

Εργαστήριο Ψηφιακής Σχεδίασης Συστημάτων - VHDL

Ασκήσεις VHDL

Ερώτημα 1. (25%) Σχεδιάστε ένα κύκλωμα με VHDL για έναν μετρητή (counter) που μετράει πόσες φορές η απόλυτη τιμή στην είσοδο του κυκλώματος είναι μεγαλύτερη του 4 (ερμηνεύετε την είσοδο ως signed). Η είσοδος είναι των 4 bit και η έξοδος των 3bit. Εάν ο μετρητής φτάσει στη τιμή 6, τότε η επόμενη τιμή του είναι η 1 (ουσιαστικά ξεκινάει πάλι από την αρχή). Σας δίδεται ο ορισμός:

```
entity Counter is
Port ( CLK: in std_logic;
      RESET: in STD_logic;
      Value: in std_logic_vector(3 downto 0);
      Times: out std_logic_vector(2 downto 0));
end Counter;
```

Ερώτημα 2. (30%) Σχεδιάστε ένα Mealy FSM με VHDL, που ανιχνεύει τις ακολουθίες εισόδου 00011 και 00001. Είσοδος και έξοδος του κυκλώματος είναι τύπου std_logic. Σας δίδεται ο ορισμός.

```
entity Mealy_FSM is
Port ( CLK : in STD_LOGIC;
      RESET : in STD_LOGIC;
      X : in STD_LOGIC;
      Y : out STD_LOGIC);
end Mealy_FSM;
```

Ερώτημα 3. (55%) Έστω ότι η ακόλουθη μέθοδος (δίνεται ως παράδειγμα) είναι ο τρόπος για να ορίσετε όχι μόνο ένα FSM αλλά και να **καθορίσετε εσείς** τον τρόπο κωδικοποίησης των καταστάσεων από τη VHDL/Vivado.

-- state definition – One Hot κωδικοποίηση

```
type FSM_states is (S0, S1, S2, S3, S4);
```

```
attribute enum_encoding: string;
```

```
attribute enum_encoding of FSM_states: type is "00001 00010 00100 01000 10000";
```

Έστω ότι ανιχνεύετε την ακολουθία 01101. Για τα 3 διαφορετικά είδη κωδικοποίησης (binary,

Gray, one-hot) συγκρίνετε το συνδυαστικό κύκλωμα επόμενης κατάστασης και το συνδυαστικό κύκλωμα της εξόδου που προκύπτουν ανάλογα την κωδικοποίηση. Τι παρατηρείτε σε ταχύτητα (Post Implementation), χρήση πόρων (σε RTL αναπαράσταση και Post Implementation); Θα πρέπει να αναπτύξετε τον κατάλληλο κώδικα και να παραθέσετε αναλυτικά τα δεδομένα με βάση τα οποία καταλήγετε στα όποια συμπεράσματα. Το FSM είναι τύπου Moore

Σας δίδεται ο ορισμός.

```
entity FSM is
    Port ( CLK : in STD_LOGIC;
          RESET : in STD_LOGIC;
          X : in STD_LOGIC;
          Y : out STD_LOGIC);
end FSM;
```