Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Πανεπιστήμιο Θεσσαλίας

ECE333 - Εργαστήριο Ψηφιακών Συστημάτων

Χειμερινό Εξάμηνο — Ακαδημαϊκό Έτος 2022-2023

Εργαστηριακή Εργασία 1^{η}

Οδηγός Ένδειξης 7-τμημάτων

7-Segment Display Driver

18/10/2022 έως 1/11/2022

Χ. Σωτηρίου

Περιεχόμενα

1 Στόχος της 1ης Εργασίας

Ο στόχος της $1^{ης}$ εργαστηριακής εργασίας είναι η υλοποίηση ενός οδηγού των τεσσάρων ενδείξεων 7-τμημάτων LED της πλακέτας **Nexys A7-100T**, για να επιτευχθεί η περιστροφική παρουσίαση ενός μηνύματος μεγέθους 16 χαρακτήρων. Το προτεινόμενο (αλλά βαρετό) μήνυμα είναι οι 16 χαρακτήρες των ψηφίων του δεκαεξαδικού συστήματος, δηλ. `0123456789abcdeF`, αλλά μπορείτε, αν θέλετε, να υλοποιήσετε ένα εναλλακτικό μήνυμα της αρεσκείας σας, κάνοντας τις κατάλληλες τροποποιήσεις στις παρακάτω οδηγίες.

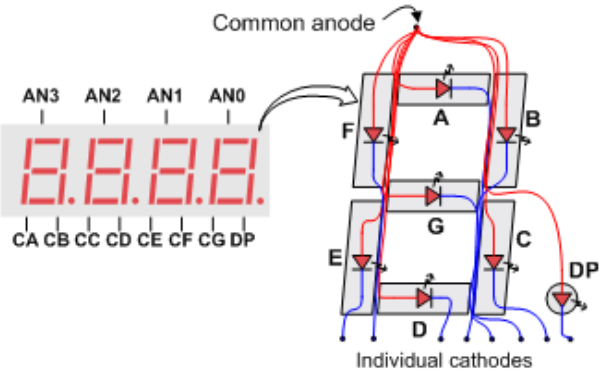
Το μήνυμα θα παρουσιαστεί μετατοπίζοντας τους χαρακτήρες, έναν προς έναν, κατάλληλα προς τα αριστερά, είτε:

- (1) με το πάτημα ενός κουμπιού, είτε
- (2) μετά από ένα δεδομένο χρονικό διάστημα.

Μετά τον τελευταίο χαρακτήρα του μηνύματος, θα ακολουθεί ο πρώτος, έτσι το μήνυμα ουσιαστικά θα περιστρέφεται διαρκώς.

2 Οι Τέσσερις Ενδείξεις 7 Τμημάτων

Στη σελίδα 23 του τεχνικού δελτίου της πλακέτας Nexys A7, παρουσιάζεται αναλυτικά η συνδεσμολογία των τεσσάρων ενδείξεων 7-τμημάτων στην πλακέτα. Τα τέσσερα ψηφία μοιράζονται τα 7 συν 1 (δεκαδικό σημείο) σήματα `A`, `B`, `C`, `D`, `E`, `F`, `G` και `DP`, όπως φαίνεται στο Σχήμα 1.

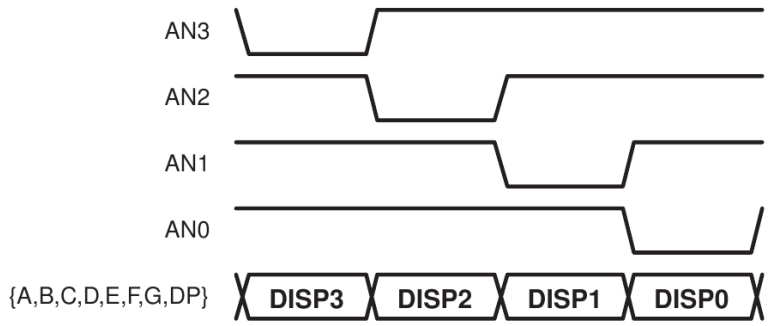


Σχήμα 1: Δομή Τεσσάρων Ψηφίων των 7-Ενδείξεων LED

Για να ανάψει ένα συγκεκριμένο τμήμα LED ενός ψηφίου πρέπει:

- το σχετικό σήμα ανόδου, `AN0-3`, που επιλέγει το ψηφίο, να οδηγείται στο **μηδέν (0)**,
- το σχετικό σήμα καθόδου τμήματος, `CA-G/DP`, που επιλέγει το τμήμα, να οδηγείται στο **ένα (1)**.

