



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Αναφορά 2ης Εργασίας : Pong Game - Δεύτερο βήμα
“Εργαστήριο Ψηφιακών Συστημάτων”

ΟΝΟΜΑΤΕΠΩΝΥΜΑ	Οικονόμου Γεώργιος - Χαλβατζόγλου Ιωάννα (ΑΜ: el21103) (ΑΜ: el21906)
ΕΞΑΜΗΝΟ	6ο
ΟΜΑΔΑ	A_Ομάδα 17
ΔΙΔΑΣΚΟΝΤΕΣ	Δ. Πνευματικάτος, Γ. Παναγόπουλος, Ι. Παναγοδήμος

Αθήνα
2024

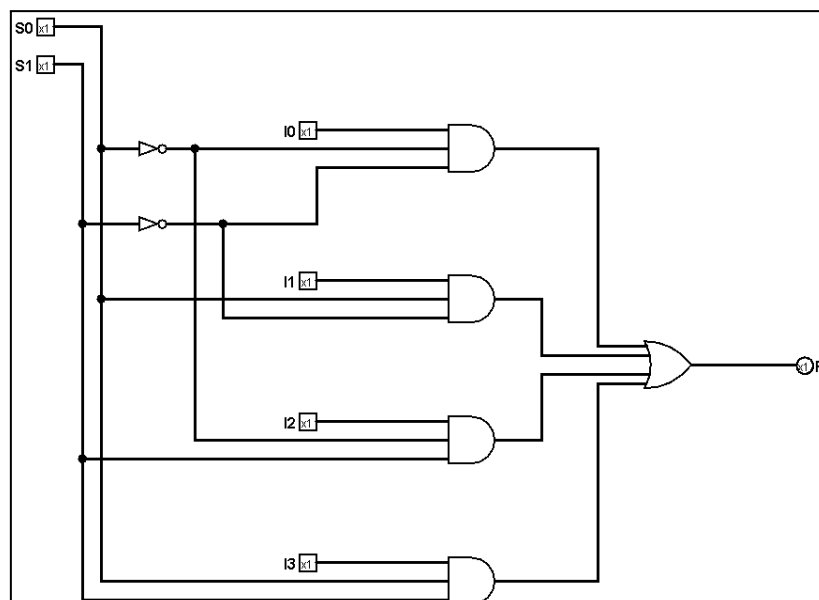
Εισαγωγή

Αρχικά, η υλοποίηση του κυκλώματος σε σχηματικό και η προσομοίωση του παρουσιάζονται με την Προσέγγιση του Καταχωρητή Αμφίδρομης Ολίσθησης, για τους λόγους που αναφέρθηκαν στην προηγούμενη αναφορά. Προχωράμε, ωστόσο, σε μία μικρή τροποποίηση στις διάφορες λειτουργίες των θυρών διαδοχικών πολυπλεκτών. Συγκεκριμένα,

- Όταν $S1S0 = 00$ να πραγματοποιεί αριστερή κυκλική ολίσθηση. Η κατάσταση αυτή μοντελοποιεί την «κατεύθυνση» της μπάλας προς τα αριστερά.
- Όταν $S1S0 = 01$ να επιτελεί παράλληλη φόρτωση από τις γραμμές δεδομένων τον αριθμό 10..0 στο δυαδικό σύστημα αρίθμησης. Η κατάσταση αυτή μοντελοποιεί τον πιεστικό διακόπτη «Reset» στο σχήμα.
- Όταν $S1S0 = 10$ να επιτελεί παράλληλη φόρτωση από τις γραμμές δεδομένων τον αριθμό 10..0 στο δυαδικό σύστημα αρίθμησης. Η κατάσταση αυτή μοντελοποιεί τον πιεστικό διακόπτη «Reset» στο σχήμα.
- Όταν $S1S0 = 11$ να πραγματοποιεί δεξιά κυκλική ολίσθηση. Η κατάσταση αυτή μοντελοποιεί την «κατεύθυνση» της μπάλας προς τα δεξιά.

