

Τελικό Project (20% Συνολικού Βαθμού + Bonus)

Σχεδιασμός Δυαδικού Πολλαπλασιαστή

Υλοποίηση συνδυαστικού και ακολουθιακού κυκλώματος

Εισαγωγή

Στις προηγούμενες εργαστηριακές ασκήσεις παρουσιάστηκε η χρήση συνδυαστικής και ακολουθιακής λογικής για τον σχεδιασμό ψηφιακών συστημάτων και χρησιμοποιήθηκε τόσο το λογισμικό MAX+PLUS II για τον σχεδιασμό και την προσομοίωση λογικής, όσο και υλοποιήσεις με πραγματικά διακριτά και προγραμματιζόμενα στοιχεία. Στην άσκηση αυτή καλείστε να σχεδιάσετε ένα μεγαλύτερο κύκλωμα χρησιμοποιώντας όλη την εμπειρία που αποκομίσατε στις προηγούμενες εργαστηριακές ασκήσεις, καθ' όλη την διάρκεια του εξαμήνου. Στόχος αυτής της άσκησης είναι ο σχεδιασμός ενός κυκλώματος που σαν είσοδο παίρνει δύο μη-προσημασμένους αριθμούς 4-bit και παράγει το γινόμενο τους, ένα δηλαδή μη προσημασμένο αριθμό των 8-bit. Υπάρχουν πολλοί τρόποι για να υλοποιηθεί το πιο πάνω κύκλωμα, πολλοί από τους οποίους αναφέρονται και στο βιβλίο των Mano και Kime. Για μεγαλύτερη εμπάθυνση στο θέμα και για την απόκτηση **απαραίτητων** γνώσεων για την επιτυχή διεκπεραίωση της άσκησης, **καλείστε να διαβάσετε τις παραπομπές που δίνονται πιο κάτω και οι οποίες σας έχουν δοθεί σε φωτοαντίγραφο**. Στο project αυτό θα πραγματοποιήσετε τους σχεδιασμούς σας στο MAX+PLUS II και θα κατεβάσετε τον τελευταίο από αυτούς στην πλακέτα UP2 της Altera. Θα μελετήσετε δύο μεθόδους για υλοποίηση ενός πολλαπλασιαστή δυαδικών αριθμών: μία συνδυαστική με την χρήση ROM (Read-Only Memory) και μία ακολουθιακή ακολουθώντας μια διαδικασία απαρίθμησης.

Το project αυτό είναι διάρκειας 4 εβδομάδων, από τις οποίες μόνο για τις δύο πρώτες θα μπορείτε να δουλέψετε στο εργαστήριο Π102 (εργαστήριο λογισμικού) υπό την επιτήρηση των υπευθύνων του εργαστηρίου. Το υπόλοιπο μέρος θα κληθείτε να το διεκπεραιώσετε σε δικό σας χρόνο, είτε στο εργαστήριο Π102, είτε στο σπίτι σας. Η άσκηση είναι **αυστηρώς προσωπική** και στο τέλος θα υπάρχει προφορική εξέταση για τον κάθε φοιτητή ξεχωριστά. Project τα οποία είναι πανομοιότυπα και, κατά την κρίση των διδασκόντων, είναι προϊόν αντιγραφής θα μηδενίζονται αυτόματα, με πιθανές επιπλέον κυρώσεις για τους εμπλεκόμενους. Η εξέταση του project θα περιλαμβάνει την φόρτωση του σχεδιασμού σας στην πλακέτα UP2 για την επίδειξη της λειτουργίας, την προφορική περιγραφή της λειτουργίας και της διαδικασίας ανάπτυξης του σχεδιασμού καθώς και την απάντηση ερωτήσεων που το αφορούν. Επιπλέον, κάθε φοιτητής καλείται να παραδώσει προσωπική αναφορά με την περιγραφή των σχεδιασμών και την πλήρη διαδικασία που ακολούθησε για την υλοποίηση του καθενός απ' αυτούς. Η βαθμολόγηση για το κάθε μέρος του project θα είναι ως εξής:

- Αναφορά 40 %
- Ορθότητα Λειτουργίας 30 %
- Προφορική Εξέταση 30 %
- Bonus 5% ΕΠΙ ΤΗΣ ΣΥΝΟΛΙΚΗΣ ΒΑΘΜΟΛΟΓΙΑΣ ΓΙΑ ΤΟ ΕΡΓΑΣΤΗΡΙΟ HMY 211.

Ως ημερομηνία παράδοσης της αναφοράς και όλων των σχετικών αρχείων ορίζεται η 20 Δεκεμβρίου.

Σημείωση: Η εξέταση περιλαμβάνει την προσομοίωση του 1^{ου} και 2^{ου} σχεδιασμού στο MAX+Plus II και φόρτωση του 2^{ου} σχεδιασμού στην πλακέτα UP2. Για τον λόγο αυτό την ημέρα εξέτασης θα πρέπει να έχετε μαζί σας (ή να μπορείτε να κατεβάσετε από τον λογαριασμό σας) ΟΛΑ τα ζητούμενα αρχεία δηλαδή σχηματικά και κυματομορφές εισόδου του MAX+Plus II, αρχείο κώδικα C, αρχεία espresso κλπ.

Παραπομπές

Οι πιο κάτω παραπομπές αφορούν το σύγγραμμα, *Introduction to Digital Logic Design*, του John Hayes, Addison-Wesley, 1993. Φωτοαντίγραφα των παραπομπών που πρέπει να μελετήσετε στα πλαίσια του project θα σας δοθούν.

1. Για περιγραφή παραδειγμάτων υλοποιήσεων δυαδικού πολυπλέκτη με ROM δείτε τις σελίδες 73-74. Την μέθοδο αυτή θα την χρησιμοποιήσετε στο μέρος Α αυτού του project.
2. Ο σχεδιασμός ενός δυαδικού πολλαπλασιαστή 2x2 παρουσιάζεται στις σελίδες 340-342 ως ένα παράδειγμα συνδυαστικής λογικής πολλαπλών εξόδων, υλοποιημένων με βασικές πύλες (Για περισσότερες λεπτομέρειες για πίνακες κάλυψης (cover tables και reduced cover tables) δείτε τις σελίδες 316-340 του βιβλίου του Hayes).
3. Ένας σχεδιασμός ενός γενικού $N \times N$ διάδικου πολλαπλασιαστή με μετρητές φαίνεται στον Παράδειγμα 8.8 στις σελίδες 660-664. Είναι ακολουθιακό κύκλωμα που αποτελείται από 3 μετρητές και μια μονάδα ελέγχου (controller) που είναι FSM πέντε καταστάσεων, σχεδιασμένο με one-hot λογική, και υλοποιημένο με στοιχεία SSI και MSI. Η one-hot λογική περιγράφεται αναλυτικά στις σελίδες 656-660. Για το project αυτό δεν θα χρησιμοποιήσετε one-hot λογική αλλά η ανάγνωση θα σας βοηθήσει να κατανοήσετε το παράδειγμα 8.8. Μια εναλλακτική μέθοδος σχεδιασμού πολλαπλασιαστή με μετρητές, με την χρήση Αλγοριθμικών Μηχανών Πεπερασμένων Καταστάσεων (ASM διάγραμμα) παρουσιάζεται στις σελίδες 684-691. Στο project αυτό θα χρησιμοποιήσουμε μια παραλλαγή της προσέγγισης αυτής. Για περισσότερες πληροφορίες για ASM δείτε και τις σελίδες 674 – 681, καθώς και τις σελίδες 684-691.

