



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΗΜΜΥ

ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ

ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΤΟ ΜΑΘΗΜΑ:

ΗΡΥ 302 – Οργάνωση Υπολογιστών

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2025

Χαράλαμπος Μυλωνάκης 2022030133

Κωνσταντίνος Κυρκός 2022030112

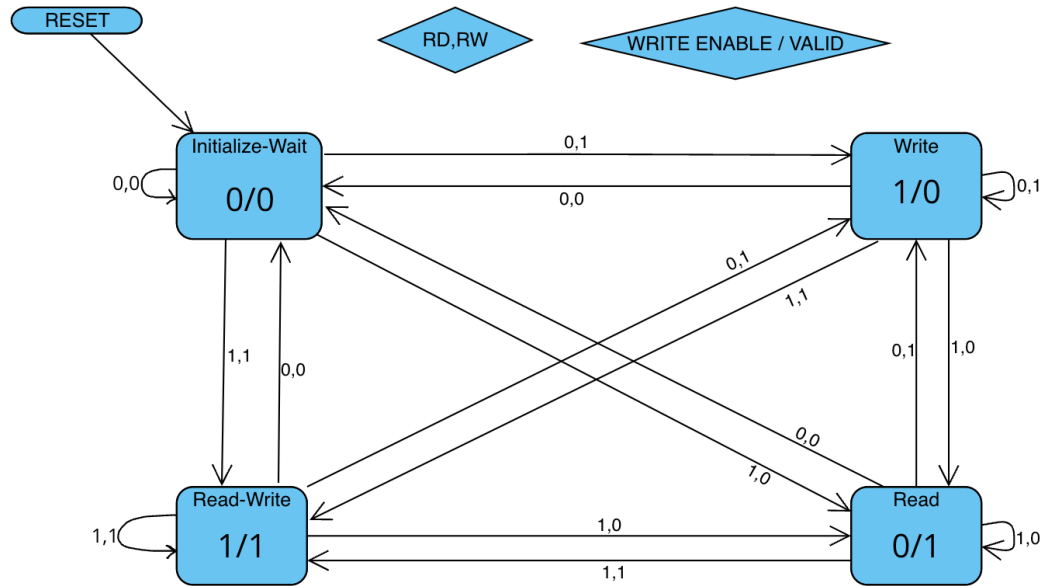
1 Σκοπός - Ζητούμενα

Σκοπός της άσκησης ήταν να μοντελοποιήσουμε και να υλοποιήσουμε μια μονάδα μνήμης όπου έχει δυνατότητα εγγραφής και ανάγνωσης 16 бит αριθμών με την χρήση του Core Generator μέσω ενός κυκλώματος με ΕΙΣΟΔΟΥΣ-ΕΞΕΔΟΥΣ

ΟΝΟΜΑ	ΙΝ/ΟΥΤ	ΒΙΤ
CLK	ΙΝ	1
RST	ΙΝ	1
AddrWrite	ΙΝ	5
AddrRead	ΙΝ	5
Write	ΙΝ	1
Read	ΙΝ	1
NumberIN	ΙΝ	16
NumberOUT	ΟΥΤ	16
Valid	ΟΥΤ	1

Πίνακας 1: : Είσοδοι - έξοδοι του κυκλώματος.

