

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ**  
**ΤΜΗΜΑ ΜΗΧ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ**  
**ΤΟΜΕΑΣ ΥΛΙΚΟΥ ΚΑΙ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ**  
**Εργαστήριο Ηλεκτρονικών & Λογικού Σχεδιασμού**

**ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ  
ΛΟΓΙΚΟΥ ΣΧΕΔΙΑΣΜΟΥ**

**Γ.Φ. ΑΛΕΞΙΟΥ - Ε.Ζ. ΨΑΡΑΚΗΣ**

**Πάτρα 2018**

**Ακαδ. Έτος : 2020-21**

**Εαρινό Εξάμηνο : 2020-21**

**Όνοματεπώνυμο: Λουκάς Ιωάννης Προδρόμου  
Α.Μ. : 1084590**

**Όνομ/νυμο Συνεργάτη: Ανδρέας Κατσαρός  
Α.Μ. : 1084522**

**Όνομ/νυμο Συνεργάτη: Νίκος Χουσελάς  
Α.Μ. : 1067432**

**Τμήμα : -----**

**Ασκηση:\_\_ ΑΣΚΗΣΗ 3 \_\_**

***Αναλυτική Βαθμολογία Πρακτικού Μέρους:***

	<b>Βαθμός Αριθ.</b>	<b>Αξιολογητή Ολογραφ.</b>	<b>Υπογραφή Αξιολογητή</b>
<i>Ασκηση 1</i>			
<i>Ασκηση 3</i>			
<i>Ασκηση 5</i>			
<i>Ασκηση 6</i>			
<i>Ασκηση 7</i>			
<b>M.O.</b>			

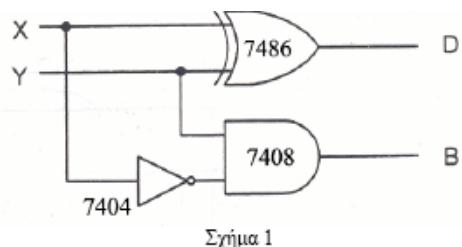
## ΑΣΚΗΣΗ 3

### ΗΜΕΡΟΜΗΝΙΑ:

**ΘΕΜΑ: ΗΜΙΑΦΑΙΡΕΤΗΣ, ΑΦΑΙΡΕΤΗΣ - HALF, FULL SUBTRACTOR** (Morris Mano, Κεφ. 1- 4).

### ΘΕΩΡΙΑ

**ΗΜΙΑΦΑΙΡΕΤΗΣ (HALF SUBTRACTOR, HS).** Ο HS είναι ένα συνδυαστικό κύκλωμα δύο εισόδων **X**, **Y** και δύο εξόδων **D** (Difference) και **B** (Borrow). Το κύκλωμα αυτό δίνει στην έξοδο του **D** τη διαφορά (**X**- **Y**) των δύο εισόδων και στο **B** μας λέει αν χρειαστήκαμε να δανειστούμε μια μονάδα για την αφαίρεση. Το κύκλωμα και οι λογικές συναρτήσεις των δύο εξόδων ακολουθούν.



$$D = X' Y + X Y' = X \text{ xor } Y$$

$$B = X' Y$$

### ΔΙΑΛΙΚΑΣΙΑ

- Σχεδιάστε στο simulator, ένα συνδυαστικό κύκλωμα που δέχεται ως **είσοδο** ένα προσημασμένο αριθμό των 4-bits σε αναπαράσταση "συμπληρώματος ως προς 2", **X** =  $X_3X_2X_1X_0$  και δίνει ως **έξοδο** την αριθμητική τιμή της εισόδου σε αναπαράσταση **"Πρόσημου-Μέτρου"** (Sign-Magnitude), **Y** =  $sY_2Y_1Y_0$ .

**Ισοδύναμα**, το κύκλωμα πρέπει να έχει τις ακόλουθες ιδιότητες:

- Όταν  $X \geq 0$  ( $X_3=0$ ) τότε  $Y = X$
- Όταν  $X < 0$  ( $X_3=1$ ), το **Y** παριστάνει την Sign-Magnitude τιμή της εισόδου.

Δικαιολογείστε την απάντησή σας (πίνακας αληθείας, απλοποιήσεις, κλπ.) και δώστε τα ζητούμενα screenshots από τον simulator.

**Παρατηρήσεις:**

- Χρησιμοποιείστε ολοκληρωμένα της επιλογής σας, **της βιβλιοθήκης TTL**. Υπόδειξη: μετά από απλοποίηση το κύκλωμα δύναται να υλοποιηθεί μόνο με 3 ολοκληρωμένα (συνολικά 5 πύλες).
- Συνδέστε τις εισόδους,  $X_3X_2X_1X_0$ , σε pins εισόδου για να τους αποδώσετε τιμές και τις εξόδους,  $sY_2Y_1Y_0$ , σε LEDs για να βεβαιωθείτε ότι το κύκλωμα σας δουλεύει σωστά.
- Συνδέστε τις εξόδους KAI σε pins εξόδου, γιατί θα χρησιμοποιήσετε το κύκλωμα αυτό ως υποκύκλωμα στο Ερώτ. 3.B.

### ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ

δεκαδικό	X3	X2	X1	X0		s	Y2	Y1	Y0
0	0	0	0	0		0	0	0	0(ή 1000)
1	0	0	0	1		0	0	0	1
2	0	0	1	0		0	0	1	0
3	0	0	1	1		0	0	1	1
4	0	1	0	0		0	1	0	0
5	0	1	0	1		0	1	0	1
6	0	1	1	0		0	1	1	0
7	0	1	1	1		0	1	1	1
-8	1	0	0	0		-	-	-	-
-7	1	0	0	1		1	1	1	1
-6	1	0	1	0		1	1	1	0
-5	1	0	1	1		1	1	0	1
-4	1	1	0	0		1	1	0	0
-3	1	1	0	1		1	0	1	1
-2	1	1	1	0		1	0	1	0
-1	1	1	1	1		1	0	0	1

Το σύνολο των αριθμών που παριστάνονται σε μορφή προσημασμένου μέτρου είναι :  $[-(2^{n-1}-1), 2^{n-1}-1]$ .

Το σύνολο των αριθμών που παριστάνονται σε μορφή συμπληρώματος ως προς 2 είναι  $[-2^{n-1}, 2^{n-1}-1]$ .

Άρα το -8 δεν μπορεί να παρασταθεί σε μορφή προσημασμένου μέτρου διότι για την αναπαράστασή του χρειάζονται τουλάχιστον 5 bits, άρα η έξοδος του είναι ένας αδιάφορος όρος για το κύκλωμα.

S=x3

Y1=Σ(2,3,6,7,9,10,13,14)

Y2=Σ(4,5,6,7,9,10,11,12)

$Y_0=X_0$

Το -8 αποτελεί αδιάφορο όρο.

Από τον παραπάνω πίνακα και την υλοποίηση των συναρτήσεων σε χάρτη Karnaugh καταληγουμε στα εξής :

ΛΟΓΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ ΕΞΟΔΩΝ  $s, Y_2, Y_1, Y_0$

