

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ



ΕΡΓΑΣΤΗΡΙΟ

ΣΧΕΔΙΑΣΗΣ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ - VHDL

9^ο Εργαστηριακό Μάθημα
3^ο Εργαστήριο

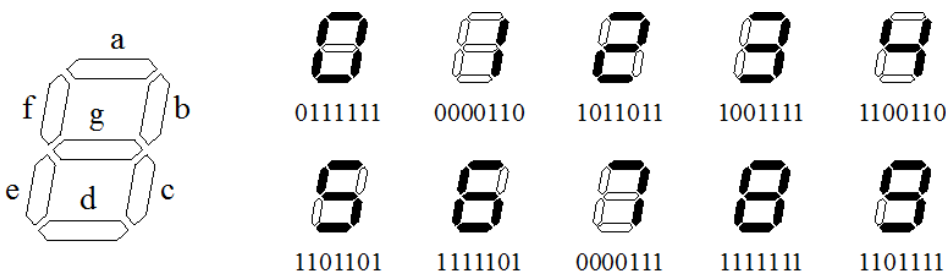
Βασιλόπουλος Διονύσης

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ
2024-2025

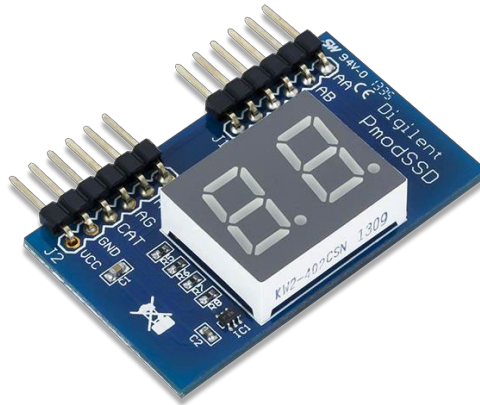
Άσκηση: Αποκωδικοποιητής 4-bit binary σε 7-segment

Να σχεδιάσετε, να προσομοιώσετε και να υλοποιήσετε μία **Αριθμητική και Λογική Μονάδα (ALU)**, η οποία είτε θα προσθέτει δύο μη προσημασμένους αριθμούς (σήματα a και b) των 3 bit είτε θα διπλασιάζει το a. Το τι πράξη θα κάνει εξαρτάται από ένα σήμα ελέγχου (ctr) του ενός bit. Όταν η τιμή του είναι '1' τότε κάνει την πρόσθεση ενώ όταν είναι '0' τότε κάνει το διπλασιασμό του a. Το αποτέλεσμα αποθηκεύεται στο σήμα result των 4 bit (άρα δεν θα ασχοληθείτε με το θέμα του overflow/carry γιατί το αποτέλεσμα χωράει στα 4bit). Η είσοδος θα γίνεται με τους διακόπτες (SW0, SW1, SW2 για το σήμα a, τους διακόπτες SW3, SW4, SW5 για το σήμα b και το διακόπτη SW6 για το σήμα ctr) και η έξοδος θα εμφανίζεται τόσο στα led (LD0, LD1, LD2, LD3) όσο και στην κάρτα rmod. Για την επιλογή του ψηφίου στο rmod στο οποίο θα εμφανίσετε το αποτέλεσμα θα χρησιμοποιήσετε το διακόπτη SW7(digit_selection_in και digit_selection_out η είσοδος και η έξοδος αντίστοιχα τύπου std_logic).

