

## Μέρος Α (20%)

### Ασκήσεις Λογικής Σχεδίασης

**Ερώτημα 1.** (10%) Υλοποιήστε ένα Full Adder (του ενός bit) με έναν αποκωδικοποιητή και δύο πύλες OR .

**Ερώτημα 2.** (10%) Έστω ένα A-B Flip Flop, το οποίο δέχεται ένα clock (CLK) και 2 εισόδους A, B. Στην ανοδική ακμή του ρολογιού ενημερώνει την έξοδο Q. Εάν τα A, B είναι και τα δύο '0' τότε η έξοδος Q διατηρεί την προηγούμενη τιμή της. Εάν μόνο το A είναι '1' τότε το Q γίνεται '1'. Εάν μόνο το B είναι '1' τότε το Q γίνεται '0'. Εάν  $A=B=1$  τότε η νέα τιμή του Q γίνεται η αντίθετη της παρούσης. Βρείτε τον πίνακα αληθείας για το  $Q(t+1)$  και βρείτε την εξίσωση που προκύπτει για το  $Q(t+1)$  με τη χρήση K-maps. Γράψτε σε VHDL την αρχιτεκτονική του. Κατασκευάστε το A-B Flip-Flop χρησιμοποιώντας D Flip-Flop και συνδυαστική λογική (απλές πύλες 2 εισόδων και πολυπλέκτες).

Η απάντηση θα είναι σε αρχείο με το όνομα: `dsd_assign1_arithmos_mhtrnou_partA.zip`

## Μέρος Β (80%)

### Άσκηση VHDL

**Άσκηση 1.** Σχεδιάστε μία αριθμητική και λογική μονάδα (ALU). Στην είσοδο δέχεται 2 σήματα a και b, των 3 bit το καθένα που αναπαριστούν **προσημασμένους** αριθμούς, καθώς και ένα σήμα Control του ενός bit. Η ALU ανάλογα με τη τιμή του σήματος Control κάνει τις ακόλουθες πράξεις.

- '0': αφαίρεση ( $a-b$ )
- '1': τη διαίρεση  $b/2$  (με δεξιά ολίσθηση)

Στην έξοδο υπάρχει το σήμα Result των 3bit με το αποτέλεσμα της πράξης, το σήμα OVF που έχει τιμή '1' μόνο σε περίπτωση overflow και το σήμα Carry.

Θα δημιουργήσετε ένα νέο project στο Vivado με το όνομα ALU, θα χρησιμοποιήσετε τη δήλωση της οντότητας που σας δίδεται (την οποία **ΔΕΝ** μπορείτε να αλλάξετε), και θα προτείνετε αρχιτεκτονικές που δίνουν λύση στο πρόβλημα ως εξής:

- α) Αρχιτεκτονική Dataflow χωρίς τη χρήση selected statements (when/with select) και χωρίς τη χρήση της συνάρτησης resize.
- β) Αρχιτεκτονική Dataflow με τη χρήση selected statements (when/with select) και τη χρήση της συνάρτησης resize.
- γ) Αρχιτεκτονική Behavioral και if elsif else (μόνο) και δυνατότητα χρήσης μόνο μεταβλητής.
- δ) Αρχιτεκτονική Behavioral και case (μόνο) και δυνατότητα χρήσης μόνο εσωτερικού σήματος.

- e) Επεκτείνετε την περίπτωση (c) με τη χρήση generic (και όνομα παραμέτρου WIDTH) ώστε το πρόγραμμα να δέχεται μεταβλητό πλήθος bit στα σήματα a,b.
- f) Αρχιτεκτονική Structural με επέκταση της περίπτωσης (e) ώστε η οντότητα ALU να χρησιμοποιεί δύο component, το **Sub\_ab** και το **Div\_b**, που υλοποιούν την πρόσθεση και τη **διαίρεση** αντίστοιχα.

Θα πρέπει να γράψετε τον αντίστοιχο κώδικα προσομοίωσης, να εμφανίσετε το behavioral και Post Synthesis Timing Simulation και τα αντίστοιχα RTL και Synthesis διαγράμματα (Schematic).

Θα πρέπει να παραδώσετε τα ακόλουθα:

1. Αρχείο με τον κώδικα του design (τα αρχεία του Vivado, είτε 1 συνολικό είτε 6 διαφορετικά)
2. Αρχείο με τον κώδικα της προσομοίωσης (το αρχείο του Vivado)
3. Αρχείο pdf ή word όπου θα υπάρχουν:
  - I. Τα διαγράμματα (schematic) που προκύπτουν από το RTL Analysis (Schematic) για κάθε μια από τις 6 περιπτώσεις.
  - II. Την κυματομορφή που προκύπτει από το behavioral simulation (θα πρέπει να είναι ίδια σε όλες τις περιπτώσεις οπότε θέλω μία).
  - III. Το διάγραμμα (schematic) που προκύπτει μετά τη φάση της Σύνθεσης (εάν είναι ίδιο σε όλες τις περιπτώσεις το θέλω μία φορά).
  - IV. Την κυματομορφή που προκύπτει από το Post Synthesis Timing simulation.
  - V. Να βρείτε το utilization μετά το Implementation
  - VI. Την καθυστέρηση διάδοσης και την καθυστέρηση μόλυνσης του κυκλώματος στην περίπτωση a) και τα αντίστοιχα μονοπάτια (print\_screen από το vivado).

**Η απάντηση θα είναι σε αρχείο με το όνομα:dsd\_assign1\_arithmos\_mhtrvou\_partB.zip**

Για να ανεβάσετε την άσκηση στο eclass θα βάλετε σε ένα zip και τα δύο αρχεία (μέρος A και μέρος B).