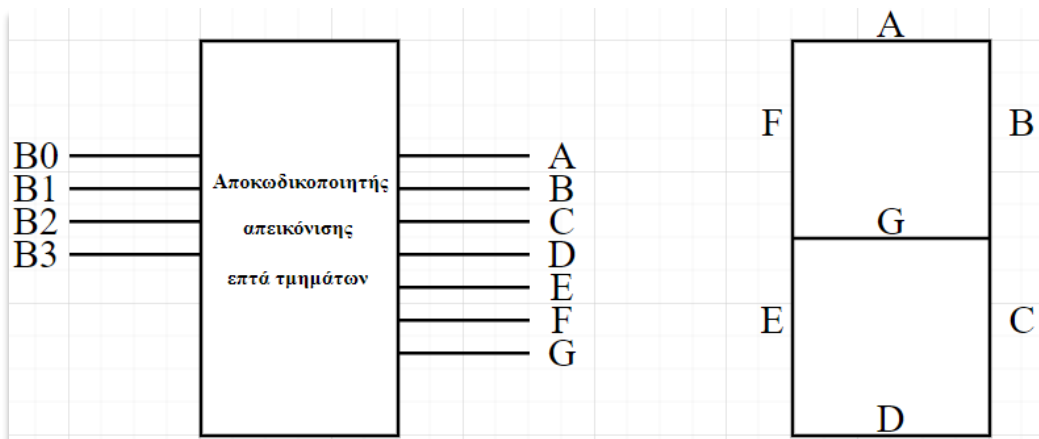


Figure 13 Αναπαράσταση αποκωδικοποιητή απεικόνισης επτά τμημάτων.

B0-B4	A	B	C	D	E	F	G
0000	1	1	1	1	1	1	0
0001	0	1	1	0	0	0	0
0010	1	1	0	1	1	0	1
0011	1	1	1	1	0	0	1
0100	0	1	1	0	0	1	1
0101	1	0	1	1	0	1	1
0110	1	0	1	1	1	1	1
0111	1	1	1	0	0	0	0
1000	1	1	1	1	1	1	1
1001	1	1	1	0	0	1	1
ΥΠΟΛΟΙΠΕΣ	0	0	0	0	0	0	0

1.6 Πολυπλέκτης 2 σε 1

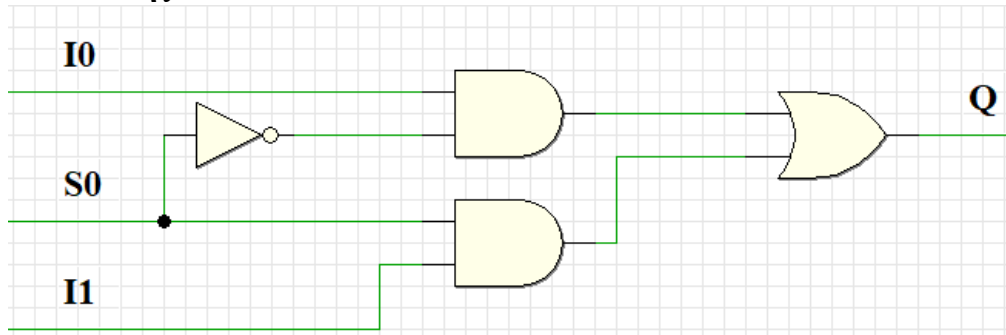


Figure 14 Αναπαράσταση πολυπλέκτη 2 σε 1 με χρήση λογικών πυλών.

Στο πολυπλέκτη 2 σε 1 γίνεται επιλογή εισόδου με βάση την τιμή του S. Αν $S = 0$ τότε $Q = I_0$ και $S = 1$ τότε $Q = I_1$.

S	I0	I1	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

X\D0D1	00	01	11	10
0	0	0	1	1
1	0	1	1	0

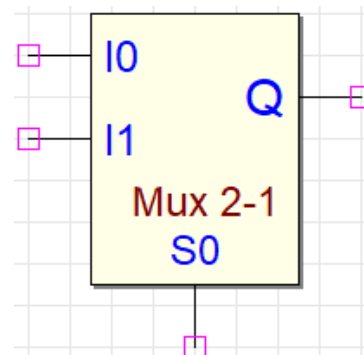


Figure 15 Πολυπλέκτης 2-1 με $I_0=D_0$, $I_1=D_1$ και $Q=X$

$$X = D_0 \bar{S} + D_1 S$$

1.7 Πολυπλέκτης 4 σε 1

Στο πολυπλέκτη 4 σε 1 γίνεται επιλογή μίας από τις τέσσερις εισόδους δεδομένων I_0 , I_1 , I_2 και I_3 με βάση την τιμή του S.

S0	S1	Q
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

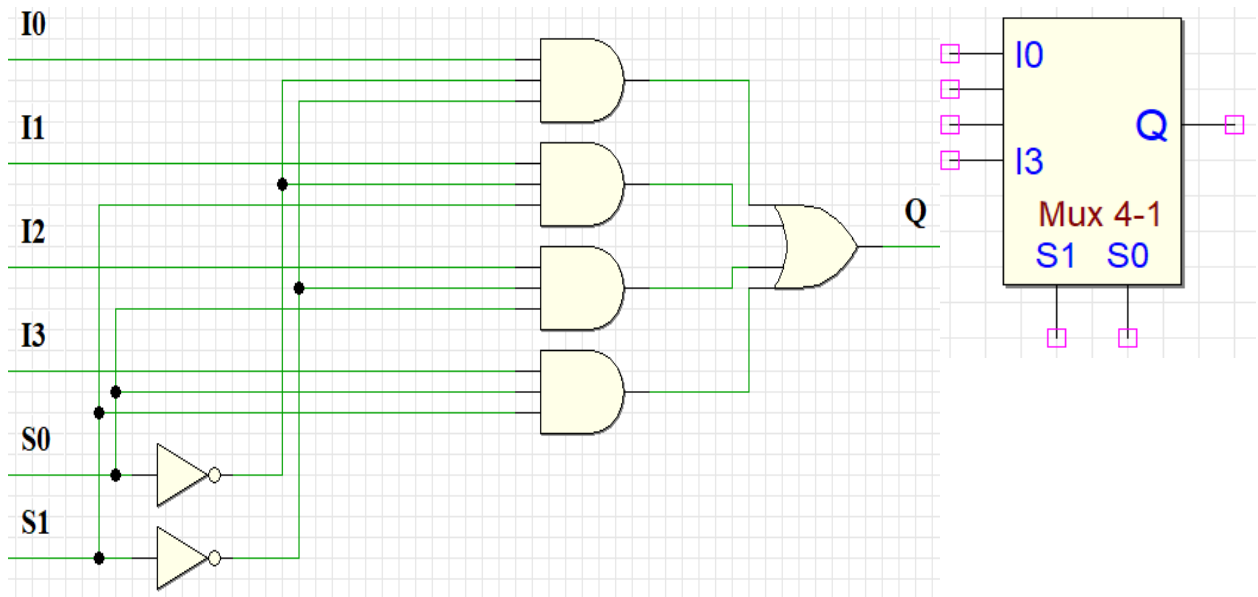


Figure 16 Υλοποίηση πολυπλέκτη σε λογική αθροίσματος γινομένων.

Εξίσωση Boole που προκύπτει για το πολυπλέκτη 4-1 είναι η: $X = \overline{S0S1} I0 + \overline{S0} S1I1 + S0\overline{S1} I2 + S0S1I3$

1.8 Αποπλέκτης 1-4

Ο αποπλέκτης 4 σε 1 συνδέει μία είσοδο IN με μία από τις τέσσερις εξόδους Q0, Q1, Q2, Q3 ανάλογα του σήματος επιλογής.

IN	S0	S1	Q0	Q1	Q2	Q3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

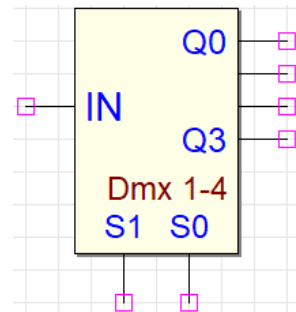


Figure 17 Απεικόνιση εξαρτήματος αποπλέκτη 1 σε 4 σε λογικό προσομοιωτή.

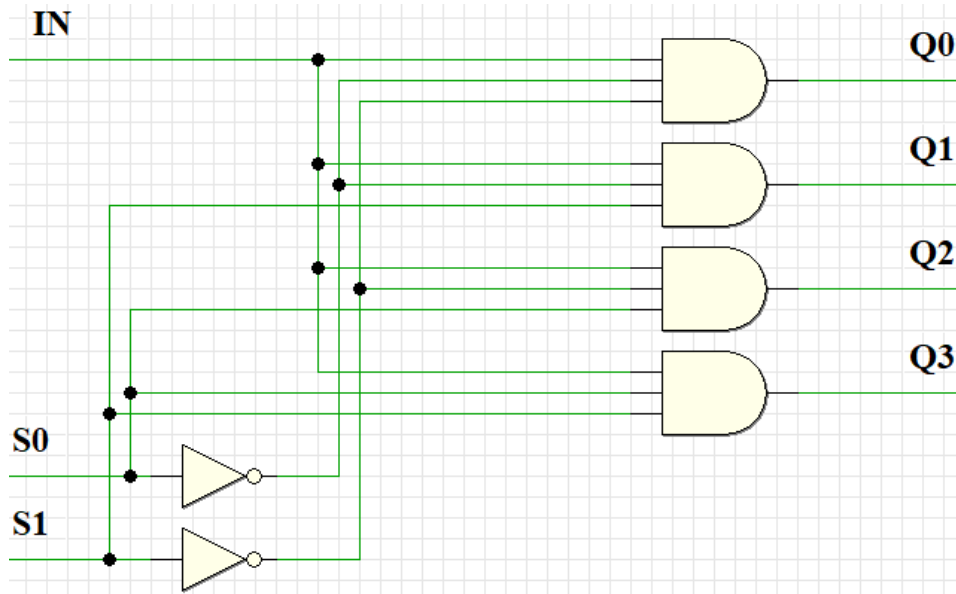
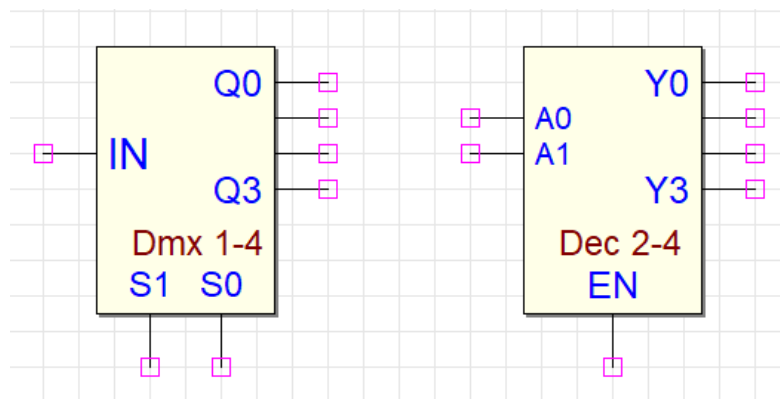


Figure 18 Υλοποίηση απολυπλέκτη σε λογική αθροίσματος γινομένων.

Ο αποπλέκτης 1 – 4 ισοδυναμεί με αποκωδικοποιητή 2 – 4 με είσοδο επίτρεψης (enable). Τα κυκλώματα με λίγες εξόδους υλοποιούνται καλύτερα με πολυπλέκτες, ενώ αυτά με πολλές εξόδους υλοποιούνται καλύτερα με αποκωδικοποιητές.



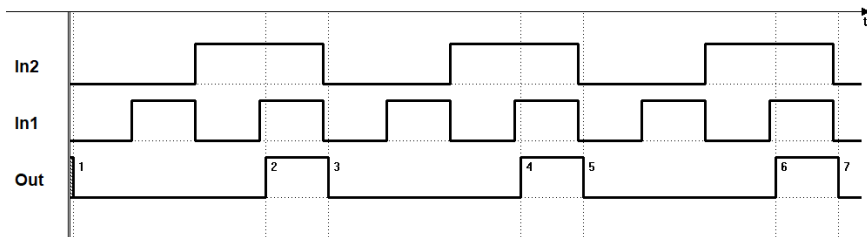
Input	S1 S0	Q0 Q1 Q2 Q3
1	00	1 0 0 0
1	01	0 1 0 0
1	10	0 0 1 0
1	11	0 0 0 1

EN	A0 A1	Y0 Y1 Y2 Y3
0	X X	0 0 0 0
1	0 0	1 0 0 0
1	0 1	0 1 0 0
1	1 0	0 0 1 0
1	1 1	0 0 0 1

1.9 Χρονισμός ψηφιακού κυκλώματος

Το διάγραμμα χρονισμού απεικονίζει τη χρονική εξέλιξη των σημάτων εισόδου και εξόδου ενός ψηφιακού κυκλώματος. Στο διάγραμμα αυτό ο οριζόντιος άξονας αντιστοιχεί στον χρόνο, ενώ ο κατακόρυφος στις λογικές στάθμες των εισόδων ή της εξόδου. Στα διαγράμματα

χρονισμού οι αδιάφορες τιμές των εισόδων συμβολίζονται με σκιασμένες περιοχές καθώς και οι αδιάφορες ή οι απροσδιόριστες τιμές των εξόδων.



Εικόνα 2 Διάγραμμα χρονισμού λογικής πύλης And.

1.10 Σύντομη και κρίσιμη διαδρομή

Σύντομη ονομάζεται η διαδρομή που διέρχεται από λιγότερες πύλες και οι πύλες αυτές έχουν τις λιγότερες εισόδους.

Κρίσιμη ονομάζεται η διαδρομή που διέρχεται από τις περισσότερες πύλες και τις περισσότερες εισόδους καθώς μειώνει την συχνότητα λειτουργίας του κυκλώματος.

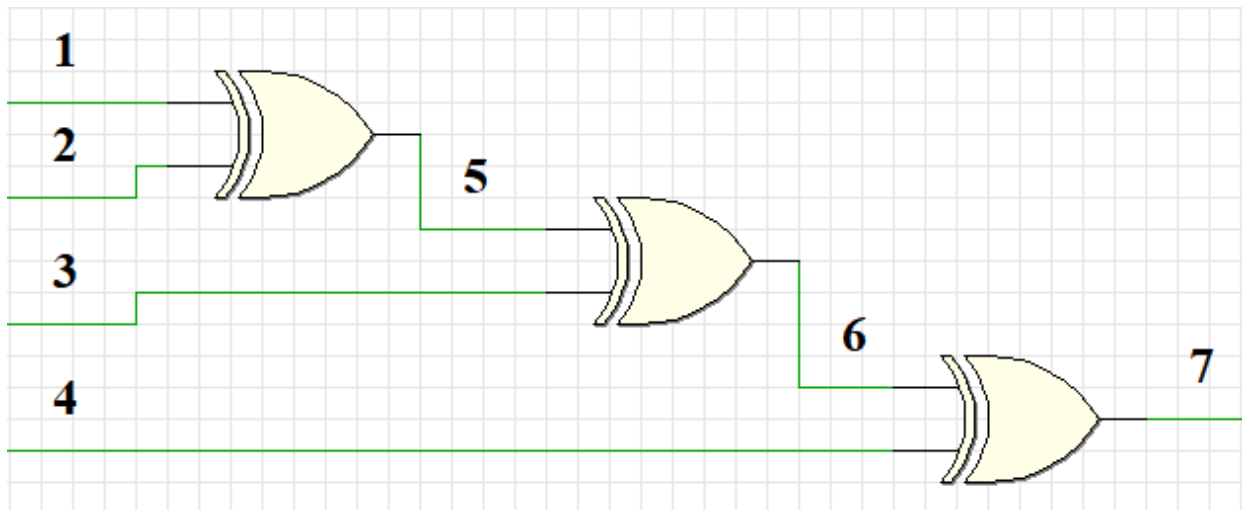


Figure 19 Παράδειγμα σύντομης και κρίσιμης διαδρομής με χρήση λογικών πυλών.

Στο παραπάνω παράδειγμα σύντομη είναι η διαδρομή που διέρχεται από τα τμήματα 4, 7 ενώ κρίσιμη είναι η διαδρομή που διέρχεται από τα τμήματα 1, 5, 6, και 7.

1.11 Ακολουθιακά κυκλώματα

Ακολουθιακά κυκλώματα ονομάζονται τα κυκλώματα που η επόμενη κατάσταση τους εξαρτάται από τις εισόδους και την παρούσα έξοδο τους. Τα ακολουθιακά κυκλώματα

αποτελούνται από συνδυαστικά κυκλώματα και στοιχεία μνήμης. Κάθε κύκλωμα με μνήμη έχει αποθηκευμένη μία κατάσταση η οποία μπορεί να αναγνωστεί και μπορεί να τροποποιηθεί.

Το στοιχείο που μπορεί να διατηρήσει σταθερές δύο καταστάσεις 0 και 1 ονομάζεται δισταθές και αποτελεί δομικό στοιχείο μνήμης. Αυτό επιτυγχάνεται με την ανάδραση δηλαδή την είσοδο του ενός στοιχείου να αποτελεί έξοδο του άλλου.

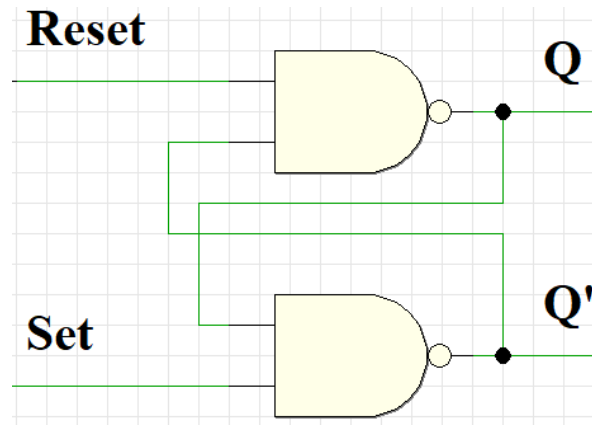


Figure 20 Αναπαράσταση ενός δισταθούς στοιχείου με χρήση λογικών πυλών.

Το κύκλωμα δεν έχει εισόδους, αλλά όταν τίθεται σε λειτουργία πηγαίνει ανεξέλεγκτα στην σταθερή κατάσταση 0 ή στην σταθερή κατάσταση 1 με ενδεχόμενο για ένα χρονικό διάστημα να παραμείνει σε μία μα τρίτη πιθανή κατάσταση, την μετασταθερή (metastable) κατάσταση.

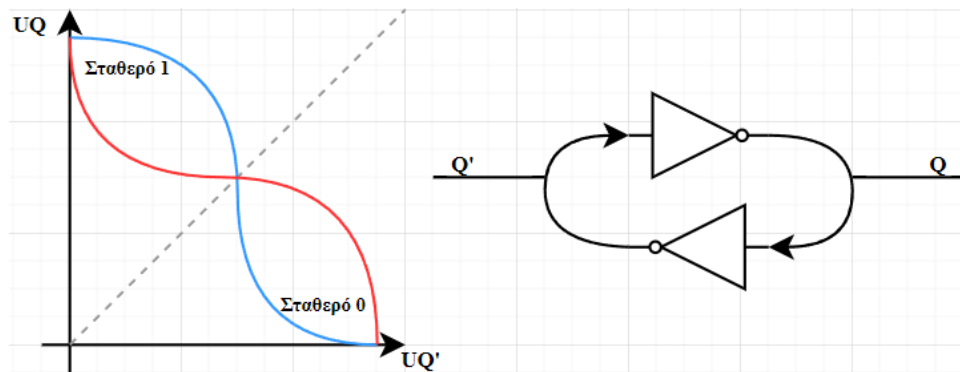


Figure 21 Απεικόνιση μετασταθερής κατάστασης σε διάγραμμα.

1.11.1 SR Latch

Το Set – Reset Latch αποτελεί το απλό δισταθές ακολουθιακό κύκλωμα. Αποτελείται από δύο πύλες NOR και σε σχέση με τους διαζευγμένους αντιστροφείς το SR Latch μπορεί ελέγξει τη κατάσταση του μέσω των εισόδων S(Set) και R(Reset) παράγοντας δύο εξόδους οι οποίες είναι συμπληρωματικές και μπορεί να κρατά αποθηκευμένη μία κατάσταση μεγέθους ενός bit (0 ή 1) που ταυτίζεται με τη δεύτερη έξοδο.

S	R	Q	\bar{Q}
0	0	Q_{prev}	\bar{Q}_{prev}
0	1	0	1
1	0	1	0
1	1	0	0

Εικόνα 3 Αποθηκεύει την προηγούμενη κατάσταση $Q = Q_{prev}$

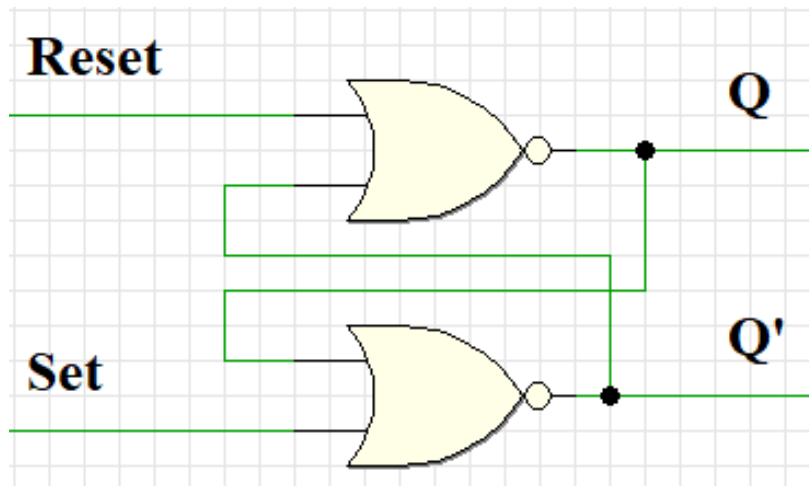


Figure 22 Τα 2,1 αντιστοιχούν στα Q' , Q αντίστοιχα.

Με $S = 1$ και $R = 0$ μεταβαίνει στη κατάσταση $Q = 1$.

Με $S = 0$ και $R = 1$ μεταβαίνει στην κατάσταση $Q = 0$.

Με $S = 0$ και $R = 0$ αποθηκεύει την κατάσταση $Q = Q_{prev}$.

Όταν ενεργοποιηθούν και τα δύο σήματα εισόδου το SR Latch καταλήγει να έχει ως εξόδους $Q = 0$ και $\bar{Q} = 0$ το οποίο θεωρείται μη έγκυρη κατάσταση με αποτέλεσμα το κύκλωμα να πηγαίνει σε ταλαντώσεις ή στη μετασταθερή κατάσταση.

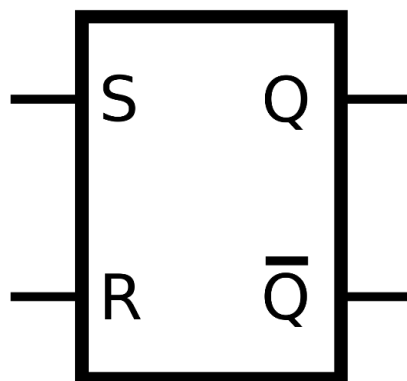


Figure 23 Εξάρτημα SR Latch σε προσομοιωτή.

1.11.2 D Latch

Το κύκλωμα Data Latch επιλύει το πρόβλημα της μη έγκυρης κατάστασης του SR Latch. Αποτελείται από δύο εισόδους: την είσοδο CLK (ρολογιού) που ελέγχει κάθε πότε το κύκλωμα μας θα αλλάζει τη κατάσταση του και την είσοδο D που καθορίζει τι τιμή πρέπει να έχει η έξοδος Q. Το κύκλωμα έχει δύο συμπληρωματικές εξόδους \bar{Q} και Q καθώς και μπορεί να διατηρήσει αποθηκευμένη μία κατάσταση Q (0 ή 1).

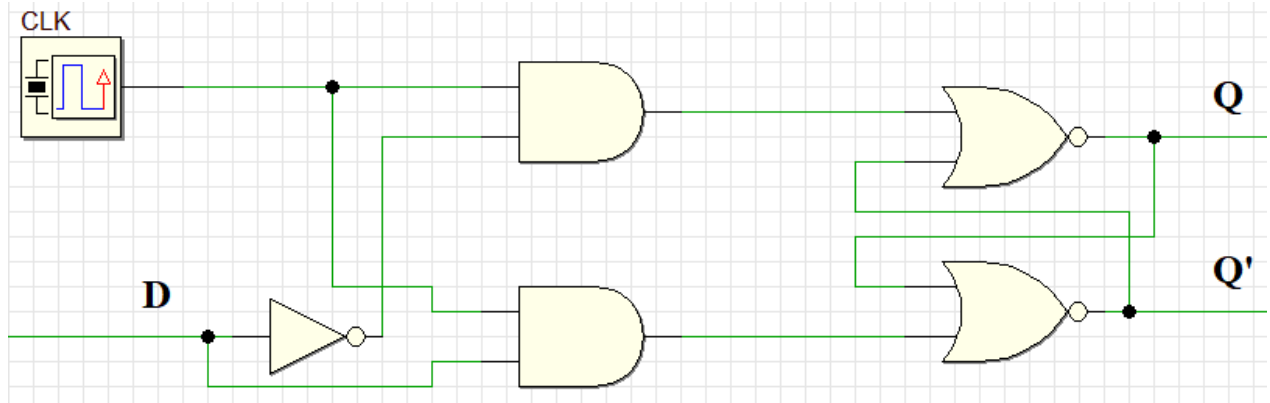


Figure 24 Απεικόνιση D Latch σε λογικό προσομοιωτή.

Μπορούμε να εντοπίσουμε ένα D Latch σε συνδυασμό δύο πυλών and μίας not και ενός στοιχείου clock.

CLK	D	Q	\bar{Q}
0	0	Q _{prev}	\bar{Q}_{prev}
0	1	Q _{prev}	\bar{Q}_{prev}
1	0	0	1
1	1	1	0

Το D Latch δεν καταλήγει σε μη έγκυρη κατάσταση λόγω της τιμής του σήματος clk που είναι το βασικό στοιχείο όλου του κυκλώματος.

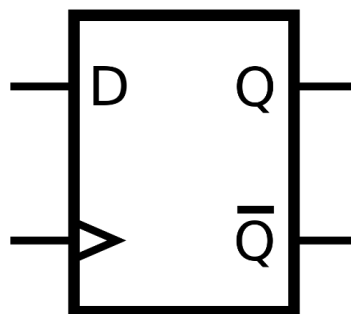


Figure 25 Εξάρτημα D Latch σε προσομοιωτή.

1.11.3 Flip - Flops

Τα διασταθή στοιχεία που ενημερώνουν την κατάσταση τους σε μία ακμή του ρολογιού (clk) στην ανερχόμενη κατάσταση ή στην κατερχόμενη ονομάζονται Flip Flops.



