



Δυαδικές Μονάδες Μνήμης: Μανδαλωτής SR, D και JK Flip-Flops Σχεδιασμός Μετρητής Ριπής

ΕΡΓΑΣΤΗΡΙΟ ΛΟΓΙΣΜΙΚΟΥ/ΥΛΙΚΟΥ 4

Εισαγωγή

Στην προηγούμενη εργαστηριακή άσκηση μελετήσαμε την ανάπτυξη και την υλοποίηση συνδυαστικών λογικών συναρτήσεων. Στην άσκηση αυτή μελετούμε τα δυαδικά στοιχεία μνήμης τα οποία ονομάζονται μανδαλωτές (latches) και flip-flops και τα οποία, μαζί με τα συνδυαστικά κυκλώματα, επιτρέπουν τον σχεδιασμό ακολουθιακών λογικών κυκλωμάτων. Αρχικά, εξετάζουμε τρία από τα πιο γνωστά στοιχεία μνήμης: τον μανδαλωτή SR και τα flip-flops D και JK (βλέπε σελ. 244-254 του βιβλίου των Mano&Kime). Ακολούθως, χρησιμοποιούμε τα JK flip-flops για να σχεδιάσουμε ένα modulo 16/10 μετρητή ριπής (ripple counter, βλέπε σελ. 331-333 του βιβλίου των Mano&Kime). Τέλος, ο ίδιος μετρητής θα περιγραφεί με κώδικα VHDL και θα προσομοιωθεί στο MAX+PLUS II simulator.

Μέρος Α (πρώτη εβδομάδα)

Σημείωση: Τα πιο κάτω αρχεία θα χρησιμοποιηθούν στην άσκηση αυτή. Πριν ξεκινήσετε κατεβάστε τα αρχεία αυτά από την ιστοσελίδα του μαθήματος (επιλογή Πρόγραμμα/Σημειώσεις) ή ακολουθήστε τις οδηγίες των υπευθύνων του εργαστηρίου. Αντιγράψτε τα αρχεία αυτά στο κατάλογο που θα αποθηκεύσετε τον σχεδιασμό σας για αυτή την άσκηση.

