

# HY220: Εργαστήριο Ψηφιακών Κυκλωμάτων

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης  
Χειμερινό Εξάμηνο 2013

## Εργαστήριο 4: Υλοποίηση Μηχανισμού Απεικόνισης «Διευθύνσεων»

11 Νοεμβρίου έως 15 Νοεμβρίου 2013 (5<sup>η</sup> εβδομάδα)

### 1.1 Σκοπός της Εργαστηριακής Άσκησης

Σε αυτήν την Άσκηση θα επιχειρήσουμε να δημιουργήσουμε κύκλωμα που να υλοποιεί ένα μηχανισμό απεικόνισης «διευθύνσεων» (8-bit). Το κύκλωμα διαχειρίζεται ένα αποθηκευτικό χώρο 8 θέσεων, μέσα στον οποίο φυλλάσει «διευθύνσεις» που δίδονται ως είσοδοι. Για κάθε θέση, φυλάσσει την πληροφορία εάν η θέση είναι σε χρήση. Το κύκλωμα επιτρέπει την αποδέσμευση θέσεων, και επίσης δίνει ένδειξη για το εάν υπάρχουν διαθέσιμες θέσεις. Με χρήση πλήκτρων, διακοπών και LEDs θα πρέπει να επιδείξετε την λειτουργία του συστήματος αυτού. Το παρακάτω σχήμα δείχνει το ζητούμενο κύκλωμα.

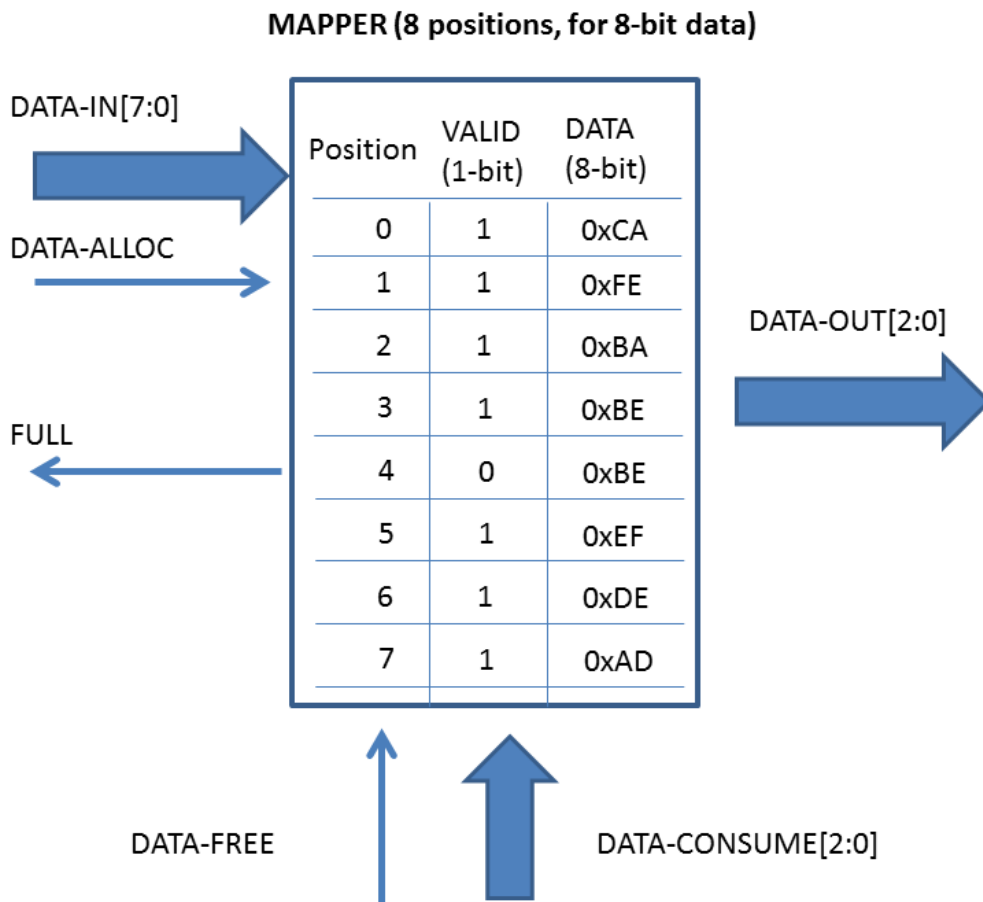


Figure 1: Σύστημα απεικόνισης «διευθύνσεων» (8-bit).

