ΣΧΕΔΙΑΣΗ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
Εργασία 1

ΜΑΡΙΑ ΣΧΟΙΝΑΚΗ 3210191 p3210191@aueb.gr
ΧΡΗΣΤΟΣ ΣΤΑΜΟΥΛΟΣ 3210188 p3210188@aueb.gr

Πρόβλημα 1
(a)
F(x1,x2,x3,x4,x5) = Σm(0,2,6,10,11,15,16,26,27,30) + D(4,8,14,17,20,21,31)

Ο πίνακας των 5 μεταβλητών είναι:

Αρ.	x1	x2	x3	x4	x5	F
0	0	0	0	0	0	1
1	0	0	0	0	1	0
2	0	0	0	1	0	1
3	0	0	0	1	1	0
4	0	0	1	0	0	d
5	0	0	1	0	1	0
6	0	0	1	1	0	1
7	0	0	1	1	1	0
8	0	1	0	0	0	d
9	0	1	0	0	1	0
10	0	1	0	1	0	1
11	0	1	0	1	1	1
12	0	1	1	0	0	0
13	0	1	1	0	1	0
14	0	1	1	1	0	d
15	0	1	1	1	1	1
16	1	0	0	0	0	1
17	1	0	0	0	1	d
18	1	0	0	1	0	0
19	1	0	0	1	1	0
20	1	0	1	0	0	d
21	1	0	1	0	1	d
22	1	0	1	1	0	0
23	1	0	1	1	1	0
24	1	1	0	0	0	0
25	1	1	0	0	1	0
26	1	1	0	1	0	1
27	1	1	0	1	1	1
28	1	1	1	0	0	0
29	1	1	1	0	1	0
30	1	1	1	1	0	1
31	1	1	1	1	1	d

Το Σm εκφράζει τις θέσεις στον πίνακα που η συνάρτηση F δίνει αποτέλεσμα 1. Η D εκφράζει τους αδιάφορους όρους. Δηλαδή μπορούμε να τους δώσουμε ότι τιμή μας βολεύει περισσότερο γιατί οι συγκεκριμένες τιμές δεν επηρεάζουν το τελικό αποτέλεσμα. Έχοντας λοιπόν τον αναλυτικό πίνακα τιμών μπορούμε να σχεδιάσουμε τους πίνακες Karnaugh, αφού έχουμε 5 μεταβλητές και δεν αρκεί ένας. Κάθε κελί περιγράφεται από κάθε συνδυασμό των μεταβλητών. Π.χ το κελί 10 00 στον 1ο πίνακα (10 στήλη, 00 γραμμή) αντιπροσωπεύει τον συνδυασμό x1=1, x2=0, x3=0, x4=0, x5=0. Το POS εκφράζει τις μέγιστες δυνατές ομάδες μηδενικών που μπορούν να γίνουν στους πίνακες και ορίζει το ελάχιστο δυνατό κόστος ενώ το SOP εκφράζει τις μέγιστες δυνατές ομάδες άσσων που μπορούν να γίνουν στους πίνακες και ορίζει επίσης ελάχιστο δυνατό κόστος.(Οι ομάδες πρέπει να είναι πλήθους δύναμης του 2)

ΠΙΝΑΚΕΣ KARNAUGH

POS

x5 = 0

x2’ + x4

x3 x4 \ x1 x2	00	01	11	10
00	1_{0}	d_{8}	0_{24}	1_{16}
01	1_{2}	1_{10}	1_{26}	0_{18}
11	1_{6}	d_{14}	1_{30}	0_{22}
10	d_{4}	0_{12}	0_{28}	d_{20}

x1’ + x2 + x4’

x5 = 1

x3 x4 \ x1 x2	00	01	11	10
00	0_{1}	0_{9}	0_{25}	d_{17}
01	0_{3}	1_{11}	1_{27}	0_{19}
11	0_{7}	1_{15}	d_{31}	0_{23}
10	0_{5}	0_{13}	0_{29}	d_{21}

x2 + x5’

