

**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: Κατσαρός Ανδρέας**

**A.M. : 1084522**

**Όνομ/νυμ Συνεργάτη: Προδρόμου Λουκάς**

**A.M. : 1084590**

**Όνομ/νυμ Συνεργάτη: Χουσελάς Νικόλαος**

**A.M. : 1067432**

**Τμήμα : \_\_\_\_\_**

**Άσκηση:\_\_ ΑΣΚΗΣΗ 5 \_\_**

***Αναλυτική Βαθμολογία Πρακτικού Μέρους:***

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

## ΑΣΚΗΣΗ 5

### ΗΜΕΡΟΜΗΝΙΑ:

#### ΘΕΜΑ: ΑΡΙΘΜΗΤΙΚΗ\_ΛΟΓΙΚΗ ΜΟΝΑΔΑ (ARITHMETIC LOGIC UNIT, ALU)

#### ΘΕΩΡΙΑ

##### ALU

Η ALU είναι ένα συνδυαστικό κύκλωμα δύο εισόδων 4 bits που μπορεί να εκτελέσει διάφορες αριθμητικές και λογικές πράξεις. Το είδος της πράξης καθορίζεται από τον συνδυασμό που θα δώσουμε στις εισόδους επιλογής ( $S_3, S_2, S_1, S_0, M, C_n$ ). Ο παρακάτω πίνακας συνοψίζει τις διαθέσιμες λειτουργίες της ALU (συμβουλευτείτε επίσης το παράρτημα για το 74181).

**ΠΡΟΣΟΧΗ: Η είσοδος  $C_n$  της ALU είναι αντεστραμμένη (active low!) που σημαίνει ότι όταν  $C_n=0$  τότε νοάρχει κρατούμενο στην είσοδο και όταν  $C_n=1$  τότε δεν λαμβάνεται υπόψη (Δείτε στο Παράτημα, ποδαράκι 7)**

**ΠΡΟΣΟΧΗ: Η έξοδος  $C_{n4}$  της ALU είναι αντεστραμμένη (active low!) που σημαίνει ότι όταν από τη πράξη προκύπτει κρατούμενο τότε  $C_{n4}=0$  και όταν δεν προκύπτει τότε  $C_{n4}=1$  (Δείτε στο Παράτημα, ποδαράκι 16)**

**Πίνακας 1.** Διαθέσιμες Λειτουργίες του 74LS181

$S_3S_2S_1S_0$	M=1	M=0, $C_n=1$ ,	M=0, $C_n=0$
0 0 0 0	$F = A'$	$F = A$	$F = A \text{ PLUS } 1$
0 0 0 1	$F = (A + B)'$	$F = A + B$	$F = (A + B) \text{ PLUS } 1$
0 0 1 0	$F = A'B$	$F = A + B'$	$F = (A + B') \text{ PLUS } 1$
0 0 1 1	$F = 0$	$F = \text{MINUS } 1$	$F = \text{ZERO}$
0 1 0 0	$F = (AB)'$	$F = A \text{ PLUS } AB'$	$F = A \text{ PLUS } AB' \text{ PLUS } 1$
0 1 0 1	$F = B'$	$F = (A + B) \text{ PLUS } AB'$	$F = (A + B) \text{ PLUS } AB' \text{ PLUS } 1$
0 1 1 0	$F = A \oplus B$	$F = A \text{ MINUS } B \text{ MINUS } 1$	$F = A \text{ MINUS } B$
0 1 1 1	$F = AB'$	$F = AB' \text{ MINUS } 1$	$F = AB'$
1 0 0 0	$F = A' + B$	$F = A \text{ PLUS } AB$	$F = A \text{ PLUS } AB \text{ PLUS } 1$
1 0 0 1	$F = A \odot B$	$F = A \text{ PLUS } B$	$F = A \text{ PLUS } B \text{ PLUS } 1$
1 0 1 0	$F = B$	$F = (A + B') \text{ PLUS } AB$	$F = (A + B') \text{ PLUS } AB \text{ PLUS } 1$
1 0 1 1	$F = AB$	$F = AB \text{ MINUS } 1$	$F = AB$
1 1 0 0	$F = 1$	$F = A \text{ PLUS } A^\dagger$	$F = A \text{ PLUS } A \text{ PLUS } 1$
1 1 0 1	$F = A + B'$	$F = (A + B) \text{ PLUS } A$	$F = (A + B) \text{ PLUS } A \text{ PLUS } 1$
1 1 1 0	$F = A + B$	$F = (A + B') \text{ PLUS } A$	$F = (A + B') \text{ PLUS } A \text{ PLUS } 1$
1 1 1 1	$F = A$	$F = A \text{ MINUS } 1$	$F = A$

Όπου “+” → OR, “⊕” → XOR, “⊖” → XNOR, “PLUS” → αριθμ. πρόσθεση, “MINUS” → αριθμ. αφαίρεση

## ΔΙΑΔΙΚΑΣΙΑ

**ΠΡΟΣΟΧΗ :** Πριν ξεκινήσετε την υλοποίηση της άσκησης κατεβάστε **ΟΠΩΣΔΗΠΟΤΕ** από το eclass του εργαστηρίου (από την κατηγορία “Εγγραφα\Logisim-Evolution”) το .zip αρχείο με την ανανεωμένη βιβλιοθήκη για το Logisim (**CEID\_LoGicDesign\_LogisimEV\_library\_v1.1**). Για το πως εισάγουμε βιβλιοθήκη στο Logisim-Ev, συμβουλευτείτε την παρουσίαση για τον simulator που βρίσκεται στα “Εγγραφα/Logisim-Evolution” του eclass του εργαστηρίου.

- Συμπληρώστε τον παρακάτω πίνακα, θέτοντας τις εισόδους επιλογής ( $S_3, S_2, S_1, S_0, M, C_n$ ) της ALU (74181) στις κατάλληλες τιμές, (συμβουλευτείτε τον Πίνακα 1), και ελέγξτε τις αντίστοιχες αριθμητικές και λογικές λειτουργίες της ALU δίνοντας τα κατάλληλα screenshots.

