



Πανεπιστήμιο Δυτικής Αττικής

Σχολή Μηχανικών

Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Εργαστήριο Σχεδίαση Ψηφιακών Συστημάτων

Τμήμα 1

Άσκηση 1

Αναστασία Αλμπάνη

Ice19390009

## 1. Τριπλός πολυπλέκτης 2-σε-1

Το κύκλωμα υλοποιείται στο αρχείο mux\_double\_2to1.vhd και το testbench στο αρχείο test\_double\_2to1.vhd. Ο τριπλός πολυπλέκτης 2-σε-1 έχει την ίδια λειτουργία με τον πολυπλέκτη 2-σε-1 με την διαφορά ότι δέχεται εισόδους με τρία bit αντί για ένα. Δέχεται δύο εισόδους a, b με τρία bit και μία είσοδο s με ένα bit η οποία ανάλογα με την τιμή της θα καθοριστεί η έξοδος d με τρία bit. Αν η είσοδος s είναι 1 τότε η έξοδος d θα έχει τις τιμές της εισόδου a, αλλιώς θα έχει τις τιμές της εισόδου b.

Οπότε για τις ακόλουθες τιμές των s, a, b η έξοδος d θα είναι:

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

Χρησιμοποιώντας το testbench τα σήματα A1, B1, c1, s1 αντιστοιχούν στις θύρες a, b, d, s του πολυπλέκτη.

/A1	001	001	010	111	101	010	000	101	111
/B1	010	010	100	011	111	001	101	010	101
/c1	010	010	100	011	111	010	000	101	111
/s1	0								

## 2. Πολυπλέκτης 4-σε-1

Το κύκλωμα υλοποιείται στο αρχείο mux\_4to1.vhd και το testbench στο αρχείο test\_mux\_4to1.vhd. Ο πολυπλέκτης 4-σε-1 δέχεται μία είσοδο a με τέσσερα bit και μία είσοδο s με δύο bit η οποία ανάλογα την τιμή της η έξοδος d με ένα bit παίρνει την τιμή ενός ψηφίου από την είσοδο.

Ακολουθεί πίνακας αληθείας ο οποίος περιγράφει την συμπεριφορά του κυκλώματος:

s2	s1	d
0	0	a1
0	1	a2
1	0	a3
1	1	a4

Για τους ακόλουθους συνδυασμούς εισόδων υπολογίζεται η έξοδος d:

a	s	d
0000	00	0
0101	01	0
1010	10	0
1100	11	1

Χρησιμοποιώντας το testbench τα σήματα A1, s1, d1 αντιστοιχούν στις θύρες a, s, d του πολυπλέκτη.

A1	0000	0000	0101	1010	1100
s1	00	00	01	10	11
d1	0				

### 3. Αποκωδικοποιητής 2-σε-4

Το κύκλωμα υλοποιείται στο αρχείο dec2to4.vhd και το testbench στο αρχείο test\_dec2to4.vhd. Ο αποκωδικοποιητής 2-σε-4 δέχεται μία είσοδο a με δύο bit και ανάλογα με τον συνδυασμό τιμών της ορίζεται η έξοδος d με τέσσερα bit.

Οπότε για τις ακόλουθες τιμές του a η έξοδος d θα είναι:

a	d
00	1000
01	0100
10	0010
11	0001

Χρησιμοποιώντας το testbench τα σήματα A1, d1 αντιστοιχούν στις θύρες a, d του αποκωδικοποιητή.

a	00	00	01	10	11	
d	1000	1000	0100	0010	0001	

#### 4. Αποκωδικοποιητής 2-σε-4 με επίτρεψη

Το κύκλωμα υλοποιείται στο αρχείο dec\_2to4.vhd και το testbench στο αρχείο test\_dec\_2to4.vhd. Ο αποκωδικοποιητής 2-σε-4 με επίτρεψη έχει παρόμοια λειτουργία με τον αποκωδικοποιητή χωρίς επίτρεψη. Η διαφορά τους είναι ότι ο αποκωδικοποιητής με επίτρεψη έχει μία επιπλέον είσοδο en με ένα bit η οποία ανάλογα την τιμή της ελέγχεται η λειτουργία του κυκλώματος. Αν έχει είσοδο 0 τότε δεν λειτουργεί το κύκλωμα και όλες οι έξοδοι d θα είναι 0, αλλιώς αν έχει είσοδο 1 τότε το κύκλωμα λειτουργεί κανονικά όπως ο αποκωδικοποιητής χωρίς επίτρεψη.