lab4a.gdf lab4a.scf lab4b.scf lab4c.vhd lab4c.scf

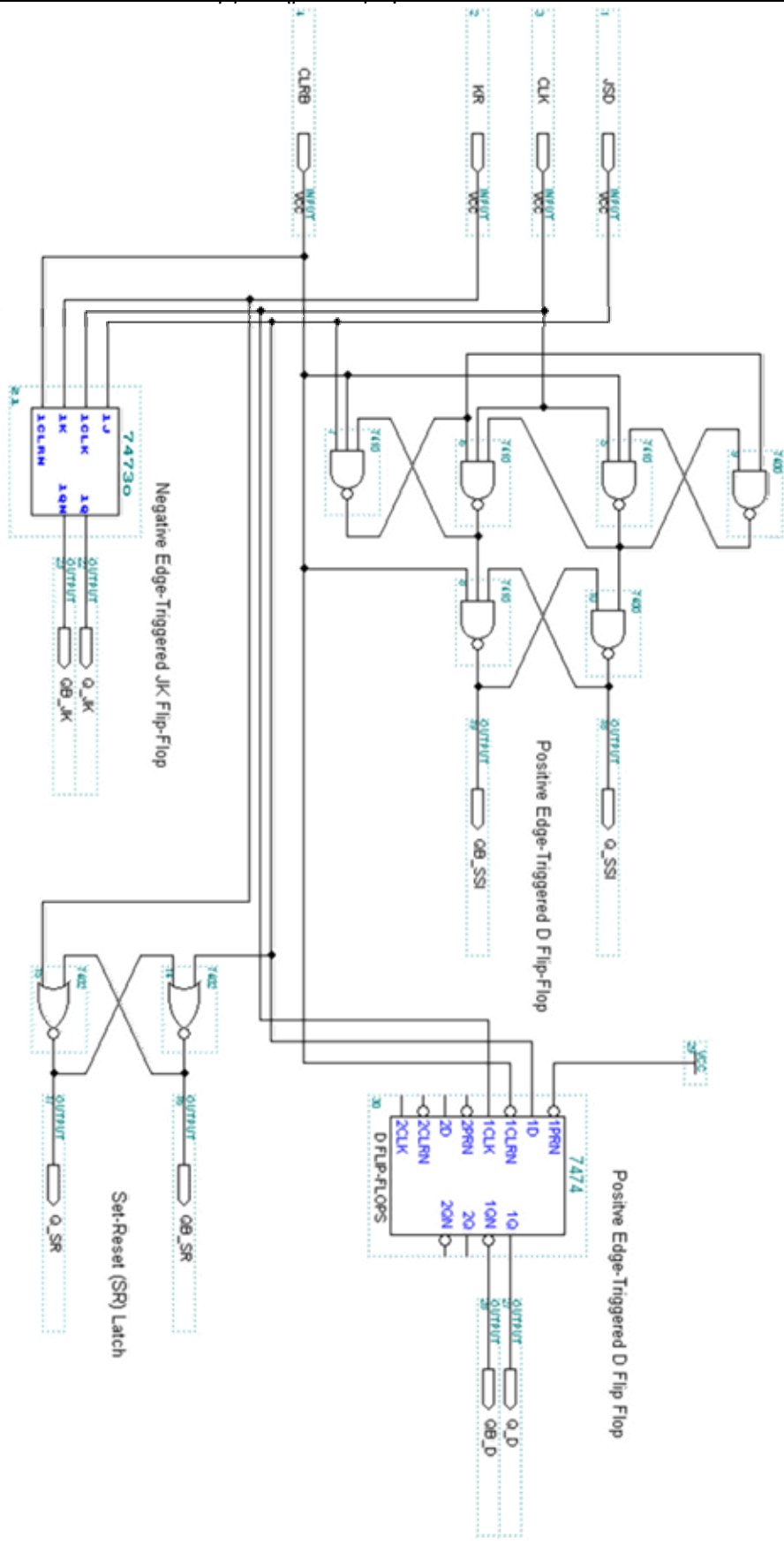
I. Μέρος Α.Ι: Μανδαλωτές (Latches) και Flip-Flops για τον σχεδιασμό στοιχείων μνήμης

Στο σχηματικό του Graphic Editor στην επόμενη σελίδα φαίνονται τέσσερις διαφορετικές διατάξεις μνήμης ενός bit. Το κύκλωμα των πυλών NAND (σύμβολα 5 έως 10) αποτελούν μια SSI υλοποίηση ενός θετικής λογικής ακμοπυροδότητου (positive edge-triggered) D flip-flop με ασύγχρονη είσοδο clear αρνητικής λογικής (active-low) και συμπληρωματικές εξόδους. Αυτό είναι ισοδύναμο, σε επίπεδο πύλης, με μια μονάδα MSI 7474A, που φαίνεται στην πάνω και δεξιά γωνιά του σχηματικού, με μόνη διαφορά ότι το τελευταίο έχει μια επιπλέον είσοδο αρνητικής λογικής για ασύγχρονο preset (καθορισμό της τιμής του). Οι πύλες NOR που είναι συνδεδεμένες χιαστή (σύμβολα 14 και 15), συνιστούν ένα Μανδαλωτή Set-Reset (SR Latch) με συμπληρωματικές εξόδους. Στην κάτω αριστερή γωνία φαίνεται ένα ολοκληρωμένο MSI 7473A που είναι ένα αρνητικής λογικής ακμοπυροδότητο (negative edge-triggered) JK flip-flop με ασύγχρονη είσοδο clear αρνητικής λογικής (active-low) και συμπληρωματικές εξόδους.

Στο μέρος Α.Ι, θα σχεδιαστεί και θα προσομοιωθεί το σχηματικό της επόμενης σελίδας. Το 7476A JK flip-flop, το οποίο διαθέτει τόσο clear όσο και preset εισόδους, θα χρησιμοποιηθεί στα μέρη Α.ΙΙ και Β.

