Τίτλος: «Σχεδίαση και προσομοίωση παράλληλης αριθμητικής λογικής μονάδας (ALU) για την επεξεργασία δυαδικών αριθμών εύρους 4-bit, με το πρόγραμμα Multisim»

ΦΟΙΤΗΤΡΙΑ :

ΒΟΥΛΓΑΡΙΔΟΥ ΜΑΡΙΑ, ΑΕΜ: 2109

ΕΠΙΒΛΕΠΩΝ :

ΚΑΛΟΜΟΙΡΟΣ ΙΩΑΝΝΗΣ, ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ

Εισαγωγή

- Βάση της εργασίας αποτέλεσε ο προσομοιωτής **Multisim**: ένα περιβάλλον λογισμικού που επιτρέπει τη σχεδίαση και την προσομοίωση της λειτουργίας των ηλεκτρονικών κυκλωμάτων.

- Σκοπός της παρούσας εργασίας είναι η σχεδίαση και προσομοίωση μιας παράλληλης αριθμητικής λογικής μονάδας (ALU) για την επεξεργασία δυαδικών αριθμών εύρους 4-bit με το πρόγραμμα Multisim. Συγκεκριμένα θα εκτελεί τις λογικές πράξεις AND και XOR αλλά και τις αριθμητικές ADD και SUBTRACT.

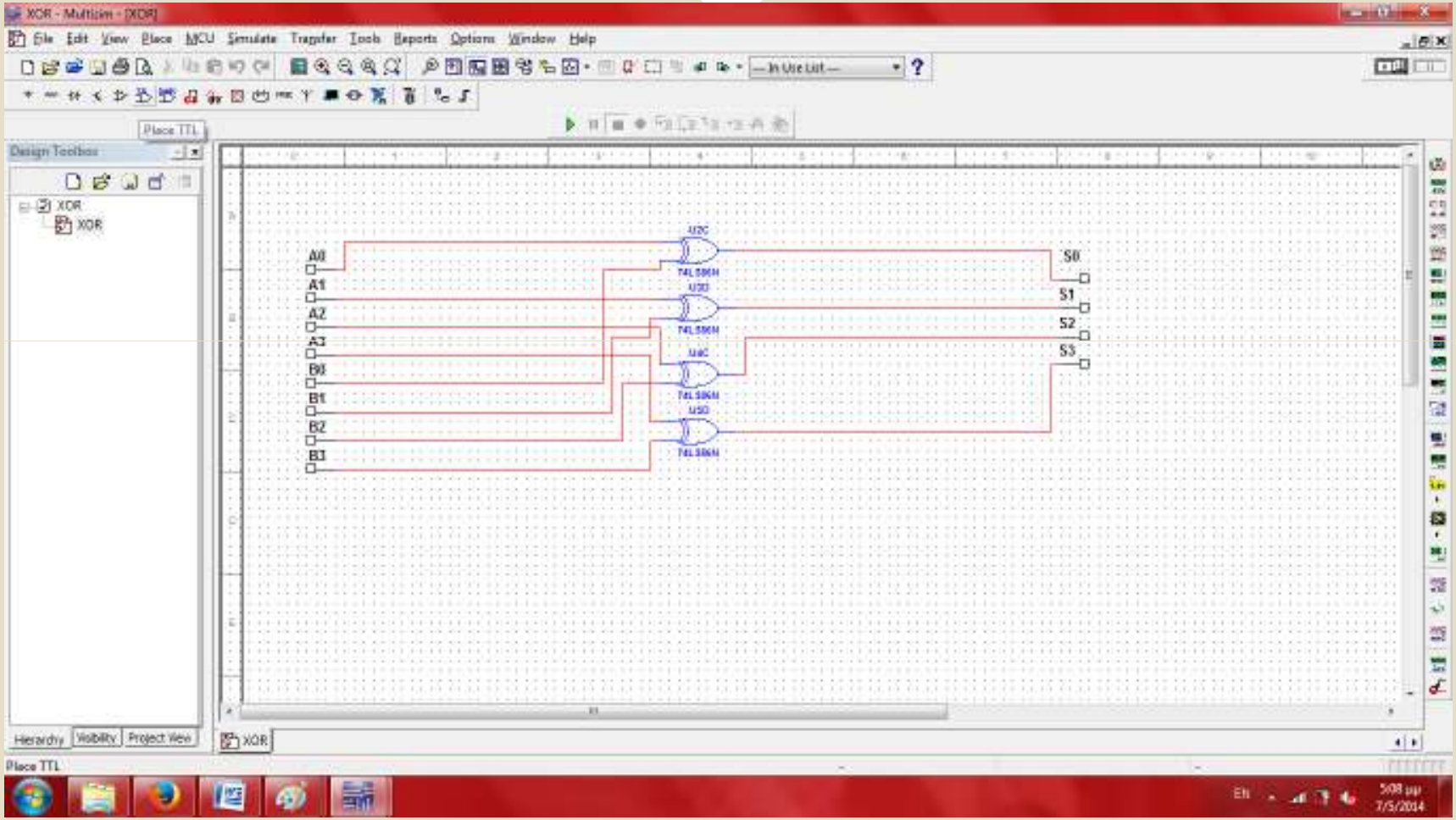
Πύλη XOR

- Η πύλη **XOR** εκτελεί την λογική πράξη XOR (αποκλειστικό Η’) μεταξύ των εισόδων της.

- Ο πίνακας αληθείας της λογικής πύλης XOR φαίνεται στο εξής σχήμα:

Είσοδοι		Έξοδος
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Πύλη XOR


XOR - Multisim - [XOR]
File Edit View Place MCU Simulate Transfer Tools Reports Options Window Help
Design Toolbox
XOR
XOR
A0
A1
A2
A3
B0
B1
B2
B3
S0
S1
S2
S3
74LS86N
Hierarchy
Visibility
Project View
Place TTL