Στο Σχήμα 2 φαίνεται η κατάλληλη πολύπλεξη στον χρόνο που απαιτείται για να εμφανιστούν δεδομένα και στα τέσσερα ψηφία. Οι τέσσερις άνοδοι πρέπει να οδηγούνται εναλλάξ στο μηδέν, ενώ αλλάζουν κατάλληλα τα σήματα των 7 συν 1 ενδείξεων. Η διαδικασία πρέπει να επαναλαμβάνεται διαρκώς. Παρότι που τα LED ουσιαστικά αναβοσβήνουν, λόγω



Σχήμα 2: Χρονοδιάγραμμα Οδήγησης Ψηφίων

της γρήγορης εναλλαγής και της διατήρησης της φωτεινότητας τους, στο ανθρώπινο μάτι φαντάζουν σταθερά.

Όταν μία άνοδος, `ANx`, επιστρέφει στο ένα, οι τιμές των ενδείξεων δε διατηρούνται, και καθώς το κύκλωμα των `LED` εκφορτίζεται, τα φωτισμένα τμήματα σβήνουν. Κατά την εναλλάξ διαδικασία σάρωσης, η άνοδος και τα σήματα των τμημάτων πρέπει να μένουν σταθερά για ένα σχετικά μεγάλο χρονικό διάστημα, ως προς το ρολόι της πλακέτας, λόγω της μεγάλης χωρητικότητας του κυκλώματος των `LED`. Έτσι, τα σήματα θα πρέπει να μείνουν σταθερά για τουλάχιστον `1ms`.

Ωστόσο, ως χρόνος εναλλαγής των ανόδων προτείνονται τα `0.20μs`, τα οποία είναι πολλαπλάσια του ρολογιού των `100MHz` της FPGA. Παρακάτω παρουσιάζονται τα μέρη υλοποίησης της εργασίας. **Τα Μέρη Β [4], Γ [5] και Δ [6] απαιτούν και δοκιμή προγραμματισμού της συσκευής**.

3 Μέρος Α - Υλοποίηση Αποκωδικοποιητή 7-τμημάτων

Για την αποκωδικοποίηση των 7 συν 1 τμημάτων υλοποιήστε ένα συνδυαστικό αποκωδικοποιητή, ο οποίος θα έχει ως είσοδο το ψηφίο ή χαρακτήρα που θέλετε να εμφανίσετε, και θα παράγει ως έξοδο τις τιμές οδήγησης των 7 συν 1 τμημάτων. Για την αναπαράσταση των 16 ψηφίων απαιτείται οι χρήση αριθμών 4-bit, ενώ τα 7 τμήματα απαιτούν 7 ξεχωριστά bit (το δεκαδικό ψηφίο, `DP`, μπορεί να μένει σβηστό, οδηγώντας το μόνιμα στο 0).

Έτσι, μια μορφή του αποκωδικοποιητή 7-τμημάτων είναι η εξής (Σχήμα 3):

```
module LEDdecoder(char, LED);

  input [3:0] char;
  output [6:0] LED;

  ...

endmodule
```

Σχήμα 3: Μια Μορφή Υλοποίησης του Αποκωδικοποιητή

Συμπληρώστε την υλοποίηση του αποκωδικοποιητή και ελέγξτε λειτουργικά, μέσω προσομοίωσης, ότι οι τιμές που προκύπτουν στα ψηφία του `LED` είναι οι κατάλληλες, και ότι έχουν τη σωστή πολικότητα για τα αντίστοιχα σήματα καθόδου *(0 = σβηστό και 1 = αναμμένο)*. Για ευκολία στην επαλήθευση, αλλά και για την οδήγηση τους στην πλακέτα,

διαχωρίστε τα ψηφία του αποτελέσματος LED στις ξεχωριστές εξόδους των τμημάτων CA έως CG.

Όταν η κυκλωματική υλοποίηση είναι ορθή, επιδείξτε τον κώδικα Verilog που γράψατε για το κύκλωμα (RTL Design) και το πλαίσιο δοκιμής (Testbench) και τα αποτελέσματα της προσομοίωσης σε επιτηρητή του εργαστηρίου.