Ο πίνακας αληθείας του SR Latch που φαίνεται στη σελίδα 3 δηλώνει ότι αν και οι δύο είσοδοι (S και R) είναι High (1) τότε και οι δύο εξόδοι είναι Low (0). Αυτή η λειτουργία είναι αντίθετη με την κανονική λειτουργία των δύο εξόδων που τις θέλει να είναι συμπληρωματικές ($Q = (QB)'$). Για το λόγο αυτό, και για την αποφυγή ταλαντώσεων (oscillations) στις τιμές εξόδου όταν αλλάζει η κατάσταση αυτή, ο σχεδιαστής πρέπει να διασφαλίσει ότι οι είσοδοι S και R δεν είναι ποτέ High (1) ταυτόχρονα. Οι πίνακες αληθείας για τα D και JK flip-flops φαίνονται επίσης στη σελίδα 3. Σημειώστε ότι αυτά είναι flip-flops με ρολόι (clock) τα οποία ενεργοποιούνται με ακμή (ακμοπυροδοτούνται = edge-triggered) στην είσοδο

Δωδικές Μονάδες Μνήμης



S	R	Q ⁺	QB ⁺
0	0	Q	QB
0	1	0	1
1	0	1	0
1	1	0	0

Μανδαλωτής με θετικής λογικής Set και Reset

7474a D Flip-Flop

Θετικής Λογικής Ακμοπυροδότητο (Positive edge-triggered) με ασύγχρονα Preset (PREN) και Clear (CLRΝ) αρνητικής λογικής (active-low)

CLRΝ	PREN	D	CLK	Q ⁺	QN ⁺
0	0	X	X	1	1
0	1	X	X	0	1
1	0	X	X	1	0
1	1	X	≠↑	Q	QN
1	1	0	↑	0	1
1	1	1	↑	1	0

7473a (*χωρίς είσοδο PREN) και 7476a JK Flip-Flops

Αρνητικής Λογικής Ακμοπυροδότητα (Negative edge-triggered) με ασύγχρονα Clear (CLRΝ) και Preset(PREN*) αρνητικής λογικής (active-low)

CLRΝ	PREN*	J	K	CLKN	Q ⁺	QN ⁺
0	0	X	X	X	1	1
0	1	X	X	X	0	1
1	0	X	X	X	1	0
1	1	0	0	↓	Q	QN
1	1	0	1	↓	0	1
1	1	1	0	↓	1	0
1	1	1	1	↓	(Q)'	(QN)'
1	1	X	X	≠↓	Q	QN

clock. Οι άλλες είσοδοι πρέπει να παραμένουν σταθερές για συγκεκριμένα χρονικά διαστήματα μετά την ενεργοποίηση του ρολογιού.

a) Προσομοίωση με το MAX+PLUS II

1. Δημιουργήστε ένα νέο κατάλογο με το όνομα lab4 στο χώρο σας. Αντιγράψετε το αρχείο του graphic editor, lab4a.gdf, καθώς και το αρχείο κυματομορφών, lab4a.scf, από την ιστοσελίδα του μαθήματος, όπως περιγράφηκε πιο πριν.
2. Ανοίξτε το MAX+PLUS II. Ανοίξτε το αρχείο lab4a.gdf από τον κατάλογο σας.
Ο σχεδιασμό που υπάρχει στην προηγούμενη σελίδα πρέπει να ανοίξει στον Graphic Editor.
3. Μεταφράστε τον σχεδιασμό με την επιλογή για χρονική ανάλυση (timing analysis).
4. Ανοίξτε το Simulator με το αρχείο mg4a.scf σαν είσοδο.
5. Τρέξτε την προσομοίωση για 1000ns και μετά ανοίξτε το αρχείο κυματομορφών για να δείτε τα αποτελέσματα.
6. Κρατήστε μία εκτύπωση της κυματομορφής.
7. Μελετήστε στην έξοδο της προσομοίωσης και σχολιάστε τα αποτελέσματα. Σημειώστε κυρίως τις διαφορές ανάμεσα σε θετικά και αρνητικά ακμοπυροδότητα flip-flops και μεταξύ των flip-flops με ρολόι και των μανδαλωτών. Σημειώστε τις διακυμάνσεις στις εξόδους του μανδαλωτή RS και τις συνθήκες για τις οποίες συμβαίνουν. Σημειώστε ότι οι καθυστερήσεις που έχουν να κάνουν με την NAND υλοποίηση του D flip-flop είναι διαφορετικές από αυτές του 7476A flip-flop. Ποιο από τα δύο στοιχεία μνήμης παρουσιάζει πραγματικά συμπληρωματική συμπεριφορά στις εξόδους Q και QN για όλες τις χρονικές στιγμές;
8. Κλείστε όλα τα παράθυρα, αλλά παραμείνετε στο MAX+PLUS II.