Figure 26 Απεικόνιση ανερχόμενων και κατερχόμενων ακμών σε διάγραμμα χρονισμού.

Το τρίγωνο στο λογικό σύμβολο ενός Flip Flop υποδηλώνει λειτουργία σε θετική ακμή του παλμού ενώ το τρίγωνο με κύκλο στην αρνητική ακμή καθώς μας παρέχονται οι δύο συμπληρωματικές έξοδοι και ασύγχρονες εισοδοι θέσης και μηδένισης.

1.11.4 Τρόποι ενεργοποίησης (triggering) Flip – Flop

Οι τρόποι ενεργοποίησης των Flip – Flop είναι δύο. Η ενεργοποίηση από θετική ή αρνητική ακμή του ρολογιού και η ενεργοποίηση τύπου Master – Slave. Ο δεύτερος τρόπος ενεργοποίησης γίνεται με την χρήση δύο latches έτσι ώστε να γίνεται ενεργοποίηση επιπέδου για να υπάρχει εναλλαγή κατάστασης μόνο μια φορά σε κάθε περίοδο ρολογιού.

1.11.5 D Flip - Flops

Το D Flip – Flop αποτελείται από δύο D Latches τα οποία ελέγχονται από ένα ρολόι και είναι συνδεδεμένα σε σειρά. Το πρώτο στην σειρά D Latch αποκαλείται αφέντης ενώ το δεύτερο σκλάβος και χρησιμοποιεί το Master – Slave τρόπο ενεργοποίησης.

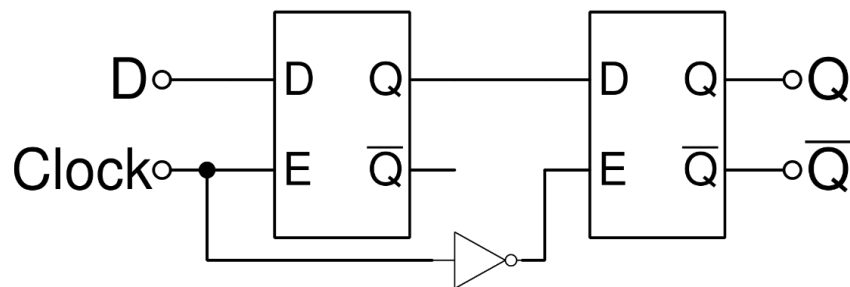


Figure 27 D Flip Flop σε διάταξη master-slave.

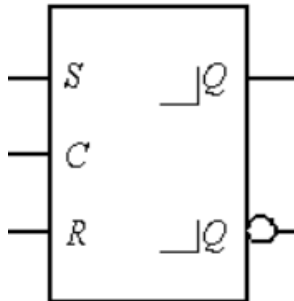
$$Q(t+1) = D$$

CLK	D	Q	\bar{Q}	ΠΕΡΙΓΡΑΦΗ
«↓»0	X	Q	\bar{Q}	Σταθερή μνήμη
«↑»1	0	0	1	Reset Q = 0
«↑»1	1	1	0	Set Q = 1

Εικόνα 4 Τα σύμβολα ↓ και ↑ δηλώνουν την φορά της ακμής του ρολογιού.

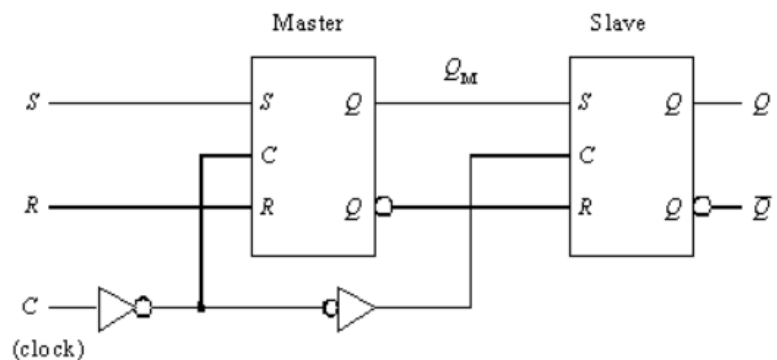
1.11.6 SR Flip – Flops

Για να αποφευχθεί η ασταθής συμπεριφορά γίνεται σύνδεση δύο latches σε Master – Slave διάταξη.



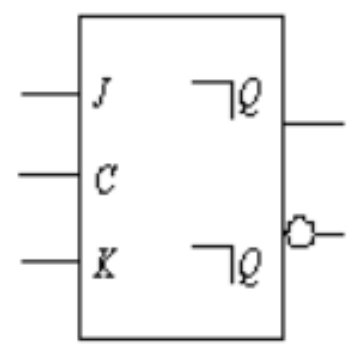
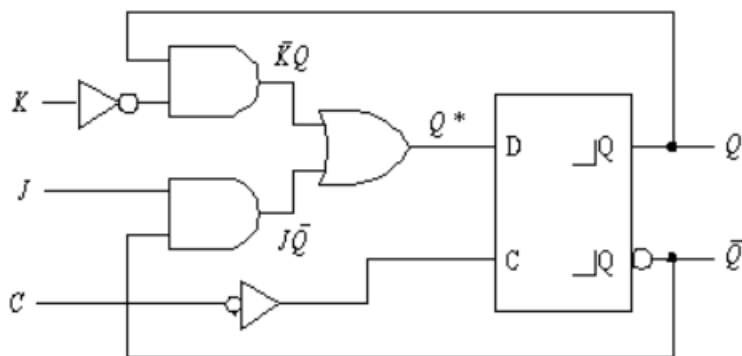
Εικόνα 5 Pulse Triggered SR FF logic symbol

S	R	C	Q	\bar{Q}	
0	0	1	Q0	$\bar{Q}0$	Store
0	1	1	0	1	Reset
1	0	1	1	0	Set
1	1	1	1	1	Disallowed
X	X	0	Q0	$\bar{Q}0$	Store



1.11.7 JK Flip – Flops

Τα JK Flip – Flop λειτουργούν όπως τα SR Flip – Flop ωστόσο δεν επιτρέπουν τη χρήση του συνδυασμού $S = R = 1$ διότι μπορούν να εναλλάσσουν τιμές καταστάσεων όταν παρατηρείται ο συνδυασμός $J = K = 1$. Αν το $clk = 1$ για αρκετό χρόνο τότε οι έξοδοι θα εναλλάσσονται συνεχώς λόγω της χρήσης Master – Slave Flip Flop.



ΕΙΚΟΝΑ 6 JK FF LOGIC SYMBOL

Figure 28 Δημιουργία JK Flip Flop απο D Flip Flop.

$$Q(t+1) = J\bar{Q} + \bar{K}Q$$

Q	J	K	Q(t+1)
0	0	0	0 Διατήρηση
0	0	1	0 Set - Reset
0	1	0	1 Set - Reset
0	1	1	1 Αντιστροφή
1	0	0	1 Διατήρηση
1	0	1	0 Set - Reset
1	1	0	1 Set - Reset
1	1	1	0 Αντιστροφή

Με J , K = 1 η θετική ακμή εξασφαλίζει ότι θα γίνεται μία φορά εναλλαγή.

1.11.8 T Flip – Flop

Το T Flip - Flop πρόκειται για μια παραλλαγή του JK Flip - Flop, δέχεται μόνο μία είσοδο T ενώ όταν T = 1 αντιστρέφει τη κατάσταση του.

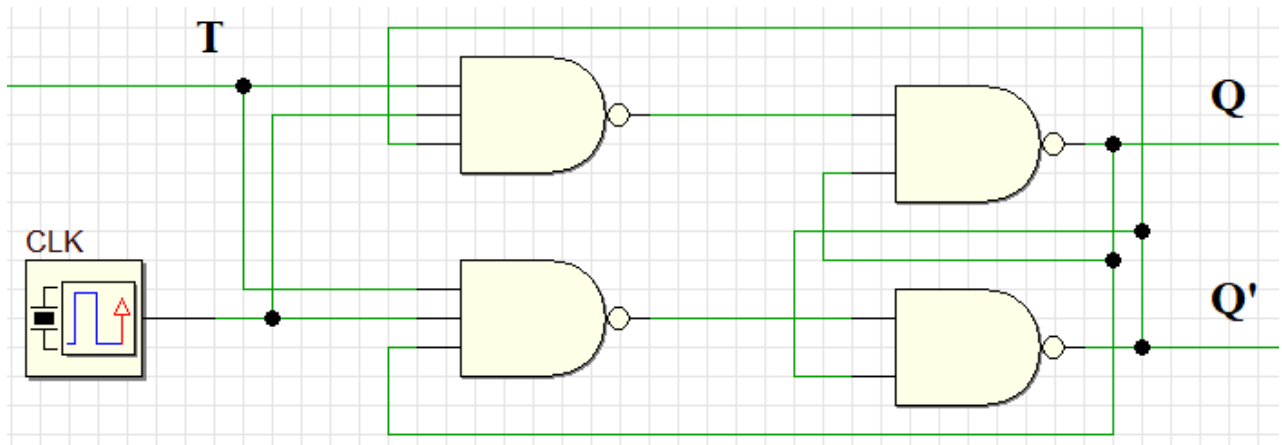
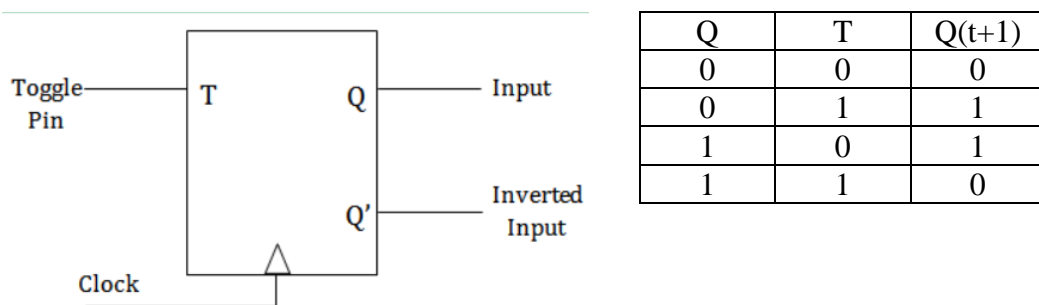
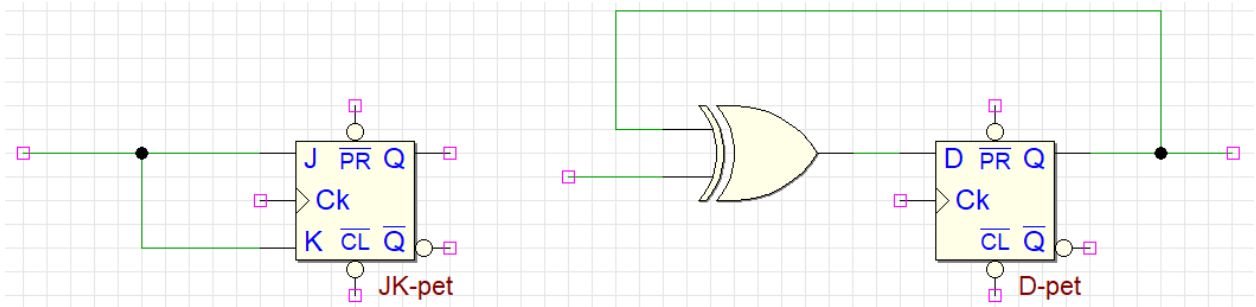


Figure 29 Δημιουργία T Flip Flop από λογικές πύλες.



$$Q(t+1) = T \text{ xor } Q$$

Η δημιουργία του T Flip – Flop μπορεί να γίνει από JK και D Flip – Flop με τους ακόλουθους τρόπους :



Με τις εισόδους των παραπάνω Flip – Flop να αφορούν την είσοδο T.

1.12 Σύγχρονα ακολουθιακά κυκλώματα

Οι κυκλικές διαδρομές που παρατηρούνται στα ασύγχρονα ακολουθιακά κυκλώματα μπορούν να δημιουργήσουν ασταθή συμπεριφορά ή συνθήκες ανταγωνισμού (εναλλαγή Q πριν το clk αλλάξει) στο κύκλωμα, φαινόμενα τα οποία είναι ανεπιθύμητα.

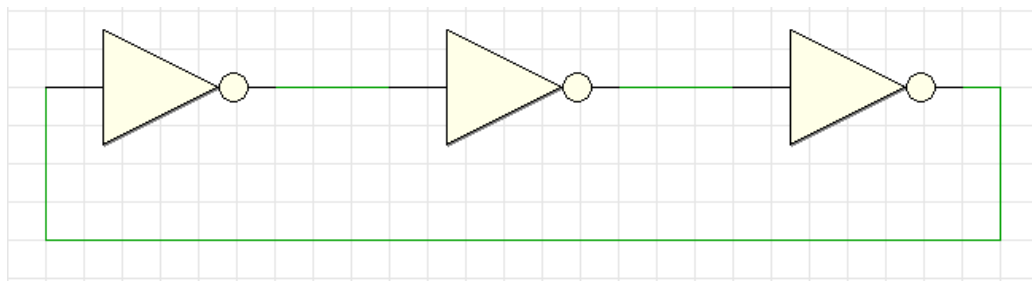


Figure 30 Αναπαράσταση κυκλικής διαδρομής.

Το παραπάνω πρόβλημα λύνεται με την είσοδο καταχωριστών στο κύκλωμα μας.

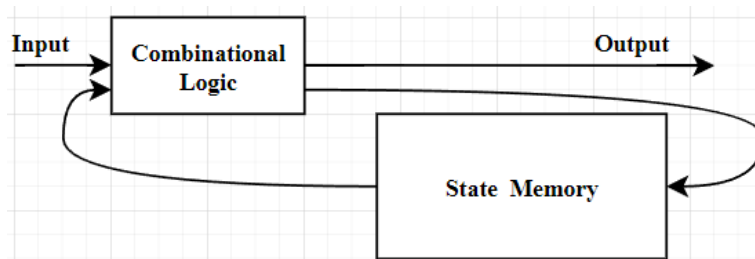


Figure 31 Τοποθέτηση καταχωρητών σε κυκλική διαδρομή.

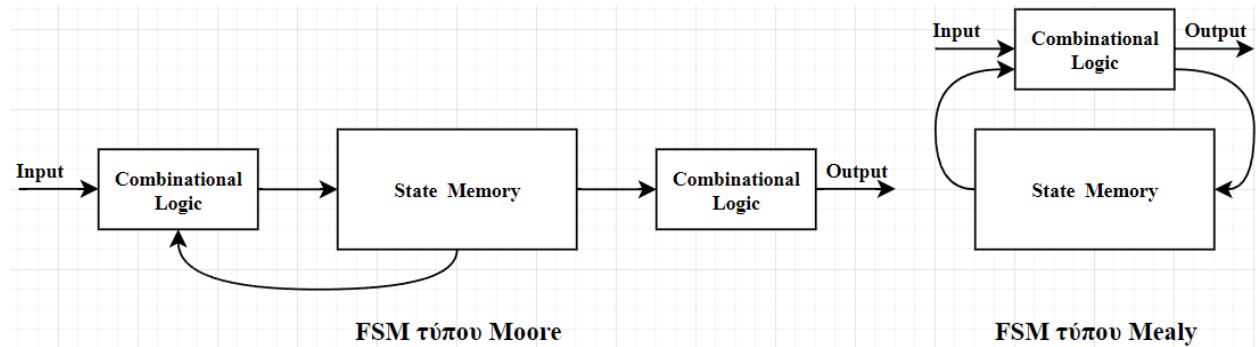
Κάθε στοιχείο του κυκλώματος θα πρέπει να είναι καταχωρητής (τουλάχιστον ένας) ή συνδυαστικό κύκλωμα, κάθε κυκλική διαδρομή θα πρέπει να παρεμβάλλεται από ένα καταχωρητή καθώς και κάθε καταχωρητής του κυκλώματος μας θα πρέπει να έχει την ίδια είσοδο clk. Χαρακτηριστικό των σύγχρονων ακολουθιακών κυκλωμάτων είναι ότι μπορούν να περιγράψουν πλήρως την επόμενη τους κατάσταση και τη τιμή κάθε εξόδου τους για οποιοδήποτε συνδυασμό εισόδων.

1.13 Finite State Machines FSM

Το σύγχρονο ακολουθιακό κύκλωμα που έχει A εισόδους, B εξόδους, σήμα επαναφοράς και σήμα ρολογιού μπορεί να θεωρηθεί μια μηχανή πεπερασμένων καταστάσεων. Αν διαθέτει έναν καταχωρητή καταστάσεων n bit που αποθηκεύει έως 2^n καταστάσεις, next state logic που θα μπορεί να υπολογίσει την επόμενη κατάσταση $Q(t+1)$ ως συνάρτηση του Q και των εισόδων

A και output logic που υπολογίζει τις τιμές των εξόδων ως συνάρτηση της κατάστασης $Q(t)$ (τρέχουσας) και των εισόδων A (optional). Το σήμα reset είναι υποχρεωτικό στο καταχωρητή καταστάσεων ώστε να ορίζουμε την αρχική κατάσταση όταν η μηχανή πεπερασμένων καταστάσεων ενεργοποιείται για πρώτη φορά.

Οι μηχανές πεπερασμένων καταστάσεων χωρίζονται σε δύο κατηγορίες τις FSM τύπου Moore και τις FSM τύπου Mealy. Στις πρώτες οι εξόδοι εξαρτώνται μόνο από την παρούσα κατάσταση (current state) ενώ στις δεύτερες εξαρτώνται και από τις εισόδους.



Στις μηχανές καταστάσεων τύπου Moore παρατηρείται μεγαλύτερη ταχύτητα εξόδου και μεγαλύτερος αριθμός καταστάσεων ενώ στις μηχανές καταστάσεων τύπου Mealy παρατηρείται μεγαλύτερη ταχύτητα της λογικής επόμενης κατάστασης ($Q(t+1)$).

1.14 Διάγραμμα μεταβολής κατάστασης

Οι πιθανές καταστάσεις ενός FSM, το πως αυτές μεταβάλλονται με τις εισόδους του κυκλώματος κατά την ανερχόμενη ακμή του clk και τις εξόδους τύπου Moore / Mealy περιγράφονται από το διάγραμμα μεταβολής καταστάσεων. Οι κύκλοι προσδιορίζουν το current state $Q(t)$ ενώ τα βέλη την μετάβαση στην επόμενη κατάσταση με την είσοδο να αναγράφεται πάνω στο βέλος.

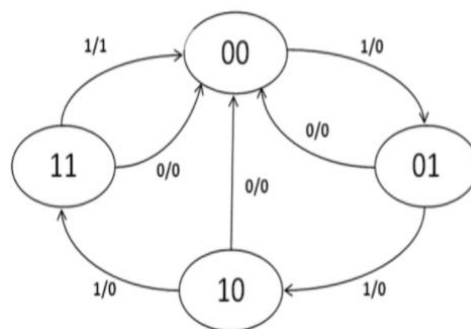


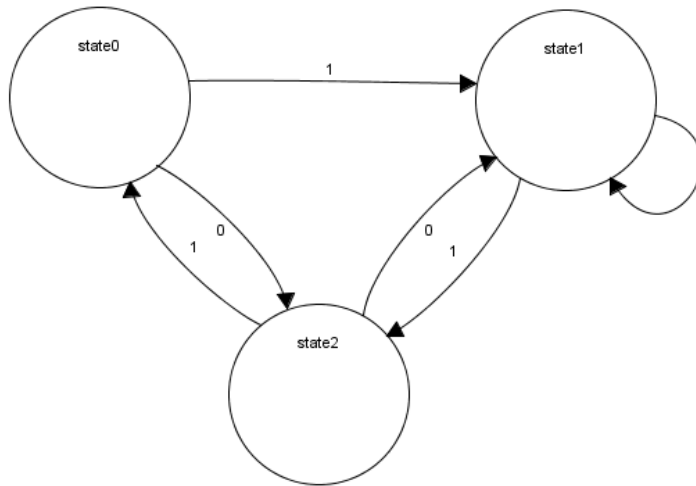
Figure 32 A, B states με 2 εισόδους x, y .

1.15 Σχεδίαση FSM

Η διαδικασία σχεδίασης μηχανών FSM ξεκινά με την ανάλυση και το καθορισμό των εισόδων, των εξόδων και όποιων διακριτών τιμών υπάρχουν σε συνδυασμό με τις μεταβάσεις που υπάρχουν σε αυτές τις καταστάσεις. Διακριτή χαρακτηρίζεται η κατάσταση όπου μία έξοδος εξαρτάται από τη τρέχουσα κατάσταση ή έχουν τις ίδιες εισόδους αλλά διαφορετικές επόμενες καταστάσεις ανεξάρτητη της εισόδου. Στην συνέχεια, γίνεται σχεδιασμός των διαγραμμάτων μεταβολής κατάστασης και του πίνακα μεταβολής κατάστασης (πίνακας αληθείας). Πραγματοποιείται μετατροπή σε πίνακα αληθείας από κωδικοποίηση των καταστάσεων. Γίνεται απλοποίηση με χρήση χάρτη Karnaugh και εξισώσεων boole για τις

επόμενες καταστάσεις καθώς και υπολογισμός των εξόδων με τον αντίστοιχο πίνακα αληθείας. Τέλος, γίνεται ξανά απλοποίηση με χρήση χάρτη Karnaugh και εξισώσεων Boole για τις εξόδους με τελικό βήμα τη σχεδίαση του διαγράμματος.

Με βάση τη παραπάνω μεθοδολογία θα σχεδιαστεί το σχηματικό διάγραμμα του παρακάτω FSM.



Q(t)	Input	Q(t+1)
State0	0	State2
State0	1	State1
State1	0	State1
State1	1	State2
State2	0	State1
State2	1	State0

State	Δυαδική Κωδικοποίηση
State0	00
State1	01
State2	10

Q(t)	Input	Q(t+1)
0 0	0	1 0
0 0	1	0 1
0 1	0	0 1
0 1	1	1 0
1 0	0	0 1
1 0	1	0 0

Πραγματοποιείται ελαχιστοποίηση με K-map για τους παραπάνω κύκλους και υπολογίζονται οι εξισώσεις Boole των μεταβλητών των επόμενων καταστάσεων.

Input\Q(t)Q(t')	00	01	11	10
0	1	0	X	0
1	0	1	X	0

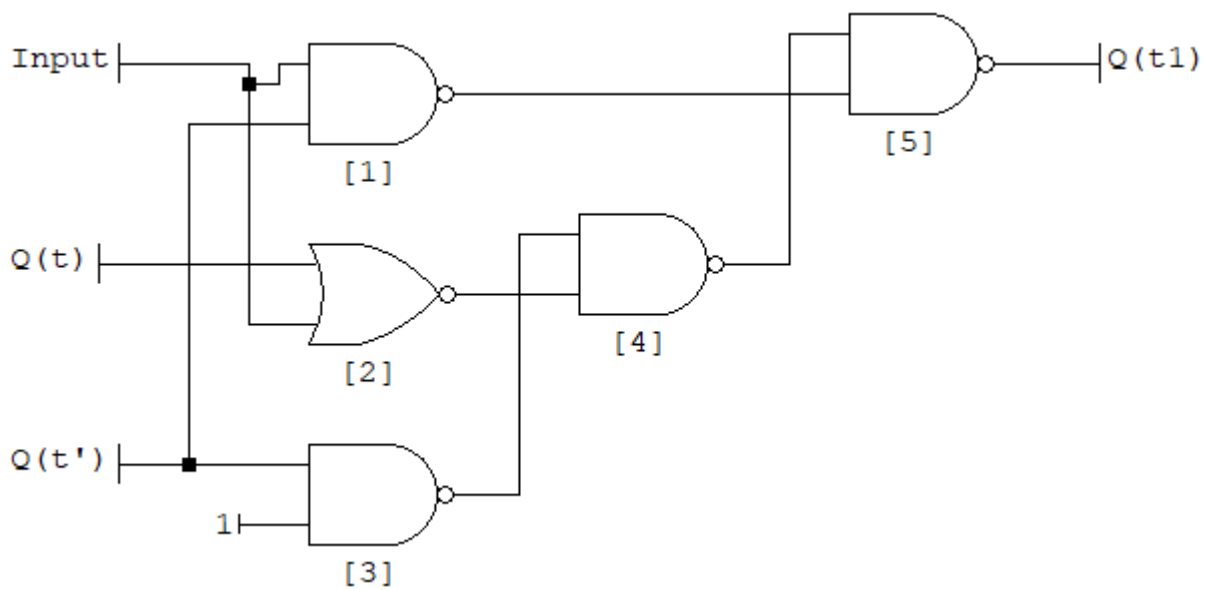
Input\Q(t)Q(t')	00	01	11	10
0	0	1	X	1
1	1	0	X	0

$$Q(t+1) = \overline{Input} \overline{Q(t)} \overline{Q(t')} + Input Q(t')$$

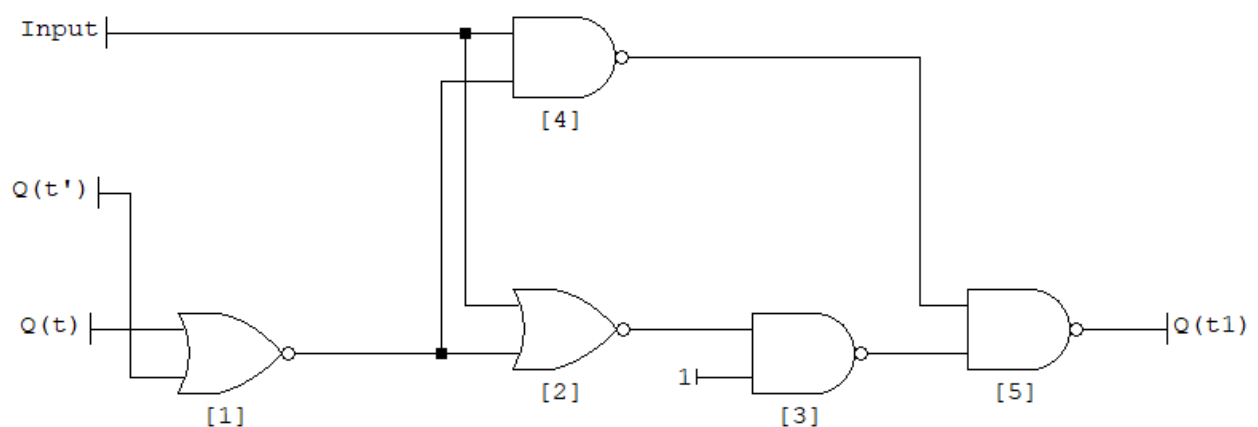
$$Q(t+1) = \overline{Input} Q(t') + \overline{Input} Q(t) + Input \overline{Q(t)} \overline{Q(t')}$$

$$Q(t+1) \text{ minimized} = Input \overline{Q(t')} \overline{Q(t)} + \overline{Input} Q(t') + \overline{Input} Q(t);$$

