

Ολοκληρωμένα Κυκλώματα

Κυκλώματα συνδυαστικής λογικής σε SystemVerilog

Γιώργος Δημητρακόπουλος

Σε αυτό το εργαστήριο θα σχεδιάσουμε σε SystemVerilog ένα απλό συνδυαστικό κύκλωμα το οποίο θα μας παρέχει κάποιες χρήσιμες πληροφορίες για την εξέλιξη του παιχνιδιού της τρίλιζας.

Η τρίλιζα παίζεται από δύο παίκτες πάνω σε ένα δισδιάστατο ταμπλό με διαστάσεις 3×3 όπως αυτό που φαίνεται στο σχήμα (α). Ο πρώτος παίκτης τοποθετεί στις κενές θέσεις του ταμπλό τα σύμβολα X και ο δεύτερος τα O. Κάποιος παίκτης κερδίζει αν καταφέρει να τοποθετήσει τα 3 σύμβολα που του αντιστοιχούν σε κάποια από τις γραμμές, στήλες ή κύριες διαγωνίους του ταμπλό. Για παράδειγμα, στο στιγμιότυπο του σχήματος (β) ο δεύτερος παίκτης έχει κερδίσει εφόσον σχημάτισε μια τριάδα από O στη δεύτερη γραμμή του ταμπλό.

X		
O	O	
		X

(α)

X		
O	O	O
		X

(β)

0	1	2
3	4	5
6	7	8

(γ)

X	O	
O	O	
	X	X

(δ)

Σχήμα 1: Στιγμιότυπα της τρίλιζας

Για να μπορέσουμε να αντιμετωπίσουμε τη σχεδίαση ενός κυκλώματος που θα μπορεί να παίζει τρίλιζα θα πρέπει πρώτα απ' όλα να ξεκαθαρίσουμε την κωδικοποίηση που έχουμε επιλέξει για την τοποθέτηση των X και των O πάνω στο ταμπλό. Για λόγους απλότητας και μόνο επιλέξαμε να κωδικοποιήσουμε τη θέση των X και των O με δυο 9μπιτες ποσότητες. Χρησιμοποιούμε 9 καλώδια για το X και 9 καλώδια για το O. Για τα X, όταν ένα από τα καλώδια φέρει το λογικό 1 σημαίνει πως στην αντίστοιχη θέση έχει τοποθετηθεί ένα X. Το ίδιο ισχύει και για τα O. Όταν το καλώδιο μιας θέσης είναι και για τα X και για τα O ίσο με λογικό 0 σημαίνει πως η αντίστοιχη θέση του ταμπλό είναι κενή.

Η σειρά με την οποία αριθμήσαμε τις θέσεις του ταμπλό φαίνεται στο σχήμα (γ). Επομένως το στιγμιότυπο που αντιστοιχεί στο σχήμα (δ) κωδικοποιείται ως εξής:

X(8 έως 0): 110 000 001

O(8 έως 0): 000 011 010

Το καλώδιο X(8)=1 δείχνει ότι στη θέση 8 υπάρχει ένα X. Ενώ το καλώδιο O(5)=0 δείχνει ότι στη θέση 5 δεν υπάρχει κάποιο O. Εφόσον και το X(5)=0 και το O(5)=0 τότε αυτό σημαίνει πως η θέση 5 είναι κενή.

Το συνδυαστικό κύκλωμα που καλείστε να σχεδιάσετε σε SystemVerilog θα δέχεται ως είσοδο τα σήματα X και O, και θα παράγει 5 εξόδους.

- Η πρώτη έξοδος error θα ενεργοποιείται όταν τα δεδομένα που αναπαριστούν τα 9μπιτα σήματα X και O της εισόδου δεν είναι έγκυρα. Τα δεδομένα του ταμπλό θεωρούνται έγκυρα όταν κάθε θέση του ταμπλό περιέχει είτε μόνο ένα X, είτε μόνο ένα O είτε είναι κενή.
- Η δεύτερη έξοδος με όνομα full θα ενεργοποιείται όταν δεν υπάρχει κάποια κενή θέση στο ταμπλό. Δηλαδή θεωρώντας πως τα δεδομένα του ταμπλό είναι έγκυρα, σε κάθε θέση του ταμπλό πρέπει να υπάρχει είτε κάποιο O ή είτε κάποιο X υποδηλώνοντας έμμεσα τη λήξη του παιχνιδιού εφόσον δε μπορεί να προστεθεί κάποιο επιπλέον σύμβολο.