POS -> F(x1,x2,x3,x4,x5) = (x2’ + x4)(x2 + x5’)(x1’ + x2 + x4’)

Με κόστος:

1 and 3 εισόδων = 4
2 or 2 εισόδων = 6
1 or 3 εισόδων = 4

14

SOP

x1 x2 / x3 x4 — x5 = 0

x2’x4’x5’

x3x4 \ x1x2	00	01	11	10
00	1_{0}	d_{8}	0_{24}	1_{16}
01	1_{2}	1_{10}	1_{26}	0_{18}
11	1_{6}	d_{14}	1_{30}	0_{22}
10	d_{4}	0_{12}	0_{28}	d_{20}

x1’x2’x5’ x2x4

x1 x2 / x3 x4 — x5 = 1

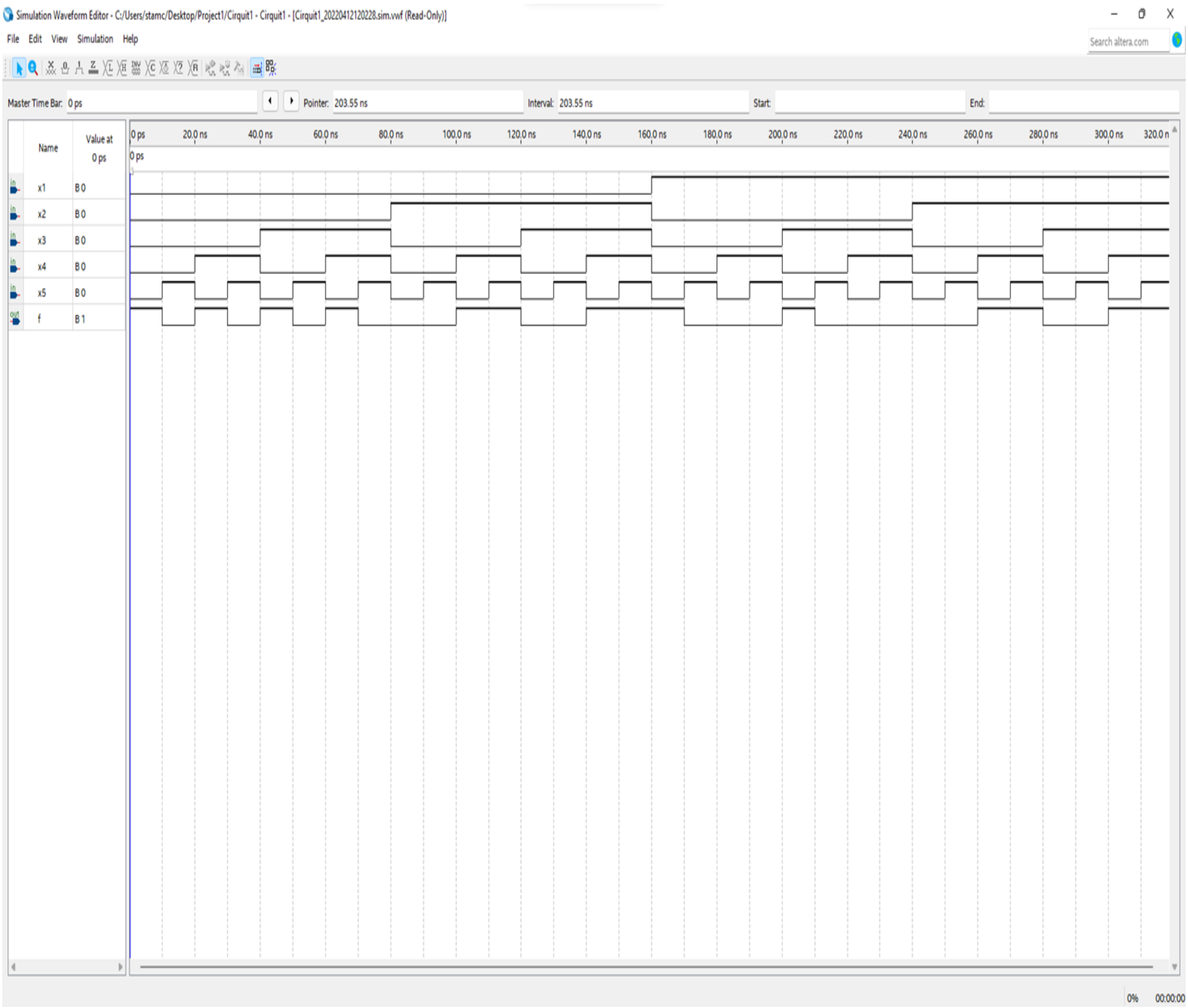
x3x4 \ x1x2	00	01	11	10
00	0_{1}	0_{9}	0_{25}	d_{17}
01	0_{3}	1_{11}	1_{27}	0_{19}
11	0_{7}	1_{15}	d_{31}	0_{23}
10	0_{5}	0_{13}	0_{29}	d_{21}

