



Πανεπιστήμιο Πατρών



***Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας
Υπολογιστών***

Εργαστήριο Σχεδίασης Ολοκληρωμένων Κυκλωμάτων

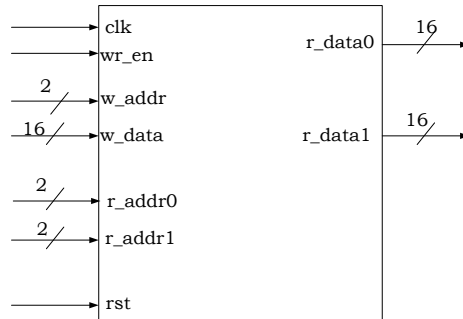
Σχεδιασμός Ολοκληρωμένων Ψηφιακών Συστημάτων

Χειμερινό Εξάμηνο 2023

Εργαστήριο 3

Άσκηση 1: Αρχείο καταχωρητών (Register File)

Περιγράψτε σε VHDL ένα αρχείο καταχωρητών που έχει μία θύρα εισόδου (write port) εγγραφής και δύο θύρες εξόδου (read ports) και περιέχει τέσσερις καταχωρητές εύρους 16 ψηφίων ο καθένας. Ο κάθε καταχωρητής έχει δικό του σήμα επίτρεψης εγγραφής. Το δομικό διάγραμμα του κυκλώματος είναι το ακόλουθο.



Η λειτουργία των σημάτων εισόδου εξόδου είναι η ακόλουθη:

Είσοδοι:

Clk, reset (asynchronous)

wr_en (write enable) – σήμα επίτρεψης εγγραφής στους καταχωρητές

w_addr (write address) – καθορισμός σε ποιο καταχωρητή θα γίνει εγγραφή, κάθε φορά γράφεται ένας μόνο καταχωρητής

w_data (write data) – δεδομένα εγγραφής

r_addr0 (read address0) – καθορίζει τα δεδομένα ποιου καταχωρητή θα εμφανιστούν στην θύρα εξόδου 0

r_addr1 – καθορίζει τα δεδομένα ποιου καταχωρητή θα εμφανιστούν στην θύρα εξόδου 1

Έξοδοι:

r_data0 (read_data0) – δεδομένα που διαβάζονται από τη θύρα εξόδου 0

r_data1 (read_data1) – δεδομένα που διαβάζονται από τη θύρα εξόδου 1

Προσέξτε όμως ότι κάθε φορά γράφεται ένας καταχωρητής (μία θέση μνήμης) ενώ οι άλλοι διατηρούν τις προηγούμενες τιμές.

Άσκηση 2: Universal Binary Counter

Περιγράψτε σε VHDL ένα δυαδικό απαριθμητή 4 bit η λειτουργία περιγράφεται στον ακόλουθο πίνακα. Τα σήματα του απαριθμητή είναι:

Είσοδοι: clk, reset(asynchronous), synch_clr, load, en, up, d (parallel input)

Έξοδοι: Q, max, min

Τα σήματα max και min ενεργοποιούνται όταν η έξοδος ισούται με τη μέγιστη και την ελάχιστη τιμή, αντίστοιχα.

Προσέξτε ότι πρέπει να υποστηρίζεται ασύγχρονο reset αλλά και σύγχρονο clear.

Synch_clr	Load	En	Up	Q(t+1)	Λειτουργία
1	—	—	—	0000	Synchronous clear
0	1	—	—	d	Parallel load
0	0	1	1	Q+1	Count up
0	0	1	0	Q-1	Count down
0	0	0	—	Q	Pause

Άσκηση 3: Δεκαδικός απαριθμητής

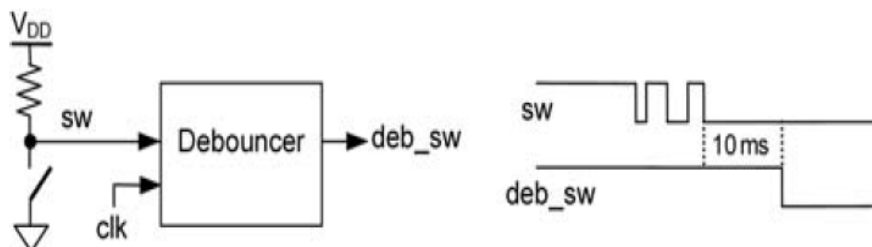
Ένας δεκαδικός απαριθμητής είναι ένα κύκλωμα που εκτελεί απαρίθμηση (counting) στο δεκαδικό σύστημα χρησιμοποιώντας αναπαράσταση BCD. Για παράδειγμα, ο αριθμός 139_{10} αναπαριστάται σε BCD ως “0001”, “0011”, “1001”. Περιγράψτε σε VHDL έναν δεκαδικό απαριθμητή που να μετρά από το 000 ως το 999. Όταν η έξοδος φτάνει στο 999 θα πρέπει να γυρίζει στο 000. Το κύκλωμα να υποστηρίζει ασύγχρονη επανεκκίνηση (asynchronous reset).

Άσκηση 4: Κύκλωμα φιλτραρίσματος αναπηδήσεων μηχανικού διακοπτή

Στο επόμενο σχήμα απεικονίζεται ένας μηχανικός διακόπτης ο οποίος παράγει ένα λογικό “0” όταν είναι κλειστός και ένα “1” όταν είναι ανοικτός, η οποία έξοδος χρησιμοποιείται σε κάποιο επόμενο κύκλωμα. Όμως, επειδή ο διακόπτης παρουσιάζει αναπηδήσεις (bounces), όταν είναι κλειστός, πρέπει να χρησιμοποιηθεί ένα debouncing κύκλωμα. Ο σκοπός αυτού του κυκλώματος είναι να καθαρίσει το θόρυβο στο σήμα sw, που παράγεται από το διακόπτη, έτσι ώστε να παραχθεί ένα καθαρό σήμα (bounce-free) deb_sw (debounced switch). Η διαδικασία αυτή δείχνεται στις κυματομορφές του επόμενου σχήματος.

Το κύκλωμα έχει ως εισόδους τις sw, clk. Όταν ο μηχανικός διακόπτης κλείνει τότε θα έπρεπε να ισχύει sw = 0. Όμως, λόγω ατελειών του παρουσιάζει αναπηδήσεις και μετά από κάποιο χρόνο έρχεται στην κατάσταση sw = 0.

Σχεδιάστε και περιγράψτε σε VHDL το παραπάνω κύκλωμα. Θεωρείστε ότι η νέα τιμή θα πρέπει να ανατίθεται στο σήμα deb_sw όταν το σήμα sw παραμένει στη κατάσταση sw = 0 για τουλάχιστον 10 ms. Θεωρείστε επίσης, ότι ο συχνότητα του ρολογιού είναι μία γενική παράμετρος.



Οι ασκήσεις να υλοποιηθούν χρησιμοποιώντας concurrent και sequential κώδικα όχι όμως με δομικό κώδικα δηλαδή χρήση port maps.