Παναγιώτου Δήμητρα

ΑΕΜ 2595

ECE333

LAB 4(LCD Controller: Sitronix ST7066U)

Γενικά:

Ο στόχος της τέταρτης εργαστηριακής εργασίας είναι η υλοποίηση ενός Οδηγού ΄Ενδειξης LCD (Liquid Crystal Display) για την πλακέτα Spartan 3E. Στην τελική μορφή ο οδηγός LCD θα πρέπει να εμφανίζει ενα μήνυμα 31 ή περισσότερων χαρακτήρων, μαζί με ενα δρομέα (cursor), ο οποίος θα επιδεικνύει την ανανέωση της οθόνης.

Pins: Υπάρχουν δύο δυνατές υλοποιήσεις του LCD controller , η πρώτη υλοποίηση διαθέτει εύρος 8 bits ενώ η δεύτερη διαθέτει εύρος 4 bits. Οι σχεδιαστές της Spartan-3E επέλεξαν να χρησιμοποιήσουν την δεύτερη υλοποίηση των 4 bits. Το σήμα LCD\_RW θα μείνει στο 0 καθώς στην συγκεκριμένη εργαστηριακή άσκηση δεν απαιτείται να διαβαστούν δεδομένα από το display. Το σήμα LCD\_E οδηγείται στο 0 , όταν επιθυμούμε να αγνοηθούν όλα τα inputs. Το σήμα LCD\_RS οδηγείται στην τιμή 1 όταν εκκινείται μία διαδικασία εγγραφής , ένω όταν το σήμα LCD\_RS τίθεται στο 0 σηματοδοτεί την εκτέλεση μίας εντολής.

LCD Memory: Το LCD device διαθέτει τρεις εσωτερικές μνήμες:

* DDRAM(Data Display RAM) ,η οποία περιλαμβάνει τα δεδομένα που θα προβληθούν στην οθόνη.
* CGRAM(Character Generator RAM), η οποία αποθηκεύει τα καθορισμένα από το χρήστη μοτίβα που δεν περιλαμβάνονται στο by-default μοτίβα του ελεγκτή.
* CG ROM(Character Generator), η οποία περιλαμβάνει κάποια συγκεκριμένα μοτίβα χαρακτήρων που αντιστοιχούν σε χαρακτήρες ASCII.