SOP -> F(x1,x2,x3,x4,x5) = x2’x4’x5’ + x1’x2’x5’ + x2x4
Με κόστος:

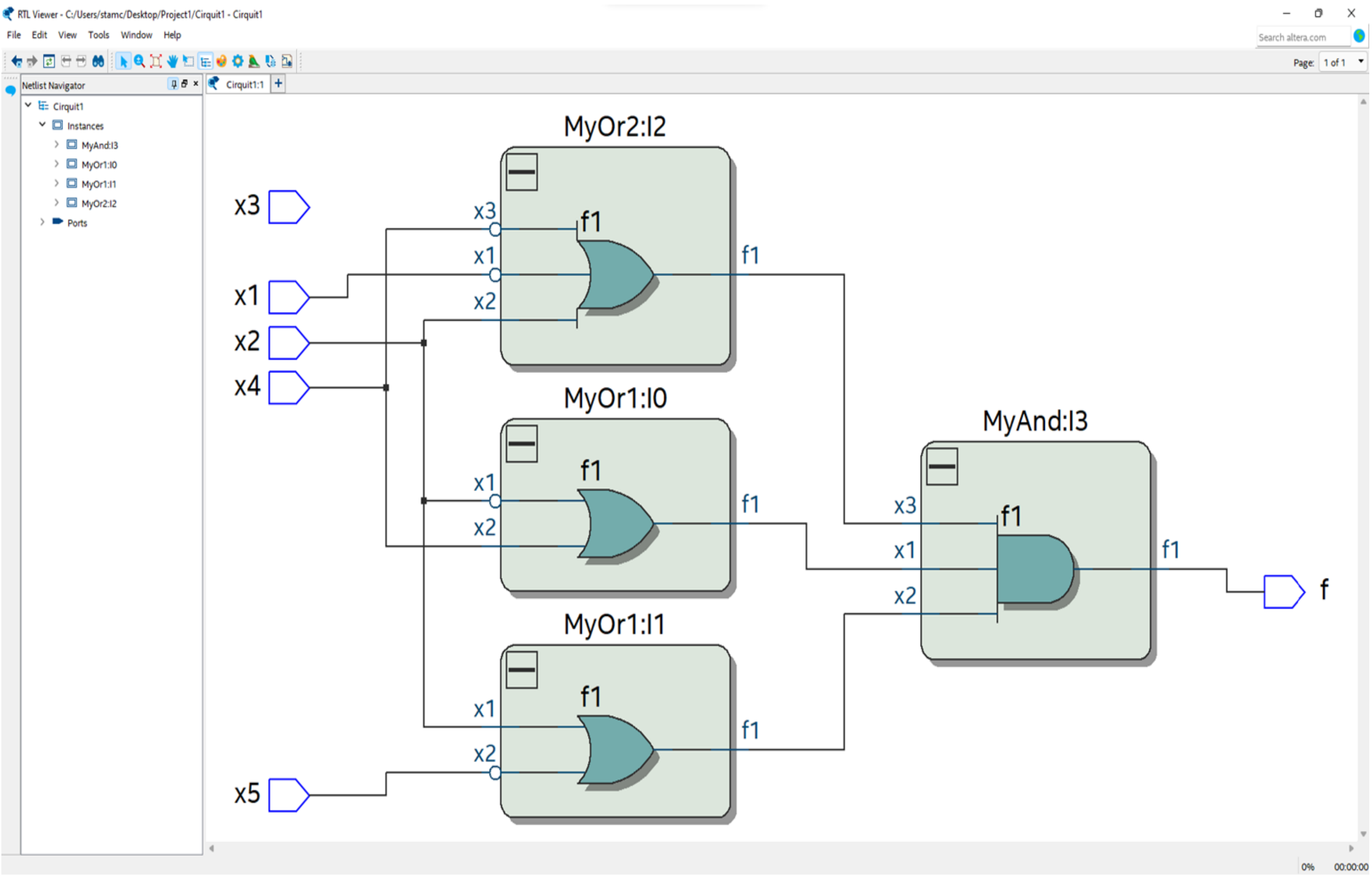
$$\left.\begin{array}{l} \text{1 or 3 εισόδων} = 4 \\ \text{1 and 2 εισόδων} = 3 \\ \text{2 and 3 εισόδων} = 8 \end{array}\right\} 15$$