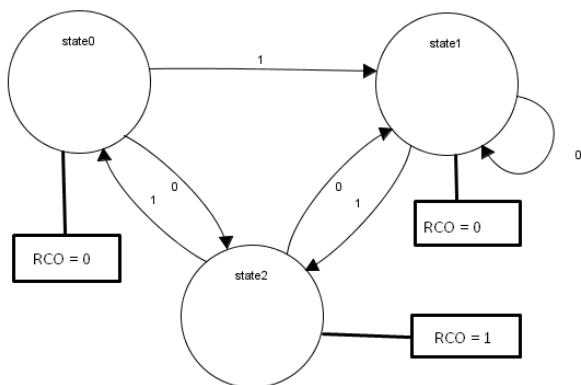
Απεικόνιση συνάρτησης $Q(t+1)$ με χρήση πυλών:



Απεικόνιση απλοποιημένης συνάρτησης $Q(t+1)$ με χρήση πυλών:



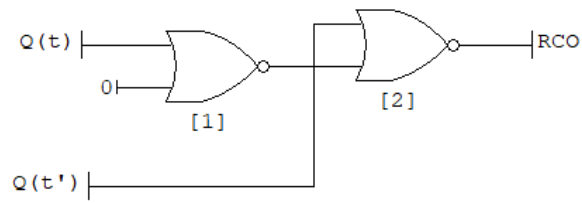
Εάν υπήρχε απαίτηση ripple carry output (RCO) για όταν το κύκλωμα βρισκόταν στην κατάσταση 2 τότε:



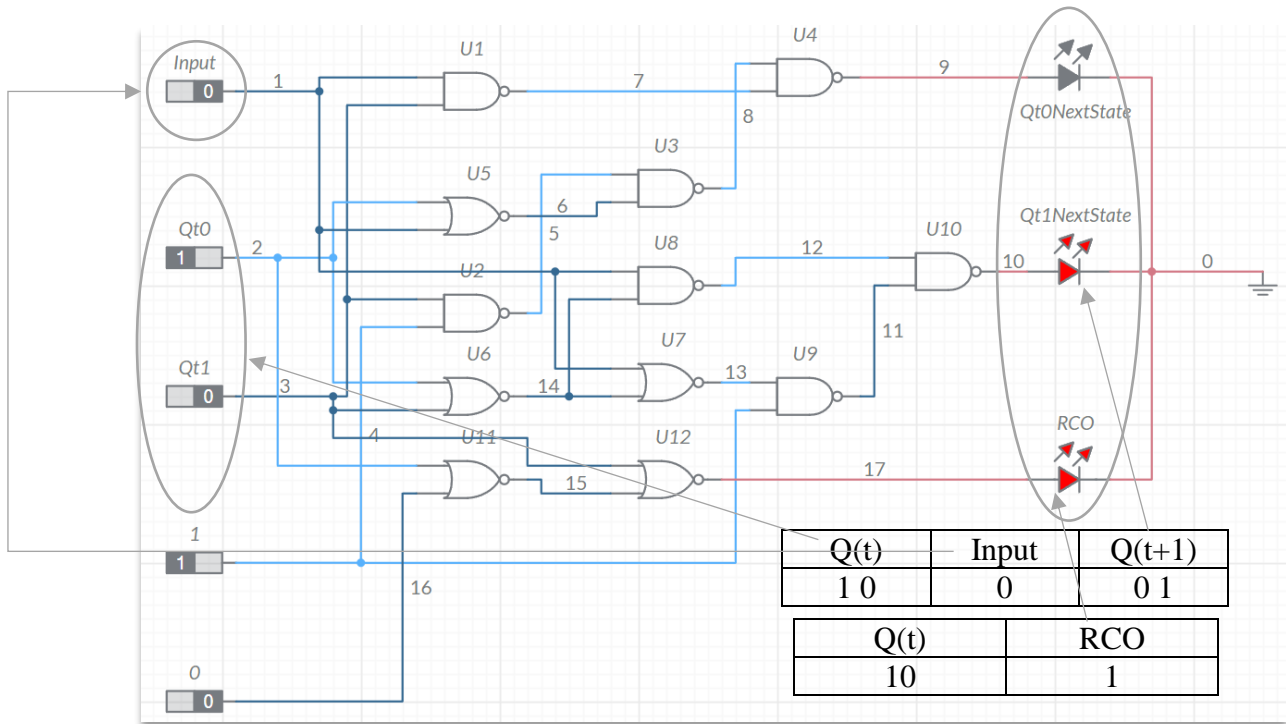
Q(t)	RCO
00	0
01	0
10	1

$$RCO = Q(t) \overline{Q(t')}$$

Απεικόνιση συνάρτησης RCO με χρήση πυλών:



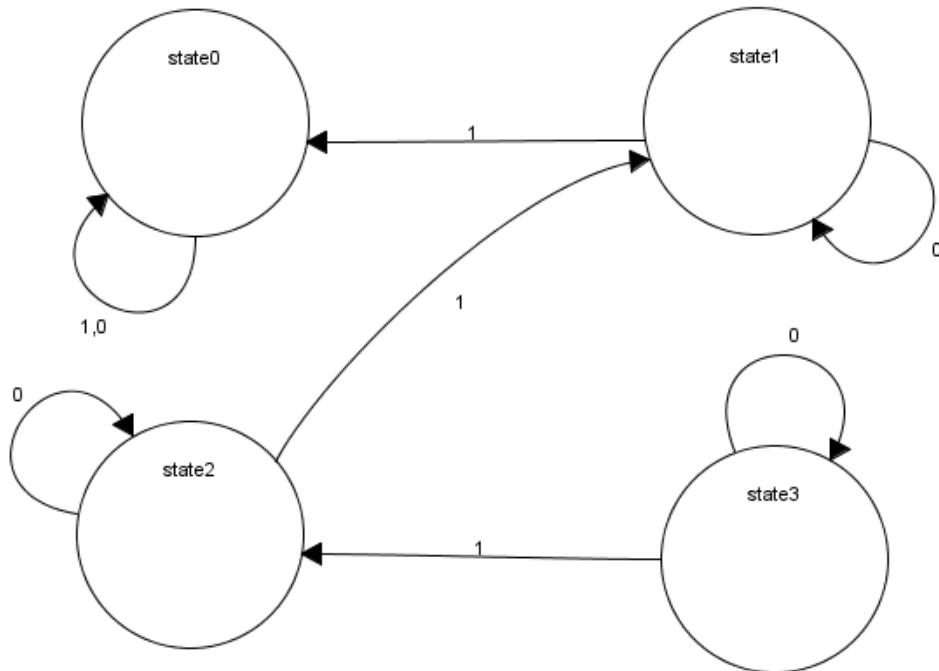
Με συνδυαστικό συνολικό διάγραμμα $Q(t+1)$ $Q(t+1)$ και RCO:



Εικόνα 7 Συνδυαστικό κύκλωμα για είσοδο 10 και input 0

1.16 Δημιουργία 2-ψήφιου συγχρόνου δυαδικού μετρητή

Ο μετρητής για Input = 1 θα εκτελεί αντίστροφη μέτρηση (3, 2, 1, 0) ενώ για Input = 0 θα παραμένει στην ίδια κατάσταση.



Για το παραπάνω διάγραμμα θα χρησιμοποιηθούν 2 T Flip - Flop καθώς και θα χρειαστεί έξοδος μεγέθους 2-bit για την αναπαράσταση του αποτελέσματος της κατάστασης στην οποία βρίσκεται το κύκλωμα που θα είναι ίδιο με την νούμερο της κατάστασης. Τα Ta, Tb συνδέονται πριν τα flip flops ενώ η εξίσωση του πεδίου output μετά τα flip flop.

Q(t)	Input	Q(t+1)	Ta Tb	Output
00	0	00	00	00
00	1	00	00	00
01	0	01	00	01
01	1	00	01	00
10	0	10	00	10
10	1	01	11	01
11	0	11	00	11
11	1	10	11	10

Για T_a :

$$T_a = Q(t)Q(t')'Input + Q(t)Q(t')Input$$

$$T_a = InputQ(t)$$

Για T_b :

$$T_b = Q(t)'Q(t')Input + Q(t)Q(t')'Input + Q(t)Q(t')Input'$$

Για **Output** :

Input\Q(t+1)Q(t+1')	00	01	11	10
0	0	0	1	1
1	0	0	1	0

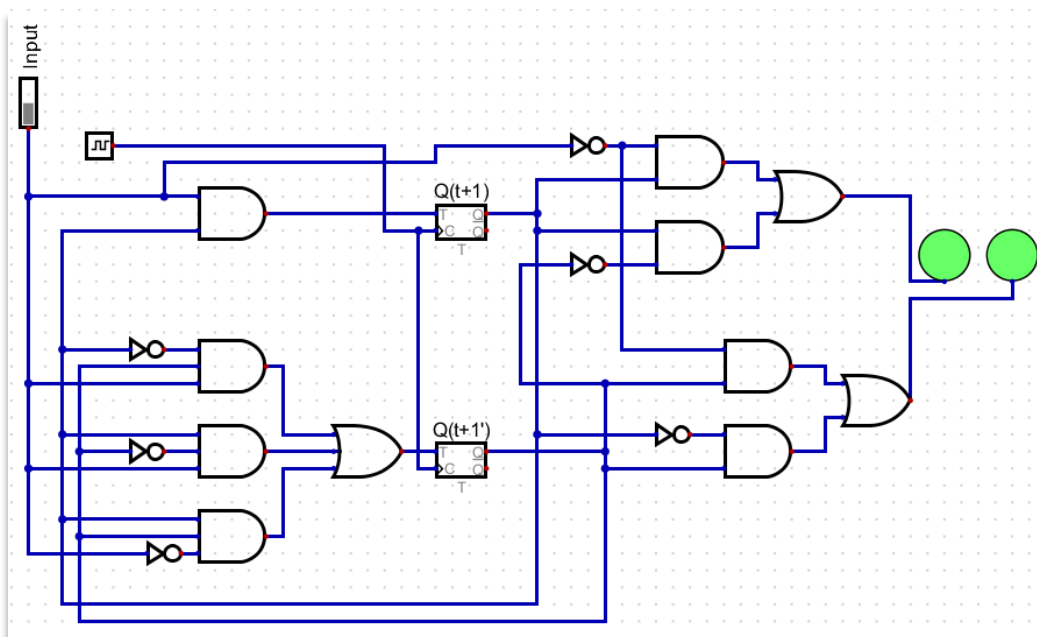
$$Output = Input' Q(t+1) + Q(t+1) Q(t+1')$$

Για **Output** :

Input\Q(t+1)Q(t+1')	00	01	11	10
0	0	1	1	0
1	0	1	0	0

$$Output = Input' Q(t+1') + Q(t+1)' Q(t+1')$$

Με τελικό κύκλωμα που εκτελεί τη λειτουργία που περιγράφει το FSM του 1.16 να είναι το ακόλουθο:



2 Προσομοιωτές Ψηφιακών Κυκλωμάτων

Η ανάγκη για δημιουργία, έλεγχο και βελτιστοποίηση της συμπεριφοράς και σωστής λειτουργίας ψηφιακών κυκλωμάτων που απαρτίζονται από λογικές πύλες σε επίπεδα στα οποία μπορούν να λειτουργήσουν ώθησε στην δημιουργία λογισμικού για την δημιουργία και προσομοίωση κυκλωμάτων. Με ένα προσομοιωτή δίνεται στο χρήστη του η δυνατότητα να πετύχει τη δημιουργία ψηφιακών κυκλωμάτων που περιγράφονται από λογικές πύλες, πίνακες αληθείας, λογικές εκφράσεις και διαγράμματα καταστάσεων. Έτσι επιτυγχάνεται η μοντελοποίηση του εκάστοτε κυκλώματος από ένα σύνολο εισόδων και εξόδων που περιγράφουν το κύκλωμα αυτό.

2.1 Digital Logic Sim

Ο προσομοιωτής Digital Logic Sim [1] είναι ένας δωρεάν open source προσομοιωτής για Windows\MacOS\Linux του έτους 2020. Πρόκειται για ένα προσομοιωτή επιπέδου πύλης με συνολικό μέγεθος 54.9 mb.

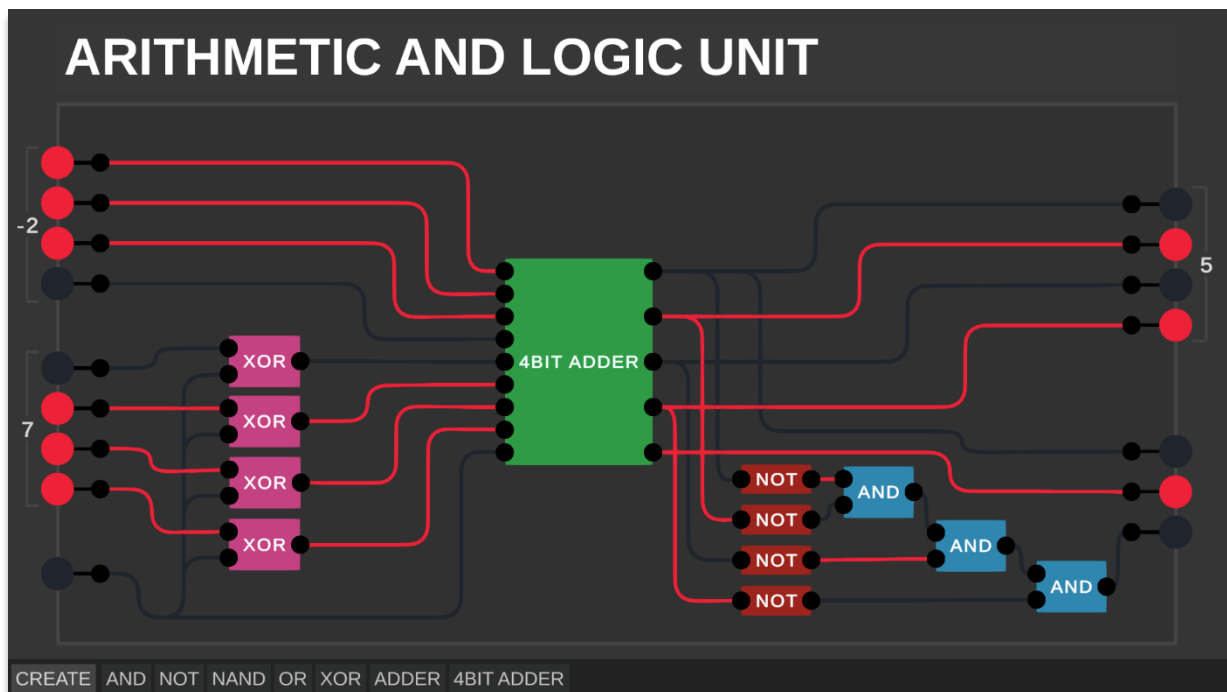


Figure 33 Διεπαφή χρήστη στο Digital Logic Sim.

2.1.1 Λειτουργικά χαρακτηριστικά

Το Digital Logic Sim περιέχει μόνο τις λογικές πύλες and και not καθώς και μπορεί να δεχθεί έως 2 εισόδους στην πύλη and. Τα καλώδια μπορούν να πάρουν προσαρμοσμένη διαδρομή και να αναδείξουν τη τιμή τους κατά τη διάρκεια της προσομοίωσης όλα μαζί με το χρώμα κόκκινο να αντιστοιχεί στην τιμή 1 ενώ με μαύρο το 0, με τις εισόδους να απεικονίζονται στα αριστερά και τις εξόδους στα δεξιά. Στο συγκεκριμένο προσομοιωτή μπορούν να σχεδιαστούν κυκλώματα με τις πύλες and και not, να γίνουν αποθήκευση σαν νέο εξάρτημα και μετά να χρησιμοποιηθούν σαν συντόμευση του εξαρτήματος αυτού. Όπως φαίνεται στη παραπάνω φωτογραφία έχουν δημιουργηθεί οι Nand, OR, Xor, Adder και 4bit adder και μπορούν να χρησιμοποιηθούν σαν συντομεύσεις. Τέλος, μπορεί να υλοποιήσει λειτουργίες save και load για το κύκλωμα μας.

2.2 Multimedia Logic

Ο προσομοιωτής Multimedia Logic [2] είναι ένας δωρεάν windows based προσομοιωτής του έτους 2013. Πρόκειται για ένα προσομοιωτή επιπέδου πύλης με συνολικό μέγεθος 1.7 mb. Οι καταστάσεις που μπορεί το κύκλωμα να βρεθεί περιγράφονται από τις τιμές 0 (Low\Off\False), 1 (Hight\On\True), U (Unkown), - (Pulled down, συμπεριφέρεται σαν 0 αλλά μπορεί και να ξεπεραστεί), + (Pulled up, συμπεριφέρεται σαν 1 αλλά μπορεί και να ξεπεραστεί) και ? (Floating, συμπεριφέρεται σαν U αλλά μπορεί και να ξεπεραστεί).

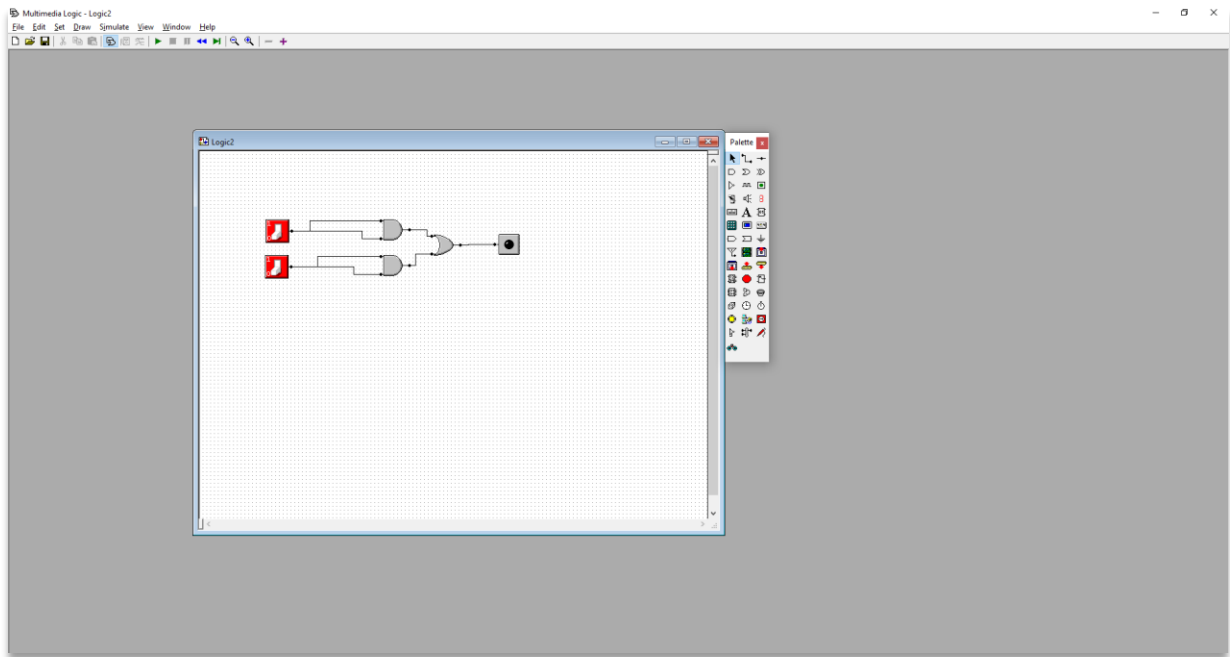


Figure 34 Διεπαφή χρήστη στο Multimedia Logic.

2.2.1 Λειτουργικά χαρακτηριστικά

Το Multimedia Logic περιέχει μόνο τις λογικές πύλες and, or, xor και not καθώς και μπορεί να δεχθεί έως 4 εισόδους στις πύλες αυτές. Τα καλώδια μπορούν να πάρουν προσαρμοσμένη διαδρομή μόνο με την χρήση του node από το palette καθώς και να αναδείξουν τη τιμή τους με το Toggle Probe κατά τη διάρκεια της προσομοίωσης ένα την φορά και όχι όλα μαζί. Υπάρχουν Flip Flops και εξάρτημα clock καθώς και πολυπλέκτες και μετρητές. Τέλος, μπορεί να υλοποιήσει λειτουργίες αποθήκευσης και εκτύπωσης για το κύκλωμα μας.

2.3 CircuitLogix

Ο προσομοιωτής CircuitLogix – Student Edition [3] είναι ένας προσομοιωτής του 2011 που διανέμεται σε δωρεάν δοκιμαστική έκδοση 15 ημερών με τη χρήση των ακαδημαϊκών στοιχείων του χρήστη εάν υπάρχουν, υποστηρίζεται από Windows\MacOS\Linux και πρόκειται για ένα προσομοιωτή επιπέδου πύλης και τρανζίστορ. Τέλος, έχει συνολικό μέγεθος 15mb.

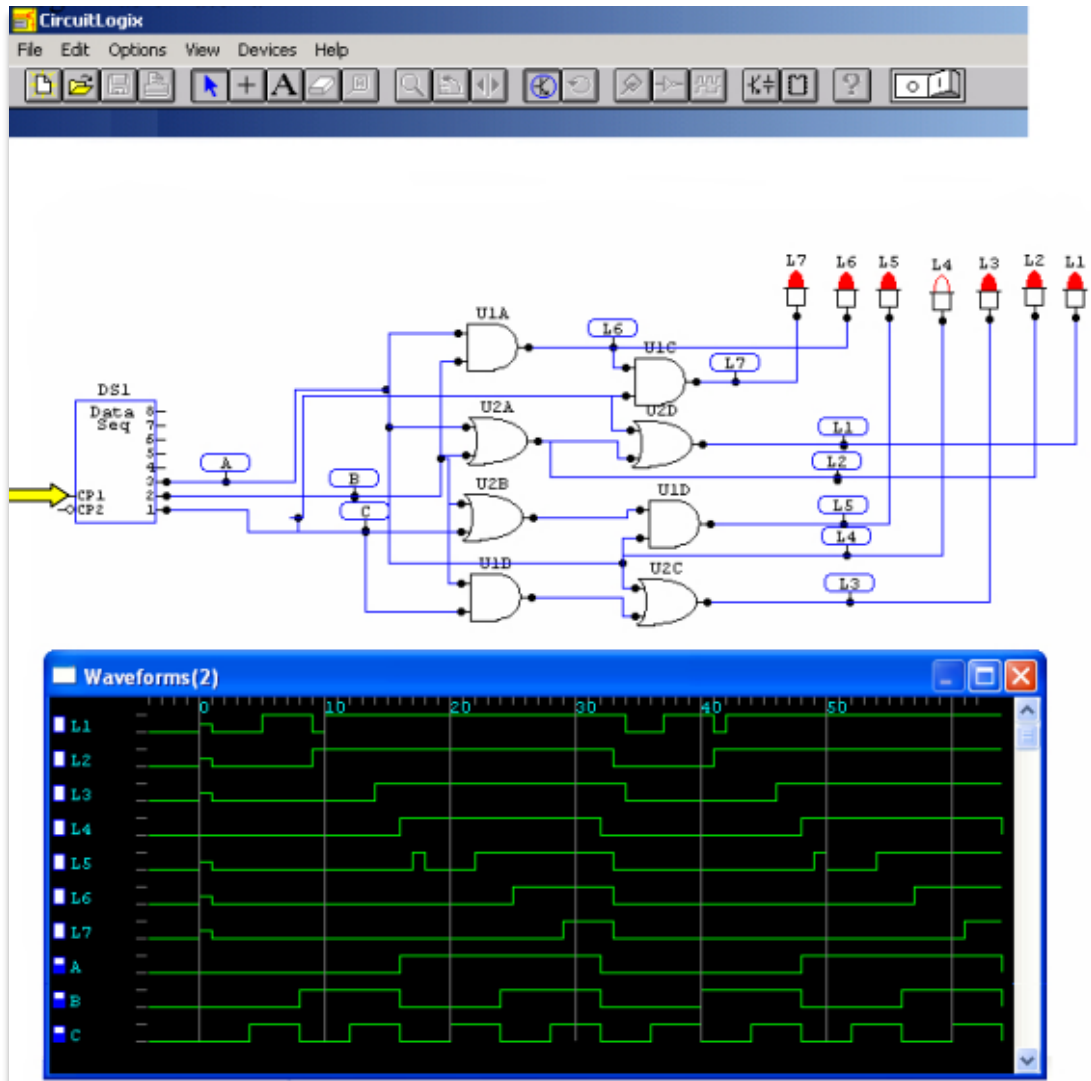


Figure 35 Διεπαφή χρήστη στο CircuitLogix.

2.3.1 Λειτουργικά χαρακτηριστικά

Το CircuitLogix περιέχει όλες τις λογικές πύλες εκτός της XNOR και μπορεί να δεχθεί έως και 2 εισόδους για τις πύλες του. Μπορεί να έχει προσαρμοσμένη διαδρομή στα καλώδια του. Υπάρχουν τα D, RS, JK Flip Flops και μετρητές. Ο χρήστης έχει δυνατότητα προβολής διαγράμματος χρονισμού μέσω του waveforms για το κύκλωμα του. Τα καλώδια μπορούν να πάρουν προσαρμοσμένη διαδρομή καθώς και να αναδείξουν τη τιμή τους εάν ο χρήστης πατήσει επάνω τους. Τέλος, μπορεί να υλοποιήσει λειτουργίες αποθήκευσης και εκτύπωσης για το εκάστοτε κύκλωμα και τη κυματομορφή που έχει σχηματιστεί.

2.4 Logic Gate Simulator

Ο προσομοιωτής Logic gate Simulator [4] είναι ένας δωρεάν open-source προσομοιωτής του έτους 2009, είναι γραμμένος σε C#/WPF χρησιμοποιεί .NET 4 για αυτό και χρειάζεται να είναι εγκαταστημένο στον υπολογιστή του χρήστη .NET Framework. Πρόκειται για ένα προσομοιωτή επιπέδου πύλης με συνολικό μέγεθος 357 kb.

