

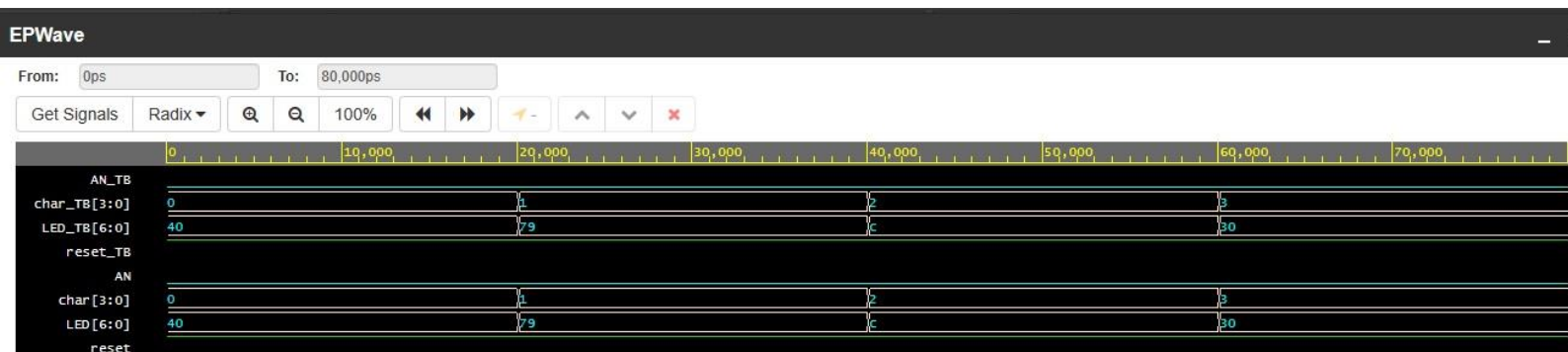
# Ψηφιακά Συστήματα HW I

ΕΡΓΑΣΙΑ ΕΞΑΜΗΝΟΥ

Νίκος Χαλαλάρης | Ομάδα 21 | AEM 9740

## Α' μέρος:

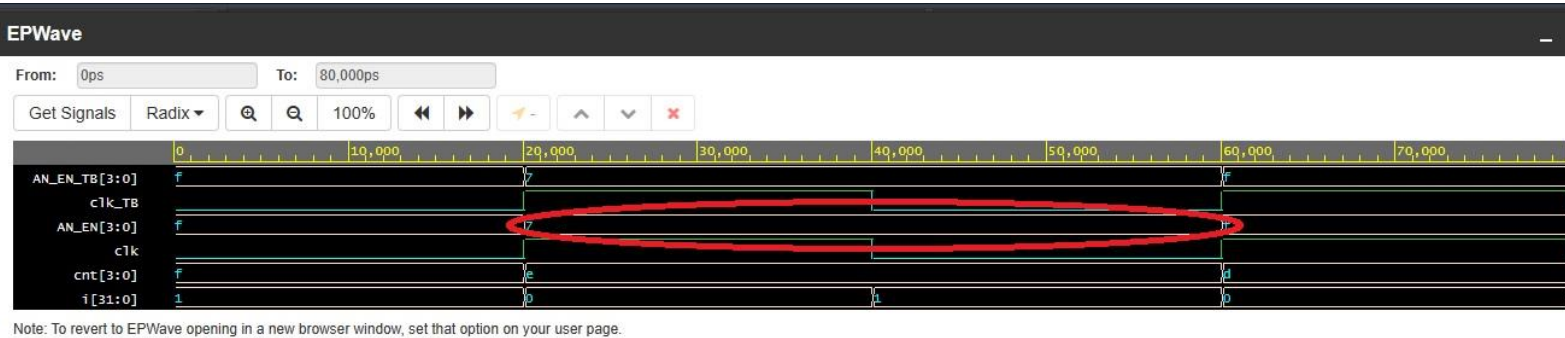
Το μέρος αυτό πραγματεύεται την δημιουργία ενός LED αποκωδικοποιητή 4 σε 7 με βάση το πρότυπο εμφάνισης ψηφίων στις οδηγίες της εργασίας. Έτσι θεωρήσαμε πως οι θέσεις A,B,C,D,E,F,G κάθε LED ψηφίου αντιστοιχούν στις θέσεις 0,1,2,3,4,5,6,7 του αριθμού 8-bit που παράγεται με το 7 -> MSB και το 0 -> LSB. Το δοκιμαστικό testbench διάγραμμα σημάτων έχει την ακόλουθη μορφή :



Note: To revert to EPWave opening in a new browser window, set that option on your user page.