$S = X_3$

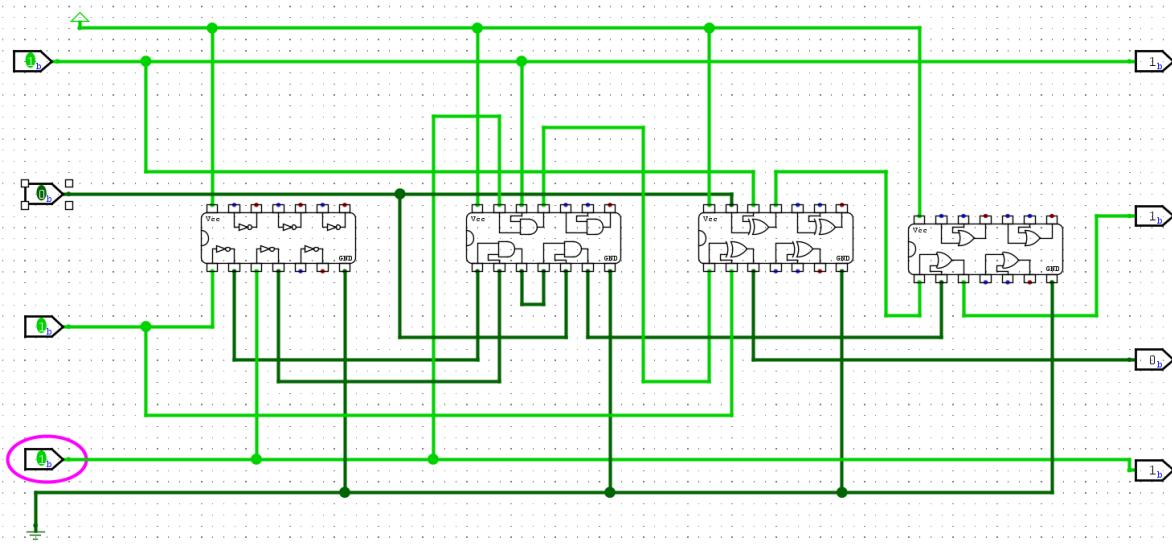
$$Y_2 = X_3X_2' + X_3'X_2 + X_2X_1'X_0' = (X_3 \text{ (XOR)} X_2) + X_2X_1'X_0'$$

$$Y_1 = X_3X_1'X_0 + X_3'X_1 + X_1X_0' = \dots = X_1 \text{ (XOR)} X_0X_3$$

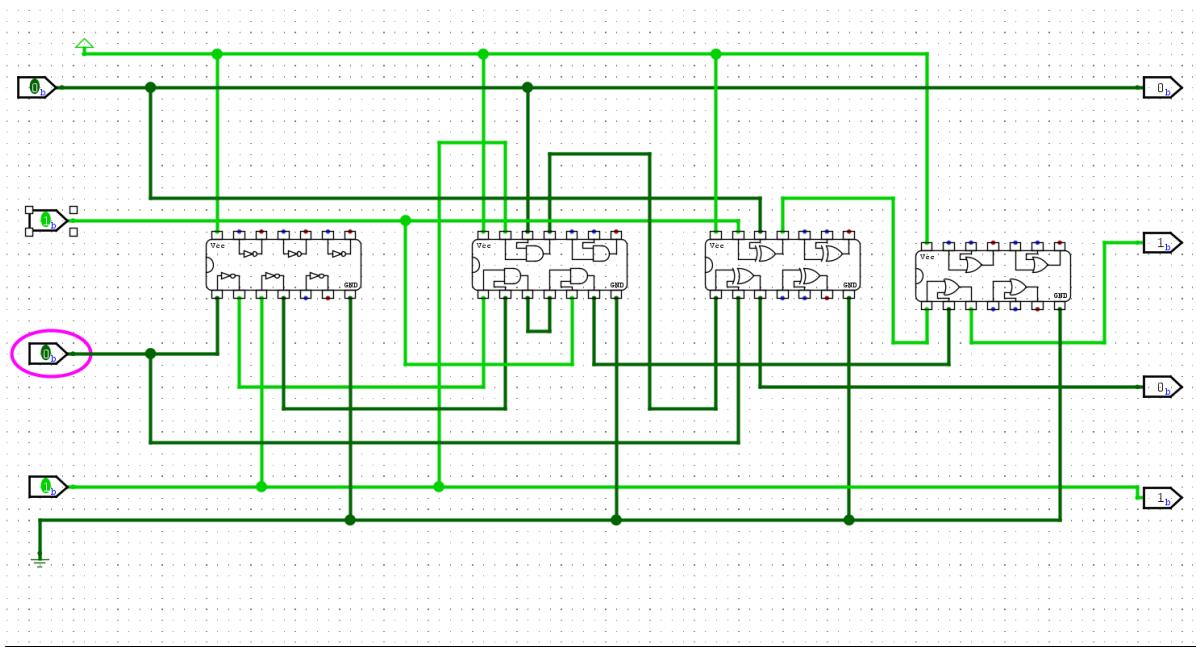
$$Y_0 = X_0$$

## ΚΥΚΛΩΜΑ

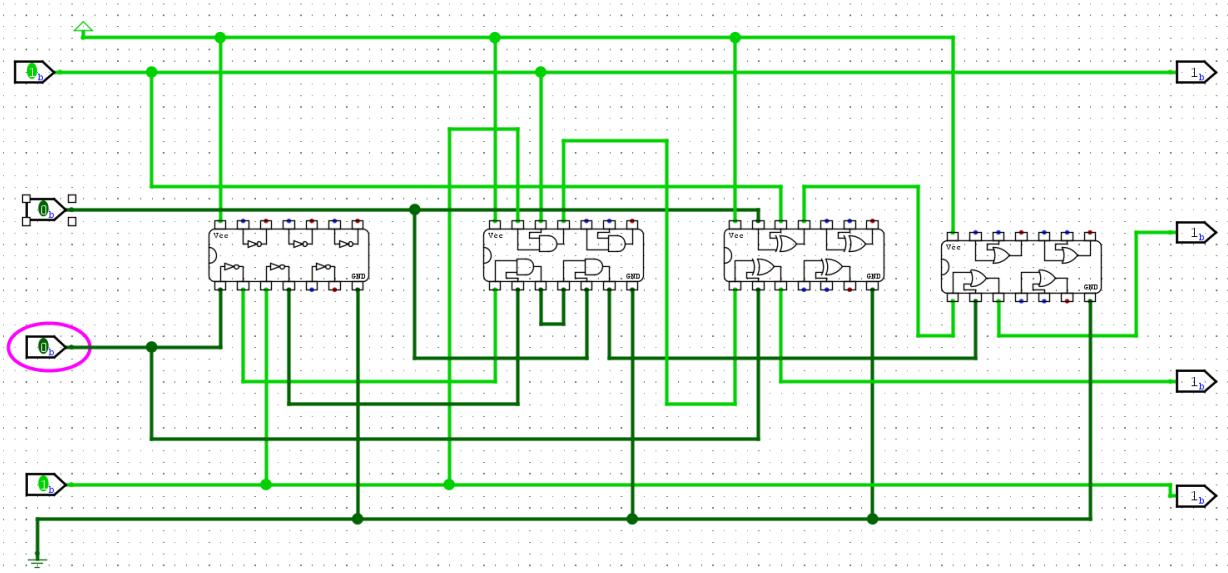
Για τυχαία είσοδο ίση με την τιμή 1011:



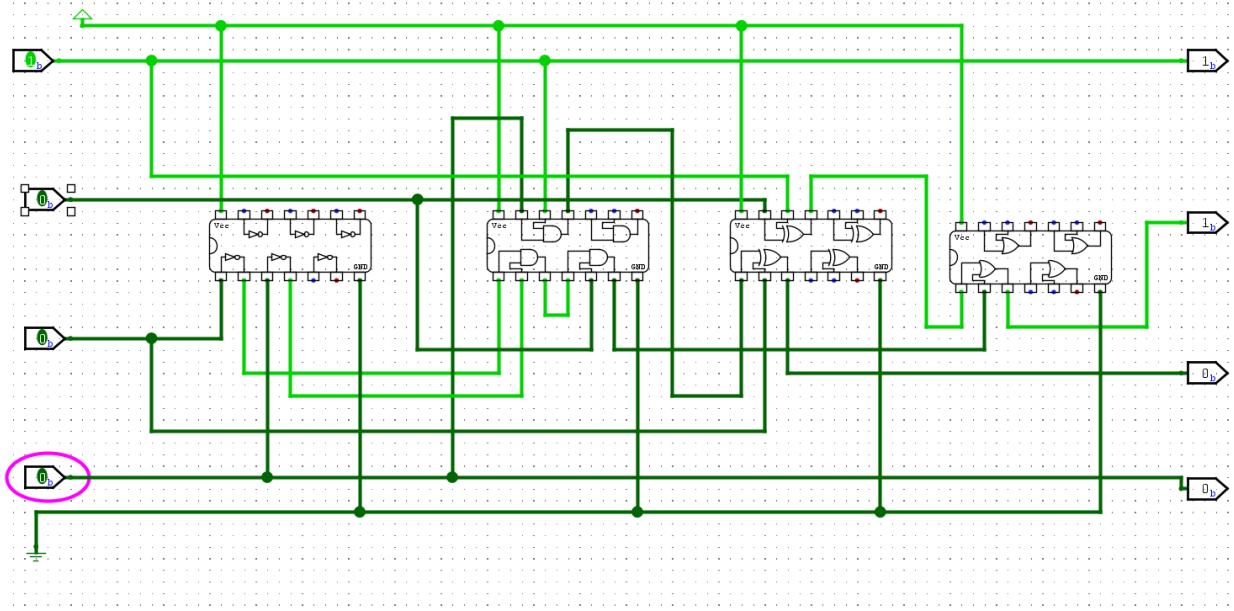
- Screenshot για την περίπτωση  $X = 5$



- Screenshot για την περίπτωση X = -7



- Screenshot για την περίπτωση X = -8



2. Ένας **Πλήρης Αφαιρέτης** (Full Subtractor, FS) είναι ένα συνδυαστικό κύκλωμα τριών εισόδων **X**, **Y**, **Bin** και δύο εξόδων **Bout**, **D**. Η έξοδος **Bout** είναι το πιθανό δανεισμένο bit και η έξοδος **D** είναι η διαφορά των τριών εισόδων (**X - Y - Bin**).

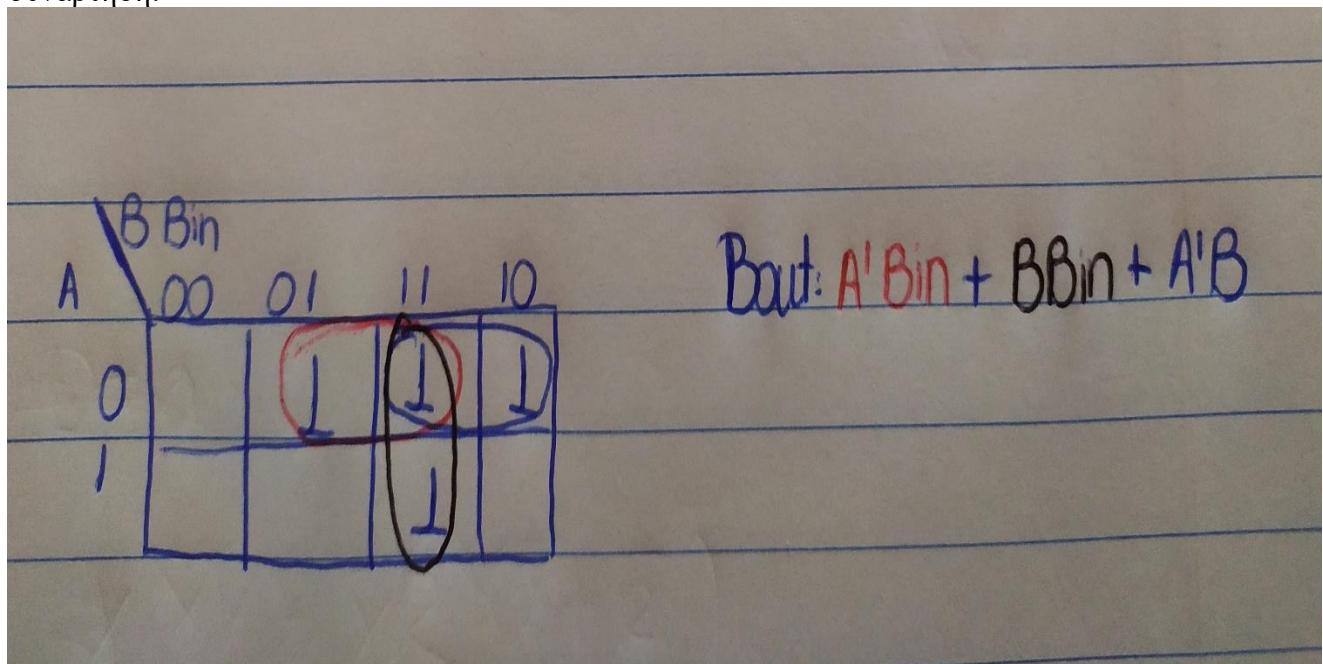
α. Γράψτε τον πίνακα αληθείας του **FS** και υπολογίστε τις λογικές συναρτήσεις των εξόδων **Bout**, **D**.

### ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ ΚΑΙ ΛΟΓΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ ΓΙΑ Bout, D

X	Y	Bin	Bout	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Το **D** προκύπτει από το XOR των **X Y** και **Bin**... Οπότε γίνεται 1 όποτε έχουμε περιττό αριθμό εμφανίσεων άσσων και 0 όποτε έχουμε ζυγό!

**Bout** = Από τον πίνακα αληθείας κάνουμε ένα πίνακα Karnaugh για να βρούμε την λογική συνάρτηση:



Επομένως **Bout**:  $X'Bin + BBin + A'B$ .

$$D = (X \wedge Y \wedge Bin)$$

β. Σχεδιάστε στον simulator έναν **FS**, χρησιμοποιώντας ως υποκυκλώματα **δύο** ημι-αφαιρέτες (**Half-Subtractors, HS**) και δώστε τα ζητούμενα screenshots από τον simulator.

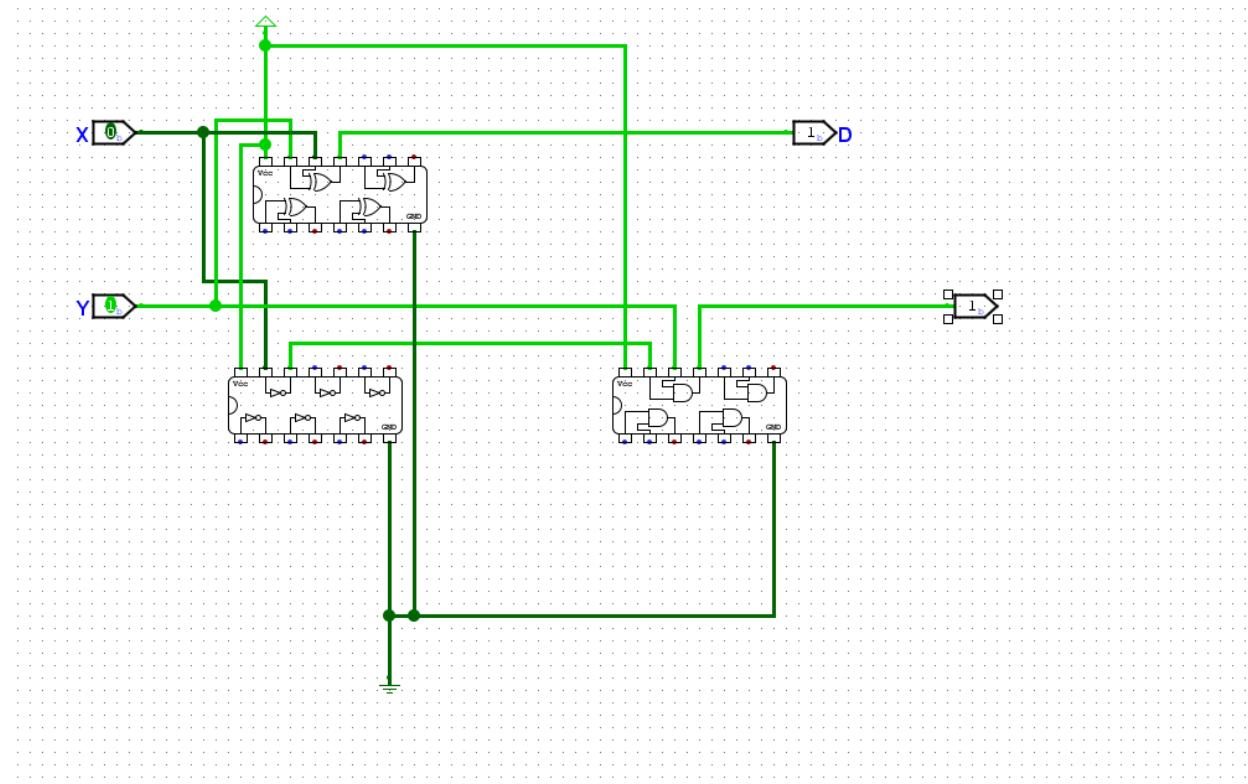
#### **Παρατηρήσεις για τον HS (δείτε το Σχήμα 1):**