II. Μέρος A.II : Μετρητής Ριπής modulo 16/10 (Modulo 16/10 Ripple Counter) με JK flip-flops

Ένα JK flip-flop μετατρέπεται σε T flip-flop (από την λέξη Toggle που σημαίνει αλλάζει την κατάσταση) όταν και οι δύο εισοδοί κρατηθούν High (1). Με αυτές τις συνθήκες το flip-flop αλλάζει κατάσταση (από 0 σε 1 ή από 1 σε 0) σε κάθε ακμή του σήματος του ρολογιού. Στο μέρος αυτό θα σχεδιάσετε ένα μετρητή ριπής με την χρήση τεσσάρων αρνητικής λογικής ακμοπυροδότητα (negative edge-triggered) JK flip-flops (7476A) με ασύγχρονα preset και clear αρνητικής λογικής (active-low).

Ένα σήμα ρολογιού (clock) με το όνομα *CLK* θα οδηγεί την είσοδο του ρολογιού του πρώτου flip-flop (το πρώτο flip-flop αντιστοιχεί στο λιγότερο σημαντικό ψηφίο – LSB), η έξοδος Q του πρώτου flip-flop θα οδηγεί την είσοδο του ρολογιού του δεύτερου flip-flop (επόμενου σημαντικού ψηφίου), και το ίδιο για τα επόμενα δύο flip-flops. Όλες οι J και K εισοδοί, καθώς και τα presets θα τεθούν σταθερά στο High (*Vcc*). Έτσι όταν η έξοδος κάποιου flip-flop πάει από High σε Low (negative edge), θα προκαλεί μια αλλαγή κατάστασης στο επόμενο flip-flop και η ομάδα των τεσσάρων flip-flop θα λειτουργεί σαν ένας δυαδικός μετρητής ριπής τεσσάρων bit.

Άλλη μια είσοδος, με όνομα *CLRB*, θα θέτει όλα τα flip-flops στο 0 όταν το *CLRB* είναι Low (low-active clear). Θα υπάρχει, επίσης, και μια τρίτη είσοδος, η *MOD10*, η οποία καθορίζει πότε τα flip-flops θα μηδενιστούν. Όταν το *MOD10* είναι High, ο μετρητής θα μηδενίζεται όταν η μέτρηση φτάνει στο δυαδικό αριθμό 1010. Έτσι όταν το *MOD10* είναι Low και το *CLRB* είναι High ο μετρητής θα απαριθμεί κυκλικά και τις 16 δυαδικές τιμές που μπορούν να αναπαρασταθούν με 4 bit, ενώ όταν και τα δύο σήματα (*MOD10* και *CLRB*) είναι High, ο μετρητής θα απαριθμεί κυκλικά μόνο τους αριθμούς από το 0000 έως το 1001 (γίνετε modulo 10 μετρητής).

Η έξοδος του κυκλώματος είναι το σύνολο των τεσσάρων εξόδων των flip-flop συνδεδεμένες με ένα διάυλο σε μία μοναδική έξοδο, την *CNT[3..0]*. Η έξοδος του πρώτου flip-flop (που οδηγείτε από το *CLK*) είναι το λιγότερο σημαντικό ψηφίο (LSB) του μετρητή, το *CNT0*.

