

Πανεπιστήμιο Δυτικής Αττικής
Σχολή Μηχανικών
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Εργαστήριο Σχεδίασης Ψηφιακών Συστημάτων

Εργαστηριακή Άσκηση 3: Γνωριμία με το περιβάλλον προσομοίωσης

1. Γνωριμία με το εργαλείο (1)

Ακολουθώντας τις οδηγίες του αρχείου «Προσομοίωση στο εργαλείο Modelsim Altera Starter edition 6_uniwa» (Μέρος Α) υλοποιήστε και προσομοιώστε ένα κύκλωμα πολυπλέκτη 2-σε-1

2. Γνωριμία με το εργαλείο (2)

Ακολουθώντας τις οδηγίες του αρχείου «Προσομοίωση στο εργαλείο Modelsim Altera Starter edition 6_uniwa» (Μέρος Β) υλοποιήστε και προσομοιώστε ένα κύκλωμα πολυπλέκτη 2-σε-1 με χρήση testbench

3. Τριπλός πολυπλέκτης 2-σε-1

Υλοποιήστε κύκλωμα τριπλού πολυπλέκτη 2-σε-1, ο οποίος παίρνει δύο 3-bit εισόδους a και b, μία είσοδο επιλογής s και δίνει στην έξοδο ένα 3-bit σήμα d, ανάλογα με το εάν η είσοδος s είναι 1 ή 0. Η περιγραφή του entity είναι η ακόλουθη:

```
entity mux_double_2to1 is
port
(a, b: in std_logic_vector(2 downto 0);
 s: in std_logic;
 d: out std_logic_vector(2 downto 0));
end mux_double_2to1;
```

Προσομοιώστε το κύκλωμα για τους ακόλουθους συνδυασμούς εισόδων:

s	a	b	d
0	001	010	
0	010	100	
0	111	011	
0	101	111	
1	010	001	
1	000	101	
1	101	010	
1	111	101	

Χρησιμοποιώντας tb που θα αναπτύξετε για το σκοπό αυτό.

4. Πολυπλέκτης 4-σε-1

Ένας πολυπλέκτης 4-σε-1 παίρνει μια 4-ψήφια είσοδο δεδομένων και μια 2-ψήφια είσοδο ελέγχου και έχει μια μονοψήφια είσοδο η οποία είναι κάποια από τις εισόδους δεδομένων ανάλογα με το συνδυασμό τιμών της εισόδου επιλογής. Η περιγραφή του entity είναι η ακόλουθη:

```
entity mux_4to1 is port (
  a: in std_logic_vector(4 downto 1);
  s: in std_logic_vector(2 downto 1);
  d: out std_logic);
end mux_4to1;
```

Προσομοιώστε το κύκλωμα για τους ακόλουθους συνδυασμούς εισόδων:

a	s	d
0000	00	

0101	01	
1010	10	
1100	11	

5. Αποκωδικοποιητής 2-σε-4

Υλοποιήστε κύκλωμα αποκωδικοποιητή 2-σε-4. Η περιγραφή του entity είναι η ακόλουθη:

```
entity dec2to4 is
port (
  a: in  std_logic_vector(2 downto 1);
  d: out std_logic_vector(4 downto 1));
end dec2to4;
```

Υλοποιήστε το κύκλωμα χρησιμοποιώντας μόνο την εντολή <= (με βάση το λογικό κύκλωμα)

Ελέγξτε το κύκλωμα για τους ακόλουθους συνδυασμούς χρησιμοποιώντας εντολές force:

a	d
00	
01	
10	
11	

6. Αποκωδικοποιητής 2-σε-4 με επίτρεψη

Υλοποιήστε κύκλωμα αποκωδικοποιητή 2-σε-4 με επίτρεψη. Αν η είσοδος επίτρεψης είναι 0, όλες οι έξοδοι είναι 0. Η περιγραφή του entity είναι η ακόλουθη:

```
entity dec_2to4 is
port (
  a: in  std_logic_vector(2 downto 1);
  en: in  std_logic;
  d: out std_logic_vector(4 downto 1) );
end dec_2to4;
```

Υλοποιήστε το κύκλωμα χρησιμοποιώντας μόνο την εντολή <= (με βάση το λογικό κύκλωμα)

Ελέγξτε το κύκλωμα για τους ακόλουθους συνδυασμούς

a	en	d
00	0	
01	0	
10	0	
11	0	
00	1	
01	1	
10	1	
11	1	

7. Αποκωδικοποιητής 4-σε-16

Υλοποιήστε κύκλωμα αποκωδικοποιητή 4-σε-16. Η περιγραφή του entity είναι η ακόλουθη:

```
entity dec_4to16 is port (
  a: in  std_logic_vector( 4 downto 1);
  d: out std_logic_vector(16 downto 1));
end dec4to16;
```

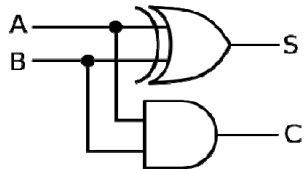
Υλοποιήστε το κύκλωμα χρησιμοποιώντας μόνο την εντολή <= (με βάση το λογικό κύκλωμα)

Ελέγξτε το κύκλωμα για τους ακόλουθους συνδυασμούς με εντολές force

a	d
0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

8. Ημιαθροιστής

Υλοποιήστε κύκλωμα ημιαθροιστή. Η περιγραφή του entity είναι η ακόλουθη:

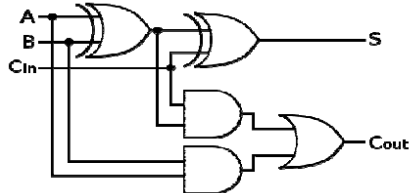
<pre>entity ha is port (A, B : in bit; S, C : out bit); end ha;</pre>	
--	--

Ελέγξτε το κύκλωμα για του ακόλουθους συνδυασμούς:

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

9. Πλήρης αθροιστής

Υλοποιήστε κύκλωμα πλήρους αθροιστή. Η περιγραφή του entity είναι η ακόλουθη:

<pre>entity fa is port (A, B, Cin : in bit; S, Cout : out bit); end fa;</pre>	
--	--

Ελέγξτε το κύκλωμα για τους ακόλουθους συνδυασμούς

A	B	Cin	Cout	S
0	0	0	0	0
0	0	1	0	1

0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

10. Αθροιστής 4 bit

Υλοποιήστε κύκλωμα αθροιστή 4 bit. Η περιγραφή του entity είναι η ακόλουθη:

```
ENTITY adder4 IS PORT (
  Cin  : IN  STD_LOGIC;
  X, Y : IN  STD_LOGIC_VECTOR(3 DOWNTO 0);
  S     : OUT STD_LOGIC_VECTOR(3 DOWNTO 0);
  Cout  : OUT STD_LOGIC);
END adder4;
```

Ελέγξτε το κύκλωμα για τους ακόλουθους συνδυασμούς

A	B	Cin	Cout	S
0000	0000	0	0	0000
1111	1111	0	1	1110
1111	1111	1	1	1111

Στη συνέχεια ελέγξτε το κύκλωμα για την πρόσθεση των ακόλουθων αριθμών

- 3 + 5
- -2 + (3)
- -8 + (+7)