Εκφώνηση/Καθορισμός Project

Σημείωση: Τα πιο κάτω αρχεία είναι χρήσιμα για την διεκπεραίωση του Project αυτού. Πριν αρχίσετε αντιγράψετε τα αρχεία αυτά από την ιστοσελίδα τους μαθήματος στο κατάλογο στον οποίο θα δουλέψετε (π.χ. lab6).

lab6a.scf

lab6b_5000.scf

lab6b_50000.scf

Μέρος Α : Υλοποίηση δυαδικού πολλαπλασιαστή με ROM

Ένας απλός τρόπος περιγραφής ενός πολλαπλασιαστή είναι σαν ένα συνδυαστικό κύκλωμα το οποίο παίρνει δύο αριθμούς N -bit ως είσοδο και παράγει το γινόμενο τους σαν έξοδο, δηλαδή ένα αριθμό $2N$ -bits. Αυτή η προσέγγιση είναι πρακτική για N μικρότερα από 10. Μία ROM είναι μία καλή επιλογή για την υλοποίηση ενός πολλαπλασιαστή επειδή μας δίνει το ζητούμενο κύκλωμα με την χρήση ενός μόνο ολοκληρωμένου. Από την άλλη ο τρόπος υλοποίησης (διαβάστε τις αντίστοιχες παραγράφους που σας έχουν δοθεί από το βιβλίο του Hayes) απαιτεί την εισαγωγή των κατάλληλων τιμών στις θέσεις της ROM. Αφού, λοιπόν, στο κύκλωμα έχουμε $2N$ εισόδους, τότε θα υπάρχουν 2^{2N} ελαχιστόροι ή διευθύνσεις μνήμης στην ROM και κάθε διεύθυνση θα πρέπει να περιέχει έναν αριθμό $2N$ bits, ο οποίος θα αντιστοιχεί στο γινόμενο των εισόδων. Για την συγκεκριμένη άσκηση θέτουμε το $N=4$ και άρα χρειαζόμαστε μία ROM με 256 λέξεις των 8 bits. Για να εξομοιώσουμε την ROM 256×8 θα χρησιμοποιήσουμε την συνάρτηση `lpm_rom` του MAX+PLUS II. Το σχηματικό σας στον Graphic Editor θα αποτελείται από αυτή την συνάρτηση, δύο θύρες εισόδου και μία θύρα εξόδου. Για την προσομοίωση θα πρέπει να δημιουργήσετε ένα αρχείο το οποίο θα περιέχει τα περιεχόμενα της ROM σε συγκεκριμένη μορφή. Ακολουθήστε την διαδικασία που περιγράφεται πιο κάτω:

a) Κατασκευή σχηματικού με τον Graphic Editor

1. Δημιουργήστε ένα νέο αρχείο στον graphic editor και αποθηκεύσετε το ως **lab6a.gdf** στο κατάλογο με το όνομα **lab6** στον χώρο σας στο nicosia.

2. Εισάγεται τα ακόλουθα στοιχεία κυκλώματος: **wire**, **input**, **output** και **lpm_rom**.

Το στοιχείο **wire** χρησιμοποιείται μόνο για την μετονομασία (renaming) ενός κόμβου ή ενός διαύλου και δεν έχει καμία ιδιαίτερη λογική συμπεριφορά..

Μόλις εισάγεται την **lpm_rom**, με το Enter Symbol, θα ανοίξει το παράθυρο διαλόγου **Edit Ports/Parameters**. Αφήστε το παράθυρο ανοιχτό και προχωρήστε στο επόμενο βήμα για να καθορίσετε την ROM.

3. Στο πεδίο **Name**: επιλέξτε **inclock** και αλλάξτε την κατάσταση σε **Unused**. Κάντε το ίδιο και για τα **outclock** και **memenab**.

Για τον σχεδιασμό αυτό θέλουμε ασύγχρονο διάβασμα της μνήμης (unregistered), δηλαδή οι θύρες διευθύνσεων και εξόδου δεν χρειάζονται ρολόι σαν είσοδο. Η ROM θα δουλεύει πάντα και έτσι μια θύρα ενεργοποίησης λειτουργίας (enable) δεν είναι αναγκαία.

4. Στο πεδίο **Parameters**: καθορίστε τις τιμές των παραμέτρων σύμφωνα με το σχεδιασμό που επιθυμείτε.

LPM_WIDTH: Μέγεθος θύρας εξόδου.

LPM_WIDTHHAD: Μέγεθος θύρας εισόδου για διευθύνσεις.

LPM_NUMWORDS: Αριθμών λέξεων στην μνήμη.

LPM_FILE: Καθορίστε το στο

"<χώρος εργασίας>\lab6\ROM_file.mif".

Το αρχείο αυτό περιέχει τα περιεχόμενα της ROM και θα δημιουργηθεί αργότερα.

LPM_ADDRESS_CONTROL και **LPM_OUTDATA**: θέστε τα σε "UNREGISTERED".

Ανατρέξτε στην βοήθεια του MAX+PLUS II για την LPM_ROM για λεπτομερή περιγραφή των παραμέτρων.

5. Αντιγράψτε το **input** και το **wire** ώστε να δημιουργήσετε δύο σύμβολα από το καθένα.
6. Μετονομάστε τις θύρες εισόδου, **A[3..0]** και **B[3..0]**, και την θύρα εξόδου σε **P[7..0]**.
7. Συνδέστε κάθε θύρα **input** σε ένα σύμβολο **wire** με γραμμή διαύλου.

Με την σύνδεση μίας θύρας εισόδου σε ένα στοιχείο **wire**, ο κόμβος αυτός μπορεί αν μετονομαστεί ώστε να μπορεί να συνδεθεί με την είσοδο διεύθυνσης (address) του συμβόλου της ROM.

8. Συνδέστε την έξοδο κάθε **wire** στην είσοδο της ROM μετονομάζοντας τους κόμβους

Οι εισόδοι A0 έως A3 πρέπει να συνδεθούν στις εισόδους 0 έως 3 της ROM. Οι εισόδοι B0 έως B3 πρέπει να συνδεθούν στις εισόδους 4 έως 7 της ROM.

9. Συνδέστε της έξοδο της ROM στην θύρα εξόδου, P[7..0].
10. Αποθηκεύσετε και ελέγξτε τον σχεδιασμό. Κρατήστε μία εκτύπωση του σχηματικού σας.

Ο compiler θα δώσει σφάλμα που αφορά το αρχείο με το περιεχόμενο της μνήμης. Αγνοήστε το μήνυμα αφού το αρχείο αυτό θα δημιουργηθεί στην επόμενη παράγραφο πριν εκτελεστεί η προσομοίωση.

11. Μεταφράστε τον σχεδιασμό με την επιλογή για χρονική ανάλυση.

Αν σας δώσει κάποιο λάθος που αφορά την οικογένεια του ολοκληρωμένου που επιλέξατε και το γεγονός ότι δεν υποστηρίζει ROM δοκιμάστε να αλλάξετε την οικογένεια του ολοκληρωμένου σε FLEK10K από το μενού **Assign → Device**. Δοκιμάστε να ξαναμεταφράσετε το κύκλωμα.

b) Προσομοίωση

1. Αν προσπαθήσετε να υπολογίσετε και να βάλετε «με το χέρι» 256 τιμές, οι οποίες θα αντιπροσωπεύουν τα περιεχόμενα των θέσεων μνήμης για την ROM, θα αντιληφθείτε ότι δεν είναι μόνο χρονοβόρο αλλά και επισφαλές όσο αφορά στην ορθότητα του αποτελέσματος. Για τον λόγο αυτό θα πρέπει να γράψετε ένα πρόγραμμα (περίπου 15 γραμμές σε C, μαζί με τα σχόλια) για να δημιουργήσετε το αρχείο που θα περιέχει τα περιεχόμενα της ROM, το οποίο θα ονομάζεται `ROM_file.mif`, και θα έχει την ακόλουθη μορφή:

```
-- memory initialization file
-- for 4-bit multiplier

WIDTH = 8;
DEPTH = 256;

ADDRESS_RADIX = HEX;
DATA_RADIX = HEX;

CONTENT BEGIN

00:00;
01:00;
...
FE:D2;
FF:E1;

END;
```

Αυτό το αρχείο περιλαμβάνει δύο γραμμές σχολίων, 4 γραμμές που καθορίζουν τον τύπο των δεδομένων που αποθηκεύει η ROM, μία γραμμή που δηλώνει την αρχή των περιεχομένων της μνήμης την οποία ακολουθούν 256 γραμμές με τα περιεχόμενα της μνήμης (μία γραμμή για κάθε θέση μνήμης). Κάθε γραμμή περιγράφει (σε δεκαεξαδική μορφή) τα περιεχόμενα κάθε θέσης μνήμης. Η άνω-κάτω τελεία (:) χρησιμοποιείται για να ξεχωρίζεται η διεύθυνση της ROM από το περιεχόμενο της θέσης στην συγκεκριμένη διεύθυνση.