4 Μέρος Β - Οδήγηση Τεσσάρων Ψηφίων

Όπως εξηγήθηκε νωρίτερα, στην Ενότητα 2, για τη χρήση των τεσσάρων ψηφίων απαιτείται η εναλλάξ οδήγηση του κάθε ψηφίου με προτεινόμενη ελάχιστη καθυστέρηση φόρτισης τα 0.20μs. Ο ευκολότερος τρόπος οδήγησης των ανόδων στην ταχύτητα των 0.20μs είναι να πολλαπλασιαστεί η περίοδος του ρολογιού της FPGA (περιόδου 10ns) κατά είκοσι φορές, και κατόπιν η οδήγηση των σημάτων να είναι ορισμένη στη Verilog ανά κύκλο, βασισμένη στο νέο αργότερο ρολόι. Εναλλακτικά, θα έπρεπε να χρησιμοποιηθεί μετρητής ο οποίος, κάθε είκοσι λ.χ. κύκλους του ρολογιού των 10ns, θα σηματοδοτούσε την αλλαγή των σημάτων των ανόδων και των τμημάτων.

4.1 Δομή της Μονάδας

Για τον πολλαπλασιασμό της περιόδου (ή ισοδύναμα διαίρεση της συχνότητας του ρολογιού), σας προτείνεται να χρησιμοποιήσετε μια δομική μονάδα MMCM (Mixed-Mode Clock Manager). Η δομική αυτή μονάδα θα πρέπει να βρίσκεται εσωτερικά στην κύρια μονάδα του οδηγού που θα υλοποιήσετε, μαζί με την εμφάνιση του LEDdecoder, που υλοποιήσατε στο πρώτο μέρος, όπως φαίνεται στο Σχήμα 4 παρακάτω:

```
module FourDigitLEDdriver(reset, clk, an3, an2, an1, an0, \
a, b, c, d, e, f, g, dp);

  input clk, reset;
  output an3, an2, an1, an0;
  output a, b, c, d, e, f, g, dp;

  ...

  MMCME2_BASE #(
    ...
  )
  MMCME2_BASE_inst (
    ...
  );

  ...

  LEDdecoder LEDdecoder_inst (...);

  ...

endmodule
```

Σχήμα 4: Μια Μορφή Υλοποίησης του Οδηγού

Για να εμφανίσετε το πρότυπο της μονάδας MMCM, το οποίο θα πρέπει και να παραμετροποιήσετε και να συνδέσετε κατάλληλα, ακολουθείστε τα εξής βήματα στο μενού περιήγησης:

1. Επιλέξτε το μενού Tool→Language Templates
2. Επιλέξτε απο τα πρότυπα Verilog→Device Primitive Instantiation→ →Artix-7→Clock Components→MMCM/PLL→Base Mixed Mode Clock Manager.
3. Αντιγράψτε το πρότυπο, ορίστε τις παραμέτρους πολλαπλασιασμού και διαίρεσης, οδηγήστε το CLKIN1 και χρησιμοποιείστε το CLKOUT1 σαν το νέο ρολόι.

4.2 Οδήγηση των Ανόδων

Έχοντας διαιρέσει το ρολόι, θα πρέπει να οδηγήσετε τα σήματα των ανόδων όπως στο Σχήμα 2. Τα σήματα των ανόδων δε θα πρέπει να επικαλύπτονται όσο βρίσκονται στο μηδέν, αλλιώς θα αλλοιωθούν οι ενδείξεις. Έτσι, απαιτείται και ένα περιθώριο ασφαλείας από την επιστροφή ενός σήματος ανόδου στο λογικό ένα, μέχρι την πτώση του επόμενου.

Η μη-επικαλυπτώμενη οδήγηση των ανόδων μπορεί να επιτευχθεί με τη χρήση ενός περιστροφικού μετρητή, όπου κάποιες τιμές του μετρητή θα ενεργοποιούν (στο μηδέν — logic-0) εναλλάξ τις ανόδους, ενώ οι ενδιάμεσες θα τις επιστρέφουν στο λογικό ένα (logic-1). Παραδείγματος χάριν, ένας 4-bit μετρητής θα μπορούσε να ενεργοποιεί τις ανόδους όπως φαίνεται στον παρακάτω πίνακα, αρχικοποιημένος στην τιμή 1111.

