

# Ολοκληρωμένα Κυκλώματα

## Εργαστηριακή άσκηση: Υλοποίηση συστήματος σε FPGA με χρήση κυκλώματος διαιτησίας Round Robin

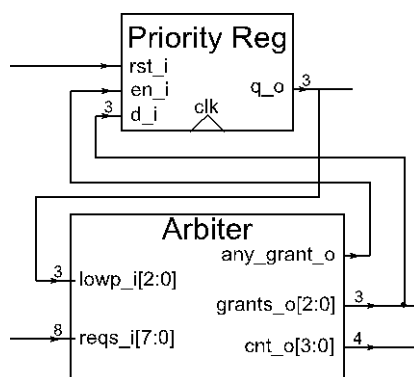
Δημήτρης Κωνσταντίνου, Γιώργος Δημητρακόπουλος

Στο προηγούμενο εργαστήριο περιγράψαμε εκτενώς την λειτουργία ενός κυκλώματος διαιτησίας Round Robin, και κληθήκατε να αναπτύξετε ένα testbench το οποίο είναι ικανό να εντοπίζει πιθανά σφάλματα του κυκλώματος που σας δόθηκε. Σκοπός του προηγούμενου εργαστηρίου ήταν η κατανόηση της λειτουργίας του κυκλώματος (specifications) και ο έλεγχος της λειτουργίας σύμφωνα με τους κανόνες, μίας άγνωστης υλοποίησης του κυκλώματος (Verification).

Σε αυτό το εργαστήριο, αρχικά, καλείστε να υλοποιήσετε το κύκλωμα της διαιτησίας Round Robin (RTL design), το οποίο εν τέλη θα είναι μέρος ενός συστήματος, όπως θα δούμε παρακάτω.

### Κύκλωμα Round Robin

Για την περιγραφή της λειτουργίας του arbiter (κύκλωμα διαιτησίας), ανατρέξτε στην εκφώνηση του προηγούμενου εργαστηρίου. Η υλοποίησή σας θα πρέπει να γίνει σε δύο μέρη όπως φαίνεται στο παρακάτω σχήμα.

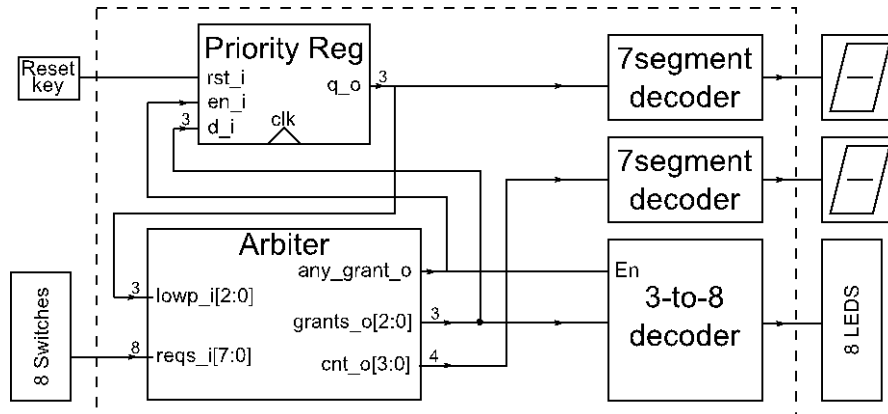


- 1) **Priority Register:** Στο module αυτό θα αποθηκεύεται η θέση των request με την **χαμηλότερη** προτεραιότητα. Δεν είναι τίποτα άλλο από έναν 3bit register ο οποίος αποθηκεύει το σήμα d\_i μόνο όταν το en\_i είναι '1', ενώ η αποθηκευμένη τιμή περνάει αυτούσια στην έξοδο και οδηγεί το κύκλωμα του arbiter. ΠΡΟΣΟΧΗ, η προτεραιότητα είναι σε **weighted binary** αναπαράσταση (π.χ. αν lowest priority =6 τότε q\_o = 'b110)
- 2) **Arbiter:** το module αυτό υλοποιεί την λογική διαιτησίας (βλέπε προηγούμενο εργαστήριο). Οι εισοδοί του module είναι η προτεραιότητα που ορίζει την σειρά του πιθανού νικητή (3bit weighted binary), και τα 8bit των request ώστε επιλεγεί το πολύ ένα. Αντίστοιχα, οι έξοδοι είναι τα any\_grant δηλώνοντας ότι υπάρχει νικητής, ο νικητής grants\_o σε **αναπαράσταση weighted binary των 3bit.**, και τέλος το cnt\_o το οποίο πάλι σε weighted binary αναπαράσταση υπολογίζει το πλήθος των request (π.χ. req\_i="00101111" -> cnt\_o = 5 = "0101")

