

ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑΤΑ Ι

Λύσεις θεμάτων εξεταστικής περιόδου Ιανουαρίου – Φεβρουαρίου 2015

ΘΕΜΑ 1^ο (4,0 μονάδες)

Να σχεδιάσετε λογικό κύκλωμα που πολλαπλασιάζει δύο διψήφιους δυαδικούς αριθμούς, $A = A_1A_0$ και $B = B_1B_0$, και παράγει στην έξοδό του το αριθμητικό γινόμενο των δύο αριθμών σε δυαδική μορφή:

α. Με δύο ημιαθροιστές και λογικές πύλες.

β. Με μία κατάλληλη προγραμματιζόμενη ROM. Να προσδιορίσετε το μέγεθός της (αριθμό λέξεων, αριθμό bits ανά λέξη, μέγεθος σε bytes) και να δείξετε την εσωτερική δομή της.

Λύση

α. Ο πολλαπλασιασμός δυαδικών αριθμών εκτελείται με τον ίδιο ακριβώς τρόπο που εκτελείται και ο πολλαπλασιασμός δεκαδικών αριθμών. Τα ψηφία του πολλαπλασιαστέου πολλαπλασιάζονται με κάθε ψηφίο του πολλαπλασιαστή ξεκινώντας από το ελάχιστο σημαντικό ψηφίο του πολλαπλασιαστή. Με τον τρόπο αυτό σχηματίζονται τα μερικά γινόμενα, δηλαδή τα αριθμητικά γινόμενα μεταξύ δύο ψηφίων, ενός από τον πολλαπλασιαστέο και ενός από τον πολλαπλασιαστή κάθε φορά. Στην περίπτωσή μας, τα μερικά γινόμενα που σχηματίζονται είναι $A_0 \times B_0$, $A_0 \times B_1$, $A_1 \times B_0$ και $A_1 \times B_1$, τα οποία διατάσσονται όπως φαίνεται στο παρακάτω σχήμα, σύμφωνα με τον αλγόριθμο του πολλαπλασιασμού. Το αποτέλεσμα του πολλαπλασιασμού (το αριθμητικό γινόμενο), $P_3P_2P_1P_0$, προκύπτει από το αριθμητικό άθροισμα των μερικών γινομένων, λαμβάνοντας υπόψη και τυχόν κρατούμενα που δημιουργούνται. Επομένως,

- το P_0 είναι ίσο με το μερικό γινόμενο $A_0 \times B_0$,
- το P_1 προκύπτει ως το άθροισμα των μερικών γινομένων $A_0 \times B_1$ και $A_1 \times B_0$,
- το P_2 είναι ίσο με το άθροισμα του μερικού γινομένου $A_1 \times B_1$ και του κρατούμενου που τυχόν προκύπτει από το άθροισμα των $A_0 \times B_1$ και $A_1 \times B_0$ και, τέλος,
- το P_3 ισούται με το κρατούμενο που τυχόν προκύπτει από την αμέσως προηγούμενη άθροιση.

Ο πολλαπλασιασμός (αριθμητικό γινόμενο) δύο δυαδικών ψηφίων A_i και B_j δίνει 1 μόνο όταν και τα δύο ψηφία είναι 1, αλλιώς δίνει 0. Το αποτέλεσμα αυτό, όμως, είναι το ίδιο με εκείνο που δίνει η λογική πράξη AND μεταξύ δύο δυαδικών ψηφίων:

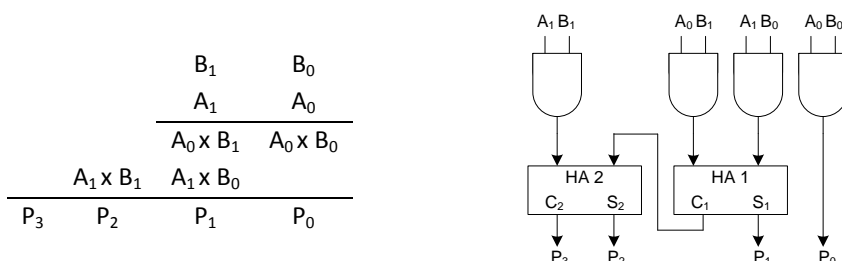
A_i	B_j	$A_i \times B_j$	$A_i B_j$
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

Επομένως, χρειαζόμαστε τέσσερις πύλες AND δύο εισόδων για να σχηματίσουμε τα μερικά γινόμενα και συγκεκριμένα: $A_0 \times B_0 = A_0 B_0$, $A_0 \times B_1 = A_0 B_1$, $A_1 \times B_0 = A_1 B_0$ και $A_1 \times B_1 = A_1 B_1$, αντίστοιχα.