1. Κατασκευάστε ένα κύκλωμα το οποίο να συμφωνεί με τις προδιαγραφές που τέθηκαν στην προηγούμενη παράγραφο, με την χρήση δύο ολοκληρωμένων 7476A (περιέχουν συνολικά 4 flip-flops) και ένα ολοκληρωμένο 7410 (περιέχει 3 NAND 3 εισόδων).
2. Δημιουργήστε ένα νέο σχεδιασμό, lab4b.gdf, και αποθηκεύσετε το στο κατάλογο σας lab4. Σχεδιάστε το κύκλωμα που έχετε κάνει στο προηγούμενο βήμα. Κρατήστε μια εκτύπωση ολόκληρου του σχεδιασμού.

Ο σχεδιασμός σας θα περιέχει μία σύνδεση διαύλου στην θύρα εξόδου. Ανατρέξτε στην εργαστηριακή άσκηση 3 για υπενθύμιση της χρήσης διαύλων. Επίσης, μπορείτε να χρησιμοποιήσετε την ονομασία κόμβων ώστε να εξοικονομήσετε χώρο και να είναι το κύκλωμα σας πιο ξεκάθαρο και ευανάγνωστο.

3. Αντιγράψτε το αρχείο με τις κυματομορφές, lab4b.scf, από την ιστοσελίδα του μαθήματος στο κατάλογο σας lab4.
4. Μεταφράστε και προσομοιώστε το σχεδιασμό του μετρητή με το πιο πάνω αρχείο για 3000ns με την επιλογή για χρονική ανάλυση απενεργοποιημένη.
5. Κρατήστε δύο εκτυπώσεις του αποτελέσματος της προσομοίωσης. Ένα από 0 έως 3100ns και ένα από 1380 έως 1720ns.

6. Κλείστε όλα τα παράθυρα.

III. Μέρος Α.III : Προσομοίωση σε VHDL του Μετρητή Ριπής Modulo 16/10

Στην άσκηση αυτή θα ανοίξετε ένα έτοιμο αρχείο VHDL που περιγράφει την δομή (structural description) ενός μετρητής ριπής mod 16/10, θα το μεταφράσετε και θα το προσομοιώσετε. Με την άσκηση αυτή θα δείξετε τις δυνατότητες που έχει το λογισμικό σας για την προσομοίωση αρχείων VHDL που περιγράφουν λογικούς σχεδιασμούς όπως ακριβώς γίνεται και με τα αρχεία του graphic editor. Επίσης, μπορούν να προσομοιωθούν projects που περιέχουν, τόσο σχηματικές περιγραφές όσο και περιγραφές σε VHDL, τα οποία όμως δεν θα δείξουμε σε αυτή την εργαστηριακή άσκηση.

1. Αντιγράψετε τα αρχεία lab4c.vhd και lab4c.scf από την ιστοσελίδα όπως περιγράφηκε πιο πριν, στο κατάλογο σας lab4.
2. Ανοίξτε το lab4c.vhd στο MAX+PLUS II.

Το VHDL αρχείο θα ανοίξει στην εφαρμογή Text Editor. Παρατηρήστε για λίγο το VHDL κώδικα. Αν δεν είναι ήδη επιλεγμένο, επιλέξτε το **Syntax Coloring** (Options menu). Οι λέξεις-κλειδιά της VHDL θα αλλάξουν χρώμα, κάτι που επιτρέπει καλύτερη ανάγνωση και αποσφαλμάτωση (debugging) του κώδικα.

Παρόλο που γενικά ένα αρχείο VHDL μπορεί να δημιουργηθεί με οποιονδήποτε editor, το MAX+PLUS II προσφέρει διάφορα εργαλεία τα οποία διευκολύνουν την διαδικασία σχεδιασμού. Όταν θα κληθείτε να αναπτύξετε μόνοι σας κώδικα VHDL θα βοηθηθείτε πολύ, χρησιμοποιώντας τα templates της VHDL (Templates menu) καθώς και την επιλογή **Auto-Indent** (Options menu) επιπλέον του χρωματισμού του κώδικα που είδαμε στο σημείο αυτό.