Τιμή μετρητή	AN3	AN2	AN1	AN0
1111	1	1	1	1
1110	**0**	1	1	1
1101	1	1	1	1
1100	1	1	1	1
1011	1	1	1	1
1010	1	**0**	1	1
1001	1	1	1	1
1000	1	1	1	1
0111	1	1	1	1
0110	1	1	**0**	1
0101	1	1	1	1
0100	1	1	1	1
0011	1	1	1	1
0010	1	1	1	**0**
0001	1	1	1	1
0000	1	1	1	1

Πρακτικά, ο μετρητής χρησιμοποιείται εμμέσως ως ΜΠΚ (Μηχανή Πεπερασμένων Καταστάσεων), όπου η μέτρηση του αντιστοιχεί πρακτικά σε μια κατάσταση.

Συμπληρώστε την υλοποίηση της οδήγησης των ανόδων, είτε βάση της προτεινόμενης υλοποίησης, είτε με κάποια εναλλακτική, οδηγώντας τις τιμές των ενδείξεων σταθερά, λ.χ. 0123. Για την οδήγηση των σημάτων των ενδείξεων χρησιμοποιείστε πάλι τιμές του μετρητή, έτσι ώστε να υπάρχει χρόνος προετοιμασίας (πρόθεσης - setup) μεταξύ των ανόδων και των δεδομένων των ενδείξεων. Δώστε περιθώριο αλλαγής τουλάχιστον ενός βήματος. Λ.χ. για την οδήγηση της AN3 στο μηδέν στο 1110, αλλάξτε τα δεδομένα στη μέτρηση 0000, για την οδήγηση της AN2 στο μηδέν στο 1010, αλλάξτε τα δεδομένα στο 1100, κ.ο.κ.. Ο λόγος που απαιτείται και αυτό το περιθώριο είναι η μεγάλη χωρητικότητα των ενδείξεων στην πλακέτα.

4.3 Αρχικοποίηση

Για την αρχικοποίηση του κυκλώματος χρησιμοποιείστε ένα **ασύγχρονο** σήμα αρχικοποίησης `reset` σε κάθε ακολουθιακό τμήμα `always`. Το σήμα αρχικοποίησης θα πρέπει να οδηγείται από το κουμπί `BTNC` (User Reset) της συσκευής, το οποίο και αντιστοιχεί στην ακίδα `N17` της FPGA. Προαιρετικά, χρησιμοποιείστε κύκλωμα αποφυγής αναπηδήσεων (anti-bounce) για το σήμα του κουμπιού.

4.4 Περιορισμοί `XDC` (`Xilinx Design Constraints`)

Στο παρακάτω Σχήμα 5, σας παραθέτεται το σχετικό `XDC` (`Xilinx Design Constraints`)που πρέπει να χρησιμοποιήσετε, το οποίο περιέχει την αντιστοίχηση των σημάτων στις σχετικές ακίδες της συσκευής:

```
## This file is a general .xdc for the Nexys A7-100T

### Clock Signal
set_property -dict { PACKAGE_PIN E3    IOSTANDARD LVCMOS33 } [get_ports { clk }];
create_clock -add -name sys_clk -period 10.00 -waveform {0 5} [get_ports { clk }];

### 7-Segment Display
set_property -dict { PACKAGE_PIN T10   IOSTANDARD LVCMOS33 } [get_ports { a }];
set_property -dict { PACKAGE_PIN R10   IOSTANDARD LVCMOS33 } [get_ports { b }];
set_property -dict { PACKAGE_PIN K16   IOSTANDARD LVCMOS33 } [get_ports { c }];
set_property -dict { PACKAGE_PIN K13   IOSTANDARD LVCMOS33 } [get_ports { d }];
set_property -dict { PACKAGE_PIN P15   IOSTANDARD LVCMOS33 } [get_ports { e }];
set_property -dict { PACKAGE_PIN T11   IOSTANDARD LVCMOS33 } [get_ports { f }];
set_property -dict { PACKAGE_PIN L18   IOSTANDARD LVCMOS33 } [get_ports { g }];
set_property -dict { PACKAGE_PIN H15   IOSTANDARD LVCMOS33 } [get_ports { dp }];
set_property -dict { PACKAGE_PIN J17   IOSTANDARD LVCMOS33 } [get_ports { an0 }];
set_property -dict { PACKAGE_PIN J18   IOSTANDARD LVCMOS33 } [get_ports { an1 }];
set_property -dict { PACKAGE_PIN T9    IOSTANDARD LVCMOS33 } [get_ports { an2 }];
set_property -dict { PACKAGE_PIN J14   IOSTANDARD LVCMOS33 } [get_ports { an3 }];

### Button(s)
set_property -dict { PACKAGE_PIN N17   IOSTANDARD LVCMOS33 } [get_ports { reset }];
```