Για την άθροιση των μερικών γινομένων $A_0 \times B_1$ και $A_1 \times B_0$ απαιτείται ένας ημιαθροιστής, ο οποίος στην έξοδο του αθροίσματος S_1 παράγει το ψηφίο P_1 του αποτελέσματος του πολλαπλασιασμού και στην έξοδο C_1 παράγει ένα κρατούμενο.

Ακολουθώντας, χρειαζόμαστε έναν δεύτερο ημιαθροιστή για την πρόσθεση του μερικού γινομένου $A_1 \times B_1$ και του κρατούμενου C_1 που προκύπτει από την αμέσως προηγούμενη άθροιση των $A_0 \times B_1$ και $A_1 \times B_0$. Η έξοδος S_2 του δεύτερου ημιαθροιστή μας δίνει το ψηφίο P_2 του αποτελέσματος του πολλαπλασιασμού και στην έξοδο C_2 παράγεται ένα κρατούμενο που μας δίνει το ψηφίο P_3 του αποτελέσματος του πολλαπλασιασμού.

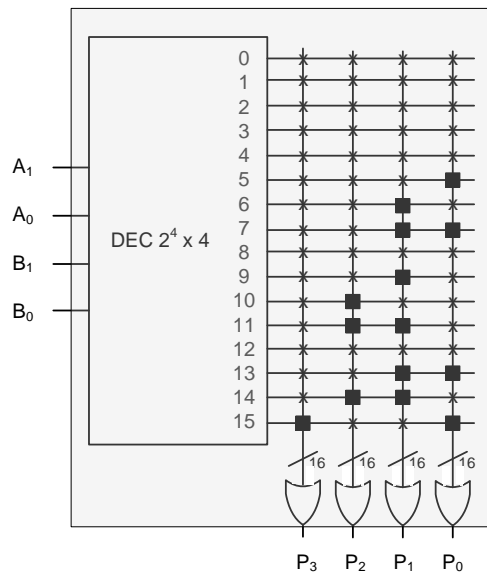
Με βάση τα παραπάνω, σχεδιάζουμε το λογικό κύκλωμα που φαίνεται στο παρακάτω σχήμα.



β. Ο πίνακας αλήθειας που περιγράφει την οργάνωση της ROM φαίνεται στο παρακάτω σχήμα. Έχουμε τέσσερις εισόδους, $A_1A_0 B_1B_0$, και απαιτούνται τέσσερις έξοδοι, $P_3P_2P_1P_0$, αφού το μέγιστο αριθμητικό γινόμενο που μπορεί να παραχθεί είναι $3_{10} (11_2) \times 3_{10} (11_2) = 9_{10} (1001_2)$. Επομένως, η κατάλληλη ROM θα έχει 4 εισόδους διεύθυνσης ($k = 4$) και 4 εξόδους δεδομένων ($n = 4$). Η ROM θα αποτελείται από έναν αποκωδικοποιητή με $k = 4$ εισόδους και $2^k = 2^4 = 16$ εξόδους, καθώς και 4 πύλες OR, μία για κάθε έξοδο της ROM. Κάθε έξοδος του αποκωδικοποιητή αντιπροσωπεύει και μια διεύθυνση μνήμης (λέξη). Επομένως θα έχουμε $2^k = 2^4 = 16$ λέξεις των 4 bits, αφού οι έξοδοι δεδομένων της ROM είναι $n = 4$ bits. Το μέγεθος της ROM δίνεται από τη σχέση $2^k \times n = 2^4 \times 4 = 16 \times 4 = 64$ bits = 8 bytes (1 byte = 8 bits).

