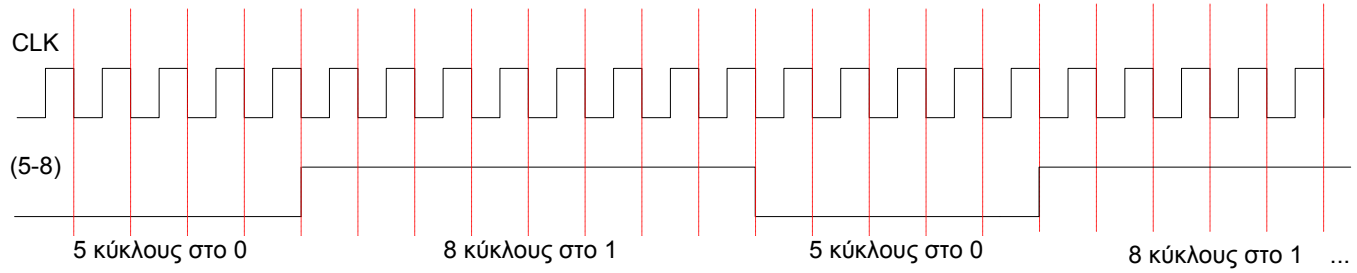
ΕΞΕΤΑΣΕΙΣ ΣΕΠΤΕΜΒΡΙΟΥ 2008

Θέμα 1 **[2,5 μονάδες – 35΄]**

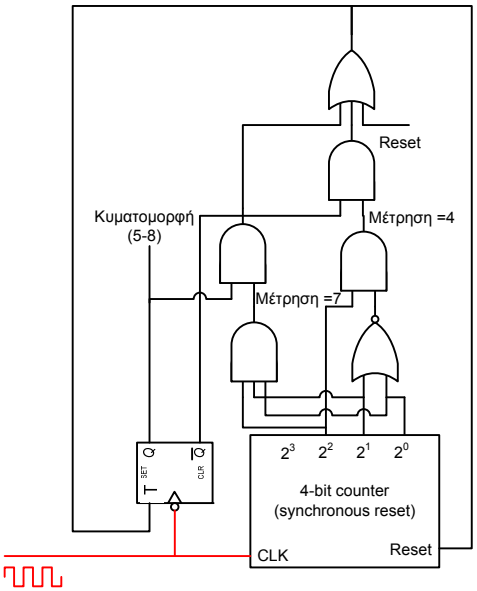
Σας δίνεται ως είσοδος η κυματομορφή ρολογιού CLK του παρακάτω σχήματος. Χρησιμοποιώντας ως ακολουθιακά στοιχεία ένα μετρητή 4 δυαδικών ψηφίων και ένα FF της επιλογής σας, προτείνετε σύγχρονο ακολουθιακό κύκλωμα για τη δημιουργία της κυματομορφής (5-8).



Είναι προφανές ότι θέλουμε ένα ακολουθιακό στοιχείο το οποίο θα αλλάζει κατάσταση κάθε 5 ή κάθε 8 κύκλους, ανάλογα με την έξοδο του μετρητή. Αν συνεπώς χρησιμοποιήσουμε έναν μετρητή με σύγχρονο σήμα καθαρισμού, θα πρέπει να τον αφήσουμε να μετράει από το 0 έως το 4, να τον μηδενίζουμε, να τον αφήσουμε να μετράει από το 0 έως το 7, να τον μηδενίζουμε και πάλι από την αρχή. Καθώς στο θέλουμε στο πρώτο διάστημα η έξοδός μας να είναι 0 και στο δεύτερο 1, καταλήγουμε στην εξής συνθήκη μηδενισμού για το μετρητή μας :

((Έξοδος ακολουθιακού στοιχείου == 0) and (μέτρηση == 4)) or
((Έξοδος ακολουθιακού στοιχείου == 1) and (μέτρηση ==7)) or reset

Αν συνεπώς επιλέξουμε ένα TFF για ακολουθιακό στοιχείο, παίρνουμε το ακόλουθο κύκλωμα :