Το ερωτηματικό (;) είναι απαραίτητο για κάθε γραμμή. Έτσι αριστερά από το : βρίσκονται οι δύο παράγοντες του πολλαπλασιασμού (πολλαπλασιαστέος και πολλαπλασιαστής), ενώ δεξιά από το : παρουσιάζεται το γινόμενο τους με δύο ψηφία, όλα σε δεκαεξαδική μορφή. Το τέλος του αρχείου δηλώνεται με το `END;`.

2. Κρατήστε μίαν εκτύπωση του κώδικά σας για να την συμπεριλάβετε στην αναφορά σας. Αποθηκεύσετε το παραγόμενο αρχείο ως `ROM_file.mif` στον κατάλογο σας `lab6`.
Η τοποθεσία του αρχείου πρέπει να συνάδει με αυτή που έχετε δώσει στην παράμετρο `LPM_FILE`: στο βήμα a) 4 στην σελίδα 3.
3. Αντιγράψτε το αρχείο προσομοίωσης , `lab6a.scf`, στον κατάλογο σας `lab6`.
4. Ανοίξτε την εφαρμογή Simulator του MAX+PLUS II.
5. Προσομοιώστε τον σχεδιασμό για 800ns με την χρήση του αρχείου `lab6a.scf` ως είσοδο.
6. Ανοίξτε το αρχείο `.scf` για να παρατηρήσετε τα αποτελέσματα της προσομοίωσης. Κρατήστε εκτύπωση των αποτελεσμάτων.
7. Μετακινήστε της κυματομορφές με τα βελάκια αριστερά και δεξιά. Καθορίστε την καθυστέρηση της ROM βρίσκοντας την διαφορά μεταξύ του χρόνου αλλαγής της εισόδου και του χρόνου εμφάνισης της σωστής τιμής στην έξοδο.
8. Μελετήστε τις κυματομορφές εξόδου και σχολιάστε τα αποτελέσματα, συμπεριλαμβάνοντας και τις τιμές της τυπικής καθυστέρησης που βρήκατε στο βήμα 7. Κλείστε όλα τα ανοικτά παράθυρα στο MAX+PLUS II.

Μέρος Β : Μια Ακολουθιακή Υλοποίηση

Μια υλοποίηση πολλαπλασιαστή με ROM καθίσταται μη πρακτική για μεγάλες τιμές του N . Για παράδειγμα μια ROM υλοποίηση πολλαπλασιασμού δύο ακεραίων 16-bit χρειάζεται πάνω από 4 τρισεκατομμύρια λέξεις των 32 bit (2^{32}), ή αλλιώς 16 Gigabyte μνήμης. Στις παραπομπές 3 και 4 του Hayes, υπάρχουν αρκετές προσεγγίσεις για ακολουθιακά κυκλώματα πολλαπλασιαστών, τα οποία απαιτούν λιγότερη μνήμη από την υλοποίηση με ROM αλλά χρειάζονται περισσότερο χρόνο εκτέλεσης.

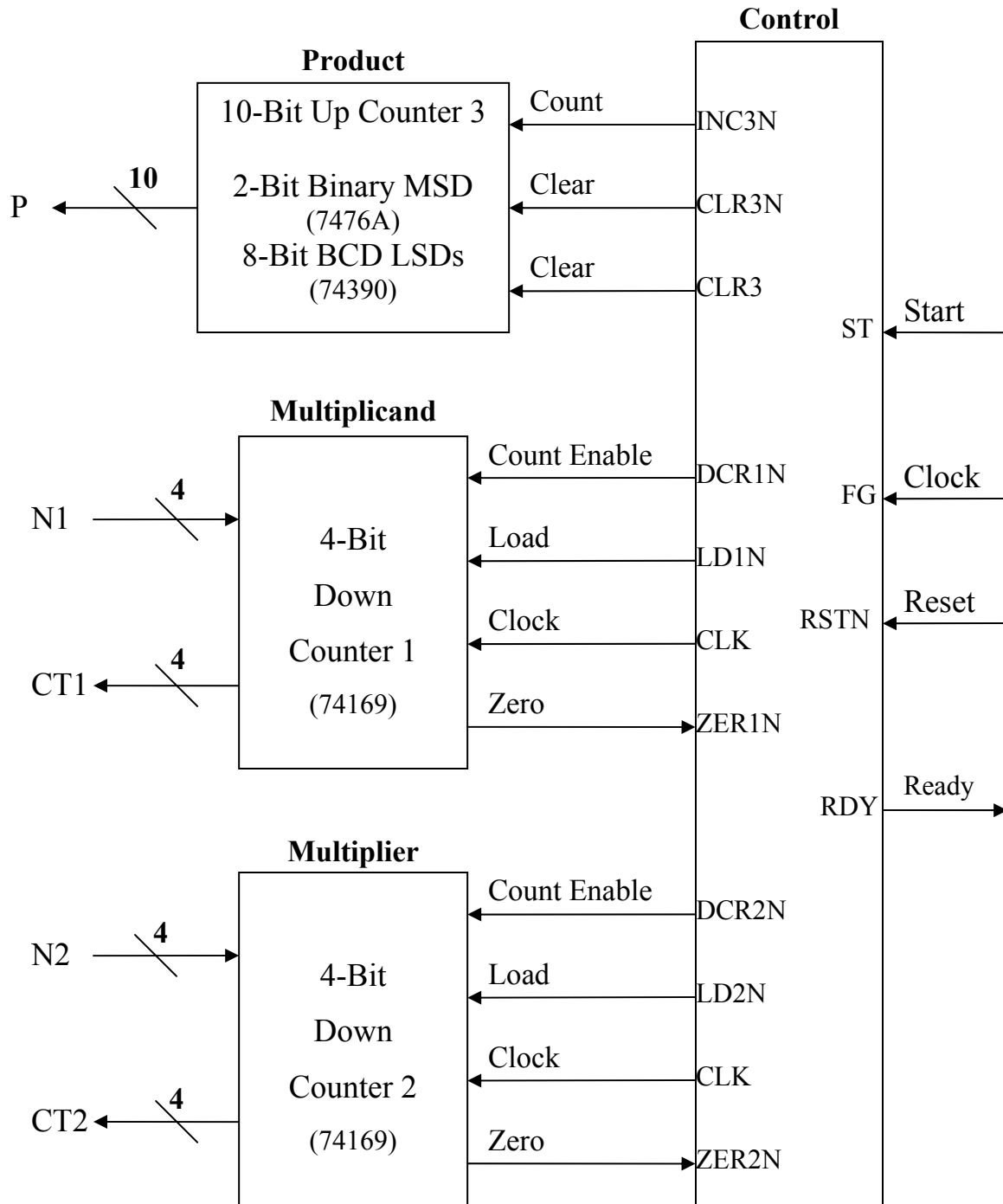
Ο σχεδιασμός με μετρητές (counter-based design) στην παραπομπή 3 αφορά στην χρήση τριών μετρητών : ενός 2N-bit μετρητή προς τα πάνω (up-counter), για το αποτέλεσμα, και δύο N-bit μετρητές προς τα κάτω (down-counters), ένα για κάθε ένα από τους πολλαπλασιαστή και πολλαπλασιαστέο. Είναι βασισμένος στην παρατήρηση ότι ο πολλαπλασιασμός είναι μια επαναλαμβανόμενη πρόσθεση και ότι η πρόσθεση με την σειρά της είναι μια επαναλαμβανόμενη λειτουργία αύξησης. Θα υλοποιήσουμε και θα προσομοιώσουμε μια παραλλαγή αυτού του είδους του πολλαπλασιαστή με την χρήση του MAX+PLUS II.