Οι 16 έξοδοι του αποκωδικοποιητή είναι συνδεδεμένες με καθεμιά από τις 4 πύλες OR που παράγουν τις 4 εξόδους της ROM. Θα έχουμε $16 \times 4 = 64$ εσωτερικές συνδέσεις, οι οποίες είναι “προγραμματιζόμενες”. Η διαδικασία προγραμματισμού της ROM αντιστοιχεί στο “κάψιμο” των συνδέσεων σύμφωνα με τον πίνακα αλήθειας. Στο λογικό κύκλωμα του παρακάτω σχήματος, τα σκούρα τετράγωνα συμβολίζουν τις υφιστάμενες συνδέσεις, ενώ τα ‘x’ συμβολίζουν τις συνδέσεις που έχουν “καεί”. Οι υφιστάμενες συνδέσεις αντιστοιχούν στα λογικά 1 του πίνακα αλήθειας, ενώ οι καμένες συνδέσεις στα λογικά 0.

Πίνακας Αλήθειας			
A_1A_0	B_1B_0	$P_3P_2P_1P_0$	
0 0	0 0	0 0	0 0
0 0	0 1	0 0	0 0
0 0	1 0	0 0	0 0
0 0	1 1	0 0	0 0
0 1	0 0	0 0	0 0
0 1	0 1	0 0	0 1
0 1	1 0	0 0	1 0
0 1	1 1	0 0	1 1
1 0	0 0	0 0	0 0
1 0	0 1	0 0	1 0
1 0	1 0	0 1	0 0
1 0	1 1	0 1	1 0
1 1	0 0	0 0	0 0
1 1	0 1	0 0	1 1
1 1	1 0	0 1	1 0
1 1	1 1	1 0	0 1



Αριθμός λέξεων: $2^4 = 16$

Αριθμός bits ανά λέξη: 4

Μέγεθος μνήμης:

16 λέξεις x 4 bits/λέξη = 64 bits = 8 bytes

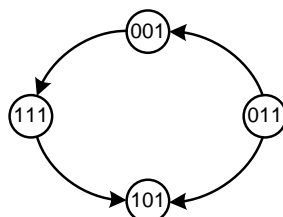
ΘΕΜΑ 2^ο (4,0 μονάδες)

α. Να σχεδιάσετε με JK flip-flop σύγχρονο κυκλικό δυαδικό μετρητή φθίνουσας μέτρησης, που απαριθμεί τις καταστάσεις 7, 5, 3, 1. (3,0 μον.)

β. Εάν το σύστημα βρεθεί στην κατάσταση $Q_2Q_1Q_0 = 110$, να προσδιορίσετε σε ποια κατάσταση θα μεταβεί μετά από έναν ωρολογιακό παλμό. (1,0 μον.)

Λύση:

α. Το Διάγραμμα Καταστάσεων του ζητούμενου μετρητή είναι το ακόλουθο:



Με βάση τον πίνακα διέγερσης του JK flip-flop συμπληρώνουμε τον πίνακα (μετάβασης) καταστάσεων:

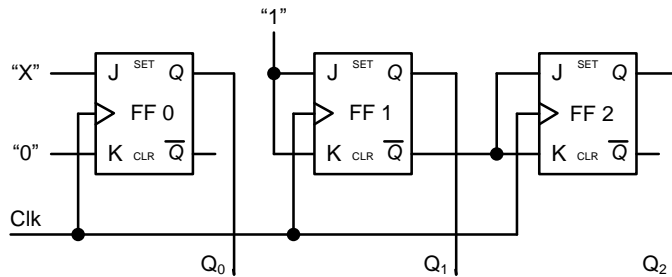
Παρούσα Κατάσταση $Q_2 Q_1 Q_0$	Επόμενη Κατάσταση $Q_2^+ Q_1^+ Q_0^+$	Είσοδοι FF		
		$J_2 K_2$	$J_1 K_1$	$J_0 K_0$
0 0 1	1 1 1	1 X	1 X	X 0
0 1 1	0 0 1	0 X	X 1	X 0
1 0 1	0 1 1	X 1	1 X	X 0
1 1 1	1 0 1	X 0	X 1	X 0

Οι μη χρησιμοποιούμενες καταστάσεις (000, 010, 100 και 110) είναι αδιάφορες καταστάσεις (X).