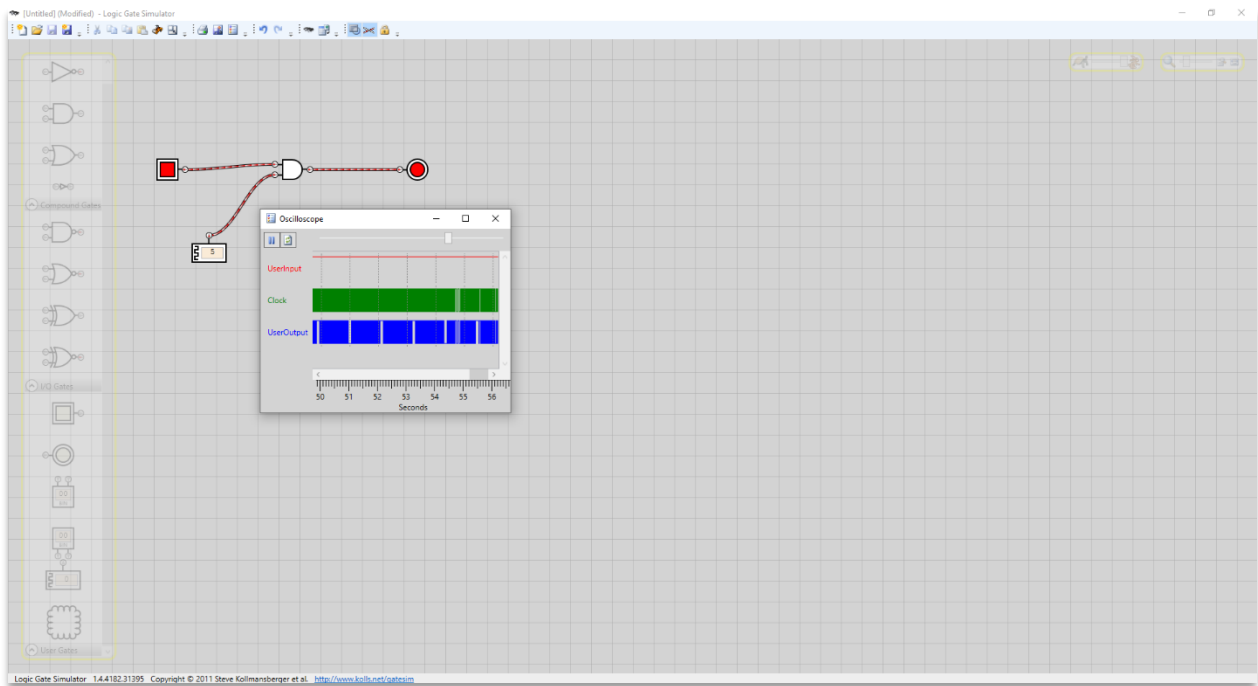


Figure 36 Διεπαφή χρήστη στο Logic Gate Simulator.

2.4.1 Λειτουργικά χαρακτηριστικά

Το Logic gate Simulator περιέχει όλες τις λογικές πύλες καθώς και δεν έχει κάποιο μικρό όριο για τον αριθμό των εισόδων στις πύλες του. Δεν μπορεί να έχει προσαρμοσμένη διαδρομή στα καλώδια του αλλά παρέχει τη δυνατότητα τα καλώδια να αναδείξουν την τιμή τους σε πραγματικό χρόνο όλα μαζί με το κόκκινο χρώμα να περιγράφεται η τιμή 1 ενώ με το λευκό χρώμα η τιμή 0. Δεν υπάρχουν Flip Flop αλλά υπάρχει εξάρτημα clock. Μπορεί να δημιουργήσει ειδικά κυκλώματα σε μορφή συντομεύσεων με τη λειτουργία create IC ενώ μπορεί και να εισάγει στο κύκλωμα έτοιμα IC από αρχεία .ic που υπάρχουν αποθηκευμένα. Υπάρχει δυνατότητα οπτικοποίησης του χρονισμού του κυκλώματος μέσω διαγράμματος χρονισμού. Τέλος, μπορεί να υλοποιήσει αποθήκευση για το κύκλωμα μας.

2.5 CircuitMod

Ο προσομοιωτής CircuitMod [5] είναι ένας δωρεάν προσομοιωτής του έτους 2013, υποστηρίζεται από Windows/Linux, είναι γραμμένος σε Java για αυτό και χρειάζεται να είναι εγκαταστημένο στον υπολογιστή java environment 1.6.0 ή νεότερο. Πρόκειται για ένα προσομοιωτή επιπέδου πύλης και Transistor με συνολικό μέγεθος 3.93 mb.

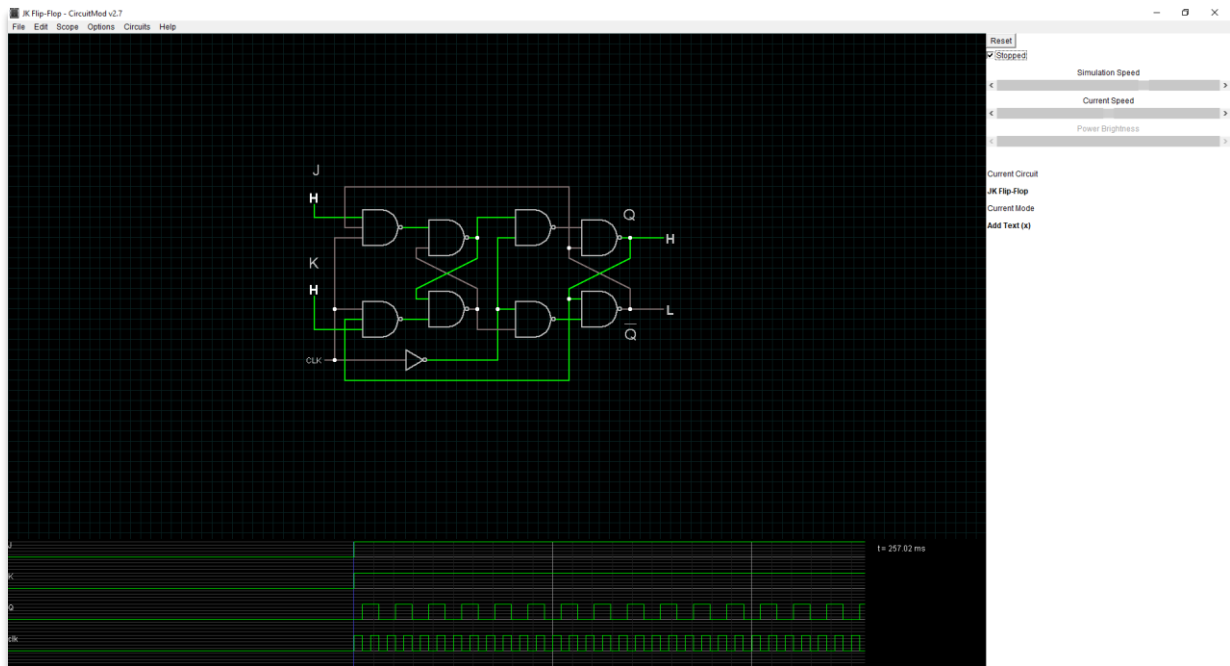


Figure 37 Διεπαφή χρήστη στο CircuitMod.

2.5.1 Λειτουργικά χαρακτηριστικά

Το CircuitMod περιέχει όλες τις λογικές πύλες εκτός της XNOR καθώς και δεν έχει κάποιο μικρό όριο για τον αριθμό των εισόδων στις πύλες του. Μπορεί να έχει προσαρμοσμένη διαδρομή στα καλώδια του και παρέχει τη δυνατότητα τα καλώδια να αναδείξουν την τιμή του σε πραγματικό χρόνο όλα μαζί με το πράσινο χρώμα να περιγράφεται η τιμή 1/H ενώ με το λευκό χρώμα η τιμή 0/L. Υπάρχουν Flip Flops, εξάρτημα clock, πολυπλέκτες και μετρητές. Επίσης, ο προσομοιωτής παρέχει διαγράμματα χρονοισμού σε περίπτωση που υπάρχει CLK εξάρτημα στο κύκλωμα εμφανίζοντας τα σήματα από τις αντίστοιχες ετικέτες του κυκλώματος στο κάτω μέρος τις οθόνης. Τέλος, μπορεί να υλοποιήσει λειτουργίες αποθήκευσης κυκλώματος και άνοιγμα έτοιμων σχεδίων.

2.6 Simulator.io

Ο προσομοιωτής Simulator.io [6] είναι ένας δωρεάν online προσομοιωτής του έτους 2015, υποστηρίζεται από Windows/MacOS/Linux και πρόκειται για ένα προσομοιωτή επιπέδου πύλης.

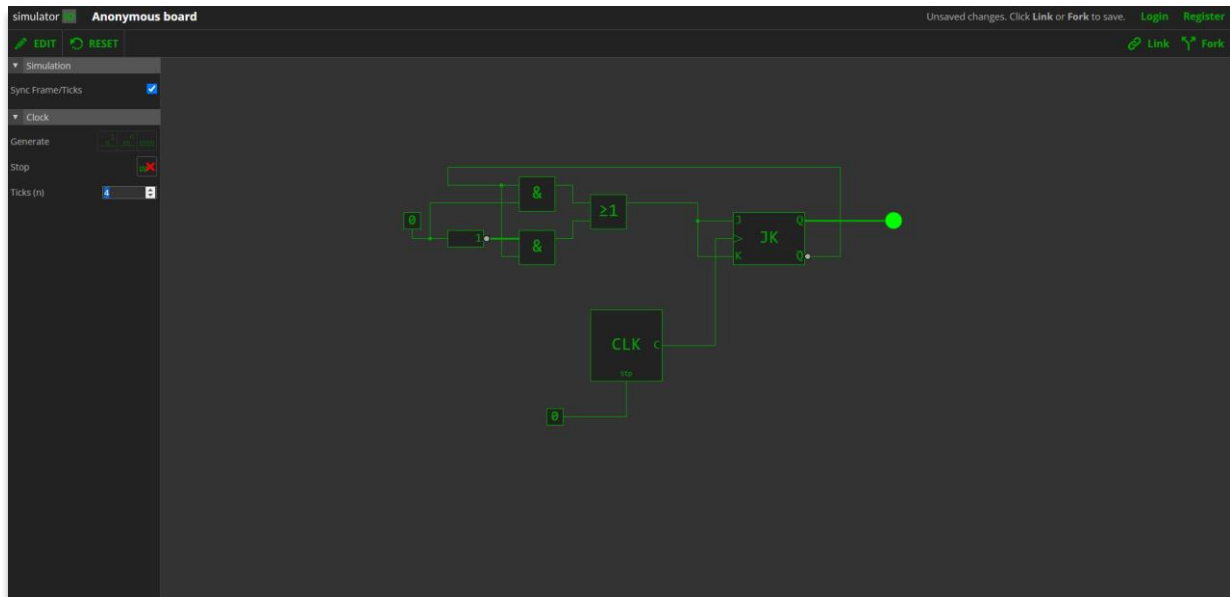


Figure 38 Διεπαφή χρήστη στο Simulator.io.

2.6.1 Λειτουργικά χαρακτηριστικά

Το Simulator.io περιέχει όλες τις λογικές πύλες εκτός της NOR και XNOR και μπορεί να δεχθεί έως και 32 εισόδους για τις πύλες του. Μπορεί να έχει προσαρμοσμένη διαδρομή στα καλώδια του και παρέχει τη δυνατότητα τα καλώδια να αναδείξουν την τιμή του σε πραγματικό χρόνο όλα μαζί με έντονο πράσινο χρώμα να περιγράφεται η τιμή 1 ενώ με το σκούρο πράσινο χρώμα η τιμή 0. Υπάρχουν τα JK και RS Flip Flops, πολυπλέκτες, αποπλέκτες, αποκωδικοποιητές, ημιαθροιστές, αθροιστές και εξάρτημα clock. Τέλος, με τη δημιουργία λογαριασμού δίνεται η δυνατότητα να κοινοποιηθούν τα σχέδια του χρήστη με άλλους χρήστες μέσω της επιλογής Link, στη συνέχεια να δοθεί η δυνατότητα σε άλλους χρήστες να τροποποιήσουν τα σχέδια αυτά να αντικαθιστώντας τα μέσω της επιλογής Fork. Τέλος, ο χρήστης μπορεί να δημιουργήσει ομάδες με άλλους χρήστες και να συμμετάσχει σε ομαδικές συνομιλίες με τα μέλη της ομάδας του.

2.7 Logic Circuit

Ο προσομοιωτής Logic Circuit [7] είναι ένας δωρεάν open-source προσομοιωτής του έτους 2009. Πρόκειται για ένα προσομοιωτή επιπέδου πύλης και behavioral με συνολικό μέγεθος 3.93 mb.

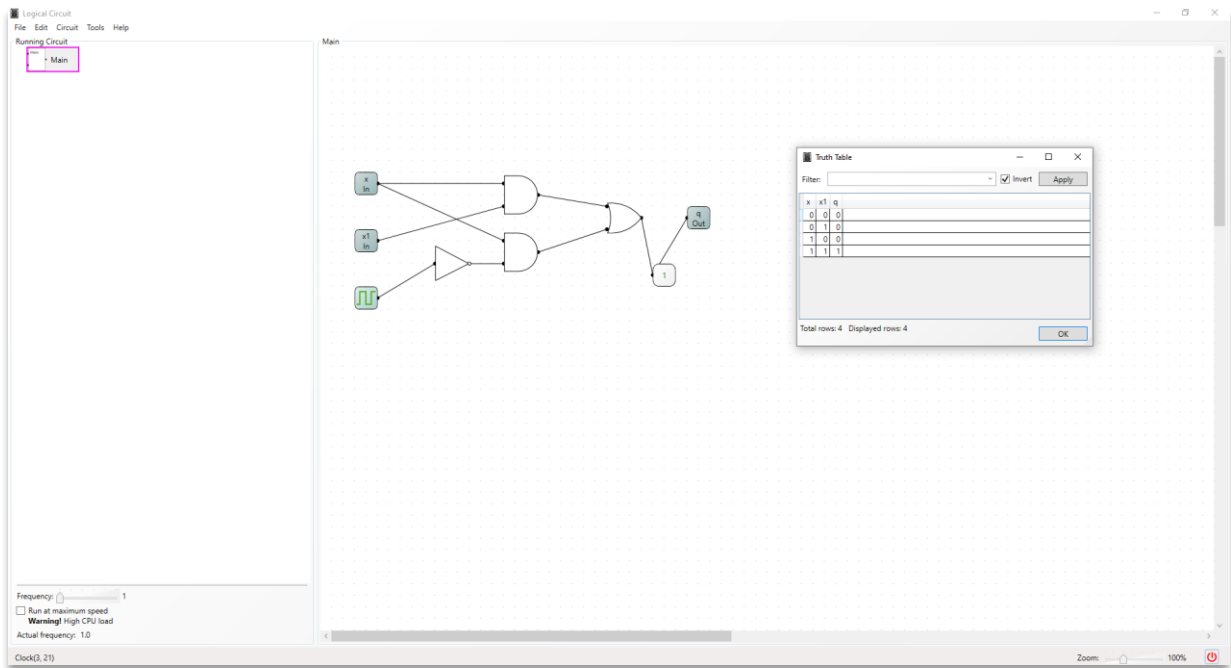


Figure 39 Διεπαφή χρήστη στο Logic Circuit.

2.7.1 Λειτουργικά χαρακτηριστικά

Το Logic Circuit περιέχει όλες τις λογικές πύλες και μπορεί να δεχθεί έως και 18 εισόδους για τις πύλες του. Μπορεί να έχει προσαρμοσμένη διαδρομή στα καλώδια του και παρέχει τη δυνατότητα τα καλώδια να αναδειξουν την τιμή τους σε πραγματικό χρόνο με χρήση του Probe το οποίο μπορεί να αναδείξει τη στιγμιαία τιμή ενός ή περισσότερων καλωδίων απλά με την τοποθέτηση του επάνω στο επιθυμητό σημείο καλωδίου που ο χρήστης θέλει να γίνει η μέτρηση. Ο προσομοιωτής παρέχει στο χρήστη του τη δυνατότητα ανάλυσης του σχεδιασμένου κυκλώματος σε πίνακα αληθείας για τις εισόδους – εξόδους που έχουν τοποθετηθεί. Επίσης, δίνεται η δυνατότητα οπτικοποίησης του χρονισμού του κυκλώματος με τη λειτουργία oscillation. Εκτός από τη δυνατότητα του πίνακα αληθείας για να γίνει κατανοητό το κύκλωμα που έχει δημιουργηθεί ο προσομοιωτής Logic Circuit δίνει την δυνατότητα να δοκιμαστεί το κύκλωμα με scripting μέσω της IronPython Console εισάγοντας επιθυμητές εισόδους στο εκάστοτε κύκλωμα που έχει δημιουργηθεί. Έτσι, για το παραπάνω κύκλωμα (εικόνα 39) εισάγοντας τις ακόλουθες εντολές στην κονσόλα επιτυγχάνεται η είσοδος της τιμής 0 στην είσοδο x και ο χρήστης λαμβάνει την τιμή 0 ως αποτέλεσμα της πράξης αυτής με τα αποτελέσματα εξόδου να μπορούν να δεχθούν τις τιμές 0 για ανεξάρτητη κατάσταση, 1 για κατάσταση 0 και 2 για κατάσταση 1.

```
IronPython Console
IronPython 2.7.11
>>> print "hello, world!"
hello, world!
>>> tester = App.CreateTester("Main")
>>> tester.SetInput("x", 0)
>>> tester.Evaluate()
True
>>> tester.GetOutput("q")
0
>>>
```

Figure 40 Κονσόλα χρήσης IronPython.

Τέλος, το Logic Circuit μπορεί να εισάγει κυκλώματα μέσω αρχείου, να τα εκτυπώσει, να τα αποθηκεύσει αλλά και να δημιουργήσει πολλαπλά στρώματα από κυκλώματα για την σχεδίαση πολύπλοκων κυκλωμάτων.

2.8 CEDAR Logic

Ο προσομοιωτής CEDAR Logic [8] είναι ένας δωρεάν προσομοιωτής του έτους 2007, είναι γραμμένος σε C++ και περιέχει low-level λογικές πύλες καθώς και high-level εξαρτήματα. Πρόκειται για ένα προσομοιωτή επιπέδου πύλης και Transistor με συνολικό μέγεθος 10.1 mb.

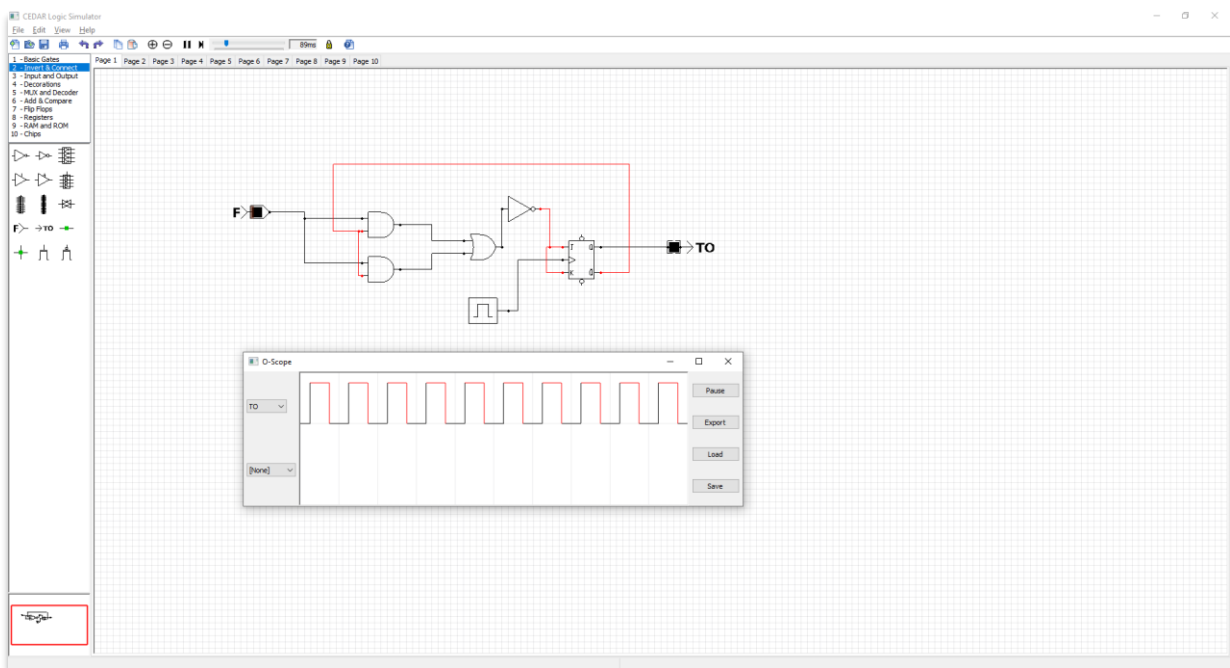


Figure 41 Διεπαφή χρήστη στο CEDAR Logic.

2.8.1 Λειτουργικά χαρακτηριστικά

Το CEDAR Logic περιέχει όλες τις λογικές πύλες μαζί και τις NORX, NANDX και μπορεί να δεχθεί έως και 8 εισόδους για τις πύλες του. Μπορεί να έχει προσαρμοσμένη διαδρομή στα καλώδια του και παρέχει τη δυνατότητα τα καλώδια να αναδείξουν την τιμή του σε πραγματικό χρόνο όλα μαζί με κόκκινο χρώμα να περιγράφεται η τιμή 1 ενώ με το μαύρο

χρώμα η τιμή 0. Υπάρχουν τα JK και D Flip Flops, πολυπλέκτες, αποκωδικοποιητές, αθροιστές και εξάρτημα clock. Επίσης, δίνεται στο χρήστη η δυνατότητα οπτικοποίησης του χρονισμού του κυκλώματος μέσω του o-score. Τέλος, μπορεί να αποθηκεύσει τα κυκλώματα, να ανοίξει έτοιμα σχέδια και να τα εκτυπώσει.

2.9 Multisim Live

Ο προσομοιωτής Multisim Live [9] είναι ένας δωρεάν online προσομοιωτής, υποστηρίζεται από Windows\MacOS\Linux και πρόκειται για ένα προσομοιωτή επιπέδου πύλης και Transistor. Παρέχεται στο χρήστη η δυνατότητα μέσω του Multisim Live να δημοσιεύσει τα σχέδια του στην επίσημη ιστοσελίδα του προσομοιωτή καθώς και τις μετρήσεις του μοιράζοντας πληροφορίες στο διαδίκτυο με άλλους χρήστες βοηθώντας να εντοπιστούν ενδεχομένως λάθη που δεν μπορεί να διακρίνει ο ίδιος αλλά και τυχόν βελτιώσεις του κυκλώματος του.

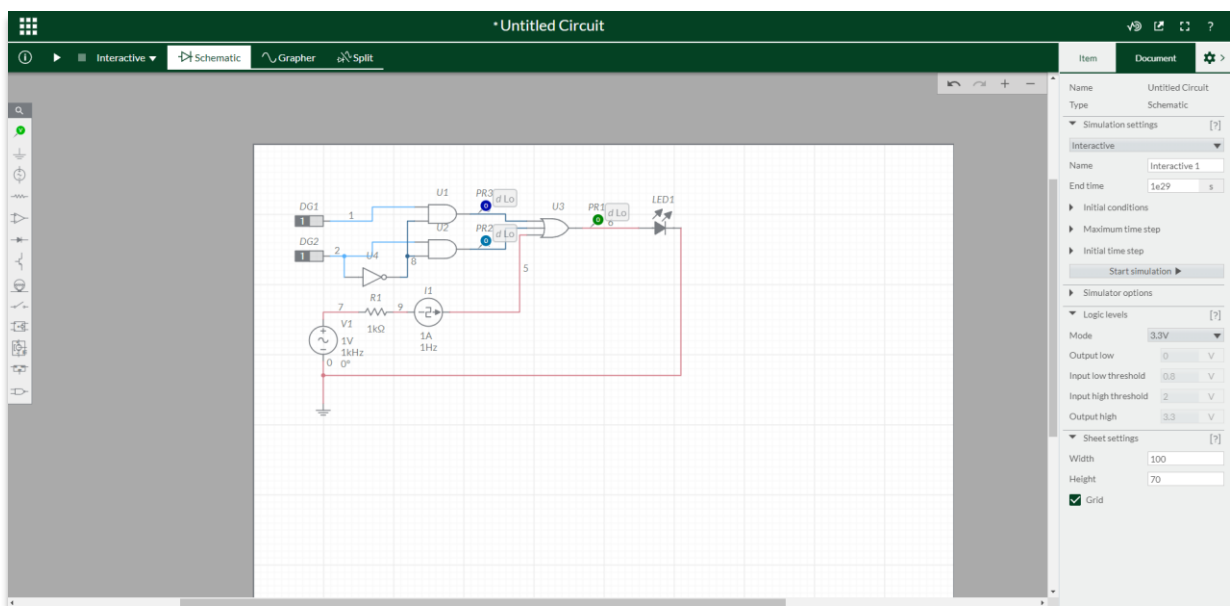


Figure 42 Διεπαφή χρήστη στο Multisim Live.

2.9.1 Λειτουργικά χαρακτηριστικά

Το Multisim Live περιέχει όλες τις λογικές πύλες και μπορεί να δεχθεί έως και 8 εισόδους για τις πύλες του. Μπορεί να έχει προσαρμοσμένη διαδρομή στα καλώδια του και παρέχει τη δυνατότητα τα καλώδια να αναδείξουν την τιμή του σε πραγματικό χρόνο όλα μαζί. Υπάρχει εξάρτημα clock το οποίο θα πρέπει να συνδεθεί με ρεύμα. Επίσης, υπάρχει η δυνατότητα οπτικοποίησης του χρονισμού του κυκλώματος μέσω του Grapher αφού έχουν τοποθετηθεί στα κατάλληλα σημεία probes.

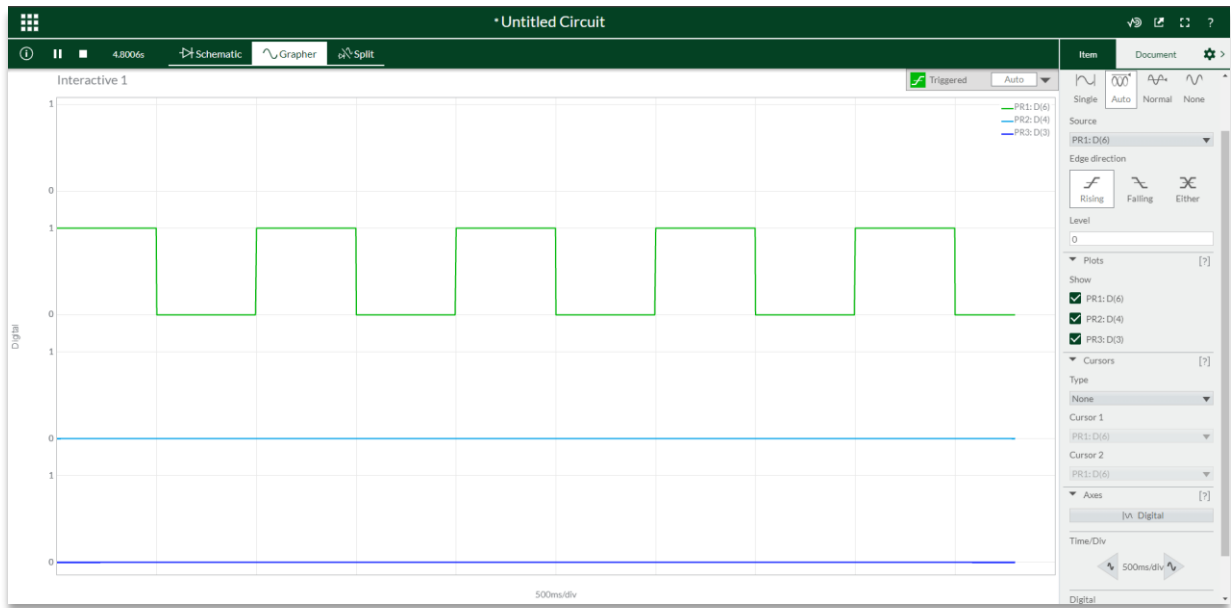


Figure 43 Διάγραμμα χρονισμού σε Multisim Live.