Ένα μπλοκ-διάγραμμα του πολλαπλασιαστή με μετρητές φαίνεται στην σελίδα 7. Το γινόμενο παράγεται στον μετρητή 3, ο οποίος αποτελείται από ένα 74390 (δύο μετρητές BCD), το οποίο χρησιμοποιήσαμε και στην εργαστηριακή άσκηση 5 και ακολουθείται από ένα μετρητή ριπής 2-bit υλοποιημένο με ένα 7476A (δύο JK flip-flop) . Αυτός ο σύνθετος μετρητής έχει εύρος από το 0 έως το 399 (άρα είναι ένας Modulo 400) το οποίο είναι αρκετά μεγάλο για να αναπαραστήσει το μεγαλύτερο γινόμενο το οποίο μπορεί να υπάρξει με αριθμό 8-bit δηλαδή $15 \times 15 = 225$.

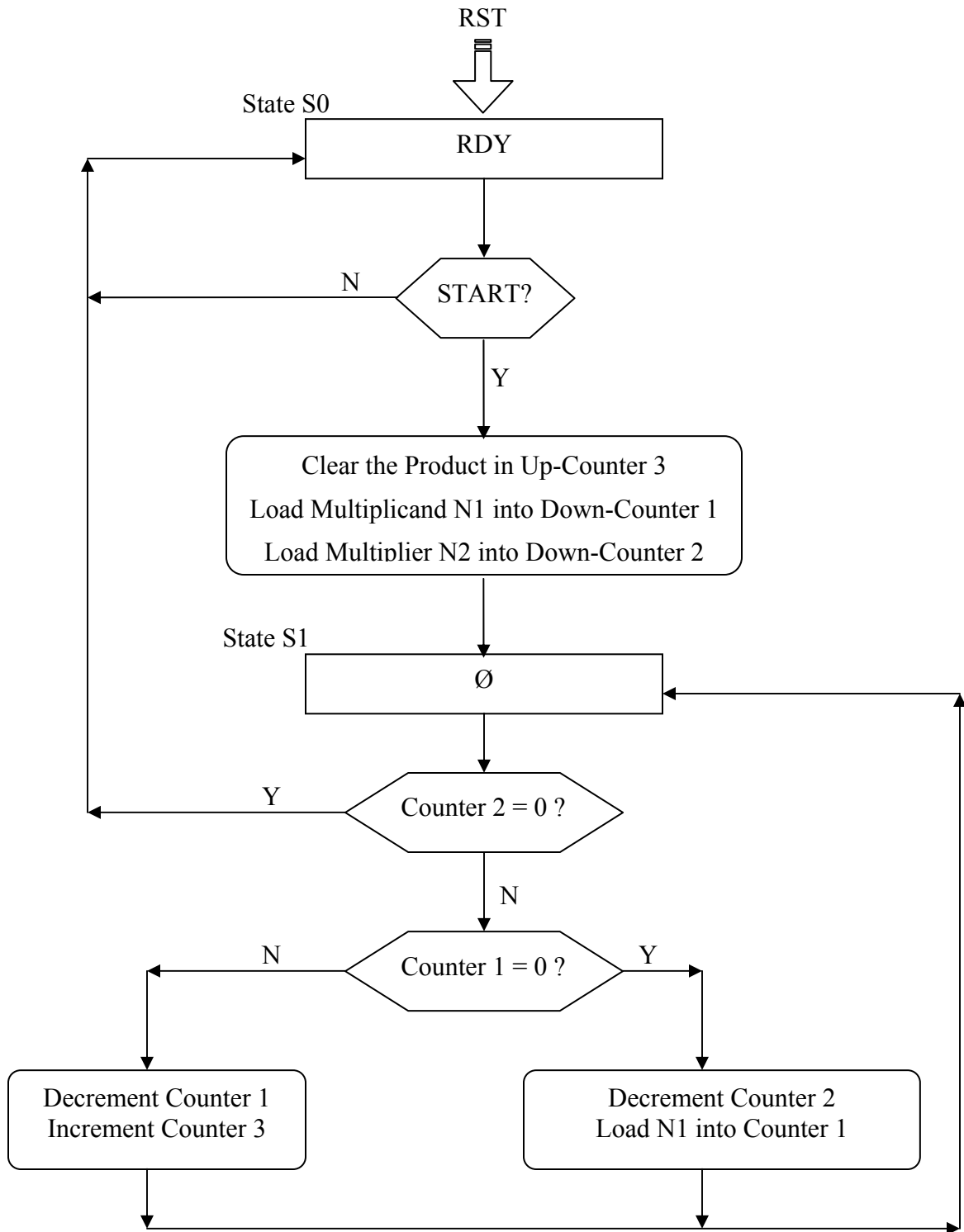
Δύο 74169 (δυναμικοί 4-bit μετρητές προς τα πάνω και προς τα κάτω) ρυθμίζονται για να χρησιμοποιηθούν ως οι προς τα κάτω μετρητές 1 και 2, και αντιστοιχούνται στους πολλαπλασιαστέο και πολλαπλασιαστή. Θεωρούμε ότι οι τιμές τόσο του πολλαπλασιαστέου $N1[3..0]$, όσο και του πολλαπλασιαστή $N2[3..0]$ παραμένουν σταθερές κατά την διάρκεια του πολλαπλασιασμού, έτσι ώστε να μην χρειάζονται επιπρόσθετοι καταχωρητές. Η λειτουργία του μετρητή 74169 μπορεί να εξαχθεί άμεσα από το σύμβολο του, το οποίο πρέπει να χρησιμοποιήσετε στο σχηματικό σας στον Graphic Editor.

Η μονάδα ελέγχου (control unit) είναι μια μηχανή πεπερασμένων καταστάσεων (Finite State Machine - FSM) η οποία υλοποιεί τον αλγόριθμο πολλαπλασιασμού σύμφωνα με το ASM διάγραμμα της εικόνας 2 στην σελίδα 8. Σημειώστε ότι αυτό το ASM διάγραμμα, περιέχει δύο μόνο καταστάσεις, σε αντίθεση με αυτό της σελίδα 684 του Hayes, που χρησιμοποιεί πέντε. Ο controller που παρουσιάζεται στο Hayes σελίδα 690 χρησιμοποιεί 5 flip-flops σε ένα σχεδιασμό one-hot, ενώ ο δικός μας σχεδιασμός χρειάζεται ένα μόνο flip-flop.

Η υλοποίηση στο MAX+PLUS II για την προσομοίωση αυτού του σχεδιασμού αποτελείται από τρία σχηματικά διαγράμματα δομημένα σε δύο ιεραρχικά επίπεδα. Το υψηλότερο επίπεδο (1^ο επίπεδο) αντιστοιχεί πλήρως στο μπλοκ-διάγραμμα της εικόνας 1 στην σελίδα 7, το οποίο συμπεριλαμβάνει τους 74169 μετρητές 1 και 2, ένα σύμβολο για τον σύνθετο μετρητή 3 και ένα σύμβολο για τον ελεγκτή. Θα δημιουργήσετε ανεξάρτητα τα σύμβολα για το μετρητή 3 καθώς και για τον ελεγκτή, στο αμέσως χαμηλότερο (2^ο) επίπεδο ιεραρχίας και στην συνέχεια θα δημιουργήσετε τα σύμβολα τα οποία θα εισάγετε στο σχηματικό του υψηλότερου (1^{ου}) επιπέδου ιεραρχίας.