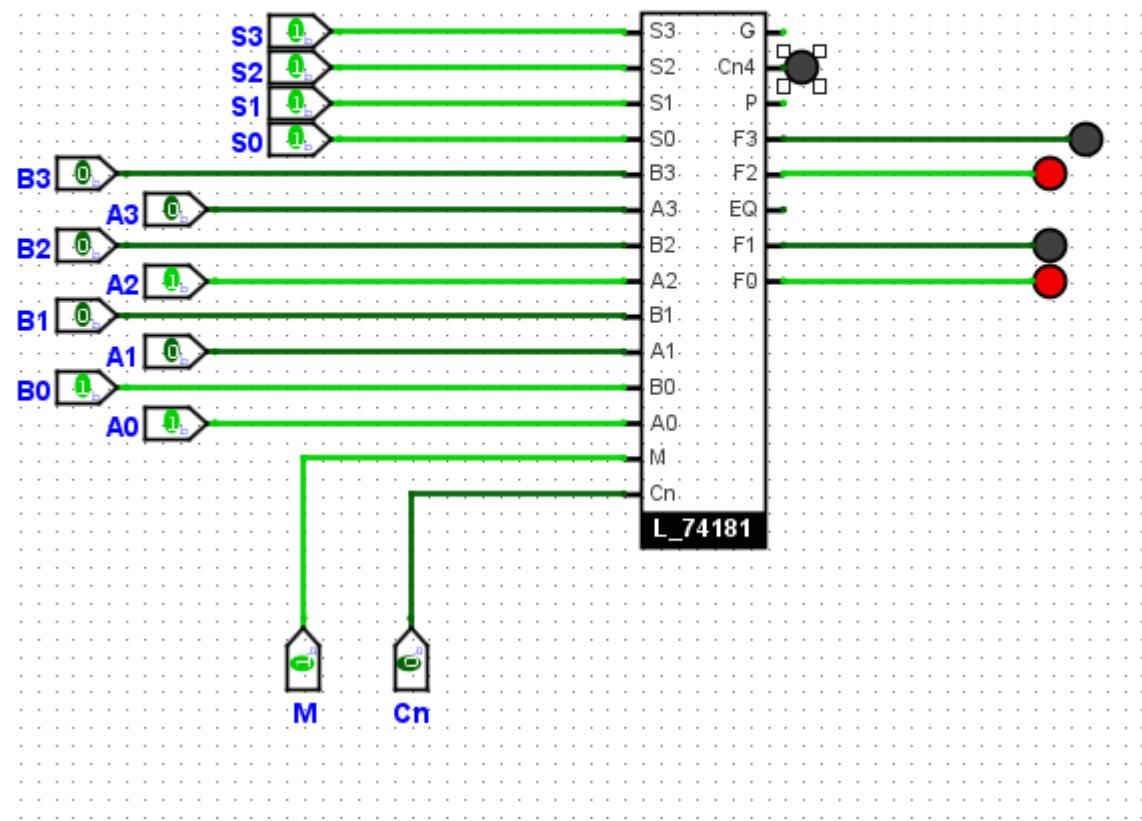
Πράξη ALU	$S_3$	$S_2$	$S_1$	$S_0$	$M$	$C_n$
$F = A$	1	1	1	1	1	X
$F = B$	1	0	1	0	1	X
$F = A \text{ XNOR } B$	1	0	0	1	1	X
$F = A \text{ MINUS } B$	0	1	1	0	0	0
$F=A \text{ PLUS } B$	1	0	0	1	0	1

### Παρατηρήσεις:

- Χρησιμοποιείστε το ολοκληρωμένο 74181 της βιβλιοθήκης **CEID\_LoGicDesign\_LogisimEV**
- Συνδέστε τα σήματα επιλογής  $S_3, S_2, S_1, S_0, M, C_n$  σε pins εισόδου και αποδώστε τιμές σε αυτά βάση του παραπάνω πίνακα που συμπληρώσατε
- Συνδέστε τα A, B σε pins εισόδου και αποδώστε τις τιμές  $A = 5_{10}$  και  $B = 1_{10}$
- Συνδέστε τις εξόδους,  $nC_{n4}, Y_4, Y_3, Y_2, Y_1$ , σε LEDs για να επιβεβαιωθείτε τη εκάστοτε λειτουργία της ALU.
- ΠΡΟΣΟΧΗ: Η είσοδος  $C_n$  της ALU είναι αντεστραμμένη (active low!). Η έξοδος  $C_{n4}$  της ALU είναι αντεστραμμένη (active low!) (δείτε στο Παράτημα, ποδαράκια 7, 16)**

