Συνδιαστική Λογική με Πολυπλέκτες και Αποκοδικοποιητές: Σχεδιασμός ενός Πλήρους Αθροιστή

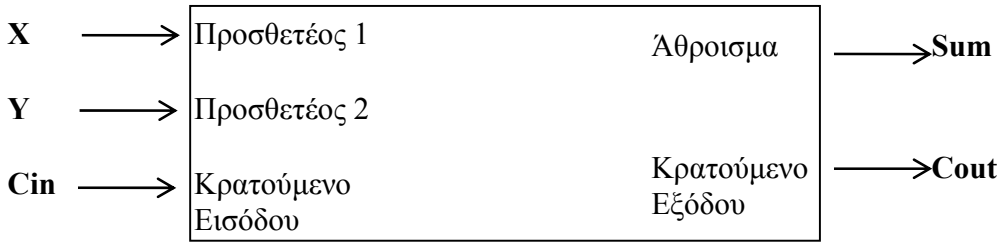
ΕΡΓΑΣΤΗΡΙΟ ΛΟΓΙΣΜΙΚΟΥ 3

Εισαγωγή

Σε προηγούμενα πειράματα έχουμε εξετάσει την χρήση τριών συνόλων πυλών που είναι συναρτησιακά πλήρης για την πραγματοποίηση όλων των συνδυαστικών λογικών πράξεων: 1) πύλες AND, OR, και NOT , 2) πύλες NAND, και 3) πύλες NOR. Έχουμε επίσης εξετάσει την πύλη XOR, η οποία παρόλο που δεν είναι συναρτησιακά πλήρης, είναι χρήσιμη για την κατασκευή χρήσιμων συναρτήσεων όπως λ.χ. την παραγωγή περιττής ισοτιμίας. Μέχρι στιγμής έχουμε χρησιμοποιήσει ολοκληρωμένα τεχνολογίας SSI (Small-Scale Integration) στο εργαστήριο για να υλοποιήσουμε τους σχεδιασμούς μας.

Σε αυτό το Εργαστήριο, θα χρησιμοποιήσουμε δύο ολοκληρωμένα τεχνολογίας MSI (Medium-Scale Integration), έναν **ΠΟΛΥΠΛΕΚΤΗ** (Multiplexer - MUX) και ένα **ΑΠΟΚΩΔΙΚΟΠΟΙΗΤΗ** (Decoder). Το κάθε ένα από αυτά θα χρησιμοποιηθεί, από μόνο του, **για τον σχεδιασμό ενός πλήρους δυαδικού αθροιστή 1-bit (1-bit binary full adder)**.

Όπως φαίνεται στο πιο κάτω διάγραμμα, ένας πλήρης αθροιστής ενός bit έχει τρεις εισόδους του 1-bit, (προσθετέος X, προσθετέος Y και κρατούμενο εισόδου (carry-in) Cin). Ακόμα, διαθέτει δύο εξόδους (κρατούμενο εξόδου (carry-out) Cout, και άθροισμα Sum). Ένας πλήρης αθροιστής ριπής N-bit (N-bit Ripple Carry Adder), αποτελείται από N διαδοχικούς πλήρεις αθροιστές 1-bit. Αυτή η μέθοδος θα χρησιμοποιηθεί για **την κατασκευή ενός αθροιστή/αφαιρέτη 4-bit** στο μέρος Δ αυτής της άσκησης. Μια πιθανή υλοποίηση ενός πλήρους αθροιστή 1-bit φαίνεται στην εικόνα 5-4 σελ. 205 του βιβλίου σας (Mano & Kime, 2004). Επίσης, η εικόνα 5-5 σελ. 206 δείχνει το ιεραρχικό σχηματικό για έναν πλήρη αθροιστή ριπής 4-bit (4-bit Ripple Carry Adder).



Σε αυτό το Εργαστήριο, θα χρησιμοποιηθεί η εφαρμογή του MAX+PLUS II **Timing Analyzer** για πρώτη φορά. Η εφαρμογή αυτή παρουσιάζει τις καθυστερήσεις στο κύκλωμα οι οποίες υπολογίζονται από τον Compiler σε μια ευανάγνωστη μορφή πίνακα. Το MAX+PLUS II θεωρεί ότι ο σχεδιασμός θα υλοποιηθεί με διατάξεις προγραμματιζόμενης λογικής (PLDs) και όχι με τα ολοκληρωμένα που θα χρησιμοποιήσετε. Έτσι οι καθυστερήσεις δεν θα είναι ακριβώς οι ίδιες με αυτές που θα υπάρχουν στο κύκλωμα σας, παρόλα αυτά είναι χρήσιμο για να κατανοήσετε την έννοια της καθυστέρησης.

