Οδηγός Ένδειξης 7-τμημάτων

Εργαστήριο Ψηφιακών Συστημάτων (2023-24)

Ιωάννης Αθανασιάδης 03491

09/11/2023

# Περίληψη

Αναφορά για την εργασία του μαθήματος του Εργαστηρίου Ψηφιακών Κυκλωμάτων (ECE333), μέσω της οποία γίνεται ανάλυση των μεθόδων *ανάπτυξης* και *debugging* ενός *RTL design* στα πλαίσια του προγράμματος ***Xilinx Vivado*** και της ***Digilent Nexys A7-100T*** *FPGA*. Για να το κάνουμε αυτό αναλύουμε με διάφορους μεθόδους (όπως σχήματα ροής δεδομένων) τις κυκλωματική υλοποίηση της ***Verilog*** που αποτελούνε την εργασίας.

# Εισαγωγή

Ο στόχος της εργασίας ήταν η οδήγηση μιας τετραψήφιας οθόνης 7-τμημάτων που είναι ενσωματωμένη στην *Nexys A7-100T*. Πιο αναλυτικά η περιστροφική παρουσίαση ενός μηνύματος ακριβώς 16 χαρακτήρων. Η περιστροφή θα λειτουργεί είτε με το πάτημα ενός κουμπιού είτε μετά από ένα χρονικό διάστημα, κάνοντας ολίσθηση προς τα δεξιά σε κάθε περίπτωση. Η εργασία θεωρήθηκε επιτυχημένη αφού όλα τα μέρη που την αποτελούν όπως και οι στόχοι που αναφέραμε παραπάνω ολοκληρώθηκαν με επιτυχία.

# Μέρος Α – Υλοποίηση Αποκωδικοποιητή 7-τμημάτων

Το πρώτο μέρος της εργασίας είναι αρκετά απλό μιας και είναι η υλοποίηση ενός απλού αποκωδικοποιητή μέσα στο *module* ***LEDdecoder***.

Για την αντιστοιχία των τιμών εισόδων/εξόδων χρησιμοποιήθηκε ο παρακάτω πίνακας στον οποίο αντιστοιχίζεται κάθε τιμή ενός μονοψήφιου δεκαεξαδικού αριθμού με την αντίστοιχη εμφάνιση του στην οθόνη, για παράδειγμα όταν:



|  |  |
| --- | --- |
| char | LED |
| 0x0 | 0000001 |
| 0x1 | 1001111 |
| 0x2 | 0010010 |
| 0x3 | 0000110 |
| 0x4 | 1001100 |
| 0x5 | 0100100 |
| 0x6 | 0100000 |
| 0x7 | 0001111 |
| 0x8 | 0000000 |
| 0x9 | 0000100 |
| 0xA | 0001000 |
| 0xB | 1100000 |
| 0xC | 0110001 |
| 0xD | 1000010 |
| 0xE | 0110000 |
| 0xF | 0111000 |

|  |
| --- |
| initial begin          char = 4'h0;      #10 char = 4'h1;      #10 char = 4'h2;      #10 char = 4'h3;      #10 char = 4'h4;      #10 char = 4'h5;      #10 char = 4'h6;      #10 char = 4'h7;      #10 char = 4'h8;      #10 char = 4'h9;      #10 char = 4'ha;      #10 char = 4'hb;      #10 char = 4'hc;      #10 char = 4'hd;      #10 char = 4'he;      #10 char = 4'hf;      #10 $finish;  end |

*Σημείωση:* παρόλο που η εκφώνηση μιλάει για οδήγηση του κάθε τμήματος της οθόνης στο 1 ισχύει το αντίθετο, έτσι χρειάστηκε να βρω τους αντίστροφους του τιμών του LED σε κάθε περίπτωση.

## Επαλήθευση

Το πλαίσιο δοκιμών του πρώτου μέρους είναι αρκετά απλό γιατί το κύκλωμα μας είναι συνδυαστικό οπότε δεν χρειάζονται σήματα όπως *ρολόι* και *reset* .

Τα διανύσματα που χρησιμοποιήθηκαν είναι πρακτικά όλοι οι συνδυασμοί ενός *4-bit* αριθμού, δηλαδή *από 0 έως 16 (δεκαδικό).* Το *testbench* θα μπορούσαμε να χρησιμοποιήσουμε μία ***for-loop*** αλλά τα διανύσματα προς δοκιμή θεωρήθηκαν λίγα ώστε να είναι εύκολη η γραφή με τον τρόπο που φαίνεται στο πλαίσιο στα δεξιά (*χειροκίνητα*).

