

Άσκηση 2

Συνοδευτικά αρχεία : 2_1.v, 2_2.v

Μέρος 1 : Εισαγωγή

Στο παρακάτω σχήμα δίνεται το σχηματικό διάγραμμα της λογικής πύλης AND (πύλη ΚΑΙ) με 2 εισόδους (γραμμές A και B) και μία έξοδο (γραμμή C). Η πύλη AND δίνει στην έξοδο της την τιμή 1 μόνο όταν όλα τα σήματα εισόδου έχουν την τιμή 1, αλλιώς η έξοδος παίρνει την τιμή 0. Η λογική συνάρτηση AND συμβολίζεται με μία τελεία ή και χωρίς (όπως και ο πολλαπλασιασμός στην Άλγεβρα των πραγματικών αριθμών), δηλαδή με απλή παράθεση των σημάτων εισόδου. Έτσι για παράδειγμα για τη πύλη του σχήματος μπορούμε να γράψουμε $C=AB$ ή $C=A \bullet B$.



Ο πίνακας αλήθειας της λογικής πύλης AND, όπως έχει αναλυθεί στο αμφιθέατρο είναι :

A	B	C=AB
0	0	0
0	1	0
1	0	0
1	1	1

Στη συνέχεια παρατίθεται ο κώδικας σε γλώσσα Verilog για τη περιγραφή αυτού του κυκλώματος καθώς και ο κώδικας για την εξομοίωσή του (οι κώδικες χωρίς σχόλια υπάρχουν και στο συνοδευτικό αρχείο 2_1.v).

```
module myand(A, B, C);           // Ορίζουμε ένα νέο σχεδιασμό με όνομα myand. Αυτός συνδέεται με τον υπόλοιπο κόσμο με
                                // 3 ακροδέκτες A, B και C
    input A, B;                  // Οι ακροδέκτες A και B θα είναι είσοδοι προς το σχεδιασμό μας
    output C;                    // ενώ ο C θα είναι η έξοδος του
    and i0 (C, A, B);           // Στο σημείο αυτό χρησιμοποιούμε τη πρωταρχική συνάρτηση AND της Verilog. Η
                                // συνάρτηση αυτή υπάρχει για να περιγράφει πύλες AND. Η δήλωσή της έχει τη μορφή
                                // AND (έξοδος, είσοδος1, είσοδος2, ..., είσοδοςn). Στην ουσία πρόκειται για μια γενικευμένη
                                // συνάρτηση που περιγράφει AND πύλες όσων εισόδων θέλουμε. Παρατηρήστε ότι στη
                                // πρωταρχική συνάρτηση η έξοδος είναι το πρώτο όρισμα. Εξ ου λοιπόν και η κλήση με
                                // παραμέτρους (C, A, B) που υποδεικνύει ότι στο σχεδιασμό μας το C είναι η έξοδος και τα A
                                // και B είσοδοι.

endmodule

module myand_test();            // Ο σχεδιασμός αυτός δεν έχει κανένα ακροδέκτη. Θα υπάρχουν μόνο τοπικές μεταβλητές.
                                // Κάποιες από αυτές θα τις οδηγούμε εμείς σε επιθυμητές δυαδικές τιμές, ενώ κάποιες άλλες
                                // θα οδηγούνται από τον υποσχεδιασμό μας. Όσες οδηγούμε εμείς πρέπει να δηλωθούν με
                                // τύπο reg. Όλες οι υπόλοιπες θα είναι τύπου wire.



    reg X, Y;
    wire Z;

    initial                      // Στον κώδικα που ακολουθεί δίνουμε δυαδικές τιμές στις τοπικές μεταβλητές μας.
    begin                        // Κάθε νέα τιμή ανατίθεται 100 χρονικές στιγμές μετά τη προηγούμενη.
        #100 X = 0; Y = 0;
        #100 X = 0; Y = 1;
        #100 X = 1; Y = 1;
        #100;
        #100 $stop;             // Στο σημείο αυτό πρέπει εσείς να βάλετε τον εναπομείναντα συνδυασμό.
                                // Παύση της εξομοίωσης μετά από 100 χρονικές στιγμές
    end

    myand inst1 (X, Y, Z);       // Στο σημείο αυτό καλούμε ένα αντίγραφο της συνάρτησης myand που φτιάξαμε
                                // προηγούμενα. Προσέξτε ότι εφόσον έχουμε αναθέσει τιμές στις τοπικές μεταβλητές μας
                                // τύπου reg και αυτές τις χρησιμοποιούμε ως ορίσματα για τις εισόδους της myand στην
                                // ουσία είναι σα να βάζουμε τα σήματα εισόδου της myand στις επιθυμητές τιμές. Η τιμή
                                // δηλαδή X=0 αντιστοιχεί με το βάλω 0 στον ακροδέκτη A της myand. Τέλος παρατηρήστε
                                // τη σειρά αναγραφής των ορισμάτων. Δεν είναι τυχαία, αλλά ακολουθεί τη δήλωση της
                                // myand. Δηλαδή η τοπική μεταβλητή X θα συνδεθεί στο A, η Y στο B και η Z στο C.

endmodule
```