Εικόνα1. Μπλοκ-Διάγραμμα Πολλαπλασιαστή με μετρητές



Εικόνα 2. ASM διάγραμμα για τον Πολλαπλασιαστή με μετρητές

Διαβάστε προσεκτικά τις παραπομπές που σας έχουν δοθεί για να κατανοήσετε πλήρως την λογική συμπεριφορά του πολλαπλασιαστή. Παρακάτω φαίνονται τα ολοκληρωμένα τα οποία είναι διαθέσιμα για την υλοποίηση του κυκλώματος αυτού. Αυτά τα ολοκληρωμένα πρέπει να σας είναι οικεία από τις προηγούμενες εργαστηριακές ασκήσεις. Δεν επιτρέπεται η χρήση άλλων ολοκληρωμένων για την υλοποίηση του πολλαπλασιαστή σας. Τα πέντε ολοκληρωμένα στα οποία αναγράφεται ακριβής περιγραφή της εφαρμογής που έχουν στην άσκηση ΔΕΝ μπορούν να χρησιμοποιηθούν για άλλο σκοπό.

IC	Περιγραφή	Εφαρμογή
74LS00	Quad 2-input NAND	
74LS00	Quad 2-input NAND	
74LS02	Quad 2-input NOR	
74LS04	Hex Inverter	
74LS08	Quad 2-input AND	
74LS10	Triple 3-input NAND	
74LS20	Dual 4-input NAND	
74LS32	Quad 2-input OR	
74LS74A	Dual D Flip-Flop	
74LS76A	Dual JK Flip-Flop	Περισσότερα σημαντικά 2 bits του σύνθετου μετρητή 3
74LS76A	Dual JK Flip-Flop	
74LS86	Quad 2-input XOR	
74LS138	1 to 8 Decoder	
74LS153	Dual 4-input MUX	
74LS157	Quad 2-input MUX	
74LS169B	4-bit Up/Dn Counter	Πολλαπλασιαστής, down-counter 1
74LS169B	4-bit Up/Dn Counter	Πολλαπλασιαστής, down-counter 2
74LS175	Quad D Flip-Flop	1 για έλεγχο καταστάσεων, 2 για το Start, 1 έξτρα
74LS194	4-bit Shift Register	
74LS390	Dual Decade Counter	Υπόλοιπα 8 bits (2 ψηφία BCD) για τον σύνθετο μετρητή 3
MC14584	Hex CMOS Schmitt	

c) Σχεδιασμός Ελεγκτή Πολλαπλασιαστή

Ο ελεγκτής είναι ένα ακολουθιακό FSM του οποία το διάγραμμα ASM φαίνεται στην εικόνα 2 στην σελίδα 8. Αφού το διάγραμμα έχει μόνο δύο καταστάσεις τότε απαιτείται η χρήση ενός μόνο flip-flop, με έξοδο QA. Θα αντιστοιχήσουμε τώρα τις καταστάσεις S0 και S1 στις τιμές του QA ως QA=0 και QA=1, αντίστοιχα. Αφού η έξοδος του ελεγκτή CLR3 είναι απλά το συμπλήρωμα του CLR3N, τότε αφαιρούμε την έξοδο CLR3 από την διαδικασία σχεδιασμού, σε αυτή την φάση, σημειώνοντας απλά ότι αυτή χρειάζεται, και στο τελικό σχεδιασμό πρέπει να υλοποιηθεί με κάποιο τρόπο.

1. Ανατρέξτε στα σύμβολα των ολοκληρωμένων των μετρητών για να δείτε τις λειτουργίες και τις προτεραιότητες των διαφόρων σημάτων ελέγχου. Για παράδειγμα, σημειώστε ότι ένα ενεργό σήμα LD1N (τιμή 0) έχει μεγαλύτερη προτεραιότητα και άρα αναιρεί το σήμα ενεργοποίησης μέτρησης (count-enable) DCR1N. Το γεγονός αυτό έχει σαν αποτέλεσμα την σημείωση του σαν όρου αδιαφορίας (don't care) στον πιο κάτω πίνακα, ο οποίος συμβολίζεται με παύλα (-).
2. Χρησιμοποιώντας τις αναθέσεις των καταστάσεων και τις καταστάσεις λειτουργίας των μετρητών, ακολουθήστε όλα τα μονοπάτια στο διάγραμμα ASM και συμπληρώστε το αντίστοιχο κομμάτι του πίνακα καταστάσεων πιο κάτω. Οι έξοδοι είναι χωρισμένοι με έντονη γραμμή από τις εισόδους, ενώ το DA συμβολίζει την επόμενη κατάσταση στο FSM, QA(t+1). Συμπληρώστε τον πίνακα και για τους 16 συνδυασμούς των τεσσάρων μεταβλητών εισόδου (προσέξτε ότι οι πρώτοι 8 συνδυασμοί σας δίνονται, και τους υπόλοιπους μπορούν να αναπαρασταθούν από τις 3 τελευταίες γραμμές). Όπως αναφέρθηκε και πριν η παύλα αντιστοιχεί σε όρο αδιαφορίας, συμπεριλαμβάνει δηλαδή και τις δύο λογικές τιμές, ενώ στην αντίστοιχη συνάρτηση δηλώνει την έλλειψη μιας μεταβλητής από ένα όρο σε ένα άθροισμα γινομένων.

QA	START	ZER1N	ZER2N	DA	LD1N	DCR1N	LD2N	DCR2N	CLR3N	INC3N
0	0	-	-	0	1	1	1	1	1	1
0	1	-	-	1	0	-	0	-	0	-
1										
1										
1										

3. Ο ολοκληρωμένος πίνακας πιο πάνω, δηλώνει για καθεμία από τις επτά εξόδους, την λογική συνάρτηση που της αντιστοιχεί σε σχέση με τις μεταβλητές εισόδου, σε μορφή αθροίσματος γινομένων (SOP). Παρόλο που αυτή η μορφή χρησιμοποιείται συνήθως σε υλοποιήσεις με PLA, μπορεί επίσης να χρησιμοποιηθεί και για λογική επιπέδου πύλης. Η λογική μπορεί να απλοποιηθεί περισσότερο με την χρήση του προγράμματος ελαχιστοποίησης λογικών εκφράσεων *espresso*. Με την χρήση της εφαρμογής Text Editor, δημιουργήστε ένα αρχείο με όνομα **lab6b.espin** που να περιέχει τον πίνακα σας από το βήμα 2, με την πιο κάτω μορφή (συμπληρώστε τις τιμές του πίνακα που λείπουν, στο τέλος της σελίδας, βασισμένοι στον πίνακα που συμπληρώσατε πιο πάνω) :

```

.i 4
.o 7
.ilb QA START ZER1N ZER2N
.ob DA LD1N DCR1N LD2N DCR2N CLR3N INC3N

00-- 0111111
01-- 10-0-0-
1
1
1
.e

```

Οι πρώτες δύο γραμμές καθορίζουν τον αριθμό των μεταβλητών των εισόδων και των εξόδων, αντίστοιχα, ενώ οι επόμενες δύο δίνουν τα ονόματα των μεταβλητών με την σειρά που εμφανίζονται στον πίνακα. Οι επόμενες γραμμές (γραμμές δεδομένων), είναι αντιγραμμένες ακριβώς, από τον πίνακα του βήματος 2. Το πρόγραμμα espresso μπορείτε να το κατεβάσετε από την ιστοσελίδα του μαθήματος μαζί με την σχετική τεκμηρίωση (ένα εγχειρίδιο χρήσης και μια περιγραφή της μορφής του αρχείου εισόδου/εξόδου). Για να το εκτελέσετε πρέπει πάτε στην γραμμή εντολών του DOS (start→run→cmd). Αποθηκεύσετε το αρχείο με επέκταση .espιn και χρησιμοποιήστε τον ελαχιστοποιητή ως εξής:

```
espresso I ab6b. espιn
```

Το espresso θα επιστρέφει την απλοποιημένη έξοδο στην οθόνη, στην ίδια μορφή όπως και η είσοδος, αλλά συνήθως θα περιέχει λιγότερους όρους γινομένου και άρα περισσότερες παύλες, που με την σειρά τους θα δώσουν λιγότερες πύλες και λιγότερες διασυνδέσεις μεταξύ των πυλών.

