

Άσκηση 3

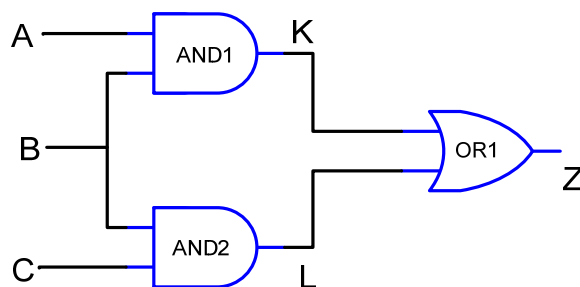
Συνοδευτικά αρχεία : 3_1.v, 3_2.v

Μέρος 1 : Εισαγωγή

Σκοπός της άσκησης είναι η εξοικείωσή σας με το δομικό τρόπο περιγραφής κυκλωμάτων στη Verilog. Εχοντας ολοκληρώσει την άσκηση θα πρέπει να μπορείτε να περιγράφετε ένα συνδυαστικό κύκλωμα χρησιμοποιώντας τις πρωταρχικές συναρτήσεις των λογικών πυλών που μας παρέχει η Verilog.

Ας ξεκινήσουμε από το απλό κύκλωμα του σχήματος 3.1. Το κύκλωμα αποτελείται από 2 πύλες AND (ΚΑΙ) και 1 πύλη OR (Η), κάθε μία των 2 εισόδων. Το κύκλωμά μας έχει 3 σήματα εισόδου (A, B, C) και ένα εξόδο (Z). Από το κύκλωμα βλέπουμε ότι:

- Η πύλη AND1 οδηγείται από τα σήματα A και B και οδηγεί το σήμα εξόδου K.
- Η πύλη AND2 οδηγείται από τα σήματα εισόδου B και C και οδηγεί το σήμα εξόδου L.
- Η πύλη OR1 οδηγείται από τα σήματα K και L και παράγει την έξοδο Z.



Σχήμα 3. 1

Αφού το κύκλωμά μας έχει 3 κύριες εισόδους μπορούμε να συμπεράνουμε ότι στην είσοδο μπορούν να εφαρμοστούν $2^3=8$ διαφορετικοί συνδυασμοί. Οι τιμές των ενδιάμεσων σημάτων καθώς και της εξόδου γι' αυτούς τους συνδυασμούς φαίνονται στον πίνακα 3.1.

Πίνακας 3. 1

Κύριες Είσοδοι			Ενδιάμεσες Γραμμές		Κύρια έξοδος
A	B	C	K	L	Z
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	1	1
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	1	0	1
1	1	1	1	1	1

Σκοπός στη συνέχεια είναι να περιγράψουμε το ίδιο κύκλωμα σε Verilog χρησιμοποιώντας δομικό τρόπο περιγραφής. Η Verilog για το σκοπό αυτό μας παρέχει μια μεγάλη σειρά πρωταρχικών συναρτήσεων. Για παράδειγμα η συνάρτηση and (z, x, y) είναι μια πρωταρχική συνάρτηση περιγραφής μιας πύλης AND με εισόδους τα σήματα x, y και έξοδο το σήμα z. Η συνάρτηση nor (w, x, y, z) είναι μια πρωταρχική συνάρτηση περιγραφής μιας πύλης NOR με εισόδους τα σήματα x, y και z και έξοδο το σήμα w. Όπως μπορείτε να φανταστείτε υπάρχουν πρωταρχικές συναρτήσεις για όλες τις λογικές πύλες οσωνδήποτε εισόδων και αν χρειαστείτε. Σε κάθε τέτοια συνάρτηση, η έξοδος είναι η μεταβλητή που αναγράφεται πρώτη, ακολουθούμενη από τις εισόδους.