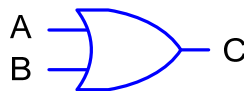
Στη συνέχεια περιγράφεται για μια ακόμη φορά η διαδικασία εξομοίωσης του παραπάνω κυκλώματος.

- Δημιουργήστε το φάκελλο C:\TEST. Μεταφέρετε εκεί το αρχείο 2_1.v
- Ξεκινείτε τον εξομοιωτή.
- Από το μενού File → Change Directory οδηγηθείτε στο C:\TEST και πατήστε open. Οτι αρχείο δημιουργηθεί στο εξής για αυτό το σχεδιασμό θα βρίσκεται εντός του φακέλλου C:\TEST
- Δημιουργήστε μια νέα βιβλιοθήκη μέσω του File → New → Library. Αφήστε το προεπιλεγμένο όνομα work και την επιλογή Create: a new library and a logical mapping to it, και κάντε κλικ στο OK. Τώρα έχει δημιουργηθεί μέσα στο directory σας η βιβλιοθήκη work. Παρατηρήστε ότι τώρα έχει εμφανιστεί στη λίστα του Workspace η βιβλιοθήκη work.
- Στη συνέχεια ακολουθεί η διαδικασία μετάφρασης. Επιλέξτε Compile → Compile ή κάντε κλικ στο εικονίδιο  που βρίσκεται κάτω από το μενού Compile. Όποιο τρόπο και να επιλέξετε θα ανοίξει το παράθυρο διαλόγου compile HDL Source Files. Στο παράθυρο αυτό παρατηρήστε ότι στο πλαίσιο Library υπάρχει το work και στο πλαίσιο Look in υπάρχει ο φάκελος C:\TEST. Επιλέξτε το αρχείο 2_1.v και παρατηρείστε τα μηνύματα στο δεξί μισό του παραθύρου κατά την ώρα της μετάφρασης.
- Μετά το τέλος της μετάφρασης στη βιβλιοθήκη work στο workspace έχουν αποθηκευτεί όλοι οι σχεδιασμοί σας. Πατήστε done, για να κλείσει το παράθυρο της επιλογής αρχείου για μετάφραση.
- Ακολουθεί η διαδικασία της λογικής εξομοίωσης (simulation) του σχεδιασμού myand_test. Μπορείτε να ξεκινήσετε την εξομοίωση με 2 εναλλακτικούς τρόπους :
 - Από τη βιβλιοθήκη work (αφού κάνετε expand ώστε να φαίνονται τα περιεχόμενά της) κάντε διπλό κλικ στο myand_test ή
 - Από το μενού simulate → simulate. Ανοίγει το παράθυρο διαλόγου εξομοίωσης. Εκεί πατήστε στο + αριστερά από τη βιβλιοθήκη work και στη συνέχεια κάντε κλικ στο myand_test. Τέλος κάντε κλικ στο OK.
- Το παράθυρο του εξομοιωτή παίρνει πλέον άλλη μορφή για να υποδείξει ότι είστε εντός διαδικασίας εξομοίωσης. Από το View → Signals αποκτάτε πρόσβαση στα σήματα που υπάρχουν στον επιλεγόμενο σχεδιασμό.
- Μέσω Add → Wave → Signals in Region από το παράθυρο των σημάτων, μπορείτε να προσθέσετε όλα τα σήματα του παραθύρου σε ένα παράθυρο κυματομορφών. (Σημείωση: Αν επιλέξετε Add → Wave → Signals in Design, στο παράθυρο σημάτων θα βλέπετε όλα τα σήματα του σχεδιασμού που εξομοιώνεται, ανεξαρτήτως επιπέδου, σε αντίθεση με το Signal in Region όπου φαίνονται μόνο τα σήματα του επιλεγέντος σχεδιασμού).
- Για να εκτελέσουμε εξομοίωση η οποία θα σταματήσει αυτόματα στο χρονικό δηλαδή σημείο που έχουμε βάλει την εντολή \$stop; Πατάμε το  (Run All).
- Ηρθε η ώρα να παρατηρήσουμε τις κυματομορφές μας. Στην αρχή και οι τρεις γραμμές είναι κόκκινες και αυτό συμβαίνει γιατί στην αρχή του χρόνου και μέχρι τη χρονική στιγμή 100 δεν εφαρμόζουμε καμμία γνωστή τιμή στους σχεδιασμούς μας. Ο εξομοιωτής επισημαίνει αυτή την απροσδιοριστία με κόκκινο χρώμα στις κυματομορφές μας.
- Στη συνέχεια η είσοδος A και B (X και Y αντίστοιχα) οδηγούνται στο 0. Έτσι η έξοδος C (μεταβλητή Z) παίρνει την τιμή 0. Οι τιμές αυτές παραμένουν σταθερές για άλλες 100 χρονικές στιγμές της εξομοίωσης. Αντίστοιχα μπορείτε να παρατηρήσετε τι συμβαίνει στις υπόλοιπες χρονικές περιόδους των 100 χρονικών στιγμών.