Πύλη AND

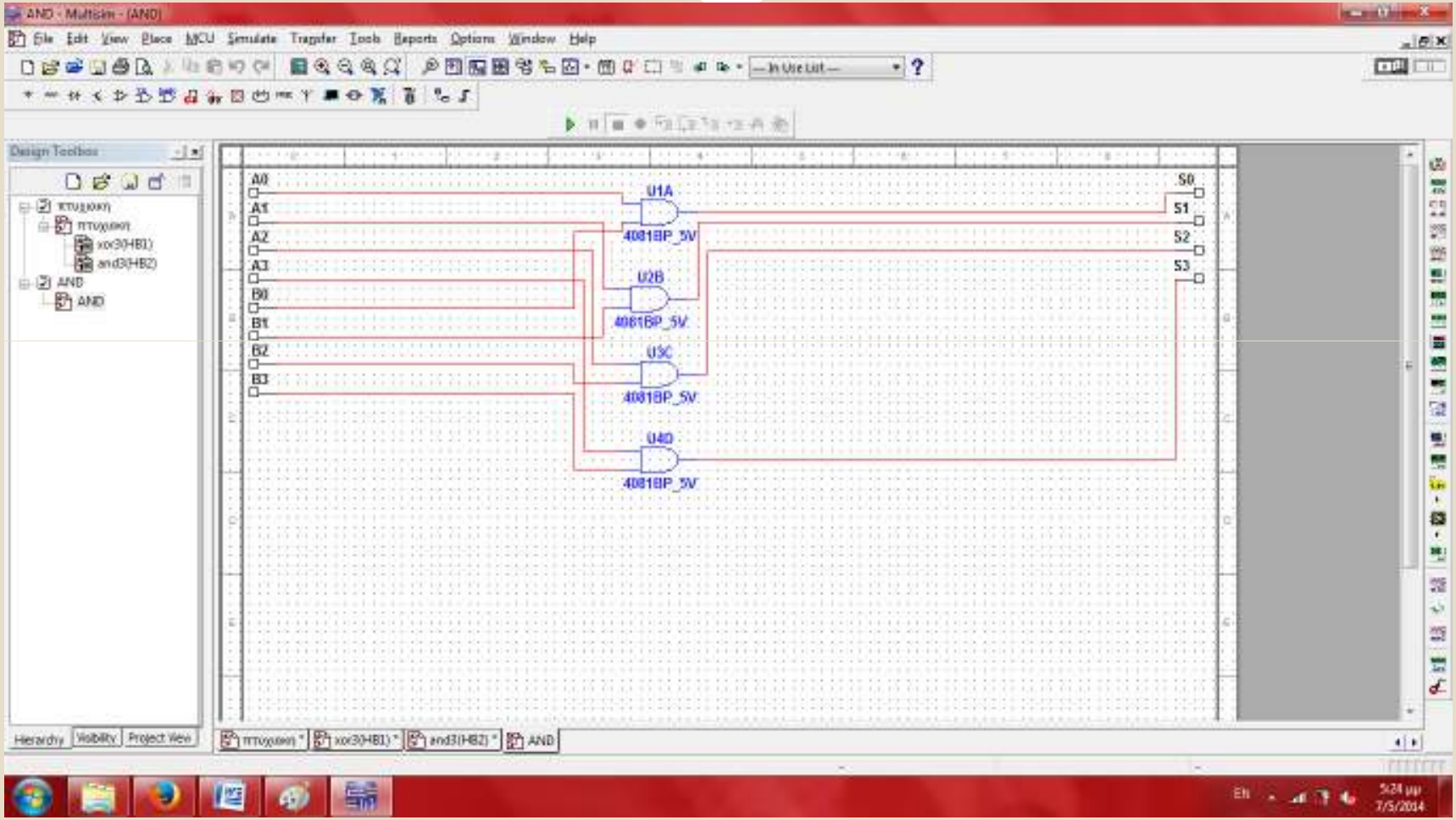
- Η πύλη **AND** εκτελεί την λογική πράξη AND (ΚΑΙ) μεταξύ των εισόδων της.

$$c = a*b$$

- Ο πίνακας αληθείας της λογικής πύλης AND φαίνεται στο εξής σχήμα:

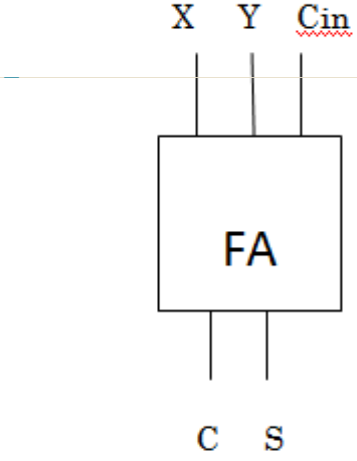
Είσοδοι		Έξοδος
A	B	A ΚΑΙ B
0	0	0
0	1	0
1	0	0
1	1	1

Πύλη AND


AND - Multisim - (AND)
File Edit View Place MCU Simulate Transfer Tools Reports Options Window Help
Design Toolbox
A0
A1
A2
A3
B0
B1
B2
B3
U1A
U2B
U3C
U4D
4081BP_5V
S0
S1
S2
S3
Hierarchy
Visibility
Project View
AND