## 1.2 Σχεδιασμός Συστήματος

Πρώτο βήμα για την υλοποίηση της εργασίας είναι ο σχεδιασμός, στο χαρτί ή και σε ηλεκτρονική μορφή, το κύκλωμα που υλοποιεί την λογική ελέγχου, σύμφωνα με τους παρακάτω κανόνες.

- Το σύστημα διαθέτει 8 θέσεις αποθήκευσης για «διευθύνσεις» πλάτους 8 bits (πεδίο DATA στο παραπάνω σχήμα), για καθεμία από τις οποίες υπάρχει ένδειξη για το εάν είναι σε χρήση ή όχι (πεδίο VALID στο παραπάνω σχήμα).
- Ο χρήστης εισάγει μια «διεύθυνση» προς αποθήκευση μέσω των εισόδων (DATA-IN[7:0]), έχοντας θέσει το σήμα DATA-ALLOC. Το κύκλωμα αναζητά ελεύθερη θέση για χρήση.
  - Εφόσον βρει, σημειώνει ότι η θέση αυτή είναι σε χρήση (θέτοντας το πεδίο VALID), αποθηκεύει την είσοδο, και στην έξοδο (DATA-OUT, πλάτους 3 bits) αναφέρει τον αριθμό αυτής της θέσης. Στο παραπάνω σχήμα, η θέση 4 είναι ελεύθερη για χρήση, οπότε η έξοδος θα έπρεπε να αναφέρει τον αριθμό 4.
  - Εάν δεν βρεθεί ελεύθερη θέση, τότε το κύκλωμα θέτει το σήμα FULL. Καμία από τις ήδη αποθηκευμένες τιμές δεν επηρεάζεται.
- Για να ελευθερώσει θέση, ο χρήστης θέτει το σήμα DATA-FREE, και στην είσοδο DATA-CONSUME παρέχει τον αριθμό της θέσης που θέλει να αποδεσμεύσει.
- Η αρχική κατάσταση του κυκλώματος είναι: FULL=0, VALID=0 για τις 8 θέσεις αποθήκευσης.

## 1.3 Συγγραφή του Κώδικα Verilog και Προσομοίωση

Αφού ετοιμάσατε το σχέδιο του κυκλώματος, θα πρέπει να το μετατρέψετε σε κώδικα Verilog. Αμέσως μετά πρέπει να δημιουργήσετε ένα testbench module με το οποίο θα δοκιμάσετε να προσομοιώσετε το σύστημα.

## 1.4 Δημιουργία UCF και Τοποθέτηση

Αφού ολοκληρώσετε τα προηγούμενα βήματα, πρέπει να δημιουργήσετε το Αρχείο Περιορισμών Χρήστη (User Constraints File – UCF) δηλώνοντας τις εισόδους και τις εξόδους που θα χρησιμοποιήσετε στην πλακέτα και την τάση που θα τους ασκήσετε, όπως δείχνει το ucf αρχείο που λάβατε στο εργαστήριο 0 το οποίο είναι κομμάτι του master ucf που έχει δοθεί από τους κατασκευαστές και υπάρχει στο Documentation της πλακέτας στη σελίδα: <http://zedboard.org/documentation/1521>

## 1.5 Παράδοση

Θα παραδώσετε στους βοηθούς του μαθήματος αναφορά που θα περιλαμβάνει το σχέδιο του κυκλώματος, και τον κώδικα για την προσομοίωση.