Οπότε για τις ακόλουθες τιμές των a, en η έξοδος d θα είναι:

a	en	d
00	0	0000
01	0	0000
10	0	0000
11	0	0000
00	1	1000
01	1	0100
10	1	0010
11	1	0001

Χρησιμοποιώντας το testbench τα σήματα A1, e1, d1 αντιστοιχούν στις θύρες a, en, d του αποκωδικοποιητή.

A1	00	00	01	10	11	00	01	10	11
e1	0								
d1	0000	0000				1000	0100	0010	0001

## 5. Αποκωδικοποιητής 4-σε-16

Το κύκλωμα υλοποιείται στο αρχείο `dec_4to16.vhd` και το testbench στο αρχείο `test_dec_4to16.vhd`. Ο αποκωδικοποιητής 4-σε-16 ενσωματώνει πέντε αποκωδικοποιητές 2-σε-4. Οι τέσσερις πρώτοι αποκωδικοποιητές `u4` έως `u1` έχουν ως εισόδους τα δύο λιγότερο σημαντικά bit `a(1)`, `a(2)` της εισόδου του 4-σε-16 και οι έξοδοι `d(4 downto 1)` τους οι οποίες έχουν τέσσερα bit αντιστοιχούν στις εξόδους `d(16 downto 0)` του 4-σε-16. Ο αποκωδικοποιητής `u0` έχει ως εισόδους τα δύο πιο σημαντικά bit `a(3)`, `a(4)` της εισόδου του 4-σε-16 και οι έξοδοι του `u0` συνδέονται με τα enable των υπόλοιπων 2-σε-4. Το enable του `u0` θέτεται σε 1 για να βρίσκεται ο αποκωδικοποιητής 4-σε-16 σε συνεχή λειτουργία.

Για τις ακόλουθες τιμές του `a` η έξοδος `d` θα είναι:

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

Χρησιμοποιώντας το testbench τα σήματα `A1`, `d1` αντιστοιχούν στις θύρες `a`, `d` του αποκωδικοποιητή.

A1	0000	0000		0001		0010		0011		
d1	1000000000000000	1000000000000000	0100000000000000	0010000000000000	0001000000000000					
A1	0100	0100		0101		0110		0111		
d1	0000100000000000	0000100000000000	0000010000000000	0000001000000000	0000000100000000	0000000010000000				
A1	1000	1000		1001		1010		1011		
d1	0000000010000000	0000000010000000	0000000001000000	0000000000100000	0000000000010000	0000000000001000				
A1	1100	1100		1101		1110		1111		
d1	0000000000001000	0000000000001000	0000000000000100	0000000000000010	0000000000000001					

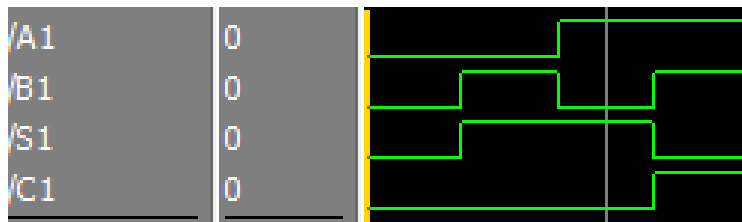
## 6. Ημιαθροιστής

Το κύκλωμα υλοποιείται στο αρχείο `half_adder.vhd` και το testbench στο αρχείο `test_half_adder.vhd`. Ο ημιαθροιστής δέχεται δύο εισόδους A, B με ένα bit και αναθέτει ανάλογα τις τιμές του τις εξόδους S, C με ένα bit. Αν μόνο μία από τις δύο εξόδους έχει τιμή 1 τότε η έξοδος sum S θα έχει τιμή 1 αλλιώς θα έχει τιμή 0. Αν και οι δύο έξοδοι έχουν τιμή 1 τότε η έξοδος carry C θα έχει τιμή 1 αλλιώς θα έχει τιμή 0.

Για τις ακόλουθες τιμές των A, B οι έξοδοι C, S θα είναι:

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

Χρησιμοποιώντας το testbench τα σήματα A1, B1, S1, C1 αντιστοιχούν στις θύρες A, B, S, C του ημιαθροιστή.



## 7. Πλήρης αθροιστής

Το κύκλωμα υλοποιείται στο αρχείο `full_adder.vhd` και το testbench στο αρχείο `test_full_adder.vhd`. Ο πλήρης αθροιστής ενσωματώνει δύο ημιαθροιστές. Ο πρώτος ημιαθροιστής `ha1` δέχεται στις εισόδους `A`, `B` του τις εισόδους `A`, `B` του πλήρη αθροιστή και η έξοδος `S` του `ha1` (σήμα `s1`) συνδέεται με την είσοδο `A` του δεύτερου ημιαθροιστή `ha2`. Η δεύτερη είσοδος `B` του `ha2` συνδέεται με την είσοδο `Cin` του πλήρη αθροιστή και η έξοδος `S` του `ha2` συνδέεται με την τελική έξοδο `S` του πλήρη αθροιστή. Η έξοδος `Cout` του πλήρη αθροιστή είναι η πράξη `or` των εξόδων `C` των `ha1` (σήμα `co1`) και `ha2` (σήμα `co2`).