Με βάση τα παραπάνω, ας προσπαθήσουμε να περιγράψουμε το κύκλωμα του σχήματος 3.1 σε κώδικα Verilog (ο κώδικας χωρίς σχόλια υπάρχει και στο συνοδευτικό αρχείο 3_1.v).

```
module circuit3_1(A, B, C, Z); // Το κύκλωμά μας θα ονομάζεται circuit3_1 και έχει 4 σημεία
    // διασύνδεσης A, B, C & Z με τον υπόλοιπο κόσμο
    // Προσέξτε ότι η σειρά αναγραφής είναι δική μας υπόθεση καθώς
    // εδώ δεν έχουμε μια πρωταρχική συνάρτηση, αλλά ορίζουμε μια
    // νέα δικιά μας
    input A, B, C; // Τα A, B, C είναι σήματα εισόδου
    output Z; // Το Z είναι σήμα εξόδου
    wire K, L; // Τα K, L είναι τοπικές μεταβλητές. Θα μπορούσαν και να μη
    // δηλωθούν καθόλου. Αν τα δηλώσετε πρέπει να είναι τύπου wire

    and AND1 (K, A, B); // Χρησιμοποιούμε ένα αντίγραφο της πρωταρχικής συνάρτησης
    // AND, που του δίνουμε το όνομα AND1. Οι εισοδοί θα είναι A
    // και B και η έξοδος του K. Αντιπροσωπεύει τη πύλη AND1 του
    // σχήματος 3.1
    and AND2 (L, B, C); // Αντίστοιχα με το παραπάνω ένα δεύτερο αντίγραφο για τη
    // δεύτερη πύλη
    or OR1 (Z, K, L); // Προσέξτε την υπονοούμενη διασύνδεση. Οι μεταβλητές K, & L
    // που οδηγούνται από τις AND είναι και οι εισοδοί της OR.
endmodule
```

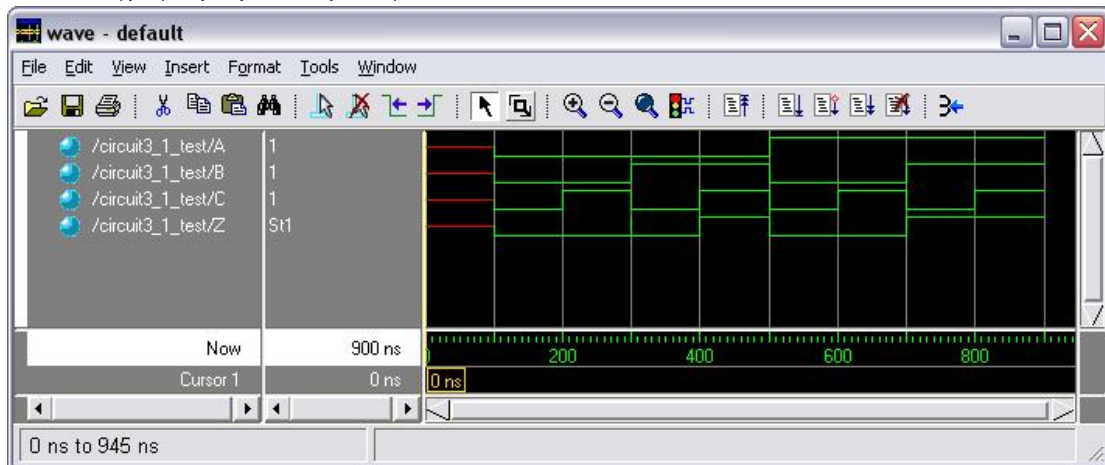
Για να εξομοιώσουμε τη περιγραφή του κυκλώματός μας, χρειάζεται να επιβάλλουμε τιμές πάνω στις εισόδους του. Αυτό μπορεί να γίνει με ένα νέο module της Verilog που ονομάζεται testbench. Ο απαραίτητος κώδικας παρατίθεται πιο κάτω

```
module circuit3_1_test(); // Δεν υπάρχουν εισοδοί και εξοδοί
    reg A, B, C; // Τοπικές μεταβλητές. Επειδή αυτές θέλω να
    // κρατάνε τιμές, πρέπει να τις ορίσω σε reg.
    wire Z; // Τοπική μεταβλητή. Απλά θα μεταφέρει την
    // έξοδο του κυκλώματός μου, άρα είναι τύπου
    // wire.
    circuit3_1 kyklwma3_1(A, B, C, Z); // Διασύνδεση με το κύκλωμά μου. Οτι τιμή
    // βάζω στις τοπικές μεταβλητές A, B, C θα
    // επιβάλλεται και στο κύκλωμά μου. Η απόκρισή
    // του θα υπάρχει στη τοπική μεταβλητή Z.

    initial
    begin
        #100 A = 0; B = 0; C = 0; // Ανά 100 χρονικές στιγμές βάζω μια νέα τιμή
        // στις τοπικές μεταβλητές.
        #100 A = 0; B = 0; C = 1;
        #100 A = 0; B = 1; C = 0;
        #100 A = 0; B = 1; C = 1;
        #100 A = 1; B = 0; C = 0;
        #100 A = 1; B = 0; C = 1;
        #100 A = 1; B = 1; C = 0;
        #100 A = 1; B = 1; C = 1;
        #100 $stop;
    end
endmodule
```