Αφού υλοποιήσετε το παραπάνω κύκλωμα θα είστε σε θέση να χρησιμοποιήσετε το testbench του προηγούμενου εργαστηρίου ώστε να σιγουρευτείτε για την ορθή λειτουργία και να συνεχίσετε στο επόμενο κύκλωμα. (Καλό είναι να μην παρακάμψετε το simulation με την χρήση του modelsim)

## Τελικό κύκλωμα

Για την κατανόηση του κυκλώματος, αντί να έχουμε π.χ. τα interrupt σήματα μίας cpu από τα οποία θα έπρεπε διαλέγουμε ποιο θα εξυπηρετήσουμε, θα εισάγουμε τα requests με την βοήθεια των διακοπών της FPGA. Αντίστοιχα οι έξοδοι θα πρέπει να είναι εμφανής σε φυσικό επίπεδο, για το λόγο αυτό, το τελικό κύκλωμα μετατρέπει τις τιμές ώστε να είναι πιο εύκολα κατανοητές σε εμάς, διαμορφώνοντας το τελικό κύκλωμα που φαίνεται παρακάτω.



Για να μπορέσει να γίνει η προβολή των τιμών του κυκλώματος διαιτησίας στα περιφερειακά της FPGA, θα πρέπει να υλοποιηστέ δύο ειδών decoders. Έναν weighted **binary-to-7segment** ο οποίος θα παίρνει τιμές π.χ. 3 και θα ενεργοποιεί τα κατάλληλα σήματα ώστε να σχηματιστεί ο χαρακτήρας 3 του αριθμού σε οθόνη 7-segment. Η διάταξη των bit στα διάφορα led εμφανίζονται στην παρακάτω εικόνα. Περισσότερες πληροφορίες μπορείτε να βρείτε και στο manual της πλακέτας ([https://www.intel.com/content/dam/altera-www/global/en\\_US/portal/dsn/42/doc-us-dsnbk-42-4904342209-de1-usermanual.pdf](https://www.intel.com/content/dam/altera-www/global/en_US/portal/dsn/42/doc-us-dsnbk-42-4904342209-de1-usermanual.pdf)) (σελ. 28)

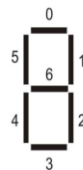


Figure 4.7. Position and index of each segment in a 7-segment display.

Ο άλλος decoder θα πρέπει να μετατρέπει weighted binary μορφή σε onehot αναπαράσταση ώστε να ανάβει μόνο ένα led, αυτό της αντίστοιχης θέσης (π.χ. 5= '101' → '00100000' ή 0 → '00000001'). Προσοχή, η έξοδος θα οδηγείται μόνο όταν υπάρχει any\_grant, αλλιώς όλα τα leds θα πρέπει να είναι απενεργοποιημένα.

## Για το εργαστήριο

Για το εργαστήριο θα πρέπει να υλοποιήσετε τα παραπάνω modules σε System Verilog. Το top level το οποίο θα πρέπει να περιέχει τον arbiter, το priority register, τους απαραίτητους decoders και φυσικά την απαραίτητη διασύνδεση. Όσους εισόδους θα πρέπει να υπάρχει το σήμα του ρολογιού και 8bit που θα προέρχονται από τους διακόπτες της fpga και έξοδοι θα είναι τα σήματα για δύο 7segment displays και τα 8bit τα οποία θα οδηγούν τα Leds. Καλό θα είναι να έχετε ετοιμάσει και το .qsf με την απαραίτητη αντιστοίχιση των pins.