

Σχεδίαση Ψηφιακών Συστημάτων - VHDL

1^η Εργαστηριακή Άσκηση

Σχεδιάστε μια ALU που εκτελεί αριθμητικές και λογικές λειτουργίες παράλληλα. Σας δίδεται η οντότητα:

```
entity ALU is
  port (
    A, B : in std_logic_vector(31 downto 0);
    OP   : in std_logic_vector(2 downto 0);
    Signed_mode : in std_logic;
    Result: out std_logic_vector(31 downto 0);
    Z : out std_logic_vector(3 downto 0);
    N : out std_logic_vector(3 downto 0);
    C : out std_logic_vector(3 downto 0);
    V : out std_logic_vector(3 downto 0);
    S : out std_logic_vector(3 downto 0)
  );
end entity ALU;
```

Ειδικότερα:

1. A και B είναι οι αριθμοί πάνω στους οποίους θα γίνουν οι πράξεις

2. op είναι η κωδικοποίηση της πράξης ως εξής:

000	ADD	Addition
001	SUB	Subtraction (A-B)
010	AND	Bitwise AND
011	XOR	Bitwise XOR
100	MAX	Maximum of A and B
101	SAT_ADD	Saturating Addition
110	LSL	Logical shift left A (κατά 1 bit)
111	ASR	Arithmetic shift right B (κατά 1 bit)

3. Signed_mode: Υποδεικνύει εάν οι αριθμοί στους οποίους εφαρμόζονται οι πράξεις είναι signed (τιμή '1') ή unsigned (τιμή '0').

4. Ως προς τις σημαίες:

Σε περίπτωση signed αριθμού

N: Ένδειξη για αρνητικό αριθμό

V: Ένδειξη για υπερχείλιση προσημασμένων αριθμών. **Στην περίπτωση της εντολής SAT_ADD είναι 0.**

S: Ένδειξη για saturation (https://en.wikipedia.org/wiki/Saturation_arithmetic) . **Γίνεται 1 μόνο όταν έχουμε αντίστοιχη πράξη και υπάρχει υπέρβαση των ορίων. Διαφορετικά έχει τιμή 0.**

Z: Ένδειξη για αποτέλεσμα ίσο με μηδέν

C: Θα είναι '0'

Σε περίπτωση unsigned αριθμού

N: Θα είναι '0'

V: Θα είναι '0'

S: Ένδειξη για saturation (https://en.wikipedia.org/wiki/Saturation_arithmetic) . **Γίνεται 1 μόνο όταν έχουμε αντίστοιχη πράξη και υπάρχει υπέρβαση των ορίων. Διαφορετικά έχει τιμή 0.**

Z: Ένδειξη για αποτέλεσμα ίσο με μηδέν

C: Ένδειξη για υπερχείλιση μη προσημασμένων αριθμών. **Στην περίπτωση της εντολής SAT_ADD είναι 0.**

Θεωρείστε ότι τα A και B μεταφέρουν 4 αριθμούς των 8 bit το καθένα. Για το σήμα A αυτοί θα είναι A0=a(7 downto 0), A1=a(15 downto 8), A2=a(23 downto 16), A3=a(31 downto 24). Αντίστοιχα για το σήμα B. Οι πράξεις που ορίζονται στο OP θα γίνονται ταυτόχρονα και στους τέσσερις αριθμούς. Σε αυτή την περίπτωση η σημαία Z(0) θα δείχνει αν η πράξη πάνω στους αριθμούς A0 και B0 έχει αποτέλεσμα "000" ή όχι. Η Z(1) θα αφορά την πράξη πάνω στους A1 και B1, κλπ. Αντίστοιχα και οι άλλες σημαίες N, V, S, C.

5. Result: Το σήμα που θα έχει το αποτέλεσμα των πράξεων.

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). **Χρησιμοποιείτε όποιο τύπο αρχιτεκτονικής θέλετε (60%).**

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 γράψετε κώδικα προσομοίωσης (**20%**).

Θα πρέπει να παραδώσετε τα ακόλουθα:

1. Αρχείο με τον κώδικα του design (τα αρχεία του Vivado)
2. Αρχείο με τον κώδικα της προσομοίωσης (το αρχείο του Vivado)
3. Αρχείο pdf ή word όπου θα υπάρχουν:

- I. **Την κυματομορφή που προκύπτει από το behavioral Simulation και το Post Synthesis Timing simulation. (10%)**
- II. Την καθυστέρηση διάδοσης και την καθυστέρηση μόλυνσης του κυκλώματος και τα αντίστοιχα μονοπάτια (print_screen από το vivado) μετά από τη **Σύνθεση** (10%)

Για να ανεβάσετε την άσκηση στο eclass θα βάλετε τα αρχεία σε ένα zip (**dsd_assign1_arithmos_mhtrvou**).