Πλήρης αθροιστής

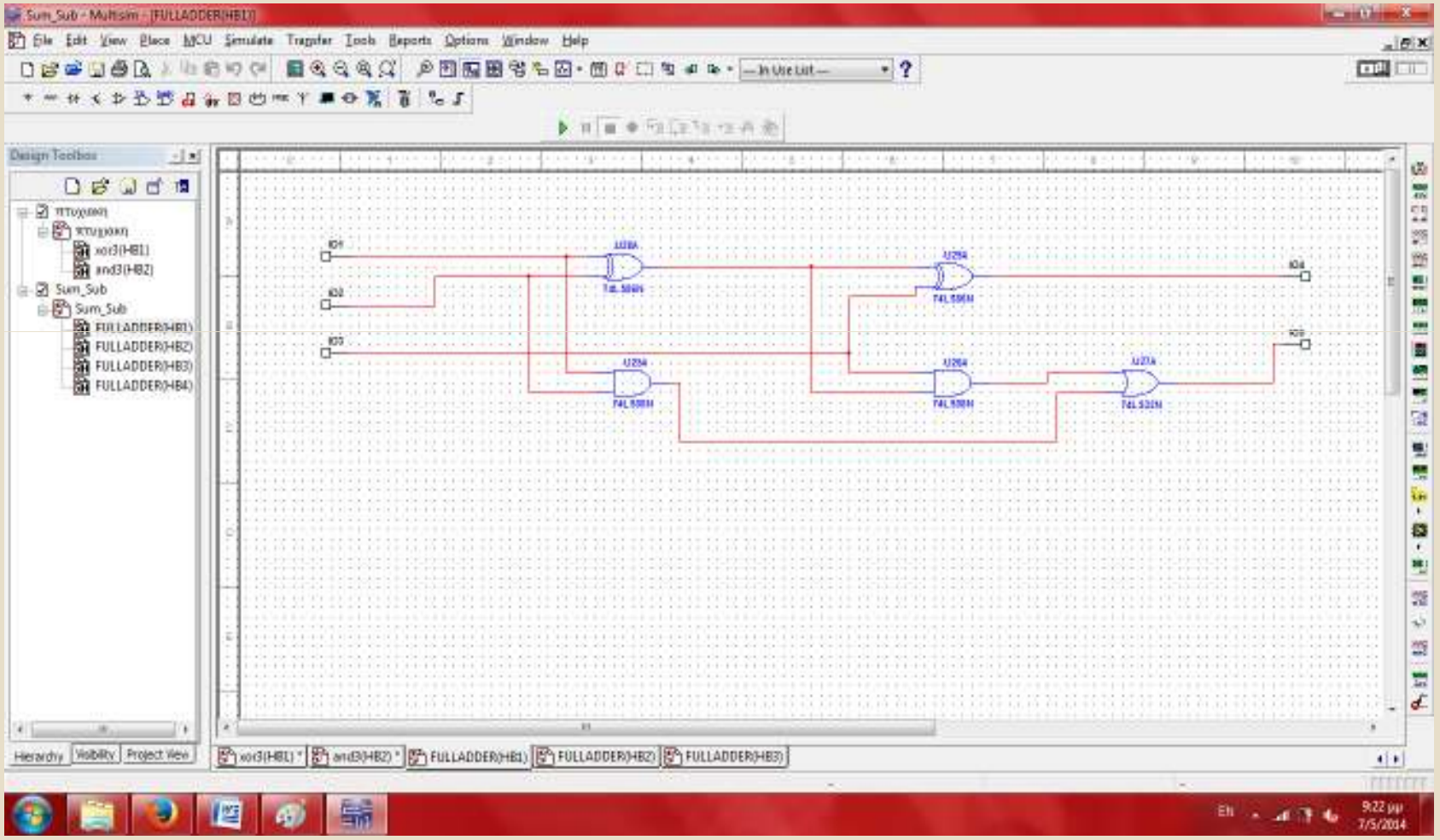
- Ο πλήρης αθροιστής είναι ένα συνδυαστικό κύκλωμα που εκτελεί την πρόσθεση δύο δυαδικών ψηφίων λαμβάνοντας υπόψη και την ύπαρξη κρατουμένου προηγούμενης τάξης.
- Το παρακάτω block διάγραμμα αναφέρεται σε έναν πλήρη 2-bits.



(σχήμα 2.11)Πλήρης αθροιστής 2-bits

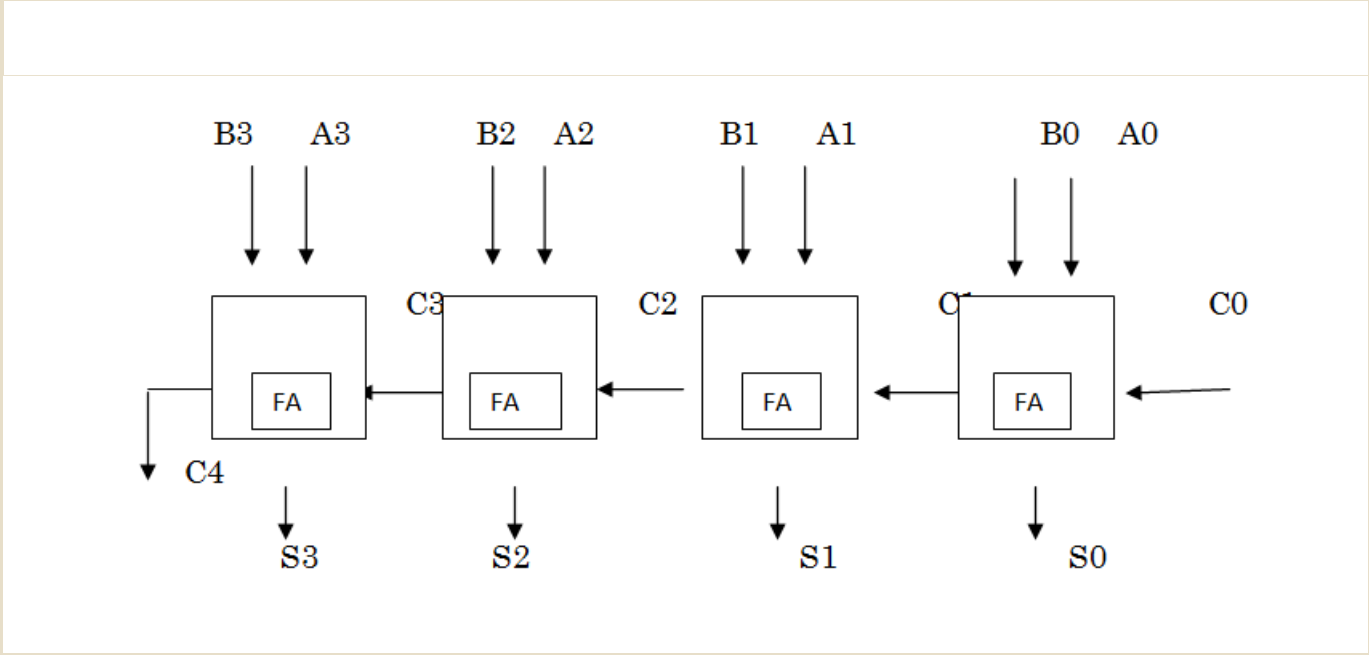
Οι δύο είσοδοι x,y παριστάνουν τα bit της ίδιας τάξης που θα προστεθούν ενώ το Cin παριστάνει το κρατούμενο από την προηγούμενη αμέσως λιγότερη σημαντική θέση. Οι έξοδοι S και C παριστάνουν το αποτέλεσμα και το πιθανό κρατούμενο αντίστοιχα.