Για τις ακόλουθες τιμές των `A`, `B`, `Cin` οι έξοδοι `Cout`, `S` θα είναι:

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

Χρησιμοποιώντας το testbench τα σήματα `a1`, `v1`, `ci1`, `s1`, `co1` αντιστοιχούν στις θύρες `A`, `B`, `Cin`, `S`, `Cout` του πλήρη αθροιστή.

a1	0								
b1	0								
ci1	0								
s1	0								
co1	0								



## 8. Αθροιστής 4 bit

Το κύκλωμα υλοποιείται στο αρχείο `adder4.vhd` και το testbench στο αρχείο `test_adder4.vhd`. Ο αθροιστής 4 bit ενσωματώνει τέσσερις πλήρεις αθροιστές οι οποίοι έχουν υλοποιηθεί στο αρχείο `adder4_fa.vhd` με τύπο port `std_logic`. Οι τέσσερις πλήρεις αθροιστές `fa1` έως `fa4` δέχονται στις εισόδους τους `a`, `b` τις εισόδους `X`, `Y` (`X(3 downto 0)` και `Y(3 downto 0)`) του αθροιστή 4 bit και οι 1-bit έξοδοι τους `s` αντιστοιχούν στην έξοδο `S` του αθροιστή 4 bit. Η είσοδος `cin` του πρώτου πλήρη αθροιστή `fa1` συνδέεται με την είσοδο `Cin` του αθροιστή 4 bit και η έξοδος `cout` (σήμα `co1(0)`) του `fa1` συνδέεται με την είσοδο `cin` του `fa2`. Η έξοδος `Cout` (σήμα `co1(1)`) του `fa2` συνδέεται με την είσοδο `Cin` του `fa3` και η έξοδος `Cout` (σήμα `s1(2)`) του `fa3` συνδέεται με την είσοδο `Cin` του `fa4`. Τέλος η έξοδος `Cout` (σήμα `s1(3)`) του `fa4` συνδέεται με την έξοδο `Cout` του αθροιστή 4 bit.

Για τους ακόλουθους συνδυασμούς των `A`, `B`, `Cin` οι έξοδοι `Cout`, `S` θα είναι:

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

Χρησιμοποιώντας το testbench τα σήματα `A`, `B`, `s1`, `ci1`, `co1` αντιστοιχούν στις θύρες `A`, `B`, `S`, `Cin`, `Cout` του αθροιστή 4 bit.

A	0000	0000	1111			
B	0000	0000	1111			
s1	0000	0000	1110	1111		
ci1	0					
co1	0					

Ακολουθεί έλεγχος του κυκλώματος με πράξεις πρόσθεσης των αριθμών:

- $3 + 5$

Για  $A = +3 = 0011$ ,  $B = +5 = 0101$  και  $C_{in} = 0$  το κύκλωμα έχει εξόδους:

/A	0011	0011	
/B	0101	0101	
/s1	1000	1000	
/ci1	0		
/co1	0		

Παρατηρούμε ότι  $C_{out} = 0$  και  $S = 1000 = 8$ , δηλαδή η πρόσθεση των δύο θετικών αριθμών έδωσε θετικό αριθμό και  $carry = 0$ .

- $-2 + (3)$

Για  $X = -A = 1110$ ,  $B = +3 = 0011$  και  $C_{in} = 0$  το κύκλωμα έχει εξόδους:

A	1110	1110	
B	0011	0011	
s1	0001	0001	
ci1	0		
co1	1		

Παρατηρούμε ότι  $C_{out} = 1$ ,  $S = 0001 = 1$ , δηλαδή η πρόσθεση του ενός αρνητικού αριθμού με ένα μεγαλύτερο θετικό έδωσε θετικό αριθμό και  $carry = 1$ .

- $-8 + (+7)$

Για  $A = -8 = 1000$ ,  $B = +7 = 0111$  και  $C_{in} = 0$  το κύκλωμα έχει εξόδους:

/A	1000	1000	
/B	0111	0111	
/s1	1111	1111	
/ci1	0		
/co1	0		

Παρατηρούμε ότι  $C_{out} = 0$ ,  $S = 1111 = 15$ , δηλαδή η πρόσθεση του  $-8$  ο οποίος έχει την ίδια τιμή με το  $8$  έδωσε λάθος αποτέλεσμα και  $carry = 0$ .