3. Μεταφράστε και προσομοιώστε το lab4c project με την χρήση του αρχείου lab4c.scf που έχετε αντιγράψει στο βήμα 1. Κρατήστε μια εκτύπωση του κώδικα καθώς και των αποτελεσμάτων της προσομοίωσης.

Αναφορά Μέρους Α

- Μέρος I. Συμπεριλάβετε τις προσομοιώσεις από το α.6 και την συζήτηση από το α.7.
- Μέρος II. Σχολιάστε τα αποτελέσματα σας, συμπεριλαμβανομένων και των πλεονεκτημάτων και των μειονεκτημάτων ενός μετρητή ριπής έναντι ενός σύγχρονου μετρητή (synchronous counter) (βλέπε σχετική ύλη στο βιβλίο των Mano&Kime, σελ. 331-335) καθώς και την χρήση των τριών NAND για παραλλαγές στο υπόλοιπο (modulus variations). Συμπεριλάβετε τις εκτυπώσεις από το σχηματικό του βήματος 2 και τις δύο εξόδους προσομοίωσης από το βήμα 5.
- Μέρος III. Συμπεριλάβετε τις κυματομορφές μαζί με τα αποτελέσματα της προσομοίωσης καθώς και τον κώδικα από το αρχείο lab4c.vhd από το βήμα 3.

Μέρος Β.Ι (δεύτερη εβδομάδα)

Σε αυτή την εργαστηριακή άσκηση υλικού θα επαληθεύσετε πειραματικά την λειτουργία του modulo 16/10 μετρητή ριπής, που προσομοιώσατε στα μέρη Α.Ι και Α.ΙΙ. Η πλακέτα UP2 περιλαμβάνει ένα ταλαντωτή (oscillator) για παραγωγή παλμών συχνότητας 25.175 MHz. Ο ταλαντωτής αυτός είναι συνδεδεμένος με το ολοκληρωμένο FLEX 10K και επιτρέπει την υλοποίηση ακολουθιακών κυκλωμάτων, αφού χρησιμοποιείται σαν το ρολόι του κυκλώματος. Η έξοδος του μετρητή θα φαίνεται στο SSD αφού πρώτα μετατρέψετε τις εξόδους του με ένα αποκωδικοποιητή hex to seven-segment, όπως κάνατε στην προηγούμενη άσκηση (3). Παρόλα αυτά, το ρολόι είναι πολύ γρήγορο και δεν θα μπορέσετε να ελέγξετε την λειτουργία του μετρητή για αυτό σας παρέχετε μια συνάρτηση για μείωση (διαίρεση) της συχνότητας του ρολογιού.