## ΚΥΚΛΩΜΑ

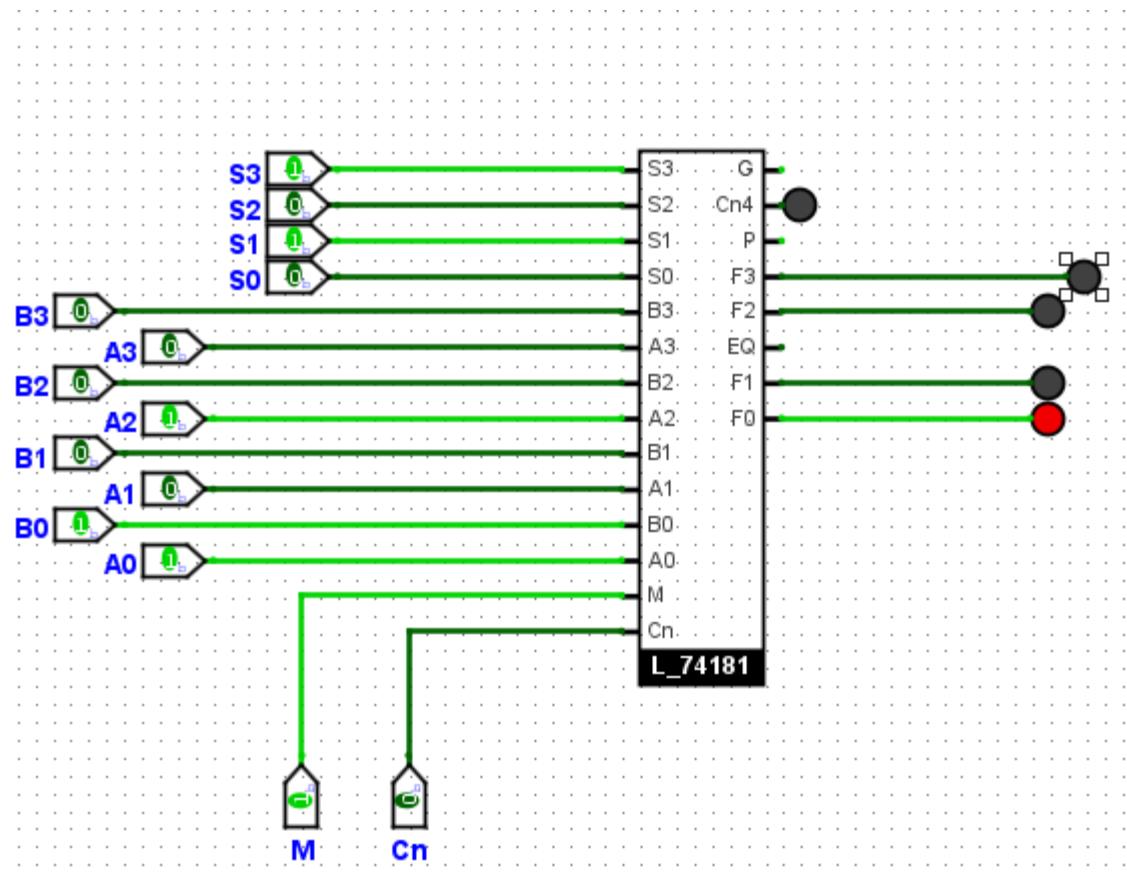
- Screenshot για την πράξη  $F=A$  ( $A = 5_{10}$ ,  $B=1_{10}$ )



Σύντομη επεξήγηση των αποτελέσματος:

Προκύπτει από τον πίνακα αληθείας!

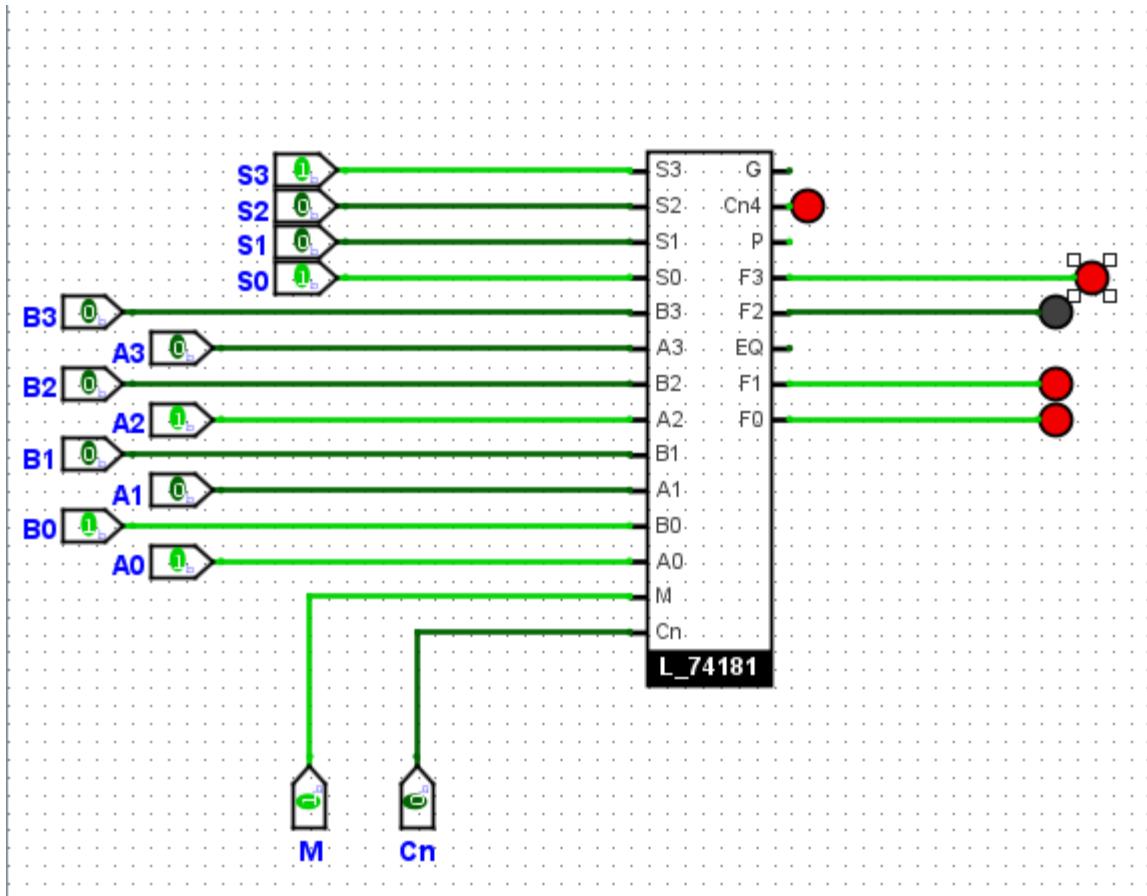
- Screenshot για την πράξη  $F=B(A = 5_{10}, B=1_{10})$



Σύντομη επεξήγηση των αποτελέσματος

Προκύπτει από τον πίνακα αληθείας!