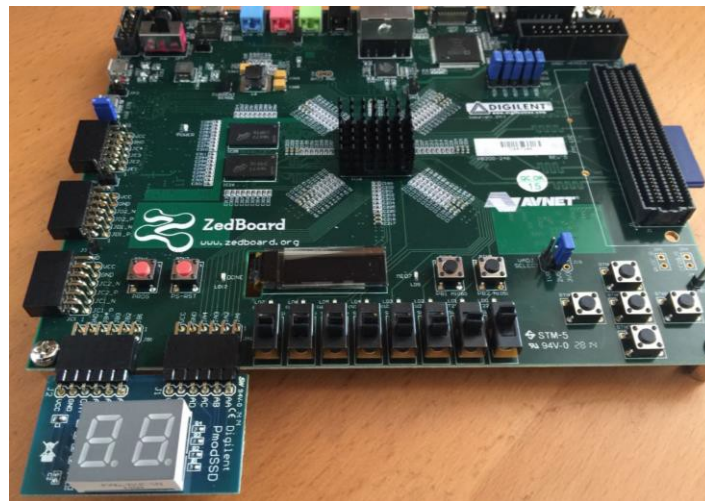
Πολλά ψηφιακά συστήματα εμφανίζουν τους δεκαδικούς αριθμούς με χρήση οθονών 7 τμημάτων (7-segment displays). Κάθε ψηφίο της οθόνης αποτελείται από επτά ξεχωριστά φώτα, που τοποθετούνται όπως φαίνεται στην παρακάτω εικόνα. Αν έχουμε ένα ψηφίο κωδικοποιημένο με BCD και πρέπει να δείξουμε το ψηφίο σε μια οθόνη 7 τμημάτων, χρειαζόμαστε έναν *αποκωδικοποιητή 7 τμημάτων (7-segment decoder)*. Μιλώντας αυστηρά, θα το ονομάζαμε «μετατροπέα κώδικα 7 τμημάτων» αφού μετατρέπει μια είσοδο binary των 4 bit σε μια έξοδο με κώδικα 7 τμημάτων. Ωστόσο, ο όρος «αποκωδικοποιητής 7 τμημάτων» χρησιμοποιείται ευρύτητα. Υποθέτοντας ότι ένα τμήμα ανάβει αν η είσοδος του είναι 1, χρειαζόμαστε έναν κώδικα των 7 bit για την αναπαράσταση των ψηφίων από 0 μέχρι 9. Η κωδική λέξη για κάθε ψηφίο έχει ένα bit με τιμή 1 για κάθε τμήμα που είναι αναμμένο και ένα bit 0 για κάθε τμήμα που δεν είναι αναμμένο. Τότε, ένας αποκωδικοποιητής 7 τμημάτων μετατρέπει μεταξύ του binary αριθμού και αυτού του κώδικα των 7 bit. Ένας πιθανός κώδικας φαίνεται στην παρακάτω εικόνα, με τα bit να αντιστοιχούν από αριστερά προς τα δεξιά στα τμήματα g μέχρι a (g, f, e, d, c, b, a). Το σήμα εξόδου για το 7-segment display είναι το seven_segment (σήμα 7 bit).



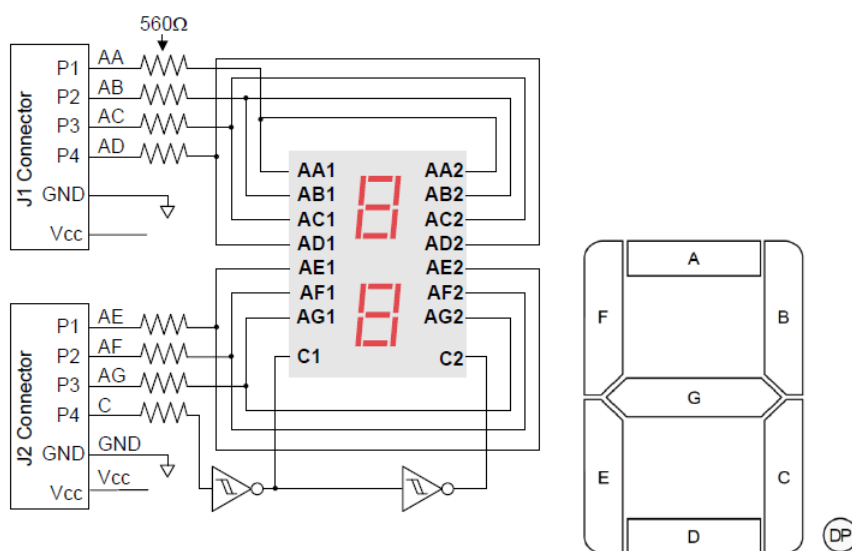
Το Pmod SSD υποστηρίζει δύο ψηφία (segments). Τα segments ανάβουν οδηγώντας με λογικό 1 την αντίστοιχη είσοδο του rmod σε όποιο από τα δύο ψηφία έχει επιλεγεί. **Κάθε στιγμή μόνο ΕΝΑ από τα δύο ψηφία μπορεί να είναι αναμμένο (εκ κατασκευής του pmod)**. Ο χρήστης μπορεί να επιλέξει ένα συγκεκριμένο ψηφίο χρησιμοποιώντας την είσοδο DIP switch SW7 και οδηγώντας αντίστοιχα το Digit Selection pin (C) του Pmod SSD σε λογικό 1 (για το αριστερό ψηφίο) ή 0 (για το δεξί ψηφίο). Το datasheet του Pmod SSD βρίσκεται στο e-class.