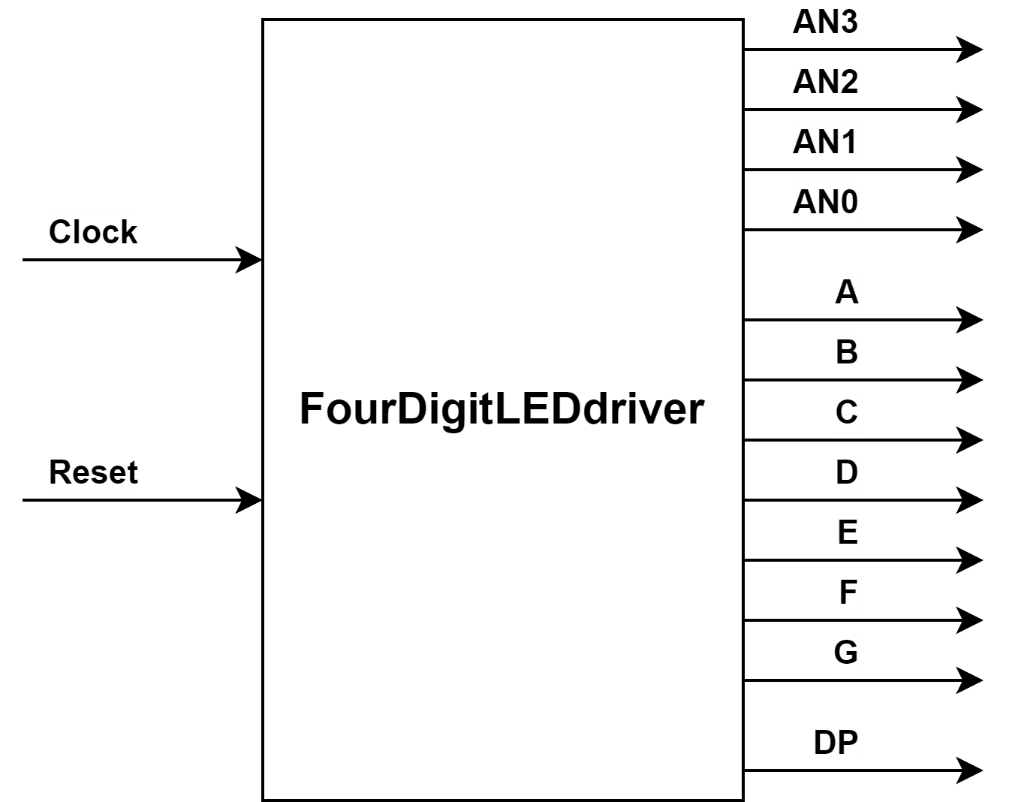
# Μέρος Β – Οδήγηση Τεσσάρων Ψηφίων

Το ζητούμενο στο δεύτερο μέρος της εργασίας είναι η οδήγηση και των τεσσάρων ψηφίων που αποτελούν την οθόνη μας.

Για την οδήγηση του κάθε ψηφίου χρησιμοποιούνται 4 είσοδοι (***AN3, AN2, AN1, AN0***) μέσω των οποίων επιλέγεται ένα ψηφίο την φορά (*που οδηγείται στο* ***0*)**. Για παράδειγμα όταν:

## Μονάδα *FourDigitLEDdriver*

Για *top-level* μονάδα έχουμε το ***FourDigitLEDdriver***:



## Μονάδα *MMCME2*

Για να πετύχουμε ένα πιο αργό clock όπως ζητάει η εκφώνηση με την χρήση του ***MMCME2*** *block* χρειάζεται να επιλέξουμε κάποιες παραμέτρους που θα μας μετασχηματίζουν το ρολόι της *FPGA (@100MHz)* σε ένα άλλο με περίοδο .20 μs *(ή* ***5MHz****).*

Σαν πρώτο βήμα παίρνουμε το έτοιμο template του *MMCME2\_base* που μας παρέχει το Vivado και κάνουμε της παρακάτω αλλαγές:

* Περνάμε στην παράμετρο *CLKIN1\_PERIOD* την περίοδο του ρολογιού εισόδου *(δηλαδή των 100MHz)* που είναι στα 10ns, δηλαδή τιμή ***10.0*** .
* Περνάμε στην παράμετρο *CLKFBOUT\_MULT\_F* την τιμή **6.0** καθώς το VCO πρέπει να έχει εύρος τιμών μεταξύ *600MHz~1.600MHz.*
* Βάζουμε ως διαιρετή *CLKOUT1\_DIVIDE* την τιμή **120**

Έτσι στην πράξη:

*Σημείωση:* για την σωστή λειτουργία του MMCME2 χρειάζεται ένα wire να που κάνει feedback την μονάδα διαχείρισης του ρολογιού, για χάρη του οποίου ορίστηκε το clkfb.