Όπως βλέπουμε, το POS έχει μικρότερο κόστος από το SOP(14<15).
Το κόστος ουσιαστικά είναι το πόσες πύλες και είσοδοι χρησιμοποιήθηκαν για την διεξαγωγή του κυκλώματος. Οι υπολογιστικοί πόροι είναι πολύτιμοι και γιαυτό προσπαθούμε για μια μέγιστη απόδοση με όσο το δυνατόν χαμηλότερο κόστος σε πόρους. Άρα στην προσπάθεια διεξαγωγής ενός κυκλώματος που υλοποιεί την συγκεκριμένη συνάρτηση σοφό θα ήταν να χρησιμοποιήσουμε την μορφή POS καθώς έχει ίδια απόδοση με χαμηλότερο κόστος.

Μετά τον προγραμματισμό της συνάρτησης του POS στο περιβάλλον του quartus με την γλώσσα VHDL και έγκριση αυτής (compilation), έγινε λειτουργική προσομοίωση για την ορθότητά της (simulation) και εξήχθη η παρακάτω κυμματομορφή.



Όπως βλέπουμε η κυμματομορφή στην ευθεία του x1 παίρνει ανά 10 ns την τιμή που έχει ανά γραμμή στον πίνακα τιμών. Το ίδιο ισχύει και για τις x2, x3, x4, x5. Η καθεμία εναλλάσσεται με διαφορετικό τρόπο ώστε να επιτευχθούν όλες οι δυνατές διατάξεις αυτών των 5 μεταβλητών με τιμές 0 και 1, που είναι 2^5=32. Τελικά, η F έχει στο μήκος της κυμματομορφής τις τιμές που έχει και στον πίνακα τιμών, δηλαδή τις τιμές που θα έπρεπε να έχει ώστε να είναι λειτουργική σύμφωνα με την εκφώνηση της άσκησης. Άρα ο κώδικας ήταν επιτυχής.
Το διάγραμμα RTL είναι



Σχόλιο

Η μεταβλητή x3 είναι αδιάφορη καθώς μετά την διεξαγωγή του POS δεν λάμβανε μέρος μέσα στην συνάρτηση (έκφραση). Η άσκηση όμως ορίζει ότι υπάρχουν 5 μεταβλητές. Δηλαδή στην συνάρτηση

POS -> F(x1,x2,x3,x4,x5) = (x2’ + x4)(x2 + x5’)(x1’ + x2 + x4’)

η μεταβλητή x3 δεν υπάρχει. Υπάρχουν όμως οι υπόλοιπες 4 μεταβλητές x1, x2, x3, x4.

