HY220: Εργαστήριο Ψηφιακών Κυκλωμάτων

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης
Εαρινό Εξάμηνο 2024

Εργαστήριο 0

- **Εβδομάδες 19/03 έως 22/03 και 26/03 έως 29/03**

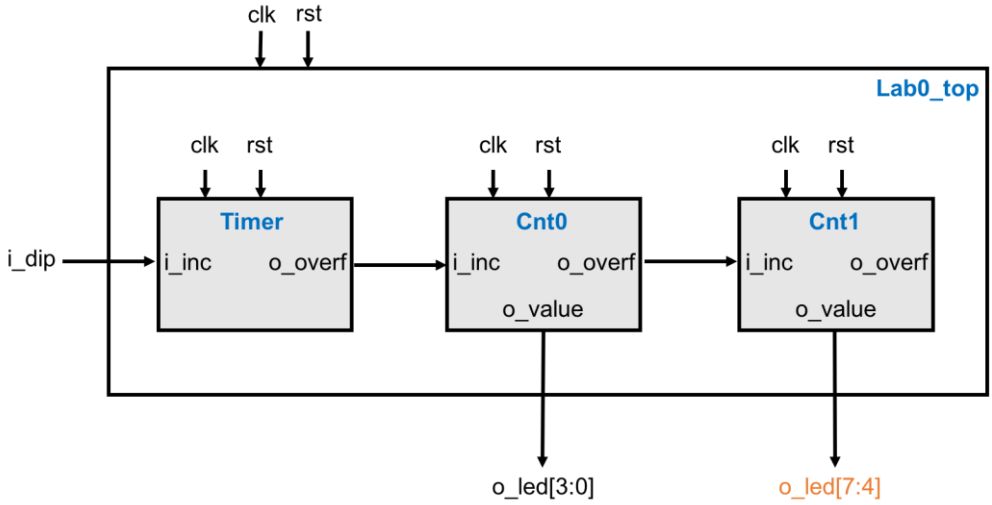
(Γκρουπ Α*)	Πέμπτη	21/03	16:00 – 17:00	στην αίθουσα Β.110
(Γκρουπ Β*)	Πέμπτη	21/03	17:00 – 18:00	στην αίθουσα Β.110
(Γκρουπ Γ)	Πέμπτη	28/03	16:00 – 17:00	στην αίθουσα Β.110
(Γκρουπ Δ)	Πέμπτη	28/03	17:00 – 18:00	στην αίθουσα Β.110
(Γκρουπ Ε)	Πέμπτη	28/03	18:00 – 19:00	στην αίθουσα Β.110

*** Κατ' εξαίρεση Πέμπτη λόγω των αργιών Καθ. Δευτέρας και 25ης Μαρτίου.**

Ο σκοπός του *Εργαστηρίου 0* είναι να εξοικειωθείτε με προσομοίωση SystemVerilog, την πλακέτα και τα εργαλεία προγραμματισμού της.

Το Εργαστήριο 0 δεν έχει παραδοτέα και δεν βαθμολογείται. Κάντε προετοιμασία πριν το εργαστήριο μελετώντας την εκφώνηση και τα αρχεία και δοκιμάστε την προσομοίωση.
Η παρουσία σας στο Lab 0 είναι υποχρεωτική!

Για το *Εργαστήριο 0* σας δίνεται έτοιμος κώδικας SystemVerilog και testbench που εμφανίζει δύο μετρητές 4-bit πάνω σε λαμπάκια (led). Θα πρέπει να ακολουθήσετε την ροή του εργαλείου Xilinx Vivado και τα βήματα που χρειάζεται για να κάνετε προσομοίωση, να «κατεβάσετε» το σχέδιο στην FPGA και να το δείτε να δουλεύει. Ο κώδικας SystemVerilog που σας δίνεται υλοποιεί το παρακάτω σχέδιο:



Το σχέδιο έχει την εξής ιεραρχία:

- **lab0_top** *(lab0_top.sv)*: περιέχει 3 instances (Timer, Cnt0, Cnt1) του μπλοκ **counter**, τα συνδέει μεταξύ τους και συνδέει τις εισόδους και τις εξόδους όπως φαίνεται στο σχήμα.
- **counter** *(counter.sv)*: είναι ένα απλό μπλοκ που υλοποιεί ένα παραμετρικό μετρητή N bits τον οποίο εμφανίζει στην έξοδο *o_value*. Το μπλοκ έχει την είσοδο *i_increase* και σε κάθε κύκλο που είναι ενεργοποιημένη η είσοδος αυξάνει την τιμή του μετρητή κατά ένα. Όταν ο μετρητής κάνει wrap-round τότε γεννάει στην έξοδο *o_overflow* ένα παλμό

ενός κύκλου ρολογιού. Το μπλοκ έχει επίσης την παράμετρο MAX που καθορίζει (at compile time) σε ποια τιμή θα κάνει wrap-around ο μετρητής.