Επίσης στην άσκηση αυτή θα χρησιμοποιήσετε δίαυλους (buses) για πρώτη φορά. Η χρήση των διαύλων θα εξηγηθεί στην συνέχεια με λεπτομέρεια. Οι περισσότερες διαδικασίες αυτής της άσκησης έχουν κα-

λυφθεί εκτεταμένα στις προηγούμενες εργαστηριακές ασκήσεις και άρα οι επεξηγήσεις τους δεν επαναλαμβάνονται εδώ. Αν δε θυμάστε πώς πρέπει να γίνουν οι ζητούμενες ενέργειες, συμβουλευτείτε τις προηγούμενες ασκήσεις ή το σύστημα βοήθειας του MAX+PLUS II.

Σημείωση: Τα πιο κάτω αρχεία χρειάζονται για αυτή την άσκηση. Πριν ξεκινήσετε, αντιγράψετε τα αρχεία αυτά από το φάκελο του μαθήματος σε ένα δικό σας φάκελο με το όνομα lab3. Ακολουθήστε τις οδηγίες των υπευθύνων του εργαστηρίου για το που βρίσκονται τα αρχεία αυτά: lab3.scf, lab3_fas4.scf

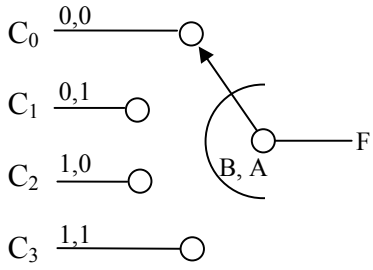
Μέρος Α (πρώτη εβδομάδα)

Εργασία Σημειωματάριου:

1. Καταγράψτε τον πίνακα αληθείας ενός πλήρους αθροιστή 1-bit με τις εξής στήλες: X, Y, C, Cout, Sum. Οι τιμές εισόδου πρέπει να σχηματίζουν μιαν ακολουθία δυαδικής απαρίθμησης (binary counter).
2. Καταγράψτε τις εξόδους Sum και Cout σαν **άθροισμα ελαχιστόρων** (**sum of minterms)**, τόσο σαν εκφράσεις Boolean όσο και στην σύντομη σημειογραφία που έχετε μάθει (Σm). Σημειώστε ότι το bit C είναι το λιγότερο σημαντικό bit (LSB) και το X το περισσότερο σημαντικό bit (MSB).

I. Υλοποίηση Πλήρους Αθροιστή ενός Bit με Πολυπλέκτες.

Ένας πολυπλέκτης (multiplexer - MUX) είναι το ισοδύναμο, λειτουργικά, ολοκληρωμένο ενός διακόπτη επιλογής, του οποίου η μία έξοδος μπορεί να ενωθεί με οποιαδήποτε από τις **N** διαφορετικές εισόδους. Η «θέση» του διακόπτη καθορίζεται από ένα δυαδικό αριθμό ο οποίος είναι επίσης είσοδος στο κύκλωμα ή από μία διεύθυνση που αποτελείτε από $\log_2$ (**N**) bits.Παραδείγματα τέτοιων ολοκληρωμένων τεχνολογίας Low-Power Schottky (LS) είναι οι διατάξεις: 74LS157 (τέσσερις 2-σε-1 MUX), 74LS153 (δύο 4-σε-1 MUX) και 74LS151 (ένας 8-σε-1 MUX) πολυπλέκτες. Μόνο οι 74LS157 και 74LS153 υπάρχουν στον κουτί σας και μόνο ο δεύτερος θα χρησιμοποιηθεί στην άσκηση αυτή.
Το συναρτησιακό ισοδύναμο ενός 4-σε-1 πολυπλέκτη δίνετε πιο κάτω. Τα δύο σήματα ελέγχου B και A καθορίζουν την θέση του διακόπτη, έτσι που η έξοδος F να είναι ενωμένη σε ακριβώς μία από τις εισόδους C_0 έως C_3. Η πράξη που εκτελεί ο διακόπτης καθορίζεται από την δυαδική συνάρτηση που φαίνεται στην εξίσωση [1].

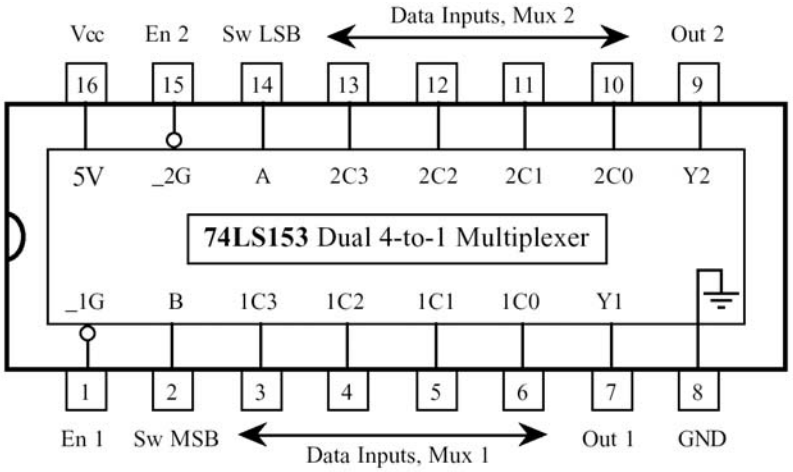


$$F = (\overline{B} \cdot \overline{A}) \cdot C_0 + (\overline{B} \cdot A) \cdot C_1 + (B \cdot \overline{A}) \cdot C_2 + (B \cdot A) \cdot C_3 \quad [1]$$