- Η τρίτη έξοδος με όνομα winX θα ενεργοποιείται όταν τα X έχουν σχηματίσει μια τριάδα είτε σε κάποια από τις 3 γραμμές, είτε σε κάποια από τις 3 στήλες είτε σε κάποια από τις 2 διαγωνίους του ταμπλό.
- Η τέταρτη έξοδος με όνομα winO θα ενεργοποιείται όταν τα O έχουν σχηματίσει μια τριάδα είτε σε κάποια από τις 3 γραμμές, είτε σε κάποια από τις 3 στήλες είτε σε κάποια από τις 2 διαγωνίους του ταμπλό.
- Η πέμπτη έξοδος με όνομα noWin θα ενεργοποιείται όταν κανένα από τα O ή τα X δεν έχουν κερδίσει.

Όλες οι έξοδοι, full, winX, winO, noWin θεωρούν πως πάντα η κωδικοποίηση των σημάτων X και O είναι έγκυρη. Αν δεν είναι, δε χρειάζεται να το λάβουν υπόψιν τους, εφόσον την κατάσταση αυτή θα τη δείξει η ενεργοποίηση του σήματος error.

Κατά τη σχεδίαση του κυκλώματος σας καλό είναι να αντιμετωπίσετε την κάθε έξοδο χωριστά και να φτιάξετε γι'αυτήν ένα χωριστό υποκύκλωμα που θα την υλοποιεί. Το μόνο σήμα που μπορεί να φτιαχθεί πιο απλά είναι το noWin του οποίου η ενεργοποίηση δε χρειάζεται να προκύπτει απευθείας από τις εισόδους αλλά από τα σήματα winX και winO.

Η εξωτερική δομή του κυκλώματος που καλείσαι να σχεδιάσετε και να εξομοιώστε πλήρως πριν έρθετε στο εργαστήριο είναι η παρακάτω:

```
module tictactoe (
    input logic [8:0] x,
    input logic [8:0] o,
    output logic error,
    output logic full,
    output logic winX,
    output logic winO,
    output logic noWin
);
```

```
...
endmodule;
```

Για την επιβεβαίωση της ορθής λειτουργίας του κυκλώματος σας θα πρέπει να τροφοδοτήσετε το κύκλωμα σας με τις κατάλληλες εισόδους και παρατηρώντας τις εξόδους.

Για την επίδειξη του κυκλώματος σας θα πρέπει να το αντιστοιχίσετε στην FPGA μέσω του Quartus. Μαζί με τη λογική συνθεση, τη χωροθέτηση και τη διασύνδεση του σχεδιασμού σας, θα πρέπει να ορίσετε τους ακροδέκτες εισόδου και εξόδου που θα χρησιμοποιήσετε. Όλοι οι ακροδέκτες της πλακέτας DE1 περιλαμβάνονται στο αρχείο DE1.qsf το οποίο σας έχει διατεθεί. Από το αρχείο αυτό διαλέξτε τις γραμμές που χρειάζεστε και προσθέστε τις στο qsf αρχείο που έχει δημιουργηθεί στο project του σχεδιασμού σας από το Quartus. Για παράδειγμα αν θέλετε να συνδέσετε το σήμα error του σχεδιασμού σας στο κόκκινο LED[0] δεν έχετε παρά να ακολουθήσετε τα παρακάτω βήματα. Εντοπίστε στο DE1.qsf τη γραμμή που αφορά στο κόκκινο LED που επιθυμείτε. Η γραμμή αυτή είναι η εξής:

```
set_location_assignment PIN_R20 -to LEDR[0]
```

Η γραμμή αυτή μας λέει δηλαδή πως το κόκκινο LED είναι προσυνδεδεμένο στην πλακέτα με το PIN_R20 της FPGA. Επομένως, για να οδηγήσουμε εμείς το σήμα εξόδου error του σχεδιασμού μας σε αυτό το LED πρέπει να το συνδέσουμε με τον ακροδέκτη PIN_R20 της FPGA. Για να το πετύχουμε αυτό αντιγράφουμε την παρακάτω γραμμή στο qsf αρχείο του project που έχουμε δημιουργήσει.

```
set_location_assignment PIN_R20 -to error
```

Αφού κάνουμε την ίδια δουλειά για όλα τις εισόδους και τις εξόδους του κυκλώματος μας δεν ξεχνάμε να το επανα-αντιστοιχίσουμε στην FPGA.