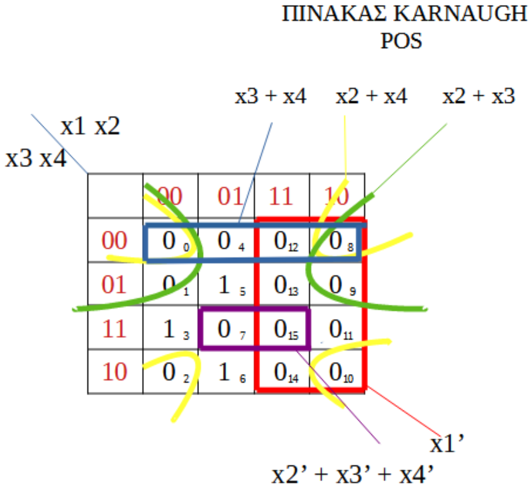
Πρόβλημα 2

(a)

Σύμφωνα με την εκφώνηση η f(x1,x2,x3,x4) έχει έξοδο 1 μόνο όταν το x1=0 και ακριβώς 2 από τις υπόλοιπες μεταβλητές (x2,x3,x4) είναι ίσες με 1. Υπάρχουν 2^4=16 δυνατές διατάξεις μεταξύ των μεταβλητών x1, x2, x3, x4. Αρχικά μηδενίζουμε όλους τους συνδυασμούς που έχουν x1=1. Μετά στους εναπομείναντες συνδυασμούς ψάχνουμε εκείνους που απαρτίζουν έναν συνδυασμό που μετά το x1=0 έχουν 1 μηδενικό και 2 άσσους. Αυτό γίνεται με 3 δυνατούς τρόπους. Τον 0011, 0101, 0110.

Ο πίνακας των 4 μεταβλητών είναι:

Αρ.	x1	x2	x3	x4	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0



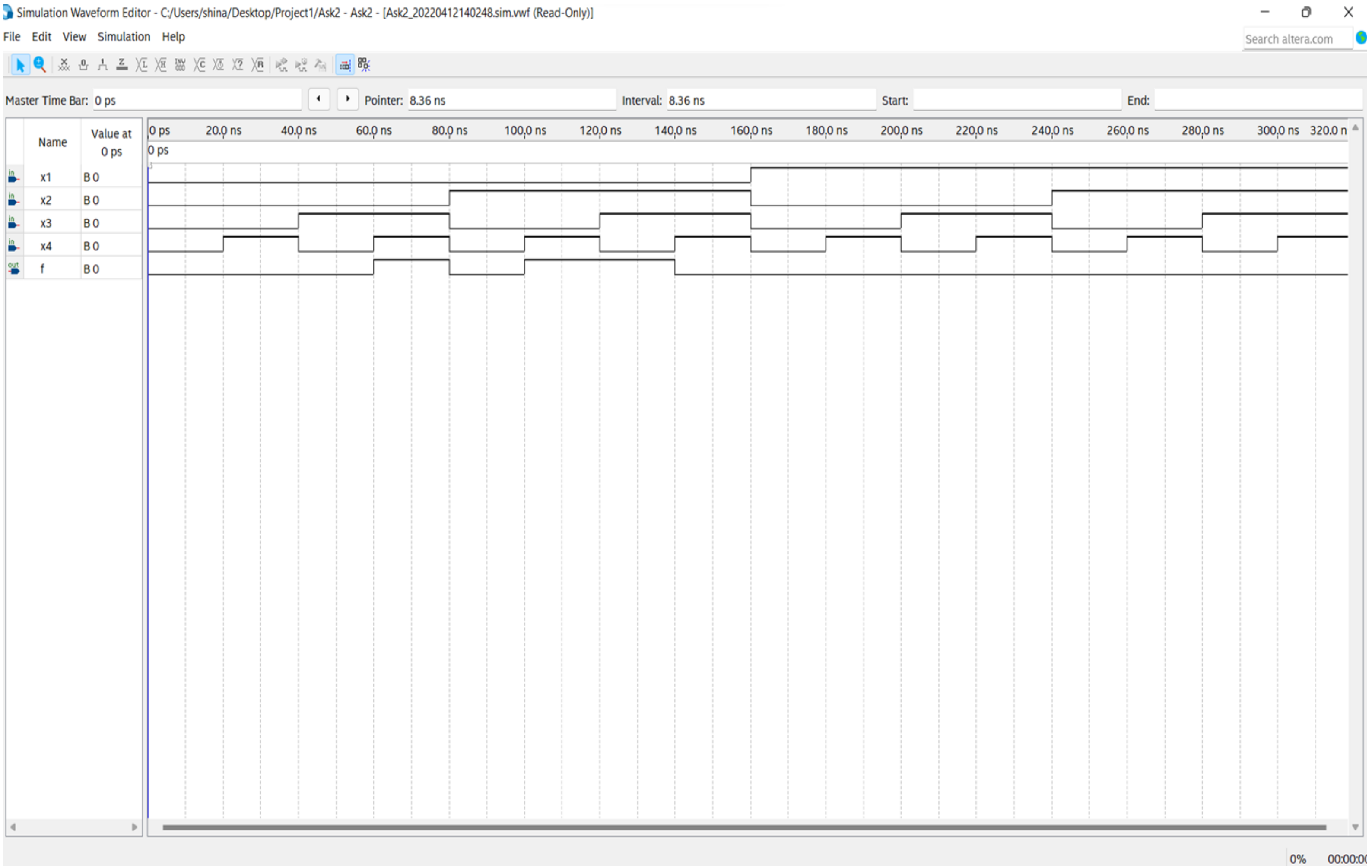
x3 x4 \ x1 x2	00	01	11	10
00	0_0	0_4	0_{12}	0_8
01	0_1	1_5	0_{13}	0_9
11	1_3	0_7	0_{15}	0_{11}
10	0_2	1_6	0_{14}	0_{10}