Υλοποίηση LCD Ελεγκτή

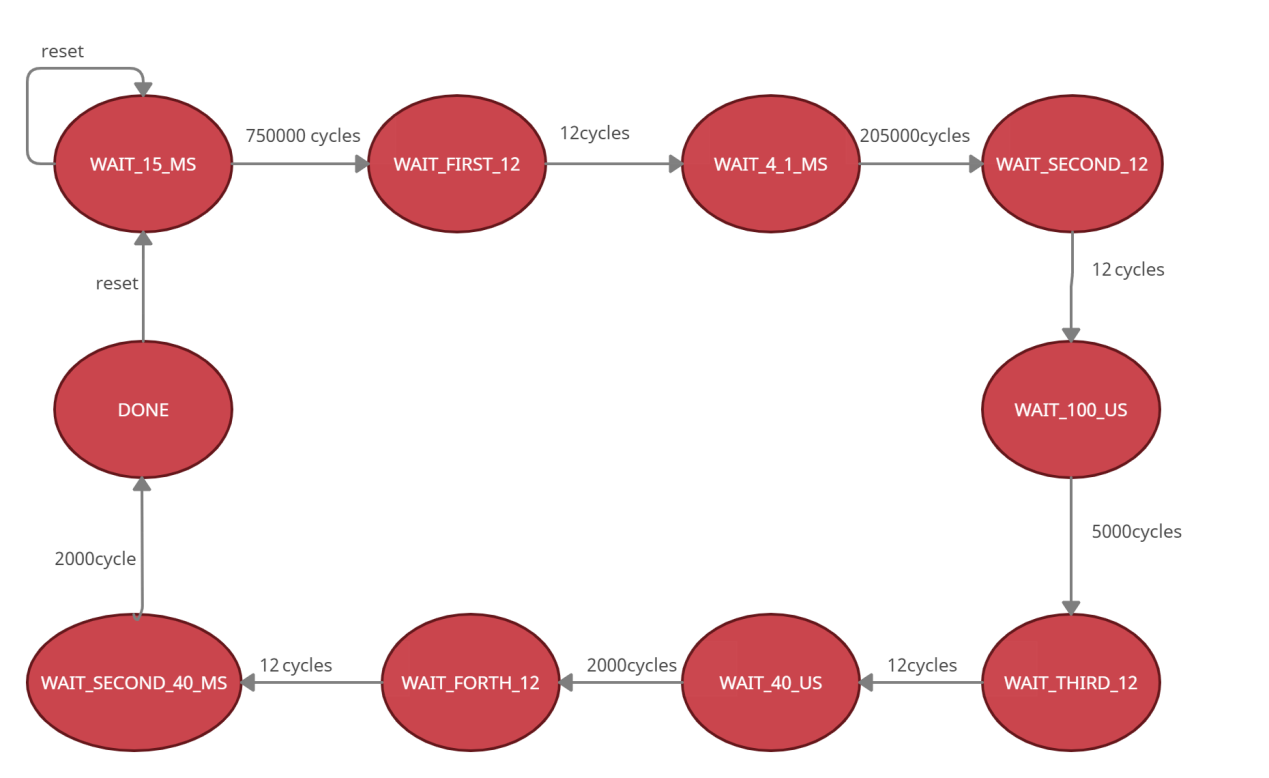
Η υλοποίηση του ελεγκτή περιλαμβάνει τρεις μονάδες πολλαπλών καταστάσεων την μονάδα αρχικοποίησης , την μονάδα για το επίπεδο εντολής και την κύρια μονάδα. Τα 3 αυτά modules συνδέονται κατάλληλα για να προβάλουν τελικά ένα μήνυμα 32 χαρακτήρων στο display. Παρακάτω παρουσιάζεται η αναλυτική λειτουργία των μονάδων με τα σχετικά διαγράμματα καταστάσεων καθώς και τις κυματομορφές που επιβεβαιώνουν την ορθή λειτουργία τους.

Initialization Fsm

Η ΜΠΚ αρχικοποίησης έχει ως βασικό στόχο την αρχικοποίηση του συστήματος. Η διαδικασία της αρχικοποίησης περιλαμβάνει την διαχείριση κάποιων σημάτων για συγκεκριμένο χρονικό διάστημα .Παρακάτω παρουσιάζεται επιγραμματικά η διαδικασία αρχικοποίησης καθώς και τα στάδια της ΜΠΚ που υλοποιήθηκε.

|  |  |  |  |
| --- | --- | --- | --- |
| STATE | I\_LCD\_E | I\_SF\_D | TIMING |
| WAIT\_15\_MS | 0 | 0x00 | 15ms |
| WAIT\_FIRST\_12 | 1 | 0x03 | 240ns |
| WAIT\_4\_1\_MS | 0 | 0x00 | 4.1ms |
| WAIT\_SECOND\_12 | 1 | 0x03 | 240ns |
| WAIT\_100\_US | 0 | 0x00 | 100us |
| WAIT\_THIRD\_12 | 1 | 0x03 | 240ns |
| WAIT\_40\_US | 0 | 0x00 | 40us |
| WAIT\_FORTH\_12 | 1 | 0x02 | 240ns |
| WAIT\_SECOND\_40\_US | 0 | 0x00 | 40us |
| DONE | 0 | 0x00 | until next reset |

Σχηματικά:



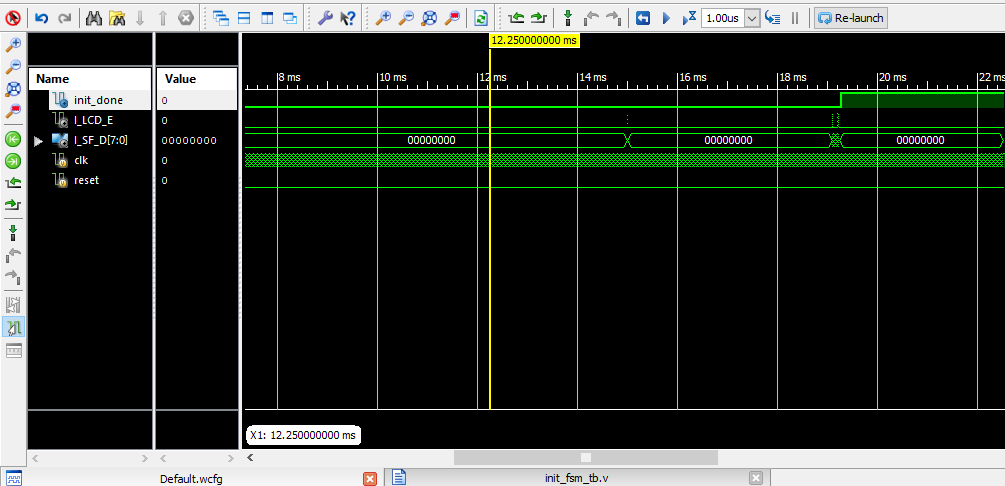
Η αρχικοποίηση ενεργοποιεί την ένδειξη, και στέλνει τις πρώτες και απαραίτητες εντολές διαμόρφωσης των κατάλληλων παραμέτρων, και εκκαθάρισης της οθόνης. Στην ΜΠΚ παράγεται ένα σήμα init\_done το οποίο φέρει την πληροφορία για την περάτωση της αρχικοποίησης και την ετοιμότητα του συστήματος για την διαδικασία διαμόρφωσης.

ΥΛΟΠΟΙΗΣΗ

Η υλοποίηση της ΜΠΚ για την αρχικοποιήση που περιγράφηκε παραπάνω περιλαμβάνει 3 always block:

* Το 1 always block είναι υπεύθυνο για την διαχείριση των τιμών του counter για την αλλαγή των states και την επίτευξη του κατάλληλου χρονισμού.
* Το 2 always block ενεργοποιείται σε κάθε θετική ακμή ρολογιού και είναι υπεύθυνο για την ανάθεση state = next\_state.
* Το 3 always block ενεργοποιείται σε κάθε αλλαγή του counter ή του state και περιλαμβάνει την διαχείριση των σημάτων για κάθε state καθώς και την μετάβαση στο επόμενο στάδιο ανάλογα με την τιμή του counte.