Με την εκκίνηση της λειτουργίας του κυκλώματος εκτελείται αυτόματος έλεγχος σφαλμάτων και ο χρήστης ενημερώνεται σε περίπτωση που εντοπίσει κάποιο.

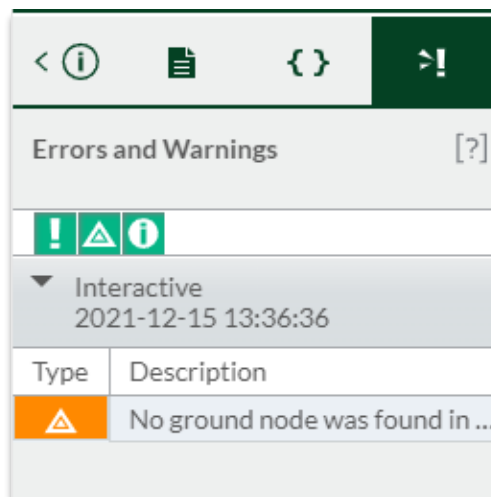


Figure 44 Έλεγχος σφαλμάτων στο Multisim Live

Τέλος, μπορεί να αποθηκεύσει τα σχέδια, να ανοίξει έτοιμα και να αποθηκεύσει σε .csv αρχείο όλες της μετρήσεις που πάρθηκαν στο simulation.

2.10 CircuitVerse

Ο προσομοιωτής CircuitVerse [10] είναι ένας δωρεάν online προσομοιωτής, υποστηρίζεται από Windows/MacOS/Linux και πρόκειται για ένα προσομοιωτή επιπέδου πύλης, behavioral και functional. Παρέχετε στο χρήστη η δυνατότητα μέσω του CircuitVerse να δημοσιεύσουμε τα σχέδια μας online βοηθώντας με αυτό το τρόπο να εντοπιστούν ενδεχομένως λάθη ή βελτιώσεις του εκάστοτε κυκλώματος.

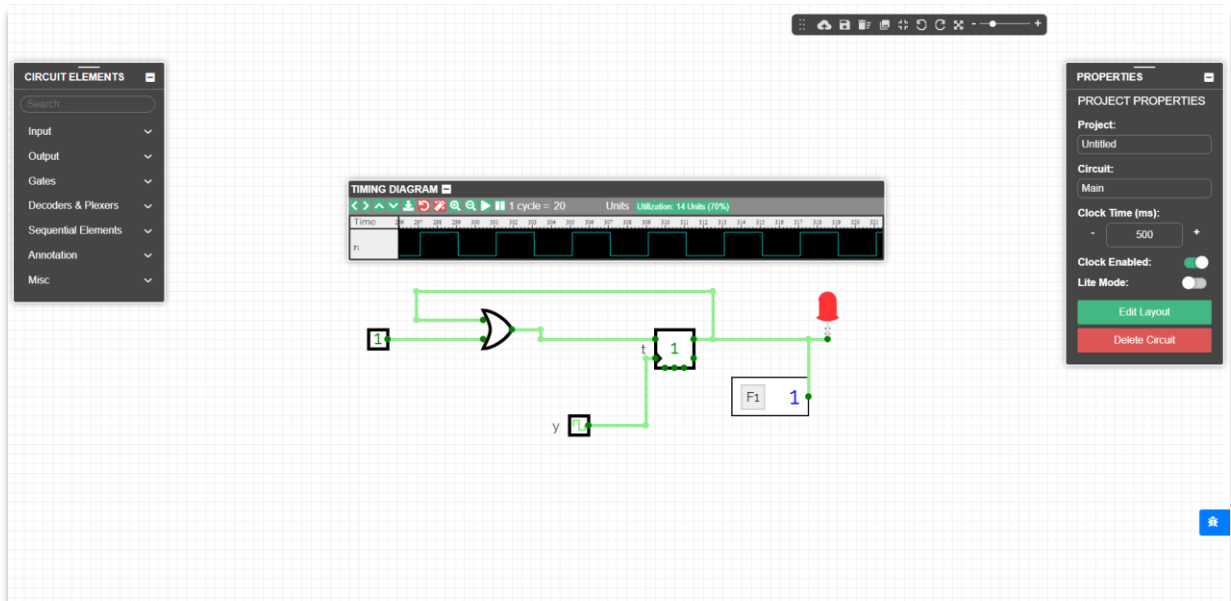


Figure 45 Διεπαφή χρήστη στο CircuitVerse.

2.10.1 Λειτουργικά χαρακτηριστικά

Το CircuitVerse περιέχει όλες τις λογικές πύλες και μπορεί να δεχθεί έως και 10 εισόδους για τις πύλες του. Μπορεί να έχει προσαρμοσμένη διαδρομή στα καλώδια του και παρέχει τη δυνατότητα τα καλώδια να αναδείξουν την τιμή τους σε πραγματικό χρόνο όλα μαζί. Υπάρχουν τα D, SR, T, JK Flip Flops, πολυπλέκτες, αποπλέκτες, αποκωδικοποιητές, μετρητές και εξάρτημα clock. Επίσης, δίνεται στο χρήστη η δυνατότητα οπτικοποίησης του χρονισμού του κυκλώματος μέσω του Timing Diagram αφού έχει βάλει στα κατάλληλα σημεία flags. Το CircuitVerse δίνει την δυνατότητα να δημιουργηθούν κυκλώματα από πίνακες αληθείας και από λογικές εκφράσεις με την επιλογή του combinational analysis.

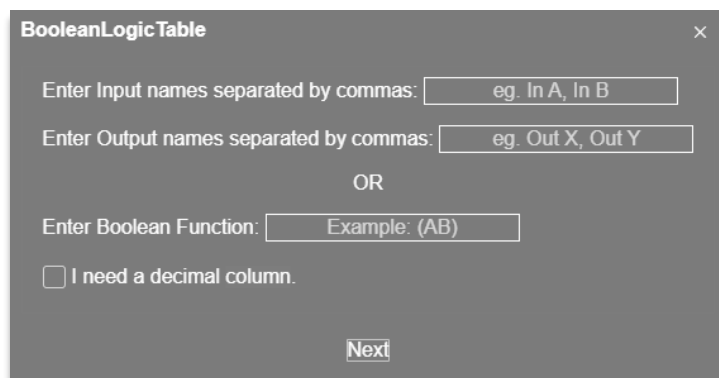


Figure 46 Επιλογές δημιουργίας κυκλωμάτων μέσω εισόδων, εξόδων και λογικών συναρτήσεων.

Επίσης, μπορεί να εξάγει ολόκληρο το κύκλωμα σε Verilog. Τέλος, μπορεί να αποθηκεύσει τα σχέδια του χρήστη, να τα δημοσιεύσει στην επίσημη ιστοσελίδα του προσομοιωτή, να ανοίξει έτοιμα κυκλώματα και να αποθηκεύσει το κύκλωμα σε εικόνα.

2.11 Logic Friday

Ο προσομοιωτής Logic Friday [11] είναι ένας windows based δωρεάν προσομοιωτής του έτους 2013. Πρόκειται για ένα προσομοιωτή επιπέδου πύλης, functional και behavioral με συνολικό μέγεθος 2.79mb.

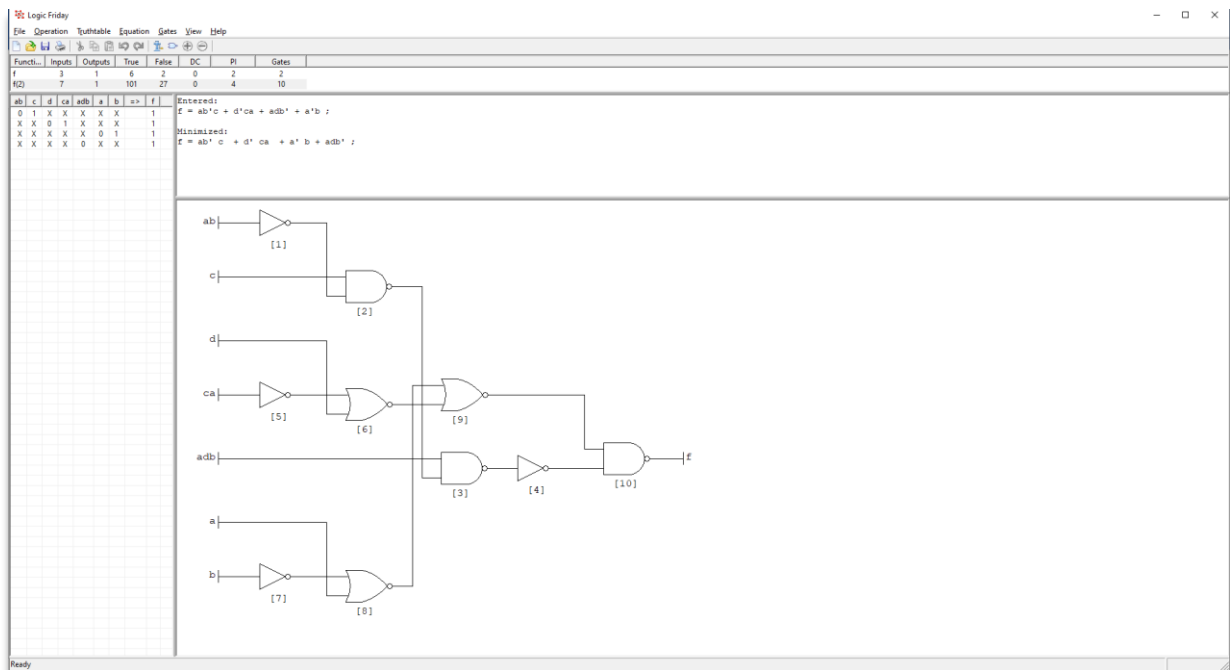


Figure 47 Διεπαφή χρήστη στο Logic Friday.

2.11.1 Λειτουργικά χαρακτηριστικά

Το Logic Friday περιέχει όλες τις λογικές πύλες εκτός της XNOR και μπορεί να δεχθεί έως και 16 εισόδους για τις πύλες του. Μπορεί να έχει προσαρμοσμένη διαδρομή στα καλώδια του. Υπάρχουν μόνο πολυπλέκτες και δίνεται η δυνατότητα να δημιουργηθεί ένα κύκλωμα από λογικές πύλες, πίνακα αληθείας και από λογική συνάρτηση. Επίσης, μπορεί να απλοποιήσει το όποιο κύκλωμα του χρήστη με όποιον τρόπο και αν το έχει δημιουργήσει βάζοντας ως επιλογή συγκεκριμένες πύλες που θα ήθελε να αποτελέσει το τελικό απλοποιημένο κύκλωμα. Το Logic Friday μπορεί να μετατρέψει το κύκλωμα σε κώδικα C και να το εξάγει σε αρχείο .txt καθώς και τα δεδομένα του πίνακα αληθείας σε .csv αρχείο. Τέλος, μπορεί να αποθηκεύσει τα σχέδια, να ανοίξει έτοιμα κυκλώματα ή πίνακες αληθείας, να εξάγει πίνακες αληθείας και να εκτυπώσει το κύκλωμα που σχεδιάστηκε.

```
lookup.c - Notepad
File Edit Format View Help
/*
 *lookup.c
 *Generated by Logic Friday on Fri Dec 17 14:13:17 2021
 */
int lu_f( int ab, int c, int d, int ca, int adb, int a, int b );
int lu_f( int ab, int c, int d, int ca, int adb, int a, int b )
{
    const unsigned int nOutAry[] =
    {0XF4FFF4F4, 0XFFFFFFF, 0XF4FFF4F4, 0XF4FFF4F4};

    unsigned int nTerm=0, nX, nBit;

    if( ab ) nTerm |= 1<<6;
    if( c ) nTerm |= 1<<5;
    if( d ) nTerm |= 1<<4;
    if( ca ) nTerm |= 1<<3;
    if( adb ) nTerm |= 1<<2;
    if( a ) nTerm |= 1<<1;
    if( b ) nTerm |= 1;
    nX = nTerm / 32;
    nBit = 31 - nTerm % 32;
    if( nOutAry[nX] & 1<<nBit )
        return 1;
    else
        return 0;
}
```

Figure 48 Παραγόμενο αρχείο .C από κύκλωμα.

2.12 Logic.ly

Ο προσομοιωτής Logic.ly [12] είναι ένας δωρεάν online προσομοιωτής του 2008, υποστηρίζεται από Windows\MacOS\Linux και πρόκειται για ένα προσομοιωτή επιπέδου πύλης και behavioral. Οι καταστάσεις που μπορεί το κύκλωμα να βρεθεί περιγράφονται από τα χρώματα μπλε (high>true\1), λευκό (low/false\0), γκρι (high impedance state) και κόκκινο (error state).

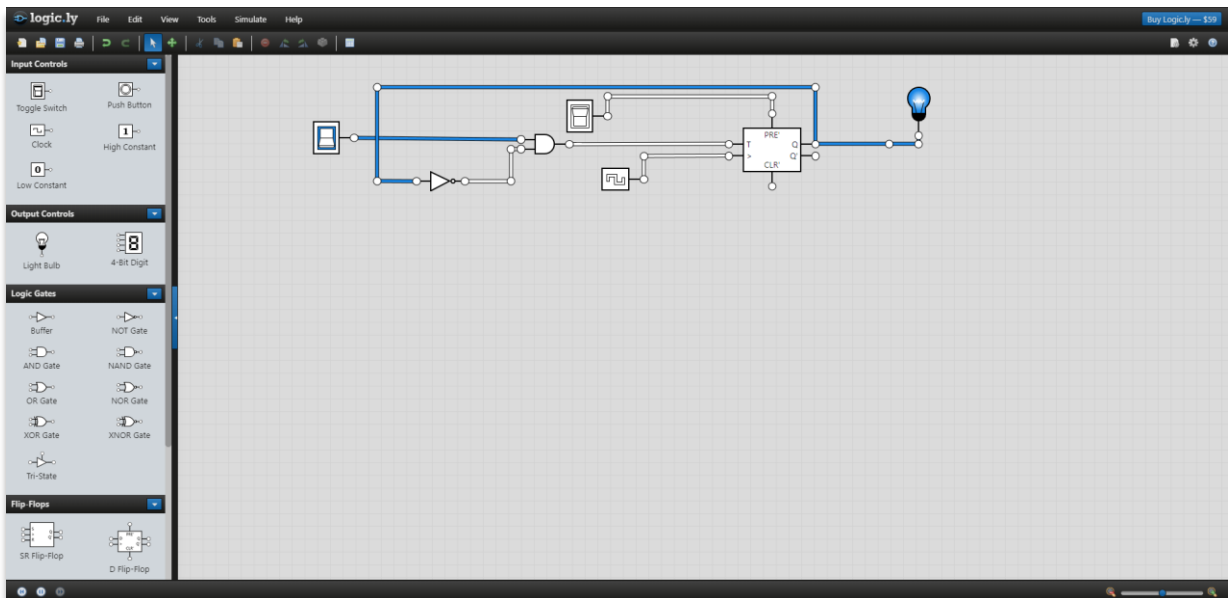


Figure 49 Διεπαφή χρήστη στο Logic.ly.

2.12.1 Λειτουργικά χαρακτηριστικά

Το Logic.ly περιέχει όλες τις λογικές πύλες και μπορεί να δεχθεί έως και 8 εισόδους για τις πύλες του. Μπορεί να έχει προσαρμοσμένη διαδρομή στα καλώδια του και παρέχει τη

δυνατότητα τα καλώδια να αναδειξουν την τιμή του σε πραγματικό χρόνο όλα μαζί. Υπάρχουν τα D, SR, T, JK Flip Flops και εξάρτημα clock. Το Logic.ly δίνει την δυνατότητα να δημιουργηθούν κυκλώματα από πίνακες αληθείας και από τα ολοκληρωμένα κυκλώματα που ήδη υπάρχουν. Τέλος, μπορεί να αποθηκεύσει τα σχέδια, να ανοίξει έτοιμα κυκλώματα και να αποθηκεύσει το κύκλωμα σε εικόνα.

2.13 BOOLR

Ο προσομοιωτής BOOLR [13] είναι ένας δωρεάν open source προσομοιωτής του 2017 που δημιουργήθηκε από Ολλανδούς μαθητές, υποστηρίζεται από Windows\MacOS\Linux, είναι γραμμένος σε JavaScript, HTML5, CSS3 και πρόκειται για ένα προσομοιωτή επιπέδου πύλης συνολικό μέγεθος 129mb.

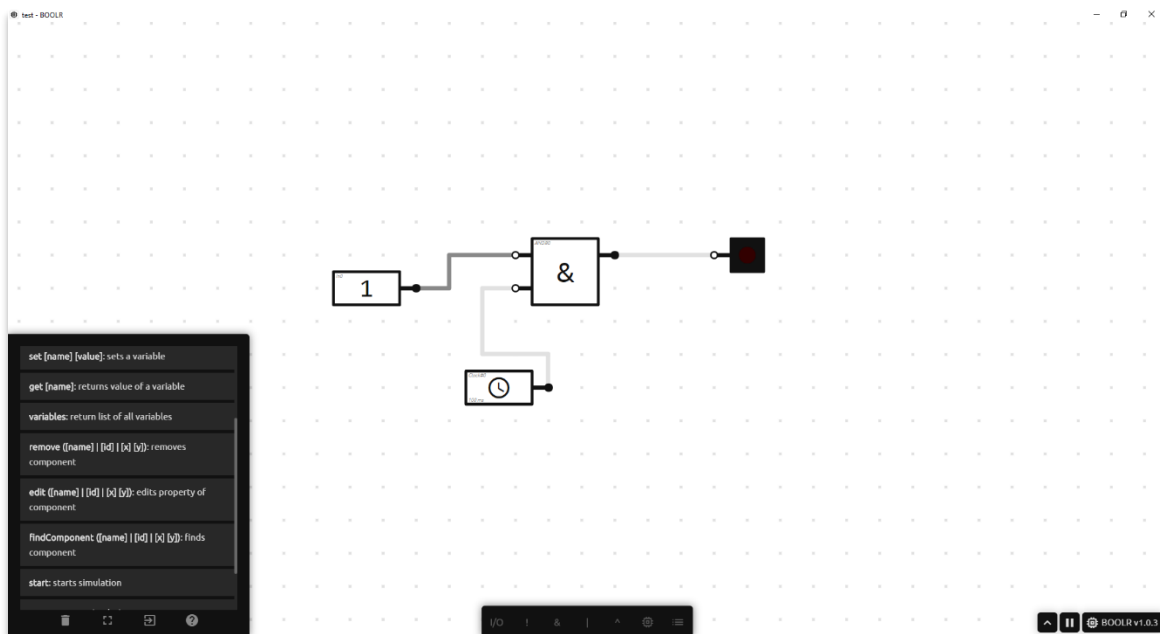


Figure 50 Διεπαφή χρήστη στο BOOLR.

2.13.1 Λειτουργικά χαρακτηριστικά

Το BOOLR περιέχει όλες τις λογικές πύλες εκτός των Nand, Xnor και μπορεί να δεχθεί έως και 2 εισόδους για τις πύλες του. Μπορεί να έχει προσαρμοσμένη διαδρομή στα καλώδια του και παρέχει τη δυνατότητα τα καλώδια να αναδειξουν την τιμή του σε πραγματικό χρόνο όλα μαζί. Υπάρχουν εξάρτημα clock και μετρητή. Το BOOLR δίνει την δυνατότητα να τροποποιηθεί και να ενημερωθεί το κύκλωμα μέσω κονσόλας. Τέλος, μπορεί να αποθηκεύσει τα σχέδια και να ανοίξει έτοιμα κυκλώματα.

2.14 Deeds

Ο προσομοιωτής Deeds [14] είναι ένας δωρεάν προσομοιωτής του 2002 που δημιουργήθηκε από Ιταλούς φοιτητές, υποστηρίζεται από Windows\MacOS\Linux, και πρόκειται για ένα προσομοιωτή επιπέδου πύλης συνολικό μέγεθος 66.8mb. Με την εγκατάσταση του Deeds παρατηρούνται 3 υποπρογράμματα το Deeds-DcC (Digital Circuit Simulator) που αφορά την σχεδίαση ψηφιακών κυκλωμάτων, το Deeds- FsM (Finite State Machine Simulator) που αφορά την σχεδίαση καταστάσεων και το Deed- McE (Micro

Computer Emulator) που αφορά την σχεδίαση micro computers με 8-bit CPU, ROM, RAM και I/O ports.

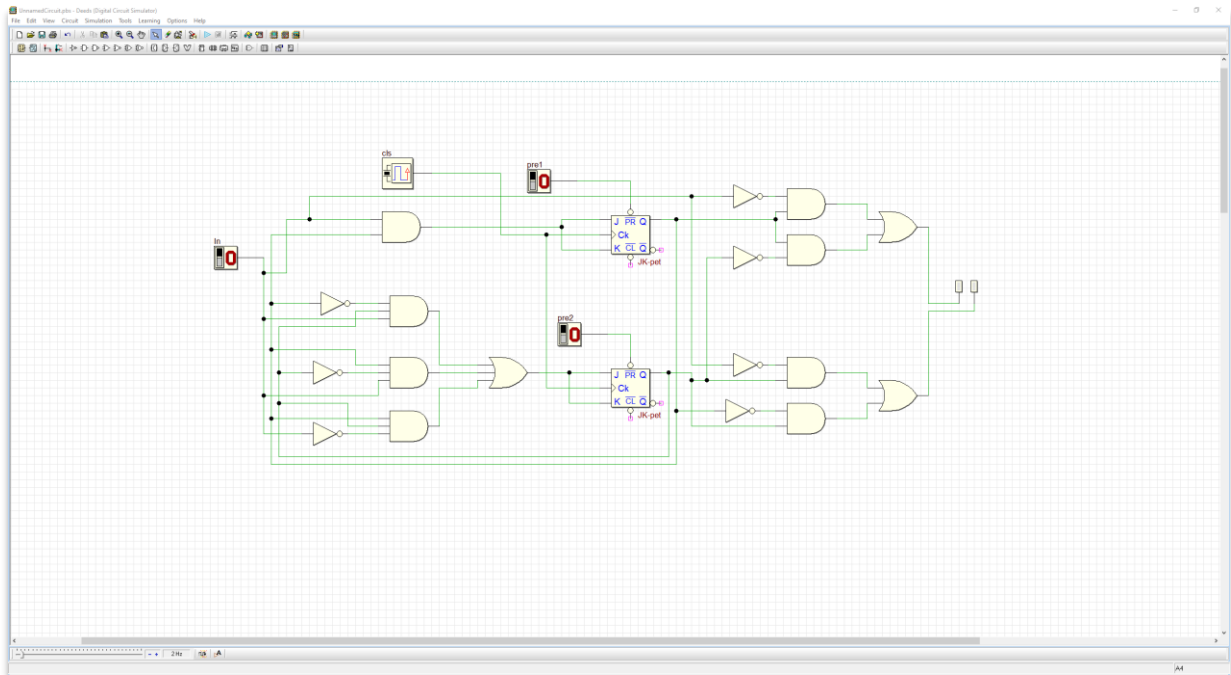


Figure 51 Διεπαφή χρήστη στο Deeds.

2.14.1 Λειτουργικά χαρακτηριστικά

Το Deeds περιέχει όλες τις λογικές πύλες και μπορεί να δεχθεί έως και 8 εισόδους για τις πύλες του. Μπορεί να έχει προσαρμοσμένη διαδρομή στα καλώδια του. Υπάρχουν τα D, RS, E, JK Flip Flops, πολυπλέκτες, αποπλέκτες, αποκωδικοποιητές, μετρητές, πολλαπλασιαστές και εξάρτημα clock εξάρτημα clock. Το Deeds δίνει την δυνατότητα να για πλήρη έλεγχο του χρονισμού μέσω timing diagram. Υπάρχει και η λειτουργία του error check που γίνεται έλεγχος για τυχόν λάθη στο κύκλωμα αλλά γίνεται και έλεγχος σε πραγματικό χρόνο στο simulation με τα κατάλληλα ενημερωτικά μηνύματα σφαλμάτων να εκτυπώνονται στην κονσόλα στο κάτω μέρος της οθόνης. Γίνεται να εκτελεστούν τα κυκλώματα σε FPGA που παρέχονται και γίνεται να μετατραπεί το εκάστοτε κύκλωμα σε VHDL. Τέλος, μπορεί να αποθηκεύσει τα σχέδια, να ανοίξει έτοιμα κυκλώματα και να αποθηκεύσει το κύκλωμα σε εικόνα αλλά και να την εκτυπώσει. Ενώ στο Deeds- FsM (Finite State Machine Simulator) υπάρχει η δυνατότητα να σχεδιαστούν FSMs μοντέλου moore ή mealy με τους αντίστοιχους πίνακες αληθείας τους.

2.15 Logisim

Ο προσομοιωτής Logisim [15] είναι ένας δωρεάν προσομοιωτής του 2001, υποστηρίζεται από Windows\MacOS\Linux και πρόκειται για ένα προσομοιωτή επιπέδου πύλης, behavioral και functional. Οι καταστάσεις που μπορεί το κύκλωμα να βρεθεί περιγράφονται από τα χρώματα γκρι (άγνωστη κατάσταση), μπλε (high impedance), σκούρο πράσινο (0), ανοικτό πράσινο (1), μαύρο (multi bit value), κόκκινο (σφάλμα) και πορτοκαλί (δεν μεταφέρει τιμές μεταξύ των εξαρτημάτων). Τέλος, έχει συνολικό μέγεθος 6.64mb.