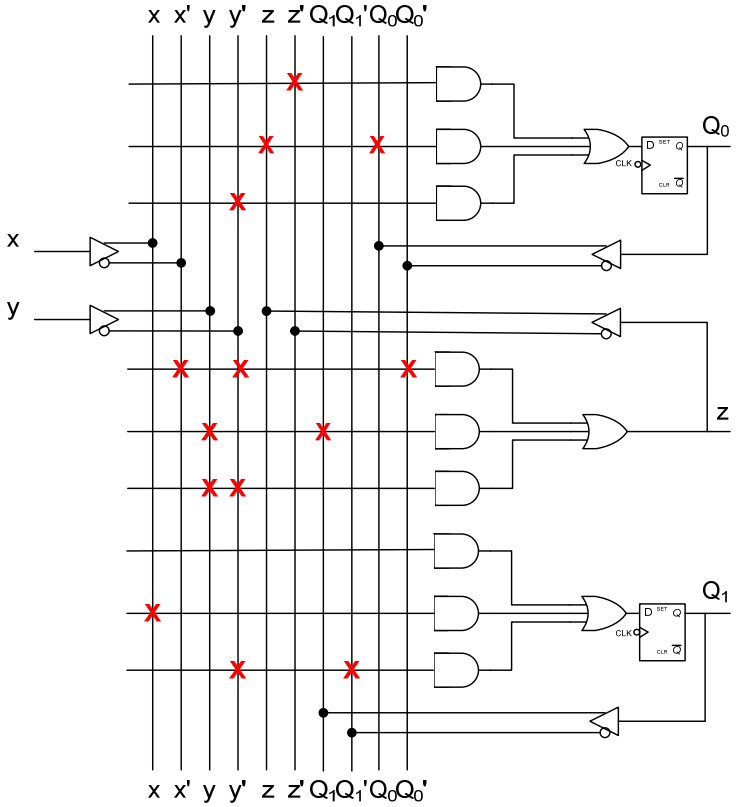
Μπορείτε να πιστοποιήσετε την ορθή λειτουργία του παραπάνω κυκλώματος χρησιμοποιώντας την ακόλουθη περιγραφή του :

```
module five_eight (clk, reset, f_e);
 input clk, reset ;
 output f_e;
 reg[3:0] count;
 reg      myff;
 wire     tog;

assign tog = ((count == 4'h4) & ~myff) ? 1'b1 :
          ((count == 4'h7) &  myff) ? 1'b1 :
                    (reset) ? 1'b1 : 1'b0;

 always @(negedge clk)
  if (reset) myff <= 1'b0;
   else if (tog) myff <= ~myff;
 always @(negedge clk)
  if (tog) count <= 4'h0;
   else count <= count +1;
 assign f_e = myff;
endmodule
```

```
module test();
 reg reset;
 reg clk;
 wire f_e;

 initial
  begin
    clk = 1;
    reset = 0;
  #2 reset = 1;
  #5 reset = 0;
  end

 always #5 clk = ~clk;

 five_eight i0 (clk, reset, f_e);
endmodule
```

Θέμα 2 **[2,5 μονάδες – 37΄]**

Χρησιμοποιώντας T FF ως ακολουθιακά στοιχεία, δώστε κύκλωμα ισοδύναμο με το εμφωλευμένο εντός της PAL, που απεικονίζεται στο παρακάτω σχηματικό. Τα X αναπαριστούν τους συνδέσμους της PAL που έχουν μείνει άθικτοι κατά τον προγραμματισμό της.



Η παρατήρηση του κυκλώματος αποκαλύπτει ότι πρόκειται για ΣΑΚ με 2 εισόδους (x και y) και έξοδο z. Καταγράφουμε τις εξισώσεις εξόδου και επόμενης κατάστασης και έχουμε :

$z = x'y'Q'_0 + yQ_1 + \cancel{yy'}$ (1)

$Q_0(t+1) = D_0(t+1) = z' + zQ_0 + y'$, η οποία λόγω της (1) γίνεται :

$Q_0(t+1) = D_0(t+1) = (x + y + Q_0)\,(y' + Q'_1) + yQ_1Q_0 + y' = \cancel{xy'} + xQ'_1 + \cancel{y}Q'_1 + \cancel{y'Q_0} + Q'_1Q_0 + \cancel{y}Q_1Q_0 + y' =$