- Screenshot για την πράξη  $F=A \text{ XNOR } B$  ( $A=5_{10}$ ,  $B=1_{10}$ )



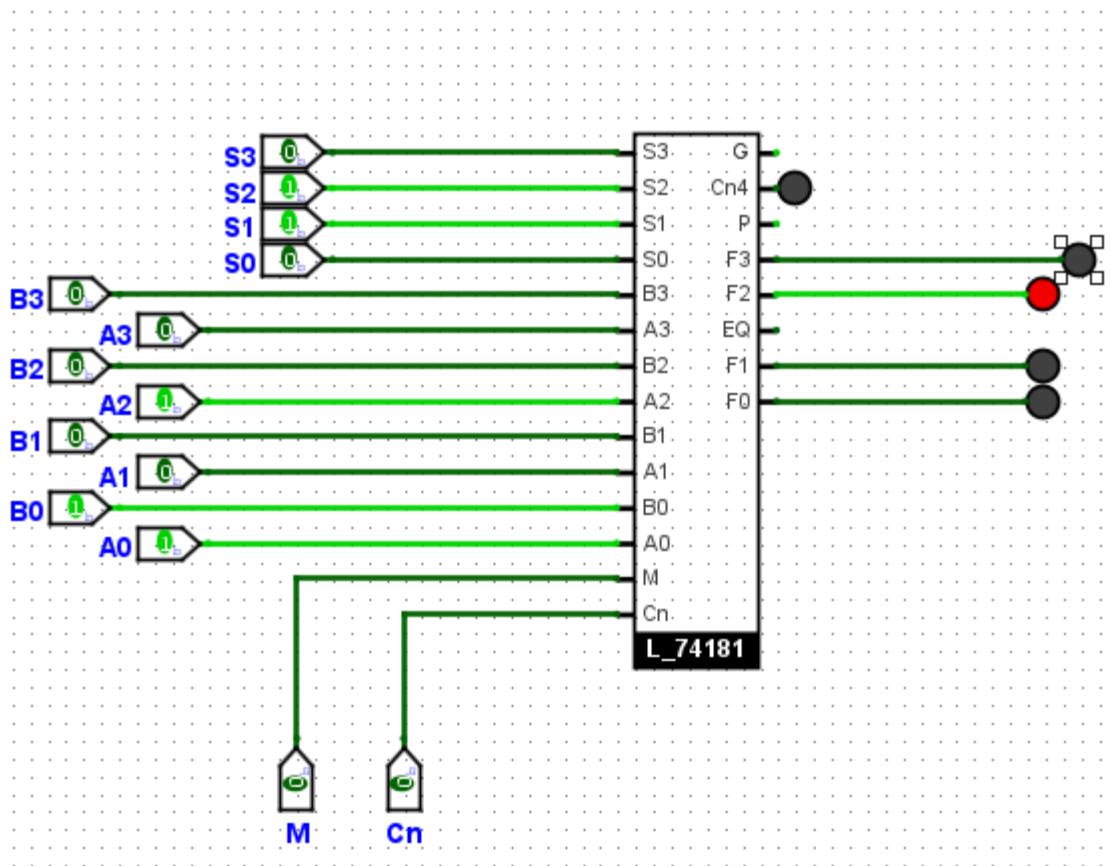
Σύντομη επεξήγηση των αποτελέσματος

Από τον πίνακα αληθείας για την λογική πράξη XNOR :

Inputs	Output	
A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

Η έξοδος δηλαδή της XNOR γίνεται 1 όταν και τα 2 bits είναι 1 ή 0. Επομένως οι εξόδοι F3, F1, F0 είναι 1 αφού B3=A3, B1=A1, B0=A0.

- Screenshot για την πράξη  $F=A \text{ MINUS } B$  ( $A = 5_{10}$ ,  $B=1_{10}$ )

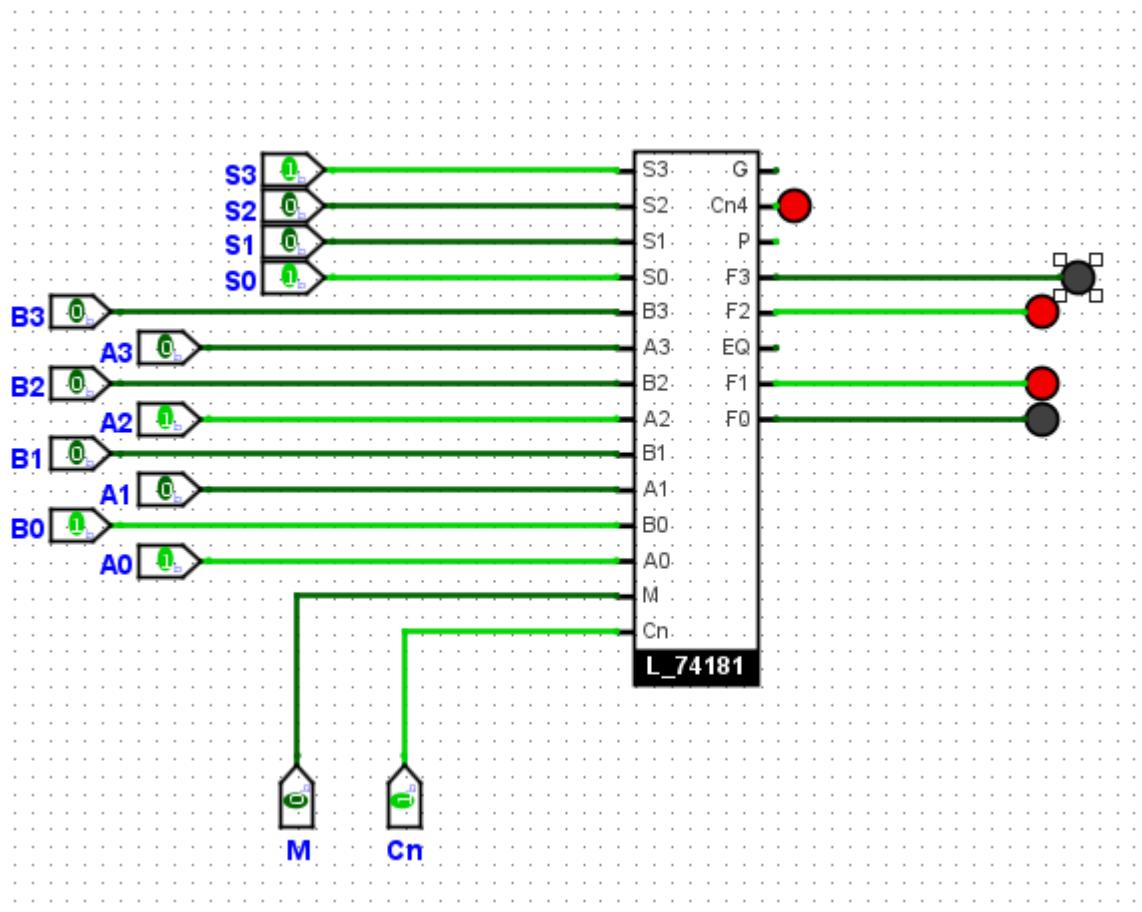


Σύντομη επεξήγηση των αποτελέσματος

$A - B = A + B' + 1$  δηλαδή :