Όπως παρατηρούμε, για μηδενική επίτρεψη ανόδου *AN* και για ΜΗ μηδενικό *reset* επιτυγχάνεται η αποκωδικοποίηση του εισαγόμενου διανύσματος χαρακτήρα *char[3:0]*. Σε αυτό το σημείο πρέπει να τονιστεί ότι η υλοποίηση του αποκωδικοποιητή και του απαριθμητή πραγματοποιήθηκαν ξεχωριστά ως μονάδες, χωρίς να συνδυαστούν για να παράξουν την οθόνη 4 LED φωτών, καθώς αγνοούσα με ποιον τρόπο είναι δυνατός ο συνδυασμός δύο διαφορετικών module και υπήρχε ζήτημα με τα διαφορετικά αρχεία στο EDA Playground. Ο μεταγλωττιστής δημιούργησε ένα σχηματικό \*.svg αρχείο για τον αποκωδικοποιητή και ένα για τον απαριθμητή. Και τα δύο

παράτιθενται συνημμένα στο συμπιεσμένο αρχείο που έχει ανέβει. Το δοκιμαστικό testbench σημάτων για τον απαριθμητή (counter) έχει την ακόλουθη μορφή :



Παρατηρούμε ότι για συχνότητα ρολογιού 20 ns, μετατρέπει την είσοδο του ρολογιού και την ακολουθία αριθμών για την παραγωγή επίτρεψης ανόδου για τη 2πλάσια χρονική διάρκεια. Εμείς θέλουμε την 16πλάσια, ωστόσο για να φανεί καλύτερα στο παραπάνω διάγραμμα πειράξαμε το ακόλουθο σημείο του κώδικα:

EDA playground

Run

Save

Copy

Testbench + Design

SystemVerilog/Verilog

UVM / OVM

None

Other Libraries

None

Enable TL-Verilog

Enable Easier UVM

Enable VUnit

Tools & Simulators

Icarus Verilog 0.9.7

Compile Options

-Wall

Run Options

Run Options

Open EPWave after run

Download files after run

Examples

Community

Collaborate

Forum

testbench.sv

```

1 `timescale 1ns/1ps
2 module Counter_TB();
3   reg clk_TB;
4   wire [3:0] AN_EN_TB;
5
6   Counter Counter_TB (AN_EN_TB, clk_TB);
7
8   initial begin
9     $dumpfile("dump.vcd"); $dumpvars;
10    clk_TB = 0;
11    #20 clk_TB=1; $display("AN_EN = %b", AN_EN_TB);
12    #20 clk_TB=0; $display("AN_EN = %b", AN_EN_TB);
13    #20 clk_TB=1; $display("AN_EN = %b", AN_EN_TB);
14    #20 clk_TB=0; $display("AN_EN = %b", AN_EN_TB);
15  end
16 endmodule

```

design.sv

```

4 integer i = 0;
5 reg [3:0] cnt=4'b1111;
6
7 always @(clk)
8 begin
9   i = i + 1;
10  if (i >= 16)
11  begin
12    i = 0;
13    cnt = cnt - 1;
14  end
15  case(cnt)
16    4'b1111 : AN_EN[3:0] = 4'b1111;
17    4'b1110 : AN_EN[3:0] = 4'b0111;
18    4'b1101 : AN_EN[3:0] = 4'b1111;
19    4'b1100 : AN_EN[3:0] = 4'b1111;
20    4'b1011 : AN_EN[3:0] = 4'b1111;
21    4'b1010 : AN_EN[3:0] = 4'b1011;
22    4'b1001 : AN_EN[3:0] = 4'b1111;
23  endcase

```

Log

Share

[2023-01-30 16:01:27 EST] iverilog '-Wall' design.sv testbench.sv && unbuffer vvp a.out

VCD info: dumpfile dump.vcd opened for output.

AN\_EN = 1111

AN\_EN = 0111

AN\_EN = 0111

AN\_EN = 1111

Finding VCD file...

./dump.vcd

[2023-01-30 16:01:27 EST] Opening EPWave...

Done

Χρησιμοποιώντας έναν μετρητή, μόλις οι χτύποι του ρολογιού φτάσουν την 16<sup>η</sup> περίοδο του, εναλλάσσεται η κωδικοποιημένη

αλληλουχία αριθμών με αυτό να προσφέρει την επιθυμητή καθυστέρηση στην έξοδο.

## **Β' Μέρος:**

Για τη δημιουργία του ελεγκτή Baud Controller, το σκεπτικό είναι παρόμοιο με εκείνο του απαριθμητή. Οι κυματομορφές του δοκιμαστικού testbench φαίνονται παρακάτω :



Παρατηρούμε ότι για περίοδο ρολογιού 20 ns, μηδενικό *reset* , με είσοδο *baud\_select[3:0]* αυτή της πιο γρήγορης δειγματοληψίας, ενεργοποιείται το σήμα *sample\_ENABLE* κάθε δειγματοληπτική περίοδο, παραμένει στο 1 για ένα χτύπο ρολογιού κι απενεργοποιείται για την επόμενη δειγματοληπτική περίοδο. Όπως είπαμε το σκεπτικό είναι παρόμοιο. Αποθηκεύσαμε στη μεταβλητή *sample[31:0]* τον αριθμό των περιόδων που υποδηλώνει η κάθε συχνότητα δειγματοληψίας και δημιουργήσαμε έναν counter που μετράει τις περιόδους που περνάνε μέχρι την ολοκλήρωση του απαιτούμενου αριθμού.

■ ■ ■