Παρόλο που ο πολυπλέκτης χρησιμοποιείται συχνά σαν διακόπτης μπορεί να χρησιμοποιηθεί και για την παραγωγή συνδυαστικής λογικής. Ένας πολυπλέκτης με M σήματα ελέγχου (διευθύνσεων) μπορεί να υλοποιήσει οποιαδήποτε συνάρτηση με τις M αυτές μεταβλητές αφού υπάρχει μία ακριβώς είσοδος που αντιστοιχεί σε κάθε ελαχιστόρο. Θέτοντας κατάλληλη τιμή σε κάθε μία από τις εισόδους (αντίστοιχη μεταβλητή, 1 ή 0) παίρνουμε τους ελαχιστόρος που επιθυμούμε σύμφωνα με τον πίνακα αληθείας της συνάρτησης. Ένας πολυπλέκτης και ένας αντιστροφέας μπορούν να υλοποιήσουν κάθε συνάρτηση M+1 μεταβλητών, όπου το συμπλήρωμα της επιπλέον μεταβλητής, το 0 ή το 1 συνδέονται σε κάθε γραμμή εισόδου όπως καθορίζεται από τον πίνακα αληθείας της συνάρτησης. Για παράδειγμα θεωρείστε ότι έχουμε μία συνάρτηση τριών μεταβλητών F(X, Y, W), και θέλουμε να την υλοποιήσουμε με ένα πολυπλέκτη 4-σε-1. Αν χρησιμοποιήσουμε τις μεταβλητές X και Y ως τα δύο σήματα ελέγχου του πολυπλέκτη, τότε το W γίνεται η «επιπλέον» μεταβλητή του πολυπλέκτη. Από τον πίνακα αληθείας μπορείτε να βρείτε ότι οι τιμές των εισόδων του πολυπλέκτη πρέπει να τεθούν στο (W, W', 0, or 1). Εναλλακτικά, οι τέσσερις είσοδοι του πολυπλέκτη C0 έως C3 μπορούν να υπολογιστούν από την εξίσωση [1] αντικαθιστώντας συγκεκριμένες τιμές του X και του Y στην συνάρτηση [2] πιο κάτω. Οι υπολογισμοί που θα κάνετε θα σας οδηγήσουν στις τιμές (0, 1, W ή W') που πρέπει να ανατεθούν στις εισόδους του πολυπλέκτη. Το F στην [2] αναφέρεται στη συνάρτηση που θέλουμε να υλοποιήσουμε με τον πολυπλέκτη.

$$C_0 = F(0, 0, Z) \qquad C_1 = F(0, 1, Z) \qquad C_2 = F(1, 0, Z) \qquad C_3 = F(1, 1, Z) \qquad [2]$$

Ο πίνακας αληθείας και το διάγραμμα σύνδεσης για το 74LS153 φαίνεται πιο κάτω. Σημειώστε ότι το B είναι το MSB της διεύθυνσης 2-bit και το σήμα επιλογής **_G** είναι αρνητικής λογικής[1].



Switch B	Switch A	En _G	Output Y
X	X	1	0
0	0	0	C0
0	1	0	C1
1	0	0	C2
1	1	0	C3

Εργασία Σημειωματάριου:

1. Σχεδιάστε ένα κύκλωμα που να υλοποιεί ένα πλήρη αθροιστή με την χρήση ενός ολοκληρωμένου με δύο πολυπλέκτες 4-σε-1 και ένα αντιστροφέα. Χρησιμοποιήστε τις εξισώσεις που έχετε στο σημειωματάριο (άσκηση 2, σελ. 3) με τους υπολογισμούς των εξισώσεων [2] πιο πάνω. Επαληθεύσετε τα αποτελέσματα σας παρατηρώντας τους πίνακες αληθείας.