2 FSM Διάγραμμα καταστάσεων



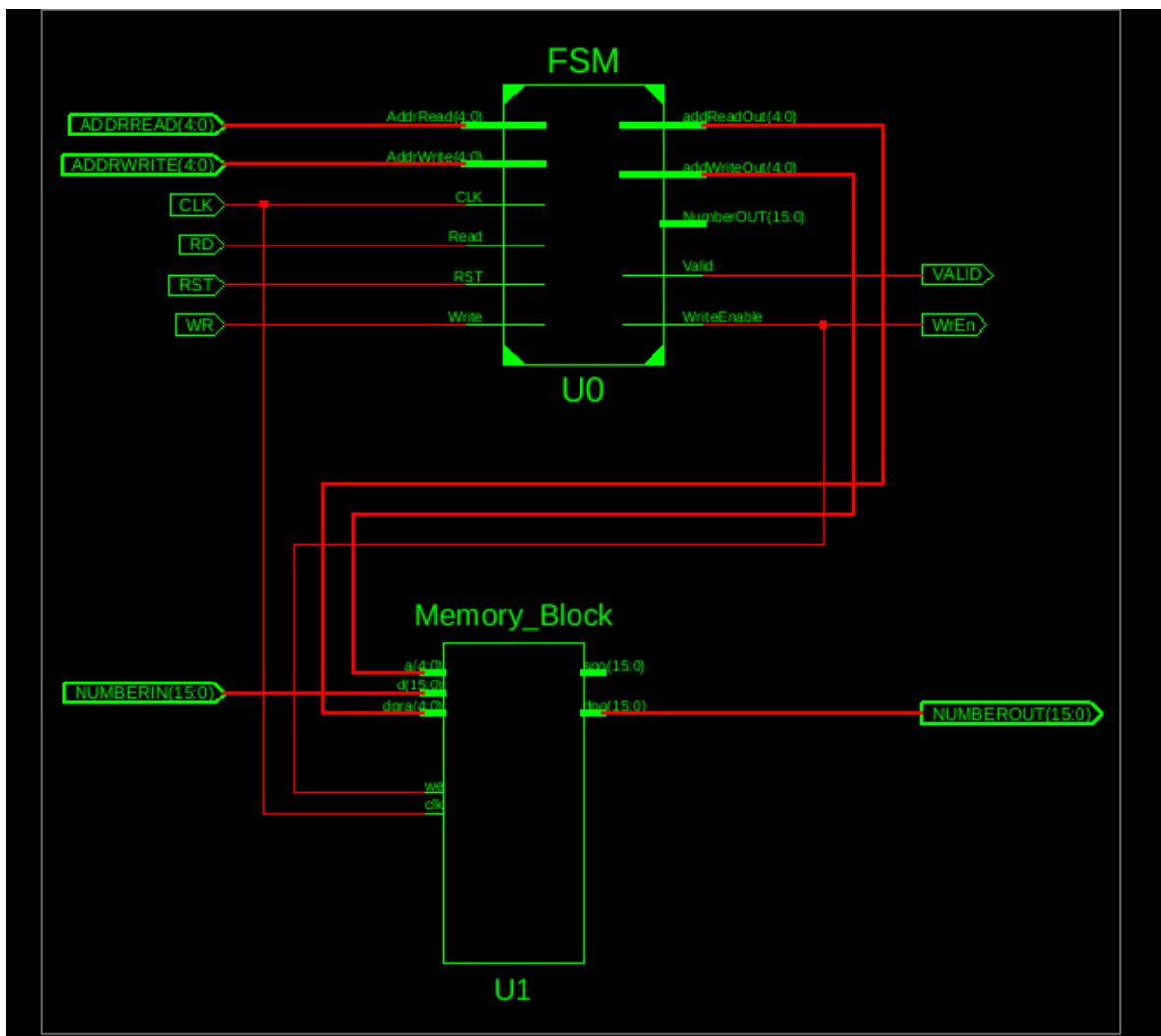
Σχήμα 1: FSM STATE DIAGRAM

3 Σχεδίαση FSM-MemoryBlock –Block Diagram

Αρχικά, δημιουργήσαμε το memory block χρησιμοποιώντας το Distributed Memory Block του Core Generator Xilinx και συγκεκριμένα το dual port Ram (έγινε και αρχικοποίηση με coe file όπως ορίζεται από την εκφώνηση) και την μηχανή καταστάσεων. Έπειτα, ενώσαμε τα δύο components με ένα TopLevel vhdl entity στο οποίο δημιουργήσαμε 3 signals τα οποία συνδέουν κατάλληλα την μνήμη με την μηχανή καταστάσεων. Πιο συγκεκριμένα, έχουμε:

- Ένα καλώδιο που ενώνουμε την διεύθυνση που θέλουμε να γράψουμε στην μνήμη(input) με την διεύθυνση της μνήμης.
- Ένα καλώδιο που ενώνουμε το Write Enable της μηχανής καταστάσεων με την μνήμη για να δηλώσουμε πότε θέλουμε να γράψουμε.
- Ένα καλώδιο που ενώνουμε την διεύθυνση που θέλουμε να διαβάσουμε από την μνήμη με την μνήμη.

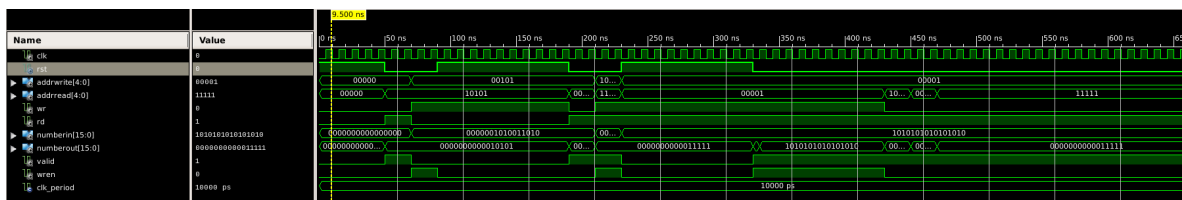
Τέλος, δημιουργήσαμε ένα output Valid έτσι ώστε να ξεχωρίζουμε πότε το NumberOut που βλέπουμε είναι αποτέλεσμα ανάγνωσης και όχι invalid value.



Σχήμα 2: Block Diagram

4 Δοκιμή-TestBench

Οι περιπτώσεις είναι (Αριθμός Καταστάσεων * Αριθμός Πιθανών Έπόμενων Καταστάσεων) $4*4=16$ πιθανές περιπτώσεις. Στο σχήμα 3 παρακάτω παρουσιάζονται ακριβώς η απόκριση του συστήματος σε όλες τις περιπτώσεις. Το κύκλωμα μας είναι πλήρως λειτουργικό και συμπεριφέρεται ακριβώς όπως ορίζεται από την εκφώνηση. Η μεταπήδηση από κατάσταση σε κατάσταση, η εγγραφή και η ανάγνωση από και προς την μνήμη και ορθή λειτουργία των σημάτων επιτυγχάνονται πλήρως. Αυτό αποδεικνύεται από τα TestBench και τα screenshot που ακολουθούν:



Σχήμα 3: TestBench

5 Παρατηρήσεις - Συμπεράσματα

Παρατηρήσαμε στο TestBench ότι το WriteEnable και το Valid δεν ήταν συγχρονισμένα με το ίδιο το Write και Read αντίστοιχα για αυτό, επιλέξαμε να τα αλλάζουμε με το που έχουμε ακμή του ρολογιού και έτσι συγχρονίστηκαν.

Γενικά, η επιτυχία της σχεδίασης επιβεβαιώθηκε από τα αποτελέσματα που έδειξε το TestBench.