Πλήρης Αθροιστής



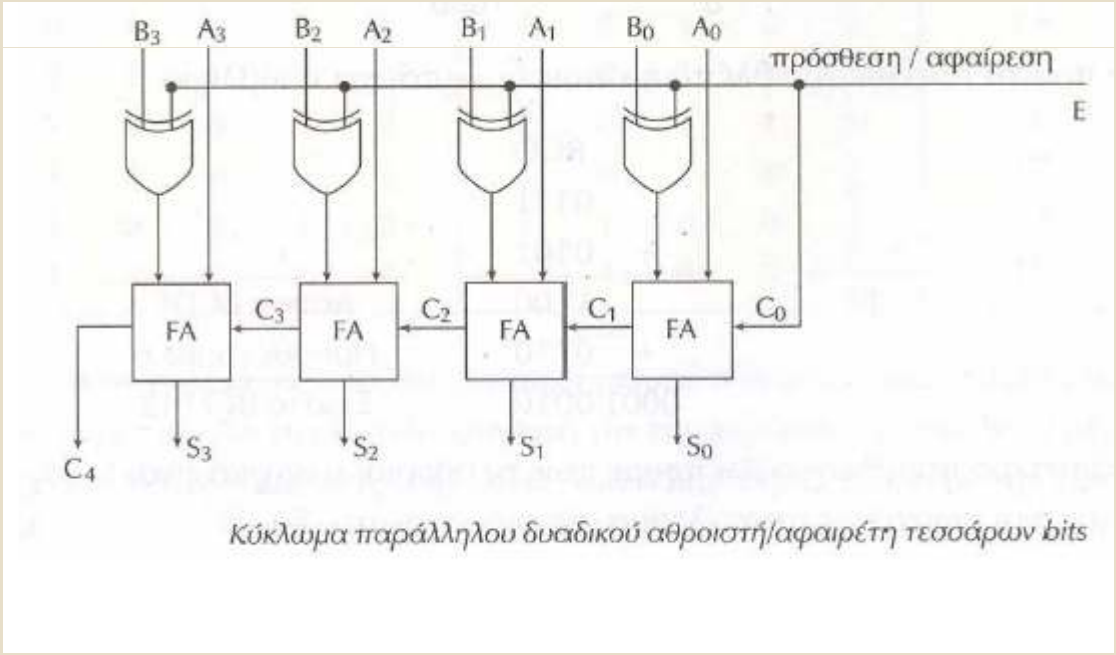
Πλήρης Αθροιστής 4-bit

- Για την πρόσθεση αριθμών των τεσσάρων bits όπου και χρειαζόμαστε εμείς χρησιμοποιούμε το κύκλωμα του παράλληλου πλήρη αθροιστή με διάδοση κρατουμένου. Ο αθροιστής αυτός προσθέτει δύο ψηφιολέξεις A3A2A1A0 και B3B2B1B0.Αποτελείται από τέσσερις πλήρεις αθροιστές που ο καθένας αθροίζει δύο bits.



Δυαδικός Αφαιρέτης

- Η πράξη της αφαίρεσης μπορεί να γίνει με αθροιστές χρησιμοποιώντας το συμπλήρωμα ως προς 2 του αφαιρετέου. Με τον τρόπο αυτό μπορούμε να χρησιμοποιήσουμε το ίδιο κύκλωμα τόσο για την πράξη της πρόσθεσης όσο και για την πράξη της αφαίρεσης.