II. Υλοποίηση ενός Πλήρους Αθροιστή ενός Bit με Αποκωδικοποιητή.

Ένας αποκωδικοποιητής (decoder) είναι ένα ολοκληρωμένο MSI συνδυαστικής λογικής με **N** εισόδους και με 2^N εξόδους, που αντιστοιχούν στους ελαχιστόρους των εισόδων, έτσι που ακριβώς μια από τις εξόδους ενεργοποιείται για κάθε συνδυασμό τιμών στις εισόδους. Κάθε συνάρτηση **N** μεταβλητών μπορεί να υλοποιηθεί παίρνοντας το λογικό OR των ελαχιστόρων που περιγράφουν την συνάρτηση, από τις εξόδους του αποκωδικοποιητή. Παραδείγματα αποκωδικοποιητών τεχνολογίας Low-Power Schottky είναι τα: 74LS155A (δύο 2-σε-4 Decoders), 74LS138 (ένας 3-σε-8 Decoder) και το 74LS42 (ένας 4-σε-10 BCD-σε-δεκαδικό) αποκωδικοποιητές. Αφού ο πλήρης αθροιστής έχει 3 εισόδους θα χρησιμοποιήσουμε το ολοκληρωμένο 74LS138 3-σε-8 αποκωδικοποιητή για αυτό το πείραμα. Ο πίνακας αληθείας και το διάγραμμα των pins του ολοκληρωμένου ακολουθούν στην επόμενης σελίδα. Σημειώστε ότι οι έξοδοι είναι αρνητικής λογικής (active low) , όπως δηλώνουν οι κύκλοι μπροστά από τα ονόματα τους. Επίσης, παρατηρήστε ότι οι είσοδοι ενεργοποίησης (Enable Inputs) είναι θετικής και αρνητικής λογικής. Υπάρχουν τρεις είσοδοι ενεργοποίησης (**G1, _G2A** και **_G2B).** Ο αποκωδικοποιητής ενεργοποιείτε μόνο όταν το **G1** είναι λογικό High (1) και τα δύο **_G2A** and **_G2B** είναι λογικό Low (0).

Αν το **G1** χρησιμοποιηθεί σαν σήμα εισόδου, τότε το ολοκληρωμένο ονομάζετε αποπλέκτης (demultiplexer), όπου το συμπλήρωμα του **G1** οδηγείτε στην συγκεκριμένη έξοδο Y που επιλέγεται με την «διεύθυνση» των εισόδων 3-bit (C, B, and A). Ο αντίστοιχος του διακόπτης θα ήταν ένας διακόπτης 4 θέσεων όπως αυτός στην σελίδα 3, με διάδοση του σήματος από δεξιά προς τα αριστερά.

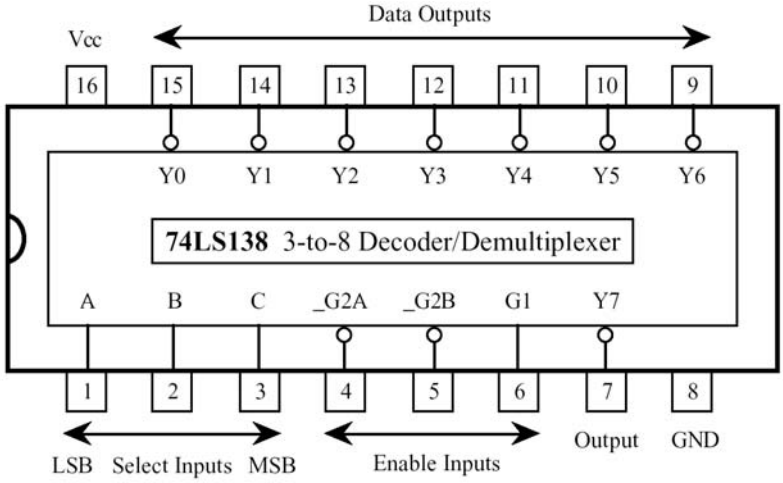
[1] Το πρόθεμα "_" μπροστά από ένα σήμα υπονοεί ότι το σήμα είναι αρνητικής λογικής (ενεργοποιείται με low).

Εργασία Σημειωματάριου:

1. Παρατηρώντας τον πίνακα αληθείας του Πλήρους Αθροιστή (άσκηση 1, σελ. 3), καθορίστε τις κατάλληλες συνδέσεις για την υλοποίηση του με ένα 3-σε-8 αποκωδικοποιητή. Θεωρήστε θετική λογική για τις εξόδους τους αποκωδικοποιητή, και βρείτε ένα κύκλωμα με εξόδους Sum και Cout, χρησιμοποιώντας τον απαραίτητο αριθμό πυλών OR.
2. Αφού οι έξοδοι του 74LS138 είναι αρνητική λογικής, δείξετε ότι μετατρέποντας κατάλληλα το κύκλωμα πιο πάνω (σημείο 1), μπορείτε να καταλήξετε σε ένα κύκλωμα με έναν αποκωδικοποιητή και δύο πύλες NAND, που υλοποιούν έναν πλήρη αθροιστή. Το 74LS20 περιέχει δύο πύλες NAND 4 εισόδων. Έτσι, ο σχεδιασμός αυτός απαιτεί ένα μόνο 74LS138 και ένα μόνο 74LS20.