$0101 + 1111 = 0100$  με κρατούμενο εξόδου 1 και επειδή η έξοδος  $Cn4$  είναι αντεστραμμένη το λεντάκι είναι σβησμένο.

- Screenshot για την πράξη **F=A PLUS B (A = 5<sub>10</sub>, B=1<sub>10</sub>)**



Σύντομη επεξήγηση των αποτελέσματος  
 $A + B = 0101 + 0001 = 0110$  με κρατούμενον εξόδου 0 αλλά επειδή η έξοδος C4n είναι αντεστραμμένη το λεντάκι είναι αναμμένο!

2. Χρησιμοποιώντας την ALU (74181) και τα κατάλληλα ολοκληρωμένα, υλοποιείστε στον simulator συνδυαστικό κύκλωμα το οποίο να συγκρίνει δύο αριθμούς των τεσσάρων bits, δικαιολογώντας πλήρως την απάντησή σας και δίνοντας τα απαραίτητα screenshots. Οι τελικές έξοδοι του κυκλώματός σας θα είναι 3: **G, E, L** και θα παίρνουν τις τιμές **G=1 μόνο** όταν  $A > B$ , **E=1 μόνο** όταν  $A = B$  και **L=1 μόνο** όταν  $A < B$ . Ισοδύναμα

	Τέξοδοι (1-bit)		
	G (Greater)	E (Equal)	L (Less)
<b>A&gt;B</b>	1	0	0
<b>A=B</b>	0	1	0
<b>A&gt;B</b>	0	0	1

#### Παρατηρήσεις:

- Από την βιβλιοθήκη **CEID\_LogicDesign\_LogisimEV** χρησιμοποιείστε **MONO** το ολοκληρωμένο 74181. Από την βιβλιοθήκη **TTL** μπορείτε να χρησιμοποιήσετε ολοκληρωμένα της επιλογής σας.
- Για την σχεδίαση του κυκλώματος σύγκρισής λαμβάνετε υπόψη σας **MONO** τις εξόδους **C<sub>n4</sub>,F<sub>3</sub>,F<sub>2</sub>,F<sub>1</sub>,F<sub>0</sub>**, της ALU.
- **ΠΡΟΣΟΧΗ: Η είσοδος C<sub>n</sub> της ALU είναι αντεστραμμένη (active low!). Η έξοδος C<sub>n4</sub> της ALU είναι αντεστραμμένη (active low!)** (δείτε στο Παράτημα, ποδαράκια 7, 16)
- Συνδέστε τα σήματα επιλογής S<sub>3</sub>, S<sub>2</sub>, S<sub>1</sub>, S<sub>0</sub>, M, C<sub>n</sub> σε pins εισόδου και αποδώσετε τιμές σε αυτά για την πράξη που θεωρείτε ότι πρέπει να γίνει (σύμφωνα με τον Πίνακα 1 της θεωρίας). Συνδέστε επίσης τις εισόδους A, B σε pins εισόδου και τις εξόδους του κυκλώματός σας (G, E, L) σε LEDs για να επιβεβαιώσετε τη σωστή λειτουργία του κυκλώματος που σχεδιάσατε. *Υπόδειξη: μπορείτε να συνδέσετε και τις εξόδους της ALU (C<sub>n4</sub>,F<sub>3</sub>,F<sub>2</sub>,F<sub>1</sub>,F<sub>0</sub>) σε LEDs για να παρατηρείτε κάθε φορά το αποτέλεσμα της ALU*
- **Σε περίπτωση σχεδίασης υποκυκλώματος πρέπει να συμπεριλάβετε και ξεχωριστό screenshot για το υποκύκλωμα που χρησιμοποιείτε!!**