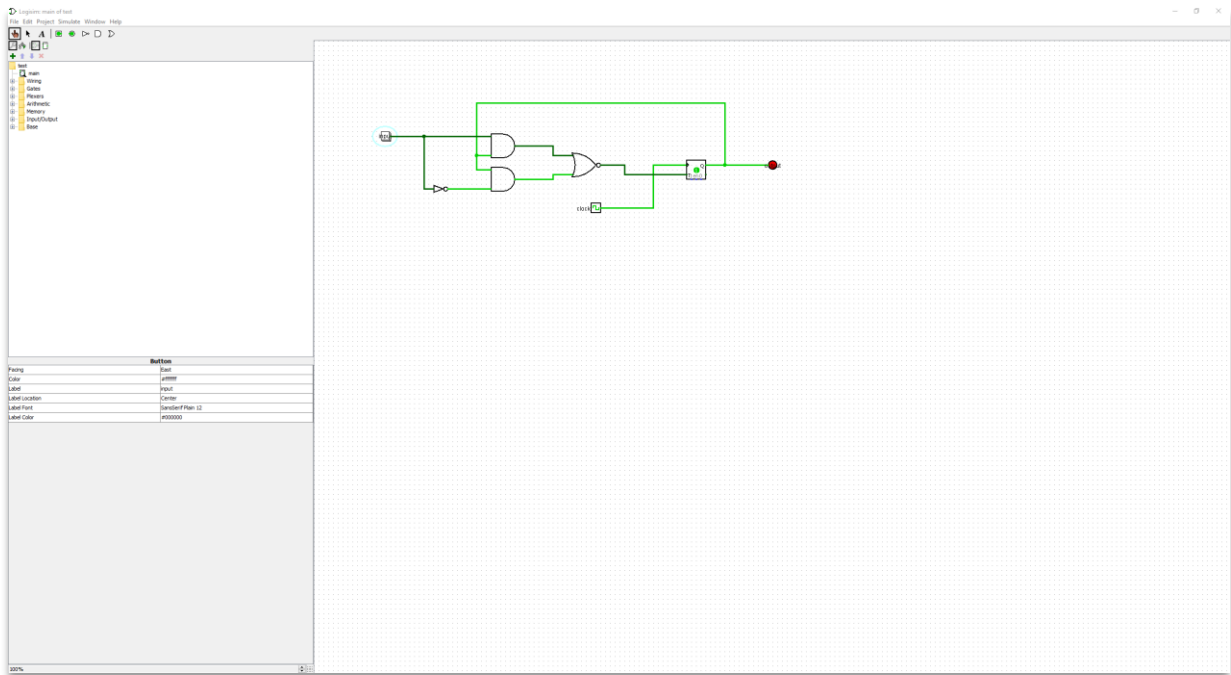


Figure 52 Διεπαφή χρήστη στο Logisim.

2.15.1 Λειτουργικά χαρακτηριστικά

Το Logisim περιέχει όλες τις λογικές πύλες και μπορεί να δεχθεί έως και 32 εισόδους για τις πύλες του. Μπορεί να έχει προσαρμοσμένη διαδρομή στα καλώδια του. Υπάρχουν τα D, RS, T, JK Flip Flops, πολυπλέκτες, αποπλέκτες, αποκωδικοποιητές, μετρητές (πρόσθεσης, αφαίρεσης, πολλαπλασιασμού, διαίρεσης, μετατόπισης, πρόσθεσης bit και εύρεσης bit) και εξάρτημα clock. Ο χρήστης μπορεί να έχει ανάλυση σε λογική συνάρτηση απλοποιημένη ή μη, πίνακα αληθείας αλλά και logs από όλες τις εισόδους και όλες τις εξόδους που δοκιμάστηκαν. Ο προσομοιωτής Logisim παρέχει τη δυνατότητα να τροποποιηθούν οι προτιμήσεις τις εφαρμογής με τη σύνταξη “java -jar jarFileName [options] [filename] ” με υποστηριζόμενες παραμέτρους για την προηγούμενη εντολή τις -plain, -empty, -template, -gates, -locale, -accents[yes|no], -clearprops, -nosplash, -help και -version. Τέλος, μπορεί να αποθηκεύσει τα σχέδια, να ανοίξει έτοιμα κύκλωμα και να αποθηκεύσει το κύκλωμα σε εικόνα αλλά και να την εκτυπώσει.

2.16 Hades

Ο προσομοιωτής Hades [16] είναι ένας δωρεάν προσομοιωτής του 1997, υποστηρίζεται από Windows/Linux και πρόκειται για ένα προσομοιωτή επιπέδου πύλης που είναι γραμμένος σε Java και για αυτό θα πρέπει ο χρήστης να έχει εγκαταστημένη την 1.4 ή νεότερη έκδοση της Java εγκαταστημένη στον υπολογιστή του. Οι καταστάσεις που μπορεί το κύκλωμα να βρεθεί περιγράφονται από τα χρώματα γκρι (λογικό 0), κόκκινο (λογικό 1), κίτρινο (tri-state), γαλάζιο (απροσδιόριστο), μωβ (άκυρο ή άγνωστο). Μπορεί να τρέξει σαν applet σε οποιοδήποτε browser έχει ενεργοποιημένο jdk/ jre 1.4.2 ή νεότερο, αυτό προτείνεται για να αναδείξει αποθηκευμένα σχέδια καθώς οι περισσότερες λειτουργίες του προσομοιωτή είναι απενεργοποιημένες στην applet έκδοση. Εάν δεν λειτουργεί ο προσομοιωτής μετά την

εγκατάσταση, με χρήση της εντολής `java -Xmx256M -jar hades.jar` σε κάποιο shell στο κατάλληλο directory που είναι αποθηκευμένο το `hades.jar` επιτυγχάνεται η εκκίνηση της λειτουργίας του προσομοιωτή. Τέλος, έχει συνολικό μέγεθος 4.7mb.

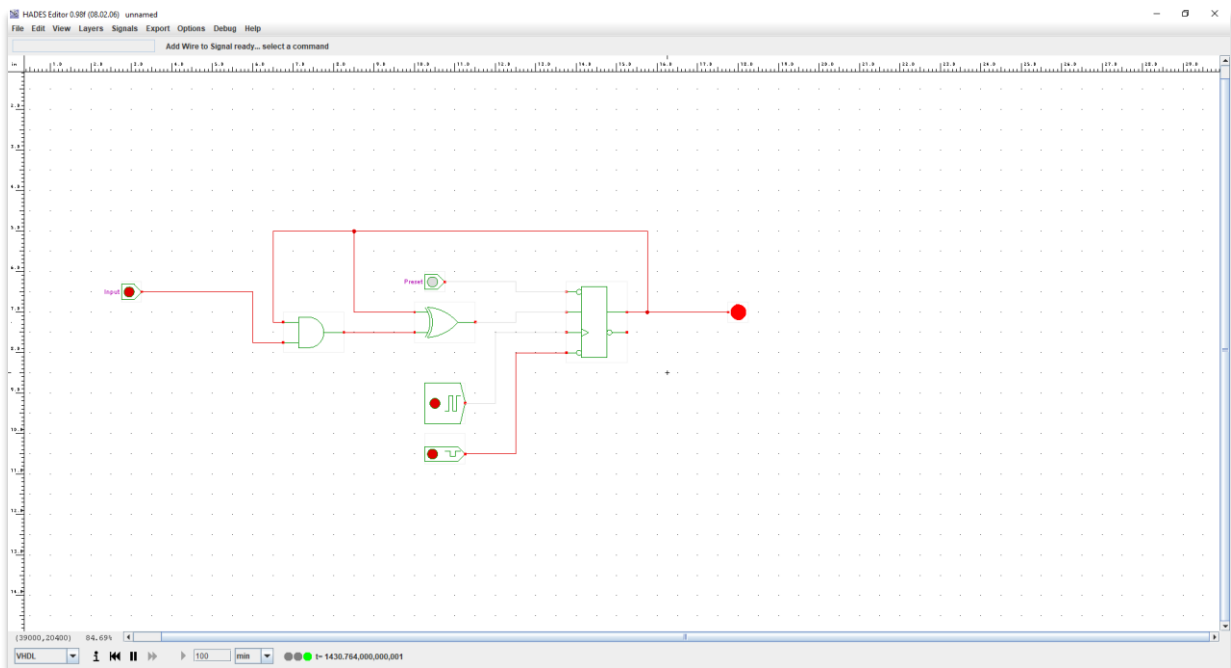


Figure 53 Διεπαφή χρήστη στο Hades.

2.16.1 Λειτουργικά χαρακτηριστικά

Το Hades περιέχει όλες τις λογικές πύλες και μπορεί να δεχθεί έως και 4 εισόδους για τις πύλες του. Μπορεί να έχει προσαρμοσμένη διαδρομή στα καλώδια του. Υπάρχουν τα D, JK Flip Flops, πολυπλέκτες, αποπλέκτες, μετρητές και εξάρτημα clock. Ο χρήστης μπορεί να έχει ανάλυση σε logs από όλες τις εισόδους και όλες τις εξόδους που δοκιμάστηκαν και error check για ολόκληρο το κύκλωμα του.

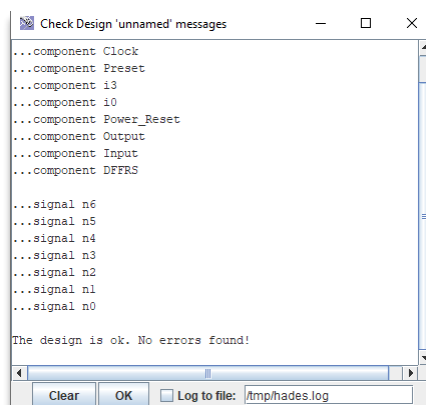


Figure 54 Έλεγχος λαθών στο Hades.

Υπάρχει δυνατότητα ανάλυση του χρονισμού του κυκλώματος μέσω του `show waves`.

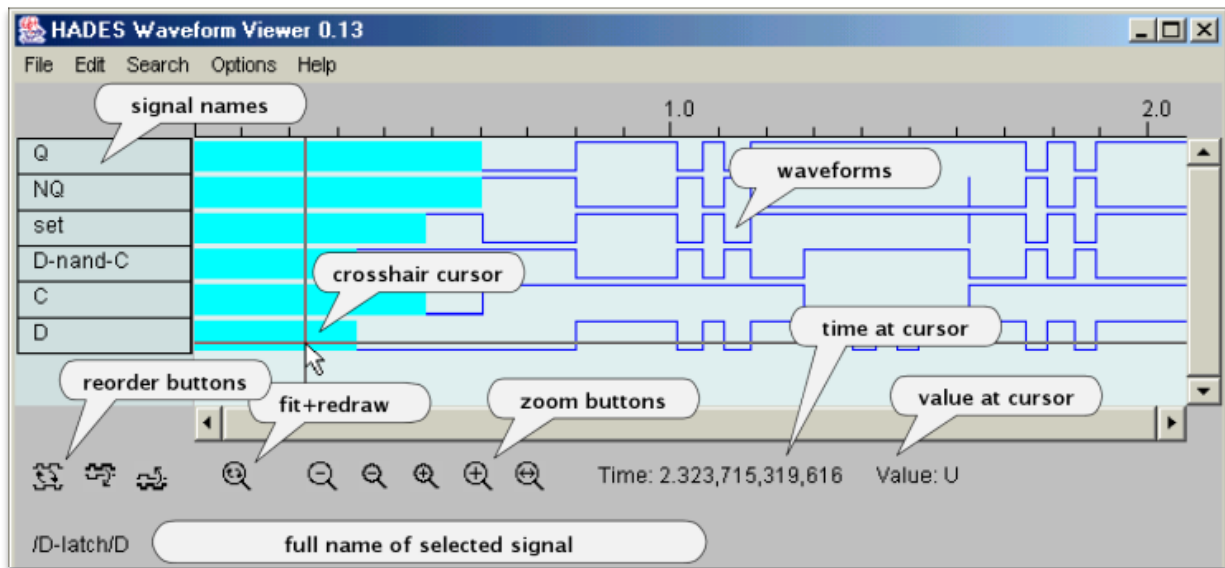


Figure 55 Διάγραμμα χρονισμού στο Hades.

Γίνεται να τροποποιηθεί το main() πρόγραμμα εισάγοντας java classes που θα μπορούν να ικανοποιήσουν διάφορες λειτουργίες. Επίσης, ο χρήστης μπορεί να έχει τη δυνατότητα scripting μέσω της Jython (Java based version of Python) η οποία έχει απλή σύνταξη, πολλές βιβλιοθήκες και πολύ καλή απόδοση. Τέλος, μπορεί να αποθηκεύσει τα σχέδια, να ανοίξει έτοιμα κυκλώματα μέσω αρχείου ή URL, να αποθηκεύσει το κύκλωμα σε εικόνα αλλά και να την εκτυπώσει.

2.17 Digital Logic Design

Ο προσομοιωτής Digital Logic Design [17] είναι ένας δωρεάν προσομοιωτής του 2014, υποστηρίζεται από Windows και πρόκειται για ένα προσομοιωτή επιπέδου πύλης με συνολικό μέγεθος 3.43mb.

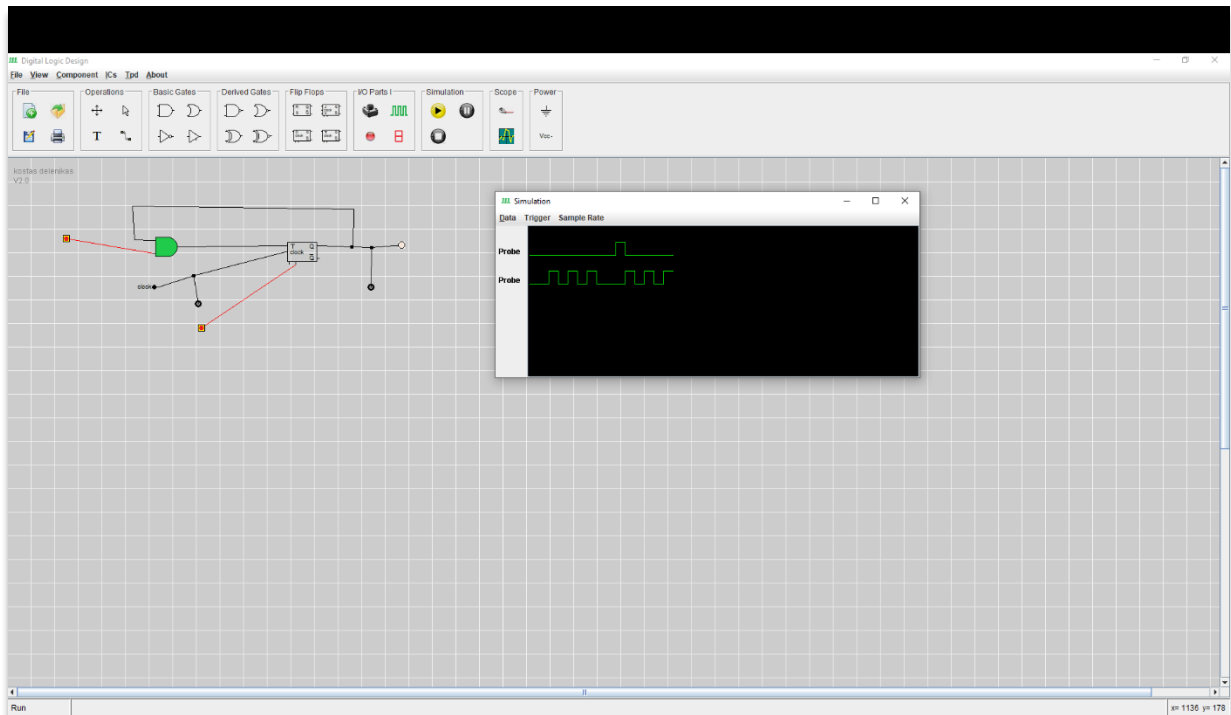


Figure 56 Διεπαφή χρήστη στο Digital Logic Design.

2.17.1 Λειτουργικά χαρακτηριστικά

Το Digital Logic Design περιέχει όλες τις λογικές πύλες και μπορεί να δεχθεί έως και 4 εισόδους για τις πύλες του .Μπορεί να έχει προσαρμοσμένη διαδρομή στα καλώδια του. Υπάρχουν τα D, RS, T, JK Flip Flops, πολυπλέκτες, αποπλέκτες, αποκωδικοποιητές, μετρητές και εξάρτημα clock. Επίσης, ο προσομοιωτής παρέχει διαγράμματα χρονισμού εμφανίζοντας τα σήματα από τις αντίστοιχες ετικέτες του κυκλώματος όπου ο χρήστης έχει τοποθετήσει probes. Τέλος, μπορεί να αποθηκεύσει τα σχέδια, να ανοίξει έτοιμα κυκλώματα και να αποθηκεύσει το κύκλωμα σε εικόνα, να το μετατρέψει σε IC αλλά και να το εκτυπώσει.

2.18 Digital

Ο προσομοιωτής Digital [18] είναι ένας δωρεάν προσομοιωτής του έτους 2016 που υποστηρίζεται από Windows\MacOS\Linux. Τα χρώματα που απεικονίζονται με σκούρο πράσινο και ανοικτό πράσινο αντιπροσωπεύουν τις τιμές 0 και 1 αντίστοιχα. Πρόκειται για ένα προσομοιωτή επιπέδου πύλης, behavioral και functional με συνολικό μέγεθος 20.9mb.

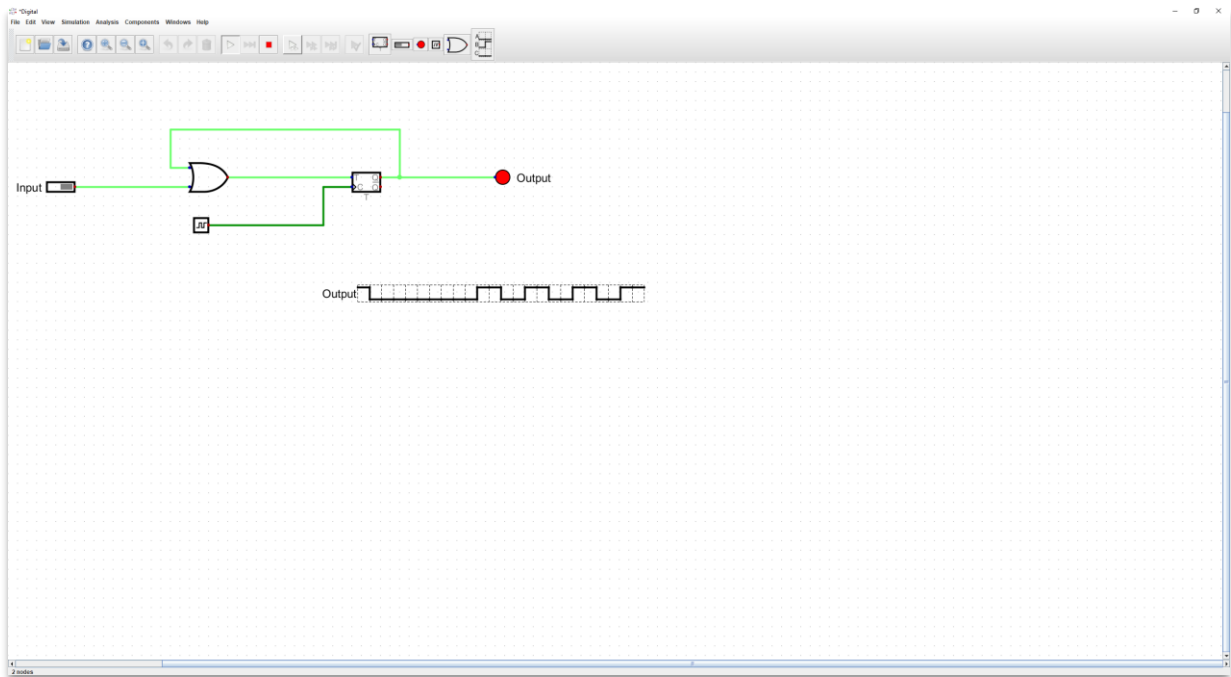


Figure 57 Διεπαφή χρήστη στο Digital.

2.18.1 Λειτουργικά χαρακτηριστικά

Το Digital περιέχει όλες τις λογικές πύλες και μπορεί να δεχθεί έως και 5 εισόδους για τις πύλες του. Μπορεί να έχει προσαρμοσμένη διαδρομή στα καλώδια του. Υπάρχουν τα D, RS, T, JK Flip Flops, πολυπλέκτες, αποπλέκτες, αποκωδικοποιητές, μετρητές και εξάρτημα clock. Επίσης, ο προσομοιωτής παρέχει διαγράμματα χρονισμού εμφανίζοντας τα σήματα από τις αντίστοιχες ετικέτες του κυκλώματος όπου έχουν τοποθετήσει ετικέτες. Υπάρχει δυνατότητα ανάλυσης ολόκληρου του κυκλώματος σε πίνακα αληθείας με όλες τις εισόδους αλλά και τις εξόδους καθώς και όλες τις πιθανές λύσεις του.

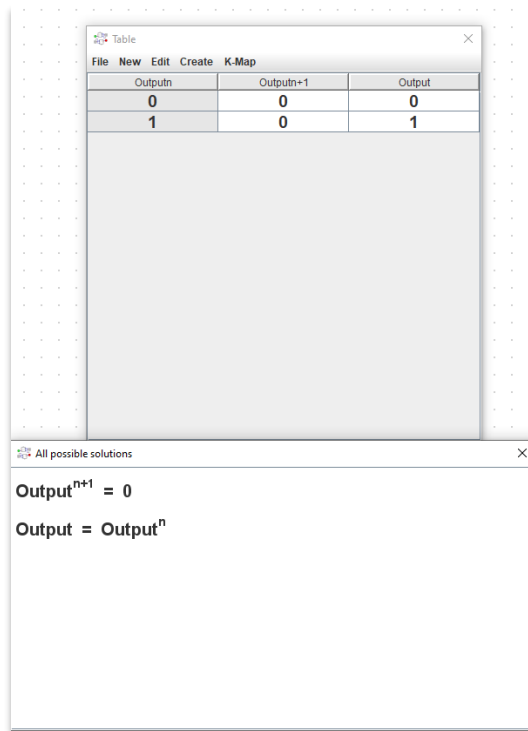


Figure 58 Ανάλυση κυκλώματος από πίνακα αληθείας και πολυπλοκότητα κυκλώματος.

Ο προσομοιωτής δίνει την δυνατότητα να σχεδιαστούν κυκλώματα απλά συμπληρώνοντας πίνακες αληθείας επιλέγοντας το τρόπο σχεδίασης του κυκλώματος (με JK FF, με 2 εισόδους, με 3 εισόδους και με nand πύλες). Επιπρόσθετα, δίνει την επιλογή για combinational και sequential πίνακες αληθείας από μίας έως οκτώ μεταβλητές αλλά και οπτικοποίηση όλων των παραπάνω δυνατοτήτων σε χάρτη Karnaugh. Στο Digital ο χρήστης μπορεί να σχεδιάσει τα δικά του FSMs και να δημιουργήσει κυκλώματα μέσα από λογικές εκφράσεις.

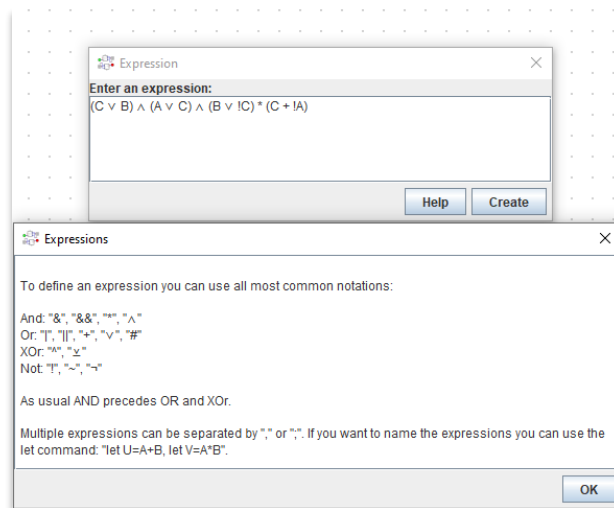


Figure 59 Δημιουργία κυκλώματος από λογικές εκφράσεις.

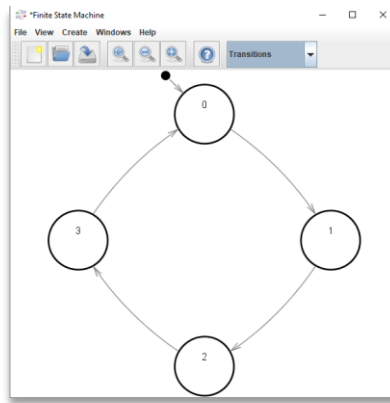


Figure 60 Δημιουργία διάγραμμα καταστάσεων.

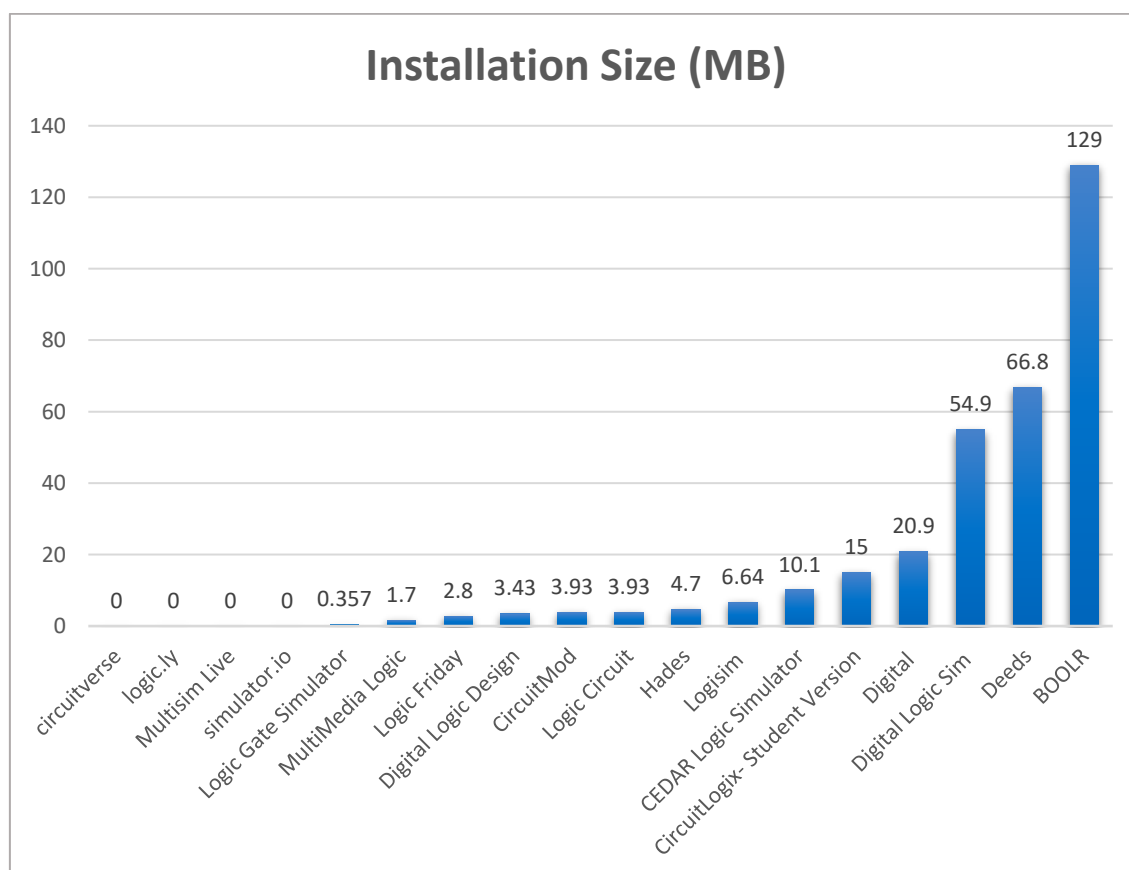
Υπάρχει δυνατότητα ελέγχου για λάθη και ο χρήστης μπορεί υλοποιήσει έλεγχο με εξωτερικό test file από shell για να εκτελεστεί έλεγχο στο κύκλωμα του αφού βρίσκεται στο κατάλληλο directory με την ακόλουθη εντολή `java -cp Digital.jar CLI test [file to test] [-tests [optional file with test cases]]`. Τέλος, μπορεί να αποθηκεύσουμε τα κυκλώματα του εκάστοτε χρήστη, να ανοίξει έτοιμα projects και να πραγματοποιήσει μετατροπή του κυκλώματος σε SVG, PNG, VHDL, Verilog και zip file.