Από τον πίνακα καταστάσεων προσδιορίζουμε τις συναρτήσεις των εισόδων των Flip-Flop, σε απλοποιημένη μορφή.

Είναι προφανές ότι $J_2 = Q_1'$, $K_2 = Q_1'$, $J_1 = 1$, $K_1 = 1$, $J_0 = X$ και $K_0 = 0$.

Το κύκλωμα του μετρητή είναι το ακόλουθο:



β. Αφού $J_0 = X$, θα πρέπει να εξετάσουμε τη συμπεριφορά του συστήματος για $J_0 = 0$ και για $J_0 = 1$:

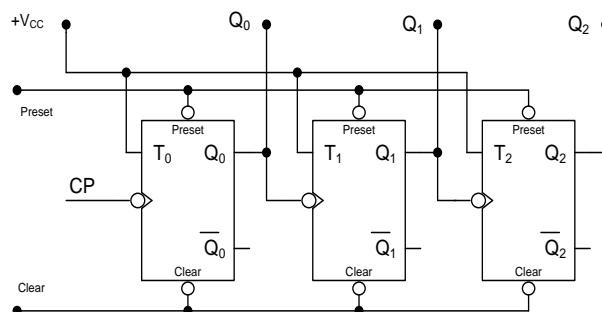
Παρούσα Κατάσταση			Είσοδοι FF				Επόμενη Κατάσταση				
Q_2	Q_1	Q_0	$J_2 = Q_1'$	$K_2 = Q_1'$	$J_1 = 1$	$K_1 = 1$	$J_0 = 0$	$K_0 = 0$	Q_2^+	Q_1^+	Q_0^+
1	1	0	0	0	1	1	0	0	1	0	0
							$J_0 = 1$	$K_0 = 1$			
1	1	0	0	0	1	1	1	0	1	0	1

Παρατήρηση: Βλέπουμε ότι για $J_0 = 1$ το σύστημα μεταβαίνει σε χρησιμοποιούμενη κατάσταση, ενώ για $J_0 = 0$ το σύστημα μεταβαίνει σε μη χρησιμοποιούμενη κατάσταση. Εάν εξετάσουμε τη συμπεριφορά του συστήματος και για τις άλλες μη χρησιμοποιούμενες καταστάσεις, θα δούμε ότι για $J_0 = 0$, το σύστημα θα απαριθμεί τις άρτιες (ζυγές) καταστάσεις. Άρα, για $J_0 = 1$ το σύστημα είναι αυτοδιορθούμενο, ενώ για $J_0 = 0$ το σύστημα είναι μη αυτοδιορθούμενο.

ΘΕΜΑ 3^ο (4,0 μονάδες)

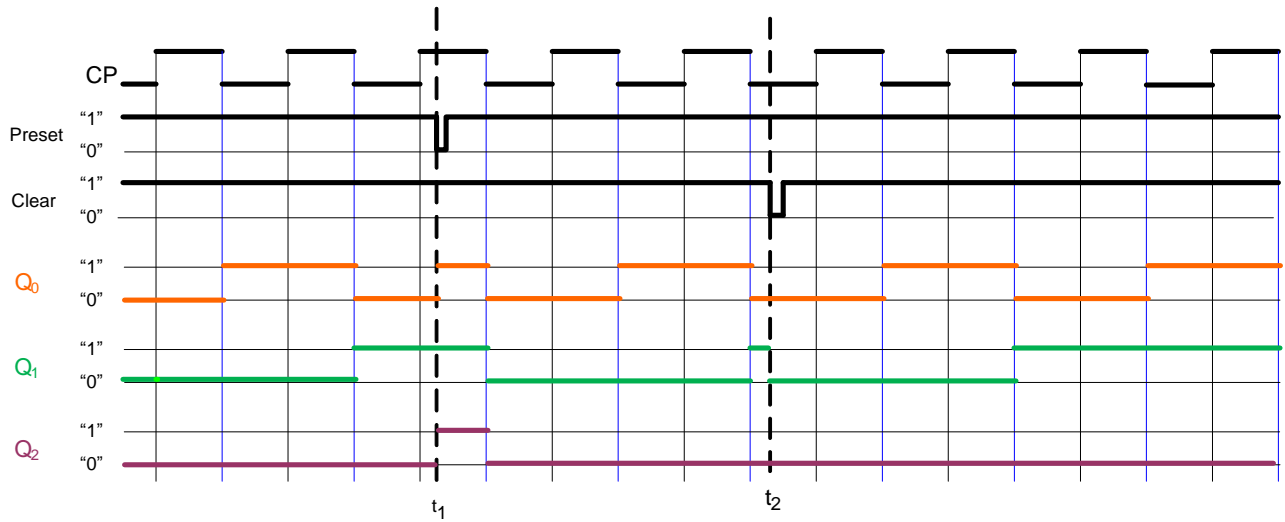
Στο παρακάτω Σχήμα 1 δίνεται το κύκλωμα ενός ασύγχρονου μετρητή αύξουσας μέτρησης, με αρχική κατάσταση $Q_2 Q_1 Q_0 = 000$.