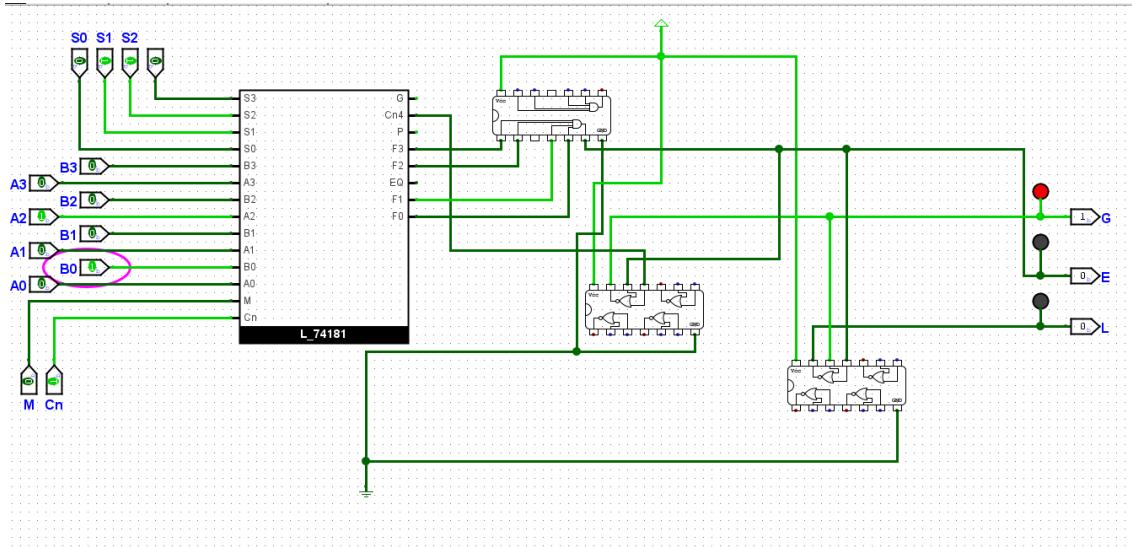
## ΣΥΝΔΙΑΣΤΙΚΟ ΚΥΚΛΩΜΑ

**Σύντομη δικαιολόγηση της επιλογής σχεδίασης:**

Χρησιμοποιούμε από τον πίνακα 1 της ALU , την τιμή (για τα  $s_3s_2s_1s_0$ ) 0110 και τις τιμές  $M=0$  και  $C_n=1$ . Έτσι έχουμε πρόσβαση στην πράξη  $A-B-1$ . Έτσι όταν  $A=B$  τότε το αποτέλεσμα αυτής της πράξης είναι -1. Το -1 σε συμπλήρωμα ως προς 2 παριστάνεται ως 1111 , άρα όλες οι έξοδοι  $F_3F_2F_1F_0$  είναι στο 1 όταν  $A=B$ . Άρα παίρνουμε όλες τις  $F$  σε μια πύλη AND και αν είναι όλες ίδιες δηλαδή 1, τότε  $A=B$ (πύλη  $E$ ).

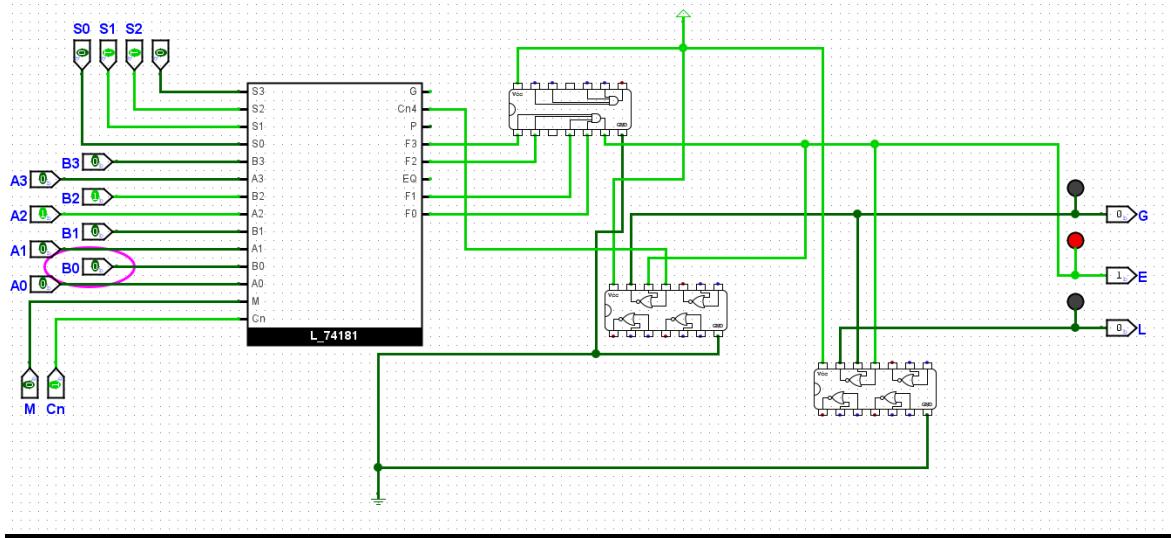
Τέλος παίρνουμε την έξοδο της AND μαζί με το  $C_{n4}$  και τα βάζουμε σε μια πύλη NOR και μας δείχνει αν  $A>B(G)$  και αντίστοιχα άλλη μια πύλη NOR που μας δείχνει εύκολα αν  $B>A(L)$ . Αυτό το κάνουμε γιατί αν  $A < B$  , τότε στη πράξη  $A-B-1$  θα προκύψει υπερχείλιση και γι' αυτό θα έχουμε  $C_{n4}$  ίσο με 0, ενώ αν  $A>B$  δεν θα υπάρξει υπερχείλιση και έτσι το  $C_{n4}$  θα είναι ίσο με 1.