Σχήμα 5: Μια Προτεινόμενη Μορφή των Περιορισμών της πλακέτας Nexys A7-100T.

4.5 Προχωρημένη Χρήση της `MMCM` Μονάδας *(Προαιρετικό)*

Προαιρετικά, προσπαθήστε να επιτύχετε ως ελάχιστη καθυστέρηση φόρτισης τα `25.0ns` με αποκλειστική χρήση της `MMCM` μονάδας. Για να ικανοποιηθεί η συγκεκριμένη καθυστέρηση θα πρέπει να προσαρμόσετε αναλόγως τις παραμέτρους της μονάδας σύμφωνα με τον αντίστοιχο οδηγό χρήσης των παροχών ρολογιού της πλακέτας.

4.6 Λειτουργικός Έλεγχος

Όταν ολοκληρώσετε την υλοποίηση του Μέρους Β, πραγματοποιείστε έλεγχο λειτουργίας, με προσομοίωση σε επίπεδο συμπεριφοράς, και κατόπιν σε επίπεδο πυλών, χρησιμοποιώντας ένα testbench που οδηγεί κατάλληλα το ρολόι. Όταν βεβαιωθείτε ότι η υλοποίηση

του κυκλώματος είναι ορθή, παράξτε το κατάλληλο bitfile, προγραμματίστε τη συσκευή και ελέγξτε ότι και στην πράξη, οι τέσσερις ενδείξεις οδηγούνται σωστά.

Όταν η κυκλωματική υλοποίηση είναι ορθή, επιδείξτε τον κώδικα Verilog που γράψατε για το κύκλωμα (RTL Design) και το πλαίσιο δοκιμής (Testbench), τα αποτελέσματα της προσομοίωσης, και το κύκλωμα εν λειτουργία σε επιτηρητή του εργαστηρίου.

5 Μέρος Γ - Βηματική Περιστροφή του Μηνύματος με χρήση Κουμπιού

Στο τρίτο μέρος θα υλοποιήσετε την περιστροφή του μηνύματος στο πάτημα ενός κουμπιού. Για αυτήν, θα χρειαστεί καταρχήν να αποθηκευτεί σε δομή μνήμης το μήνυμα. Η μνήμη μπορεί να φτιαχτεί από καταχωρητές, και για το εν λόγω μήνυμα απαιτούνται 16 στοιχείων των 8-bit. Επιπλέον, τα τέσσερα ψηφία προς αποκωδικοποίηση, θα πρέπει τώρα να είναι μεταβλητά και όχι σταθερά, ανάλογα με την ενεργή διεύθυνση της μνήμης, η οποία θα περιστρέφεται στο μηδέν. Αυτό μπορεί να επιτευχθεί, χρησιμοποιώντας έναν μετρητή - δείκτη στην ενεργή διεύθυνση του μηνύματος, από τον οποίο θα προέρχονται (διαβάζοντας τα μνήμη) τα τέσσερα ψηφία. Κατόπιν, θα αποκωδικοποιούνται όπως και στο δεύτερο μέρος.

Η μνήμη μπορεί να οριστεί ως εξής:

```
...
reg [3:0] message [0:15];
...
```

Σχήμα 6: Μια Μορφή Υλοποίησης Μνήμης Μηνύματος από Καταχωρητές

Ο μετρητής - δείκτης θα αυξάνεται στο πάτημα ενός κουμπιού της πλακέτας. Χρησιμοποιήστε το κουμπί `BTNR`, το οποίο αντιστοιχεί στην ακίδα `M17` της συσκευής, και υλοποιήστε κύκλωμα αποφυγής αναπηδήσεων (anti-bounce) για το σήμα του κουμπιού.