- Χρησιμοποιείστε τα ολοκληρωμένα 7404, 7408, 7486 **από την βιβλιοθήκη TTL**.
- Συνδέστε τις εισόδους X, Y, Bin και τις εξόδους, Bout, D, σε pins εισόδου και εξόδου αντίστοιχα, γιατί θα χρησιμοποιήσετε το κύκλωμα του HS ως υποκύκλωμα για τον FS.
- Συμπεριλάβετε σε **ξεχωριστό screenshot το κύκλωμα του HS**

#### **Παρατηρήσεις για τον FS:**

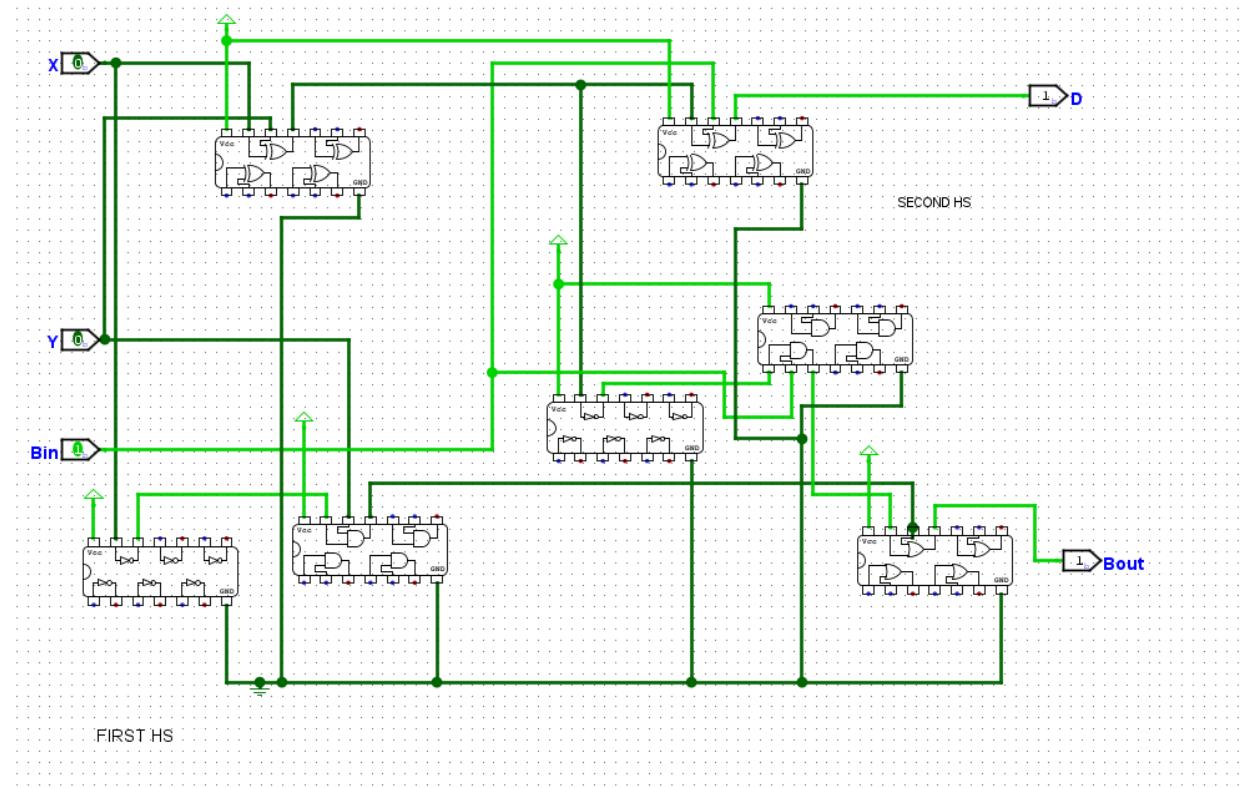
- Χρησιμοποιείστε τον HS ως υποκύκλωμα και επιπλέον ένα ολοκληρωμένο 7432, **από την βιβλιοθήκη TTL**.
- Συνδέστε τις εξόδους σας **και** σε LEDs για να βεβαιωθείτε ότι το κύκλωμά σας δουλεύει σωστά.