ΠΑΡΑΔΟΤΕΟ 1 : Ο κώδικας που εξετάσαμε δεν επιβάλλει όλους τους πιθανούς συνδυασμούς στις εισόδους του σχεδιασμού. Εμπλουτίστε το παραπάνω κώδικα, βάζοντας το συνδυασμό που λείπει στο σημείο που σας υποδεικνύεται παραπάνω. Παραδίδετε τον τροποποιημένο κώδικα του αρχείου myand_test.v και αφού εκτελέσετε την εξομοίωση του παραπάνω κώδικα επισυνάψτε ένα screenshot των κυματομορφών εξομοίωσης.

Μέρος 2 : Ζητούμενα

Ζητείται να δώσετε κώδικα για τη περιγραφή του σχεδιασμού του πιο κάτω σχήματος, κώδικα για την εξομοίωσή του και screenshot των κυματομορφών εξομοίωσης. Ο σχεδιασμός είναι μόνο μια πύλη OR (πύλη Ή) με 2 εισόδους (γραμμές A και B) και μία έξοδο (γραμμή C). Η λογική πύλη OR υλοποιεί τη λογική συνάρτηση που δίνει στην έξοδο της την τιμή 0 μόνο όταν όλες οι εισοδοί έχουν τη τιμή 0. Η συνάρτηση OR συμβολίζεται με ένα σταυρό (+) (όπως και η πρόσθεση στην Άλγεβρα των πραγματικών αριθμών). Για το σχήμα μας είναι $C=A+B$.



ΠΑΡΑΔΟΤΕΟ 2 : Αφού εκτελέσετε την εξομοίωση του παραπάνω κώδικα δώστε ένα screenshot των κυματομορφών εξομοίωσης στο οποίο να φαίνονται οι γραφικές παραστάσεις όλων των σημάτων.

Μέρος 3 : Ενδεικτική Λύση

Συμβουλευτείτε το συνοδευτικό αρχείο 2_2.v