Επιπλέον απλοποίηση, μπορεί να αποκτηθεί κάποιες φορές αν η πολικότητα μίας ή περισσότερων εξόδων αλλάξει. Αν εκτελέσετε το espresso με την επιλογή -Dopo, το πρόγραμμα θα δοκιμάσει διάφορους συνδυασμούς πολικότητας των εξόδων και θα δώσει μια «βελτιωμένη» διάταξη, πάντα με την ίδια μορφή όπως η είσοδος. Επιπλέον, θα σημειώσει ως σχόλιο μια γραμμή με τις πολικότητες των εξόδων ως εξής:

```
#. phase 0111010
```

Ένα μηδέν στην πιο πάνω συμβολοσειρά δηλώνει ότι οι τιμές των συναρτήσεων θα πρέπει να υπολογίζονται για την συμπληρωματική έξοδο, ενώ ένας άσπος ότι η συνάρτηση που προκύπτει αφορά την κανονική έξοδο. Έτσι μπορείτε να υλοποιήσετε το κύκλωμα σας με τις συναρτήσεις που προκύπτουν, αγνοώντας την πολικότητα, και μετά να προσθέσετε αντιστροφείς στις αντίστοιχες εξόδους. Δοκιμάστε την επιλογή .phase με την πιο κάτω εκτέλεση του προγράμματος:

```
espresso -Dopo I ab6B. espιn
```

Για να κρατήσετε την έξοδο του προγράμματος σε ένα αρχείο, τρέξετε το espresso ως εξής:

espresso –Dopo l'ab6B.espin > l'ab6Bout.espin

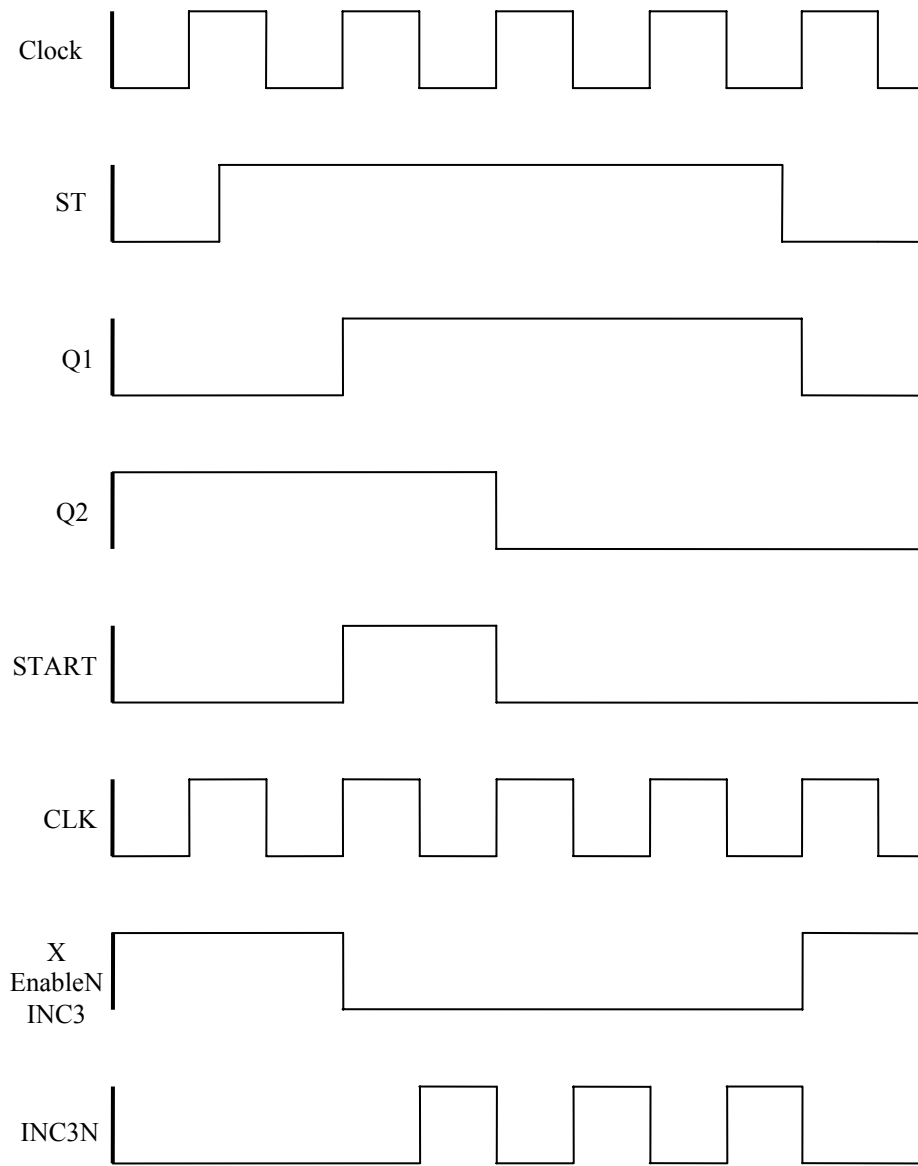
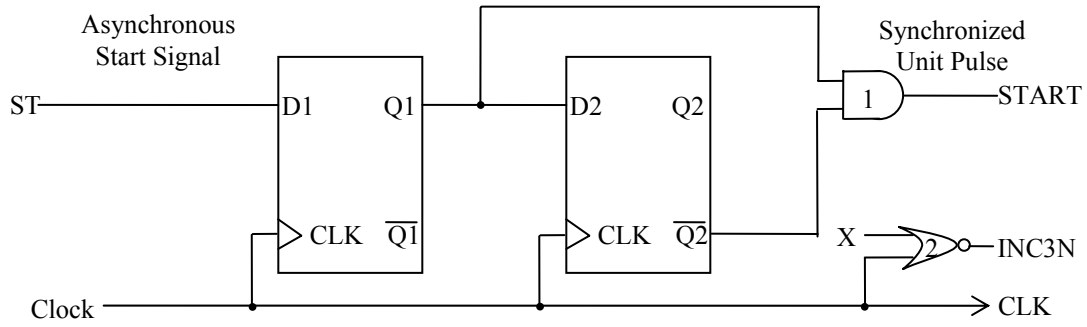
Κρατήστε εκτυπώσεις από τα αρχεία εισόδου και εξόδου του espresso για να τα συμπεριλάβετε στην αναφορά σας. Παρατηρώντας τις εξόδους του espresso σχηματίστε το ανάλογο κύκλωμα χρησιμοποιώντας τον ελάχιστο αριθμό πυλών, από αυτές που σας δίνονται στην λίστα της σελίδας 9 και δεν έχουν άλλη σαφή εφαρμογή. Σημειώστε ότι μπορείτε να χρησιμοποιήσετε μόνο τρία ολοκληρωμένα για την μονάδα ελέγχου (control unit).