Οι δύο καταστάσεις των θυρών $I1 - I2$ είναι ίδιες. Ουσιαστικά, λοιπόν, θέλουμε τους πολυπλέκτες να μοντελοποιούν τις δύο βασικές ιδιότητες της «κατεύθυνσης» της μπάλας και του διακόπτη «Reset». Προκειμένου, λοιπόν, να μην έχουμε μία αχρησιμοποίητη κατάσταση στους MUX 4 - 1, και αυτή να μπορεί να οδηγήσει σε απροσδιόριστη, εισάγουμε την πανομοιότυπη του «Reset», που εν τέλει ήταν χρήσιμη απόφαση και για την μοντελοποίηση.

Καθώς οι πολυπλέκτες του LogiSim Evolution δεν επιτρέπουν την εισαγωγή δύο ξεχωριστών select bits, αλλά μόνο την εισαγωγή ενός pin μεγέθους δύο, κατασκευάζουμε ένα subcircuit MUX_Four_to_One, που απλά αποτελεί έναν πολυπλέκτη 4 - 1.



Διαδικασία σχεδίασης

Βήμα 1

Αρχικά, είναι χρήσιμο να καθορίσουμε τις διεπαφές μεταξύ των δύο κομματιών Core και Control. Συγκεκριμένα, η διεπαφή του Core παρέχει πληροφορίες σχετικά με την κατάσταση των LED και την τρέχουσα κατάσταση θέσης του καταχωρητή ολίσθησης. Η διεπαφή του Control επικοινωνεί με το Core προκειμένου να γίνει ανάγνωση της κατάστασης της οθόνης και της θέσης του καταχωρητή ολίσθησης, καθώς και παρέχει εντολές για τον έλεγχο της οθόνης και την κίνηση του καταχωρητή ολίσθησης. Οι διεπαφές μεταξύ των τμημάτων Core και Control παρουσιάζονται στον πίνακα:

ΟΝΟΜΑ	ΠΛΑΤΟΣ (bits)	ΚΑΤΕΥΘΥΝΣΗ	ΠΕΡΙΓΡΑΦΗ
CLK	1 bit	Input στο Core	Clock
		Output του Control	
S1 - S0	1 bit το καθένα	Input στο Core	Σήματα επίτρεψης των MUX που κωδικοποιούν λειτουργίες του κυκλώματος, όπως η «κατεύθυνση».
		Output του Control	
Reset_Left - Reset_Right	1 bit το καθένα	Input στο Core	Σήματα ελέγχου για την αρχικοποίηση της «μπάλας» σε ήττα στα ακραία LED.
		Output του Control	
Output_n, όπου n = 0..7	1 bit το καθένα	Output του Core	LED λαμπάκια που παριστάνουν την κίνηση της «μπάλας».
PlayerOne	1 bit	Input στο Control	Πιεστικός διακόπτης που προσφέρεται για «απόκρουση» της μπάλας από τον παίκτη.
Reset	1 bit	Input στο Control	Πιεστικός διακόπτης που προσφέρεται για αρχικοποίηση από τον παίκτη.