Enable Inputs		Select Inputs			Outputs							
G1	_G2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	1	X	X	X	1	1	1	1	1	1	1	1
0	X	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1	1	1
1	0	1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	1	0	1	1
1	0	1	1	0	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1	1	1	0

_G2* = _G2A + _G2B



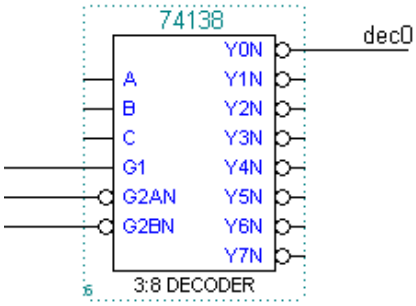
III. Υλοποίηση ενός Πλήρους Αθροιστή 1-bit με το MAX+PLUS II

Κατασκευή σχηματικού με Graphic Editor

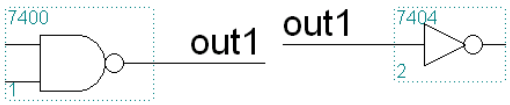
Κατασκευάστε ένα σχηματικό διάγραμμα που να περιέχει τόσο την υλοποίηση σας με τον πολυπλέκτη όσο και την υλοποίηση σας με τον αποκωδικοποιητή. Τρεις θύρες εισόδου δίνουν τα X, Y και Cin και για τους δύο σχεδιασμούς. Ο κάθε σχεδιασμό πρέπει να έχει, προφανώς, τις δικές του θύρες εξόδου.

1. Κατασκευάστε έναν κατάλογο με όνομα lab3 κάτω από το χώρο του μαθήματος.
2. Ανοίξετε το MAX+PLUS II και δημιουργήστε ένα νέο αρχείο του Graphic Editor. Αποθηκεύστε το με το όνομα lab3.gdf, και καθορίστε το ως το τρέχον project.
3. Προσθέστε τα ακόλουθα σύμβολα από το παράθυρο διαλόγου Enter Symbol : gnd, input, output, vcc, 7404 (inverter), 74138 (3-to-8 Decoder), 74153 (dual 4-to-1 Multiplexer), και 7420 (Dual 4-input NAND).
4. Αντιγράψετε τα στοιχεία που χρειάζεται σύμφωνα με τον σχεδιασμό σας.

   Αν κατά τη διάρκεια του σχεδιασμού αντιληφθείτε ότι το αρχείο του Graphic Editor δεν είναι αρκετά μεγάλο για τον σχεδιασμό σας, μπορείτε να το μεγεθύνετε με την επιλογή Size (File Menu).
5. Συνδέστε όλα τα σήματα επιλογής (enable) των πολυπλεκτών και του αποκωδικοποιητή είτε στην γείωση είτε στη πηγή (Vcc), με κατάλληλο τρόπο, ώστε τα στοιχεία να είναι πάντα επιλεγμένα (enable).
6. Μετονομάστε τις εισόδους σε X, Y και Cin.
7. Σχεδιάστε ένα μικρό κόμβο («καλώδιο») από την έξοδο Y0N του αποκωδικοποιητή και ονομάστε τον ως dec0 κάνοντας αριστερό κλικ πάνω του.



Η ονομασία κόμβων είναι πολύ πρακτική διαδικασία. Δύο κόμβοι με το ίδιο όνομα θεωρούνται συνδεδεμένοι στο MAX+PLUS II, ακόμη κι αν δεν υπάρχει καλώδιο μεταξύ τους. Ως παράδειγμα θεωρήστε το πιο κάτω σχήμα.



Σε περιπτώσεις που υπάρχουν πολλοί κόμβοι σε μικρή περιοχή, η σύνδεση με καλώδια μπορεί να δημιουργήσει συγχύσεις, τουλάχιστον οπτικές. Με την κατάλληλη ονομασία των κόμβων ξεπερνούμε την δυσκολία αυτή.

8. Ονομάστε τις υπόλοιπες εξόδους του αποκωδικοποιητή με την σειρά , dec1 έως dec7.
9. Ονομάστε τις εισόδους των πυλών NAND ώστε να αντιστοιχούν με τις εξόδους του αποκωδικοποιητή και να ακολουθούν το σχεδιασμό που έχετε κάνει.
10. Συνδέστε τα υπόλοιπα στοιχεία σύμφωνα με τους σχεδιασμούς σας στα μέρη I και II.
11. Ονομάστε τις θύρες εξόδου με τα ονόματα: Sum_Mux, Cout_Mux, Sum_Dcd και Cout_Dcd για τους πολυπλέκτες και τον αποκωδικοποιητή αντίστοιχα.
12. Αποθηκεύσετε και ελέγξτε το αρχείο για λάθη.
13. Κρατήστε μία εκτύπωση του σχηματικού σας.

14. Κλείστε τον Graphic Editor.

Προσομοίωση

15. Ανοίξτε τον compiler.
16. Επιλέξτε το **Timing SNF Extractor** (Processing menu).

    Ο compiler έχει οριστεί για να κατασκευάσει τα απαραίτητα αρχεία που περιέχουν χρονικές και λογικές πληροφορίες για χρονική ανάλυση και προσομοίωση του κυκλώματος.

17. Μεταφράστε το σχεδιασμό σας.

    Σημειώστε ότι η διαδικασία της μετάφρασης έχει μεγαλώσει σε διάρκεια (σε σχέση με αυτές που έχουμε δει σε προηγούμενα Εργαστήρια) αφού επιπλέον βήματα έχουν προστεθεί στην διαδικασία.