ΕΠΑΛΗΘΕΥΣΗ ΛΕΙΤΟΥΡΓΙΑΣ

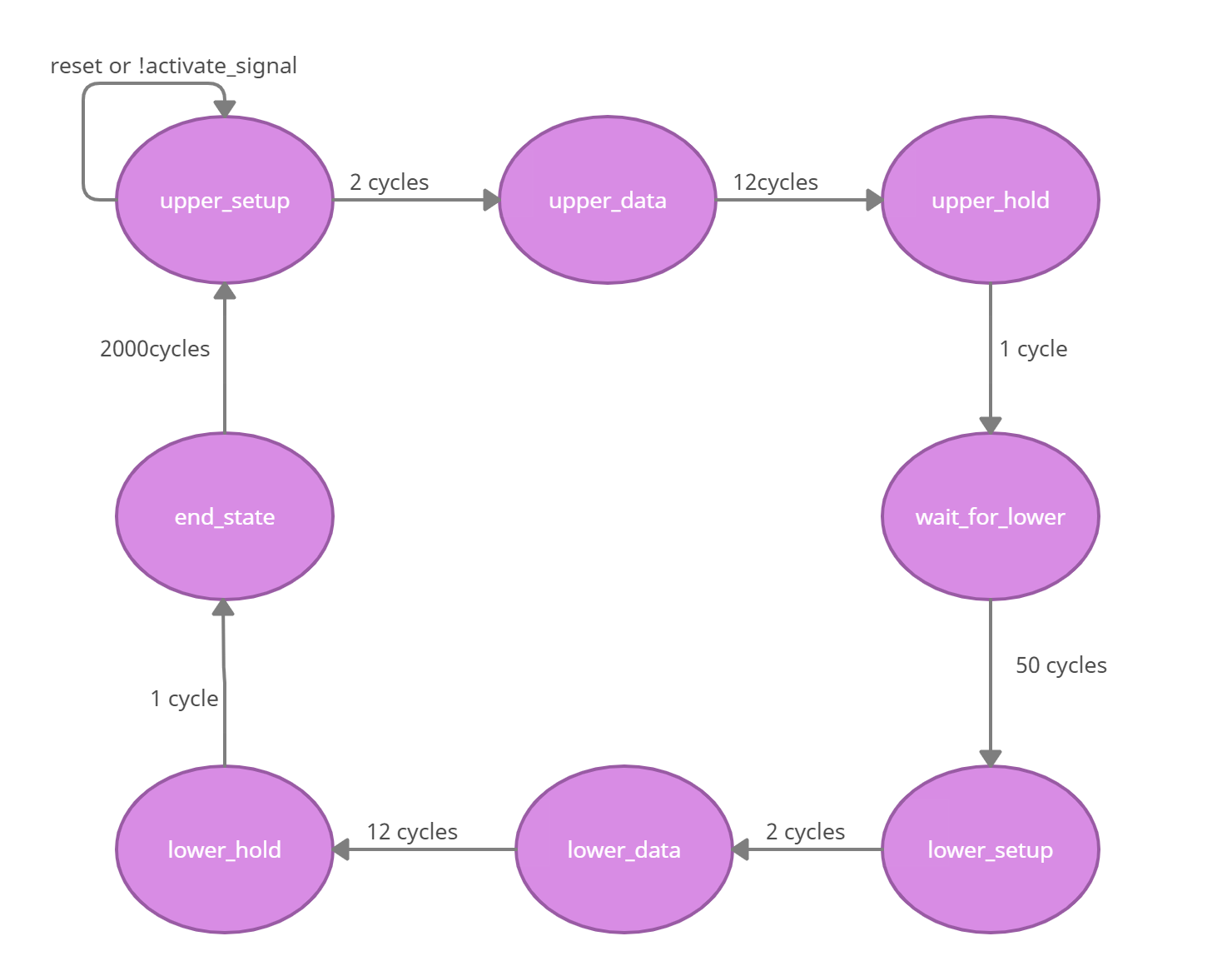


Instruction Fsm

Η ΜΠΚ μεταφοράς έχει σαν βασικό στόχο την διασφάλιση του κατάλληλου χρονισμού για τα σήματα LCD\_E,LCD\_RS και SF\_D<11:8>.Πιο συγκεκριμένα εξασφαλίζει την διατήρηση των παραπάνω σημάτων για συγκεκριμένα χρονικά διαστήματα ανάλογα με το state της μονάδας πολλαπλών καταστάσεων.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STATE | LCD\_E | LCD\_RS | SF\_D | TIMING |
| upper\_setup | 0 | CONSTANT | SF\_D<7:4> | 40ns |
| upper\_data | 1 | CONSTANT | SF\_D<7:4> | 240ns |
| upper\_hold | 0 | CONSTANT | SF\_D<7:4> | 20ns |
| wait\_for\_lower | 0 | CONSTANT | SF\_D<7:4> | 1μs |
| lower\_setup | 0 | CONSTANT | SF\_D<3:0> | 40ns |
| lower\_data | 1 | CONSTANT | SF\_D<3:0> | 240ns |
| lower\_hold | 0 | CONSTANT | SF\_D<3:0> | 20ns |
| end\_state | 0 | CONSTANT | SF\_D<3:0> | 40μs |