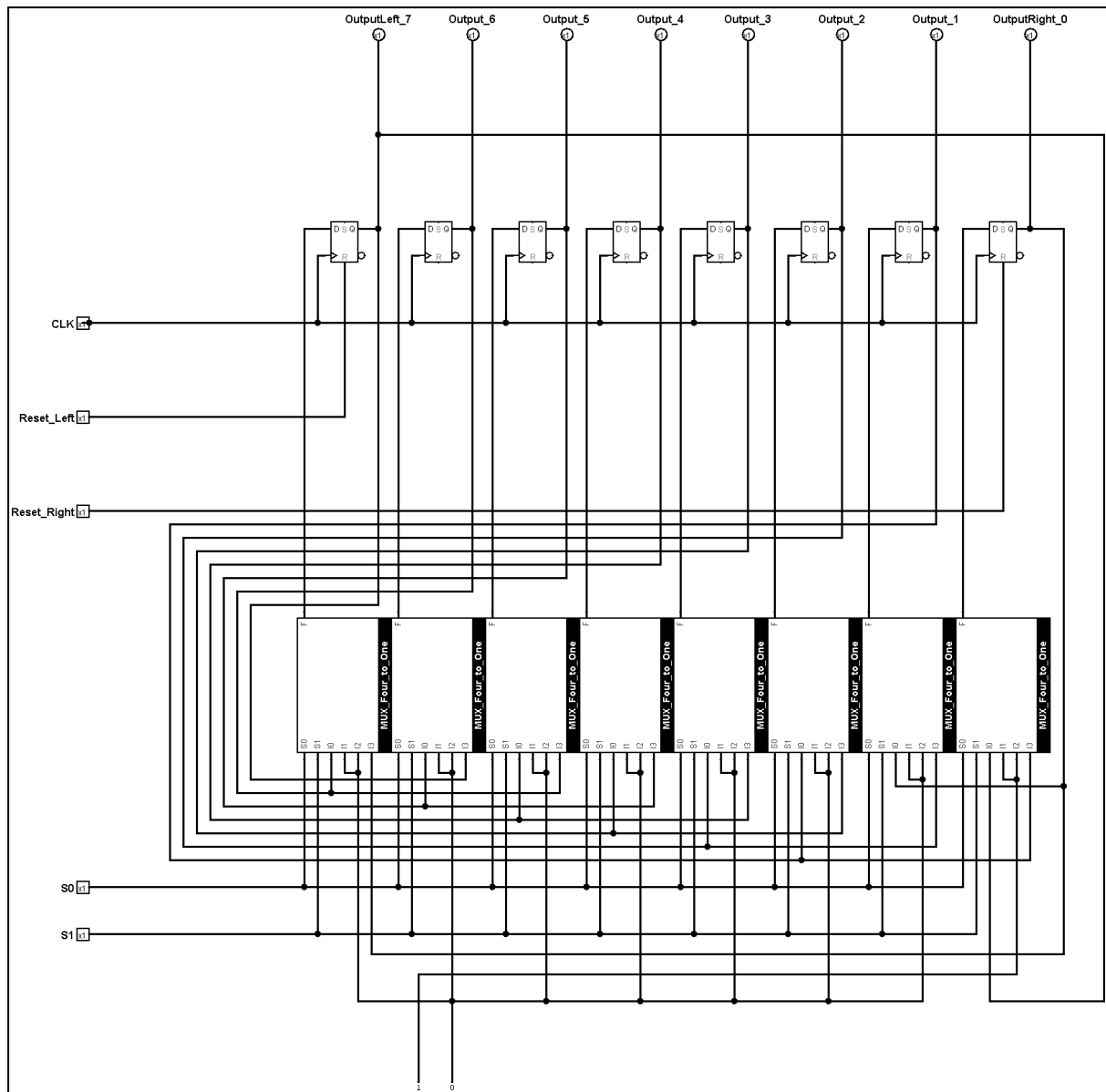
Οι παραπάνω διεπαφές όπως φαίνεται είναι ανεξάρτητες των Control και Core καθώς δεν επηρεάζονται από τον αν κατά την υλοποίηση έχουμε καταχωρητή ολίσθησης ή μετρητή ριπής.

Βήμα 2

Παρουσιάζουμε, αρχικά, την υλοποίηση του Core, το τμήμα που περιέχει το σύνολο των D Flip - Flop και των MUX 4-1. Όλα τα Flip - Flop είναι σύγχρονα με κοινό ρολόι, το οποίο έρχεται από το τμήμα του Control, αλλά διαφέρουν ως προς τα Reset. Έχοντας κατανοήσει πως τα Reset των Flip - Flop, όταν είναι ON, οδηγούν αυτά σε παρούσα κατάσταση ίση με 0, τα χρησιμοποιούμε για την

μοντελοποίηση της ήπτας του παίκτη, όταν δεν αντικρούει την μπάλα στα άκρα LED. Οπότε, διατηρούμε τα Reset των ενδιάμεσων LED ανέγγιχτα.