- **Screenshot για την περίπτωση  $A = 4_{10}, B=1_{10}$**

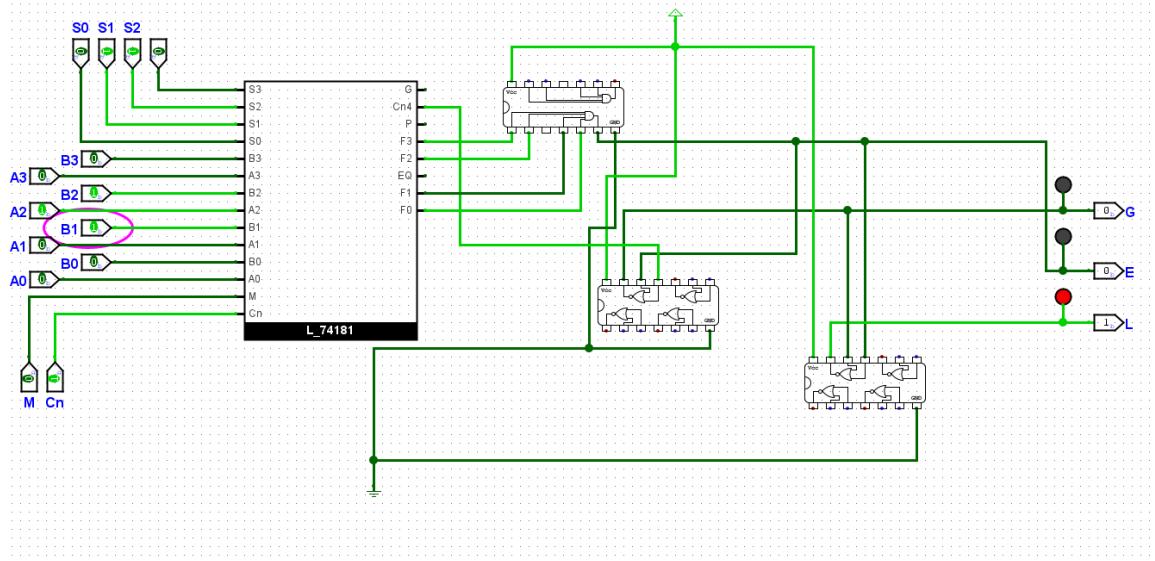


## Σελίδα 12

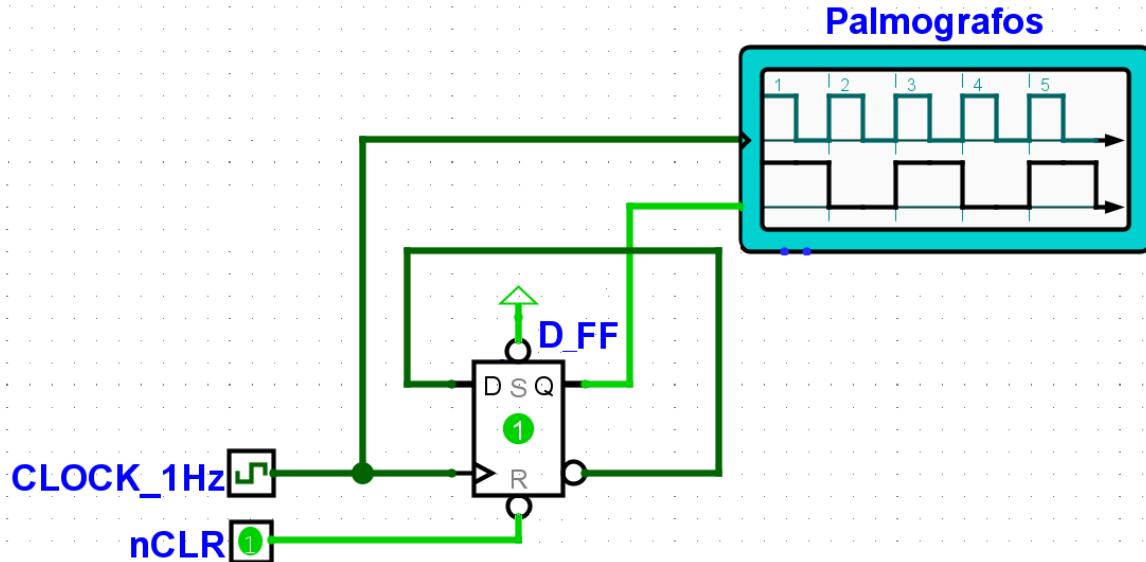
- Screenshot για την περίπτωση  $A = 4_{10}$ ,  $B=4_{10}$



- Screenshot για την περίπτωση  $A = 4_{10}$ ,  $B=6_{10}$



**3.Α.** Το κύκλωμα του Σχήματος 1 χρησιμοποιεί ένα D Flip-Flop για να παράγει έναν τετραγωνικό παλμό στη έξοδο του, Q, με συχνότητα  $f_1 = f_0/2$ , όπου  $f_0$  η συχνότητα του του τετραγωνικού παλμού στην είσοδο CLOCK του Flip-Flop. Υλοποιήστε στον simulator το κύκλωμα του Σχήματος 1 χρησιμοποιώντας το ολοκληρωμένο 74LS74 και επιβεβαιώστε τη διαίρεση συχνότητας στον παλμογράφο.



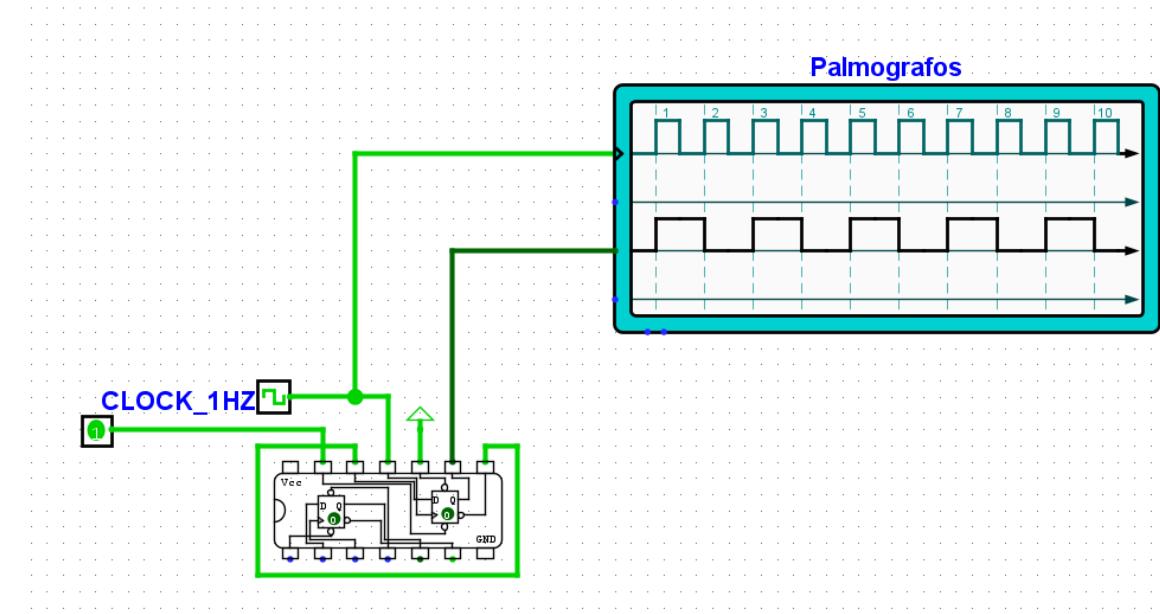
Σχήμα 1. Κύκλωμα Διαίρεσης Συχνότητας Τετραγωνικού Παλμού