POS -> f(x1,x2,x3,x4) = (x1’)(x2 + x3)(x2 + x4)(x3 + x4)(x2’ + x3’ + x4’)

Με κόστος:

$$\left.\begin{array}{l}3 \text{ or } 2 \text{ εισόδων} = 9\\ 1 \text{ or } 3 \text{ εισόδων} = 4\\ 1 \text{ and } 5 \text{ εισόδων} = 6\end{array}\right\} \quad 19$$

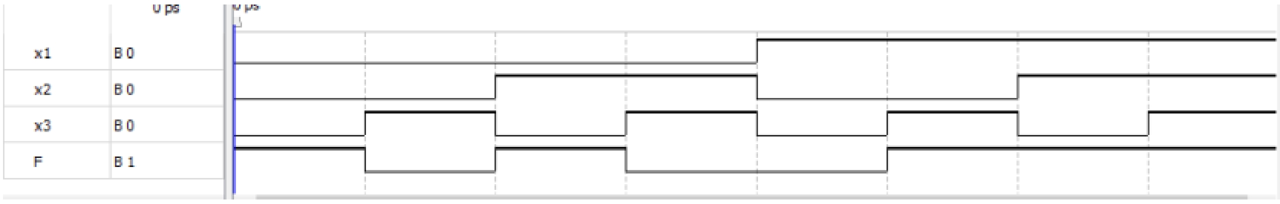
Μετά τον προγραμματισμό της συνάρτησης του POS στο περιβάλλον του quartus με block diagram και έγκριση αυτής (compilation), έγινε λειτουργική προσομοίωση για την ορθότητά της (simulation) και εξήχθη η παρακάτω κυμματομορφή.



Όπως βλέπουμε η κυμματομορφή στην ευθεία του x1 παίρνει ανά 20 ns την τιμή που έχει ανά γραμμή στον πίνακα τιμών. Το ίδιο ισχύει και για τις x2, x3, x4. Η καθεμία εναλλάσσεται με διαφορετικό τρόπο ώστε να επιτευχθούν όλες οι δυνατές διατάξεις αυτών των 4 μεταβλητών με τιμές 0 και 1. Τελικά, η f έχει στο μήκος της κυμματομορφής τις τιμές που έχει και στον πίνακα τιμών, δηλαδή τις τιμές που θα έπρεπε να έχει ώστε να είναι λειτουργική σύμφωνα με την εκφώνηση της άσκησης. Άρα η σχεδίαση ήταν επιτυχής.