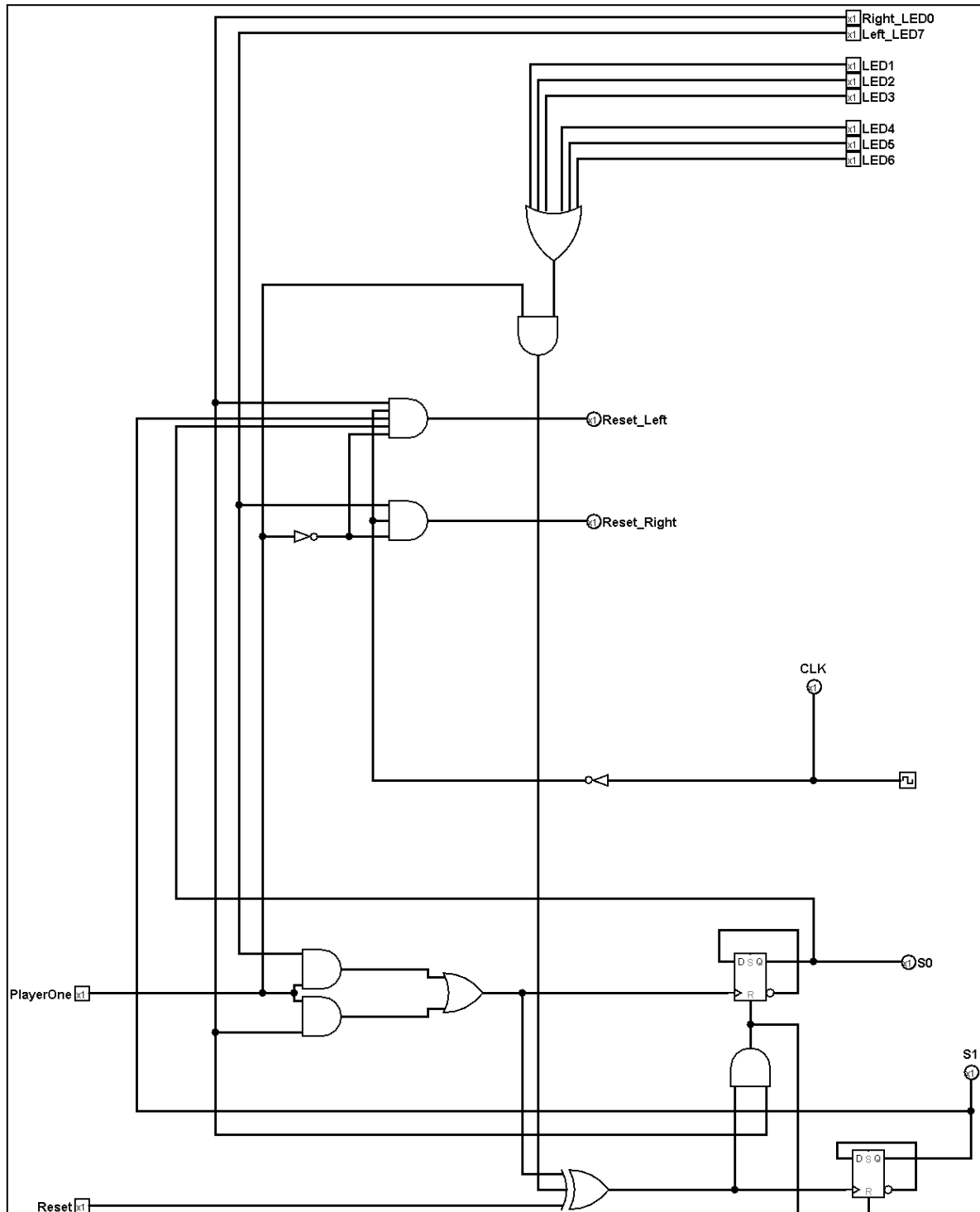
Έπειτα, όταν ο παίκτης έχει αντικρούσει λανθασμένα την «μπάλα» στα ενδιάμεσα LED ή έχει πατήσει τον διακόπτη Reset, αρχικοποιούμε πρώτα την «θέση» και μετά την «κατεύθυνση» της μπάλας, δηλαδή θέτουμε τα $S1S0 = 10$ ή 01 ώστε να γίνει παράλληλη φόρτωση του αριθμού $10..0$ και να ανάψει το πρώτο LED, και, τέλος, τα $S1S0 = 00$ ώστε η μπάλα να κινηθεί προς τα αριστερά - αριστερή κυκλική ολίσθηση.