18. Με την χρήση του windows explorer, αντιγράψετε το αρχείο κυματομορφών, lab3.scf στο κατάλογο που περιέχει τον σχεδιασμό σας (περισσότερες πληροφορίες για το που βρίσκετε το αρχεία από τους υπευθύνους του εργαστηρίου).

    Σε αυτή την άσκηση, οι κυματομορφές εισόδου έχουν γίνει από πριν. **Οι κυματομορφές θα δουλέψουν σωστά μόνο αν τα ονόματα που έχετε δώσει συμφωνούν με αυτά που δόθηκαν στα βήματα 6 και 11 του (α).**

19. Ανοίξετε τον Simulator. Επιβεβαιώστε ότι το αρχείο lab3.scf χρησιμοποιείται σαν είσοδος της προσομοίωσης σας.
20. Εκτελέστε την προσομοίωση και ανοίξετε το αρχείο SCF.

    Παρατηρήστε ότι στις εξόδους υπάρχουν καθυστερήσεις.

21. Κρατήστε μια εκτύπωση των κυματομορφών.

    Για να πάρετε καλύτερη εκτύπωση επιλέξτε όπως η σελίδα σας τυπωθεί Landscape στο παράθυρο διαλόγου Print Setup (File menu).

Χρονική Ανάλυση (Timing Analysis)

Η χρονική συμπεριφορά του κυκλώματος θα εξεταστεί μέσω της εφαρμογής του Timing Analyzer η οποία καταγράφει τις καθυστερήσεις του κυκλώματος σε μια ευανάγνωστη μορφή. *Σημείωση:* ο Timing Analyzer θα τρέξει μόνο αν έχει προηγηθεί επιτυχής μετάφραση του σχεδιασμού με την επιλογή για Timing SNF Extractor.

22. Ανοίξτε τον **Timing Analyzer** (MAX+plus II menu).
23. Επιλέξτε **Delay Matrix** (Analysis Menu).

    Η επιλογή αυτή καθορίζει τον Timing Analyzer ώστε να παρουσιάζει τα αποτελέσματα της ανάλυσης της καθυστέρησης διάδοσης (propagation delay).

24. Επιλέξτε Start στο παράθυρο του Timing Analyzer.

Ο πίνακας θα γεμίσει με τιμές που αναπαριστούν τις καθυστερήσεις που υπάρχουν στο κύκλωμα.

25. Καταγράψτε τις τιμές των καθυστερήσεων στο σημειωματάριο σας.

    Αν για ένα ζεύγος εισόδου/εξόδου υπάρχουν δύο τιμές καθυστέρησης οφείλετε στο ότι η καθυστέρηση διαφέρει για διαφορετικές τιμές εισόδου. Σε αυτή την περίπτωση καταγράψτε μόνο την μέγιστη καθυστέρηση Αν δεν υπάρχει τιμή για δύο κόμβους, τότε οι κόμβοι αυτοί δεν συνδέονται.

26. Κλείστε όλα τα παράθυρα αλλά μην βγείτε από το MAX+PLUS II.

IV. Σχεδιασμός ενός Αθροιστή/Αφαιρέτη τεσσάρων bit

Σε αυτό το σημείο θα κατασκευάσετε ένα **μη-προσημασμένο** (unsigned) αθροιστή/αφαιρέτη με την χρήση ενός πλήρους αθροιστή 1-bit και μιας πύλης XOR 2-εισόδων. Ο αθροιστής/αφαιρέτης έχει και μια τέταρτη είσοδο, την SUB. Όταν το SUB είναι στο λογικό 1, το κύκλωμα λειτουργεί σαν αφαιρέτης, ενώ όταν το SUB είναι στο λογικό 0, λειτουργεί σαν αθροιστής. Στόχος αυτής της άσκησης είναι να κατασκευάσετε ένα αθροιστή/αφαιρέτη 4-bit με την παράθεση τεσσάρων προσημασμένων αθροιστών/αφαιρετών 1-bit όπως φαίνεται στην σελίδα 11. Στα επόμενα βήματα θα μετατρέψετε το σχηματικό lab3.gdf για να φτιάξετε ένα προσημασμένο αθροιστή/αφαιρέτη 1-bit με την προσθήκη μιας θύρας εισόδου και μίας πύλης 74LS86A XOR στο κύκλωμα του πολυπλέκτη 74LS153. Το σχηματικό θα αποθηκευτεί σαν lab3_fas.gdf, και θα δημιουργηθεί ένα σύμβολο με το ίδιο όνομα. Το σύμβολο θα αναπαραχθεί 4 φορές σε ένα σχηματικό με το όνομα lab3_fas4.gdf και θα γίνουν οι απαραίτητες συνδέσεις για να δημιουργηθεί ο αθροιστής/αφαιρέτης των 4 bit. Ο σχεδιασμός μετά θα μεταφραστεί και θα προσομοιωθεί.

a) Δημιουργία σχηματικού Πλήρους Αθροιστή/Αφαιρέτη 1 bit.