#### **ΚΥΚΛΩΜΑ HS (screenshot)**

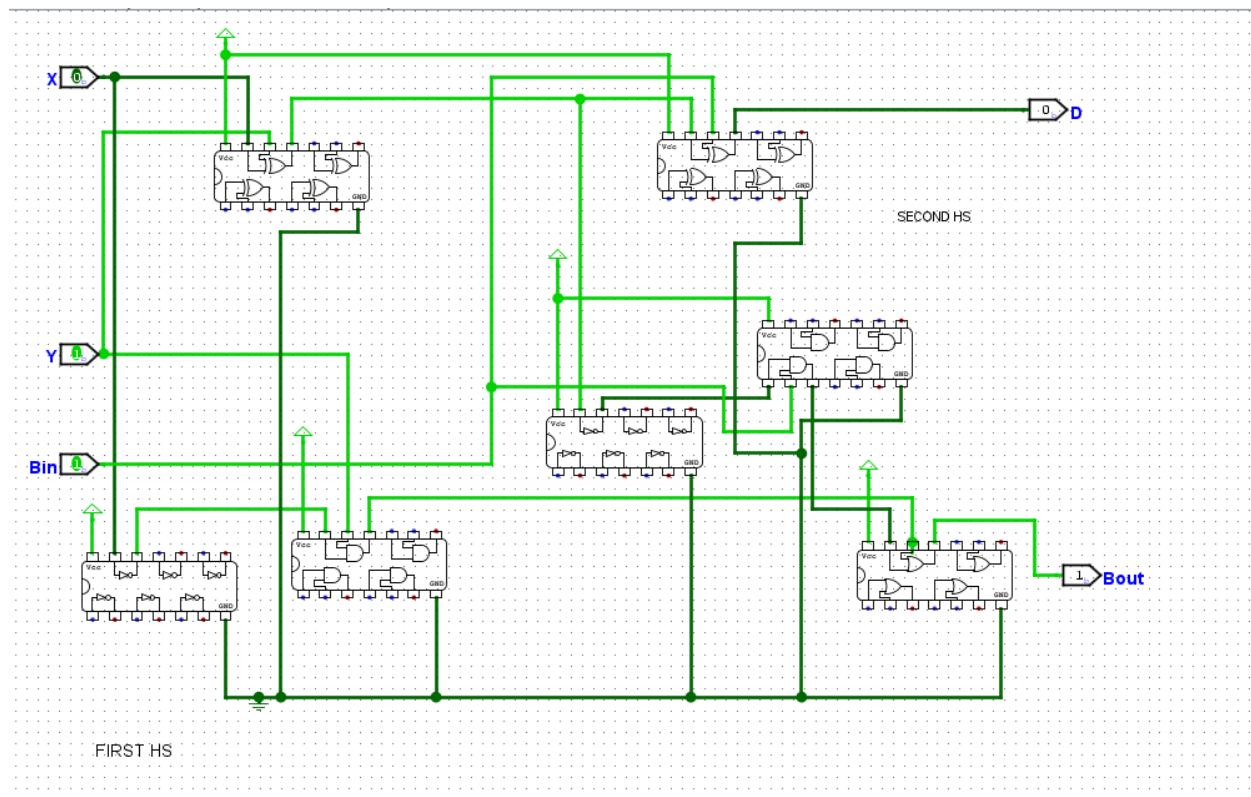


### ΚΥΚΛΩΜΑ FS (με 2 HS ως υποκυκλώματα)

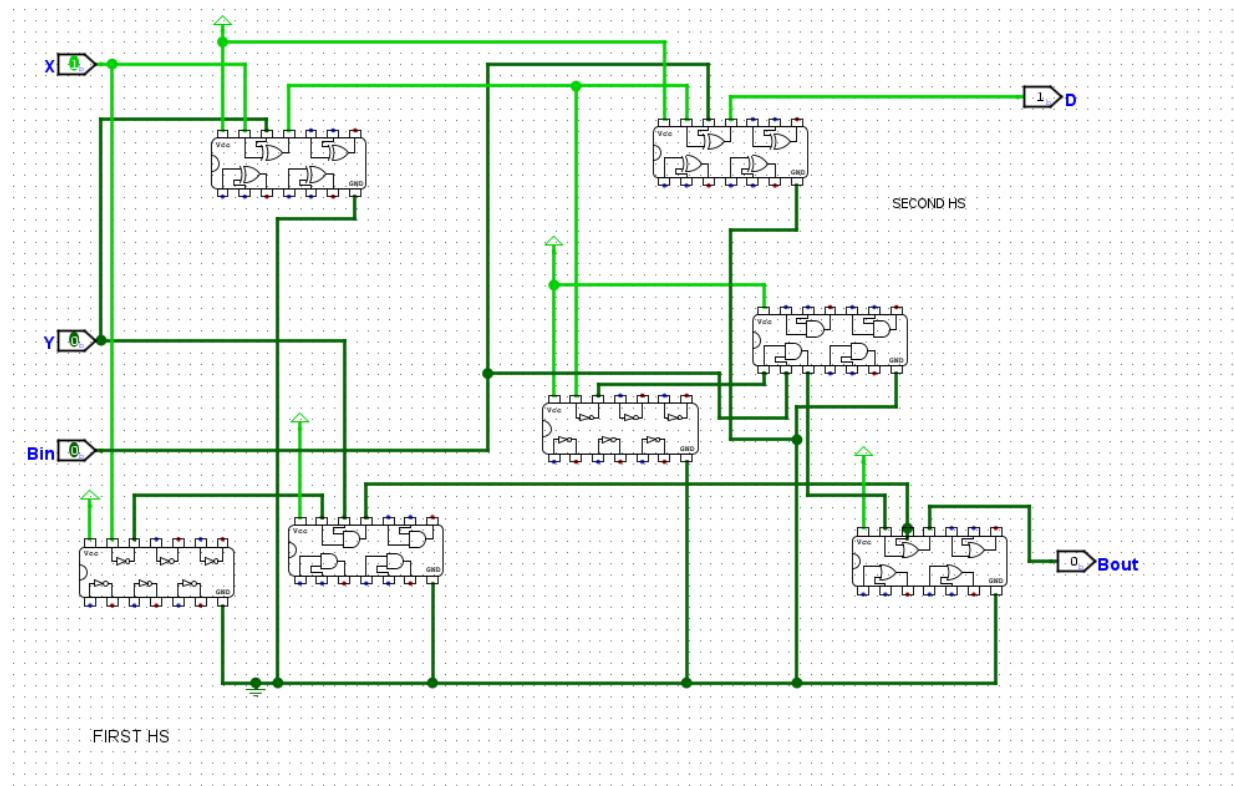
- Screenshot για την περίπτωση X = 0, Y=0, Bin=1



- Screenshot για την περίπτωση  $X = 0, Y=1, \text{Bin}=1$



- Screenshot για την περίπτωση  $X = 1, Y=0, \text{Bin}=0$



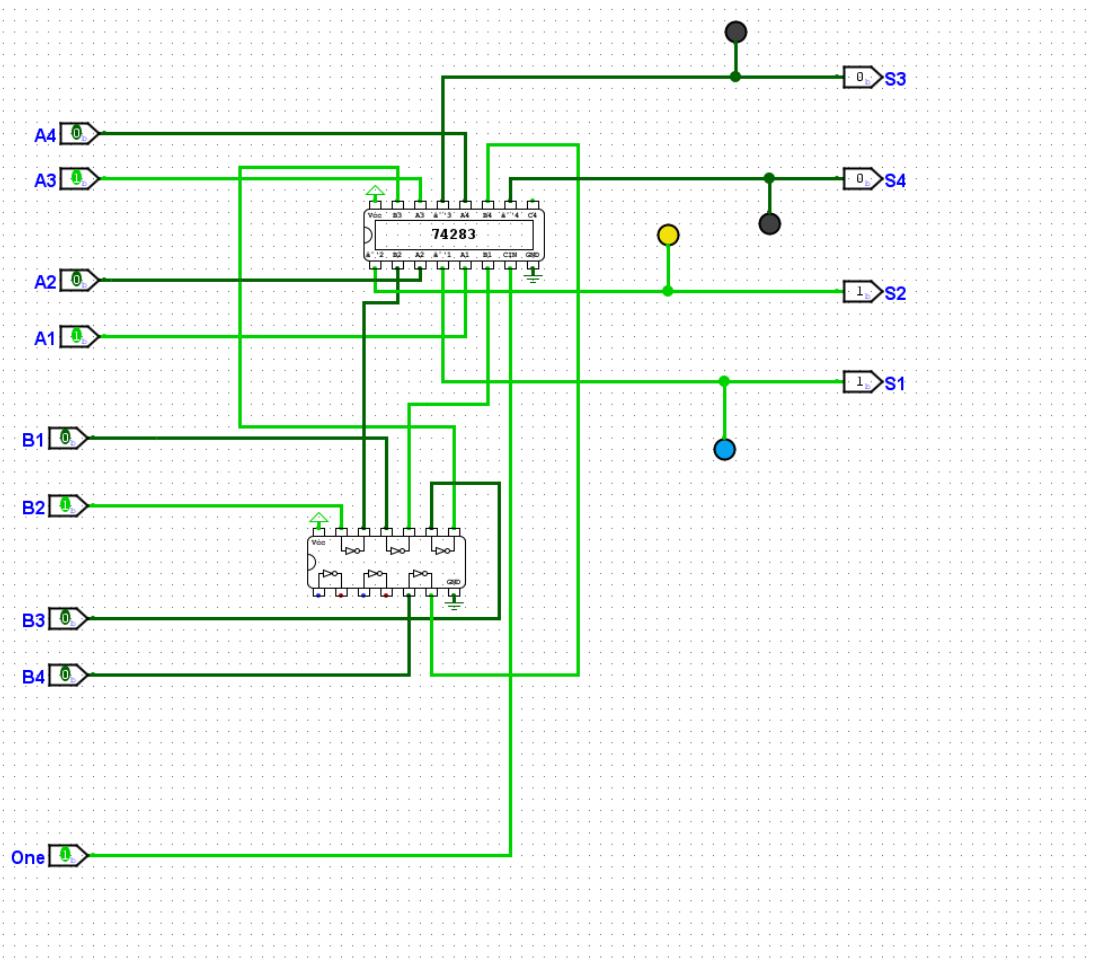
**3.A.** Σχεδιάστε στον simulator έναν **αφαιρέτη των 4-bits**, χρησιμοποιώντας **από την βιβλιοθήκη TTL:** ένα (1) ολοκληρωμένο **74LS283**, (4-bit Full-Adder, αθροιστής, βλ. ΠΑΡΑΡΤΗΜΑ, σελ17), και οποιοδήποτε άλλο ολοκληρωμένο κρίνετε απαραίτητο (πάλι από την βιβλιοθήκη TTL) και δώστε τα **ζητούμενα screenshots από τον simulator**.

### Παρατηρήσεις:

- Συνδέστε τις εισόδους του αφαιρέτη σε pins εισόδου (για απόδοση τιμών) και τις εξόδους του σε LEDs για να επιβεβαιώσετε τη σωστή λειτουργία του (αποτέλεσμα σε 2's complement).
- Συνδέστε τις εξόδους του αφαιρέτη **KAI** σε pins εξόδου, για να χρησιμοποιήσετε τον αφαιρέτη ως υποκύκλωμα στο συνολικό κύκλωμα του Ερωτ. 3.B (*Σχήμα 2*).

