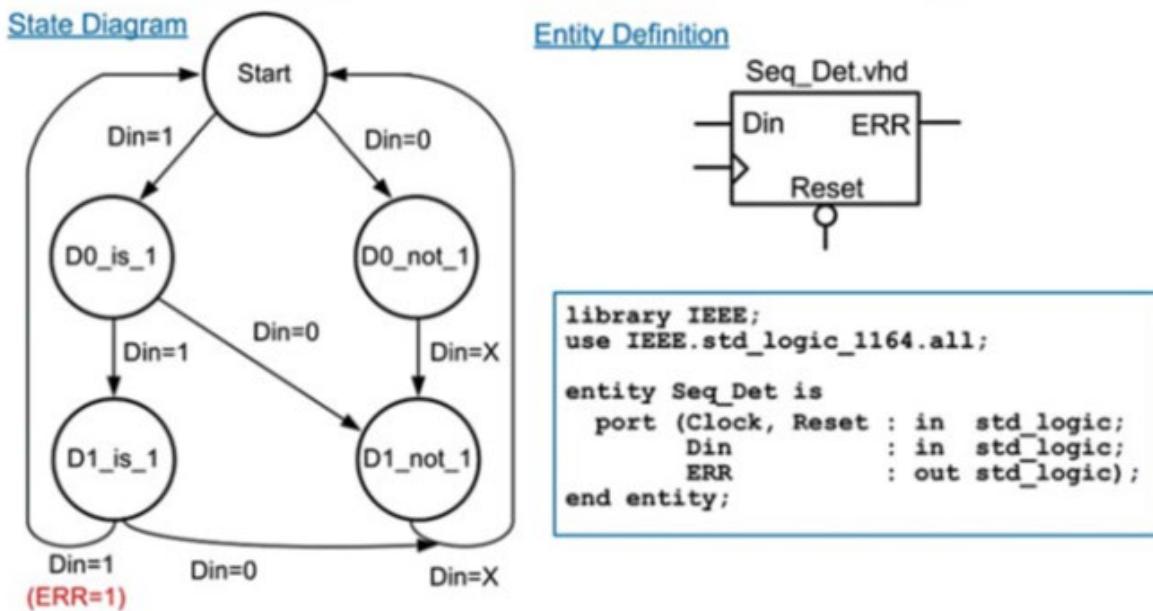


Σχεδίαση Ψηφιακών Συστημάτων - VHDL

2^η Εργαστηριακή Άσκηση

Ερώτημα 1. (30%) Σας δίδεται το ακόλουθο διάγραμμα καταστάσεων μαζί με τον ορισμό της οντότητάς σας (Seq_Dat).



Το κύκλωμα δέχεται μία σειριακή ακολουθία από bit. Η λήψη της ακολουθίας “111” υποδεικνύει την ύπαρξη λάθους στην μετάδοση των bit. Το FSM παρακολουθεί την ακολουθία εισόδου και θέτει το σήμα εξόδου ERR σε τιμή ‘1’ όταν ανιχνεύσει λάθος.

(a) Υλοποιήστε σε VHDL το κύκλωμα του FSM (15%),

(b) αναπτύξτε το αντίστοιχο testbench (5%) και

(c) Αναφέρεται την ταχύτητα στην οποία δουλεύει το κύκλωμά σας (slack, κρίσιμο μονοπάτι και Tc/F), μετά την υλοποίηση (5%)

(d) Αναφέρεται την καθυστέρηση μόλυνσης (slack, μονοπάτι, χρόνος μόλυνσης), μετά την υλοποίηση (5%)

(e) Αναφέρεται τους πόρους που χρησιμοποιεί μετά την υλοποίηση (5%)

(f) Η πιο γρήγορη (και απόλυτα σωστή υλοποίηση) +5%

Ερώτημα 2. (70%) Υλοποιήστε τον αλγόριθμο bubble sort σε VHDL. Τα δεδομένα προς ταξινόμηση είναι σε μνήμη ROM. Τα δεδομένα ταξινομούνται με βάση τον αλγόριθμο σε αύξουσα σειρά, και κατόπιν αποθηκεύονται σε μνήμη RAM. Σας δίδεται η οντότητα του design:

```
entity Bubble_sort is
generic (
AWIDTH: natural:=10;
DWIDTH: natural:=32
);
Port (
clk: in std_logic;
reset: in std_logic;
Data_Out: out std_logic_vector(DWIDTH-1 downto 0)
);
end Bubble_sort;
```

AWIDTH είναι το εύρος των διευθύνσεων της ROM/RAM και DWIDTH το μέγεθος κάθε λέξης της ROM/RAM. Το σήμα Data_Out είναι το τρέχων περιεχόμενο της θέσης μνήμης RAM στην οποία δείχνει το πρόγραμμά σας. Θα πρέπει να χρησιμοποιήσετε τις οντότητες της ROM και της RAM που σας δίνονται (υπάρχουν και αναλύονται στις διαφάνειες του κ.Πασχάλη). Μπορείτε να χρησιμοποιήσετε όποια δομή και τύπο θέλετε για την υλοποίηση.

- (a)Υλοποίηση σε VHDL (45%),
- (b) Ανάπτυξη testbench (20%) και
- (c)Αναφέρεται την ταχύτητα στην οποία δουλεύει το κύκλωμά σας (slack, κρίσιμο μονοπάτι και Tc/F), μετά την υλοποίηση (5%)
- (d)Αναφέρεται την καθυστέρηση μόλυνσης (slack, μονοπάτι, χρόνος μόλυνσης), μετά την υλοποίηση (5%)
- (e)Αναφέρεται τους πόρους που χρησιμοποιεί το κύκλωμα μετά την υλοποίηση (5%)
- (f)Η πιο γρήγορη υλοποίηση (post implementation, και απόλυτα σωστή υλοποίηση) +5%
- (g)Η υλοποίηση με τους λιγότερους πόρους (post implementation, και απόλυτα σωστή υλοποίηση) +5%
- (h)Δώστε μια σύντομη ανάλυση (2 σελίδων) του προγράμματός σας (κυρίως του τρόπου σκέψης σας για να λύσετε την άσκηση) (5%)

Link για να δείτε τον αλγόριθμο

<https://www.geeksforgeeks.org/dsa/bubble-sort-algorithm/>

https://en.wikipedia.org/wiki/Bubble_sort

Για να ανεβάσετε την άσκηση στο eclass θα βάλετε τα αρχεία σε ένα zip (**dsd_assign2_arithmos_mhtrvou**).