3 Συνολική ανάλυση χαρακτηριστικών προσομοιωτών

Σε αυτή την ενότητα θα γίνει εκτενής ανάλυση των δυνατοτήτων και λειτουργιών όλων των προσομοιωτών που έχουν αναφερθεί έως τώρα με την βοήθεια διαγραμμάτων, με σκοπό να γίνει κατανοητό το ποιοι λογικοί προσομοιωτές ξεχώρισαν βάση δυνατοτήτων και της εμπειρίας χρήσης που σχηματίστηκε.

3.1 Μέγεθος και Λειτουργικό σύστημα

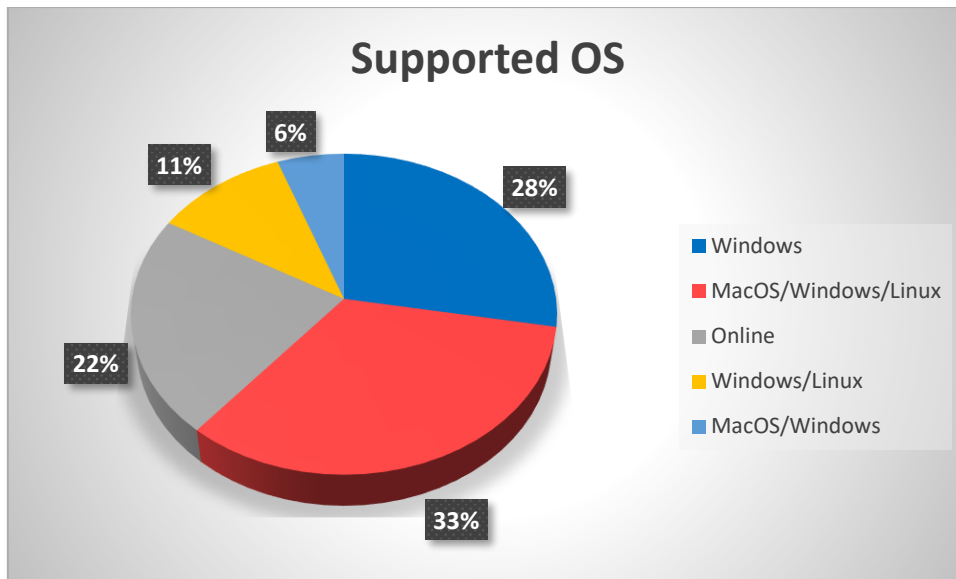
Αρχικά κριτήρια επιλογής προσομοιωτή είναι το μέγεθος του αλλά και τα υποστηριζόμενα λειτουργικά συστήματα.



Εικόνα 8 Γράφημα με το μέγεθος κάθε προσομοιωτή.

Η συνολική διακύμανση του μεγέθους κάθε προσομοιωτή είναι από μερικά MB έως τα 129 MB. Η μέση τιμή που προκύπτει από τους προσομοιωτές που επιλέχθηκαν για ανάλυση κυμαίνεται κοντά στα 25 MB με τους προσομοιωτές DEEDS, BOOLRM, Digital Logic Sim να ξεπερνούν κατά πολύ τη μέση τιμή αυτή.

Ακολούθως, τα υποστηριζόμενα λειτουργικά συστήματα Windows/MacOS/Linux σε συνδυασμό με τους online λογικούς προσομοιωτές μπορούν συνδυαστικά να απευθυνθούν στη πλειοψηφία των χρηστών κατέχοντας το μεγαλύτερο ποσοστό από το σύνολο των λογικών προσομοιωτών που μελετήθηκαν (55%).

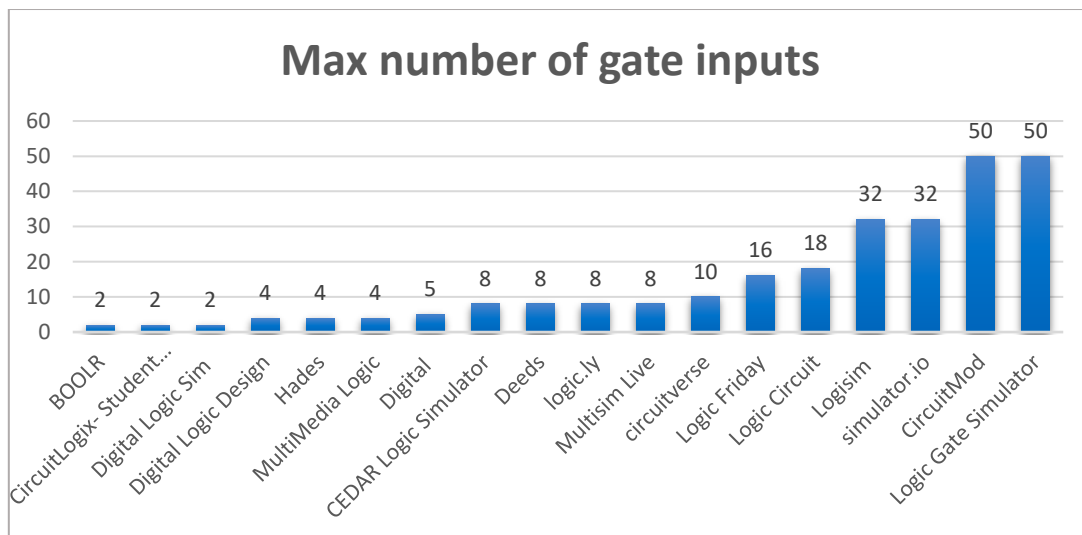


Εικόνα 9 Γράφημα με τα υποστηριζόμενα λειτουργικά συστήματα.

Τα παραπάνω αποτελέσματα δείχνουν τα υποστηριζόμενα λειτουργικά συστήματα κάθε προσομοιωτή. Οι προσομοιωτές που λειτουργούν σε κάθε λειτουργικό σύστημα (Logisim, Deeds, Cedar, CircuitLogix, Digital Logic Sim, Digital) αλλά και εκείνοι που είναι online (Logic.ly, Simulator.io, Multisim Live, Circuitverse) απασχολούν το μεγαλύτερο ποσοστό χρηστών καθώς αυτοί οι προσομοιωτές λειτουργούν σε όλα τα συστήματα.

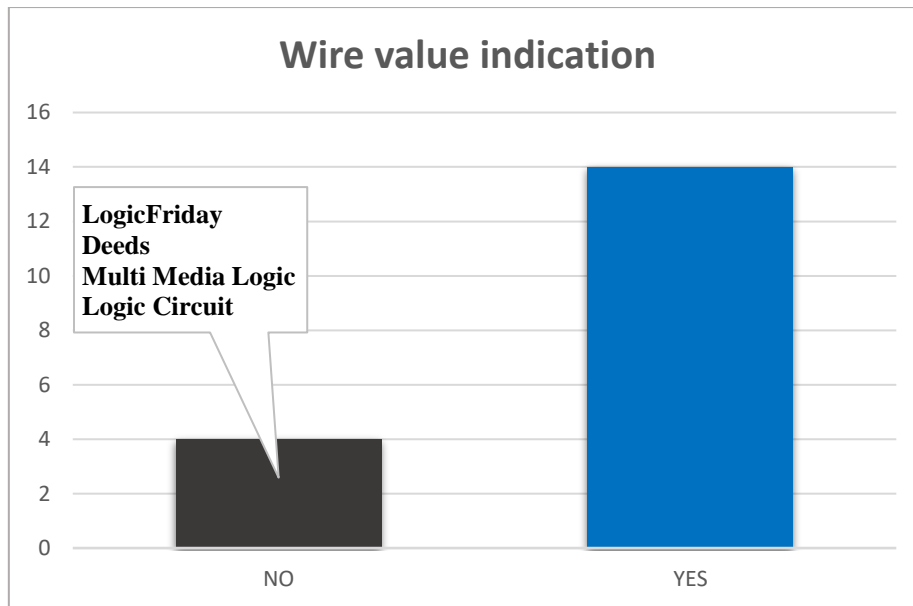
3.2 Αριθμός εισόδων, Διαδρομή καλωδίων και Απεικόνιση τιμής στα καλώδια

Στη συνέχεια, θα παρουσιαστούν συνολικά τα δεδομένα που αφορούν το μέγιστο αριθμό εισόδων, την προσαρμοσμένη διαδρομή καλωδίων και την απεικόνιση τιμής επάνω στα καλώδια του κυκλώματος.



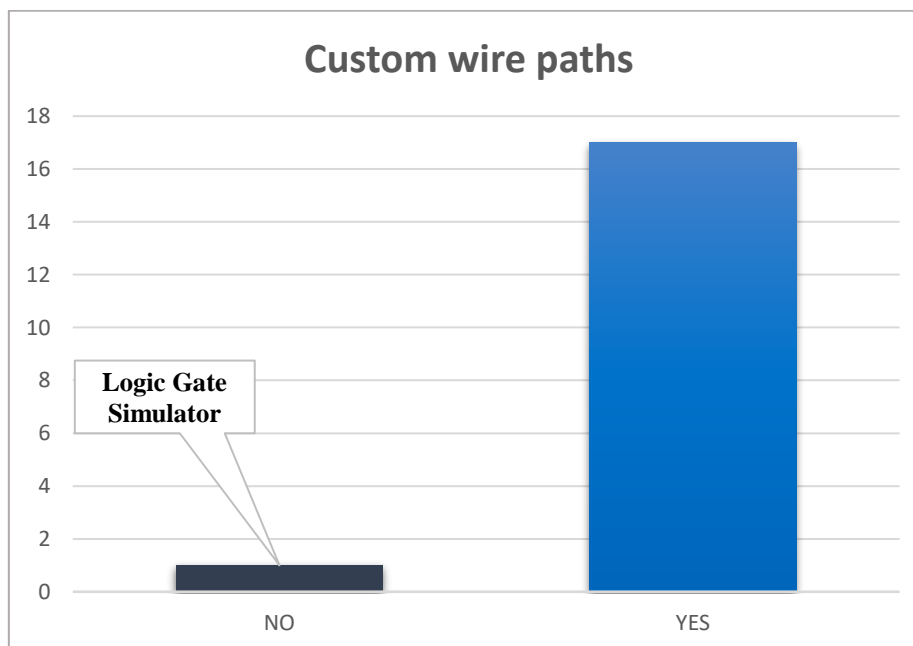
Εικόνα 10 Γράφημα με τον μέγιστο αριθμό εισόδων στις πύλες κάθε προσομοιωτή.

Όσον αφορά τους προσομοιωτές Logic Gate Simulator και CircuitMod δεν έχουν όριο στον αριθμό των εισόδων τους. Οι προσομοιωτές CircuitLogix, BOOLR και Digital Logic Sim δέχονται δύο εισόδους στις πύλες τους, νούμερο το οποίο μπορεί να ξεπεραστεί εύκολα και να προκύψουν σχεδιαστικά προβλήματα λόγω των αναγκών του εκάστοτε κυκλώματος.



Εικόνα 11 Γράφημα για την απεικόνιση τιμών επάνω στα καλώδια.

Η δυνατότητα του προσομοιωτή να δείχνει την κάθε τιμή που έχουν τα καλώδια του σε πραγματικό χρόνο είναι μία λειτουργία που μπορεί να φανεί χρήσιμη σε πολλούς χρήστες που σχεδιάζουν κυκλώματα και μπορεί να μην είναι εξοικειωμένοι με τη σχεδίαση ή να έχουν φτάσει κάποιο σχέδιο αυξημένης δυσκολίας. Σε περίπτωση σφάλματος η δυνατότητα του χρήστη να γνωρίζει ακριβώς τι συμβαίνει σε κάθε καλώδιο του κυκλώματος του μπορεί να τον γλιτώσει από μία διαδικασία που μπορεί να αποβεί χρονοβόρα μέχρι να εντοπιστεί το σφάλμα. Έτσι, οι προσομοιωτές που δεν έχουν τη δυνατότητα προβολής της τιμής που φέρουν τα καλώδια είναι λιγότερο χρηστικοί σε περίπτωση σφάλματος αλλά και κατά τη διάρκεια της προσομοίωσης.

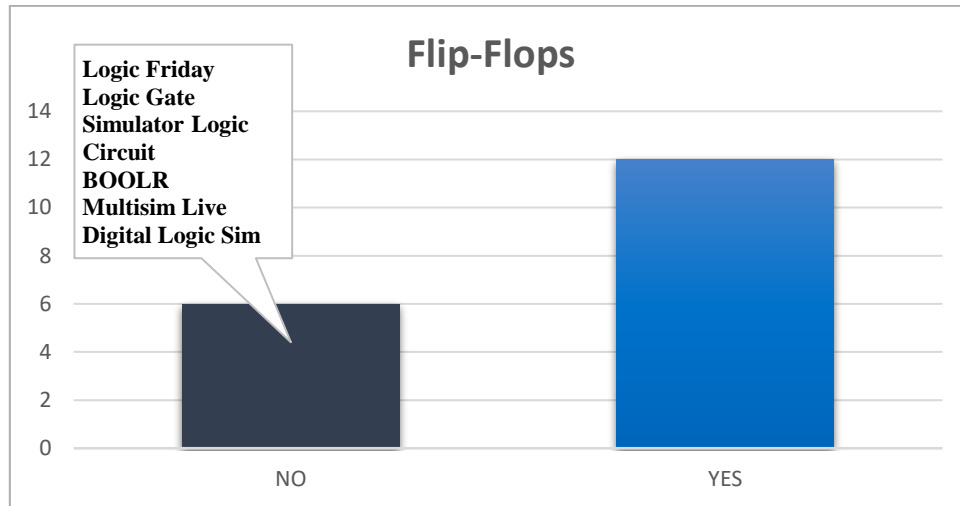


Εικόνα 12 Γράφημα για δυνατότητα προσαρμοσμένης διαδρομής στα καλώδια του κυκλώματος.

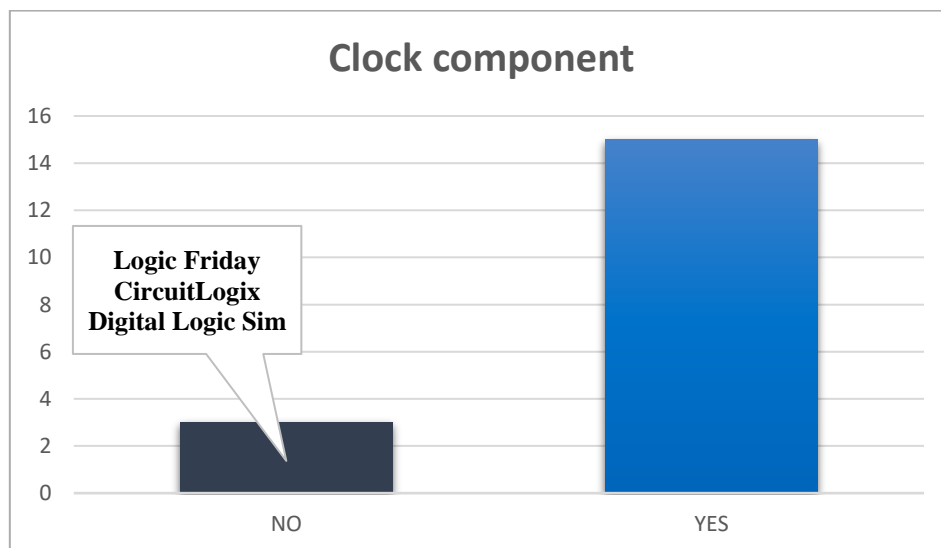
Σε σχέση με τη λειτουργία της ανάδειξης τιμών οι δυνατότητα των προσαρμοσμένων καλωδίων είναι απαραίτητη καθώς μπορούν να προκύψουν προβλήματα κατανομής των πυλών και σύνδεσης των καλωδίων μεταξύ τους εντός του εκάστοτε προσομοιωτή που δεν προσφέρει αυτή τη δυνατότητα.

3.3 Εξαρτήματα Flip-Flop, εξάρτημα ρολογιού και διάγραμμα χρονισμού

Εξίσου σημαντικό με όλα τα προηγούμενα στοιχεία είναι να γίνει αναφορά για το ποιοι προσομοιωτές υποστηρίζουν Flip-Flop, εξάρτημα ρολογιού και διάγραμμα χρονισμού.

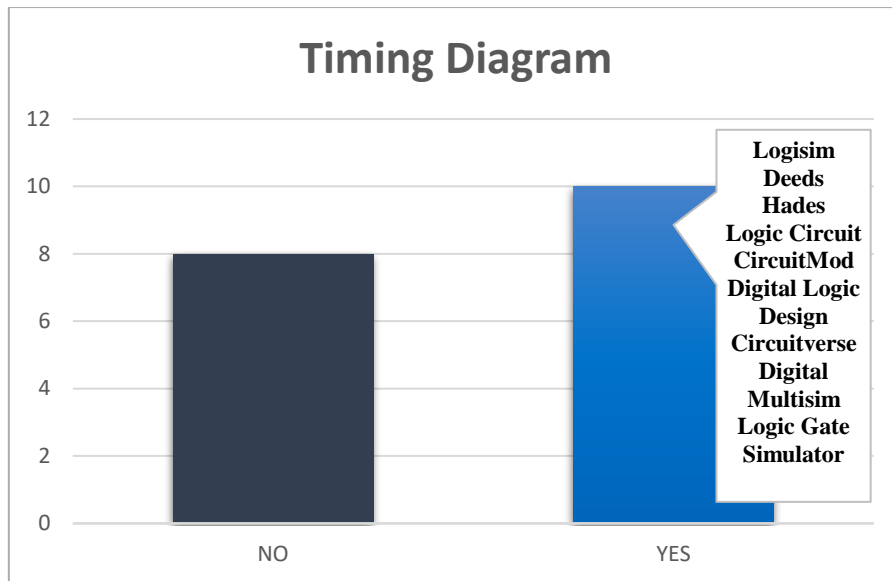


Εικόνα 13 Γράφημα για προσομοιωτές που υποστηρίζουν Flip-Flop.



Εικόνα 14 Γράφημα για προσομοιωτές που υποστηρίζουν εξάρτημα clock.

Η ύπαρξη Flip-Flop χωρίς να υπάρχει εξάρτημα ρολογιού είναι ανεπαρκής ακόμα και εάν δημιουργηθεί το οποιοδήποτε Flip-Flop από το μηδέν με χρήση λογικών πυλών. Έτσι ο συνδυασμός απουσίας Flip-Flop μαζί με εξάρτημα ρολογιού στους προσομοιωτές Logic Friday και Digital Logic Sim καθιστούν τη σχεδίαση σύγχρονων κυκλωμάτων αδύνατη.

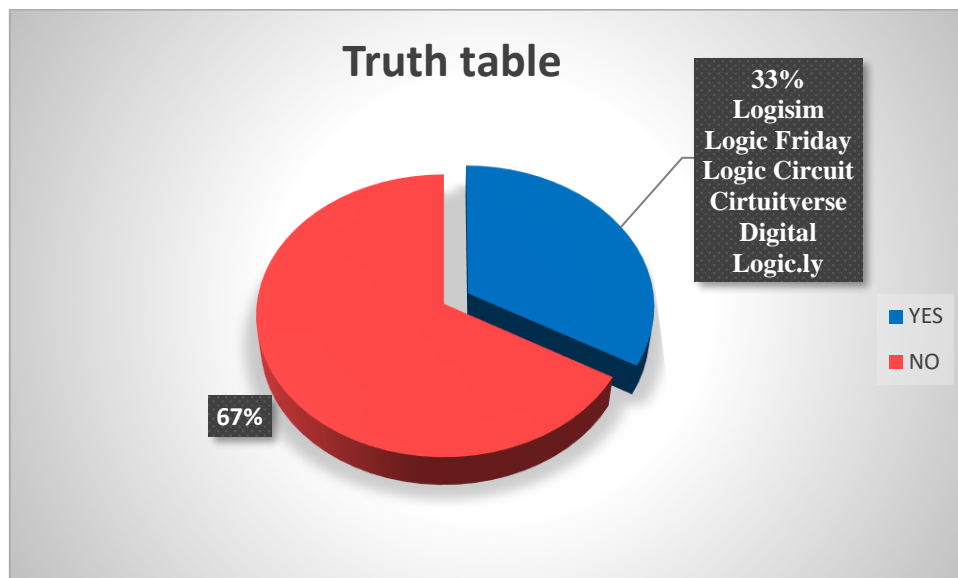


Εικόνα 15 Γράφημα για λειτουργία διαγράμματος χρονισμού.

Από τους δεκαπέντε προσομοιωτές που έχουν εξάρτημα ρολογιού μόνο οι δέκα από αυτούς έχουν δυνατότητα ανάλυσης του χρονισμού τους σε διάγραμμα με αυτούς να αναφέρονται ονοματικά στο παραπάνω γράφημα της εικόνας 15.

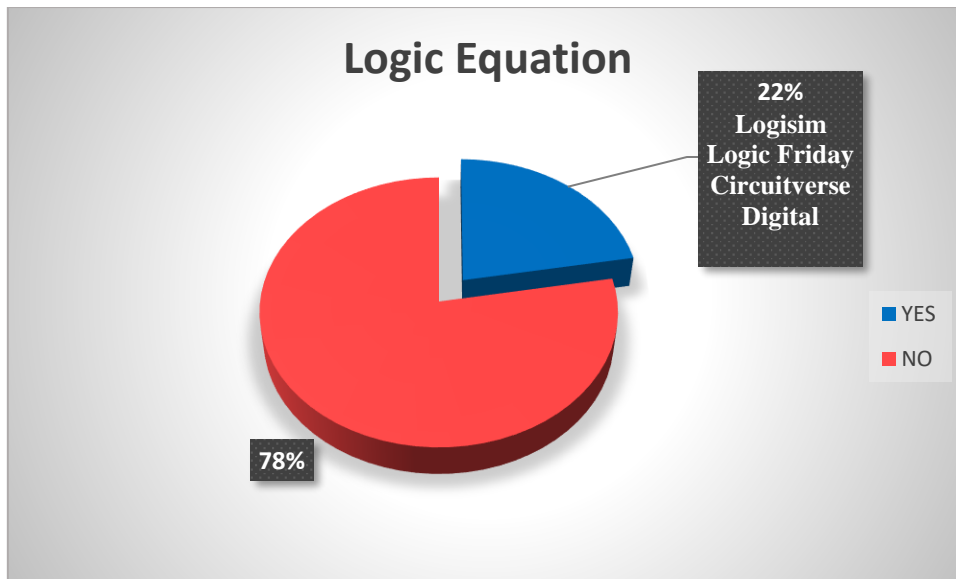
3.4 Πίνακες αληθείας, Λογικές Εκφράσεις και Απλοποίηση κυκλώματος

Ακολούθως, θα αναλύσουμε ποιοι προσομοιωτές έχουν τη δυνατότητα να σχεδιάσουν κυκλώματα από λογικές εκφράσεις και πίνακες αληθείας και να απλοποιούν τα κυκλώματα τους.



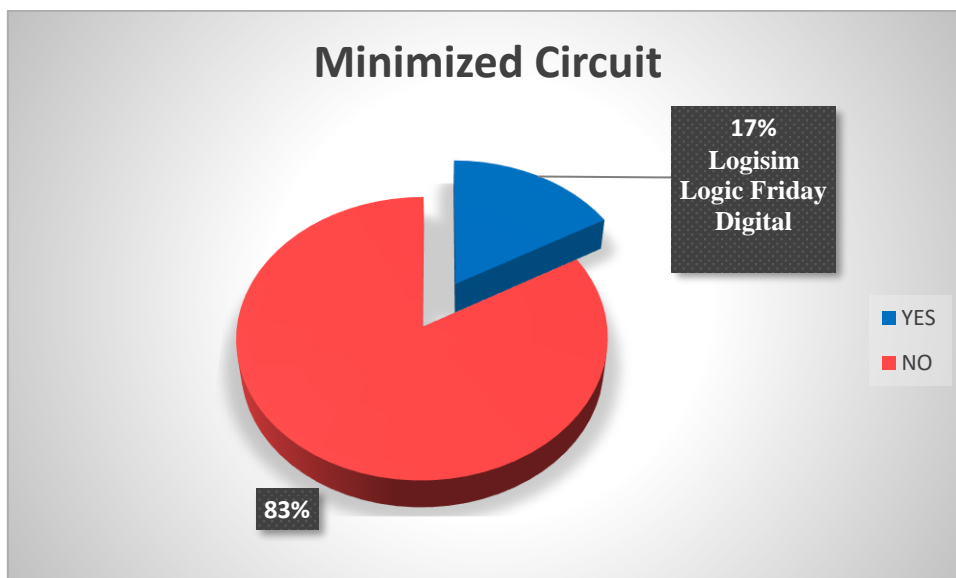
Εικόνα 16 Γράφημα προσομοιωτών που υποστηρίζουν πίνακες αληθείας.

Η αυτόματη σχεδίαση κυκλωμάτων από τη συμπλήρωση ενός πίνακα αληθείας αλλά και η περιγραφή ολοκληρωμένων κυκλωμάτων σε πίνακες αληθείας καθιστά σημαντική την συγκεκριμένη λειτουργία καθώς μπορεί επιτευχθεί μείωση του συνολικού χρόνου που απαιτείται για τη σχεδίαση ή την περιγραφή του εκάστοτε κυκλώματος.



Εικόνα Γράφημα για λειτουργία λογικών εκφράσεων.

Όπως και με τους πίνακες αληθείας που έγινε αναφορά προηγουμένως εξίσου σημαντικό χαρακτηριστικό είναι και η σχεδίαση και ανάλυση κυκλωμάτων από λογικές εκφράσεις. Όσοι προσομοιωτές αναλύθηκαν στο προηγούμενο κεφάλαιο που υποστηρίζουν λογικές εκφράσεις υποστηρίζουν και πίνακες αληθείας. Έτσι, οι προσομοιωτές που ικανοποιούν και τις δύο αυτές λειτουργίες είναι οι Logisim, Logic Friday, Circuitverse και Digital.