4. Για διάφορους πρακτικούς λόγους το σήμα Start πρέπει να είναι συγχρονισμένο με το ρολόι καθώς και όλες οι λειτουργίες που εκτελεί το κύκλωμα σας, άρα και η μονάδα ελέγχου. Για τον λόγο αυτό θα πρέπει να εξασφαλίσετε τον συγχρονισμό του σήματος Start με το ρολόι αλλά και ότι ο μετρητής 3 (σύνθετος, προς τα πάνω) μετράει συγχρονισμένα προς το ρολόι. Παρατηρήστε ότι ο μετρητής αυτός δεν έχει καθόλου σήμα ενεργοποίησης μέτρησης, αλλά μετράει όποτε το ρολόι του γίνει 0 από 1 (είναι αρνητικά ακμοπυροδοτούμενο). Έτσι θα παρουσιαστεί ένα κύκλωμα το οποίο διευθετεί τα δύο αυτά θέματα, συγχρονίζει δηλαδή όλα τα σήματα εξόδου της μονάδας ελέγχου, με το ρολόι. Το κύκλωμα αυτό φαίνεται στο σχήμα 3 της σελίδας 13.

Ως σήμα Clock ονομάζουμε το εξωτερικό ρολόι το οποίο δίνεται ως είσοδος στην προσομοίωση του σχεδιασμού μας. Το σήμα CLK είναι το σήμα που δίνουμε ως έξοδο της μονάδας ελέγχου (το CLK δηλαδή στην εικόνα 1). Χρησιμοποιούμε την διάταξη με τα δύο flip-flops για να διασφαλίσουμε ότι το σήμα Start που θα δώσει η μονάδα ελέγχου θα είναι συγχρονισμένη με το ρολόι εισόδου.

Για τον μετρητή 3 (up-counter) παρατηρούμε ότι είναι αρνητικά ακμοπυροδοτούμενος και επιπλέον δεν διαθέτει σήμα ενεργοποίησης μέτρησης (count enable). Το πρόβλημα που δημιουργείτε είναι ότι η εντολή που θα δώσει η μονάδα ελέγχου για αύξηση του εν λόγω μετρητή (το INC3N) δεν θα είναι συγχρονισμένο με το ρολόι του συστήματος. Για τον λόγο αυτό τοποθετούμε την πύλη NOR με αριθμό 2. Έτσι η πύλη θα επιτρέψει την αποστολή εντολής INC3N μόνο στην θετική ακμή του ρολογιού και επιπλέον, λόγω της αντιστροφής θα πυροδοτεί άμεσα τον μετρητή 3. Με τον τρόπο αυτό διασφαλίζουμε ότι ο μετρητής 3 θα μετρά προς τα πάνω ακριβώς την ίδια χρονική στιγμή που και ο μετρητής 1 μετράει προς τα κάτω. Το σήμα X που παίρνει σαν 2^η είσοδο η πύλη NOR είναι ακριβώς το σήμα INC3N όπως το έχουμε πάρει από την απλοποίηση της συνάρτησης που κάναμε με το *espresso*. Στην εικόνα 3 φαίνεται η έξοδος INC3N της πύλης NOR και ο τρόπος με τον οποίο θα «πει» στο μετρητή 3 να αρχίσει να μετράει.

5. Χρησιμοποιήστε το σχεδιασμό σας από το βήμα 3 μαζί με την αλλαγές που κάνατε στο βήμα 4 για να σχεδιάσετε την μονάδα ελέγχου. Χρησιμοποιήστε μόνο ολοκληρωμένα από τον πίνακα της σελίδας 9, χωρίς ο αριθμός τους να υπερβαίνει τα τρία. Στα τρία ολοκληρωμένα **δεν** συμπεριλαμβάνονται τα ολοκληρωμένα των οποίων η χρήση έχει σημειωθεί ως συγκεκριμένη. Για παράδειγμα, για το ολοκληρωμένο των D Flip-Flop (74LS175) γίνεται συγκεκριμένη χρήση και δεν πρέπει να υπολογιστεί στα τρία.



Εικόνα 3. Έλεγχος Σημάτων Clock και Start

d) Δημιουργία Σχηματικού με τον Graphic Editor

1. Ανοίξετε ένα νέο αρχείο στον Graphic Editor και αποθηκεύσετε το με το όνομα **lab6b_control.gdf** στον κατάλογο σας με όνομα **lab6**.
2. Εισάγετε τα στοιχεία του κυκλώματος σύμφωνα με τον σχεδιασμό σας από το βήμα a) 5. Οι εισόδοι και οι εξόδοι πρέπει να έχουν τα **ίδια** ονόματα με αυτά της μονάδας ελέγχου στην εικόνα 1 στην σελίδα 6. Ονομάστε τις εισόδους: ZER1N, ZER2N, RSTN, FG και ST και τις εξόδους: CLR3, CLR3N, INC3N, LD1N, DCR1N, LD2N, DCR2N, RDY και CLK.

3. Αντιγράψετε και μετακινήστε τα στοιχεία ώστε να συμφωνούν με τον σχεδιασμό σας.

4. Συνδέστε τα στοιχεία, πάντα σύμφωνα με το σχεδιασμό σας.

Αν στον σχεδιασμό σας, δύο θύρες εξόδου προέρχονται από την ίδια λογική πρέπει να χρησιμοποιήσετε το στοιχείο **wire**, όπως στο μέρος I, για να συνδέσετε τις θύρες εξόδου στον ίδιο κόμβο (σημείο τους κυκλώματος).
5. Ελέγξτε και αποθηκεύσετε το σχηματικό. Δημιουργήστε ένα σύμβολο για το σχηματικό με την επιλογή **Create Default Symbol (File menu)**.
6. Ανοίξετε το σύμβολο **lab5b_control.sym** στον Symbol Editor και αλλάξτε το σχήμα του και τις θέσεις των θυρών εισόδου/εξόδου, αν το επιθυμείτε.
7. Κλείστε το σχηματικό του controller.
8. Ανοίξετε ένα νέο αρχείο στον Graphic Editor και αποθηκεύσετε το ως **lab6b_upctr.gdf**.
9. Εισάγετε τα ακόλουθα στοιχεία κυκλώματος: **74390**, **7476a**, **input**, **output** και **vcc**. Οι θύρες εισόδου και εξόδου πρέπει να έχουν τα ονόματα που φαίνονται μέσα στο διάγραμμα του Up-Counter 3 στην Εικόνα 1, σελίδα 6: Είσοδοι **INC3N**, **CLR3N** και **CLR3** και εξόδοι **P[9..0]**.
10. Φτιάξτε ένα μετρητή 10-bit modulo 400, όπου τα 8 λιγότερα σημαντικά bits να αντιστοιχούν στα δύο ψηφία BCD, όπως είχαμε δείξει στο σχηματικό της εργαστηριακής άσκησης 5, και τα δύο περισσότερο σημαντικά ψηφία να είναι ένας modulo 4 μετρητής ριπής 2 bit υλοποιημένος με JK flip-flops ενωμένα ως T flip-flops.
11. Επαναλάβετε τα βήματα 5-7 όπως πιο πάνω για να ελέγξετε, να αποθηκεύσετε το σχεδιασμό σας και να δημιουργήσετε ένα σύμβολο.
12. Ανοίξετε ένα νέο αρχείο στον Graphic Editor και αποθηκεύσετε το ως **lab6b_top.gdf**.
13. Εισάγετε τα ακόλουθα στοιχεία κυκλώματος: **74169**, **input**, **output** και **gnd**. Επίσης εισάγετε και τα σύμβολα **lab6b_control.gdf** και **lab6b_upctr.gdf** και έχετε δημιουργήσει στα βήματα 5 και 11. Τα σύμβολα εισάγονται με το παράθυρο διαλόγου **Enter Symbol**.
14. Οι θύρες εισόδου και εξόδου πρέπει να έχουν τα ονόματα των εξωτερικών σημάτων του μπλοκ-διαγράμματος της εικόνας 1 στην σελίδα 6: Είσοδοι **RSTN**, **CLK**, **ST**, **N1[3..0]** και **N2[3..0]** και εξόδοι **P[9..0]**, **RDY**, **CT1[3..0]** και **CT2[3..0]**.