### ΚΥΚΛΩΜΑ ΑΦΑΙΡΕΤΗ των 4-bits

- Screenshot για ένα οποιοδήποτε ζευγάρι τιμών των A, B



### **Σύντομη δικαιολόγηση του κυκλώματος που σχεδιάσατε**

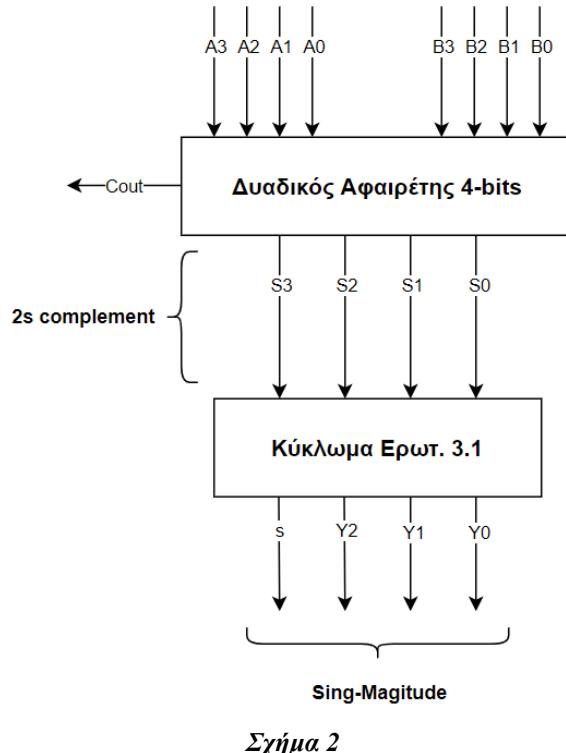
Στο κύκλωμα αυτό βλέπουμε έναν αφαιρετή με τη χρήση ενός full adder.

Έχουμε: **A – B = A + συμπλήρωμα (B) = A + B' + 1.**

Ουσιαστικά γίνεται μια πρόσθεση του A με το B αλλά με τα συμπλήρωμα το B ως προς 1 με συν 1.

**3B.** Χρησιμοποιήστε τον αφαιρέτη του Ερωτ. 3.A ως υποκύκλωμα και σχεδιάστε το συνολικό κύκλωμα του Σχήματος 2. Συνδέστε τις εξόδους του αφαιρέτη ( $S_3S_2S_1S_0$ ), αγνοώντας το τελικό κρατούμενο, στο συνδυαστικό κύκλωμα του **ερωτήματος 3.1**, επαληθεύστε ότι παίρνετε τα σωστά αποτελέσματα σε αναπαράσταση “**Signed-Magnitude**” (Σχήμα 2), και δώστε τα ζητούμενα screenshots από τον simulator

**Κύκλωμα Αφαιρέτη των 4-bits + Κύκλωμα Ερωτ. 3.1.**



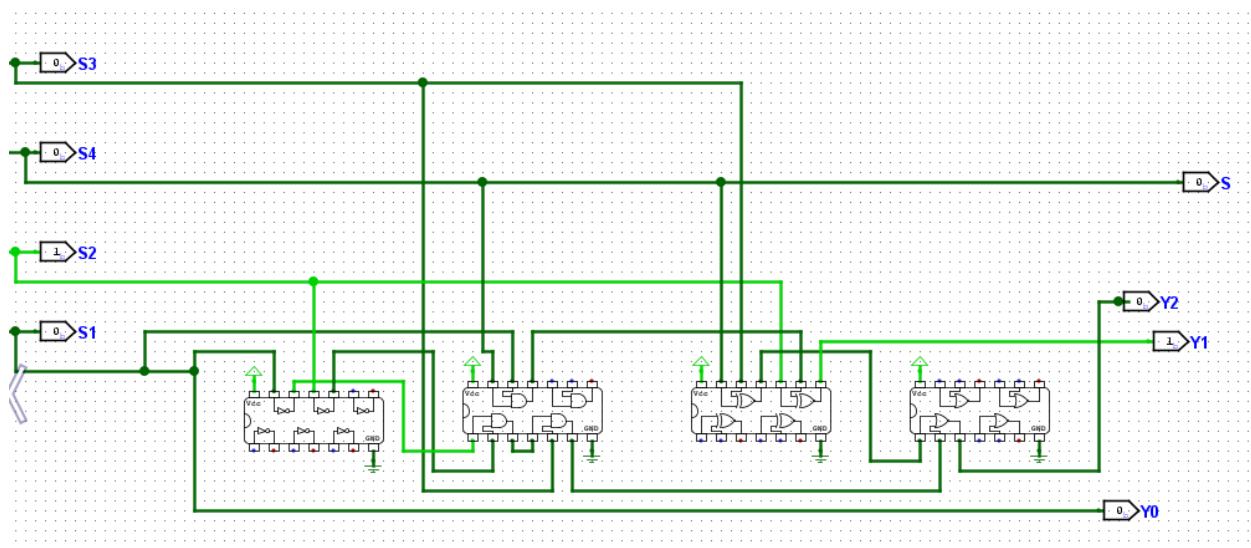
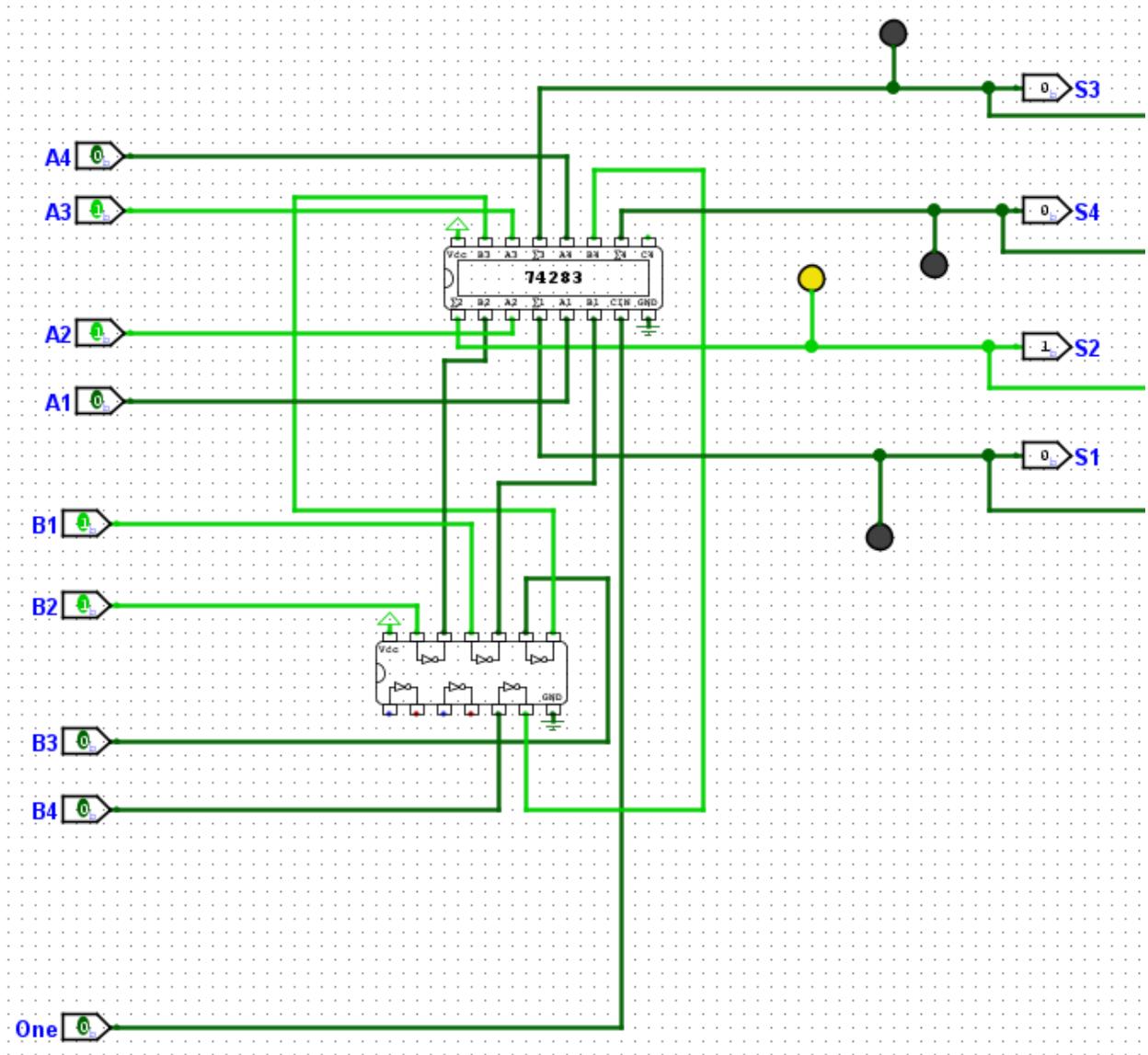
### Παρατηρήσεις:

- Συνδέστε τις εισόδους του ολικού κυκλώματος σε pins εισόδου για να αποδώσετε τιμές στα  $A_i, B_i, (i = 0 \dots 3)$  και τις εξόδους του κυκλώματος του Ερωτ. 3.1 ( $sY_2Y_1Y_0$ ) σε LEDs, για να βεβαιωθείτε ότι το κύκλωμα σας δίνει τα σωστά τελικά αποτελέσματα **σε signed-magnitude αναπαράσταση**.
- Μπορείτε να συνδέσετε και τις εξόδους του αφαιρέτη ( $S_3S_2S_1S_0$ ) σε LEDs ώστε να παρατηρείτε το αποτέλεσμα της αφαίρεσης και σε συμπλήρωμα ως προς 2 ( $2's complement$ )

### ΚΥΚΛΩΜΑ ΑΦΑΙΡΕΤΗ των 4-bits + ΚΥΚΛΩΜΑ ΕΡΩΤΗΜΑΤΟΣ 3.1 (Σχήμα 2)

- Screenshot για την περίπτωση  $A = 6_{10} = 0110_2, B = 3_{10} = 0011_2$

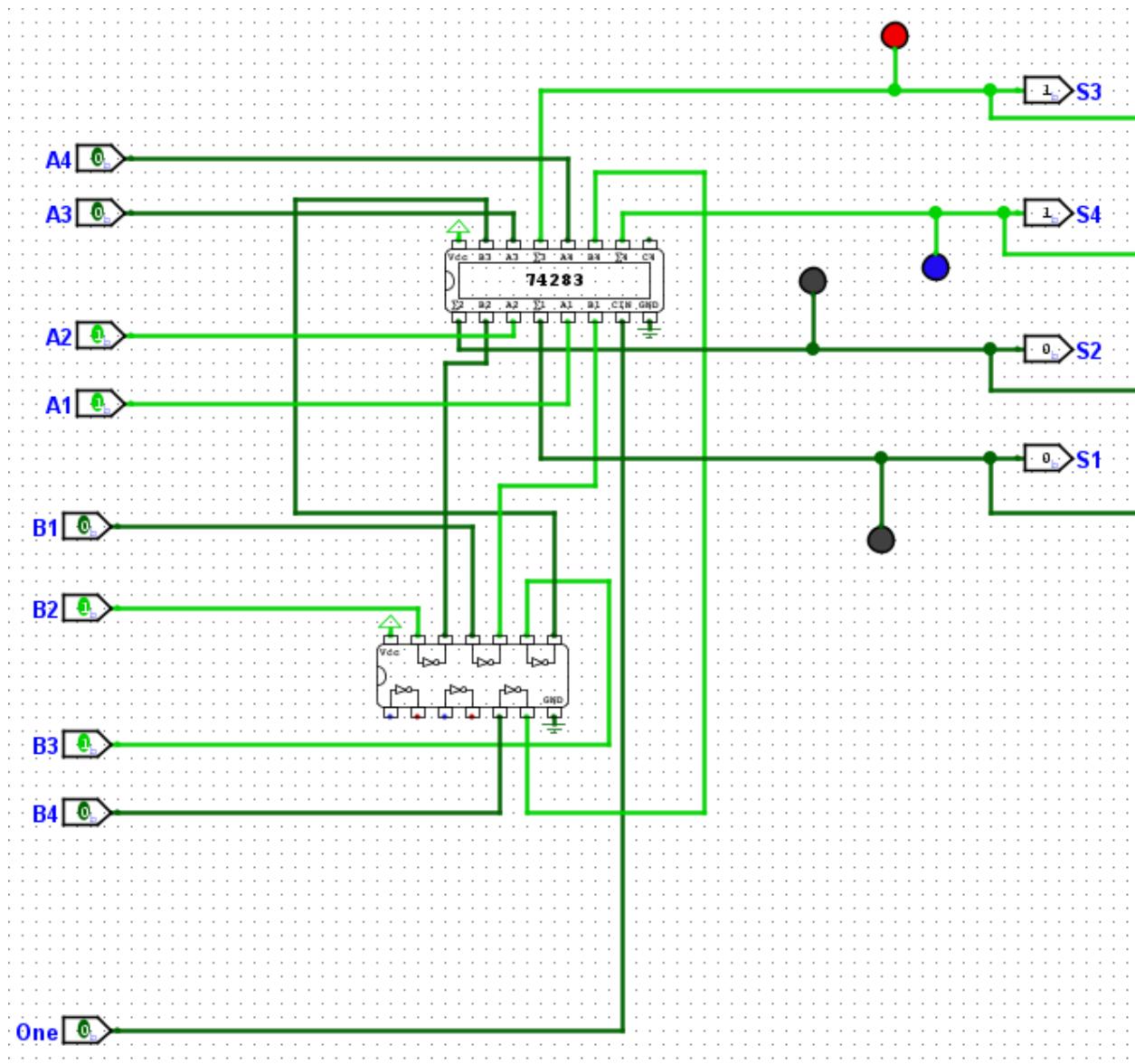
Το κύκλωμα είναι αρκετά μεγάλο για να χωρέσει σε μία φωτογραφία

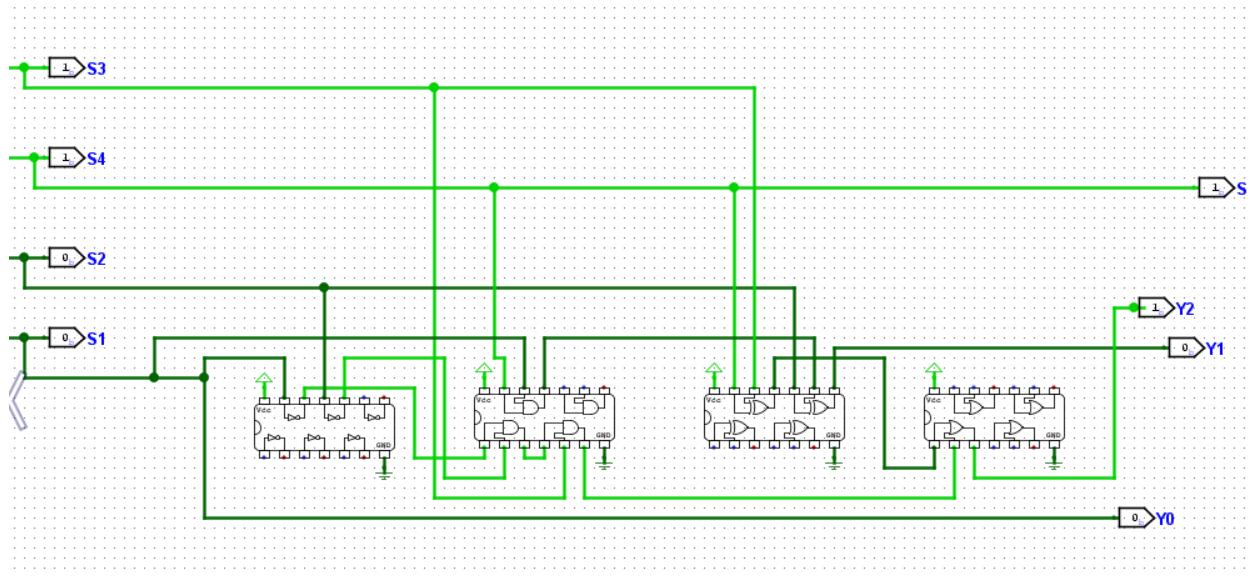


Δικαιολόγηση αποτελέσματος ( $A - B = 6 - 3$ )

$$A - B = 6 - 3 = 3 = 0011$$

- Screenshot για την περίπτωση περίπτωση  $A = 3_{10} = 0011_2$ ,  $B = 6_{10} = 0110_2$