Όταν ολοκληρώσετε την υλοποίηση του Μέρους Γ, πραγματοποιείστε έλεγχο λειτουργίας, με προσομοίωση σε επίπεδο συμπεριφοράς, και κατόπιν σε επίπεδο πυλών, χρησιμοποιώντας ένα testbench που οδηγεί κατάλληλα το ρολόι και προσομοιώνει και τη συμπεριφορά του κουμπιού για ένα η δύο πατήματα. Όταν βεβαιωθείτε ότι η υλοποίηση του κυκλώματος είναι ορθή, παράξτε το κατάλληλο bitfile, προγραμματίστε τη συσκευή και ελέγξτε ότι οι τέσσερις ενδείξεις οδηγούνται σωστά.

Όταν η κυκλωματική υλοποίηση είναι ορθή, επιδείξτε τον κώδικα Verilog που γράψατε για το κύκλωμα (RTL Design) και το πλαίσιο δοκιμής (Testbench), τα αποτελέσματα της προσομοίωσης, και το κύκλωμα εν λειτουργία σε επιτηρητή του εργαστηρίου.

6 Μέρος Δ - Βηματική Περιστροφή του Μηνύματος με σταθερή Καθυστέρηση

Στο τέταρτο μέρος πρέπει να αντικαταστήσετε τη λειτουργία του κουμπιού με μια σταθερή καθυστέρηση. Έτσι, κάθε 1 δευτερόλεπτο, ή λίγο περισσότερο, θα πρέπει να περιστρέφεται

κατά ένα χαρακτήρα το μήνυμα. Η σταθερή καθυστέρηση μπορεί να υλοποιηθεί με έναν μετρητή κύκλων ρολογιού, ο μηδενισμός του οποίου θα επιδεικνύει το πέρας του χρόνου. Σας προτείνεται η υλοποίηση ενός μετρητή 23- bit, ο οποίος μετράει 1,6777214 δευτερόλεπτα.

Όταν ολοκληρώσετε την υλοποίηση του Μέρους Δ, πραγματοποιείστε έλεγχο λειτουργίας, με προσομοίωση σε επίπεδο συμπεριφοράς, και κατόπιν σε επίπεδο πυλών χρησιμοποιώντας ένα testbench (πλαίσιον προσομοίωσης) που οδηγεί κατάλληλα το ρολόι. Επειδή 1,6777214 δευτερόλεπτα είναι πολύ μεγάλος χρόνος προσομοίωσης, δοκιμάστε το κύκλωμα με μικρότερο μετρητή (λιγότερων ψηφίων), και κατόπιν αλλάξτε το πλάτος του. Όταν βεβαιωθείτε ότι η υλοποίηση του κυκλώματος είναι ορθή, παράξτε το κατάλληλο bitfile, προγραμματίστε τη συσκευή και ελέγξτε ότι οι τέσσερις ενδείξεις οδηγούνται σωστά.

Όταν η κυκλωματική υλοποίηση είναι ορθή, επιδείξτε τον κώδικα Verilog που γράψατε για το κύκλωμα (RTL Design) και το πλαίσιο δοκιμής (Testbench), τα αποτελέσματα της προσομοίωσης, και το κύκλωμα εν λειτουργία σε επιτηρητή του εργαστηρίου.

7 Προθεσμία Παράδοσης, Υποβολή της Εργασίας και Αναφορά

Η προθεσμία παράδοσης της $1^{ης}$ εργασίας είναι η **1/11/2022**.

Μέχρι την προθεσμία της εργασίας θα πρέπει:

- να έχετε επιδείξει όλα τα επιμέρους μέρη της εργασίας στους επιτηρητές,
- να έχετε υποβάλλει τον κώδικα σας (RTL Design και Testbench) για κάθε σκέλος υλοποίησης ξεχωριστά,
- να έχετε υποβάλλει μια εργαστηριακή αναφορά, βάση των σημειώσεων απο το βιβλίο του εργαστηρίου, **με όλη την απαραίτητη πληροφορία** (*Dataflow, FSM State Graph(s), Simulation Screenshot(s), etc.*).

8 Ερωτήσεις και Απορίες

Για οποιεσδήποτε ερωτήσεις και απορίες εκμεταλλευτείτε τον ιστιότοπο του e-Class και το χρόνο του εργαστηρίου ρωτόντας τους επιτηρητές.