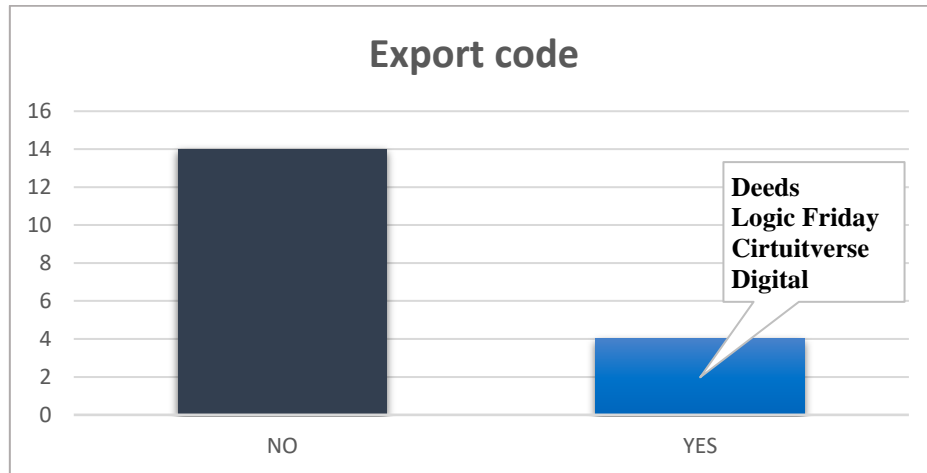


Εικόνα 17 Γράφημα για ελαχιστοποίηση κυκλωμάτων.

Τα σχέδια που απαρτίζονται από πολλές λογικές πύλες, σύνθετες λογικές εκφράσεις και μεγάλους πίνακες αληθείας μπορούν πολλές φορές να απλοποιηθούν με αποτέλεσμα την αύξηση της απόδοσης τους. Σε κάθε τέτοια περίπτωση οι επιλογή ενός εκ των Logisim, Logic Friday και Digital που υποστηρίζουν τη λειτουργία της απλοποίησης κυκλωμάτων θα κάνει τα ολοκληρωμένα κυκλώματα που έχουν σχεδιαστεί αποδοτικότερα σε περίπτωση που μπορούν να δεχθούν απλοποίηση.

3.5 Δημιουργία κώδικα από γλώσσας περιγραφής υλικού και υποστήριξη scripting

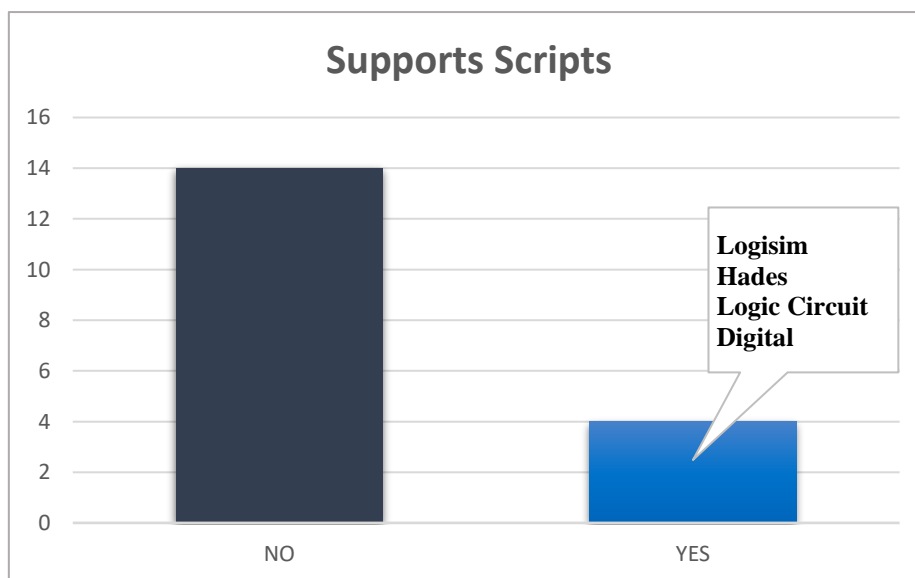
Αρκετοί από τους προσομοιωτές που έγιναν αναφορά στο προηγούμενο κεφάλαιο προσφέρουν δυνατότητες που εμπλέκουν κώδικα. Πιο συγκεκριμένα, υποστηρίζουν αρχεία να γίνουν export σε VHDL, VERILOG, C, δεχόντουσαν scripts για το testing κατά τη διάρκεια της προσομοίωσης τους και δεχόντουσαν ως είσοδο .jar αρχεία το οποία χρησιμοποιούνται για testing ή για να αλλάξουν οι προτιμήσεις του εκάστοτε προσομοιωτή.



Εικόνα 18 Γράφημα για δυνατότητα export σε κάποια γλώσσα περιγραφής υλικού.

	VHDL	VERILOG	C
Deeds	✓		
Logic Friday			✓
Cirtuitverse		✓	
Digital	✓	✓	

Οι προσομοιωτές Deeds, Logic Friday, Circuitverse και Digital υποστηρίζουν εξαγωγή των σχεδιασμένων κυκλωμάτων σε κάποια γλώσσα περιγραφής υλικού εκ των VHDL, VERILOG και C. Πιο συγκεκριμένα ο προσομοιωτής DEEDS σε VHDL, ο Logic Friday σε C, ο Circuitverse σε VERILOG και ο Digital σε VHDL και VERILOG.

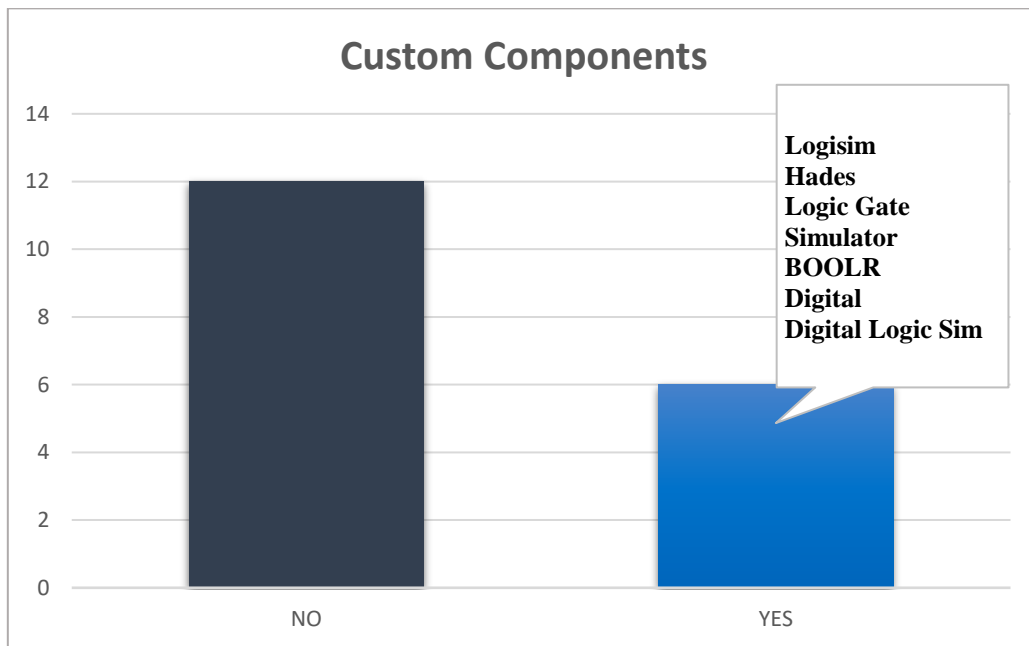


Εικόνα 19 Γράφημα για δυνατότητα Scripting.

Η ανάγκη του χρήστη να σχεδιάσει δικά του αντικείμενα που εξυπηρετούν συγκεκριμένες λειτουργίες, η είσοδος συγκεκριμένων τιμών σε συγκεκριμένες λογικές πύλες με σκοπό να εξεταστεί μία περίπτωση που δεν γίνεται κατά κανονική λειτουργία του κυκλώματος αντιληπτή και η τροποποίηση των παραμέτρων και των προτιμήσεων του προσομοιωτή είναι μερικές ανάγκες μπορεί να προκύψουν κατά την χρήση του οποιουδήποτε προσομοιωτή. Οι προσομοιωτές που παρέχουν λειτουργίες που εξυπηρετούν τις παραπάνω ανάγκες φαίνονται στο παραπάνω γράφημα στην περιοχή YES. Συγκεκριμένα, ο προσομοιωτής Logisim μπορεί να δημιουργήσει νέα αντικείμενα και να αλλάξει τις προτιμήσεις του μέσω της Java από .jar file, ο προσομοιωτής Hades εξυπηρετεί τις ίδιες λειτουργίες με το Logisim μέσω Jython (Java based version of Python), ο προσομοιωτής Logic Circuit μπορεί μέσω της Python να πετύχει εισόδους συγκεκριμένων τιμών σε συγκεκριμένες λογικές πύλες και ο προσομοιωτής Digital μπορεί να εισάγει .jar αρχείο για έλεγχο λειτουργίας του κυκλώματος με σκοπό να γίνει εντοπισμός τυχόν λαθών.

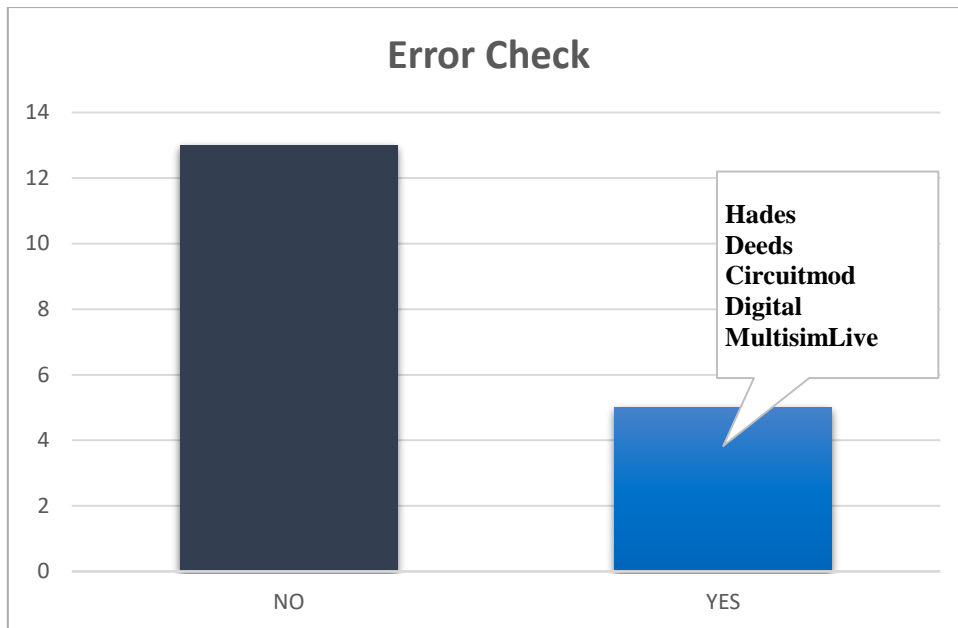
3.6 Ειδικά εξαρτήματα, έλεγχος σφαλμάτων και εκτύπωση κυκλώματος

Ακολούθως, θα γίνει αναφορά στις λειτουργίες δημιουργίας ειδικών πυλών χωρίς χρήση κώδικα, αυτόματου ελέγχου σφαλμάτων και εκτύπωσης κυκλώματος.



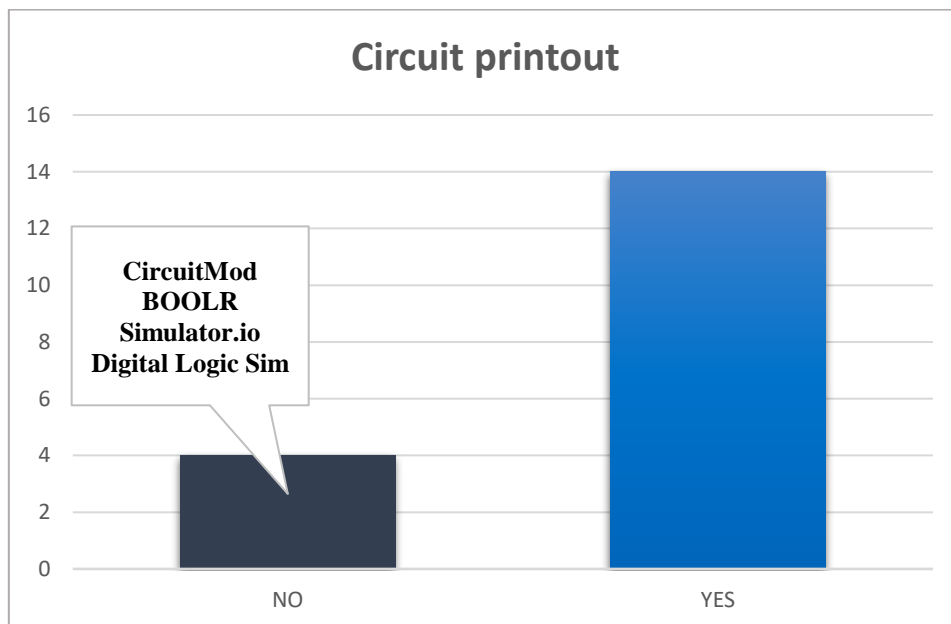
Εικόνα 20 Γράφημα για χρήση ειδικών αντικειμένων.

Η δημιουργία ειδικών εξαρτημάτων με σκοπό την υλοποίηση κάποιας πράξης που δεν πραγματοποιείται από τις ήδη υπάρχουσες λογικές πύλες φαίνεται στο παραπάνω γράφημα στην περιοχή του YES. Οι προσομοιωτές που μας δίνουν αυτή την δυνατότητα χωρίς την χρήση κώδικα είναι οι Logic Gate Simulator, BOOLR, Digital, Digital Logic Sim.



Εικόνα 21 Γράφημα για λειτουργία ελέγχου σφαλμάτων.

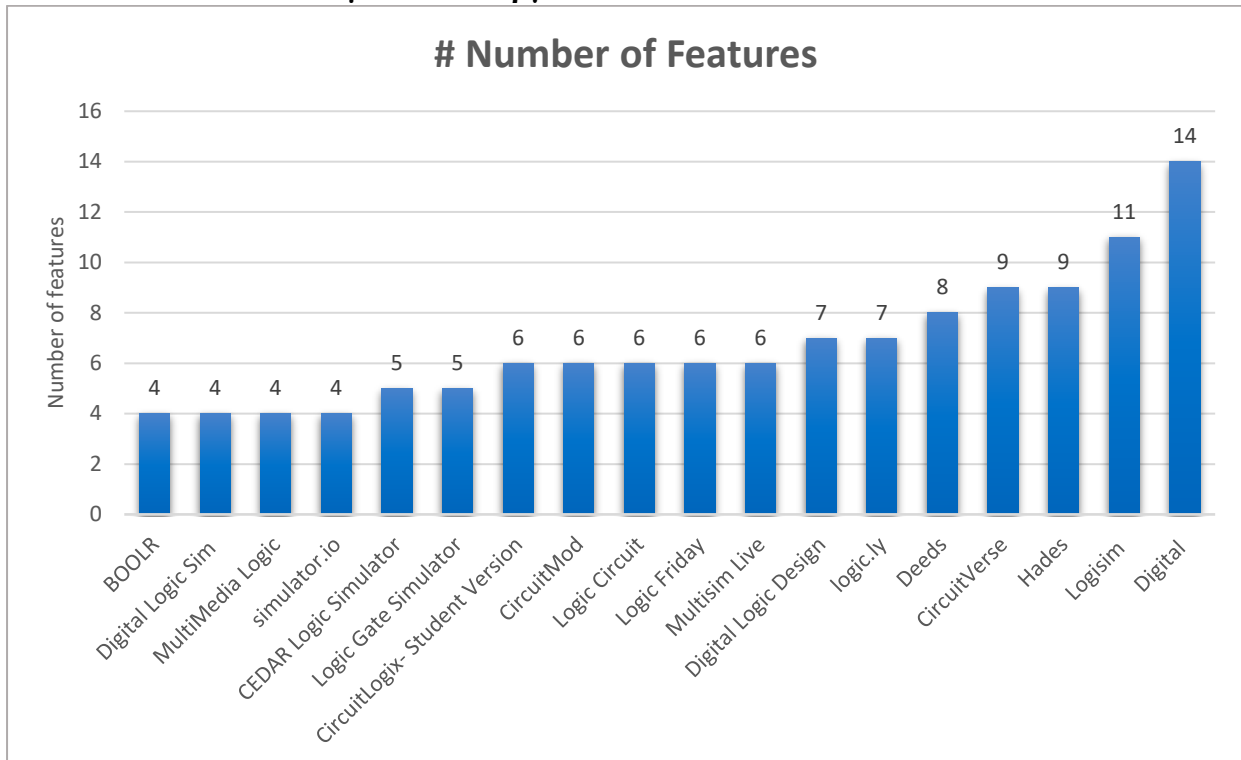
Η λειτουργία του ελέγχου σφαλμάτων κατά τη διάρκεια της προσομοίωσης ή ακόμα και πριν αυτή ξεκινήσει μας δίνει την δυνατότητα να εντοπίσουμε τυχόν λάθη που έχουν προκύψει στο κύκλωμα μας μειώνοντας έτσι αρκετά τον χρόνο σχεδίασης του κυκλώματος μας αποδίδοντάς κυκλώματα με λιγότερά εμφανή και μη εμφανή σφάλματα που εντοπίζονται από το εκάστοτε προσομοιωτή. Συγκεκριμένα, στο παραπάνω γράφημα στην στήλη YES φαίνονται αναλυτικά όλοι οι προσομοιωτές που μας παρέχουν την δυνατότητα αυτή.



Εικόνα 22 Γράφημα για λειτουργία εκτύπωσης κυκλώματος.

Η λειτουργία εκτύπωσης του κυκλώματος μας δεν μπορεί να εκτελεστεί από όσους προσομοιωτές βρίσκονται στην στήλη NO του παραπάνω γραφήματος.

3.7 Συνολικά αποτελέσματα λειτουργιών



Εικόνα 23 Συγκεντρωτικό γράφημα δυνατοτήτων προσομοιωτών.

Τέλος, με βάση όσες λειτουργίες αναφέρθηκαν σε αυτό και το προηγούμενο κεφάλαιο για κάθε προσομοιωτή ξεχωριστά δημιουργήθηκε το γράφημα της εικόνας 26 όπου απεικονίζονται αριθμημένες όλες οι δυνατότητες που έχει κάθε προσομοιωτής που μελετήθηκε στο προηγούμενο κεφάλαιο. Ειδικότερα, ο προσομοιωτής Digital κατέχει την πλειοψηφία των δυνατοτήτων που μελετήθηκαν έως τώρα ενώ οι προσομοιωτές BOOLR, Digital Logic Sim, Multimedia Logic και Simulator.io κατέχουν τις λιγότερες. Οι προσομοιωτές που έχουν τον ίδιο αριθμό λειτουργιών δεν είναι σε καμία περίπτωση ισάξιοι καθώς εκτός από τις διαφορές στη φιλικότητα προς το χρήστη και του αριθμού παρεχόμενων εξαρτημάτων που παρατηρήθηκαν μπορούν να έχουν διαφορές στις υποστηριζόμενες λειτουργίες. Για παράδειγμα, ο προσομοιωτής BOOLR έχει τον ίδιο αριθμό λειτουργιών με το προσομοιωτή Multimedia Logic αλλά ο πρώτος δεν παρέχει εξαρτήματα Flip-Flop ενώ ο δεύτερος παρέχει.

4 Συμπεράσματα

Το σύνολο των επιλεγμένων λογικών προσομοιωτών που επιλέχθηκαν είναι ικανοί να λειτουργήσουν στη πλειοψηφία των υπολογιστών καθώς και υπάρχουν λογικοί προσομοιωτές που απευθύνονται για κάθε επίπεδο σχεδίασης ικανοποιώντας κάθε μέσο χρήστη. Σε αυτό το κεφάλαιο θα γίνει αναφορά όλων των διαθέσιμων αποτελεσμάτων από τα προηγούμενα κεφάλαια με σκοπό την αναφορά ενός προσομοιωτή για σύσταση με κριτήρια τις δυνατότητες του και την εμπειρία χρήσης. Ο διαχωρισμός θα γίνει σε δύο κατηγορίες προσομοιωτών σε αυτό που απευθύνεται να γίνει εγκατάσταση σε λειτουργικά συστήματα Windows/MacOS/Linux και σε αυτό που απευθύνεται σε online χρήστες. Επιπροσθέτως, θα γίνει αναφορά και στους λιγότερο ικανούς προσομοιωτές βάση των στοιχείων που έχουν συγκεντρωθεί έως τώρα.

4.1 Σύσταση Windows/MacOS/Linux Based προσομοιωτή

Προτεινόμενος Windows/MacOS/Linux based προσομοιωτής θα επιλεγεί ο προσομοιωτής Digital. Το συνολικό του μέγεθος 20.9mb που ήταν μικρότερο του μέσου όρου το οποίο ήταν 23mb, η εμπειρία χρήσης του που ήταν ιδιαίτερα θετική με κύριους παράγοντες να είναι: το περιβάλλον χρήσης, πλήθος εξαρτημάτων, η σύνδεση καλωδίων, η τοποθέτηση αντικειμένων και η παροχή πλήρους βοηθητικών εγγράφων (documentation). Επίσης, η δυνατότητα σχεδίασης ακολουθιακών κυκλωμάτων, η χρήση λογικών συναρτήσεων και πινάκων αληθείας για παραγωγή κυκλωμάτων τα οποία μετατρέπονται κατευθείαν σε απλοποιημένη μορφή και επιλογές όπως η σχεδίαση FSMs, ανάλυση χρονισμού κυκλώματος και testing μέσω εξωτερικού αρχείου αλλά και η μετατροπή του κυκλώματος σε γλώσσα περιγραφής υλικού ήταν μερικές από τις δυνατότητες που παρέχει ο προσομοιωτής Digital σε κάθε του χρήστη. Οι δυνατότητες αυτού του προσομοιωτή μπορούν να εξυπηρετήσουν κάθε μέσο χρήστη καθώς πρόκειται για ένα ολοκληρωμένο δωρεάν λογισμικό σχεδίασης ψηφιακών κυκλωμάτων.

4.2 Σύσταση Online προσομοιωτή

Προτεινόμενος online προσομοιωτής θα επιλεγεί ο προσομοιωτής CircuitVerse. Η εμπειρία χρήσης του ήταν θετική σε συνδυασμό με τις δυνατότητες που δημιουργούν ένα ολοκληρωμένο σύνολο παρόμοιο με το προσομοιωτή της προηγούμενης κατηγορίας (4.1) με το παράγοντα του διαδικτύου να παίζει αναμφισβήτητο θετικό ρόλο σε αυτό προσθέτοντας τις δυνατότητες διαμοιρασμού υλικού online στην επίσημη σελίδα του προσομοιωτή αλλά και σε χρήστες που βρίσκονται στην ίδια ομάδα.

4.3 Αναφορά λιγότερο αποτελεσματικών προσομοιωτών

Από τους τέσσερις προσομοιωτές που είχαν μόνο τέσσερις λειτουργίες που αναφέρθηκαν στα προηγούμενα κεφάλαια επιλέχθηκαν δύο για την αναφορά σε αυτήν τη κατηγορία. Ο κύριος λόγος που επιλέγονται δύο και όχι και οι τέσσερις είναι ότι παρατηρήθηκε διαφορά στη φιλικότητα προς το χρήστη, στον αριθμό των εξαρτημάτων που παρείχαν και στο συνολικό μέγεθος των προσομοιωτών. Τη πρωτιά σε αυτή τη κατηγορία κατέχει ο προσομοιωτής BOOLR όπου εκτός τις περιορισμένες δυνατότητες που είχε κατέχει και το μεγαλύτερο μέγεθος προσομοιωτή 129mb. Με το προσομοιωτή Digital Logic Sim που ομοίως έχει λίγες δυνατότητες και μεγάλο μέγεθος 54.9mb να ολοκληρώνει αυτή τη κατηγορία.

5 Αναφορές και Βιβλιογραφία

M. Mano Morris, Michael D. Ciletti. (2014). Ψηφιακή Σχεδίαση (5η έκδοση). Αθήνα: Εκδόσεις Παπασωτηρίου

Κοσσιδάς Α.Θ. (1996). Σχεδίαση Ψηφιακών Κυκλωμάτων. Αθήνα: Εκδόσεις Μπέρος

Balabanian N., Carlson B. (2007). Digital Logic Design Principles. John Wiley

Holdsworth Brian, Woods Clive (2002). Digital Logic Design (4η έκδοση)

Nelson V., Nagle H., Carroll B., Irwin J. (1995). Digital Logic Circuit Analysis and Design. Prentice-Hall

Φραγκάκης Γ. (1975). Λογικά Κυκλώματα. Αθήνα

[1] Sebastian Lague. (2021). Digital Logic Sim. <https://sebastian.itch.io/digital-logic-sim> (Accessed November 2021)

[2] Multimedia Logic. (2013). <https://sourceforge.net/projects/multimedialogic/> (Accessed November 2021)

[3] Logic Design Inc. (2011). CircuitLogix Student Version. https://www.circuitlogix.com/student_version.php (Accessed November 2021)

[4] Steve Kollmanberger et al. (2011). Logic Gate Simulator. <https://kolls.net/gatesim/> (Accessed November 2021)

[5] CircuitMod. (2007). <http://circuitmod.sourceforge.net/> (Accessed November 2021)

[6] Bastian Born. (2015). Simulator.io. <https://simulator.io> (Accessed November 2021)

[7] Logic Circuit. (2009). <https://www.logiccircuit.org/> (Accessed November 2021)

[8] Cedarville University, Benjamin Sprague, Matthew Lewellyn, David Knierim, Joshua Lansford, Nathan Harro. (2007). CEDAR Logic Simulator. <https://sourceforge.net/projects/cedarlogic/> (Accessed November 2021)

[9] National Instruments Corp. (2022). Multisim Live. <https://www.multisim.com> (Accessed November 2021)

[10] CircuitVerse. (2022). CircuitVerse. <https://circuitverse.org> (Accessed November 2021)

[11] Steve Rickman. (2006-2011). Logic Friday. https://download.cnet.com/Logic-Friday/3000-20415_4-75848245.html (Accessed November 2021)

[12] Bowler Hat LLC. (2008-2022). Logic.ly. <https://logic.ly> (Accessed November 2021)

[13] Jaap Dechering, Gees Brouwer, Teun de Theije. (2017). BOOLR. <http://boolr.me> (Accessed November 2021)

[14] University of Genoa, Italy. (2002-2021). Deeds. <https://www.digitalelectronicsdeeds.com/index.html> (Accessed November 2021)

[15] Carl Burch. (2005). Logisim. <http://www.cburch.com/logisim/index.html> (Accessed November 2021)

[16] Norman Hendrich, Dept. Computer Science, Univ. of Hamburg. (1997-2005). Hades. <https://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/index.html> (Accessed November 2021)

[17] Digital Circuit Design. (2022). Digital Logic Design. <https://www.digitalcircuitdesign.com> (Accessed November 2021)

[18] H. Neemann. (2016-2021). Digital. <https://github.com/hneemann/Digital> (Accessed November 2021)