1. Ανοίξετε το σχηματικό lab4b.gdf.
2. Αντιγράψετε τα περιεχόμενα από τον κατάλογο της βιβλιοθήκης UP1core library functions στο κατάλογο σας lab4.
3. Τοποθετήστε μια πύλη not στην είσοδο MOD10.
Ο μετρητής θα μετρά μέχρι το 9 όταν ο διακόπτης πίεσης 2 είναι πατημένος.
4. Εισάγετε ένα σύμβολο dec_7seg και ένα σύμβολο clk_div στο σχηματικό σας
Το clk_div είναι υλοποιημένο σε VHDL. Κάντε Double-click στο σύμβολο για να ανοίξετε το κώδικα του σε VHDL και παρατηρήστε πώς η συνάρτηση διαιρεί το σήμα του ρολογιού σε μικρότερες συχνότητες.
5. Συνδέστε την θύρα εισόδου CLK στην είσοδο του συμβόλου clk_div και την έξοδο του clock_1Hz στην είσοδο του ρολογιού του πρώτου flip-flop.
Η συχνότητα 1 Hz είναι αρκετά αργή για να σας επιτρέψει να δείτε την έξοδο του μετρητή.
6. Διαγράψετε την έξοδο CNT[3..0], και τραβήξετε μια γραμμή στην είσοδο του συμβόλου dec_7seg. Ονομάστε τον κόμβο (γραμμή) αυτό CNT[3..0].
7. Εισάγετε 7 νέες θύρες εξόδου και ονομάστε τις με τα λατινικά γράμματα a έως g. Συνδέστε τις θύρες εξόδου στις εξόδους του συμβόλου dec_7seg.
8. Αναθέστε το project στο ολοκληρωμένο FLEX 10K EPF10K70RC240-4 chip.
9. Αναθέστε το pin 91 (ταλαντωτής 25.175 MHz) στην είσοδο CLK.
10. Αναθέστε το pin 28 (push button 1) στην είσοδο CLRB.
11. Αναθέστε το pin 29 (push button 2) στην είσοδο MOD10.
12. Αναθέστε τα pins 17 έως 24 πλην του 22 στις εξόδους a έως g, αντίστοιχα.
13. Αποθηκεύσετε, ελέγξτε και μεταφράστε τον σχεδιασμό σας.
14. Κατεβάστε το σχεδιασμό σας στο ολοκληρωμένο FLEX 10K της πλακέτας UP2 με την εφαρμογή Programmer.
15. Ελέγξτε τον σχεδιασμό. Σχολιάστε τα αποτελέσματα και εξηγήστε γιατί ο μετρητής ονομάζεται μετρητής ριπής (ripple counter).

Μέρος Β.ΙΙ (Δεύτερη Εβδομάδα)

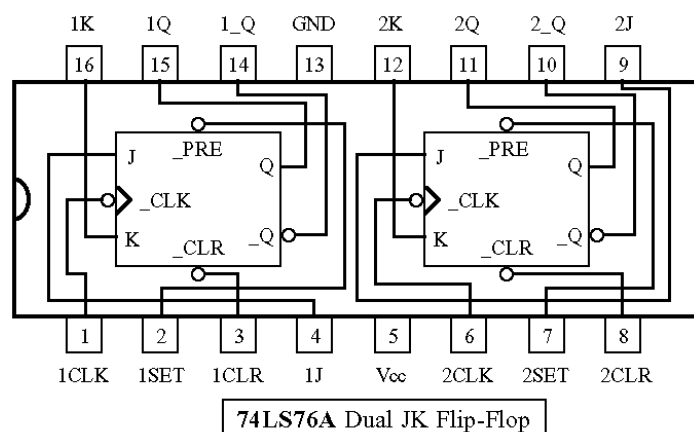
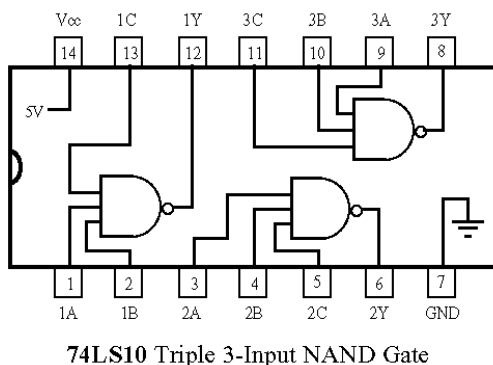
Σε αυτή η άσκηση θα επαληθεύσετε πειραματικά την λειτουργία του μετρητή ριπής modulo 16/10 που προσομοιώθηκαν στα μέρη Α.ΙΙ και Α.ΙΙΙ. **Φερέτε το σχηματικό lab4b στο εργαστήριο κάνοντας την προετοιμασία που περιγράφεται πιο κάτω:**