Πρόβλημα 3

(a)

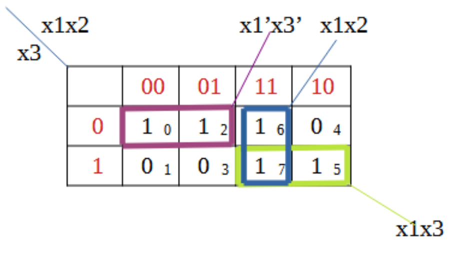


Όπως βλέπουμε στην κυμματομορφή της εκφώνησης υπάρχει μια συνάρτηση F η οποία αποτελείται από 3 μεταβλητές x1, x2, x3 οι οποίες αλλάζουν διαδοχικά τις τιμές 1 και 0 ώστε να μπορούν να δημιουργηθούν όλες οι δυνατές διατάξεις μεταξύ αυτών των 3 μεταβλητών που είναι $2^3=8$. Η F σε όλο το μήκος της κυμματομορφής αλλάζει τις τιμές 1 και 0. Όταν οι μεταβλητές x1=0, x2=0, x3=0 η F=1. Αντίστοιχα θα γράψουμε και τις υπόλοιπες τιμές της F σύμφωνα με τις διατάξεις των 3 μεταβλητών στο πίνακα τιμών.

Ο πίνακας των 3 μεταβλητών είναι:

Αρ.	x1	x2	x3	F
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

ΠΙΝΑΚΑΣ KARNAUGH
SOP



SOP -> f(x1,x2,x3) = x1x2 + x1x3 + x1’x3’

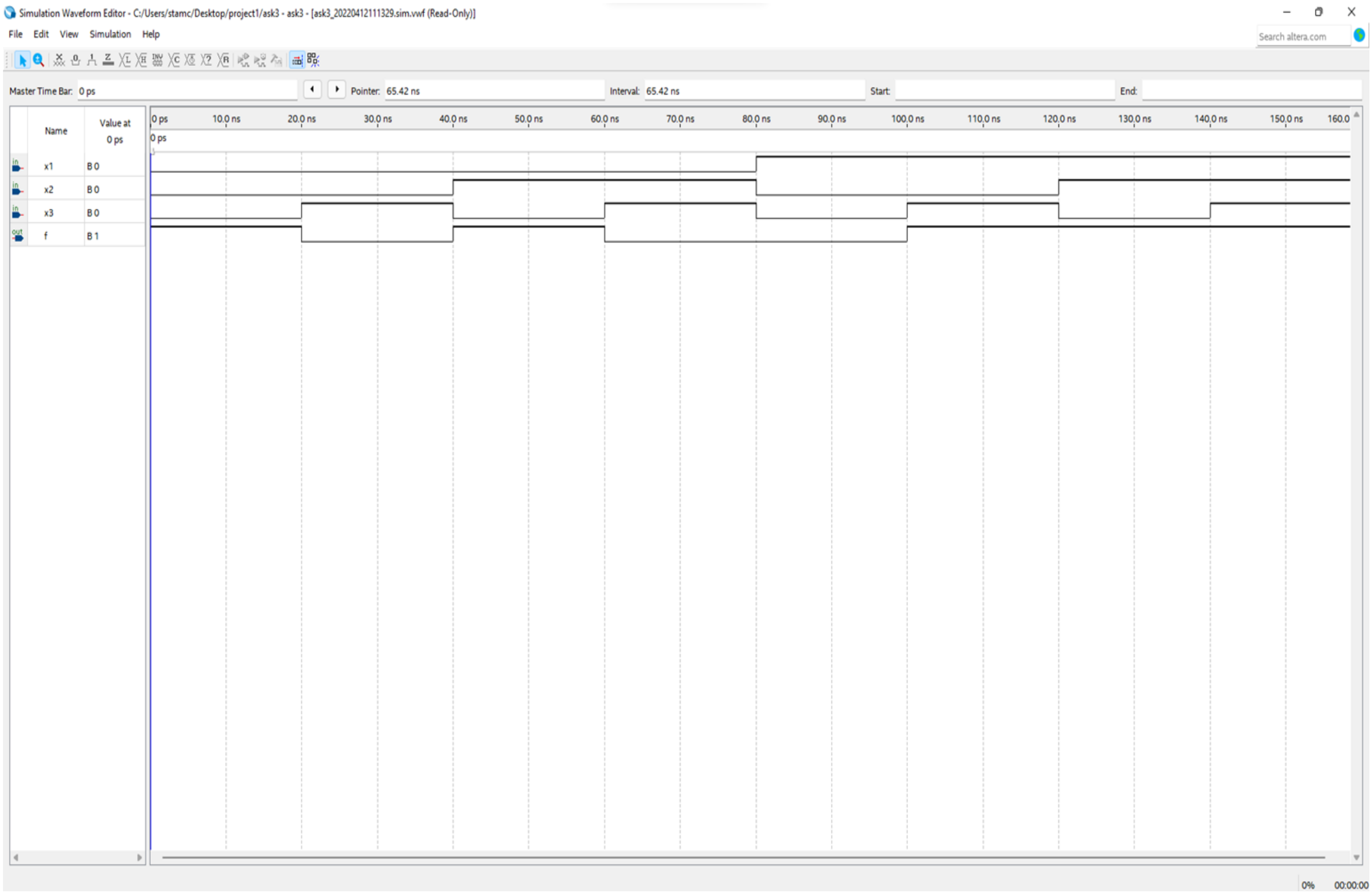
Με κόστος:

3 or 2 εισόδων = 9

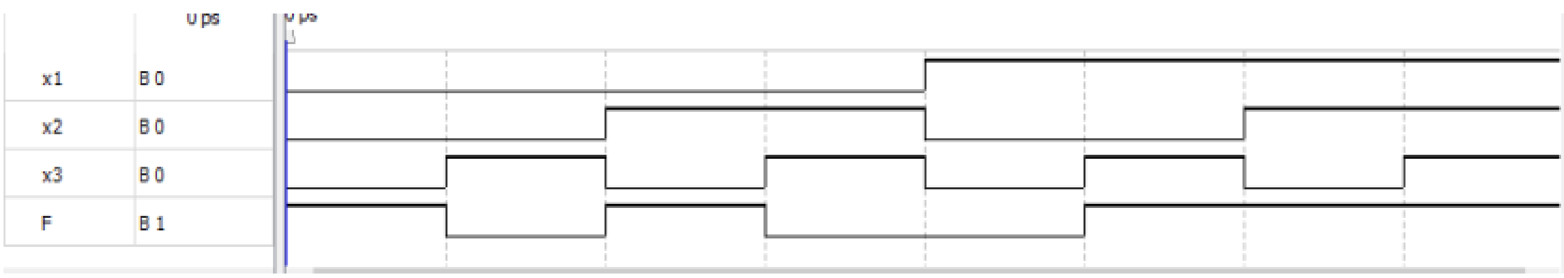
1 and 3 εισόδων = 4

13

Μετά τον προγραμματισμό της συνάρτησης του SOP στο περιβάλλον του quartus με την γλώσσα VHDL και έγκριση αυτής (compilation), έγινε λειτουργική προσομοίωση για την ορθότητά της (simulation) και εξήχθη η παρακάτω κυμματομορφή.



Η κυμματομορφή της εκφώνησης είναι ίδια με αυτήν που διεξήχθη από τον προγραμματισμό μας σε VHDL. Άρα ο κώδικας ήταν επιτυχής.



Το διάγραμμα RTL είναι