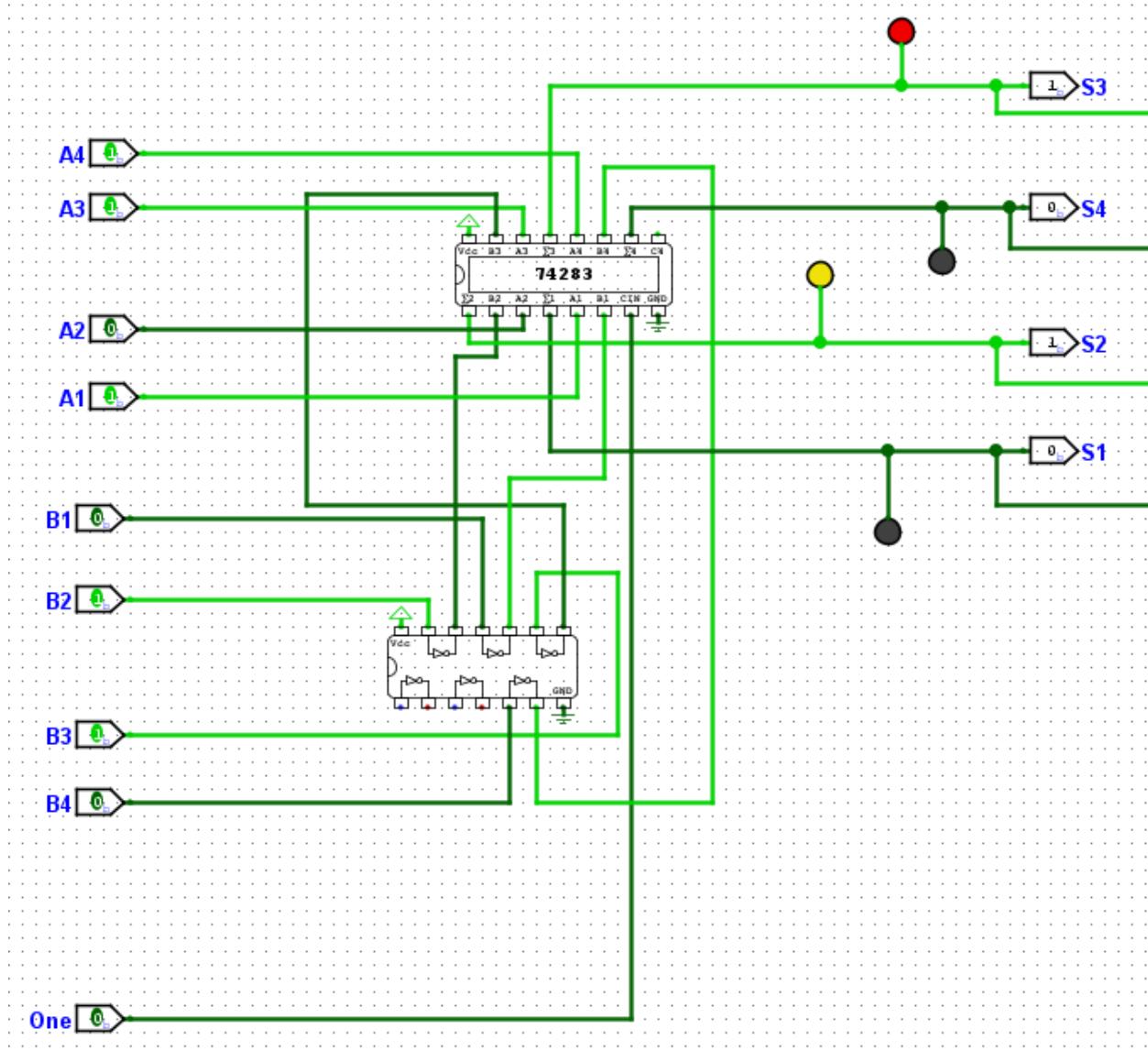


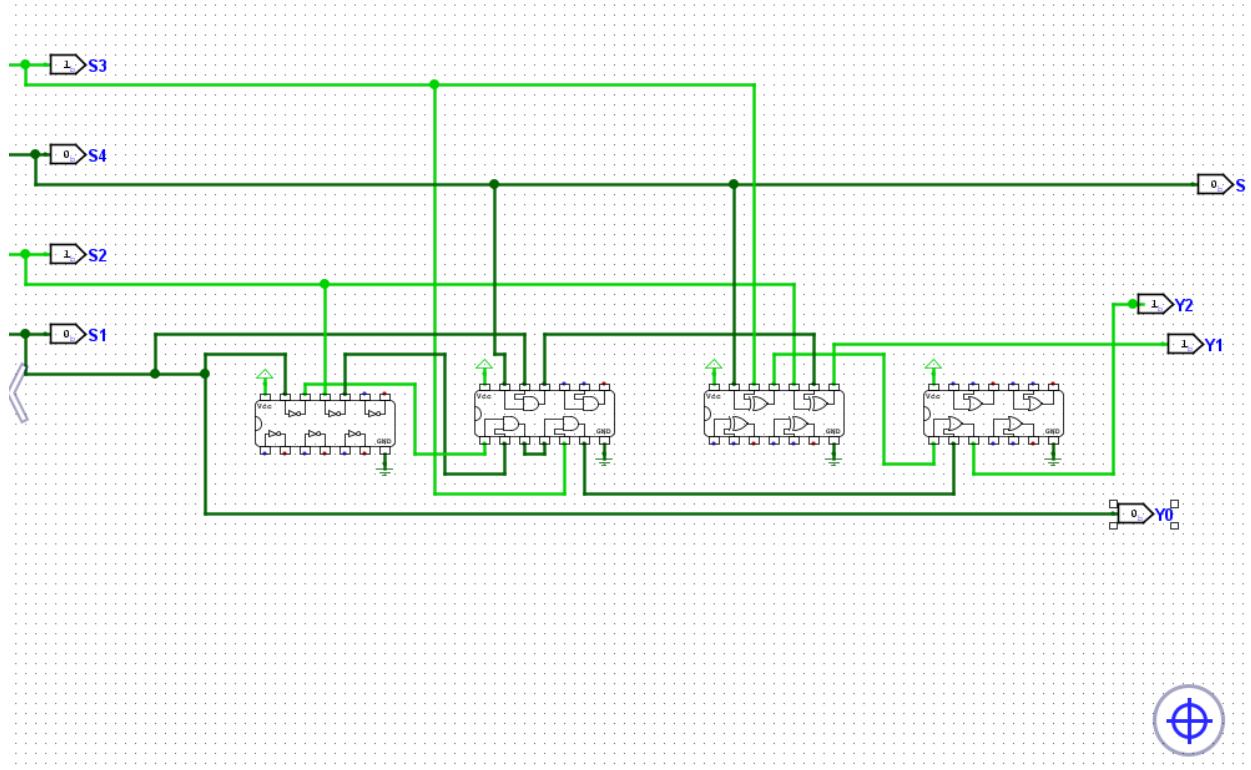


Δικαιολόγηση αποτελέσματος ( $A - B = 3 - 6$ )

$$A - B = 3 - 6 = -3 = (3)' = (0011)' = 1100 = 1011 \text{ σε προσημασμένο.}$$

- Screenshot για την περίπτωση περίπτωση  $A = -3_{10} = 1101_2$ ,  $B = 6_{10} = 0110_2$





*Δικαιολόγηση αποτελέσματος ( $A - B = -3 - 6$ )*

$A - B = -3 - 6 = -9$ , έχουμε υπερχείλιση διότι για να αναπαραστήσουμε το -9 χρειαζόμαστε 5 bit ενώ εμείς έχουμε μόνο 4.

**ΤΕΛΟΣ ΑΝΑΦΟΡΑΣ!!!**