## Μονάδα *digit\_driver\_module*

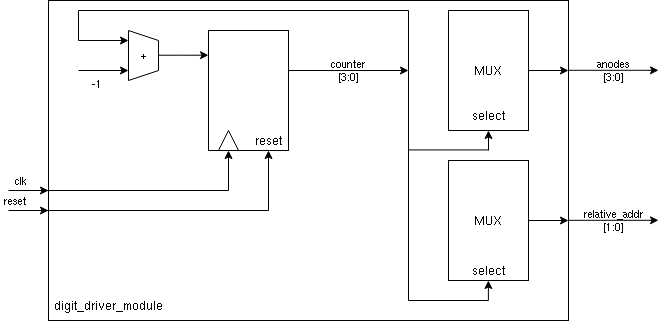
Αυτή είναι η μονάδα που οδηγεί τα 4 διαφορετικά ψηφία της οθόνης της FPGA. Έχει εισόδους το ρολόι clk και το ασύγχρονο reset, ενώ για εξόδους έχει διαύλους το anodes, για να επιλέγουμε το ψηφίο που θα γράψουμε, και το relative\_addr το οποίο επιλέγει τα δεδομένα που θα γραφτούν σε κάθε ψηφίο.



Ο μηχανισμός στον οποίον βασίζεται το relative\_addr είναι σχετικός με τα επόμενα μέρη της εργασίας όπου έχουμε το μήνυμα αποθηκευμένο σε ένα κομμάτι μνήμης. Παρόλα αυτά για να διατηρήσουμε ένα modularity χρησιμοποιούμε το ίδιο module και εδώ. Πρακτικά αυτή η έξοδος είναι ένας μετρητής που για το πρώτο ψηφίο δείχνει στην *θέση 0* της μνήμης, για το δεύτερο στην *θέση 1* της μνήμης και ου το καθεξής.

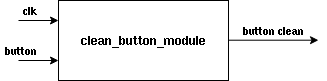
Πιο αναλυτικά:

* Μια ***always*** ακολουθιακής λογικής με *λίστα ευαισθησίας* στην *θετική ακμή του clk και reset* για τον μετρητή *counter*
* Μια ***always*** συνδυαστικής λογικής με *λίστα ευαισθησίας* τον *counter* για να μεταβάλλονται οι τιμές των εξόδων.



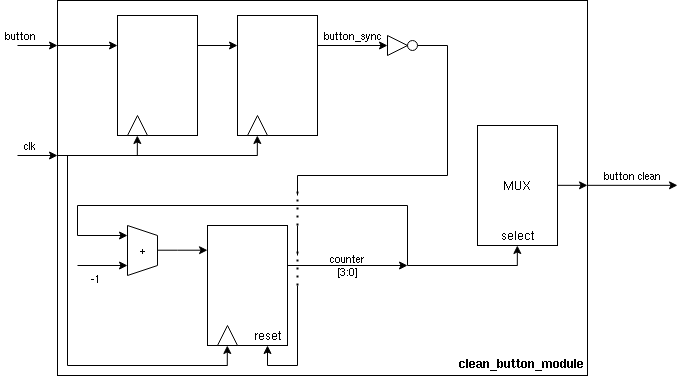
## Μονάδα *clean\_button\_module*

Η μονάδα αυτή είναι για την αποφυγή της εισόδου εξωτερικού σήματος κοντά σε κρίσιμο χρόνο (*όπως στο set-up time*) και για εφαρμογή συστήματος ***anti-bounce***, όπως το *reset* και το *button* για να ολισθαίνει το μήνυμα *(Μέρος Γ)*.



Πιο αναλυτικά:

* Δύο σε σειρά ***always*** *blocks* ακολουθιακής λογικής για τον κρίσιμο χρόνο που παίρνουν το εξωτερικό σήμα και το *“περνάνε”* μέσα στο κύκλωμα μετά το *hold* και πριν το set-up time.
* Ένα ***always*** *block* ακολουθιακής λογικής που *“περνάει”* το σήμα στο υπόλοιπο κύκλωμα μόνο όταν αυτό διατηρήσει την τιμή του *θετική* για ένα προκαθορισμένο χρονικό διάστημα.



*Σημείωση:* Όλα τα ***always*** της μονάδας έχουν *λίστα ευαισθησίας* την θετική ακμή του ρολογιού.

## Επαλήθευση

# Μέρος Γ — Βηματική Περιστροφή Μηνύματος με Χρήση Κουμπιού

# Μέρος Δ — Βηματική Περιστροφή Μηνύματος με Σταθερή Καθυστέρηση