Σχηματικά:



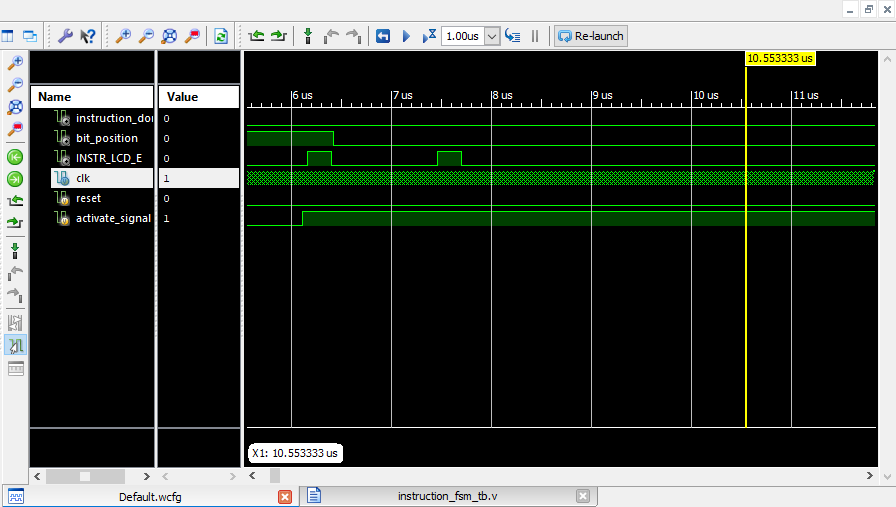
Πρακτικά η μονάδα λαμβάνει ένα σήμα ενεργοποίησης δηλαδή την πληροφορία ότι μία εντολή είναι έτοιμη για εκτέλεση (activate\_signal=instruction\_enable), με σκοπό να εκκινήσει την διαδικασία μετάδοσης της. Η μονάδα αυτή παράγει και ένα σήμα instruction\_done, το οποίο έχει σκοπό να πληροφορήσει το σύστημα για την περάτωση της μεταφοράς μίας εντολής. Παράλληλα, μέσω ενός σήματος bit\_position, το σύστημα γνωρίζει ποία bit του SF\_D να επιλέξει ανάλογα με το στάδιο στο οποίο βρισκόμαστε.

ΥΛΟΠΟΙΗΣΗ

Η υλοποίηση της ΜΠΚ για την διασφάλιση του χρονισμού που περιγράφηκε παραπάνω περιλαμβάνει 3 always block και 1 assign statement:

* Το 1 always block είναι υπεύθυνο για την διαχείριση των τιμών του counter για την αλλαγή των states και την επίτευξη του κατάλληλου χρονισμού.
* Το 2 always block ενεργοποιείται σε κάθε θετική ακμή ρολογιού και είναι υπεύθυνο για την ανάθεση state = next\_state.
* Το 3 always block ενεργοποιείται σε κάθε αλλαγή του counter ή του state και περιλαμβάνει την διαχείριση των σημάτων για κάθε state καθώς και την μετάβαση στο επόμενο στάδιο ανάλογα με την τιμή του counter.
* Το assign statement είναι υπεύθυνο για την ενημέρωση του συστήματος σχετικά με το ποία bit του SF\_D θα πρέπει να επιλέξει για μεταφορά.

