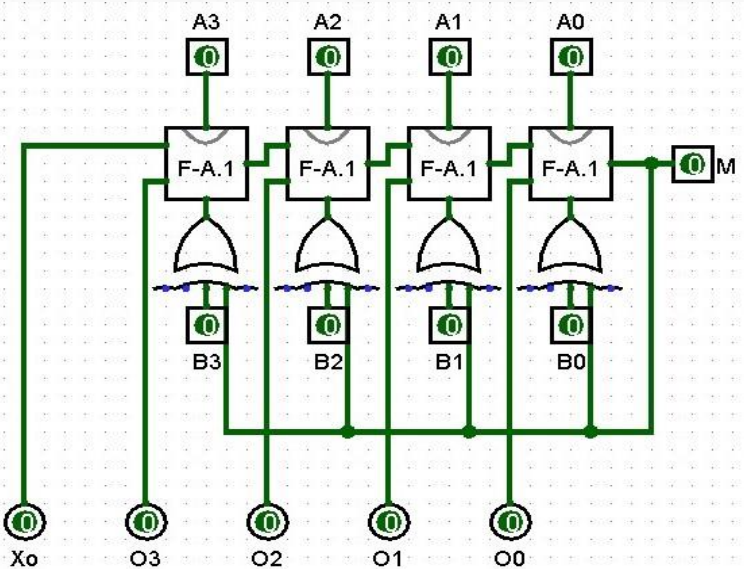


1 ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ ΑΘΡΟΙΣΤΗ-ΑΦΑΙΡΕΤΗ

- 1. Εισάγουμε δύο αριθμούς A και B σε δυαδική μορφή.
- 2. Διαλέγουμε την πράξη από την είσοδο M όπου M=0: πρόσθεση, ή M=1: αφαίρεση.

- 3. α) M=0: Φέρουμε ήδη πως να εκτελούμε την πρόσθεση δύο αριθμών από τη σχεδίαση του πλήρους αθροιστή 4 bit
- β) M=1: Θα χρησιμοποιήσουμε την μέθοδο του συμπληρώματος ως προς 2* για να εκτελέσουμε την πράξη της αφαίρεσης σαν να ήταν πρόσθεση



3

* ΑΦΑΙΡΕΣΗ ΜΕ ΣΥΜΠΛΗΡΩΜΑ ΟΣ ΠΡΟΣ 2

$(A-B)_2 = A + 2\text{'s complement}(B)$

$2\text{'s complement}(B) = 1\text{'s complement}(B) + 1$

ΑΡΑ

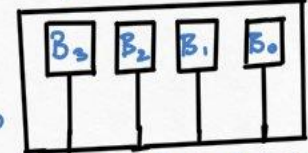
Για να φέρουμε το B σε μορφή συμπληρώματος ως προς 2 πρέπει:

- 1. Να το φέρουμε σε μορφή συμπληρώματος ως προς 1. Δηλαδή στο κώδική μας θέλουμε τα 4 bit εισόδου του B να αντιστρέφονται όταν το M=1, δηλαδή→

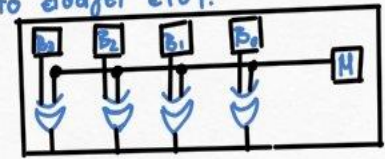
M	B	B _{τελικό}
0	0	0
0	1	1
1	0	1
1	1	0

ποτατιζούμε πως το $B_{τελικό} = M \oplus B$, έτσι για το βήμα 1, αρκεί να χρησιμοποιήσουμε πύλες XOR στα ψηφία του B και το M

Αν ο πλήρης αθροιστής 4 bit εισάγει το B έτσι:

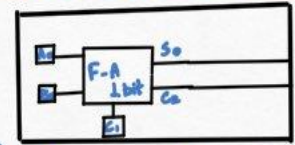


Τότε ο αθροιστής/αφαιρέτης το εισάγει έτσι:

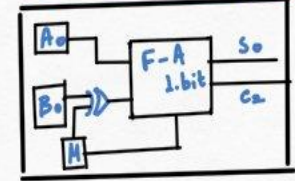


- 2. Να προσθέσουμε 1. Αυτό μπορεί να γίνει αλλά αξιολογώντας την είσοδο C1 του πλήρους αθροιστή 4 bit. Η C1 αντιπροσωπεύει το υπόλοιπο της προηγούμενης πρόσθεσης το οποίο δεν μας χρειάζεται γιατί ο αθροιστής/αφαιρέτης λειτουργεί αυτότελος. Έτσι θέλουμε όταν M=1, C1=1 (ενώ όταν εκτελούμε πρόσθεση το C1 θέλουμε να είναι 0)

πλήρης αθροιστής 4 bit



αθροιστής/αφαιρέτης



<div>2</div> <div>ΠΡΑΞΕΙΣ ΜΕ ΤΟ ΧΕΡΙ</div> <div>ΠΡΟΣΟΜΟΙΩΣΗ ΣΤΟ LOGICIM</div>	<div>+</div> <div>M=0</div> <div>-</div> <div>M=1</div>	<div>A=1011</div> <div>B=0101</div>	<div>A=0110</div> <div>B=0100</div>
		<div>Carries</div> <div> <div>A</div> <div>B</div> <div> <div>1011</div> <div>+ 0101</div> <div>10000</div> </div> <div>X₀ Output(0)</div> </div>	<div>Carries</div> <div> <div>A</div> <div>B</div> <div> <div>0110</div> <div>+ 0100</div> <div>01010</div> </div> <div>X₀ Output(0)</div> </div>
		<div>B τετλ.ν.0 =</div> <div>= B' + 1 =</div> <div>= 1010</div> <div>+ 0001</div> <div>1011</div>	<div>B τετλ.ν.0 =</div> <div>= B' + 1 =</div> <div>= 0101</div> <div>+ 0001</div> <div>1000</div>
		<div>Carries</div> <div> <div>A</div> <div>B</div> <div> <div>1011</div> <div>+ 1011</div> <div>10110</div> </div> <div>X₀ Output(0)</div> </div>	<div>Carries</div> <div> <div>A</div> <div>B</div> <div> <div>0110</div> <div>+ 1100</div> <div>10010</div> </div> <div>X₀ Output(0)</div> </div>
	+		
	-		

4

Υπερχεινιση: στο κύκλωμά μας συμβαίνει όταν η προσθαφαιρεση των 4 bit αριθμών μας οδηγεί σε αριθμό που καταλαμβάνει 5 bit.

- Θεωρώντας πως οι Α και Β είναι μη-προσημασμένοι, η υπερχεινιση μπορεί να συμβεί μόνο όταν το υπόλοιπο της πρόσθεσης των τελευταίων ψηφίων ($A_3 + B_3$) είναι 1. Αυτή την περίπτωση την ελέγχουμε ήδη με το X_0 . Άρα γνωρίζουμε πως αν το X_0 είναι 1 τότε υπάρχει υπερχεινιση, ενώ αν είναι 0 δεν υπάρχει

Σημείωση: Ξεκίνησα την εργασία με διαφορετικά ονόματα στις πύλες από αυτά που είχατε στο pdf και είναι αδύνατο να τα αλλάξω.

το f μετατράπηκε σε 0 (από το output)

το Z μετατράπηκε σε X_0 (όπου $x=C$ ή $x=B$ και 0 από το output)

τα A, B, M παρέμειναν ίδια.