Οι συνθήκες, λοιπόν, των Reset είναι διαφορετικές και τρεις στον αριθμό, πέρα φυσικά του ίδιου του διακόπτη Reset που καθορίζεται ελεύθερα από τον παίκτη, και παρουσιάζονται στο τμήμα του Control. Αναλυτικά, τα Reset σχεδιάστηκαν ώστε να οδηγούν σε διαφορετικές καταστάσεις, για παράδειγμα είτε όταν ο

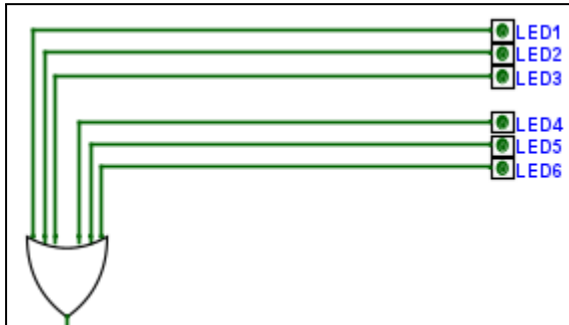
παίκτης πατάει τον διακόπτη «Reset» είτε όταν ο παίκτης χάνει στις ενδιάμεσες καταστάσεις, αρχικοποιούμε με το πρώτο LED ON και στον επόμενο κύκλο, με τον επόμενο χτύπο του διακόπτη «Reset», να ξεκινάει το παιχνίδι από την αρχή. Αντίθετα, όταν ο παίκτης χάνει, επειδή δεν προλαβαίνει στα άκρα να αντικρούσει, τότε το παιχνίδι τελειώνει και όλα τα LED είναι OFF. Αυτό είναι και το τέλος του παιχνιδιού, χρήσιμο για να μην έχουμε αενάως λειτουργία του κυκλώματος.

Στην συνέχεια, παρουσιάζουμε την υλοποίηση του Control.



Αρχικά, παρατηρούμε πως οι δύο «κατευθύνσεις» αποτελούν κατά την Άλγεβρα Boole ένα απλό ζεύγος αντιστρόφων καταστάσεων, δηλαδή, εποπτικά έχουμε ότι $(00)' = 11$ και $(11)' = 00$. Αλλάζουμε «κατεύθυνση» σημαίνει, λοιπόν, ότι αντιστρέφεται το περιεχόμενο της παρούσας κατάστασης. Προκύπτει η ανάγκη να αποθηκεύουμε προσωρινά την παρούσα «κατεύθυνση» ώστε να επηρεάσουμε την επόμενη και, έτσι, επιλέγουμε δύο D Flip - Flop για τα σήματα S1 και S0, τα οποία είναι ασύγχρονα με το CLK ρολόι.

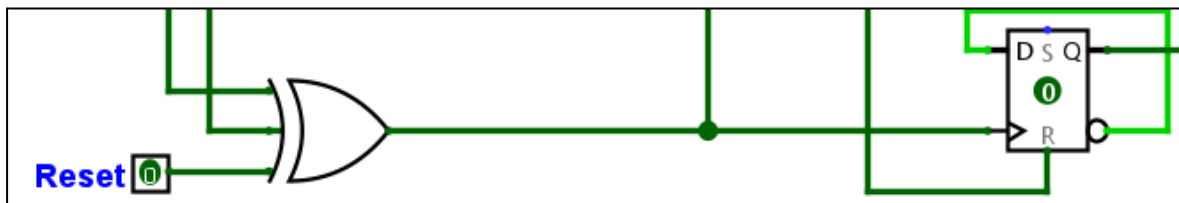
Οι πύλες ακολουθούν την εξής λογική:



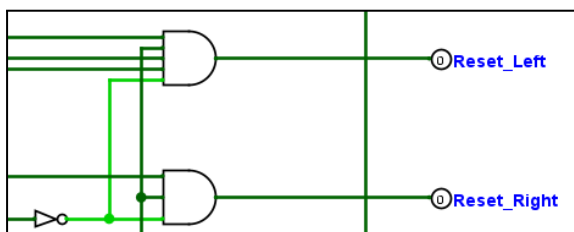
Μέσω της πύλης OR ανιχνεύουμε αν είμαστε σε κάποιο από τα ενδιάμεσα LED.