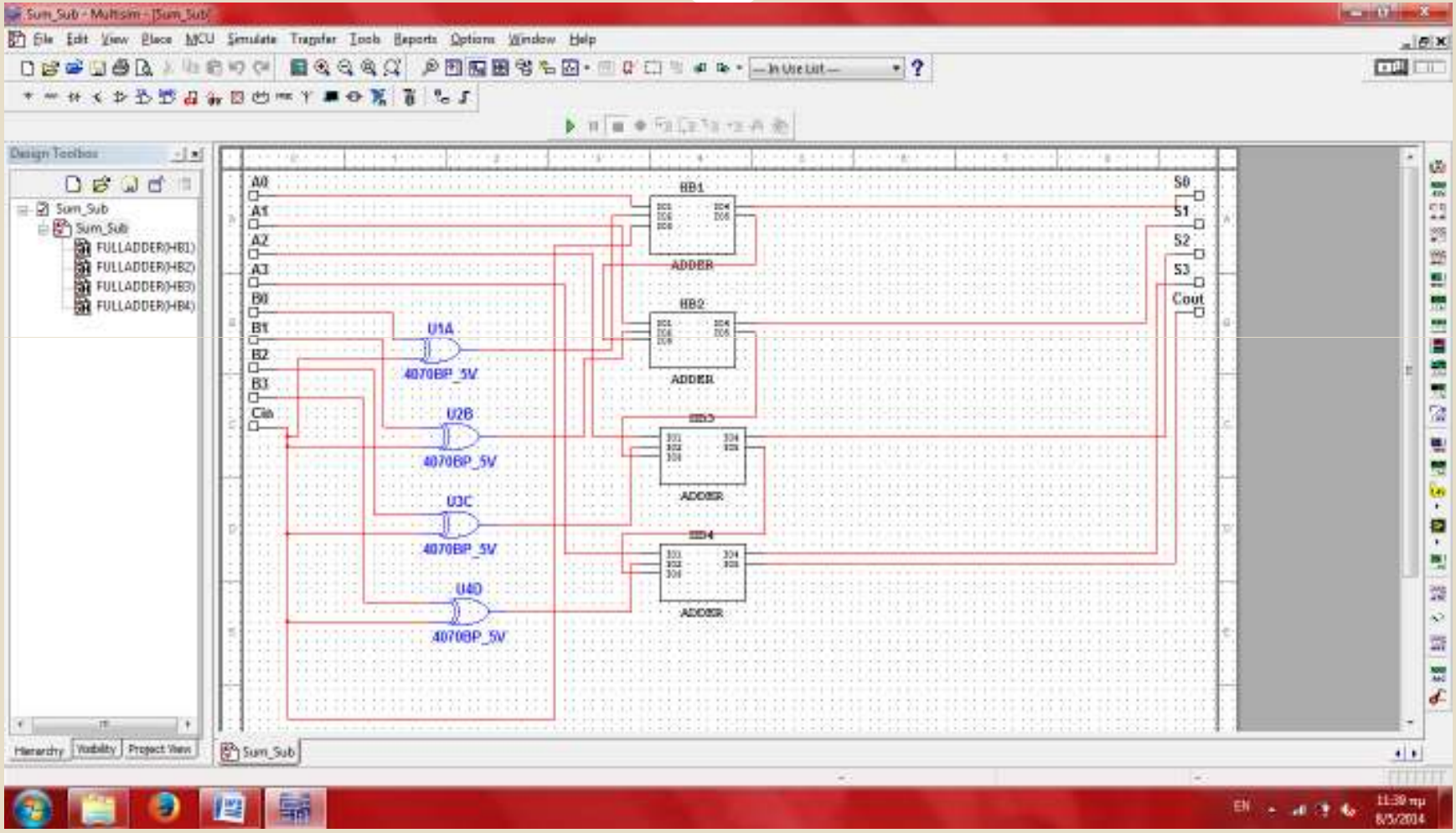


Κύκλωμα παράλληλου δυαδικού αθροιστή/αφαιρέτη τεσσάρων bits

Κύκλωμα Πρόσθεσης/Αφαίρεσης

- Το τελικό κύκλωμα θα υλοποιεί τη πράξη της πρόσθεσης ή της αφαίρεσης ανάλογα με την τιμή που θα δίνουμε εμείς στο κρατούμενο εισόδου. Συγκεκριμένα όταν το Cin θα παίρνει την τιμή 0 θα εκτελεί την πράξη της πρόσθεσης και αντίστοιχα όταν τι Cin θα παίρνει την τιμή 1 θα εκτελεί την πράξη της αφαίρεσης.

Κύκλωμα Πρόσθεσης/Αφαίρεσης



ALU

- Αυτά τα 3 κυκλώματα που σχεδιάσαμε θα αποτελέσουν και την ALU.
- Θα δημιουργήσουμε μια νέα βαθμίδα με όνομα ALU η οποία θα έχει 10 εισόδους (A0,A1,A2,A3,B0,B1,B2,B3,M0,M1) και 5 εξόδους (S0,S1,S2,S3,Cout). Το A0….B3 αποτελούν τις δύο ψηφιολέξεις μας και τα M0 M1 είναι οι είσοδοι όπου ανάλογα με τις τιμές τους θα εκτελείται και η αντίστοιχη πράξη. Τα αποτελέσματα θα εμφανίζονται στην έξοδο S0..S3 και το Cout θα εμφανίζει το πιθανό κρατούμενο που θα προκύψει από την πράξη της πρόσθεσης ή της αφαίρεσης.

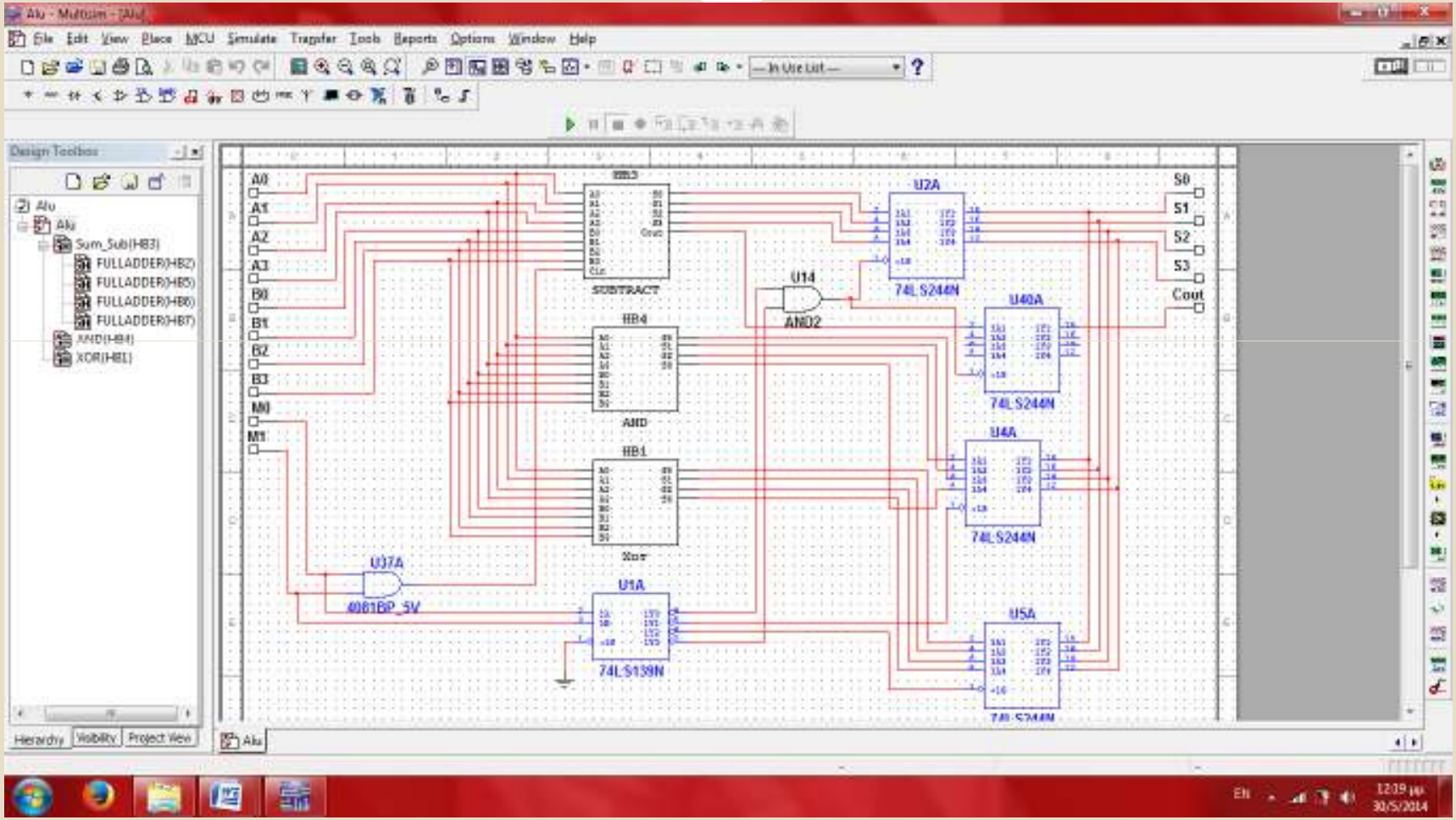
ALU

- Αφού συνδεθούν σωστά οι είσοδοι της ALU θα πρέπει να τοποθετήσουμε μια πύλη AND. Η πύλη αυτή μας εξυπηρετεί στην επιλογή πράξης.
- Απαραίτητο για την σωστή λειτουργία της ALU είναι και ένας αποκωδικοποιητής 2:4.
- Εμείς συγκεκριμένα θα χρησιμοποιήσουμε τον 74ls139