1. Ανοίξετε το σχηματικό lab3.gdf στο Graphic Editor.
2. Αποθηκεύσετε το σαν lab3_fas.gdf και θέστε το current project στο αρχείο αυτό.
3. Επιλέξτε και διαγράψτε όλα τα στοιχεία που έχουν σχέση με την υλοποίηση του αθροιστή με το αποκωδικοποιητή, δηλαδή τα 74138, 7420, κλπ.
4. Εισάγετε μια θύρα εισόδου και ονομάστε την SUB. Μετονομάστε τις υπόλοιπες δύο εξόδους σε Sum και Cout.
5. Προσθέστε μια πύλη 7486 XOR (74LS86A στο κουτί με τα ολοκληρωμένα) για να χρησιμοποιηθεί με την είσοδο SUB. Συνδέστε την XOR για αντιστροφή του Y όταν SUB= 1.

   Με τον τρόπο αυτό υλοποιούμε το συμπλήρωμα ως προς ένα (ones complement του Y). Το συμπλήρωμα ως προς δύο (twos complement του Y) θα υλοποιηθεί χρησιμοποιώντας το SUB σαν είσοδο στο Cin(0) στο ψηλότερο επίπεδο του σχηματικού 4-bit, lab3_fas4.gdf.

6. Αποθηκεύστε και ελέγξτε το σχηματικό.

7. Δημιουργήστε το σύμβολο επιλέγοντας **Create a Default Symbol** (File menu).

   Παρόλο που δεν υπάρχει κάποια οπτική διαφοροποίηση ή μήνυμα το σύμβολο έχει δημιουργηθεί και αποθηκευτεί.

Symbol Editor

8. Ανοίξετε το αρχείο lab3_fas.sym που περιέχει το σύμβολο (περισσότερες πληροφορίες για το που βρίσκετε το αρχεία από τους υπευθύνους του εργαστηρίου).

   Το σύμβολο θα ανοίξει στην εφαρμογή Symbol Editor. Από εδώ μπορείτε να αλλάξετε την εμφάνιση του συμβόλου σας, περιλαμβανομένων του σχήματος και του μεγέθους του καθώς και την θέση των εισόδων και εξόδων.

9. Αλλάξτε το σύμβολο ώστε να συμφωνεί με το lab3_fas.gdf του σχηματικού της σελίδας 11.

10. Αποθηκεύστε το σύμβολο και κλείστε το αρχείο.

Δημιουργία Σχηματικού Πλήρους Αθροιστή/Αφαιρέτη 4-bit.

11. Δημιουργήστε ένα αρχείο σχεδιασμού και αποθηκεύστε το σαν lab3_fas4.gdf. Καθορίστε το σαν τρέχον project.

12. Εισάγετε 4 αντίγραφα (στιγμιότυπα) του συμβόλου lab3_fas από το παράθυρο διαλόγου Insert Symbol καθώς και τρεις θύρες εισόδου, δύο θύρες εξόδου και ένα σύμβολο γείωσης και τοποθετήστε τα όπως το σχήμα της σελίδας 11.

13. Μετονομάστε τις θύρες όπως το σχήμα στην σελίδα 11.

    Οι θύρες εισόδου X[3..0], Y[3..0] και Z[3..0] είναι σε μορφή δίαυλου (bus). Ένα pin διαύλου χρησιμοποιείται για την δημιουργία μιας ομάδας από pins. Όταν συνδέουμε ένα δίαυλο με ένα pin χρησιμοποιούμε την ονομασία NAME[n..0], όπου n είναι ο αριθμός των bits μείων ένα. Έτσι, μία θύρα εισόδου με όνομα A[3..0] είναι ακριβώς το ίδιο με το να είχαμε 4 διαφορετικές εισόδους με ονόματα A3, A2, A1 και A0.

14. Σχεδιάστε μικρές γραμμές από τα pin των διαύλων που να μην είναι ενωμένα πουθενά. Επιλέξτε αυτές γραμμές και από το μενού RMB επιλέξτε line style. Επιλέξτε την πιο χοντρή συνεχόμενη γραμμή η οποία είναι δεύτερη στην λίστα επιλογών.

    Αυτό θα δημιουργήσει γραμμές διαύλου εισόδου και εξόδου. Οι γραμμές διαύλου δεν είναι απαραίτητες, αλλά κάνουν τον σχεδιασμό μας πιο ευανάγνωστο, δηλαδή καταλαβαίνει κάποιος πιο εύκολα ότι πρόκειται για δίαυλο. Παρόλα αυτά, αν μια λεπτή γραμμή (αντί χο-

ντρής) χρησιμοποιηθεί για δίαυλος, ο compiler θα επιστρέψει λάθος.

15. Ονομάστε τις εισόδου και εξόδους του συμβόλου lab3_fas σύμφωνα με το σχήμα της σελίδας 11 με τον τρόπο που το κάνατε στο μέρος ΙΙ για το αποκωδικοποιητή.