Χρησιμοποιούμε μία πύλη AND. Αν ο διακόπτης «Player» είναι ON και η OR δίνει, επίσης, ON σημαίνει πως χάσαμε στα ενδιάμεσα επειδή προσπαθούμε να αντικρίσουμε.

Η ήττα αυτή μέσω μιας πύλης XOR οδηγείται σαν παλμός στο ρολόι ενός εκ των δύο D Flip Flop των σημάτων, στο σχήμα το S0, και κάνει Reset, ανοίγοντας μία εκ των θυρών I1 - I2 των MUX (αν είμαστε στην κατάσταση S1S0 = 00, θα οδηγηθούμε στην S1S0 = 01, I1 Reset, αν είμαστε στην S1S0 = 11, θα οδηγηθούμε στην S1S0 = 10, I2 Reset).



Επίσης, έχουμε να υλοποιήσουμε και δύο ήττες για κάθε ένα από τα δύο άκρα LED. Τα υποκυκλώματα αυτά δεν αποτελούν τίποτα άλλο από τα Reset_Left, Reset_Right και παρουσιάζονται εξής:

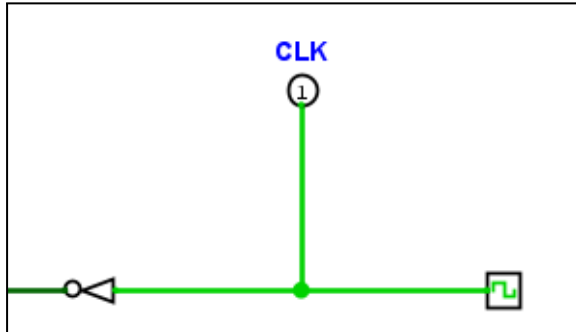


Ο παίκτης χάνει όταν ο διακόπτης του «Player» είναι OFF και το αντίστοιχο LED είναι ON. Χρειαζόμαστε, δηλαδή, δύο πύλες AND.

Ωστόσο, χρειάστηκε να αντιμετωπίσουμε και το πρόβλημα του χρονισμού. Θέλουμε τα πάντα να συμβαίνουν εντός ενός παλμού του ρολογιού. Αν δεν πατήσουμε τον διακόπτη «Player», το εκάστοτε άκρο LED δεν θα ανοίξει καθόλου.

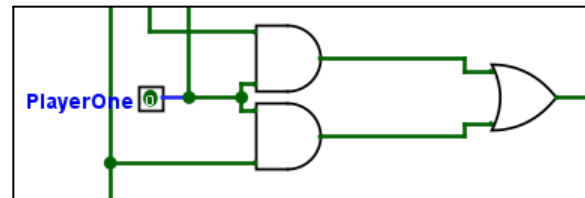
Στην πράξη, στα κλάσματα του χρόνου θα είναι ON αλλά δεν προλαβαίνουμε να

το δούμε εφόσον σβήνει αμέσως λόγω των AND. Σαν λύση, διαιρούμε έναν παλμό του ρολογιού σε δύο τμήματα, το ON (1) και το OFF (2). Εφόσον χρησιμοποιούμε μόνο θετικά ακμοπυρόδοτητα Flip - Flop θέλουμε στο High - ON του ρολογιού τα LED να είναι αναμέννα και στο Low - OFF του ρολογιού να ενημερώνεται το αντίστοιχο Reset του εκάστοτε D Flip - Flop στο Core.



Το επιτυγχάνουμε οδηγώντας στις δύο AND το NOT CLK.