Απομονωτές/buffers

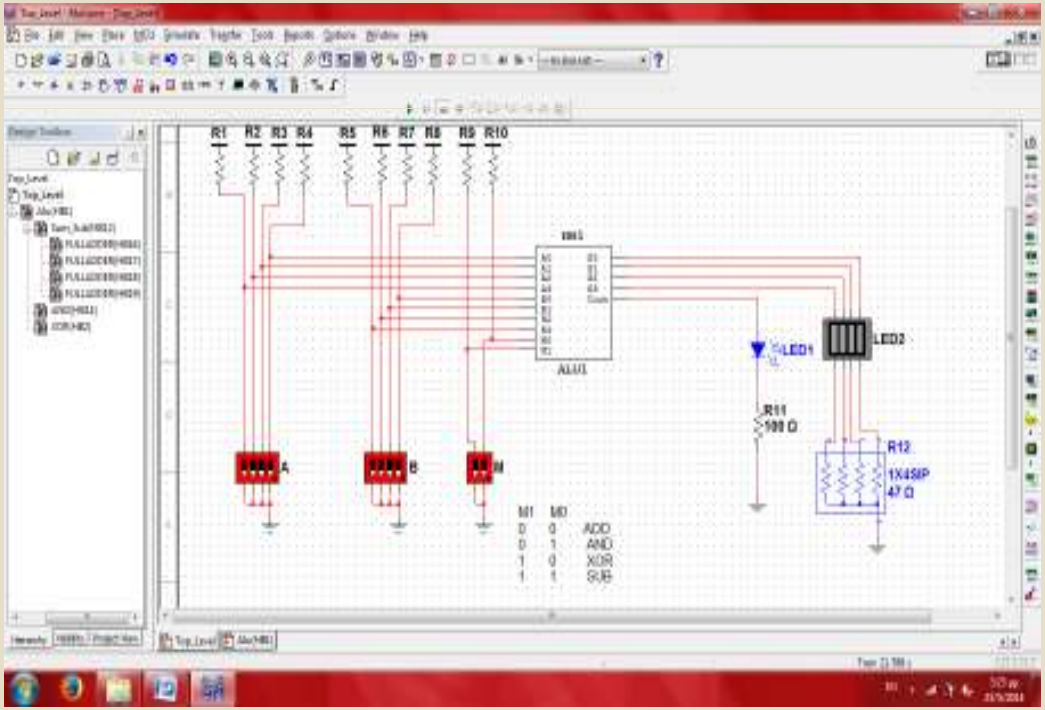
- Λόγω του ότι η αριθμητική λογική μονάδα θα εκτελεί παραπάνω από μία πράξεις είναι απαραίτητοι 3 απομονωτές (buffers). Συνδέοντας τις εξόδους του αποκωδικοποιητή με τις γραμμές ελέγχου των buffer οδηγούμε στην έξοδο της ALU ένα αποτέλεσμα κάθε φορά.

Κύκλωμα ALU


A0
A1
A2
A3
B0
B1
B2
B3
M0
M1
HB3
SUBTRACT
HB4
AND
HB1
Xor
U37A
4081BP_5V
U1A
74LS139N
U14
AND2
U2A
74LS244N
U40A
74LS244N
U4A
74LS244N
U5A
S0
S1
S2
S3
Cout
Design Toolbox
Alu
Sum_Sub(HB3)
FULLADDER(HB2)
FULLADDER(HB5)
FULLADDER(HB6)
FULLADDER(HB7)
XOR(HB1)
Hierarchy
Visibility
Project View