1. Αν στο Preset τη χρονική στιγμή t_1 και στο Clear τη χρονική στιγμή t_2 δοθούν τα σήματα που φαίνονται στο Σχήμα 2 της επόμενης σελίδας, να συμπληρώσετε το διάγραμμα χρονισμού του μετρητή. (1,5 μον.)
2. Πού πρέπει να συνδεθούν οι ασύγχρονες εισοδοί Preset και Clear προκειμένου ο μετρητής να λειτουργήσει ως ένας πλήρης 3-bit μετρητής; (0,5 μον.)
3. Να δείξετε πώς ο μετρητής αυτός μπορεί να μετατραπεί σε μετρητή MOD(6) και να συμπληρώσετε το αντίστοιχο διάγραμμα χρονισμού (Σχήμα 3 στην επόμενη σελίδα). (2,0 μον.)



Λύση

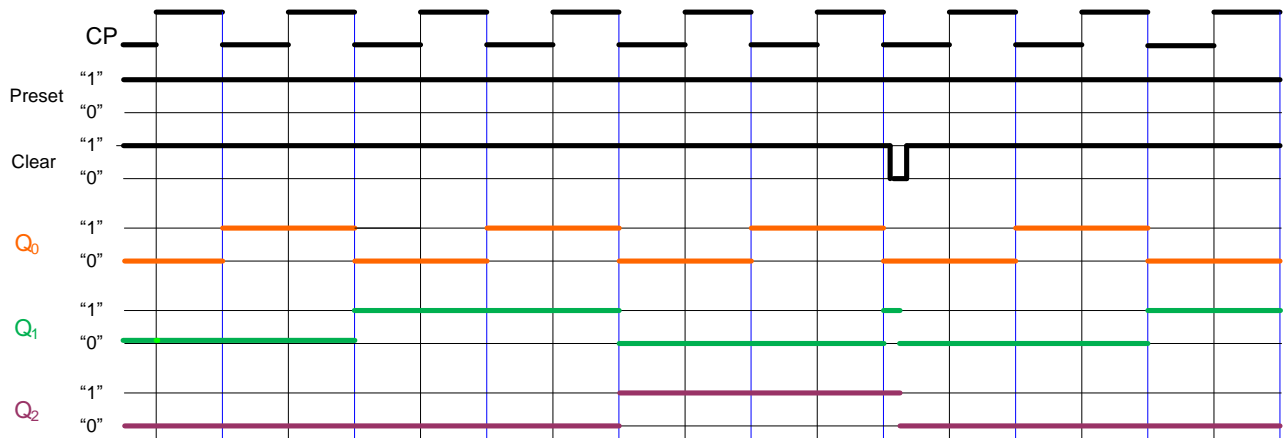
1.



Σχήμα 2

2. Επειδή όπως βλέπουμε από το κύκλωμα το Preset και το Clear είναι **active low**, προκειμένου ο μετρητής να λειτουργήσει ως ένας πλήρης 3-bit μετρητής θα πρέπει το Preset και το Clear να συνδεθούν στο λογικό '1' (+Vcc).

3.



Σχήμα 3