Τέλος, η κατάσταση την νίκης του παίκτη ελέγχεται μέσω δύο AND, όταν δηλαδή ο διακόπτης «Player» είναι ON και, επίσης, κάποιο από τα άκρα LED. Οι δύο νίκες οδηγούνται σε μία πύλη OR.



Το σήμα της OR οδηγείται ταυτόχρονα στα ρολόγια των δύο D Flip Flop των σημάτων S1, S0 και, έτσι, αλλάζει η «κατεύθυνση».

Βήματα 3 & 4

Προκειμένου να πραγματοποιηθεί ο έλεγχος ορθής λειτουργικότητας του κυκλώματος, είναι φρόνιμο να γίνει και ο αντίστοιχος έλεγχος των επιμέρους τμημάτων του. Έτσι, μειώνεται ο κίνδυνος να βρεθούμε αντιμέτωποι με κάποιο σφάλμα στο Top Level της κατασκευής.

Ωστόσο, ο τρόπος αποσπασματικού σχεδιασμού που επιλέχθηκε για τα μέρη που στελεχώνουν την προσομοίωση δεν μας επιτρέπει άμεσα να ελέγξουμε διεξοδικά όλα τα πιθανά σενάρια. Οπότε, για το debugging των ξεχωριστών τμημάτων Core και Control αρκεστήκαμε στο μερικό τρέξιμο των προσομοιώσεων που δημιουργήθηκαν, καθώς και στον έλεγχο των λογικών πράξεων που μας οδήγησαν στην σύλληψη της ιδέας υλοποίησης από την αρχή.

Έχοντας πλέον φτάσει στο επίπεδο του «TopLevel», όπου και πραγματοποιήθηκε η ολοκληρωτική σύνθεση της κατασκευής, το τρέξιμο της προσομοίωσης που δημιουργήθηκε γίνεται πλέον εφικτό και είμαστε σε θέση να εντοπίσουμε με μεγαλύτερη σαφήνεια σφάλματα που πιθανόν προέκυψαν και δεν ήταν εύκολα εντοπίσιμα με «το μάτι».

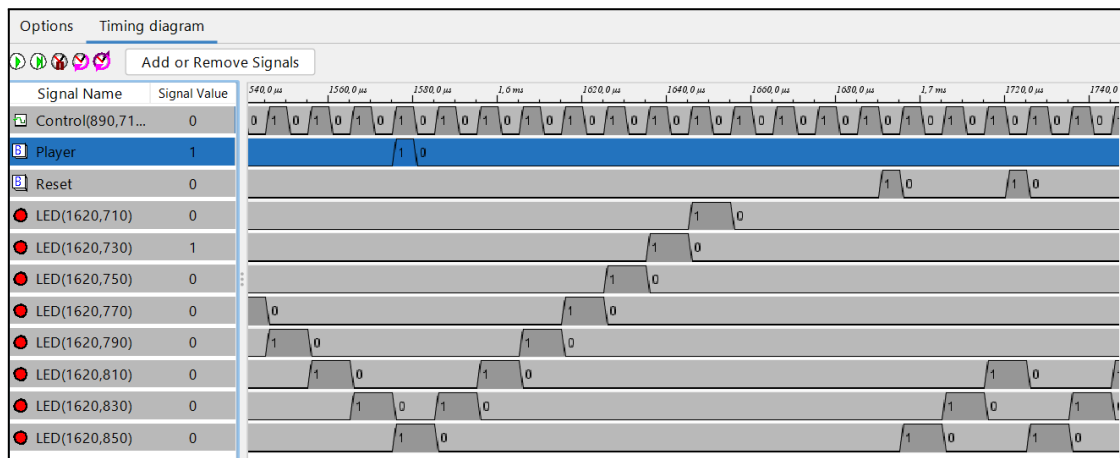
Παρακάτω, ακολουθούν τα επιμέρους κρίσιμα σημεία των οποίων η σωστή λειτουργία μας απασχολεί ιδιαίτερα, καθώς και οι γραφικές παραστάσεις (Timing

Diagrams) που προκύπτουν κατά το τρέξιμο της προσομοίωσης που μας επιτρέπουν να επιβεβαιώσουμε τα ζητούμενα.