15. Συνδέστε τα στοιχεία σύμφωνα με τον σχεδιασμό σας. Ελέγξτε και αποθηκεύστε το σχηματικό. Μεταφράστε το σχεδιασμό σας με την επιλογή για χρονική ανάλυση.

Οι εισοδοί ENTN και LDN στους 74169 πρέπει να είναι στο λογικό 0 για να επιτρέπουν παράλληλη φόρτωση. Οι εισοδοί U/DN και ENPN πρέπει να είναι στο λογικό 0 για να επιτρέπουν την έναρξη της μέτρησης.

e) Προσομοίωση

1. Αντιγράψτε τα αρχεία προσομοίωσης, lab6b_5000.scf και lab6b_50000.scf στον κατάλογο σας lab6.
2. Ανοίξτε την εφαρμογή Simulator. Εκτελέστε την προσομοίωση για 5000ns χρησιμοποιώντας το αρχείο lab6b_5000.scf ως είσοδο. Προσομοιώστε μόνο την λογική λειτουργία.
3. Ανοίξτε το αρχείο με επέκταση .scf για να παρατηρήσετε τα αποτελέσματα της προσομοίωσης. Κρατήστε μια εκτύπωση της κυματομορφής από 0 έως 5000 ns.

Αν τα αποτελέσματα απέχουν από τα αναμενόμενα τότε η χρήση «αφανών» κόμβων (buried nodes) σας βοηθήσει να εντοπίσετε το σημείο που υπάρχει λάθος. Οι αφανείς κόμβοι περιγράφηκαν στην άσκηση 5. Για περισσότερες λεπτομέρειες ανατρέξτε στην βοήθεια του MAX+PLUS II.

4. Μετακινείτε τις κυματομορφές ώστε να μπορέσετε να καθορίσετε την διάρκεια υπολογισμού του γινομένου για τις δύο περιπτώσεις πολλαπλασιασμού (3x2 και 2x3), μετρώντας τον χρόνο από την ακμή καθόδου του σήματος RDY έως την ακμή ανόδου του ίδιου σήματος. Συμπεριλάβετε τις τιμές αυτές στην αναφορά σας και εξηγήστε γιατί είναι διαφορετικές.
5. Εκτελέστε την δεύτερη προσομοίωση με την χρήση του αρχείου lab6b_50000.scf ως είσοδο για 50000 ns. Μελετήστε και σχολιάστε τα αποτελέσματα.

Δοκιμάστε να ζουμάρετε σε συγκεκριμένες περιοχές της κυματομορφής σας για να δείτε λεπτομέρειες για τις μεταβάσεις καταστάσεων. Επίσης, δοκιμάστε να ελέγξετε τον πολλαπλασιαστή σας και για άλλες τιμές, για να επιβεβαιώσετε την ορθότητα του.

6. Κρατήστε μian εκτύπωση για το χρονικό διάστημα : 45000 έως 49000 ns.

Bonus Βαθμολογίας (5% επί του συνολικού βαθμού στο εργαστήριο HMY 211)

Σας ζητείτε να ξαναυλοποιήσετε τον controller του πολλαπλασιαστή στο Μέρος Β με διαφορετικό τρόπο από ότι καθορίζεται στην εκφώνηση. Για να επιτύχετε την επιπλέον αυτή βαθμολογία πρέπει να σχεδιάσετε και να υλοποιήσετε το controller που παρουσιάζεται στην σελίδα 7 και η λειτουργία του παρουσιάζεται στην σελίδα 8, με την χρήση Διαγραμμάτων Πεπερασμένων Καταστάσεων (Finite State Machines – FSMs). Επιπλέον, πρέπει να προσομοιώσετε τον ολοκληρωμένο πολλαπλασιαστή με τα αρχεία που σας δόθηκαν. Δεν υπάρχουν περιορισμοί στον αριθμό ή το είδος των ολοκληρωμένων που θα χρησιμοποιήσετε. Αναλυτικά σας ζητείται να παραδώσετε:

1. Το διάγραμμα FSM του controller
2. Περιγραφή της Διαδικασίας Σχεδιασμού και Υλοποίησης (πίνακας καταστάσεων, εξισώσεις, τύποι flip-flop που χρησιμοποιήσατε κλπ).
3. Υλοποίηση και προσομοίωση του σχεδιασμού σας στο MAX+Plus II. Η προσομοίωση να γίνει όπως και στην αρχική υλοποίηση.
4. Γραπτή αναφορά με τα πιο πάνω που να συμπεριληφθεί στην συνολική αναφορά.

Τελική Αναφορά

Γράψετε δύο με τρεις σελίδες για την περιγραφή και σχολιασμό των αποτελεσμάτων που πήρατε, τις δυσκολίες που συναντήσατε και τις γνώσεις που αποκτήσατε σε αυτό το project. Συμπεριλάβατε όλους τους πίνακες, σχηματικά, κυματομορφές, το αρχείο με τον κώδικα C για τα περιεχόμενα της ROM και το αρχείο του espresso όπως σας έχουν ζητηθεί από την διαδικασία περιγραφής της άσκησης. Κάντε τις κατάλληλες αναφορές στην περιγραφή σας, σε όλα αυτά που έχετε κρατήσει από όλη την διαδικασία. Η αναφορά σας **πρέπει** είναι σε ηλεκτρονική μορφή και να περιλαμβάνει και ένα εξώφυλλο με τον τίτλο της, τα στοιχεία σας και την ημερομηνία παράδοσης. Επίσης, πρέπει να παραδοθεί και μια εκτύπωση της αναφοράς καθώς και ένας ψηφιακός δίσκος (CD) με όλα τα αρχεία που περιγράφονται πιο πάνω.

Τα επισυναπτόμενα πρέπει να περιλαμβάνουν:

- | | | |
|----|-------------------------|---|
| 1. | Σελίδα 2: a) 10 | Σχηματικό Υλοποίησης με ROM. |
| 2. | Σελίδα 3: b) 2 | Κώδικας σε C για παραγωγή αρχείου περιεχομένων ROM. |
| 3. | Σελίδα 3: b) 7 | Προσομοίωση με καθυστερήσεις για την υλοποίηση με ROM. |
| 4. | Σελίδα 3: b) 8 | Μετρημένη καθυστέρηση για ROM. |
| 5. | Σελίδα 10: a) 3 | Τελικές μορφές αρχείων εισόδου και εξόδου από espresso. |
| 6. | Σελίδα 14: b) 5, 11, 15 | Σχηματικά για lab6b_control.gdf, lab6b_upctr.gdf και lab6b_top.gdf. |
| 6. | Σελίδα 16: c) 3 | Χρονική Προσομοίωση του lab6b_top από 0 έως 5000 ns. |
| 7. | Σελίδα 16: c) 4 | Μετρημένοι χρόνοι υπολογισμού για 2x3 και 3x2. |
| 8. | Σελίδα 16: c) 6 | Χρονική Προσομοίωση του lab6b_top από 45000 έως 49000ns. |

Όλα τα σχηματικά του μέρους Β, στα οποία αναφέρεστε στους σχολιασμούς σας, πρέπει να δείχνουν ξεκάθαρα τους αριθμούς των pins και τους κωδικούς των ολοκληρωμένων που αντιστοιχούν στο πραγματικό υλικό για την υλοποίηση του κυκλώματος.