    Για να χρησιμοποιήσετε τα δεδομένα μιας γραμμής διαύλου, ενώστε τη γραμμή διαύλου σε οποιαδήποτε από τις εισόδους του κάθε αθροιστή, ονομάζοντας κατάλληλα τη γραμμή/είσοδο βάση του bit που αντιπροσωπεύει. Για το δίαυλο Α που συζητήθηκε πιο πάνω, η σωστή ονομασία είναι A3, A2, A1 και A0 για την κάθε είσοδο, διαφορετικά ο compiler θα μας δώσει λάθος.

    Να θυμάστε:

    - Ένα διάνυσμα κόμβων (πολλαπλοί κόμβοι) πρέπει να ενωθεί με μια αντίστοιχη γραμμή διαύλου για σωστή μετάφραση.
    - Ένας κόμβος εισόδου με όνομα A[3..0] είναι το ίδιο με 4 ξεχωριστούς κόμβους εισόδου με όνομα A3, A2, A1 και A0.
    - Ο κόμβος εισόδου δεν χρειάζεται να είναι ενωμένος γραφικά με τα Pins ενός συμβόλου για να θεωρείτε

16. Συνδέστε τα υπόλοιπα στοιχεία για να ολοκληρωθεί το σχηματικό.
17. Αποθηκεύσετε και ελέγξτε το αρχείο για λάθη. Κρατήστε μια εκτύπωση του.

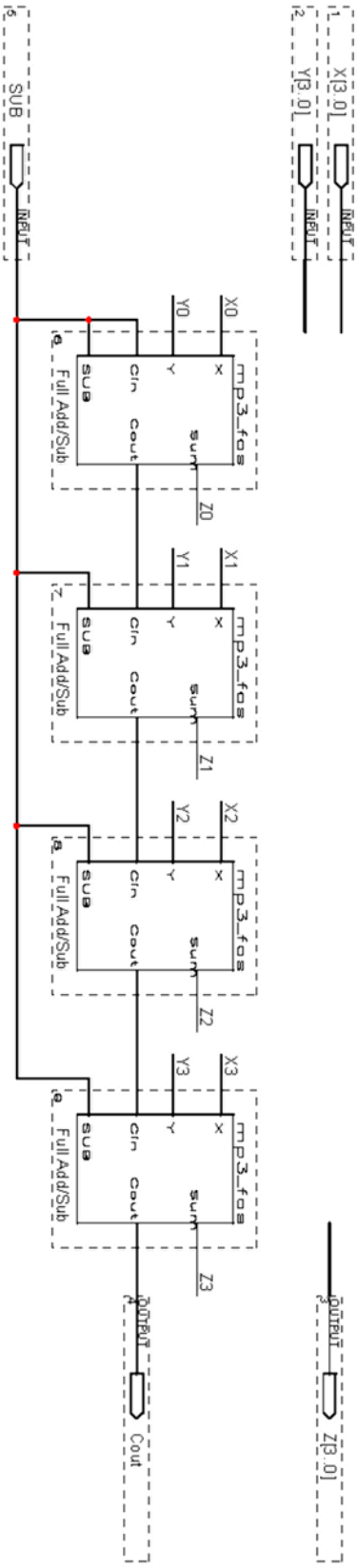
Προσομοίωση

18. Μεταφράστε το project για έλεγχο **μόνο της λογικής συμπεριφοράς** του κυκλώματος.
19. Αντιγράψετε το αρχείο κυματομορφών, lab3_fas4.scf στον κατάλογο που έχετε αποθηκεύσει τον σχεδιασμό σας.
20. Προσομοιώστε τον αθροιστή 4-bit για 800ns με το πιο πάνω αρχείο ως είσοδο.
21. Τυπώστε τα αποτελέσματα της προσομοίωσης.
22. Βγείτε από το MAX+PLUS II χωρίς να αποθηκεύσετε οτιδήποτε.

Αναφορά Μέρους Α

Η αναφορά σας πρέπει να περιλαμβάνει τον Πίνακα Αληθείας και τις εκφράσεις των ελαχιστόρων για τις ασκήσεις 1 και 2 στην σελίδα 2, την παραγωγή του πολυπλέκτη από το Ι.1, την παραγωγή του αποκωδικοποιητή από τα ΙΙ.1 και ΙΙ.2, τα σχηματικά από το ΙΙΙ.a.13 και IV.c.7, τα αποτελέσματα της προσομοίωσης από το ΙΙΙ.b.7 και IV.d.4, και μια σύντομη περιγραφή και σχολιασμό της διαδικασίας και των αποτελεσμάτων.

Ύλη σχετική για αυτό το Εργαστήριο (αποκωδικοποιητές, πολυπλέκτες, αθροιστές) υπάρχει στο βιβλίο σας (Mano & Kime, 2004) στις σελίδες 147-152, 156-161, και 202-206.


SUB
INPUT
X[3..0]
Y[3..0]
INPUT
INPUT
mp3_fas
Full Add/Sub
sub
cin
cout
sum
x
y
X0
Y0
Z0
X1
Y1
Z1
X2
Y2
Z2
X3
Y3
Z3
OUTPUT
Cout
OUTPUT
Z[3..0]