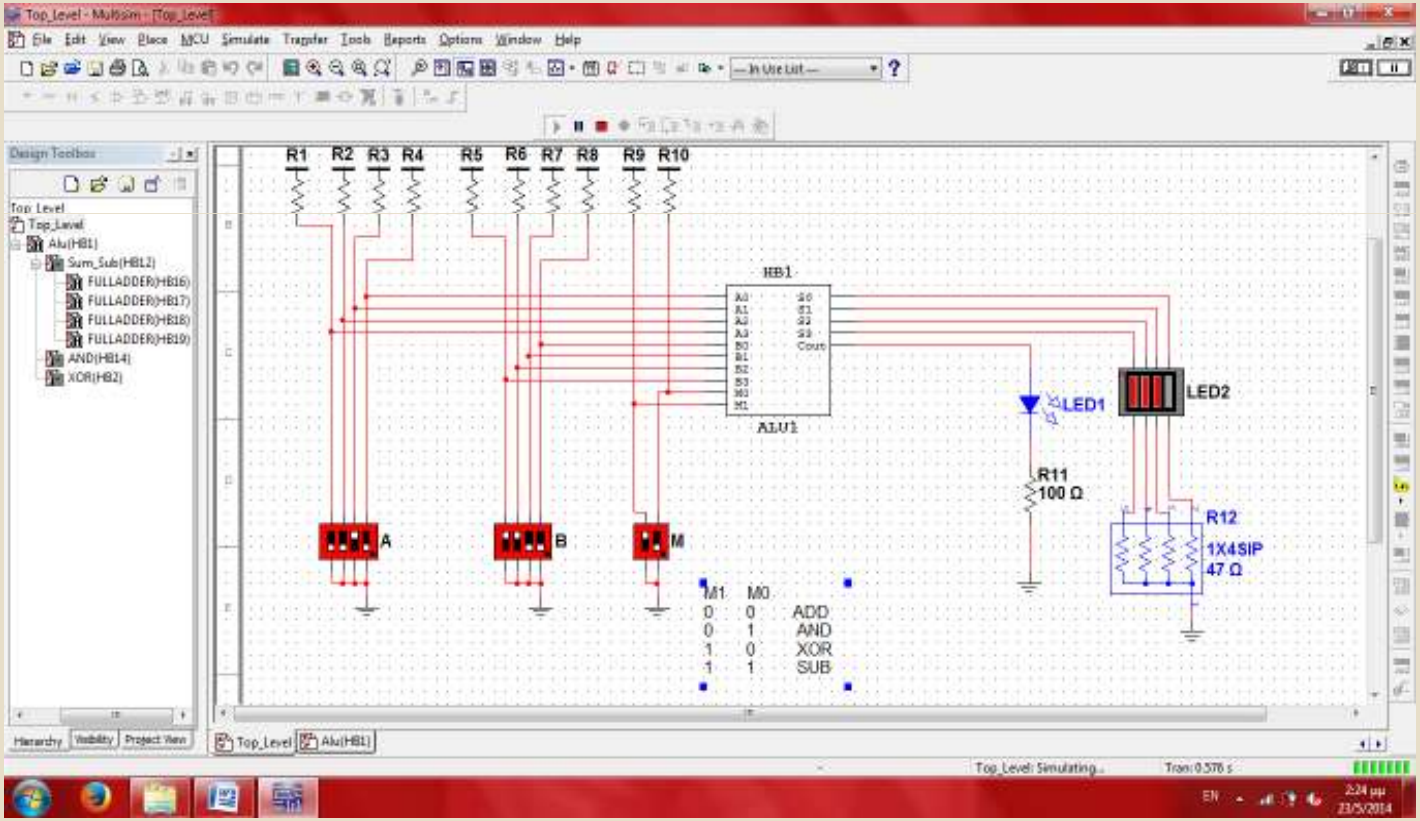
Τελικό κύκλωμα

- Εφόσον σχεδιάσαμε σε αυτό το επίπεδο και την αριθμητική λογική μονάδα(ALU) είμαστε στο στάδιο στο οποίο μπορούμε να δούμε με παραδείγματα πως λειτουργεί το κύκλωμά μας. Η τελική του μορφή είναι η παρακάτω (top level).



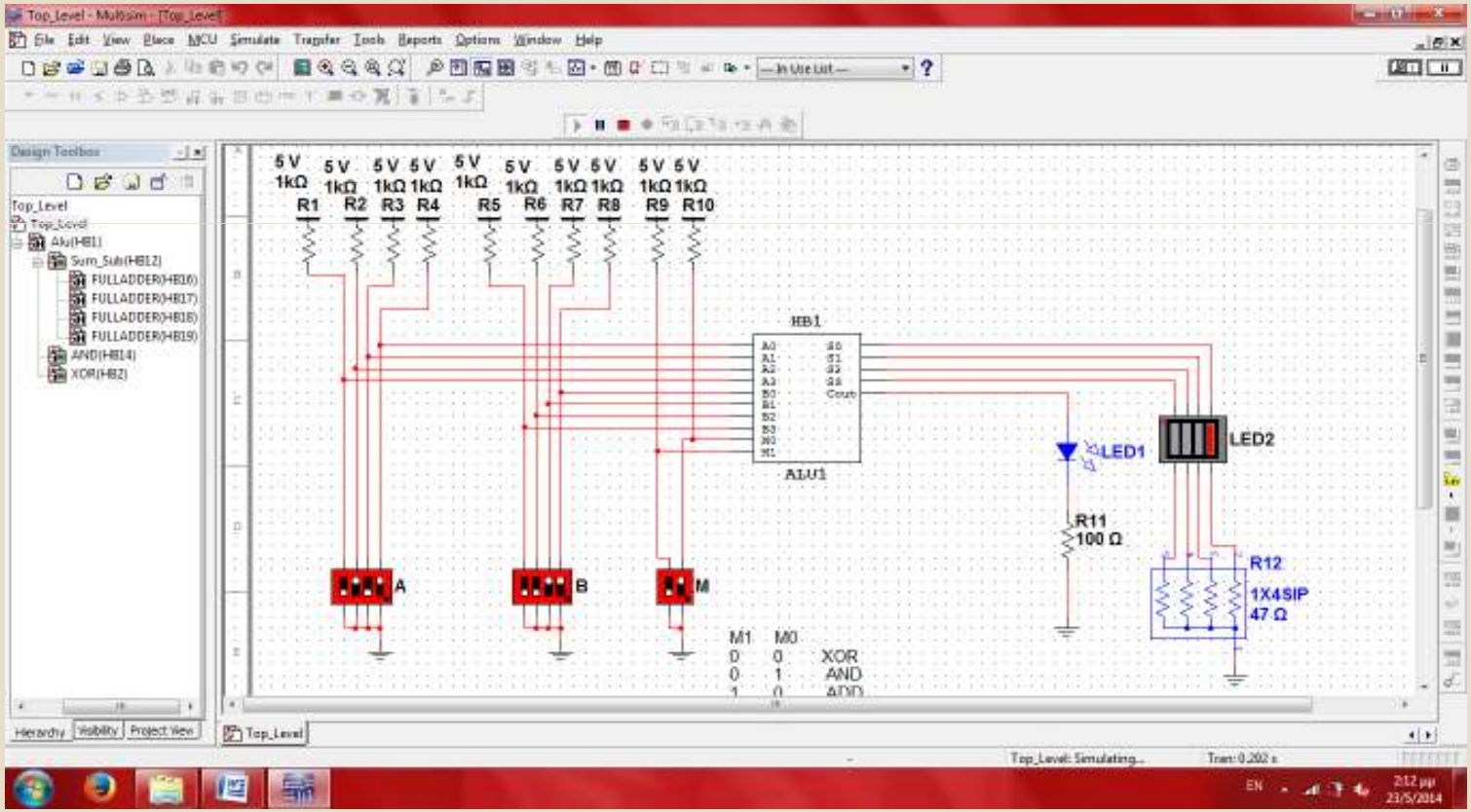
Παράδειγμα XOR

- Παράδειγμα: Για την λογική πράξη XOR ανάμεσα στο «0010» και «1100»:



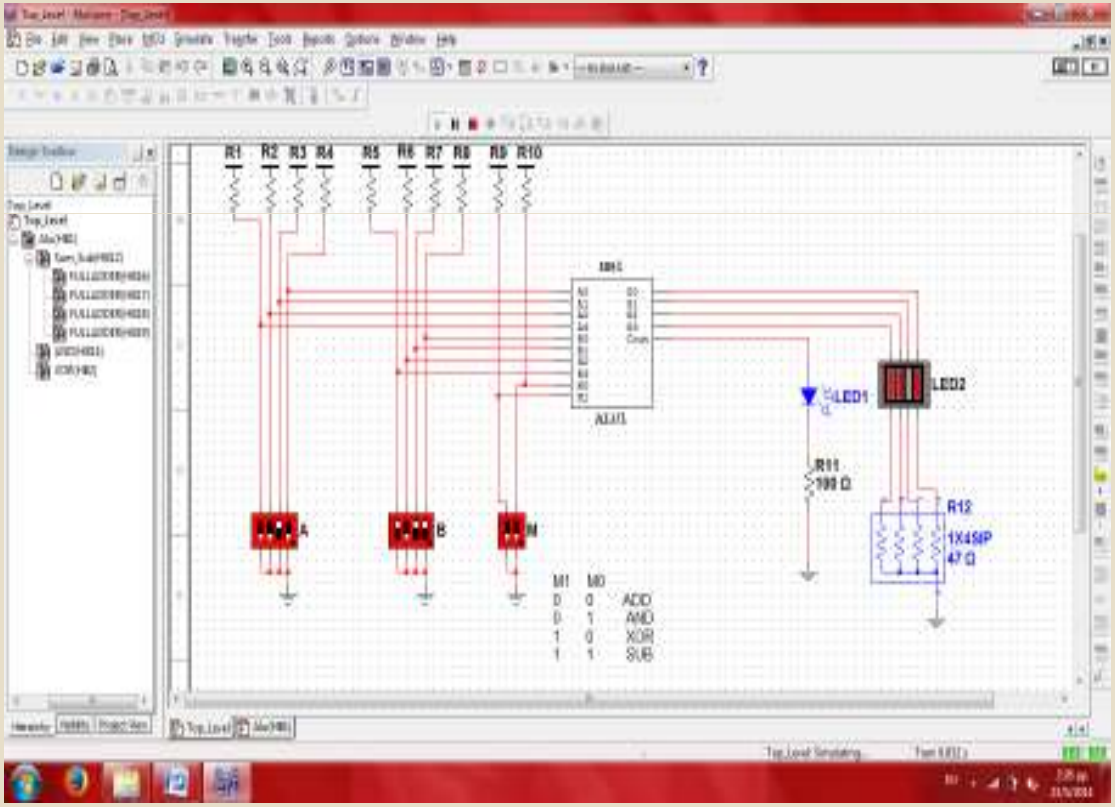
Παράδειγμα AND

- Για την λογική πράξη AND ανάμεσα στους αριθμούς «0101» και «0011»:



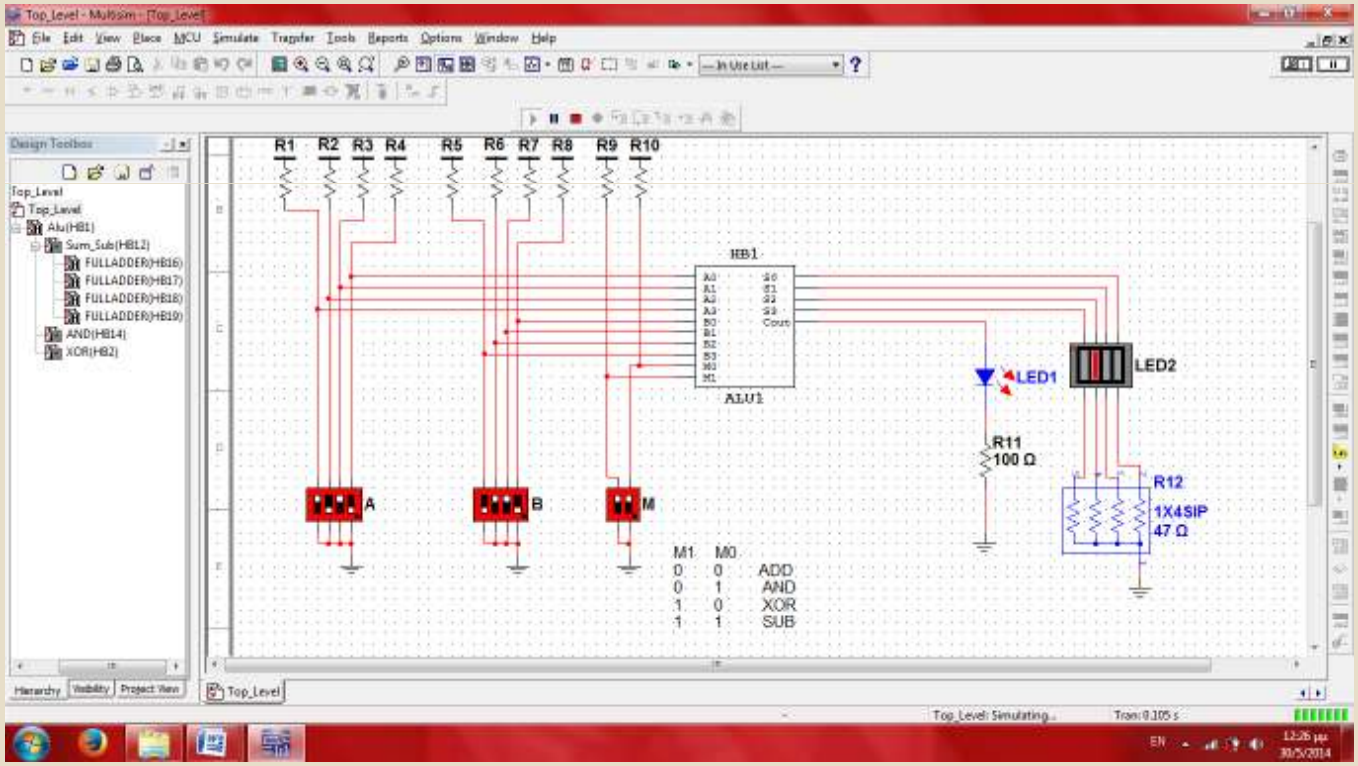
Παράδειγμα Πρόσθεσης

- Για την πρόσθεση ανάμεσα στους αριθμούς «0010» και «1011»:



Παράδειγμα Αφαίρεσης

- Για την πράξη της αφαίρεσης ανάμεσα στους αριθμούς «1010» και «0110»



Συμπεράσματα

- Βάση της παρούσας εργασίας αποτέλεσε το λογισμικό Multisim, ένα περιβάλλον λογισμικού που επιτρέπει τη σχεδίαση αλλά και την προσομοίωση της λειτουργίας των ηλεκτρονικών κυκλωμάτων.

- Έτσι, αναφερθήκαμε:
- στη σχεδίαση και προσομοίωση της παράλληλης αριθμητικής λογικής μονάδας (ALU)
- στις λογικές πύλες XOR και AND,
- στον πλήρη αθροιστή,
- στη διάδοση κρατούμενων.
- δυαδικό αφαιρετή.

- Κατ’ αυτόν τον τρόπο δόθηκε σχηματικά και ανά βήμα η δημιουργία του κυκλώματος, ενώ τρέχοντας το πρόγραμμα είδαμε πως μπορούμε να προχωρήσουμε στην προσομοίωση της λειτουργίας του κυκλώματος, με πραγματικές εισόδους και εξόδους. Η προσομοίωση δείχνει ότι το κύκλωμα λειτουργεί κανονικά.