Εικόνα 1. Pmod SSD της Digilent



Εικόνα 2. Σύνδεση του Pmod SSD στα JA1 και JB1 Pmod connectors της ZedBoard



Εικόνα 3. Σχηματικό διάγραμμα του Pmod SSD της Digilent

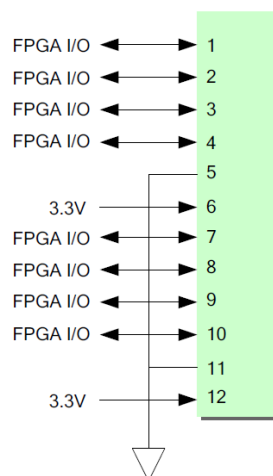
Header J1			Header J2		
Pin	Signal	Description	Pin	Signal	Description
1	AA	Segment A	1	AE	Segment E
2	AB	Segment B	2	AF	Segment F
3	AC	Segment C	3	AG	Segment G
4	AD	Segment D	4	C	Digit Selection pin
5	GND	Power Supply Ground	5	GND	Power Supply Ground
6	VCC	Positive Power Supply	6	VCC	Positive Power Supply

Πίνακας 1. Pinout του Pmod SSD της Digilent

Στα πλαίσια της άσκησης θα πρέπει να γράψετε το κατάλληλο αρχείο .xdc με τα constraints για τα pin assignments λαμβάνοντας υπόψιν α) το σχηματικό διάγραμμα του Pmod SSD (Εικόνα 3), β) τον Πίνακα 1 με το pinout του Pmod SSD γ) τον Πίνακα 2 που περιγράφει τις συνδέσεις των DIP switches με τα pins του Zynq FPGA, δ) στο σχηματικό διάγραμμα των συνδέσεων του Pmod SSD (Εικόνα 4) και ε) τον Πίνακα 3 που περιγράφει τις συνδέσεις των Pmod JA1 και Pmod JB1 με τα pins του Zynq FPGA. **Θα βρείτε μόνοι σας τα pins για τα LED.**

Signal Name	Zynq pin
SW0	F22
SW1	G22
SW2	H22
SW3	F21
SW4	H19
SW5	H18
SW6	H17
SW7	M15

Πίνακας 2. Συνδέσεις των DIP switches στα pins του Zynq FPGA



Εικόνα 4. Σχηματικό διάγραμμα των συνδέσεων του Pmod SSD της Digilent

Pmod	Signal Name	Zynq pin	Pmod	Signal Name	Zynq pin
JA1	JA1	Y11	JB1	JB1	W12
	JA2	AA11		JB2	W11
	JA3	Y10		JB3	V10
	JA4	AA9		JB4	W8
	JA7	AB11		JB7	V12
	JA8	AB10		JB8	W10
	JA9	AB9		JB9	V9
	JA10	AA8		JB10	V8

Πίνακας 3. Συνδέσεις των JA1 και JB1 Pmods στα pins του Zynq FPGA

Στο eclass θα σας έχει δοθεί ένα template πηγαίου κώδικα με την περιγραφή του entity καθώς και το αντίστοιχο testbench σε VHDL.

- Συμπληρώστε το architecture του entity, ώστε να υλοποιούνται οι δύο πράξεις (μπορείτε να το κάνετε πριν έρθετε στο εργαστήριο).
- Γράψτε το κατάλληλο αρχείο .xdc με τα pin constraints (που θα αφορά μόνο τα **Switches και τα Led**) (μπορείτε να το κάνετε πριν έρθετε στο εργαστήριο).
- Ελέγξτε τη λειτουργία του κυκλώματος τόσο στη προσομοίωση όσο και στο υλικό προγραμματίζοντας το FPGA
- Συμπληρώστε το αρχείο .xdc ώστε να ανάβει και το αντίστοιχο ψηφίο του pmod
- Συμπληρώστε τον παρακάτω πίνακα κοιτάζοντας το report μετά το implementation:

FF:	
LUT:	
I/O	