- **lab0_tb** *(lab0_tb.sv)*: ένα απλό testbench *μόνο για προσομοίωση* που δημιουργεί το ρολόι και το reset.

Στο σχέδιο ο μετρητής Timer παίρνει την είσοδο *i_increase* από το σήμα *i_dip* το οποίο θα συνδεθεί σε ένα διακόπτη (DIP switch) πάνω στην πλακέτα. Όταν ο διακόπτης είναι ενεργοποιημένος τότε θα αυξάνεται η εσωτερική τιμή του Timer. Το ρολόι του κυκλώματος είναι 100MHz (περίοδος ρολογιού 10ns) οπότε στον κώδικα ο Timer έχει τις κατάλληλες παραμέτρους ώστε να κάνει wrap-around μία φορά ανά δευτερόλεπτο και άρα γεννάει το σήμα *o_overflow* μία φορά ανά δευτερόλεπτο. Με τη σύνδεση που σας δίνεται, ο μετρητής Cnt0 (4-bits) αυξάνει την τιμή του μία φορά ανά δευτερόλεπτο και όταν αυτός με τη σειρά του κάνει overflow (κάθε 16 δευτερόλεπτα) τότε θα αυξάνει ο μετρητής Cnt1. Η έξοδος *o_value* του μετρητή Cnt0 είναι συνδεδεμένη σε 4 εξωτερικά λαμπάκια led έτσι ώστε να βλέπετε την τιμή του (στο δυαδικό).

Τι πρέπει να κάνετε στο εργαστήριο με την καθοδήγηση του βοηθού:

1. Χρησιμοποιείστε το testbench που σας δίνεται για να τρέξετε προσομοίωση με τον simulator του Vivado. Βάλτε στις κυματομορφές (waveforms) τα όλα σήματα εισόδου και εξόδου από τα 3 counter instances (Timer, Cnt0, Cnt1) και παρατηρείστε τα σήματα *o_overflow*. Για την προσομοίωση η MAX τιμή του Timer είναι ορισμένη στο 10 έτσι ώστε η προσομοίωση να τελειώνει γρηγορότερα. Σημαδέψτε με marker τη στιγμή που ο Cnt1 βγάζει στην έξοδό του *o_value* την τιμή 5. Δείξτε τις κυματομορφές στο βοηθό.
    - Στα αρχεία του Lab0 σας δίνεται και ο φάκελος **run** με κατάλληλο **Makefile** για να κάνετε εύκολα προσομοίωση από τερματικό πριν το εργαστήριο.

2. Ακολουθήστε τα βήματα που σας παρουσιάστηκαν στο φροντιστήριο του Vivado και κάντε ανάθεση pins (pin assignment) μέσω του constraint file που σας δίνεται (lab0.xdc). Ολοκληρώσετε τα βήματα σύνθεσης (synthesis), υλοποίησης (implementation), και γέννησης αρχείου προγραμματισμού (bitstream) και «κατεβάστε» το bitstream του design στην πλακέτα. Δείξτε στο βοηθό ότι δουλεύει.

3. Συνδέστε την έξοδο του μετρητή Cnt1 στα υπόλοιπα λαμπάκια (o_led[7:4]) έτσι ώστε να βλέπετε και τους 2 μετρητές. Ακολουθείστε πάλι τα βήματα και «κατεβάστε» το νέο bitstream στην FPGA. Δείξτε στον βοηθό ότι δουλεύει.

4. Αλλάξτε την παράμετρο MAX των Cnt0 και Cnt1 έτσι ώστε ο μετρητές να κάνουν wrap-around μετά από 10 αυξήσεις. Ακολουθείστε πάλι τα βήματα και «κατεβάστε» το νέο bitstream στην FPGA. Δείξτε στον βοηθό ότι δουλεύει.

5. Αλλάξτε τις παραμέτρους του Timer έτσι ώστε οι μετρητές να αυξάνονται κάθε 5 δευτερόλεπτα. Ακολουθείστε πάλι τα βήματα και «κατεβάστε» το νέο bitstream στην FPGA. Δείξτε στον βοηθό ότι δουλεύει.