$y' + \cancel{x}Q'_1 + Q'_1 + Q'_1Q_0 + Q_1Q_0 = y' + Q'_1 + Q_0$

$Q_1(t+1) = D_1(t+1) = x + y'Q'_1$

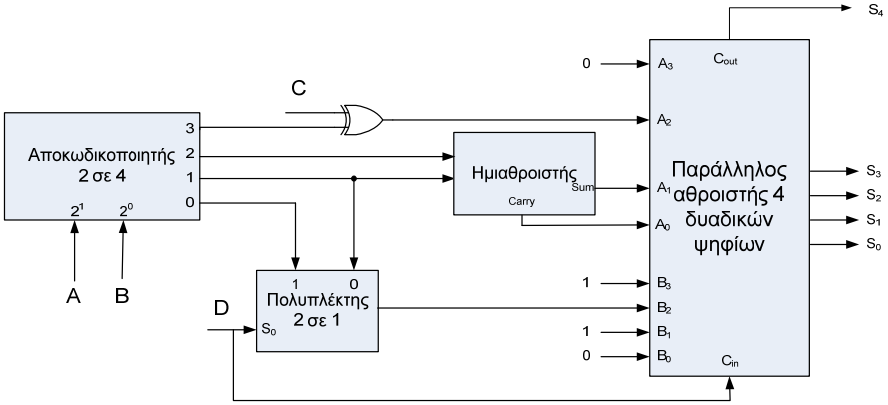
Με βάση τις παραπάνω εξισώσεις καταστρώνουμε το πίνακα μεταβάσεων του κυκλώματος και τις εισόδους των T-FF στο ισοδύναμο κύκλωμα. Προφανώς η υλοποίηση της εξόδου δεν αλλάζει στο ισοδύναμο κύκλωμα.

x	y	Q_1	Q_0	$Q_1(t+1)$	$Q_0(t+1)$	T_1	T_0
0	0	0	0	1	1	1	1
0	0	0	1	1	1	1	0
0	0	1	0	0	1	1	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	0	0
0	1	1	0	0	0	1	0
0	1	1	1	0	1	1	0
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	1	1
1	1	0	1	1	1	1	0
1	1	1	0	1	0	0	0
1	1	1	1	1	1	0	0

Από το πίνακα είναι $T_1\,(x, y, Q_1, Q_0) = \Sigma\,(0, 1, 2, 3, 6, 7, 8, 9, 12, 13)$ και $T_2\,(x, y, Q_1, Q_0) = \Sigma\,(0, 2, 4, 8, 10, 12)$. Η απλοποίηση των παραπάνω δίνει $T_1 = x'y' + (x \oplus Q_1)$ και $T_2 = Q'_0(Q'_1 + y')$, που μαζί με την (1) μας δίνουν τη ζητούμενη υλοποίηση.

Θέμα 3 **[2,5 μονάδες – 30΄]**

Δίδεται το κύκλωμα :



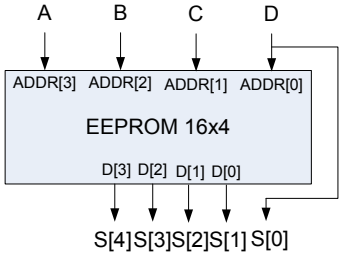
Ζητείται να δώσετε μια ισοδύναμη υλοποίησή του, χρησιμοποιώντας μια όσο το δυνατόν μικρότερη EEPROM. Η απάντησή σας θα πρέπει **οπωσδήποτε** να περιλαμβάνει :

- Το μέγεθος της EEPROM που θα χρειαστείτε.
- Τα περιεχόμενα κάθε θέσης της EEPROM.
- Ενα σχηματικό διάγραμμα, που θα δείχνει πως θα συνδεθεί κάθε γραμμή διευθύνσεων της EEPROM και σε ποιες γραμμές δεδομένων της, θα προκύψουν οι ζητούμενες συναρτήσεις.

Παρατηρούμε πως στο κύκλωμά μας υλοποιούνται 5 συναρτήσεις (S_4-S_0) των 4 μεταβλητών. Καταστρώνουμε τον πίνακα αληθείας τους.

A	B	C	D	$A_3A_2A_1A_0$	$B_3B_2B_1B_0$	C_{in}	$S_4S_3S_2S_1S_0$
0	0	0	0	0000	1010	0	01010
0	0	0	1	0000	1110	1	01111
0	0	1	0	0100	1010	0	01110
0	0	1	1	0100	1110	1	10011
0	1	0	0	0010	1110	0	10000
0	1	0	1	0010	1010	1	01101
0	1	1	0	0110	1110	0	10100
0	1	1	1	0110	1010	1	10001
1	0	0	0	0010	1010	0	01100
1	0	0	1	0010	1010	1	01101
1	0	1	0	0110	1010	0	10000
1	0	1	1	0110	1010	1	10001
1	1	0	0	0100	1010	0	01110
1	1	0	1	0100	1010	1	01111
1	1	1	0	0000	1010	0	01010
1	1	1	1	0000	1010	1	01011

Από τον πίνακα προκύπτει ότι S_0 = D και συνεπώς δεν απαιτείται υλοποίηση με μνήμη αυτής της συνάρτησης. Για την υλοποίηση των υπολοίπων συναρτήσεων θα χρειαστούμε μνήμη 16 θέσεων των 4 δυαδικών ψηφίων ανά θέση. Σε κάθε θέση μνήμης αποθηκεύονται τα 4 πιο σημαντικά ψηφία της τελευταίας στήλης του πιο πάνω πίνακα. Για την υλοποίηση των συναρτήσεων, οι μεταβλητές συνδέονται στις γραμμές διευθύνσεων ανάλογα με τη σημαντικότητά τους στο παραπάνω πίνακα και οι συναρτήσεις προκύπτουν από τις γραμμές δεδομένων. Το ζητούμενο διάγραμμα είναι το ακόλουθο :