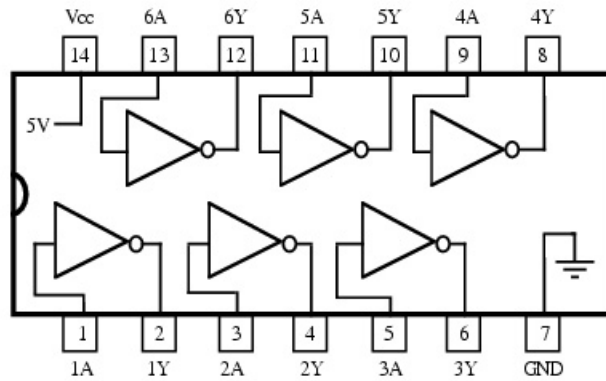
Προετοιμασία:

Σημειώστε στα στοιχεία του σχηματικού lab4b τα ονόματα των ολοκληρωμένων και τους αριθμούς των pins που θα χρησιμοποιήσετε. Ανατρέξτε στα διαγράμματα των ολοκληρωμένων πιο κάτω για την σωστή ανάθεση των pins στις εισόδους και τις εξόδους στο σχηματικό σας. Αν επιθυμείτε σημειώστε και τα Vcc και GND στο σχηματικό σας. Το ολοκληρωμένο 74LS10 έχει το Gnd στο Pin 7 και το Vcc στο Pin 14, όπως συνήθως. Αντιθέτως, το 74LS76A flip-flop έχει στο Pin 5 το Vcc και στο Pin 13 το Gnd.

Εισάγετε ένα 74LS04 Hex Inverter στην δεύτερη στήλη διαχωριστικού της πλακέτας ETS-7000. Εισάγεται δύο 74LS76A JK Flip-Flops (δύο FFs σε κάθε ολοκληρωμένο) στην τρίτη στήλη της πλακέτας και ένα 74LS10, NAND τριών εισόδων, στην τέταρτη στήλη. Η διάταξη και η αριθμηση των pins των ολοκληρωμένων φαίνεται πιο κάτω.

1. Χρησιμοποιήστε τους διακόπτες S0 και S1 ως τις εισόδους *CLRB* και *MOD10*, αντίστοιχα. Το σήμα *CLK* θα το πάρετε από την έξοδο του αντιστροφέα 74LS04 του οποίου η είσοδος θα ενωθεί στην έξοδο OUT της γεννήτριας παλμών, αφού πρώτα έχετε επιλέξει τετραγωνικό παλμό και θέσετε το πλάτος κατάλληλα (μεταξύ $\frac{1}{4}$ και $\frac{1}{2}$). Χρησιμοποιήστε τους λαμπτήρες L7, L6, L5 και L4 για τις εξόδους CNT(3:0). Επίσης, δείξτε τις εξόδους σας στο δεξιά SSD ενώνοντας το D1 στο GND, το L4 στο A, το L5 στο B, το L6 στο C και το L7 στο D του D2. Σχεδιάστε αυτές τις συνδέσεις καθώς και τις αναθέσεις των pins στο διάγραμμα στο σημειωματάριο σας. Επίσης δείξτε την τροφοδοσία και την γείωση στα αντίστοιχα pins.
2. Θέσετε τη συχνότητα ρολογιού στα 2 Hz και παρατηρήστε την συμπεριφορά του κυκλώματος σε σχέση με τις εισόδους. Καταγράψτε στο σημειωματάριο σας την ακολουθία απαρίθμησης και για τις δύο τιμές του *MOD10* (S1). Σημειώστε ότι το SSD παρουσιάζει μόνο τις τιμές που αντιστοιχούν στα δεκαδικά ψηφία 0 έως 9.
3. Σχολιάστε τα αποτελέσματα και εξηγήστε γιατί αυτό ο μετρητής ονομάζεται μετρητής ριπής.





74LS04 Hex Inverter

Λίστα με στοιχεία και όργανα που θα χρησιμοποιηθούν:

A/A	Qty	Part No	Description
1	1	WK-1	Wire Kit
2	1	74LS04	Hex Inverter
3	1	74LS10	Triple 3-Input NAND Gate
4	2	74LS76A	Dual JK Flip – Flop
5	1	ETS-7000	Digital – Analog Training System unit