Αναλυτικά, ελέγχονται τα ακόλουθα:

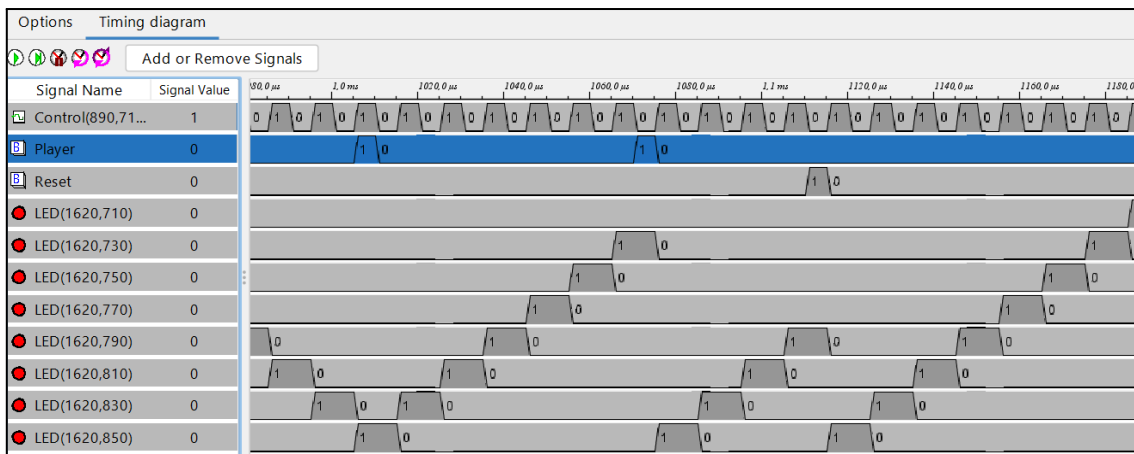
- Ήττα κατά τις ακραίες θέσεις:

Το φαινόμενο μπορεί να εντοπιστεί τόσο στην αριστερή όσο και στην δεξιά ακραία θέση της μπάλας. Όταν αυτό συμβεί στόχος είναι το κύκλωμα να δώσει εντολή Reset και η μπάλα να τοποθετηθεί στην εκκίνηση LED0. Αυτό που αναμένεται βλέπουμε να συμβαίνει στην υλοποίηση μέσω της παρακάτω γραφικής, η οποία αντιστοιχεί σε παράδειγμα ενός από τους συνδυασμούς ήττας σε ακραία θέση.



- Ήττα κατά τις ενδιάμεσες θέσεις:

Η ήττα σε αυτήν την περίπτωση προέρχεται από την προσπάθεια απόκρουσης της μπάλας από τον παίκτη όσο εκείνη βρίσκεται σε κάποιο από τα LED 1 έως 6. Όταν αυτό συμβεί, όπως και στην περίπτωση της ήττας στις ακραίες θέσεις αναμένουμε να γίνει Reset του κυκλώματος. Αυτό το ενδεχόμενο επαληθεύτηκε εξίσου κατά την προσομοίωση καθώς λάβαμε το ακόλουθο διάγραμμα:



- Ορθή απόκριση μπάλας και αμφίδρομη μετακίνησή της:

Σε αυτό το στάδιο ελέγχου είναι σημαντικό να διαπιστωθεί πως η μπάλα είναι ικανή να ανταποκριθεί στην αντίδραση του παίκτη μέσω του μπουτόν Player στις ακραίες θέσεις, δηλαδή στα LED 0 και 7, και να κινηθεί αμφίδρομα. Για άλλη μια φορά η επιβεβαίωση της θεωρίας συμπληρώθηκε με την αντίστοιχη απόδειξη μέσω γραφικής που παρουσίασε η προσομοίωσή μας.

Συγκεκριμένα, η δεξιά ολίσθηση της μπάλας όσο αυτό «ανεβαίνει», ενώ όσο «κατεβαίνει» μας παρουσιάζει την αριστερή ολίσθηση της μπάλας.