Είμαστε πλέον έτοιμοι να εξομοιώσουμε τη περιγραφή μας. Ακολουθούμε συνεπώς τα εξής βήματα:

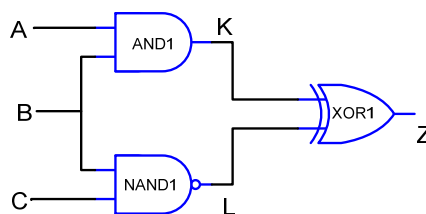
- Αντιγράφουμε τα αρχεία 3_1.v μέσα σε κάποιο φάκελο. Στη συνέχεια υποθέτουμε ότι είναι ο C:\LD1.
- Ανοίγουμε το Modelsim.
- Μεταβαίνουμε (menu File → Change Directory) στο φάκελο C:\LD1.
- Δημιουργούμε μια νέα βιβλιοθήκη. Αφήνουμε το default όνομα work.
- Κάνουμε compile το αρχείο 3_1.v
- Ξεκινάμε εξομοίωση του μεταφρασμένου object circuit3_1_test.
- Ανοίγουμε τα παράθυρα signals και wave.
- Αφού τοποθετήσουμε τα σήματα που θέλουμε πατάμε το Run all και μας προκύπτει η γραφική παράσταση που φαίνεται πιο κάτω.



Η γραφική παράσταση απεικονίζει τα σήματα εισόδου (A, B και C) και το σήμα εξόδου Z. Αν παρατηρήσετε τη γραφική παράσταση θα δείτε ότι κατά τα πρώτα 100ns όλα τα σήματα φαίνονται με κόκκινο χρώμα που σημαίνει ότι η τιμή τους είναι απροσδιόριστη. Στη συνέχεια και κάθε 100ns τα σήματα εισόδου παίρνουν διαδοχικά τις τιμές του πίνακα 3.1, ενώ στην έξοδο παράγεται το θεωρητικά αναμενόμενο (πίνακας 3.1) σήμα Z. Αν συγκρίνετε τις κυματομορφές με τις τιμές του πίνακα 3.1 θα δείτε ότι είναι οι ίδιες με μόνη διαφορά την ύπαρξη μιας στιγμιαίας μεταβολής της τιμής της εξόδου τη χρονική στιγμή 300ns όπου το Z είναι 0 μετά ακαριαία μεταβαίνει στο 1 και μετά στο 0. Το φαινόμενο αυτό ονομάζεται αιχμή (spike). Αιχμές μεγαλύτερης διάρκειας ονομάζονται glitches. Ο εξομοιωτής έχει απόλυτο δίκιο: μιας και στιγμιαία οι εισοδοί B και C παίρνουν την τιμή 1 πολύ σωστά μας υποδεικνύει ότι το κύκλωμά μας στιγμιαία θα μας δώσει έναν 1 στην έξοδο.

Μέρος 2 : Ζητούμενα

Ζητείται να εξάγετε θεωρητικά τον πίνακα αληθείας του κυκλώματος του πιο κάτω σχήματος. Περιγράψτε το κύκλωμα σε Verilog και επιβεβαιώστε μέσω εξομοίωσης το πίνακα αληθείας. Υπάρχουν αιχμές στην έξοδο του κυκλώματος ?



ΠΑΡΑΔΟΤΕΑ :

- *Κώδικας Verilog*
- *Ενα screenshot των κυματομορφών εξομοίωσης.*
- *Παρατήρηση αιχμών και δικαιολόγησή τους*

(Για τη λύση αυτής της άσκησης ακολουθείστε ακριβώς τα ίδια βήματα με το 1^ο Μέρος. Μέρος της απάντησης είναι στο συνοδευτικό αρχείο 3_2.v)