Θέμα 4 **[2,5 (= 1,5 + 1,0) μονάδες – 30΄]**

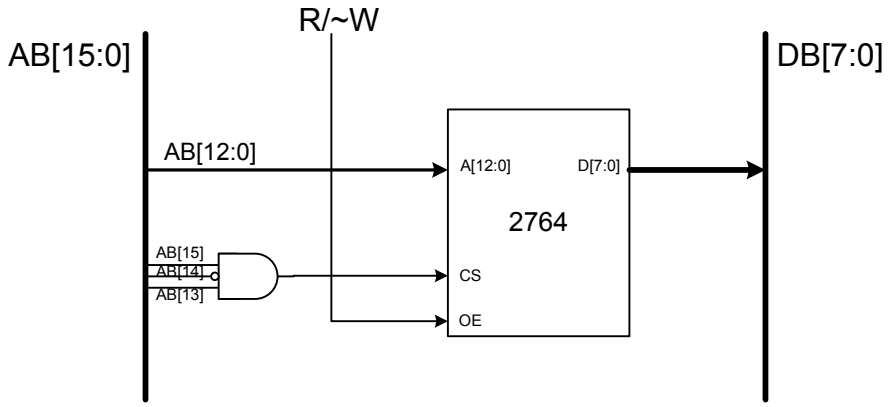
Σας δίδεται ένα ολοκληρωμένο ROM 2764. Το ολοκληρωμένο αυτό παρέχει μνήμη 64 Kbit, με οργάνωση 8K (8192) θέσεις και 8 δυαδικά ψηφία ανά θέση.

- Θεωρείστε επεξεργαστή με αρτηρίες διευθύνσεων και δεδομένων των 16 και 8 δυαδικών ψηφίων αντίστοιχα. Δώστε σε λογικό διάγραμμα την απαραίτητη διασύνδεση ώστε το 2764 να καλύψει το χώρο διευθύνσεων που ξεκινά από τη διεύθυνση μνήμης $A000_{HEX}$.
- Χρησιμοποιώντας επιπλέον SSI / MSI δείξτε πως μπορεί το 2764 να μετατραπεί σε μια ROM με οργάνωση 64K (65536) θέσεων με 1 δυαδικό ψηφίο ανά θέση.

Το ολοκληρωμένο μας έχει 8K θέσεις = 2^{13} θέσεις και συνεπώς χρειάζεται 13 δυαδικά ψηφία διευθύνσεων για τη προσπέλασή του. Συνεπώς αν το τοποθετήσουμε στη περιοχή μνήμης που ξεκινά από τη θέση A000, θα καλύπτει το ακόλουθο εύρος διευθύνσεων :

$A000_{16}$ = 1010 0000 0000 0000
$A001_{16}$ = 1010 0000 0000 0001
...
$BFFF_{16}$ = 1011 1111 1111 1111

Θα πρέπει συνεπώς να απαντά σε κάθε προσπέλαση που έχει διεύθυνση της οποίας τα 3 πρώτα ψηφία είναι 101_2. Μπορούμε συνεπώς να χρησιμοποιήσουμε αυτά τα ψηφία για την επιλογή του ολοκληρωμένου και όλα τα υπόλοιπα για τη διευθυνσιοδότησή του. Ετσι το ζητούμενο λογικό διάγραμμα είναι το ακόλουθο :



Μια μνήμη 65536 = 2^{16} θέσεων θα δέχεται μια διεύθυνση των 16 δυαδικών ψηφίων. Μπορούμε να χρησιμοποιήσουμε τα 13 από αυτά (συνήθως τα περισσότερο σημαντικά) για τη προσπέλαση μιας 8άδας ψηφίων που είναι αποθηκευμένα εντός του 2764, συνδέοντάς τα στις γραμμές διευθύνσεών του. Τα υπόλοιπα 3 θα πρέπει να τα χρησιμοποιήσουμε ώστε να επιλέξουμε το ένα από τα 8 δυαδικά ψηφία που εμφανίζονται στις γραμμές δεδομένων του. Αυτό μπορούμε να το επιτύχουμε χρησιμοποιώντας έναν πολυπλέκτη 8 σε 1 με 3 γραμμές επιλογής. Ενδεικτικό λογικό διάγραμμα είναι το παρακάτω :