#### Παρατηρήσεις:

- Χρησιμοποιείστε **MONO το ολοκληρωμένο 7474** από την βιβλιοθήκη **TTL** και τον παλμογράφο (*Digital oscilloscope*) από την βιβλιοθήκη **Input/Output-Extra**
- ΠΡΟΣΟΧΗ:** Οι είσοδοι CLEAR(ή *Reset*, R, *nCLR*) και PRESET(ή *Set*, S, *nPRE*) στο 7474 είναι **Active Low!**
- ΠΡΟΣΟΧΗ:** Για να ξεκινήσει το Flip-Flop να λειτουργεί σωστά πρέπει πρώτα να θέσετε την είσοδο CLEAR(ή *Reset*, R, *nCLR*) στο “0” (**CLEAR=0, εκκαθάριση**) και έπειτα να τη θέσετε στο λογικό “1”. Η είσοδος PRESET(*Set*, S, *nPRE*) πρέπει να συνδεθεί εξαρχής στο λογικό “1”
- Επιβεβαιώστε πριν την έναρξη της εξομοίωσης ότι η συχνότητα λειτουργίας (tick frequency) του CLOCK\_1Hz **είναι ίση με 1Hz**. (*Συμβουλεύετείτε την παρουσίαση για τον simulator που βρίσκεται στα “Εγγραφα” του eclass του εργαστηρίου*)

ΚΥΚΛΩΜΑ

- Screenshot κυκλώματος διαίρεσης συχνότητας (*να φαίνονται οι κυματομορφές*)

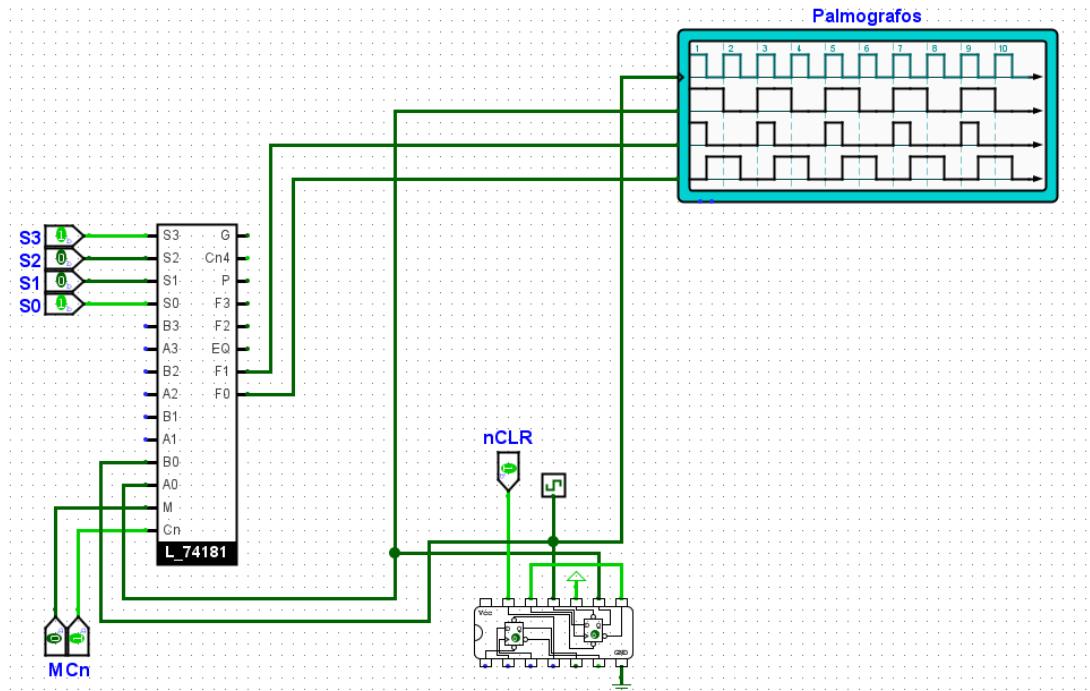


**3.B.** Χρησιμοποιώντας την ALU (74181) και συνδέοντας τις εισόδους επιλογής της τις κατάλληλες τιμές, υλοποιείστε στον simulator κύκλωμα που να κάνει "ΑΡΙΘΜΗΤΙΚΗ ΠΡΟΣΘΕΣΗ" (PLUS) δύο αριθμών **του 1-bit**. Τροφοδοτείστε τις εισόδους A<sub>0</sub> και B<sub>0</sub> της ALU με τους δύο τετραγωνικούς παλμούς (με λόγο συχνότητας 1/2 ( $f_1/f_0 = 1/2$ )) από το κύκλωμα του Ερωτ. 3.A, αντίστοιχα. Βεβαιωθείτε για τη σωστή λειτουργία της ALU ελέγχοντας τις εξόδους F<sub>0</sub> και F<sub>1</sub> με τον παλμογράφο.

## Παρατηρήσεις:

- Από την βιβλιοθήκη **CEID\_LogicDesign\_LogisimEV** χρησιμοποιείστε **MONO** το ολοκληρωμένο 74181. Από την βιβλιοθήκη **TTL MONO το ολοκληρωμένο 7474** και από την βιβλιοθήκη **Input/Output-Extra**, τον παλμογράφο (Digital oscilloscope).
  - Ρυθμίστε τον παλμογράφο ετσι ώστε να απεικονίζει συνολικά τα 4 σήματα CLOCK\_1Hz, Q, F<sub>1</sub>, F<sub>0</sub>, και να δείχνει όλους τους δυνατούς συνδυασμούς των 2 εισόδων A<sub>0</sub> και B<sub>0</sub> της ALU.

**ΚΥΚΛΩΜΑ (screenshot) όπου φαίνονται οι κυματομορφές εξόδου (όλοι οι συνδυασμοί):**



**Σύντομη Δικαιολόγηση των κυματομορφών:**

Η έξοδος **F1** δηλώνει το κρατούμενο επομένως γίνεται 1 όταν τόσο το A όσο και το B είναι 1. Η έξοδος **F0** γίνεται 1 όταν είτε το A είτε το B είναι 1 ενώ γίνεται 0 όταν και τα δυο είναι 1. Επίσης το **F0** είναι το least significant bit ενώ το **F1** είναι το most significant bit.