ΕΠΑΛΗΘΕΥΣΗ ΛΕΙΤΟΥΡΓΙΑΣ

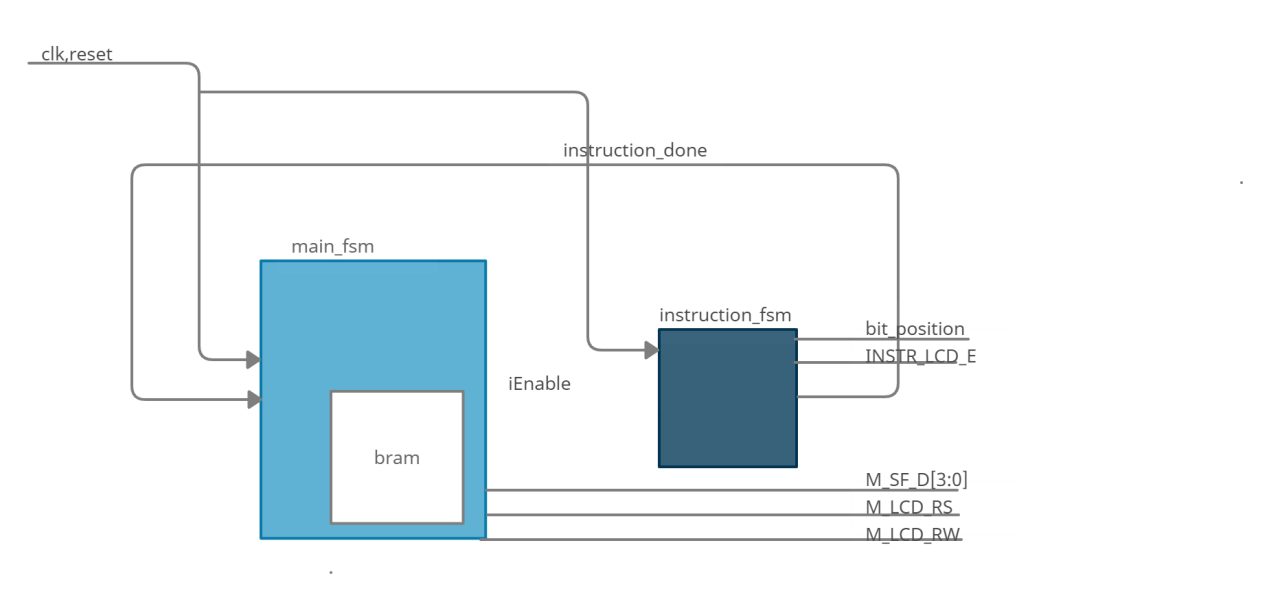


Post and Route Simulation instruction\_fsm

Μain Fsm

H MΠΚ αυτή είναι υπεύθυνη για την διαμόρφωση και τελικά την απεικόνιση των δεδομένων στην οθόνη. Εσωτερικά αρχικοποιείται κατάλληλα μια bram με τα δεδομένα που θέλουμε να προβάλουμε στην οθόνη και με ορθή προσπέλαση της μνήμης επιτυγχάνουμε τον τελικό μας στόχο. Η λειτουργία της main fsm είναι άρρηκτα συνδεδεμένη με την λειτουργία της instruction fsm , αφού η δεύτερη κρίνεται απαραίτητη για την περάτωση μίας εντολής. Πιο πρακτικά η main fsm διαθέτει ένα σήμα iEnable το οποίο συνδέεται με την instruction fsm κάθε φορά που πρέπει να αποσταλεί μία εντολή, η instruction fsm με την σειρά της, αν και εφόσον έχει ολοκληρώσει τους χρονισμούς για την περάτωση της εντολής, ενημερώνει την κύρια ΜΠΚ αναθέτοντας την τιμή 1 στο σήμα instruction\_done.

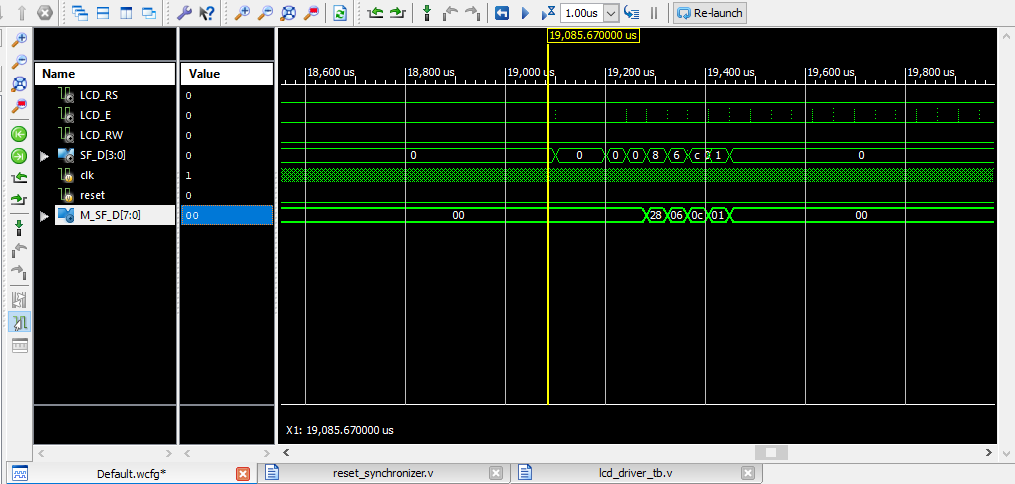
Σχηματικά η συνδεσμολογία που περιγράφηκε παραπάνω:



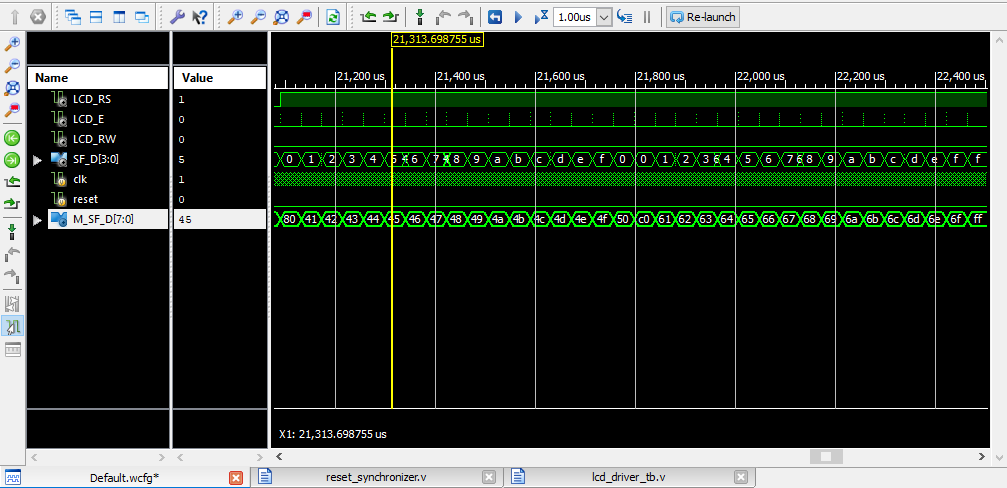
Συνοπτικά τα states:

|  |  |  |  |
| --- | --- | --- | --- |
| STATES | M\_SF\_D[7:0] | M\_LCD\_RS | M\_LCD\_RW |
| IDLE | 0x00 | 0 | 0 |
| FUNCTION\_SET | 0x28 | 0 | 0 |
| SET\_DISPLAY | 0x06 | 0 | 0 |
| CLEAR\_DISPLAY | 0x0C | 0 | 0 |
| WAIT\_1\_64\_MS | 0x01 | 0 | 0 |
| SET\_DDRAM\_ADD\_TOP | 0x00 | 0 | 0 |
| DISPLAY\_TOP\_CHARACTER | 0x80 | 1 | 0 |
| SET\_DDRAM\_ADD\_BOTTOM | char | 1 | 0 |
| DISPLAY\_BOTTOM\_CHARACTER | 0xC0 | 1 | 0 |
| REFRESH\_STATE | 0x00 | 1 | 0 |
| RETURN\_CURSOR\_HOME | 0x02(|3) | 1 | 0 |

ΕΠΑΛΗΘΕΥΣΗ ΛΕΙΤΟΥΡΓΙΑΣ TOP MODULE



Behavioral Simulation Configuration Process



Behavioral Simulation Display Process